

基于 ARM® 32 位 Cortex®-M4 微控制器+FPU，配有 64 K 字节到 128 K 字节闪存、sLib、11 个定时器、8 个通信接口、2 个 ADC、2 个 DAC、2 个比较器、4 个运算放大器

■ 内核：带有FPU的ARM@32位的Cortex™-M4F CPU

- 最高 180 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
- 内建浮点运算单元（FPU）
- 具有 DSP 指令集

■ 存储器

- 64 K 字节到 128 K 字节的闪存存储器
- 28 K 字节的启动程序代码区作启动加载程序（Bootloader）用，可一次性配置成一般用户区
- 1 K 字节的 OTP 存储器
- 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
- 16 K 字节的 SRAM

■ 电源控制（PWC）

- 2.4V 至 3.6V 供电
- 上电复位（POR）、低电压复位（LVR）、电源电压监测器（PVM）
- 低功耗模式：睡眠、深睡眠、和待机，4 个 WKUP 引脚可唤醒待机模式
- 支持 20 个 32 位的电池供电寄存器（BPR）

■ 时钟和复位管理（CRM）

- 4 至 25 MHz 晶体振荡器（HEXT）
- 内置经出厂调校的 48 MHz 高速时钟（HICK），25 °C 达 1 % 精度，-40 °C 至 +105 °C 达 2.5 % 精度
- 32 kHz 晶振（LEXT）
- 低速内部时钟（LICK）

■ 模拟模块

- 2 个 12 位 2.5 MSPS A/D 转换器，多达 18 个外部输入通道，内部通道与 OP 互连；每通道独立数据寄存器
- 温度传感器（VTS）和内部参考电压（VINTRV）
- 2 个 6 位 D/A 转换器与 CMP 和 OP 互连
- 2 个 比较器（CMP）带输出消隐功能
- 4 个 运算放大器（OP），带宽 8.5 MHz，支持 PGA 模式，两输入端内置钳位二极管

■ DMA

- 1 个 7 通道 DMA 控制器，支持完全弹性映射

■ 多达46个快速GPIO端口

- 所有 GPIO 口可以映像到 16 个外部中断（EXTINT）
- 几乎所有 GPIO 口可容忍 5V 输入信号

■ 多达11个定时器（TMR）

- 1 个 16 位 8 通道高级定时器，包括 4 对互补通道 PWM 输出，带死区控制和紧急停止功能
- 多达 5 个 16 位通用定时器，每个定时器最多达 4 个用于输入捕获/输出比较/PWM 或脉冲计数的通道和增量编码器输入
- 2 个 16 位基本定时器
- 2 个 看门狗定时器（WDT 和 WWDT）
- 系统滴答定时器：24 位递减计数器

■ ERTC：增强型RTC，具有自动唤醒、闹钟、亚秒级精度及硬件日历，带校准功能

■ 多达8个通信接口

- 2 个 I²C 接口，支持 SMBus/PMBus
- 2 个 USART 接口；支持主同步 SPI 和调制解调器控制；具有 ISO7816 接口、LIN、IrDA、和 RS485 驱动使能；支持 TX/RX 可配置引脚互换
- 2 个 SPI 接口（36 M 位/秒），均可复用为半双工 I²S 接口
- 1 个 CAN 接口，内置 1408 字节的专用缓存（其中 AT32M416 支持 CAN FD 协议）
- 红外发射器（IRTMR）

■ CRC计算单元

■ 96位的芯片唯一代码（UID）

■ 串行线调试（SWD）

■ 温度范围：-40至+105 °C

■ 封装

- LQFP48 7 x 7 mm
- LQFP32 7 x 7 mm
- QFN32 4 x 4 mm
- TSSOP24 7.8 x 4.4 mm

■ 选型列表

AT32M412 选型列表

闪存存储器	型号
128 K 字节	AT32M412CBT7 , AT32M412KBT7 , AT32M412KBU7-4, AT32M412EBP7
64 K 字节	AT32M412C8T7 , AT32M412K8T7 , AT32M412K8U7-4, AT32M412E8P7

AT32M416 选型列表

闪存存储器	型号
128 K 字节	AT32M416CBT7 , AT32M416KBT7 , AT32M416KBU7-4, AT32M416EBP7
64 K 字节	AT32M416C8T7 , AT32M416K8T7 , AT32M416K8U7-4, AT32M416E8P7

目 录

1 系统架构	31
1.1 系统概述	32
1.1.1 ARM Cortex™-M4F处理器	32
1.1.2 位带	32
1.1.3 中断和异常向量	35
1.1.4 系统嘀嗒定时器 (SysTick)	38
1.1.5 复位流程	38
1.2 寄存器描述缩写说明	39
1.3 器件特征信息	40
1.3.1 闪存容量寄存器	40
1.3.2 器件电子签名	40
2 存储器资源	41
2.1 内部存储器地址映射	41
2.2 Flash存储器	41
2.3 SRAM存储器	42
2.4 外设地址映射	42
3 电源控制 (PWC)	45
3.1 简介	45
3.2 主要特点	45
3.3 上电下电复位	46
3.4 电压监测器 (PVM)	46
3.5 电源域划分	47
3.6 省电模式	47
3.7 PWC寄存器	49
3.7.1 电源控制寄存器 (PWC_CTRL)	49
3.7.2 电源控制及状态寄存器 (PWC_CTRLSTS)	50
3.7.3 电源标志清除寄存器 (PWC_CLR)	52
3.7.4 LDO调校寄存器 (PWC_LDOOV)	53
4 时钟和复位管理 (CRM)	54
4.1 时钟	54

4.1.1	时钟源	54
4.1.2	系统时钟	55
4.1.3	外设时钟	55
4.1.4	时钟失效检测	55
4.1.5	自动滑顺频率切换	56
4.1.6	内部时钟输出	56
4.1.7	中断	56
4.2	复位	56
4.2.1	系统复位	56
4.2.2	电池供电域复位	56
4.3	CRM寄存器描述	57
4.3.1	时钟控制寄存器 (CRM_CTRL)	57
4.3.2	PLL时钟配置寄存器 (CRM_PLLCFG)	59
4.3.3	时钟配置寄存器 (CRM_CFG)	60
4.3.4	时钟中断寄存器 (CRM_CLKINT)	61
4.3.5	AHB外设复位寄存器1 (CRM_AHBRST1)	62
4.3.6	APB1外设复位寄存器 (CRM_APB1RST)	63
4.3.7	APB2外设复位寄存器 (CRM_APB2RST)	64
4.3.8	AHB外设时钟使能寄存器1 (CRM_AHBEN1)	65
4.3.9	APB1外设时钟使能寄存器 (CRM_APB1EN)	65
4.3.10	APB2外设时钟使能寄存器 (CRM_APB2EN)	66
4.3.11	AHB外设时钟低功耗使能寄存器1 (CRM_AHBLPEN1)	67
4.3.12	APB1外设时钟低功耗使能寄存器 (CRM_APB1LPEN)	67
4.3.13	APB2外设时钟低功耗使能寄存器 (CRM_APB2LPEN)	69
4.3.14	外设独立时钟选择寄存器 (CRM_PICLKS)	69
4.3.15	电池供电域控制寄存器 (CRM_BPDC)	70
4.3.16	控制/状态寄存器 (CRM_CTRLSTS)	71
4.3.17	额外寄存器 (CRM_MISC1)	72
4.3.18	额外寄存器2 (CRM_MISC2)	73
5	闪存控制器 (FLASH)	74
5.1	FLASH介绍	74

5.2 主存储器操作	76
5.2.1 解锁/锁定	76
5.2.2 擦除	76
5.2.3 编程	78
5.2.4 读取	79
5.3 主存扩展区操作	79
5.4 OTP操作	80
5.5 用户系统数据区操作	80
5.5.1 解锁/锁定	80
5.5.2 擦除	80
5.5.3 编程	81
5.5.4 读取	82
5.6 闪存保护	82
5.6.1 访问保护	82
5.6.2 擦写保护	83
5.7 读取性能	83
5.8 特殊功能	84
5.8.1 安全库区设定	84
5.8.2 启动程序代码区域作为主存扩展使用	84
5.8.3 CRC校验	85
5.9 FLASH寄存器	85
5.9.1 闪存性能选择寄存器 (FLASH_PSR)	86
5.9.2 闪存解锁寄存器 (FLASH_UNLOCK)	86
5.9.3 闪存用户系统数据解锁寄存器 (FLASH_USD_UNLOCK)	86
5.9.4 闪存状态寄存器 (FLASH_STS)	86
5.9.5 闪存控制寄存器 (FLASH_CTRL)	87
5.9.6 闪存地址寄存器 (FLASH_ADDR)	87
5.9.7 用户系统数据寄存器 (FLASH_USD)	88
5.9.8 擦除编程保护状态寄存器 (FLASH_EPPS)	88
5.9.9 闪存安全库区状态寄存器0 (SLIB_STS0)	88
5.9.10 闪存安全库区状态寄存器1 (SLIB_STS1)	89

5.9.11	闪存安全库区密码清除寄存器 (SLIB_PWD_CLR)	89
5.9.12	闪存安全库区额外状态寄存器 (SLIB_MISC_STS)	90
5.9.13	闪存CRC校验地址寄存器 (FLASH_CRC_ADDR)	90
5.9.14	闪存CRC校验控制寄存器 (FLASH_CRC_CTRL)	90
5.9.15	闪存CRC校验结果寄存器 (FLASH_CRC_CHK)	90
5.9.16	闪存安全库区密码设定寄存器 (SLIB_SET_PWD)	90
5.9.17	闪存安全库区地址设定寄存器 (SLIB_SET_RANGE)	91
5.9.18	主存扩展存储区域安全库区设定寄存器 (EM_SLIB_SET)	91
5.9.19	启动程序代码区模式设定寄存器 (BTM_MODE_SET)	92
5.9.20	闪存安全库区解锁寄存器 (SLIB_UNLOCK)	92
6	通用和复用功能I/O (GPIO和IOMUX)	93
6.1	简介	93
6.2	功能描述	93
6.2.1	GPIO结构	93
6.2.2	GPIO复位状态	94
6.2.3	通用功能输入配置	94
6.2.4	模拟配置	94
6.2.5	通用功能输出配置	94
6.2.6	GPIO端口写保护	95
6.2.7	IOMUX功能结构	95
6.2.8	复用功能配置	96
6.2.9	IOMUX功能输入/输出	97
6.2.10	外设复用功能引脚配置	103
6.2.11	IOMUX映射优先级	103
6.2.12	外部中断/唤醒线	103
6.3	GPIO寄存器	103
6.3.1	GPIO配置寄存器 (GPIOx_CFG) (x=A..F)	104
6.3.2	GPIO输出模式寄存器 (GPIOx_OMODE) (x=A..F)	104
6.3.3	GPIO电流推动/吸入能力切换控制寄存器 (GPIOx_ODRVR) (x=A..F)	105
6.3.4	GPIO上/下拉寄存器 (GPIOx_PULL) (x=A..F)	105
6.3.5	GPIO输入数据寄存器 (GPIOx_IDT) (x=A..F)	105

6.3.6	GPIO输出数据寄存器 (GPIOx_ODT) (x=A..F)	105
6.3.7	GPIO设置/清除寄存器 (GPIOx_SCR) (x=A..F)	106
6.3.8	GPIO写保护寄存器 (GPIOx_WPR) (x=A..F)	106
6.3.9	GPIO复用低位寄存器 (GPIOx_MUXL) (x=A..F)	106
6.3.10	GPIO复用高位寄存器 (GPIOx_MUXH) (x=A..F)	107
6.3.11	GPIO位清除寄存器 (GPIOx_CLR) (x=A..F)	107
6.3.12	GPIO位翻转寄存器 (GPIOx_TOGR) (x=A..F)	107
6.3.13	极大电流推动/吸入能力切换控制寄存器 (GPIOx_HDRV) (x=A..F)	107
7	系统配置控制器 (SCFG)	108
7.1	简介	108
7.2	SCFG寄存器	108
7.2.1	SCFG配置寄存器1 (SCFG_CFG1)	109
7.2.2	SCFG配置寄存器2 (SCFG_CFG2)	109
7.2.3	SCFG外部中断配置寄存器1 (SCFG_EXINTC1)	110
7.2.4	SCFG外部中断配置寄存器2 (SCFG_EXINTC2)	111
7.2.5	SCFG外部中断配置寄存器3 (SCFG_EXINTC3)	112
7.2.6	SCFG外部中断配置寄存器4 (SCFG_EXINTC4)	113
7.2.7	SCFG超高电流推动/吸入能力 (SCFG_UHDRV)	114
7.2.8	SCFG IO配置寄存器 (SCFG_IOCFG)	115
7.2.9	SCFG IO配置写保护寄存器 (SCFG_IOKEY)	115
8	外部中断/事件控制器 (EXINT)	116
8.1	EXINT介绍	116
8.2	功能描述和配置流程	116
8.3	EXINT寄存器描述	117
8.3.1	中断使能寄存器 (EXINT_INTEN)	117
8.3.2	事件使能寄存器 (EXINT_EVTEN)	117
8.3.3	极性配置寄存器1 (EXINT_POLCFG1)	118
8.3.4	极性配置寄存器2 (EXINT_POLCFG2)	118
8.3.5	软件触发寄存器 (EXINT_SWTRG)	118
8.3.6	中断状态寄存器 (EXINT_INTSTS)	118
9	DMA控制器 (DMA)	119

9.1 简介	119
9.2 特性	119
9.3 功能描述	119
9.3.1 通道配置	119
9.3.2 握手机制	120
9.3.3 仲裁	120
9.3.4 可编程数据传输宽度	120
9.3.5 错误事件	121
9.3.6 中断	122
9.4 多路复用器 (DMAMUX)	122
9.4.1 DMAMUX功能描述	122
9.4.2 DMAMUX 溢出中断	124
9.5 DMA寄存器	125
9.5.1 DMA状态寄存器 (DMA_STS)	127
9.5.2 DMA标志清除寄存器 (DMA_CLR)	129
9.5.3 DMA通道x配置寄存器 (DMA_CxCTRL) (x = 1...7)	131
9.5.4 DMA通道x数据传输量寄存器 (DMA_CxDTCNT) (x = 1...7)	132
9.5.5 DMA通道x外设地址寄存器 (DMA_CxPADDR) (x = 1...7)	132
9.5.6 DMA通道x存储器地址寄存器 (DMA_CxMADDR) (x = 1...7)	132
9.5.7 DMAMUX选择寄存器 (DMA_MUXSEL)	132
9.5.8 DMAMUX通道x控制寄存器 (DMA_MUXCxCTRL) (x = 1...7)	133
9.5.9 DMAMUX生成器x控制寄存器 (DMA_MUXGxCTRL) (x = 1...4)	134
9.5.10 DMAMUX通道同步状态寄存器 (DMA_MUXSYNCSTS)	134
9.5.11 DMAMUX通道中断清除标志寄存器 (DMA_MUXSYNCCCLR)	134
9.5.12 DMAMUX发生器中断状态寄存器 (DMA_MUXGSTS)	135
9.5.13 DMAMUX发生器中断清除标志寄存器 (DMA_MUXGCLR)	135
10 CRC计算单元 (CRC)	136
10.1 CRC介绍	136
10.2 CRC功能说明	136
10.3 CRC寄存器	137
10.3.1 数据寄存器 (CRC_DT)	137

10.3.2 通用数据寄存器 (CRC_CDT)	137
10.3.3 控制寄存器 (CRC_CTRL)	138
10.3.4 初始化寄存器 (CRC_IDT)	138
10.3.5 生成多项式系数寄存器 (CRC_POLY)	138
11 I ² C接口	139
11.1 I ² C简介	139
11.2 I ² C的主要特点	139
11.3 I ² C总线特性	139
11.4 I ² C接口	140
11.4.1 I ² C时序控制	141
11.4.2 数据传输管理	143
11.4.3 I ² C主机通信流程	144
11.4.4 I ² C从机通信流程	148
11.4.5 SMBus功能	152
11.4.6 SMBus主机通信流程	154
11.4.7 SMBus从机通信流程	157
11.4.8 DMA传输	161
11.4.9 错误管理	161
11.5 I ² C中断	163
11.6 I ² C调试模式	163
11.7 I ² C寄存器	163
11.7.1 控制寄存器1 (I2C_CTRL1)	164
11.7.2 控制寄存器2 (I2C_CTRL2)	165
11.7.3 地址寄存器1 (I2C_OADDR1)	166
11.7.4 地址寄存器2 (I2C_OADDR2)	166
11.7.5 时序寄存器 (I2C_CLKCTRL)	166
11.7.6 超时寄存器 (I2C_TIMEOUT)	167
11.7.7 状态寄存器 (I2C_STS)	167
11.7.8 状态清除寄存器 (I2C_CLR)	169
11.7.9 PEC寄存器 (I2C_PEC)	169
11.7.10 接收寄存器 (I2C_RXDT)	169

11.7.11	发送寄存器 (I2C_TXDT)	169
12	通用同步异步收发器 (USART)	170
12.1	USART介绍	170
12.2	全双工半双工选择器简述和配置流程	171
12.3	模式选择器简述和配置流程	172
12.3.1	模式选择器简述	172
12.3.2	模式选择器配置方法	172
12.4	USART帧格式简述和配置流程	176
12.5	DMA传输简述和配置流程	177
12.5.1	DMA发送配置流程	177
12.5.2	DMA接收配置流程	178
12.6	波特率发生器简述及配置流程	178
12.6.1	波特率发生器简述	178
12.6.2	波特率发生器配置方法	178
12.7	发送器简述和配置流程	179
12.7.1	发送器简述	179
12.7.2	发送器配置流程	179
12.8	接收器简述和配置流程	180
12.8.1	接收器简述	180
12.8.2	接收器配置流程	180
12.8.3	起始侦测和噪声检测	181
12.9	Tx/Rx可配置引脚互换	182
12.10	中断	182
12.11	I/O引脚控制	183
12.12	USART寄存器描述	183
12.12.1	状态寄存器 (USART_STS)	184
12.12.2	数据寄存器 (USART_DT)	185
12.12.3	波特比率寄存器 (USART_BAUDR)	185
12.12.4	控制寄存器1 (USART_CTRL1)	185
12.12.5	控制寄存器2 (USART_CTRL2)	187
12.12.6	控制寄存器3 (USART_CTRL3)	188

12.12.7 保护时间和预分频寄存器 (GDIV)	189
12.12.8 接收器超时检测值寄存器 (RTOV)	189
12.12.9 中断标志位清除寄存器 (IFC)	189
13 串行外设接口 (SPI)	190
13.1 串行外设接口 (SPI) 简介	190
13.2 SPI功能描述	190
13.2.1 SPI简述	190
13.2.2 全双工半双工选择器简述和配置流程	191
13.2.3 CS控制器简述和配置流程	193
13.2.4 SPI_SCK控制器简述和配置流程	194
13.2.5 CRC简述和配置流程	194
13.2.6 DMA传输简述和配置流程	194
13.2.7 TI模式简述和配置流程	195
13.2.8 发送器简述和配置流程	195
13.2.9 接收器简述和配置流程	196
13.2.10 Motorola模式通信时序	196
13.2.11 TI模式通信时序	198
13.2.12 中断	200
13.2.13 IO管脚控制	200
13.3 I ² S功能描述	200
13.3.1 I ² S简述	200
13.3.2 操作模式选择器简述和配置流程	201
13.3.3 音频协议选择器简述和配置流程	203
13.3.4 I ² S_CLK控制器简述和配置流程	204
13.3.5 DMA传输简述和配置流程	205
13.3.6 发送器接收器简述和配置流程	205
13.3.7 I ² S通信时序	206
13.3.8 中断	207
13.3.9 IO管脚控制	207
13.4 SPI寄存器	207
13.4.1 SPI控制寄存器1 (SPI_CTRL1) (I ² S模式下不使用)	207

13.4.2 SPI控制寄存器2 (SPI_CTRL2)	209
13.4.3 SPI状态寄存器 (SPI_STS)	210
13.4.4 SPI数据寄存器 (SPI_DT)	210
13.4.5 SPICRC多项式寄存器 (SPI_CPOLY) (I2S模式下不使用)	210
13.4.6 SPIRxCRC寄存器 (SPI_RCRC) (I2S模式下不使用)	211
13.4.7 SPITxCRC寄存器 (SPI_TCRC)	211
13.4.8 SPI_I2S配置寄存器 (SPI_I2SCTRL)	211
13.4.9 SPI_I2S预分频寄存器 (SPI_I2SCLKP)	212
14 定时器 (TIMER)	213
14.1 基本定时器 (TMR6和TMR7)	213
14.1.1 TMR6和TMR7简介	213
14.1.2 TMR6和TMR7的主要功能	213
14.1.3 TMR6和TMR7的功能	214
14.1.3.1 计数时钟	214
14.1.3.2 计数模式	214
14.1.3.3 OVFIF位重映射	215
14.1.3.4 调试模式	215
14.1.4 TMR6和TMR7寄存器	216
14.1.4.1 TMR6和TMR7控制寄存器1 (TMRx_CTRL1)	216
14.1.4.2 TMR6和TMR7控制寄存器2 (TMRx_CTRL2)	217
14.1.4.3 TMR6和TMR7 DMA/中断使能寄存器 (TMRx_IDEN)	217
14.1.4.4 TMR6和TMR7中断状态寄存器 (TMRx_ISTS)	217
14.1.4.5 TMR6和TMR7软件事件寄存器 (TMRx_SWEVT)	217
14.1.4.6 TMR6和TMR7计数值 (TMRx_CVAL)	217
14.1.4.7 TMR6和TMR7分频系数 (TMRx_DIV)	217
14.1.4.8 TMR6和TMR7周期寄存器 (TMRx_PR)	218
14.2 通用定时器 (TMR3至TMR4)	219
14.2.1 TMR3至TMR4简介	219
14.2.2 TMR3至TMR4主要功能	219
14.2.3 TMR3至TMR4功能描述	219
14.2.3.1 计数时钟	219
14.2.3.2 计数模式	222
14.2.3.3 OVFIF位重映射	225
14.2.3.4 TMR输入部分	225
14.2.3.5 TMR输出部分	228
14.2.3.6 定时器同步	232

14.2.3.7 TMR DMA	235
14.2.3.8 调试模式	235
14.2.4 TMR3至TMR4寄存器描述	236
14.2.4.1 TMRx控制寄存器1 (TMRx_CTRL1)	237
14.2.4.2 TMRx控制寄存器2 (TMRx_CTRL2)	237
14.2.4.3 TMRx次定时器控制寄存器 (TMRx_STCTRL)	238
14.2.4.4 TMRx DMA/中断使能寄存器 (TMRx_IDEN)	239
14.2.4.5 TMRx 中断状态寄存器 (TMRx_ISTS)	240
14.2.4.6 TMRx软件事件寄存器 (TMRx_SWEVT)	241
14.2.4.7 TMRx通道模式寄存器1 (TMRx_CM1)	241
14.2.4.8 TMRx通道模式寄存器2 (TMRx_CM2)	244
14.2.4.9 TMRx通道控制寄存器 (TMRx_CCTRL)	245
14.2.4.10 TMRx计数值寄存器 (TMRx_CVAL)	245
14.2.4.11 TMRx分频系数寄存器 (TMRx_DIV)	246
14.2.4.12 TMRx周期寄存器 (TMRx_PR)	246
14.2.4.13 TMRx通道1数据寄存器 (TMRx_C1DT)	246
14.2.4.14 TMRx通道2数据寄存器 (TMRx_C2DT)	246
14.2.4.15 TMRx通道3数据寄存器 (TMRx_C3DT)	246
14.2.4.16 TMRx通道4数据寄存器 (TMRx_C4DT)	246
14.2.4.17 TMRx DMA控制寄存器 (TMRx_DMACTRL)	247
14.2.4.18 TMRx DMA数据寄存器 (TMRx_DMADT)	247
14.2.4.19 TMRx 选项寄存器1 (TMRx_OPT1)	247
14.2.4.20 TMR3 选项寄存器2 (TMR3_OPT2)	247
14.2.4.21 TMR4 选项寄存器2 (TMR4_OPT2)	247
14.2.4.22 TMR3 通道选择寄存器 (TMR3_CHSEL)	248
14.2.4.23 TMR4 通道选择寄存器 (TMR4_CHSEL)	248
14.2.4.24 TMR3 定时器消隐控制寄存器 (TMR3_BLKCTRL)	249
14.2.4.25 TMR4 定时器消隐控制寄存器 (TMR4_BLKCTRL)	249
14.3 通用定时器 (TMR9到TMR11)	251
14.3.1 TMR9到TMR11简介	251
14.3.2 TMR9到TMR11主要功能	251
14.3.2.1 TMR9主要功能	251
14.3.2.2 TMR10和TMR11主要功能	251
14.3.3 TMR9到TMR11功能描述	252
14.3.3.1 计数时钟	252
14.3.3.2 计数模式	254
14.3.3.3 OVFIF位重映射	256
14.3.3.4 TMR输入部分	256
14.3.3.5 TMR输出部分	259
14.3.3.6 TMR刹车功能	262
14.3.3.7 TMR同步	264

14.3.3.8 TMR DMA	265
14.3.3.9 调试模式	265
14.3.4 TMR9寄存器描述	266
14.3.4.1 TMR9控制寄存器1 (TMR9_CTRL1)	267
14.3.4.2 TMR9控制寄存器2 (TMR9_CTRL2)	268
14.3.4.3 TMR9次定时器控制寄存器 (TMR9_STCTRL)	268
14.3.4.4 TMR9 DMA中断使能寄存器 (TMR9_IDEN)	269
14.3.4.5 TMR9中断状态寄存器 (TMR9_ISTS)	270
14.3.4.6 TMR9软件事件寄存器 (TMR9_SWEVT)	271
14.3.4.7 TMR9通道模式寄存器1 (TMR9_CM1)	271
14.3.4.8 TMR9通道控制寄存器 (TMR9_CCTRL)	274
14.3.4.9 TMR9计数值寄存器 (TMR9_CVAL)	276
14.3.4.10 TMR9预分频器寄存器 (TMR9_DIV)	276
14.3.4.11 TMR9周期寄存器 (TMR9_PR)	276
14.3.4.12 TMR9重复周期寄存器 (TMR9_RPR)	276
14.3.4.13 TMR9通道1数据寄存器 (TMR9_C1DT)	276
14.3.4.14 TMR9通道2数据寄存器 (TMR9_C2DT)	276
14.3.4.15 TMR9刹车寄存器 (TMR9_BRK)	277
14.3.4.16 TMR9 DMA控制寄存器 (TMR9_DMACTRL)	278
14.3.4.17 TMR9 DMA数据寄存器 (TMR9_DMADT)	278
14.3.4.18 TMR9选项寄存器2 (TMR9_OPT2)	279
14.3.4.19 TMR9通道选择寄存器 (TMR9_CHSEL)	279
14.3.5 TMR10、TMR11寄存器描述	280
14.3.5.1 TMR10和TMR11控制寄存器1 (TMRx_CTRL1)	281
14.3.5.2 TMR10和TMR11控制寄存器2 (TMRx_CTRL2)	281
14.3.5.3 TMR10和TMR11 DMA/中断使能寄存器 (TMRx_IDEN)	282
14.3.5.4 TMR10和TMR11中断状态寄存器 (TMRx_ISTS)	282
14.3.5.5 TMR10和TMR11软件事件寄存器 (TMRx_SWEVT)	283
14.3.5.6 TMR10和TMR11通道模式寄存器1 (TMRx_CM1)	284
14.3.5.7 TMR10和TMR11通道控制寄存器 (TMRx_CCTRL)	285
14.3.5.8 TMR10和TMR11计数值寄存器 (TMRx_CVAL)	287
14.3.5.9 TMR10和TMR11预分频器寄存器 (TMRx_DIV)	287
14.3.5.10 TMR10和TMR11周期寄存器 (TMRx_PR)	287
14.3.5.11 TMR10和TMR11周期寄存器 (TMRx_RPR)	287
14.3.5.12 TMR10和TMR11通道1数据寄存器 (TMRx_C1DT)	287
14.3.5.13 TMR10和TMR11刹车寄存器 (TMRx_BRK)	287
14.3.5.14 TMR10和TMR11 DMA控制寄存器 (TMRx_DMACTRL)	289
14.3.5.15 TMR10和TMR11 DMA数据寄存器 (TMRx_DMADT)	289
14.3.5.16 TMR10和TMR11选项寄存器2 (TMRx_OPT2)	289
14.3.5.17 TMR10通道选择寄存器 (TMR10_CHSEL)	290
14.3.5.18 TMR11通道选择寄存器 (TMR11_CHSEL)	290

14.4 高级控制定时器 (TMR1)	291
14.4.1 TMR1简介	291
14.4.2 TMR1主要特性	291
14.4.3 TMR1功能描述	292
14.4.3.1 计数时钟	292
14.4.3.2 计数模式	294
14.4.3.3 OVFIF位重映射	298
14.4.3.4 TMR输入部分	299
14.4.3.5 TMR输出部分	301
14.4.3.6 TMR刹车功能	307
14.4.3.7 TMR同步	309
14.4.3.8 TMR DMA	310
14.4.3.9 调试模式	311
14.4.4 TMR1寄存器描述	312
14.4.4.1 TMR1控制寄存器1 (TMR1_CTRL1)	313
14.4.4.2 TMR1控制寄存器2 (TMR1_CTRL2)	314
14.4.4.3 TMR1次定时器控制寄存器 (TMR1_STCTRL)	315
14.4.4.4 TMR1 DMA/中断使能寄存器 (TMR1_IDEN)	316
14.4.4.5 TMR1中断状态寄存器 (TMR1_ISTS)	317
14.4.4.6 TMR1软件事件寄存器 (TMR1_SWEVT)	318
14.4.4.7 TMR1通道模式寄存器1 (TMR1_CM1)	319
14.4.4.8 TMR1通道模式寄存器2 (TMR1_CM2)	322
14.4.4.9 TMR1通道控制寄存器 (TMR1_CCTRL)	323
14.4.4.10 TMR1计数值寄存器 (TMR1_CVAL)	325
14.4.4.11 TMR1预分频器寄存器 (TMR1_DIV)	325
14.4.4.12 TMR1周期寄存器 (TMR1_PR)	325
14.4.4.13 TMR1重复周期寄存器 (TMR1_RPR)	325
14.4.4.14 TMR1通道1数据寄存器 (TMR1_C1DT)	325
14.4.4.15 TMR1通道2数据寄存器 (TMR1_C2DT)	325
14.4.4.16 TMR1通道3数据寄存器 (TMR1_C3DT)	326
14.4.4.17 TMR1通道4数据寄存器 (TMR1_C4DT)	326
14.4.4.18 TMR1刹车寄存器 (TMR1_BRK)	326
14.4.4.19 TMR1 DMA控制寄存器 (TMR1_DMACTRL)	328
14.4.4.20 TMR1 DMA数据寄存器 (TMR1_DMADT)	328
14.4.4.21 TMR1 选项寄存器1 (TMR1_OPT1)	328
14.4.4.22 TMR1 通道模式寄存器3 (TMR1_CM3)	328
14.4.4.23 TMR1 通道5数据寄存器 (TMR1_C5DT)	329
14.4.4.24 TMR1通道6数据寄存器 (TMR1_C6DT)	329
14.4.4.25 TMR1选项寄存器2 (TMR1_OPT2)	329
14.4.4.26 TMR1选项寄存器3 (TMR1_OPT3)	330
14.4.4.27 TMR1通道选择寄存器 (TMR1_CHSEL)	330

14.4.4.28 TMR1通道1非对称数据寄存器 (TMR1_C1ADT)	330
14.4.4.29 TMR1通道2非对称数据寄存器 (TMR1_C2ADT)	331
14.4.4.30 TMR1通道3非对称数据寄存器 (TMR1_C3ADT)	331
14.4.4.31 TMR1通道4非对称数据寄存器 (TMR1_C4ADT)	331
15 窗口看门狗 (WWDT)	332
15.1 WWDT简介	332
15.2 WWDT主要特性	332
15.3 WWDT功能描述	332
15.4 调试模式	333
15.5 WWDT寄存器	333
15.5.1 控制寄存器 (WWDT_CTRL)	333
15.5.2 配置寄存器 (WWDT_CFG)	333
15.5.3 状态寄存器 (WWDT_STS)	334
16 看门狗 (WDT)	335
16.1 WDT简介	335
16.2 WDT主要特性	335
16.3 WDT功能描述	335
16.4 调试模式	336
16.5 WDT寄存器	337
16.5.1 命令寄存器 (WDT_CMD)	337
16.5.2 预分频寄存器 (WDT_DIV)	337
16.5.3 重装载寄存器 (WDT_RLD)	337
16.5.4 状态寄存器 (WDT_STS)	338
16.5.5 窗口寄存器 (WDT_WIN)	338
17 实时时钟 (ERTC)	339
17.1 ERTC简介	339
17.2 ERTC主要特性	339
17.3 ERTC功能说明	339
17.3.1 ERTC时钟	339
17.3.2 ERTC初始化	340
17.3.3 周期性自动唤醒	341
17.3.4 ERTC校准	342

17.3.5 参考时钟检测	342
17.3.6 时间戳	342
17.3.7 入侵检测	342
17.3.8 复用功能输出	343
17.3.9 ERTC唤醒	343
17.4 ERTC寄存器描述	344
17.4.1 ERTC时间寄存器(ERTC_TIME)	345
17.4.2 ERTC日期寄存器(ERTC_DATE)	345
17.4.3 ERTC控制寄存器(ERTC_CTRL)	345
17.4.4 ERTC初始化和状态寄存器(ERTC_STS)	347
17.4.5 ERTC预分频器寄存器(ERTC_DIV)	348
17.4.6 ERTC唤醒定时器寄存器(ERTC_WAT)	348
17.4.7 ERTC闹钟A寄存器(ERTC_ALA)	348
17.4.8 ERTC闹钟B寄存器(ERTC_ALB)	349
17.4.9 ERTC写保护寄存器(ERTC_WP)	349
17.4.10 ERTC亚秒寄存器(ERTC_SBS)	349
17.4.11 ERTC时间微调寄存器(ERTC_TADJ)	349
17.4.12 ERTC时间戳时间寄存器(ERTC_TSTM)	350
17.4.13 ERTC时间戳日期寄存器(ERTC_TS DT)	350
17.4.14 ERTC时间戳亚秒寄存器(ERTC_TSSBS)	350
17.4.15 ERTC精密校准寄存器(ERTC_SCAL)	350
17.4.16 ERTC入侵配置寄存器(ERTC_TAMP)	351
17.4.17 ERTC闹钟A亚秒寄存器(ERTC_ALASBS)	352
17.4.18 ERTC闹钟A亚秒寄存器(ERTC_ALASBS)	352
17.4.19 ERTC 电池供电数据寄存器(ERTC_BPRx)	352
18 模拟/数字转换 (ADC)	353
18.1 ADC简介	353
18.2 ADC主要特征	353
18.3 ADC架构	354
18.4 ADC功能介绍	354
18.4.1 通道管理	354
18.4.1.1 内部温度传感器	355

18.4.1.2 内部参考电压	356
18.4.2 ADC操作流程	356
18.4.2.1 上电与校准	356
18.4.2.2 触发	357
18.4.2.3 采样与转换时序	358
18.4.3 转换顺序管理	358
18.4.3.1 序列模式	358
18.4.3.2 抢占自动转换模式	358
18.4.3.3 反复模式	359
18.4.3.4 分割模式	359
18.4.4 转换中止	360
18.4.5 触发转换失败	360
18.4.6 过采样器	360
18.4.6.1 普通通道过采样	361
18.4.6.2 抢占通道过采样	362
18.4.7 数据管理	362
18.4.7.1 数据内容处理	362
18.4.7.2 数据获取	362
18.4.8 电压监测	363
18.4.8.1 状态标志与中断	363
18.5 主从模式	364
18.5.1 数据管理	364
18.5.2 同时模式	365
18.5.3 抢占交错触发模式	366
18.5.4 普通位移模式	366
18.6 ADC寄存器	367
18.6.1 ADC状态寄存器 (ADC_STS)	368
18.6.2 ADC控制寄存器1 (ADC_CTRL1)	369
18.6.3 ADC控制寄存器2 (ADC_CTRL2)	370
18.6.4 ADC采样时间寄存器1 (ADC_SPT1)	372
18.6.5 ADC采样时间寄存器2 (ADC_SPT2)	373
18.6.6 ADC抢占通道数据偏移寄存器x (ADC_PCDTOx) (x=1..4)	375
18.6.7 ADC电压监测高边界寄存器 (ADC_VMHB)	375
18.6.8 ADC电压监测低边界寄存器 (ADC_VMLB)	375
18.6.9 ADC普通序列寄存器1 (ADC_OSQ1)	375

18.6.10 ADC普通序列寄存器2 (ADC_OSQ2)	376
18.6.11 ADC普通序列寄存器3 (ADC_OSQ3)	376
18.6.12 ADC抢占序列寄存器 (ADC_PSQ)	376
18.6.13 ADC抢占数据寄存器x (ADC_PDTx) (x= 1..4)	377
18.6.14 ADC普通数据寄存器 (ADC_ODT)	377
18.6.15 ADC采样时间寄存器3 (ADC_SPT3)	377
18.6.16 ADC过采样寄存器 (ADC_OVSP)	378
18.6.17 ADC通用状态寄存器 (ADC_CSTS)	379
18.6.18 ADC通用控制寄存器 (ADC_CCTRL)	380
18.6.19 ADC通用普通数据寄存器 (ADC_CODT)	381
18.6.20 ADC _i 公共普通数据寄存器x(ADC _i _ODTx,i=1/2,x=1~16)	381
19 数字/模拟转换 (DAC)	382
19.1 简介	382
19.2 主要特性	382
19.3 设计提示	382
19.4 功能描述	383
19.4.1 触发事件	383
19.4.2 数据配置	383
19.5 DAC寄存器	384
19.5.1 DAC控制寄存器 (DAC_CTRL)	384
19.5.2 DAC软件触发寄存器 (DAC_SWTRG)	385
19.5.3 DAC1的6位数据保持寄存器 (DAC_D1DTH)	386
19.5.4 DAC2的6位数据保持寄存器 (DAC_D2DTH)	386
19.5.5 双DAC的6位数据保持寄存器 (DAC_DDDTH)	386
19.5.6 DAC1数据输出寄存器 (DAC_D1ODT)	386
19.5.7 DAC2数据输出寄存器 (DAC_D2ODT)	386
19.5.8 DAC状态寄存器 (DAC_STS)	386
19.5.9 DAC输出到GPIO使能寄存器 (DAC_OUTEN)	387
20 比较器 (CMP)	388
20.1 简介	388
20.2 主要特性	388

20.3 中断管理	388
20.4 设计提示	388
20.5 功能描述	389
20.5.1 模拟比较器	389
20.5.2 干扰滤波器	390
20.6 CMP寄存器	390
20.6.1 比较器1控制和状态寄存器(CMP1_CTRLSTS)	391
20.6.2 比较器1干扰滤波器使能寄存器(CMP1_G_FILTER_EN)	392
20.6.3 比较器1干扰滤波器高脉冲数(CMP1_HIGH_PULSE)	392
20.6.4 比较器1干扰滤波器低脉冲数(CMP1_LOW_PULSE)	392
20.6.5 比较器2控制和状态寄存器(CMP2_CTRLSTS)	392
20.6.6 比较器2干扰滤波器使能寄存器(CMP2_G_FILTER_EN)	393
20.6.7 比较器2干扰滤波器高脉冲数(CMP2_HIGH_PULSE)	393
20.6.8 比较器2干扰滤波器低脉冲数(CMP2_LOW_PULSE)	394
21 运算放大器 (OP)	395
21.1 简介	395
21.2 运算放大器主要功能	395
21.3 运算放大器功能描述	395
21.3.1 运算放大器时钟和复位	395
21.3.2 运算放大器模式	395
21.3.3 独立模式	396
21.3.4 跟随模式	397
21.3.5 可配置增益模式 (正向增益模式)	398
21.3.6 可配置增益模式 (带外部偏置正向增益或反向增益模式)	399
21.3.7 可配置增益模式 (带外部滤波器的正向增益模式)	400
21.3.8 可配置增益模式 (带外部滤波器的带外部偏置正向增益或反向增益模式)	401
21.3.9 运算放大器通道连接	402
21.4 校准	403
21.5 Timer自动切换功能	404
21.6 OP 控制寄存器	404
21.6.1 OP1 控制以及状态寄存器 (OP1_CTRL)	404

21.6.2 OP2 控制以及状态寄存器 (OP2_CTRL)	406
21.6.3 OP3 控制以及状态寄存器 (OP3_CTRL)	408
21.6.4 OP4 控制以及状态寄存器 (OP4_CTRL)	410
21.6.5 OP1 timer 控制寄存器 (OP1_TC)	411
21.6.6 OP2 timer 控制寄存器 (OP2_TC)	412
21.6.7 OP3 timer 控制寄存器 (OP3_TC)	412
21.6.8 OP4 timer 控制寄存器 (OP4_TC)	413
22 控制器区域网络 (CAN)	414
22.1 概述	414
22.1.1 CAN控制器内核	414
22.1.2 CAN控制器协议	414
22.1.3 Classic CAN2.0B以及CANFD	414
22.1.4 时间触发	417
22.1.5 CiA 603时间戳	417
22.2 特性	417
22.2.1 特征列表	417
22.2.2 中断	418
22.2.3 软件操作接口	419
22.3 操作指南	421
22.3.1 接收过滤器	421
22.3.2 帧接收	422
22.3.3 帧接收处理	423
22.3.4 帧发送处理	423
22.3.5 中止帧传输	424
22.3.6 STB满	425
22.3.7 错误处理	425
22.3.8 总线离线	426
22.3.9 扩展状态和错误报告	426
22.3.9.1 可编程的错误警告界限	426
22.3.9.2 仲裁丢失捕获 (ALC)	426
22.3.9.3 错误类型 (KOER)	427
22.3.9.4 接收所有数据帧 (RBALL)	427
22.3.9.5 传输状态 (TSTAT_1和TSTAT_2)	427

22.3.9.6 帧拒绝发送	429
22.3.10 比特率切换和收发模式转换	429
22.3.11 扩展特性	429
22.3.11.1 限制重传和重新仲裁	429
22.3.11.2 只听模式（LOM）	430
22.3.11.3 受限操作（ROP）	430
22.3.11.4 环回模式（LBMI和LBME）	430
22.3.11.5 收发器待机模式	431
22.3.11.6 错误计数器复位	431
22.3.11.7 软件复位	431
22.4 时间触发CAN (TTCAN)	434
22.4.1 介绍	434
22.4.2 TTCAN模式下TBUF配置	434
22.4.2.1 当TTTBM=1时TBUF的行为	435
22.4.2.2 当TTTBM=0时TBUF的行为	435
22.4.3 TTCAN操作	435
22.4.4 TTCAN时序	435
22.4.5 TTCAN触发类型	436
22.4.5.1 立即触发	436
22.4.5.2 时间触发	436
22.4.5.3 单次发送触发	436
22.4.5.4 发送启动触发	437
22.4.5.5 发送停止触发	437
22.4.6 TTCAN窗口触发	437
22.5 CiA 603时间戳	437
22.6 CAN位时间	438
22.6.1 位速率	438
22.6.2 位时间定义	438
22.6.3 CANFD位速率转换和采样点	439
22.6.4 TDC和RDC	439
22.7 CAN寄存器	440
22.7.1 CAN CiA 603时间戳及节点控制寄存器 (TNCFG)	441
22.7.2 Classic CAN2.0B位时序寄存器 (ACTIME)	441
22.7.3 CANFD位时序寄存器 (FDTIME)	442
22.7.4 CAN限制和位时间配置寄存器 (LBTCFG)	442

22.7.5 CAN状态寄存器 (STS)	443
22.7.6 CAN发送状态寄存器 (TSTAT)	444
22.7.7 CAN发送时间戳32位 (TTS)	444
22.7.8 CAN控制和状态寄存器 (CTRLSTAT)	444
22.7.9 CAN错误配置及状态寄存器 (ERR)	448
22.7.10 CAN TTCAN : 参考信息 (REFMSG)	449
22.7.11 CAN TTCAN : 触发配置寄存器 (TTCFG)	449
22.7.12 CAN TTCAN : 触发时间 (TTTRIG)	450
22.7.13 CAN接收过滤器控制器 (ACFCTRL)	450
22.7.14 CAN过滤器标识符ID区 (FCID)	451
22.7.15 CAN过滤器标识符格式区 (FCFMT)	451
22.7.16 CAN过滤器标识符类型区 (FCTYP)	451
22.7.17 CAN过滤器标识符接收场 (FCACF)	451
22.7.18 CAN过滤器掩码ID区 (FMID)	451
22.7.19 CAN过滤器掩码格式区 (FMFMT)	451
22.7.20 CAN过滤器掩码类型区 (FMTYP)	451
22.7.21 CAN过滤器掩码接收场 (FMACF)	452
22.7.22 CAN接收缓冲区ID区 (RBID)	452
22.7.23 CAN接收缓冲区格式区 (RBFMT)	452
22.7.24 CAN接收缓冲区类型区 (RBTYP)	452
22.7.25 CAN接收缓冲区接收场 (RBACF)	452
22.7.26 CAN接收缓冲区数据区 (RBDATn)	452
22.7.27 CAN接收缓冲区CiA 603接收时间戳1 (RBCIA1)	452
22.7.28 CAN接收缓冲区CiA 603接收时间戳2 (RBCIA2)	452
22.7.29 CAN接收缓冲区TTCAN周期时间 (RBTTCAN)	452
22.7.30 CAN发送缓冲区ID区 (TBID)	453
22.7.31 CAN发送缓冲区格式区 (TBFMT)	453
22.7.32 CAN发送缓冲区类型区 (TBTYP)	453
22.7.33 CAN发送缓冲区接收场 (TBACF)	453
22.7.34 CAN发送缓冲区数据区 (TBDATn)	453
22.7.35 CAN LLC帧计算输入 (LLCFORMAT)	453

22.7.36 CAN LLC帧计算输出 (LLCSIZE)	453
22.7.37 CAN中断使能寄存器 (INTEN)	454
23 红外线接口 (IRTMR)	455
24 调试 (DEBUG)	456
24.1 简介	456
24.2 调试与跟踪功能	456
24.3 I/O控制	456
24.4 DEBUG寄存器	456
24.4.1 DEBUG设备ID (DEBUG_IDCODE)	457
24.4.2 DEBUG控制寄存器 (DEBUG_CTRL)	458
24.4.3 DEBUG APB1暂停控制寄存器 (DEBUG_APB1_PAUSE)	458
24.4.4 DEBUG APB2暂停控制寄存器 (DEBUG_APB2_PAUSE)	459
24.4.5 DEBUG APB3暂停控制寄存器 (DEBUG_APB3_PAUSE)	460
24.4.6 DEBUG SERIES ID寄存器 (DEBUG_SER_ID)	460
25 版本历史	461

图目录

图 1-1 AT32M412/AT32M416 系列微控制器系统架构	31
图 1-2 CortexTM-M4F 内部框图	32
图 1-3 位带区与位带别名区的膨胀关系图 A.....	32
图 1-4 位带区与位带别名区的膨胀关系图 B.....	33
图 1-5 复位流程	38
图 1-6 MSP 及 PC 初始化的一个范例	38
图 2-1 AT32M412/AT32M416 地址配置	41
图 3-1 各电源域框图.....	45
图 3-2 上电/低电压复位波形图	46
图 3-3 PVM 的阈值与输出	46
图 4-1 AT32M412/AT32M416 时钟结构图	54
图 4-2 系统复位电路	56
图 5-1 主存储器扇区擦除流程图.....	77
图 5-2 主存储器整片擦除流程图.....	78
图 5-3 主存储器编程流程图	79
图 5-4 系统数据区擦除图	81
图 5-5 系统数据区编程图	82
图 6-1 GPIO 基本结构	93
图 6-2 IOMUXU 复用结构	95
图 8-1 外部中断/事件控制器框图.....	116
图 9-1 DMA 框图	119
图 9-2 请求/应答对后重新仲裁	120
图 9-3 PWIDHT: byte, MWIDHT: half-word	121
图 9-4 PWIDHT: half-word, MWIDHT: word	121
图 9-5 PWIDHT: word, MWIDHT: byte	121
图 9-6 DMAMUX 框图	122
图 9-7 DMAMUX 请求同步模式	124
图 9-8 DMAMUX 事件生成	125
图 10-1 CRC 计算单元框图	136
图 10-2 字节翻转示意图	137
图 11-1 I ² C 总线协议	139
图 11-2 I ² C 框图.....	140
图 11-3 建立和保持时间	142
图 11-4 I ² C 主机发送流程图	146
图 11-5 I ² C 主机发送时序图	146
图 11-6 I ² C 主机接收流程图	147
图 11-7 I ² C 主机接收时序图	147
图 11-8 10 位地址读访问 READH10=1	148
图 11-9 10 位地址读访问 READH10=0	148
图 11-10 I ² C 从机发送流程图	150
图 11-11 I ² C 从机发送时序图	150
图 11-12 I ² C 从机接收流程图	151
图 11-13 I ² C 从机接收时序图	151
图 11-14 SMBus 主机发送流程图	155
图 11-15 SMBus 主机发送时序图	156
图 11-16 SMBus 主机接收流程图	156
图 11-17 SMBus 主机接收时序图	157
图 11-18 SMBus 从机发送流程图	159
图 11-19 SMBus 从机发送时序图	159
图 11-20 SMBus 从机接收流程图	160
图 11-21 SMBus 从机接收时序图	160

图 12-1 USART 框图	170
图 12-2 LIN 模式下的 BFF 检测与 FERR 检测	172
图 12-3 Smartcard frame format	173
图 12-4 IrDA DATA(3/16)-普通模式	173
图 12-5 Hardware flow control	174
图 12-6 Mute mode using Idle line or Address mark detection	175
图 12-7 8-bit format USART 同步模式	175
图 12-8 字长设置	176
图 12-9 配置停止位	177
图 12-10 发送时 TDC/TDBE 的变化情况	179
图 12-11 检测噪声的数据采样	182
图 12-12 USART 中断映像图	183
图 13-1 SPI 框图	190
图 13-2 数据时钟时序图	191
图 13-3 SPI 双线单向全双工连接示意图	192
图 13-4 SPI 作主机单线单向只收连接示意图	192
图 13-5 SPI 作从机单线单向只收连接示意图	192
图 13-6 SPI 作单线双向半双工连接示意图	193
图 13-7 主机全双工通信	197
图 13-8 从机全双工通信	197
图 13-9 主机半双工发送通信	197
图 13-10 从机半双工接收通信	198
图 13-11 从机半双工发送通信	198
图 13-12 主机半双工接收通信	198
图 13-13 TI 模式连续通信时序	199
图 13-14 TI 模式带 dummy CLK 的连续通信时序	199
图 13-15 TI 模式非连续通信时序	199
图 13-16 TI 模式 SCK 及从机 MISO 释放时刻	199
图 13-17 SPI 中断	200
图 13-18 I ² S 框图	201
图 13-19 I ² S 从设备发送连接示意图	202
图 13-20 I ² S 从设备接收连接示意图	202
图 13-21 I ² S 主设备发送连接示意图	202
图 13-22 I ² S 主设备接收连接示意图	203
图 13-23 SPI 作主机 CK & MCK 来源示意图	204
图 13-24 各音频标准时序	206
图 13-25 I ² S 中断	207
图 14-1 基本定时器框图	214
图 14-2 使用 CK_INT 且分频系数为 1	214
图 14-3 计数器基本结构	214
图 14-4 PRBEN=0 时的溢出事件	215
图 14-5 PRBEN=1 时的溢出事件	215
图 14-6 计数器时序图, 内部时钟分频因子为 4	215
图 14-7 通用定时器框图	219
图 14-8 计数时钟	219
图 14-9 使用 CK_INT 计数, TMRx_DIV=0x0, 周期寄存器 TMRx_PR=0x16	220
图 14-10 外部时钟模式 A 框图	221
图 14-11 使用外部时钟模式 A 计数, PR=0x32, DIV=0x0	221
图 14-12 外部时钟模式 B 框图	221
图 14-13 使用外部时钟模式 B 计数, PR=0x32, DIV=0x0	221
图 14-14 当预分频器的参数从 1 变到 4 时, 计数器的时序图	222
图 14-15 计数器基本结构	222

图 14-16 PRBEN=0 时的溢出事件	223
图 14-17 PRBEN=1 时的溢出事件	223
图 14-18 计数器时序图, 内部时钟分频因子为 4	223
图 14-19 计数器时序图, 内部时钟分频因子为 1, TMRx_PR=0x32	224
图 14-20 编码模式结构	224
图 14-21 编码模式计数实例 (编码器模式 C)	225
图 14-22 输入/输出通道 1 的主电路	226
图 14-23 通道 1 输入部分	226
图 14-24 输入消隐功能图示	227
图 14-25 PWM 输入模式配置实例	227
图 14-26 PWM 输入模式	228
图 14-27 捕获/比较通道的输出部分 (通道 1 至 4)	228
图 14-28 向上计数下 PWM 模式 A	229
图 14-29 中央双向对齐计数下 PWM 模式 B	229
图 14-30 计数值与 C1DT 值匹配时翻转 C1ORAW	230
图 14-31 单周期模式	230
图 14-32 可再触发单周期模式 B	230
图 14-33 非对称 PWM 模式 B	231
图 14-34 组合 PWM 模式 B	231
图 14-35 EXT 清除 CxORAW(PWM 模式 B)	232
图 14-36 复位模式例子	232
图 14-37 挂起模式下例子	233
图 14-38 触发器模式例子	233
图 14-39 主/次定时器连接框图	233
图 14-40 主定时器启动次定时器例子	234
图 14-41 外部触发同时启动主、次定时器	235
图 14-42 通用定时器 TMR9 框图	251
图 14-43 通用定时器 TMR10/11 框图	252
图 14-44 计数时钟	252
图 14-45 使用 CK_INT 计数, TMRx_DIV=0x0, 周期寄存器 TMRx_PR=0x16	252
图 14-46 外部时钟模式 A 框图	253
图 14-47 使用外部时钟模式 A 计数, PR=0x32, DIV=0x0	253
图 14-48 当预分频器的参数从 1 变到 4 时, 计数器的时序图	254
图 14-49 计数器基本结构	254
图 14-50 PRBEN=0 时的溢出事件	255
图 14-51 PRBEN=1 时的溢出事件	255
图 14-52 计数器时序图, 内部时钟分频因子为 4	255
图 14-53 计数器时序图, 内部时钟分频因子为 1, TMRx_PR=0x32	255
图 14-54 向上计数模式和中央双向对齐计数模式时 OVFIF	256
图 14-55 输入/输出通道 1 的主电路	257
图 14-56 通道 1 输入部分	257
图 14-57 PWM 输入模式配置实例	258
图 14-58 PWM 输入模式	258
图 14-59 通道 1 输出部分	259
图 14-60 向上计数下 PWM 模式 A	259
图 14-61 计数值与 C1DT 值匹配时翻转 C1ORAW	260
图 14-62 单周期模式	260
图 14-63 可再触发单脉冲模式 B	261
图 14-64 组合 PWM 模式 B	261
图 14-65 带死区插入的互补输出	262
图 14-66 TMR 刹车电路	263
图 14-67 TMR 输出控制	263

图 14-68 TMR 刹车功能的例子	264
图 14-69 复位模式例子	264
图 14-70 挂起模式下例子	264
图 14-71 触发器模式例子	265
图 14-72 高级控制定时器框图	291
图 14-73 计数时钟	292
图 14-74 使用 CK_INT 计数, TMRx_DIV=0x0, 周期寄存器 TMRx_PR=0x16	292
图 14-75 外部时钟模式 A 框图	293
图 14-76 使用外部时钟模式 A 计数, PR=0x32, DIV=0x0	293
图 14-77 外部时钟模式 B 框图	293
图 14-78 使用外部时钟模式 B 计数, PR=0x32, DIV=0x0	294
图 14-79 当预分频器的参数从 1 变到 4 时, 计数器的时序图	294
图 14-80 计数器基本结构	295
图 14-81 PRBEN=0 时的溢出事件	295
图 14-82 PRBEN=1 时的溢出事件	295
图 14-83 计数器时序图, 内部时钟分频因子为 4	296
图 14-84 计数器时序图, 内部时钟分频因子为 1, TMRx_PR=0x32	296
图 14-85 向上计数模式和中央双向对齐计数模式时 OVFIF	297
图 14-86 编码模式结构	297
图 14-87 编码模式计数实例 (编码器模式 C)	298
图 14-88 输入/输出通道 1 的主电路	299
图 14-89 通道 1 输入部分	299
图 14-90 PWM 输入模式配置实例	300
图 14-91 PWM 输入模式	301
图 14-92 通道 1 至 4 输出部分	301
图 14-93 通道 5 和 6 输出部分	301
图 14-94 向上计数下 PWM 模式 A	302
图 14-95 中央双向对齐计数下 PWM 模式 B	303
图 14-96 计数值与 C1DT 值匹配时翻转 C1ORAW	303
图 14-97 单周期模式	303
图 14-98 可再触发单周期模式 B	304
图 14-99 非对称 PWM 模式 B	304
图 14-100 组合 PWM 模式 B	305
图 14-101 组合 PWM 模式 B+组合三相 PWM 模式	305
图 14-102 EXT 清除 CxORAW(PWM 模式 B)	306
图 14-103 带死区插入的互补输出	307
图 14-104 TMR 刹车电路	308
图 14-105 TMR 输出控制	308
图 14-106 TMR 停止功能的例子	309
图 14-107 复位模式例子	309
图 14-108 挂起模式下例子	310
图 14-109 触发器模式例子	310
图 15-1 窗口看门狗框图	332
图 15-2 窗口看门狗时序图	333
图 16-1 看门狗框图	336
图 17-1 ERTC 框图	339
图 18-1 ADC1 框图	354
图 18-2 ADC 基础操作流程	356
图 18-3 ADC 上电与校准	357
图 18-4 序列模式	358
图 18-5 抢占自动转换模式	359
图 18-6 反复模式	359

图 18-7 分割模式	359
图 18-8 ADABRT 时序	360
图 18-9 普通过采样重转模式选择	361
图 18-10 普通过采样触发模式	361
图 18-11 抢占过采样	362
图 18-12 数据内容处理	362
图 18-13 ADC_VM 输出	363
图 18-14 主从模式的 ADC 框图	364
图 18-15 普通同时模式	365
图 18-16 抢占同时模式	365
图 18-17 抢占交错触发模式	366
图 18-18 普通位移模式	366
图 19-1 DAC1/DAC2 模块框图	382
图 20-1 比较器的框图	388
图 20-2 H_PULSE_CNT=1 和 L_PULSE_CNT =0 时干扰滤波器时序图	390
图 20-3 H_PULSE_CNT=2 和 L_PULSE_CNT = 1 时干扰滤波器时序图	390
图 21-1 独立模式电路	396
图 21-2 跟随模式电路	397
图 21-3 可配置增益模式 (x2/x4/x8/x16/x32/x64)	398
图 21-4 可配置增益模式 (x2/x4/x8/x16/x32/x64 或 x-1/x-3/x-7/x-15/x-31/x-63)	399
图 21-5 带外部滤波器的可配置增益模式 (x2/x4/x8/x16/x32/x64)	400
图 21-6 带外部滤波器可配置增益模式 (x2/x4/x8/x16/x32/x64 或 x-1/x-3/x-7/x-15/x-31/x-63)	401
图 21-7 timer 切换功能	404
图 22-1 CAN 控制器与 CAN 总线连接示意图	414
图 22-2 CANFD 帧类型	415
图 22-3 CAN2.0 帧类型	416
图 22-4 CAN1 发送中断的产生	418
图 22-5 CAN1 接收中断的产生	418
图 22-6 CAN1 状态中断的产生	418
图 22-7 CAN1 错误中断的产生	419
图 22-8 接收过滤器示例	422
图 22-9 接收缓冲器 RB 结构图	423
图 22-10 PTB 和 STB 在 FIFO 模式下的结构图	424
图 22-11 环回模式：内部环回 (LBMI) 和外部环回 (LBME)	431
图 22-12 系统矩阵示例	434
图 22-13 时间戳和 CDC 机制	437
图 22-14 CAN 位时序规范	438
图 22-15 CANFD 帧 BRS 位比特率切换	439
图 22-16 发送器延迟	439
图 23-1 IRTMR 结构框图	455

表目录

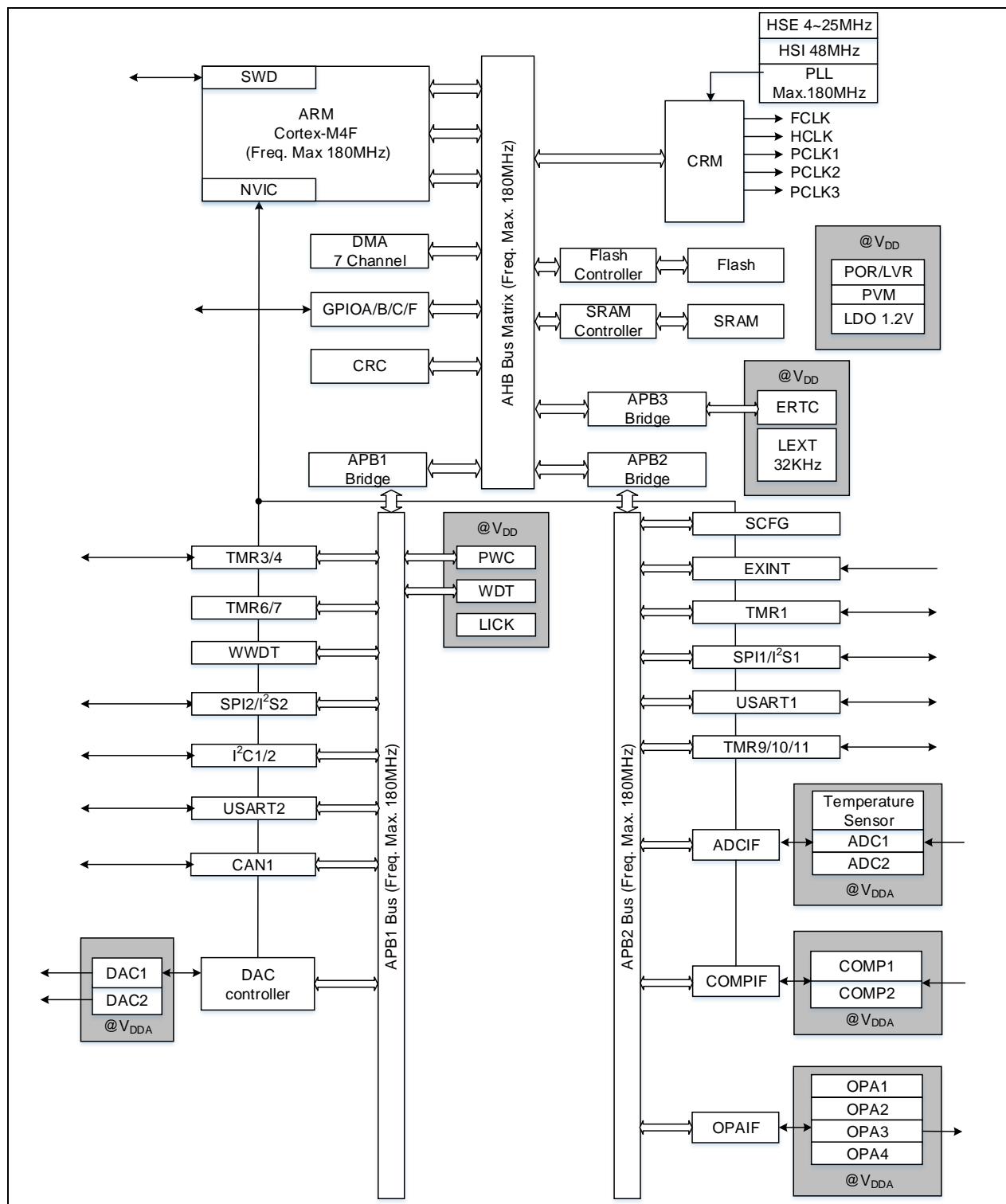
表 1-1 SRAM 区中的位带地址映射	33
表 1-2 外设区中的位带地址映射	34
表 1-3 AT32M412/AT32M416 产品的向量表	35
表 1-4 寄存器描述缩写说明	39
表 1-5 器件特征信息相关寄存器地址和复位值	40
表 2-1 128KB 闪存模块的组织	42
表 2-2 64KB 闪存模块的组织	42
表 2-3 各外设起始地址	42
表 3-1 PWC 寄存器的映像和复位值	49
表 4-1 CRM 寄存器的映像和复位值	57
表 5-1 闪存存储结构 (128K)	74
表 5-2 闪存存储结构 (64K)	74
表 5-3 用户系统数据说明	74
表 5-4 OTP 的 DATA 与 LOCK 关系	80
表 5-5 闪存访问权限	83
表 5-6 闪存接口寄存器映射和复位值	85
表 6-1 通过 GPIOA_MUX* 寄存器配置端口 A 的复用功能	97
表 6-2 通过 GPIOB_MUX* 寄存器配置端口 B 的复用功能	99
表 6-3 通过 GPIOC_MUX* 寄存器配置端口 C 的复用功能	101
表 6-4 通过 GPIOF_MUX* 寄存器配置端口 F 的复用功能	102
表 6-5 硬件抢占功能	103
表 6-6 GPIO 寄存器地址映像和复位值	104
表 7-1 SCFG 寄存器地址映像和复位值	108
表 8-1 外部中断/事件控制器寄存器映像和复位值	117
表 9-1 DMA 错误事件	121
表 9-2 DMA 中断	122
表 9-3 DMA 请求弹性映射	123
表 9-4 DMAMUX EXINT LINE 用于触发输入和同步输入	124
表 9-5 DMA 寄存器的映像和复位值	125
表 10-1 CRC 计算单元寄存器映像	137
表 11-1 I ² C 时间规范	143
表 11-2 I ² C 配置表	144
表 11-3 SMBus 超时规范	153
表 11-4 SMBus 超时检测配置	153
表 11-5 SMBus 模式配置表	153
表 11-6 I ² C 错事件	161
表 11-7 I ² C 中断请求	163
表 11-8 寄存器映像和复位值	163
表 12-1 设置波特率时的误差计算	178
表 12-2 检测起始位和噪声的数据采样	181
表 12-3 检测有效数据和噪声的数据采样	181
表 12-4 最大允许偏差	181
表 12-5 USART 中断请求	182
表 12-6 USART 寄存器映像和复位值	183
表 13-1 使用系统时钟得到精确的音频频率	204
表 13-2 SPI 寄存器列表及其复位值	207
表 14-1 TMR 功能对比	213
表 14-2 TMR6 和 TMR7 寄存器和复位值	216
表 14-3 TMRx 内部触发连接	222
表 14-4 计数方向与编码器信号的关系	225
表 14-5 TMR3 至 TMR4 寄存器图和复位值	236

表 14-6 标准 CxOUT 通道的输出控制位	245
表 14-7 TMR9 内部触发连接	254
表 14-8 TMR9 寄存器图和复位值	266
表 14-9 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位	275
表 14-10 TMR10、TMR11 寄存器和复位值	280
表 14-11 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位	286
表 14-12 TMR1 内部触发连接	294
表 14-13 计数方向与编码器信号的关系	298
表 14-14 TMR1 寄存器图和复位值	312
表 14-15 带停止功能的互补输出通道 CxOUT 和 CxCOUT 的控制位	324
表 15-1 PCLK1 频率为 180MHz 时，最大和最小看门狗超时时间	332
表 15-2 WWDT 寄存器映像和复位值	333
表 16-1 看门狗超时时间（当 LICK=40kHz 时）	336
表 16-2 WDT 寄存器映像和复位值	337
表 17-1 ERTC 寄存器配置表	340
表 17-2 ERTC 唤醒低功耗模式	343
表 17-3 中断控制位	344
表 17-4 ERTC-寄存器映像和复位值	344
表 18-1 ADC 通道输入源	355
表 18-2 普通通道与抢占通道触发来源	357
表 18-3 最大累加数据与过采样倍数及位移系数关系	360
表 18-4 主从模式 DMA 模式	364
表 18-5 ADC 寄存器映像和复位值	367
表 19-1 触发源选择	383
表 19-2 DAC 寄存器映像和复位值	384
表 20-1 比较器 1 输出映射到外部 I/O 口及外部 I/O 口复用配置	388
表 20-2 比较器 2 输出映射到外部 I/O 口及外部 I/O 口复用配置	389
表 20-3 比较器 1 和比较器 2 输入选择	389
表 20-4 CMP1 和 CMP2 寄存器映像和复位值	390
表 21-1 INP/INM/GPIO 通道	402
表 21-2 内部连接通道	402
表 21-3 INP 通道使能控制	402
表 21-4 独立模式/跟随模式通道控制	403
表 21-5 可配置增益模式通道控制	403
表 21-6 OP 寄存器地址映像和复位值	404
表 22-1 LLC 帧（逻辑链路控制帧）定义（包含时间戳）	419
表 22-2 LLC 帧缩写定义	420
表 22-3 DLC 的定义	421
表 22-4 CAN 节点错误状态	426
表 22-5 RBALL 和 KOER	427
表 22-6 TSTAT 状态编码	428
表 22-7 TSTAT 状态事件	428
表 22-8 软件复位	431
表 22-9 SPI 寄存器列表及其复位值	440
表 24-1 DEBUG 寄存器地址和复位值	456

1 系统架构

AT32M412/AT32M416 系列微控制器内部集成了：32 位 ARM ®Cortex™-M4F 处理器，多个 16 位的定时器，红外线接口 IRTMR，DMA 控制器，实时时钟 ERTC，SPI 通信接口，I²C 通信接口，USART 通信接口，CAN 总线控制器，12 位 ADC，6 位 DAC，OPA，CMP 和 PVM 模块等外设。大量的外设和存储器。Cortex™-M4F 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，并且具有单精度（IEEE-754）浮点运算单元（FPU）。系统详细架构见下图。

图 1-1 AT32M412/AT32M416 系列微控制器系统架构



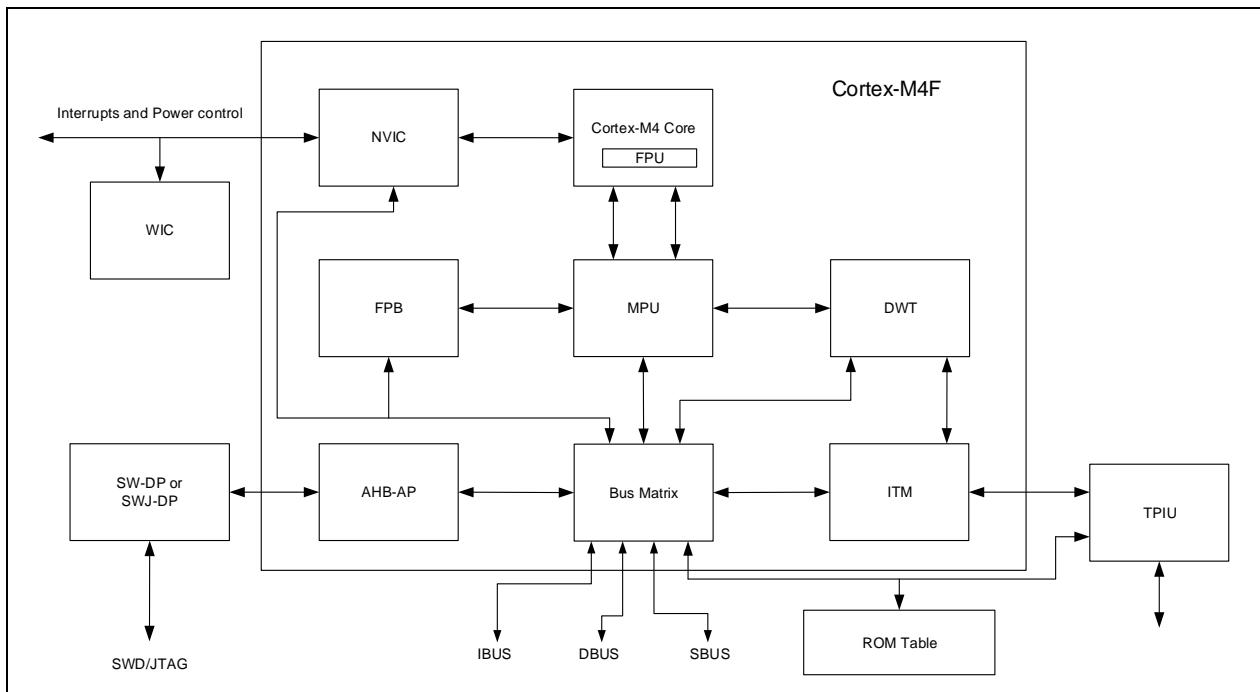
1.1 系统概述

1.1.1 ARM CortexTM-M4F处理器

CortexTM-M4F 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 指令集与浮点运算功能，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。CortexTM-M4F 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 CortexTM-M4F 处理器的内部框图，请参阅《ARM® Cortex-M4 技术参考手册》了解关于 CortexTM-M4F 更详尽信息

图 1-2 CortexTM-M4F 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 CortexTM-M4F 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

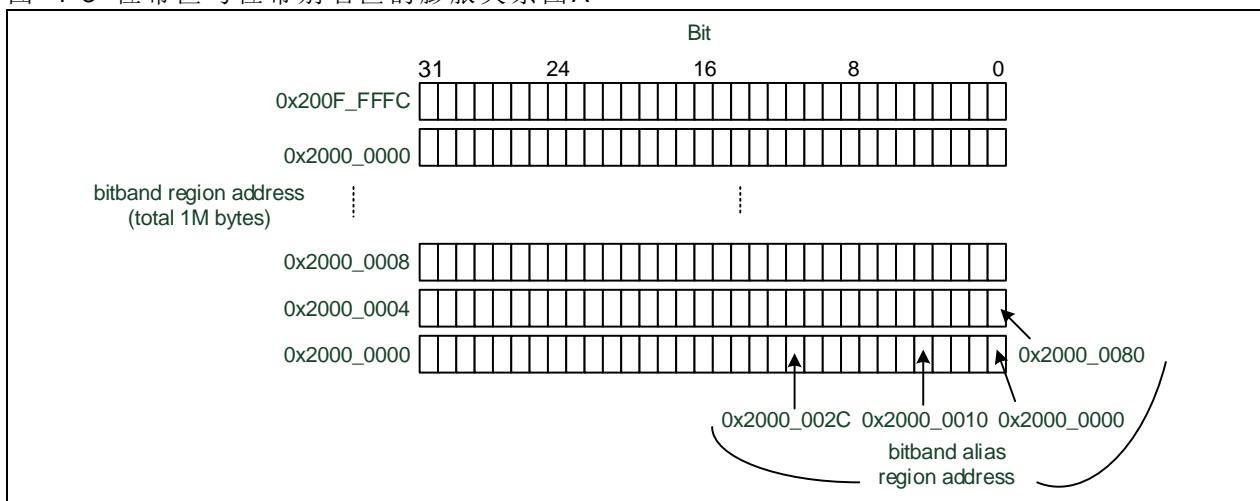
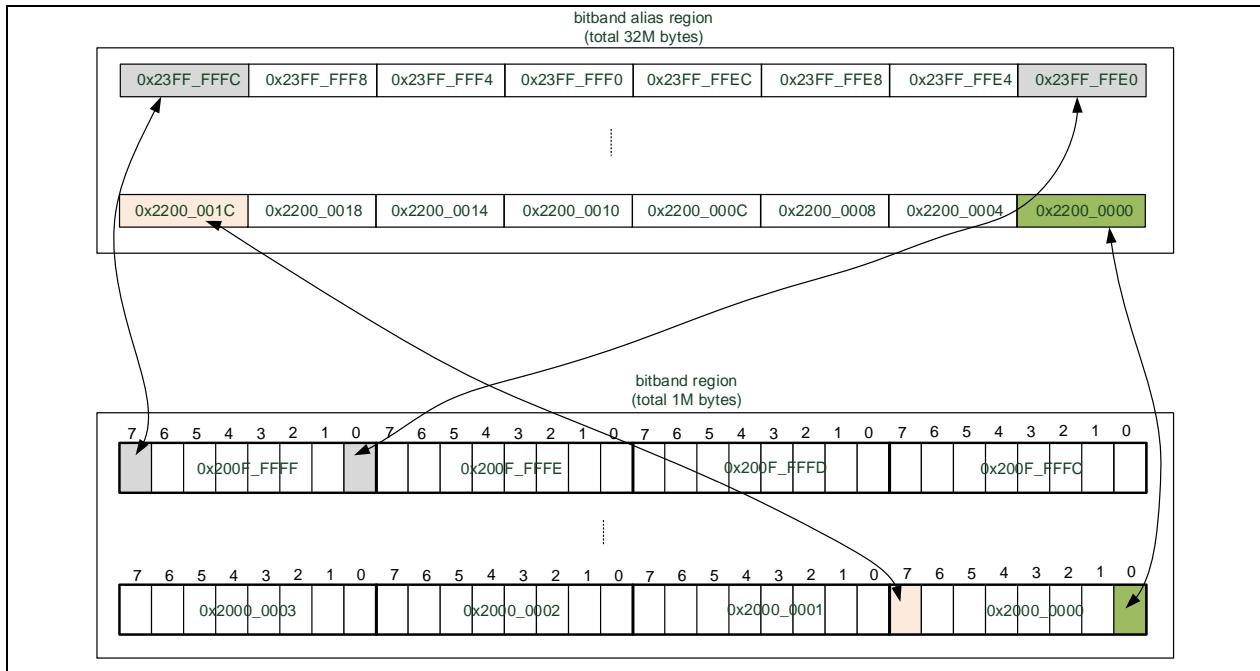


图 1-4 位带区与位带别名区的膨胀关系图B



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

位带操作的优越性最容易想到的是通过 GPIO 的管脚来单独控制每盏 LED 的点亮与熄灭。另一方面，也对操作串行接口提供很大的方便。总之，位带操作对于硬件 I/O 密集型的底层程序最有用处。

位带操作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 AT32M412/AT32M416 产品的向量表。

表 1-3 AT32M412/AT32M416 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) 连接到 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	SysCNTRick	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 的电源电压检测 (PVM) 中 断	0x0000_0044
2	9	可设置	TAMPER	侵入检测中断	0x0000_0048
3	10	可设置	ERTC_WKUP	实时时钟 (ERTC) 唤醒中断	0x0000_004C
4	11	可设置	FLASH	闪存全局中断	0x0000_0050
5	12	可设置	CRM	时钟和复位管理 (CRM) 中断	0x0000_0054
6	13	可设置	EXINT0	EXINT 线 0 中断	0x0000_0058
7	14	可设置	EXINT1	EXINT 线 1 中断	0x0000_005C
8	15	可设置	EXINT2	EXINT 线 2 中断	0x0000_0060
9	16	可设置	EXINT3	EXINT 线 3 中断	0x0000_0064
10	17	可设置	EXINT4	EXINT 线 4 中断	0x0000_0068
11	18	可设置	DMA1 通道 1	DMA1 通道 1 全局中断	0x0000_006C
12	19	可设置	DMA1 通道 2	DMA1 通道 2 全局中断	0x0000_0070
13	20	可设置	DMA1 通道 3	DMA1 通道 3 全局中断	0x0000_0074
14	21	可设置	DMA1 通道 4	DMA1 通道 4 全局中断	0x0000_0078
15	22	可设置	DMA1 通道 5	DMA1 通道 5 全局中断	0x0000_007C
16	23	可设置	DMA1 通道 6	DMA1 通道 6 全局中断	0x0000_0080

17	24	可设置	DMA1 通道 7	DMA1 通道 7 全局中断	0x0000_0084
18	25	可设置	ADC1/2	ADC 的全局中断	0x0000_0088
19	26	可设置	CAN1_TX	CAN1 发送中断	0x0000_008C
20	27	可设置	CAN1_RX	CAN1 接收中断	0x0000_0090
21	28	可设置	CAN1_STAT	CAN1 接收 1 中断	0x0000_0094
22	29	可设置	CAN1_ERR	CAN1 状态错误中断	0x0000_0098
23	30	可设置	EXINT9_5	EXINT 线[9: 5]中断	0x0000_009C
24	31	可设置	TMR1_BRK_TMR9	TMR1 停止中断和 TMR9 全局中断	0x0000_00A0
25	32	可设置	TMR1_OVF_TMR10	TMR1 溢出中断和 TMR10 全局中断	0x0000_00A4
26	33	可设置	TMR1_TRG_HALL_TM R11	TMR1 触发和 HALL 中断和 TMR11 全局中断	0x0000_00A8
27	34	可设置	TMR1_CH	TMR1 通道中断	0x0000_00AC
28	35	-	-	-	0x0000_00B0
29	36	可设置	TMR3	TMR3 全局中断	0x0000_00B4
30	37	可设置	TMR4	TMR4 全局中断	0x0000_00B8
31	38	可设置	I2C1_EVT	I ² C1 事件中断	0x0000_00BC
32	39	可设置	I2C1_ERR	I ² C1 错误中断	0x0000_00C0
33	40	可设置	I2C2_EVT	I ² C2 事件中断	0x0000_00C4
34	41	可设置	I2C2_ERR	I ² C2 错误中断	0x0000_00C8
35	42	可设置	SPI1	SPI1 全局中断	0x0000_00CC
36	43	可设置	SPI2	SPI2 全局中断	0x0000_00D0
37	44	可设置	USART1	USART1 全局中断	0x0000_00D4
38	45	可设置	USART2	USART2 全局中断	0x0000_00D8
39	46	-	-	-	0x0000_00DC
40	47	可设置	EXINT15_10	EXINT 线[15: 10]中断	0x0000_00E0
41	48	可设置	ERTCAlarm	连到 EXINT 的 ERTC 闹钟中断	0x0000_00E4
42	49	-	-	-	0x0000_00E8
43	50	-	-	-	0x0000_00EC
44	51	-	-	-	0x0000_00F0
45	52	-	-	-	0x0000_00F4
46	53	-	-	-	0x0000_00F8
47	54	-	-	-	0x0000_00FC
48	55	-	-	-	0x0000_0100
49	56	-	-	-	0x0000_0104
50	57	-	-	-	0x0000_0108
51	58	-	-	-	0x0000_010C

52	59	-	-	-	0x0000_0110
53	60	-	-	-	0x0000_0114
54	61	可设置	TMR6_DAC	TMR6 全局中断 DAC1 和 DAC2 下溢错误中断	0x0000_0118
55	62	可设置	TMR7	TMR7全局中断	0x0000_011C
56	63	-	-	-	0x0000_0120
57	64	-	-	-	0x0000_0124
58	65	-	-	-	0x0000_0128
59	66	-	-	-	0x0000_012C
60	67	-	-	-	0x0000_0130
61	68	-	-	-	0x0000_0134
62	69	-	-	-	0x0000_0138
63	70	-	-	-	0x0000_013C
64	71	可设置	CMP1/2	比较器中断	0x0000_0140
65	72	-	-	-	0x0000_0144
66	73	-	-	-	0x0000_0148
67	74	-	-	-	0x0000_014C
68	75	-	-	-	0x0000_0150
69	76	-	-	-	0x0000_0154
70	77	-	-	-	0x0000_0158
71	78	-	-	-	0x0000_015C
72	79	-	-	-	0x0000_0160
73	80	-	-	-	0x0000_0164
74	81	-	-	-	0x0000_0168
75	82	-	-	-	0x0000_016C
76	83	-	-	-	0x0000_0170
77	84	-	-	-	0x0000_0174
78	85	-	-	-	0x0000_0178
79	86	-	-	-	0x0000_017C
80	87	-	-	-	0x0000_0180
81	88	可设置	FPU	FPU例外中断	0x0000_0184
82	89	-	-	-	0x0000_0188
83	90	-	-	-	0x0000_018C
84	91	-	-	-	0x0000_0190
85	92	-	-	-	0x0000_0194
86	93	-	-	-	0x0000_0198

87	94	-	-	-	0x0000_019C
88	95	-	-	-	0x0000_01A0
89	96	-	-	-	0x0000_01A4
90	97	-	-	-	0x0000_01A8
91	98	-	-	-	0x0000_01AC
92	99	-	-	-	0x0000_01B0
93	100	-	-	-	0x0000_01B4
94	101	可设置	DMAMUX	DMAMUX溢出中断	0x0000_01B8
95	102	-	-	-	0x0000_01BC
96	103	-	-	-	0x0000_01C0
97	104	-	-	-	0x0000_01C4
98	105	-	-	-	0x0000_01C8
99	106	-	-	-	0x0000_01CC
100	107	-	-	-	0x0000_01D0
101	108	-	-	-	0x0000_01D4
102	109	-	-	-	0x0000_01D8
103	110	-	-	-	0x0000_01DC

1.1.4 系统嘀嗒定时器 (SysTick)

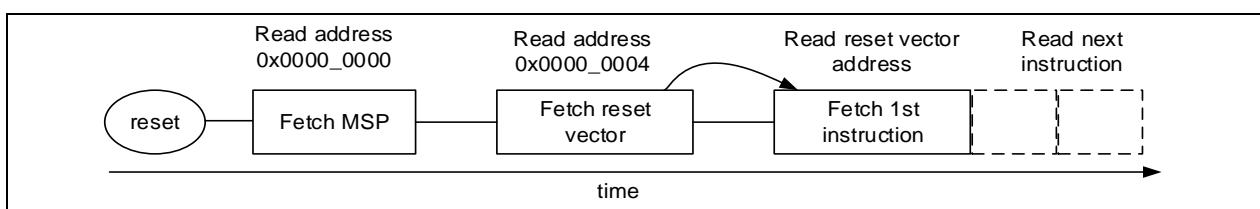
系统嘀嗒定时器是一个 24 位递减计数器，递减至零可自动重载计数初值。可产生周期性异常，用作嵌入式操作系统的多任务调度计数器，或对于无嵌入式操作系统，可用于调用需周期性执行的任务。系统嘀嗒定时器校准值固定值 9000，当系统嘀嗒时钟设定为 9MHz，产生 1ms 时间基准。

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。

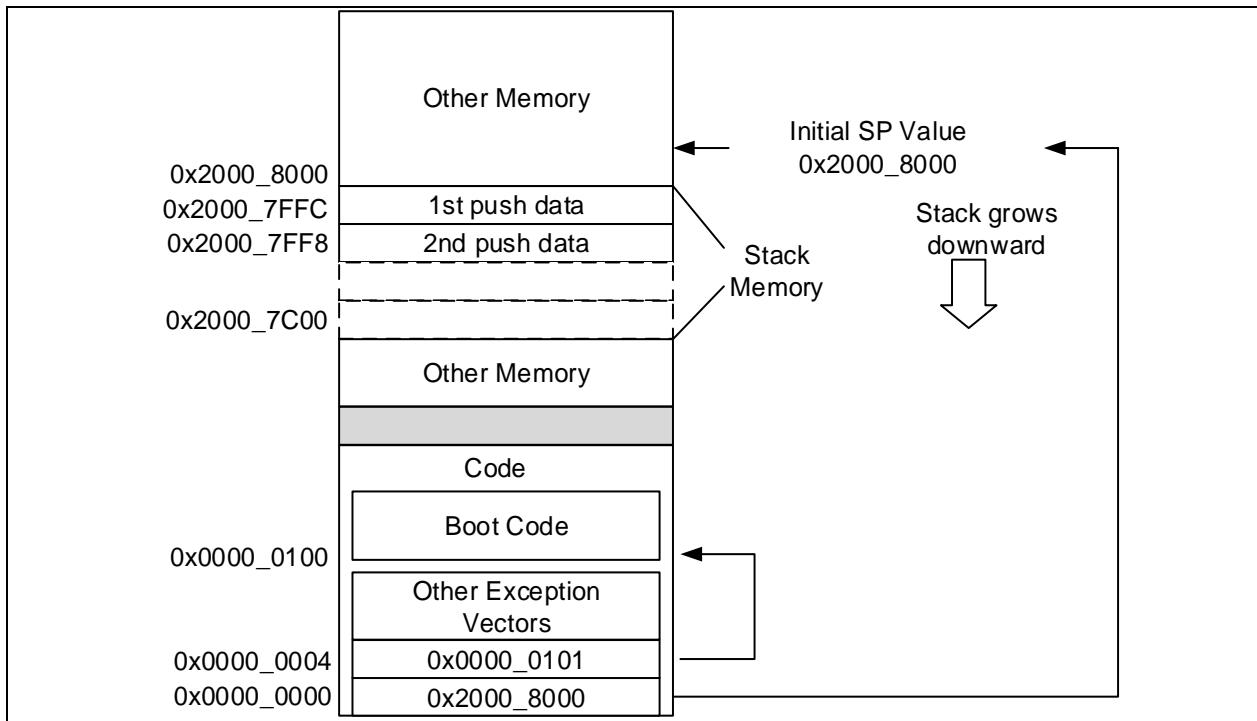
图 1-5 复位流程



Cortex™-M4F 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™-M4F 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 AT32M412/AT32M416 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、USART2 或 USB 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]	0xFFFF 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]	0xFFFF 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]	0xFFFF XXXX	ro	UID 的 bit95 到 bit64 信息

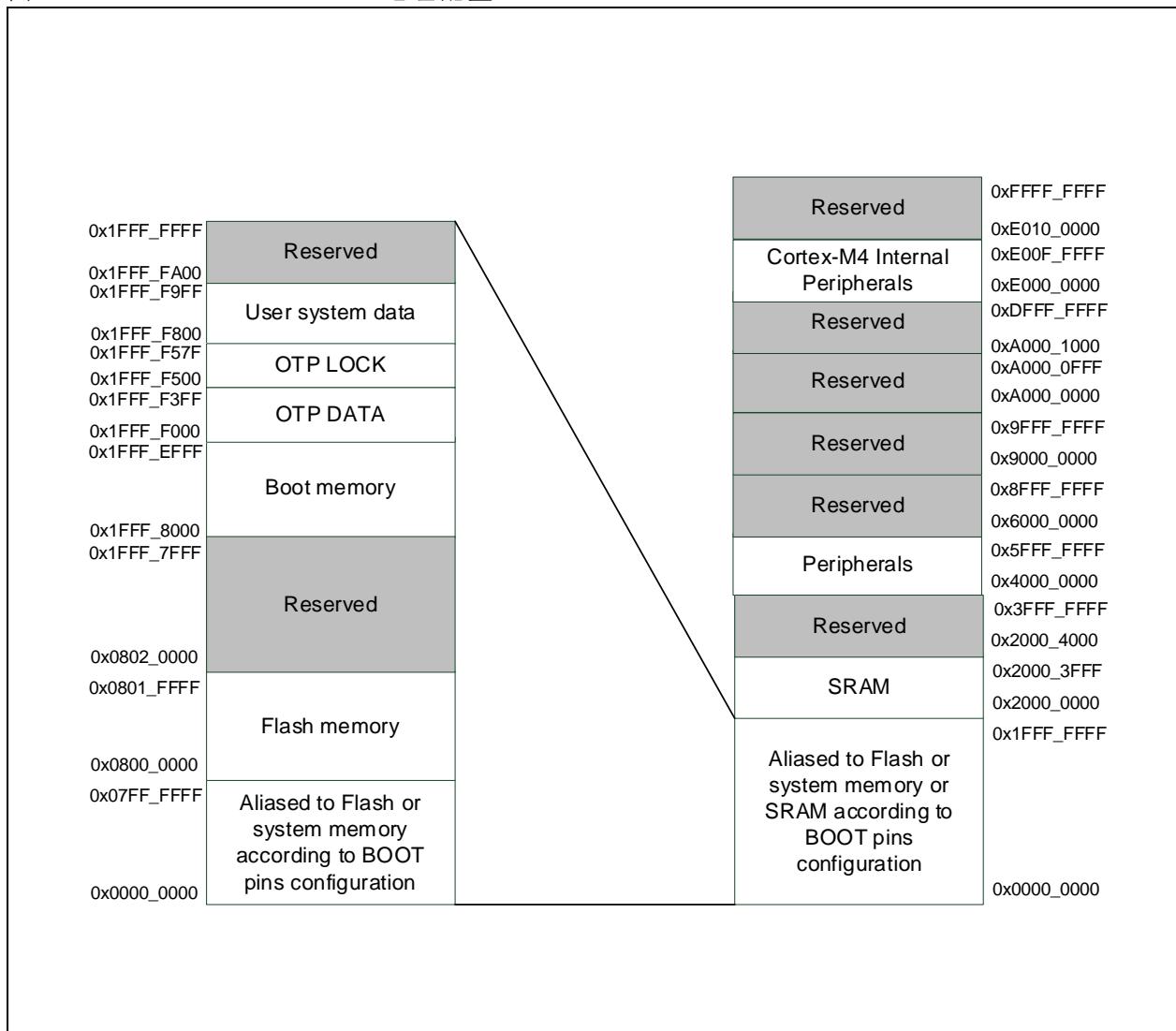
注：UID[95:88]为 Series ID，M412 为 0x19，M416 为 0x1A。

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 AT32M412/AT32M416地址配置



2.2 Flash存储器

AT32M412/AT32M416 系列提供最大 128KB 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。

闪存存储结构 (128K)

主存储器只有闪存容量为 128K 字节的片 1 闪存，包含 128 个扇区，每扇区大小为 1K 字节。

表 2-1 128KB 闪存模块的组织

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 3	0x0800 0C00 – 0x0800 0FFF
	扇区 4	0x0800 1000 – 0x0800 13FF
	.	.
	扇区 127	0x0801 FC00 – 0x0801 FFFF
	启动程序代码区 28KB	0x1FFF 8000 – 0x1FFF EFFF
	OTP DATA 1KB	0x1FFF F000 – 0x1FFF F3FF
	OTP LOCK 128B	0x1FFF F500 – 0x1FFF F57F
	用户系统数据区 512B	0x1FFF F800 – 0x1FFF F9FF

闪存存储结构 (64K)

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 2-2 64KB 闪存模块的组织

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 3	0x0800 0C00 – 0x0800 0FFF
	扇区 4	0x0800 1000 – 0x0800 13FF
	.	.
	扇区 63	0x0800 FC00 – 0x0800 FFFF
	启动程序代码区 28KB	0x1FFF 8000 – 0x1FFF EFFF
	OTP DATA 1KB	0x1FFF F000 – 0x1FFF F3FF
	OTP LOCK 128B	0x1FFF F500 – 0x1FFF F57F
	用户系统数据区 512B	0x1FFF F800 – 0x1FFF F9FF

2.3 SRAM 存储器

AT32M412/AT32M416 系列内置最高可达 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 2-3 各外设起始地址

总线	起始地址	外设
AHB	0xC000 0000 - 0xFFFF FFFF	保留
	0xB000 0000 - 0xBFFF FFFF	保留
	0xA000 1000 - 0xAF00 FFFF	保留
	0xA000 0000 - 0xA000 0FFF	保留
	0x9000 0000 - 0x9FFF FFFF	保留
	0x6000 0000 - 0x8FFF FFFF	保留

	0x5004 0000 - 0x5FFF FFFF	保留
	0x5000 0000 - 0x5003 FFFF	保留
	0x4002 6800 - 0x4FFF FFFF	保留
	0x4002 6400 - 0x4002 67FF	保留
	0x4002 6000 - 0x4002 63FF	DMA1
	0x4002 4000 - 0x4002 5FFF	保留
	0x4002 3C00 - 0x4002 3FFF	闪存存储器接口 (FLASH)
	0x4002 3800 - 0x4002 3BFF	时钟和复位管理 (CRM)
	0x4002 3400 - 0x4002 37FF	保留
	0x4002 3000 - 0x4002 33FF	CRC
	0x4002 2000 - 0x4002 2FFF	保留
	0x4002 1C00 - 0x4002 1FFF	保留
	0x4002 1800 - 0x4002 1BFF	保留
	0x4002 1400 - 0x4002 17FF	GPIO 端口 F
	0x4002 1000 - 0x4002 13FF	保留
	0x4002 0C00 - 0x4002 0FFF	保留
	0x4002 0800 - 0x4002 0BFF	GPIO 端口 C
	0x4002 0400 - 0x4002 07FF	GPIO 端口 B
	0x4002 0000 - 0x4002 03FF	GPIO 端口 A
APB3	0x4000 2800 - 0x4000 2BFF	ERTC
	0x4001 8000 - 0x4001 FFFF	保留
	0x4001 7C00 - 0x4001 7FFF	保留
	0x4001 7800 - 0x4001 7BFF	保留
	0x4001 7400 - 0x4001 77FF	保留
	0x4001 4C00 - 0x4001 73FF	保留
	0x4001 4800 - 0x4001 4BFF	TMR11 定时器
	0x4001 4400 - 0x4001 47FF	TMR10 定时器
	0x4001 4000 - 0x4001 43FF	TMR9 定时器
	0x4001 3C00 - 0x4001 3FFF	EXINT
	0x4001 3800 - 0x4001 3BFF	SCFG/CMP/OP
APB2	0x4001 3400 - 0x4001 37FF	保留
	0x4001 3000 - 0x4001 33FF	SPI1/I2S1
	0x4001 2400 - 0x4001 2FFF	保留
	0x4001 2000 - 0x4001 23FF	ADC
	0x4001 1800 - 0x4001 1FFF	保留
	0x4001 1400 - 0x4001 17FF	保留
	0x4001 1000 - 0x4001 13FF	USART1
	0x4001 0C00 - 0x4001 0FFF	保留
	0x4001 0800 - 0x4001 0BFF	保留
	0x4001 0400 - 0x4001 07FF	保留
	0x4001 0000 - 0x4001 03FF	TMR1 定时器

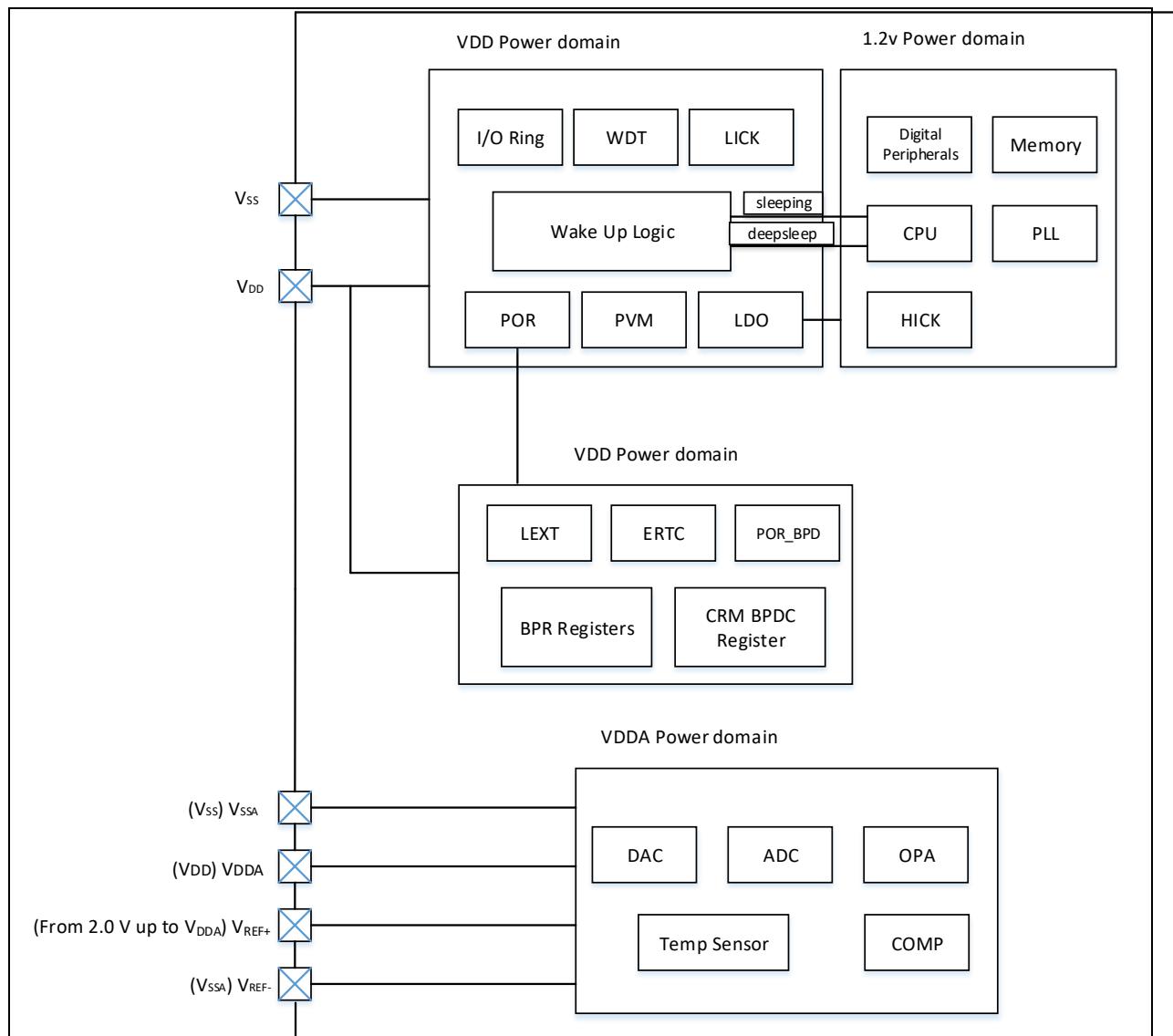
APB1	0x4000 8000 - 0x4000 FFFF	保留
	0x4000 7C00 - 0x4000 7FFF	保留
	0x4000 7800 - 0x4000 7BFF	保留
	0x4000 7400 - 0x4000 77FF	DAC
	0x4000 7000 - 0x4000 73FF	电源控制 (PWC)
	0x4000 6C00 - 0x4000 6FFF	保留
	0x4000 6800 - 0x4000 6BFF	保留
	0x4000 6400 - 0x4000 67FF	CAN1
	0x4000 6000 - 0x4000 63FF	保留
	0x4000 5C00 - 0x4000 5FFF	保留
	0x4000 5800 - 0x4000 5BFF	I ² C2
	0x4000 5400 - 0x4000 57FF	I ² 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/I2S2
	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 2400 - 0x4000 27FF	保留
	0x4000 2000 - 0x4000 23FF	保留
	0x4000 1C00 - 0x4000 1FFF	保留
	0x4000 1800 - 0x4000 1BFF	保留
	0x4000 1400 - 0x4000 17FF	TMR7 定时器
	0x4000 1000 - 0x4000 13FF	TMR6 定时器
	0x4000 0C00 - 0x4000 0FFF	保留
	0x4000 0800 - 0x4000 0BFF	TMR4 定时器
	0x4000 0400 - 0x4000 07FF	TMR3 定时器
	0x4000 0000 - 0x4000 03FF	保留

3 电源控制 (PWC)

3.1 简介

AT32M412/AT32M416 系列设备工作电压范围为 2.4V 至 3.6V，正常工作温度范围为 -40~+105°C。AT32M412/AT32M416 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。AT32M412/AT32M416 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



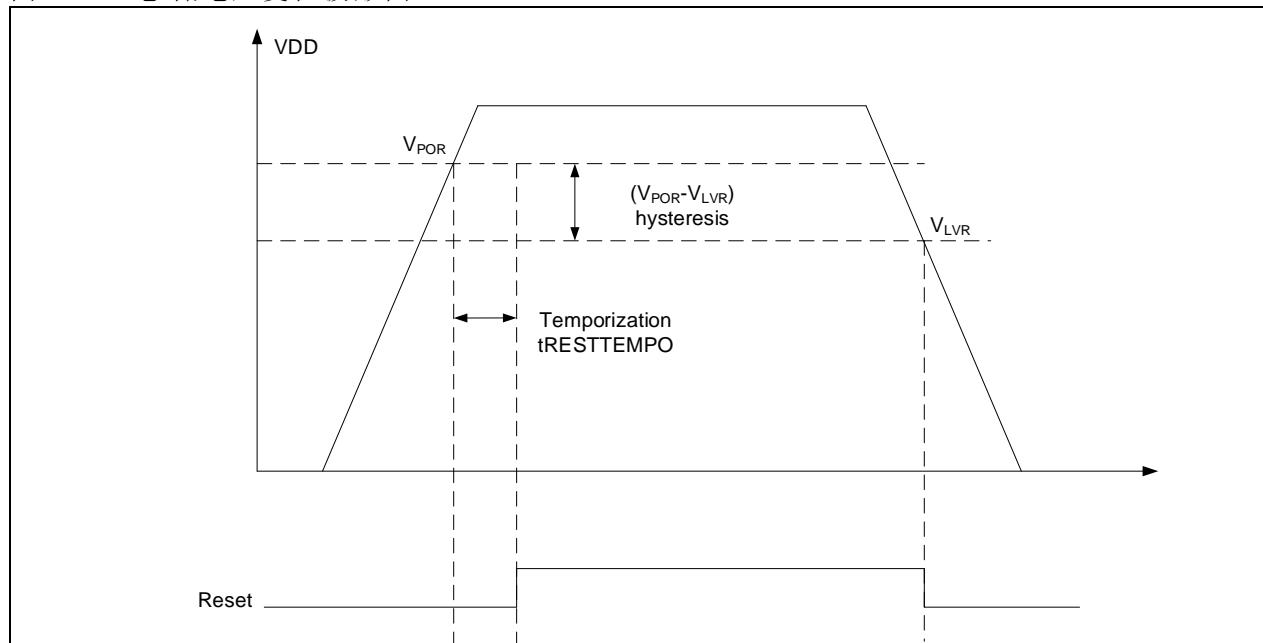
3.2 主要特点

- 具备两个电源域：VDD/VDDA 域、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，能在电压低于阈值或高于阈值时产生中断或事件。
- 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} 时刻被低电压复位。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，同时上电低电压复位具有一定迟滞。

图 3-2 上电/低电压复位波形图

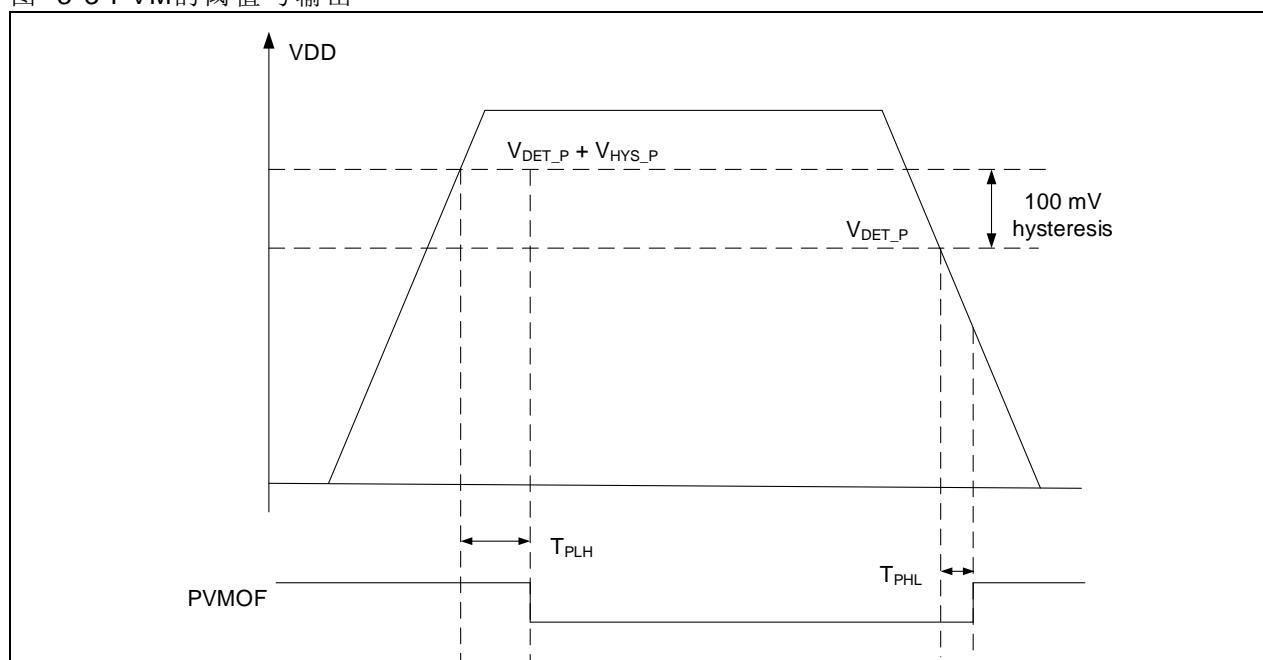


3.4 电压监测器 (PVM)

电压监测器 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 位开启电压监测功能，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/低电压复位 (POR/LVR)、电压调节器 LDO、ERTC 电路、LEXT 振荡器以及所有 PAD 电路等。VDDA 域包括 DAC/ADC (DA/AD 转换器)、温度传感器 Temp Sensor 等。

一般来说，为保证低电压时 ADC/DAC 的高精度，数字电路由 VDD 供电，模拟电路由 VDDA 独立供电，在 64 PIN 封装及以下型号中，外部参考电压 VREF+连接至 VDDA 管脚，VREF-连接至 VSSA 管脚。

在芯片正常工作 (Run Mode) 时，电压调节器 (LDO) 为 1.2V 域供电。可配置 LDO 调校寄存器 (PWC_LDOOV) 来选择 LDO 的输出电压，不同的输出电压，系统可以工作的最高工作频率不同，详细参见 AT32M412/AT32M416 数据手册。

注意：限定在系统时钟使用 HEXT 或 HICK 时方可改变 LDO 的输出电压。

LDO 输出电压调节

- 1) 系统时钟切换至 HICK 或 HEXT
- 2) 修改 LDO 电压 (LDOOVSEL[1: 0])
- 3) 设置闪存性能选择寄存器 (FLASH_PSR)
- 4) 设置 PLL 相关寄存器至目标频率，开启 PLL，等待 PLL_STBL
- 5) 设置 AHB 及 APB 预除频系数
- 6) 若 PLL 频率大于 108MHz，打开顺滑切换
- 7) 切换系统时钟至 PLL

3.6 省电模式

当 CPU 无需继续运行时，AT32M412/AT32M416 支持三种低功耗模式（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。结合 Cortex™-M4F 系统控制寄存器中的 SLEEPONEXIT 位的设定，提供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电源 (CPU 内核、内存和内嵌外设)，由 LDO 调校寄存器 (PWC_LDOOV) 来配置 LDO 的输出电压。

- 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™-M4F 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。

低功耗 deepsleep LDO 电压调节流程（注：sleep 和 standby 没有此限制）

- 1) 系统时钟切换至 HICK
- 2) 修改 LDO 电压 (LDOOVSEL[1: 0]) 为 1.0V 以及 VREXLPE 位
- 3) 配置 LDO 是否工作在低功耗模式 (VRSEL)
- 4) 系统进入 deepsleep 状态
- 5) 系统退出 deepsleep 状态（满足唤醒条件后）
- 6) 修改 LDO 电压 (LDOOVSEL[1: 0])
- 7) 设置闪存性能选择寄存器 (FLASH_PSR)
- 8) 若 PLL 时钟源为 HEXT，需要开启 HEXT 并等待 HEXTSTBL
- 9) 设置 PLL 相关寄存器至目标频率
- 10) 开启 PLL，等待 PLL_STBL
- 11) 设置 AHB 及 APB 预除频系数
- 12) 若 PLL 频率大于 108MHz，打开顺滑切换
- 13) 切换系统时钟至 PLL

注：若期望低功耗唤醒后保持进低功耗前的时钟状态，前述 7/9/11 步骤可省略。

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™-M4F 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) 中的所有唤醒事件标志位的情况下，最后执行 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 管脚和被使能的唤醒管脚之外，所有的 I/O 管脚处于高阻态。

当产生 WKUP 管脚的有效沿、ERTC 闹钟事件的上升沿、ERTC 入侵事件、ERTC 时间戳、ERTC 周期性自动唤醒、NRST 管脚上外部复位、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™-M4F 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

可以用半字（16位）或字（32位）的方式操作这些外设寄存器。

表 3-1 PWC 寄存器的映像和复位值

寄存器简称	基址偏移量	复位值
PWC_CTRL	0x00	0x0000 0000
PWC_CTRLSTS	0x04	0x0000 0000
PWC_CLR	0x08	0x0000 0000
PWC_LDOOV	0x10	0x0000 0012

3.7.1 电源控制寄存器 (PWC_CTRL)

域	简称	复位值	类型	功能
位 31	保留	0x0000000	resd	保持默认值。
位 30	SWP7POL	0x0	rw	待机唤醒引脚 7 极性选择 (Standby wake-up pin7 polarity) 0: 上升沿触发 1: 下降沿触发
位 29	SWP6POL	0x0	rw	待机唤醒引脚 6 极性选择 (Standby wake-up pin6 polarity) 0: 上升沿触发 1: 下降沿触发
位 28: 26	保留	0x00	resd	保持默认值。
位 25	SWP2POL	0x0	rw	待机唤醒引脚 2 极性选择 (Standby wake-up pin2 polarity) 0: 上升沿触发 1: 下降沿触发
位 24	SWP1POL	0x0	rw	待机唤醒引脚 1 极性选择 (Standby wake-up pin1 polarity) 0: 上升沿触发 1: 下降沿触发
位 23	保留	0x0	resd	保持默认值。
位 22	SWPEN7	0x0	rw	待机唤醒引脚 7 使能 (Standby wake-up pin7 enable) 0: 关闭 (该引脚可用作通用 I/O) ; 1: 开启 (该引脚被强置为输入下拉/上拉模式, 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。
位 21	SWPEN6	0x0	rw	待机唤醒引脚 6 使能 (Standby wake-up pin6 enable) 0: 关闭 (该引脚可用作通用 I/O) ; 1: 开启 (该引脚被强置为输入下拉/上拉模式, 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。
位 20: 18	保留	0x00	resd	保持默认值。
位 17	SWPEN2	0x0	rw	待机唤醒引脚 2 使能 (Standby wake-up pin2 enable) 0: 关闭 (该引脚可用作通用 I/O) ; 1: 开启 (该引脚被强置为输入下拉/上拉模式, 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。
位 16	SWPEN1	0x0	rw	待机唤醒引脚 1 使能 (Standby wake-up pin1 enable) 0: 关闭 (该引脚可用作通用 I/O) ; 1: 开启 (该引脚被强置为输入下拉/上拉模式, 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。

位 15: 9	保留	0x0	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	保持默认值。
位 3: 2	保留	0x0	resd	SLEEPDEEP 状态下的低功耗模式选择位 (Low power mode select when Cortex™-M4F sleepdeep) 0: 进入 DEEPSLEEP 模式; 1: 进入待机模式。
位 1	LPSEL	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。
位 0	VRSEL	0x0	rw	保持默认值。

3.7.2 电源控制及状态寄存器 (PWC_CTRLSTS)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0000 0	resd	保持默认值。
位 15	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 将其清零或写 CLSWEF 位将其清零。 唤醒事件将由以下几种情况产生: 当产生使能中断的 RTC 闹钟/入侵/时间戳/周期性自动唤醒事件时, 将对应置位此标志。
位 14: 10	保留	0x0	resd	保持默认值。
位 9	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 8	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 7	保留	0x0	resd	保持默认值。 待机唤醒事件 7 标志 (Standby wake-up7 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 CLSW7EF 位将其清零。 唤醒事件将由以下几种情况产生: 在待机唤醒引脚 7 出现上升/下降沿时, 将产生唤醒事件; 当引脚极性配置为上升沿触发时, 待机唤醒引脚保持高电平期间使能该待机唤醒引脚, 将产生唤醒事件。
位 6	SW7EF	0x0	ro	待机唤醒事件 6 标志 (Standby wake-up6 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 CLSW6EF 位将其清零。 唤醒事件将由以下几种情况产生: 在待机唤醒引脚 6 出现上升/下降沿时, 将产生唤醒事件; 当引脚极性配置为上升沿触发时, 待机唤醒引脚保持高电平期间使能该待机唤醒引脚, 将产生唤醒事件。
位 5	SW6EF	0x0	ro	待机唤醒事件 5 标志 (Standby wake-up5 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 CLSW5EF 位将其清零。 唤醒事件将由以下几种情况产生: 在待机唤醒引脚 5 出现上升/下降沿时, 将产生唤醒事件; 当引脚极性配置为上升沿触发时, 待机唤醒引脚保持高电平期间使能该待机唤醒引脚, 将产生唤醒事件。
位 4: 2	保留	0x00	resd	保持默认值。 待机唤醒事件 2 标志 (Standby wake-up2 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 CLSW2EF 位将其清零。 唤醒事件将由以下几种情况产生: 在待机唤醒引脚 2 出现上升/下降沿时, 将产生唤醒事件; 当引脚极性配置为上升沿触发时, 待机唤醒引脚保持高电平期间使能该待机唤醒引脚, 将产生唤醒事件。
位 1	SW2EF	0x0	ro	待机唤醒事件 1 标志 (Standby wake-up1 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 CLSW1EF 位将其清零。 唤醒事件将由以下几种情况产生: 在待机唤醒引脚 1 出现上升/下降沿时, 将产生唤醒事件; 当引脚极性配置为上升沿触发时, 待机唤醒引脚保持高电平期间使能该待机唤醒引脚, 将产生唤醒事件。
位 0	SW1EF	0x0	ro	待机唤醒事件 0 标志 (Standby wake-up0 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 CLSW0EF 位将其清零。 唤醒事件将由以下几种情况产生: 在待机唤醒引脚 0 出现上升/下降沿时, 将产生唤醒事件; 当引脚极性配置为上升沿触发时, 待机唤醒引脚保持高电平期间使能该待机唤醒引脚, 将产生唤醒事件。

3.7.3 电源标志清除寄存器 (PWC_CLR)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0000	resd	保持默认值。 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, 且任何时候读取该位返回值均是零。
位 15	CLSWEF	0x0	wo	
位 14: 9	保留	0x00	resd	保持默认值。
位 8	CLSEF	0x0	wo	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, 且任何时候读取该位返回值均是零。
位 7	保留	0x0	resd	保持默认值。
位 6	CLSW7EF	0x0	wo	清除 SW7EF 标志 (Clear SW7EF flag) 0: 无效; 1: 清除 SW7EF 标志。 注: 实际 SW7EF 标志的清除大约需要 2 个系统时钟周期; 该位在清除 SW7EF 后由硬件将其清零, 且任何时候读取该位返回值均是零。
位 5	CLSW6EF	0x0	wo	清除 SW6EF 标志 (Clear SW6EF flag) 0: 无效; 1: 清除 SW6EF 标志。 注: 实际 SW6EF 标志的清除大约需要 2 个系统时钟周期; 该位在清除 SW6EF 后由硬件将其清零, 且任何时候读取该位返回值均是零。
位 4: 2	保留	0x0	resd	保持默认值。
位 1	CLSW2EF	0x0	wo	清除 SW2EF 标志 (Clear SW2EF flag) 0: 无效; 1: 清除 SW2EF 标志。 注: 实际 SW2EF 标志的清除大约需要 2 个系统时钟周期; 该位在清除 SW2EF 后由硬件将其清零, 且任何时候读取该位返回值均是零。
位 0	CLSW1EF	0x0	wo	清除 SW1EF 标志 (Clear SW1EF flag) 0: 无效; 1: 清除 SW1EF 标志。 注: 实际 SW1EF 标志的清除大约需要 2 个系统时钟周期; 该位在清除 SW1EF 后由硬件将其清零, 且任何时候读取该位返回值均是零。

3.7.4 LDO调校寄存器 (PWC_LDOOV)

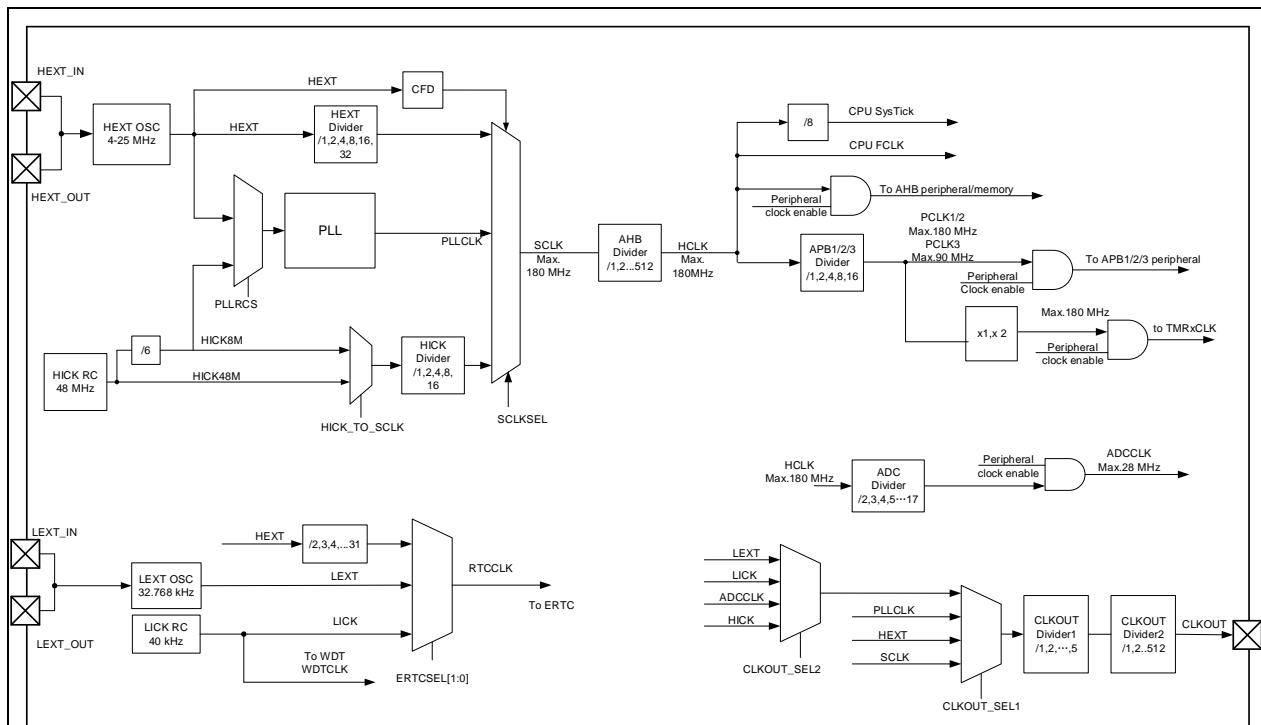
域	简称	复位值	类型	功能
位 31: 5	保留	0x0000000	resd	保持默认值。 内部电压调节器额外低功耗模式使能位 (Voltage regulator extra low power mode enable) 与电源控制寄存器 (PWC_CTRL) 的 LPSEL 和 VRSEL 位协同工作，在 VRSEL = 1 时，且芯片进入深度睡眠模式 时才有效。
位 4	VREXL PEN	0x1	rw	0: 内部电压调节器额外低功耗模式关闭。 1: 内部电压调节器额外低功耗模式开启。 注： 如需启动额外低功耗模式，请先配置 VREXL PEN，再配置 LPSEL 和 VRSEL 位。 开启此位并进入 DEEPSLEEP 模式后，LDO 驱动能力受限，此时禁止开启 DEEPSLEEP_DEBUG 进行调试。
位 3: 2	保留	0x0	resd	保持默认值。
位 1: 0	LDOOVSEL	0x2	rw	LDOOVSEL: 内部电压调节器 LDO 输出 电压选择 (LDO output voltage select) 00: 1.0V 01: 保留 10: 1.2V 11: 1.3V

4 时钟和复位管理 (CRM)

4.1 时钟

AT32M412/AT32M416 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 AT32M412/AT32M416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、APB3、APB2 和 APB1 的频率都支持多种分频。AHB 域的最大频率是 180MHz，APB1 域的最大允许频率是 180MHz，APB2 域的最大允许频率是 180MHz，APB3 域的最大允许频率是 90MHz。

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 引脚，HEXT_OUT 引脚可以释放给 GPIO 控制。

- 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。

当 HICK 作为系统时钟时，去更改 CRM_MISC1 寄存器中的 HICK_TO_SCLK 位，需要遵循如下的配置流程：

1. 配置 CRM_MISC2 寄存器中的 HICK_TO_SCLK_DIV 位为 0x4

2. 等待 50 个系统时钟
3. 更改 CRM_MISC1 寄存器中的 HICK_TO_SCLK 值
4. 重新配置 HICK_TO_SCLK_DIV 位为需要设定的值

● PLL 时钟

PLL 可以选择 HICK 时钟或 HEXT 时钟作为输入时钟，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 2MHz~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 公式如下：

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

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

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

例如：当 PLL 输入时钟为 25 MHz 时，可以配置 PLL 输出频率= $25 \times 144 / (5 \times 4) = 180 \text{ MHz}$

● 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 引脚，并且 LEXT_OUT 引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 CPU FCLK (HCLK) 或从 CPU Systick (HCLK 的 8 分频) 作为时钟。

ADC 使用 HCLK 或是 PLL 时钟的 2、3、4、5…17 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 振荡器分频时钟，LEXT 振荡器时钟和 LICK 振荡器时钟。ERTC 的时钟源一旦选择后就不可再更改，只有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 TMR1/9/10/11 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

当系统时钟源切换或是 AHB 分频因子更改时，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

微控制器允许输出时钟信号到外部 CLKOUT 引脚。ADCCLK、SCLK、LICK、LEXT、HICK、HEXT、PLLCLK 等时钟信号可作为 CLKOUT 时钟。作为 CLKOUT 时钟输出脚时，相应的 GPIO 端口寄存器必须被配置为相应功能。

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

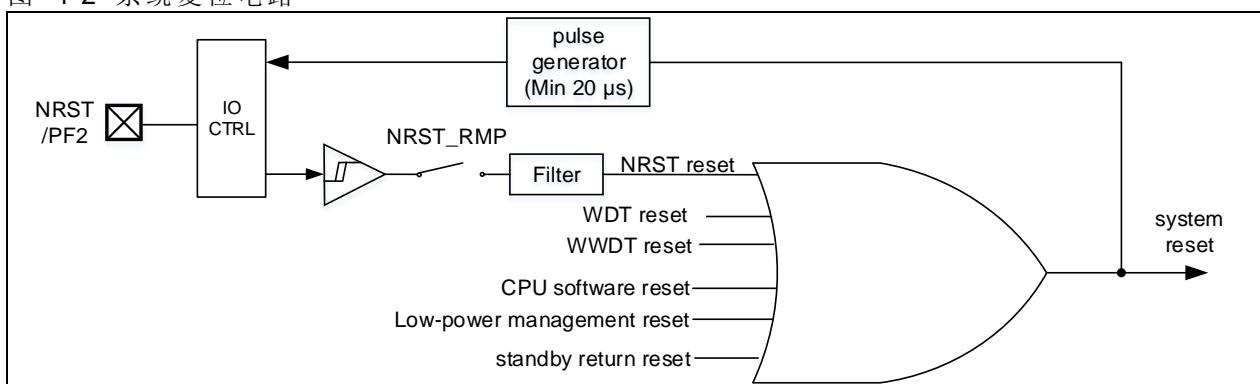
AT32M412/AT32M416 系统复位包括以下复位源：

- 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™-M4F 软件复位
- 低功耗管理复位：将用户系统数据中的 nSTDBY_RST 位清 0 并进入待机模式，将产生低功耗管理复位；将用户系统数据中的 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；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。

其中 SCFG_IOCFG.NRST_RMP 用于控制 NRST/PF2 作为 NRST 还是 PF2 使用。上电复位后 NRST/PF2 默认作为 NRST 使用，因此在上电阶段需要保证 NRST/PF2 为高电平。NRST/PF2 作为 PF2 使用时建议只使用推挽输出功能。

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寄存器描述

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_PLLCFG	0x004	0x0003 3002
CRM_CFG	0x008	0x4000 0000
CRM_CLKINT	0x00C	0x0000 0000
CRM_AHBRST1	0x010	0x0000 0000
CRM_APB1RST	0x020	0x0000 0000
CRM_APB2RST	0x024	0x0000 0000
CRM_AHBEN1	0x030	0x0000 0000
CRM_APB1EN	0x040	0x0000 0000
CRM_APB2EN	0x044	0x0000 0000
CRM_AHBLPEN1	0x050	0x0041 9027
CRM_APB1LPEN	0x060	0x3262 4836
CRM_APB2LPEN	0x064	0x0007 5311
CRM_PICLKS	0x068	0x0000 0000
CRM_BPDC	0x070	0x0000 0000
CRM_CTRLSTS	0x074	0x0C00 0000
CRM_PLLRESVD	0x07C	0x0000 0004
CRM_MISC1	0xA0	0x000F 1000
CRM_MISC2	0xA4	0x0380 500D

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 31: 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	HEXTBYP.S	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 PLL时钟配置寄存器 (CRM_PLLCFG)

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	请保持为复位值。
位 22	PLLRCS	0x0	rw	PLL 输入时钟选择 (PLL reference clock select) 由软件置'1'或清'0'来选择 PLL 输入时钟源。只能在关闭 PLL 时才能写入此位。 0: HICK 时钟作为 PLL 输入时钟； 1: HEXT 时钟作为 PLL 输入时钟源。
位 21: 19	保留	0x0	resd	请保持为复位值。
位 18: 16	PLL_FR	0x3	rw	PLL 后分频配置值 (PLL post-division) 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
位 15	保留	0x0	resd	请保持为复位值。
位 14: 6	PLL_NS	0x0C0	rw	PLL 倍频系数(PLL Multiplication Factor) PLL_NS 适用范围 (31~500) 适用 00000000 ~ 000011110: 禁止使用 000011111: 31 000100000: 32 000100001: 33 111110011: 499 111110100: 500 111110101~111111111: 禁止使用
位 5: 4	保留	0x0	resd	请保持为复位值。
位 3: 0	PLL_MS	0x2	rw	PLL 预分频系数(PLL pre-division) PLL_MS 适用范围 (1~15) 0000: 禁止使用 0001: 1 0010: 2 0011: 3 1110: 14 1111: 15

注意：PLL 时钟计算公式

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

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

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

4.3.3 时钟配置寄存器 (CRM_CFG)

访问：0 到 2 个等待周期，字，半字和字节访问，只有当访问发生在时钟切换时，才会插入 1 或 2 个等待周期。

域	简称	复位值	类型	功能
位 31: 30	CLKOUT_SEL1	0x1	rw	<p>内部输出时钟选择 1 (clock output selection 1) 由软件置'1'或清零。</p> <p>00: 系统时钟 (SCLK) 输出； 01: 二级时钟输出，由 CRM_MISC1 寄存器的 CLKOUT_SEL2 位选择二级时钟输出 10: 外部振荡器时钟 (HEXT) 输出； 11: PLL 时钟输出；</p> <p>注意：- 该时钟输出在启动和切换 CLKOUT 时钟源时可能会被截断。 - 在系统时钟作为输出至 CLKOUT 引脚时，请保证输出时钟频率不超过 50MHz (I/O 口最高频率)。</p>
位 29: 27	CLKOUTDIV1	0x0	rw	<p>CLKOUT 分频因子 1 (Clock output division1) 0xx: CLKOUT 100: CLKOUT/2 101: CLKOUT/3 110: CLKOUT/4 111: CLKOUT/5</p>
位 26: 21	保留	0x00	resd	请保持为复位值。
位 20: 16	ERTCDIV	0x00	rw	<p>ERTC 时钟源 HEXT 分频因子 (HEXT division for ERTC clock) 由软件置'1'或清'0'来控制 ERTC 时钟源 HEXT 的预分频系数。 此控制位必须在 ERTC 时钟源之前设定。</p> <p>00000: 禁止使用 00001: 禁止使用 00010: HEXT/2 00011: HEXT/3 00100: HEXT/4 ... 11110: HEXT/30 11111: HEXT/31</p>
位 15: 13	APB2DIV	0x0	rw	<p>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</p>
位 12: 10	APB1DIV	0x0	rw	<p>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</p>
位 9: 8	保留	0x0	resd	请保持为复位值。
位 7: 4	AHBDIV	0x0	rw	<p>AHB 分频因子 (AHB division) 0xxx: SCLK 不分频 1000: SCLK 2 分频 1100: SCLK 64 分频 1001: SCLK 4 分频 1101: SCLK 128 分频 1010: SCLK 8 分频 1110: SCLK 256 分频 1011: SCLK 16 分频 1111: SCLK 512 分频 注意：确保自动顺滑 AUTO_STEP_EN 开启后，再更改 AHBDIV</p>
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。
				系统时钟选择 (System clock select) 00: HICK; 01: HEXT; 10: PLL;
位 1: 0	SCLKSEL	0x0	rw	11: 保留, 保持默认值。 注意: 确保自动顺滑 AUTO_STEP_EN 开启后, 再更改 SCLKSEL

4.3.4 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	请保持为复位值。
位 23	CFDFC	0x0	wo	清除时钟失效标志 (Clock failure detection interrupt clear) 由软件置'1'来清除 CFDF。 0: 无作用; 1: 清除 CFDF 位。
位 22: 21	保留	0x0	resd	请保持为复位值。
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	请保持为复位值。
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	请保持为复位值。
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。

4.3.5 AHB外设复位寄存器1 (CRM_AHBRST1)

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	0x00	resd	请保持为复位值。
位 22	DMA1RST	0x0	rw	DMA1 复位 (DMA1 reset) 0: 无复位; 1: 复位。
位 21: 13	保留	0x000	resd	请保持为复位值。
位 12	CRCRST	0x0	rw	CRC 复位 (CRC reset) 0: 无复位; 1: 复位。
位 11: 6	保留	0x00	resd	请保持为复位值。
位 5	GPIOFRST	0x0	rw	IO 端口 F 复位 (IO port F reset) 0: 无复位; 1: 复位。
位 4: 3	保留	0x00	resd	请保持为复位值。
位 2	GPIOCRST	0x0	rw	IO 端口 C 复位 (IO port C reset) 0: 无复位; 1: 复位。
位 1	GPIOBRST	0x0	rw	IO 端口 B 复位 (IO port B reset) 0: 无复位; 1: 复位。
位 0	GPIOARST	0x0	rw	IO 端口 A 复位 (IO port A reset) 0: 无复位; 1: 复位。

4.3.6 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: 30	保留	0x0	resd	请保持为复位值。
位 29	DACRST	0x0	rw	DAC 接口复位 (DAC interface reset) 0: 无复位； 1: 复位。
位 28	PWCRST	0x0	rw	电源接口复位 (Power interface reset) 0: 无复位； 1: 复位。
位 27: 26	保留	0x0	resd	请保持为复位值。
位 25	CAN1RST	0x0	rw	CAN1 复位 (CAN1 reset) 0: 无复位； 1: 复位。
位 24: 23	保留	0x0	resd	请保持为复位值。
位 22	I2C2RST	0x0	rw	I ² C2 复位 (I ² C2 reset) 0: 无复位； 1: 复位。
位 21	I2C1RST	0x0	rw	I ² C1 复位 (I ² C1 reset) 0: 无复位； 1: 复位。
位 20: 18	保留	0x0	resd	请保持为复位值。
位 17	USART2RST	0x0	rw	USART2 复位 (USART2 reset) 0: 无复位； 1: 复位。
位 16: 15	保留	0x0	resd	请保持为复位值。
位 14	SPI2RST	0x0	rw	SPI2 复位 (SPI2 reset) 0: 无复位； 1: 复位。
位 13: 12	保留	0x0	resd	请保持为复位值。
位 11	WWDTRST	0x0	rw	窗口看门狗复位 (Window watchdog reset) 0: 无复位； 1: 复位。
位 10: 6	保留	0x0	resd	请保持为复位值。
位 5	TMR7RST	0x0	rw	定时器 7 复位 (Timer7 reset) 0: 无复位； 1: 复位。
位 4	TMR6RST	0x0	rw	定时器 6 复位 (Timer6 reset) 0: 无复位； 1: 复位。
位 3	保留	0x0	resd	请保持为复位值。
位 2	TMR4RST	0x0	rw	定时器 4 复位 (Timer4 reset) 0: 无复位； 1: 复位。
位 1	TMR3RST	0x0	rw	定时器 3 复位 (Timer3 reset) 0: 无复位； 1: 复位。
位 0	保留	0x0	resd	请保持为复位值。

4.3.7 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	0x0	resd	请保持为复位值。
位 18	TMR11RST	0x0	rw	定时器 11 复位 (Timer11 reset) 0: 无复位; 1: 复位。
位 17	TMR10RST	0x0	rw	定时器 10 复位 (Timer10 reset) 0: 无复位; 1: 复位。
位 16	TMR9RST	0x0	rw	定时器 9 复位 (Timer9 reset) 0: 无复位; 1: 复位。
位 15	保留	0x0	resd	请保持为复位值。
位 14	SCFGRST	0x0	rw	SCFG 复位 (SCFG reset) 0: 无复位; 1: 复位。
位 13	保留	0x0	resd	请保持为复位值。
位 12	SPI1RST	0x0	rw	SPI1 复位 (SPI1 reset) 0: 无复位; 1: 复位。
位 11: 9	保留	0x0	resd	请保持为复位值。
位 8	ADCRST	0x0	rw	ADC 接口复位 (ADC interface reset) 0: 无复位; 1: 复位。
位 7: 5	保留	0x0	resd	请保持为复位值。
位 4	USART1RST	0x0	rw	USART1 复位 (USART1 reset) 0: 无复位; 1: 复位。
位 3: 1	保留	0x0	resd	请保持为复位值。
位 0	TMR1RST	0x0	rw	TMR1 定时器复位 (TMR1 timer reset) 0: 无复位; 1: 复位。

4.3.8 AHB外设时钟使能寄存器1 (CRM_AHBEN1)

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	0x00	resd	请保持为复位值。
位 22	DMA1EN	0x0	rw	DMA1 时钟使能 (DMA1 clock enable) 0: 关闭; 1: 开启。
位 21: 13	保留	0x000	resd	请保持为复位值。
位 12	CRCEN	0x0	rw	CRC 时钟使能 (CRC clock enable) 0: 关闭; 1: 开启。
位 11: 6	保留	0x00	resd	请保持为复位值。
位 5	GPIOFEN	0x0	rw	IO 端口 F 时钟使能 (IO port F clock enable) 0: 关闭; 1: 开启。
位 4: 3	保留	0x000	resd	请保持为复位值。
位 2	GPIOCEN	0x0	rw	IO 端口 C 时钟使能 (IO port C clock enable) 0: 关闭; 1: 开启。
位 1	GPIOBEN	0x0	rw	IO 端口 B 时钟使能 (IO port B clock enable) 0: 关闭; 1: 开启。
位 0	GPIOAEN	0x0	rw	IO 端口 A 时钟使能 (IO port A clock enable) 0: 关闭; 1: 开启。

4.3.9 APB1外设时钟使能寄存器 (CRM_APB1EN)

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: 30	保留	0x0	resd	请保持为复位值。
位 29	DACEN	0x0	rw	DAC 接口时钟使能 (DAC interface clock enable) 0: 关闭; 1: 开启。
位 28	PWCEN	0x0	rw	电源接口时钟使能 (Power interface clock enable) 0: 关闭; 1: 开启。
位 27: 26	保留	0x0	resd	请保持为复位值。
位 25	CAN1EN	0x0	rw	CAN1 时钟使能 (CAN1 clock enable) 0: 关闭; 1: 开启。
位 24: 23	保留	0x0	resd	请保持为复位值。
位 22	I2C2EN	0x0	rw	I ² C2 时钟使能 (I ² C2 clock enable) 0: 关闭; 1: 开启。
位 21	I2C1EN	0x0	rw	I ² C1 时钟使能 (I ² C1 clock enable) 0: 关闭; 1: 开启。
位 20: 18	保留	0x0	resd	请保持为复位值。
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	请保持为复位值。
位 14	SPI2EN	0x0	rw	SPI2 时钟使能 (SPI 2 clock enable) 0: 关闭; 1: 开启。
位 13: 12	保留	0x0	resd	请保持为复位值。

				窗口看门狗时钟使能 (Window watchdog clock enable) 0: 关闭; 1: 开启。
位 11	WWDTEN	0x0	rw	保留 请保持为复位值。
位 10: 6	保留	0x0	resd	定时器 7 时钟使能 (Timer 7 clock enable) 0: 关闭; 1: 开启。
位 5	TMR7EN	0x0	rw	定时器 6 时钟使能 (Timer 6 clock enable) 0: 关闭; 1: 开启。
位 4	TMR6EN	0x0	rw	定时器 4 时钟使能 (Timer 4 clock enable) 0: 关闭; 1: 开启。
位 3	保留	0x0	resd	定时器 3 时钟使能 (Timer 3 clock enable) 0: 关闭; 1: 开启。
位 2	TMR4EN	0x0	rw	定时器 2 时钟使能 (Timer 2 clock enable) 0: 关闭; 1: 开启。
位 1	TMR3EN	0x0	rw	定时器 1 时钟使能 (Timer 1 clock enable) 0: 关闭; 1: 开启。
位 0	保留	0x0	resd	请保持为复位值。

4.3.10 APB2外设时钟使能寄存器 (CRM_APB2EN)

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	0x0	resd	请保持为复位值。
位 18	TMR11EN	0x0	rw	定时器 11 时钟使能 (TMR11 clock enable) 0: 关闭; 1: 开启。
位 17	TMR10EN	0x0	rw	定时器 10 时钟使能 (TMR10 clock enable) 0: 关闭; 1: 开启。
位 16	TMR9EN	0x0	rw	定时器 9 时钟使能 (TMR9 clock enable) 0: 关闭; 1: 开启。
位 15	保留	0x0	resd	请保持为复位值。
位 14	SCFGGEN	0x0	rw	SCFG 和 CMP 及 OP 时钟使能 (SCFG, CMP, and OP clock enable) 0: 关闭; 1: 开启。
位 13	保留	0x0	resd	请保持为复位值。
位 12	SPI1EN	0x0	rw	SPI1 时钟使能 (SPI1 clock enable) 0: 关闭; 1: 开启。
位 11: 10	保留	0x0	resd	请保持为复位值。
位 9	ADC2EN	0x0	rw	ADC2 接口时钟使能 (ADC2 interface clock enable) 0: 关闭; 1: 开启。
位 8	ADC1EN	0x0	rw	ADC1 接口时钟使能 (ADC1 interface clock enable) 0: 关闭; 1: 开启。
位 7: 5	保留	0x0	resd	请保持为复位值。
位 4	USART1EN	0x0	rw	USART1 时钟使能 (USART1 clock enable) 0: 关闭; 1: 开启。
位 3: 1	保留	0x0	resd	请保持为复位值。
位 0	TMR1EN	0x0	rw	TMR1 定时器时钟使能 (TMR1 timer clock enable) 0: 关闭; 1: 开启。

4.3.11 AHB外设时钟低功耗使能寄存器1 (CRM_AHBLPEN1)

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	0x00	resd	请保持为复位值。
位 22	DMA1LPEN	0x1	rw	睡眠模式下 DMA1 时钟使能 (DMA1 clock enable during sleep mode) 0: 关闭; 1: 开启。
位 21: 17	保留	0x00	resd	请保持为复位值。
位 16	SRAMLPEN	0x1	rw	睡眠模式下 SRAM 时钟使能 (SRAM clock enable during sleep mode) 0: 关闭; 1: 开启。
位 15	FLASHLPEN	0x1	rw	睡眠模式下 FLASH 时钟使能 (FLASH clock enable during sleep mode) 0: 关闭; 1: 开启。
位 14: 13	保留	0x0	resd	请保持为复位值。
位 12	CRCLPEN	0x1	rw	睡眠模式下 CRC 时钟使能 (CRC clock enable during sleep mode) 0: 关闭; 1: 开启。
位 11: 6	保留	0x0	resd	请保持为复位值。
位 5	GPIOFLPEN	0x1	rw	睡眠模式下 IO 端口 F 时钟使能 (IO port F clock enable during sleep mode) 0: 关闭; 1: 开启。
位 4: 3	保留	0x0	resd	请保持为复位值。
位 2	GPIOCLPEN	0x1	rw	睡眠模式下 IO 端口 C 时钟使能 (IO port C clock enable during sleep mode) 0: 关闭; 1: 开启。
位 1	GPIOBLPEN	0x1	rw	睡眠模式下 IO 端口 B 时钟使能 (IO port B clock enable during sleep mode) 0: 关闭; 1: 开启。
位 0	GPIOALPEN	0x1	rw	睡眠模式下 IO 端口 A 时钟使能 (IO port A clock enable during sleep mode) 0: 关闭; 1: 开启。

4.3.12 APB1外设时钟低功耗使能寄存器 (CRM_APB1LPEN)

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: 30	保留	0x0	resd	请保持为复位值。
位 29	DACLPE	0x1	rw	睡眠模式下 DAC 接口时钟使能 (DAC interface clock enable during sleep mode) 0: 关闭; 1: 开启。
位 28	PWCLPE	0x1	rw	睡眠模式下 PWC 时钟使能 (Power interface clock enable during sleep mode) 0: 关闭; 1: 开启。
位 27: 26	保留	0x0	resd	请保持为复位值。
位 25	CAN1LPEN	0x1	rw	睡眠模式下 CAN1 时钟使能 (CAN1 clock enable during sleep mode) 0: 关闭; 1: 开启。

位 24: 23	保留	0x0	resd	请保持为复位值。
位 22	I2C2LPEN	0x1	rw	睡眠模式下 I ² C2 时钟使能 (I ² C2 clock enable during sleep mode) 0: 关闭; 1: 开启。
位 21	I2C1LPEN	0x1	rw	睡眠模式下 I ² C1 时钟使能 (I ² C1 clock enable during sleep mode) 0: 关闭; 1: 开启。
位 20: 18	保留	0x0	resd	请保持为复位值。
位 17	USART2LPEN	0x1	rw	睡眠模式下 USART2 时钟使能 (USART2 clock enable during sleep mode) 0: 关闭; 1: 开启。
位 16: 15	保留	0x0	resd	请保持为复位值。
位 14	SPI2LPEN	0x1	rw	睡眠模式下 SPI2 时钟使能 (SPI 2 clock enable during sleep mode) 0: 关闭; 1: 开启。
位 13: 12	保留	0x0	resd	请保持为复位值。
位 11	WWDTLPEN	0x1	rw	睡眠模式下 WWDT 时钟使能 (Window watchdog clock enable during sleep mode) 0: 关闭; 1: 开启。
位 10: 6	保留	0x0	resd	请保持为复位值。
位 5	TMR7LPEN	0x1	rw	睡眠模式下 TMR7 时钟使能 (Timer 7 clock enable during sleep mode) 0: 关闭; 1: 开启。
位 4	TMR6LPEN	0x1	rw	睡眠模式下 TMR6 时钟使能 (Timer 6 clock enable during sleep mode) 0: 关闭; 1: 开启。
位 3	保留	0x0	resd	请保持为复位值。
位 2	TMR4LPEN	0x1	rw	睡眠模式下 TMR4 时钟使能 (Timer 4 clock enable during sleep mode) 0: 关闭; 1: 开启。
位 1	TMR3LPEN	0x1	rw	睡眠模式下 TMR3 时钟使能 (Timer 3 clock enable during sleep mode) 0: 关闭; 1: 开启。
位 0	保留	0x0	resd	请保持为复位值。

4.3.13 APB2外设时钟低功耗使能寄存器 (CRM_APB2LPEN)

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	0x0	resd	请保持为复位值。
位 18	TMR11LPEN	0x1	rw	睡眠模式下定时器 11 时钟使能 (TMR11 clock enable during sleep mode) 0: 关闭; 1: 开启。
位 17	TMR10LPEN	0x1	rw	睡眠模式下定时器 10 时钟使能 (TMR10 clock enable during sleep mode) 0: 关闭; 1: 开启。
位 16	TMR9LPEN	0x1	rw	睡眠模式下定时器 9 时钟使能 (TMR9 clock enable during sleep mode) 0: 关闭; 1: 开启。
位 15	保留	0x0	resd	请保持为复位值。
位 14	SCFGLPEN	0x1	rw	睡眠模式下 SCFG 和 CMP 及 OP 时钟使能 (SCFG, CMP, and OP clock enable during sleep mode) 0: 关闭; 1: 开启。
位 13	保留	0x0	resd	请保持为复位值。
位 12	SPI1LPEN	0x1	rw	睡眠模式下 SPI1 时钟使能 (SPI1 clock enable during sleep mode) 0: 关闭; 1: 开启。
位 11: 10	保留	0x0	resd	请保持为复位值。
位 9	ADC2LPEN	0x1	rw	睡眠模式下 ADC2 接口时钟使能 (ADC2 interface clock enable during sleep mode) 0: 关闭; 1: 开启。
位 8	ADC1LPEN	0x1	rw	睡眠模式下 ADC1 接口时钟使能 (ADC1 interface clock enable during sleep mode) 0: 关闭; 1: 开启。
位 7: 5	保留	0x0	resd	请保持为复位值。
位 4	USART1LPEN	0x1	rw	睡眠模式下 USART1 时钟使能 (USART1 clock enable during sleep mode) 0: 关闭; 1: 开启。
位 3: 1	保留	0x0	resd	请保持为复位值。
位 0	TMR1LPEN	0x1	rw	睡眠模式下 TMR1 定时器时钟使能 (TMR1 timer clock enable during sleep mode) 0: 关闭; 1: 开启。

4.3.14 外设独立时钟选择寄存器 (CRM_PICLKS)

域	简称	复位值	类型	功能
位 31: 26	保留	0x0000	resd	请保持为复位值。
位 25: 24	CAN1_CLKSEL	0x0	rw	CAN1 主机时钟选择位 (CAN1 HOST CLK SEL) 00: HSE 01: PLL 10: PCLK 11: Reserved
位 23: 0	保留	0x00	resd	请保持为复位值。

4.3.15 电池供电域控制寄存器 (CRM_BPDC)

只能由电池供电域复位有效复位

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、LEXTBYPs、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	请保持为复位值。
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'1'或清'0' 0: 关闭 ERTC 时钟; 1: 开启 ERTC 时钟。
位 14: 10	保留	0x00	resd	请保持为复位值。
位 9: 8	ERTCSEL	0x0	rw	ERTC 时钟源选择 (ERTC clock source 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 分频时钟 (分频由 CRM_CFG 的 ERTC_DIV 设定)。
位 7: 5	保留	0x00	resd	请保持为复位值。
位 4: 3	LEXTDRV	0x3	rw	LEXT 驱动 (Low speed external crystal driving strength) 00: LOW; 01: MEDIUM LOW; 10: MEDIUM HIGH; 11: HIGH。
位 2	LEXTBYPs	0x0	rw	LEXT 旁路使能 (Low speed external crystal bypass) 0: 关闭; 1: 开启。
位 1	LEXTSTBL	0x0	ro	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 0	LEXTEN	0x0	rw	LEXT 使能 (External low-speed oscillator enable) 0: 关闭; 1: 开启。

4.3.16 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/低电压复位标志 (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 引脚复位标志 (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	请保持为复位值。
位 24	RSTFC	0x0	rw	清除复位标志 (Reset flag clear) 由软件置'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	请保持为复位值。
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。

4.3.17 额外寄存器 (CRM_MISC1)

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: 28	CLKOUTDIV2	0x0	rw	CLKOUT 分频因子 2 (Clock output division2) 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 27: 20	保留	0x00	resd	请保持为复位值。
位 19: 16	CLKOUT_SEL2	0xF	rw	CLKOUT 时钟选择位 2 (Clock output sel2) 0000: 保留 0001: ADC 时钟输出 0010: 内部 RC 振荡器时钟 (HICK) 除频输出 0011: LICK 时钟输出 0100: LEXT 时钟输出 0101~1111: 保留
位 15	保留	0x0	resd	请保持为复位值。
位 14	HICK_TO_SCLK	0x0	rw	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, SCLK 的频率为 0: 固定是 8MHz, 即选择 HICK 时钟的 6 分频; 1: 48MHz。
位 13	保留	0x0	resd	请保持为复位值。
位 12	保留	0x1	resd	请保持为复位值。
位 11: 8	保留	0x0	resd	请保持为复位值。
位 7: 0	HICKCAL_KEY	0x00	rw	HICKCAL 写入键值 (HICK calibration key) 此字段为 0x5A 时, HICKCAL [7: 0]才可被写入。

4.3.18 额外寄存器2 (CRM_MISC2)

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: 26	保留	0x000	resd	请保持为复位值。
位 25: 24	VBATHDIV	0x3	rw	VBATAHB 分频因子 (VBATAHB division) 0x: HCLK 不分频 10: HCLK 2 分频 11: HCLK 4 分频
位 23: 22	HEXTDRV	0x2	rw	HEXT 驱动 (High speed external crystal driving strength) 00: LOW; 01: MEDIUM LOW; 10: MEDIUM HIGH; 11: HIGH。
位 21: 19	HEXT_TO_SCLK_DIV	0x0	rw	HEXT 作为系统时钟的分频因子 (HEXT as system clock frequency division) 000: HEXT 001: HEXT/2 010: HEXT/4 011: HEXT/8 100: HEXT/16 101: HEXT/32 其他: 保留
位 18: 16	HICK_TO_SCLK_DIV	0x0	rw	HICK 作为系统时钟的分频因子 (HICK as system clock frequency division) 000: HICK 001: HICK/2 010: HICK/4 011: HICK/8 100: HICK/16 其他: 保留
位 15	保留	0x0	resd	请保持为复位值。
位 14: 12	APB3DIV	0x5	rw	APB3 分频因子 (APB3 division) HCLK 分频后作为 APB3 时钟。 0xx: 不分频 100: 2 分频 101: 4 分频 110: 8 分频 111: 16 分频
位 11: 6	保留	0x0	resd	请保持为复位值。
位 5: 4	AUTO_STEP_EN	0x0	rw	自动滑顺频率切换 (auto step system clock switch enable) 为使切换系统时钟源或是切换 AHB 分频因子时平顺，建议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； 01: 保留； 10: 保留； 11: 开启，当 AHBDIV 或 SCLKSEL 这两个控制位被改动时，会自动触发自动滑顺频率切换功能。
位 3: 0	保留	0xD	resd	固定为 0xD，请勿修改。

5 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 128K 字节
- 信息块由 28K 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、USART2 或 USB 实现 ISP 编程

主存储器只有闪存容量为 128K 字节的片 1 闪存，包含 128 扇区，每扇区大小为 1K 字节。

表 5-1 闪存存储结构 (128K)

结构		名称	地址范围
主存储器	片 1 (Bank1) 128KB	扇区 0	0x0800 0000 – 0x0800 03FF
		扇区 1	0x0800 0400 – 0x0800 07FF
		扇区 2	0x0800 0800 – 0x0800 0BFF
	
		扇区 127	0x0801 FC00 – 0x0801 FFFF
		启动程序代码区 28KB	0x1FFF 8000 – 0x1FFF EFFF
		OTP DATA 1KB	0x1FFF F000 – 0x1FFF F3FF
		OTP LOCK 128B	0x1FFF F500 – 0x1FFF F57F
		用户系统数据区 512B	0x1FFF F800 – 0x1FFF F9FF

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 5-2 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 28KB	0x1FFF 8000 – 0x1FFF EFFF
		OTP DATA 1KB	0x1FFF F000 – 0x1FFF F3FF
		OTP LOCK 128B	0x1FFF F500 – 0x1FFF F57F
		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 FLASH_USD 以及 FLASH_EPPS 寄存器中。

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 5-3 用户系统数据说明

地址	位	内容
0xFFFF_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
		nFAP[7:0]: FAP[7:0] 的反码
	[15:8]	SSB[7:0]: 系统配置字节 (存放在寄存器 FLASH_USD[9:2]) 位 7 保留不用
		位 6 (nSTDBY_WDT) 0: 进入待机模式时停止计数 1: 进入待机模式时不停止计数
		位 5 (nDEPSLP_WDT) 0: 进入深度睡眠模式时停止计数
	[23:16]	

			1: 进入深度睡眠模式时不停止计数 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
		位 4 (nBOOT1)	保留不用
		位 3	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
		位 2 (nSTDBY_RST)	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
		位 1 (nDEPSLP_RST)	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
		位 0 (nWDT_ATO_EN)	0: 看门狗自启动开启 1: 看门狗自启动关闭
	[31:24]	nSSB[7:0]: SSB[7:0]的反码	
0x1FFF_F804	[7:0]	Data0[7:0]: 用户数据 0 (存放在寄存器 FLASH_USD[17:10])	
	[15:8]	nData0[7:0]: Data0[7:0]的反码	
	[23:16]	Data1[7:0]: 用户数据 1 (存放在寄存器 FLASH_USD[25:18])	
	[31:24]	nData1[7:0]: Data1[7:0]的反码	
	[7:0]	EPP0[7:0]: 闪存擦写保护字节 0 (存放在寄存器 FLASH_EPPS[7:0]) 用于保护 128KB 与 64KB 主闪存存储器的扇区 0 ~ 扇区 31, 每个比特位保护 4K 字节 0: 擦写保护启动 1: 擦写保护解除	
0x1FFF_F808	[15:8]	nEPP0[7:0]: EPP0[7:0]的反码	
	[23:16]	EPP1[7:0]: 闪存擦写保护字节 1 (存放在寄存器 FLASH_EPPS[15:8]) 用于保护 128KB 与 64KB 主闪存存储器的扇区 32 ~ 扇区 63, 每个比特位保 护 4K 字节 0: 擦写保护启动 1: 擦写保护解除	
	[31:24]	nEPP1[7:0]: EPP1[7:0]的反码	
	[7:0]	EPP2[7:0]: 闪存擦写保护字节 2 (存放在寄存器 FLASH_EPPS[23:16]) 用于保护 128KB 主闪存存储器的扇区 64~ 扇区 95, 64KB 主闪存存储器此字 节保留不用, 每个比特位保护 4K 字节 0: 擦写保护启动 1: 擦写保护解除	
0x1FFF_F80C	[15:8]	nEPP2[7:0]: EPP2[7:0]的反码	
	[23:16]	EPP3[7:0]: 闪存擦写保护字节 3 (存放在寄存器 FLASH_EPPS[31:24]) 位 6 到位 0 用于保护 128KB 主闪存存储器的扇区 96~ 扇区 123, 64KB 主闪 存存储器位 6 到位 0 保留不用, 每个比特位保护 4K 字节 位 7 用于保护保护 128KB 主闪存存储器的扇区 124~ 扇区 127, 位 7 同时用 于 128KB 与 64KB 主闪存存储器的主存扩展区 0: 擦写保护启动 1: 擦写保护解除	
	[31:24]	nEPP3[7:0]: EPP3[7:0]的反码	
0x1FFF_F810	[7:0]	Data2[7:0]: 用户数据 2	
	[15:8]	nData2[7:0]: Data2[7:0]的反码	
	[23:16]	Data3[7:0]: 用户数据 3	
	[31:24]	nData3[7:0]: Data3[7:0]的反码	
0x1FFF_F814	[7:0]	Data4[7:0]: 用户数据 4	
	[15:8]	nData4[7:0]: Data4[7:0]的反码	
	[23:16]	Data5[7:0]: 用户数据 5	
	[31:24]	nData5[7:0]: Data5[7:0]的反码	
.	.	.	.
0x1FFF_F9FC	[7:0]	Data248[7:0]: 用户数据 248	
	[15:8]	nData248[7:0]: Data248[7:0]的反码	
	[23:16]	Data249[7:0]: 用户数据 249	
	[31:24]	nData249[7:0]: 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 FLASH_CTRL 寄存器，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 FLASH_UNLOCK 寄存器顺序写入键值 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

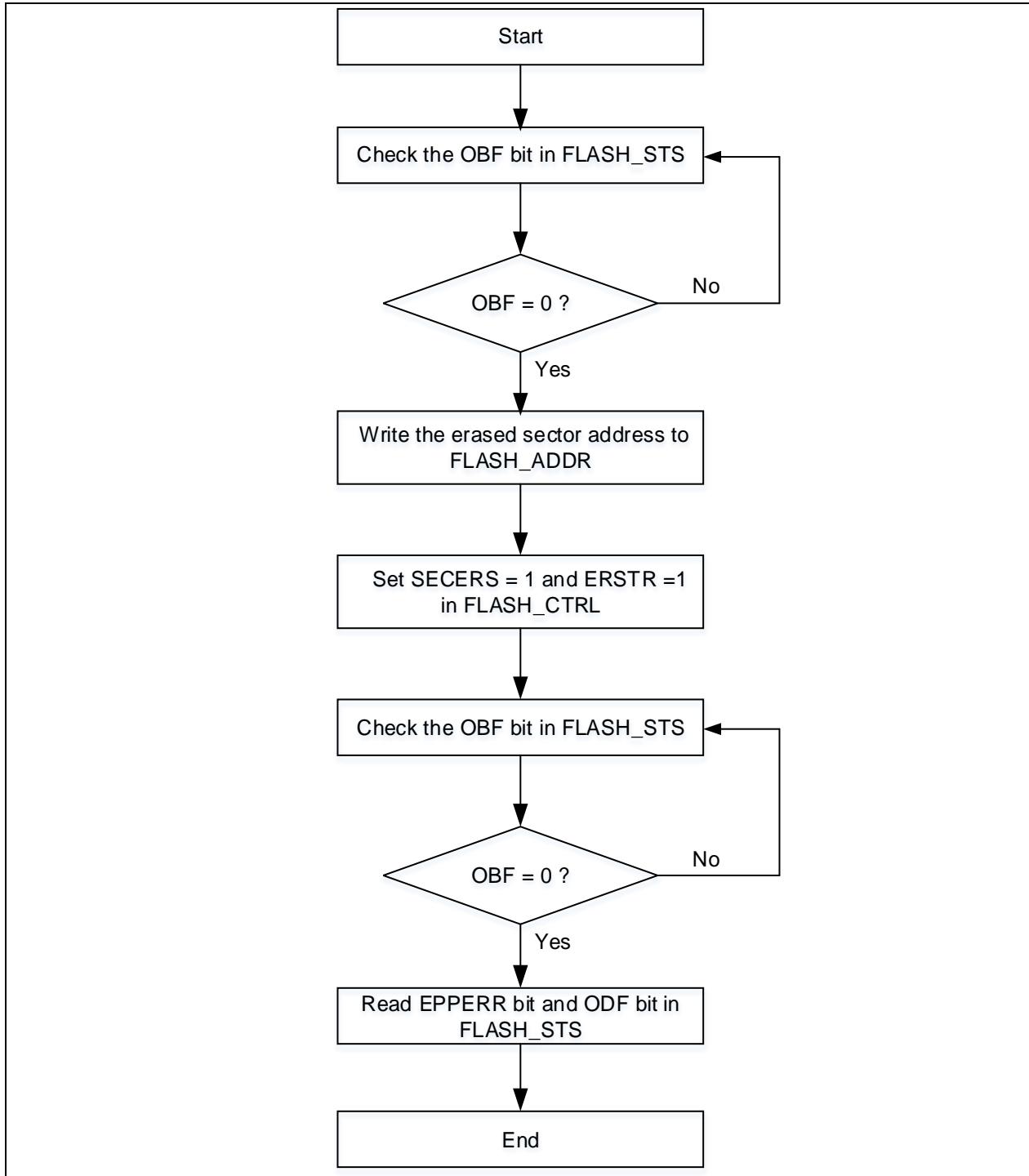
主闪存存储器的每一扇区以及主存扩展区都可以使用扇区擦除功能独立擦除。

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FLASH_ADDR寄存器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EPPER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：

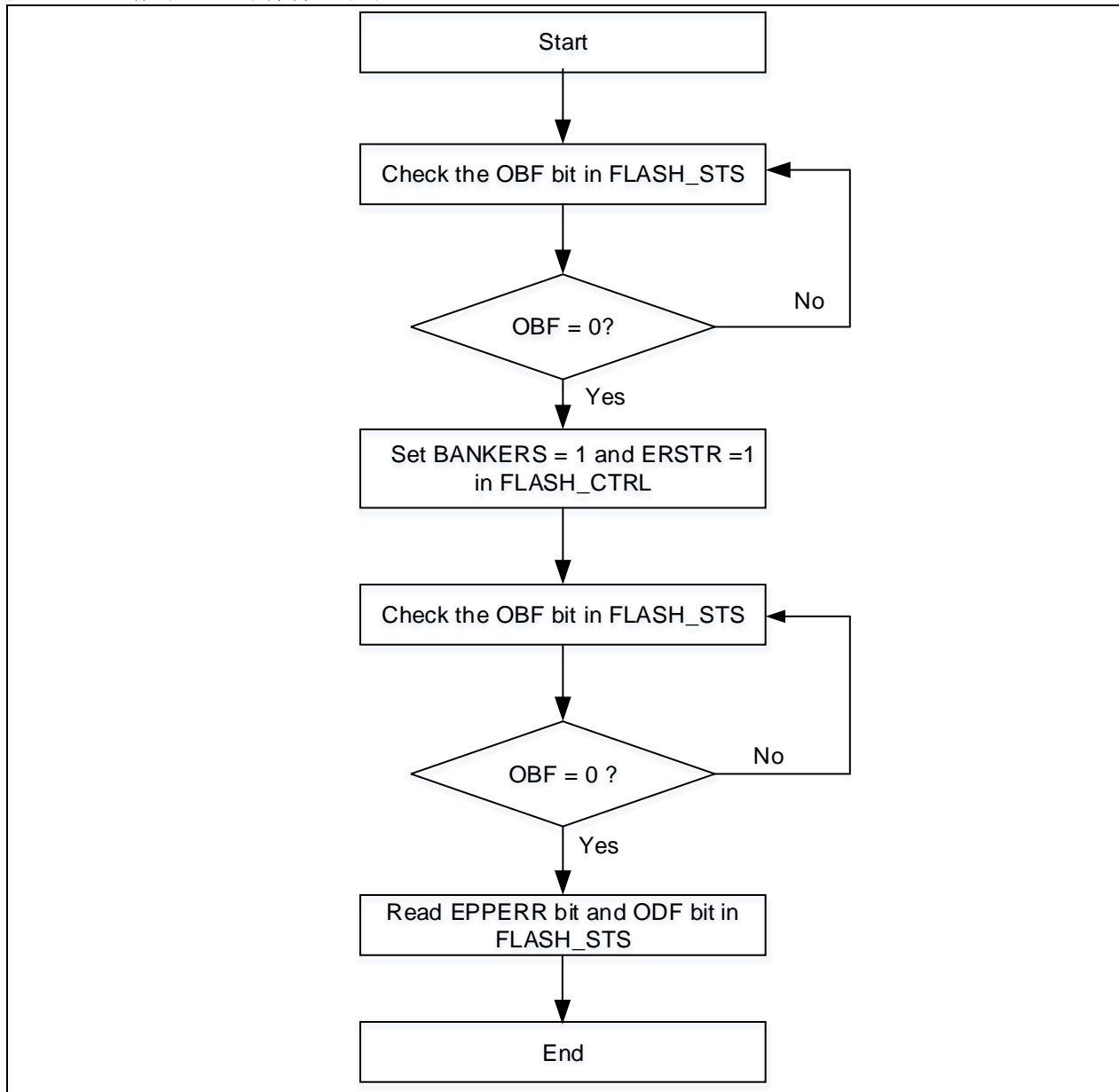
- 检查闪存状态寄存器（FLASH_STS）的OBF位，确认没有正在进行的闪存操作；
- 对闪存控制寄存器（FLASH_CTRL）的BANKERS位以及ERSTR位均置1，启动整片擦除；
- 等待闪存状态寄存器（FLASH_STS）的OBF位变为‘0’，并查询闪存状态寄存器（FLASH_STS）的EPPERR位和ODF位，确认擦除结果。

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

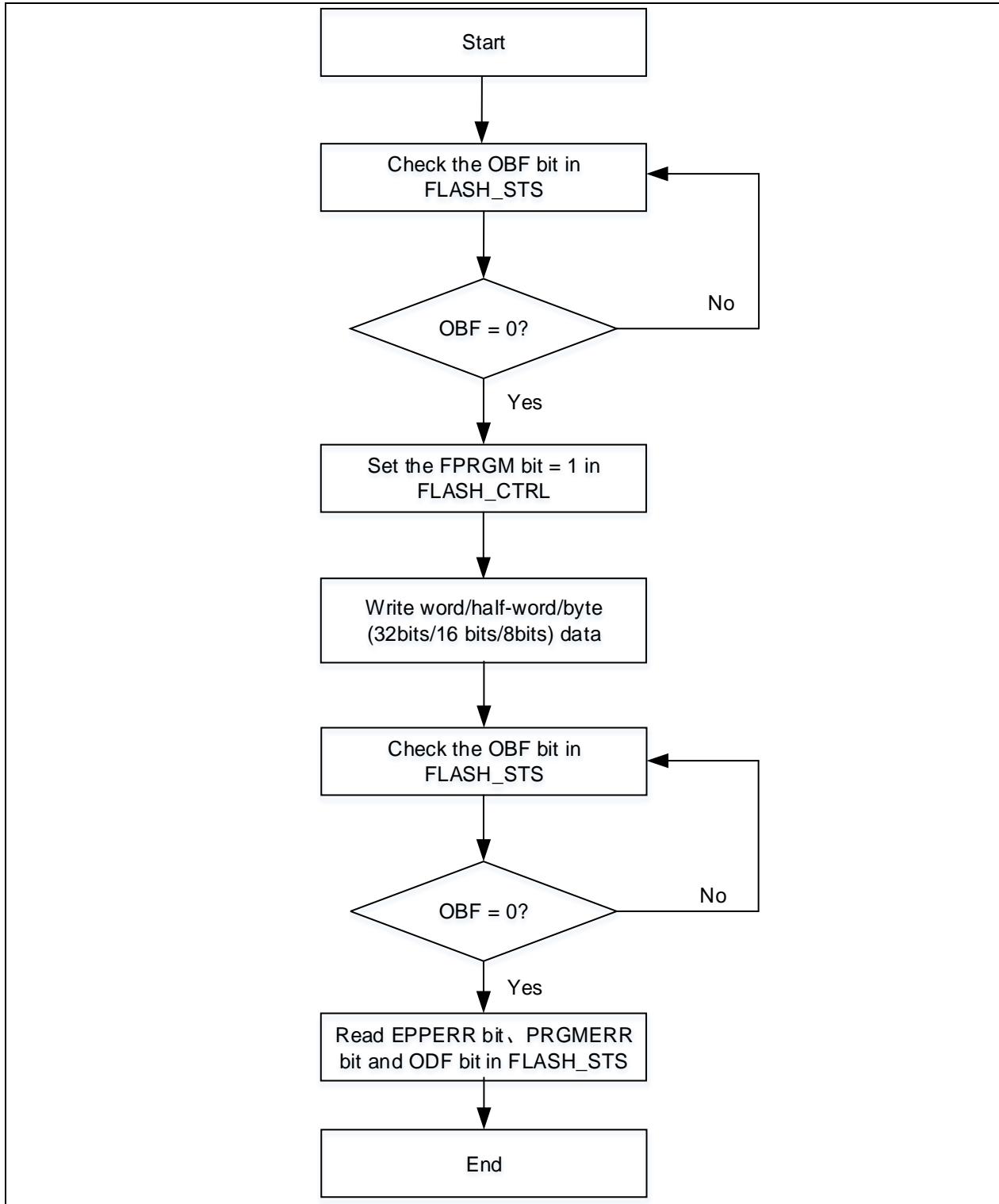
当想要改写主存储器的内容时，可以通过主存储器编程流程完成一次写入 32 位、16 位或 8 位的数据。主存储器编程流程：

- 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 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 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，并置位闪存状态寄存器（FLASH_STS）的 PRGMERR 位来告知编程发生错误。
- 2) 编程期间，CPU 会被暂停直到编程完成才处理后续操作。
- 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 OTP操作

OTP 的 DATA 与 LOCK 区不可擦除，其操作方法，包括读取、解锁、编程都跟主存储器相同。每 32 字节 DATA 对应 1 字节 LOCK，当 LOCK 字节被写为 0x00 时，该 32 字节 DATA 无法再写入。DATA 与 LOCK 同样有访问保护，请参照表闪存访问权限。

表 5-4 OTP 的 DATA 与 LOCK 关系

OTP LOCK 地址范围	OTP DATA 地址范围
LOCK0	0x1FFF F500 – 0x1FFF F50F
LOCK1	0x1FFF F501 – 0x1FFF F50F
LOCK2	0x1FFF F502 – 0x1FFF F50F
...	...
LOCK31	0x1FFF F510 – 0x1FFF F51F

5.5 用户系统数据区操作

5.5.1 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 FLASH_UNLOCK 寄存器顺序写入键值 KEY1 (0x45670123) 和键值 KEY2 (0xCDEF89AB)；对 FLASH_USD_UNLOCK 寄存器顺序写入键值 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。

5.5.2 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：

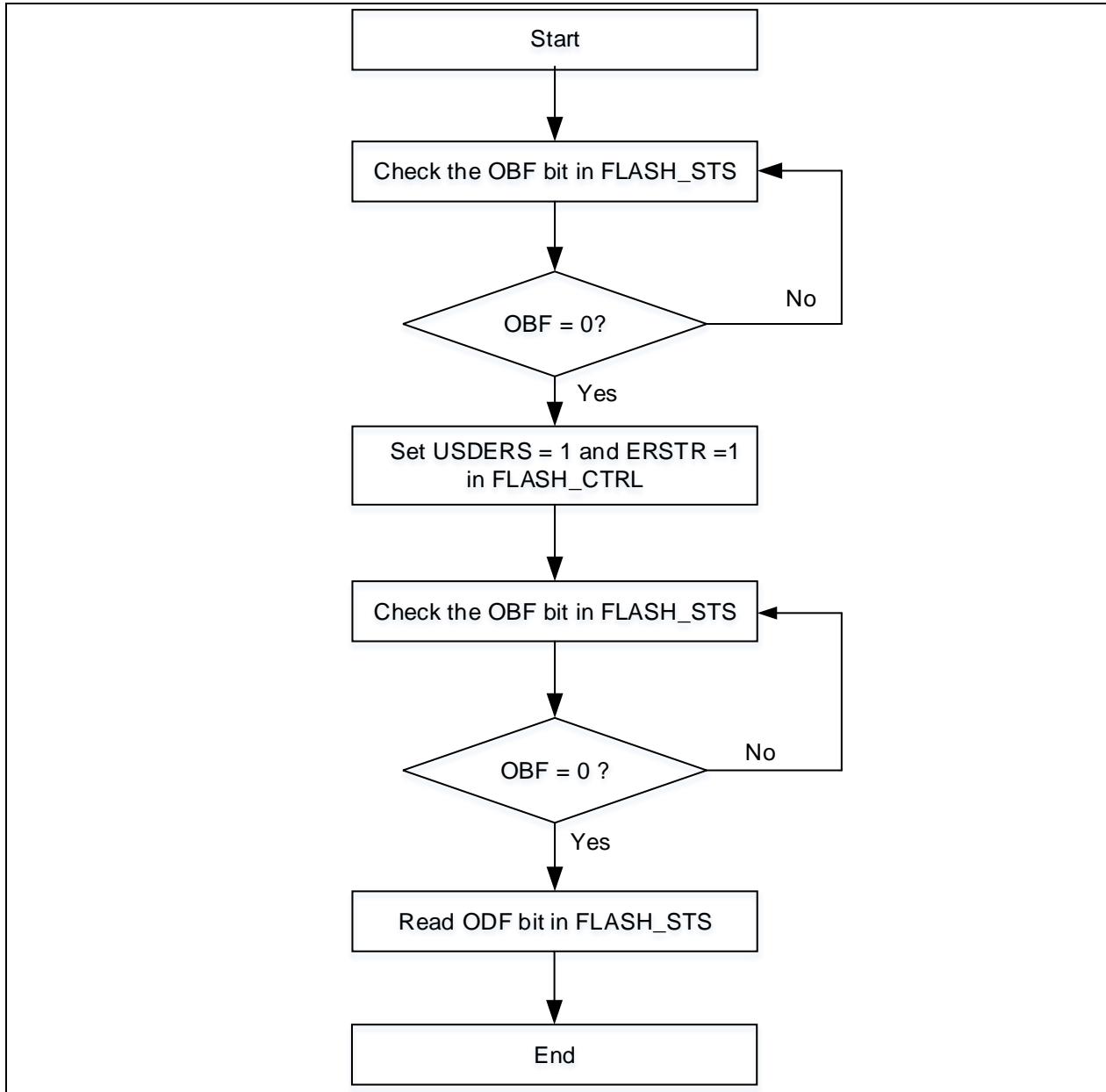
- 检查闪存状态寄存器 (FLASH_STS) 的 OBF 位，确认没有正在进行的闪存操作；
- 对闪存控制寄存器 (FLASH_CTRL) 的 USDRS 位以及 ERSTR 位均置 1，启动整块系统数据区擦除；
- 等待闪存状态寄存器 (FLASH_STS) 的 OBF 位变为 ‘0’，并查询闪存状态寄存器 (FLASH_STS) 的 ODF 位，确认擦除结果。

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



5.5.3 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：

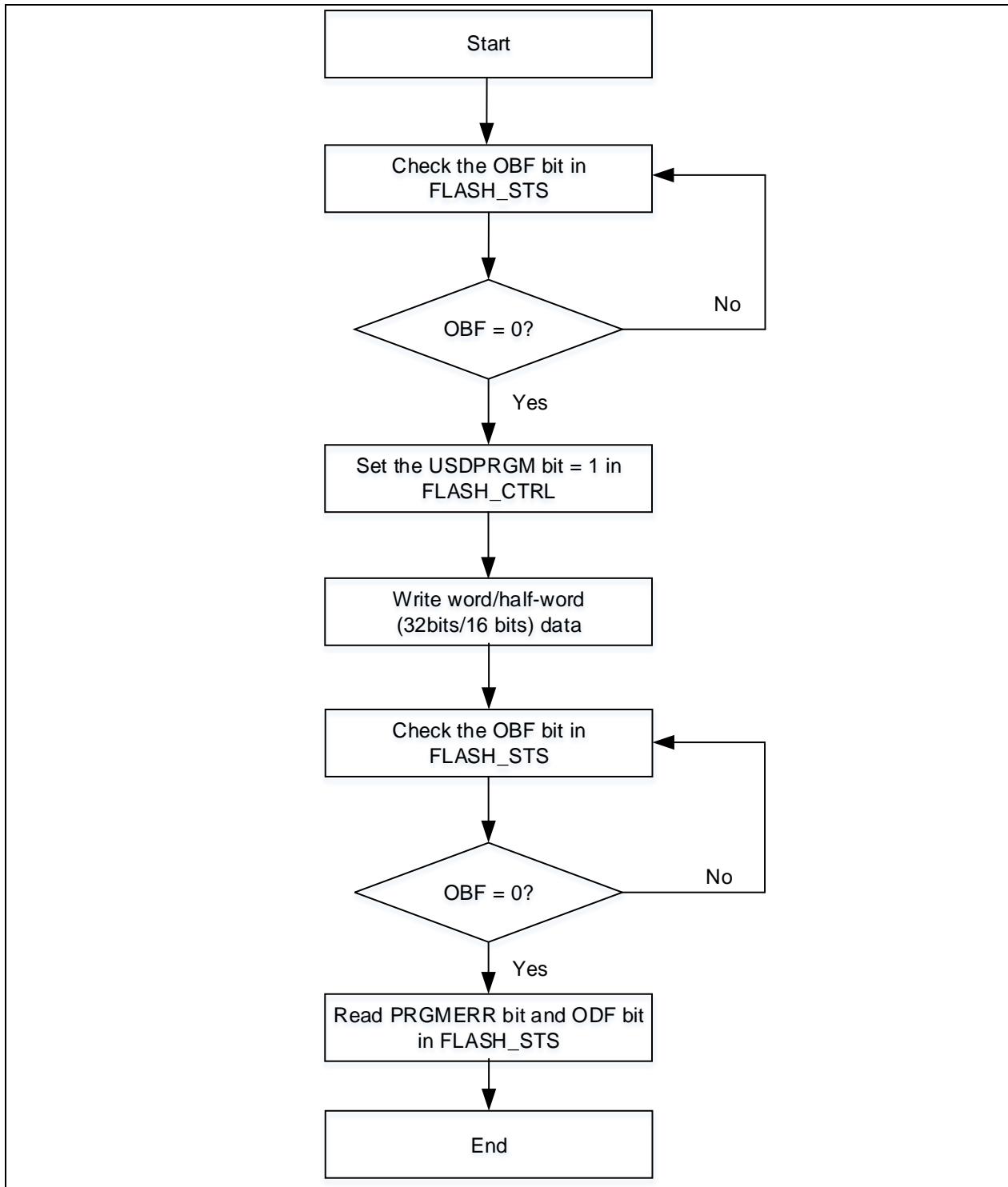
- 检查闪存状态寄存器（FLASH_STS）的OBF位，确认没有正在进行的闪存操作；
- 对闪存控制寄存器（FLASH_CTRL）的USDPRGM位置1，此时可以接受对用户系统数据区的编程指令；
- 对指定的地址写入要编程的数据（任意字/半字）；
- 等待闪存状态寄存器（FLASH_STS）的OBF位变为‘0’，并查询闪存状态寄存器（FLASH_STS）的PRGMERR位和ODF位，确认编程结果。

注意：

编程期间，CPU 会被暂停直到编程完成才处理后续操作。

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



5.5.4 读取

通过 CPU 的 AHB 总线可以直接寻址访问用户系统数据区。

5.6 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。

5.6.1 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，将不能被解除，并且禁止用户以任何方式重新擦除以及写入系统数据区。

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		无 (3)			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		无 (3)		允许		禁止
OTP	低级访问保护		禁止		允许		禁止
	高级访问保护		无 (3)		允许		禁止

(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) 高级访问保护开启时，系统自动从主闪存启动

5.6.2 擦写保护

擦写保护的基本单位为 4KB。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

5.7 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。

5.8 特殊功能

5.8.1 安全库区设定

设定以密码保护主存中指定范围的程序区，即安全库区，仅能被执行，无法写入与删除，除非输入指定密码。安全库区划分为 指令区 与 唯读区。指令区无法读取，唯独区可被读取。

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，除非输入方案商指定密码，也无法删除（包含 ISP/IAP/SWD）；

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 CPU 指令读取指令库区；

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 SLIB_UNLOCK 寄存器写入 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 SLIB_UNLOCK 寄存器写入 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 SLIB_SET_RANGE 寄存器设定要保护的扇区，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 EM_SLIB_SET 寄存器。
- 等待 OBF 位变为'0'；
- 在 SLIB_SET_PWD 寄存器设定安全库区密码；
- 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 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 SLIB_PWD_CLR 寄存器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 SLIB_STS0 寄存器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。

5.8.2 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 SLIB_STS0 的位 0，获知启动程序代码区域当前的模式

- 对 SLIB_UNLOCK 寄存器写入 0xA35F6D24，以进行启动程序代码区域模式设定解锁
- 写非 0xFF 到 BTM_MODE_SET 寄存器的位 7-0
- 等待 OBF 位变为 '0'；
- 进行系统复位，重装载设定字；
- 读出 SLIB_STS0 寄存器用于判断设定结果。

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
当开启该功能后，原启动程序存储器启动模式将强制变为主闪存存储器启动模式。

5.8.3 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。

- CRC 生成多项式：0x4C11DB7，
即 $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- CRC 初始值：0x00000000

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 FLASH_CRC_ADDR 寄存器设定要校验的代码起始地址；
- 在 FLASH_CRC_CTRL 寄存器位 15-0，设定要校验的代码数量（单位是扇区）；
- 在 FLASH_CRC_CTRL 寄存器设置位 16，启动 CRC 校验；
- 等待 OBF 位变为 '0'；
- 读出 FLASH_CRC_CHKRCR 寄存器用于判断 CRC 校验结果。

注意：

FLASH_CRC_ADDR 寄存器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。

5.9 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 闪存接口寄存器映射和复位值

寄存器简称	基址偏移量	复位值
FLASH_PSR	0x00	0x0000 01F0
FLASH_UNLOCK	0x04	0xFFFF XXXX
FLASH_USD_UNLOCK	0x08	0xFFFF XXXX
FLASH_STS	0x0C	0x0000 0000
FLASH_CTRL	0x10	0x0000 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_CHKRCR	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

5.9.1 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	1	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, 后续请勿修改。
位 7	PFT_ENF2	1	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	1	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	保留	0	resd	须保持为 0。
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 ≤ 128MHz 使用; 100: 四个等待周期, 128MHz < 系统时钟 ≤ 160MHz 使用; 101: 五个等待周期, 160MHz < 系统时钟 ≤ 180MHz 使用。

5.9.2 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。

5.9.3 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。

5.9.4 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	rwc1	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	rwc1	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	rwc1	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。

5.9.5 闪存控制寄存器（FLASH_CTRL）

域	简称	复位值	类型	功能
位 31:13	保留	0x0000	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。

5.9.6 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。

5.9.7 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: 未用 位 8: nSTDBY_WDT 位 7: nDEPSLP_WDT 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

5.9.8 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。

5.9.9 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 ... 00011011: 扇区 27 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
-----	------------	---	----	--

5.9.10 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 ... 0000111111: 扇区 63 (64KB 主闪存存储器的最后扇区) ... 0001111111: 扇区 127 (128KB 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 ... 0000011111: 扇区 63 (64KB 主闪存存储器的最后扇区) ... 0000111111: 扇区 127 (128KB 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	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 ... 0000011111: 扇区 63 (64KB 主闪存存储器的最后扇区) ... 0000111111: 扇区 127 (128KB 主闪存存储器的最后扇区)

5.9.11 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闪存安全库区额外状态寄存器 (SLIB_MISC_STS) 位 0 与位 1 中体现。

5.9.12 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	0x00000000	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。
				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。
位 0	SLIB_PWD_ERR	0	ro	

5.9.13 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。

5.9.14 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。 硬件启动 CRC 后，会自动清除该位。 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

5.9.15 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。

5.9.16 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。

5.9.17 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 ... 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 ... 0000011111: 扇区 63 (64KB 主闪存存储器的最后扇区) ...
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 ... 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。

5.9.18 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 ... 00011011: 扇区 27 11111111: 无指令安全区 其余设定值：无效 注意：

位 15:0	EM_SLIB_SET	0x000	wo	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) 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。

5.9.19 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	0x000000	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。

5.9.20 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

AT32M412/AT32M416 支持多达 46 个双向 I/O 引脚，这些引脚分为 4 组，分别为 PA0-PA15、PB0-PB15、PC13-PC15、PF0-PF2 与 PF6-PF13，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。每个引脚都可以软件配置成通用输入、通用输出、复用功能和模拟引脚。当配置成通用输出或复用功能时，可选择推挽或是开漏输出。

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写保护功能。

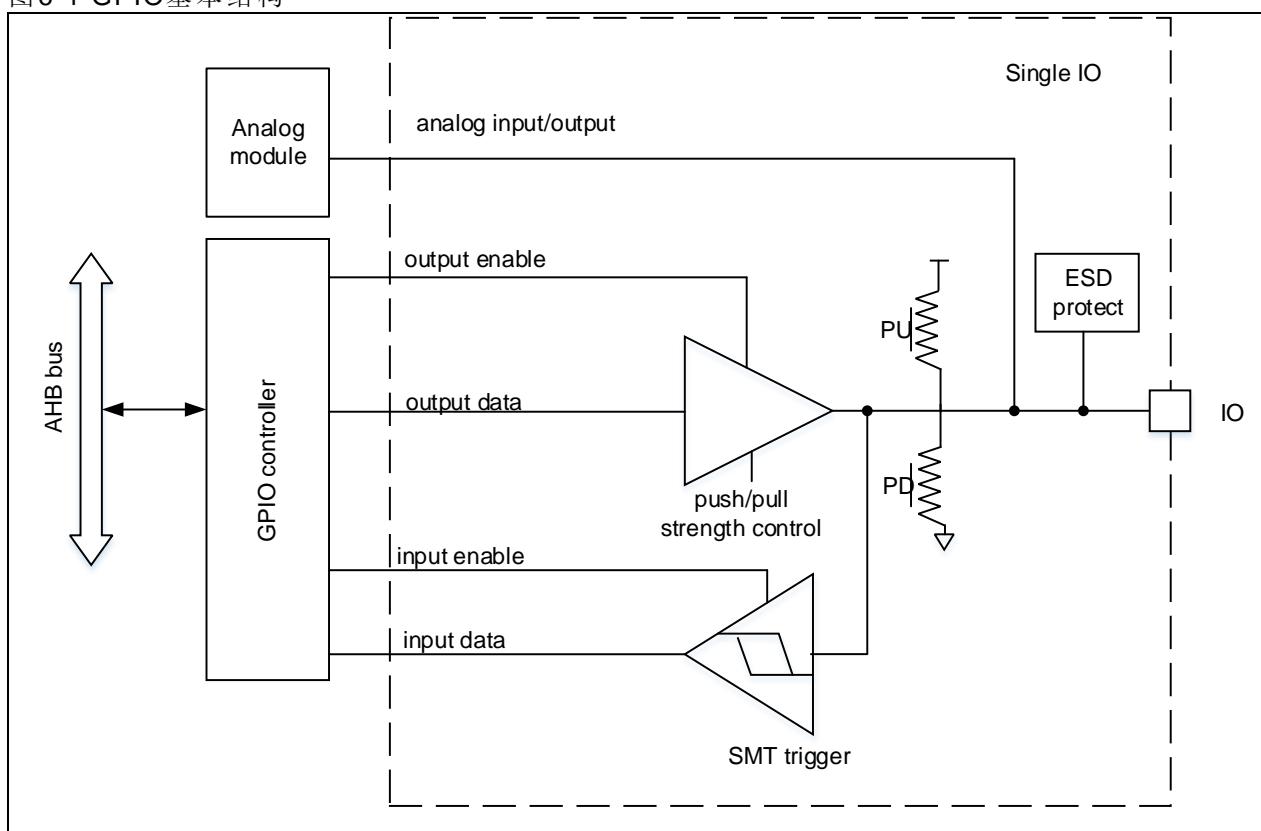
6.2 功能描述

6.2.1 GPIO 结构

每个 I/O 端口对应的寄存器允许半字或字节访问，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，都被配置为模拟模式。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，PB3/SWO 为复用功能无上拉下拉模式。

6.2.3 通用功能输入配置

配置模式	IOMC	PULL
通用浮空输入		00 或 11
通用下拉输入	00	10
通用上拉输入		01

当引脚被配置为通用输入时：

- 引脚状态可通过对输入数据寄存器的读访问得到
- 施密特触发器有效。

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，则配置为模拟模式。

6.2.4 模拟配置

配置模式	IOMC	PULL
模拟	11	不使用

当引脚被配置为模拟时：

- 施密特触发器无效
- 输出数据寄存器无效
- 设置上拉/下拉电阻无效。
- 引脚状态不可通过输入数据寄存器访问得到。

6.2.5 通用功能输出配置

配置模式	IOMC	OM	HDRV	ODRV[1: 0]	PULL
通用推挽无上拉/下拉	01	0			00 或 11
通用推挽上拉	01	0			01
通用推挽下拉	01	0			10
通用开漏无上拉/下拉	01	1			00 或 11
通用开漏上拉	01	1			01
通用开漏下拉	01	1			10

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

当引脚被配置为通用输出时：

- 施密特触发器有效
- 在开漏输出模式时，当输出数据寄存器配置为数字0时，引脚输出数字0；当输出数据寄存器配置为数字1时，无内部上拉和下拉时，引脚为高阻状态，有内部上拉时，引脚上拉到数字1，有内部下拉时，引脚下拉到数字0
- 在推挽输出模式时，可通过输出数据寄存器输出数字0/1
- 通过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端口写保护

为了防止误操作导致 GPIO 功能混乱，提供每个对应引脚的写保护机制。一旦设定写保护，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 GPIOx_MUXL（从引脚 0 到引脚 7）或 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。

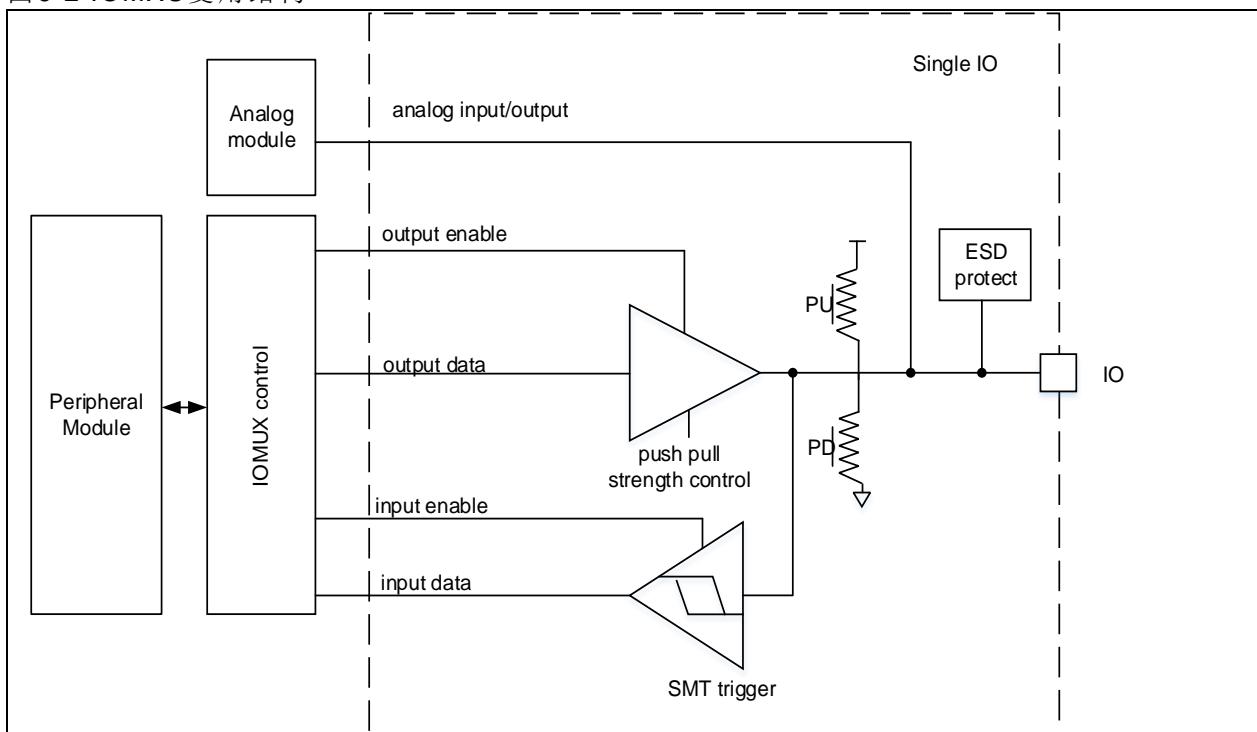
每个引脚通过设定 GPIOx_MUXL 或 GPIOx_MUXH，只会和单一外设的单一脚进行对应，不存在单一引脚多个外设抢占的冲突。

引脚作为复用输入功能时，引脚和 GPIO 控制器断开，由 IOMUX 控制器进行控制。

引脚作为复用输入时，配置 GPIOx_CFGR 将该引脚设定为复用模式，根据外设特性可配置 GPIOx_PULL 将引脚配置为浮空、上拉或下拉。

引脚作为复用输出或双向复用功能时，配置 GPIOx_CFGR 将该引脚设定为复用模式，根据外设特性可配置 GPIOx_OMODE 将引脚配置为推挽或开漏模式，可配置 GPIOx_ODRVR 和 GPIOx_HDRV 配置引脚的输出驱动能力以及 GPIOx_PULL 配置引脚的上下拉。

图 6-2 IOMUXU 复用结构



6.2.8 复用功能配置

配置模式	IOMC	OM	HDRV	ODRV[1: 0]	PULL
复用推挽无上拉/下拉	10	0			00 或 11
复用推挽上拉	10	0			01
复用推挽下拉	10	0			10
复用开漏无上拉/下拉	10	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	00 或 11
复用开漏上拉	10	1			01
复用开漏下拉	10	1			10
复用输入无上拉/下拉	10	x		仅对复用输出或双向复用功能起作用	00 或 11
复用输入上拉	10	x			01
复用输入下拉	10	x			10

当引脚被配置为复用功能时：

- 施密特触发器有效
- 在开漏输出模式时，当外设输出为数字0时，引脚输出数字0；当外设输出为数字1时，无内部上拉和下拉时，引脚为高阻状态，有内部上拉时，引脚上拉到数字1，有内部下拉时，引脚下拉到数字0
- 在推挽输出模式时，根据外设的输出引脚输出数字0/1
- 引脚状态可通过对输入数据寄存器的读访问得到

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 GPIOx_MUXL (从引脚 0 到引脚 7) 或 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		TMR1_EXT		TMR9_CH2C	I2C2_SCL	SPI2_SCK/ I2S2_CK		USART2_ CTS
PA1	ERTC_ REFIN			TMR9_CH1C	I2C2_SDA	SPI1_SCK/ I2S1_CK	I2C1_SMBA	USART2_ RTS_DE
PA2				TMR9_CH1		SPI1_MOSI/ I2S1_SD		USART2_TX
PA3				TMR9_CH2		SPI2_MISO / I2S2_MCK	I2S2_MCK	USART2_RX
PA4		TMR3_CH2			I2C1_SCL	SPI1_CS/ I2S1_WS	SPI2_CS/ I2S2_WS	USART2_CK
PA5				TMR10_CH1C		SPI1_SCK/ I2S1_CK		
PA6		TMR1_BRK	TMR3_CH1	TMR10_CH1		SPI1_MISO/ I2S1_MCK	I2S2_MCK	
PA7		TMR1_CH1C	TMR3_CH2	TMR11_CH1		SPI1_MOSI/ I2S1_SD	I2C2_SCL	
PA8	CLKOUT	TMR1_CH1	TMR4_EXT		I2C1_SDA	I2S2_MCK	I2C2_SCL	USART1_CK
PA9	CLKOUT	TMR1_CH2		TMR9_BRK	I2C1_SCL	SPI2_SCK/ I2S2_CK	I2C2_SMBA	USART1_TX
PA10	ERTC_ REFIN	TMR1_CH3		TMR11_BRK	I2C1_SDA	SPI2_MISO/ I2S2_MCK	I2C1_SMBA	USART1_RX
PA11		TMR1_CH4	TMR4_CH1		I2C2_SCL	SPI2_MOSI/ I2S2_SD	I2C1_SMBA	USART1_ CTS
PA12		TMR1_EXT	TMR4_CH2	TMR10_CH1	I2C2_SDA	SPI2_MISO/ I2S2_MCK		USART1_ RTS_DE
PA13	SWDIO	IR_OUT	TMR4_CH3	TMR10_CH1C	I2C1_SDA	SPI2_MISO / I2S2_MCK	I2C1_SCL	
PA14	SWCLK	TMR1_BRK			I2C1_SMBA	SPI2_MOSI / I2S2_SD	I2C1_SDA	
PA15		TMR1_BRK			I2C1_SCL	SPI1_CS/ I2S1_WS	SPI2_CS / I2S2_WS	USART1_TX

引脚名	MUX8	MUX9	MUX10	MUX11	MUX12	MUX13	MUX14	MUX15
PA0	USART2_RX				CMP1_OUT			EVENTOUT
PA1								EVENTOUT
PA2		CAN1_RX			CMP2_OUT			EVENTOUT
PA3		CAN1_TX						EVENTOUT
PA4								EVENTOUT
PA5								EVENTOUT
PA6					CMP1_OUT			EVENTOUT
PA7					CMP2_OUT			EVENTOUT
PA8	USART2_TX							EVENTOUT
PA9								EVENTOUT
PA10								EVENTOUT
PA11	USART2_RX	CAN1_RX	TMR1_CH1C	TMR1_BRK2	CMP1_OUT			EVENTOUT
PA12	USART2_RX	CAN1_TX	TMR1_CH2C		CMP2_OUT			EVENTOUT
PA13								EVENTOUT
PA14	USART2_TX							EVENTOUT
PA15	USART2_RX							EVENTOUT

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		TMR1_CH2C	TMR3_CH3			SPI1_MISO / I2S1_MCK	I2S1_MCK	USART2_RX
PB1		TMR1_CH3C	TMR3_CH4			SPI1_MOSI / I2S1_SD	SPI2_SCK / I2S2_CK	USART2_CK
PB2			TMR3_EXT		I2C1_SMBA	SPI2_MISO / I2S2_MCK		
PB3	SWO	TMR1_CH2	TMR3_CH3	TMR4_EXT	I2C2_SDA	SPI1_SCK / I2S1_CK	SPI2_SCK / I2S2_CK	USART1_RTS_DE
PB4		TMR1_CH4C	TMR3_CH1	TMR11_BRK	I2C2_SDA	SPI1_MISO / I2S1_MCK	SPI2_MISO / I2S2_MCK	USART1_CTS
PB5			TMR3_CH2	TMR10_BRK	I2C1_SMBA	SPI1_MOSI / I2S1_SD	SPI2_MOSI / I2S2_SD	USART1_CK
PB6		TMR1_CH3C	TMR4_CH1	TMR10_CH1C	I2C1_SCL	I2S1_MCK	SPI2_CS / I2S2_WS	USART1_TX
PB7		TMR1_CH1C	TMR4_CH2	TMR11_CH1C	I2C1_SDA	SPI2_MOSI / I2S2_SD		USART1_RX
PB8		TMR1_CH2C	TMR4_CH3	TMR10_CH1	I2C1_SCL	SPI2_SCK / I2S2_CK		USART1_TX
PB9	IR_OUT	TMR1_CH3C	TMR4_CH4	TMR11_CH1	I2C1_SDA	SPI2_CS / I2S2_WS	I2C2_SDA	
PB10		TMR1_BRK			I2C2_SCL	SPI2_SCK / I2S2_CK		
PB11				TMR10_BRK	I2C2_SDA	SPI2_MOSI / I2S2_SD		
PB12		TMR1_BRK		TMR9_BRK	I2C2_SMBA	SPI2_CS / I2S2_WS		
PB13	CLKOUT	TMR1_CH1C		TMR9_CH1C	I2C2_SCL	SPI2_SCK / I2S2_CK		
PB14		TMR1_CH2C		TMR9_CH1	I2C2_SDA	SPI2_MISO / I2S2_MCK		
PB15	ERTC_REFIN	TMR1_CH3C		TMR9_CH1C		SPI2_MOSI / I2S2_SD		

引脚名	MUX8	MUX9	MUX10	MUX11	MUX12	MUX13	MUX14	MUX15
PB0					CMP1_OUT			EVENTOUT
PB1								EVENTOUT
PB2								EVENTOUT
PB3	USART2_TX		TMR3_EXT					EVENTOUT
PB4	USART2_RX			TMR10_CH1				EVENTOUT
PB5	USART2_CK	CAN1_RX		TMR11_CH1	CMP2_OUT			EVENTOUT
PB6		CAN1_TX						EVENTOUT
PB7		CAN1_STB	TMR3_CH4					EVENTOUT
PB8		CAN1_RX	TMR1_BRK	TMR9_BRK	CMP1_OUT			EVENTOUT
PB9		CAN1_TX			CMP2_OUT			EVENTOUT
PB10					CMP1_OUT			EVENTOUT
PB11					CMP2_OUT			EVENTOUT
PB12		CAN1_RX		TMR9_CH2C				EVENTOUT
PB13		CAN1_TX						EVENTOUT
PB14								EVENTOUT
PB15				TMR9_CH2				EVENTOUT

表6-3通过GPIOC_MUX*寄存器配置端口C的复用功能

引脚名	MUX0	MUX1	MUX2	MUX3	MUX4	MUX5	MUX6	MUX7
PC13		TMR1_BRK						
PC14		TMR1_BRK2						
PC15				TMR9_BRK				

引脚名	MUX8	MUX9	MUX10	MUX11	MUX12	MUX13	MUX14	MUX15
PC13			TMR1_CH1C	TMR1_CH4C				EVENTOUT
PC14								EVENTOUT
PC15								EVENTOUT

表6-4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		TMR1_CH1		TMR11_CH1	I2C1_SDA	SPI2_CS / I2S2_WS		USART1_RX
PF1		TMR1_CH2C		TMR9_CH1C	I2C1_SCL	SPI2_SCK / I2S2_CK		USART1_TX
PF2	CLKOUT							
PF6					I2C2_SCL			
PF7					I2C2_SDA			
PF8								
PF9			TMR4_CH1	TMR9_CH1				
PF10			TMR4_CH2	TMR9_CH2				
PF11								
PF12								
PF13								

引脚名	MUX8	MUX9	MUX10	MUX11	MUX12	MUX13	MUX14	MUX15
PF0			TMR1_CH3C					EVENTOUT
PF1								EVENTOUT
PF2								EVENTOUT
PF6								EVENTOUT
PF7								EVENTOUT
PF8								EVENTOUT
PF9								EVENTOUT
PF10								EVENTOUT
PF11								EVENTOUT
PF12								EVENTOUT
PF13								EVENTOUT

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浮空输入/上拉输入/下拉输入
- 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 GPIOx_MUXL/GPIOx_MUXH 寄存器得到唯一外设复用。

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 6-5 硬件抢占功能

引脚名字	抢占使能位	说明
PA0	PWC_CTRL[16]=1	抢占使能位有效之后，PA0 引脚直接作为 PWC 的 WKUP1 功能使用
PC13	PWC_CTRL[17]=1	抢占使能位有效之后，PC13 引脚直接作为 PWC 的 WKUP2 功能使用
PB5	PWC_CTRL[21]=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_CTRL[22]=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	(ERTC_CTRL[23]=1) (ERTC_CTRL[22: 21]!=00) (ERTC_CTRL[11]=1& ERTC_TAMP[17]=0) (ERTC_TAMP[0]=1& ERTC_TAMP[16]=0)	抢占使能位有效之后，PC13 作为 RTC 通道使用
PA0	(ERTC_CTRL[11]=1& ERTC_TAMP[17]=1) (ERTC_TAMP[0]=1& ERTC_TAMP[16]=1) (ERTC_TAMP[3]=1)	抢占使能位有效之后，PA0 引脚直接作为 TAMPER2_BPR 功能使用
PC14	CMR_BPDC[0]=1	抢占使能位有效之后，PC14 作为 LEXT 通道使用
PC15	CMR_BPDC[0]=1 & CMR_BPDC[2]=0	抢占使能位有效之后，PC15 作为 LEXT 通道使用
PF0	CMR_CTRL[16]=1	抢占使能位有效之后，PF0 作为 HEXT 通道使用
PF1	CMR_CTRL[16]=1& CMR_CTRL[18]=0	抢占使能位有效之后，PF1 作为 HEXT 通道使用

注意： PA0 或者 PC13 不能同时使能 TAMPER_BPR 功能和 PWC 的 WKUP 功能。

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6-6 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	0xEBFF FFFF
GPIOx_CFGR(x =B,C,F)	0x00	0xFFFF FF8F(B) 0xFFFF FFFF
GPIOx_OMODER	0x04	0x0000 0000
GPIOx_ODRVR	0x08	0x0000 0000
GPIOA_PULL	0x0C	0x2400 0000(A)
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_TOGR	0x2C	0x0000 0000
GPIOx_HDRV	0x3C	0x0000 0000

6.3.1 GPIO配置寄存器 (GPIOx_CFGR) (x=A..F)

偏移地址: 0x00

复位值: 0xEBFF_FFFF 端口 A

0xFFFF_FFBF 端口 B

0xFFFF_FFFF 其它端口

域	简称	复位值	类型	功能
位 2y+1: 2y	IOMCy	0xEBFF FFFF 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 01: 通用输出 10: 复用功能 11: 模拟 (复位后的模式)

6.3.2 GPIO输出模式寄存器 (GPIOx_OMODE) (x=A..F)

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（GPIO_x_ODRVR） (x=A..F)

偏移地址: 0x08

复位值: 0x00000000

域	简称	复位值	类型	功能
位 2y+1: 2y	ODRV _y	0x0000 0000	rw	GPIO _x 的驱动能力配置 (y=0...15) (GPIO _x 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（GPIO_x_PULL）(x=A..F)

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	PULL _y	0x2400 0000	rw	GPIO _x 的上下拉配置 (y=0...15) (GPIO _x pull configure) 用于配置相应的 I/O 上拉或下拉。 00: 无作用 01: 上拉 10: 下拉 11: 无作用

6.3.5 GPIO输入数据寄存器（GPIO_x_IDT）(x=A..F)

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	0XXXX	ro	GPIO _x 输入的数据 (GPIO _x input data) GPIO _x 对应 IO 口的输入电平状态，每一位对应 GPIO _x 的一个 IO。

6.3.6 GPIO输出数据寄存器（GPIO_x_ODT）(x=A..F)

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	GPIO _x 输出的数据 (IO output data)。 每一位对应 GPIO _x 的一个 IO。 GPIO _x 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..F)

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..F)

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..F)

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 1000: MUX8 1001: MUX9 1010: MUX10 1011: MUX11 1100: MUX12 1101: MUX13 1110: MUX14 1111: MUX15

6.3.10 GPIO复用高位寄存器 (GPIOx_MUXH) (x=A..F)

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 1000: MUX8 1001: MUX9 1010: MUX10 1011: MUX11 1100: MUX12 1101: MUX13 1110: MUX14 1111: MUX15

6.3.11 GPIO位清除寄存器 (GPIOx_CLR) (x=A..F)

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 GPIO位翻转寄存器 (GPIOx_TOGR) (x=A..F)

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	IOTB	0x0000	wo	翻转 GPIOx 位 (GPIOx toggle bit) 写'1'的位其对应 ODT 寄存器位会翻转原有的值，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: 对应位翻转。

6.3.13 极大电流推动/吸入能力切换控制寄存器 (GPIOx_HDRV) (x=A..F)

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：

- 管理连接到 GPIO 口的外部中断
- 管理存储器映射方式
- 管理部分 IRTMR GPIO 配置
- 管理部分 GPIO 复用/重映射配置
- 管理 NRST/PF2 复用配置

为防止从 PF2 切换到 NRST 产生不预期的行为，推荐配置流程如下：

- 1) 配置 PF2 为推挽通用输出模式并输出 1;
- 2) 配置 SCFG_IOCFG 寄存器中的 NRST_RMP 为 0，将 NRST/PF2 从 PF2 切换到 NRST;
- 3) 检查 SCFG_IOCFG 寄存器中的 NRST_RMP 确实成功写为 0

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_CFG2	0x04	0x0000 0000
SCFG_EXINTC1	0x08	0x0000 0000
SCFG_EXINTC2	0x0C	0x0000 0000
SCFG_EXINTC3	0x10	0x0000 0000
SCFG_EXINTC4	0x14	0x0000 0000
SCFG_UHDRV	0x2C	0x0000 0000
SCFG_IOCFG	0X60	0x0000 0000
SCFG_IOKEY	0X64	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: 8	保留	0x000000	resd	请保持为复位值。
位 7: 6	IR_SRC_SEL	0x0	rw	红外调制包络信号源选择 (Infrared modulation envelope signal source selection) 用于选择红外调制包络信号源： 00: TMR10 01: USART1 10: USART2 11: 保留
位 5	IR_POL	0x0	rw	红外输出极性选择 (Infrared output polarity selection) 0: 红外线发射输出 (IR_OUT) 不反向 1: 红红外线发射输出 (IR_OUT) 反向
位 4	PA11_12_RMP	0x0	rw	PA11 和 PA12 的重映射 (PA11 and PA12 remap) 由软件设置和清除该位。此位控制着小封装 (20PIN) 上 PA9/PA10 和 PA11/PA12 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)
位 3: 2	保留	0x0	resd	请保持为复位值。
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: 从系统存储器启动 11: 从内置 SRAM 启动

7.2.2 SCFG配置寄存器2 (SCFG_CFG2)

域	简称	复位值	类型	功能
位 31: 25	保留	0x0000 000	resd	请保持为复位值。
位 24	CAN1_TST_SEL	0x0	rw	CAN1 时间戳计数源选择位 (CAN1 timestamp counter source sel) 0: GPTM3 1: GPTM4
位 23: 3	保留	0x0000 000	resd	请保持为复位值。
位 2	PVM_LK	0x0	rw	PVM 锁使能位(PVM lock enable) 0: 断开 PVM 中断与 TIM1/TIM9/TIM10/11/刹车输入的连接。PVMSEL 和 PVMEN 位可以被软件修改。 1: 使能 PVM 中断与 TIM1/TIM9/TIM10/11/刹车输入的连接。PVMSEL 和 PVMEN 位为只读位，不能被软件修改
位 1	保留	0x0	resd	请保持为复位值。
位 0	CPU_LK	0x0	rw	CPU 锁使能位(CPU lock enable) 0: 断开 CPU 锁与 TIM1/TIM9/TIM10/11/刹车输入的连接。 1: 使能 CPU 锁与 TIM1/TIM9/TIM10/11/刹车输入的连接。

7.2.3 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	保持默认值。
位 15: 12	EXINT3	0x0	rw	配置 EXINT3 的输入源 (configure EXINT3 source) 选择 EXINT3 外部中断的输入源。 0000: GPIOA 引脚 3 0001: GPIOB 引脚 3 0010: GPIOC 引脚 3 0011: GPIOD 引脚 3 0100: GPIOE 引脚 3 其他: 保留
位 11: 8	EXINT2	0x0	rw	配置 EXINT2 的输入源 (configure EXINT2 source) 选择 EXINT2 外部中断的输入源。 0000: GPIOA 引脚 2 0001: GPIOB 引脚 2 0010: GPIOC 引脚 2 0011: GPIOD 引脚 2 0100: GPIOE 引脚 2 0101: GPIOF 引脚 2 其他: 保留
位 7: 4	EXINT1	0x0	rw	配置 EXINT1 的输入源 (configure EXINT1 source) 选择 EXINT1 外部中断的输入源。 0000: GPIOA 引脚 1 0001: GPIOB 引脚 1 0010: GPIOC 引脚 1 0011: GPIOD 引脚 1 0100: GPIOE 引脚 1 0101: GPIOF 引脚 1 其他: 保留
位 3: 0	EXINT0	0x0	rw	配置 EXINT0 的输入源 (configure EXINT0 source) 选择 EXINT0 外部中断的输入源。 0000: GPIOA 引脚 0 0001: GPIOB 引脚 0 0010: GPIOC 引脚 0 0011: GPIOD 引脚 0 0100: GPIOE 引脚 0 0101: GPIOF 引脚 0 其他: 保留

7.2.4 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	保持默认值。
位 15: 12	EXINT7	0x0	rw	配置 EXINT7 的输入源 (configure EXINT7 source) 选择 EXINT7 外部中断的输入源。 0000: GPIOA 引脚 7 0001: GPIOB 引脚 7 0010: GPIOC 引脚 7 0011: GPIOD 引脚 7 0100: GPIOE 引脚 7 其他: 保留
位 11: 8	EXINT6	0x0	rw	配置 EXINT6 的输入源 (configure EXINT6 source) 选择 EXINT6 外部中断的输入源。 0000: GPIOA 引脚 6 0001: GPIOB 引脚 6 0010: GPIOC 引脚 6 0011: GPIOD 引脚 6 0100: GPIOE 引脚 6 0101: GPIOF 引脚 6 其他: 保留
位 7: 4	EXINT5	0x0	rw	配置 EXINT5 的输入源 (configure EXINT5 source) 选择 EXINT5 外部中断的输入源。 0000: GPIOA 引脚 5 0001: GPIOB 引脚 5 0010: GPIOC 引脚 5 0011: GPIOD 引脚 5 0100: GPIOE 引脚 5 其他: 保留
位 3: 0	EXINT4	0x0	rw	配置 EXINT4 的输入源 (configure EXINT4 source) 选择 EXINT4 外部中断的输入源。 0000: GPIOA 引脚 4 0001: GPIOB 引脚 4 0010: GPIOC 引脚 4 0011: GPIOD 引脚 4 0100: GPIOE 引脚 4 其他: 保留

7.2.5 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	保持默认值。
位 15: 12	EXINT11	0x0	rw	配置 EXINT11 的输入源 (configure EXINT11 source) 选择 EXINT11 外部中断的输入源。 0000: GPIOA 引脚 11 0001: GPIOB 引脚 11 0010: GPIOC 引脚 11 0011: GPIOD 引脚 11 0100: GPIOE 引脚 11 其他: 保留
位 11: 8	EXINT10	0x0	rw	配置 EXINT10 的输入源 (configure EXINT10 source) 选择 EXINT10 外部中断的输入源。 0000: GPIOA 引脚 10 0001: GPIOB 引脚 10 0010: GPIOC 引脚 10 0011: GPIOD 引脚 10 0100: GPIOE 引脚 10 0101: GPIOF 引脚 10 其他: 保留
位 7: 4	EXINT9	0x0	rw	配置 EXINT9 的输入源 (configure EXINT9 source) 选择 EXINT9 外部中断的输入源。 0000: GPIOA 引脚 9 0001: GPIOB 引脚 9 0010: GPIOC 引脚 9 0011: GPIOD 引脚 9 0100: GPIOE 引脚 9 0101: GPIOF 引脚 9 其他: 保留
位 3: 0	EXINT8	0x0	rw	配置 EXINT8 的输入源 (configure EXINT8 source) 选择 EXINT8 外部中断的输入源。 0000: GPIOA 引脚 8 0001: GPIOB 引脚 8 0010: GPIOC 引脚 8 0011: GPIOD 引脚 8 0100: GPIOE 引脚 8 0101: GPIOF 引脚 8 其他: 保留

7.2.6 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	保持默认值。
位 15: 12	EXINT15	0x0	rw	配置 EXINT15 的输入源 (configure EXINT15 source) 选择 EXINT15 外部中断的输入源。 0000: GPIOA 引脚 15 0001: GPIOB 引脚 15 0010: GPIOC 引脚 15 0011: GPIOD 引脚 15 0100: GPIOE 引脚 15 其他: 保留
位 11: 8	EXINT14	0x0	rw	配置 EXINT14 的输入源 (configure EXINT14 source) 选择 EXINT14 外部中断的输入源。 0000: GPIOA 引脚 14 0001: GPIOB 引脚 14 0010: GPIOC 引脚 14 0011: GPIOD 引脚 14 0100: GPIOE 引脚 14 其他: 保留
位 7: 4	EXINT13	0x0	rw	配置 EXINT13 的输入源 (configure EXINT13 source) 选择 EXINT13 外部中断的输入源。 0000: GPIOA 引脚 13 0001: GPIOB 引脚 13 0010: GPIOC 引脚 13 0011: GPIOD 引脚 13 0100: GPIOE 引脚 13 其他: 保留
位 3: 0	EXINT12	0x0	rw	配置 EXINT12 的输入源 (configure EXINT12 source) 选择 EXINT12 外部中断的输入源。 0000: GPIOA 引脚 12 0001: GPIOB 引脚 12 0010: GPIOC 引脚 12 0011: GPIOD 引脚 12 0100: GPIOE 引脚 12 其他: 保留

7.2.7 SCFG超高电流推动/吸入能力 (SCFG_UHDRV)

域	简称	复位值	类型	功能
位 31: 7	保留	0x0000 000	resd	保持默认值。
位 6	PA12_UH	0x0	rw	<p>PA12 超高电流推动/吸入能力 (PA12 Ultra high sourcing/sinking strength) 这些位可由软件读写, 用于控制 PA12 PAD 电流推动/吸入能力。 0: 无效 1: 对应的 GPIO 切换为超高电流推动/吸入能力 当该位使能时, 所对应的 GPIO 电流推动/吸入能力 (GPIOx_ODRVR&GPIOx_HDRV) 控制 bit 无效</p>
位 5	PA11_UH	0x0	rw	<p>PA11 超高电流推动/吸入能力 (PA11 Ultra high sourcing/sinking strength) 这些位可由软件读写, 用于控制 PA11 PAD 电流推动/吸入能力。 0: 无效 1: 对应的 GPIO 切换为超高电流推动/吸入能力 当该位使能时, 所对应的 GPIO 电流推动/吸入能力 (GPIOx_ODRVR&GPIOx_HDRV) 控制 bit 无效</p>
位 4	保留	0x0	resd	保持默认值。
位 3	PB8_UH	0x0	rw	<p>PB8 超高电流推动/吸入能力 (PB8 Ultra high sourcing/sinking strength) 这些位可由软件读写, 用于控制 PB8 PAD 电流推动/吸入能力。 0: 无效 1: 对应的 GPIO 切换为超高电流推动/吸入能力 当该位使能时, 所对应的 GPIO 电流推动/吸入能力 (GPIOx_ODRVR&GPIOx_HDRV) 控制 bit 无效</p>
位 2	保留	0x0	resd	保持默认值。
位 1	PB9_UH	0x0	rw	<p>PB9 超高电流推动/吸入能力 (PB9 Ultra high sourcing/sinking strength) 这些位可由软件读写, 用于控制 PB9 PAD 电流推动/吸入能力。 0: 无效 1: 对应的 GPIO 切换为超高电流推动/吸入能力 当该位使能时, 所对应的 GPIO 电流推动/吸入能力 (GPIOx_ODRVR&GPIOx_HDRV) 控制 bit 无效</p>
位 0	保留	0x0	resd	保持默认值。

7.2.8 SCFG IO配置寄存器 (SCFG_IOCFG)

域	简称	复位值	类型	功能
位 31: 1	保留	0x00000000	resd	请保持为复位值。 NRST 复用选择 (NRST remap) 0: NRST; 1: PF2。
位 0	NRST_RMP	0x0	rw	注 1: 该寄存器受 SCFG_IOKEY 寄存器保护。 注 2: 该寄存器位于 VDD 域, 对该寄存器写的话, 请读回来验证是否写成功。 该位不受系统复位影响, 只有上电复位才可以复位此位。

7.2.9 SCFG IO配置写保护寄存器 (SCFG_IOKEY)

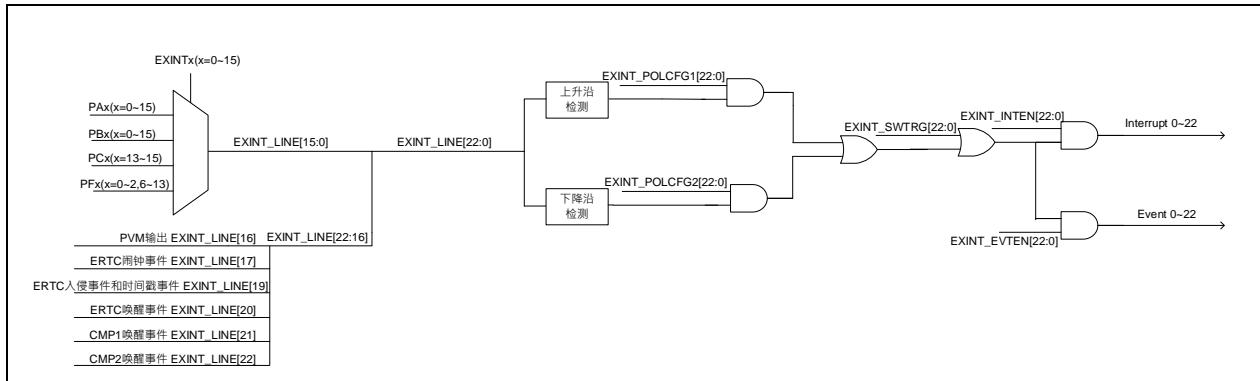
域	简称	复位值	类型	功能
位 31: 16	保留	0x0000	resd	请保持为复位值。 IOCFG 写入键值 (IOCFG key)
位 15:0	IOKEY	0x0	w	写入 0x7A83, 解除 SCFG_IOCFG 写保护; 写入其它值, 开启 SCFG_IOCFG 写保护; 读返回值为 0

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[22:0] (其中位 18 为保留位), 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 22 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[22:0] (其中位 18 为保留位), 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 入侵事件和时间戳事件, ERTC 唤醒事件以及 CMP1/CMP2 唤醒事件共六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/PF0 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: 23	保留	0x0	resd	硬件强制为 0。
位 22: 0	INTENx	0x00000	rw	线 x 上的中断使能/禁止位 (Interrupt enable or disable on line x) 0: 禁止中断请求; 1: 使能中断请求。 注: 位 18 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: 23	保留	0x0	resd	硬件强制为 0。
位 22: 0	EVTENx	0x00000	rw	线 x 上的事件使能/禁止位 (Event enable or disable on line x) 0: 禁止事件请求; 1: 使能事件请求。 注: 位 18 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: 23	保留	0x0	resd	硬件强制为 0。
位 22: 0	RPx	0x00000	rw	线 x 上的上升沿触发事件配置位 (Rising polarity configuration bit of line x) 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 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: 23	保留	0x0	resd	硬件强制为 0。
位 22: 0	FRx	0x00000	rw	线 x 上的下降沿触发事件配置位 (Falling polarity event configuration bit of line x) 这些位用于选择线 x 由下降沿触发中断和事件。 0: 禁止下降沿触发; 1: 允许下降沿触发。 注: 位 18 为保留位, 未使用。

8.3.5 软件触发寄存器 (EXINT_SWTRG)

域	简称	复位值	类型	功能
位 31: 23	保留	0x0	resd	硬件强制为 0。
位 22: 0	SWTx	0x00000	rw	软件触发线 x (Software trigger on line x) 当 EXINT_INTEN 寄存器中的对应位为 1, 则软件写此位 硬件将自动置起 EXINT_INTSTS 寄存器中的对应位并产 生中断。 当 EXINT_EVTEN 寄存器中的对应位为 1, 则软件写此位 硬件将自动产生对应中断线上的事件。 0: 默认值; 1: 产生软件触发。 注: 通过清除 EXINT_INTSTS 的对应位 (写入 1), 可 以清除该位为 0。 注: 位 18 为保留位, 未使用。

8.3.6 中断状态寄存器 (EXINT_INTSTS)

域	简称	复位值	类型	功能
位 31: 23	保留	0x0	resd	硬件强制为 0。
位 22: 0	LINEx	0x00000	rw1c	线 x 状态位 (Line x state bit) 0: 没有发生中断; 1: 发生了中断。 注: 在该位中写入 '1' 可以清除它。 注: 位 18 为保留位, 未使用。

9 DMA 控制器 (DMA)

9.1 简介

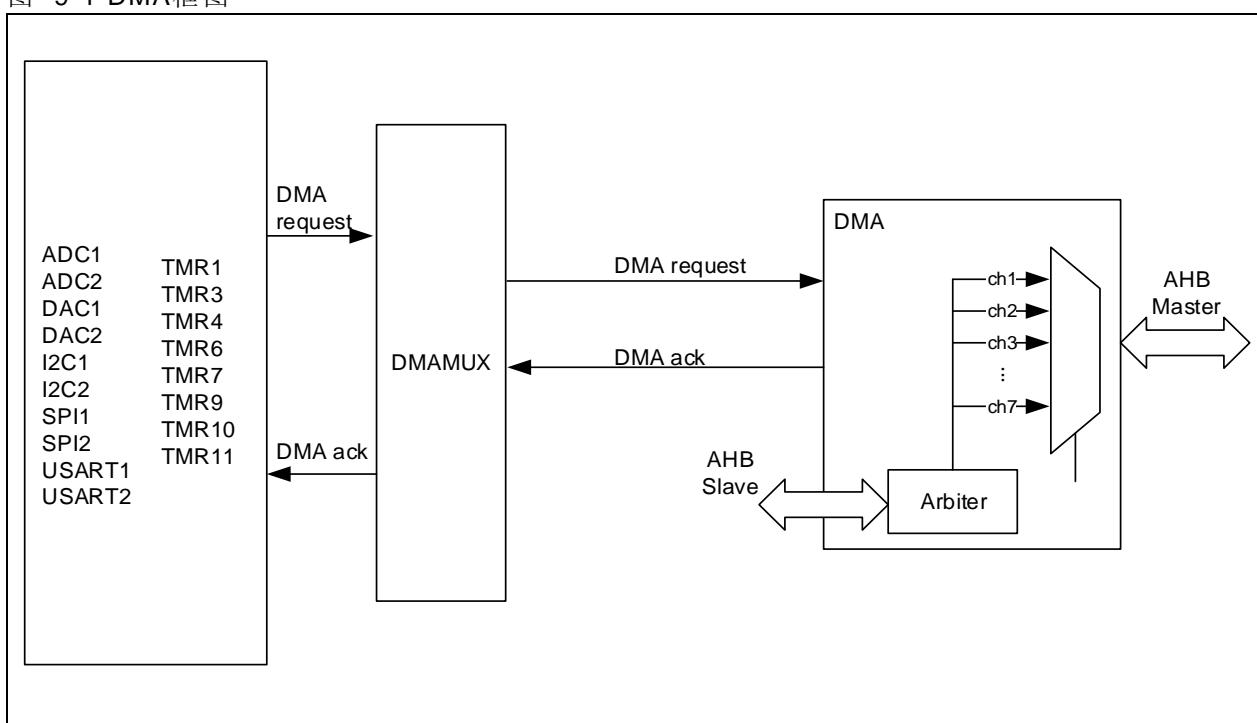
直接存储器存取 (DMA) 用来提供在外设和存储器之间或者存储器和存储器之间的高速数据传输。无须 CPU 干预，数据可以通过 DMA 快速地移动，这就节省了 CPU 的资源来做其他操作。

一个处理器包含 1 个 DMA 控制器。每个控制器各有 7 个 DMA 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 7 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
- 支持多路复用器

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_CPBAX寄存器)

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_CMBAX寄存器)

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_DTCNTx寄存器)

可编程的传输数据长度最大为 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_CHCTRLx寄存器）

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 (PWIDTH/MWIDTH)。

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_DTCNTx 寄存器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_CHCTRLx 寄存器的 CHEN 位)

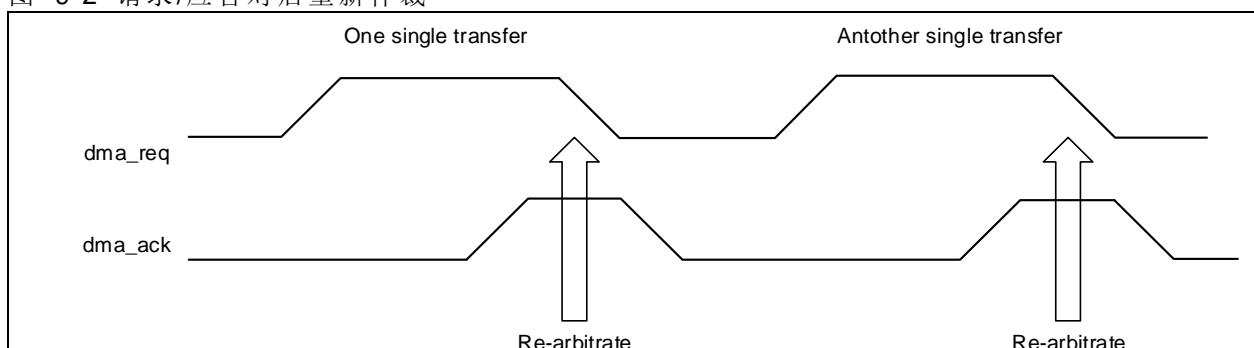
9.3.2 握手机制

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。优先级最高的通道将在先前的通道完成占用主控制器之后，具有主控制器优先级。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_CHCTRLx 中的 PWIDTH 和 MWIDTH 位可以对源数据和目标数据的数据宽度进行编程，当 PWIDTH 不等于 MWIDTH 时，会依据 PWIDTH/ MWIDTH 设定将资料对齐。

图 9-3 PWIDHT: byte, MWIDTH: half-word

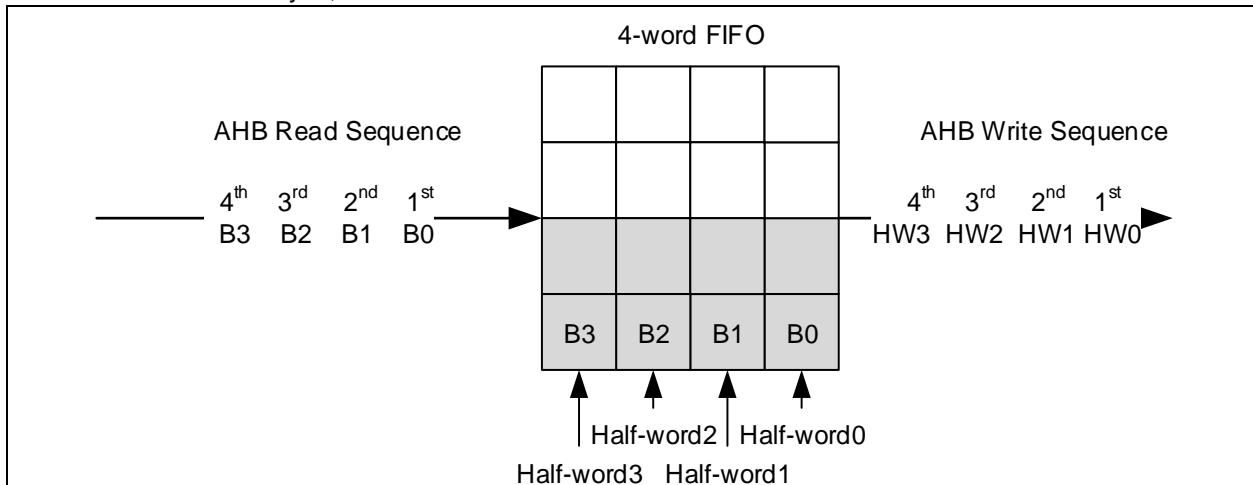


图 9-4 PWIDHT: half-word, MWIDHT: word

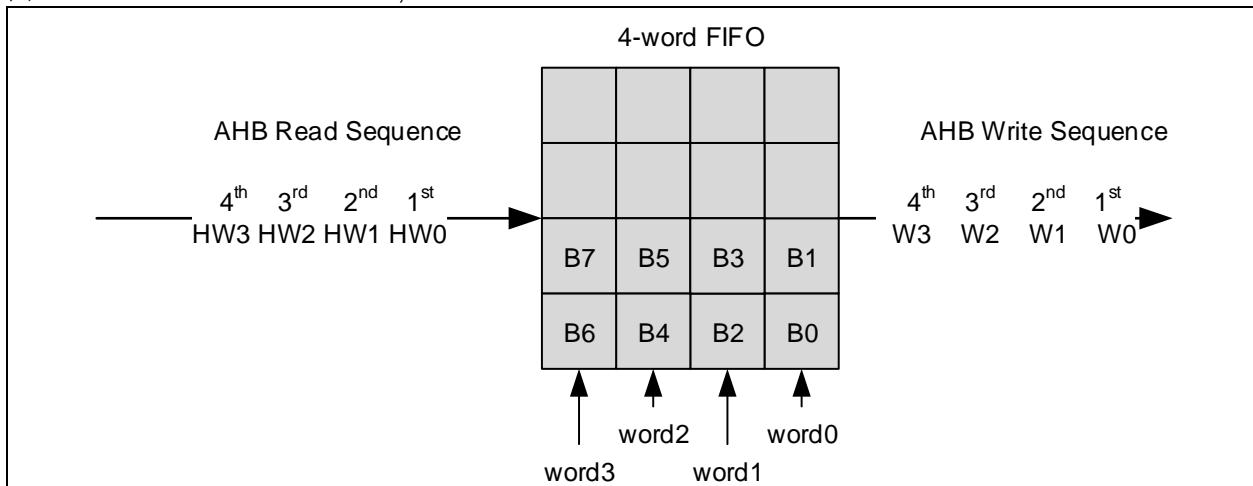
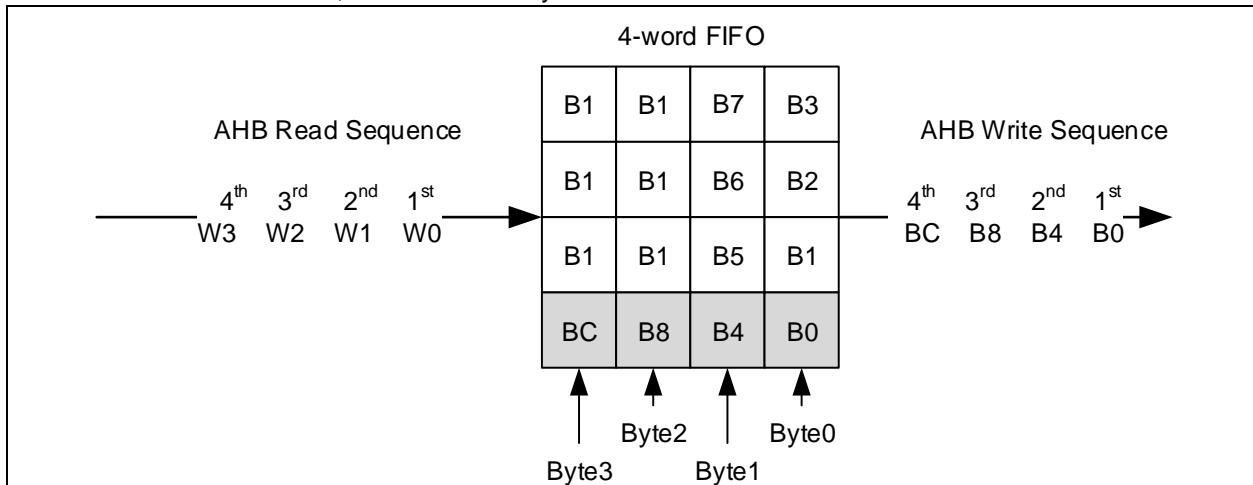


图 9-5 PWIDHT: word, MWIDHT: byte



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

9.4 多路复用器 (DMAMUX)

DMAMUX 在外设和 DMA 控制器之间路由 DMA 请求/确认。

DMA 控制器通过 DMA_MUXSEL 中的 TBL_SEL 位选择 DMA 映射表。每个 DMA 控制器流可从弹性映射表中选择一个唯一的 DMA 请求。在弹性映射中，每个通道可以通过 DMA_MUXCxCTRL 中的 REQSEL [6: 0] 字段旁路或同步来自外设或生成器的 127 个可能的通道请求。

9.4.1 DMAMUX 功能描述

DMAMUX 具有两个功能：请求生成器和请求多路复用器。

每个 DMAMUX 生成器通道 x 在 DMA_MUXGxCTRL 中都只有一个使能 GEN 位。通过 SIGSEL 字段选择 DMAMUX 生成器的触发输入，通常 DMA 请求的数量为 GREQCNT + 1。通过 DMA_MUXGxCTRL 中的 GPOL 字段选择触发事件，该事件可以是上升沿，下降沿或任一沿。

每个 DMAMUX 多路复用器流 x 来自弹性映射 all_req [127: 1] 请求组输入。

在弹性映射中，可以通过 DMA_MUXSxCTRL 中的 SYNCEN 位来同步选定的 DMA 请求输入。当通道处于同步模式时，通过 DMA_MUXSxCTRL 中的 SYNCSEL 字段选择同步输入，然后一旦通过 DMA_MUXSxCTRL 中的 SYNCPOL [1: 0] 字段检测到同步输入的有效沿，则所选的 DMA 请求输入将传播到多路复用器请求输出 chx_mux_req [7: 0]。另外，当通过 DMA_MUXCxCTRL 中的 EGE 位启用流的事件生成时，可编程请求计数器 GREQCNT 用于请求输出生成和事件生成。

图 9-6 DMAMUX 框图

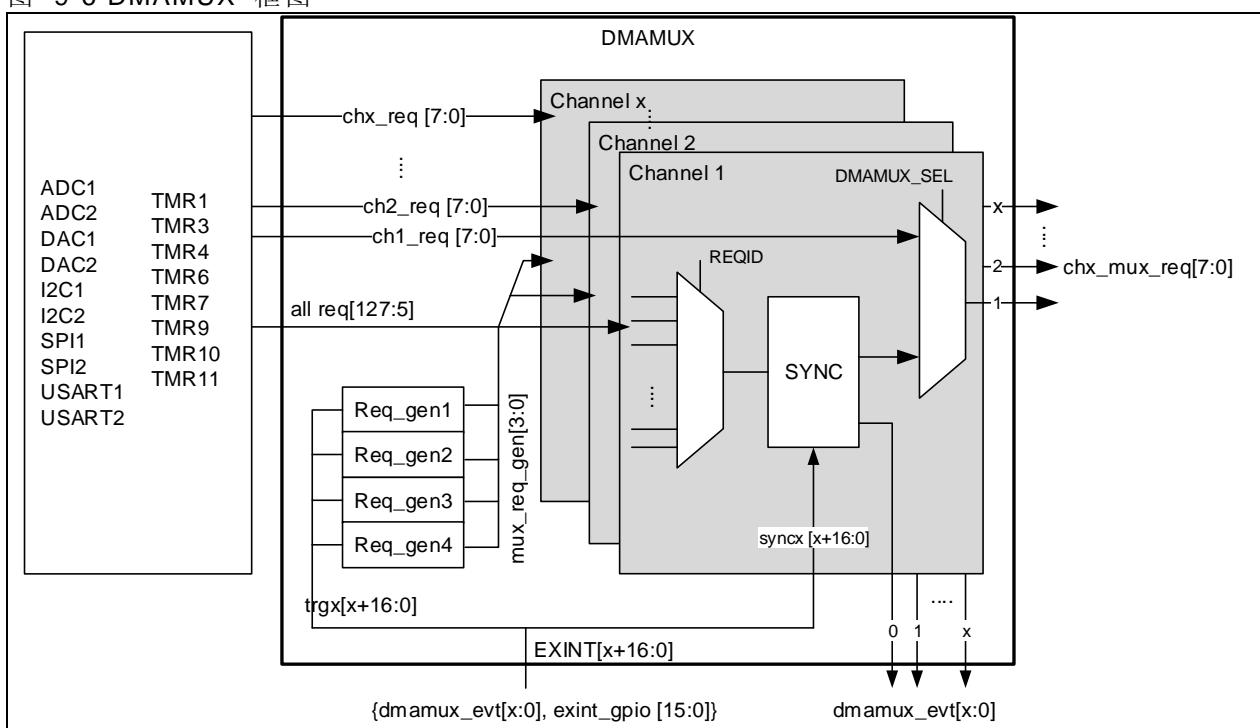


表 9-3 DMA请求弹性映射

CHx_SRC	请求来源	CHx_SRC	请求来源	CHx_SRC	请求来源	CHx_SRC	请求来源
1	DMA_MUXREQG1	33	reserved	65	TMR3_OVERFLOW	97	reserved
2	DMA_MUXREQG2	34	reserved	66	TMR3_TRIG	98	reserved
3	DMA_MUXREQG3	35	reserved	67	TMR4_CH1	99	reserved
4	DMA_MUXREQG4	36	ADC2	68	TMR4_CH2	100	reserved
5	ADC1	37	reserved	69	TMR4_CH3	101	reserved
6	DAC1	38	reserved	70	TMR4_CH4	102	reserved
7	reserved	39	reserved	71	TMR4_OVERFLOW	103	reserved
8	TMR6_OVERFLOW	40	reserved	72	reserved	104	reserved
9	TMR7_OVERFLOW	41	DAC2	73	reserved	105	reserved
10	SPI1_RX	42	TMR1_CH1	74	reserved	106	reserved
11	SPI1_TX	43	TMR1_CH2	75	reserved	107	reserved
12	SPI2_RX	44	TMR1_CH3	76	reserved	108	reserved
13	SPI2_TX	45	TMR1_CH4	77	reserved	109	reserved
14	reserved	46	TMR1_OVERFLOW	78	TMR9_CH1	110	reserved
15	reserved	47	TMR1_TRIG	79	TMR9_OVERFLOW	111	reserved
16	I2C1_RX	48	TMR1_HALL	80	TMR9_TRIG	112	reserved
17	I2C1_TX	49	reserved	81	TMR9_HALL	113	reserved
18	I2C2_RX	50	reserved	82	TMR10_CH1	114	reserved
19	I2C2_TX	51	reserved	83	TMR10_OVERFLOW	115	reserved
20	reserved	52	reserved	84	TMR11_CH1	116	reserved
21	reserved	53	reserved	85	TMR11_OVERFLOW	117	reserved
22	reserved	54	reserved	86	reserved	118	reserved
23	reserved	55	reserved	87	reserved	119	reserved
24	USART1_RX	56	reserved	88	reserved	120	reserved
25	USART1_TX	57	reserved	89	reserved	121	reserved
26	USART2_RX	58	reserved	90	reserved	122	reserved
27	USART2_TX	59	reserved	91	reserved	123	reserved
28	reserved	60	reserved	92	reserved	124	TMR9_CH2
29	reserved	61	TMR3_CH1	93	reserved	125	reserved
30	reserved	62	TMR3_CH2	94	reserved	126	reserved
31	reserved	63	TMR3_CH3	95	reserved	127	TMR4_TRIG
32	reserved	64	TMR3_CH4	96	reserved		

表 9-4 DMAMUX EXINT LINE 用于触发输入和同步输入

EXINT LINE	来源	EXINT LINE	来源	EXINT LINE	来源	EXINT LINE	来源
0	exint_gpio[0]	8	exint_gpio[8]	16	DMA_MUXevt1	24	reserved
1	exint_gpio[1]	9	exint_gpio[9]	17	DMA_MUXevt2	25	reserved
2	exint_gpio[2]	10	exint_gpio[10]	18	DMA_MUXevt3	26	reserved
3	exint_gpio[3]	11	exint_gpio[11]	19	DMA_MUXevt4	27	reserved
4	exint_gpio[4]	12	exint_gpio[12]	20	DMA_MUXevt5	28	reserved
5	exint_gpio[5]	13	exint_gpio[13]	21	DMA_MUXevt6	39	reserved
6	exint_gpio[6]	14	exint_gpio[14]	22	DMA_MUXevt7	30	reserved
7	exint_gpio[7]	15	exint_gpio[15]	23	reserved	31	reserved

9.4.2 DMAMUX 溢出中断

在 DMAMUX 请求生成中，如果在请求生成器计数器 GREQCNT 下溢之前发生了新的触发输入，则在 DMA_MUXGSTS 中声明请求触发溢出标志位 TRGOVFX。通过将 DMA_MUXGCLR 中的相关位 TRGOVFCx 置 1，可以清除溢出标志 TRGOVFX。

如果在 DMA_MUXGxCTRL 中设置了中断允许位 TRGOVIEN，则 DMAMUX 请求触发溢出标志会产生一个中断。

在 DMAMUX 同步模式下，如果在请求计数器 REQCNT 下溢之前发生了新的同步输入，则在 DMA_MUXSYNCSTS 中声明同步溢出标志位 SYNCOVFX。通过将 DMA_MUXSYNCCLR 中的相关位 SYNCOVFCx 置 1，可以清除溢出标志 SYNCOVFX。

如果在 DMA_MUXSxCTRL 中设置了中断允许位 SYNCOVIEN，则 DMAMUX 同步溢出标志会产生一个中断。

图 9-7 DMAMUX 请求同步模式

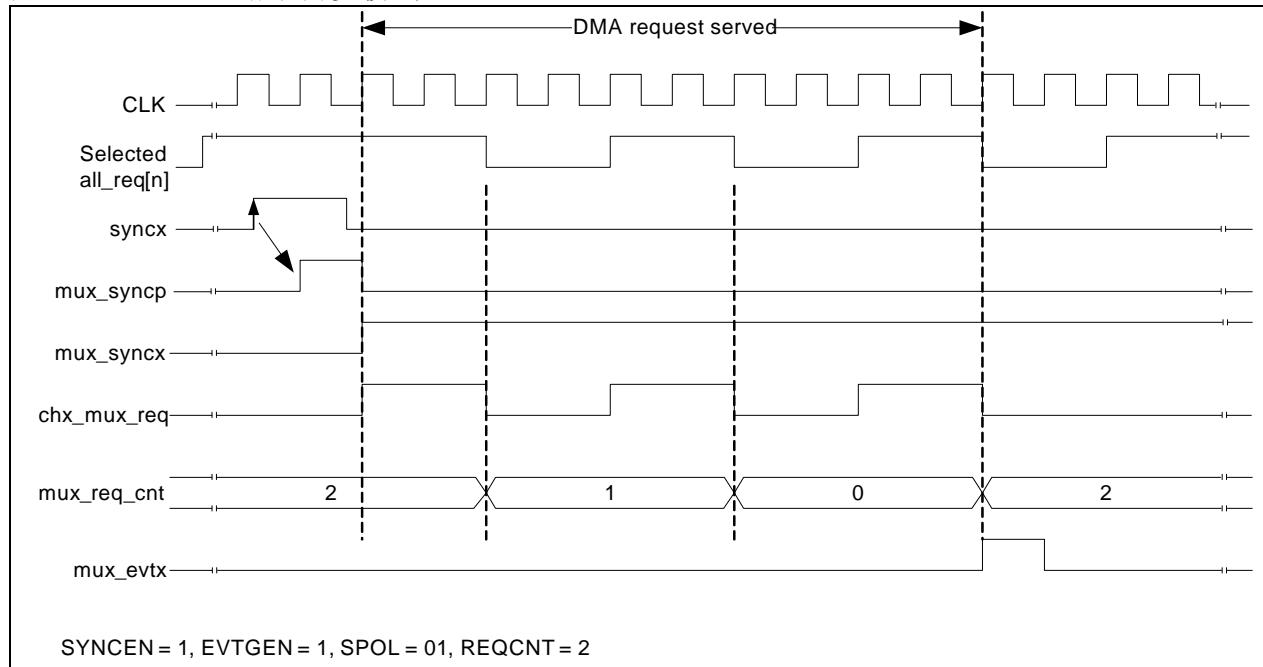
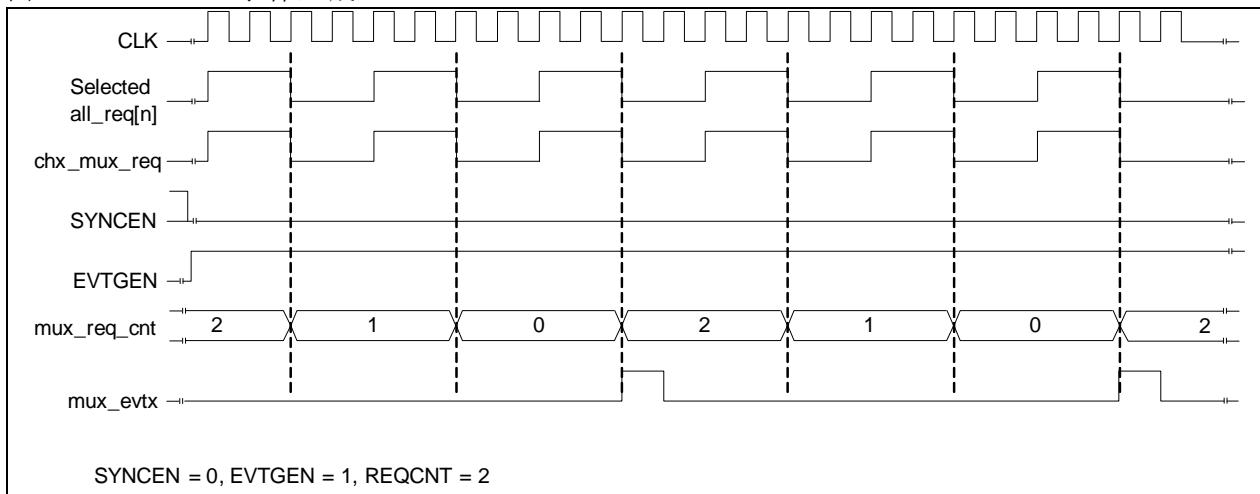


图 9-8 DMAMUX 事件生成



9.5 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 9-5 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
DMA_C6CTRL	0x6c	0x0000 0000
DMA_C6DTCNT	0x70	0x0000 0000

DMA_C6PADDR	0x74	0x0000 0000
DMA_C6MADDR	0x78	0x0000 0000
DMA_C7CTRL	0x80	0x0000 0000
DMA_C7DTCNT	0x84	0x0000 0000
DMA_C7PADDR	0x88	0x0000 0000
DMA_C7MADDR	0x8c	0x0000 0000
DMA_MUXSEL	0x100	0x0000 0000
DMA_MUXC1CTRL	0x104	0x0000 0000
DMA_MUXC2CTRL	0x108	0x0000 0000
DMA_MUXC3CTRL	0x10c	0x0000 0000
DMA_MUXC4CTRL	0x110	0x0000 0000
DMA_MUXC5CTRL	0x114	0x0000 0000
DMA_MUXC6CTRL	0x118	0x0000 0000
DMA_MUXC7CTRL	0x11c	0x0000 0000
DMA_MUXG1CTRL	0x120	0x0000 0000
DMA_MUXG2CTRL	0x124	0x0000 0000
DMA_MUXG3CTRL	0x128	0x0000 0000
DMA_MUXG4CTRL	0x12c	0x0000 0000
DMA_MUXSYNCSTS	0x130	0x0000 0000
DMA_MUXSYNCLR	0x134	0x0000 0000
DMA_MUXGSTS	0x138	0x0000 0000
DMA_MUXGCLR	0x13c	0x0000 0000

9.5.1 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
位 31: 28	保留	0x0	resd	保持默认值。
位 27	DTERRF7	0x0	ro	通道 7 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 26	HDTF7	0x0	ro	通道 7 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 25	FDTF7	0x0	ro	通道 7 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 24	GF7	0x0	ro	通道 7 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 23	DTERRF6	0x0	ro	通道 6 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 22	HDTF6	0x0	ro	通道 6 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 21	FDTF6	0x0	ro	通道 6 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 20	GF6	0x0	ro	通道 6 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 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

9.5.2 DMA标志清除寄存器 (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
位 31: 28	保留	0x0	resd	保持默认值。
位 27	DTERRFC7	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 26	HDTFC7	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 25	FDTFC7	0x0	rw1c	清除通道 7 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 FDTF7 标志
位 24	GFC7	0x0	rw1c	清除通道 7 的全局中断标志 (Global flag clear) 0: 无效 1: 清除 DMA_STS 寄存器中 DTERRF7、HDTF7、FDTF7 和 GF7 标志
位 23	DTERRFC6	0x0	rw1c	清除通道 6 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 DTERRF6 标志
位 22	HDTFC6	0x0	rw1c	清除通道 6 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 HDTF6 标志
位 21	FDTFC6	0x0	rw1c	清除通道 6 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 FDTF6 标志
位 20	GFC6	0x0	rw1c	清除通道 6 的全局中断标志 (Global flag clear) 0: 无效 1: 清除 DMA_STS 寄存器中 DTERRF6、HDTF6、FDTF6 和 GF6 标志
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 3 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 3 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_ISTS 寄存器中 DTERRF1、HDTF1 FDTF1 和 GF1 标志
-----	------	-----	------	--

9.5.3 DMA通道x配置寄存器 (DMA_CxCTRL) (x = 1…7)

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 1: 允许半数据传输中断
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

9.5.4 DMA通道x数据传输量寄存器 (DMA_CxDTCNT) (x = 1…7)

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。

9.5.5 DMA通道x外设地址寄存器 (DMA_CxPADDR) (x = 1…7)

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) 外设数据寄存器的基地址，作为数据传输的源或目标。

9.5.6 DMA通道x存储器地址寄存器 (DMA_CxMADDR) (x = 1…7)

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。

9.5.7 DMAMUX选择寄存器 (DMA_MUXSEL)

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: 1	保留	0x0000 0000	resd	保持默认值。
位 0	TBL_SEL	0x0	rw	多路复用器表选择 (multiplexer table select) 0x1: 弹性映射表

9.5.8 DMAMUX通道x控制寄存器 (DMA_MUXCxCTRL) (x = 1...7)

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 31:25	保留	0x00	resd	保持默认值。
位 28: 24	SYNCSEL	0x00	rw	同步标识 (Synchronization select)
位 23: 19	REQCNT	0x00	rw	DMA 请求数 (Request count) 定义同步事件之后向 DMA 控制器的 DMA 请求数，和/或生成输出事件之前的 DMA 请求数。 仅当 SYNCEN 和 EVTGEN 位均为 LOW 时，才写入字段保留。
位 18: 17	SYNCPOL	0x0	rw	同步极性 (Synchronization polarity) 定义所选同步输入的边缘极性。 0x0: 无事件 0x1: 上升沿 0x2: 下降沿 0x3: 上升沿和下降沿
位 16	SYNCEN	0x0	rw	同步启用 (Synchronization enable) 0: 禁用同步 1: 同步使能
位 15: 10	保留	0x00	resd	保持默认值。
位 9	EVTGEN	0x0		启用事件 (Event generate enable) 0: 禁用事件生成 1: 启用事件生成
位 8	SYNCOVIEN	0x0		同步溢出中断启用 (Synchronization overrun interrupt enable) 0: 禁止中断 1: 中断使能
位 7	保留	0x0	resd	保持默认值。
位 6: 0	REQSEL	0x00		DMA 请求标识 (DMA Request select) 选择输入的 DMA 请求。请参考 DMAMUX 表以了解资源的多路复用器输入的分配。

9.5.9 DMAMUX生成器x控制寄存器 (DMA_MUXGxCTRL) (x = 1…4)

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: 24	保留	0x00	resd	保持默认值。
位 23: 19	GREQCNT	0x00	rw	DMA 请求生成数 (Request generation count) 定义触发事件后将生成的 DMA 请求 (GNBREQ + 1) 的数量。 仅当禁用 GEN 位时，才写入字段保留。
位 18: 17	GPOL	0x0	rw	DMA 请求生成器触发极性 (Generation polarity) 定义所选触发输入的边缘极性。 0x0: 无事件 0x1: 上升沿 0x2: 下降沿 0x3: 上升沿和下降沿
位 16	GEN	0x0	rw	DMA 请求生成器 (DMA request generation enable) 0: 禁用 DMA 请求生成器 1: 启用 DMA 请求生成器
位 15: 9	保留	0x00	resd	保持默认值。
位 8	TRGOVIEN	0x0	rw	触发溢出中断使能 (Trigger overrun interrupt enable) 0: 禁止中断 1: 中断使能
位 7: 5	保留	0x0	resd	保持默认值。
位 4: 0	SIGSEL	0x00	rw	信号识别 (Signal select) 选择用于 DMA 请求生成器的 DMA 触发输入。

9.5.10 DMAMUX通道同步状态寄存器 (DMA_MUXSYNCSTS)

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: 7	保留	0x0000 00	resd	保持默认值。
位 6: 0	SYNCOVF	0x00	ro	同步溢出中断标志 (Synchronization overrun interrupt flag) 当 DMA 请求计数器低于 REQCNT 时发生，如果发生新的同步事件，将设置该标志。

9.5.11 DMAMUX通道中断清除标志寄存器 (DMA_MUXSYNCCLR)

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: 7	保留	0x0000 00	resd	保持默认值。
位 6: 0	SYNCOVFC	0x00	rw1c	清除同步溢出中断标志 (Synchronization overrun interrupt flag clear) 在每个位写入 1 以清除 MUXSYNCSTS 寄存器中的相应溢出标志 SYNCOVF。

9.5.12 DMAMUX发生器中断状态寄存器 (DMA_MUXGSTS)

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: 4	保留	0x0000 000	resd	保持默认值。
位 3: 0	TRGOVF	0x00	ro	触发溢出中断标志 (Trigger overrun interrupt flag)。当 DMA 请求计数器低于 GREQCNT 时，如果发生新的触发事件，将设置该标志。

9.5.13 DMAMUX发生器中断清除标志寄存器 (DMA_MUXGCLR)

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: 4	保留	0x0000 000	resd	保持默认值。
位 3: 0	TRGOVFC	0x00	rw1c	清除触发器溢出中断标志 (Trigger overrun interrupt flag clear)。在每个位写入 1，以清除 DMA_MUXGSTS 寄存器中的相应溢出标志 TRGOVF。

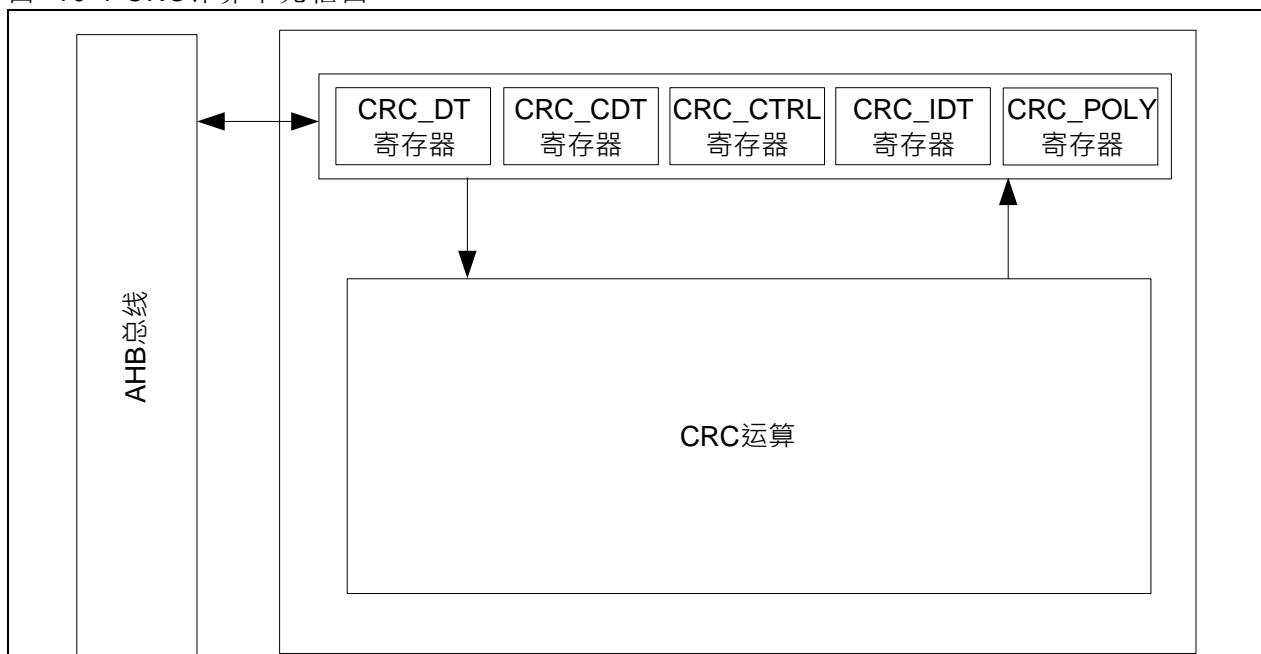
10 CRC 计算单元 (CRC)

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计算单元预设采用CRC32/MPEG-2。用户可以通过软件编程配置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CRC_IDT中的值搬入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CRC_DT寄存器的方式，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。

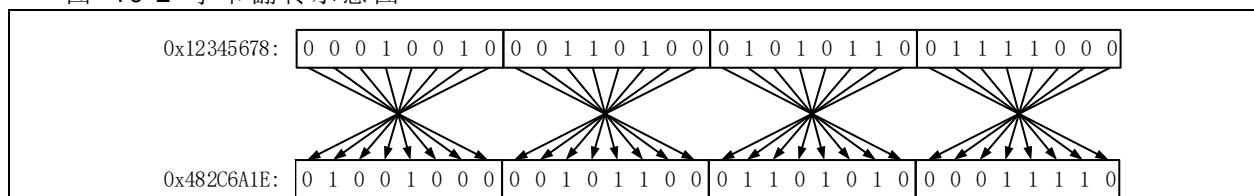
CRC-32/MPEG-2 参数说明

- 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	0x000000	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。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 CRC_CTRL 寄存器的 RST 位产生的 CRC 复位时，初始化寄存器中的数值将作为 CRC_DT 寄存器的初始值写入。

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 I²C 接口

11.1 I²C 简介

I²C总线接口处理微控制器和串行I²C总线之间的通信，支持主机和从机模式，最大通信速度为1Mbit/s(增强快速模式)。

11.2 I²C的主要特点

- I²C 总线
 - 主机和从机模式
 - 多主机功能
 - 标准模式（standard mode, 最高 100kHz）、快速模式（fast mode, 最高 400kHz）和增强快速模式（fast mode plus, 最高 1 MHz）
 - 7-bit 和 10-bit 地址模式
 - 两组 7 位从地址（2 个地址，其中一个可屏蔽）
 - 广播呼叫
 - 可编程数据建立和保持时间
 - 时钟延展功能
- 支持 DMA 功能
- 可编程数字噪声滤波器
- 支持 SMBus 2.0 版协议
 - PEC 产生和检查
 - 命令和数据的应答控制
 - ARP(地址解析协议)
 - 主机功能
 - 设备功能
 - SMBus 提醒功能
 - 超时检测
 - 空闲检测
- PMBus

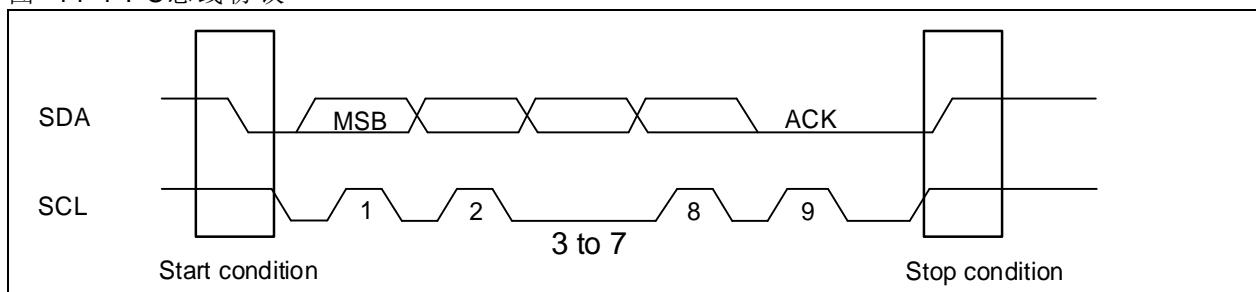
11.3 I²C总线特性

I²C 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，增强快速模式可达到 1MHz。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。

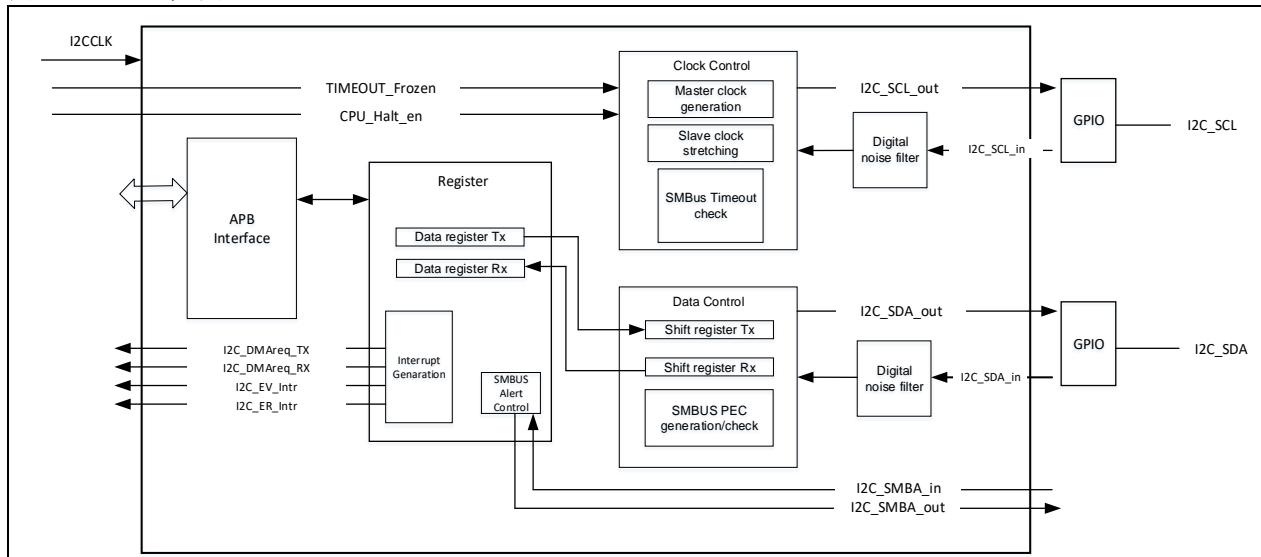
图 11-1 I²C总线协议



11.4 I²C 接口

I²C 接口的功能框图示于下图：

图 11-2 I²C 框图



1. 接口工作模式

I²C 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，当产生了一个起始信号后，I²C 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，I²C 总线接口自动返回为从机模式。

2. 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

3. 数字滤波功能

SCL 和 SDA 总线上均有数字滤波器，数字滤波器可以有效降低总线上的干扰，通过配置 I²C_CTRL1 的 DFLT[3: 0]位（范围 0~15），可以启用数字滤波器，滤波时间为 DFLT × T_{I²C_CLK}，当 I²C 启用时不允许更改数字滤波器的配置。

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 (ADDR1MODE=0)
 - 单地址模式 ADDR1EN=1, ADDR2EN=0：此时只匹配OADDR1；
 - 双地址模式 ADDR1EN=1, ADDR2EN=1：此时匹配OADDR1和OADDR2。
- 10 位地址模式 (ADDR1MODE=1)
 - 只支持单地址模式 ADDR1EN=1, ADDR2EN=0，匹配OADDR1

从机地址屏蔽功能

从机地址 2 (OADDR2) 支持地址屏蔽功能，通过设置 ADDR2MASK[2: 0]启用地址屏蔽功能

- 0：匹配地址位[7: 1]；

- 1: 只匹配地址位[7: 2];
- 2: 只匹配地址位[7: 3];
- 3: 只匹配地址位[7: 4];
- 4: 只匹配地址位[7: 5];
- 5: 只匹配地址位[7: 6];
- 6: 只匹配地址位[7];
- 7: 所有非I²C保留地址都会响应。

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 (DEVADDREN = 1) 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 (HADDREN = 1) 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 在设备模式下, 并且拉低 SMBbus ALERT 引脚 (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 ADDR置 1, SDIR 指示传输方向
 - 如果SDIR=0从机进入接收模式, 开始接收数据
 - 如果SDIR=1从机进入发送模式, 开始发送数据

5. 时钟延展功能

在默认情况下, 从机的时钟延展功能是打开的, 即 I2C_CTRL1 的 STRETCH 位为 0, 从机可以根据需要将 SCL 信号拉低, 延展 SCL 信号的低电平时间以便执行软件的操作, 如果主机不支持时钟延展功能, 则 I2C_CTRL1 的 STRETCH 位需配置为 1。需要注意的是 I²C 从机的时钟延展模式必须在使能 I²C 外设之前配置。

从机时钟延展

I²C 从机在以下情况下延展 SCL 时钟:

- 地址接收阶段: 从机接收到的地址与启用的本机地址匹配 (I2C_STS 的 ADDR=1) 时会将 SCL 线拉低, 直到软件将 I2C_CLR 的 ADDRC 位置 1 清掉 ADDR 时释放时钟延展;
- 数据接收阶段: I2C_RXDT 寄存器的数据尚未被读取, 移位寄存器也接收到一个新的数据, 这时会将 SCL 线拉低直到 I2C_RXDT 寄存器的数据被读取;
- 数据发送阶段: ADDR 被清除后未写入数据时, I2C_STS 的 TDBE= 1, 这时会将 SCL 线拉低直到数据写入 I2C_TXDT;
- 数据发送阶段: 上一笔的数据传输已完成, 尚未有新的数据写到 I2C_TXDT, 这时会将 SCL 线拉低直到数据写入 I2C_TXDT;
- 当启用从机字节控制模式(I2C_CTRL1 的 SCTRL 为 1)且 I2C_CTRL2 的 RLDE 位为 1, 如果 TCRLD = 1, 代表此时最后一个数据字节已经传输完成, 直到向 I2C_CTRL2 的 CNT 位写入一个非零值, 硬件自动将 TCRLD 清 0, 从机释放 SCL 线。

从机不带时钟延展

当 I2C_CTRL1 的 STRETCH 位为 1 时, I²C 从机不会延展 SCL 信号, 需要特别注意下列情况:

- 地址接收阶段: 从机接收到的地址与启用的本机地址匹配 (I2C_STS 的 ADDR=1) 时 SCL 时钟不会延展;
- 数据接收阶段: 下一个字节的 ACK 发生前, 尚未将数据从 I2C_RXDT 寄存器读走, 会发生上溢, I2C_STS 的 OUF 位会被置 1;
- 数据发送阶段: 上一笔的数据传输已完成, 尚未有新的数据写到 I2C_TXDT, 会发生下溢, I2C_STS 的 OUF 位会被置 1。

11.4.1 I²C时序控制

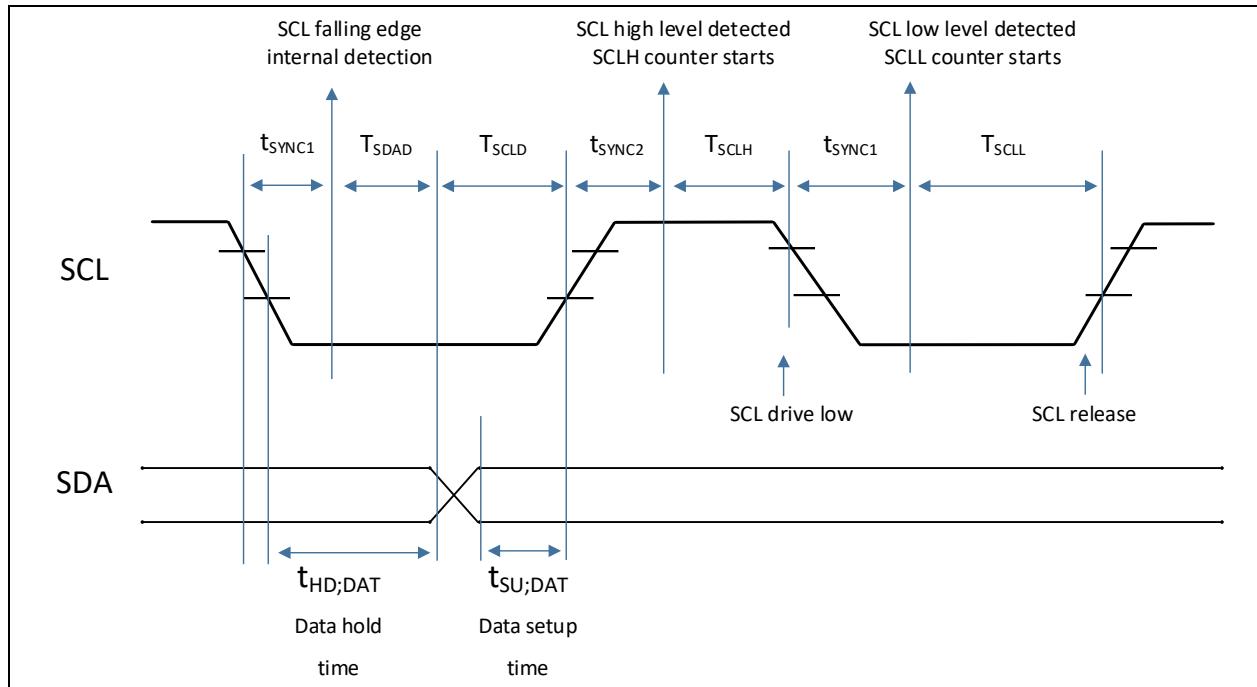
I²C 内核由 I2C_CLK 提供时钟, I2C_CLK 由 PCLK1 提供, PCLK1 周期需满足小于 4/3 SCL 周期。

通过 I2C_CLKCTRL 寄存器的各个位，配置各个时序。

- DIV[7: 0]: I²C 时钟分频；
- SDAD[3: 0]: 数据保持时间 ($t_{HD;DAT}$)；
- SCLD[3: 0]: 数据建立时间 ($t_{SU;DAT}$)；
- SCLH[7: 0]: SCL 高电平时间；
- SCLL[7: 0]: SCL 低电平时间。

注：当 I²C 启用时不允更改时序配置。

图 11-3 建立和保持时间



通过配置 I2C_CLKCTRL 的 DIV[7: 0]、SDAD[3: 0]、SCLD[3: 0]，可以灵活的配置数据保持时间($t_{HD;DAT}$) 和数据建立时间($t_{SU;DAT}$)

- 数据保持时间 ($t_{HD;DAT}$)：SCL 下降沿到 SDA 输出的延时

$$t_{HD;DAT} = T_{SDAD} + t_{SYNC}$$

$$T_{SDAD} = SDAD \times (DIV + 1) \times T_{I2C_CLK}$$

$$t_{SYNC} = (DFLT + 3) \times T_{I2C_CLK} - t_f$$

t_{SYNC} 由以下几部分组成

- SCL下降沿时间 t_f
- 数字滤波器的输入延迟 ($DFLT \times T_{I2C_CLK}$)
- SCL和I2C_CLK的同步延时 (2~3个I2C_CLK)

- 数据建立时间 ($t_{SU;DAT}$)：SDA 输出后到 SCL 上升沿的延时

$$t_{SU;DAT} = SCLD \times (DIV+1) \times T_{I2C_CLK} - t_r$$

在主机模式下，通过配置 I2C_CLKCTRL 的 DIV[7: 0]、SCLH[7: 0]、SCLL[7: 0]，可以灵活的配置 SCL 高电平宽度和低电平宽度

低电平控制：当检测到 SCL 总线为低电平时，内部 SCLL 计数器开始计数，当计数值达到 SCLL 值时，释放 SCL 线，SCL 线变为高电平。

高电平控制：当检测到 SCL 总线为高电平时，内部 SCLH 计数器开始计数，当计数值达到 SCLH 值时，拉低 SCL 线，SCL 线变为低电平，当在高电平期间，如果被外部总线拉低，那么内部 SCLH 计数器停止计数，并开始低电平计数，这为时钟同步提供了条件。

- 高电平宽度：

$$t_{HIGH} = T_{SCLH} = (SCLH + 1) \times (DIV + 1) \times T_{I2C_CLK}$$

- 低电平宽度：

$$t_{LOW} = T_{SCLL} = (SCLL + 1) \times (DIV + 1) \times T_{I2C_CLK}$$

表 11-1 I²C 时间规范

参数		标准模式 Standard mode		快速模式 Fast mode		增强快速模式 Fast mode plus		SMBus 模式	
		最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值
f _{SCL} (kHz)	SCL 时钟频率	100		400		1000		100	
t _{LOW} (us)	SCL 时钟低电平	4.7		1.3		0.5		4.7	
t _{HIGH} (us)	SCL 时钟高电平	4.0		0.6		0.26		4.0	50
t _{H:DAT} (us)	数据保持时间	0		0	0.9	0	0.45	300	
t _{SU:DAT} (ns)	数据建立时间	250		100		50		250	
tr (ns)	SCL、SDA 上升沿	1000		300		120		1000	
tf (ns)	SCL、SDA 下降沿	300		300		120		300	

11.4.2 数据传输管理

I²C 内部有一个管理数据传输的计数器，用来管理通信流程，主要应用于以下地方：

- NACK发送：主机接收模式；
- STOP发送：主机接收/发送模式；
- RESTART产生：主机接收/发送模式；
- ACK控制：从机模式（SMBus下）；
- PEC发送/接收：主机/从机模式。

通常字节传输管理计数器（I2C_CTRL2 的 CNT[7:0]位配置）应用于主模式，在从机模式下是关闭的，只有在 SMBus 模式下，从机为了对每一个字节进行 ACK 的控制和 PEC 的接收，才会使用此计数器，在 SMBus 模式下从机可以通过 I2C_CTRL1 的 SCTRL 位来启用字节计数器功能。

主机字节控制

I2C_CTRL2 的 CNT[7:0]用于配置发送字节数个数，配置范围为 1~255，当一次传输的数据超过 255 个以上需要将 I2C_CTRL2 的 RLDEN 位置 1，使能重载模式，所以发送数据个数分为≤255 字节和>255 字节两种情况：

- ≤255字节，例如要传输数据个数为100个字节
 - 步骤 1：配置 RLDEN=0，关闭重载模式；
 - 步骤 2：配置 CNT[7:0]=100；
- >255字节，例如要传输数据个数为600个字节
 - 步骤 1：配置 RLDEN=1，使能重载模式；
 - 步骤 2：配置 CNT[7:0]=255，此时还剩下 600-255=345 字节；
 - 步骤 3：当这 255 字节传输完成后，I2C_STS 的 TCRLD 位置 1，然后配置 CNT[7:0]=255，继续传输，此时还剩下 345-255=90；
 - 步骤 4：当这 255 字节传输完成后，I2C_STS 的 TCRLD 位置 1，然后配置 RLDEN=0 禁止重载模式，再配置 CNT[7:0]=90，继续传输。

当是最后一笔数据传输时（重载模式禁止，RLDEN=0），有两种结束数据传输模式

- 自动结束模式（I2C_CTRL2的ASTOPEN=1）
 - 当发送了 CNT[7:0]个字节后，主机自动发送 STOP 条件。
- 软件结束模式（I2C_CTRL2的ASTOPEN=0）
 - 当发送了 CNT[7:0]个字节后，I2C_STS 的 TDC 位置 1，此时 SCL 被拉低，如果使能了 TDCIEN，那么可以产生一个中断，此时软件可以设置 I2C_CTRL2 的 GENSTOP=1 产生一个 STOP 条件，也可以设置 I2C_CTRL2 的 GENSTART=1 产生一个 RESTART 条件，然后软件清除 TDC 标志。

从机字节控制

通过 I2C_CTRL1 的 SCTRL 位来启用从机字节控制功能，该功能可以让从机对接收到的每一个字节进行单独的 ACK/NACK 控制。

- 使用步骤：
 - 设置 SCTRL=1，启用从机字节控制功能；
 - 在从机地址匹配后 (ADDRF=1)，使能重载模式 (RLDEN=1)，并设置 CNT[7:0]=1；
 - 当接收到一个字节后，I2C_STS 的 TCRLD 置 1，从机在 SCL 的第 8 个和第 9 个时钟沿中间拉低 SCL 总线，此时用户读取 RXDT 寄存器，然后根据需要配置 I2C_CTRL2 的 NACKEN 位，来产生一个 ACK 或 NACK；
 - 如果产生一个 NACK，那么通信结束；
 - 如果产生一个 ACK，通信继续，此时写入 CNT[7:0]=1，硬件自动清除 TCRLD 标志，从机释放 SCL 总线，继续接收下一个字节。

当然 CNT[7:0] 值不仅仅局限于 1，例如想接收 8 个数据，但只想控制第 8 个数据的 ACK/NACK，此时就可以设置 CNT[7:0]=8，那么从机就会连续接收 7 个数据，并且回复 ACK，在第 8 个数据接收完后，从机拉低总线，然后同上述步骤，决定 ACK/NACK 的回复。

需要注意的是：要使用从机字节控制模式必须要使能时钟延展 (I2C_CTRL1 的 STRETCH 位=0)。

表 11-2 I²C 配置表

功能	RLDEN	ASTOPEN	SCTRL
主机发送/接收 RESTART	0	0	×
主机发送/接收 STOP	0	1	×
从机接收（每个字节 ACK/NACK 控制）	1	×	1
从机发送/接收（所有字节响应 ACK）	×	×	0

11.4.3 I²C 主机通信流程

1. I²C 时钟初始化（配置 I2C_CLKCTRL 寄存器）

- I²C 时钟分频：DIV[7:0]
- 数据保持时间 (t_{HOLD;DAT})：SDAD[3:0]
- 数据建立时间 (t_{SETUP;DAT})：设置 SCLD[3:0]
- SCL 高电平时间：设置 SCLH[7:0]
- SCL 低电平时间：设置 SCLL[7:0]

该寄存器的配置可以使用 Artery_I2C_Timing_Configuration 时钟配置工具计算。

2. 设置传输字节数

- ≤255 字节
 - 配置 I2C_CTRL2 的 RLDEN=0，关闭重载模式
 - 配置 I2C_CTRL2 的 CNT[7:0]=N
- >255 字节
 - 配置 I2C_CTRL2 的 RLDEN=1，使能重载模式
 - 配置 I2C_CTRL2 的 CNT[7:0]=255
 - 剩余传输字节数 N=N-255

3. 设置传输结束模式

- ASTOPEN=0：软件结束模式，当数据传输完成后，I2C_STS 的 TDC 标志置 1，软件设置 GENSTOP=1 或者 GENSTART=1，发送 STOP 条件或者 START 条件。
- ASTOPEN=1：自动结束模式，当数据传输完成后，自动发送 STOP 条件。

4. 设置从机地址

- 设置寻址的从机地址值 (I2C_CTRL2 的 SADDR)
- 设置从机地址模式 (I2C_CTRL2 的 ADDR10)：
 - ADDR10=0：7 位地址模式
 - ADDR10=1：10 位地址模式

5. 设置传输方向 (I2C_CTRL2 的 DIR)

- DIR=0：主机接收数据

- DIR=1：主机发送数据

6. 开始传输

设置 I2C_CTRL2 的 GENSTART=1，主机开始在总线上发送 START 条件和从机地址，当从机响应了地址之后，主机的 I2C_STS 的 ADDR=1，设置 I2C_CLR 的 ADDRC=1 清除 ADDR 标志后，开始数据传输。

7. 主机发送数据

1. I2C_TXDT 数据寄存器为空，移位寄存器为空，I2C_STS 的 TDIS=1；
2. 向 TXDT 数据寄存器写入数据 1，然后数据将被立即放进移位寄存器；
3. 此时 TXDT 数据寄存器为空，TDIS 又置 1；
4. 向 TXDT 数据寄存器写入数据 2，TDIS 被清 0；
5. 重复 2、3 步骤直到发送 CNT[7:0] 个数据；
6. 如果此时 I2C_STS 的 TCRLD=1（重载模式），分为以下两种情况：
剩余字节数 $N > 255$ ：向 CNT 写入 255， $N=N-255$ ，TCRLD 被自动清 0，传输继续；
剩余字节数 $N \leq 255$ ：关闭重载模式（RLDEN=0），向 CNT 写入 N，TCRLD 被自动清 0，传输继续。

8. 主机接收数据

1. 当正确匹配了从机地址之后，主机的 I2C_STS 的 ADDR=1，设置 I2C_CLR 的 ADDRC=1 清除 ADDR 标志后，开始数据传输；
2. 当收到数据后，RDBF=1，读取 RXDT 数据寄存器，RDBF 被自动清零；
3. 重复步骤 2 直到接收 CNT[7:0] 个数据；
4. 如果此时 I2C_STS 的 TCRLD=1（重载模式），分为以下两种情况：
剩余字节数 $N > 255$ ：向 CNT 写入 255， $N=N-255$ ，TCRLD 被自动清 0，传输继续；
剩余字节数 $N \leq 255$ ：关闭重载模式（RLDEN=0），向 CNT 写入 N，TCRLD 被自动清 0，传输继续。
5. 当在接收到最后一个字节时，主机会自动发送一个 NACK。

9. 结束时序

- 停止条件产生：

软件结束模式（ASTOPEN=0）：此时 I2C_STS 的 TDC 置 1，设置 GENSTOP=1 产生 STOP 条件；

自动结束模式（ASTOPEN=1）：自动产生 STOP 条件。

- 等待产生 STOP 条件，当 STOP 条件产生时，I2C_STS 的 STOPF 置 1，将 I2C_CLR 的 STOPOC 写 1，清除 STOPF 标志，传输结束。

主机在传输过程中，如果收到了 NACK，那么此时 I2C_STS 的 ACKFAIL 置 1，并自动发送 STOP 条件结束通信，无论当前是软件结束模式（ASTOPEN=0）还是自动结束模式（ASTOPEN=1）。

主机发送流程

图 11-4 I²C 主机发送流程图

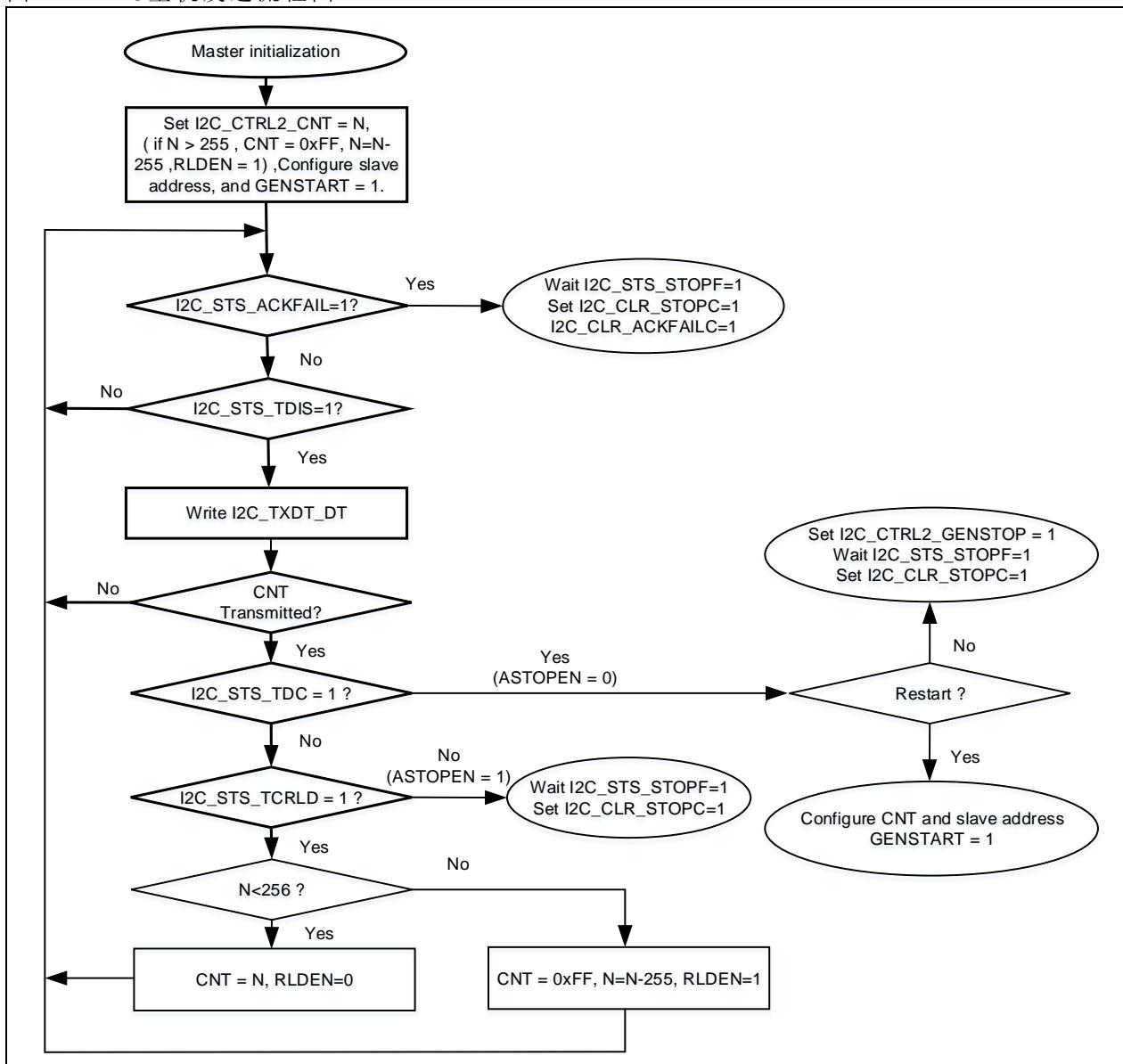
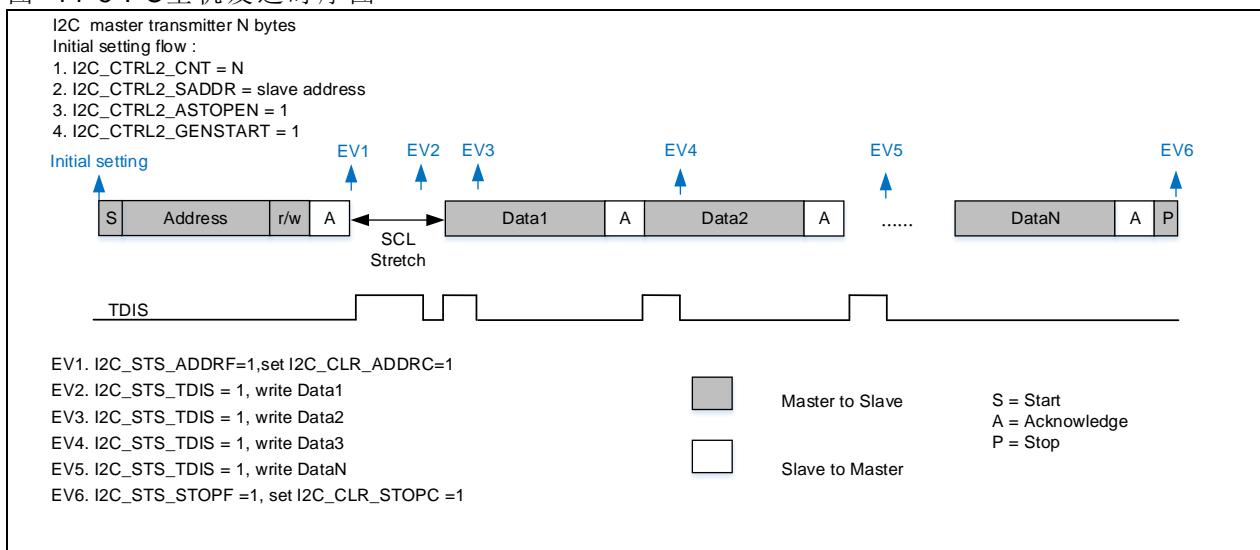


图 11-5 I²C 主机发送时序图



主机接收流程

图 11-6 I²C 主机接收流程图

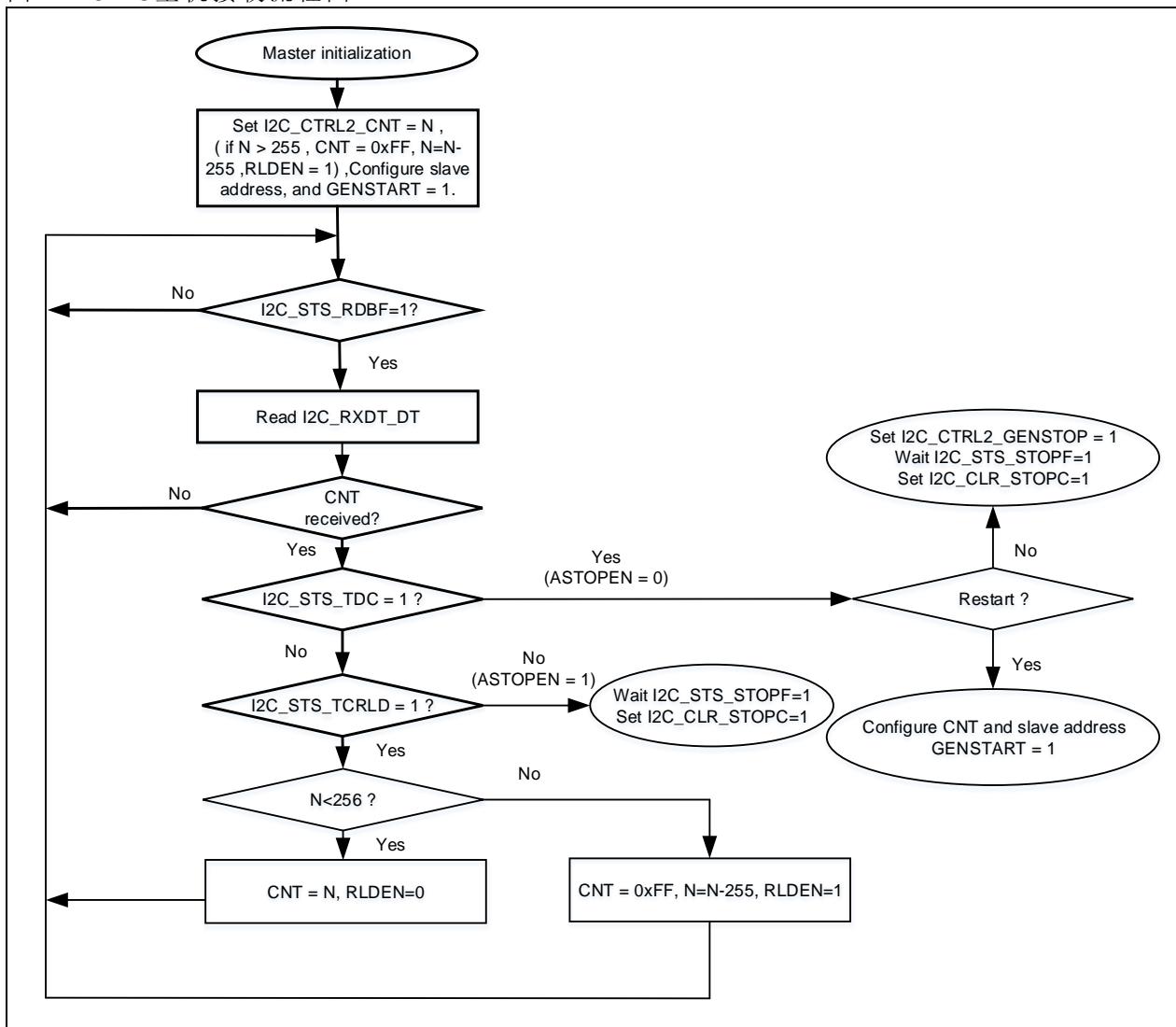
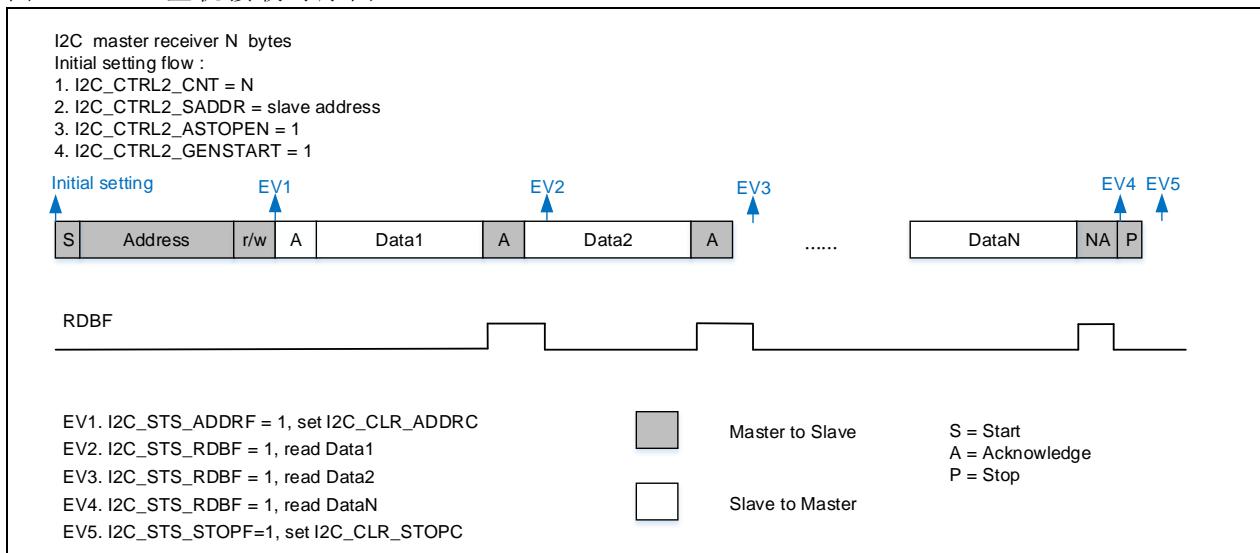


图 11-7 I²C 主机接收时序图



主机特殊传输时序支持

在 10 位地址传输模式下, I2C_CTRL2 的 READH10 用于产生特殊时序, 当 READH10=1 时, 支持如下传输序: 主机先发送数据给从机, 然后再从从机读取数据, 传输时序图如下图所示:

使用方法:

主机在软件结束模式 (ASTOPEN = 0) 下, 发送数据到从机, 当数据发送完成后设置 READH10=0, 然后再从从机接收数据。

图 11-8 10位地址读访问 READH10=1

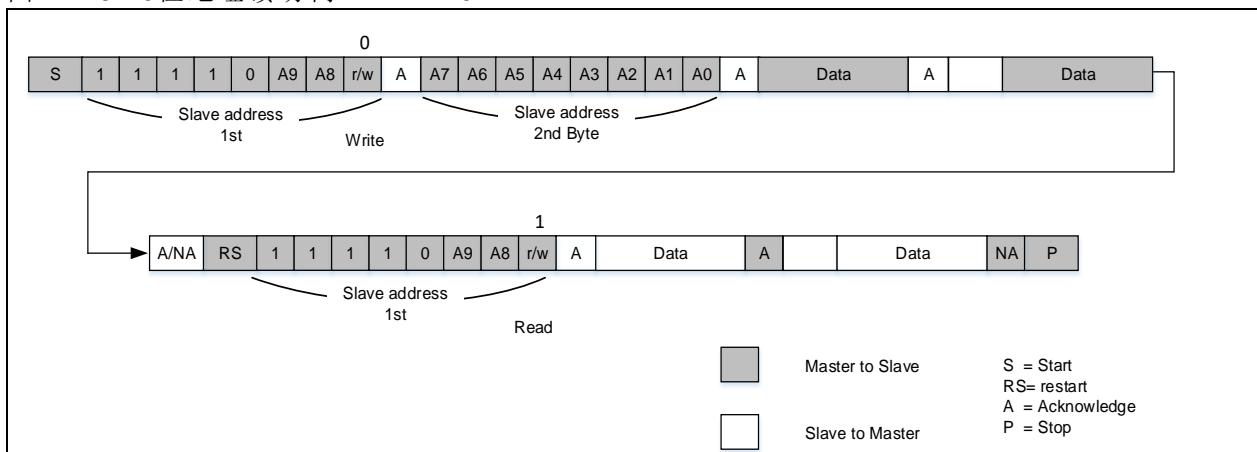
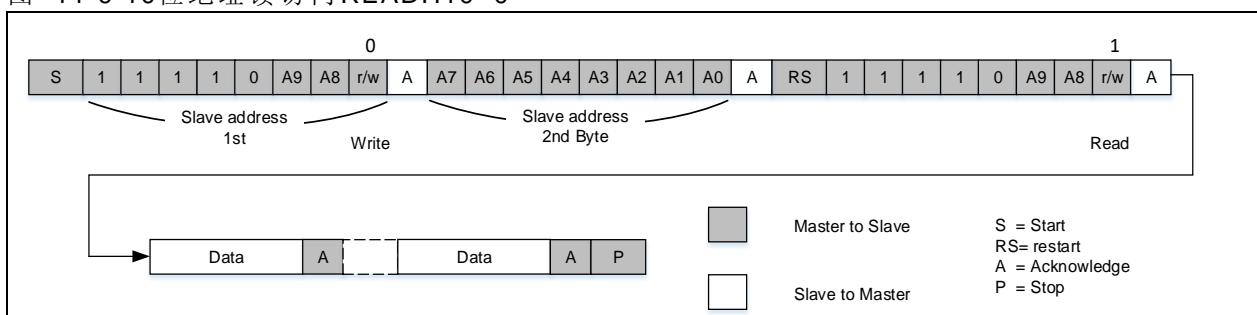


图 11-9 10位地址读访问 READH10=0



11.4.4 I²C从机通信流程

1. I²C时钟初始化（配置I2C_CLKCTRL寄存器）

- I²C 时钟分频: DIV[7: 0]
- 数据保持时间 (t_{HLD;DAT}) : SDAD[3: 0]
- 数据建立时间 (t_{SU;DAT}) : 设置 SCLD[3: 0]

该寄存器的配置可以使用 Artery_I2C_Timing_Configuration 时钟配置工具计算。

2. 设置本机地址1

- 设置地址模式:
 - 7 位地址: 设置 I2C_OADDR1 的 ADDR1MODE = 0
 - 10 位地址: 设置 I2C_OADDR1 的 ADDR1MODE = 1
- 设置地址 1: 设置 I2C_OADDR1 的 ADDR1
- 使能地址 1: 设置 I2C_OADDR1 的 ADDR1EN=1

3. 设置本机地址2

- 设置地址 2: 设置 I2C_OADDR2 的 ADDR2
- 设置地址 2 屏蔽位: 设置 I2C_OADDR2 的 ADDR2MASK
- 使能地址 2: 设置 I2C_OADDR2 的 ADDR2EN=1

需要注意的是, 地址 2 只支持 7 位地址模式, 并且可以通过 ADDR2MASK 位来灵活的屏蔽一些地址位, 使从机响应一些特定的地址, 关于 ADDR2MASK 位的详细用法, 请参照章节 14.2。

在只使用一个地址的情况下，此地址不用配置，只需要配置地址 1。

4. 等待地址匹配

当接收到本机地址后，I2C_STS的ADDRF标志置1，此时可以读取I2C_STS的SDIR位，得到数据传输方向，当SDIR=0时数据传输方向为从机接收数据，当SDIR=1时，数据方向为从机发送数据，通过读取I2C_STS的ADDR[6:0]标志，可以知道接收到的地址是多少，这在双地址模式以及使用了地址2的屏蔽功能模式下，比较有用。

当通过设置I2C_CLR的ADDRC=1清除ADDRF标志后，开始数据传输。

5. 传输数据（从机发送，开启时钟延展，STRETCH=0）

当在地址匹配后：

1. I2C_TXDT 数据寄存器为空，移位寄存器为空，I2C_STS 的 TDIS=1；
2. 向 TXDT 数据寄存器写入数据 1，然后数据将被立即放进移位寄存器；
3. 此时 TXDT 数据寄存器为空，TDIS 又置 1；
4. 向 TXDT 数据寄存器写入数据 2，TDIS 被清 0；
5. 重复 3、4 步骤直到数据发送完成；
6. 等待收到 NACK 条件，当收到 NACK 条件时，I2C_STS 的 ACKFAILF 置 1，将 I2C_CLR 的 ACKFAILC 写 1，清除 ACKFAILF 标志；
7. 等待收到 STOP 条件，当收到 STOP 条件时，I2C_STS 的 STOPF 置 1，将 I2C_CLR 的 STOPC 写 1，清除 STOPF 标志，传输结束。

需要注意的是，在时钟延展关闭（STRETCH=1）的情况下，如果在将要传输数据的第一个 Bit 位开始发送之前，也就是 SDA 边沿产生之前，如果数据还未写入 TXDT 数据寄存器，那么会发生欠载错误，此时 I2C_STS 的 OUF 将会置 1，并将 0xFF 发送到总线。

为了能及时的写入数据，可以在通信开始前，先将数据写入到 DT 寄存器，写入的方式有如下两种：

- 直接写入：软件先将 TDBE 置 1，目的是为了清空 TXDT 寄存器的数据，然后将第一个数据写入 TXDT 寄存器，此时 TDBE 清零；
- 通过中断或者 DMA 写入：软件先将 TDBE 置 1，目的是为了清空 TXDT 寄存器的数据，再将 TDIS 置 1，目的是为了产生一个 TDIS 事件，TDIS 事件可以产生一个中断或者 DMA 请求，此时就可以通过 DMA 或者在中断函数内将数据写入 TXDT 寄存器。

6. 传输数据（从机接收，开启时钟延展，STRETCH=0）

当在地址匹配后：

1. I2C_RXDT 数据寄存器为空，移位寄存器为空，I2C_STS 的 RDBF=0；
2. 当收到数据后，RDBF=1，读取 RXDT 数据寄存器，RDBF 被自动清零；
3. 重复步骤 2 直到所有数据接收完成；
4. 等待收到 STOP 条件，当收到 STOP 条件时，I2C_STS 的 STOPF 置 1，将 I2C_CLR 的 STOPC 写 1，清除 STOPF 标志，传输结束。

在从机接收模式下，可以选择使用从机字节控制模式（SCTRL=1）接收数据，在从机字节控制模式下，可以对每一个接收到的字节进行 ACK/NACK 的控制，这种模式通常使用在 SMBus 协议中，关于从机字节控制模式的详细用法，请参照 11.4.2 数据传输管理章节。

需要注意的是，在时钟延展关闭（STRETCH=1）的情况下，在收到数据后从机应该及时的将数据读走，如果在已经收到 1 个字节后，在下一个字节接收完成之前，如果还未将数据读走，从机将产生过载错误，此时 I2C_STS 的 OUF 将会置 1，并自动回复 NACK。

上述提到的标志，均可通过相应的中断使能位，产生中断，具体的对应关系，请见中断章节。

从机发送流程

图 11-10 I²C从机发送流程图

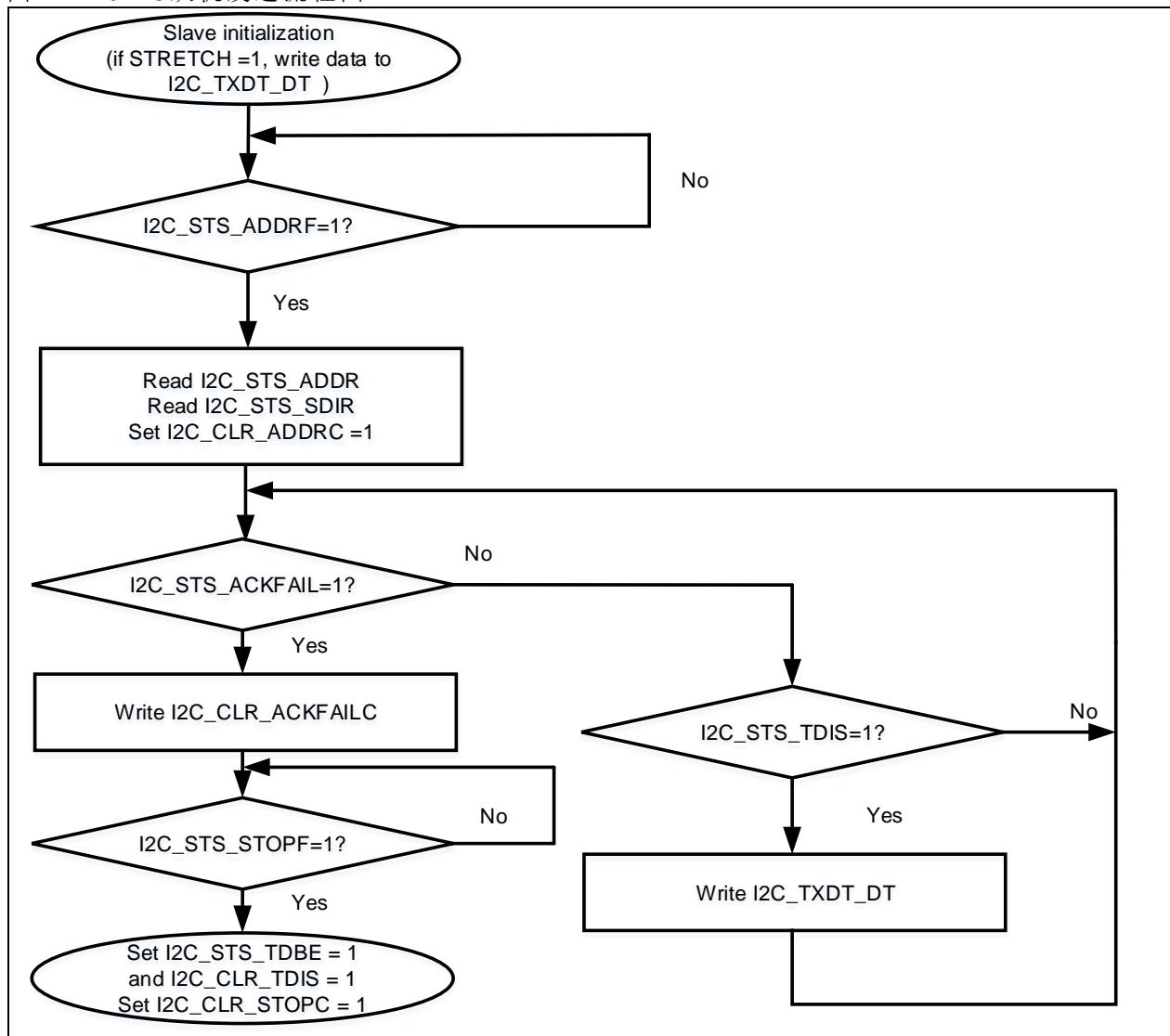
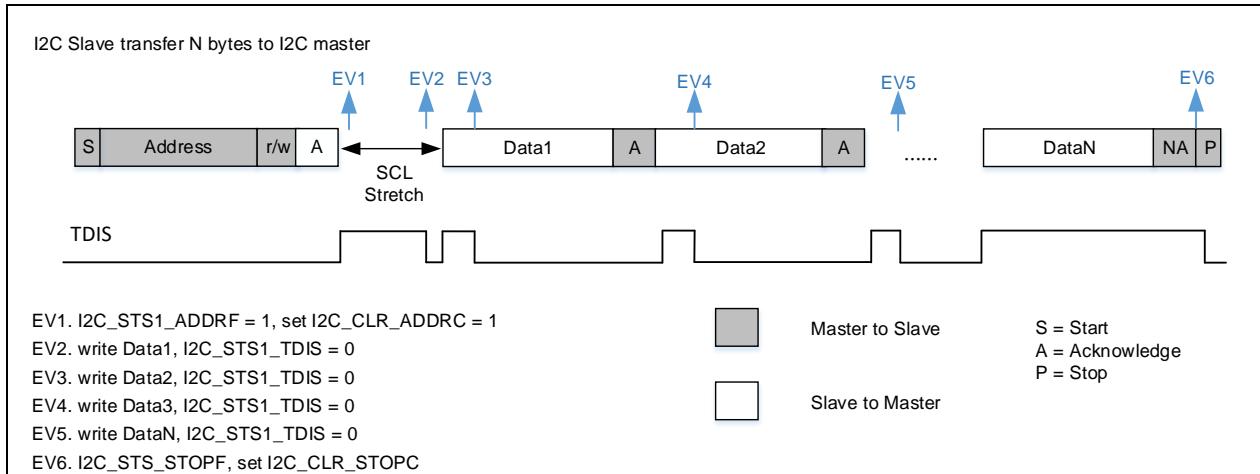
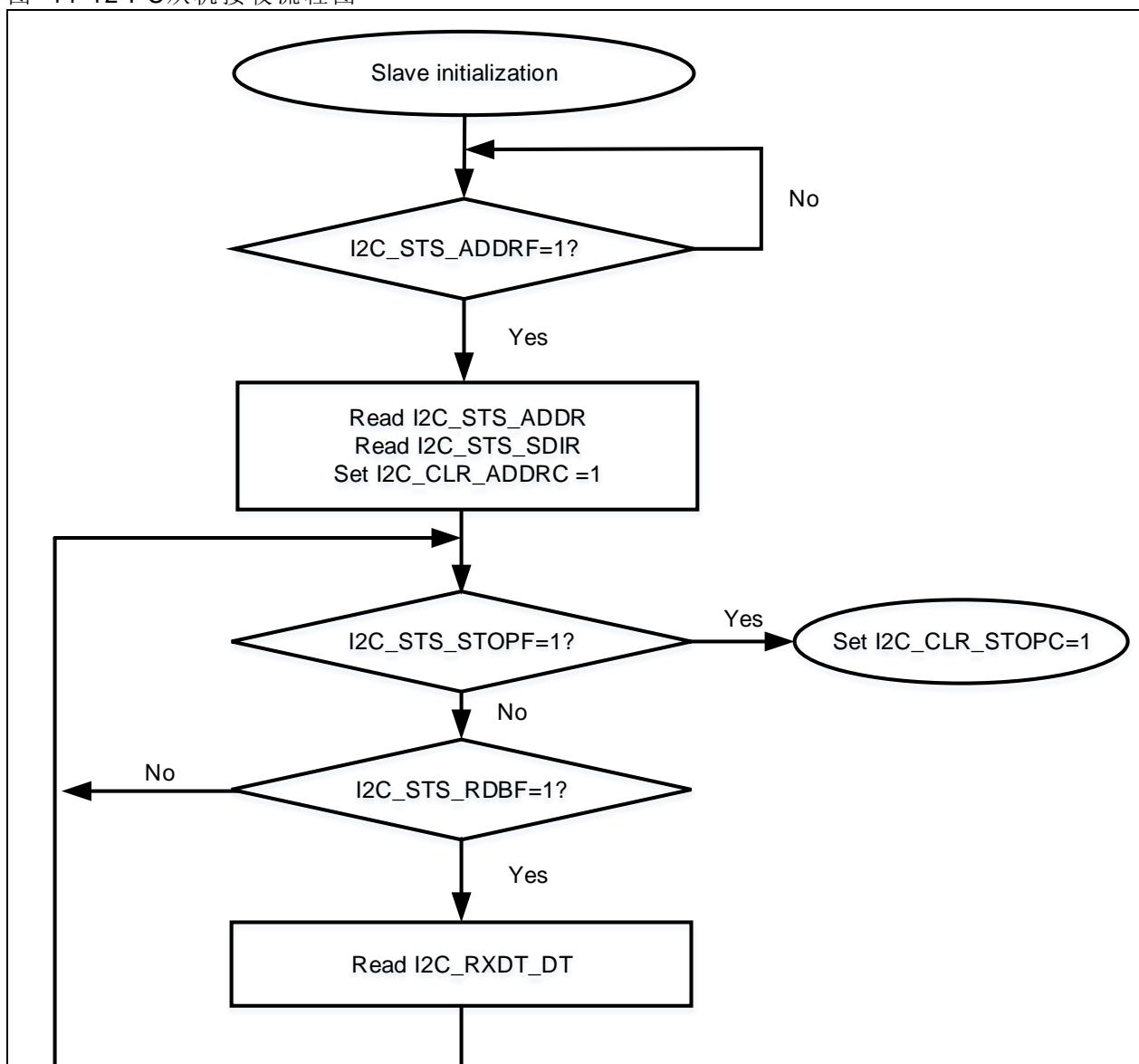
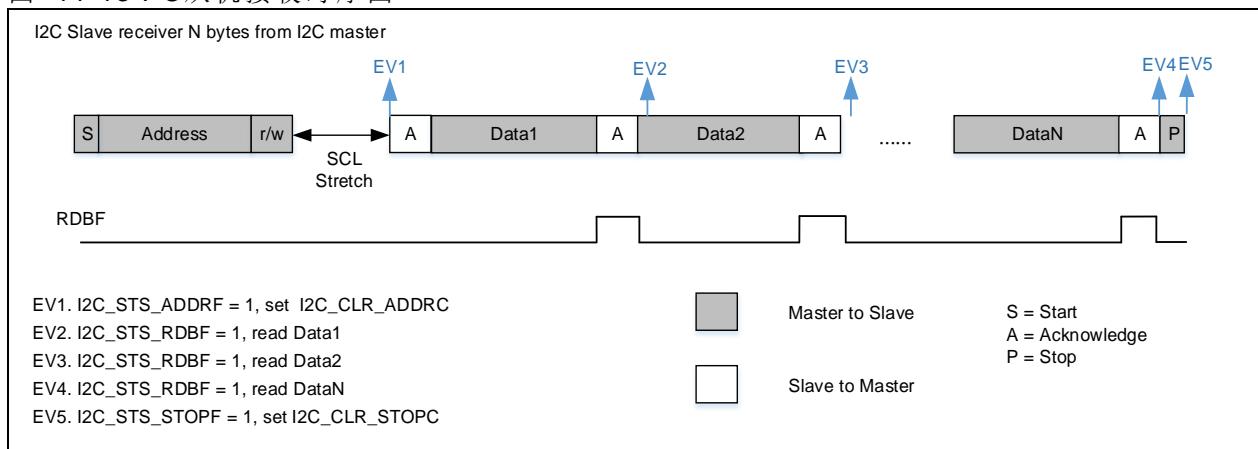


图 11-11 I²C从机发送时序图



从机接收流程

图 11-12 I²C从机接收流程图图 11-13 I²C从机接收时序图

11.4.5 SMBus功能

SMBus即系统管理总线是一双线制总线,基于I²C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 I²C 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,只要在保持一定传递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,I²C则无
2. SMBus传输速度从最小10kHz到最大100kHz,I²C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有超时机制,I²C则无超时机制

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I2C_CTRL1寄存器的DEVADDREN位,可以使能I²C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。

当在主机模式(HADDREN=1)下,I²C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. 等待主机通过发送ARA地址获取从机地址;
3. 上报自己的地址,如果发生了仲裁丢失,继续等待;
4. 地址上报成功,释放ALERT引脚(SMBALERT=0)。

包错误检查

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:

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

PEC计算:当I2C_CTRL1的PECEN=1时启动PEC计算,检验数据包括地址以及数据。

PEC 传输:

- 主机:当I2C_CTRL2的PECTEN=1时,启动PEC传输使能,当数据传输个数达到N-1(CNT=N)了之后,主机会自动发送PEC;
- 从机:当I2C_CTRL2的PECTEN=1时,启动PEC传输使能,当数据传输个数达到N-1(CNT=N)了之后,从机会自动将第N个数据作为PEC检验,如果PEC校验不正确将回

复NACK，并且I2C_STS的PECERR标志将会置1，当在从机发送模式下，无论校验是否正确，都将回复NACK。

SMBus 超时

在 SMBus 协议里面，主要有三种超时检测：

- 低电平超时 $T_{TIMEOUT}$: 时钟线 SCL 单次低电平时间（主机从机都会检测，主动被动拉低都会计算）；
- 从机低电平累积超时 $T_{LOW:SEXT}$: 从机在 START 条件到 STOP 条件之间的主动拉低 SCL 时间总和；
- 主机低电平累积超时 $T_{LOW:MEXT}$: 主机在上一个字节的 ACK 到下一次数据的第 8 个 BIT 位之间（单个字节），主动拉低拉低 SCL 时间总和。

需要注意的是 $T_{LOW:SEXT}$ 和 $T_{LOW:MEXT}$ 和只计算自己主动拉低的时间，外部拉低的时间不计算在内，而 $T_{TIMEOUT}$ 主动、被动拉低都会计算。

表 11-3 SMBus超时 规范

超时类型	最小值	最大值	单位
低电平超时 $T_{TIMEOUT}$	25	35	ms
从机低电平累积超时 $T_{LOW:SEXT}$	-	25	ms
主机低电平累积超时 $T_{LOW:MEXT}$	-	10	ms

I²C 外设内置两个计数器用于检测超时，在 I2C_TIMEOUT 寄存器中配置，当发生超时事件时 I2C_STS 的 TMOUT 置 1，将 I2C_CLR 的 TMOUTC 置 1 可清 0。

- EXTTIME: 用于主机、从机低电平累积超累计超时检测
超时时间 = (EXTTIME + 1) × 2048 × T_{I2C_CLK}
- TOTIME: 时钟电平超时检测，可以通过 TOMODE 位来选择检测电平
TOMODE=0: 低电平超时检测，超时时间 = (TOTIME + 1) × 2048 × T_{I2C_CLK}
TOMODE=1: 高电平超时检测，超时时间 = (TOTIME + 1) × 4 × T_{I2C_CLK}

表 11-4 SMBus超时 检测配置

超时类型	其他配置	使能	超时时间计算
低电平超时 $T_{TIMEOUT}$	TOMODE=0	TOEN=1	(TOTIME + 1) × 2048 × T _{I2C_CLK}
从机低电平累积超时 $T_{LOW:SEXT}$	-	EXTEN=1	(EXTTIME + 1) × 2048 × T _{I2C_CLK}
主机低电平累积超时 $T_{LOW:MEXT}$	-	EXTEN=1	(EXTTIME + 1) × 2048 × T _{I2C_CLK}

从机字节控制

在从机接收模式下，如果需要对接收到的每一个字节进行 ACK/NACK 的控制，可以选择从机接收数据控制模式 (SCTRL=1) 接收数据，关于从机字节控制模式的用法，请参照 11.4.2 数据传输管理章节。

表 11-5 SMBus模式 配置表

传输模式	PECEN	PECTEN	RLDEN	ASTOPEN	SCTRL
主机发送/接收 +STOP	1	1	0	1	-
主机发送/接收 +RESTART	1	1	0	0	-
从机接收	1	1	1	-	1
从机发送	1	1	0	-	-

SMBus 使用流程

1. 设置SMBus默认地址响应:

如果HADDREN=1：响应主机默认地址（0b0001000x）；
如果DEVADDREN=1：响应设备默认地址（0b1100001x）。

2. PEC配置:

3. 如果是从机，可以根据需要使能从机字节控制模式（I2C_CTRL1的SCTRL）；

4. 其他配置和I²C使用配置一样。

需要注意的是各种 SMBus 协议需要由软件来实现，I²C 接口只提供了这些协议的地址识别。

11.4.6 SMBus 主机通信流程

SMBus 主机通信流程和 I²C 主机通信流程类似。

1. I²C时钟初始化（配置I2C_CLKCTRL寄存器）

- I²C 时钟分频：DIV[7: 0]
- 数据保持时间（t_{HOLD;DAT}）：SDAD[3: 0]
- 数据建立时间（t_{SETUP;DAT}）：设置 SCLD[3: 0]
- SCL 高电平时间：设置 SCLH[7: 0]
- SCL 低电平时间：设置 SCLL[7: 0]

该寄存器的配置可以使用 Artery_I2C_Timing_Configuration 时钟配置工具计算。

2. SMBus相关初始化

- 选择为 SMBus 主机：设置 HADDREN=1，响应主机默认地址（0b0001000x）
- 使能 PEC 计算：设置 I2C_CTRL1 的 PECEN=1
- 使能 PEC 传输：设置 I2C_CTRL2 的 PECTEN=1

3. 设置传输字节数

- 配置 I2C_CTRL2 的 RLDE=0，关闭重载模式
- 配置 I2C_CTRL2 的 CNT[7:0]=N

SMBus模式下单次传输字节数<255

4. 设置传输结束模式

- ASTOPEN=0：软件结束模式，当数据传输完成后，I2C_STS 的 TDC 标志置 1，软件设置 GENSTOP=1 或者 GENSTART=1，发送 STOP 条件或者 START 条件。
- ASTOPEN=1：自动结束模式，当数据传输完成后，自动发送 STOP 条件。

5. 设置从机地址

- 设置寻址的从机地址值（I2C_CTRL2 的 SADDR）
- 设置从机地址模式为 7 位地址模式（I2C_CTRL2 的 ADDR10=0）

6. 设置传输方向（I2C_CTRL2的DIR）

- DIR=0：主机接收数据
- DIR=1：主机发送数据

7. 开始传输

设置 I2C_CTRL2 的 GENSTART=1，主机开始在总线上发送 START 条件和从机地址，当从机响应了地址之后，主机的 I2C_STS 的 ADDR=1，设置 I2C_CLR 的 ADDR=1 清除 ADDR 标志后，开始数据传输。

8. 主机发送数据

1. I2C_TXDT 数据寄存器为空，移位寄存器为空，I2C_STS 的 TDIS=1；
2. 向 TXDT 数据寄存器写入数据 1，然后数据将被立即放进移位寄存器；
3. 此时 TXDT 数据寄存器为空，TDIS 又置 1；
4. 向 TXDT 数据寄存器写入数据 2，TDIS 被清 0；
5. 重复 2、3 步骤直到发送 N-1 个数据；
6. 此时主机将自动发送第 N 个数据，也就是 PEC。

9. 主机接收数据

1. 当收到数据后，RDBF=1，读取 RXDT 数据寄存器，RDBF 被自动清零；
2. 重复步骤 1 直到接收 N 个数据，第 N 个数据为 PEC，在接收第 N 个字节，也就是 PEC 时，无论 PEC 是否正确，主机会自动发送一个 NACK。

10. 结束时序

- 停止条件产生：
 - 软件结束模式 (ASTOPEN=0)：此时 I2C_STS 的 TDC 置 1，设置 GENSTOP=1 产生 STOP 条件；
 - 自动结束模式 (ASTOPEN=1)：自动产生 STOP 条件。
- 等待产生 STOP 条件，当 STOP 条件产生时，I2C_STS 的 STOPF 置 1，将 I2C_CLR 的 STOPC 写 1，清除 STOPF 标志，传输结束。

SMBus 主机发送流程

图 11-14 SMBus 主机发送流程图

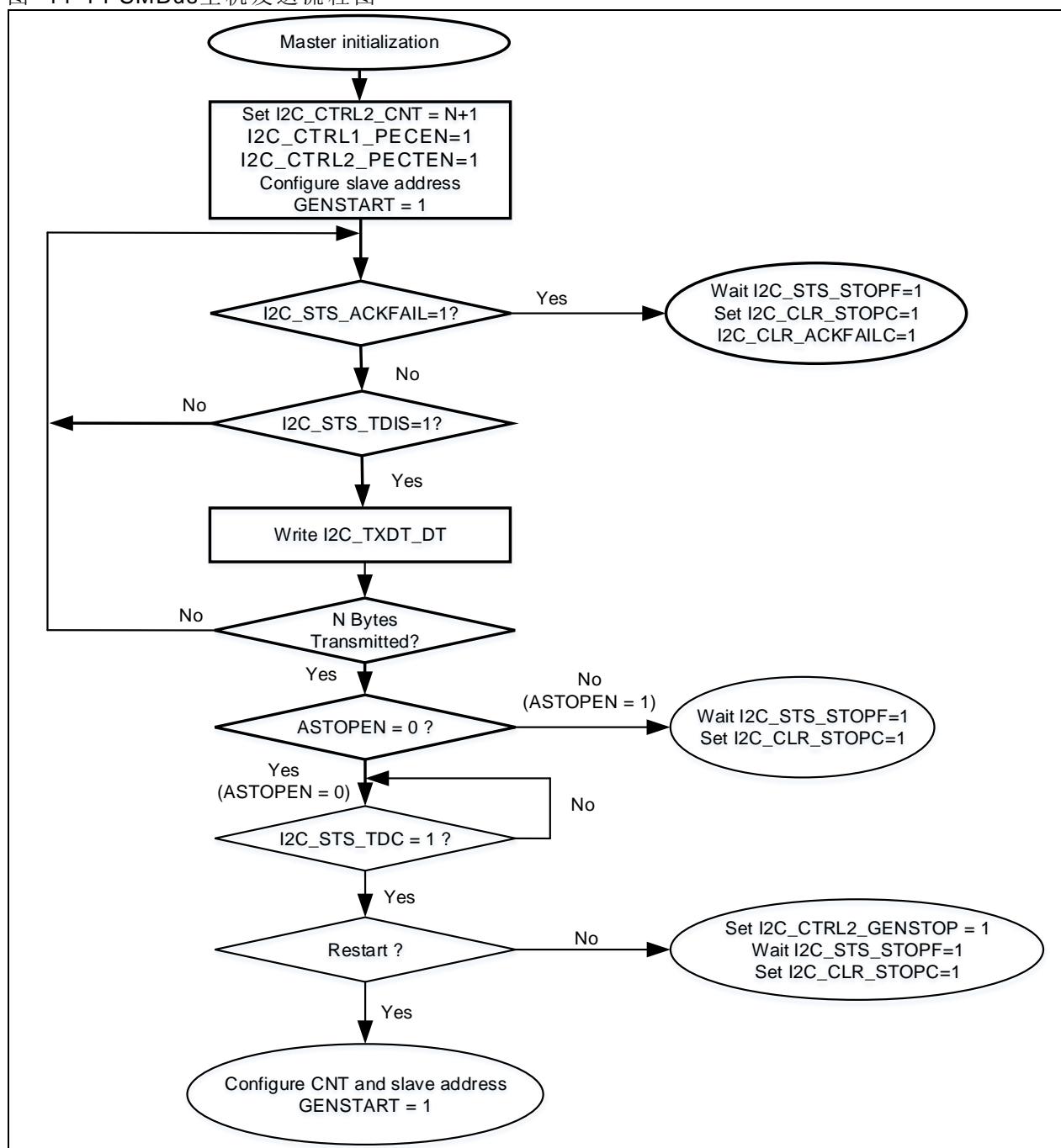


图 11-15 SMBus 主机发送时序图

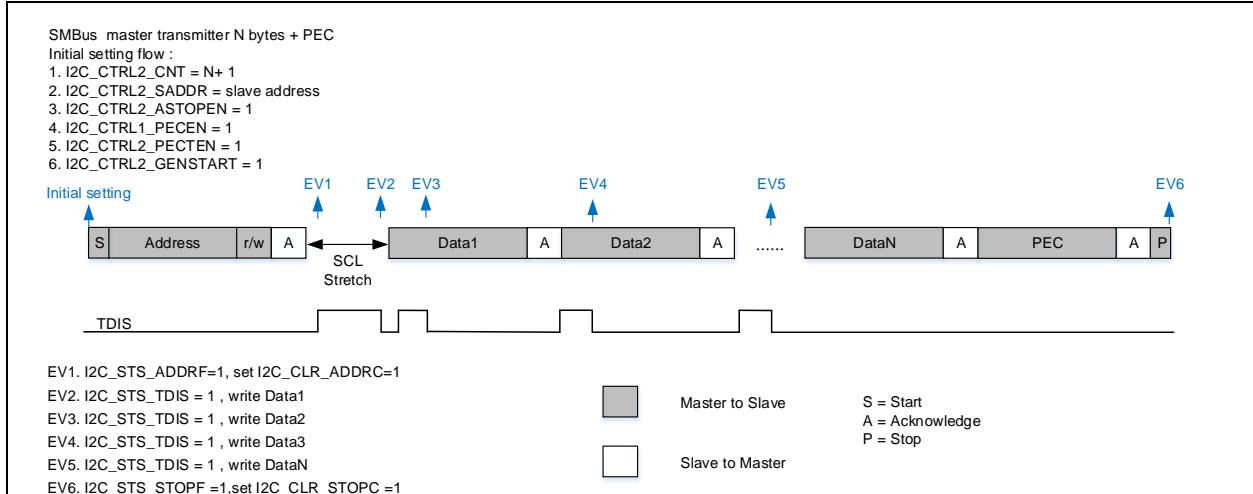
**SMBus 主机接收流程**

图 11-16 SMBus 主机接收流程图

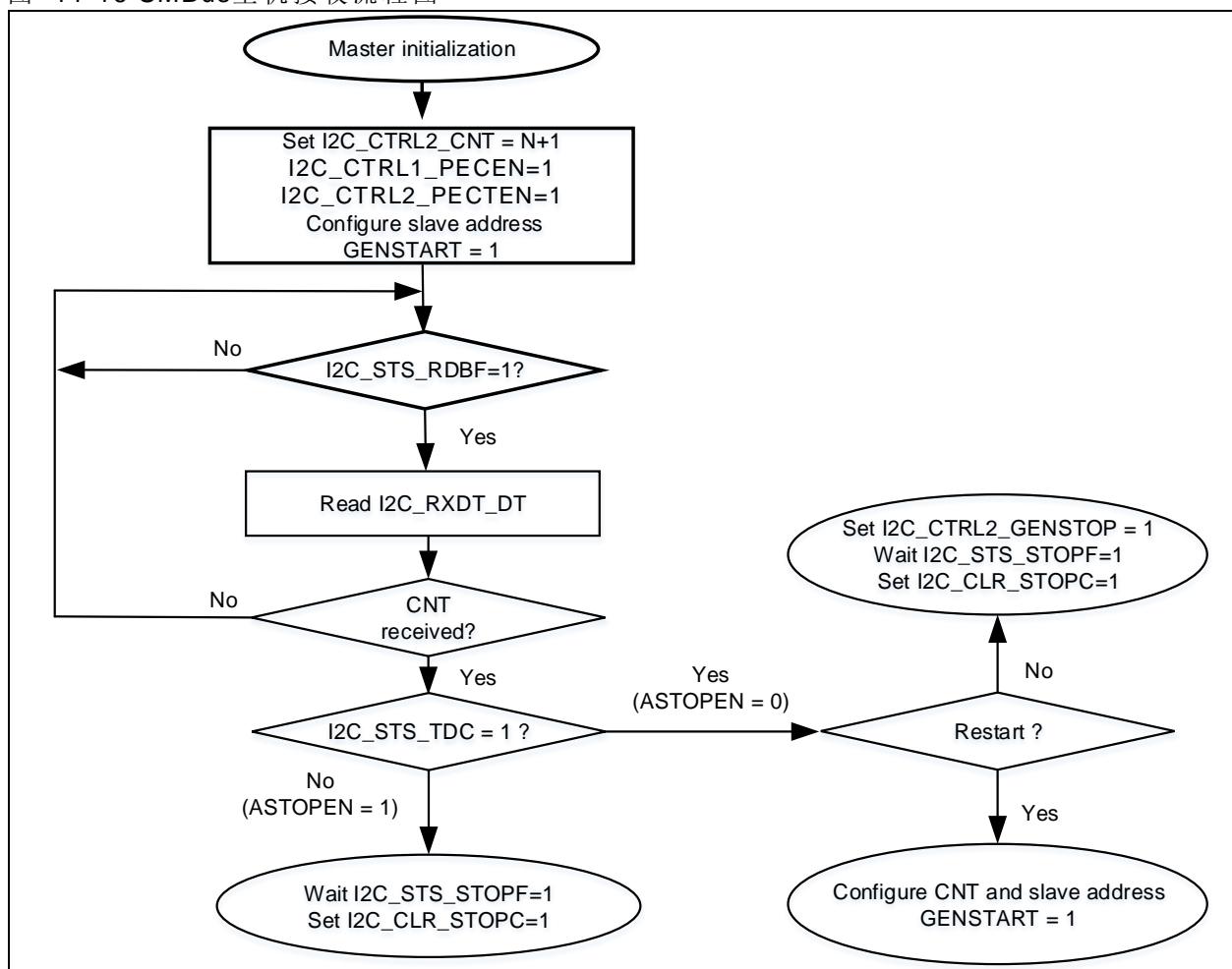
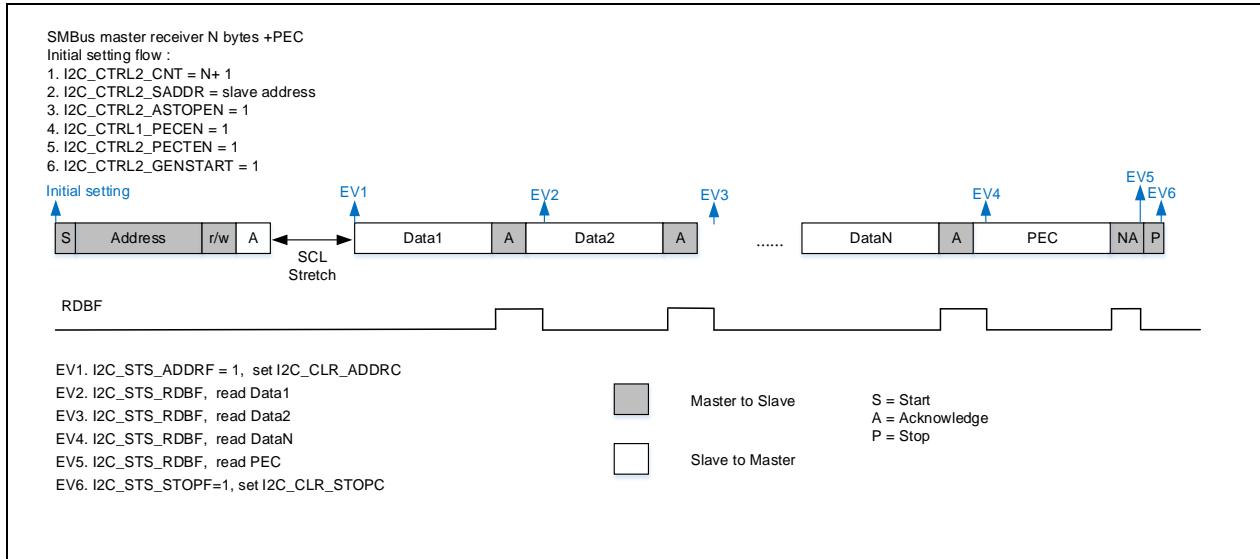


图 11-17 SMBus 主机接收时序图



11.4.7 SMBus 从机通信流程

SMBus 从机通信流程和 I²C 从机通信流程类似。

1. I²C 时钟初始化（配置 I2C_CLKCTRL 寄存器）

- I²C 时钟分频：DIV[7: 0]
- 数据保持时间 (t_{HD;DAT})：SDAD[3: 0]
- 数据建立时间 (t_{SU;DAT})：设置 SCLD[3: 0]

该寄存器的配置可以使用 Artery_I2C_Timing_Configuration 时钟配置工具计算。

2. 设置本机地址

- 设置地址模式为 7 位：设置 I2C_OADDR1 的 ADDR1MODE = 0
- 设置地址 1：设置 I2C_OADDR1 的 ADDR1
- 使能地址 1：设置 I2C_OADDR1 的 ADDR1EN=1

3. SMBus 相关初始化

- 选择为 SMBus 设备：设置 DEVADDREN=1，响应设备默认地址 (0b1100001x)
- 使能 PEC 计算：设置 I2C_CTRL1 的 PECEN=1
- 设置从机字节控制模式：
从机发送：关闭字节控制模式，设置 I2C_CTRL1 的 SCTRL=0；
从机接收：使能字节控制模式，设置 I2C_CTRL1 的 SCTRL=1。

4. 等待地址匹配

当接收到本机地址后，I2C_STS 的 ADDRF 标志置 1，此时可以读取 I2C_STS 的 SDIR 位，得到数据传输方向，当 SDIR=0 时数据传输方向为从机接收数据，当 SDIR=1 时，数据方向为从机发送数据，通过读取 I2C_STS 的 ADDR[6:0] 标志，可以知道接收到的地址是多少。

使能 PEC 传输：设置 I2C_CTRL2 的 PECTEN=1。

设置传输个数：

- 从机发送：设置 I2C_CTRL2 的 CNT=N；
- 从机接收：设置 I2C_CTRL2 的 CNT=1；

重载模式设置：

- 从机发送：设置 I2C_CTRL2 的 RLDEN=0；
- 从机接收：设置 I2C_CTRL2 的 RLDEN=1；

设置 I2C_CTRL 的 CLR=1，清除 ADDRF 标志，开始数据传输。

5. 传输数据（从机发送，开启时钟延展，STRETCH=0）

当在地址匹配后：

1. I2C_TXDT 数据寄存器为空，移位寄存器为空，I2C_STS 的 TDIS=1；
2. 向 TXDT 数据寄存器写入数据 1，然后数据将被立即放进移位寄存器；
3. 此时 TXDT 数据寄存器为空，TDIS 又置 1；
4. 向 TXDT 数据寄存器写入数据 2，TDIS 被清 0；
5. 重复 3、4 步骤直到数据发送 N-1 个数据；
6. 此时从机将自动发送第 N 个数据，也就是 PEC；
7. 等待收到 NACK 条件，当收到 NACK 条件时，I2C_STS 的 ACKFAILF 置 1，将 I2C_CLR 的 ACKFAILC 写 1，清除 ACKFAILF 标志；
8. 等待收到 STOP 条件，当收到 STOP 条件时，I2C_STS 的 STOPF 置 1，将 I2C_CLR 的 STOPC 写 1，清除 STOPF 标志，传输结束。

6. 传输数据（从机接收，开启时钟延展，STRETCH=0）

当在地址匹配后：

1. I2C_RXDT 数据寄存器为空，移位寄存器为空，I2C_STS 的 RDBF=0；
2. 当收到一个字节后，RDBF=1，TCRLD 置 1，从机拉住 SCL 总线；
3. 读取 RXDT 数据寄存器，RDBF 被自动清零；
4. 根据需要配置 I2C_CTRL2 的 NACKEN 位，来产生一个 ACK 或 NACK；
如果产生一个NACK，通信结束；
如果产生一个ACK，通信继续，此时写入CNT=1，硬件自动清除TCRLD标志，从机释放SCL总线，继续接收下一个字节；
5. 重复步骤 2、3、4 直到接收 N-1 个数据；
6. 设置 I2C_CTRL2 的 RLDEN=0，关闭重载模式，设置 CNT=1，重复 2、3 步骤接收 PEC
如果 PEC 校验错误，PECERR 标志将会置 1；
7. 等待收到 STOP 条件，当收到 STOP 条件时，I2C_STS 的 STOPF 置 1，将 I2C_CLR 的 STOPC 写 1，清除 STOPF 标志，传输结束。

SMBus 从机发送

图 11-18 SMBus从机发送流程图

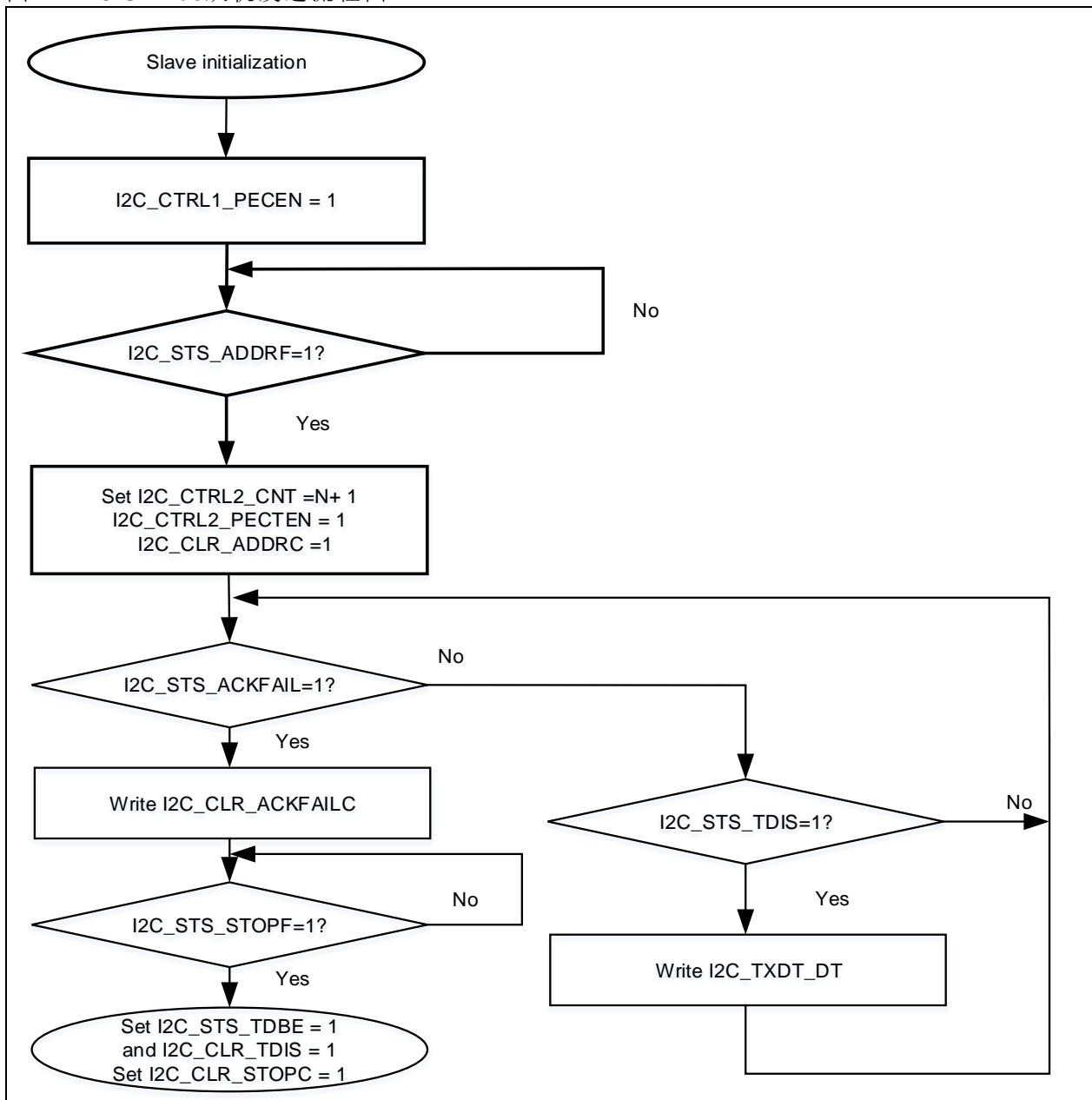


图 11-19 SMBus从机发送时序图

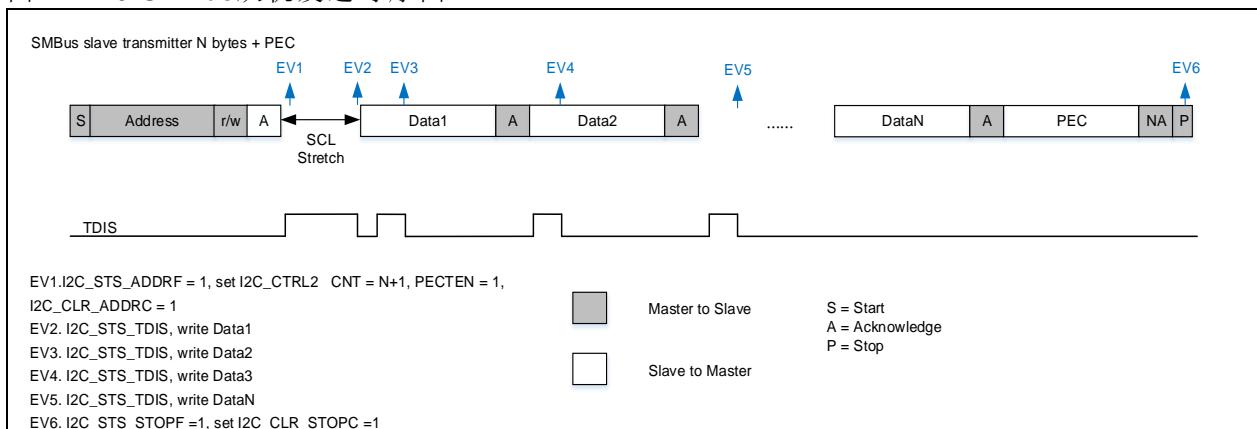
**SMBus 从机接收**

图 11-20 SMBus从机接收流程图

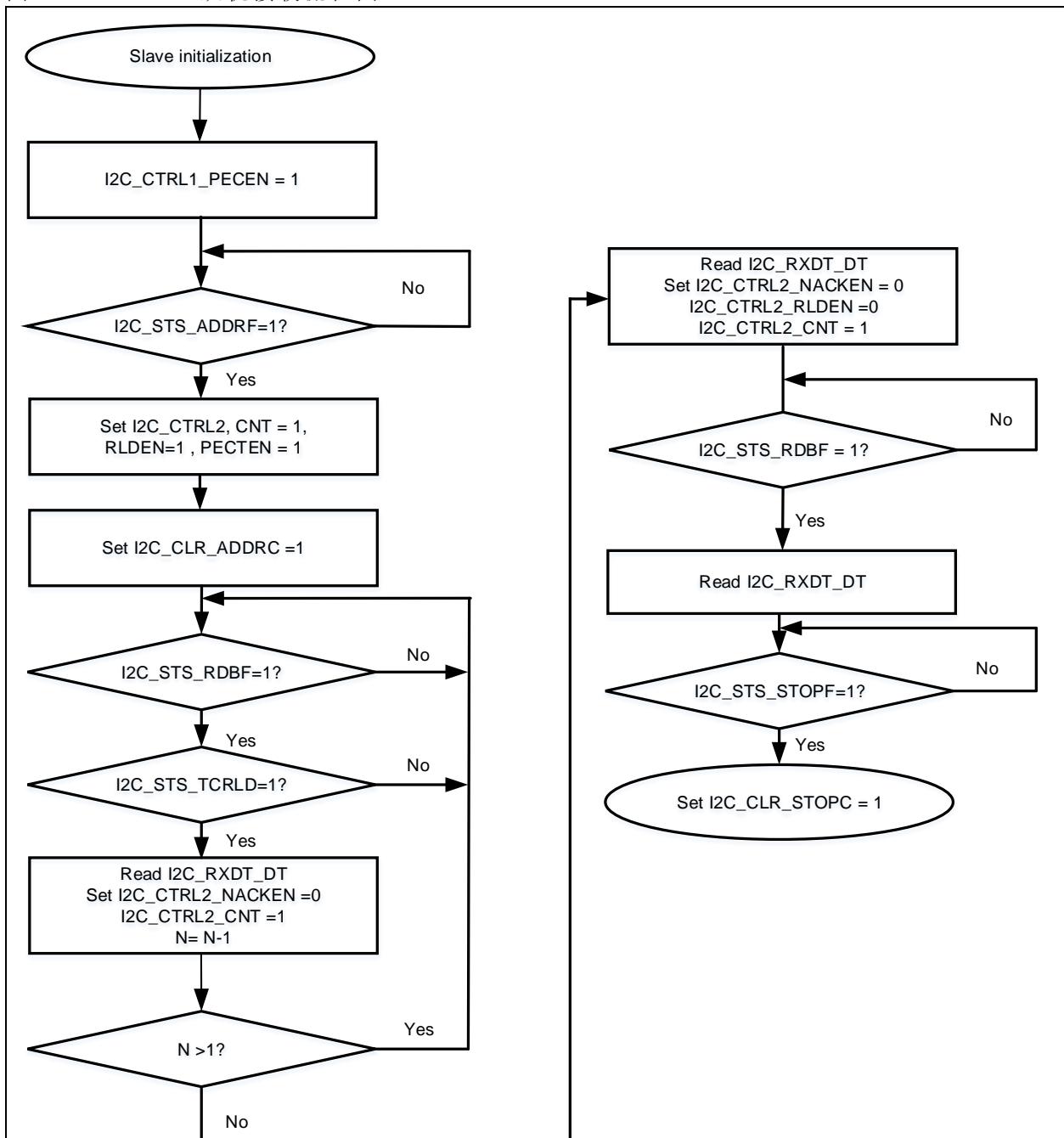
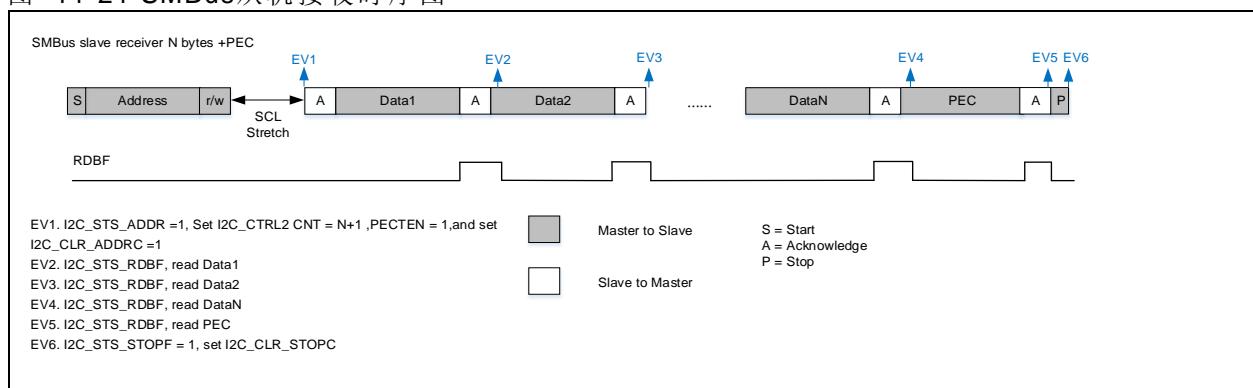


图 11-21 SMBus从机接收时序图



11.4.8 DMA传输

I²C 可以使用 DMA 进行数据传输，降低 CPU 负担，在使用 DMA 传输时，TDIEN 和 RDIEN 应该保持关闭。

DMA 发送 (DMATEN=1)

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_TXDT) 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I²C DMA请求 (控制寄存器2 (I2C_CTRL2) 的DMAEN=1)，当状态寄存器1 (I2C_STS) 的TDIS位被置1时，DMA将数据从内存地址传输到数据寄存器(I2C_TXDT)
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：停止时序见I²C主机通信流程章节。
从机发送模式：停止时序见I²C从机机通信流程章节。

DMA 接收 (DMAREN=1)

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_RXDT) 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I²C DMA请求 (控制寄存器2 (I2C_CTRL2) 的DMAEN=1)，当状态寄存器1 (I2C_STS) 的RDBF位被置1时，DMA将数据从I2C_DT寄存器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：停止时序见I²C主机通信流程章节。
从机接收模式：停止时序见I²C从机机通信流程章节。

11.4.9 错误管理

I²C 内部有多种错误管理，可以极大的提高通信的可靠性，支持错误管理的事件如下：

表 11-6 I²C 错事件

错误事件	事件标志	中断使能位	事件清除位
SMBus 提醒	ALERTF	ERRIEN	ALERTC
超时错误	TMOUT	ERRIEN	TMOUTC
PEC 错误	PECERR	ERRIEN	PECERRC
过载/欠载	OUF	ERRIEN	OUFC
仲裁丢失	ARLOST	ERRIEN	ARLOSTC
总线错误	BUSERR	ERRIEN	BUSERRC

过载或者欠载（OBF）

只有在从机模式下，且关闭时钟延展（I2C_CTRL1 的 STRETCH=1）时，才有可能出现欠载或者过载错误。

从机发送模式：如果在将要传输数据的第一个 bit 位开始发送之前，也就是 SDA 边沿产生之前，如果数据还未写入 TXDT 数据寄存器，那么会发生欠载错误，此时 I2C_STS 的 OBF 将会置 1，并将 0xFF 发送到总线。

从机接收模式：在收到数据后从机应该及时的将数据读走，如果在已经收到 1 个字节后，在下一个字节接收完成之前，如果还未将数据读走，从机将产生过载错误，此时 I2C_STS 的 OBF 将会置 1，并自动回复 NACK。

仲裁丢失（ARLOST）

当设备控制 SDA 线输出高电平，但是总线上实际输出低电平时，发生仲裁丢失事件。

- 主机发送数据：仲裁可以发生在地址传输、数据传输阶段；
- 主机接收数据：仲裁可以发生在地址传输、响应 ACK 阶段；
- 从机发送数据：仲裁可以发生在数据传输阶段；
- 从机接收数据：仲裁可以发生在响应 ACK 阶段。

当在发生仲裁丢失后，硬件自动将 I2C_STS 的 ARLOST 置 1，无论是主机还是从机，都将会立即释放 SCL、SDA 总线，并自动回到从机状态。

总线错误(BUSERR)

在数据传输阶段，在 SCL 高周期区间 SDA 线必须保持稳定，当 SCL 信号为低时，SDA 才能改变，否则将会出现总线错误，当 SCL 为高电平时：

- SDA 从 1 变成 0：错误的开始条件；
- SDA 从 0 变成 1：错误的停止条件；

以上两种情况都会触发总线错误，硬件自动将 I2C_STS 的 BUSERR 置 1。

PEC 错误（PECERR）

只有在 SMBus 模式下才存在 PEC，当在主机接收模式和从机接收模式下，如果接收到的 PEC 和内部计算的 PEC 不相等时，会出现 PEC 错误，此时硬件自动将 I2C_STS 的 PECERR 置 1。

在从机接收模式下，如果 PEC 不正确，从机将回复 NACK。

当在主机接收模式下，无论 PEC 是否正确，主机都将回复 NACK。

SMBus 提醒（ALERTF）

在 SMBus 主机模式下（HADDREN=1）且启用了 SMBus 提醒模式（SMBALERT=1）时，SMBus 提醒功能可以使用，当 ALERT 引脚上产生了提醒事件时（ALERT 引脚电平由高变低），硬件自动将 I2C_STS 的 ALERTF 置 1。

超时错误（TMOUT）

超时错误是 SMBus 协议所定义的，用来提高系统稳定性的机制，用来避免主机或者从机出现故障时一直拉低总线，导致总线无法使用的情况。当发生了超时事件时（SMBus 章节所定义的），硬件自动将 I2C_STS 的 TMOUT 置 1。如果是在从机模式下发生超时事件时，从机将立即释放 SCL 和 SDA 总线；如果是主机发生超时事件时，主机将自动发送一个 STOP 条件结束通信。

11.5 I²C 中断

下表列出了所有的 I²C 中断请求。

表 11-7 I²C 中断请求

中断事件	事件标志	中断使能位
地址匹配	ADDRF	ADDRIEN
应答失败	ACKFAIL	ACKFAILIEN
停止条件产生完成	STOPF	STOPIEN
发送中断状态	TDIS	TDIEN
接收数据缓冲器满	RDBF	RDIEN
传输完成等待加载数据	TCRLD	TDCIEN
数据传输完成	TDC	
SMBus 提醒	ALERTF	ERRIEN
超时错误	TMOUT	
PEC 错误	PECERR	
过载/欠载	OUF	
仲裁丢失	ARLOST	
总线错误	BUSERR	

11.6 I²C 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 超时控制可以继续正常工作或者可以停止。

11.7 I²C 寄存器

下表给出了 I²C 寄存器映像和复位值。

必须以字 (32 位) 的方式操作这些外设寄存器。

表 11-8 寄存器映像和复位值

寄存器简称	基址偏移量	复位值
I2C_CTRL1	0x00	0x00000000
I2C_CTRL2	0x04	0x00000000
I2C_OADDR1	0x08	0x00000000
I2C_OADDR2	0x0C	0x00000000
I2C_CLKCTRL	0x10	0x00000000
I2C_TIMEOUT	0x14	0x00000000
I2C_STS	0x18	0x00000000
I2C_CLR	0x1C	0x00000000
I2C_PEC	0x20	0x00000000
I2C_RXDT	0x24	0x00000000
I2C_TXDT	0x28	0x00000000

11.7.1 控制寄存器1 (I2C_CTRL1)

域	简称	复位值	类型	功能
位 31: 24	保留	0x00	res	保持默认值。
位 23	PECEN	0x0	rw	PEC 计算使能 (PEC calculation enable) 0: 关闭; 1: 开启。
位 22	SMBALERT	0x0	rw	SMBus 提醒功能使能/引脚设置 (SMBus alert enable / pin set) SMBus 主机, 提醒功能使能: 0: 关闭; 1: 开启。 SMBus 从机, 提醒地址使能: 0: 置高; 1: 置低, 响应 0001100x。
位 21	DEVADDREN	0x0	rw	SMBus 设备默认地址使能 (SMBus device default address enable) 0: 关闭; 1: 开启, 响应设备默认地址 1100001x。
位 20	HADDREN	0x0	rw	SMBus 主机地址使能 (SMBus host default address enable) 0: 关闭; 1: 开启, 响应主机地址 0001000x。
位 19	GCAEN	0x0	rw	广播地址使能 (General call address enable) 0: 关闭; 1: 开启, 响应地址 0000000x。
位 18	保留	0x0	res	保持默认值。
位 17	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 16	SCTRL	0x0	rw	从机接收数据控制 (Slave receiving data control) 0: 关闭; 1: 开启。
位 15	DMAREN	0x0	rw	DMA 数据接收使能 (DMA receive data request enable) 0: 关闭; 1: 开启。
位 14	DMATEN	0x0	rw	DMA 数据发送使能 (DMA Transmit data request enable) 0: 关闭; 1: 开启。
位 13: 12	保留	0x0	res	保持默认值。
位 11: 8	DFLT	0x0	rw	数字滤波值 (Digital filter value) 总线上小于此宽度的毛刺将被滤除, 滤波时间 = DFLT x T _{I2C_CLK} 。
位 7	ERRIEN	0x0	rw	错误中断使能 (Error interrupt enable) 0: 关闭; 1: 开启。
位 6	TDCIEN	0x0	rw	数据传输完成中断使能 (Transfer data complete interrupt enable) 0: 关闭; 1: 开启。
位 5	STOPIEN	0x0	rw	停止条件产生完成中断使能 (Stop generation complete interrupt enable) 0: 关闭; 1: 开启。
位 4	ACKFAILIEN	0x0	rw	应答失败中断使能 (Acknowledge fail interrupt enable) 0: 关闭; 1: 开启。

位 3	ADDRIEN	0x0	rw	地址匹配中断使能 (Address match interrupt enable) 0: 关闭; 1: 开启。
位 2	RDIEN	0x0	rw	数据接收中断使能 (Receive data interrupt enable) 0: 关闭; 1: 开启。
位 1	TDIEN	0x0	rw	数据发送中断使能 (Transmit data interrupt enable) 0: 关闭; 1: 开启。
位 0	I2CEN	0x0	rw	I ² C 外设使能 (I ² C peripheral enable) 0: 关闭; 1: 开启。

11.7.2 控制寄存器2 (I2C_CTRL2)

域	简称	复位值	类型	功能
位 31: 27	保留	0x00	res	保持默认值。
位 26	PECTEN	0x0	rw	请求 PEC 传输使能 (Request PEC transmission enable) 0: 停止传输; 1: 启动传输。
位 25	ASTOPEN	0x0	rw	自动发送停止条件使能 (Automatically send stop condition enable) 0: 关闭 (软件结束模式); 1: 开启 (自动结束模式)。
位 24	RLDEN	0x0	rw	发送数据重载模式使能 (Send data reload mode enable) 0: 关闭; 1: 开启。
位 23: 16 CNT[7: 0]		0x00	rw	发送数据个数 (Transmit data counter)
位 15	NACKEN	0x0	rw	不应答使能 (Not acknowledge enable) 0: 应答; 1: 不应答。
位 14	GENSTOP	0x0	rw	产生停止条件 (Generate stop condition) 0: 未产生; 1: 产生。
位 13	GENSTART	0x0	rw	产生起始条件 (Generate start condition) 0: 未产生; 1: 产生。
位 12	READH10	0x0	rw	10 位地址头读取时序使能 (10-bit address header read enable) 0: 关闭; 1: 开启。
位 11	ADDR10	0x0	rw	主机发送 10 位地址模式使能 (Host send 10-bit address mode enable) 0: 7 位地址; 1: 10 位地址。
位 10	DIR	0x0	rw	主机数据传输方向 (Master data transmission direction) 0: 发送; 1: 接收。
位 9: 0	SADDR[9: 0]	0x000	rw	主机发送的从机地址 (The slave address sent by the master) 当在 7 位地址模式下时 BIT0 以及 BIT[9: 8]不关心。

11.7.3 地址寄存器1 (I2C_OADDR1)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0000	res	保持默认值。
位 15	ADDR1EN	0x0	rw	本机地址 1 使能 (Own Address 1 enable) 0: 关闭; 1: 开启。
位 14: 11	保留	0x0	res	保持默认值。
位 10	ADDR1MODE	0x0	rw	本机地址 1 模式 (Own Address mode) 0: 7 位地址; 1: 10 位地址。
位 9: 0	ADDR1[9: 0]	0x000	rw	本机地址 1 (Own address 1) 当在 7 位地址模式下时 BIT0 以及 BIT[9: 8]不关心。

11.7.4 地址寄存器2 (I2C_OADDR2)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0000	res	保持默认值。
位 15	ADDR2EN	0x0	rw	本机地址 2 使能 (Own address 2 enable) 0: 关闭; 1: 开启。
位 14: 11	保留	0x0	res	保持默认值。
位 10: 8	ADDR2MASK[2: 0]	0x0	rw	本机地址 2 位屏蔽 (Own address 2-bit mask) 000: 匹配地址位[7: 1]; 001: 只匹配地址位[7: 2]; 010: 只匹配地址位[7: 3]; 011: 只匹配地址位[7: 4]; 100: 只匹配地址位[7: 5]; 101: 只匹配地址位[7: 6]; 110: 只匹配地址位[7]; 111: 所有非 I ² C 保留地址都会响应。
位 7: 1	ADDR2[7: 1]	0x00	rw	本机地址 2 (Own address 2) 7 位地址。
位 0	保留	0x0	res	保持默认值。

11.7.5 时序寄存器 (I2C_CLKCTRL)

域	简称	复位值	类型	功能
位 31: 28	DIVL[3: 0]	0x0	rw	时钟分频值低 4 位 (Low 4 bits of clock divider value)
位 27: 24	DIVH[7: 4]	0x0	rw	时钟分频值高 4 位 (High 4 bits of clock divider value) DIV = (DIVH << 4) + DIVL。
位 23: 20	SCLD[3: 0]	0x0	rw	SCL 输出延时 (SCL output delay) $T_{SCLD} = (SCLD + 1) \times (DIV + 1) \times T_{I2C_CLK}$ 。
位 19: 16	SDAD[3: 0]	0x0	rw	SDA 输出延时 (SDA output delay) $T_{SDAD} = (SDAD + 1) \times (DIV + 1) \times T_{I2C_CLK}$ 。
位 15: 8	SCLH[7: 0]	0x00	rw	SCL 高电平 (SCL high level) $T_{SCLH} = (SCLH + 1) \times (DIV + 1) \times T_{I2C_CLK}$ 。
位 7: 0	SCLL[7: 0]	0x00	rw	SCL 低电平 (SCL low level) $T_{SCLL} = (SCLL + 1) \times (DIV + 1) \times T_{I2C_CLK}$ 。

11.7.6 超时寄存器 (I2C_TIMEOUT)

域	简称	复位值	类型	功能
位 31	EXTEN	0x0	rw	累计时钟延展超时使能 (Cumulative clock low extend timeout enable) 0: 关闭; 1: 开启。 SMBus 协议里面的 $T_{LOW:SEXT} / T_{LOW:MEXT}$ 。
位 30: 28	保留	0x0	res	保持默认值。
位 27: 16	EXTTIME[11:0]	0x000	rw	累计时钟延展超时时间 (Cumulative clock low extend timeout value) 超时时间 = $(EXTTIME + 1) \times 2048 \times T_{I2C_CLK}$ 。
位 15	TOEN	0x0	rw	时钟电平超时检测使能 (Detect clock low/high timeout enable) 0: 关闭; 1: 开启。 SMBus 协议里面的 $T_{TIMEOUT}$ 。
位 14: 13	保留	0x0	res	保持默认值。
位 12	TOMODE	0x0	rw	时钟电平超时检测模式 (Clock timeout detection mode) 0: 检测低电平; 1: 检测高电平。
位 11: 0	TOTIME[11:0]	0x000	rw	时钟电平超时检测时间 (Clock timeout detection time) 当检测低电平 (TOMODE = 0) 时: 超时时间 = $(TOTIME + 1) \times 2048 \times T_{I2C_CLK}$ 。 当检测高电平 (TOMODE = 1) 时: 超时时间 = $(TOTIME + 1) \times 4 \times T_{I2C_CLK}$ 。

11.7.7 状态寄存器 (I2C_STS)

域	简称	复位值	类型	功能
位 31: 24	保留	0x00	res	保持默认值。
位 23: 17	ADDR[6: 0]	0x00	r	从机地址匹配值 (Slave address matching value) 7 位地址下: 从机接收到的地址。 10 位地址下: 从机接收到的 10 位地址头。
位 16	SDIR	0x0	r	从机数据传输方向 (Slave data transmit direction) 0: 接收数据; 1: 发送数据。
位 15	BUSYF	0x0	r	总线忙标志 (Bus busy flag transmission mode) 0: 空闲; 1: 忙。 当检测到 START 条件置起, 检测到停止条件自动清零。
位 14	保留	0x00	res	保持默认值。
位 13	ALERTF	0x0	r	SMBus 提醒标志 (SMBus alert flag) SMBus 主机: 指示提醒信号接收 (ALERT 引脚由高变低) 0: 未收到; 1: 收到。
位 12	TMOUT	0x0	r	SMBus 超时标志 (SMBus timeout flag) 0: 未超时; 1: 超时。
位 11	PECERR	0x0	r	PEC 接收错误标志 (PEC receive error flag) 0: 正确; 1: 错误。
位 10	OUF	0x0	r	过载或者欠载标志 (Overflow or underflow 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。

位 9	ARLOST	0x0	r	仲裁丢失标志 (Arbitration lost flag) 0: 正常; 1: 仲裁丢失。
位 8	BUSERR	0x0	r	总线错误标志 (Bus error flag) 0: 正常; 1: 错误。
位 7	TCRLD	0x0	r	传输完成, 等待加载数据 (Transmission is complete, waiting to load data) 0: 未完成; 1: 已完成。 当数据被发送完成 (CNT = 1), 并且使能了重载模式 (RLDEN=1) 时被置起, 当写入 CNT 值时自动清零。在主机模式或者当从机在 SCTRL=1 时使用。
位 6	TDC	0x0	r	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, 数据已经完全发送到总线上)。 当在软件结束模式, 并且数据传输完成后置起 (ASTOPEN = 0, RLDEN = 0, CNT = 0)。 收到 START 或者 STOP 条件后自动清零。
位 5	STOPF	0x0	r	停止条件产生完成标志 (Stop condition generation complete flag) 0: 未产生; 1: 已产生。
位 4	ACKFAILF	0x0	r	应答失败标志 (Acknowledge failure flag) 0: 正常; 1: 失败。
位 3	ADDRF	0x0	r	地址匹配标志 (0~7 bit address match flag) 0: 未匹配; 1: 已匹配。
位 2	RDBF	0x0	r	接收数据缓冲器满 (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。
位 1	TDIS	0x0	rw1s	发送中断状态 (Transmit data interrupt status) 0: 数据已写入 I2C_TXDT; 1: 数据已从 I2C_TXDT 发送到移位寄存器, I2C_TXDT 为空, 这时必须把要发的数据写到 I2C_TXDT 寄存器。在禁止时钟延展模式下, 可以向此位写 1, 以生成一个 TDIS 事件以便提前写数据到 I2C_TXDT 寄存器。
位 0	TDBE	0x0	rw1s	发送数据寄存器空标志 (Transmit data buffer empty flag) 0: I2C_TXDT 有数据, 不为空; 1: I2C_TXDT 无数据, 为空。 该位只用以表示当前 I2C_TXDT 的状态, 并可以通过软件写 1, 清除 I2C_TXDT 寄存器里的数据。

11.7.8 状态清除寄存器 (I2C_CLR)

域	简称	复位值	类型	功能
位 31: 14	保留	0x00000	res	保持默认值。
位 13	ALERTC	0x0	w	清除 SMBus 提醒标志 (Clear SMBus alert flag) 写 1 清除。
位 12	TMOUTC	0x0	w	清除 SMBus 超时标志 (Clear SMBus timeout flag) 写 1 清除。
位 11	PECERRC	0x0	w	清除 PEC 接收错误标志 (Clear PEC receive error flag) 写 1 清除。
位 10	OUFC	0x0	w	清除溢出标志 (Clear overload / underload flag) 写 1 清除。
位 9	ARLOSTC	0x0	w	清除仲裁丢失标志 (Clear arbitration lost flag) 写 1 清除。
位 8	BUSERRC	0x0	w	清除总线错误标志 (Clear bus error flag) 写 1 清除。
位 7: 6	保留	0x0	res	保持默认值。
位 5	STOPC	0x0	w	清除停止条件产生完成标志 (Clear stop condition generation complete flag) 写 1 清除。
位 4	ACKFAILC	0x0	w	清除应答失败标志 (Clear acknowledge failure flag) 写 1 清除。
位 3	ADDRC	0x0	w	清除地址匹配标志 (Clear 0~7 bit address match flag) 写 1 清除。
位 2: 0	保留	0x0	res	保持默认值。

11.7.9 PEC寄存器 (I2C_PEC)

域	简称	复位值	类型	功能
位 31: 8	保留	0x000000	res	保持默认值。
位 7: 0	PECVL[7: 0]	0x00	r	PEC 值 (PEC value)

11.7.10接收寄存器 (I2C_RXDT)

域	简称	复位值	类型	功能
位 31: 8	保留	0x000000	res	保持默认值。
位 7: 0	DT[7: 0]	0x00	r	接收数据寄存器 (Receive data register)

11.7.11发送寄存器 (I2C_TXDT)

域	简称	复位值	类型	功能
位 31: 8	保留	0x000000	res	保持默认值。
位 7: 0	DT[7: 0]	0x00	rw	发送数据寄存器 (Transmit data register)

12 通用同步异步收发器 (USART)

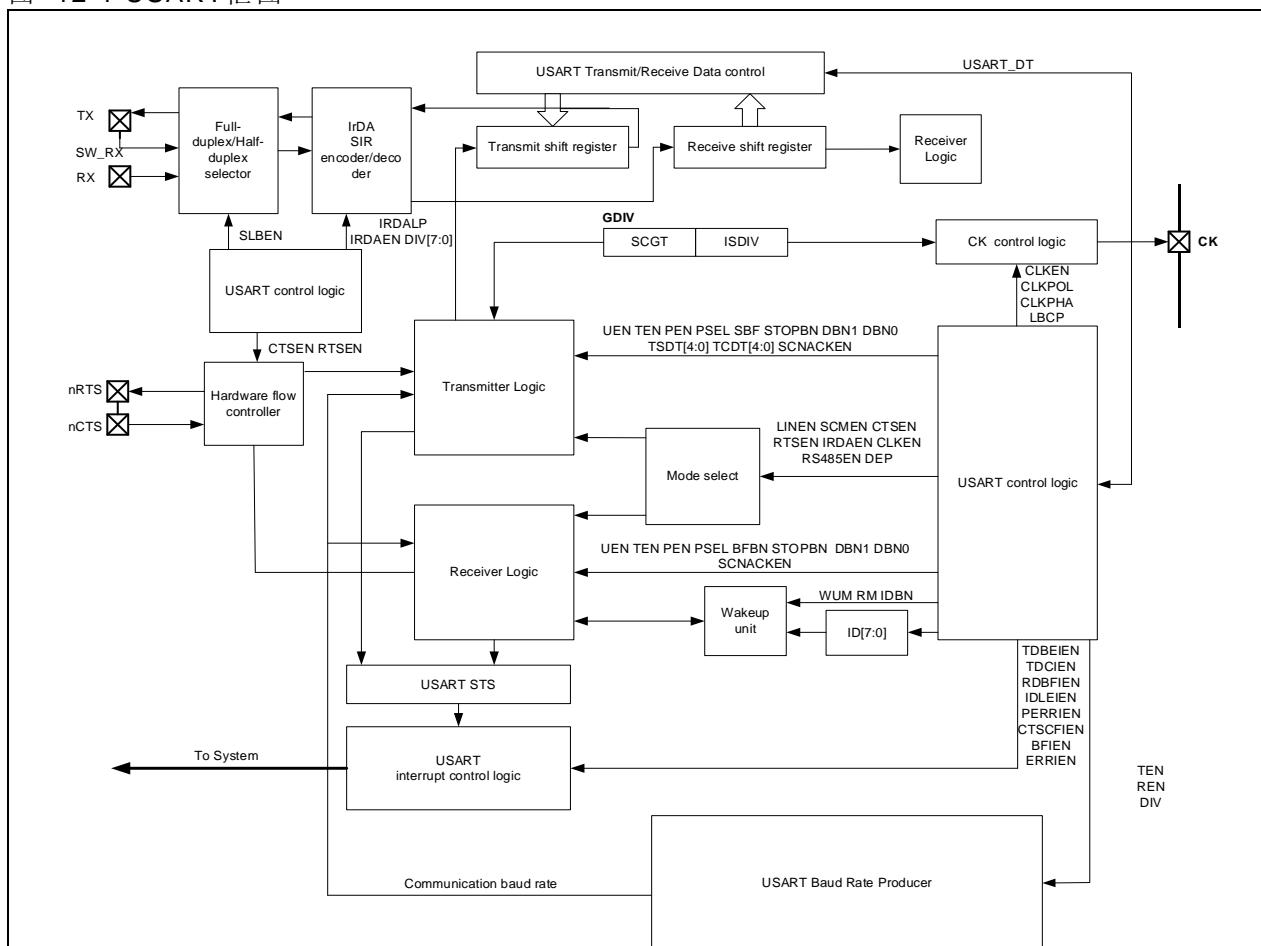
12.1 USART介绍

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，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，RS485 和 Modbus。

USART 还支持多处理器通信，以及可配置通过空闲帧或地址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
 - RS-485
 - 通过静默模式实现多处理器通信 (具有地址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
- 可编程配置的帧格式
 - 可编程的数据位位数（7位或8位或9位）
 - 可编程的停止位位数-支持0.5或1或1.5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
 - 可编程配置的数据发送顺序（MSB/LSB）
 - 可编程配置的Tx/Rx引脚极性
 - 可编程配置的DT数据极性
- 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
 - 传输结束标志
- 四个错误检测标志
 - 溢出错误
 - 噪音错误
 - 帧错误
 - 校验错误
- 可编程配置的12个带标志的中断源
 - CTS电平改变
 - LIN断开符检测
 - 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
 - 噪音错误
 - 校验错误
 - 接收器超时检测
 - 字节匹配检测

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引脚用于数据输出，RX引脚用于数据输入，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SLBEN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引脚无效，TX引脚和SW_RX引脚互连，对USART来说，TX引脚用于数据输出，SW_RX用于数据输入，对外设来说，数据都从TX引脚映射的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

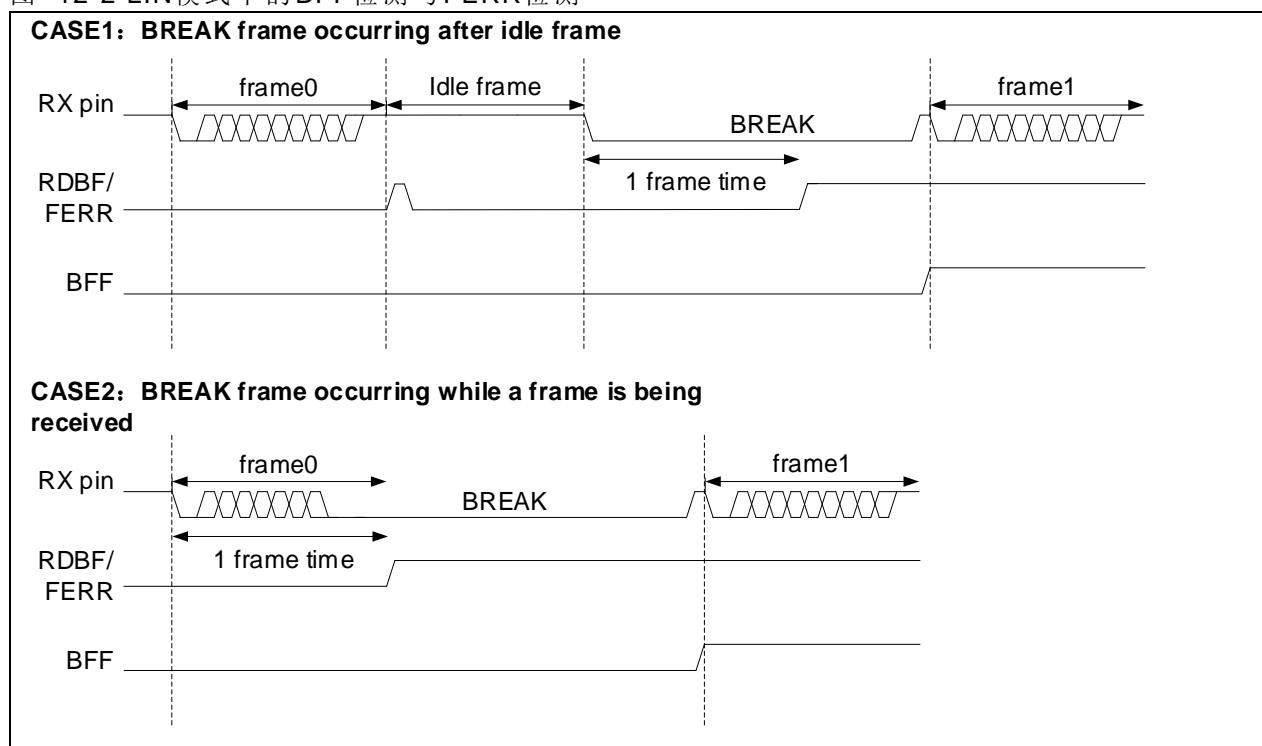
用户可以通过不同的配置以此选择不同的工作模式，配置方法分列如下所列，请将如下配置方法配合本章后述的接收器和发送器配置方法结合使用以完成 USART 初始化配置。

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，SLBEN位置0，IRDAEN位置0，DBN[1: 0]=00。

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，SLBEN位置0，IRDAEN位置0，CLKEN位置1，DBN[1: 0]=01，PEN位置1，STOPBN[1: 0]=11。

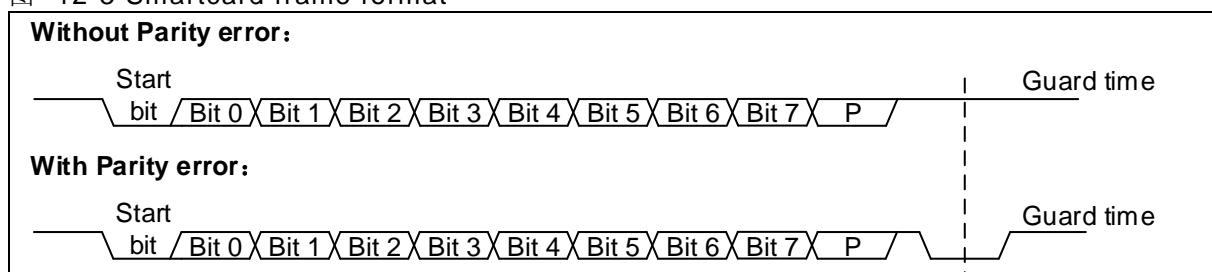
可以选择配置CLKPOL位和CLKPH位以及LBCP位以满足不同的时钟极性以及时钟相位和时钟脉冲个数，具体可见同步模式部分。

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。

注意：发送和接收时，都只可配置为1.5个停止位，且必须确保智能卡保护时间皆满足1.5位长才能正常接收连续数据。

图 12-3 Smartcard frame format

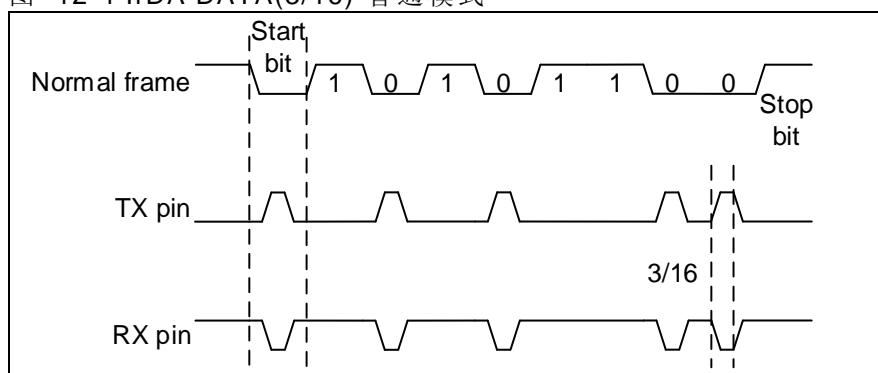


3. 红外模式

基础设置：IRDAEN位置1，CLKEN位置0，STOPBN[1: 0]位置0，SCMEN位置0，SLBEN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. Modbus

USART仅提供实现Modbus/RTU和Modbus/ASCII所需的基础的硬件支持，这意味着协议的控制部分必须由软件完成，USART只是提供块结束侦测。

在Modbus/RTU中，块结束侦测通过可编程的超时功能识别接收线为空闲状态的时间大于2个字节时间实现，用户可以通过配置RTOV寄存器设定需要的超时值（时间单位是1bit位宽），通过RTODEN位置1开启超时检测，当USART接收器检测到接收线为空闲的时间等于设定的超时值时，USART会置起RTODF，如果RTODIE位置1，产生中断，可以通过RTODCF位写1清除RTODF位；

在Modbus/ASCII中，块结束的侦测通过字节匹配功能识别特殊的字节序列（CR/LF），通过软件编程将LF ASCII码写入ID[7: 0]，通过CMDIE位置1开启字节匹配功能，当USART接收器接收到的数据和ID[7: 0]匹配时，USART会置起CMDF，如果CMDIE位置1，产生中断，可以通过CMDCF位写1清除CMDF位。

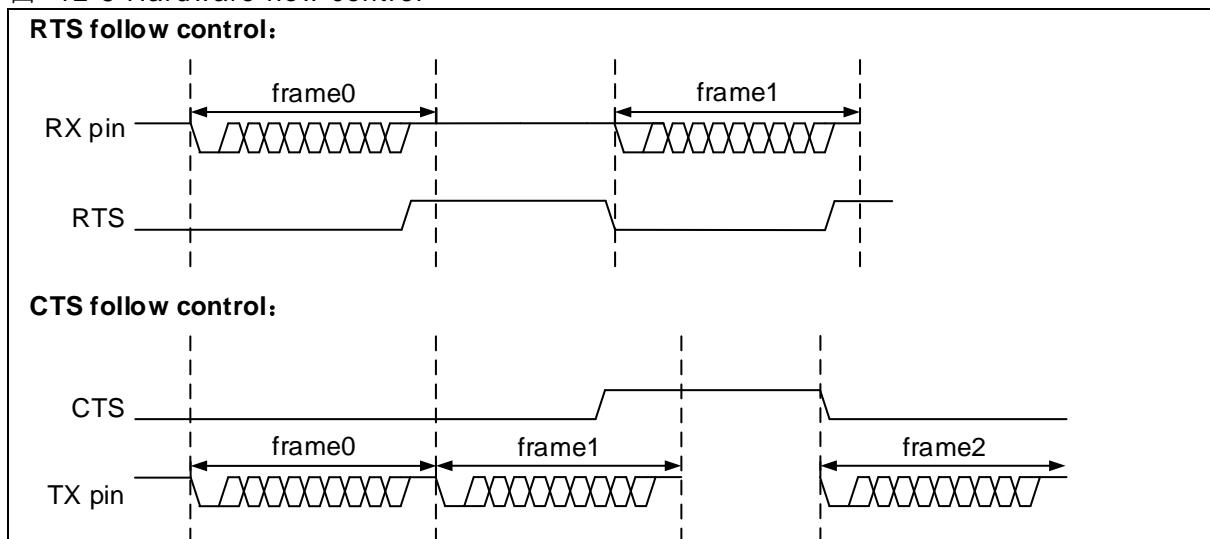
5. 硬件流控制模式

RTSEN位置1和CTSEN位置1可以分别开启RTS和CTS流控制。

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



6. RS485模式

RS485EN位置1开启RS485模式，驱动使能信号从RTS引脚输出，用户可以通过配置DEP位选择DE信号的极性，用户可以通过分别配置TSDT[4: 0]和TCDT[4: 0]设置发送器开始发送起始位前延迟的时间，和发送器发送完最后一笔数据的停止位后置起TC标志前延迟的时间。

7. 静默模式

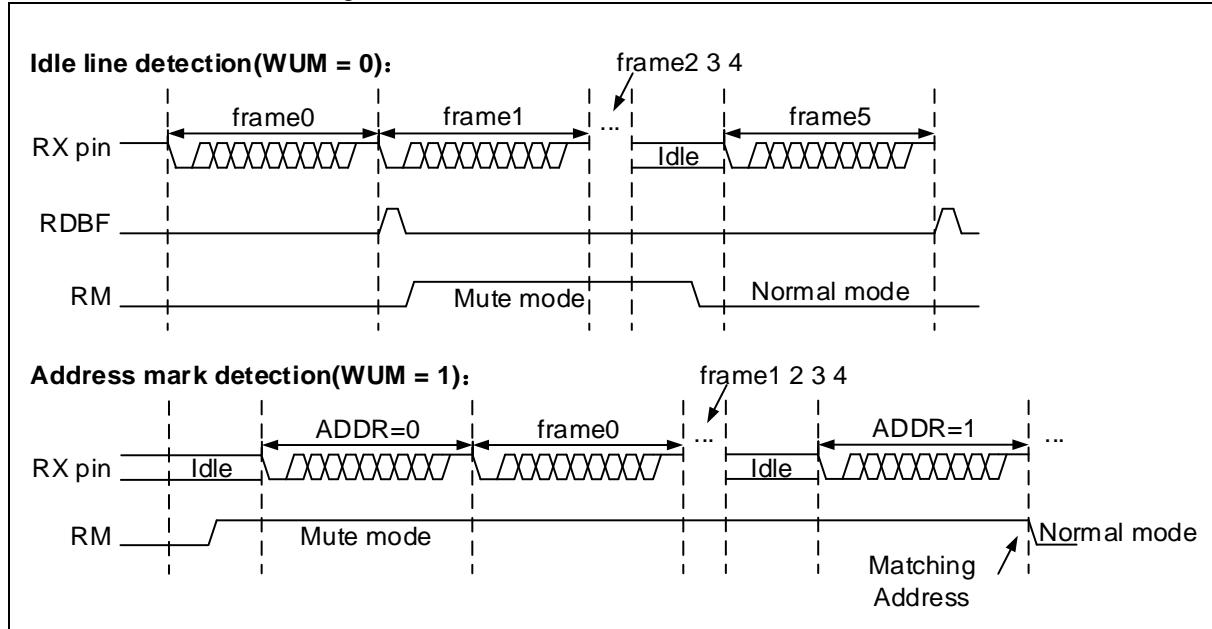
RM位置1进入静默模式，根据WUM位置1和置0，可以分别通过ID匹配和空闲总线从静默模式中唤醒，其中ID号ID[7: 0]可编程配置，并且可以通过配置IDBN选择使用ID[7: :0]或ID[3: 0]，当选择ID匹配时，数据位的MSB为1表示当前数据是ID。

关闭奇偶校验功能时，当DBN[1: 0]=10时，MSB是USART_DT[6]，当DBN[1: 0]=00时，MSB是USART_DT[7]，当DBN[1: 0]=01时，MSB是USART_DT[8]。

开启奇偶校验功能时，当DBN[1: 0]=10时，MSB是USART_DT[5]，当DBN[1: 0]=00时，MSB是USART_DT[6]，当DBN[1: 0]=01时，MSB是USART_DT[7]。

当选择ID[3: 0]时，数据位的4个LSB表示ID值；当选择ID[7: 0]时，数据位除上述奇偶校验位以外，所有的LSB位表示ID值。

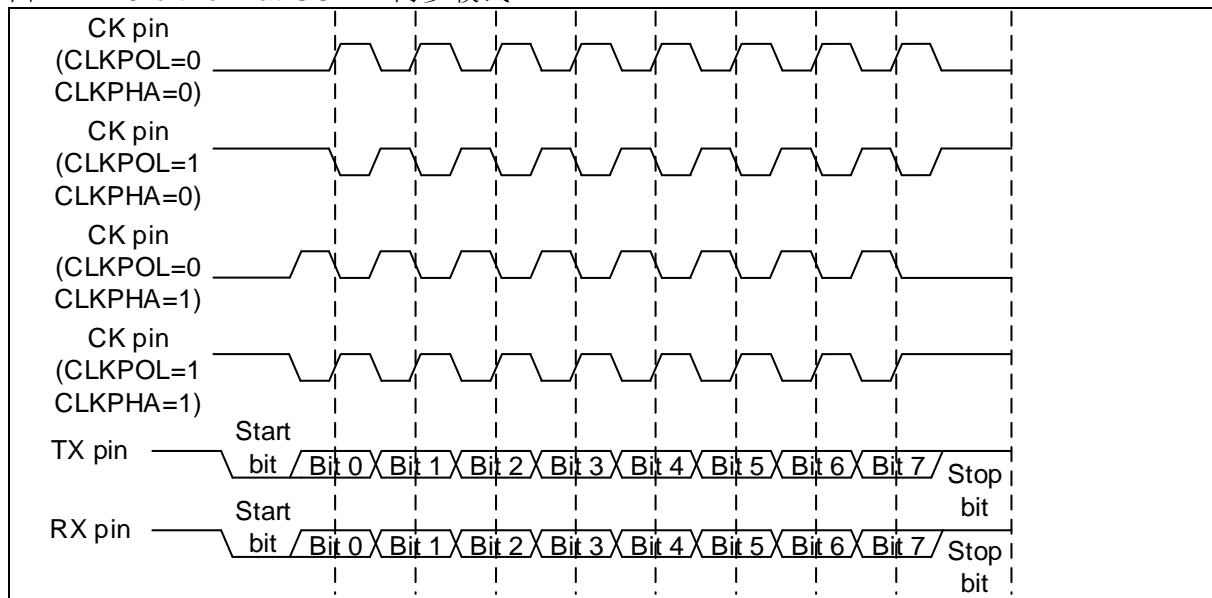
图 12-6 Mute mode using Idle line or Address mark detection



8. 同步模式

CLKEN位置1开启同步模式并使能时钟引脚输出，通过配置CLKPOL位置1或0可以选择空闲状态下CK引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



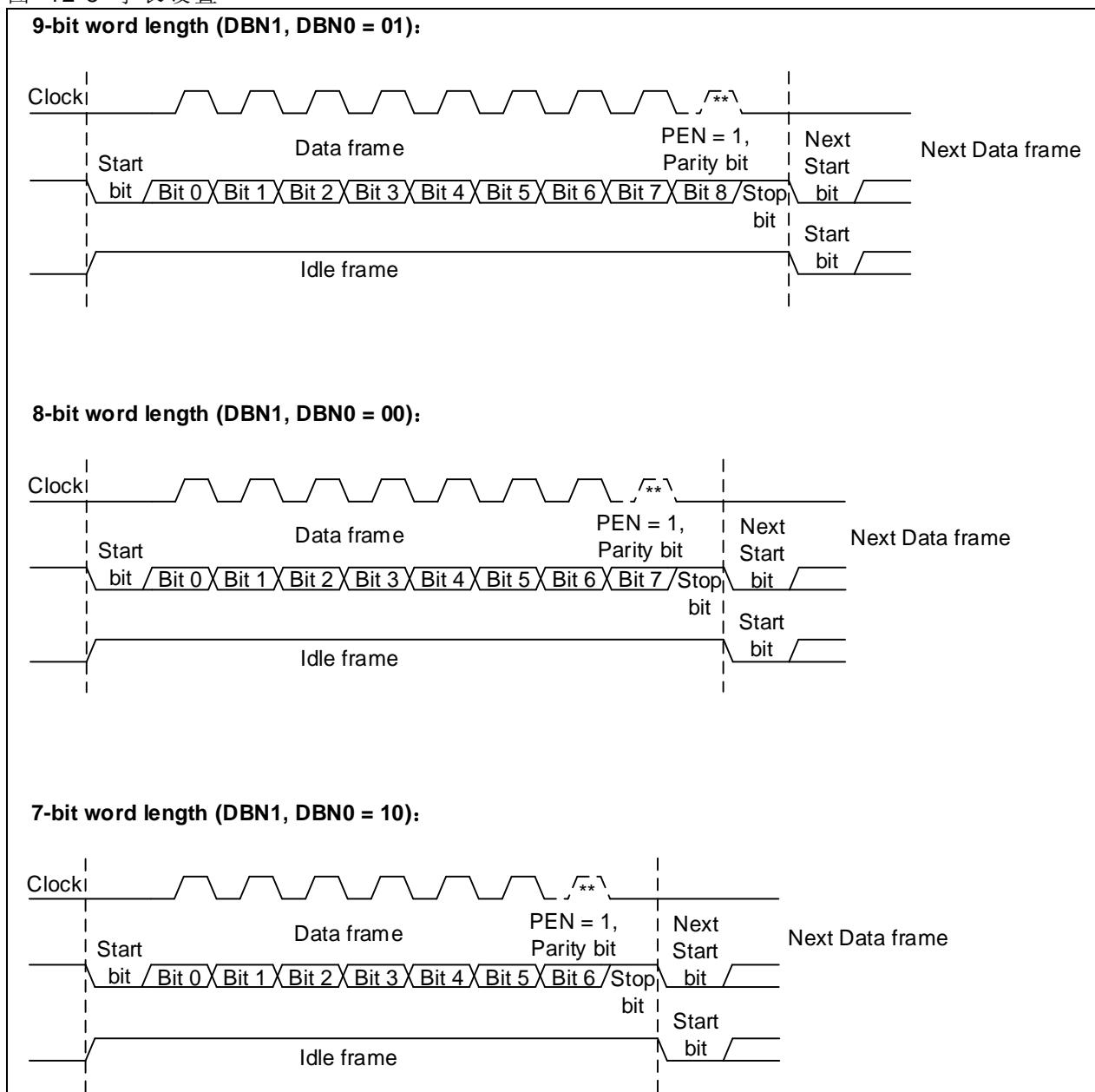
12.4 USART帧格式简述和配置流程

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一笔断开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。需要特别注意的是，在非LIN模式下，发送和检测断开帧的长度都需遵守此规则，例DBN[1:0]=00，那么发送和检测的断开帧长度就是10位低电平加停止位。LIN模式请参考模式选择器简述和配置流程部分。通过DBN1位和DBN0配置7位(DBN[1:0]=10)8位(DBN[1:0]=00)或9位(DBN[1:0]=01)数据位。

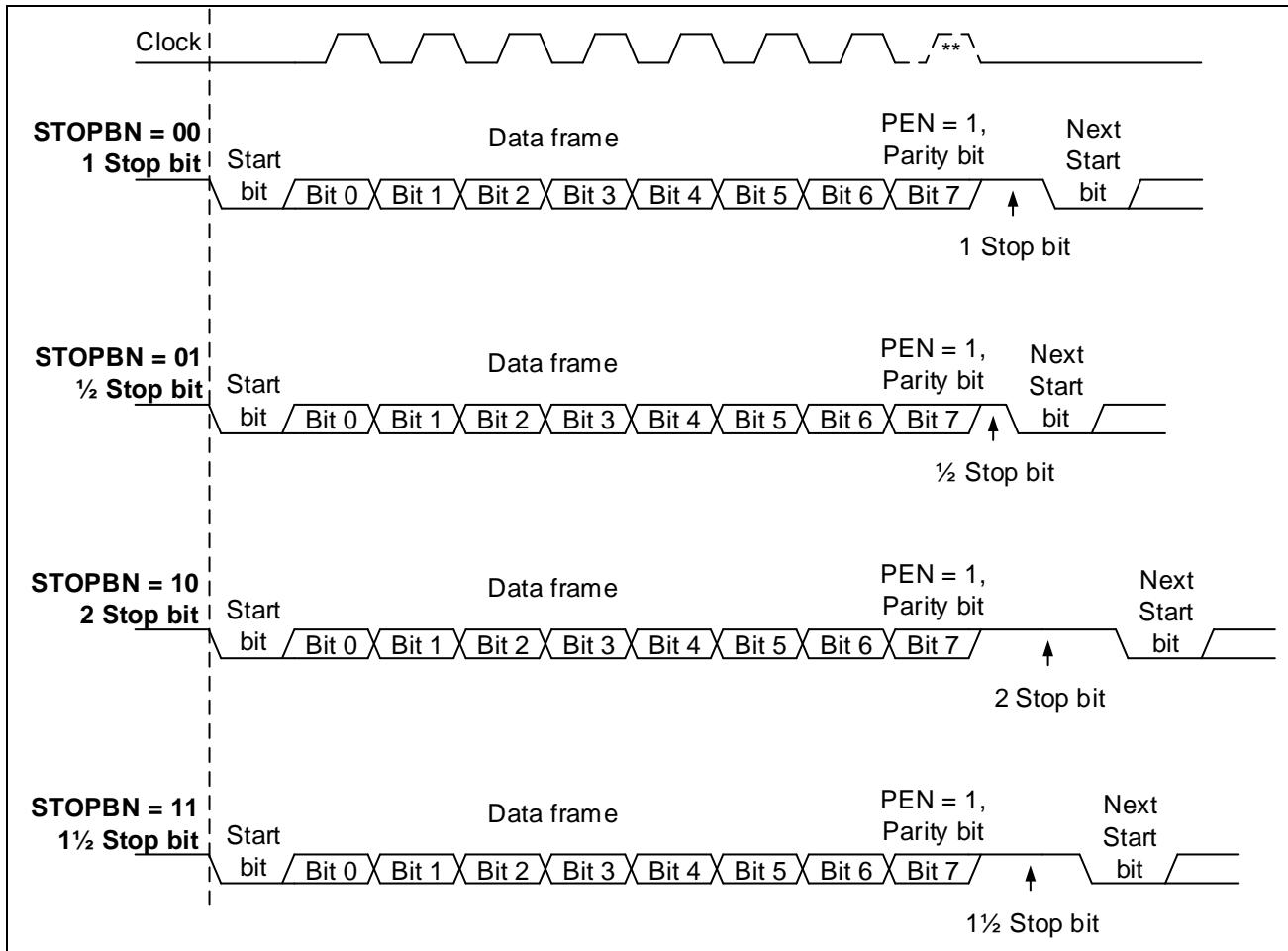
图 12-8 字长设置



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)或偶校验(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



通过 MTF 位配置数据是先传输 MSB (MTF=1) 还是 LSB (MTF=0)。

通过 DTREV 位配置 USART_DT 是以 1=L,0=H (DTREV=1) 还是 0=L,1=H (DTREV=0) 的方式发送和接收。

通过 TXREV 位配置 USART_TX 引脚上的信号是以 VDD=0/mark,Gnd=1/idle (TXREV=1) 还是 VDD=1/idle,Gnd=0/mark (TXREV=0) 的方式传输。

通过 RXREV 位配置 USART_RX 引脚上的信号是以 VDD=0/mark,Gnd=1/idle (RXREV=1) 还是 VDD=1/idle,Gnd=0/mark (RXREV=0) 的方式传输。

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 USART_DT 寄存器地址，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 USART_DT 寄存器中。
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

- 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。
- 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USART_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USART的USART_DT寄存器的地址，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
- 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。
- 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。

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 (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 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（PCLK1 用于 USART2, PCLK2 用于 USART1）

注意：1. 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		fPCLK=36MHz			fPCLK=72MHz		
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 TDBEIEN 将会产生中断。

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 引脚输入。

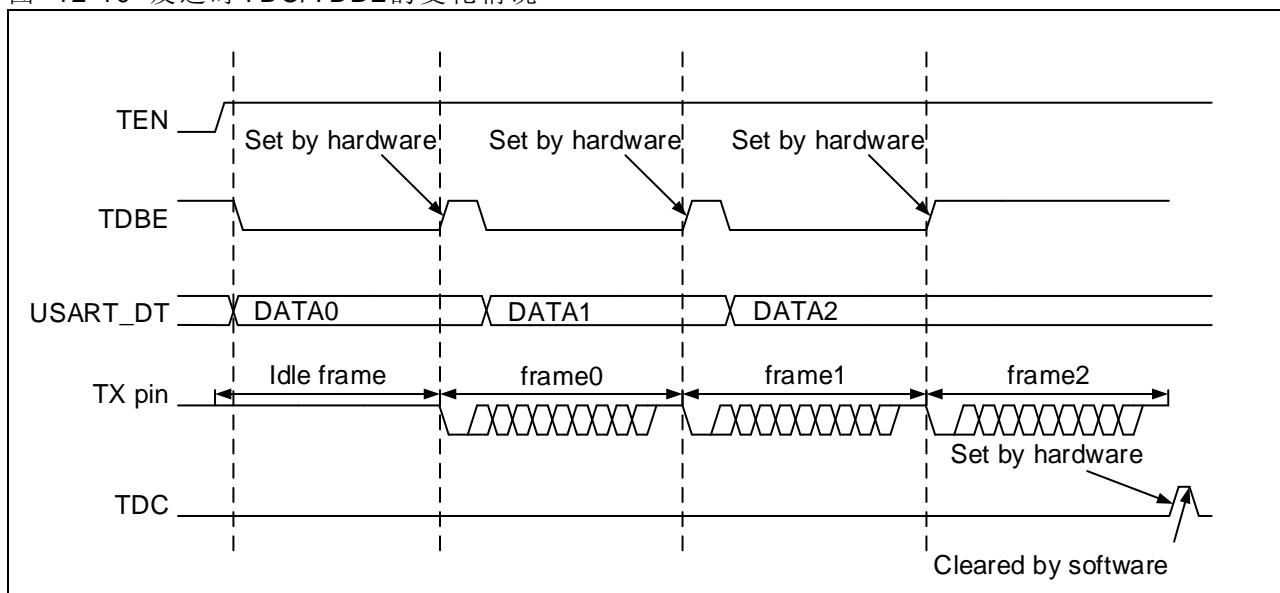
注意：1. 在数据传输期间不能复位 TEN 位，否则将破坏 TX 脚上的数据。

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)。
3. 模式配置: 具体参见模式选择器配置部分 (12.3)。
4. 帧格式配置: 具体参见帧格式配置部分 (12.4)。
5. 中断配置: 具体参见中断发生器配置部分 (12.10)。
6. DMA发送配置: 如果选择使用DMA发送，DMATEN位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)。
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USART_DT寄存器(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USART_STS寄存器，再写一次USART_DT寄存器的方式来清除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(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 引脚输出。

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)。
3. 模式配置: 具体参见模式选择器配置部分 (12.3)。
4. 帧格式配置: 具体参见帧格式配置部分 (12.4)。
5. 中断配置: 具体参见中断发生器配置部分 (12.10)。
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)。
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, 噪音或溢出错误, 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 USART_DT 寄存器完成对 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。

当一个断开帧被接收到时:

- 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 接收器按断开帧处理, 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 USART_DT 寄存器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 USART_STS, 再读 USART_DT 寄存器, 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 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音，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 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[1: 0]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

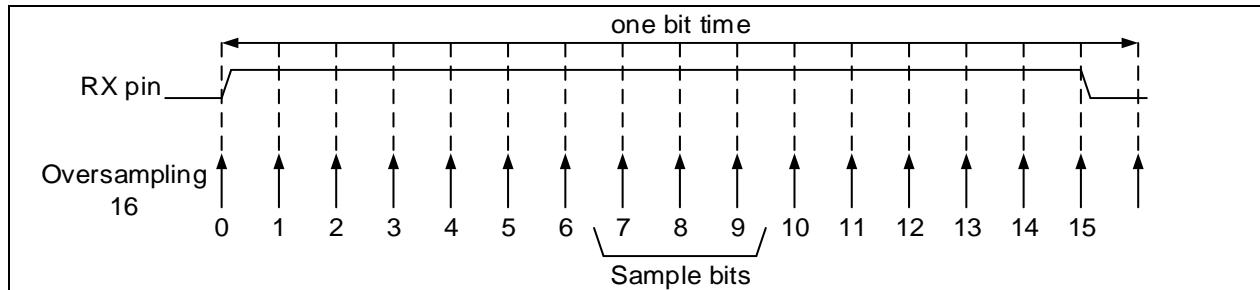
DBN[1:0]	DIV[3:0] = 0	DIV[3:0] != 0
00	3.75%	3.33%
01	3.41%	3.03%
10	4.16%	3.7%

当 USART 接收器在数据帧中检测到噪音时：

- 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 USART_STS，再读 USART_DT 寄存器，将清除 NERR 位。

图 12-11 检测噪声的数据采样

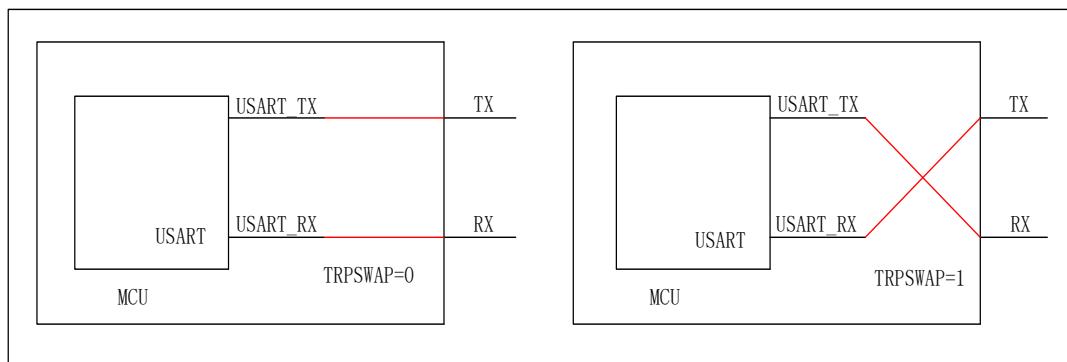


12.9 Tx/Rx可配置引脚互换

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：TRPSWAP (USART_CTRL2[15]) 位必须在 USART 未被使能 (UEN=0) 时才能被改写

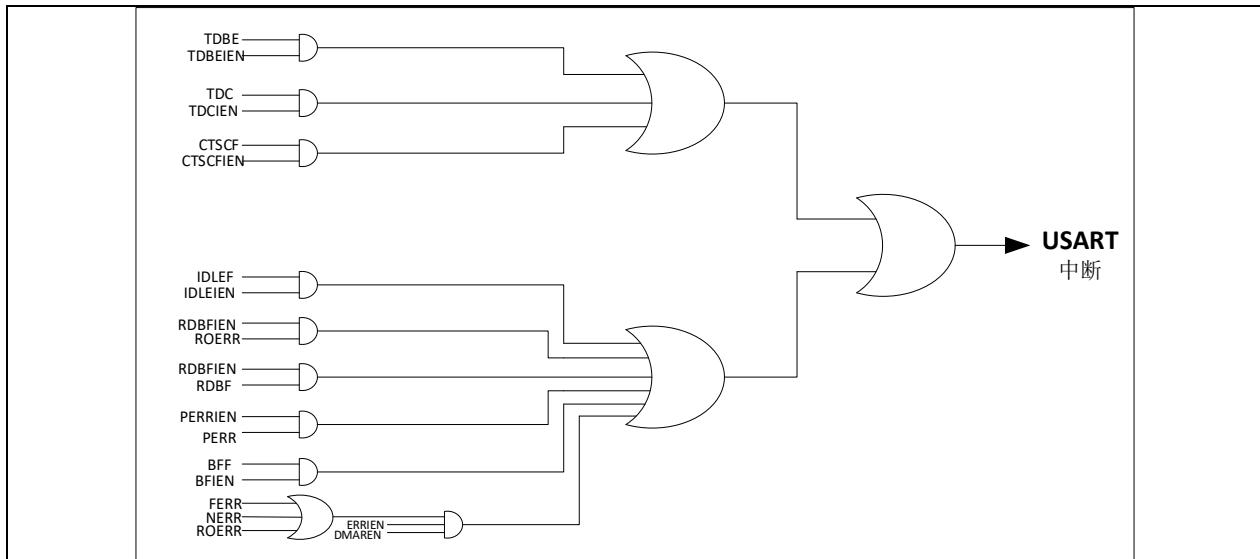
12.10 中断

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 12-12 USART中断映像图



12.11 I/O引脚控制

USART 通过五个接口外部设备进行通信，引脚定义如下：

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

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	0x00C0
USART_DT	0x04	0x0000
USART_BAUDR	0x08	0x0000
USART_CTRL1	0x0C	0x0000
USART_CTRL2	0x10	0x0000
USART_CTRL3	0x14	0x0000
USART_GDIV	0x18	0x0000
USART_RTOV	0x1C	0x0000
USART_IFC	0x20	0x0000

12.12.1 状态寄存器 (USART_STS)

域	简称	复位值	类型	功能
位 31: 18				
位 16: 12	保留	0x000000	resd	硬件强制为 0。
位 10				
位 17	CMDF	0	r	字节匹配检测标志 当接收到由 ID[7:0]定义的字节时，该位被硬件置起，由软件将其清零。 0: 无； 1: 有。
位 11	RTODF	0	r	接收器超时检测标志 当超时值达到 RTOV 寄存器编程的值，若无任何通信，该位被硬件置起，由软件将其清零。 0: 无； 1: 有。
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 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 CMplete) 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 USART_STS，再写 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 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 USART_STS，再读 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 USART_STS，再读 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	杂讯错误 (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 USART_STS，再读 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 USART_STS，再读 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 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 0x000 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: 29 保留位		0x000000	resd	硬件强制为 0。
位 28 DBN1 0x0 rw				数据位个数高位（Data bit num） 该位和 DBN0 位一起定义了数据位的个数。 10: 7 位； 00: 8 位； 01: 9 位； 11: 禁止写入，否则数据异常。
位 27 RTODEN 0 rw				接收器超时检测使能位（receiver time out detection enable） 0: 关闭； 1: 开启。
位 26 RETODIE 0 rw				接收器超时检测中断使能位（receiver time out detection interrupt enable） 0: 关闭； 1: 开启。
位 25: 21 TSDT 0x00 rw				发送器开始延迟时间（transmit start delay time） RS485 模式下一系列连续发送的第一笔数据会在数据写入后延迟一段时间后发送，以确保外部收发器已将传输方向切换为发送，该时间由 TSDT 的值决定，时间单位为 1/16 个波特率周期。
位 20: 16 TCDT 0x00 rw				发送器完成延迟时间（transmit complete delay time） RS485 模式下一系列连续发送的最后一笔数据会在数据最后一个停止位发送完成后延迟一段时间结束，以确保外部收发器转接器已将传输方向切换为接收，该时间由 TCDT 的值决定，时间单位为 1/16 个波特率周期。
位 15 保留 0 resd				保持默认值。
位 14 CMDIE 0 rw				字节匹配检测中断使能位（character match detection interrupt enable） 0: 关闭； 1: 开启。
位 13 UEN 0x0 rw				USART 使能（USART enable）

				0: 关闭; 1: 开启。
位 12	DBN0	0x0	rw	数据位个数高位 (Data bit num) 该位和 DBN1 位一起定义了数据位的个数。 10: 7 位; 00: 8 位; 01: 9 位; 11: 禁止写入, 否则数据异常。
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: 28	IDH	0x0	rw	USART 的 ID 号高 4 位 (USART identification) 可配置的 USART 的 ID 号。
位 27: 20	保留	0x000	resd	保持默认值。
位 19	MTF	0	rw	MSB 先传输 (MSB transmit first) 该位用于选择数据先传输 MSB 还是 LSB。 0: LSB; 1: MSB。 注意: 使能 MTF 时, 不支持奇偶校验。
位 18	DTREV	0	rw	DT 寄存器极性反向 (DT register polarity reverse) 0: 0: 1=H, 0=L 1: 1=L, 0=H
位 17	TXREV	0	rw	TX 引脚极性反向 (TX polarity reverse) 0: VDD=1/idle,Gnd=0/mark 1: VDD=0/mark,Gnd=1/idle
位 16	RXREV	0	rw	RX 引脚极性反向 (RX polarity reverse) 0: VDD=1/idle,Gnd=0/mark 1: VDD=0/mark,Gnd=1/idle
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引脚。 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引脚上总线空闲时钟输出的极性。 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引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 puLEXT) 在同步模式下, 使用该位来控制是否在时钟引脚上输出数据的最后一一位对应的时钟脉冲 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	IDBN	0	rw	ID 号位数 (Identification bit num) 此位用于选择 ID 号位数。

位 3: 0	IDL	0x0	rw	0: 4 位; 1: 数据位1位。 注意: 当该位置'1'时, 在7、8或9位数据模式下, ID号位数分别为低6、7或者8位, 如有配置奇偶校验则 为5、6或者7位。
位 3: 0	IDL	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: 16	保留位	0x000000	resd	硬件强制为0。
位 15	DEP	0	rw	DE信号极性选择 (DE polarity selection) 0: 高电平有效。 1: 低电平有效。
位 14	RS485EN	0	rw	RS485使能 (RS485 enable) 此位用于使能RS485模式, RS485模式下USART通过控制信号DE控制外部收发器传输方向。 0: 禁止RS485模式, 控制信号DE禁止输出, RTS引脚作RS232模式使用。 1: 使能RS485模式, 控制信号DE在RTS引脚上输出。
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。
位 2	IRDALP	0x0	rw	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 1	IRDAEN	0x0	rw	红外功能使能 (IrDA enable) 0: 关闭; 1: 开启。
位 0	ERRIEN	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。

12.12.7 保护时间和预分频寄存器 (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/smaERTCard 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;

12.12.8 接收器超时检测值寄存器 (RTOV)

域	简称	复位值	类型	功能
位 31: 24	保留位	0x00	resd	硬件强制为 0。
位 23: 0	RTOV	0x00	rw	接收器超时检测值 (receiver time out value) 单位为 1bit 位宽

12.12.9 中断标志位清除寄存器 (IFC)

域	简称	复位值	类型	功能
位 31: 18				
位 16: 12	保留位	0x00	resd	硬件强制为 0。
位 10: 0				
位 17	CMDFC	0	w1	字节匹配检测标志清除位 (character match detection flag clear)
位 11	RTODFC	0	w1	接收器超时检测标志清除位 (receiver time out detection flag clear)

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 I²S 使用。本章将分 SPI 和 I²S 分别介绍 SPI 作 SPI 或 I²S 的功能特性以及配置流程。

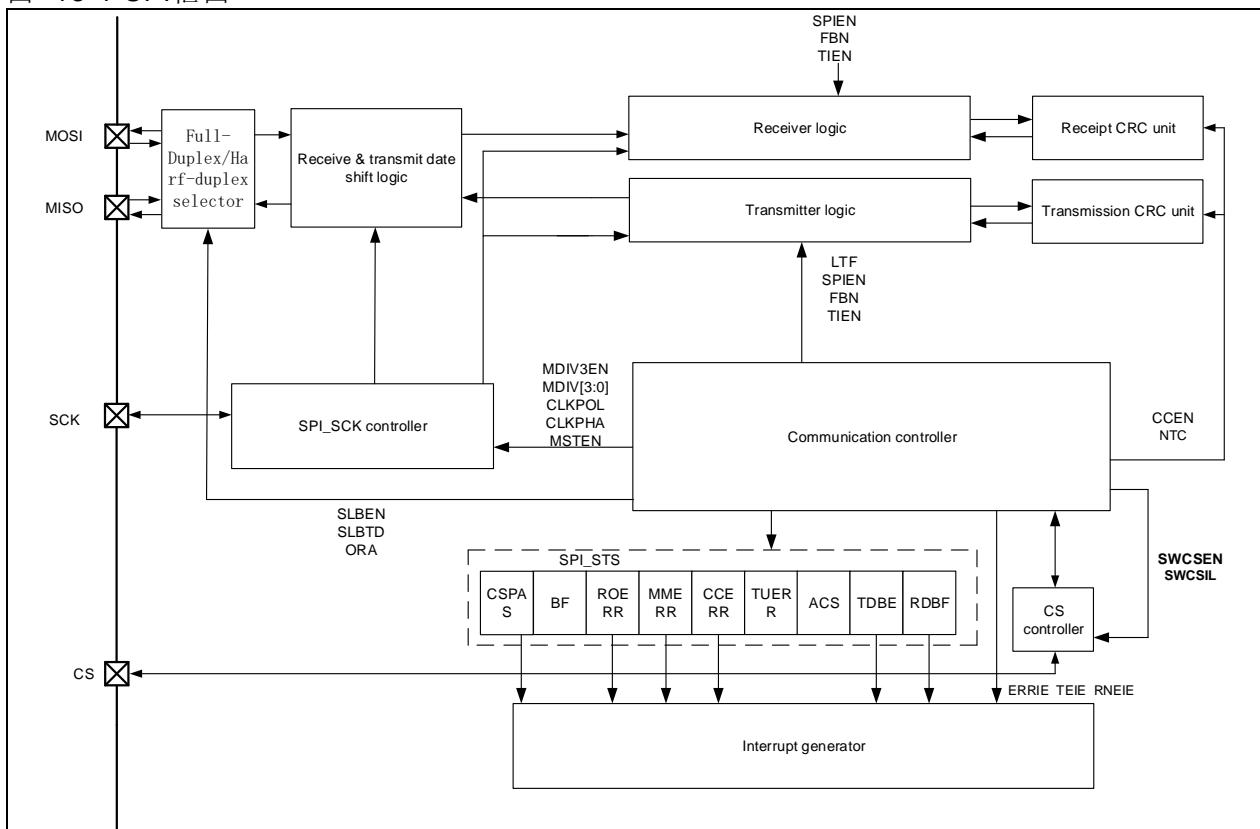
13.2 SPI 功能描述

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，同时可以通过软件编程配置使 SPI 接口兼容 TI 协议。

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_{PCLK}/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（CS 脉冲异常，接收器溢出错误，主模式错误，CRC 校验错误）

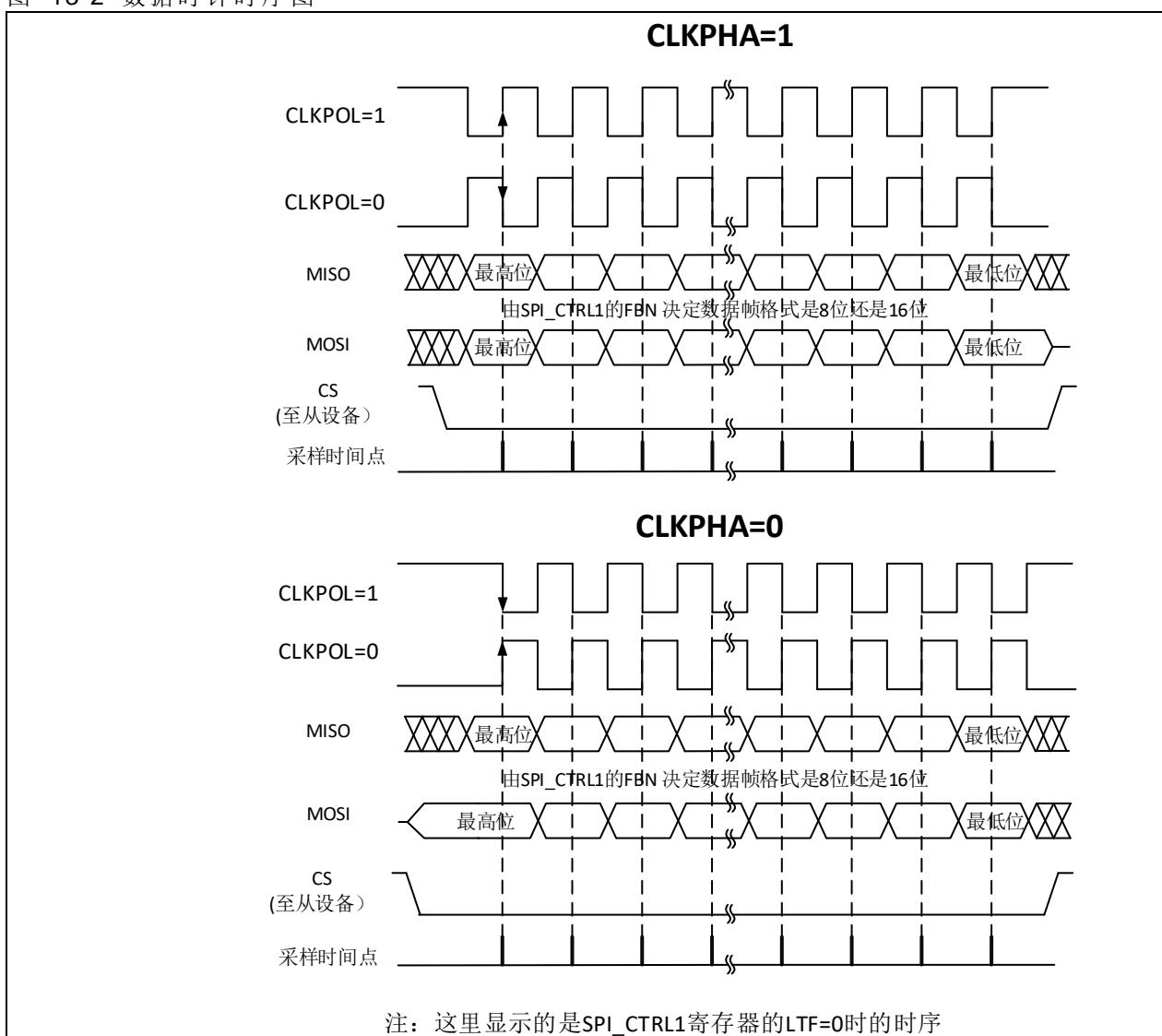
- 可编程配置的发送数据缓冲器空中断以及接收数据缓冲器满中断
- 支持 DMA 发送和接收
- 支持硬件 CRC 发送和校验
- 具备通信忙标志
- 兼容 TI 协议

时钟信号的相位和极性

SPI_CTRL1 寄存器的 CLKPOL 和 CLKPHA 位，能够组合四种可能的时序关系。CLKPOL（时钟极性）位控制在没有数据传输时时钟的空闲状态电平，此位对主模式和从模式下的设备都有效。如果 CLKPOL 被清‘0’，SCK 引脚在空闲状态保持低电平；如果 CLKPOL 被置‘1’，SCK 引脚在空闲状态保持高电平。

如果 CLKPHA（时钟相位）位被置‘1’，SCK 时钟的第二个边沿（CLKPOL 位为 0 时就是下降沿，CLKPOL 位为‘1’时就是上升沿）进行数据位的采样，数据在第二个时钟边沿被锁存。如果 CLKPHA 位被清‘0’，SCK 时钟的第一边沿（CLKPOL 位为‘0’时就是上升沿，CLKPOL 位为‘1’时就是下降沿）进行数据位采样，数据在第一个时钟边沿被锁存。

图 13-2 数据时钟时序图



13.2.2 全双工半双工选择器简述和配置流程

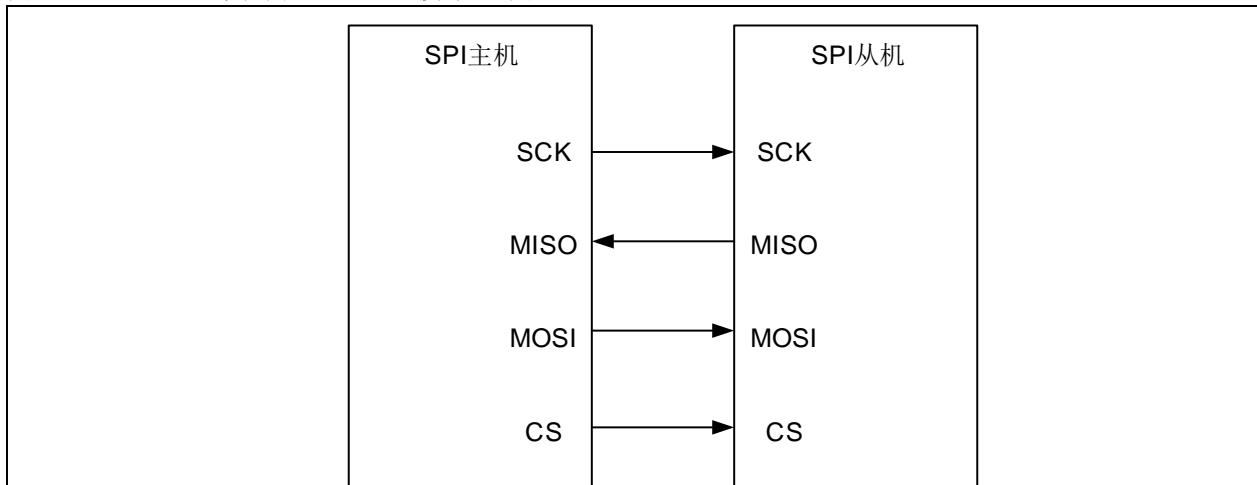
SPI 全双工半双工选择器通过软件编程配置的方式，可以使 SPI 接口作为 SPI 使用时，可以工作在双线单向全双工，单线单向只收，单线双向半双工发送和单线双向半双工接收四种同步模式。

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 13-3 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 13-4 SPI 作主机单线双向只收连接示意图

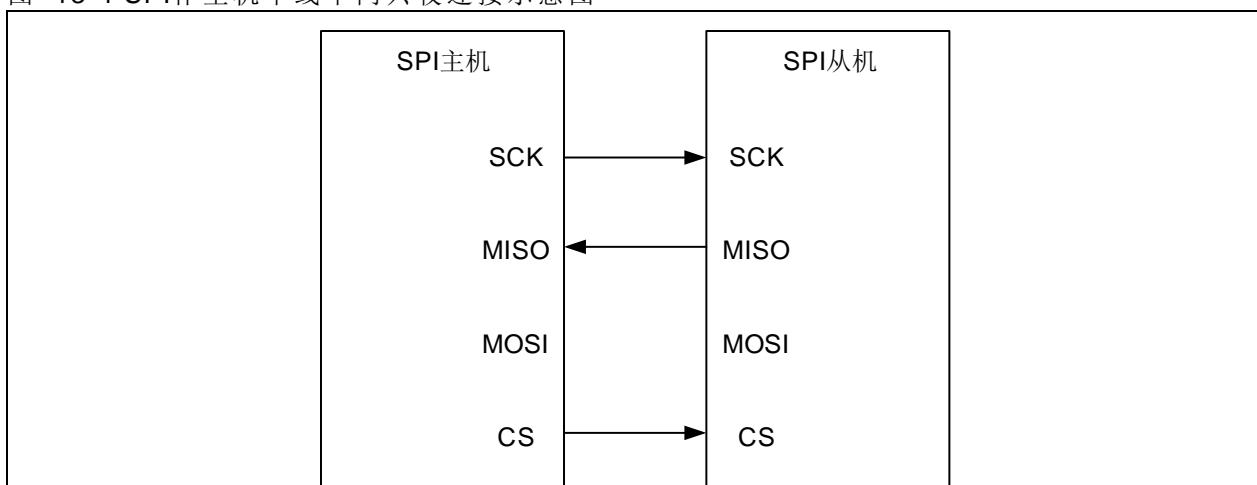
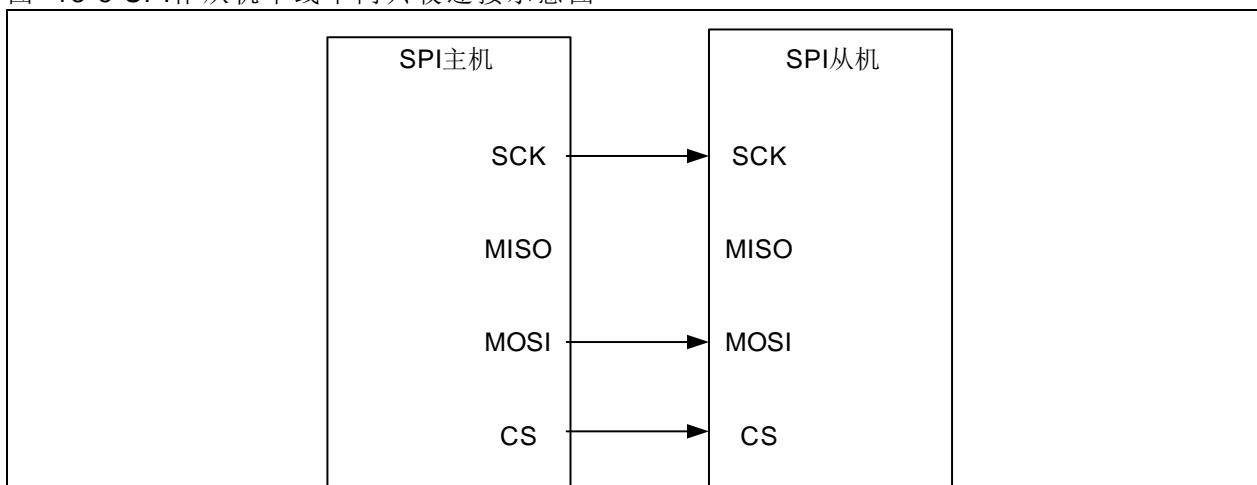


图 13-5 SPI 作从机单线双向只收连接示意图



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

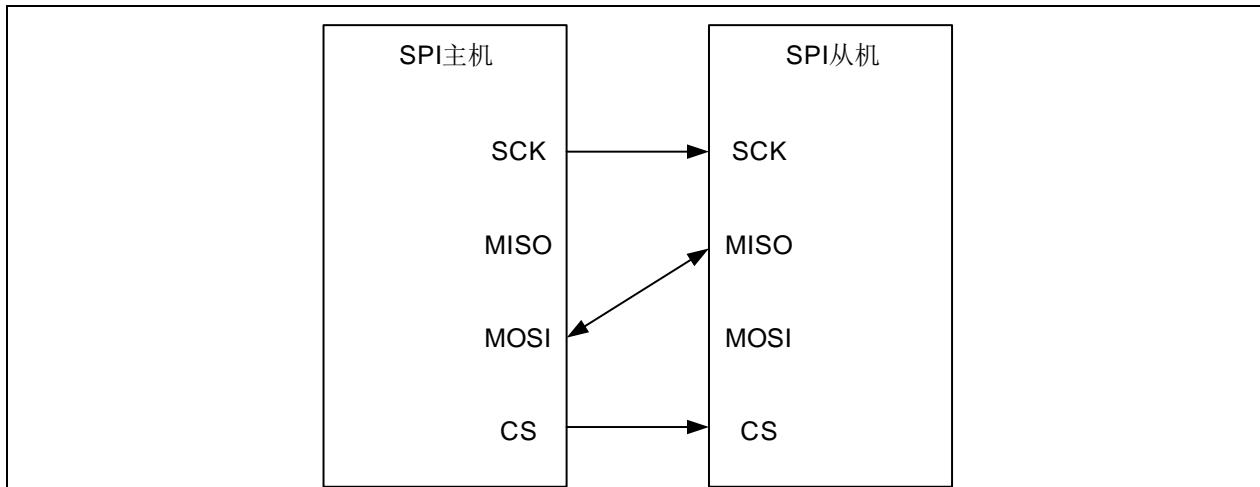
前需要等待 BF=0。

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 13-6 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_STS 再写 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_STS 再写 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] 或 MDIV3EN 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_DT 寄存器中，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_DT 寄存器地址，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_DT 寄存器中。
- 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_DT 寄存器的地址，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 TI 模式简述和配置流程

SPI 接口支持 TI 协议，用户可以通过将 TIEN 位置 1 使能 TI 模式。

使能 TI 模式后，SPI 接口将按照 TI 协议要求产生通信时钟 SPI_CLK，这意味着 SPI_CLK 的极性和相位会强制符合 TI 协议要求，用户无需也无法通过配置 CLKPOL 和 CLKPHA 来改变 SPI_CLK 的极性和相位。

使能 TI 模式后，SPI 接口将按照 TI 协议要求产生 CS 信号，这意味着 CS 的输出和输入将强制符合 TI 协议的要求，用户无需也无法通过配置 SWCSEN 位，SWCSIL 位，HWCSOE 位来管理 CS 信号。

使能 TI 模式后，SPI 从机只会在数据发送期间控制 MISO 引脚，这意味着 SPI 从机的 MISO 引脚将会在空闲状态保持高阻态。

使能 TI 模式后，SPI 接口作从机使用时，SPI 接口会在数据传输期间侦测错误的 CS 脉冲，并在侦测到错误的 CS 脉冲后置起 CSPAS 位(该位可以通过软件读 SPI_STS 清除)，此时 SPI 会忽略掉该错误的脉冲，但由于此时 CS 信号已经出现异常，软件应当关闭 SPI 从机，重新配置 SPI 主机，再打开 SPI 从机以重新开始通信。

13.2.8 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_DT 中的数据保存到发送器中的数据缓冲器(有别于 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

13.2.9 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_DT，此时读 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_DT 时，RDBF 置位，代表 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_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_DT 寄存器和 SPI_STS 寄存器可将 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。

13.2.10 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；

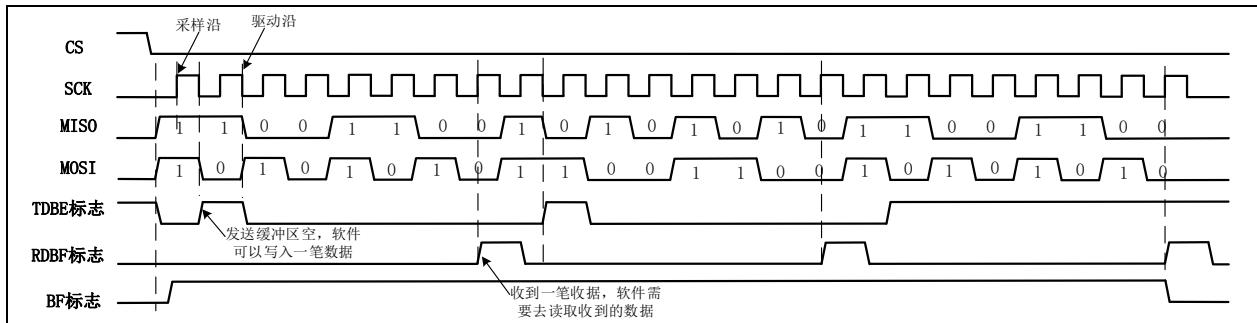
CLKPOL=0, CLKPHA=0：SCK 空闲输出低电平，第一个边沿作为采样边沿；

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 13-7 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；

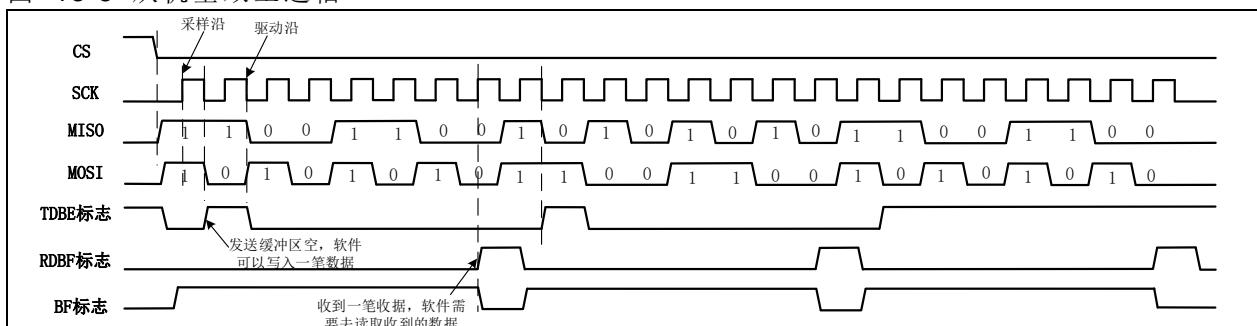
CLKPOL=0, CLKPHA=0: SCK 空闲输出低电平，第一个边沿作为采样边沿；

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 13-8 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；

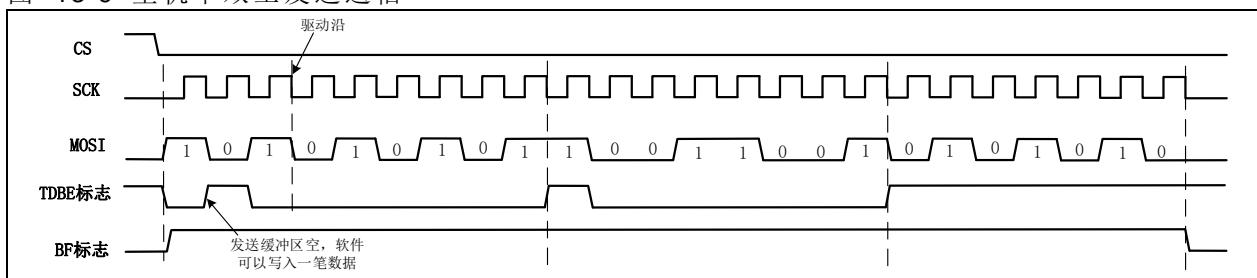
SLBTD=1: 发送模式；

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 13-9 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；

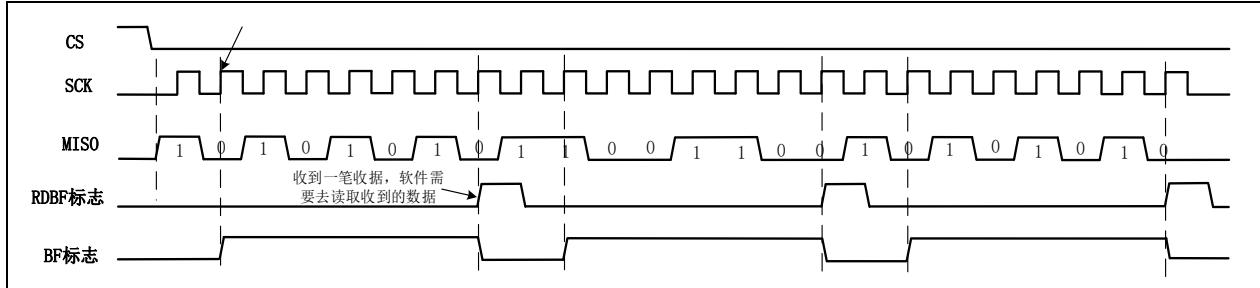
SLBTD=0: 接收模式；

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 13-10 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;

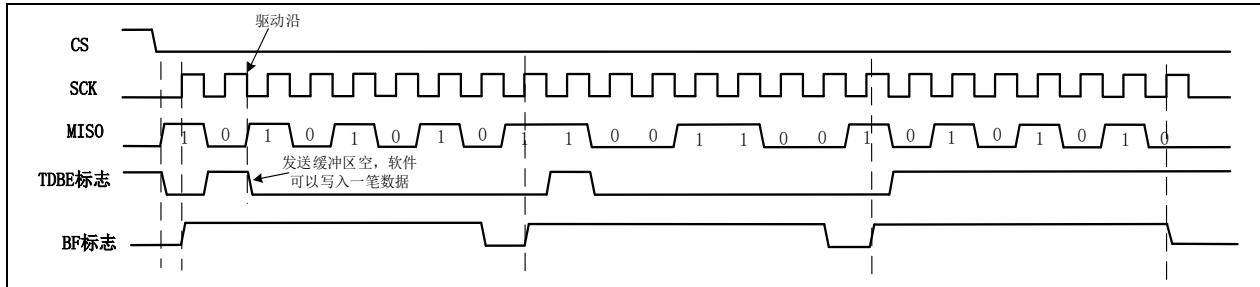
SLBTD=1: 发送模式;

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 13-11 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;

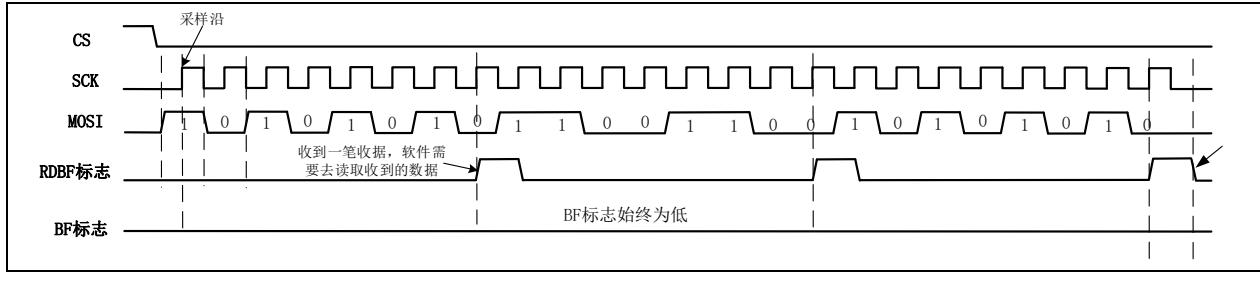
SLBTD=0: 接收模式;

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 13-12 主机半双工接收通信

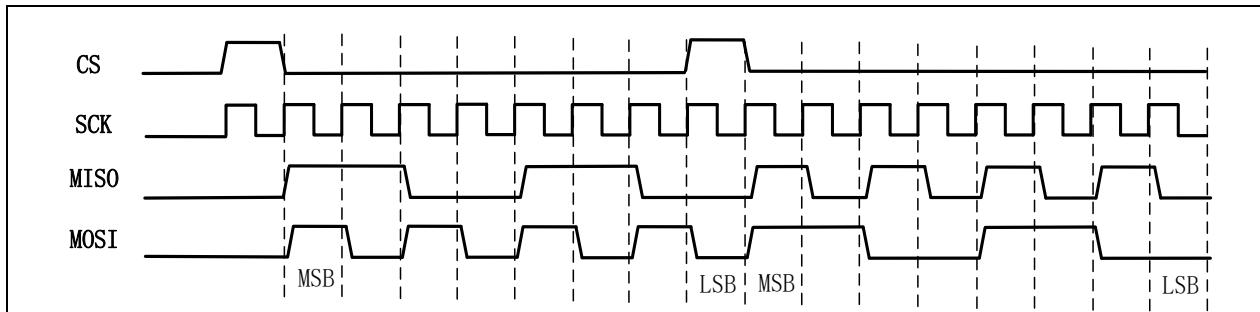


13.2.11 TI模式通信时序

SPI 接口支持 TI 模式，用户可以通过 TIEN 位置 1 使能该模式。

TI 模式下，连续与不连续通信的时序图稍有区别。当待发送数据在当前发送数据帧最后一位对应时钟 SCK 的上升沿之前写入，则通信连续，每笔数据的中间不存在 dummy CLK，主机在发送当前数据帧的最后一位数据的同时发出 CS 有效脉冲。

图 13-13 TI模式连续通信时序



TI 模式下，当待发送数据在当前发送数据帧最后一位对应时钟 SCK 的上升沿与下降沿之间写入，则每笔数据的中间存在一个 dummy CLK.

图 13-14 TI模式带dummy CLK的连续通信时序

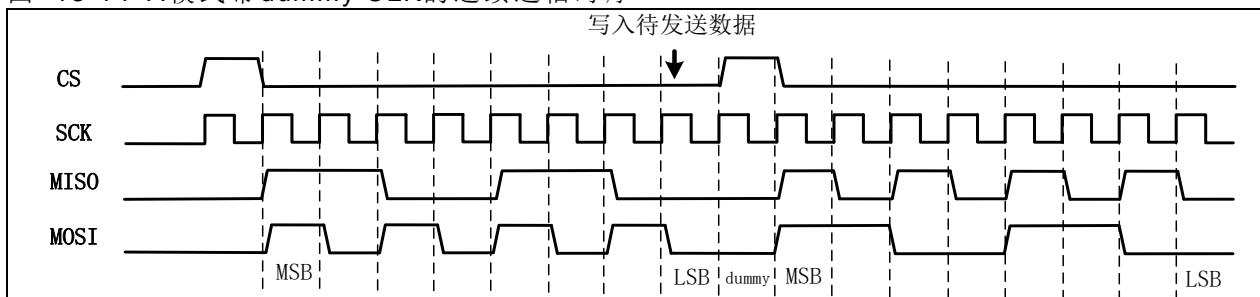
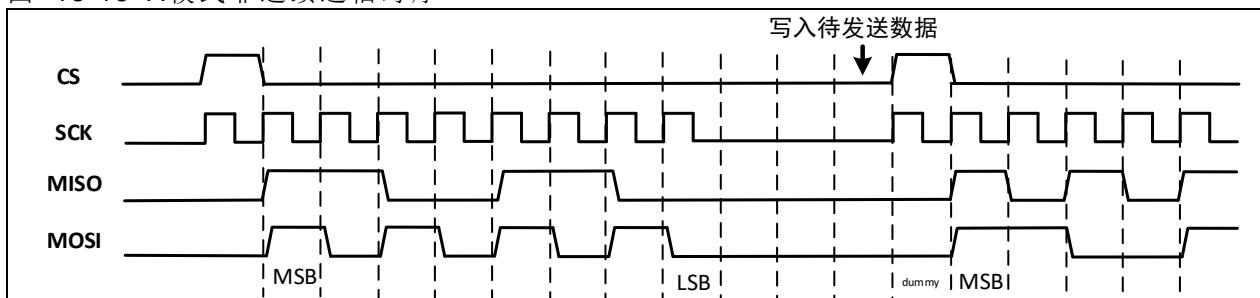
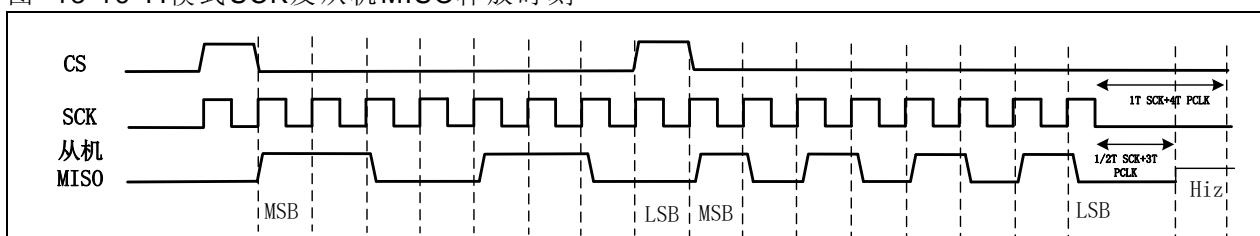


图 13-15 TI模式非连续通信时序



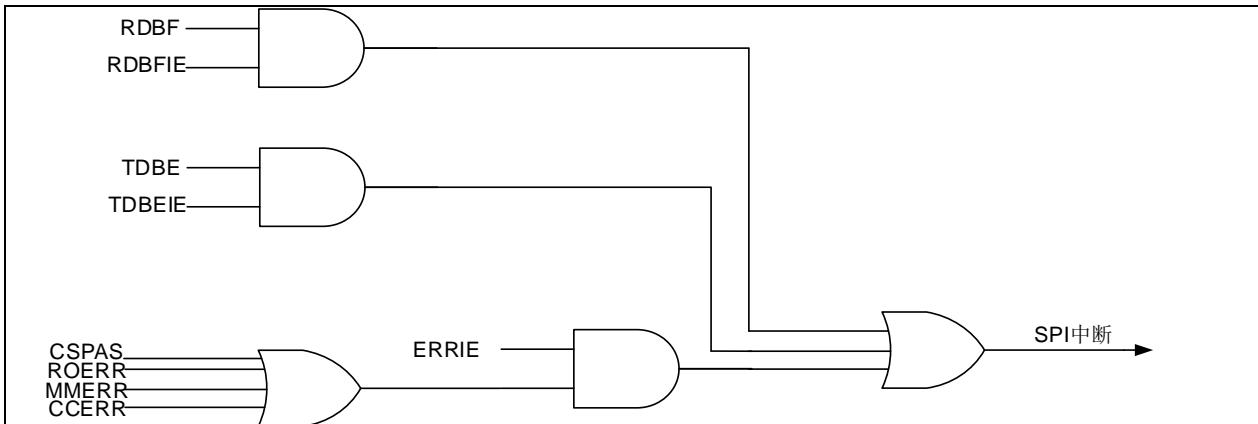
TI 模式下，当待发送数据在当前发送数据帧最后一位对应时钟 SCK 的下降沿之后写入，则主机固定在 $1T_{SCK} + 4T_{PCLK}$ 后才能重新发出有效 SCK 时钟，从机在接收当前数据帧最后一位时，仍然没有检测到有效的 CS 脉冲，则在 $1/2T_{SCK} + 3T_{PCLK}$ 后关闭 MISO 的输出功能，控制 MISO 浮空。

图 13-16 TI模式SCK及从机MISO释放时刻



13.2.12 中断

图 13-17 SPI中断



13.2.13 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。

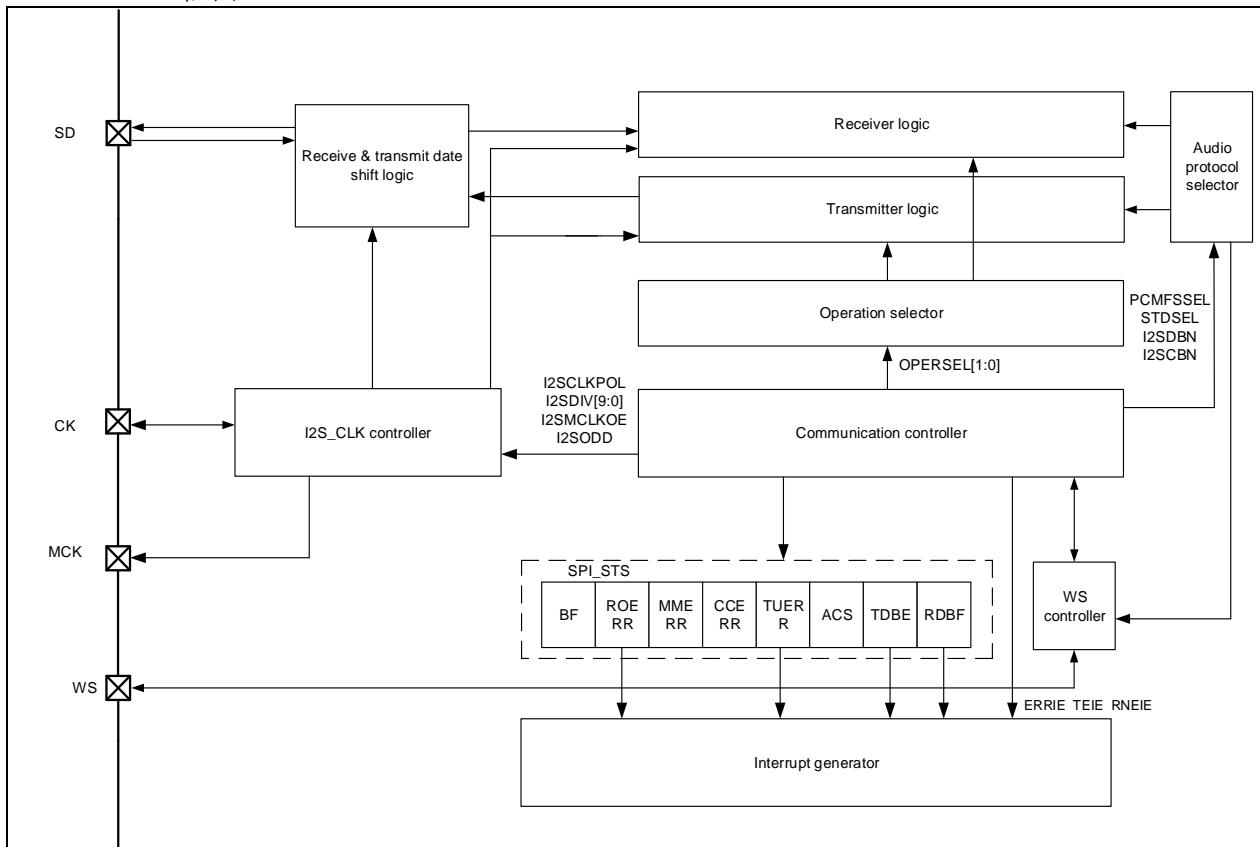
CS: 片选信号。这是一个可选的管脚，用来选中主/从设备，具体使用方式可以参见 CS 控制器章节。

13.3 I²S功能描述

13.3.1 I^2S 简述

I²S 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。

$|^2S$ 的框图如下图所示：

图 13-18 I²S 框图

SPI 接口作为 I²S 使用时主要特征如下：

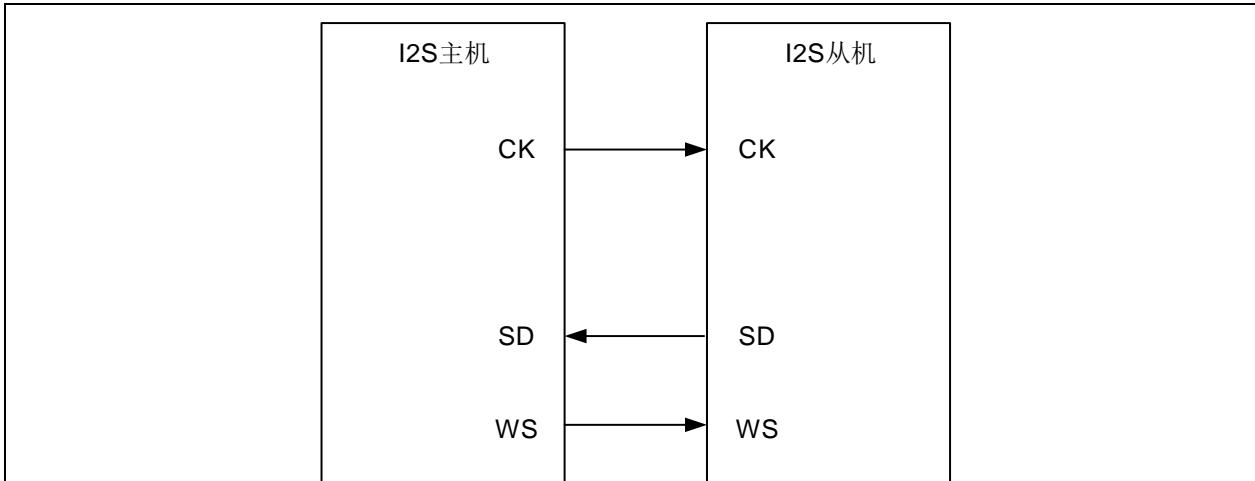
- 可编程配置的操作模式
 - 从设备发送
 - 从设备接收
 - 主设备发送
 - 主设备接收
- 可编程配置的时钟极性
- 可编程配置的时钟频率（8KHz 到 192KHz）
- 可编程配置的数据位数（16位， 24位， 32位）
- 可编程配置的声道位数（16位， 32位）
- 可编程配置的音频协议
 - I²S 飞利浦标准
 - 高字节对齐标准（左对齐）
 - 低字节对齐标准（右对齐）
 - PCM 标准（带长或短帧同步的通道帧）
- 支持 DMA 传输
- 支持提供频率固定比例为 256 倍 F_s（音频采样频率）的外设主时钟

13.3.2 操作模式选择器简述和配置流程

SPI 接口作 I²S 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。

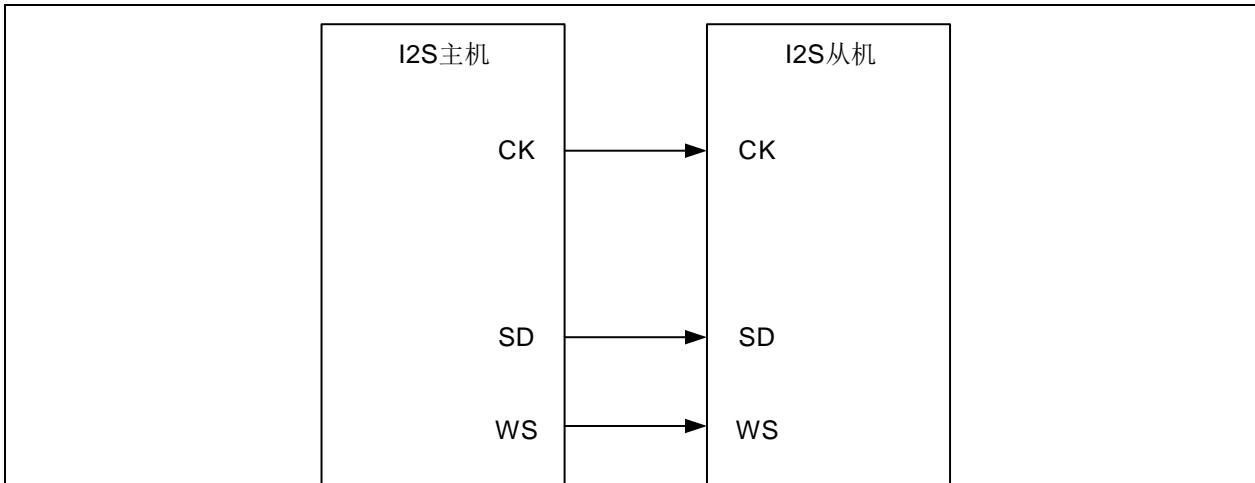
从设备发送：

置位 I²SMSEL 位，配置 OPERSEL[1: 0]位为 00，I²S 将工作在从设备发送模式下

图 13-19 I²S从设备发送连接示意图

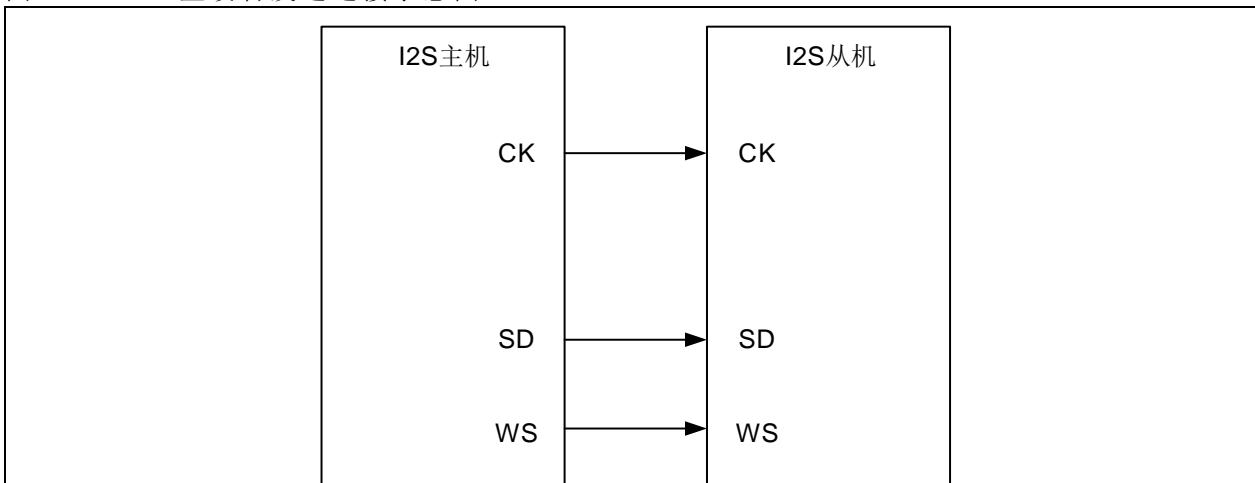
从设备接收:

置位 I2SMSEL 位, 配置 OPERSEL[1: 0]位为 01, I²S 将工作在从设备接收模式下

图 13-20 I²S从设备接收连接示意图

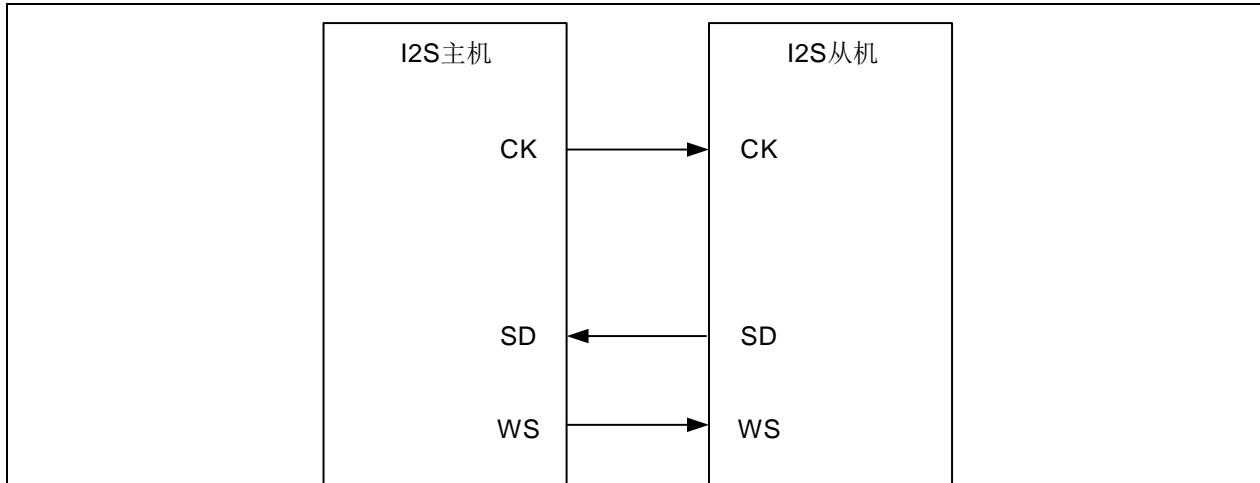
主设备发送:

置位 I2SMSEL 位, 配置 OPERSEL[1: 0]位为 10, I²S 将工作在主设备发送模式下

图 13-21 I²S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, I²S 将工作在主设备接收模式下

图 13-22 I²S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I²S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
 - I2SDBN =0：16位
 - I2SDBN =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_DT寄存器，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_DT寄存器，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_DT寄存器，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_DT寄存器，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_DT寄存器，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_DT寄存器，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

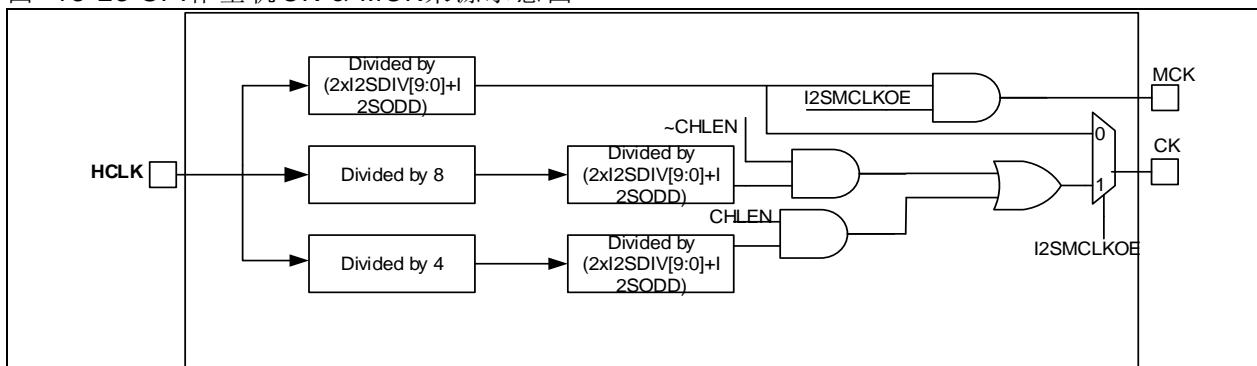
13.3.4 I2S_CLK控制器简述和配置流程

SPI接口作I²S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I²S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13-23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13-23。

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13-23。

图 13-23 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
180	No	192000	14	1	193965.5	1.02%	7	1	187500	2.34%
180	No	96000	29	1	95338.98	0.69%	14	1	96982.76	1.02%
180	No	48000	58	1	48076.92	0.16%	29	1	47669.49	0.69%
180	No	44100	64	0	43945.31	0.35%	32	0	43945.31	0.35%
180	No	32000	88	0	31960.23	0.12%	44	0	31960.22	0.12%
180	No	22050	127	1	22058.82	0.04%	64	0	21972.66	0.35%
180	No	16000	176	0	15980.11	0.12%	88	0	15980.11	0.12%
180	No	11025	255	0	11029.41	0.04%	127	1	11029.42	0.04%
180	No	8000	351	1	8001.42	0.02%	176	0	7990.06	0.12%
180	Yes	96000	3	1	100446.4	4.63%	3	1	100446.4	4.63%
180	Yes	48000	7	1	46875	2.34%	7	1	46875	2.34%
180	Yes	44100	8	0	43945.31	0.35%	8	0	43945.31	0.35%
180	Yes	32000	11	0	31960.23	0.12%	11	0	31960.23	0.12%
180	Yes	22050	16	0	21972.66	0.35%	16	0	21972.66	0.35%
180	Yes	16000	22	0	15980.11	0.12%	22	0	15980.11	0.12%
180	Yes	11025	32	0	10986.33	0.35%	32	0	10986.33	0.35%
180	Yes	8000	44	0	7990.06	0.12%	44	0	7990.06	0.12%

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 I²S 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 I²S 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_DT寄存器地址，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_DT寄存器中。
 - 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_DT寄存器的地址，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 I²S 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 I²S 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：

- I²S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。
 - I²S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）。
 - I²S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。
 - I²S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。

- I²S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I²S。
 - 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I²S。
 - 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I²S。

下面给出发送器和接收器的配置流程

I²S 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I2S_CLK控制器
- 配置DMA(若需要开启DMA传输)
- 置位I2SEN位开启I²S

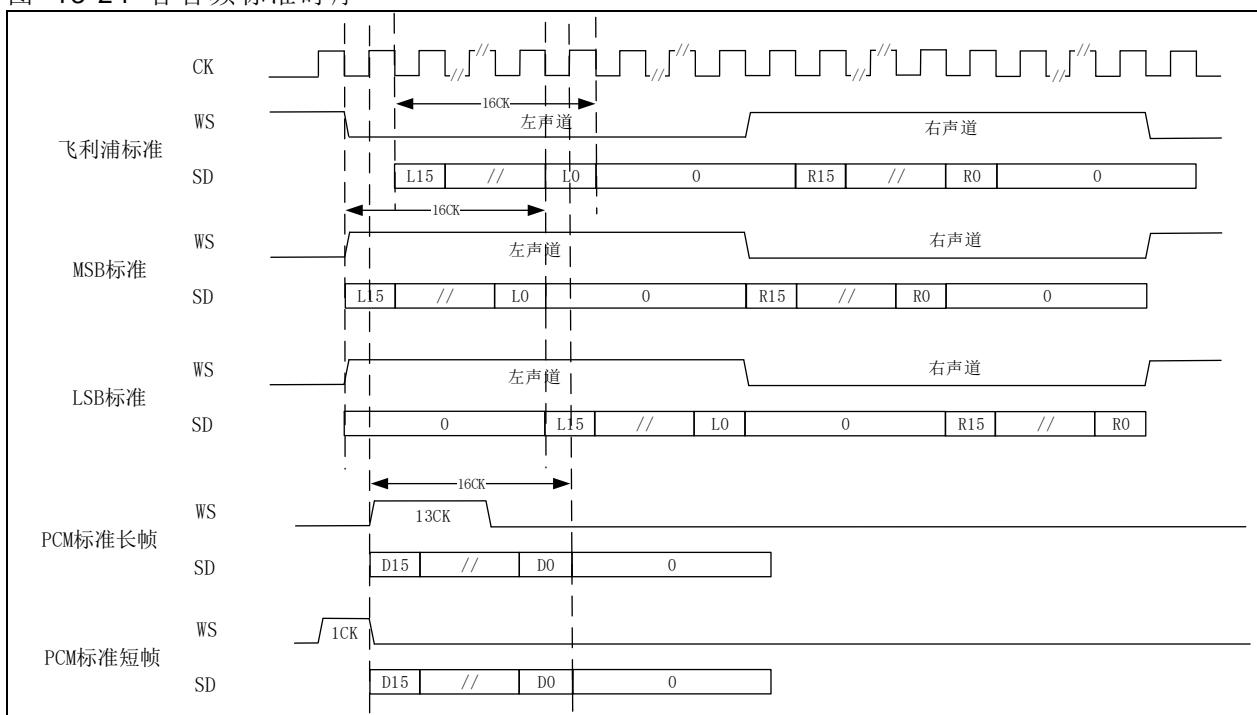
I²S 接收器配置流程:

- 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I²S

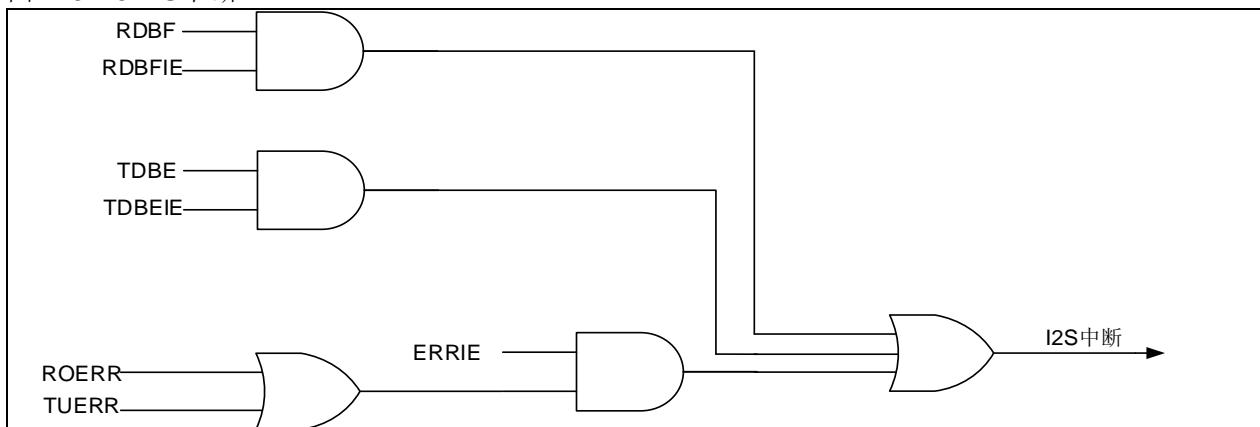
13.3.7 I²S通信时序

I²S 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 13-24 各音频标准时序



13.3.8 中断

图 13-25 I²S中断

13.3.9 IO管脚控制

SPI 接口作 I²S 使用时，I²S 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 I²S 和 SPI 使用，所以 I²S 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。

13.4 SPI寄存器

必须用字（32 位）的方式操作这些外设寄存器。

表 13-2 SPI寄存器列表及其复位值

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) (I2S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]位在 SPI_CTRL2 寄存器, 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 I²S 模式下，SPI_CTRL1 寄存器需置 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: 10	保留	0x00	resd	硬件强制为 0
位 9	MDIV3EN	0x0	rw	主模式时钟频率三分频使能 (Master clock frequency3 division enable) 0: 关闭； 1: 开启。 注：该位开启时，MDIV[3: 0]无效，SPI 时钟被强制为 PCLK 三分频。
位 8	MDIV[3]	0x0	rw	主模式时钟频率分频系数 (Master clock frequency division) 详见 MDIV[2: 0]在 SPI_CTRL1 寄存器。
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	TIEN	0x0	rw	TI 模式使能 (TI mode enable) 0: 关闭 (Motorola 模式)； 1: 开启 (TI 模式)。 注：该位在 I ² S 模式下没有用，在 I ² S 模式下需保持为 0。
位 3	保留	0x0	resd	保持默认值。
位 2	HWCSONE	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: 9	保留	0x00	resd	硬件强制为 0
位 8	CSPAS	0x0	ro	CS 脉冲异常置位标志 (CS pulse abnormal setting flag) 0: 无异常; 1: 有异常置位; 注: 该位用于 TI slave mode, 由软件读 STS 寄存器清零。
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_STS 寄存器, 再写 SPI_CTRL1 寄存器)。 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_STS 寄存器)。 0: 无; 1: 有。 注: 该位只在 I ² 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 ² 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) (I²S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 SPIxRCR寄存器 (SPI_RCRC) (I2S模式下不使用)

域	简称	复位值	类型	功能
位 15: 0	RCRC	0x0000	ro	<p>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_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。</p>

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	<p>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_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。</p>

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	<p>I²S 模式选择 (I²S mode select) 0: SPI 模式; 1: I²S 模式。</p>
位 10	I2SEN	0x0	rw	<p>I²S 使能 (I²S enable) 0: 关闭; 1: 开启。</p>
位 9: 8	OPERSEL	0x0	rw	<p>I²S 操作选择 (I²S 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。</p>
位 7	PCMFSSEL	0x0	rw	<p>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。</p>
位 6	保留位	0x0	resd	保持默认值。
位 5: 4	STDSEL	0x0	rw	<p>I²S 标准选择 (I²S 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。</p>
位 3	I2SCLKPOL	0x0	rw	<p>I²S 时钟极性 (I²S 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。</p>
位 2: 1	I2SDBN	0x0	rw	<p>I²S 数据位个数 (I²S data bit num) 00: 16 位; 01: 24 位; 10: 32 位; 11: 不允许。</p>
位 0	I2SCBN	0x0	rw	I ² S 声道位个数 (I ² S channel bit num)

该位只有在 I²S 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 ² S 主设备时钟输出使能 (I ² S Master clock output enable) 0: 关闭； 1: 开启。
位 8	I2SODD	0x0	rw	I ² S 分频系数配置奇数 (Odd result for I ² 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 ² S 分频系数 (I ² S division) I2SDIV[9: 0]禁止设置为 0 或者 1。

14 定时器 (TIMER)

AT32M412/AT32M416 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 14.1~14.4 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	ETR 输入模式	刹车输入
高级控制定时器	TMR1	16	向上 向下 向上/向下	16 位	1~65536	支持	4	支持	支持	支持
通用定时器	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
	TMR4	16	向上 向下 向上/向下	8 位	1~65536	支持	2	支持	不支持	支持
	TMR9	16	向上 向下 向上/向下	8 位	1~65536	支持	1	不支持	不支持	支持
	TMR10	16	向上 向下 向上/向下	8 位	1~65536	支持	不支持	不支持	不支持	不支持
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
高级控制定时器	TMR7	16	向上	不支持	1~65536	支持	不支持	不支持	不支持	不支持
	TMR1	16	向上 向下 向上/向下	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	定时器同步 /ADC/DAC
	TMR4	16	向上 向下 向上/向下	支持	支持	支持	支持	不支持	不支持	定时器同步 ADC/DAC
	TMR9	16	向上 向下 向上/向下	支持	支持	支持	支持	不支持	不支持	定时器同步 ADC/DAC
通用定时器	TMR10	16	向上 向下 向上/向下	支持	支持	支持	支持	不支持	不支持	定时器
	TMR11	16	向上 向下 向上/向下	支持	支持	支持	支持	不支持	不支持	ADC/DAC
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	ADC/DAC

14.1 基本定时器 (TMR6和TMR7)

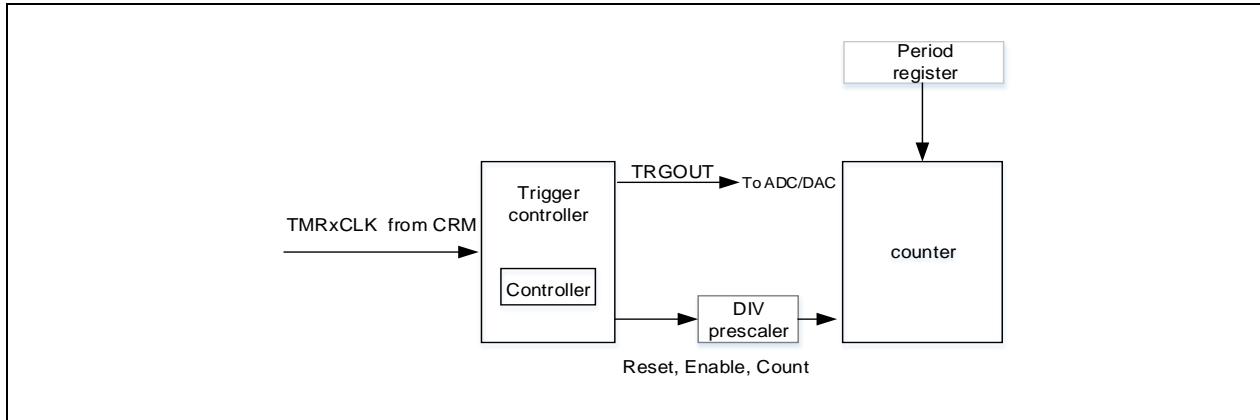
14.1.1 TMR6和TMR7简介

基本定时器 (TMR6 和 TMR7) 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。可用于简单的定时功能以及为 DAC 提供时钟。

14.1.2 TMR6和TMR7的主要功能

- 16位向上计数器，可自动装载
- 16位预分频器，用于对TMR_CLK时钟分频，分频系数为1~65536之间的任意数值
- 触发DAC的同步电路

图 14-1 基本定时器框图

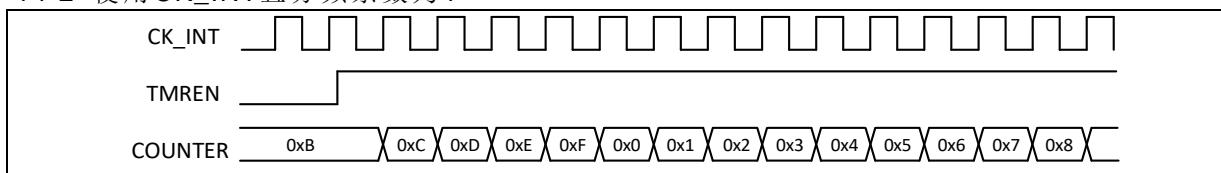


14.1.3 TMR6和TMR7的功能

14.1.3.1 计数时钟

TMR6 和 TMR7 由内部时钟源（CK_INT）经由预分频器提供计数器计数。

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。

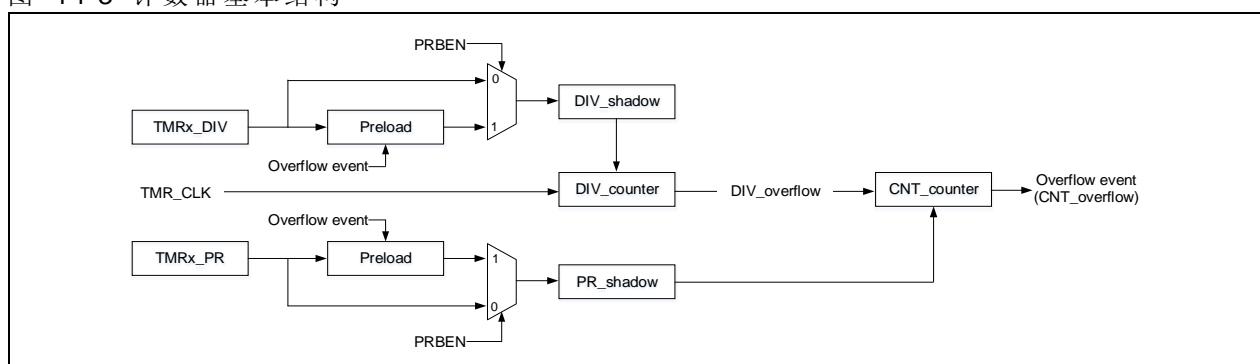
TMRx_DIV 寄存器用于设置计数器计数频率，每（DIV[15:0]+1）个计数时钟周期，计数器计数一次。和 TMRx_PR 寄存器类似，开启周期缓冲功能后，TMRx_DIV 寄存器值在溢出事件时更新至它的影子寄存器。

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 OVFEN=1 将禁止溢出事件产生。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 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

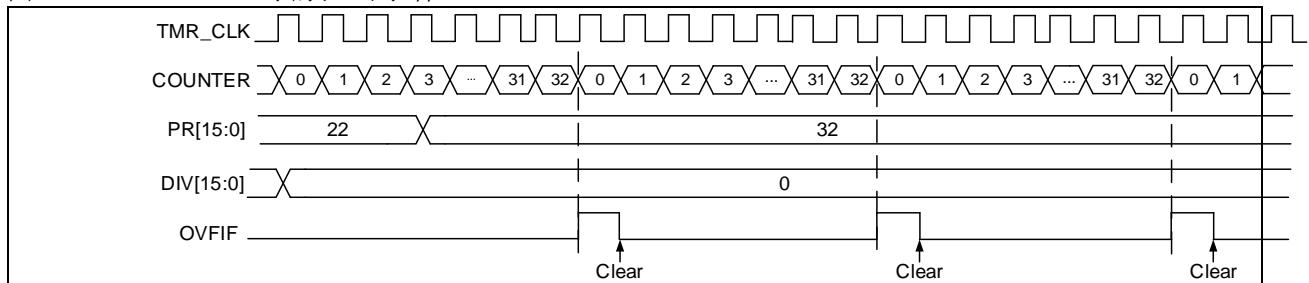


图 14-5 PRBEN=1 时的溢出事件

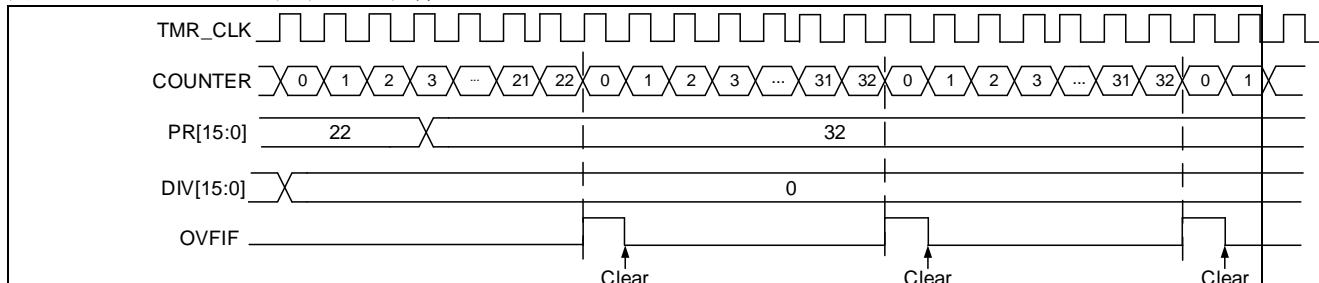
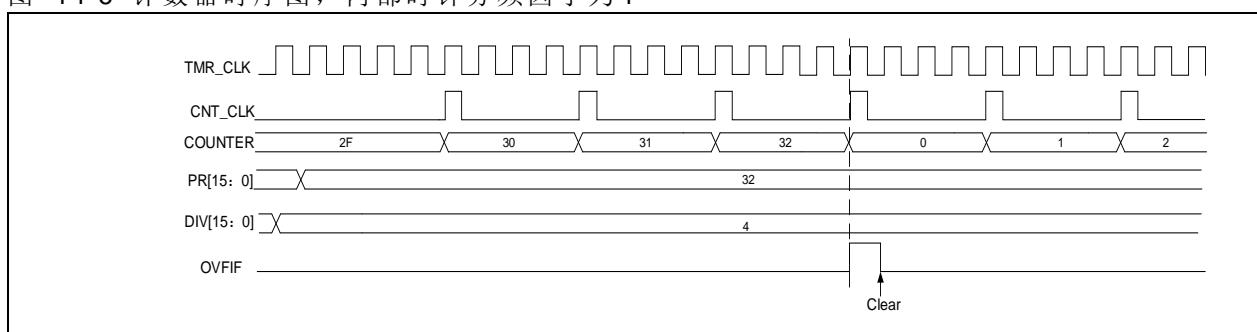


图 14-6 计数器时序图，内部时钟分频因子为 4



14.1.3.3 OVFIF 位重映射

TMRx_CTRL1 寄存器中 OVFIFRMP 位强制将溢出中断标记 OVFIF 复制到定时器计数值寄存器 $\text{TMRx_CVAL}[31]$ 中。这样便可读取计数值的同时读取 OVFIF 来判断溢出中断标记是否置起，有利于简化计算。

14.1.3.4 调试模式

当微控制器进入调试模式（Cortex®-M4F 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。

14.1.4 TMR6和TMR7寄存器

必须用字（32位）的方式操作这些外设寄存器。

下表中将 TMR6、7 的所有寄存器映射到一个 16 位可寻址（编址）空间。

表 14-2 TMR6和TMR7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_SWEVT	0x14	0x0000
TMRx_CVAL	0x24	0x0000
TMRx_DIV	0x28	0x0000
TMRx_PR	0x2C	0x0000

14.1.4.1 TMR6和TMR7控制寄存器1 (TMRx_CTRL1)

域	简称	复位值	类型	功能
位 15: 12	保留	0x0	resd	保持默认值。
位 11	OVFIFRMP	0x0	rw	OVFIF 状态位重映射 (OVFIF status bit remapping) 0: 无重映射; 1: 使能重映射, OVFIF 位复制到 TMRx_CVAL[31]。
位 10: 8	保留	0x0	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和TMR7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: 软件溢出; 001: 使能; 010: 溢出;
位 3: 0	保留	0x0	resd	保持默认值。

14.1.4.3 TMR6和TMR7 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	OVFIEN	0x0	rw	溢出中断使能 (Overflow interrupt enable) 0: 关闭; 1: 开启。

14.1.4.4 TMR6和TMR7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 OVREN=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和TMR7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	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。

14.1.4.6 TMR6和TMR7计数值 (TMRx_CVAL)

域	简称	复位值	类型	功能
位 31	OVFICPY	0x0	ro	OVFIF 副本 (OVFIF copy) OVFIFRMP 为 0 时, 该位保留; OVFIFRMP 为 1 时, 该位为 OVFIF 位的只读副本。
位 30: 16	保留	0x0000	resd	保持默认值。
位 15: 0	CVAL	0x0000	rw	计数值 (Counter value)

14.1.4.7 TMR6和TMR7分频系数 (TMRx_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。

14.1.4.8 TMR6和TMR7周期寄存器 (TMRx_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。

14.2 通用定时器（TMR3至TMR4）

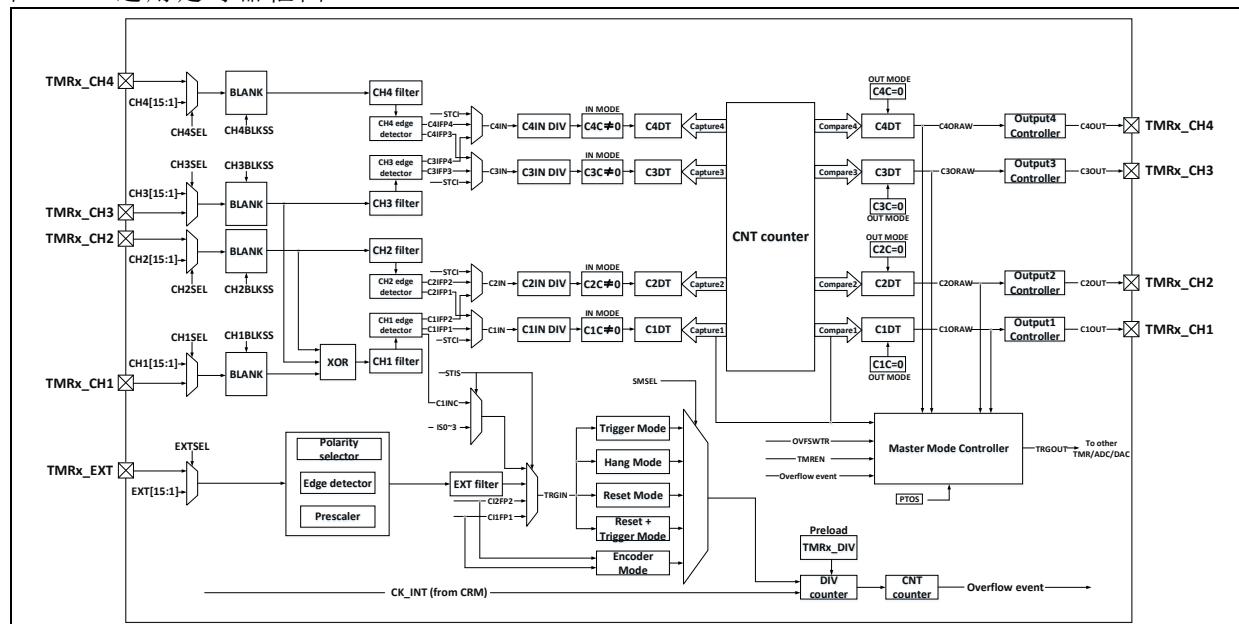
14.2.1 TMR3至TMR4简介

通用定时器 TMR3 至 TMR4 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至TMR4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。
- PWM模式支持非对称模式和组合模式
- 支持输入捕获消隐模式
- 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 DMA Burst 传输

图 14-7 通用定时器框图

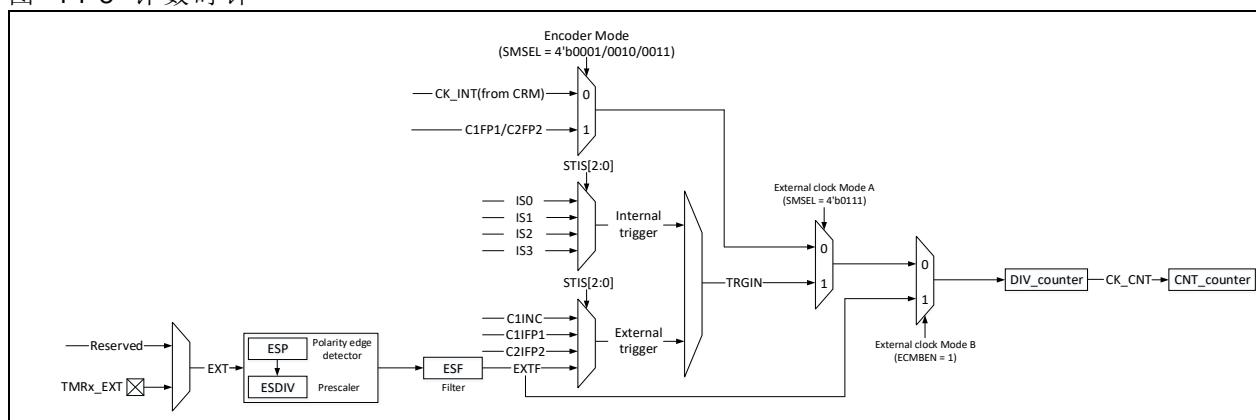


14.2.3 TMR3至TMR4功能描述

14.2.3.1 计数时钟

TMR3 至 TMR4 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

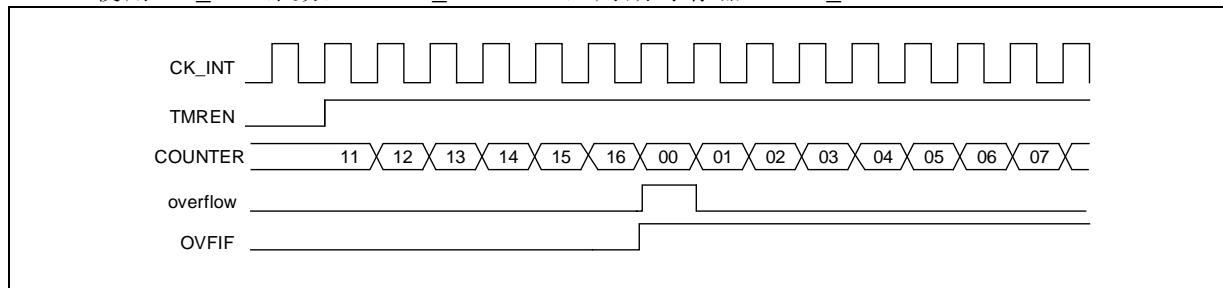


内部时钟 (CK_INT)

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=4'b0111 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。

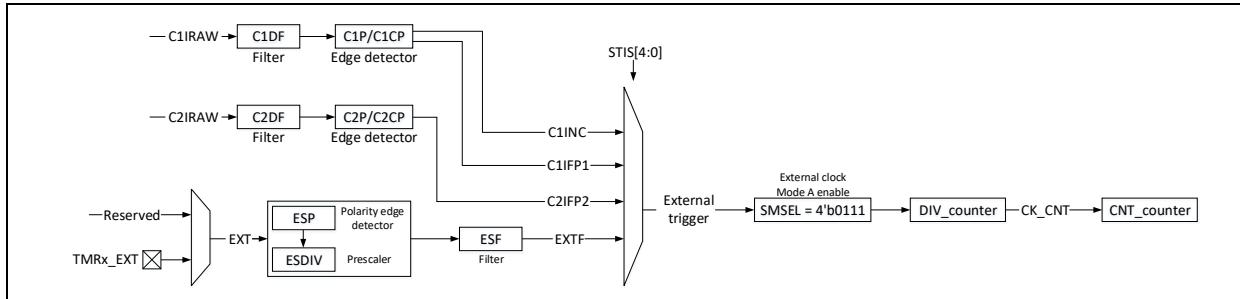
若要使用外部时钟模式 A, 可按如下步骤配置:

- 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[2:0], 设置 TRGIN 信号来源。
- 配置 TMRx_STCTRL 寄存器 SMSEL=4'b0111, 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

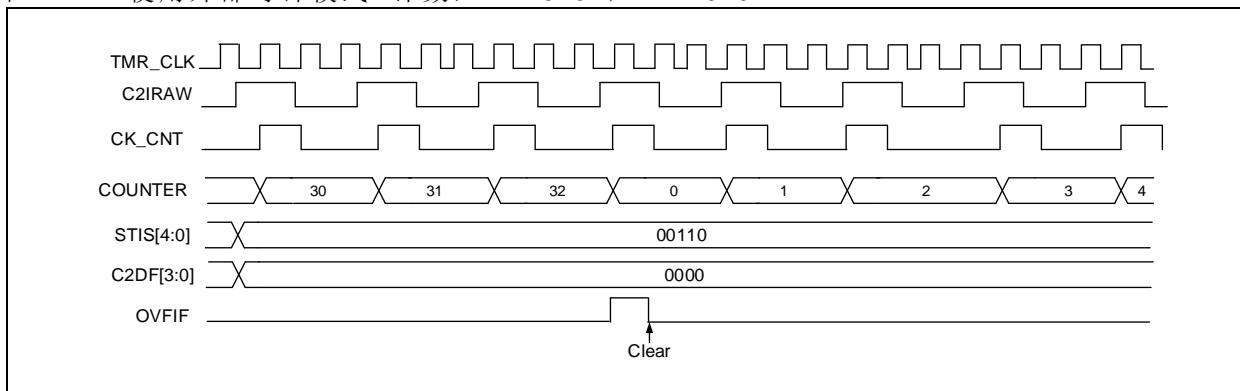
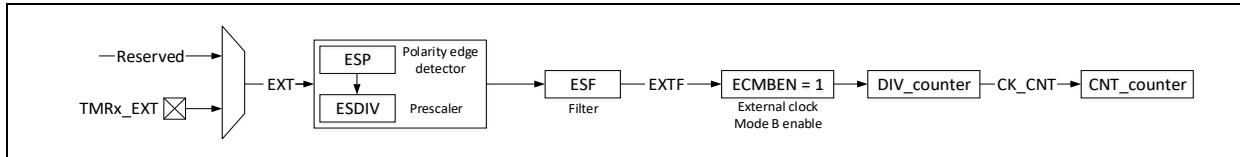
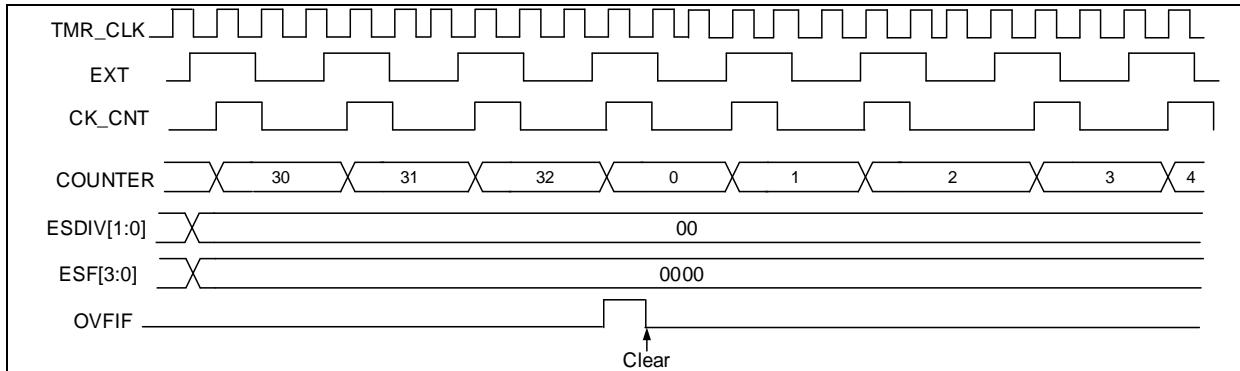


图 14-12 外部时钟模式 B 框图



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 至 TMR4 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 TMRx_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：

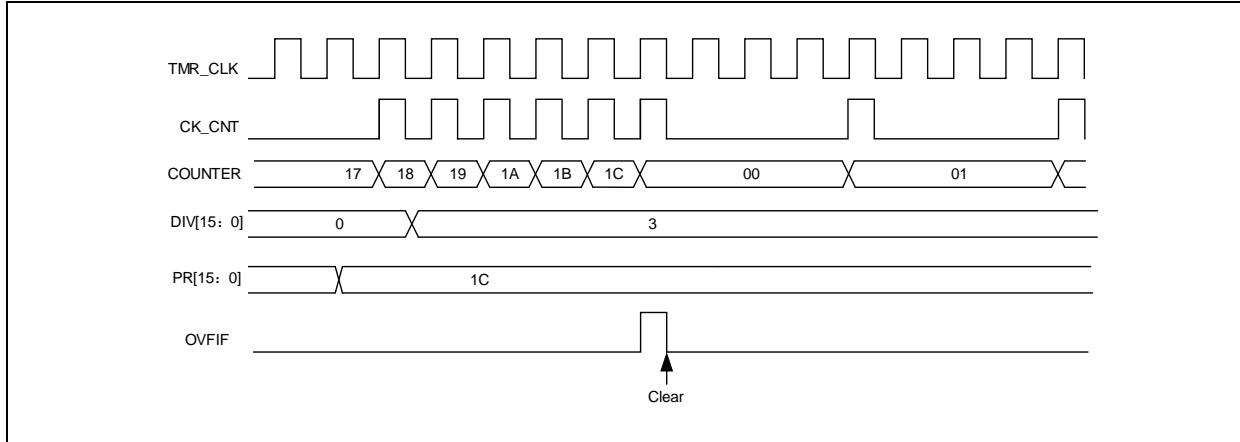
- - 配置 TMRx_PR 寄存器，设置计数器计数周期。
- - 配置 TMRx_DIV 寄存器，设置计数器计数频率。
- - 配置 TMRx_CTRL1 寄存器 TWCMSEL[1:0]位，设置计数器计数模式。
- - 配置 TMRx_STCTRL 寄存器 STIS[2:0]位范围为 3'b000~3'b011，选择内部触发。
- - 配置 TMRx_STCTRL 寄存器 SMSEL[3:0]=4'b0111，选择外部时钟模式 A。

- 配置 TMRx_CTRL1 寄存器 TMREN 位，使能 TMRx 计数。
- 表 14-3 TMRx 内部触发连接

次定时器	IS0 (STIS = 000)	IS1 (STIS = 001)	IS2 (STIS = 010)	IS3 (STIS = 011)
TMR3	TMR1		TMR9	TMR4
TMR4	TMR1		TMR3	TMR9

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 至 TMR4 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、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。

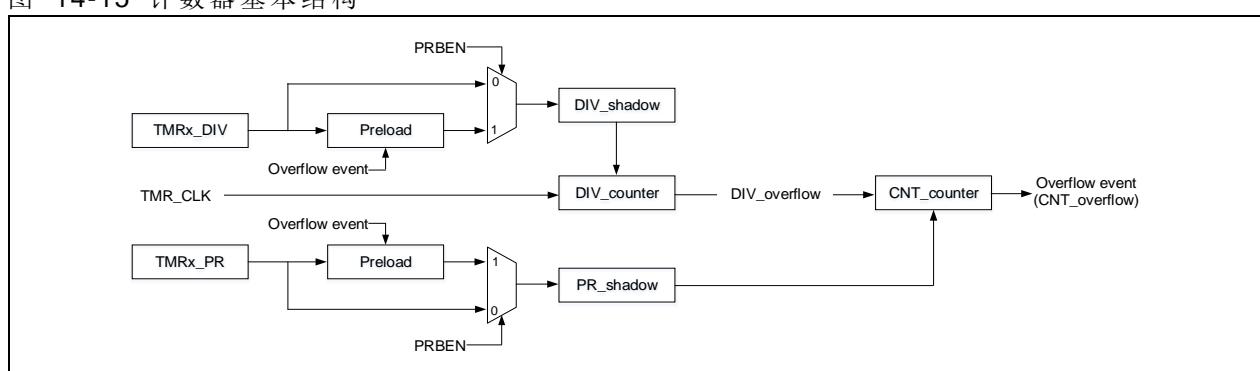
TMRx_DIV 寄存器用于设置计数器计数频率，每（DIV[15:0]+1）个计数时钟周期，计数器计数一次。和 TMRx_PR 寄存器类似，开启周期缓冲功能后，TMRx_DIV 寄存器值在溢出事件时更新至它的影子寄存器。

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 OVFEN=1 将禁止溢出事件产生。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

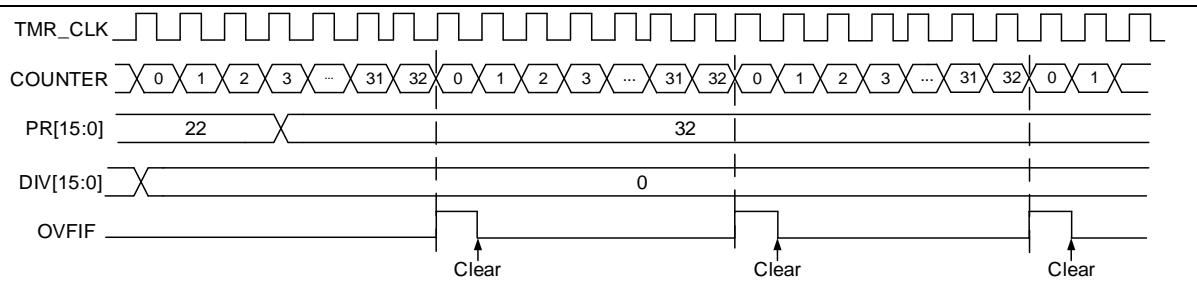
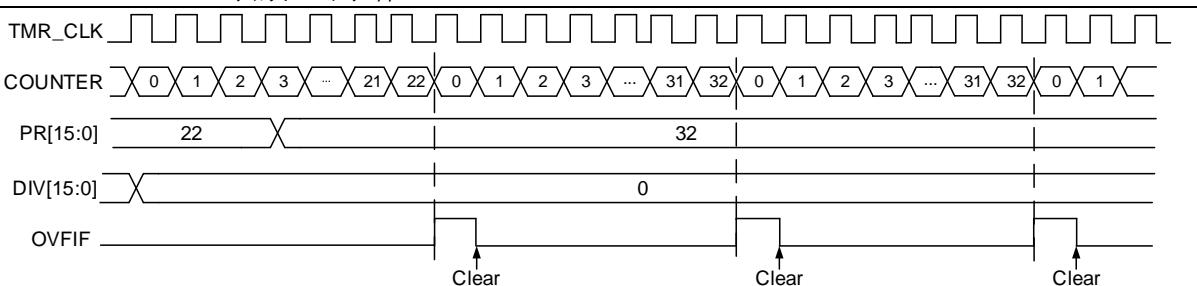


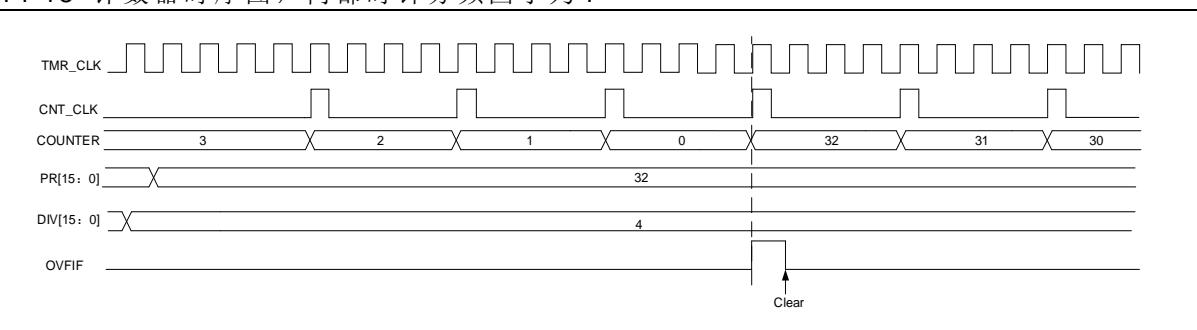
图 14-17 PRBEN=1时的溢出事件



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



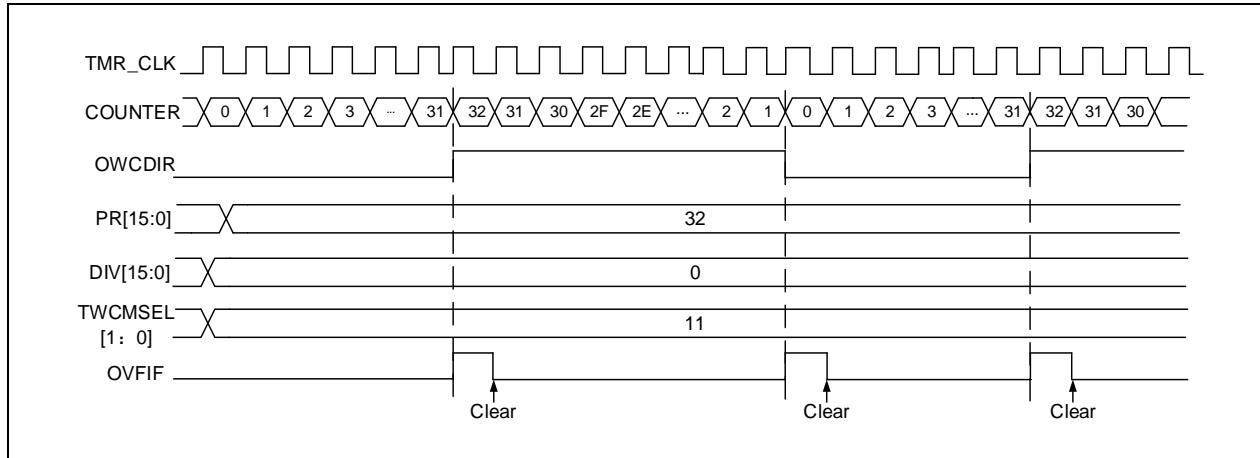
中央双向对齐计数模式

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。

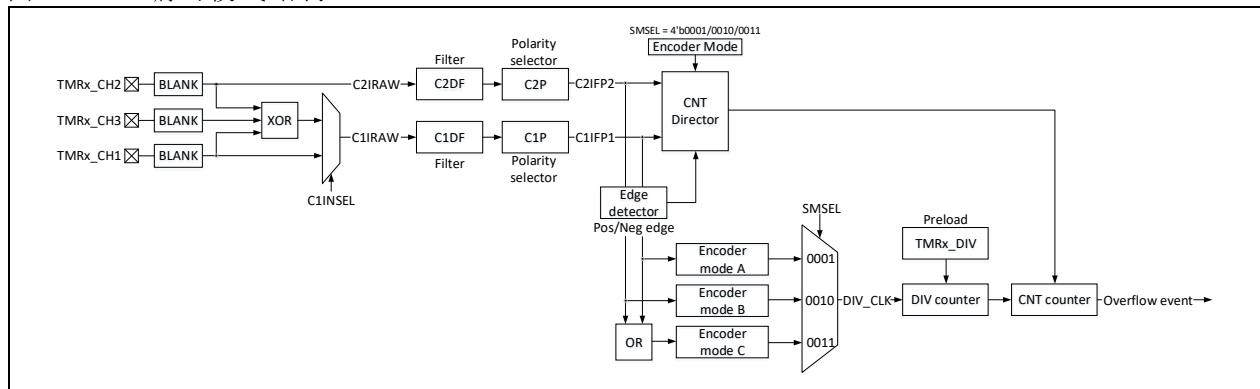
图 14-19 计数器时序图，内部时钟分频因子为1，TMRx_PR=0x32



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=4'b0001，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=4'b0010，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=4'b0011，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：

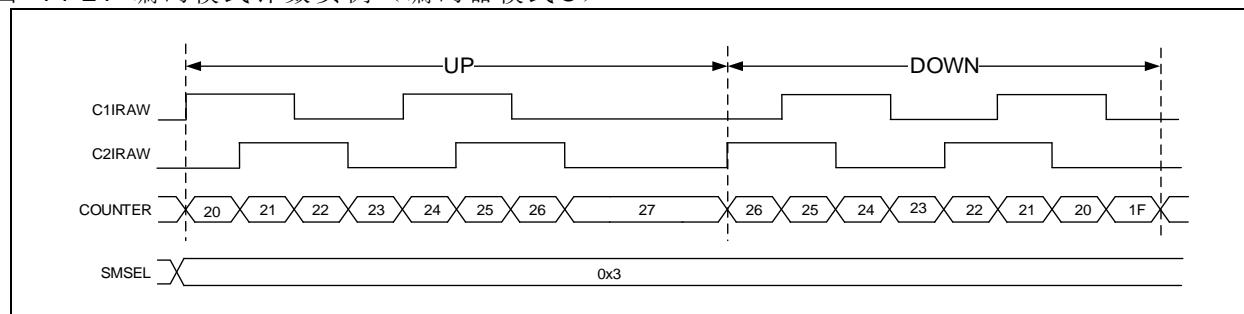
- 配置 TMRx_CM1 寄存器 C1DF[3:0]，设置通道 1 输入信号滤波；配置 TMRx_CCTRL 寄存器 C1P，设置通道 1 输入信号有效电平。
- 配置 TMRx_CM1 寄存器 C2DF[3:0]，设置通道 2 输入信号滤波；配置 TMRx_CCTRL 寄存器 C2P，设置通道 2 输入信号有效电平。
- 配置 TMRx_CM1 寄存器 C1C[1:0]，设置通道 1 为输入模式；配置 TMRx_CM1 寄存器 C2C[1:0]，设置通道 2 为输入模式；
- 配置 TMRx_STCTRL 寄存器 SMSEL[3:0]，选择编码器模式 A (SMSEL=4'b0001)、编码器模式 B (SMSEL=4'b0010) 或编码器模式 C (SMSEL=4'b0011)。
- 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 OVFIF位重映射

TMRx_CTRL1 寄存器中 OVFIFRMP 位强制将溢出中断标记 OVFIF 复制到定时器计数值寄存器 TMRx_CVAL[31]中。这样便可读取计数值的同时读取 OVFIF 来判断溢出中断标记是否置起，有利于简化计算。

14.2.3.4 TMR输入部分

TMR3 至 4 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[15:0]经过 CHxSEL 选择、输入消隐和预处理输出 CxIRAW。配置 C1INSEL 位，选择 C1IRAW 来源是 TMRx_CH1[x]或是 TMRx_CH1[x]、TMRx_CH2[x]、TMRx_CH3[x]异或。C2IRAW、C3IRAW、C4IRAW 来源是 TMRx_CH2[x]、TMRx_CH3[x]、TMRx_CH4[x]。

-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、CyIFPx、STCI。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）；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

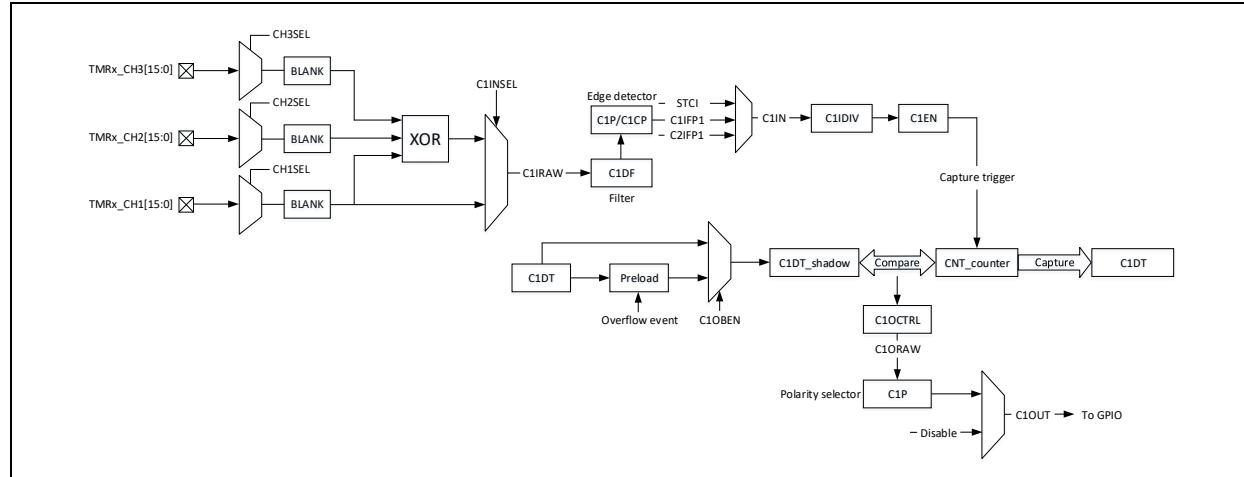
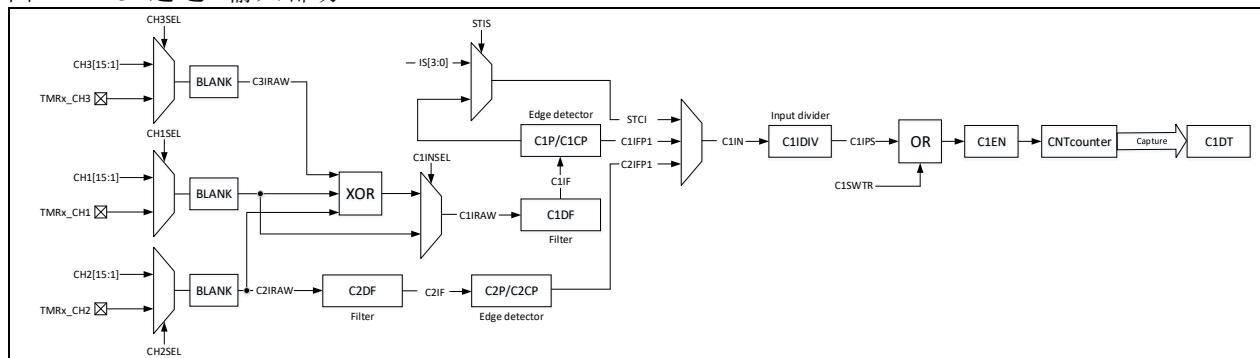


图 14-23 通道 1 输入部分



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（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。

多输入异或

通道 1 的输入端可选择 TMRx_CH1[x]、TMRx_CH2[x] 和 TMRx_CH3[x] 经异或逻辑后输入。将 TMRx_CTRL2 寄存器中的 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。

输入消隐

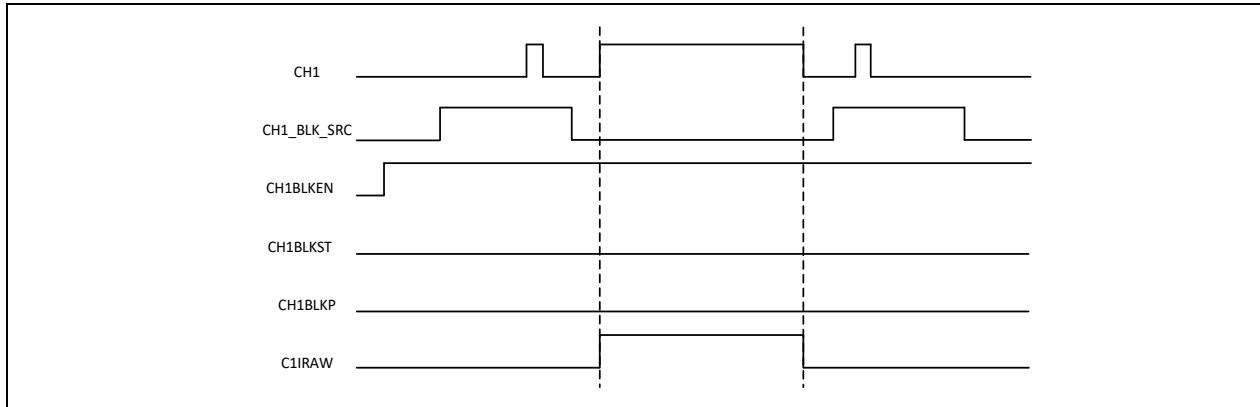
4 通道输入端都有独立的消隐器，可选择将输入信号消隐后再做捕获。将 TMRx_BLKCTRL 寄存器中 CHxBLKEN 位置 1 可开启此功能，CHxBLKSS 选择输入信号的消隐源 CHx_BLK_SRC[x]，通过控制 CHxBLKST 可将输入信号消隐为指定电平，CHxBLKP 可选择消隐有效电平，默认高电平有效。

若要开启通道 1 的输入消隐功能，可按如下进行配置：

- 配置 CH1BLKSS 选择通道 1 输入信号消隐源；
- 选择输入消隐源极性 CH1BLKP；
- 配置输入信号消隐状态 CH1BLKST；
- 使能通道 1 输入消隐功能。

若需要对输入消隐后的信号做输入捕获，如上配置完成后，请再配置输入捕获功能。

图 14-24 输入消隐功能图示



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[x]，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=4'b0100，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 14-25 PWM 输入模式配置实例

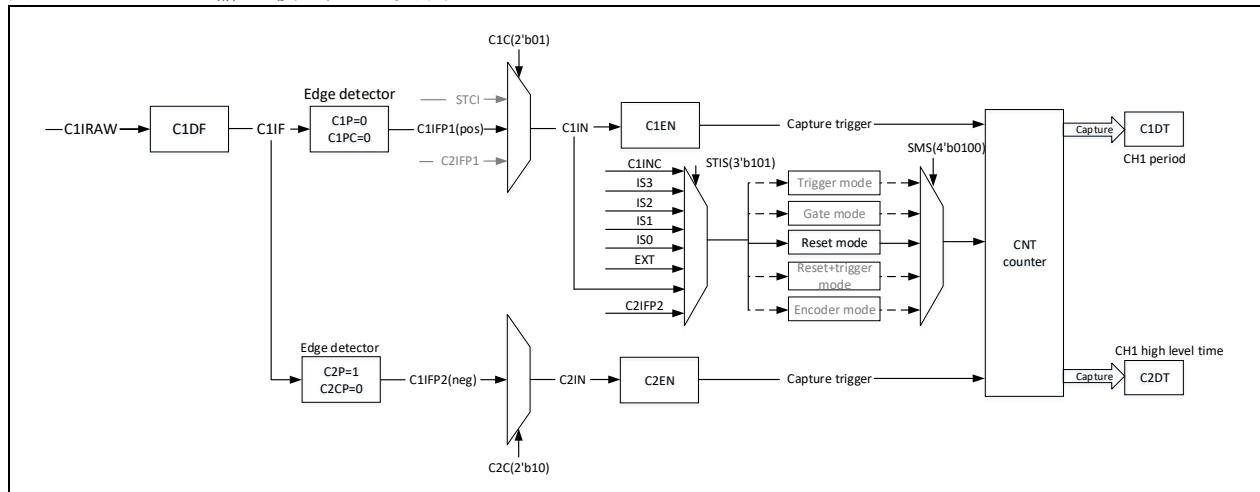
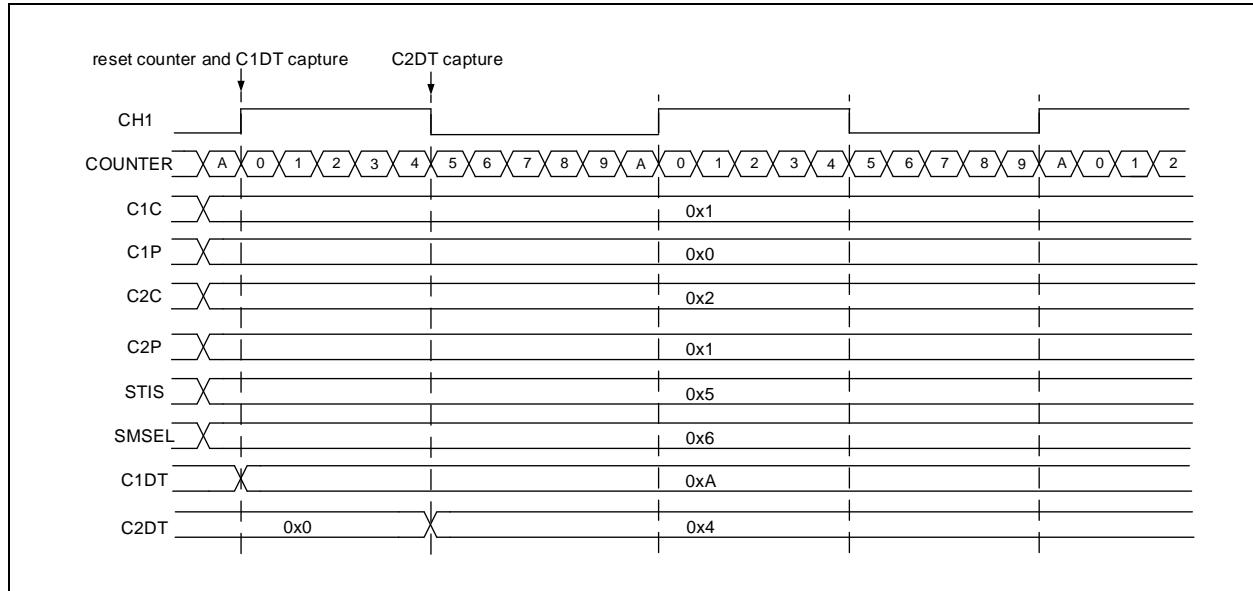


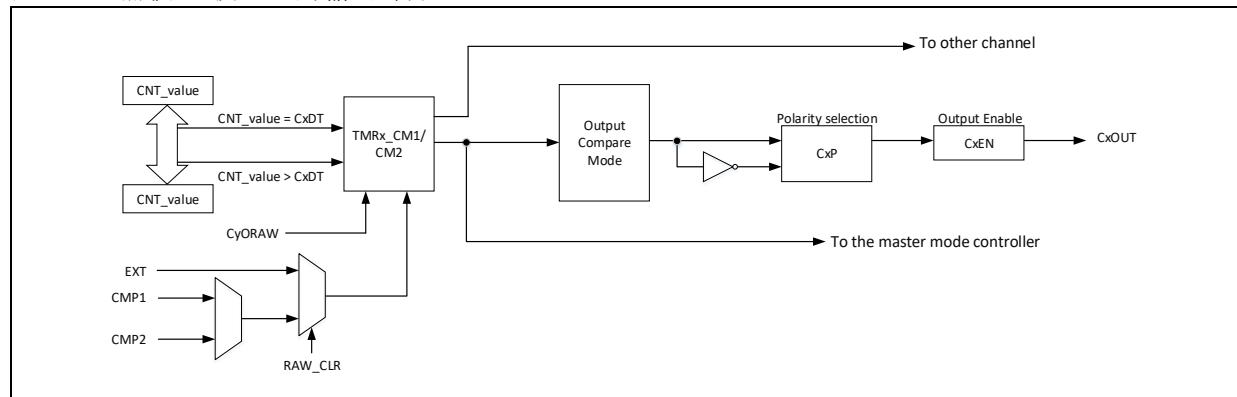
图 14-26 PWM输入模式



14.2.3.5 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 14-27 捕获/比较通道的输出部分（通道 1 至 4）



输出模式

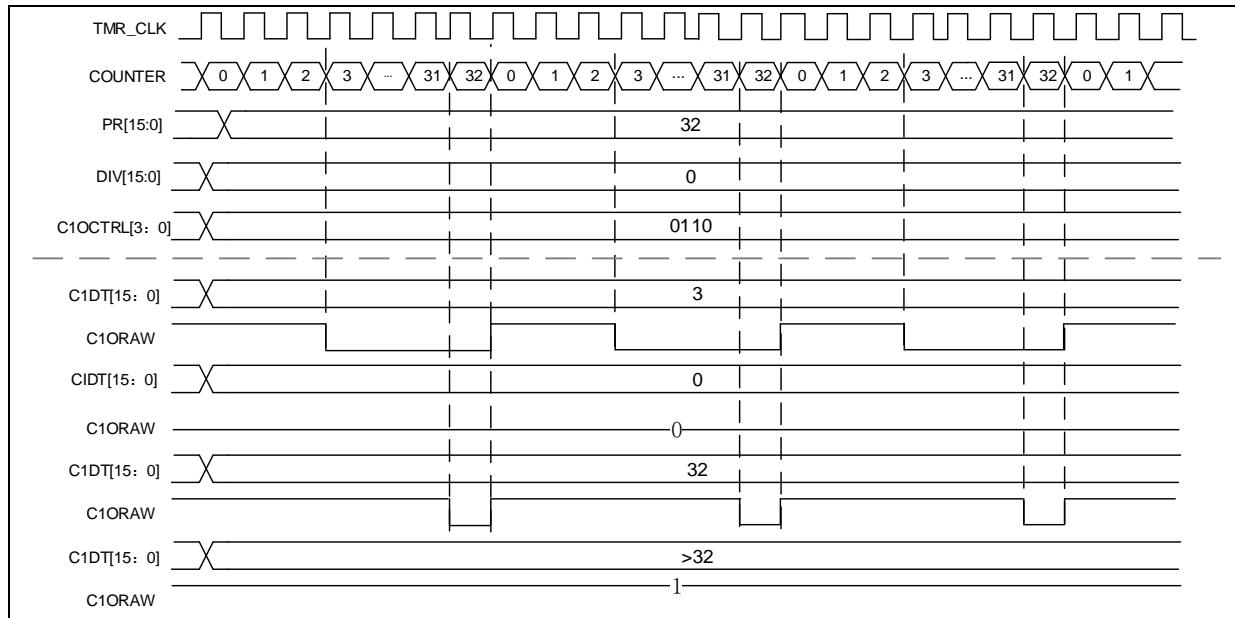
配置 $Cx\text{C}[1:0] \neq 2'b00$ 将通道配置为输出可实现多种输出模式，此时，计数器计数值将与 $Cx\text{DT}$ 寄存器值比较，并根据 $Cx\text{OCTRL}[3:0]$ 位配置的输出模式，产生中间信号 $Cx\text{ORAW}$ ，再经过输出控制逻辑处理后输送到 IO。输出信号的周期由 TMRx_PR 寄存器值配置，占空比则由 $Cx\text{DT}$ 寄存器值配置。

输出比较模式有以下子类：

PWM 模式 A: $Cx\text{OCTRL}=4'b0110$ 时，开启 PWM 模式 A。向上计数时， $\text{TMRx_C1DT} > \text{TMRx_CVAL}$ 时 $C1\text{ORAW}$ 输出高电平，否则为低电平；向下计数时， $\text{TMRx_C1DT} < \text{TMRx_CVAL}$ 时 $C1\text{ORAW}$ 输出低电平，否则为高电平。图 14-28 展示了计数器向上计数与 PWM 模式 A 配合的例子， $\text{PR}=0x32$ ， $Cx\text{DT}$ 配置为不同的值时输出信号的翻转情况。若要使用 PWM 模式 A，可按如下方式配置。

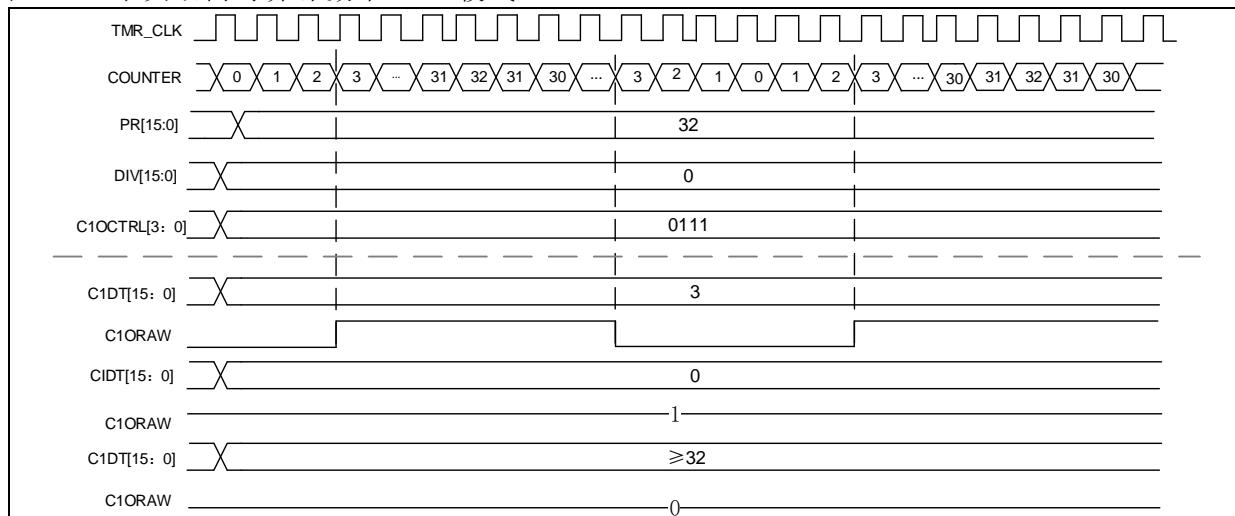
- 配置 TMRx_PR 寄存器，设置 PWM 周期。
- 配置 TMRx_CxDT 寄存器，设置 PWM 占空比。
- 配置 TMRx_CM1/CM2 寄存器 $Cx\text{OCTRL}$ 位为 $4'b0110$ ，设置输出模式为 PWM 模式 A。
- 配置 TMRx_DIV 寄存器，设置计数器计数频率。
- 配置 TMRx_CTRL1 寄存器 $TWCMSEL[1:0]$ 位，设置计数器计数模式。
- 配置 TMRx_CCTRL 寄存器 $Cx\text{P}$ 位、 $Cx\text{CP}$ 位，设置输出极性。
- 配置 TMRx_CCTRL 寄存器 $Cx\text{EN}$ 位、 $Cx\text{CEN}$ 位，使能通道输出。
- 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。

图 14-28 向上计数下 PWM 模式 A



PWM 模式 B: CxOCTRL=4'b0111 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。图 14-29 展示了计数器中央双向对齐计数与 PWM 模式 B 配合的例子，PR=0x32，CxDT 配置为不同的值时输出信号的翻转情况。

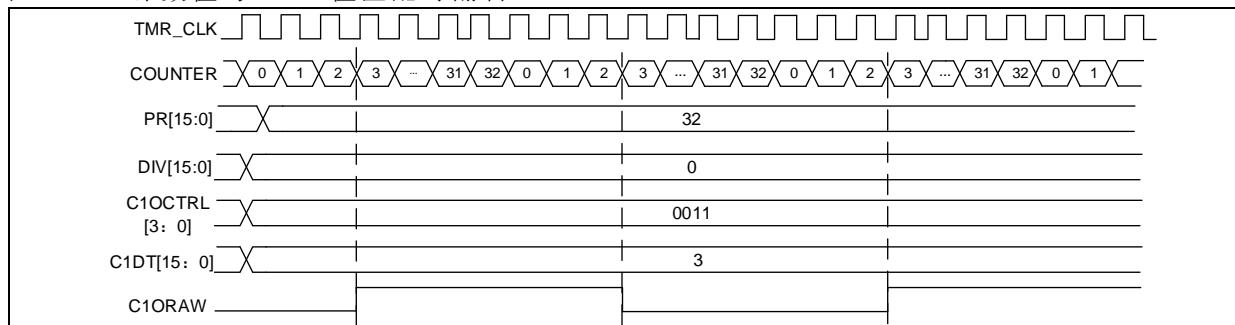
图 14-29 中央双向对齐计数下 PWM 模式 B



强制输出模式: CxOCTRL=4'b0100/0101 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。

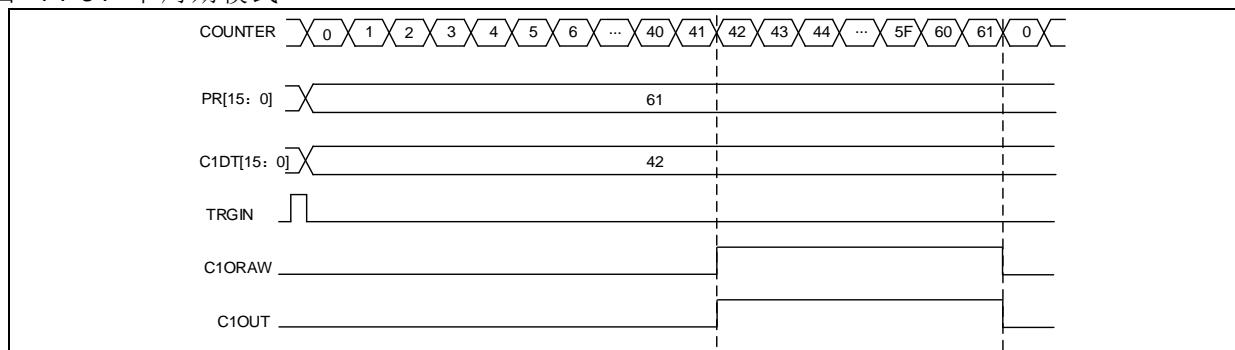
输出比较模式: CxOCTRL=4'b0001/0010/0011 时，开启输出比较模式。此时，当计数值与 CxDT 值匹配时，CxORAW 强制输出高电平（CxOCTRL=4'b0001）、低电平（CxOCTRL=4'b0010）或进行电平翻转（CxOCTRL=4'b0011）。图 14-30 展示了输出比较模式（翻转）的例子，C1DT=0x3，当计数值等于 0x3 时，输出电平 C1OUT 被翻转。

图 14-30 计数值与 C1DT 值匹配时翻转 C1ORAW



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 $CVAL > CxDT$ 。图 14-31 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 14-31 单周期模式

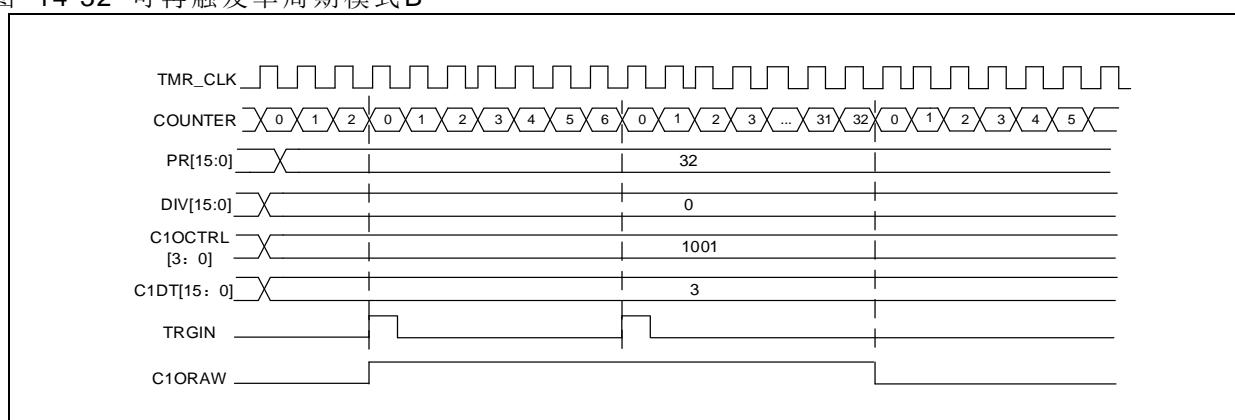


可再触发单周期模式 A: $CxOCTRL=4'b1000$ 时，开启可再触发单周期模式 A。向上计数时，检测到 TRGIN 前，C1ORAW 为高，检测到 TRGIN 后，CxORAW 以 PWM 模式 A 运行，下一次溢出事件时，CxORAW 再次为高；向下计数时，检测到 TRGIN 前，C1ORAW 为低，检测到 TRGIN 后，CxORAW 以 PWM 模式 A 运行，下一次溢出事件时，CxORAW 再次为低；检测到触发信号时，输出脉冲，若在计数周期结束前再次检测到触发信号，脉冲将延长。需搭配组合复位+触发模式使用，且不支持中央双向对齐计数模式。

可再触发单周期模式 B: $CxOCTRL=4'b1001$ 时，开启可再触发单周期模式 B。向上计数时，检测到 TRGIN 前，C1ORAW 为低，检测到 TRGIN 后，CxORAW 以 PWM 模式 B 运行，下一次溢出事件时，CxORAW 再次为低；向下计数时，检测到 TRGIN 前，C1ORAW 为高，检测到 TRGIN 后，CxORAW 以 PWM 模式 B 运行，下一次溢出事件时，CxORAW 再次为高；检测到触发信号时，输出脉冲，若在计数周期结束前再次检测到触发信号，脉冲将延长。需搭配组合复位+触发模式使用，且不支持中央双向对齐计数模式。

图 14-32 展示了可再触发 PWM 模式 B 的例子。

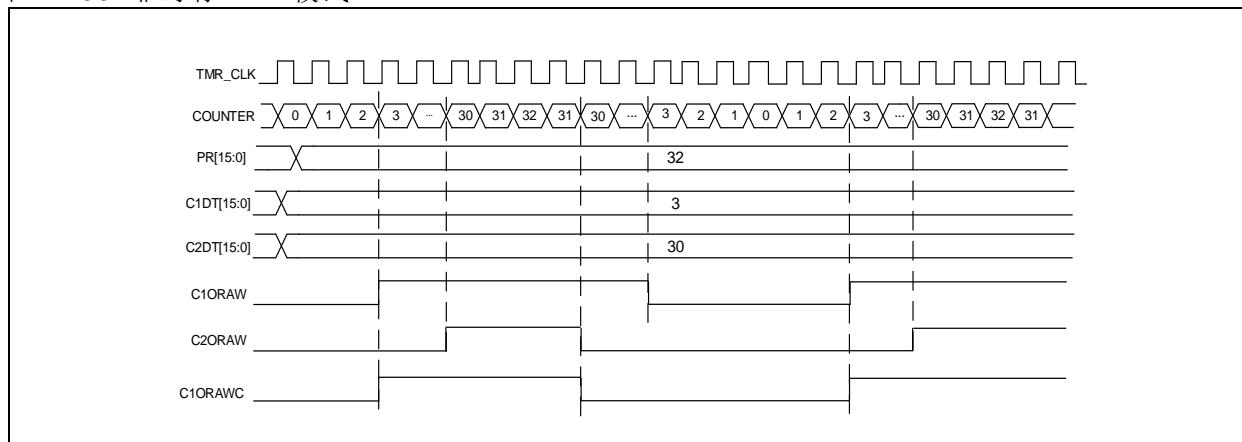
图 14-32 可再触发单周期模式 B



非对称 PWM 模式 A: CxOCTRL=4'b1110 时, 开启非对称 PWM 模式 A。CxORAW 和 CyORAW 以 PWM 模式 A 运行, 向上计数时, CxORAWC 输出 CxORAW; 向下计数时, CxORAWC 输出 CyORAW (对于通道 1, y=2; 对于通道 2, y=1; 对于通道 3, y=4; 对于通道 4, y=3)。若此时将 IASYMPEN 位使能, 将开启独立非对称 PWM 模式 A。向上计数时, 输出为计数器与 CxDT 比较产生; 向下计数时, 输出为计数器与 CxDAT 比较产生。

非对称 PWM 模式 B: CxOCTRL=4'b1111 时, 开启非对称 PWM 模式 B。CxORAW 和 CyORAW 以 PWM 模式 B 运行, 向上计数时, CxORAWC 输出 CxORAW; 向下计数时, CxORAWC 输出 CyORAW (对于通道 1, y=2; 对于通道 2, y=1; 对于通道 3, y=4; 对于通道 4, y=3)。若此时将 IASYMPEN 位使能, 将开启独立非对称 PWM 模式 B。向上计数时, 输出为计数器与 CxDT 比较产生; 向下计数时, 输出为计数器与 CxDAT 比较产生。图 14-33 展示了非对称 PWM 模式 B 的例子。

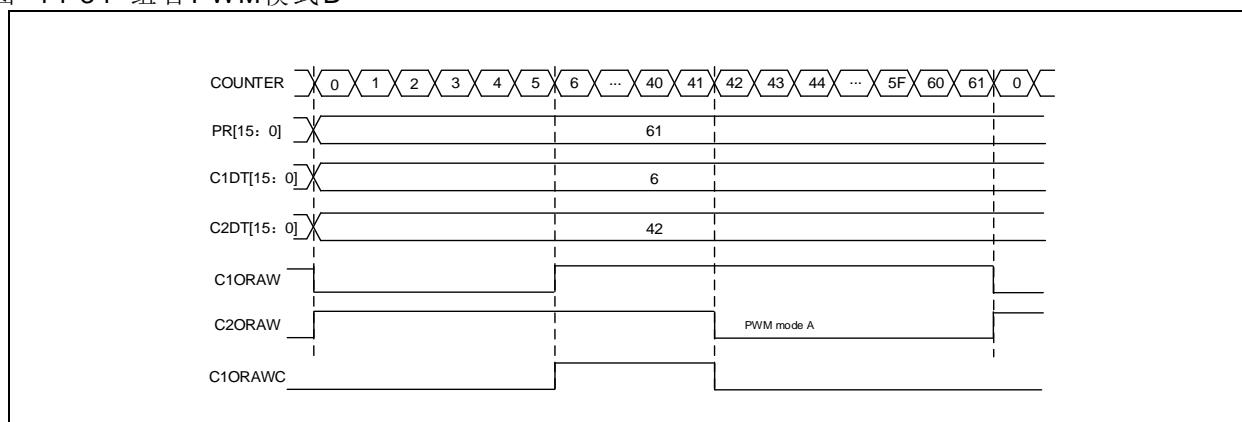
图 14-33 非对称 PWM 模式 B



组合 PWM 模式 A: CxOCTRL=4'b1100 时, 开启组合 PWM 模式 A。CxORAW 以 PWM 模式 A 运行, CyORAW 运行模式由通道 y 配置决定, CxORAWC 是 CxORAW 和 CyORAW 逻辑或运算后的输出 (对于通道 1, y=2; 对于通道 2, y=1; 对于通道 3, y=4; 对于通道 4, y=3)。

组合 PWM 模式 B: CxOCTRL=4'b1101 时, 开启组合 PWM 模式 B。CxORAW 以 PWM 模式 B 运行, CyORAW 运行模式由通道 y 配置决定, CxORAWC 是 CxORAW 和 CyORAW 逻辑与运算后的输出 (对于通道 1, y=2; 对于通道 2, y=1; 对于通道 3, y=4; 对于通道 4, y=3)。图 14-34 展示了组合 PWM 模式 B 的例子。

图 14-34 组合 PWM 模式 B



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。

注: 非对称 PWM 模式 A/B、组合 PWM 模式 A/B 须在中央对齐计数模式下工作。

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。

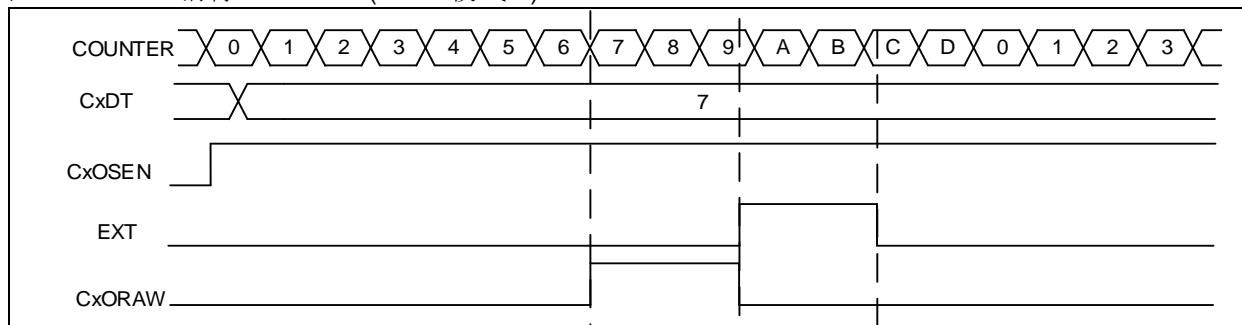
- PTOS=3'b000, TRGOUT输出软件溢出事件（TMRx_SWEVT寄存器OVFSWTR位）或复位事件。
- PTOS=3'b001, TRGOUT输出计数器使能信号。
- PTOS=3'b010, TRGOUT输出计数器溢出事件。
- PTOS=3'b011, TRGOUT输出捕获、比较事件。
- PTOS=3'b100, TRGOUT输出C1ORAW信号。
- PTOS=3'b101, TRGOUT输出C2ORAW信号。
- PTOS=3'b110, TRGOUT输出C3ORAW信号。
- PTOS=3'b111, TRGOUT输出C4ORAW信号。

CxORAW 信号清除

将 CxOSEN 位置 1 后，指定通道的 CxORAW 信号由 EXT 或比较器输出的高电平清 0，COSSEL 位用于选择清除信号来源是 EXT 或是比较器输出，若选择比较器输出，CORAW_OFF 位用于选择清除信号是来自比较器 1 或是比较器 2 输出。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 14-35 EXT 清除 CxORAW(PWM 模式 B)



14.2.3.6 定时器同步

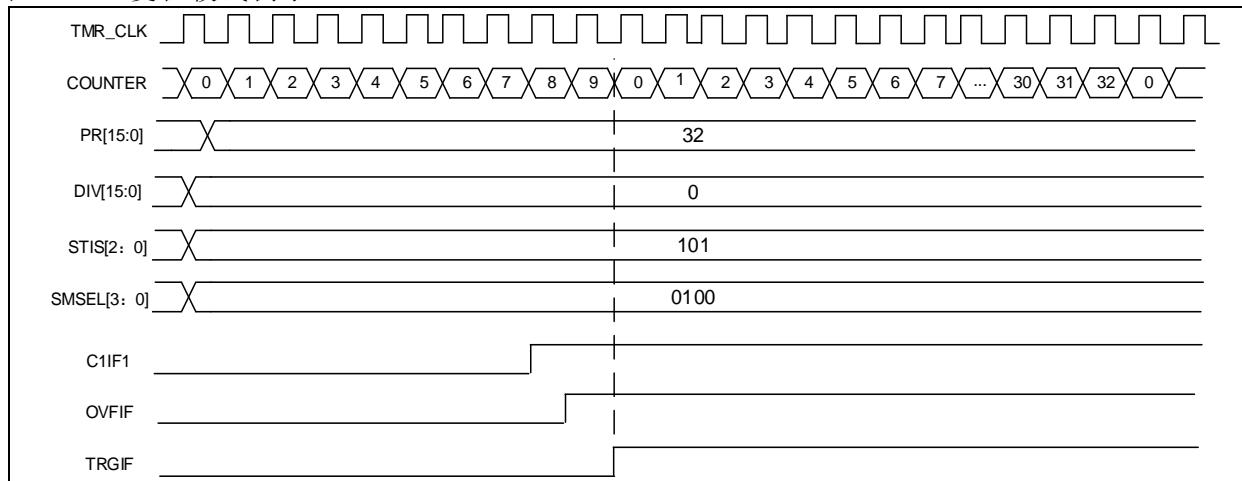
主次定时器之间可由内部连接信号进行同步。主定时器可由 PTOS[2: 0]位选择主定时器输出，即同步信息；次定时器由 SMSEL[3: 0]位选择从模式，即次定时器的工作模式。

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。

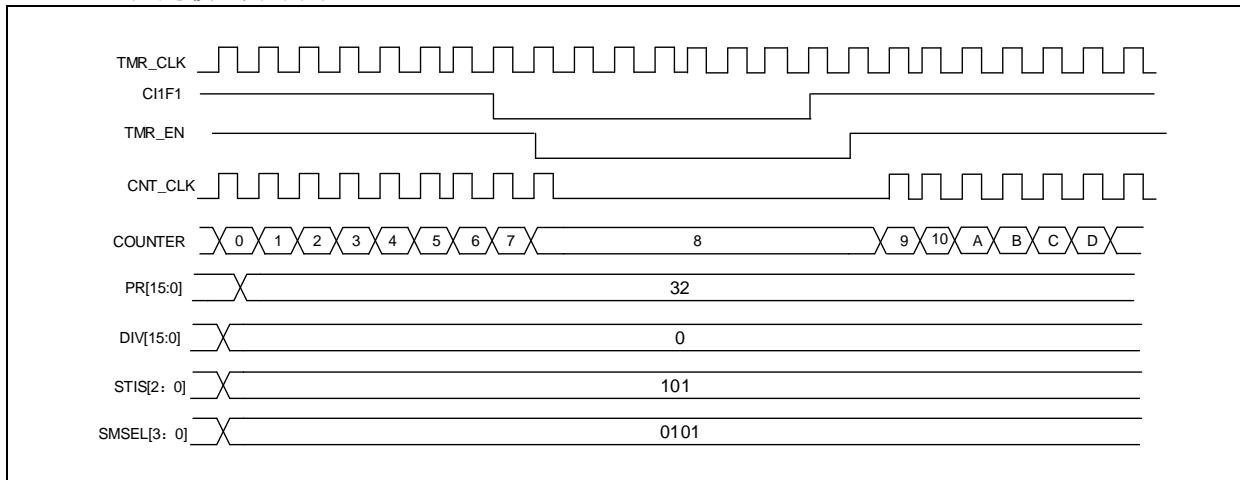
图 14-36 复位模式例子



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。

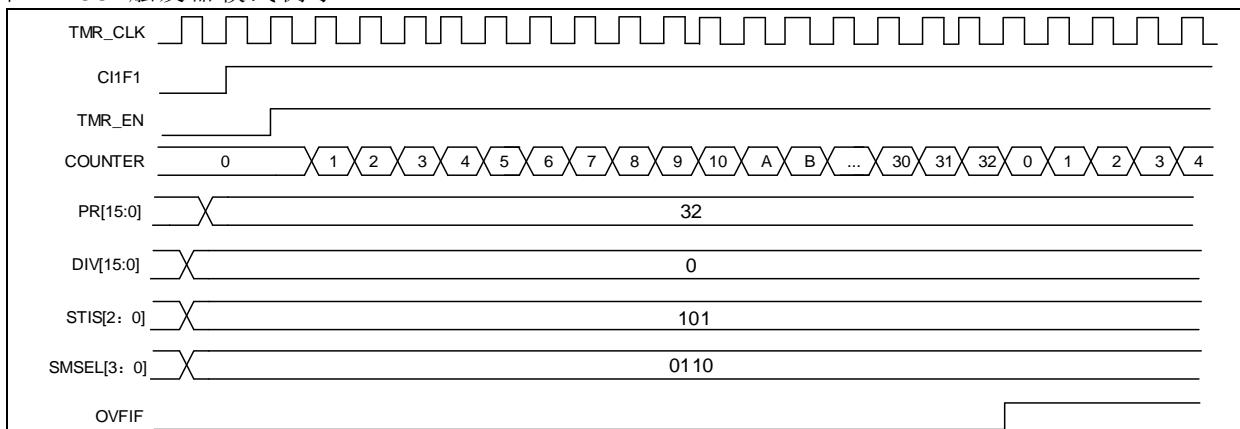
图 14-37 挂起模式下例子



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 14-38 触发器模式例子



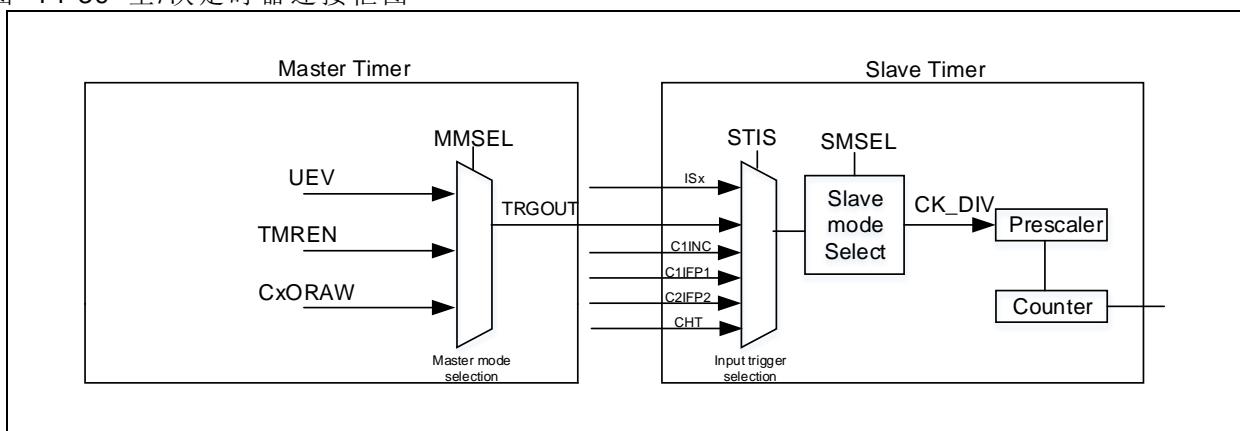
从模式：组合复位+触发模式

计数器将在选中的触发输入上升沿初始化计数器，并产生溢出事件，需搭配可再触发单周期模式使用，详见可再触发单周期模式部分描述。

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 14-39 主/次定时器连接框图



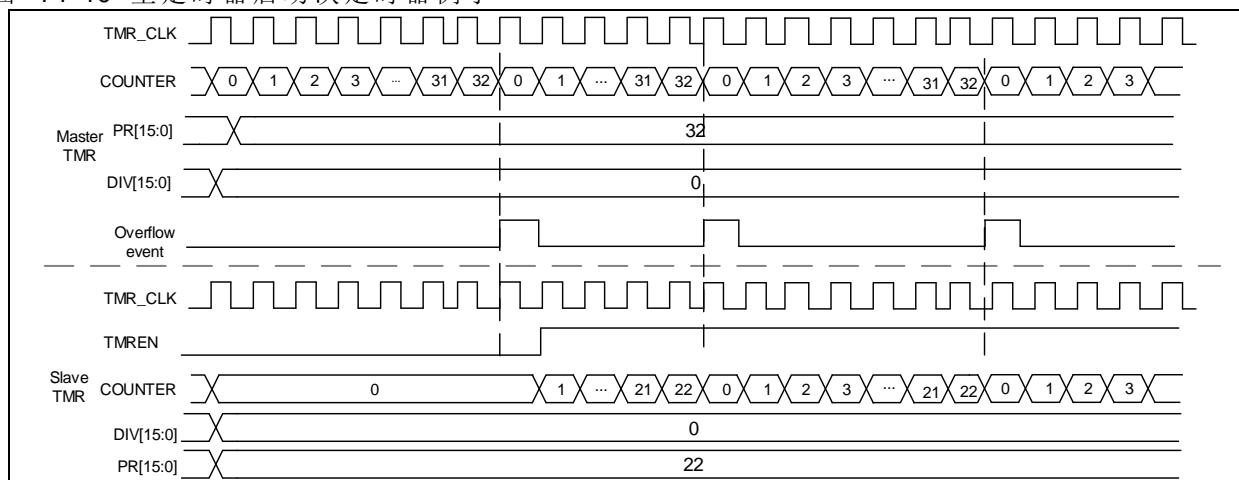
主定时器为次定时器提供时钟：

- 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（TMRx_PR 寄存器）。
- 配置次定时器触发输入信号 TRGIN 为主定时器输出（TMRx_STCTRL 寄存器的 STIS[2: 0]）。
- 配置次定时器使用外部时钟模式 A（TMRx_STCTRL 寄存器的 SMSEL[3: 0]=4'b0111）。
- 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（TMRx_PR 寄存器）。
- 配置次定时器触发输入 TRGIN 为主定时器输出。
- 置主定时器 TMREN=1 以启动主定时器。

图 14-40 主定时器启动次定时器例子

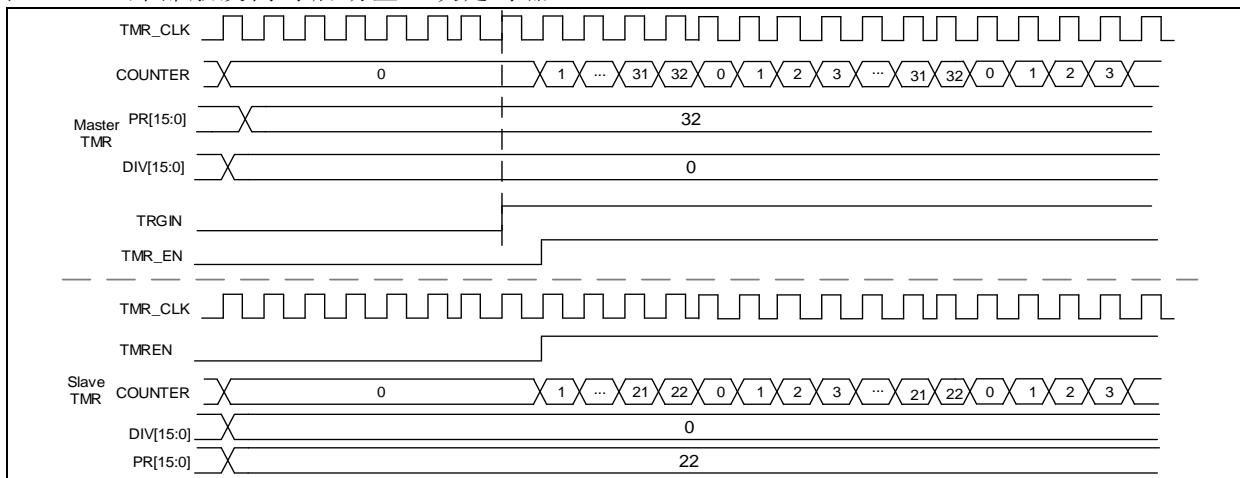


外部触发信号同步启动主、次定时器：

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=4'b0110）。

图 14-41 外部触发同时启动主、次定时器



14.2.3.7 TMR DMA

TMR 中有溢出事件 DMA 请求、触发事件 DMA 请求、霍尔事件 DMA 请求以及通道事件 DMA 请求，通过使能 **TMRx_IDEN** 寄存器中相应位，开启 DMA 请求功能，产生相应事件时触发输出 DMA 请求给到 DMA 外设。

TMR DMA Burst 功能

TMR 还支持 TMR DMA Burst 功能，通过 **TMRx_IDEN** 寄存器使能某个事件的 DMA 请求，触发 DMA 改写多个 TMR 连续寄存器。

以溢出事件触发 TMR DMA Burst 功能为例，配置如下：

- 配置 **TMRx_IDEN** 寄存器中 **OVFDEN** 位，使能溢出事件触发 DMA 请求；
- 配置 **TMRx_DMACTRL** 寄存器中 **DTB** 位设置 Burst 传输次数；
- 配置 **TMRx_DMACTRL** 寄存器中 **ADDR** 位设置 Burst 传输的起始地址；
- 使能计数器。

以上配置的 TMR DMA Burst 传输流程如下：

当发生溢出事件时，TMR 将会产生溢出事件的 DMA 请求给到 DMA，DMA 根据请求将数据写入 **TMRx_DMADT** 寄存器，在 TMR 内部则会将 DMADT 位的数据写入 Burst 传输起始地址寄存器，并发送 ACK 信号给到 TMR，TMR 接收到 ACK 信号后，将清除当前 DMA 请求；TMR 检测到 Burst 传输并未全部完成，会重新置起溢出事件 DMA 请求给到 DMA，DMA 根据请求将数据再次写入 **TMRx_DMADT** 寄存器，在 TMR 内部则会将 DMADT 寄存器的数据写入 Burst 传输起始地址+0x4 地址寄存器，并发送 ACK 信号给到 TMR；以此往复，直到 Burst 传输最后一次操作时，检测到此时 Burst 传输已全部完成，溢出事件 DMA 请求 REQ 将不会再被重新置起，直到下一次新的溢出事件到来。

注：使用 TMR DMA Burst 功能时，起始地址到结束地址这个区间，不应有空寄存器以及不应包含 **TMRx_DMACTRL** 和 **TMRx_DMADT** 寄存器。

14.2.3.8 调试模式

当微控制器进入调试模式（Cortex®-M4F 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。

14.2.4 TMR3至TMR4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 至 TMR4 的所有寄存器映射到一个 32 位可寻址（编址）空间。

表 14-5 TMR3至TMR4寄存器图和复位值

寄存器简称	基址偏移量	复位值
TMRx_CTRL1	0x00	0x0000 0000
TMRx_CTRL2	0x04	0x0000 0000
TMRx_STCTRL	0x08	0x0000 0000
TMRx_IDEN	0x0C	0x0000 0000
TMRxISTS	0x10	0x0000 0000
TMRx_SWEVT	0x14	0x0000 0000
TMRx_CM1	0x18	0x0000 0000
TMRx_CM2	0x1C	0x0000 0000
TMRx_CCTRL	0x20	0x0000 0000
TMRx_CVAL	0x24	0x0000 0000
TMRx_DIV	0x28	0x0000 0000
TMRx_PR	0x2C	0x0000 0000
TMRx_C1DT	0x34	0x0000 0000
TMRx_C2DT	0x38	0x0000 0000
TMRx_C3DT	0x3C	0x0000 0000
TMRx_C4DT	0x40	0x0000 0000
TMRx_DMACTRL	0x48	0x0000 0000
TMRx_DMADT	0x4C	0x0000 0000
TMRx_OPT1	0x50	0x0000 0000
TMRx_OPT2	0x60	0x0000 0000
TMRx_CHSEL	0x68	0x0000 0000
TMRx_BLKCTRL	0x70	0x0000 0000

14.2.4.1 TMRx控制寄存器1 (TMRx_CTRL1)

域	简称	复位值	类型	功能
位 31: 12	保留	0x000000	resd	保持默认值。
位 11	OVFIFRMP	0x0	rw	OVFIF 状态位重映射 (OVFIF status bit remapping) 0: 无重映射; 1: 使能重映射, OVFIF 位复制到 TMRx_CVAL[31]。
位 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。 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 TMRx控制寄存器2 (TMRx_CTRL2)

域	简称	复位值	类型	功能
位 31: 8	保留	0x000000	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) TMRx 输出到次定时器的信号选择: 000: 软件溢出或复位; 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 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 TMRx次定时器控制寄存器 (TMRx_STCTRL)

域	简称	复位值	类型	功能
位 31: 22	保留	0x000	resd	保持默认值。
位 19: 17	保留	0x0	resd	保持默认值。
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 2 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 trigger signal division) 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	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: CxORAW_OFF 来自 CMP1 或 CMP2, 取决于 CORAW_OFF 位; 1: CxORAW_OFF 来自 EXT。
位 16, 2: 0	SMSEL	0x0	rw	次定时器模式选择 (Subordinate TMR mode selection) 0000: 关闭从模式; 0001: 编码模式 A; 0010: 编码模式 B; 0011: 编码模式 C; 0100: 复位模式 - TRGIN 输入上升沿时, 重新初始化计数器; 0101: 挂起模式 - TRGIN 输入高电平时, 计数器计数; 0110: 触发模式 - TRGIN 输入上升沿时, 产生触发事件; 0111: 外部时钟模式 A - TRGIN 输入上升沿提供时钟; 1000: 组合复位+触发模式 - TRGIN 输入上升沿时, 重新初始化计数器, 产生溢出事件并启动计数器; 其他值: 保留。 注: 编码模式 A/B/C 配置方法请查看计数模式模式章节。

14.2.4.4 TMRx DMA/中断使能寄存器 (TMRx_IDEN)

域	简称	复位值	类型	功能
位 31: 15	保留	0x00000	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。 通道 1 中断使能 (Channel 1 interrupt enable)
位 1	C1IEN	0x0	rw	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 TMRx 中断状态寄存器 (TMRx_STS)

域	简称	复位值	类型	功能
位 31: 13	保留	0x00000	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。 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	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 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。

14.2.4.6 TMRx软件事件寄存器 (TMRx_SW_EVT)

域	简称	复位值	类型	功能
位 31: 7	保留	0x00000000	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 TMRx通道模式寄存器1 (TMRx_CM1)

输出比较模式:

域	简称	复位值	类型	功能
位 31: 25	保留	0x0	resd	保持默认值。
位 23: 17	保留	0x0	resd	保持默认值。
位 15	C2OSEN	0x0	rw	通道 2 输出开关使能 (Channel 2 output switch enable)
位 24, 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 16, 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。 0000: 断开。断开 C1ORAW 到 C1OUT 的输出; 0001: 当 TMRx_CVAL=TMRx_C1DT 时, 设置 C1ORAW 为高。 0010: 当 TMRx_CVAL=TMRx_C1DT 时, 设置 C1ORAW 为低。 0011: 当 TMRx_CVAL=TMRx_C1DT 时, 切换 C1ORAW 的电平。 0100: 固定 C1ORAW 为低;

0101: 固定 C1ORAW 为高;
 0110: 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。
 0111: 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。
 1000: 可再触发单周期模式 A;
 一向上计数时, 检测到 TRGIN 前, C1ORAW 为高, 检测到 TRGIN 后, C1ORAW 以 PWM 模式 A 运行, 下一次溢出事件时, C1ORAW 再次为高;
 一向下计数时, 检测到 TRGIN 前, C1ORAW 为低, 检测到 TRGIN 后, C1ORAW 以 PWM 模式 A 运行, 下一次溢出事件时, C1ORAW 再次为低。
 1001: 可再触发单周期模式 B;
 一向上计数时, 检测到 TRGIN 前, C1ORAW 为低, 检测到 TRGIN 后, C1ORAW 以 PWM 模式 B 运行, 下一次溢出事件时, C1ORAW 再次为低;
 一向下计数时, 检测到 TRGIN 前, C1ORAW 为高, 检测到 TRGIN 后, C1ORAW 以 PWM 模式 B 运行, 下一次溢出事件时, C1ORAW 再次为高。
 1010: 保留;
 1011: 保留;
 1100: 组合 PWM 模式 A, C1ORAW 以 PWM 模式 A 运行, C1ORAWC 是 C1ORAW 和 C2ORAW 逻辑或运算后的输出;
 1101: 组合 PWM 模式 B, C1ORAW 以 PWM 模式 B 运行, C1ORAWC 是 C1ORAW 和 C2ORAW 逻辑与运算后的输出;
 1110: 非对称 PWM 模式 A, C1ORAW 以 PWM 模式 A 运行, 向上计数时, C1ORAWC 输出 C1ORAW; 向下计数时, C1ORAWC 输出 C2ORAW;
 1111: 非对称 PWM 模式 B, C1ORAW 以 PWM 模式 B 运行, 向上计数时, C1ORAWC 输出 C1ORAW; 向下计数时, C1ORAWC 输出 C2ORAW。
 注: 除'0000'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: 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。

输入模式:

域	简称	复位值	类型	功能
位 31: 16	保留	0x0	resd	保持默认值。
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)
				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 9: 8	C2C	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; 0001: 采样频率 $f_{SAMPLING} = f_{CK_INT}$, N=2; 0010: 采样频率 $f_{SAMPLING} = f_{CK_INT}$, N=4; 0011: 采样频率 $f_{SAMPLING} = f_{CK_INT}$, N=8; 0100: 采样频率 $f_{SAMPLING} = f_{DTS}/2$, N=6; 0101: 采样频率 $f_{SAMPLING} = f_{DTS}/2$, N=8; 0110: 采样频率 $f_{SAMPLING} = f_{DTS}/4$, N=6; 0111: 采样频率 $f_{SAMPLING} = f_{DTS}/4$, N=8; 1000: 采样频率 $f_{SAMPLING} = f_{DTS}/8$, N=6; 1001: 采样频率 $f_{SAMPLING} = f_{DTS}/8$, N=8; 1010: 采样频率 $f_{SAMPLING} = f_{DTS}/16$, N=5; 1011: 采样频率 $f_{SAMPLING} = f_{DTS}/16$, N=6; 1100: 采样频率 $f_{SAMPLING} = f_{DTS}/16$, N=8; 1101: 采样频率 $f_{SAMPLING} = f_{DTS}/32$, N=5; 1110: 采样频率 $f_{SAMPLING} = f_{DTS}/32$, N=6; 1111: 采样频率 $f_{SAMPLING} = f_{DTS}/32$, N=8。
位 7: 4	C1DF	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 3: 2	C1IDIV	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。
位 1: 0	C1C	0x0	rw	

14.2.4.8 TMRx通道模式寄存器2 (TMRx_CM2)

输出比较模式:

域	简称	复位值	类型	功能
位 31: 25	保留	0x0	resd	保持默认值。
位 23: 17	保留	0x0	resd	保持默认值。
位 15	C4OSEN	0x0	rw	通道 4 输出开关使能 (Channel 4 output switch enable)
位 24, 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 16, 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：				
位 1: 0	C3C	0x0	rw	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 31: 16	保留	0x0	resd	保持默认值。
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)
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: 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)
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：				
位 1: 0	C3C	0x0	rw	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 TMRx通道控制寄存器 (TMRx_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。 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 1	C1P	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。
位 0	C1EN	0x0	rw	

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)
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。

14.2.4.10 TMRx计数值寄存器 (TMRx_CVAL)

域	简称	复位值	类型	功能
位 31	OVFIFCPY	0x0	ro	OVFIF 副本 (OVFIF copy) OVFIFRMP 为 0 时, 该位保留; OVFIFRMP 为 1 时, 该位为 OVFIF 位的只读副本。
位 30: 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 TMRx分频系数寄存器 (TMRx_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 TMRx周期寄存器 (TMRx_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 TMRx通道1数据寄存器 (TMRx_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 TMRx通道2数据寄存器 (TMRx_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 TMRx通道3数据寄存器 (TMRx_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 TMRx通道4数据寄存器 (TMRx_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 TMRx DMA控制寄存器 (TMRx_DMACTRL)

域	简称	复位值	类型	功能
位 31: 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 TMRx_CTRL1 所在地址开始的偏移量： 00000: TMRx_CTRL1, 00001: TMRx_CTRL2, 00010: TMRx_STCTRL,

14.2.4.18 TMRx DMA数据寄存器 (TMRx_DMADT)

域	简称	复位值	类型	功能
位 31: 16	保留	0x00	resd	保持默认值。
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： TMRx 外设地址 + ADDR*4 至 TMRx 外设地址 + ADDR*4 + DTB*4。

14.2.4.19 TMRx 选项寄存器1 (TMRx_OPT1)

域	简称	复位值	类型	功能
位 31: 1	保留	0x0	resd	保持默认值。
位 0	CORAW_OFF	0x0	rw	CORAW_OFF 源选择 (CORAW_OFF source selection) 0: CMP1; 1: CMP2。

14.2.4.20 TMR3 选项寄存器2 (TMR3_OPT2)

域	简称	复位值	类型	功能
位 31: 18	保留	0x00	resd	保持默认值。
位 17: 14	EXTSEL	0x0	rw	EXT 选择源 (EXT source selection) 0000: GPIO; 0001: CMP1; 0010: CMP2; 1001: TMR4_EXT; 其他: 保留。
位 13: 0		0x00	resd	保持默认值。

14.2.4.21 TMR4 选项寄存器2 (TMR4_OPT2)

域	简称	复位值	类型	功能
位 31: 18	保留	0x00	resd	保持默认值。
位 17: 14	EXTSEL	0x0	rw	EXT 选择源 (EXT source selection) 0000: GPIO; 0001: CMP1; 0010: CMP2; 1000: TMR3_EXT; 其他: 保留。
位 13: 0		0x00	resd	保持默认值。

14.2.4.22 TMR3 通道选择寄存器 (TMR3_CHSEL)

域	简称	复位值	类型	功能
位 31: 28	保留	0x00	resd	保持默认值。
位 27: 24	CH4SEL	0x0	rw	CH4 输入选择 (CH4 input selection) 0000: TMR3_CH4; 0001: CMP2; 其他: 保留。
位 23: 20	保留	0x0	resd	保持默认值。
位 19: 16	CH3SEL	0x0	rw	CH3 输入选择 (CH3 input selection) 0000: TMR3_CH3; 0001: CMP1; 0010: TMR4_CH3; 其他: 保留。
位 15: 12	保留	0x0	resd	保持默认值。
位 11: 8	CH2SEL	0x0	rw	CH2 输入选择 (CH2 input selection) 0000: TMR3_CH2; 0001: CMP1; 0010: CMP2; 1000: TMR4_CH2; 其他: 保留。
位 7: 4	保留	0x0	resd	保持默认值。
位 3: 0	CH1SEL	0x0	rw	CH1 输入选择 (CH1 input selection) 0000: TMR3_CH1; 0001: CMP1; 0010: CMP2; 1000: TMR4_CH1; 其他: 保留。
位 13: 0		0x00	resd	保持默认值。

14.2.4.23 TMR4 通道选择寄存器 (TMR4_CHSEL)

域	简称	复位值	类型	功能
位 31: 28	保留	0x00	resd	保持默认值。
位 27: 24	CH4SEL	0x0	rw	CH4 输入选择 (CH4 input selection) 0000: TMR4_CH4; 0001: CMP1; 其他: 保留。
位 23: 20	保留	0x0	resd	保持默认值。
位 19: 16	CH3SEL	0x0	rw	CH3 输入选择 (CH3 input selection) 0000: TMR4_CH3; 0001: CMP2; 0010: TMR3_CH3; 其他: 保留。
位 15: 12	保留	0x0	resd	保持默认值。
位 11: 8	CH2SEL	0x0	rw	CH2 输入选择 (CH2 input selection) 0000: TMR4_CH2; 0001: CMP1; 0010: CMP2; 1000: TMR3_CH2; 其他: 保留。
位 7: 4	保留	0x0	resd	保持默认值。
位 3: 0	CH1SEL	0x0	rw	CH1 输入选择 (CH1 input selection) 0000: TMR4_CH1; 0001: CMP1; 0010: CMP2; 1000: TMR3_CH1; 其他: 保留。
位 13: 0		0x00	resd	保持默认值。

14.2.4.24 TMR3 定时器消隐控制寄存器 (TMR3_BLKCTRL)

域	简称	复位值	类型	功能
位 31: 20	保留	0x00	resd	保持默认值。
位 19	CH4BLKP	0x0	rw	CH4 消隐极性 (CH4 blanking polarity)
位 18	CH3BLKP	0x0	rw	CH3 消隐极性 (CH3 blanking polarity)
位 17	CH2BLKP	0x0	rw	CH2 消隐极性 (CH2 blanking polarity)
				CH1 消隐极性 (CH1 blanking polarity)
位 16	CH1BLKP	0x0	rw	0: 消隐输入源有效电平为高; 1: 消隐输入源有效电平为低。
位 15	CH4BLKEN	0x0	rw	CH4 消隐使能 (CH4 blanking enable)
位 14	CH3BLKEN	0x0	rw	CH3 消隐使能 (CH3 blanking enable)
位 13	CH2BLKEN	0x0	rw	CH2 消隐使能 (CH2 blanking enable)
				CH1 消隐使能 (CH1 blanking enable)
位 12	CH1BLKEN	0x0	rw	0: 关闭消隐功能; 1: 使能消隐功能。
位 11	CH4BLKST	0x0	rw	CH4 消隐状态 (CH4 blanking state)
位 10	CH3BLKST	0x0	rw	CH3 消隐状态 (CH3 blanking state)
位 9	CH2BLKST	0x0	rw	CH2 消隐状态 (CH2 blanking state)
				CH1 消隐状态 (CH1 blanking state)
位 8	CH1BLKST	0x0	rw	CH1 消隐使能的情况下: 0: 消隐有效时, 将输入信号消隐为 0; 1: 消隐有效时, 将输入信号消隐为 1。
				CH4 消隐源选择 (CH4 blanking sources selection) 00: TMR1_OC4; 01: TMR4_OC4; 10: TMR9_OC2; 11: TMR10_OC1.
				CH3 消隐源选择 (CH3 blanking sources selection) 00: TMR1_OC4; 01: TMR3_OC4; 10: TMR9_OC2; 11: TMR10_OC1.
				CH2 消隐源选择 (CH2 blanking sources selection) 00: TMR1_OC4; 01: TMR3_OC4; 10: TMR9_OC2; 11: TMR10_OC1.
				CH1 消隐源选择 (CH1 blanking sources selection) 00: TMR1_OC4; 01: TMR3_OC4; 10: TMR9_OC2; 11: TMR10_OC1.
位 1: 0	CH1BLKSS	0x0	rw	

14.2.4.25 TMR4 定时器消隐控制寄存器 (TMR4_BLKCTRL)

域	简称	复位值	类型	功能
位 31: 20	保留	0x00	resd	保持默认值。
位 19	CH4BLKP	0x0	rw	CH4 消隐极性 (CH4 blanking polarity)
位 18	CH3BLKP	0x0	rw	CH3 消隐极性 (CH3 blanking polarity)
位 17	CH2BLKP	0x0	rw	CH2 消隐极性 (CH2 blanking polarity)
				CH1 消隐极性 (CH1 blanking polarity)
位 16	CH1BLKP	0x0	rw	0: 消隐输入源有效电平为高; 1: 消隐输入源有效电平为低。
位 15	CH4BLKEN	0x0	rw	CH4 消隐使能 (CH4 blanking enable)
位 14	CH3BLKEN	0x0	rw	CH3 消隐使能 (CH3 blanking enable)
位 13	CH2BLKEN	0x0	rw	CH2 消隐使能 (CH2 blanking enable)
				CH1 消隐使能 (CH1 blanking enable)
位 12	CH1BLKEN	0x0	rw	0: 关闭消隐功能; 1: 使能消隐功能。
位 11	CH4BLKST	0x0	rw	CH4 消隐状态 (CH4 blanking state)

位 10	CH3BLKST	0x0	rw	CH3 消隐状态 (CH3 blanking state)
位 9	CH2BLKST	0x0	rw	CH2 消隐状态 (CH2 blanking state)
位 8	CH1BLKST	0x0	rw	CH1 消隐状态 (CH1 blanking state) CH1 消隐使能的情况下: 0: 消隐有效时, 将输入信号消隐为 0; 1: 消隐有效时, 将输入信号消隐为 1。
位 7: 6	CH4BLKSS	0x0	rw	CH4 消隐源选择 (CH4 blanking sources selection) 00: TMR1_OC4; 01: TMR3_OC4; 10: TMR9_OC2; 11: TMR11_OC1。
位 5: 4	CH3BLKSS	0x0	rw	CH3 消隐源选择 (CH3 blanking sources selection) 00: TMR1_OC4; 01: TMR4_OC4; 10: TMR9_OC2; 11: TMR11_OC1。
位 3: 2	CH2BLKSS	0x0	rw	CH2 消隐源选择 (CH2 blanking sources selection) 00: TMR1_OC4; 01: TMR4_OC4; 10: TMR9_OC2; 11: TMR11_OC1。
位 1: 0	CH1BLKSS	0x0	rw	CH1 消隐源选择 (CH1 blanking sources selection) 00: TMR1_OC4; 01: TMR4_OC4; 10: TMR9_OC2; 11: TMR11_OC1。

14.3 通用定时器（TMR9到TMR11）

14.3.1 TMR9到TMR11简介

通用定时器 TMR9 到 TMR11 包含一个支持向上、向下、中央双向对齐计数的 16 位计数器、可实现输入捕获、可编程 PWM 输出。

TMR9 包含 2 个捕获/比较寄存器、2 组独立的通道、可实现嵌入死区。

TMR10、TMR11 包含 1 个捕获/比较寄存器、1 组独立的通道、可实现嵌入死区。

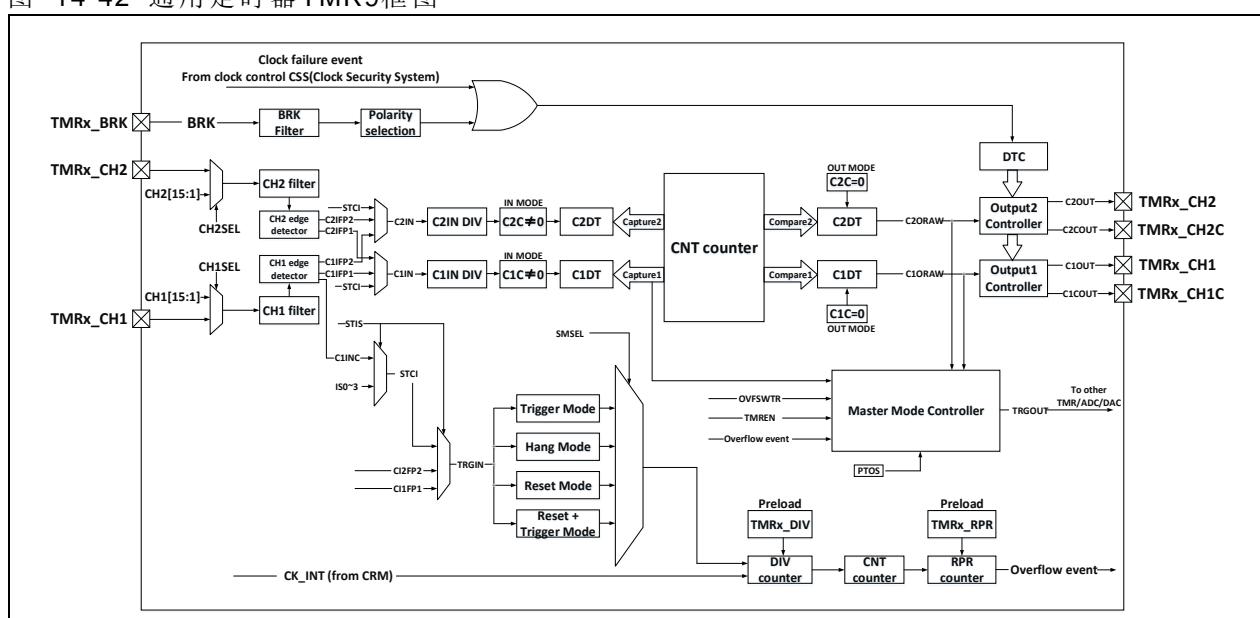
14.3.2 TMR9到TMR11主要功能

14.3.2.1 TMR9主要功能

TMR9 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、向下、双向计数器、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
- PWM模式支持组合模式
- 2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 DMA Burst传输

图 14-42 通用定时器 TMR9 框图

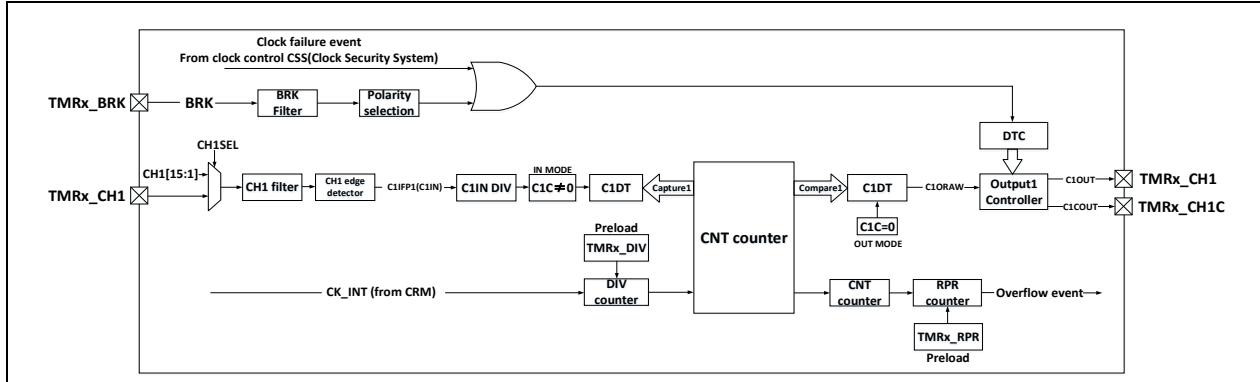


14.3.2.2 TMR10和TMR11主要功能

通用 TMRx (TMR10、TMR11) 定时器功能包括：

- 由内部用作计数时钟
- 16位支持向上、向下、双向计数器、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 DMA Burst传输

图 14-43 通用定时器 TMR10/11 框图



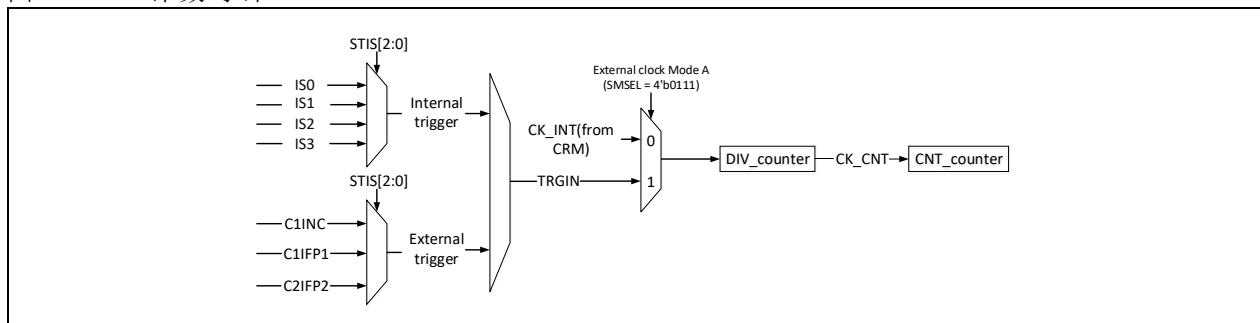
14.3.3 TMR9到TMR11功能描述

14.3.3.1 计数时钟

通用定时器 TMR9 计数时钟可从内部时钟 (CK_INT)、外部时钟 (外部时钟模式 A)、内部触发输入这些时钟源提供。

通用定时器 TMR10 和 TMR11 计数时钟仅能从内部时钟（CK_INT）提供。

图 14-44 计数时钟

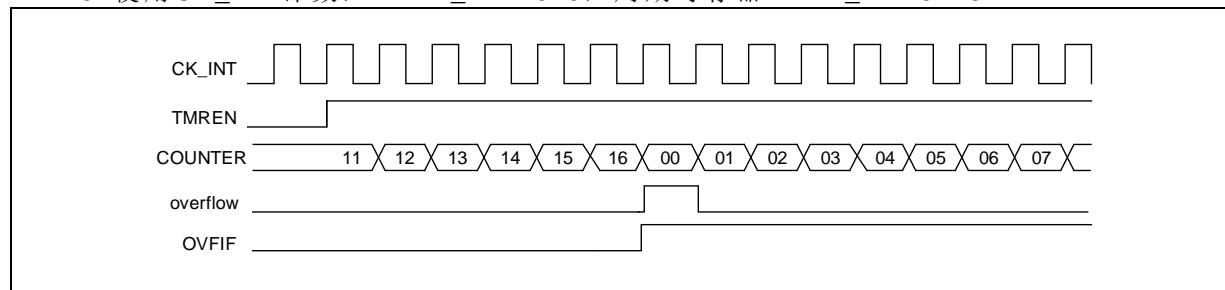


内部时钟 (CK_INT)

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 14-45 使用 CK_INT 计数, TMRx DIV=0x0, 周期寄存器 TMRx PR=0x16



外部时钟（仅 TMR9 支持）

计数时钟由 TRGIN 外部时钟源提供。

当 SMSEL=4'b0111 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[2:0], 设置 TRGIN 信号来源。

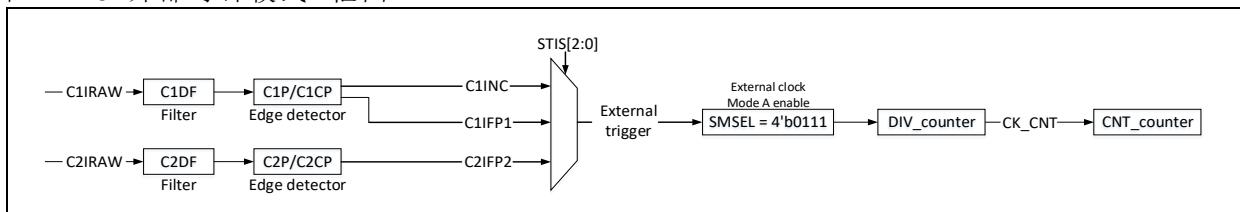
-配置 TMRx_STCTRL 寄存器 SMSEL=4'b0111, 使能外部时钟模式 A。

-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。

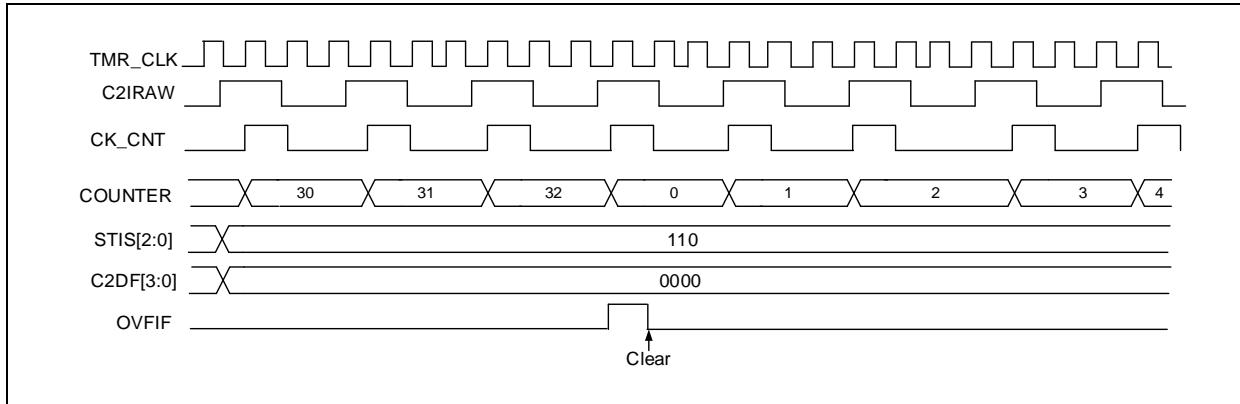
-配置 TMRx_CTRL1 寄存器 TMREN, 使能计数器。

图 14-46 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 14-47 使用外部时钟模式A计数, PR=0x32, DIV=0x0



内部触发输入 (ISx, 仅 TMR9 支持)

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。

TMR9 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 TMR9_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。

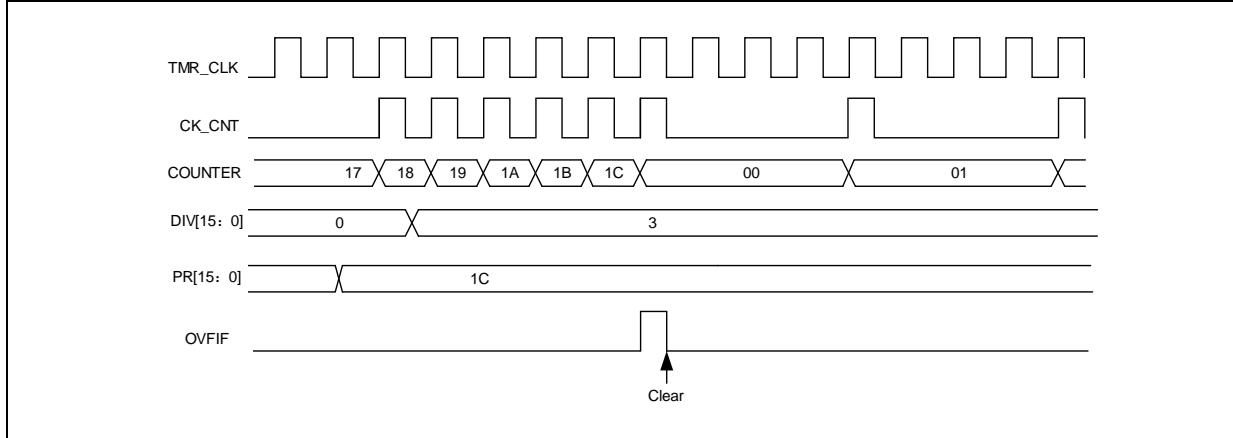
内部触发输入配置流程如下:

- 配置 TMR9_PR 寄存器, 设置计数器计数周期。
- 配置 TMR9_DIV 寄存器, 设置计数器计数频率。
- 配置 TMR9_CTRL1 寄存器 TWCMSEL[1:0]位, 设置计数器计数模式。
- 配置 TMR9_STCTRL 寄存器 STIS[2:0]位范围为 3'b000~3'b011, 选择内部触发。
- 配置 TMR9_STCTRL 寄存器 SMSEL[3:0]=4'b0111, 选择外部时钟模式 A。
- 配置 TMR9_CTRL1 寄存器 TMREN 位, 使能 TMRx 计数。

表 14-7 TMR9 内部触发连接

次定时器	IS0 (STIS=000)	IS1 (STIS=001)	IS2 (STIS=010)	IS3 (STIS=011)
TMR9		TMR3	TMR10_C1OUT	TMR11_C1OUT

图 14-48 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

TMR9 到 TMR11 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。

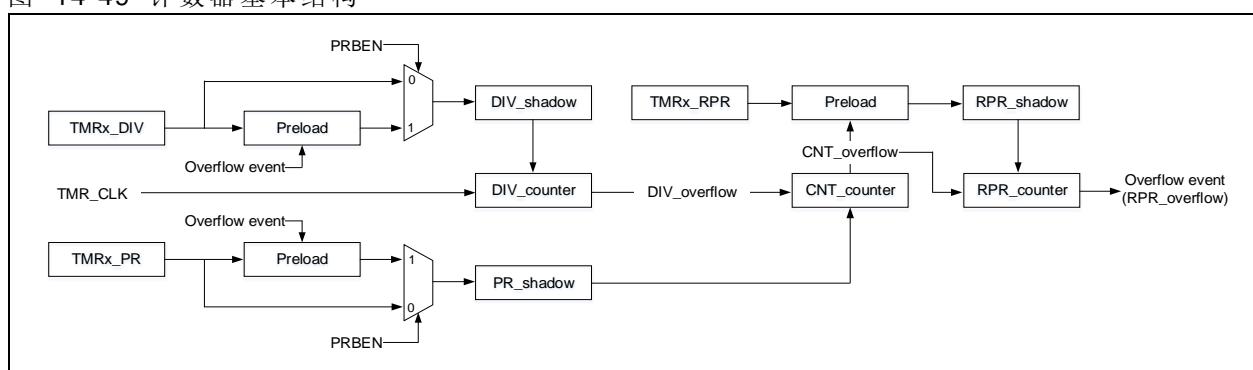
TMRx_DIV 寄存器用于设置计数器计数频率，每 (DIV[15:0]+1) 个计数时钟周期，计数器计数一次。和 TMRx_PR 寄存器类似，开启周期缓冲功能后，TMRx_DIV 寄存器值在溢出事件时更新至它的影子寄存器。

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 将禁止溢出事件产生。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 14-49 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 14-50 PRBEN=0时的溢出事件

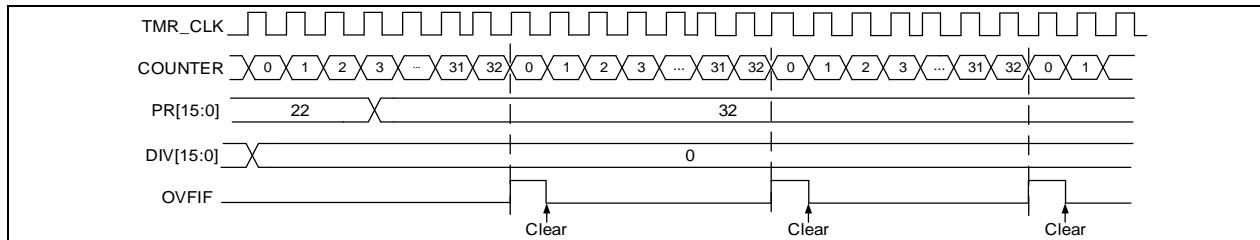
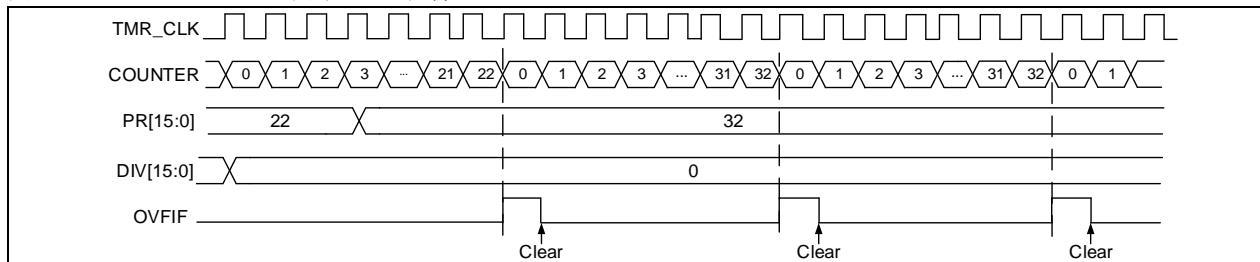


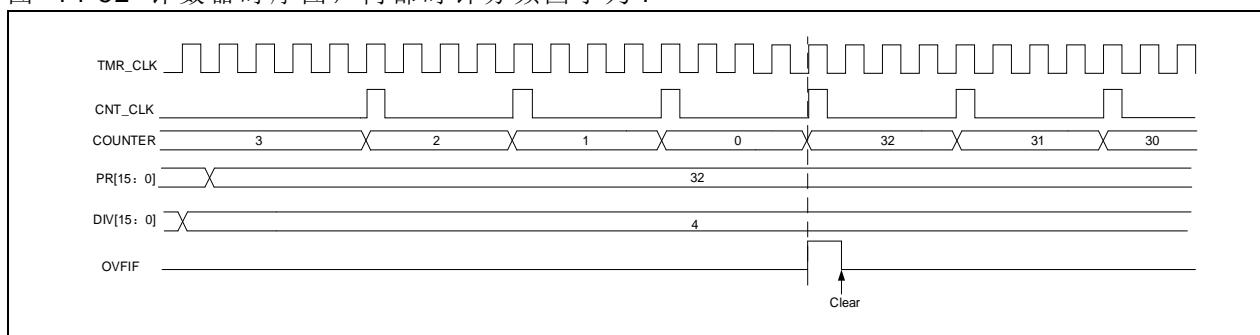
图 14-51 PRBEN=1时的溢出事件



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 14-52 计数器时序图，内部时钟分频因子为4



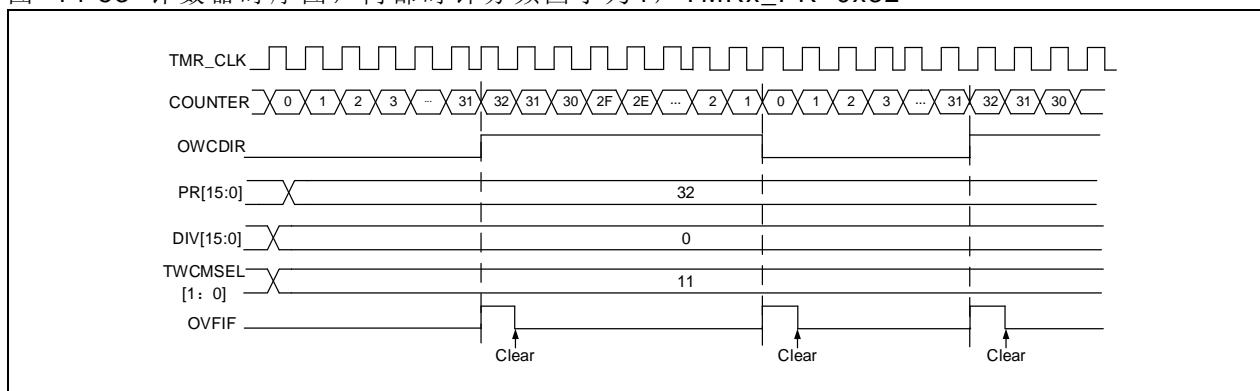
中央双向对齐计数模式

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。

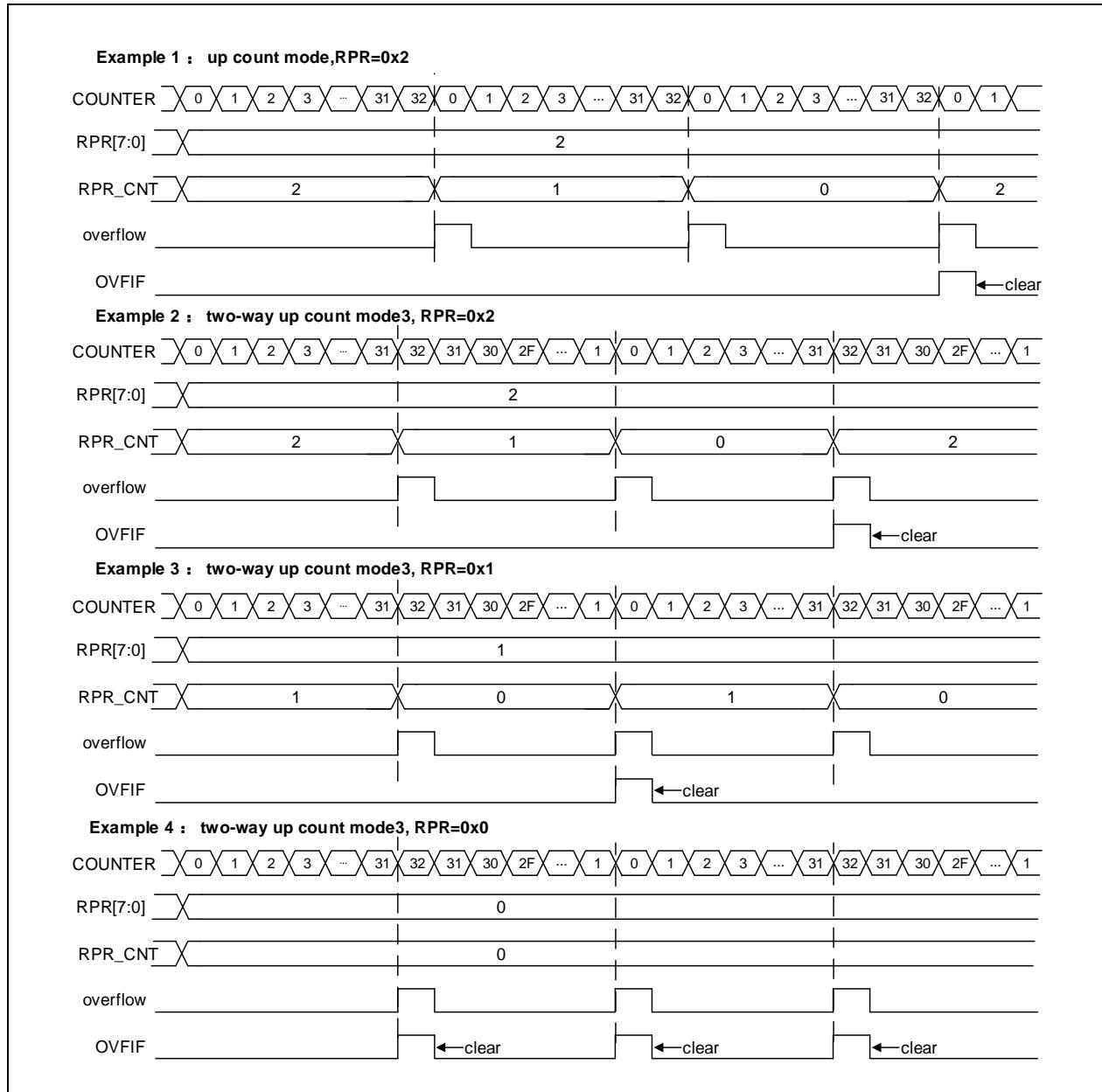
图 14-53 计数器时序图，内部时钟分频因子为1，TMRx_PR=0x32



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 14-54 向上计数模式和中央双向对齐计数模式时 OVFIF

**14.3.3.3 OVFIF位重映射**

TMRx_CTRL1 寄存器中 OVFIFRMP 位强制将溢出中断标记 OVFIF 复制到定时器计数值寄存器 TMRx_CVAL[31]中。这样便可读取计数值的同时读取 OVFIF 来判断溢出中断标记是否置起, 有利于简化计算。

14.3.3.4 TMR输入部分

TMR9 拥有 2 个独立通道, TMR10、11、拥有一个独立通道。每个通道可配置为输入或输出, 当配置为输入时, 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 CxC 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≠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）；STCI 来自次定时器控制器，由 STIS 位选择来源。单通道 TMR 仅支持选择 CxIN 来源为 CxIFPx。

-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 14-55 输入/输出通道 1 的主电路

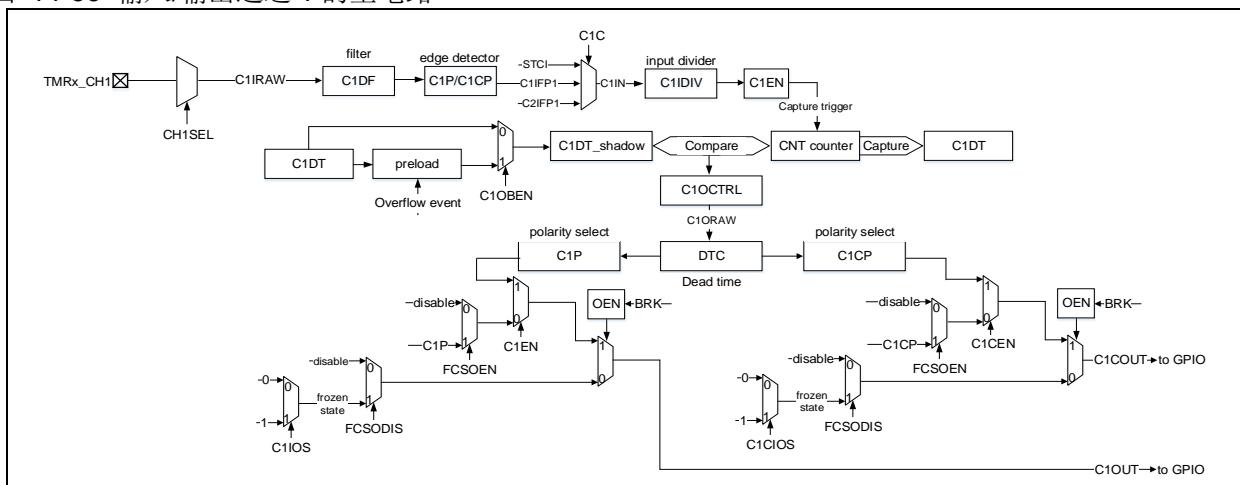
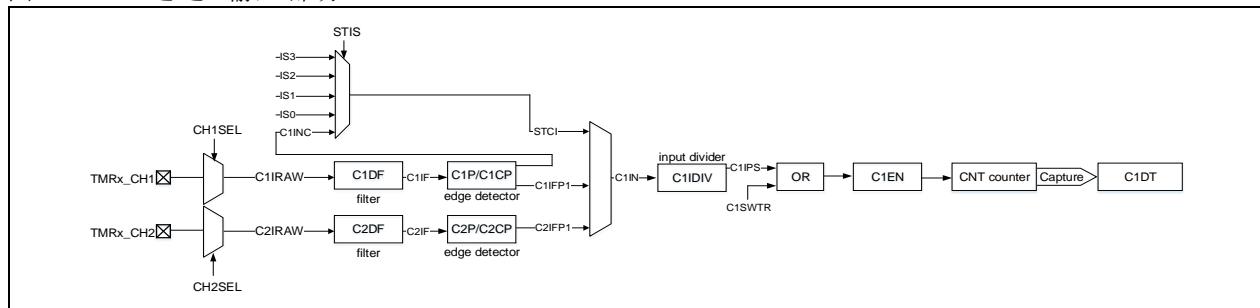


图 14-56 通道 1 输入部分



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 位、TMRx_IDEN 寄存器中的 C1DEN 位，选择中断请求或 DMA 请求。

PWM 输入（仅 TMR9 支持）

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 C2IFP2。
- 配置 C2P=1'b1，选择 C2IFP2 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=4'b0100，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 14-57 PWM 输入模式配置实例

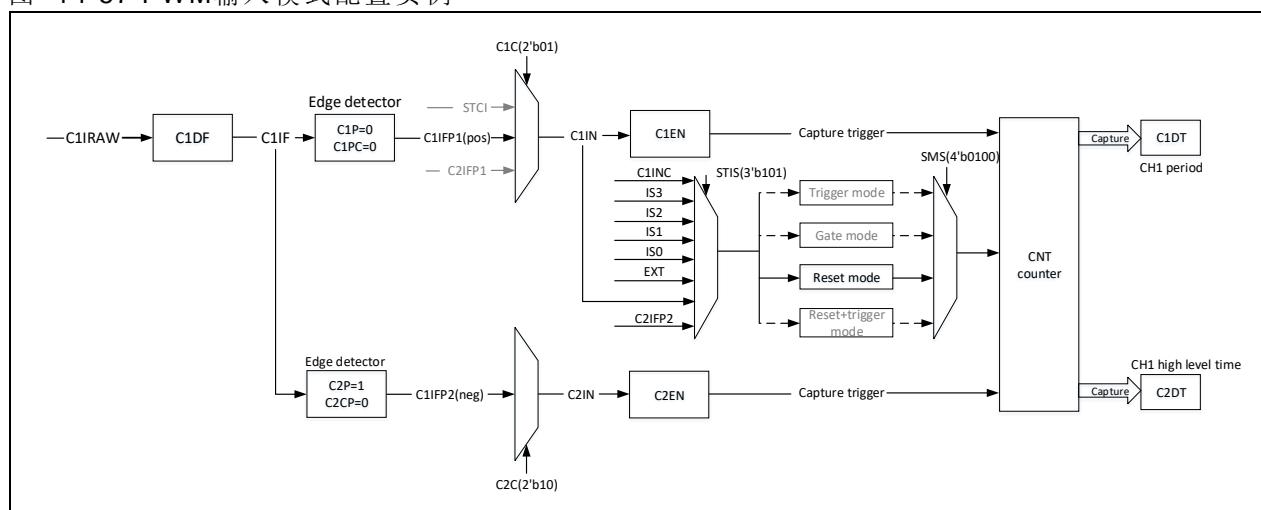
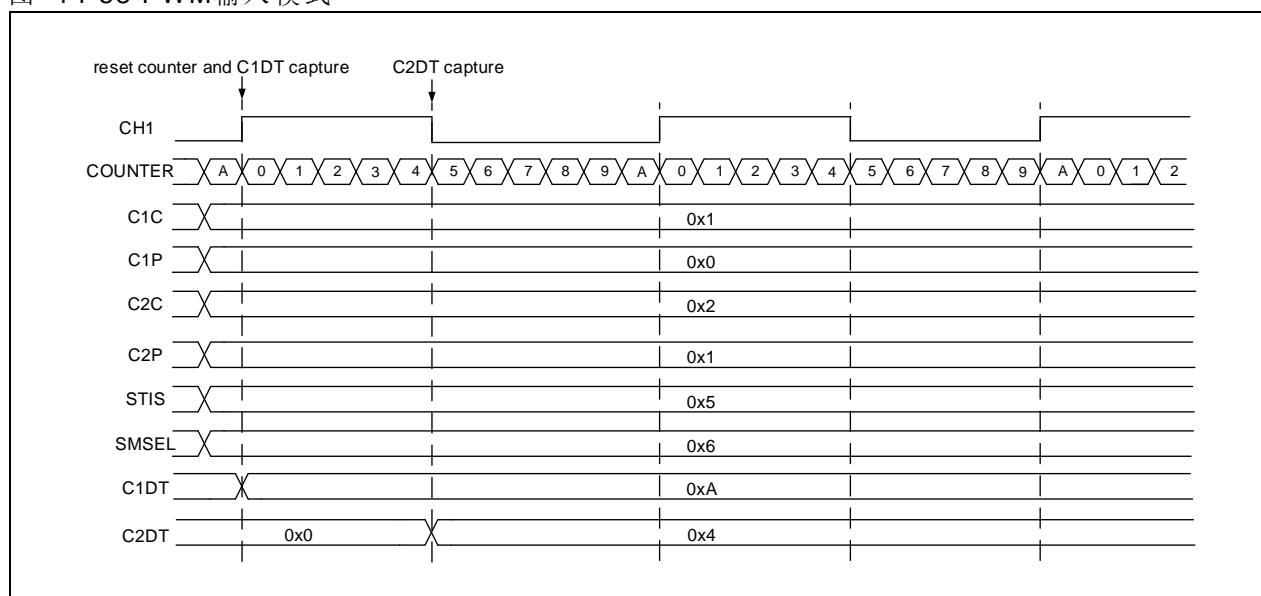


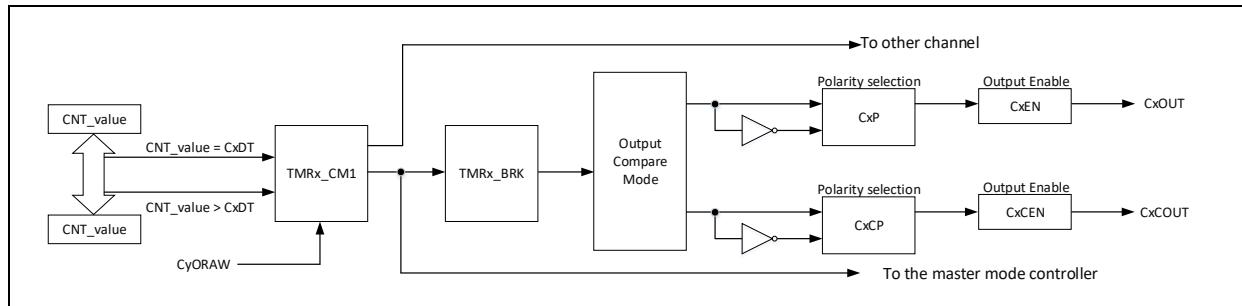
图 14-58 PWM 输入模式



14.3.3.5 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。TMR9 含有两个通道，TMR10/11 含有一个通道，如下图所示：

图 14-59 通道 1 输出部分



输出模式

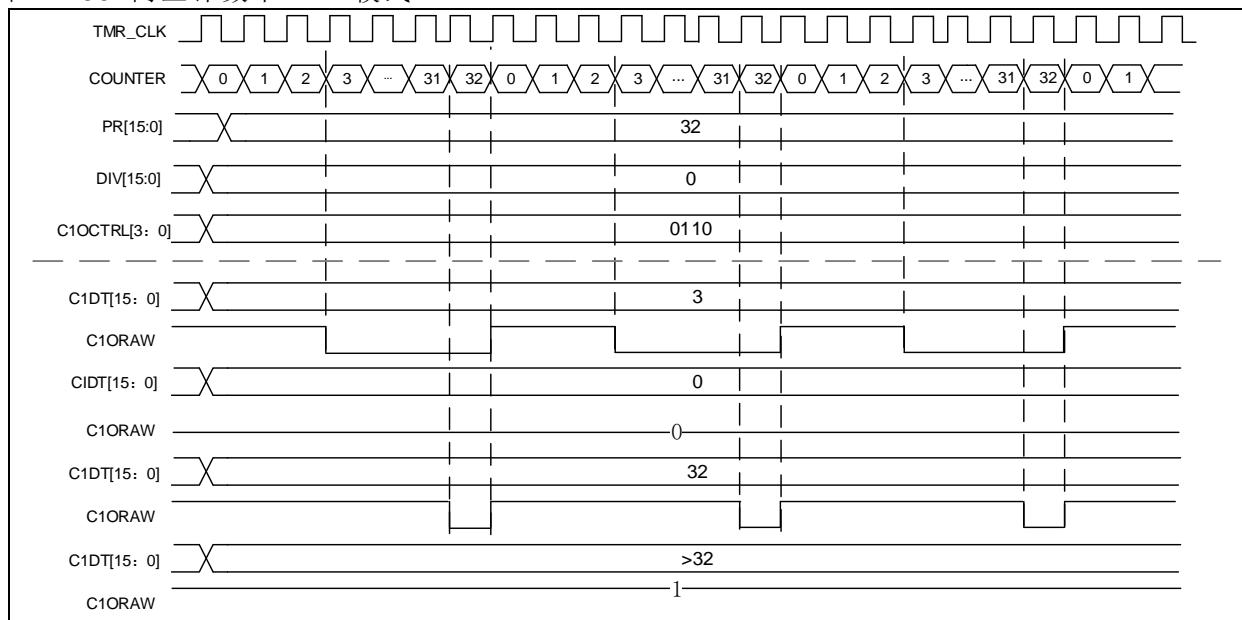
配置 $Cx\text{C}[1:0] \neq 2'b00$ 将通道配置为输出可实现多种输出模式，此时，计数器计数值将与 $Cx\text{DT}$ 寄存器值比较，并根据 $Cx\text{OCTRL}[3:0]$ 位配置的输出模式，产生中间信号 $Cx\text{ORAW}$ ，再经过输出控制逻辑处理后输送到 IO。输出信号的周期由 TMRx_PR 寄存器值配置，占空比则由 $Cx\text{DT}$ 寄存器值配置。

输出比较模式有以下子类：

PWM 模式 A: $Cx\text{OCTRL}=4'b0110$ 时，开启 PWM 模式 A。向上计数时， $\text{TMRx_C1DT} > \text{TMRx_CVAL}$ 时 $C1\text{ORAW}$ 输出高电平，否则为低电平；向下计数时， $\text{TMRx_C1DT} < \text{TMRx_CVAL}$ 时 $C1\text{ORAW}$ 输出低电平，否则为高电平。图 14-60 展示了计数器向上计数与 PWM 模式 A 配合的例子， $\text{PR}=0x32$ ， $Cx\text{DT}$ 配置为不同的值时输出信号的翻转情况。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 $Cx\text{OCTRL}$ 位为 $4'b0110$ ，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 $\text{TWCMSEL}[1:0]$ 位，设置计数器计数模式。
- -配置 TMRx_CCTRL 寄存器 $Cx\text{P}$ 位、 $Cx\text{CP}$ 位，设置输出极性。
- -配置 TMRx_CCTRL 寄存器 $Cx\text{EN}$ 位、 $Cx\text{CEN}$ 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。

图 14-60 向上计数下 PWM 模式 A

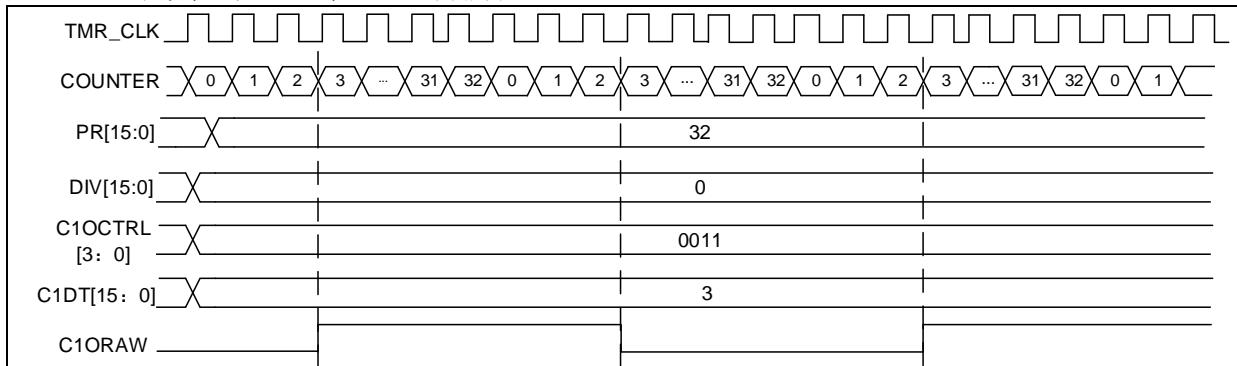


PWM 模式 B: CxOCTRL=4'b0111 时，开启 PWM 模式 B。向上计数时，TMRx_C1DT>TMRx_CVAL 时 C1ORAW 输出低电平，否则为高电平；向下计数时，TMRx_C1DT<TMRx_CVAL 时 C1ORAW 输出高电平，否则为低电平。

强制输出模式: CxOCTRL=4'b0100/0101 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。

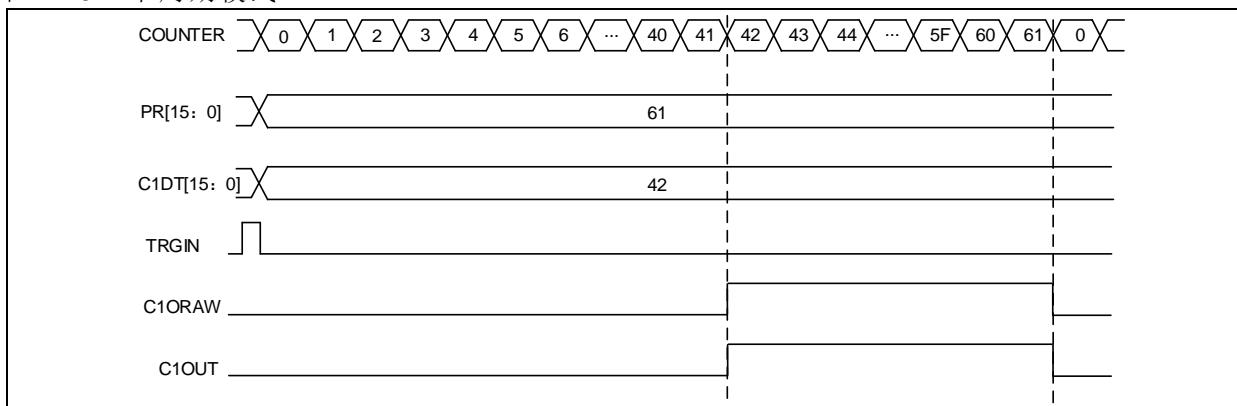
输出比较模式: CxOCTRL=4'b0001/0010/0011 时，开启输出比较模式。此时，当计数值与 CxDT 值匹配时，CxORAW 强制输出高电平（CxOCTRL=4'b0001）、低电平（CxOCTRL=4'b0010）或进行电平翻转（CxOCTRL=4'b0011）。图 14-61 展示了输出比较模式（翻转）的例子，C1DT=0x3，当计数值等于 0x3 时，输出电平 C1OUT 被翻转。

图 14-61 计数值与 C1DT 值匹配时翻转 C1ORAW



单周期模式(仅 TMR9): 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。图 14-62 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 14-62 单周期模式

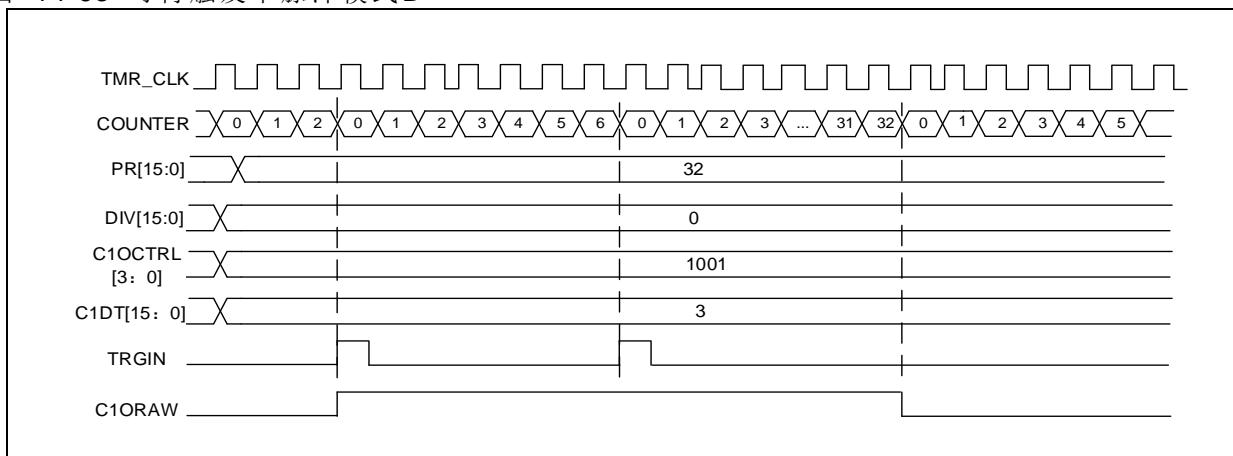


可再触发单周期模式 A(仅 TMR9): CxOCTRL=4'b1000 时，开启可再触发单周期模式 A。向上计数时，检测到 TRGIN 前，C1ORAW 为高，检测到 TRGIN 后，CxORAW 以 PWM 模式 A 运行，下一次溢出事件时，CxORAW 再次为高；向下计数时，检测到 TRGIN 前，C1ORAW 为低，检测到 TRGIN 后，CxORAW 以 PWM 模式 A 运行，下一次溢出事件时，CxORAW 再次为低；检测到触发信号时，输出脉冲，若在计数周期结束前再次检测到触发信号，脉冲将延长。需搭配组合复位+触发模式使用，且不支持中央双向对齐计数模式。

可再触发单周期模式 B(仅 TMR9): CxOCTRL=4'b1001 时，开启可再触发单周期模式 B。向上计数时，检测到 TRGIN 前，C1ORAW 为低，检测到 TRGIN 后，CxORAW 以 PWM 模式 B 运行，下一次溢出事件时，CxORAW 再次为低；向下计数时，检测到 TRGIN 前，C1ORAW 为高，检测到 TRGIN 后，CxORAW 以 PWM 模式 B 运行，下一次溢出事件时，CxORAW 再次为高；检测到触发信号时，输出脉冲，若在计数周期结束前再次检测到触发信号，脉冲将延长。需搭配组合复位+触发模式使用，且不支持中央双向对齐计数模式。

数周期结束前再次检测到触发信号，脉冲将延长。需搭配组合复位+触发模式使用，且不支持中央双向对齐计数模式。图 14-63 展示了可再触发 PWM 模式 B 的例子。

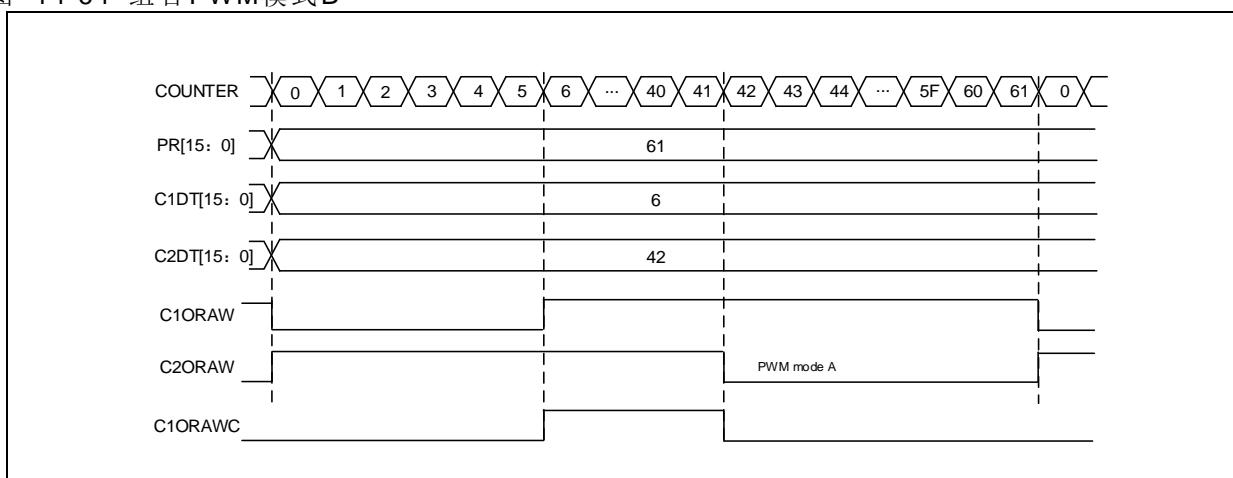
图 14-63 可再触发单脉冲模式B



组合 PWM 模式 A(仅 TMR9): CxOCTRL=4'b1100 时，开启组合 PWM 模式 A。CxORAW 以 PWM 模式 A 运行，CyORAW 运行模式由通道 y 配置决定，CxORAWC 是 CxORAW 和 CyORAW 逻辑或运算后的输出（对于通道 1，y=2；对于通道 2，y=1）。

组合 PWM 模式 B(仅 TMR9): CxOCTRL=4'b1101 时，开启组合 PWM 模式 B。CxORAW 以 PWM 模式 B 运行，CyORAW 运行模式由通道 y 配置决定，CxORAWC 是 CxORAW 和 CyORAW 逻辑与运算后的输出（对于通道 1，y=2；对于通道 2，y=1）。图 14-64 展示了组合 PWM 模式 B 的例子。

图 14-64 组合 PWM 模式B



快速输出模式(仅 TMR9): 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。

主定时器事件输出 (TMR9 支持)

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) 或复位事件。
- 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

TMR9 到 TMR11 通道包含反向通道输出, 通过 CxCEN 使能, 通过 CxCP 配置极性。

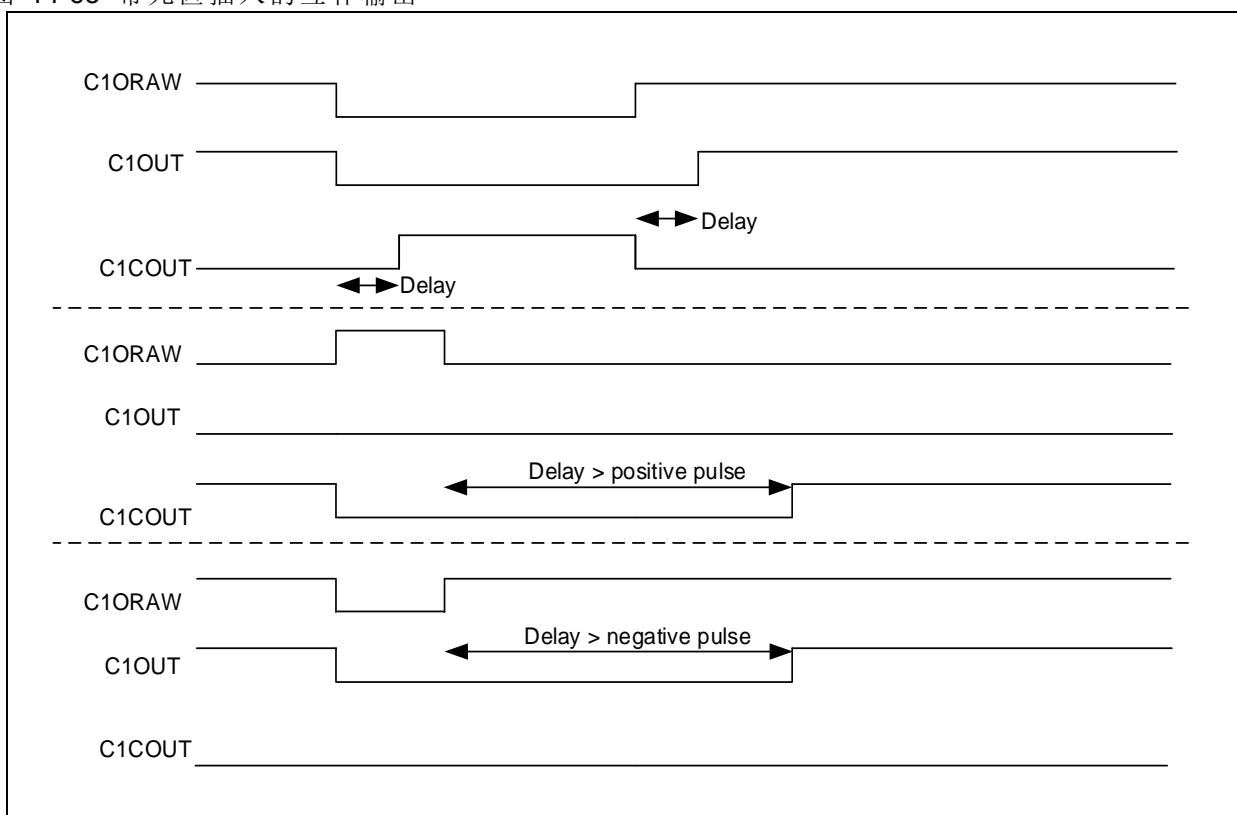
当转换为 IDLEF 状态, 即 OEN 下降到 0, 死区被激活。

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 14-65 带死区插入的互补输出



14.3.3.6 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。

刹车信号来源包括刹车输入引脚、比较器 1 输出、比较器 2 输出以及软件刹车请求(BRKSWTR)。BRK 极性由 BRKV 位控制, 比较器输出刹车信号极性由 BRKCMPP (BRK2CMPP) 位控制; 刹车输入引脚极性由 BRKxINP 位控制。来自刹车输入引脚通过 BRKINEN (BRK2INEN) 使能, 比较器 1 或 2 输出通过 BRKCMPP1EN 和 BRKCMPP2EN (BRK2CMPP1EN 和 BRK2CMPP2EN) 使能。

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CxCOUP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 t_{mr_clk} 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 14-66 TMR刹车电路

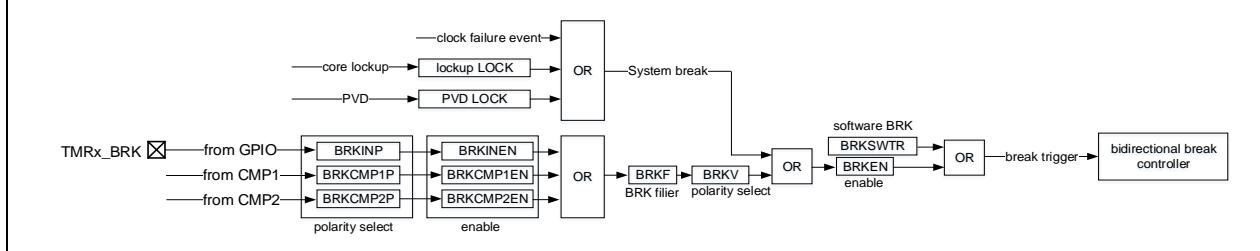


图 14-67 TMR输出控制

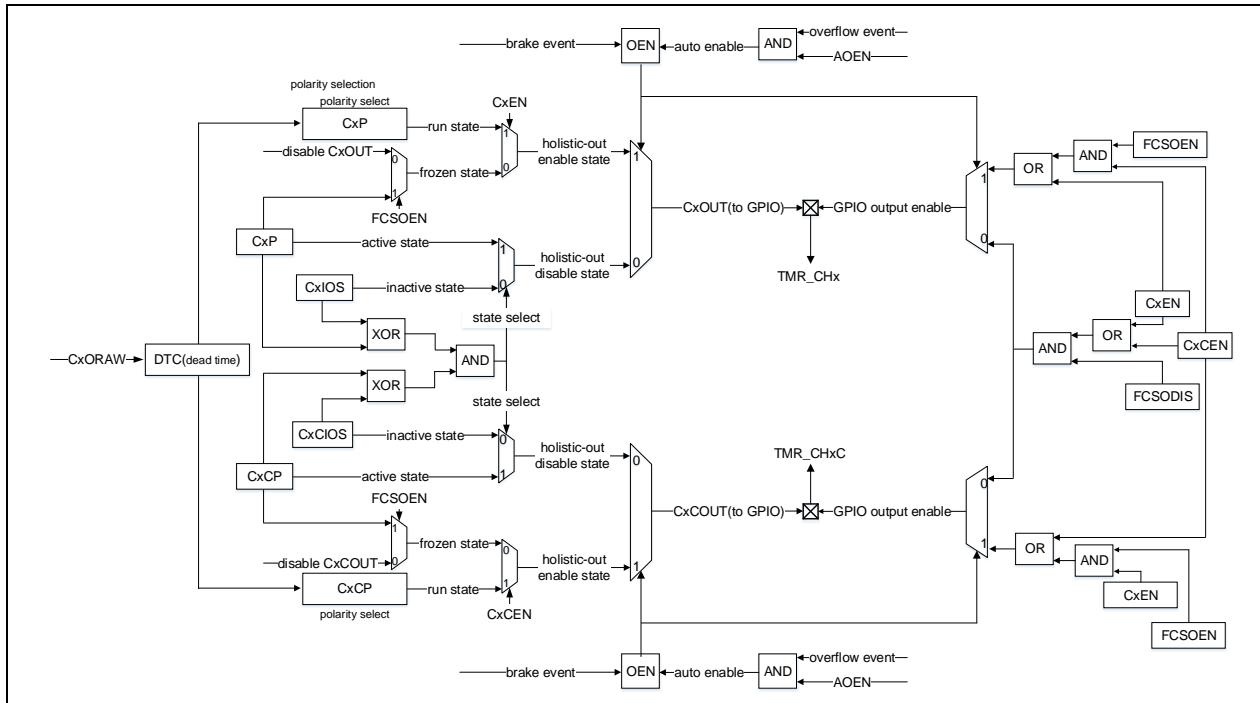
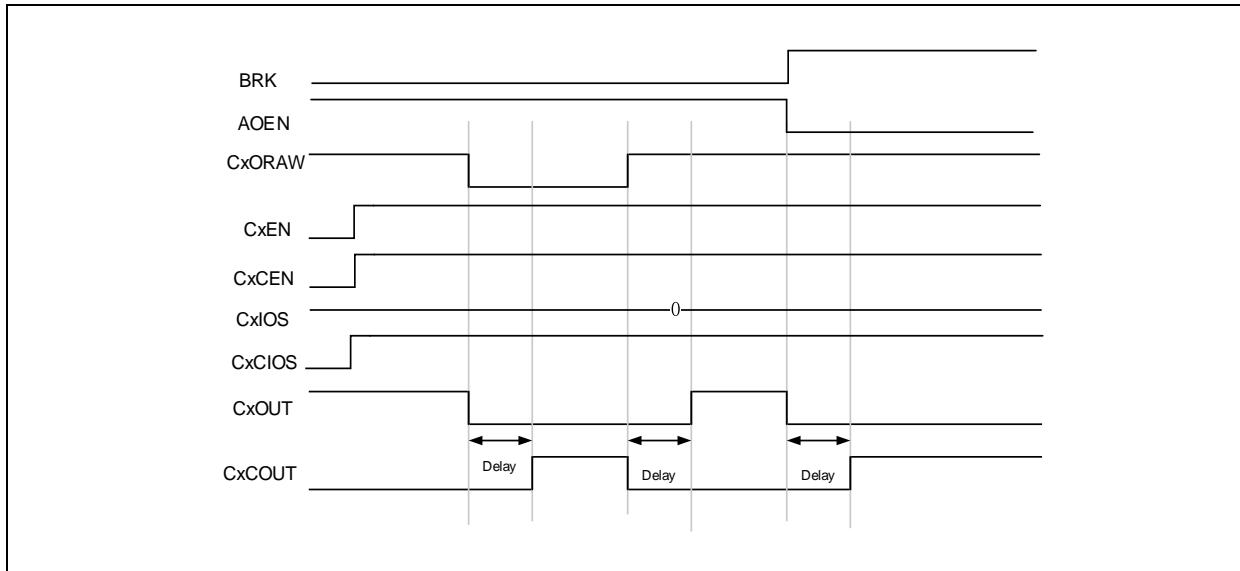


图 14-68 TMR刹车功能的例子



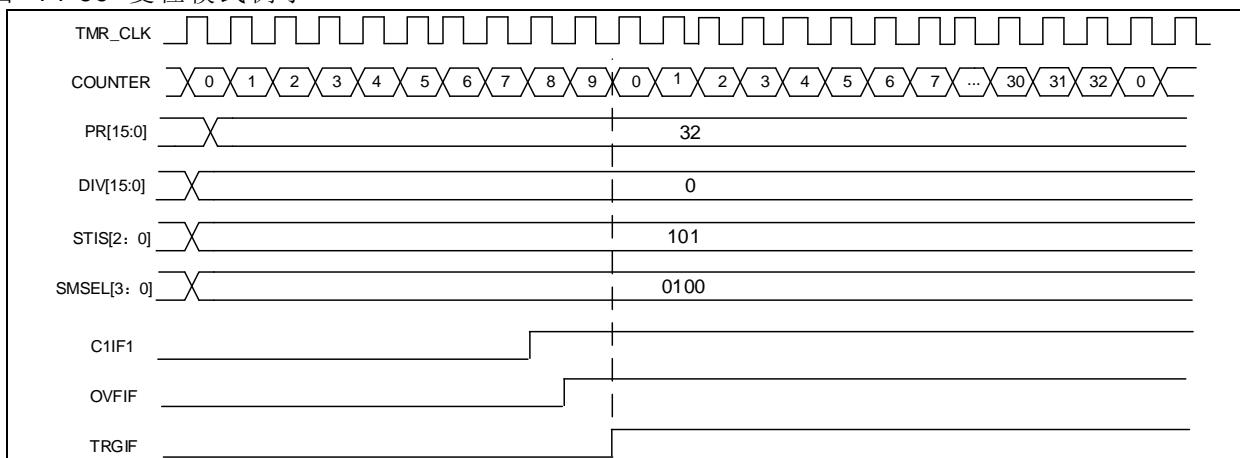
14.3.3.7 TMR同步

TMR9 可作为次定时器与主定时器由内部信号进行同步，次定时器由 **SMSEL[3: 0]**位选择从模式，即次定时器的工作模式。

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。

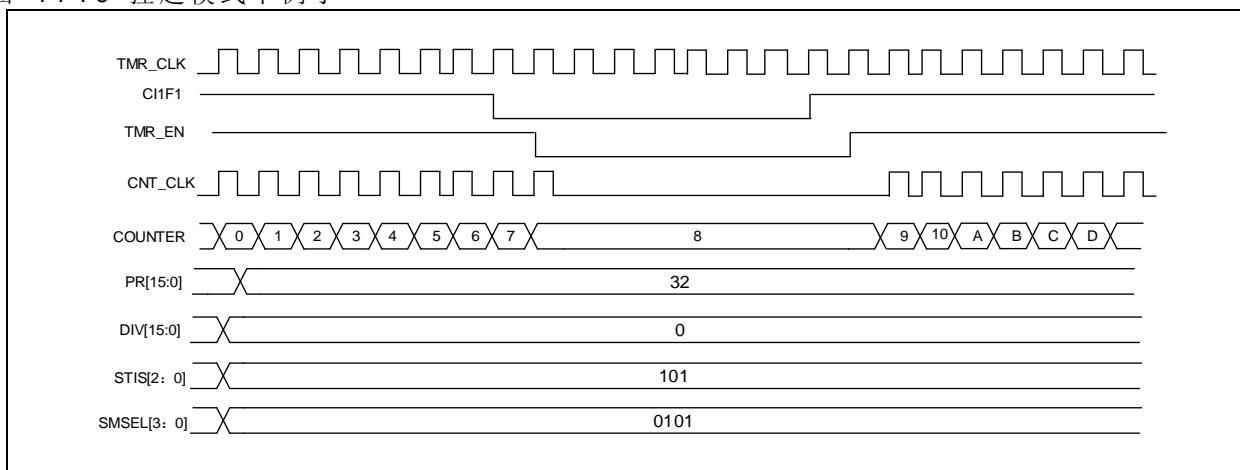
图 14-69 复位模式例子



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。

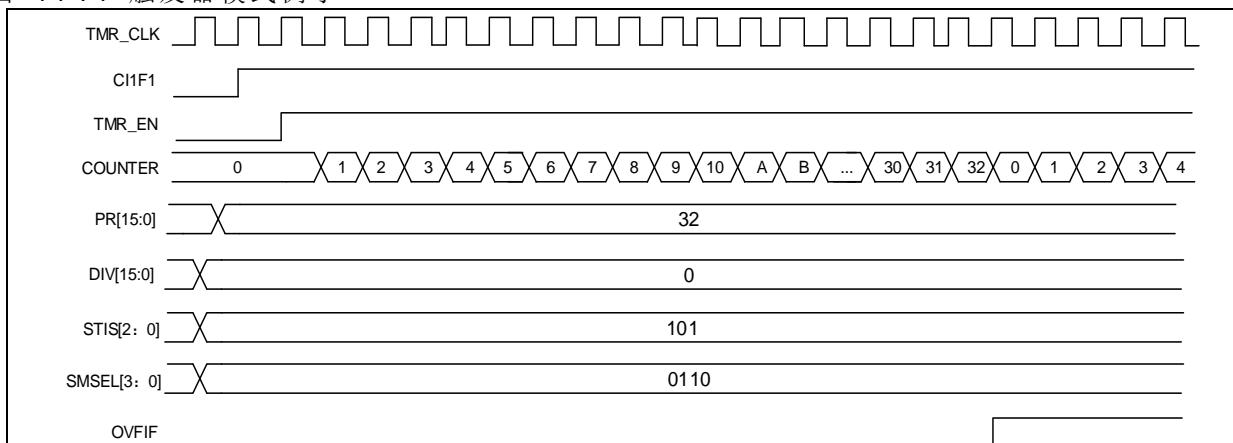
图 14-70 挂起模式下例子



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 14-71 触发器模式例子

**从模式：组合复位+触发模式**

计数器将在选中的触发输入上升沿初始化计数器，并产生溢出事件，需搭配可再触发单周期模式使用，详见可再触发单周期模式部分描述。

14.3.3.8 TMR DMA

TMR 中有溢出事件 DMA 请求、触发事件 DMA 请求、霍尔事件 DMA 请求以及通道事件 DMA 请求，通过使能 TMRx_IDEN 寄存器中相应位，开启 DMA 请求功能，产生相应事件时触发输出 DMA 请求给到 DMA 外设。

TMR DMA Burst 功能

TMR 还支持 TMR DMA Burst 功能，通过 TMRx_IDEN 寄存器使能某个事件的 DMA 请求，触发 DMA 改写多个 TMR 连续寄存器。

以溢出事件触发 TMR DMA Burst 功能为例，配置如下：

- 配置 TMRx_IDEN 寄存器中 OVFDEN 位，使能溢出事件触发 DMA 请求；
- 配置 TMRx_DMACTRL 寄存器中 DTB 位设置 Burst 传输次数；
- 配置 TMRx_DMACTRL 寄存器中 ADDR 位设置 Burst 传输的起始地址；
- 使能计数器。

以上配置的 TMR DMA Burst 传输流程如下：

当发生溢出事件时，TMR 将会产生溢出事件的 DMA 请求给到 DMA，DMA 根据请求将数据写入 TMRx_DMADT 寄存器，在 TMR 内部则会将 DMADT 位的数据写入 Burst 传输起始地址寄存器，并发送 ACK 信号给到 TMR，TMR 接收到 ACK 信号后，将清除当前 DMA 请求；TMR 检测到 Burst 传输并未全部完成，会重新置起溢出事件 DMA 请求给到 DMA，DMA 根据请求将数据再次写入 TMRx_DMADT 寄存器，在 TMR 内部则会将 DMADT 寄存器的数据写入 Burst 传输起始地址 +0x4 地址寄存器，并发送 ACK 信号给到 TMR；以此往复，直到 Burst 传输最后一次操作时，检测到此时 Burst 传输已全部完成，溢出事件 DMA 请求 REQ 将不会再被重新置起，直到下一次新的溢出事件到来。

注：使用 TMR DMA Burst 功能时，起始地址到结束地址这个区间，不应有空寄存器以及不应包含 TMRx_DMACTRL 和 TMRx_DMADT 寄存器。

14.3.3.9 调试模式

当微控制器进入调试模式（Cortex®-M4F 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。

14.3.4 TMR9寄存器描述

表 14-8 TMR9寄存器图和复位值

寄存器简称	基址偏移量	复位值
TMR9_CTRL1	0x00	0x0000 0000
TMR9_CTRL2	0x04	0x0000 0000
TMR9_STCTRL	0x08	0x0000 0000
TMR9_IDEN	0x0C	0x0000 0000
TMR9ISTS	0x10	0x0000 0000
TMR9_SWEVT	0x14	0x0000 0000
TMR9_CM1	0x18	0x0000 0000
TMR9_CCTRL	0x20	0x0000 0000
TMR9_CVAL	0x24	0x0000 0000
TMR9_DIV	0x28	0x0000 0000
TMR9_PR	0x2C	0x0000 0000
TMR9_RPR	0x30	0x0000 0000
TMR9_C1DT	0x34	0x0000 0000
TMR9_C2DT	0x38	0x0000 0000
TMR9_BRK	0x44	0x0000 0000
TMR9_DMACTRL	0x48	0x0000 0000
TMR9_DMADT	0x4C	0x0000 0000
TMR9_OPT2	0x60	0x0001 0000
TMR9_CHSEL	0x68	0x0000 0000

14.3.4.1 TMR9控制寄存器1 (TMR9_CTRL1)

域	简称	复位值	类型	功能
位 31: 12	保留	0x00	resd	保持默认值。
位 11	OVFIFRMP	0x0	rw	OVFIF 状态位重映射 (OVFIF status bit remapping) 0: 无重映射; 1: 使能重映射, OVFIF 位复制到 TMRx_CVAL[31]。
位 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: 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。

14.3.4.2 TMR9控制寄存器2 (TMR9_CTRL2)

域	简称	复位值	类型	功能
位 31: 12	保留	0x00000	resd	保持默认值。
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)
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	保留	0x0	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: 软件溢出或复位; 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 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。

14.3.4.3 TMR9次定时器控制寄存器 (TMR9_STCTRL)

域	简称	复位值	类型	功能
位 31: 17	保留	0x00	resd	保持默认值。
位 15: 8	保留	0x00	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) ; 其他值: 保留。

				关于每个定时器中 ISx 的细节，参见表 14-7。
位 3	保留	0x0	resd	保留，保持默认值。
位 16, 2: 0	SMSEL	0x0	rw	次定时器模式选择 (Subordinate TMR mode selection) 0000: 关闭从模式; 0001: 编码模式 A; 0010: 编码模式 B; 0011: 编码模式 C; 0100: 复位模式 - TRGIN 输入上升沿时，重新初始化计数器; 0101: 挂起模式 - TRGIN 输入高电平时，计数器计数; 0110: 触发模式 - TRGIN 输入上升沿时，产生触发事件; 0111: 外部时钟模式 A - TRGIN 输入上升沿提供时钟; 1000: 组合复位+触发模式 - TRGIN 输入上升沿时，重新初始化计数器，产生溢出事件并启动计数器; 注：编码器模式 A/B/C 配置方法请查看计数模式章节。

14.3.4.4 TMR9 DMA中断使能寄存器 (TMR9_IDEN)

域	简称	复位值	类型	功能
位 31: 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	0x00	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	0x00	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。

14.3.4.5 TMR9中断状态寄存器 (TMR9_ISTS)

域	简称	复位值	类型	功能
位 31: 14	保留	0x00	resd	保持默认值。
位 13	SBRKIF	0x0	rw0c	系统刹车中断标记 (System brake interrupt flag) 用于标记系统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 12: 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 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。

14.3.4.6 TMR9软件事件寄存器 (TMR9_SWEVT)

域	简称	复位值	类型	功能
位 31: 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: 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	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。

14.3.4.7 TMR9通道模式寄存器1 (TMR9_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 31: 25	保留	0x0	resd	保持默认值。
位 23: 17	保留	0x0	resd	保持默认值。
位 15	保留	0x0	resd	保持默认值。
位 24, 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	保留	0x0	resd	保持默认值。
位 16, 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。 0000: 断开。断开 C1ORAW 到 C1OUT 的输出；

				0001: 当 TMRx_CVAL=TMRx_C1DT 时, 设置 C1ORAW 为高。 0010: 当 TMRx_CVAL=TMRx_C1DT 时, 设置 C1ORAW 为低。 0011: 当 TMRx_CVAL=TMRx_C1DT 时, 切换 C1ORAW 的电平。 0100: 固定 C1ORAW 为低; 0101: 固定 C1ORAW 为高; 0110: 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。 0111: 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。 1000: 可再触发单周期模式 A; —向上计数时, 检测到 TRGIN 前, C1ORAW 为高, 检测到 TRGIN 后, C1ORAW 以 PWM 模式 A 运行, 下一次溢出事件时, C1ORAW 再次为高; —向下计数时, 检测到 TRGIN 前, C1ORAW 为低, 检测到 TRGIN 后, C1ORAW 以 PWM 模式 A 运行, 下一次溢出事件时, C1ORAW 再次为低。 1001: 可再触发单周期模式 B; —向上计数时, 检测到 TRGIN 前, C1ORAW 为低, 检测到 TRGIN 后, C1ORAW 以 PWM 模式 B 运行, 下一次溢出事件时, C1ORAW 再次为低; —向下计数时, 检测到 TRGIN 前, C1ORAW 为高, 检测到 TRGIN 后, C1ORAW 以 PWM 模式 B 运行, 下一次溢出事件时, C1ORAW 再次为高。 1010: 保留; 1011: 保留; 1100: 组合 PWM 模式 A, C1ORAW 以 PWM 模式 A 运行, C1ORAWC 是 C1ORAW 和 C2ORAW 逻辑或运算后的输出; 1101: 组合 PWM 模式 B, C1ORAW 以 PWM 模式 B 运行, C1ORAWC 是 C1ORAW 和 C2ORAW 逻辑与运算后的输出; 1110: 保留; 1111: 保留。 注: 除'0000'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: 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。

输入捕获模式:

域	简称	复位值	类型	功能
位 31: 16	保留	0x0	resd	保持默认值。
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; 0001: 采样频率 $f_{SAMPLING} = f_{CK_INT}$, N=2; 0010: 采样频率 $f_{SAMPLING} = f_{CK_INT}$, N=4; 0011: 采样频率 $f_{SAMPLING} = f_{CK_INT}$, N=8; 0100: 采样频率 $f_{SAMPLING} = f_{DTS}/2$, N=6; 0101: 采样频率 $f_{SAMPLING} = f_{DTS}/2$, N=8; 0110: 采样频率 $f_{SAMPLING} = f_{DTS}/4$, N=6; 0111: 采样频率 $f_{SAMPLING} = f_{DTS}/4$, N=8; 1000: 采样频率 $f_{SAMPLING} = f_{DTS}/8$, N=6; 1001: 采样频率 $f_{SAMPLING} = f_{DTS}/8$, N=8; 1010: 采样频率 $f_{SAMPLING} = f_{DTS}/16$, N=5; 1011: 采样频率 $f_{SAMPLING} = f_{DTS}/16$, N=6; 1100: 采样频率 $f_{SAMPLING} = f_{DTS}/16$, N=8; 1101: 采样频率 $f_{SAMPLING} = f_{DTS}/32$, N=5; 1110: 采样频率 $f_{SAMPLING} = f_{DTS}/32$, N=6; 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。

14.3.4.8 TMR9通道控制寄存器 (TMR9_CCTRL)

域	简称	复位值	类型	功能
位 31: 8	保留	0x00	resd	保持默认值。
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 14-9 带刹车功能的互补输出通道 OCx 和 OCxN 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	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 (与定时器断开) 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。	输出禁止 (与定时器断开) 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	关闭状态 (输出使能且为无效电平) 异步地: CxOUT=CxP, Cx_EN=1, CxCOUT=CxCP, CxCEN=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。	关闭状态 (输出使能且为无效电平) 异步地: CxOUT=CxP, Cx_EN=1, CxCOUT=CxCP, CxCEN=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。
		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。

14.3.4.9 TMR9计数值寄存器 (TMR9_CVAL)

域	简称	复位值	类型	功能
位 31	OVFIFCPY	0x0	ro	OVFIF 副本 (OVFIF copy) OVFIFRMP 为 0 时, 该位保留, 读为 0; OVFIFRMP 为 1 时, 该位是 OVFIF 位的只读副本。
位 30: 16	保留	0x0000	resd	保持默认值。
位 15: 0	CVAL	0x0000	rw	计数器的值 (Counter value)

14.3.4.10 TMR9预分频器寄存器 (TMR9_DIV)

域	简称	复位值	类型	功能
位 31: 16	保留	0x00	resd	保持默认值。
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。

14.3.4.11 TMR9周期寄存器 (TMR9_PR)

域	简称	复位值	类型	功能
位 31: 16	保留	0x00	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。

14.3.4.12 TMR9重复周期寄存器 (TMR9_RPR)

域	简称	复位值	类型	功能
位 31: 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。

14.3.4.13 TMR9通道1数据寄存器 (TMR9_C1DT)

域	简称	复位值	类型	功能
位 31: 16	保留	0x00	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。

14.3.4.14 TMR9通道2数据寄存器 (TMR9_C2DT)

域	简称	复位值	类型	功能
位 31: 16	保留	0x00	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。

14.3.4.15 TMR9刹车寄存器 (TMR9_BRK)

域	简称	复位值	类型	功能
位 31: 29	保留	0x0	resd	保持默认值。
位 28	BRKBID	0x0	resd	刹车双向 (Brake bidirectional) 0: 关闭双向刹车模式; 1: 开启双像刹车模式。
位 27	保留	0x0	resd	保持默认值。
位 26	BRKRLS	0x0	rw	刹车解除(Brake release) 0: 启动刹车输入; 1: 解除刹车输入。
位 25: 20	保留	0x00	resd	保持默认值。
位 19: 16	BRKF	0x0	rw	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; 0001: 采样频率 $f_{SAMPLING} = f_{CK_INT}$, N=2; 0010: 采样频率 $f_{SAMPLING} = f_{CK_INT}$, N=4; 0011: 采样频率 $f_{SAMPLING} = f_{CK_INT}$, N=8; 0100: 采样频率 $f_{SAMPLING} = f_{DTS}/2$, N=6; 0101: 采样频率 $f_{SAMPLING} = f_{DTS}/2$, N=8; 0110: 采样频率 $f_{SAMPLING} = f_{DTS}/4$, N=6; 0111: 采样频率 $f_{SAMPLING} = f_{DTS}/4$, N=8; 1000: 采样频率 $f_{SAMPLING} = f_{DTS}/8$, N=6; 1001: 采样频率 $f_{SAMPLING} = f_{DTS}/8$, N=8; 1010: 采样频率 $f_{SAMPLING} = f_{DTS}/16$, N=5; 1011: 采样频率 $f_{SAMPLING} = f_{DTS}/16$, N=6; 1100: 采样频率 $f_{SAMPLING} = f_{DTS}/16$, N=8; 1101: 采样频率 $f_{SAMPLING} = f_{DTS}/32$, N=5; 1110: 采样频率 $f_{SAMPLING} = f_{DTS}/32$, N=6; 1111: 采样频率 $f_{SAMPLING} = f_{DTS}/32$, N=8。 注: 如无法保证刹车输入干净, 建议开启刹车输入滤波。
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: TMRx_BRK: BRKBID、BRKRLS、BRKEN、BRKV、 BRKF、AOEN 和 DTC TMRx_CTRL2: CxIOS 和 CxCOS 10: 2 级写保护, 除 3 级写保护的内容外, 以下位也受写 保护: TMRx_CCTRL: CxP 和 CxCP TMRx_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: TMRx_CM1: CxOCTRL 和 CxOBEN 注: WPC>0 时将无法再次被修改, 直到系统复位。
位 7: 0	DTC	0x00	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, BRKF、AOEN、BRKBID、BRKRLS、BRKV、BRKEN、
FCSODIS、FCSOEN 和 DTC[7: 0]位均可被写保护, 有必要在第一次写入
TMR9_BRK 寄存器时对它们进行配置。

14.3.4.16 TMR9 DMA控制寄存器 (TMR9_DMACTRL)

域	简称	复位值	类型	功能
位 31: 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 TMRx_CTRL1 所在地址开始的偏移量: 00000: TMRx_CTRL1, 00001: TMRx_CTRL2, 00010: TMRx_STCTRL,

14.3.4.17 TMR9 DMA数据寄存器 (TMR9_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: TMRx 外设地 址 + ADDR*4 至 TMRx 外设地址 + ADDR*4 + DTB*4。

14.3.4.18 TMR9选项寄存器2 (TMR9_OPT2)

域	简称	复位值	类型	功能
位 31: 12	保留	0x0	rw	保持默认值。
位 11	BRKCMPP2P	0x0	rw	BRK CMP2 输入极性 (BRK CMP2 input polarity) 0: 高电平有效; 1: 低电平有效。
位 10	BRKCMPP1P	0x0	rw	BRK CMP1 输入极性 (BRK CMP1 input polarity) 0: 高电平有效; 1: 低电平有效。
位 9	BRKINP	0x0	rw	BRK BKIN 输入极性 (BRK BKIN input polarity) 0: 高电平有效; 1: 低电平有效。
位 8: 3	保留	0x0	resd	保持默认值。
位 2	BRKCMPP2EN	0x0	rw	BRK CMP2 使能 (BRK CMP2 enable) 0: 禁止 CMP2 输入; 1: 使能 CMP2 输入。
位 1	BRKCMPP1EN	0x0	rw	BRK CMP1 使能 (BRK CMP1 enable) 0: 禁止 CMP1 输入; 1: 使能 CMP1 输入。
位 0	BRKINEN	0x1	rw	BRK BKIN 使能 (BRK BKIN enable) 0: 禁止 BKIN 输入; 1: 使能 BKIN 输入。

14.3.4.19 TMR9通道选择寄存器 (TMR9_CHSEL)

域	简称	复位值	类型	功能
位 31: 12	保留	0x0000	resd	保持默认值。
位 11: 8	CH2SEL	0x0	rw	CH2 输入选择 (CH2 input selection) 0000: TMR9_CH2; 0001: CMP1; 0010: CMP2; 其他值: 保留。
位 7: 4	保留	0x0	resd	保持默认值。
位 3: 0	CH1SEL	0x0	rw	CH1 输入选择 (CH1 input selection) 0000: TMR9_CH1; 0001: CMP1; 0010: CMP2; 其他值: 保留。

14.3.5 TMR10、TMR11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 TMR10、TMR11 的所有寄存器映射到一个 32 位可寻址（编址）空间。

表 14-10 TMR10、TMR11寄存器和复位值

寄存器简称	基址偏移量	复位值
TMRx_CTRL1	0x00	0x0000 0000
TMRx_CTRL2	0x04	0x0000 0000
TMRx_IDEN	0x0C	0x0000 0000
TMRxISTS	0x10	0x0000 0000
TMRx_SW EVT	0x14	0x0000 0000
TMRx_CM1	0x18	0x0000 0000
TMRx_CCTRL	0x20	0x0000 0000
TMRx_CVAL	0x24	0x0000 0000
TMRx_DIV	0x28	0x0000 0000
TMRx_PR	0x2C	0x0000 0000
TMRx_RPR	0x30	0x0000 0000
TMRx_C1DT	0x34	0x0000 0000
TMRx_BRK	0x44	0x0000 0000
TMRx_DMACTRL	0x48	0x0000 0000
TMRx_DMADT	0x4C	0x0000 0000
TMRx_OPT2	0x60	0x0000 0001
TMRx_CHSEL	0x68	0x0000 0000

14.3.5.1 TMR10和TMR11控制寄存器1 (TMRx_CTRL1)

域	简称	复位值	类型	功能
位 31: 12	保留	0x00	resd	保持默认值。
位 11	OVFIFRMP	0x0	rw	OVFIF 状态位重映射 0: 无重映射; 1: OVFIF 位映射到 TMRx_CVAL 位 31。
位 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。

14.3.5.2 TMR10和TMR11控制寄存器2 (TMRx_CTRL2)

域	简称	复位值	类型	功能
位 31: 10	保留	0x00	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 ($HOEN = 0$)，死区发生后： 0: C1OUT=0; 1: C1OUT=1。
位 7: 4	保留	0x0	resd	保持默认值。

				DMA 请求源 (DMA request source)
位 3	DRS	0x0	rw	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 flash 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。

14.3.5.3 TMR10和TMR11 DMA/中断使能寄存器 (TMRx_IDEN)

域	简称	复位值	类型	功能
位 31: 10	保留	0x00	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。

14.3.5.4 TMR10和TMR11中断状态寄存器 (TMRxISTS)

域	简称	复位值	类型	功能
位 31: 14	保留	0x0	resd	保持默认值。
位 13	SBRKIF	0x0	rw0c	系统刹车中断标记 (System brake interrupt flag) 用于标记系统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 12: 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。 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。
位 5	HALLIF	0x0	rw0c	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 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 OVFS=0、 OVFS=0 时： - 当 TMR1_SWEVE 寄存器的 OVFSWTR=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。

14.3.5.5 TMR10和TMR11软件事件寄存器 (TMRx_SWEVT)

域	简称	复位值	类型	功能
位 31: 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	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； 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。

14.3.5.6 TMR10和TMR11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 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 31: 17	保留	0x000	resd	保持默认值。
位 15: 7	保留	0x000	resd	保持默认值。
位 16, 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。 0000: 断开。断开 C1ORAW 到 C1OUT 的输出； 0001: 当 TMRx_CVAL=TMRx_C1DT 时，设置 C1ORAW 为高。 0010: 当 TMRx_CVAL=TMRx_C1DT 时，设置 C1ORAW 为低。 0011: 当 TMRx_CVAL=TMRx_C1DT 时，切换 C1ORAW 的电平。 0100: 固定 C1ORAW 为低； 0101: 固定 C1ORAW 为高； 0110: 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。 0111: 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； 其他值：保留。 注：除'0000'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		功能
位 31: 8	保留	0x00	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； 0001: 采样频率 $f_{SAMPLING} = f_{CK_INT}$, N=2; 0010: 采样频率 $f_{SAMPLING} = f_{CK_INT}$, N=4; 0011: 采样频率 $f_{SAMPLING} = f_{CK_INT}$, N=8; 0100: 采样频率 $f_{SAMPLING} = f_{DTS}/2$, N=6; 0101: 采样频率 $f_{SAMPLING} = f_{DTS}/2$, N=8; 0110: 采样频率 $f_{SAMPLING} = f_{DTS}/4$, N=6; 0111: 采样频率 $f_{SAMPLING} = f_{DTS}/4$, N=8; 1000: 采样频率 $f_{SAMPLING} = f_{DTS}/8$, N=6; 1001: 采样频率 $f_{SAMPLING} = f_{DTS}/8$, N=8; 1010: 采样频率 $f_{SAMPLING} = f_{DTS}/16$, N=5; 1011: 采样频率 $f_{SAMPLING} = f_{DTS}/16$, N=6; 1100: 采样频率 $f_{SAMPLING} = f_{DTS}/16$, N=8; 1101: 采样频率 $f_{SAMPLING} = f_{DTS}/32$, N=5; 1110: 采样频率 $f_{SAMPLING} = f_{DTS}/32$, N=6; 1111: 采样频率 $f_{SAMPLING} = f_{DTS}/32$, N=8。
位 7: 4	C1DF	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 3: 2	C1IDIV	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。
位 1: 0	C1C	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。

14.3.5.7 TMR10和TMR11通道控制寄存器 (TMRx_CCTRL)

域	简称	复位值	类型	功能
位 31: 4	保留	0x000	resd	保持默认值。 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 3	C1CP	0x0	rw	通道 1 互补使能 (Channel 1 complementary enable) 0: 禁止输出； 1: 使能输出。
位 2	C1CEN	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。
位 1	C1P	0x0	rw	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 OCx 和 OCxN 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	输出禁止 (与定时器断开) CxOUT=0, CxCEN=0
		0	0	1	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	0	CxORAW+极性, CxOUT= CxORAW xor CxP, Cx_EN=1	输出禁止 (与定时器断开) CxOUT=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) CxOUT=CxP, 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=CxP, 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 (与定时器断开) 异步地: CxOUT=CxP , Cx_EN=0 , CxCOUT=CxCP , CxEN=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	关闭状态 (输出使能且为无效电平) 异步地: CxOUT =CxP, Cx_EN=1, CxCOUT=CxCP , CxEN=1; 若时钟存在: 经过一个死区 时间后 CxOUT =C1IOS, CxCOUT=CxCIOS, 假设 CxIOS 与 CxCIOS 并不都对应 CxOUT 和 CxCOUT 的有效电平。	
		1	1	0		
		1	1	1		
		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。

14.3.5.8 TMR10和TMR11计数值寄存器 (TMRx_CVAL)

域	简称	复位值	类型	功能
位 31	OVFIFCPY	0x0	ro	OVFIF 副本 (OVFIF copy) OVFIFRMP 为 0 时, 该位保留; OVFIFRMP 为 1 时, 该位为 OVFIF 位的只读副本。
位 30: 16	保留	0x0000	resd	保持默认值。
位 15: 0	CVAL	0x0000	rw	计数值 (Counter value)

14.3.5.9 TMR10和TMR11预分频器寄存器 (TMRx_DIV)

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	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。

14.3.5.10 TMR10和TMR11周期寄存器 (TMRx_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。

14.3.5.11 TMR10和TMR11周期寄存器 (TMRx_RPR)

域	简称	复位值	类型	功能
位 31: 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。

14.3.5.12 TMR10和TMR11通道1数据寄存器 (TMRx_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。

14.3.5.13 TMR10和TMR11刹车寄存器 (TMRx_BRK)

域	简称	复位值	类型	功能
位 31: 29	保留	0x0	resd	保持默认值。
位 28	BRKBID	0x0	resd	刹车双向 (Brake bidirectional) 0: 关闭双向刹车模式; 1: 开启双像刹车模式。
位 27	保留	0x0	resd	保持默认值。
位 26	BRKRLS	0x0	rw	刹车解除(Brake release) 0: 启动刹车输入; 1: 解除刹车输入。
位 25: 20	保留	0x00	resd	保持默认值。
位 19: 16	BRKF	0x0	rw	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; 0001: 采样频率 $f_{SAMPLING} = f_{CK_INT}$, N=2; 0010: 采样频率 $f_{SAMPLING} = f_{CK_INT}$, N=4;

				0011: 采样频率 $f_{SAMPLING} = f_{CK_INT}$, N=8; 0100: 采样频率 $f_{SAMPLING} = f_{DTS}/2$, N=6; 0101: 采样频率 $f_{SAMPLING} = f_{DTS}/2$, N=8; 0110: 采样频率 $f_{SAMPLING} = f_{DTS}/4$, N=6; 0111: 采样频率 $f_{SAMPLING} = f_{DTS}/4$, N=8; 1000: 采样频率 $f_{SAMPLING} = f_{DTS}/8$, N=6; 1001: 采样频率 $f_{SAMPLING} = f_{DTS}/8$, N=8; 1010: 采样频率 $f_{SAMPLING} = f_{DTS}/16$, N=5; 1011: 采样频率 $f_{SAMPLING} = f_{DTS}/16$, N=6; 1100: 采样频率 $f_{SAMPLING} = f_{DTS}/16$, N=8; 1101: 采样频率 $f_{SAMPLING} = f_{DTS}/32$, N=5; 1110: 采样频率 $f_{SAMPLING} = f_{DTS}/32$, N=6; 1111: 采样频率 $f_{SAMPLING} = f_{DTS}/32$, N=8。 注: 如无法保证刹车输入干净, 建议开启刹车输入滤波。
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: TMRx_BRK: BRKBID、BRKRLS、BRKEN、BRKV、 BRKF、AOEN 和 DTC TMRx_CTRL2: C1IOS 和 C1CIOS 10: 2 级写保护, 除 3 级写保护的内容外, 以下位也受写保护: TMRx_CCTRL: C1P 和 C1CP TMRx_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: TMRx_CM1: C1OCTRL 和 C1OBEN 注: WPC>0 时将无法再次被修改, 直到系统复位。
位 7: 0	DTC	0x00	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，BRKF、AOEN、BRKBID、BRKRLS、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。

14.3.5.14 TMR10和TMR11 DMA控制寄存器 (TMRx_DMCTRL)

域	简称	复位值	类型	功能
位 31: 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,

14.3.5.15 TMR10和TMR11 DMA数据寄存器 (TMRx_DMADT)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0	resd	保持默认值。
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。

14.3.5.16 TMR10和TMR11选项寄存器2 (TMRx_OPT2)

域	简称	复位值	类型	功能
位 31: 12	保留	0x0	rw	保持默认值。
位 11	BRKCMP2P	0x0	rw	BRK CMP2 输入极性 (BRK CMP2 input polarity) 0: 高电平有效； 1: 低电平有效。
位 10	BRKCMP1P	0x0	rw	BRK CMP1 输入极性 (BRK CMP1 input polarity) 0: 高电平有效； 1: 低电平有效。
位 9	BRKINP	0x0	rw	BRK BKIN 输入极性 (BRK BKIN input polarity) 0: 高电平有效； 1: 低电平有效。
位 8: 3	保留	0x0	resd	保持默认值。
位 2	BRKCMP2EN	0x0	rw	BRK CMP2 使能 (BRK CMP2 enable) 0: 禁止 CMP2 输入； 1: 使能 CMP2 输入。
位 1	BRKCMP1EN	0x0	rw	BRK CMP1 使能 (BRK CMP1 enable) 0: 禁止 CMP1 输入； 1: 使能 CMP1 输入。
位 0	BRKINEN	0x1	rw	BRK BKIN 使能 (BRK BKIN enable) 0: 禁止 BKIN 输入； 1: 使能 BKIN 输入。

14.3.5.17 TMR10通道选择寄存器 (TMR10_CHSEL)

域	简称	复位值	类型	功能
位 31: 4	保留	0x0000	resd	保持默认值。
位 3: 0	CH1SEL	0x0	rw	CH1 输入选择 (CH1 input selection) 0000: TMR10_CH1; 0001: CMP1; 0010: CMP2; 其他值: 保留。

14.3.5.18 TMR11通道选择寄存器 (TMR11_CHSEL)

域	简称	复位值	类型	功能
位 31: 4	保留	0x0000	resd	保持默认值。
位 3: 0	CH1SEL	0x0	rw	CH1 输入选择 (CH1 input selection) 0000: TMR11_CH1; 0001: CMP1; 0010: CMP2; 0011: 保留; 0100: 保留; 0101: 保留; 0110: 保留; 0111: 保留; 1000: CLK_OUT; 1001: HEXT_DIV; 1010: RTCCLK; 1011: LEXT; 1100: LICK; 其他值: 保留。

14.4 高级控制定时器 (TMR1)

14.4.1 TMR1简介

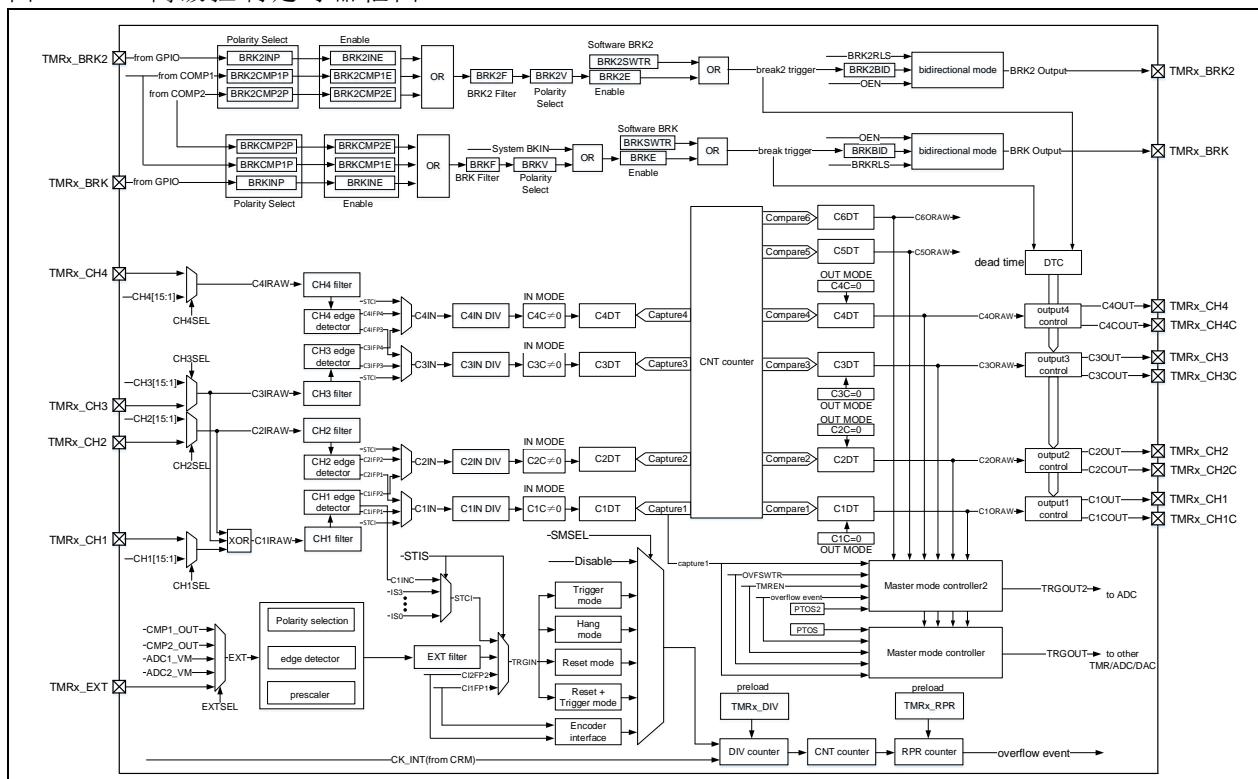
高级定时器 TMR1 包含一个支持向上、向下计数的 16 位计数器、6 个捕获/比较寄存器、6 组独立的通道（通道 5/6 只做内部连接）。可实现嵌入死区、输入捕获、可编程 PWM 输出。

14.4.2 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
- 6组独立通道，通道5和6仅支持支持输出比较；通道1至4支持输入捕获、输出比较、PWM 生成、单周期模式、死区插入
- 通道输入、刹车输入、外部触发输入、内部触发输入支持输入源选择
- PWM模式支持非对称模式和组合模式
- 4组支持互补输出的独立通道
- 支持TMR刹车功能，支持双向刹车功能
- 定时器之间可互联同步
- 支持溢出事件、触发事件、两组刹车输入、通道事件触发中断/DMA
- 支持TMR DMA Burst传输

图 14-72 高级控制定时器框图

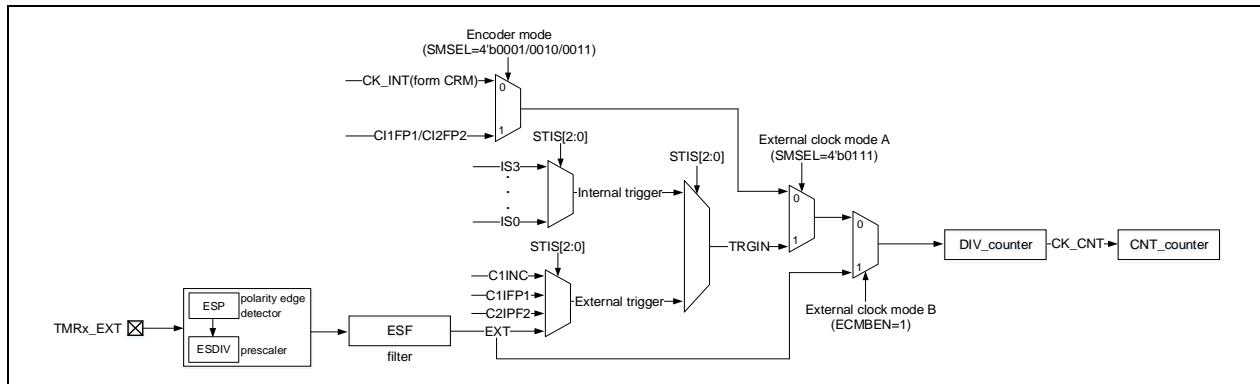


14.4.3 TMR1功能描述

14.4.3.1 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 14-73 计数时钟

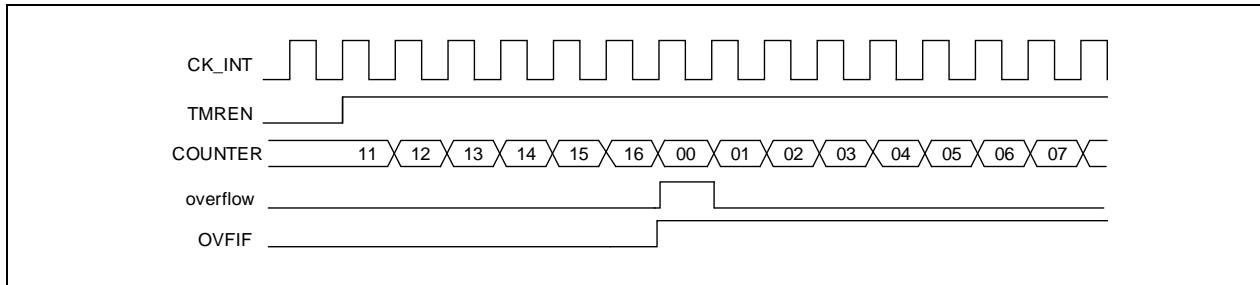


内部时钟 (CK_INT)

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 14-74 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=4'b0111 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)。

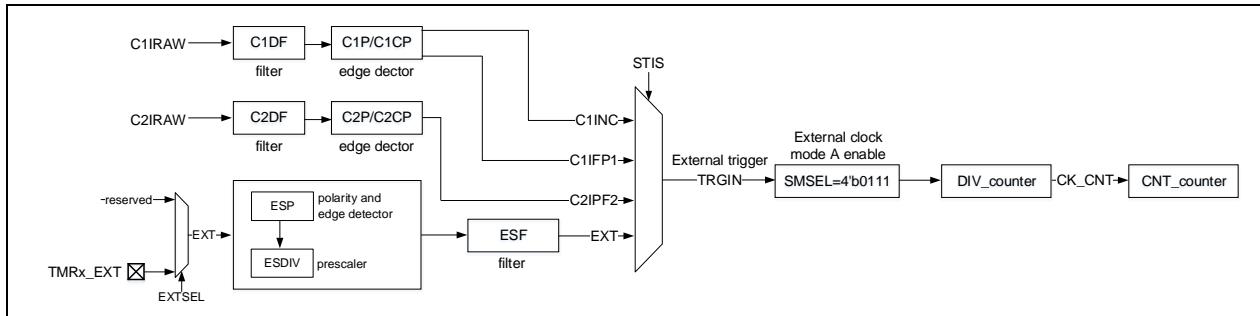
若选择 TRGIN 来源为 TMRx_EXT, 需配置外部信号极性 (TMRx_STCTRL 寄存器 ESP)、外部信号分频 (TMRx_STCTRL 寄存器 ESDIV[1:0]) 和外部信号滤波 (TMRx_STCTRL 寄存器 ESF[3:0])。

- 配置 TMRx_STCTRL 寄存器 STIS[2:0]，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 14-75 外部时钟模式 A 框图



注意：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 14-76 使用外部时钟模式 A 计数，PR=0x32，DIV=0x0

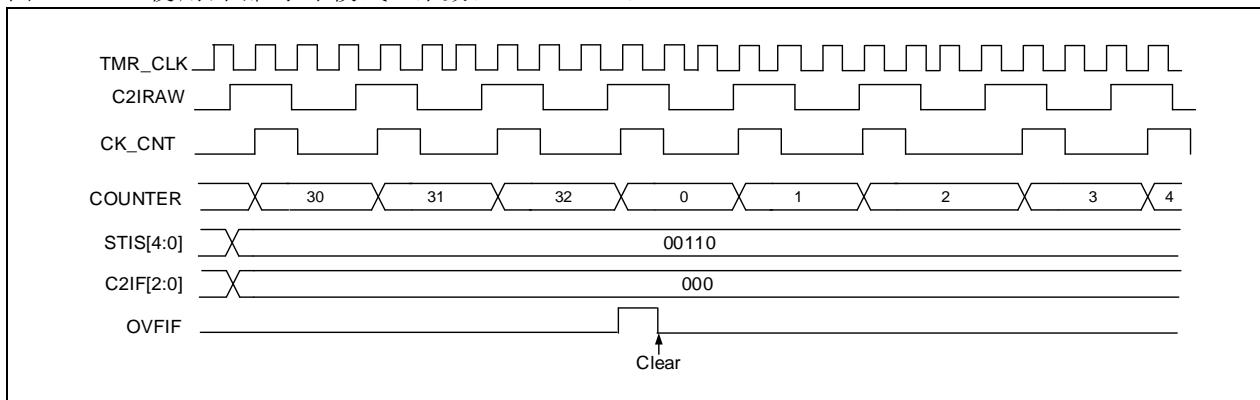
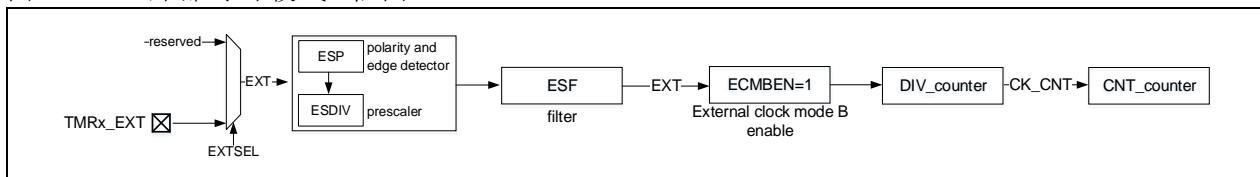
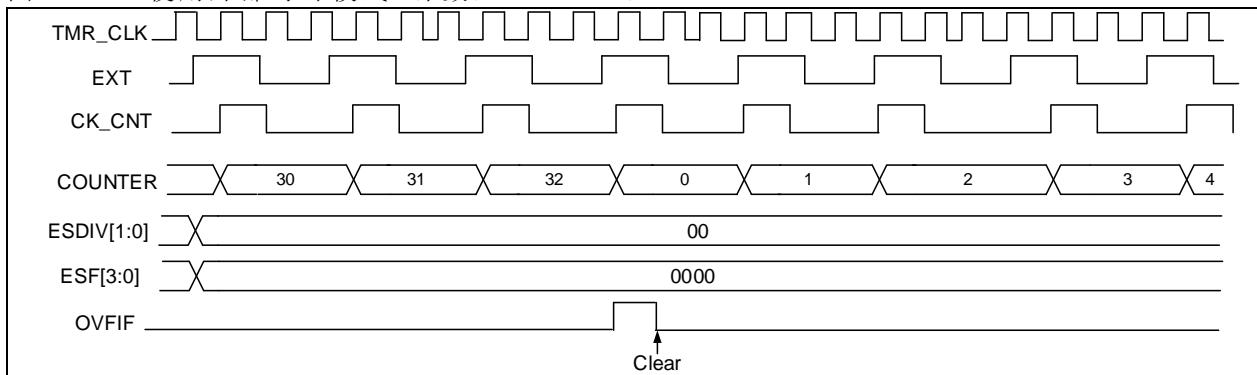


图 14-77 外部时钟模式 B 框图



注意：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 14-78 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。

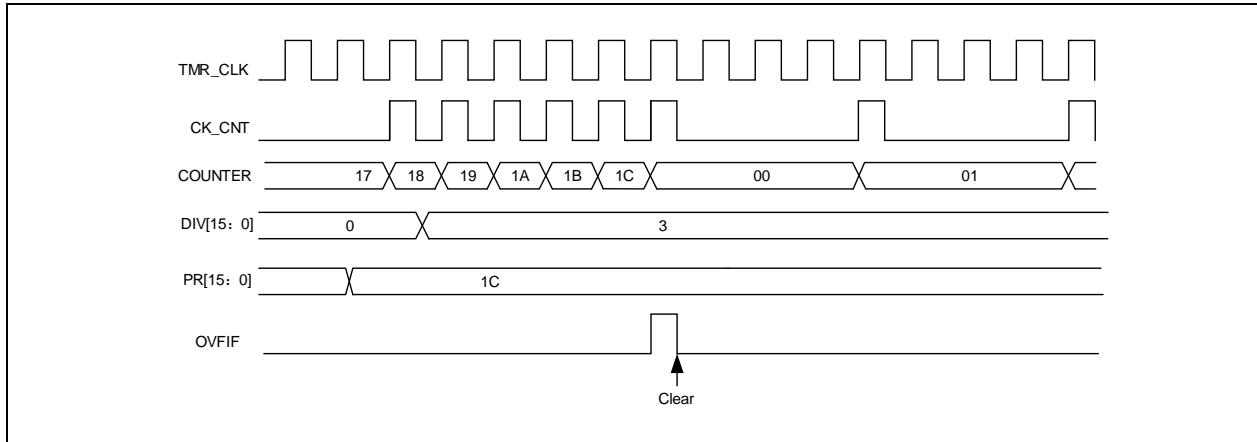
内部触发输入配置流程如下：

- 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[3:0]=4'b0111**，选择外部时钟模式 A。
- 配置 **TMRx_CTRL1** 寄存器 **TMREN** 位，使能 **TMRx** 计数。

表 14-12 TMR1 内部触发连接

次定时器	IS0 (STIS=000)	IS1 (STIS=001)	IS2 (STIS=010)	IS3 (STIS=011)
TMR1	TMR9	-	TMR3	TMR4

图 14-79 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

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。

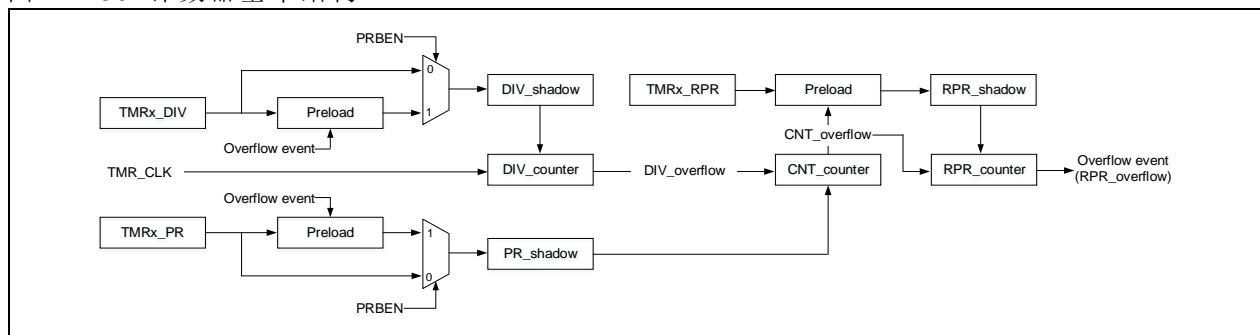
TMRx_DIV 寄存器用于设置计数器计数频率，每 (**DIV[15:0]+1**) 个计数时钟周期，计数器计数一次。和 **TMRx_PR** 寄存器类似，开启周期缓冲功能后，**TMRx_DIV** 寄存器值在溢出事件时更新至它的影子寄存器。

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** 将禁止溢出事件产生。**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 14-80 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 14-81 PRBEN=0时的溢出事件

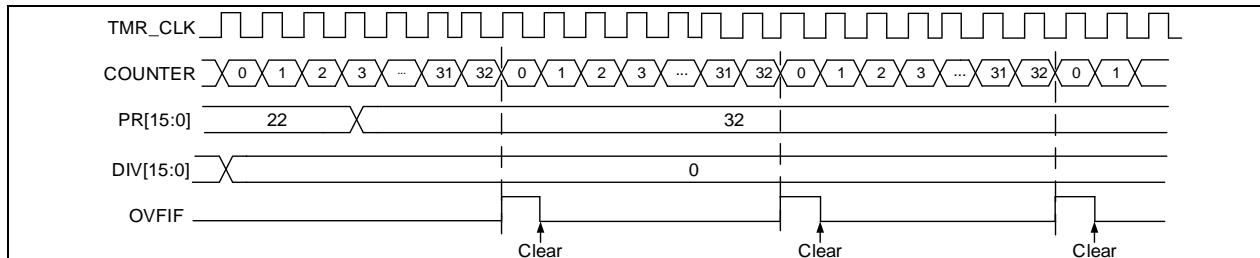
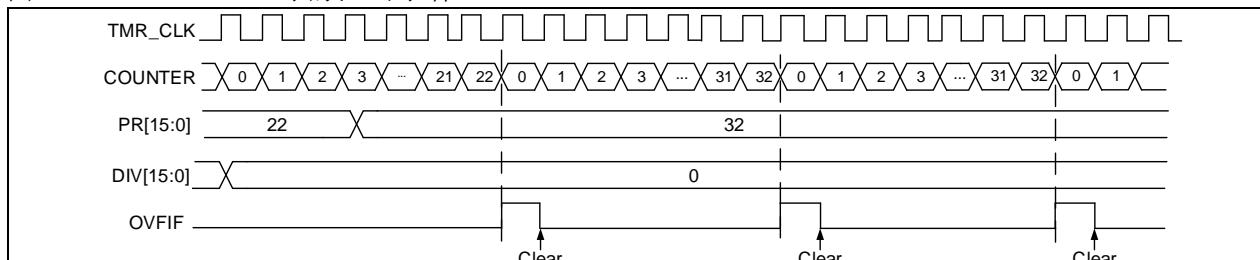


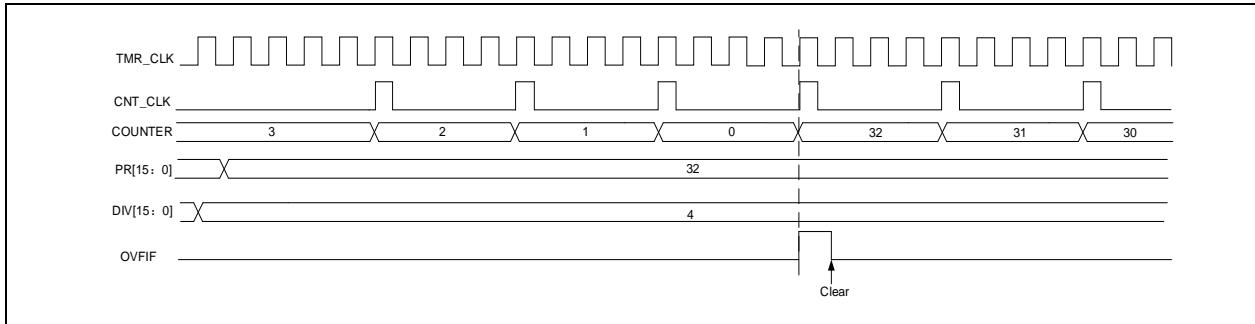
图 14-82 PRBEN=1时的溢出事件



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 14-83 计数器时序图，内部时钟分频因子为4

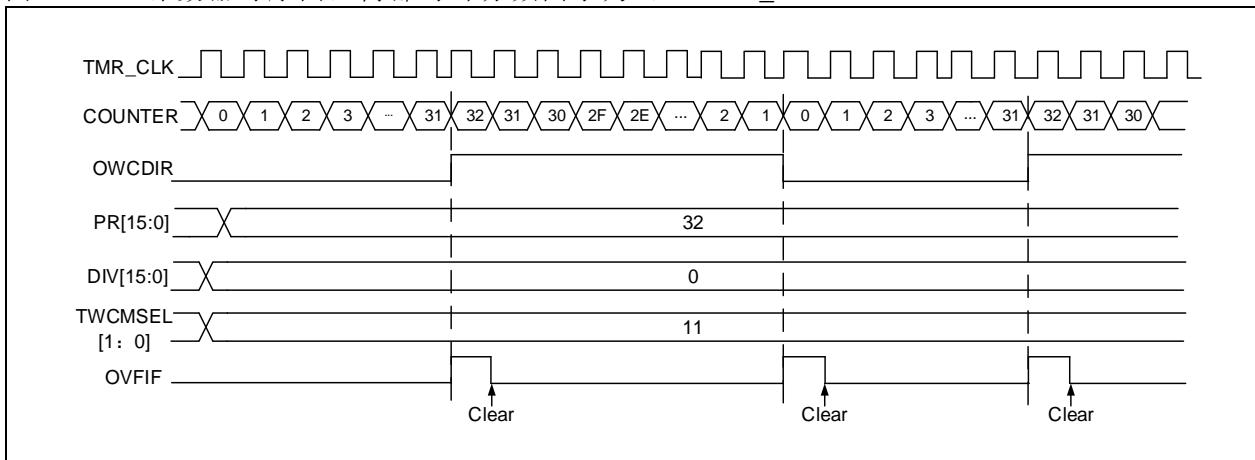


中央双向对齐计数模式

配置 `TMRx_CTRL1` 寄存器 `CMSEL[1:0]` \neq 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 `TWCMSEL[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 (`TWCMSEL[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 (`TWCMSEL[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 (`TWCMSEL[1:0]=2'b11`) 允许 `CxIF` 标志位在计数器向上和向下计数时置起。

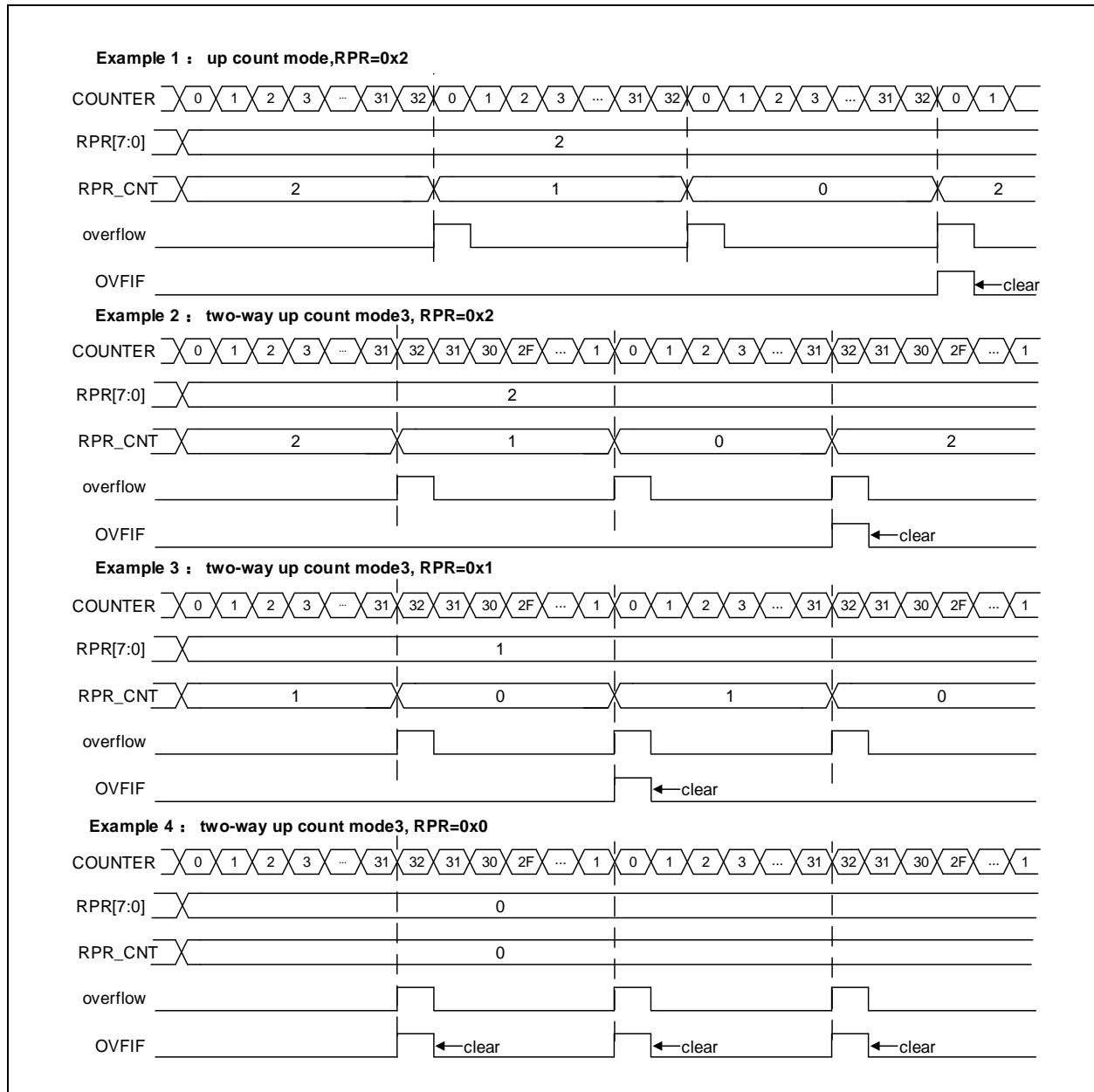
注意：中央双向对齐计数模式下，`OWCDIR` 位为只读位。

图 14-84 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[15:0]+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。

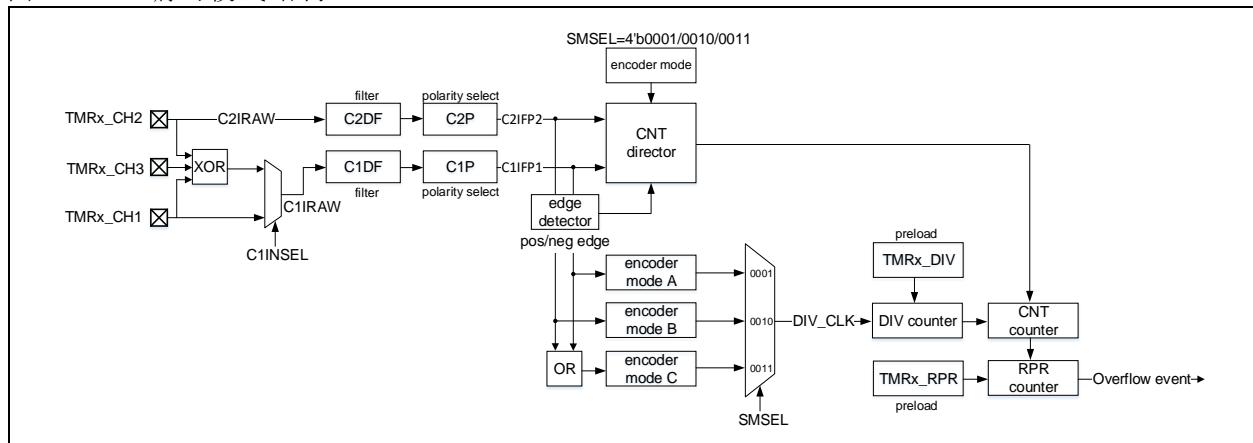
图 14-85 向上计数模式和中央双向对齐计数模式时 OVFIF



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 14-86 编码模式结构



编码器模式 A: SMSEL=4'b0001, 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=4'b0010, 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=4'b0011, 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。

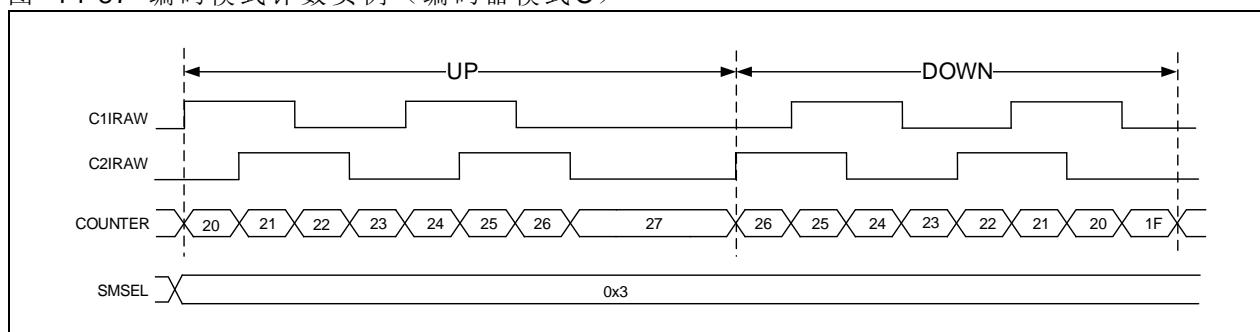
若要使用编码器模式可按下面步骤配置：

- 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[3:0], 选择编码器模式 A (SMSEL=4'b0001)、编码器模式 B (SMSEL=4'b0010) 或编码器模式 C (SMSEL=4'b0011)。
- 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。

表 14-13 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 14-87 编码模式计数实例（编码器模式 C）



14.4.3.3 OVFIF位重映射

TMRx_CTRL1 寄存器中 OVFIFRMP 位强制将溢出中断标记 OVFIF 复制到定时器计数值寄存器 TMRx_CVAL[31]中。这样便可读取计数值的同时读取 OVFIF 来判断溢出中断标记是否置起，有利于简化计算。

14.4.3.4 TMR输入部分

TMR1 拥有 6 个独立通道，除通道 5 和通道 6 外，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[15:0] 经过 CHxSEL 选择和预处理后输出 CxIRAW。配置 C1INSEL 位，选择 C1IRAW 来源是 TMRx_CH1[x] 或是 TMRx_CH1[x]、TMRx_CH2[x]、TMRx_CH3[x] 异或。C2IRAW、C3IRAW、C4IRAW 来源是 TMRx_CH2[x]、TMRx_CH3[x]、TMRx_CH4[x]。
 - 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 来源于 CxIFPx、CyIFPx、STCI。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）；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 14-88 输入/输出通道 1 的主电路

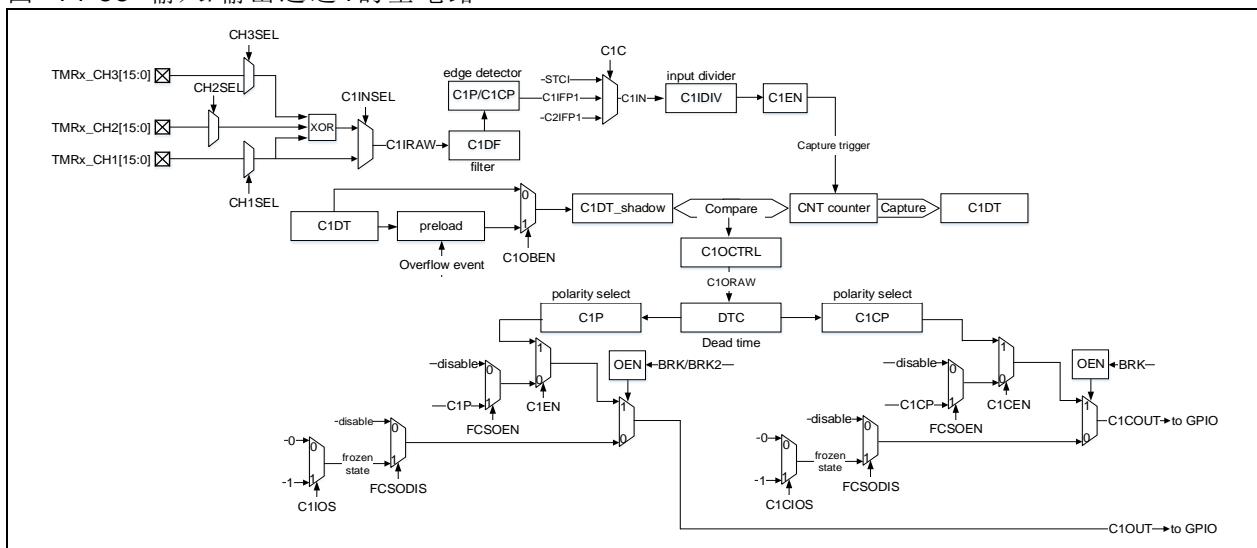
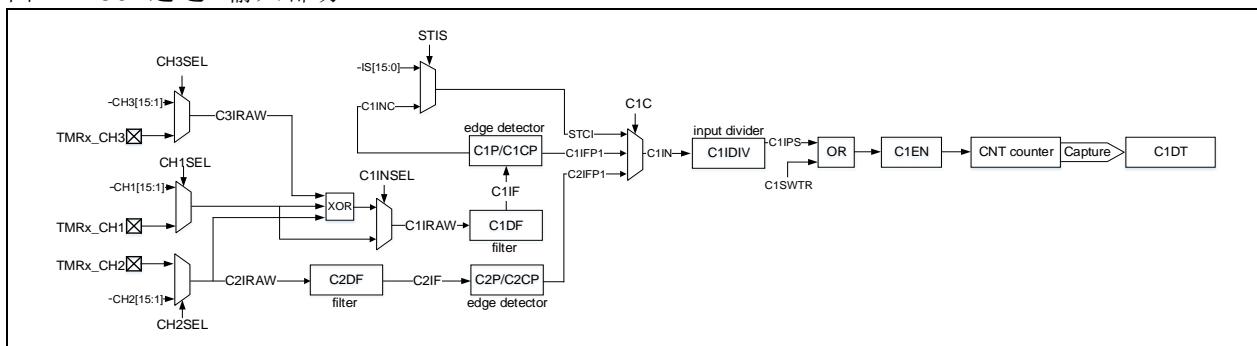


图 14-89 通道1输入部分



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（TMRx_CM1）中的 C1C 位配置为 01，选择 C1IN 作为通道 1 输入。
- 配置 CH1SEL，选择通道 1 输入源。
- 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。

多输入异或

通道 1 的输入端可选择 TMRx_CH1[x]、TMRx_CH2[x] 和 TMRx_CH3[x] 经异或逻辑后输入。将 TMRx_CTRL2 寄存器中的 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[x]，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=4'b0100，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 14-90 PWM 输入模式配置实例

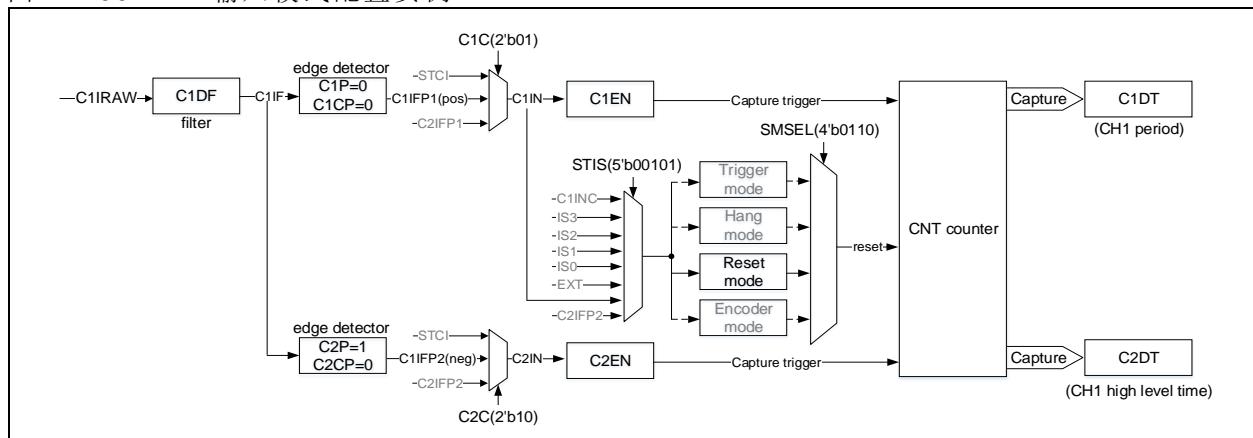
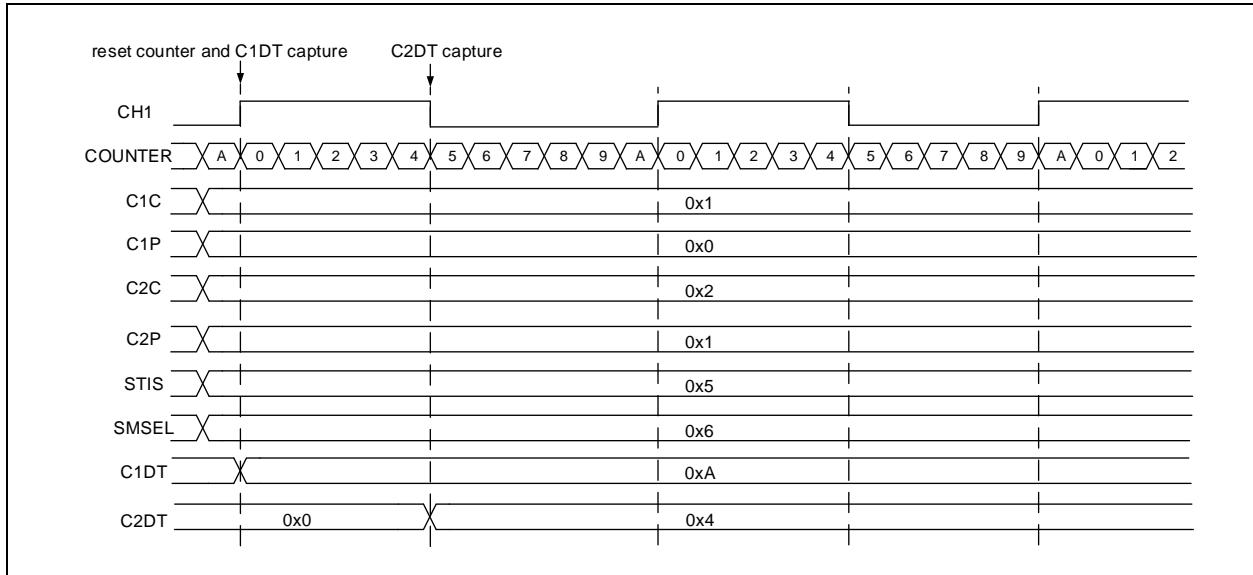


图 14-91 PWM输入模式



14.4.3.5 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。通道 5 和 6 的输出连接至 TRGOUT2，当 TRGO2EN=1 时，TRGOUT2 输出通过 MMS2 位选择，通道 1 至 4 如下图所示：

图 14-92 通道1至4输出部分

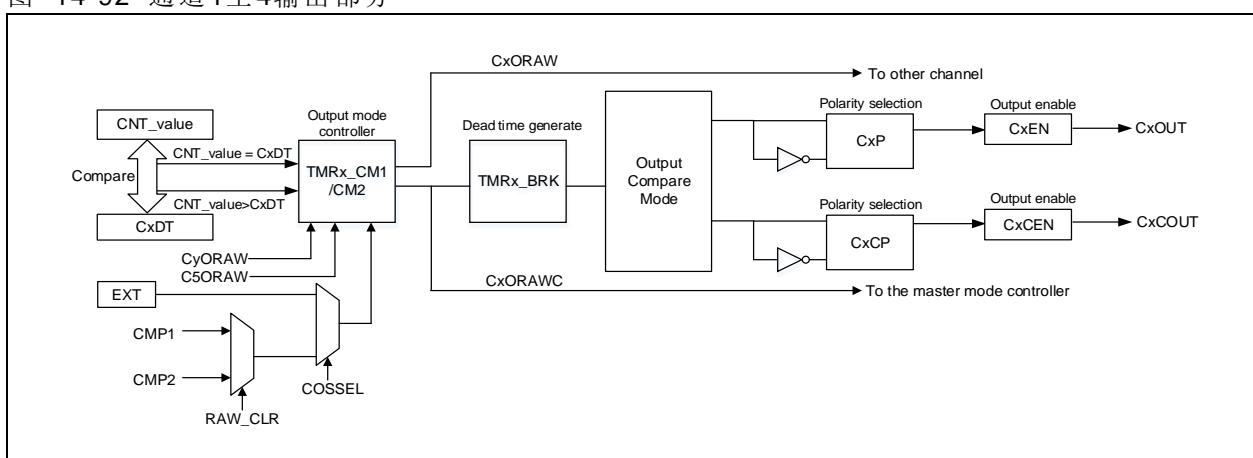
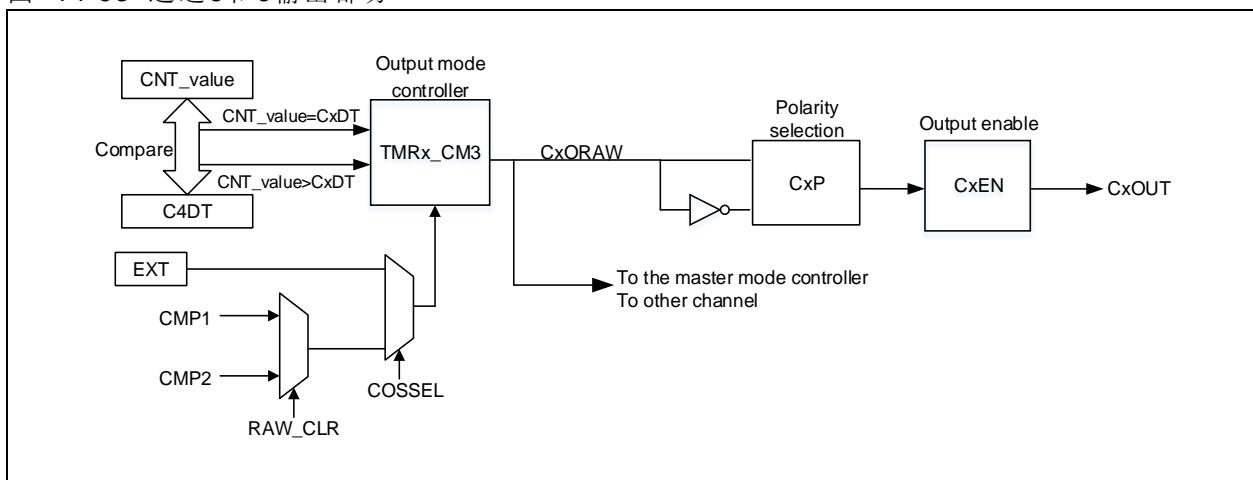


图 14-93 通道5和6输出部分



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 $CxOCTRL[3:0]$ 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 $TMR1_PR$ 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: $CxOCTRL=4'b0110$ 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。图 14-94 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。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。

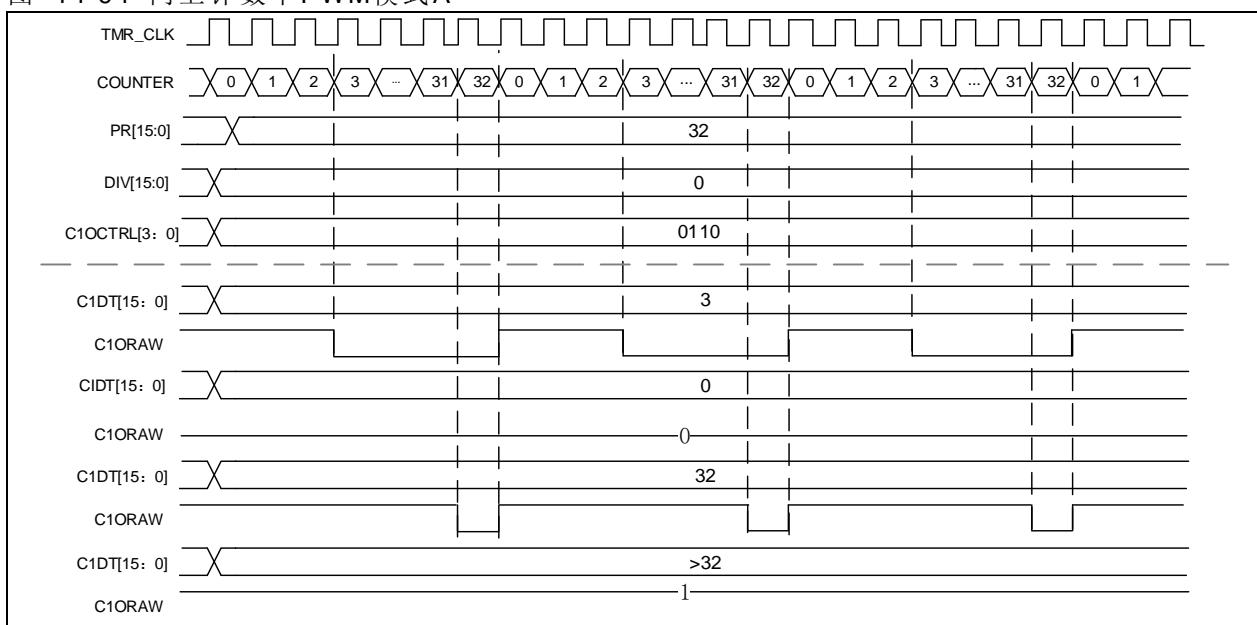
- 配置 $TMRx_CCTRL$ 寄存器 $CxEN$ 位、 $CxCEN$ 位，使能通道输出。

- 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。

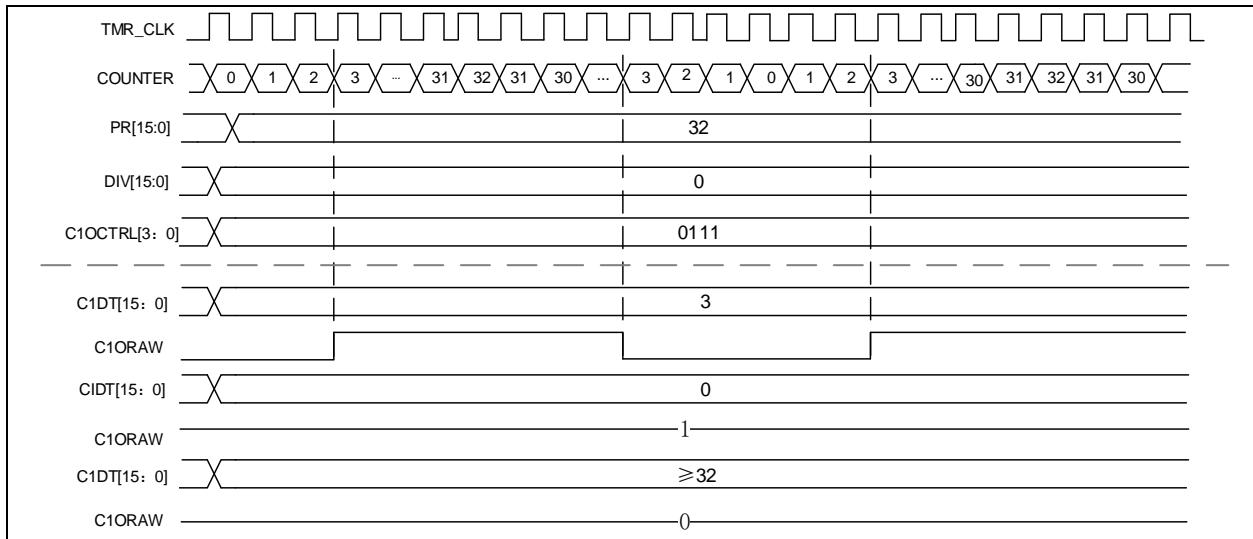
- 配置 $TMRx_CTRL1$ 寄存器 $TMREN$ 位，使能 $TMRx$ 计数。

图 14-94 向上计数下 PWM 模式 A



PWM 模式 B: $CxOCTRL=4'b0111$ 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。图 14-95 展示了计数器中央双向对齐计数与 PWM 模式 B 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。

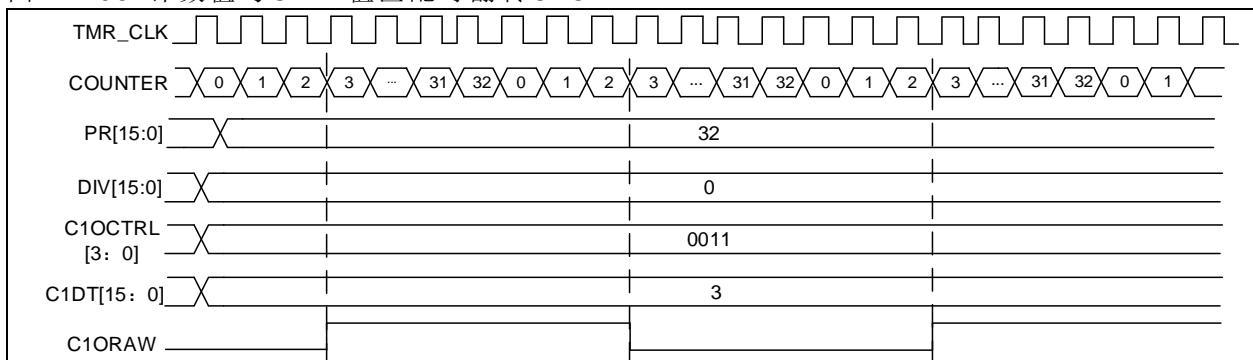
图 14-95 中央双向对齐计数下 PWM 模式 B



强制输出模式: CxOCTRL=4'b0100/0101 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。

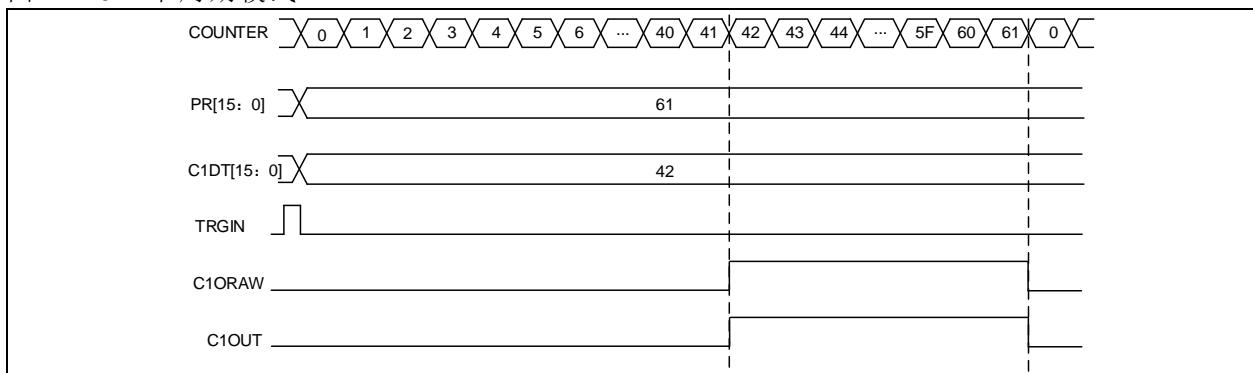
输出比较模式: CxOCTRL=4'b0001/0010/0011 时，开启输出比较模式。此时，当计数值与 CxDT 值匹配时，CxORAW 强制输出高电平 (CxOCTRL=4'b0001)、低电平 (CxOCTRL=4'b0010) 或进行电平翻转 (CxOCTRL=4'b0011)。图 14-96 展示了输出比较模式 (翻转) 的例子，C1DT=0x3，当计数值等于 0x3 时，输出电平 C1OUT 被翻转。

图 14-96 计数值与 C1DT 值匹配时翻转 C1ORAW



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。图 14-97 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 14-97 单周期模式

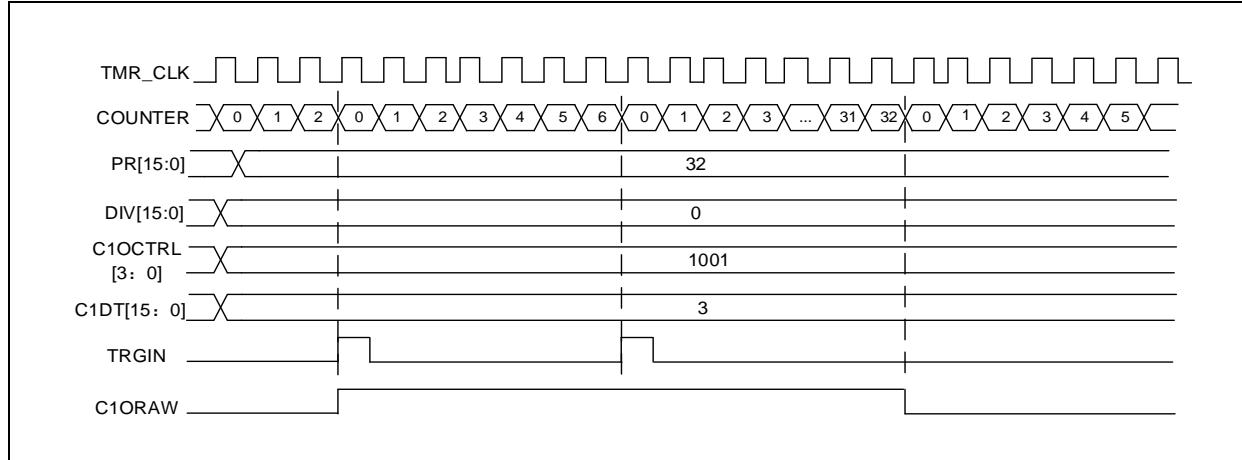


可再触发单周期模式 A: CxOCTRL=4'b1000 时, 开启可再触发单周期模式 A。向上计数时, 检测到 TRGIN 前, C1ORAW 为高, 检测到 TRGIN 后, CxORAW 以 PWM 模式 A 运行, 下一次溢出事件时, CxORAW 再次为高; 向下计数时, 检测到 TRGIN 前, C1ORAW 为低, 检测到 TRGIN 后, CxORAW 以 PWM 模式 A 运行, 下一次溢出事件时, CxORAW 再次为低; 检测到触发信号时, 输出脉冲, 若在计数周期结束前再次检测到触发信号, 脉冲将延长。需搭配组合复位+触发模式使用, 且不支持中央双向对齐计数模式。

可再触发单周期模式 B: CxOCTRL=4'b1001 时, 开启可再触发单周期模式 B。向上计数时, 检测到 TRGIN 前, C1ORAW 为低, 检测到 TRGIN 后, CxORAW 以 PWM 模式 B 运行, 下一次溢出事件时, CxORAW 再次为低; 向下计数时, 检测到 TRGIN 前, C1ORAW 为高, 检测到 TRGIN 后, CxORAW 以 PWM 模式 B 运行, 下一次溢出事件时, CxORAW 再次为高; 检测到触发信号时, 输出脉冲, 若在计数周期结束前再次检测到触发信号, 脉冲将延长。需搭配组合复位+触发模式使用, 且不支持中央双向对齐计数模式。

图 14-98 展示了可再触发 PWM 模式 B 的例子。

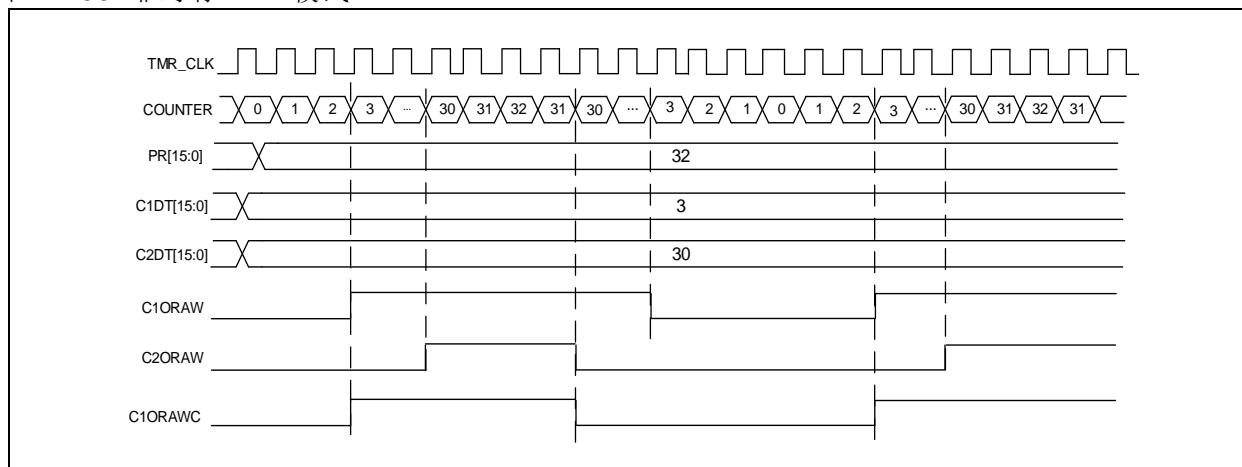
图 14-98 可再触发单周期模式 B



非对称 PWM 模式 A: CxOCTRL=4'b1110 时, 开启非对称 PWM 模式 A。CxORAW 和 CyORAW 以 PWM 模式 A 运行, 向上计数时, CxORAWC 输出 CxORAW; 向下计数时, CxORAWC 输出 CyORAW (对于通道 1, y=2; 对于通道 2, y=1; 对于通道 3, y=4; 对于通道 4, y=3)。若此时将 IASYMEM 位使能, 将开启独立非对称 PWM 模式 A。向上计数时, 输出为计数器与 CxDT 比较产生; 向下计数时, 输出为计数器与 CxDAT 比较产生。

非对称 PWM 模式 B: CxOCTRL=4'b1111 时, 开启非对称 PWM 模式 B。CxORAW 和 CyORAW 以 PWM 模式 B 运行, 向上计数时, CxORAWC 输出 CxORAW; 向下计数时, CxORAWC 输出 CyORAW (对于通道 1, y=2; 对于通道 2, y=1; 对于通道 3, y=4; 对于通道 4, y=3)。若此时将 IASYMEM 位使能, 将开启独立非对称 PWM 模式 B。向上计数时, 输出为计数器与 CxDT 比较产生; 向下计数时, 输出为计数器与 CxDAT 比较产生。图 14-99 展示了非对称 PWM 模式 B 的例子。

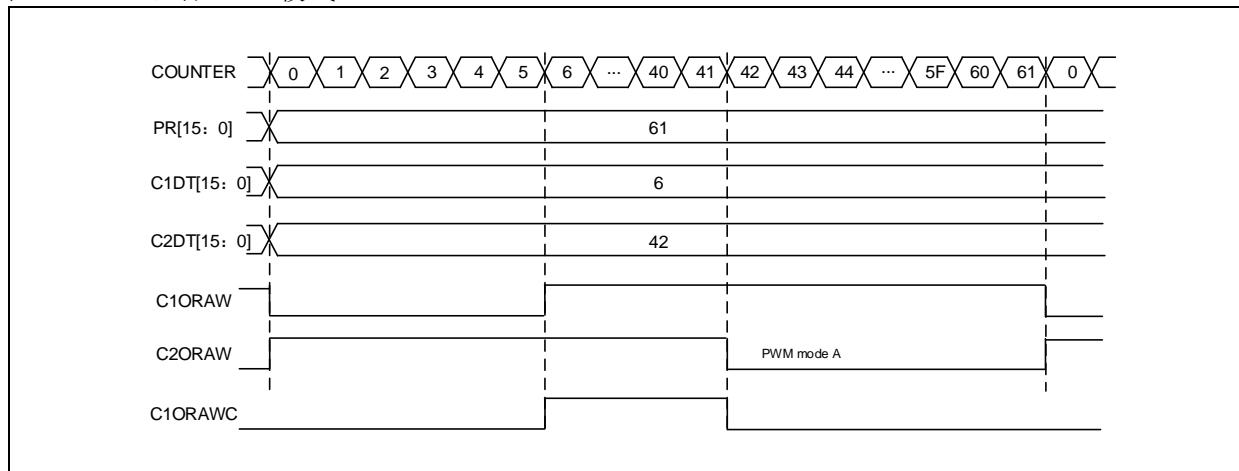
图 14-99 非对称 PWM 模式 B



组合 PWM 模式 A: CxOCTRL=4'b1100 时，开启组合 PWM 模式 A。CxORAW 以 PWM 模式 A 运行，CyORAW 运行模式由通道 y 配置决定，CxORAWC 是 CxORAW 和 CyORAW 逻辑或运算后的输出（对于通道 1, y=2; 对于通道 2, y=1; 对于通道 3, y=4; 对于通道 4, y=3）。

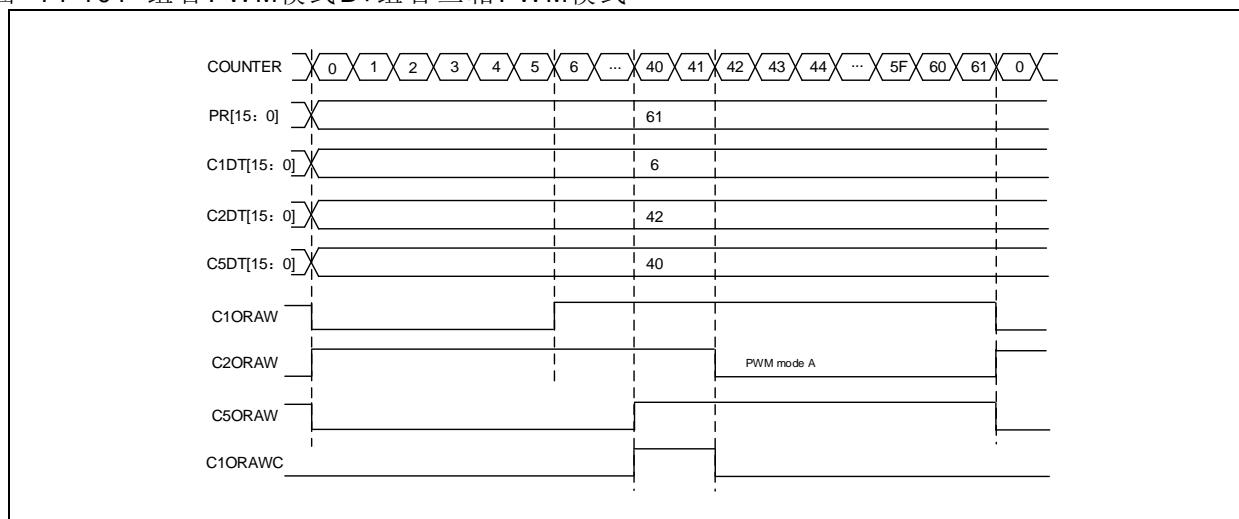
组合 PWM 模式 B: CxOCTRL=4'b1101 时，开启组合 PWM 模式 B。CxORAW 以 PWM 模式 B 运行，CyORAW 运行模式由通道 y 配置决定，CxORAWC 是 CxORAW 和 CyORAW 逻辑与运算后的输出（对于通道 1, y=2; 对于通道 2, y=1; 对于通道 3, y=4; 对于通道 4, y=3）。图 14-100 展示了组合 PWM 模式 B 的例子。

图 14-100 组合 PWM 模式 B



组合三相 PWM 模式: CC5Cx 位置 1，开启组合三相 PWM 模式。CxORAWC 是 CxORAWC 和 C5ORAW 逻辑与后的结果 (x=1~3)。图 14-101 展示了组合 PWM 模式 B+组合三相 PWM 模式的例子。

图 14-101 组合 PWM 模式 B+组合三相 PWM 模式



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。

注意: 非对称 PWM 模式 A/B、组合 PWM 模式 A/B、组合三相 PWM 模式须在中央对齐计数模式下工作。

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 和 TRGOUT2 信号输出到次定时器，选择信号分别为 TMRxCTRL2 寄存器 PTOS 位和 PTO2S 位。

TRGOUT 信号可选择如下信号源：

- PTOS=3'b000, TRGOUT 输出软件溢出事件（TMRx_SWEVT 寄存器 OVFSWTR 位）或复位事件。
- 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 C1ORAWC 信号。
- PTOS=3'b101, TRGOUT 输出 C2ORAWC 信号。
- PTOS=3'b110, TRGOUT 输出 C3ORAWC 信号。
- PTOS=3'b111, TRGOUT 输出 C4ORAWC 信号。

TRGOUT2 可选择如下信号源：

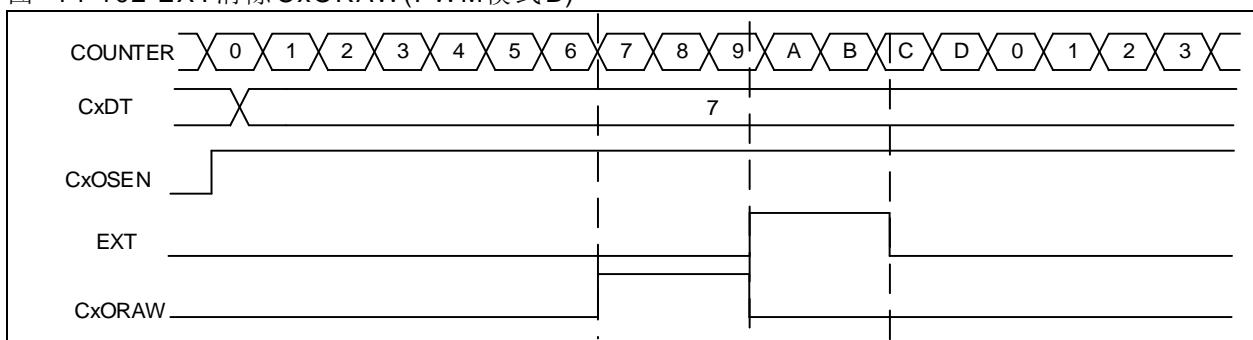
- PTO2S=4'b0000, TRGOUT2 输出软件溢出事件（TMRx_SWEVT 寄存器 OVFSWTR 位）或复位事件。
- PTO2S=4'b0001, TRGOUT2 输出计数器使能信号。
- PTO2S=4'b0010, TRGOUT2 输出计数器溢出事件。
- PTO2S=4'b0011, TRGOUT2 输出捕获、比较事件。
- PTO2S=4'b0100, TRGOUT2 输出 C1ORAWC 信号。
- PTO2S=4'b0101, TRGOUT2 输出 C2ORAWC 信号。
- PTO2S=4'b0110, TRGOUT2 输出 C3ORAWC 信号。
- PTO2S=4'b0111, TRGOUT2 输出 C4ORAWC 信号。
- PTO2S=4'b1000, TRGOUT2 输出 C5ORAW 信号。
- PTO2S=4'b1001, TRGOUT2 输出 C6ORAW 信号。
- PTO2S=4'b1010, TRGOUT2 输出 C4ORAWC 上升沿或下降沿信号。
- PTO2S=4'b1011, TRGOUT2 输出 C6ORAW 上升沿或下降沿信号。
- PTO2S=4'b1100, TRGOUT2 输出 C4ORAWC 或 C6ORAW 上升沿信号。
- PTO2S=4'b1101, TRGOUT2 输出 C4ORAWC 上升沿或 C6ORAW 下降沿信号。
- PTO2S=4'b1110, TRGOUT2 输出 C5ORAW 或 C6ORAW 上升沿信号。
- PTO2S=4'b1111, TRGOUT2 输出 C5ORAW 上升沿或 C6ORAW 下降沿信号。

CxORAW 信号清除

将 CxOSEN 位置 1 后，指定通道的 CxORAW 信号由 EXT 或比较器输出的高电平清 0，COSSEL 位用于选择清除信号来源是 EXT 或是比较器输出，若选择比较器输出，CORAW_OFF 位用于选择清除信号是来自比较器 1 或是比较器 2 输出。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 14-102 EXT 清除 CxORAW(PWM 模式 B)



死区插入

高级定时器通道 1 至 4 包含一组反向通道输出，通过 CxCEN 使能，通过 CxCP 配置极性。

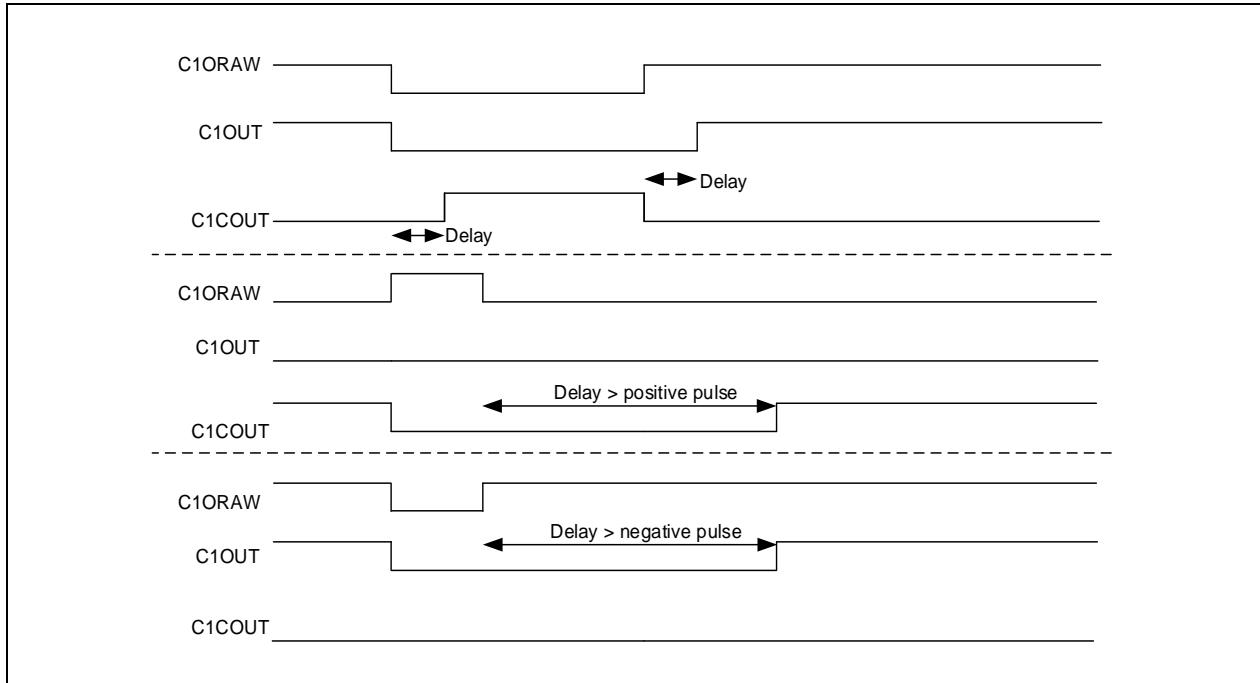
当转换为 IDLEF 状态，即 OEN 下降到 0，死区被激活。

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；CxCOUP 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 C1COUP 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 14-103 带死区插入的互补输出



14.4.3.6 TMR刹车功能

高级定时器支持两路刹车输入（BRK 和 BRK2），BRK2 功能开启前必须配置 BRK2 刹车滤波，开启刹车功能后（BRKEN 或 BRK2EN 位置 1），CxOUT 和 CxCOUP 由 OEN、FCSODIS、FCSOEN、CxIOS 和 CxCIOS 共同控制。但 CxOUT 和 CxCOUP 输出总是不能同时处于有效电平上的。

BRK 和 BRK2 来源包括刹车输入引脚、比较器 1 输出、比较器 2 输出以及软件刹车请求(BRKSWTR 和 BRK2SWTR)，BRK 来源还包括系统刹车输入(SBRK)。BRK (BRK2) 极性由 BRKV (BRK2V) 位控制，比较器输出刹车信号极性由 BRKCMPPxP (BRK2CMPPxP) 位控制；刹车输入引脚极性由 BRKxINP 位控制。来自刹车输入引脚通过 BRKINEN (BRK2INEN) 使能，比较器 1 或 2 输出通过 BRKCMPP1EN 和 BRKCMPP2EN (BRK2CMPP1EN 和 BRK2CMPP2EN) 使能。

通过 BRKIF、BRK2IF 和 SBIF 可确认刹车触发信号的来源是 BRK、BRK2 或 SBRK。

区别于 BRK，BRK2 只能禁止 PWM 输出，而不能将输出设定为预设的安全状态，BRK2 必须在 FCSOEN=FCSODIS=1 时使用。BRK 优先级高于 BRK2。

BRK 和 BRK2 支持双向刹车功能，将 BRKBID (BRK2BID) 置 1 启用此功能，另外需要将 I/O 配置为低电平有效的开漏模式，当刹车事件有效时，刹车 I/O 被强制为 0，用来告知外部器件 TMR 已处于刹车状态。将 BRKRLS (BRK2RLS) 位置 1，会释放刹车输出，避免刹车 I/O 强制为 0 后，TMR 一直处于刹车状态。

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 tmr_clk 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 (BRK2IF) 也不能被清除。

图 14-104 TMR 刹车电路

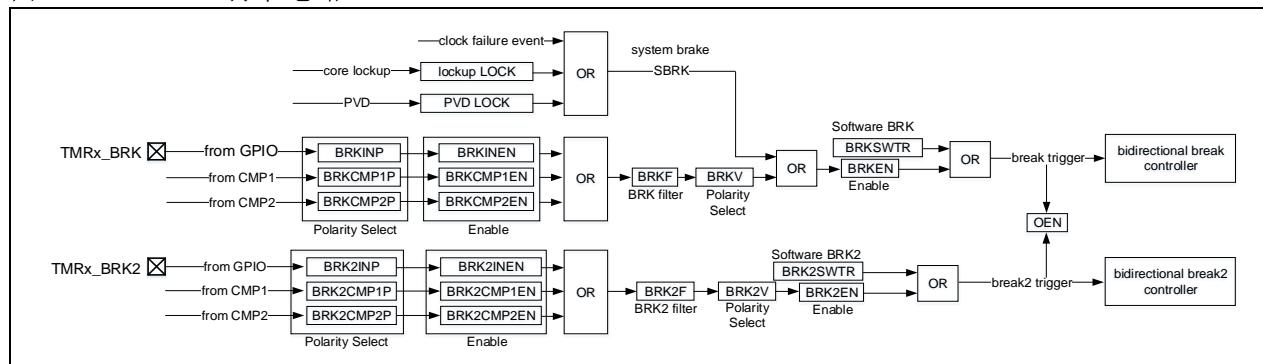


图 14-105 TMR 输出控制

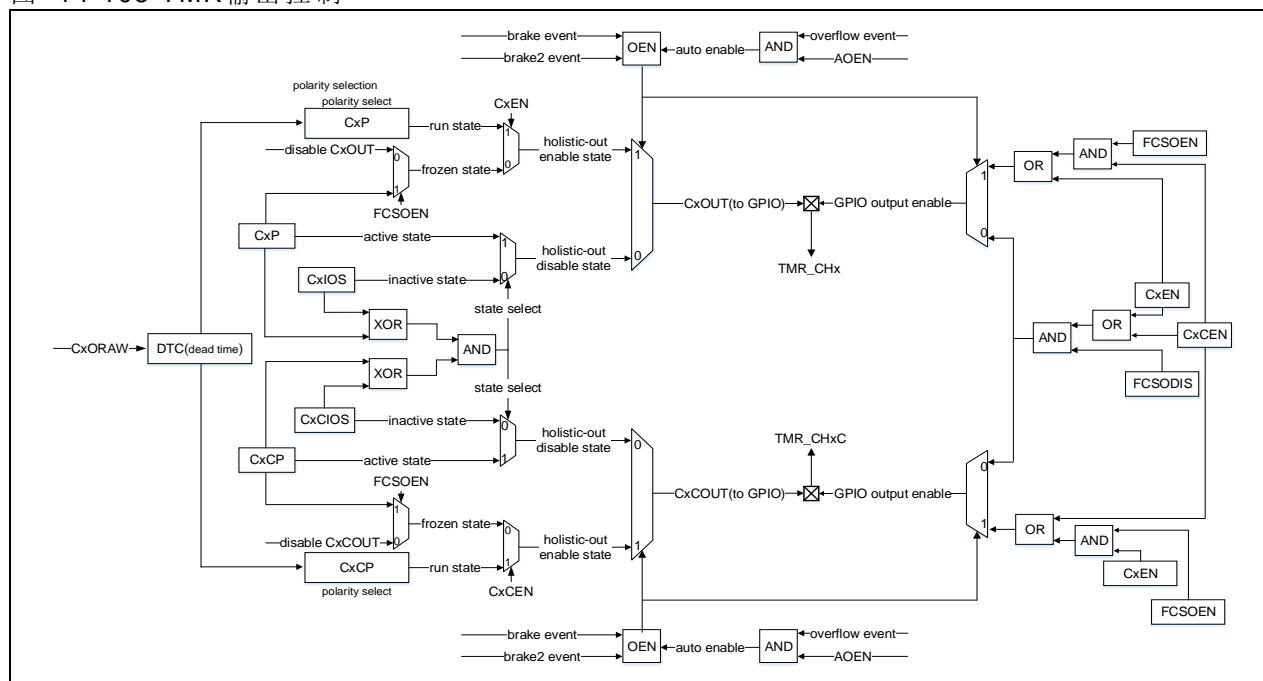
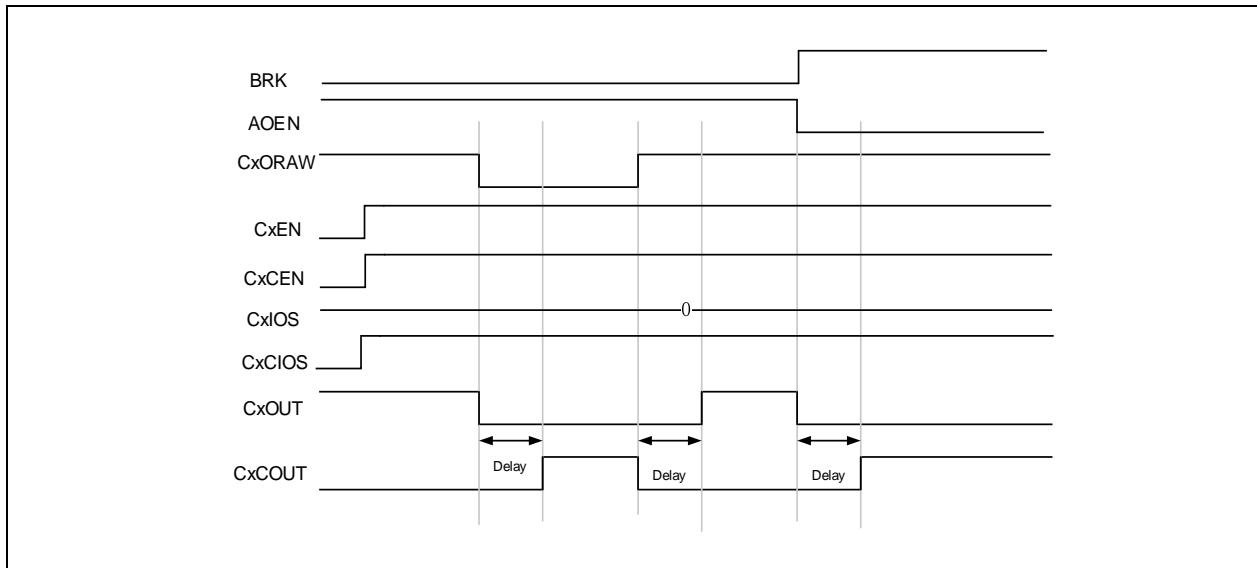


图 14-106 TMR停止功能的例子



14.4.3.7 TMR同步

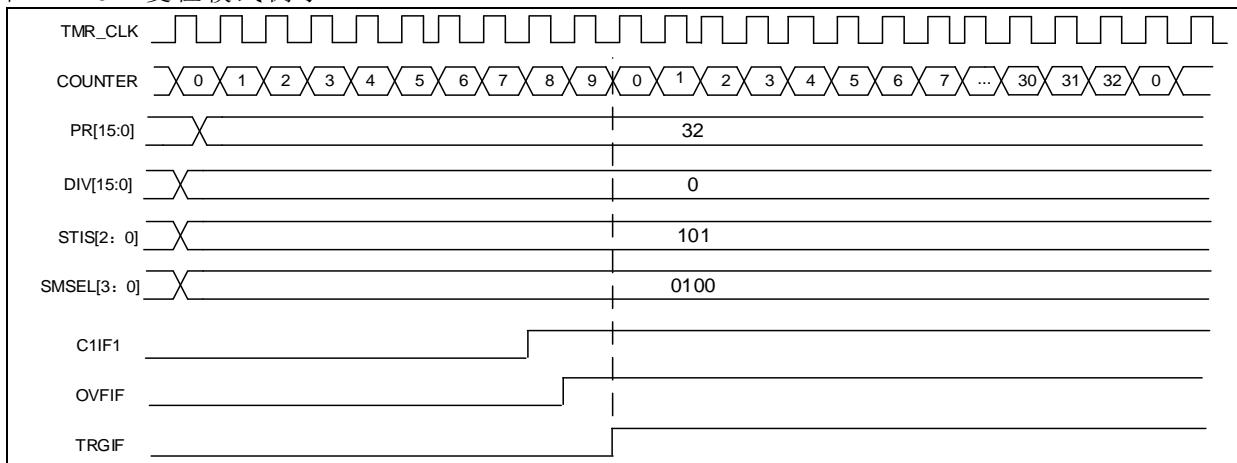
主次定时器之间可由内部连接信号进行同步。主定时器可由 PTOS[2: 0]位选择主定时器输出，即同步信息；次定时器由 SMSEL[3: 0]位选择从模式，即次定时器的工作模式。

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。

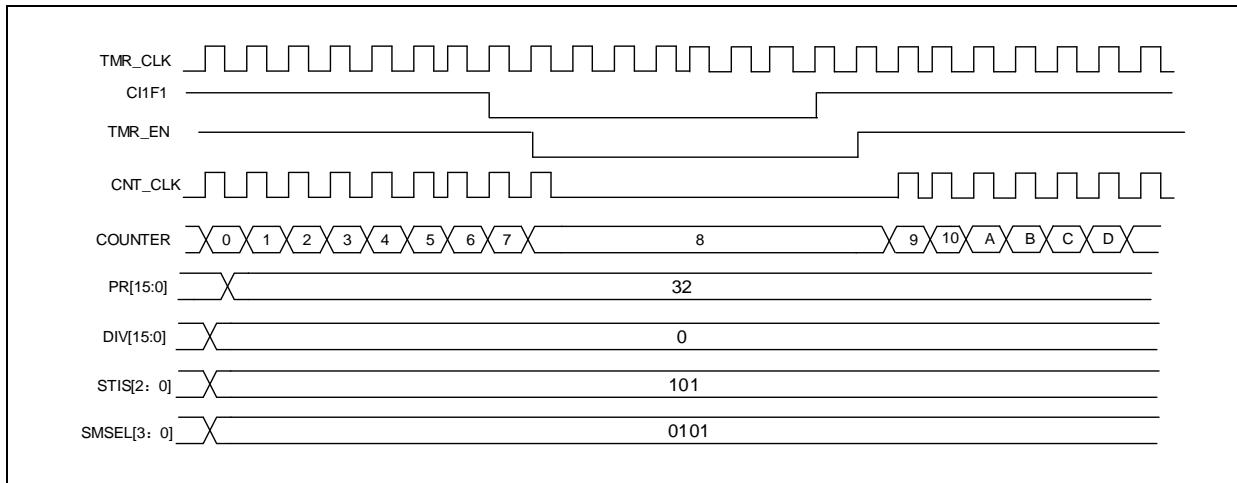
图 14-107 复位模式例子



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。

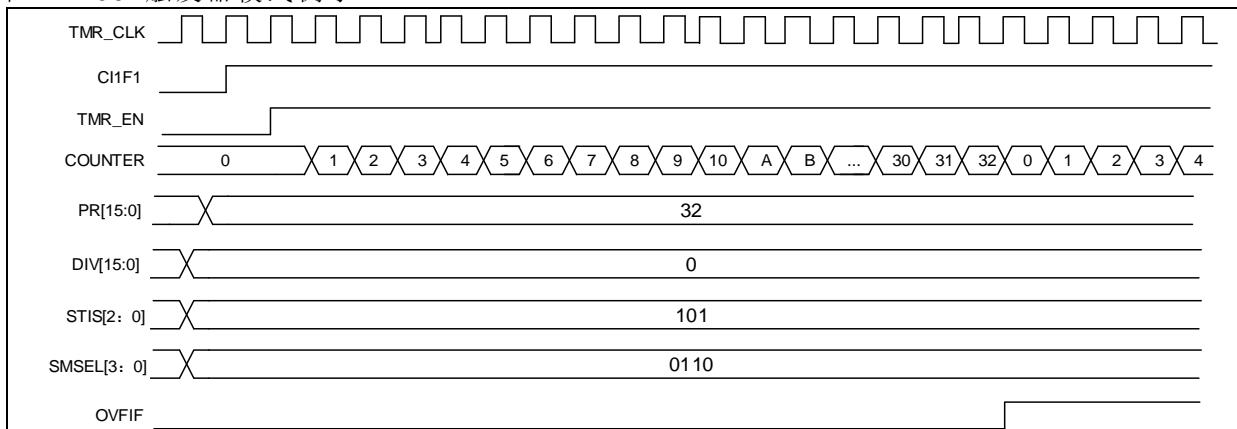
图 14-108 挂起模式下例子



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 14-109 触发器模式例子



从模式：组合复位+触发模式

计数器将在选中的触发输入上升沿初始化计数器，并产生溢出事件，需搭配可再触发单周期模式使用，详见可再触发单周期模式部分描述。

14.4.3.8 TMR DMA

TMR 中有溢出事件 DMA 请求、触发事件 DMA 请求、霍尔事件 DMA 请求以及通道事件 DMA 请求，通过使能 TMRx_IDEN 寄存器中相应位，开启 DMA 请求功能，产生相应事件时触发输出 DMA 请求给到 DMA 外设。

TMR DMA Burst 功能

TMR 还支持 TMR DMA Burst 功能，通过 TMRx_IDEN 寄存器使能某个事件的 DMA 请求，触发 DMA 改写多个 TMR 连续寄存器。

以溢出事件触发 TMR DMA Burst 功能为例，配置如下：

- 配置 TMRx_IDEN 寄存器中 OVFDEN 位，使能溢出事件触发 DMA 请求；
- 配置 TMRx_DMACTRL 寄存器中 DTB 位设置 Burst 传输次数；
- 配置 TMRx_DMACTRL 寄存器中 ADDR 位设置 Burst 传输的起始地址；
- 使能计数器。

以上配置的 TMR DMA Burst 传输流程如下：

当发生溢出事件时，TMR 将会产生溢出事件的 DMA 请求给到 DMA，DMA 根据请求将数据写入 TMRx_DMADT 寄存器，在 TMR 内部则会将 DMADT 位的数据写入 Burst 传输起始地址寄存器，并发送 ACK 信号给到 TMR，TMR 接收到 ACK 信号后，将清除当前 DMA 请求；TMR 检测到 Burst 传输并未全

部完成，会重新置起溢出事件 DMA 请求给到 DMA，DMA 根据请求将数据再次写入 TMRx_DMADT 寄存器，在 TMR 内部则会将 DMADT 寄存器的数据写入 Burst 传输起始地址+0x4 地址寄存器，并发送 ACK 信号给到 TMR；以此往复，直到 Burst 传输最后一次操作时，检测到此时 Burst 传输已全部完成，溢出事件 DMA 请求信号将不会再被重新置起，直到下一次新的溢出事件到来。

注意：使用 TMR DMA Burst 功能时，起始地址到结束地址这个区间，不应有空寄存器以及不应包含 TMRx_DMACTRL 和 TMRx_DMADT 寄存器。

14.4.3.9 调试模式

当微控制器进入调试模式（Cortex®-M4F 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。

14.4.4 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32位可寻址（编址）空间

表 14-14 TMR1寄存器图和复位值

寄存器简称	基址偏移量	复位值
TMR1_CTRL1	0x00	0x0000 0000
TMR1_CTRL2	0x04	0x0000 0000
TMR1_STCTRL	0x08	0x0000 0000
TMR1_IDEN	0x0C	0x0000 0000
TMR1ISTS	0x10	0x0000 0000
TMR1_SWEVT	0x14	0x0000 0000
TMR1_CM1	0x18	0x0000 0000
TMR1_CM2	0x1C	0x0000 0000
TMR1_CCTRL	0x20	0x0000 0000
TMR1_CVAL	0x24	0x0000 0000
TMR1_DIV	0x28	0x0000 0000
TMR1_PR	0x2C	0x0000 0000
TMR1_RPR	0x30	0x0000 0000
TMR1_C1DT	0x34	0x0000 0000
TMR1_C2DT	0x38	0x0000 0000
TMR1_C3DT	0x3C	0x0000 0000
TMR1_C4DT	0x40	0x0000 0000
TMR1_BRK	0x44	0x0000 0000
TMR1_DMACTRL	0x48	0x0000 0000
TMR1_DMADT	0x4C	0x0000 0000
TMR1_OPT1	0x50	0x0000 0000
TMR1_CM3	0x54	0x0000 0000
TMR1_C5DT	0x58	0x0000 0000
TMR1_C6DT	0x5C	0x0000 0000
TMR1_OPT2	0x60	0x0000 0001
TMR1_OPT3	0x64	0x0000 0001
TMR1_CHSEL	0x68	0x0000 0000
TMR1_AC1DT	0x80	0x0000 0000
TMR1_AC2DT	0x84	0x0000 0000
TMR1_AC3DT	0x88	0x0000 0000
TMR1_AC4DT	0x8C	0x0000 0000

14.4.4.1 TMR1控制寄存器1 (TMR1_CTRL1)

域	简称	复位值	类型	功能
位 31: 12	保留	0x00	resd	保持默认值。
位 11	OVFIFRMP	0x0	rw	OVFIF 状态位重映射 0: 无重映射; 1: OVFIF 位映射到 TMRx_CVAL 位 31。
位 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: 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。

14.4.4.2 TMR1控制寄存器2 (TMR1_CTRL2)

域	简称	复位值	类型	功能
位 31	TRGOUT2EN	0x0	rw	TRGOUT2 输出使能 (TRGOUT2 enable) 0: 禁止 1: 开启
位 30	IASYMEN	0x0	rw	独立非对称 PWM 模式使能 (Independently asymmetric PWM mode enable) 用于选择非对称 PWM 模式 A、B 向下计数时，用于产生 CxORAW 信号的通道数据寄存器。 0: 非对称 PWM 模式递减计数时，CxORAW 使用 CxDT 寄存器匹配； 1: 非对称 PWM 模式递减计数时，CxORAW 使用 CxADT 寄存器匹配。
位 29: 24	保留	0x0000	resd	保持默认值
位 23: 20	PTO2S	0x0	rw	主定时器输出信号 2 选择 (Primary TMR output2 selection) TMRx 输出到次定时器的信号 2 选择： 0000: 软件溢出或复位； 0001: 使能； 0010: 溢出； 0011: 比较脉冲； 0100: C1ORAW 信号； 0101: C2ORAW 信号； 0110: C3ORAW 信号； 0111: C4ORAW 信号； 1000: C5ORAW 信号； 1001: C6ORAW 信号； 1010: C4ORAW 信号上升沿或下降沿； 1011: C6ORAW 信号上升沿或下降沿； 1100: C4ORAW 信号或 C6ORAW 信号上升沿； 1101: C4ORAW 信号上升沿或 C6ORAW 信号下降沿； 1110: C5ORAW 信号或 C6ORAW 信号的上升沿； 1111: C5ORAW 信号上升沿或 C6ORAW 信号下降沿。
位 19	保留	0x0	resd	保持默认值
位 18	C6IOS	0x0	rw	通道 6 空闲输出状态 (Channel 6 idle output state)
位 17	保留	0x0	resd	保持默认值
位 16	C5IOS	0x0	rw	通道 5 空闲输出状态 (Channel 5 idle output state)
位 15	C4CIOS	0x0	rw	通道 4 互补空闲输出状态 (Channel 4 complementary idle output state)
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)
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	C1INSEL	0x0	rw	C1IN 选择 (C1IN selection) 0: CH1 引脚连到 C1IRAW 输入； 1: CH1、CH2 和 CH3 引脚异或结果连到 C1IRAW 输入。
位 6: 4	PTOS	0x0	rw	主定时器输出信号选择 (Primary TMR output selection)

				TMRx 输出到次定时器的信号选择: 000: 软件溢出或复位; 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 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。

14.4.4.3 TMR1次定时器控制寄存器 (TMR1_STCTRL)

域	简称	复位值	类型	功能
位 31: 17	保留	0x000	resd	保持默认值。
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 14-12。
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: CxORAW_OFF 来自 CMP1 或 CMP2，取决于 CORAW_OFF 位； 1: CxORAW_OFF 来自 EXT。
位 16, 2: 0	SMSEL	0x0	rw	次定时器模式选择 (Subordinate TMR mode selection) 0000: 关闭从模式； 0001: 编码模式 A； 0010: 编码模式 B； 0011: 编码模式 C； 0100: 复位模式 - TRGIN 输入上升沿时，重新初始化计数器； 0101: 挂起模式 - TRGIN 输入高电平时，计数器计数； 0110: 触发模式 - TRGIN 输入上升沿时，产生触发事件； 0111: 外部时钟模式 A - TRGIN 输入上升沿提供时钟； 1000: 组合复位+触发模式 - TRGIN 输入上升沿时，重新初始化计数器，产生溢出事件并启动计数器； 其他值：保留。 注意：编码器模式 A/B/C 配置方法请查看计数模式章节。

14.4.4.4 TMR1 DMA/中断使能寄存器 (TMR1_IDEN)

域	简称	复位值	类型	功能
位 31: 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)
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)
位 2	C2IEN	0x0	rw	通道 2 中断使能 (Channel 2 interrupt enable)
位 1	C1IEN	0x0	rw	通道 1 中断使能 (Channel 1 interrupt enable)
位 0	OVFIEN	0x0	rw	溢出中断使能 (Overflow interrupt enable) 0: 关闭; 1: 开启。

14.4.4.5 TMR1中断状态寄存器 (TMR1ISTS)

域	简称	复位值	类型	功能
位 31: 18	保留	0x0	resd	保持默认值。
位 17	C6IF	0x0	rw0c	通道 6 中断标记 (Channel 6 interrupt flag) 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 16	C5IF	0x0	rw0c	通道 5 中断标记 (Channel 5 interrupt flag) 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 15: 14	保留	0x0	resd	保持默认值。
位 13	SBRKIF	0x0	rw0c	系统刹车中断标记 (System brake interrupt flag) 用于标记系统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 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、CxLEN、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。

14.4.4.6 TMR1软件事件寄存器 (TMR1_SWEVT)

域	简称	复位值	类型	功能
位 31: 9	保留	0x00	resd	保持默认值。
位 8	BRK2SWTR	0x0	wo	软件触发刹车 2 事件 (Brake2 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 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。 注意：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。

14.4.4.7 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 31: 25	保留	0x0	resd	保持默认值。
位 23: 17	保留	0x0	resd	保持默认值。
位 15	C2OSEN	0x0	rw	通道 2 输出开关使能 (Channel 2 output switch enable)
位 24, 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 16, 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。 0000: 断开。断开 C1ORAW 到 C1OUT 的输出; 0001: 当 TMRx_CVAL=TMRx_C1DT 时，设置 C1ORAW 为高。 0010: 当 TMRx_CVAL=TMRx_C1DT 时，设置 C1ORAW 为低。 0011: 当 TMRx_CVAL=TMRx_C1DT 时，切换 C1ORAW 的电平。 0100: 固定 C1ORAW 为低; 0101: 固定 C1ORAW 为高; 0110: 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。

0111: 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。

1000: 可再触发单周期模式 A

– 向上计数时, 检测到 TRGIN 前, C1ORAW 为高, 检测到 TRGIN 后, C1ORAW 以 PWM 模式 A 运行, 下一次溢出事件时, C1ORAW 再次为高;

– 向下计数时, 检测到 TRGIN 前, C1ORAW 为低, 检测到 TRGIN 后, C1ORAW 以 PWM 模式 A 运行, 下一次溢出事件时, C1ORAW 再次为低。

1001: 可再触发单周期模式 B

– 向上计数时, 检测到 TRGIN 前, C1ORAW 为低, 检测到 TRGIN 后, C1ORAW 以 PWM 模式 B 运行, 下一次溢出事件时, C1ORAW 再次为低;

– 向下计数时, 检测到 TRGIN 前, C1ORAW 为高, 检测到 TRGIN 后, C1ORAW 以 PWM 模式 B 运行, 下一次溢出事件时, C1ORAW 再次为高。

1010: 保留;

1011: 保留;

1100: 组合 PWM 模式 A, C1ORAW 以 PWM 模式 A 运行, C1ORAWC 是 C1ORAW 和 C2ORAW 逻辑或运算后的输出;

1101: 组合 PWM 模式 B, C1ORAW 以 PWM 模式 B 运行, C1ORAWC 是 C1ORAW 和 C2ORAW 逻辑与运算后的输出;

1110: 非对称 PWM 模式 A, C1ORAW 以 PWM 模式 A 运行, 向上计数时, C1ORAWC 输出 C1ORAW; 向下计数时, C1ORAWC 输出 C2ORAW;

1111: 非对称 PWM 模式 B, C1ORAW 以 PWM 模式 B 运行, 向上计数时, C1ORAWC 输出 C1ORAW; 向下计数时, C1ORAWC 输出 C2ORAW。

注意:除'0000'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 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。
-----	--------	-----	----	---

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。
--------	-----	-----	----	---

输入模式

域	简称	复位值	类型	功能
位 31: 16	保留	0x0	resd	保持默认值。
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; 0001: 采样频率 $f_{SAMPLING} = f_{CK_INT}$, N=2; 0010: 采样频率 $f_{SAMPLING} = f_{CK_INT}$, N=4; 0011: 采样频率 $f_{SAMPLING} = f_{CK_INT}$, N=8; 0100: 采样频率 $f_{SAMPLING} = f_{DTS}/2$, N=6; 0101: 采样频率 $f_{SAMPLING} = f_{DTS}/2$, N=8; 0110: 采样频率 $f_{SAMPLING} = f_{DTS}/4$, N=6; 0111: 采样频率 $f_{SAMPLING} = f_{DTS}/4$, N=8; 1000: 采样频率 $f_{SAMPLING} = f_{DTS}/8$, N=6; 1001: 采样频率 $f_{SAMPLING} = f_{DTS}/8$, N=8; 1010: 采样频率 $f_{SAMPLING} = f_{DTS}/16$, N=5; 1011: 采样频率 $f_{SAMPLING} = f_{DTS}/16$, N=6; 1100: 采样频率 $f_{SAMPLING} = f_{DTS}/16$, N=8; 1101: 采样频率 $f_{SAMPLING} = f_{DTS}/32$, N=5; 1110: 采样频率 $f_{SAMPLING} = f_{DTS}/32$, N=6; 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。 注意: 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 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 31: 25	保留	0x0	resd	保持默认值。
位 23: 17	保留	0x0	resd	保持默认值。
位 15	C4OSEN	0x0	rw	通道 4 输出开关使能 (Channel 4 output switch enable)
位 24, 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。				
位 9: 8	C4C	0x0	rw	
位 7	C3OSEN	0x0	rw	通道 3 输出开关使能 (Channel 3 output switch enable)
位 16, 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。				
位 1: 0	C3C	0x0	rw	

输入模式

域	简称	复位值	类型	功能
位 31: 16	保留	0x0	resd	保持默认值。
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)
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 9: 8	C4C	0x0	rw	
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)
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。				
位 1: 0	C3C	0x0	rw	

14.4.4.9 TMR1通道控制寄存器 (TMR1_CCTRL)

域	简称	复位值	类型	功能
位 31: 22	保留	0x0	resd	保持默认值。
位 21	C6P	0x0	rw	通道 6 极性 (Channel 6 polarity) 见 C1P 的描述。
位 20	C6EN	0x0	rw	通道 6 使能 (Channel 6 enable) 见 C1EN 的描述。
位 19: 18	保留	0x0	resd	保持默认值。
位 17	C5P	0x0	rw	通道 5 极性 (Channel 5 polarity) 见 C1P 的描述。
位 16	C5EN	0x0	rw	通道 5 使能 (Channel 5 enable) 见 C1EN 的描述。
位 15	C4CP	0x0	rw	通道 4 互补极性 (Channel 4 complementary polarity) 见 C1CP 的描述。
位 14	C4CEN	0x0	rw	通道 4 互补使能 (Channel 4 complementary enable) 见 C1CEN 的描述。
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 14-15 带停止功能的互补输出通道CxOUT和CxCOUT的控制位

控制位					输出状态	
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	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, CxEN=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	CxEN=CxCEN=0 时: 输出禁止 (对应 IO 与定时器断开, IO 浮空); 其它情况下: 关闭状态 (对应通道输出无效电平) 异步地: CxOUT =CxP, Cx_EN=1, CxCOUT=CxCP, CxEN=CxCEN=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。	
		1	0	0		
		1	1	1		
		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。

14.4.4.10 TMR1计数值寄存器 (TMR1_CVAL)

域	简称	复位值	类型	功能
位 31	OVFIFCPY	0x0	ro	OVFIF 副本 (OVFIF copy) OVFIFRMP 为 0 时, 该位保留, 读为 0; OVFIFRMP 为 1 时, 该位是 OVFIF 位的只读副本。
位 30: 16	保留	0x0	resd	保持默认值。
位 15: 0	CVAL	0x0000	rw	计数值 (Counter value)

14.4.4.11 TMR1预分频器寄存器 (TMR1_DIV)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0	resd	保持默认值。
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。

14.4.4.12 TMR1周期寄存器 (TMR1_PR)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0	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。

14.4.4.13 TMR1重复周期寄存器 (TMR1_RPR)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0	resd	保持默认值。
位 15: 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。

14.4.4.14 TMR1通道1数据寄存器 (TMR1_C1DT)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0	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。

14.4.4.15 TMR1通道2数据寄存器 (TMR1_C2DT)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0	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。

14.4.4.16 TMR1通道3数据寄存器 (TMR1_C3DT)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0	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。

14.4.4.17 TMR1通道4数据寄存器 (TMR1_C4DT)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0	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。

14.4.4.18 TMR1刹车寄存器 (TMR1_BRK)

域	简称	复位值	类型	功能
位 31: 30	保留	0x0	resd	保持默认值
位 29	BRK2BID	0x0	rw	刹车 2 双向 (Brake2 bidirectional) 参考 BRKBID 位描述。
位 28	BRKBID	0x0	rw	刹车双向 (Brake bidirectional) 0: 关闭双向刹车模式; 1: 开启双像刹车模式。
位 27	BRK2RLS	0x0	rw	刹车 2 解除(Brake2 release) 参考 BRKRLS 位描述。
位 26	BRKRLS	0x0	rw	刹车解除(Brake release) 0: 启动刹车输入; 1: 解除刹车输入。
位 25	BRK2V	0x0	rw	刹车 2 输入信号的有效性 (Brake2 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 24	BRK2EN	0x0	rw	刹车 2 功能使能 (Brake2 enable) 用于开启刹车功能。 注意: 刹车 2 功能必须搭配 BRK2F 使用。
位 23: 20	BRK2F	0x0	rw	刹车 2 输入滤波 (brake2 input filter) 参考 BRKF 位描述。
位 19: 16	BRKF	0x0	rw	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; 0001: 采样频率 $f_{SAMPLING} = f_{CK_INT}$, N=2; 0010: 采样频率 $f_{SAMPLING} = f_{CK_INT}$, N=4; 0011: 采样频率 $f_{SAMPLING} = f_{CK_INT}$, N=8; 0100: 采样频率 $f_{SAMPLING} = f_{DTS}/2$, N=6; 0101: 采样频率 $f_{SAMPLING} = f_{DTS}/2$, N=8; 0110: 采样频率 $f_{SAMPLING} = f_{DTS}/4$, N=6; 0111: 采样频率 $f_{SAMPLING} = f_{DTS}/4$, N=8; 1000: 采样频率 $f_{SAMPLING} = f_{DTS}/8$, N=6; 1001: 采样频率 $f_{SAMPLING} = f_{DTS}/8$, N=8;

				1010: 采样频率 $f_{SAMPLING} = f_{DTS}/16$, N=5; 1011: 采样频率 $f_{SAMPLING} = f_{DTS}/16$, N=6; 1100: 采样频率 $f_{SAMPLING} = f_{DTS}/16$, N=8; 1101: 采样频率 $f_{SAMPLING} = f_{DTS}/32$, N=5; 1110: 采样频率 $f_{SAMPLING} = f_{DTS}/32$, N=6; 1111: 采样频率 $f_{SAMPLING} = f_{DTS}/32$, N=8。 注意: 如无法保证刹车输入干净, 建议开启刹车输入滤波。
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: BRK2BID、BRKBID、BRK2RLS、BRKRRLS、 BRK2V、BRKV、BRK2EN、BRKEN、BRK2F、BRKF、 AOEN 和 DTC 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: CxOCTRL 和 CxOBEN 注意: WPC>0 时将无法再次被修改, 直到系统复位。
位 7: 0	DTC	0x00	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，AOEN、BRK2BID、BRKBID、BRK2RLS、BRKRLS、BRK2V、BRKV、BRK2EN、BRKEN、FCSODIS、FCSONEN 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。

14.4.4.19 TMR1 DMA控制寄存器 (TMR1_DMACTRL)

域	简称	复位值	类型	功能
位 31: 14	保留	0x0	resd	保持默认值。
位 13: 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： 000000: 1 个字节 000001: 2 个字节 000010: 3 个字节 000011: 4 个字节 011101: 30 个字节 011110: 31 个字节
位 7: 6	保留	0x0	resd	保持默认值。
位 5: 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： 000000: TMR1_CTRL1 000001: TMR1_CTRL2 000010: TMR1_STCTRL

14.4.4.20 TMR1 DMA数据寄存器 (TMR1_DMADT)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0	resd	保持默认值。
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。

14.4.4.21 TMR1 选项寄存器1 (TMR1_OPT1)

域	简称	复位值	类型	功能
位 31: 1	保留	0x0	resd	保持默认值。
位 0	CORAW_OFF	0x0	rw	CORAW_OFF 源选择 (CORAW_OFF source selection) 0: CMP1; 1: CMP2。

14.4.4.22 TMR1 通道模式寄存器3 (TMR1_CM3)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0	resd	保持默认值
位 15	C6OSEN	0x0	rw	通道 6 输出开关使能 (Channel 6 output switch enable)
位 24, 14: 12	C6OCTRL	0x0	rw	通道 6 输出控制 (Channel 6 output control)
位 11	C6OBEN	0x0	rw	通道 6 输出缓存使能 (Channel 6 output buffer enable)
位 10	C6OIEN	0x0	rw	通道 6 输出立即使能 (Channel 6 output immediately enable)
位 9: 8	保留	0x0	resd	保持默认值。
位 7	C5OSEN	0x0	rw	通道 5 输出开关使能 (Channel 5 output switch enable)
位 16, 6: 4	C5OCTRL	0x0	rw	通道 5 输出控制 (Channel 5 output control)
位 3	C5OBEN	0x0	rw	通道 5 输出缓存使能 (Channel 5 output buffer enable)
位 2	C5OIEN	0x0	rw	通道 5 输出立即使能 (Channel 5 output immediately enable)
位 1: 0	保留	0x0	resd	保持默认值。

14.4.4.23 TMR1 通道5数据寄存器 (TMR1_C5DT)

域	简称	复位值	类型	功能
位 31	CC5C3	0x0	rw	通道 5 和通道 3 组合 (Combine channel 5 and channel 3) 0: C5ORAW 对 C3ORAWC 无影响; 1: C3ORAWC 是 C3ORAWC 和 C5ORAW 逻辑与运算的结果。
位 30	CC5C2	0x0	rw	通道 5 和通道 2 组合 (Combine channel 5 and channel 2) 0: C5ORAW 对 C2ORAWC 无影响; 1: C2ORAWC 是 C2ORAWC 和 C5ORAW 逻辑与运算的结果。
位 29	CC5C1	0x0	rw	通道 5 和通道 1 组合 (Combine channel 5 and channel 1) 0: C5ORAW 对 C1ORAWC 无影响; 1: C1ORAWC 是 C1ORAWC 和 C5ORAW 逻辑与运算的结果。
位 28: 16	保留	0x0	resd	保持默认值。
位 15: 0	C5DT	0x0	rw	通道 5 数据寄存器值 (Channel 5 data register) C5DT 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 (C5OBEN), 并根据设置在 C5OUT 上产生相应的输出。

14.4.4.24 TMR1通道6数据寄存器 (TMR1_C6DT)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0	resd	保持默认值。
位 15: 0	C6DT	0x0	rw	通道 6 数据寄存器值 (Channel 6 data register) C6DT 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 (C6OBEN), 并根据设置在 C6OUT 上产生相应的输出。

14.4.4.25 TMR1选项寄存器2 (TMR1_OPT2)

域	简称	复位值	类型	功能
位 31: 18	保留	0x0	resd	保持默认值。
位 17: 14	EXTSEL	0x0	rw	EXT 源选择 (EXT source selection) 0000: GPIO; 0001: CMP1; 0010: CMP2; 1000: ADC1_VM; 1001: ADC2_VM; 其他值: 保留。
位 13: 12	保留	0x0	rw	保持默认值。
位 11	BRKCMMP2P	0x0	rw	BRK CMP2 输入极性 (BRK CMP2 input polarity) 0: 高电平有效; 1: 低电平有效。
位 10	BRKCMMP1P	0x0	rw	BRK CMP1 输入极性 (BRK CMP1 input polarity) 0: 高电平有效; 1: 低电平有效。
位 9	BRKINP	0x0	rw	BRK BKIN 输入极性 (BRK BKIN input polarity) 0: 高电平有效; 1: 低电平有效。
位 8: 3	保留	0x0	resd	保持默认值。
位 2	BRKCMMP2EN	0x0	rw	BRK CMP2 使能 (BRK CMP2 enable) 0: 禁止 CMP2 输入; 1: 使能 CMP2 输入。
位 1	BRKCMMP1EN	0x0	rw	BRK CMP1 使能 (BRK CMP1 enable) 0: 禁止 CMP1 输入; 1: 使能 CMP1 输入。
位 0	BRKINEN	0x1	rw	BRK BKIN 使能 (BRK BKIN enable)

0: 禁止 BKIN 输入;
1: 使能 BKIN 输入。

14.4.4.26 TMR1选项寄存器3 (TMR1_OPT3)

域	简称	复位值	类型	功能
位 31: 12	保留	0x0	resd	保持默认值。
位 11	BRK2CMP2P	0x0	rw	BRK2 CMP2 输入极性 (BRK2 CMP2 input polarity) 0: 高电平有效; 1: 低电平有效。
位 10	BRK2CMP1P	0x0	rw	BRK2 CMP1 输入极性 (BRK2 CMP1 input polarity) 0: 高电平有效; 1: 低电平有效。
位 9	BRK2INP	0x0	rw	BRK2 BKIN 输入极性 (BRK2 BKIN input polarity) 0: 高电平有效; 1: 低电平有效。
位 8: 3	保留	0x0	resd	保持默认值。
位 2	BRK2CMP2EN	0x0	rw	BRK2 CMP2 使能 (BRK2 CMP2 enable) 0: 禁止 CMP2 输入; 1: 使能 CMP2 输入。
位 1	BRK2CMP1EN	0x0	rw	BRK2 CMP1 使能 (BRK2 CMP1 enable) 0: 禁止 CMP1 输入; 1: 使能 CMP1 输入。
位 0	BRK2INEN	0x1	rw	BRK2 BKIN 使能 (BRK2 BKIN enable) 0: 禁止 BKIN 输入; 1: 使能 BKIN 输入。

14.4.4.27 TMR1通道选择寄存器 (TMR1_CHSEL)

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	CH4SEL	0x0	rw	CH4 输入选择 (CH4 input selection) 0000: TMR1_CH4; 其他值: 保留。
位 23: 20	保留	0x0	resd	保持默认值。
位 19: 16	CH3SEL	0x0	rw	CH3 输入选择 (CH3 input selection) 0000: TMR1_CH3; 其他值: 保留。
位 15: 12	保留	0x0	resd	保持默认值。
位 11: 8	CH2SEL	0x0	rw	CH2 输入选择 (CH2 input selection) 0000: TMR1_CH2; 0001: CMP1; 0010: CMP2; 其他值: 保留。
位 7: 4	保留	0x0	resd	保持默认值。
位 3: 0	CH1SEL	0x0	rw	CH1 输入选择 (CH1 input selection) 0000: TMR1_CH1; 0001: CMP1; 0010: CMP2; 其他值: 保留。

14.4.4.28 TMR1通道1非对称数据寄存器 (TMR1_C1ADT)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0	resd	保持默认值。
位 15: 0	C1ADT	0x0	rw	通道 1 非对称数据寄存器值 (Channel 1 asymmetric data register) C1ADT 是非对称 PWM 模式向下计数时将要和 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。

14.4.4.29 TMR1通道2非对称数据寄存器（TMR1_C2ADT）

域	简称	复位值	类型	功能
位 31: 16	保留	0x0	resd	保持默认值。
位 15: 0	C2ADT	0x0	rw	通道 2 非对称数据寄存器值（Channel 2 asymmetric data register） C2ADT 是非对称 PWM 模式向下计数时将要和 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。

14.4.4.30 TMR1通道3非对称数据寄存器（TMR1_C3ADT）

域	简称	复位值	类型	功能
位 31: 16	保留	0x0	resd	保持默认值。
位 15: 0	C3ADT	0x0	rw	通道 3 非对称数据寄存器值（Channel 3 asymmetric data register） C3ADT 是非对称 PWM 模式向下计数时将要和 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。

14.4.4.31 TMR1通道4非对称数据寄存器（TMR1_C4ADT）

域	简称	复位值	类型	功能
位 31: 16	保留	0x0	resd	保持默认值。
位 15: 0	C4ADT	0x0	rw	通道 4 非对称数据寄存器值（Channel 4 asymmetric data register） C4ADT 是非对称 PWM 模式向下计数时将要和 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。

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。

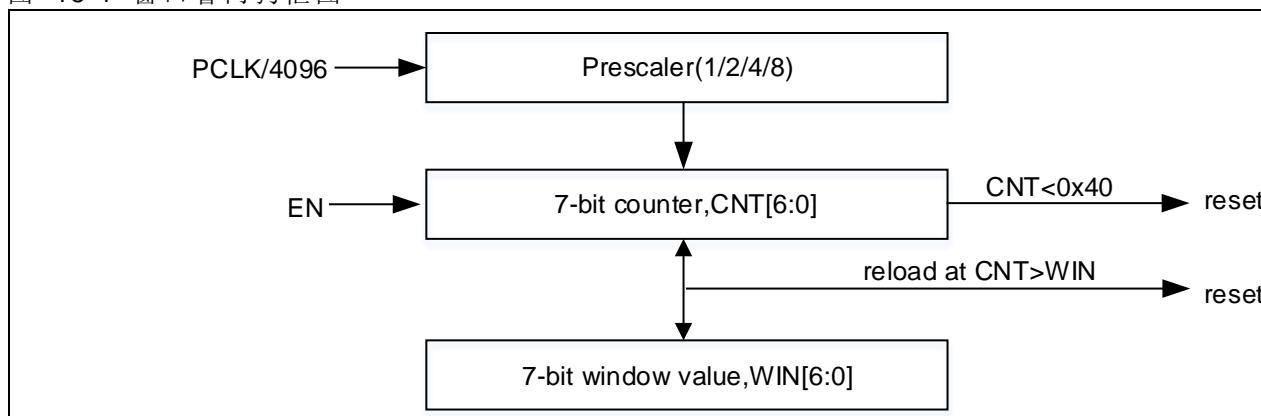
15.3 WWDT功能描述

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 WWDT_CFG 寄存器 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 WWDT_CTRL 寄存器时，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：

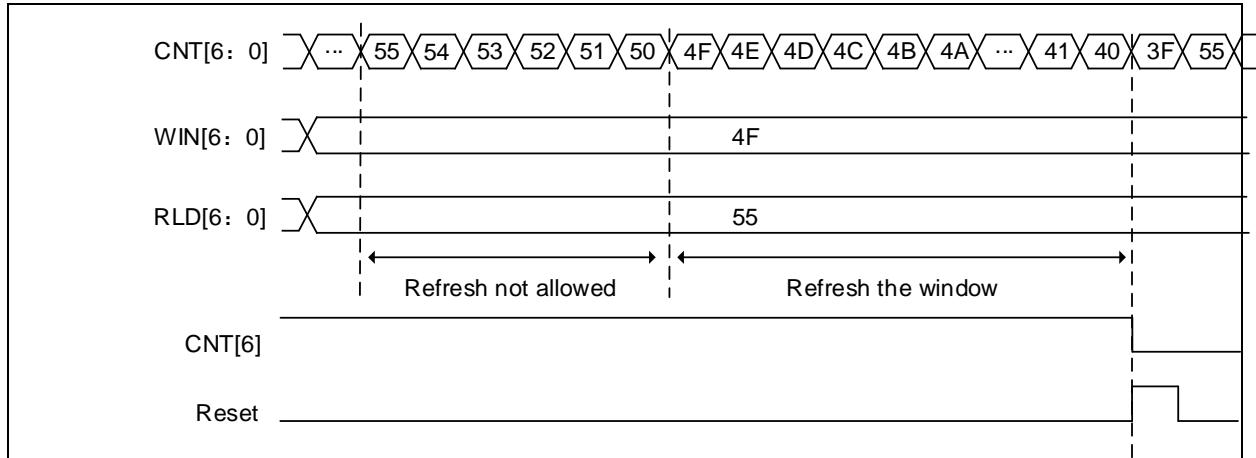
$$T_{WWDT} = T_{PCLK1} \times 4096 \times 2^{DIV[1: 0]} \times (CNT[5: 0] + 1); \quad (\text{ms})$$

下表给出了当 PCLK1 频率为 180MHz 时，最大和最小看门狗超时时间。

表 15-1 PCLK1频率为180MHz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	22.76 μs	1.46ms
1	45.51 μs	2.91ms
2	91.02 μs	5.83ms
3	182.04 μs	11.65ms

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®-M4F 核心停止。将 DEBUG 模块中 WWDT_PAUSE 位置 1 可以将 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		0x0000 007F
WWDT_CFG		0x04		0x0000 007F
WWDT_STS		0x08		0x0000 0000

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	0x0000000	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: 启开。 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	0x0000000	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: 启开。
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）
- 可选择在DEEPSLEEP、STANDBY模式下是否停止计数
- 支持两种复位方式：
 - 当递减计数器递减至0。
 - 当递减计数器在窗口外被重新装载。

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 WDT_CMD 寄存器写入 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 WDT_CMD 寄存器写入 0xAAAA 重载计数器值。此外，若将 WIN[11: 0]设置为非默认值（0xFFFF）将开启窗口看门狗功能，在计数值大于窗口值时重载计数器值将会产生系统复位。

WDT 写保护:

WDT_DIV、WDT_RLD、WDT_WIN 寄存器受写保护，向 WDG_CMD 寄存器写入 0x5555 可解锁寄存器写保护，之后可对其进行配置。这三个寄存器的更新状态分别由 WDT_STS 寄存器中 DIVF、RLDF、WINF 指示。向 WDG_CMD 寄存器写入其它值将重新启动 WDT_DIV、WDT_RLD、WDT_WIN 寄存器写保护。向 WDG_CMD 寄存器写入 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。

WDT 低功耗计数模式:

WDT 能够在 SLEEP、DEEPSLEEP、STANDBY 模式下运行，用户可选择进入 DEEPSLEEP、STANDBY 模式后计数器是否停止计数，可由用户系统数据区中的 nDEPSLP_WDT、nSTDBY_WDT 位配置。

如果设置了停止计数，当进入了 DEEPSLEEP、STANDBY 模式后，看门狗计数器停止递减，意味着看门狗在这两种低功耗模式下不会发生复位，当从这两种模式唤醒后，计数器从进入时的值继续递减。

图 16-1看门狗框图

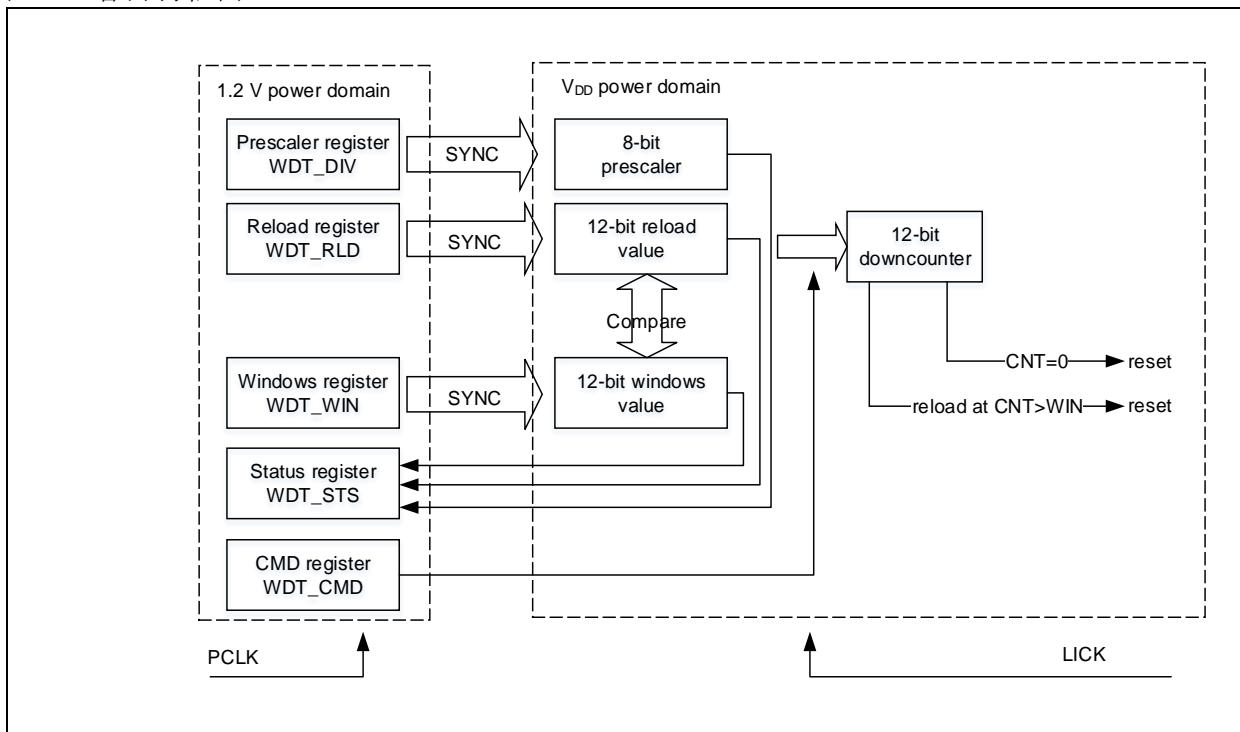


表 16-1看门狗超时时间 (当LICK=40kHz时)

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

当微控制器处于调试模式时，意味着 Cortex®-M4F 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
WDT_WIN	0x10	0x0000 0FFF

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) 0xAAAA: 重载计数器; 0x5555: 解锁 WDT_DIV、WDT_RLD、WDT_WIN 写保护;
位 15: 0	CMD	0x0000	wo	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。

16.5.2 预分频寄存器 (WDT_DIV)

(待机模式时不复位)

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)

(待机模式时不复位)

域	简称	复位值	类型	功能
位 31: 12	保留	0x00000	resd	保持默认值。 重载值 (Reload value)
位 11: 0	RLD	0xFFFF	rw	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: 3	保留	0x0000 0000	resd	保持默认值。
位 2	WINF	0x0	ro	窗口值更新完成标志 (Window value update complete flag) 0: 更新完成; 1: 正在更新。 只有当 RLDF 为 0 时才能写 WDT_WIN 寄存器。
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 WDT_RLD 寄存器。
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 WDT_DIV 寄存器。

16.5.5 窗口寄存器 (WDT_WIN)

(待机模式不复位)

域	简称	复位值	类型	功能
位 31: 12	保留	0x0000000	resd	保持默认值。
位 11: 0	WIN	0xFFFF	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 0~窗口值。

17 实时时钟 (ERTC)

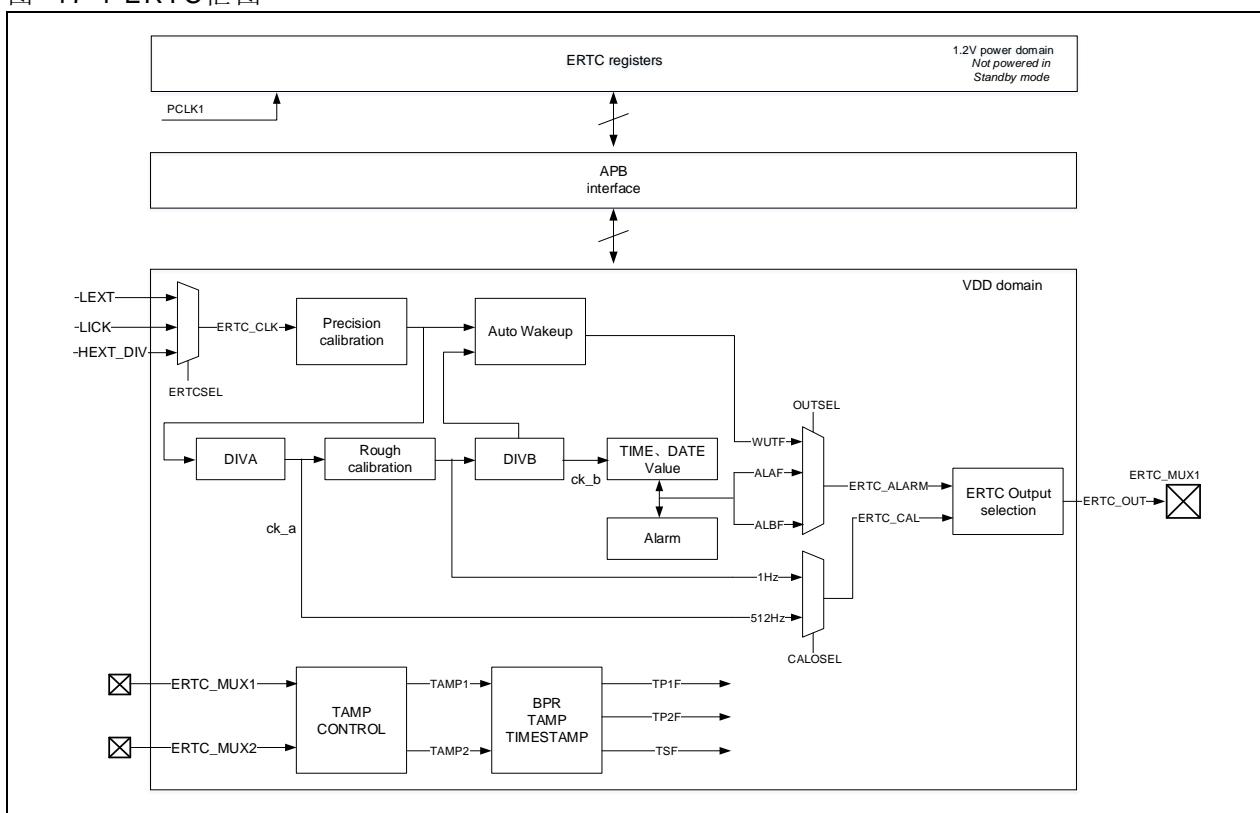
17.1 ERTC简介

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，支持两组闹钟
- 周期性唤醒
- 参考时钟检测
- 两组可配置入侵检测，支持时间戳功能
- 支持精密校准
- 20个电池供电寄存器
- 5组中断：闹钟A、闹钟B、周期性唤醒、入侵检测、时间戳
- 复用功能输出，校准时钟输出、闹钟事件或唤醒事件
- 复用功能输入，参考时钟输入、两路入侵检测、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、分频后的 HEXT 中选择（由 CRM_BPDC 寄存器 ERTCSEL[1: 0]配置），HEXT 的分频值由 CRM_CFG 寄存器 ERTC_DIV[4: 0]配置。

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 用于更新日历和周期性唤醒。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。

注意: 若 ERTC_CLK 选用分频后的 HEXT, 应先配置 HEXT 分频值, 再将时钟源切换为 HEXT。

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_STS[14: 8]、ERTC_TAMP 和 ERTC_BPRx 寄存器外, 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: CRM_APB1EN 的 PWCEN=1
- 2、解锁电池供电域写保护: PWC_CTRL 的 BPWEN=1
- 3、依次向 ERTC_WP 寄存器写入 0xCA, 0x53, 若向 ERTC_WP 寄存器写入错误的值, 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 ERTC_WP 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 7、6、4 需要	-
ERTC_STS	除位[14: 8]外	-	-
ERTC_DIV	是	是	-
ERTC_WAT	是	否	WATWF 为 1 时可配置
ERTC_ALA	是	否	ALAWF 为 1 时可配置
ERTC_ALB	是	否	ALBWF 为 1 时可配置
ERTC_WP	-	-	-
ERTC_SBS	-	-	-
ERTC_TADJ	是	否	TADJF 为 0 时可配置
ERTC_TSTM	-	-	-
ERTC_TSDT	-	-	-
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_ALBSBS	是	否	ALBWF 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_SBS、ERTC_TIME、ERTC_DATE 寄存器值来自同一时刻，读取低阶寄存器时会将高阶寄存器值锁定，直到读取 ERTC_DATE 寄存器。例如读取 ERTC_SBS，会将 ERTC_TIME、ERTC_DATE 寄存器值锁定。

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、闹钟 B 两个闹钟，并分别提供了闹钟 A 中断、闹钟 B 中断。

闹钟值由 ERTC_ALASBS/ERTC_ALA（ERTC_ALBSBS/ERTC_ALB）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、B 的配置可按以下步骤进行：

1. 关闭闹钟 A 或闹钟 B（设置 ALAEN=0 或 ALBEN=0）。
2. 等待闹钟 A 或闹钟 B 寄存器允许写（等待 ALAWF 或 ALBWF 位置 1）。
3. 配置闹钟 A 或闹钟 B 寄存器（ERTC_ALA/ERTC_ALASBS、ERTC_ALB/ERTC_ALBSBS）。
4. 使能闹钟 A 或闹钟 B（设置 ALAEN=1 或 ALBEN=1）。

注：当 ERTC_ALA 或 ERTC_ALB 中 MASK1 为 0 时，DIVB 至少为 3 才能使闹钟正常工作。

17.3.3 周期性自动唤醒

周期性唤醒功能用于 ERTC 周期性的自动唤醒低功耗模式，唤醒周期由 VAL[15: 0]设定（WATCLK[2]=1 时扩展为 17 位，唤醒计数值为 $VAL + 2^{16}$ ）。当唤醒计数器值由 VAL 值递减至 0 时，WATF 标志置 1，产生唤醒事件，同时唤醒计数器值重载 VAL 值。若使能周期性唤醒中断，将产生周期性唤醒中断。

驱动唤醒定时器的时钟通过 WATCLK[2: 0]设定，可选 16/8/4/2 分频后的 ERTC_CLK 或 ck_b(通常 1Hz)，结合唤醒计数值可灵活调整唤醒周期。

周期性唤醒功能可按以下步骤进行配置：

1. 关闭周期性唤醒（设置 WATEN=0）。
2. 等待唤醒自动重载定时器和 WATCLK[2: 0]位允许写（WATWF 位置 1）。
3. 配置唤醒定时器计数值和唤醒时钟（VAL[15: 0]、WATCLK[2: 0]位）。
4. 使能定时器（设置 WATEN=1）。

注：系统复位以及低功耗模式（睡眠、深度睡眠和待机）对唤醒定时器没有任何影响。

注：DEBUG 模式下，若唤醒时钟选择 ERTC_CLK，用于周期性唤醒的计数器正常运行。

17.3.4 ERTC校准

ERTC 支持使用精密数字校准校准时钟。

精密数字校准:

精密校准的均匀性很好。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_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。

17.3.5 参考时钟检测

为保证日历长时间运行的精确性，ERTC 提供了时钟同步功能（低功耗模式不可用），用精度更高的参考时钟（一般用 50Hz 或者 60Hz 的市电）校准更新日历的 1Hz 时钟。

参考时钟检测功能开启后，在每次更新日历值的前 7 个 ck_a 周期检测参考时钟边沿，若检测到边沿，将时用此边沿更新日历值，后续采用 3 个 ck_a 周期检测参考时钟边沿。每一次检测到参考时钟边沿时，都会将分频器 A 的值进行重载，这会使得内部 1Hz 的日历时钟与参考时钟边沿刚好对齐，当内部 1Hz 时钟出现微小偏移时，利用更精确的参考时钟，将 1Hz 时钟微调至与参考时钟边沿对齐。当没有检测到参考时钟边沿时，ERTC 会利用原来的时钟源更新日历。

需要注意的是，使能参考时钟功能后，需要将 DIVA、DIVB 设置为复位值（0x7F、0xFF）。

17.3.6 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：

- 选择时间戳引脚（设置 TSPIN）
- 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

17.3.7 入侵检测

ERTC 提供了两组入侵检测 TAMP1、TAMP2，且两组入侵检测可分别配置为滤波后的电平检测或边沿检测。TAMP1 通过 TSPIN 选择入侵引脚 ERTC_MUX1 或 ERTC_MUX2；TAMP2 只能选择 ERTC_MUX2。

当检测到有效的入侵事件后，TP1F 或 TP2F 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 或 TP2EDG)。
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. 若选择 TAMP1 进行入侵检测，选择 TAMP1 映射关系为 ERTC_MUX1 或是 ERTC_MUX2 (TP1PIN) 并将 TAMP1 使能 (TP1EN 置 1)；若选择 TAMP2 进行入侵检测，则只需使能 TAMP2 (TP2EN 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 或 TP2EDG)。
 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. 若选择 TAMP1 进行入侵检测，选择 TAMP1 映射关系为 ERTC_MUX1 或是 ERTC_MUX2 (TP1PIN) 并将 TAMP1 使能 (TP1EN 置 1)；若选择 TAMP2 进行入侵检测，则只需使能 TAMP2 (TP2EN 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。

注：当电池供电域电源关闭时，入侵检测失效。

注：standby 模式下，TAMP2 (PA0 引脚) 无法使用预充电功能。

17.3.8 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)
3. 闹钟B (OUTSEL=2)
4. 唤醒事件 (OUTSEL=3)

当输出闹钟事件或者唤醒事件时 (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。

17.3.9 ERTC唤醒

ERTC 可由闹钟、周期性唤醒、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	√	×	×
	闹钟 B	√	×	×
	周期性唤醒	√	×	×
	时间戳	√	×	×
	入侵事件	√	×	×
LICK	闹钟 A	√	√	√
	闹钟 B	√	√	√
	周期性唤醒	√	√	√

	时间戳	√	√	√
	入侵事件	√	√	√
	闹钟 A	√	√	√
	闹钟 B	√	√	√
LEXT	周期性唤醒	√	√	√
	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
闹钟 B	ALBF	ALBIEN	17
周期性唤醒	WATF	WATIEN	22
时间戳	TSF	TSIEN	21
入侵事件	TP1F/TP2F	TPIEN	21

17.4 ERTC 寄存器描述

必须以字（32 位）的方式操作这些外设寄存器。

ERTC 寄存器是 32 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_WAT	0x14	0x0000 FFFF
ERTC_ALA	0x1C	0x0000 0000
ERTC_ALB	0x20	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_ALBSBS	0x48	0x0000 0000
ERTC_BPRx	0x50-0x9C	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: 闹钟 B; 11: 唤醒事件。
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	WATIEN	0x0	rw	唤醒定时器中断使能 (Wakeup timer interrupt enable) 0: 关闭; 1: 开启。
位 13	ALBIEN	0x0	rw	闹钟 B 中断使能 (Alarm B interrupt enable) 0: 关闭; 1: 开启。
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	WATEN	0x0	rw	唤醒定时器使能 (Wakeup timer enable) 0: 关闭; 1: 开启。
位 9	ALBEN	0x0	rw	闹钟 B 使能 (Alarm B enable) 0: 关闭; 1: 开启。
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	WATCLK	0x0	rw	唤醒定时器时钟选择 (Wakeup timer clock selection) 000: ERTC_CLK/16; 001: ERTC_CLK/8; 010: ERTC_CLK/4; 011: ERTC_CLK/2; 10x: ck_b; 11x: ck_b, 唤醒计数值增加 2^{16} , 唤醒时间 = ERTC_WAT + 2^{16} 。 注: 在 WATEN=0 且 WATWF=1 时可对这些位进行写操作。

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	保留	0x0	resd	保持默认值。
位 14	TP2F	0x0	rw0c	入侵检测 2 标志 (Tamper detection 2 flag) 0: 无入侵事件发生; 1: 有入侵事件发生。
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	WATF	0x0	rw0c	唤醒定时器标志 (Wakeup timer flag) 0: 无唤醒事件发生; 1: 有唤醒事件发生。 注：该位清 0 后 2 个 APB_CLK 后生效。
位 9	ALBF	0x0	rw0c	闹钟 B 标志 (Alarm clock B 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 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。 当对时间调整寄存器 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	WATWF	0x1	ro	唤醒定时器寄存器允许写标志 (Wakeup timer register allows write flag) 0: 不允许写； 1: 允许写。
位 1	ALBWF	0x1	ro	闹钟 B 允许写标志 (Alarm B register allows write flag) 0: 不允许写； 1: 允许写。
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唤醒定时器寄存器(ERTC_WAT)

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	VAL	0xFFFF	rw	唤醒定时器重载值 (Wakeup timer reload value)

17.4.7 ERTC闹钟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: 闹钟和日期/星期无关。
位 30	WKSEL	0x0	rw	日期/星期选择 (Date/week mode select) 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)
位 23	MASK3	0x0	rw	小时屏蔽 (Hour mask) 0: 无屏蔽； 1: 闹钟和小时无关。
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	MASK2	0x0	rw	分钟屏蔽 (Minute mask) 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)
位 7	MASK1	0x0	rw	秒钟屏蔽 (Second mask) 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)
-------	----	-----	----	---------------------

17.4.8 ERTC闹钟B寄存器(ERTC_ALB)

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: 闹钟和日期/星期无关。
位 30	WKSEL	0x0	rw	日期/星期选择 (Date/week mode select) 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)
位 23	MASK3	0x0	rw	小时屏蔽 (Hour mask) 0: 无屏蔽; 1: 闹钟和小时无关。
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	MASK2	0x0	rw	分钟屏蔽 (Minute mask) 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)
位 7	MASK1	0x0	rw	秒钟屏蔽 (Second mask) 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)

17.4.9 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。

17.4.10 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)。

17.4.11 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))。

17.4.12 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。

17.4.13 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。

17.4.14 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 ERTC_STS/TSF 位复位时, 清零该寄存器。

17.4.15 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。

17.4.16 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	TSPIN	0x0	rw	时间戳检测引脚选择 (Time stamp detection pin selection) 0: ERTC_MUX1; 1: ERTC_MUX2。
位 16	TP1PIN	0x0	rw	入侵检测 1 引脚选择 (Tamper detection pin selection) 0: ERTC_MUX1; 1: ERTC_MUX2。
位 15	TPPU	0x0	rw	入侵检测上拉 (Tamper detection pull-up) 0: 开启; 1: 关闭。
位 14:13	TPPR	0x0	rw	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 12:11	TPFLT	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 10:8	TPFREQ	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 7	TPTSEN	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 6: 5	保留	0x0	resd	保持默认值。
位 4	TP2EDG	0x0	rw	入侵检测 2 有效边沿 (Tamper detection 2 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 3	TP2EN	0x0	rw	入侵检测 2 使能 (Tamper detection 2 enable) 0: 关闭; 1: 开启。
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。

17.4.17 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)

17.4.18 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)

17.4.19 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	0x0000 0000	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、10位、8位或6位的数字信号的外设。采样率最高可达2.5MSPS。多达26个通道源（包括内部及外部通道）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、10位、8位或6位的转换
- 自校准时间：205个ADC时钟周期
- ADC转换时间
 - ADC时钟在最大频率35MHz时转换时间为0.4 μs
- ADC供电要求：参考Datasheet
- ADC输入范围： $V_{SSA} \leq V_{IN} \leq V_{DDA}$

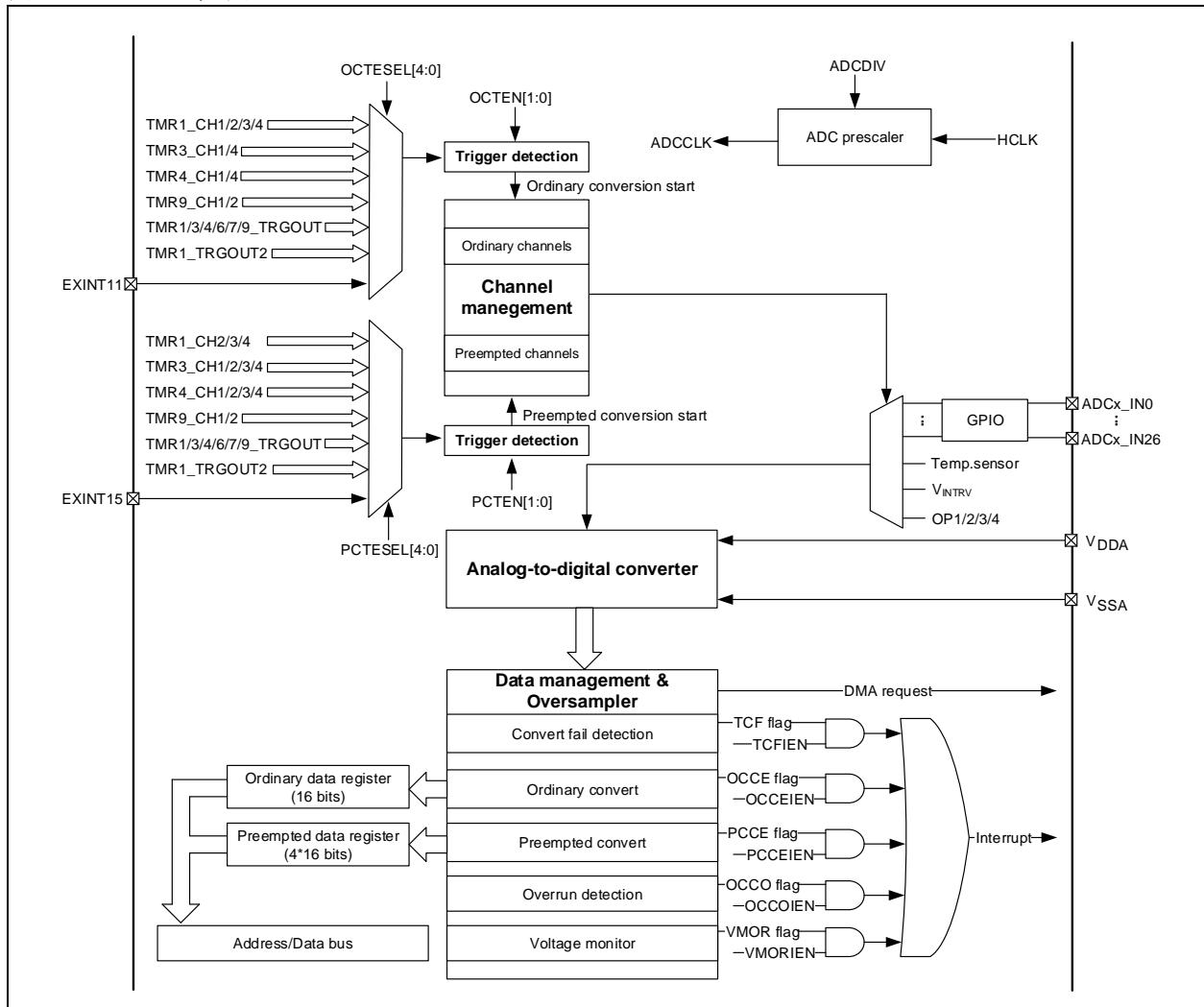
数字控制方面有以下特征：

- 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
- 过采样器
- 可选择的数据对齐方式
- 可配置的电压监测边界，支持ADC_VM输出
- 支持DMA传输的普通通道数据
- 联动多ADC的主从模式
- 可设定以下事件发生时响应中断
 - 普通通道转换数据溢出
 - 抢占通道组转换结束
 - 普通通道转换结束
 - 电压监测超出范围
 - 触发转换失败
- 可配置多ADC之间时序位移长度的主从位移模式
 - 高效能需求可使用主从模式搭配DMA

18.3 ADC 架构

ADC1 的架构如下图所示。

图 18-1 ADC1 框图



输入引脚介绍：

- V_{DDA}：模拟电源，ADC 模拟电源，可与 V_{DD} 连接，或 2.4V ≤ V_{DDA} ≤ V_{DD} (3.6V)
- V_{SSA}：模拟电源地，ADC 模拟电源地，必须与 V_{SS} 连接
- ADCx_IN：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 26。

- ADC1 的 ADC1_IN0/2/4/6/9/11/12/14/15/23/24/26 为外部模拟输入，ADC1_IN16 为内部温度传感器，ADC1_IN17 为内部参考电压，ADC1_IN19~IN22 为 OP 输入。
- ADC2 的 ADC2_IN0~IN12、ADC2_IN14 为外部模拟输入，ADC2_IN16 为 VDDA，ADC2_IN17 为 VSSA，ADC2_IN19 至 ADC2_IN20 为 OP 输入。

表 18-1 ADC通道输入源

通道编号	ADC1 输入源	ADC2 输入源
IN0	PA0	PA0
IN1	保留	PA1
IN2	PA2	PA2
IN3	保留	PA3
IN4	PA4	PA4
IN5	保留	PA5
IN6	PA6	PA6
IN7	保留	PA7
IN8	保留	PB0
IN9	PB1	PB1
IN10	保留	PB2
IN11	PB11	PB11
IN12	PB12	PB12
IN13	保留	保留
IN14	PB14	PB14
IN15	PB15	保留
IN16	温度传感器	VDDA
IN17	内部参考电压	VSSA
IN18	保留	保留
IN19	OP1_OUT	OP1_OUT
IN20	OP2_OUT	OP2_OUT
IN21	OP3_OUT	保留
IN22	OP4_OUT	保留
IN23	PF9	保留
IN24	PF10	保留
IN25	保留	保留
IN26	PF12	保留

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_CCTRL 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。

温度传感器输出电压与温度呈线性变化，电压和温度曲线的平均斜率值参考数据手册。由于生产制造的偏差，温度曲线在不同芯片间存在差异，因此温度传感器更适合监测温度变化而非绝对温度。使用 ADC 转换 ADC_IN16，计算当前温度传感器输出电压，结合温度计算公式，可计算出当前温度。

温度计算公式如下：

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

V_{25} : 温度传感器 25°C 时输出的电压值，参数值详见数据手册。

V_{TS} : 温度传感器当前输出的电压值。

Avg_Slope: 温度传感器温度曲线平均斜率，参数值详见数据手册。

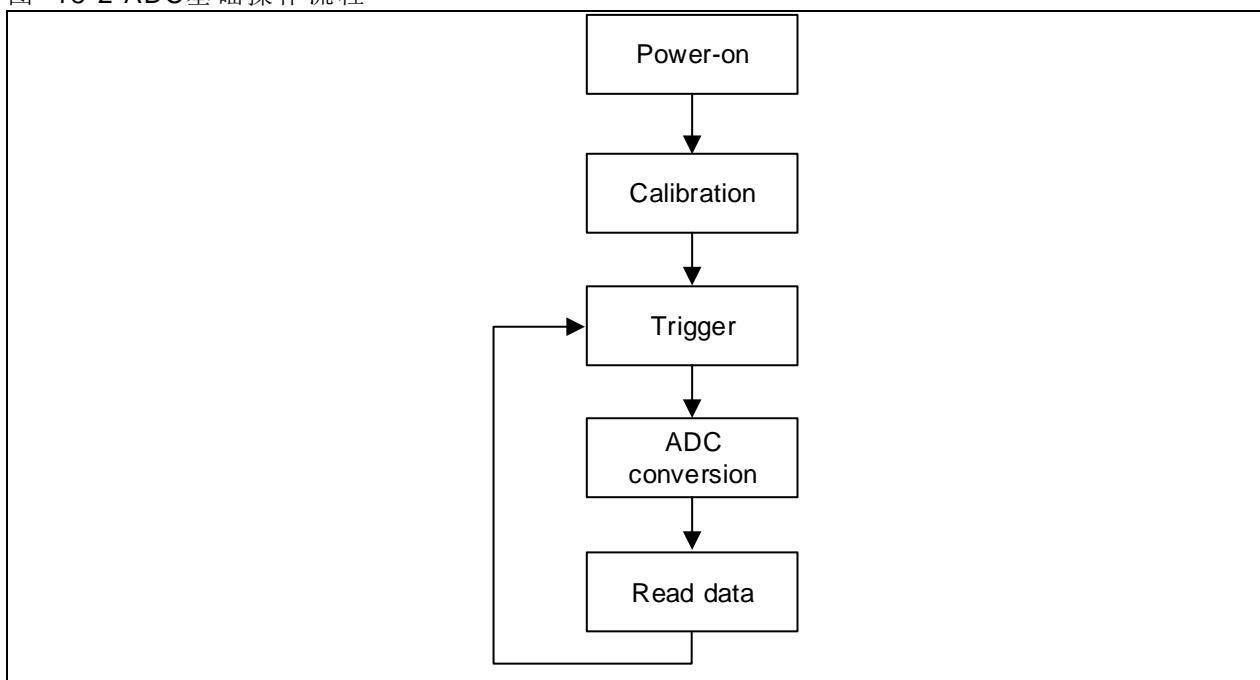
18.4.1.2 内部参考电压

典型值 1.2V 的内部参考电压接到 ADC1_IN17，必须先使能 ADC_CCTRL 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 的基础操作流程如下图所示，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 CRM_APB2EN 的 ADCxEN，以使能 ADC 的时钟：PCLK2 与 ADCCLK。

时钟使能后必须配置 ADC 预分频器（ADC_CCTRL 的 ADCDIV），将 ADCCLK 调整至需求的频率。

注意：ADCCLK 不可大于 35MHz，ADCCLK 频率不可高于 PCLK2。

ADCCLK 频率调整完后，即可使能 ADC_CTRL2 的 ADCEN 位使 ADC 上电，等待 RDY 标志置位后才可对 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。

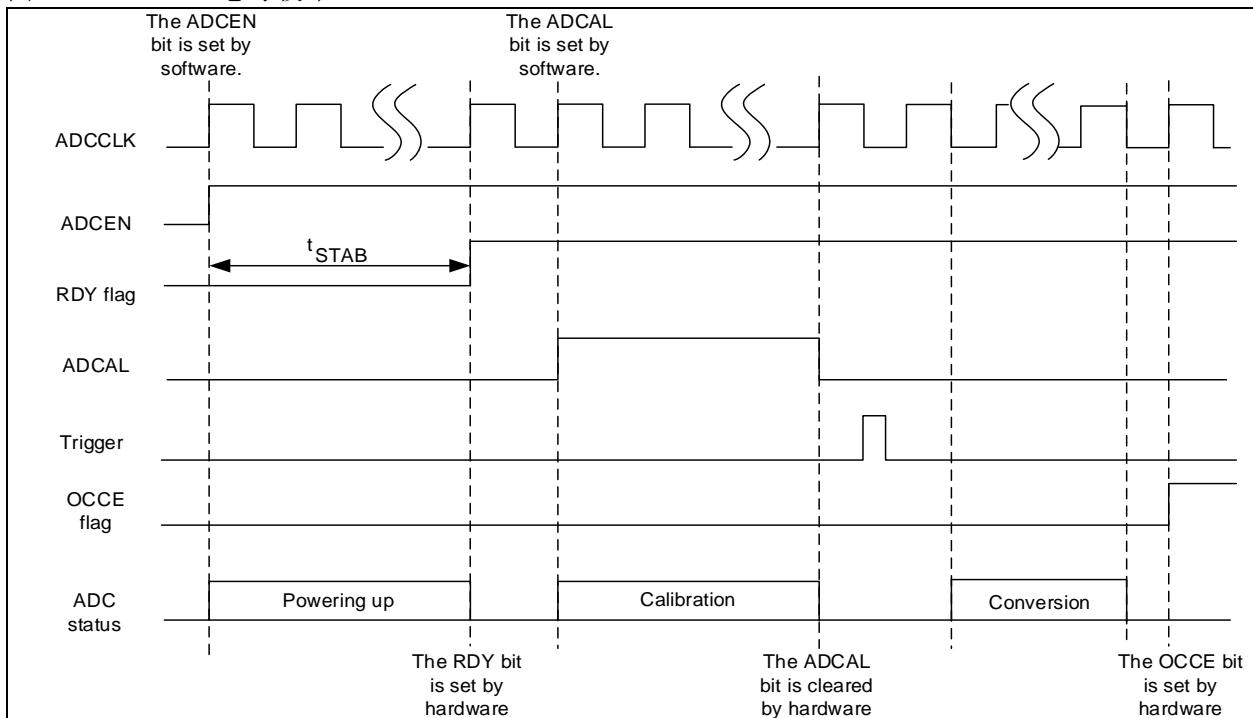
校准

上电完成后可设置 ADC_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_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 OCCE 标志，不会产生中断或 DMA 请求。

注意：只能在分辨率 12 位时进行校准，校准完成后才可切换分辨率，切换分辨率后等待 RDY 标志置位即可触发。

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。针对外部触发来源的有效极性是由 ADC_CTRL2 的 OCETE 与 PCETE 选择，ADC 会检测触发来源并响应转换。

触发来源可分为软件写寄存器触发 (ADC_CTRL2 的 OCSWTRG 与 PCSWTRG) 以及外部触发，外部触发包含定时器触发与引脚触发，由 ADC_CTRL2 的 OCTESEL 与 PCTESEL 选择触发来源，如下表所示。

表 18-2 普通通道与抢占通道触发来源

OCTESEL	触发来源	PCTESEL	触发来源
00000	TMR1_CH1 event	00000	TMR1_CH4 event
00001	TMR1_CH2 event	00001	TMR1_TRGOUT event
00010	TMR1_CH3 event	00010	TMR9_CH1 event
00011	TMR9_CH2 event	00011	TMR9_TRGOUT event
00100	保留	00100	TMR3_CH2 event
00101	保留	00101	TMR3_CH4 event
00110	TMR9_TRGOUT event	00110	TMR4_CH1 event
00111	TMR3_CH1 event	00111	TMR4_CH2 event
01000	TMR3_TRGOUT event	01000	TMR4_CH3 event
01001	TMR4_CH4 event	01001	TMR4_TRGOUT event
01010	保留	01010	保留
01011	保留	01011	保留
01100	保留	01100	TMR1_CH2 event
01101	TMR1_CH4 event	01101	保留
01110	保留	01110	保留
01111	EXINT line11 External pin	01111	EXINT line15 External pin
10000~10010	保留	10000~10010	保留
10011	保留	10011	TMR1_TRGOUT2 event
10100~10101	保留	10100~10101	保留
10110	TMR1_TRGOUT2 event	10110	TMR3_CH3 event
10111	TMR4_TRGOUT event	10111	TMR3_TRGOUT event

11000	TMR6_TRGOUT event	11000	TMR3_CH1 event
11001	TMR3_CH4 event	11001	TMR6_TRGOUT event
11010	TMR4_CH1 event	11010	TMR4_CH4 event
11011	TMR1_TRGOUT event	11011	TMR1_CH3 event
11100	TMR9_CH1 event	11100	保留
11101	保留	11101	保留
11110	TMR7_TRGOUT event	11110	TMR7_TRGOUT event
11111	保留	11111	保留

18.4.2.3 采样与转换时序

用户可于 ADC_SPT1 与 ADC_SPT2 的 CSPTx 配置各个通道 (ADC_INx) 的采样周期。藉由配置 ADC_CTRL1 的 CRSEL 可改变转换数据的分辨率，较低的分辨率所需的转换时间较短。一次转换所需的时间可利用以下公式推得：

$$\text{一次转换所需的时间(ADCCLK 的周期)} = \text{采样时间} + \text{分辨率位数} + 0.5$$

示例：

CSPTx 选择 1.5 周期，CRSEL 选择 12 位，一次转换需要 $1.5+12.5=14$ 个 ADCCLK 周期。

18.4.3 转换顺序管理

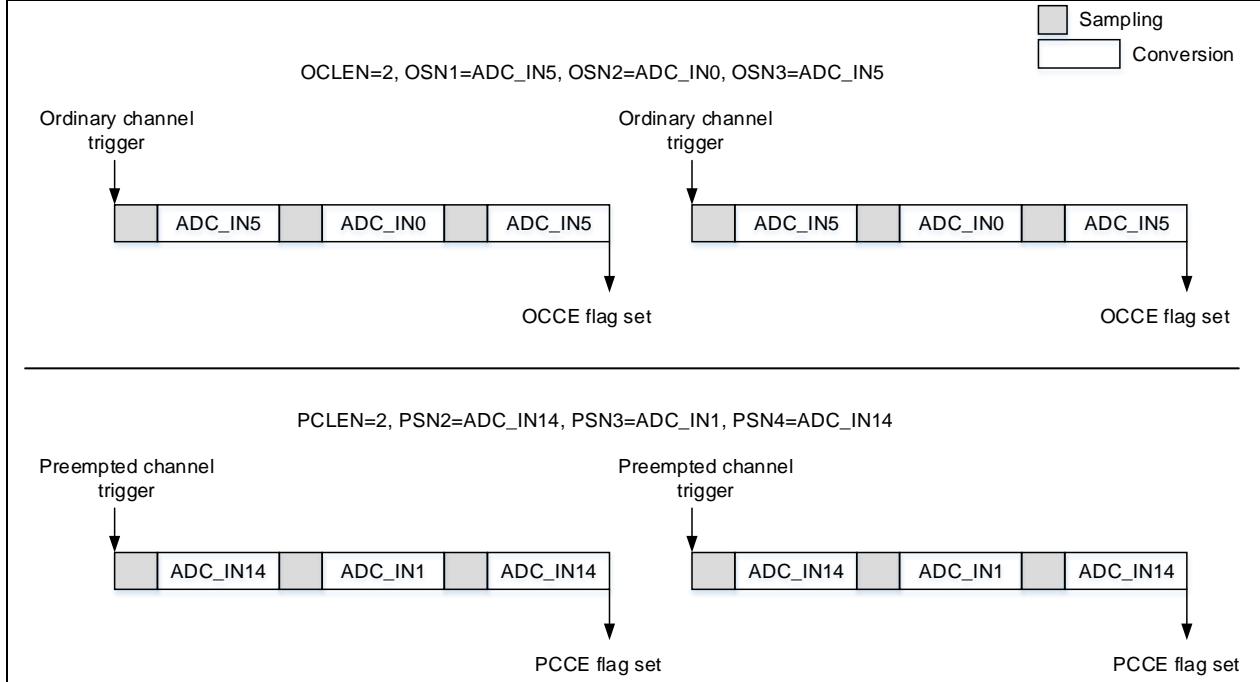
默认模式下，每次触发只会转换单个通道，即 OSN1 (普通触发) 或 PSN4 (抢占触发) 记录的通道。

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_CTRL1 的 SQEN，即开启序列模式，用户于 ADC_OSQx 配置普通通道顺序与总数，于 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 PSN_x 开始转换起， $x=4-PCLEN$ ，下图示范了序列模式的行为。

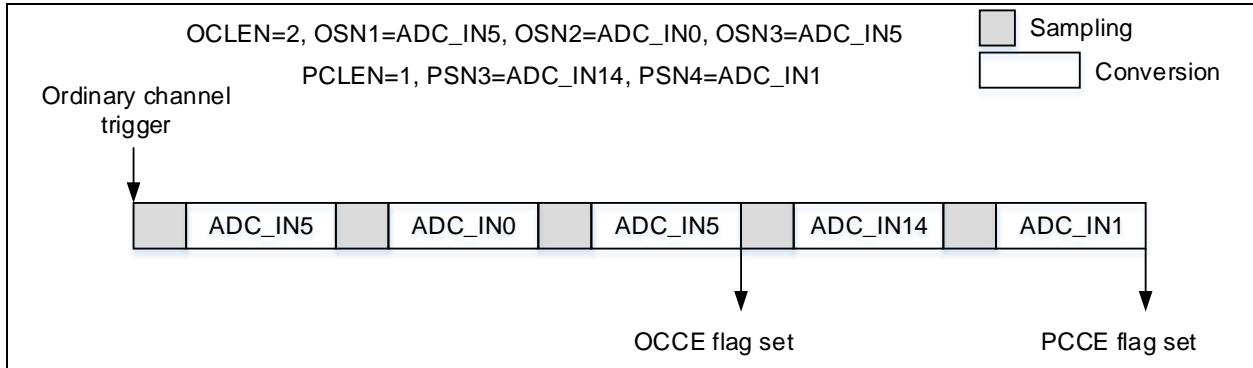
图 18-4 序列模式



18.4.3.2 抢占自动转换模式

使能 ADC_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。

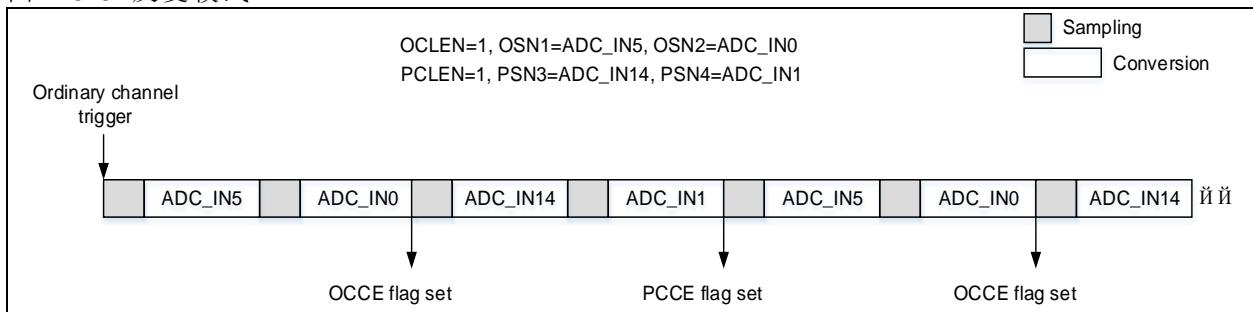
图 18-5 抢占自动转换模式



18.4.3.3 反复模式

使能 `ADC_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

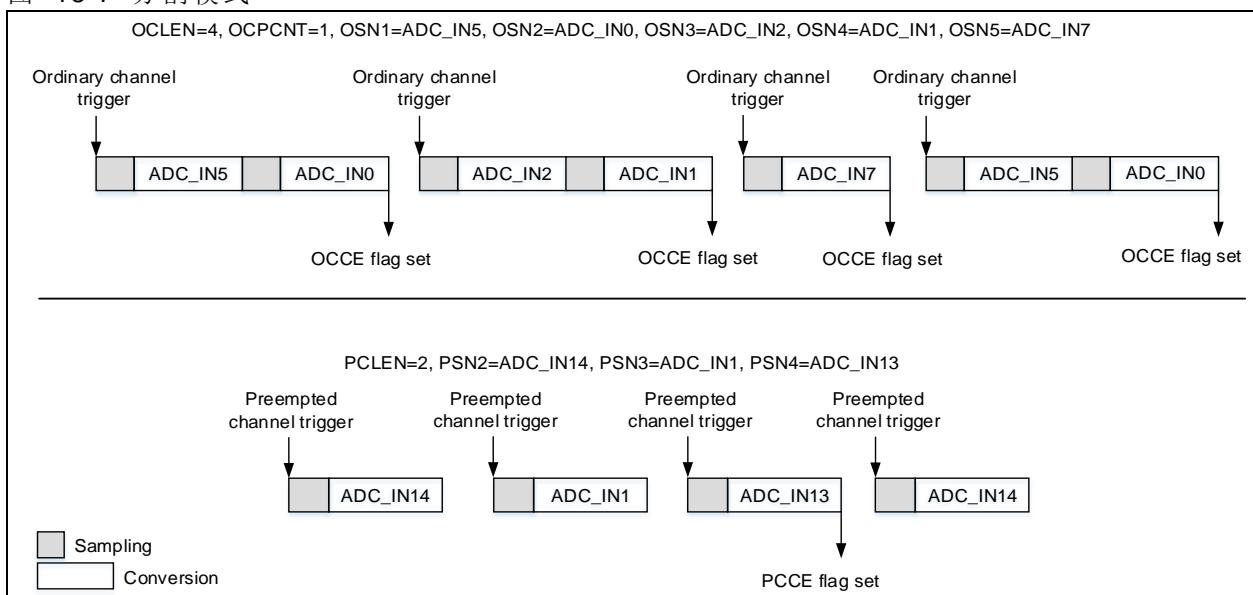


18.4.3.4 分割模式

使能 `ADC_CTRL1` 的 OCPEN，即开启普通通道的分割模式，此模式将 `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_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_CTRL1` 的 PCPEN，即开启抢占通道的分割模式，此模式将 `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。

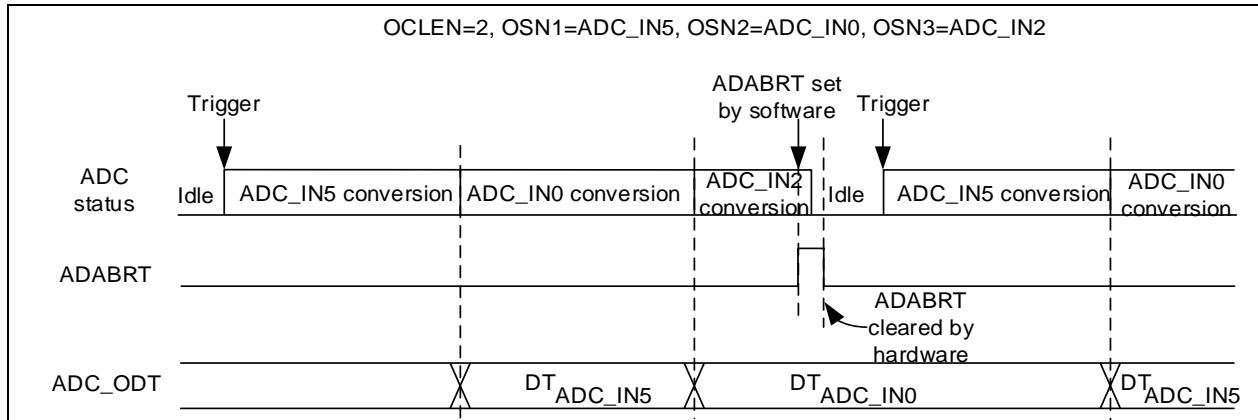
图 18-7 分割模式



18.4.4 转换中止

用户可利用 ADC_CTRL2 的 ADABRT 使 ADC 停止转换，转换顺序回归第一个通道，用户即可重新编排通道顺序，再次触发 ADC 将从头开始新序列的转换。若 ADABRT 已置位，需等待 ADABRT 被硬件清零后进行其他 ADC 操作。

图 18-8 ADABRT 时序



18.4.5 触发转换失败

ADC 转换发生异常时，TCF 标志置起，用于指示此次 ADC 转换失败，普通或抢占数据寄存器内的数据无效。默认情况下，ADC 不会因转换失败停止转换。

若要在转换失败时停止 ADC 转换，可在触发转换失败中断(通过 TCFIEN 位使能)中重新配置转换序列或通道。另外，同时开启触发转换失败中断和触发转换失败后自动终止转换(通过 TCFACA 位使能)功能后，发生转换失败时，ADC 会自动停止当前进行转换的序列或通道，并将 ADC 复位为空闲状态，之后的有效触发会重新触发 ADC 重新从第一个通道开始转换。

18.4.6 过采样器

一次过采样转换数据是透过转换多次相同通道，累加转换数据后作平均实现的。

- 由 ADC_OVSP 的 OSRSEL 选择过采样率，此位用来定义过采样倍数，通过多次转换同个通道实现。
- 由 ADC_OVSP 的 OSSSEL 选择过采样移位，此位用来定义平均系数，通过右移位实现。

若平均后数据大于 16 位，只取靠右 16 位数据，放入 16 位数据寄存器。如下表所示。

示例：

若 OSRSEL 选择 4 倍，一次过采样转换同个通道转换 4 次，并将这 4 次转换数据累加。若 OSSSEL 选择 6 位，累加数据除以 2^6 ，以四舍五入进位。

表 18-3 最大累加数据与过采样倍数及位移系数关系

过采样率	2x	4x	8x	16x	32x	64x	128x	256x
最大累加数据	0x1FFE	0x3FFC	0x7FF8	0xFFFF0	0x1FFE0	0x3FFC0	0x7FF80	0xFFFF00
不移位	0x1FFE	0x3FFC	0x7FF8	0xFFFF0	0xFFE0	0xFFC0	0xFF80	0xFF00
移 1 位	0x0FFF	0x1FFE	0x3FFC	0x7FF8	0xFFE0	0xFFE0	0xFFC0	0xFF00
移 2 位	0x0800	0x0FFF	0x1FFE	0x3FFC	0x7FF8	0xFFE0	0xFFE0	0xFFC0
移 3 位	0x0400	0x0800	0x0FFF	0x1FFE	0x3FFC	0x7FF8	0xFFE0	0xFFE0
移 4 位	0x0200	0x0400	0x0800	0x0FFF	0x1FFE	0x3FFC	0x7FF8	0xFFFF0
移 5 位	0x0100	0x0200	0x0400	0x0800	0x0FFF	0x1FFE	0x3FFC	0x7FF8
移 6 位	0x0080	0x0100	0x0200	0x0400	0x0800	0x0FFF	0x1FFE	0x3FFC
移 7 位	0x0040	0x0080	0x0100	0x0200	0x0400	0x0800	0x0FFF	0x1FFE
移 8 位	0x020	0x0040	0x0080	0x0100	0x0200	0x0400	0x0800	0x0FFF

使用过采样时，忽视 DTALIGN 与 PCDTOx，数据一律靠右摆放。切换 CRSEL 位改变过采样转换中每一次通道转换结果的分辨率，但是不改变过采样数据运算方式。

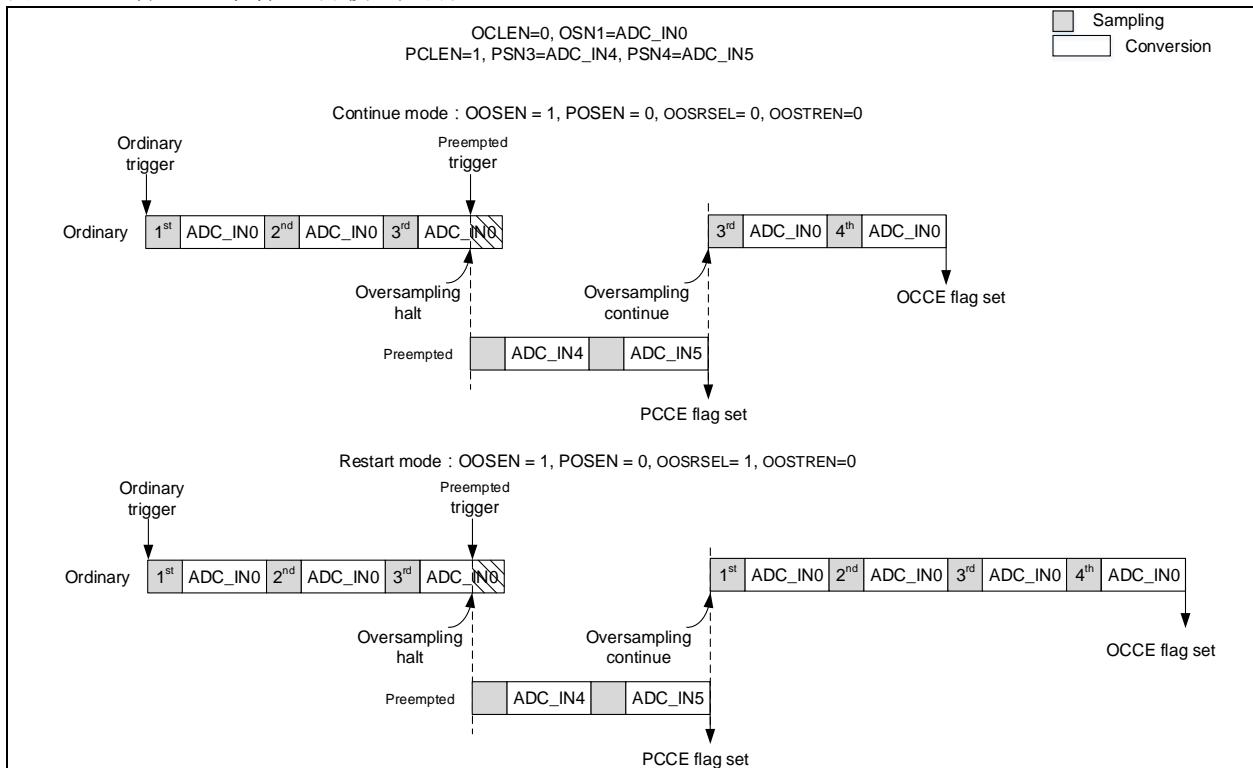
18.4.6.1 普通通道过采样

由 ADC_OVSP 的 OOSRSEL 控制普通过采样被打断后恢复的动作。

- OOSRSEL=0: 接续模式，普通过采样中途被抢占通道转换插入后，保留已累加的数据，再次开始转换时将从打断处转换。
- OOSRSEL=1: 重转模式。普通过采样中途被抢占通道转换插入后，累加的数据被清空。再次开始转换时是重新该通道的过采样转换。

下图以 4x 过采样率与序列模式下，示范了普通过采样接续模式与重转模式的差别。

图 18-9 普通过采样重转模式选择

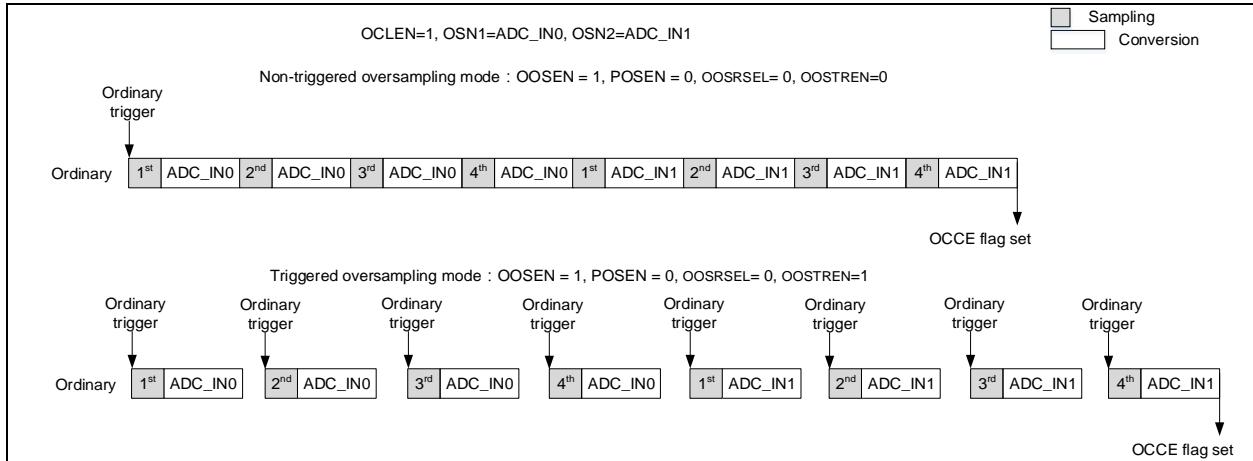


使能 ADC_OVSP 的 OOSTREN 即可使用触发模式。用户须触发普通过采样转换中每一次的普通通道转换。此模式下，中途被抢占通道触发打断后，须重新触发普通通道才会恢复转换普通通道过采样。

当触发模式与转换顺序管理模式搭配使用时，触发方式遵循触发模式，转换完成标志则遵循转换顺序管理模式。下图以 4x 过采样率与序列模式下，示范了普通过采样触发模式与恢复模式共用。

注意：触发模式与反复模式不可共用。

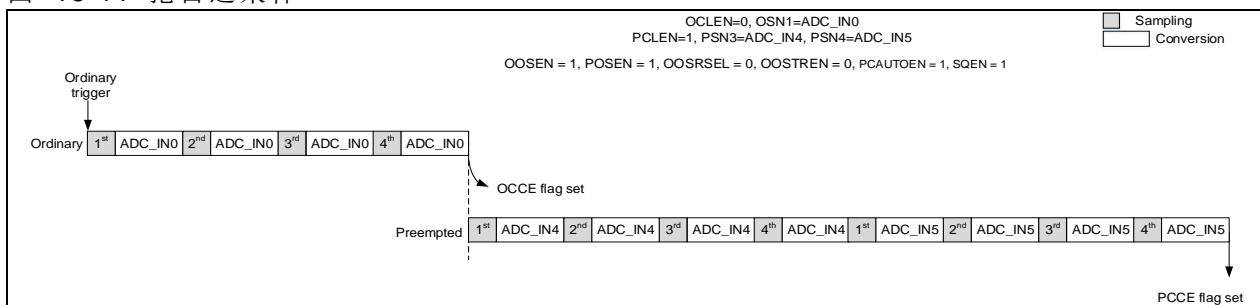
图 18-10 普通过采样触发模式



18.4.6.2 抢占通道过采样

抢占过采样可与普通过采样同时使用，也可分别使用。抢占过采样不影响到普通过采样的各种模式。下图以 4x 过采样率与抢占自动转换模式下，示范了抢占过采样与普通过采样触发模式共用。

图 18-11 抢占过采样



18.4.7 数据管理

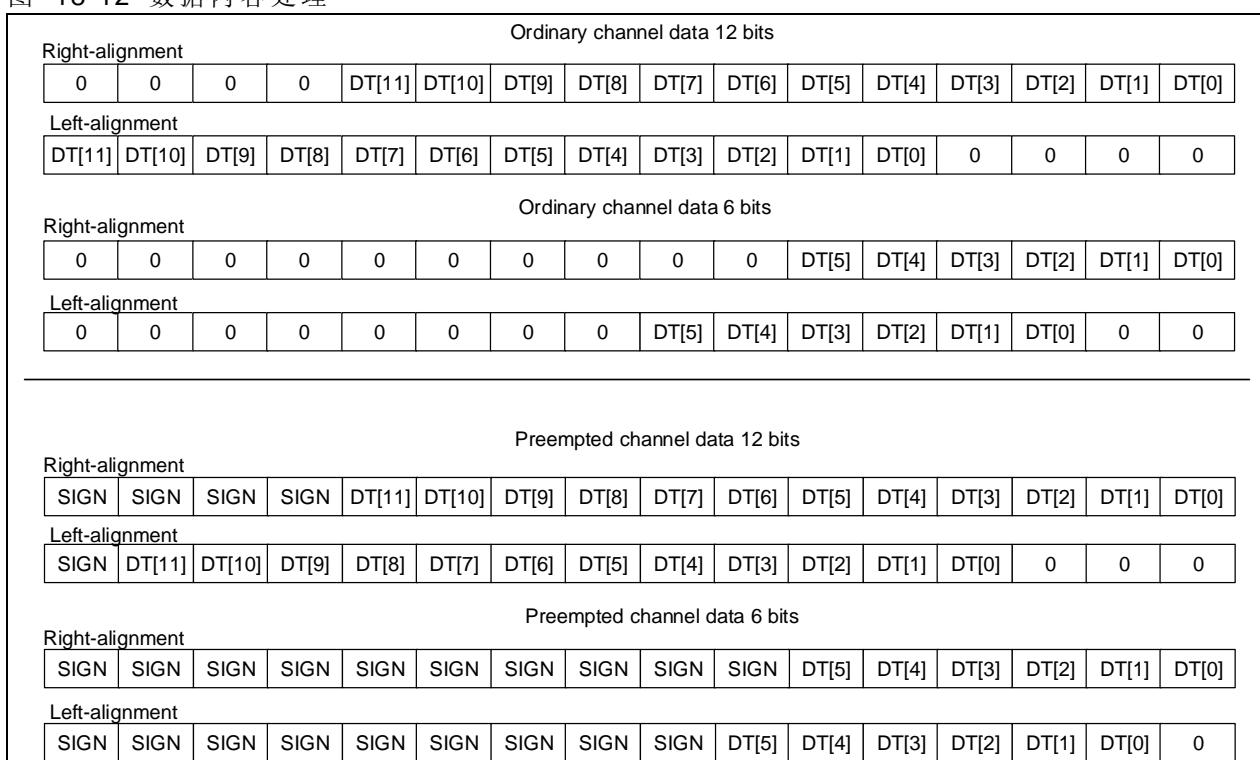
普通通道转换完成后数据存储于普通数据寄存器 (ADC_ODT)，抢占通道转换完成后数据存储于抢占数据寄存器 x (ADC_PDTx)。

18.4.7.1 数据内容处理

由 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抢占数据偏移寄存器 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。

分辨率 CRSEL 为 6 位时，数据存储方式以字节为基准摆放，其余皆以半字为基准摆放。如下图所示。

图 18-12 数据内容处理



18.4.7.2 数据获取

普通通道转换数据可藉由 CPU 或 DMA 读取普通数据寄存器 (ADC_ODT)，或通过 CPU 读取公共普通数据寄存器 x(ADC_i_ODTx)获得。每次普通通道转换完成，ADC_ODT 寄存器更新；普通序列中第 x 通道转换完成，ADC_i_ODTx 寄存器更新。ADC_ODT 寄存器与 ADC_i_ODTx 寄存器会同步更新。

抢占通道数据只可藉由 CPU 读取抢占数据寄存器 x (ADC_PDTx) 获得。

由 ADC_CTRL2 的 EOCSFEN 选择普通通道转换结束标志置位于序列结束或是每次普通数据寄存器更新时。当选择每次普通通道数据寄存器更新置位 OCCE 时，可判断 ADC_ODT 寄存器或 ADC_i_ODTx 寄存器是否更新。

使能 ADC_CTRL2 的 OCDMAEN 后，ADC 会在每次 ADC_ODT 寄存器更新时请求 DMA(ADCI_ODTx 寄存器不支持 DMA)。

当 EOCSFEN 或是 OCDMAEN 置位时，ADC 会自动开启溢出检测。若发生溢出事件，OCCO 溢出事件标志置起，ADC 转换停下，数据寄存器存储着最后一个有效数据。若使用 DMA，DMA 请求持续置起以请求 DMA 读取最后的有效数据。软件需将 OCCO 清除，并且重新触发 ADC，ADC 将从有效数据的下一个通道开始转换，如此即使中途发生溢出事件，所有被读取的数据皆是有效且按照顺序的。

配置 ADC_CTRL2 的 OCDRCEN 来选择是否要在 DMA 传输数量寄存器归零后持续请求 DMA。

18.4.8 电压监测

使能 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_VMH[11:0]寄存器或是小于低边界 ADC_VMLB[11:0]寄存器时，电压监测超出标志 VMOR 会置起。

电压监测功能使能后，ADC_VM 输出同时使能。ADC_VM 指示受监测通道转换结果是否超出边界寄存器设定值，超出时，ADC_VM 输出高电平；未超出时，ADC_VM 输出低电平。当未受监测通道转换结果超出边界寄存器设定值时，ADC_VM 输出电平不会变化。当关闭 ADC、关闭电压检测或 ADC 转换终止时 (ADABRT 或因触发转换失败而终止转换)，ADC_VM 输出复位为低电平，清除电压监测超出范围标志 (VMOR) 不会复位 ADC_VM 电平。

若选择 10 位分辨率，ADC_VMH[11:10]和 ADC_VMLB[11:10]要设置成 2'b00。

若选择 8 位分辨率，ADC_VMH[11:8]和 ADC_VMLB[11:8]要设置成 4'b0000。

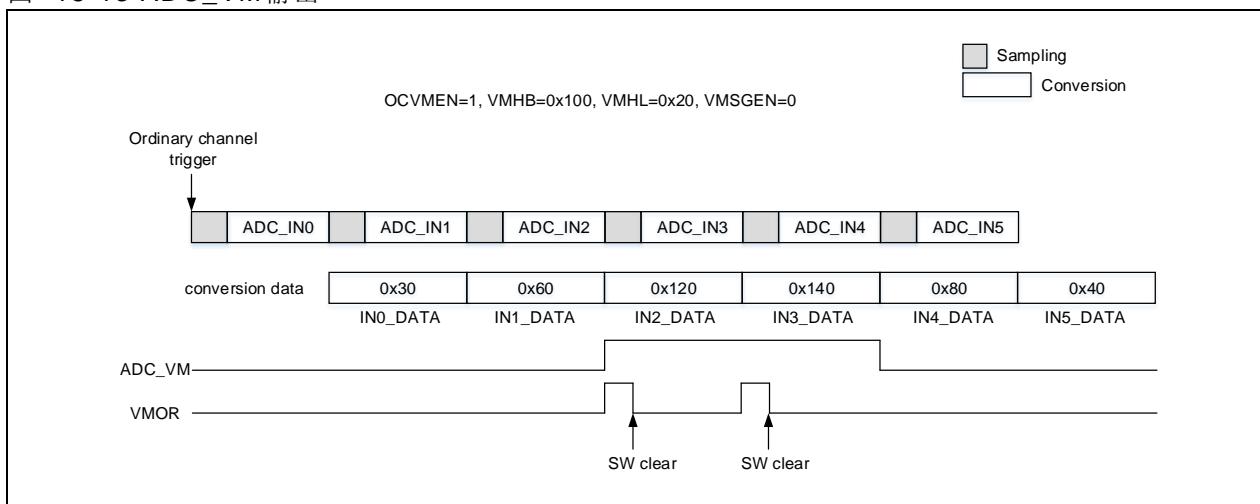
若选择 6 位分辨率，ADC_VMH[11:6]和 ADC_VMLB[11:6]要设置成 6'b000000。

透过 VMSGGEN 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 CRSEL、PCDTOx 与 DTALIGN 位的设定。

若是使用过采样器，则是以 ADC_VMH[15:0]与 ADC_VMLB[15:0]完整的 16 位寄存器与过采样数据作比较。

图 18-13 ADC_VM 输出



18.4.8.1 状态标志与中断

ADC 拥有自己的状态寄存器 (ADC_STS)：触发转换失败标志 (TCF)、准备就绪标志 (RDY)、溢出事件标志 (OCCO)、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)、普通通道转换结束标志 (OCCE) 及电压监测超出标志 (VMOR)。

两个 ADCx_STS 会映射到通用状态寄存器 (ADC_CSTS)，因此只需要读取通用状态寄存器即可一次获得两个 ADC 的状态。

其中触发转换失败标志、溢出事件标志、抢占通道组转换结束标志、普通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 主从模式

开启主从模式即可通过触发主机来联动从机进行通道转换，并且将通用普通数据寄存器作为获取主从 ADC 普通通道数据的单一接口。

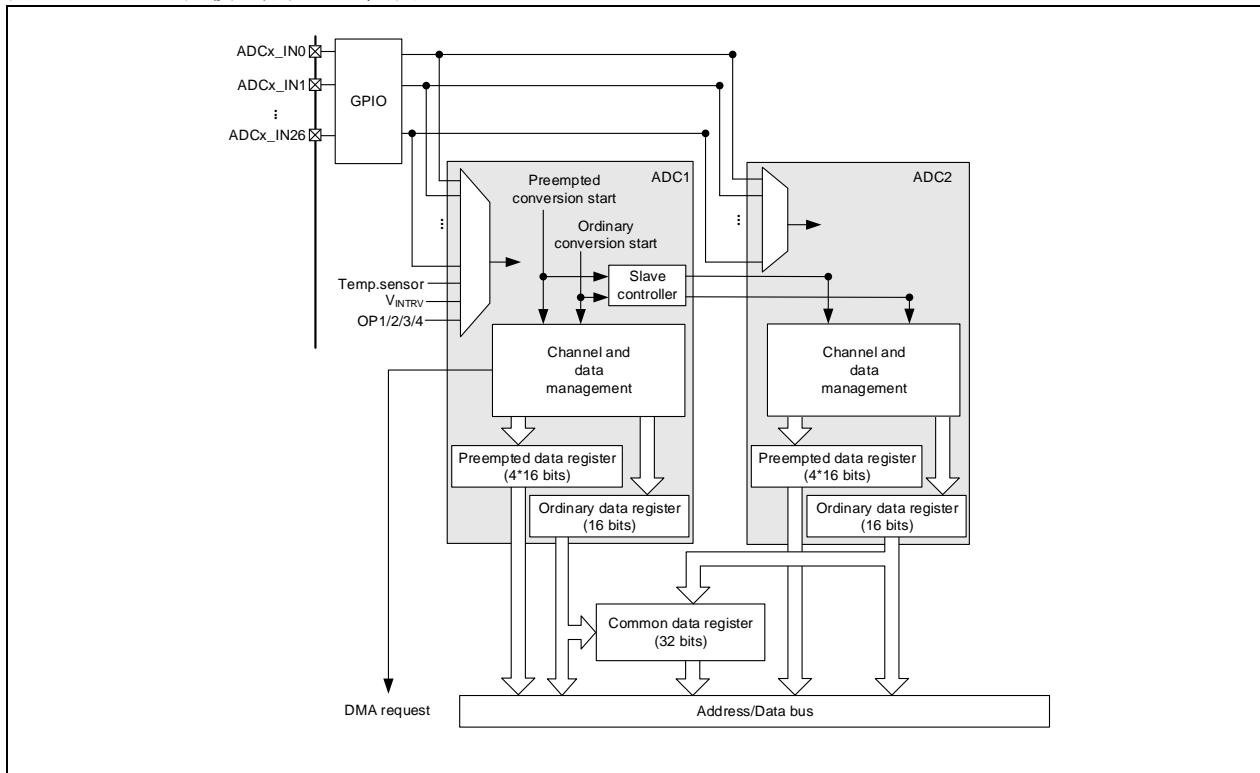
单从机主从模式以 ADC1 作为主机，ADC2 作为从机。主从模式下，需要同时使能主及从 ADC 的触发模式。

注意：转换中止 (ADABRT) 不可使用于主从模式，若在主从模式下想停止 ADC 的转换需清除每个 ADC 的 ADCEN。

注意：为避免主从间失去同步，主从机必须配置相同分辨率。

注意：若有同时使用多个 ADC 低解析度转换的需求，建议使用主从模式搭配 DMA1。

图 18-14 主从模式的 ADC 框图



18.5.1 数据管理

主从模式时，普通通道数据会共同存储于通用普通数据寄存器 ADC_CODT 中，存储方式透过 ADC_CCTRL 的 MSDMASEL 位配置，提供 3 种模式，如下表所示。只要 MSDMASEL 不为 0，就会在每次数据备齐时使用 ADC1 的 DMA 通道请求 DMA，主机与从机的溢出检测也会启动，发生溢出事件的 ADC 停下转换，同时 DMA 请求也停下，主机与从机可能因此失去同步，建议发生溢出事件后重新初始化 ADC 再触发。

表 18-4 主从模式 DMA 模式

MSDMASEL	主从模式	DMA 请求	ADC_CODT[31:0]
001	单从机	第一次	16 位 0 , ADC1_ODT[15:0]
		第二次	16 位 0 , ADC2_ODT [15:0]
		第三次	16 位 0 , ADC1_ODT [15:0]
010	单从机	第一次	ADC2_ODT[15:0], ADC1_ODT[15:0]
		第二次	ADC2_ODT[15:0], ADC1_ODT[15:0]
011	单从机	第一次	16 位 0 , ADC2_ODT[7:0], ADC1_ODT[7:0]
		第二次	16 位 0 , ADC2_ODT[7:0], ADC1_ODT[7:0]

配置 ADC_CCTRL 的 MSDRCEN 来决定 DMA 请求会一直持续不断直到没有数据可传输或是当 DMA 传输数量寄存器归零就停止。

18.5.2 同时模式

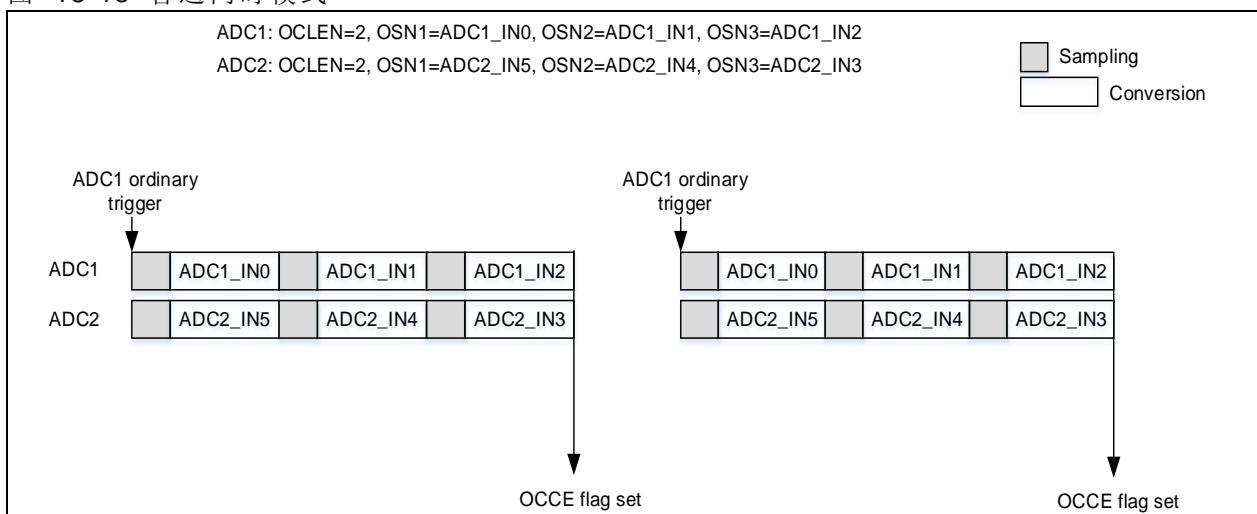
普通同时模式

配置 ADC_CCTRL 的 MSSEL 至普通同时模式后，可触发主机普通通道，使主机与从机同时转换普通通道。在此模式下，必须使用相同的采样时间以及相同的序列长度，以避免主从之间失去同步，遗失数据。下图示范了序列模式下的普通同时模式。

可搭配传输模式（MSDMASEL）中的模式 1、2 与 3。

注意：同样的通道不可同时被多个 ADC 采样，因此禁止将相同通道安排在不同 ADC 的同样序列位置。

图 18-15 普通同时模式

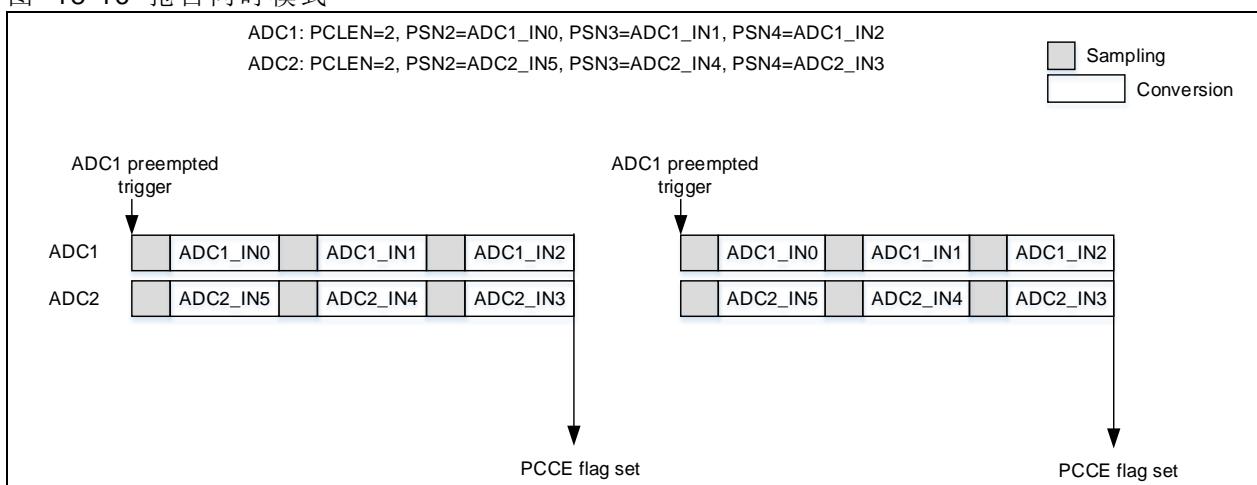


抢占同时模式

配置 ADC_CCTRL 的 MSSEL 至抢占同时模式后，可触发主机抢占通道，使主机与从机同时转换抢占通道。下图示范了序列模式下的抢占同时模式。

注意：同样的通道不可同时被多个 ADC 采样，因此禁止将相同通道安排在不同 ADC 的同样序列位置。

图 18-16 抢占同时模式



混合的普通同时+抢占同时模式

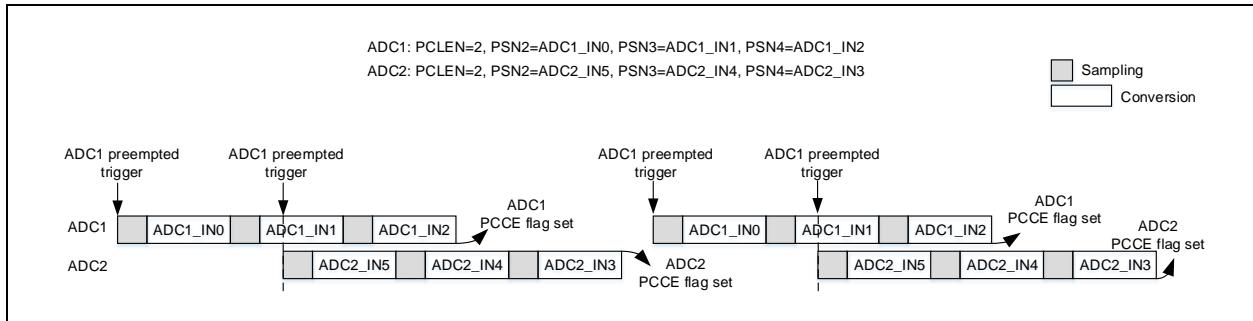
配置 ADC_CCTRL 的 MSSEL 至混合的普通同时+抢占同时模式后，可触发主机普通通道使主机与从机同时转换普通通道，也可触发主机抢占通道使主机与从机同时转换抢占通道。

18.5.3 抢占交错触发模式

抢占交错触发模式

配置 ADC_CCTRL 的 MSSEL 至抢占交错触发模式后，可多次触发主机的抢占通道，促使主从 ADC 轮流转换抢占通道。下图示范了序列模式下的抢占交错触发模式。

图 18-17 抢占交错触发模式



混合的普通同时+抢占交错触发模式

配置 ADC_CCTRL 的 MSSEL 至混合的普通同时+抢占交错触发模式后，可触发主机普通通道使主机与从机同时转换普通通道，也可多次触发主机的抢占通道促使主从 ADC 轮流转换抢占通道。

当普通通道转换被抢占通道触发打断，所有的 ADC 停下普通通道转换，其中一个 ADC 进入抢占通道转换，此时主机将无视抢占通道触发，直到普通通道恢复转换后才会再接受抢占通道触发。

18.5.4 普通位移模式

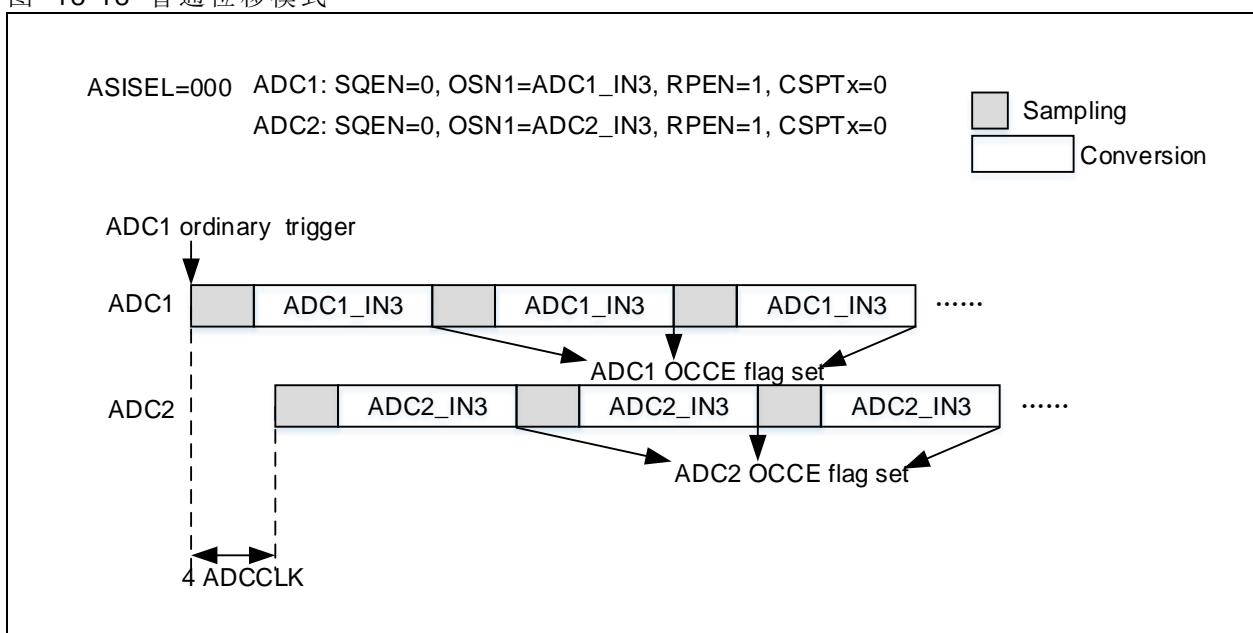
配置 ADC_CCTRL 的 MSSEL 至普通位移模式后，可触发主机普通通道，使 ADC 之间自动在普通通道的转换上时序位移。

配置 ADC_CCTRL 寄存器中的 ASISEL 位选择位移的长度。如下图所示。

这个模式硬件会保证不同 ADC 间的采样间隔至少 2.5 个 ADCCLK，因此当 ASISEL 的配置无法满足这个条件时，ASISEL 将无效。利用这个特性，将相同通道安排在不同 ADC 的同样序列位置时，也不用担心采样时间重叠。

可搭配传输模式 (MSDMASEL) 中的模式 1、2 与 3。

图 18-18 普通位移模式



注意：此模式下禁止抢占通道触发、禁止从机的普通通道触发。

18.6 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 18-5 ADC寄存器映像和复位值

寄存器简称	基址偏移量	复位值
ADC1_STS	0x000	0x0000 0000
ADC1_CTRL1	0x004	0x0000 0000
ADC1_CTRL2	0x008	0x0000 0000
ADC1_SPT1	0x00C	0x0000 0000
ADC1_SPT2	0x010	0x0000 0000
ADC1_PCDTO1	0x014	0x0000 0000
ADC1_PCDTO2	0x018	0x0000 0000
ADC1_PCDTO3	0x01C	0x0000 0000
ADC1_PCDTO4	0x020	0x0000 0000
ADC1_VMHB	0x024	0x0000 FFFF
ADC1_VMLB	0x028	0x0000 0000
ADC1_OSQ1	0x02C	0x0000 0000
ADC1_OSQ2	0x030	0x0000 0000
ADC1_OSQ3	0x034	0x0000 0000
ADC1_PSQ	0x038	0x0000 0000
ADC1_PDT1	0x03C	0x0000 0000
ADC1_PDT2	0x040	0x0000 0000
ADC1_PDT3	0x044	0x0000 0000
ADC1_PDT4	0x048	0x0000 0000
ADC1_ODT	0x04C	0x0000 0000
ADC1_SPT3	0x050	0x0000 0000
ADC1_OVSP	0x080	0x0000 0000
ADC2_STS	0x100	0x0000 0000
ADC2_CTRL1	0x104	0x0000 0000
ADC2_CTRL2	0x108	0x0000 0000
ADC2_SPT1	0x10C	0x0000 0000
ADC2_SPT2	0x110	0x0000 0000
ADC2_PCDTO1	0x114	0x0000 0000
ADC2_PCDTO2	0x118	0x0000 0000
ADC2_PCDTO3	0x11C	0x0000 0000
ADC2_PCDTO4	0x120	0x0000 0000
ADC2_VMHB	0x124	0x0000 FFFF
ADC2_VMLB	0x128	0x0000 0000
ADC2_OSQ1	0x12C	0x0000 0000
ADC2_OSQ2	0x130	0x0000 0000
ADC2_OSQ3	0x134	0x0000 0000
ADC2_PSQ	0x138	0x0000 0000
ADC2_PDT1	0x13C	0x0000 0000
ADC2_PDT2	0x140	0x0000 0000
ADC2_PDT3	0x144	0x0000 0000
ADC2_PDT4	0x148	0x0000 0000
ADC2_ODT	0x14C	0x0000 0000
ADC2_OVSP	0x180	0x0000 0000
ADC_CSTS	0x300	0x0000 0000
ADC_CCCTRL	0x304	0x0000 0000
ADC_CODT	0x308	0x0000 0000
ADC1_ODT1	0x310	0x0000 0000
ADC1_ODT2	0x314	0x0000 0000
ADC1_ODT3	0x318	0x0000 0000
ADC1_ODT4	0x31C	0x0000 0000
ADC1_ODT5	0x320	0x0000 0000
ADC1_ODT6	0x324	0x0000 0000
ADC1_ODT7	0x328	0x0000 0000
ADC1_ODT8	0x32C	0x0000 0000
ADC1_ODT9	0x330	0x0000 0000
ADC1_ODT10	0x334	0x0000 0000
ADC1_ODT11	0x338	0x0000 0000
ADC1_ODT12	0x33C	0x0000 0000

ADC1_ODT13	0x340	0x0000 0000
ADC1_ODT14	0x344	0x0000 0000
ADC1_ODT15	0x348	0x0000 0000
ADC1_ODT16	0x34C	0x0000 0000
ADC2_ODT1	0x350	0x0000 0000
ADC2_ODT2	0x354	0x0000 0000
ADC2_ODT3	0x358	0x0000 0000
ADC2_ODT4	0x35C	0x0000 0000
ADC2_ODT5	0x360	0x0000 0000
ADC2_ODT6	0x364	0x0000 0000
ADC2_ODT7	0x368	0x0000 0000
ADC2_ODT8	0x36C	0x0000 0000
ADC2_ODT9	0x370	0x0000 0000
ADC2_ODT10	0x374	0x0000 0000
ADC2_ODT11	0x378	0x0000 0000
ADC2_ODT12	0x37C	0x0000 0000
ADC2_ODT13	0x380	0x0000 0000
ADC2_ODT14	0x384	0x0000 0000
ADC2_ODT15	0x388	0x0000 0000
ADC2_ODT16	0x38C	0x0000 0000

18.6.1 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: 8	保留	0x00000000	resd	请保持默认值。
位 7	TCF	0x0	rw0c	ADC 触发转换失败标志(Trigger convert fail flag) 当次触发转换失败 0: 转换成功 1: 转换失败
位 6	RDY	0x0	ro	ADC 准备就绪标志 (ADC ready to conversion flag) 只读位，该位在 ADC 上电完毕后由硬件置位。 0: 未就绪； 1: 已就绪。
位 5	OCCO	0x0	rw0c	普通通道转换溢出标志 (Ordinary channel conversion overflow flag) 该位被硬件置起，由软件将其清零（对自身写零） 0: 未发生溢出 1: 发生溢出 注：溢出检测仅在使能 DMA 传输或者 EOCSFEN = 1 时有效。
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	OCCE	0x0	rw0c	普通通道转换结束标志 (Ordinary 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_ODT 寄存器清零。 0: 未结束； 1: 已结束。 注：读取 ADC _i _ODTx 寄存器不会清 0 OCCE 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。
-----	------	-----	------	--

18.6.2 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: 29	保留	0x00	resd	请保持默认值。
位 28	TCFACA	0x0	rw	触发转换失败后自动终止转换(Trigger conversion fail auto conversion abort) 0: 关闭; 1: 开启。 注: 只可在非组合模式下开启此功能, 并且也只能在 TCFIEN 开启状态下使用此功能。
位 27	TCFIEN	0x0	rw	触发转换失败中断使能(Trigger conversion fail interrupt enable) 0: 关闭; 1: 开启。
位 26	OCCOIE	0x0	rw	普通通道转换溢出中断使能 (Ordinary channel conversion overflow interrupt enable) 0: 关闭; 1: 开启。
位 25:24	CRSEL	0x0	rw	转换分辨率选择 (Conversion resolution select) 00: 12 位; 01: 10 位; 10: 8 位; 11: 6 位。
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。 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 9	VMSGEN	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：如果开启了多通道模式，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 OCCE 或 PCCE 中断。
位 8	SQEN	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 7	PCCEIEN	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 6	VMORIEN	0x0	rw	普通通道转换结束中断使能（Ordinary channel conversion end interrupt enable） 0: 关闭； 1: 开启。
位 5	OCCEIEN	0x0	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； 10001: ADC_IN17 通道； 10100: ADC_IN20 通道； 10101: ADC_IN21 通道； 11010: ADC_IN26 通道。 11011~11111: 未用，禁止配置。
位 4: 0	VMCSEL	0x00	rw	

18.6.3 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: 26	保留	0x00	resd	请保持默认值。
位 30	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 29:28	OCETE	0x0	rw	普通通道组的外部触发边沿选择（Ordinary channel's external trigger edge select） 00: 禁止边沿触发； 01: 上升沿触发； 10: 下降沿触发； 11: 任意边沿触发。
位 31 位 27:24	OCTESEL	0x00	rw	普通通道组转换的触发事件选择（Ordinary channel's conversion trigger event select） 注：各 bit 的定义参考 18.4.2.2 章节 。
位 22	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）。

				抢占通道组的外部触发边沿选择（Preempted channel's external trigger edge select） 00: 禁止边沿触发； 01: 上升沿触发； 10: 下降沿触发； 11: 任意边沿触发。
位 21:20	PCETE	0x0	rw	抢占通道组转换的触发事件选择（Preempted channel's conversion trigger event select） 注：各 bit 的定义参考 18.4.2.2 章节 。
位 23 位 19:16	PCTESEL	0x00	rw	抢占通道组转换的触发事件选择（Preempted channel's conversion trigger event select） 注：各 bit 的定义参考 18.4.2.2 章节 。
位 15:12	保留	0x0	resd	请保持默认值。
位 11	DTALIGN	0x0	rw	数据对齐方式（Data alignment） 0: 右对齐； 1: 左对齐。
位 10	EOCSFEN	0x0	rw	每个普通通道转换置位 OCCE 标志使能（Each ordinary channel conversion set OCCE flag enable） 0: 关闭； 1: 开启。 注：开启此位时，溢出检测被自动使能。
位 9	OCDRCEN	0x0	rw	普通通道数据的 DMA 请求接续使能（Ordinary channel's DMA request continuation enable for independent mode） 0: 关闭（传输完 DMA 设定个数后，普通通道转换完毕不会再产生 DMA 请求）； 1: 开启（不关心 DMA 设定的个数，每个普通通道转换完毕均产生 DMA 请求）。 注：此位仅作用于设定为非主从模式，且 OCDMAEN = 1 时。
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: 5	保留	0x0	resd	请保持默认值。
位 4	ADABRT	0x0	rw	ADC 转换中止（ADC conversion abort） 0: 无中止转换命令执行； 1: 中止当前正在执行的转换。 注：当转换已中止时硬件将该位清零。该位被清零后，可重新进行触发转换。若 ADABRT 已置位，需等待 ADABRT 被硬件清零后进行其他 ADC 操作。
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。应用程序需注意，在转换器上电至转换开始有一个延迟 t_{STAB} 。

18.6.4 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	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。
位 20: 18	CSPT16	0x0	rw	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。
位 17: 15	CSPT15	0x0	rw	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。
位 14: 12	CSPT14	0x0	rw	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。
位 11: 9	CSPT13	0x0	rw	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。
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。

18.6.5 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。
--------	-------	-----	----	--

18.6.6 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

18.6.7 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: 16	保留	0x0000	resd	请保持默认值。
位 15: 0	VMHB	0xFFFF	rw	电压监测高边界设定 (Voltage monitoring high boundary)

18.6.8 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: 16	保留	0x0000	resd	请保持默认值。
位 15: 0	VMLB	0x0000	rw	电压监测低边界设定 (Voltage monitoring low boundary)

18.6.9 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	0x00	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 0~26, 示例: 设定为 3 就代表第 13 个转换的是 ADC_IN3 通道。

18.6.10 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 8th 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 0~26，示例：设定为 8 就代表第 7 个转换的是 ADC_IN8 通道。

18.6.11 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 0~26，示例：设定为 17 就代表第 1 个转换的是 ADC_IN17 通道。

18.6.12 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 0~26，比如设定为 4 时其代表的就是 ADC_IN4 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]=100110 001010 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。

18.6.13 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)

18.6.14 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)

18.6.15 ADC采样时间寄存器3 (ADC_SPT3)

访问: 字访问

域	简称	复位值	类型	功能
位 31: 21	保留	0x0	resd	请保持默认值。
位 20: 18	CSPT26	0x0	rw	选择 ADC_IN26 通道的采样时间 (Selection sample time of channel ADC_IN26) 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	CSPT25	0x0	rw	选择 ADC_IN25 通道的采样时间 (Selection sample time of channel ADC_IN25) 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	CSPT24	0x0	rw	选择 ADC_IN24 通道的采样时间 (Selection sample time of channel ADC_IN24) 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	CSPT23	0x0	rw	选择 ADC_IN23 通道的采样时间 (Selection sample time of channel ADC_IN23) 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	CSPT22	0x0	rw	选择 ADC_IN22 通道的采样时间 (Selection sample time of channel ADC_IN22) 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	CSPT21	0x0	rw	选择 ADC_IN21 通道的采样时间 (Selection sample time of channel ADC_IN21) 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	CSPT20	0x0	rw	选择 ADC_IN20 通道的采样时间 (Selection sample time of channel ADC_IN20) 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。

18.6.16 ADC过采样寄存器 (ADC_OVSP)

访问: 字访问

域	简称	复位值	类型	功能
位 31: 11	保留	0x0000	resd	请保持默认值。
位 10	OOSRSEL	0x0	rw	普通过采样重转模式选择 (Ordinary oversampling restart mode select) 当普通过采样中途被抢占通道转换插入后，依据如下设定，选择普通过采样从哪里开始恢复转换。 0: 继续模式（普通过采样缓冲区会被保留）； 1: 重转模式（普通过采样缓冲区会被清零，即当前通道之前的采样次数被清零）。
位 9	OOSTREN	0x0	rw	普通过采样触发模式使能 (Ordinary oversampling trigger mode enable) 0: 关闭（通道的所有过采样转换仅需一次触发）； 1: 开启（通道的每个过采样转换均需进行触发）。
位 8: 5	OSSSEL	0x0	rw	过采样移位选择 (Oversampling shift select) 此位定义应用到最终过采样结果的右移位数。 0000: 不进行移位； 0001: 移 1 位； 0010: 移 2 位； 0011: 移 3 位； 0100: 移 4 位； 0101: 移 5 位； 0110: 移 6 位； 0111: 移 7 位； 1000: 移 8 位；

				1001~1111: 未用, 禁止配置。 过采样率选择 (Oversampling ratio select) 000: 2x; 001: 4x; 010: 8x; 011: 16x; 100: 32x; 101: 64x; 110: 128x; 111: 256x。
位 4: 2	OSRSEL	0x0	rw	抢占过采样使能 (Preempted oversampling enable) 0: 关闭; 1: 开启。
位 0	OSEN	0x0	rw	普通过采样使能 (Ordinary oversampling enable) 0: 关闭; 1: 开启。

18.6.17 ADC通用状态寄存器 (ADC_STS)

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	TCF2	0x0	ro	ADC2 触发转换失败标志(Trigger convert fail flag) 当次触发转换失败 0: 转换成功 1: 转换失败
位 14	RDY2	0x0	ro	ADC2 的准备就绪标志(ADC ready to conversion flag of ADC2) 该位是 ADC2_STS 寄存器中 RDY 位的映射。
位 13	OCCO2	0x0	ro	ADC2 的普通通道转换溢出标志(Ordinary channel conversion overflow flag of ADC2) 该位是 ADC2_STS 寄存器中 OCCO 位的映射。
位 12	OCCS2	0x0	ro	ADC2 的普通通道转换开始标志(Ordinary channel conversion start flag of ADC2) 该位是 ADC2_STS 寄存器中 OCCS 位的映射。
位 11	PCCS2	0x0	ro	ADC2 的抢占通道转换开始标志(Preempted channel conversion start flag of ADC2) 该位是 ADC2_STS 寄存器中 PCCS 位的映射。
位 10	PCCE2	0x0	ro	ADC2 的抢占通道组转换结束标志(Preempted channels conversion end flag of ADC2) 该位是 ADC2_STS 寄存器中 PCCE 位的映射。
位 9	OCCE2	0x0	ro	ADC2 的普通通道转换结束标志(Ordinary channels conversion end flag of ADC2) 该位是 ADC2_STS 寄存器中 OCCE 位的映射。
位 8	VMOR2	0x0	ro	ADC2 的电压检测超出范围标志 (Voltage monitoring out of range flag of ADC2) 该位是 ADC2_STS 寄存器中 VMOR 位的映射。
位 7	TCF1	0x0	ro	ADC1 触发转换失败标志(Trigger convert fail flag) 当次触发转换失败 0: 转换成功 1: 转换失败
位 6	RDY1	0x0	ro	ADC1 的准备就绪标志(ADC ready to conversion flag of ADC1) 该位是 ADC1_STS 寄存器中 RDY 位的映射。
位 5	OCCO1	0x0	ro	ADC1 的普通通道转换溢出标志(Ordinary channel conversion overflow flag of ADC1) 该位是 ADC1_STS 寄存器中 OCCO 位的映射。
位 4	OCCS1	0x0	ro	ADC1 的普通通道转换开始标志(Ordinary channel conversion start flag of ADC1) 该位是 ADC1_STS 寄存器中 OCCS 位的映射。
位 3	PCCS1	0x0	ro	ADC1 的抢占通道转换开始标志(Preempted channel conversion start flag of ADC1)

				该位是 ADC1_STS 寄存器中 PCCS 位的映射。
位 2	PCCE1	0x0	ro	ADC1 的抢占通道组转换结束标志(Preempted channels conversion end flag of ADC1) 该位是 ADC1_STS 寄存器中 PCCE 位的映射。
位 1	OCCE1	0x0	ro	ADC1 的普通通道转换结束标志(Ordinary channels conversion end flag of ADC1) 该位是 ADC1_STS 寄存器中 OCCE 位的映射。
位 0	VMOR1	0x0	ro	ADC1 的电压检测超出范围标志 (Voltage monitoring out of range flag of ADC1) 该位是 ADC1_STS 寄存器中 VMOR 位的映射。

18.6.18 ADC通用控制寄存器 (ADC_CCTRL)

访问: 字访问

域	简称	复位值	类型	功能
位 31: 24	保留	0x0000	resd	请保持默认值。
位 23	ITSRVEN	0x0	rw	内部温度传感器及 V _{INTRV} 使能 (Internal temperature sensor and V _{INTRV} enable) 0: 关闭; 1: 开启。
位 22: 20	保留	0x0	resd	请保持默认值。
位 19: 16	ADCDIV	0x0	rw	ADC 分频因子 (ADC division) 0000: HCLK 2 分频 0001: HCLK 3 分频 ... 1111: HCLK 17' 分频 注: 经此分频后的时钟将被所有 ADC 共用; ADCCLK 额定最大为 35MHz, 经分频后 ADCCLK 不能高于 PCLK2。
位 28	MSDMASEL	0x0	rw	主从模式下普通通道数据的 DMA 传输模式选择 (Ordinary channel's DMA transfer mode select for master slave mode) MSDMASEL[2]位于 ADC_CCTRL 寄存器的第 28 位, 其中 MSDMASEL[2: 0]定义如下:
位 15: 14				000: 禁止 DMA 传输; 001: 使用 DMA 模式 1; 010: 使用 DMA 模式 2; 011: 使用 DMA 模式 3; 100~111: 未用, 禁止配置。 注: 此位仅作用于设定为主从模式时, 具体 DMA 模式 x 的使用规则请参考 18.5.1 章节 描述。
位 13	MSDRcen	0x0	rw	主从模式下普通通道数据的 DMA 请求接续使能 (Ordinary channel's DMA request continuation enable for master slave mode) 0: 关闭 (传输完 DMA 设定个数后, 普通通道转换完毕不会再产生 DMA 请求); 1: 开启 (不关心 DMA 设定的个数, 每个普通通道转换完毕均产生 DMA 请求)。 注: 此位仅作用于设定为主从模式时, 且 MSDMASEL 选择为 DMA 模式 x 时。
位 12	保留	0x0	resd	请保持默认值。
位 11: 8	ASISEL	0x0	rw	普通位移模式下相邻 ADC 间的采样间隔时间选择 (Adjacent ADC sampling interval select for ordinary shifting mode) 0000: 4 * TADCCLK 0001: 5 * TADCCLK 0010: 6 * TADCCLK ... 1111: 19 * TADCCLK 注: 这些位用于设定普通位移模式下相邻 ADC 转换间的时间间隔, 具体间隔规则请参考 18.5.4 章节 描述。
位 7: 5	保留	0x0	resd	请保持默认值。

位 4: 0 MSSEL 0x00

rw

主从组合模式选择 (Master slave mode select)

00000: 非组合模式

00001: 混合的普通同时+抢占同时模式;

00010: 混合的普通同时+抢占交错触发模式;

00011~00100: 未用, 禁止配置;

00101: 抢占同时模式;

00110: 普通同时模式;

00111: 普通位移模式;

01000: 未用, 禁止配置;

01001: 抢占交错触发模式;

01010~11111: 未用, 禁止配置;

注: 在主从组合模式中, 修改配置会导致主从丢失同步规则。建议在修改前先关闭主从组合模式。

18.6.19 ADC通用普通数据寄存器 (ADC_CODT)

访问: 字访问

域	简称	复位值	类型	功能
位 31: 16	CODTH	0x0000	rw	主从模式下普通转换的高半字数据 (Ordinary conversion's high halfword data for master slave mode) 注: 在不同的 DMA 模式中, 这些位代表的数据含义存在区别。具体数据存储规则请参考 18.5.1 章节描述 。
位 15: 0	CODTL	0x0000	rw	主从模式下普通转换的低半字数据 (Ordinary conversion's low halfword data for master slave mode) 注: 在不同的 DMA 模式中, 这些位代表的数据含义存在区别。具体数据存储规则请参考 18.5.1 章节描述 。

18.6.20 ADC*i*公共普通数据寄存器x(ADC*i*_ODTx,*i*=1/2,*x*=1~16)

访问: 字访问

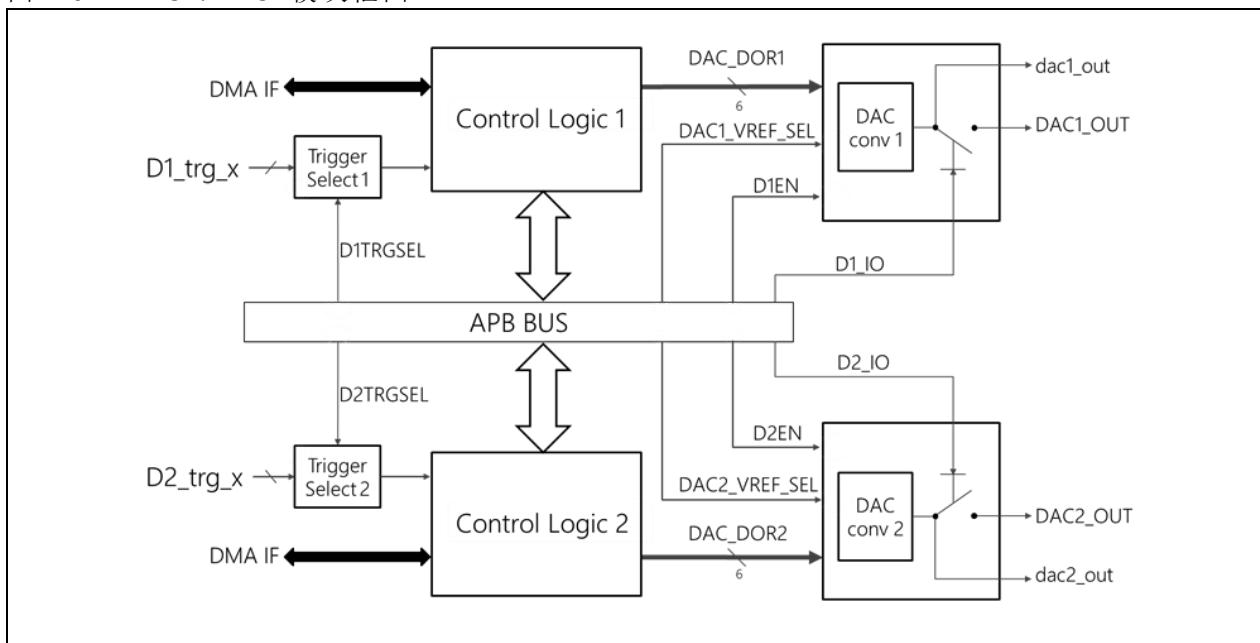
域	简称	复位值	类型	功能
位 31: 16	保留	0x0000	resd	请保持默认值。
位 15: 0	ODTx	0x0000	ro	普通通道的转换数据 (Conversion data of ordinary channel) 存放序列中第 <i>x</i> 个转换的普通通道数据(<i>x</i> =1~16) 注: 此寄存器不支持 DMA, 读取此寄存器不会清除 OCCE 标志。

19 数字/模拟转换 (DAC)

19.1 简介

数模转换器 (DAC) 采用 6 位数字输入，产生 0 至参考电压之间的模拟输出，主要提供电压给内部比较器、运算放大器使用，可透过配置寄存器使能输出至 GPIO。数字部分可以支持单/双 DAC，同时可以与 DMA 配合使用。两个 DAC1/DAC2 各有一个数模转换器，每个 DAC1/DAC2 可以独立进行数模转换，也可以双 DAC 同时触发进行转换，参考电压可选用系统电压 VDDA 或数入参考电压 V_{REF+} 可以使转换操作更加精确。

图 19-1 DAC1/DAC2模块框图



19.2 主要特性

- 单/双 DAC 6 位数字输入
- 双 DAC 或者单个 DAC1/DAC2 独立转换
- 每个 DAC1/DAC2 支持 DMA 模式
- 软件触发或者外部触发转换
- 支持切换系统电压 VDDA 或输入参考电压 V_{REF+}

19.3 设计提示

DAC 有以下提示仅供设计参考。

- 模拟模块配置
- DAC1/DAC2 的模拟部分由 DAC 控制寄存器 (DAC_CTRL) DxEN 位控制开启，数字部分则不受该位控制。
- DMA 功能
- 任一 DAC1/DAC2 支持 DMA 功能，通过设置 DAC 控制寄存器 (DAC_CTRL) 的 DxDMAEN 位使能 DMA 请求。当触发使能位 DxTRGEN 有效，触发信号有效时，即产生 DMA 请求。DAC 的 DMA 请求不会累计，未来得及处理的 DMA 请求将被忽略，也不会产生错误信息。
- 在双 DAC 模式下，程序可以只使用一个 DMA 请求，一个 DMA 通道的情况下，处理工作在双 DAC 模式的 2 个 DAC1/DAC2。
- DMA 下溢处理
- 当开启 DAC 的 DMA 功能，如果第二个外部触发到达时尚未收到 DMA 反馈的第一个外部触发的确认信号，即产生 DMA 下溢。此时将不会处理新的外部触发，并且不会发出

新的 DMA 请求，同时 DAC 状态寄存器 (DAC_STS) DxDMAUDRF 位指示发生 DMA 下溢错误，配置 DAC 控制寄存器 (DAC_CTRL) DxDMAUDRIEN 位为 ‘1’，将产生相应的 DMA 下溢错误中断。

- 软件通过写入 ‘1’ 来清除 DAC 状态寄存器 (DAC_STS) DxDMAUDRF 位，在重新开始 DMA 传输之前，应该先清除 DAC 控制寄存器 (DAC_CTRL) 的 DxDMAEN 位，然后重新初始化 DMA 和 DAC。
- 输入输出配置
- 数字输入经过 DAC 线性地转换为模拟电压输出，其范围为 0 至参考电压。
- 模拟 DAC 模块采用 VDDA 供电，输入正模拟参考电压可配置 DACx_VREF_SEL 来选用 VDDA 或 V_{REF+}。
- DAC 本身没有输出增益，主要提供电压给内部比较器或运算放大器使用，可配置 D1_IO 或 D2_IO 位来使能 DAC1 或 DAC2 输出到 PA4 或 PA5 上。
- 当 PA4 或者 PA5 作为模拟输出时，为避免寄生干扰和额外的功耗，需设置为模拟模式。
- DAC 输出 = V_{REF+} × (DxDOT[5: 0]/ 63) 或 VDDA × (DxDOT[5: 0]/ 63)
- DAC I/O test-mode 注意：
- DAC 转换频率：100K~300Khz
- 电压表内阻抗: >10MΩ
- 禁止对 DAC 负载施加电容性负载

19.4 功能描述

19.4.1 触发事件

如果 DAC_CTRL 寄存器的 DxTRGEN 位被置 1，DAC 转换可以由某外部事件（定时器计数器、外部中断线）或者软件触发，触发事件源由 DxTRGSEL[2: 0] 进行选择。

表 19-1 触发源选择

触发源	DxTRGSEL [2: 0]	说明
TMR6_TRGOUT	000	片上信号
TMR3_TRGOUT	001	
TMR7_TRGOUT	010	
TMR9_TRGOUT	011	
保留	100	
TMR4_TRGOUT	101	
EXINT_9	110	外部信号
DxSWTRG	111	软件触发

当 DxTRGEN 位被置 1 时，每次 DAC 侦测到有效的触发事件，存放在 DAC_DxDTH(data holding register) 中的数据就会被传动到寄存器 DAC_DxDOT 中，当选择软件触发时，触发标志 DxSWTRG 在被软件置 ‘1’ 后，硬件自动清零。一旦数据装入 DAC_DxDOT，经过一段时间，模拟的数模转换器输出即有效。当 DxTRGEN 位被清 ‘0’ 时，每次写入数据寄存器值时，数据即被传送到寄存器 DAC_DxDOT 中，无需等待触发事件。

19.4.2 数据配置

DAC 支持单 DAC 或者双 DAC 模式，根据模式的不同，数据配置有以下方式：

单 DAC 数据配置方式：对寄存器 DAC_DxDTH [5: 0] 位写入对应 DACx 值。

双 DAC 数据配置方式：对寄存器 DAC_DDDTH [5: 0] 位写入 DAC1 值及对寄存器 DAC_DDDTH [13: 8] 位写入 DAC2 值。

注意：双 DAC 数据被配置之后会自动分配到对应的 DAC_DxDTH。

19.5 DAC寄存器

下表列出了所有 DAC 寄存器。

必须以字（32 位）的方式操作这些外设寄存器。

表 19-2 DAC寄存器映像和复位值

寄存器简称	基址偏移量	复位值
DAC_CTRL	000h	0x0000 0000
DAC_SWTRG	004h	0x0000 0000
DAC_D1DTH	008h	0x0000 0000
DAC_D2DTH	00Ch	0x0000 0000
DAC_DDDTH	010h	0x0000 0000
DAC_D1ODT	014h	0x0000 0000
DAC_D2ODT	018h	0x0000 0000
DAC_STS	01Ch	0x0000 0000
DAC_OUTEN	020h	0x0000 0000

19.5.1 DAC控制寄存器 (DAC_CTRL)

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	D2DMAUDRIEN	0x0	rw	DAC2 的 DMA 传输下溢中断使能 (DAC2 DMA transfer underrun interrupt enable) 该位由软件设置和清除。 0: 关闭 DAC2 DMA 下溢中断; 1: 使能 DAC2 DMA 下溢中断。
位 28	D2DMAEN	0x0	rw	DAC2 的 DMA 传输使能 (DAC2 DMA transfer enable) 该位由软件设置和清除。 0: 关闭 DAC2 DMA 模式; 1: 使能 DAC2 DMA 模式。
位 27: 22	保留	0x0	resd	请保持默认值。
位 21: 19	D2TRGSEL	0x0	rw	DAC2 的触发事件选择 (DAC2 trigger select) 000: TMR6 TRGOUT 事件; 001: TMR3 TRGOUT 事件; 010: TMR7 TRGOUT 事件; 011: TMR9 TRGOUT 事件; 100: 保留 101: TMR4 TRGOUT 事件; 110: 外部中断线 9; 111: 软件触发。 注意: 这些位只能在 D2TRGEN = 1 时设置。
位 18	D2TRGEN	0x0	rw	DAC2 触发使能 (DAC2 trigger enable) 0: 关闭; 1: 开启。 注: 关闭触发时, 写入寄存器 DAC_D2DTH 或 DAC_DDDTH 的数据在 1 个 APB1 时钟周期后传入寄存器 DAC_D2ODT。 开启触发时, 写入寄存器 DAC_D2DTH 或 DAC_DDDTH 的数据在 3 个 APB1 时钟周期后传入寄存器 DAC_D2ODT。 如果选择软件触发, 写入寄存器 DAC_D2DTH 或 DAC_DDDTH 的资料只需要 1 个 APB1 时钟周期就可以传入寄存器 DAC_D2ODT。

位 17	DAC2_VREF_SEL	0x0	rw	DAC2 参考电压切换 (DAC2 reference voltage switch) 0: VDDA; 1: VREF。
位 16	D2EN	0x0	rw	DAC2 使能 (DAC2 enable) 0: 关闭; 1: 开启。
位 15: 14	保留	0x0	resd	请保持默认值。
位 13	D1DMAUDRIEN	0x0	rw	DAC1 的 DMA 传输下溢中断使能 (DAC1 DMA transfer underrun interrupt enable) 该位由软件设置和清除。 0: 关闭 DAC1 DMA 下溢中断; 1: 使能 DAC1 DMA 下溢中断。
位 12	D1DMAEN	0x0	rw	DAC1 的 DMA 传输使能 (DAC1 DMA transfer enable) 该位由软件设置和清除。 0: 关闭 DAC1 DMA 模式; 1: 使能 DAC1 DMA 模式。
位 11: 6	保留	0x0	resd	请保持默认值。
位 5: 3	D1TRGSEL	0x0	rw	DAC1 的触发事件选择 (DAC1 trigger select) 000: TMR6 TRGOUT 事件; 001: TMR3 TRGOUT 事件; 010: TMR7 TRGOUT 事件; 011: TMR9 TRGOUT 事件; 100: 保留 101: TMR4 TRGOUT 事件; 110: 外部中断线 9; 111: 软件触发。 注: 这些位只能在 D1TRGEN= 1 时设置。
位 2	D1TRGEN	0x0	rw	DAC1 触发使能 (DAC1 trigger enable) 0: 关闭; 1: 开启。 注: 关闭触发时, 写入寄存器 DAC_D1DTHx 的数据在 1 个 APB1 时钟周期后传入寄存器 DAC_D1ODT 开启触发时, 写入寄存器 DAC_D1DTHx 的数据在 3 个 APB1 时钟周期后传入寄存器 DAC_D1ODT。 如果选择软件触发, 写入寄存器 DAC_D1DTHx 的资料只需要 1 个 APB1 时钟周期就可以传入寄存器 DAC_D1ODT。
位 1	DAC1_VREF_SEL	0x0	rw	DAC1 参考电压切换 (DAC1 reference voltage switch) 0: VDDA; 1: VREF。
位 0	D1EN	0x0	rw	DAC1 使能 (DAC1 enable) 0: 关闭; 1: 开启。

19.5.2 DAC软件触发寄存器 (DAC_SWTRG)

域	简称	复位值	类型	功能
位 31: 2	保留	0x0000 0000	resd	请保持默认值。
位 1	D2SWTRG	0x0	rw	DAC2 软件触发 (DAC2 software trigger) 0: 不触发; 1: 触发。 注: 一旦寄存器 DAC_D2DTH 或 DAC_DDDTH 的数据传入寄存器 DAC_D2ODT, (1 个 APB1 时钟周期后) 该位由硬件清零。
位 0	D1SWTRG	0x0	rw	DAC1 软件触发 (DAC1 software trigger) 0: 不触发; 1: 触发。

注：一旦寄存器 DAC_D1DTH 或 DAC_DDDTH 的数据传入寄存器 DAC_D1ODT，（1 个 APB1 时钟周期后）该位由硬件清零。

19.5.3 DAC1的6位数据保持寄存器 (DAC_D1DTH)

域	简称	复位值	类型	功能
位 31: 12	保留	0x00000	resd	请保持默认值。
位 5: 0	D1DTH	0x00	rw	DAC1 的 6 位保持数据 (DAC1 6-bit data)

19.5.4 DAC2的6位数据保持寄存器 (DAC_D2DTH)

域	简称	复位值	类型	功能
位 31: 6	保留	0x0000	resd	请保持默认值。
位 5: 0	D2DTH	0x00	rw	DAC2 的 6 位保持数据 (DAC2 6-bit data)

19.5.5 双DAC的6位数据保持寄存器 (DAC_DDDTH)

域	简称	复位值	类型	功能
位 31: 14	保留	0x00000	resd	请保持默认值。
位 13: 8	D2DTH	0x00	rw	DAC2 的 6 位保持数据 (DAC2 6-bit data)
位 5: 0	D1DTH	0x00	rw	DAC1 的 6 位保持数据 (DAC1 6-bit data)

19.5.6 DAC1数据输出寄存器 (DAC_D1ODT)

域	简称	复位值	类型	功能
位 31: 6	保留	0x000000	resd	请保持默认值。
位 5: 0	D1ODT	0x00	rw	DAC1 的 6 位输出数据 (DAC1 output data)

19.5.7 DAC2数据输出寄存器 (DAC_D2ODT)

域	简称	复位值	类型	功能
位 31: 6	保留	0x000000	resd	请保持默认值。
位 5: 0	D2ODT	0x00	rw	DAC2 的 6 位输出数据 (DAC2 output data)

19.5.8 DAC状态寄存器 (DAC_STS)

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	D2DMAUDRF	0x0	w1c	DAC2 的 DMA 传输下溢标志 (DAC2 DMA transfer underrun flag) 0: DAC2 未发生 DMA 传输下溢。 1: DAC2 发生 DMA 传输下溢。 注：软件写‘1’清除该标志。
位 28: 14	保留	0x0000	resd	请保持默认值
位 13	D1DMAUDRF	0x0	w1c	DAC1 的 DMA 传输下溢标志 (DAC1 DMA transfer underrun flag) 0: DAC1 未发生 DMA 传输下溢。 1: DAC1 发生 DMA 传输下溢。 注：软件写‘1’清除该标志。
位 12: 0	保留	0x0000	resd	请保持默认值

19.5.9 DAC输出到GPIO使能寄存器 (DAC_OUTEN)

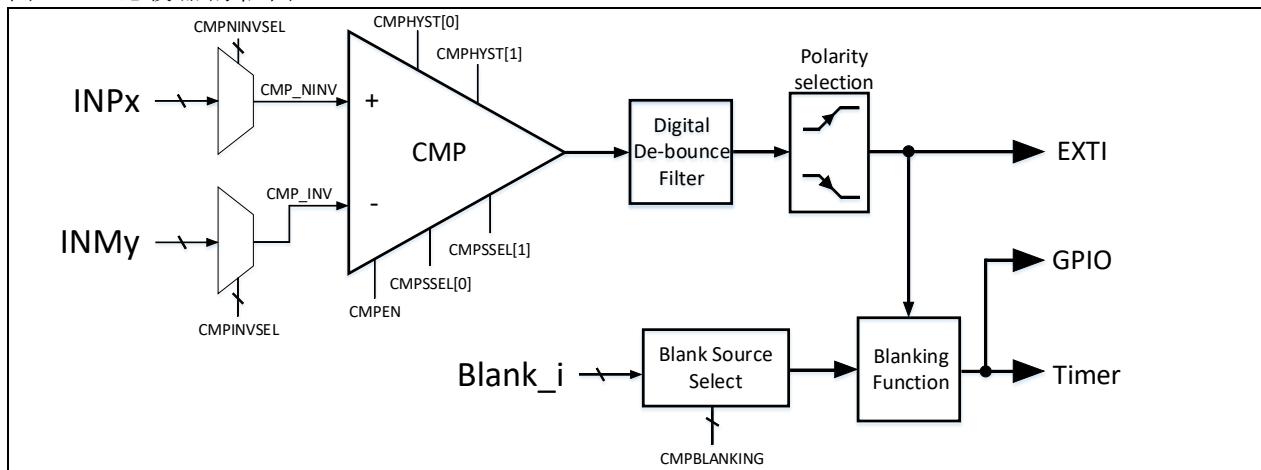
域	简称	复位值	类型	功能
位 31: 2	保留	0x0000000	resd	请保持默认值。
位 1	D2_IO	0x0	rw	DAC2 输出到 IO 使能 0: 不使能; 1: 使能。
位 0	D1_IO	0x0	rw	DAC1 输出到 IO 使能 0: 不使能; 1: 使能。

20 比较器 (CMP)

20.1 简介

超低功耗比较器 CMP1 和 CMP2，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 20-1 比较器的框图.



20.2 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
- 结合 EXINT 产生中断，从低功耗模式唤醒

20.3 中断管理

模拟比较器 1 和比较器 2 的输出经过各自的数字滤波器滤波以及极性选择，输入至 EXINT 第 21 和 22 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。

20.4 设计提示

比较器有以下提示仅供设计参考。

- 输入配置
- I/Os 作为比较器输入时，必须配置为模拟模式。
- 比较器 1 输出可以重映射到外部 I/O 口

表 20-1 比较器 1 输出映射到外部 I/O 口及外部 I/O 口复用配置

复用配置	bit	配置值	输出端口
GPIOA_MUXL	[3: 0]	1100	PA0
GPIOA_MUXL	[27: 24]	1100	PA6
GPIOA_MUXH	[15: 12]	1100	PA11
GPIOB_MUXL	[3: 0]	1100	PB0

GPIOB_MUXH	[3: 0]	1100	PB8
GPIOB_MUXH	[11: 8]	1100	PB10

- 比较器 1 输出可以重映射到外部 I/O 口

表 20-2 比较器2输出映射到外部I/O口及外部I/O口复用配置

复用配置	bit	配置值	输出端口
GPIOA_MUXL	[11: 8]	1100	PA2
GPIOA_MUXL	[31: 28]	1100	PA7
GPIOA_MUXH	[19: 16]	1100	PA12
GPIOB_MUXL	[27: 24]	1100	PB5
GPIOB_MUXH	[7: 4]	1100	PB9
GPIOB_MUXH	[15: 12]	1100	PB11

- 锁定功能

比较器控制和状态寄存器 CMP1_CTRLSTS 和 CMP2_CTRLSTS 具有写保护功能，一旦编程完成，对各 CMP1_CTRLSTS 或 CMP2_CTRLSTS 的 CMPWP 位设置为 1，则整个 CMP1_CTRLSTS 或 CMP2_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。

- 控制系统从低功耗模式中唤醒提示

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）。

20.5 功能描述

20.5.1 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 DAC 输出电压和或者 I/O 口或者 VSSA。

表 20-3 比较器1和比较器2输入选择

输入选择	CMP1	CMP2
INP0	PA5	PA7
INP1	PA1	PA3
INP2	PA0	PA2
INP3	PB1	保留
INM0	PA1	保留
INM1	PB1	保留
INM2	DAC1_OUT	DAC1_OUT
INM3	DAC2_OUT	DAC2_OUT
INM4	PA4	PA4
INM5	PA5	PA5
INM6	PA0	PA2
INM7	PA2	PA6

- 迟滞功能

寄存器 CMP1_CTRLSTS 和 CMP2_CTRLSTS 的 CMPHYST[1: 0]控制比较器 1 和比较器 2 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 CMP1_CTRLSTS 和 CMP2_CTRLSTS 的 CMPSSEL[1: 0]位用于控制比较器 1 和比较器 2 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 1 和比较器 2 在性能与功耗之间的折中。

● 输出消隐功能

寄存器 CMP1_CTRLSTS 和 CMP2_CTRLSTS 的 CMPBLANKING[6: 0]位用于选择比较器 1 和比较器 2 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。

20.5.2 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 值下的时序图。

图 20-2 H_PULSE_CNT=1 和 L_PULSE_CNT =0 时干扰滤波器时序图

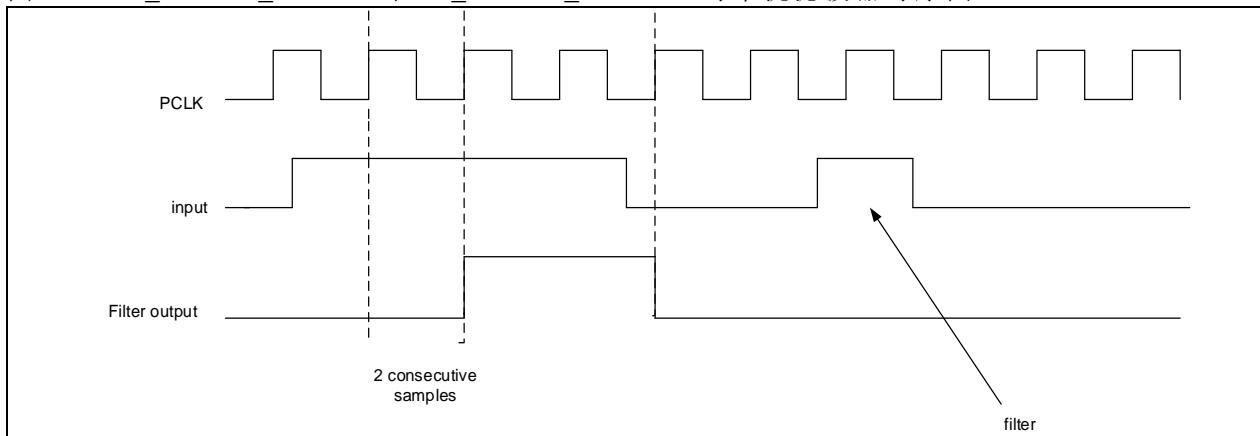
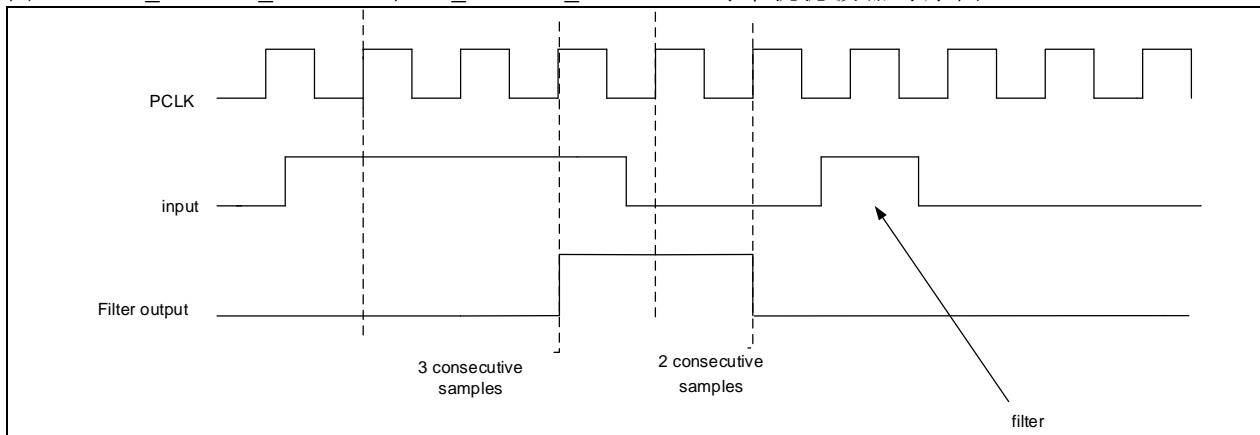


图 20-3 H_PULSE_CNT=2 和 L_PULSE_CNT = 1 时干扰滤波器时序图



20.6 CMP 寄存器

必须以字（32 位）的方式操作这些外设寄存器。

表 20-4 CMP1 和 CMP2 寄存器映像和复位值

寄存器简称	基址偏移量	复位值
CMP1_CTRLSTS	0x30	0x0000 0120
CMP1_G_FILTER_EN	0x34	0x0000 0000
CMP1_HIGH_PULSE	0x38	0x0000 0000
CMP1_LOW_PULSE	0x3C	0x0000 0000
CMP2_CTRLSTS	0x40	0x0000 0120
CMP2_G_FILTER_EN	0x44	0x0000 0000
CMP2_HIGH_PULSE	0x48	0x0000 0000
CMP2_LOW_PULSE	0x4C	0x0000 0000

20.6.1 比较器1控制和状态寄存器(CMP1_CTRLSTS)

域	简称	复位值	类型	功能
位 31	CMPWWP	0x0	rw0c	比较器写保护(Comparator write protect) 该位仅写一次,由软件置1,由系统复位清零。它将锁定 比较器1控制寄存器CMP1_CTRLSTS[31:0]的全部内容 0: 比较器的CMP1_CTRLSTS[31:0]可读/可写 1: 比较器的CMP1_CTRLSTS[31:0]仅读
位 30	CMPVALUE	0x0	ro	比较器1输出值(Comparator 1 output value) 此位为仅读。
位 29:26	保留	0x00	resd	请保持默认值。
位 25:19	CMPBLANKING	0x00	rw	比较器消隐窗口来源(Comparator blank source) 0000000: 没有输出消隐功能 xxxxxx1: TMR1_OC4 xxxxxx1x: TMR1_OC5 xxxx1xx: 保留 xxx1xxx: TMR3_OC3 xx1xxxx: TMR9_OC2 x1xxxxx: TMR9_OC1 1xxxxxx: TMR4_OC3
位 18	保留	0x0	resd	请保持默认值。
位 17:16	CMPHYST	0x0	rw	比较器迟滞(Comparator hysteresis) 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 14:10	保留	0x0	resd	请保持默认值。
位 9:8	CMPINVSEL	0x1	rw	比较器同相输入选择(Comparator non-inverting input selection) 00: INP0 01: INP1 10: INP2 11: INP3
位 7	保留	0x0	resd	请保持默认值。
位 6:4	CMPINVSEL	0x0	rw	比较器的反相端输入选择(Comparator inverting selection) 000: INM0 001: INM1 010: INM2 011: INM3 100: INM4 101: INM5 110: INM6 111: INM7
位 3:2	CMPSSSEL	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	保留	0x0	resd	请保持默认值。
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

20.6.2 比较器1干扰滤波器使能寄存器(CMP1_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

20.6.3 比较器1干扰滤波器高脉冲数(CMP1_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。
位 5:0	H_PULSE_CNT	0x00	rw	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。 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

20.6.4 比较器1干扰滤波器低脉冲数(CMP1_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。
位 5:0	L_PULSE_CNT	0x00	rw	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。 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

20.6.5 比较器2控制和状态寄存器(CMP2_CTRLSTS)

域	简称	复位值	类型	功能
位 31	CMPWP	0x0	rw0c	比较器 2 写保护(Comparator write protect) 该位仅写一次，由软件置 1，由系统复位清零。它将锁定比较器 2 控制寄存器 CMP2_CTRLSTS[31:0]的全部内容 0: 比较器的 CMP2_CTRLSTS[31:0]可读/可写 1: 比较器的 CMP2_CTRLSTS[31:0] 仅读
位 30	CMPVALUE	0x0	ro	比较器 2 输出值(Comparator 2 output value) 此位为只读。
位 29:26	保留	0x00	resd	请保持默认值。
位 25:19	CMPBLANKING	0x00	rw	比较器消隐窗口来源(Comparator blank source) 0000000: 没有输出消隐功能 xxxxxx1: TMR1_OC4 xxxxxx1x: TMR1_OC5 xxxx1xx: 保留 xxx1xxx: TMR3_OC3 xx1xxxx: TMR9_OC2 x1xxxx: TMR9_OC1 1xxxxxx: TMR4_OC3
位 18	保留	0x0	resd	请保持默认值。

				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 14:10	保留	0x0	resd	请保持默认值。
				比较器同相输入选择(Comparator non-inverting input selection)
位 9:8	CMPNINVSEL	0x1	rw	00: INP0 01: INP1 10: INP2 11: INP3
				比较器的反相端输入选择(Comparator inverting selection)
位 6:4	CMPINVSEL	0x0	rw	000: INM0 001: INM1 010: INM2 011: INM3 100: INM4 101: INM5 110: INM6 111: INM7
				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
位 3:2	CMPSSEL	0x0	rw	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	保留	0x0	resd	请保持默认值。
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

20.6.6 比较器2干扰滤波器使能寄存器(CMP2_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

20.6.7 比较器2干扰滤波器高脉冲数(CMP2_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。
位 5:0	H_PULSE_CNT	0x00	rw	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。 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

20.6.8 比较器2干扰滤波器低脉冲数(CMP2_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

21 运算放大器 (OP)

21.1 简介

AT32M412 / M416 系列多达 4 个运算放大器单元。

单一的运放单元有可选的 INP/INM 通道，INP 有 4 个 channel，INM 有 2 个 channel，有可选的输出到 GPIO / 内部 ADC 通道。运算放大器也可以配置为独立模式，跟随模式，可配置增益模式。
INP channel 输入可选 GPIO 或者 DAC。

21.2 运算放大器主要功能

- 轨到轨的输入输出范围
- 可配置的快速模式
- 用户可自定义校准值
- 可通过 TMR1 的输出切换 INP/INM channel
- 多种功能模式：独立模式，跟随模式，可配置增益模式

21.3 运算放大器功能描述

运算放大器有多种模式，每个运算放大器都有各自独立使能信号，每个运算放大器功能都可以独立配置。

21.3.1 运算放大器时钟和复位

运算放大器的时钟使能信号复用 SCFG 的时钟使能信号。

运算放大器的软件复位使能信号也是复用 SCFG 的软件复位使能信号。

当配置好运算放大器之后可以关闭时钟，运算放大器还是可以正常工作。

21.3.2 运算放大器模式

每个运算放大器都可以独立的配置成不同的模式：

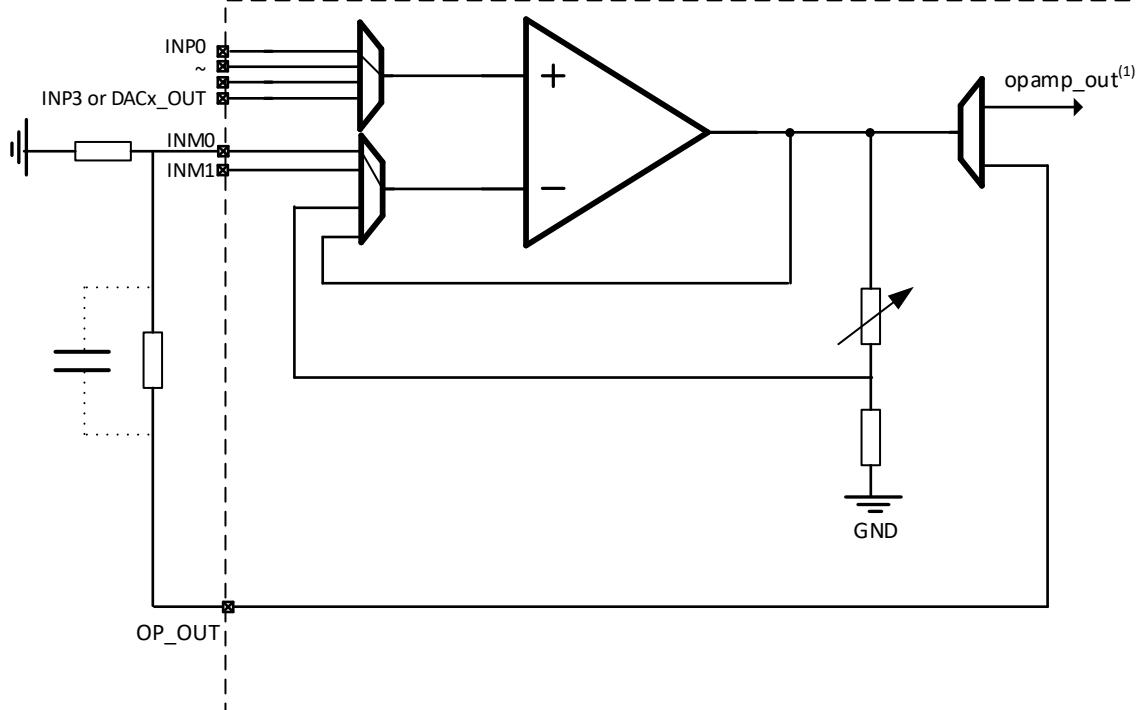
- 独立模式
- 跟随模式
- 可配置增益模式

21.3.3 独立模式

用户可通过如下配置进入独立模式：

- 配置 OPx_CTRL.INMSEL[1:0] =0x
- 配置对应的 INP/INM 通道为模拟模式
- 如配置了 OPx_CTRL.OPOUTSEL=1 则 OP_OUT 对应的 GPIO 可以释放，运算放大器的输出到对应的 ADC 通道上。
- 如配置了 OPx_CTRL.OPFSTMD=1 运算放大器进入快速模式。
- 当 OPx_CTRL.OPEN 配置为 1 之后就进入了独立模式。

图 21-1 独立模式电路



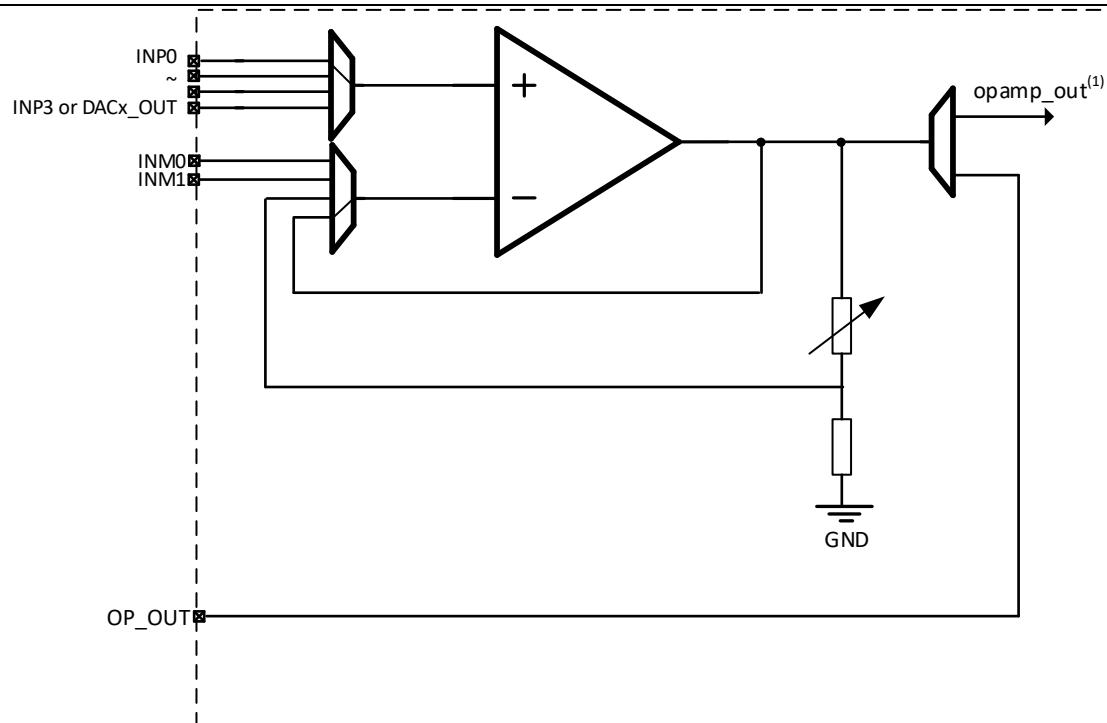
1. 通过设置 OPOUTSEL 位, *opamp_out* 可以在内部重定向到 ADC 通道。在这种情况下, *OP_OUT* 对应的 GPIO 可以释放, 用于其他目的。

21.3.4 跟随模式

用户可通过如下配置进入跟随模式：

- 配置 OPx_CTRL.INMSEL[1:0] = 11
- 配置 OPx_CTRL.INPSEL[1:0] 对应的 GPIO 通道为模拟模式，INM 对应的 GPIO 释放出来。
- 如配置了 OPx_CTRL.OPOUTSEL=1 则 OP_OUT 对应的 GPIO 可以释放，运算放大器的输出到对应的 ADC 通道上。
- 如配置了 OPx_CTRL.OPFSTMD=1 运算放大器进入快速模式。
- OPx_CTRL.OPEN 配置为 1 之后就进入了跟随模式。

图 21-2 跟随模式电路



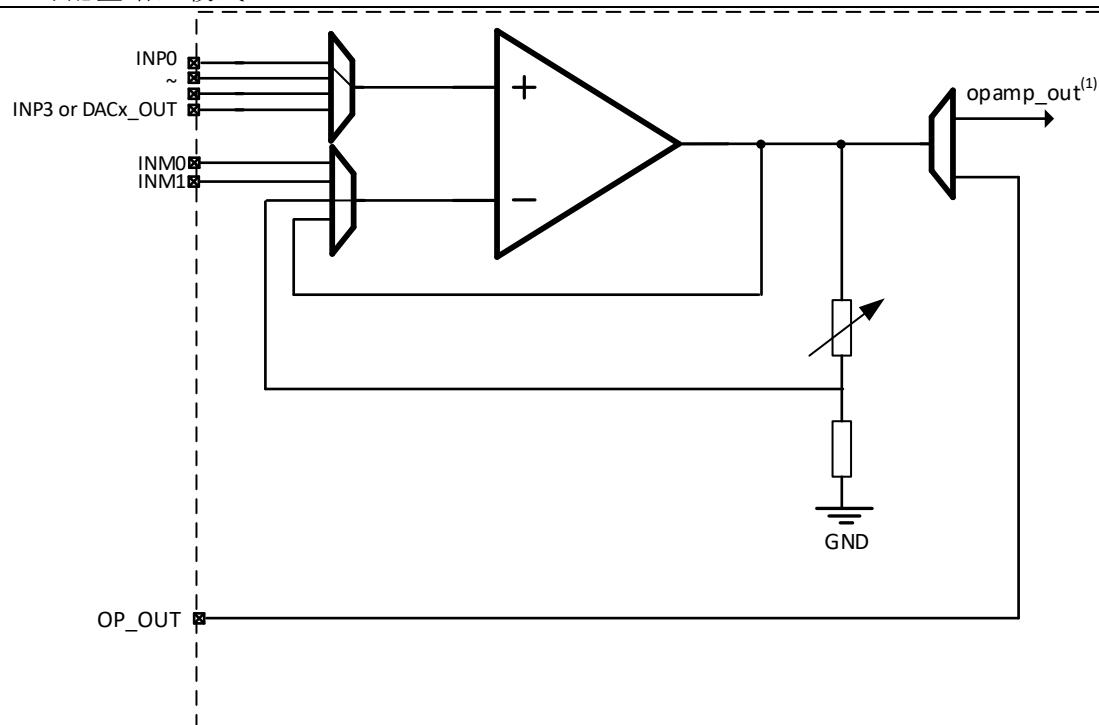
1. 通过设置 OPOUTSEL 位, *opamp_out* 可以在内部重定向到 ADC 通道。在这种情况下, *OP_OUT* 对应的 GPIO 可以释放, 用于其他目的。

21.3.5 可配置增益模式（正向增益模式）

用户可通过如下配置进入可配置增益模式：

- 配置 OPx_CTRL.INMSEL[1:0] = 10
- 可通过配置 OPx_CTRL.OPGAIN [4:0] 为 00000 到 00101 来配置内部增益（内部增益对应为 2,4,8,16,32,64）。
- 配置 OPx_CTRL.INPSEL[1:0] 对应的 GPIO 通道为模拟模式，INM 对应的 GPIO 释放出来。
- 如配置了 OPx_CTRL.OPOUTSEL=1 则 OP_OUT 对应的 GPIO 可以释放，运算放大器的输出到对应的 ADC 通道上。

图 21-3 可配置增益模式（x2/x4/x8/x16/x32/x64）



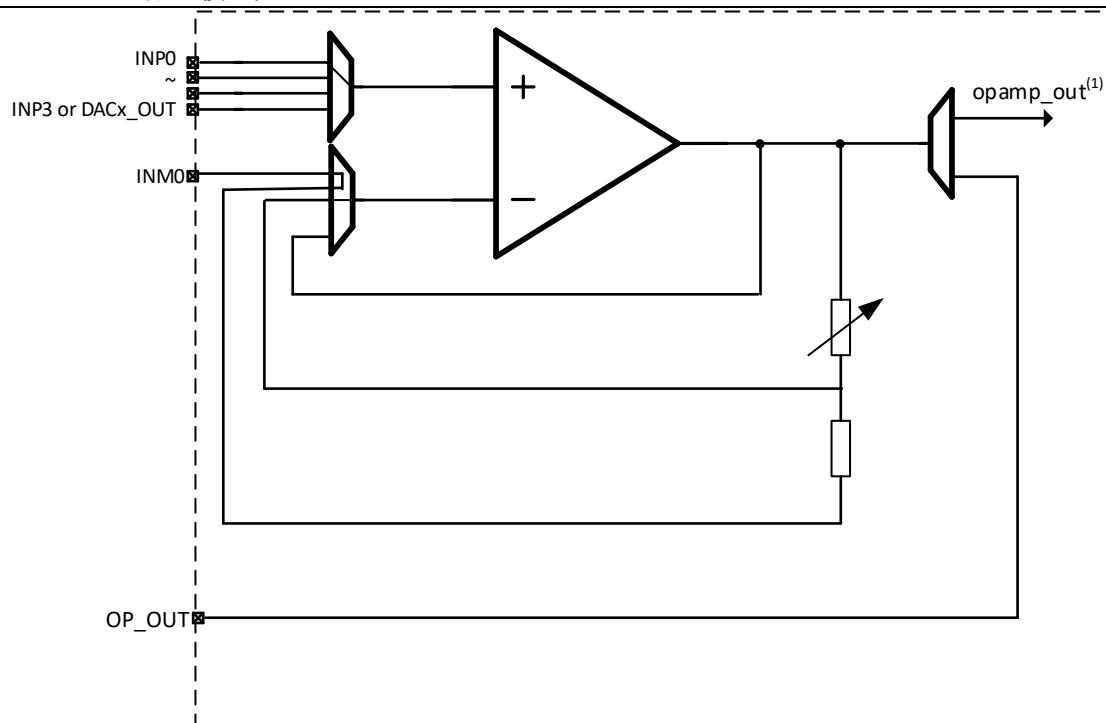
1. 通过设置 OPOUTSEL 位, opamp_out 可以在内部重定向到 ADC 通道。在这种情况下, OP_OUT 对应的 GPIO 可以释放, 用于其他目的。

21.3.6 可配置增益模式（带外部偏置正向增益或反向增益模式）

用户可通过如下配置进入带外部偏置正向增益或反向增益模式：

- 配置 OPx_CTRL.INMSEL [1:0] = 10
- 可通过配置 OPx_CTRL.OPGAIN [4:0] 为 01000 到 01101 来实现带外部偏置的内部正向增益配置和反向增益配置（内部正向增益对应为 2,4,8,16,32,64，内部反向增益对应为 -1,-3,-7,-15,-31,-63）。
- 配置 OPx_CTRL.INPSEL [1:0] 对应的 GPIO 通道为模拟模式。
- 如配置了 OPx_CTRL.OPOUTSEL=1 则 OP_OUT 对应的 GPIO 可以释放，运算放大器的输出到对应的 ADC 通道上。

图 21-4 可配置增益模式 ($x2/x4/x8/x16/x32/x64$ 或 $x-1/x-3/x-7/x-15/x-31/x-63$)



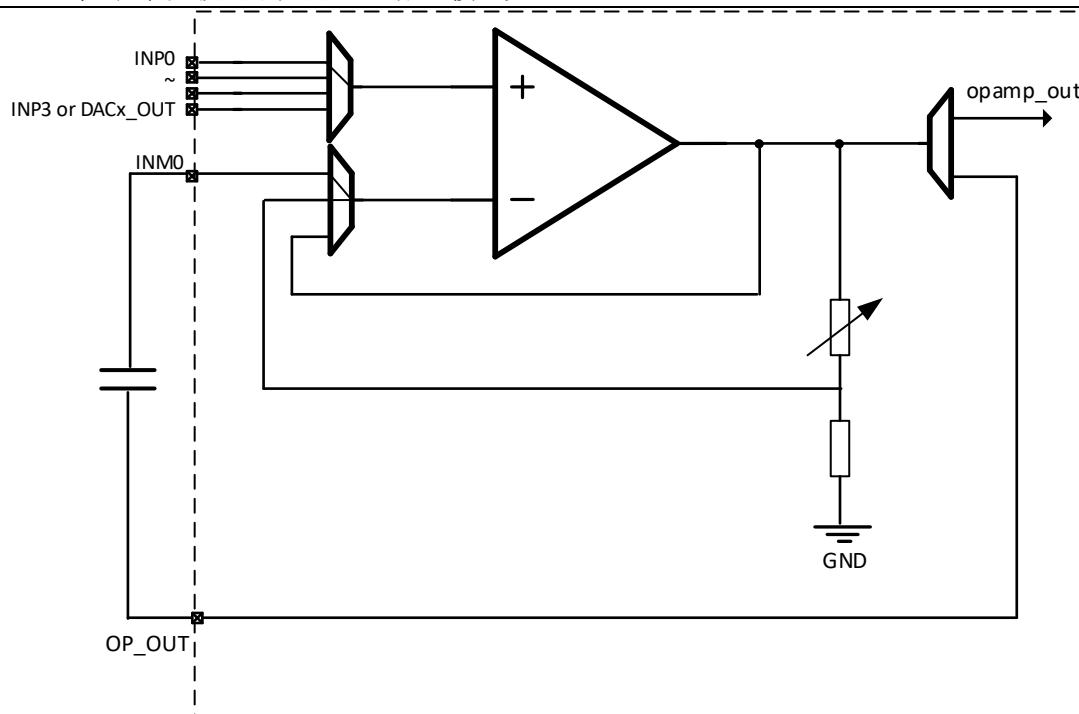
1. 通过设置 OPOUTSEL 位, *opamp_out* 可以在内部重定向到 ADC 通道。在这种情况下, *OP_OUT* 对应的 GPIO 可以释放, 用于其他目的。

21.3.7 可配置增益模式（带外部滤波器的正向增益模式）

用户可通过如下配置进入带外部滤波器的可配置正向增益模式：

- 配置 `OPx_CTRL.INMSEL[1:0] = 10`
- 可通过配置 `OPx_CTRL.OPGAIN [4:0]` 为 10000 到 10101 来实现带外部滤波器的正向增益配置（内部增益对应为 2,4,8,16,32,64）。
- 配置 `OPx_CTRL.INPSEL[1:0]` 对应的 GPIO 通道为模拟模式，`INM0` 对应的 GPIO 设置为模拟模式。
- 配置了 `OPx_CTRL.OPOUTSEL = 0`，`OP_OUT` 对应的 GPIO 配置为模拟模式。
- 用户可在芯片外 `OP_OUT` 和 `INM0` 之间外接电容作为滤波器。

图 21-5 带外部滤波器的可配置增益模式（x2/x4/x8/x16/x32/x64）

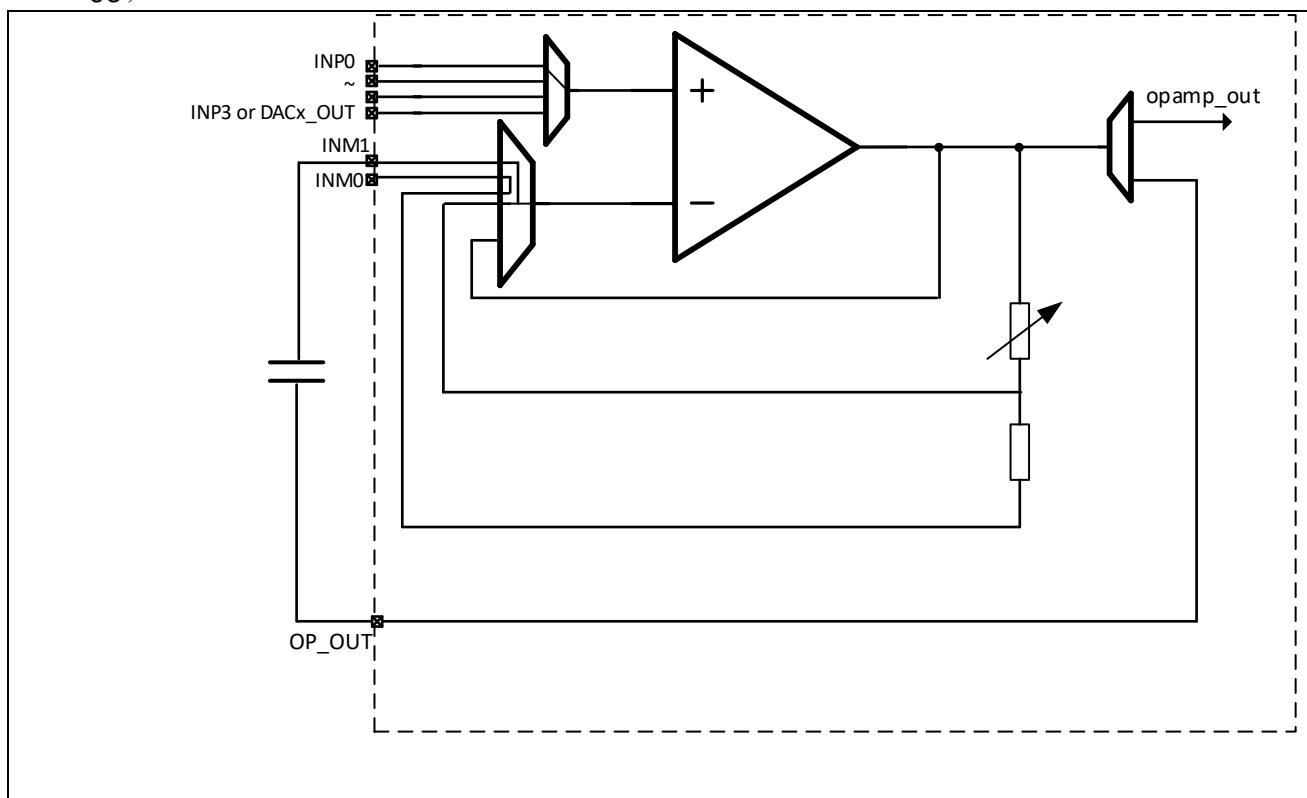


21.3.8 可配置增益模式（带外部滤波器的带外部偏置正向增益或反向增益模式）

用户可通过如下配置进入使用带外部滤波器的带外部偏置的正增益或者反向增益模式：

- 配置 OPx_CTRL.INMSEL[1:0] = 10
- 可通过配置 OPx_CTRL.OPGAIN [4:0] 为 11000 到 11101 来实现带外部偏置的内部正向增益配置和反向增益配置（内部正向增益对应为 2,4,8,16,32,64，内部反向增益对应为 -1,-3,-7,-15,-31,-63）。
- 配置 OPx_CTRL.INPSEL[1:0] 对应的 GPIO 通道为模拟模式，INM0/INM1 对应的 GPIO 设置为模拟模式。
- 配置 OPx_CTRL.OPOUTSEL = 0，OP_OUT 对应的 GPIO 配置为模拟模式。
- 用户可在芯片外 OP_OUT 和 INM1 之间外接电容作为滤波器。

图 21-6 带外部滤波器可配置增益模式 ($x2/x4/x8/x16/x32/x64$ 或 $x-1/x-3/x-7/x-15/x-31/x-63$)



21.3.9 运算放大器通道连接

每个运算放大器有各自独立的 INP/INM 以及输出通道，INP 通道通过 OPx_CTRL.INPSEL[1:0]进行选择 INM 通道通过 OPx_CTRL.INMSEL[1:0]进行选择，通过 OPx_CTRL.OPOUTSEL 选择输出到 GPIO 或者是内部 ADC。

所有输入输出到 GPIO 的通道都必须保证对应的 GPIO 设置为了模拟模式。

表 21-1 INP/INM/GPIO 通道

	OP1	OP2	OP3	OP4
INP0	PA1	PA7	PB0	PB13
INP1	PA3	PB14	PB13	DAC1_OUT (内部连接)
INP2	PA7	PB0	PA1	PB11
INP3	DAC1_OUT (内部连接)	DAC2_OUT (内部连接)	DAC1_OUT (内部连接)	DAC2_OUT (内部连接)
INM0	PA3	PA5	PB2	PB10
INM1	NA	NA	PB10	NA
OP_OUT	PA2	PA6	PB1	PB12

注意：当配置了对应的通道之后需确保这些通道对应的 GPIO 处于模拟模式。

表 21-2 内部连接通道

	OP1	OP2	OP3	OP4
ADC1	IN19	IN20	IN21	IN22
ADC2	IN19	IN20	NA	NA

注意：内部 ADC 和外部 GPIO 不会同时输出。

INP 的通道是通过 OPx_CTRL.INPSEL 信号进行选择的，如果配置了 timer 自动切换功能 OPx_TC.INPSSEL 对应的通道也需要配置为模拟模式。

表 21-3 INP 通道使能控制

OPEN	VRFSEL	CALEN	INPSEL[1]或者 INPSSEL[1]	INPSEL[0] 或者 INPSSEL[0]	INP 使能
0	X	X	X	X	运算放大器关闭
1	0	0	0	0	通道 INP0 使能
1	0	0	0	1	通道 INP1 使能
1	0	0	1	0	通道 INP2 使能
1	0	0	1	1	通道 INP4 使能
1	0	1	X	X	通道 INP0/INP1/INP2/INP3 不使能
1	1	0	X	X	通道 INP0/INP1/INP2/INP3 不使能
1	1	1	X	X	通道 INP0/INP1/INP2/INP3 不使能

INM 的通道是通过 OPx_CTRL.INMSEL 信号进行选择的，如果配置了 timer 自动切换功能 {OPx_CTRL.INMSEL[1],OPx_TC.INMSSEL} 对应的通道也需要配置为模拟模式。

表 21-4 独立模式/跟随模式通道控制

OPEN	CALEN	INMSEL1	INMSEL0 或者 INMSSEL1	INM 使能
0	X	X	X	运算放大器关闭
1	1	X	X	通道 INM0/INM1 不使能
1	0	0	0	通道 INM0 使能
1	0	0	1	通道 INM1 使能
1	0	1	1	通道 INM0/INM1 不使能

注意：独立模式 $INMSEL=00$, 跟随模式 $INMSEL=01$,

表 21-5 可配置增益模式通道控制

OPEN	CALEN	OPGAIN[3]	OPGAIN[4]	INM 使能
1	0	0	0	通道 INM0/INM1 不使能
1	0	0	1	通道 INM0 使能 通道 INM1 不使能
1	0	1	0	通道 INM0 使能 通道 INM1 不使能
1	0	1	1	通道 INM0/INM1 使能

注意：可配置增益模式 $INMSEL=10$

21.4 校准

运算放大器校准功能可以将其偏移误差降到最低。芯片出厂会有一个默认的校准值，用户也可以通过配置相关寄存器进行校准。

通过配置 $OPx_CTRL.USERTRIM$ 值为 1 选择用户设定校准值，分别设置 $OPx_CTRL.TRIMVALN / OPx_CTRL.TRIMVALP$ 的值进行校准。

- 当 $OPx_CTRL.USERTRIM$ 值为 0，则使用的是芯片出厂时设置的校准值。

当开始进行校准时， $OPx_CTRL.OPCAL$ 会置 1，校准成功之后 $OPx_CTRL.OPCAL$ 会置 0。

用户校准流程：

- 配置 $OPx_CTRL.OPEN = 1$
- 配置 $OPx_CTRL.USERTRIM = 1$
- 配置 $OPx_CTRL.CALSEL = 01$ ，用于 PMOS 校准：
 - 如果配置的是 OP 输出到 GPIO ($OPx_CTRL.OPOUTSEL=0$)，把 $OPx_CTRL.TRIMVALP$ 的值调到'b00000, 再慢慢往上增加，这个时候去读 $OPx_CTRL.OPCAL$ 的值，当从 1 变为 0 时，所对应的校准值，就是我们所需要的。
 - 如果配置的是 OP 输出到 ADC ($OPx_CTRL.OPOUTSEL=1$)，把 $OPx_CTRL.TRIMVALP$ 的值调到'b00000, 再慢慢往上增加，通过读 ADC 转换出来的值从接近最大 ADC code 的值变化到接近最小 ADC code 的值时，所对应的校准值，就是我们所需要的。
- 配置 $OPx_CTRL.CALSEL = 11$ ，用于 NMOS 校准：
 - 如果配置的是 OP 输出到 GPIO ($OPx_CTRL.OPOUTSEL=0$)，把 $OPx_CTRL.TRIMVALN$ 的值调到'b00000, 再慢慢往上增加，这个时候去读 $OPx_CTRL.OPCAL$ 的值，当从 1 变为 0 时，所对应的校准值，就是我们所需要的。
 - 如果配置的是 OP 输出到 ADC ($OPx_CTRL.OPOUTSEL=1$)，把 $OPx_CTRL.TRIMVALP$ 的值调到'b00000, 再慢慢往上增加，通过读 ADC 转换出来的值从接近最大 ADC code 的值变化到接近最小 ADC code 的值时，所对应的校准值，就是我们所需要的。

注意：校准时，每次更新 $TRIMVALP$ 和 $TRIMVALN$ 的值后，至少等待 1ms 让 OP 内部电路稳定后再读取 $OPCAL$ 的值。

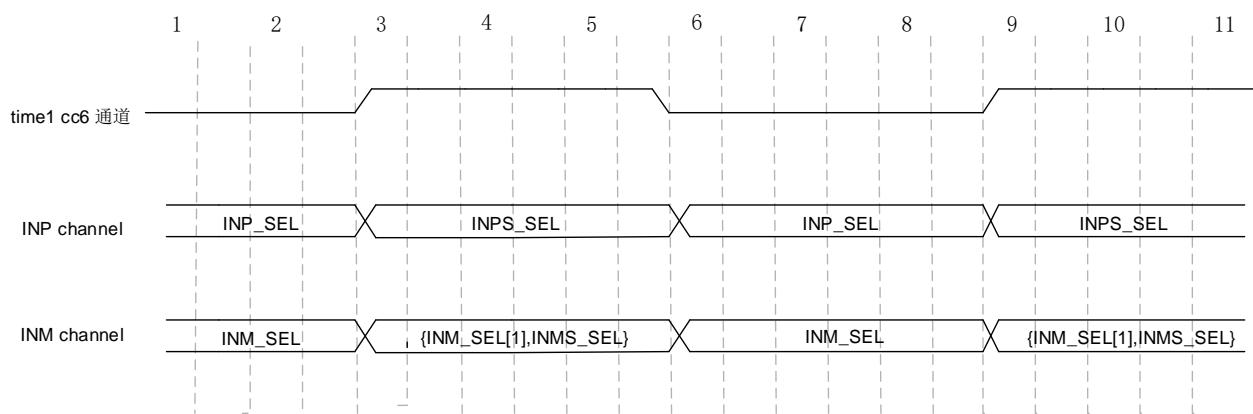
21.5 Timer自动切换功能

OP 模块中，可通过配置 OPx_TC 寄存器自动切换 INM/INP channel，以及自动切换模式。自动切换的触发是通过 timer1 的 CC6 通道。

用户配置流程：

- 配置 OPx_TC.TMR1EN = 1, 配置对应的 OPx_TC.INPSSEL/OPx_TC.INMSSEL。
- 配置 OPx_CTRL 所有对应的寄存器。
- 配置 TIMER1 对应的功能。
- 配置 OPx_CTRL.OPEN =1, OP 会通过 timer1 CC6 通道输入的高低电平切换配置的 OPx_CTRL.INPSEL/ OPx_CTRL.INMSEL 和 OPx_TC.INPSSEL/ OPx_TC.INMSSEL。

图 21-7 timer 切换功能



21.6 OP 控制寄存器

表 21-6 OP 寄存器地址映像和复位值

寄存器简称	基址偏移量	复位值
OP1_CTRL	0x00	0x0000 0000
OP2_CTRL	0x04	0x0000 0000
OP3_CTRL	0x08	0x0000 0000
OP4_CTRL	0x0C	0x0000 0000
OP1_TC	0x18	0x0000 0000
OP2_TC	0x1C	0x0000 0000
OP3_TC	0x20	0x0000 0000
OP4_TC	0x24	0x0000 0000

21.6.1 OP1 控制以及状态寄存器 (OP1_CTRL)

域	简称	复位值	类型	功能
31	WPC	0x0	rw	OP 锁存控制 (Operational amplifier write protect control) 写 1 之后不能通过写 0 的方式清除，只能通过 reset 清 0 0: OP1_CTRL 寄存器可以读写 1: OP1_CTRL 寄存器只读
30	OPCAL	0x0	r	校准成功信号 (Operational amplifier Calibration successful signal) 0: 校准中 1: 校准成功。
29	保留	0x0	resd	请保持为复位值。

28:24	TRIMVALN	0x0	rw	NMOS 的校准值 (Operational amplifier Calibration values of NMOS)
23:19	TRIMVALP	0x0	rw	PMOS 的校准值 (Operational amplifier Calibration values of PMOS)
18:14	OPGAIN	0x0	rw	<p>OPGAIN : 可配置的增益值 (Operational amplifier Configurable gain value) :</p> <p>00000: 正向增益为 2 00001: 正向增益为 4 00010: 正向增益为 8 00011: 正向增益为 16 00100: 正向增益为 32 00101: 正向增益为 64 00110: 保留 00111: 保留 01000: INMO 带外部偏置的正向增益为 2 或者反向增益为-1 01001: INMO 带外部偏置的正向增益为 4 或者反向增益为-3 01010: INMO 带外部偏置的正向增益为 8 或者反向增益为-7 01011: INMO 带外部偏置的正向增益为 16 或者反向增益为-15 01100: INMO 带外部偏置的正向增益为 32 或者反向增益为-31 01101: INMO 带外部偏置的正向增益为 64 或者反向增益为-63 01110: 保留 01111: 保留 10000: 带外部滤波器的正向增益为 2 10001: 带外部滤波器的正向增益为 4 10010: 带外部滤波器的正向增益为 8 10011: 带外部滤波器的正向增益为 16 10100: 带外部滤波器的正向增益为 32 10101: 带外部滤波器的正向增益为 64 10110: 保留 10111: 保留 11000: INMO 带外部偏置,INM1 带外部滤波的正向增益为 2 或者反向增益为-1 11001: INMO 带外部偏置,INM1 带外部滤波的正向增益为 4 或者反向增益为-3 11010: INMO 带外部偏置,INM1 带外部滤波的正向增益为 8 或者反向增益为-7 11011: INMO 带外部偏置,INM1 带外部滤波的正向增益为 16 或者反向增益为-15 11100: INMO 带外部偏置,INM1 带外部滤波的正向增益为 32 或者反向增益为-31 11101: INMO 带外部偏置,INM1 带外部滤波的正向增益为 64 或者反向增益为-63 11110: 保留 11111: 保留</p>
13:12	CALSEL	0x0	rw	<p>校准选择 (Operational amplifier Calibration selection) :</p> <p>用于校准, 当 CALEN 为 1 或者 VRFSEL 为 1 时有效</p> <p>00: OP 的输入为 $0.033 \times VDDA$ 01: OP 的输入为 $0.1 \times VDDA$, 用于 PMOS 的校准 10: OP 的输入为 $0.5 \times VDDA$ 11: OP 的输入为 $0.9 \times VDDA$, 用于 NMOS 的校准</p>
11	CALEN	0x0	rw	校准使能 (Operational amplifier Calibration enable) :
10:9	保留	0x0	resd	请保持为复位值。

				OP 输出选择 (Operational amplifier Output selection) : 0: OP 输出到 GPIO 上 1: OP 输出到 ADC 的通道上
8	OPOUTSEL	0x0	resd	OP 快速模式 (Operational amplifier fast mode) : 0: OP 普通模式 1: OP 快速模式
7	OPFSTMD	0x0	rw	INM 通道选择 (Operational amplifier Inverting input selection) : 00: 选择 INM0 通道 01: 选择 INM1 通道 10: 选择可配置增益模式 11: 选择跟随模式
6:5	INMSEL	0x0	rw	用户校准模式 (Operational amplifier User trimming enable) : 0: 选择出厂设置校准值 1: 选择用户设定的校准值
4	USERTRIM	0x0	rw	INP 通道选择 (Operational amplifier Non inverting input selection) : 00: 选择 INP0 通道 01: 选择 INP1 通道 10: 选择 INP2 通道 11: 选择 INP3 通道
3:2	INPSEL	0x0	rw	内部参考信号选择 (Operational amplifier Internal reference signal selection) : 0: 正常模式 1: 校准模式, 正增益作为校准电压
1	VRFSEL	0x0	rw	运算放大器使能 (Operational amplifier enable) : 0: 运算放大器不使能 1: 运算放大器使能
0	OPEN	0x0	rw	

21.6.2 OP2 控制以及状态寄存器 (OP2_CTRL)

域	简称	复位值	类型	功能
31	WPC	0x0	rw	OP 锁存控制 (Operational amplifier write protect control) 写 1 之后不能通过写 0 的方式清除, 只能通过 reset 清 0 0: OP2_CTRL 寄存器可以读写 1: OP2_CTRL 寄存器只读
30	OPCAL	0x0	r	校准成功信号 (Operational amplifier Calibration successful signal) 0: 校准中 1: 校准成功。
29	保留	0x0	resd	请保持为复位值。
28:24	TRIMVALN	0x0	rw	NMOS 的校准值 (Operational amplifier Calibration values of NMOS)
23:19	TRIMVALP	0x0	rw	PMOS 的校准值 (Operational amplifier Calibration values of PMOS)
18:14	OPGAIN	0x0	rw	OPGAIN : 可配置的增益值 (Operational amplifier Configurable gain value) : 00000: 正向增益为 2 00001: 正向增益为 4 00010: 正向增益为 8 00011: 正向增益为 16 00100: 正向增益为 32 00101: 正向增益为 64 00110: 保留 00111: 保留 01000: INM0 带外部偏置的正向增益为 2 或者反向增益为 -1

				01001: INM0 带外部偏置的正向增益为 4 或者反向增益为-3 01010: INM0 带外部偏置的正向增益为 8 或者反向增益为-7 01011: INM0 带外部偏置的正向增益为 16 或者反向增益为-15 01100: INM0 带外部偏置的正向增益为 32 或者反向增益为-31 01101: INM0 带外部偏置的正向增益为 64 或者反向增益为-63 01110: 保留 01111: 保留 10000: 带外部滤波器的正向增益为 2 10001: 带外部滤波器的正向增益为 4 10010: 带外部滤波器的正向增益为 8 10011: 带外部滤波器的正向增益为 16 10100: 带外部滤波器的正向增益为 32 10101: 带外部滤波器的正向增益为 64 10110: 保留 10111: 保留 11000: INM0 带外部偏置,INM1 带外部滤波的正向增益为 2 或者反向增益为-1 11001: INM0 带外部偏置,INM1 带外部滤波的正向增益为 4 或者反向增益为-3 11010: INM0 带外部偏置,INM1 带外部滤波的正向增益为 8 或者反向增益为-7 11011: INM0 带外部偏置,INM1 带外部滤波的正向增益为 16 或者反向增益为-15 11100: INM0 带外部偏置,INM1 带外部滤波的正向增益为 32 或者反向增益为-31 11101: INM0 带外部偏置,INM1 带外部滤波的正向增益为 64 或者反向增益为-63 11110: 保留 11111: 保留
13:12	CALSEL	0x0	rw	校准选择 (Operational amplifier Calibration selection) : 用于校准, 当 CALEN 为 1 或者 VRFSEL 为 1 时有效 00: OP 的输入为 $0.033 \times VDDA$ 01: OP 的输入为 $0.1 \times VDDA$, 用于 PMOS 的校准 10: OP 的输入为 $0.5 \times VDDA$ 11: OP 的输入为 $0.9 \times VDDA$, 用于 NMOS 的校准
11	CALEN	0x0	rw	校准使能 (Operational amplifier Calibration enable) : 0: 正常模式 1: 校准模式
10:9	保留	0x0	resd	请保持为复位值。
8	OPOUTSEL	0x0	resd	OP 输出选择 (Operational amplifier Output selection) : 0: OP 输出到 GPIO 上 1: OP 输出到 ADC 的通道上
7	OPFSTMD	0x0	rw	OP 快速模式 (Operational amplifier fast mode) : 0: OP 普通模式 1: OP 快速模式
6:5	INMSEL	0x0	rw	INM 通道选择 (Operational amplifier Inverting input selection) : 00: 选择 INM0 通道 01: 选择 INM1 通道 10: 选择可配置增益模式 11: 选择跟随模式
4	USERTRIM	0x0	rw	用户校准模式 (Operational amplifier User trimming enable) : 0: 选择出厂设置校准值

				1: 选择用户设定的校准值 INP 通道选择 (Operational amplifier Non inverting input selection) : 00: 选择 INP0 通道 01: 选择 INP1 通道 10: 选择 INP2 通道 11: 选择 INP3 通道
3:2	INPSEL	0x0	rw	内部参考信号选择 (Operational amplifier Internal reference signal selection) : 0: 正常模式 1: 校准模式, 正增益作为校准电压
1	VRFSEL	0x0	rw	运算放大器使能 (Operational amplifier enable) : 0: 运算放大器不使能 1: 运算放大器使能
0	OPEN	0x0	rw	

21.6.3 OP3 控制以及状态寄存器 (OP3_CTRL)

域	简称	复位值	类型	功能
31	WPC	0x0	rw	OP 锁存控制 (Operational amplifier write protect control) 写 1 之后不能通过写 0 的方式清除, 只能通过 reset 清 0 0: OP3_CTRL 寄存器可以读写 1: OP3_CTRL 寄存器只读
30	OPCAL	0x0	r	校准成功信号 (Operational amplifier Calibration successful signal) 0: 校准中 1: 校准成功。
29	保留	0x0	resd	请保持为复位值。
28:24	TRIMVALN	0x0	rw	NMOS 的校准值 (Operational amplifier Calibration values of NMOS)
23:19	TRIMVALP	0x0	rw	PMOS 的校准值 (Operational amplifier Calibration values of PMOS)
18:14	OPGAIN	0x0	rw	OPGAIN : 可配置的增益值 (Operational amplifier Configurable gain value) : 00000: 正向增益为 2 00001: 正向增益为 4 00010: 正向增益为 8 00011: 正向增益为 16 00100: 正向增益为 32 00101: 正向增益为 64 00110: 保留 00111: 保留 01000: INMO 带外部偏置的正向增益为 2 或者反向增益为 -1 01001: INMO 带外部偏置的正向增益为 4 或者反向增益为 -3 01010: INMO 带外部偏置的正向增益为 8 或者反向增益为 -7 01011: INMO 带外部偏置的正向增益为 16 或者反向增益为 -15 01100: INMO 带外部偏置的正向增益为 32 或者反向增益为 -31 01101: INMO 带外部偏置的正向增益为 64 或者反向增益为 -63 01110: 保留 01111: 保留 10000: 带外部滤波器的正向增益为 2 10001: 带外部滤波器的正向增益为 4 10010: 带外部滤波器的正向增益为 8 10011: 带外部滤波器的正向增益为 16

				10100: 带外部滤波器的正向增益为 32 10101: 带外部滤波器的正向增益为 64 10110: 保留 10111: 保留 11000: INM0 带外部偏置,INM1 带外部滤波的正向增益为 2 或者 反向增益为-1 11001: INM0 带外部偏置,INM1 带外部滤波的正向增益为 4 或者 反向增益为-3 11010: INM0 带外部偏置,INM1 带外部滤波的正向增益为 8 或者 反向增益为-7 11011: INM0 带外部偏置,INM1 带外部滤波的正向增益为 16 或者 反向增益为-15 11100: INM0 带外部偏置,INM1 带外部滤波的正向增益为 32 或者 反向增益为-31 11101: INM0 带外部偏置,INM1 带外部滤波的正向增益为 64 或者 反向增益为-63 11110: 保留 11111: 保留
13:12	CALSEL	0x0	rw	校准选择 (Operational amplifier Calibration selection) : 用于校准, 当 CALEN 为 1 或者 VRFSEL 为 1 时有效 00: OP 的输入为 $0.033 \times VDDA$ 01: OP 的输入为 $0.1 \times VDDA$, 用于 PMOS 的校准 10: OP 的输入为 $0.5 \times VDDA$ 11: OP 的输入为 $0.9 \times VDDA$, 用于 NMOS 的校准
11	CALEN	0x0	rw	校准使能 (Operational amplifier Calibration enable) : 0: 正常模式 1: 校准模式
10:9	保留	0x0	resd	请保持为复位值。
8	OPOUTSEL	0x0	resd	OP 输出选择 (Operational amplifier Output selection) : 0: OP 输出到 GPIO 上 1: OP 输出到 ADC 的通道上
7	OPFSTMD	0x0	rw	OP 快速模式 (Operational amplifier fast mode) : 0: OP 普通模式 1: OP 快速模式
6:5	INMSEL	0x0	rw	INM 通道选择 (Operational amplifier Inverting input selection) : 00: 选择 INM0 通道 01: 选择 INM1 通道 10: 选择可配置增益模式 11: 选择跟随模式
4	USERTRIM	0x0	rw	用户校准模式 (Operational amplifier User trimming enable) : 0: 选择出厂设置校准值 1: 选择用户设定的校准值
3:2	INPSEL	0x0	rw	INP 通道选择 (Operational amplifier Non inverting input selection) : 00: 选择 INP0 通道 01: 选择 INP1 通道 10: 选择 INP2 通道 11: 选择 INP3 通道
1	VRFSEL	0x0	rw	内部参考信号选择 (Operational amplifier Internal reference signal selection) : 0: 正常模式 1: 校准模式, 正增益作为校准电压
0	OPEN	0x0	rw	运算放大器使能 (Operational amplifier enable) : 0: 运算放大器不使能 1: 运算放大器使能

21.6.4 OP4 控制以及状态寄存器 (OP4_CTRL)

域	简称	复位值	类型	功能
31	WPC	0x0	rw	OP 锁存控制 (Operational amplifier write protect control) 写 1 之后不能通过写 0 的方式清除，只能通过 reset 清 0 0: OP4_CTRL 寄存器可以读写 1: OP4_CTRL 寄存器只读
30	OPCAL	0x0	r	校准成功信号 (Operational amplifier Calibration successful signal) 0: 校准中 1: 校准成功。
29	保留	0x0	resd	请保持为复位值。
28:24	TRIMVALN	0x0	rw	NMOS 的校准值 (Operational amplifier Calibration values of NMOS)
23:19	TRIMVALP	0x0	rw	PMOS 的校准值 (Operational amplifier Calibration values of PMOS)
18:14	OPGAIN	0x0	rw	OPGAIN : 可配置的增益值 (Operational amplifier Configurable gain value) : 00000: 正向增益为 2 00001: 正向增益为 4 00010: 正向增益为 8 00011: 正向增益为 16 00100: 正向增益为 32 00101: 正向增益为 64 00110: 保留 00111: 保留 01000: INM0 带外部偏置的正向增益为 2 或者反向增益为 -1 01001: INM0 带外部偏置的正向增益为 4 或者反向增益为 -3 01010: INM0 带外部偏置的正向增益为 8 或者反向增益为 -7 01011: INM0 带外部偏置的正向增益为 16 或者反向增益为 -15 01100: INM0 带外部偏置的正向增益为 32 或者反向增益为 -31 01101: INM0 带外部偏置的正向增益为 64 或者反向增益为 -63 01110: 保留 01111: 保留 10000: 带外部滤波器的正向增益为 2 10001: 带外部滤波器的正向增益为 4 10010: 带外部滤波器的正向增益为 8 10011: 带外部滤波器的正向增益为 16 10100: 带外部滤波器的正向增益为 32 10101: 带外部滤波器的正向增益为 64 10110: 保留 10111: 保留 11000: INM0 带外部偏置,INM1 带外部滤波的正向增益为 2 或者反向增益为 -1 11001: INM0 带外部偏置,INM1 带外部滤波的正向增益为 4 或者反向增益为 -3 11010: INM0 带外部偏置,INM1 带外部滤波的正向增益为 8 或者反向增益为 -7 11011: INM0 带外部偏置,INM1 带外部滤波的正向增益为 16 或者反向增益为 -15 11100: INM0 带外部偏置,INM1 带外部滤波的正向增益为 32 或者反向增益为 -31

				11101: INM0 带外部偏置,INM1 带外部滤波的正向增益为 64 或者反向增益为-63 11110: 保留 11111: 保留
13:12	CALSEL	0x0	rw	校准选择 (Operational amplifier Calibration selection) : 用于校准, 当 CALEN 为 1 或者 VRFSEL 为 1 时有效 00: OP 的输入为 $0.033 \times VDDA$ 01: OP 的输入为 $0.1 \times VDDA$, 用于 PMOS 的校准 10: OP 的输入为 $0.5 \times VDDA$ 11: OP 的输入为 $0.9 \times VDDA$, 用于 NMOS 的校准
11	CALEN	0x0	rw	校准使能 (Operational amplifier Calibration enable) : 0: 正常模式 1: 校准模式
10:9	保留	0x0	resd	请保持为复位值。
8	OPOUTSEL	0x0	resd	OP 输出选择 (Operational amplifier Output selection) : 0: OP 输出到 GPIO 上 1: OP 输出到 ADC 的通道上
7	OPFSTMD	0x0	rw	OP 快速模式 (Operational amplifier fast mode) : 0: OP 普通模式 1: OP 快速模式
6:5	INMSEL	0x0	rw	INM 通道选择 (Operational amplifier Inverting input selection) : 00: 选择 INM0 通道 01: 选择 INM1 通道 10: 选择可配置增益模式 11: 选择跟随模式
4	USERTRIM	0x0	rw	用户校准模式 (Operational amplifier User trimming enable) : 0: 选择出厂设置校准值 1: 选择用户设定的校准值
3:2	INPSEL	0x0	rw	INP 通道选择 (Operational amplifier Non inverting input selection) : 00: 选择 INP0 通道 01: 选择 INP1 通道 10: 选择 INP2 通道 11: 选择 INP3 通道
1	VRFSEL	0x0	rw	内部参考信号选择 (Operational amplifier Internal reference signal selection) : 0: 正常模式 1: 校准模式, 正增益作为校准电压
0	OPEN	0x0	rw	运算放大器使能 (Operational amplifier enable) : 0: 运算放大器不使能 1: 运算放大器使能

21.6.5 OP1 timer 控制寄存器 (OP1_TC)

域	简称	复位值	类型	功能
31	WPC	0x0	rw	OP1 timer 控制寄存器的锁存控制 (Operational amplifier write protect control) 写 1 之后不能通过写 0 的方式清除, 只能通过 reset 清 0 0: OP1_TC 寄存器可以读写 1: OP1_TC 寄存器只读
30:4	保留	0x0	resd	请保持为复位值。
3	TMR1EN	0x0	rw	Timer1 切换 INP/INM 通道使能信号 (Timer1 Switching INP/INM channel enable signal) 该信号使能之后, 通过 timer1 的 CC6 通道输出自动切换 INP/INM 通道。 0: 自动切换关闭

				1: 自动切换打开 INP 通道切换值 (INP channel switching value) 当 TMR1EN 使能之后, timer1 的 CC6 为 1 时选择 INPSSEL, 为 0 时选择 INPSEL
2:1	INPSSEL	0x0	rw	00: 选择 INP0 通道 01: 选择 INP1 通道 10: 选择 INP2 通道 11: 选择 INP3 通道
0	INMSSEL	0x0	rw	INM 通道切换值 (INM channel switching value) 当 TMR1EN 使能之后, timer1 的 CC6 为 1 时选择 INMSSEL, 为 0 时选择 INMSEL[0] 当 INMSEL 为 00 或 01 时 0: 选择 INM0 通道 1: 选择 INM1 通道 当 INMSEL[1] 为 1 时 0: 可配置增益模式 1: 跟随模式

21.6.6 OP2 timer 控制寄存器 (OP2_TC)

域	简称	复位值	类型	功能
31	WPC	0x0	rw	OP2 timer 控制寄存器的锁存控制 (Operational amplifier write protect control) 写 1 之后不能通过写 0 的方式清除, 只能通过 reset 清 0 0: OP2_TC 寄存器可以读写 1: OP2_TC 寄存器只读
30:4	保留	0x0	resd	请保持为复位值。
3	TMR1EN	0x0	rw	Timer1 切换 INP/INM 通道使能信号 (Timer1 Switching INP/INM channel enable signal) 该信号使能之后, 通过 timer1 的 CC6 通道输出自动切换 INP/INM 通道。 0: 自动切换关闭 1: 自动切换打开
2:1	INPSSEL	0x0	rw	INP 通道切换值 (INP channel switching value) 当 TMR1EN 使能之后, timer1 的 CC6 为 1 时选择 INPSSEL, 为 0 时选择 INPSEL 00: 选择 INP0 通道 01: 选择 INP1 通道 10: 选择 INP2 通道 11: 选择 INP3 通道
0	INMSSEL	0x0	rw	INM 通道切换值 (INM channel switching value) 当 TMR1EN 使能之后, timer1 的 CC6 为 1 时选择 INMSSEL, 为 0 时选择 INMSEL[0] 当 INMSEL 为 00 或 01 时 0: 选择 INM0 通道 1: 选择 INM1 通道 当 INMSEL[1] 为 1 时 0: 可配置增益模式 1: 跟随模式

21.6.7 OP3 timer 控制寄存器 (OP3_TC)

域	简称	复位值	类型	功能
31	WPC	0x0	rw	OP3 timer 控制寄存器的锁存控制 (Operational amplifier write protect control) 写 1 之后不能通过写 0 的方式清除, 只能通过 reset 清 0 0: OP3_TC 寄存器可以读写 1: OP3_TC 寄存器只读
30:4	保留	0x0	resd	请保持为复位值。
3	TMR1EN	0x0	rw	Timer1 切换 INP/INM 通道使能信号 (Timer1 Switching INP/INM channel enable signal)

				该信号使能之后，通过 timer1 的 CC6 通道输出自动切换 INP/INM 通道。 0: 自动切换关闭 1: 自动切换打开
2:1	INPSSEL	0x0	rw	INP 通道切换值 (INP channel switching value) 当 TMR1EN 使能之后，timer1 的 CC6 为 1 时选择 INPSSEL, 为 0 时选择 INPSEL 00: 选择 INP0 通道 01: 选择 INP1 通道 10: 选择 INP2 通道 11: 选择 INP3 通道
0	INMSSEL	0x0	rw	INM 通道切换值 (INM channel switching value) 当 TMR1EN 使能之后，timer1 的 CC6 为 1 时选择 INMSSEL, 为 0 时选择 INMSEL[0] 当 INMSEL 为 00 或 01 时 0: 选择 INM0 通道 1: 选择 INM1 通道 当 INMSEL[1] 为 1 时 0: 可配置增益模式 1: 跟随模式

21.6.8 OP4 timer 控制寄存器 (OP4_TC)

域	简称	复位值	类型	功能
31	WPC	0x0	rw	OP4 timer 控制寄存器的锁存控制 (Operational amplifier write protect control) 写 1 之后不能通过写 0 的方式清除，只能通过 reset 清 0 0: OP4_TC 寄存器可以读写 1: OP4_TC 寄存器只读
30:4	保留	0x0	resd	请保持为复位值。
3	TMR1EN	0x0	rw	Timer1 切换 INP/INM 通道使能信号 (Timer1 Switching INP/INM channel enable signal) 该信号使能之后，通过 timer1 的 CC6 通道输出自动切换 INP/INM 通道。 0: 自动切换关闭 1: 自动切换打开
2:1	INPSSEL	0x0	rw	INP 通道切换值 (INP channel switching value) 当 TMR1EN 使能之后，timer1 的 CC6 为 1 时选择 INPSSEL, 为 0 时选择 INPSEL 00: 选择 INP0 通道 01: 选择 INP1 通道 10: 选择 INP2 通道 11: 选择 INP3 通道
0	INMSSEL	0x0	rw	INM 通道切换值 (INM channel switching value) 当 TMR1EN 使能之后，timer1 的 CC6 为 1 时选择 INMSSEL, 为 0 时选择 INMSEL[0] 当 INMSEL 为 00 或 01 时 0: 选择 INM0 通道 1: 选择 INM1 通道 当 INMSEL[1] 为 1 时 0: 可配置增益模式 1: 跟随模式

22 控制器区域网络 (CAN)

AT32M416 额外支持 CANFD 协议, AT32M412 则不支持。

22.1 概述

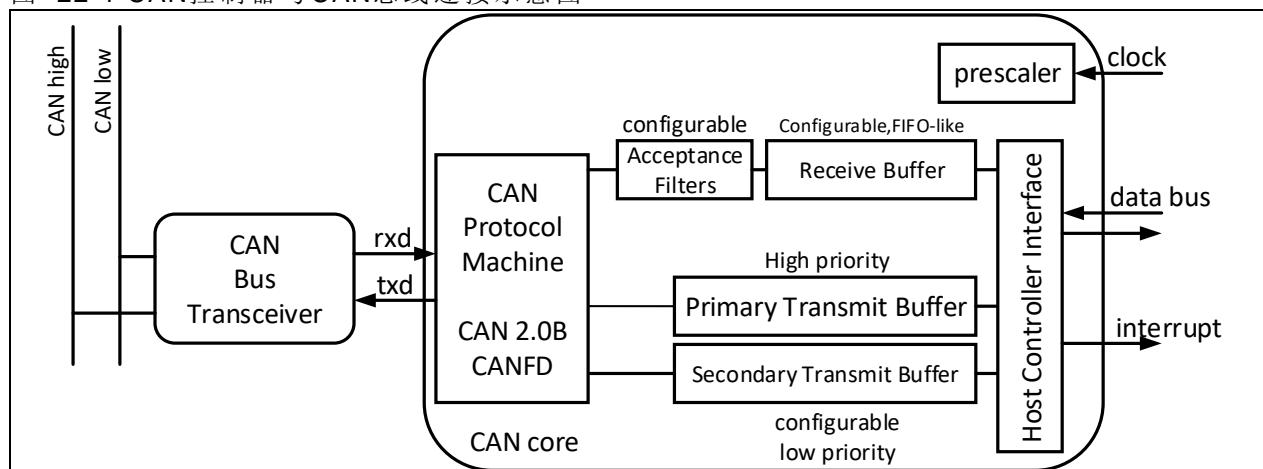
22.1.1 CAN控制器内核

作为串行通信控制器, CAN 控制器内核是基于 CAN 协议实现串行通信功能, 并且满足 CAN 规范的所有约束性要求。

- Classic CAN2.0B
- CANFD (ISO 11898-1:2015)

CAN 协议定义了网络节点之间数据帧的传输 (即通信对象), 以及错误处理的管理机制。CAN 控制器可以实现在各种部件之间建立经济可靠的连接, 无需主机控制器参与处理 CAN 协议。对于微控制器而言, CAN IP 核相当于一个内存映射 IO 设备。CPU 访问 CAN 控制器, 控制双线 CAN 总线实现帧数据的收发。CAN IP 与 CAN 总线的连接方式如下图所示。

图 22-1 CAN控制器与CAN总线连接示意图



22.1.2 CAN控制器协议

CAN 控制器以数据帧的方式进行通信, 所有错误帧和过载帧均由 CAN 控制器自动处理。数据帧用于传输主机应用程序的数据, 几种格式的数据帧如图 22-2 和图 22-3 所示。Classic CAN2.0B 的一个数据帧可传输数据高达 8 字节, CANFD 可传输数据高达 64 字节。

帧标识符用于数据寻址, 在 CAN 网络中, 同时只有一个节点发送带有某个标识符的帧数据, 其它节点处于接收状态。主机控制器必须判断其收到的帧是否符合需求, 为了减少主机控制器负载, CAN 节点可以使用接收过滤器。过滤器将接收到的帧标识符与用户定义的标识符进行比较, 只有通过了接收过滤器的帧才会存放至接收缓存中, 并通知主机控制器。

CAN 帧数据的标识符用于总线仲裁, 当优先级低的标识符的帧与优先级较高的标识符的帧同时发送时, CAN 控制器将停止发送优先级低的标识符的帧, 之后会在总线空闲时自动重新开始发送被中断的帧。

CAN2.0B 定义的数据比特率最高为 1Mbit/s. 对于 CANFD 而言, 理论上是没有限制的, 但实际上会受到所使用的收发器的电气特性和总线结构的影响。CANFD 的比特率是可以切换的, 如果开启了 CANFD 比特率切换功能, 那么帧的数据负载将以更高速度传输, 而帧头将以较低速度传输。

22.1.3 Classic CAN2.0B 以及 CANFD

CAN 协议历经多年的发展与完善, 最早的 CAN 协议为 Classic CAN2.0B, 之后进版为 CANFD (ISO11898-1:2015), CAN 控制器可以接收和发送所有的 CANFD 以及 Classic CAN2.0B 帧, 另外, CAN 控制器可以向上兼容。

CAN 控制器可以动态选择当前帧数据是作为 Classic CAN2.0B 帧、CANFD 帧发送, 同样接收缓存的帧数据相应位会显示所接收到的帧数据是 Classic CAN2.0B 帧, 还是 CANFD 帧。Classic CAN2.0B 帧类

型见图 22-3 所示，CANFD 帧类型见图 22-2 所示。
图 22-2 CANFD 帧类型

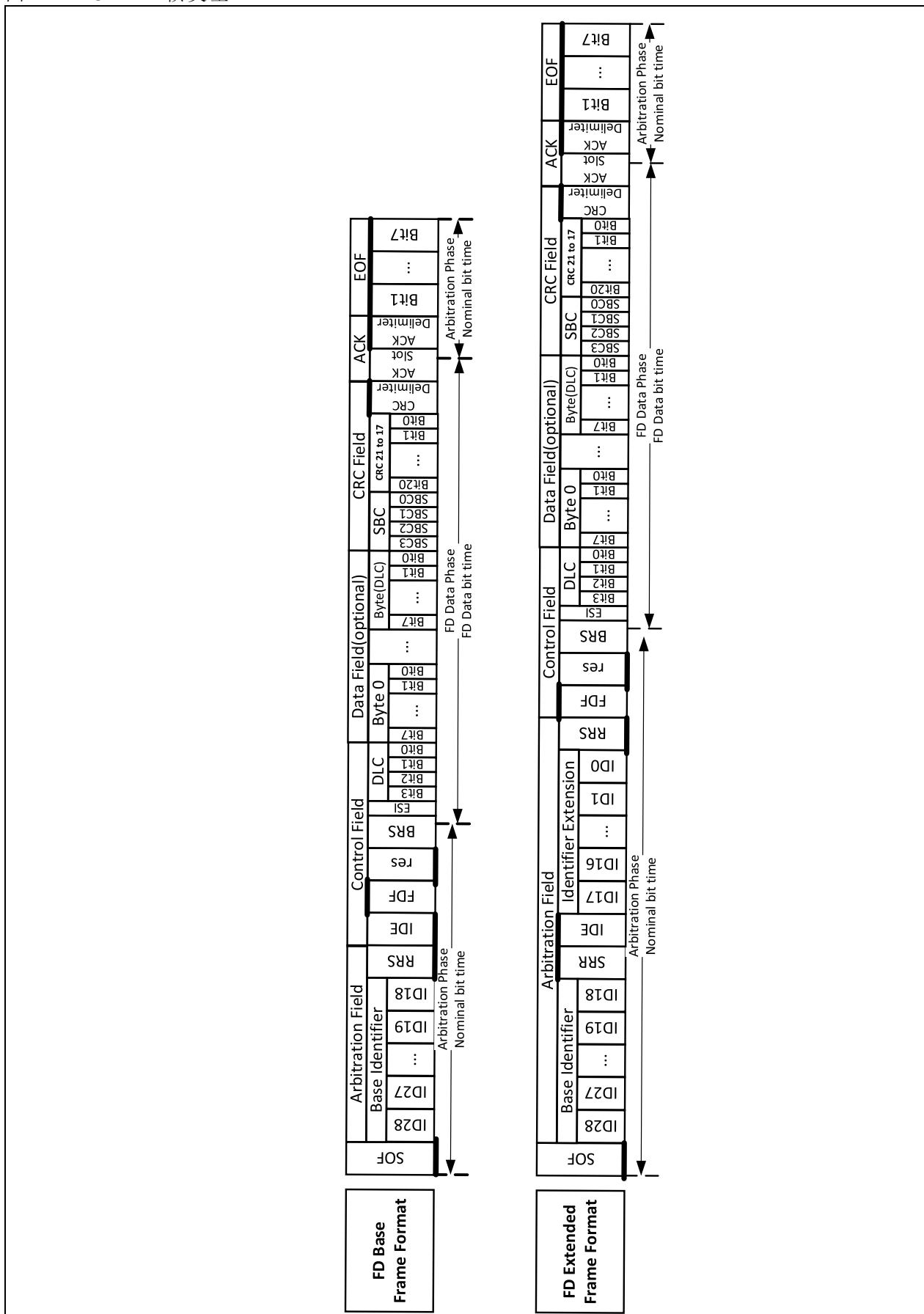
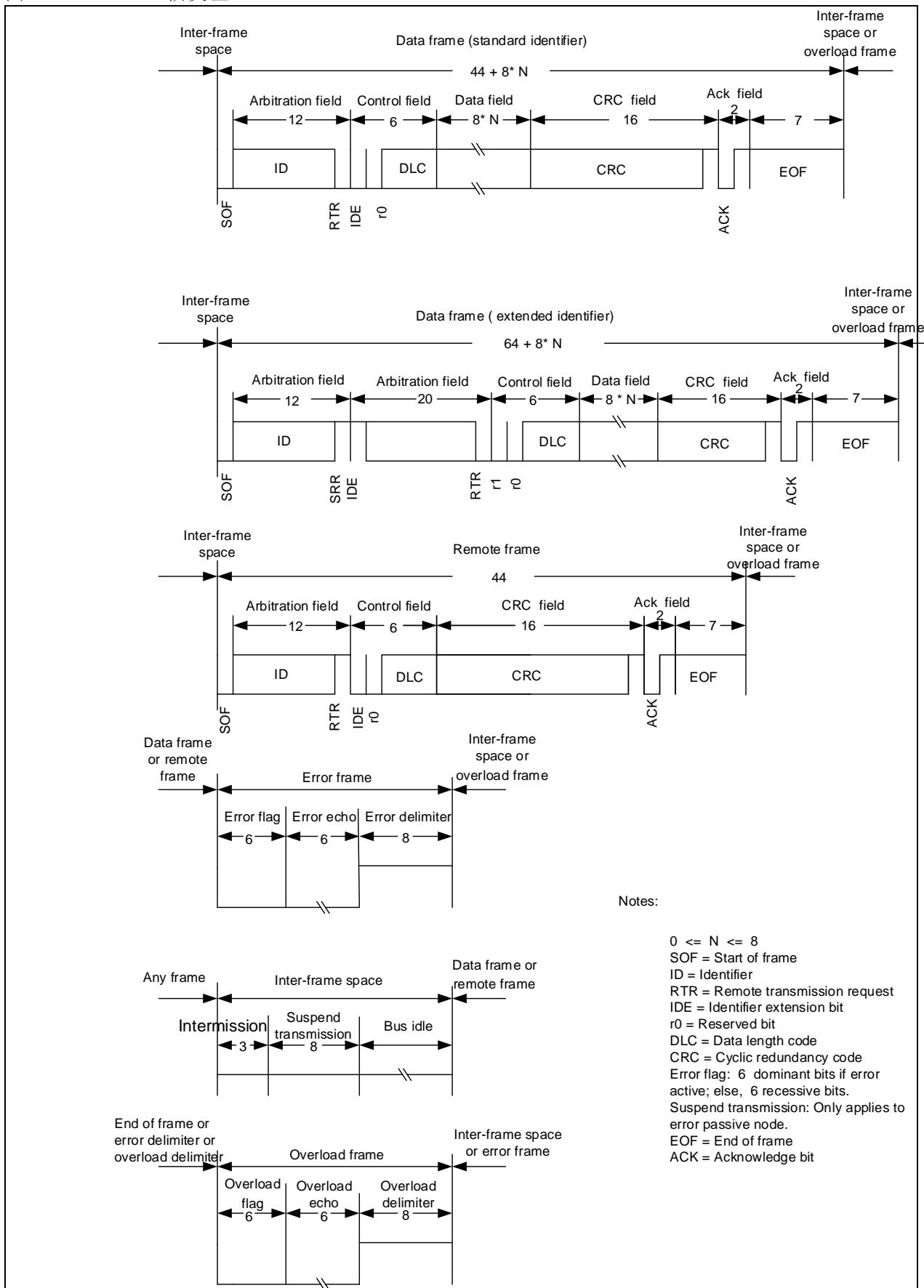


图 22-3 CAN2.0帧类型



22.1.4 时间触发

基于 ISO11898-4 协议, CAN 控制器可以用于时间触发 CAN 通信 (TTCAN), CAN 控制器提供部分硬件支持, 在此模式下, 需要主机软件实时交互。

TTCAN 的基本概念是通过定时器记录接收帧数据的时间戳和触发帧数据发送, CAN 网络中存在一个时间主机, 时间主机用于发送参考报文, 周期时间开始于参考报文, 两个参考报文之间为一个基本周期, 在基本周期内, 报文可以在时间窗口内被发送。TTCAN 系统管理者定义脱机设置期间, 每个时间窗的开始时间和持续时间。

总共有 3 种时间窗:

- 专用时间窗 (只允许一个节点传输一个带有特定 ID 的帧)
- 自由时间窗 (保留用作网络系统扩展)
- 仲裁时间窗 (多个节点可以传输一个帧, 此时引发仲裁)

当接收到一帧数据, 该节点记录实际的周期时间作为时间戳, 同时, CAN 控制器提供了一个硬件触发, 用于在预先设定的周期内发送一个预先设定的帧。

CAN 控制器在接收时自动检测参考报文并开始周期时间, 硬件定时器是以 CAN 位时间为基准的 16 位定时器, 满足 ISO11898-4 Level1 协议, CAN 控制器可以用于时间主机。

除了硬件触发帧传输之外, CAN 控制器还提供了一个监视触发器用于监测参考报文的缺失。

所谓的部分硬件支持, 它指的是主机控制器需要为每个时间窗预先设置节点操作。比如, 主机需要定义下一个待传输的帧, 并为之设置触发时间。

22.1.5 CiA 603时间戳

CAN 控制器包含 CAN 总线组织 (CiA) 在 CiA 603 协议中定义的一个最低 16 位的时间戳, CiA 603 时间戳独立于 TTCAN 运行。CiA 603 的基本概念是包含一个自由运行的计数器, 该计数器根据时钟周期而非 CAN 位时间进行计数, 计数器的精度至少 10us (16 位) 或 1us (32 位或更多), 时间戳在 CAN/CANFD 帧的 SOF 或者 EOF 位置被记录。

CiA 603 支持 AUTOSAR 时间戳和时间同步功能, 对于 AUTOSAR 而言, CAN 网络中的一个节点就是一个时间主机, 时间主机发送同步报文 (SYNC 报文), SYNC 报文的时间戳被主机和所有的从机获取, 从发出指令传输 SYNC 报文到 SYNC 报文实际传输之间存在时间差, 该时间差是由主机以 FUP 报文的方式进行发送的。CiA 603 规定了读取和修改定时器的相关规则, CAN 控制器不包含该定时器, 而使用外部定时器。CAN 控制器只包含时间戳运行机制, 寄存器存储发送时间戳 (TTS), 缓存存储所有接收帧的接收时间戳。

22.2 特性

22.2.1 特征列表

- 支持 CAN 协议
 - CAN2.0B: 高达 8 字节有效数据
 - CANFD: 高达 64 字节有效载荷, 支持 ISO 11898-1:2015 或 non-ISO Bosch
- 帧速率可自由配置
 - CAN2.0B 定义的速率最高为 1Mbit/s
 - CANFD 受限于收发器和 CAN 控制器的时钟频率
- 可编程的波特率分频器 (从 1 到 1/32 可选)
- 接收缓冲器 (RB) 为 6 级深度
 - 接收缓冲器操作机制类似 FIFO
 - 接收到的“不正确”或“不能过滤”的帧不会覆盖已存储的帧
- 两个发送缓冲器
 - 主发送缓冲器 (PTB) (一个帧槽)
 - 额外 3 级深度次发送缓冲器 (STB), 运行于 FIFO 或优先级模式
- 独立的可编程内部接收过滤器
 - 16 个接收过滤器
- 扩展特性
 - 重新仲裁和重新发送的限制 (1 到 7 次或者“无限”尝试)

- 只听模式
- 环回模式（包括内部和外部）
- 收发器待机模式
- 扩展状态和错误报告
 - 传输帧的状态报告
 - 获取最近一次发生错误的类型和仲裁丢失位置
 - 可编程的错误警告限制次数
- 可设置中断源
- 时间戳
 - ISO 11898-4时间触发CAN，部分硬件支持
 - CiA 603时间戳
- 兼容AUTOSAR
- 针对SAE J1939进行了优化

22.2.2 中断

CAN 控制器具有 4 个中断向量，通过配置 CAN 控制器中断使能寄存器（INTEN），可以控制相应的中断开启或关闭。

图 22-4 CAN1发送中断的产生

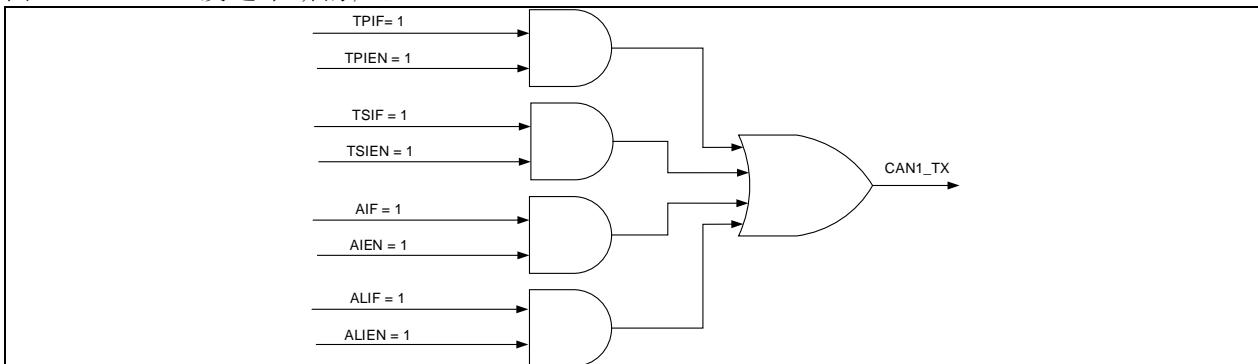


图 22-5 CAN1接收中断的产生

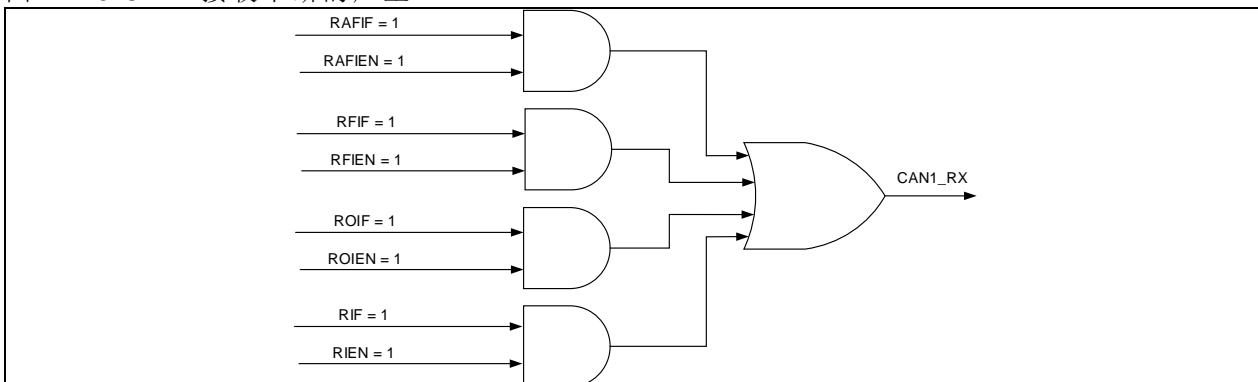


图 22-6 CAN1状态中断的产生

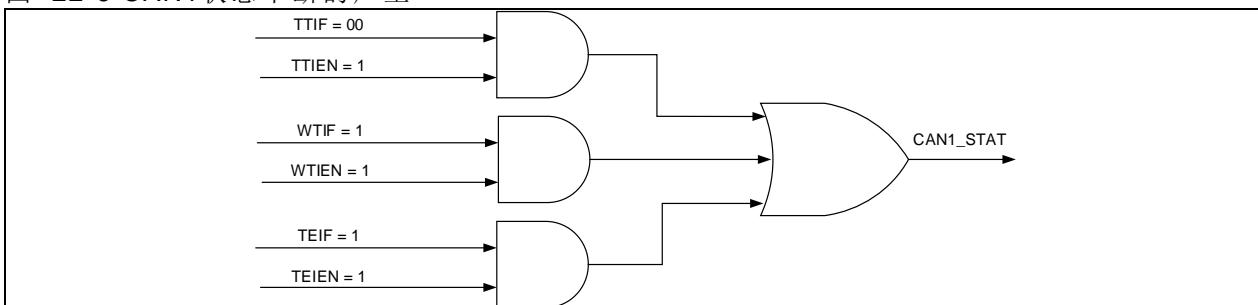
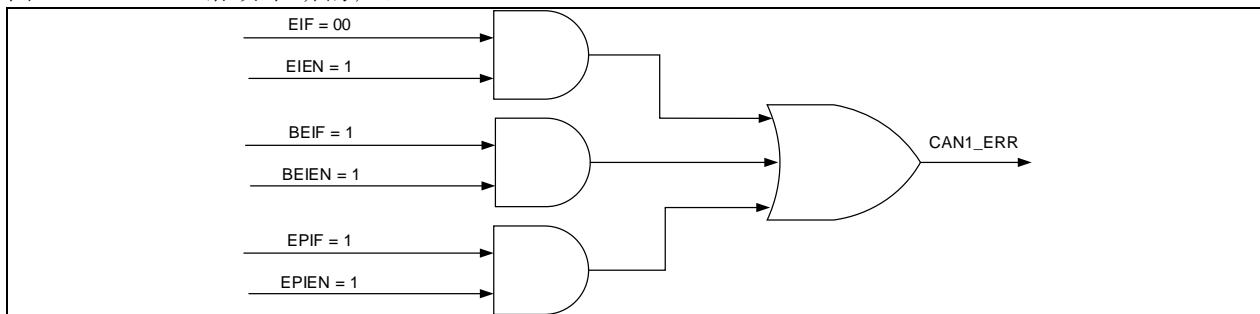


图 22-7 CAN1错误中断的产生



22.2.3 软件操作接口

表 22-1 显示了带有时间戳的逻辑链路控制帧 (LLC 帧) 的定义，主机应用与 CAN 控制器之间按照 LLC 帧格式进行数据交换，这个统一的定义用于往 TBUF 中存储待发送数据、从 RBUF 中读出接收到的帧数据以及配置接收过滤器 (包含 ACFC 和 ACFM)。

对于接收过滤器，只有包含标识符、格式、类型的帧头是有意义的，应用于配置 ACFC 和 ACFM 寄存器，负载数据、CiA 603 和 TTCAN 时间戳对于 ACF 不可用。

接收时间戳 (CiA 603 和 TTCAN) 只对于 RBUF 有意义，对于 TBUF 不可用。

接收时间戳 (CiA 603 和 TTCAN) 的地址偏移位置 T 是动态变化的，T 值基于字对齐，通过使用 LLCSIZE 寄存器，应用可以方便的处理 LLC 帧。

表 22-1 LLC 帧 (逻辑链路控制帧) 定义 (包含时间戳)

地址	比特位								功能	
偏移值	7	6	5	4	3	2	1	0		
0	ID(7:0)								标识符 (ID)	
1	ID(15:8)									
2	ID(23:16)									
3	TTSEN	-		ID(28:24)						
4					DLC(3:0)				格式 (FMT)	
5	-									
6	-		RMF	-	BRS	FDF	IDE			
7	-		LBF	ESI	KOER(2:0)					
8	-								类型 (TYP)	
9	-									
10	-									
11	HANDLE(7:0)									
12	AF(7:0)								接收场 (ACF)	
13	AF(15:8)									
14	AF(23:16)									
15	AF(31:24)									
16	d1(7:0)								负载数据	
17	d2(7:0)									
...	...									
79	D64(7:0)									
T+0	RTS(7:0)								CiA 603	

...	...	(接收)
T+7	RTS(63:56)	
T+8	CYCLE_TIME(7:0)	
T+9	CYCLE_TIME(15:8)	
T+10	-	TTCAN (接收)
T+11	-	

表 22-2 LLC帧缩写定义

比特位	描述		
ID	帧标识符		
	接收: 有效	发送: 有效	过滤器: 有效
TTSEN	发送时间戳 (Time-Stamp) 使能 对于 CiA 603 时间戳，可以选择是否获取发送时间戳 TTS 0: 当前帧的发送时间戳不更新 1: TTS 更新使能		
	接收: 无效	发送: 有效	过滤器: 无效
DLC	数据长度代码, DLC 定义了一帧数据的负载数据长度		
	接收: 有效	发送: 有效	过滤器: 有效
IDE	扩展标识符标志 0: 标准帧格式, ID(28:18) 1: 扩展帧格式, ID(28:0)		
	接收: 有效	发送: 有效	过滤器: 有效
FDF	CANFD 帧格式 FDF=0: CAN2.0 帧 (高达 8 字节负载) FDF=1: CANFD 帧 (高达 64 字节负载)		
	接收: 有效	发送: 有效	过滤器: 有效
BRS	CANFD 比特率切换使能 0: 整帧采用“慢速”比特率 (标称比特率) 1: CANFD 帧的数据场和 CRC 场切换至“快速”比特率 (数据比特率)		
	接收: 有效	发送: 有效	过滤器: 有效
RMF	远程帧 0: 数据帧 1: 远程帧		
	接收: 有效	发送: 有效	过滤器: 有效
KOER	错误类型 对于接收帧, 当 RBALL=1 时, KOER 变得有意义, 如果 RBALL=1, 则接收过滤器通常被关闭。		
	接收: 有效	发送: 无效	接收过滤器: 无效
ESI	错误状态指示, 协议机自动将正确的 ESI 值嵌入到发送的帧数据中 0: CAN 节点处于主动错误状态 1: CAN 节点处于被动错误状态 ESI 只包含在 CANFD 帧数据中, CAN2.0B 帧该位固定为 0.		
	接收: 有效	发送: 无效	过滤器: 有效
LBF	环回帧 对于接收帧而言, 如果启用了环回模式, 则 LBF 设置为 1, 并且 CAN 控制器接收自己发送的帧数据。如果 LBME=1, 而总线上其他节点也正在传输, LBF 可能很有用。		
	接收: 有效	发送: 无效	过滤器: 有效
HANDLE	帧标识手柄 帧标识手柄应用于 TSTAT 中, 建议应用程序将软件计数值写入至 HANDLE.		
	接收: 无效	发送: 有效	过滤器: 无效

AF	接收场, 只对于 CAN XL 帧格式可用。		
	接收: 有效	发送: 有效	过滤器: 有效
Payload Data	帧的负载数据, Classic CAN2.0B 最多为 8 个字节, CANFD 最多为 64 个字节, 参考表 22-3 见具体细节, RBUF 未使用的负载数据填充无效数据, 直接忽略。		
	接收: 有效	发送: 有效	过滤器: N/A
RTS	CiA 603 时间戳的接收时间戳 每接收一帧数据存储一次 RTS, 与 TTS 相对, RTS 与一个特定的接收帧相关 如果 TSEN=0, 则 RTS=0		
	接收: 有效	发送: N/A	过滤器: N/A
CYCLE_TIME	周期时间 (TTCAN 时间戳) 只有接收帧才存储 CYCLE_TIME, 为帧 SOF 位的周期时间, 参考帧的周期时间恒为 0		
	接收: 有效	发送: N/A	过滤器: N/A

表 22-3 DLC 的定义

DLC (二进制)	帧类型	负载数据字节
DLC(3:0)=0000 至 1000	Classic CAN2.0B 和 CANFD	0 到 8
DLC(3:0)=1001 至 1111	Classic CAN2.0B	8
DLC(3:0)=1001	CANFD	12
DLC(3:0)=1010	CANFD	16
DLC(3:0)=1011	CANFD	20
DLC(3:0)=1100	CANFD	24
DLC(3:0)=1101	CANFD	32
DLC(3:0)=1110	CANFD	48
DLC(3:0)=1111	CANFD	64

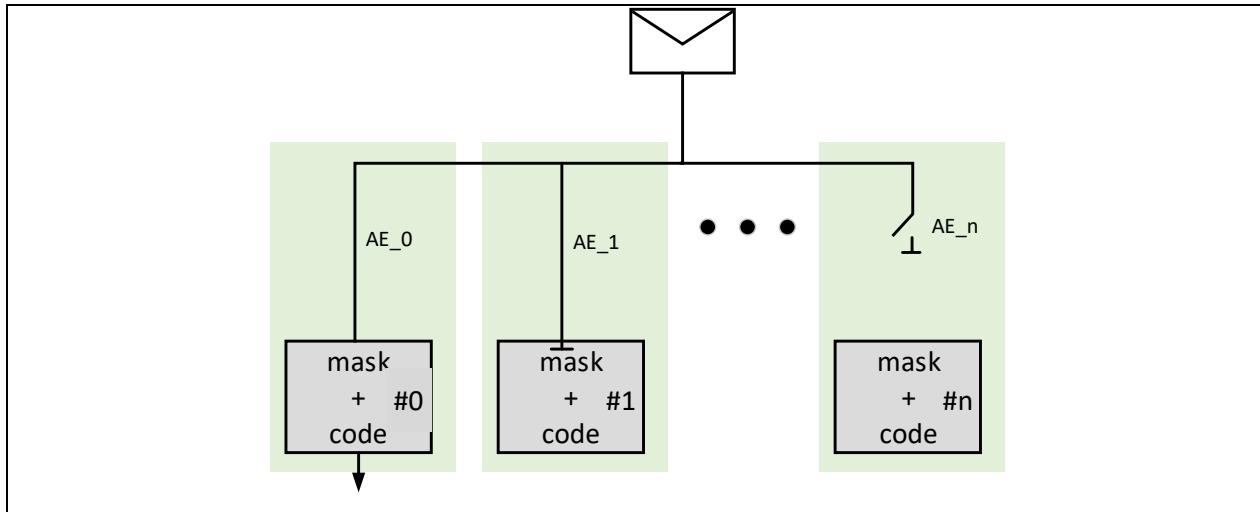
22.3 操作指南

22.3.1 接收过滤器

CAN 控制器使用接收过滤器以减轻主机控制器接收帧的负载, 在接收过滤过程中, CAN 控制器会检查帧头。只有通过了过滤器的帧才会被接受, 若过滤通过, 通过接收过滤器的帧被将存放在 RB 内, 并置起 RIF 标志。如果过滤失败, 则不会置起 RIF 标志, 且 RB FIFO 指针也不会增加。未通过接收过滤器的帧会被丢弃, 并被下一个帧所覆盖, 未通过过滤的帧不会覆盖已接收并存储的有效帧数据。

接收过滤器仅适用于有效帧。(如果在传输过程中出现错误,CAN 控制器将根据 CAN 协议进行错误处理)。

图 22-8 接收过滤器示例



接收屏蔽定义的是需要比较哪些位，而接收代码定义的是相应值。若将验收屏蔽位设置为 0，则使能比较所选的接收代码位与接收帧的相应位。若将接收屏蔽位设置为 1，则会关闭接收过滤器检查，并直接接收该帧。（“屏蔽”意味着“不关心”）。

示例：ID(28)指所有 CAN 帧标识符的最高有效位，是最先发送的。假设 ACFM 的 ID(28)位为 0，其他位为 1，那么若要接收一个接收帧，则该接收帧的 ID(28)的值必须等于 ACFC 的 ID(28)，其他位在进行接收过滤时被忽略。

图 22-8 示例中，通过使用多个过滤器演示如何进行验收过滤，在本例中，通过 ACFCTRL 寄存器的 AE_0 和 AE_1 位使能过滤器 0 和过滤器 1，其他的过滤器处于关闭状态，所以不会接收任何帧。对于这两个已使能的过滤器而言，使用接收屏蔽位和接收代码来定义是否接收帧。在本例中，过滤器 0 接受该帧，而过滤器 1 不接收该帧。

注：设置 $AE_x=0$ 将禁用该过滤器，并禁止接收帧。与此相反，ACFM 中的屏蔽位关闭了对应位的接收过滤检查，从而导致接收该帧。

在上电复位后，CAN 控制器配置为接收所有帧（通过设置 $AE_0=1$ 使能过滤器 0，并且 ACFM 的所有位为 1，其他过滤器禁用，只有过滤器 0 定义了 ACFC/ACFM 复位值，其他过滤器没有定义复位值）。

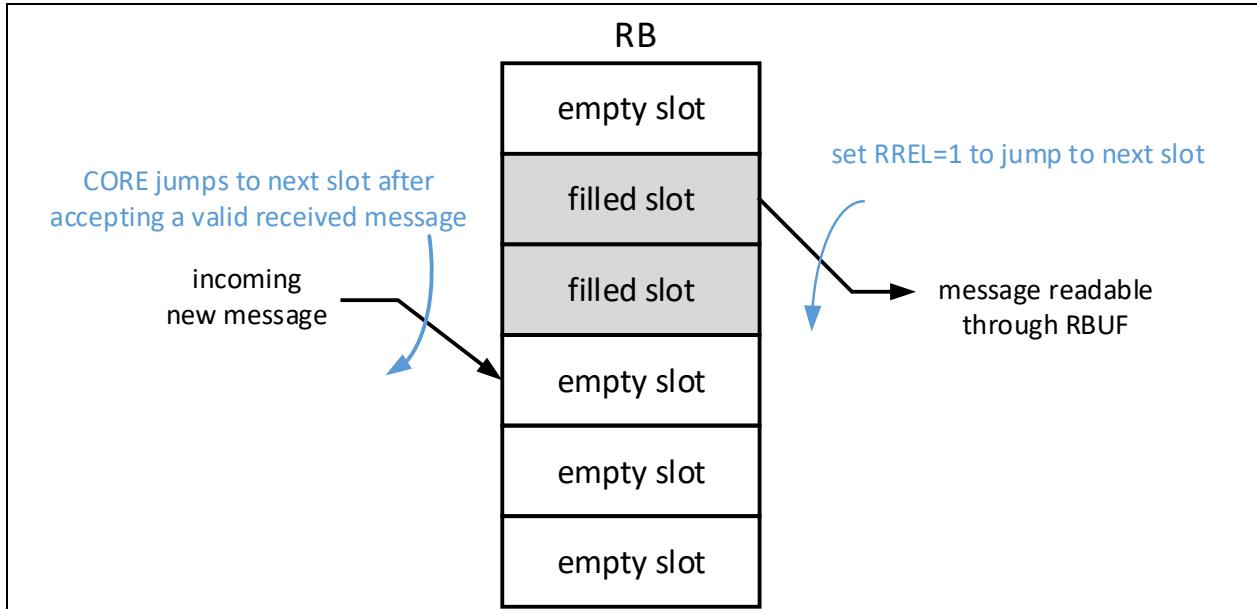
其它注意事项：

- 通过正确设置屏蔽位，一个单独的接收过滤器可以接受多组帧。CAN控制器最多有16个验收过滤器，意味着最多可以定义16个过滤器组。
- 接收过滤器的结构是基于LLC帧定义进行设计的，目的是提供最灵活的验收过滤组定义。例如，可以通过将BaseID/PriorityID的最重要位（即ID(28)）设置为某个值，即可接受各种类别的CAN帧（如CAN2.0B, CANFD）。
- 有些位并不适用于所有CAN帧类型（CAN2.0B, CANFD），或者是强制为固定值。例如，对于IDE位，其掩码设置为0，代码设置为1，那么就意味着只接受IDE=1的帧。
- 主机应用程序负责正确配置接收过滤器，例如，取消屏蔽LLC帧的未使用位是没有意义的。CAN控制器不提供错误配置保护机制（即CAN控制器无法防止此类配置错误），因此，除了一些必需的位之外，必须屏蔽所有位（包括“未使用的位”）

22.3.2 帧接收

如下图所示，所接收到的数据是存放在 RB 内，RB 有 6 级深度，RB 的运作模式类似 FIFO。如果接收到的数据是有效的，且已通过过滤，则会置起 RIF=1，RSTAT 的值取决于填充状态。当所填充的缓冲器的数量等于设定值 AFWL，将置起 RAFIF 标志。一旦所有的缓冲器都被填满，则将置位 RFIF 标志。

图 22-9 接收缓冲器 RB 结构图



RBUF 寄存器始终映射到 RB 中存储最早接收帧的槽空间，Classic CAN2.0B 帧的最大有效载荷为 8 字节，CANFD 帧为 64 字节。帧长度由 DLC 设置。RB 提供了槽空间足够大，可以存储最大帧长度。需要通过主机控制器设置 RREL 跳转至下一个 RB 槽，槽中所有 RBUF 字节可以按照任何顺序读取。

如果 RB 已满，则收到的下一个帧将被临时储存直到它有效接收（第 6 个 EOF 位），之后，当 ROM=0 时，则最早接收的帧会被最新接收帧所覆盖；当 ROM=1 时，则最新接收帧被丢弃，在这两种情况下，则将置起 ROIF 标志。如果在新接收帧生效之前，主机控制器读取了最早接收的帧并置起 RREL，那么就不会丢失任何帧。

22.3.3 帧接收处理

如果没有接收过滤器，则 CAN 控制器会指示接收每个帧，并且要求主机控制器来决定是否对该帧进行寻址，这就给主机控制器带来相当大的负载。

除了通过接收过滤器降低负载之外，还可以关闭中断。基本的操作就是，如果 CAN 控制器已经收到一个有效帧的情况下，RIF 设置为 1。为减少接收中断数量，可以使用 RAI/E/RAFIF(RB 几乎满中断)或者 RFIE/RFIF(RB 满中断)代替 RIE/RIF(接收中断)。“几乎满限制”通过 AFWL 进行设置。

RB 包含 6 个 RB 槽，读取 RB 的顺序如下：

- 通过 RBUF 寄存器读取 RB FIFO 中最早接收的帧。
- 通过设置 RREL=1 释放 RB 槽，该操作是用于选择下一个帧（即下一个 FIFO 槽）。RBUF 将自动更新。
- 重复这些操作直到 RSTAT 指示 RB 为空。

如果 RB FIFO 已满，并且新接收的帧被视为有效（第 6 个 EOF 位），那么会丢失一个帧数据（请参考 ROM 位），在此事件之前，不会丢失任何帧。因此，需要确保在 RB FIFO 已满且所选中断生成之后，主机控制器能够有足够的时间从 RB 读取至少一个帧。

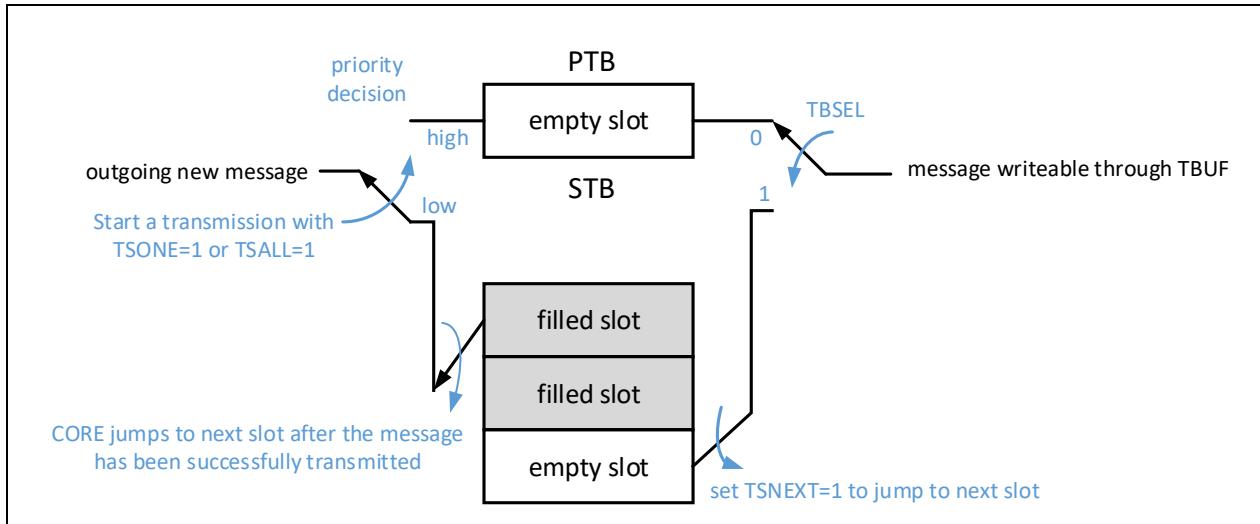
22.3.4 帧发送处理

在启动帧发送之前，必须确保至少其中有一个发送缓冲器（PTB 或者 STB）加载了帧数据，如下图所示，TSSTAT 指示 STB 填充状态（如果 TPE=1，则 PTB 写入被锁定）。TBUF 寄存器用于访问 PTB 和 STB，设置流程建议如下：

- 通过 TBSEL 选择 PTB 或者 STB
- 向 TBUF 寄存器写入帧数据（所有 TBUF 字节可按照任何顺序写入）
- 对于 STB 而言，设置 TSNEXT=1 表示完成 STB 槽的加载

- 对于 PTB 而言，可重复写入，来覆盖之前写入的数据，不可配置 TSNEXT=1

图 22-10 PTB 和 STB 在 FIFO 模式下的结构图



CAN2.0B 帧的最大有效载荷的长度为 8 字节，CANFD 为 64 字节，DLC 位用于定义每个帧的长度。对于 CAN2.0B 远程帧 (RMF 位)，DLC 是无效的，因为 classic CAN 远程帧的数据长度始终是 0 字节。当使用 PTB 时，应该通过设置 TPE 位启动传输。如果需要使用 STB，则必须设置 TSONE 启动单个帧，或者设置 TSALL 启动发送所有帧。

PTB 的优先级始终高于 STB，如果这两个发送缓冲器同时收到发送指令，则始终先发送 PTB 帧，而不受帧标识符的影响。如果 STB 发送已经被激活，那么该发送会继续完成，之后在下一个可发送位置（下一个帧间隙），开始 PTB 的发送。在完成和中止 PTB 发送之后，CAN 控制器回头处理 STB 中挂起未完成的帧。

在完成发送之后，会置起以下传输中断：

- 对于 PTB 来说：将置起 TPIF 标志
- 对于选择了 TSONE 的 STB 来说，如果已经传输完单个帧，将置起 TSIF 标志
- 对于选择了 TSALL 的 STB 来说，如果所有帧已传输完成，将置起 TSIF 标志。换言之，如果 STB 为空，将置起 TSIF 标志。因此，在启动 TSALL 发送之后，如果主机控制器又向 STB 写入一个帧，则该帧也会被发送，然后再置起 TSIF 标志。

22.3.5 中止帧传输

发送缓冲器中的帧因为优先级较低而无法发送时，会较长时间堵塞该缓冲器。为避免此种情况发生，主机控制器可以在传输还未启动之前，通过分别设置 TPA 和 TSA 撤销传输请求。

TPA 和 TSA 共用一个中断标志：AIF，CAN 协议机只有在未向 CAN 总线发送任何信息时才会执行中止命令（也就是说，当前正在进行中的发送不会被中止）。因此，应遵循以下规则：

- 总线仲裁期间不会中止发送。
- 如果节点失去仲裁，则会在稍后执行中止发送命令。
- 如果节点赢得仲裁，则继续执行帧发送。
- 帧发送期间不会中止发送。
- 如果一帧数据发送成功，则会向主机控制器指示成功发送的信号。在这种情况下，不会发出中止信号，同时产生相应的中断和状态信息。
- 当因为某种错误导致帧发送失败时，错误计数器会累加，并执行中止发送动作。
- 如果 STB 中至少还有一帧数据，同时主机已命令发送所有帧数据 (TSALL=1)，那么向主机指示发送完成和中止发送的信号都会被置起。

因为以上这些因素的影响，中止发送可能需要花费一些时间，时间的长短取决于 CAN 通讯速度和帧长度。因此，如果中止发送被执行，将会导致以下情况发生：

- TPA 释放 PTB，导致 TPE=0。在释放 PTB 之后，帧数据仍然存放在 PTB 内。

- TSA释放STB的单个帧槽或者所有帧槽，具体情况取决于选择了TSONE还是TSALL来启动帧发送。TSSTAT会进行相应的更新，释放STB帧将导致该帧被丢弃（原因是主机无法访问该帧）。

另外，不建议同时设置 TPA 和 TSA，如果主机控制器同时设置 TPA 和 TSA，将置起 AIF 标志，并且 PTB 和 STB 发送会被中止。已经开始发送的帧会在执行中止命令之前继续完成发送，之后会发出成功发送的信号。因此，以下中断标志可能会置起（在使能的情况下）：

- AIF (一旦中止PTB和STB传输，即会置起该标志)
 - TPIF+AIF
 - TSIF+AIF
 - TPIF+TSIF (很少见，只有在主机没有及时处理TPIF的情况下，才会发生)
 - TPIF+TSIF+AIF (很少见，只有当主机没有及时处理TPIF和TSIF的情况下，才会发生)

若需清除整个 STB，则需要同时置起 TSALL 和 TSA。主机可以使用 ALIF/ALIE 来检测某个帧是否因为失去仲裁而导致长时间没有发送。

22.3.6 STB满

在向 STB 写入帧之后，如果置起 TSNEXT=1，即表示缓冲器槽满，并跳转到下一个可用的帧槽。在此操作后，CAN 控制器自动将 TSNEXT 复位为 0。

如果最后一个帧槽已满，即所有帧槽都被占用，此时 TSNEXT 会保持置位状态直到有一个新的空闲槽出现，当 TSNEXT=1 时，CAN 控制器将禁止写 TBUF。

当一个槽空间空闲时，CAN 控制器自动复位 TSNEXT 为 0，如果 STB 内的一帧数据被成功发送，或者主机请求终止发送 (TSA=1)，或者发送已经达到 RETLIM 或 REALIM 设定的限值，那么一个槽空间变为空。如果 TSALL 发送被终止，那么 TSNEXT 被复位，另外整个 STB 被标记为空。

22.3.7 错误处理

一方面，CAN 控制器会自动执行错误处理，这就意味着在通常情况下，主机控制器无须关注错误问题，包括自动重新发送帧数据和自动侦测接收帧错误。另一方面，应主机请求，CAN 控制器可以选择性地提供关于错误的具体信息，并通过中断向主机发出错误信号，该特性可以实现主机运行类似 CAN 总线检测器的应用程序。

每个 CAN 节点有 3 种错误处理状态：

- 主动错误：节点在监测到错误时会自动发送主动错误帧
 - 被动错误：节点在监测到错误时发送被动错误帧，意味着节点不会向总线发送显性位，而是期望其他CAN节点发送错误帧。处于被动错误状态的节点可以发送帧数据，但是如果这些节点在之前的帧传输中已经处于发送状态，那么需要在帧间隔之后挂起8个位，才开始新的一帧数据的发送。
 - 总线关闭：如果出现太多错误，节点会进入“总线关闭”状态，不再影响总线。

为处理这 3 种错误状态，每个 CAN 节点配备了 2 个错误计数器，即发送错误计数器和接收错误计数器（通过 TECNT 和 RECNT 进行读取），这两个计数器会根据 CAN 协议进行递增和递减。一旦达到 CAN 协议定义的计数值后，节点会进入相应的错误状态，请参考表 22-4。

如果监测到错误发生，错误计数器会递增。由某个节点可能引发的严重错误将会使计数器增加 8，而由其他节点可能引发的错误将会使计数器增加 1。有效的帧发送和帧接收会使计数器值递减，所有这些均由 CAN 协议进行定义，并由 CAN 控制器自动处理。

错误帧（错误标志产生）与数据帧不相同，错误帧代表的是至少 6 个连续显性位的显性脉冲，这违反了其他节点的位填充规则。如果一个 CAN 节点检测到此违规行为，该 CAN 节点会发送错误标志，该错误标志会产生错误帧，所有节点的错误帧会进行叠加。（因此，错误帧的持续时长为 6 到 12 个位时间）。主机

控制器不能命令发送错误帧，这些均由 CAN 控制器自动处理。

如果 CAN 控制器接收命令发送一帧数据，则 CAN 控制器将尽可能地快的尝试重传直到该帧发送成功，或者该节点进入“总线离线”状态。如果 CAN 控制器接收到一帧数据且监测到错误，则该接收帧将被丢弃。因为自动发送的错误帧，发送者将重新发送该帧数据，RBUF 中的帧不会被包含错误的帧所覆盖，只有有效的接收帧才会导致 RBUF 溢出。

重新发送的次数受 TECNT 所限制（如果节点进入“总线离线”状态），除此之外，还可以使用 RETLIM 和 REALIM 进行限制。

表 22-4 CAN 节点错误状态

错误状态	TECNT	RECNT	EPASS	BUSOFF
主动错误	两者均小于 128		0	0
被动错误	其中之一大于 128		1	0
离线	大于 255	-	1	1

22.3.8 总线离线

CTRLSTAT 寄存器的 BUSOFF 位用于指示“总线离线”状态，当 CAN 节点的发送错误计数器超过 255 时，CAN 节点将自动进入“总线离线”状态。此时，该节点将不会再进行通讯，直到其重新恢复到错误主动状态。BUSOFF 被软件配置为 1 时（总线从“总线离线”状态恢复），也能启动 EIF 中断。如果 CAN 节点由上电复位触发复位，或者当该节点接收到 128 次 11 个隐性位时（恢复序列），CAN 节点将恢复到错误主动状态。注意：在各个恢复序列之间，总线可能包含显性位。（注意：两个有效帧之间的最长时间足以被识别为恢复序列）。

在“总线离线”状态下，RECNT 和 TECNT 计数保持不变。需要注意的是，在进入“总线离线”状态时，TECNT 会翻转，因此可能会保持为较小的值。因此，建议在节点进入总线离线状态之前，使用 TECNT，在进入总线离线状态之后，使用 BUSOFF 标志。节点从“总线离线”状态恢复之后，RECNT 和 TECNT 自动设置为 0。

如果某个帧被挂起等待发送，但是该 CAN 节点已进入总线离线状态，那么该帧将继续挂起。当节点从“总线离线”状态恢复至错误主动状态，则该挂起的帧将会被重新发送。若不需要重新发送，那么应该通过主机控制器中止帧发送。

22.3.9 扩展状态和错误报告

CAN 总线通讯期间可能会发生错误，下述功能支持错误侦测和错误分析，可用于扩展总线监测。

22.3.9.1 可编程的错误警告界限

RECNT 和 TECNT 计数器负责对收发器期间发生的错误进行计数，寄存器 ERR 包含了一个可编程的错误警告界限 EWL，使主机控制器能够灵活处理错误事件。错误警告的界限值以 8 为步长，范围是 8 到 128。

$$\text{错误计数限制} = (\text{EWL}+1)*8$$

如果 EIE 使能情况下，在下列情况下将置起 EIF 中断：

- RECNT 或者 TECNT 越过错误警告界限，高于或者低于错误警告界限。
- BUSOFF 位硬件置 1 或者清除。

22.3.9.2 仲裁丢失捕获 (ALC)

仲裁丢失不属于错误，因此与 TECNT 和 RECNT 无关，但是也是不希望发生的。

CAN 控制器能够检测发生仲裁丢失的仲裁位域中具体的位，同时将置起 ALIF 中断标志，用于指示发生仲裁丢失事件。在下一个仲裁丢失事件发生前，ALC 的值会保持不变。

ALC 值的定义：帧从 SOF 位开始，然后发送 ID 的第一位，第一个 ID 位（即 ID(28)）的 ALC 值为 0，第

二个 ID 位的 ALC 值为 1，以此类推，请参考图 22-2 和图 22-3 了解关于各类 CAN 帧的位顺序。

注意事项：

- 只允许在仲裁场进行仲裁，因此，ALC 的最大值为 31，该位为扩展帧的RTR/RRS位。
- 如果标准远程帧与扩展帧发生仲裁，则扩展帧在IDE位将丢失仲裁，ALC的值为12。发送标准远程帧的节点不会指示已经发生仲裁，原因是该节点已赢得仲裁。
- 仲裁场外不可能出现仲裁丢失的情况，因为这种情况将被视为位错误。

22.3.9.3 错误类型 (KOER)

CAN 控制器能够识别 CAN 总线上的错误，并将最近一次错误事件存储在 KOER 位，同时监测到 CAN 总线错误时会置起 BEIF 标志。新的错误事件会覆盖 KOER 之前存放的值，因此，主机控制器必须快速响应错误事件。

每次出现新的错误事件时 KOER 会随之更新。因此在成功接收或者发送帧时，KOER 保持不变，这为延迟错误调查提供了机会。当 RBALL=1 时，错误帧也会存放在 RB 内，另外 LLC 帧也包含了 KOER 信息。

当 RBALL=1 时，出错的数据帧也会存放在 RB 内。另外，LLC 帧也包含了 KOER 信息。

如果 ROP=1 错误事件将导致协议例外。出现这类错误时，BEIF 中断也会置起，同时更新 KOER，与之相对的是，向上兼容协议例外不会导致 BEIF 中断，也不会更新 KOER。

22.3.9.4 接收所有数据帧 (RBALL)

当 RBALL=1 时，所有接收帧（包含出错的数据帧）都会存储在接收缓冲器 RB 内，在环回模式下也是如此，并且还会禁用接收过滤器，接收缓冲器 RB 只存储数据帧，错误帧或者过载帧不存储。

如果 CiA 603 时间戳被使能 (TSEN=1) 且时间戳位配置为 EOF 位 (TSPOS=1) 时，那么在发生错误的情况下，在错误帧开始处获取时间戳。

大多数错误只发生在节点作为发送器时，在这种情况下，如果启动了环回模式，则数据帧只存储在 RBUF 中。根据错误类型，存储在 RBUF 槽内的帧数据可能部分是有效的，而其它部分是未知的。如下表列出了各种可能的情况：

表 22-5 RBALL 和 KOER

KOER	节点状态	描述
无错误	所有	成功接收
位错误	接收器	只发生在应答场，所有存储的数据均有效
	发送器	负载数据始终无效，包含 ID 的帧头信息可能有效。在仲裁场，检测到错误的位属于仲裁过程，所以不属于位错误。但是如果在仲裁场的填充位检测到错误，则归属于位错误，在此情况下，帧头信息无效，但如果该帧头信息与预期的帧头信息相匹配，则需要使用该帧头信息做进一步判断。
格式错误	所有	只包含数据帧中的格式错误，包括 CRC 界定符、ACK 界定符、以及 EOF 位，所有存储的数据均有效。
填充错误	接收器	填充错误的位置未知，所有存储的数据均有效。
	发送器	仅发生在仲裁场，所有存储的数据均有效。
应答错误	接收器	仅发生在节点处于 LOM 模式时，所有存储的数据均有效。
	发送器	仅发生在不含 self-ACK 的环回模式，所有存储的数据均有效。
CRC 错误	接收器	所有存储的数据均有效。

22.3.9.5 传输状态 (TSTAT_1 和 TSTAT_2)

CAN 控制器包含各种状态和中断位，这些状态和中断位用于跟踪帧的传输状态，例如 TPIF、ALIF、AIF 以及 BEIF 等，除此之外，发送状态寄存器 TSTAT_1 和 TSTAT_2 也提供了详细的相关信息。

TSTAT_1 记录的是当前发送帧的状态信息，所以，TSTAT_1 中的内容是意失性的。一旦一帧数据成功完成发送，或者因重新仲裁、达到重新发送的限值而停止发送（详见 RETLIM 和 REALIM 寄存器定义），那么这些信息会存放在 TSTAT_2 内，换言之，当一帧数据的传输达到其最终状态时，TSTAT_2 会相应更

新。

发送状态包含以下信息：

- 用于标识帧的句柄
- 句柄所指向帧的发送状态
- 句柄所指向的相关时间戳

TTS 寄存器提供了时间戳信息，只有当 LLC 帧中的 TTSEN=1 时，才会更新时间戳（见表 22-1），TTS 寄存器只与 TSTAT_2 有关。

TSTAT_1 和 TSTAT_2 根据表 22-6 所述帧传输状态进行编码。

HANDLE 用于标识帧，主机应用程序选择标识符，其包含在 LLC 帧内（表 22-2），建议使用软件计数器的计数值作为 HANDLE，该 HANDLE 会在每发送一帧新的数据时递增，计数值可能发生翻转。

表 22-6 TSTAT 状态编码

值	标签	描述
000	IDLE	没有正在进行的传输
001	ONGOING	发送正在进行，且无错误
010	LOST_ARBITRATION	仲裁丢失，根据 REALIM 配置重新进行仲裁
011	TRANSMITTED	成功完成发送
100	ABORTED	发送中止（TPA、TSA）
101	DISTURBED	发送错误，根据 RETLIM 配置重新进行发送
110	REJECTED	在 LLC 帧中，帧格式配置错误
111	-	保留

存在一些可能的情况会导致 TSTAT_1 和 TSTAT_2 两者的更新，表 22-7 介绍了这些情况的详细信息。

表 22-7 TSTAT 状态事件

事件	TSTAT_1	TSTAT_2
电源上电复位或 RESET 有效	IDLE	IDLE
开始发送（CAN 协议机从 TBUF 中预取数据）	ONGOING	-unchanged-
成功完成一次发送	IDLE	TRANSMITTED
发送中止（TPA 或 TSA）	IDLE	ABORTED
仲裁丢失，没有达到 REALIM 配置的重新仲裁限制次数	LOST_ARBITRATION	unchanged
仲裁丢失，达到 REALIM 配置的重新仲裁限制次数	IDLE	LOST_ARBITRATION
发送错误，没有达到 RETLIM 配置的重新发送限制次数	DISTURBED	-unchanged-
发送错误，达到 RETLIM 配置的重新发送限制次数	IDLE	DISTURBED
总线离线恢复	ONGOING	-unchanged-
在发送的起始检测到 LLC 帧的错误配置	IDLE	REJECTED

除了发送中止事件，其它导致 TSTAT_1 或 TSTAT_2 更新的事件都是一步一步进行的，所以主机应用程序可以跟踪其状态。与之相对的是，发送中止命令则是随时可以执行的。除此之外，一些较复杂的发送中止情况是无法通过 TSTAT_1 和 TSTAT_2 进行正确指示标识的。

- 如果 STB 内不止一帧数据，且选择了 TSALL 传输模式，同时执行了 TSA 指令，那么指令发送会被中止，而 TSTAT_2 却只能反映一个帧数据的状态。
- 如果同时是能了 PTB 和 STB，当前正在进行 PTB 的发送和 STB 发送中止，那么可能会同时发生 PTB 发送成功事件和 STB 发送中止事件，这会导致连续两次更改 TSTAT_2 的内容，导

- 致主机应用程序无法跟踪。
- 如果使用的是 STB，且选择 TSALL 传输模式，TSA 在前一帧已经发送成功之后，而下一帧还没启动之前短暂置起，这会导致连续两次更改 TSTAT_2 的内容，导致主机应用程序无法跟踪。

因此，只有在下列条件都满足时，TSTAT_2 才会置起 ABORTED 状态：

- 在 PTB 传输时置位 TPA，或者在 STB 传输时置位 TSA。
- TSTAT_1 标识当前正在进行的发送处于 ONGOING、LOST_ARBITRATION 或者 DISTURBED 状态。

22.3.9.6 帧拒绝发送

如果一帧数据被指定发送，同时该帧数据的帧格式配置错误，则这帧数据将被拒绝，不会予以发送。帧拒绝将会导致 TPIF 或 TSIF 中断（如果相应的中断使能位已置起的话，具体时 TPIF 还是 TSIF 取决于所使用的时 PTB 还是 STB），并根据 22.3.9.5 章节所述，标识传输状态位“REJECTED”。帧格式配置错误的原因在于上层应用出现严重错误，在正常操作中不应该发生，只会发生在应用开发阶段出现错误时。

22.3.10 比特率切换和收发模式转换

经典 CAN 帧是以一个恒定的比特率（“慢速”标称比特率）进行传输，当 BRS=1 时，CANFD 帧可以切换帧内的比特率，这些帧以“慢速”标称比特率开始传输，然后在 BRS 位采样点切换到“快速”比特率，在 CRC 界定场采样点或者出现错误时返回到原来的“慢速”标称比特率。

要实现更高的比特率取决于总线拓扑结构和收发器的电器特性，对于 CANFD 帧而言，对应的收发器具有改善信号的能力（CANFD SIC 收发器），因此能够达到更高的比特率。

经典 CAN 和 CANFD 收发器采用牵引电阻的概念，逻辑 1 是由牵引电阻引起的，因此称为“隐性”，而逻辑 0 是节点主动驱动的，因此称作“显性”，使用牵引电阻会导致总线的不对称，从而限制最高比特率。

22.3.11 扩展特性

22.3.11.1 限制重传和重新仲裁

因为反复尝试重新传输或重新仲裁会导致严重延迟，增加总线负载，所以有些时候并不需要自动开启重新传输或重新仲裁。通过 RETLIM 和 REALIM 可以限制这两种行为。这两种限制行为是互相独立的。

重新传输发生在错误之后，而重新仲裁发生在仲裁丢失之后。如果节点进入总线离线状态，则会停止重新传输。

在发生相应事件后，重新传输和重新仲裁的计数器都将递增，并且在开启一次新的传输时（配置 TPE、TSONE 或者 TSALL）都将复位至 0，或者配置 TSALL 开启传输几帧数据并且其中任意一帧数据发送成功，或帧数据没有发送成功但重新发送/重新仲裁达到限定次数时，重新传输和重新仲裁的计数器都将复位至 0。选择 PTB 或者 STB 作为传输数据的来源不会对重新传输和重新仲裁的计数器产生影响，只有错误事件、仲裁丢失或者成功发送才会影响到这两个计数器。下面举例逐步解释这两个计数器的行为逻辑：

- 出现错误导致 STB 传输失败 → 重新传输计数器递增
- STB 传输丢失仲裁 → 重新仲裁计数器递增
- PTB 成功传输 → 两个计数器均复位（STB 传输时两个计数器从 0 开始递增）

如果立即实现了成功传输，这与正常情况没有区别。但是如果传输不成功，则会发生下列行为：

- TPIF 标志会置起，相应的传输缓冲槽会被清除。
- 在报错的情况下，KOER 和 错误计数器会更新，BEIF 标志会置志，并且其他的错误中断标志也会相应的置位。

因此，如果重新传输或重新仲裁受到限制，那么单纯 TPIF 标志无法指示该帧是否已经发送成功。因此，如果需要收到传输成功的反馈，则重新传输或者重新仲裁限制机制需要配合 BEIF 和 ALIF 一起使用。通过 TSTAT_1 和 TSTAT_2（查看 22.3.9.5）也可以帮助跟踪事件状态。

如果重新传输或重新仲裁受到限制, TSALL=1 置起, 并且 STB 内包含不止一帧数据, 则计数器会记录每帧数据的尝试次数。在达到限制次数之前, 如果有任何一帧数据没有成功传输, 则 CAN 控制器会跳到下一帧数据, 并且在 STB 变空之后, CAN 控制器停止发送。在此情况下, 用户就很难判断发生了什么状况, 只有错误计数器 TECNT 会指示当前状况。如果两个帧中有一个帧出错, 主机就没法侦测到底哪一个帧被成功传输了, 因此这种情况会导致评估变得很复杂。

限制重新传输和重新仲裁能够保证在时间间隙内进行操作以确保最大限度的报文延迟。

22.3.11.2 只听模式 (LOM)

LOM 能够在不影响总线的情况下监测 CAN 总线。

- LOM 与 ISO 11898-1:2015 定义的总线监控功能类似, 但并不完全兼容。不同点在于 ACK, LOM 模式下, CAN 控制器依赖于总线其它的 CAN 节点发出的显性 ACK 信号。
- LOM 与 ISO 11898-1:2015 定义的受限操作模式类似, 但并不完全相同, 受限操作参考 22.3.11.3 章节。

LOM 模式下可以通过 KOER 和 BEIF 标志检测总线错误。

在 LOM 模式下, CAN 控制器不能向总线写入显性位 (即无法发出主动错误标志、过载帧以及无法产生有效应答 ACK), 该行为参考如下规则:

- 如果 LOM=1, 则协议机类似处于受限操作模式 (Restricted Operation Mode), 即每个错误会导致协议例外事件 (Protocol Exception Event)。
- 如果 LOM=1, 则协议机不会产生显性 ACK.
- 错误计数器不会产生任何变化。

在 LOM 模式下, 有关 ACK 的相关信息:

- 如果节点发送了一帧数据, 那么只有当总线上连接了不止一个额外的节点, 并且在额外节点生成 ACK 的情况下, 才会在总线上生成可见的 ACK, 之后, 假如没有发生错误, 所有节点会接收该帧, 否则, 该帧数据因 ACK 错误无效。
- 如果发生 ACK 错误, 则处于 LOM 模式的节点能够检测到该错误。

注: 在传输正在进行时, 不能开启 LOM 模式, 如果此时开启 LOM 模式, 则不会启动传输。

环回模式 (外部 LBME) 对于 LOM 模式的行为起到非常重要的作用, 如果禁用了 LBME 模式, 则 LOM 模式的行为如上所述, 节点不能向总线写入任何显性位。如果使能 LBME 模式, 则允许节点发送包含自应答 (self-ACK) 信号的帧, 但节点不会通过 ACK 来响应从其他节点发送的帧, 也不会发出主动错误标志和过载帧。总而言之, LOM 和 LBME 结合起来使用相当于“一个静默接收器, 在特殊情况下具有发送功能”。

22.3.11.3 受限操作 (ROP)

在受限操作模式下, 除了潜在的时间主机, CAN 节点无法发送帧数据, 作为发送主机的节点可以发送时间参考报文以启动网络。

在受限操作模式下, CAN 节点能够接收帧数据, 并在接收到有效帧数据时产生 ACK 响应, 一旦发生错误或者总线过载情况, 处于受限操作模式下 (ROP=1) 的 CAN 节点会将这些视为协议异常, 并进入总线集成状态 (Bus Intergration), 在 ROP=1 的情况下, 错误计数器保持不变。

注: 当 CAN 总线正在进行传输时, 无法激活 ROP。

22.3.11.4 环回模式 (LBMI 和 LBME)

因 CAN 控制器支持两种环回模式: 内部环回模式 (LBMI) 和外部环回模式 (LBME), 两种模式均可实现接收自己发送的帧数据, 这对于自检测非常有用, 具体参考图 22-11.

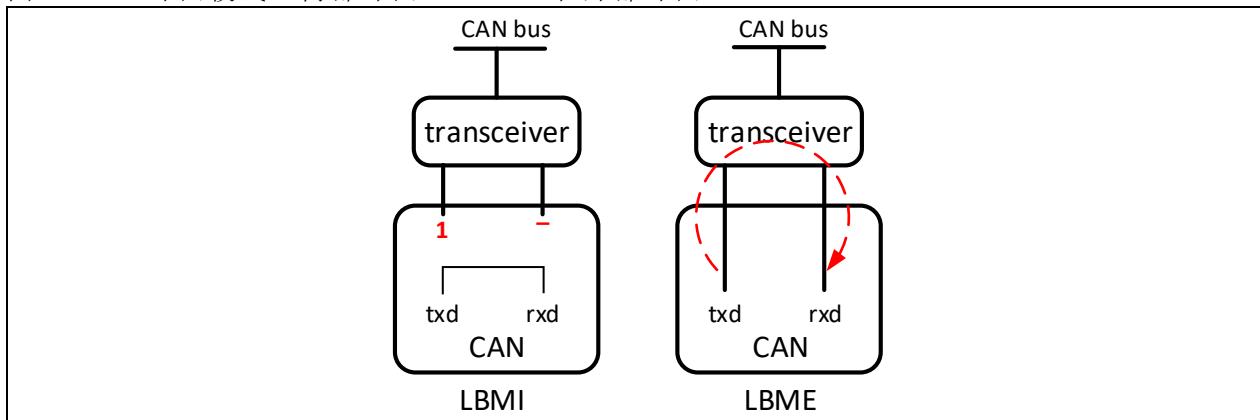
在 LBMI 模式下, CAN 控制器与 CAN 总线断开连接, <txd> 输出隐性电平, 输出数据通过内部反馈给输入端, 在 LBMI 模式中, 节点会产生 self-ACK 以避免产生 ACK 错误。

在 LBME 模式下, CAN 控制器始终与收发器相连, 在总线上可以看到节点发送的帧数据, CAN 控制器通过收发器接收自己发送的帧数据。在 LBME 模式下, 当 SACK=0 时, 节点不会产生 self-ACK; 当 SACK=1

时，节点产生 self-ACK。因此，在 LBME 模式且 SACK=0 时，发送帧数据时可能会出现两种结果：

- 其他节点接收帧数据并产生 ACK，从而成功完成发送与接收。
- 没有其他节点与总线相连，导致ACK错误，因此建议 RETLIM=0，以避免重新传输而引起错误计数器递增。

图 22-11 环回模式：内部环回（LBMI）和外部环回（LBME）



环回模式下，CAN 控制器接收自己发送的帧数据，并将其放在 RBUF 中，置起相应的接收中断标志，当接收到自己发送的帧数据时会置起 LBF。

LBMI 模式可以用于芯片内部测试和软件测试，LBME 可以用于收发器测试，以及节点与收发器的连接测试。

注：LBMI 和 LBME 不可与 TPE、TSONE 或 TSALL 同时更改。LBME 可以与 LOM 模式配合使用。

22.3.11.5 收发器待机模式

通过 STBY 配置位可以驱动<STB>信号，用于激活收发器的待机模式，这种操作模式与 NXP TJA1049 收发器及其他具有类似功能的收发器是兼容的。

一旦激活待机模式，就不能再进行传输，也不能设置 TPE、TSONE 和 TSALL 位，另外 CAN 控制器不允许在发送过程中（TPE、TSONE 或者 TSALL 置位时）设置 STBY 模式。

一旦 STBY 置起，收发器进入低功耗模式。在此模式下，无法以全速接收帧，但是可以监测 CAN 总线的显性状态。如果显性状态在规定时间（即收发器数据特性表定义的时间内）内保持生效状态，那么收发器将拉低<rxd>信号。如果<rxd>变低，则 CAN 控制器将自动清零 STBY，从而关闭收发器的待机模式，并且不会干扰总线上其他主机（退出待机模式后，将收到来自其他节点发出的帧数据）。

收发器需要花一些时间从待机模式切换到活跃模式，所以无法成功接收到初始唤醒帧。因此，当前处于待机模式下的节点不会进行 ACK 响应，如果总线上没有 CAN 节点对唤醒帧进行 ACK 响应，则会导致唤醒帧发送节点产生 ACK 错误，然后该发送节点将自动重新发送该帧。在重新发送时，收发器将回归到活跃模式，CAN 控制器将接收到帧，并以 ACK 进行响应。

22.3.11.6 错误计数器复位

根据 CAN 规范，RECNT 计数接收错误，TECNT 计数发送错误。在发生太多发送错误之后，CAN 节点进入总线离线状态，寄存器 RESET 位不会改变错误计数器或者总线离线状态。CAN 规范制定了从总线离线状态恢复和错误计数器递减的规则，如果只是暂时的错误导致了问题，一个好的节点会自动的从总线离线状态恢复。为避免低级错误，Classic CAN2.0B 规范要求硬件具备上述自动行为，而无需主机控制器的交互操作。

CANFD 规范放宽了上述限制，该规范允许主机控制器对错误计数器的手动控制，但是这种使用方式需要格外小心，建议仅在调试过程中使用。

配置 BUSOFF 位为 1 将复位错误计数器，从而强制节点从离线状态恢复，同时置位 EIF 标志。

22.3.11.7 软件复位

通过将 CTRLSTAT 的 RESET 位置 1，可以启动软件复位。当 RESET=1 时，部分寄存器会进入复位状态，其他寄存器则只适用于 IP 的复位，但无论是软件复位还是 IP 的复位，所有的复位值始终一致。

表 22-8 软件复位

寄存器	是否复位	注释
ACFADR	否	-

ACFC	否	当 RESET=1 时，寄存器可写，否则写锁定。
ACFM	否	当 RESET=1 时，寄存器可写，否则写锁定。
AC_SEG_1	否	当 RESET=1 时，寄存器可写，否则写锁定。
AC_SEG_2	否	当 RESET=1 时，寄存器可写，否则写锁定。
AC_SJW	否	当 RESET=1 时，寄存器可写，否则写锁定。
AE_x	否	-
AFWL	否	-
AIF	是	-
ALC	是	-
ALIF	是	-
BEIF	是	-
BUSOFF	(否)	通过设置 BUSOFF=1 复位错误计数器也会复位 BUSOFF 位
EIF	否	-
EPASS	否	-
EPIF	是	-
EWARN	否	-
EWL	是	-
FD_ISO	否	当 RESET=1 时，寄存器可写，否则写锁定。
FD_SEG_1	否	当 RESET=1 时，寄存器可写，否则写锁定。
FD_SEG_2	否	当 RESET=1 时，寄存器可写，否则写锁定。
FD_SWJ	否	当 RESET=1 时，寄存器可写，否则写锁定。
FD_SSPOFF	否	当 RESET=1 时，寄存器可写，否则写锁定。
KOER	是	-
LBME	是	-
LBMI	是	-
LLCAOT	否	-
LLCFORMAT	否	-
LLCPBYTES	否	-
LOM	否	-
PRESC	否	当 RESET=1 时，寄存器可写，否则写锁定。
RAFIF	是	-
RBALL	是	-
RBUF	(是)	所有 RB 槽空间被标记为空，RBUF 存储值未知。
REALIM	是	-
RECNT	否	通过设置 BUSOFF=1 可以复位错误计数器。
REF_ID	否	-
REF_IDE	否	-
RETLIM	是	-
RFIF	是	-
RIF	是	-

ROIF	是	-
ROM	否	-
ROP	否	-
ROV	是	所有 RB 槽空间被标记为空。
RREL	是	-
RSTAT	是	-
SACK	是	-
STBY	否	-
TBE	是	-
TBF	是	-
TBPTR	否	-
TBSEL	是	TBUF 固定指向 PTB
TBUF	(是)	所有 STB 槽空间被标记为空, TBUF 指向 PTB。
TECNT	否	通过设置 BUSOFF 可以复位错误计数器。
TEIF	是	-
TEW	否	-
TPA	是	-
TPE	是	-
TSA	是	-
TSALL	是	-
TSMODE	否	-
TSEN	否	-
TSPOS	否	-
TSNEXT	是	-
TSONE	是	-
TPIF	是	-
TSFF	是	所有 STB 槽空间被标记为空。
TSIF	是	-
TSSTAT	是	所有 STB 槽空间被标记为空。
TTEN	是	-
TTIF	是	-
TTPTR	否	-
TTS	否	-
TTTBM	否	-
TTYPE	否	-
TT_TRIG	否	-
TT_WTRIG	否	-
T_TPRES	否	-
WTIF	是	-

22.4 时间触发CAN (TTCAN)

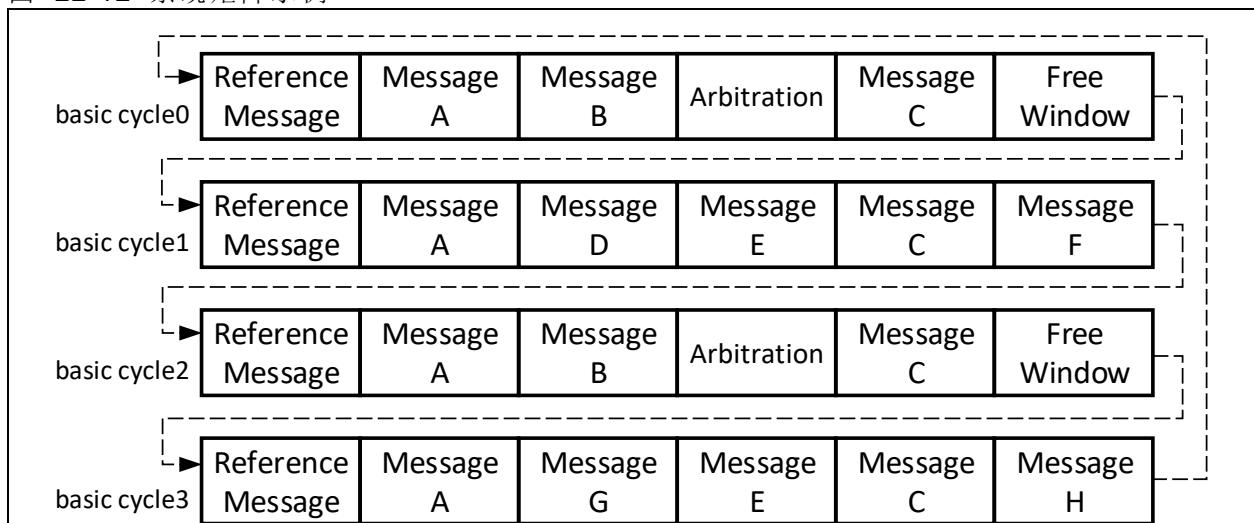
22.4.1 介绍

时间触发 CAN 是基于 ISO11898-4 协议的一种操作模式，所有帧只能在预先设定的时间窗内传输，总共有三种时间窗：

- 专用时间窗（只允许一个节点传输特定 ID 的帧）
- 自由时间窗（保留用作网络系统扩展）
- 仲裁时间窗（多个节点同时传输一帧数据，此时引发仲裁）

时间由 TTCAN 系统管理者离线设置，并按照图 22-12 所示的系统矩阵进行分配，横轴表示的是一个基本周期，开始于参考报文，参考报文由时间主机发送，其它报文由包括时间主机的任意节点发送。

图 22-12 系统矩阵示例



报文的 SOF 时间成为 Sync_Mark，参考报文的 Sync_Mark 称为 Ref_Mark，计时通过一个自由运行的 16 位计时器完成，计时器与 Ref_Mark 之间的时间差称为周期时间。换言之，每个基本周期是从 Ref_Mark 开始计算周期时间，周期时间作为时间戳存储在 RBUF 中。

TTCAN 系统管理者负责定义每个时间窗的长度，时间窗的长度设置要确保能够完整的传输一帧报文，因此，报文必须在单次传输模式下传输，存在一种例外情况：如果多个仲裁时间窗合并，那么可以使能重新传输，但是为了避免影响后续时间窗，需要尽早结束重传。

单次传输的含义：禁用重新传输和重新仲裁功能。

触发器在激活后会通过中断告知主机，然后主机需要为下一个时间窗配置一个触发器，这就要求主机进行实时响应。一个时间窗的时长指的是允许使用主机的时间。

如果需要将 CAN 控制器作为时间主机运行，那么参考报文需要放在 TBUF 槽（像其他的报文一样），然后在硬件触发器激活后进行传输。通过正确设置 REF_MSG_x 寄存器中参考报文的 REF_ID 和 REF_IDE 位，时间主机及所有时间从机将自动检测参考报文。一旦检测到参考报文，CAN 控制器将自动更新 Ref_Mark，启动周期时间。

除了用于报文传输的触发器之外，CAN 控制器还提供了窗口触发器。该窗口触发器用于检查距离上一次参考报文的时间是否过长。主机应该为周期性操作或者事件触发操作配置窗口触发器，在周期性操作中，每一个基本周期之后紧跟着下一个基本周期；在事件触发操作中，两个基本周期之间存在时间间隙，下一个周期开始于事件生成。如果窗口触发器使能，则会窗口产生中断。

除了支持 ISO 11898-4 之外，CAN 控制器还支持事件触发 CAN 通信与接收报文时间戳两种功能配合使用。通过设置寄存器位 TTTBM=0，可以使能该模式。在该模式下，CAN 控制器就类似于事件触发 CAN 通信，只是可以检测到参考报文且提供接收帧的时间戳。此外，该模式仅支持事件触发器和窗口触发

22.4.2 TTCAN模式下TBUF配置

事件 TBUF 的行为取决于寄存器 TTTBM 位，如果 TTTBM=1，那么每个 TBUF 槽空间均可由主机控制器寻址，该条件是使用传输触发的必备要求；如果 TTTBM=0，则仅支持时间戳和事件触发。

22.4.2.1 当TTTBM=1时TBUF的行为

在 TTCAN 模式下当 TTTBM=1 时, STB 用作报文槽的队列, 每个报文槽可由 TBPTR 寻址。主机通过 TBF 和 TBE 这两个位将报文槽标记为“满”或“空”, 已“满”的报文槽是写保护的。在 TTCAN 模式中, TBSEL 和 TSNEXT 两个位没有实际意义而被忽略。

通过设置 TBPTR=0 可以寻址 PTB, 这使得 PTB 可以作为任何一个 STB 槽使用。实际上, PTB 在 TTCAN 模式中没有特殊性, 它与 STB 相关联, 因此, 发送成功始终是 TSIF 位被标记。

在 TTCAN 模式中, TBUF 既没有 FIFO 模式也没有优先级决定模式。此外, 只能选择一帧一帧的数据进行发送。

报文触发定义的是何时需要发送(即时间窗的开始)一帧报文, 并通过指针 TTPTP 选择要发送的报文。如果触发事件发生, 那么选择的报文随即开始发送。最后, 触发中断被置位以便通知主机为下一步行动作准备。

所有的传输只能通过一次触发来启动, 在 TTCAN 模式中, TPE, TSONE, TSALL 和 TPA 固定为 0, 并被忽略。

22.4.2.2 当TTTBM=0时TBUF的行为

设置 TTTBM=0 提供了事件触发 CAN 通信和接收报文时间戳的一种组合, 在此模式下, PTB 和 STB 的行为就类似于 TTEN=0 时的情况。PTB 的优先级始终高于 STB, 而 STB 可以运行于 FIFO 模式(即 TSMODE=0)或优先级决定模式(即 TSMODE=1)。

22.4.3 TTCAN操作

上电之后, 时间主机需按照 ISO 11898-4 的要求进行初始化, 一个 CAN 网络中可能最多有 8 个时间主机, 每个时间主机有其自身的参考报文 ID(即 ID 的最后 3 位), 潜在的时间主机根据其优先级顺序发送参考报文, 优先级较低的时间主机会稍晚发送参考报文。

如果 TTEN 位已置起, 则 16 位定时器开始运行, 如果检测到参考报文, 或时间主机成功地发送了参考报文, 则 CAN 控制器将会复制该报文的 Sync_Mark 至 Ref_Mark, 同时将周期时间置 0。接收到报文后, 会置位 RIF, 成功发送报文会置位 TPIF 或 TSIF, 之后主机需要为下一次的动作配置触发。

触发动作可以是一次接收触发, 仅仅产生触发中断, 该触发动作可以用于检测是否收到所需要的报文。这种触发器也可以用于其它应用, 但具体情况取决于主机应用程序。

另一种触发器是发送触发器, 此触发器通过 TTPTP 指向 TBUF 槽, 启动 TBUF 槽内的帧发送。如果 TBUF 槽标记为空, 则不会启动帧传输, 但是会置起中断标志。

如果主机应用程序需要, 主机任务可以通过新报文来更新报文槽, 比如, 一个新的传感器值。随后, 如果 TTCAN 需要传输这则报文, 则由触发器进行激活。换言之, 一个主机任务负责 TTCAN, 另一个主机任务负责更新报文槽。因此这就要求一个报文槽专用于一帧报文(比如, 槽空间 1 专用于温度传感器)。如果没有足够的 TBUF 槽空间可用, 那么不同的报文可以共用槽空间。

TTCAN 主机应用程序需要跟踪系统矩阵, 如果触发器被激活, 那么主机应用程序需要准备下一次的传输。同时主机还需要处理基本周期, 参考报文包括了周期数。

注: ISO 11898-4 协议中的大多数操作都要求单次传输。

22.4.4 TTCAN时序

CAN 控制器支持 ISO 11898-4 等级 1, 内置一个 16 位定时器, 该定时器以 CAN 位时间(由 PRESC, AC_SEG_1, AC_SEG_2 定义)为基本时间单元, 另外 T-PRESC 定义了一个额外预分频器, 当 TTEN=1 时, 该定时器持续计数。

在报文的 SOF 位置, 该定时器的值为 Sync_Mark。如果该报文是参考报文, 则定时器值的 Sync_Mark 值复制到 Ref_mark。周期时间就等于定时器值减去 Ref_mark, 这个时间用于接收报文的时间戳, 或者用于待发送报文的触发时间。该定时器内置溢出保护, 因此周期时间在一个基本周期内, 是持续单向递增的。ISO 11898-4 不支持 CANFD 帧波特率切换功能, 因此 CAN 控制器总是按照较慢的正常位速率运行定时器。该定时器是自由运行的, 不受同步操作或者波特率切换的影响, 所以定时器滴答与 CAN 位的开始不同步。

定时器运行于 CAN 时钟域, 而所有的控制位和状态位均位于主机时钟域, 因为读取定时器值需要经由跨时钟域同步, 所以主机无法读取到绝对的定时器值, 需要通过触发事件将主机的动作与该定时器进行同步。

由触发事件, 接收和发送引发的中断均需要经由时钟域同步, 因此会有几个时钟延迟, 但是这种情况仅针

对于当主机应用程序在触发后决定是否启动发送时才有相关意义（因此建议使用传输触发）。在其他情形下，主机应用程序有足够的时间准备下一个触发事件（即下一个时间窗）。基本上所有的主机运行都足够快，以确保在 CAN 帧持续期间（即时间窗内）内完成多项任务。

ISO 11898-4 级别 1 的时序也并非完美，因为 CAN 位不能基于 CAN 同步进行缩短或延长，所以周期时间计数值不能与 CAN 位保持同步计数，一个节点的周期时间可能与其他节点的周期时间不同。通常情况下，这个差异是 $+/-1$ 滴答，但也不限于此差值，在发送参考报文时，在新的基本周期的开始，所有的节点会同步。

发送触发器被激活后，适当的发送只能在下一个 CAN 位启动，所以一个帧的 SOF 传输的最早时间点是 TT_TRIG+1。

22.4.5 TTCAN 触发类型

触发类型由 TTTYPE 位配置，TTPTP 指针指向 TB 报文槽，TT_TRIG 定义触发的周期时间。

但是所有触发以及相关的操作必须在达到最大周期时间值“0xFFFF”之前完成，因为这个值定义的是一个基本周期的最大长度值。除了立即触发，所有的触发都会置起 TTIF。

当设置 TTTBM=0，表示仅支持时间触发，在此模式下启动其他触发均会置起 TEIF。

当写 TT_TRIG 位 (TTTRIG[15:0]) 时，触发器被激活，此时 TTCFG[15:0] 的写操作将被锁定，除非触发时间已到达（如果 TTIE 有使能的话，会置起 TTIF）或者检测到错误（TEIF 置起），才能解锁，所以新的触发操作不会覆盖当前正在进行的触发。通过设置 TTEN=0，也可以解锁写访问。

22.4.5.1 立即触发

立即触发立即发送由 TTPTP 所指向的帧，而且不置起 TTIF，如果需要启动立即触发，则对 TTTRIG[15:8] 执行写操作，写入 TTTRIG 寄存器的值对立即触发没有影响。

在 TTCAN 模式中，TPE、TSONE 以及 TSALL 不可用，取而代之的是立即触发，帧数据的发送只能通过立即触发进行启动，在第一帧数据发送完成之前（发送成功或发送失败），主机不可产生第二个立即触发指令。

对于立即触发而言，可以通过 RETLIM 和 REALIM 寄存器限制重新仲裁和重新发送，TSA 可以用来中止发送（使用 TPA 没有意义）。

如果立即触发的 TTPTP 指针指向空槽，则 TEIF 会置起。

22.4.5.2 时间触发

时间触发只是通过设置 TTIF 来生成事件并产生中断，不会有其他动作。

时间触发可以用作接收触发。如果节点期望在一个时间窗内接收到一帧报文，那么如果该报文丢失且 RIF 未置起，则可以使用接收触发发出信号反映此异常。应该在预期能够成功接收报文的最后时刻之后设置接收触发。

如果 TT_TRIF 低于实际的周期时间，则会置起 TEIF 标志。

当 TTTBM=0 时，可以使用时间触发。这是此模式下唯一可用的触发类型。

22.4.5.3 单次发送触发

单次发送触发用于专用时间窗，在专用时间窗内，报文需要以单次发送模式进行发送，所选报文由 TTPTP 指定。单次发送模式是自动启动的，不受 RETLIM 和 REALIM 配置的影响，RETLIM 和 REALIM 寄存器被忽略且保持不变。

单次发送触发用于专用时间窗，为此，ISO 11898-4 定义的一个发送使能窗口，该使能窗口最多为 16 个周期时间滴答，寄存器位 TEW(3:0)+1 用于定义滴答数。如果总线被其他帧占用，则无法启动帧发送。这种情形不应该发生在专用时间窗，但是发送使能窗口可以确保不会发生启动延迟，因为启动延迟会与下一个时间窗冲突。如果发送使能窗口关闭，帧无法启动，则会中止帧发送。帧的 TB 槽会因此被标记为空，并置起 AIF 标志。TB 槽内的帧数据保持不变（不会受影响），如果下次尝试传输该数据，则只需要将 TB 槽标记为满。

如果 TT_TRIG 低于实际周期时间，则 TEIF 标志置起，不会执行任何动作。

22.4.5.4 发送启动触发

传输启动触发是专门用于已合并的仲裁时间窗，在这个时间窗内，多个节点可以发送报文，并且会发生 CAN 仲裁，发送的报文由 TTPTR 指定，RETLIM 和 REALIM 寄存器决定是否进行重新仲裁或者重新发送。

如果所选的报文无法被发送（仲裁丢失，发生错误后的多次发送），那么可以使用发送停止触发来中止报文发送。

如果 TT_TRIG 低于实际周期时间，则 TEIF 标志将置起，不执行任何动作。

22.4.5.5 发送停止触发

发送停止触发专用于中止发送操作，即中止通过发送启动触发启动的发送操作（参考 22.4.5.4 章节）。如果发送中止，则帧数据被丢弃。AIF 标志会置起，并且帧数据的 TB 槽会标记为空。TB 槽中的帧数据不会受影响，所以只需要在下次尝试发送该笔数据时将槽标记为“满”。

发送停止触发类似于上述的单次发送触发所使用的中止发送使能窗口。如果发送中止触发指向空槽（意思是说，帧数据已传输），那么不会执行任何动作，也不会置起 TEIF 标志。

如果 TT_TRIG 低于实际周期时间，则 TEIF 标志将置起，会执行中止动作。

22.4.6 TTCAN窗口触发

窗口触发不同于 22.4.5 章节所讲的通用触发类型，它有一个专用的中断标志 WTIF，如果周期计数等于 TT_WTRIF 的设定值，则 WTIF 将置起。

窗口触发用于判断距离上一次有效参考报文的时间是否过长，参考报文可以在周期性循环中或事件之后进行接收，主机应用程序需要关注此动作，并且对窗口触发进行相应调整。

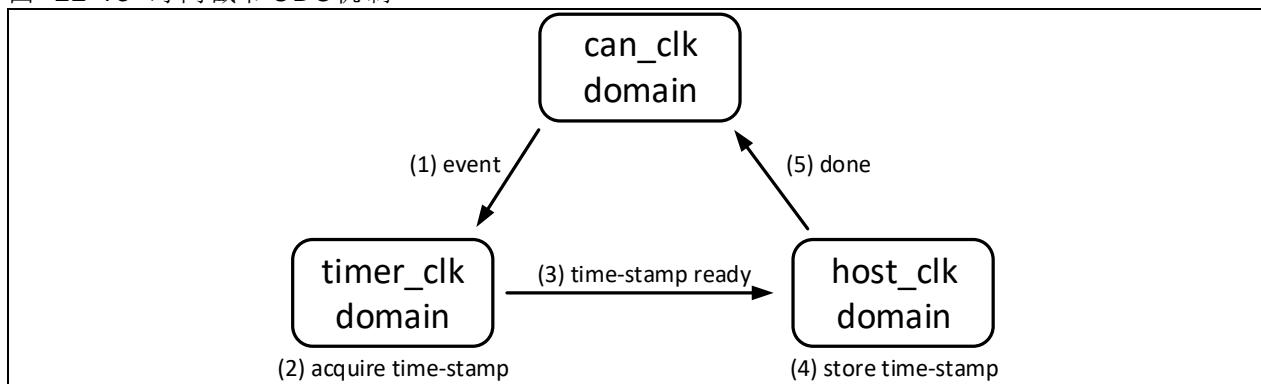
窗口触发默认值“0xFFFF”自动生成。

如果 TT_WTRIG 更新且低于实际周期时间，则将置起 TEIF 标志。

22.5 CiA 603时间戳

CiA 603 协议中的 CAN in Automation (CiA) 定义了生成至少 16 位时间戳的方法，可以选择由 CAN 控制器进行支持。从基本概念来讲，CiA 603 是使用一个自由运行的定时器来计数时钟周期，而不是计数 CAN 位时间，如图 22-13 所示，在发生事件后，定时器值被获取，这个就是时间戳。该时间戳分别存放在 RTS 或者 TTS，可以由主机读取，存储时间戳之后，会通过握手信号指示该操作已完成，从而触发 CAN 控制器 3 个时钟域的跨时钟域同步机制 (CDC)。

图 22-13 时间戳和 CDC 机制



可以通过设置 TSPOS 位，选择 SOF 或 EOF 采样点获取时间戳，ACK 分隔符之后的 7 个隐性位构成 CAN 或者 CANFD 帧的 EOF 位，对于接收器而言，一帧数据在 EOF 的倒数第二位有效，而对于发送器而言，一帧数据在 EOF 的最后一位有效。

在大多数系统中，基于接收和发送中断的软件时间戳被广泛使用，在有效帧的 EOF 处获取时间戳与软件时间戳类似。

CiA 603 支持时间戳和 AUTOSAR 的时间同步。对于 AUTOSAR 来说，CAN 网络中的一个节点就是时间主机。时间主机发送同步报文(SYNC 报文)，时间主机和所有时间从机均可以获取 SYNC 报文的时间戳。从发出 SYNC 报文指令到 SYNC 报文被真正发送之间的时间差将由时间主机通过随后的报文发出。因此，CAN 控制器针对发送帧 (TTS) 仅提供一个时间戳，而针对全部接收帧 (RTS) 提供单独的时间戳是可以的。

满足使用要求的。可以通过 TBUF 中的 TTSEN 位，单独使能和关闭每一个帧的发送帧的时间戳。CiA 定义的读取和更改定时器的相关规则，CAN 控制器不具有定时器，它依靠的是一个外部定时器，外部定时器的来源可通过 SCFG_CFG2 寄存器的 CAN1_TST_SEL 位来配置。CAN 控制器仅包含时间戳机制，存储 TTS 的寄存器和存储 RTS 的寄存器。

寄存器 TNCFG 的 TSEN 位用于关闭和启用时间戳，如果时间戳关闭，则 TTS 和 RTS 均是 0。

22.6 CAN 位时间

22.6.1 位速率

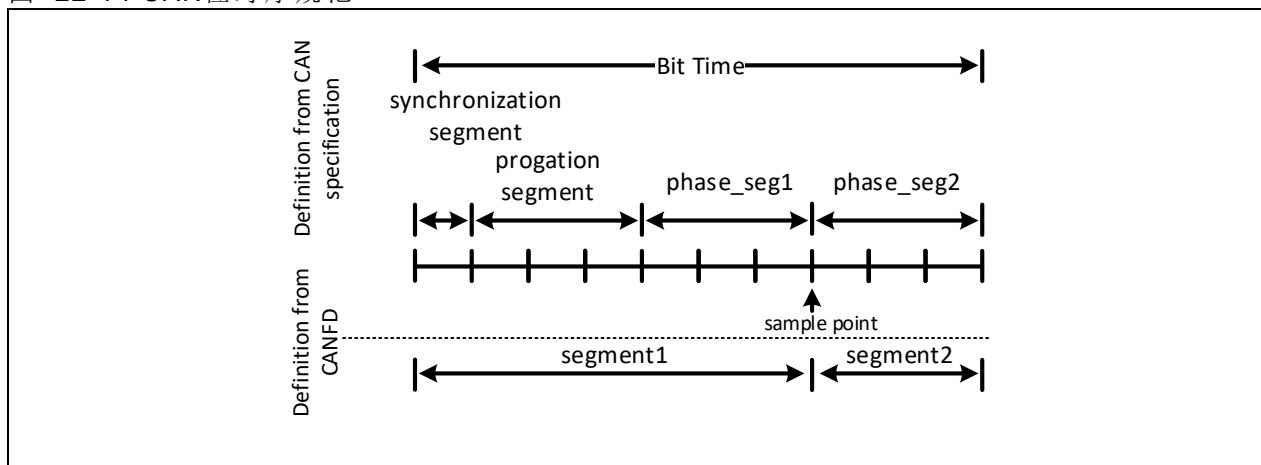
Classic CAN2.0B 定义了高达 1Mbit/s 比特率，对于 CANFD 帧则没有固定限制，对于实际通信系统，数据速率取决于所使用的收发器、所需的 CAN 总线拓扑结构和 CAN 控制器可实现时钟频率的限制。

CANFD 可以被配置为任意比特率，仅受位时序和分频寄存器适当配置范围的限制。

Classic CAN2.0B 在整帧中以一个选定的比特率通信，与之相对的是，CANFD 在帧的开头以“慢速”（标称比特率）通信，在帧的中间切换至“快速”（数据比特率），之后再帧的结尾或者检测到错误之后切换至“慢速”（标称比特率）。

22.6.2 位时间定义

图 22-14 CAN 位时序规范



CAN 通信定义了称为“时钟数据恢复”（重同步）的机制，作为 CAN 协议规范的一部分，这些重同步机制基于 CAN 位的定义，如图 22-14 所示。对于 Classic CAN2.0B 和 CANFD，CAN 位时间的定义遵循相同的概念：CAN 位时间 BT 由几个段组成，每段由 n_{TQ} 个时间片单元组成，时间片单元的时间长度 t_{TQ} 为：

$$t_{TQ} = \frac{n_{prescaler}}{f_{can_clk}}$$

n_{TQ} 和 $n_{prescaler}$ 的值需要根据系统时钟频率 f_{can_clk} 来配置，以匹配期望的位时间 BT，比特率 $BR = 1/BT$ ，其中 x 是以 AC 或 FD 为前缀的占位符。

$$\begin{aligned} BT &= n_{TQ} * t_{TQ} = n_{TQ} * \frac{n_{prescaler}}{f_{can_clk}} = t_{Seg_1} + t_{Seg_2} \\ t_{Seg_1} &= (x_SEG_1 + 2) * t_{TQ} \\ t_{Seg_2} &= (x_SEG_2 + 1) * t_{TQ} \\ t_{SJW} &= (x_SJW + 1) * t_{TQ} \\ n_{prescaler} &= PRESC + 1 \end{aligned}$$

CAN 控制器简化了位时间配置，如图所示，同步段、传播段和 phase_seg1 段被组合为段 1，位段 1 的时间长度为 t_{Seg_1} ，位段 2 的时间长度为 t_{Seg_2} 。

SJW 是同步跳跃宽度，为 TQ 的整数倍，同步的目的是确保接收到的数据信号电平的转换处于同步段中，以将采样点移动到接收信号稳定的位置。如果电平转换提早或延迟到达，CAN 节点可以移除或者增加一些 TQ 来缩短或者延长 CAN 位时间，SJW 限制了一次同步所能移除或者增加 TQ 的最大数量。信号在 CAN 网络中传播需要一定时间，信号反射（“振铃”）是造成传播延时的物理效应之一，信号

传播应该在 CAN 位的传播段内完成。

CAN 节点对接收到的比特流进行重同步，重同步发生在隐性电平到显性电平的边沿，CAN 节点插入填充位，以保证重同步事件发生的已知最大间隔。因此，在重同步之间会接收到几位的数据，由于振荡器容差的缘故，发送节点和接收节点将发生轻微的漂移，下一次重同步事件将对此进行补偿。`phase_seg1` 和 `phase_seg2` 用于保留发送节点和接收节点之间的相位偏差，因此，`phase_seg1` 和 `phase_seg2` 段持续时间相同。

`SJW` 的最大值等于 `phase_seg2` 段，换言之，重同步的最大数量与两次重同步事件之间预期的最大相位偏差一致。

配置 CAN 位时间以实现最快的比特率是一个复杂的议题，简单言之：

- 信号在网络中的传播与传播段相关
- 发送节点和接收节点之间的振荡器容差影响段1和段2的相位

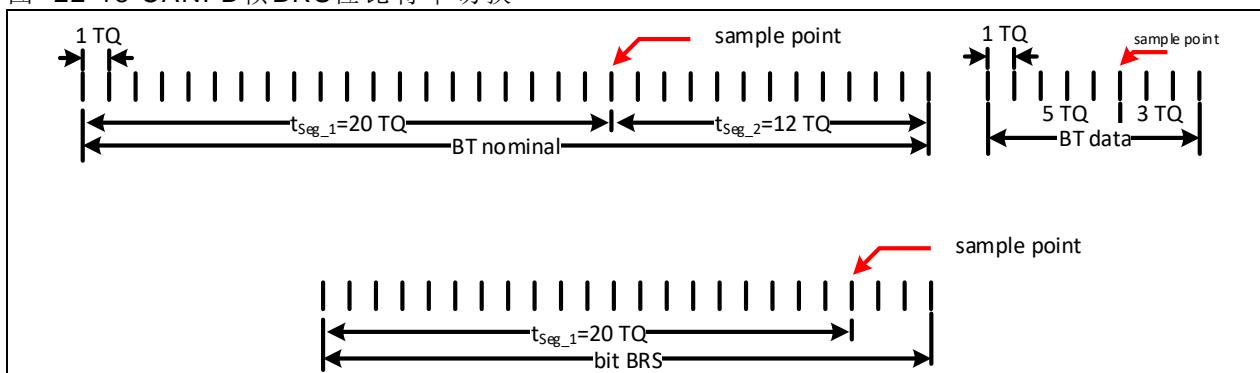
合适的 CAN 位时间配置如下：

- 采样点应该大致在位时间的 60% 到 80% 范围内，通常，将其设置为 75% 到 80% 是最佳选择
- 如果可能的话，选择较低的分频器配置，从而每个位时间可以产生更多的时间片 TQ
- 在大多数情况下，优先设置 `SJW` 等于段 2 的时间长度

22.6.3 CANFD 位速率转换和采样点

在 CANFD 帧的比特率切换有效 (`BRS=1`) 的情况下，对于 CAN 网络中的所有节点，需要具有相同确切位置的采样点。在 `BRS` 位和 CRC 界定符的采样点位置，比特率进行切换，所以这些位的时间长度介于“慢速”和“快速”之间。如下图 22-15 所示，采样点的位置对 `BRS` 位的绝对时间长度由很大影响，如果数据比特率远远大于标称比特率，那么较早或较晚的采样点可能会导致快速数据比特率下的错误采样。

图 22-15 CANFD 帧 BRS 位比特率切换



总之，采样点的位置对于比特率切换是非常重要的，所以在标称比特率和 CANFD 数据比特率中，建议在 CAN 通信网络中，所有的 CANFD 帧节点采用相同的时序参数。换言之，所有节点的段 1 和段 2 应该设置为相同的时间长度。此外建议采用相同时间长度的时间片 TQ，以便使所有的节点同步相等。

22.6.4 TDC 和 RDC

对于 CANFD 帧节点，发送器延迟补偿 (TDC) 可以选择性使能，而接收器延迟补偿 (RDC) 自动激活 (Classic CAN2.0 帧节点不支持 TDC 和 RDC 功能)。TDC 和 RDC 机制的应用基于以下背景：采用 CANFD 的数据比特率进行通讯时，发送器的发送延迟或者总线延迟可能会大于一个位时间，这种情况下，需要使用 TDC 进行延时补偿，否则 CANFD 帧数据场的比特率会受到限制，因为如果发送器在比特位的采样点无法检测到自己最新传输的比特位，则发送器会检测到位错误。

图 22-16 发送器延迟

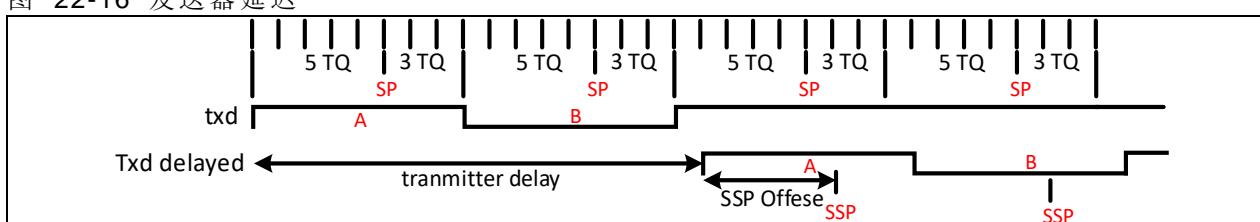


图 22-16 给出了一个发送器延时较大时对总线影响的示例，图中，发送器开始发送 A 和 B 两个比特位，位时间 $t_{Seg_1} = 5TQ$, $t_{Seg_2} = 3TQ$, 采样点 (SP) 位于段 1 和段 2 之间。在示例中，发送器的延时大于 2 个位时间，因此原始的采样点 (SP) 无法采到正确的位值，CANFD 规范定义了附加二次采样点 (SSP)。如果 TDC 使能 ($SSPOFF \neq 0$)，CAN 控制器自动确定发送器延时。SSP 的位置采用发送器的延时加上 SSP 延时 (由 $SSPOFF$ 配置) 得到， $SSPOFF$ 必须配置为 TQ 的整数倍，并建议采用 $t_{Seg_1} + 1$ 。

注：建议 $SSPOFF$ 采用 $t_{Seg_1} + 1$ 而不是 t_{Seg_1} ，是因为采样一些数据时存在量化误差。在 CANFD 帧， FDF 位与 res 位的下降沿位置，发送器的延时自动进行处理。

ISO 11898-1:2015 规定在 TDC 使能情况下，只能配置分频寄存器 $PRESC=0$ 或 1 ，对于 CAN 控制器而言，在数据比特率下，能够自动检测高达 3 个位时间的发送器延时。RDC 在接收过程中自动完成，不论 TDC 是否开启。

注：RDC 只能在 $n_{prescaler} > 1$ 的情况下起作用。

22.7 CAN 寄存器

除 ACF/RBUF/TBUF 相关寄存器，外设寄存器可以按字节（8 位）的方式进行写操作，但只能按字（32 位）的方式进行读操作。

ACF/RBUF/TBUF 相关寄存器只能按字（32 位）的方式进行读写操作。

表 22-9 SPI 寄存器列表及其复位值

寄存器简称	基址偏移量	复位值
TNCFG	0000h	0x0200 0000
ACTIME	0004h	0x0505 0008
FDTIME	0008h	0x0202 0003
LBTCFG	0010h	0x7700 0000
STS	0014h	0x0000 0000
TSTAT	001Ch	0x0000 0000
TTS	0020h	0x0000 0000
CTRLSTAT	0028h	0x0090 0080
ERR	002Ch	0x0000 0010
REFMSG	0030h	0x0000 0000
TTCFG	0034h	0x0000 0000
TTTRIG	0038h	0xFFFF 0000
ACF	0044h~0064h	
ACFCTRL	0044h	0x0001 0000
FCID	0048h	0XXXXX XXXX
FCFMT	004Ch	0XXXXX XXXX
FCTYP	0050h	0XXXXX XXXX
FCACF	0054h	0XXXXX XXXX
FMID	0058h	0XXXXX XXXX
FMFMT	005Ch	0XXXXX XXXX
FMTYP	0060h	0XXXXX XXXX
FMACF	0064h	0XXXXX XXXX
RBUF (包括 RTS)	0070h~00CBh	
RBID	0070h	0XXXXX XXXX

RBFMT	0074h	0XXXXX XXXX
RBTP	0078h	0XXXXX XXXX
RBACF	007Ch	0XXXXX XXXX
RBDAT1	0080h	0XXXXX XXXX
RBDAT2	0084h	0XXXXX XXXX
...
RBCIA1	T+0	0XXXXX XXXX
RBCIA2	T+4	0XXXXX XXXX
RBTCAN	T+8	0XXXXX XXXX
TBUF	00CCh~011Bh	
TBID	00CCh	0XXXXX XXXX
TBFMT	00D0h	0XXXXX XXXX
TBTYP	00D4h	0XXXXX XXXX
TBACF	00D8h	0XXXXX XXXX
TBDATA1	00DCh	0XXXXX XXXX
TBDATA2	00E0h	0XXXXX XXXX
...
LLCFORMAT	0124h	0x0000 0000
LLCSIZE	0128h	0x0010 0000
INTEN	0140h	0x0000 0000

22.7.1 CAN CiA 603时间戳及节点控制寄存器 (TNCFG)

域	简称	复位值	类型	功能
位 31: 26	保留	0x0	resd	请保持默认值。
位 25	TSPOS	0x1	rw	时间戳位置 (Time stamping position) 0: SOF 1: EOF TSPOS 在 TSEN 配置为 0 时可更改，也可以在配置 TSEN 为 1 的同时修改 TSPOS 配置。
位 24	TSEN	0x0	rw	时间戳使能 (Time stamping enable) 0: 禁用 1: 使能
位 23: 18	保留	0x0	resd	请保持默认值。
位 17	ROP	0x0	rw	受限操作 (Restricted operation) 0: 受限操作关闭; 1: 受限操作打开。 注：当传输正在进行中时，ROP 禁止修改。如果 ROP 配置为 1，则禁止发送数据。
位 16: 0	保留	0x0	resd	请保持默认置

22.7.2 Classic CAN2.0B位时序寄存器 (ACTIME)

域	简称	复位值	类型	功能
位 31	保留	0x0	resd	请保持默认值。
位 30: 24	AC_SJW	0x5	ro	重新跳跃宽度 $t_{SJW} = (AC_SJW + 1) * t_{TQ}$
位 23	保留	0x0	resd	保持默认值。
位 22: 16	AC_SEG_2	0x5	rw	位时间段 2 设置为 $t_{SEG_2} = (AC_SEG_2 + 1) * t_{TQ}$

位 15: 9	保留	0x0	resd	请保持默认值
位 8: 0	AC_SEG_1	0x8	rw	位时间段 1 采样点设置为 $t_{SEG_1} = (AC_SEG_1 + 2) * t_{TQ}$

22.7.3 CANFD位时序寄存器 (FDTIME)

域	简称	复位值	类型	功能
位 31	保留	0x0	resd	请保持默认值。
位 30: 24	FD_SJW	0x2	rw	重新跳跃宽度 $t_{SJW} = (FD_SJW + 1) * t_{TQ}$
位 23	保留	0x0	resd	请保持默认值。
位 22: 16	FD_SEG_2	0x2	rw	位时间段 2 设置为 $t_{SEG_2} = (FD_SEG_2 + 1) * t_{TQ}$
位 15: 8	保留	0x0	resd	请保持默认值
位 7: 0	FD_SEG_1	0x3	rw	位时间段 1 采样点设置为 $t_{SEG_1} = (FD_SEG_1 + 2) * t_{TQ}$

22.7.4 CAN限制和位时间配置寄存器 (LBTCFG)

域	简称	复位值	类型	功能
位 31	保留	0x0	resd	请保持默认值
位 30: 28	RETLIM	0x7	rw	重发次数限制 (Retransmission limit) 111: 无限制 (只被发送错误计数器 TECNT 限制) 110: 7 次 ... 000: 1 次 (不重发) 如果在发送过程中出错, CAN 节点在总线空闲时立即重新发送。重新发送的次数限制通过 RETLIM 进行配置。 RETLIM 可以在任意时刻进行更新, 但是当 RETLIM 更新后, 需要将该寄存器同步至 can_clk 时钟域供 CAN 协议状态机使用, 在此期间, RETLIM 的写过程被锁定。
位 27	保留	0x0	resd	请保持默认值
位 26: 24	REALIM	0x7	rw	重新仲裁次数限制 (Re-arbitration limit) 111: 无限制 110: 7 次 ... 000: 1 次 (不重新仲裁) 如果两个或者更多 CAN 节点同时发送报文, 低优先级的报文丢失仲裁退出总线竞争。当总线处于空闲状态时, 丢失仲裁的 CAN 节点立即重新参与总线仲裁, 重新仲裁的次数限制由 REALIM 进行配置。REALIM 可以在任意时刻进行更新, 但是当 REALIM 更新后, 需要将该寄存器同步至 can_clk 时钟域供 CAN 协议状态机使用, 在此期间, REALIM 的写过程被锁定。
位 23: 16	保留	0x0	resd	请保持默认值
位 15: 8	FD_SSPOFF	0x0	rw	二次采样点偏移 (Secondary sample point offset) 当 FD_SSPOFF 不等于 0 且 CES 配置为 1 时, 发送延迟补偿 (TDC) 使能。假如 BRS 被激活, TDC 在 CAN FD 帧的数据场起作用。 发送器延迟加上 FD_SSPOFF 定义的时间为 TDC 之后的第二次采样点。FD_SSPOFF 是时间片 TQ 的整数。
位 7: 5	保留	0x0	resd	请保持默认值
位 4: 0	PRES	0x0	rw	分频器 (Prescaler) 系统时钟根据分频器配置分频得到时间片 tq。 $Nprescaler = PRES + 1$ 或者 $Ttq = Nprescaler/fcan_clk$

22.7.5 CAN状态寄存器 (STS)

域	简称	复位值	类型	功能
位 31	EWARN	0x0	ro	达到错误警告限制 (Error warning limit reached) 0: RECNT 和 TECNT 计数器小于 EWL 1: RECNT 或者 TECNT 其中一个计数器大于等于 EWL
位 30	EPASS	0x0	ro	错误被动状态 (Error passive status) 0: 节点处于错误主动 1: 节点处于错误被动
位 29: 14	保留	0x00	resd	保持默认值。
位 13	WTIF	0x0	rw1c	TTCAN: 窗口触发中断标志 (Watch trigger interrupt flag) 当周期计数达到 TT_WTRIG 配置限制值并且 WTIE 使能情况下, WTIF 被置位。
位 12	TEIF	0x0	rw1c	TTCAN: 触发错误中断标志 (Trigger error interrupt flag) TEIF 的置位条件如文中描述, 没有对应的寄存器用于禁用 TEIF 的状态显示。
位 11	TTIF	0x0	rw1c	TTIF: 时间触发中断标志 (Time trigger interrupt flag) 当 TTIE 被配置时, 周期时间等于触发时间 TT_TRIG, 则 TTIF 被置位。该位写 1 清零, 写 0 无影响。如果 TT_TRIG 不被更新, 那么在下一次的基本周期, TTIF 不会被置位, 即仅置位一次。
位 10	EPIF	0x0	rw1c	错误被动中断使能 (Error passive interrupt flag) 如果 EPIE = 1, 当错误状态从主动错误状态切换至被动错误状态, 则 EPIF 被置位, 反之亦然。
位 9	ALIF	0x0	rw1c	仲裁丢失中断标志 (Arbitration lost interrupt flag)
位 8	BEIF	0x0	rw1c	总线错误中断标志 (Bus error interrupt flag)
位 7	RIF	0x0	rw1c	接收中断标志 (Receiv interrupt flag) 0: 未接收到帧数据 1: 接收到一帧数据帧或者远程帧, 并存储至接收缓存
位 6	ROIF	0x0	rw1c	RB 溢出中断标志 (RB overflow interrupt flag) 0: RB 没有数据覆盖 1: RB 中至少有一帧接收的数据被覆盖
位 5	RFIF	0x0	rw1c	RB 满中断标志 (RB full interrupt flag) 0: RB FIFO 未满 1: 所有 RB 满。如果在下一帧有效数据接收完成之前, 没有 RB 被释放, 则最先存储的帧数据丢失。
位 4	RAFIF	0x0	rw1c	RB 几乎满中断标志 (RB almost full interrupt flag) 0: 填充的 RB 槽数量 < AFWL_i 1: 填充的 RB 槽数量 ≥ AFWL_i
位 3	TPIF	0x0	rw1c	发送主中断标志 (Transmission primary interrupt flag) 0: PTB 没有发送完成 1: PTB 的发送请求已成功执行 在 TTCAN 模式中, TPIF 不会被置位, 只有 TSIF 标志可用。
位 2	TSIF	0x0	rw1c	发送次中断标志 (Transmission secondary interrupt flag) 0: STB 没有发送完成 1: STB 的发送请求已成功执行 在 TTCAN 模式中, 无论帧存储的位置, 所有成功的发送都会置位 TSIF。
位 1	EIF	0x0	rw1c	错误中断标志 (Error interrupt flag) 0: 状态无任何改变 1: 越过错误警告界限, 高于或者低于错误警告界限 BUSOFF 位硬件置 1 或者清除
位 0	AIF	0x0	rw1c	中止中断标志 (Abort interrupt flag) 0: 未中止发送 1: 当在设置 TPA 或者 TSA 之后, 帧数据发送中止。不建议同时将 TPA 和 TSA 置位, 因为两者都会置位 AIF。 AIF 标志位无对应的使能位。

22.7.6 CAN发送状态寄存器 (TSTAT)

域	简称	复位值	类型	功能
位 31: 27	保留	0x0	resd	保持默认值 当前发送状态 (Current transmission status code) 000: 无传输进程 (IDLE) 001: 正在传输进程, 且无任何问题 (ONGOING) 010: 仲裁丢失, 根据 REALIM 配置进行重新仲裁 (LOST_ARBITRATION) 011: 传输成功完成 (TRANSMITTED) 100: 传输中止, 配置 TPA 或者 TSA (ABORTED) 101: 传输错误, 根据 RETLIM 配置进行重新发送 (DISTURBED) 110: LCC 帧格式配置错误 (REJECTED) 111: 保留
位 26: 24	TSTAT_2	0x0	ro	建议使用软件计数器作为 HANDLE 值, 该值每发送一帧新的数据进行递增。
位 23: 16	HANDLE_2	0x0	ro	用于帧识别的句柄 (Handle for frame identification)
位 15: 11	保留	0x0	resd	保持默认值 最终发送状态 (Final transmission status code) 000: 无传输进程 (IDLE) 001: 正在传输进程, 且无任何问题 (ONGOING) 010: 仲裁丢失, 根据 REALIM 配置进行重新仲裁 (LOST_ARBITRATION) 011: 传输成功完成 (TRANSMITTED) 100: 传输中止, 配置 TPA 或者 TSA (ABORTED) 101: 传输错误, 根据 RETLIM 配置进行重新发送 (DISTURBED) 110: LCC 帧格式配置错误 (REJECTED) 111: 保留
位 10: 8	TSTAT_1	0x0	ro	建议使用软件计数器作为 HANDLE 值, 该值每发送一帧新的数据进行递增。
位 7: 0	HANDLE_1	0x0	ro	用于帧识别的句柄 (Handle for frame identification)

22.7.7 CAN发送时间戳32位 (TTS)

域	简称	复位值	类型	功能
位 31: 0	TTS	0x0	ro	发送时间戳 (Transmission time stamp) TTS 记录上一帧发送数据对应的 CiA 603 时间戳。如果更新使能控制位 TTSEN = 1, 则每发送一帧新的数据, 覆盖 TTS。时间戳为 32 位, 未使用位强制为 0。TTS 用于获取同步信息的时间戳。

22.7.8 CAN控制和状态寄存器 (CTRLSTAT)

域	简称	复位值	类型	功能
位 31	SACK	0x0	rw	自响应 (self-acknowledge) 0: 禁用自响应 1: 当 LBME=1 时自响应
位 30	ROM	0x0	rw	接收缓存溢出策略 (Receive buffer overflow mode) 0: 最先接收的帧数据被覆盖 1: 新接收的帧数据丢弃
位 29	ROV	0x0	ro	接收缓存溢出 (Receive buffer overflow) 0: 没有溢出 1: 溢出, 至少一帧数据丢失 ROV 在配置 RREL=1 时被硬件清除。
位 28	RREL	0x0	rw	接收缓存释放 (Receive buffer release) 上位机 (软件) 读 RB 槽然后释放缓存。之后硬件指针指向下一个 RB 槽, RSTAT 更新。 0: 无释放操作

位 27	RBALL	0x0	rw	1: 释放: 上位机(软件)已读缓存 RB 接收缓存存储所有数据帧 (Receive buffer stores all data frames) 0: 正常操作 1: RB 缓存存储所有正确或者错误的帧数据
位 26	保留	0x0	resd	保持默认值 接收缓存状态 (Receive buffer status) 00: 空 01: 非空非几乎满 (AFWL 配置) 10: 大于等于几乎满 (AFWL 配置阈值), 非满且无上溢 11: 满 (当上溢时, 维持置位状态, 上溢标志参考 ROV)
位 25: 24	RSTAT	0x00	ro	CAN FD ISO 模式 (CAN FD ISO mode) 0: Bosch CAN FD (non-ISO) 模式 1: ISO CAN FD 模式 (ISO 11898-1:2015)
位 23	FD_ISO	0x1	rw	ISO CAN FD 模式的 CRC 初始值不同, 另外还有额外的填充位计数器。两个模式在同一个 CAN 网络中并不兼容。该位配置对 CAN2.0B 无影响。 该位只能在 RESET=1 时进行改写。
位 22	TSNEXT	0x0	rw	下一个发送缓冲区 (Transmit buffer secondary next) 0: 无动作 1: STB 槽已填充, 选择下一个槽 当所有的帧数据写入 TBUF 寄存器, 上位机(软件)需要置位 TSNEXT 来指示当前槽已填充, 之后 TBUF 寄存器指向 STB 需要填充帧数的下一个槽。一旦槽被标记为已填充一帧数据, 配置 TSONE 和 TSALL 寄存器可以开始传输。 TSNEXT 和 TSONE 或者 TSALL 可以在同一个写周期配置, TSNEXT 被软件设置, 之后硬件立马自动清零。 当 TBSEL=0 时, 设置 TSNEXT 无意义, 此时设置 TSNEXT 对硬件无影响, 并被硬件自动清零。 如果 STB 满, TSNEXT 保持置位直到 STB 有一个槽被释放。在 TTCAN 模式, TSNEXT 不可用, 被固定为 0.
位 21	TSMODE	0x0	rw	STB 操作模式 (Transmit buffer secondary operation mode) 0: FIFO 模式 1: 优先级决定模式 FIFO 模式下, 帧数据按写入 STB 的先后顺序依次发送。 优先级决定模式下, STB 拥有最高优先级的帧数据自动最先发送。数据帧的 ID 号决定帧数据的优先级, 帧数据拥有最小 ID 号则优先级最高。PTB 中的帧数据无论 ID 号大小, 总是最高优先级。 TSMODE 只在完成一次软件复位之后 (置位与清除 RESET 位), 并且 STB 为空时进行切换。
位 20	TTTBM	0x1	rw	TTCAN 发送缓存模式 (TTCAN transmit buffer mode) 如果 TTEN=0 那么 TTTBM 不起作用, 否则: 0: PTB 和 STB 彼此独立, TSMODE 决定行为 1: 支持 TTCAN: TBPTR 和 TTPTR 选择缓存槽 对于事件驱动的 CAN 通讯 (TTEN=0), 系统提供 PTB 和 STB, TSMODE 配置决定 STB 的行为, TTTBM 不起作用。 对于时间触发的 CAN 通讯 (TTEN=1), 支持时间触发通讯的所有特征, TTTBM 需要被配置为 1。TTPTR 和 TBPTR 指向发送缓存槽的指定位置。 对于时间触发的 CAN 通讯 (TTEN=1), 只支持接收报文时间戳, 则 TTTBM 可以被配置为 0。发送缓存采用事件触发模式, TSMODE 决定其行为。 TTTBM 只能在 TBUF 为空时进行更改。
位 19	保留	0x0	resd	保持默认值

位 18	TSFF	0x0	rw	如果 TTEN=0 或 TTTBM=0: STB 缓存满标志 (Transmit secondary buffer full flag) 1: STB 已填充最大数量的帧数据 0: STB 未填充最大数量的帧数据 如果 TTEN=1 且 TTTBM=1: 发送缓存槽满标志 1: TBPTR 指定的缓存槽满 0: TBPTR 指定的缓存槽空 当 TSNEXT=1 之后的一个额外的<host_clk>周期 TSFF 被更新。因此设置 TSNEXT=1 时, TSFF 不可以立马被读。
位 17: 16	TSSTAT	0x0	ro	STB 状态位 (Transmit secondary status bits) 如果 TTEN=0 或 TTTBM=0 00: STB 空 01: STB 小于等于 1/2 满 10: STB 大于 1/2 满 11: STB 满 如果 TTEN=1 且 TTTBM=1: 00: PTB 和 STB 空 01: PTB 和 STB 非空非满 11: PTB 和 STB 满 当 TSNEXT=1 之后的一个额外的<host_clk>周期 TSSTAT 被更新。因此设置 TSNEXT=1 时, TSSTAT 不可以立马被读。
位 15	TBSEL	0x0	rw	传输缓存选择 (Transmit buffer select) 选择发送缓存来导入一帧数据。使用 TBUF 寄存器进行访问。当 TBUF 寄存器正在进行写操作或者 TSNEXT 被置位时, TBSEL 需要保持稳定。 0: PTB (高优先级缓存) 1: STB
位 14	LOM	0x0	rw	当 TTEN=1 和 TTTBM=1 时, 该位被复位成硬件复位值。 只听模式 (Listen only mode) 0: 禁用 1: 使能 当传输过程正在进行时 LOM 不可被更改。当 LOM 使能和 LBME 禁用情况下, 传输启动禁止。 LOM=1 且 LBME=0 禁用所有传输 LOM=1 且 LBME=1 禁止接收帧或者错误帧回复 ACK, 但本节点内部传输。
位 13	STBY	0x0	rw	收发器待机模式 (Transceiver standby mode) 0: 禁用 1: 使能 该寄存器为外部输出信号 stby, 控制收发器进入待机模式。 当 TPE, TSONE 或者 TSALL 等于 1 时, STBY 不能被置位。如果上位机控制 STBY 切换至 0, 主机在请求新的传输之前需要等待收发器启动。
位 12	TPE	0x0	rw	PTB 发送使能 (Transmit primary enable) 0: 无 PTB 的传输 1: 高优先级 PTB 发送帧数据使能 如果 TPE 被置位, PTB 中的帧数据会在下一帧进行发送。在此之前, STB 中正在进行的传输进程会完成, 新的挂起的帧数据会延迟至 PTB 帧数据传输完成之后再进行发送。 TPE 使能在当前帧数据发送成功或者配置 TPA 中止当前帧发送时, 才可被硬件清零。 当 RESET=1, STBY=1, (LOM=1 且 LBME=1), (TTEN=1 且 TTTBM=1) 或者 ROP=1 时, 该位被复位成硬件复位值。
位 11	TPA	0x0	rw	PTB 发送中止 (Transmit primary abort) 0: 无中止

				1: 中止 PTB 已经被请求发送 (TPE=1) 但是还没有开始发送的传输进程，PTB 中的帧数据依然保存。 该位被上位机（软件）置位，硬件清零。置位 TPA 将自动清除 TPE 位。 上位机（软件）可以配置 TPA 为 1，但是不可以复位为 0。在硬件复位该位期间，该位不可以被置位。 当 RESET=1 或者 (TTEN=1 且 TTTBM=1) 时，该位被复位成硬件复位值。TPA 和 TPE 不应该同时被置位。
位 10	TSONE	0x0	rw	STB 发送一帧数据 (Transmit secondary one frame) 0: 无 STB 的传输 1: STB 发送一帧数据使能。在 FIFO 模式，最先写入的帧数据被发送；在优先级模式，高优先级的一帧数据被发送。在优先级模式，如果同时 STB 写入新的帧数据，则并不总是清楚哪一帧数据会被优先发送。 当总线空闲且 PTB 没有请求挂起 (TPE=1) 时，控制器开始发送数据。 TSONE 使能在当前帧数据发送成功或者配置 TSA 中止当前帧发送时，才可被硬件清零。 当 RESET=1, STBY=1, (LOM=1 且 LBME=1), (TTEN=1 且 TTTBM=1) 或者 ROP=1 时，该位被复位成硬件复位值。
位 9	TSALL	0x0	rw	STB 发送所有帧数据 (Transmit secondary all frames) 0: 无 STB 的传输 1: STB 发送所有帧数据使能 当总线空闲且 PTB 没有请求挂起 (TPE=1) 时，控制器开始发送数据。 TSALL 使能在当前帧数据发送成功或者配置 TSA 中止当前帧发送时，才可被硬件清零。 当 RESET=1, STBY=1, (LOM=1 且 LBME=1), (TTEN=1 且 TTTBM=1) 或者 ROP=1 时，该位被复位成硬件复位值。
位 8	TSA	0x0	rw	STB 发送中止 (Transmit secondary abort) 0: 无中止 1: 中止 STB 已经被请求发送但是还没有开始发送的传输进程，对于 TSONE 传输，只有一帧数据被中止；对于 TSALL 传输，所有帧数据被中止。同时一帧或者所有帧存储被释放，TSSTAT 寄存器状态被更新。 所有被中止的帧数据丢失，不能再被访问。在优先级模式，TSONE 传输被中止，则当新的一帧数据同时被写入到 STB 时，当前中止的帧数据才会被清除。
位 7	RESET	0x1	rw	该位被上位机（软件）置位，硬件清零。置位 TSA 将自动清除 TSONE 和 TSALL 位。 上位机（软件）可以设置 TSA 为 1，但是不可以复位为 0。当 RESET=1 时，该位被复位为硬件复位值。 TSA 不应该与 TSONE 或者 TSALL 位同时被置位。
位 6	LBME	0x0	rw	RESET 请求位 (RESET request bit) 0: CAN-CTRL 无局部复位 1: 主控制器产生 CAN-CTRL 局部复位 一些寄存器（如节点配置）只能在 RESET = 1 时进行修改。寄存器 RESET 强制 CAN-CTRL 部分组件进入复位状态，当节点进入离线状态，RESET 被自动置位。 当 RESET 切换至 0，在检测到 11 个连续隐性电平之后，CAN 节点参与到 CAN 总线通信。 如果 RESET 被设置为 1，并马上切换至 0，CAN-CTRL 需要等待一段时间回到非复位状态，因为从<host_clk>时钟域同步至<can_clk>时钟域需要时间。
				回环模式，外部 (Loop back mode, external) 0: 禁用 1: 使能

				LBME 在传输进行时禁止更改。LBME 和 TPE, TSONE, TSALL 不应该同时更改。
位 5	LBMI	0x0	rw	回环模式, 内部 (Loop back mode, internal) 0: 禁用 1: 使能 LBMI 在传输进行时禁止更改。LBMI 和 TPE, TSONE, TSALL 不应该同时更改。
位 4: 1	保留	0x0	resd	保持默认值
位 0	BUSOFF	0x0	rw	总线离线 (Bus off) 0: 控制器处于在线状态 1: 控制器处于离线状态 写寄存器 BUSOFF 为 1 会复位 TECNT 和 RECNT, 该操作只能用于 debug 模式。

注: 同时设置 TSONE 和 TSALL 无意义, 在 TSONE 被置位的情况下, TSALL 不能被置位, 反之亦然。如果 TSONE 和 TSALL 被同时置位, TSALL 优先级更高, TSONE 被硬件清除。

22.7.9 CAN错误配置及状态寄存器 (ERR)

域	简称	复位值	类型	功能
位 31: 24	TECNT	0x0	ro	发送错误计数器 (Transmit error count) TECNT 按照 CAN 协议进行递增或递减。 当处于离线状态时, TECNT 可能溢出。
位 23: 16	RECNT	0x0	ro	接收错误计数器 (Receive error count) RECNT 按照 CAN 协议进行递增或递减。 RECNT 不会溢出。
位 15: 13	KOER	0x0	ro	错误类型 (Error code) 000: 无错误 001: 位错误 010: 格式错误 011: 填充错误 100: 应答错误 101: CRC 错误 110: 其它错误 (包括: 在错误标志之后检测到显性位, 接收显性错误标志过长, 在 ACK 错误之后发送隐性错误标志期间检测到显性位。) 111: 保留 每检测到新的错误时 KOER 更新, 因此当成功发送接收一帧数据时, 该寄存器保持不变。
位 12: 8	ALC	0x0	ro	仲裁丢失捕捉 (指定一帧数据仲裁丢失的位置)
位 7: 4	AFWL	0x1	rw	接收缓存几乎满警戒界限 (Receive buffer almost full warning limit) AFWL 根据可用的 RB 缓存槽数量 NRB 决定内部警戒界限。 如果 RB 缓存槽填充的帧数据数量等于 AFWL 则 RAFIF 被置位。有效的范围 AFWL=[1...6] AFWL 配置为 0 无意义, 硬件自动按 0x1 计算。 AFWL 大于 6 时无意义, 硬件自动按 6 计算。 AFWL 等于 6 有效, 此时 RFIF 也被置位。
位 3:0	EWL	0xB	rw	可编程错误警告界限=(EWL+1)*8 (Programmable error warning limit) 可能的值为: 8, 16, ... 128. EWL 的值决定 EIF 的状态。 EWL 需要经过时钟域同步同步至 CAN 时钟域。在同步过程中, EWL 寄存器的写操作被锁定, 直到同步完成。

22.7.10 CAN TTCAN: 参考信息 (REFMSG)

域	简称	复位值	类型	功能
位 31	REF_IDE	0x0	rw	参考信息 IDE 位 (Reference message IDE bit)
位 30: 29	保留	0x0	resd	请保持默认值
位 28: 0	REF_ID	0x0	rw	<p>参考信息 ID (Reference message identifier) 如果 REF_IDE 为: 1: REF_ID(28:0)可用 (基础 ID + 扩展 ID) 0: REF_ID(10:0)可用 (基础 ID)</p> <p>REF_ID 用于 TTCAN 模式识别参考信息, 适用于从机 (接收) 和主机 (发送)。如果参考信息被识别且没有错误, 则帧的 Sync_Mark 作为 Ref_Mark。</p> <p>REF_ID(2:0)不会检测, 对应的寄存器位强制设置为 0, 因此最多系统存在 8 个时间主机。</p> <p>硬件只能通过 ID 号识别参考信息, 数据长度无效。</p> <p>另外: 时间主机采用正常帧的方式发送参考信息, REF_ID 的作用是检测成功的传输了一帧参考信息。</p>

22.7.11 CAN TTCAN: 触发配置寄存器 (TTCFG)

域	简称	复位值	类型	功能
位 31: 27	保留	0x0	resd	保持默认值
位 26: 25	T_PRESC	0x0	rw	<p>TTCAN 计时器分频 (TTCAN timer prescaler) 00: 1 01: 2 10: 4 11: 8</p> <p>TTCAN 的时间基准是 CAN 位时间, 该时间被 PRESC, AC_SEG_1 和 AC_SEG_2 进行配置。T_PRESC 是额外的分频因子, 支持 1,2,4 或者 8 分频。</p> <p>T_PRESC 只能在 TTEN=0 时进行修改, T_PRESC 和 TTEN 可以在同一个写周期进行配置。</p>
位 24	TTEN	0x0	rw	<p>时间触发使能 (Time trigger enable) 1: TTCAN 使能, 计时器运行 0: 禁用</p>
位 23	TBE	0x0	rw	<p>指定 TB 槽为空 (Set TB slot to "Empty") 1: TBPTR 选中的槽空间被标记为空 0: 无动作</p> <p>只要槽空间被标记为空且 TSFF=0, TBE 立即被硬件自动复位为 0。如果当前 TB 槽正在进行数据传输, 则 TBE 保持置位状态, 直到当前传输完成, 或者产生传输错误, 或者仲裁丢失, 传输过程不再继续进行。</p> <p>如果 TBF 和 TBE 同时被置位, 则 TBE 优先。</p>
位 22	TBF	0x0	rw	<p>指定 TB 槽为满 (Set TB slot to "Filled") 1: TBPTR 选中的槽空间被标记为满 0: 无动作</p> <p>只要槽空间被标记为满且 TSFF=1, TBF 立即被硬件自动复位为 0。</p> <p>如果 TBF 和 TBE 同时被置位, 则 TBE 优先。</p>
位 21: 16	TB PTR	0x00	rw	<p>指向 TB 帧槽的指针 (Pointer to a TB frame slot) 0x00: 指针指向 PTB Others: 指针指向 STB 的一个槽</p> <p>TB PTR 指向的帧槽通过 TBUF 寄存器进行读/写操作。只有在 TSFF=0 时, 写操作有效。配置 TBF 为 1 标记槽空间满, 配置 TBE 为 1 标记槽空间空。</p> <p>在 TTCAN 模式, TBSEL 和 TSNEXT 配置不可用。</p> <p>TB PTR 只能指向硬件存在的缓存槽空间, 未被使用 TB PTR 位固定为 0。</p> <p>TB PTR 指向的槽空间最大限制为 PTB 和 63 个 STB 槽, TTCAN 模式下, 不能使用更多的槽空间。</p>

				如果 TBPTR 太大，指向了一个不可用的槽空间，则 TBF 和 TBE 被自动复位，不会进行任何动作。
位 15: 12 TEW	0x0	rw		发送使能窗口 (Transmit enable window) 对于单次传输触发，周期时间最多有 16 个时间刻度，指定帧传输的时刻。TWE+1 定义时间刻度的数量。 TEW=0 是可用的配置，指定发送允许窗口包含 1 个时间刻度。
位 11 保留	0x0	resd		保持默认值 触发类型 (Trigger type) 000: 立即触发，对应于立即传输 001: 时间触发，对应于接收触发 010: 单次发送触发，对应于专用时间窗口 011: 发送起始触发，对应于合并的仲裁窗口 100: 发送结束触发，对应于合并的仲裁窗口 其它：不可用 触发的时间由 TT_TRIG 定义，TTPTR 选择发送触发的 TB 槽。
位 10: 8 TTYPE	0x00	rw		
位 7: 6 保留	0x0	resd		保留默认值 发送触发 TB 槽指针 (Transmit trigger TB slot pointer) 如果 TT PTR 太大，指向不可用的槽空间，则 TEIF 被置位，只有当写入寄存器 TT TRIG_1 有效时，新的触发才被激活。 如果 TT PTR 指向一个空的槽空间，则当到达触发时间时，TEIF 被置位。
位 5: 0 TT PTR	0x00	rw		

22.7.12 CAN TTCAN: 触发时间 (TTTRIG)

域	简称	复位值	类型	功能
位 31: 16 TT_WTRIG	0xFFFF	rw		窗口触发时间 (Watch trigger time) TT_WTRIG[15: 0] 定义窗口触发的周期时间。初始的窗口触发采用最大的周期时间 0Xffff.
位 15: 0 TT_TRIG	0x0000	rw		触发时间 (Trigger time) TT_TRIG[15: 0] 定义触发的周期时间。对于发送触发，最早发送节点发送合适帧的 SOF 时间点是 TT_TRIG+1。

注：对 TT_TRIG 位 (TTTRIG[15:0]) 位的有效写操作会启动触发配置到 CAN 时钟域的数据同步，同时激活触发器，如果触发器被激活，寄存器 TTCFG[31:24] 被锁定，TT_TRIG 位处于激活状态，当到达触发时间 (TTIE 使能时 TTIF 被置位) 或者发生错误时 (TEIF 被置位)，寄存器写入锁定失效。

对 TT_WTRIG 位 (TTTRIG[31:16]) 位的有效写操作会启动触发配置到 CAN 时钟域的数据同步，同时激活触发器，如果触发器被激活，寄存器 TT_WTRIG 位 (TTTRIG[31:16]) 写入被锁定，为激活一次新的触发，对 TT_WTRIG 位 (TTTRIG[31:16]) 执行一次写操是必要的。

22.7.13 CAN 接收过滤器控制器 (ACFCTRL)

域	简称	复位值	类型	功能
位 31: 16 AE_x	0x01	rw		接收过滤器使能 (Acceptance filter enable) 1: 接收过滤器使能 0: 接收过滤器禁用 每个接收过滤器 (ACFC/ACFM) 可以被单独使能或禁用。过滤器编号 0 在硬件复位后默认使能。 禁用过滤器将拒收一帧数据。只有在使能过滤器且与 ACFC/ACFM 配置相匹配的情况下，帧数据才能被接收。
位 15: 4 保留	0x0	resd		保持默认值 接收过滤器地址 (Acceptance filter address)
位 3: 0 ACFADR	0x00	rw		ACFADR 指向一个特定的接收过滤器，选中的过滤器使用寄存器 ACFC 和 ACFM 进行访问。

当 ACFADR>15 时，该值无意义，并被自动按值 15 处理。

22.7.14 CAN 过滤器标识符 ID 区 (FCID)

参考表 22-1 标识符 (ID)

域	简称	复位值	类型	功能
位 31: 0	FCID_X	0x	rw	过滤器标识符 ID 区 (Filter code identifier) 寄存器配置值跟总线上接收到的数据对应的电平完全一致。

22.7.15 CAN 过滤器标识符格式区 (FCFMT)

参考表 22-1 格式 (FMT)

域	简称	复位值	类型	功能
位 31: 0	FCFMT_X	0x	rw	过滤器标识符格式区 (Filter code format) 寄存器配置值跟总线上接收到的数据对应的电平完全一致。

22.7.16 CAN 过滤器标识符类型区 (FCTYP)

参考表 22-1 类型 (TYP)

域	简称	复位值	类型	功能
位 31: 0	FCTYP_X	0x	rw	过滤器标识符类型区 (Filter code type) 寄存器配置值跟总线上接收到的数据对应的电平完全一致。

22.7.17 CAN 过滤器标识符接收场 (FCACF)

参考表 22-1 接收场 (ACF)

域	简称	复位值	类型	功能
位 31: 0	FCACF_X	0x	rw	过滤器标识符接收场 (Filter code acceptance field) 寄存器配置值跟总线上接收到的数据对应的电平完全一致。 注：只对于 CAN XL 帧格式可用。

22.7.18 CAN 过滤器掩码 ID 区 (FMID)

参考表 22-1 标识符 (ID)

域	简称	复位值	类型	功能
位 31: 0	FMID_X	0x	rw	过滤器掩码 ID 区 (Filter mask identifier) 0: 接收标识符的过滤检查被使能 1: 接收标识符的过滤检查被禁用

22.7.19 CAN 过滤器掩码格式区 (FMFMT)

参考表 22-1 格式 (FMT)

域	简称	复位值	类型	功能
位 31: 0	FMFMT_X	0x	rw	过滤器掩码格式区 (Filter mask format) 0: 接收标识符的过滤检查被使能 1: 接收标识符的过滤检查被禁用

22.7.20 CAN 过滤器掩码类型区 (FMTYP)

参考表 22-1 类型 (TYP)

域	简称	复位值	类型	功能
位 31: 0				

位 31: 0	FMTYP_X	0x	rw	过滤器掩码类型区 (Filter mask type) 0: 接收标识符的过滤检查被使能 1: 接收标识符的过滤检查被禁用
---------	---------	----	----	---

22.7.21 CAN过滤器掩码接收场 (FMACF)

参考表 22-1 接收场 (ACF)

域	简称	复位值	类型	功能
位 31: 0	FMACF_X	0x	rw	过滤器掩码接收场 (Filter mask acceptance field) 0: 接收标识符的过滤检查被使能 1: 接收标识符的过滤检查被禁用 注: 只对于 CAN XL 帧格式可用。

22.7.22 CAN接收缓冲区ID区 (RBID)

参考表 22-1 标识符 (ID)

域	简称	复位值	类型	功能
位 31: 0	RBID	0x	rw	接收缓冲区 ID 区 (RBUF Identifier)

22.7.23 CAN接收缓冲区格式区 (RBFMT)

参考表 22-1 格式 (FMT)

域	简称	复位值	类型	功能
位 31: 0	RBFMT	0x	rw	接收缓冲区格式区 (RBUF Format)

22.7.24 CAN接收缓冲区类型区 (RBTYP)

参考表 22-1 类型 (TYP)

域	简称	复位值	类型	功能
位 31: 0	RBTYP	0x	rw	接收缓冲区类型区 (RBUF Type)

22.7.25 CAN接收缓冲区接收场 (RBACF)

参考表 22-1 接收场 (ACF)

域	简称	复位值	类型	功能
位 31: 0	RBACF	0x	rw	接收缓冲区接收场 (RBUF Acceptance field) 注: 只对于 CAN XL 帧格式可用。

22.7.26 CAN接收缓冲区数据区 (RBDATn)

域	简称	复位值	类型	功能
位 31: 0	RBDATn	0x	rw	接收缓冲区数据区 (RBUF Data n)

22.7.27 CAN接收缓冲区 CiA 603 接收时间戳1 (RBCIA1)

域	简称	复位值	类型	功能
位 31: 0	RBCIA1	0x	rw	接收缓冲区 CIA 603 接收时间戳 1 (RBUF CiA 603)

22.7.28 CAN接收缓冲区 CiA 603 接收时间戳2 (RBCIA2)

域	简称	复位值	类型	功能
位 31: 0	RBCIA2	0x	rw	接收缓冲区 CIA 603 接收时间戳 2 (RBUF CiA 603)

22.7.29 CAN接收缓冲区 TT CAN 周期时间 (RBTTCAN)

域	简称	复位值	类型	功能
位 15: 0	RBTTCAN	0x	rw	接收缓冲区 TT CAN 周期时间 (RBTTCAN)

22.7.30 CAN发送缓冲区ID区 (TBID)

参考表 22-1 标识符 (ID)

域	简称	复位值	类型	功能
位 31: 0	TBID	0x	rw	发送缓冲区 ID 区 (TBUF Identifier)

22.7.31 CAN发送缓冲区格式区 (TBFMT)

参考表 22-1 格式 (FMT)

域	简称	复位值	类型	功能
位 31: 0	TBFMT	0x	rw	发送缓冲区格式区 (TBUF Format)

22.7.32 CAN发送缓冲区类型区 (TBTYP)

参考表 22-1 类型 (TYP)

域	简称	复位值	类型	功能
位 31: 0	TBTYP	0x	rw	发送缓冲区类型区 (TBUF Type)

22.7.33 CAN发送缓冲区接收场 (TBACF)

参考表 22-1 接收场 (ACF)

域	简称	复位值	类型	功能
位 31: 0	TBACF	0x	rw	发送缓冲区接收场 (TBUF Acceptance field) 注：只对于 CAN XL 帧格式可用。

22.7.34 CAN发送缓冲区数据区 (TBDATn)

域	简称	复位值	类型	功能
位 31: 0	TBDATn	0x	rw	发送缓冲区数据区 (TBUF Data n)

22.7.35 CAN LLC帧计算输入 (LLCFORMAT)

域	简称	复位值	类型	功能
位 31: 0	LLCFORMAT	0x00	rw	对 LLCFORMAT 寄存器进行写操作，硬件会自动计算一帧 LLC 数据的大小，并存储至 LLCSIZE 中。 LLCFORMAT 的布局与 LLC 帧的格式字的布局是完全相同的。 只有 LLCSIZE 计算相关的位才会存储至 LLCFORMAT 中，其它的位固定为 0。相关的位是：DLC，FDF，和 RMF。

22.7.36 CAN LLC帧计算输出 (LLCSIZE)

域	简称	复位值	类型	功能
位 31: 16	LLCAOT	0x10	ro	LLC 时间戳偏移地址 (LLC address offset of time-stamps) 基于 LLCFORMAT 中的值，LLCAOUT 提供 LLC 帧的时间戳偏移地址 (值 T)。 值 T 字对齐。 例如：对于负载为 1 个字节的帧数据，值 T： $T=16+4=20$ 。
位 15: 0	LLCPBYTES	0x0	ro	LLC 帧负载字节数 (LLC frame payload bytes) 基于 LLCFORMAT 中的值，LLCPBYTES 提供 LLC 帧数据的负载字节数。

22.7.37 CAN中断使能寄存器 (INTEN)

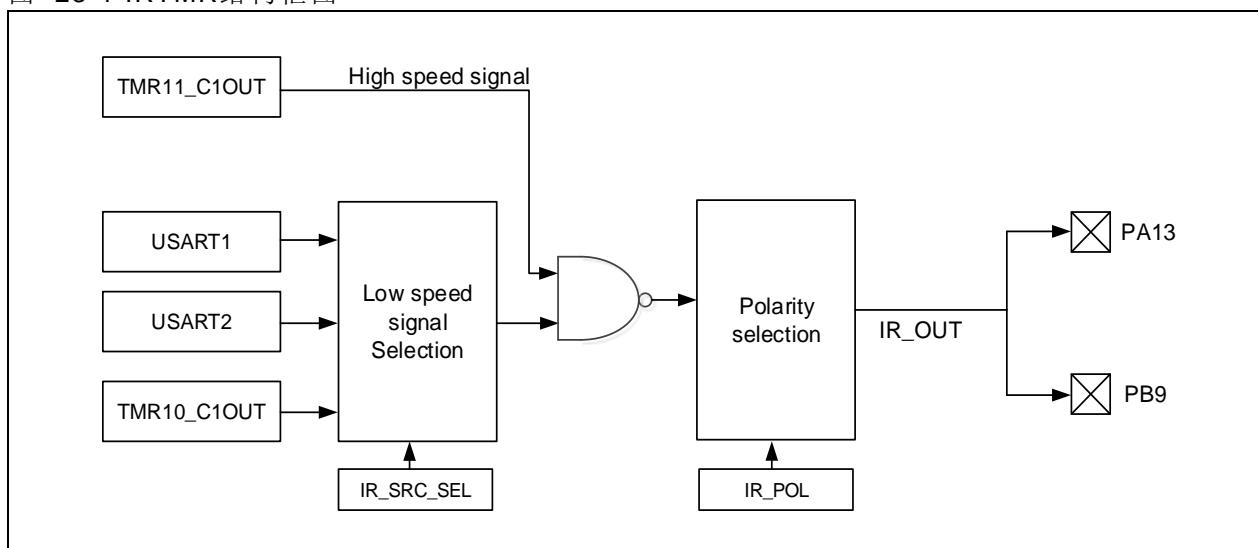
域	简称	复位值	类型	功能
位 31: 14	保留	0x00	resd	请保持默认值
位 13	WTIEN	0x0	rw	WTIF 标志输出中断使能， 默认关闭。
位 12	TEIEN	0x0	rw	TEIF 标志输出中断使能， 默认关闭。
位 11	TTIEN	0x0	rw	TTIF 标志输出中断使能， 默认关闭。
位 10	EPIEN	0x0	rw	EPIF 标志输出中断使能， 默认关闭。
位 9	ALIEN	0x0	rw	ALIF 标志输出中断使能， 默认关闭。
位 8	BEIEN	0x0	rw	BEIF 标志输出中断使能， 默认关闭。
位 7	RIEN	0x0	rw	RIF 标志输出中断使能， 默认关闭。
位 6	ROIEN	0x0	rw	ROIF 标志输出中断使能， 默认关闭。
位 5	RFIEN	0x0	rw	RFIF 标志输出中断使能， 默认关闭。
位 4	RAFIEN	0x0	rw	RAFIF 标志输出中断使能， 默认关闭。
位 3	TPIEN	0x0	rw	TPIF 标志输出中断使能， 默认关闭。
位 2	TSIEN	0x0	rw	TSIF 标志输出中断使能， 默认关闭。
位 1	EIEN	0x0	rw	EIF 标志输出中断使能， 默认关闭。
位 0	AIEN	0x0	rw	AIF 标志输出中断使能， 默认关闭。

23 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 TMR10_C1OUT、USART1、USART2 三者之中选择；高频载波信号由 TMR11_C1OUT 提供，SCFG_CFG1 寄存器 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 23-1 IRTMR 结构框图



24 调试 (DEBUG)

24.1 简介

Cortex™-M4F 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™-M4F 内核的调试可以通过串行调试接口实现。

ARM Cortex™-M4F 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

24.2 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 CAN，用户可以选择在调试期间是否停止或继续更新接收寄存器；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。在深度睡眠模式下，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。

24.3 I/O 控制

AT32M412/AT32M416 在所有的封装里都支持 SWD 调试，该调试共使用 2 个普通 I/O 口。复位以后，SWD 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 和 IOMUX 寄存器来切换这些 I/O 口功能。

24.4 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 24-1 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	0xXXXX XXXX
DEBUG_CTRL	0xE004 2004	0x0000 0000
DEBUG_APB1_PAUSE	0xE004 2008	0x0000 0000
DEBUG_APB2_PAUSE	0xE004 200C	0x0000 0000
DEBUG_APB3_PAUSE	0xE004 2010	0x0000 0000
DEBUG_SER_ID	0xE004 2020	0x0000 XX0X

24.4.1 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, 基地址为 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
0x7002_21C0	AT32M412CBT7	128KB	LQFP48
0x7002_2101	AT32M412C8T7	64KB	LQFP48
0x7002_21C2	AT32M412KBT7	128KB	LQFP32
0x7002_2103	AT32M412K8T7	64KB	LQFP32
0x7002_21C4	AT32M412KBU7-4	128KB	QFN32
0x7002_2105	AT32M412K8U7-4	64KB	QFN32
0x7002_21C6	AT32M412EBP7	128KB	TSSOP24
0x7002_2107	AT32M412E8P7	64KB	TSSOP24
0x7002_21C8	AT32M416CBT7	128KB	LQFP48
0x7002_2109	AT32M416C8T7	64KB	LQFP48
0x7002_21CA	AT32M416KBT7	128KB	LQFP32
0x7002_210B	AT32M416K8T7	64KB	LQFP32
0x7002_21CC	AT32M416KBU7-4	128KB	QFN32
0x7002_210D	AT32M416K8U7-4	64KB	QFN32
0x7002_21CE	AT32M416EBP7	128KB	TSSOP24
0x7002_210F	AT32M416E8P7	64KB	TSSOP24

24.4.2 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: 3	保留	0x0000 0000	resd	必须保持为 0。
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。

24.4.3 DEBUG APB1暂停控制寄存器 (DEBUG_APB1_PAUSE)

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	0x0	resd	请保持为复位值。
位 27	I2C2_SMBUS_TIMEOUT	0x0	rw	I ² C2 暂停控制位。 0: 正常工作； 1: I ² C2 SMBUS 的超时控制停止工作。
位 26	保留	0x0	resd	请保持为复位值。
位 25	CAN1_PAUSE	0x0	rw	CAN1 暂停控制位。 0: CAN1 仍然正常运行； 1: CAN1 的接收寄存器不继续接收数据。
位 24	I2C1_SMBUS_TIMEOUT	0x0	rw	I ² C1 暂停控制位。 0: 正常工作； 1: I ² C1 SMBUS 的超时控制停止工作。
位 23: 13	保留	0x00	resd	请保持为复位值。
位 12	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 11	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 10: 6	保留	0x0	rw	请保持为复位值。
位 5	TMR7_PAUSE	0x0	rw	TMR7 暂停控制位。 0: 定时器正常工作; 1: 定时器停止工作。
位 4	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 3	保留	0x0	rw	请保持为复位值。
位 2	TMR4_PAUSE	0x0	rw	TMR4 暂停控制位。 0: 定时器正常工作; 1: 定时器停止工作。
位 1	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 0	保留	0x0	rw	请保持为复位值。

24.4.4 DEBUG APB2暂停控制寄存器 (DEBUG_APB2_PAUSE)

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: 19	保留	0x0000	resd	请保持为复位值。
位 18	TMR11_PAUSE	0x0	rw	TMR11 暂停控制位 0: 定时器正常工作; 1: 定时器停止工作。
位 17	TMR10_PAUSE	0x0	rw	TMR10 暂停控制位 0: 定时器正常工作; 1: 定时器停止工作。
位 16	TMR9_PAUSE	00	rw	TMR9 暂停控制位 0: 定时器正常工作; 1: 定时器停止工作。
位 15: 1	保留	0x0000	resd	请保持为复位值。
位 0	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。

24.4.5 DEBUG APB3暂停控制寄存器 (DEBUG_APB3_PAUSE)

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: 1	保留	0x0000	resd	请保持为复位值。
位 0	ERTC_PAUSE	0x0	rw	ERTC 暂停控制位。 0: ERTC 正常运行； 1: ERTC 停止工作。

24.4.6 DEBUG SERIES ID寄存器 (DEBUG_SER_ID)

DEBUG_SER_ID 寄存器用于识别 MCU 系列和版本号。被映射到外部 PPB 总线，寄存器由 PORESET 异步复位 (不被系统复位所复位)。使用 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: 16	保留	0x0000	resd	请保持为复位值。
位 15: 8	SER_ID	0xXX	ro	MCU 型号识别标识 AT32M412: 0x19 AT32M416: 0x1A
位 7: 3	保留	0x0000	resd	请保持为复位值。
位 2: 0	REV_ID	0xX	ro	版本标识 0x0: A 版

25 版本历史

日期	版本	变更
2024.09.06	2.00	最初版本

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) 航天应用或航天环境，(D) 武器，且/或(E)其他可能导致人身伤害、死亡及财产损失的应用。如果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，风险及法律责任仍将由采购商单独承担，且采购商应独立负责在前述应用中满足所有法律和法规要求。

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。

© 2024 雅特力科技 保留所有权利