

基于ARM®32位的Cortex®-M4微控制器，带16K字节至64K字节内部闪存、  
sLib、10个定时器、1个ADC、7个通信接口

## 功能

### ■ 内核：ARM®32位 Cortex®-M4 CPU

- 最高120 MHz工作频率，带存储器保护单元(MPU)，内建单周期乘法和硬件除法
- 具有DSP指令集

### ■ 存储器

- 16 K至64 K字节的内部闪存程序/数据存储器
- 4 K字节的启动程序存储器作启动加载程序(Bootloader)用，可一次性配置成一般用户程序和数据区
- sLib：将指定之主存储区设为执行代码安全库区，此区代码仅能调用无法读取
- 8 K到16 K字节的SRAM

### ■ 时钟、复位和电源管理

- 2.4至3.6伏供电和I/O引脚
- 上电/下电复位(POR/LVR)
- 电压监测器(PVM)
- 4至25 MHz晶体振荡器
- 内嵌经出厂调校的8 MHz RC振荡器(25 °C达1 %精度，-40 °C至+105 °C达2 %精度)
- 内嵌40 kHz RC振荡器
- 32.768 kHz晶体振荡器

### ■ 低功耗

- 睡眠、深度睡眠和待机模式

### ■ 1个12位A/D转换器，15个外部输入通道

- 转换范围：0至3.6 V

### ■ 1个比较器，5个外部输入通道和1个内部参考电压通道

### ■ 2个运算放大器

### ■ DMA：5通道DMA控制器

- 支持的外设：定时器、ADC、I²S、SPI、I²C和USART

### ■ 调试模式

- 串行单线调试(SWD)接口

### ■ 39个快速I/O端口

- 所有I/O口都可以映像到外部中断
- 几乎所有I/O可容忍5 V输入电压
- 所有均为快速I/O，寄存器存取速度最高f<sub>AHB</sub>

### ■ 10个定时器

- 1个16位7通道高级定时器，有6通道PWM输出，带死区控制和紧急刹车功能
- 5个16位定时器，每个定时器最多达4个用于输

入/输出/PWM或脉冲计数的通道并支持编码器模式

- 1个16位基本定时器

- 2个看门狗定时器

- 系统滴嗒定时器：24位递减计数器

### ■ ERTC：增强型RTC

### ■ 7个通信接口

- 2个I²C接口(支持SMBus/PMBus)
- 2个USART/UART接口(支持ISO7816接口、LIN、IrDA接口和调制解调器控制)
- 2个SPI接口，均可复用为I²S接口
- 红外发射器

### ■ CRC计算单元

### ■ 96位的芯片唯一码 (UID)

### ■ 封装

- LQFP48 7 x 7 mm
- LQFP32 7 x 7 mm
- QFN32 5 x 5 mm
- QFN32 4 x 4 mm
- QFN28 4 x 4 mm
- TSSOP20 6.5 x 4.4 mm

### ■ 选型列表

闪存存储器	型号
64 K 字节	AT32F421C8T7 AT32F421K8T7 AT32F421K8U7 AT32F421K8U7-4 AT32F421F8P7 AT32F421G8U7 AT32F421C8T7
32 K 字节	AT32F421C6T7 AT32F421K6T7 AT32F421K6U7 AT32F421K6U7-4 AT32F421F6P7 AT32F421G6U7
16 K 字节	AT32F421C4T7 AT32F421K4T7 AT32F421K4U7 AT32F421K4U7-4 AT32F421F4P7 AT32F421G4U7

## 目 录

1 系统架构 .....	23
1.1 系统概述 .....	23
1.1.1 ARM Cortex®-M4处理器 .....	23
1.1.2 位带 .....	24
1.1.3 中断和异常向量 .....	26
1.1.4 系统嘀嗒定时器 ( SysTick ) .....	28
1.1.5 复位流程 .....	28
1.2 寄存器描述缩写说明 .....	29
1.3 器件特征信息 .....	29
1.3.1 闪存容量寄存器 .....	29
1.3.2 器件电子签名 .....	30
2 存储器资源 .....	31
2.1 内部存储器地址映射 .....	31
2.2 Flash存储器 .....	31
2.3 SRAM存储器 .....	32
2.4 外设地址映射 .....	32
3 电源控制 ( PWC ) .....	35
3.1 简介 .....	35
3.2 主要特点 .....	35
3.3 上电下电复位 .....	35
3.4 电压检测器 ( PVM ) .....	36
3.5 电源域划分 .....	36
3.6 省电模式 .....	37
3.7 PWC寄存器 .....	38
3.7.1 电源控制寄存器 ( PWC_CTRL ) .....	38
3.7.2 电源控制及状态寄存器 ( PWC_CTRLSTS ) .....	39
3.7.3 电源控制寄存器2 ( PWC_CTRL2 ) .....	40
4 时钟和复位管理 ( CRM ) .....	41
4.1 时钟 .....	41
4.1.1 时钟源 .....	41
4.1.2 系统时钟 .....	42
4.1.3 外设时钟 .....	42

4.1.4	时钟失效检测 .....	43
4.1.5	自动滑顺频率切换 .....	43
4.1.6	内部时钟输出 .....	43
4.1.7	中断 .....	43
4.2	复位 .....	43
4.2.1	系统复位 .....	43
4.2.2	电池供电域复位 .....	44
4.3	CRM寄存器 .....	44
4.3.1	时钟控制寄存器 ( CRM_CTRL ) .....	44
4.3.2	时钟配置寄存器 ( CRM_CFG ) .....	45
4.3.3	时钟中断寄存器 ( CRM_CLKINT ) .....	47
4.3.4	APB2外设复位寄存器 ( CRM_APB2RST ) .....	48
4.3.5	APB1外设复位寄存器 ( CRM_APB1RST ) .....	49
4.3.6	AHB外设时钟使能寄存器 ( CRM_AHBEN ) .....	49
4.3.7	APB2外设时钟使能寄存器 ( CRM_APB2EN ) .....	50
4.3.8	APB1外设时钟使能寄存器 ( CRM_APB1EN ) .....	51
4.3.9	电池供电域控制寄存器 ( CRM_BPDC ) .....	51
4.3.10	控制/状态寄存器 ( CRM_CTRLSTS ) .....	52
4.3.11	AHB外设复位寄存器 ( CRM_AHBRST ) .....	53
4.3.12	PLL配置寄存器 ( CRM_PLL ) .....	53
4.3.13	额外寄存器1 ( CRM_MISC1 ) .....	54
4.3.14	额外寄存器2 ( CRM_MISC2 ) .....	54
5	内嵌闪存控制器 ( FLASH ) .....	55
5.1	FLASH介绍 .....	55
5.2	主存储器操作 .....	57
5.2.1	解锁/锁定 .....	57
5.2.2	擦除 .....	57
5.2.3	编程 .....	59
5.2.4	读取 .....	60
5.3	主存扩展区操作 .....	60
5.4	用户系统数据区操作 .....	61
5.4.1	解锁/锁定 .....	61
5.4.2	擦除 .....	61

5.4.3 编程 .....	62
5.4.4 读取 .....	63
5.5 闪存保护 .....	63
5.5.1 访问保护 .....	63
5.5.2 擦写保护 .....	64
5.6 读取性能 .....	64
5.7 特殊功能 .....	65
5.7.1 安全库区设定 .....	65
5.7.2 启动程序代码区域作为主存扩展使用 .....	66
5.7.3 CRC校验 .....	66
5.8 FLASH寄存器 .....	66
5.8.1 闪存性能选择寄存器 ( FLASH_PSR ) .....	67
5.8.2 闪存解锁寄存器 ( FLASH_UNLOCK ) .....	67
5.8.3 闪存用户系统数据解锁寄存器 ( FLASH_USD_UNLOCK ) .....	67
5.8.4 闪存状态寄存器 ( FLASH_STS ) .....	68
5.8.5 闪存控制寄存器 ( FLASH_CTRL ) .....	68
5.8.6 闪存地址寄存器 ( FLASH_ADDR ) .....	69
5.8.7 用户系统数据寄存器 ( FLASH_USD ) .....	69
5.8.8 擦除编程保护状态寄存器 ( FLASH_EPPS ) .....	69
5.8.9 闪存安全库区状态寄存器0 ( SLIB_STS0 ) .....	69
5.8.10 闪存安全库区状态寄存器1 ( SLIB_STS1 ) .....	70
5.8.11 闪存安全库区密码清除寄存器 ( SLIB_PWD_CLR ) .....	71
5.8.12 闪存安全库区额外状态寄存器 ( SLIB_MISC_STS ) .....	71
5.8.13 闪存CRC校验地址寄存器 ( FLASH_CRC_ADDR ) .....	71
5.8.14 闪存CRC校验控制寄存器 ( FLASH_CRC_CTRL ) .....	71
5.8.15 闪存CRC校验结果寄存器 ( FLASH_CRC_CHK ) .....	71
5.8.16 闪存安全库区密码设定寄存器 ( SLIB_SET_PWD ) .....	72
5.8.17 闪存安全库区地址设定寄存器 ( SLIB_SET_RANGE ) .....	72
5.8.18 主存扩展存储区域安全库区设定寄存器 ( EM_SLIB_SET ) .....	73
5.8.19 启动程序代码区模式设定寄存器 ( BTM_MODE_SET ) .....	73
5.8.20 闪存安全库区解锁寄存器 ( SLIB_UNLOCK ) .....	73
6 通用和复用功能I/O ( GPIO和IOMUX ) .....	74
6.1 简介 .....	74

6.2 功能描述 .....	74
6.2.1 GPIO结构 .....	74
6.2.2 GPIO复位状态 .....	74
6.2.3 通用功能输入配置 .....	75
6.2.4 模拟输入/输出配置 .....	75
6.2.5 通用功能输出配置 .....	75
6.2.6 GPIO端口保护 .....	76
6.2.7 IOMUX功能结构 .....	76
6.2.8 复用功能上下拉配置 .....	76
6.2.9 IOMUX功能输入/输出 .....	77
6.2.10 外设复用功能引脚配置 .....	78
6.2.11 IOMUX映射优先级 .....	78
6.2.12 外部中断/唤醒线 .....	79
6.3 GPIO寄存器 .....	79
6.3.1 GPIO配置寄存器 ( GPIOx_CFGR ) ( x=A..H ) .....	79
6.3.2 GPIO输出模式寄存器 ( GPIOx_OMODE ) ( x=A..H ) .....	79
6.3.3 GPIO电流推动/吸入能力切换控制寄存器 ( GPIOx_ODRVR ) ( x=A..H ) .....	80
6.3.4 GPIO上/下拉寄存器 ( GPIOx_PULL ) ( x=A..H ) .....	80
6.3.5 GPIO输入数据寄存器 ( GPIOx_IDT ) ( x=A..H ) .....	80
6.3.6 GPIO输出数据寄存器 ( GPIOx_ODT ) ( x=A..H ) .....	80
6.3.7 GPIO设置/清除寄存器 ( GPIOx_SCR ) ( x=A..H ) .....	80
6.3.8 GPIO写保护寄存器 ( GPIOx_WPR ) ( x=A..H ) .....	81
6.3.9 GPIO复用低位寄存器 ( GPIOx_MUXL ) ( x=A..H ) .....	81
6.3.10 GPIO复用高位寄存器 ( GPIOx_MUXH ) ( x=A..H ) .....	81
6.3.11 GPIO位清除寄存器 ( GPIOx_CLR ) ( x=A..H ) .....	82
6.3.12 极大电流推动/吸入能力切换控制寄存器 ( GPIOx_HDRV ) ( x=A..H ) .....	82
7 系统配置控制器 ( SCFG ) .....	83
7.1 简介 .....	83
7.2 SCFG寄存器 .....	83
7.2.1 SCFG配置寄存器1 ( SCFG_CFG1 ) .....	83
7.2.2 SCFG外部中断配置寄存器1 ( SCFG_EXINTC1 ) .....	84
7.2.3 SCFG外部中断配置寄存器2 ( SCFG_EXINTC2 ) .....	85
7.2.4 SCFG外部中断配置寄存器3 ( SCFG_EXINTC3 ) .....	85

7.2.5    SCFG外部中断配置寄存器4 ( SCFG_EXINTC4 ) .....	86
8    外部中断/事件控制器 ( EXINT ) .....	87
8.1    EXINT介绍 .....	87
8.2    功能描述和配置流程 .....	87
8.3    EXINT寄存器描述 .....	88
8.3.1    中断使能寄存器 ( EXINT_INTEN ) .....	88
8.3.2    事件使能寄存器 ( EXINT_EVTEN ) .....	88
8.3.3    极性配置寄存器1 ( EXINT_POLCFG1 ) .....	88
8.3.4    极性配置寄存器2 ( EXINT_POLCFG2 ) .....	89
8.3.5    软件触发寄存器 ( EXINT_SWTRG ) .....	89
8.3.6    中断状态寄存器 ( EXINT_INTSTS ) .....	89
9    DMA控制器 ( DMA ) .....	90
9.1    简介 .....	90
9.2    特性 .....	90
9.3    功能描述 .....	90
9.3.1    通道配置 .....	90
9.3.2    握手机制 .....	91
9.3.3    仲裁 .....	91
9.3.4    可编程数据传输宽度 .....	91
9.3.5    错误事件 .....	92
9.3.6    中断 .....	92
9.3.7    DMA固定请求映射 .....	92
9.4    DMA寄存器 .....	93
9.4.1    DMA状态寄存器 ( DMA_STS ) .....	94
9.4.2    DMA状态清除寄存器 ( DMA_CLR ) .....	95
9.4.3    DMA通道x配置寄存器 ( DMA_CxCTRL ) ( x = 1...5 ) .....	97
9.4.4    DMA通道x数据传输量寄存器 ( DMA_CxDTCNT ) ( x = 1...5 ) .....	98
9.4.5    DMA通道x外设地址寄存器 ( DMA_CxPADDR ) ( x = 1...5 ) .....	98
9.4.6    DMA通道x存储器地址寄存器 ( DMA_CxMADDR ) ( x = 1...5 ) .....	98
10    CRC计算单元 ( CRC ) .....	99
10.1    CRC介绍 .....	99
10.2    CRC功能说明 .....	99
10.3    CRC寄存器 .....	100

10.3.1 数据寄存器 ( CRC_DT ) .....	100
10.3.2 通用数据寄存器 ( CRC_CDT ) .....	100
10.3.3 控制寄存器 ( CRC_CTRL ) .....	101
10.3.4 初始化寄存器 ( CRC_IDT ) .....	101
10.3.5 生成多项式系数寄存器 ( CRC_POLY ) .....	101
11 I <sup>2</sup> C接口 .....	102
11.1 I <sup>2</sup> C简介 .....	102
11.2 I <sup>2</sup> C主要特点 .....	102
11.3 I <sup>2</sup> C总线特性 .....	102
11.4 I <sup>2</sup> C接口 .....	102
11.4.1 I <sup>2</sup> C从机通信流程 .....	105
11.4.2 I <sup>2</sup> C主机通信流程 .....	106
11.4.3 利用DMA传输 .....	111
11.4.4 SMBus .....	112
11.4.5 I <sup>2</sup> C中断请求 .....	113
11.4.6 I <sup>2</sup> C调试模式 .....	114
11.5 I <sup>2</sup> C寄存器描述 .....	114
11.5.1 控制寄存器1(I2C_CTRL1) .....	114
11.5.2 控制寄存器2(I2C_CTRL2) .....	116
11.5.3 自身地址寄存器1(I2C_OADDR1) .....	116
11.5.4 自身地址寄存器2(I2C_OADDR2) .....	116
11.5.5 数据寄存器(I2C_DT) .....	117
11.5.6 状态寄存器1(I2C_STS1) .....	117
11.5.7 状态寄存器2(I2C_STS2) .....	119
11.5.8 时钟控制寄存器(I2C_CLKCTRL) .....	119
11.5.9 时钟上升寄存器(I2C_TMRISE) .....	120
12 通用同步异步收发器 ( USART ) .....	121
12.1 USART介绍 .....	121
12.2 全双工半双工选择器简述和配置流程 .....	122
12.3 模式选择器简述和配置流程 .....	122
12.3.1 模式选择器简述 .....	122
12.3.2 模式选择器配置方法 .....	123
12.4 USART帧格式简述和配置流程 .....	125

12.5 DMA传输简述和配置流程 .....	127
12.5.1 DMA发送配置流程 .....	127
12.5.2 DMA接收配置流程 .....	127
12.6 波特率发生器简述及配置流程 .....	128
12.6.1 波特率发生器简述 .....	128
12.6.2 波特率发生器配置方法 .....	128
12.7 发送器简述和配置流程 .....	128
12.7.1 发送器简述 .....	128
12.7.2 发送器配置流程 .....	129
12.8 接收器简述和配置流程 .....	129
12.8.1 接收器简述 .....	129
12.8.2 接收器配置流程 .....	130
12.8.3 起始侦测和噪声检测 .....	130
12.9 Tx/Rx可配置引脚互换 .....	131
12.10 中断 .....	132
12.11 I/O管脚控制 .....	132
12.12 USART寄存器描述 .....	133
12.12.1 状态寄存器 ( USART_STS ) .....	133
12.12.2 数据寄存器 ( USART_DT ) .....	134
12.12.3 波特比率寄存器 ( USART_BAUDR ) .....	134
12.12.4 控制寄存器1 ( USART_CTRL1 ) .....	134
12.12.5 控制寄存器2 ( USART_CTRL2 ) .....	135
12.12.6 控制寄存器3 ( USART_CTRL3 ) .....	136
12.12.7 保护时间和预分频寄存器 ( USART_GDIV ) .....	137
13 串行外设接口 ( SPI ) .....	138
13.1 串行外设接口 ( SPI ) 简介 .....	138
13.2 SPI功能描述 .....	138
13.2.1 SPI简述 .....	138
13.2.2 全双工半双工选择器简述和配置流程 .....	139
13.2.3 CS控制器简述和配置流程 .....	140
13.2.4 SPI_SCK控制器简述和配置流程 .....	141
13.2.5 CRC简述和配置流程 .....	141
13.2.6 DMA传输简述和配置流程 .....	142

13.2.7	发送器简述和配置流程 .....	142
13.2.8	接收器简述和配置流程 .....	143
13.2.9	Motorola模式通信时序 .....	143
13.2.10	中断 .....	146
13.2.11	IO管脚控制 .....	146
13.2.12	注意事项 .....	146
13.3	I <sup>2</sup> S功能描述 .....	146
13.3.1	I <sup>2</sup> S简述 .....	146
13.3.2	操作模式选择器简述和配置流程 .....	147
13.3.3	音频协议选择器简述和配置流程 .....	149
13.3.4	I2S_CLK控制器简述和配置流程 .....	150
13.3.5	DMA传输简述和配置流程 .....	151
13.3.6	发送器接收器简述和配置流程 .....	152
13.3.7	I2S通信时序 .....	153
13.3.8	中断 .....	153
13.3.9	IO管脚控制 .....	153
13.4	SPI寄存器 .....	154
13.4.1	SPI控制寄存器1 ( SPI_CTRL1 ) ( I <sup>2</sup> S模式下不使用 ) .....	154
13.4.2	SPI控制寄存器2 ( SPI_CTRL2 ) .....	155
13.4.3	SPI状态寄存器 ( SPI_STS ) .....	156
13.4.4	SPI数据寄存器 ( SPI_DT ) .....	157
13.4.5	SPICRC多项式寄存器 ( SPI_CPOLY ) ( I <sup>2</sup> S模式下不使用 ) .....	157
13.4.6	SPIRxCRC寄存器 ( SPI_RCRC ) ( I <sup>2</sup> S模式下不使用 ) .....	157
13.4.7	SPITxCRC寄存器 ( SPI_TCRC ) .....	157
13.4.8	SPI_I2S配置寄存器 ( SPI_I2SCTRL ) .....	157
13.4.9	SPI_I2S预分频寄存器 ( SPI_I2SCLKP ) .....	158
14	定时器 ( TIMER ) .....	159
14.1	基本定时器 ( TMR6 ) .....	159
14.1.1	TMR6简介 .....	159
14.1.2	TMR6主要特性 .....	159
14.1.3	TMR6功能 .....	160
14.1.3.1	计数时钟 .....	160
14.1.3.2	计数模式 .....	160
14.1.3.3	调试模式 .....	161

14.1.4 TMR6寄存器 .....	161
14.1.4.1 TMR6控制寄存器1 (TMRx_CTRL1) .....	162
14.1.4.2 TMR6控制寄存器2 (TMRx_CTRL2) .....	162
14.1.4.3 TMR6 DMA/中断使能寄存器 (TMRx_IDEN) .....	162
14.1.4.4 TMR6中断状态寄存器 (TMRx_ISTS) .....	163
14.1.4.5 TMR6软件事件寄存器 (TMRx_SWEVT) .....	163
14.1.4.6 TMR6计数值 (TMRx_CVAL) .....	163
14.1.4.7 TMR6分频系数 (TMRx_DIV) .....	163
14.1.4.8 TMR6周期寄存器 (TMRx_PR) .....	163
14.2 通用定时器 (TMR3) .....	164
14.2.1 TMR3简介 .....	164
14.2.2 TMR3主要功能 .....	164
14.2.3 TMR3功能描述 .....	164
14.2.3.1 计数时钟 .....	164
14.2.3.2 计数模式 .....	167
14.2.3.3 TMR输入部分 .....	170
14.2.3.4 TMR输出部分 .....	172
14.2.3.5 定时器同步 .....	175
14.2.3.6 调试模式 .....	177
14.2.4 TMR3寄存器描述 .....	178
14.2.4.1 控制寄存器1 (TMR3_CTRL1) .....	178
14.2.4.2 控制寄存器2 (TMR3_CTRL2) .....	179
14.2.4.3 次定时器控制寄存器 (TMR3_STCTRL) .....	179
14.2.4.4 DMA/中断使能寄存器 (TMR3_IDEN) .....	180
14.2.4.5 中断状态寄存器 (TMR3_ISTS) .....	181
14.2.4.6 软件事件寄存器 (TMR3_SWEVT) .....	182
14.2.4.7 通道模式寄存器1 (TMR3_CM1) .....	182
14.2.4.8 通道模式寄存器2 (TMR3_CM2) .....	184
14.2.4.9 通道控制寄存器 (TMR3_CCTRL) .....	185
14.2.4.10 计数值 (TMR3_CVAL) .....	186
14.2.4.11 分频系数 (TMR3_DIV) .....	186
14.2.4.12 周期寄存器 (TMR3_PR) .....	186
14.2.4.13 通道1数据寄存器 (TMR3_C1DT) .....	186
14.2.4.14 通道2数据寄存器 (TMR3_C2DT) .....	186
14.2.4.15 通道3数据寄存器 (TMR3_C3DT) .....	187
14.2.4.16 通道4数据寄存器 (TMR3_C4DT) .....	187
14.2.4.17 DMA控制寄存器 (TMR3_DMACTRL) .....	187
14.2.4.18 DMA数据寄存器 (TMR3_DMADT) .....	187
14.3 通用定时器 (TMR14) .....	188
14.3.1 TMR14简介 .....	188

14.3.2 TMR14主要特性 .....	188
14.3.3 TMR14功能描述 .....	188
14.3.3.1 计数时钟 .....	188
14.3.3.2 计数模式 .....	189
14.3.3.3 TMR输入部分 .....	190
14.3.3.4 TMR输出部分 .....	191
14.3.3.5 调试模式 .....	192
14.3.4 TMR14寄存器描述 .....	192
14.3.4.1 控制寄存器1 (TMR14_CTRL1) .....	193
14.3.4.2 中断使能寄存器 (TMR14_IDEN) .....	193
14.3.4.3 中断状态寄存器 (TMR14ISTS) .....	193
14.3.4.4 软件事件寄存器 (TMR14_SWEVT) .....	194
14.3.4.5 通道模式寄存器1 (TMR14_CM1) .....	194
14.3.4.6 通道控制寄存器 (TMR14_CCTRL) .....	196
14.3.4.7 计数值 (TMR14_CVAL) .....	196
14.3.4.8 预分频器 (TMR14_DIV) .....	196
14.3.4.9 周期寄存器 (TMR14_PR) .....	196
14.3.4.10 通道1数据寄存器 (TMR14_C1DT) .....	197
14.3.4.11 通道输入重映射寄存器 (TMR14_RMP) .....	197
14.4 通用定时器 (TMR15) .....	198
14.4.1 TMR15简介 .....	198
14.4.2 TMR15主要特性 .....	198
14.4.3 TMR15功能描述 .....	198
14.4.3.1 计数时钟 .....	198
14.4.3.2 计数模式 .....	200
14.4.3.3 TMR输入部分 .....	201
14.4.3.4 TMR输出部分 .....	203
14.4.3.5 TMR刹车功能 .....	206
14.4.3.6 TMR同步 .....	207
14.4.3.7 调试模式 .....	208
14.4.4 TMR15寄存器描述 .....	209
14.4.4.1 TMR15控制寄存器1 (TMR15_CTRL1) .....	209
14.4.4.2 TMR15控制寄存器2 (TMR15_CTRL2) .....	210
14.4.4.3 TMR15次定时器控制寄存器 (TMR15_STCTRL) .....	210
14.4.4.4 TMR15 DMA/中断使能寄存器 (TMR15_IDEN) .....	211
14.4.4.5 TMR15中断状态寄存器 (TMR15ISTS) .....	211
14.4.4.6 TMR15软件事件寄存器 (TMR15_SWEVT) .....	212
14.4.4.7 TMR15通道模式寄存器1 (TMR15_CM1) .....	213
14.4.4.8 TMR15通道控制寄存器 (TMR15_CCTRL) .....	215
14.4.4.9 TMR15计数值 (TMR15_CVAL) .....	216

14.4.4.10 TMR15预分频器 (TMR15_DIV) .....	216
14.4.4.11 TMR15周期寄存器 (TMR15_PR) .....	216
14.4.4.12 TMR15重复周期寄存器 (TMR15_RPR) .....	216
14.4.4.13 TMR15通道1数据寄存器 (TMR15_C1DT) .....	217
14.4.4.14 TMR15通道2数据寄存器 (TMR15_C2DT) .....	217
14.4.4.15 TMR15刹车寄存器 (TMR15_BRK) .....	217
14.4.4.16 TMR15 DMA控制寄存器 (TMR15_DMACTRL) .....	218
14.4.4.17 TMR15 DMA数据寄存器 (TMR15_DMADT) .....	218
14.5 通用定时器 (TMR16和TMR17) .....	219
14.5.1 TMR16和TMR17简介 .....	219
14.5.2 TMR16和TMR17主要功能 .....	219
14.5.3 TMR16和TMR17功能描述 .....	219
14.5.3.1 计数时钟 .....	219
14.5.3.2 计数模式 .....	220
14.5.3.3 TMR输入部分 .....	221
14.5.3.4 TMR输出部分 .....	222
14.5.3.5 TMR刹车功能 .....	225
14.5.3.6 调试模式 .....	226
14.5.4 TMR16和TMR17寄存器描述 .....	227
14.5.4.1 TMR16和TMR17控制寄存器1 (TMRx_CTRL1) .....	227
14.5.4.2 TMR16和TMR17控制寄存器2 (TMRx_CTRL2) .....	228
14.5.4.3 TMR16和TMR17 DMA/中断使能寄存器 (TMRx_IDEN) .....	228
14.5.4.4 TMR16和TMR17中断状态寄存器 (TMRxISTS) .....	229
14.5.4.5 TMR16和TMR17软件事件寄存器 (TMRx_SWEVT) .....	229
14.5.4.6 TMR16和TMR17通道模式寄存器1 (TMRx_CM1) .....	230
14.5.4.7 TMR16和TMR17通道控制寄存器 (TMRx_CCTRL) .....	231
14.5.4.8 TMR16和TMR17计数值 (TMRx_CVAL) .....	232
14.5.4.9 TMR16和TMR17预分频器 (TMRx_DIV) .....	233
14.5.4.10 TMR16和TMR17周期寄存器 (TMRx_PR) .....	233
14.5.4.11 TMR16和TMR17重复周期寄存器 (TMRx_RPR) .....	233
14.5.4.12 TMR16和TMR17通道1数据寄存器 (TMRx_C1DT) .....	233
14.5.4.13 TMR16和TMR17刹车寄存器 (TMRx_BRK) .....	233
14.5.4.14 TMR16和TMR17 DMA控制寄存器 (TMRx_DMACTRL) .....	234
14.5.4.15 TMR16和TMR17 DMA数据寄存器 (TMRx_DMADT) .....	234
14.6 高级控制定时器 (TMR1) .....	235
14.6.1 TMR1简介 .....	235
14.6.2 TMR1主要特性 .....	235
14.6.3 TMR1功能描述 .....	235
14.6.3.1 计数时钟 .....	235
14.6.3.2 计数模式 .....	238

14.6.3.3 TMR输入部分 .....	242
14.6.3.4 TMR输出部分 .....	245
14.6.3.5 TMR刹车功能 .....	249
14.6.3.6 TMR同步 .....	250
14.6.3.7 调试模式 .....	251
14.6.4 TMR1寄存器描述 .....	252
14.6.4.1 TMR1控制寄存器1 (TMR1_CTRL1) .....	252
14.6.4.2 TMR1控制寄存器2 (TMR1_CTRL2) .....	253
14.6.4.3 TMR1次定时器控制寄存器 (TMR1_STCTRL) .....	254
14.6.4.4 TMR1 DMA/中断使能寄存器 (TMR1_IDEN) .....	255
14.6.4.5 TMR1中断状态寄存器 (TMR1_ISTS) .....	256
14.6.4.6 TMR1软件事件寄存器 (TMR1_SWEVT) .....	257
14.6.4.7 TMR1通道模式寄存器1 (TMR1_CM1) .....	257
14.6.4.8 TMR1通道模式寄存器2 (TMR1_CM2) .....	259
14.6.4.9 TMR1通道控制寄存器 (TMR1_CCTRL) .....	260
14.6.4.10 TMR1计数值 (TMR1_CVAL) .....	262
14.6.4.11 TMR1预分频器 (TMR1_DIV) .....	262
14.6.4.12 TMR1周期寄存器 (TMR1_PR) .....	262
14.6.4.13 TMR1重复周期寄存器 (TMR1_RPR) .....	262
14.6.4.14 TMR1通道1数据寄存器 (TMR1_C1DT) .....	262
14.6.4.15 TMR1通道2数据寄存器 (TMR1_C2DT) .....	262
14.6.4.16 TMR1通道3数据寄存器 (TMR1_C3DT) .....	263
14.6.4.17 TMR1通道4数据寄存器 (TMR1_C4DT) .....	263
14.6.4.18 TMR1刹车寄存器 (TMR1_BRK) .....	263
14.6.4.19 TMR1 DMA控制寄存器 (TMR1_DMACTRL) .....	264
14.6.4.20 TMR1 DMA数据寄存器 (TMR1_DMADT) .....	264
15 窗口看门狗 (WWDT) .....	265
15.1 WWDT简介 .....	265
15.2 WWDT主要特性 .....	265
15.3 WWDT功能描述 .....	265
15.4 调试模式 .....	266
15.5 WWDT寄存器 .....	266
15.5.1 控制寄存器 (WWDT_CTRL) .....	266
15.5.2 配置寄存器 (WWDT_CFG) .....	266
15.5.3 状态寄存器 (WWDT_STS) .....	267
16 看门狗 (WDT) .....	268
16.1 WDT简介 .....	268
16.2 WDT主要特性 .....	268

16.3 WDT功能描述 .....	268
16.4 调试模式 .....	269
16.5 WDT寄存器 .....	269
16.5.1 命令寄存器 ( WDT_CMD ) .....	269
16.5.2 预分频寄存器 ( WDT_DIV ) .....	269
16.5.3 重装载寄存器 ( WDT_RLD ) .....	270
16.5.4 状态寄存器 ( WDT_STS ) .....	270
17 实时时钟 ( ERTC ) .....	271
17.1 ERTC简介 .....	271
17.2 ERTC主要特性 .....	271
17.3 ERTC功能说明 .....	271
17.3.1 ERTC时钟 .....	271
17.3.2 ERTC初始化 .....	272
17.3.3 ERTC校准 .....	273
17.3.4 时间戳 .....	274
17.3.5 入侵检测 .....	274
17.3.6 复用功能输出 .....	274
17.3.7 ERTC唤醒 .....	275
17.4 ERTC寄存器 .....	275
17.4.1 ERTC时间寄存器(ERTC_TIME) .....	276
17.4.2 ERTC日期寄存器(ERTC_DATE) .....	276
17.4.3 ERTC控制寄存器(ERTC_CTRL) .....	276
17.4.4 ERTC初始化和状态寄存器(ERTC_STS) .....	277
17.4.5 ERTC预分频器寄存器(ERTC_DIV) .....	278
17.4.6 ERTC闹钟A寄存器(ERTC_ALA) .....	278
17.4.7 ERTC写保护寄存器(ERTC_WP) .....	279
17.4.8 ERTC亚秒寄存器(ERTC_SBS) .....	279
17.4.9 ERTC时间微调寄存器(ERTC_TADJ) .....	279
17.4.10 ERTC时间戳时间寄存器(ERTC_TSTM) .....	279
17.4.11 ERTC时间戳日期寄存器(ERTC_TS DT) .....	280
17.4.12 ERTC时间戳亚秒寄存器(ERTC_TSSBS) .....	280
17.4.13 ERTC精密校准寄存器(ERTC_SCAL) .....	280
17.4.14 ERTC入侵配置寄存器(ERTC_TAMP) .....	280

17.4.15 ERTC闹钟A亚秒寄存器(ERTC_ALASBS) .....	281
17.4.16 ERTC电池供电数据寄存器(ERTC_BPRx) .....	281
18 模拟/数字转换 ( ADC ) .....	282
18.1 ADC简介 .....	282
18.2 ADC主要特征 .....	282
18.3 ADC架构 .....	283
18.4 ADC功能介绍 .....	283
18.4.1 通道管理 .....	283
18.4.1.1 内部温度传感器 .....	284
18.4.1.2 内部参考电压 .....	284
18.4.2 ADC操作流程 .....	284
18.4.2.1 上电与校准 .....	284
18.4.2.2 触发 .....	285
18.4.2.3 采样与转换时序 .....	285
18.4.3 转换顺序管理 .....	286
18.4.3.1 序列模式 .....	286
18.4.3.2 抢占自动转换模式 .....	286
18.4.3.3 反复模式 .....	286
18.4.3.4 分割模式 .....	287
18.4.4 数据管理 .....	287
18.4.4.1 数据内容处理 .....	287
18.4.4.2 数据获取 .....	288
18.4.5 电压监测 .....	288
18.4.6 状态标志与中断 .....	288
18.5 ADC寄存器 .....	288
18.5.1 ADC状态寄存器 ( ADC_STS ) .....	289
18.5.2 ADC控制寄存器1 ( ADC_CTRL1 ) .....	289
18.5.3 ADC控制寄存器2 ( ADC_CTRL2 ) .....	291
18.5.4 ADC采样时间寄存器1 ( ADC_SPT1 ) .....	293
18.5.5 ADC采样时间寄存器2 ( ADC_SPT2 ) .....	294
18.5.6 ADC抢占通道数据偏移寄存器x ( ADC_PCDTOx ) ( x=1..4 ) .....	296
18.5.7 ADC电压监测高边界寄存器 ( ADC_VMHB ) .....	297
18.5.8 ADC电压监测低边界寄存器 ( ADC_VMLB ) .....	297
18.5.9 ADC普通序列寄存器1 ( ADC_OSQ1 ) .....	297
18.5.10 ADC普通序列寄存器2 ( ADC_OSQ2 ) .....	297

18.5.11 ADC普通序列寄存器3 ( ADC_OSQ3 ) .....	298
18.5.12 ADC抢占序列寄存器 ( ADC_PSQ ) .....	298
18.5.13 ADC抢占数据寄存器x ( ADC_PDTx ) ( x= 1..4 ) .....	299
18.5.14 ADC普通数据寄存器 ( ADC_ODT ) .....	299
19 比较器 ( CMP ) .....	300
19.1 简介 .....	300
19.2 主要特性 .....	300
19.3 中断管理 .....	300
19.4 设计提示 .....	300
19.5 功能描述 .....	301
19.5.1 模拟比较器 .....	301
19.5.2 干扰滤波器 .....	301
19.6 CMP寄存器 .....	302
19.6.1 比较器控制和状态寄存器1(CMP_CTRLSTS) .....	302
19.6.2 干扰滤波器使能寄存器(G_FILTER_EN) .....	303
19.6.3 干扰滤波器高脉冲数(HIGH_PULSE) .....	304
19.6.4 干扰滤波器低脉冲数(LOW_PULSE) .....	304
20 运算放大器 ( OP ) .....	305
20.1 简介 .....	305
20.2 主要特性 .....	305
20.3 功能描述 .....	305
21 红外线接口 ( IRTMR ) .....	307
22 调试 ( DEBUG ) .....	308
22.1 简介 .....	308
22.2 调试与跟踪功能 .....	308
22.3 I/O控制 .....	308
22.4 DEBUG寄存器 .....	308
22.4.1 DEBUG设备ID ( DEBUG_IDCODE ) .....	308
22.4.2 DEBUG控制寄存器 ( DEBUG_CTRL ) .....	309
23 版本历史 .....	311

## 图目录

图 1-1 AT32F421 系列微控制器系统架构 .....	23
图 1-2 Cortex®-M4 内部框图 .....	24
图 1-3 位带区与位带别名区的膨胀关系图 A .....	24
图 1-4 位带区与位带别名区的膨胀关系图 B .....	25
图 1-5 复位流程 .....	28
图 1-6 MSP 及 PC 初始化的一个范例 .....	28
图 2-1 AT32F421 地址映射 .....	31
图 3-1 各电源域框图 .....	35
图 3-2 上电/下电复位波形图 .....	36
图 3-3 PVM 的阈值与输出 .....	36
图 4-1 AT32F421 时钟结构图 .....	41
图 4-2 系统复位电路 .....	44
图 5-1 主存储器扇区擦除流程图 .....	58
图 5-2 主存储器整片擦除流程图 .....	59
图 5-3 主存储器编程流程图 .....	60
图 5-4 系统数据区擦除图 .....	62
图 5-5 系统数据区编程图 .....	63
图 6-1 GPIO 基本结构 .....	74
图 6-2 IOMUX 复用结构 .....	76
图 8-1 外部中断/事件控制器框图 .....	87
图 9-1 DMA 框图 .....	90
图 9-2 请求/应答对后重新仲裁 .....	91
图 9-3 PWIDHT: byte, MWIDHT: half-word .....	91
图 9-4 PWIDHT: half-word, MWIDHT: word .....	92
图 9-5 PWIDHT: word, MWIDHT: byte .....	92
图 10-1 CRC 计算单元框图 .....	99
图 11-1 I²C 总线协议 .....	102
图 11-2 I²C 的功能框图 .....	103
图 11-3 从发送器的传送序列图 .....	105
图 11-4 从接收器的传送序列图 .....	106
图 11-5 主发送器传送序列图 .....	107
图 11-6 主接收器传送序列图 .....	108
图 11-7 N>2 主接收器传送序列图 .....	109
图 11-8 N=2 主接收器传送序列图 .....	110
图 11-9 N=1 主接收器传送序列图 .....	111
图 12-1 USART 框图 .....	121
图 12-2 LIN 模式下的 BFF 检测与 FERR 检测 .....	123
图 12-3 Smartcard frame format .....	123
图 12-4 IrDA DATA(3/16)-普通模式 .....	124
图 12-5 Hardware flow control .....	124
图 12-6 Mute mode using Idle line or Address mark detection .....	125
图 12-7 8-bit format USART 同步模式 .....	125
图 12-8 字长设置 .....	126
图 12-9 配置停止位 .....	127
图 12-10 发送时 TDC/TDBE 的变化情况 .....	129
图 12-11 检测噪声的数据采样 .....	131
图 12-12 Tx/Rx 可配置引脚互换 .....	132
图 12-13 USART 中断映像图 .....	132
图 13-1 SPI 框图 .....	138
图 13-2 SPI 双线单向全双工连接示意图 .....	139
图 13-3 SPI 作主机单线单向只收连接示意图 .....	139

图 13-4 SPI 作从机单线单向只收连接示意图.....	140
图 13-5 SPI 作单线双向半双工连接示意图 .....	140
图 13-6 主机全双工通信 .....	144
图 13-7 从机全双工通信 .....	144
图 13-8 主机半双工发送通信.....	144
图 13-9 从机半双工接收通信.....	145
图 13-10 从机半双工发送通信.....	145
图 13-11 从机半双工接收通信 .....	145
图 13-12 SPI 中断 .....	146
图 13-13 I <sup>2</sup> S 框图 .....	147
图 13-14 I <sup>2</sup> S 从设备发送连接示意图 .....	148
图 13-15 I <sup>2</sup> S 从设备接收连接示意图 .....	148
图 13-16 I <sup>2</sup> S 主设备发送连接示意图 .....	148
图 13-17 I <sup>2</sup> S 主设备接收连接示意图 .....	149
图 13-18 SPI 作主机 CK & MCK 来源示意图 .....	150
图 13-19 各音频标准时序 .....	153
图 13-20 I <sup>2</sup> S 中断 .....	153
图 14-1 基本定时器框图 .....	160
图 14-2 使用 CK_INT 且分频系数为 1.....	160
图 14-3 计数器基本结构 .....	161
图 14-4 PRBEN=0 时的溢出事件 .....	161
图 14-5 PRBEN=1 时的溢出事件 .....	161
图 14-6 计数器时序图, 内部时钟分频因子为 4 .....	161
图 14-7 通用定时器框图 .....	164
图 14-8 计数时钟 .....	164
图 14-9 使用 CK_INT 计数, TMRx_DIV=0x0, 周期寄存器 TMRx_PR=0x16 .....	165
图 14-10 外部时钟模式 A 框图 .....	166
图 14-11 使用外部时钟模式 A 计数, PR=0x32, DIV=0x0 .....	166
图 14-12 外部时钟模式 B 框图 .....	166
图 14-13 使用外部时钟模式 B 计数, PR=0x32, DIV=0x0 .....	166
图 14-14 当预分频器的参数从 1 变到 4 时, 计数器的时序图 .....	167
图 14-15 计数器基本结构 .....	167
图 14-16 PRBEN=0 时的溢出事件 .....	168
图 14-17 PRBEN=1 时的溢出事件 .....	168
图 14-18 计数器时序图, 内部时钟分频因子为 4 .....	168
图 14-19 计数器时序图, 内部时钟分频因子为 1, TMRx_PR=0x32 .....	169
图 14-20 编码模式结构 .....	169
图 14-21 编码模式计数实例 (编码器模式 C) .....	170
图 14-22 输入/输出通道 1 的主电路 .....	170
图 14-23 通道 1 输入部分 .....	171
图 14-24 PWM 输入模式配置实例 .....	172
图 14-25 PWM 输入模式 .....	172
图 14-26 捕获/比较通道的输出部分 (通道 1 至 4) .....	172
图 14-27 计数值与 C1DT 值匹配时翻转 C1ORAW .....	173
图 14-28 向上计数下 PWM 模式 A .....	174
图 14-29 中央双向对齐计数下 PWM 模式 A .....	174
图 14-30 单周期模式 .....	174
图 14-31 EXT 清除 CxORAW(PWM 模式 A).....	175
图 14-32 复位模式例子 .....	175
图 14-33 挂起模式下例子 .....	176
图 14-34 触发器模式例子 .....	176
图 14-35 主/次定时器连接框图 .....	176

图 14-36 主定时器启动次定时器例子	177
图 14-37 外部触发同时启动主、次定时器	177
图 14-38 通用定时器 TMR14 框图	188
图 14-39 计数时钟	188
图 14-40 使用 CK_INT 计数, TMRx_DIV=0x0, 周期寄存器 TMRx_PR=0x16	188
图 14-41 当预分频器的参数从 1 变到 4 时, 计数器的时序图	189
图 14-42 计数器基本结构	189
图 14-43 PRBEN=0 时的溢出事件	189
图 14-44 PRBEN=1 时的溢出事件	190
图 14-45 输入/输出通道 1 的主电路	190
图 14-46 通道 1 输入部分	190
图 14-47 捕获/比较通道的输出部分 (通道 1)	191
图 14-48 计数值与 C1DT 值匹配时翻转 C1ORAW	192
图 14-49 向上计数下 PWM 模式 A	192
图 14-50 TMR15 定时器框图	198
图 14-51 计数时钟	198
图 14-52 使用 CK_INT 计数, TMRx_DIV=0x0, 周期寄存器 TMRx_PR=0x16	199
图 14-53 外部时钟模式 A 框图	199
图 14-54 使用外部时钟模式 A 计数, PR=0x32, DIV=0x0	199
图 14-55 当预分频器的参数从 1 变到 4 时, 计数器的时序图	200
图 14-56 计数器基本结构	201
图 14-57 PRBEN=0 时的溢出事件	201
图 14-58 PRBEN=1 时的溢出事件	201
图 14-59 RPR=2 时的 OVFIF	201
图 14-60 输入/输出通道 1 的主电路	202
图 14-61 通道 1 输入部分	202
图 14-62 PWM 输入模式配置实例	203
图 14-63 PWM 输入模式	203
图 14-64 通道 1 输出部分	204
图 14-65 通道 2 输出部分	204
图 14-66 计数值与 C1DT 值匹配时翻转 C1ORAW	205
图 14-67 向上计数下 PWM 模式 A	205
图 14-68 单周期模式	205
图 14-69 带死区插入的互补输出	206
图 14-70 TMR 输出控制	207
图 14-71 TMR 刹车功能的例子	207
图 14-72 复位模式例子	208
图 14-73 挂起模式下例子	208
图 14-74 触发器模式例子	208
图 14-75 TMR16、17 定时器框图	219
图 14-76 计数时钟	219
图 14-77 使用 CK_INT 计数, TMRx_DIV=0x0, 周期寄存器 TMRx_PR=0x16	220
图 14-78 计数器基本结构	220
图 14-79 PRBEN=0 时的溢出事件	220
图 14-80 PRBEN=1 时的溢出事件	221
图 14-81 RPR=2 时的 OVFIF	221
图 14-82 输入/输出通道 1 的主电路	221
图 14-83 通道 1 输入部分	222
图 14-84 通道 1 输出部分	222
图 14-85 计数值与 C1DT 值匹配时翻转 C1ORAW	224
图 14-86 向上计数下 PWM 模式 A	224
图 14-87 单周期模式	224

图 14-88 带死区插入的互补输出 .....	225
图 14-89 TMR 输出控制 .....	226
图 14-90 TMR 刹车功能的例子 .....	226
图 14-91 高级控制定时器框图 .....	235
图 14-92 计数时钟 .....	236
图 14-93 使用 CK_INT 计数, TMRx_DIV=0x0, 周期寄存器 TMRx_PR=0x16 .....	236
图 14-94 外部时钟模式 A 框图 .....	237
图 14-95 使用外部时钟模式 A 计数, PR=0x32, DIV=0x0 .....	237
图 14-96 外部时钟模式 B 框图 .....	237
图 14-97 使用外部时钟模式 B 计数, PR=0x32, DIV=0x0 .....	237
图 14-98 当预分频器的参数从 1 变到 4 时, 计数器的时序图 .....	238
图 14-99 计数器基本结构 .....	239
图 14-100 PRBEN=0 时的溢出事件 .....	239
图 14-101 PRBEN=1 时的溢出事件 .....	239
图 14-102 计数器时序图, 内部时钟分频因子为 4 .....	239
图 14-103 计数器时序图, 内部时钟分频因子为 1, TMRx_PR=0x32 .....	240
图 14-104 向上计数模式和中央双向对齐计数模式时 OVFIF .....	241
图 14-105 编码模式结构 .....	241
图 14-106 编码模式计数实例 (编码器模式 C) .....	242
图 14-107 输入/输出通道 1 的主电路 .....	243
图 14-108 通道 1 输入部分 .....	243
图 14-109 PWM 输入模式配置实例 .....	244
图 14-110 PWM 输入模式 .....	245
图 14-111 通道 1 至 3 输出部分 .....	245
图 14-112 通道 4 输出部分 .....	245
图 14-113 计数值与 C1DT 值匹配时翻转 C1ORAW .....	246
图 14-114 向上计数下 PWM 模式 A .....	247
图 14-115 中央双向对齐计数下 PWM 模式 .....	247
图 14-116 单周期模式 .....	247
图 14-117 EXT 清除 CxORAW(PWM 模式 A) .....	248
图 14-118 带死区插入的互补输出 .....	249
图 14-119 TMR 输出控制 .....	250
图 14-120 TMR 刹车功能的例子 .....	250
图 14-121 复位模式例子 .....	251
图 14-122 挂起模式下例子 .....	251
图 14-123 触发器模式例子 .....	251
图 15-1 窗口看门狗框图 .....	265
图 15-2 窗口看门狗时序图 .....	266
图 16-1 看门狗框图 .....	268
图 17-1 ERTC 框图 .....	271
图 18-1 ADC1 框图 .....	283
图 18-2 ADC 基础操作流程 .....	284
图 18-3 ADC 上电与校准 .....	285
图 18-4 序列模式 .....	286
图 18-5 抢占自动转换模式 .....	286
图 18-6 反复模式 .....	287
图 18-7 分割模式 .....	287
图 18-8 数据内容处理 .....	288
图 21-1 IRTMR 结构框图 .....	307

## 表目录

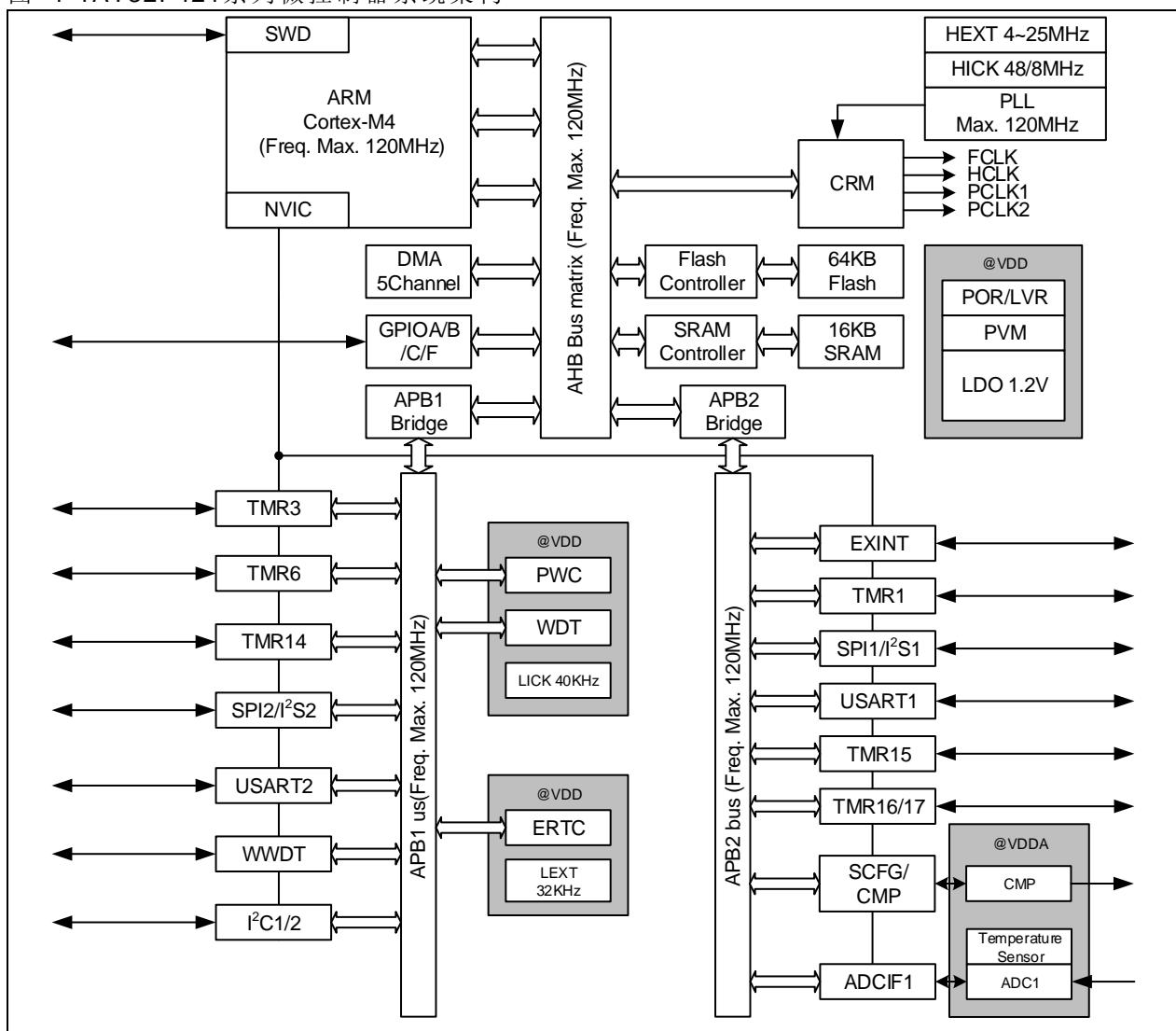
表 1-1 SRAM 区中的位带地址映射 .....	25
表 1-2 外设区中的位带地址映射 .....	26
表 1-3 AT32F421 产品的向量表 .....	26
表 1-4 寄存器描述缩写说明 .....	29
表 1-5 器件特征信息相关寄存器地址和复位值 .....	29
表 2-1 闪存存储结构 (64K) .....	31
表 2-2 闪存存储结构 (32K) .....	31
表 2-3 闪存存储结构 (16K) .....	32
表 2-4 各外设起始地址 .....	32
表 3-1 PWC 寄存器映像和复位值 .....	38
表 4-1 CRM 寄存器的映像和复位值 .....	44
表 5-1 闪存存储结构 (64K) .....	55
表 5-2 闪存存储组织 (32K) .....	55
表 5-3 闪存存储组织 (16K) .....	55
表 5-4 用户系统数据说明 .....	55
表 5-5 闪存访问权限 .....	64
表 5-6 闪存接口寄存器映像和复位值 .....	66
表 6-1 通过 GPIOA_MUX* 寄存器配置端口 A 的复用功能 .....	77
表 6-2 通过 GPIOB_MUX* 寄存器配置端口 B 的复用功能 .....	77
表 8-1 外部中断/事件控制器寄存器映像和复位值 .....	88
表 9-1 DMA 错误事件 .....	92
表 9-2 DMA 中断 .....	92
表 9-3 DMA 各通道的外设请求 .....	92
表 9-4 DMA 寄存器的映像和复位值 .....	93
表 10-1 CRC 计算单元寄存器映像 .....	100
表 11-1 I <sup>2</sup> C 寄存器地址映像和复位值 .....	114
表 12-1 设置波特率时的误差计算 .....	128
表 12-2 检测起始位和噪声的数据采样 .....	131
表 12-3 检测有效数据和噪声的数据采样 .....	131
表 12-4 最大允许偏差 .....	131
表 12-5 USART 中断请求 .....	132
表 12-6 USART 寄存器映像和复位值 .....	133
表 13-1 使用系统时钟得到精确的音频频率 .....	150
表 13-2 SPI 寄存器映像及其复位值 .....	154
表 14-1 TMR 功能对比 .....	159
表 14-2 TMR6 寄存器映像和复位值 .....	162
表 14-3 TMR3 内部触发连接 .....	167
表 14-4 计数方向与编码器信号的关系 .....	170
表 14-5 TMR3 寄存器映像和复位值 .....	178
表 14-6 标准 CxOUT 通道的输出控制位 .....	186
表 14-7 TMR14 寄存器映像和复位值 .....	192
表 14-8 标准 CxOUT 通道的输出控制位 .....	196
表 14-9 TMR15 内部触发连接 .....	200
表 14-10 TMR15 寄存器和复位值 .....	209
表 14-11 带刹车功能的互补输出通道 CxOUT 和 CxCOUT 的控制位 .....	215
表 14-12 TMR16、17 寄存器和复位值 .....	227
表 14-13 带刹车功能的互补输出通道 CxOUT 和 CxCOUT 的控制位 .....	232
表 14-14 TMR1 内部触发连接 .....	238
表 14-15 计数方向与编码器信号的关系 .....	242
表 14-16 TMR1 寄存器映像和复位值 .....	252
表 14-17 带刹车功能的互补输出通道 CxOUT 和 CxCOUT 的控制位 .....	261

表 15-1 PCLK1 频率为 72MHz 时，最大和最小看门狗超时时间.....	265
表 15-2 WWDT 寄存器映像和复位值.....	266
表 16-1 看门狗超时时间（LICK=40kHz）.....	269
表 16-2 WDT 寄存器映像和复位值 .....	269
表 17-1 ERTC 寄存器配置表.....	272
表 17-2 ERTC 唤醒低功耗模式.....	275
表 17-3 中断控制位 .....	275
表 17-4 ERTC 寄存器映像和复位值.....	275
表 18-1 ADC1 的触发来源.....	285
表 18-2 ADC 寄存器映像和复位值 .....	288
表 19-1 CMP 寄存器图和复位值 .....	302
表 22-1 DEBUG 寄存器地址和复位值.....	308

# 1 系统架构

AT32F421 系列微控制器内部集成了：32 位 ARM®Cortex®-M4 处理器，多个 16 位和 32 位的定时器，DMA 控制器，实时时钟 ERTC，SPI 通信接口，I2C 通信接口，USART 通信接口，比较器 CMP，12 位 ADC 和 PVM 模块等外设。大量的外设和存储器。Cortex®-M4 处理器支持增强的高效 DSP 指令集，包含扩展的单周期 16/32 位乘法累加器（MAC）、双 16 位 MAC 指令、优化的 8/16 位 SIMD 运算及饱和运算指令。系统详细架构见下图。

图 1-1 AT32F421 系列微控制器系统架构



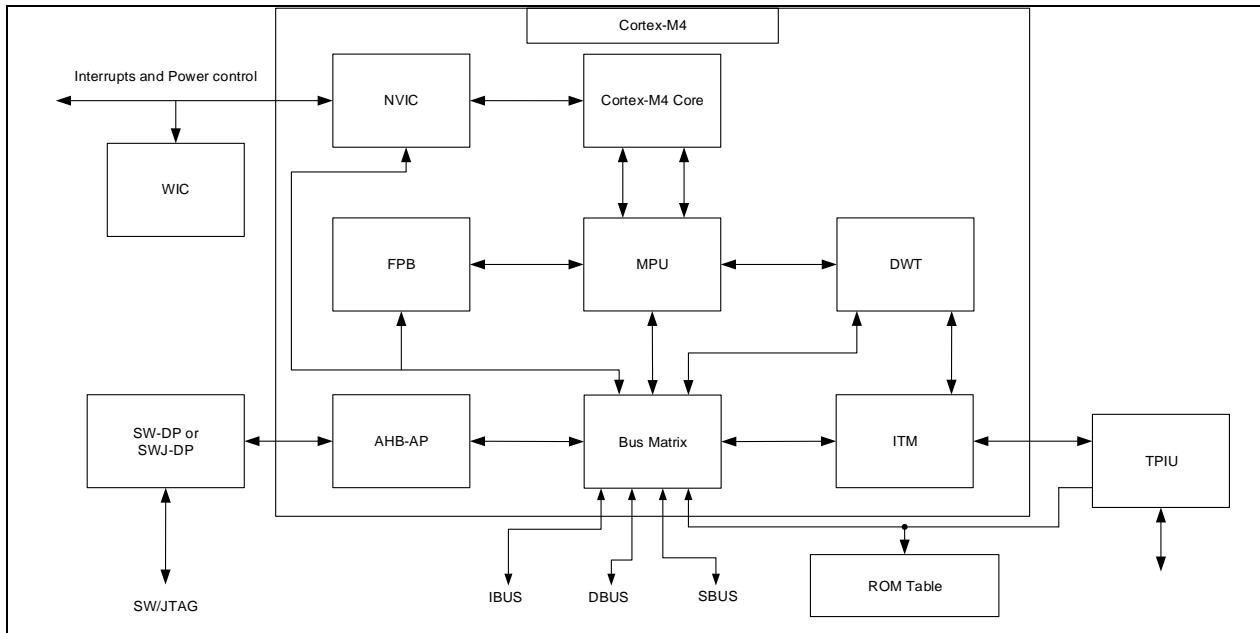
## 1.1 系统概述

### 1.1.1 ARM Cortex®-M4 处理器

Cortex®-M4 处理器是一款低功耗处理器，具有低门数，低中断延迟和低成本调试的特点。支持包括 DSP 指令集，特别适合用于深度嵌入式应用程序需要快速中断响应功能。Cortex®-M4 处理器是基于 ARMv7-M 架构，既支持 Thumb 指令集也支持 DSP 指令集。

下图为 Cortex®-M4 处理器的内部框图，请参阅《ARM®Cortex-M4 技术参考手册》了解关于 Cortex®-M4 更详尽信息。

图 1-2 Cortex®-M4 内部框图



## 1.1.2 位带

利用位带操作，可以使用普通的加载/存储操作来对单一比特进行读写访问。在 Cortex®-M4 中提供了两个位带区：SRAM 最低 1M 字节空间和外设区间的最低 1M 字节空间。这两个区中的地址除了可以像普通存储器一样访问外，还可以通过它们各自的位带别名区来快捷访问这两个区中任意地址的任意比特位，位带别名区将位带区每个比特膨胀成一个 32 位的字。当你访问位带别名区的一个地址时，等同于直接访问位带区的一个比特位。

图 1-3 位带区与位带别名区的膨胀关系图 A

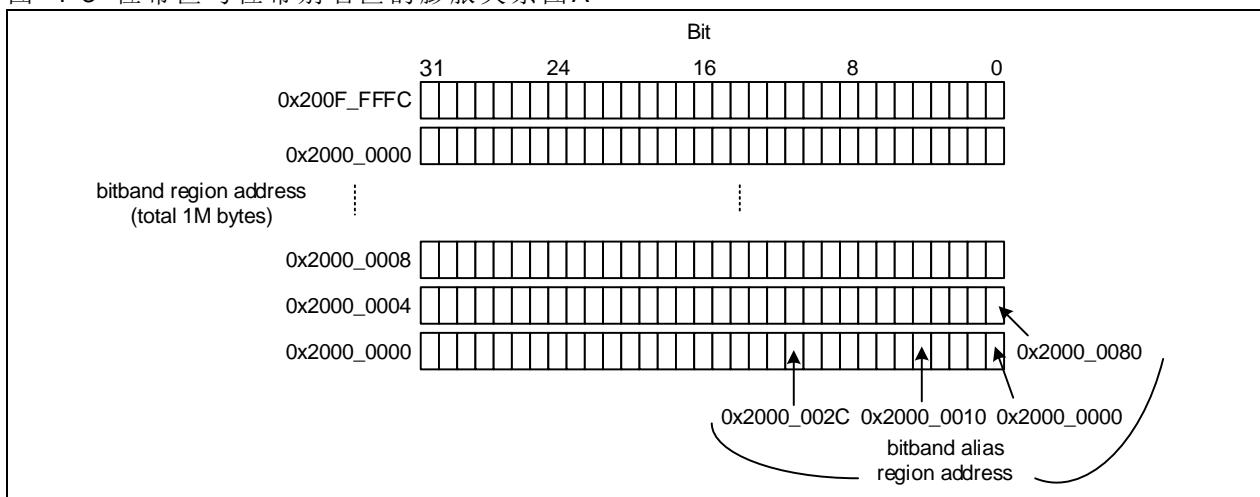
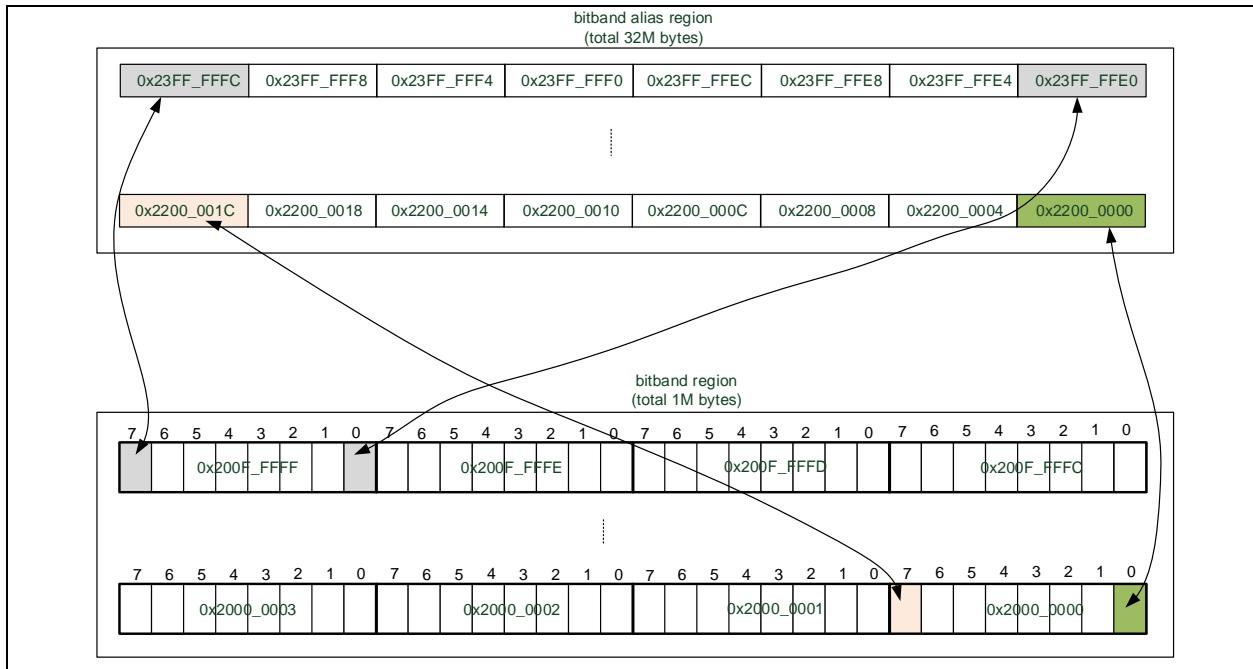


图 1-4 位带区与位带别名区的膨胀关系图B



位带区：支持位带操作的地址区

位带别名区：对别名区地址的访问最终作用到位带区的访问上

在位带区中，每个比特都映射到别名地址区的一个字（这是只有 LSB 有效的字）。当一个位带别名区地址被访问时，会先把该地址转换成位带区地址。对于读操作，读取位带区地址中的一个字，再把需要的位右移到 LSB，并把 LSB 返回。对于写操作，把需要写的位左移到对应的位序号处，然后执行一个比特级的“读-改-写”过程。

支持位带操作的两个内存区的地址范围为：

SRAM 区中的最低 1M 字节：0x2000\_0000~0x200F\_FFFF

外设区间的最低 1M 字节：0x4000\_0000~0x400F\_FFFF

对于 SRAM 位带区的某个比特，如果所在字节地址为 A，位序号为 n(0<=n<=7)，则该比特在别名区的地址为：

$\text{AliasAddr} = 0x2200\_0000 + (A - 0x2000\_0000) * 32 + n * 4$

对于外设区间位带区的某个比特，如果所在字节地址为 A，位序号为 n(0<=n<=7)，则该比特在别名区的地址为：

$\text{AliasAddr} = 0x4200\_0000 + (A - 0x4000\_0000) * 32 + n * 4$

对于 SRAM 区中，位带区与位带别名区的映射如下表所示：

表 1-1 SRAM区中的位带地址映射

位带区	等效别名区地址
0x2000_0000.0	0x2200_0000.0
0x2000_0000.1	0x2200_0004.0
0x2000_0000.2	0x2200_0008.0
...	...
0x2000_0000.31	0x2200_007C.0
0x2000_0004.0	0x2200_0080.0
0x2000_0004.1	0x2200_0084.0
0x2000_0004.2	0x2200_0088.0
...	...
0x200F_FFFC.31	0x23FF_FFFC.0

对于外设区中，位带区与位带别名区的映射如下表所示：

表 1-2 外设区中的位带地址映射

位带区	等效别名区地址
0x4000_0000.0	0x4200_0000.0
0x4000_0000.1	0x4200_0004.0
0x4000_0000.2	0x4200_0008.0
...	...
0x4000_0000.31	0x4200_007C.0
0x4000_0004.0	0x4200_0080.0
0x4000_0004.1	0x4200_0084.0
0x4000_0004.2	0x4200_0088.0
...	...
0x400F_FFFC.31	0x43FF_FFFC.0

位带操作对串行接口提供很大的方便，还能简化跳转的判断。当跳转依据是某个位时，以前必须这样做：

- 读取整个寄存器
- 屏蔽不需要的位
- 比较并跳转

现在只需要：

- 从位带别名区读取该位的状态
- 比较并跳转

使代码更简洁，这只是位带操作优越性的初步体现，位带操作还有一个重要的好处是在多任务以及多任务环境中，将以前的读-改-写需要的三条指令，做成了一个硬件级别支持的原子操作，消除了以前读-改-写可能被中断，导致出现紊乱的情况。

### 1.1.3 中断和异常向量

下面列出了 AT32F421 产品的向量表。

表 1-3 AT32F421 产品的向量表

位置	优先级 类型	名称	说明	地址
-	-	-	保留	0x0000_0000
-3	固定	Reset	复位	0x0000_0004
-2	固定	NMI	不可屏蔽中断 CRM 时钟失效检测 (CFD) 联接到 NMI 向量	0x0000_0008
-1	固定	硬件失效 (HardFault)	所有类型的失效	0x0000_000C
0	可设置	存储管理 (MemoryManage)	存储器管理	0x0000_0010
1	可设置	总线错误 (BusFault)	预取指失败，存储器访问失败	0x0000_0014
2	可设置	错误应用 (UsageFault)	未定义的指令或非法状态	0x0000_0018
-	-	-	保留	0x0000_001C ~0x0000_002B
3	可设置	SVCall	通过 SWI 指令的系统服务调用	0x0000_002C
4	可设置	调试监控 (DebugLENonitor)	调试监控器	0x0000_0030

-	-	-	保留	0x0000_0034	
5	可设置	PendSV	可挂起的系统服务	0x0000_0038	
6	可设置	SysTick	系统嘀嗒定时器	0x0000_003C	
0	7	可设置	WWDT	窗口定时器中断	0x0000_0040
1	8	可设置	PVM	连到 EXINT 线 16 的电源电压检测 (PVM) 中断	0x0000_0044
2	9	可设置	ERTC	连接到 EXINT 线 17, 19 的 ERTC 中断	0x0000_0048
3	10	可设置	FLASH	闪存全局中断	0x0000_004C
4	11	可设置	CRM	时钟和复位控制 (CRM) 中断	0x0000_0050
5	12	可设置	EXINT1_0	EXINT 线 1_0 中断	0x0000_0054
6	13	可设置	EXINT3_2	EXINT 线 3_2 中断	0x0000_0058
7	14	可设置	EXINT15_4	EXINT 线 15_4 中断	0x0000_005C
8	15	可设置	-	保留	0x0000_0060
9	16	可设置	DMA 通道 1	DMA 通道 1 全局中断	0x0000_0064
10	17	可设置	DMA 通道 3_2	DMA 通道 3_2 全局中断	0x0000_0068
11	18	可设置	DMA 通道 5_4	DMA 通道 5_4 全局中断	0x0000_006C
12	19	可设置	ADC_CMP	ADC 和 CMP 全局中断	0x0000_0070
13	20	可设置	TMR1_BRK TMR1_UP TMR1_TRG TMR1_COM	TMR1 中断	0x0000_0074
14	21	可设置	TMR1_CH	TMR1 捕获比较中断	0x0000_0078
15	22	可设置	-	保留	0x0000_007C
16	23	可设置	TMR3	TMR3 全局中断	0x0000_0080
17	24	可设置	TMR6	TMR6 全局中断	0x0000_0084
18	25	可设置	-	保留	0x0000_0088
19	26	可设置	TMR14	TMR14 全局中断	0x0000_008C
20	27	可设置	TMR15	TMR15 全局中断	0x0000_0090
21	28	可设置	TMR16	TMR16 全局中断	0x0000_0094
22	29	可设置	TMR17	TMR17 全局中断	0x0000_0098
23	30	可设置	I2C1_EVT	I <sup>2</sup> C1 事件中断	0x0000_009C
24	31	可设置	I2C2_EVT	I <sup>2</sup> C2 事件中断	0x0000_00A0
25	32	可设置	SPI1	SPI1 全局中断	0x0000_00A4
26	33	可设置	SPI2	SPI2 全局中断	0x0000_00A8
27	34	可设置	USART1	USART1 全局中断	0x0000_00AC
28	35	可设置	USART2	USART2 全局中断	0x0000_00B0
29	36	可设置	-	保留	0x0000_00B4
30	37	可设置	-	保留	0x0000_00B8

31	38	可设置	-	保留	0x0000_00BC
32	39	可设置	I2C1_ERR	I <sup>2</sup> C1 错误中断	0x0000_00C0
33	40	可设置	-	保留	0x0000_00C4
34	41	可设置	I2C2_ERR	I <sup>2</sup> C2 错误中断	0x0000_00C8

### 1.1.4 系统嘀嗒定时器 (SysTick)

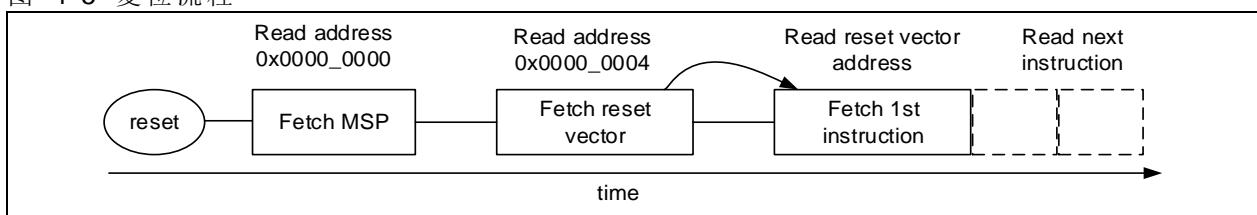
系统嘀嗒定时器是一个 24 位递减计数器，递减至零可自动重载计数初值。可产生周期性异常，用作嵌入式操作系统的多任务调度计数器，或对于无嵌入式操作系统，可用于调用需周期性执行的任务。系统嘀嗒定时器校准值固定值 9000，当系统嘀嗒时钟设定为 9MHz，产生 1ms 时间基准。

### 1.1.5 复位流程

系统复位后以及处理器开始执行程序前，处理器会从 CODE 存储器中读出前两个字。

- 从地址 0x0000\_0000 处取出主栈指针 (MSP) 的初始值。
- 从地址 0x0000\_0004 处取出程序计数器 (PC) 的初始值，这个值是复位向量，LSB 必须是 1。然后从这个值所对应的地址处取指。

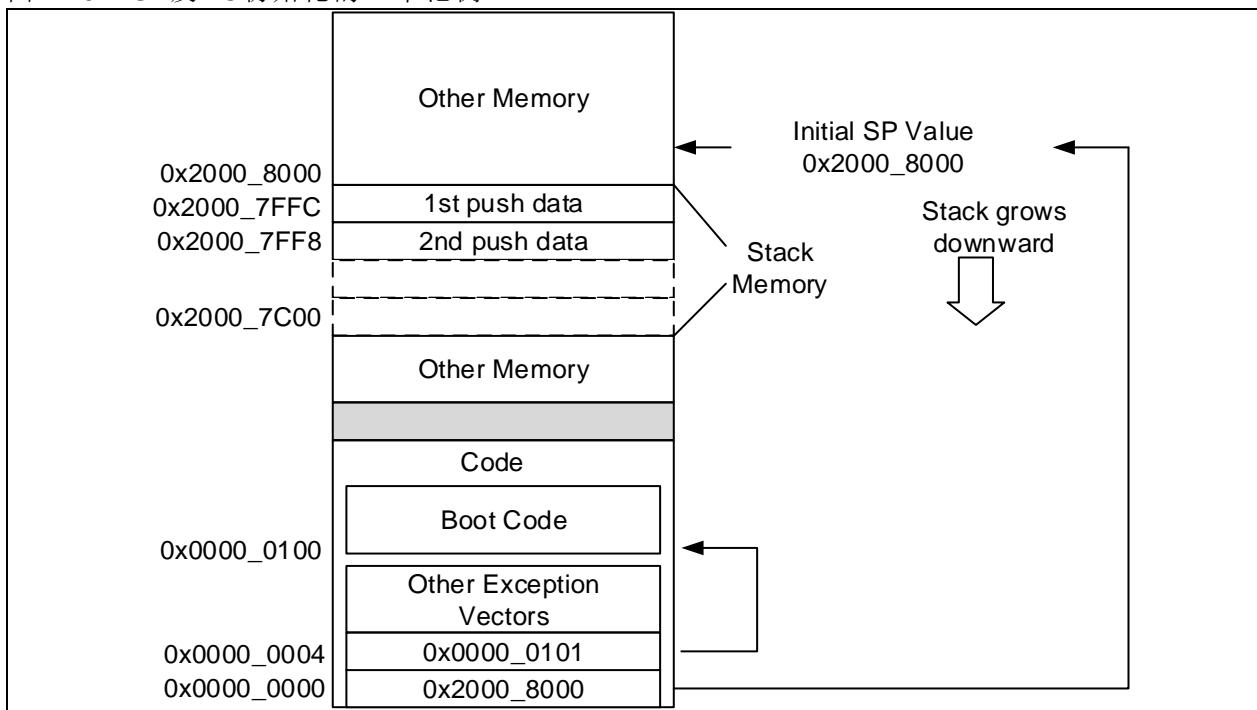
图 1-5 复位流程



Cortex®-M4 使用的是向下生长的满栈，所以 MSP 的初始值必须是堆栈内存的末地址加 1。举例来说，堆栈区域设定在 0x2000\_7C00~0x2000\_7FFF 之间，那么 MSP 的初始值必须是 0x2000\_8000。

向量表跟随在 MSP 的初始值之后。Cortex®-M4 是在 Thumb 态下执行，所以向量表中的每个数值都必须将 LSB 置 1，所以，下图中使用 0x0000\_0101 来表示地址 0x0000\_0100。当 0x0000\_0100 处的指令得到执行后，就正式开始程序的执行。在此之前初始化 MSP 是必须的，因为可能第一条指令还没执行就会被 NMI 或是其他 fault 打断。MSP 初始化好后就可以为它们的服务程序准备好堆栈空间。

图 1-6 MSP 及 PC 初始化的一个范例



在 AT32F421 中，可以将主闪存存储器、启动程序存储器或片上 SRAM 这三块存储器重映射到 0x0000\_0000~0x07FF\_FFFF 的 CODE 区，nBOOT1 对应用户系统数据区 (USD) 里面系统配置字节

(SSB) 位 nBOOT1 的值, 由 nBOOT1 和 BOOT0 管脚来设定 CODE 从哪块存储器启动:  
当{nBOOT1, BOOT0}=00/10 时, CODE 从主闪存存储器启动。  
当{nBOOT1, BOOT0}=11 时, CODE 从启动程序存储器启动。  
当{nBOOT1, BOOT0}=01 时, CODE 从片上 SRAM 启动。  
系统复位后或从待机模式退出时, nBOOT1 和 BOOT0 管脚值都会被重新锁存。  
当加载为片上 SRAM 启动模式后, BOOT 状态会被锁定, 此时系统复位无法加载新的启动模式, 必须上电复位后才能重新加载新的启动模式。  
启动程序存储器中包含内嵌的 Bootloader 程序, 可提供 flash 编程功能, 通过 USART1 或 USART2 接口对 flash 进行重新编程; 也可以提供通信协议栈等额外的固件, 可被软件开发人员通过 API 调用。

## 1.2 寄存器描述缩写说明

表 1-4 寄存器描述缩写说明

寄存器类型	说明
rw	可以读或写这些位
ro	只能读这些位
wo	只能写这些位; 如果读这些位, 则返回它们的复位值
rrc	可以读, 读取这些位时, 自动清除这些位
rw0c	可以读并写'0'清除这些位, 写'1'将不对该位产生影响
rw1c	可以读并写'1'清除这些位, 写'0'将不对该位产生影响
rw1s	可以读并写'1'设置这些位, 写'0'将不对该位产生影响
tog	可以读, 写'1'将翻转此位值, 写'0'将不对该位产生影响
rwt	可以读, 写任何值时, 将触发事件
resd	保留

## 1.3 器件特征信息

表 1-5 器件特征信息相关寄存器地址和复位值

寄存器简称	基地址	复位值
F_SIZE	0x1FFF F7E0	0xFFFF
UID[31:0]	0x1FFF F7E8	0xFFFF XXXX
UID[63:32]	0x1FFF F7EC	0xFFFF XXXX
UID[95:64]	0x1FFF F7F0	0xFFFF XXXX

### 1.3.1 闪存容量寄存器

闪存容量寄存器提供该芯片闪存容量信息, 用户可透过该寄存器取得闪存容量。

域	简称	复位值	类型	功能
位 15: 0	F_SIZE	0xFFFF	ro	闪存容量, 以 KByte 为单位 例如: 0x0040 = 64KByte

### 1.3.2 器件电子签名

器件电子签名包含产品容量信息和器件唯一 ID（96 位 UID），它位于闪存的信息区块中。96 位器件唯一 ID 对任何器件来说都是独一无二的，且用户不可更改。ID 可以用来作为下列用途：

- 序列号；例如 USB 字串序列
- 或者做为密钥的一部分

域	简称	复位值	类型	功能
位 31: 0	UID[31: 0]	0xXXXX XXXX ro		UID 的 bit31 到 bit0 信息
域	简称	复位值	类型	功能
位 31: 0	UID[63: 32]	0xXXXX XXXX ro		UID 的 bit63 到 bit32 信息
域	简称	复位值	类型	功能
位 31: 0	UID[95: 64]	0xXXXX XXXX ro		UID 的 bit95 到 bit64 信息

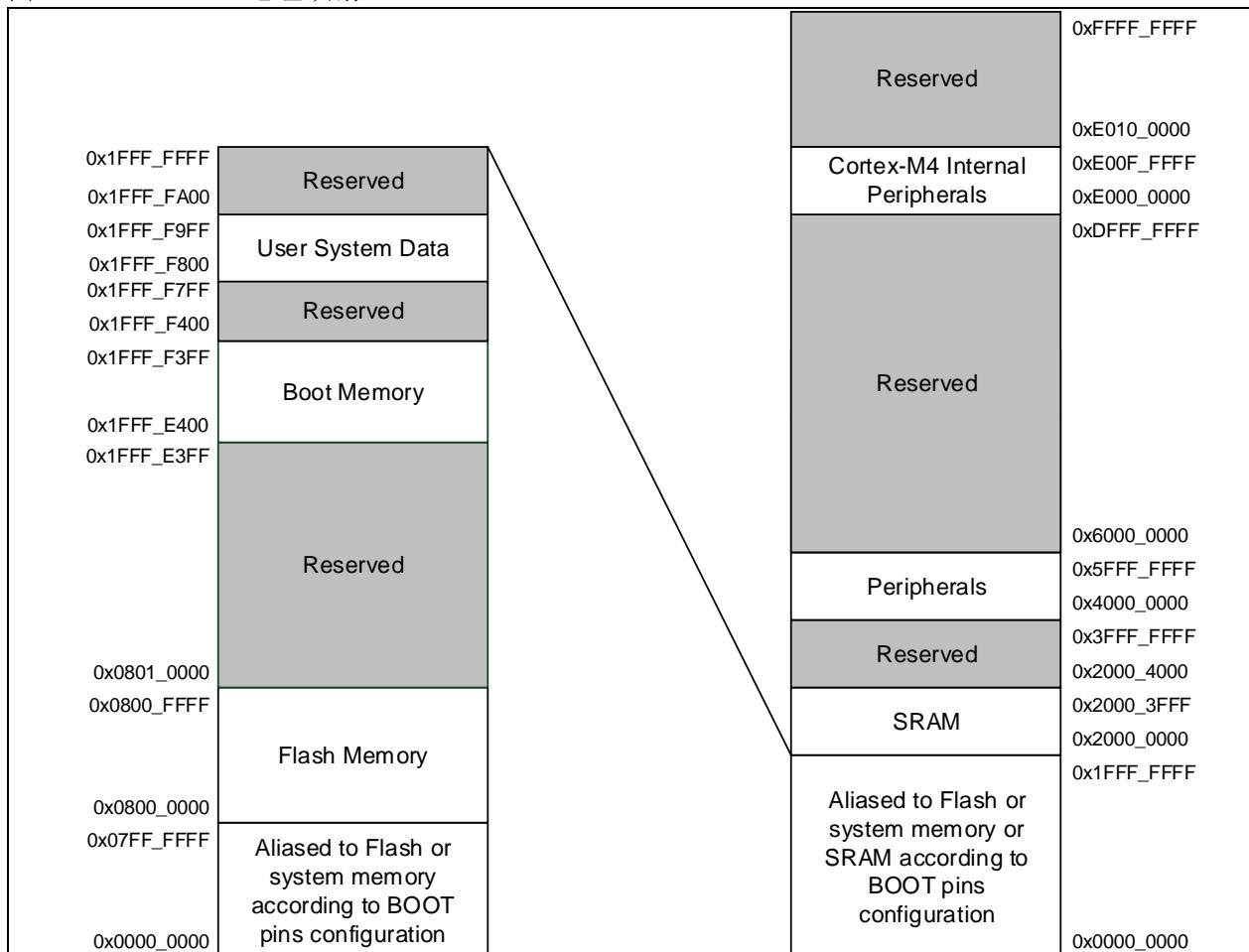
注：UID[95:88]为 Series ID，AT32F421 为 0x09。

## 2 存储器资源

### 2.1 内部存储器地址映射

芯片内部存储器包括程序存储器 flash, 数据存储器 SRAM, 外设寄存器和内核寄存器等。各区域地址映射如下图:

图 2-1 AT32F421 地址映射



### 2.2 Flash存储器

AT32F421 系列提供最大 64KB 的片上闪存，支持单周期最大 32 位读取操作。

闪存存储器由闪存控制器操作，有关闪存控制器的操作与寄存器配置信息请参考第 5 章节。

主存储器只有闪存容量为 64K 字节的片 1 闪存，包含 64 个扇区，每扇区大小为 1K 字节。

表 2-1 闪存存储结构 (64K)

结构	名称	地址范围
主存储器	扇区 0	0x0800_0000 – 0x0800_03FF
	扇区 1	0x0800_0400 – 0x0800_07FF
	扇区 2	0x0800_0800 – 0x0800_0BFF
	...	...
	扇区 63	0x0800_FC00 – 0x0800_FFFF
信息块	启动程序代码区 4K	0x1FFF_E400 – 0x1FFF_F3FF
	用户系统数据区 512B	0x1FFF_F800 – 0x1FFF_F9FF

主存储器只有闪存容量为 32K 字节的片 1 闪存，包含 32 个扇区，每扇区大小为 1K 字节。

表 2-2 闪存存储结构 (32K)

结构	名称	地址范围
----	----	------

主存储器	片 1 (Bank1) 32KB	扇区 0	0x0800 0000 – 0x0800 03FF
		扇区 1	0x0800 0400 – 0x0800 07FF
		扇区 2	0x0800 0800 – 0x0800 0BFF
		...	...
		扇区 31	0x0800 7C00 – 0x0800 7FFF
信息块	启动程序代码区 4K	0x1FFF E400 – 0x1FFF F3FF	
	用户系统数据区 512B	0x1FFF F800 – 0x1FFF F9FF	

表 2-3 闪存存储结构 (16K)

结构	名称	地址范围
主存储器	扇区 0	0x0800 0000 – 0x0800 03FF
	扇区 1	0x0800 0400 – 0x0800 07FF
	扇区 2	0x0800 0800 – 0x0800 0BFF
	...	...
	扇区 15	0x0800 3C00 – 0x0800 3FFF
信息块	启动程序代码区 4K	0x1FFF E400 – 0x1FFF F3FF
	用户系统数据区 512B	0x1FFF F800 – 0x1FFF F9FF

## 2.3 SRAM存储器

AT32F421 系列内置 16K 字节的片上 SRAM，起始地址为 0x2000\_0000。它可以以字节、半字（16 位）或全字（32 位）访问。

## 2.4 外设地址映射

表 2-4 各外设起始地址

总线	起始地址	外设
AHB	0x6000 0000 - 0xFFFF FFFF	保留
	0x5004 0000 - 0x5FFF FFFF	保留
	0x5000 0000 - 0x5003 FFFF	保留
	0x4800 1800 – 0x4FFF FFFF	保留
	0x4800 1400 - 0x4800 17FF	GPIOF
	0x4800 0C00 - 0x4800 13FF	保留
	0x4800 0800 - 0x4800 0BFF	GPIOC
	0x4800 0400 - 0x4800 07FF	GPIOB
	0x4800 0000 - 0x4800 03FF	GPIOA
	0x4002 8000 - 0x47FF FFFF	保留
	0x4002 3400 - 0x4002 7FFF	保留
	0x4002 3000 - 0x4002 33FF	CRC
	0x4002 2000 - 0x4002 23FF	闪存存储器接口 (FLASH)
	0x4002 1400 - 0x4002 1FFF	保留
	0x4002 1000 - 0x4002 13FF	时钟和复位控制 (CRM)
	0x4002 0800 - 0x4002 0FFF	保留
	0x4002 0400 - 0x4002 07FF	保留
	0x4002 0000 - 0x4002 03FF	DMA
APB2	0x4001 8400 - 0x4001 7FFF	保留
	0x4001 8000 - 0x4001 83FF	保留
	0x4001 7C00 - 0x4001 7FFF	保留

总线	起始地址	外设
APB1	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 6400 - 0x4001 67FF	保留
	0x4001 6000 - 0x4001 63FF	保留
	0x4001 5C00 - 0x4001 5FFF	保留
	0x4001 5800 - 0x4001 5BFF	保留
	0x4001 5400 - 0x4001 57FF	保留
	0x4001 5000 - 0x4001 53FF	保留
	0x4001 4C00 - 0x4001 4FFF	保留
	0x4001 4800 - 0x4001 4BFF	TMR17 定时器
	0x4001 4400 - 0x4001 47FF	TMR16 定时器
	0x4001 4000 - 0x4001 43FF	TMR15 定时器
	0x4001 3C00 - 0x4001 4BFF	保留
	0x4001 3800 - 0x4001 3BFF	USART1
	0x4001 3400 - 0x4001 37FF	保留
	0x4001 3000 - 0x4001 33FF	SPI1/I <sup>2</sup> S1
	0x4001 2C00 - 0x4001 2FFF	TMR1 定时器
	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 0BFF	保留
	0x4001 0400 - 0x4001 07FF	EXINT
	0x4001 0000 - 0x4001 03FF	SCFG/CMP
APB2	0x4000 8000 - 0x4000 FFFF	保留
	0x4000 7C00 - 0x4000 7FFF	保留
	0x4000 7800 - 0x4000 7BFF	保留
	0x4000 7400 - 0x4000 77FF	保留
	0x4000 7000 - 0x4000 73FF	电源控制 (PWC)
	0x4000 6800 - 0x4000 6BFF	保留
	0x4000 6400 - 0x4000 67FF	保留
	0x4000 6000 - 0x4000 63FF	保留
	0x4000 5C00 - 0x4000 5FFF	保留
	0x4000 5800 - 0x4000 5BFF	保留

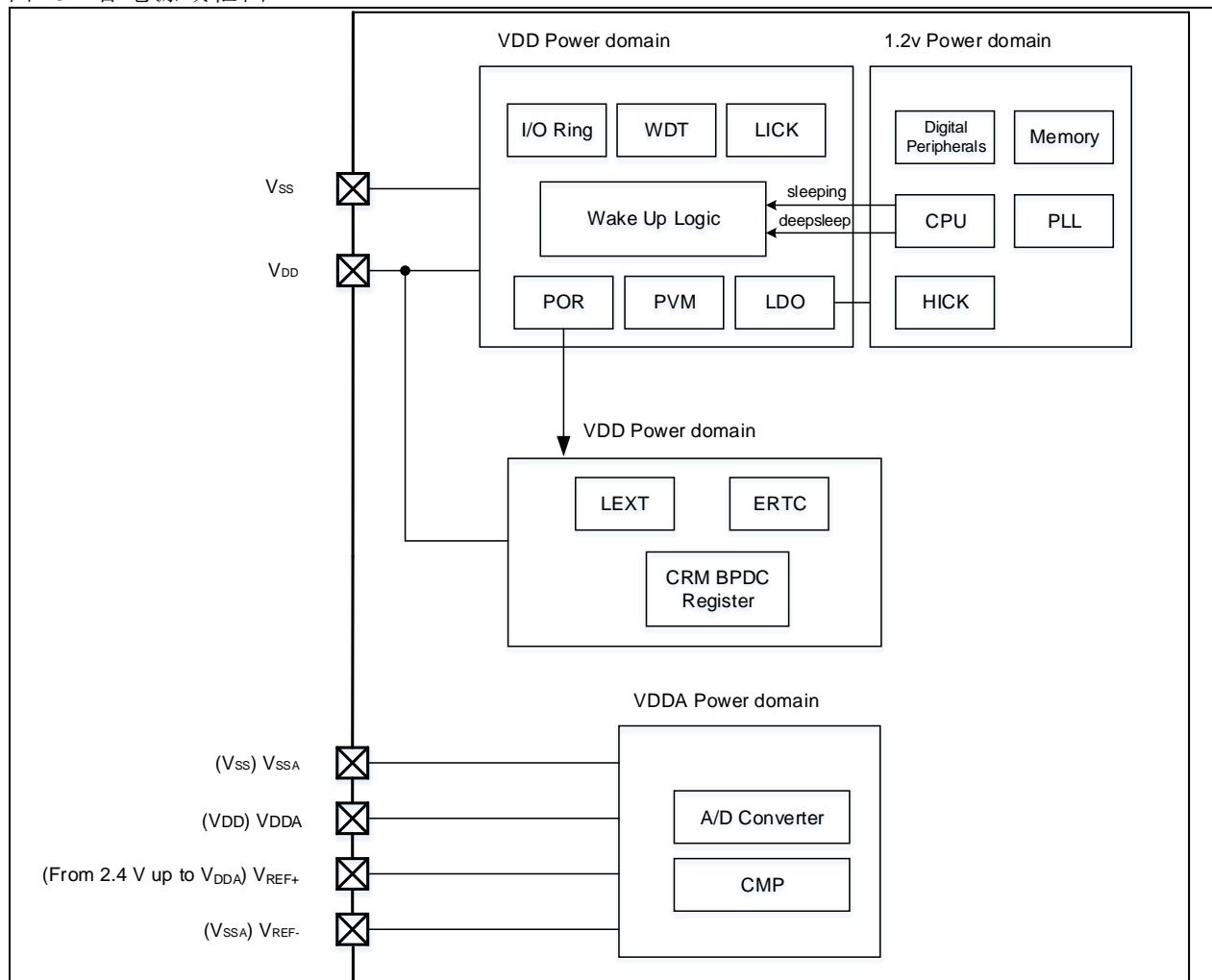
总线	起始地址	外设
	0x4000 5800 - 0x4000 5BFF	I <sup>2</sup> C2
	0x4000 5400 - 0x4000 57FF	I <sup>2</sup> C1
	0x4000 5000 - 0x4000 53FF	保留
	0x4000 4C00 - 0x4000 4FFF	保留
	0x4000 4800 - 0x4000 4BFF	保留
	0x4000 4400 - 0x4000 47FF	USART2
	0x4000 4000 - 0x4000 43FF	保留
	0x4000 3C00 - 0x4000 3FFF	保留
	0x4000 3800 - 0x4000 3BFF	SPI2/I <sup>2</sup> S2
	0x4000 3400 - 0x4000 37FF	保留
	0x4000 3000 - 0x4000 33FF	看门狗 (WDT)
	0x4000 2C00 - 0x4000 2FFF	窗口看门狗 (WWDT)
	0x4000 2800 - 0x4000 2BFF	ERTC
	0x4000 2400 - 0x4000 27FF	保留
	0x4000 2000 - 0x4000 23FF	TMR14 定时器
	0x4000 1C00 - 0x4000 1FFF	保留
	0x4000 1800 - 0x4000 1BFF	保留
	0x4000 1400 - 0x4000 17FF	保留
	0x4000 1000 - 0x4000 13FF	TMR6 定时器
	0x4000 0C00 - 0x4000 0FFF	保留
	0x4000 0800 - 0x4000 0BFF	保留
	0x4000 0400 - 0x4000 07FF	TMR3 定时器
	0x4000 0000 - 0x4000 03FF	保留

## 3 电源控制 (PWC)

### 3.1 简介

AT32F421 系列设备工作电压范围为 2.6V 至 3.6V，正常工作温度范围为 -40~+105°C。AT32F421 系列设备为了降低功耗，使用户可以在 CPU 运行时间要求、速度和功耗进行折中取舍，提供了三种省电模式——睡眠模式，深度睡眠模式和待机模式。AT32F421 系列设备有两个电源域——VDD/VDDA 域，1.2V 域。其中 VDD/VDDA 域由电源直接供电，1.2V 域由 VDD/VDDA 域中嵌入的 LDO 供电。

图 3-1 各电源域框图



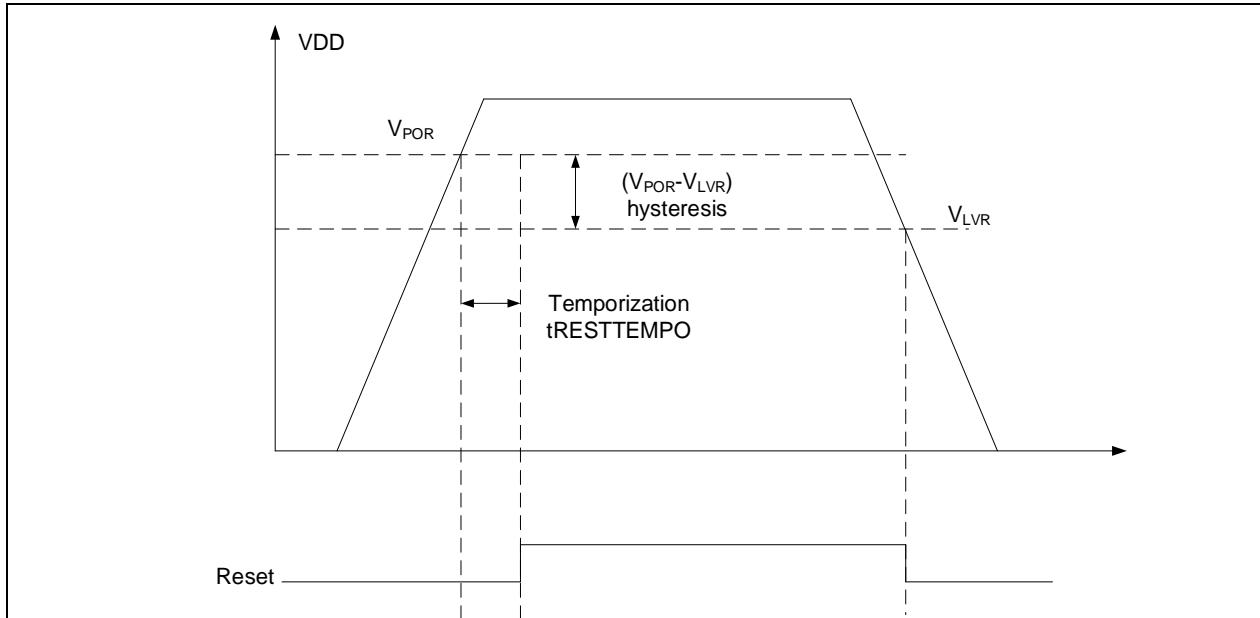
### 3.2 主要特点

- 具备两个电源域：VDD/VDDA 域和 1.2V 内核域。
- 支持三种省电模式：睡眠模式、深度睡眠模式和待机模式。
- 内建电压调节器，提供 1.2V 给内核域。
- 提供电压监测器，能在电压低于或高于阈值时发出中断或事件。
- VDD/VDDA 采用独立的数字和模拟地，用于隔离电源噪声。

### 3.3 上电下电复位

VDD/VDDA 域内置一个 POR 模拟模块用于产生电源复位，当 VDD 由 0V 上升至工作电压过程中，电源复位信号在  $V_{POR}$  时刻被上电释放。当 VDD 由工作电压下降至 0V 过程中，电源复位信号在  $V_{LVR}$  时刻被下电复位。上电复位过程，复位信号的释放相较于 VDD 升压过程存在一定的时间延迟，同时上电下电复位具有一定迟滞。

图 3-2 上电/下电复位波形图

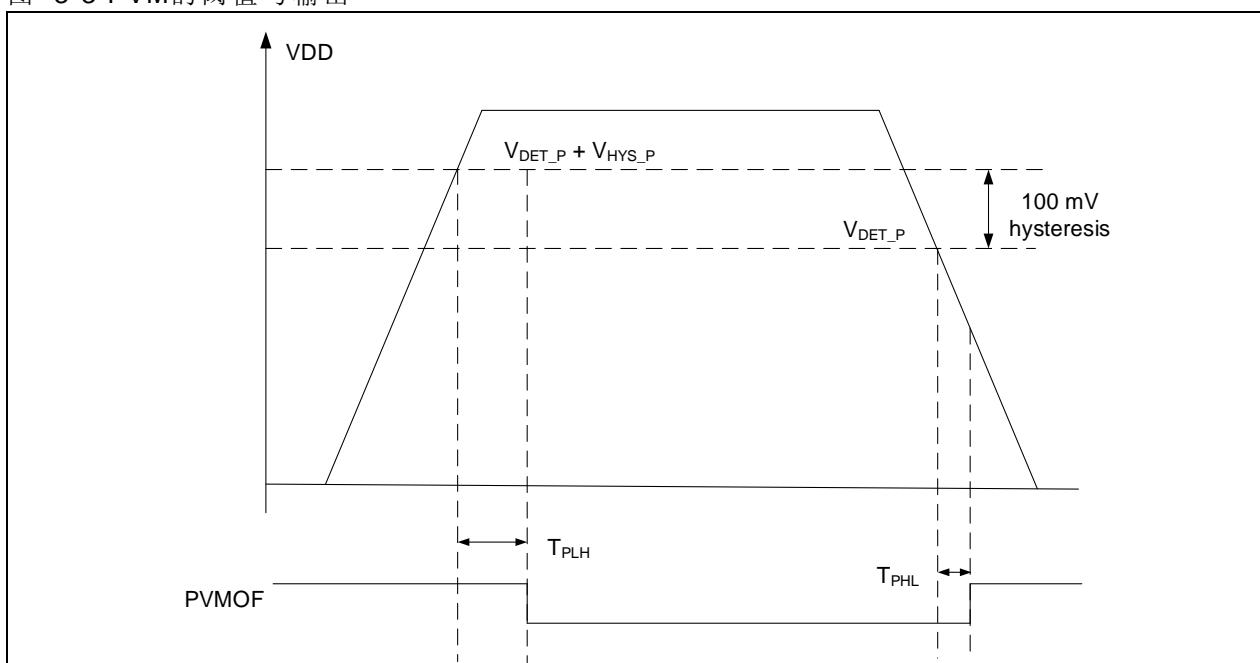


### 3.4 电压检测器 (PVM)

电压检测器 PVM 主要用来监控供电电源的跳变，可通过电源控制寄存器 (PWC\_CTRL) 中的 PVMEN 位开启电压检测功能，并通过 PVMSEL[2: 0]来选择监控阈值。

电压监测器开启后，电源控制及状态寄存器 (PWC\_CTRLSTS) 中的 PVMOF 位会指示 VDD 与设定阈值比较的结果，迟滞电压 VHYS\_P 为 100mV。当 VDD 越过 PVM 阈值边界时，产生的 PVMOF 位电平变化可以通过外部中断第 16 号线产生 PVM 中断。

图 3-3 PVM 的阈值与输出



### 3.5 电源域划分

#### 1.2V 域

1.2V 内核域包括 CPU 内核、存储器 SRAM、内嵌数字外设以及时钟锁相环 PLL，由 LDO (电压调节器) 供电。

#### VDD/VDDA 域

VDD/VDDA 域包括 VDD 域和 VDDA 域两部分。VDD 域包括 I/O 电路、省电模式唤醒电路、看门狗 WDT、

上电/掉电复位 (POR/LVR)、电压调节器 LDO、ERTC 电路、LEXT 振荡器以及所有的 PAD 电路等。VDDA 域包括 ADC (AD 转换器), 比较器 (CMP) 等。

一般来说, 为保证低电压时 ADC 的高精度, 数字电路由 VDD 供电, 模拟电路由 VDDA 独立供电, 外部参考电压 VREF+连接至 VDDA 引脚, VREF-连接至 VSSA 引脚。

## 3.6 省电模式

当 CPU 无需继续运行时, AT32F421 提供三种低功耗模式 (睡眠模式、深度睡眠模式、待机模式) 可以实现更低的功耗。用户可以在启动时间, 唤醒源, 电源消耗等方面进行折中。此外在运行模式下, 还可以通过降低系统时钟或关闭 APB 和 AHB 总线上未被使用的外设时钟来降低功耗。

### 睡眠模式 (Sleep Mode)

执行 WFI 或 WFE 指令可以进入睡眠状态。根据 Cortex®-M4 系统控制寄存器中的 SLEEPONEXIT 位有两种进入睡眠模式的机制:

#### SLEEP-NOW 模式

当 SLEEPDEEP=0, SLEEPONEXIT=0 时, 执行 WFI 或 WFE 指令, 此时可立即进入睡眠模式。

#### SLEEP-ON-EXIT 模式

当 SLEEPDEEP=0, SLEEPONEXIT=1 时, 执行 WFI 指令, 此时当系统从最低优先级的中断处理程序中退出时, 可立即进入睡眠模式。

在睡眠模式下, CPU 时钟关闭, 其他时钟均正常工作, 电压调节器正常工作, 所有的 I/O 管脚都保持它们在运行模式时的状态, 调节器 LDO 以正常功耗模式提供 1.2V 电源 (CPU 内核、内存和内嵌外设)。

1) 执行 WFI 指令进入睡眠模式时, 只要产生外设中断, 都能使系统退出睡眠模式。

2) 执行 WFE 指令进入睡眠模式时, 存在两种方式的唤醒事件, 使系统退出睡眠模式:

- 使能任一外设中断 (未在 NVIC 中使能) 且使能 SEVONPEND 位可以产生唤醒事件。

系统唤醒后, 需清除外设中断挂起位和 NVIC 通道挂起位。

- 配置内部 EXINT 线为事件模式来产生唤醒事件。

从执行 WFE 指令进入睡眠模式唤醒所需的时间最短, 因为没有时间损失在中断的进入或退出上。

### 深度睡眠模式 (Deepsleep Mode)

通过设置 Cortex®-M4 系统控制寄存器中的 SLEEPDEEP 位, 清除电源控制寄存器 (PWC\_CTRL) 中的 LPSEL 位, 再执行 WFI 或 WFE 指令即可进入深度睡眠模式。

还可以通过设置电源控制寄存器 (PWC\_CTRL) 中 VRSEL 位选择深度睡眠模式下电压调节器的工作状态。当 VRSEL=0, 电压调节器正常工作, 当 VRSEL=1, 电压调节器处于低功耗模式。

另外, 深度睡眠模式下, 在设置 VRSEL=1, 控制电压调节器处于低功耗模式的基础上, 可以通过设置内部电压调节器额外低功耗模式使能位 (VREXLPE), 进一步降低深度睡眠模式下整个系统的功耗。

在深度睡眠模式下, 所有 1.2V 时钟关闭, HICK 和 HEXT 振荡器都被关闭, 电压调节器以正常工作或低功耗工作状态给 1.2V 域供电, 所有 I/O 管脚都保持它们在运行模式时的状态, SRAM 和寄存器内容保持。

- 1) 执行 WFI 指令进入深度睡眠模式, 任一外部中断线在中断模式下产生的中断, 即可使系统退出深度睡眠模式。
- 2) 如果执行 WFE 指令进入深度睡眠模式, 任一外部中断线在事件模式下产生的事件, 即可使系统退出深度睡眠模式。

系统从深度睡眠模式退出时, HICK RC 振荡器开启并在稳定后被选为系统时钟。当电压调节器处于低功耗模式时, 退出深度睡眠模式时, 需要额外等待电压调节器稳定, 从而会增加一段额外的唤醒时间。

### 待机模式 (Standby Mode)

待机模式可最大限度的降低系统功耗, 在该模式下, 电压调节器关闭, 只有 VDD/VDDA 域维持供电, 其他的 1.2V 供电区域, PLL、HICK 和 HEXT 振荡器都被断电。寄存器和 SRAM 中的内容也会丢失。

通过设置 Cortex®-M4 系统控制寄存器中的 SLEEPDEEP 位, 设置电源控制寄存器(PWC\_CTRL)中 LPSEL 位, 并清除电源控制及状态寄存器 (PWC\_CTRLSTS) 中的 SWEF 位的情况下, 执行 WFI 或 WFE 指令即可进入待机模式。

在待机模式下, 除了复位管脚、被设置为防侵入或校准输出时的 TAMPER 引脚和唤醒引脚之外, 所有的 I/O 引脚处于高阻态。

当发生 WKUPx 引脚的上升沿、ERTC 闹钟事件的上升沿、ERTC 入侵事件、ERTC 时间戳、NRST 引脚

上外部复位、WDT 复位时，微控制器将退出待机模式。

### 调试配置

默认情况下，在进行调试时，微处理器一旦进入深度睡眠或待机模式，会因为 Cortex®-M4 的内核失去了时钟而失去调试连接。只需通过设置 DEBUG 控制寄存器（DEBUG\_CTRL）中的某些配置位，就可以在低功耗模式下继续调试软件。

## 3.7 PWC 寄存器

必须用字（32 位）的方式操作这些外设寄存器。

表 3-1 PWC 寄存器映像和复位值

寄存器简称	基址偏移量	复位值
PWC_CTRL	0x00	0x0000 0000
PWC_CTRLSTS	0x04	0x0000 0000
PWC_CTRL2	0x20	0x0000 00XX

### 3.7.1 电源控制寄存器（PWC\_CTRL）

域	简称	复位值	类型	功能
位 31: 9	保留	0x000000	resd	保持默认值。 电池供电区域的写入使能（Battery powered domain write enable） 0: 关闭； 1: 开启。 注： 复位后，电池供电区域禁止写入。要对电池供电区域进行写操作的话，需先设置这位为允许写入状态。
位 8	BPWEN	0x0	rw	电压监测临界值选择（Power voltage monitoring boundary select） 000: 未用，禁止配置； 001: 2.3V； 010: 2.4V； 011: 2.5V； 100: 2.6V； 101: 2.7V； 110: 2.8V； 111: 2.9V。
位 7: 5	PVMSEL	0x0	rw	电压监测使能（Power voltage monitoring enable） 0: 关闭； 1: 开启。
位 4	PVMEN	0x0	rw	清除 SEF 标志（Clear SEF flag） 0: 无效； 1: 清除 SEF 标志。 注：该位在清除 SEF 后由硬件将其清零，且任何时刻读取该位返回值均是零。
位 3	CLSEF	0x0	wo	清除 SWEF 标志（Clear SWEF flag） 0: 无效； 1: 清除 SWEF 标志。 注：该位在清除 SWEF 后由硬件将其清零，且任何时刻读取该位返回值均是零。
位 2	CLSWEF	0x0	wo	实际 SWEF 标志的清除大约需要 2 个系统时钟周期； 该位在清除 SWEF 后由硬件将其清零，且任何时刻读取该位返回值均是零。
位 1	LPSEL	0x0	rw	SLEEPDEEP 状态下的低功耗模式选择位（Low power mode select when Cortex®-M4 sleepdeep） 0: 进入 DEEPSLEEP 模式； 1: 进入待机模式
位 0	VRSEL	0x0	rw	DEEPSLEEP 模式下电压调节器状态选择（Voltage regulator state select when deepsleep mode）

0: 正常开启；  
1: 低功耗模式。

### 3.7.2 电源控制及状态寄存器 (PWC\_CTRLSTS)

与标准的 APB 读相比，读此寄存器需要额外的 APB 周期

域	简称	复位值	类型	功能
位 31: 15 保留		0x00000	resd	保持默认值。
位 14 SWPEN7		0x0	rw	待机唤醒管脚 7 使能 (Standby wake-up pin enable) 0: 关闭 (该管脚可用作通用 I/O)； 1: 开启 (该管脚被强置为输入下拉模式，且无法再用作通用 I/O)。 注：在系统复位时硬件将清除这一位。 在待机模式下，无论唤醒管脚是否使能，都被强置为输入下拉模式。
位 13 SWPEN6		0x0	rw	待机唤醒管脚 6 使能 (Standby wake-up pin enable) 0: 关闭 (该管脚可用作通用 I/O)； 1: 开启 (该管脚被强置为输入下拉模式，且无法再用作通用 I/O)。 注：在系统复位时硬件将清除这一位。 在待机模式下，无论唤醒管脚是否使能，都被强置为输入下拉模式。
位 12: 10 保留		0x0	resd	保持默认值。
位 9 SWPEN2		0x0	rw	待机唤醒管脚 2 使能 (Standby wake-up pin enable) 0: 关闭 (该管脚可用作通用 I/O)； 1: 开启 (该管脚被强置为输入下拉模式，且无法再用作通用 I/O)。 注：在系统复位时硬件将清除这一位。 在待机模式下，无论唤醒管脚是否使能，都被强置为输入下拉模式。
位 8 SWPEN1		0x0	rw	待机唤醒管脚 1 使能 (Standby wake-up pin enable) 0: 关闭 (该管脚可用作通用 I/O)； 1: 开启 (该管脚被强置为输入下拉模式，且无法再用作通用 I/O)。 注：在系统复位时硬件将清除这一位。 在待机模式下，无论唤醒管脚是否使能，都被强置为输入下拉模式。
位 7: 3 保留		0x00	resd	保持默认值。
位 2 PVMOF		0x0	ro	电源电压检测输出标志 (Power voltage monitoring output flag) 0: 电源电压高于临界值； 1: 电源电压低于临界值。 注：待机模式下电压监测停止工作。
位 1 SEF		0x0	ro	进入待机模式标志 (Standby mode entry flag) 0: 未进过待机模式； 1: 有进过待机模式。 注：该位被硬件置起（进入待机模式时），由 POR/LVR 或写 CLSEF 位将其清零。
位 0 SWEF		0x0	ro	待机唤醒事件标志 (Standby wake-up event flag) 0: 无唤醒事件产生； 1: 有唤醒事件产生。 注： 该位被硬件置起（产生唤醒事件时），由 POR/LVR 或写 CLSWEF 位将其清零。 唤醒事件将由以下几种情况产生： 在待机唤醒管脚上出现上升沿时，将产生唤醒事件； 出现 RTC 闹钟事件时，将产生唤醒事件； 待机唤醒管脚保持高电平期间使能该待机唤醒管脚，将产生唤醒事件。

### 3.7.3 电源控制寄存器2 (PWC\_CTRL2)

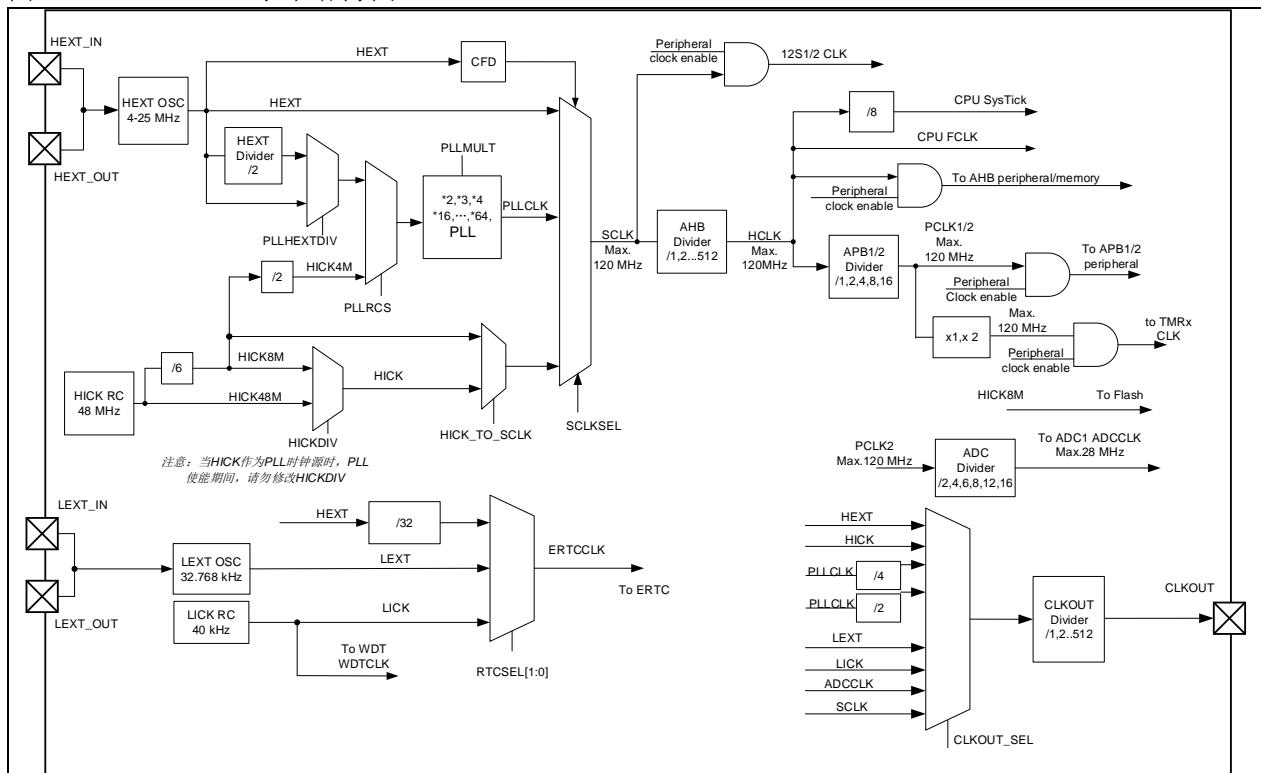
域	简称	复位值	类型	功能
位 31: 6	保留	0x0000000	resd	保持默认值。
位 5	VREXL PEN	0x0	rw	内部电压调节器额外低功耗模式使能位(Voltage regulator extra low power mode enable) 与 PWC_CTRL 的 LPSEL 和 VRSEL 位协同工作，在 VRSEL = 1 时，且芯片进入深度睡眠模式时才有效。 0: 内部电压调节器额外低功耗模式关闭。 1: 内部电压调节器额外低功耗模式开启。 注：如需启动额外低功耗模式，请先配置 LPSEL 和 VRSEL 位，再配置 VREXL PEN。
位 4: 0	保留	0xXX	resd	保持出厂设置值，请勿修改。

## 4 时钟和复位管理 (CRM)

### 4.1 时钟

AT32F421 的时钟源包含：HEXT 振荡器时钟，HICK 振荡器时钟，PLL 时钟，LEXT 振荡器时钟和 LICK 振荡器时钟。时钟结构如下：

图 4-1 AT32F421 时钟结构图



HEXT: High speed external crystal 高速外部晶振

LEXT: Low speed external crystal 低速外部晶振

HICK: High speed internal clock 高速内部时钟

LICK: Low speed internal clock 低速内部时钟

AHB、APB1 和 APB2 的频率都支持多种分频。AHB 域的最大频率是 120MHz，APB1 和 APB2 域的最大允许频率是 120MHz。

#### 4.1.1 时钟源

- HEXT 振荡器时钟

包括 HEXT 晶体/陶瓷谐振器和 HEXT 旁路时钟两个时钟源。

HEXT 晶体/陶瓷谐振器外接一颗频率范围为 4~25MHz HEXT 的晶体，可为系统提供高精度的时钟。HEXT 时钟直到时钟稳定后才会被释放出来。

HEXT 旁路时钟可以提供频率高达 25MHz 的外部时钟。外部时钟信号必须连到 HEXT\_IN 管脚，并且 HEXT\_OUT 管脚也一定要保持悬空。

- HICK 振荡器时钟

HICK 振荡器时钟由芯片内的高速 RC 振荡器提供。HICK 时钟的内部频率为 48MHz，频率精度较差，但启动时间比 HEXT 晶体振荡器短，每颗芯片的 HICK 时钟频率在出厂前已经被校准到 1% (25° C)，工厂校准值被装载到时钟控制寄存器的 HICKCAL[7: 0]位。考虑不同的电压或环境温度对 HICK 的 RC 振荡器的影响，用户可以通过时钟控制寄存器里的 HICKTRIM[5: 0]位来调整 HICK 频率。

HICK 时钟直到稳定后才会被释放出来。

- PLL 时钟

PLL 的输入时钟源可以选择 HICK 时钟或 HEXT 时钟，PLL 的输入时钟在 PLL 内部经过预分频器分频后送给 VCO 倍频，VCO 输出频率经过后分频器分频后输出。其中预分频后时钟需保证在 2M~16MHz 之

间，VCO 的工作频率需保证在 500MHz~1000MHz 之间。使用 PLL 前，一定要先配置 PLL 参数，否则，PLL 使能后，这些参数将无法改动。PLL 时钟直到稳定后才会被释放出来。

PLL 有两种配置使用方式，常规整数倍频配置模式和灵活配置模式：

### 1) 常规整数倍频配置模式（默认模式）

根据 PLL 时钟计算公式：

$$\text{PLL输出时钟} = \text{PLL输入时钟} \times \text{PLL倍频系数}$$

配置流程：

a) 清除 PLLCFGGEN (RCC\_PLL[31])

b) 设定当前 PLL 的输入时钟频率，具体参考 PLL\_FREF (RCC\_PLL[26:24])

c) 设定当前 PLL 的倍频系数，具体参考 PLLMUL (RCC\_CFG[30:29], RCC\_CFG[21:18])

d) 使能 PLL

e) 等待 PLL 稳定工作

### 2) 灵活配置模式

根据 PLL 时钟计算公式：

$$\text{PLL输出时钟} = \text{PLL输入时钟} \times \text{PLL倍频系数} / (\text{PLL预分频系数} \times \text{PLL后分频系数})$$

$$500\text{MHz} \leq \text{PLL 输入时钟} \times \text{PLL 倍频系数} / \text{PLL 预分频系数} \leq 1000\text{MHz}$$

$$2\text{MHz} \leq \text{PLL 输入时钟} / \text{PLL 预分频系数} \leq 16\text{MHz}$$

配置流程：

a) 计算出 PLL 倍频系数，PLL 预分频系数和 PLL 后分频系数后，对 RCC\_PLL 寄存器中的 PLL\_NS/PLL\_MS/PLL\_FR 对应位进行配置

b) 使能 PLLCFGGEN (RCC\_PLL[31])

c) 使能 PLL

d) 等待 PLL 稳定工作

注：灵活配置模式，PLL\_FREF 和 PLLMUL 寄存器不起作用。

注：灵活配置模式，可以实现非整数倍频，

例如：当 PLL 输入时钟为 12.288 MHz 时，可以配置 PLL 输出频率=  $12.288 \times 125 / (2 \times 8) = 96\text{MHz}$

例如：当 PLL 输入时钟为 5 MHz 时，可以配置 PLL 输出频率=  $5 \times 108 / (5 \times 1) = 108\text{MHz}$

### ● LEXT 振荡器时钟

LEXT 振荡器时钟包括 LEXT 晶体/陶瓷谐振器和 LEXT 旁路时钟两个时钟源。

#### LEXT 晶体/陶瓷谐振器

LEXT 晶体/陶瓷谐振器提供一个低功耗且精确的 32.768kHz 低速时钟源。LEXT 时钟直到稳定后，才会被释放出来。

#### LEXT 旁路时钟

在 LEXT 旁路模式下，可以提供最高频率达 32.768kHz 的外部时钟源。外部时钟信号必须连到 LEXT\_IN 管脚，并且 LEXT\_OUT 管脚也一定要保持悬空。

### ● LICK 振荡器时钟

LICK 振荡器时钟由芯片内的低速 RC 振荡器提供，作为一个频率在 30kHz 和 60kHz 之间的低功耗时钟源，它可以为看门狗和自动唤醒单元提供时钟，并能在深度睡眠和待机模式下保持运行。

LICK 时钟直到稳定后，才会被释放出来。

## 4.1.2 系统时钟

系统复位以后，系统时钟使用 HICK 时钟作为默认时钟。系统时钟可在 HICK 振荡器时钟、HEXT 振荡器时钟和 PLL 时钟之间进行灵活切换，只有当目标时钟源稳定后，系统时钟切换才会发生。当 HICK 振荡器时钟直接作为系统时钟或间接通过 PLL 作为系统时钟时，它将无法被停止。

## 4.1.3 外设时钟

大多数外设使用系统时钟 HCLK、PCLK1 或 PCLK2 时钟。个别外设还有专用时钟。

系统滴答定时器 (SysTick) 使用 HCLK 或 HCLK 的 8 分频作为时钟。

ADC 使用 APB2 时钟的 2、4、6、8、12、16 分频作为时钟。

定时器使用 APB1/2 作为时钟，特别地，当 APB 预分频系数是 1 时，定时器的时钟频率等于 APB1/2 的时钟频率；当 APB 预分频系数不为 1 时，定时器的时钟频率等于 APB1/2 时钟频率的 2 倍。

ERTC 的时钟源有：HEXT 振荡器 32 分频时钟，LEXT 振荡器时钟 LICK 振荡器时钟。ERTC 的时钟源一旦选择后就不可再更改，只有将电池供电域复位后才能重新配置 ERTC 时钟源。当 VDD 掉电时，ERTC 使用 LEXT 作为时钟的话，ERTC 可以继续工作，但 ERTC 使用 HEXT 或 LICK 作为时钟源时，由于 HEXT 和 LICK 均掉电，会导致 ERTC 状态不定。

看门狗使用 LICK 振荡器时钟作为时钟源。硬件选项或软件开启看门狗后，将强制打开 LICK 振荡器，LICK 振荡器稳定后，才给看门狗提供时钟。

#### 4.1.4 时钟失效检测

当 HEXT 时钟直接或间接作为系统时钟时，为防止 HEXT 时钟出现故障，特设计了时钟失效检测模块（CFD）。当 HEXT 时钟出现故障，CFD 侦测到失效后，将时钟失效事件送到 TMR1 的刹车输入端，并产生 CFD 中断，此 CFD 中断直接连到 CPU 的 NMI 中断，供软件完成营救操作。NMI 中断将一直重复执行，直到 CFD 中断挂起位被清除为止，所以在 NMI 的处理程序中必须清除 CFD 中断。当 HEXT 时钟出现故障时，将导致系统时钟切换到 HICK 时钟，同时关闭 CFD，关闭 HEXT 时钟，如果 HEXT 时钟通过 PLL 做为系统时钟时，也会关闭 PLL 模块。

#### 4.1.5 自动滑顺频率切换

当系统时钟源从其他时钟切换到 PLL 或是 AHB 预分频由大切换到小时，为了使系统稳定顺滑切换，特设计了自动顺滑频率切换功能，当系统频率操作目标大于 108MHz 时，建议开启自动顺滑频率切换功能。

当自动顺滑频率切换功能开启时，硬件会暂停 AHB 总线，直到整个自动顺滑频率切换才恢复。此期间 DMA 仍正常工作，中断事件会被记忆并待 AHB 总线恢复后 NVIC 即可处理。

#### 4.1.6 内部时钟输出

微控制器允许输出内部时钟信号到外部 CLKOUT 管脚。ADC CLK、SCLK、LICK、LEXT、HICK、HEXT、除 2 的 PLL 时钟以及除 4 的 PLL 时钟这 8 个时钟信号可输出到 CLKOUT。

#### 4.1.7 中断

微控制器为每个时钟源设计了一个稳定标志，当用户开启一个时钟源后，可查询对应的时钟源的稳定标志来判断时钟是否稳定。当用户开启对应时钟源的中断使能的话，将产生中断请求。

当 HEXT 时钟出现故障，CFD 侦测到失效后，将产生 CFD 中断，此中断直接连到 CPU 的 NMI 中断。

### 4.2 复位

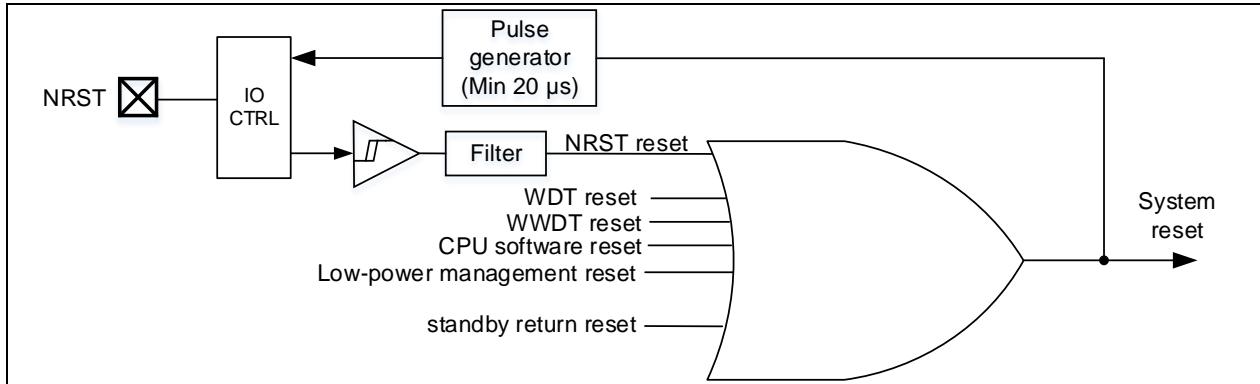
#### 4.2.1 系统复位

AT32F421 系统复位包括以下复位源：

- NRST 复位：外部 NRST 管脚复位
- WDT 复位：看门狗溢出复位
- WWDT 复位：窗口看门狗溢出复位
- CPU 软件复位：Cortex®-M4 软件复位
- 低功耗管理复位：将用户选择字节中的 nSTDBY\_RST 位清 0 并进入待机模式，将产生低功耗管理复位；将用户选择字节中的 nDEPSLP\_RST 位清 0 并进入深度睡眠模式，也将产生低功耗管理复位。
- 从待机模式中返回等事件产生复位。

NRST 复位，WDT 复位，WWDT 复位，软件复位和低功耗管理复位将复位所有寄存器至它们的复位状态，时钟控制器的控制/状态寄存器（CRM\_CTRLSTS）和电池供电域中的寄存器除外；上电复位、掉电复位或者从待机模式中返回等事件产生复位会复位所有寄存器至复位状态，电池供电寄存器除外。

图 4-2 系统复位电路



## 4.2.2 电池供电域复位

电池供电域复位包括以下复位源：

- 电池供电域软件复位：设置电池供电域控制寄存器（CRM\_BPDC）中的 BPDRST 位来产生复位。
  - 在 VDD 掉电的前提下，VDD 再上电将产生复位。
- 电池供电域软件复位只影响电池供电域。

## 4.3 CRM 寄存器

下表列出了 CRM 寄存器的映像和复位值。

可以用字节（8 位）、半字（16 位）或字（32 位）的方式操作这些外设寄存器。

表 4-1 CRM 寄存器的映像和复位值

寄存器简称	基址偏移量	复位值
CRM_CTRL	0x000	0x0000 XX83
CRM_CFG	0x004	0x0000 0000
CRM_CLKINT	0x008	0x0000 0000
CRM_APB2RST	0x00C	0x0000 0000
CRM_APB1RST	0x010	0x0000 0000
CRM_AHBEN	0x014	0x0000 0014
CRM_APB2EN	0x018	0x0000 0000
CRM_APB1EN	0x01C	0x0000 0000
CRM_BPDC	0x020	0x0000 0000
CRM_CTRLSTS	0x024	0x0C00 0000
CRM_AHBRST	0x028	0x0000 0000
CRM_PLL	0x2C	0x0000 1F10
CRM_MISC1	0x030	0x0010 0000
CRM_MISC2	0x054	0x0000 000D

### 4.3.1 时钟控制寄存器（CRM\_CTRL）

访问：无等待状态，字，半字和字节访问

域	简称	复位值	类型	功能
位 30: 26	保留	0x00	resd	请保持默认值。
位 25	PLLSTBL	0x0	ro	PLL 时钟稳定（PLL clock stable） 该位待 PLL 稳定后由硬件置起。 0: 未稳定； 1: 已稳定。
位 24	PLLEN	0x0	rw	PLL 使能（PLL enable）

					该位可由软件置起或清除，也可在进入待机或深度睡眠模式时，由硬件清除。当系统时钟为 PLL 时钟时，该位无法清除。 0: 关闭； 1: 开启。
位 23: 20	保留	0x0	resd	保持默认值。	
位 19	CFDEN	0x0	rw	时钟失效检测使能（Clock Failure Detection enable） 0: 关闭； 1: 开启。	
位 18	HEXTBYP5	0x0	rw	HEXT 旁路使能（High speed external crystal bypass） 只有在 HEXT 关闭时，软件才能操作该位。 0: 关闭； 1: 开启。	
位 17	HEXTSTBL	0x0	ro	HEXT 时钟稳定（High speed external crystal stable） 该位待 HEXT 稳定后由硬件置起。 0: 未稳定； 1: 已稳定。	
位 16	HEXTEN	0x0	rw	HEXT 使能（High speed external crystal enable） 该位可由软件置起或清除，也可在进入待机或深度睡眠模式时，由硬件清除。当系统时钟有用到 HEXT 时，该位无法清除。 0: 关闭； 1: 开启。	
位 15: 8	HICKCAL	0xXX	rw	HICK 时钟校准值（High speed internal clock calibration） 默认值为出厂校准初始值。 HICK 输出频率为 48 MHz 时，每 HICKCAL 数值的变化对应频率调整 240 kHz（设计值）；HICK 输出频率是 8 MHz 时，每 HICKCAL 数值的变化对应频率调整 40 kHz（设计值）。 注意：此位只有在 HICKCAL_KEY[7: 0]为 0x5A 的时候可被写入。	
位 7: 2	HICKTRIM	0x20	rw	HICK 时钟调整值（High speed internal clock trimming） 该数值和 HICKCAL[7: 0]数值共同决定 HICK 振荡器的频率，默认数值为 32，可以把 HICK 调整到精度±1%。	
位 1	HICKSTBL	0x1	ro	HICK 时钟稳定（High speed internal clock stable） 该位待 HICK 稳定后由硬件置起。 0: 未稳定； 1: 已稳定。	
位 0	HICKEN	0x1	rw	HICK 使能（High speed internal clock enable） 该位可由软件置起或清除，在退出待机或深度睡眠模式，或 HEXT 发生故障时，该位也可被硬件置起。当系统时钟有用到 HICK 时，该位无法清除。 0: 关闭； 1: 开启。	

#### 4.3.2 时钟配置寄存器 (CRM\_CFG)

域	简称	复位值	类型	功能
位 31	保留	0x00	resd	请保持默认值。
位 26: 24	CLKOUT_SEL	0x0	rw	内部时钟输出选择（Clock output selection） CLKOUT_SEL[3]在额外寄存器 1 (CRM_MISC1) 的位 16。 0000: 无； 0001: 保留； 0010: LICK； 0011: LEXT； 0100: SCLK； 0101: HICK； 0110: HEXT； 0111: PLL/2；

				1100: PLL/4; 1101: 保留; 1110: ADC。
位 27 位 23: 22	保留	0x00	resd	请保持默认值。
位 30: 29 位 21: 18	PLLMULT	0x00	rw	PLL 倍频系数 (PLL multiplication factor) { 位 30: 29, 位 21: 18} 000000: 2 倍频 000001: 3 倍频; 000010: 4 倍频 000011: 5 倍频; ..... 001100: 14 倍频 001101: 15 倍频; 001110: 16 倍频 001111: 16 倍频; 010000: 17 倍频 010001: 18 倍频; 010010: 19 倍频 010011: 20 倍频; ..... 111110: 63 倍频 111111: 64 倍频。
位 17	PLLHEXTDIV	0x0	rw	HEXT 分频后作为 PLL 输入时钟源 (HEXT division selection for PLL entry clock) 0: 不分频; 1: 分频系数为 2。
位 16	PLLRCSC	0x0	rw	PLL 输入时钟选择 (PLL reference clock select) 0: HICK 分频时钟 (4MHz) 作为 PLL 输入时钟; 1: HEXT 时钟作为 PLL 输入时钟源。
位 28 位 15: 14	ADCDIV	0x0	rw	ADC 分频因子 (ADC division) PCLK 分频后作为 ADC 时钟。 000: 2 分频; 001: 4 分频; 010: 6 分频; 011: 8 分频; 100: 2 分频; 101: 12 分频; 110: 8 分频; 111: 16 分频。
位 13: 11	APB2DIV	0x0	rw	APB2 分频因子 (APB2 division) HCLK 分频后作为 APB2 时钟。 0xx: 不分频; 100: 2 分频; 101: 4 分频; 110: 8 分频; 111: 16 分频。 注意: 软件必须保证 APB2 时钟频率不超过 120MHz。
位 10: 8	APB1DIV	0x0	rw	APB1 分频因子 (APB1 division) HCLK 分频后作为 APB1 时钟。 0xx: 不分频; 100: 2 分频; 101: 4 分频; 110: 8 分频; 111: 16 分频。 注意: 软件必须保证 APB1 时钟频率不超过 120MHz。
位 7: 4	AHBDIV	0x0	rw	AHB 分频因子 (AHB division) SCLK 分频后作为 AHB 时钟。 0xxx: 不分频; 1000: 2 分频; 1001: 4 分频; 1010: 8 分频; 1011: 16 分频; 1100: 64 分频; 1101: 128 分频; 1110: 256 分频; 1111: 512 分频。
位 3: 2	SCLKSTS	0x0	ro	系统时钟选择状态位 (System clock select status)

				00: HICK; 01: HEXT; 10: PLL; 11: 保留, 保持默认值。
位 1: 0	SCLKSEL	0x0	rw	系统时钟选择 (System clock select) 00: HICK; 01: HEXT; 10: PLL; 11: 保留, 保持默认值。

### 4.3.3 时钟中断寄存器 (CRM\_CLKINT)

访问: 无等待周期, 字, 半字和字节访问

域	简称	复位值	类型	功能
位 31: 24	保留	0x00	resd	保持默认值。
位 23	CFDFC	0x0	wo	清除时钟失效标志 (Clock failure detection flag clear) 由软件写'1'清除 CFDF。 0: 不清除; 1: 清除。
位 22: 21	保留	0x0	resd	保持默认值。
位 20	PLLSTBLFC	0x0	wo	清除 PLL 稳定标志 (PLL stable flag clear) 由软件写'1'清除 PLLSTBLF。 0: 不清除; 1: 清除。
位 19	HEXTSTBLFC	0x0	wo	清除 HEXT 稳定标志 (HEXT stable flag clear) 由软件写'1'清除 HEXTSTBLF。 0: 不清除; 1: 清除。
位 18	HICKSTBLFC	0x0	wo	清除 HICK 稳定标志 (HICK stable flag clear) 由软件写'1'清除 HICKSTBLF。 0: 不清除; 1: 清除。
位 17	LEXTSTBLFC	0x0	wo	清除 LEXT 稳定标志 (LEXT stable flag clear) 由软件写'1'清除 LEXTSTBLF。 0: 不清除; 1: 清除。
位 16	LICKSTBLFC	0x0	wo	清除 LICK 稳定标志 (LICK stable flag clear) 由软件写'1'清除 LICKSTBLF。 0: 不清除; 1: 清除。
位 15: 13	保留	0x0	resd	保持默认值。
位 12	PLLSTBLIEN	0x0	rw	PLL 稳定中断使能 (PLL stable interrupt enable) 0: 关闭; 1: 开启。
位 11	HEXTSTBLIEN	0x0	rw	HEXT 稳定中断使能 (HEXT stable interrupt enable) 0: 关闭; 1: 开启。
位 10	HICKSTBLIEN	0x0	rw	HICK 稳定中断使能 (HICK stable interrupt enable) 0: 关闭; 1: 开启。
位 9	LEXTSTBLIEN	0x0	rw	LEXT 稳定中断使能 (LEXT stable interrupt enable) 0: 关闭; 1: 开启。
位 8	LICKSTBLIEN	0x0	rw	LICK 稳定中断使能 (LICK stable interrupt enable) 0: 关闭; 1: 开启。
位 7	CFDF	0x0	ro	时钟失效标志 (Clock Failure Detection flag) 在 HEXT 时钟出现故障时, 由硬件置起。 0: 未出现; 1: 出现。
位 6: 5	保留	0x0	resd	保持默认值。

位 4	PLLSTBLF	0x0	ro	PLL 稳定标志 (PLL stable flag) 由硬件置起。 0: 未稳定; 1: 已稳定。
位 3	HEXTSTBLF	0x0	ro	HEXT 稳定标志 (HEXT stable flag) 由硬件置起。 0: 未稳定; 1: 已稳定。
位 2	HICKSTBLF	0x0	ro	HICK 稳定标志 (HICK stable flag) 由硬件置起。 0: 未稳定; 1: 已稳定。
位 1	LEXTSTBLF	0x0	ro	LEXT 稳定标志 (LEXT stable flag) 由硬件置起。 0: 未稳定; 1: 已稳定。
位 0	LICKSTBLF	0x0	ro	LICK 稳定中断标志 (LICK stable flag) 由硬件置起。 0: 未稳定; 1: 已稳定。

#### 4.3.4 APB2外设复位寄存器 (CRM\_APB2RST)

访问: 无等待周期, 字, 半字和字节访问

域	简称	复位值	类型	功能
位 31: 19	保留	0x0000	resd	保持默认值。
位 18	TMR17RST	0x0	rw	TMR17 复位 (TMR17 reset) 0: 无复位; 1: 复位。
位 17	TMR16RST	0x0	rw	TMR16 复位 (TMR16 reset) 0: 无复位; 1: 复位。
位 16	TMR15RST	0x0	rw	TMR15 复位 (TMR15 reset) 0: 无复位; 1: 复位。
位 15	保留	0x0	resd	保持默认值。
位 14	USART1RST	0x0	rw	USART1 复位 (USART1 reset) 0: 无复位; 1: 复位。
位 13	保留	0x0	resd	保持默认值。
位 12	SPI1RST	0x0	rw	SPI1 复位 (SPI1 reset) 0: 无复位; 1: 复位。
位 11	TMR1RST	0x0	rw	TMR1 复位 (TMR1 reset) 0: 无复位; 1: 复位。
位 10	保留	0x0	resd	保持默认值。
位 9	ADCRST	0x0	rw	ADC 复位 (ADC reset) 0: 无复位; 1: 复位。
位 8: 2	保留	0x00	resd	保持默认值。
位 1	EXINTRST	0x0	rw	EXINT 复位 (EXINT reset) 0: 无复位; 1: 复位。 注意软件读该位, 恒为 0。
位 0	SCFGCMPRST	0x0	rw	SCFG 和 CMP 复位 (SCFG and CMP reset) 0: 无复位; 1: 复位。

### 4.3.5 APB1外设复位寄存器 (CRM\_APB1RST)

访问：无等待周期，字，半字和字节访问

域	简称	复位值	类型	功能
位 31: 29	保留	0x0	resd	保持默认值。
位 28	PWCRST	0x0	rw	PWC 复位 (PWC reset) 0: 无复位； 1: 复位。
位 27: 23	保留	0x00	resd	保持默认值。
位 22	I2C2RST	0x0	rw	I2C2 复位 (I2C2 reset) 0: 无复位； 1: 复位。
位 21	I2C1RST	0x0	rw	I2C1 复位 (I2C1 reset) 0: 无复位； 1: 复位。
位 20: 18	保留	0x0	resd	保持默认值。
位 17	USART2RST	0x0	rw	USART2 复位 (USART2 reset) 0: 无复位； 1: 复位。
位 16: 15	保留	0x0	resd	保持默认值。
位 14	SPI2RST	0x0	rw	SPI2 复位 (SPI2 reset) 0: 无复位； 1: 复位。
位 13: 12	保留	0x0	resd	保持默认值。
位 11	WWDTRST	0x0	rw	窗口看门狗复位 (WWDT reset) 0: 无复位； 1: 复位。
位 10: 9	保留	0x0	resd	保持默认值。
位 8	TMR14RST	0x0	rw	TMR14 复位 (TMR14 reset) 0: 无复位； 1: 复位。
位 7: 5	保留	0x0	resd	保持默认值。
位 4	TMR6RST	0x0	rw	TMR6 复位 (TMR6 reset) 0: 无复位； 1: 复位。
位 3: 2	保留	0x0	resd	保持默认值。
位 1	TMR3RST	0x0	rw	TMR3 复位 (TMR3 reset) 0: 无复位； 1: 复位。
位 0	保留	0x0	resd	保持默认值。

### 4.3.6 AHB外设时钟使能寄存器 (CRM\_AHBEN)

访问：无等待周期，字、半字和字节访问

域	简称	复位值	类型	功能
位 31: 23	保留	0x000	resd	保持默认值。
位 22	GPIOFEN	0x0	rw	GPIOF 时钟使能 (GPIOF clock enable) 0: 关闭； 1: 开启。
位 21: 20	保留	0x0	resd	保持默认值。
位 19	GPIOCEN	0x0	rw	GPIOC 时钟使能 (GPIOC clock enable) 0: 关闭； 1: 开启。
位 18	GPIOBEN	0x0	rw	GPIOB 时钟使能 (GPIOB clock enable) 0: 关闭； 1: 开启。
位 17	GPIOAEN	0x0	rw	GPIOA 时钟使能 (GPIOA clock enable) 0: 关闭； 1: 开启。
位 16: 7	保留	0x000	resd	保持默认值。

位 6	CRCEN	0x0	rw	CRC 时钟使能 (CRC clock enable) 0: 关闭; 1: 开启。
位 5	保留	0x0	resd	保持默认值。
位 4	FLASHEN	0x1	rw	闪存时钟使能 (Flash clock enable) 该位配置睡眠或深度睡眠模式下闪存时钟使能。 0: 关闭; 1: 开启。
位 3	保留	0x0	resd	保持默认值。
位 2	SRAMEN	0x1	rw	SRAM 时钟使能 (SRAM clock enable) 该位配置睡眠或深度睡眠模式下 SRAM 时钟使能。 0: 关闭; 1: 开启。
位 1	保留	0x0	resd	保持默认值。
位 0	DMA1EN	0x0	rw	DMA 时钟使能 (DMA clock enable) 0: 关闭; 1: 开启。

#### 4.3.7 APB2外设时钟使能寄存器 (CRM\_APB2EN)

访问：字，半字和字节访问

但在 APB2 总线上的外设被访问时，将插入等待状态直到 APB2 的外设访问结束。

域	简称	复位值	类型	功能
位 31: 19	保留	0x0000	resd	保持默认值。
位 18	TMR17EN	0x0	rw	TMR17 时钟使能 (TMR17 clock enable) 0: 关闭; 1: 开启。
位 17	TMR16EN	0x0	rw	TMR16 时钟使能 (TMR16 clock enable) 0: 关闭; 1: 开启。
位 16	TMR15EN	0x0	rw	TMR15 时钟使能 (TMR15 clock enable) 0: 关闭; 1: 开启。
位 15	保留	0x0	resd	保持默认值。
位 14	USART1EN	0x0	rw	USART1 时钟使能 (USART1 clock enable) 0: 关闭; 1: 开启。
位 13	保留	0x0	resd	保持默认值。
位 12	SPI1EN	0x0	rw	SPI1 时钟使能 (SPI1 clock enable) 0: 关闭; 1: 开启。
位 11	TMR1EN	0x0	rw	TMR1 时钟使能 (TMR1 clock enable) 0: 关闭; 1: 开启。
位 10	保留	0x0	resd	保持默认值。
位 9	ADC1EN	0x0	rw	ADC1 时钟使能 (ADC1 clock enable) 0: 关闭; 1: 开启。
位 8: 1	保留	0x00	rw	保持默认值。
位 0	SCFGCMPEN	0x0	rw	SCFG 和 CMP 时钟使能 (SCFG and CMP clock enable) 0: 关闭; 1: 开启。

### 4.3.8 APB1外设时钟使能寄存器 (CRM\_APB1EN)

访问：字、半字和字节访问

通常无访问等待周期。但在 APB1 总线上的外设被访问时，将插入等待状态直到 APB1 外设访问结束。

域	简称	复位值	类型	功能
位 31: 29	保留	0x0	resd	保持默认值。
位 28	PWCEN	0x0	rw	PWC 时钟使能 (Power control clock enable) 0: 关闭; 1: 开启。
位 27: 23	保留	0x00	resd	保持默认值。
位 22	I2C2EN	0x0	rw	I2C2 时钟使能 (I2C2 clock enable) 0: 关闭; 1: 开启。
位 21	I2C1EN	0x0	rw	I2C1 时钟使能 (I2C1 clock enable) 0: 关闭; 1: 开启。
位 20: 18	保留	0x0	resd	保持默认值。
位 17	USART2EN	0x0	rw	USART2 时钟使能 (USART2 clock enable) 0: 关闭; 1: 开启。
位 16: 15	保留	0x0	resd	保持默认值。
位 14	SPI2EN	0x0	rw	SPI2 时钟使能 (SPI2 clock enable) 0: 关闭; 1: 开启。
位 13: 12	保留	0x0	resd	保持默认值。
位 11	WWDTEN	0x0	rw	窗口看门狗时钟使能 (WWDT clock enable) 0: 关闭; 1: 开启。
位 10: 9	保留	0x0	resd	保持默认值。
位 8	TMR14EN	0x0	rw	TMR14 时钟使能 (TMR14 clock enable) 0: 关闭; 1: 开启。
位 7: 5	保留	0x0	resd	保持默认值。
位 4	TMR6EN	0x0	rw	TMR6 时钟使能 (TMR6 clock enable) 0: 关闭; 1: 开启。
位 3: 2	保留	0x0	resd	保持默认值。
位 1	TMR3EN	0x0	rw	TMR3 时钟使能 (TMR3 clock enable) 0: 关闭; 1: 开启。
位 0	保留	0x0	resd	保持默认值。

### 4.3.9 电池供电域控制寄存器 (CRM\_BPDC)

访问：0 到 3 等待周期，字、半字和字节访问；当连续对该寄存器进行访问时，将插入等待状态。

注意：电池供电域控制寄存器中 (CRM\_BPDC) LEXTEN、LEXTBYP、ERTCSEL 和 ERTCEN 位处于电池供电域。因此，这些位在复位后处于写保护状态，只有在电源控制寄存器 (PWC\_CTRL) 中的 BPWEN 位置位后才能对这些位进行改动。这些位只能由电池供电域软件复位清除。任何内部或外部复位都不会影响这些位。

域	简称	复位值	类型	功能
位 31: 17	保留	0x0000	resd	保持默认值。
位 16	BPDRST	0x0	rw	电池供电域软件复位 (Battery powered domain software reset) 0: 无复位; 1: 复位。
位 15	ERTCEN	0x0	rw	ERTC 时钟使能 (ERTC clock enable) 由软件置位或清零。 0: 关闭; 1: 开启。
位 14: 10	保留	0x00	resd	保持默认值。

				ERTC 时钟选择 (ERTC clock selection) 确定了 ERTC 时钟选择后, 如果想要再次更改, 必须设置 BPDRST 位复位后, 才能重新改写 ERTC 时钟选择。 00: 无; 01: LEXT; 10: LICK; 11: HEXT/32。
位 9: 8	ERTCSEL	0x0	rw	保持默认值。
位 7: 3	保留	0x00	resd	LEXT 旁路使能 (Low speed external crystal bypass) 0: 关闭; 1: 开启。
位 2	LEXTBYP	0x0	rw	LEXT 稳定 (External low-speed oscillator stable) 该位待 LEXT 稳定后由硬件置起。 0: 未稳定; 1: 已稳定。
位 1	LEXTSTBL	0x0	ro	LEXT 使能 (External low-speed oscillator enable) 0: 关闭; 1: 开启。
位 0	LEXTEN	0x0	rw	

#### 4.3.10 控制/状态寄存器 (CRM\_CTRLSTS)

除复位标志外由系统复位清除, 复位标志能由电源复位或写 RSTFC 位进行清除。访问: 0 到 3 等待周期, 字、半字和字节访问; 当连续对该寄存器进行访问时, 将插入等待状态。

域	简称	复位值	类型	功能
位 31	LPRSTF	0x0	ro	低功耗复位标志 (Low-power reset flag) 该位由硬件置起, 软件写 RSTFC 位清除。 0: 无; 1: 有。
位 30	WWDTRSTF	0x0	ro	窗口看门狗复位标志 (WWDT reset flag) 该位由硬件置起, 软件写 RSTFC 位清除。 0: 无; 1: 有。
位 29	WDTRSTF	0x0	ro	看门狗复位标志 (WDT reset flag) 该位由硬件置起, 软件写 RSTFC 位清除。 0: 无; 1: 有。
位 28	SWRSTF	0x0	ro	软件复位标志 (Software reset flag) 该位由硬件置起, 软件写 RSTFC 位清除。 0: 无; 1: 有。
位 27	PORRSTF	0x1	ro	上电/掉电复位标志 (POR/LVR reset flag) 该位由硬件置起, 软件写 RSTFC 位清除。 0: 无; 1: 有。
位 26	NRSTF	0x1	ro	NRST 管脚复位标志 (NRST reset flag) 该位由硬件置起, 软件写 RSTFC 位清除。 0: 无; 1: 有。
位 25	保留	0x0	resd	保持默认值。
位 24	RSTFC	0x0	rw	清除复位标志 (Reset flag clear) 由软件写'1'来清除复位标志。 0: 无作用; 1: 清除复位标志。
位 23: 2	保留	0x000000	resd	保持默认值。
位 1	LICKSTBL	0x0	ro	LICK 稳定 (LICK stable) 0: 未稳定; 1: 已稳定。
位 0	LICKEN	0x0	rw	LICK 使能 (LICK enable) 0: 关闭; 1: 开启。

### 4.3.11 AHB外设复位寄存器 (CRM\_AHBRST)

访问：无等待周期，字，半字和字节访问

域	简称	复位值	类型	功能
位 31: 23	保留	0x000	resd	保持默认值。
位 22	GPIOFRST	0x0	rw	GPIOF 复位 (GPIOF reset) 由软件置'1'或清'0' 0: 无作用; 1: 复位 GPIOF。
位 21: 20	保留	0x0	resd	保持默认值。
位 19	GPIOCRST	0x0	rw	GPIOC 复位 (GPIOC reset) 由软件置'1'或清'0' 0: 无作用; 1: 复位 GPIOC。
位 18	GPIOBRST	0x0	rw	GPIOB 复位 (GPIOB reset) 由软件置'1'或清'0' 0: 无作用; 1: 复位 GPIOB。
位 17	GPIOARST	0x0	rw	GPIOA 复位 (GPIOA reset) 由软件置'1'或清'0' 0: 无作用; 1: 复位 GPIOA。
位 16: 0	保留	0x00000	resd	保持默认值。

### 4.3.12 PLL配置寄存器 (CRM\_PLL)

域	简称	复位值	类型	功能
位 31	PLLCFGEN	0x0	rw	PLL 配置使能 (PLL Configure Enable) 0: PLL 使用常规整数倍频配置模式, 使用 PLL_FREF 和 PLLMUL 寄存器配置 PLL 1: PLL 使用灵活配置模式, 使用 PLL_MS/PLL_NS/PLL_FR 寄存器值配置 PLL。
位 30: 27	保留	0x0	resd	保持默认值。
位 26: 24	PLL_FREF	0x0	rw	PLL 输入时钟选择, 仅在 PLLCFGGEN=0 时起作用 000: PLL 使用 3.9 MHz ~ 5 MHz 输入时钟; 001: PLL 使用 5.2 MHz ~ 6.25 MHz 输入时钟; 010: PLL 使用 7.8125 MHz ~ 8.33 MHz 输入时钟; 011: PLL 使用 8.33 MHz ~ 12.5 MHz 输入时钟; 100: PLL 使用 15.625 MHz ~ 20.83 MHz 输入时钟; 101: PLL 使用 20.83 MHz ~ 31.255 MHz 输入时钟; 110: 保留; 111: 保留。
位 23: 17	保留	0x00	resd	保持默认值。
位 16: 8	PLL_NS	0x01F	rw	PLL 倍频系数 PLL_NS 范围 (31~500)
位 7: 4	PLL_MS	0x1	rw	PLL 预分频系数 PLL_MS 范围 (1~15)
位 3	保留	0x0	resd	保持默认值。
位 2: 0	PLL_FR	0x0	rw	PLL 后分频配置值 PLL_FR 范围 (0~5) 000: PLL 后分频系数为 1, 1 分频输出; 001: PLL 后分频系数为 2, 2 分频输出; 010: PLL 后分频系数为 4, 4 分频输出; 011: PLL 后分频系数为 8, 8 分频输出; 100: PLL 后分频系数为 16, 16 分频输出; 101: PLL 后分频系数为 32, 32 分频输出; 其他: 保留 请注意 PLL_FR 值和后分频系数对应关系

### 4.3.13 额外寄存器1 (CRM\_MISC1)

域	简称	复位值	类型	功能
位 31: 28	CLKOUTDIV	0x0	rw	<p>CLKOUT 分频因子 (Clock output division) CLKOUT 输出频率的分频值设定。 0xxx: 不分频; 1000: 2 分频; 1001: 4 分频; 1010: 8 分频; 1011: 16 分频; 1100: 64 分频; 1101: 128 分频; 1110: 256 分频; 1111: 512 分频。</p>
位 27: 26	保留	0x0	resd	保持默认值。
位 25	HICKDIV	0x0	rw	<p>HICK 6 分频 (HICK 6 divider selection) 该位选择使用 HICK 时钟还是 HICK 的 6 分频时钟, 选择 HICK 的 6 分频时钟的话, 时钟频率为 8 MHz, 选择不分频的话, 时钟频率为 48 MHz。 0: 分频; 1: 不分频。 注意: 1、当 HICK 作为 PLL 时钟源时, PLL 使能期间, 请勿修改 HICKDIV。 2、不论何种情况 HICK 输入到 PLL 时的频率都固定为 4 MHz。</p>
位 24: 21	保留	0x0	resd	保持默认值。
位 20: 17	保留	0x8	resd	保持默认值。
位 16	CLKOUT_SEL[3]	0x0	rw	内部时钟输出选择 (Clock output selection) 搭配 CRM_CFG 寄存器位 26: 24 使用。
位 15: 8	保留	0x00	resd	保持默认值。
位 7: 0	HICKCAL_KEY	0x00	rw	HICKCAL 写入键值 (HICK calibration key) 此字段为 0xA 时, HICKCAL [7: 0]才可被写入。

### 4.3.14 额外寄存器2 (CRM\_MISC2)

域	简称	复位值	类型	功能
位 31: 10	保留	0x000000	resd	保持默认值。
位 9	HICK_TO_SCLK	0x0	rw	<p>HICK 作为系统时钟的频率选择位 (HICK as system clock frequency select) 当 SCLKSEL 选择 HICK 为时钟源时, SYSCLK 的频率为 0: 固定是 8Mhz, 即选择 HICK 时钟的 6 分频; 1: 根据 HICKDIV 设定可能为 48M 或 8M。</p>
位 8: 6	保留	0x0	resd	保持默认值。
位 5: 4	AUTO_STEP_EN	0x0	rw	<p>自动滑顺频率切换使能 (auto step system clock switch enable) 为使切换系统时钟源到 PLL 或是切换 AHB 预分频由大到小平顺(系统频率由小变大), 建议系统频率操作目标大于 108MHz 时启动自动滑顺频率切换。 当自动滑顺频率切换功能作用时, 硬件会暂停 AHB 总线, 直到整个自动滑顺频率切换完成才恢复。此期间 DMA 仍正常工作, 中断事件会被记忆并待 AHB 总线恢复后 NVIC 即可处理。 00: 关闭自动滑顺频率切换; 01: 保留; 10: 保留; 11: 开启自动滑顺频率切换, 当 AHBDIV 或 SCLKSEL 两个寄存器被改动时, 会自动触发自动滑顺频率切换功能。</p>
位 3: 0	保留	0xd	resd	固定为 0xd, 请勿修改。

# 5 内嵌闪存控制器 (FLASH)

## 5.1 FLASH介绍

闪存由主存储器、信息块、闪存寄存器这三个部分组成。

- 主存储器容量可达 64K 字节
- 信息块由 4K 字节的系统启动程序代码区和用户系统数据区组成。系统启动程序使用 USART1 或 USART2 实现 ISP 编程
- 主存储器只有闪存容量为 64K 字节的片 1 闪存，包含 64 扇区，每扇区大小为 1K 字节。

表 5-1 闪存存储结构 (64K)

结构		名称	地址范围
主存储器	片 1 (Bank1) 64KB	扇区 0	0x0800 0000 – 0x0800 03FF
		扇区 1	0x0800 0400 – 0x0800 07FF
		扇区 2	0x0800 0800 – 0x0800 0BFF
		...	...
		扇区 63	0x0800 FC00 – 0x0800 FFFF
		启动程序代码区 4KB	0x1FFF E400 – 0x1FFF F3FF
信息块		用户系统数据区 512B	0x1FFF F800 – 0x1FFF F9FF

主存储器只有闪存容量为 32K 字节的片 1 闪存，包含 32 扇区，每扇区大小为 1K 字节。

表 5-2 闪存存储组织 (32K)

结构		名称	地址范围
主存储器	片 1 (Bank1) 32KB	扇区 0	0x0800 0000 – 0x0800 03FF
		扇区 1	0x0800 0400 – 0x0800 07FF
		扇区 2	0x0800 0800 – 0x0800 0BFF
		...	...
		扇区 31	0x0800 7C00 – 0x0800 7FFF
		启动程序代码区 4KB	0x1FFF E400 – 0x1FFF F3FF
信息块		用户系统数据区 512B	0x1FFF F800 – 0x1FFF F9FF

主存储器只有闪存容量为 16K 字节的片 1 闪存，包含 16 扇区，每扇区大小为 1K 字节。

表 5-3 闪存存储组织 (16K)

结构		名称	地址范围
主存储器	片 1 (Bank1) 16KB	扇区 0	0x0800 0000 – 0x0800 03FF
		扇区 1	0x0800 0400 – 0x0800 07FF
		扇区 2	0x0800 0800 – 0x0800 0BFF
		...	...
		扇区 15	0x0800 3C00 – 0x0800 3FFF
		启动程序代码区 4KB	0x1FFF E400 – 0x1FFF F3FF
信息块		用户系统数据区 512B	0x1FFF F800 – 0x1FFF F9FF

### 用户系统数据区

每次系统复位后将从闪存信息块中读出系统数据信息并保存在用户系统数据寄存器 (FLASH\_USD) 以及擦除编程保护状态寄存器 (FLASH\_EPPS) 中。

每个系统数据实际占用 2 个字节，低字节对应系统数据的内容，高字节对应系统数据的反码，用于验证选择位的正确性。当读出的高字节不等于低字节的反码时（高字节及低字节均为 0xFF 时除外），系统数据装载器会产生一个系统数据错误的标志 (USDERR)，并把对应的系统数据及其反码都强置为 0xFF。

注意：用户系统数据内容的更新需要一次系统复位才能真正实现。

表 5-4 用户系统数据说明

地址	位	内容
0x1FFF_F800	[7:0]	FAP[7:0]: 闪存访问保护 (访问保护启动/解除结果存放在寄存器 FLASH_USD[1] 以及 [26])

		0xA5: 闪存访问保护解除 0xCC: 高级闪存访问保护启动 其他值: 低级闪存访问保护启动
	[15:8]	<b>nFAP[7:0]:</b> FAP[7:0]的反码 <b>SSB[7:0]:</b> 系统配置字节 (存放在寄存器 FLASH_USD[9:2])
	[23:16]	<b>位 7:5</b> 保留不用 <b>位 4 (nBOOT1)</b> nBOOT1: 和 BOOT0 引脚一起决定启动模式, 当 BOOT0 = 1 时: 0: 由 SRAM 启动 1: 由系统启动程序代码区启动 <b>位 3</b> 保留不用 <b>位 2 (nSTDBY_RST)</b> 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 <b>位 1 (nDEPSLP_RST)</b> 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 <b>位 0 (nWDT_ATO_EN)</b> 0: 看门狗自启动开启 1: 看门狗自启动关闭
	[31:24]	<b>nSSB[7:0]:</b> SSB[7:0]的反码
0x1FFF_F804	[7:0]	<b>Data0[7:0]:</b> 用户数据 0 (存放在寄存器 FLASH_USD[17:10])
	[15:8]	<b>nData0[7:0]:</b> Data0[7:0]的反码
	[23:16]	<b>Data1[7:0]:</b> 用户数据 1 (存放在寄存器 FLASH_USD[25:18])
	[31:24]	<b>nData1[7:0]:</b> Data1[7:0]的反码
	[7:0]	<b>EPP0[7:0]:</b> 闪存擦写保护字节 0 (存放在寄存器 FLASH_EPPS[7:0]) 用于保护主闪存存储器的扇区 0 ~ 扇区 31, 每个比特位保护 4 个扇区 (1K 字节/扇区) 0: 擦写保护启动 1: 擦写保护解除
0x1FFF_F808	[15:8]	<b>nEPP0[7:0]:</b> EPP0[7:0]的反码
	[23:16]	<b>EPP1[7:0]:</b> 闪存擦写保护字节 1 (存放在寄存器 FLASH_EPPS[15:8]) 用于保护主闪存存储器的扇区 32 ~ 扇区 63, 每个比特位保护 4 个扇区 (1K 字节/扇区) 0: 擦写保护启动 1: 擦写保护解除
	[31:24]	<b>nEPP1[7:0]:</b> EPP1[7:0]的反码
0x1FFF_F80C	[7:0]	<b>EPP2[7:0]:</b> 闪存擦写保护字节 2 (存放在寄存器 FLASH_EPPS[23:16]) 保留不用
	[15:8]	<b>nEPP2[7:0]:</b> EPP2[7:0]的反码
	[23:16]	<b>EPP3[7:0]:</b> 闪存擦写保护字节 3 (存放在寄存器 FLASH_EPPS[31:24]) 其中位 6:0 保留不用 位 7 用于保护主存扩展区 0: 擦写保护启动 1: 擦写保护解除
0x1FFF_F810	[31:24]	<b>nEPP3[7:0]:</b> EPP3[7:0]的反码
	[7:0]	<b>Data2[7:0]:</b> 用户数据 2
	[15:8]	<b>nData2[7:0]:</b> Data2[7:0]的反码
	[23:16]	<b>Data3[7:0]:</b> 用户数据 3
0x1FFF_F814	[31:24]	<b>nData3[7:0]:</b> Data3[7:0]的反码
	[7:0]	<b>Data4[7:0]:</b> 用户数据 4
	[15:8]	<b>nData4[7:0]:</b> Data4[7:0]的反码
	[23:16]	<b>Data5[7:0]:</b> 用户数据 5
0x1FFF_F9FC	[31:24]	<b>nData5[7:0]:</b> Data5[7:0]的反码
	.	.
	[7:0]	<b>Data248[7:0]:</b> 用户数据 248
	[15:8]	<b>nData248[7:0]:</b> Data248[7:0]的反码
	[23:16]	<b>Data249[7:0]:</b> 用户数据 249
	[31:24]	<b>nData249[7:0]:</b> Data249[7:0]的反码

## 5.2 主存储器操作

### 5.2.1 解锁/锁定

复位后，主存储器默认是被锁定的，此时不允许配置闪存控制寄存器（FLASH\_CTRL），需要对闪存解锁后才能成功实现对闪存的写入与擦除操作。

#### 解锁流程：

对闪存解锁寄存器（FLASH\_UNLOCK）顺序写入键值 KEY1(0x45670123)和键值 KEY2(0xCDEF89AB)，能够解锁对应区域闪存。

**注意：**解锁必须顺序写入正确的键值，否则会产生总线错误并且闪存会被锁死，直到下一次复位才能恢复。

#### 锁定流程：

软件置起闪存控制寄存器（FLASH\_CTRL）中的 OPLK 位，锁定对应区域闪存。

### 5.2.2 擦除

编程之前必须先进行擦除操作，主存储器有扇区擦除和整片擦除两种擦除方式。

#### 扇区擦除

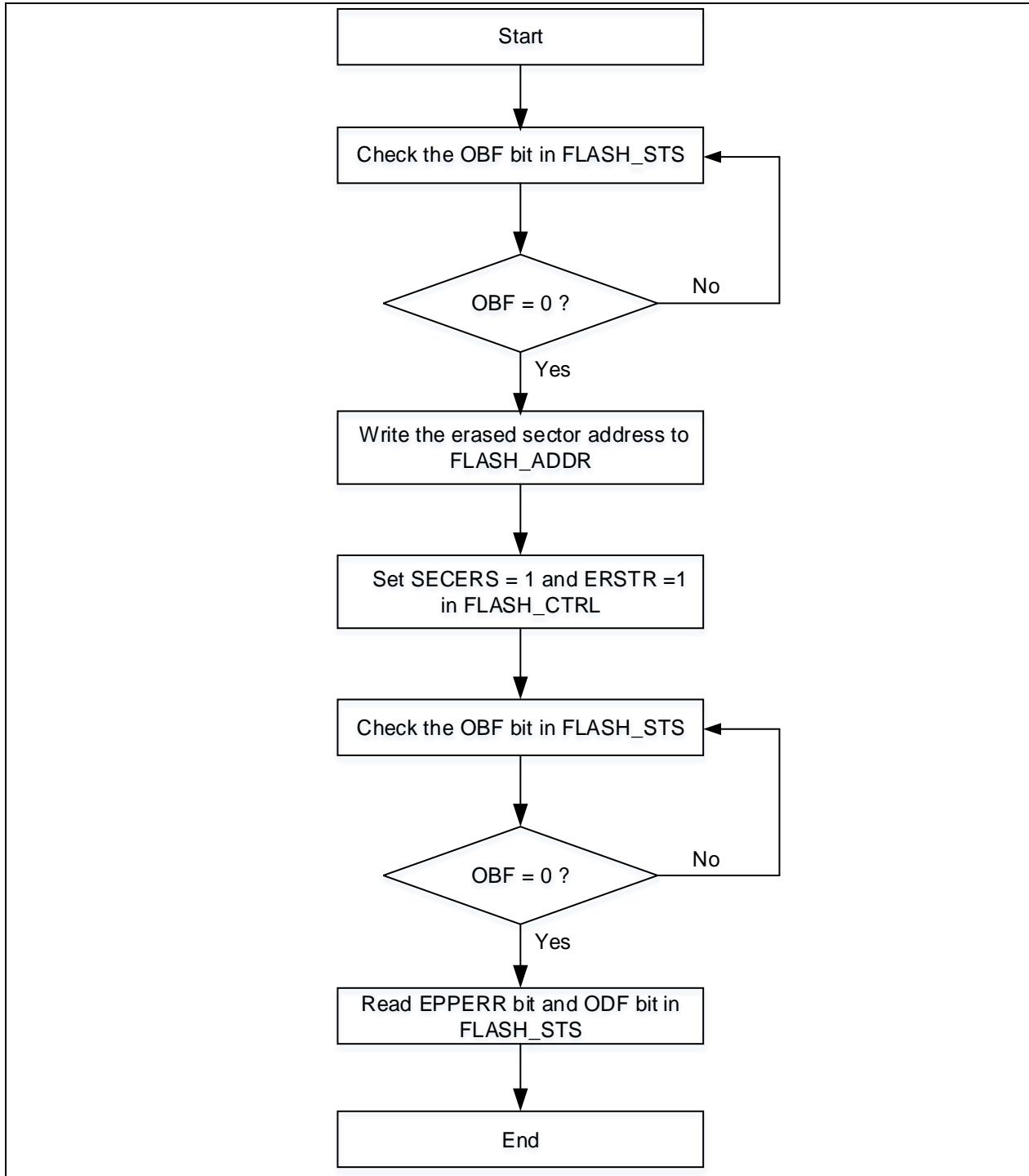
主闪存存储器的每一扇区都可以使用扇区擦除功能独立擦除。

擦除流程如下：

- 检查 FLASH\_STS 寄存器的 OBF 位，确认没有正在进行的闪存操作；
- 对闪存地址寄存器（FLASH\_ADDR）写入要擦除的扇区地址；
- 对闪存控制寄存器（FLASH\_CTRL）的 SECERS 位以及 ERSTR 位均置 1，启动扇区擦除；
- 等待 FLASH\_STS 寄存器的 OBF 位变为‘0’，并查询 FLASH\_STS 寄存器的 EPPERR 位和 ODF 位，确认擦除结果。

**注意：**当启动程序代码区域是设定为主存扩展区时，执行扇区擦除实际上是对整个主存扩展区的擦除。

图 5-1 主存储器扇区擦除流程图



### 整片擦除

主闪存存储器可以使用整片擦除功能直接擦除。

擦除流程如下：

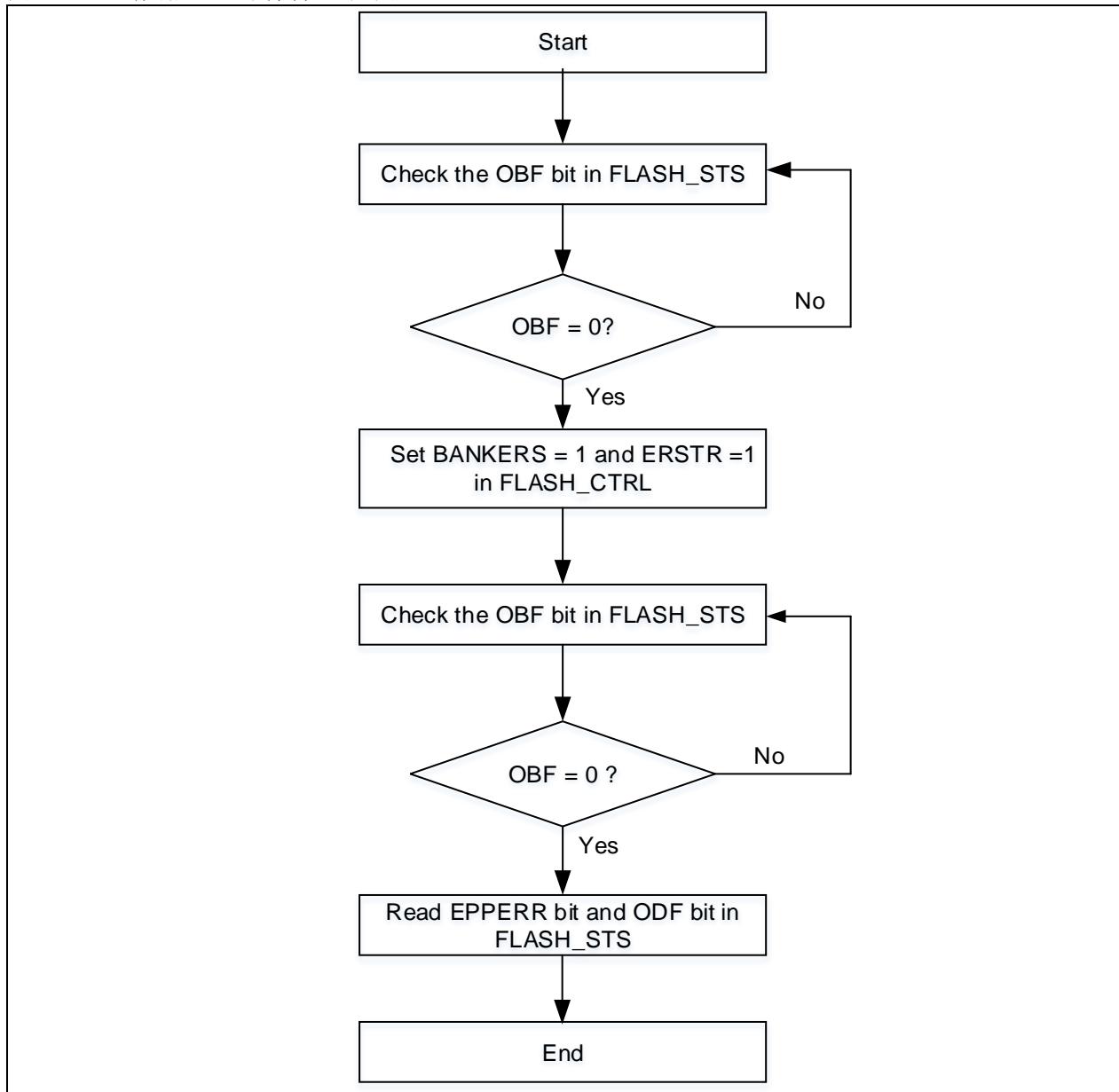
- 检查 FLASH\_STS 寄存器的 OBF 位，确认没有正在进行的闪存操作；
- 对闪存控制寄存器（FLASH\_CTRL）的 BANKERS 位以及 ERSTR 位均置 1，启动整片擦除；
- 等待 FLASH\_STS 寄存器的 OBF 位变为‘0’，并查询 FLASH\_STS 寄存器的 EPPERR 位和 ODF 位，确认擦除结果。

注意：

- 1) 当启动程序代码区域是设定为主存扩展区时，执行整片擦除操作会自动擦除主闪存以及主存扩展区。

- 2) 擦除期间进行读闪存的操作，将导致 CPU 会被暂停直到擦除完成才处理读闪存操作。  
 3) 擦除操作前必须保证内部的 HICK 有打开

图 5-2 主存储器整片擦除流程图



### 5.2.3 编程

当想要改写主存储器的内容时，可以通过主存储器编程流程完成一次写入 32 位、16 位或 8 位的数据。主存储器编程流程：

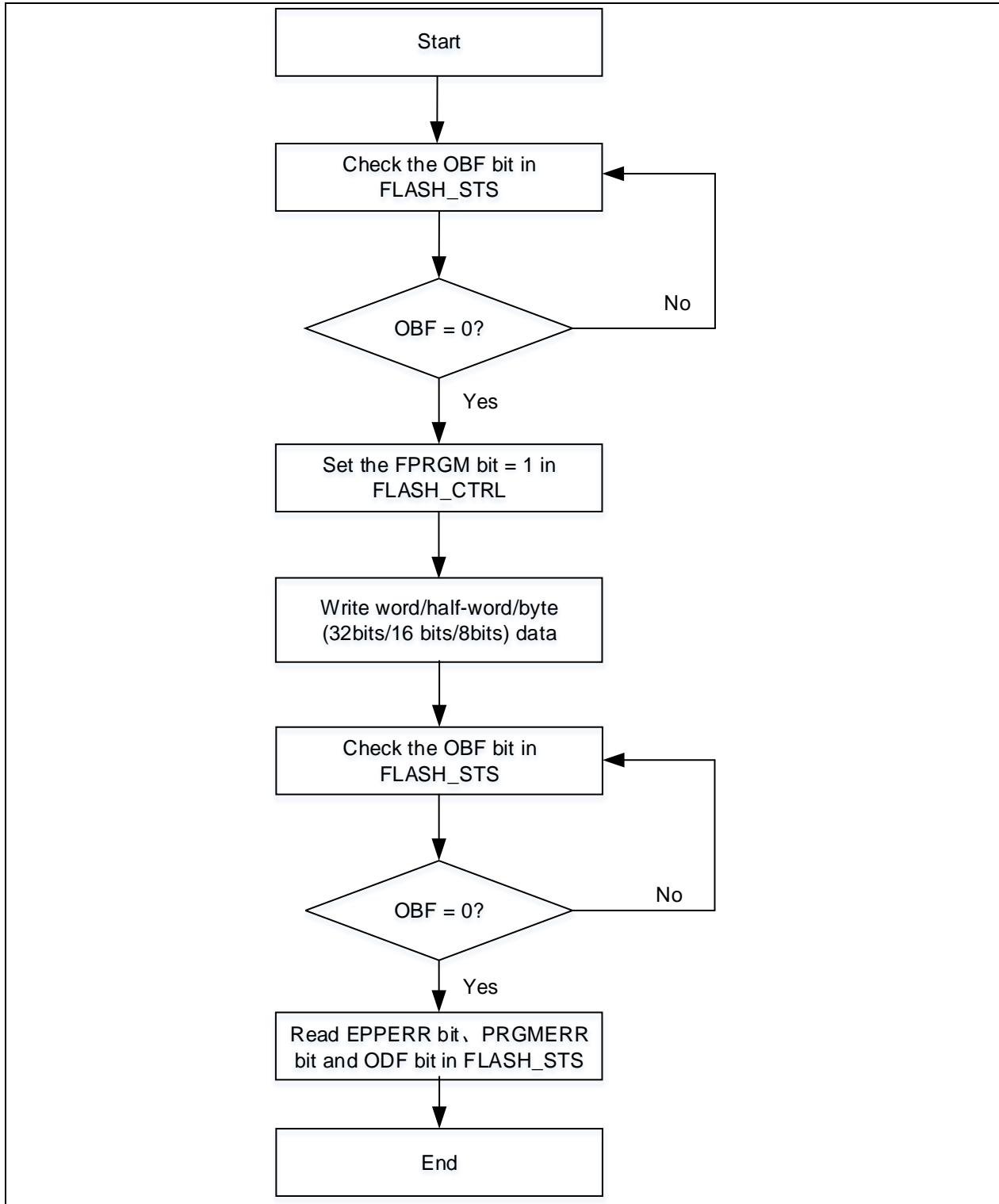
检查 FLASH\_STS 寄存器的 OBF 位，确认没有正在进行的闪存操作；

- 对闪存控制寄存器（FLASH\_CTRL）的 FPRGM 位置 1，此时可以接受对主闪存的编程指令；
- 对指定的地址写入要编程的数据（任意字/半字/字节）；
- 等待 FLASH\_STS 寄存器的 OBF 位变为‘0’，并查询 FLASH\_STS 寄存器的 EPPERR 位、PRGMERR 位和 ODF 位，确认编程结果。

注意：

- 1) 当要写入的地址未被提前擦除时，除非要写入的数据值是全 0，否则编程不被执行，并置位 FLASH\_STS 寄存器的 PRGMERR 位来告知编程发生错误。
- 2) 编程期间进行读闪存的操作，将导致 CPU 会被暂停直到编程完成才处理读闪存操作。
- 3) 编程操作前必须保证内部的 HICK 有打开

图 5-3 主存储器编程流程图



#### 5.2.4 读取

通过 CPU 的 AHB 总线可以直接寻址访问主闪存存储区。

### 5.3 主存扩展区操作

启动程序代码区也可以设定为主存扩展区存放用户应用代码。当作为主存扩展区时，其操作方法，包括读取、解锁、擦除、编程都跟主存储器相同。

## 5.4 用户系统数据区操作

### 5.4.1 解锁/锁定

复位后，用户系统数据区默认是锁定的，需要在闪存解锁后再对用户系统数据区解锁才能成功实现写入与擦除操作。

#### 解锁流程：

对闪存解锁寄存器(FLASH\_UNLOCK)顺序写入键值 KEY1(0x45670123)和键值 KEY2(0xCDEF89AB); 对闪存用户系统数据解锁寄存器 (FLASH\_USD\_UNLOCK) 顺序写入键值 KEY1 (0x45670123) 和键值 KEY2 (0xCDEF89AB)，闪存控制寄存器 (FLASH\_CTRL) 中的 USDULKS 位将被硬件自动置起，表示允许对用户系统数据区的写、擦除操作。

**注意：**解锁必须顺序写入正确的键值，否则会产生总线错误并且闪存会被锁死，直到下一次复位才能恢复。

#### 锁定流程：

软件清除闪存控制寄存器 (FLASH\_CTRL) 中的 USDULKS 位，锁定用户系统数据区。

### 5.4.2 擦除

在编程之前必须先进行擦除操作，用户系统数据区域可单独实现擦除功能。

#### 擦除流程如下：

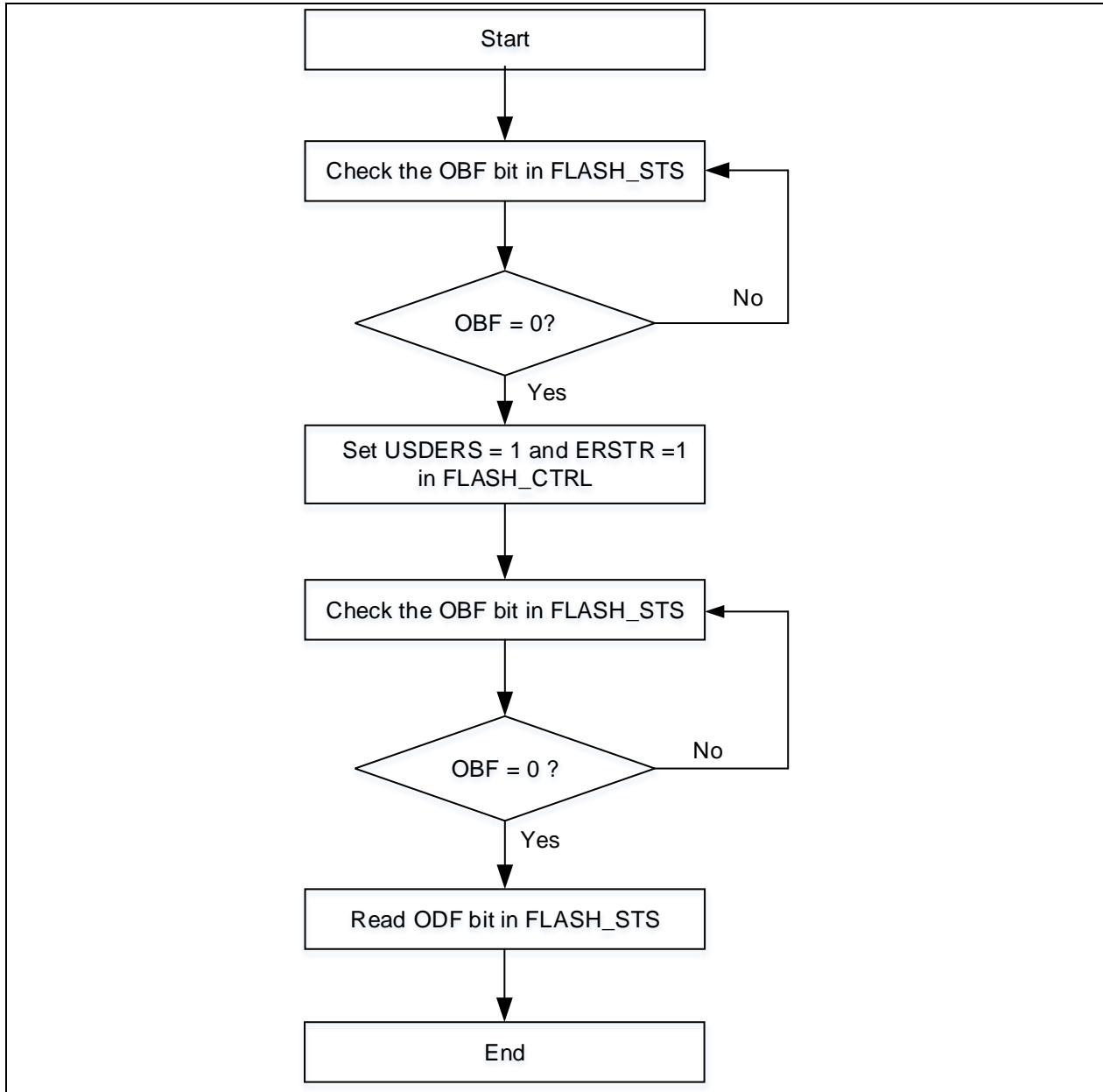
- 检查闪存状态寄存器 (FLASH\_STS) 的 OBF 位，确认没有正在进行的闪存操作；
- 对闪存控制寄存器 (FLASH\_CTRL) 的 USDRS 位以及 ERSTR 位均置 1，启动整块系统数据区擦除；
- 等待闪存状态寄存器 (FLASH\_STS) 的 OBF 位变为‘0’，并查询闪存状态寄存器 (FLASH\_STS) 的 ODF 位，确认擦除结果。

#### 注意：

擦除期间进行读闪存的操作，将导致 CPU 会被暂停直到擦除完成才处理读闪存操作。

擦除操作前必须保证内部的 HICK 有打开

图 5-4 系统数据区擦除图



### 5.4.3 编程

当想要改写用户系统数据区域的内容时，可以通过用户系统数据区编程流程完成一次写入 32 位或 16 位数据。

系统数据区的编程流程：

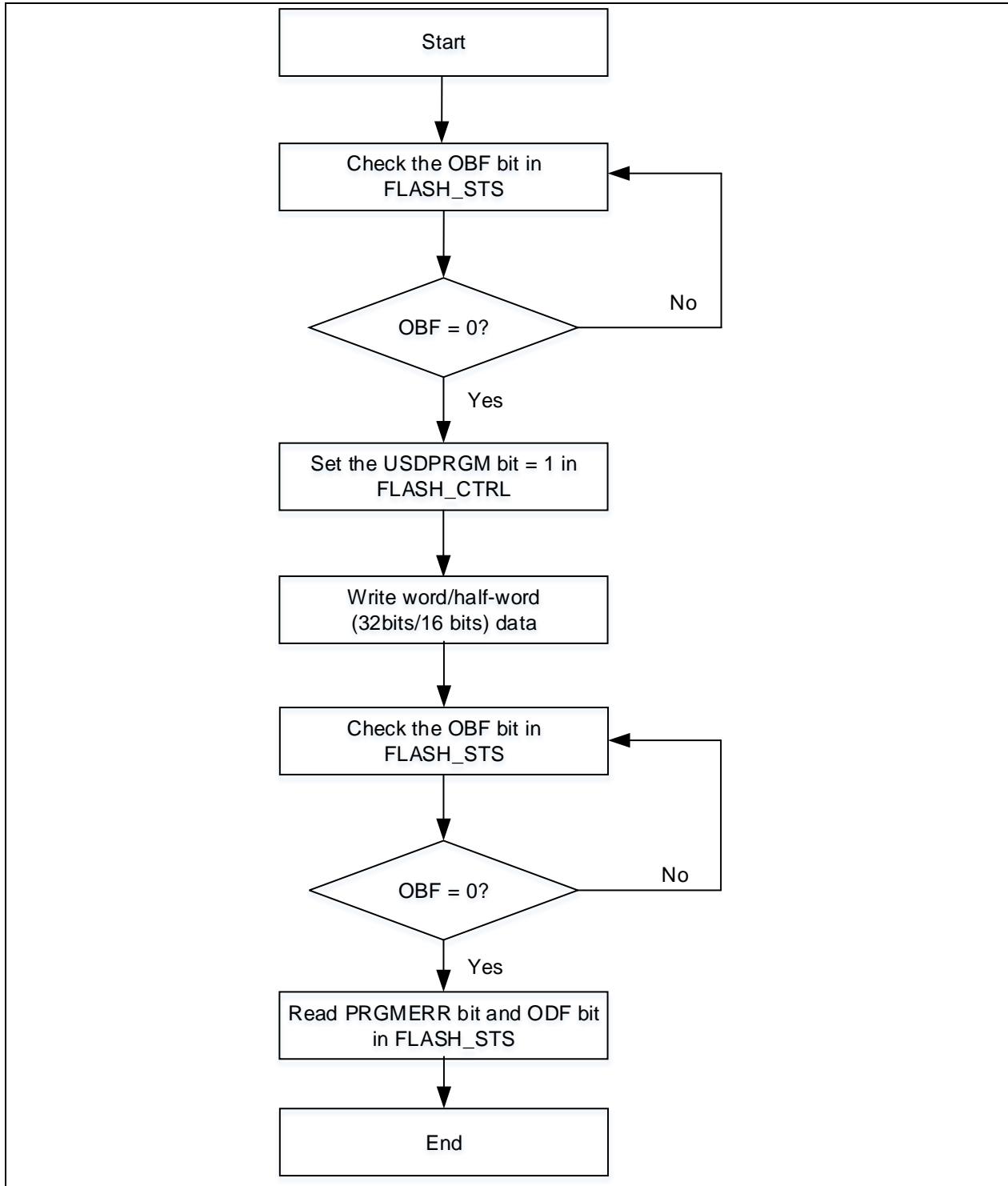
- 检查闪存状态寄存器（FLASH\_STS）的 OBF 位，确认没有正在进行的闪存操作；
- 对闪存控制寄存器（FLASH\_CTRL）的 USDPRGM 位置 1，此时可以接受对用户系统数据区的编程指令；
- 对指定的地址写入要编程的数据（任意字/半字）；
- 等待 FLASH\_STS 寄存器的 OBF 位变为‘0’，并查询闪存状态寄存器（FLASH\_STS）的 PRGMERR 位和 ODF 位，确认编程结果。

注意：

编程期间进行读闪存的操作，将导致 CPU 会被暂停直到编程完成才处理读闪存操作。

编程操作前必须保证内部的 HICK 有打开

图 5-5 系统数据区编程图



#### 5.4.4 读取

通过 CPU 的 AHB 总线可以直接寻址访问用户系统数据区。

### 5.5 闪存保护

闪存存储器有访问保护以及擦写保护两种保护方式。

#### 5.5.1 访问保护

闪存访问保护分为两类：闪存低级访问保护，闪存高级访问保护。

访问保护启动后，只允许闪存程序对闪存存储器数据进行读出访问，禁止在调试模式下或是从非主闪存存

储器启动对闪存存储器数据的读出访问。

#### 闪存低级访问保护

当 nFAP 字节和 FAP 字节存放的内容不等于 0x5A 和 0xA5 以及不等于 0x33 和 0xCC 时，闪存在系统复位后，将启动闪存低级访问保护。

此保护下，用户可以重新擦除系统数据区，并对 FAP 字节写入 0xA5 解除闪存低级访问保护（从低级保护状态变为未保护状态，将自动产生对主闪存以及主存扩展区的整片擦除操作），最后进行系统复位，系统数据装载器重新加载系统数据信息，更新闪存访问保护解除信息（FAP 字节）。

#### 闪存高级访问保护

当 nFAP 字节存放的内容等于 0x33，并且 FAP 字节存放的内容等于 0xCC 时，闪存在系统复位后，将启动闪存高级访问保护。

一旦此保护启动后，将禁止用户重新擦除以及写入系统数据区。该保护的解除只能通过设置寄存器 FAP\_HL\_DIS 位来实现硬件自动擦除用户系统数据区。

**注意：**

- 1) 主存扩展区也支持访问保护功能
- 2) 如果访问保护被置位的时候仍然处于调试模式，必须用 POR（上电复位）代替系统复位 清除调试模式，才能恢复闪存程序访问闪存存储器数据的权限。

下表是启动闪存访问保护后，闪存不同区域访问权限说明：

表 5-5 闪存访问权限

区域		保护等级			访问权限		
		调试模式或是从 SRAM 启动以及从启动 程序代码区启动			从主闪存启动		
		读	写	擦除	读	写	擦除
主闪存区	低级访问保护	禁止		禁止 (1) (2)	允许		
	高级访问保护 (4)	禁止			允许		
用户系统数据区	低级访问保护	禁止		允许	允许		
	高级访问保护 (4)	禁止		禁止 (3)	禁止	禁止 (3)	

(1) 主闪存区会在解除闪存访问保护时被硬件自动擦除

(2) 只禁止扇区擦除，允许整片擦除

(3) 用户系统数据区只能通过设置寄存器 FAP\_HL\_DIS 位被硬件自动擦除

(4) 闪存高级访问保护的功能是实现闪存访问及用户系统数据区误擦除保护

## 5.5.2 擦写保护

在 64K 及以下容量的闪存中，擦写保护的基本单位为 4 扇区。通过擦写保护可以防止程序在跑飞时闪存存储器的内容被意外更改。

在下面列出的情况下，擦写将不被允许，并会置位 EPPERR 位：

- 对被设置为擦写保护的扇区（主闪存以及闪存扩展区）做扇区擦除操作以及编程操作将不被允许
- 对存在任一扇区被设置为擦写保护的主闪存以及闪存扩展区做整片擦除将不被允许
- 闪存访问保护启动后，主闪存扇区 0~3 将被自动擦写保护，不允许做扇区擦除操作以及编程操作
- 闪存访问保护启动后，主存储器和主存扩展区在调试模式或是从非主闪存存储器启动下被自动擦写保护，不允许做扇区擦除操作以及编程操作

## 5.6 读取性能

提升系统时钟频率前须先按照闪存性能选择寄存器 (FLASH\_PSR) 的 WTCYC 位说明配置读取闪存须插入的延迟时间。

使能闪存性能选择寄存器（FLASH\_PSR）的 PFT\_EN 位、PFT\_EN2 位与 PFT\_LAT\_DIS 位可降低去闪存读取次数。

另提供一 HFCYC\_EN 位，使能该位后读取闪存时节省半个系统时钟，对连续读取大块常数表执行效率上有助益，但使能时有系统时钟频率上的限制，请参考 AT32F421 系列数据手册。

## 5.7 特殊功能

### 5.7.1 安全库区设定

设定以密码保护主存中指定范围的程式区，即安全库区，此区域仅能被执行，无法读取（I-Code, D-Code 总线除外），以及写入与删除，除非输入指定密码。安全库区划分为 指令区 与 唯读区。

#### 设定安全库区的益处：

以密码保护安全库区，方案商可刻录核心算法到此区域；

安全库区仅能执行，无法被读取，也无法删除（包含 ISP/IAP/SWD），除非输入方案商指定密码；

其余空白程序区可以提供给方案商客户进行二次开发；

方案商可以藉由安全库功能销售核心算法，不需要每个客户都开发完整方案；

设定安全库区，可防止蓄意破坏或更改终端产品应用程序代码。

#### 注意：

安全库区代码必须以扇区为单位进行烧录，并且起始地址与主存地址对齐；

仅允许 I-Code 总线读取指令库区；

允许 I-Code 总线以及 D-Code 总线读取唯读库区；

写入或擦除安全库区代码（指令区和唯读区），将在闪存状态寄存器（FLASH\_STS）的 EPPERR 位置'1'提出警告；

执行主存的整片擦除时，将不会擦除安全库区。

默认状态下，安全库区设定寄存器始终是不可读且被锁定的。要想对安全库区设定寄存器进行写操作，首先要对安全库区解锁，对闪存安全库区解锁寄存器（SLIB\_UNLOCK）写入 0xA35F6D24 值，通过查看闪存安全库区额外状态寄存器（SLIB\_MISC\_STS）的位 SLIB\_ULKF 确认解锁成功，随后对安全库区设定寄存器写入设定值。

启动安全库区的流程如下：

- 检查闪存状态寄存器（FLASH\_STS）的 OBF 位，以确认没有其他正在进行的编程操作；
- 对闪存安全库区解锁寄存器（SLIB\_UNLOCK）写入 0xA35F6D24，以进行安全库区解锁；
- 检查闪存安全库区额外状态寄存器（SLIB\_MISC\_STS）的 SLIB\_ULKF 位，以确认解锁成功；
- 如果安全库区设在主闪存内，需要在闪存安全库区地址设定寄存器（SLIB\_SET\_RANGE）设定要保护的扇区，包含指令区与唯读区的地址；如果安全库区设在主存扩展区域，需要设定主存扩展存储区域安全库区设定寄存器（EM\_SLIB\_SET）。
- 等待 OBF 位变为'0'；
- 在闪存安全库区密码设定寄存器（SLIB\_SET\_PWD）设定安全库区密码；
- 等待 OBF 位变为'0'；
- 烧录将存入安全库区的代码；
- 进行系统复位，重装载安全库区设定字；
- 读出闪存安全库区状态寄存器 0（SLIB\_STS0）/STS1 寄存器用于判断安全库区设定结果。

#### 注意：

不支持同时设定主闪存和主存扩展区域为安全库区；

启动安全库区的流程需要在闪存访问保护未启动时执行

解除安全库区的流程是：

- 在闪存安全库区密码清除寄存器（SLIB\_PWD\_CLR）写入先前设置的安全区域密码；
- 等待 OBF 位变为'0'；

- 进行系统复位，重装载安全库区设定字；
- 读出闪存安全库区状态寄存器0（SLIB\_STS0）用于判断安全库区解除结果。

**注意：**解除安全库区将会自动执行主存及主存扩展区的整片擦除，以及安全库设定块擦除。

## 5.7.2 启动程序代码区域作为主存扩展使用

用户只有一次机会将启动程序代码区域作为主存扩展使用。一旦设定成功，主存扩展区将具有主闪存特性。设定启动程序代码区域作为主存扩展使用的流程是：

- 用户读取闪存安全库区状态寄存器0（SLIB\_STS0）的位0，获知启动程序代码区域当前的模式
- 对闪存安全库区解锁寄存器（SLIB\_UNLOCK）写入0xA35F6D24，以进行启动程序代码区域模式设定解锁
- 写非0xFF到启动程序代码区模式设定寄存器（BTM\_MODE\_SET）的位7-0
- 等待OBF位变为'0'；
- 进行系统复位，重装载设定字；
- 读出闪存安全库区状态寄存器0（SLIB\_STS0）用于判断设定结果。

**注意：**启动设定主存扩展区的流程需要在闪存访问保护未启动时执行

## 5.7.3 CRC校验

以扇区为单位对安全库区代码或用户代码进行可选的CRC校验。

校验流程如下：

- 检查闪存状态寄存器（FLASH\_STS）的OBF位，以确认没有其他正在进行的编程操作；
- 在闪存CRC校验地址寄存器（FLASH\_CRC\_ADDR）设定要校验的代码起始地址；
- 在闪存CRC校验控制寄存器（FLASH\_CRC\_CTRL）位15-0，设定要校验的代码数量（单位是扇区）；
- 在闪存CRC校验控制寄存器（FLASH\_CRC\_CTRL）设置位16，启动CRC校验；
- 等待OBF位变为'0'；
- 读出闪存CRC校验结果寄存器（FLASH\_CRC\_CHK）用于判断CRC校验结果。

**注意：**

闪存CRC校验地址寄存器（FLASH\_CRC\_ADDR）设定值必须与扇区起始地址对齐；不允许跨主存及主存扩展区的CRC校验。

## 5.8 FLASH寄存器

必须用字（32位）的方式操作这些外设寄存器。

表 5-6 闪存接口寄存器映像和复位值

寄存器简称	基址偏移量	复位值
FLASH_PSR	0x00	0x0000 0030
FLASH_UNLOCK	0x04	0xFFFFFFFF
FLASH_USD_UNLOCK	0x08	0xFFFFFFFF
FLASH_STS	0x0C	0x0000 0000
FLASH_CTRL	0x10	0x0002 0080
FLASH_ADDR	0x14	0x0000 0000
FLASH_USD	0x1C	0x03FF FFFC
FLASH_EPPS	0x20	0xFFFF FFFF
SLIB_STS0	0x74	0x00FF 0000
SLIB_STS1	0x78	0xFFFF FFFF
SLIB_PWD_CLR	0x7C	0xFFFF FFFF
SLIB_MISC_STS	0x80	0x0000 0000

FLASH_CRC_ADDR	0x84	0x0000 0000
FLASH_CRC_CTRL	0x88	0x0000 0000
FLASH_CRC_CHK	0x8C	0x0000 0000
SLIB_SET_PWD	0x160	0x0000 0000
SLIB_SET_RANGE	0x164	0x0000 0000
EM_SLIB_SET	0x168	0x0000 0000
BTM_MODE_SET	0x16C	0x0000 0000
SLIB_UNLOCK	0x170	0x0000 0000

### 5.8.1 闪存性能选择寄存器 (FLASH\_PSR)

域	简称	复位值	类型	功能
位 31:9	保留	0x0000000	resd	保持为默认值
位 8	PFT_LAT_DIS	0	rw	预取时延无效 (Prefetch latency disable) 0: 闪存预取缓冲区时延开启, 访问缓冲区需要等待 1 个系统时钟周期; 1: 闪存预取缓冲区时延关闭, 访问缓冲区零等待。 推荐此位配置为 1, 后续请勿更改。
位 7	PFT_ENF2	0	ro	预取使能标志 2 (Prefetch enabled flag2) 该位置起时, 表示启动闪存预取缓冲区第二数据块
位 6	PFT_EN2	0	rw	预取使能 2 (Prefetch enable2) 0: 闪存预取缓冲区第二数据块关闭; 1: 闪存预取缓冲区第二数据块开启。 推荐此位配置为 1, 后续请勿修改。
位 5	PFT_ENF	1	ro	预取使能标志 (Prefetch enabled flag) 该位置起时, 表示启动闪存预取缓冲区
位 4	PFT_EN	1	rw	预取使能 (Prefetch enable) 0: 闪存预取缓冲区关闭; 1: 闪存预取缓冲区开启。
位 3	HFCYC_EN	0x0	rw	半周期加速访问使能 (Half cycle acceleration access enable) 0: 关闭; 1: 开启。
位 2:0	WTCYC	0x0	rw	等待周期 (Wait cycle) 需要根据系统时钟大小来设定闪存访问的等待周期, 以系统时钟为单位。 000: 零个等待周期, 0MHz < 系统时钟 ≤ 32MHz 使用; 001: 一个等待周期, 32MHz < 系统时钟 ≤ 64MHz 使用; 010: 两个等待周期, 64MHz < 系统时钟 ≤ 96MHz 使用; 011: 三个等待周期, 96MHz < 系统时钟 ≤ 120MHz 使用。

### 5.8.2 闪存解锁寄存器 (FLASH\_UNLOCK)

域	简称	复位值	类型	功能
位 31:0	UKVAL	0xFFFF XXXX wo		解锁键值 (Unlock key value) 该寄存器用于解锁主闪存及闪存扩展区。

注意：所有这些位是只写的，读出时返回 0。

### 5.8.3 闪存用户系统数据解锁寄存器 (FLASH\_USD\_UNLOCK)

域	简称	复位值	类型	功能
位 31:0	USD_UKVAL	0xFFFF XXXX wo		用户系统数据解锁键值 (User system data unlock key value)

注意：所有这些位是只写的，读出时返回 0。

## 5.8.4 闪存状态寄存器 (FLASH\_STS)

域	简称	复位值	类型	功能
位 31:6	保留	0x0000000	resd	保持为默认值
位 5	ODF	0	rw1c	操作完成标志 (Operation done flag) 当闪存操作 (编程/擦除) 成功完成时, 硬件会置起该位, 软件写'1'可以清除。
位 4	EPPERR	0	rw1c	擦写保护错误 (Erase/Program protection error) 当擦除或编程的闪存地址在擦写保护设定范围内时, 硬件会置起该位, 软件写'1'可以清除。
位 3	保留	0	resd	保持为默认值
位 2	PRGMERR	0	rw1c	编程错误 (Program error) 当编程的闪存地址的值为非擦除状态时, 硬件会置起该位, 软件写'1'可以清除。
位 1	保留	0	resd	保持为默认值
位 0	OBF	0	ro	操作忙标志 (Operation busy flag) 该位置起表示闪存操作正在进行, 该位清除表示操作结束。

## 5.8.5 闪存控制寄存器 (FLASH\_CTRL)

域	简称	复位值	类型	功能
位 31:18	保留	0x0000	resd	保持为默认值
位 17	LPMEN	1	rw	低功耗模式使能 (Low power mode enable) 0: 关闭; 1: 开启。 设置该位后, 一旦 MCU 进入深度睡眠模式, 闪存控制器也会让闪存同步进入低功耗模式。
位 16	FAP_HL_DIS	0x0	rw	闪存访问高级保护解除 (Flash access protection high level disable) 设置该位, 将触发硬件自动对用户系统数据区做擦除, 复位后将解除闪存访问高级保护, 维持低级访问保护。 该位置为 1 后, 硬件将自动清除此位。
位 15:13	保留	0x0	resd	保持为默认值
位 12	ODFIE	0	rw	操作完成中断使能 (Operation done flag interrupt enable) 0: 关闭; 1: 开启。
位 11,8,3	保留	0	resd	保持为默认值
位 10	ERRIE	0	rw	错误中断使能 (Error interrupt enable) 开启后 EPPERR 或 PRGMERR 都会产生中断。 0: 关闭; 1: 开启。
位 9	USDULKS	0	rw	用户系统数据解锁成功 (User system data unlock success) 一旦用户系统数据区解锁成功, 该位将被硬件自动置起, 表示允许对用户系统数据的编程/擦除操作。软件写'0'可以清除此位, 重新锁定用户系统数据区。
位 7	OPLK	1	rw	操作锁定 (Operation lock) 该位默认处于置起状态, 表示闪存锁定, 锁定时不允许操作, 解锁成功后, 硬件会自动清除此位, 表示允许闪存编程/擦除操作。软件写'1'可以重新锁定闪存操作。
位 6	ERSTR	0	rw	擦除开始 (Erasing start) 软件置起该位, 开始执行擦除操作。擦除完成后硬件自动清除该位。
位 5	USDERS	0	rw	用户系统数据擦除 (User system data erase) 用户系统数据区擦除。
位 4	USDPRGM	0	rw	用户系统数据编程 (User system data program) 用户系统数据编程。
位 2	BANKERS	0	rw	片擦除 (Bank erase)

				擦除片操作。
位 1	SECERS	0	rw	扇区擦除 (Sector erase) 擦除扇区操作。
位 0	FPRGM	0	rw	闪存编程 (Flash program) 编程操作。

## 5.8.6 闪存地址寄存器 (FLASH\_ADDR)

域	简称	复位值	类型	功能
位 31:0	FA	0x0000 0000	wo	闪存地址 (Flash address) 扇区擦除时选择对应的闪存扇区地址。

## 5.8.7 用户系统数据寄存器 (FLASH\_USD)

域	简称	复位值	类型	功能
位 31:27	保留	0x00	resd	保持为默认值
位 26	FAP_HL	0	ro	闪存访问保护高级 (Flash access protection high level) 闪存访问保护状态使用 {位 26, 位 1}联合判断。 00: 未启动访问保护, 且 FAP 值=0xA5 01: 启动低级访问保护, 且 FAP 值非 0xCC 以及 0xA5 10: 保留 11: 启动高级访问保护, 且 FAP 值=0xCC
位 25:18	USER_D1	0xFF	ro	用户数据 1
位 17:10	USER_D0	0xFF	ro	用户数据 0
位 9:2	SSB	0xFF	ro	系统配置字节 (System setting byte) 这里包含加载的用户系统数据区中的系统配置字节 位 9:7: 未用 位 6: nBOOT1 位 5: 未用 位 4: nSTDBY_RST 位 3: nDEPSLP_RST 位 2: nWDT_ATO_EN
位 1	FAP	0	ro	闪存访问保护 (Flash access protection)
位 0	USDERR	0	ro	用户系统数据错误 (User system data error) 该位置起表示用户系统数据中某字节和它的反码不匹配。 此时该字节和它的反码读出值将被硬件自动强制置为 0xFF

## 5.8.8 擦除编程保护状态寄存器 (FLASH\_EPPS)

域	简称	复位值	类型	功能
位 31:0	EPPS	0xFFFF FFFF	ro	擦除/编程保护状态 (Erase/Program protection status) 该寄存器反映的是加载的用户系统数据中的擦写保护字节 状态。

## 5.8.9 闪存安全库区状态寄存器0 (SLIB\_STS0)

专用于闪存安全库区。

域	简称	复位值	类型	功能
位 31:24	保留	0x00	resd	保持为默认值
位 23:16	EM_SLIB_INST_SS	0xFF	ro	主存扩展存储区安全库区指令起始扇区 (Extension memory sLib instruction start sector) 00000000: 扇区 0 00000001: 扇区 1 00000010: 扇区 2 00000011: 扇区 3 11111111: 无指令安全区 其余设定值: 无效

位 15:4	保留	0x000	resd	保持为默认值
位 3	SLIB_ENF	0	ro	sLib 使能标志 (sLib enabled flag) 该位置起时，表示闪存主存区域部分或是全部（依照 SLIB_STS1 设定）作为安全库代码。
位 2	EM_SLIB_ENF	0	ro	主存扩展存储区 sLib 使能标志 (Extension memory sLib enabled flag) 该位置起时，表示启动程序代码区域是作为主闪存扩展区域 (BTM_AP_ENF 置起)，并且存放的应用代码为安全库代码
位 1	保留	0	resd	保持为默认值
位 0	BTM_AP_ENF	0	ro	启动程序代码区域存放应用代码使能标志 (Boot memory store application code enabled flag) 该位置起时，表示启动程序代码区域可以作为主存扩展区域存放用户应用代码；否则仅用于存放系统启动代码

## 5.8.10 闪存安全库区状态寄存器1 (SLIB\_STS1)

专用于闪存安全库区。

域	简称	复位值	类型	功能
位 31:22	SLIB_ES	0x3FF	ro	主存安全库区结束扇区 (sLib end sector) 0000000000: 扇区 0 0000000001: 扇区 1 0000000010: 扇区 2 ... 0000001111: 扇区 15 (16KB 主闪存存储器的最后扇区) ... 0000011111: 扇区 31 (32KB 主闪存存储器的最后扇区) ... 0000111111: 扇区 63 (64KB 主闪存存储器的最后扇区)
位 21:11	SLIB_INST_SS	0x7FF	ro	主存安全库区指令区起始扇区 (sLib instruction start sector) 0000000000: 扇区 0 0000000001: 扇区 1 0000000010: 扇区 2 ... 00000001111: 扇区 15 (16KB 主闪存存储器的最后扇区) ... 00000011111: 扇区 31 (32KB 主闪存存储器的最后扇区) ... 00000111111: 扇区 63 (64KB 主闪存存储器的最后扇区) 11111111111: 无安全库区指令区
位 10:0	SLIB_SS	0x7FF	ro	主存安全库区起始扇区 (sLib start sector) 0000000000: 扇区 0 0000000001: 扇区 1 0000000010: 扇区 2 ... 00000001111: 扇区 15 (16KB 主闪存存储器的最后扇区) ... 00000011111: 扇区 31 (32KB 主闪存存储器的最后扇区) ... 00000111111: 扇区 63 (64KB 主闪存存储器的最后扇区)

### 5.8.11 闪存安全库区密码清除寄存器（SLIB\_PWD\_CLR）

专用于闪存安全库区。

域	简称	复位值	类型	功能
位 31:0	SLIB_PCLR_VAL	0x0000 0000	wo	安全库区密码清除（sLib password clear value） 用于写入正确的安全库区密码，将实现解除安全库区功能。 此寄存器写入状态将在 SLIB_MISC_STS 位 0 与位 1 中体现。

### 5.8.12 闪存安全库区额外状态寄存器（SLIB\_MISC\_STS）

专用于闪存安全库区。

域	简称	复位值	类型	功能
位 31:3	保留	0x0000 000	resd	保持为默认值
位 2	SLIB_ULKF	0	ro	SLib 解锁标志（sLib unlock flag） 当该位置起时表示 SLib 相关设定寄存器允许配置。
位 1	SLIB_PWD_OK	0	ro	密码正确（sLib password ok） 当密码正确，该位被硬件置起。
位 0	SLIB_PWD_ERR	0	ro	密码错误（sLib password error） 当密码错误，并且设定的密码清除寄存器的值不等于 0xFFFF FFFF，该位被硬件置起。 注意：当该位置起后，硬件将不再接受重新设定密码清除寄存器，直到再次复位。

### 5.8.13 闪存CRC校验地址寄存器（FLASH\_CRC\_ADDR）

专用于主闪存以及主存扩展区域。

域	简称	复位值	类型	功能
位 31:0	CRC_ADDR	0x0000 0000	wo	CRC 地址（CRC address） 选择要校验的闪存扇区起始地址。 注意：必须与扇区起始地址对齐

注意：所有这些位是只写的，读出无反应。

### 5.8.14 闪存CRC校验控制寄存器（FLASH\_CRC\_CTRL）

专用于主闪存以及主存扩展区域。

域	简称	复位值	类型	功能
位 31:17	保留	0x0000	resd	保持为默认值
位 16	CRC_STRT	0	wo	启动 CRC 校验（CRC start） 设置该位去启动用户代码或是安全库代码的 CRC 校验功能。 注意： 校验数据从 CRC_ADDR ~ CRC_ADDR+CRC_SN*1 扇区
位 15:0	CRC_SN	0x0000	wo	CRC 校验扇区数量（CRC sector number） 设定本次 CRC 校验的数据量，单位是扇区

### 5.8.15 闪存CRC校验结果寄存器（FLASH\_CRC\_CHK）

专用于主闪存以及主存扩展区域。

域	简称	复位值	类型	功能
位 31:0	CRC_CHK	0x0000 0000	ro	CRC 校验结果（CRC check result）

注意：所有这些位是只读的，写入无反应。

### 5.8.16 闪存安全库区密码设定寄存器（SLIB\_SET\_PWD）

专用于闪存安全库区密码设定。

域	简称	复位值	类型	功能
位 31:0	SLIB_PSET_VAL	0x0000 0000	wo	安全库区密码 (sLib password setting value) 注意：在解除安全库区锁定后，此寄存器才允许被写入，用于设定安全库区启动密码。但写入 0xFFFF_FFFF 以及 0x0000_0000 值无效。

注意：所有这些位是只写入，读出为 0。

### 5.8.17 闪存安全库区地址设定寄存器（SLIB\_SET\_RANGE）

专用于主存安全库区地址设定。

域	简称	复位值	类型	功能
位 31:22	SLIB_ES_SET	0x000	wo	主存安全库区结束扇区设定 (sLib end sector setting) 用于设定启动安全库区时的安全库区结束扇区位置 0000000000: 扇区 0 0000000001: 扇区 1 0000000010: 扇区 2 ... 0000001111: 扇区 15 (16KB 主闪存存储器的最后扇区) ... 0000011111: 扇区 31 (32KB 主闪存存储器的最后扇区) ... 0000111111: 扇区 63 (64KB 主闪存存储器的最后扇区)
位 21:11	SLIB_ISS_SET	0x000	wo	主存安全库区指令区起始扇区设定 (sLib instruction start sector setting) 用于设定启动安全库区时的指令区起始扇区位置 0000000000: 扇区 0 0000000001: 扇区 1 0000000010: 扇区 2 ... 0000001111: 扇区 15 (16KB 主闪存存储器的最后扇区) ... 0000001111: 扇区 31 (32KB 主闪存存储器的最后扇区) ... 0000011111: 扇区 63 (64KB 主闪存存储器的最后扇区) 1111111111: 无安全库区指令区
位 10:0	SLIB_SS_SET	0x000	wo	主存安全库区起始扇区设定 (sLib start sector setting) 用于设定启动安全库区时的安全库区起始扇区位置 0000000000: 扇区 0 0000000001: 扇区 1 0000000010: 扇区 2 ... 0000000111: 扇区 15 (16KB 主闪存存储器的最后扇区) ... 0000001111: 扇区 31 (32KB 主闪存存储器的最后扇区) ... 0000011111: 扇区 63 (64KB 主闪存存储器的最后扇区)

注意：

所有这些位是只写入，读出为 0。

在解除安全库区锁定后，此寄存器才允许被写入。  
超过主闪存存储地址范围均是无效设定。

### 5.8.18 主存扩展存储区域安全库区设定寄存器（EM\_SLIB\_SET）

专用于主存扩展区域。

域	简称	复位值	类型	功能
位 31:24	保留	0x00	resd	保持为默认值
位 23:16	EM_SLIB_ISS_SET	0x000	wo	主存扩展区安全库区指令区起始扇区设定（Extension memory sLib instruction start sector setting） 用于设定启动安全库区时的指令区起始扇区位置 00000000: 扇区 0 00000001: 扇区 1 00000010: 扇区 2 00000011: 扇区 3 11111111: 无指令安全区 其余设定值：无效 注意： 当设为 0xFF，表示主存扩展区从扇区 0 至扇区 3 都为安全库区，且整个安全库区作为唯读取； 主存扩展存储区 sLib 设定（Extension memory sLib setting）
位 15:0	EM_SLIB_SET	0x000	wo	写入 0x5AA5 将启动主存扩展区作为存放安全库区代码功能

注意：所有这些位是只写的，读出无反应。

### 5.8.19 启动程序代码区模式设定寄存器（BTM\_MODE\_SET）

专用于启动程序代码区域。

域	简称	复位值	类型	功能
位 31:8	保留	0x0000000	resd	保持为默认值
位 7:0	BTM_MODE_SET	0x00	wo	启动程序代码区模式设定（Boot memory mode setting） 0xFF：启动程序代码区域作为系统区域，存放系统启动代码功能 其他值：启动程序代码区域作为主存扩展区域，存放应用代码功能 注意：此寄存器的设定需要在闪存访问保护未启动下进行

注意：所有这些位是只写的，读出无反应。

### 5.8.20 闪存安全库区解锁寄存器（SLIB\_UNLOCK）

专用于安全库区寄存器的解锁设定。

域	简称	复位值	类型	功能
位 31:0	SLIB_UKVAL	0x0000 0000	wo	安全库区解锁键值（sLib unlock key value） 固定键值 0xA35F_6D24，用于安全库区设定寄存器的解锁。

注意：所有这些位是只写入，读出为 0。

# 6 通用和复用功能 I/O (GPIO 和 IOMUX)

## 6.1 简介

AT32F421 支持多达 39 个双向 I/O 引脚，分别为 PA0-PA15、PB0-PB15、PC13-PC15、PF0-PF1 与 PF6-PF7，每个引脚都可以实现与外部的通讯、控制以及数据采集的功能。

每个引脚都支持通用功能输入输出 (GPIO) 或复用功能输入输出 (IOMUX)。

每个引脚都可以软件配置成浮空输入、上拉/下拉输入、模拟输入/输出、通用推挽/开漏输出、复用推挽/开漏输出。

每个引脚都有独立的弱上拉/下拉功能。

每个引脚都可以软件配置输出驱动能力。

每个引脚都可以配置为外部中断输入。

每个引脚都支持配置锁定功能。

## 6.2 功能描述

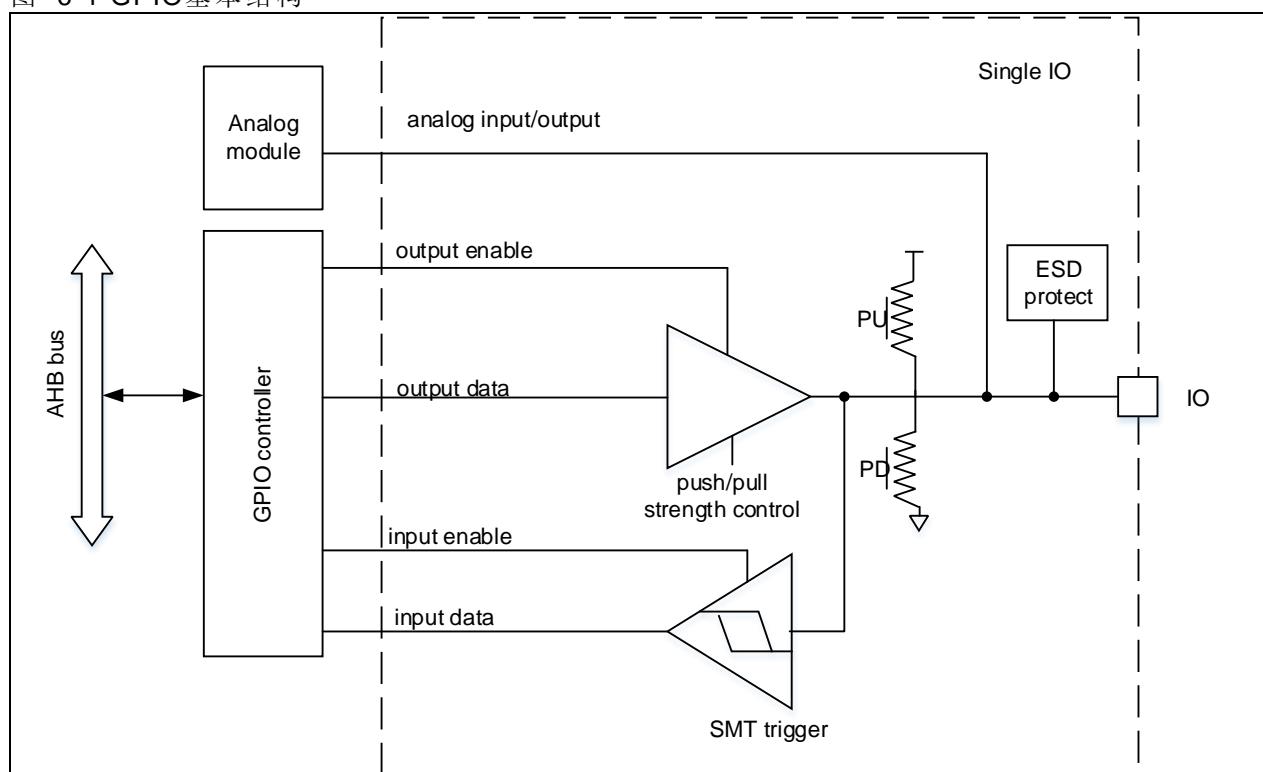
### 6.2.1 GPIO结构

每个引脚可以由软件配置成四种输入模式（输入浮空、输入上拉、输入下拉、模拟输入）和四种输出模式（开漏输出、推挽式输出、推挽式复用、开漏复用）。

每个 I/O 端口对应的寄存器允许字、半字或字节访问，每个 I/O 端口位可以自由编程。

下图给出了一个 I/O 端口位的基本结构。

图 6-1 GPIO 基本结构



### 6.2.2 GPIO复位状态

系统上电或复位后，所有引脚除了 SWD 相关引脚以外，都被配置为浮空输入模式。SWD 相关引脚则配置为：PA13/ SWDIO 为复用功能上拉模式，PA14/SWCLK 为复用功能下拉模式。

### 6.2.3 通用功能输入配置

配置模式	IOMC	PUPD
通用浮空输入		00
通用下拉输入	00	10
通用上拉输入		01

当引脚配置为输入时：

- 引脚状态可通过对输入数据寄存器的读访问得到
- 可配置引脚为浮空输入、上拉输入或下拉输入
- 施密特触发器有效
- 不能对该引脚进行输出。

注意：如果是浮空输入模式，为避免复杂环境下，没有使用的引脚有干扰，导致漏电，建议，如引脚不使用，则配置为模拟输入模式。

### 6.2.4 模拟输入/输出配置

配置模式	IOMC	PUPD
模拟输入输出	11	不使用

当 GPIO 端口被配置为模拟输入配置时：

- 施密特触发无效
- 不能对该引脚进行数字输入输出
- 对应的引脚，不能设置上拉/下拉电阻。

### 6.2.5 通用功能输出配置

配置模式	IOMC	OM	HDRV	ODRV[1: 0]	PUPD
通用推挽无上拉/下拉	01	0		000: 输出模式，适中电流推动/吸入能力 001: 输出模式，较大电流推动/吸入能力 010: 输出模式，适中电流推动/吸入能力 011: 输出模式，适中电流推动/吸入能力 1xx: 输出模式，极大电流推动/吸入能力	任意值
通用开漏无上拉/下拉	01	1		000: 输出模式，适中电流推动/吸入能力 001: 输出模式，较大电流推动/吸入能力 010: 输出模式，适中电流推动/吸入能力 011: 输出模式，适中电流推动/吸入能力 1xx: 输出模式，极大电流推动/吸入能力	任意值

当 GPIO 端口被配置为输出时：

- 施密特触发器有效
  - 可通过输出寄存器让对应引脚输出
  - 上拉和下拉电阻不能被使用
  - 在开漏模式时，可强输出 0，可用外部上拉电阻输出 1。
  - 在推挽模式时，可通过输出寄存器输出数字 0/1。
  - 提供 GPIO 设置/清除寄存器控制对应的 GPIO 数据输出寄存器的设置/清除
- 注意：GPIO 设置/清除寄存器 对应同一个引脚的 IOCB/IOSB 同时写 1，IOSB 优先级高于 IOCB

## 6.2.6 GPIO端口保护

为了防止误操作导致 GPIO 功能混乱，提供每个对应引脚的锁定机制。一旦锁定，在下次复位或者上电之前都不能进行对应引脚的 GPIO 配置。

## 6.2.7 IOMUX功能结构

大多数引脚支持多个外设的输出功能映射，通过 IOMUX 功能输入/输出章节查找表来选择每个引脚对应的外设输入输出功能。通过引脚所对应的 GPIO 复用低位寄存器 (GPIOx\_MUXL) (从引脚 0 到引脚 7) 或 GPIO 复用高位寄存器 (GPIOx\_MUXH) (从引脚 8 到引脚 15) 进行对应的设置，单一引脚有多达 16 种不同的 IOMUX 映射方案，方便灵活选用。

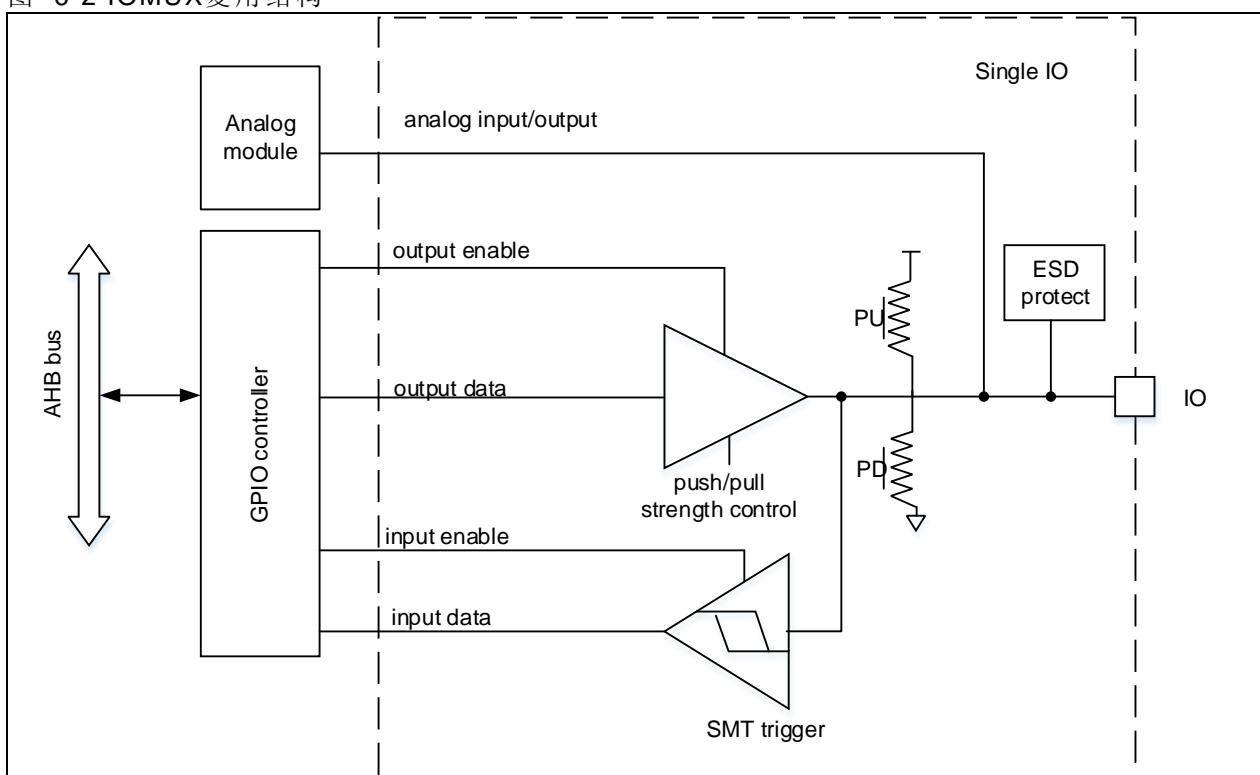
每个引脚通过设定 GPIO 复用低位寄存器 (GPIOx\_MUXL) 或 GPIO 复用高位寄存器 (GPIOx\_MUXH)，只会和单一外设的单一脚进行对应，不存在单一引脚多个外设抢占的冲突。

引脚作为复用输入功能时，与通用输入功能一样，端口配置成输入模式（浮空、上拉、下拉）。

要实现复用输出功能，必须配置 GPIO 配置寄存器 (GPIOx\_CFGR) 或 GPIO 输出模式寄存器 (GPIOx\_OMODE) 将该端口设定为复用功能输出模式以及推挽或开漏模式。此时引脚和 GPIO 控制器断开，由 IOMUX 控制器进行控制。

要实现双向复用功能，与复用输出功能一样，将该端口设定为复用功能输出模式（推挽或开漏）即可。由 IOMUX 控制器进行控制。

图 6-2 IOMUX 复用结构



## 6.2.8 复用功能上下拉配置

配置模式	IOMC	PUPD
复用浮空		00
复用下拉	10	10
复用上拉		01

当引脚配置为输入时：

- 引脚状态可通过对输入数据寄存器的读访问得到
- 可配置引脚为浮空输入、上拉输入或下拉输入
- 施密特触发器有效

- 不能对该引脚进行 GPIO 输出。

### 6.2.9 IOMUX功能输入/输出

选择每个端口线的有效复用功能是通过 GPIO 复用低位寄存器 (GPIOx\_MUXL) (从引脚 0 到引脚 7) 或 GPIO 复用高位寄存器 (GPIOx\_MUXH) (从引脚 8 到引脚 15) 进行设置。

表 6-1 通过 GPIOA\_MUX\* 寄存器配置端口 A 的复用功能

引脚名	MUX0	MUX1	MUX2	MUX3	MUX4	MUX5	MUX6	MUX7
PA0		USART2_CTS			I2C2_SCL	TMR1_ETR		COMP_OUT
PA1	EVENTOUT	USART2 RTS			I2C2_SDA	TMR15_CH1N		
PA2	TMR15_CH1	USART2_TX						
PA3	TMR15_CH2	USART2_RX				I2S2_MCK		
PA4	SPI1_CS/I2S1_WS	USART2_CK			TMR14_CH1			
PA5	SPI1_SCK/I2S1_CK							
PA6	SPI1_MISO/I2S1_MCK	TMR3_CH1	TMR1_BKIN	I2S2_MCK		TMR16_CH1	EVENTOUT	COMP_OUT
PA7	SPI1_MOSI/I2S1_SD	TMR3_CH2	TMR1_CH1N		TMR14_CH1	TMR17_CH1	EVENTOUT	
PA8	CLKOUT	USART1_CK	TMR1_CH1	EVENTOUT	USART2_TX			I2C2_SCL
PA9	TMR15_BKIN	USART1_TX	TMR1_CH2		I2C1_SCL	CLKOUT		I2C2_SMBA
PA10	TMR17_BKIN	USART1_RX	TMR1_CH3		I2C1_SDA			
PA11	EVENTOUT	USART1_CTS	TMR1_CH4		I2C1_SMBA	I2C2_SCL		COMP_OUT
PA12	EVENTOUT	USART1_RTS	TMR1_ETR			I2C2_SDA		
PA13	SWDIO	IR_OUT					SPI2_MISO/I2S2_MCK	
PA14	SWCLK	USART2_TX					SPI2_MOSI/I2S2_SD	
PA15	SPI1_CS/I2S1_WS	USART2_RX		EVENTOUT			SPI2_CS/I2S2_WS	

表 6-2 通过 GPIOB\_MUX\* 寄存器配置端口 B 的复用功能

引脚名	MUX0	MUX1	MUX2	MUX3	MUX4	MUX5	MUX6	MUX7
PB0	EVENTOUT	TMR3_CH3	TMR1_CH2N	USART2_RX			I2S1_MCK	
PB1	TMR14_CH1	TMR3_CH4	TMR1_CH3N				SPI2_SCK/I2S2_CK	
PB2			TMR3_ETR					
PB3	SPI1_SCK/I2S1_CK	EVENTOUT					SPI2_SCK/I2S2_CK	
PB4	SPI1_MISO/I2S1_MCK	TMR3_CH1	EVENTOUT			TMR17_BKIN	SPI2_MISO/I2S2_MCK	I2C2_SDA
PB5	SPI1_MOSI/I2S1_SD	TMR3_CH2	TMR16_BKIN	I2C1_SMBA			SPI2_MOSI/I2S2_SD	
PB6	USART1_TX	I2C1_SCL	TMR16_CH1N				I2S1_MCK	
PB7	USART1_RX	I2C1_SDA	TMR17_CH1N					

PB8		I2C1_SCL	TMR16_CH1					
PB9	IR_OUT	I2C1_SDA	TMR17_CH1	EVENTOUT		I2S1_MC_K		SPI2_CS/I2S2_WS
PB10		I2C2_SCL						SPI2_SCK/I2S2_CK
PB11	EVENTOUT	I2C2_SDA						
PB12	SPI2_CS/I2S2_W_S	EVENTOUT	TMR1_BKIN			TMR15_B_KIN		I2C2_SMBA
PB13	SPI2_SCK/I2S2_CK	-	TMR1_CH1N			I2C2_SCL		
PB14	SPI2_MISO/I2S2_MCK	TMR15_CH1	TMR1_CH2N			I2C2_SDA		
PB15	SPI2_MOSI/I2S2_SD	TMR15_CH2	TMR1_CH3N	TMR15_CH1_N				

表 6-3 通过GPIOF\_MUX\*寄存器配置端口F的复用功能

引脚名	MUX0	MUX1	MUX2	MUX3	MUX4	MUX5	MUX6	MUX7
PF0		I2C1_SDA						
PF1		I2C1_SCL						
PF6	I2C2_SCL							
PF7	I2C2_SDA							

注意：EVENTOUT 是 Cortex-M 的 TXEV 信号

## 6.2.10 外设复用功能引脚配置

当外设需要使用 IOMUX 复用功能时：

- 如果外设引脚需要作为复用输出则对应的引脚配置成复用推挽/开漏输出
- 如果外设引脚需要作为复用输入则对应的引脚配置成复用模式（浮空/上拉/下拉）
- ADC 外设需要将模拟通道对应的引脚配置为模拟输入/输出模式
- I2C 外设需要对应引脚作为双向复用功能时，需把对应的引脚配置复用开漏模式

## 6.2.11 IOMUX映射优先级

除了极个引脚可能会被硬件直接抢占，其他外设都可通过配置 GPIO 复用低位寄存器 (GPIOx\_MUXL) / GPIO 复用高位寄存器 (GPIOx\_MUXH) 得到唯一外设复用。

某些引脚不管 GPIO 配置为任何模式，都会被特定的硬件功能直接占用。

表 6-4 硬件抢占功能

引脚名字	抢占使能位	说明
PA0	PWC_CTRLSTS[8] =1	抢占使能位有效之后，PA0 引脚直接作为 PWC 的 WKUP1 功能使用
PB5	PWC_CTRLSTS[13] =1	抢占使能位有效之后，PB5 引脚直接作为 PWC 的 WKUP6 功能使用
PB15	PWC_CTRLSTS[14] =1	抢占使能位有效之后，PB15 引脚直接作为 PWC 的 WKUP7 功能使用
PC13	PWC_CTRLSTS[9] = 1 (ERTC_CTRL[23: 21] != 3'b000)   (ERTC_CTRL[11] != 0)   (ERTC_TAMP[0] != 0)	抢占使能位有效之后，PC13 引脚直接作为 RTC 通道输入输出使用
PC14	CRM_BPDC[0]=1	抢占使能位有效之后，PC14 作为 LEXT 通道使用
PC15	CRM_BPDC[0]=1	抢占使能位有效之后，PC15 作为 LEXT 通道使用
PF0	CRM_CTRL[16]=1	抢占使能位有效之后，PF0 作为 HEXT 通道使用
PF1	CRM_CTRL[16]=1	抢占使能位有效之后，PF1 作为 HEXT 通道使用

## 6.2.12 外部中断/唤醒线

每个引脚都支持作为外部中断的输入，对应的引脚须配置为输入模式。

## 6.3 GPIO寄存器

下面列出了 GPIO 寄存器映像和复位数值。

可以用字节（8位）、半字（16位）或字（32位）的方式操作这些外设寄存器。

表 6-5 GPIO 寄存器地址映像和复位值

寄存器简称	基址偏移量	复位值
GPIOA_CFGR	0x00	0x2800 0000
GPIOx_CFGR(x = B,C,F)	0x00	0x0000 0000
GPIOx_OMODE	0x04	0x0000 0000
GPIOA_ODRVR	0x08	0x0C00 0000
GPIOx_ODRVR(x = B,C,F)	0x08	0x0000 0000
GPIOA_PULL	0x0C	0x2400 0000
GPIOx_PULL(x = B,C,F)	0x0C	0x0000 0000
GPIOx_IDT	0x10	0x0000 XXXX
GPIOx_ODT	0x14	0x0000 0000
GPIOx_SCR	0x18	0x0000 0000
GPIOx_WPR	0x1C	0x0000 0000
GPIOx_MUXL	0x20	0x0000 0000
GPIOx_MUXH	0x24	0x0000 0000
GPIOx_CLR	0x28	0x0000 0000
GPIOx_HDRV	0x3C	0x0000 0000

### 6.3.1 GPIO配置寄存器 (GPIOx\_CFGR) (x=A..H)

偏移地址: 0x00

复位值: 0x28000000 端口 A

0x00000000 其它端口

域	简称	复位值	类型	功能
位 2y+1: 2y	IOMCy	0x2800 0000	rw	GPIOx 模式配置 (y=0~15) (GPIOx mode configurate) 用于配置 GPIOx 的工作模式: 00: 输入 (复位后的模式) 01: 通用输出 10: 复用功能 11: 模拟

### 6.3.2 GPIO输出模式寄存器 (GPIOx\_OMODE) (x=A..H)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0000	resd	始终读为 0。
位 15: 0	OM	0x0000	rw	GPIOx 的输出模式配置 (y=0..15) (GPIOx output mode configurate) 当 GPIOx 用作输出时, 可选择以下两种输出模式: 0: 推挽 (复位状态) 1: 开漏

### 6.3.3 GPIO电流推动/吸入能力切换控制寄存器（GPIOx\_ODRVR） (x=A..H)

偏移地址: 0x08

复位值: 0x0C00 0000 端口 A

0x00000000 其它端口

域	简称	复位值	类型	功能
位 2y+1: 2y	ODRVy	0x0000 0000	rw	GPIOx 的驱动能力配置 (y=0...15) (GPIOx drive capability) 用于配置相应的 I/O 端口电流能力: x0: 适中电流推动/吸入能力 01: 较大电流推动/吸入能力 11: 适中电流推动/吸入能力

### 6.3.4 GPIO上/下拉寄存器（GPIOx\_PULL）(x=A..H)

偏移地址: 0x0C

复位值: 0x2400 0000 端口 A

0x00000000 其它端口

域	简称	复位值	类型	功能
位 2y+1: 2y	PULLy	0x2400 0000	rw	GPIOx 的上下拉配置 (y=0...15) (GPIOx pull configurate) 用于配置相应的 I/O 上拉或下拉。 00: 无作用 01: 上拉 10: 下拉

### 6.3.5 GPIO输入数据寄存器（GPIOx\_IDT）(x=A..H)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0000	resd	始终读为 0。
位 15: 0	IDT	0xFFFF	ro	GPIOx 输入的数据 (GPIOx input data) GPIOx 对应 IO 口的输入电平状态，每一位对应 GPIOx 的一个 IO。

### 6.3.6 GPIO输出数据寄存器（GPIOx\_ODT）(x=A..H)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0000	resd	始终读为 0。
位 15: 0	ODT	0x0000	rw	GPIOx 输出的数据 (IO output data)。 每一位对应 GPIOx 的一个 IO。 GPIOx 对应 IO 口的输出电平状态。 0: 低电平； 1: 高电平。

### 6.3.7 GPIO设置/清除寄存器（GPIOx\_SCR）(x=A..H)

域	简称	复位值	类型	功能
位 31: 16	IOCB	0x0000	wo	清除 GPIOx 位 (GPIOx clear bit) 写'1'的位其对应 ODT 寄存器位会清除，写'0'的位其对应 ODT 寄存器位维持不变，相当于 ODT 寄存器的位操作。

				0: 对应位不变; 1: 对应位清除。
位 15: 0	IOSB	0x0000	wo	设置 GPIOx 位 (GPIOx set bit) 写'1'的位其对应 ODT 寄存器位会置起, 写'0'的位其对应 ODT 寄存器位维持不变, 相当于 ODT 寄存器的位操作。 如果 IOCB 和 IOSB 同一个位都写'1', 那么优先级更高的 IOSB 会生效。
				0: 对应位不变; 1: 对应位置起。

### 6.3.8 GPIO写保护寄存器 (GPIOx\_WPR) (x=A..H)

域	简称	复位值	类型	功能
位 31: 17	保留	0x0000	resd	保持为默认值。
位 16	WPSEQ	0x0	rw	写保护使能序列 (Write protect sequence) 想保护某些 IO 位不被写入, 需配合同时操作写保护使能序列位和 WPEN 位。 写保护使能位操作按照以下方式操作 4 次, 写'1' ->写'0' ->写'1' ->读, 操作期间 WPEN 位值不可修改。
位 15: 0	WPEN	0x0000	rw	写保护使能 (Write protect enable) 每一位对应 GPIOx 的一个 IO。 0: 无写保护; 1: 写保护。

### 6.3.9 GPIO复用低位寄存器 (GPIOx\_MUXL) (x=A..H)

偏移地址: 0x20

复位值: 0x00000000

域	简称	复位值	类型	功能
位 4y+3: 4y	MUXLy	0x0	rw	GPIOx 引脚 y 的复用功能选择 (y=0...7) (GPIOx pin y muxing) 用于配置对应 IO 口的复用功能。 0000: MUX0 0001: MUX1 0010: MUX2 0011: MUX3 0100: MUX4 0101: MUX5 0110: MUX6 0111: MUX7 1xxx: 保留

### 6.3.10 GPIO复用高位寄存器 (GPIOx\_MUXH) (x=A..H)

域	简称	复位值	类型	功能
位 4y+3: 4y	MUXHy	0x0	rw	端口 x 引脚 y 的复用功能选择 (y=8...15) (GPIOx pin y muxing) 用于配置对应 IO 口的复用功能。 0000: MUX0 0001: MUX1 0010: MUX2 0011: MUX3 0100: MUX4 0101: MUX5 0110: MUX6 0111: MUX7

---

1xxx: 保留

---

### 6.3.11 GPIO位清除寄存器（GPIOx\_CLR）（x=A..H）

域	简称	复位值	类型	功能
位 31: 16	保留	0x0000	resd	保持为默认值。
位 15: 0	IOCB	0x0000	wo	清除 GPIOx 的位（GPIOx clear bit） 写'1'的位其对应 ODT 寄存器位会清除，写'0'的位其对应 ODT 寄存器位维持不变，相当于 ODT 寄存器的位操作。 0: 对应位不变； 1: 对应位清除。

### 6.3.12 极大电流推动/吸入能力切换控制寄存器（GPIOx\_HDRV）（x=A..H）

域	简称	复位值	类型	功能
位 31: 16	保留	0x0000	resd	保持为默认值。
位 15: 0	HDRV	0x0000	rw	极大电流推动/吸入能力切换控制寄存器 0: 无效 1: GPIO 切换为极大电流推动/吸入能力

## 7 系统配置控制器 (SCFG)

### 7.1 简介

该器件具有一组配置寄存器。系统配置控制器的主要用途如下：

- 重映射部分 DMA 触发源到其它不同 DMA 通道上
- 管理连接到 GPIO 口的外部中断

### 7.2 SCFG寄存器

下面列出了 SCFG 寄存器映像和复位数值。

必须以字（32 位）的方式操作这些外设寄存器。

表 7-1 SCFG 寄存器地址映像和复位值

寄存器简称	基址偏移量	复位值
SCFG_CFG1	0x00	0x0000 000X
SCFG_EXINTC1	0x08	0x0000 0000
SCFG_EXINTC2	0x0C	0x0000 0000
SCFG_EXINTC3	0x10	0x0000 0000
SCFG_EXINTC4	0x14	0x0000 0000

#### 7.2.1 SCFG配置寄存器1 (SCFG\_CFG1)

域	简称	复位值	类型	功能
位 31:13	保留	0x00000	resd	保持为默认值
位 12	TMR17_DMA_RMP	0x0	rw	TMR17 DMA 请求重映射位 由软件设置和清除该位。 0x0: 无重映射 (TMR17_CH1 与 TMR17_OVERFLOW 的 DMA 请求映射在 DMA 通道 1 上) 0x1: 重映射 1 (TMR17_CH1 与 TMR17_OVERFLOW 的 DMA 请求映射在 DMA 通道 2 上)
位 11	TMR16_DMA_RMP	0x0	rw	TMR16 DMA 请求重映射位 由软件设置和清除该位。 0x0: 无重映射 (TMR16_CH1 与 TMR16_OVERFLOW 的 DMA 请求映射在 DMA 通道 3 上) 0x1: 重映射 1 (TMR16_CH1 与 TMR16_OVERFLOW 的 DMA 请求映射在 DMA 通道 4 上)
位 10	USART1_RX_DMA_RMP	0x0	rw	USART1 RX DMA 请求重映射位 由软件设置和清除该位。它控制着 USART1 RX DMA 通道请求的重映射。 0: 无重映射 (USART1_RX 的 DMA 请求映射在 DMA 通道 3 上) 1: 重映射 (USART1_RX 的 DMA 请求映射在 DMA 通道 5 上)
位 9	USART1_TX_DMA_RMP	0x0	rw	USART1 TX DMA 请求重映射位 由软件设置和清除该位。它控制着 USART1 TX DMA 通道请求的重映射。 0: 无重映射 (USART1_TX 的 DMA 请求映射在 DMA 通道 2 上) 1: 重映射 (USART1_TX 的 DMA 请求映射在 DMA 通道 4 上)
位 8	ADC_DMA_RMP	0x0	rw	ADC DMA 请求重映射位 由软件设置和清除该位。它控制着 ADC DMA 通道请求的重映射。 0: 无重映射 (ADC 的 DMA 请求映射在 DMA 通道 1 上) 1: 重映射 (ADC 的 DMA 请求映射在 DMA 通道 2 上)

				红外调制包络信号源选择 用于选择红外调制包络信号源： 00: TMR16 01: USART1 10: USART2 11: 保留
位 7:6	IR_SRC_SEL	0x0	rw	
位 5	IR_POL	0x0	rw	红外输出极性选择 0: 红外线发射输出 (IR_OUT) 不反向 1: 红外线发射输出 (IR_OUT) 反向
位 4	PA11_12_RMP	0x0	rw	小封装(20 脚)上 PA11 和 PA12 的重映射 由软件设置与清除该位。它控制着小封装上 PA9/10 或 PA11/12 脚的映射 0: 没有重映射(PA9/PA10 脚对应 PA9/PA10 脚) 1: 重映射 (PA11/PA12 脚映射到 PA9/PA10 脚上)
位 3:2	保留	0xX	resd	保持为默认值
位 1:0	MEM_MAP_SEL	0xX	ro	启动模式状态位 此位仅供读取，显示复位后的启动区域。 X0: 从主存存储器启动 01: 从启动程序存储器启动 11: 从内置 SRAM 启动

## 7.2.2 SCFG外部中断配置寄存器1 (SCFG\_EXINTC1)

域	简称	复位值	类型	功能
位 31:16	保留	0x0000	resd	保持为默认值
位 15:12	EXINT3	0x0	rw	EXINT3 配置 (EXINT3 configuration) 这些位可由软件读写，用于选择 EXINT3 外部中断的输入源。 0000: PA[3]引脚 0001: PB[3]引脚
位 11:8	EXINT2	0x0	rw	EXINT2 配置 (EXINT2 configuration) 这些位可由软件读写，用于选择 EXINT2 外部中断的输入源。 0000: PA[2]引脚 0001: PB[2]引脚
位 7:4	EXINT1	0x0	rw	EXINT1 配置 (EXINT1 configuration) 这些位可由软件读写，用于选择 EXINT1 外部中断的输入源。 0000: PA[1]引脚 0001: PB[1]引脚 0101: PF[1]引脚
位 3:0	EXINT0	0x0	rw	EXINT0 配置 (EXINT0 configuration) 这些位可由软件读写，用于选择 EXINT0 外部中断的输入源。 0000: PA[0]引脚 0001: PB[0]引脚 0101: PF[0]引脚

### 7.2.3 SCFG外部中断配置寄存器2 (SCFG\_EXINTC2)

域	简称	复位值	类型	功能
位 31:16	保留	0x0000	resd	保持为默认值
位 15:12	EXINT7	0x0	rw	EXINT7 配置 (EXINT7 configuration) 这些位可由软件读写, 用于选择 EXINT7 外部中断的输入源。 0000: PA[7]引脚 0001: PB[7]引脚 0101: PF[7]引脚
位 11:8	EXINT6	0x0	rw	EXINT6 配置 (EXINT6 configuration) 这些位可由软件读写, 用于选择 EXINT6 外部中断的输入源。 0000: PA[6]引脚 0001: PB[6]引脚 0101: PF[6]引脚
位 7:4	EXINT5	0x0	rw	EXINT5 配置 (EXINT5 configuration) 这些位可由软件读写, 用于选择 EXINT5 外部中断的输入源。 0000: PA[5]引脚 0001: PB[5]引脚
位 3:0	EXINT4	0x0	rw	EXINT4 配置 (EXINT4 configuration) 这些位可由软件读写, 用于选择 EXINT4 外部中断的输入源。 0000: PA[4]引脚 0001: PB[4]引脚

### 7.2.4 SCFG外部中断配置寄存器3 (SCFG\_EXINTC3)

域	简称	复位值	类型	功能
位 31:16	保留	0x0000	resd	保持为默认值
位 15:12	EXINT11	0x0	rw	EXINT11 配置 (EXINT11 configuration) 这些位可由软件读写, 用于选择 EXINT11 外部中断的输入源。 0000: PA[11]引脚 0001: PB[11]引脚
位 11:8	EXINT10	0x0	rw	EXINT10 配置 (EXINT10 configuration) 这些位可由软件读写, 用于选择 EXINT10 外部中断的输入源。 0000: PA[10]引脚 0001: PB[10]引脚
位 7:4	EXINT9	0x0	rw	EXINT9 配置 (EXINT9 configuration) 这些位可由软件读写, 用于选择 EXINT9 外部中断的输入源。 0000: PA[9]引脚 0001: PB[9]引脚
位 3:0	EXINT8	0x0	rw	EXINT8 配置 (EXINT8 configuration) 这些位可由软件读写, 用于选择 EXINT8 外部中断的输入源。 0000: PA[8]引脚 0001: PB[8]引脚

## 7.2.5 SCFG外部中断配置寄存器4 (SCFG\_EXINTC4)

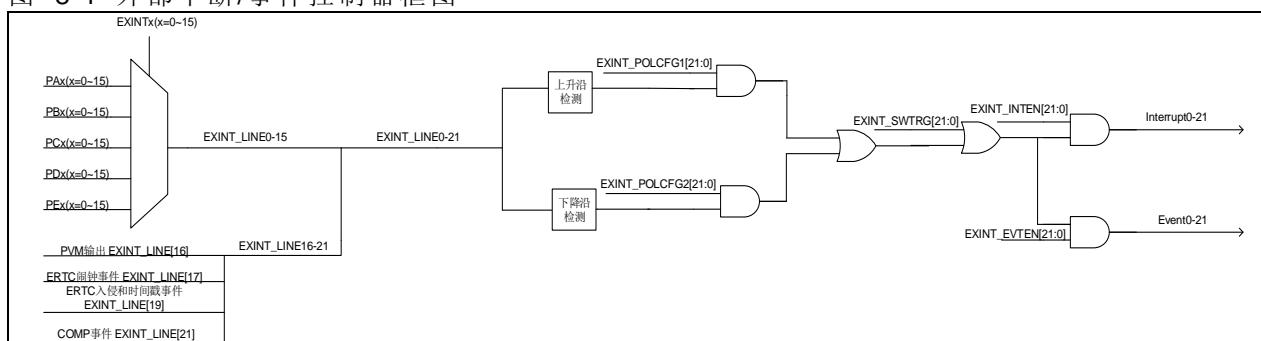
域	简称	复位值	类型	功能
位 31:16	保留	0x0000	resd	保持为默认值
位 15:12	EXINT15	0x0	rw	EXINT15 配置 (EXINT15 configuration) 这些位可由软件读写, 用于选择 EXINT15 外部中断的输入源。 0000: PA[15]引脚 0001: PB[15]引脚 0002: PC[15]引脚
位 11:8	EXINT14	0x0	rw	EXINT14 配置 (EXINT14 configuration) 这些位可由软件读写, 用于选择 EXINT14 外部中断的输入源。 0000: PA[14]引脚 0001: PB[14]引脚 0002: PC[14]引脚
位 7:4	EXINT13	0x0	rw	EXINT13 配置 (EXINT13 configuration) 这些位可由软件读写, 用于选择 EXINT13 外部中断的输入源。 0000: PA[13]引脚 0001: PB[13]引脚 0002: PC[13]引脚
位 3:0	EXINT12	0x0	rw	EXINT8 配置 (EXINT12 configuration) 这些位可由软件读写, 用于选择 EXINT12 外部中断的输入源。 0000: PA[12]引脚 0001: PB[12]引脚

# 8 外部中断/事件控制器 (EXINT)

## 8.1 EXINT介绍

EXINT 共计有 22 条中断线 EXINT\_LINE[21:0] (线 18、20 保留), 每条中断线均支持通过边沿检测触发和软件触发来产生中断或事件。EXINT 可以根据软件配置, 独立的使能或禁止中断或事件, 并采取不同的边沿检测方式 (检测上升沿或检测下降沿或同时检测上升沿和下降沿) 以及触发方式 (边沿检测触发或软件触发或边沿检测和软件同时触发) 响应触发源独立的产生中断或事件。

图 8-1 外部中断/事件控制器框图



### EXINT 控制器的主要特性:

- 中断线 0~15 所映射的 IO 可以独立的配置
- 每个中断线都有独立的触发方式选择
- 每个中断都有独立的使能位
- 每个事件都有独立的使能位
- 共 20 个可独立产生和清除的软件触发
- 每个中断都有独立的状态位
- 每个中断都可以被独立的清除

## 8.2 功能描述和配置流程

EXINT 共计有 22 条中断线 EXINT\_LINE[21:0] (线 18、20 保留), 可以通过边沿检测的方式分别检测来自 GPIO 的外部中断源以及包括 PVM 输出, ERTC 闹钟事件, ERTC 入侵和时间戳事件以及 COMP 事件共四种芯片内部的中断源, 其中来自 GPIO 的中断源可以通过软件编程配置 SCFG 中的外部中断配置寄存器 x (SCFG\_EXINTCx) 灵活的选择, 需要注意的是这些输入源是互斥的, 例如 EXINT\_LINE0 只能选择 PA0/PB0/PC0/PD0…中的某一个, 而不能同时选择 PA0 和 PB0 作为输入源。

EXINT 支持多种边沿检测方式, 每条中断线可以通过软件编程配置极性配置寄存器 1 (EXINT\_POLCFG1) 和极性配置寄存器 2 (EXINT\_POLCFG2) 独立的选择上升沿检测或下降沿检测或同时进行上升沿和下降沿检测, 中断线上检测到的有效边沿触发可以用于产生事件或中断。

EXINT 支持独立的软件触发产生中断或事件, 即除了来自中断线上的有效边沿外, 用户可以通过软件编程配置软件触发寄存器 (EXINT\_SWTRG) 对应位来产生对应的中断或事件。

EXINT 具备独立的中断和事件使能位, 用户可以通过软件编程配置中断使能寄存器 (EXINT\_INTEN) 和事件使能寄存器 (EXINT\_EVTEN) 来使能或关闭对应的中断或事件, 这意味着无论是通过边沿检测还是软件触发产生中断或事件, 都需要提前使能对应的中断或事件。

EXINT 具备独立的中断状态位, 用户可以通过中断状态寄存器 (EXINT\_INTSTS) 读取对应的中断状态并通过对该寄存器相应位写 1 来清除已置位的状态标志。

### 中断初始化流程

#### 1. 选择中断源

即配置复用外部中断配置寄存器 x (SCFG\_EXINTCx) (如果需要使用 GPIO 作为中断源需要该步骤)。

#### 2. 选择触发方式

即配置极性配置寄存器 1 (EXINT\_POLCFG1) 和极性配置寄存器 2 (EXINT\_POLCFG2)。

### 3. 使能中断或事件

即配置中断使能寄存器（EXINT\_INTEN）和事件使能寄存器（EXINT\_EVTEN）。

### 4. 产生软件触发

即配置软件触发寄存器（EXINT\_SWTRG）产生软件触发（此步骤仅适用于需软件触发产生中断的应用）。

**注意：**若需要更改中断源配置，应先关闭中断使能寄存器和事件使能寄存器后，再重新开始中断初始化流程的配置。

### 中断清除流程

- 清除标志，即对中断状态寄存器（EXINT\_INTSTS）对应位写 1 来清除已产生的中断，同时该操作会同步清除软件触发寄存器（EXINT\_SWTRG）中的对应位。

## 8.3 EXINT 寄存器描述

下表列出了 EXINT 寄存器的映像和复位值。

必须以字（32 位）的方式操作这些外设寄存器。

表 8-1 外部中断/事件控制器寄存器映像和复位值

寄存器简称	基址偏移量	复位值
EXINT_INTEN	0x00	0x0000 0000
EXINT_EVTEN	0x04	0x0000 0000
EXINT_POLCFG1	0x08	0x0000 0000
EXINT_POLCFG2	0x0C	0x0000 0000
EXINT_SWTRG	0x10	0x0000 0000
EXINT_INTSTS	0x14	0x0000 0000

### 8.3.1 中断使能寄存器（EXINT\_INTEN）

域	简称	复位值	类型	功能
位 31: 22	保留	0x000	resd	硬件强制为 0。 线 x 上的中断使能/禁止位（Interrupt enable or disable on line x）
位 21: 0	INTENx	0x00000	rw	0: 禁止中断请求； 1: 使能中断请求。 注：位 18、位 20 为保留位，未使用。

### 8.3.2 事件使能寄存器（EXINT\_EVTEN）

域	简称	复位值	类型	功能
位 31: 22	保留	0x000	resd	硬件强制为 0。 线 x 上的事件使能/禁止位（Event enable or disable on line x）
位 21: 0	EVTENx	0x00000	rw	0: 禁止事件请求； 1: 使能事件请求。 注：位 18、位 20 为保留位，未使用。

### 8.3.3 极性配置寄存器 1（EXINT\_POLCFG1）

域	简称	复位值	类型	功能
位 31: 22	保留	0x000	resd	硬件强制为 0。 线 x 上的上升沿触发事件配置位（Rising polarity configuration bit of line x）
位 21: 0	RPx	0x00000	rw	这些位用于选择线 x 由上升沿触发中断和事件。 0: 禁止上升沿触发； 1: 使能上升沿触发。 注：位 18、位 20 为保留位，未使用。

### 8.3.4 极性配置寄存器2（EXINT\_POLCFG2）

域	简称	复位值	类型	功能
位 31: 22	保留	0x000	resd	硬件强制为 0。
位 21: 0	FPx	0x00000	rw	<p>线 x 上的下降沿触发事件配置位（Falling polarity event configuration bit of line x）</p> <p>这些位用于选择线 x 由下降沿触发中断和事件。</p> <p>0: 禁止下降沿触发； 1: 允许下降沿触发。</p> <p>注：位 18、位 20 为保留位，未使用。</p>

### 8.3.5 软件触发寄存器（EXINT\_SWTRG）

域	简称	复位值	类型	功能
位 31: 22	保留	0x000	resd	硬件强制为 0。
位 21: 0	SWTx	0x00000	rw	<p>软件触发线 x（Software trigger on line x）</p> <p>当中断使能寄存器（EXINT_INTEN）中的对应位为 1，则软件写此位硬件将自动置起中断状态寄存器（EXINT_INTSTS）中的对应位并产生中断。</p> <p>当事情使能寄存器（EXINT_EVTEN）中的对应位为 1，则软件写此位硬件将自动产生对应中断线上的事件。</p> <p>0: 默认值； 1: 产生软件触发。</p> <p>注：通过清除中断状态寄存器（EXINT_INTSTS）的对应位（写入 1），可以清除该位为 0。</p> <p>注：位 18、位 20 为保留位，未使用。</p>

### 8.3.6 中断状态寄存器（EXINT\_INTSTS）

域	简称	复位值	类型	功能
位 31: 22	保留	0x000	resd	硬件强制为 0。
位 21: 0	LINEx	0x00000	rw1c	<p>线 x 状态位（Line x state bit）</p> <p>0: 没有发生中断； 1: 发生了中断。</p> <p>注：在该位中写入 '1' 可以清除它。</p> <p>注：位 18、位 20 为保留位，未使用。</p>

## 9 DMA 控制器 (DMA)

### 9.1 简介

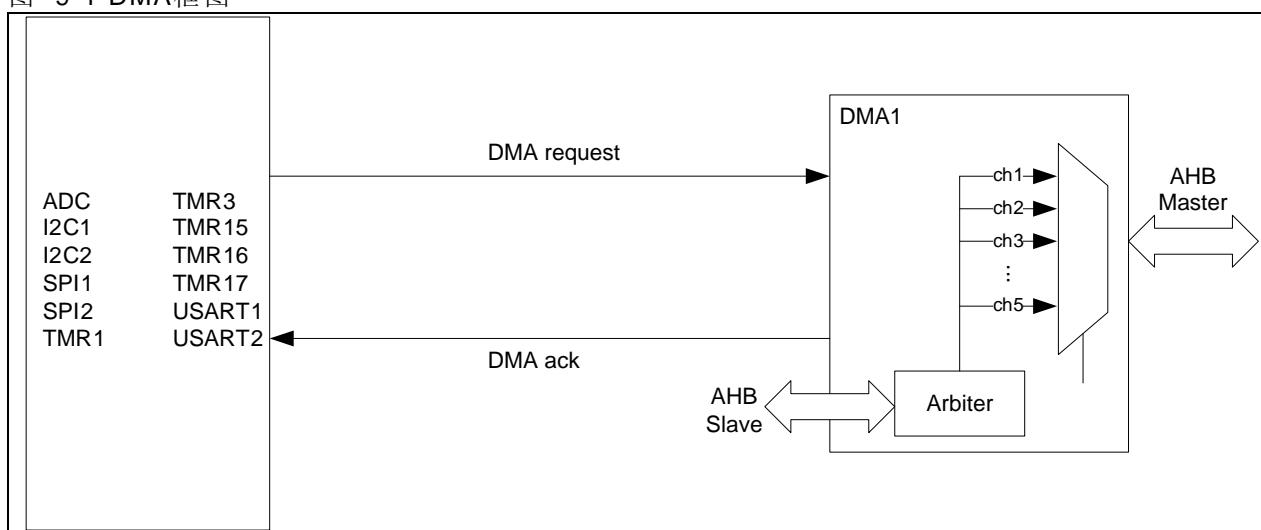
直接存储器访问 (DMA) 控制器，不仅旨在增强系统性能并减少处理器的中断生成，而且还针对 32 位 MCU 应用程序而设计。

一个控制器有 5 个 DMA 通道，每个通道管理来自于外设对存储器访问的请求，并由仲裁器来协调各个 DMA 请求的优先权。

### 9.2 特性

- 符合 AMBA 规范 (Rev. 2.0)
- 仅支持 AHB OKAY 和 ERROR 响应
- 不支持 AHB 主接口的 HBUSREQ 和 HGRANT
- 支持 5 个通道
- 支持外设到存储器，存储器到外设和存储器到存储器的传输
- 支持硬件握手
- 支持 8 位，16 位和 32 位数据宽度传输
- 传输数据长度最大为 65535，可由编程配置

图 9-1 DMA 框图



注意：根据不同型号，图中 DMA 外设可能会有所减少。

### 9.3 功能描述

#### 9.3.1 通道配置

1. 设置外设地址 (**DMA通道x外设地址寄存器 (DMA\_CxPADDR)**)  
数据传输的初始外设地址，在传输过程中不会被改变。
2. 设置存储器地址 (**DMA通道x存储器地址寄存器 (DMA\_CxMADDR)**)  
数据传输的初始存储器地址，在传输过程中不会被改变。
3. 配置数据传输量 (**DMA通道x数据传输量寄存器 (DMA\_CxDTCNT)**)  
可编程的数据传输长度最大为 65535。在传输过程中，该传输数据量的值会逐渐递减。
4. 配置通道设定 (**DMA通道x配置寄存器 (DMA\_CxCTRL)**)  
包含通道优先级，数据传输的方向、宽度，地址增量模式、循环模式和中断方式。

##### ● 通道优先级 (CHPL)

分为 4 个等级，最高优先级、高优先级、中等优先级和低优先级。

若有 2 个通道优先级设定相同，则较低编号的通道有较高的优先权。举例，通道 1 优先于通道 2。

- **数据传输方向 (DTD)**

分为存储器到外设 (M2P)，外设到存储器 (P2M)。

- **地址增量模式 (PINCM/MINCM)**

当设置为增量模式时，下一笔传输的地址将是前一笔传输地址加上传输宽度 (PWIDHT/MWIDHT)。

- **循环模式 (LM)**

当通道配置设定为循环模式时，在最后一次传输后 DMA 通道 x 数据传输量寄存器 (DMA\_CxDTCNT) 的内容会恢复成初始值。

- **存储器到存储器模式 (M2M)**

存储器到存储器模式是 DMA 在没有外设请求的情况下进行数据传输。

循环模式与存储器到存储器模式不能同时使用。

5. 使能该通道的DMA传输 (DMA通道x配置寄存器 (DMA\_CxCTRL) 的CHEN位)

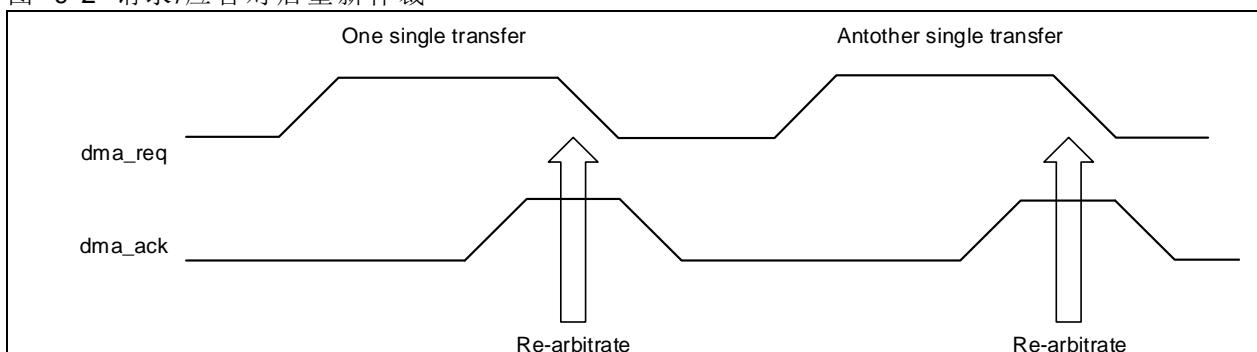
### 9.3.2 握手机制

在 P2M 和 M2P 传输模式，外设需要向 DMA 控制器发送请求信号。该通道将发出外设传输 (单次)，直到请求信号被应答为止。外设传输完成后，DMA 控制器将应答信号发送到外设。外设从 DMA 控制器获得应答信号后立即释放其请求。一旦外设取消了请求，DMA 控制器将释放应答信号。

### 9.3.3 仲裁

当同时启用多个通道时，仲裁器将在主控制器完全传输数据后重新进行仲裁。优先级最高的通道等待当前占用主控制器的通道完成数据传输后，将具有主控制器使用权。每当通道以外设主控制器的优先级完成一个单次传输后，外设主控制器就会重新仲裁以服务其他通道。

图 9-2 请求/应答对后重新仲裁



### 9.3.4 可编程数据传输宽度

通过 DMA 通道 x 配置寄存器 (DMA\_CxCTRL) 中的 PWIDHT 和 MWIDHT 位可以对源数据和目标数据的数据宽度进行编程，当 PWIDHT 不等于 MWIDHT 时，会依据 PWIDHT/ MWIDHT 设定将资料对齐。

图 9-3 PWIDHT: byte, MWIDHT: half-word

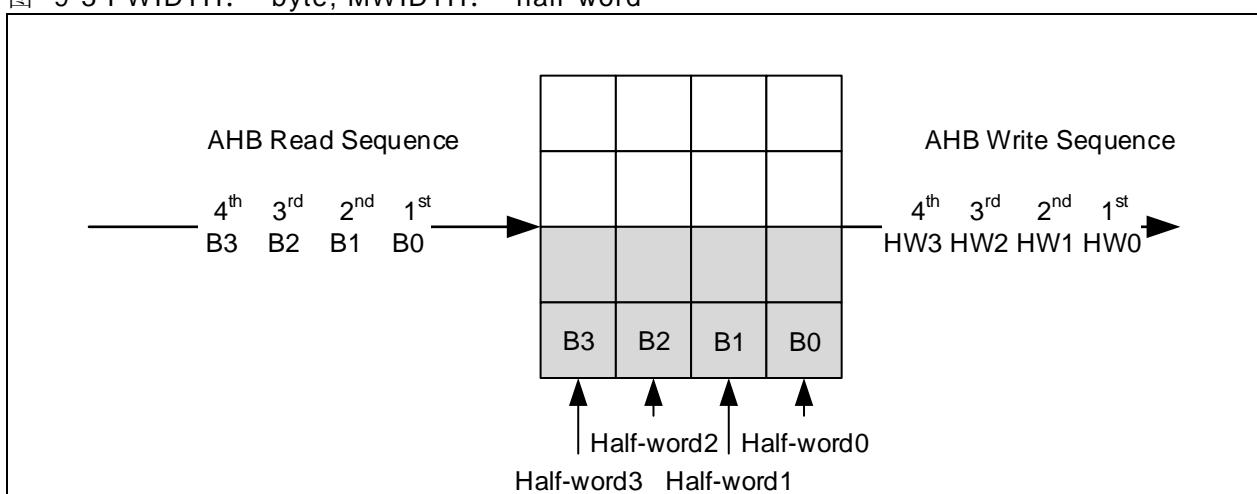


图 9-4 PWIDHT: half-word, MWIDHT: word

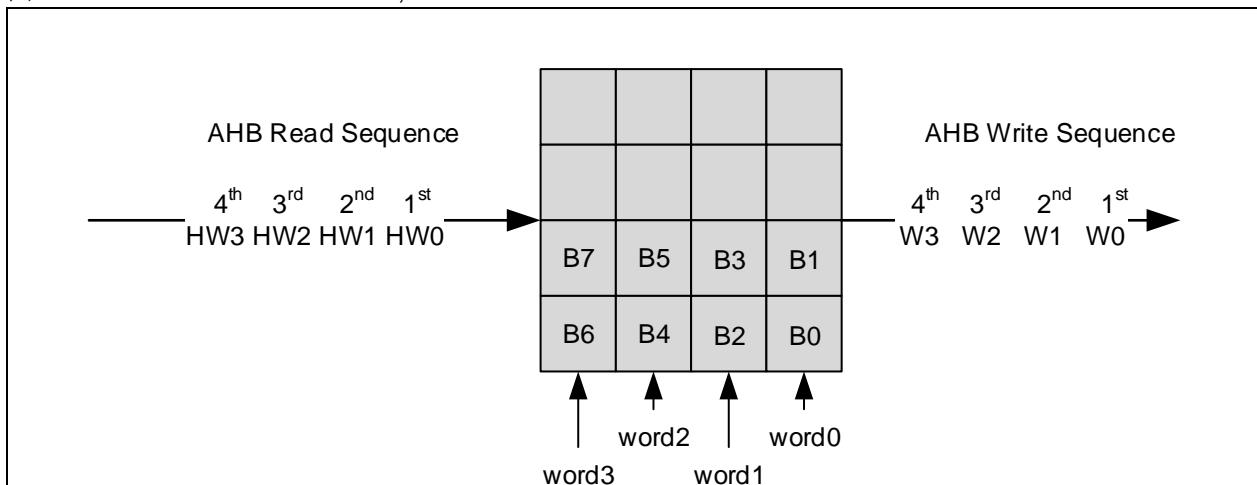
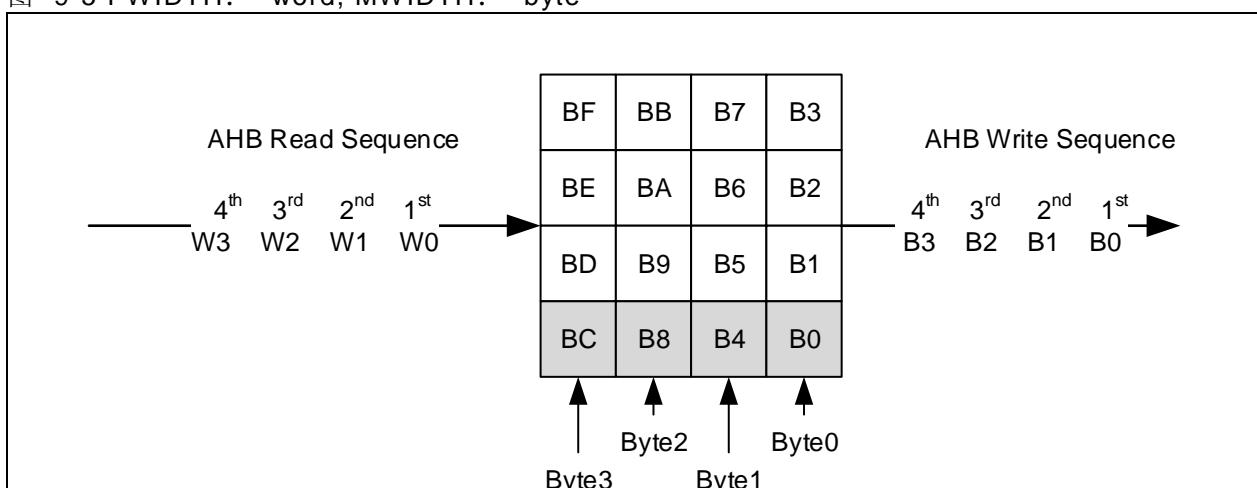


图 9-5 PWIDHT: word, MWIDHT: byte



### 9.3.5 错误事件

表 9-1 DMA错误事件

错误事件

传输错误 DMA 读/写访问期间发生 AHB 响应错误

### 9.3.6 中断

DMA 可在传输过半、传输完成和传输错误时产生中断。每个通道的中断都有专用标志，清除和使能位如下表所示。

表 9-2 DMA中断

中断事件	事件标志位	清除控制位	使能控制位
半传输	HDTF	HDTFC	HDTIEN
传输完成	FDTF	FDTFC	FDTIEN
传输错误	DTERRF	DTERRFC	DTERRIEN

### 9.3.7 DMA固定请求映射

数个外设请求通过逻辑“OR”运算映射到一个 DMA 通道，用户必须确保在一个通道上一次仅激活一个外设请求。另外，通过设置相应外设寄存器中的控制位，可以独立地开启或关闭外设的 DMA 请求。

表 9-3 DMA各通道的外设请求

外设	通道 1	通道 2	通道 3	通道 4	通道 5
ADC	ADC <sup>(1)</sup>	ADC <sup>(2)</sup>			
SPI/I2S		SPI1/I2S1_RX	SPI1/I2S1_TX	SPI2/I2S2_RX	SPI2/I2S2_TX
USART1		USART1_TX <sup>(1)</sup>	USART1_RX <sup>(1)</sup>	USART1_TX <sup>(2)</sup>	USART1_RX <sup>(2)</sup>
USART2				USART2_TX	USART2_RX
I2C		I2C1_TX	I2C1_RX	I2C2_TX	I2C2_RX

TMR1		TMR1_CH1	TMR1_CH2	TMR1_CH4 TMR1_TRIG TMR1_HALL	TMR1_CH3 TMR1_OVERFLOW
MR3		TMR3_CH3	TMR3_CH4 TMR3_OVERFLOW	TMR3_CH1 TMR3_TRIG	
TMR6			TMR6_OVERFLOW		
TMR15					TMR15_CH1 TMR15_OVERFLOW TMR15_TRIG TMR15_HALL TMR15_CH2
TMR16			TMR16_CH1 <sup>(1)</sup> TMR16_OVERFLOW W <sup>(1)</sup>	TMR16_CH1 <sup>(2)</sup> TMR16_OVERFLOW <sup>(2)</sup>	
TMR17	TMR17_CH1 <sup>(1)</sup> TMR17_OVERFLOW W <sup>(1)</sup>	TMR17_CH1 <sup>(2)</sup> TMR17_OVERFLOW <sup>(2)</sup>			

1: SCFG\_CFG1中相应的remap位为0时，相应DMA请求映射到此通道

2: SCFG\_CFG1中相应的remap位为1时，相应DMA请求映射到此通道

## 9.4 DMA寄存器

下表列出了 DMA 寄存器的映像和复位值。

可以用字节（8 位）、半字（16 位）或字（32 位）的方式操作这些外设寄存器。

表 9-4 DMA寄存器的映像和复位值

寄存器简称	基址偏移量	复位值
DMA_STS	0x00	0x0000 0000
DMA_CLR	0x04	0x0000 0000
DMA_C1CTRL	0x08	0x0000 0000
DMA_C1DTCNT	0x0C	0x0000 0000
DMA_C1PADDR	0x10	0x0000 0000
DMA_C1MADDR	0x14	0x0000 0000
DMA_C2CTRL	0x1C	0x0000 0000
DMA_C2DTCNT	0x20	0x0000 0000
DMA_C2PADDR	0x24	0x0000 0000
DMA_C2MADDR	0x28	0x0000 0000
DMA_C3CTRL	0x30	0x0000 0000
DMA_C3DTCNT	0x34	0x0000 0000
DMA_C3PADDR	0x38	0x0000 0000
DMA_C3MADDR	0x3C	0x0000 0000
DMA_C4CTRL	0x44	0x0000 0000
DMA_C4DTCNT	0x48	0x0000 0000
DMA_C4PADDR	0x4C	0x0000 0000
DMA_C4MADDR	0x50	0x0000 0000
DMA_C5CTRL	0x58	0x0000 0000
DMA_C5DTCNT	0x5C	0x0000 0000
DMA_C5PADDR	0x60	0x0000 0000
DMA_C5MADDR	0x64	0x0000 0000

### 9.4.1 DMA状态寄存器 (DMA\_STS)

访问：无等待状态，字，半字和字节访问

域	简称	复位值	类型	功能
31: 20	保留	0x0	resd	保持默认值。
位 19	DTERRF5	0x0	ro	通道 5 数据传输错误事件标志 (data transfer error event flag) 0: 未发生错误传输事件 1: 发生错误传输事件
位 18	HDTF5	0x0	ro	通道 5 半数据传输事件标志 (half data transfer event flag) 0: 未发生半传输事件 1: 发生半传输事件
位 17	FDTF5	0x0	ro	通道 5 数据传输完成事件标志 (full data transfer event flag) 0: 未发生传输完成事件 1: 发生传输完成事件
位 16	GF5	0x0	ro	通道 5 全局事件标志 (Global event flag) 0: 未发生传输错误、半传输完成或传输完成事件 1: 发生传输错误、半传输完成或传输完成事件
位 15	DTERRF4	0x0	ro	通道 4 数据传输错误事件标志 (data transfer error event flag) 0: 未发生错误传输事件 1: 发生错误传输事件
位 14	HDTF4	0x0	ro	通道 4 半数据传输事件标志 (half data transfer event flag) 0: 未发生半传输事件 1: 发生半传输事件
位 13	FDTF4	0x0	ro	通道 4 数据传输完成事件标志 (full data transfer event flag) 0: 未发生传输完成事件 1: 发生传输完成事件
位 12	GF4	0x0	ro	通道 4 全局事件标志 (Global event flag) 0: 未发生传输错误、半传输完成或传输完成事件 1: 发生传输错误、半传输完成或传输完成事件
位 11	DTERRF3	0x0	ro	通道 3 数据传输错误事件标志 (data transfer error event flag) 0: 未发生错误传输事件 1: 发生错误传输事件
位 10	HDTF3	0x0	ro	通道 3 半数据传输事件标志 (half data transfer event flag) 0: 未发生半传输事件 1: 发生半传输事件
位 9	FDTF3	0x0	ro	通道 3 数据传输完成事件标志 (full data transfer event flag) 0: 未发生传输完成事件 1: 发生传输完成事件
位 8	GF3	0x0	ro	通道 3 全局事件标志 (Global event flag) 0: 未发生传输错误、半传输完成或传输完成事件 1: 发生传输错误、半传输完成或传输完成事件
位 7	DTERRF2	0x0	ro	通道 2 数据传输错误事件标志 (data transfer error event flag) 0: 未发生错误传输事件 1: 发生错误传输事件

位 6	HDTF2	0x0	ro	通道 2 半数据传输事件标志 (half data transfer event flag) 0: 未发生半传输事件 1: 发生半传输事件
位 5	FDTF2	0x0	ro	通道 2 数据传输完成事件标志 (full data transfer event flag) 0: 未发生传输完成事件 1: 发生传输完成事件
位 4	GF2	0x0	ro	通道 2 全局事件标志 (Global event flag) 0: 未发生传输错误、半传输完成或传输完成事件 1: 发生传输错误、半传输完成或传输完成事件
位 3	DTERRF1	0x0	ro	通道 1 数据传输错误事件标志 (data transfer error event flag) 0: 未发生错误传输事件 1: 发生错误传输事件
位 2	HDTF1	0x0	ro	通道 1 半数据传输事件标志 (half data transfer event flag) 0: 未发生半传输事件 1: 发生半传输事件
位 1	FDTF1	0x0	ro	通道 1 数据传输完成事件标志 (full data transfer event flag) 0: 未发生传输完成事件 1: 发生传输完成事件
位 0	GF1	0x0	ro	通道 1 全局事件标志 (Global event flag) 0: 未发生传输错误、半传输完成或传输完成事件 1: 发生传输错误、半传输完成或传输完成事件

#### 9.4.2 DMA状态清除寄存器 (DMA\_CLR)

访问：无等待状态，字，半字和字节访问

域	简称	复位值	类型	功能
31: 20	保留	0x0	resd	保持默认值。
位 19	DTERRFC5	0x0	rw1c	清除通道 5 的数据传输错误标志 (data transfer error flag clear) 0: 无效 1: 清除 DMA_STS 寄存器中 DTERRF5 标志
位 18	HDTFC5	0x0	rw1c	清除通道 5 的半数据传输标志 (half data transfer flag clear) 0: 无效 1: 清除 DMA_STS 寄存器中 HDTF5 标志
位 17	FDTFC5	0x0	rw1c	清除通道 5 的数据传输完成标志 (full data transfer flag clear) 0: 无效 1: 清除 DMA_STS 寄存器中 FDTF5 标志
位 16	GFC5	0x0	rw1c	清除通道 5 的全局中断标志 (Global flag clear) 0: 无效 1: 清除 DMA_STS 寄存器中 DTERRF5、HDTF5、FDTF5 和 GF5 标志
位 15	DTERRFC4	0x0	rw1c	清除通道 4 的数据传输错误标志 (data transfer error flag clear) 0: 无效 1: 清除 DMA_STS 寄存器中 DTERRF4 标志
位 14	HDTFC4	0x0	rw1c	清除通道 4 的半数据传输标志 (half data transfer flag clear) 0: 无效 1: 清除 DMA_STS 寄存器中 HDTF4 标志

位 13	FDTFC4	0x0	rw1c	清除通道 4 的数据传输完成标志 (full data transfer flag clear) 0: 无效 1: 清除 DMA_STS 寄存器中 FDTF4 标志
位 12	GFC4	0x0	rw1c	清除通道 4 的全局中断标志 (Global flag clear) 0: 无效 1: 清除 DMA_STS 寄存器中 DTERRF4、HDTF4 FDTF4 和 GF4 标志
位 11	DTERRFC3	0x0	rw1c	清除通道 7 的数据传输错误标志 (data transfer error flag clear) 0: 无效 1: 清除 DMA_STS 寄存器中 DTERRF7 标志
位 10	HDTFC3	0x0	rw1c	清除通道 7 的半数据传输标志 (half data transfer flag clear) 0: 无效 1: 清除 DMA_STS 寄存器中 HDTF7 标志
位 9	FDTFC3	0x0	rw1c	清除通道 3 的数据传输完成标志 (full data transfer flag clear) 0: 无效 1: 清除 DMA_STS 寄存器中 FDTF3 标志
位 8	GFC3	0x0	rw1c	清除通道 3 的全局中断标志 (Global flag clear) 0: 无效 1: 清除 DMA_STS 寄存器中 DTERRF3、HDTF3 FDTF3 和 GF3 标志
位 7	DTERRFC2	0x0	rw1c	清除通道 2 的数据传输错误标志 (data transfer error flag clear) 0: 无效 1: 清除 DMA_STS 寄存器中 DTERRF2 标志
位 6	HDTFC2	0x0	rw1c	清除通道 2 的半数据传输标志 (half data transfer flag clear) 0: 无效 1: 清除 DMA_STS 寄存器中 HDTF2 标志
位 5	FDTFC2	0x0	rw1c	清除通道 2 的数据传输完成标志 (full data transfer flag clear) 0: 无效 1: 清除 DMA_STS 寄存器中 FDTF2 标志
位 4	GFC2	0x0	rw1c	清除通道 2 的全局中断标志 (Global flag clear) 0: 无效 1: 清除 DMA_STS 寄存器中 DTERRF2、HDTF2 FDTF2 和 GF2 标志
位 3	DTERRFC1	0x0	rw1c	清除通道 1 的数据传输错误标志 (data transfer error flag clear) 0: 无效 1: 清除 DMA_STS 寄存器中 DTERRF1 标志
位 2	HDTFC1	0x0	rw1c	清除通道 1 的半数据传输标志 (half data transfer flag clear) 0: 无效 1: 清除 DMA_STS 寄存器中 HDTF1 标志
位 1	FDTFC1	0x0	rw1c	清除通道 1 的数据传输完成标志 (full data transfer flag clear) 0: 无效 1: 清除 DMA_STS 寄存器中 FDTF1 标志
位 0	GFC1	0x0	rw1c	清除通道 1 的全局中断标志 (Global flag clear) 0: 无效 1: 清除 DMA_STS 寄存器中 DTERRF1、HDTF1 FDTF1 和 GF1 标志

### 9.4.3 DMA通道x配置寄存器 (DMA\_CxCTRL) (x = 1…5)

访问：无等待状态，字，半字和字节访问

域	简称	复位值	类型	功能
位 31: 15	保留	0x00000	resd	保持默认值。
位 14	M2M	0x0	rw	存储器到存储器模式 (Memory to memory mode) 0: 关闭 1: 开启
位 13: 12	CHPL	0x0	rw	通道优先级 (Channel preemptive level) 00: 低优先级 01: 中优先级 10: 高优先级 11: 最高优先级
位 11: 10	MWIDTH	0x0	rw	存储器数据宽度 (Memory data bit width) 00: 8 bit 位宽 01: 16 bit 位宽 10: 32 bit 位宽 11: 保留
位 9: 8	PWIDTH	0x0	rw	外设数据宽度 (Peripheral data bit width) 00: 8 bit 位宽 01: 16 bit 位宽 10: 32 bit 位宽 11: 保留
位 7	MINCM	0x0	rw	存储器地址递增模式 (Memory address increment mode) 0: 关闭 1: 开启
位 6	PINCM	0x0	rw	外设地址递增模式 (Peripheral address increment mode) 0: 关闭 1: 开启
位 5	LM	0x0	rw	循环模式 (Loop mode) 0: 关闭 1: 开启
位 4	DTD	0x0	rw	数据传输方向 (Data transfer direction) 0: 外设为源 1: 存储器为源
位 3	DTERRIEN	0x0	rw	允许数据传输错误中断 (data transfer error interrupt enable) 0: 禁止数据传输错误中断 1: 允许数据传输错误中断
位 2	HDTIEN	0x0	rw	允许半数据传输中断 (half data transfer interrupt enable) 0: 禁止半数据传输中断
位 1	FDTIEN	0x0	rw	允许数据传输完成中断 (full data transfer interrupt enable) 0: 禁止数据传输完成中断 1: 允许数据传输完成中断
位 0	CHEN	0x0	rw	通道使能 (Channel enable) 0: 关闭 1: 开启

#### 9.4.4 DMA通道x数据传输量寄存器 (DMA\_CxDTCNT) (x = 1…5)

访问：无等待状态，字，半字和字节访问

域	简称	复位值	类型	功能
位 31: 16	保留	0x0000	resd	保持默认值。
位 15: 0	CNT	0x0000	rw	DMA 通道数据传输个数 (Number of data to transfer) DMA 通道传输数据个数范围为 0x0~0xFFFF，在更改 DMA 通道传输数据个数时需要确保对应通道的 CHEN 位为 0，否则无法写入；DMA 控制器每传输完一笔数据，此值会硬件减 1。 注：此寄存器为传输数据个数，不是传输数据量大小；传输数据量大小需要根据数据宽度换算得到。

#### 9.4.5 DMA通道x外设地址寄存器 (DMA\_CxPADDR) (x = 1…5)

访问：无等待状态，字，半字和字节访问

域	简称	复位值	类型	功能
位 31: 0	PADDR	0x0000 0000	rw	外设端基地址 (Peripheral base address) 外设数据寄存器的地址，作为数据传输的源或目标。 注：确保对应通道的 CHEN 位为 0，否则无法写入。

#### 9.4.6 DMA通道x存储器地址寄存器 (DMA\_CxMADDR) (x = 1…5)

访问：无等待状态，字，半字和字节访问

域	简称	复位值	类型	功能
位 31: 0	MADDR	0x0000 0000	rw	储存器端基地址 (Memory base address) 储存器地址作为数据传输的源或目标。 注：确保对应通道的 CHEN 位为 0，否则无法写入。

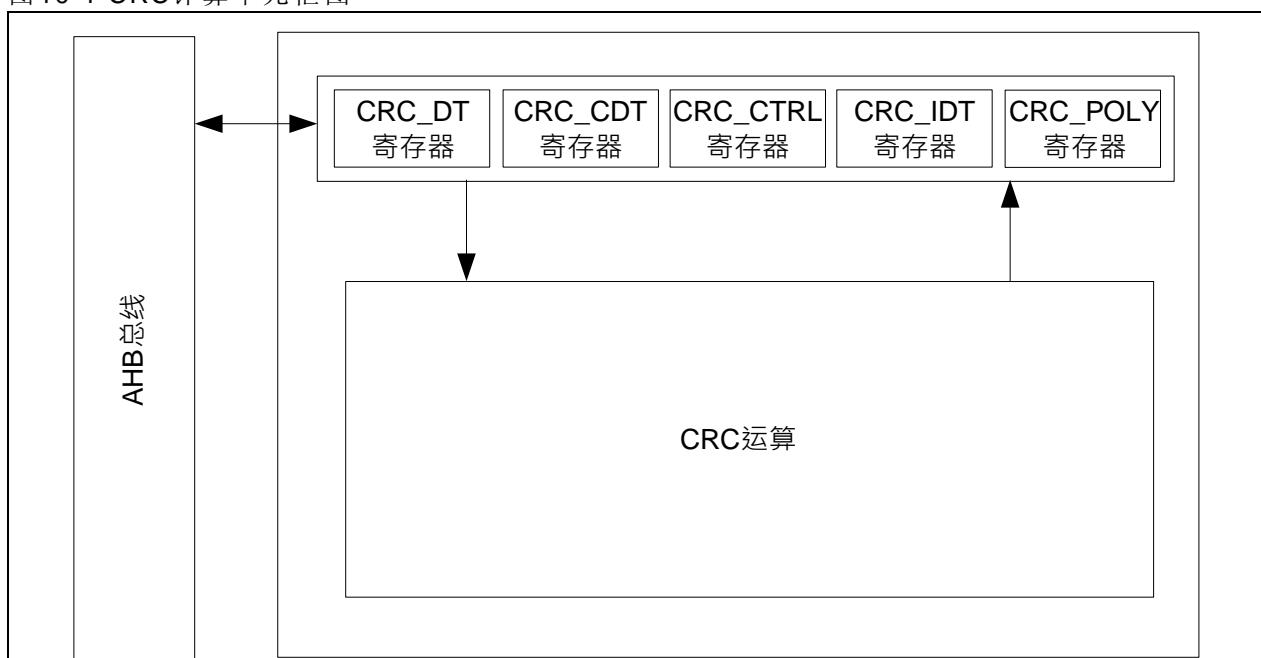
# 10 CRC 计算单元 (CRC)

## 10.1 CRC介绍

CRC计算单元是一个独立的具备CRC计算功能的外设，CRC计算单元采用CRC32/MPEG-2。用户可以通过软件编程配置控制寄存器(CRC\_CTRL)选择是否进行输出数据翻转(全字翻转, REVOD=1)或输入数据翻转(字节翻转, REVID=01; 半字翻转, REVID=10; 全字翻转, REVID=11), CRC计算单元还提供初始化功能, 每次RESET操作后, CRC计算单元会将初始化寄存器(CRC\_IDT)中的值搬入数据寄存器(CRC\_DT)。CRC\_POLY寄存器可让用户软件编程不同的生成多项式系数, 并透过CRC\_CTRL的POLY\_SIZE将生成多项式的大小配置为7/8/16/32位。

用户通过写和读数据寄存器(CRC\_DT)的方式, 写入想要进行计算的值, 读出计算的结果, 注意每次的CRC计算结果是前一次计算结果与当前待计算值的组合。

图 10-1 CRC计算单元框图



### CRC 主要特性:

- 预设采用 CRC-32 标准
- 可编程生成多项式
- 一次 CRC 计算需要 4 个 HCLK
- 输入输出数据格式可翻转
- 待计算值的写入和计算结果的读出都通过写和读 CRC\_DT 实现
- 配置 CRC\_IDT 写入初始化值, 在每次 CRC 复位后该值会加载到 CRC\_DT

## 10.2 CRC功能说明

CRC的计算原理是将输入数据做为被除数, 与作为除数的生成多项式进行模二除法, 得到的余数即为CRC值。

### CRC 运算流程

- 输入翻转, 即数据输入后, 先依据 CRC\_CTRL 的 REVID 值进行输入数据翻转
- 初始化, 首次计算会与 CRC\_IDT 设定的初始值做 XOR。若非首次计算, 则初始值为上次的计算结果。
- CRC 计算, 与生成多项式(0x4C11DB7)进行模二除法, 所得余数为 CRC 值
- 输出翻转, 依据 CRC\_CTRL 的 REVOD 决定是否将 CRC 值执行全字翻转后再输出。
- 对结果进行 XOR 运算, 结果异或值固定为 0x0000 0000。

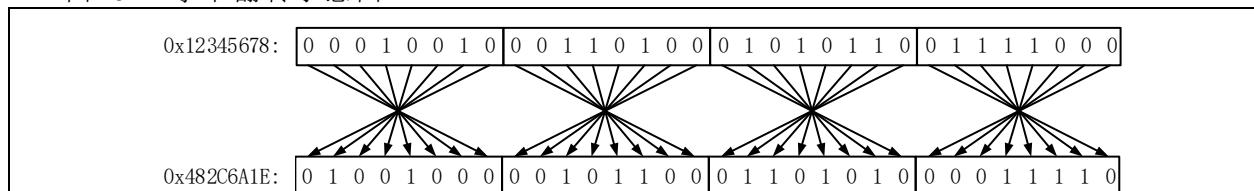
**CRC-32/MPEG-2 参数说明**

- 生成多项式: 0x4C11DB7,  
即  $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- 初始值: 0xFFFF FFFF, 目的为避免待测数据为 1 字节 0x00 和多字节 0x00 的结果相同。
- 结果异或值: 0x0000 0000, 此值表示不对 CRC 结果再进行一次 XOR 运算

**翻转功能说明**

- 选择以字节翻转, 则 8bit 为一组, 组内排列顺序颠倒。如下图所示, 若原数据为 0x12345678, 翻转后为 0x482C6A1E。
- 选择以半字翻转, 则 16bit 为一组, 组内排列顺序颠倒。
- 选择以字翻转, 则 32bit 为一组, 组内排列顺序颠倒。

图 10-2 字节翻转示意图



## 10.3 CRC 寄存器

除 CRC\_DT 可以用字节 (8 位)、半字 (16 位) 或字 (32 位) 的方式操作之外, 其他寄存器必须以字 (32 位) 的方式操作。

表 10-1 CRC 计算单元寄存器映像

寄存器简称	基址偏移量	复位值
CRC_DT	0x00	0xFFFF FFFF
CRC_CDT	0x04	0x0000 0000
CRC_CTRL	0x08	0x0000 0000
CRC_IDT	0x10	0xFFFF FFFF
CRC_POLY	0x14	0x04C1 1DB7

### 10.3.1 数据寄存器 (CRC\_DT)

域	简称	复位值	类型	功能
位 31: 0	DT	0xFFFF FFFF	rw	数据寄存器位 (Data value) 写入 CRC 计算器的新数据时, 作为输入寄存器读取时返回 CRC 计算的结果。

### 10.3.2 通用数据寄存器 (CRC\_CDT)

域	简称	复位值	类型	功能
位 31: 8	保留	0x0000000	resd	保持默认值。
位 7: 0	CDT	0x00	rw	通用 8 位数据寄存器位 (Common 8-bit data value) 可用于临时存放 1 字节的数据。控制寄存器 (CRC_CTRL) 的 RST 位产生的 CRC 复位对本寄存器没有影响。

### 10.3.3 控制寄存器 (CRC\_CTRL)

域	简称	复位值	类型	功能
位 31: 8	保留	0x000000	resd	保持默认值。
位 7	REVOD	0x0	resd	输出数据翻转 (Reverse output data) 由软件置起或清零。该位控制是否翻转输出数据。 0: 不翻转; 1: 全字翻转。
位 6: 5	REVID	0x0	rw	输入数据翻转 (Reverse input data) 由软件置起或清零。该位控制如何翻转输入数据。 00: 不翻转; 01: 字节翻转; 10: 半字翻转; 11: 全字翻转。
位 4: 3	POLY_SIZE	0x0	rw	生成多项式位宽(Polynomial size) 该位控制生成多项式的位宽大小，与 CRC_POLY 寄存器相配合。 00: 位宽为 32 位 01: 位宽为 16 位 10: 位宽为 8 位 11: 位宽为 7 位
位 2: 1	保留	0x0	resd	保持默认值。
位 0	RST	0x0	wo	RESET 位 (Reset CRC calculation unit) 由软件置起，由硬件自动清零。复位 CRC 计算单元，设置数据寄存器为 0xFFFF FFFF。 0: 无作用; 1: 复位。

### 10.3.4 初始化寄存器 (CRC\_IDT)

域	简称	复位值	类型	功能
位 31: 0	IDT	0xFFFF FFFF	rw	初始化数据寄存器 (Initial data value) 当控制寄存器 (CRC_CTRL) 的 RST 位产生的 CRC 复位时，初始化寄存器中的数值将作为数据寄存器 (CRC_DT) 的初始值写入。

### 10.3.5 生成多项式系数寄存器 (CRC\_POLY)

域	简称	复位值	类型	功能
位 31: 0	POLY	0x04C1 1DB7	rw	生成多项式系数寄存器 (polynomial coefficient) 生成多项式为 CRC 计算中的除数，预设使用 CRC32 参数模型，所以系数为 0x4C11DB7。用户亦可自行编程该生成多项式。

# 11 I<sup>2</sup>C 接口

## 11.1 I<sup>2</sup>C 简介

I<sup>2</sup>C 总线接口处理微控制器和串行 I<sup>2</sup>C 总线之间的通信，支持主机和从机模式，最大通信速度为 400kbit/s。

## 11.2 I<sup>2</sup>C 主要特点

- I<sup>2</sup>C 总线
  - 主机和从机模式
  - 多主机功能
  - 标准模式(100kHz)和快速模式(400kHz)
  - 7-bit 和 10-bit 地址模式
  - 广播呼叫模式
  - 状态标志
  - 错误标志
  - 时钟延展功能
  - 通讯事件中断
  - 错误中断
- 支持 DMA 传输
- 支持部分 SMBus2.0 协议
  - PEC 产生及检查
  - SMBus 提醒功能
  - ARP(地址解析协议)
  - 超时机制
- PMBus

注意：I<sup>2</sup>C 总线频率可以最高增加到 1 MHz。想要获得更完整详细的解决方案，可以联系邻近的雅特力销售处寻求技术支持。

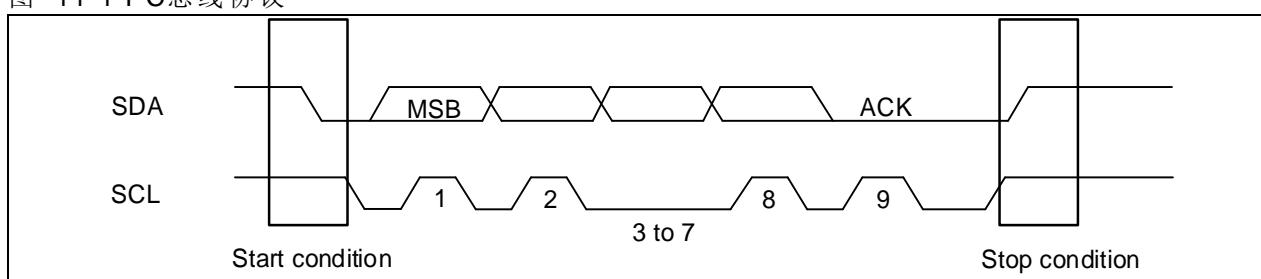
## 11.3 I<sup>2</sup>C 总线特性

I<sup>2</sup>C 总线是由数据线 SDA 和时钟线 SCL 构成，在标准模式下通信速度可达到 100kHz，快速模式下则可以达到 400kHz，一帧数据传输从开始信号开始，在结束信号后停止。在收到开始信号后总线的状态被认为是繁忙的，当收到结束信号后，总线被认为再次空闲。

开始信号：SCL 为高电平时，SDA 由高电平变为低电平。

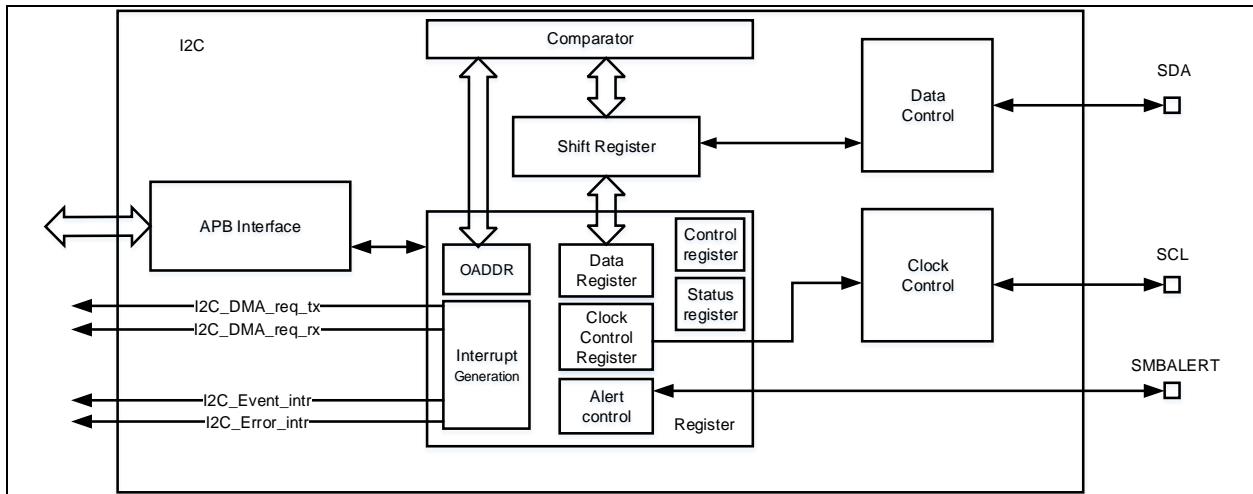
结束信号：SCL 为高电平时，SDA 由低电平变为高电平。

图 11-1 I<sup>2</sup>C 总线协议



## 11.4 I<sup>2</sup>C 接口

I<sup>2</sup>C 接口的功能框图示于下图。

图 11-2 I<sup>2</sup>C 的功能框图

### 1. I<sup>2</sup>C时钟

I<sup>2</sup>C 时钟由 APB1 或者 APB2 提供，可通过设置控制寄存器 2 (I2C\_CTRL2) 的 CLKFREQ[7: 0]对 I<sup>2</sup>C 时钟进行分频，在不同模式下对时钟有最低的速率要求，标准模式必须设定至少 2MHz，而快速模式下则是至少要 4MHz。

### 2. 接口工作模式

I<sup>2</sup>C 总线接口可以工作在主机模式与从机模式，并且可以相互切换。默认情况下处于从机模式，当设置了 GENSTART=1 产生了一个起始信号后，I<sup>2</sup>C 总线接口切换成主模式，当数据传输完成之后，也就是结束信号产生了之后，I<sup>2</sup>C 总线接口自动返回为从机模式。

- 主机发送模式
- 主机接收模式
- 从机发送模式
- 从机接收模式

### 3. 通信流程

- 主机模式通信流程：
  1. 产生开始信号
  2. 发送地址
  3. 发送或接收数据
  4. 产生结束信号
  5. 通信结束
- 从机模式通信流程：
  1. 等待地址匹配
  2. 发送或接收数据
  3. 等待结束信号产生
  4. 通信结束

### 4. 地址控制

主机和从机都支持 7 位和 10 位地址模式

#### 从机地址模式：

- 7 位地址模式
  - 单地址模式 ADDR2EN=0：此时只匹配OADDR1
  - 双地址模式 DUALEN=1：此时匹配OADDR1和OADDR2
- 10 位地址模式
  - 只匹配OADDR1

#### 从机特殊地址支持：

- 广播地址 (0b0000000x)：当 GCAEN=1 时该地址启用

- **SMBus** 设备默认地址（0b1100001x）：当在 **SMBus** 设备模式下该地址启用，该地址用于 **SMBus** 地址解析协议
- **SMBus** 主机默认地址（0b0001000x）：当在 **SMBus** 主机模式下该地址启用，该地址用于 **SMBus** 主机通知协议
- **SMBus** 提醒地址（0b0001100x）：当在 **SMBus** 主机模式下并且 **SMBALERT = 1** 下该地址启用，该地址用于 **SMBus** 提醒响应协议  
关于 **SMBus** 协议更详细的信息请参考 **SMBus2.0** 协议。

#### 从机地址匹配流程：

- 收到开始信号
- 匹配地址
- 若地址成功匹配，从机回一个 ACK
- 此时 ADDR7F 置 1，DIRF 指示传输方向
  - 如果DIRF=0从机进入接收模式，开始接收数据
  - 如果DIRF=1从机进入发送模式，开始发送数据

### 5. 时钟延展功能

时钟延展的功能的主要作用是当从机因为某些情况下不能及时的处理数据时，从机通过主动拉低 **SCL** 线，使通信暂停，避免数据丢失，软件可以通过设定控制寄存器 1(I2C\_CTRL1)的 **STRETCH** 位选择是否允许时钟延展。

- 在发送器模式：
  - 允许时钟延展：在发送下个字节（下一个数据的第一个 **SCL** 上升沿）前，若没有新数据写入数据寄存器(I2C\_DT)，则 I<sup>2</sup>C 接口拉低 **SCL** 总线，等待数据写入数据寄存器(I2C\_DT)
  - 不允许时钟延展：发送下个字节（下一个数据的第一个 **SCL** 上升沿）前，若没有新数据写入数据寄存器(I2C\_DT)，则发生欠载错误。
- 在接收器模式
  - 允许时钟延展：数据寄存器(I2C\_DT)内的数据未被读出，然后移位寄存器又接收完一个字节，I<sup>2</sup>C 接口拉低 **SCL** 总线，等待读取数据寄存器(I2C\_DT)
  - 不允许时钟延展：数据寄存器(I2C\_DT)内的数据未被读出，然后移位寄存器又接收完一个字节，此时如果又接收到一个数据，则发生过载错误。

## 11.4.1 I<sup>2</sup>C从机通信流程

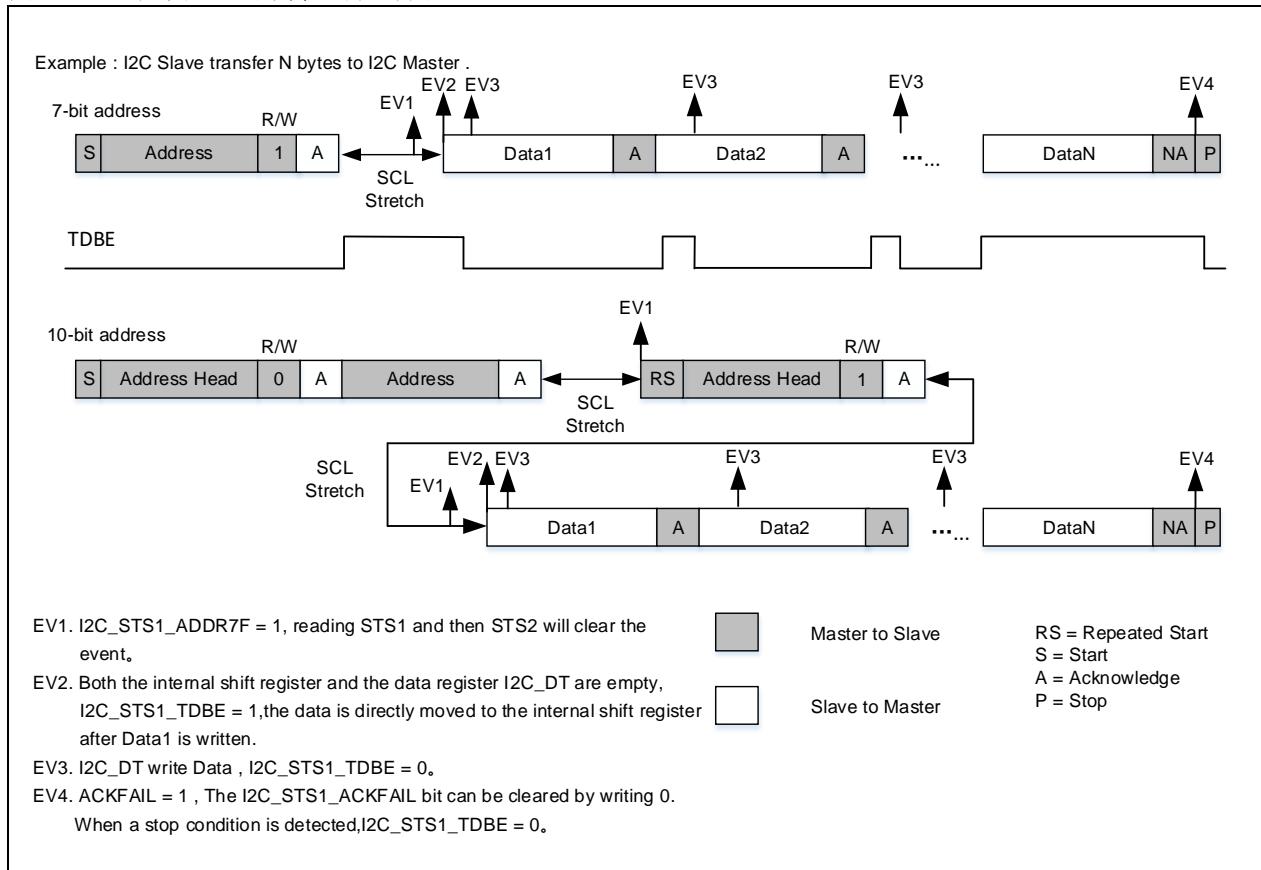
### 初始化

使能 I<sup>2</sup>C 外设时钟并配置控制寄存器 2 (I<sup>2</sup>C\_CTRL2) 中时钟相关寄存器确保正确的时序，接着等待 I<sup>2</sup>C 主机发送开始讯号。

### 发送器

从机发送数据主要有以下操作流程，初始化后软件可以参照以下步骤进行操作：

图 11-3 从发送器的传送序列图



### 7位地址模式：

1. 等待主机发送地址
2. EV1：成功匹配到地址 (ADDR7F=1)，从机将SCL总线拉低，软件先读取STS1，再读取STS2 清除ADDR7F位，此时进入发送阶段，DT寄存器和内部移位寄存器皆为空，硬件将TDBE位置1
3. EV2：向DT寄存器写入数据，此时数据会被立即送到移位寄存器并释放SCL总线，此时TDBE仍然为1
4. EV3：此时DT数据寄存器空，移位寄存器非空，向DT寄存器写入数据，此时TDBE清零
5. EV4：收到主机发送的ACKFAIL事件，此时ACKFAIL=1，向ACKFAIL写0清除该事件
6. 通信结束

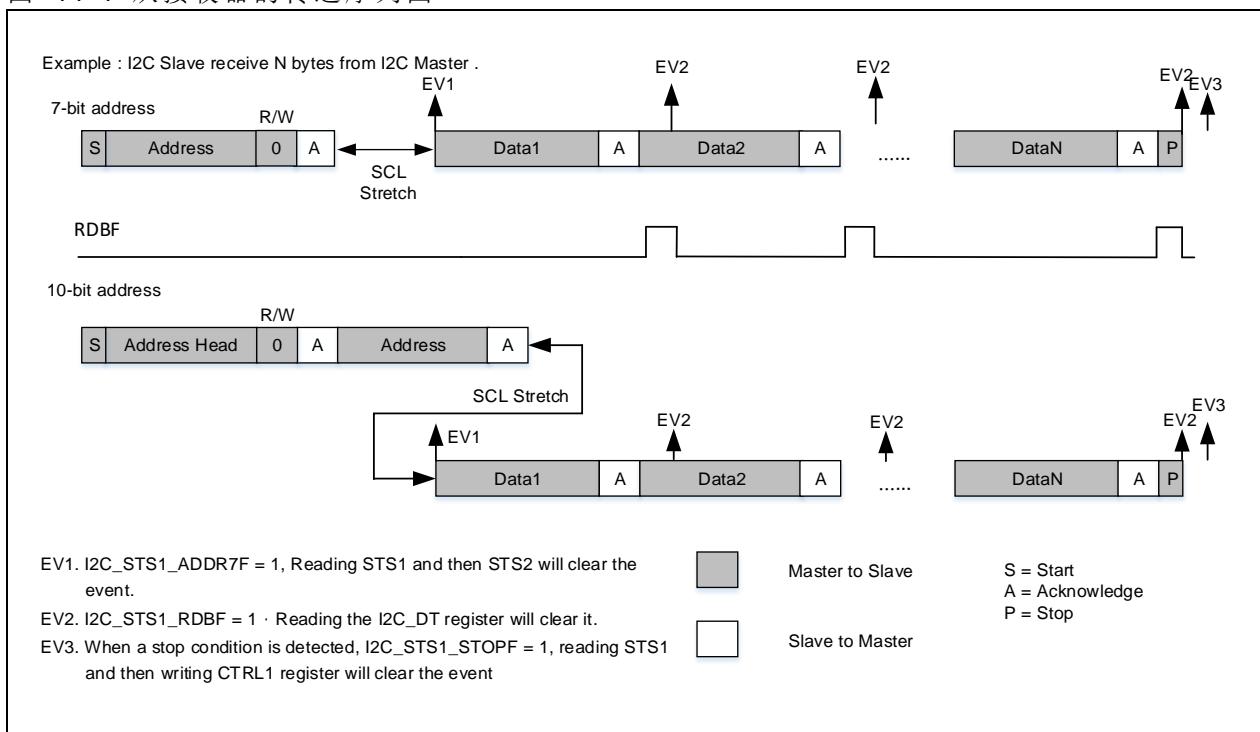
### 10位地址模式：

1. 等待主机发送地址
2. EV1：成功匹配到地址 (ADDR7F=1)，从机将SCL总线拉低，软件先读取STS1，再读取STS2 清除ADDR7F位，等待主机发送重复开始信号
3. EV1：成功匹配到地址 (ADDR7F=1)，软件先读取STS1，再读取STS2再次清除ADDR7F位，此时进入发送阶段，此时DT寄存器和内部移位寄存器皆为空，硬件将TDBE位置1
4. EV2：向DT寄存器写入数据，此时数据会被立即送到移位寄存器并释放SCL总线，此时TDBE仍然为1
5. EV3：此时DT数据寄存器空，移位寄存器非空，向DT寄存器写入数据，此时TDBE清零
6. EV4：收到主机发送的ACKFAIL事件，此时ACKFAIL=1，向ACKFAIL写0清除该事件
7. 通信结束

### 从接收器

从机接收数据主要有以下操作流程，初始化后软件可以参照以下步骤进行操作：

图 11-4 从接收器的传送序列图



### 7位地址模式：

1. 等待主机发送地址
2. EV1：成功匹配到地址 (ADDR7F=1)，从机将SCL总线拉低，软件可通过读取STS1在读取STS2清除ADDR7F位，此时从机释放SCL总线，进入接收阶段
3. 从机内部移位寄存器接收来自总在线的数据，并存入DT寄存器
4. EV2：在接收到字节后，RDBF位被置1，软件读取数据寄存器(I2C\_DT)，RDBF位被清0
5. EV3：收到主机发送的结束信号，STOPF=1，软件读取STS1，再写CTRL1寄存器清除该事件
6. 通信结束

### 10位地址模式：

1. 等待主机发送地址
2. EV1：成功匹配到地址 (ADDR7F=1)，从机将SCL总线拉低，软件先读取STS1，再读取STS2清除ADDR7F位，此时从机释放SCL总线进入接收阶段
3. 从机内部移位寄存器接收来自总在线的数据，并存入DT寄存器
4. EV2：在接收到字节后，RDBF位被置1，软件读取数据寄存器(I2C\_DT)，RDBF位被清0
5. EV3：收到主机发送的结束信号，STOPF=1，软件读取STS1，再写CTRL1寄存器清除该事件
6. 通信结束

## 11.4.2 I<sup>2</sup>C主机通信流程

### 主机模式初始化

1. 设置输入时钟以产生正确的时序（控制寄存器2 (I2C\_CTRL2) 中的CLKFREQ位）；
2. 设置I<sup>2</sup>C的通信速度（时钟控制寄存器(I2C\_CLKCTRL)）；
3. 设置总线最大上升时间 (I2C\_TMRISE 寄存器)；
4. 设置控制寄存器1 (I2C\_CTRL1)；
5. 启动外设，若设置GENSTART位在启动外设时会在总在线产生开始信号，设备会进入主机模式

### 从机地址发送

从机地址可分为 7 位和 10 位地址模式，主机会根据送出的地址最低位决定进入发送器模式或是接收器模式。

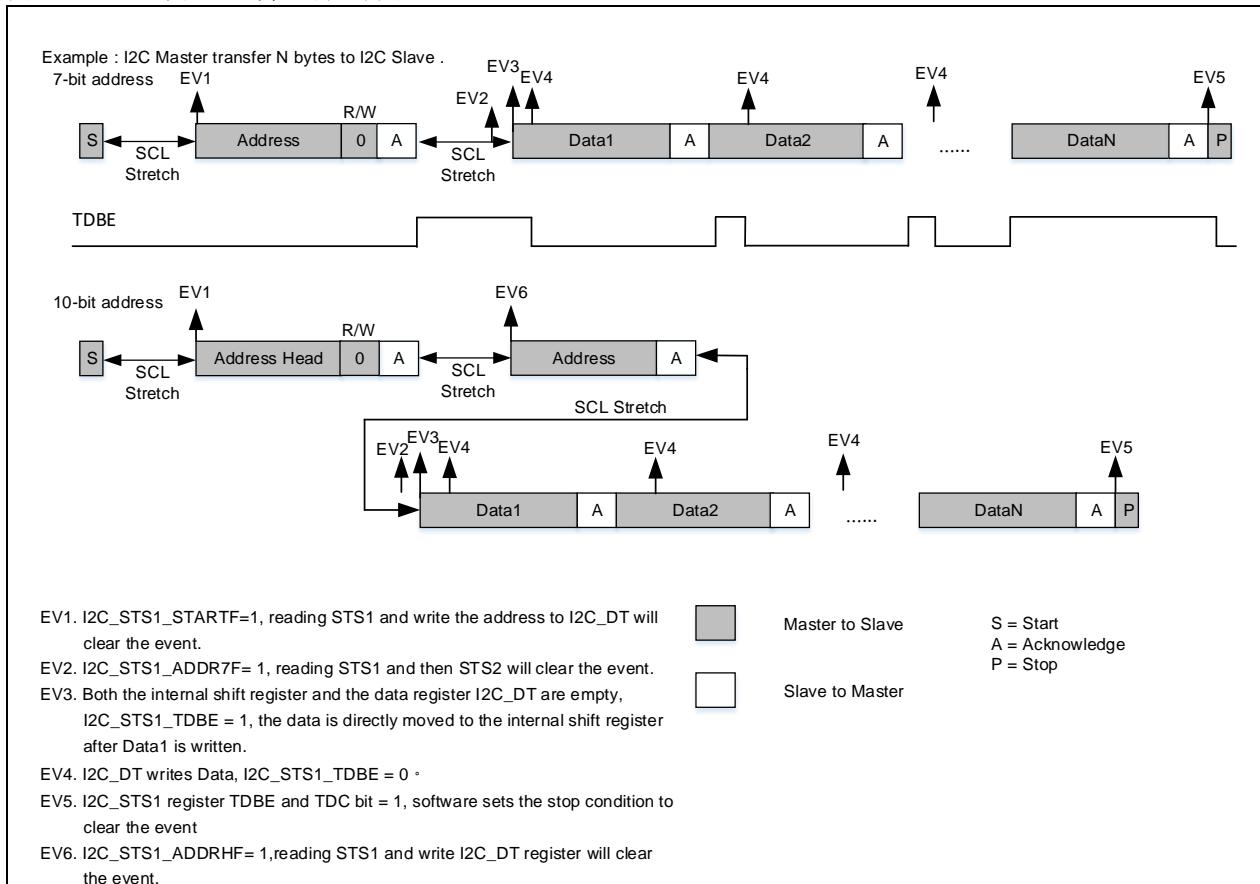
- 7位地址模式：
 

发送模式：发送的地址最低位为0时，进入发送器模式；  
接收模式：发送的地址最低位为1时，进入接收器模式。
- 10位地址模式：
 

发送模式：先发送从机地址头 0b11110xx0 (xx 为地址[9: 8])，再发送从机地址[7: 0]，主机进入发送器模式；  
接收模式：先发送从机地址头 0b11110xx0 (xx 为地址[9: 8])，再发送从机地址[7: 0]，再发送从机地址头 0b11110xx1 (xx 为地址[9: 8])，主机进入接收器模式。

### 主发送器

图 11-5 主发送器传送序列图



- 7位地址模式：
  1. 发送开始信号 (GENSTART=1)
  2. EV1: 开始信号产生完成 (STARTF=1)，软件先读取STS1，然后将地址写入DT寄存器
  3. EV2: 成功匹配到地址 (ADDR7F=1)，软件先读取STS1，再读取STS2清除ADDR7F位，此时主机进入发送阶段，DT寄存器和内部移位寄存器皆为空，硬件将TDDE位置1
  4. EV3: 向DT寄存器写入数据，此时数据会被立即送到移位寄存器并释放SCL总线，此时TDDE仍然为1
  5. EV4: 此时DT数据寄存器空，移位寄存器非空，向DT寄存器写入数据，此时TDDE清零
  6. TDDE位在倒数第二个字节发送完成后置起
  7. EV5: TDC=1，字节发送结束，主机发送结束信号 (STOPF=1)，硬件自动清除TDDE位和TDC位
  8. 通信结束
- 10位地址模式：
  1. 发送开始信号 (GENSTART=1)
  2. EV1: 开始信号产生完成，STARTF=1，软件先读取STS1，然后将地址写入DT寄存器
  3. EV6: 10位地址头序列已发送，软件可通过读取STS1再写入DT寄存器清除ADDRHF位
  4. EV2: 成功匹配到地址 (ADDR7F=1)，软件先读取STS1，再读取STS2清除ADDR7F位，此时主机进入发送阶段，DT寄存器和内部移位寄存器皆为空，硬件将TDDE位置1

5. EV3: 向DT寄存器写入数据, 此时数据会被立即送到移位寄存器并释放SCL总线, 此时TDBE仍然为1
6. EV4: 此时DT数据寄存器空, 移位寄存器非空, 向DT寄存器写入数据, 此时TDBE清零
7. TDBE位在倒数第二个字节发送完成后置起
8. EV5: TDC=1, 字节发送结束, 主机发送结束信号 (STOPF=1), 硬件自动清除TDBE位和TDC位
9. 通信结束

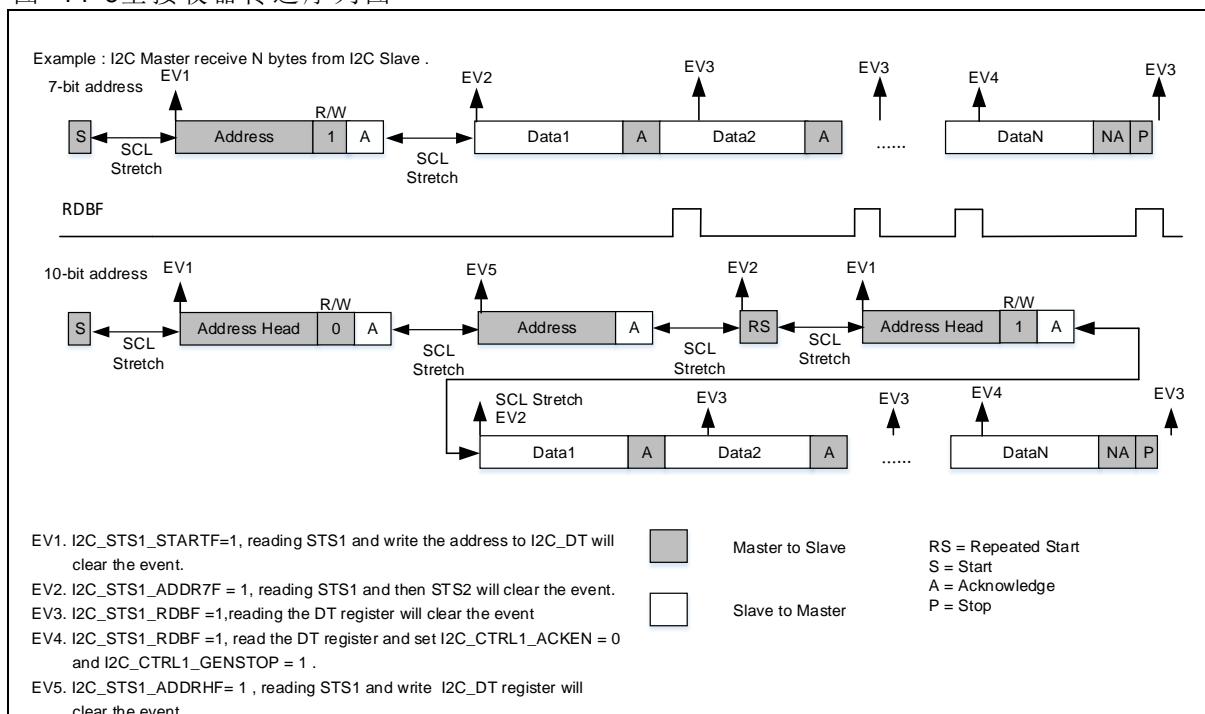
### 主接收器

主机接收数据可依 I<sup>2</sup>C 中断优先级分为几种情况:

#### 1. I<sup>2</sup>C中断为最高优先级

- 在读倒数第二个字节后需清除控制寄存器 1 (I2C\_CTRL1) 的 ACKEN 位并设置同一寄存器的 GENSTOP 位以产生结束信号。
- 若只接收一个字节时, 需在清除 ADDR7F 标志后设置控制寄存器 1 (I2C\_CTRL1) 的 ACKEN 和 GENSTOP 位。
- 接收到字节后硬件会将 I2C\_STS1\_RDBF 位置 1, 在软件读数据寄存器(I2C\_DT)后会被清 0。

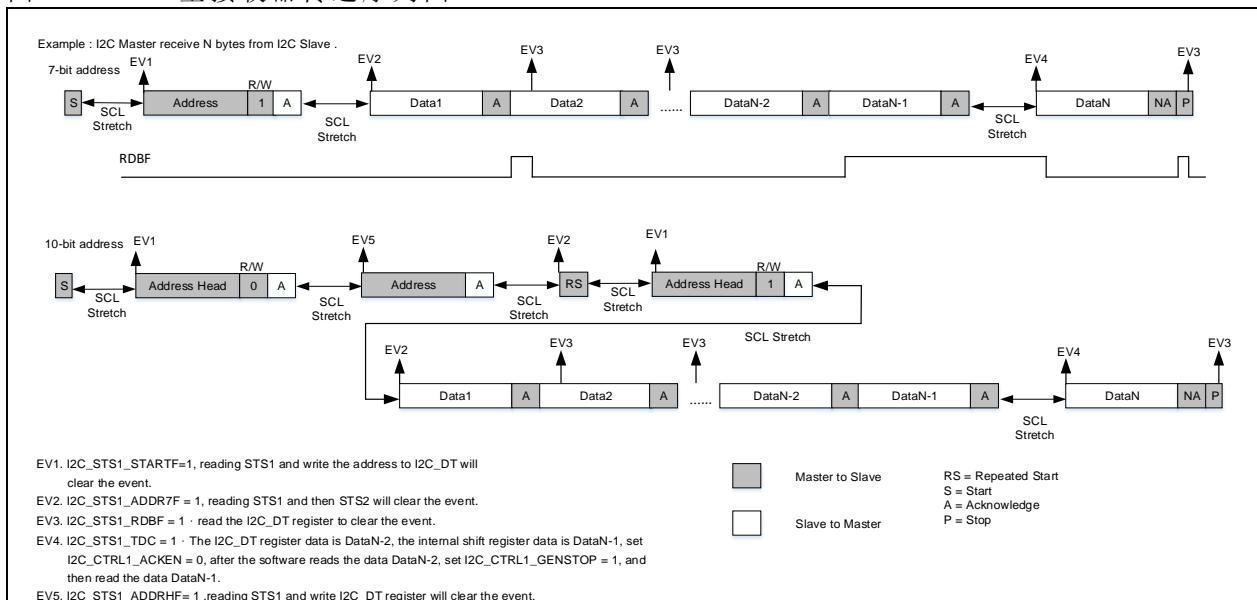
图 11-6 主接收器传送序列图



- 7 位地址模式:
  1. 发送开始信号 (GENSTART=1)
  2. EV1: 开始信号产生完成 (STARTF=1), 软件先读取STS1, 然后将地址写入DT寄存器
  3. EV2: 成功匹配到地址 (ADDR7F=1), 软件先读取STS1, 再读取STS2清除ADDR7F位, 此时主机进入接收阶段
  4. EV3: 在接收到字节后, RDBF位被置1, 软件读取数据寄存器(I2C\_DT), RDBF位被清0
  5. EV4: 接收完倒数第二个字节后, 软件需立即将ACKEN位清0, GENSTOP位置1
  6. EV3: 在接收到字节后, RDBF位被置1, 软件读取数据寄存器(I2C\_DT), RDBF位被清0
  7. 通信结束
- 10 位地址模式:
  1. 发送开始信号 (GENSTART=1)
  2. EV1: 开始信号产生完成 (STARTF=1), 软件先读取STS1, 然后将地址写入DT寄存器
  3. EV5: 10位地址头序列已发送, 软件可通过读取STS1再写入DT寄存器清除ADDRHF位
  4. EV2: 成功匹配到地址 (ADDR7F=1), 软件先读取STS1, 再读取STS2清除ADDR7F位, 主机发送重复开始信号 (GENSTART=1)
  5. EV1: 重复开始信号产生完成 (STARTF=1), 软件先读取STS1, 然后将地址写入DT寄存器

6. EV2: 成功匹配到地址 (ADDR7F=1), 软件先读取STS1, 再读取STS2清除ADDR7F位, 此时主机进入接收阶段
  7. EV3: 在接收到字节后, RDBF位被置1, 软件读取数据寄存器(I2C\_DT), RDBF位被清0
  8. EV4: 接收完倒数第二个字节后, 软件需立即将ACKEN位清0, GENSTOP位置1
  9. EV3: 在接收到字节后, RDBF位被置1, 软件读取数据寄存器(I2C\_DT), RDBF位被清0
  10. 通信结束
2. **I<sup>2</sup>C中断非最高优先级且要接收的字节数大于2**
- 在接收到倒数第三个字节(N-2)时不进行读取, 待收到倒数第二个字节(N-1)时, 清除控制寄存器1(I2C\_CTRL1)的ACKEN位, 接着读取倒数第三个字节(N-2), 设置控制寄存器1(I2C\_CTRL1)的GENSTOP位后读取倒数第二个字节(N-1), 接着总线开始接收最后字节。

图 11-7 N>2主接收器传送序列图



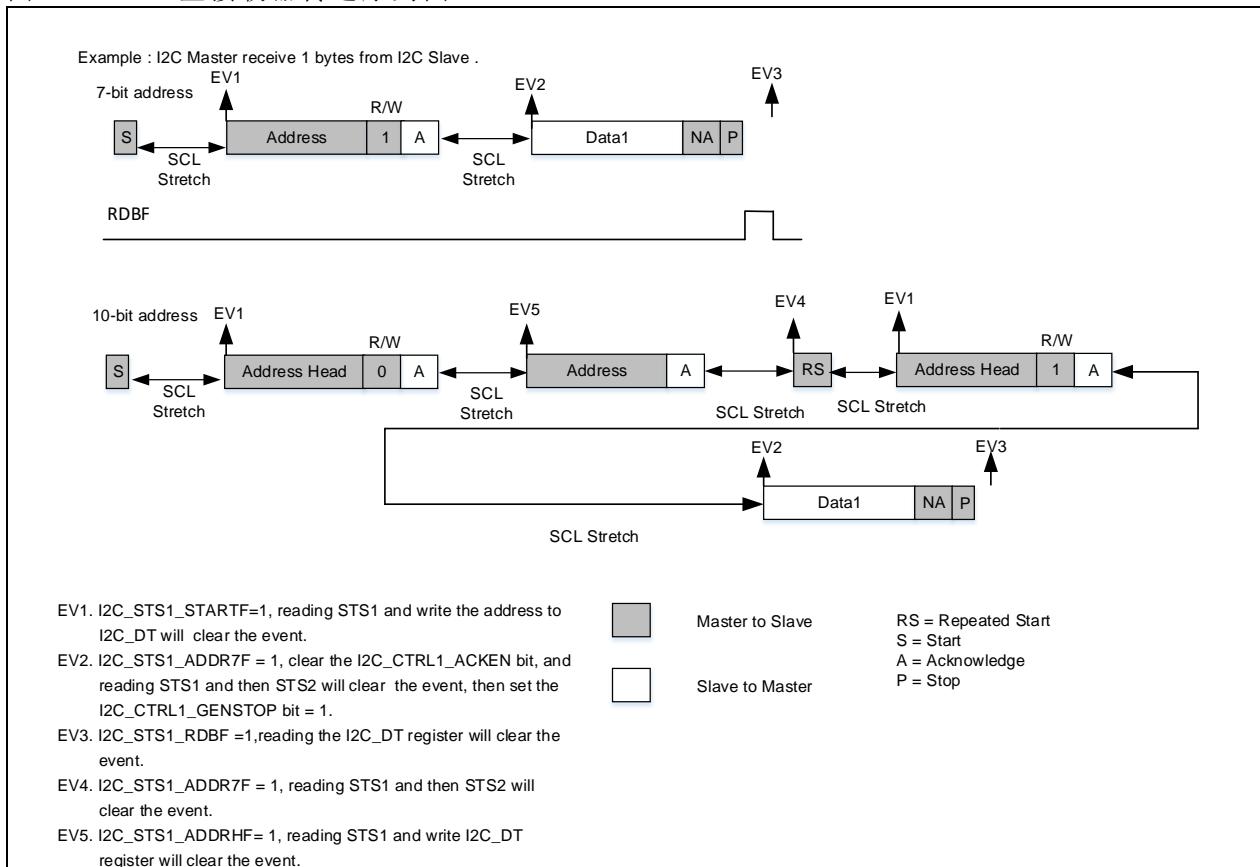
- 7位地址模式:
  1. 发送开始信号 (GENSTART=1)
  2. EV1: 开始信号产生完成 (STARTF=1), 软件先读取STS1, 然后将地址写入DT寄存器
  3. EV2: 成功匹配到地址 (ADDR7F=1), 软件先读取STS1, 再读取STS2清除ADDR7F位, 此时主机进入接收阶段
  4. EV3: 在接收到字节后, RDBF位被置1, 软件读取数据寄存器(I2C\_DT), RDBF位被清0
  5. EV4: TDC=1, 数据寄存器(I2C\_DT)内容为N-2, 移位寄存器内容为数据N-1, 软件将ACKEN位置0并读取读数据N-2, 接着设置GENSTOP=1, 然后读数据N-1
  6. EV3: 在接收到字节后, RDBF位被置1, 软件读取数据寄存器(I2C\_DT), RDBF位被清0
  7. 通信结束
- 10位地址模式:
  1. 发送开始信号 (GENSTART=1)
  2. EV1: 开始信号产生完成 (STARTF=1), 软件先读取STS1, 然后将地址写入DT寄存器
  3. EV5: 10位地址头序列已发送, 软件可通过读取STS1再写入DT寄存器清除ADDRHF位
  4. EV2: 成功匹配到地址 (ADDR7F=1), 软件先读取STS1, 再读取STS2清除ADDR7F位, 主机发送重复开始信号 (GENSTART=1)
  5. EV1: 重复开始信号产生完成, STARTF=1, 软件先读取STS1, 然后将地址写入DT寄存器
  6. EV2: 成功匹配到地址 (ADDR7F=1), 软件先读取STS1, 再读取STS2清除ADDR7F位, 此时主机进入接收阶段
  7. EV3: 在接收到字节后, RDBF位被置1, 软件读取数据寄存器(I2C\_DT), RDBF位被清0
  8. EV4: TDC=1, 数据寄存器(I2C\_DT)内容为N-2, 移位寄存器内容为数据N-1, 软件将ACKEN位置0并读取读数据N-2, 接着设置GENSTOP=1, 然后读数据N-1
  9. EV3: 在接收到字节后, RDBF位被置1, 软件读取数据寄存器(I2C\_DT), RDBF位被清0

## 10. 通信结束

### 3. I<sup>2</sup>C中断非最高优先级且要接收的字节数等于2

- 在接收数据前设置控制寄存器1(I2C\_CTRL1)的MACKCTRL位，待地址匹配后，先清除ACKEN位，后清除ADDR7F位，待TDC位置1后设置控制寄存器1(I2C\_CTRL1)的GENSTOP位，接着读取DT寄存器。

图 11-8 N=2主接收器传送序列图



### ● 7位地址模式：

1. 设置控制寄存器1(I2C\_CTRL1)的MACKCTRL=1
2. 发送开始信号(GENSTART=1)
3. EV1：开始信号产生完成(STARTF=1)，软件先读取STS1，然后将地址写入DT寄存器
4. EV2：成功匹配到地址(ADDR7F=1)，首先清除ACKEN位，然后先读取STS1，再读取STS2清除ADDR7F位，此时主机进入接收阶段
5. EV2：TDC=1，接着设置GENSTOP=1然后读数据寄存器(I2C\_DT)两次
6. 通信结束

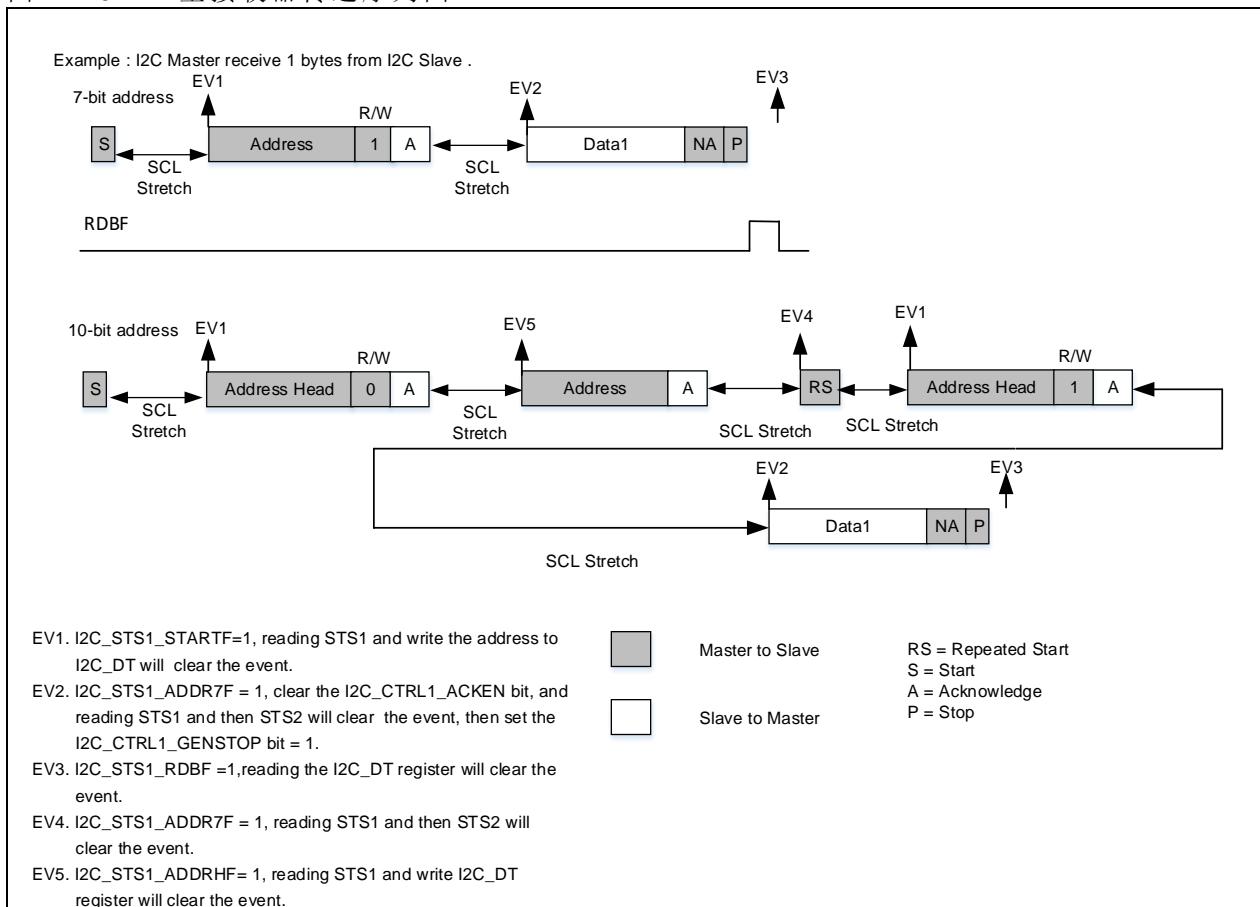
### ● 10位地址模式：

1. 设置控制寄存器1(I2C\_CTRL1)的MACKCTRL=1
2. 发送开始信号(GENSTART=1)
3. EV1：开始信号产生完成(STARTF=1)，软件先读取STS1，然后将地址写入DT寄存器
4. EV4：10位地址头序列已发送，软件可通过读取STS1再写入DT寄存器清除ADDRHF位
5. EV2：成功匹配到地址(ADDR7F=1)，软件先读取STS1，再读取STS2清除ADDR7F位，主机发送重复开始信号(GENSTART=1)
6. EV1：重复开始信号产生完成，STARTF=1，软件先读取STS1，然后将地址写入DT寄存器
7. EV2：成功匹配到地址(ADDR7F=1)，首先清除ACKEN位，然后先读取STS1，再读取STS2清除ADDR7F位，此时主机进入接收阶段
8. EV3：TDC=1，接着设置GENSTOP=1然后读数据寄存器(I2C\_DT)两次
9. 通信结束

### 4. I<sup>2</sup>C中断非最高优先级且要接收的字节数等于1

- 待地址匹配后，先清除ACKEN位，后清除ADDR7F位，接着设置控制寄存器1(I2C\_CTRL1)的GENSTOP位，待RDBF位置1后读取DT寄存器内字节。

主机接收数据主要有以下操作流程，主机模式初始化后软件可以参照以下步骤进行操作：  
图 11-9 N=1主接收器传送序列图



- 7位地址模式：
  1. 发送开始信号 (GENSTART=1)
  2. EV1: 开始信号产生完成 (STARTF=1)，软件先读取STS1，然后将地址写入DT寄存器
  3. EV2: 成功匹配到地址 (ADDR7F=1)，首先清除ACKEN位，然后先读取STS1，再读取STS2 清除ADDR7F位，接着设置GENSTOP=1，此时主机进入接收阶段
  4. EV3: RDBF=1，读取数据寄存器(I2C\_DT)，RDBF位被清0
  5. 通信结束
- 10位地址模式：
  1. 发送开始信号 (GENSTART=1)
  2. EV1: 开始信号产生完成 (STARTF=1)，软件先读取STS1，然后将地址写入DT寄存器
  3. EV5: 10位地址头序列已发送，软件可通过读取STS1再写入DT寄存器清除ADDRHF位
  4. EV4: 成功匹配到地址 (ADDR7F=1)，软件先读取STS1，再读取STS2清除ADDR7F位，机发送重复开始信号 (GENSTART=1)
  5. EV1: 重复开始信号产生完成，STARTF=1，软件先读取STS1，然后将地址写入DT寄存器
  6. EV2: 成功匹配到地址 (ADDR7F=1)，首先清除ACKEN位，然后先读取STS1，再读取STS2 清除ADDR7F位，接着设置GENSTOP=1，此时主机进入接收阶段
  7. EV3: RDBF=1，读取数据寄存器(I2C\_DT)，RDBF位被清0
  8. 通信结束

### 11.4.3 利用DMA传输

I<sup>2</sup>C 可以使用 DMA 进行数据传输，可通过使能传输完成中断位产生中断，当利用 DMA 进行传输时，控制寄存器 2 (I2C\_CTRL2) 的 DATAIEN 位需设为 0，以下说明软件利用 DMA 进行数据传输的操作流程。

#### DMA发送

1. 设置外设地址 (DMA通道x外设地址寄存器(DMA\_CxPADDR)=数据寄存器(I2C\_DT)地址)
2. 设置数据存储地址 (DMA通道x存储器地址寄存器 (DMA\_CxMADDR) =数据存储地址)

3. 设置传输方向为内存到外设 (DMA\_CHCTRL的DTD=1)
4. 设置传输字节数 (DMA通道x数据传输量寄存器 (DMA\_CxDTCNT))
5. 设置DMA通道的其他配置, 例如: 优先级、存储器数据宽度、外设数据宽度、中断等 (DMA\_CHCTRL)
6. 使能DMA通道 (DMA通道x配置寄存器 (DMA\_CxCTRL) 的CHEN=1)。
7. 使能I<sup>2</sup>C DMA请求 (控制寄存器2 (I2C\_CTRL2) 的DMAEN=1), 当状态寄存器1 (I2C\_STS1) 的TDBE位被置1时, DMA将数据从内存地址传输到数据寄存器(I2C\_DT)
8. 等待传输字节数DMA通道x数据传输量寄存器 (DMA\_CxDTCNT) =0时, 数据传输完成, (可以通过DMA传输完成中断来等待)。
9. 主机发送模式: 等待TDC标志置1, 产生STOP条件, 传输完成。  
从机发送模式: 等待ACKFAIL标志置1, 清除ACKFAIL标志, 传输完成。

#### DMA 接收

1. 设置外设地址 (DMA通道x外设地址寄存器 (DMA\_CxPADDR)=数据寄存器(I2C\_DT)地址)
2. 设置数据存储地址 (DMA通道x存储器地址寄存器 (DMA\_CxMADDR) =数据存储地址)
3. 设置传输方向为外设到内存 (DMA\_CHCTRL的DTD=0)
4. 设置传输字节数 (DMA通道x数据传输量寄存器 (DMA\_CxDTCNT))
5. 设置DMA通道的其他配置, 例如: 优先级、存储器数据宽度、外设数据宽度、中断等 (DMA\_CHCTRL)
6. 使能DMA通道 (DMA通道x配置寄存器 (DMA\_CxCTRL) 的CHEN=1)。
7. 使能I<sup>2</sup>C DMA请求 (控制寄存器2 (I2C\_CTRL2) 的DMAEN=1), 当状态寄存器1 (I2C\_STS1) 的RDBF位被置1时, DMA将数据从数据寄存器(I2C\_DT)传输到数据存储地址。
8. 等待传输字节数DMA\_TCNTx=0时, 数据传输完成, (可以通过DMA传输完成中断来等待)。
9. 主机接收模式: 清除ACKFAIL标志, 产生STOP条件, 传输完成(当传输数据>=2, 且DMAEND=1时, 当数据传输完成了之后 (DMA\_CxDTCNT=0) , 将会自动产生一个NACK)。  
从机接收模式: 等待STOPF标志置1, 清除STOPF标志, 传输完成。

### 11.4.4 SMBus

SMBus 即系统管理总线是一双线制总线, 基于 I<sup>2</sup>C 的操作原理, 系统中各设备之间通过 SMBus 总线传送和接收讯息, 通过 SMBus 总线, 设备可以提供制造商信息, 告诉系统型号, 报告不同类型错误, 接受控制参数等。关于 SMBus 更加详细的信息请参考 SMBus2.0 协议。

#### SMBus 和 I<sup>2</sup>C 的差异

1. SMBus需维持最低10kHz以上的运作频率主要为了管理监控, 只要在保持一定传速运作的情况下加入参数, 就可轻松获知总线目前是否处于闲置 (Idle) 中, 省去逐一侦测传输过程中的停断 (STOP) 信号, 或持续保有停断侦测并辅以额外参数侦测, I<sup>2</sup>C则无
2. SMBus传输速度从最小10kHz到最大100kHz, I<sup>2</sup>C则是无最小传输速度, 根据不同模式有不同的最大传输速度, 分为标准模式(100kHz)和快速模式(400kHz)
3. SMBus对接口被重置(Reset)后的恢复时间(Timeout)是35ms, I<sup>2</sup>C则无时间限制

#### SMBus 使用流程

1. 将I<sup>2</sup>C接口设置SMBus模式, 控制寄存器1 (I2C\_CTRL1) 的PERMODE=1
2. 选择SMBus模式:

SMBMODE=1: SMBus主机  
SMBMODE=0: SMBus设备

3. 其他配置和I<sup>2</sup>C使用配置一样

各种 SMBus 协议需要由软件来实现, I<sup>2</sup>C 接口只提供了这些协议的地址识别。

#### SMBus 地址解析协议(ARP)

通过 ARP 协议可以给总线上的设备动态的分配一个唯一的新地址, 解决地址冲突问题。关于 ARP 协议更详细的信息请参考 SMBus2.0 协议。

通过使能 ARPEN 位, 可以使能 I<sup>2</sup>C 接口对设备默认地址 (0b1100001x) 的识别, 但是像唯一设备标识 (UDID) 以及具体的协议实现过程, 需要由软件来处理。

#### SMBus 主机通知协议

通过 SMBus 主机通知协议, 可让从设备发送数据到主设备, 例如从机可以通过此协议通知主机进行

ARP。关于 SMBus 主机通知协议更详细的信息请参考 SMBus2.0 协议。

当使能了 ARP 模式 (ARPEN=1) 以及在主机模式 (SMBMODE=1) 下, I<sup>2</sup>C 接口使能对主机默认地址 (0b0001000x) 的识别。

#### SMBus 提醒协议 (SMBus Alert)

SMBALERT 是一个可选信号, 连接主机和从机的 ALERT 引脚, 用于从机通知主机访问从机, SMBALERT 是一个线与信号。关于 SMBus 提醒协议更详细的信息请参考 SMBus2.0 协议。

操作流程如下:

##### SMBus 主机

1. 启用 SMBus 提醒模式 (SMBALERT=1)
2. 根据实际需求启用 ALERT 中断
3. 当 ALERT 引脚上产生了提醒事件时 (ALERT 引脚电平由高变低)
4. 如果使能了中断, 主机将产生 ALERT 中断
5. 主机处理该中断并向从机发送提醒响应地址 ARA (Alert Response Address) 地址 (0001100x), 访问所有设备, 获取从机地址, 只有那些将 SMBALERT 拉低的设备才会应答
6. 主机通过获取到的从机地址进行下一步操作。

##### SMBus 从机

1. 产生提醒事件, ALERT 引脚由高变低 (SMBALERT=1), 此时从机响应 ARA (Alert Response Address) 地址 (0001100x)
2. 根据实际需求启用 ALERT 中断 (当收到 ARA 地址时会产生中断)
3. 等待主机通过发送 ARA 地址获取从机地址
4. 上报自己的地址, 如果发生了仲裁丢失, 继续等待
5. 地址上报成功, 释放 ALERT 引脚 (SMBALERT=0)

#### 包错误校验(PEC)

包错误校验(PEC)用于保证数据传输的正确性和完整性, 使用 CRC-8 进行校验, 多项式为:

$$C(x) = x^8 + x^2 + x + 1$$

当 PECEN=1 时启动 PEC 计算, 检验数据包括地址以及数据, 当在仲裁丢失时 PEC 计算会失效。

PEC 发送:

- 正常模式: 最后一次 TDBE 事件后设置 PECTRA=1, 让 PEC 在最后一个字节后被发送
- DMA 模式: 在最后一个字节传输完成后自动发送 PEC, 例如: 传输的数据为 8 个那么设置 DMA\_TCNTx=8

PEC 接收:

- 正常模式: 最后一个 RDBF 事件后设置 PECTRA 位, PECTRA 位必须在接收当前字节的 ACK 脉冲之前被设置
- DMA 模式: 接收时会自动把最后一个字节当作 PECVAL 并检查, 例如: 传输的数据为 8 个那么设置 DMA\_TCNTx=9

在接收模式下, 当 PEC 校验失败时, 将产生一个 NACK。

### 11.4.5 I<sup>2</sup>C 中断请求

下表列出了所有的 I<sup>2</sup>C 中断请求。

中断事件	事件标志	使能位
已发送起始条件(主机)	STARTF	EVTIEN
地址已发送(主机)或 地址匹配(从机)	ADDR7F	
10 位地址头已发送(主机)	ADDRHF	
数据传输完成	TDC	
收到停止条件(从机)	STOPF	
发送缓冲区空	TDBE	EVTIEN 和 DATAIEN
接收缓冲区非空	RDBF	

SMBus 提醒	ALERTF	ERRIEN
超时错误	TMOUT	
PEC 错误	PECERR	
过载/欠载	OUF	
应答失败	ACKFAIL	
仲裁丢失	ARLOST	
总线错误	BUSERR	

#### 11.4.6 I<sup>2</sup>C 调试模式

当微控制器进入调试模式 (Cortex®-M4 核心处于停止状态) 时，根据 DEBUG 模块中的 I2Cx\_SMBUS\_TIMEOUT 配置位，SMBUS 超时控制或者继续正常工作或者可以停止。

#### 11.5 I<sup>2</sup>C 寄存器描述

必须以字（32 位）的方式操作这些外设寄存器。

表 11-1 I<sup>2</sup>C 寄存器地址映像和复位值

寄存器简称	基址偏移量	复位值
I2C_CTRL1	0x00	0x0000
I2C_CTRL2	0x04	0x0000
I2C_OADDR1	0x08	0x0000
I2C_OADDR2	0x0C	0x0000
I2C_DT	0x10	0x0000
I2C_STS1	0x14	0x0000
I2C_STS2	0x18	0x0000
I2C_CLKCTRL	0x1C	0x0000
I2C_TMRISE	0x20	0x0002

#### 11.5.1 控制寄存器1(I2C\_CTRL1)

域	简称	复位值	类型	功能
位 15	RESET	0x0	rw	I <sup>2</sup> C 外设复位 (I <sup>2</sup> C peripheral reset) 0: 不复位; 1: 复位。 注: 该位可以用于 BUSYF 位为'1', 在总线上又没有检测到停止条件时。
位 14	保留	0x0	rstd	保持默认值。
位 13	SMBALERT	0x0	rw	SMBus 提醒引脚设置 (SMBus alert pin set) 软件可以使其置 1 或清为 0; 当 I2CEN=0 时, 由硬件清除。 0: 置高; 1: 置低。
位 12	PECTEN	0x0	rw	请求 PEC 传输使能 (Request PEC transmission enable) 软件可以使其置 1 或清为 0; 当传送 PECTEN 后, 开始或结束信号时, 由硬件清除。 0: 停止传输; 1: 启动传输。
位 11	MACKCTRL	0x0	rw	主机接收模式应答控制 (Master receiving mode acknowledge control) 0: ACKEN 位效果作用于当前传字节;

				1: ACKEN 位效果作用于第二个传输字节。 该位只在主机接收两个字节模式下使用，目的是为了让主机及时的回 ACK。
位 10	ACKEN	0x0	rw	应答使能(Acknowledge enable) 软件可以使其置 1 或清为 0; 0: 关闭, 不发送应答; 1: 开启。
位 9	GENSTOP	0x0	rw	产生停止条件 (Generate stop condition) 软件可以使其置 1 或清为 0; 或当检测到结束信号时, 由硬件清除; 当检测到超时错误时, 硬件将其置位。 0: 未产生; 1: 产生。 如果在从模式下当设置了此位, 从机将释放 SCL 和 SDA 总线。
位 8	GENSTART	0x0	rw	产生起始条件 (Generate start condition) 软件可以使其置 1 或清为 0; 或当起始条件发出后, 由硬件清除。 0: 未产生; 1: 产生。
位 7	STRETCH	0x0	rw	时钟延展模式 (Clock stretching mode) 0: 开启; 1: 关闭。 注: 只在从机模式下有效。
位 6	GCAEN	0x0	rw	广播地址使能 (General call address enable) 0: 开启; 1: 关闭。
位 5	PECEN	0x0	rw	PEC 计算使能 (PEC calculation enable) 0: 关闭; 1: 开启。
位 4	ARPEN	0x0	rw	SMBus ARP 协议使能 (SMBus address resolution protocol enable) 0: 关闭; 1: 开启。 SMBus 主机: 响应主机地址 0001000x; SMBus 设备: 响应设备默认地址 0001100x。
位 3	SMBMODE	0x0	rw	SMBus 设备模式 (SMBus device mode) 0: SMBus 设备; 1: SMBus 主机。
位 2	保留	0x0	resd	硬件强制为 0。
位 1	PERMODE	0x0	rw	I <sup>2</sup> C 外设模式 (I <sup>2</sup> C peripheral mode) 0: I <sup>2</sup> C 模式; 1: SMBus 模式。
位 0	I2CEN	0x0	rw	I <sup>2</sup> C 外设使能 (I <sup>2</sup> C peripheral enable) 0: 关闭; 1: 开启。 在通讯结束后发生 I2CEN=0, 所有的位被清除。 在主模式下, 通讯结束之前, 绝不能清除该位。

**注意:** 当 GENSTART、GENSTOP 或 PECTEN 设置后, 软件应该在相应位被硬件清零后控制寄存器 1 (I2C\_CTRL1), 否则有可能产生第二次 GENSTART、GENSTOP 或 PECTEN 请求。

### 11.5.2 控制寄存器2(I2C\_CTRL2)

域	简称	复位值	类型	功能
位 15: 13	保留	0x0	resd	硬件强制为 0
位 12	DMAEND	0x0	rw	DMA 传输结束指示 (DMA transfer end indication) 0: 将要传输不是最后一笔数据; 1: 将要传输最后一笔数据。
位 11	DMAEN	0x0	rw	启动 DMA 传输 (DMA transfer enable) 0: 关闭; 1: 开启。
位 10	DATAIEN	0x0	rw	数据传输中断使能 (Data transmission interrupt enable) TDBE 或 RDBF 位置 1 时产生中断 0: 关闭; 1: 开启。
位 9	EVTIEN	0x0	rw	事件中断使能 (Event interrupt enable) 在下列条件下, 将产生该中断: - STARTF = 1 (主模式) - ADDR7F = 1 (主/从模式) - ADDRHF = 1 (主模式) - STOPF = 1 (从模式) - TDC = 1, 但是没有 TDBE 或 RDBF 事件 - 如果 DATAIEN = 1, TDBE 事件为 1 - 如果 DATAIEN = 1, RDBF 事件为 1
位 8	ERRIEN	0x0	rw	错误中断使能 (Error interrupt enable) 0: 关闭; 1: 开启。 在下列条件下, 将产生该中断: - BUSERR = 1 - ARLOST = 1 - ACKFAIL = 1 - OVER = 1 - PECERR = 1 - TMOUT = 1 - ALERTF = 1
位 7: 0	CLKFREQ	0x00	rw	I <sup>2</sup> C 输入时钟频率 (I <sup>2</sup> C input clock frequency) 必须设置正确的输入时钟频率以产生正确的时序, 允许的范围在 2~120MHz 之间: 范围 2~120MHz。 2: 2MHz; 3: 3MHz; ..... 120: 120MHz。

### 11.5.3 自身地址寄存器1(I2C\_OADDR1)

域	简称	复位值	类型	功能
位 15	ADDR1MODE	0x0	rw	地址模式 (Address mode) 0: 7 位地址; 1: 10 位地址。
位 14: 10	保留	0x00	resd	保持默认值。
位 9: 0	ADDR1	0x000	rw	本机地址 1 (Own address) 当在 7 位地址模式下时 BIT[9: 8]不关心。

### 11.5.4 自身地址寄存器2(I2C\_OADDR2)

域	简称	复位值	类型	功能
位 15: 8	保留	0x00	resd	保持默认值。
位 7: 1	ADDR2	0x00	rw	本机地址 2 (Own address 2) 7 位地址。
位 0	ADDR2EN	0x0	rw	本机地址 2 使能 (Own address 2 enable)

0: 在 7 位地址模式下, 只有 OADDR1 被识别;  
1: 在 7 位地址模式下, OADDR1 和 OADDR2 都被识别。

### 11.5.5 数据寄存器(I2C\_DT)

域	简称	复位值	类型	功能
位 15: 8	保留	0x00	resd	保持默认值。
位 7: 0	DT[7: 0]	0x00	rw	用于存放接收到或待发送的数据 发送器模式: 当写一个字节至 DT 寄存器时, 自动启动数据传输。一旦传输开始(TDE=1), 如果能及时把下一个需传输的数据写入 DT 寄存器, I2C 模块将保持连续的数据流。 接收器模式: 接收到的字节被拷贝到 DT 寄存器(RDNE=1)。在接收到下一个字(RDNE=1)之前读出数据寄存器, 即可实现连续的数据传送。 注: 如果在处理 ACK 脉冲时发生 ARLOST 事件, 接收到的字节不会被拷贝到数据寄存器里, 因此不能读到它。

### 11.5.6 状态寄存器1(I2C\_STS1)

域	简称	复位值	类型	功能
位 15	ALERTF	0x0	rw0c	SMBus 提醒标志 (SMBus alert flag) 在 SMBus 主机模式下: 0: 未收到; 1: 收到。 SMBus 从机: 指示设备默认地址接收 (0001100x) 0: 未收到; 1: 收到。 软件可以使其清为 0; 当 I2CEN=0 时, 由硬件清除。
位 14	TMOUT	0x0	rw0c	SMBus 超时标志 (SMBus timeout flag) 0: 无超时错误; 1: 超时。 软件可以使其清为 0; 当 I2CEN=0 时, 由硬件清除。 注: 这个功能仅在 SMBUS 模式下有效
位 13	保留	0x0	resd	保持默认值。
位 12	PECERR	0x0	rw0c	PEC 接收错误标志 (PEC receive error flag) 0: 正确; 1: 错误。 软件可以使其清为 0。
位 11	OUF	0x0	rw0c	溢出标志 (Overload / underload flag) 当传输方向为发送数据时: 0: 正常; 1: 欠载。 当传输方向为接收数据时: 0: 正常; 1: 过载。 软件可以使其清为 0; 当 I2CEN=0 时, 由硬件清除。
位 10	ACKFAIL	0x0	rw0c	应答失败标志 (Acknowledge failure flag) 0: 正常; 1: 失败。 当没有返回应答时, 硬件将置该位为'1' 软件可以使其清为 0; 当 I2CEN=0 时, 由硬件清除。
位 9	ARLOST	0x0	rw0c	仲裁丢失标志 (Arbitration lost flag) 0: 正常; 1: 仲裁丢失。 软件可以使其清为 0; 当 I2CEN=0 时, 由硬件清除。 在 ARLOST 事件之后, I <sup>2</sup> C 接口自动切换回从模式。
位 8	BUSERR	0x0	rw0c	总线错误标志 (Bus error flag) 0: 正常; 1: 错误。 当接口检测到错误的起始或停止条件, 硬件将该位置'1'。

				软件可以使其清为 0; 当 I2CEN=0 时, 由硬件清除 发送缓冲器空标志 (Transmit data buffer empty flag) 0: 数据正在从数据寄存器 (DT) 发送到移位寄存器, 数据寄存器还装着数据; 1: 数据已经从数据寄存器 (DT) 发送到移位寄存器, 数据寄存器空。 当 DT 为空的时候, 该标志值起, 向 DT 写数据时, 此标志清除。 注: 在写入第 1 个要发送的数据后, 或设置了 TDC 时写入数据, 都不能清除 TDBE 位, 这是因为数据寄存器仍然为空。
位 7	TDBE	0x0	ro	
位 6	RDBF	0x0	ro	接收数据缓冲器满标志 (Receive data buffer full flag) 0: 数据寄存器 (DT) 未接收到数据; 1: 数据寄存器 (DT) 接收到数据。 读取 DT 寄存器时, 此标志清除。 在发生 ARLOST 事件时, RDBF 不被置位。
位 5	保留	0x0	resd	保持默认值。
位 4	STOPF	0x0	ro	停止条件产生完成标志 (Stop condition generation complete flag) 0: 未产生; 1: 已产生。 当 ACKEN=1, 从设备在总线上检测到停止条件时, 硬件将该位置'1' 先读取 STS1 寄存器, 然后写 CTRL1 寄存器清除标志。
位 3	ADDRHF	0x0	ro	主机 9~8 位地址头匹配标志 (master 9~8 bit address header match flag) 0: 未匹配; 1: 已匹配。 在 10 位地址模式下, 当主设备已经将第一个字节发送出去时, 硬件将该位置'1' 软件读取 STS1 寄存器后, 对 CTRL1 寄存器的写操作将清除该位, 或当 PEN=0 时, 硬件清除该位 注: 收到一个 NACK 后, ADDR10F 位不被置位。
位 2	TDC	0x0	ro	数据传输完成标志 (Transmit data complete flag) 0: 未完成 (移位寄存器还有数据); 1: 已完成 (移位寄存器空闲) 读或写 DT 寄存器, 或者收到开始或结束信号自动清除。 当 STRETCH=0 接收时收到一个新字节(包括 ACK 脉冲)且数据寄存器还未被读取(RDBF=1) 发送时, 当一个新数据将被发送且数据寄存器还未被写入新的数据 (TDBE=1) 上述两种情况 TDC 位会置 1
位 1	ADDR7F	0x0	ro	0~7 位地址匹配标志 (0~7 bit address match flag) 0: 未产生; 1: 地址在主机模式下被发送或从机模式下接收到匹配地址。 在软件读取 STS1 寄存器后, 对 STS2 寄存器的读操作将清除该位 注: 在收到 NACK 后, ADDR7F 位不会被置位。
位 0	STARTF	0x0	ro	起始条件产生完成标志 (Start condition generation complete flag) 0: 未产生; 1: 已产生。 先读取 STS1 寄存器, 然后写 DT 寄存器清除标志。

### 11.5.7 状态寄存器2(I2C\_STS2)

域	简称	复位值	类型	功能
位 15: 8	PECVAL	0x00	ro	PEC 值 (PEC value) 当 PECEN 重置时清零。
位 7	ADDR2F	0x0	ro	接收到地址 2 标志 (Received address 2 flag) 0: 接收到的地址与 OADDR1 内的内容相匹配; 1: 接收到的地址与 OADDR2 内的内容相匹配。 当收到 STOP/START 条件自动清除, 或 I2CEN=0 时, 硬件将该位清除
位 6	HOSTADDRF	0x0	ro	SMBus 主机地址接收标志 (SMBus host address receiving flag) 0: 未接收; 1: 已接收。 当收到 STOP/START 条件自动清除, 或 I2CEN=0 时, 硬件将该位清除
位 5	DEVADDRF	0x0	ro	SMBus 设备地址接收标志 (SMBus device address receiving flag) 0: 未接收; 1: 已接收。 当收到 STOP/START 条件自动清除, 或 I2CEN=0 时, 硬件将该位清除
位 4	GCADDRF	0x0	ro	广播地址接收标志 (General call address reception flag) 0: 未接收; 1: 已接收。 当收到 STOP/START 条件自动清除, 或 I2CEN=0 时, 硬件将该位清除
位 3	保留	0x0	resd	保持默认值。
位 2	DIRF	0x0	ro	传输方向标志 (Transmission direction flag) 0: 接收数据; 1: 发送数据。 当收到 STOP 条件自动清除。
位 1	BUSYF	0x0	ro	总线忙标志 (Bus busy flag transmission mode) 0: 空闲; 1: 忙。 当检测到 SDA/SCL 变低时置起, 检测到停止条件清零。
位 0	TRMODE	0x0	ro	传输模式 (Transmission mode) 0: 从机; 1: 主机。 当设置了 GENSTART 并发出 START 后, 该位置起, 当检测到停止时, 该位清零。

### 11.5.8 时钟控制寄存器(I2C\_CLKCTRL)

域	简称	复位值	类型	功能
位 15	SPEEDMODE	0x0	rw	速度模式选择 (Speed mode selection) 0: 标准模式 (最快 100 kHz); 1: 快速模式 (最快 400 kHz)。 在快速模式下, 当 I <sup>2</sup> C 时钟为 10MHz 整数倍时, 可以产生准确的 400kHz 时钟
位 14	DUTYMODE	0x0	rw	快速模式占空比 (Fast mode duty cycle) 0: 高电平与低电平比值为 1: 2; 1: 低电平与高电平比值为 9: 16。
位 13: 12	保留	0x0	resd	保持默认值。
位 11: 0	SPEED	0x000	rw	I <sup>2</sup> C 总线速度配置 (I <sup>2</sup> C bus speed config) 在标准模式下: 高电平 = SPEED × T <sub>I2C_CLK</sub> ; 低电平 = SPEED × T <sub>I2C_CLK</sub> ; 在快速模式下: DUTYMODE = 0;

高电平= SPEED x T<sub>I2C\_CLK</sub> x 1;  
 低电平= SPEED x T<sub>I2C\_CLK</sub> x 2;  
 DUTYMODE = 1:  
 高电平= SPEED x T<sub>I2C\_CLK</sub> x 9;  
 低电平= SPEED x T<sub>I2C\_CLK</sub> x 16。  
 标准模式下最小值为 4，快速模式下最小值为 1。  
 只有在关闭 I2C 时(I2CEN=0)才能设置 CLKCTRL 寄存器；

注意：只有当 I<sup>2</sup>C 被关闭时(I2CEN=0)才能设置 CLKCTRL 寄存器。

### 11.5.9 时钟上升寄存器(I2C\_TMRISE)

域	简称	复位值	类型	功能
位 15 : 6 保留位		0x000	resd	硬件强制为 0
位 5 : 0 RISETIME		0x02	RW	<p>I2C 总线上升时间 ( I2C bus rise time )</p> <p>时间= RISETIME x T<sub>I2C_CLK</sub> ;</p> <p>标准模式下 I2C 协议标准为 1000ns . 计算公式为  <math>RISETIME = F_{I2C\_CLK} + 1</math> ;</p> <p>例如 I2C 时钟为 48MHz : RISETIME = 48+1 。</p> <p>快速模式下 I2C 协议标准为 300ns . 计算公式为 :  <math>RISETIME = F_{I2C\_CLK} \times 0.3 + 1</math> ;</p> <p>例如 I2C 时钟为 48MHz : RISETIME = 48x0.3+1</p> <p>注 : 只有当 I2C 被禁用(I2CEN=0)时 , 才能设置 RISETIME[5 : 0] 。</p>

# 12 通用同步异步收发器 (USART)

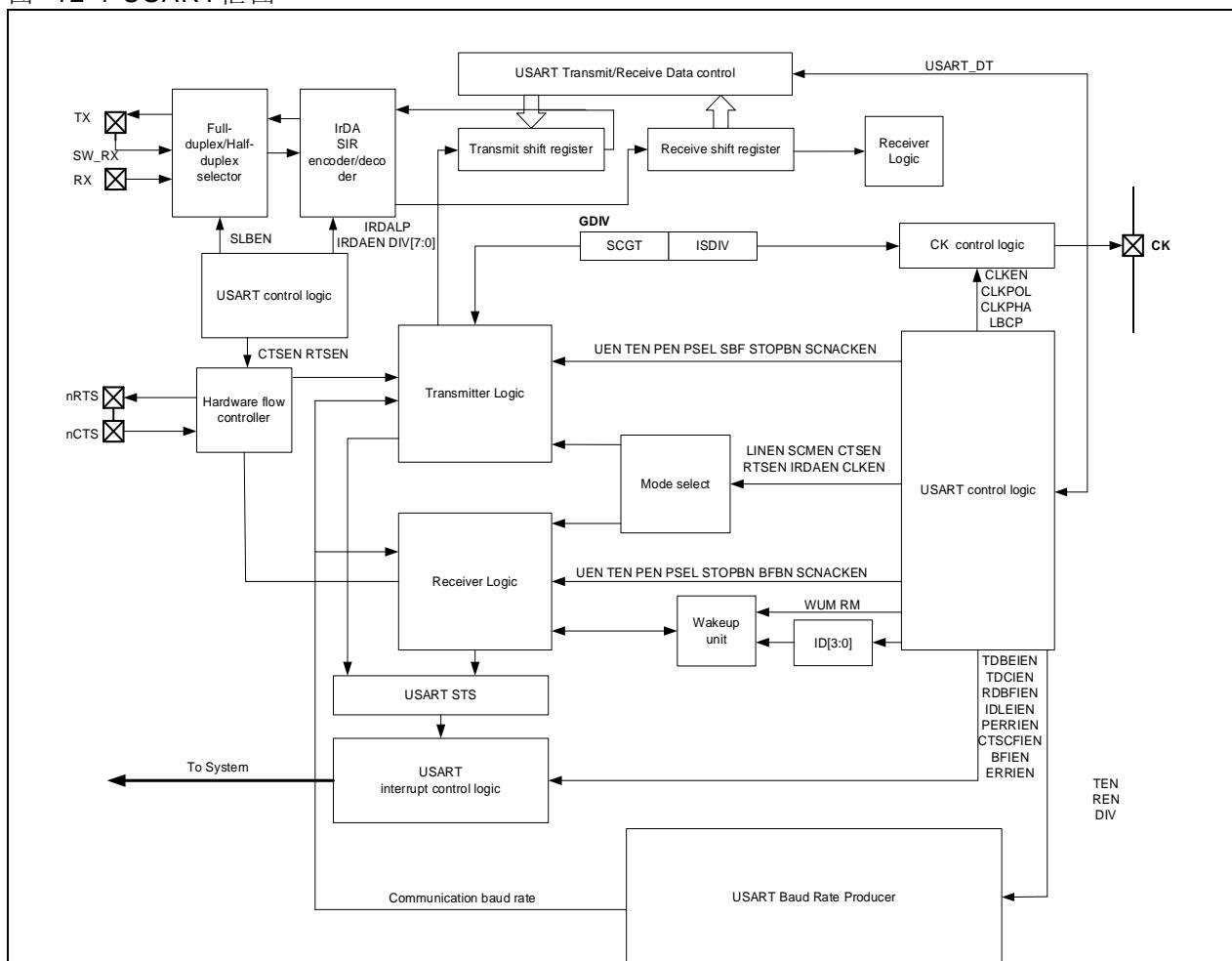
## 12.1 USART介绍

通用同步异步收发器 (USART) 是一个能通过多种不同的配置与使用不同的数据格式的外设进行通信的通用接口，同时支持异步全双工，异步半双工以及同步传输。USART 提供了可编程的波特率发生器，根据系统频率以及分频系数的不同，可产生高达 7.5MBits/s 的波特率，用户可以通过配置系统时钟以及分频系数以此产生所需要的特定通信频率。

USART 除了支持标准的 NRZ 异步以及同步收发通信协议外，还支持一些常用的其他类型的串行通信协议，如 LIN(局域互联网)，IrDA (红外数据组织) SIRENDEC 规范，ISO7816-3 标准的异步智能卡协议，以及 CTS/RTS (Clear To Send/Request To Send) 硬件流操作。

USART 还支持多处理器通信，以及可配置通过空闲帧或 ID 匹配唤醒的静默模式，以此搭建 USART 网络，并且同时支持使用 DMA 进行数据的收发，以此实现高速通信。

图 12-1 USART 框图



### USART 主要特性如下所列：

- 可编程配置的全双工或半双工通信
  - 全双工异步通信
  - 单线半双工通信
- 可编程配置的通信模式
  - NRZ标准格式 (Mark/Space)
  - LIN (局域互联网)
  - IrDA SIR (串行红外)
  - ISO7816-3标准里定义的异步智能卡协议： 智能卡模式支持0.5或1.5个停止位
  - RS-232 CTS/RTS (Clear To Send/Request To Send) 硬件流操作
  - 通过静默模式实现多处理器通信(具有ID匹配和总线空闲两种可编程配置的唤醒方式)

- 同步模式
- 可编程配置的波特率发生器
  - 发送和接收共用的可编程波特率，最高达 7.5MBits/s
- 可编程配置的帧格式
  - 可编程的数据位位数（8位或9位）
  - 可编程的停止位位数-支持1或2个停止位
  - 可编程的校验控制：发送方具备发送校验位的能力，接收方具备对接收到的数据进行校验的能力
- 可编程配置的 DMA 多缓冲器通信
- 可编程配置的独立的发送器和接收器使能位
- 可编程配置的输出 CLK 的相位和极性以及频率
- 检测标志
  - 接收缓冲器满
  - 发送缓冲器空
  - 传输完成标志
- 四个错误检测标志
  - 溢出错误
  - 噪声错误
  - 帧错误
  - 校验错误
- 可编程配置的 10 个带标志的中断源
  - CTSF 改变
  - LIN间隔帧检测
  - 发送数据寄存器空
  - 发送完成
  - 接收数据寄存器满
  - 检测到总线为空闲
  - 溢出错误
  - 帧错误
  - 噪声错误
  - 校验错误

## 12.2 全双工半双工选择器简述和配置流程

USART 全双工半双工选择器通过软件编程配置相应寄存器的方式，使得 USART 可以采用全双工或半双工的方式和外设进行数据交换。

USART 默认选择使用双线单向全双工时，TX 管脚用于数据输出，RX 管脚用于数据输入，USART 接收器和发送器相互独立，这使得 USART 可以同时进行数据发送和数据接收，以此实现全双工通信。

USART 在 HALFSEL 位置 1 时选择使用单线双向半双工的方式进行数据通信，在此条件下，LINEN 位，CLKEN 位，SCMEN 位以及 IRDAEN 位需置 0，此时在 USART 内部，RX 管脚无效，TX 管脚和 SW\_RX 管脚互连，对 USART 来说，TX 管脚用于数据输出，SW\_RX 用于数据输入，对外设来说，数据都从 TX 管脚映射的 IO 双向传输。

## 12.3 模式选择器简述和配置流程

### 12.3.1 模式选择器简述

USART 模式选择器通过软件编程配置相应寄存器的方式，使得 USART 可以根据软件的不同配置工作在不同的工作模式下，以此能与使用不同通信协议的外设之间实现数据交换。

USART 默认支持 NRZ 标准格式（Mark/Space），根据 USART 模式选择器配置的不同，USART 还可以支持 LIN（局域互联网），IrDA SIR（串行红外），ISO7816-3 标准里定义的异步智能卡协议，RS-232 CTS/RTS（Clear To Send/Request To Send）硬件流操作以及静默模式和同步模式。

### 12.3.2 模式选择器配置方法

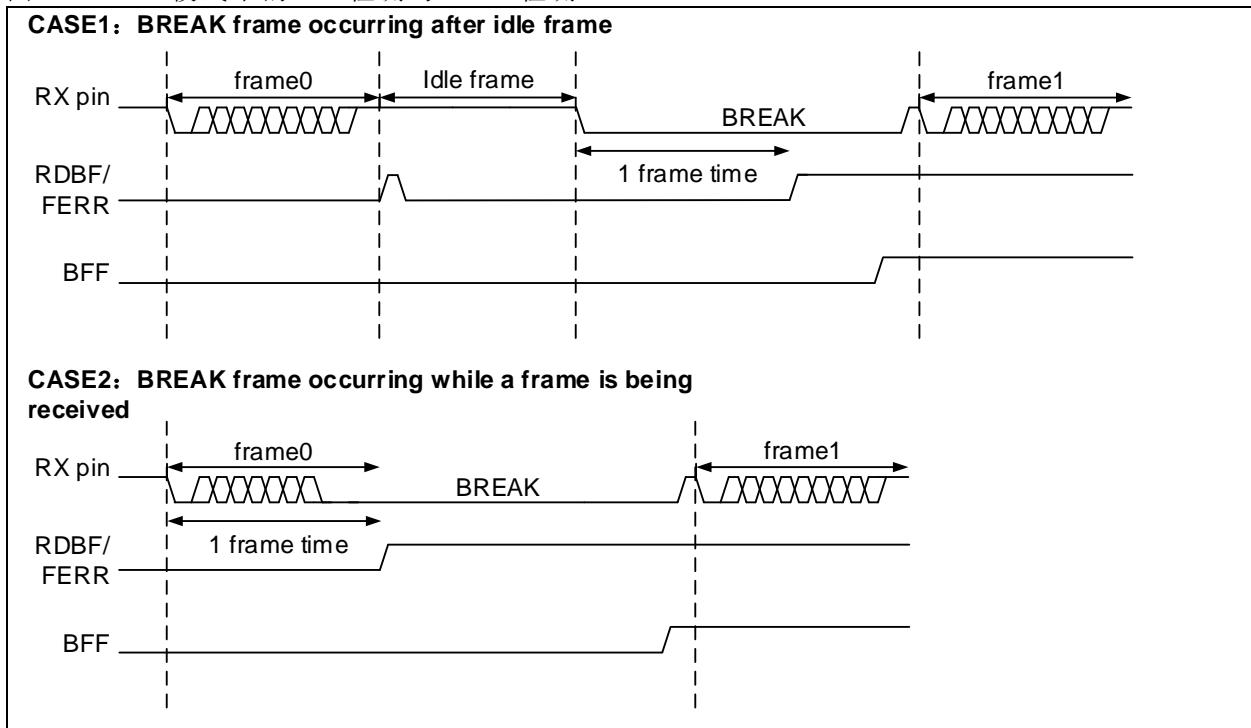
用户可以通过不同的配置以此选择不同的工作模式，配置方法分列如下所列，请将如下配置方法配合本章后述的接收器和发送器配置方法结合使用以完成 USART 初始化配置。

#### 1. LIN模式

基础设置：LINEN位置1，CLKEN位置0，STOPBN[1: 0]位置0，SCMEN位置0，SLHDEN位置0，IRDAEN位置0，DBN位置0。

LIN主机有发送间隔帧的能力，可以使用SBF位置1发送13位低电平的LIN同步间隔帧。同时LIN从机也有检测间隔帧的能力，可以选择BFBN位置1或0来选择是11位还是10位间隔帧检测。

图 12-2 LIN模式下的BFF检测与FERR检测



#### 2. 智能卡模式

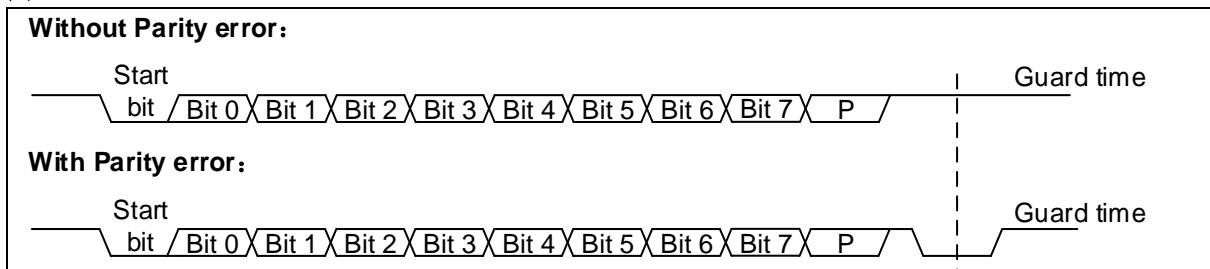
基础设置：SCMEN位置1，LINEN位置0，SLHDEN位置0，IRDAEN位置0，CLKEN位置1，DBN位置1，PEN位置1，STOPBN[1: 0]=11。

可以选择配置CLKPOL位和CLKPH位以及LBCP位以满足不同的时钟极性以及时钟相位和时钟脉冲个数，具体可见同步模式部分。

通过配置SCGT[7: 0]位选择保护时间，使TDC标志的置起可以得到延时，直到保护时间计数器向上计数到SCGT[7: 0]的值，TDC才得以置起。

而智能卡属于单线双向半双工通信，可以通过配置SCNACKEN位选择是否在校验出错时发送NACK，以告知数据没有被正确接收。

图 12-3 Smartcard frame format

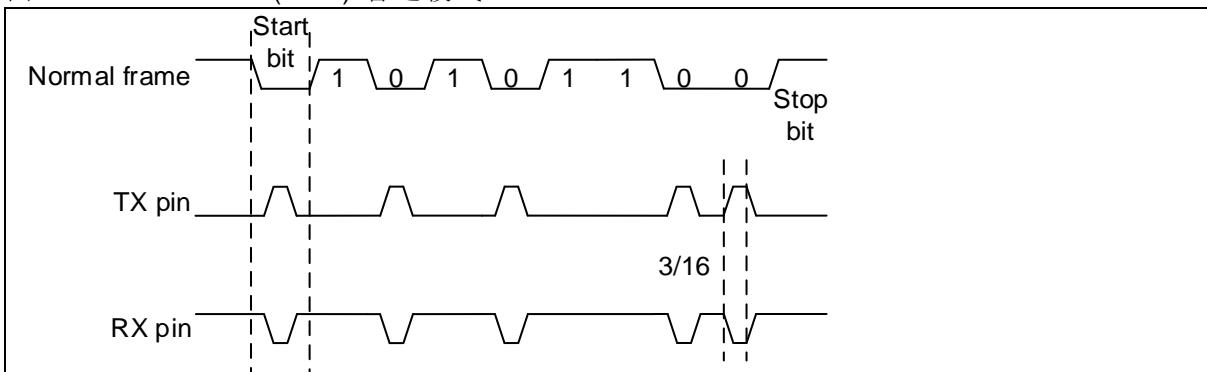


#### 3. 红外模式

基础设置：IRDAEN位置1，CLKEN位置0，STOPBN[1: 0]位置0，SCMEN位置0，SLHDEN位置0。

可以选择IRDALP位置1以开启红外低功耗模式，在普通模式下持续时间为3/16位，在红外低功耗模式下位持续时间可调，并配合ISDIV[7: 0]配置想要产生的低功耗频率。

图 12-4 IrDA DATA(3/16)-普通模式



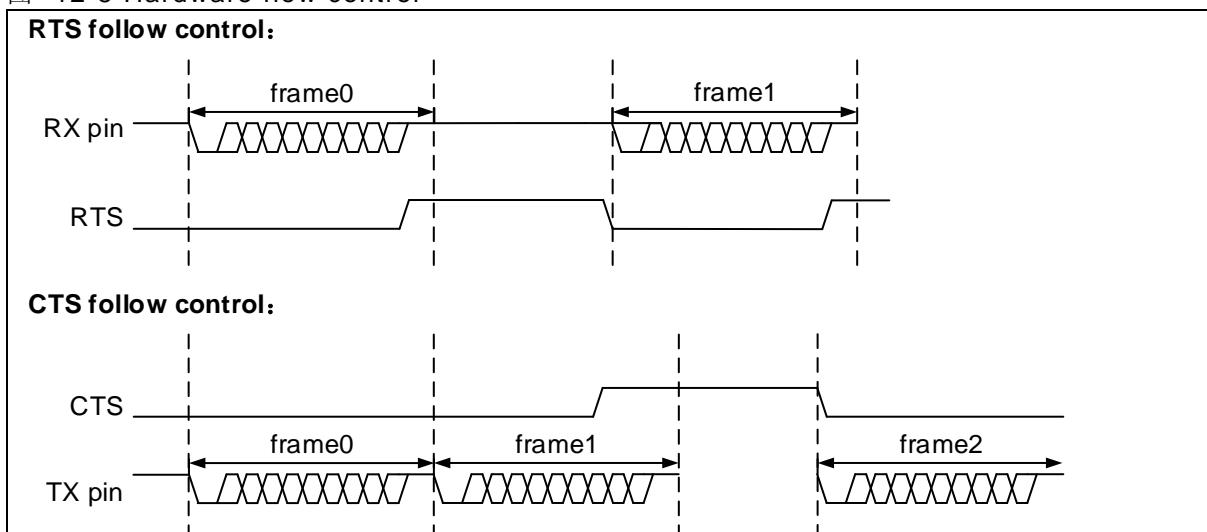
#### 4. 硬件流控制模式

通过RTSEN位置1和CTSEN位置1可以分别开启RTS和CTS流控制，控制两设备之间的串型数据流。

**RTS流控制：** USART接收器准备好接收新的数据，RTS就变成有效（下拉为低电平）。当接收寄存器内有数据到达时（在每个stop位开始时），RTS被置位，由此表明希望在当前帧结束时停止数据传输。

**CTS流控制：** USART发送器在发送下一帧前检查CTS输入。如果CTS有效（也即CTS为低电平），则下一个数据被发送；若CTS在传输期间被变成无效（也即CTS为高电平），当前的传输完成后停止发送。

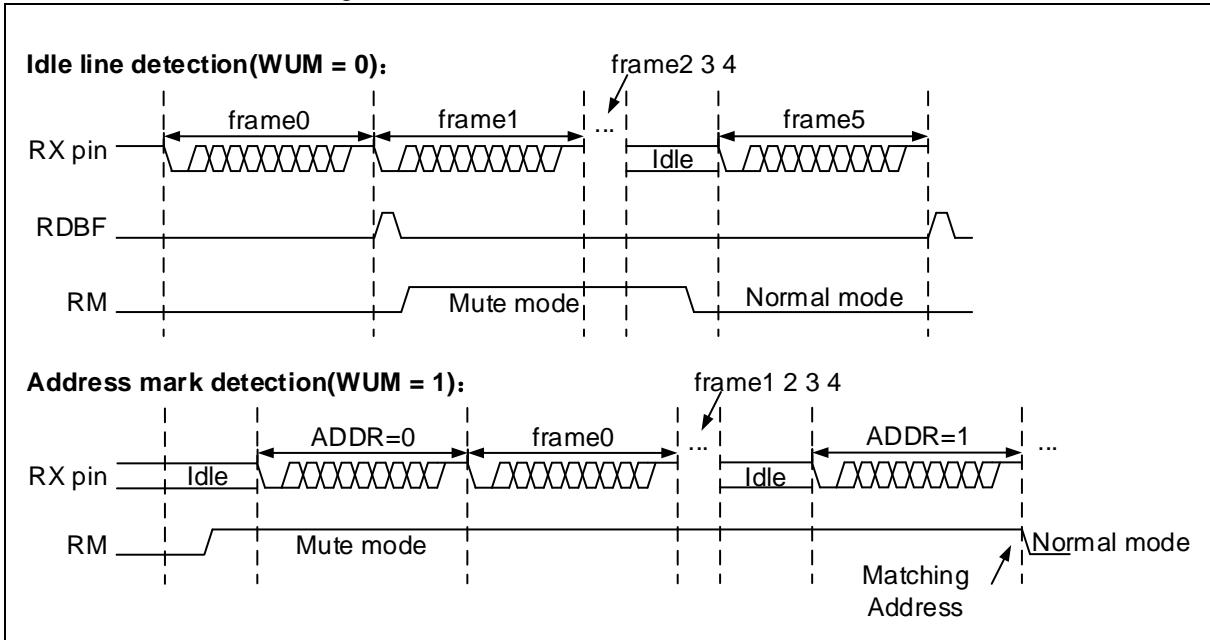
图 12-5 Hardware flow control



#### 5. 静默模式

RM位置1进入静默模式，根据WUM位置1和置0，可以分别通过ID匹配和空闲总线从静默模式中唤醒，其中ID号ID[3: 0]可编程配置，当选择ID匹配时，数据位的MSB为1表示当前数据是ID，4个LSB表示ID值。

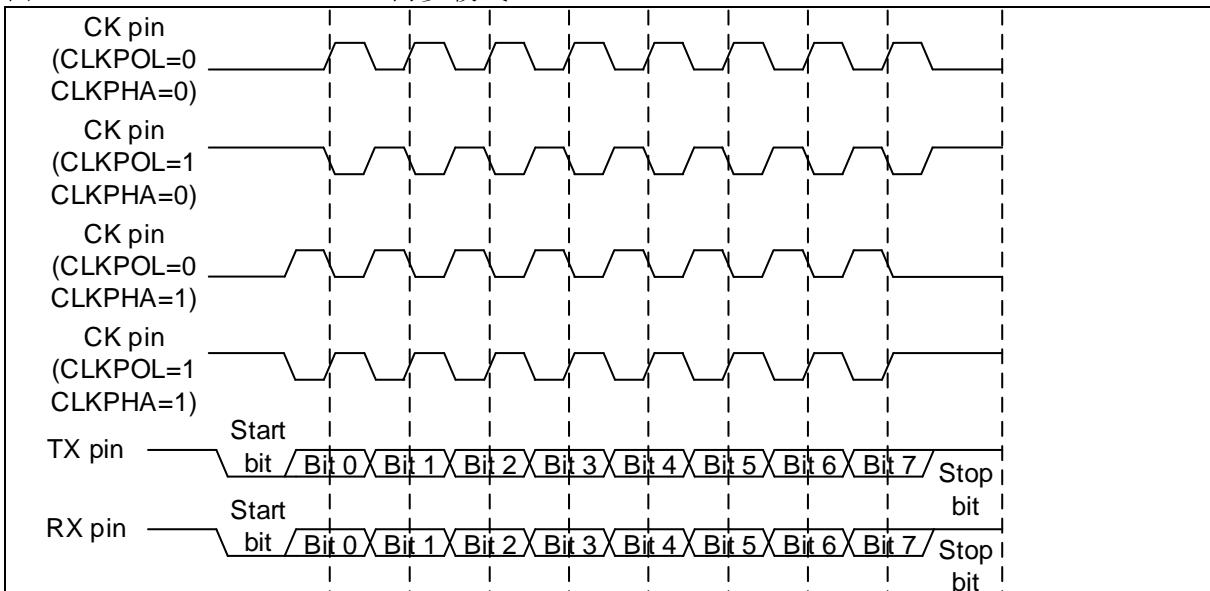
图 12-6 Mute mode using Idle line or Address mark detection



## 6. 同步模式

CLKEN位置1开启同步模式并使能时钟管脚输出，通过配置CLKPOL位置1或0可以选择空闲状态下CK管脚上的电平为高或低，通过配置CLKPHA位置1或0可以选择在时钟的第二个或第一个边沿开始采样数据，通过配置LBCP位置1或0可以选择最后一位数据是否输出时钟，通过配置ISDIV[4: 0]可以选择想要输出的时钟频率。

图 12-7 8-bit format USART 同步模式



## 12.4 USART帧格式简述和配置流程

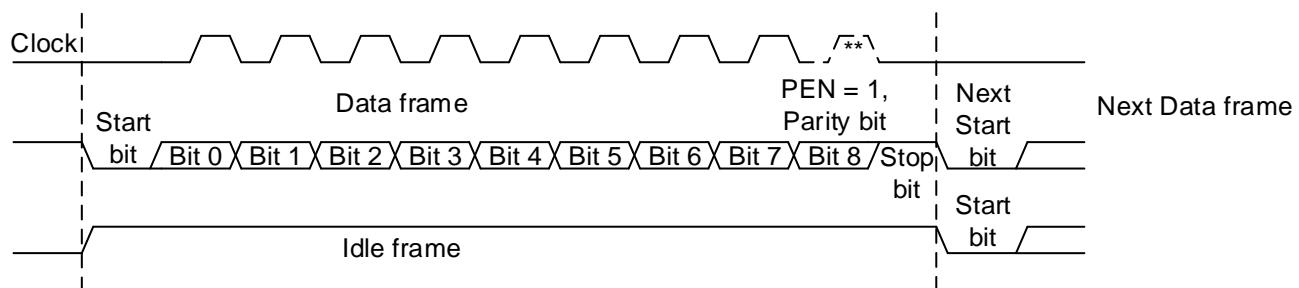
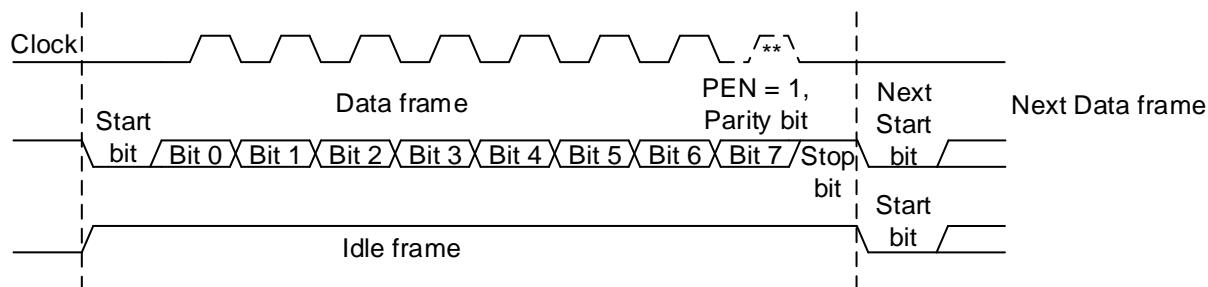
USART一笔数据帧由起始位，数据位，停止位依次组成，最后一位数据位可以作为校验位。

USART一笔空闲帧的长度等于当前配置下数据帧的长度，但所有位都为1。

USART一笔间隔帧的长度等于当前配置下数据帧的长度加上停止位，停止位之前的所有位都等于0。

通过DBN位配置8位(DBN=0)或9位(DBN=1)数据位。

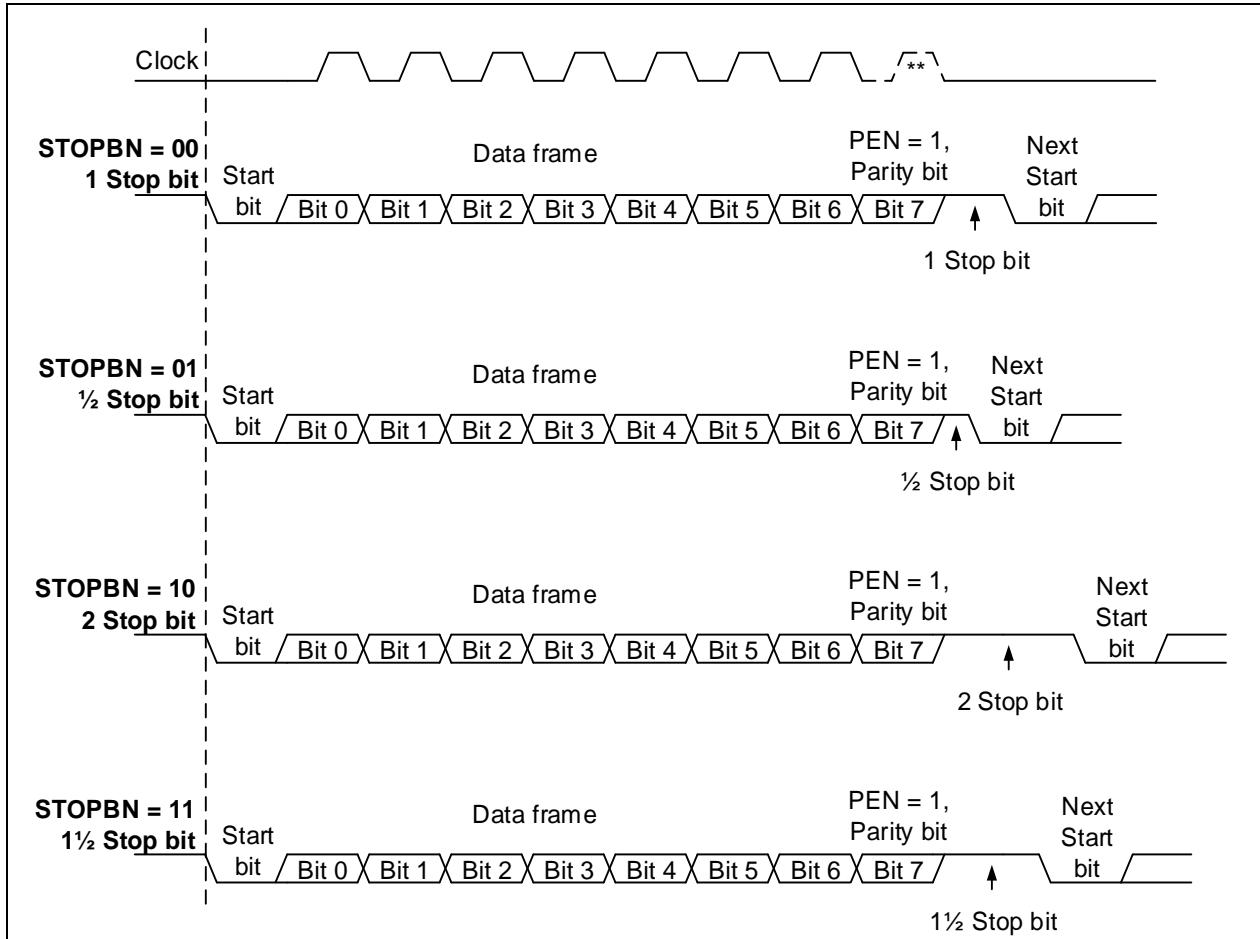
图 12-8 字长设置

**9-bit word length (DBN = 1):****8-bit word length (DBN = 0):**

通过 STOPBN 位配置 1 位 (STOPBN=00), 0.5 位 (STOPBN=01), 2 位 (STOPBN=10), 1.5 位 (STOPBN=11) 停止位。

通过 PEN 位置“1”配置校验控制使能，通过 PSEL 位配置奇校验 (PSEL=1) 或偶检验 (PSEL=0)，校验控制使能后数据位的 MSB 将由奇偶校验位替代，即有效数据位减少一位。

图 12-9 配置停止位



## 12.5 DMA传输简述和配置流程

USART 可以使用 DMA 操作发送数据缓冲器和接收数据缓冲期以实现高速连续传输，USART 的 DMA 传输需要配合 DMA 使用，下方会简述配置流程，但具体和 DMA 配置相关部分请参见 DMA 章节的描述。

### 12.5.1 DMA发送配置流程

1. 选择DMA传输通道：在DMA章节DMA通道映射表中选择用于当前所用USART的DMA通道。
2. 配置DMA传输目标地址：在DMA控制寄存器中DMA传输目的地址位写入当前所使用的USART的数据寄存器（USART\_DT）地址，DMA将会在接收到发送请求后将代发送的数据写入该地址。
3. 配置DMA传输源地址：在DMA控制寄存器中DMA传输源地址位写入代发送数据存放的地址，DMA将会在接收到发送请求后将该地址内的数据写入到目标地址中，即写入到当前所使用的USART的数据寄存器（USART\_DT）中。
4. 配置DMA传输字节个数：在DMA控制寄存器相关位置配置期望传输的字节个数
5. 配置DMA传输通道优先级：在DMA控制寄存器相关位置配置当前所使用通道的USART的DMA传输通道优先级。
6. 配置DMA中断产生时机：在DMA控制寄存器相关位置配置是在传输完成或传输完成一半时产生DMA中断。
7. 使能DMA传输通道：在DMA控制寄存器相关位置使能当前所选用的DMA通道

### 12.5.2 DMA接收配置流程

1. 选择DMA传输通道：在DMA章节DMA通道映射表中选择用于当前所用USART的DMA通道。
2. 配置DMA传输目标地址：在DMA控制寄存器中DMA传输目的地址位写入期望存放接收数据的地址，DMA将会在接收到接收请求后，将当前所使用的USART的数据寄存器（USART\_DT）中的数据存放在目的地址中。
3. 配置DMA传输源地址：在DMA控制寄存器中DMA传输源地址位写入当前所使用的USART

的数据寄存器（**USART\_DT**）的地址，DMA将会在接收到接收请求后将该地址内的数据写入到目标地址中，即写入到期望存放接收数据的地址。

4. 配置 DMA 传输字节个数：在 DMA 控制寄存器相关位置配置期望传输的字节个数
5. 配置 DMA 传输通道优先级：在 DMA 控制寄存器相关位置配置当前所使用通道的 **USART** 的 DMA 传输通道优先级。
6. 配置 DMA 中断产生时机：在 DMA 控制寄存器相关位置配置是在传输完成或传输完成一半时产生 DMA 中断。
7. 使能 DMA 传输通道：在 DMA 控制寄存器相关位置使能当前所选用的 DMA 通道

## 12.6 波特率发生器简述及配置流程

### 12.6.1 波特率发生器简述

**USART** 波特率发生器通过使用内部计数器，以 PCLK 为基准，DIV（波特比率寄存器（**USART\_BAUDR**）[15: 0]）即为该计数器的溢出值，该计数器计满一次代表一位数据，所以每位数据位宽为 DIV 个 PCLK 周期。

由于 **USART** 的接收器和发送器共用同一个波特率发生器，并且接收器将每位数据拆分为 16 份等长的部分以此来实现过采样，所以数据位宽不得小于 16 个 PCLK 周期，即 DIV 中的值必须大于或等于 16。

### 12.6.2 波特率发生器配置方法

用户可通过配置不同的系统时钟以及在波特比率寄存器（**USART\_BAUDR**）中写入不同的值以此产生特定的波特率，具体的运算关系见如下公式

$$\text{TX/RX 波特率} = \frac{f_{CK}}{\text{DIV}}$$

这里的  $f_{CK}$  是指 **USART** 的系统时钟（即对应的 PCLK1/PCLK2）

注：1. 波特比率寄存器（**USART\_BAUDR**）中的值需要在 UEN 之前写入，且 UEN=1 时，不可更改这些位。

2. 关闭 **USART** 接收器或发送器会使内部计数器复位，波特率发生中断。

表 12-1 设置波特率时的误差计算

波特率			<b>fPCLK=36MHz</b>		<b>fPCLK=72MHz</b>		
序号	Kbps	实际	置于波特率寄存器中的值	误差%	实际	置于波特率寄存器中的值	误差%
1	2.4	2.4	15000	0%	2.4	30000	0%
2	9.6	9.6	3750	0%	9.6	7500	0%
3	19.2	19.2	1875	0%	19.2	3750	0%
4	57.6	57.6	625	0%	57.6	1250	0%
5	115.2	115.384	312	0.15%	115.2	625	0%
6	230.4	230.769	156	0.16%	230.769	312	0.16%
7	460.8	461.538	78	0.16%	461.538	156	0.16%
8	921.6	923.076	39	0.16%	923.076	78	0.16%
9	2250	2250	16	0%	2250	32	0%
10	4500	不可能	不可能	不可能	4500	16	0%

以波特率 115.2Kbps 为例，假设 fPCLK 为 36MHz，此时波特率寄存器应设置为 312(0x138)，经由公式计算： $36000000 / 312 = 115384 = 115.384\text{Kbps}$

而它们的误差计算为(实际值 - 理论值) / 理论值 \* 100%： $(115.384 - 115.2) / 115.2 * 100\% = 0.15\%$

## 12.7 发送器简述和配置流程

### 12.7.1 发送器简述

**USART** 发送器具有独立的使能位 TEN，发送器与接收器共用同一个波特率且该波特率可编程配置，**USART** 具有一个发送数据缓冲器（TDR）和一个发送移位寄存器，当发送数据缓冲器（TDR）为空时，TDBE 置起，如果设置了 TDBEIE 会产生中断。

软件写入的值会先存储在发送数据缓冲器（TDR）中，当发送移位寄存器为空时，**USART** 会将发送数据缓冲器中的值移入到发送移位寄存器，**USART** 发送器将以 LSB 的方式将发送移位寄存器中的数据从 TX 脚输出，具体的输出格式取决于软件配置的帧格式。

如若选择了同步传输或者配置了时钟输出，**USART** 发送器将时钟脉冲从 CK 脚输出，如若选择了硬件流

控制, USART 发送器将控制信号将从 CTS 管脚输入。

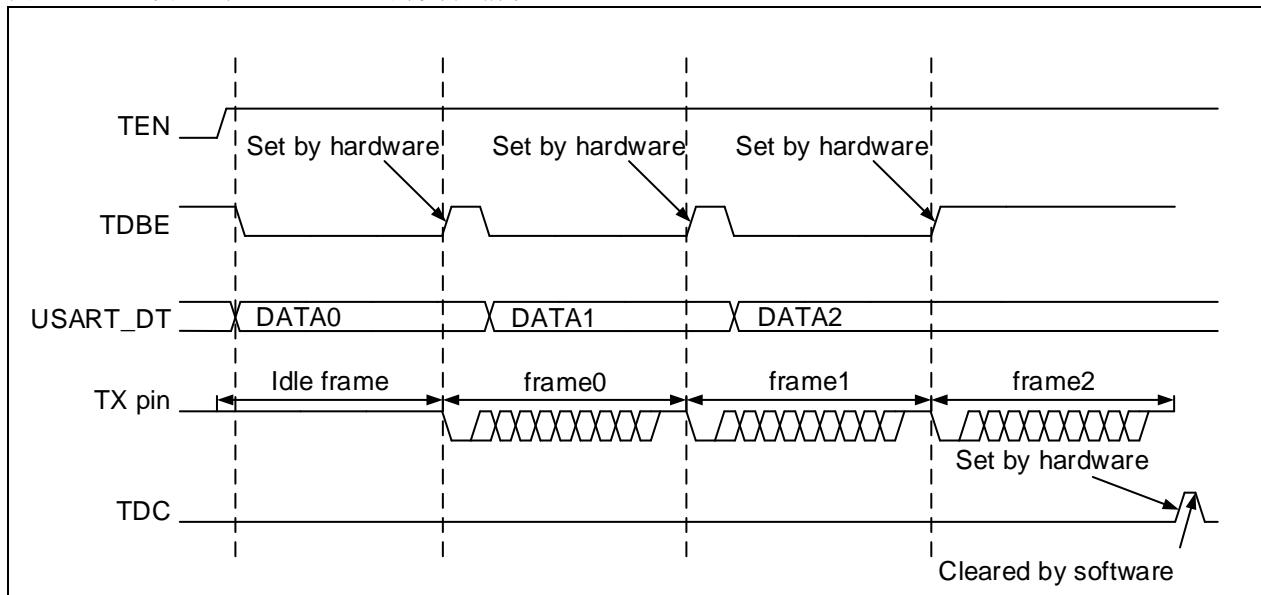
注意: 1. 在数据传输期间不能复位 TEN 位, 否则将破坏 TX 脚上的数据。

2. TEN 位被激活后, USART 将自动发送一个空闲帧。

## 12.7.2 发送器配置流程

1. USART使能: UEN位置1。
2. 全双工半双工配置: 具体参见全双工半双工选择器配置部分 (12.2错误!未找到引用源。)。
3. 模式配置: 具体参见模式选择器配置部分 (12.3错误!未找到引用源。)。
4. 帧格式配置: 具体参见帧格式配置部分 (12.4错误!未找到引用源。)。
5. 中断配置: 具体参见中断发生器配置部分 (12.10错误!未找到引用源。)。
6. DMA发送配置: 如果选择使用DMA发送, DMATEN位(控制寄存器3 (USART\_CTRL3) [7])置1, 并按照DMA传输中的描述配置DMA寄存器。
7. 波特率配置: 具体参见波特率发生器配置部分 (12.6错误!未找到引用源。)。
8. 发送器使能: TEN位置1, 置1后USART发送器会自动发送一个空闲帧。
9. 数据写入: 等待TDBE位置起后, 将要发送的数据写入数据寄存器 (USART\_DT) (此操作会清除 TDBE位), 在非DMA模式下, 重复此操作。
10. 在写入最后一个期望传输的数据后, 等待TDC位置起, 这表示最后一个数据帧的传输结束, 在该标志置起前, 禁止关闭USART, 否则传输可能出错。
11. 在TDC=1后, 可以采用先读一次状态寄存器 (USART\_STS), 再写一次数据寄存器 (USART\_DT) 的方式来清除TDC; 也可以采用软件对它写 '0' 来清除, 但此方法只推荐在DMA模式下使用。

图 12-10 发送时 TDC/TDBE 的变化情况



## 12.8 接收器简述和配置流程

### 12.8.1 接收器简述

USART 接收器具有独立的接收器使能位 REN(控制寄存器 1 (USART\_CTRL1) [2]), 接收器和发送器共用同一个波特率且该波特率可编程配置, USART 具有一个接收数据缓冲器 (RDR) 和一个接收移位寄存器。

数据从 USART 的 RX 脚输入, 当接收器判断到一个有效的起始位后, 接收器会以 LSB 的方式将接收到的数据依次移入接收移位寄存器, 并根据软件配置的帧格式, 在接收到一个完整的数据帧后将接收移位寄存器中的值移入接收数据缓冲器并置起 RDBF, 如果设置了 RDBFIEN 将会产生中断。

如若选择了硬件流控制, USART 接收器将控制信号将从 RTS 管脚输出。

在数据接收过程中, USART 接收器会根据软件的配置检测帧错误, 溢出错误, 奇偶校验错误以及噪声错误, 并根据相应的中断使能位是否置位来判断是否产生相应的中断。

## 12.8.2 接收器配置流程

配置步骤：

1. USART使能： UEN位置1。
2. 全双工半双工配置： 具体参见全双工半双工选择器配置部分（12.2错误!未找到引用源。）。
3. 模式配置： 具体参见模式选择器配置部分（12.3错误!未找到引用源。）。
4. 帧格式配置： 具体参见帧格式配置部分（12.4错误!未找到引用源。）。
5. 中断配置： 具体参见中断发生器配置部分（12.10错误!未找到引用源。）。
6. DMA接收配置： 如果选择使用DMA接收， DMAREN位置1，并按照DMA传输中的描述配置DMA寄存器。
7. 波特率配置： 具体参见波特率发生器配置部分（12.6错误!未找到引用源。）。
8. 接收器使能： REN位置1。

当一字符被接收到时：

- RDBF 位被置位。它表明移位寄存器的内容被转移到 RDR (Receiver Data Register)。换句话说，数据已经被接收并且可以被读出（包括与之有关的错误标志）。
- 如果 RDBFIEN 位被设置，则产生中断。
- 在接收期间如果检测到帧错误，噪声或溢出错误，错误标志将被置起。
- 在 DMA 传输时，RDBF 在每个字节接收后被置起，并由 DMA 对数据寄存器的读操作而清零。
- 在非 DMA 传输时，由软件读数据寄存器 (USART\_DT) 完成对 RDBF 位清除。RDBF 标志也可以通过对它写 0 来清除。RDBF 位必须在下一帧数据接收结束前被清零，以避免溢出错误。

当一个间隔帧被接收到时：

- 非 LIN 模式： USART 接收器按照帧错误处理，并置起 FERR 位，若相应中断使能，中断产生，具体可见下方错误帧的描述。
- LIN 模式： USART 接收器按间隔帧处理，并置起 BFF 位，若 BFIEN 置位，则中断产生。

当一个空闲帧被接收到时：

- USART 接收器按数据帧处理，并置起 IDLEF 位，若 IDLEIEN 置位，则中断产生。

当一个帧错误产生时：

- FERR 位置位。
- USART 接收器将错误的数据从接收移位寄存器转移到接收数据缓冲器。
- 在非 DMA 传输时，这个位和 RDBF 位同时置起，后者将产生中断。在 DMA 传输时，如果 ERRIEN 置位的话，将产生中断。

当一个溢出错误产生时：

- ROERR 位被置位。
- 接收数据缓冲器中的数据不会被覆盖，读数据寄存器 (USART\_DT) 仍能得到先前的数据。
- 接收移位寄存器中的内容会被覆盖，随后接收到的数据都将丢失。
- 如果 RDBFIEN 位置位或 ERRIEN 和 DMAREN 位都被置位，中断产生。
- 先读状态寄存器 (USART\_STS)，再读数据寄存器 (USART\_DT)，可清除 ROERR。

注意： 当 ROERR 置位时，表明至少有 1 个数据已经丢失。有两种可能性：

如果 RDBF=1，上一个有效数据还存储在接收数据缓冲器中，可以被读出。如果 RDBF=0，这意味着上一个有效数据已经从接收数据缓冲器中读走。

注意： 在接收数据时，REN 位不应该被复位。如果 REN 位在接收时被清零，当前字节的接收被丢失。

## 12.8.3 起始侦测和噪声检测

USART 接收器在 REN 位置位后便开始侦测起始位，USART 接收器通过过采样技术，在第 3、5、7、8、9、10 位共 6 个点进行数据采样，以此侦测有效起始位以及识别噪声，具体的噪声和有效起始位的判别方式可以参见下方检测起始位和噪声的数据采样表。

表 12-2 检测起始位和噪声的数据采样

采样值 (3 • 5 • 7)	采样值 (8 • 9 • 10)	NERR 位	起始位有效性
000	000	0	有效
001/010/100	001/010/100	1	有效
001/010/100	000	1	有效
000	001/010/100	1	有效
111/110/101/011	任意值	0	无效
任意值	111/110/101/011	0	无效

注意：如果在第 3、5、7、8、9、10 位的采样值满足不了上表任意一种组合，则 USART 接收器认为没有接受到正确的起始位，将退出起始位侦测并回到空闲状态等待下降沿。USART 接收器具备噪声检测功能，在非同步模式时，使用过采样技术，在第 7、8、9 采样点，根据不同的采样值，区别有效输入数据和噪声，并恢复数据和置起噪声错误标志位 NERR。具体的采样方法以及噪声和有效数据的判别方式可以参见下方检测有效数据和噪声的数据采样表。

表 12-3 检测有效数据和噪声的数据采样

采样值	NERR 位	接收的位	数据有效性
000	0	0	有效
001	1	0	无效
010	1	0	无效
011	1	1	无效
100	1	0	无效
101	1	1	无效
110	1	1	无效
111	0	1	有效

USART 接收器在最大允许偏差下，皆可以正常接收数据，其值取决于 USART\_CTRL1 的 DBN 以及 USART\_BAUDR 的 DIV[3: 0]。

注意：以下表格的最大允许偏差是以波特率 115.2Kbps 为基准进行计算，实际接收器最大允许偏差会随着波特率设定大小有所改变，波特率越大时其最大允许偏差会越小，反过来波特率越小其最大允许偏差会越大。

表 12-4 最大允许偏差

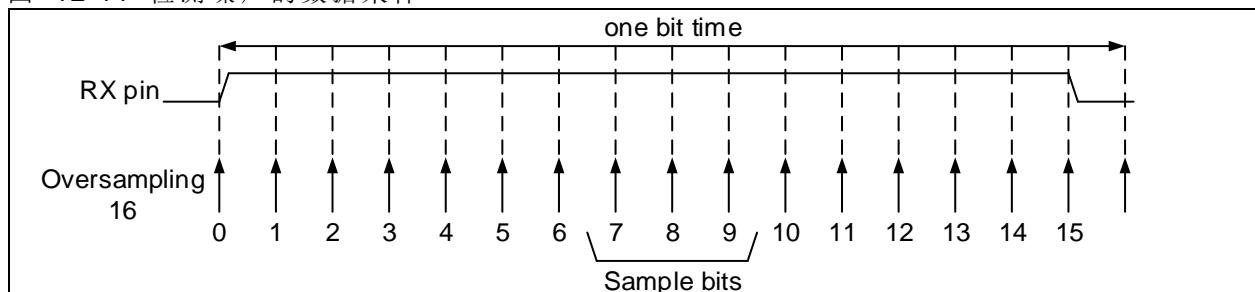
DBN	DIV[3:0] = 0	DIV[3:0] != 0
0	3.75%	3.33%
1	3.41%	3.03%

当 USART 接收器在数据帧中检测到噪声时：

- 在 RDBF 位置起的同时置起 NERR 位。
- USART 接收器将错误数据从接收移位寄存器转移到接收数据缓冲器。
- 在非 DMA 传输时，没有噪声中断产生。然而，因为 NERR 位和 RDBF 位是同时置位，RDBF 将产生中断。在 DMA 传输时，如果 ERRIEN 位置位，中断产生。

先读状态寄存器 (USART\_STS)，再读数据寄存器 (USART\_DT)，将清除 NERR 位。

图 12-11 检测噪声的数据采样



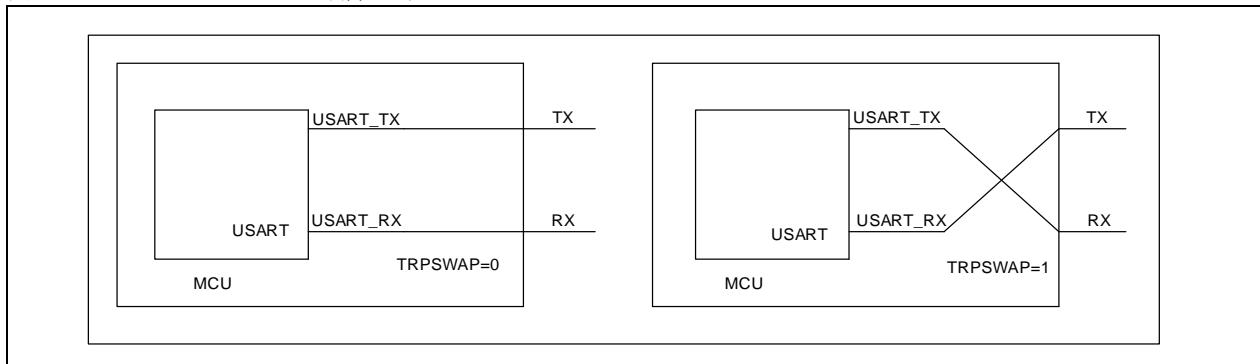
## 12.9 Tx/Rx可配置引脚互换

如果 TRPSWAP (USART\_CTRL2[15]) 位被使能，MCU的Tx/Rx引脚顺序将被交换。以下举例两种常见应用场景：

若用户在外接 RS-232 芯片时不慎将 Tx/Rx 接反，可通过修改 TRPSWAP 位更换引脚顺序，无需修改硬件连接。

若用户在全双工模式下只将主机的 Tx 和从机的 Rx 连接，在主机和从机互换后，也可通过 TRPSWAP 位更换引脚顺序，无需修改硬件连接。

图 12-12 Tx/Rx 可配置引脚互换



注意：SWAP（USART\_CTRL2[15]）位必须在 USART 未被使能（UEN=0）时才能被改写

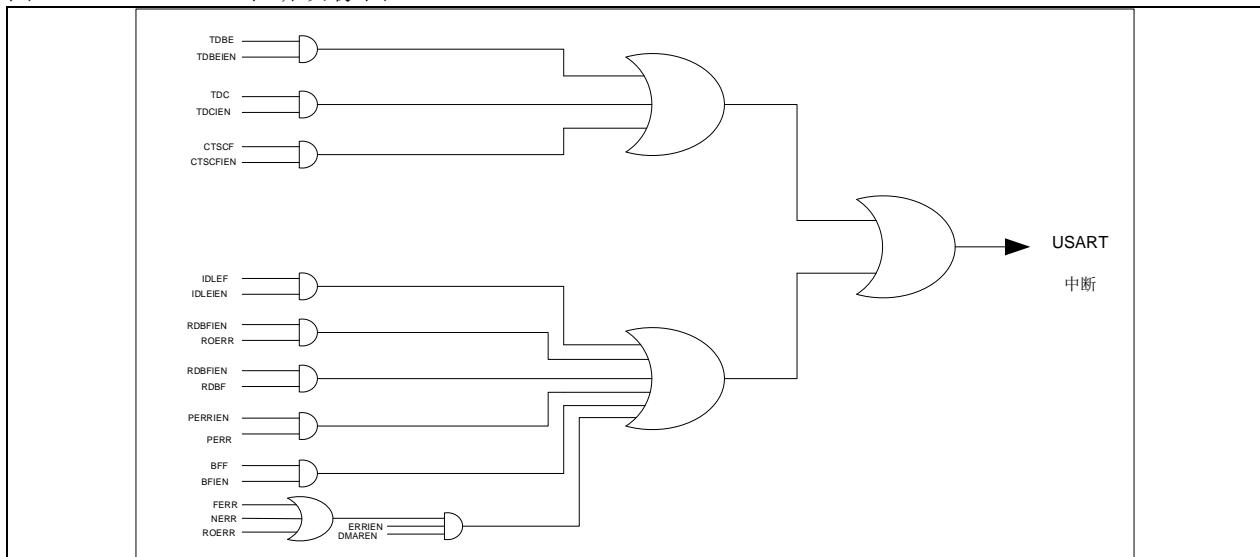
## 12.10 中断

USART 中断发生器是 USART 中断的控制中枢，USART 中断产生器会实时监测 USART 内部的中断源，并根据软件配置的相应中断源的中断使能位，以此决定是否产生中断，下表所示为 USART 的中断源以及相应的中断使能位，对相应的中断使能位置 1 时，即可在相应事件出现后产生中断。

表 12-5 USART 中断请求

中断事件	事件标志	使能位
发送数据寄存器空	TDBE	TDBEIEN
CTS 标志	CTSCF	CTSCFIEN
发送完成	TDC	TDCIEN
接收数据就绪可读	RDBF	RDBFIEN
检测到数据溢出	ROERR	
检测到空闲线路	IDLEF	IDLEIEN
奇偶检验错	PERR	PERRIEN
断开标志	BFF	BFIEN
噪声标志，多缓冲通信中的溢出错误和帧错误	NERR 或 ROERR 或 FERR	ERRIEN (1)

图 12-13 USART 中断映像图



## 12.11 I/O 管脚控制

USART 通过五个接口外部设备进行通信，管脚定义如下：

**RX:** 串行数据输入端。

**TX:** 串行数据输出端。在单线半双工模式和智能卡模式里，TX 脚作为 I/O 使用，即用于发送数据也用于接收数据。

CK: 发送器时钟输出。输出的 CLK 相位和极性以及频率均可编程配置。  
 CTS: 发送器输入端, 硬件流控制模式发送使能信号。

RTS: 接收器输出端, 硬件流控制模式发送请求信号。

## 12.12 USART寄存器描述

必须以字(32位)的方式操作这些外设寄存器。

表 12-6 USART寄存器映像和复位值

寄存器简称	基址偏移量	复位值
USART_STS	0x00	0x0000 00C0
USART_DT	0x04	0x0000 0000
USART_BAUDR	0x08	0x0000 0000
USART_CTRL1	0x0C	0x0000 0000
USART_CTRL2	0x10	0x0000 0000
USART_CTRL3	0x14	0x0000 0000
USART_GDIV	0x18	0x0000 0000

### 12.12.1 状态寄存器 (USART\_STS)

域	简称	复位值	类型	功能
位 31: 10	保留	0x0000000	resd	硬件强制为 0。
位 9	CTSCF	0x0	rw0c	CTS 变化标志 (CTS change flag) 当 CTS 线发送变化时, 该位被硬件置起, 由软件将其清零。 0: 无; 1: 有。
位 8	BFF	0x0	rw0c	间隔帧标志 (break frame flag) 当检测到间隔帧时, 该位被硬件置起, 由软件将其清零。 0: 无; 1: 有。
位 7	TDBE	0x1	ro	发送缓冲器空 (Transmit data buffer empty) 当发送缓冲器为空, 可以再次写入数据时, 该位被硬件置起。对数据寄存器 (USART_DT) 的写操作, 将清零该位。 0: 非空; 1: 空。
位 6	TDC	0x1	rw0c	发送数据完成 (Transmit data complete) 当发送数据完成, 该位被硬件置起, 由软件将其清零 (方式 1: 先读状态寄存器 (USART_STS), 再写数据寄存器 (USART_DT); 方式 2: 操作该位写'0')。 0: 未完成; 1: 完成。
位 5	RDBF	0x0	rw0c	接收数据缓冲器满 (Receive data buffer full) 当接收到数据时, 该位被硬件置起, 由软件将其清零 (方式 1: 读数据寄存器 (USART_DT); 方式 2: 操作该位写'0')。 0: 未收到; 1: 收到。
位 4	IDLEF	0x0	ro	总线空闲 (Idle flag) 当检测到总线空闲时, 该位被硬件置起, 由软件将其清零 (先读状态寄存器 (USART_STS), 再读数据寄存器 (USART_DT))。 0: 无; 1: 有。
位 3	ROERR	0x0	ro	接收器溢出错误 (Receiver overflow error)

				当 RDBF 仍然置起没有清除的时候，如果此时又收到数据，该位被硬件置起，由软件将其清零（先读状态寄存器（USART_STS），再读数据寄存器（USART_DT））。 0: 无； 1: 有。 注意：该位被置位时，DT 寄存器中的数据不会丢失，但是后续的数据会被覆盖。
位 2	NERR	0x0	ro	噪声错误（Noise error） 接收到的数据有杂讯时，该位被硬件置起，由软件将其清零（先读状态寄存器（USART_STS），再读数据寄存器（USART_DT））。 0: 无； 1: 有。
位 1	FERR	0x0	ro	帧错误（Framing error） 当检测到停止位异常（检测到低电平）、过多的杂讯噪声或者检测到间隔帧，该位被硬件置起，由软件将其清零（先读状态寄存器（USART_STS），再读数据寄存器（USART_DT））。 0: 无； 1: 有。
位 0	PERR	0x0	ro	校验错误（Parity error） 接收如果出现奇偶校验错误，该位被硬件置起，由软件将其清零（先读 USART_STS，再读数据寄存器（USART_DT））。 0: 无； 1: 有。

## 12.12.2 数据寄存器（USART\_DT）

域	简称	复位值	类型	功能
位 31: 9	保留位	0x000000	resd	硬件强制为 0。
位 8: 0	DT	0x00	rw	数据值（Data value） 该寄存器包含读和写的功能。当奇偶校验位使能，发送操作时，写到 MSB 的值会被校验位取代。接收操作时，读到的 MSB 位是接收到的校验位。

## 12.12.3 波特比率寄存器（USART\_BAUDR）

注意：如果 TEN 或 REN 均被禁止，波特计数器停止计数。

域	简称	复位值	类型	功能
位 31: 16	保留位	0x0000	resd	硬件强制为 0。
位 15: 0	DIV	0x0000	rw	分频系数（Division） 这 16 位定义了 USART 分频系数。

## 12.12.4 控制寄存器1（USART\_CTRL1）

域	简称	复位值	类型	功能
位 31: 14	保留位	0x000000	resd	硬件强制为 0。
位 13	UEN	0x0	rw	USART 使能（USART enable） 0: 关闭； 1: 开启。
位 12	DBN	0x0	rw	数据位个数（Data bit num） 该位定义了数据位的个数。 0: 8 位； 1: 9 位。
位 11	WUM	0x0	rw	唤醒方式（Wake up mode） 该位定义静默状态下被唤醒的方式。 0: 空闲帧唤醒； 1: ID 匹配唤醒。
位 10	PEN	0x0	rw	奇偶校验使能（Parity enable） 该位定义使能硬件奇偶校验（对于发送来说就是校验位的产生；对于接收来说就是校验位的检测）。当使能了该

				位，硬件将发送数据的最高位替换成校验位；对接收到的数据检查其校验位是否正确。 0: 关闭； 1: 开启。
位 9	PSEL	0x0	rw	奇偶校验选择 (Parity selection) 该位定义是采用奇校验还是偶校验。 0: 偶校验； 1: 奇校验。
位 8	PERRIEN	0x0	rw	PERR 中断使能 (PERR interrupt enable) 0: 关闭； 1: 开启。
位 7	TDBEIEN	0x0	rw	发送数据缓冲器空中断使能 (TDBE interrupt enable) 0: 关闭； 1: 开启。
位 6	TDCIEN	0x0	rw	发送数据完成中断使能 (TDC interrupt enable) 0: 关闭； 1: 开启。
位 5	RDBFIEN	0x0	rw	接收数据缓冲器满中断使能 (RDBF interrupt enable) 0: 关闭； 1: 开启。
位 4	IDLEIEN	0x0	rw	总线空闲中断使能 (IDLE interrupt enable) 0: 关闭； 1: 开启。
位 3	TEN	0x0	rw	发送使能 (Transmitter enable) 该位定义发送端的使能。 0: 关闭； 1: 开启。
位 2	REN	0x0	rw	接收使能 (Receiver enable) 该位定义接收端的使能。 0: 关闭； 1: 开启。
位 1	RM	0x0	rw	接收静默 (Receiver mute) 该位定义接收端静默的开启，可由软件置起或清零。当配置为空闲帧唤醒时，唤醒后硬件也会将其清零，当配置为匹配地址唤醒时，收到匹配地址唤醒后硬件会将其清零，收到不匹配地址后硬件会再次将其置起进入静默状态。 0: 普通； 1: 静默。
位 0	SBF	0x0	rw	发送间隔帧 (Send break frame) 使用该位来发送间隔帧。该位可以由软件置起或清零。常规用法是软件置起该位，间隔帧发送完成后，由硬件将该位清零。 0: 无； 1: 发送。

## 12.12.5 控制寄存器2 (USART\_CTRL2)

域	简称	复位值	类型	功能
位 31: 16	保留位	0x00000	resd	硬件强制为 0。
位 15	TRPSWAP	0x0	rw	收发管脚交换 (Transmit receive pin swap) 0: 关闭； 1: 开启。
位 14	LINEN	0x0	rw	LIN 模式使能 (LIN mode enable) 0: 关闭； 1: 开启。
位 13: 12	STOPBN	0x0	rw	停止位个数 (STOP bit num) 这 2 位用来设置停止位的个数 00: 1 位； 01: 0.5 位； 10: 2 位； 11: 1.5 位；

位 11	CLKEN	0x0	rw	时钟使能 (Clock enable) 该位用来使能同步模式或智能卡模式的时钟管脚。 0: 关闭; 1: 开启。
位 10	CLKPOL	0x0	rw	时钟极性 (Clock polarity) 在同步模式或智能卡模式下, 可以用该位选择时钟管脚上总线空闲时时钟输出的极性。 0: 低电平; 1: 高电平。
位 9	CLKPHA	0x0	rw	时钟相位 (Clock phase) 在同步模式或智能卡模式下, 可以用该位选择时钟管脚上时钟输出的相位。 0: 第一个边沿进行数据捕获; 1: 第二个边沿进行数据捕获。
位 8	LBCP	0x0	rw	最后一位时钟脉冲 (Last bit clock pulse) 在同步模式下, 使用该位来控制是否在时钟管脚上输出数据的最后一位对应的时钟脉冲 0: 不输出; 1: 输出。
位 7	保留位	0x0	resd	保持默认值。
位 6	BFIEN	0x0	rw	间隔帧中断使能 (break frame interrupt enable) 0: 关闭; 1: 开启。
位 5	BFBN	0x0	rw	间隔帧位数 (break frame bit num) 该位用来选择是 11 位还是 10 位的间隔帧。 0: 10 位; 1: 11 位。
位 4	保留位	0x0	resd	保持默认值。
位 3: 0	ID	0x0	rw	USART 的 ID 号 (USART identification) 可配置的 USART 的 ID 号。

注意： 在使能发送后不能改写这三个位 (CLKPOL、CLKPHA、LBCP)。

## 12.12.6 控制寄存器3 (USART\_CTRL3)

域	简称	复位值	类型	功能
位 31: 11	保留位	0x000000	resd	硬件强制为 0。
位 10	CTSCFIEN	0x0	rw	CTSCF 中断使能 (CTSCF interrupt enable) 0: 关闭; 1: 开启。
位 9	CTSEN	0x0	rw	CTS 使能 (CTS enable) 0: 关闭; 1: 开启。
位 8	RTSEN	0x0	rw	RTS 使能 (RTS enable) 0: 关闭; 1: 开启。
位 7	DMATEN	0x0	rw	DMA 发送使能 (DMA transmit enable) 0: 关闭; 1: 开启。
位 6	DMAREN	0x0	rw	DMA 接收使能 (DMA receiver enable) 0: 关闭; 1: 开启。
位 5	SCMEN	0x0	rw	智能卡模式使能 (Smart card mode enable) 0: 关闭; 1: 开启。
位 4	SCNACKEN	0x0	rw	智能卡 NACK 使能 (Smart card NACK enable) 该位用于配置校验错误出现时, 发送 NACK。 0: 不发送; 1: 发送。
位 3	SLBEN	0x0	rw	单线双向半双工模式使能 (Single line bidirectional half-duplex enable) 0: 关闭;

				1: 开启。 红外低功耗模式配置 (IrDA low-power mode) 该位用来配置红外低功耗模式。 0: 关闭; 1: 开启。
位 2	IRDALP	0x0	rw	红外功能使能 (IrDA enable) 0: 关闭; 1: 开启。
位 1	IRDAEN	0x0	rw	错误中断使能 (Error interrupt enable) 当有帧错误、接收溢出错误或者杂讯错误时产生中断。 0: 关闭; 1: 开启。
位 0	ERRIEN	0x0	rw	

## 12.12.7 保护时间和预分频寄存器 (USART\_GDIV)

域	简称	复位值	类型	功能
位 31: 16	保留位	0x0000	resd	硬件强制为 0。
位 15: 8	SCGT	0x00	rw	智能卡保护时间值 (Smart card guard time) 在智能卡模式下, 当保护时间过去后, 才会设置发送完成标志, 这几位配置保护时间值。
位 7: 0	ISDIV	0x00	rw	红外或者智能卡分频系数 (IrDA/smartcard division) 红外 (IrDA) 模式: 8 位[7: 0]有效, 普通模式无效且只能设置为 00000001, 低功耗模式分频系数对外设时钟进行分频, 作为脉冲宽度的基数周期; 00000000: 保留 - 不要写入该值; 00000001: 1 分频; 00000010: 2 分频; ..... 智能卡模式: 低 5 位[4: 0]有效, 分频系数对外设时钟进行分频, 给智能卡提供时钟。可以设置为如下值: 00000: 保留 - 不要写入该值; 00001: 2 分频; 00010: 4 分频; 00011: 6 分频; .....

# 13 串行外设接口 (SPI)

## 13.1 串行外设接口 (SPI) 简介

SPI 接口提供软件编程配置选项，根据软件编程配置方式不同，可以分别作为 SPI 和 I<sup>2</sup>S 使用。本章将分 SPI 和 I<sup>2</sup>S 分别介绍 SPI 作 SPI 或 I<sup>2</sup>S 的功能特性以及配置流程。

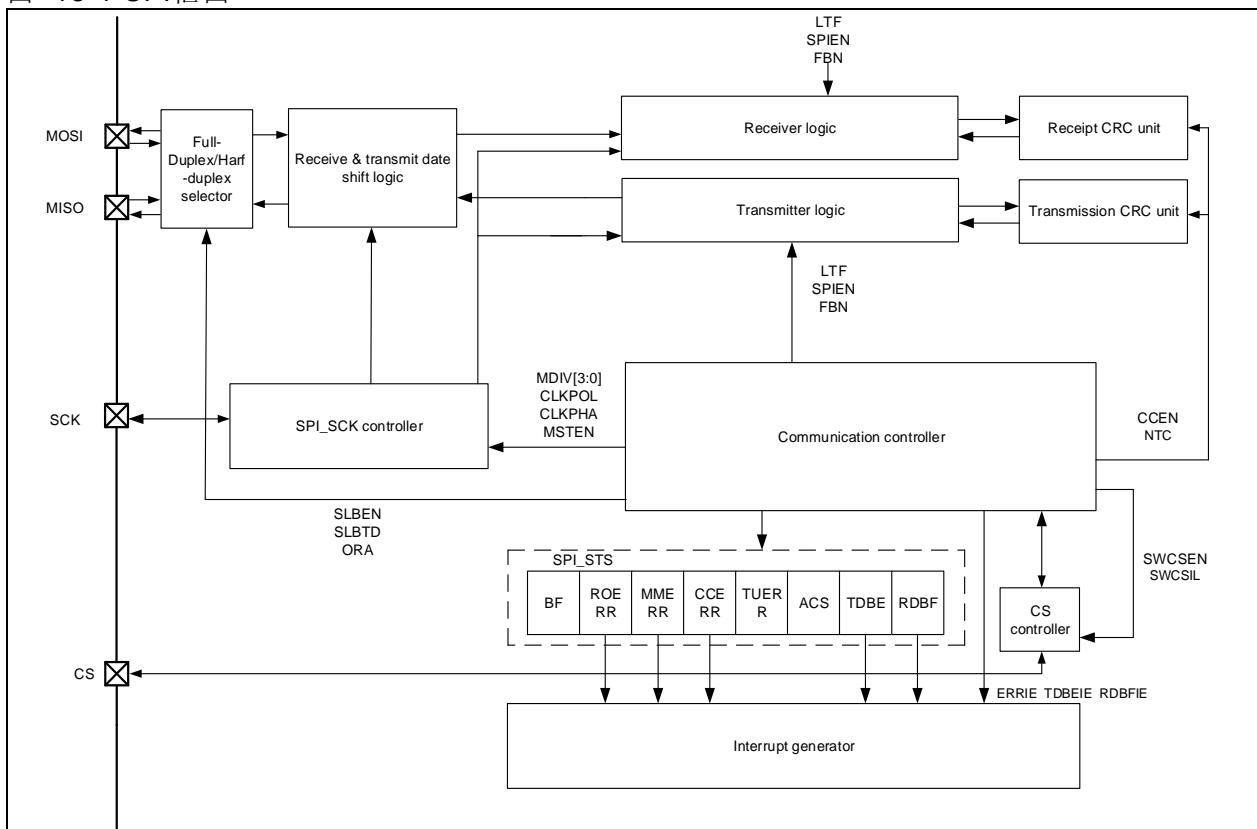
## 13.2 SPI 功能描述

### 13.2.1 SPI 简述

串行外设接口 (SPI) 根据软件编程配置的方式不同，可以分别作为主机和从机使用，又可以分别工作在全双工，全双工只收，半双工只发/只收四种不同的模式下，并且还提供 DMA 传输，SPI 内部硬件自动 CRC 计算和校验等功能。

**SPI 的架构框图见下图：**

图 13-1 SPI 框图



**SPI 接口作为 SPI 使用时主要特征如下：**

- 可编程配置的全双工或半双工通信
  - 全双工同步通信（可以选择全双工只收以此释放用于发送的 IO）
  - 半双工同步通信（可以根据软件编程配置选择传输方向：发送或接收）
- 可编程配置主/从模式
- 可编程配置的 CS 信号处理方式
  - 硬件处理 CS
  - 软件处理 CS
- 可编程配置的 8 位或 16 位帧位数
- 可编程配置的通信频率以及分频系数（最大分频系数为 f<sub>PCLK</sub>/2）
- 可编程配置的时钟极性和相位
- 可编程配置的数据传输顺序(先发 MSB/LSB)
- 可编程配置的错误中断标志（接收器溢出错误，主模式错误，CRC 校验错误）
- 可编程配置的发送数据缓冲器空中断以及接收数据缓冲器满中断

- 支持 DMA 发送和接收
- 支持硬件 CRC 发送和校验
- 具备通信忙标志

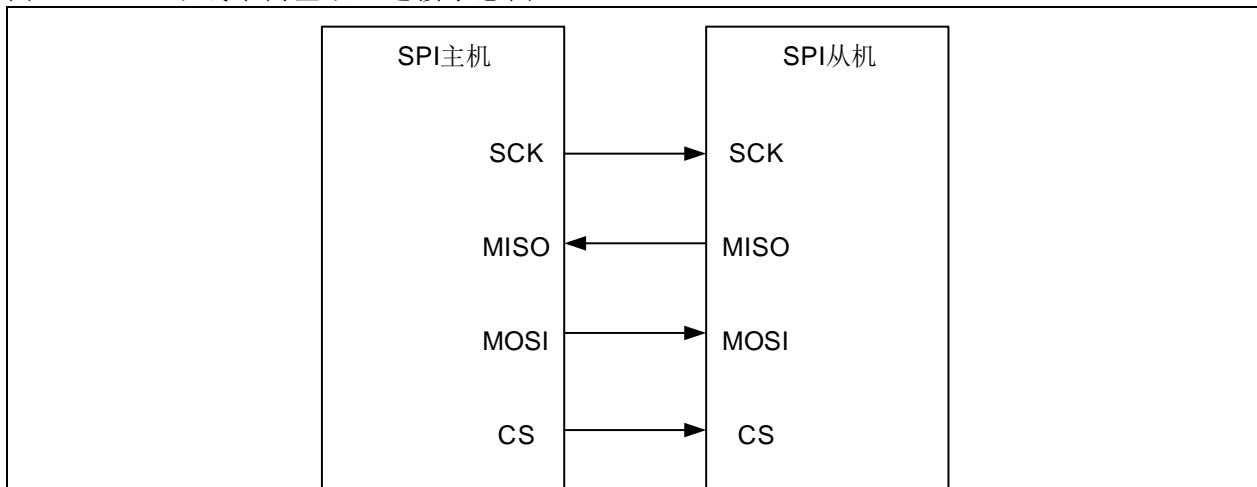
### 13.2.2 全双工半双工选择器简述和配置流程

SPI 全双工半双工选择器通过软件编程配置的方式，可以使 SPI 接口作为 SPI 使用时，可以工作在双线单向全双工，单线单向只收，单线双向半双工发送和单线双向半双工接收四种同步模式。

**双线单向全双工模式配置方式以及 SPI IO 连接方式如下：**

SLBEN 位置 0，ORA 置 0 时，SPI 工作在双线单向全双工，此时 SPI 可以同时进行数据的收发，IO 连接方式如下图。

图 13-2 SPI 双线单向全双工连接示意图



SPI 作主机或从机在此模式下，关闭 SPI 或进入省电模式（或关闭 SPI 系统时钟）之前需要等待 RDBF 置位，TDBE 置位，并等待 BF=0。

**单线双向只收模式配置方式以及 SPI IO 连接方式如下：**

SLBEN 位置 0，ORA 置 1 时，SPI 工作在单线双向只收模式，此时 SPI 只能作为数据接收方，无法发送数据。作为主机时使用 MISO 接收数据，MOSI 管脚所映射的 IO 释放。作为从机时使用 MOSI 接收数据，MISO 管脚所映射的 IO 释放。

图 13-3 SPI 作主机单线双向只收连接示意图

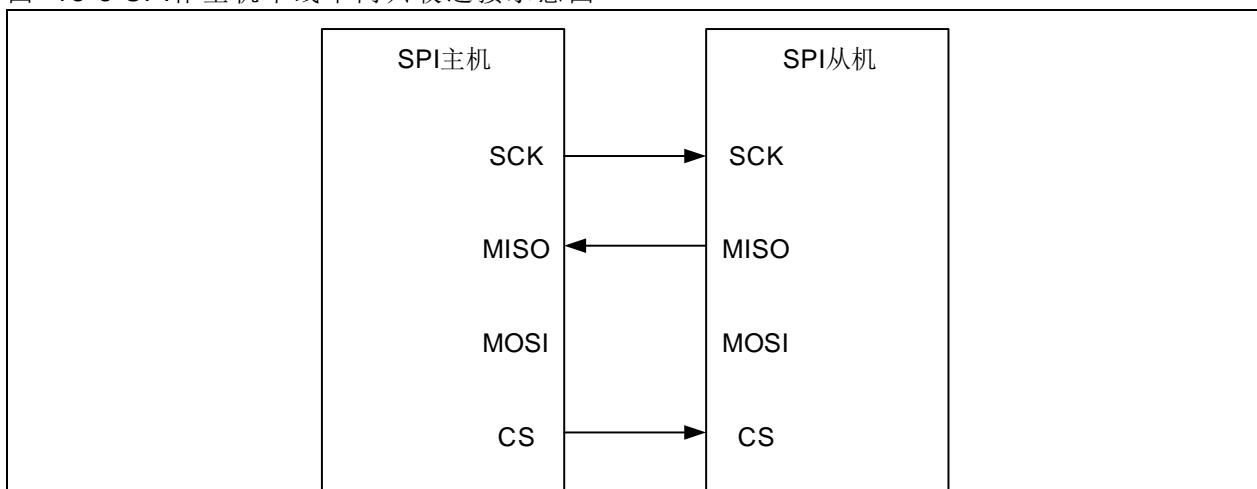
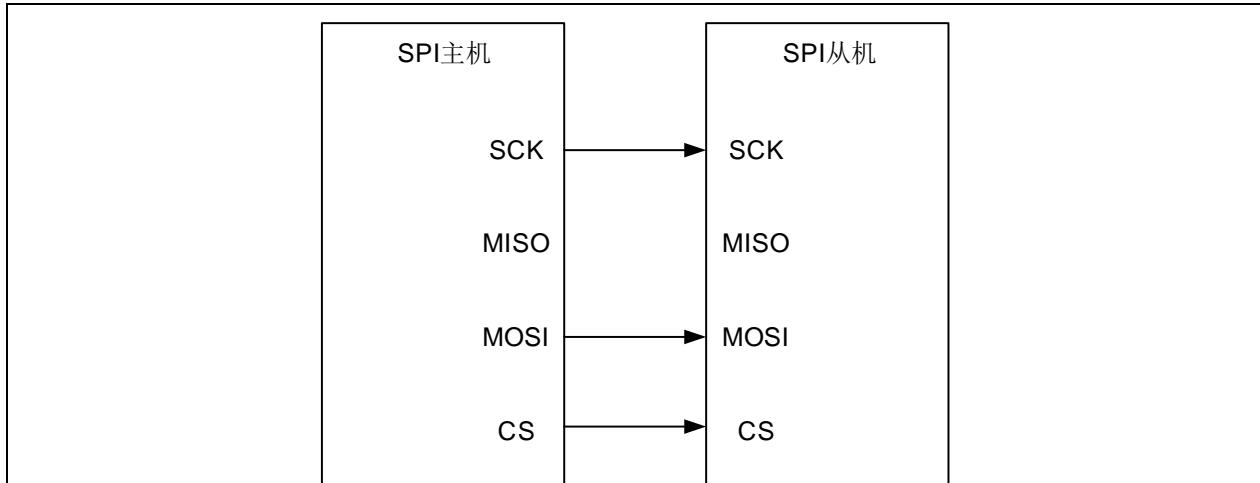


图 13-4 SPI作从机单线单向只收连接示意图



SPI 作主机时，在此模式下，需要等待倒数第二个 RDBF 置起，关闭 SPI 之前等待一个 SPI\_SCK 周期，在进入省电模式(或关闭 SPI 系统时钟)之前等待最后一个 RDBF=1。

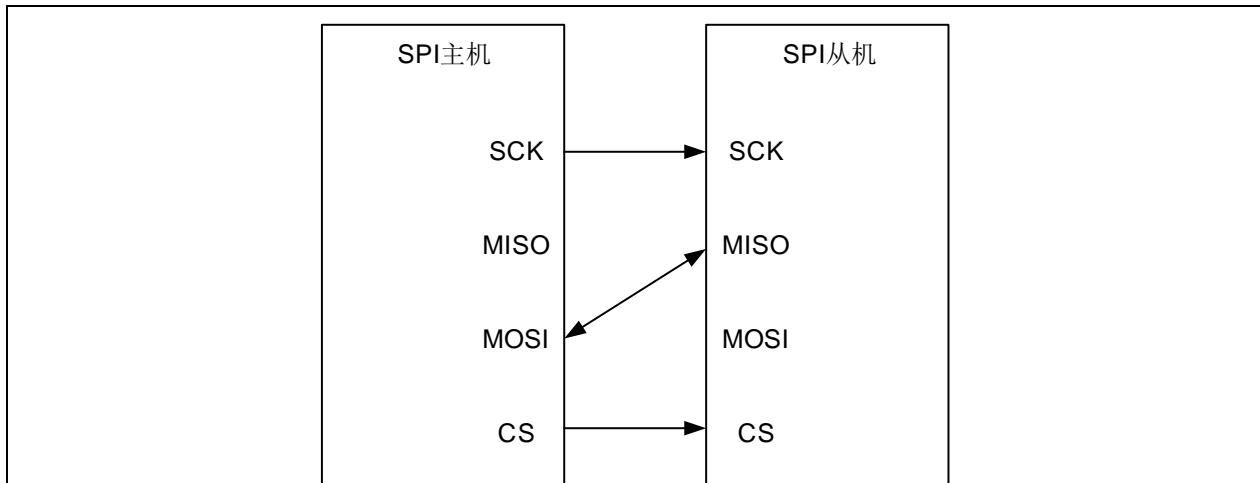
SPI 作从机时，在此模式下，关闭 SPI 无需判断任何标志，但是在进入省电模式(或关闭 SPI 系统时钟)之前需要等待 BF=0。

**单线双向半双工模式配置方式以及 SPI IO 连接方式如下：**

SLBEN 位置 1 时，SPI 工作在单线双向半双工模式，此时 SPI 可以分时进行数据收发。作为主机时使用 MOSI 收发数据，MISO 管脚所映射的 IO 释放。作为从机时使用 MISO 收发数据，MOSI 管脚所映射的 IO 释放。

软件通过编程控制 SLBTD 位控制传输方向，SLBTD 位置 1 时，SPI 只能发送数据，SLBTD 位置 0 时，SPI 只能接收数据。

图 13-5 SPI作单线双向半双工连接示意图



SPI 作主机或从机时，在单线双向半双工，传输方向选择为发送时，需要等待 TDBE 置位，BF=0 后才能关闭 SPI，在关闭 SPI 后才可以进入省电模式(或关闭 SPI 系统时钟)。

SPI 作主机时，在单线双向半双工，传输方向选择为接收时，需要等待倒数第二个 RDBF 置起，关闭 SPI 之前等待一个 SPI\_SCK 周期，在进入省电模式(或关闭 SPI 系统时钟)之前等待最后一个 RDBF=1。

SPI 作从机时，在单线双向半双工，传输方向选择为接收时，关闭 SPI 无需判断任何标志，但是在进入省电模式(或关闭 SPI 系统时钟)之前需要等待 BF=0。

### 13.2.3 CS控制器简述和配置流程

SPI 的 CS 控制器提供通过软件可编程配置的方式选择硬件控制 CS 信号或软件控制 CS 信号，以此实现 CS 信号的控制，用于多处理器模式下主机从机选择，以及通过 CS 信号后于 SCK 信号使能，有效地屏蔽总线上的干扰，下面将分软件 CS 以及硬件 CS 来介绍 CS 控制器的配置流程，并会简述在主机和从机模式下软件和硬件 CS 的输入输出方式。

**硬件 CS 配置流程：**

当 SPI 作主机，硬件 CS 输出时，HWCSOE 位置 1，SWCSEN 置 0，开启硬件 CS 控制，SPI 在使能之

后会在 CS 管脚上输出低电平，在 SPI 关闭并且发送完成后，释放 CS 信号。

当 SPI 作主机，硬件 CS 输入时，HWCSOE 位置 0，SWCSEN 置 0，开启硬件 CS 控制，此时一旦主机 SPI 检测到 CS 管脚为低电平时，SPI 硬件自动关闭 SPI 并进入从机模式，模式错误标志 MMERR 同时置位，若使能了错误中断（ERRIE=1），将会产生中断，在 MMERR 置位期间，硬件不允许软件置位 SPIEN 和 MSTEN 位，通过读或写 SPI 状态寄存器（SPI\_STS）再写 SPI 控制寄存器 1（SPI\_CTRL1）可以清除 MMERR。

当 SPI 作从机，硬件 CS 输入时，HWCSOE 位置 0，SWCSEN 置 0，开启硬件 CS 控制，从机根据 CS 管脚上的电平判断是否发送或接收数据，只有 CS 管脚上为低电平时，从机才会被选中并进行数据的收发。

#### 软件 CS 配置流程：

当 SPI 作主机，软件 CS 输入时，SWCSEN 位置 1，开启软件 CS 控制，当 SWCSIL 位置 0 时，SPI 硬件自动关闭 SPI 并进入从机模式，模式错误标志 MMERR 同时置位，若使能了错误中断（ERRIE=1），将会产生中断，在 MMERR 置位期间，硬件不允许软件置位 SPIEN 和 MSTEN 位，通过读或写 SPI 状态寄存器（SPI\_STS）再写 SPI 控制寄存器 1（SPI\_CTRL1）可以清除 MMERR。

当 SPI 作从机，软件 CS 输入时，SWCSEN 位置 1，开启软件 CS 控制，SPI 根据 SWCSIL 位判断 CS 信号电平，不使用 CS 管脚，当 SWCSIL=0 时，从机才会被选中并进行数据的收发。

### 13.2.4 SPI\_SCK控制器简述和配置流程

SPI 协议采用同步传输，所以 SPI 接口在作为 SPI 使用时，作主机时，需要产生通信时钟用于 SPI 接口的数据收发，并且需要将该通信时钟通过 IO 输出给从机，用于从机的数据收发；作从机时，需要外设提供通信时钟从 IO 输入到 SPI 接口内部作为通信时钟使用，所以实际上，SPI\_SCK 控制器便是扮演着产生 SPI\_SCK 以及分配 SPI\_SCK 的角色，详细的配置方法如下所述。

#### SPI\_SCK 控制器配置流程：

- 时钟极性相位选择：配置 CLKPOL,CLKPHA 选择需要的极性和相位。
- 时钟分频系数选择：配置 CRM 选择需要的 PCLK 频率，配置 MDIV[3: 0]选择需要的分频系数。
- 主机或从机选择：配置 MSTEN 选择 SPI 作主机或从机使用，注意主机只收模式在 SPI 使能后就会开始输出时钟，直到 SPI 被关闭且接收完成。

### 13.2.5 CRC简述和配置流程

SPI 接口内部具有独立的发送和接收 CRC 计算单元，通过软件编程配置，SPI 接口在作为 SPI 使用时，可以同时在用户使用 DMA 读写数据或 CPU 读写数据的情况下，自动进行 CRC 计算以及 CRC 校验，如果在传输过程中，硬件检测到接收到的数据与 SPI\_RCRC 中的数据不符，且该笔数据又是 CRC 数据时，CCERR 位会置起，若使能了错误中断（ERRIE=1），将会产生中断。

下面分 DMA 和 CPU 操作数据寄存器分别描述 SPI 的 CRC 功能以及 CRC 配置流程。

#### CRC 配置流程

- CRC计算多项式配置：配置 SPI\_CPOLY 选择 CRC 计算多项式。
- 使能CRC：置起 CCEN 位使能 CRC 计算，该操作将会复位 SPI\_RCRC 以及 SPI\_TCRC。
- 根据 DMA 或 CPU 操作数据寄存器选择是否以及何时置位 NTC 位，具体请参见下方描述。

#### DMA 发送模式：

在采用 DMA 写入待发送的数据时，当使能 CCEN 后，硬件会根据 SPI\_CPOLY 中的值以及每笔发送的数据自动计算 CRC 值，并在最后一笔数据发送完成后自动发送 CRC 值，该值即 SPI\_TCRC 中的值。

#### DMA 接收模式：

在采用 DMA 读取待接收的数据时，当使能 CCEN 后，硬件会根据 SPI\_CPOLY 中的值以及每笔接收的数据自动计算 CRC 值，并在最后一笔数据接收完成后等待 CRC 数据接收完成，并将收到的 CRC 值和 SPI\_RCRC 中的值作比较，若校验出错，会置起 CCERR 标志，若使能了 ERRIE 位，则产生错误中断。

#### CPU 发送模式：

相较于 DMA 发送模式，该模式需要软件在写入最后一笔待发送的数据后，在最后一笔数据发送完成之前置起 NTC 位。

#### CPU 接收模式：

在双线单向全双工模式下，按照 CPU 发送模式操作 NTC 位，CPU 接收模式的 CRC 计算和校验会自动完成，在单线单向只接收以及单线双向只接收模式下，相较于 DMA 接收需要软件在接收到倒数第二笔数

据之后，接收到最后一笔数据之前置起 NTC 位。

### 13.2.6 DMA传输简述和配置流程

SPI 接口支持使用 DMA 进行发送数据的写入，接收数据的读取，具体配置流程分别见下述的 DMA 发送配置流程以及 DMA 接收配置流程。

需要特别注意的是，在开启CRC计算和校验时，DMA发送数据的个数配置为待发送的数据个数，DMA 读取数据的个数配置为待接收的数据个数，此时硬件在所有数据传输完毕后自动进行CRC传输，且接收方还会自动进行CRC校验，需要注意，接收到的CRC数据，硬件会搬到SPI数据寄存器（SPI\_DT）中，并置位RDBF，以及在开启了DMA传输时发出DMA读请求，所以这里推荐当CRC接收完毕后软件要去读 DT寄存器来取走CRC值，防止后续传输出错。

**DMA发送配置流程：**

- 选择 DMA 传输通道：在 DMA 章节 DMA 通道映射表中选择用于当前所用 SPI 的 DMA 通道。
- 配置 DMA 传输目标地址：在 DMA 控制寄存器中 DMA 传输目的地址位写入当前所使用的 SPI 的 SPI 数据寄存器（SPI\_DT）地址，DMA 将会在接收到发送请求后将待发送的数据写入该地址。
- 配置 DMA 传输源地址：在 DMA 控制寄存器中 DMA 传输源地址位写入待发送数据存放的地址，DMA 将会在接收到发送请求后将该地址内的数据写入到目标地址中，即写入到当前所使用的 SPI 的 SPI 数据寄存器（SPI\_DT）中。
- 配置 DMA 传输数据个数：在 DMA 控制寄存器相关位置配置期望传输的数据个数
- 配置 DMA 传输通道优先级：在 DMA 控制寄存器相关位置配置当前所使用通道的 SPI 的 DMA 传输通道优先级。
- 配置 DMA 中断产生时机：在 DMA 控制寄存器相关位置配置是在传输完成或传输完成一半时产生 DMA 中断。
- 使能 DMA 传输通道：在 DMA 控制寄存器相关位置使能当前所选用的 DMA 通道

**DMA接收配置流程：**

- 选择 DMA 传输通道：在 DMA 章节 DMA 通道映射表中选择用于当前所用 SPI 的 DMA 通道。
- 配置 DMA 传输目标地址：在 DMA 控制寄存器中 DMA 传输目的地址位写入期望存放接收数据的地址，DMA 将会在接收到接收请求后，将当前所使用的 SPI 的 SPI 数据寄存器（SPI\_DT）中的数据存放在目的地址中。
- 配置 DMA 传输源地址：在 DMA 控制寄存器中 DMA 传输源地址位写入当前所使用的 SPI 的 SPI 数据寄存器（SPI\_DT）的地址，DMA 将会在接收到接收请求后将该地址内的数据写入到目标地址中，即写入到期望存放接收数据的地址。
- 配置 DMA 传输数据个数：在 DMA 控制寄存器相关位置配置期望传输的数据个数。
- 配置 DMA 传输通道优先级：在 DMA 控制寄存器相关位置配置当前所使用通道的 SPI 的 DMA 传输通道优先级。
- 配置 DMA 中断产生时机：在 DMA 控制寄存器相关位置配置是在传输完成或传输完成一半时产生 DMA 中断。
- 使能 DMA 传输通道：在 DMA 控制寄存器相关位置使能当前所选用的 DMA 通道

### 13.2.7 发送器简述和配置流程

SPI 发送器时钟由 SPI\_SCK 控制器提供，根据软件编程配置，发送器可以输出不同的数据帧格式，SPI 具有一个数据缓冲寄存器 SPI\_DT，软件需要将待发送的数据先写入 SPI\_DT，发送器在有时钟时，会把 SPI 数据寄存器（SPI\_DT）中的数据保存到发送器中的数据缓冲器(有别于 SPI 数据寄存器（SPI\_DT），SPI 发送器中的数据缓冲器由 SPI\_SCK 驱动，且硬件自动控制，软件不可操作)，并按照配置好的帧格式将数据依次发出。

用户可以选择 DMA 或 CPU 来控制数据的写入，若选择 DMA 传输，详细配置请参见 DMA 传输章节，若选择 CPU 传输，则用户需要判断 TDBE 位，该位复位值为 1，代表 SPI\_DT 为空，若 TDBEIE 置位，则产生中断，数据写入后，TDBE 拉低，直到数据被同步到发送器中的数据缓冲器后，TDBE 再次被拉起，即用户只可以在 TDBE 置位时写入待发送的数据。

发送器配置完成并使能 SPI 后，SPI 将进入数据发送状态，所以在此之前，应需要参考全双工半双工章节

配置通信选用的是全双工或半双工等，并参考 CS 控制器章节配置选用的 CS 控制模式，还需要参考 SPI\_SCK 控制章节配置通信时钟，若使用了 CRC 以及 DMA，还需参考 CRC 以及 DMA 传输章节配置 CRC 以及 DMA，如下为推荐的发送器配置流程。

#### 发送器配置流程：

- 配置全双工半双工选择器。
- 配置 CS 控制器
- 配置 SPI\_SCK 控制器
- 配置 CRC（若需要使用 CRC 自动计算和校验功能）
- 配置 DMA 传输（若需要使用 DMA 传输功能）
- 若没有选择 DMA 传输功能，软件需要判断 TDBE 位，软件需要根据需求判断是否要打开发送数据中断，即置位 TDBIE。
- 配置帧格式：配置 LTF 位选择 MSB/LSB 格式，配置 FBN 选择 8/16 位数据。
- 置位 SPIEN 位使能 SPI

### 13.2.8 接收器简述和配置流程

SPI 接收器时钟由 SPI\_SCK 控制器提供，根据软件编程配置，接收器可以接收不同的数据帧格式，SPI 接收器具有一个接收数据缓冲寄存器，该寄存器由 SPI\_SCK 驱动，在每笔传输的最后一个 CLK，数据从移位寄存器压入该接收数据缓冲寄存器，随后发送器会给出数据接收完成的标志给到 SPI 的控制逻辑，SPI 的控制逻辑在检测到该标志后会自动把接收器中的数据缓冲器中的值压入 SPI\_DT，RDBF 随之置起，这意味着有数据被收到，且该数据已被压入 SPI 数据寄存器(SPI\_DT)，此时读 SPI 数据寄存器(SPI\_DT) 可以读出该笔数据，同时 RDBF 随之清除。

用户可以选择 DMA 或 CPU 来控制数据的读出，若选择 DMA 传输，详细配置请参见 DMA 传输章节，若选择 CPU 传输，则用户需要判断 RDBF 位，该位复位值为 0，代表 SPI\_DT 为空，当有数据被接收到，且数据被移入 SPI 数据寄存器 (SPI\_DT) 时，RDBF 置位，代表 SPI 数据寄存器 (SPI\_DT) 内有数据等待读取，此时若 RDBFIE 置位则产生中断。

若在下一笔接收器接收到的数据准备压入 SPI 数据寄存器(SPI\_DT)时，之前接收到的数据仍未被读走，即 RDBF 仍为 1，则代表数据溢出，在此之前接收到的数据不会丢失，但之后的数据都将丢失，此时 ROERR 置起，若 ERRIE 置位，则产生错误中断，依次读 SPI 数据寄存器(SPI\_DT)和 SPI 状态寄存器(SPI\_STS)可将 ROERR 清除，如下为推荐的接收器配置流程。

#### 接收器配置流程：

- 配置全双工半双工选择器。
- 配置 CS 控制器。
- 配置 SPI\_SCK 控制器。
- 配置 CRC（若需要使用 CRC 自动计算和校验功能）。
- 配置 DMA 传输（若需要使用 DMA 传输功能）。
- 若没有选择 DMA 传输功能，软件需要判断 RDBF 位，软件需要根据需求判断是否要打开接收数据中断，即置位 RDBFIE。
- 配置帧格式：配置 LTF 位选择 MSB/LSB 格式，配置 FBN 选择 8/16 位数据。
- 置位 SPIEN 位使能 SPI。

### 13.2.9 Motorola模式通信时序

本节介绍 SPI 通信时序，包括全双工和半双工的主/从通信时序。

#### 全双工通信-主机通信时序

其中主机端配置如下：

MSTEN=1：设备为主机；

SLBEN=0：全双工模式；

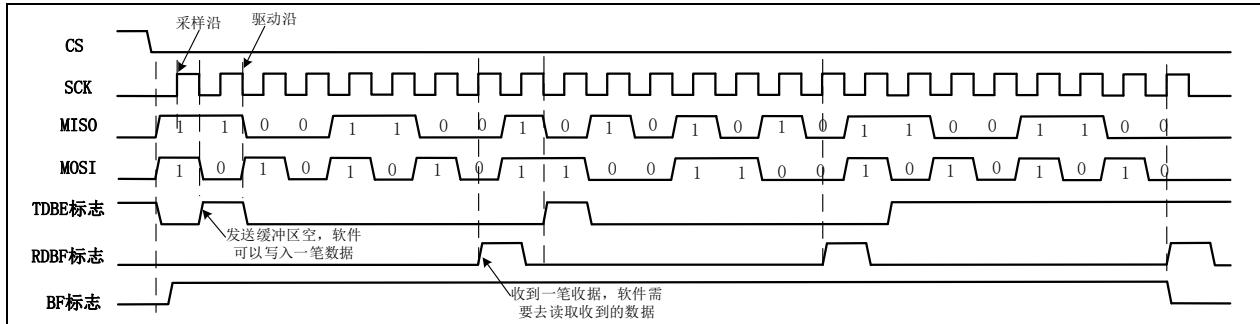
CLKPOL=0, CLKPHA=0：SCK 空闲输出低电平，第一个边沿作为采样边沿；

FBN=0：帧数据的长度为 8 位；

主机发送数据 (MOSI): 0xaa, 0xcc, 0xaa

从机发送数据 (MISO): 0xcc, 0xaa, 0xcc

图 13-6 主机全双工通信



### 全双工通信-从机通信时序

其中从机端配置如下：

MSTEN=0: 设备为从机；

SLBEN=0: 全双工模式；

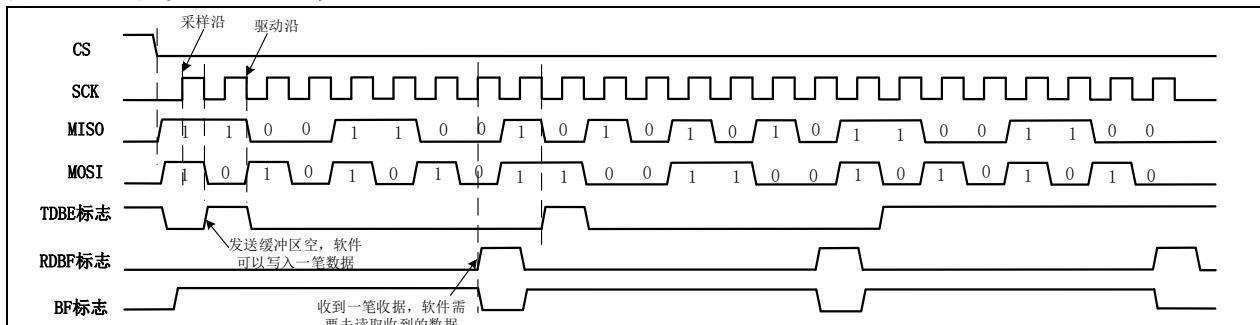
CLKPOL=0, CLKPHA=0: SCK 空闲输出低电平，第一个边沿作为采样边沿；

FBN=0: 帧数据的长度为 8 位；

主机发送数据 (MOSI): 0xaa, 0xcc, 0xaa

从机发送数据 (MISO): 0xcc, 0xaa, 0xcc

图 13-7 从机全双工通信



### 半双工通信-主机发送时序

MSTEN=1: 设备为主机；

SLBEN=1: 单线双向模式；

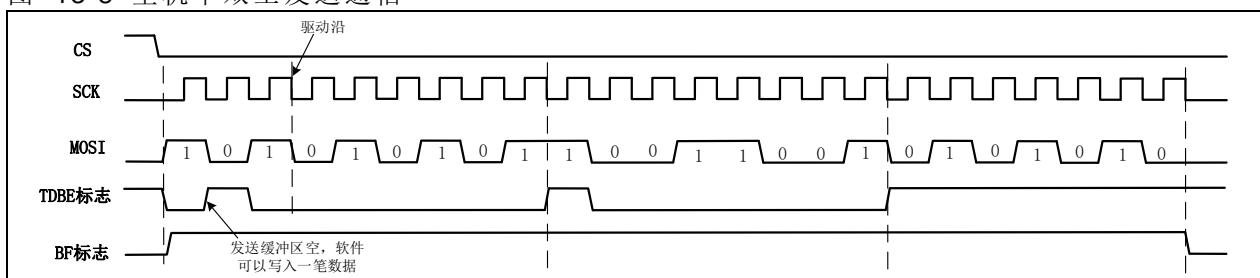
SLBTD=1: 发送模式；

CLKPOL=0, CLKPHA=0: SCK 空闲输出低电平，第一个边沿为采样边沿；

FBN=0: 帧数据的长度为 8 位；

主机发送数据: 0xaa, 0xcc, 0xaa

图 13-8 主机半双工发送通信



### 半双工通信-从机接收时序

MSTEN=0: 设备为从机；

SLBEN=1: 单线双向模式；

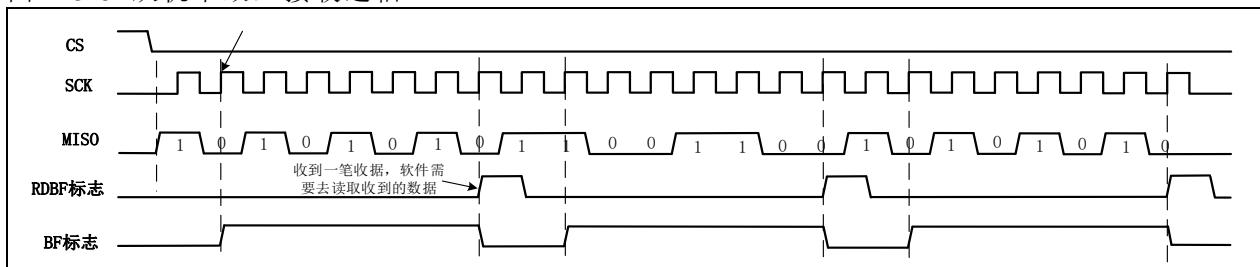
SLBTD=0: 接收模式；

CLKPOL=0, CLKPHA=0: SCK 空闲输出低电平，第一个边沿为采样边沿；

FBN=0: 帧数据的长度为 8 位；

从机接收数据: 0xaa, 0xcc, 0xaa

图 13-9 从机半双工接收通信



### 半双工通信-从机发送时序

MSTEN=0: 设备为从机;

SLBEN=1: 单线双向模式;

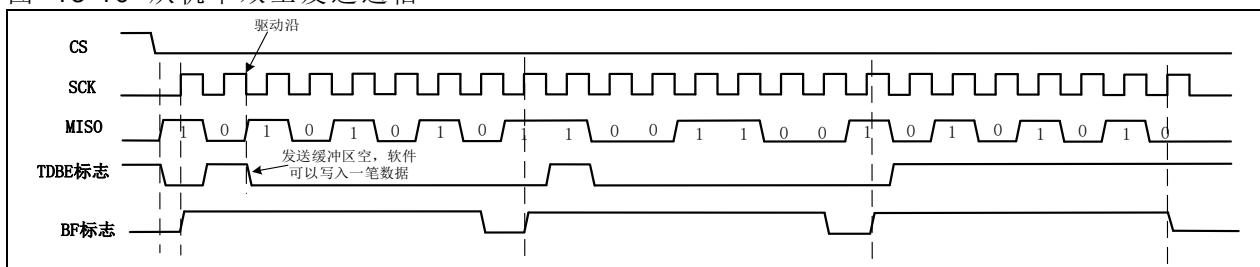
SLBTD=1: 发送模式;

CLKPOL=0, CLKPHA=0: SCK 空闲输出低电平, 第一个边沿为采样边沿;

FBN=0: 帧数据的长度为 8 位;

从机发送数据: 0xaa, 0xcc, 0xaa

图 13-10 从机半双工发送通信



### 半双工通信-主机接收时序

MSTEN=1: 设备为主机;

SLBEN=1: 单线双向模式;

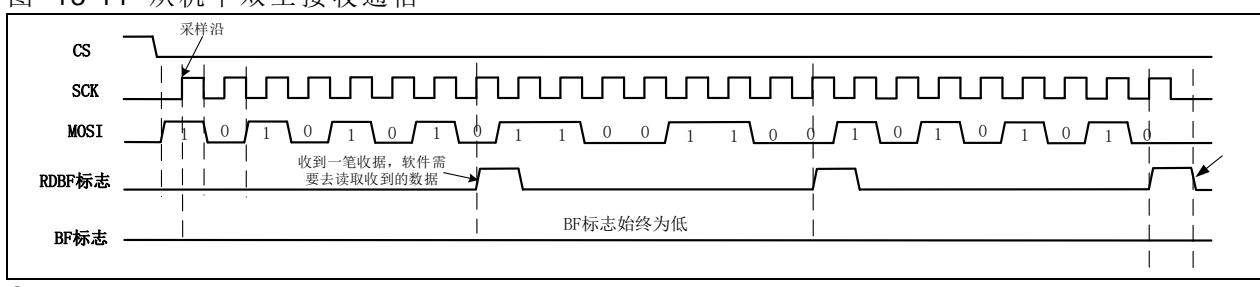
SLBTD=0: 接收模式;

CLKPOL=0, CLKPHA=0: SCK 空闲输出低电平, 第一个边沿为采样边沿;

FBN=0: 帧数据的长度为 8 位;

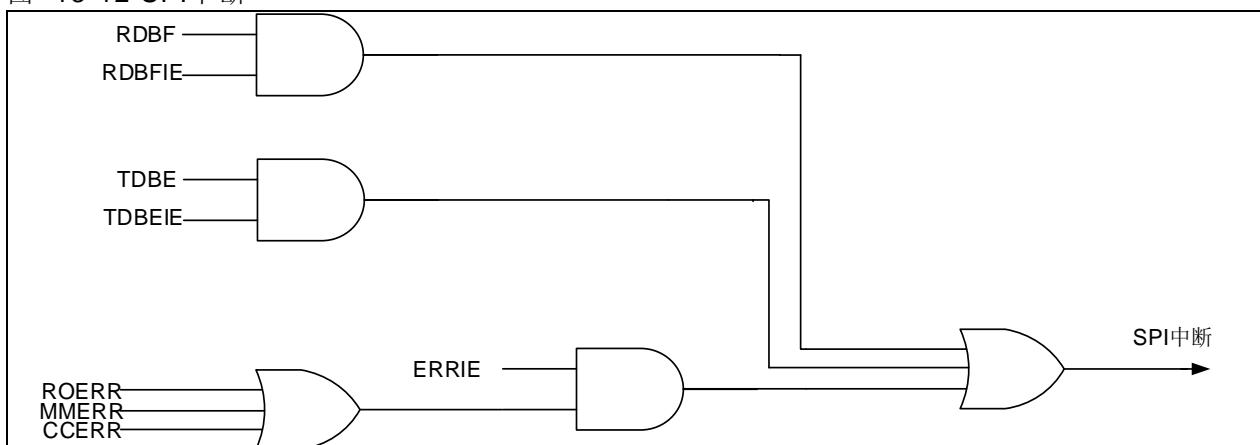
主机接收数据: 0xaa, 0xcc, 0xaa

图 13-11 从机半双工接收通信



### 13.2.10 中断

图 13-12 SPI中断



### 13.2.11 IO管脚控制

SPI 接口作为 SPI 使用时最多可有 4 根管脚与外设相连，各管脚的使用方法可以参见全双工半双工选择器简述和配置流程以及 CS 控制器简述和配置流程章节，各管脚的定义如下。

- **MISO:** 主机输入/从机输出管脚。在 SPI 接口作 SPI 主机使用时，从机送出的数据从该管脚输入。在 SPI 接口作 SPI 从机使用时，从机待发送的数据从该管脚输出。
- **MOSI:** 主设备输出/从设备输入管脚。在 SPI 接口作 SPI 主机使用时，主机待发送的数据从该管脚输出。在 SPI 接口作 SPI 从机使用时，主机送出的数据从该管脚输入。
- **SCK:** SPI 的通信时钟管脚。在 SPI 接口作 SPI 主机使用时，通信时钟从此管脚输出送给外设。在 SPI 接口作 SPI 从机使用时，主机提供的通信时钟从该管脚输入以作为 SPI 接口的通信时钟。
- **CS:** 片选信号。这是一个可选的管脚，用来选中主/从设备，具体使用方式可以参见 CS 控制器章节。
- 注：PA13/14 是 SWDIO 和 SWCK 和 SPI2 的 MISO/MOSI 重合，因此需要将 PA13/14 复用为 SPI2 时，需要在 remap GPIO 前加一段延时以给代码下载时间。

### 13.2.12 注意事项

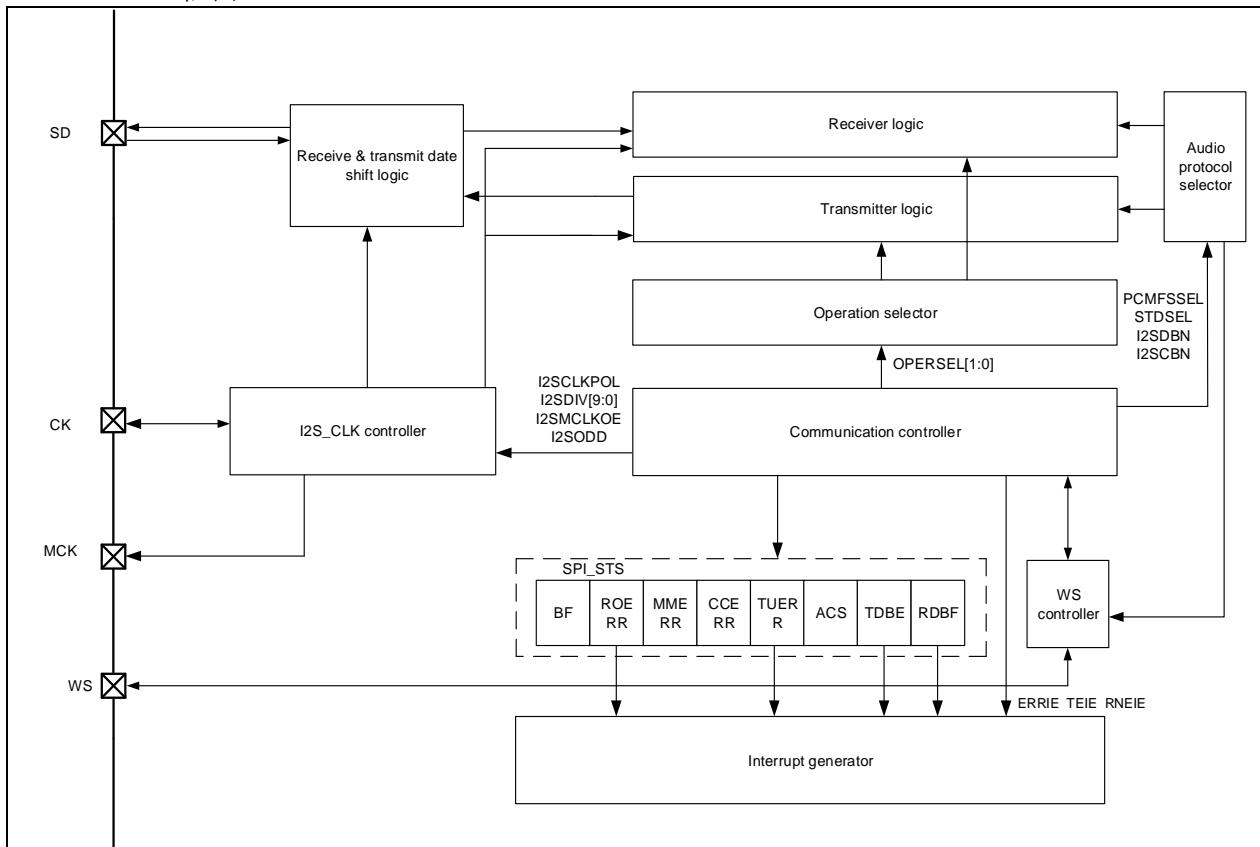
CRC 接收完成后要软件读 DT 寄存器来读出 CRC 值。

## 13.3 I<sup>2</sup>S功能描述

### 13.3.1 I<sup>2</sup>S简述

I<sup>2</sup>S 根据软件配置的不同，可以分别工作在主机接收，主机发送，从机接收，从机发送四种操作模式，并且可以分别支持包括飞利浦标准，高字节对齐标准，低字节对齐标准，PCM 标准在内的共四种音频标准，并同时支持 DMA 传输。

I<sup>2</sup>S 的框图如下图所示：

图 13-13 I<sup>2</sup>S 框图

**SPI 接口作为 I<sup>2</sup>S 使用时主要特征如下：**

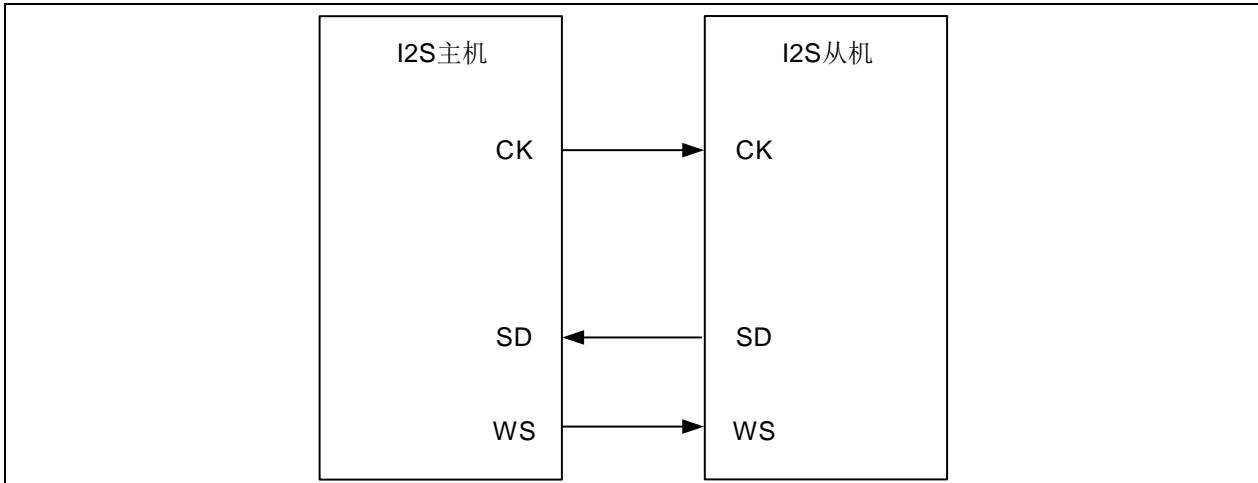
- 可编程配置的操作模式
  - 从设备发送
  - 从设备接收
  - 主设备发送
  - 主设备接收
- 可编程配置的时钟极性
- 可编程配置的时钟频率（8KHz 到 192KHz）
- 可编程配置的数据位数（16 位， 24 位， 32 位）
- 可编程配置的声道位数（16 位， 32 位）
- 可编程配置的音频协议
  - I<sup>2</sup>S 飞利浦标准
  - 高字节对齐标准（左对齐）
  - 低字节对齐标准（右对齐）
  - PCM 标准（带长或短帧同步的通道帧）
- 支持 DMA 传输
- 支持提供频率固定比例为 256 倍 Fs（音频采样频率）的外设主时钟

### 13.3.2 操作模式选择器简述和配置流程

SPI 接口作 I<sup>2</sup>S 选择器使用时提供了多种操作模式，用户可以通过软件编程控制操作模式选择器，选择需要的操作模式，本节会分从设备发送，从设备接收，主设备发送，主设备接收四种操作模式简单介绍配置流程以及连接方式。

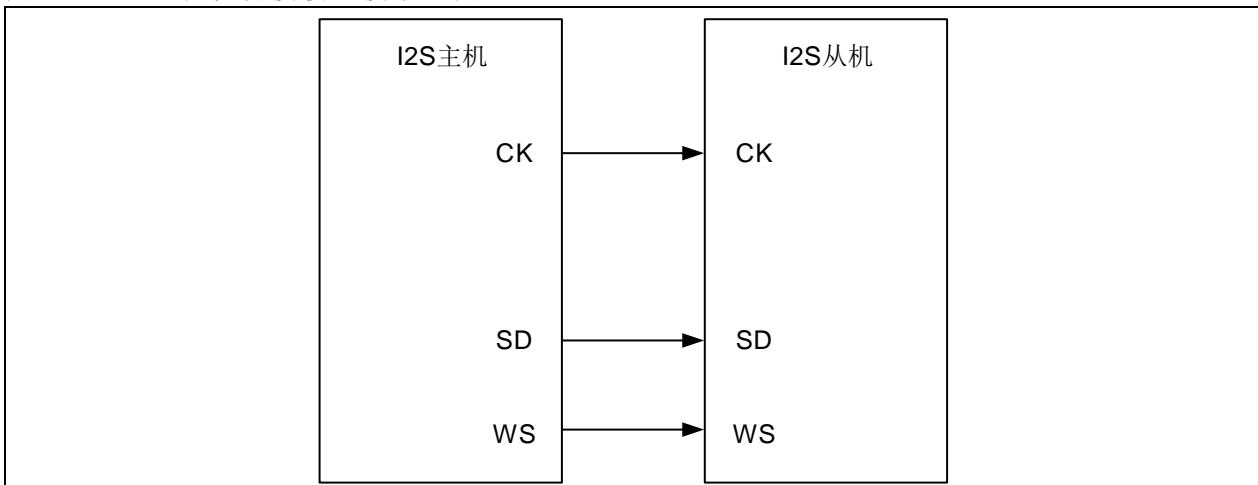
**从设备发送：**

置位 I2SMSEL 位，配置 OPERSEL[1: 0]位为 00，I<sup>2</sup>S 将工作在从设备发送模式下

图 13-14 I<sup>2</sup>S从设备发送连接示意图

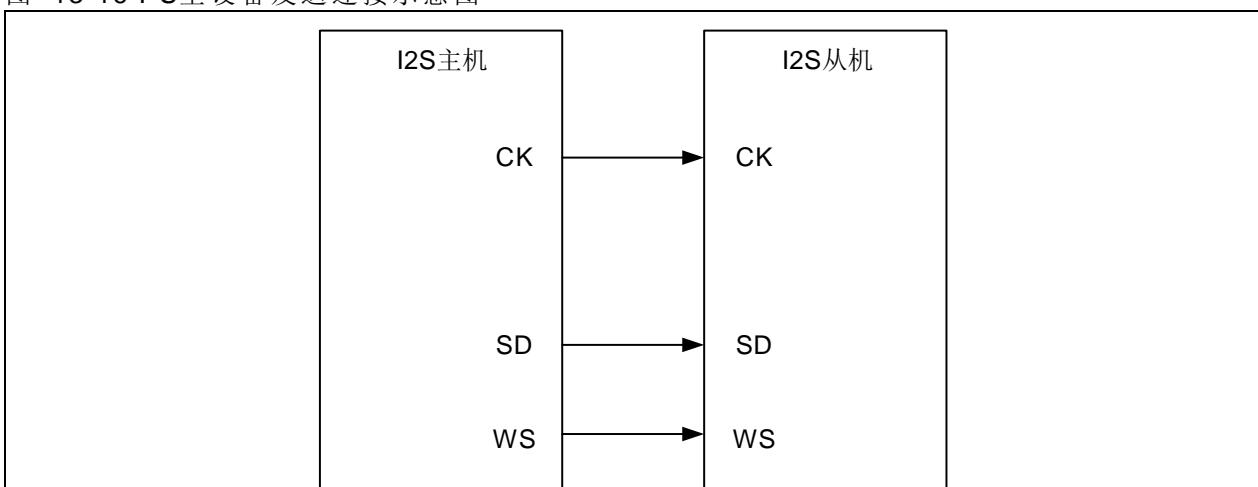
从设备接收:

置位 I2SMSEL 位, 配置 OPERSEL[1: 0]位为 01, I<sup>2</sup>S 将工作在从设备接收模式下

图 13-15 I<sup>2</sup>S从设备接收连接示意图

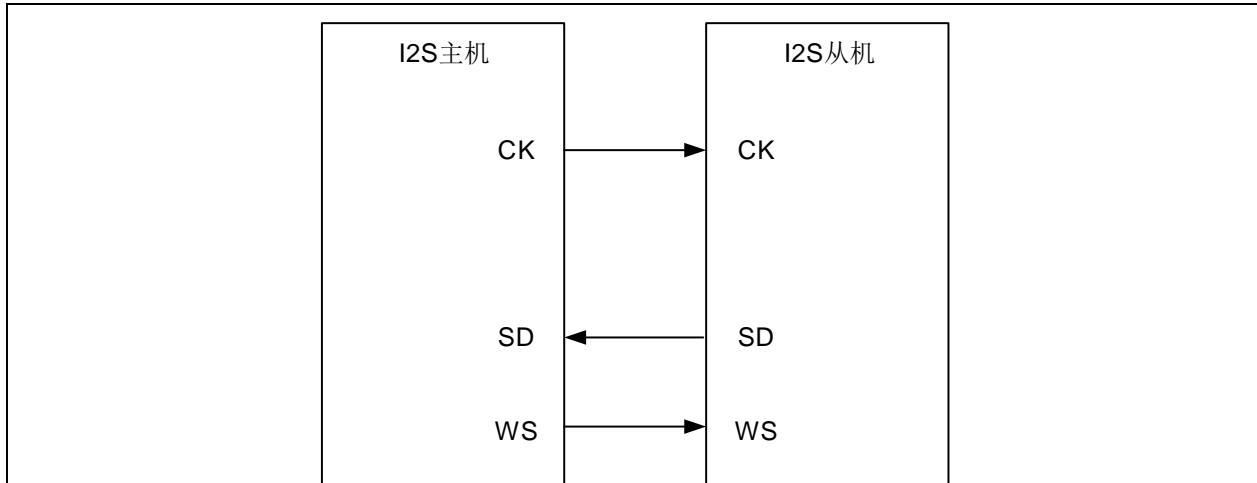
主设备发送:

置位 I2SMSEL 位, 配置 OPERSEL[1: 0]位为 10, I<sup>2</sup>S 将工作在主设备发送模式下

图 13-16 I<sup>2</sup>S主设备发送连接示意图

主设备接收:

置位 I2SMSEL 位, 配置 OPERSEL[1: 0]位为 11, I<sup>2</sup>S 将工作在主设备接收模式下

图 13-17 I<sup>2</sup>S主设备接收连接示意图

### 13.3.3 音频协议选择器简述和配置流程

SPI接口作为I<sup>2</sup>S使用时支持多种音频协议，用户可以通过软件编程控制音频协议选择器选择需要的音频协议，数据位个数以及声道位个数同样由音频协议选择器控制，用户同样可以通过软件编程配置的方式选择需要的数据位个数以及声道位个数，同时，音频协议选择器会自动控制WS控制器，输出或检测符合协议要求的WS信号，具体的配置流程如下。

- 音频协议选择：配置 STDSEL 位选择需要的音频协议

STDSLE=00：飞利浦标准

STDSLE=01：高字节对齐标准（左对齐）

STDSLE=10：低字节对齐标准（右对齐）

STDSLE=11：PCM 标准

- PCM 帧同步格式选择：配置 PCM 长帧同步（PCMFSSEL=1）或短帧同步（PCMFSSEL=0）（该步骤在选择 PCM 协议时需要）

- 数据位个数选择：配置 I2SDBN 位选择需要的数据位个数

I2SDBN=00：16 位

I2SDBN =01：24 位

I2SDBN =10：32 位

- 声道位个数选择：配置 I2SCBN 位选择需要的声道位个数

I2SCBN =0：16 位

I2SCBN =1：32 位

需要注意的是，不同的音频协议以及不同的数据位数和声道位数组合所对应的数据写入方式存在较大不同，下面将依次罗列所有的允许的配置组合以及其数据的读写方式。

- 飞利浦标准或 PCM 标准或高字节或低字节标准，16 位数据，16 位声道  
数据位数和声道位数一致，每个声道只需读写一次 SPI 数据寄存器（SPI\_DT），DMA 传输个数为 1。

- 飞利浦标准或 PCM 标准或高字节标准，16 位数据，32 位声道  
数据位数和声道位数不一致，每个声道只需读写一次 SPI 数据寄存器（SPI\_DT），DMA 传输个数为 1。只有前 16 位是有效数据，后 16 位数据硬件默认输出和接收 0。

- 飞利浦标准或 PCM 标准或高字节标准，24 位数据，32 位声道  
数据位数和声道位数不一致，每个声道需读写二次 SPI 数据寄存器（SPI\_DT），DMA 传输个数为 2。前 16 位发送和接收第一笔 16 位数据，后 16 位发送和接收高 8 位数据，低 8 位数据硬件默认输出和接收 0。

- 飞利浦标准或 PCM 标准或高字节或低字节标准，32 位数据，32 位声道  
数据位数和声道位数一致，每个声道需读写二次 SPI 数据寄存器（SPI\_DT），DMA 传输个数为 2。

数据分两次，依次发送和接收 16 位数据。

- 低字节标准，16位数据，32位声道

数据位数和声道位数不一致，每个声道只需读写一次 SPI 数据寄存器（SPI\_DT），DMA 传输个数为 1。只有后 16 位是有效数据，前 16 位数据硬件默认输出和接收 0。

- 低字节标准，24位数据，32位声道

数据位数和声道位数不一致，每个声道需读写二次 SPI 数据寄存器（SPI\_DT），DMA 传输个数为 2。

前 16 位数据只有低八位有效，高八位数据硬件默认输出和接收 0，后 16 位发送和接收第二笔 16 位数据

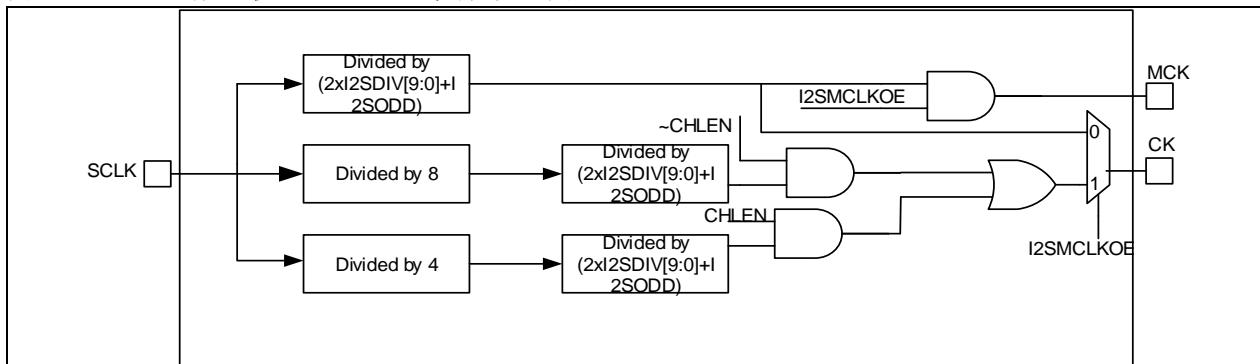
### 13.3.4 I2S\_CLK 控制器简述和配置流程

SPI 接口作 I<sup>2</sup>S 使用时，所有该接口支持的音频协议均为同步协议，作主机时，需要产生通信时钟用于 SPI 接口的数据收发，并且需要将该通信时钟通过 IO 输出给从机，用于从机的数据收发；作从机时，需要主机提供通信时钟从 IO 输入到 SPI 接口内部作为通信时钟使用，所以实际上，I2S\_CLK 控制器便是扮演着产生 I2S\_CLK 以及分配 I2S\_CLK 的角色。

SPI 接口作 I<sup>2</sup>S 主机时支持提供通信时钟 CK 以及外设主时钟 MCK，CK 和 MCK 的来源如图 13-18 所示，CK 和 MCK 都是由 HCLK 分频得到，其中 MCK 的分频系数由 I2SDIV 以及 I2SODD 决定，具体计算公式见图 13-18。

CK 的分频系数与是否给外设提供主时钟有关，为了满足主时钟始终是音频采样频率的 256 倍，取决于是否提供主时钟以及声道位个数，当需要给外设提供主时钟时，CK 需要先做 8（I2SCBN=0 时）或 4（I2SCBN=1 时）的预分频，随后再做和 MCK 相同分频系数的分频得到最终的通信时钟 CK；如果不需要给外设提供主时钟，则 CK 的分频系数只由 I2SDIV 以及 I2SODD 决定，具体计算公式见图 13-18。

图 13-18 SPI 作主机 CK & MCK 来源示意图



除了根据上面的描述自行配制想要的时钟外，我们也提供一些特定的时钟频率其对应的 I2SDIV, I2SODD 的值，以及相应的误差，用户可以直接按此表配置 I2SDIV 和 I2SODD。

表 13-1 使用系统时钟得到精确的音频频率

SCLK (MHz)	MCL K	Target Fs (Hz)	16bit				32bit			
			I2S DIV	I2S_ODD	RealFs	Error	I2S DIV	I2S_ODD	RealFs	Error
120	No	192000	10	0	187500	2.34%	10	0	187500	2.34%
120	No	96000	19	1	96153	0.16%	10	0	93750	2.34%
120	No	48000	39	0	48076	0.16%	19	1	48076	0.16%
120	No	44100	42	1	44117	0.04%	21	1	43604	1.12%
120	No	32000	58	1	32051	0.16%	29	1	31779	0.69%
120	No	22050	85	0	22058	0.04%	42	1	22058	0.04%
120	No	16000	117	0	16025	0.16%	58	1	16025	0.16%
120	No	11025	170	0	11029	0.04%	85	0	11029	0.04%
120	No	8000	234	1	7995	0.05%	117	1	8012	0.16%
120	Yes	96000	2	1	93750	2.34%	2	1	937500	2.34%
120	Yes	48000	5	0	46875	2.34%	5	0	46875	2.34%
120	Yes	44100	5	1	42613	3.37%	5	1	42613	3.37%
120	Yes	32000	7	1	31250	2.34%	7	1	31250	2.34%

120	Yes	22050	10	1	22321	1.23%	10	1	22321	1.23%
120	Yes	16000	14	1	16163	1.02%	14	1	16163	1.02%
120	Yes	11025	21	1	10901	1.023%	21	1	10901	1.023%
120	Yes	8000	29	1	7944	0.68%	29	1	7944	0.68%
108	No	192000	9	0	187500.0	2.34%	4	1	187500.0	2.34%
108	No	96000	17	1	96428.57	0.446%	9	0	93750	2.34%
108	No	48000	35	0	48214.29	0.446%	17	11	48214.29	0.446%
108	No	44100	38	1	43831.11	0.61%	19	00	44407.89	0.698%
108	No	32000	52	1	32142.86	0.446%	26	11	31839.62	0.501%
108	No	22050	76	1	22058.82	0.04%	38	10	21915.58	0.609%
108	No	16000	110	1	15995.26	0.029%	55	01	16071.43	0.446%
108	No	11025	153	0	11029.41	0.04%	76	11	11029.41	0.04%
108	No	8000	221	0	7997.63	0.029%	105	10	7997.63	0.029%
108	Yes	96000	2	0	105468.8	9.86%	2	0	105468.8	9.86%
108	Yes	48000	4	1	46875.0	2.34%	4	1	46875.0	2.34%
108	Yes	44100	5	0	42187.5	2.34%	5	0	42187.5	2.34%
108	Yes	32000	6	0	32451.92	1.41%	6	0	32451.92	1.41%
108	Yes	22050	9	1	22203.95	0.698%	9	1	22203.95	0.698%
108	Yes	16000	13	0	16225.96	1.41%	13	0	16225.96	1.41%
108	Yes	11025	19	0	11101.97	0.698%	17	1	11101.97	0.698%
108	Yes	8000	26	1	7959.906	0.501%	26	1	7959.906	0.501%
72	No	192000	6	0	187500	2.34%	3	0	187500	2.34%
72	No	96000	11	1	97826.09	1.90%	6	0	93750	2.34%
72	No	48000	32	1	34615.38	27.88%	11	1	48913.04	1.90%
72	No	44100	25	1	44117.65	0.04%	13	0	43269.23	1.88%
72	No	32000	35	0	32142.86	0.45%	17	1	32142.86	0.45%
72	No	22050	51	0	22058.82	0.04%	25	1	22058.82	0.04%
72	No	16000	70	1	15957.45	0.27%	35	0	16071.43	0.45%
72	No	11025	102	0	11029.41	0.04%	51	0	11029.41	0.04%
72	No	8000	140	1	8007.117	0.09%	70	1	7978.723	0.27%
72	Yes	96000	2	0	70312.5	26.76%	2	0	70312.5	26.76%
72	Yes	48000	3	0	46875	2.34%	3	0	46875	2.34%
72	Yes	44100	3	0	46875	6.29%	3	0	46875	6.29%
72	Yes	32000	4	1	31250	2.34%	4	1	31250	2.34%
72	Yes	22050	6	1	21634.62	1.88%	6	1	21634.62	1.88%
72	Yes	16000	9	0	15625	2.34%	9	0	15625	2.34%
72	Yes	11025	13	0	10817.31	1.88%	13	0	10817.31	1.88%
72	Yes	8000	17	1	8035.714	0.45%	17	1	8035.714	0.45%

### 13.3.5 DMA传输简述和配置流程

SPI 接口支持使用 DMA 进行发送数据的写入，接收数据的读取，由于无论 SPI 接口作 I<sup>2</sup>S 使用还是作 SPI 使用，对 DMA 来说，读写请求的来源都是同一个外设，所以实际上 SPI 接口作 I<sup>2</sup>S 使用时 DMA 传输的配置方法和作 SPI 使用并无不同，具体配置流程分别见下述的 DMA 发送配置流程以及 DMA 接收配置流程。

#### DMA发送配置流程：

- 选择 DMA 传输通道：在 DMA 章节 DMA 通道映射表中选择用于当前所用 SPI 的 DMA 通道。
- 配置 DMA 传输目标地址：在 DMA 控制寄存器中 DMA 传输目的地址位写入当前所使用的 SPI 的 SPI 数据寄存器（SPI\_DT）地址，DMA 将会在接收到发送请求后将待发送的数据写入该地址。

- 配置 DMA 传输源地址：在 DMA 控制寄存器中 DMA 传输源地址位写入待发送数据存放的地址，DMA 将会在接收到发送请求后将该地址内的数据写入到目标地址中，即写入到当前所使用的 SPI 的 SPI 数据寄存器（SPI\_DT）中。
- 配置 DMA 传输数据个数：在 DMA 控制寄存器相关位置配置期望传输的数据个数
- 配置 DMA 传输通道优先级：在 DMA 控制寄存器相关位置配置当前所使用通道的 SPI 的 DMA 传输通道优先级。
- 配置 DMA 中断产生时机：在 DMA 控制寄存器相关位置配置是在传输完成或传输完成一半时产生 DMA 中断。
- 使能 DMA 传输通道：在 DMA 控制寄存器相关位置使能当前所选用的 DMA 通道

#### DMA接收配置流程：

- 选择 DMA 传输通道：在 DMA 章节 DMA 通道映射表中选择用于当前所用 SPI 的 DMA 通道。
- 配置 DMA 传输目标地址：在 DMA 控制寄存器中 DMA 传输目的地址位写入期望存放接收数据的地址，DMA 将会在接收到接收请求后，将当前所使用的 SPI 的 SPI\_DT 寄存器中的数据存放在目的地址中。
- 配置 DMA 传输源地址：在 DMA 控制寄存器中 DMA 传输源地址位写入当前所使用的 SPI 的 SPI 数据寄存器（SPI\_DT）的地址，DMA 将会在接收到接收请求后将该地址内的数据写入到目标地址中，即写入到期望存放接收数据的地址。
- 配置 DMA 传输数据个数：在 DMA 控制寄存器相关位置配置期望传输的数据个数。
- 配置 DMA 传输通道优先级：在 DMA 控制寄存器相关位置配置当前所使用通道的 SPI 的 DMA 传输通道优先级。
- 配置 DMA 中断产生时机：在 DMA 控制寄存器相关位置配置是在传输完成或传输完成一半时产生 DMA 中断。
- 使能 DMA 传输通道：在 DMA 控制寄存器相关位置使能当前所选用的 DMA 通道

### 13.3.6 发送器接收器简述和配置流程

由于无论 SPI 接口作 I<sup>2</sup>S 使用还是作 SPI 使用，对于 CPU 来说都是同一个外设，共用同一个基地址，并且 SPI 接口内部，作 I<sup>2</sup>S 使用和作 SPI 使用时，都共用同一个数据寄存器 SPI\_DT，并且实际上发送器和接收器也是共用的，所以 SPI 接口的发送器和接收器只是根据通信控制器的配置发送和接收期望的数据帧格式，所以如 TDBE 和 RDBF 以及 ROERR 等状态标志，以及 TDBEIE 和 RDBFIE 以及 ERRIE 等中断使能位都是共用的。

但需要特别注意的是：

- I<sup>2</sup>S 不支持 CRC 校验，所以和 CRC 有关的操作，以及 CCERR 标志和与之相对应的中断都不能使用。
- I<sup>2</sup>S 协议需要解析当前的声道状态，用户可以根据 ACS 位判断当前传输是左声道（ACS=0）还是右声道（ACS=1）。
- I<sup>2</sup>S 使用 TUERR 位表示当前是否发生欠载，TUERR=1，表示当前发送器出现了欠载错误，如果 ERRIE 置位，则产生中断。
- I<sup>2</sup>S 在不同的音频协议和数据位数以及声道位数的组合下，操作 SPI\_DT 寄存器的方式是不同的，具体可以参考音频协议选择器简述和配置流程部分描述。
- I<sup>2</sup>S 的关闭方式同样需要特别注意，依据不同的配置方式罗列如下
  - I2SDBN=00, I2SCBN=1, STDSLE=10: 等待倒数第二个RDBF=1, 等待17个CK周期, 关闭I<sup>2</sup>S。
  - I2SDBN=00, I2SCBN=1, STDSLE=00或STDSLE=01或STDSLE=11: 等待最后一个RDBF=1, 等待一个CK时钟周期, 关闭I<sup>2</sup>S。
  - 其它I2SDBN, I2SCBN,STDSLE组合: 等待倒数第二个RDBF=1, 等待一个CK时钟周期, 关闭I<sup>2</sup>S。

下面给出发送器和接收器的配置流程

#### I<sup>2</sup>S 发送器配置流程：

- 配置操作模式选择器
- 配置音频协议选择器
- 配置 I<sup>2</sup>S\_CLK 控制器

- 配置 DMA(若需要开启 DMA 传输)
- 置位 I2SEN 位开启 I<sup>2</sup>S

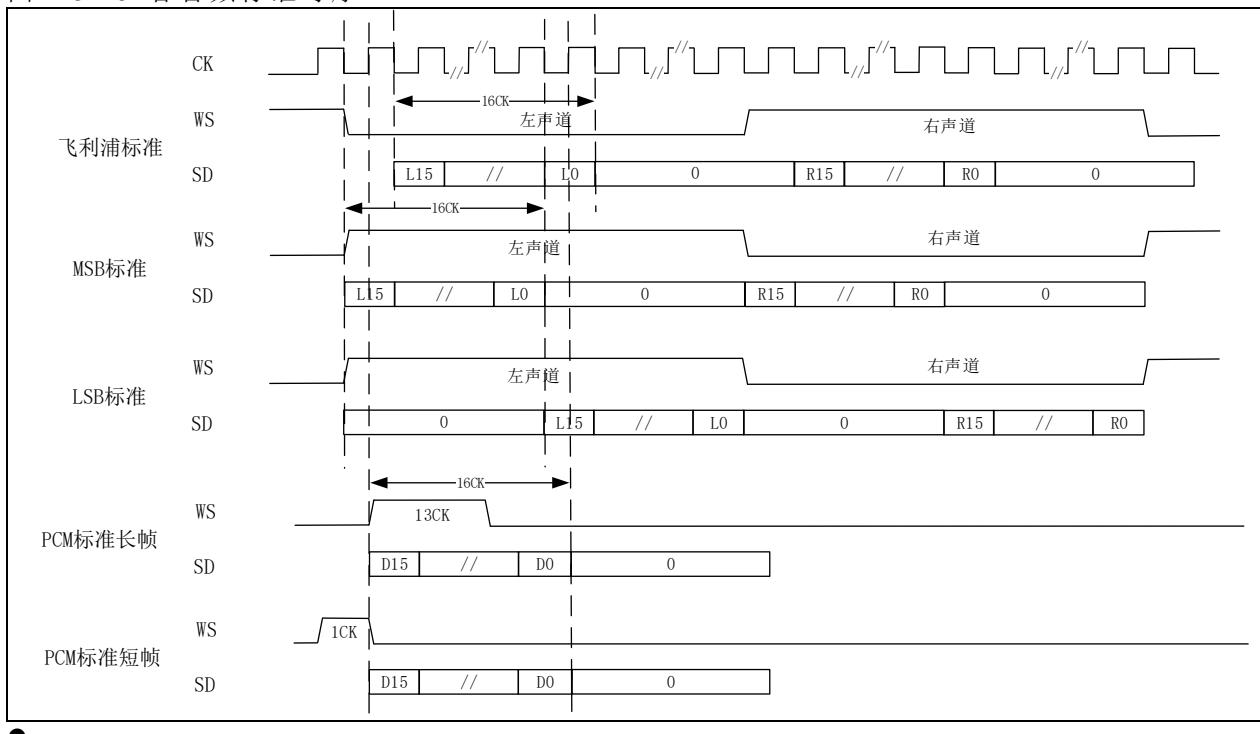
#### I<sup>2</sup>S 接收器配置流程:

- 配置操作模式选择器
- 配置音频协议选择器
- 配置 I2S\_CLK 控制器
- 配置 DMA(若需要开启 DMA 传输)
- 置位 I2SEN 位开启 I<sup>2</sup>S

### 13.3.7 I<sup>2</sup>S通信时序

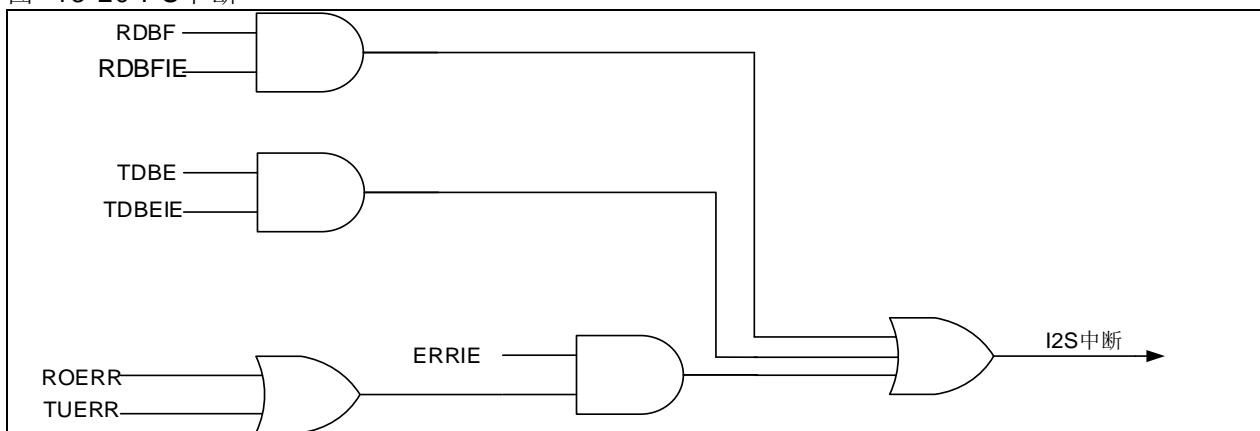
I<sup>2</sup>S 支持以下 4 种音频协议：飞利浦标准，高字节对齐标准（左对齐），低字节对齐标准（右对齐），PCM 标准，各标准音频时序如下。

图 13-19 各音频标准时序



### 13.3.8 中断

图 13-20 I<sup>2</sup>S 中断



### 13.3.9 IO管脚控制

SPI 接口作 I<sup>2</sup>S 使用时，I<sup>2</sup>S 传输需要三个管脚，分别是数据管脚 SD，同步管脚 WS，通信时钟管脚 CK，如果需要给外设提供主时钟还需要主时钟输出管脚 MCLK，由于一个 SPI 接口不可能同时作 I<sup>2</sup>S 和 SPI 使用，所以 I<sup>2</sup>S 和 SPI 部分管脚映射是共用的各管脚的映射和定义如下。

- SD: 数据管脚（和 MOSI 管脚共用同样的 GPIO 映射关系），数据的双向收发管脚。
- WS: 同步管脚（和 CS 管脚共用同样的 GPIO 映射关系），通信同步信号的双向控制管脚，主模式输出，从模式输入。
- CK: 通信时钟管脚（和 SCK 管脚共用同样的 GPIO 映射关系），通信时钟双向输入输出管脚，主模式输出，从模式输入。
- MCLK: 主时钟管脚（独立映射），主时钟输出管脚，用于给外设提供主时钟，输出的时钟频率固定为音频采样频率的 256 倍，部分 MCLK 管脚和 MISO 管脚有相同的 GPIO 映射。

## 13.4 SPI寄存器

必须以字（32位）的方式操作这些外设寄存器。

表 13-2 SPI寄存器映像及其复位值

寄存器简称	基址偏移量	复位值
SPI_CTRL1	0x00	0x0000
SPI_CTRL2	0x04	0x0000
SPI_STS	0x08	0x0002
SPI_DT	0x0C	0x0000
SPI_CPOLY	0x10	0x0007
SPI_RCRC	0x14	0x0000
SPI_TCRC	0x18	0x0000
SPI_I2SCTRL	0x1C	0x0000
SPI_I2SCLKP	0x20	0x0002

### 13.4.1 SPI控制寄存器1 (SPI\_CTRL1) (I<sup>2</sup>S模式下不使用)

域	简称	复位值	类型	功能
位 15	SLBEN	0x0	rw	单线双向半双工模式使能 (Single line bidirectional half-duplex enable) 0: 关闭; 1: 开启。
位 14	SLBTD	0x0	rw	单线双向半双工模式传输方向 (Single line bidirectional half-duplex transmission direction) 和 SLBEN 位一起决定在“单线双向半双工”模式下数据的传输方向 0: 接收模式; 1: 发送模式。
位 13	CCEN	0x0	rw	CRC 校验使能 (CRC calculation enable) 0: 关闭; 1: 开启。
位 12	NTC	0x0	rw	下一笔传输数据为 CRC (Next transmission CRC) 该位置起表示下一笔传输的数据为 CRC 数据。 0: 普通数据; 1: CRC 数据。
位 11	FBN	0x0	rw	帧位个数 (frame bit num) 该位配置发送/接收时数据帧位个数。 0: 8 位; 1: 16 位。
位 10	ORA	0x0	rw	仅接收有效 (Only receive active) 在“双线单向”模式时，该位置起表示只有接收有效，发送被禁止。 0: 发送和接收; 1: 仅接收。
位 9	SWCSEN	0x0	rw	软件 CS 模式使能 (Software CS enable)

				当该位被置起时，CS 管脚上的电平由 SWCSIL 位的值决定，此时在 CS 管脚上的 I/O 电平状态无效。 0：关闭； 1：开启。
位 8	SWCSIL	0x0	rw	软件 CS 内部电平（Software CS internal level） 该位只在 SWCSEN 位置起时有意义，它决定了 CS 上的内部电平状态。 做主设备时，该位必须设置置起。 0：低电平； 1：高电平。
位 7	LTF	0x0	rw	LSB 先传输（LSB transmit first） 该位用于选择数据先传输 MSB 还是 LSB。 0：MSB； 1：LSB。
位 6	SPIEN	0x0	rw	SPI 使能（SPI enable） 0：关闭； 1：开启。
位 5：3	MDIV	0x0	rw	主模式时钟频率分频系数（Master clock frequency division） 作主模式时，分频系数对外设时钟进行分频，作为 SPI 时钟，MDIV[3: 0]位在 SPI 控制寄存器 2（SPI_CTRL2），MDIV[3: 0]: 0000：2 分频 0001：4 分频 0010：8 分频 0011：16 分频 0100：32 分频 0101：64 分频 0110：128 分频 0111：256 分频 1000：512 分频 1001：1024 分频
位 2	MSTEN	0x0	rw	主模式使能（Master enable） 0：关闭（从设备）； 1：开启（主设备）。
位 1	CLKPOL	0x0	rw	时钟极性（Clock polarity） 空闲时时钟输出的极性。 0：低电平； 1：高电平。
位 0	CLKPHA	0x0	rw	时钟相位（Clock phase） 0：第一个边沿进行数据捕获； 1：第二个边沿进行数据捕获。

注：在 I<sup>2</sup>S 模式下，SPI 控制寄存器 1（SPI\_CTRL1）需置 0。

### 13.4.2 SPI控制寄存器2（SPI\_CTRL2）

域	简称	复位值	类型	功能
位 15：9	保留位	0x00	resd	硬件强制为 0
位 8	MDIV	0x0	rw	主模式时钟频率分频系数（Master clock frequency division） 详见 MDIV[2: 0]在 SPI 控制寄存器 1（SPI_CTRL1）。
位 7	TDBEIE	0x0	rw	发送数据缓冲器空中断使能（Transmit data buffer empty interrupt enable） 0：关闭； 1：开启。
位 6	RDBFIE	0x0	rw	接收数据缓冲器满中断使能（Receive data buffer full interrupt enable） 0：关闭； 1：开启。
位 5	ERRIE	0x0	rw	错误中断使能（Error interrupt enable）

				当错误 (CCERR、MMERR、ROERR、TUERR) 产生时, 该位控制是否产生中断 0: 关闭; 1: 开启。
位 4: 3	保留位	0x0	resd	保持默认值。
位 2	HWCSOE	0x0	rw	硬件 CS 输出使能 (Hardware CS output enable) 该位做主设备时才有意义, 设置为'1'时, CS 脚 I/O 口输出低电平, 设置为'0'时, 必须保证 CS 脚 I/O 口输入为高电平。 0: 关闭; 1: 开启。
位 1	DMATEN	0x0	rw	DMA 发送使能 (DMA transmit enable) 0: 关闭; 1: 开启。
位 0	DMAREN	0x0	rw	DMA 接收使能 (DMA receive enable) 0: 关闭; 1: 开启。

### 13.4.3 SPI状态寄存器 (SPI\_STS)

域	简称	复位值	类型	功能
位 15: 8	保留位	0x00	resd	硬件强制为 0
位 7	BF	0x0	ro	通信忙标志 (Busy flag) 0: 通信空闲; 1: 通信忙。
位 6	ROERR	0x0	ro	接收器溢出错误 (Receiver overflow error) 0: 无; 1: 有。
位 5	MMERR	0x0	ro	主模式错误 (Master mode error) 该位由硬件置位, 软件清除 (先读或写 SPI 状态寄存器 (SPI_STS), 再写 SPI 控制寄存器 1 (SPI_CTRL1))。 0: 无; 1: 有。
位 4	CCERR	0x0	rw0c	CRC 校验错误 (CRC calculation error) 该位由硬件置起, 由软件清除。 0: 正确; 1: 错误。
位 3	TUERR	0x0	ro	发送器欠载错误 (Transmitter underload error) 该位由硬件置起, 软件清除 (读 SPI 状态寄存器 (SPI_STS))。 0: 无; 1: 有。 注: 该位只在 I <sup>2</sup> S 模式使用。
位 2	ACS	0x0	ro	音频通道状态 (Audio channel state) 该位表示当前传输的音频左右声道状态。 0: 左声道; 1: 右声道。 注: 该位只在 I <sup>2</sup> S 模式使用。
位 1	TDBE	0x1	ro	发送数据缓冲器空 (Transmit data buffer empty) 0: 非空; 1: 空。
位 0	RDBF	0x0	ro	接收数据缓冲器满 (Receive data buffer full) 0: 未满; 1: 满。

### 13.4.4 SPI数据寄存器 (SPI\_DT)

域	简称	复位值	类型	功能
位 15: 0	DT	0x0000	rw	数据值 (Data value) 该寄存器包含读和写的功能, 当数据位配置为 8 位时, 该寄存器只有低 8 位[7: 0]有效。

### 13.4.5 SPICRC多项式寄存器 (SPI\_CPOLY) (I<sup>2</sup>S模式下不使用)

域	简称	复位值	类型	功能
位 15: 0	CPOLY	0x0007	rw	CRC 多项式寄存器 (CRC polynomial) 该寄存器为 CRC 计算时用到的多项式, 可以根据应用设置。 注: 该寄存器只在 SPI 模式下使用。

### 13.4.6 SPIRxCRC寄存器 (SPI\_RCRC) (I<sup>2</sup>S模式下不使用)

域	简称	复位值	类型	功能
位 15: 0	RCRC	0x0000	ro	接收 CRC 寄存器 (receive CRC) CRC 使能后, 该寄存器值为根据接收到的数据计算得到的 CRC 值, 要复位该寄存器, 需操作 SPI 控制寄存器 1 (SPI_CTRL1) 的 CCEN 位先清除再置起。 当数据位配置为 8 位时, 该寄存器只有低 8 位[7: 0]有效, 按照 CRC8 计算; 当数据位配置为 16 位时, 按照 CRC16 计算。 注: 该寄存器只在 SPI 模式下使用。

### 13.4.7 SPITxCRC寄存器 (SPI\_TCRC)

域	简称	复位值	类型	功能
位 15: 0	TCRC	0x0000	ro	发送 CRC 寄存器 (transmit CRC) CRC 使能后, 该寄存器值为根据发送的数据计算得到的 CRC 值。要复位该寄存器, 需操作 SPI 控制寄存器 1 (SPI_CTRL1) 的 CCEN 位先清除再置起。 当数据位配置为 8 位时, 该寄存器只有低 8 位[7: 0]有效, 按照 CRC8 计算; 当数据位配置为 16 位时, 按照 CRC16 计算。 注: 该寄存器只在 SPI 模式下使用。

### 13.4.8 SPI\_I2S配置寄存器 (SPI\_I2SCTRL)

域	简称	复位值	类型	功能
位 15: 12	保留位	0x0	resd	硬件强制为 0
位 11	I2SMSEL	0x0	rw	I <sup>2</sup> S 模式选择 (I <sup>2</sup> S mode select) 0: SPI 模式; 1: I <sup>2</sup> S 模式。
位 10	I2SEN	0x0	rw	I <sup>2</sup> S 使能 (I <sup>2</sup> S enable) 0: 关闭; 1: 开启。
位 9: 8	OPERSEL	0x0	rw	I <sup>2</sup> S 操作选择 (I <sup>2</sup> S operation select) 00: 从设备发送; 01: 从设备接收; 10: 主设备发送; 11: 主设备接收。
位 7	PCMFSSEL	0x0	rw	PCM 帧同步 (PCM frame synchronization select) 该位只在使用 PCM 标准时才有意义。 0: 短帧同步; 1: 长帧同步。
位 6	保留位	0x0	resd	保持默认值。
位 5: 4	STDSEL	0x0	rw	I <sup>2</sup> S 标准选择 (I <sup>2</sup> S standard select)

				00: 飞利浦标准; 01: 高字节对齐标准（左对齐）; 10: 低字节对齐标准（右对齐）; 11: PCM 标准。
位 3	I2SCLKPOL	0x0	rw	I <sup>2</sup> S 时钟极性 (I <sup>2</sup> S clock polarity) 时钟管脚上总线空闲时时钟输出的极性。 0: 低电平; 1: 高电平。
位 2: 1	I2SDBN	0x0	rw	I <sup>2</sup> S 数据位个数 (I <sup>2</sup> S data bit num) 00: 16 位; 01: 24 位; 10: 32 位; 11: 不允许。
位 0	I2SCBN	0x0	rw	I <sup>2</sup> S 声道位个数 (I <sup>2</sup> S channel bit num) 该位只有在 I <sup>2</sup> S 数据位个数为 16 位时配置才有意义，否则都由硬件固定为 32 位。 0: 16 位宽; 1: 32 位宽。

### 13.4.9 SPI\_I2S预分频寄存器 (SPI\_I2SCLKP)

域	简称	复位值	类型	功能
位 15: 12	保留位	0x0	resd	硬件强制为 0
位 9	I2SMCLKOE	0x0	rw	I <sup>2</sup> S 主设备时钟输出使能 (I <sup>2</sup> S Master clock output enable) 0: 关闭; 1: 开启。
位 8	I2SODD	0x0	rw	I <sup>2</sup> S 分频系数配置奇数 (Odd result for I <sup>2</sup> S division) 0: 实际分频系数=I2SDIV*2; 1: 实际分频系数=(I2SDIV*2)+1。
位 11: 10 位 7: 0	I2SDIV	0x02	rw	I <sup>2</sup> S 分频系数 (I <sup>2</sup> S division) I2SDIV[9: 0]禁止设置为 0 或者 1。

## 14 定时器 (TIMER)

AT32F421 定时器种类有基本定时器、通用定时器、高级控制定时器，详细功能模式可参考 [14.1~14.6](#) 节说明，下表为各种类型定时器的功能总表。

表 14-1 TMR功能对比

Timer	类型	Timer	计数位数	计数方式	重复计数器	预分频系数	DMA 请求产生	捕获/比较通道	PWM 输入模式	EXT 输入	刹车输入
高级控制定时器	TMR1	16	向上 向下 向上/向下	8 位	1~65536	支持	4	支持	支持	支持	支持
	TMR3	16	向上 向下 向上/向下	不支持	1~65536	支持	4	支持	支持	不支持	
	TMR14	16	向上	不支持	1~65536	不支持	1	不支持	不支持	不支持	
	TMR15	16	向上	支持	1~65536	支持	2	支持	不支持	支持	
	TMR16 TMR17	16	向上	支持	1~65536	支持	1	不支持	不支持	支持	
基本定时器	TMR6	16	向上	不支持	1~65536	支持	不支持	不支持	不支持	不支持	不支持
Timer	类型	Timer	计数位数	计数方式	PWM 输出	单周期输出	互补输出	死区	编码器接口连接	霍尔传感器接口连接	联动外设
高级控制定时器	TMR1	16	向上 向下 向上/向下	支持	支持	支持	支持	支持	支持	支持	定时器同步/ADC
	TMR3	16	向上 向下 向上/向下	支持	支持	不支持	不支持	支持	支持	支持	定时器同步/ADC
	TMR14	16	向上	支持	支持	不支持	不支持	不支持	不支持	不支持	无
	TMR15	16	向上	支持	支持	支持	支持	不支持	不支持	不支持	定时器同步/ADC
	TMR16 TMR17	16	向上	支持	支持	支持	支持	不支持	不支持	不支持	无
基本定时器	TMR6	16	向上	不支持	不支持	不支持	不支持	不支持	不支持	不支持	无

### 14.1 基本定时器 (TMR6)

#### 14.1.1 TMR6简介

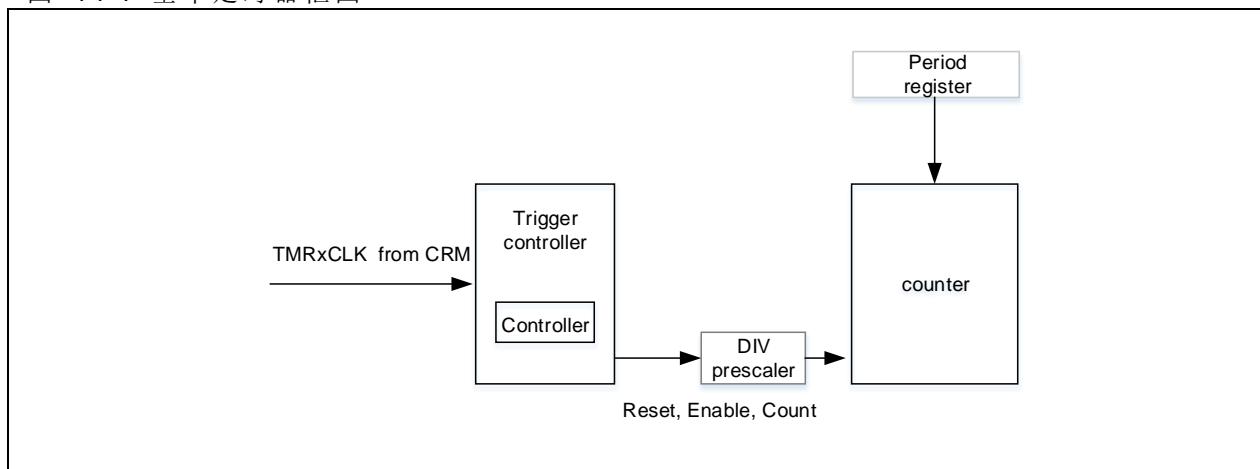
基本定时器 (TMR6) 包含一个 16 位向上计数器以及对应的控制逻辑，没有外部 I/O 接入。

#### 14.1.2 TMR6主要特性

- 由内部时钟用作计数时钟
- 16 位向上计数器

- 支持溢出事件中断和 DMA 请求

图 14-1 基本定时器框图

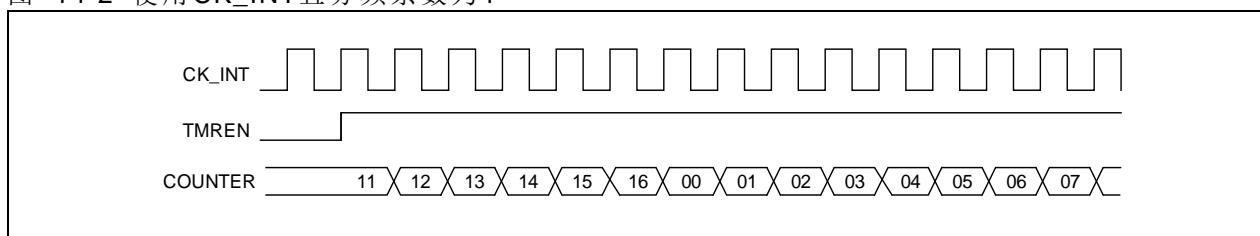


## 14.1.3 TMR6功能

### 14.1.3.1 计数时钟

TMR6 由内部时钟源 (CK\_INT) 经由预分频器提供计数器计数。当 TMR 对应的 APB 时钟预分频系数是 1 时，CK\_INT 频率等于 APB 时钟频率，否则 CK\_INT 频率等于 APB 时钟频率的 2 倍。

图 14-2 使用 CK\_INT 且分频系数为 1



### 14.1.3.2 计数模式

基本定时器仅提供向上计数模式。其内部拥有一个 16 位计数器。

TMRx\_PR 寄存器用于设置计数器计数周期。默认 TMRx\_PR 寄存器值会立即传入它的影子寄存器；当开启周期缓冲功能后 (PRBEN 置 1)，TMRx\_PR 寄存器值在溢出事件发生时传入它的影子寄存器。

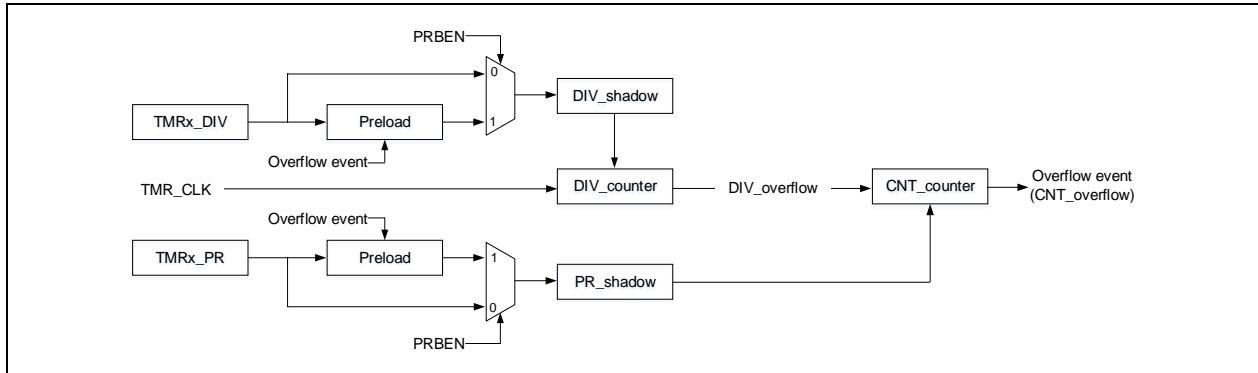
TMRx\_DIV 寄存器用于设置计数器计数频率，每 (DIV[15:0]+1) 个计数时钟周期，计数器计数一次。和 TMRx\_PR 寄存器类似，开启周期缓冲功能后，TMRx\_DIV 寄存器值在溢出事件时更新至它的影子寄存器。

读取 TMRx\_CNT 寄存器会返回当前计数器计数值，写入 TMRx\_CNT 寄存器会更新计数器当前计数值为写入值。

默认允许产生溢出事件，设置 TMRx\_CTRL1 寄存器 OVREN=1 将禁止更新事件产生。TMRx\_CTRL1 寄存器 OVFS 用于选择溢出事件来源，默认计数器上溢或下溢、置位 OVFSWTR、复位模式次定时器控制器产生的复位信号产生溢出事件。置位 OVFS 后，只有计数器上溢或下溢产生溢出事件。

TMREN 位置 1 将使能定时器计数，由于同步逻辑，实际驱动计数器的使能信号 TMR\_EN 相对于 TMREN 延迟一个时钟周期。

图 14-3 计数器基本结构



### 向上计数模式

上计数模式计数值达到  $\text{TMRx\_PR}$  值时，重新从 0 向上计数，计数器上溢并产生溢出事件，同时 OVFIF 位置 1。若禁止产生溢出事件，计数器溢出后不再重载预分频值和周期值，否则预分频值和周期值在溢出事件后更新。

图 14-4 PRBEN=0 时的溢出事件

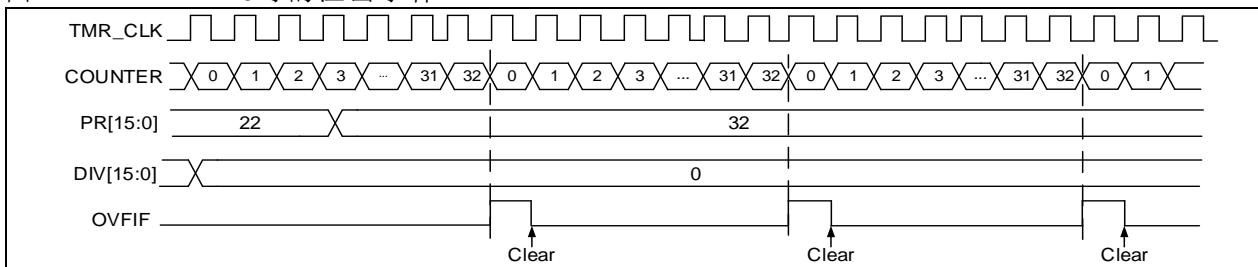


图 14-5 PRBEN=1 时的溢出事件

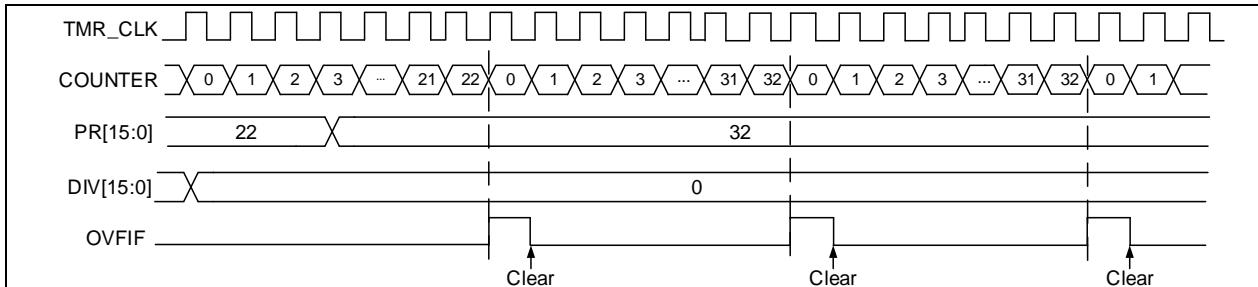
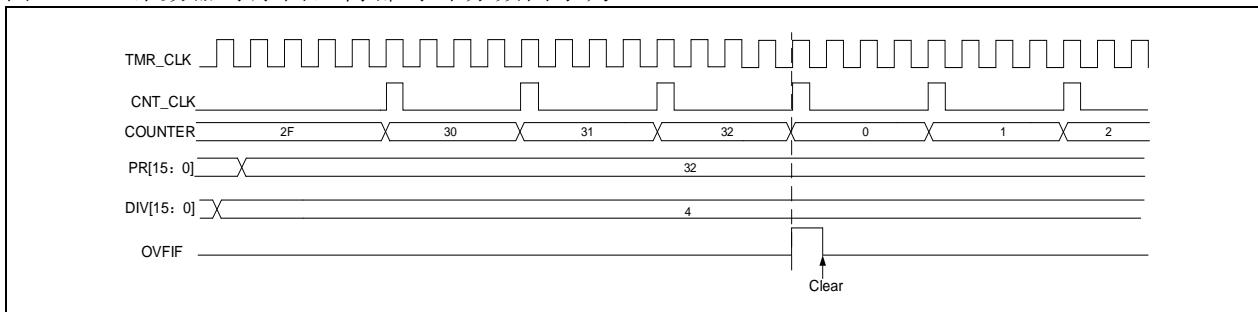


图 14-6 计数器时序图，内部时钟分频因子为 4



### 14.1.3.3 调试模式

当微控制器进入调试模式（Cortex®-M4 核心停止）时，将 DEBUG 模块中的  $\text{TMRx\_PAUSE}$  置 1，可以使  $\text{TMRx}$  计数器暂停计数。

### 14.1.4 TMR6 寄存器

必须以字（32 位）的方式操作这些外设寄存器。

下表中将 TMR6 的所有寄存器映射到一个 16 位可寻址（编址）空间。

表 14-2 TMR6寄存器映像和复位值

寄存器简称	基址偏移量	复位值
TMRx_CTRL1	0x00	0x0000
TMRx_CTRL2	0x04	0x0000
TMRx_IDEN	0x0C	0x0000
TMRxISTS	0x10	0x0000
TMRx_SWEVT	0x14	0x0000
TMRx_CVAL	0x24	0x0000
TMRx_DIV	0x28	0x0000
TMRx_PR	0x2C	0x0000

#### 14.1.4.1 TMR6控制寄存器1 (TMRx\_CTRL1)

域	简称	复位值	类型	功能
位 15: 8	保留	0x00	resd	保持默认值。
位 7	PRBEN	0x0	rw	周期缓冲使能 (Period buffer enable) 0: 缓冲关闭; 1: 缓冲开启。
位 6: 4	保留	0x0	resd	保持默认值。
位 3	OCMEN	0x0	rw	单周期使能 (One cycle mode enable) 该功能用于选择溢出事件后, 计数器是否停止。 0: 关闭; 1: 开启。
位 2	OVFS	0x0	rw	溢出事件源选择 (Overflow event source) 配置溢出事件或 DMA 请求来源。 0: 来源于计数器溢出、设置 OVFSWTR 位或次定时器控制器产生的溢出事件; 1: 只能来源于计数器溢出。
位 1	OVFEN	0x0	rw	溢出事件使能 (Overflow event enable) 该位用于允许或禁止溢出事件 (OEV) 产生。 0: 允许溢出事件产生, 溢出事件可以由下列事件产生: - 计数器溢出 - 将 OVFSWTR 位置 1 - 通过次定时器控制器产生的溢出事件 1: 禁止溢出事件产生。 如果将 OVFSWTR 位置 1 或次定时器控制器产生了一个硬件复位, 则计数器和预分频器将被重新初始化。 注: 该位由软件置 1 和清 0。
位 0	TMREN	0x0	rw	使能定时器 (TMR enable) 0: 关闭; 1: 开启。

#### 14.1.4.2 TMR6控制寄存器2 (TMRx\_CTRL2)

域	简称	复位值	类型	功能
位 15: 7	保留	0x000	resd	保持默认值。
位 6: 4	PTOS	0x0	rw	主定时器输出信号选择 (Primary TMR output selection) TMRx 输出到次定时器的信号选择: 000: 复位; 001: 使能; 010: 更新;
位 3: 0	保留	0x0	resd	保持默认值。

#### 14.1.4.3 TMR6 DMA/中断使能寄存器 (TMRx\_IDEN)

域	简称	复位值	类型	功能
位 15: 9	保留	0x00	resd	保持默认值。

位 8	OVFDEN	0x0	rw	溢出事件的 DMA 请求使能 (overflow event DMA request enable) 0: 关闭; 1: 开启。
位 7: 1	保留	0x00	resd	保持默认值。
位 0	OVFLEN	0x0	rw	溢出中断使能 (overflow interrupt enable) 0: 关闭; 1: 开启。

#### 14.1.4.4 TMR6中断状态寄存器 (TMRxISTS)

域	简称	复位值	类型	功能
位 15: 1	保留	0x0000	resd	保持默认值。
位 0	OVFIF	0x0	rw0c	溢出中断标记 (Overflow interrupt flag) 当溢出事件发生时由硬件置'1'，由软件清'0'。 0: 无溢出事件发生; 1: 发生溢出事件，若 TMRx_CTRL1 的 OVFEN=0、 OVFS=0 时： - 当 TMRx_SWEVE 寄存器的 OVFG=1 时产生溢出事件； - 当计数值 CVAL 被触发事件重初始化时产生溢出事件。

#### 14.1.4.5 TMR6软件事件寄存器 (TMRx\_SWEVT)

域	简称	复位值	类型	功能
位 15: 1	保留	0x0000	resd	保持默认值。
位 0	OVFSWTR	0x0	rw0c	软件触发溢出事件 (Overflow event triggered by software) 通过软件触发一个溢出事件。 0: 无作用; 1: 制造一个溢出事件。

#### 14.1.4.6 TMR6计数值 (TMRxCVAL)

域	简称	复位值	类型	功能
位 15: 0	CVAL	0x0000	rw	计数值 (Counter value)

#### 14.1.4.7 TMR6分频系数 (TMRxDIV)

域	简称	复位值	类型	功能
位 15: 0	DIV	0x0000	rw	分频系数 (Divider value) 计数器时钟频率 $f_{CK\_CNT} = f_{TMR\_CLK} / (DIV[15: 0]+1)$ 。 DIV 为溢出事件发生时写入的分频系数。

#### 14.1.4.8 TMR6周期寄存器 (TMRxPR)

域	简称	复位值	类型	功能
位 15: 0	PR	0x0000	rw	周期值 (Period value) 定时器计数的周期值。当周期值为 0 时，定时器不工作。

## 14.2 通用定时器 (TMR3)

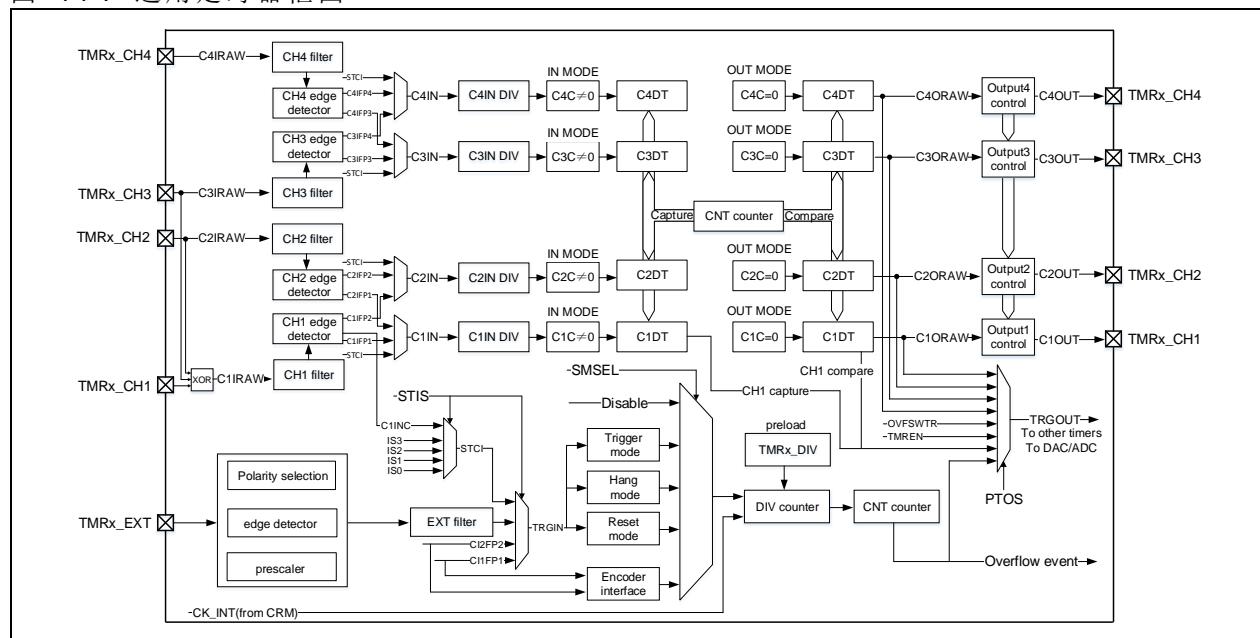
### 14.2.1 TMR3简介

通用定时器 TMR3 包含一个支持向上、向下、中央双向对齐计数的 16 位计数器、4 个捕获/比较寄存器、4 组独立的通道。可实现输入捕获、可编程 PWM 输出。

### 14.2.2 TMR3主要功能

- 可选内部、外部、内部触发输入用作计数时钟
- 16 位支持向上、向下、双向、编码器模式的计数器
- 4 组独立通道，支持输入捕获、输出比较、PWM 生成、单周期模式
- 定时器之间可互联同步
- 支持溢出事件、触发事件、通道事件触发中断/DMA
- 支持 TMR burst DMA 传输

图 14-7 通用定时器框图

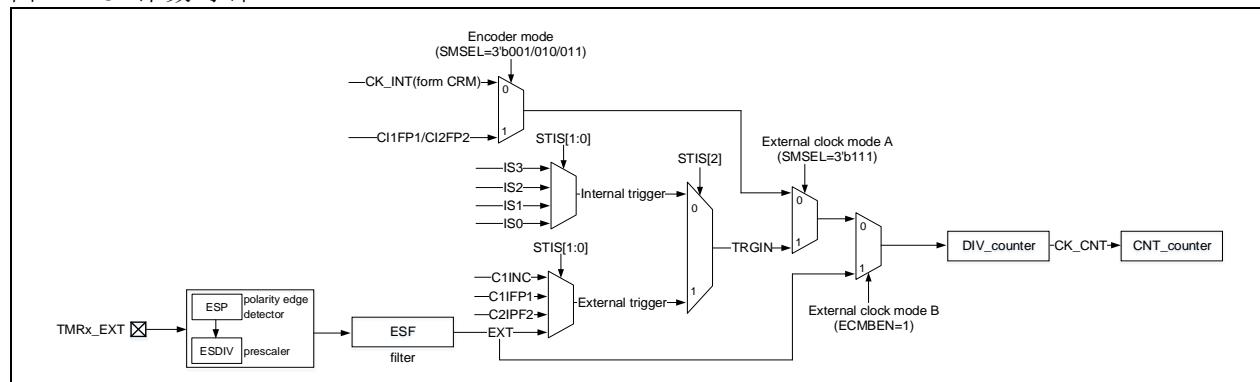


### 14.2.3 TMR3功能描述

#### 14.2.3.1 计数时钟

TMR3 计数时钟可从内部时钟 (CK\_INT)、外部时钟 (外部时钟模式 A、B)、内部触发输入 (ISx) 这些时钟源提供。

图 14-8 计数时钟



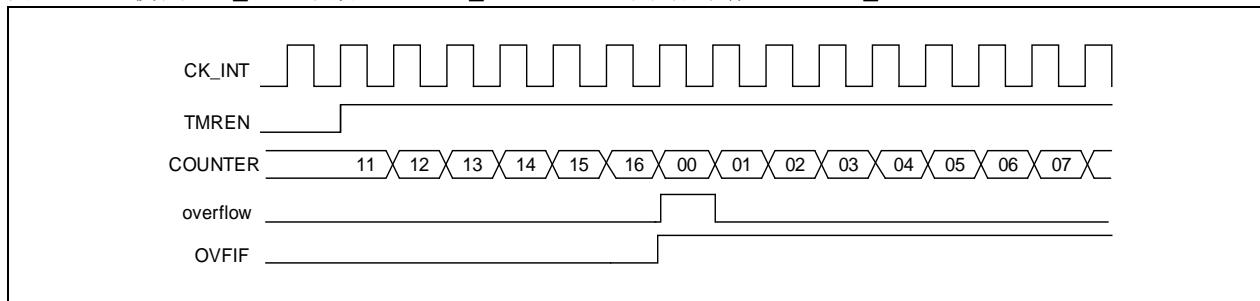
#### 内部时钟 (CK\_INT)

默认下使用 CK\_INT 经由预分频器驱动计数器计数, 当 TMR 对应的 APB 时钟预分频系数是 1 时, CK\_INT

频率等于 APB 时钟频率，否则 CK\_INT 频率等于 APB 时钟频率的 2 倍。相关配置流程如下：

- 配置 TMRx\_CTRL1 寄存器 TWCMSEL[1:0]，选择计数模式，若选择单向对齐计数模式，还需配置 TMRx\_CTRL1 寄存器 OWCDIR 选择计数方向。
- 配置 TMRx\_DIV 寄存器，设置计数器计数频率。
- 配置 TMRx\_PR 寄存器，设置计数器计数周期。
- 配置 TMRx\_CTRL1 寄存器 TMREN，使能计数器。

图 14-9 使用 CK\_INT 计数，TMRx\_DIV=0x0，周期寄存器 TMRx\_PR=0x16



### 外部时钟 (TRGIN/EXT)

计数时钟可由两种外部时钟源提供，分别为 TRGIN 和 EXT 信号。

当 SMSEL=3'b111 时，外部时钟模式 A 被选中，配置 STIS[2: 0] 来选择外部时钟源 TRGIN 信号驱动计数器计数。外部时钟源 TRGIN 可选则 C1INC (STIS=3'b100，通道 1 上升沿和下降沿信号)、C1IFP1 (STIS=3'b101，通道 1 滤波且极性选择后信号)、C2IFP2 (STIS=3'b110，通道 2 滤波且极性选择后信号) 和 EXT (STIS=3'b111，外部输入经极性选择、分频和滤波后信号)。

当 ECMBEN=1 时，外部时钟模式 B 被选中，计数器由外部输入经极性选择、分频和滤波后 EXT 信号驱动计数。外部时钟模式 B 等效于外部时钟模式 A 选择 EXT 信号作为外部时钟源 TRGIN。

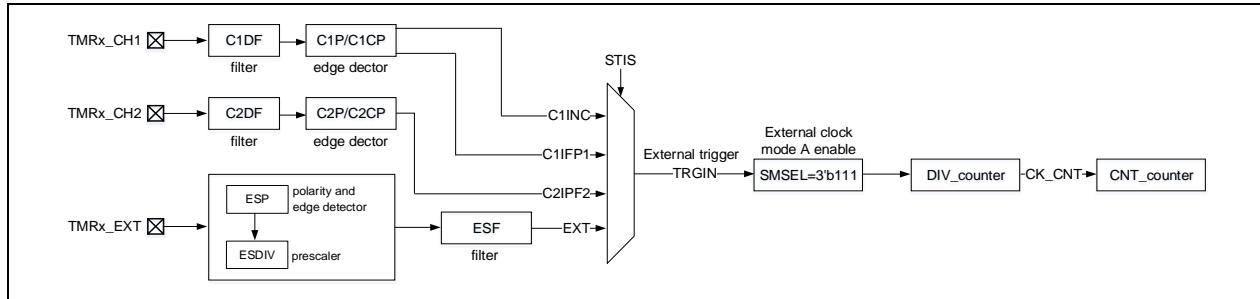
若要使用外部时钟模式 A，可按如下步骤配置：

- 配置外部时钟源 TRGIN 参数。
  - 若选择 TRGIN 来源为 TMRx\_CH1，需配置通道 1 输入滤波 (TMRx\_CM1 寄存器 C1DF[3:0]) 和通道 1 输入极性 (TMRx\_CCTRL 寄存器 C1P/C1CP)。
  - 若选择 TRGIN 来源为 TMRx\_CH2，需配置通道 2 输入滤波 (TMRx\_CM1 寄存器 C2DF[3:0]) 和通道 1 输入极性 (TMRx\_CCTRL 寄存器 C2P/C2CP)。
  - 若选择 TRGIN 来源为 TMRx\_EXT，需配置外部信号极性 (TMRx\_STCTRL 寄存器 ESP)、外部信号分频 (TMRx\_STCTRL 寄存器 ESDIV[1:0]) 和外部信号滤波 (TMRx\_STCTRL 寄存器 ESF[3:0])。
- 配置 TMRx\_STCTRL 寄存器 STIS[1:0]，设置 TRGIN 信号来源。
- 配置 TMRx\_STCTRL 寄存器 SMSEL=3'b111，使能外部时钟模式 A。
- 配置 TMRx\_DIV 寄存器 DIV[15:0]，设置计数器计数频率。
- 配置 TMRx\_PR 寄存器 PR[15:0]，设置计数器计数周期。
- 配置 TMRx\_CTRL1 寄存器 TMREN，使能计数器。

若要使用外部时钟模式 B，可按如下步骤配置：

- 配置 TMRx\_STCTRL 寄存器 ESP，设置外部信号极性。
- 配置 TMRx\_STCTRL 寄存器 ESDIV[1:0]，设置外部信号分频。
- 配置 TMRx\_STCTRL 寄存器 ESF[3:0]，设置外部信号滤波。
- 配置 TMRx\_STCTRL 寄存器 ECMBEN，使能外部时钟模式 B。
- 配置 TMRx\_DIV 寄存器 DIV[15:0]，设置计数器计数频率。
- 配置 TMRx\_PR 寄存器 PR[15:0]，设置计数器计数周期。
- 配置 TMRx\_CTRL1 寄存器 TMREN，使能计数器。

图 14-10 外部时钟模式 A 框图



注：由于同步逻辑，输入端信号与计数器实际时钟之间存在一定延时。

图 14-11 使用外部时钟模式 A 计数，PR=0x32, DIV=0x0

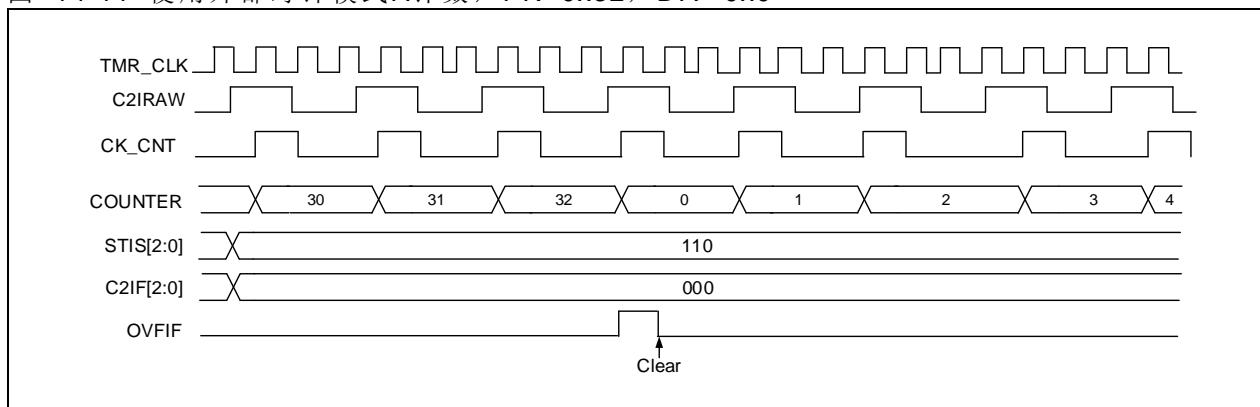
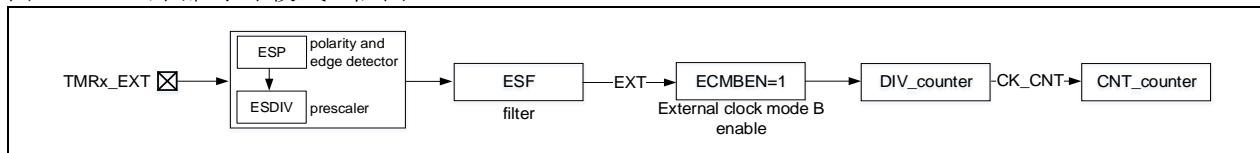
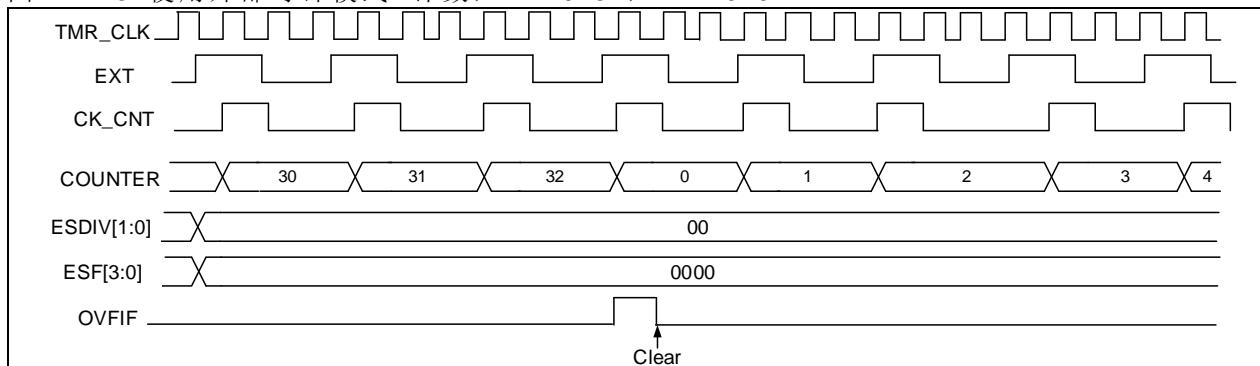


图 14-12 外部时钟模式 B 框图



注：由于同步逻辑。输入端 EXT 信号与计数器实际时钟之间存在一定延时。

图 14-13 使用外部时钟模式 B 计数，PR=0x32, DIV=0x0



### 内部触发输入 (ISx)

定时器之间支持互联同步，因此一个定时器的 TMR\_CLK 可由另一个定时器输出信号 TRGOUT 提供。配置 STIS[2: 0]选择内部触发信号驱动计数器计数。

TMR3 定时器内含一个 16 位预分频器，用于产生驱动计数器计数的时钟 CK\_CNT，通过配置 TMR3\_DIV 寄存器值，可灵活调整 CK\_CNT 与 TMR\_CLK 之间的分频关系。预分频值可在任何时刻修改，但只在下一个溢出事件发生时，新值才会生效。

内部触发输入配置流程如下：

- - 配置 TMRx\_PR 寄存器，设置计数器计数周期。
- - 配置 TMRx\_DIV 寄存器，设置计数器计数频率。
- - 配置 TMRx\_CTRL1 寄存器 TWCMSEL[1:0]位，设置计数器计数模式。
- - 配置 TMRx\_STCTRL 寄存器 STIS[2:0]位范围为 3'b000~3'b011，选择内部触发。
- - 配置 TMRx\_STCTRL 寄存器 SMSEL[2:0]=3'b111，选择外部时钟模式 A。

-配置 TMRx\_CTRL1 寄存器 TMREN 位，使能 TMRx 计数。

表 14-3 TMR3 内部触发连接

次定时器	ISO (STIS = 000)	IS1 (STIS = 001)	IS2 (STIS = 010)	IS3 (STIS = 011)
TMR3	TMR1	-	TMR15	-

注意 1：如果某个产品中没有相应的定时器，则对应的触发信号 ISx 也不存在。

图 14-14 当预分频器的参数从 1 变到 4 时，计数器的时序图

### 14.2.3.2 计数模式

TMR3 定时器提供了多种计数模式，用来满足不同的应用场景。其内部拥有一个支持 16 位向上计、向下、中央双向对齐计数的计数器。

TMRx\_PR 寄存器用于设置计数器计数周期。默认 TMRx\_PR 寄存器值会立即传入它的影子寄存器；当开启周期缓冲功能后（PRBEN 置 1），TMRx\_PR 寄存器值在溢出事件发生时传入它的影子寄存器。

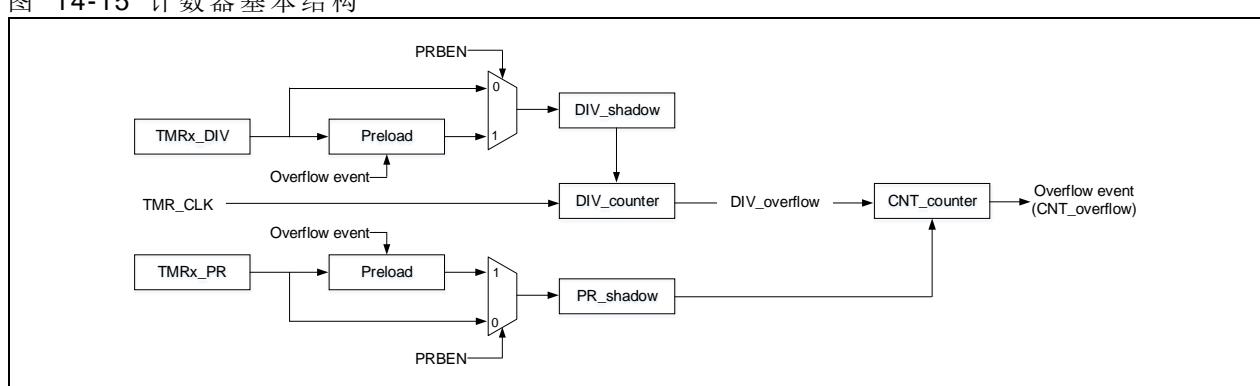
TMRx\_DIV 寄存器用于设置计数器计数频率，每（DIV[15:0]+1）个计数时钟周期，计数器计数一次。和 TMRx\_PR 寄存器类似，开启周期缓冲功能后，TMRx\_DIV 寄存器值在溢出事件时更新至它的影子寄存器。

读取 TMRx\_CNT 寄存器会返回当前计数器计数值，写入 TMRx\_CNT 寄存器会更新计数器当前计数值为写入值。

默认允许产生溢出事件，设置 TMRx\_CTRL1 寄存器 OVREN=1 将禁止更新事件产生。TMRx\_CTRL1 寄存器 OVFS 用于选择溢出事件来源，默认计数器上溢或下溢、置位 OVFSWTR、复位模式次定时器控制器产生的复位信号产生溢出事件。置位 OVFS 后，只有计数器上溢或下溢产生溢出事件。

TMREN 位置 1 将使能定时器计数，由于同步逻辑，实际驱动计数器的使能信号 TMR\_EN 相对于 TMREN 延迟一个时钟周期。

图 14-15 计数器基本结构



#### 向上计数模式

配置 TMRx\_CTRL1 寄存器 CMSEL[1:0]=2'b00，OWCDIR=1'b0 开启向上计数模式，计数值达到 TMRx\_PR 值时，重新从 0 向上计数，计数器上溢并产生溢出事件，同时 OVFIF 位置 1。若禁止产生溢出事件，计数器溢出后不再重载预分频值和周期值，否则预分频值和周期值在溢出事件后更新。

图 14-16 PRBEN=0时的溢出事件

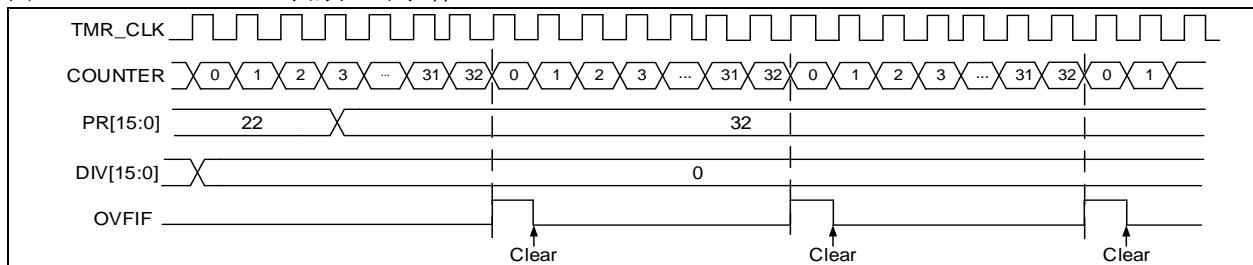
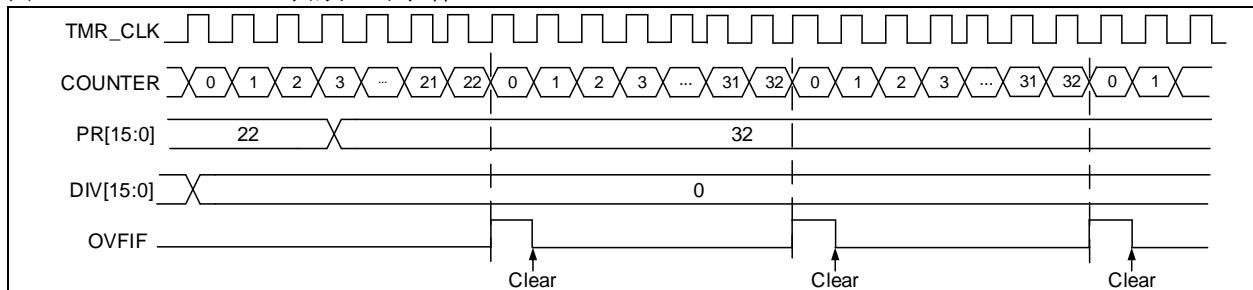


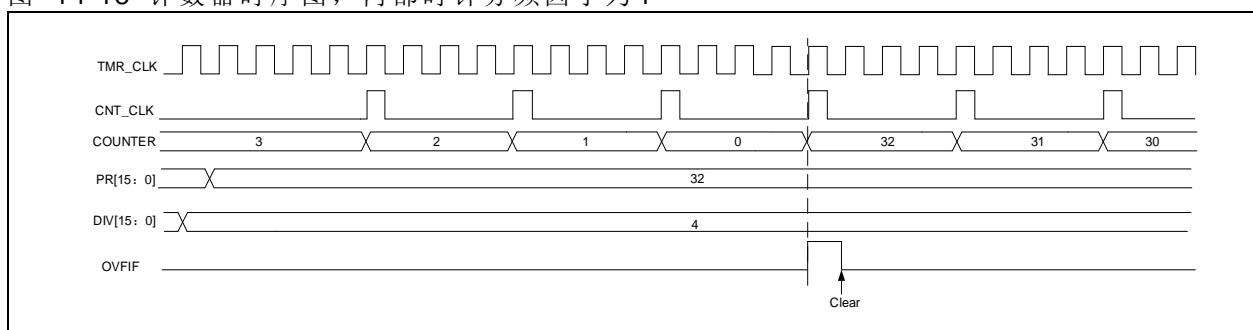
图 14-17 PRBEN=1时的溢出事件



### 向下计数模式

配置 TMRx\_CTRL1 寄存器 CMSEL[1:0]=2'b00, OWCDIR=1'b1 开启向下计数模式，计数值达到 0 值并重新从 TMRx\_PR 向上下数时，计数器下溢并产生溢出事件。

图 14-18 计数器时序图，内部时钟分频因子为4



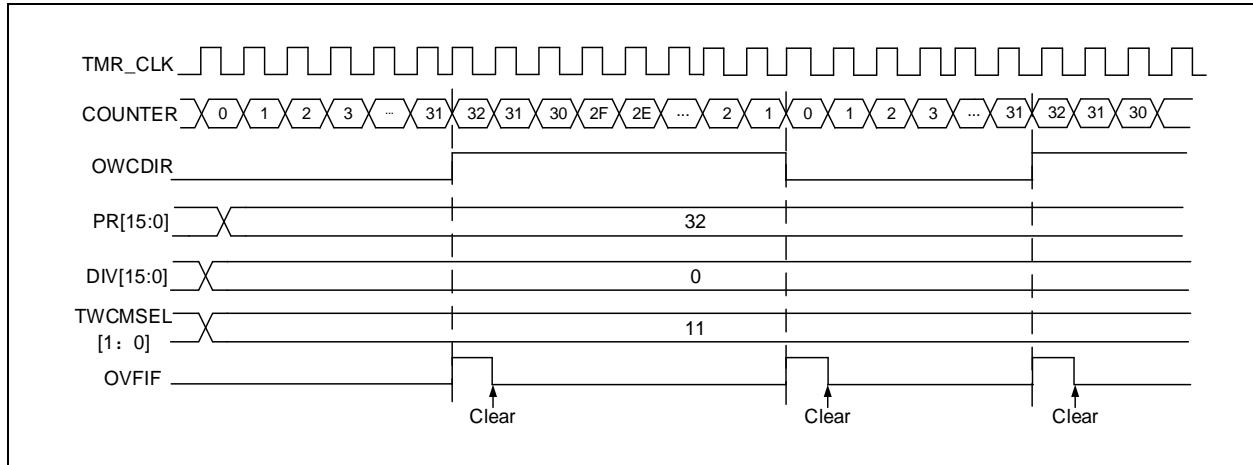
### 中央双向对齐计数模式

配置 TMRx\_CTRL1 寄存器 CMSEL[1:0]≠2'b00 开启中央双向对齐计数模式，中央双向对齐计数模式下计数器交替向上、向下计数。计数值从 TMRx\_PR 值向下计数到 1 值，产生下溢事件，然后从 0 开始向上计数；向上计数到 TMRx\_PR 值-1，产生上溢事件，之后从 TMRx\_PR 值向下计数。计数器计数方向由计数器方向控制位（OWCDIR）实时查看。

TMRx\_CTRL1 寄存器 TWCSEL[1:0]位还用于选择中央双向对齐计数模式下 CxIF 标志置起方式，中央双向对齐计数模式 1 (TWCSEL[1:0]=2'b01) 仅允许 CxIF 标志位在计数器向下计数时置起；双向对齐计数模式 2 (TWCSEL[1:0]=2'b10) 仅允许 CxIF 标志位在计数器向上计数时置起；双向对齐计数模式 3 (TWCSEL[1:0]=2'b11) 允许 CxIF 标志位在计数器向上和向下计数时置起。

注意： 中央双向对齐计数模式下，OWCDIR 位为只读位。

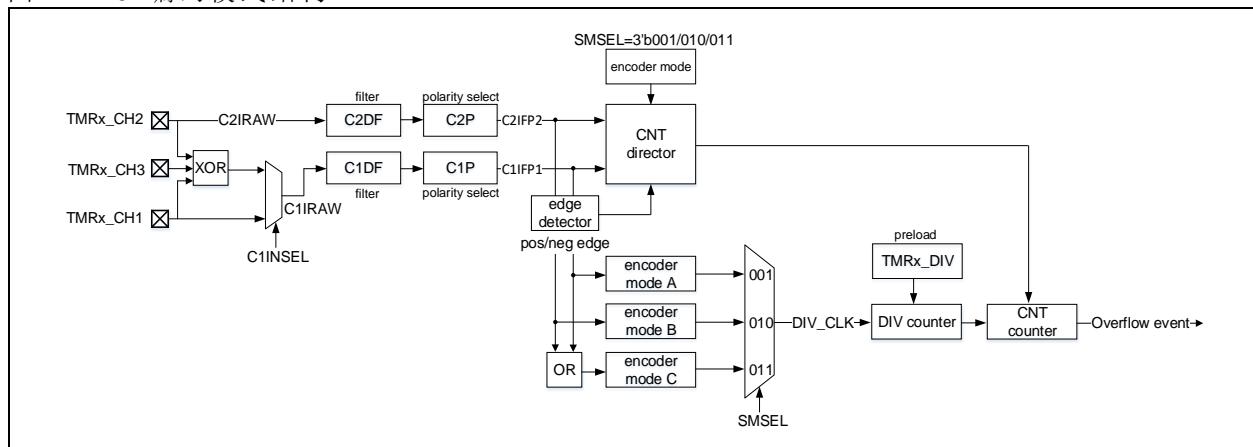
图 14-19 计数器时序图，内部时钟分频因子为1，TMRx\_PR=0x32



### 编码器模式

编码器模式下需提供两组输入信号 TMRx\_CH1 和 TMRx\_CH2，根据一组输入信号电平值，计数器在另一组输入信号边沿向上或向下计数。计数方向由 OWCDIR 值指示。

图 14-20 编码模式结构



编码器模式 A: SMSEL=3'b001，计数器在 C1IFP1 边沿计数（上升沿和下降沿），计数方向由 C1IFP1 边沿方向和 C2IFP2 电平高低共同决定。

编码器模式 B: SMSEL=3'b010，计数器在 C2IFP2 边沿计数（上升沿和下降沿），计数方向由 C2IFP2 边沿方向和 C1IFP1 电平高低共同决定。

编码器模式 C: SMSEL=3'b011，计数器在 C1IFP1 和 C2IFP2 边沿计数（上升沿和下降沿），计数方向由 C1IFP1 边沿方向和 C2IFP2 电平高低、C2IFP2 边沿方向和 C1IFP1 电平高低共同决定共同决定。

若要使用编码器模式可按下面步骤配置：

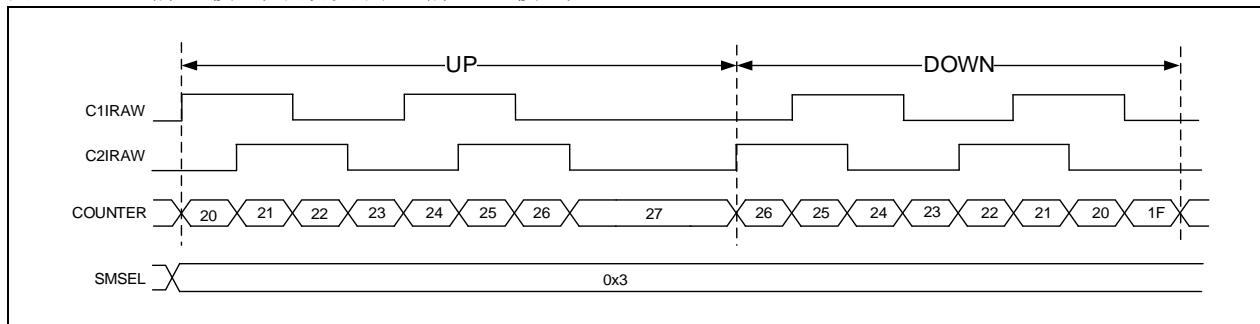
- 配置 TMRx\_CM1 寄存器 C1DF[3:0]，设置通道 1 输入信号滤波；配置 TMRx\_CCTRL 寄存器 C1P，设置通道 1 输入信号有效电平。
- 配置 TMRx\_CM1 寄存器 C2DF[3:0]，设置通道 2 输入信号滤波；配置 TMRx\_CCTRL 寄存器 C2P，设置通道 2 输入信号有效电平。
- 配置 TMRx\_CM1 寄存器 C1C[1:0]，设置通道 1 为输入模式；配置 TMRx\_CM1 寄存器 C2C[1:0]，设置通道 2 为输入模式；
- 配置 TMRx\_STCTRL 寄存器 SMSEL[2:0]，选择编码器模式 A (SMSEL=3'b001)、编码器模式 B (SMSEL=3'b010) 或编码器模式 C (SMSEL=3'b011)。
- 配置 TMRx\_PR 寄存器 PR[15:0]，设置计数器计数周期。
- 配置 TMRx\_DIV 寄存器 DIV[15:0]，设置计数器计数频率。
- 配置 TMRx\_CH1 和 TMRx\_CH2 对应 IO 为复用模式。
- 配置 TMRx\_CTRL1 寄存器 TMREN，使能计数器。

编码模式下计数器计数方向如下表所示：

表 14-4 计数方向与编码器信号的关系

计数边沿	计数边沿相对信号的电平 (C1IFP1 边沿对应 C2IFP2 电平, C2IFP2 边沿对应 C1IFP1 电平)	C1IFP1 边沿方向		C2IFP2 边沿方向	
		上升	下降	上升	下降
C1IFP1	高	向下计数	向上计数	不计数	不计数
	低	向上计数	向下计数	不计数	不计数
C2IFP2	高	不计数	不计数	向上计数	向下计数
	低	不计数	不计数	向下计数	向上计数
C1IFP1 和 C2IFP2	高	向下计数	向上计数	向上计数	向下计数
	低	向上计数	向下计数	向下计数	向上计数

图 14-21 编码模式计数实例 (编码器模式 C)



### 14.2.3.3 TMR输入部分

TMR3 拥有 4 个独立通道，每个通道可配置为输入或输出，当配置位输入时，每个通道输入信号依次经过以下处理：

-TMRx\_CHx 经过预处理输出 CxIRAW。配置 C1INSEL 位，选择 C1IRAW 来源是 TMRx\_CH1 或是 TMRx\_CH1、TMRx\_CH2、TMRx\_CH3 异或。C2IRAW、C3IRAW、C4IRAW 来源是 TMRx\_CH2、TMRx\_CH3、TMRx\_CH4。

-CxIRAW 输入数字滤波器，输出滤波后信号 CxIF。数字滤波器通过 CxDI 位配置采样频率和次数。

-CxIF 输入边沿检测器，输出边沿选择后信号 CxIFPx。边沿选择由 CxP 和 CxCP 位共同控制，可选择输入上升沿、下降沿或双边沿有效。

-CxIFPx 输入捕获信号选择器，输出选择后信号 CxIN。捕获信号选择器由 CxC 控制，可选择 CxIN 来源为 CxIFPx、CxIFPy、STCI。其中 CxIFPy ( $x \neq y$ ) 是来自通道 y 的 CxIFPx 经通道 x 边沿检测器处理后的信号（例如 C1IFP2 是来自通道 1 的 C1IFP1 信号经过通道 2 边沿检测器处理后的信号）；STCI 来自次定时器控制器，由 STIS 位选择来源。

-CxIN 经由输入通道分频器，输出分频后信号 CxIPS。分频系数由 CxIDIV 位配置为不分频、2 分频、4 分频或 8 分频。

图 14-22 输入/输出通道 1 的主电路

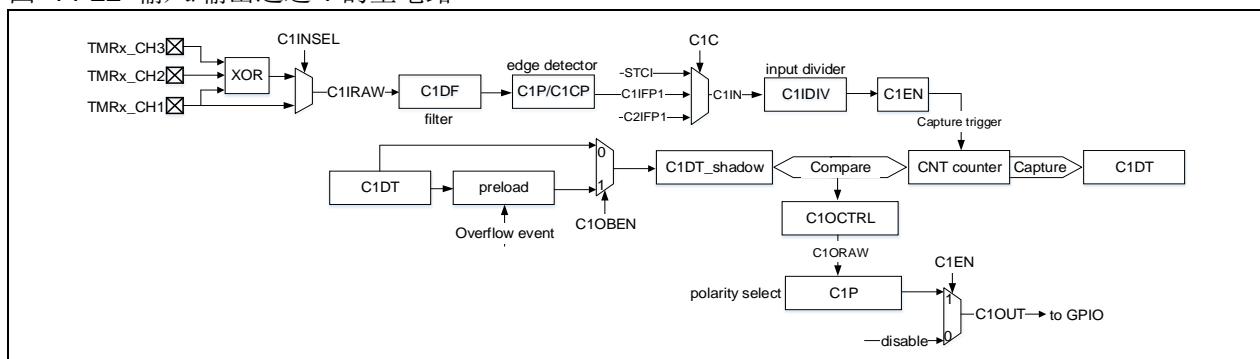
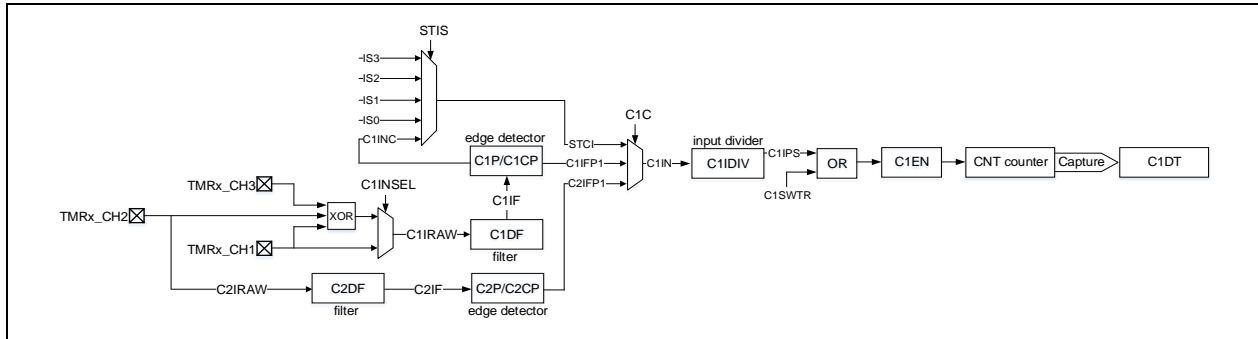


图 14-23 通道1输入部分



### 输入模式

此模式下，当选中的触发信号被检测到，通道寄存器（TMRx\_CxDT）记录当前计数器计数值，并将捕获比较中断标志位（CxIF）置 1，若已使能通道中断（CxIEN）、通道 DMA 请求（CxDEN）则产生相应的中断和 DMA 请求。若在 CxIF 置 1 后检测到触发信号，将产生捕获溢出事件，TMRx\_CxDT 会使用当前计数器计数值覆盖之前记录的计数器计数值，同时通道再捕获标志位（CxRF）置 1。

若要捕获 C1IN 输入的上升沿，可按如下进行配置：

- 将通道模式寄存器 1（TMR3\_CM1）中的 C1C 位配置为 01，选择 C1IN 作为通道 1 输入。
- 配置 C1IN 信号滤波器带宽（CxDF[3: 0]）。
- 配置 C1IN 通道的有效沿，在通道控制寄存器（TMR3\_CCTRL）中写入 C1P=0（上升沿）。
- 配置 C1IN 信号捕获频率（C1DIV[1: 0]）。
- 使能通道 1 输入捕获（C1EN=1）。
- 根据需要设置 DMA/中断使能寄存器（TMR3\_IDEN）中的 C1IEN 为、DMA/中断使能寄存器（TMR3\_IDEN）中的 C1DEN 位，选择中断请求或 DMA 请求。

### 多输入异或

通道 1 的输入端可选择 TMR3\_CH1、TMR3\_CH2 和 TMR3\_CH3 经异或逻辑后输入。将控制寄存器 2（TMR3\_CTRL2）中的 C1INSEL 位置 1 可开启此功能。

多输入异或功能可用于连接霍尔传感器，例如，将异或输入的三个输入端分别连接到三个霍尔传感器，通过分析三路霍尔传感器信号可计算出转子的位置和速度。

### PWM 输入

PWM 输入模式适用于通道 1 和 2，要使用此模式，需要将 C1IN 和 C2IN 映射到同一 TMRx\_CHx，并且通道 1 或 2 的 CxIFPx 配置成触发次定时器控制器复位。

PWM 输入模式可用于测量输入信号的周期和占空比，如需测量通道 1 输入信号的周期和占空比，操作步骤如下：

- 配置 C1C=2'b01，选择 C1IN 为 C1IFP1。
- 配置 C1P=1'b0，选择 C1IFP1 上升沿有效。
- 配置 C2C=2'b10，选择 C2IN 为 C1IFP2。
- 配置 C2P=1'b1，选择 C1IFP2 下降沿有效。
- 配置 STIS=3'b101，选择次定时器触发信号为 C1IFP1。
- 配置 SMSEL=3'b100，选择次定时器模式为复位模式。
- 配置 C1EN=1'b1，C2EN=1'b1。使能通道 1 和输入捕获。

上述配置下，通道 1 输入信号的上升沿会触发捕获并将捕获值存储到 C1DT 寄存器，同时通道 1 输入信号上升沿复位计数器。通道 1 输入信号下降沿触发捕获并将捕获值存储到 C2DT 寄存器。通道 1 输入信号的周期可通过 C1DT 计算，占空比可通过 C2DT 计算。

图 14-24 PWM输入模式配置实例

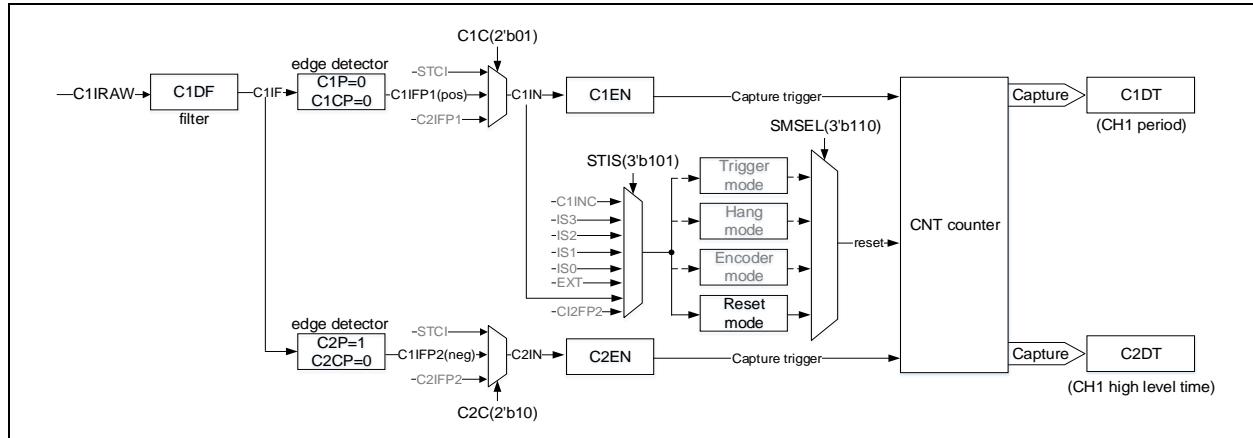
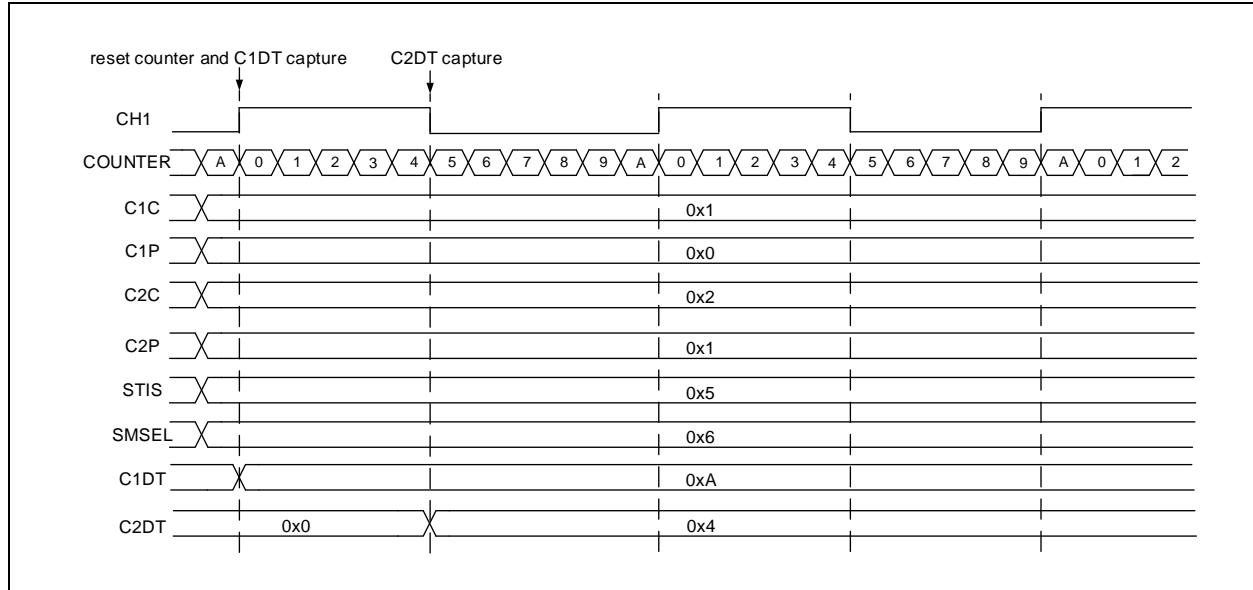


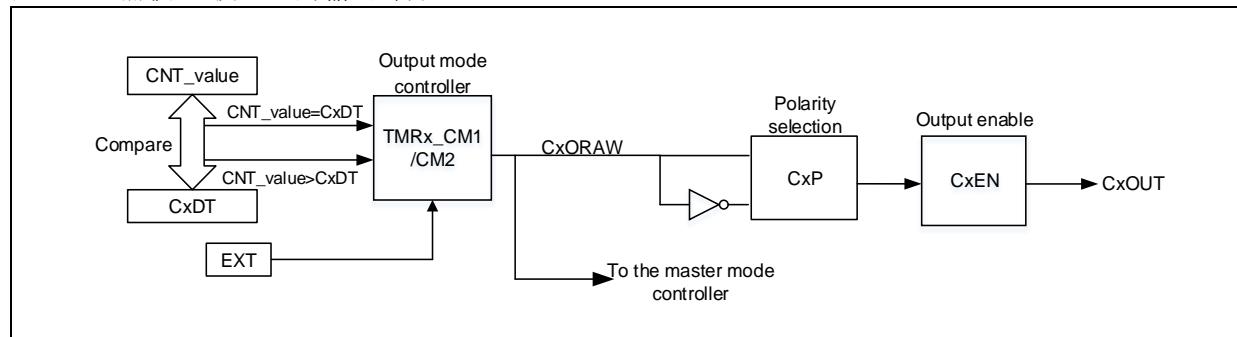
图 14-25 PWM输入模式



#### 14.2.3.4 TMR输出部分

TMR 的输出部分由比较器和输出控制构成，用于编程输出信号的周期、占空比、极性。

图 14-26 捕获/比较通道的输出部分（通道 1 至 4）



##### 输出模式

配置  $CxC[1: 0] \neq 2'b00$  将通道配置为输出可实现多种输出模式，此时，计数器计数值将与通道寄存器（TMR3\_CxDT）值比较，并根据 CxOCTRL[2: 0]位配置的输出模式，产生中间信号 CxORAW，再经过输出控制逻辑处理后输送到 IO。输出信号的周期由周期寄存器（TMR3\_PR）值配置，占空比则由通道寄存器（TMR3\_CxDT）值配置。

输出比较模式有以下子类：

**PWM 模式 A:** CxOCTRL=3'b110 时，开启 PWM 模式 A。向上计数时， $TMRx_C1DT > TMRx_CVAL$  时 C1ORAW 输出高电平，否则为低电平；向下计数时， $TMRx_C1DT < TMRx_CVAL$  时 C1ORAW 输出低电平，否则为高电平。若要使用 PWM 模式 A，可按如下方式配置。

- 配置 TMRx\_PR 寄存器，设置 PWM 周期。
- 配置 TMRx\_CxDT 寄存器，设置 PWM 占空比。
- 配置 TMRx\_CM1/CM2 寄存器 CxOCTRL 位为 3'b110，设置输出模式为 PWM 模式 A。
- 配置 TMRx\_DIV 寄存器，设置计数器计数频率。
- 配置 TMRx\_CTRL1 寄存器 TWCMSEL[1:0]位，设置计数器计数模式。
- 配置 TMRx\_CCTRL 寄存器 CxP 位、CxCP 位，设置输出极性。
- 配置 TMRx\_CCTRL 寄存器 CxEEN 位、CxCPEN 位，使能通道输出。
- 配置 TMRx\_BRK 寄存器 OEN 位，使能 TMRx 输出。
- 配置 TMRx 输出通道对应 GPIO 为对应的复用模式。
- 配置 TMRx\_CTRL1 寄存器 TMREN 位，使能 TMRx 计数。

**PWM 模式 B:** CxOCTRL=3'b111 时，开启 PWM 模式 B。向上计数时， $TMRx\_C1DT > TMRx\_CVAL$  时 C1ORAW 输出低电平，否则为高电平；向下计数时， $TMRx\_C1DT < TMRx\_CVAL$  时 C1ORAW 输出高电平，否则为低电平。

**强制输出模式:** CxOCTRL=3'b100/101 时，开启强制输出模式。此时，CxORAW 信号的电平被强制输出为配置的电平，而与计数值无关。虽然输出信号不依赖于比较结果，但通道标志位和 DMA 请求仍依赖于比较结果。

**输出比较模式:** CxOCTRL=3'b001/010/011 时，开启输出比较模式。此时，当计数值与 CxDT 值匹配时，CxORAW 强制输出高电平（CxOCTRL=3'b001）、低电平（CxOCTRL=3'b010）或进行电平翻转（CxOCTRL=3'b011）。

**单周期模式:** PWM 模式的特例，将 OCMEN 位置 1 可开启单周期模式，此模式下，仅在当前计数周期中进行比较匹配，完成当前计数后，TMREN 位清 0，因此仅输出一个脉冲。当配置为向上计数模式时，需要严格配置  $CVAL < CxDT \leq PR$ ；向下计数时，需严格配置  $CVAL > CxDT$ 。

**快速输出模式:** 将 CxOIEN 位置 1 可开启此功能，开启后 CxORAW 电平值不再在计数值与 CxDT 匹配时变化，而是在当前计数周期开始时，也就是说，比较结果被提前了，计数器值与通道寄存器（TMR3\_CxDT）的比较结果将会提前决定 CxORAW 的电平。

图 14-27 展示了输出比较模式（翻转）的例子， $C1DT=0x3$ ，当计数值等于 0x3 时，输出电平 C1OUT 被翻转。

图 14-28 展示了计数器向上计数与 PWM 模式 A 配合的例子， $PR=0x32$ ，CxDT 配置为不同的值时输出时输出信号的翻转情况。

图 14-29 展示了计数器中央双向对齐计数与 PWM 模式 A 配合的例子， $PR=0x32$ ，CxDT 配置为不同的值时输出时输出信号的翻转情况。

图 14-30 展示了计数器向上计数与单周期模式下 PWM 模式 B 配合的例子，计数器仅计数了一个周期，输出信号在这个周期中只输出了一个脉冲。

图 14-27 计数值与 C1DT 值匹配时翻转 C1ORAW

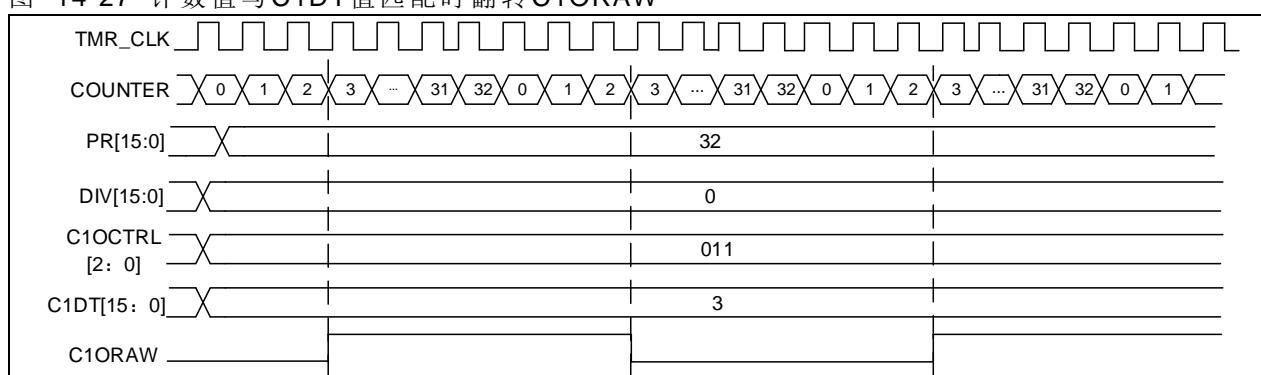


图 14-28 向上计数下 PWM 模式 A

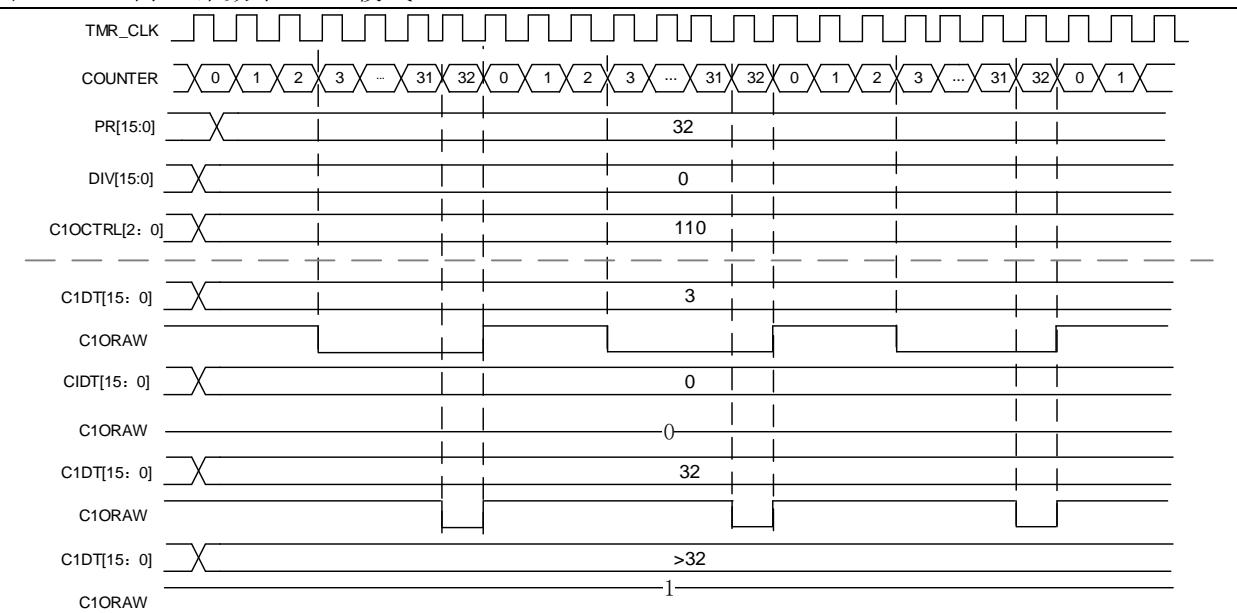


图 14-29 中央双向对齐计数下 PWM 模式 A

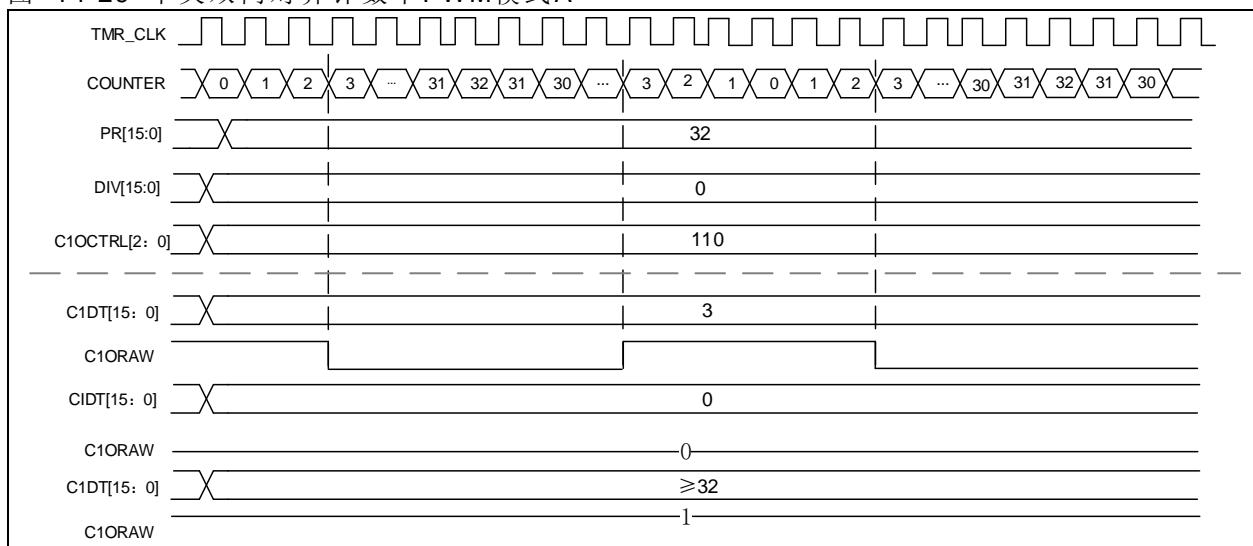
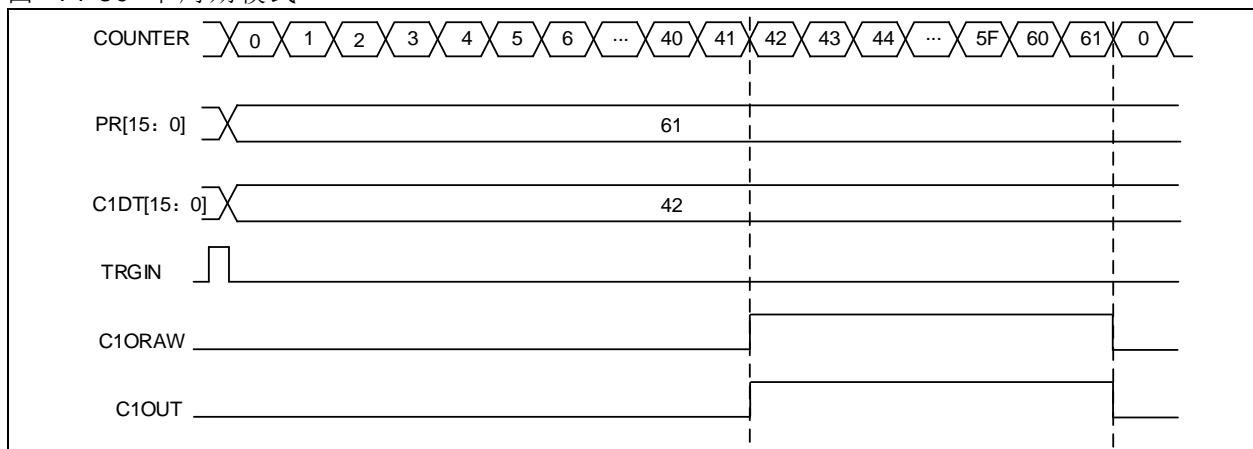


图 14-30 单周期模式



### 主定时器事件输出

当 TMR 作为主定时器时，可选择如下信号源作为 TRGOUT 信号输出到次定时器，选择信号为 TMRxCTRL2 寄存器 PTOS 位。

-PTOS=3'b000, TRGOUT 输出软件溢出事件 (TMRx\_SWEVT 寄存器 OVFSWTR 位)。

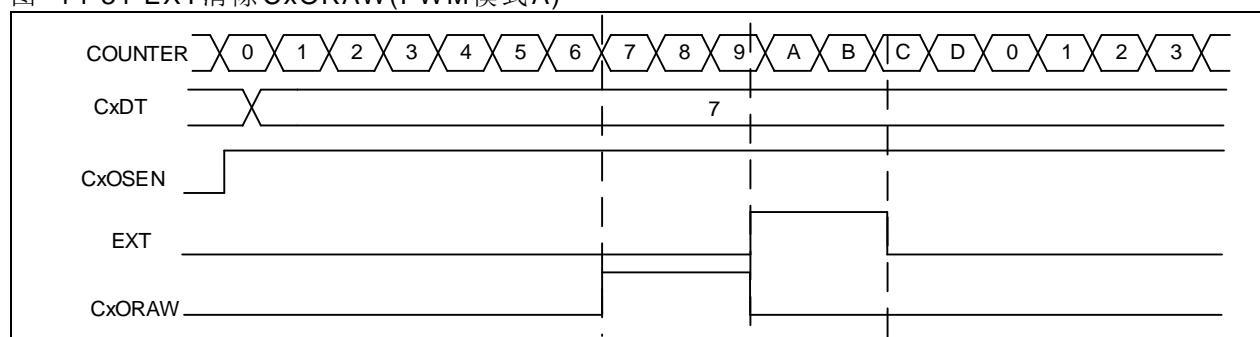
- PTOS=3'b001, TRGOUT 输出计数器使能信号。
- PTOS=3'b010, TRGOUT 输出计数器溢出事件。
- PTOS=3'b011, TRGOUT 输出捕获、比较事件。
- PTOS=3'b100, TRGOUT 输出 C1ORAW 信号。
- PTOS=3'b101, TRGOUT 输出 C2ORAW 信号。
- PTOS=3'b110, TRGOUT 输出 C3ORAW 信号。
- PTOS=3'b111, TRGOUT 输出 C4ORAW 信号。

### CxORAW 信号清除

将 CxOSEN 位置 1 后, 指定通道的 CxORAW 信号由 EXT 高电平清 0, 在下一次溢出事件发生前 CxORAW 信号无法被改变。

强制输出模式时, CxORAW 信号清除功能不可用, 只有在输出比较模式或 PWM 模式, 此功能有效。下图显示了使用 EXT 信号清除 CxORAW 的例子, 当 EXT 为高电平时, 原本为高电平的 CxORAW 信号被拉低, 当 EXT 为低电平时, CxORAW 根据计数值和 CxDT 比较结果输出电平。

图 14-31 EXT 清除 CxORAW(PWM 模式 A)



### 14.2.3.5 定时器同步

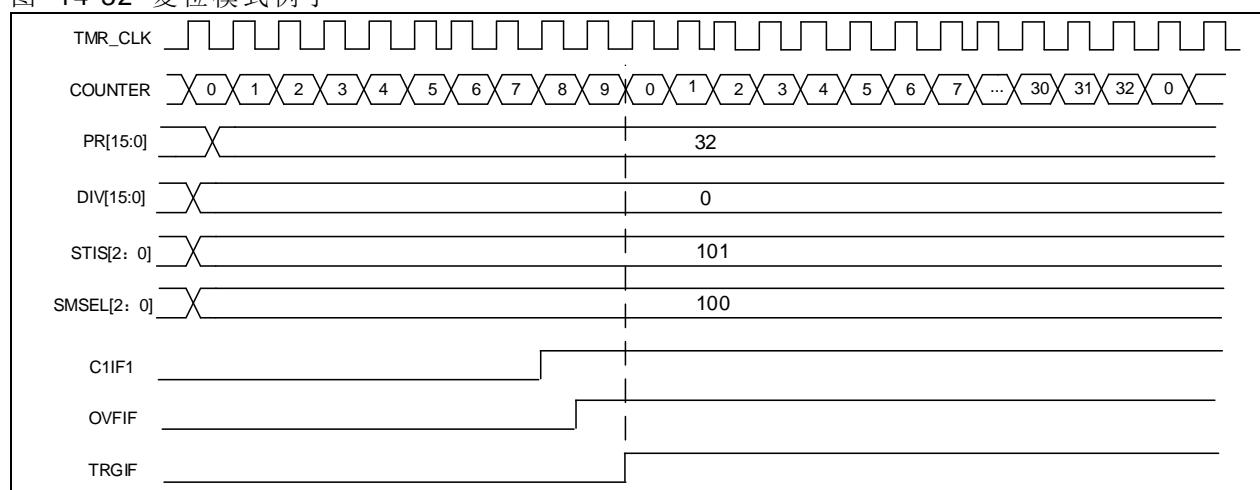
主次定时器之间可由内部连接信号进行同步。主定时器可由 PTOS[2: 0]位选择主定时器输出, 即同步信息; 次定时器由 SMSEL[2: 0]位选择从模式, 即次定时器的工作模式。

定时器从模式有以下几种:

#### 从模式: 复位模式

选中的触发信号将复位计数器和预分频器, 若 OVFS 位为 0, 将产生一个溢出事件。

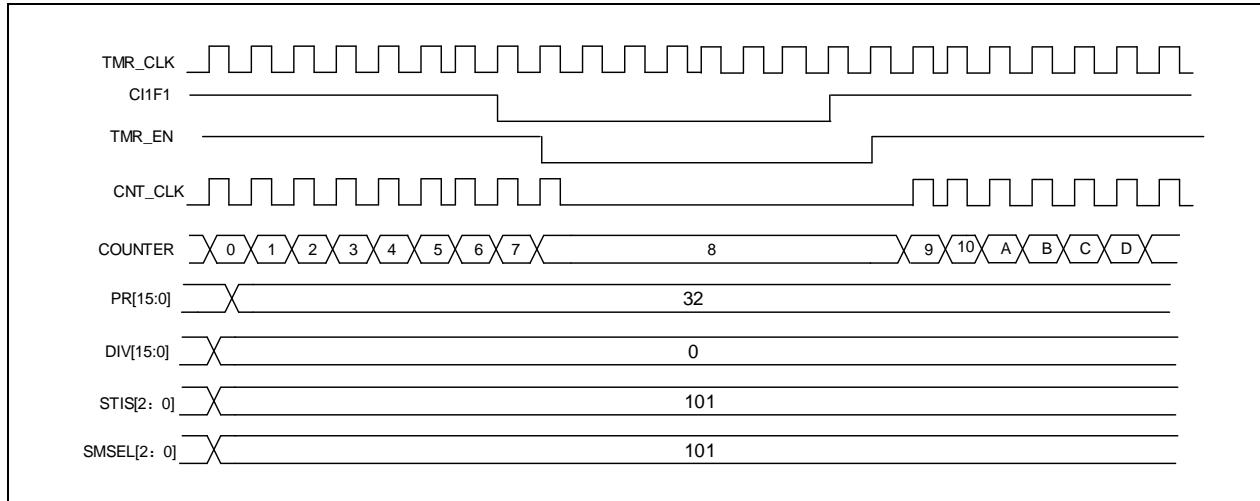
图 14-32 复位模式例子



#### 从模式: 挂起模式

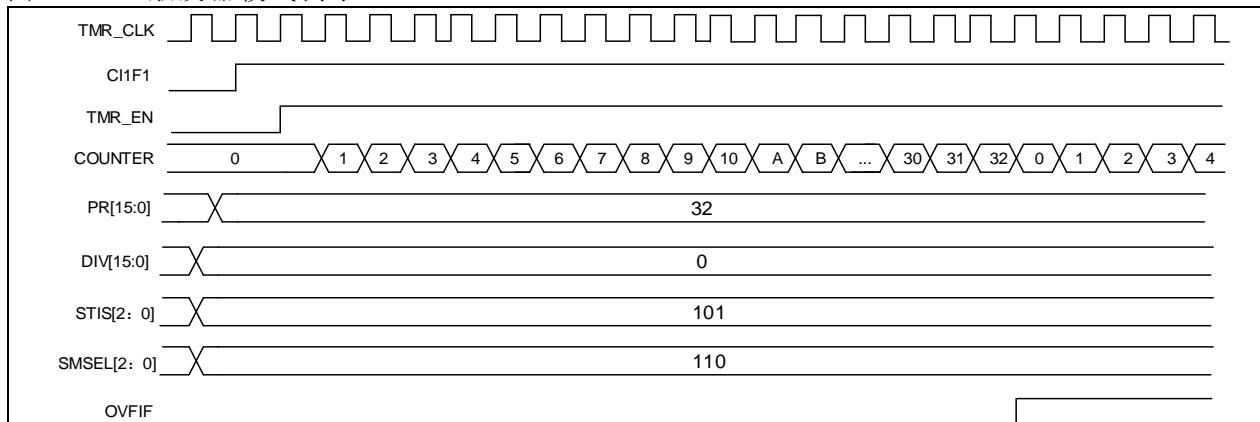
挂起模式下, 计数的计数和停止受选中触发输入信号控制, 当触发输入为高电平时计数器开始计数; 当为低电平时, 计数器暂停计数。

图 14-33 挂起模式下例子

**从模式：触发模式**

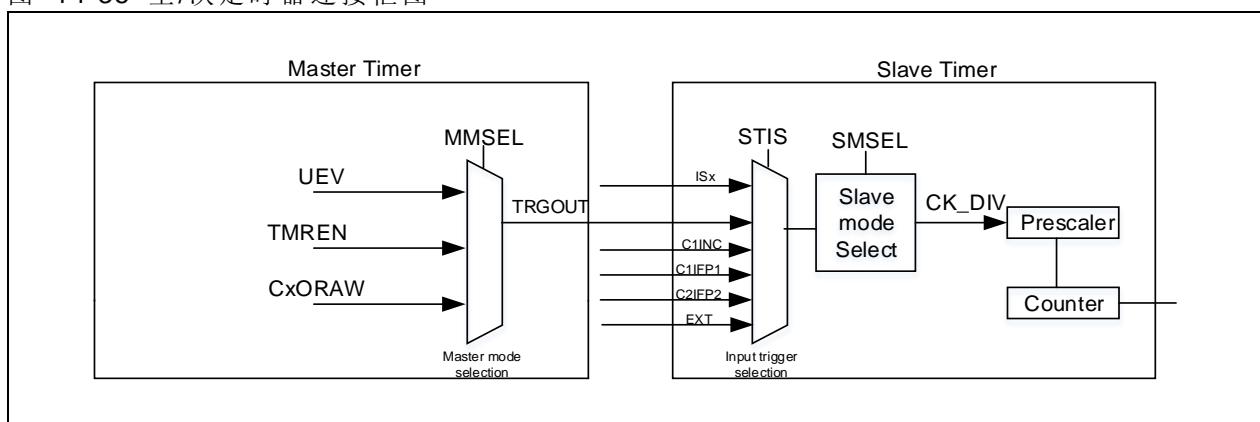
计数器将在选中的触发输入上升沿启动计数（将 TMR\_EN 置 1）。

图 14-34 触发器模式例子

**主/次定时器互联实例**

主/次定时器可分别配置不同的主模式和从模式，两者搭配可实现多种功能，一下提供了一些定时器互联的例子。

图 14-35 主/次定时器连接框图



主定时器为次定时器提供时钟：

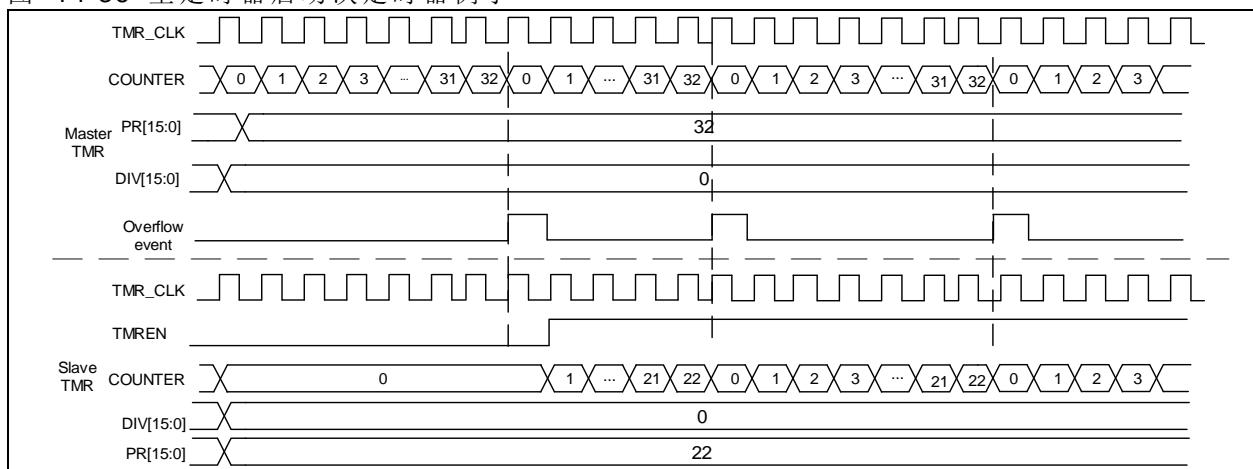
- 配置主定时器输出信号 TRGOUT 为溢出事件，配置 PTOS[2: 0]=3'b010，主定时器每次计数器溢出输出一个脉冲信号，用作次定时器计数时钟。
- 配置主定时器计数周期（周期寄存器（TMRx\_PR））。
- 配置次定时器触发输入信号 TRGIN 为主定时器输出（次定时器控制寄存器（TMRx\_STCTRL）的 STIS[2: 0]）。

- 配置次定时器使用外部时钟模式 A（次定时器控制寄存器（TMRx\_STCTRL）的 SMSEL[2: 0]=3'b111）。
- 将主定时器和次定时器的 TMREN 位置 1 启动定时器。

主定时器启动次定时器：

- 配置主定时器输出信号 TRGOUT 为溢出事件，配置 PTOS[2: 0]=3'b010，主定时器每次计数器溢出输出一个脉冲信号，用作次定时器计数时钟。
- 配置主定时器计数周期（周期寄存器（TMRx\_PR））。
- 配置次定时器触发输入 TRGIN 为主定时器输出。
- 配置次定时器为触发模式（TMRx\_STCTRL 寄存器的 SMSEL=3'b110）
- 置主定时器 TMREN=1 以启动主定时器。

图 14-36 主定时器启动次定时器例子

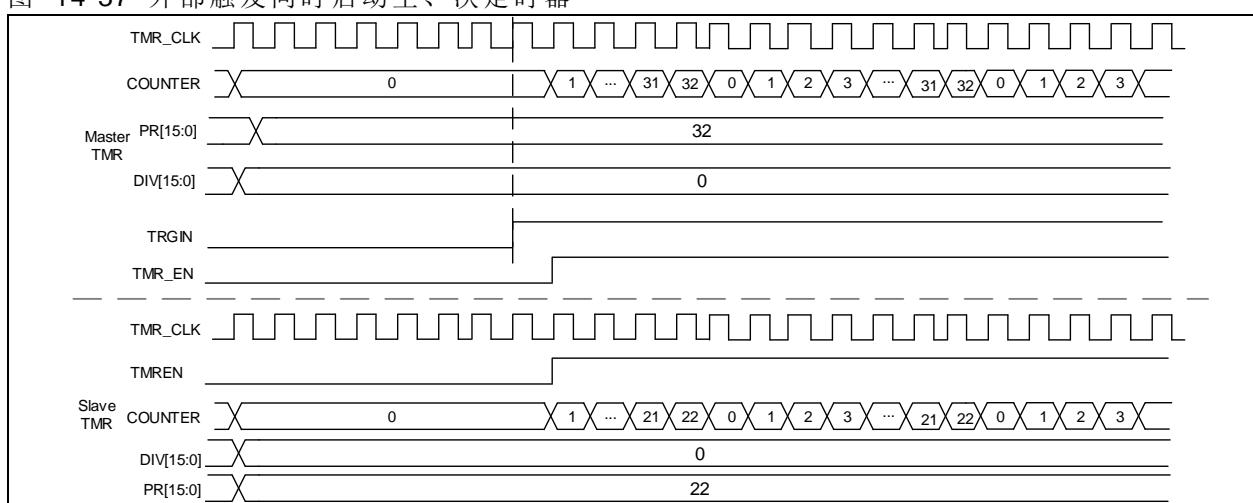


外部触发信号同步启动主、次定时器：

这个例子中，主定时器同时作为主定时器和次定时器，将主定时器的次定时器同步功能开启，此模式用于将主定时器和次定时器保持同步。

- 配置主定时器 STS 位为 1。
- 配置主定时器输出信号 TRGOUT 为溢出事件，配置 PTOS[2: 0]=3'b010，主定时器每次计数器溢出输出一个脉冲信号，用作次定时器计数时钟。
- 配置主定时器的次定时模式为触发模式，触发源选择 C1IN。
- 配置次定时器触发输入 TRGIN 为主定时器输出。
- 配置次定时器为触发模式（TMRx\_STCTRL 寄存器的 SMSEL=3'b110）。

图 14-37 外部触发同时启动主、次定时器



### 14.2.3.6 调试模式

当微控制器进入调试模式（Cortex®-M4 核心停止）时，将 DEBUG 模块中的 TMR3\_PAUSE 置 1，可以使 TMR3 计数器暂停计数。

## 14.2.4 TMR3寄存器描述

必须以字（32位）的方式操作这些外设寄存器。

下表中将TMR3的所有寄存器映射到一个16位可寻址（编址）空间。

表 14-5 TMR3寄存器映像和复位值

寄存器简称	基址偏移量	复位值
TMR3_CTRL1	0x00	0x0000
TMR3_CTRL2	0x04	0x0000
TMR3_STCTRL	0x08	0x0000
TMR3_IDEN	0x0C	0x0000
TMR3ISTS	0x10	0x0000
TMR3_SWEVT	0x14	0x0000
TMR3_CM1	0x18	0x0000
TMR3_CM2	0x1C	0x0000
TMR3_CCTRL	0x20	0x0000
TMR3_CVAL	0x24	0x0000 0000
TMR3_DIV	0x28	0x0000
TMR3_PR	0x2C	0x0000 0000
TMR3_C1DT	0x34	0x0000 0000
TMR3_C2DT	0x38	0x0000 0000
TMR3_C3DT	0x3C	0x0000 0000
TMR3_C4DT	0x40	0x0000 0000
TMR3_DMACTRL	0x48	0x0000
TMR3_DMADT	0x4C	0x0000

### 14.2.4.1 控制寄存器1 (TMR3\_CTRL1)

域	简称	复位值	类型	功能
位 15: 10	保留	0x00	resd	保持默认值。
位 9: 8	CLKDIV	0x0	rw	时钟除频 (Clock divider) 此位用于设置数字滤波器采样频率 $f_{DTS}$ 和定时器时钟频率 $f_{CK\_INT}$ 之间的分频比。 00: 无除频, $f_{DTS}=f_{CK\_INT}$ ; 01: 2 除频, $f_{DTS}=f_{CK\_INT}/2$ ; 10: 4 除频, $f_{DTS}=f_{CK\_INT}/4$ ; 11: 保留。
位 7	PRBEN	0x0	rw	周期缓冲使能 (Period buffer enable) 0: 缓冲关闭; 1: 缓冲开启。
位 6: 5	TWCMSEL	0x0	rw	中央双向对齐计数模式选择 (Two-way count mode selection) 00: 单向对齐计数模式, 方向由 OWCDIR 配置; 01: 中央双向对齐计数模式 1, 上下交替计数, CxIF 位只在计数器向下计数时被置起; 10: 中央双向对齐计数模式 2, 上下交替计数, CxIF 位只在计数器向上计数时被置起; 11: 中央双向对齐计数模式 3, 上下交替计数, CxIF 位在计数器向上和向下计数时皆被置起。
位 4	OWCDIR	0x0	rw	单向对齐计数方向 (One-way count direction) 0: 向上; 1: 向下。

位 3	OCMEN	0x0	rw	单周期使能 (One cycle mode enable) 该功能用于选择溢出事件后，计数器是否停止。 0: 关闭； 1: 开启。
位 2	OVFS	0x0	rw	溢出事件源选择 (Overflow event source) 配置溢出事件或 DMA 请求来源。 0: 来源于计数器溢出、设置 OVFSWTR 位或次定时器控制器产生的溢出事件； 1: 只能来源于计数器溢出。
位 1	OVFEN	0x0	rw	溢出事件使能 (Overflow event enable) 0: 开启； 1: 关闭。
位 0	TMREN	0x0	rw	使能定时器 (TMR enable) 0: 关闭； 1: 开启。

#### 14.2.4.2 控制寄存器2 (TMR3\_CTRL2)

域	简称	复位值	类型	功能
位 15: 8	保留	0x00	resd	保持默认值。
位 7	C1INSEL	0x0	rw	C1IN 选择 (C1IN selection) 0: CH1 管脚连到 C1IRAW 输入； 1: CH1、CH2 和 CH3 管脚异或结果连到 C1IRAW 输入。
位 6: 4	PTOS	0x0	rw	主定时器输出信号选择 (Primary TMR output selection) TMR3 输出到次定时器的信号选择： 000: 复位； 001: 使能； 010: 更新； 011: 比较脉冲； 100: C1ORAW 信号； 101: C2ORAW 信号； 110: C3ORAW 信号； 111: C4ORAW 信号。
位 3	DRS	0x0	rw	DMA 请求源 (DMA request source) DMA 请求来源。 0: 捕获/比较事件； 1: 溢出事件。
位 2: 0	保留	0x0	resd	保持默认值。

#### 14.2.4.3 次定时器控制寄存器 (TMR3\_STCTRL)

域	简称	复位值	类型	功能
位 15	ESP	0x0	rw	外部信号极性 (External signal polarity) 用于选择外部方式。 0: 高电平或上升沿； 1: 低电平或下降沿。
位 14	ECMBEN	0x0	rw	外部时钟模式 B 使能 (External clock mode B enable) 用于启用外部时钟模式 B 0: 关闭； 1: 开启。
位 13: 12	ESDIV	0x0	rw	外部信号除频 (External signal divide) 用于选择降低外部触发频率的除频。 00: 关闭分频； 01: 2 分频； 10: 4 分频； 11: 8 分频。
位 11: 8	ESF	0x0	rw	外部信号滤波 (External signal filter) 用于过滤外部信号，当外部信号产生了 N 次之后才能被采样。 0000: 无滤波器，以 $f_{DTS}$ 采样

				0001: $f_{SAMPLING} = f_{CK\_INT}$ , N=2; 0010: $f_{SAMPLING} = f_{CK\_INT}$ , N=4; 0011: $f_{SAMPLING} = f_{CK\_INT}$ , N=8; 0100: $f_{SAMPLING} = f_{DTS}/2$ , N=6; 0101: $f_{SAMPLING} = f_{DTS}/2$ , N=8; 0110: $f_{SAMPLING} = f_{DTS}/4$ , N=6; 0111: $f_{SAMPLING} = f_{DTS}/4$ , N=8; 1000: $f_{SAMPLING} = f_{DTS}/8$ , N=6; 1001: $f_{SAMPLING} = f_{DTS}/8$ , N=8; 1010: $f_{SAMPLING} = f_{DTS}/16$ , N=5; 1011: $f_{SAMPLING} = f_{DTS}/16$ , N=6; 1100: $f_{SAMPLING} = f_{DTS}/16$ , N=8; 1101: $f_{SAMPLING} = f_{DTS}/32$ , N=5; 1110: $f_{SAMPLING} = f_{DTS}/32$ , N=6; 1111: $f_{SAMPLING} = f_{DTS}/32$ , N=8.
位 7	STS	0x0	rw	次定时器同步 (Subordinate TMR synchronization) 该位开启后，主次定时器可实现高度同步。 0: 关闭; 1: 开启。
位 6: 4	STIS	0x0	rw	次定时器输入选择 (Subordinate TMR input selection) 用于次定时器的输入选择。 000: 内部选择 0 (IS0); 001: 内部选择 1 (IS1); 010: 内部选择 2 (IS2); 011: 内部选择 3 (IS3); 100: C1IRAW 的输入检测器 (C1INC); 101: 滤波输入 1 (C1IF1); 110: 滤波输入 2 (C2IF2); 111: 外部输入 (EXT)。 关于每个定时器中 ISx 的细节，参见表 14-3。
位 3	保留	0x0	resd	保持默认值。
位 2: 0	SMSEL	0x0	rw	次定时器模式选择 (Subordinate TMR mode selection) 000: 关闭从模式; 001: 编码模式 A; 010: 编码模式 B; 011: 编码模式 C; 100: 复位模式 - TRGIN 输入上升沿时，重新初始化计数器; 101: 挂起模式 - TRGIN 输入高电平时，计数器计数; 110: 触发模式 - TRGIN 输入上升沿时，产生触发事件; 111: 外部时钟模式 A - TRGIN 输入上升沿提供时钟; 注：编码模式 A/B/C 配置方法请查看计数模式章节。

#### 14.2.4.4 DMA/中断使能寄存器 (TMR3\_IDEN)

域	简称	复位值	类型	功能
位 15	保留	0x0	resd	保持默认值。
位 14	TDEN	0x0	rw	触发 DMA 请求使能 (Trigger DMA request enable) 0: 关闭; 1: 开启。
位 13	保留	0x0	resd	保持默认值。
位 12	C4DEN	0x0	rw	通道 4 的 DMA 请求使能 (Channel 4 DMA request enable) 0: 关闭; 1: 开启。
位 11	C3DEN	0x0	rw	通道 3 的 DMA 请求使能 (Channel 3 DMA request enable) 0: 关闭; 1: 开启。
位 10	C2DEN	0x0	rw	通道 2 的 DMA 请求使能 (Channel 2 DMA request enable) 0: 关闭; 1: 开启。

位 9	C1DEN	0x0	rw	通道 1 的 DMA 请求使能 (Channel 1 DMA request enable) 0: 关闭; 1: 开启。
位 8	OVFDEN	0x0	rw	溢出事件的 DMA 请求使能 (overflow event DMA request enable) 0: 关闭; 1: 开启。
位 7	保留	0x0	resd	保持默认值。
位 6	TIEN	0x0	rw	触发中断使能 (Trigger interrupt enable) 0: 关闭; 1: 开启。
位 5	保留	0x0	resd	保持默认值。
位 4	C4IEN	0x0	rw	通道 4 中断使能 (Channel 4 interrupt enable) 0: 关闭; 1: 开启。
位 3	C3IEN	0x0	rw	通道 3 中断使能 (Channel 3 interrupt enable) 0: 关闭; 1: 开启。
位 2	C2IEN	0x0	rw	通道 2 中断使能 (Channel 2 interrupt enable) 0: 关闭; 1: 开启。
位 1	C1IEN	0x0	rw	通道 1 中断使能 (Channel 1 interrupt enable) 0: 关闭; 1: 开启。
位 0	OVFIEN	0x0	rw	溢出中断使能 (overflow interrupt enable) 0: 关闭; 1: 开启。

#### 14.2.4.5 中断状态寄存器 (TMR3\_ISTS)

域	简称	复位值	类型	功能
位 15: 13	保留	0x0	resd	保持默认值。
位 12	C4RF	0x0	rw0c	通道 4 再捕获标记 (Channel 4 recapture flag) 见 C1RF 的描述。
位 11	C3RF	0x0	rw0c	通道 3 再捕获标记 (Channel 3 recapture flag) 见 C1RF 的描述。
位 10	C2RF	0x0	rw0c	通道 2 再捕获标记 (Channel 2 recapture flag) 见 C1RF 的描述。
位 9	C1RF	0x0	rw0c	通道 1 再捕获标记 (Channel 1 recapture flag) C1IF 的状态已经为'1'时是否再次发生了捕获, 由硬件置'1', 写'0'清除。 0: 无捕获发生; 1: 捕获发生。
位 8: 7	保留	0x0	resd	保持默认值。
位 6	TRGIF	0x0	rw0c	触发中断标记 (Trigger interrupt flag) 当发生触发事件时由硬件置'1', 写'0'清除。 0: 无触发事件发生; 1: 发生触发事件。
位 5	保留	0x0	resd	保持默认值。
位 4	C4IF	0x0	rw0c	通道 4 中断标记 (Channel 4 interrupt flag) 参考 C1IF 描述。
位 3	C3IF	0x0	rw0c	通道 3 中断标记 (Channel 3 interrupt flag) 参考 C1IF 描述。
位 2	C2IF	0x0	rw0c	通道 2 中断标记 (Channel 2 interrupt flag) 参考 C1IF 描述。
位 1	C1IF	0x0	rw0c	通道 1 中断标记 (Channel 1 interrupt flag) 若通道 1 为输入模式时: 捕获事件发生时由硬件置 '1', 由软件清 '0' 或读 TMR3_C1DT 清'0'。

位 0	OVFIF	0x0	rw0c	0: 无捕获事件发生; 1: 发生捕获事件。 若通道 1 为输出模式时: 比较事件发生时由硬件置'1', 由软件清'0'。 0: 无比较事件发生; 1: 发生比较事件。
				溢出中断标记 (Overflow interrupt flag) 当溢出事件发生时由硬件置'1', 由软件清'0'。 0: 无溢出事件发生; 1: 发生溢出事件, 若 TMR3_CTRL1 的 OVFEN=0、OVFS=0 时: - 当 TMR3_SWEVE 寄存器的 OVFG=1 时产生溢出事件; - 当计数值 CVAL 被触发事件重初始化时产生溢出事件。

#### 14.2.4.6 软件事件寄存器 (TMR3\_SWEVT)

域	简称	复位值	类型	功能
位 15: 7	保留	0x000	resd	保持默认值。
位 6	TRGSWTR	0x0	rw	软件触发触发事件 (Trigger event triggered by software) 通过软件触发一个触发事件。 0: 无作用; 1: 制造一个触发事件。
位 5	保留	0x0	resd	保持默认值。
位 4	C4SWTR	0x0	wo	软件触发通道 4 事件 (Channel 4 event triggered by software) 见 C1M 的描述。
位 3	C3SWTR	0x0	wo	软件触发通道 3 事件 (Channel 3 event triggered by software) 见 C1M 的描述。
位 2	C2SWTR	0x0	wo	软件触发通道 2 事件 (Channel 2 event triggered by software) 见 C1M 的描述。
位 1	C1SWTR	0x0	wo	软件触发通道 1 事件 (Channel 1 event triggered by software) 通过软件触发一个通道 1 事件。 0: 无作用; 1: 制造一个通道 1 事件。
位 0	OVFSWTR	0x0	wo	软件触发溢出事件 (Overflow event triggered by software) 通过软件触发一个溢出事件。 0: 无作用; 1: 制造一个溢出事件。

#### 14.2.4.7 通道模式寄存器1 (TMR3\_CM1)

输出比较模式:

域	简称	复位值	类型	功能
位 15	C2OSEN	0x0	rw	通道 2 输出开关使能 (Channel 2 output switch enable)
位 14: 12	C2OCTRL	0x0	rw	通道 2 输出控制 (Channel 2 output control)
位 11	C2OBEN	0x0	rw	通道 2 输出缓存使能 (Channel 2 output buffer enable)
位 10	C2OIEN	0x0	rw	通道 2 输出立即使能 (Channel 2 output immediately enable)
位 9: 8	C2C	0x0	rw	通道 2 配置 (Channel 2 configure) 当 C2EN='0'时, 这些位用于选择通道 2 为输出或输入, 以及输入时的映射选择: 00: 输出; 01: 输入, C2IN 映射在 C2IFP2 上; 10: 输入, C2IN 映射在 C1IFP2 上; 11: 输入, C2IN 映射在 STCI 上, 只有在 STIS 选择内部触发输入时才工作。
位 7	C1OSEN	0x0	rw	通道 1 输出开关使能 (Channel 1 output switch enable)

				0: EXT 输入不影响 C1ORAW; 1: 当 EXT 输入高电平时, 将 C1ORAW 清 0。
位 6: 4	C1OCTRL	0x0	rw	通道 1 输出控制 (Channel 1 output control) 这些位用于设置原始信号 C1ORAW 的工作状态。 000: 断开。断开 C1ORAW 到 C1OUT 的输出; 001: 设置 C1ORAW 为高: TMR3_CVAL=TMR3_C1DT 时。 010: 设置 C1ORAW 为低: TMR3_CVAL=TMR3_C1DT 时。 011 : 切换 C1ORAW 的电平 : 当 TMR3_CVAL=TMR3_C1DT 时。 100: 固定 C1ORAW 为低。 101: 固定 C1ORAW 为高。 110: PWM 模式 A – OWCDIR=0, 若 TMR3_C1DT>TMR3_CVAL 时设置 C1ORAW 为高, 否则为低; – OWCDIR=1, 若 TMR3_C1DT < TMR3_CVAL 时设置 C1ORAW 为低, 否则为高。 111: PWM 模式 B – OWCDIR=0, 若 TMR3_C1DT > TMR3_CVAL 时设置 C1ORAW 为低, 否则为高; – OWCDIR=1, 若 TMR3_C1DT < TMR3_CVAL 时设置 C1ORAW 为高, 否则为低。 注: 除'000'外, 其余配置下 C1OUT 将连接到 C1ORAW, C1OUT 的输出电平除了会根据 C1ORAW 变化外, 还与 CCTRL 所配置的输出极性有关。
位 3	C1OBEN	0x0	rw	通道 1 输出缓存使能 (Channel 1 output buffer enable) 0: 关闭 TMR3_C1DT 的缓存功能, 写入 TMR3_C1DT 的内容会立即生效。 1: 启用 TMR3_C1DT 的缓存功能, 写入 TMR3_C1DT 的内容将保存到缓存寄存器中, 当发生溢出事件时再更新到 TMR3_C1DT 中。
位 2	C1OIEN	0x0	rw	通道 1 输出立即使能 (Channel 1 output immediately enable) 在 PWM 模式 A 或模式 B 下, 该位能够缩短触发事件到通道 1 的输出响应间的时间。 0: 需要比较 CVAL 与 C1DT 的值之后再产生输出。 1: 无需比较 CVAL 与 C1DT 的值, 当发生触发事件时立即产生输出。
位 1: 0	C1C	0x0	rw	通道 1 配置 (Channel 1 configure) 当 C1EN='0'时, 这些位用于选择通道 1 为输出或输入, 以及输入时的映射选择: 00: 输出; 01: 输入, C1IN 映射在 C1IFP1 上; 10: 输入, C1IN 映射在 C2IFP1 上; 11: 输入, C1IN 映射在 STCI 上, 只有在 STIS 选择内部触发输入时才工作。
<b>输入模式:</b>				
域	简称	复位值	类型	功能
位 15: 12	C2DF	0x0	rw	通道 2 滤波器 (Channel 2 digital filter)
位 11: 10	C2IDIV	0x0	rw	通道 2 分频系数 (Channel 2 input divider)
位 9: 8	C2C	0x0	rw	通道 2 配置 (Channel 2 configure) 当 C2EN='0'时, 这些位用于选择通道 2 为输出或输入, 以及输入时的映射选择: 00: 输出; 01: 输入, C2IN 映射在 C2IFP2 上; 10: 输入, C2IN 映射在 C1IFP2 上; 11: 输入, C2IN 映射在 STCI 上, 只有在 STIS 选择内部触发输入时才工作。
位 7: 4	C1DF	0x0	rw	通道 1 滤波器 (Channel 1 digital filter)

这些位用于配置通道 1 的滤波器。滤波的个数为 N，则表示发生了 N 次采样事件后输入边沿才能通过滤波器：

- 0000: 无滤波器，以  $f_{DTS}$  采样
- 1000: 采样频率  $f_{SAMPLING} = f_{DTS}/8$ , N=6
- 0001: 采样频率  $f_{SAMPLING} = f_{CK\_INT}$ , N=2
- 1001: 采样频率  $f_{SAMPLING} = f_{DTS}/8$ , N=8
- 0010: 采样频率  $f_{SAMPLING} = f_{CK\_INT}$ , N=4
- 1010: 采样频率  $f_{SAMPLING} = f_{DTS}/16$ , N=5
- 0011: 采样频率  $f_{SAMPLING} = f_{CK\_INT}$ , N=8
- 1011: 采样频率  $f_{SAMPLING} = f_{DTS}/16$ , N=6
- 0100: 采样频率  $f_{SAMPLING} = f_{DTS}/2$ , N=6
- 1100: 采样频率  $f_{SAMPLING} = f_{DTS}/16$ , N=8
- 0101: 采样频率  $f_{SAMPLING} = f_{DTS}/2$ , N=8
- 1101: 采样频率  $f_{SAMPLING} = f_{DTS}/32$ , N=5
- 0110: 采样频率  $f_{SAMPLING} = f_{DTS}/4$ , N=6
- 1110: 采样频率  $f_{SAMPLING} = f_{DTS}/32$ , N=6
- 0111: 采样频率  $f_{SAMPLING} = f_{DTS}/4$ , N=8
- 1111: 采样频率  $f_{SAMPLING} = f_{DTS}/32$ , N=8

通道 1 分频系数 (Channel 1 input divider)

这些位定义了通道 1 的分频系数。

00: 不分频，每一个有效的边沿都会产生一次输入；

位 3: 2 C1IDIV 0x0

rw

01: 每 2 个有效的边沿产生一次输入；

10: 每 4 个有效的边沿产生一次输入；

11: 每 8 个有效的边沿产生一次输入。

注：C1EN='0'时，分频系数复位。

通道 1 配置 (Channel 1 configure)

当 C1EN='0'时，这些位用于选择通道 1 为输出或输入，以及输入时的映射选择：

位 1: 0 C1C 0x0

rw

00: 输出；

01: 输入，C1IN 映射在 C1IFP1 上；

10: 输入，C1IN 映射在 C2IFP1 上；

11: 输入，C1IN 映射在 STCI 上，只有在 STIS 选择内部触发输入时才工作。

#### 14.2.4.8 通道模式寄存器2 (TMR3\_CM2)

输出比较模式：

域	简称	复位值	类型	功能
位 15	C4OSEN	0x0	rw	通道 4 输出开关使能 (Channel 4 output switch enable)
位 14: 12	C4OCTRL	0x0	rw	通道 4 输出控制 (Channel 4 output control)
位 11	C4OBEN	0x0	rw	通道 4 输出缓存使能 (Channel 4 output buffer enable)
位 10	C4OIEN	0x0	rw	通道 4 输出立即使能 (Channel 4 output immediately enable)

通道 4 配置 (Channel 4 configure)

当 C4EN='0'时，这些位用于选择通道 4 为输出或输入，以及输入时的映射选择：

位 9: 8 C4C 0x0

rw

00: 输出；

01: 输入，C4IN 映射在 C4IFP4 上；

10: 输入，C4IN 映射在 C3IFP4 上；

11: 输入，C4IN 映射在 STCI 上，只有在 STIS 选择内部触发输入时才工作。

位 7 C3OSEN 0x0

rw

通道 3 输出开关使能 (Channel 3 output switch enable)

位 6: 4 C3OCTRL 0x0

rw

通道 3 输出控制 (Channel 3 output control)

位 3 C3OBEN 0x0

rw

通道 3 输出缓存使能 (Channel 3 output buffer enable)

位 2 C3OIEN 0x0

rw

通道 3 输出立即使能 (Channel 3 output immediately enable)

通道 3 配置 (Channel 3 configure)

位 1: 0 C3C 0x0

rw

当 C3EN='0'时，这些位用于选择通道 3 为输出或输入，以及输入时的映射选择：

00: 输出；

01: 输入，C3IN 映射在 C3IFP3 上；

10: 输入, C3IN 映射在 C4IFP3 上;  
11: 输入, C3IN 映射在 STCI 上, 只有在 STIS 选择内部触发输入时才工作。

**输入模式:**

域	简称	复位值	类型	功能
位 15: 12	C4DF	0x0	rw	通道 4 滤波器 (Channel 4 digital filter)
位 11: 10	C4IDIV	0x0	rw	通道 4 分频系数 (Channel 4 input divider)
位 9: 8	C4C	0x0	rw	通道 4 配置 (Channel 4 configure) 当 C4EN=0'时, 这些位用于选择通道 4 为输出或输入, 以及输入时的映射选择: 00: 输出; 01: 输入, C4IN 映射在 C4IFP4 上; 10: 输入, C4IN 映射在 C3IFP4 上; 11: 输入, C4IN 映射在 STCI 上, 只有在 STIS 选择内部触发输入时才工作。
位 7: 4	C3DF	0x0	rw	通道 3 滤波器 (Channel 3 digital filter)
位 3: 2	C3IDIV	0x0	rw	通道 3 分频系数 (Channel 3 input divider)
位 1: 0	C3C	0x0	rw	通道 3 配置 (Channel 3 configure) 当 C3EN=0'时, 这些位用于选择通道 3 为输出或输入, 以及输入时的映射选择: 00: 输出; 01: 输入, C3IN 映射在 C3IFP3 上; 10: 输入, C3IN 映射在 C4IFP3 上; 11: 输入, C3IN 映射在 STCI 上, 只有在 STIS 选择内部触发输入时才工作。

**14.2.4.9 通道控制寄存器 (TMR3\_CCTRL)**

域	简称	复位值	类型	功能
位 15: 14	保留	0x0	resd	保持默认值。
位 13	C4P	0x0	rw	通道 4 极性 (Channel 4 polarity) 见 C1P 的描述。
位 12	C4EN	0x0	rw	通道 4 使能 (Channel 4 enable) 见 C1EN 的描述。
位 11	C3CP	0x0	rw	通道 3 互补极性 (Channel 3 complementary polarity) 定义输入信号的有效沿, 详见 C1P 位描述。
位 10	保留	0x0	resd	保持默认值。
位 9	C3P	0x0	rw	通道 3 极性 (Channel 3 polarity) 见 C1P 的描述。
位 8	C3EN	0x0	rw	通道 3 使能 (Channel 3 enable) 见 C1EN 的描述。
位 7	C2CP	0x0	rw	通道 2 互补极性 (Channel 2 complementary polarity) 定义输入信号的有效沿, 详见 C1P 位描述。
位 6	保留	0x0	resd	保持默认值。
位 5	C2P	0x0	rw	通道 2 极性 (Channel 2 polarity) 见 C1P 的描述。
位 4	C2EN	0x0	rw	通道 2 使能 (Channel 2 enable) 见 C1EN 的描述。
位 3	C1CP	0x0	rw	通道 1 互补极性 (Channel 1 complementary polarity) 定义输入信号的有效沿, 详见 C1P 位描述。
位 2	保留	0x0	resd	保持默认值。
位 1	C1P	0x0	rw	通道 1 极性 (Channel 1 polarity) 通道 1 配置为输出: 0: C1OUT 的有效电平为高 1: C1OUT 的有效电平为低 通道 1 配置为输入: C1CP/C1P 位共同定义输入信号有效沿。 00: C1IN 的有效边沿为上升沿; 作为外部触发使用时, C1IN 不反相。

位 0	C1EN	0x0	rw	01: C1IN 的有效边沿为下降沿; 作为外部触发使用时, C1IN 反相。 10: 保留 11: C1IN 的有效边沿为上升沿和下降沿; 作为外部触发使用时, C1IN 不反相。
				通道 1 使能 (Channel 1 enable) 0: 禁止输入或输出; 1: 使能输入或输出。

表 14-6 标准 CxOUT 通道的输出控制位

CxEN 位	CxOUT 输出状态
0	禁止输出 (CxOUT=0, Cx_EN=0)
1	CxOUT = CxORAW + 极性, Cx_EN=1

注意: 连接到标准 CxOUT 通道的外部 I/O 管脚状态, 取决于 CxOUT 通道状态和 GPIO 以及 IOMUX 寄存器。

#### 14.2.4.10 计数值 (TMR3\_CVAL)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0000	resd	保持默认值
位 15: 0	CVAL	0x0000	rw	计数值 (Counter value)

#### 14.2.4.11 分频系数 (TMR3\_DIV)

域	简称	复位值	类型	功能
位 15: 0	DIV	0x0000	rw	分频系数 (Divider value) 计数器时钟频率 $f_{CK\_CNT} = f_{TMR\_CLK} / (DIV[15: 0]+1)$ 。 DIV 为溢出事件发生时写入的分频系数。

#### 14.2.4.12 周期寄存器 (TMR3\_PR)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0000	resd	保持默认值
位 15: 0	PR	0x0000	rw	周期值 (Period value) 定时器计数的周期值。当周期值为 0 时, 定时器不工作。

#### 14.2.4.13 通道1数据寄存器 (TMR3\_C1DT)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0000	resd	保持默认值
位 15: 0	C1DT	0x0000	rw	通道 1 数据寄存器值 (Channel 1 data register) 若通道 1 配置为输入: C1DT 是前一次通道 1 输入事件 (C1IN) 所保存的 CVAL。 若通道 1 配置为输出: C1DT 是将要和 CVAL 进行比较的值, 写入的值是否会立即生效取决于输出缓存使能位 (C1OBEN), 并根据设置在 C1OUT 上产生相应的输出。

#### 14.2.4.14 通道2数据寄存器 (TMR3\_C2DT)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0000	resd	保持默认值
位 15: 0	C2DT	0x0000	rw	通道 2 数据寄存器值 (Channel 2 data register) 若通道 2 配置为输入: C2DT 是前一次通道 2 输入事件 (C2IN) 所保存的 CVAL。 若通道 2 配置为输出: C2DT 是将要和 CVAL 进行比较的值, 写入的值是否会立即生效取决于输出缓存使能位 (C2OBEN), 并根据设置在 C2OUT 上产生相应的输出。

#### 14.2.4.15 通道3数据寄存器 (TMR3\_C3DT)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0000	resd	保持默认值
位 15: 0	C3DT	0x0000	rw	通道 3 数据寄存器值 (Channel 3 data register) 若通道 3 配置为输入： C3DT 是前一次通道 3 输入事件 (C3IN) 所保存的 CVAL。 若通道 3 配置为输出： C3DT 是将要和 CVAL 进行比较的值，写入的值是否会立即生效取决于输出缓存使能位 (C3OBEN)，并根据设置在 C3OUT 上产生相应的输出。

#### 14.2.4.16 通道4数据寄存器 (TMR3\_C4DT)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0000	resd	保持默认值
位 15: 0	C4DT	0x0000	rw	通道 4 数据寄存器值 (Channel 4 data register) 若通道 4 配置为输入： C4DT 是前一次通道 4 输入事件 (C4IN) 所保存的 CVAL。 若通道 4 配置为输出： C4DT 是将要和 CVAL 进行比较的值，写入的值是否会立即生效取决于输出缓存使能位 (C4OBEN)，并根据设置在 C4OUT 上产生相应的输出。

#### 14.2.4.17 DMA控制寄存器 (TMR3\_DMACTRL)

域	简称	复位值	类型	功能
位 15: 13	保留	0x0	resd	保持默认值。
位 12: 8	DTB	0x00	rw	DMA 传输字节 (DMA transfer bytes) 这些位定义了传输的字节个数： 00000: 1 个字节      00001: 2 个字节 00010: 3 个字节      00011: 4 个字节 ..... 10000: 17 个字节      10001: 18 个字节
位 7: 5	保留	0x0	resd	保持默认值。
位 4: 0	ADDR	0x00	rw	DMA 传输地址偏移 (DMA transfer address offset) ADDR 定义了从 TMR3_CTRL1 所在地址开始的偏移量： 00000: TMR3_CTRL1, 00001: TMR3_CTRL2, 00010: TMR3_STCTRL, .....

#### 14.2.4.18 DMA数据寄存器 (TMR3\_DMADT)

域	简称	复位值	类型	功能
位 15: 0	DMADT	0x0000	rw	DMA 传输的数据寄存器 (DMA data register) 通过对 DMADT 寄存器的读写能够实现对任意 TMR 寄存器的操作，其操作的寄存器地址范围是： TMR3 外设地址 + ADDR*4 至 TMR3 外设地址 + ADDR*4 + DTB*4。

## 14.3 通用定时器 (TMR14)

### 14.3.1 TMR14简介

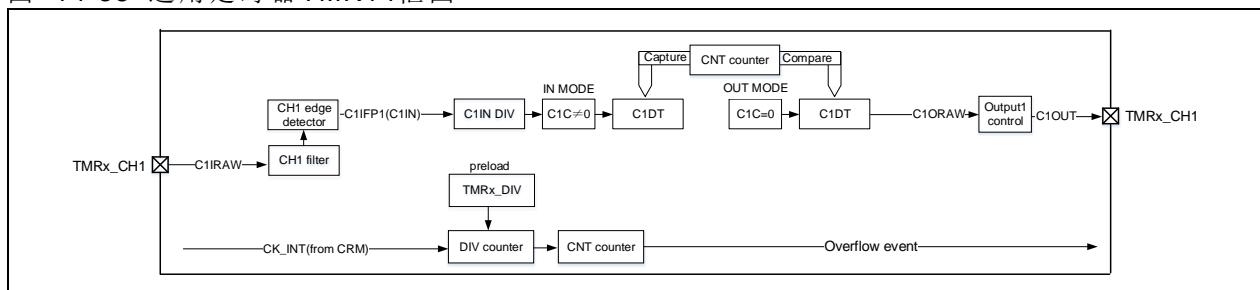
通用定时器 TMR14 支持 16 位向上计数，可通过同步功能进行互联。

### 14.3.2 TMR14主要特性

通用 TMR14 定时器功能包括：

- 由内部用作计数时钟
- 16 位向上计数器
- 1 组独立通道，支持输入捕获、输出比较、PWM 生成
- 定时器之间可互联同步
- 支持溢出事件、通道事件触发中断

图 14-38 通用定时器 TMR14框图

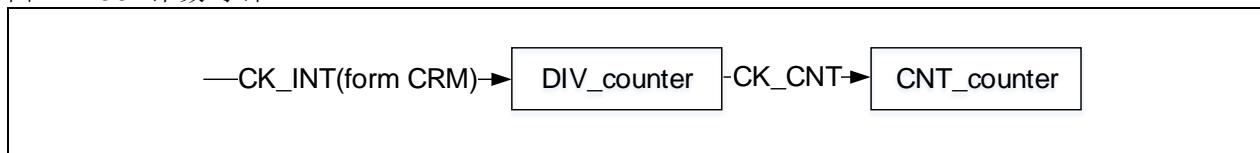


### 14.3.3 TMR14功能描述

#### 14.3.3.1 计数时钟

通用定时器 TMR14 计数时钟可从内部时钟 (CK\_INT) 时钟源提供。

图 14-39 计数时钟



#### 内部时钟 (CK\_INT)

默认下使用 CK\_INT 经由预分频器驱动计数器计数，当 TMR 对应的 APB 时钟预分频系数是 1 时，CK\_INT 频率等于 APB 时钟频率，否则 CK\_INT 频率等于 APB 时钟频率的 2 倍。相关配置流程如下：

- 配置 TMRx\_DIV 寄存器，设置计数器计数频率。
- 配置 TMRx\_PR 寄存器，设置计数器计数周期。
- 配置 TMRx\_CTRL1 寄存器 TMREN，使能计数器。

图 14-40 使用 CK\_INT 计数，TMRx\_DIV=0x0，周期寄存器 TMRx\_PR=0x16

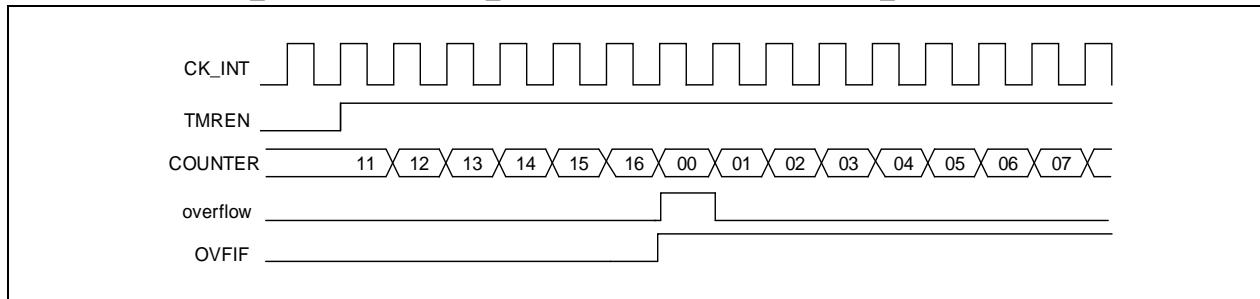
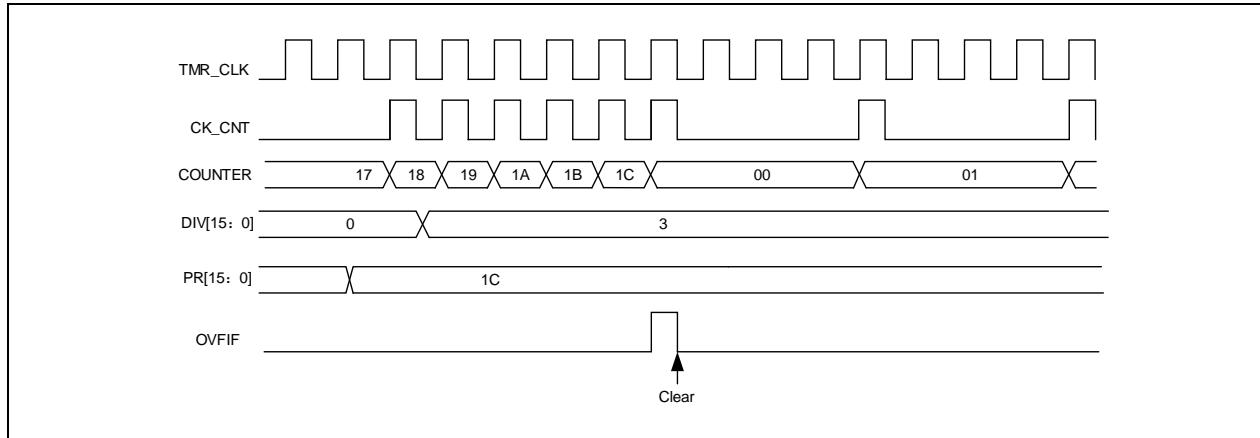


图 14-41 当预分频器的参数从1变到4时，计数器的时序图



### 14.3.3.2 计数模式

通用定时器 TMR14 仅提供向上计数模式，其内部拥有一个支持 16 位计数的计数器。

TMRx\_PR 寄存器用于设置计数器计数周期。默认 TMRx\_PR 寄存器值会立即传入它的影子寄存器；当开启周期缓冲功能后（PRBEN 置 1），TMRx\_PR 寄存器值在溢出事件发生时传入它的影子寄存器。

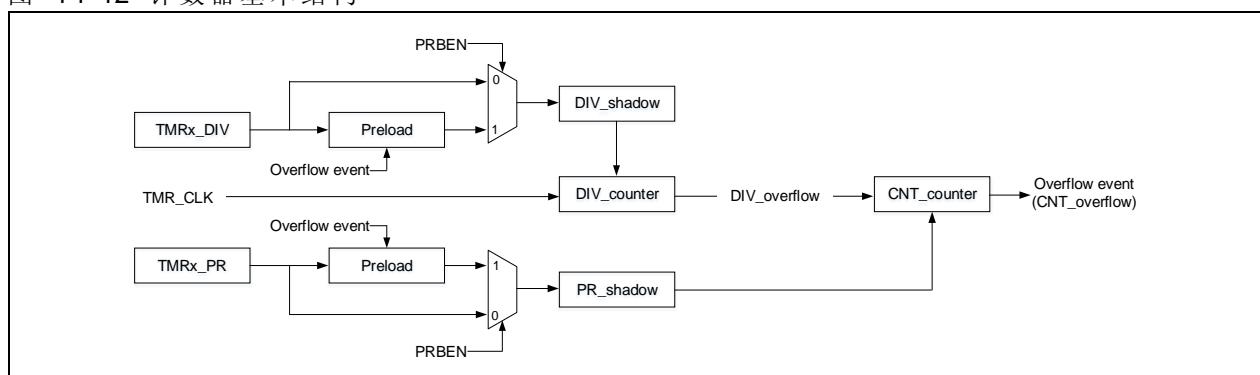
TMRx\_DIV 寄存器用于设置计数器计数频率，每（DIV[15:0]+1）个计数时钟周期，计数器计数一次。和 TMRx\_PR 寄存器类似，开启周期缓冲功能后，TMRx\_DIV 寄存器值在溢出事件时更新至它的影子寄存器。

读取 TMRx\_CNT 寄存器会返回当前计数器计数值，写入 TMRx\_CNT 寄存器会更新计数器当前计数值为写入值。

默认允许产生溢出事件，设置 TMRx\_CTRL1 寄存器 OVFEN=1 将禁止更新事件产生。TMRx\_CTRL1 寄存器 OVFS 用于选择溢出事件来源，默认计数器上溢或下溢、置位 OVFSWTR、复位模式次定时器控制器产生的复位信号产生溢出事件。置位 OVFS 后，只有计数器上溢或下溢产生溢出事件。

TMREN 位置 1 将使能定时器计数，由于同步逻辑，实际驱动计数器的使能信号 TMR\_EN 相对于 TMREN 延迟一个时钟周期。

图 14-42 计数器基本结构



#### 向上计数模式

配置 TMRx\_CTRL1 寄存器 CMSEL[1:0]=2'b00，OWCDIR=1'b0 开启向上计数模式，计数值达到 TMRx\_PR 值时，重新从 0 向上计数，计数器上溢并产生溢出事件，同时 OVFIF 位置 1。若禁止产生溢出事件，计数器溢出后不再重载预分频值和周期值，否则预分频值和周期值在溢出事件后更新。

图 14-43 PRBEN=0 时的溢出事件

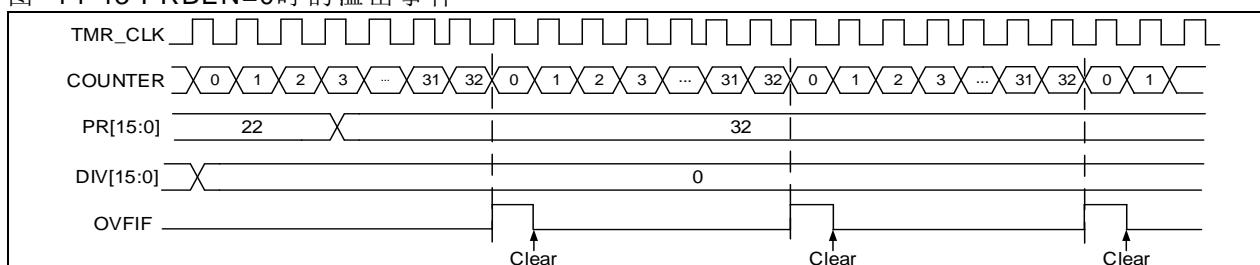
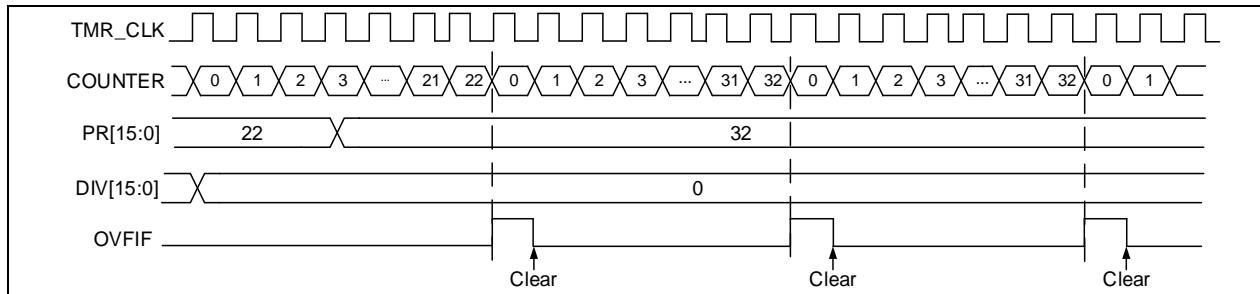


图 14-44 PRBEN=1时的溢出事件



### 14.3.3.3 TMR输入部分

TMR14 拥有一个独立通道，可配置为输入或输出，当配置位输入时，每个通道输入信号依次经过以下处理：

- TMRx\_CHx 经过预处理输出 CxIRAW。配置 C1INSEL 位，选择 CxIRAW 来源是 TMRx\_CHx。
- CxIRAW 输入数字滤波器，输出滤波后信号 CxIF。数字滤波器通过 CxDL 位配置采样频率和次数。
- CxIF 输入边沿检测器，输出边沿选择后信号 CxIFP<sub>x</sub>。边沿选择由 CxP 和 CxCP 位共同控制，可选择输入上升沿、下降沿或双边沿有效。
- CxIFP<sub>x</sub> 输入捕获信号选择器，输出选择后信号 CxIN。捕获信号选择器由 CxC 控制，可选择 CxIN 来源为 CxIFPx。其中 CxIFPx ( $x \neq y$ ) 是来自通道 y 的 CxIFPy 经通道 x 边沿检测器处理后的信号（例如 C1IFP2 是来自通道 1 的 C1IFP1 信号经过通道 2 边沿检测器处理后的信号）。
- CxIN 经由输入通道分频器，输出分频后信号 CxIPS。分频系数由 CxIDIV 位配置为不分频、2 分频、4 分频或 8 分频。

图 14-45 输入/输出通道 1 的主电路

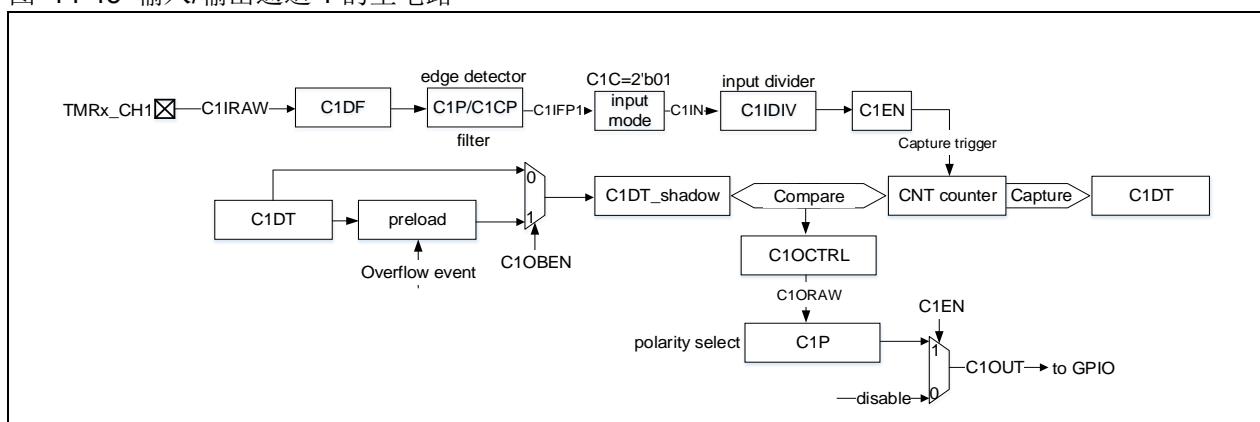
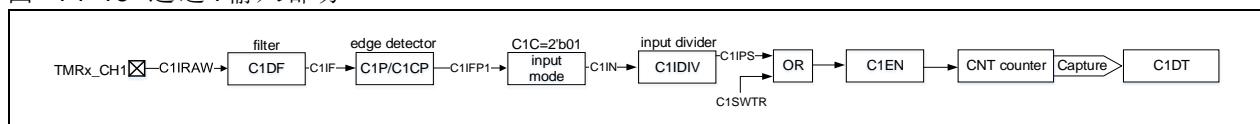


图 14-46 通道 1 输入部分



#### 输入模式

此模式下，当选中的触发信号被检测到，通道寄存器（TMRx\_CxDT）记录当前计数器计数值，并将捕获比较中断标志位（CxIF）置 1，若已使能通道中断（CxIEN）、通道 DMA 请求（CxDEN）则产生相应的中断和 DMA 请求。若在 CxIF 置 1 后检测到触发信号，将产生捕获溢出事件，TMRx\_CxDT 会使用当前计数器计数值覆盖之前记录的计数器计数值，同时通道再捕获标志位（CxRF）置 1。

若要捕获 C1IN 输入的上升沿，可按如下进行配置：

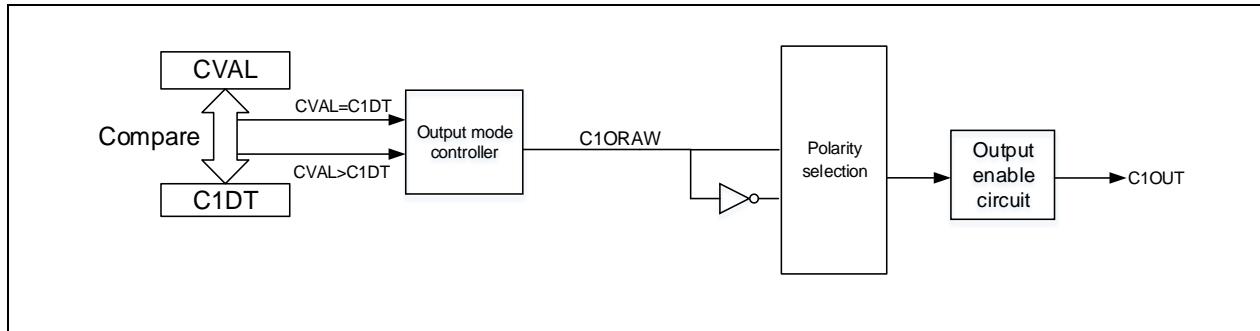
- 将通道模式寄存器 1（TMR14\_CM1）中的 C1C 位配置为 01，选择 C1IN 作为通道 1 输入。
- 配置 C1IN 信号滤波器带宽（CxDF[3: 0]）。
- 配置 C1IN 通道的有效沿，在通道控制寄存器（TMR14\_CCTRL）中写入 C1P=0（上升沿）。

- 配置 C1IN 信号捕获分频 (C1DIV[1: 0])。
- 使能通道 1 输入捕获 (C1EN=1)。
- 根据需要设置中断使能寄存器 (TMR14\_IDEN) 中的 C1IEN 位，选择中断请求。

#### 14.3.3.4 TMR输出部分

TMR 的输出部分由比较器和输出控制构成，用于编程输出信号的周期、占空比、极性。

图 14-47 捕获/比较通道的输出部分（通道 1）



#### 输出模式

配置  $CxC[1: 0] \neq 2'b00$  将通道配置为输出可实现多种输出模式，此时，计数器计数值将与通道寄存器 (TMR14\_CxDT) 值比较，并根据  $CxOCTRL[2: 0]$  位配置的输出模式，产生中间信号  $CxORAW$ ，再经过输出控制逻辑处理后输送到 IO。输出信号的周期由周期寄存器 (TMR14\_PR) 值配置，占空比则由通道寄存器 (TMR14\_CxDT) 值配置。

输出比较模式有以下子类：

**PWM 模式 A:**  $CxOCTRL=3'b110$  时，开启 PWM 模式 A。向上计数时， $TMRx_C1DT > TMRx_CVAL$  时  $C1ORAW$  输出高电平，否则为低电平；向下计数时， $TMRx_C1DT < TMRx_CVAL$  时  $C1ORAW$  输出低电平，否则为高电平。若要使用 PWM 模式 A，可按如下方式配置。

- 配置 TMRx\_PR 寄存器，设置 PWM 周期。
- 配置 TMRx\_CxDT 寄存器，设置 PWM 占空比。
- 配置 TMRx\_CM1/CM2 寄存器  $CxOCTRL$  位为 3'b110，设置输出模式为 PWM 模式 A。
- 配置 TMRx\_DIV 寄存器，设置计数器计数频率。
- 配置 TMRx\_CTRL1 寄存器  $TWCMSEL[1:0]$  位，设置计数器计数模式。
- 配置 TMRx\_CCTRL 寄存器  $CxP$  位、 $CxCP$  位，设置输出极性。
- 配置 TMRx\_CCTRL 寄存器  $CxEN$  位、 $CxCEN$  位，使能通道输出。
- 配置 TMRx\_BRK 寄存器  $OEN$  位，使能 TMRx 输出。
- 配置 TMR 输出通道对应 GPIO 为对应的复用模式。
- 配置 TMRx\_CTRL1 寄存器  $TMREN$  位，使能 TMRx 计数。

**PWM 模式 B:**  $CxOCTRL=3'b111$  时，开启 PWM 模式 B。向上计数时， $TMRx_C1DT > TMRx_CVAL$  时  $C1ORAW$  输出低电平，否则为高电平；向下计数时， $TMRx_C1DT < TMRx_CVAL$  时  $C1ORAW$  输出高电平，否则为低电平。

**强制输出模式:**  $CxOCTRL=3'b100/101$  时，开启强制输出模式。此时， $CxORAW$  信号的电平被强制输出为配置的电平，而与计数值无关。虽然输出信号不依赖于比较结果，但通道标志位和 DMA 请求仍依赖于比较结果。

**输出比较模式:**  $CxOCTRL=3'b001/010/011$  时，开启输出比较模式。此时，当计数值与  $CxDT$  值匹配时， $CxORAW$  被强制为高电平、低电平或进行电平翻转。

图 14-48 展示了输出比较模式（翻转）的例子， $C1DT=0x3$ ，当计数值等于 0x3 时，输出电平 C1OUT 被翻转。

图 14-49 展示了计数器向上计数与 PWM 模式 A 配合的例子， $PR=0x32$ ， $CxDT$  配置为不同的值时输出时输出信号的翻转情况。

图 14-48 计数值与 C1DT 值匹配时翻转 C1ORAW

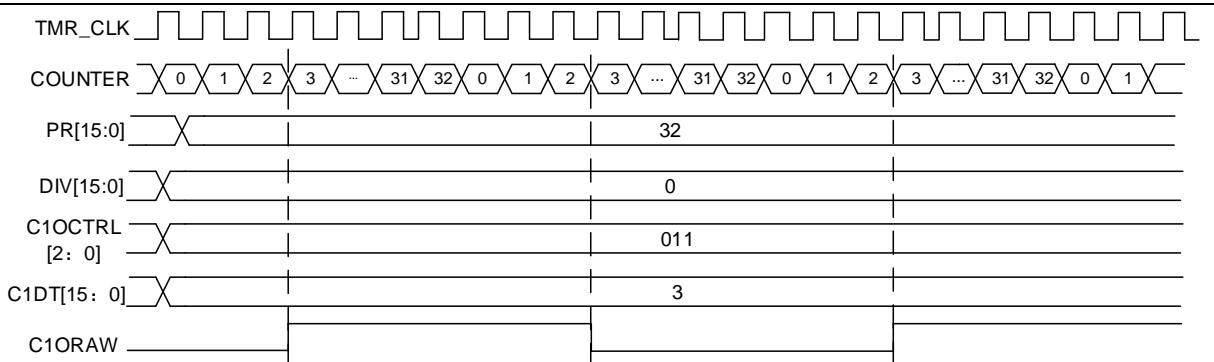
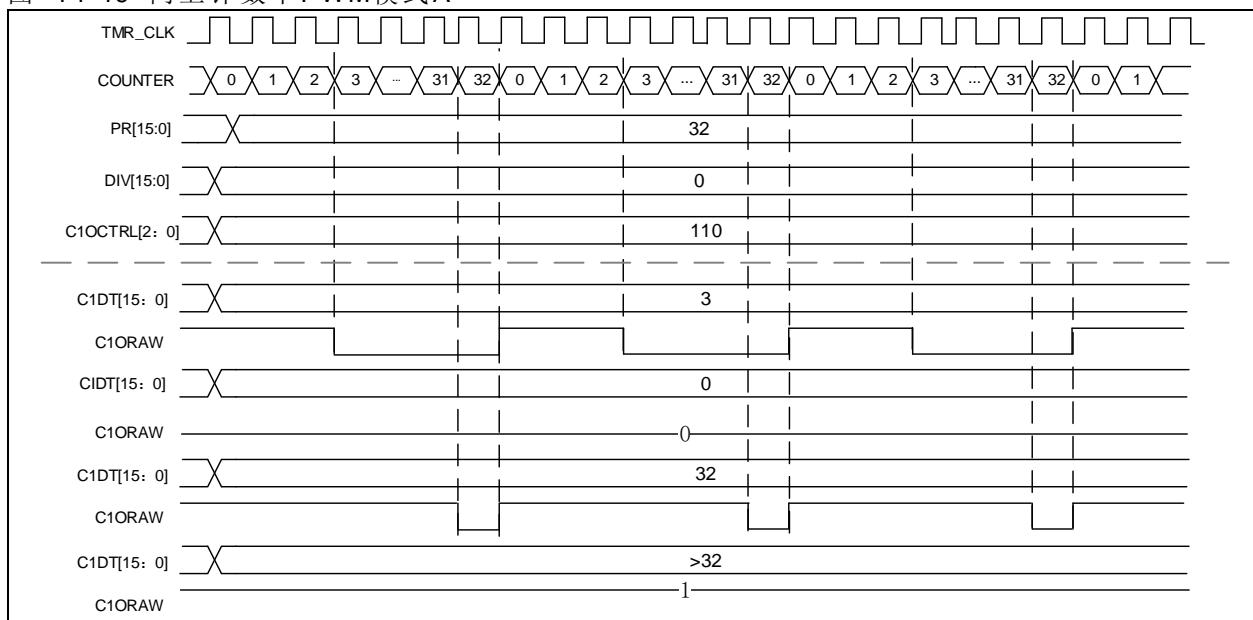


图 14-49 向上计数下 PWM 模式 A



### 14.3.3.5 调试模式

当微控制器进入调试模式 (Cortex®-M4 核心停止) 时, 将 DEBUG 模块中的 TMR14\_PAUSE 置 1, 可以使 TMR14 计数器暂停计数。

### 14.3.4 TMR14 寄存器描述

必须以字 (32 位) 的方式操作这些外设寄存器。

下表中将 TMR14 的所有寄存器映射到一个 16 位可寻址 (编址) 空间。

表 14-7 TMR14 寄存器映像和复位值

寄存器简称	基址偏移量	复位值
TMR14_CTRL1	0x00	0x0000
TMR14_IDEN	0x0C	0x0000
TMR14ISTS	0x10	0x0000
TMR14_SWEVT	0x14	0x0000
TMR14_CM1	0x18	0x0000
TMR14_CCTRL	0x20	0x0000
TMR14_CVAL	0x24	0x0000
TMR14_DIV	0x28	0x0000
TMR14_PR	0x2C	0x0000
TMR14_C1DT	0x34	0x0000

TMR14_RMP	0x50	0x0000
-----------	------	--------

#### 14.3.4.1 控制寄存器1 (TMR14\_CTRL1)

域	简称	复位值	类型	功能
位 15: 10	保留	0x00	resd	保持默认值。
位 9: 8	CLKDIV	0x0	rw	时钟除频 (Clock divider) 此位用于设置数字滤波器采样频率 $f_{DTS}$ 和定时器时钟频率 $f_{CK\_INT}$ 之间的分频比。 00: 无除频, $f_{DTS}=f_{CK\_INT}$ ; 01: 2 除频, $f_{DTS}=f_{CK\_INT}/2$ ; 10: 4 除频, $f_{DTS}=f_{CK\_INT}/4$ ; 11: 保留。
位 7	PRBEN	0x0	rw	周期缓冲使能 (Period buffer enable) 0: 缓冲关闭; 1: 缓冲开启。
位 6: 4	保留	0x0	resd	保持默认值。
位 3	OCMEN	0x0	rw	单周期使能 (One cycle mode enable) 该功能用于选择溢出事件后, 计数器是否停止。 0: 关闭; 1: 开启。
位 2	OVFS	0x0	rw	溢出事件源选择 (Overflow event source) 配置溢出事件或 DMA 请求来源。 0: 来源于计数器溢出、设置 OVFSWTR 位或次定时器控制器产生的溢出事件; 1: 只能来源于计数器溢出。
位 1	OVFEN	0x0	rw	溢出事件使能 (Overflow event enable) 0: 开启; 1: 关闭。
位 0	TMREN	0x0	rw	使能定时器 (TMR enable) 0: 关闭; 1: 开启。

#### 14.3.4.2 中断使能寄存器 (TMR14\_IDEN)

域	简称	复位值	类型	功能
位 15: 2	保留	0x0000	resd	保持默认值。
位 1	C1IEN	0x0	rw	通道 1 中断使能 (Channel 1 interrupt enable) 0: 关闭; 1: 开启。
位 0	OVFIEN	0x0	rw	溢出中断使能 (overflow interrupt enable) 0: 关闭; 1: 开启。

#### 14.3.4.3 中断状态寄存器 (TMR14ISTS)

域	简称	复位值	类型	功能
位 15: 10	保留	0x00	resd	保持默认值。
位 9	C1RF	0x0	rw0c	通道 1 再捕获标记 (Channel 1 recapture flag) C1IF 的状态已经为'1'时是否再次发生了捕获, 由硬件置'1', 写'0'清除。 0: 无捕获发生; 1: 捕获发生。
位 8: 2	保留	0x00	resd	保持默认值。
位 1	C1IF	0x0	rw0c	通道 1 中断标记 (Channel 1 interrupt flag) 若通道 1 为输入模式时: 捕获事件发生时由硬件置'1', 由软件清'0'或读 TMR14_C1DT 清'0'。 0: 无捕获事件发生; 1: 发生捕获事件。 若通道 1 为输出模式时: 比较事件发生时由硬件置'1', 由软件清'0'。

位 0	OVFIF	0x0	rw0c	0: 无比较事件发生; 1: 发生比较事件。 溢出中断标记 (Overflow interrupt flag) 当溢出事件发生时由硬件置'1', 由软件清'0'。 0: 无溢出事件发生; 1: 发生溢出事件, 若 TMR14_CTRL1 的 OVFEN=0、 OVFS=0 时: - 当 TMR14_SWEVE 寄存器的 OVFG=1 时产生溢出事 件; - 当计数值 CVAL 被触发事件重初始化时产生溢出事 件。
-----	-------	-----	------	--

#### 14.3.4.4 软件事件寄存器 (TMR14\_SWEVT)

域	简称	复位值	类型	功能
位 15: 2	保留	0x0000	resd	保持默认值。 软件触发通道 1 事件 (Channel 1 event triggered by software)
位 1	C1SWTR	0x0	wo	通过软件触发一个通道 1 事件。 0: 无作用; 1: 制造一个通道 1 事件。
位 0	OVFSWTR	0x0	wo	软件触发溢出事件 (Overflow event triggered by software) 通过软件触发一个溢出事件。 0: 无作用; 1: 制造一个溢出事件。

#### 14.3.4.5 通道模式寄存器1 (TMR14\_CM1)

通道可用于输入（捕获模式）或输出（比较模式），通道的方向由相应的 CxC 定义。该寄存器其它位的作用在输入和输出模式下不同。CxOx 描述了通道在输出模式下的功能，CxIx 描述了通道在输出模式下的功能。因此必须注意，同一个位在输出模式和输入模式下的功能是不同的。

##### 输出比较模式:

域	简称	复位值		功能
位 15: 7	保留	0x000	resd	保持默认值。 通道 1 输出控制 (Channel 1 output control) 这些位用于设置原始信号 C1ORAW 的工作状态。 000: 断开。断开 C1ORAW 到 C1OUT 的输出; 001: 设置 C1ORAW 为高: TMR14_CVAL=TMR14_C1DT 时。 010: 设置 C1ORAW 为低: TMR14_CVAL=TMR14_C1DT 时。 011: 切换 C1ORAW 的电平: 当 TMR14_CVAL=TMR14_C1DT 时。 100: 固定 C1ORAW 为低。 101: 固定 C1ORAW 为高。 110: PWM 模式 A - OWCDIR=0, 若 TMR14_C1DT>TMR14_CVAL 时设置 C1ORAW 为高, 否则为低; - OWCDIR=1, 若 TMR14_C1DT<TMR14_CVAL 时设置 C1ORAW 为低, 否则为高。 111: PWM 模式 B - OWCDIR=0, 若 TMR14_C1DT>TMR14_CVAL 时设置 C1ORAW 为低, 否则为高; - OWCDIR=1, 若 TMR14_C1DT<TMR14_CVAL 时设置 C1ORAW 为高, 否则为低。 注: 除'000'外, 其余配置下 C1OUT 将连接到 C1ORAW, C1OUT 的输出电平除了会根据 C1ORAW 变化外, 还与 CCTRL 所配置的输出极性有关。
位 6: 4	C1OCTRL	0x0	rw	通道 1 输出缓存使能 (Channel 1 output buffer enable)
位 3	C1OBEN	0x0	rw	

				0: 关闭 TMR14_C1DT 的缓存功能，写入 TMR14_C1DT 的内容会立即生效。 1: 启用 TMR14_C1DT 的缓存功能，写入 TMR14_C1DT 的内容将保存到缓存寄存器中，当发生溢出事件时再更新到 TMR14_C1DT 中。
位 2	C1OIEN	0x0	rw	通道 1 输出立即使能 (Channel 1 output immediately enable) 在 PWM 模式 A 或模式 B 下，该位能够缩短触发事件到通道 1 的输出响应间的时间。 0: 需要比较 CVAL 与 C1DT 的值之后再产生输出。 1: 无需比较 CVAL 与 C1DT 的值，当发生触发事件时立即产生输出。
位 1: 0	C1C	0x0	rw	通道 1 配置 (Channel 1 configure) 当 C1EN='0'时，这些位用于选择通道 1 为输出或输入，以及输入时的映射选择： 00: 输出； 01: 输入，C1IN 映射在 C1IFP1 上； 10: 保留； 11: 保留。

## 输入模式：

域	简称	复位值	类型	功能
位 15: 8	保留	0x00	resd	保持默认值
位 7: 4	C1DF	0x0	rw	通道 1 滤波器 (Channel 1 digital filter) 这些位用于配置通道 1 的滤波器。滤波的个数为 N，则表示发生了 N 次采样事件后输入边沿才能通过滤波器： 0000: 无滤波器，以 $f_{DTS}$ 采样 1000: 采样频率 $f_{SAMPLING} = f_{DTS}/8$ , N=6 0001: 采样频率 $f_{SAMPLING} = f_{CK\_INT}$ , N=2 1001: 采样频率 $f_{SAMPLING} = f_{DTS}/8$ , N=8 0010: 采样频率 $f_{SAMPLING} = f_{CK\_INT}$ , N=4 1010: 采样频率 $f_{SAMPLING} = f_{DTS}/16$ , N=5 0011: 采样频率 $f_{SAMPLING} = f_{CK\_INT}$ , N=8 1011: 采样频率 $f_{SAMPLING} = f_{DTS}/16$ , N=6 0100: 采样频率 $f_{SAMPLING} = f_{DTS}/2$ , N=6 1100: 采样频率 $f_{SAMPLING} = f_{DTS}/16$ , N=8 0101: 采样频率 $f_{SAMPLING} = f_{DTS}/2$ , N=8 1101: 采样频率 $f_{SAMPLING} = f_{DTS}/32$ , N=5 0110: 采样频率 $f_{SAMPLING} = f_{DTS}/4$ , N=6 1110: 采样频率 $f_{SAMPLING} = f_{DTS}/32$ , N=6 0111: 采样频率 $f_{SAMPLING} = f_{DTS}/4$ , N=8 1111: 采样频率 $f_{SAMPLING} = f_{DTS}/32$ , N=8
位 3: 2	C1IDIV	0x0	rw	通道 1 分频系数 (Channel 1 input divider) 这些位定义了通道 1 的分频系数。 00: 不分频，每一个有效的边沿都会产生一次输入； 01: 每 2 个有效的边沿产生一次输入； 10: 每 4 个有效的边沿产生一次输入； 11: 每 8 个有效的边沿产生一次输入。 注：C1EN='0'时，分频系数复位。
位 1: 0	C1C	0x0	rw	通道 1 配置 (Channel 1 configure) 当 C1EN='0'时，这些位用于选择通道 1 为输出或输入，以及输入时的映射选择： 00: 输出； 01: 输入，C1IN 映射在 C1IFP1 上； 10: 保留； 11: 保留。

#### 14.3.4.6 通道控制寄存器 (TMR14\_CCTRL)

域	简称	复位值	类型	功能
位 15: 4	保留	0x0	resd	保持默认值。
位 3	C1CP	0x0	rw	通道 1 互补极性 (Channel 1 complementary polarity) 定义输入信号的有效沿, 详见 C1P 位描述。
位 2	保留	0x0	resd	保持默认值。
位 1	C1P	0x0	rw	通道 1 极性 (Channel 1 polarity) 通道 1 配置为输出: 0: C1OUT 的有效电平为高 1: C1OUT 的有效电平为低 通道 1 配置为输入: C1CP/C1P 位共同定义输入信号有效沿。 00: C1IN 的有效边沿为上升沿; 作为外部触发使用时, C1IN 不反相。 01: C1IN 的有效边沿为下降沿; 作为外部触发使用时, C1IN 反相。 10: 保留 11: C1IN 的有效边沿为上升沿和下降沿; 作为外部触发 使用时, C1IN 不反相。
位 0	C1EN	0x0	rw	通道 1 使能 (Channel 1 enable) 0: 禁止输入或输出; 1: 使能输入或输出。

表 14-8 标准 CxOUT 通道的输出控制位

CxEN 位	CxOUT 输出状态
0	禁止输出 (CxOUT=0)
1	CxOUT = CxORAW + 极性

注意: 连接到标准 CxOUT 通道的外部 I/O 管脚状态, 取决于 CxOUT 通道状态和 GPIO 以及 IOMUX 寄存器。

#### 14.3.4.7 计数值 (TMR14\_CVAL)

域	简称	复位值	类型	功能
位 15: 0	CVAL	0x0000	rw	计数值 (Counter value)

#### 14.3.4.8 预分频器 (TMR14\_DIV)

域	简称	复位值	类型	功能
位 15: 0	DIV	0x0000	rw	分频系数 (Divider value) 计数器时钟频率 $f_{CK\_CNT} = f_{TMR\_CLK} / (\text{DIV}[15:0]+1)$ 溢出事件发生时该寄存器值被传送到实际的预分频寄存器中。

#### 14.3.4.9 周期寄存器 (TMR14\_PR)

域	简称	复位值	类型	功能
位 15: 0	PR	0x0000	rw	周期值 (Period value) 定时器计数的周期值。当周期值为 0 时, 定时器不工作。

#### 14.3.4.10 通道1数据寄存器 (TMR14\_C1DT)

域	简称	复位值	类型	功能
位 15: 0	C1DT	0x0000	rw	通道 1 数据寄存器值 (Channel 1 data register) 若通道 1 配置为输入: C1DT 是前一次通道 1 输入事件 (C1IN) 所保存的 CVAL。 若通道 1 配置为输出: C1DT 是将要和 CVAL 进行比较的值, 写入的值是否会立即生效取决于输出缓存使能位 (C1OBEN), 并根据设置在 C1OUT 上产生相应的输出。

#### 14.3.4.11 通道输入重映射寄存器 (TMR14\_RMP)

域简称	复位值	类型	功能
位 15: 2 保留	0x00	resd	保持默认值。
位 1: 0 TMR14_CH1_IRMP	0x0	rw	TMR14 通道 1 输入重映射 (TMR14 channel 1 input remap) 00: TMR14 通道 1 输入与 GPIO 相连接 01: ERTC_CLK 10: 32 分频后 HEXT 11: CLK_OUT

## 14.4 通用定时器 (TMR15)

### 14.4.1 TMR15简介

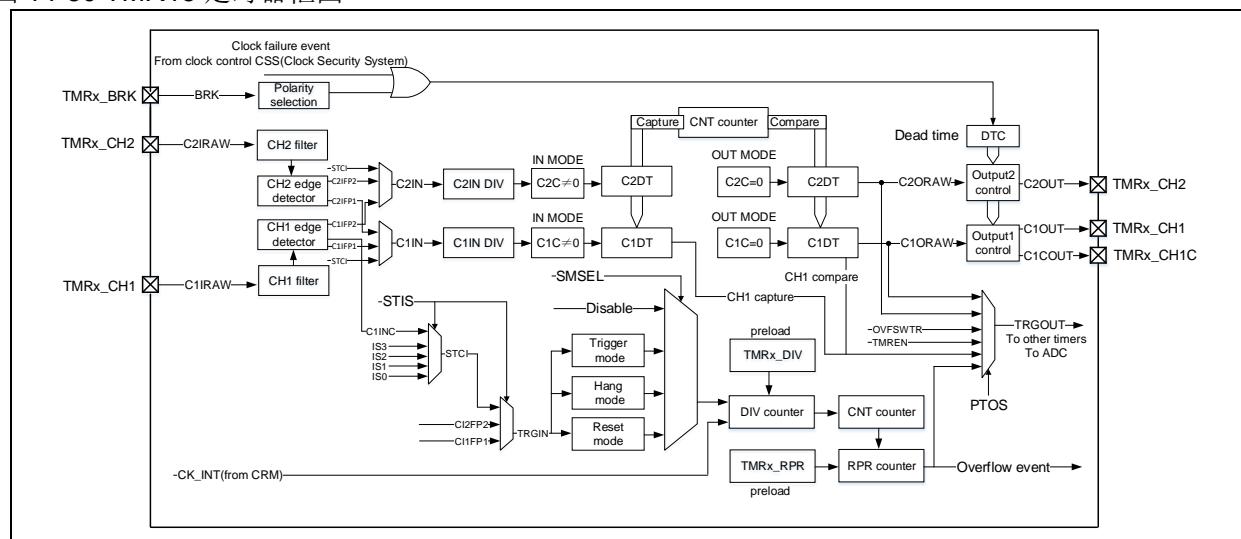
TMR15 包含一个支持向上计数的 16 位计数器、2 个捕获/比较寄存器、2 组独立的通道。可实现嵌入死区、输入捕获、可编程 PWM 输出。

### 14.4.2 TMR15主要特性

TMR15 定时器的功能包括：

- 可选内部时钟、外部输入、内部触发输入用作计数时钟
- 16 位向上计数器、8 位重复计数计数器
- 2 组独立通道，支持输入捕获、输出比较、PWM 生成、单周期模式、死区插入。
- 1 组支持互补输出的独立通道
- 支持 TMR 刹车功能
- 定时器之间可互联同步
- 支持溢出事件、触发事件、刹车输入、通道事件触发中断/DMA
- 支持 TMR burst DMA 传输

图 14-50 TMR15 定时器框图

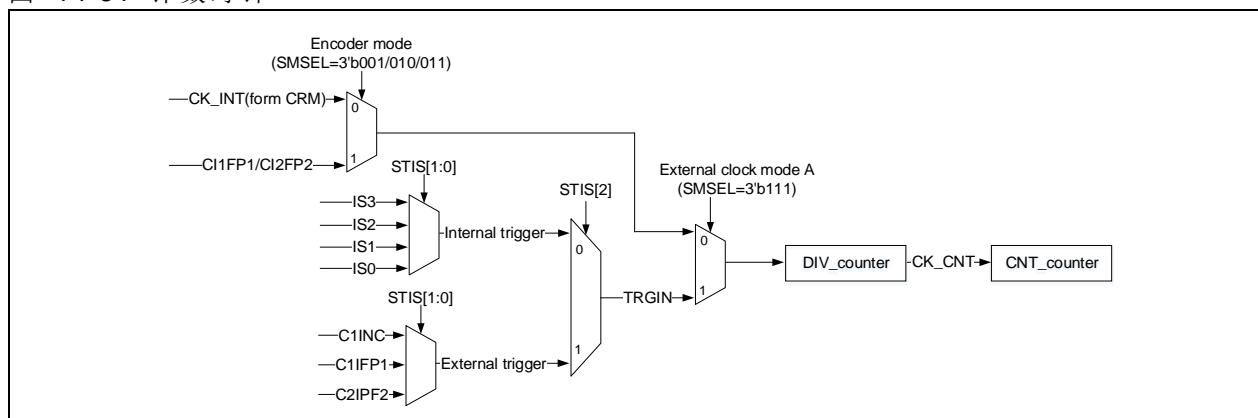


### 14.4.3 TMR15功能描述

#### 14.4.3.1 计数时钟

TMR15 计数时钟可从内部时钟 (CK\_INT)、外部时钟 (外部时钟模式 A)、内部触发输入 (ISx) 这些时钟源提供。

图 14-51 计数时钟

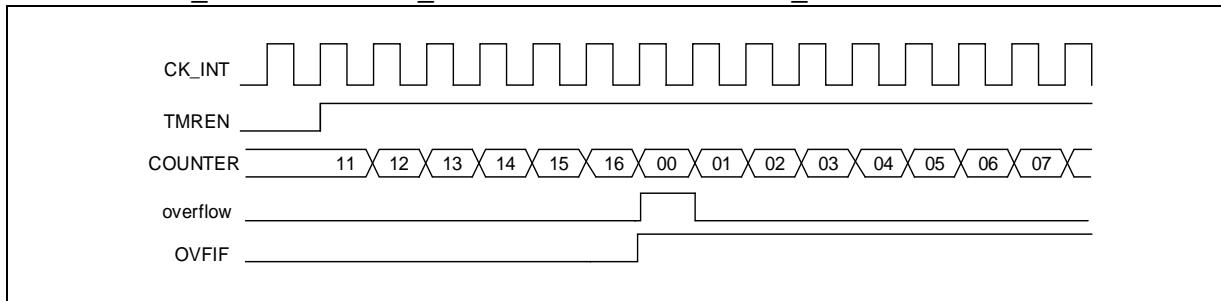


内部时钟 (CK\_INT)

默认下使用 CK\_INT 经由预分频器驱动计数器计数,当 TMR 对应的 APB 时钟预分频系数是 1 时,CK\_INT 频率等于 APB 时钟频率,否则 CK\_INT 频率等于 APB 时钟频率的 2 倍。相关配置流程如下:

- 配置 TMRx\_CTRL1 寄存器 TWCMSEL[1:0], 选择计数模式, 若选择单向对齐计数模式, 还需配置 TMRx\_CTRL1 寄存器 OWCDIR 选择计数方向。
- 配置 TMRx\_DIV 寄存器, 设置计数器计数频率。
- 配置 TMRx\_PR 寄存器, 设置计数器计数周期。
- 配置 TMRx\_CTRL1 寄存器 TMREN, 使能计数器。

图 14-52 使用 CK\_INT 计数, TMRx\_DIV=0x0, 周期寄存器 TMRx\_PR=0x16



### 外部时钟 (TRGIN/EXT)

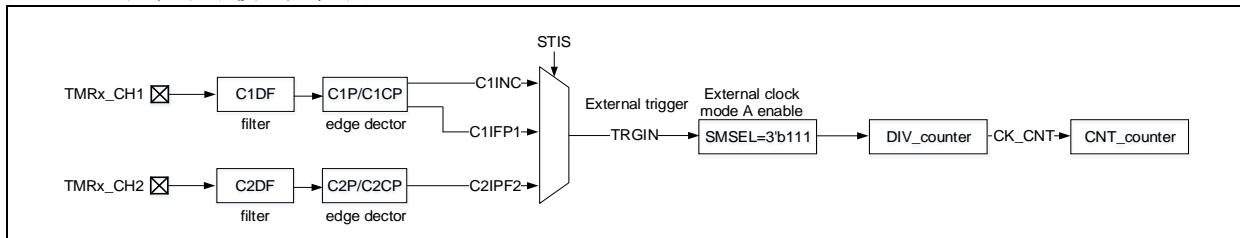
计数时钟由 TRGIN 外部时钟源提供。

当 SMSEL=3'b111 时, 外部时钟模式 A 被选中, 配置 STIS[2: 0] 来选择外部时钟源 TRGIN 信号驱动计数器计数。外部时钟源 TRGIN 可选则 C1INC (STIS=3'b100, 通道 1 上升沿和下降沿信号)、C1IFP1 (STIS=3'b101, 通道 1 滤波且极性选择后信号) 和 C2IFP2 (STIS=3'b110, 通道 2 滤波且极性选择后信号)。

若要使用外部时钟模式 A, 可按如下步骤配置:

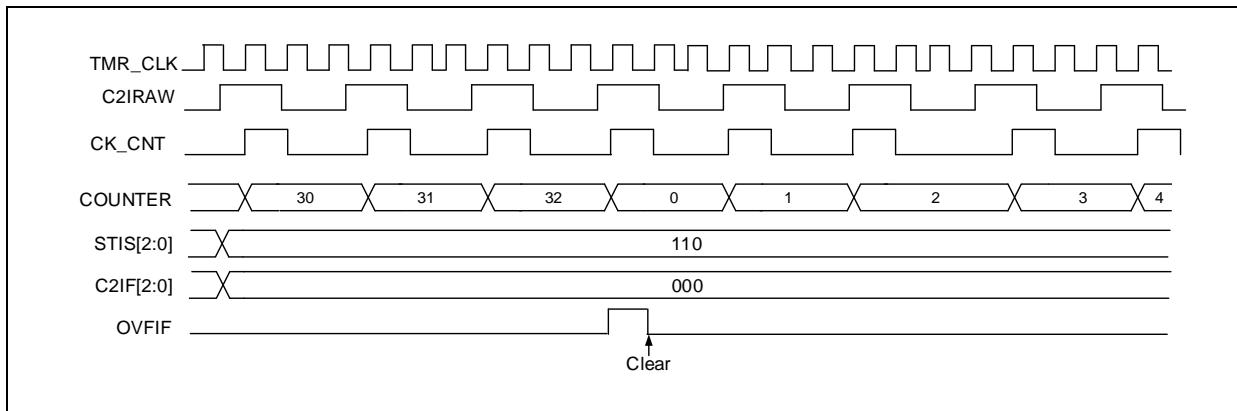
- 配置外部时钟源 TRGIN 参数。
  - 若选择 TRGIN 来源为 TMRx\_CH1, 需配置通道 1 输入滤波 (TMRx\_CM1 寄存器 C1DF[3:0]) 和通道 1 输入极性 (TMRx\_CCTRL 寄存器 C1P/C1CP)。
  - 若选择 TRGIN 来源为 TMRx\_CH2, 需配置通道 2 输入滤波 (TMRx\_CM1 寄存器 C2DF[3:0]) 和通道 1 输入极性 (TMRx\_CCTRL 寄存器 C2P/C2CP)。
- 配置 TMRx\_STCTRL 寄存器 STIS[1:0], 设置 TRGIN 信号来源。
- 配置 TMRx\_STCTRL 寄存器 SMSEL=3'b111, 使能外部时钟模式 A。
- 配置 TMRx\_DIV 寄存器 DIV[15:0], 设置计数器计数频率。
- 配置 TMRx\_PR 寄存器 PR[15:0], 设置计数器计数周期。
- 配置 TMRx\_CTRL1 寄存器 TMREN, 使能计数器。

图 14-53 外部时钟模式 A 框图



注: 由于同步逻辑, 输入端信号与计数器实际时钟之间存在一定延时。

图 14-54 使用外部时钟模式 A 计数, PR=0x32, DIV=0x0



### 内部触发输入 (ISx)

定时器之间支持互联同步，因此一个定时器的 **TMR\_CLK** 可由另一个定时器输出信号 **TRGOUT** 提供。配置 **STIS[2:0]** 选择内部触发信号驱动计数器计数。

**TMR15** 定时器内含一个 16 位预分频器，用于产生驱动计数器计数的时钟 **CK\_CNT**，通过配置 **TMR15\_DIV** 寄存器值，可灵活调整 **CK\_CNT** 与 **TMR\_CLK** 之间的分频关系。预分频值可在任何时刻修改，但只在下一个溢出事件发生时，新值才会生效。

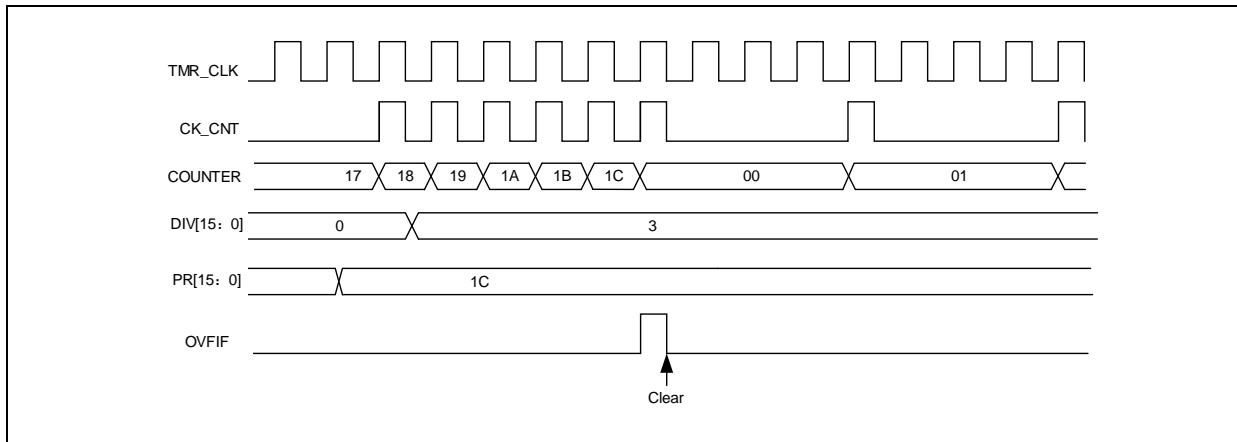
内部触发输入配置流程如下：

- 配置 **TMRx\_PR** 寄存器，设置计数器计数周期。
- 配置 **TMRx\_DIV** 寄存器，设置计数器计数频率。
- 配置 **TMRx\_STCTRL** 寄存器 **STIS[2:0]** 位范围为 3'b000~3'b011，选择内部触发。
- 配置 **TMRx\_STCTRL** 寄存器 **SMSEL[2:0]=3'b111**，选择外部时钟模式 A。
- 配置 **TMRx\_CTRL1** 寄存器 **TMREN** 位，使能 **TMRx** 计数。

表14-9 TMR15内部触发连接

次定时器	IS0 (STIS=000)	IS1 (STIS=001)	IS2 (STIS=010)	IS3 (STIS=011)
TMR15	-	TMR3	TMR16_OC	TMR17_OC

图14-55 当预分频器的参数从1变到4时，计数器的时序图



### 14.4.3.2 计数模式

**TMR15** 定时器提供了多种计数模式，用来满足不同的应用场景。

**TMRx\_PR** 寄存器用于设置计数器计数周期。默认 **TMRx\_PR** 寄存器值会立即传入它的影子寄存器；当开启周期缓冲功能后（**PRBEN** 置 1），**TMRx\_PR** 寄存器值在溢出事件发生时传入它的影子寄存器。

**TMRx\_DIV** 寄存器用于设置计数器计数频率，每 (**DIV[15:0]+1**) 个计数时钟周期，计数器计数一次。和 **TMRx\_PR** 寄存器类似，开启周期缓冲功能后，**TMRx\_DIV** 寄存器值在溢出事件时更新至它的影子寄存器。

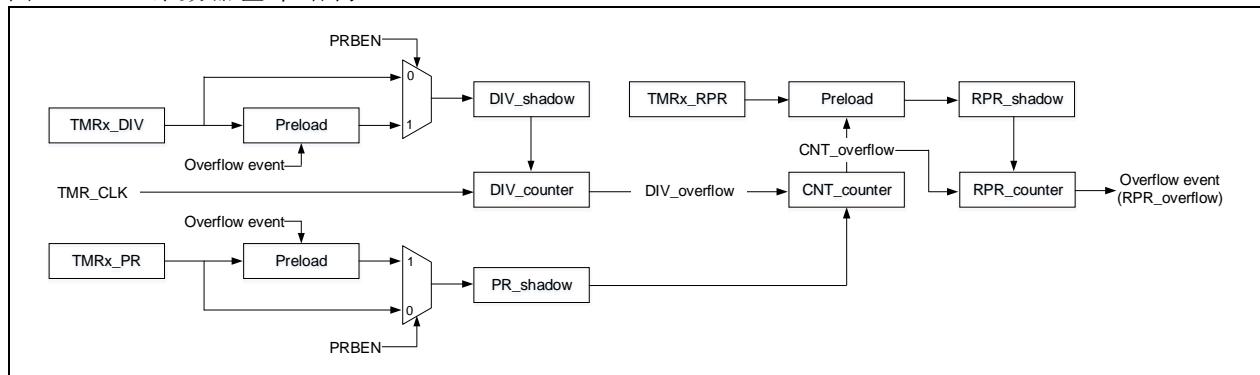
读取 **TMRx\_CNT** 寄存器会返回当前计数器计数值，写入 **TMRx\_CNT** 寄存器会更新计数器当前计数值为写入值。

默认允许产生溢出事件，设置 **TMRx\_CTRL1** 寄存器 **OVFEN=1** 将禁止更新事件产生。**TMRx\_CTRL1** 寄

存器 OVFS 用于选择溢出事件来源，默认计数器上溢或下溢、置位 OVFSWTR、复位模式次定时器控制器产生的复位信号产生溢出事件。置位 OVFS 后，只有计数器上溢或下溢产生溢出事件。

TMREN 位置 1 将使能定时器计数，由于同步逻辑，实际驱动计数器的使能信号 TMR\_EN 相对于 TMREN 延迟一个时钟周期。

图 14-56 计数器基本结构



### 向上计数模式

配置 TMRx\_CTRL1 寄存器 CMSEL[1:0]=2'b00, OWCDIR=1'b0 开启向上计数模式，计数值达到 TMRx\_PR 值时，重新从 0 向上计数，计数器上溢并产生溢出事件，同时 OVFIF 位置 1。若禁止产生溢出事件，计数器溢出后不再重载预分频值和周期值，否则预分频值和周期值在溢出事件后更新。

图 14-57 PRBEN=0 时的溢出事件

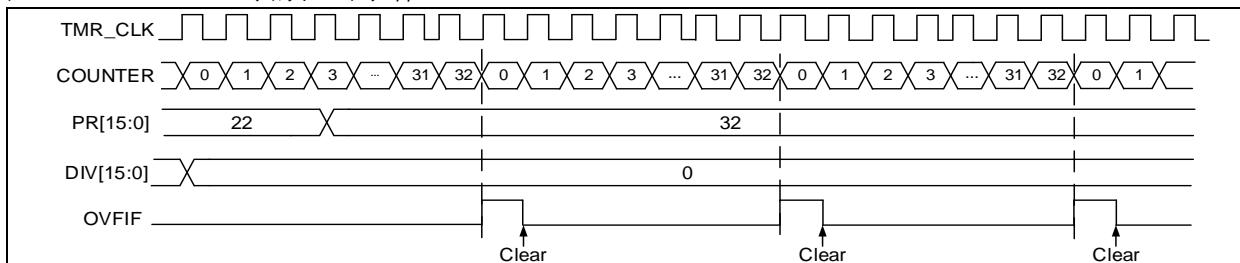
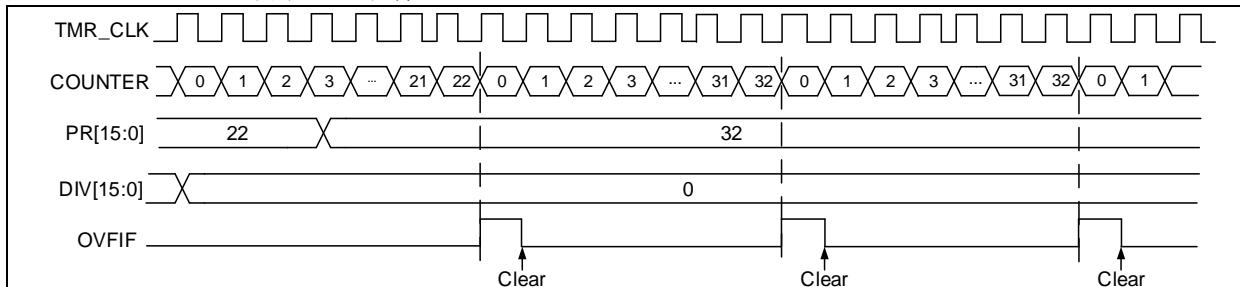


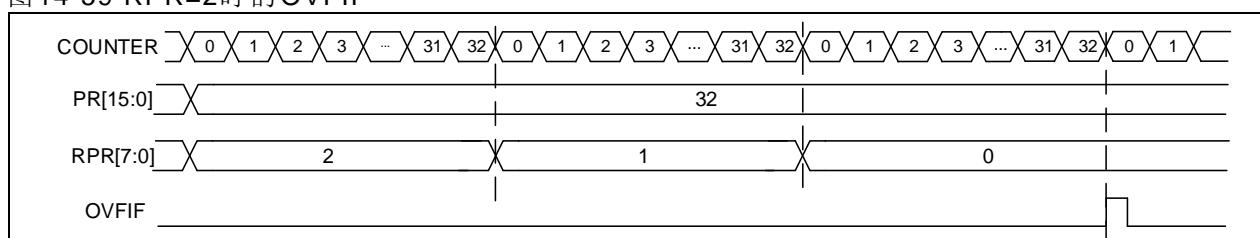
图 14-58 PRBEN=1 时的溢出事件



### 重复计数模式：

TMRx\_RPR 寄存器用于配置重复计数器计数周期，TMRx\_RPR 寄存器为非 0 值时，重复计数模式启动。重复计数模式下，每 (RPR[7:0]+1) 次计数器溢出将产生一次溢出事件。每次计数器溢出，重复计数器递减，仅当重复计数器计数值等于 0 值时，计数器溢出会产生溢出事件。通过配置不同重复计数器值，可调整溢出事件产生的频率。

图 14-59 RPR=2 时的 OVFIF



### 14.4.3.3 TMR 输入部分

TMR15 拥有 2 个独立通道，每个通道可配置为输入或输出，当配置位输入时，每个通道输入信号依次经过以下处理：

- TMRx\_CHx 经过预处理输出 CxIRAW。配置 C1INSEL 位，选择 CxIRAW 来源是 TMRx\_CHx。
- CxIRAW 输入数字滤波器，输出滤波后信号 CxIF。数字滤波器通过 CxDFF 位配置采样频率和次数。
- CxIF 输入边沿检测器，输出边沿选择后信号 CxIFPx。边沿选择由 CxP 和 CxCP 位共同控制，可选择输入上升沿、下降沿或双边沿有效。
- CxIFPx 输入捕获信号选择器，输出选择后信号 CxIN。捕获信号选择器由 CxC 控制，可选择 CxIN 来源为 CxIFPx、CylIFPx、STCI。其中 CylIFPx ( $x \neq y$ ) 是来自通道 y 的 CylIFPy 经通道 x 边沿检测器处理后的信号（例如 C1IFP2 是来自通道 1 的 C1IFP1 信号经过通道 2 边沿检测器处理后的信号）；STCI 来自次定时器控制器，由 STIS 位选择来源。
- CxIN 经由输入通道分频器，输出分频后信号 CxIPS。分频系数由 CxIDIV 位配置为不分频、2 分频、4 分频或 8 分频。

图 14-60 输入/输出通道 1 的主电路

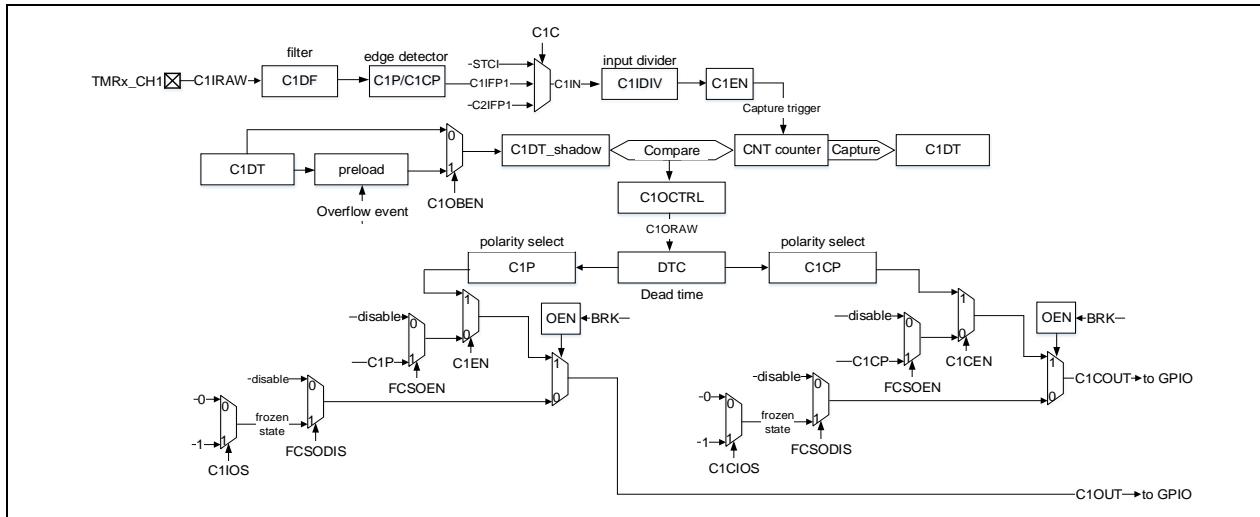
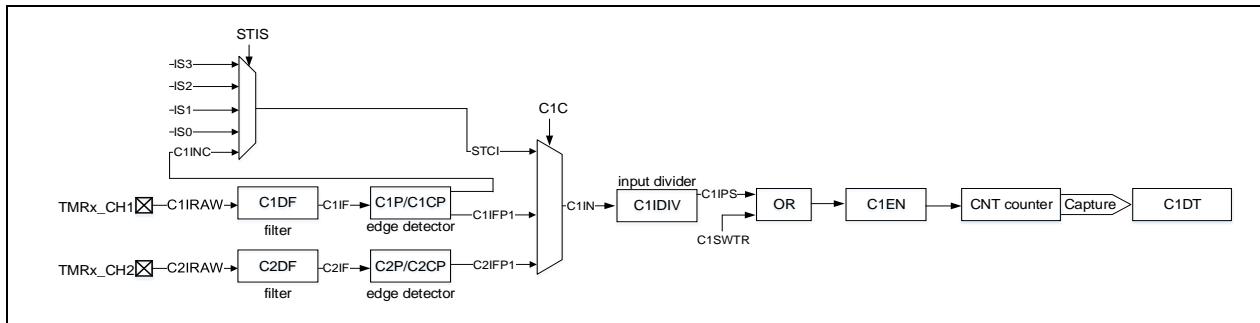


图 14-61 通道 1 输入部分



## 输入模式

此模式下，当选中的触发信号被检测到，通道寄存器（TMRx\_CxDT）记录当前计数器计数值，并将捕获比较中断标志位（CxIF）置 1，若已使能通道中断（CxIEN）、通道 DMA 请求（CxDEN）则产生相应的中断和 DMA 请求。若在 CxIF 置 1 后检测到触发信号，将产生捕获溢出事件，TMRx\_CxDT 会使用当前计数器计数值覆盖之前记录的计数器计数值，同时通道再捕获标志位（CxRF）置 1。

以若要捕获 C1IN 输入的上升沿，可按如下进行配置：

- 将通道模式寄存器1（TMR15\_CM1）中的 C1C 位配置为 01，选择 C1IN 作为通道 1 输入。
- 配置 C1IN 信号滤波器带宽（CxDFF[3:0]）。
- 配置 C1IN 通道的有效沿，在 TMR15\_CCTRL 寄存器中写入 C1P=0（上升沿）。
- 配置 C1IN 信号捕获分频（C1DIV[1:0]）。
- 使能通道 1 输入捕获（C1EN=1）。

根据需要设置 TMR15\_IDEN 寄存器中的 C1IEN 为、TMR15\_IDEN 寄存器中的 C1DEN 位，选择中断请求或 DMA 请求。

### PWM 输入

PWM 输入模式适用于通道 1 和 2，要使用此模式，需要将 C1IN 和 C2IN 映射到同一 TMRx\_CHx，并且通道 1 或 2 的 CxIFPx 配置成触发次定时器控制器复位。

PWM 输入模式可用于测量输入信号的周期和占空比，如需测量通道 1 输入信号的周期和占空比，操作步骤如下：

- 配置 C1C=2'b01，选择 C1IN 为 C1IFP1。
- 配置 C1P=1'b0，选择 C1IFP1 上升沿有效。
- 配置 C2C=2'b10，选择 C2IN 为 C1IFP2。
- 配置 C2P=1'b1，选择 C1IFP2 下降沿有效。
- 配置 STIS=3'b101，选择 次定时器触发信号为 C1IFP1。
- 配置 SMSEL=3'b100，选择 次定时器模式为 复位模式。
- 配置 C1EN=1'b1，C2EN=1'b1。使能通道 1 和 输入捕获。

上述配置下，通道 1 输入信号的上升沿会触发捕获并将捕获值存储到 C1DT 寄存器，同时通道 1 输入信号上升沿复位计数器。通道 1 输入信号下降沿触发捕获并将捕获值存储到 C2DT 寄存器。通道 1 输入信号的周期可通过 C1DT 计算，占空比可通过 C2DT 计算。

图 14-62 PWM 输入模式配置实例

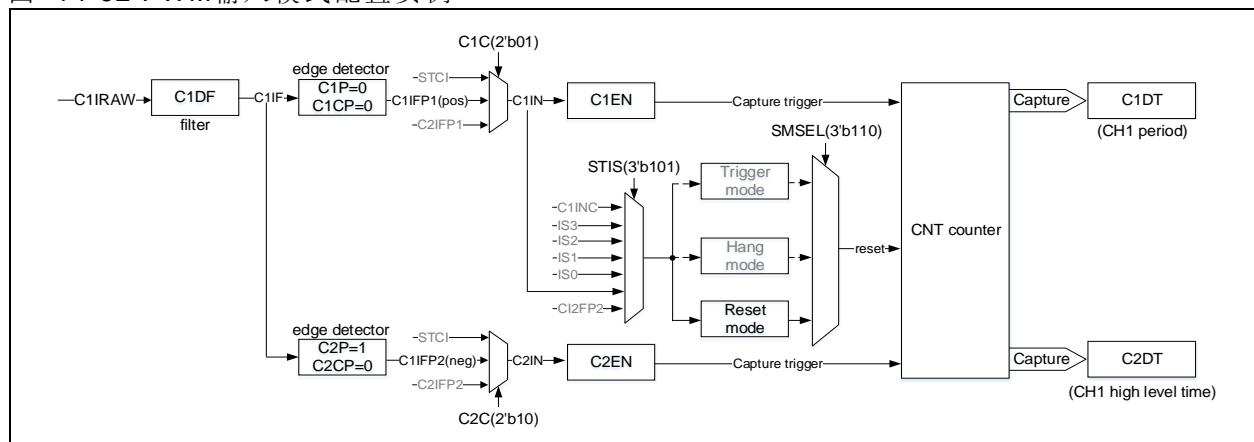
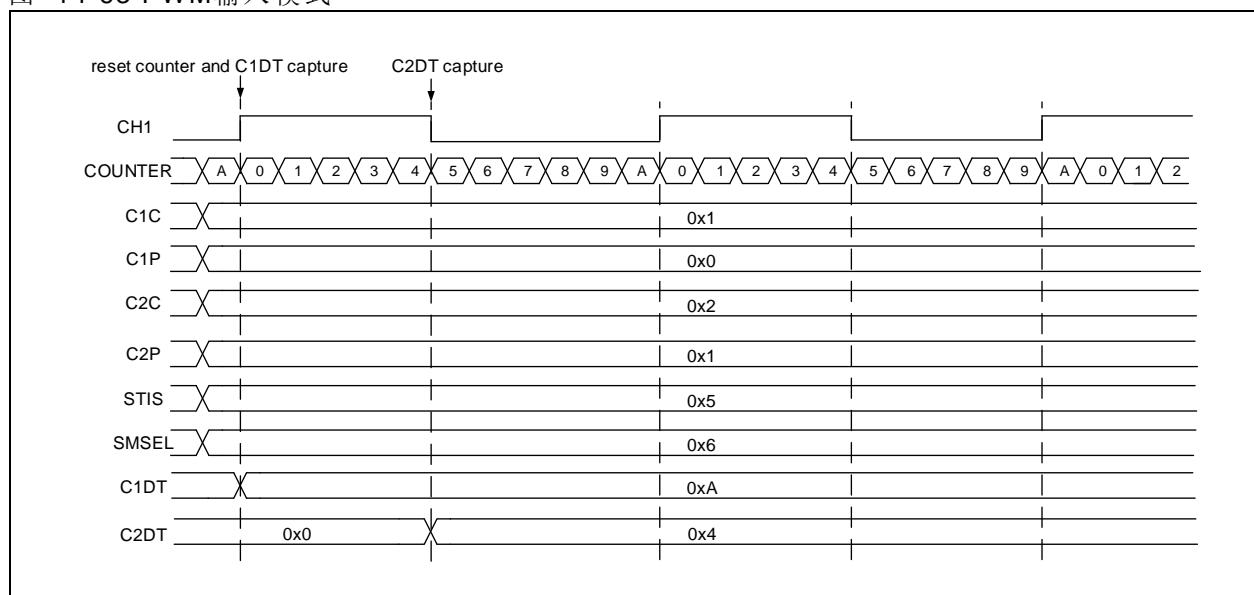


图 14-63 PWM 输入模式



### 14.4.3.4 TMR 输出部分

TMR 的输出部分由比较器和输出控制构成，用于编程输出信号的周期、占空比、极性。TMR15 的输出部分在不同通道上有所不同，如下图所示：

图 14-64 通道 1 输出部分

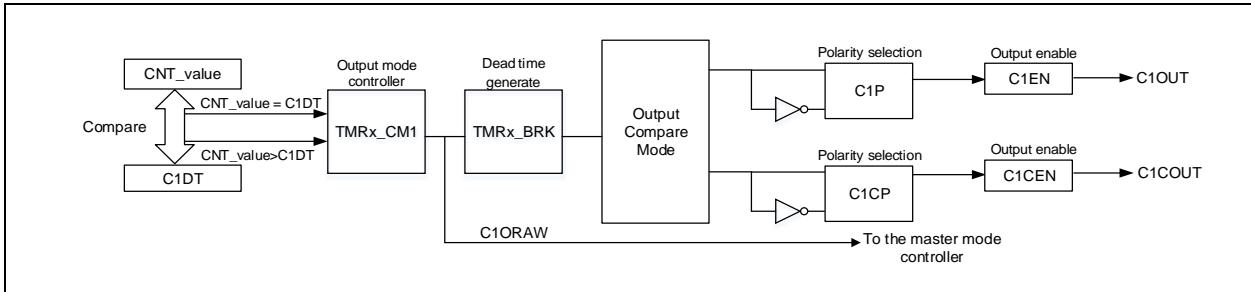
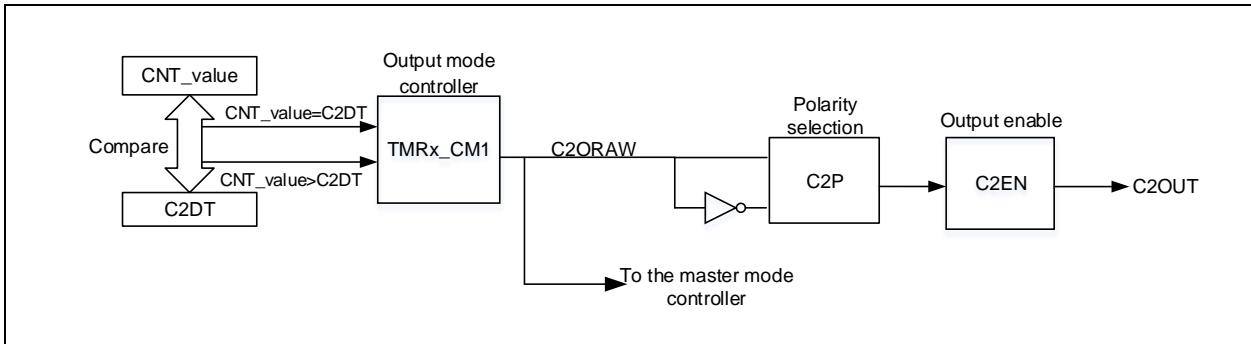


图 14-65 通道 2 输出部分



### 输出模式

配置  $CxC[1:0] \neq 2'b00$  将通道配置为输出可实现多种输出模式，此时，计数器计数值将与  $CxDT$  寄存器值比较，并根据  $CxOCTRL[2:0]$  位配置的输出模式，产生中间信号  $CxORAW$ ，再经过输出控制逻辑处理后输送到 IO。输出信号的周期由  $TMR15_PR$  寄存器值配置，占空比则由  $CxDT$  寄存器值配置。

输出比较模式有以下子类：

**PWM 模式 A:**  $CxOCTRL=3'b110$  时，开启 PWM 模式 A。向上计数时， $TMRx_C1DT > TMRx_CVAL$  时  $C1ORAW$  输出高电平，否则为低电平；向下计数时， $TMRx_C1DT < TMRx_CVAL$  时  $C1ORAW$  输出低电平，否则为高电平。若要使用 PWM 模式 A，可按如下方式配置。

- 配置  $TMRx_PR$  寄存器，设置 PWM 周期。
- 配置  $TMRx_CxDT$  寄存器，设置 PWM 占空比。
- 配置  $TMRx_CM1/CM2$  寄存器  $CxOCTRL$  位为  $3'b110$ ，设置输出模式为 PWM 模式 A。
- 配置  $TMRx_DIV$  寄存器，设置计数器计数频率。
- 配置  $TMRx_CTRL1$  寄存器  $TWCMSEL[1:0]$  位，设置计数器计数模式。
- 配置  $TMRx_CCTRL$  寄存器  $CxP$  位、 $CxCP$  位，设置输出极性。
- 配置  $TMRx_CCTRL$  寄存器  $CxEN$  位、 $CxCEN$  位，使能通道输出。
- 配置  $TMRx_BRK$  寄存器  $OEN$  位，使能  $TMRx$  输出。
- 配置 TMR 输出通道对应 GPIO 为对应的复用模式。
- 配置  $TMRx_CTRL1$  寄存器  $TMREN$  位，使能  $TMRx$  计数。

**PWM 模式 B:**  $CxOCTRL=3'b111$  时，开启 PWM 模式 B。向上计数时， $TMRx_C1DT > TMRx_CVAL$  时  $C1ORAW$  输出低电平，否则为高电平；向下计数时， $TMRx_C1DT < TMRx_CVAL$  时  $C1ORAW$  输出高电平，否则为低电平。

**强制输出模式:**  $CxOCTRL=3'b100/101$  时，开启强制输出模式。此时， $CxORAW$  信号的电平被强制输出为配置的电平，而与计数值无关。虽然输出信号不依赖于比较结果，但通道标志位和 DMA 请求仍依赖于比较结果。

**输出比较模式:**  $CxOCTRL=3'b001/010/011$  时，开启输出比较模式。此时，当计数值与  $CxDT$  值匹配时， $CxORAW$  强制输出高电平（ $CxOCTRL=3'b001$ ）、低电平（ $CxOCTRL=3'b010$ ）或进行电平翻转（ $CxOCTRL=3'b011$ ）。

**单周期模式:** PWM 模式的特例，将  $OCMEN$  位置 1 可开启单周期模式，此模式下，仅在当前计数周期中进行比较匹配，完成当前计数后， $TMREN$  位清 0，因此仅输出一个脉冲。当配置为向上计数模式时，需要严格配置  $CVAL < CxDT \leq PR$ ；向下计数时，需严格配置  $CVAL > CxDT$ 。

**快速输出模式:** 将  $CxOIEN$  位置 1 可开启此功能，开启后  $CxORAW$  电平值不再在计数值与  $CxDT$  匹配时变化，而是在当前计数周期开始时，也就是说，比较结果被提前了，计数器值与  $CxDT$  寄存器的比较结果将会提前决定  $CxORAW$  的电平。

图 14-66 展示了输出比较模式（翻转）的例子，C1DT=0x3，当计数值等于 0x3 时，输出电平 C1OUT 被翻转。

图 14-67 展示了计数器向上计数与 PWM 模式 A 配合的例子，PR=0x32，CxDT 配置为不同的值时输出时输出信号的翻转情况。

图 14-68 展示了计数器向上计数与单周期模式下 PWM 模式 B 配合的例子，计数器仅计数了一个周期，输出信号在这个周期中只输出了一个脉冲。

图 14-66 计数值与 C1DT 值匹配时翻转 C1ORAW

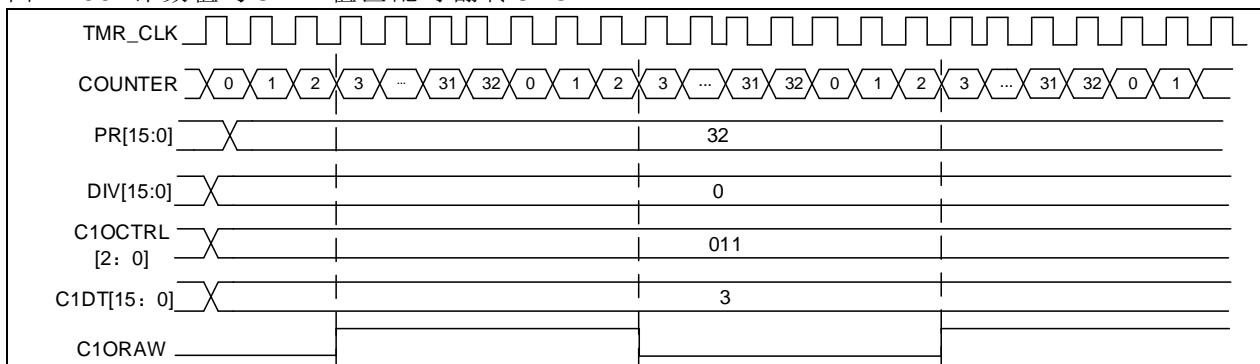


图 14-67 向上计数下 PWM 模式 A

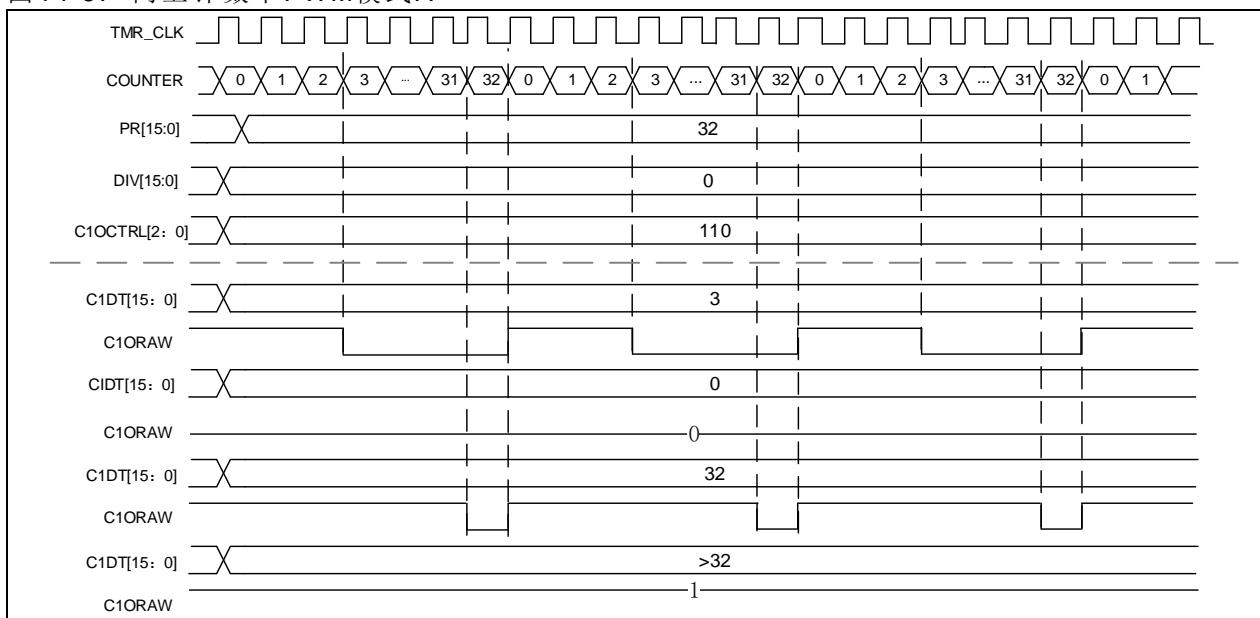
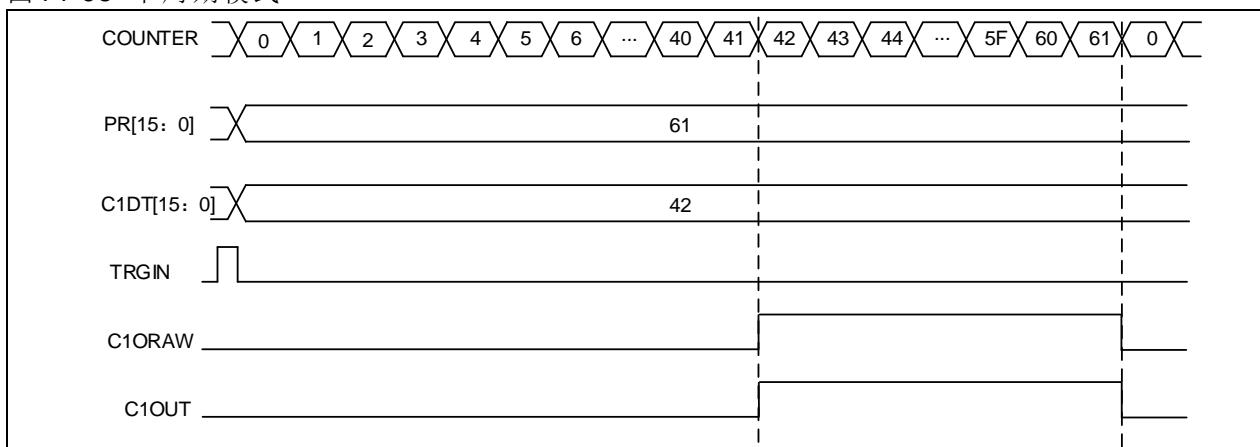


图 14-68 单周期模式



### 主定时器事件输出

当 TMR 作为主定时器时，可选择如下信号源作为 TRGOUT 信号输出到次定时器，选择信号为 TMRxCTRL2 寄存器 PTOS 位。

- PTOS=3'b000, TRGOUT 输出软件溢出事件 (TMRx\_SWEVT 寄存器 OVFSWTR 位)。
- PTOS=3'b001, TRGOUT 输出计数器使能信号。
- PTOS=3'b010, TRGOUT 输出计数器溢出事件。
- PTOS=3'b011, TRGOUT 输出捕获、比较事件。
- PTOS=3'b100, TRGOUT 输出 C1ORAW 信号。
- PTOS=3'b101, TRGOUT 输出 C2ORAW 信号。

### 死区插入

TMR15 通道 1 包含一组反向通道输出，通过 CxCEN 使能，通过 CxCP 配置极性。CxOUT 和 CxCOUT 的输出状态见表 14-11。

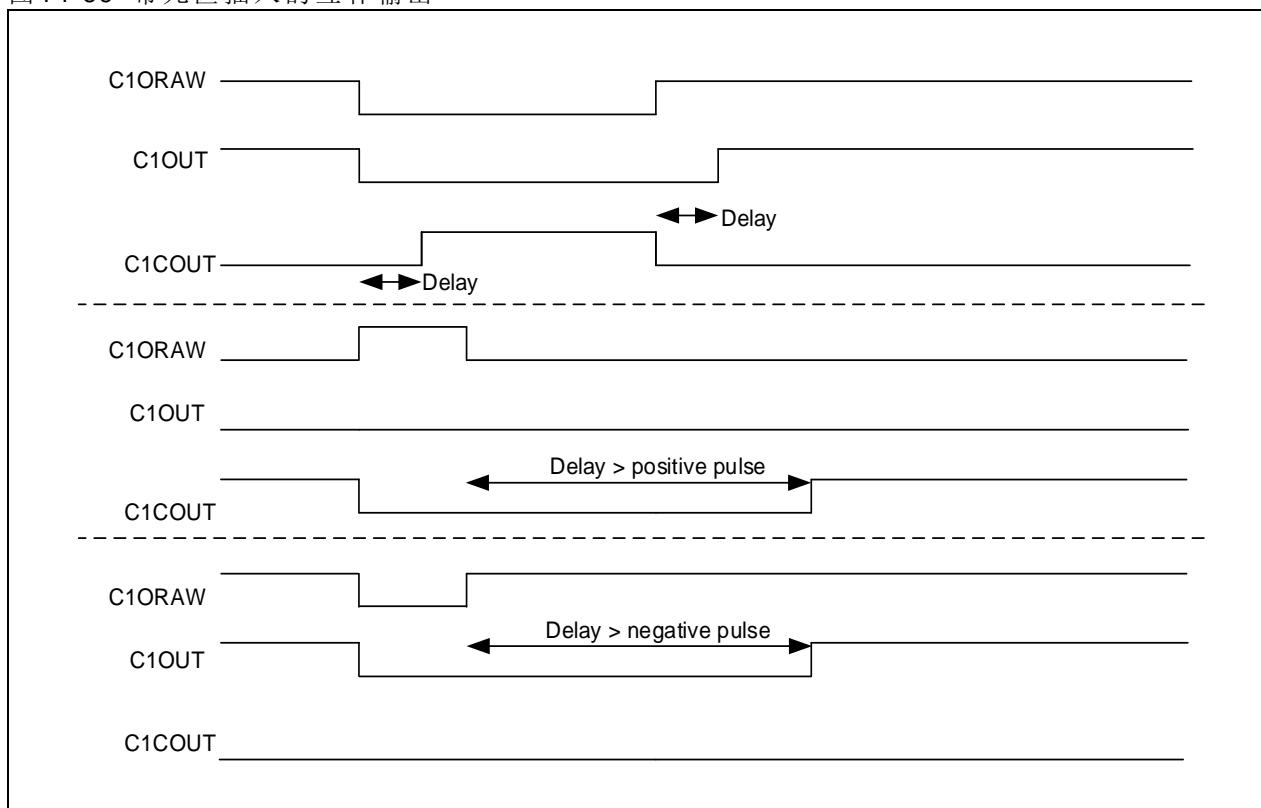
当转换为 IDLEF 状态，即 OEN 下降到 0，死区被激活。

将 CxEN 和 CxCEN 位置 1 后，通过配置 DTC[7:0]死区发生器，可插入不同时间的死区。插入死区后，CxOUT 的上升沿延迟于参考信号的上升沿；CxCOUT 的上升沿延迟于参考信号的下降沿。

如果延迟大于当前有效的输出宽度，C1OUT 和 C1COUT 不会产生相应的脉冲，死区时间应小于有效的输出宽度。

下列图显示了 CxP=0、CxCP=0、OEN=1、CxEN=1 并且 CxCEN=1 时死区插入的例子

图 14-69 带死区插入的互补输出



### 14.4.3.5 TMR刹车功能

开启刹车功能后 (BRKEN 位置 1)，CxOUT 和 CxCOUT 由 OEN、FCSODIS、FCSOEN、CxIOS 和 CxCIOS 共同控制。但 CxOUT 和 CxCOUT 输出总是不能同时处于有效电平上的。详见表 14-11 带刹车功能的互补输出通道 CxOUT 和 CxCOUT 的控制位。

刹车信号来源可以是刹车输入引脚、时钟失效事件，刹车输入信号的极性由 BRKV 位控制。

当发生刹车事件时，有下述动作：

- OEN 位异步清零，通道输出状态由 FCSODIS 位选择。关闭 MCU 的振荡器不影响该功能。
- OEN 被清零后，通道输出电平由 CxIOS 位设定。如果 FCSODIS=0，则定时器输出使能被禁止，否则输出使能始终为高。
- 当使用互补输出时：
  - 输出最开始处于复位状态，也就是无效的状态（取决于极性）。这是异步操作，定时器有无

时钟并不影响此功能。

- 定时器的时钟如果有效，会开启死区生成功能，**CxIOS** 和 **CxCIOS** 位用来配置死区之后的电平。即使在这种情况下，**CxOUT** 和 **CxCOUT** 也不能被同时驱动到有效的电平。  
注意，由于 **OEN** 位同步逻辑，死区时间较通常情况会延长一段时间（大约 2 个 **clk\_tmr** 的时钟周期）。
  - 如果 **FCSODIS=0**，定时器释放使能输出，否则保持使能输出；或一旦 **CxEN** 与 **CxCEN** 之一变高时，使能输出变为高。

注意：刹车输入电平有效时，OEN 不能被设置，状态标志 BRKIF 也不能被清除。

图 14-70 TMR输出控制

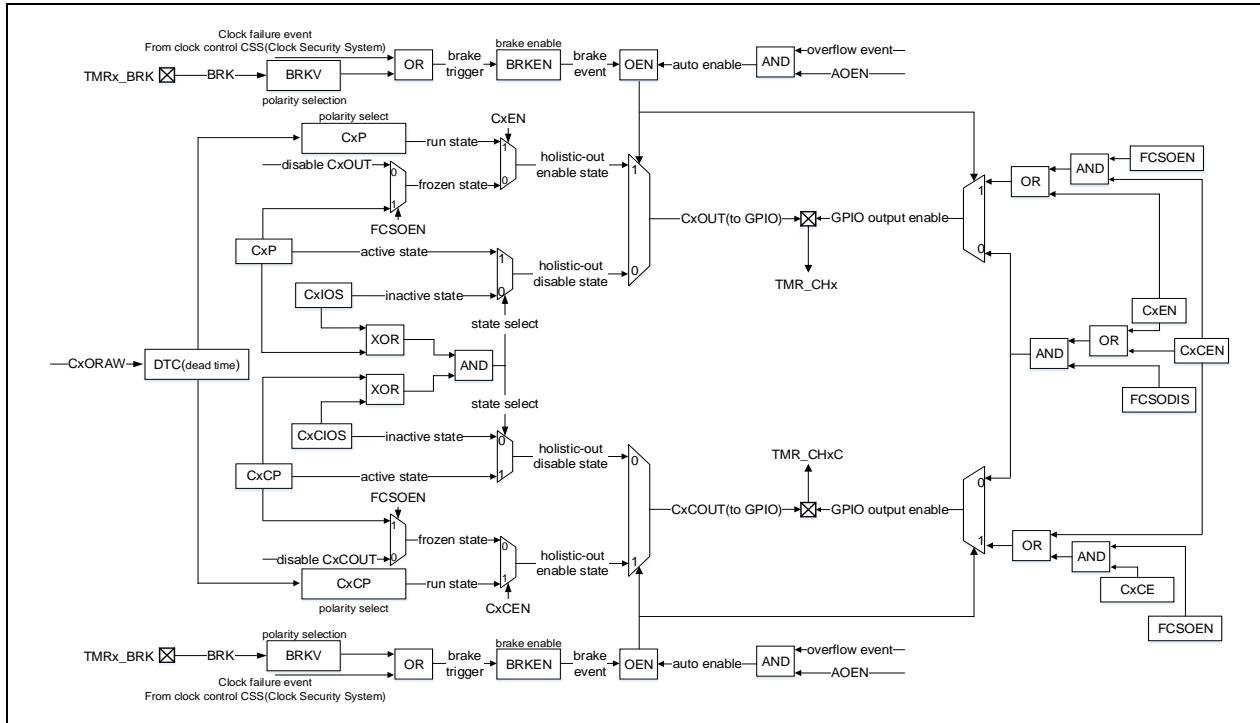
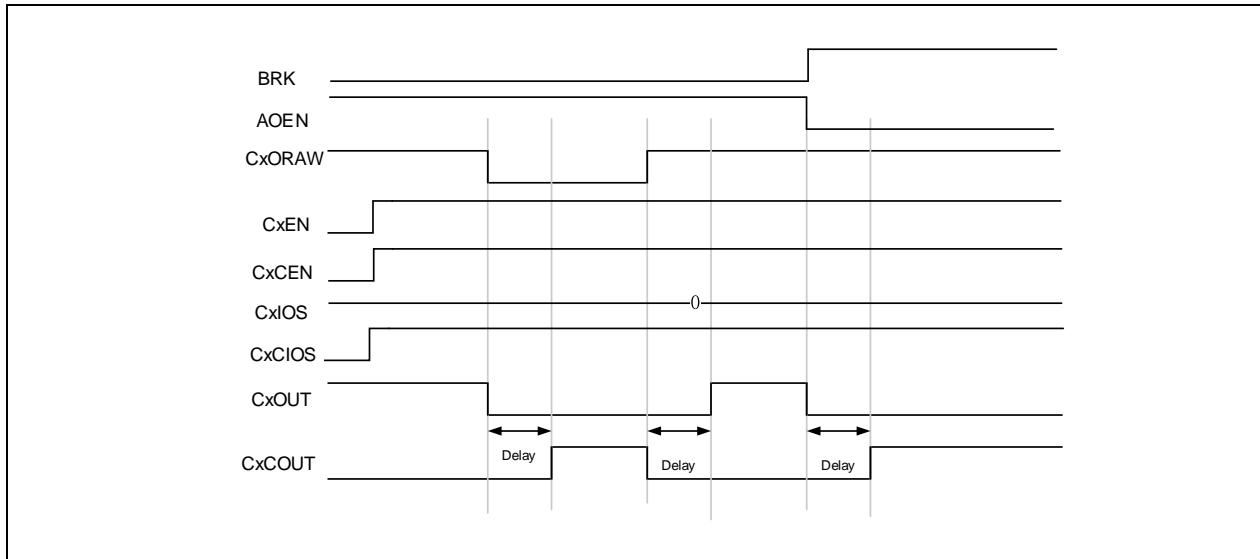


图 14-71 TMR 刹车功能的例子



#### 14.4.3.6 TMR同步

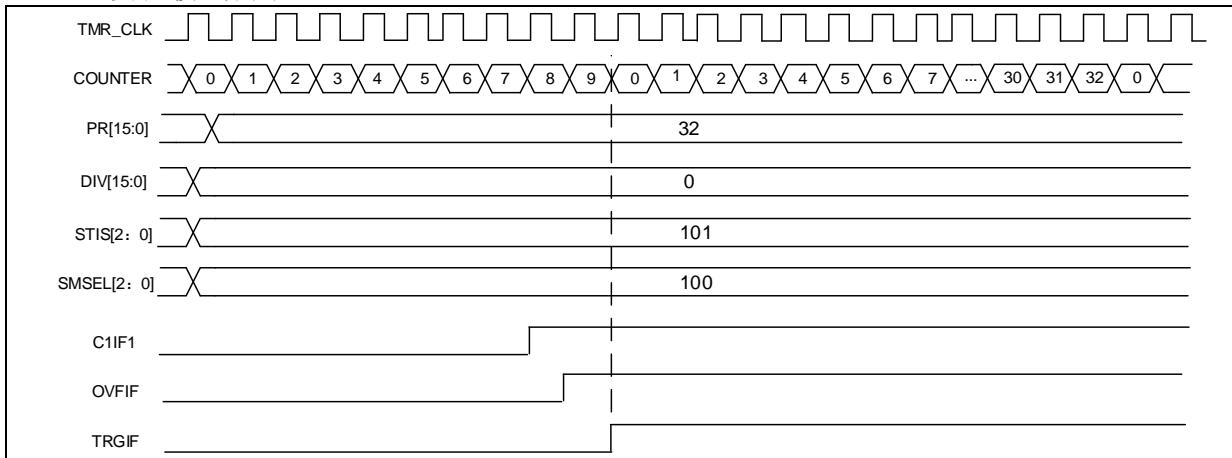
主次定时器之间可由内部连接信号进行同步。主定时器可由 PTOS[2:0]位选择主定时器输出，即同步信息；次定时器由 SMSEL[2:0]位选择从模式，即次定时器的工作模式。

定时器从模式有以下几种：

### 从模式：复位模式

选中的触发信号将复位计数器和预分频器，若 OVFS 位为 0，将产生一个溢出事件。

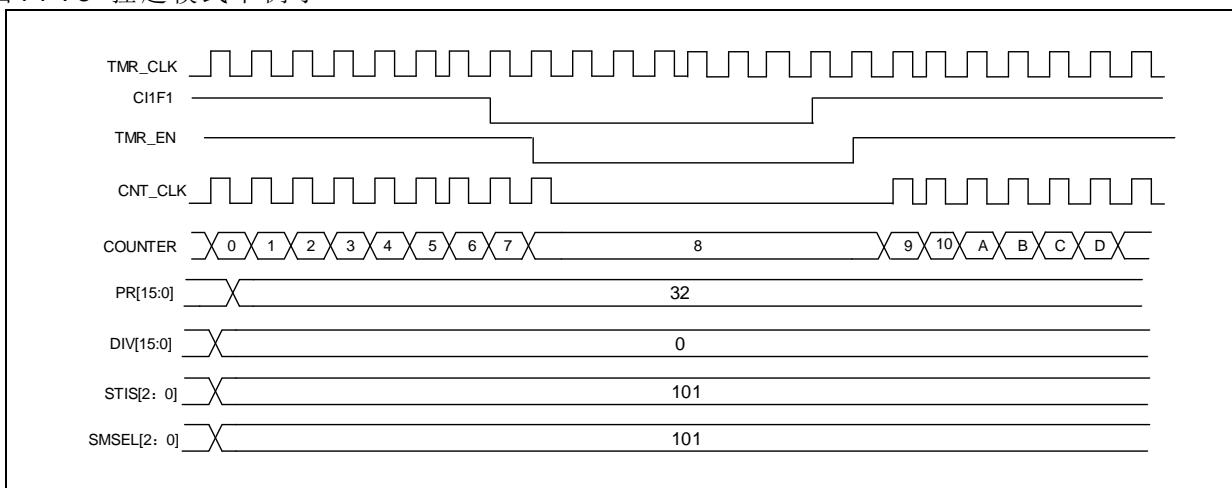
图 14-72 复位模式例子



### 从模式：挂起模式

挂起模式下，计数的计数和停止受选中触发输入信号控制，当触发输入为高电平时计数器开始计数；当为低电平时，计数器暂停计数。

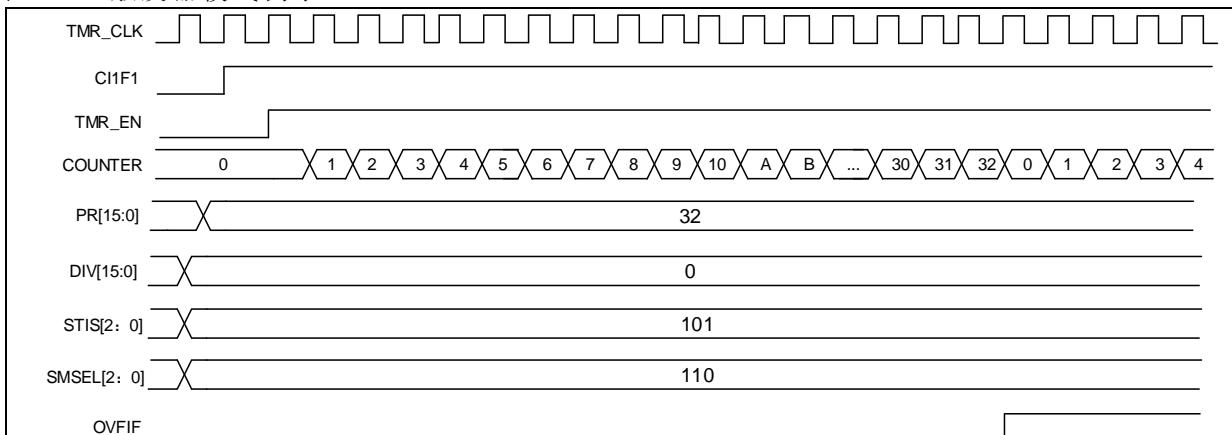
图 14-73 挂起模式下例子



### 从模式：触发模式

计数器将在选中的触发输入上升沿启动计数（将 TMR\_EN 置 1）。

图 14-74 触发器模式例子



定时器的同步的更多实例详见 14.2.3.5 节。

### 14.4.3.7 调试模式

当微控制器进入调试模式（Cortex®-M4 核心停止）时，将 DEBUG 模块中的 TMR15\_PAUSE 置 1，可以使 TMR15 计数器暂停计数。

#### 14.4.4 TMR15寄存器描述

必须以字（32位）的方式操作这些外设寄存器。

下表中将 TMR15 的所有寄存器映射到一个 16 位可寻址（编址）空间

表 14-10 TMR15寄存器和复位值

寄存器简称	基址偏移量	复位值
TMR15_CTRL1	0x00	0x0000
TMR15_CTRL2	0x04	0x0000
TMR15_STCTRL	0x08	0x0000
TMR15_IDEN	0x0C	0x0000
TMR15ISTS	0x10	0x0000
TMR15_SWEVT	0x14	0x0000
TMR15_CM1	0x18	0x0000
TMR15_CCTRL	0x20	0x0000
TMR15_CVAL	0x24	0x0000
TMR15_DIV	0x28	0x0000
TMR15_PR	0x2C	0x0000
TMR15_RPR	0x30	0x0000
TMR15_C1DT	0x34	0x0000
TMR15_C2DT	0x38	0x0000
TMR15_BRK	0x44	0x0000
TMR15_DMACTRL	0x48	0x0000
TMR15_DMADT	0x4C	0x0000

##### 14.4.4.1 TMR15控制寄存器1 (TMR15\_CTRL1)

域	简称	复位值	类型	功能
位 15: 10	保留	0x0	resd	保持默认值。
位 9: 8	CLKDIV	0x0	rw	时钟除频 (Clock divider) 此位用于设置数字滤波器采样频率 $f_{DTS}$ 和定时器时钟频率 $f_{CK\_INT}$ 之间的分频比，也用于调整死区时间的时基 $T_{DTS}$ 和定时器时钟周期 $T_{CK\_INT}$ 的分频比。 00: 无除频, $f_{DTS}=f_{CK\_INT}$ ; 01: 2 除频, $f_{DTS}=f_{CK\_INT}/2$ ; 10: 4 除频, $f_{DTS}=f_{CK\_INT}/4$ ; 11: 保留。
位 7	PRBEN	0x0	rw	周期缓冲使能 (Period buffer enable) 0: 缓冲关闭; 1: 缓冲开启。
位 6: 4	保留	0x0	resd	保持默认值
位 3	OCMEN	0x0	rw	单周期使能 (One cycle mode enable) 该功能用于选择溢出事件后，计数器是否停止。 0: 关闭; 1: 开启。
位 2	OVFS	0x0	rw	溢出事件源选择 (Overflow event source) 配置溢出事件或 DMA 请求来源。 0: 来源于计数器溢出、设置 OVFSWTR 位或次定时器控制器产生的溢出事件; 1: 只能来源于计数器溢出。
位 1	OVFEN	0x0	rw	溢出事件使能 (Overflow event enable) 0: 开启; 1: 关闭。

位 0	TMREN	0x0	rw	使能定时器 (TMR enable) 0: 关闭; 1: 开启。
-----	-------	-----	----	--

#### 14.4.4.2 TMR15控制寄存器2 (TMR15\_CTRL2)

域	简称	复位值	类型	功能
位 31: 11	保留	0x0	resd	保持默认值
位 10	C2IOS	0x0	rw	通道 2 空闲输出状态 (Channel 2 idle output state) 通道 1 互补空闲输出状态 (Channel 1 complementary idle output state)
位 9	C1CIOS	0x0	rw	输出关闭 (OEN = 0), 死区发生后: 0: C1COUT=0; 1: C1COUT=1.
位 8	C1IOS	0x0	rw	通道 1 空闲输出状态 (Channel 1 idle output state) 输出关闭 (OEN = 0), 死区发生后: 0: C1OUT=0. 1: C1OUT=1.
位 7	保留	0x0	resd	保持默认值 主定时器输出信号选择 (Primary TMR output selection) TMR15 输出到次定时器的信号选择: 000: 复位; 001: 使能; 010: 溢出; 011: 比较脉冲; 100: C1ORAW 信号; 101: C2ORAW 信号;
位 6: 4	PTOS	0x0	rw	DMA 请求源 (DMA request source) DMA 请求来源。 0: 通道事件; 1: 溢出事件。
位 3	DRS	0x0	rw	通道控制位刷新选择 (Channel control bit refresh select) 对具有互补输出的通道, 如果通道控制位有缓存时: 0: 通过设置 HALL 位刷新控制位; 1: 通过设置 HALL 位或 TRGIN 的上升沿刷新控制位。
位 1	保留	0x0	resd	保持默认值。
位 0	CBCTRL	0x0	rw	通道缓存控制 (Channel buffer control) 对具有互补输出的通道: 0: CxEN, CxCEN 和 CxOCTRL 位无缓存; 1: CxEN, CxCEN 和 CxOCTRL 位有缓存。

#### 14.4.4.3 TMR15次定时器控制寄存器 (TMR15\_STCTRL)

域	简称	复位值	类型	功能
位 31: 8	保留	0x0	resd	保持默认值
位 7	STS	0x0	rw	次定时器同步 (Subordinate TMR synchronization) 该位开启后, 主次定时器可实现高度同步。 0: 关闭; 1: 开启。
位 6: 4	STIS	0x0	rw	次定时器输入选择 (Subordinate TMR input selection) 用于次定时器的输入选择。 000: 内部选择 0 (IS0); 001: 内部选择 1 (IS1); 010: 内部选择 2 (IS2); 011: 内部选择 3 (IS3); 100: C1IRAW 的输入检测器 (C1INC); 101: 滤波输入 1 (C1IF1); 110: 滤波输入 2 (C2IF2); 关于每个定时器中 ISx 的细节, 参见表 14-9。
位 3	保留	0x0	resd	保留, 保持默认值。
位 2: 0	SMSEL	0x0	rw	次定时器模式选择 (Subordinate TMR mode selection) 000: 关闭从模式;

100: 复位模式 - TRGIN 输入上升沿时, 重新初始化计数器;  
 101: 挂起模式 - TRGIN 输入高电平时, 计数器计数;  
 110: 触发模式 - TRGIN 输入上升沿产生触发事件;  
 111: 外部时钟模式 A - TRGIN 输入上升沿提供时钟;

#### 14.4.4.4 TMR15 DMA/中断使能寄存器 (TMR15\_IDEN)

域	简称	复位值	类型	功能
位 15	保留	0x0	resd	保持默认值。
位 14	TDEN	0x0	rw	触发 DMA 请求使能 (Trigger DMA request enable) 0: 关闭; 1: 开启。
位 13	HALLDE	0x0	rw	HALL DMA 请求使能 (HALL DMA request enable) 0: 关闭; 1: 开启。
位 12: 11	保留	0x0	resd	保持默认值
位 10	C2DEN	0x0	rw	通道 2 的 DMA 请求使能 (Channel 2 DMA request enable) 0: 关闭; 1: 开启。
位 9	C1DEN	0x0	rw	通道 1 的 DMA 请求使能 (Channel 1 DMA request enable) 0: 关闭; 1: 开启。
位 8	OVFDEN	0x0	rw	溢出事件的 DMA 请求使能 (overflow event DMA request enable) 0: 关闭; 1: 开启。
位 7	BRKIE	0x0	rw	刹车中断使能 (Brake interrupt enable) 0: 关闭; 1: 开启。
位 6	TIEN	0x0	rw	触发中断使能 (Trigger interrupt enable) 0: 关闭; 1: 开启。
位 5	HALLIEN	0x0	rw	HALL 中断使能 (HALL interrupt enable) 0: 关闭; 1: 开启。
位 4: 3	保留	0x0	resd	保持默认值
位 2	C2IEN	0x0	rw	通道 2 中断使能 (Channel 2 interrupt enable) 0: 关闭; 1: 开启。
位 1	C1IEN	0x0	rw	通道 1 中断使能 (Channel 1 interrupt enable) 0: 关闭; 1: 开启。
位 0	OVFIEN	0x0	rw	溢出中断使能 (Overflow interrupt enable) 0: 关闭; 1: 开启。

#### 14.4.4.5 TMR15 中断状态寄存器 (TMR15ISTS)

域	简称	复位值	类型	功能
位 15: 11	保留	0x0	resd	保持默认值。
位 10	C2RF	0x0	rw0c	通道 2 再捕获标记 (Channel 2 recapture flag) 见 C1RF 的描述。
位 9	C1RF	0x0	rw0c	通道 1 再捕获标记 (Channel 1 recapture flag) C1IF 的状态已经为'1'时是否再次发生了捕获, 由硬件置'1', 写'0'清除。 0: 无捕获发生; 1: 捕获发生。
位 8	保留	0x0	resd	保持默认值。
位 7	BRKIF	0x0	rw0c	刹车中断标记 (Brake interrupt flag)

				用于标记刹车输入的电平是否有效，由硬件置'1'，写'0'清除。 0: 无效； 1: 有效。
位 6	TRGIF	0x0	rw0c	触发中断标记 (Trigger interrupt flag) 当发生触发事件时由硬件置'1'，写'0'清除。 0: 无触发事件发生； 1: 发生触发事件。 触发事件：在 TRGIN 接收到有效边沿，或挂起模式下接收到任意边沿。
位 5	HALLIF	0x0	rw0c	HALL 中断标记 (HALL interrupt flag) 当发生触发事件时由硬件置'1'，写'0'清除。 0: 无 HALL 事件发生； 1: 发生 HALL 事件。 HALL 事件：CxEN、CxCEN、CxOCTRL 已被更新。
位 4: 3	保留	0x0	resd	保持默认值
位 2	C2IF	0x0	rw0c	通道 2 中断标记 (Channel 2 interrupt flag) 见 C1IF 的描述。
位 1	C1IF	0x0	rw0c	通道 1 中断标记 (Channel 1 interrupt flag) 若通道 1 为输入模式时： 捕获事件发生时由硬件置'1'，由软件清'0' 或读 TMR15_C1DT 清'0'。 0: 无捕获事件发生； 1: 发生捕获事件。 若通道 1 为输出模式时： 比较事件发生时由硬件置'1'，由软件清'0'。 0: 无比较事件发生； 1: 发生比较事件。
位 0	OVFIF	0x0	rw0c	溢出中断标记 (Overflow interrupt flag) 当溢出事件发生时由硬件置'1'，由软件清'0'。 0: 无溢出事件发生； 1: 发生溢出事件，若 TMR15_CTRL1 的 OVFEN=0、 OVFS=0 时： - 当 TMR15_SWEVE 寄存器的 OVFG=1 时产生溢出事件； - 当计数值 CVAL 被触发事件重初始化时产生溢出事件。

#### 14.4.4.6 TMR15软件事件寄存器 (TMR15\_SWEVT)

域	简称	复位值	类型	功能
位 15: 8	保留	0x0	resd	保持默认值。
位 7	BRKSWTR	0x0	wo	软件触发刹车事件 (Brake event triggered by software) 通过软件触发一个刹车事件。 0: 无作用； 1: 制造一个刹车事件。
位 6	TRGSWTR	0x0	wo	软件触发触发事件 (Trigger event triggered by software) 通过软件触发一个触发事件。 0: 无作用； 1: 制造一个触发事件。
位 5	HALLSWTR	0x0	wo	软件触发 HALL 事件 (HALL event triggered by software) 通过软件产生一个 HALL 事件。 0: 无作用； 1: 产生一个 HALL 事件。 注：该位只对拥有互补输出的通道有效。
位 4: 3	保留	0x0	resd	保持默认值
位 2	C2SWTR	0x0	wo	软件触发通道 2 事件 (Channel 2 event triggered by software) 见 C1M 的描述。
位 1	C1SWTR	0x0	wo	C1SWTR：软件触发通道 1 事件 (Channel 1 event triggered by software)

				通过软件触发一个通道 1 事件。 0: 无作用; 1: 制造一个通道 1 事件。
位 0	OVFSWTR	0x0	wo	软件触发溢出事件 (Overflow event triggered by software) 通过软件触发一个溢出事件。 0: 无作用; 1: 制造一个溢出事件。

#### 14.4.4.7 TMR15通道模式寄存器1 (TMR15\_CM1)

通道可用于输入（捕获模式）或输出（比较模式），通道的方向由相应的 CxC 位定义。该寄存器其它位的作用在输入和输出模式下不同。CxOx 描述了通道在输出模式下的功能，CxIx 描述了通道在输入模式下的功能。因此必须注意，同一个位在输出模式和输入模式下的功能是不同的。

##### 输出比较模式

域	简称	复位值	类型	功能
位 15	C2OSEN	0x0	rw	通道 2 输出开关使能 (Channel 2 output switch enable)
位 14: 12	C2OCTRL	0x0	rw	通道 2 输出控制 (Channel 2 output control)
位 11	C2OBEN	0x0	rw	通道 2 输出缓存使能 (Channel 2 output buffer enable)
位 10	C2OIEN	0x0	rw	通道 2 输出立即使能 (Channel 2 output immediately enable)
位 9: 8	C2C	0x0	rw	通道 2 配置 (Channel 2 configure) 当 C2EN='0'时，这些位用于选择通道 2 为输出或输入，以及输入时的映射选择： 00: 输出; 01: 输入，C2IN 映射在 C2IFP2 上; 10: 输入，C2IN 映射在 C1IFP2 上; 11: 输入，C2IN 映射在 STI 上，只有在 STIS 选择内部触发输入时才工作。
位 7	C1OSEN	0x0	rw	通道 1 输出开关使能 (Channel 1 output switch enable) 0: EXT 输入不影响 C1ORAW; 1: 当 EXT 输入高电平时，将 C1ORAW 清 0。
位 6: 4	C1OCTRL	0x0	rw	通道 1 输出控制 (Channel 1 output control) 这些位用于设置原始信号 C1ORAW 的工作状态。 000: 断开。断开 C1ORAW 到 C1OUT 的输出; 001: 设置 C1ORAW 为高: TMR15_CVAL=TMR15_C1DT 时。 010: 设置 C1ORAW 为低: TMR15_CVAL=TMR15_C1DT 时。 011 : 切换 C1ORAW 的电平 : 当 TMR15_CVAL=TMR15_C1DT 时。 100: 固定 C1ORAW 为低。 101: 固定 C1ORAW 为高。 110: PWM 模式 A -OWCDIR=0, 若 TMR15_C1DT>TMR15_CVAL 时设置 C1ORAW 为高，否则为低; -OWCDIR=1, 若 TMR15_C1DT < TMR15_CVAL 时设置 C1ORAW 为低，否则为高。 111: PWM 模式 B -OWCDIR=0, 若 TMR15_C1DT > TMR15_CVAL 时设置 C1ORAW 为低，否则为高; -OWCDIR=1, 若 TMR15_C1DT < TMR15_CVAL 时设置 C1ORAW 为高，否则为低。 注：除'000'外，其余配置下 C1OUT 将连接到 C1ORAW，C1OUT 的输出电平除了会根据 C1ORAW 变化外，还与 CCTRL 所配置的输出极性有关。
位 3	C1OBEN	0x0	rw	通道 1 输出缓存使能 (Channel 1 output buffer enable) 0: 关闭 TMR15_C1DT 的缓存功能，写入 TMR15_C1DT 的内容会立即生效。

				1: 启用 TMR15_C1DT 的缓存功能，写入 TMR15_C1DT 的内容将保存到缓存寄存器中，当发生溢出事件时再更新到 TMR15_C1DT 中。
位 2	C1OIEN	0x0	rw	通道 1 输出立即使能 (Channel 1 output immediately enable) 在 PWM 模式 A 或模式 B 下，该位能够缩短触发事件到通道 1 的输出响应间的时间。 0: 需要比较 CVAL 与 C1DT 的值之后再产生输出。 1: 无需比较 CVAL 与 C1DT 的值，当发生触发事件时立即产生输出。
位 1: 0	C1C	0x0	rw	通道 1 配置 (Channel 1 configure) 当 C1EN=0'时，这些位用于选择通道 1 为输出或输入，以及输入时的映射选择： 00: 输出； 01: 输入，C1IN 映射在 C1IFP1 上； 10: 输入，C1IN 映射在 C2IFP1 上； 11: 输入，C1IN 映射在 STCI 上，只有在 STIS 选择内部触发输入时才工作。
<b>输入模式</b>				
域	简称	复位值	类型	功能
位 15: 12	C2DF	0x0	rw	通道 2 滤波器 (Channel 2 digital filter)
位 11: 10	C2IDIV	0x0	rw	通道 2 分频系数 (Channel 2 input divider)
位 9: 8	C2C	0x0	rw	通道 2 配置 (Channel 2 configure) 当 C2EN=0'时，这些位用于选择通道 2 为输出或输入，以及输入时的映射选择： 00: 输出； 01: 输入，C2IN 映射在 C2IFP2 上； 10: 输入，C2IN 映射在 C1IFP2 上； 11: 输入，C2IN 映射在 STCI 上，只有在 STIS 选择内部触发输入时才工作。
位 7: 4	C1DF	0x0	rw	通道 1 滤波器 (Channel 1 digital filter) 这些位用于配置通道 1 的滤波器。滤波的个数为 N，则表示发生了 N 次采样事件后输入边沿才能通过滤波器： 0000: 无滤波器，以 $f_{DTS}$ 采样 1000: 采样频率 $f_{SAMPLING} = f_{DTS}/8, N=6$ 0001: 采样频率 $f_{SAMPLING} = f_{CK\_INT}, N=2$ 1001 : 采样频率 $f_{SAMPLING} = f_{DTS}/8, N=8$ 0010: 采样频率 $f_{SAMPLING} = f_{CK\_INT}, N=4$ 1010 : 采样频率 $f_{SAMPLING} = f_{DTS}/16, N=5$ 0011: 采样频率 $f_{SAMPLING} = f_{CK\_INT}, N=8$ 1011 : 采样频率 $f_{SAMPLING} = f_{DTS}/16, N=6$ 0100: 采样频率 $f_{SAMPLING} = f_{DTS}/2, N=6$ 1100 : 采样频率 $f_{SAMPLING} = f_{DTS}/16, N=8$ 0101: 采样频率 $f_{SAMPLING} = f_{DTS}/2, N=8$ 1101 : 采样频率 $f_{SAMPLING} = f_{DTS}/32, N=5$ 0110: 采样频率 $f_{SAMPLING} = f_{DTS}/4, N=6$ 1110: 采样频率 $f_{SAMPLING} = f_{DTS}/32, N=6$ 0111: 采样频率 $f_{SAMPLING} = f_{DTS}/4, N=8$ 1111 : 采样频率 $f_{SAMPLING} = f_{DTS}/32, N=8$
位 3: 2	C1IDIV	0x0	rw	通道 1 分频系数 (Channel 1 input divider) 这些位定义了通道 1 的分频系数。 00: 不分频，每一个有效的边沿都会产生一次输入； 01: 每 2 个有效的边沿产生一次输入； 10: 每 4 个有效的边沿产生一次输入； 11: 每 8 个有效的边沿产生一次输入。 注：C1EN=0'时，分频系数复位。
位 1: 0	C1C	0x0	rw	通道 1 配置 (Channel 1 configure) 当 C1EN=0'时，这些位用于选择通道 1 为输出或输入，以及输入时的映射选择： 00: 输出；

01: 输入, C1IN 映射在 C1IFP1 上;  
 10: 输入, C1IN 映射在 C2IFP1 上;  
 11: 输入, C1IN 映射在 STCI 上, 只有在 STIS 选择内部触发输入时才工作。

#### 14.4.4.8 TMR15通道控制寄存器 (TMR15\_CCTRL)

域	简称	复位值	类型	功能
位 15: 8	保留	0x0	resd	保持默认值。
位 7	C2CP	0x0	rw	通道 2 互补极性 (Channel 2 complementary polarity) 定义输入信号的有效沿, 详见 C1P 位描述。
位 6	保留	0x0	resd	保持默认值。
位 5	C2P	0x0	rw	通道 2 极性 (Channel 2 polarity) 见 C1P 的描述。
位 4	C2EN	0x0	rw	通道 2 使能 (Channel 2 enable) 见 C1EN 的描述。
位 3	C1CP	0x0	rw	通道 1 互补极性 (Channel 1 complementary polarity) 0: C1COUT 的有效电平为高 1: C1COUT 的有效电平为低
位 2	C1CEN	0x0	rw	通道 1 互补使能 (Channel 1 complementary enable) 0: 禁止输出; 1: 使能输出。
位 1	C1P	0x0	rw	通道 1 极性 (Channel 1 polarity) 通道 1 配置为输出: 0: C1OUT 的有效电平为高 1: C1OUT 的有效电平为低 通道 1 配置为输入: C1CP/C1P 位共同定义输入信号有效沿。 00: C1IN 的有效边沿为上升沿; 作为外部触发使用时, C1IN 不反相。 01: C1IN 的有效边沿为下降沿; 作为外部触发使用时, C1IN 反相。 10: 保留 11: C1IN 的有效边沿为上升沿和下降沿; 作为外部触发使用时, C1IN 不反相。
位 0	C1EN	0x0	rw	通道 1 使能 (Channel 1 enable) 0: 禁止输入或输出; 1: 使能输入或输出。

表 14-11 带刹车功能的互补输出通道 CxOUT 和 CxCOUT 的控制位

控制位					输出状态 (1)	
OEN 位	FCSODI S 位	FCSOE N 位	CxEN 位	CxCEN 位	CxOUT 输出状态	CxCOUT 输出状态
1	X	0	0	0	输出禁止 (与定时器断开) CxOUT=0, Cx_EN=0	输出禁止 (与定时器断开) CxCOUT=0, CxCEN=0
		0	0	1	输出禁止 (与定时器断开) CxOUT=0, Cx_EN=0	CxORAW + 极性, CxCOUT=CxORAW xor CxCP, CxCEN=1
		0	1	0	CxORAW+极性, CxOUT=CxORAW xor CxP, Cx_EN=1	输出禁止 (与定时器断开) CxCOUT=0, CxCEN=0
		0	1	1	CxORAW+极性+死区, Cx_EN=1	CxORAW 反相+极性+死区, CxCEN=1
		1	0	0	输出禁止 (与定时器断开) CxOUT=CxP, Cx_EN=0	输出禁止 (与定时器断开) CxCOUT=CxCP, CxCEN=0

		1 0 1	关闭状态 (输出使能且为无效电平) CxOUT=CxP, Cx_EN=1	CxORAW + 极性, CxCOUT=CxORAW xor CxCP, CxCEN=1
		1 1 0	CxORAW + 极性, CxOUT= CxORAW xor CxP, Cx_EN=1	关闭状态 (输出使能且为无效电平) CxCOUT=CxCP, CxCEN=1
		1 1 1	CxORAW+极性+死区, Cx_EN=1	CxORAW 反相+极性+死区, CxCEN=1
0	X	0	0 0	输出禁止 (对应 IO 与定时器断开, IO 浮空) 异步地: CxOUT=CxP , Cx_EN=0 , CxCOUT=CxCP , CxCEN=0; 若时钟存在: 经过一个死区时间后 CxOUT=CxIOS, CxCOUT=CxCIOS, 假设 CxIOS 与 CxCIOS 并不都对应 CxOUT 和 CxCOUT 的有效电平。
		0	0 1	
		0	1 0	
		0	1 1	
		1	0 0	
		1	0 1	
		1	1 0	
		1	1 1	

注意: 如果一个通道的 2 个输出都没有使用 ( $CxEN = CxCEN = 0$ ) , 那么  $CxIOS$ ,  $CxCIOS$ ,  $CxP$  和  $CxCP$  都必须清零。

注意: 引脚连接到互补的  $CxOUT$  和  $CxCOUT$  通道的外部 I/O 引脚的状态, 取决于  $CxOUT$  和  $CxCOUT$  通道状态和 GPIO 以及 IOMUX 寄存器。

#### 14.4.4.9 TMR15计数值 (TMR15\_CVAL)

域	简称	复位值	类型	功能
位 15: 0	CVAL	0x0	rw	计数值 (Counter value)

#### 14.4.4.10 TMR15预分频器 (TMR15\_DIV)

域	简称	复位值	类型	功能
位 15: 0	DIV	0x0	rw	分频系数 (Divider value) 计数器时钟频率 $f_{CK_CNT} = f_{TMR_CLK} / (DIV[15: 0]+1)$ 溢出事件发生时该寄存器值被传送到实际的预分频寄存器中。

#### 14.4.4.11 TMR15周期寄存器 (TMR15\_PR)

域	简称	复位值	类型	功能
位 15: 0	PR	0x0	rw	周期值 (Period value) 定时器计数的周期值。当周期值为 0 时, 定时器不工作。

#### 14.4.4.12 TMR15重复周期寄存器 (TMR15\_RPR)

域	简称	复位值	类型	功能
位 15: 8	保留	0x0	resd	保持默认值。
位 7: 0	RPR	0x0	rw	重复周期的次数 (Repetition of period value) 这些位用于减慢溢出事件发生的速率, 当重复周期的次数 减为 0 时才会发生溢出事件。

#### 14.4.4.13 TMR15通道1数据寄存器 (TMR15\_C1DT)

域	简称	复位值	类型	功能
位 15: 0 C1DT		0x0	rw	<p>通道 1 数据寄存器值 (Channel 1 data register)</p> <p>若通道 1 配置为输入:</p> <p>C1DT 是前一次通道 1 输入事件 (C1IN) 所保存的 CVAL。</p> <p>若通道 1 配置为输出:</p> <p>C1DT 是将要和 CVAL 进行比较的值, 写入的值是否会立即生效取决于输出缓存使能位 (C1OBEN), 并根据设置在 C1OUT 上产生相应的输出。</p>

#### 14.4.4.14 TMR15通道2数据寄存器（TMR15\_C2DT）

域	简称	复位值	类型	功能
位 15: 0 C2DT		0x0	rw	<p>通道 2 数据寄存器值 (Channel 2 data register)</p> <p>若通道 2 配置为输入:</p> <p>C2DT 是前一次通道 2 输入事件 (C2IN) 所保存的 CVAL。</p> <p>若通道 2 配置为输出:</p> <p>C2DT 是将要和 CVAL 进行比较的值, 写入的值是否会立即生效取决于输出缓存使能位 (C2OBEN), 并根据设置在 C2OUT 上产生相应的输出。</p>

#### 14.4.4.15 TMR15刹车寄存器 (TMR15\_BRK)

域	简称	复位值	类型	功能
位 31: 17	保留	0x0	resd	保持默认值 刹车输入滤波 (brake input filter) 这些位用于配置刹车输入的滤波器。滤波的个数为 N，则表示发生了 N 次采样事件后输入边沿才能通过滤波器： 0000: 无滤波器, 以 $f_{DTS}$ 采样率 $f_{SAMPLING} = f_{DTS}/8$ , $N=6$ 1000: 采样频率 $f_{SAMPLING} = f_{CK\_INT}$ , $N=2$ 0001: 采样频率 $f_{SAMPLING} = f_{DTS}/8$ , $N=8$ 1001 : 采样频率 $f_{SAMPLING} = f_{DTS}/8$ , $N=8$ 1010 : 0010: 采样频率 $f_{SAMPLING} = f_{CK\_INT}$ , $N=4$ 1011 : 采样频率 $f_{SAMPLING} = f_{DTS}/16$ , $N=5$ 0011: 采样频率 $f_{SAMPLING} = f_{CK\_INT}$ , $N=8$ 1011 : 采样频率 $f_{SAMPLING} = f_{DTS}/16$ , $N=6$ 0100: 采样频率 $f_{SAMPLING} = f_{DTS}/2$ , $N=6$ 1100 : 采样频率 $f_{SAMPLING} = f_{DTS}/16$ , $N=8$ 0101: 采样频率 $f_{SAMPLING} = f_{DTS}/2$ , $N=8$ 1101 : 采样频率 $f_{SAMPLING} = f_{DTS}/32$ , $N=5$ 0110: 采样频率 $f_{SAMPLING} = f_{DTS}/4$ , $N=6$ 1110: 采样频率 $f_{SAMPLING} = f_{DTS}/32$ , $N=6$ 0111: 采样频率 $f_{SAMPLING} = f_{DTS}/4$ , $N=8$ 1111 : 采样频率 $f_{SAMPLING} = f_{DTS}/32$ , $N=8$
位 19: 16	BKF	0x0	rw	输出使能 (output enable) 对配置为输出的通道, 该位用于使能 CxOUT 和 CxCOUT 的输出。 0: 关闭; 1: 开启。
位 15	OEN	0x0	rw	输出自动使能 (automatic output enable) 用于溢出事件时将 OEN 自动置'1' 0: 关闭; 1: 开启
位 14	AOEN	0x0	rw	刹车输入信号的有效性 (brake input validity) 用于选择刹车输入信号的输入有效电平: 0: 低电平; 1: 高电平。
位 13	BRKV	0x0	rw	刹车功能使能 (Brake enable) 用于开启刹车功能。 0: 关闭; 1: 开启。
位 12	BRKEN	0x0	rw	

位 11	FCSOEN	0x0	rw	总输出开时的冻结状态 (Frozen channel status when holistic output enable) 该位用于配置具有互补输出的通道，在定时器不工作且 OEN=1 时的通道状态。 0: 关闭 CxOUT/CxCOUT 输出； 1: 开启 CxOUT/CxCOUT 输出，输出为无效电平。
位 10	FCSODIS	0x0	rw	总输出关时的冻结状态 (Frozen channel status when holistic output disable) 该位用于配置具有互补输出的通道，在定时器不工作且 OEN=0 时的通道状态。 0 : 关闭 CxOUT/CxCOUT 输出； 1 : 开启 CxOUT/CxCOUT 输出，输出为空闲电平。
位 9: 8	WPC	0x0	rw	写保护配置 (Write protected configuration) 该位用于配置写保护。 00: 写保护关闭； 01: 3 级写保护，以下位受写保护： TMR1_BRK: DTC、BRKEN、BRKV 和 AOEN TMR1_CTRL2: CxIOS 和 CxCIOS 10: 2 级写保护，除 3 级写保护的内容外，以下位也受写保护： TMR1_CCTRL: CxP 和 CxCP TMR1_BRK: FCSODIS 和 FCSOEN 11: 1 级写保护，除 2 级写保护的内容外，以下位也受写保护： TMR1_CMx: C2OCTRL 和 C2OBEN 注：WPC>0 时将无法再次被修改，直到系统复位。
位 7: 0	DTC	0x0	rw	死区配置 (Dead-time configuration) 这些位用于配置死区时间。取 DTC[7: 0] 的高 3 位为功能选择位： 0xx: DT = DTC [7: 0] * TDTS; 10x: DT = (64+ DTC [5: 0]) * TDTS * 2; 110: DT = (32+ DTC [4: 0]) * TDTS * 8; 111: DT = (32+ DTC [4: 0]) * TDTS * 16;

注意：根据锁定设置，BKF、AOEN、BRKV、BRKEN、FCSODIS、FCSOEN 和 DTC[7: 0] 位均可被写保护，有必要在第一次写入 TMR15\_BRK 寄存器时对它们进行配置。

#### 14.4.4.16 TMR15 DMA控制寄存器 (TMR15\_DMACTRL)

域	简称	复位值	类型	功能
位 15: 13	保留	0x0	resd	保持默认值。
位 12: 8	DTB	0x0	rw	DMA 传输字节 (DMA transfer bytes) 这些位定义了传输的字节个数： 00000: 1 个字节      00001: 2 个字节 00010: 3 个字节      00011: 4 个字节 ..... 10000: 17 个字节      10001: 18 个字节
位 7: 5	保留	0x0	resd	保持默认值。
位 4: 0	ADDR	0x0	rw	DMA 传输地址偏移 (DMA transfer address offset) ADDR 定义了从 TMR15_CTRL1 所在地址开始的偏移量： 00000: TMR15_CTRL1, 00001: TMR15_CTRL2, 00010: TMR15_STCTRL, .....

#### 14.4.4.17 TMR15 DMA数据寄存器 (TMR15\_DMADT)

域	简称	复位值	类型	功能
位 15: 0	DMADT	0x0	rw	DMA 传输的数据寄存器 (DMA data register) 通过对 DMADT 寄存器的读写能够实现对任意 TMR 寄存器的操作，其操作的寄存器地址范围是：TMR15 外设地址 + ADDR*4 至 TMR15 外设地址 + ADDR*4 + DTB*4。

## 14.5 通用定时器（TMR16和TMR17）

### 14.5.1 TMR16和TMR17简介

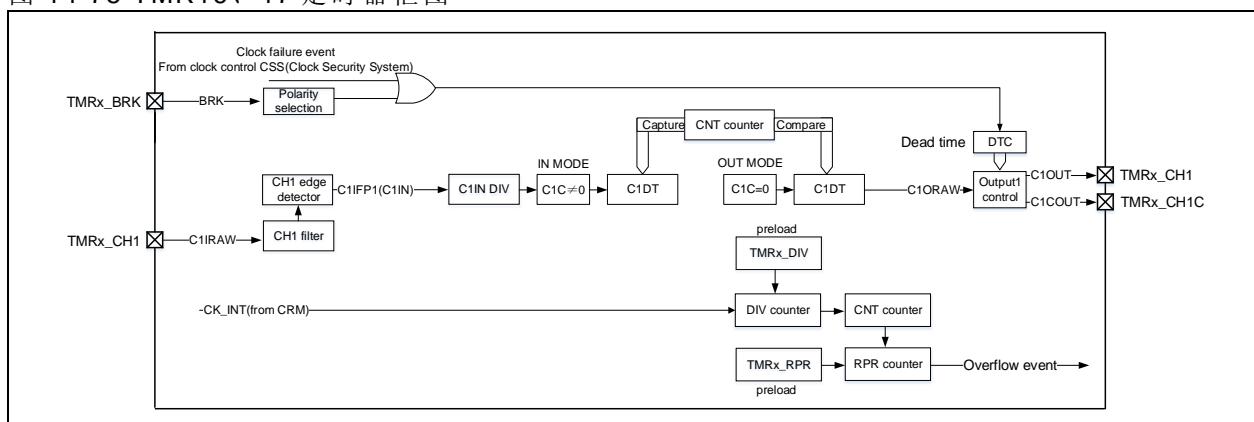
TMR16、17 包含一个支持向上计数的 16 位计数器、1 个捕获/比较寄存器、1 组独立的通道。可实现嵌入死区、输入捕获、可编程 PWM 输出。

### 14.5.2 TMR16和TMR17主要功能

TMR16、17 定时器的功能包括：

- 可选内部、外部、内部触发输入用作计数时钟
- 16位支持向上计数器、8位重复计数计数器
- 1组独立通道，支持输入捕获、输出比较、PWM生成、单周期模式、死区插入。
- 1组支持互补输出的独立通道
- 支持TMR刹车功能
- 定时器之间可互联同步
- 支持溢出事件、刹车输入、通道事件触发中断/DMA
- 支持TMR burst DMA传输

图 14-75 TMR16、17 定时器框图

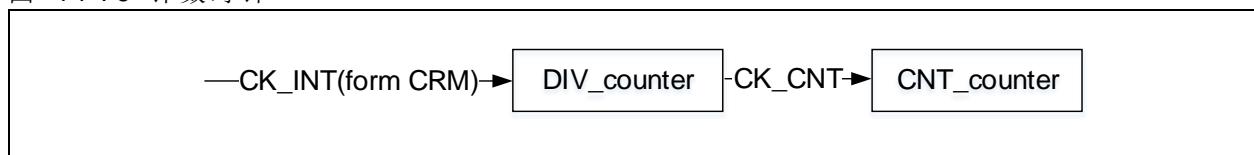


### 14.5.3 TMR16和TMR17功能描述

#### 14.5.3.1 计数时钟

TMR16、17 计数时钟仅能由内部时钟（CK\_INT）提供。

图 14-76 计数时钟



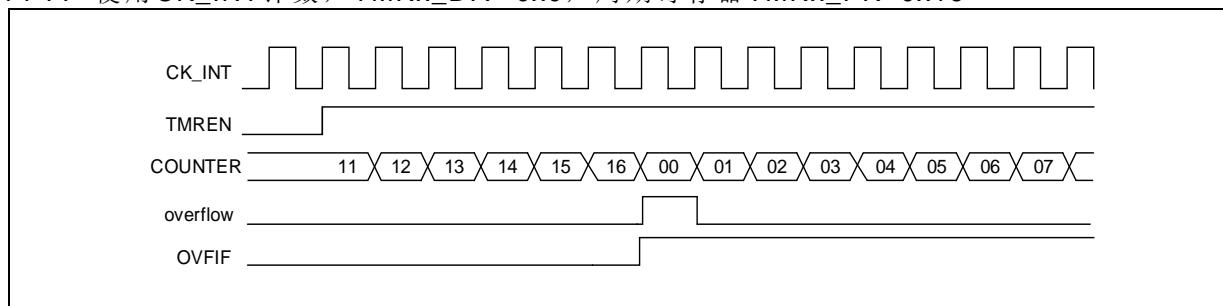
#### 内部时钟（CK\_INT）

默认下使用 CK\_INT 经由预分频器驱动计数器计数，当 TMR 对应的 APB 时钟预分频系数是 1 时，CK\_INT 频率等于 APB 时钟频率，否则 CK\_INT 频率等于 APB 时钟频率的 2 倍。相关配置流程如下：

- 配置 TMRx\_DIV 寄存器，设置计数器计数频率。
- 配置 TMRx\_PR 寄存器，设置计数器计数周期。

-配置TMRx\_CTRL1寄存器TMREN，使能计数器。

图14-77 使用CK\_INT计数，TMRx\_DIV=0x0，周期寄存器TMRx\_PR=0x16



### 14.5.3.2 计数模式

TMR16、17 提供了多种计数模式，用来满足不同的应用场景。其内部拥有一个支持 16 位向上计数的计数器。

TMRx\_PR 寄存器用于设置计数器计数周期。默认 TMRx\_PR 寄存器值会立即传入它的影子寄存器；当开启周期缓冲功能后（PRBEN 置 1），TMRx\_PR 寄存器值在溢出事件发生时传入它的影子寄存器。

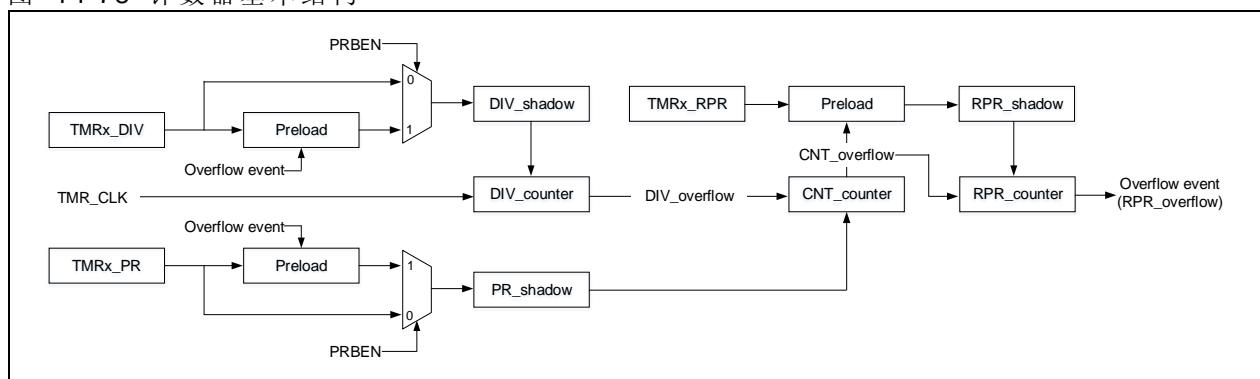
TMRx\_DIV 寄存器用于设置计数器计数频率，每（DIV[15:0]+1）个计数时钟周期，计数器计数一次。和 TMRx\_PR 寄存器类似，开启周期缓冲功能后，TMRx\_DIV 寄存器值在溢出事件时更新至它的影子寄存器。

读取 TMRx\_CNT 寄存器会返回当前计数器计数值，写入 TMRx\_CNT 寄存器会更新计数器当前计数值为写入值。

默认允许产生溢出事件，设置 TMRx\_CTRL1 寄存器 OVFS=1 将禁止更新事件产生。TMRx\_CTRL1 寄存器 OVFS 用于选择溢出事件来源，默认计数器上溢或下溢、置位 OVFSWTR、复位模式次定时器控制器产生的复位信号产生溢出事件。置位 OVFS 后，只有计数器上溢或下溢产生溢出事件。

TMREN 位置 1 将使能定时器计数，由于同步逻辑，实际驱动计数器的使能信号 TMR\_EN 相对于 TMREN 延迟一个时钟周期。

图 14-78 计数器基本结构



#### 向上计数模式

配置 TMRx\_CTRL1 寄存器 CMSEL[1:0]=2'b00，OWCDIR=1'b0 开启向上计数模式，计数值达到 TMRx\_PR 值时，重新从 0 向上计数，计数器上溢并产生溢出事件，同时 OVFIF 位置 1。若禁止产生溢出事件，计数器溢出后不再重载预分频值和周期值，否则预分频值和周期值在溢出事件后更新。

图 14-79 PRBEN=0 时的溢出事件

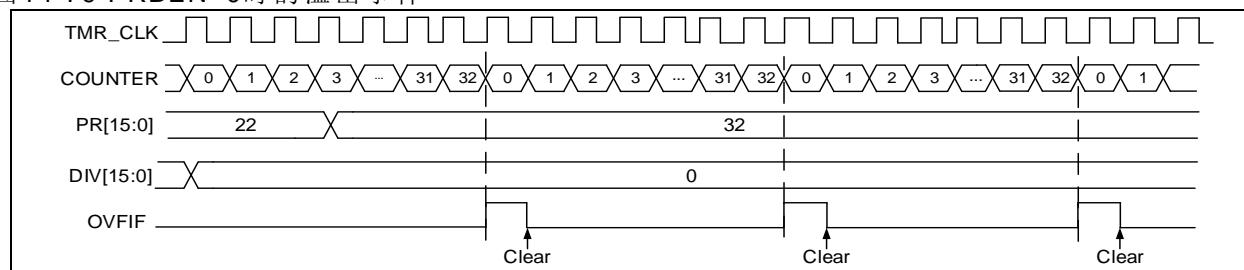
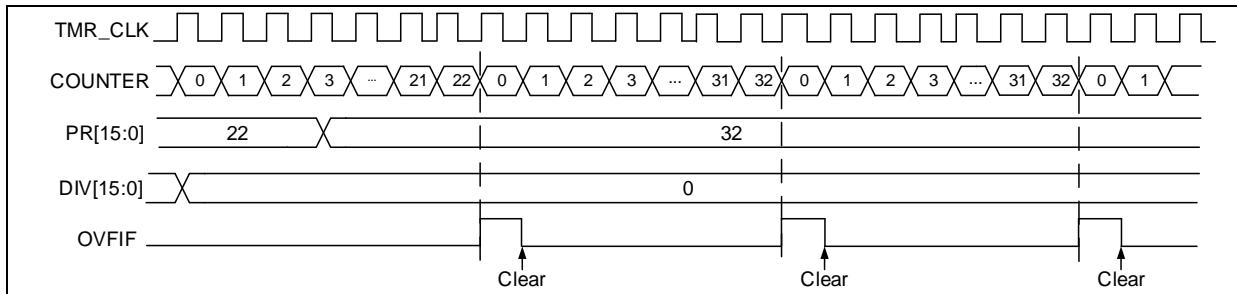
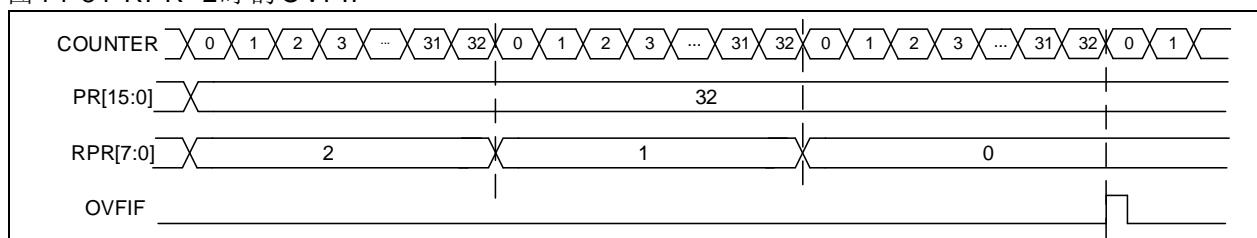


图 14-80 PRBEN=1 时的溢出事件

**重复计数模式：**

TMRx\_RPR 寄存器用于配置重复计数器计数周期，TMRx\_RPR 寄存器为非 0 值时，重复计数模式启动。重复计数模式下，每 ( $RPR[7:0]+1$ ) 次计数器溢出将产生一次溢出事件。每次计数器溢出，重复计数器递减，仅当重复计数器计数值等于 0 值时，计数器溢出会产生溢出事件。通过配置不同重复计数器值，可调整溢出事件产生的频率。

图 14-81 RPR=2 时的 OVFIF

**14.5.3.3 TMR 输入部分**

TMR16、17 拥有 1 个独立通道，可配置为输入或输出，当配置位输入时，每个通道输入信号依次经过以下处理：

- TMRx\_CHx 经过预处理输出 CxIRAW。配置 C1INSEL 位，选择 CxIRAW 来源是 TMRx\_CHx。
- CxIRAW 输入数字滤波器，输出滤波后信号 CxIF。数字滤波器通过 CxDI 位配置采样频率和次数。
- CxIF 输入边沿检测器，输出边沿选择后信号 CxIFPx。边沿选择由 CxP 和 CxCP 位共同控制，可选择输入上升沿、下降沿或双边沿有效。
- CxIFPx 输入捕获信号选择器，输出选择后信号 CxIN。捕获信号选择器由 CxC 控制，可选择 CxIN 来源为 CxIFPx。其中 CyIFPx ( $x \neq y$ ) 是来自通道 y 的 CyIFPy 经通道 x 边沿检测器处理后的信号（例如 C1IFP2 是来自通道 1 的 C1IFP1 信号经过通道 2 边沿检测器处理后的信号）。
- CxIN 经由输入通道分频器，输出分频后信号 CxIPS。分频系数由 CxIDIV 位配置为不分频、2 分频、4 分频或 8 分频。

图 14-82 输入/输出通道 1 的主电路

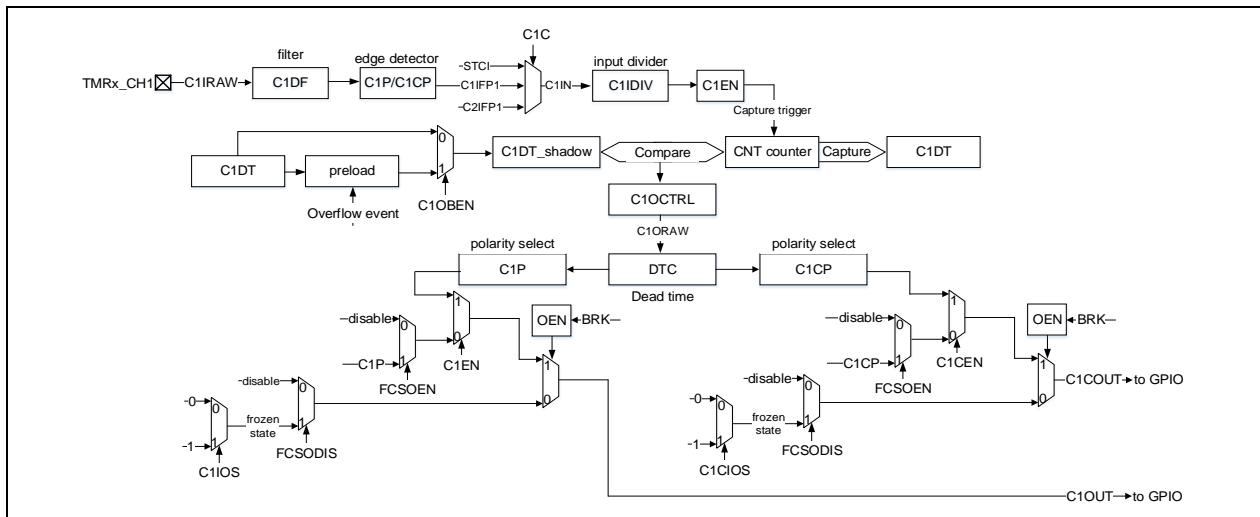
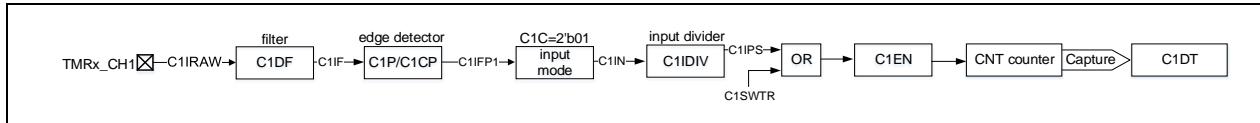


图 14-83 通道 1 输入部分



### 输入模式

此模式下，当选中的触发信号被检测到，通道寄存器（TMRx\_CxDT）记录当前计数器计数值，并将捕获比较中断标志位（CxIF）置 1，若已使能通道中断（CxIEN）、通道 DMA 请求（CxDEN）则产生相应的中断和 DMA 请求。若在 CxIF 置 1 后检测到触发信号，将产生捕获溢出事件，TMRx\_CxDT 会使用当前计数器计数值覆盖之前记录的计数器计数值，同时通道再捕获标志位（CxRF）置 1。

以若要捕获 C1IN 输入的上升沿，可按如下进行配置：

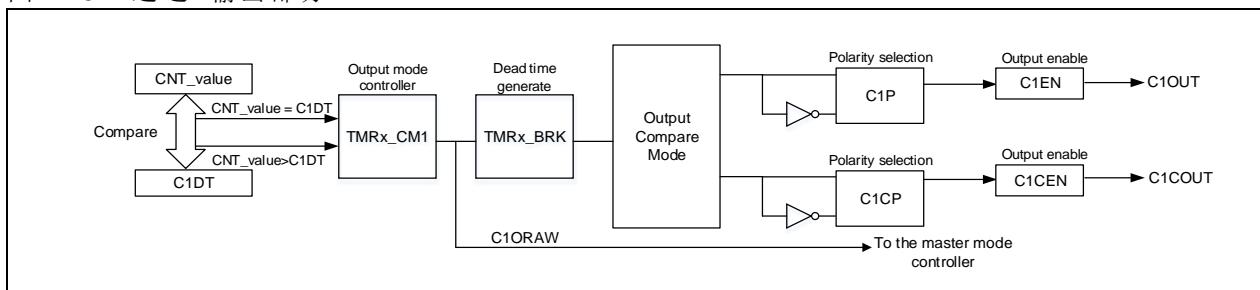
- 将通道模式寄存器1（TMRx\_CM1）中的C1C位配置为01，选择C1IN作为通道1输入。
- 配置C1IN信号滤波器带宽（CxDF[3:0]）。
- 配置C1IN通道的有效沿，在TMRx\_CCTRL寄存器中写入C1P=0（上升沿）。
- 配置C1IN信号捕获分频（C1DIV[1:0]）。
- 使能通道1输入捕获（C1EN=1）。

根据需要设置 TMRx\_IDEN 寄存器中的 C1IEN 为、TMRx\_IDEN 寄存器中的 C1DEN 位，选择中断请求或 DMA 请求。

### 14.5.3.4 TMR输出部分

TMR 的输出部分由比较器和输出控制构成，用于编程输出信号的周期、占空比、极性。输出部分如下图所示：

图 14-84 通道 1 输出部分



### 输出模式

配置  $Cx[1:0]\neq 2'b00$  将通道配置为输出可实现多种输出模式，此时，计数器计数值将与 CxDT 寄存器值比较，并根据 CxOCTRL[2:0] 位配置的输出模式，产生中间信号 CxORAW，再经过输出控制逻辑处理后输送到 IO。输出信号的周期由 TMRx\_PR 寄存器值配置，占空比则由 CxDT 寄存器值配置。

输出比较模式有以下子类：

**PWM 模式 A:** CxOCTRL=3'b110 时，开启 PWM 模式 A。向上计数时， $TMRx\_C1DT > TMRx\_CVAL$

时 C1ORAW 输出高电平，否则为低电平；向下计数时， $\text{TMRx\_C1DT} < \text{TMRx\_CVAL}$  时 C1ORAW 输出低电平，否则为高电平。若要使用 PWM 模式 A，可按如下方式配置。

- 配置 TMRx\_PR 寄存器，设置 PWM 周期。
- 配置 TMRx\_CxDT 寄存器，设置 PWM 占空比。
- 配置 TMRx\_CM1/CM2 寄存器 CxOCTRL 位为 3'b110，设置输出模式为 PWM 模式 A。
- 配置 TMRx\_DIV 寄存器，设置计数器计数频率。
- 配置 TMRx\_CTRL1 寄存器 TWCMSEL[1:0]位，设置计数器计数模式。
- 配置 TMRx\_CCTRL 寄存器 CxP 位、CxCP 位，设置输出极性。
- 配置 TMRx\_CCTRL 寄存器 CxEN 位、CxCEN 位，使能通道输出。
- 配置 TMRx\_BRK 寄存器 OEN 位，使能 TMRx 输出。
- 配置 TMR 输出通道对应 GPIO 为对应的复用模式。
- 配置 TMRx\_CTRL1 寄存器 TMREN 位，使能 TMRx 计数。

**PWM 模式 B：** CxOCTRL=3'b111 时，开启 PWM 模式 B。向上计数时， $\text{TMRx\_C1DT} > \text{TMRx\_CVAL}$  时 C1ORAW 输出低电平，否则为高电平；向下计数时， $\text{TMRx\_C1DT} < \text{TMRx\_CVAL}$  时 C1ORAW 输出高电平，否则为低电平。

**强制输出模式：** CxOCTRL=3'b100/101 时，开启强制输出模式。此时，CxORAW 信号的电平被强制输出为配置的电平，而与计数值无关。虽然输出信号不依赖于比较结果，但通道标志位和 DMA 请求仍依赖于比较结果。

**输出比较模式：** CxOCTRL=3'b001/010/011 时，开启输出比较模式。此时，当计数值与 CxDT 值匹配时，CxORAW 强制输出高电平（CxOCTRL=3'b001）、低电平（CxOCTRL=3'b010）或进行电平翻转（CxOCTRL=3'b011）。

**单周期模式：** PWM 模式的特例，将 OCMEN 位置 1 可开启单周期模式，此模式下，仅在当前计数周期中进行比较匹配，完成当前计数后，TMREN 位清 0，因此仅输出一个脉冲。当配置为向上计数模式时，需要严格配置 CVAL<CxDT≤PR；向下计数时，需严格配置 CVAL>CxDT。

**快速输出模式：** 将 CxOIEN 位置 1 可开启此功能，开启后 CxORAW 电平值不再在计数值与 CxDT 匹配时变化，而是在当前计数周期开始时，也就是说，比较结果被提前了，计数器值与 CxDT 寄存器的比较结果将会提前决定 CxORAW 的电平。

图 14-85 展示了输出比较模式（翻转）的例子，C1DT=0x3，当计数值等于 0x3 时，输出电平 C1OUT 被翻转。

图 14-86 展示了计数器向上计数与 PWM 模式 A 配合的例子，PR=0x32，CxDT 配置为不同的值时输出时输出信号的翻转情况。

图 14-87 展示了计数器向上计数与单周期模式下 PWM 模式 B 配合的例子，计数器仅计数了一个周期，输出信号在这个周期中只输出了一个脉冲。

图 14-85 计数值与C1DT值匹配时翻转C1ORAW

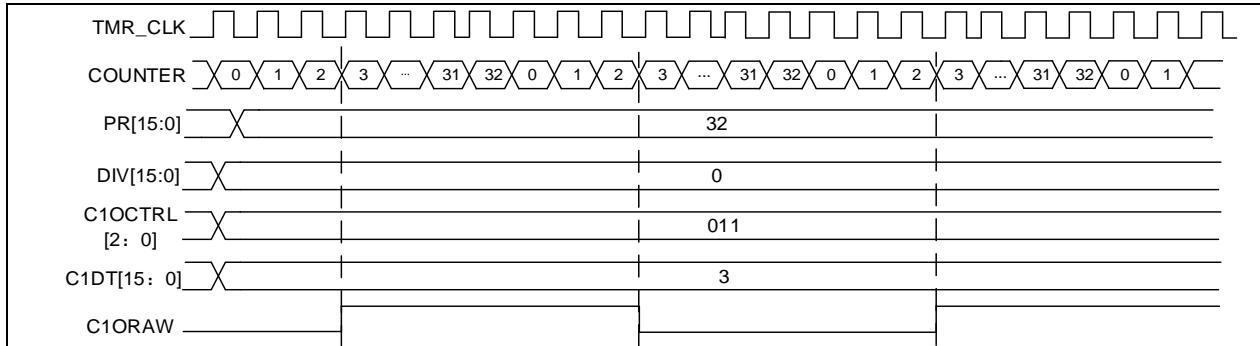


图 14-86 向上计数下 PWM 模式 A

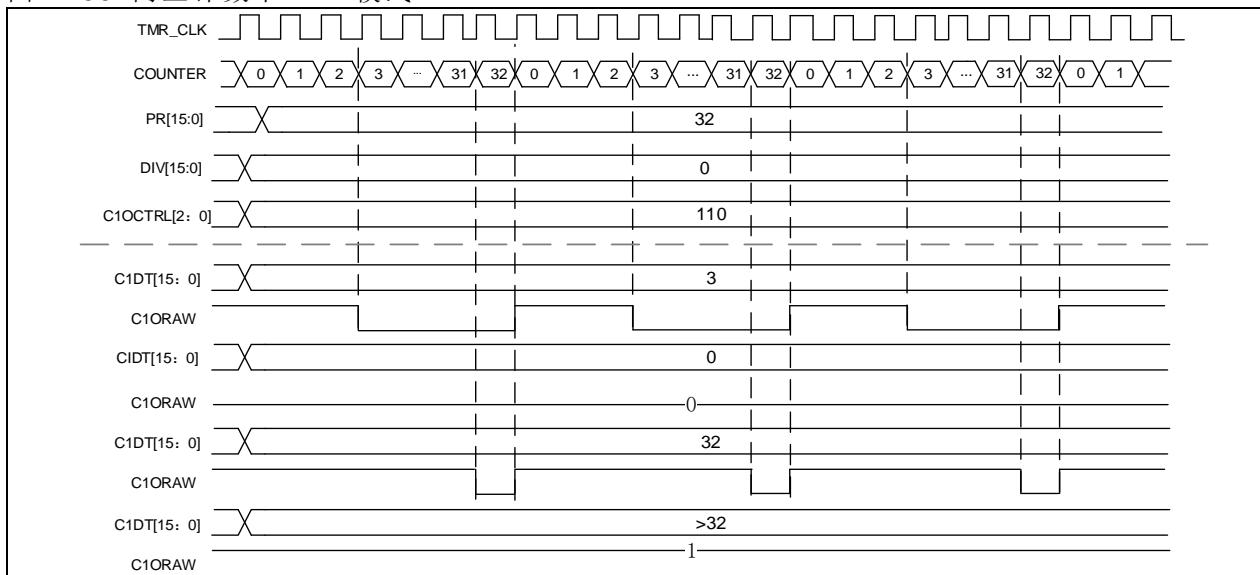
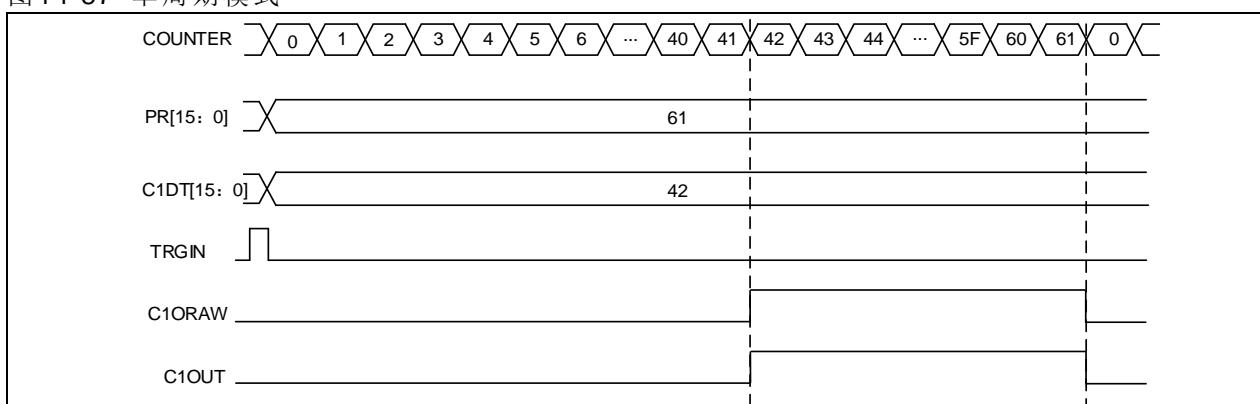


图14-87 单周期模式



## 死区插入

TMR16、17 通道 1 包含一组反向通道输出，通过 CxCEN 使能，通过 CxCP 配置极性。CxOUT 和 CxCOUT 的输出状态见表 14-13。

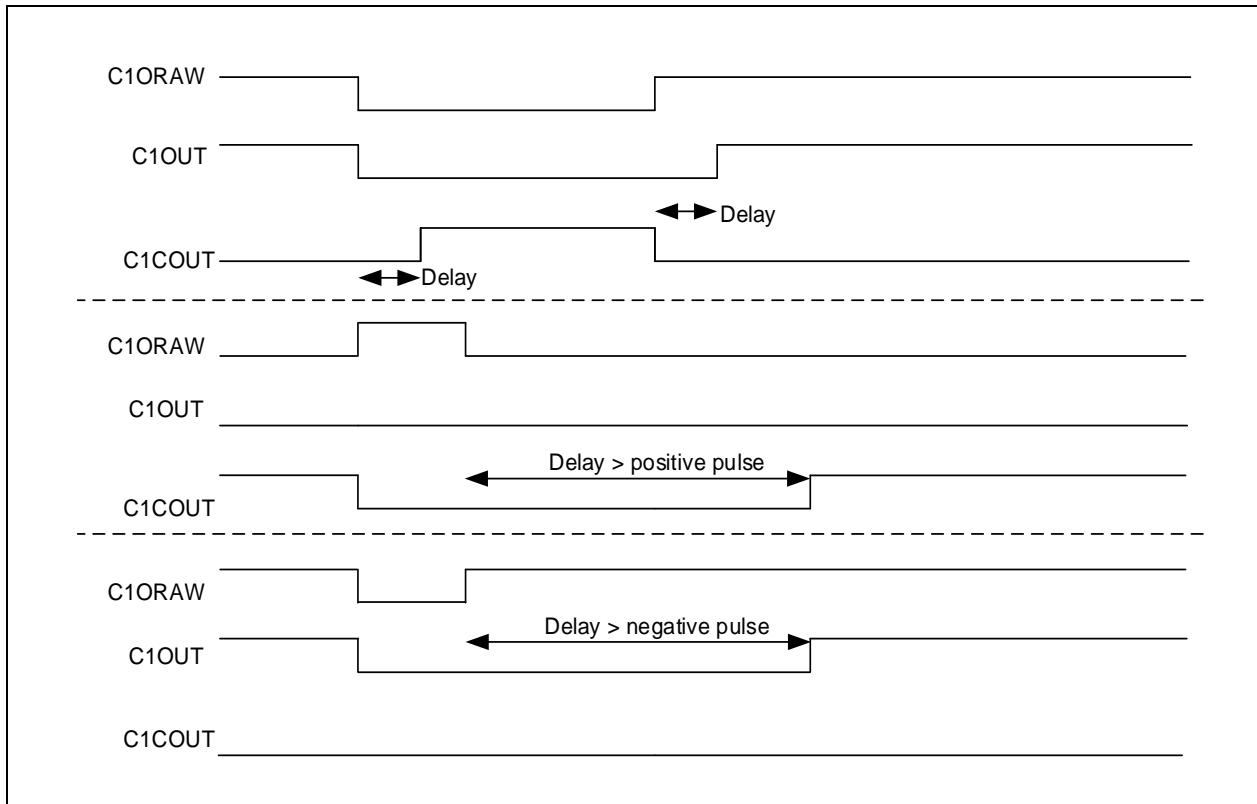
当转换为 IDLEF 状态，即 OEN 下降到 0，死区被激活。

将 CxEN 和 CxCEN 位置 1 后，通过配置 DTC[7:0]死区发生器，可插入不同时间长的死区。插入死区后，CxDOUT 的上升沿延迟于参考信号的上升沿；CxQDOUT 的上升沿延迟于参考信号的下降沿。

如果延迟大于当前有效的输出宽度, C1OUT 和 C1COUT 不会产生相应的脉冲, 死区时间应小于有效的输出宽度。

下列图展示了  $CxP=0$ ,  $CxCP=0$ ,  $QEN=1$ ,  $CxEN=1$  并且  $CxCEN=1$  时死区插入的例子。

图 14-88 带死区插入的互补输出



#### 14.5.3.5 TMR刹车功能

开启刹车功能后 (BRKEN 位置 1)，CxOUT 和 CxCOUT 由 OEN、FCSODIS、FCISOEN、CxIOS 和 CxCIOS 共同控制。但 CxOUT 和 CxCOUT 输出总是不能同时处于有效电平上的。详见表 14-13 带刹车功能的互补输出通道 CxOUT 和 CxCOUT 的控制位。

刹车信号来源可以是刹车输入引脚、时钟失效事件，刹车输入信号的极性由 BRKV 位控制。

当发生刹车事件时，有下述动作：

- OEN位异步清零，通道输出状态由FCSODIS位选择。关闭MCU的振荡器不影响该功能。
- OEN被清零后，通道输出电平由CxIOS位设定。如果FCSODIS=0，则定时器输出使能被禁止，否则输出使能始终为高。
- 当使用互补输出时：
  - 输出最开始处于复位状态，也就是无效的状态（取决于极性）。这是异步操作，定时器有无时钟并不影响此功能。
  - 定时器的时钟如果有效，会开启死区生成功能，CxIOS和CxCIOS位用来配置死区之后的电平。即使在这种情况下，CxOUT和CxCOUT也不能被同时驱动到有效的电平。
 注意，由于 OEN 位同步逻辑，死区时间较通常情况会延长一段时间（大约 2 个 clk\_tmr 的时钟周期）。
- 如果 FCSODIS=0，定时器释放使能输出，否则保持使能输出；或一旦 CxEN 与 CxCEN 之一变高时，使能输出变为高。
- 如果开启了刹车中断或DMA功能，刹车状态标志将置1，并产生刹车中断或DMA请求。
- 如果将AOEN位置1，在下一个溢出事件时OEN位被自动置1。

注意：刹车输入电平有效时，OEN 不能被设置，状态标志 BRKIF 也不能被清除。

图 14-89 TMR输出控制

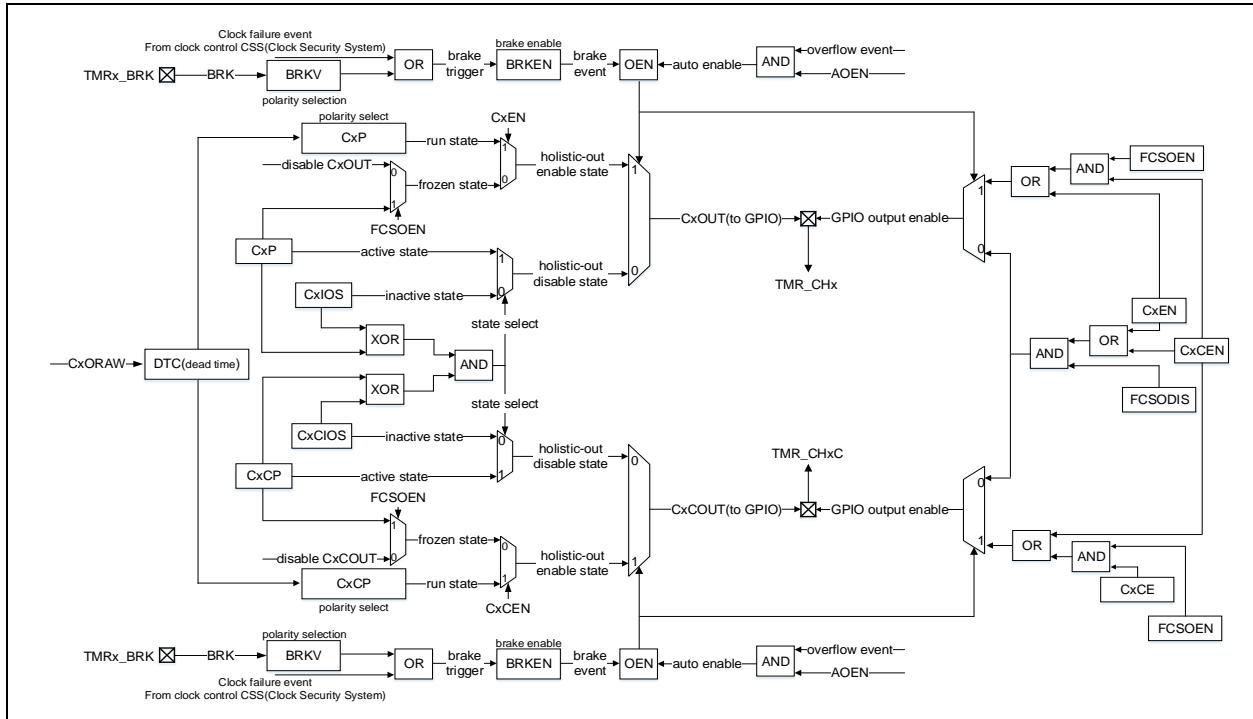
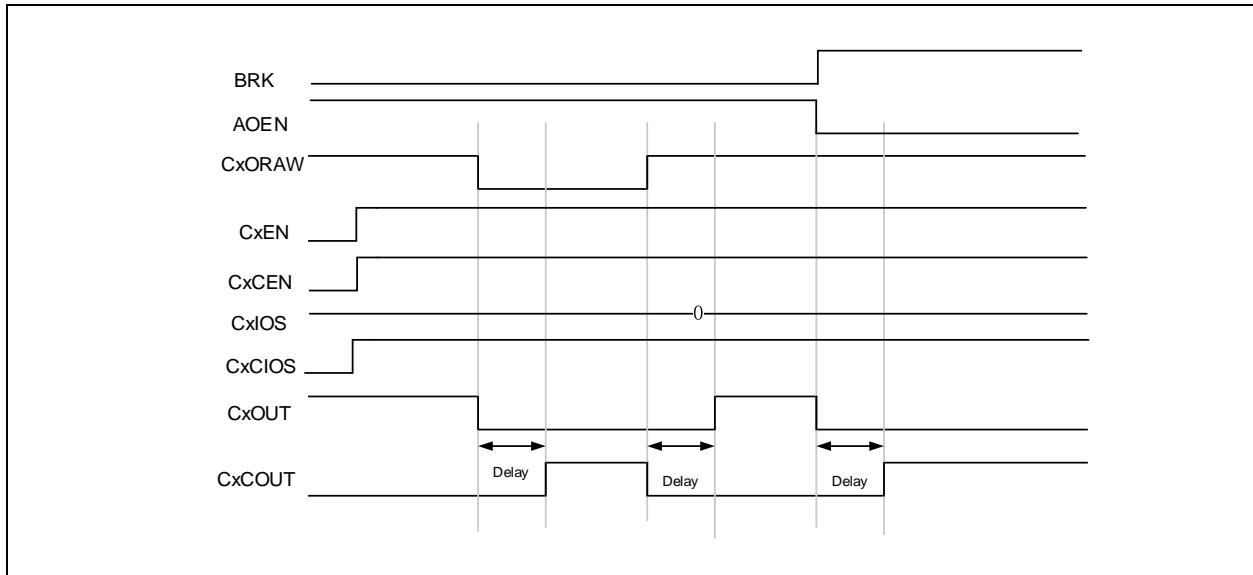


图 14-90 TMR刹车功能的例子



### 14.5.3.6 调试模式

当微控制器进入调试模式（Cortex®-M4 核心停止）时，将 DEBUG 模块中的 TMRx\_PAUSE 置 1，可以使 TMRx 计数器暂停计数。

## 14.5.4 TMR16和TMR17寄存器描述

必须以字（32位）的方式操作这些外设寄存器。

下表中将 TMR16、17 的所有寄存器映射到一个 16 位可寻址（编址）空间

表 14-12 TMR16、17 寄存器和复位值

寄存器简称	基址偏移量	复位值
TMRx_CTRL1	0x00	0x0000
TMRx_CTRL2	0x04	0x0000
TMRx_IDEN	0x0C	0x0000
TMRxISTS	0x10	0x0000
TMRx_SW_EVT	0x14	0x0000
TMRx_CM1	0x18	0x0000
TMRx_CCTRL	0x20	0x0000
TMRx_CVAL	0x24	0x0000
TMRx_DIV	0x28	0x0000
TMRx_PR	0x2C	0x0000
TMRx_RPR	0x30	0x0000
TMRx_C1DT	0x34	0x0000
TMRx_BRK	0x44	0x0000
TMRx_DMACTRL	0x48	0x0000
TMRx_DMADT	0x4C	0x0000

### 14.5.4.1 TMR16和TMR17控制寄存器1 (TMRx\_CTRL1)

域	简称	复位值	类型	功能
位 15: 10	保留	0x0	resd	保持默认值。
位 9: 8	CLKDIV	0x0	rw	时钟除频 (Clock divider) 此位用于设置数字滤波器采样频率 $f_{DTS}$ 和定时器时钟频率 $f_{CK\_INT}$ 之间的分频比，也用于调整死区时间的时基 $T_{DTS}$ 和定时器时钟周期 $T_{CK\_INT}$ 的分频比。 00: 无除频, $f_{DTS}=f_{CK\_INT}$ ; 01: 2 除频, $f_{DTS}=f_{CK\_INT}/2$ ; 10: 4 除频, $f_{DTS}=f_{CK\_INT}/4$ ; 11: 保留。
位 7	PRBEN	0x0	rw	周期缓冲使能 (Period buffer enable) 0: 缓冲关闭; 1: 缓冲开启。
位 6: 4	保留	0x0	resd	保持默认值
位 3	OCMEN	0x0	rw	单周期使能 (One cycle mode enable) 该功能用于选择溢出事件后，计数器是否停止。 0: 关闭; 1: 开启。
位 2	OVFS	0x0	rw	溢出事件源选择 (Overflow event source) 配置溢出事件或 DMA 请求来源。 0: 来源于计数器溢出、设置 OVFSWTR 位或次定时器控制器产生的溢出事件; 1: 只能来源于计数器溢出。
位 1	OVFEN	0x0	rw	溢出事件使能 (Overflow event enable) 0: 开启; 1: 关闭。
位 0	TMREN	0x0	rw	使能定时器 (TMR enable) 0: 关闭; 1: 开启。

#### 14.5.4.2 TMR16和TMR17控制寄存器2 (TMRx\_CTRL2)

域	简称	复位值	类型	功能
位 30: 10	保留	0x0	resd	保持默认值
位 9	C1CIOS	0x0	rw	通道 1 互补空闲输出状态 (Channel 1 complementary idle output state) 输出关闭 (OEN = 0), 死区发生后: 0: C1COUT=0; 1: C1COUT=1。
位 8	C1IOS	0x0	rw	通道 1 空闲输出状态 (Channel 1 idle output state) 输出关闭 (OEN = 0), 死区发生后: 0: C1OUT=0。 1: C1OUT=1。
位 7: 4	保留	0x0	resd	保持默认值
位 3	DRS	0x0	rw	DMA 请求源 (DMA request source) DMA 请求来源。 0: 通道事件; 1: 溢出事件。
位 2	CCFS	0x0	rw	通道控制位刷新选择 (Channel control bit refresh select) 对具有互补输出的通道, 如果通道控制位有缓存时: 0: 通过设置 HALL 位刷新控制位; 1: 通过设置 HALL 位或 TRGIN 的上升沿刷新控制位。
位 1	保留	0x0	resd	保持默认值。
位 0	CBCTRL	0x0	rw	通道缓存控制 (Channel buffer control) 对具有互补输出的通道: 0: CxEN, CxCEN 和 CxOCTRL 位无缓存; 1: CxEN, CxCEN 和 CxOCTRL 位有缓存。

#### 14.5.4.3 TMR16和TMR17 DMA/中断使能寄存器 (TMRx\_IDEN)

域	简称	复位值	类型	功能
位 15: 10	保留	0x0	resd	保持默认值。
位 9	C1DEN	0x0	rw	通道 1 的 DMA 请求使能 (Channel 1 DMA request enable) 0: 关闭; 1: 开启。
位 8	OVFDEN	0x0	rw	溢出事件的 DMA 请求使能 (overflow event DMA request enable) 0: 关闭; 1: 开启。
位 7	BRKIE	0x0	rw	刹车中断使能 (Brake interrupt enable) 0: 关闭; 1: 开启。
位 6	保留	0x0	resd	保持默认值
位 5	HALLIEN	0x0	rw	HALL 中断使能 (HALL interrupt enable) 0: 关闭; 1: 开启。
位 4: 2	保留	0x0	resd	保持默认值
位 1	C1IEN	0x0	rw	通道 1 中断使能 (Channel 1 interrupt enable) 0: 关闭; 1: 开启。
位 0	OVFIEN	0x0	rw	溢出中断使能 (Overflow interrupt enable) 0: 关闭; 1: 开启。

#### 14.5.4.4 TMR16和TMR17中断状态寄存器 (TMRx\_ISTS)

域	简称	复位值	类型	功能
位 15: 10	保留	0x0	resd	保持默认值。
位 9	C1RF	0x0	rw0c	通道 1 再捕获标记 (Channel 1 recapture flag) C1IF 的状态已经为'1'时是否再次发生了捕获, 由硬件置'1', 写'0'清除。 0: 无捕获发生; 1: 捕获发生。
位 8	保留	0x0	resd	保持默认值。
位 7	BRKIF	0x0	rw0c	刹车中断标记 (Brake interrupt flag) 用于标记刹车输入的电平是否有效, 由硬件置'1', 写'0'清除。 0: 无效; 1: 有效。
位 6	保留	0x0	resd	保持默认值
位 5	HALLIF	0x0	rw0c	HALL 中断标记 (HALL interrupt flag) 当发生触发事件时由硬件置'1', 写'0'清除。 0: 无 HALL 事件发生; 1: 发生 HALL 事件。 HALL 事件: CxEN、CxCEN、CxOCTRL 已被更新。
位 4: 2	保留	0x0	resd	保持默认值
位 1	C1IF	0x0	rw0c	通道 1 中断标记 (Channel 1 interrupt flag) 若通道 1 为输入模式时: 捕获事件发生时由硬件置'1', 由软件清'0'或读 TMRx_C1DT 清'0'. 0: 无捕获事件发生; 1: 发生捕获事件。 若通道 1 为输出模式时: 比较事件发生时由硬件置'1', 由软件清'0'. 0: 无比较事件发生; 1: 发生比较事件。
位 0	OVFIF	0x0	rw0c	溢出中断标记 (Overflow interrupt flag) 当溢出事件发生时由硬件置'1', 由软件清'0'. 0: 无溢出事件发生; 1: 发生溢出事件, 若 TMRx_CTRL1 的 OVFEN=0、OVFS=0 时: - 当 TMRx_SWEVE 寄存器的 OVFG=1 时产生溢出事件; - 当计数值 CVAL 被触发事件重初始化时产生溢出事件。

#### 14.5.4.5 TMR16和TMR17软件事件寄存器 (TMRx\_SWEVT)

域	简称	复位值	类型	功能
位 15: 8	保留	0x0	resd	保持默认值。
位 7	BRKSWTR	0x0	wo	软件触发刹车事件 (Brake event triggered by software) 通过软件触发一个刹车事件。 0: 无作用; 1: 制造一个刹车事件。
位 6	保留	0x0	resd	保持默认值
位 5	HALLSWTR	0x0	wo	软件触发 HALL 事件 (HALL event triggered by software) 通过软件产生一个 HALL 事件。 0: 无作用; 1: 产生一个 HALL 事件。 注: 该位只对拥有互补输出的通道有效。
位 4: 2	保留	0x0	resd	保持默认值
位 1	C1SWTR	0x0	wo	C1SWTR: 软件触发通道 1 事件 (Channel 1 event triggered by software) 通过软件触发一个通道 1 事件。 0: 无作用;

位 0	OVFSWTR	0x0	wo	1: 制造一个通道 1 事件。 软件触发溢出事件 (Overflow event triggered by software) 通过软件触发一个溢出事件。 0: 无作用; 1: 制造一个溢出事件。
-----	---------	-----	----	--

#### 14.5.4.6 TMR16和TMR17通道模式寄存器1 (TMRx\_CM1)

通道可用于输入（捕获模式）或输出（比较模式），通道的方向由相应的 CxC 位定义。该寄存器其它位的作用在输入和输出模式下不同。CxOx 描述了通道在输出模式下的功能，CxIx 描述了通道在输入模式下的功能。因此必须注意，同一个位在输出模式和输入模式下的功能是不同的。

##### 输出比较模式

域	简称	复位值	类型	功能
位 15: 8	保留	0x0	resd	保持默认值
位 7	C1OSEN	0x0	rw	通道 1 输出开关使能 (Channel 1 output switch enable) 0: EXT 输入不影响 C1ORAW; 1: 当 EXT 输入高电平时，将 C1ORAW 清 0.
位 6: 4	C1OCTRL	0x0	rw	通道 1 输出控制 (Channel 1 output control) 这些位用于设置原始信号 C1ORAW 的工作状态。 000: 断开。断开 C1ORAW 到 C1OUT 的输出; 001: 设置 C1ORAW 为高: TMRx_CVAL=TMRx_C1DT 时。 010: 设置 C1ORAW 为低: TMRx_CVAL=TMRx_C1DT 时。 011 : 切换 C1ORAW 的电平 : 当 TMRx_CVAL=TMRx_C1DT 时。 100: 固定 C1ORAW 为低。 101: 固定 C1ORAW 为高。 110: PWM 模式 A – OWCDIR=0, 若 TMRx_C1DT>TMRx_CVAL 时设置 C1ORAW 为高, 否则为低; – OWCDIR=1, 若 TMRx_C1DT < TMRx_CVAL 时设置 C1ORAW 为低, 否则为高。 111: PWM 模式 B – OWCDIR=0, 若 TMRx_C1DT > TMRx_CVAL 时设置 C1ORAW 为低, 否则为高; – OWCDIR=1, 若 TMRx_C1DT < TMRx_CVAL 时设置 C1ORAW 为高, 否则为低。 注: 除'000'外, 其余配置下 C1OUT 将连接到 C1ORAW, C1OUT 的输出电平除了会根据 C1ORAW 变化外, 还与 CCTRL 所配置的输出极性有关。
位 3	C1OBEN	0x0	rw	通道 1 输出缓存使能 (Channel 1 output buffer enable) 0: 关闭 TMRx_C1DT 的缓存功能, 写入 TMRx_C1DT 的内容会立即生效。 1: 启用 TMRx_C1DT 的缓存功能, 写入 TMRx_C1DT 的内容将保存到缓存寄存器中, 当发生溢出事件时再更新到 TMRx_C1DT 中。
位 2	C1OIEN	0x0	rw	通道 1 输出立即使能 (Channel 1 output immediately enable) 在 PWM 模式 A 或模式 B 下, 该位能够缩短触发事件到通道 1 的输出响应间的时间。 0: 需要比较 CVAL 与 C1DT 的值之后再产生输出。 1: 无需比较 CVAL 与 C1DT 的值, 当发生触发事件时立即产生输出。
位 1: 0	C1C	0x0	rw	通道 1 配置 (Channel 1 configure) 当 C1EN=0'时, 这些位用于选择通道 1 为输出或输入, 以及输入时的映射选择: 00: 输出; 01: 输入, C1IN 映射在 C1IFP1 上; 10: 保留;

11: 保留。

### 输入模式

域	简称	复位值	类型	功能
位 15: 8	保留	0x0	resd	保持默认值 通道 1 滤波器 (Channel 1 digital filter) 这些位用于配置通道 1 的滤波器。滤波的个数为 N，则表示发生了 N 次采样事件后输入边沿才能通过滤波器： 0000: 无滤波器，以 $f_{DTS}$ 采样                          1000: 采样频率 $f_{SAMPLING} = f_{DTS}/8, N=6$ 0001: 采样频率 $f_{SAMPLING} = f_{CK\_INT}, N=2$ 1001 : 采样频率 $f_{SAMPLING} = f_{DTS}/8, N=8$ 0010: 采样频率 $f_{SAMPLING} = f_{CK\_INT}, N=4$ 1010 : 采样频率 $f_{SAMPLING} = f_{DTS}/16, N=5$ 0011: 采样频率 $f_{SAMPLING} = f_{CK\_INT}, N=8$ 1011 : 采样频率 $f_{SAMPLING} = f_{DTS}/16, N=6$ 0100: 采样频率 $f_{SAMPLING} = f_{DTS}/2, N=6$ 1100 : 采样频率 $f_{SAMPLING} = f_{DTS}/16, N=8$ 0101: 采样频率 $f_{SAMPLING} = f_{DTS}/2, N=8$ 1101 : 采样频率 $f_{SAMPLING} = f_{DTS}/32, N=5$ 0110: 采样频率 $f_{SAMPLING} = f_{DTS}/4, N=6$ 1110: 采样频率 $f_{SAMPLING} = f_{DTS}/32, N=6$ 0111: 采样频率 $f_{SAMPLING} = f_{DTS}/4, N=8$ 1111 : 采样频率 $f_{SAMPLING} = f_{DTS}/32, N=8$
位 7: 4	C1DF	0x0	rw	采样频率 $f_{SAMPLING} = f_{DTS}/32, N=8$
位 3: 2	C1IDIV	0x0	rw	通道 1 分频系数 (Channel 1 input divider) 这些位定义了通道 1 的分频系数。 00: 不分频，每一个有效的边沿都会产生一次输入； 01: 每 2 个有效的边沿产生一次输入； 10: 每 4 个有效的边沿产生一次输入； 11: 每 8 个有效的边沿产生一次输入。 注：C1EN='0'时，分频系数复位。
位 1: 0	C1C	0x0	rw	通道 1 配置 (Channel 1 configure) 当 C1EN='0'时，这些位用于选择通道 1 为输出或输入，以及输入时的映射选择： 00: 输出； 01: 输入，C1IN 映射在 C1IFP1 上； 10: 保留； 11: 保留。

### 14.5.4.7 TMR16和TMR17通道控制寄存器 (TMRx\_CCTRL)

域	简称	复位值	类型	功能
位 15: 4	保留	0x0	resd	保持默认值。
位 3	C1CP	0x0	rw	通道 1 互补极性 (Channel 1 complementary polarity) 0: C1COUT 的有效电平为高 1: C1COUT 的有效电平为低
位 2	C1CEN	0x0	rw	通道 1 互补使能 (Channel 1 complementary enable) 0: 禁止输出； 1: 使能输出。
位 1	C1P	0x0	rw	通道 1 极性 (Channel 1 polarity) 通道 1 配置为输出： 0: C1OUT 的有效电平为高 1: C1OUT 的有效电平为低 通道 1 配置为输入： C1CP/C1P 位共同定义输入信号有效沿。 00: C1IN 的有效边沿为上升沿；作为外部触发使用时，C1IN 不反相。 01: C1IN 的有效边沿为下降沿；作为外部触发使用时，C1IN 反相。 10: 保留 11: C1IN 的有效边沿为上升沿和下降沿；作为外部触发使用时，C1IN 不反相。

位 0	C1EN	0x0	rw	通道 1 使能 (Channel 1 enable) 0: 禁止输入或输出; 1: 使能输入或输出。
-----	------	-----	----	--

表 14-13 带刹车功能的互补输出通道 CxOUT 和 CxCOUT 的控制位

控制位					输出状态 (1)	
OEN 位	FCSODI S 位	FCSOE N 位	CxEN 位	CxCEN 位	CxOUT 输出状态	CxCOUT 输出状态
1	X	0	0	0	输出禁止 (与定时器断开) CxOUT=0, Cx_EN=0	输出禁止 (与定时器断开) CxOUT=0, CxCEN=0
			0	0	输出禁止 (与定时器断开) CxOUT=0, Cx_EN=0	CxORAW + 极性, CxOUT=CxORAW xor CxCP, CxCEN=1
			0	1	CxORAW+极性, CxOUT=CxORAW xor CxP, Cx_EN=1	输出禁止 (与定时器断开) CxOUT=0, CxCEN=0
		1	0	1	CxORAW+极性+死区, Cx_EN=1	CxORAW 反相+极性+死区, CxCEN=1
		1	0	0	输出禁止 (与定时器断开) CxOUT=CxP, Cx_EN=0	输出禁止 (与定时器断开) CxOUT=CxCP, CxCEN=0
		1	0	1	关闭状态 (输出使能且为无效电平) CxOUT=CxP, Cx_EN=1	CxORAW + 极性, CxOUT=CxORAW xor CxCP, CxCEN=1
		1	1	0	CxORAW + 极性, CxOUT=CxORAW xor CxP, Cx_EN=1	关闭状态 (输出使能且为无效电平) CxOUT=CxCP, CxCEN=1
		1	1	1	CxORAW+极性+死区, Cx_EN=1	CxORAW 反相+极性+死区, CxCEN=1
0	X	0	0	0	输出禁止 (对应 IO 与定时器断开, IO 浮空) 异步地: CxOUT=CxP, Cx_EN=0, CxCOUT=CxCP, CxCEN=0;	CxORAW 反相+极性+死区, CxCEN=1
		0	0	1	若时钟存在: 经过一个死区时间后 CxOUT=CxIOS, CxCOUT=CxCIOS, 假设 CxIOS 与 CxCIOS 并不都对应 CxOUT 和 CxCOUT 的有效电平。	若时钟存在: 经过一个死区时间后 CxOUT=CxIOS, CxCOUT=CxCIOS, 假设 CxIOS 与 CxCIOS 并不都对应 CxOUT 和 CxCOUT 的有效电平。
		0	1	0		
		0	1	1		
		1	0	0		
		1	0	1		
		1	1	0		
		1	1	1		

注意: 如果一个通道的 2 个输出都没有使用 ( $CxEN = CxCEN = 0$ ) , 那么 CxIOS, CxCIOS, CxP 和 CxCP 都必须清零。

注意: 引脚连接到互补的 CxOUT 和 CxCOUT 通道的外部 I/O 引脚的状态, 取决于 CxOUT 和 CxCOUT 通道状态和 GPIO 以及 IOMUX 寄存器。

#### 14.5.4.8 TMR16 和 TMR17 计数值 (TMRx\_CVAL)

域	简称	复位值	类型	功能
位 15: 0	CVAL	0x0	rw	计数值 (Counter value)

#### 14.5.4.9 TMR16和TMR17预分频器 (TMRx\_DIV)

域	简称	复位值	类型	功能
位 15: 0	DIV	0x0	rw	分频系数 (Divider value) 计数器时钟频率 $f_{CK\_CNT} = f_{TMR\_CLK} / (\text{DIV}[15:0] + 1)$ 溢出事件发生时该寄存器值被传送到实际的预分频寄存器中。

#### 14.5.4.10 TMR16和TMR17周期寄存器 (TMRx\_PR)

域	简称	复位值	类型	功能
位 15: 0	PR	0x0	rw	周期值 (Period value) 定时器计数的周期值。当周期值为 0 时，定时器不工作。

#### 14.5.4.11 TMR16和TMR17重复周期寄存器 (TMRx\_RPR)

域	简称	复位值	类型	功能
位 15: 8	保留	0x0	resd	保持默认值。
位 7: 0	RPR	0x0	rw	重复周期的次数 (Repetition of period value) 这些位用于减慢溢出事件发生的速率，当重复周期的次数减为 0 时才会发生溢出事件。

#### 14.5.4.12 TMR16和TMR17通道1数据寄存器 (TMRx\_C1DT)

域	简称	复位值	类型	功能
位 15: 0	C1DT	0x0	rw	通道 1 数据寄存器值 (Channel 1 data register) 若通道 1 配置为输入： C1DT 是前一次通道 1 输入事件 (C1IN) 所保存的 CVAL。 若通道 1 配置为输出： C1DT 是将要和 CVAL 进行比较的值，写入的值是否会立即生效取决于输出缓存使能位 (C1OBEN)，并根据设置在 C1OUT 上产生相应的输出。

#### 14.5.4.13 TMR16和TMR17刹车寄存器 (TMRx\_BRK)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0	resd	保持默认值
位 15	OEN	0x0	rw	输出使能 (output enable) 对配置为输出的通道，该位用于使能 CxOUT 和 CxCOUT 的输出。 0: 关闭； 1: 开启。
位 14	AOEN	0x0	rw	输出自动使能 (automatic output enable) 用于溢出事件时将 OEN 自动置'1' 0: 关闭； 1: 开启
位 13	BRKV	0x0	rw	刹车输入信号的有效性 (Brake input validity) 用于选择刹车输入信号的输入有效电平： 0: 低电平； 1: 高电平。
位 12	BRKEN	0x0	rw	刹车功能使能 (Brake enable) 用于开启刹车功能。 0: 关闭； 1: 开启。
位 11	FCSOEN	0x0	rw	总输出开时的冻结状态 (Frozen channel status when holistic output enable) 该位用于配置具有互补输出的通道，在定时器不工作且 OEN=1 时的通道状态。 0: 关闭 CxOUT/CxCOUT 输出； 1: 开启 CxOUT/CxCOUT 输出，输出为无效电平。
位 10	FCSODIS	0x0	rw	总输出关时的冻结状态 (Frozen channel status when holistic output disable)

位 9: 8	WPC	0x0	rw	该位用于配置具有互补输出的通道，在定时器不工作且 OEN=0 时的通道状态。 0 : 关闭 CxOUT/CxCOUT 输出； 1 : 开启 CxOUT/CxCOUT 输出，输出为空闲电平。
位 7: 0	DTC	0x0	rw	写保护配置 (Write protected configuration) 该位用于配置写保护。 00: 写保护关闭； 01: 3 级写保护，以下位受写保护： TMR1_BRK: DTC、BRKEN、BRKV 和 AOEN TMR1_CTRL2: CxIOS 和 CxCIOS 10: 2 级写保护，除 3 级写保护的内容外，以下位也受写 保护： TMR1_CCTRL: CxP 和 CxCP TMR1_BRK: FCSODIS 和 FCSOEN 11: 1 级写保护，除 2 级写保护的内容外，以下位也受写 保护： TMR1_CMx: C2OCTRL 和 C2OBEN 注：WPC>0 时将无法再次被修改，直到系统复位。
位 7: 0	DTB	0x0	rw	死区配置 (Dead-time configuration) 这些位用于配置死区时间。取 DTC[7: 0] 的高 3 位为功能 选择位： 0xx: DT = DTC [7: 0] * TDTS; 10x: DT = (64+ DTC [5: 0]) * TDTS * 2; 110: DT = (32+ DTC [4: 0]) * TDTS * 8; 111: DT = (32+ DTC [4: 0]) * TDTS * 16;

注意：根据锁定设置，BKF、AOEN、BRKV、BRKEN、FCSODIS、FCSOEN 和 DTC[7: 0] 位均可被写保护，有必要在第一次写入 TMRx\_BRK 寄存器时对它们进行配置。

#### 14.5.4.14 TMR16和TMR17 DMA控制寄存器 (TMRx\_DMACTRL)

域	简称	复位值	类型	功能
位 15: 13	保留	0x0	resd	保持默认值。
位 12: 8	DTB	0x0	rw	DMA 传输字节 (DMA transfer bytes) 这些位定义了传输的字节个数： 00000: 1 个字节      00001: 2 个字节 00010: 3 个字节      00011: 4 个字节 ..... 10000: 17 个字节      10001: 18 个字节
位 7: 5	保留	0x0	resd	保持默认值。
位 4: 0	ADDR	0x0	rw	DMA 传输地址偏移 (DMA transfer address offset) ADDR 定义了从 TMRx_CTRL1 所在地址开始的偏移量： 00000: TMRx_CTRL1, 00001: TMRx_CTRL2, 00010: TMRx_STCTRL, .....

#### 14.5.4.15 TMR16和TMR17 DMA数据寄存器 (TMRx\_DMADT)

域	简称	复位值	类型	功能
位 15: 0	DMADT	0x0	rw	DMA 传输的数据寄存器 (DMA data register) 通过对 DMADT 寄存器的读写能够实现对任意 TMR 寄存器的操作，其操作的寄存器地址范围是：TMRx 外设地址 + ADDR*4 至 TMRx 外设地址 + ADDR*4 + DTB*4。

## 14.6 高级控制定时器 (TMR1)

### 14.6.1 TMR1简介

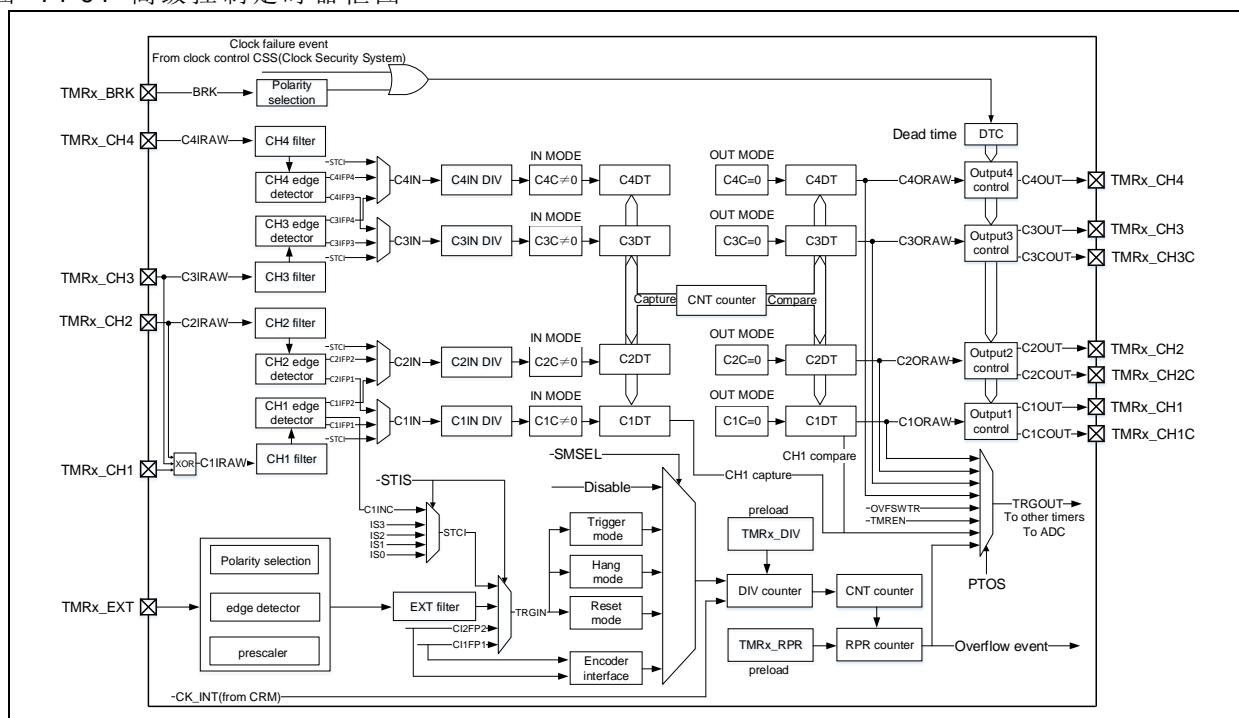
高级定时器 TMR1 包含一个支持向上、向下、中央双向对齐计数的 16 位计数器、4 个通道寄存器、4 组独立的通道。可实现嵌入死区、输入捕获、可编程 PWM 输出。

### 14.6.2 TMR1主要特性

TMR1 定时器的功能包括：

- 可选内部、外部、内部触发输入用作计数时钟
- 16 位支持向上、向下、双向、重复计数、编码器模式的计数器
- 4 组独立通道，支持输入捕获、输出比较、PWM 生成、单周期模式、死区插入。
- 3 组支持互补输出的独立通道
- 支持 TMR 刹车功能
- 定时器之间可互联同步
- 支持溢出事件、触发事件、刹车输入、通道事件触发中断/DMA
- 支持 TMR burst DMA 传输

图 14-91 高级控制定时器框图

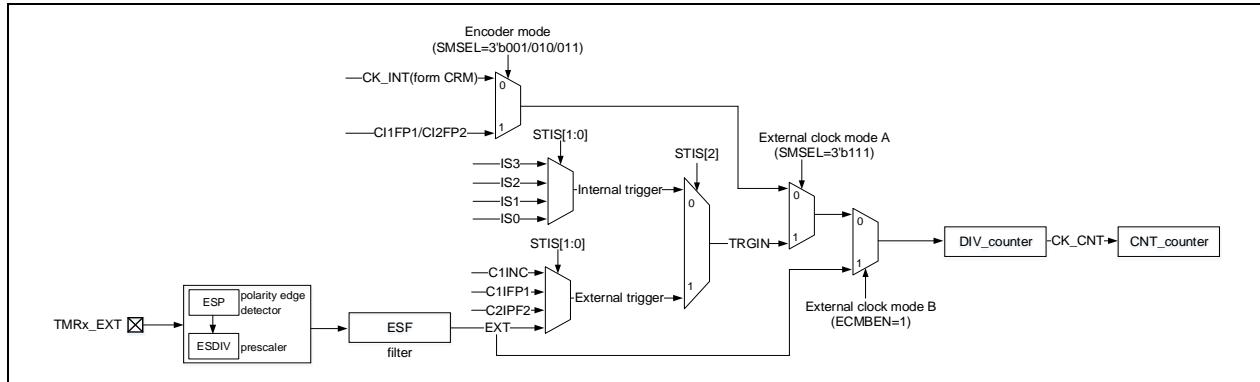


### 14.6.3 TMR1功能描述

#### 14.6.3.1 计数时钟

TMR1 计数时钟可从内部时钟 (CK\_INT)、外部时钟 (外部时钟模式 A、B)、内部触发输入 (ISx) 这些时钟源提供。

图 14-92 计数时钟

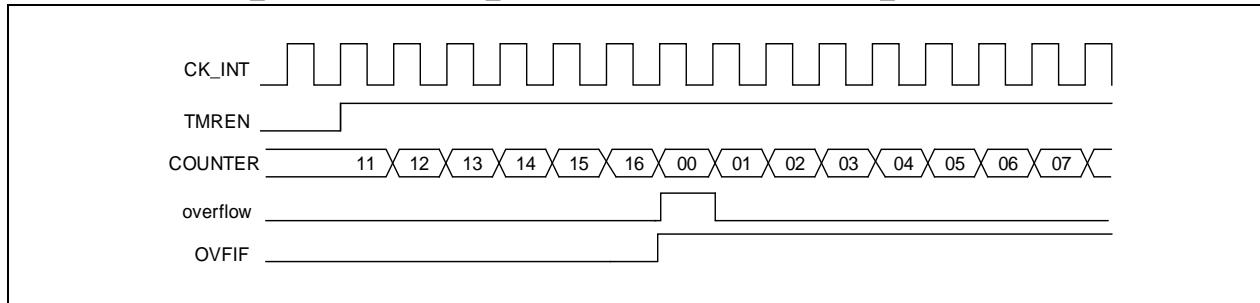


### 内部时钟 (CK\_INT)

默认下使用 CK\_INT 经由预分频器驱动计数器计数, 当 TMR 对应的 APB 时钟预分频系数是 1 时, CK\_INT 频率等于 APB 时钟频率, 否则 CK\_INT 频率等于 APB 时钟频率的 2 倍。相关配置流程如下:

- 配置 TMRx\_CTRL1 寄存器 TWCMSL[1:0], 选择计数模式, 若选择单向对齐计数模式, 还需配置 TMRx\_CTRL1 寄存器 OWCDIR 选择计数方向。
- 配置 TMRx\_DIV 寄存器, 设置计数器计数频率。
- 配置 TMRx\_PR 寄存器, 设置计数器计数周期。
- 配置 TMRx\_CTRL1 寄存器 TMREN, 使能计数器。

图 14-93 使用 CK\_INT 计数, TMRx\_DIV=0x0, 周期寄存器 TMRx\_PR=0x16



### 外部时钟 (TRGIN/EXT)

计数时钟可由两种外部时钟源提供, 分别为 TRGIN 和 EXT 信号。

当 SMSEL=3'b111 时, 外部时钟模式 A 被选中, 配置 STIS[2:0] 来选择外部时钟源 TRGIN 信号驱动计数器计数。外部时钟源 TRGIN 可选则 C1INC (STIS=3'b100, 通道 1 上升沿和下降沿信号)、C1IFP1 (STIS=3'b101, 通道 1 滤波且极性选择后信号)、C2IFP2 (STIS=3'b110, 通道 2 滤波且极性选择后信号) 和 EXT (STIS=3'b111, 外部输入经极性选择、分频和滤波后信号)。

当 ECMBEN=1 时, 外部时钟模式 B 被选中, 计数器由外部输入经极性选择、分频和滤波后 EXT 信号驱动计数。外部时钟模式 B 等效于外部时钟模式 A 选择 EXT 信号作为外部时钟源 TRGIN。

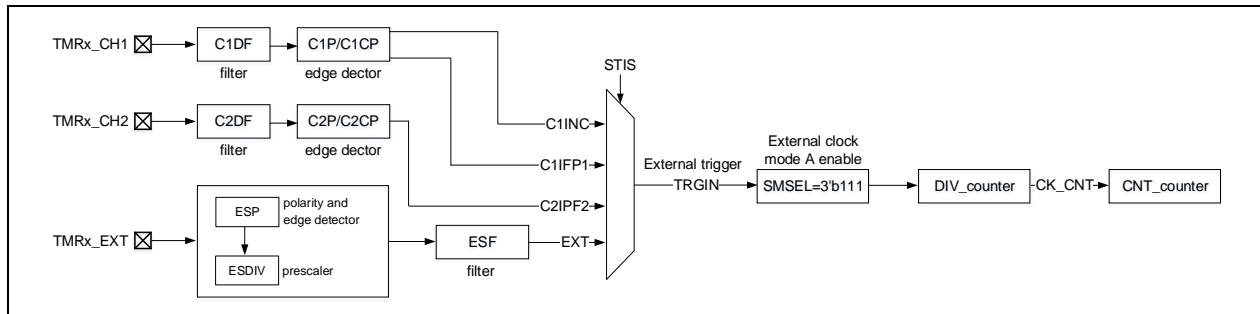
若要使用外部时钟模式 A, 可按如下步骤配置:

- 配置外部时钟源 TRGIN 参数。
  - 若选择 TRGIN 来源为 TMRx\_CH1, 需配置通道 1 输入滤波 (TMRx\_CM1 寄存器 C1DF[3:0]) 和通道 1 输入极性 (TMRx\_CCTRL 寄存器 C1P/C1CP)。
  - 若选择 TRGIN 来源为 TMRx\_CH2, 需配置通道 2 输入滤波 (TMRx\_CM1 寄存器 C2DF[3:0]) 和通道 1 输入极性 (TMRx\_CCTRL 寄存器 C2P/C2CP)。
  - 若选择 TRGIN 来源为 TMRx\_EXT, 需配置外部信号极性 (TMRx\_STCTRL 寄存器 ESP)、外部信号分频 (TMRx\_STCTRL 寄存器 ESDIV[1:0]) 和外部信号滤波 (TMRx\_STCTRL 寄存器 ESF[3:0])。
- 配置 TMRx\_STCTRL 寄存器 STIS[1:0], 设置 TRGIN 信号来源。
- 配置 TMRx\_STCTRL 寄存器 SMSEL=3'b111, 使能外部时钟模式 A。
- 配置 TMRx\_DIV 寄存器 DIV[15:0], 设置计数器计数频率。
- 配置 TMRx\_PR 寄存器 PR[15:0], 设置计数器计数周期。
- 配置 TMRx\_CTRL1 寄存器 TMREN, 使能计数器。

若要使用外部时钟模式 B，可按如下步骤配置：

- 配置 TMRx\_STCTRL 寄存器 ESP，设置外部信号极性。
- 配置 TMRx\_STCTRL 寄存器 ESDIV[1:0]，设置外部信号分频。
- 配置 TMRx\_STCTRL 寄存器 ESF[3:0]，设置外部信号滤波。
- 配置 TMRx\_STCTRL 寄存器 ECMBEN，使能外部时钟模式 B。
- 配置 TMRx\_DIV 寄存器 DIV[15:0]，设置计数器计数频率。
- 配置 TMRx\_PR 寄存器 PR[15:0]，设置计数器计数周期。
- 配置 TMRx\_CTRL1 寄存器 TMREN，使能计数器。

图 14-94 外部时钟模式 A 框图



注：由于同步逻辑，输入端信号与计数器实际时钟之间存在一定延时。

图 14-95 使用外部时钟模式 A 计数，PR=0x32, DIV=0x0

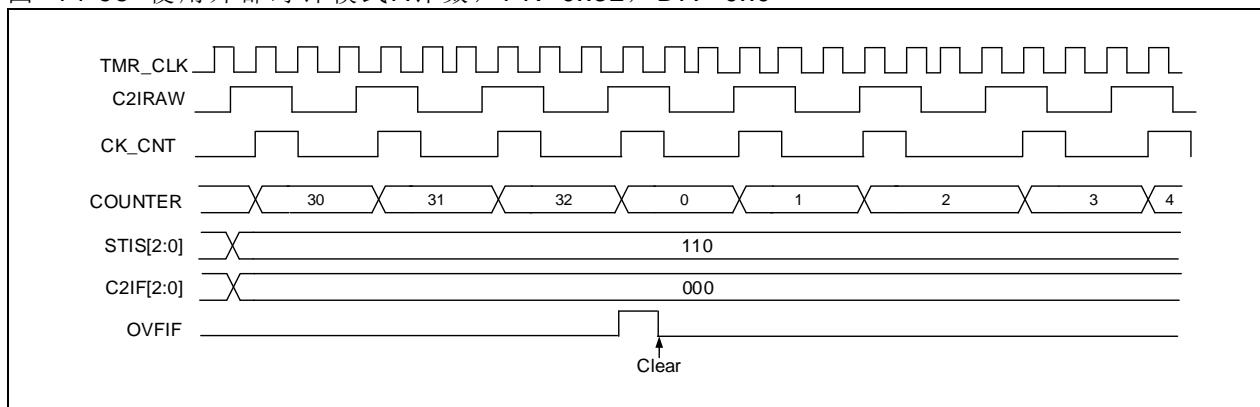
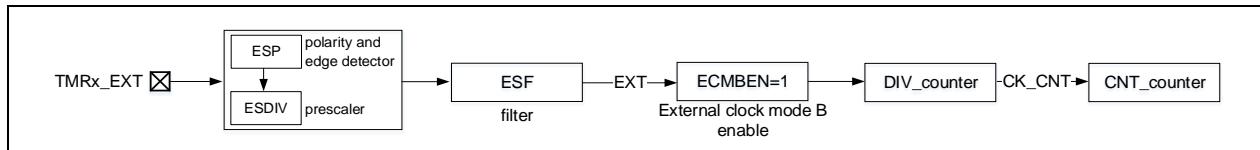
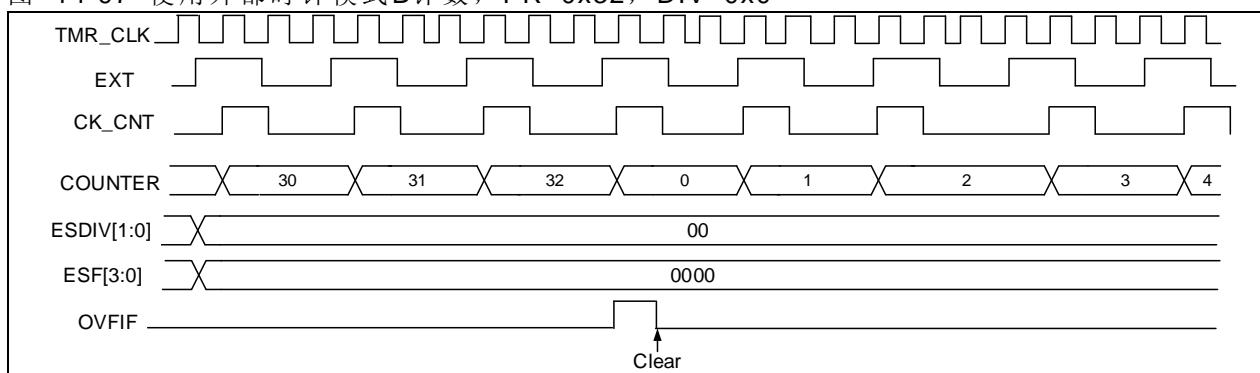


图 14-96 外部时钟模式 B 框图



注：由于同步逻辑，输入端 EXT 信号与计数器实际时钟之间存在一定延时。

图 14-97 使用外部时钟模式 B 计数，PR=0x32, DIV=0x0



### 内部触发输入 (ISx)

定时器之间支持互联同步，因此一个定时器的 TMR\_CLK 可由另一个定时器输出信号 TRGOUT 提供。配

置 STIS[2: 0]选择内部触发信号驱动计数器计数。

高级定时器内含一个 16 位预分频器，用于产生驱动计数器计数的时钟 CK\_CNT，通过配置 TMR1\_DIV 寄存器值，可灵活调整 CK\_CNT 与 TMR\_CLK 之间的分频关系。预分频值可在任何时刻修改，但只在下一个溢出事件发生时，新值才会生效。

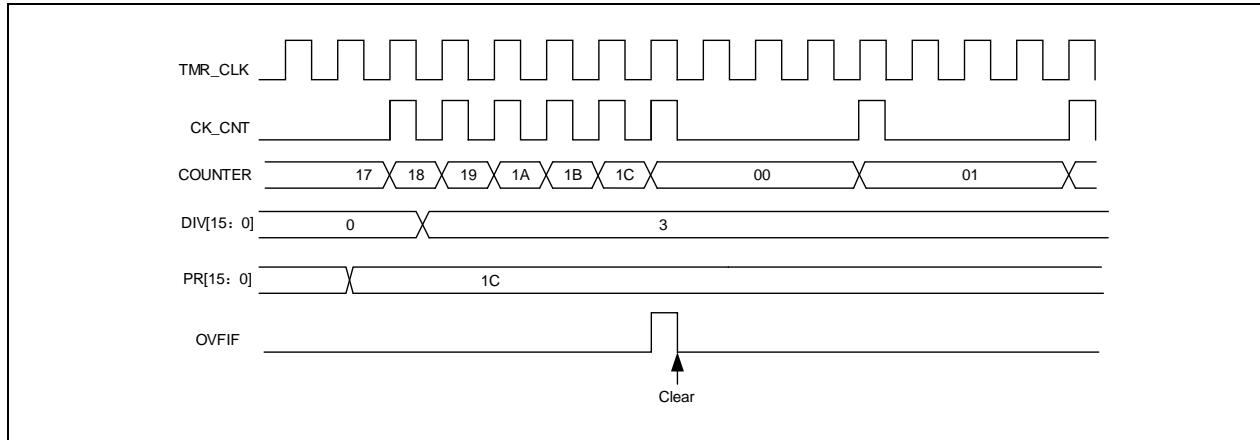
内部触发输入配置流程如下：

- 配置 TMRx\_PR 寄存器，设置计数器计数周期。
- 配置 TMRx\_DIV 寄存器，设置计数器计数频率。
- 配置 TMRx\_CTRL1 寄存器 TWCMSEL[1:0]位，设置计数器计数模式。
- 配置 TMRx\_STCTRL 寄存器 STIS[2:0]位范围为 3'b000~3'b011，选择内部触发。
- 配置 TMRx\_STCTRL 寄存器 SMSEL[2:0]=3'b111，选择外部时钟模式 A。
- 配置 TMRx\_CTRL1 寄存器 TMREN 位，使能 TMRx 计数。

表 14-14 TMR1 内部触发连接

次定时器	IS0 (STIS=000)	IS1 (STIS=001)	IS2 (STIS=010)	IS3 (STIS=011)
TMR1	TMR15	-	TMR3	-

图 14-98 当预分频器的参数从 1 变到 4 时，计数器的时序图



### 14.6.3.2 计数模式

高级定时器支持多种计数模式，用来满足不同的应用场景。其内部拥有一个支持 16 位向上计、向下、中央双向对齐计数模式计数器。

TMRx\_PR 寄存器用于设置计数器计数周期。默认 TMRx\_PR 寄存器值会立即传入它的影子寄存器；当开启周期缓冲功能后（PRBEN 置 1），TMRx\_PR 寄存器值在溢出事件发生时传入它的影子寄存器。

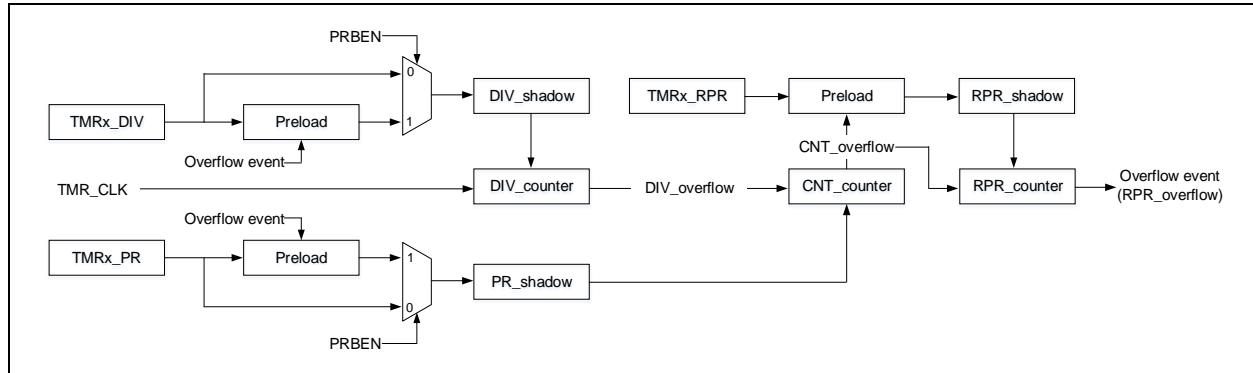
TMRx\_DIV 寄存器用于设置计数器计数频率，每（DIV[15:0]+1）个计数时钟周期，计数器计数一次。和 TMRx\_PR 寄存器类似，开启周期缓冲功能后，TMRx\_DIV 寄存器值在溢出事件时更新至它的影子寄存器。

读取 TMRx\_CNT 寄存器会返回当前计数器计数值，写入 TMRx\_CNT 寄存器会更新计数器当前计数值为写入值。

默认允许产生溢出事件，设置 TMRx\_CTRL1 寄存器 OVFEN=1 将禁止更新事件产生。TMRx\_CTRL1 寄存器 OVFS 用于选择溢出事件来源，默认计数器上溢或下溢、置位 OVFSWTR、复位模式次定时器控制器产生的复位信号产生溢出事件。置位 OVFS 后，只有计数器上溢或下溢产生溢出事件。

TMREN 位置 1 将使能定时器计数，由于同步逻辑，实际驱动计数器的使能信号 TMR\_EN 相对于 TMREN 延迟一个时钟周期。

图 14-99 计数器基本结构



### 向上计数模式

配置 TMRx\_CTRL1 寄存器 CMSEL[1:0]=2'b00, OWCDIR=1'b0 开启向上计数模式，计数值达到 TMRx\_PR 值时，重新从 0 向上计数，计数器上溢并产生溢出事件，同时 OVFIF 位置 1。若禁止产生溢出事件，计数器溢出后不再重载预分频值和周期值，否则预分频值和周期值在溢出事件后更新。

图 14-100 PRBEN=0 时的溢出事件

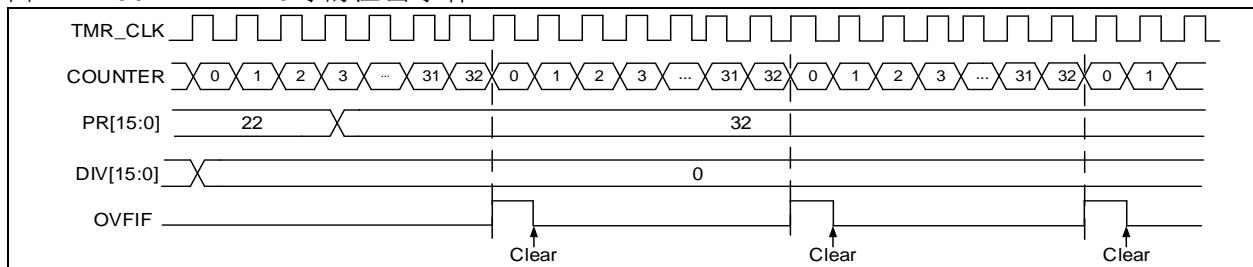
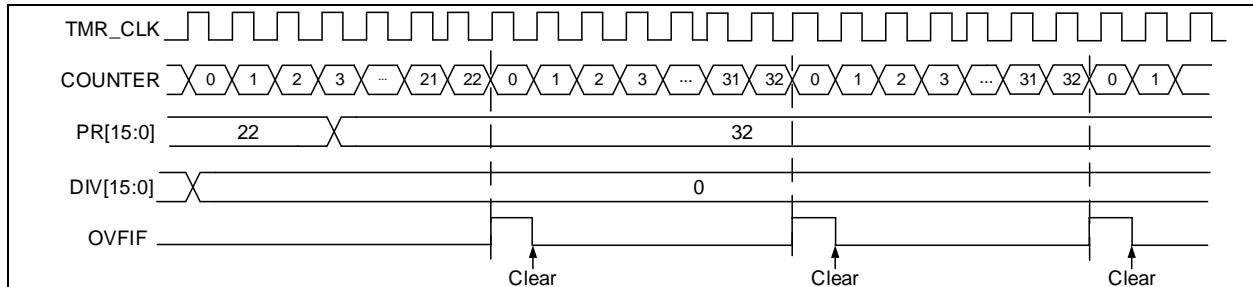


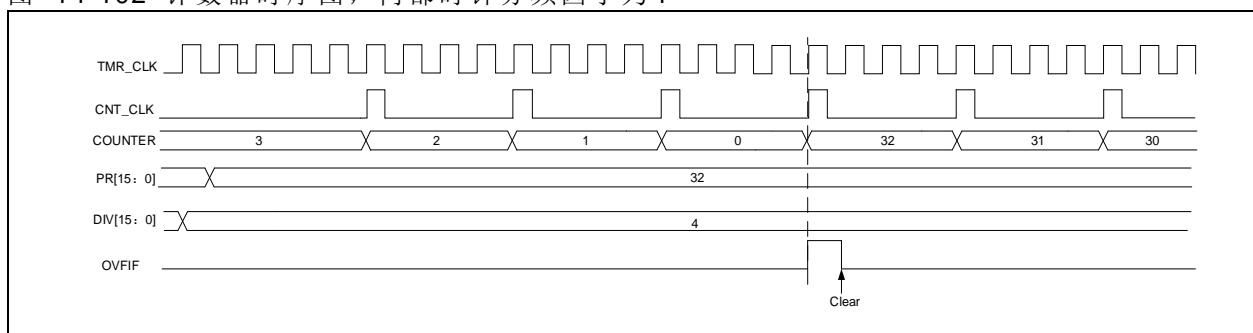
图 14-101 PRBEN=1 时的溢出事件



### 向下计数模式

配置 TMRx\_CTRL1 寄存器 CMSEL[1:0]=2'b00, OWCDIR=1'b1 开启向下计数模式，计数值达到 0 值并重新从 TMRx\_PR 向上下数时，计数器下溢并产生溢出事件。

图 14-102 计数器时序图，内部时钟分频因子为 4



### 中央双向对齐计数模式

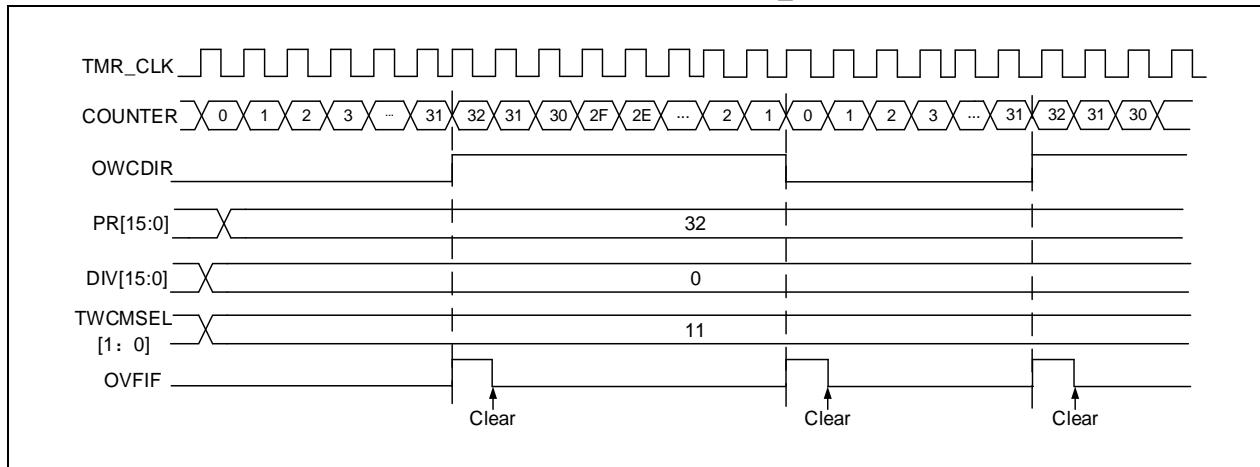
配置 TMRx\_CTRL1 寄存器 CMSEL[1:0]≠2'b00 开启中央双向对齐计数模式，中央双向对齐计数模式下计数器交替向上、向下计数。计数值从 TMRx\_PR 值向下计数到 1 值，产生下溢事件，然后从 0 开始向上计数；向上计数到 TMRx\_PR 值-1，产生上溢事件，之后从 TMRx\_PR 值向下计数。计数器计数方向由

计数器方向控制位（OWCDIR）实时查看。

TMRx\_CTRL1 寄存器 TWCSEL[1:0]位还用于选择中央双向对齐计数模式下 CxIF 标志置起方式，中央双向对齐计数模式 1（TWCSEL[1:0]=2'b01）仅允许 CxIF 标志位在计数器向下计数时置起；双向对齐计数模式 2（TWCSEL[1:0]=2'b10）仅允许 CxIF 标志位在计数器向上计数时置起；双向对齐计数模式 3（TWCSEL[1:0]=2'b11）允许 CxIF 标志位在计数器向上和向下计数时置起。

注意： 中央双向对齐计数模式下，OWCDIR 位为只读位。

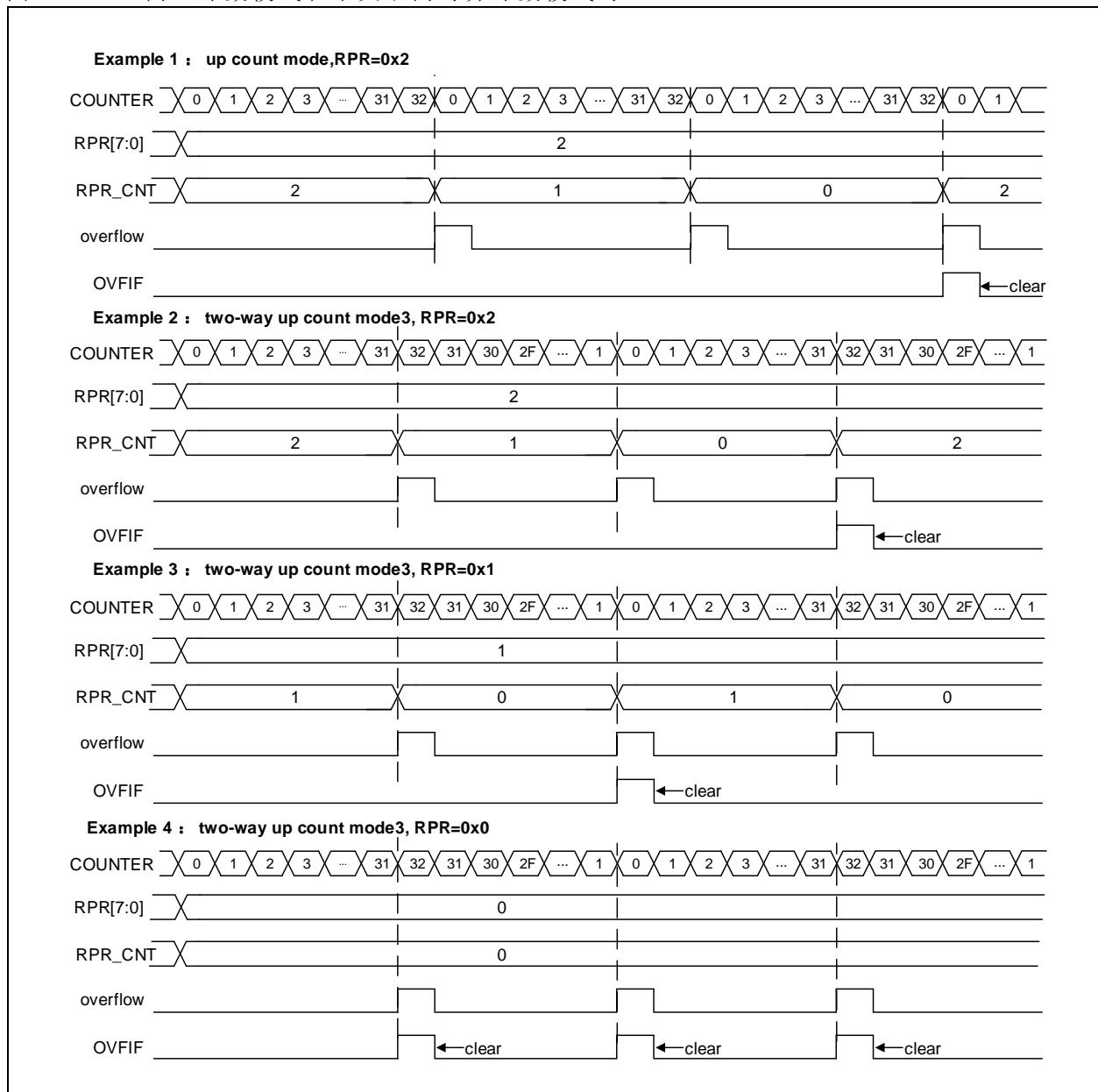
图 14-103 计数器时序图，内部时钟分频因子为 1，TMRx\_PR=0x32



#### 重复计数模式：

TMRx\_RPR 寄存器用于配置重复计数器计数周期，TMRx\_RPR 寄存器为非 0 值时，重复计数模式启动。重复计数模式下，每 ( $RPR[7:0]+1$ ) 次计数器溢出将产生一次溢出事件。每次计数器溢出，重复计数器递减，仅当重复计数器计数值等于 0 值时，计数器溢出会产生溢出事件。通过配置不同重复计数器值，可调整溢出事件产生的频率。

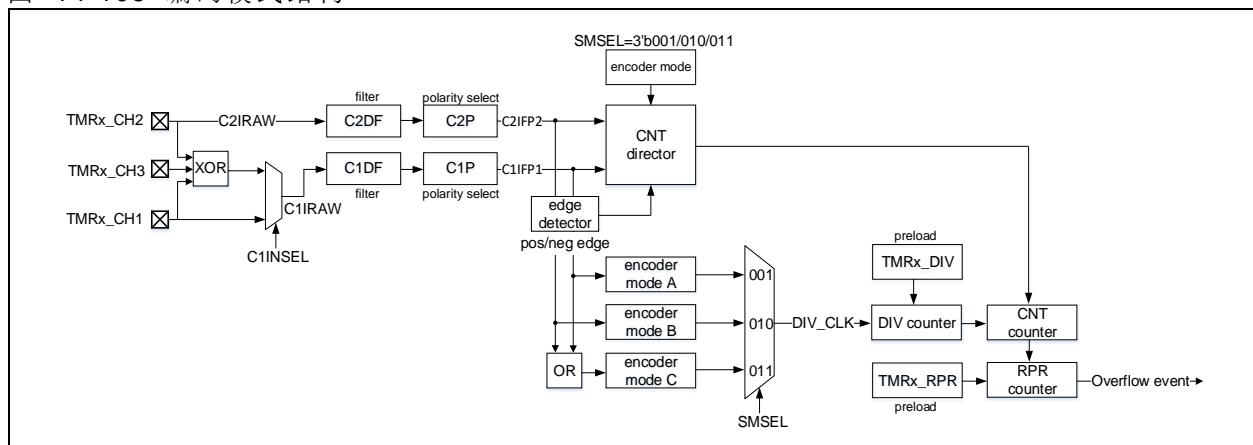
图 14-104 向上计数模式和中央双向对齐计数模式时 OVFIF



### 编码器模式

编码器模式下需提供两组输入信号 TMRx\_CH1 和 TMRx\_CH2，根据一组输入信号电平值，计数器在另一组输入信号边沿向上或向下计数。计数方向由 OWCDIR 值指示。

图 14-105 编码模式结构



编码器模式 A: SMSEL=3'b001, 计数器在 C1IFP1 边沿计数(上升沿和下降沿), 计数方向由 C1IFP1 边沿方向和 C2IFP2 电平高低共同决定。

编码器模式 B: SMSEL=3'b010, 计数器在 C2IFP2 边沿计数(上升沿和下降沿), 计数方向由 C2IFP2 边沿方向和 C1IFP1 电平高低共同决定。

编码器模式 C: SMSEL=3'b011, 计数器在 C1IFP1 和 C2IFP2 边沿计数(上升沿和下降沿), 计数方向由 C1IFP1 边沿方向和 C2IFP2 电平高低、C2IFP2 边沿方向和 C1IFP1 电平高低共同决定共同决定。

若要使用编码器模式可按下面步骤配置:

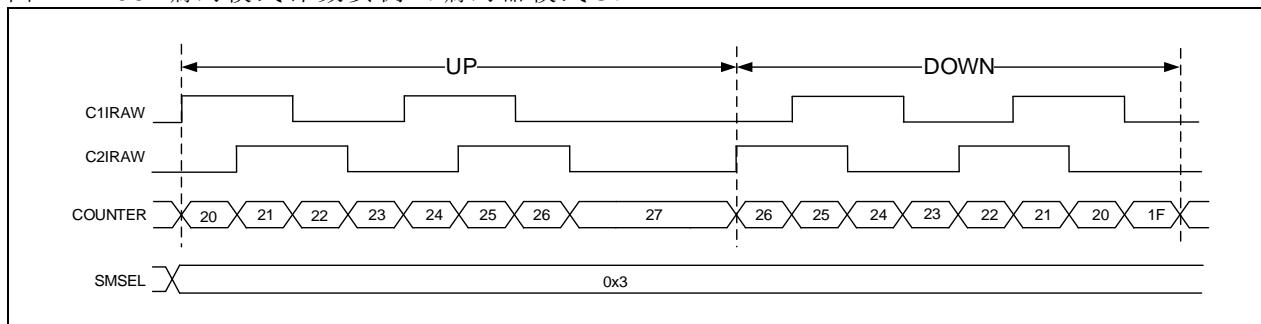
- 配置 TMRx\_CM1 寄存器 C1DF[3:0], 设置通道 1 输入信号滤波; 配置 TMRx\_CCTRL 寄存器 C1P, 设置通道 1 输入信号有效电平。
- 配置 TMRx\_CM1 寄存器 C2DF[3:0], 设置通道 2 输入信号滤波; 配置 TMRx\_CCTRL 寄存器 C2P, 设置通道 2 输入信号有效电平。
- 配置 TMRx\_CM1 寄存器 C1C[1:0], 设置通道 1 为输入模式; 配置 TMRx\_CM1 寄存器 C2C[1:0], 设置通道 2 为输入模式;
- 配置 TMRx\_STCTRL 寄存器 SMSEL[2:0], 选择编码器模式 A (SMSEL=3'b001)、编码器模式 B (SMSEL=3'b010) 或编码器模式 C (SMSEL=3'b011)。
- 配置 TMRx\_PR 寄存器 PR[15:0], 设置计数器计数周期。
- 配置 TMRx\_DIV 寄存器 DIV[15:0], 设置计数器计数频率。
- 配置 TMRx\_CH1 和 TMRx\_CH2 对应 IO 为复用模式。
- 配置 TMRx\_CTRL1 寄存器 TMREN, 使能计数器。

编码模式下计数器计数方向如下表所示:

表 14-15 计数方向与编码器信号的关系

计数边沿	计数边沿相对信号的电平 (C1IFP1 边沿对应 C2IFP2 电平, C2IFP2 边沿对应 C1IFP1 电平)	C1IFP1 边沿方向		C2IFP2 边沿方向	
		上升	下降	上升	下降
C1IFP1	高	向下计数	向上计数	不计数	不计数
	低	向上计数	向下计数	不计数	不计数
C2IFP2	高	不计数	不计数	向上计数	向下计数
	低	不计数	不计数	向下计数	向上计数
C1IFP1 和 C2IFP2	高	向下计数	向上计数	向上计数	向下计数
	低	向上计数	向下计数	向下计数	向上计数

图 14-106 编码模式计数实例 (编码器模式 C)



### 14.6.3.3 TMR输入部分

TMR1 拥有 4 个独立通道, 每个通道可配置为输入或输出, 当配置位输入时, 每个通道输入信号依次经过以下处理:

- TMRx\_CHx 经过预处理输出 CxIRAW。配置 C1INSEL 位, 选择 C1IRAW 来源是 TMRx\_CH1 或是 TMRx\_CH1、TMRx\_CH2、TMRx\_CH3 异或。C2IRAW、C3IRAW、C4IRAW 来源是 TMRx\_CH2、TMRx\_CH3、TMRx\_CH4。

- CxIRAW 输入数字滤波器，输出滤波后信号 CxIF。数字滤波器通过 CxDf 位配置采样频率和次数。
- CxIF 输入边沿检测器，输出边沿选择后信号 CxIFPx。边沿选择由 CxP 和 CxCp 位共同控制，可选择输入上升沿、下降沿或双边沿有效。
- CxIFPx 输入捕获信号选择器，输出选择后信号 CxIN。捕获信号选择器由 CxC 控制，可选择 CxIN 来源为 CxIFPx、CylIFPx、STCI。其中 CylIFPx ( $x \neq y$ ) 是来自通道 y 的 CylIFPy 经通道 x 边沿检测器处理后的信号（例如 C1IFP2 是来自通道 1 的 C1IFP1 信号经过通道 2 边沿检测器处理后的信号）；STCI 来自次定时器控制器，由 STIS 位选择来源。
- CxIN 经由输入通道分频器，输出分频后信号 CxIPS。分频系数由 CxIDIV 位配置为不分频、2 分频、4 分频或 8 分频。

图 14-107 输入/输出通道 1 的主电路

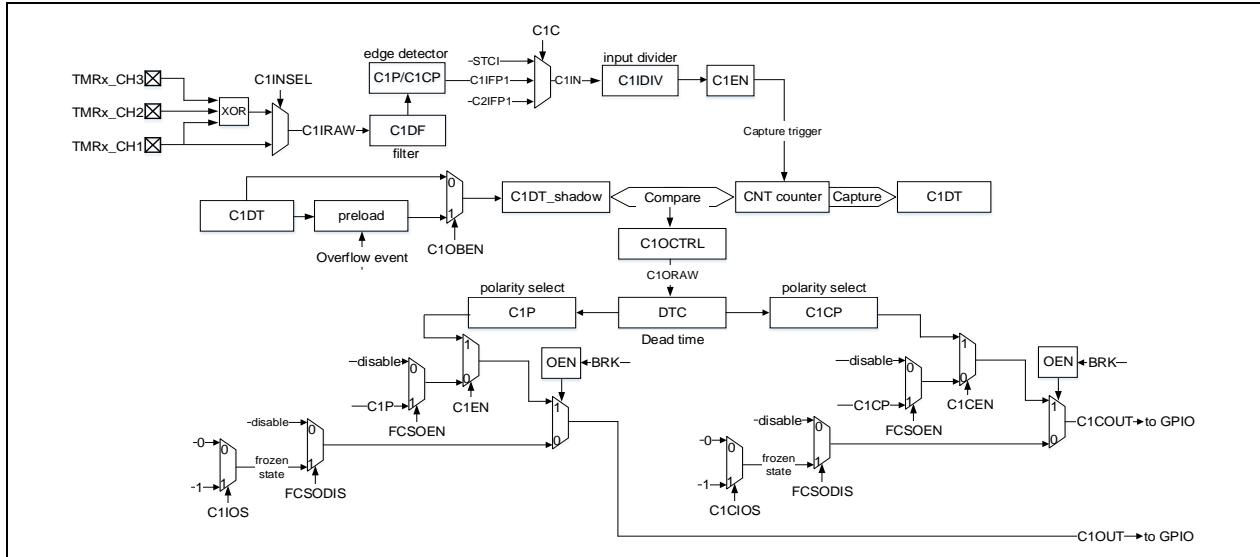
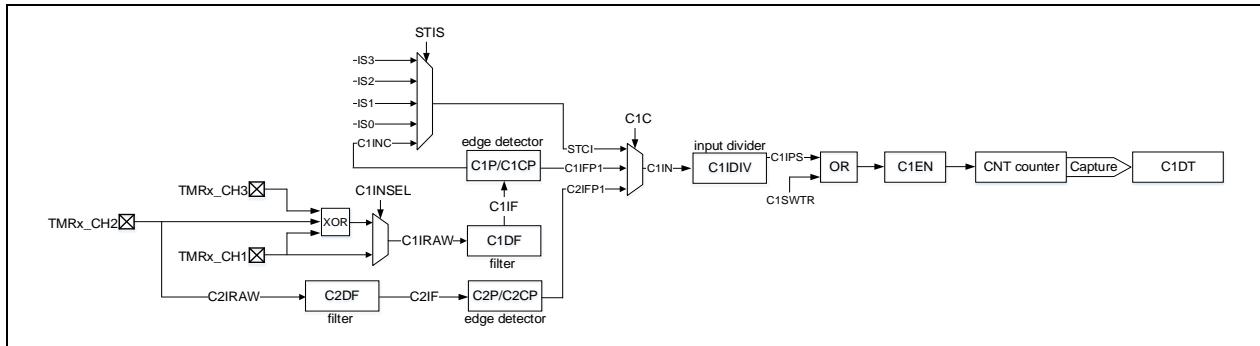


图 14-108 通道 1 输入部分



## 输入模式

此模式下，当选中的触发信号被检测到，通道寄存器（TMRx\_CxDT）记录当前计数器计数值，并将捕获比较中断标志位（CxIF）置 1，若已使能通道中断（CxIEN）、通道 DMA 请求（CxDEN）则产生相应的中断和 DMA 请求。若在 CxIF 置 1 后检测到触发信号，将产生捕获溢出事件，TMRx\_CxDT 会使用当前计数器计数值覆盖之前记录的计数器计数值，同时通道再捕获标志位（CxRF）置 1。

以若要捕获 C1IN 输入的上升沿，可按如下进行配置：

- 将通道模式寄存器 1（TMR1\_CM1）中的 C1C 位配置为 01，选择 C1IN 作为通道 1 输入。
- 配置 C1IN 信号滤波器带宽（CxDF[3: 0]）。
- 配置 C1IN 通道的有效沿，在通道控制寄存器（TMR1\_CCTRL）中写入 C1P=0（上升沿）。
- 配置 C1IN 信号捕获分频（C1DIV[1: 0]）。
- 使能通道 1 输入捕获（C1EN=1）。
- 根据需要设置 DMA/中断使能寄存器（TMR1\_IDEN）中的 C1IEN 为、DMA/中断使能寄存器（TMR1\_IDEN）中的 C1DEN 位，选择中断请求或 DMA 请求。

### 多输入异或

通道 1 的输入端可选择 TMR1\_CH1、TMR1\_CH2 和 TMR1\_CH3 经异或逻辑后输入。将控制寄存器 2 (TMR1\_CTRL2) 中的 C1INSEL 位置 1 可开启此功能。

多输入异或功能可用于连接霍尔传感器，例如，将异或输入的三个输入端分别连接到三个霍尔传感器，通过分析三路霍尔传感器信号可计算出转子的位置和速度。

### PWM 输入

PWM 输入模式适用于通道 1 和 2，要使用此模式，需要将 C1IN 和 C2IN 映射到同一 TMRx\_CHx，并且通道 1 或 2 的 CxIFPx 配置成触发次定时器控制器复位。

PWM 输入模式可用于测量输入信号的周期和占空比，如需测量通道 1 输入信号的周期和占空比，操作步骤如下：

- 配置 C1C=2'b01，选择 C1IN 为 C1IFP1。
- 配置 C1P=1'b0，选择 C1IFP1 上升沿有效。
- 配置 C2C=2'b10，选择 C2IN 为 C1IFP2。
- 配置 C2P=1'b1，选择 C1IFP2 下降沿有效。
- 配置 STIS=3'b101，选择次定时器触发信号为 C1IFP1。
- 配置 SMSEL=3'b100，选择次定时器模式为复位模式。
- 配置 C1EN=1'b1，C2EN=1'b1。使能通道 1 和输入捕获。

上述配置下，通道 1 输入信号的上升沿会触发捕获并将捕获值存储到 C1DT 寄存器，同时通道 1 输入信号上升沿复位计数器。通道 1 输入信号下降沿触发捕获并将捕获值存储到 C2DT 寄存器。通道 1 输入信号的周期可通过 C1DT 计算，占空比可通过 C2DT 计算。

图 14-109 PWM 输入模式配置实例

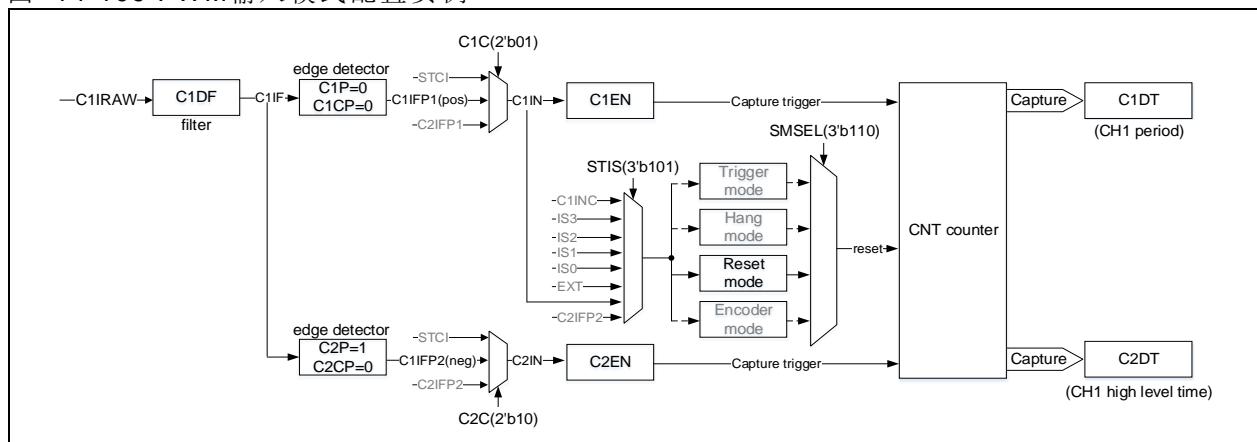
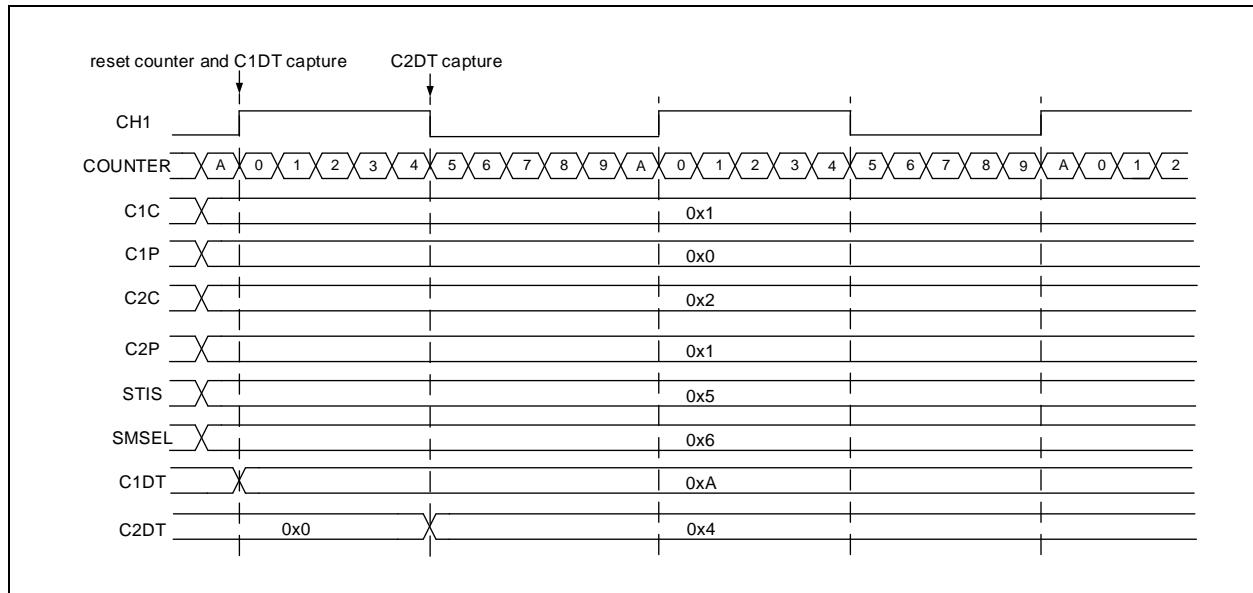


图 14-110 PWM输入模式



#### 14.6.3.4 TMR输出部分

TMR 的输出部分由比较器和输出控制构成，用于编程输出信号的周期、占空比、极性。高级定时器的输出部分在不同通道上有所不同，如下图所示：

图 14-111 通道1至3输出部分

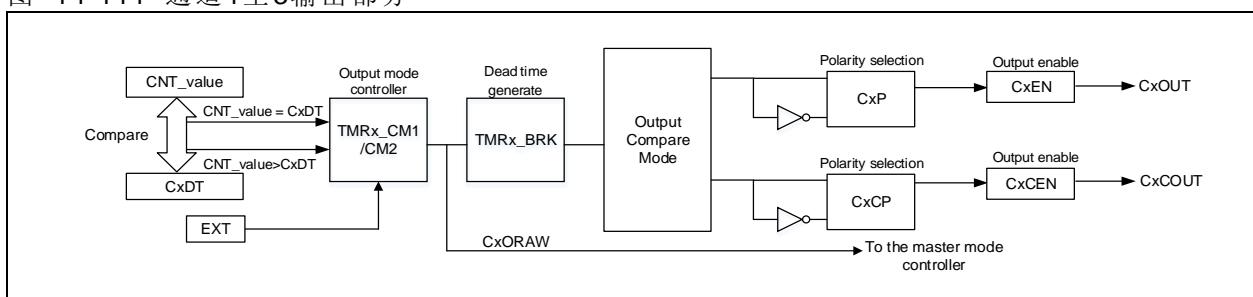
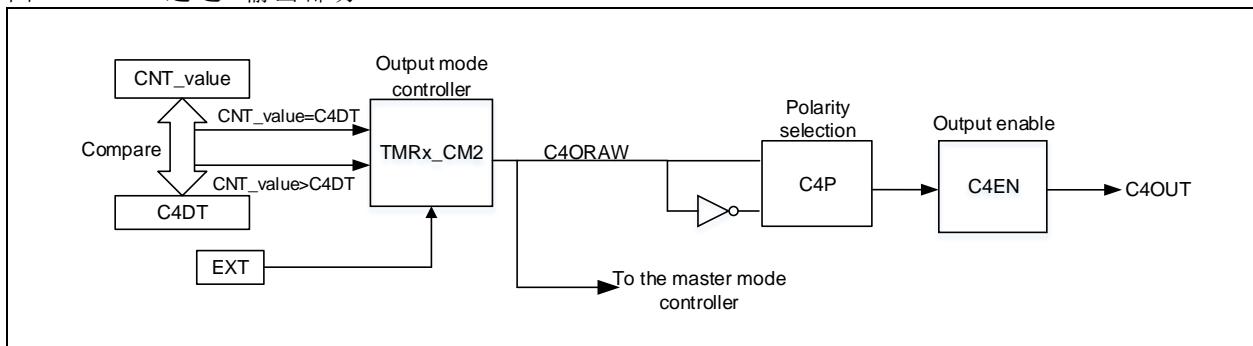


图 14-112 通道4输出部分



#### 输出模式

配置  $CxO[1:0] \neq 2'b00$  将通道配置为输出可实现多种输出模式，此时，计数器计数值将与通道寄存器 (TMR1\_CxDT) 值比较，并根据  $CxOCTRL[2:0]$  位配置的输出模式，产生中间信号  $CxORAW$ ，再经过输出控制逻辑处理后输送到 IO。输出信号的周期由周期寄存器 (TMR1\_PR) 值配置，占空比则由通道寄存器 (TMR1\_CxDT) 值配置。

输出比较模式有以下子类：

**PWM 模式 A:**  $CxOCTRL=3'b110$  时，开启 PWM 模式 A。向上计数时， $TMRx_C1DT > TMRx_CVAL$  时  $C1ORAW$  输出高电平，否则为低电平；向下计数时， $TMRx_C1DT < TMRx_CVAL$  时  $C1ORAW$  输出低电平，否则为高电平。若要使用 PWM 模式 A，可按如下方式配置。

- 配置 TMRx\_PR 寄存器，设置 PWM 周期。
- 配置 TMRx\_CxDT 寄存器，设置 PWM 占空比。

-配置 TMRx\_CM1/CM2 寄存器 CxOCTRL 位为 3'b110，设置输出模式为 PWM 模式 A。  
-配置 TMRx\_DIV 寄存器，设置计数器计数频率。

-配置 TMRx\_CTRL1 寄存器 TWCMSEL[1:0]位，设置计数器计数模式。  
-配置 TMRx\_CCTRL 寄存器 CxP 位、CxCP 位，设置输出极性。  
-配置 TMRx\_CCTRL 寄存器 CxEEN 位、CxCEEN 位，使能通道输出。  
-配置 TMRx\_BRK 寄存器 OEN 位，使能 TMRx 输出。  
-配置 TMRx 输出通道对应 GPIO 为对应的复用模式。  
-配置 TMRx\_CTRL1 寄存器 TMREN 位，使能 TMRx 计数。

**PWM 模式 B:** CxOCTRL=3'b111 时，开启 PWM 模式 B。向上计数时，TMRx\_C1DT>TMRx\_CVAL 时 C1ORAW 输出低电平，否则为高电平；向下计数时，TMRx\_C1DT<TMRx\_CVAL 时 C1ORAW 输出高电平，否则为低电平。

**强制输出模式:** CxOCTRL=3'b100/101 时，开启强制输出模式。此时，CxORAW 信号的电平被强制输出为配置的电平，而与计数值无关。虽然输出信号不依赖于比较结果，但通道标志位和 DMA 请求仍依赖于比较结果。

**输出比较模式:** CxOCTRL=3'b001/010/011 时，开启输出比较模式。此时，当计数值与 CxDT 值匹配时，CxORAW 强制输出高电平（CxOCTRL=3'b001）、低电平（CxOCTRL=3'b010）或进行电平翻转（CxOCTRL=3'b011）。

**单周期模式:** PWM 模式的特例，将 OCMEN 位置 1 可开启单周期模式，此模式下，仅在当前计数周期中进行比较匹配，完成当前计数后，TMREN 位清 0，因此仅输出一个脉冲。当配置为向上计数模式时，需要严格配置 CVAL<CxDT≤PR；向下计数时，需严格配置 CVAL>CxDT。

**快速输出模式:** 将 CxOIEN 位置 1 可开启此功能，开启后 CxORAW 电平值不再在计数值与 CxDT 匹配时变化，而是在当前计数周期开始时，也就是说，比较结果被提前了，计数器值与通道寄存器（TMR1\_CxDT）的比较结果将会提前决定 CxORAW 的电平。

图 14-113 展示了输出比较模式（翻转）的例子，C1DT=0x3，当计数值等于 0x3 时，输出电平 C1OUT 被翻转。

图 14-114 展示了计数器向上计数与 PWM 模式 A 配合的例子，PR=0x32，CxDT 配置为不同的值时输出时输出信号的翻转情况。

图 14-115 展示了计数器中央双向对齐计数与 PWM 模式 A 配合的例子，PR=0X32，CxDT 配置为不同的值时输出时输出信号的翻转情况。

图 14-116 展示了计数器向上计数与单周期模式下 PWM 模式 B 配合的例子，计数器仅计数了一个周期，输出信号在这个周期中只输出了一个脉冲。

图 14-113 计数值与C1DT值匹配时翻转C1ORAW

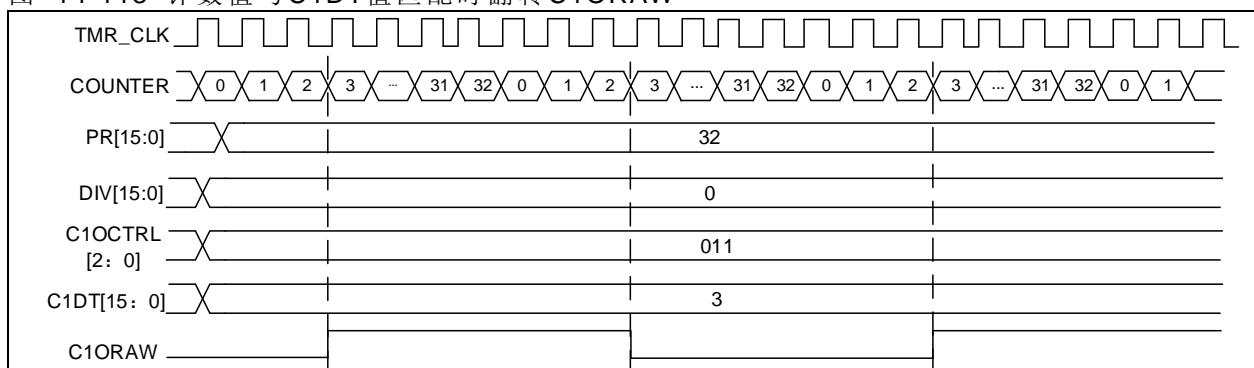


图 14-114 向上计数下 PWM 模式 A

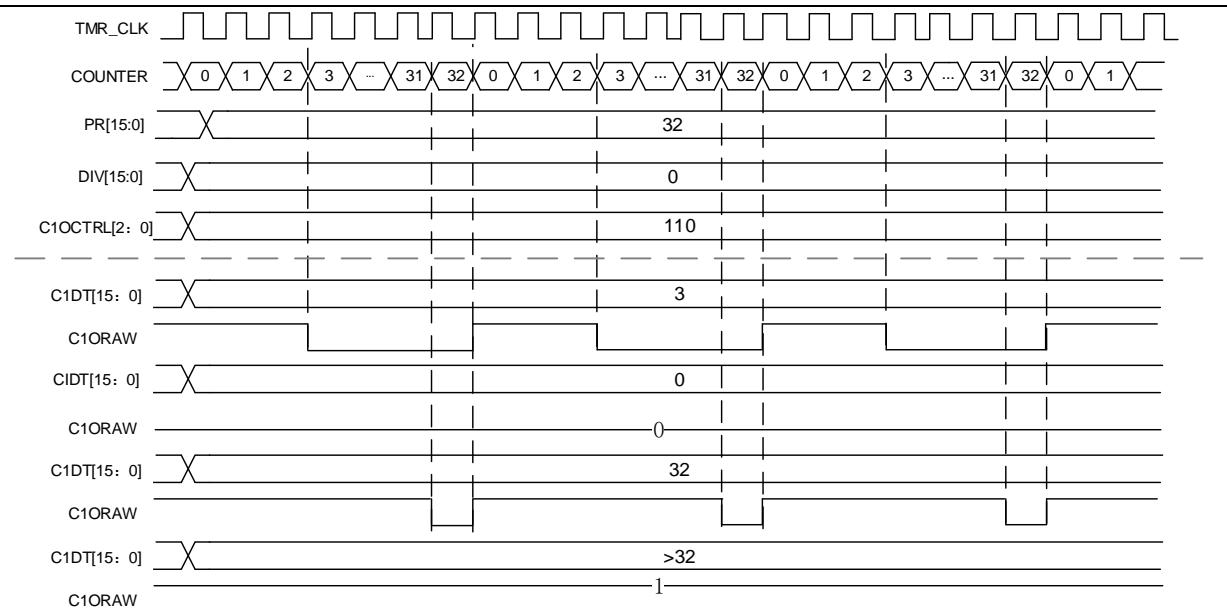


图 14-115 中央双向对齐计数下 PWM 模式

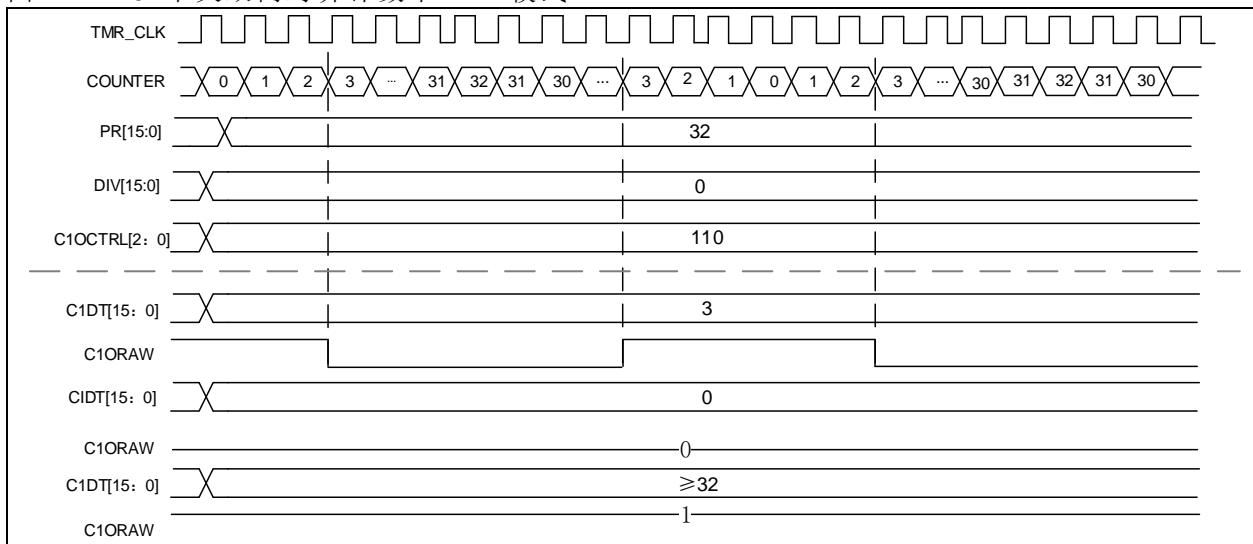
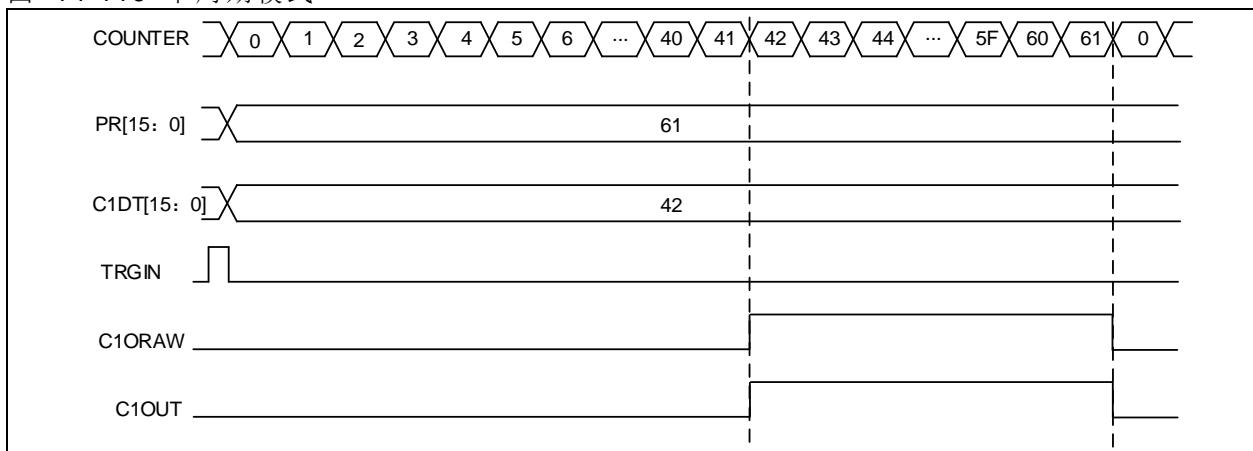


图 14-116 单周期模式



### 主定时器事件输出

当 TMR 作为主定时器时，可选择如下信号源作为 TRGOUT 信号输出到次定时器，选择信号为 TMRxCTRL2 寄存器 PTOS 位。

-PTOS=3'b000, TRGOUT 输出软件溢出事件 (TMRx\_SWEVT 寄存器 OVFSWTR 位)。

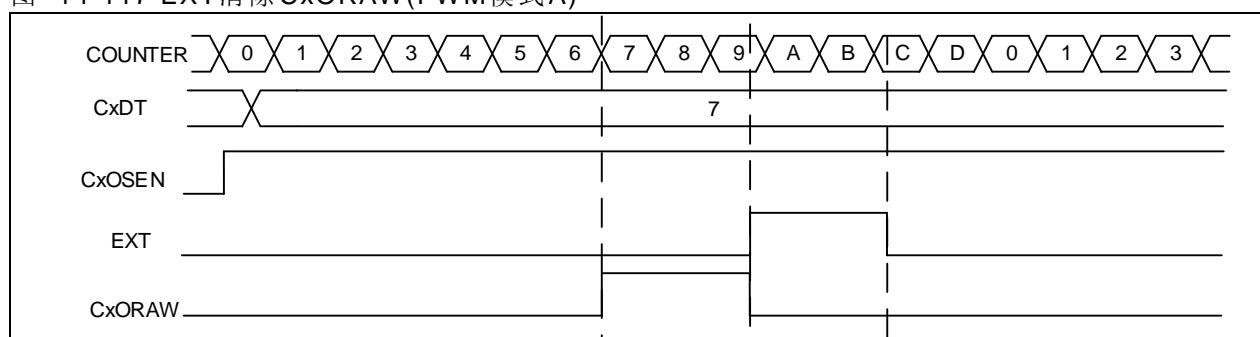
- PTOS=3'b001, TRGOUT 输出计数器使能信号。
- PTOS=3'b010, TRGOUT 输出计数器溢出事件。
- PTOS=3'b011, TRGOUT 输出捕获、比较事件。
- PTOS=3'b100, TRGOUT 输出 C1ORAW 信号。
- PTOS=3'b101, TRGOUT 输出 C2ORAW 信号。
- PTOS=3'b110, TRGOUT 输出 C3ORAW 信号。
- PTOS=3'b111, TRGOUT 输出 C4ORAW 信号。

### CxORAW 信号清除

将 CxOSEN 位置 1 后, 指定通道的 CxORAW 信号由 EXT 高电平清 0, 在下一次溢出事件发生前 CxORAW 信号无法被改变。

强制输出模式时, CxORAW 信号清除功能不可用, 只有在输出比较模式或 PWM 模式, 此功能有效。下图显示了使用 EXT 信号清除 CxORAW 的例子, 当 EXT 为高电平期间, 原本为高电平的 CxORAW 信号被拉低, 当 EXT 为低电平时, CxORAW 根据计数值和 CxDT 比较结果输出电平。

图 14-117 EXT 清除 CxORAW(PWM 模式 A)



### 死区插入

高级定时器通道 1 至 3 包含一组反向通道输出, 通过 CxCEN 使能, 通过 CxCP 配置极性。CxOUT 和 CxCOUT 的输出状态见表 14-17。

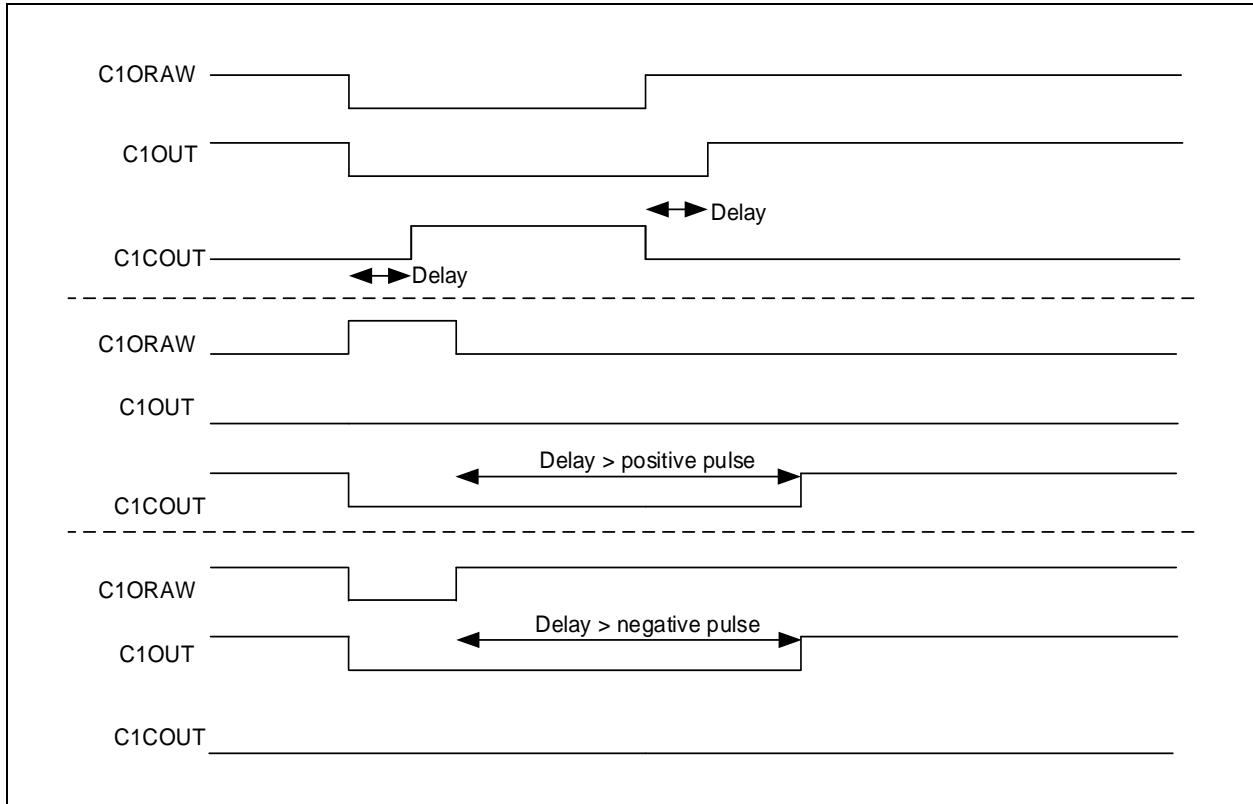
当转换为 IDLEF 状态, 即 OEN 下降到 0, 死区被激活。

将 CxEN 和 CxCEN 位置 1 后, 通过配置 DTC[7: 0]死区发生器, 可插入不同时长的死区。插入死区后, CxOUT 的上升沿延迟于参考信号的上升沿; CxCOUT 的上升沿延迟于参考信号的下降沿。

如果延迟大于当前有效的输出宽度, C1OUT 和 C1COUT 不会产生相应的脉冲, 死区时间应小于有效的输出宽度。

下列图显示了 CxP=0、CxCP=0、OEN=1、CxEN=1 并且 CxCEN=1 时死区插入的例子

图 14-118 带死区插入的互补输出



#### 14.6.3.5 TMR刹车功能

开启刹车功能后 (BRKEN 位置 1)，CxOUT 和 CxCOUT 由 OEN、FCSODIS、FCSOEN、CxIOS 和 CxCIOS 共同控制。但 CxOUT 和 CxCOUT 输出总是不能同时处于有效电平上的。详见表 14-17 带刹车功能的互补输出通道 CxOUT 和 CxCOUT 的控制位。

刹车信号来源可以是刹车输入管脚、时钟失效事件，刹车输入信号的极性由 BRKV 位控制。

当发生刹车事件时，有下述动作：

- OEN 位异步清零，通道输出状态由 FCSODIS 位选择。关闭 MCU 的振荡器不影响该功能。
- OEN 被清零后，通道输出电平由 CxIOS 位设定。如果 FCSODIS=0，则定时器输出使能被禁止，否则输出使能始终为高。
- 当使用互补输出时：
- — 输出最开始处于复位状态，也就是无效的状态（取决于极性）。这是异步操作，定时器有无时钟并不影响此功能。
  - 定时器的时钟如果有效，会开启死区生成功能，CxIOS 和 CxCIOS 位用来配置死区之后的电平。即使在这种情况下，CxOUT 和 CxCOUT 也不能被同时驱动到有效的电平。
  - 注意，由于 OEN 位同步逻辑，死区时间较通常情况会延长一段时间（大约 2 个 clk\_tmr 的时钟周期）。
  - 如果 FCSODIS=0，定时器释放使能输出，否则保持使能输出；或一旦 CxEN 与 CxCEN 之一变高时，使能输出变为高。
- 如果开启了刹车中断或 DMA 功能，刹车状态标志将置 1，并产生刹车中断或 DMA 请求。
- 如果将 AOEN 位置 1，在下一个溢出事件时 OEN 位被自动置 1。

注意：刹车输入电平有效时，OEN 不能被设置，状态标志 BRKIF 也不能被清除。

图 14-119 TMR输出控制

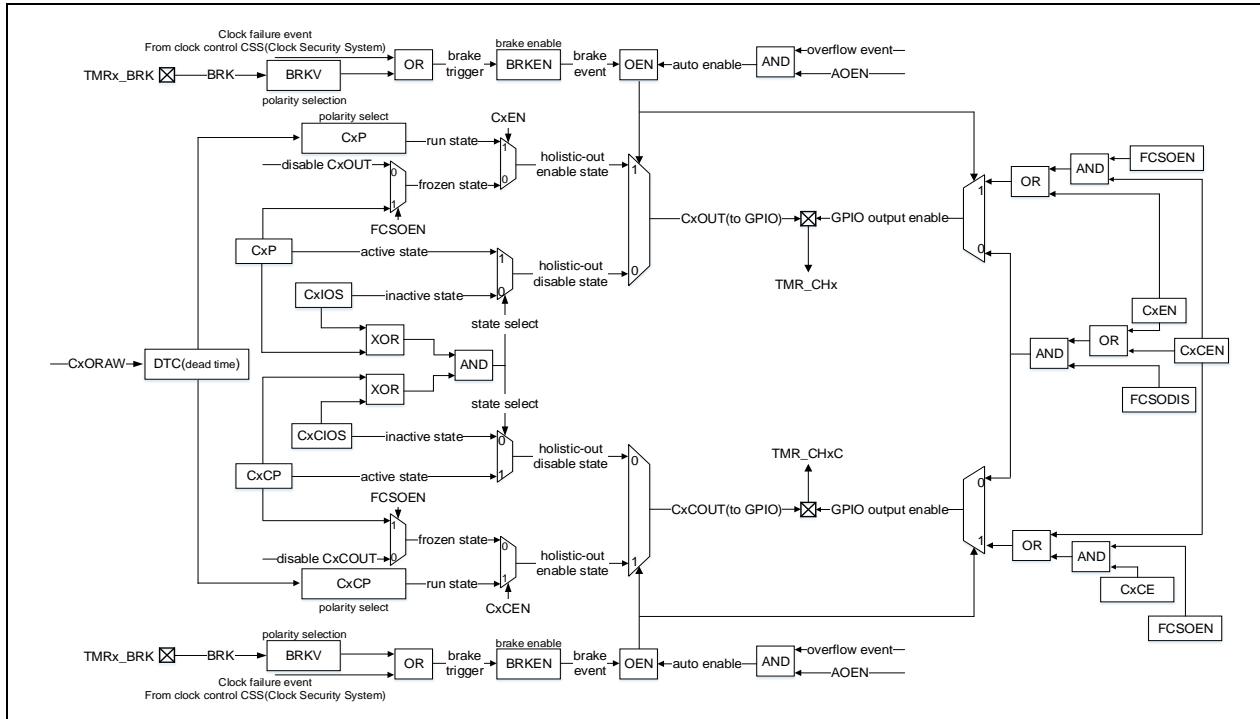
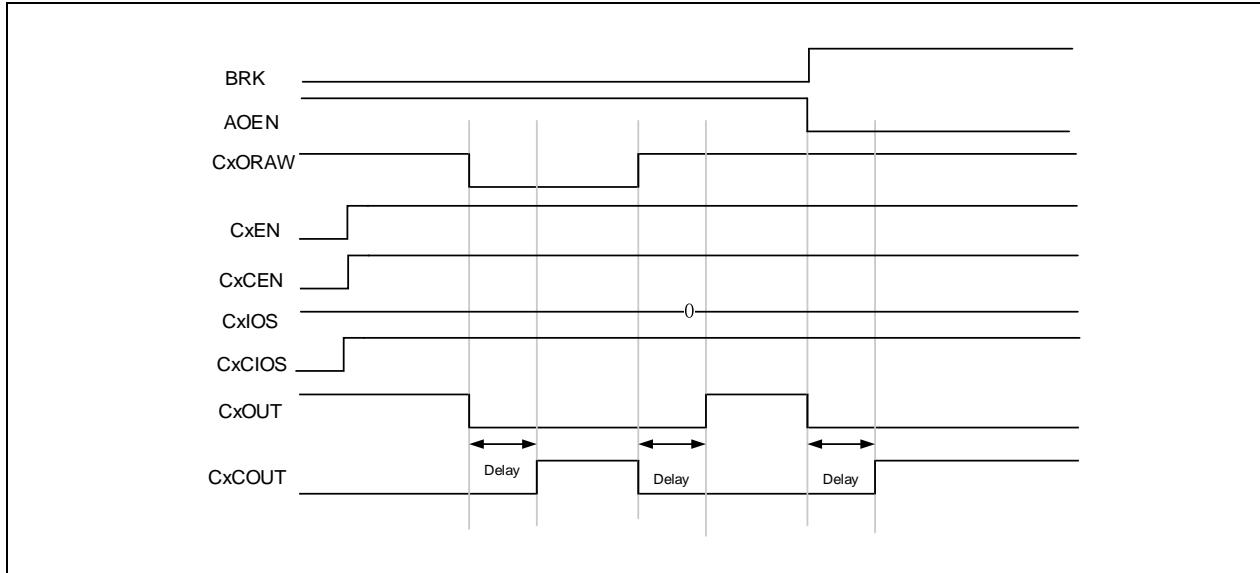


图 14-120 TMR刹车功能的例子



### 14.6.3.6 TMR同步

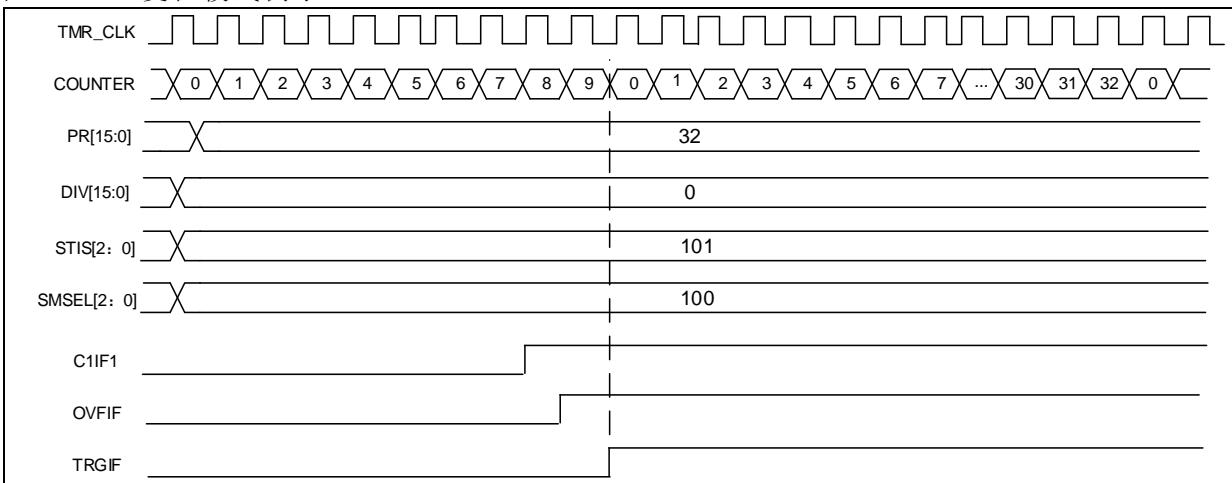
主次定时器之间可由内部连接信号进行同步。主定时器可由 PTOS[2: 0]位选择主定时器输出，即同步信息；次定时器由 SMSEL[2: 0]位选择从模式，即次定时器的工作模式。

定时器从模式有以下几种：

#### 从模式：复位模式

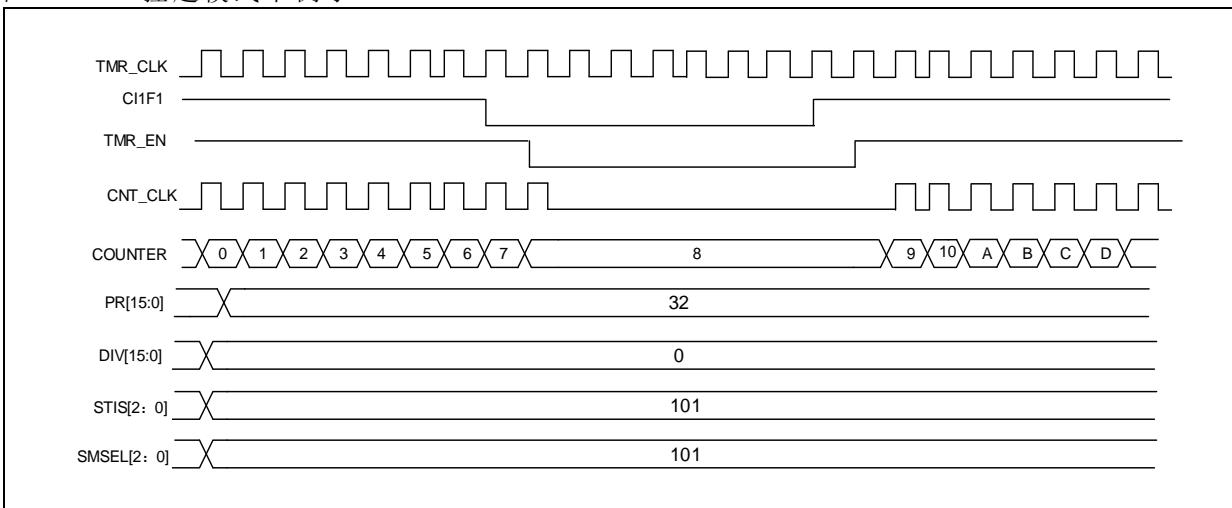
选中的触发信号将复位计数器和预分频器，若 OVFS 位为 0，将产生一个溢出事件。

图 14-121 复位模式例子

**从模式：挂起模式**

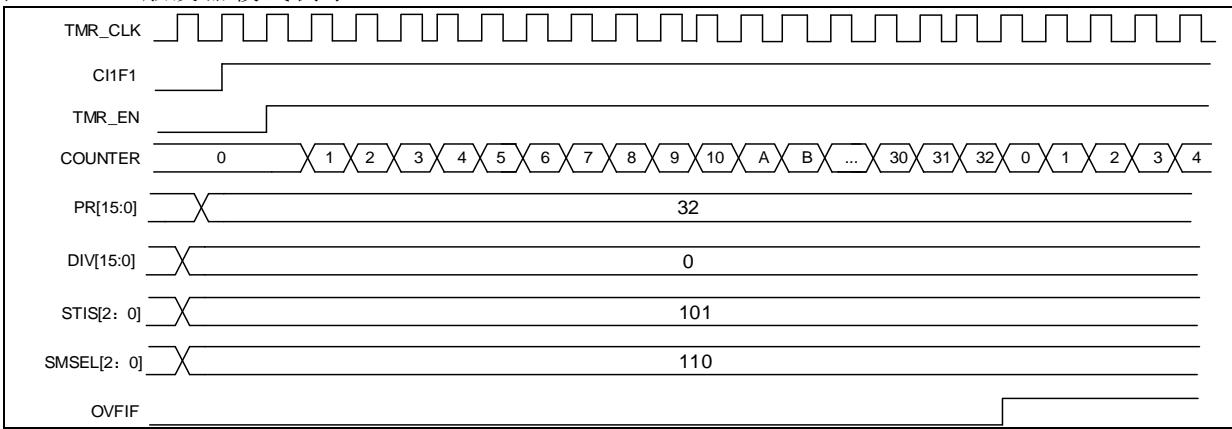
挂起模式下，计数的计数和停止受选中触发输入信号控制，当触发输入为高电平时计数器开始计数；当为低电平时，计数器暂停计数。

图 14-122 挂起模式下例子

**从模式：触发模式**

计数器将在选中的触发输入上升沿启动计数（将 TMR\_EN 置 1）。

图 14-123 触发器模式例子



定时器的同步的更多实例详见 [14.2.3.5](#) 节。

**14.6.3.7 调试模式**

当微控制器进入调试模式（Cortex®-M4 核心停止）时，将 DEBUG 模块中的 TMR1\_PAUSE 置 1，可以使 TMR1 计数器暂停计数。

## 14.6.4 TMR1寄存器描述

必须以字（32位）的方式操作这些外设寄存器。

下表中将 TMR1 的所有寄存器映射到一个 16 位可寻址（编址）空间

表 14-16 TMR1寄存器映像和复位值

寄存器简称	基址偏移量	复位值
TMR1_CTRL1	0x00	0x0000
TMR1_CTRL2	0x04	0x0000
TMR1_STCTRL	0x08	0x0000
TMR1_IDEN	0x0C	0x0000
TMR1ISTS	0x10	0x0000
TMR1_SWEVT	0x14	0x0000
TMR1_CM1	0x18	0x0000
TMR1_CM2	0x1C	0x0000
TMR1_CCTRL	0x20	0x0000
TMR1_CVAL	0x24	0x0000
TMR1_DIV	0x28	0x0000
TMR1_PR	0x2C	0x0000
TMR1_RPR	0x30	0x0000
TMR1_C1DT	0x34	0x0000
TMR1_C2DT	0x38	0x0000
TMR1_C3DT	0x3C	0x0000
TMR1_C4DT	0x40	0x0000
TMR1_BRK	0x44	0x0000
TMR1_DMACTRL	0x48	0x0000
TMR1_DMADT	0x4C	0x0000

### 14.6.4.1 TMR1控制寄存器1 (TMR1\_CTRL1)

域	简称	复位值	类型	功能
位 15: 10	保留	0x00	resd	保持默认值。
位 9: 8	CLKDIV	0x0	rw	时钟除频 (Clock divider) 此位用于设置数字滤波器采样频率 $f_{DTS}$ 和定时器时钟频率 $f_{CK\_INT}$ 之间的分频比，也用于调整死区时间的时基 $T_{DTS}$ 和定时器时钟周期 $T_{CK\_INT}$ 的分频比。 00: 无除频, $f_{DTS}=f_{CK\_INT}$ ; 01: 2 除频, $f_{DTS}=f_{CK\_INT}/2$ ; 10: 4 除频, $f_{DTS}=f_{CK\_INT}/4$ ; 11: 保留。
位 7	PRBEN	0x0	rw	周期缓冲使能 (Period buffer enable) 0: 缓冲关闭; 1: 缓冲开启。
位 6: 5	TWCMSEL	0x0	rw	中央双向对齐计数模式选择 (Two-way count mode selection) 00: 单向对齐计数模式，方向由 OWCDIR 配置; 01: 中央双向对齐计数模式 1，上下交替计数，CxIF 位只在计数器向下计数时被置起; 10: 中央双向对齐计数模式 2，上下交替计数，CxIF 位只在计数器向上计数时被置起; 11: 中央双向对齐计数模式 3，上下交替计数，CxIF 位在计数器向上和向下计数时皆被置起。

位 4	OWCDIR	0x0	rw	单向对齐计数方向 (One-way count direction) 0: 向上; 1: 向下。
位 3	OCMEN	0x0	rw	单周期使能 (One cycle mode enable) 该功能用于选择溢出事件后，计数器是否停止。 0: 关闭; 1: 开启。
位 2	OVFS	0x0	rw	溢出事件源选择 (Overflow event source) 配置溢出事件或 DMA 请求来源。 0: 来源于计数器溢出、设置 OVFSWTR 位或次定时器控制器产生的溢出事件; 1: 只能来源于计数器溢出。
位 1	OVFEN	0x0	rw	溢出事件使能 (Overflow event enable) 0: 开启; 1: 关闭。
位 0	TMREN	0x0	rw	使能定时器 (TMR enable) 0: 关闭; 1: 开启。

#### 14.6.4.2 TMR1控制寄存器2 (TMR1\_CTRL2)

域	简称	复位值	类型	功能
位 15	保留	0x0	resd	保持默认值。
位 14	C4IOS	0x0	rw	通道 4 空闲输出状态 (Channel 4 idle output state)
位 13	C3CIOS	0x0	rw	通道 3 互补空闲输出状态 (Channel 3 complementary idle output state)
位 12	C3IOS	0x0	rw	通道 3 空闲输出状态 (Channel 3 idle output state)
位 11	C2CIOS	0x0	rw	通道 2 互补空闲输出状态 (Channel 2 complementary idle output state)
位 10	C2IOS	0x0	rw	通道 2 空闲输出状态 (Channel 2 idle output state)
				通道 1 互补空闲输出状态 (Channel 1 complementary idle output state)
位 9	C1CIOS	0x0	rw	输出关闭 (OEN = 0) , 死区发生后: 0: C1OUTL=0; 1: C1OUTL=1.
				通道 1 空闲输出状态 (Channel 1 idle output state)
位 8	C1IOS	0x0	rw	输出关闭 (OEN = 0) , 死区发生后: 0: C1OUT=0. 1: C1OUT=1.
				C1IN 选择 (C1IN selection)
位 7	C1INSEL	0x0	rw	0: CH1 管脚连到 C1IRAW 输入; 1: CH1、CH2 和 CH3 管脚异或结果连到 C1IRAW 输入。
				主定时器输出信号选择 (Primary TMR output selection)
				TMR1 输出到次定时器的信号选择: 000: 复位; 001: 使能; 010: 溢出; 011: 比较脉冲; 100: C1ORAW 信号; 101: C2ORAW 信号; 110: C3ORAW 信号; 111: C4ORAW 信号。
位 6: 4	PTOS	0x0	rw	DMA 请求源 (DMA request source)
				DMA 请求来源。 0: 通道事件; 1: 溢出事件。
位 3	DRS	0x0	rw	通道控制位刷新选择 (Channel control bit refresh select)
				对具有互补输出的通道, 如果通道控制位有缓存时: 0: 通过设置 HALL 位刷新控制位; 1: 通过设置 HALL 位或 TRGIN 的上升沿刷新控制位。
位 1	保留	0x0	resd	保持默认值。

位 0	CBCTRL	0x0	rw	通道缓存控制 (Channel buffer control) 对具有互补输出的通道： 0: CxEN, CxCEN 和 CxOCTRL 位无缓存； 1: CxEN, CxCEN 和 CxOCTRL 位有缓存。
-----	--------	-----	----	--

#### 14.6.4.3 TMR1次定时器控制寄存器 (TMR1\_STCTRL)

域	简称	复位值	类型	功能
位 15	ESP	0x0	rw	外部信号极性 (External signal polarity) 用于选择外部方式。 0: 高电平或上升沿； 1: 低电平或下降沿。
位 14	ECMBEN	0x0	rw	外部时钟模式 B 使能 (External clock mode B enable) 用于启用外部时钟模式 B 0: 关闭； 1: 启开。
位 13: 12	ESDIV	0x0	rw	外部信号除频 (External signal divide) 用于选择降低外部触发频率的除频。 00: 关闭分频； 01: 2 分频； 10: 4 分频； 11: 8 分频。
位 11: 8	ESF	0x0	rw	外部信号滤波 (External signal filter) 用于过滤外部信号，当外部信号产生了 N 次之后才能被采样。 0000: 无滤波器，以 $f_{DTS}$ 采样 0001: $f_{SAMPLING} = f_{CK\_INT}$ , N=2; 0010: $f_{SAMPLING} = f_{CK\_INT}$ , N=4; 0011: $f_{SAMPLING} = f_{CK\_INT}$ , N=8; 0100: $f_{SAMPLING} = f_{DTS}/2$ , N=6; 0101: $f_{SAMPLING} = f_{DTS}/2$ , N=8; 0110: $f_{SAMPLING} = f_{DTS}/4$ , N=6; 0111: $f_{SAMPLING} = f_{DTS}/4$ , N=8; 1000: $f_{SAMPLING} = f_{DTS}/8$ , N=6; 1001: $f_{SAMPLING} = f_{DTS}/8$ , N=8; 1010: $f_{SAMPLING} = f_{DTS}/16$ , N=5; 1011: $f_{SAMPLING} = f_{DTS}/16$ , N=6; 1100: $f_{SAMPLING} = f_{DTS}/16$ , N=8; 1101: $f_{SAMPLING} = f_{DTS}/32$ , N=5; 1110: $f_{SAMPLING} = f_{DTS}/32$ , N=6; 1111: $f_{SAMPLING} = f_{DTS}/32$ , N=8。
位 7	STS	0x0	rw	次定时器同步 (Subordinate TMR synchronization) 该位开启后，主次定时器可实现高度同步。 0: 关闭； 1: 开启。
位 6: 4	STIS	0x0	rw	次定时器输入选择 (Subordinate TMR input selection) 用于次定时器的输入选择。 000: 内部选择 0 (IS0)； 001: 内部选择 1 (IS1)； 010: 内部选择 2 (IS2)； 011: 内部选择 3 (IS3)； 100: C1IRAW 的输入检测器 (C1INC)； 101: 滤波输入 1 (C1IF1)； 110: 滤波输入 2 (C2IF2)； 111: 外部输入 (EXT)。 关于每个定时器中 ISx 的细节，参见表 14-14。
位 3	COSSEL	0x0	rw	通道输出开关源选择 (Channel output switch selection) 该用于选择 CxORAW 的开关来源 0: 选择 EXT 作为 CxORAW 开关来源； 1: 选择 CxORAW_OFF 作为 CxORAW 开关来源
位 2: 0	SMSEL	0x0	rw	次定时器模式选择 (Subordinate TMR mode selection)

000: 关闭从模式;  
 001: 编码模式 A;  
 010: 编码模式 B;  
 011: 编码模式 C;  
 100: 复位模式 - TRGIN 输入上升沿时, 重新初始化计数器;  
 101: 挂起模式 - TRGIN 输入高电平时, 计数器计数;  
 110: 触发模式 - TRGIN 输入上升沿时, 产生触发事件;  
 111: 外部时钟模式 A - TRGIN 输入上升沿提供时钟;  
 注: 编码器模式 A/B/C 配置方法请查看计数模式章节。

#### 14.6.4.4 TMR1 DMA/中断使能寄存器 (TMR1\_IDEN)

域	简称	复位值	类型	功能
位 15	保留	0x0	resd	保持默认值。
位 14	TDEN	0x0	rw	触发 DMA 请求使能 (Trigger DMA request enable) 0: 关闭; 1: 开启。
位 13	HALLDE	0x0	rw	HALL DMA 请求使能 (HALL DMA request enable) 0: 关闭; 1: 开启。
位 12	C4DEN	0x0	rw	通道 4 的 DMA 请求使能 (Channel 4 DMA request enable) 0: 关闭; 1: 开启。
位 11	C3DEN	0x0	rw	通道 3 的 DMA 请求使能 (Channel 3 DMA request enable) 0: 关闭; 1: 开启。
位 10	C2DEN	0x0	rw	通道 2 的 DMA 请求使能 (Channel 2 DMA request enable) 0: 关闭; 1: 开启。
位 9	C1DEN	0x0	rw	通道 1 的 DMA 请求使能 (Channel 1 DMA request enable) 0: 关闭; 1: 开启。
位 8	OVFDEN	0x0	rw	溢出事件的 DMA 请求使能 (overflow event DMA request enable) 0: 关闭; 1: 开启。
位 7	BRKIE	0x0	rw	刹车中断使能 (Brake interrupt enable) 0: 关闭; 1: 开启。
位 6	TIEN	0x0	rw	触发中断使能 (Trigger interrupt enable) 0: 关闭; 1: 开启。
位 5	HALLIEN	0x0	rw	HALL 中断使能 (HALL interrupt enable) 0: 关闭; 1: 开启。
位 4	C4IEN	0x0	rw	通道 4 中断使能 (Channel 4 interrupt enable) 0: 关闭; 1: 开启。
位 3	C3IEN	0x0	rw	通道 3 中断使能 (Channel 3 interrupt enable) 0: 关闭; 1: 开启。
位 2	C2IEN	0x0	rw	通道 2 中断使能 (Channel 2 interrupt enable) 0: 关闭; 1: 开启。
位 1	C1IEN	0x0	rw	通道 1 中断使能 (Channel 1 interrupt enable) 0: 关闭;

位 0	OVFIEN	0x0	rw	1: 开启。 溢出中断使能 (Overflow interrupt enable)
				0: 关闭; 1: 开启。

#### 14.6.4.5 TMR1中断状态寄存器 (TMR1\_ISTS)

域	简称	复位值	类型	功能
位 15: 13	保留	0x0	resd	保持默认值。
位 12	C4RF	0x0	rw0c	通道 4 再捕获标记 (Channel 4 recapture flag) 见 C1RF 的描述。
位 11	C3RF	0x0	rw0c	通道 3 再捕获标记 (Channel 3 recapture flag) 见 C1RF 的描述。
位 10	C2RF	0x0	rw0c	通道 2 再捕获标记 (Channel 2 recapture flag) 见 C1RF 的描述。
位 9	C1RF	0x0	rw0c	通道 1 再捕获标记 (Channel 1 recapture flag) C1IF 的状态已经为'1'时是否再次发生了捕获, 由硬件置'1', 写'0'清除。 0: 无捕获发生; 1: 捕获发生。
位 8	保留	0x0	resd	保持默认值。
位 7	BRKIF	0x0	rw0c	刹车中断标记 (Brake interrupt flag) 用于标记刹车输入的电平是否有效, 由硬件置'1', 写'0'清除。 0: 无效; 1: 有效。
位 6	TRGIF	0x0	rw0c	触发中断标记 (Trigger interrupt flag) 当发生触发事件时由硬件置'1', 写'0'清除。 0: 无触发事件发生; 1: 发生触发事件。 触发事件: 在 TRGIN 接收到有效边沿, 或挂起模式下接收到任意边沿。
位 5	HALLIF	0x0	rw0c	HALL 中断标记 (HALL interrupt flag) 当发生触发事件时由硬件置'1', 写'0'清除。 0: 无 HALL 事件发生; 1: 发生 HALL 事件。 HALL 事件: CxEN、CxCEN、CxOCTRL 已被更新。
位 4	C4IF	0x0	rw0c	通道 4 中断标记 (Channel 4 interrupt flag) 见 C1IF 的描述。
位 3	C3IF	0x0	rw0c	通道 3 中断标记 (Channel 3 interrupt flag) 见 C1IF 的描述。
位 2	C2IF	0x0	rw0c	通道 2 中断标记 (Channel 2 interrupt flag) 见 C1IF 的描述。
位 1	C1IF	0x0	rw0c	通道 1 中断标记 (Channel 1 interrupt flag) 若通道 1 为输入模式时: 捕获事件发生时由硬件置'1', 由软件清'0'或读 TMR1_C1DT 清'0'。 0: 无捕获事件发生; 1: 发生捕获事件。 若通道 1 为输出模式时: 比较事件发生时由硬件置'1', 由软件清'0'。 0: 无比较事件发生; 1: 发生比较事件。
位 0	OVFIF	0x0	rw0c	溢出中断标记 (Overflow interrupt flag) 当溢出事件发生时由硬件置'1', 由软件清'0'。 0: 无溢出事件发生; 1: 发生溢出事件, 若 TMR1_CTRL1 的 OVFEN=0、OVFS=0 时: - 当 TMR1_SWEVE 寄存器的 OVFG=1 时产生溢出事件;

- 当计数值 CVAL 被触发事件重初始化时产生溢出事件。

#### 14.6.4.6 TMR1软件事件寄存器 (TMR1\_SW\_EVT)

域	简称	复位值	类型	功能
位 15: 8	保留	0x00	resd	保持默认值。
位 7	BRKSWTR	0x0	wo	软件触发刹车事件 (Brake event triggered by software) 通过软件触发一个刹车事件。 0: 无作用; 1: 制造一个刹车事件。
位 6	TRGSWTR	0x0	wo	软件触发触发事件 (Trigger event triggered by software) 通过软件触发一个触发事件。 0: 无作用; 1: 制造一个触发事件。
位 5	HALLSWTR	0x0	wo	软件触发 HALL 事件 (HALL event triggered by software) 通过软件产生一个 HALL 事件。 0: 无作用; 1: 产生一个 HALL 事件。 注: 该位只对拥有互补输出的通道有效。
位 4	C4SWTR	0x0	wo	软件触发通道 4 事件 (Channel 4 event triggered by software) 见 C1M 的描述。
位 3	C3SWTR	0x0	wo	软件触发通道 3 事件 (Channel 3 event triggered by software) 见 C1M 的描述。
位 2	C2SWTR	0x0	wo	软件触发通道 2 事件 (Channel 2 event triggered by software) 见 C1M 的描述。
位 1	C1SWTR	0x0	wo	C1SWTR: 软件触发通道 1 事件 (Channel 1 event triggered by software) 通过软件触发一个通道 1 事件。 0: 无作用; 1: 制造一个通道 1 事件。
位 0	OVFSWTR	0x0	wo	软件触发溢出事件 (Overflow event triggered by software) 通过软件触发一个溢出事件。 0: 无作用; 1: 制造一个溢出事件。

#### 14.6.4.7 TMR1通道模式寄存器1 (TMR1\_CM1)

通道可用于输入（捕获模式）或输出（比较模式），通道的方向由相应的 CxC 位定义。该寄存器其它位的作用在输入和输出模式下不同。CxOx 描述了通道在输出模式下的功能，CxIx 描述了通道在输入模式下的功能。因此必须注意，同一个位在输出模式和输入模式下的功能是不同的。

##### 输出比较模式

域	简称	复位值	类型	功能
位 15	C2OSEN	0x0	rw	通道 2 输出开关使能 (Channel 2 output switch enable)
位 14: 12	C2OCTRL	0x0	rw	通道 2 输出控制 (Channel 2 output control)
位 11	C2OBEN	0x0	rw	通道 2 输出缓存使能 (Channel 2 output buffer enable)
位 10	C2OIEN	0x0	rw	通道 2 输出立即使能 (Channel 2 output immediately enable)
通道 2 配置 (Channel 2 configure) 当 C2EN=0'时，这些位用于选择通道 2 为输出或输入，以及输入时的映射选择：				
位 9: 8	C2C	0x0	rw	00: 输出; 01: 输入，C2IN 映射在 C2IFP2 上; 10: 输入，C2IN 映射在 C1IFP2 上; 11: 输入，C2IN 映射在 STI 上，只有在 STIS 选择内部触发输入时才工作。
位 7	C1OSEN	0x0	rw	通道 1 输出开关使能 (Channel 1 output switch enable) 0: EXT 输入不影响 C1ORAW;

位 6: 4	C1OCTRL	0x0	rw	<p>1: 当 EXT 输入高电平时, 将 C1ORAW 清 0。            通道 1 输出控制 (Channel 1 output control)            这些位用于设置原始信号 C1ORAW 的工作状态。            000: 断开。断开 C1ORAW 到 C1OUT 的输出;            001: 设置 C1ORAW 为高: TMR1_CVAL=TMR1_C1DT 时。            010: 设置 C1ORAW 为低: TMR1_CVAL=TMR1_C1DT 时。            011 : 切换 C1ORAW 的电平 : 当 TMR1_CVAL=TMR1_C1DT 时。            100: 固定 C1ORAW 为低。            101: 固定 C1ORAW 为高。            110: PWM 模式 A            -OWCDIR=0, 若 TMR1_C1DT&gt;TMR1_CVAL 时设置 C1ORAW 为高, 否则为低;            -OWCDIR=1, 若 TMR1_C1DT &lt; TMR1_CVAL 时设置 C1ORAW 为低, 否则为高。            111: PWM 模式 B            -OWCDIR=0, 若 TMR1_C1DT &gt; TMR1_CVAL 时设置 C1ORAW 为低, 否则为高;            -OWCDIR=1, 若 TMR1_C1DT &lt; TMR1_CVAL 时设置 C1ORAW 为高, 否则为低。            注: 除'000'外, 其余配置下 C1OUT 将连接到 C1ORAW, C1OUT 的输出电平除了会根据 C1ORAW 变化外, 还与 CCTRL 所配置的输出极性有关。</p>
位 3	C1OBEN	0x0	rw	<p>通道 1 输出缓存使能 (Channel 1 output buffer enable)            0: 关闭 TMR1_C1DT 的缓存功能, 写入 TMR1_C1DT 的内容会立即生效。            1: 启用 TMR1_C1DT 的缓存功能, 写入 TMR1_C1DT 的内容将保存到缓存寄存器中, 当发生溢出事件时再更新到 TMR1_C1DT 中。</p>
位 2	C1OIEN	0x0	rw	<p>通道 1 输出立即使能 (Channel 1 output immediately enable)            在 PWM 模式 A 或模式 B 下, 该位能够缩短触发事件到通道 1 的输出响应间的时间。            0: 需要比较 CVAL 与 C1DT 的值之后再产生输出。            1: 无需比较 CVAL 与 C1DT 的值, 当发生触发事件时立即产生输出。</p>
位 1: 0	C1C	0x0	rw	<p>通道 1 配置 (Channel 1 configure)            当 C1EN='0'时, 这些位用于选择通道 1 为输出或输入, 以及输入时的映射选择:            00: 输出;            01: 输入, C1IN 映射在 C1IFP1 上;            10: 输入, C1IN 映射在 C2IFP1 上;            11: 输入, C1IN 映射在 STCI 上, 只有在 STIS 选择内部触发输入时才工作。</p>

### 输入模式

域	简称	复位值	类型	功能
位 15: 12	C2DF	0x0	rw	通道 2 滤波器 (Channel 2 digital filter)
位 11: 10	C2IDIV	0x0	rw	通道 2 分频系数 (Channel 2 input divider)
位 9: 8	C2C	0x0	rw	<p>通道 2 配置 (Channel 2 configure)            当 C2EN='0'时, 这些位用于选择通道 2 为输出或输入, 以及输入时的映射选择:            00: 输出;            01: 输入, C2IN 映射在 C2IFP2 上;            10: 输入, C2IN 映射在 C1IFP2 上;            11: 输入, C2IN 映射在 STCI 上, 只有在 STIS 选择内部触发输入时才工作。</p>
位 7: 4	C1DF	0x0	rw	通道 1 滤波器 (Channel 1 digital filter)

这些位用于配置通道 1 的滤波器。滤波的个数为 N，则表示发生了 N 次采样事件后输入边沿才能通过滤波器：

- 0000: 无滤波器，以  $f_{DTS}$  采样
- 1000: 采样频率  $f_{SAMPLING} = f_{DTS}/8$ , N=6
- 0001: 采样频率  $f_{SAMPLING} = f_{CK\_INT}$ , N=2
- 1001: 采样频率  $f_{SAMPLING} = f_{DTS}/8$ , N=8
- 0010: 采样频率  $f_{SAMPLING} = f_{CK\_INT}$ , N=4
- 1010: 采样频率  $f_{SAMPLING} = f_{DTS}/16$ , N=5
- 0011: 采样频率  $f_{SAMPLING} = f_{CK\_INT}$ , N=8
- 1011: 采样频率  $f_{SAMPLING} = f_{DTS}/16$ , N=6
- 0100: 采样频率  $f_{SAMPLING} = f_{DTS}/2$ , N=6
- 1100: 采样频率  $f_{SAMPLING} = f_{DTS}/16$ , N=8
- 0101: 采样频率  $f_{SAMPLING} = f_{DTS}/2$ , N=8
- 1101: 采样频率  $f_{SAMPLING} = f_{DTS}/32$ , N=5
- 0110: 采样频率  $f_{SAMPLING} = f_{DTS}/4$ , N=6
- 1110: 采样频率  $f_{SAMPLING} = f_{DTS}/32$ , N=6
- 0111: 采样频率  $f_{SAMPLING} = f_{DTS}/4$ , N=8
- 1111: 采样频率  $f_{SAMPLING} = f_{DTS}/32$ , N=8

通道 1 分频系数 (Channel 1 input divider)

这些位定义了通道 1 的分频系数。

- 00: 不分频，每一个有效的边沿都会产生一次输入；
- 01: 每 2 个有效的边沿产生一次输入；
- 10: 每 4 个有效的边沿产生一次输入；
- 11: 每 8 个有效的边沿产生一次输入。

注：C1EN='0'时，分频系数复位。

通道 1 配置 (Channel 1 configure)

当 C1EN='0'时，这些位用于选择通道 1 为输出或输入，以及输入时的映射选择：

- 00: 输出；
- 01: 输入，C1IN 映射在 C1IFP1 上；
- 10: 输入，C1IN 映射在 C2IFP1 上；
- 11: 输入，C1IN 映射在 STCI 上，只有在 STIS 选择内部触发输入时才工作。

#### 14.6.4.8 TMR1通道模式寄存器2 (TMR1\_CM2)

参看以上 CM1 寄存器描述

##### 输出比较模式

域	简称	复位值	类型	功能
位 15	C4OSEN	0x0	rw	通道 4 输出开关使能 (Channel 4 output switch enable)
位 14: 12	C4OCTRL	0x0	rw	通道 4 输出控制 (Channel 4 output control)
位 11	C4OBEN	0x0	rw	通道 4 输出缓存使能 (Channel 4 output buffer enable)
位 10	C4OIEN	0x0	rw	通道 4 输出立即使能 (Channel 4 output immediately enable)

通道 4 配置 (Channel 4 configure)

当 C4EN='0'时，这些位用于选择通道 4 为输出或输入，以及输入时的映射选择：

- 00: 输出；
- 01: 输入，C4IN 映射在 C4IFP4 上；
- 10: 输入，C4IN 映射在 C3IFP4 上；
- 11: 输入，C4IN 映射在 STCI 上，只有在 STIS 选择内部触发输入时才工作。

域	简称	复位值	类型	功能
位 7	C3OSEN	0x0	rw	通道 3 输出开关使能 (Channel 3 output switch enable)
位 6: 4	C3OCTRL	0x0	rw	通道 3 输出控制 (Channel 3 output control)
位 3	C3OBEN	0x0	rw	通道 3 输出缓存使能 (Channel 3 output buffer enable)
位 2	C3OIEN	0x0	rw	通道 3 输出立即使能 (Channel 3 output immediately enable)

通道 3 配置 (Channel 3 configure)

当 C3EN='0'时，这些位用于选择通道 3 为输出或输入，以及输入时的映射选择：

- 00: 输出；

01: 输入, C3IN 映射在 C3IFP3 上;  
 10: 输入, C3IN 映射在 C4IFP3 上;  
 11: 输入, C3IN 映射在 STCI 上, 只有在 STIS 选择内部触发输入时才工作。

**输入模式**

域	简称	复位值	类型	功能
位 15: 12	C4DF	0x0	rw	通道 4 滤波器 (Channel 4 digital filter)
位 11: 10	C4IDIV	0x0	rw	通道 4 分频系数 (Channel 4 input divider)
位 9: 8	C4C	0x0	rw	通道 4 配置 (Channel 4 configure) 当 C4EN=’0’时, 这些位用于选择通道 4 为输出或输入, 以及输入时的映射选择: 00: 输出; 01: 输入, C4IN 映射在 C4IFP4 上; 10: 输入, C4IN 映射在 C3IFP4 上; 11: 输入, C4IN 映射在 STCI 上, 只有在 STIS 选择内部触发输入时才工作。
位 7: 4	C3DF	0x0	rw	通道 3 滤波器 (Channel 3 digital filter)
位 3: 2	C3IDIV	0x0	rw	通道 3 分频系数 (Channel 3 input divider)
位 1: 0	C3C	0x0	rw	通道 3 配置 (Channel 3 configure) 当 C3EN=’0’时, 这些位用于选择通道 3 为输出或输入, 以及输入时的映射选择: 00: 输出; 01: 输入, C3IN 映射在 C3IFP3 上; 10: 输入, C3IN 映射在 C4IFP3 上; 11: 输入, C3IN 映射在 STCI 上, 只有在 STIS 选择内部触发输入时才工作。

**14.6.4.9 TMR1通道控制寄存器 (TMR1\_CCTRL)**

域	简称	复位值	类型	功能
位 15: 14	保留	0x0	resd	保持默认值。
位 13	C4P	0x0	rw	通道 4 极性 (Channel 4 polarity) 见 C1P 的描述。
位 12	C4EN	0x0	rw	通道 4 使能 (Channel 4 enable) 见 C1EN 的描述。
位 11	C3CP	0x0	rw	通道 3 互补极性 (Channel 3 complementary polarity) 见 C1P 的描述。
位 10	C3CEN	0x0	rw	通道 3 互补使能 (Channel 3 complementary enable) 见 C1EN 的描述。
位 9	C3P	0x0	rw	通道 3 极性 (Channel 3 polarity) 见 C1P 的描述。
位 8	C3EN	0x0	rw	通道 3 使能 (Channel 3 enable) 见 C1EN 的描述。
位 7	C2CP	0x0	rw	通道 2 互补极性 (Channel 2 complementary polarity) 见 C1P 的描述。
位 6	C2CEN	0x0	rw	通道 2 互补使能 (Channel 2 complementary enable) 见 C1EN 的描述。
位 5	C2P	0x0	rw	通道 2 极性 (Channel 2 polarity) 见 C1P 的描述。
位 4	C2EN	0x0	rw	通道 2 使能 (Channel 2 enable) 见 C1EN 的描述。
位 3	C1CP	0x0	rw	通道 1 互补极性 (Channel 1 complementary polarity) 0: C1COUT 的有效电平为高 1: C1COUT 的有效电平为低
位 2	C1CEN	0x0	rw	通道 1 互补使能 (Channel 1 complementary enable) 0: 禁止输出; 1: 使能输出。
位 1	C1P	0x0	rw	通道 1 极性 (Channel 1 polarity) 通道 1 配置为输出:

				0: C1OUT 的有效电平为高 1: C1OUT 的有效电平为低 通道 1 配置为输入: C1CP/C1P 位共同定义输入信号有效沿。 00: C1IN 的有效边沿为上升沿; 作为外部触发使用时, C1IN 不反相。 01: C1IN 的有效边沿为下降沿; 作为外部触发使用时, C1IN 反相。 10: 保留 11: C1IN 的有效边沿为上升沿和下降沿; 作为外部触发使用时, C1IN 不反相。
位 0	C1EN	0x0	rw	通道 1 使能 (Channel 1 enable) 0: 禁止输入或输出; 1: 使能输入或输出。

表 14-17 带刹车功能的互补输出通道CxOUT和CxCOUT的控制位

控制位			输出状态 (1)			
OEN 位	FCSODIS 位	FCSOEN 位	CxEN 位	CxCEN 位	CxOUT 输出状态	CxCOUT 输出状态
1	X		0	0	0	输出禁止 (与定时器断开) CxOUT=0, Cx_EN=0
			0	0	1	输出禁止 (与定时器断开) CxOUT=0, Cx_EN=0
			0	1	0	CxORAW+极性, CxOUT= CxORAW xor CxP, Cx_EN=1
			0	1	1	CxORAW+极性+死区, Cx_EN=1
			1	0	0	输出禁止 (与定时器断开) CxOUT=CxP, Cx_EN=0
			1	0	1	关闭状态 (输出使能且为无效电平) CxOUT=CxP, Cx_EN=1
			1	1	0	CxORAW + 极性, CxOUT= CxORAW xor CxP, Cx_EN=1
			1	1	1	CxORAW+极性+死区, Cx_EN=1
0	X		0	0	0	输出禁止 (对应 IO 与定时器断开, IO 浮空) 异步地: CxOUT=CxP , Cx_EN=0 , CxCOUT=CxCP , CxCEN=0;
			0	1	0	若时钟存在: 经过一个死区时间后 CxOUT=CxIOS, CxCOUT=CxCIOS, 假设 CxIOS 与 CxCIOS 并不都对 应 CxOUT 和 CxCOUT 的有效电平。
			0	0	1	
			0	1	1	
			1	0	0	CxEN=CxCEN=0 时: 输出禁止 (对应 IO 与定时器断 开, IO 浮空); 其他情况下: 关闭状态 (对应通道输出无效电平)
			1	1	0	

	1		1	0	异步地: CxOUT =CxP, Cx_EN=1, CxCOU=CxCP , CxCEN=1; 若时钟存在: 经过一个死区 时间后 CxOUT =C1IOS, CxCOU=CxCIO, 假设 CxIOS 与 CxCOU 并不都对应 CxOUT 和 CxCOU 的有效电平。
	1		1	1	

注意: 如果一个通道的 2 个输出都没有使用 ( $CxEN = CxCEN = 0$ ) , 那么  $CxIOS$ ,  $CxCIO$ ,  $CxP$  和  $CxCP$  都必须清零。

注意: 管脚连接到互补的  $CxOUT$  和  $CxCOU$  通道的外部 I/O 管脚的状态, 取决于  $CxOUT$ 、 $CxCOU$  通道状态和 GPIO 以及 IOMUX 寄存器。

#### 14.6.4.10 TMR1计数值 (TMR1\_CVAL)

域	简称	复位值	类型	功能
位 15: 0	CVAL	0x0000	rw	计数值 (Counter value)

#### 14.6.4.11 TMR1预分频器 (TMR1\_DIV)

域	简称	复位值	类型	功能
位 15: 0	DIV	0x0000	rw	分频系数 (Divider value) 计数器时钟频率 $f_{CK\_CNT} = f_{TMR\_CLK} / (DIV[15: 0]+1)$ 溢出事件发生时该寄存器值被传送到实际的预分频寄存器中。

#### 14.6.4.12 TMR1周期寄存器 (TMR1\_PR)

域	简称	复位值	类型	功能
位 15: 0	PR	0x0000	rw	周期值 (Period value) 定时器计数的周期值。当周期值为 0 时, 定时器不工作。

#### 14.6.4.13 TMR1重复周期寄存器 (TMR1\_RPR)

域	简称	复位值	类型	功能
位 15: 8	保留	0x00	resd	保持默认值。
位 7: 0	RPR	0x00	rw	重复周期的次数 (Repetition of period value) 这些位用于减慢溢出事件发生的速率, 当重复周期的次数减为 0 时才会发生溢出事件。

#### 14.6.4.14 TMR1通道1数据寄存器 (TMR1\_C1DT)

域	简称	复位值	类型	功能
位 15: 0	C1DT	0x0000	rw	通道 1 数据寄存器值 (Channel 1 data register) 若通道 1 配置为输入: C1DT 是前一次通道 1 输入事件 (C1IN) 所保存的 CVAL。 若通道 1 配置为输出: C1DT 是将要和 CVAL 进行比较的值, 写入的值是否会立即生效取决于输出缓存使能位 (C1OBEN), 并根据设置在 C1OUT 上产生相应的输出。

#### 14.6.4.15 TMR1通道2数据寄存器 (TMR1\_C2DT)

域	简称	复位值	类型	功能
位 15: 0	C2DT	0x0000	rw	通道 2 数据寄存器值 (Channel 2 data register) 若通道 2 配置为输入: C2DT 是前一次通道 2 输入事件 (C2IN) 所保存的 CVAL。 若通道 2 配置为输出: C2DT 是将要和 CVAL 进行比较的值, 写入的值是否会立即生效取决于输出缓存使能位 (C2OBEN), 并根据设置在 C2OUT 上产生相应的输出。

#### 14.6.4.16 TMR1通道3数据寄存器 (TMR1\_C3DT)

域	简称	复位值	类型	功能
位 15: 0	C3DT	0x0000	rw	通道 3 数据寄存器值 (Channel 3 data register) 若通道 3 配置为输入: C3DT 是前一次通道 3 输入事件 (C3IN) 所保存的 CVAL。 若通道 3 配置为输出: C3DT 是将要和 CVAL 进行比较的值, 写入的值是否会立即生效取决于输出缓存使能位 (C3OBEN), 并根据设置在 C3OUT 上产生相应的输出。

#### 14.6.4.17 TMR1通道4数据寄存器 (TMR1\_C4DT)

域	简称	复位值	类型	功能
位 15: 0	C4DT	0x0000	rw	通道 4 数据寄存器值 (Channel 4 data register) 若通道 4 配置为输入: C4DT 是前一次通道 4 输入事件 (C4IN) 所保存的 CVAL。 若通道 4 配置为输出: C4DT 是将要和 CVAL 进行比较的值, 写入的值是否会立即生效取决于输出缓存使能位 (C4OBEN), 并根据设置在 C4OUT 上产生相应的输出。

#### 14.6.4.18 TMR1刹车寄存器 (TMR1\_BRK)

域	简称	复位值	类型	功能
位 15	OEN	0x0	rw	输出使能 (Output enable) 对配置为输出的通道, 该位用于使能 CxOUT 和 CxCOUT 的输出。 0: 关闭; 1: 开启。
位 14	AOEN	0x0	rw	输出自动使能(Automatic output enable) 用于溢出事件时将 OEN 自动置'1' 0: 关闭; 1: 开启
位 13	BRKV	0x0	rw	刹车输入信号的有效性 (Brake input validity) 用于选择刹车输入信号的输入有效电平: 0: 低电平; 1: 高电平。
位 12	BRKEN	0x0	rw	刹车功能使能 (Brake enable) 用于开启刹车功能。 0: 关闭; 1: 开启。
位 11	FCSOEN	0x0	rw	总输出开时的冻结状态 (Frozen channel status when holistic output enable) 该位用于配置具有互补输出的通道, 在定时器不工作且 OEN=1 时的通道状态。 0: 关闭 CxOUT/CxCOUT 输出; 1: 开启 CxOUT/CxCOUT 输出, 输出为无效电平。
位 10	FCSODIS	0x0	rw	总输出关时的冻结状态 (Frozen channel status when holistic output disable) 该位用于配置具有互补输出的通道, 在定时器不工作且 OEN=0 时的通道状态。 0 : 关闭 CxOUT/CxCOUT 输出; 1 : 开启 CxOUT/CxCOUT 输出, 输出为空闲电平。
位 9: 8	WPC	0x0	rw	写保护配置 (Write protected configuration) 该位用于配置写保护。 00: 写保护关闭; 01: 3 级写保护, 以下位受写保护: TMR1_BRK: DTC、BRKEN、BRKV 和 AOEN TMR1_CTRL2: CxIOS 和 CxCIOS 10: 2 级写保护, 除 3 级写保护的内容外, 以下位也受写保护:

TMR1\_CCTRL: CxP 和 CxCP  
 TMR1\_BRK: FCSODIS 和 FCSOEN  
 11: 1 级写保护, 除 2 级写保护的内容外, 以下位也受写保护:  
 TMR1\_CMx: C2OCTRL 和 C2OBEN  
 注: WPC>0 时将无法再次被修改, 直到系统复位。

死区配置 (Dead-time configuration)  
 这些位用于配置死区时间。取 DTC[7: 0]的高 3 位为功能选择位:

位 7: 0	DTC	0x00	rw	0xx: DT = DTC [7: 0] * TDTS; 10x: DT = (64+ DTC [5: 0]) * TDTS * 2; 110: DT = (32+ DTC [4: 0]) * TDTS * 8; 111: DT = (32+ DTC [4: 0]) * TDTS * 16;
--------	-----	------	----	---

注意: 根据锁定设置, AOEN、BRKV、BRKEN、FCSODIS、FCSOEN 和 DTC[7: 0]位均可被写保护, 有必要在第一次写入 TMR1\_BRK 寄存器时对它们进行配置。

#### 14.6.4.19 TMR1 DMA控制寄存器 (TMR1\_DMACTRL)

域	简称	复位值	类型	功能
位 15: 13	保留	0x0	resd	保持默认值。
位 12: 8	DTB	0x00	rw	DMA 传输字节 (DMA transfer bytes) 这些位定义了传输的字节个数: 00000: 1 个字节      00001: 2 个字节 00010: 3 个字节      00011: 4 个字节 ..... 10000: 17 个字节      10001: 18 个字节
位 7: 5	保留	0x0	resd	保持默认值。
位 4: 0	ADDR	0x00	rw	DMA 传输地址偏移 (DMA transfer address offset) ADDR 定义了从 TMR1_CTRL1 所在地址开始的偏移量: 00000: TMR1_CTRL1, 00001: TMR1_CTRL2, 00010: TMR1_STCTRL, .....

#### 14.6.4.20 TMR1 DMA数据寄存器 (TMR1\_DMADT)

域	简称	复位值	类型	功能
位 15: 0	DMADT	0x0000	rw	DMA 传输的数据寄存器 (DMA data register) 通过对 DMADT 寄存器的读写能够实现对任意 TMR 寄存器的操作, 其操作的寄存器地址范围是: TMR1 外设地址 + ADDR*4 至 TMR1 外设地址 + ADDR*4 + DTB*4。

# 15 窗口看门狗 (WWDT)

## 15.1 WWDT简介

当程序正常运行时，需在一个有限的时间窗口内重载窗口看门狗递减计数器，用来避免看门狗电路产生系统复位，以此来监测系统是否正常运行。

窗口看门狗时钟由 APB1\_CLK 分频而来，由于 APB1\_CLK 的精确性，窗口看门狗可对有限的时间窗口精确控制。

## 15.2 WWDT主要特性

- 7 位递减计数器
- 启动看门狗后，当递减计数器的值小于 0x40 或是在窗口外被重新装载产系统生复位。
- 可以通过重载计数器中断重装载计数器。

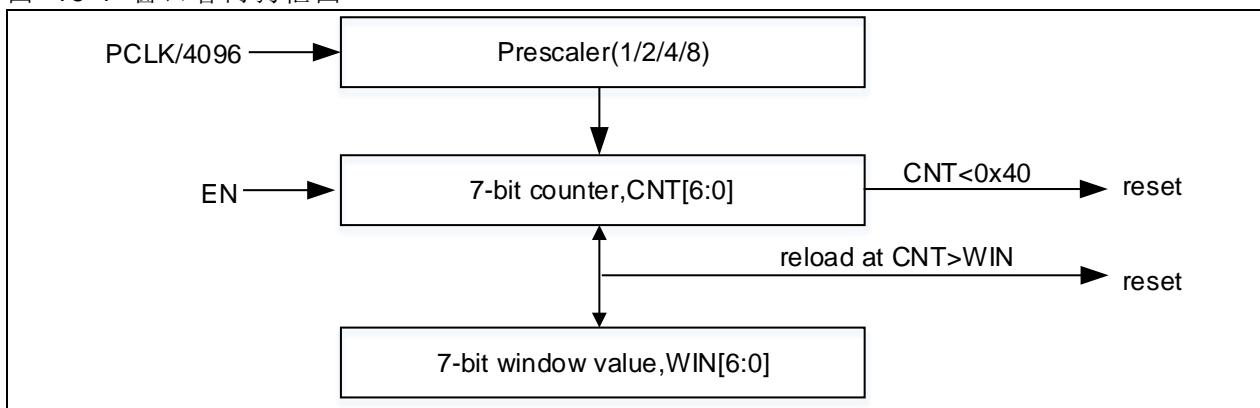
## 15.3 WWDT功能描述

启动窗口看门狗后，窗口看门狗可在以下两种情况下产生系统复位：

第一种，7 位递减计数器值由 0x40 变为 0x3F。

第二种，7 位递减计数器值大于 7 位窗口值时，重载计数器值。

图 15-1 窗口看门狗框图



为避免重载计数器值时产生复位，应在计数器值小于窗口值大于 0x40 时重载计数器值。

WWDT 计数器时钟由 APB1\_CLK 分频得到，分频系数可通过配置配置寄存器 (WWDT\_CFG) DIV[1:0] 改变。计数器值决定了 WWDT 复位前的最大计数周期数，结合 WIN[6:0] 可灵活的调整重载窗口。WWDT 提供了重载计数器中断功能，开启后，WWDT 将在计数值达到 0x40h 时将 RLDF 标志位置 1，同时产生重载计数器中断，可在中断服务程序中重载计数器值，以避免发生系统复位。需要注意的是，若在 CNT[6] 为 0 时，将 WWDTEN 置 1 会产生一个系统复位，因此当写入控制寄存器 (WWDT\_CTRL) 时，应始终保持 CNT[6] 为 1，避免使能窗口看门狗后立即产生一个系统复位。

窗口看门狗超时时间  $T_{WWDT}$  可由一下公式计算，其中  $T_{PCLK1}$  为 APB1 时钟周期，单位为 ms：

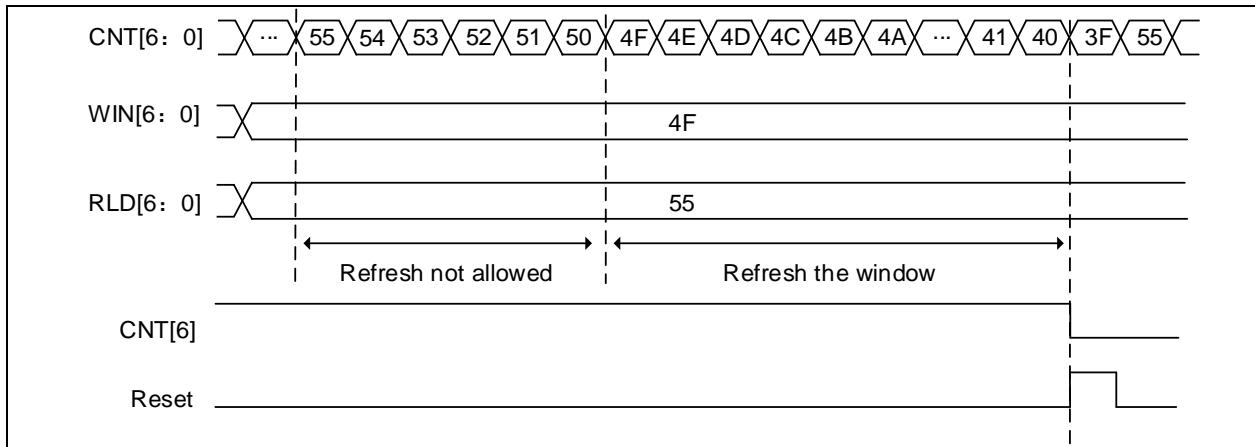
$$T_{WWDT} = T_{PCLK1} \times 4096 \times 2^{DIV[1:0]} \times (CNT[5:0] + 1); \quad (\text{ms})$$

下表给出了当 PCLK1 频率为 72MHz 时，最大和最小看门狗超时时间。

表 15-1 PCLK1频率为72MHz时，最大和最小看门狗超时时间

时钟预分频值	最小超时时间	最大超时时间
0	56.5 μs	3.64ms
1	113.5 μs	7.28ms
2	227.5 μs	14.56ms
3	455 μs	29.12ms

图 15-2 窗口看门狗时序图



## 15.4 调试模式

微控制器处于调试模式时，意味着 Cortex®-M4 核心停止。将 DEBUG 模块中 WWDT\_PAUSE 位置 1 可将 WWDT 计数器计数暂停。

## 15.5 WWDT 寄存器

必须以字（32 位）的方式操作这些外设寄存器。

表 15-2 WWDT 寄存器映像和复位值

寄存器简称	基址偏移量	复位值
WWDT_CTRL	0x00	0x7F
WWDT_CFG	0x04	0x7F
WWDT_STS	0x08	0x00

### 15.5.1 控制寄存器 (WWDT\_CTRL)

域	简称	复位值	类型	功能
位 31: 8	保留	0x000000	resd	保持默认值。
位 7	WWDTEN	0x0	rw1s	窗口看门狗使能 (Window watchdog enable) 0: 关闭; 1: 开启。 该位由软件置起，只能在复位后自动清零。
位 6: 0	CNT	0x7F	rw	递减计数器 (Decrement counter) 当计数器递减到 0x3F 时产生复位。

### 15.5.2 配置寄存器 (WWDT\_CFG)

域	简称	复位值	类型	功能
位 31: 10	保留	0x000000	resd	保持默认值。
位 9	RLDIEN	0x0	rw	重载计数器中断 (Reload counter interrupt) 0: 关闭; 1: 开启。
位 8: 7	DIV	0x0	rw	时钟预分频值 (Clock division value) 00: PCLK1 除以 4096; 01: PCLK1 除以 8192; 10: PCLK1 除以 16384; 11: PCLK1 除以 32768。
位 6: 0	WIN	0x7F	rw	窗口值 (Window value) 当计数器值大于窗口值时，此时重载计数器会产生复位，重载计数器区间为 0x40~WIN[6: 0]

### 15.5.3 状态寄存器 (WWDT\_STS)

域	简称	复位值	类型	功能
位 31: 1	保留	0x0000 0000	resd	保持默认值。
位 0	RLDF	0x0	rw0c	重载计数器中断标志 (Reload counter interrupt flag) 当递减计数器为 0x40 时，该标志会置位。 该位被硬件置起，由软件将其清零。'

# 16 看门狗 (WDT)

## 16.1 WDT简介

看门狗由专用低速时钟 (LICK) 驱动，由于 LICK 时钟精度较低，因此看门狗适用于低时间精度、能够独立于主程序之外的应用

## 16.2 WDT主要特性

- 12 位递减计数器
- 计数器由 LICK 时钟驱动（可在深睡眠和待机模式下工作）
- 看门狗使能后，将在计数器计数至 0 时产生 WDT 系统复位

## 16.3 WDT功能描述

### WDT 启动方式:

WDT 的启动方式有两种，分别为软件启动和硬件启动。软件启动通过向命令寄存器 (WDT\_CMD) 写入 0xCCCC 实现；硬件启动则需通过配置用户系统数据区来实现，使能硬件看门狗后，看门狗将在上电复位后自动开始运行。

### WDT 复位条件:

当 WDT 计数器值递减至 0 时将产生 WDT 系统复位，因此需定时向命令寄存器 (WDT\_CMD) 写入 0xAAAA 重载计数器值。

### WDT 写保护:

预分频寄存器 (WDT\_DIV)、重装载寄存器 (WDT\_RLD) 受写保护，向命令寄存器 (WDT\_CMD) 写入 0x5555 可解锁寄存器写保护，之后可对其进行配置。这两个寄存器的更新状态分别由 WDT\_STS 寄存器中 DIVF、RLDF 指示。向命令寄存器 (WDT\_CMD) 写入其它值将重新启动预分频寄存器 (WDT\_DIV)、重装载寄存器 (WDT\_RLD) 写保护。向命令寄存器 (WDT\_CMD) 写入 0xAAAA 也会启动寄存器写保护。

### WDT 时钟:

WDT 计数器由 LICK 时钟驱动，LICK 是内部 RC 时钟，范围为 30kHz~60kHz 之间，所以超时时间也是在一定区间内，使用时应注意在超时时间配置上应该留有余量，如果需要获得较为精确的看门狗超时时间，可对 LICK 进行校准，有关 LICK 校准的问题，详见 [4.1.1 节](#)。

图 16-1 看门狗框图

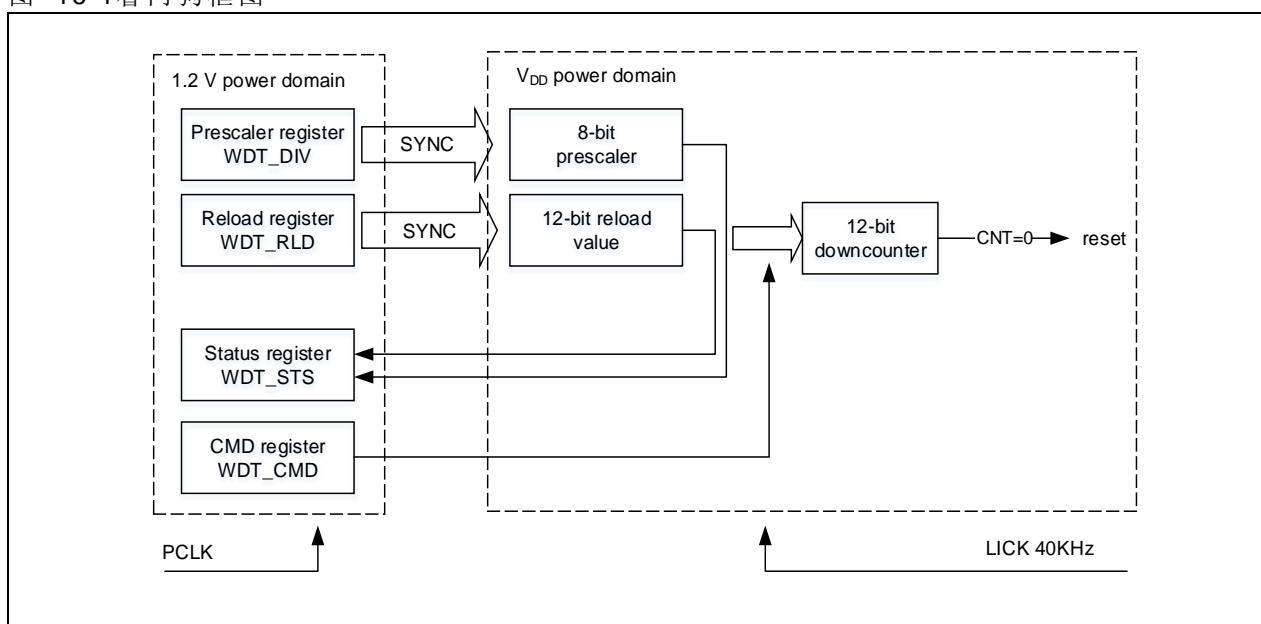


表 16-1 看门狗超时时间 (LICK=40kHz)

预分频系数	DIV[2: 0]位	最短时间 (ms) RLD[11: 0] = 0x000	最长时间 (ms) RLD[11: 0] = 0xFFFF
/4	0	0.1	409.6
/8	1	0.2	819.2
/16	2	0.4	1638.4
/32	3	0.8	3276.8
/64	4	1.6	6553.6
/128	5	3.2	13107.2
/256	(6 或 7)	6.4	26214.4

## 16.4 调试模式

微控制器处于调试模式时，意味着 Cortex®-M4 核心停止。此时将 DEBUG 模块中 WDT\_PAUSE 位置 1 会暂停 WDT 计数器计数。

## 16.5 WDT 寄存器

必须以字（32 位）的方式操作这些外设寄存器。

表 16-2 WDT 寄存器映像和复位值

寄存器简称	基址偏移量	复位值
WDT_CMD	0x00	0x0000 0000
WDT_DIV	0x04	0x0000 0000
WDT_RLD	0x08	0x0000 0FFF
WDT_STS	0x0C	0x0000 0000

### 16.5.1 命令寄存器 (WDT\_CMD)

（在待机模式复位）

域	简称	复位值	类型	功能
位 31: 16	保留	0x0000	resd	保持默认值。 命令寄存器 (Command register) 0xAEEE: 重载计数器; 0xAAAA: 解锁 WDT_DIV 和 WDT_RLD 写保护; 0xCCCC: 启动看门狗, 如果使能了硬件看门狗, 则不需要执行此操作。
位 15: 0	CMD	0x0000	wo	

### 16.5.2 预分频寄存器 (WDT\_DIV)

（在待机模式复位）

域	简称	复位值	类型	功能
位 31: 3	保留	0x0000 0000	resd	保持默认值。 递减计数器时钟预分频值 (Clock division value) 000: LICK 除以 4; 001: LICK 除以 8; 010: LICK 除以 16; 011: LICK 除以 32; 100: LICK 除以 64; 101: LICK 除以 128; 110: LICK 除以 256; 111: LICK 除以 256。 只有解锁写保护后才能写此寄存器, 只有当 DIVF 为 0 时, 才能读取此寄存器。
位 2: 0	DIV	0x0	rw	

### 16.5.3 重装载寄存器 (WDT\_RLD)

(待机模式时复位)

域	简称	复位值	类型	功能
位 31: 12	保留	0x00000	resd	保持默认值。
位 11: 0	RLD	0xFFFF	rw	重载值 (Reload value) 只有解锁写保护后才能写此寄存器，只有当 RLDF 为 0 时，才能读取此寄存器。

### 16.5.4 状态寄存器 (WDT\_STS)

(在待机模式复位)

域	简称	复位值	类型	功能
位 31: 2	保留	0x0000 0000	resd	保持默认值。
位 1	RLDF	0x0	ro	重载值更新完成标志 (Reload value update complete flag) 0: 更新完成; 1: 正在更新。 只有当 RLDF 为 0 时才能写重装载寄存器 (WDT_RLD)。
位 0	DIVF	0x0	ro	分频值更新完成标志 (Division value update complete flag) 0: 更新完成; 1: 正在更新。 只有当 DIVF 为 0 时才能写预分频寄存器 (WDT_DIV)。

# 17 实时时钟 (ERTC)

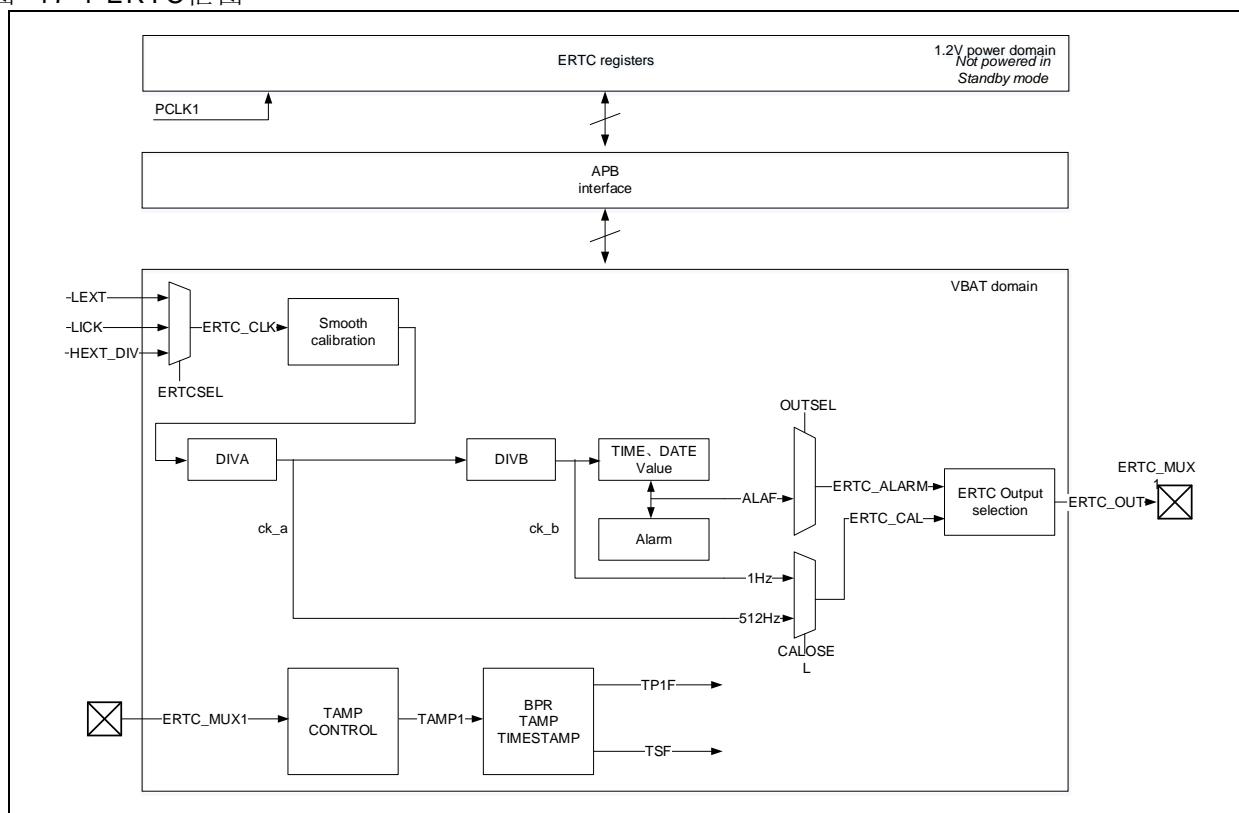
## 17.1 ERTC简介

实时时钟用于配置日历时钟，修改 ERTC 中日历寄存器值可以修改系统的当前时间和日期。ERTC 计数逻辑位于电池供电域，只要电池供电域有电（仅支持由 VDD 域供电），ERTC 便会一直运行，不受系统复位影响。

## 17.2 ERTC主要特性

- 功能强大的实时日历，自动处理月份天数 28（平年 2 月）、29（闰年 2 月）、30（小月）、31（大月），其中当年份寄存器是 4 的倍数时为闰年，支持一组闹钟
- 参考时钟检测
- 一组可配置入侵检测，支持时间戳功能
- 支持精密校准
- 5 个电池供电寄存器
- 4 组中断：闹钟 A、入侵检测、时间戳
- 复用功能输出，校准时钟输出、闹钟事件
- 复用功能输入，参考时钟输入、一路入侵检测、时间戳

图 17-1 ERTC 框图



## 17.3 ERTC功能说明

### 17.3.1 ERTC时钟

ERTC\_CLK 可从 LEXT、LICK、32 分频后的 HEXT 中选择。

ERTC 内置分频器 A 和分频器 B，分别由 DIVA[6: 0]、DIVB[14: 0]配置，推荐 DIVA 配置为较高的值，以最大程度降低功耗。ERTC\_CLK 依次经由分频器 A、分频器 B 处理，得到 ck\_a、ck\_b 时钟，ck\_a 用于更新亚秒，ck\_b 用于更新日历。ck\_a、ck\_b 时钟频率可由下式计算：

$$f_{ck\_a} = \frac{f_{ERTC\_CLK}}{DIVA + 1}$$

$$f_{ck\_b} = \frac{f_{ERTC\_CLK}}{(DIVB + 1) \times (DIVA + 1)}$$

当配置 DIVA=127, DIVB=255, 且 ERTC\_CLK 选用 32.768kHz 的 LEXT 时, 可得到 1Hz 的 ck\_b, 用于更新日历。

### 17.3.2 ERTC 初始化

**寄存器解锁:**

上电复位后所有 ERTC 寄存器处于写保护状态, 需要先解除写保护, 才能写配置 ERTC 寄存器(除 ERTC 初始化和状态寄存器(ERTC\_STS[14:8])、ERTC 入侵配置寄存器(ERTC\_TAMP) 和 ERTC 电池供电数据寄存器(ERTC\_BPRx) 外, 其它寄存器位均受写保护, 且写保护不受系统复位影响)。

**解锁步骤:**

- 1、使能电源接口时钟: APB1 外设时钟使能寄存器(CRM\_APB1EN)的 PWCEN=1
- 2、解锁电池供电域写保护: 电源控制寄存器(PWC\_CTRL)的 BPWEN=1
- 3、依次向 ERTC 写保护寄存器(ERTC\_WP)写入 0xCA, 0x53, 若向 ERTC 写保护寄存器(ERTC\_WP)写入错误的值, 将重新激活写保护。

下表列出了需要解除写保护和进入初始化模式才可以配置的 ERTC 寄存器:

表 17-1 ERTC 寄存器配置表

寄存器简称	是否受写保护	是否需要进入初始化模式	其它
ERTC_TIME	是	是	-
ERTC_DATE	是	是	-
ERTC_CTRL	是	位 6、4 需要	-
ERTC_STS	除[13:8]位外	-	-
ERTC_DIV	是	是	-
ERTC_ALA	是	否	ALAWF 为 1 时可配置
ERTC_WP	-	-	-
ERTC_SBS	-	-	-
ERTC_TADJ	是	否	TADJF 为 0 时可配置
ERTC_TSTM	-	-	-
ERTC_TS DT	-	-	-
ERTC_TSSBS	-	-	-
ERTC_SCAL	是	否	CALUPDF 为 0 时可配置
ERTC_TAMP	否	否	-
ERTC_ALASBS	是	否	ALAWF 为 1 时可配置
ERTC_BPRx	否	否	-

**时钟、日历初始化:**

寄存器解锁后, 时钟和日历的初始化配置可按以下步骤进行:

1. 将 IMEN 位置 1 进入初始化模式。
2. 等待初始化标志位 INITF 置 1。
3. 依次配置 DIVB、DIVA。
4. 配置时钟和日历值。
5. 将 IMEN 位清 0 退出初始化模式, 等待 UPDF 置 1, 表明日历值同步完成, 日历开始计数。

为了方便时间微调, ERTC 还提供了夏令时和时间调整功能。

**夏令时功能:** 用于增加 (ADD1H=1) 或减小 (DEC1H=1) 1 小时, 而无需重新进行初始化配置。

**时间调整功能:** 用于精确的调整当前时钟。若只配置 DECSBS[14:0]值, 该值将会加到分频器 B 计数器值

中，时钟因此产生延迟；若只将 ADD1S 位置 1，当前时钟将增加 1 秒；若同时配置 DECSBS[14:0]，ADD1S 位，时钟将增加零点几秒。

延迟时间 (ADD1S=0): 延迟时间=DECSBS/(DIVB+1);

提前时间 (ADD1S=1): 提前时间=1-(DECSBS/(DIVB+1))。

注：设置时间调整寄存器前，必须先确认 SBS[15]=0，以免亚秒发生上溢。

#### 日历读取：

ERTC 提供两种日历访问方式，分别为同步读取 (DREN=0) 和异步读取 (DREN=1)。

同步读取时，ERTC 通过 PCLK1 访问同步的影子寄存器来获取时钟和日历值。影子寄存器值由位于电池供电域的 ERTC 日历值同步而来，同步周期为两个 ERTC\_CLK，同步完成后 UPDF 将置 1。影子寄存器由系统复位来复位。为保证读取的 ERTC 亚秒寄存器 (ERTC\_SBS)、ERTC 时间寄存器 (ERTC\_TIME)、ERTC 日期寄存器 (ERTC\_DATE) 值来自同一时刻，读取低阶寄存器时会将高阶寄存器值锁定，直到读取 ERTC 日期寄存器 (ERTC\_DATE)。例如读取 ERTC 亚秒寄存器 (ERTC\_SBS)，会将 ERTC 时间寄存器 (ERTC\_TIME)、ERTC 日期寄存器 (ERTC\_DATE) 值锁定。

异步读取时，ERTC 通过 PCLK1 直接读取位于电池供电域的 ERTC 时钟和日历值，这样避免了由于同步时间带来的误差。异步读取时，UPDF 标志将由硬件清 0。为保证异步读取时钟和日历值的准确性，软件必须两次读取时钟和日历值，并比较两次结果是否一致，如果不一致应该再读，直到两次结果一致，另外，也可以只比较两次结果的最低位来判定。

注：在 STANDBY 和 DEEPSLEEP 模式下，当前日历值不会复制到影子寄存器中，当从这两种模式唤醒时，必须先设置 UPDF=0，然后等待 UPDF=1，以保证读取的日历值是最新的；在同步读取时，需保证 PCLK1 频率至少为 ERTC\_CLK 频率 7 倍；异步读取时，需要额外一个 APB 周期才能完成读取日历寄存器的指令。

#### 闹钟初始化：

ERTC 包含闹钟 A，并提供了闹钟 A 中断。

闹钟值由 ERTC 闹钟 A 亚秒寄存器 (ERTC\_ALASBS)、ERTC 闹钟 A 寄存器 (ERTC\_ALA) 设定，开启闹钟后，当设定的闹钟值匹配日历值时将触发闹钟事件。通过 MASKx 位，可选择性的屏蔽日历字段，被屏蔽的字段不参与闹钟匹配。

闹钟 A 的配置可按以下步骤进行：

1. 关闭闹钟 A (设置 ALAEN=0)。
2. 等待闹钟 A 寄存器允许写 (等待 ALAWF 位置 1)。
3. 配置闹钟 A 寄存器 (ERTC\_ALA, ERTC\_ALASBS)。
4. 使能闹钟 A (设置 ALAEN=1)。

注：当 ERTC 闹钟 A 寄存器 (ERTC\_ALA) 中 MASK1 为 0 时，DIVB 至少为 3 才能使闹钟正常工作。

### 17.3.3 ERTC 校准

#### 精密数字校准：

精密校准可以均匀且精确的校准 ERTC\_CLK。开启精密校准校准功能后，将均匀增加或减少 ERTC\_CLK 来达到校准的目的。

当 ERTC\_CLK 为 32.768kHz 时，精密校准周期约为  $2^{20}$  个 ERTC\_CLK(32 秒)。DEC[8:0]值指定了  $2^{20}$  个 ERTC\_CLK 中忽略的脉冲数，最多可忽略 511 个脉冲；将 ADD 置 1，可在  $2^{20}$  个 ERTC\_CLK 中插入 512 个脉冲。两者搭配使用，可在  $2^{20}$  个 ERTC\_CLK 周期进行 -511~+512 的调整。

有效校准频率  $F_{SCAL}$ :

$$F_{SCAL} = F_{ERTC\_CLK} \times [1 + \frac{ADD \times 512 - DEC}{2^{20} + DEC - ADD \times 512}]$$

当分频器 A 值小于 3 时，会按照 ADD 等于 0 校准。此时应降低分频器 B 值来实现每秒增加 8 个 ERTC\_CLK，也就是 32 秒增加 256 个 ERTC\_CLK 搭配 DEC[8:0]位，可在  $2^{20}$  个 ERTC\_CLK 周期进行 -255~+256 的调整。

此时有效校准频率  $F_{SCAL}$ :

$$F_{SCAL} = F_{ERTC\_CLK} \times [1 + \frac{256 - DEC}{2^{20} + DEC - 256}]$$

精密数字校准的校准周期还可选择 8 秒或 16 秒(由 CAL8 和 CAL16 配置)，8 秒校准周期的优先级更高，

同时使能 8 秒和 16 秒校准周期，将优先选择 8 秒校准周期。

ERTC 提供了 CALUPDF 标志用来指示校准值的状态，当配置 ERTC 精密校准寄存器 (ERTC\_SCAL) 寄存器时，CALUPDF 标志位将置 1，指示校准值正在更新；当校准值被成功应用后，标志位自动清 0，指示校准值更新完成。

### 17.3.4 时间戳

时间戳功能用于在发生时间戳事件时（入侵引脚检测到有效边沿），将当前的日历值保存到时间戳寄存器中。

当发生时间戳时，TSF 位置 1，此时若再次发生时间戳事件，TSOF 标志位将置 1，但时间戳寄存器并不会更新，可以通过 TSIEN 位设置是否使能时间戳中断。

时间戳用法有两种

1. 单独的时间戳功能，此时入侵检测引脚用来检测时间戳，使用步骤：

- 选择上升沿还是下降沿触发（设置 TSEDG）
- 使能时间戳（设置 TSEN=1）

2. 发生入侵事件时保存时间戳，使用步骤：

- 配置入侵检测相关寄存器
- 使能发生入侵事件时保存时间戳（设置 TPTSEN=1）

注：发生时间戳事件后，TSF 在两个 ck\_a 周期后置 1，建议在 TSF 已置 1 的情况下轮询 TSOF 位

### 17.3.5 入侵检测

ERTC 提供了一组入侵检测 TAMP1，可配置为滤波后的电平检测或边沿检测。TAMP1 映射到入侵引脚 ERTC\_MUX1。

当检测到有效的入侵事件后，TP1F 位将置 1，若已使能了入侵检测中断，将产生对应的中断；若 TPTSEN 位已置 1，将同时产生时间戳事件。为保证位于电池供电域中的电池供电寄存器数据安全，入侵事件发生时将复位电池供电寄存器。

边沿入侵检测配置步骤：

1. 选择入侵检测方式为边沿检测 (TPFLT=00)，并选择有效沿 (TP1EDG)。
2. 根据需要配置是否在入侵事件时激活时间戳 (TPTSEN 置 1)。
3. 根据需要开启入侵中断使能 (TPIEN 置 1)。
4. 将 TAMP1 使能 (TP1EN 置 1)。

电平入侵检测配置步骤：

1. 选择入侵检测方式为电平检测，并选择有效电平采样次数 (TPFLT≠00)。
2. 选择入侵有效电平 (TP1EDG)。
3. 选择入侵检测采样频率 (TPFREQ)。
4. 根据需要开启入侵检测上拉 (TPPU 置 1)，若开启，还需配置上拉电阻预充电时间 (TPPR)。
5. 根据需要配置是否在入侵事件时激活时间戳 (TPTSEN 置 1)。
6. 根据需要开启入侵中断使能 (TPIEN 置 1)。
7. 将 TAMP1 使能 (TP1EN 置 1)。

当配置为边沿检测时，有以下两点要注意：

1. 在使能入侵检测前，若入侵检测已被配置为上升沿有效，且入侵检测引脚已为高电平，在使能入侵检测后会立刻产生一个入侵检测事件。
2. 在使能入侵检测前，若入侵检测已被配置为下降沿有效，且入侵检测引脚已为低电平，在使能入侵检测后会立刻产生一个入侵检测事件。

### 17.3.6 复用功能输出

ERTC 提供了一组复用功能输出，可以输出以下事件：

1. 校准后的时钟 (OUTSEL=0, CALOEN=1)
  - 输出 512Hz (CALOSEL=0)
  - 输出 1Hz (CALOSEL=1)
2. 闹钟 A (OUTSEL=1)

当输出闹钟事件时 (OUTSEL≠0)，可以通过 OUTTYPE 选择输出类型为开漏或是推挽，可以通过 OUTP

配置输出极性。

### 17.3.7 ERTC唤醒

ERTC 可由闹钟、时间戳、入侵事件进行唤醒，使能 ERTC 中断可按以下操作配置：

1. 将 ERTC 对应中断的 EXINT 线配置为中断模式并使能，有效沿选择上升沿。
2. 使能 ERTC 中断对应的 NVIC 通道。
3. 使能对应的 ERTC 中断。

下表说明了 ERTC 时钟源、事件以及中断对唤醒低功耗模式的影响：

表 17-2 ERTC 唤醒低功耗模式

时钟源	事件	唤醒 SLEEP	唤醒 DEEPSLEEP	唤醒 STANDBY
HEXT	闹钟 A	√	×	×
	时间戳	√	×	×
	入侵事件	√	×	×
LICK	闹钟 A	√	√	√
	时间戳	√	√	√
	入侵事件	√	√	√
LEXT	闹钟 A	√	√	√
	时间戳	√	√	√
	入侵事件	√	√	√

表 17-3 中断控制位

中断事件	事件标志	中断使能位	EXINT 线
闹钟 A	ALAF	ALAIEN	17
时间戳	TSF	TSIEN	19
入侵事件	TP1F	TP1EN	19

## 17.4 ERTC 寄存器

必须以字（32 位）的方式操作这些外设寄存器。写 ERTC 寄存器占用 APB1 总线约 4 个 ERTC\_CLK 周期；读 ERTC 寄存器占用 APB1 总线约 8 个 APB1\_CLK 周期。

RTC 寄存器是 16 位可寻址寄存器，具体描述如下：

表 17-4 ERTC 寄存器映像和复位值

寄存器简称	基址偏移量	复位值
ERTC_TIME	0x00	0x0000 0000
ERTC_DATE	0x04	0x0000 2101
ERTC_CTRL	0x08	0x0000 0000
ERTC_STS	0x0C	0x0000 0007
ERTC_DIV	0x10	0x007F 00FF
ERTC_ALA	0x1C	0x0000 0000
ERTC_WP	0x24	0x0000 0000
ERTC_SBS	0x28	0x0000 0000
ERTC_TADJ	0x2C	0x0000 0000
ERTC_TSTM	0x30	0x0000 0000
ERTC_TSDT	0x34	0x0000 000D
ERTC_TSSBS	0x38	0x0000 0000
ERTC_SCAL	0x3C	0x0000 0000
ERTC_TAMP	0x40	0x0000 0000
ERTC_ALASBS	0x44	0x0000 0000
ERTC_BPRx	0x50-0x60	0x0000 0000

### 17.4.1 ERTC时间寄存器(ERTC\_TIME)

域	简称	复位值	类型	功能
位 31: 23 保留		0x000	resd	保持默认值。
位 22	AMPM	0x0	rw	上午/下午 (AM/PM) 0: 上午; 1: 下午。 注: 该位只用于 12 小时制, 24 小时制保持为 0。
位 21: 20 HT		0x0	rw	小时十位 (Hour tens)
位 19: 16 HU		0x0	rw	小时个位 (Hour units)
位 15	保留	0x0	resd	保持默认值。
位 14: 12 MT		0x0	rw	分钟十位 (Minute tens)
位 11: 8 MU		0x0	rw	分钟个位 (Minute units)
位 7	保留	0x0	resd	保持默认值
位 6: 4 ST		0x0	rw	秒钟十位 (Second tens)
位 3: 0 SU		0x0	rw	秒钟个位 (Second units)

### 17.4.2 ERTC日期寄存器(ERTC\_DATE)

域	简称	复位值	类型	功能
位 31: 24 保留		0x00	resd	保持默认值。
位 23: 20 YT		0x0	rw	年份十位 (Year tens)
位 19: 16 YU		0x0	rw	年份个位 (Year units)
位 15: 13 WK		0x1	rw	星期 (Week) 0: 禁用; 1: 星期一; 2: 星期二; 3: 星期三; 4: 星期四; 5: 星期五; 6: 星期六; 7: 星期日。
位 12	MT	0x0	rw	月份十位 (Month tens)
位 11: 8 MU		0x1	rw	月份个位 (Month units)
位 7: 6 保留		0x0	resd	保持默认值。
位 5: 4 DT		0x0	rw	日期十位 (Date tens)
位 3: 0 DU		0x1	rw	日期个位 (Date units)

### 17.4.3 ERTC控制寄存器(ERTC\_CTRL)

域	简称	复位值	类型	功能
位 31: 24 保留		0x00	resd	保持默认值。
位 23	CALOEN	0x0	rw	校准输出使能 (Calibration output enable) 0: 关闭; 1: 开启。
位 22: 21 OUTSEL		0x0	rw	输出源选择 (Output source selection) 00: 关闭; 01: 闹钟 A; 10: 关闭; 11: 关闭
位 20	OUTP	0x0	rw	输出极性 (Output polarity) 0: 高; 1: 低。
位 19	CALOSEL	0x0	rw	校准输出选择 (Calibration output selection) 0: 512Hz; 1: 1Hz。
位 18	BPR	0x0	rw	电池供电域数据寄存器 (Battery power domain data register) 该位在电池供电域, 不受系统复位影响, 可用来存储夏令时操作或者一些其他需要一直保存的数据。
位 17	DEC1H	0x0	wo	减少 1 小时 (Decrease 1 hour)

				0: 无作用; 1: 减少一小时。 注: 当小时不为0时才有效, 该位置1后(不要在小时递增时置1), 下一秒生效。
位 16	ADD1H	0x0	wo	增加1小时(Add 1 hour) 0: 无作用; 1: 增加一小时。 注: 该位置1后(不要在小时递增时置1), 下一秒生效。
位 15	TSIEN	0x0	rw	时间戳中断使能(Timestamp interrupt enable) 0: 关闭; 1: 开启。
位 14: 13	保留	0x0	resd	保持默认值
位 12	ALAIEN	0x0	rw	闹钟A中断使能(Alarm A interrupt enable) 0: 关闭; 1: 开启。
位 11	TSEN	0x0	rw	时间戳使能(Timestamp enable) 0: 关闭; 1: 开启。
位 10: 9	保留	0x0	resd	保持默认值
位 8	ALAEN	0x0	rw	闹钟A使能(Alarm A enable) 0: 关闭; 1: 开启。
位 7	保留	0x0	resd	保持默认值
位 6	HM	0x0	rw	小时模式(Hour mode) 0: 24小时制; 1: 12小时制。
位 5	DREN	0x0	rw	日期/时间寄存器直接读取使能(Date/time register direct read enable) 0: 关闭, ERTC_TIME、ERTC_DATE、ERTC_SBS值从同步寄存器获取, 每两个ERTC_CLK更新一次; 1: 开启, ERTC_TIME、ERTC_DATE、ERTC_SBS值从电池供电域获取。
位 4	RCDEN	0x0	rw	参考时钟检测使能(Reference clock detection enable) 0: 关闭; 1: 开启。
位 3	TSEDG	0x0	rw	时间戳触发边沿(Timestamp trigger edge) 0: 上升沿; 1: 下降沿。
位 2: 0	保留	0x0	resd	保持默认值

#### 17.4.4 ERTC初始化和状态寄存器(ERTC\_STS)

域	简称	复位值	类型	功能
位 31: 17	保留	0x0000	resd	保持默认值。
位 16	CALUPDF	0x0	ro	校准值更新完成标志(Calibration value update completed flag) 0: 更新完成; 1: 正在更新。 当对精密校准寄存器ERTC_SCAL写时, 该位自动置1, 当ERTC使用新的校准值时, 该位自动清零, 当该位为1时, 不能写ERTC_SCAL寄存器。
位 15: 14	保留	0x0	resd	保持默认值。
位 13	TP1F	0x0	rw0c	入侵检测1标志(Tamper detection 1 flag) 0: 无入侵事件发生; 1: 有入侵事件发生。
位 12	TSOF	0x0	rw0c	时间戳溢出标志(Timestamp overflow flag) 0: 正常; 1: 溢出。 当产生了时间戳事件(TSF置1)时, 又产生了时间戳事件, 该标志置1。
位 11	TSF	0x0	rw0c	时间戳标志(Timestamp flag)

					0: 无时间戳事件发生; 1: 有时间戳事件发生。 当读取了时间戳，并清除了 TSF 时，建议再检查 TSOF 标志，因为可能会存在当正在清除 TSF 时又产生了时间戳事件。 注：该位清 0 后 2 个 APB_CLK 后生效。
位 10: 9 保留		0x0	resd	保持默认值	
位 8 ALAF		0x0	rw0c	闹钟 A 标志 (Alarm clock A flag) 0: 无闹钟事件发生; 1: 有闹钟事件发生。 注：该位清 0 后 2 个 APB_CLK 后生效。	
位 7 IMEN		0x0	rw	初始化模式使能 (Initialization mode enable) 0: 关闭; 1: 开启。 当进入了初始化模式后，日历处于停止状态。	
位 6 IMF		0x0	ro	进入初始化模式标志 (Enter initialization mode flag) 0: 未进入; 1: 进入。 当使能了初始化模式 (INITEN=1)，并进入了初始化模式 (INITEF=1) 时，才能更改 ERTC_TIME、ERTC_DATE、ERTC_DIV 寄存器。	
位 5 UPDF		0x0	rw0c	日历更新标志 (Calendar update flag) 0: 正在更新; 1: 更新完成。 每当从电池供电域将日历更新到 ERTC_TIME、ERTC_DATE、ERTC_SBS 同步寄存器，该标志置 1，每两个 ERTC_CLK 更新一次。	
位 4 INITF		0x0	ro	日历初始化标志 (Calendar initialization flag) 0: 未初始化; 1: 已初始化。 当检测到 ERTC_DATE 里面的年份不为 0 时该位置 1，当年份为 0 时，该位清 0。	
位 3 TADJF		0x0	ro	时间调整标志 (Time adjustment flag) 0: 无操作; 1: 正在执行时间调整。 当对时间调整存器 ERTC_TADJ 写时，该位自动置 1，当时间调整结束后，该位自动清零。	
位 2: 1 保留		0x0	resd	保持默认值	
位 0 ALAWF		0x1	ro	闹钟 A 允许写标志 (Alarm A register allows write flag) 0: 不允许写; 1: 允许写。	

#### 17.4.5 ERTC 预分频器寄存器(ERTC\_DIV)

域	简称	复位值	类型	功能
位 31: 23 保留		0x000	resd	保持默认值。
位 22: 16 DIVA		0x7F	rw	分频器 A (Diveder A)
位 15 保留		0x0	resd	保持默认值。
位 14: 0 DIVB		0x00FF	rw	分频器 B (Diveder B) 日历时钟=ERTC_CLK/((DIVA+1)x(DIVB+1))。

#### 17.4.6 ERTC 阔钟 A 寄存器(ERTC\_ALA)

域	简称	复位值	类型	功能
位 31 MASK4		0x0	rw	日期/星期屏蔽 (Date/week mask) 0: 无屏蔽; 1: 阔钟和日期/星期无关。
位 30 WKSEL		0x0	rw	日期/星期选择 (Date/week mode) 0: 日期; 1: 星期 (DT[1: 0]不使用)。
位 29: 28 DT		0x0	rw	日期十位 (Date tens)

位 27: 24 DU	0x0	rw	日期/星期个位 (Date/week units)
			小时屏蔽 (Hour mask)
位 23 MASK3	0x0	rw	0: 无屏蔽; 1: 闹钟和小时无关。
			上午/下午 (AM/PM)
位 22 AMPM	0x0	rw	0: 上午; 1: 下午。 注: 该位只用于 12 小时制, 24 小时制保持为 0。
位 21: 20 HT	0x0	rw	小时十位 (Hour tens)
位 19: 16 HU	0x0	rw	小时个位 (Hour units)
			分钟屏蔽 (Minute mask)
位 15 MASK2	0x0	rw	0: 无屏蔽; 1: 闹钟和分钟无关。
位 14: 12 MT	0x0	rw	分钟十位 (Minute tens)
位 11: 8 MU	0x0	rw	分钟个位 (Minute units)
			秒钟屏蔽 (Second mask)
位 7 MASK1	0x0	rw	0: 无屏蔽; 1: 闹钟和秒钟无关。
位 6: 4 ST	0x0	rw	秒钟十位 (Second tens)
位 3: 0 SU	0x0	rw	秒钟个位 (Second units)

#### 17.4.7 ERTC写保护寄存器(ERTC\_WP)

域	简称	复位值	类型	功能
位 31: 8 保留		0x000000	resd	保持默认值。
位 7: 0 CMD		0x00	wo	命令寄存器 (Command register) 依次写入 0xCA、0x53 解锁所有 ERTC 寄存器写保护, 当写一个其他值时, 将重新开启写保护。

#### 17.4.8 ERTC亚秒寄存器(ERTC\_SBS)

域	简称	复位值	类型	功能
位 31: 16 保留		0x0000	resd	保持默认值。
位 15: 0 SBS		0x0000	ro	亚秒值 (Sub-second value) 亚秒为分频器 DIVB 的计数值, 时钟频率为 ERTC_CLK/(DIVA+1)。

#### 17.4.9 ERTC时间微调寄存器(ERTC\_TADJ)

域	简称	复位值	类型	功能
位 31 ADD1S		0x0	wo	增加一秒 (Add 1 second) 0: 无效果; 1: 增加 1 秒。 当 TADJF=0 时, 才能写此寄存器, 通常 ADD1S 与 DECSBS 配合使用, 达到微调时间的效果。
位 30: 15 保留		0x0000	resd	保持默认值。
位 14: 0 DECSBS		0x0000	wo	DECSBS[14: 0]: 减少亚秒值 (Decrease sub-second value) 延迟时间 (ADD1S=0) : 延迟=DECSBS/(DIVB+1); 提前时间 (ADD1S=1) : 延迟=1-(DECSBS/(DIVB+1))。 当 TADJF=0 时, 才能写此寄存器。

#### 17.4.10 ERTC时间戳时间寄存器(ERTC\_TSTM)

域	简称	复位值	类型	功能
位 31: 23 保留		0x000	resd	保持默认值。
位 22 AMPM		0x0	ro	上午/下午 (AM/PM) 0: 上午; 1: 下午。 注: 该位只用于 12 小时制, 24 小时制保持为 0。
位 21: 20 HT		0x0	ro	小时十位 (Hour tens)
位 19: 16 HU		0x0	ro	小时个位 (Hour units)

位 15	保留	0x0	resd	保持默认值。
位 14:12	MT	0x0	ro	分钟十位 (Minute tens)
位 11: 8	MU	0x0	ro	分钟个位 (Minute units)
位 7	保留	0x0	resd	保持默认值。
位 6: 4	ST	0x0	ro	秒钟十位 (Second tens)
位 3: 0	SU	0x0	ro	秒钟个位 (Second units)

注意：仅当 ERTC\_STS 中的 TSF 置 1 时，该寄存器的内容才有效。当 TSF 位复位时，清零该寄存器。

#### 17.4.11 ERTC时间戳日期寄存器(ERTC\_TS DT)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0000	resd	保持默认值。
位 15: 13	WK	0x0	ro	星期 (Week)
位 12	MT	0x0	ro	月十位 (Month tens)
位 11: 8	MU	0x0	ro	月个位 (Month units)
位 7: 6	保留	0x0	resd	保持默认值。
位 5: 4	DT	0x0	ro	日期十位 (Date tens)
位 3: 0	DU	0x0	ro	日期个位 (Date units)

注意：仅当 ERTC\_STS 中的 TSF 置 1 时，该寄存器的内容才有效。当 TSF 位复位时，清零该寄存器。

#### 17.4.12 ERTC时间戳亚秒寄存器(ERTC\_TSSBS)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0000	resd	保持默认值。
位 15: 0	SBS	0x0000	ro	亚秒值 (Sub-second value)

注意：仅当 ERTC\_STS/TSF 置 1 时，该寄存器的内容才有效。当 TSF 位复位时，清零该寄存器。

#### 17.4.13 ERTC精密校准寄存器(ERTC\_SCAL)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0000	resd	保持默认值。
位 15	ADD	0x0	rw	增加 ERTC 时钟 (Add ERTC clock) 0: 无操作; 1: 每 $2^{11}$ 个 ERTC_CLK, 插入一个 ERTC_CLK。
位 14	CAL8	0x0	rw	8 秒校准周期 (8-second calibration period) 0: 无效果; 1: 8 秒校准周期。
位 13	CAL16	0x0	rw	16 秒校准周期 (16 second calibration period) 0: 无效果; 1: 16 秒校准周期。
位 12: 9	保留	0x0	resd	保持默认值
位 8: 0	DEC	0x000	rw	减少 ERTC 时钟 (Decrease ERTC clock) 在 $2^{20}$ 个 ERTC_CLK 周期内, 屏蔽 DEC 个 ERTC_CLK。 通常和 ADD 配合使用, 当 ADD 为 1 时, 在 $2^{20}$ 个 ERTC_CLK 周期内, 实际的 ERTC_CLK 个数为 $2^{20}+512$ -DEC。

#### 17.4.14 ERTC入侵配置寄存器(ERTC\_TAMP)

域	简称	复位值	类型	功能
位 31: 19	保留	0x0000	resd	保持默认值。
位 18	OUTTYPE	0x0	rw	输出类型 (Output type) 0: 开漏; 1: 推挽。
位 17: 16	保留	0x0	resd	保持默认值
位 15	TPPU	0x0	rw	入侵检测上拉 (Tamper detection pull-up) 0: 开启; 1: 关闭。

				入侵检测预充电时间 (Tamper detection pre-charge time) 0: 1 个 ERTC_CLK; 1: 2 个 ERTC_CLK; 2: 4 个 ERTC_CLK; 3: 8 个 ERTC_CLK。
位 14:13 TPPR	0x0	rw		入侵检测滤波时间 (Tamper detection filter time) 0: 无滤波; 1: 连续 2 次采样有效, 判定入侵事件发生; 2: 连续 4 次采样有效, 判定入侵事件发生; 3: 连续 8 次采样有效, 判定入侵事件发生。
位 12:11 TPFLT	0x0	rw		入侵检测检测频率 (Tamper detection frequency) 0: ERTC_CLK/32768; 1: ERTC_CLK/16384; 2: ERTC_CLK/8192; 3: ERTC_CLK/4096; 4: ERTC_CLK/2048; 5: ERTC_CLK/1024; 6: ERTC_CLK/512; 7: ERTC_CLK/256。
位 10:8 TPFREQ	0x0	rw		入侵检测时间戳使能 (Tamper detection timestamp enable) 0: 关闭; 1: 开启, 当产生入侵事件时, 保持时间戳。
位 7 TPTSEN	0x0	rw		位 6: 3 保留 0x0 resd 保持默认值。
位 2 TPIEN	0x0	rw		入侵检测中断使能 (Tamper detection interrupt enable) 0: 关闭; 1: 开启。
位 1 TP1EDG	0x0	rw		入侵检测 1 有效边沿 (Tamper detection 1 valid edge) 当无滤波时 (TPFLT=0) : 0: 上升沿; 1: 下降沿。 当有滤波时 (TPFLT>0) : 0: 低电平; 1: 高电平。
位 0 TP1EN	0x0	rw		入侵检测 1 使能 Tamper detection 1 enable 0: 关闭; 1: 开启。

#### 17.4.15 ERTC闹钟A亚秒寄存器(ERTC\_ALASBS)

域	简称	复位值	类型	功能
位 31: 28 保留		0x0	resd	保持默认值。
位 27: 24 SBSMSK		0x0	rw	亚秒屏蔽 (Sub-second mask) 0: 不匹配亚秒, 闹钟与亚秒无关; 1: 只匹配 SBS[0]; 2: 只匹配 SBS[1: 0]; 3: 只匹配 SBS[2: 0]; ... 14: 只匹配 SBS[13: 0]; 15: 匹配 SBS[14: 0]。
位 23: 15 保留		0x000	rw	保持默认值。
位 14: 0 SBS		0x0000	rw	亚秒值 (Sub-second value)

#### 17.4.16 ERTC电池供电数据寄存器(ERTC\_BPRx)

域	简称	复位值	类型	功能
位 31: 0 DT		0x00000000	rw	电池供电域数据 (Battery powered domain data) BPR_DTx 寄存器, 可以在只由电池供电域供电下保存数据, 不会被系统复位所复位, 只能通过电池供电域复位或入侵事件进行复位。

# 18 模拟/数字转换 (ADC)

## 18.1 ADC简介

ADC 是一个将模拟输入信号转换为 12 位数字信号的外设。采样率最高可达 2MSPS。多达 18 个通道源可进行采样及转换。

## 18.2 ADC主要特征

模拟方面有以下特征：

- 支持分辨率 12 位的转换
- 自校准时间： 154 个 ADC 时钟周期
- ADC 转换时间
- ADC 时钟在最大频率 28MHz 时转换时间为 0.5  $\mu$ s (系统时钟为 120MHz 时， ADC 时钟最大频率为 20M, 转换时间则为 0.7  $\mu$ s)
- ADC 供电要求：请参考 Datasheet
- ADC 输入范围： $V_{REF-} \leq V_{IN} \leq V_{REF+}$

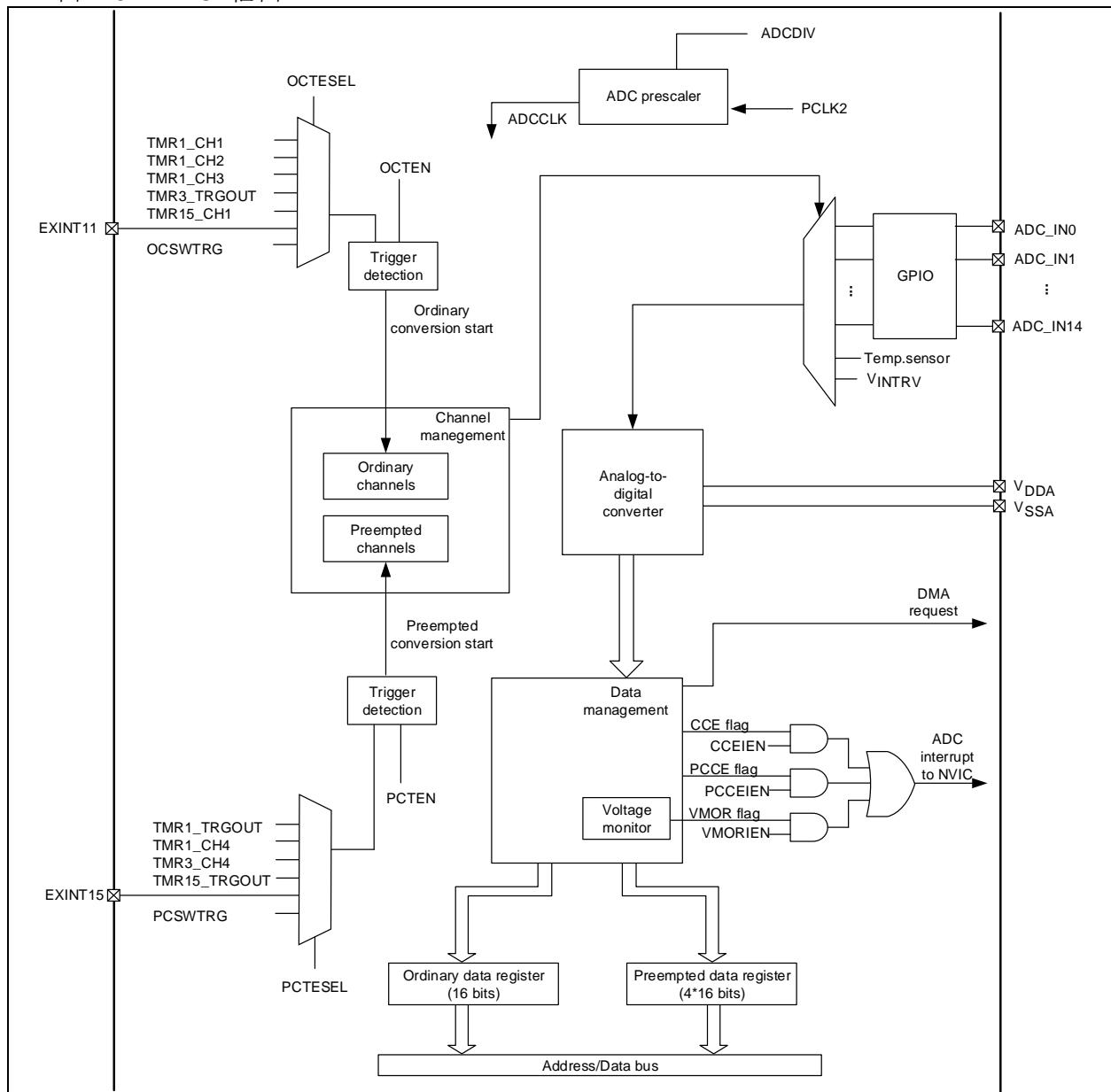
数字控制方面有以下特征：

- 通道管理区分优先权不同的普通通道与抢占通道
- 普通通道与抢占通道具备各自独立的触发侦测电路
- 各通道均可独立配置采样时间
- 转换顺序管理支持多种不同的多通道转换
- 可选择的数据对齐方式
- 可配置的电压监测边界
- 支持 DMA 传输的普通通道数据
- 可设定以下事件发生时响应中断
  - 抢占通道组转换结束
  - 通道转换结束
  - 电压监测超出范围

## 18.3 ADC架构

ADC1的架构如图 18-1所示。

图 18-1 ADC1框图



输入管脚介绍：

- V<sub>DDA</sub>：模拟电源，ADC 模拟电源
- V<sub>SSA</sub>：模拟电源地，ADC 模拟电源地
- ADC\_INx：模拟输入信号通道
- 输入管脚的连接与电压范围限制请参考 Datasheet

## 18.4 ADC功能介绍

### 18.4.1 通道管理

#### 模拟信号通道输入

每个ADC拥有多达18个模拟信号通道输入，以ADC\_INx表示，x=0至17。

- ADC\_IN0至ADC\_IN14为外部模拟输入，ADC\_IN15为V<sub>ss</sub>，ADC\_IN16为内部温度传感器，ADC\_IN17为内部参考电压。

#### 通道转换

转换区分为普通通道转换与抢占通道转换，抢占通道的转换优先权高于普通通道。

抢占通道触发若发生于普通通道转换途中，优先进行抢占通道的转换，普通通道于抢占通道转换结束后重新开始转换被打断的通道。普通通道触发若发生于抢占通道转换途中，普通通道的转换会等待抢占通道转换完成后才开始。

将通道 (ADC\_INx) 编排进普通通道序列 (ADC\_OSQx) 以及抢占通道序列 (ADC\_PSQ)，相同通道可重复编排，序列总数由 OCLEN 与 PCLEN 定义，接着即可启动普通通道转换或抢占通道转换。

#### 18.4.1.1 内部温度传感器

温度传感器接到 ADC1\_IN16，必须先使能 ADC 控制寄存器 2 (ADC\_CTRL2) 的 ITSRVEN 位并且等待上电时间后才可对温度传感通道进行转换。

转换后获得的数据，搭配数据手册的电气特性章节提供的  $25^{\circ}\text{C}$  的电压值与数据对温度斜率 (Avg\_Slope)，即可推算温度。

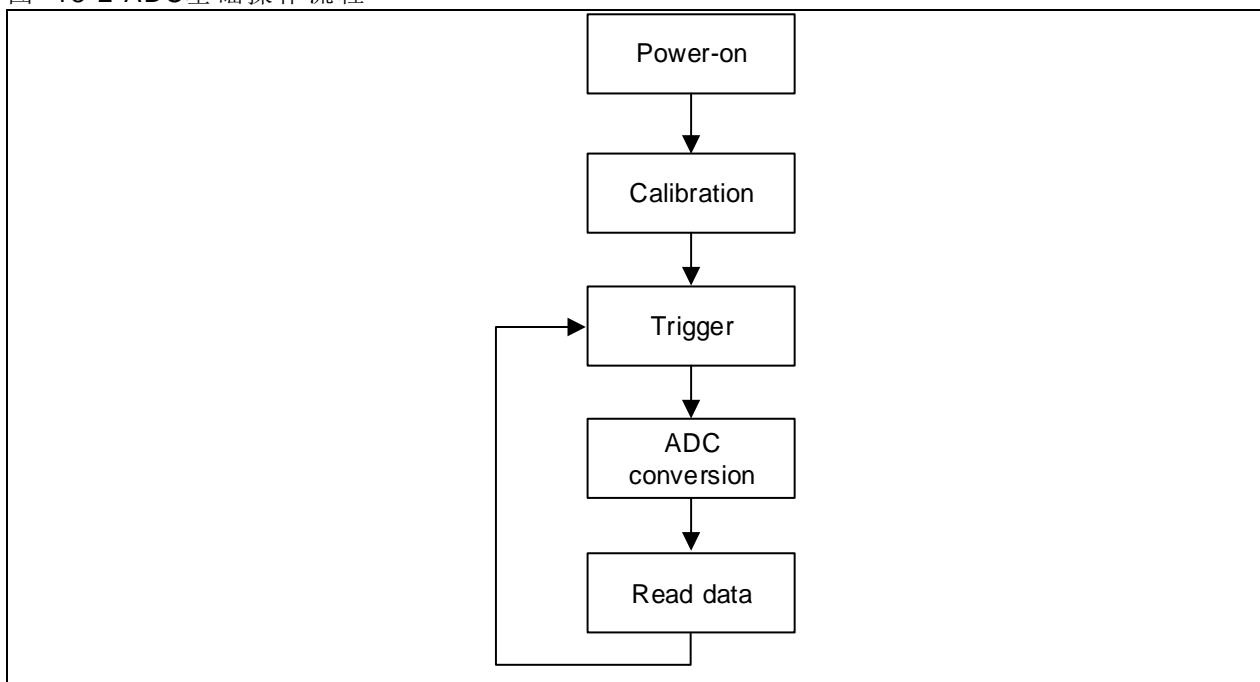
#### 18.4.1.2 内部参考电压

典型值  $1.2\text{V}$  的内部参考电压接到 ADC1\_IN17，必须先使能 ADC 控制寄存器 2 (ADC\_CTRL2) 的 ITSRVEN 位后才可对内部参考电压通道进行转换。此通道的转换数据可用于推算外部参考电压。

### 18.4.2 ADC操作流程

ADC 的基础操作流程如图所示，建议第一次上电后进行校准，以提升采样与转换准确度。待校准完成后可靠触发引起 ADC 采样转换，转换结束后即可读取数据。

图 18-2 ADC基础操作流程



#### 18.4.2.1 上电与校准

##### 上电

用户须先使能 APB2 外设时钟使能寄存器 (CRM\_APB2EN) 的 ADCEN，以使能 ADC 的时钟：PCLK2 与 ADCCLK。

时钟使能后必须配置 ADC 预分频器(CRM\_CFG 的 ADCDIV)，将 ADCCLK 调整至需求的频率。ADCCLK 由 PCLK2 除频而来。

**注意：** ADCCLK 不可大于  $28\text{MHz}$ 。

ADCCLK 频率调整完后，即可使能 ADC 控制寄存器 2 (ADC\_CTRL2) 的 ADCEN 位使 ADC 上电，等待  $t_{STAB}$  后才可对 ADC 进行后续操作。清除 ADCEN 会使 ADC 的转换中止并复位，同时 ADC 被断电以达到省电的效果。

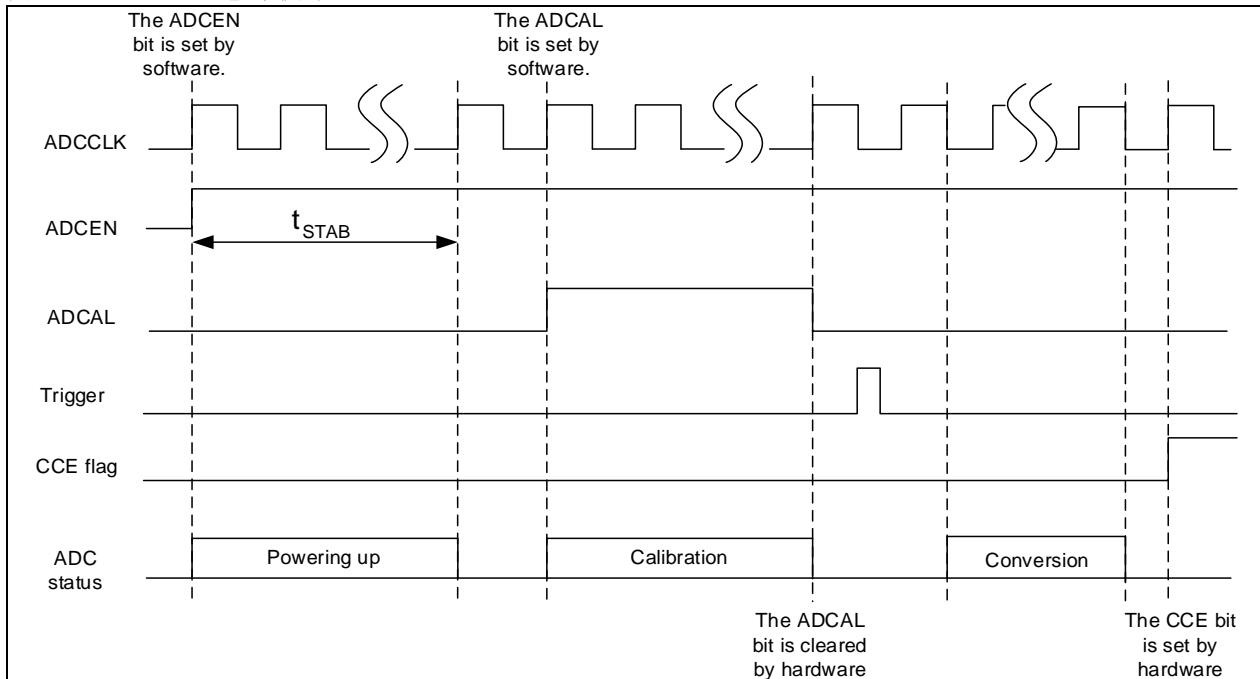
##### 校准

上电完成后可设置 ADC 控制寄存器 2 (ADC\_CTRL2) 的 ADCAL 使 ADC 进行校准，校准完成后硬件清

除 ADCAL 位，软件即可触发以进行转换。

每次校准后，校准值会被存放至 ADC 普通数据寄存器（ADC\_ODT）中，这个校准值自动反馈回 ADC 内部，以消除电容误差。该校准值的存放不会置位 CCE 标志，不会产生中断或 DMA 请求。

图 18-3 ADC 上电与校准



#### 18.4.2.2 触发

ADC 触发分为普通通道触发与抢占通道触发，普通通道触发引发普通通道转换，抢占通道触发引发抢占通道转换。使能 ADC 控制寄存器 2 (ADC\_CTRL2) 的 OCTEN 或 PCTEN 后，ADC 才会检测触发来源的上升沿并响应转换。

触发来源可分为软件写寄存器触发 ADC 控制寄存器 2 (ADC\_CTRL2) 的 OCSWTRG 与 PCSWTRG 以及外部触发，外部触发包含定时器触发与管脚触发，由 ADC 控制寄存器 2(ADC\_CTRL2)的 OCTESEL 与 PCTESEL 选择触发来源，如表 18-1 与错误!未找到引用源。所示。

普通通道还有一种特殊的触发来源，即重复使能 ADCEN 触发转换。此种情况下不需要使能 ADC 控制寄存器 2 (ADC\_CTRL2) 的 OCTEN 也可导致普通通道响应转换。

表 18-1 ADC1 的触发来源

OCTESEL	触发来源	PCTESEL	触发来源
000	TMR1_CH1 event	000	TMR1_TRGOUT event
001	TMR1_CH2 event	001	TMR1_CH4 event
010	TMR1_CH3 event	010	保留
011	保留	011	保留
100	TMR3_TRGOUT event	100	TMR3_CH4 event
101	TMR15_CH1 event	101	TMR15_TRGOUT event
110	EXINT line11 external pin	110	EXINT line15 external pin
111	OCSWTRG	111	PCSWTRG

#### 18.4.2.3 采样与转换时序

用户可于 ADC 采样时间寄存器 1 (ADC\_SPT1) 与 ADC 采样时间寄存器 2 (ADC\_SPT2) 的 CSPTx 配置各个通道 (ADC\_INx) 的采样周期。一次转换所需的时间可利用以下公式推得：

$$\text{一次转换所需的时间(ADCCLK 的周期)} = \text{采样时间} + 12.5$$

示例：

CSPTx 选择 1.5 周期，一次转换需要  $1.5+12.5=14$  个 ADCCLK 周期。

CSPTx 选择 7.5 周期，一次转换需要  $7.5+12.5=20$  个 ADCCLK 周期。

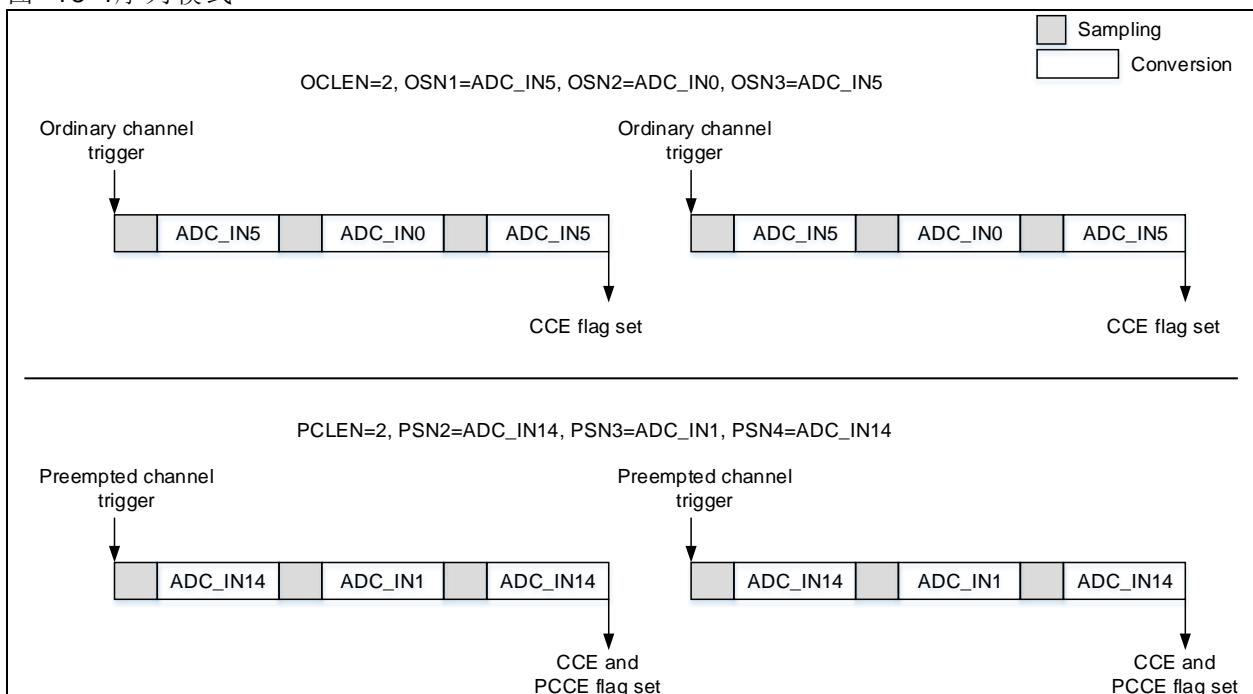
### 18.4.3 转换顺序管理

默认模式下，每次触发只会转换单个通道，即 OSN1（普通触发）或 PSN4（抢占触发）记录的通道。下面介绍不同的转换顺序模式，即可使多个通道以特定顺序做转换。

#### 18.4.3.1 序列模式

使能 ADC 控制寄存器 1 (ADC\_CTRL1) 的 SQEN，即开启序列模式，用户于 ADC 普通序列寄存器 (ADC\_OSQx) 配置普通通道顺序与总数，于 ADC 抢占序列寄存器 (ADC\_PSQ) 配置抢占通道顺序与总数，开启序列模式后，一次触发将序列中的通道依序转换一次。普通通道从 OSN1 开始转换起，抢占通道是从 PSNx 开始转换起， $x=4-PCLEN$ ，下图示范了序列模式的行为。

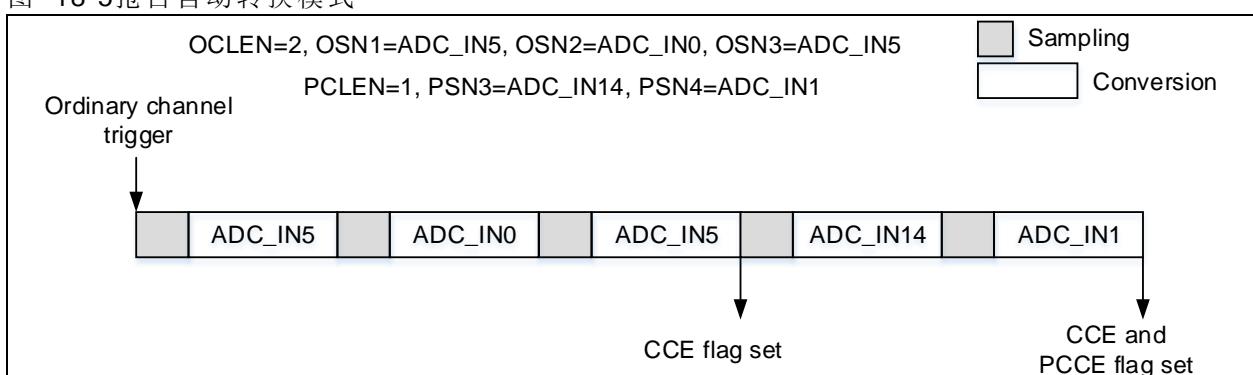
图 18-4 序列模式



#### 18.4.3.2 抢占自动转换模式

使能 ADC 控制寄存器 1 (ADC\_CTRL1) 的 PCAUTOEN，即开启抢占自动转换模式，当普通通道转换完成后，抢占通道将自动接续着转换。可与序列模式共用，当普通通道序列完成后，即会自动开始抢占序列的转换。下图示范了与序列模式共用的抢占自动转换模式行为。

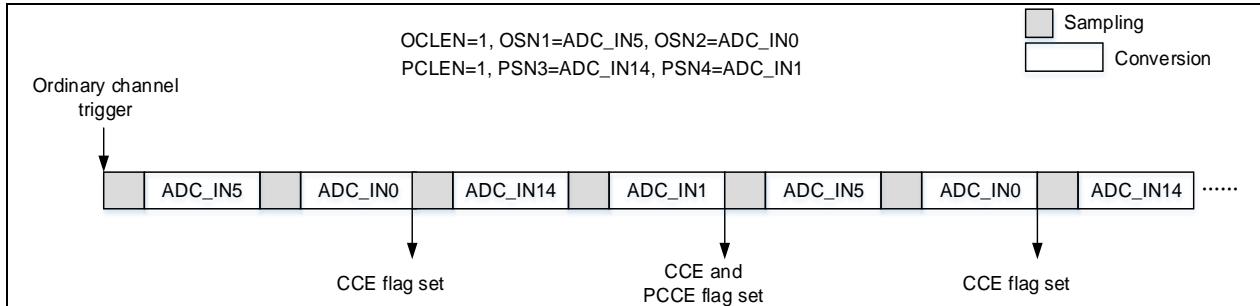
图 18-5 抢占自动转换模式



#### 18.4.3.3 反复模式

使能 ADC 控制寄存器 2 (ADC\_CTRL2) 的 RPEN，即开启反复模式。当普通通道检测到触发后就即会反复不断地转换。可与序列模式下的普通通道转换共用，将反复地转换普通通道序列。也可与抢占自动转换模式共用，将依次反复地转换普通通道序列与抢占通道序列。下图示范了与序列模式及抢占自动转换模式共用的反复模式行为。

图 18-6 反复模式



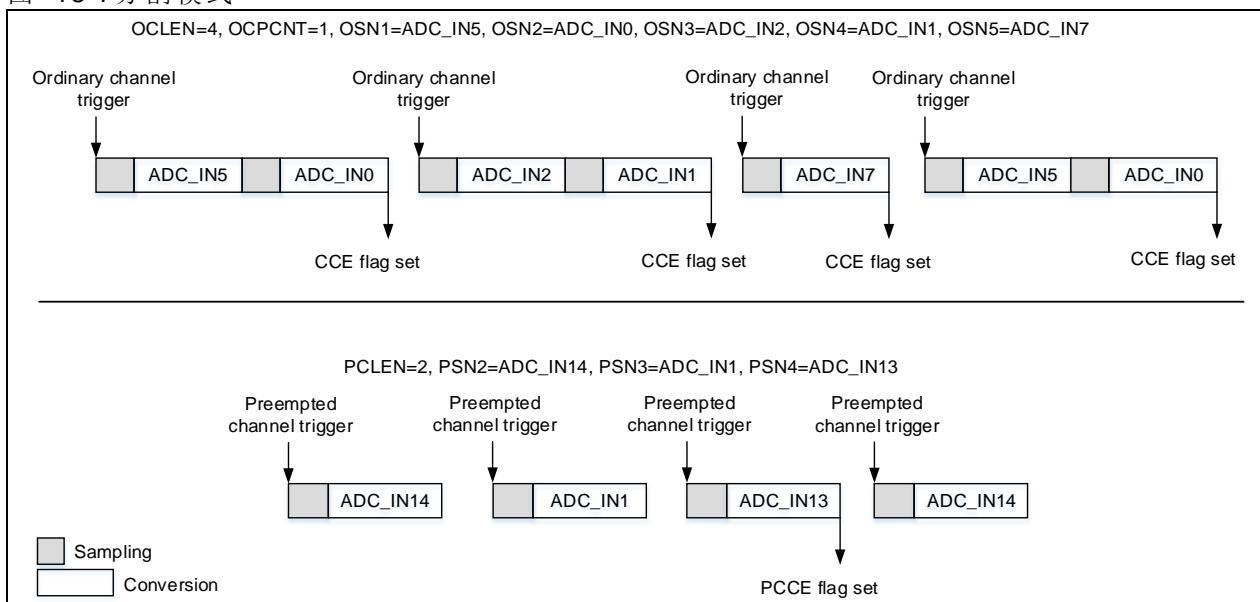
#### 18.4.3.4 分割模式

使能 ADC 控制寄存器 1 (ADC\_CTRL1) 的 OCPEN，即开启普通通道的分割模式，此模式将 ADC 普通序列寄存器 1 (ADC\_OSQ1) 的 OCLEN 的序列长度分割成长度较小的子组别，子组别的通道数于 ADC 控制寄存器 1 (ADC\_CTRL1) 的 OCPCNT 配置，一次触发将转换子组别中的所有通道。每次触发会依序选择不同的子组别。

使能 ADC 控制寄存器 1 (ADC\_CTRL1) 的 PCPEN，即开启抢占通道的分割模式，此模式将 ADC 普通序列寄存器 1 (ADC\_OSQ1) 的 PCLEN 的序列长度分割成只有一个通道的子组别，一次触发将转换子组别中的通道。每次触发会依序选择不同的子组别。

分割模式与反复模式不可共用。下图分别示范了普通分割与抢占分割模式的行为。

图 18-7 分割模式



#### 18.4.4 数据管理

普通通道转换完成后数据存储于 ADC 普通数据寄存器 (ADC\_ODT)，抢占通道转换完成后数据存储于 ADC 抢占数据寄存器 x (ADC\_PDTx)。

##### 18.4.4.1 数据内容处理

由 ADC 控制寄存器 2 (ADC\_CTRL2) 的 DTALIGN 选择转换数据靠右或是靠左对齐放置于数据寄存器，除此之外，抢占通道的数据还会减去 ADC 抢占通道数据偏移寄存器 x (ADC\_PCDTOx) 的偏移量，因此抢占通道数据有可能为负值，以 SIGN 作为符号。如图所示。

图 18-8 数据内容处理

Ordinary channel data 12 bits															
Right-alignment								DT[11]	DT[10]	DT[9]	DT[8]	DT[7]	DT[6]	DT[5]	DT[4]
Left-alignment								DT[11]	DT[10]	DT[9]	DT[8]	DT[7]	DT[6]	DT[5]	DT[4]
Preempted channel data 12 bits															
Right-alignment								SIGN	SIGN	SIGN	SIGN	DT[11]	DT[10]	DT[9]	DT[8]
Left-alignment								SIGN	DT[11]	DT[10]	DT[9]	DT[8]	DT[7]	DT[6]	DT[5]

#### 18.4.4.2 数据获取

普通通道转换数据可藉由 CPU 或 DMA 读取 ADC 普通数据寄存器 (ADC\_ODT) 获得。抢占通道数据只可藉由 CPU 读取 ADC 抢占数据寄存器 x (ADC\_PDTx) 获得。

使能 ADC 控制寄存器 2 (ADC\_CTRL2) 的 OCDMAEN 后, ADC 会在每次 ADC 普通数据寄存器 (ADC\_ODT) 更新时请求 DMA。

#### 18.4.5 电压监测

使能 ADC 控制寄存器 1 (ADC\_CTRL1) 的 OCVMEN (普通通道) 或 PCVMEN (抢占通道) 即可通过对转换结果的判定来实现电压监测。当转换结果大于高边界 ADC 电压监测高边界寄存器 (ADC\_VMHB) 或是小于低边界 ADC 电压监测低边界寄存器 (ADC\_VMLB) 时, 电压监测超出标志 VMOR 会置起。透过 VMSGEN 选择对单一特定通道或是所有通道监测。对单一通道监测的话, 由 VMCSEL 配置通道。电压监测一律以转换的原始数据与 12 位边界寄存器做比较, 无视 PCDTOx 与 DTALIGN 位的设定。

#### 18.4.6 状态标志与中断

每个 ADC 拥有自己的 ADC 状态寄存器 (ADCx\_STS): 普通通道转换开始标志 (OCCS)、抢占通道转换开始标志 (PCCS)、抢占通道组转换结束标志 (PCCE)、通道转换结束标志 (CCE) 及电压监测超出标志 (VMOR)。

其中抢占通道组转换结束标志、通道转换结束标志及电压监测超出标志拥有对应中断使能位, 只要将中断使能, 标志置起时便会对 CPU 发出中断。

### 18.5 ADC 寄存器

下表列出了 ADC 寄存器的映像和复位值。

必须以字(32 位)的方式操作这些外设寄存器。

表 18-2 ADC 寄存器映像和复位值

寄存器简称	基址偏移量	复位值
ADC_STS	0x000	0x0000 0000
ADC_CTRL1	0x004	0x0000 0000
ADC_CTRL2	0x008	0x0000 0000
ADC_SPT1	0x00C	0x0000 0000
ADC_SPT2	0x010	0x0000 0000
ADC_PCDTO1	0x014	0x0000 0000
ADC_PCDTO2	0x018	0x0000 0000
ADC_PCDTO3	0x01C	0x0000 0000
ADC_PCDTO4	0x020	0x0000 0000
ADC_VMHB	0x024	0x0000 0FFF

ADC_VMLB	0x028	0x0000 0000
ADC_OSQ1	0x02C	0x0000 0000
ADC_OSQ2	0x030	0x0000 0000
ADC_OSQ3	0x034	0x0000 0000
ADC_PSQ	0x038	0x0000 0000
ADC_PDT1	0x03C	0x0000 0000
ADC_PDT2	0x040	0x0000 0000
ADC_PDT3	0x044	0x0000 0000
ADC_PDT4	0x048	0x0000 0000
ADC_ODT	0x04C	0x0000 0000

### 18.5.1 ADC状态寄存器 (ADC\_STS)

访问：字访问

域	简称	复位值	类型	功能
位 31: 5	保留	0x00000000	resd	请保持默认值。
位 4	OCCS	0x0	rw0c	普通通道转换开始标志 (Ordinary channel conversion start flag) 该位被硬件置起，由软件将其清零（对自身写零）。 0: 未开始； 1: 已开始。
位 3	PCCS	0x0	rw0c	抢占通道转换开始标志 (Preempted channel conversion start flag) 该位被硬件置起，由软件将其清零（对自身写零）。 0: 未开始； 1: 已开始。
位 2	PCCE	0x0	rw0c	抢占通道组转换结束标志 (Preempted channels conversion end flag) 该位被硬件置起，由软件将其清零（对自身写零）。 0: 未结束； 1: 已结束。
位 1	CCE	0x0	rw0c	通道转换结束标志 (Channels conversion end flag) 该位被硬件置起，由软件将其清零（对自身写零），或由读取 ADC 普通数据寄存器 (ADC_ODT) 清零。 0: 未结束； 1: 已结束。 注：普通或抢占通道组转换结束均会置位此标志。
位 0	VMOR	0x0	rw0c	电压监测超出范围标志 (Voltage monitoring out of range flag) 该位被硬件置起，由软件将其清零（对自身写零）。 0: 无超出； 1: 有超出。

### 18.5.2 ADC控制寄存器1 (ADC\_CTRL1)

访问：字访问

域	简称	复位值	类型	功能
位 31: 24	保留	0x00	resd	请保持默认值。

位 23	OCVMEN	0x0	rw	普通通道的电压监测使能 (Voltage monitoring enable on ordinary channels) 0: 关闭; 1: 开启。
位 22	PCVMEN	0x0	rw	抢占通道的电压监测使能 (Voltage monitoring enable on preempted channels) 0: 关闭; 1: 开启。
位 21: 16	保留	0x0	resd	请保持默认值。
位 15: 13	OCPCNT	0x0	rw	分割模式下每次触发转换的普通通道个数 (Partitioned mode conversion count of ordinary channels) 000: 1 个通道; 001: 2 个通道; ..... 111: 8 个通道。 注: 抢占组在分割模式下每次触发固定只转换一个通道。
位 12	PCPEN	0x0	rw	抢占通道上的分割模式使能 (Partitioned mode enable on preempted channels) 0: 关闭; 1: 开启。
位 11	OCPEN	0x0	rw	普通通道上的分割模式使能 (Partitioned mode enable on ordinary channels) 该位由软件设置和清除, 用于开启或关闭普通通道组上的分割模式 0: 关闭; 1: 开启。
位 10	PCAUTOEN	0x0	rw	普通组转换结束后的抢占组自动转换使能 (Preempted group automatic conversion enable after ordinary group) 0: 关闭; 1: 开启。
位 9	VMSGEN	0x0	rw	单个通道的电压监测使能 (Voltage monitoring enable on a single channel) 0: 关闭 (电压监测所有通道); 1: 开启 (电压监测单一通道)。
位 8	SQEN	0x0	rw	序列模式使能 (Sequence mode enable) 0: 关闭 (转换选择的单一通道); 1: 开启 (转换设定的多个通道)。 注: 如果开启多通道模式, 且开启了 CCEIEN 或 PCCEIEN 位, 则只在最后一个通道转换完毕后才会产生 CCE 或 PCCE 中断。
位 7	PCCEIEN	0x0	rw	抢占通道组转换结束中断使能 (conversion end interrupt enable for Preempted channels) 0: 关闭; 1: 开启。
位 6	VMORIEN	0x0	rw	电压监测超出范围中断使能 (Voltage monitoring out of range interrupt enable) 0: 关闭; 1: 开启。

位 5	CCEIEN	0x0	rw	通道转换结束中断使能 (Channel conversion end interrupt enable) 0: 关闭; 1: 开启。
位 4: 0	VMCSEL	0x00	rw	电压监测通道选择 (Voltage monitoring channel select) 仅在 VMSGEN 开启时有效。 00000: ADC_IN0 通道; 00001: ADC_IN1 通道; ..... 01111: ADC_IN15 通道; 10000: ADC_IN16 通道; 10001: ADC_IN17 通道。 10010~11111: 未用, 禁止配置。

### 18.5.3 ADC控制寄存器2 (ADC\_CTRL2)

访问: 字访问

域	简称	复位值	类型	功能
位 30: 24	保留	0x00	resd	请保持默认值。
位 23	ITSRVEN	0x0	rw	内部温度传感器及 VINTRV 使能 (Internal temperature sensor and VINTRV enable) 0: 关闭; 1: 开启。
位 22	OCSWTRG	0x0	rw	软件触发普通通道转换 (Conversion trigger by software of ordinary channels) 0: 不触发; 1: 触发转换 (可由软件清除, 或在转换开始后由硬件自动清除)。
位 21	PCSWTRG	0x0	rw	软件触发抢占通道转换 (Conversion trigger by software of preempted channels) 0: 不触发; 1: 触发转换 (可由软件清除, 或在转换开始后由硬件自动清除)。
位 20	OCTEN	0x0	rw	普通通道组转换的触发模式使能 (Trigger mode enable for ordinary channels conversion) 0: 关闭; 1: 开启。
位 19: 17	OCTESEL	0x0	rw	普通通道组转换的触发事件选择 (trigger event select for ordinary channels conversion) ADC1 的触发配置如下 000: 定时器 1 的 CH1 事件; 001: 定时器 1 的 CH2 事件; 010: 定时器 1 的 CH3 事件; 011: 未用, 禁止配置; 100: 定时器 3 的 TRGOUT 事件; 101: 定时器 15 的 CH1 事件; 110: EXINT 线 11; 111: OCSWTRG。
位 16	保留	0x0	resd	请保持默认值。

位 15	PCTEN	0x0	rw	抢占通道组转换的触发模式使能 (Trigger mode enable for preempted channels conversion) 0: 关闭; 1: 开启。
位 14: 12	PCTESEL	0x0	rw	抢占通道组转换的触发事件选择 (trigger event select for preempted channels conversion) ADC1 的触发配置如下 000: 定时器 1 的 TRGOUT 事件; 001: 定时器 1 的 CH4 事件; 010: 未用, 禁止配置; 011: 未用, 禁止配置; 100: 定时器 3 的 CH4 事件; 101: 定时器 15 的 TRGOUT 事件; 110: EXINT 线 15; 111: PCSWTRG。
位 11	DTALIGN	0x0	rw	数据对齐方式 (Data alignment) 0: 右对齐; 1: 左对齐。
位 10: 9	保留	0x0	resd	请保持默认值。
位 8	OCDMAEN	0x0	rw	普通通道转换数据的 DMA 传输使能 (DMA transfer enable of ordinary channels) 0: 关闭; 1: 开启。
位 7: 4	保留	0x0	resd	请保持默认值。
位 3	ADCALINIT	0x0	rw	A/D 初始化校准 (initialize A/D calibration) 该位由软件设置并由硬件清除。在校准寄存器被初始化后该位将被清除。 0: 校准寄存器无初始化执行或初始化结束; 1: 校准寄存器初始化或初始化进行中。
位 2	ADCAL	0x0	rw	A/D 校准 (A/D Calibration) 0: 无校准执行或校准结束; 1: 开始校准或校准进行中。
位 1	RPEN	0x0	rw	反复模式使能 (Repeat mode enable) 0: 关闭 SQEN=0 时, 每次触发转换单个通道, SQEN=1 时, 每次触发转换一组通道; 1: 开启 SQEN =0 时, 一次触发后将反复转换单个通道, SQEN =1 时, 一次触发后将反复转换一组通道。直到 ADCEN 被清零。
位 0	ADCEN	0x0	rw	A/D 转换器使能 (A/D converter enable) 0: 关闭 (ADC 进入断电模式); 1: 开启。 注: 当该位为关闭状态时, 写入开启命令将把 ADC 从断电模式下唤醒。 当该位为开启状态时, 再写入开启命令时同寄存器其它位未改变, 则重复该开启命令将启动普通通道组的转换。 应用程序需注意, 在转换器上电至转换开始有一个延迟 tSTAB。

### 18.5.4 ADC采样时间寄存器1 (ADC\_SPT1)

访问：字访问

域	简称	复位值	类型	功能
位 31: 24	保留	0x00	resd	请保持默认值。
位 23: 21	CSPT17	0x0	rw	<p>选择 ADC_IN17 通道的采样时间 (Selection sample time of channel ADC_IN17) 000: 1.5 周期; 001: 7.5 周期; 010: 13.5 周期; 011: 28.5 周期; 100: 41.5 周期; 101: 55.5 周期; 110: 71.5 周期; 111: 239.5 周期。</p>
位 20: 18	CSPT16	0x0	rw	<p>选择 ADC_IN16 通道的采样时间 (Selection sample time of channel ADC_IN16) 000: 1.5 周期; 001: 7.5 周期; 010: 13.5 周期; 011: 28.5 周期; 100: 41.5 周期; 101: 55.5 周期; 110: 71.5 周期; 111: 239.5 周期。</p>
位 17: 15	CSPT15	0x0	rw	<p>选择 ADC_IN15 通道的采样时间 (Selection sample time of channel ADC_IN15) 000: 1.5 周期; 001: 7.5 周期; 010: 13.5 周期; 011: 28.5 周期; 100: 41.5 周期; 101: 55.5 周期; 110: 71.5 周期; 111: 239.5 周期。</p>
位 14: 12	CSPT14	0x0	rw	<p>选择 ADC_IN14 通道的采样时间 (Selection sample time of channel ADC_IN14) 000: 1.5 周期; 001: 7.5 周期; 010: 13.5 周期; 011: 28.5 周期; 100: 41.5 周期; 101: 55.5 周期; 110: 71.5 周期; 111: 239.5 周期。</p>
位 11: 9	CSPT13	0x0	rw	<p>选择 ADC_IN13 通道的采样时间 (Selection sample time of channel ADC_IN13) 000: 1.5 周期; 001: 7.5 周期; 010: 13.5 周期; 011: 28.5 周期; 100: 41.5 周期; 101: 55.5 周期; 110: 71.5 周期; 111: 239.5 周期。</p>

---

位 8: 6	CSPT12	0x0	rw	选择 ADC_IN12 通道的采样时间 (Selection sample time of channel ADC_IN12) 000: 1.5 周期; 001: 7.5 周期; 010: 13.5 周期; 011: 28.5 周期; 100: 41.5 周期; 101: 55.5 周期; 110: 71.5 周期; 111: 239.5 周期。
位 5: 3	CSPT11	0x0	rw	选择 ADC_IN11 通道的采样时间 (Selection sample time of channel ADC_IN11) 000: 1.5 周期; 001: 7.5 周期; 010: 13.5 周期; 011: 28.5 周期; 100: 41.5 周期; 101: 55.5 周期; 110: 71.5 周期; 111: 239.5 周期。
位 2: 0	CSPT10	0x0	rw	选择 ADC_IN10 通道的采样时间 (Selection sample time of channel ADC_IN10) 000: 1.5 周期; 001: 7.5 周期; 010: 13.5 周期; 011: 28.5 周期; 100: 41.5 周期; 101: 55.5 周期; 110: 71.5 周期; 111: 239.5 周期。

---

### 18.5.5 ADC采样时间寄存器2 (ADC\_SPT2)

访问：字访问

域	简称	复位值	类型	功能
位 31: 30	保留	0x0	resd	请保持默认值。
位 29: 27	CSPT9	0x0	rw	选择 ADC_IN9 通道的采样时间 (Selection sample time of channel ADC_IN9) 000: 1.5 周期; 001: 7.5 周期; 010: 13.5 周期; 011: 28.5 周期; 100: 41.5 周期; 101: 55.5 周期; 110: 71.5 周期; 111: 239.5 周期。

---

位 26: 24 CSPT8	0x0	rw	选择 ADC_IN8 通道的采样时间 (Selection sample time of channel ADC_IN8) 000: 1.5 周期; 001: 7.5 周期; 010: 13.5 周期; 011: 28.5 周期; 100: 41.5 周期; 101: 55.5 周期; 110: 71.5 周期; 111: 239.5 周期。
位 23: 21 CSPT7	0x0	rw	选择 ADC_IN7 通道的采样时间 (Selection sample time of channel ADC_IN7) 000: 1.5 周期; 001: 7.5 周期; 010: 13.5 周期; 011: 28.5 周期; 100: 41.5 周期; 101: 55.5 周期; 110: 71.5 周期; 111: 239.5 周期。
位 20: 18 CSPT6	0x0	rw	选择 ADC_IN6 通道的采样时间 (Selection sample time of channel ADC_IN6) 000: 1.5 周期; 001: 7.5 周期; 010: 13.5 周期; 011: 28.5 周期; 100: 41.5 周期; 101: 55.5 周期; 110: 71.5 周期; 111: 239.5 周期。
位 17: 15 CSPT5	0x0	rw	选择 ADC_IN5 通道的采样时间 (Selection sample time of channel ADC_IN5) 000: 1.5 周期; 001: 7.5 周期; 010: 13.5 周期; 011: 28.5 周期; 100: 41.5 周期; 101: 55.5 周期; 110: 71.5 周期; 111: 239.5 周期。
位 14: 12 CSPT4	0x0	rw	选择 ADC_IN4 通道的采样时间 (Selection sample time of channel ADC_IN4) 000: 1.5 周期; 001: 7.5 周期; 010: 13.5 周期; 011: 28.5 周期; 100: 41.5 周期; 101: 55.5 周期; 110: 71.5 周期; 111: 239.5 周期。

位 11: 9	CSPT3	0x0	rw	选择 ADC_IN3 通道的采样时间（Selection sample time of channel ADC_IN3） 000: 1.5 周期; 001: 7.5 周期; 010: 13.5 周期; 011: 28.5 周期; 100: 41.5 周期; 101: 55.5 周期; 110: 71.5 周期; 111: 239.5 周期。
位 8: 6	CSPT2	0x0	rw	选择 ADC_IN2 通道的采样时间（Selection sample time of channel ADC_IN2） 000: 1.5 周期; 001: 7.5 周期; 010: 13.5 周期; 011: 28.5 周期; 100: 41.5 周期; 101: 55.5 周期; 110: 71.5 周期; 111: 239.5 周期。
位 5: 3	CSPT1	0x0	rw	选择 ADC_IN1 通道的采样时间（Selection sample time of channel ADC_IN1） 000: 1.5 周期; 001: 7.5 周期; 010: 13.5 周期; 011: 28.5 周期; 100: 41.5 周期; 101: 55.5 周期; 110: 71.5 周期; 111: 239.5 周期。
位 2: 0	CSPT0	0x0	rw	选择 ADC_IN0 通道的采样时间（Selection sample time of channel ADC_IN0） 000: 1.5 周期; 001: 7.5 周期; 010: 13.5 周期; 011: 28.5 周期; 100: 41.5 周期; 101: 55.5 周期; 110: 71.5 周期; 111: 239.5 周期。

### 18.5.6 ADC抢占通道数据偏移寄存器x (ADC\_PCDTOx) (x=1..4)

访问：字访问

域	简称	复位值	类型	功能
位 31: 12	保留	0x00000	resd	请保持默认值。
位 11: 0	PCDTox	0x000	rw	抢占通道 x 的数据偏移量设定（Data offset for Preempted channel x） ADC_PDTx 内存放的转换数据 = 原始转换数据 - ADC_PCDTOx

### 18.5.7 ADC电压监测高边界寄存器 (ADC\_VMHB)

访问：字访问

域	简称	复位值	类型	功能
位 31: 12	保留	0x00000	resd	请保持默认值。
位 11: 0	VMHB	0xFFFF	rw	电压监测高边界设定 (Voltage monitoring high boundary)

### 18.5.8 ADC电压监测低边界寄存器 (ADC\_VMLB)

访问：字访问

域	简称	复位值	类型	功能
位 31: 12	保留	0x00000	resd	请保持默认值。
位 11: 0	VMLB	0x000	rw	电压监测低边界设定 (Voltage monitoring low boundary)

### 18.5.9 ADC普通序列寄存器1 (ADC\_OSQ1)

访问：字访问

域	简称	复位值	类型	功能
位 31: 24	保留	0x00	resd	请保持默认值。
位 23: 20	OCLEN	0x0	rw	普通转换序列长度 (Ordinary conversion sequence length) 0000: 1 个转换; 0001: 2 个转换; ..... 1111: 16 个转换。
位 19: 15	OSN16	0x00	rw	普通序列中第 16 个转换通道的编号 (number of 16th conversion in ordinary sequence)
位 14: 10	OSN15	0x00	rw	普通序列中第 15 个转换通道的编号 (number of 15th conversion in ordinary sequence)
位 9: 5	OSN14	0x00	rw	普通序列中第 14 个转换通道的编号 (number of 14th conversion in ordinary sequence)
位 4: 0	OSN13	0x00	rw	普通序列中第 13 个转换通道的编号 (number of 13th conversion in ordinary sequence) 注：编号可设定 0~17，示例：设定为 3 就代表第 13 个转换的是 ADC_IN3 通道。

### 18.5.10 ADC普通序列寄存器2 (ADC\_OSQ2)

访问：字访问

域	简称	复位值	类型	功能
位 31: 30	保留	0x0	resd	请保持默认值。
位 29: 25	OSN12	0x00	rw	普通序列中第 12 个转换通道的编号 (number of 12th conversion in ordinary sequence)
位 24: 20	OSN11	0x00	rw	普通序列中第 11 个转换通道的编号 (number of 11th conversion in ordinary sequence)
位 19: 15	OSN10	0x00	rw	普通序列中第 10 个转换通道的编号 (number of 10th conversion in ordinary sequence)

位 14: 10 OSN9	0x00	rw	普通序列中第 9 个转换通道的编号 (number of 9th conversion in ordinary sequence)
位 9: 5 OSN8	0x00	rw	普通序列中第 8 个转换通道的编号 (number of 8th conversion in ordinary sequence)
位 4: 0 OSN7	0x00	rw	普通序列中第 7 个转换通道的编号 (number of 7th conversion in ordinary sequence) 注：编号可设定 0~17，示例：设定为 8 就代表第 7 个转换的是 ADC_IN8 通道。

### 18.5.11 ADC 普通序列寄存器 3 (ADC\_OSQ3)

访问：字访问

域	简称	复位值	类型	功能
位 31: 30 保留		0x0	resd	请保持默认值。
位 29: 25 OSN6		0x00	rw	普通序列中第 6 个转换通道的编号 (number of 6th conversion in ordinary sequence)
位 24: 20 OSN5		0x00	rw	普通序列中第 5 个转换通道的编号 (number of 5th conversion in ordinary sequence)
位 19: 15 OSN4		0x00	rw	普通序列中第 4 个转换通道的编号 (number of 4th conversion in ordinary sequence)
位 14: 10 OSN3		0x00	rw	普通序列中第 3 个转换通道的编号 (number of 3rd conversion in ordinary sequence)
位 9: 5 OSN2		0x00	rw	普通序列中第 2 个转换通道的编号 (number of 2nd conversion in ordinary sequence)
位 4: 0 OSN1		0x00	rw	普通序列中第 1 个转换通道的编号 (number of 1st conversion in ordinary sequence) 注：编号可设定 0~17，示例：设定为 17 就代表第 1 个转换的是 ADC_IN17 通道。

### 18.5.12 ADC 抢占序列寄存器 (ADC\_PSQ)

访问：字访问

域	简称	复位值	类型	功能
位 31: 30 保留		0x0	resd	请保持默认值。
位 21: 20 PCLEN		0x0	rw	抢占转换序列长度 (Preempted conversion sequence length) 00: 1 个转换; 01: 2 个转换; 10: 3 个转换; 11: 4 个转换。
位 19: 15 PSN4		0x00	rw	抢占序列中第 4 个转换通道的编号 (number of 4th conversion in Preempted sequence)
位 14: 10 PSN3		0x00	rw	抢占序列中第 3 个转换通道的编号 (number of 3rd conversion in Preempted sequence)
位 9: 5 PSN2		0x00	rw	抢占序列中第 2 个转换通道的编号 (number of 2nd conversion in Preempted sequence)

位 4: 0 PSN1 0x00

rw

抢占序列中第 1 个转换通道的编号 (number of 1st conversion in Preempted sequence)

注:

编号可设定 0~17, 比如设定为 3 时其代表的就是 ADC\_IN3 通道。

若 PCLEN 小于 4, 则转换的序列顺序是从 (4-PCLEN)

开始。例如: ADC\_PSQ[21: 0] = 10 00110 00101

00100 00011, 意味着扫描转换将按下列通道顺序执行:

4、5、6, 而不是 3、4、5。

### 18.5.13 ADC 抢占数据寄存器 x (ADC\_PDTx) (x=1..4)

访问: 字访问

域	简称	复位值	类型	功能
位 31: 16	保留	0x0000	resd	请保持默认值。
位 15: 0	PDTx	0x0000	ro	抢占通道的转换数据 (Conversion data of preempted channel)

### 18.5.14 ADC 普通数据寄存器 (ADC\_ODT)

访问: 字访问

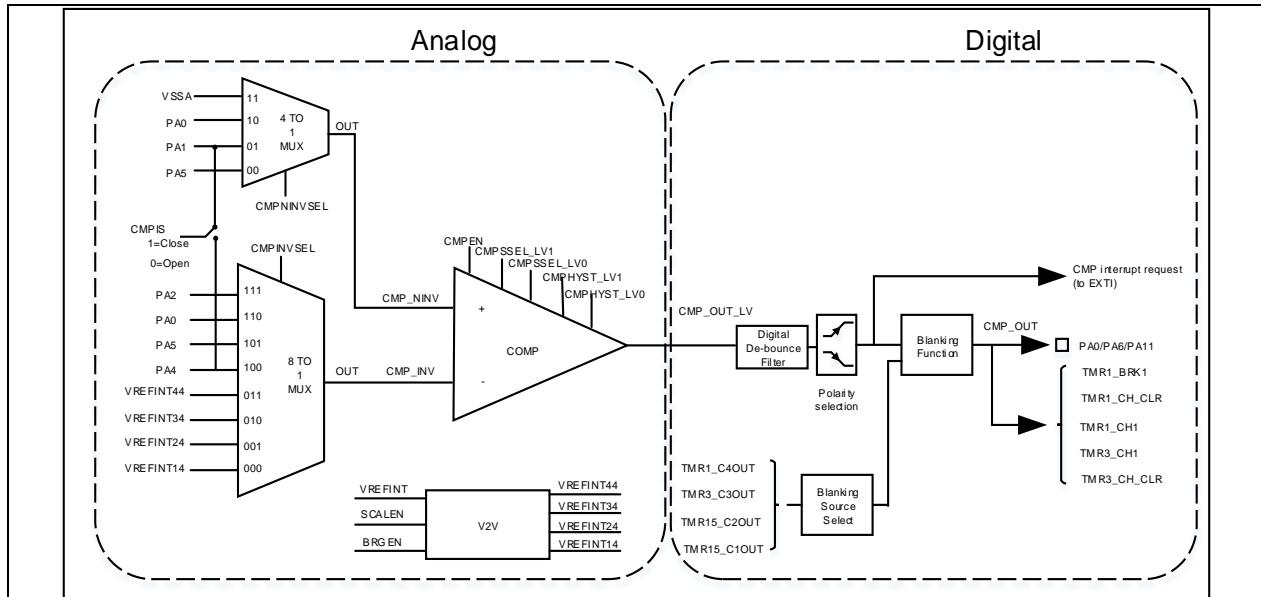
域	简称	复位值	类型	功能
位 31: 16	保留	0x0000	resd	请保持默认值。
位 15: 0	ODT	0x0000	ro	普通通道的转换数据 (Conversion data of ordinary channel)

# 19 比较器 (CMP)

## 19.1 简介

AT32F421 内置一个超低功耗比较器 CMP，可以用于多种功能，包括：外部模拟信号的监测控制及从低功耗模式唤醒，与内置定时器结合使用，进行脉冲宽度测量和 PWM 信号控制等。

图 19-1 比较器的框图



## 19.2 主要特性

- 比较器迟滞程度可配
- 定时器输出作为比较器消隐源
- 比较器输出极性可配
- 比较器输出速度可配
- 比较器同相和反相输入源可选：
  - I/O 引脚
  - 内部参考电压和三个系数分压值 (1/4, 1/2, 3/4)
- 支持输出重定向功能：
  - 普通 I/O
  - 定时器断路输入 TMRx\_BRK
  - 定时器输入捕获 TMR\_CH
  - 定时器输出比较参考值清零 TMR\_CH\_CLR
- 结合 EXINT 产生中断，从低功耗模式唤醒

## 19.3 中断管理

模拟比较器的输出经过数字滤波器滤波以及极性选择，输入至 EXINT 第 21 号中断线，产生外部中断或者事件，可以用于系统从低功耗模式中唤醒。

更多详细信息，请参见中断和事件部分。

## 19.4 设计提示

比较器有以下提示仅供设计参考。

- 输入输出配置
  - I/Os 作为比较器输入时，必须配置为模拟模式。比较器输出可以重映射到外部 I/O 口。
  - 比较器输出配置

复用配置	<b>GPIOA_MUXL[3: 0]=0111</b>	<b>GPIOA_MUXL[27: 24]=0111</b>	<b>GPIOA_MUXH[15: 12]=0111</b>
输出端口	PA0	PA6	PA11

- 锁定功能

比较器控制和状态寄存器 **CMP\_CTRLSTS** 具有写保护功能，一旦编程完成，对 **CMPWP** 位设置为 1，则整个 **CMP\_CTRLSTS** 寄存器变为只读，包括 **CMPWP** 位，只能通过系统复位解除写保护功能，该功能可用于具有特定功能安全要求的应用。

- 低功耗模式配置

比较器的时钟使能控制与 **SCFG** 共用，时钟源为 **PCLK**，复位信号采用系统复位。在深度睡眠模式下，比较器依然可以工作，用于 **EXINT** 的中断源，控制系统从低功耗模式中唤醒，但在进入深度睡眠模式之前，需要关闭比较器的数字滤波功能（**G\_FILTER\_EN** 寄存器中的 **GFE=0**）。

## 19.5 功能描述

### 19.5.1 模拟比较器

#### 同相反相输入选择

寄存器 **CMP\_CTRLSTS** 的 **CMPINVSEL[1: 0]** 位控制比较器的同相输入信号来源于 I/O 口或者 **VSSA**，**CMPINVSEL[2: 0]** 位控制比较器的反相输入信号来源于内部参考电压和三个系数分压值或者 I/O 口。系数分压值的使能由 **SCALEN** 位和 **BRGEN** 位共同控制。

#### 迟滞功能

寄存器 **CMP\_CTRLSTS** 的 **CMPHYST[1: 0]** 控制比较器迟滞输出，该功能可避开噪声信号带来的虚假传输信号，如果不需要迟滞，可以关闭掉。

#### 操作模式配置

寄存器 **CMP\_CTRLSTS** 的 **CMPSEL[1: 0]** 位用于控制比较器工作在高速/最大功耗、中速/中功耗、低速/低功耗、超低速/超低功耗，用于实现比较器在性能与功耗之间的折中。

#### 输出消隐功能

寄存器 **CMP\_CTRLSTS** 的 **CMPBLANKING[2: 0]** 位用于选择比较器消隐窗口的来源，该功能可以用于防止电流调节在 PWM 起始时刻产生的尖峰电流。

### 19.5.2 干扰滤波器

干扰滤波器可以用来滤除毛刺和噪声干扰。

滤波器的敏感性由 **H\_PULSE\_CNT** 和 **L\_PULSE\_CNT** 位控制。滤波器的敏感性会影响相同的连续采样的数量，在滤波器输入上检测到此类连续采样时，才能将某信号电平变化视为有效切换。

下图显示了在不同 **H\_PULSE\_CNT** 和 **L\_PULSE\_CNT = 0** 时干扰滤波器时序图。

图 19-2 **H\_PULSE\_CNT=1** 和 **L\_PULSE\_CNT = 0** 时干扰滤波器时序图

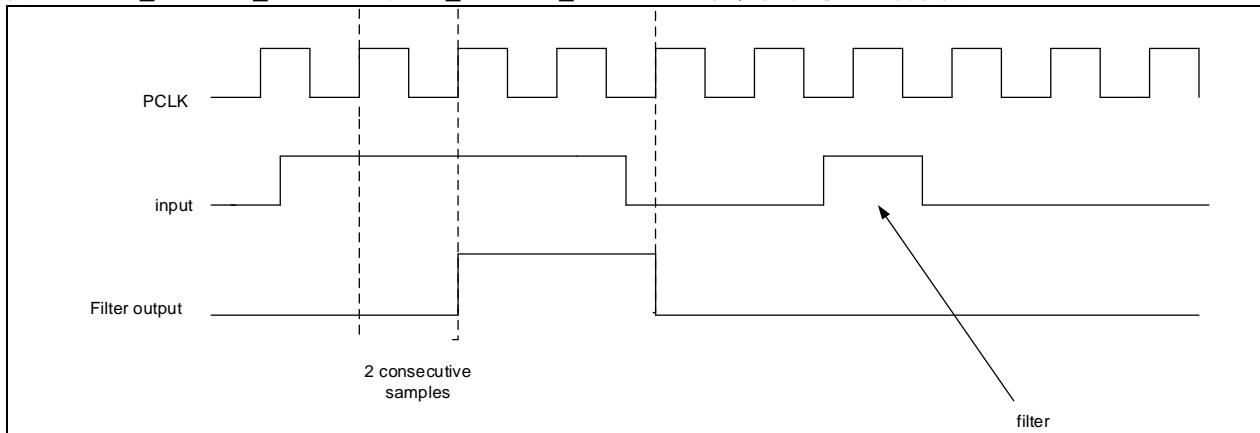
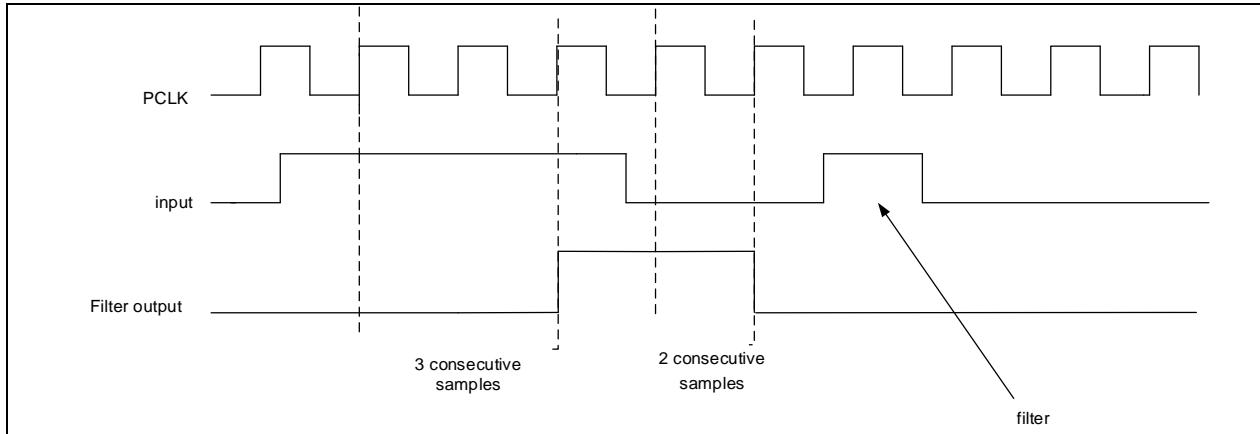


图 19-3 H\_PULSE\_CNT=2 和 L\_PULSE\_CNT = 1 时干扰滤波器时序图



## 19.6 CMP寄存器

必须以字（32位）的方式操作这些外设寄存器。

表 19-1 CMP寄存器图和复位值

寄存器简称	基址偏移量	复位值
CMP_CTRLSTS	0x1C	0x0000 0080
G_FILTER_EN	0x24	0x0000
HIGH_PULSE	0x28	0x0000
LOW_PULSE	0x2C	0x0000

### 19.6.1 比较器控制和状态寄存器1(CMP\_CTRLSTS)

域	简称	复位值	类型	功能
位 31	CMPWP	0x0	rw0c	比较器写保护(Comparator write protect) 该位只能写一次，由软件置 1，由系统复位清零。它将锁定比较器控制寄存器 CMP_CTRLSTS[31:0]的全部内容 0: 比较器的 CMP_CTRLSTS[31:0]可读/可写 1: 比较器的 CMP_CTRLSTS[31:0]只读
位 30	CMPVALUE	0x0	ro	比较器输出值(Comparator output value) 此位为只读。
位 29:24	保留	0x00	resd	请保持默认值。
位 23	SCALEN	0x0	rw	内部等分电压使能(scale voltage enable) 0: 内部等分电压不使能(VREFINT44 = VREFINT34 = VREFINT24 = VREFINT14 = 0V) 1: 内部等分电压使能(需要 CMPEN = 1, CMPINVSEL[2] = 0)
位 22	BRGEN	0x0	rw	等分电压桥使能(voltage bridge enable) 0: 等分电压桥不使能, VREFINT44 = VREFINT34 = VREFINT24 = VREFINT14 = 1.2V (需要 SCALEN = 1, CMPEN = 1, CMPINVSEL[2] = 0) 1: 等分电压桥使能, VREFINT44 = 1.2V, VREFINT34 = 0.9V, VREFINT24 = 0.6V, VREFINT14 = 0.3V (需要 SCALEN = 1, CMPEN = 1, CMPINVSEL[2] = 0)
位 21	保留	0x0	resd	请保持默认值。
位 20:18	CMPBLANKING	0x0	rw	比较器消隐窗口源(Comparator blank source) 000: 没有输出消隐功能 001: TMR1 C4OUT 作为消隐窗口源 010: 保留 011: TMR3 C3OUT 作为消隐窗口源 100: TMR15 C2OUT 作为消隐窗口源 101: 保留 110: TMR15 C1OUT 作为消隐窗口源 111: 保留

				比较器迟滞(Comparator hysteresis)
位 17:16	CMPHYST	0x0	rw	00: 没有迟滞 01: 低度迟滞 10: 中度迟滞 11: 高度迟滞 参见迟滞程度的电气特性
位 15	CMPP	0x0	rw	比较器极性选择位(Comparator polarity) 0: 比较器输出值不反相 1: 比较器输出值反相
位 14:13	保留	0x0	resd	请保持默认值。
位 12:10	CMPTAG	0x0	rw	比较器输出选择位(Comparator output target) 这些位控制比较器的输出目的地 000: 无选择 001: 定时器 1 刹车输入 010: 定时器 1 输入捕获 1 011: 定时器 1 输出比较清除 100: 保留 101: 保留 110: 定时器 3 输入捕获 1 111: 定时器 3 输出比较清除
位 9	保留	0x0	resd	请保持默认值。
位 8:7	CMPINVSEL	0x1	rw	比较器同相输入选择(Comparator non-inverting input selection) 00: INP0(PA5) 01: INP1(PA1(默认输入)) 10: INP2(PA0) 11: VSSA
位 6:4	CMPINVSEL	0x0	rw	比较器的反相端输入选择(Comparator inverting selection) 000: 1/4 VREFINT 001: 1/2 VREFINT 010: 3/4 VREFINT 011: VREFINT 100: INM4(PA4) 101: INM5(PA5) 110: INM6(PA0) 111: INM7(PA2)
位 3:2	CMPSEL	0x0	rw	比较器速度选择(Comparator speed selection) 这些位用于控制比较器的操作模式，允许调整速率和功耗 00: 高速/最大功耗 01: 中速/中功耗 10: 低速/低功耗 11: 超低速/超低功耗
位 1	CMPIS	0x0	r	比较器输入切换( Comparator input shift) 0: 开关断开 1: 开关闭合 注：该位用于开关比较器同相输入端 PA1 与 PA4 之间的连接。此开关仅用于重定向信号到高阻输入，例如比较器的同相输入(高阻开关)
位 0	CMPEN	0x0	rw	比较器使能(Comparator enable) 该位控制打开或关闭比较器 0: 关闭比较器 1: 开启比较器

## 19.6.2 干扰滤波器使能寄存器(G\_FILTER\_EN)

域	简称	复位值	类型	功能
位 15: 1	保留	0x0000	resd	请保持默认值。
位 0	GFE	0x0	rw	干扰滤波器使能 (Glitch filter enable) 0: 无作用 1: 使能滤波器

### 19.6.3 干扰滤波器高脉冲数(HIGH\_PULSE)

域	简称	复位值	类型	功能
位 15:6	保留	0x000	resd	请保持默认值。 高脉冲计数值 (High pulse Count) 滤波器输入信号有效电平变化必须至少稳定 H_PULSE_CNT+1 个时钟周期，才能将其视为有效输入，输出才会改变为高电平。
位 5:0	H_PULSE_CNT	0x0	rw	0: 1 个 pclk 时钟 1: 2 个 pclk 时钟 2: 3 个 pclk 时钟 ..... 62: 63 个 pclk 时钟 63: 64 个 pclk 时钟

### 19.6.4 干扰滤波器低脉冲数(LOW\_PULSE)

域	简称	复位值	类型	功能
位 15:6	保留	0x000	resd	请保持默认值。 低脉冲计数值 (Low pulse Count) 滤波器输入信号无效电平变化必须至少稳定 L_PULSE_CNT+1 个时钟周期，才能将其视为有效输入，输出才会改变为低电平。
位 5:0	L_PULSE_CNT	0x00	rw	0: 1 个 pclk 时钟 1: 2 个 pclk 时钟 2: 3 个 pclk 时钟 ..... 62: 63 个 pclk 时钟 63: 64 个 pclk 时钟

## 20 运算放大器 (OP)

OP 适用于 AT32F4212 系列产品。

### 20.1 简介

内置两个轨到轨运算放大器，具有高达 6 MHz 带宽和低于 4.5 mV 的偏移误差电压。

### 20.2 主要特性

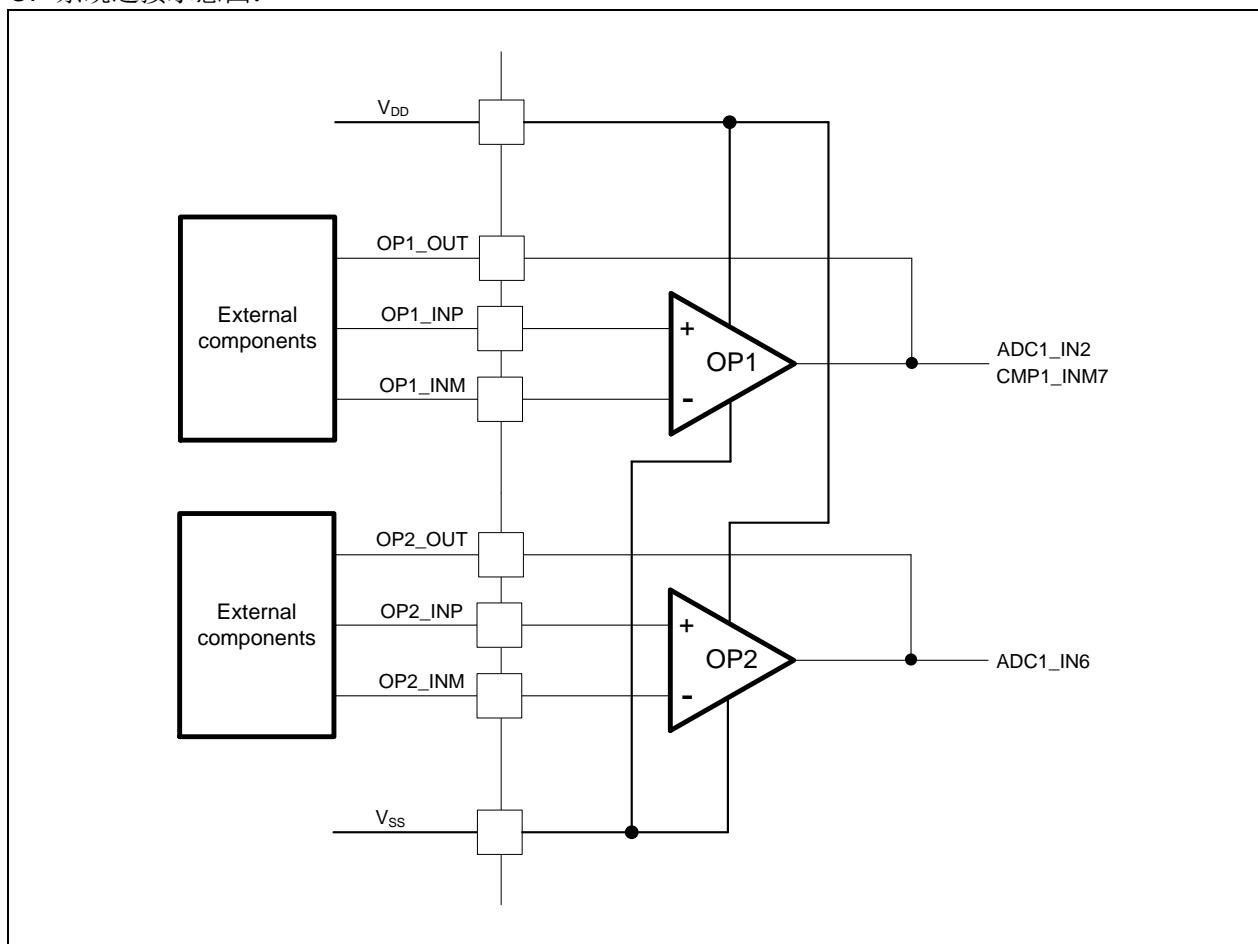
- 偏移电压：0.8mV（典型）
- 高增益：97dB（典型）
- 高增益带宽：6MHz
- 轨到轨的输入输出
- 供电电压范围：+2.4 ~ 3.6V
- 静止模式：3.3V 工作电压时 900uA（单个 OP）

### 20.3 功能描述

带 OP 系列产品，OP 自动开启无法关闭，因此与 OP 引脚复用的 GPIO 使用功能上会有一些限制。

OPx\_INy 引脚仍可作为 GPIO 或使用其上的复用功能；做为 OP 输入使用时，建议将该引脚设置为模拟模式。注意这些引脚输入电平必须小于 VDD+0.3V。建议使用者将 OPx\_OUT 引脚设置为模拟模式。该引脚随外部线路和 OP\_IN 引脚电平有各种可能输出电压。不可再当外部输入引脚使用；但该引脚内部与 ADC1\_INy 或 CMP1\_INy 连接，这些功能仍可使用。

OP 系统连接示意图：



芯片引脚和 OP 引脚对应位置：

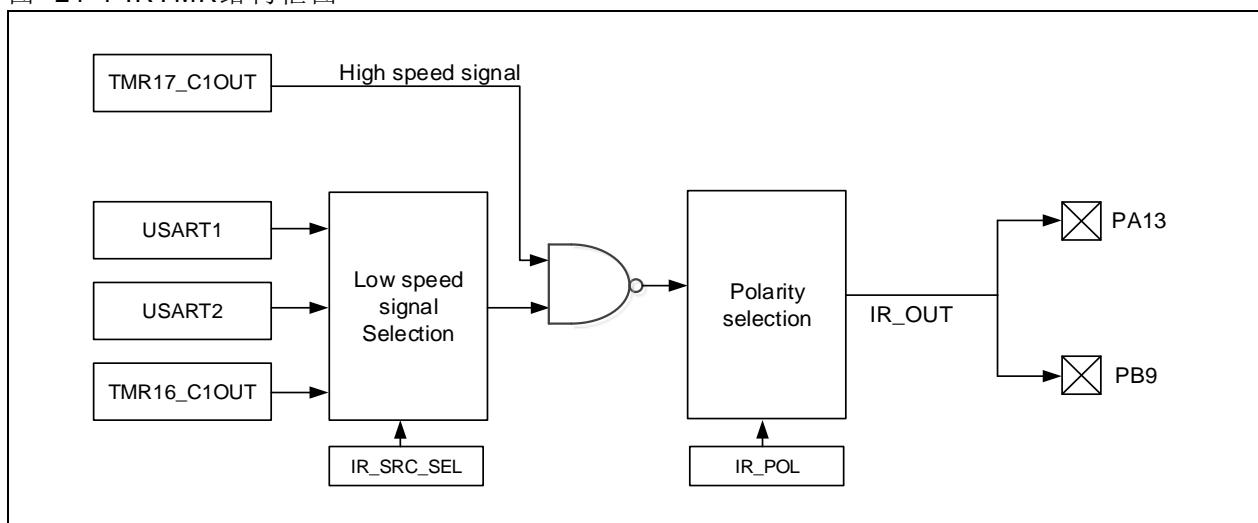
引脚名	OP 引脚
PA0	OP1_INP
PA1	OP1_INM
PA2	OP1_OUT
PA4	OP2_INP
PA5	OP2_INM
PA6	OP2_OUT

## 21 红外线接口 (IRTMR)

IRTMR 用于产生驱动红外 LED 的 IR\_OUT 信号，进而实现红外控制功能。

IR\_OUT 信号由低频调制包络信号和高频载波信号两部分组成，低频调制包络信号由 SCFG\_CFG1 寄存器 IR\_SRC\_SEL[1:0] 从 TMR16\_C1OUT、USART1、USART2 三者之中选择；高频载波信号由 TMR17\_C1OUT 提供，SCFG\_CFG1 寄存器中 IR\_POL 可控制输出的 IR\_OUT 信号是否反向。IR\_OUT 通过 PB9 或 PA13 通过复用功能输出（需提前配置为复用模式）。

图 21-1 IRTMR 结构框图



## 22 调试 (DEBUG)

### 22.1 简介

Cortex®-M4 内核具有丰富的调试特性。除了支持暂停和单步等标准的调试特性外，还可以利用跟踪特性查看程序执行的细节。Cortex®-M4 内核的调试可以通过串行调试接口。

ARM Cortex®-M4 内核相关资料，可参考：

- Cortex®-M4 技术参考手册(TRM)
- ARM 调试接口 V5
- ARM CoreSight 开发工具集(r1p0 版)技术参考手册

### 22.2 调试与跟踪功能

支持不同外设的调试，还可以设置调试时外设的工作状态。对于定时器和看门狗用户可以选择在调试时是否停止或继续计数；对于 I²C，用户可以选择在调试期间是否停止或继续 SMBUS 超时计数。

另外支持在低功耗模式下调试代码。在睡眠模式下，HCLK 与 FCLK 保持代码配置的时钟继续工作。在深度休眠模式下，HICK 振荡器将开启并为 FCLK 和 HCLK 提供时钟。

MCU 内部有多个 ID 编码，调试器可通过地址为 0xE0042000 的 DEBUG\_IDCODE 来访问。它是 DEBUG 的一个组成部分，并且映射到外部 PPB 总线上。使用 SW 调试口或通过用户代码都可以访问此编码。即使当 MCU 处于系统复位状态下这个编码也可以被访问。

### 22.3 I/O 控制

AT32F421 的 2 个普通 I/O 可用于 SW-DP 调试。系统复位以后，SW-DP 作为默认功能可立即供调试器使用。

当用户切换调试接口或不使用调试功能时，可配置 GPIO 寄存器来释放这些专用 I/O 口。用户释放相应的调试 I/O 后，GPIO 控制器将取得控制，这些 I/O 口可作为普通的 I/O 口使用。

### 22.4 DEBUG 寄存器

下面列出了 DEBUG 寄存器映象和复位数值。

必须以字（32 位）的方式操作这些外设寄存器。

表 22-1 DEBUG 寄存器地址和复位值

寄存器简称	基地址	复位值
DEBUG_IDCODE	0xE004 2000	0xFFFF XXXX
DEBUG_CTRL	0xE004 2004	0x0000 0000

#### 22.4.1 DEBUG 设备 ID (DEBUG\_IDCODE)

MCU 集成了 ID code，通过 ID 可以识别 MCU 的版本编号。DEBUG\_IDCODE 寄存器被映射到外部 PPB 总线，基址为 0xE0042000。使用 JTAG 调试口或 SW 调试口或用户代码都可以访问此编号。

域	简称	复位值	类型	功能
位 31: 0 PID		0xFFFF XXXX ro	PID 信息	
PID [31: 0]	AT32 型号	FLASH 大小	封装	
0x50020100	AT32F421C8T7	64KB	LQFP48	
0x50020101	AT32F421K8T7	64KB	LQFP32	
0x50020102	AT32F421K8U7	64KB	QFN32 (5x5)	
0x50020103	AT32F421K8U7-4	64KB	QFN32 (4x4)	
0x50020104	AT32F421F8U7	64KB	QFN20	
0x50020105	AT32F421F8P7	64KB	TSSOP20	
0x50020086	AT32F421C6T7	32KB	LQFP48	

0x50020087	AT32F421K6T7	32KB	LQFP32
0x50020088	AT32F421K6U7	32KB	QFN32 (5x5)
0x50020089	AT32F421K6U7-4	32KB	QFN32 (4x4)
0x5002008A	AT32F421F6U7	32KB	QFN20
0x5002008B	AT32F421F6P7	32KB	TSSOP20
0x5001000C	AT32F421C4T7	16KB	LQFP48
0x5001000D	AT32F421K4T7	16KB	LQFP32
0x5001000E	AT32F421K4U7	16KB	QFN32 (5x5)
0x5001000F	AT32F421K4U7-4	16KB	QFN32 (4x4)
0x50010010	AT32F421F4U7	16KB	QFN20
0x50010011	AT32F421F4P7	16KB	TSSOP20
0x50020112	AT32F421G8U7	64KB	QFN28
0x50020093	AT32F421G6U7	32KB	QFN28
0x50010014	AT32F421G4U7	16KB	QFN28

## 22.4.2 DEBUG控制寄存器 (DEBUG\_CTRL)

寄存器由 PORESET 异步复位 (不被系统复位所复位)。当内核处于复位状态下时，调试器可写。

域	简称	复位值	类型	功能
位 31: 28	保留	0x0000	resd	请保持默认值。
位 27	TMR14_PAUSE	0x0	rw	TMR14 暂停控制位 0: 定时器正常工作； 1: 定时器停止工作。
位 26: 25	保留	0x0	resd	请保持默认值。
位 24	TMR17_PAUSE	0x0	resd	TMR17 暂停控制位 0: 定时器正常工作； 1: 定时器停止工作。
位 23	TMR16_PAUSE	0x0	resd	TMR16 暂停控制位 0: 定时器正常工作； 1: 定时器停止工作。
位 22	TMR15_PAUSE	0x0	resd	TMR15 暂停控制位 0: 定时器正常工作； 1: 定时器停止工作。
位 21	ERTC_512_PAUSE	0x0	rw	ERTC512Hz 暂停控制位。 0: ERTC512Hz 时正常运行； 1: ERTC512HZ 时的接收寄存器不继续接收数据。
位 20	保留	0x0	resd	请保持默认值。
位 19	TMR6_PAUSE	0x0	rw	TMR6 暂停控制位 0: 定时器正常工作； 1: 定时器停止工作。
位 18: 17	保留	0x0	resd	请保持默认值。
位 16	I2C2_SMBUS_TIMEOUT	0x0	rw	I2C2 暂停控制位。 0: 正常工作； 1: I2C2 SMBUS 的超时控制停止工作。
位 15	I2C1_SMBUS_TIMEOUT	0x0	rw	I2C1 暂停控制位。 0: 正常工作； 1: I2C1 SMBUS 的超时控制停止工作。
位 14	ERTC_PAUSE	0x0	rw	ERTC 暂停控制位。 0: ERTC 正常运行； 1: ERTC 的接收寄存器不继续接收数据。

位 13	保留	0x0	resd	请保持默认值。
位 12	TMR3_PAUSE	0x0	rw	TMR3 暂停控制位。 0: 定时器正常工作; 1: 定时器停止工作。
位 11	保留	0x0	resd	请保持默认值。
位 10	TMR1_PAUSE	0x0	rw	TMR1 暂停控制位。 0: 定时器正常工作; 1: 定时器停止工作。
位 9	WWDT_PAUSE	0x0	rw	窗口看门狗暂停控制位。 0: 窗口看门狗正常工作; 1: 窗口看门狗停止工作。
位 8	WDT_PAUSE	0x0	rw	看门狗暂停控制位 0: 看门狗正常工作; 1: 看门狗停止工作。
位 7: 3	保留	0x00	resd	请保持默认值。
位 2	STANDBY_DEBUG	0x0	rw	待机模式调试控制位。 0: 进入待机模式时, 整个 1.2V 数字电路部分都断电; 1: 进入待机模式时, 整个 1.2V 数字电路部分不断电, 系统时钟由内部 RC 振荡器 (HICK) 提供时钟。
位 1	DEEPSLEEP_DEBUG	0x0	rw	深度睡眠模式调试控制位。 0: 进入深度睡眠模式时, 关闭所有 1.2V 域的时钟, 退出深度睡眠模式时, 系统时钟选择开启内部 RC 振荡器 (HICK), 系统时钟选择 HICK 作为系统时钟源, 软件需根据应用需求重新配置系统时钟; 1: 进入深度睡眠模式时, 系统时钟由内部 RC 振荡器 (HICK) 提供。退出深度睡眠模式时, 系统时钟选择 HICK 作为系统时钟源, 软件需根据应用需求重新配置系统时钟。
位 0	SLEEP_DEBUG	0x0	rw	睡眠模式调试控制位 0: 进入睡眠模式时, CPU HCLK 时钟关闭, 其他时钟均继续运行, 退出睡眠模式时, 不需要重新配置时钟系统; 1: 进入睡眠模式时, 所有时钟都继续运行。

## 23 版本历史

日期	版本	变更
2021.11.17	2.00	新版本发布
2022.06.27	2.01	1、修订 1.1.5 章节描述 2、修订 3.6 章节描述 3、修订 4.3.2 章节描述 4、修订 11.7.1 章节描述 5、修订 12.11.5 章节描述 6、修订 13 章，增加 SPI 协议时序图 7、修订 18.5.3 章节描述
2022.11.11	2.02	1、修订 5.8.1 章节描述 2、修订 10 章节描述 3、修订 14 章节描述 4、修订 17.2 章节描述
2023.08.02	2.03	1、修订 4.1.1 章节描述 2、修订 12.8.3 章节描述 3、修订 14 章节部分描述 4、修订 19.6.1 章节描述

**重要通知 - 请仔细阅读**

买方自行负责对本文所述雅特力产品和服务的选择和使用，雅特力概不承担与选择或使用本文所述雅特力产品和服务相关的任何责任。

无论之前是否有过任何形式的表示，本文档不以任何方式对任何知识产权进行任何明示或默示的授权或许可。如果本文档任何部分涉及任何第三方产品或服务，不应被视为雅特力授权使用此类第三方产品或服务，或许可其中的任何知识产权，或者被视为涉及以任何方式使用任何此类第三方产品或服务或其中任何知识产权的保证。

除非在雅特力的销售条款中另有说明，否则，雅特力对雅特力产品的使用和 / 或销售不做任何明示或默示的保证，包括但不限于有关适销性、适合特定用途（及其依据任何司法管辖区的法律的对应情况），或侵犯任何专利、版权或其他知识产权的默示保证。

雅特力产品并非设计或专门用于下列用途的产品：(A) 对安全性有特别要求的应用，例如：生命支持、主动植入设备或对产品功能安全有要求的系统；(B) 航空应用；(C) 汽车应用或汽车环境；(D) 航天应用或航天环境，且/或(E) 武器。如果雅特力产品不是为前述应用设计的，而采购商擅自将其用于前述应用，即使采购商向雅特力发出了书面通知，采购商仍将独自承担因此而导致的任何风险，雅特力的产品设计规格明确指定的汽车、汽车安全或医疗工业领域专用产品除外。根据相关政府主管部门的规定，ESCC、QML 或 JAN 正式认证产品适用于航天应用。

经销的雅特力产品如有不同于本文档中提出的声明和 / 或技术特点的规定，将立即导致雅特力针对本文所述雅特力产品或服务授予的任何保证失效，并且不应以任何形式造成或扩大雅特力的任何责任。

© 2019 雅特力科技 (重庆) 有限公司 保留所有权利