

基于**ARM®32位的Cortex™-M4F**微控制器+FPU，带**256K**字节至**1024K**字节内部闪存、**sLib**、**USB**、以太网、**2个CAN**、**17个**定时器、**3个ADC**、**21个**通信接口

## 功能

### ■ 内核：带有FPU的ARM®32位的Cortex™-M4F CPU

- 最高240 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(FPU)，具有DSP指令集

### ■ 存储器

- 从256 K字节至1024 K字节的内部闪存程序/数据存储区
- sLib：将指定之主存储区设为执行代码安全库区，此区代码仅能调用无法读取
- SPIM接口：额外提供高达16 M字节外部SPI闪存程序数据存储区接口
- 高达96+128 K字节的SRAM
- 带2个片选外部存储器控制器(XMC)，支持复用信号的SRAM/NOR/PSRAM和NAND存储器
- 并行LCD接口，兼容8080/6800模式

### ■ 时钟、复位和电源管理

- 2.6至3.6伏供电和I/O管脚
- 上电/低电压复位(POR/LVR)、电压监测器(PVM)
- 4至25 MHz晶体振荡器
- 内嵌经出厂调校的48 MHz的RC振荡器(25 °C达1%精度，-40 °C至+105 °C达2.5%精度)，带自动时钟校准功能(ACC)
- 内嵌40 kHz的RC振荡器
- 32.768 kHz晶体振荡器

### ■ 低功耗

- 睡眠、深度睡眠和待机模式
- 电池供电域为RTC和42个16位电池供电寄存器供电

### ■ 3个12位A/D转换器，0.5 μs转换时间(多达16个输入通道)

- 转换范围：0至3.6 V
- 三组采样和保持功能
- 温度传感器

### ■ 2个12位D/A转换器

### ■ DMA：14通道DMA控制器

- 支持的外设：定时器、ADC、SDIO、I<sup>2</sup>S、SPI、I<sup>2</sup>C和USART

### ■ 调试模式

- 串行单线调试(SWD)和JTAG接口
- Cortex™-M4F内嵌跟踪模块(ETM™)

### ■ 多达80个快速I/O端口

- 37/51/80个多功能双向的I/O口，所有I/O口可以映像到16个外部中断；几乎所有I/O口可容

忍5V输入信号

- 所有I/O口均为快速I/O，寄存器访问速度最高f<sub>AHB</sub>

### ■ 多达17个定时器

- 多达8个16位通用定时器+2个32位通用定时器，每个定时器最多支持4个用于输入/输出/PWM或脉冲计数的通道并支持编码器模式
- 多达2个16位带死区控制和紧急刹车，用于电机控制的PWM高级控制定时器
- 2个看门狗定时器
- 系统嘀嗒定时器：24位自减型计数器
- 2个16位基本定时器用于驱动DAC

### ■ 多达21个通信接口

- 多达3个I<sup>2</sup>C接口(支持SMBus/PMBus)
- 多达8个USART接口(支持ISO7816，LIN，IrDA接口和调制解调控制)
- 多达4个SPI接口(50 M位/秒)，4个均可复用为I<sup>2</sup>S接口，其中I<sup>2</sup>S2/I<sup>2</sup>S3支持全双工
- 多达2个CAN接口(2.0B主动)
- USB2.0全速设备接口，支持无晶振(crystal-less)
- 多达2个SDIO接口
- 10/100M以太网MAC：具有专用DMA和4 K字节SRAM，支持IEEE 1588硬件，MII/RMII接口

### ■ CRC计算单元

### ■ 96位的芯片唯一代码

### ■ 封装

- LQFP100 14 x 14 mm
- LQFP64 10 x 10 mm
- LQFP48 7 x 7 mm
- QFN48 6 x 6 mm

### ■ 选型列表

内部闪存存储器	型号
1024 K 字节	AT32F403ACGU7, AT32F403ACGT7, AT32F403ARGT7, AT32F403AVGT7, AT32F407RGT7, AT32F407VGT7, AT32F407AVGT7
512 K 字节	AT32F403ACEU7, AT32F403ACET7, AT32F403ARET7, AT32F403AVET7, AT32F407RET7, AT32F407VET7
256 K 字节	AT32F403ACCU7, AT32F403ACCT7, AT32F403ARCT7, AT32F403AVCT7, AT32F407RCT7, AT32F407VCT7, AT32F407AVCT7

## 目 录

1	系统架构 .....	27
1.1	系统概述 .....	29
1.1.1	ARM Cortex™-M4F处理器 .....	29
1.1.2	位带 .....	29
1.1.3	中断和异常向量 .....	31
1.1.4	系统嘀嗒定时器（SysTick） .....	34
1.1.5	复位流程 .....	34
1.2	寄存器描述缩写说明 .....	35
1.3	器件特征信息 .....	36
1.3.1	闪存容量寄存器 .....	36
1.3.2	器件电子签名 .....	36
2	存储器资源 .....	37
2.1	内部存储器地址映射 .....	37
2.2	Flash存储器 .....	37
2.3	SRAM存储器 .....	39
2.4	外设地址映射 .....	39
3	电源控制（PWC） .....	42
3.1	简介 .....	42
3.2	主要特点 .....	42
3.3	上电低电压复位 .....	42
3.4	电压监测器（PVM） .....	43
3.5	电源域划分 .....	43
3.6	省电模式 .....	44
3.7	PWC寄存器 .....	45
3.7.1	电源控制寄存器（PWC_CTRL） .....	45
3.7.2	电源控制及状态寄存器（PWC_CTRLSTS） .....	46
4	时钟和复位管理（CRM） .....	47
4.1	时钟 .....	47
4.1.1	时钟源 .....	47
4.1.2	系统时钟 .....	48
4.1.3	外设时钟 .....	48
4.1.4	时钟失效检测 .....	48
4.1.5	自动滑顺频率切换 .....	48
4.1.6	内部时钟输出 .....	49
4.1.7	中断 .....	49
4.2	复位 .....	49
4.2.1	系统复位 .....	49
4.2.2	电池供电域复位 .....	49
4.3	CRM寄存器 .....	50
4.3.1	时钟控制寄存器（CRM_CTRL） .....	50
4.3.2	时钟配置寄存器（CRM_CFG） .....	51

4.3.3	时钟中断寄存器 (CRM_CLKINT)	53
4.3.4	APB2外设复位寄存器 (CRM_APB2RST)	54
4.3.5	APB1外设复位寄存器 (CRM_APB1RST)	56
4.3.6	AHB外设时钟使能寄存器 (CRM_AHBEN)	57
4.3.7	APB2外设时钟使能寄存器 (CRM_APB2EN)	58
4.3.8	APB1外设时钟使能寄存器 (CRM_APB1EN)	60
4.3.9	电池供电域控制寄存器 (CRM_BPDC)	61
4.3.10	控制/状态寄存器 (CRM_CTRLSTS)	62
4.3.11	AHB外设复位寄存器 (CRM_AHBRSR)	63
4.3.12	额外寄存器1 (CRM_MISC1)	63
4.3.13	额外寄存器2 (CRM_MISC2)	64
4.3.14	额外寄存器3 (CRM_MISC3)	64
4.3.15	中断映射寄存器 (CRM_INTMAP)	65
5	闪存控制器 (FLASH)	66
5.1	FLASH介绍	66
5.2	主存储器操作	70
5.2.1	解锁/锁定	70
5.2.2	擦除	71
5.2.3	编程	72
5.2.4	读取	73
5.3	外部存储器操作	74
5.4	用户系统数据区操作	74
5.4.1	解锁/锁定	74
5.4.2	擦除	74
5.4.3	编程	75
5.4.4	读取	76
5.5	闪存保护	76
5.5.1	访问保护	76
5.5.2	擦写保护	77
5.6	特殊功能	77
5.6.1	安全库区设定	77
5.7	FLASH寄存器	78
5.7.1	闪存性能选择寄存器 (FLASH_PSR)	79
5.7.2	闪存解锁寄存器 (FLASH_UNLOCK)	79
5.7.3	闪存用户系统数据解锁寄存器 (FLASH_USD_UNLOCK)	79
5.7.4	闪存状态寄存器 (FLASH_STS)	79
5.7.5	闪存控制寄存器 (FLASH_CTRL)	80
5.7.6	闪存地址寄存器 (FLASH_ADDR)	80
5.7.7	用户系统数据寄存器 (FLASH_USD)	80
5.7.8	擦除编程保护状态寄存器 (FLASH_EPPS)	81
5.7.9	闪存解锁寄存器2 (FLASH_UNLOCK2)	81
5.7.10	闪存状态寄存器2 (FLASH_STS2)	81

5.7.11	闪存控制寄存器2 (FLASH_CTRL2)	81
5.7.12	闪存地址寄存器2 (FLASH_ADDR2)	82
5.7.13	闪存解锁寄存器3 (FLASH_UNLOCK3)	82
5.7.14	闪存选择寄存器 (FLASH_SELECT)	82
5.7.15	闪存状态寄存器3 (FLASH_STS3)	82
5.7.16	闪存控制寄存器3 (FLASH_CTRL3)	83
5.7.17	闪存地址寄存器3 (FLASH_ADDR3)	83
5.7.18	闪存解密地址寄存器 (FLASH_DA)	83
5.7.19	闪存安全库区状态寄存器0 (SLIB_STS0)	83
5.7.20	闪存安全库区状态寄存器1 (SLIB_STS1)	84
5.7.21	闪存安全库区密码清除寄存器 (SLIB_PWD_CLR)	84
5.7.22	闪存安全库区额外状态寄存器 (SLIB_MISC_STS)	84
5.7.23	闪存安全库区密码设定寄存器 (SLIB_SET_PWD)	84
5.7.24	闪存安全库区地址设定寄存器 (SLIB_SET_RANGE)	85
5.7.25	闪存安全库区解锁寄存器 (SLIB_UNLOCK)	85
5.7.26	闪存CRC校验控制寄存器 (FLASH_CRC_CTRL)	85
5.7.27	闪存CRC校验结果寄存器 (FLASH_CRC_CHKR)	86
6	通用功能输入输出 (GPIO)	87
6.1	简介	87
6.2	功能描述	87
6.2.1	GPIO结构	87
6.2.2	GPIO复位状态	87
6.2.3	通用功能输入配置	88
6.2.4	模拟输入/输出配置	88
6.2.5	通用功能输出配置	88
6.2.6	I/O端口保护	88
6.3	GPIO寄存器	88
6.3.1	GPIO配置低寄存器 (GPIOx_CFGLR) (x=A..E)	89
6.3.2	GPIO配置高寄存器 (GPIOx_CFGHR) (A..E)	89
6.3.3	GPIO输入数据寄存器 (GPIOx_IDT) (x=A..E)	90
6.3.4	GPIO输出数据寄存器 (GPIOx_ODT) (x=A..E)	90
6.3.5	GPIO设置/清除寄存器 (GPIOx_SCR) (x=A..E)	90
6.3.6	GPIO清除寄存器 (GPIOx_CLR) (x=A..E)	90
6.3.7	GPIO写保护寄存器 (GPIOx_WPR) (x=A..E)	90
6.3.8	GPIO极大电流推动/吸入能力切换控制寄存器 (GPIOx_HDRV) (x=A..E)	91
7	复用功能输入输出 (IOMUX)	92
7.1	简介	92
7.2	功能描述	92
7.2.1	IOMUX结构	92
7.2.2	复用功能输入配置	92
7.2.3	复用功能输出或双向复用功能配置	93
7.2.4	外设复用功能管脚配置	93

7.2.5	IOMUX映射优先级 .....	93
7.2.5.1	硬件抢占功能 .....	93
7.2.5.2	调试端口优先 .....	94
7.2.5.3	其他外设输出优先级关系 .....	94
7.2.6	外部中断/唤醒线 .....	94
7.3	IOMUX寄存器 .....	94
7.3.1	事件输出控制寄存器 (IOMUX_EVTOUT) .....	95
7.3.2	IO复用重映射寄存器 (IOMUX_REMAP) .....	95
7.3.3	复用外部中断配置寄存器1 (IOMUX_EXINTC1) .....	98
7.3.4	复用外部中断配置寄存器2 (IOMUX_EXINTC2) .....	98
7.3.5	复用外部中断配置寄存器3 (IOMUX_EXINTC3) .....	99
7.3.6	复用外部中断配置寄存器4 (IOMUX_EXINTC4) .....	100
7.3.7	IO复用重映射寄存器2 (IOMUX_REMAP2) .....	100
7.3.8	IO复用重映射寄存器3 (IOMUX_REMAP3) .....	101
7.3.9	IO复用重映射寄存器4 (IOMUX_REMAP4) .....	101
7.3.10	IO复用重映射寄存器5 (IOMUX_REMAP5) .....	102
7.3.11	IO复用重映射寄存器6 (IOMUX_REMAP6) .....	103
7.3.12	IO复用重映射寄存器7 (IOMUX_REMAP7) .....	104
7.3.13	IO复用重映射寄存器8 (IOMUX_REMAP8) .....	105
8	外部中断/事件控制器 (EXINT) .....	106
8.1	EXINT介绍 .....	106
8.2	功能描述和配置流程 .....	106
8.3	EXINT寄存器描述 .....	107
8.3.1	中断使能寄存器 (EXINT_INTEN) .....	107
8.3.2	事件使能寄存器 (EXINT_EVTEN) .....	107
8.3.3	极性配置寄存器1 (EXINT_POLCFG1) .....	107
8.3.4	极性配置寄存器2 (EXINT_POLCFG2) .....	108
8.3.5	软件触发寄存器 (EXINT_SWTRG) .....	108
8.3.6	中断状态寄存器 (EXINT_INTSTS) .....	108
9	DMA控制器 (DMA) .....	109
9.1	简介 .....	109
9.2	特性 .....	109
9.3	功能描述 .....	109
9.3.1	通道配置 .....	109
9.3.2	握手机制 .....	110
9.3.3	仲裁 .....	110
9.3.4	可编程数据传输宽度 .....	110
9.3.5	错误事件 .....	111
9.3.6	中断 .....	111
9.3.7	DMA固定请求映射 .....	112
9.3.8	DMA弹性请求映射 .....	112
9.4	DMA寄存器 .....	114

9.4.1	DMA状态寄存器 (DMA_STS) .....	115
9.4.2	DMA状态清除寄存器 (DMA_CLR) .....	117
9.4.3	DMA通道x配置寄存器 (DMA_CxCTRL) (x = 1...7) .....	119
9.4.4	DMA通道x数据传输量寄存器 (DMA_CxDTCNT) (x = 1...7) .....	120
9.4.5	DMA通道x外设地址寄存器 (DMA_CxPADDR) (x = 1...7) .....	120
9.4.6	DMA通道x存储器地址寄存器 (DMA_CxMADDR) (x = 1...7) .....	120
9.4.7	通道来源寄存器0 (DMA_SRC_SELO) .....	120
9.4.8	通道来源寄存器1 (DMA_SRC_SEL1) .....	121
10	CRC计算单元 (CRC) .....	122
10.1	CRC介绍 .....	122
10.2	CRC寄存器 .....	122
10.2.1	数据寄存器 (CRC_DT) .....	122
10.2.2	通用数据寄存器 (CRC_CDT) .....	122
10.2.3	控制寄存器 (CRC_CTRL) .....	123
10.2.4	初始化寄存器 (CRC_IDT) .....	123
11	I <sup>2</sup> C接口 .....	124
11.1	I <sup>2</sup> C简介 .....	124
11.2	I <sup>2</sup> C主要特点 .....	124
11.3	I <sup>2</sup> C总线特性 .....	124
11.4	I <sup>2</sup> C接口 .....	124
11.4.1	I <sup>2</sup> C从机通信流程 .....	127
11.4.2	I <sup>2</sup> C主机通信流程 .....	128
11.4.3	利用DMA传输 .....	133
11.4.4	SMBus .....	134
11.4.5	I <sup>2</sup> C中断请求 .....	136
11.4.6	I <sup>2</sup> C调试模式 .....	136
11.5	I <sup>2</sup> C寄存器描述 .....	136
11.5.1	控制寄存器1(I2C_CTRL1) .....	137
11.5.2	控制寄存器2(I2C_CTRL2) .....	138
11.5.3	自身地址寄存器1(I2C_OADDR1) .....	139
11.5.4	自身地址寄存器2(I2C_OADDR2) .....	139
11.5.5	数据寄存器(I2C_DT) .....	139
11.5.6	状态寄存器1(I2C_STS1) .....	139
11.5.7	状态寄存器2(I2C_STS2) .....	141
11.5.8	时钟控制寄存器(I2C_CLKCTRL) .....	142
12	通用同步异步收发器 (USART) .....	143
12.1	USART介绍 .....	143
12.2	全双工半双工选择器简述和配置流程 .....	144
12.3	模式选择器简述和配置流程 .....	144
12.3.1	模式选择器简述 .....	144
12.3.2	模式选择器配置方法 .....	145
12.4	USART帧格式简述和配置流程 .....	145
12.5	DMA传输简述和配置流程 .....	145

12.5.1	DMA发送配置流程 .....	145
12.5.2	DMA接收配置流程 .....	146
12.6	波特率发生器简述及配置流程 .....	146
12.6.1	波特率发生器简述 .....	146
12.6.2	波特率发生器配置方法 .....	146
12.7	发送器简述和配置流程 .....	147
12.7.1	发送器简述 .....	147
12.7.2	发送器配置流程 .....	147
12.8	接收器简述和配置流程 .....	147
12.8.1	接收器简述 .....	147
12.8.2	接收器配置流程 .....	148
12.8.3	起始侦测和噪声检测 .....	148
12.9	中断 .....	149
12.10	I/O管脚控制 .....	150
12.11	USART寄存器描述 .....	150
12.11.1	状态寄存器 ( USART_STS ) .....	151
12.11.2	数据寄存器 ( USART_DT ) .....	152
12.11.3	波特比率寄存器 ( USART_BAUDR ) .....	152
12.11.4	控制寄存器1 ( USART_CTRL1 ) .....	152
12.11.5	控制寄存器2 ( USART_CTRL2 ) .....	153
12.11.6	控制寄存器3 ( USART_CTRL3 ) .....	154
12.11.7	保护时间和预分频寄存器 ( USART_GDIV ) .....	155
13	串行外设接口 ( SPI ) .....	156
13.1	串行外设接口 ( SPI ) 简介 .....	156
13.2	SPI功能描述 .....	156
13.2.1	SPI简述 .....	156
13.2.2	全双工半双工选择器简述和配置流程 .....	157
13.2.3	CS控制器简述和配置流程 .....	158
13.2.4	SPI_SCK控制器简述和配置流程 .....	159
13.2.5	CRC简述和配置流程 .....	159
13.2.6	DMA传输简述和配置流程 .....	160
13.2.7	发送器简述和配置流程 .....	160
13.2.8	接收器简述和配置流程 .....	161
13.2.9	中断 .....	162
13.2.10	IO管脚控制 .....	162
13.2.11	注意事项 .....	162
13.3	I <sup>2</sup> S功能描述 .....	162
13.3.1	I <sup>2</sup> S简述 .....	162
13.3.2	I <sup>2</sup> S 全双工 .....	163
13.3.3	操作模式选择器简述和配置流程 .....	164
13.3.4	音频协议选择器简述和配置流程 .....	165
13.3.5	I <sup>2</sup> S_CLK控制器简述和配置流程 .....	166



13.3.6	DMA传输简述和配置流程.....	168
13.3.7	发送器接收器简述和配置流程 .....	169
13.3.8	中断 .....	170
13.3.9	I0管脚控制 .....	170
13.4	SPI寄存器 .....	170
13.4.1	SPI控制寄存器1 (SPI_CTRL1) (I <sup>2</sup> S模式下不使用) .....	171
13.4.2	SPI控制寄存器2 (SPI_CTRL2) .....	172
13.4.3	SPI状态寄存器 (SPI_STS) .....	172
13.4.4	SPI数据寄存器 (SPI_DT) .....	173
13.4.5	SPICRC多项式寄存器 (SPI_CPOLY) (I <sup>2</sup> S模式下不使用) .....	173
13.4.6	SPIRxCRC寄存器 (SPI_RCRC) (I <sup>2</sup> S模式下不使用) .....	173
13.4.7	SPITxCRC寄存器 (SPI_TCRC) .....	173
13.4.8	SPI_I2S配置寄存器 (SPI_I2SCTRL) .....	174
13.4.9	SPI_I2S预分频寄存器 (SPI_I2SCLKP) .....	174
14	定时器 (TIMER) .....	175
14.1	基本定时器 (TMR6和TMR7) .....	175
14.1.1	TMR6和TMR7简介 .....	175
14.1.2	TMR6和TMR7的主要特性.....	176
14.1.3	TMR6和TMR7的功能.....	176
14.1.3.1	计数时钟 .....	176
14.1.3.2	计数模式 .....	176
14.1.3.3	调试模式 .....	177
14.1.4	TMR6和TMR7寄存器.....	177
14.1.4.1	TMR6 和TMR7控制寄存器1 (TMRx_CTRL1) .....	177
14.1.4.2	TMR6和TMR7控制寄存器2 (TMRx_CTRL2) .....	178
14.1.4.3	TMR6和TMR7 DMA/中断使能寄存器 (TMRx_IDEN) .....	178
14.1.4.4	TMR6和TMR7中断状态寄存器 (TMRx_ISTS) .....	178
14.1.4.5	TMR6和TMR7软件事件寄存器 (TMRx_SWEVT) .....	178
14.1.4.6	TMR6和TMR7计数值 (TMRx_CVAL) .....	179
14.1.4.7	TMR6和TMR7分频系数 (TMRx_DIV) .....	179
14.1.4.8	TMR6和TMR7周期寄存器 (TMRx_PR) .....	179
14.2	通用定时器 (TMR2到TMR5) .....	180
14.2.1	TMRx简介 .....	180
14.2.2	TMRx主要功能 .....	180
14.2.3	TMRx功能描述 .....	180
14.2.3.1	计数时钟 .....	180
14.2.3.2	计数模式 .....	182
14.2.3.3	TMR输入部分 .....	184
14.2.3.4	TMR输出部分 .....	185
14.2.3.5	定时器同步 .....	187
14.2.3.6	调试模式 .....	190
14.2.4	TMRx寄存器描述.....	190



14.2.4.1	控制寄存器1 (TMRx_CTRL1)	190
14.2.4.2	控制寄存器2 (TMRx_CTRL2)	191
14.2.4.3	次定时器控制寄存器 (TMRx_STCTRL)	192
14.2.4.4	DMA/中断使能寄存器 (TMRx_IDEN)	193
14.2.4.5	中断状态寄存器 (TMRx_ISTS)	194
14.2.4.6	软件事件寄存器 (TMRx_SWEVT)	194
14.2.4.7	通道模式寄存器1 (TMRx_CM1)	195
14.2.4.8	通道模式寄存器2 (TMRx_CM2)	197
14.2.4.9	通道控制寄存器 (TMRx_CCTRL)	198
14.2.4.10	计数值 (TMRx_CVAL)	198
14.2.4.11	分频系数 (TMRx_DIV)	198
14.2.4.12	周期寄存器 (TMRx_PR)	199
14.2.4.13	通道1数据寄存器 (TMRx_C1DT)	199
14.2.4.14	通道2数据寄存器 (TMRx_C2DT)	199
14.2.4.15	通道3数据寄存器 (TMRx_C3DT)	199
14.2.4.16	通道4数据寄存器 (TMRx_C4DT)	200
14.2.4.17	DMA控制寄存器 (TMRx_DMACTRL)	200
14.2.4.18	DMA数据寄存器 (TMRx_DMADT)	200
14.3	通用定时器 (TMR9到TMR14)	201
14.3.1	TMRx简介	201
14.3.2	TMRx主要特性	201
14.3.2.1	TMR9和TMR12主要特性	201
14.3.2.2	TMR10、TMR11、TMR13和 TMR14主要特性	201
14.3.3	TMRx功能描述	202
14.3.3.1	计数时钟	202
14.3.3.2	计数模式	203
14.3.3.3	TMR输入部分	204
14.3.3.4	TMR输出部分	205
14.3.3.5	TMR同步	206
14.3.3.6	调试模式	207
14.3.4	TMR9和TMR12寄存器描述	208
14.3.4.1	控制寄存器 1 (TMRx_CTRL1)	208
14.3.4.2	次定时器控制寄存器 (TMRx_STCTRL)	209
14.3.4.3	DMA/中断使能寄存器 (TMRx_IDEN)	209
14.3.4.4	中断状态寄存器 (TMRx_ISTS)	209
14.3.4.5	软件事件寄存器 (TMRx_SWEVT)	210
14.3.4.6	通道模式寄存器1 (TMRx_CM1)	211
14.3.4.7	通道控制寄存器 (TMRx_CCTRL)	213
14.3.4.8	计数器 (TMRx_CVAL)	213
14.3.4.9	预分频器 (TMRx_DIV)	213
14.3.4.10	周期寄存器 (TMRx_PR)	213
14.3.4.11	通道1数据寄存器 (TMRx_C1DT)	213

14.3.4.12	通道2数据寄存器 (TMRx_C2DT)	214
14.3.5	TMR10、TMR11、TMR13和TMR14寄存器描述	214
14.3.5.1	控制寄存器1 (TMRx_CTRL1)	214
14.3.5.2	DMA/中断使能寄存器 (TMRx_IDEN)	215
14.3.5.3	中断状态寄存器 (TMRx_ISTS)	215
14.3.5.4	软件事件寄存器 (TMRx_SWEVT)	215
14.3.5.5	通道模式寄存器1 (TMRx_CM1)	216
14.3.5.6	通道控制寄存器 (TMRx_CCTRL)	218
14.3.5.7	计数值 (TMRx_CVAL)	218
14.3.5.8	预分频器 (TMRx_DIV)	218
14.3.5.9	周期寄存器 (TMRx_PR)	218
14.3.5.10	通道1数据寄存器 (TMRx_C1DT)	218
14.4	高级控制定时器 (TMR1、TMR8)	219
14.4.1	TMR1、TMR8简介	219
14.4.2	TMR1、TMR8主要特性	219
14.4.3	TMR1、TMR8功能描述	219
14.4.3.1	计数时钟	219
14.4.3.2	计数模式	221
14.4.3.3	TMR输入部分	223
14.4.3.4	TMR输出部分	224
14.4.3.5	TMR刹车功能	227
14.4.3.6	TMR同步	228
14.4.3.7	调试模式	229
14.4.4	TMR1、TMR8寄存器描述	229
14.4.4.1	TMR1、TMR8 控制寄存器1 (TMRx_CTRL1)	230
14.4.4.2	TMR1、TMR8控制寄存器2 (TMRx_CTRL2)	231
14.4.4.3	TMR1、TMR8次定时器控制寄存器 (TMRx_STCTRL)	231
14.4.4.4	TMR1、TMR8 DMA/中断使能寄存器 (TMRx_IDEN)	233
14.4.4.5	TMR1、TMR8中断状态寄存器 (TMRx_ISTS)	234
14.4.4.6	TMR1、TMR8软件事件寄存器 (TMRx_SWEVT)	235
14.4.4.7	TMR1、TMR8通道模式寄存器1 (TMRx_CM1)	235
14.4.4.8	TMR1、TMR8通道模式寄存器2 (TMRx_CM2)	238
14.4.4.9	TMR1、TMR8通道控制寄存器 (TMRx_CCTRL)	239
14.4.4.10	TMR1、TMR8计数值 (TMRx_CVAL)	241
14.4.4.11	TMR1、TMR8预分频器 (TMRx_DIV)	241
14.4.4.12	TMR1、TMR8周期寄存器 (TMRx_PR)	241
14.4.4.13	TMR1、TMR8重复周期寄存器 (TMRx_RPR)	241
14.4.4.14	TMR1、TMR8通道1数据寄存器 (TMRx_C1DT)	241
14.4.4.15	TMR1、TMR8通道2数据寄存器 (TMRx_C2DT)	241
14.4.4.16	TMR1、TMR8通道3数据寄存器 (TMRx_C3DT)	241
14.4.4.17	TMR1、TMR8通道4数据寄存器 (TMRx_C4DT)	242
14.4.4.18	TMR1、TMR8刹车寄存器 (TMRx_BRK)	242

14.4.4.19	TMR1、TMR8 DMA控制寄存器 (TMRx_DMACTRL)	243
14.4.4.20	TMR1、TMR8 DMA数据寄存器 (TMRx_DMADT)	243
15	窗口看门狗 (WWDT)	244
15.1	WWDT简介	244
15.2	WWDT主要特性	244
15.3	WWDT功能描述	244
15.4	调试模式	245
15.5	WWDT寄存器	245
15.5.1	控制寄存器 (WWDT_CTRL)	245
15.5.2	配置寄存器 (WWDT_CFG)	246
15.5.3	状态寄存器 (WWDT_STS)	246
16	看门狗 (WDT)	247
16.1	WDT简介	247
16.2	WDT主要特性	247
16.3	WDT功能描述	247
16.4	调试模式	248
16.5	WDT寄存器	248
16.5.1	命令寄存器 (WDT_CMD)	248
16.5.2	预分频寄存器 (WDT_DIV)	249
16.5.3	重装载寄存器 (WDT_RLD)	249
16.5.4	状态寄存器 (WDT_STS)	249
17	实时时钟 (RTC)	250
17.1	RTC简介	250
17.2	主要特性	250
17.3	RTC架构	250
17.4	RTC功能描述	251
17.4.1	RTC寄存器配置	251
17.4.2	RTC寄存器读取	251
17.4.3	RTC中断	251
17.5	RTC寄存器描述	252
17.5.1	RTC控制寄存器高位 (RTC_CTRLH)	253
17.5.2	RTC控制寄存器低位 (RTC_CTRLH)	253
17.5.3	RTC分频系数寄存器 (RTC_DIVH/RTC_DIVL)	254
17.5.4	RTC分频计数寄存器 (RTC_DIVCNTH/RTC_DIVCNTL)	254
17.5.5	RTC计数值寄存器 (RTC_CNTH/RTC_CNTL)	254
17.5.6	RTC闹钟寄存器 (RTC_TAH/RTC_TAL)	254
18	电池供电寄存器 (BPR)	255
18.1	BPR简介	255
18.2	BPR特性	255
18.3	BPR功能描述	255
18.4	BPR寄存器描述	255
18.4.1	电池供电数据寄存器x (BPR_DT <sub>x</sub> ) (x = 1 ... 42)	257
18.4.2	RTC校准寄存器 (BPR_RTCCAL)	257

18.4.3	电池供电控制寄存器 (BPR_CTRL)	257
18.4.4	电池供电控制/状态寄存器 (BPR_CTRLSTS)	258
19	模拟/数字转换 (ADC)	259
19.1	ADC简介	259
19.2	ADC主要特征	259
19.3	ADC架构	259
19.4	ADC功能介绍	260
19.4.1	通道管理	260
19.4.1.1	内部温度传感器	261
19.4.1.2	内部参考电压	261
19.4.2	ADC操作流程	261
19.4.2.1	上电与校准	261
19.4.2.2	触发	262
19.4.2.3	采样与转换时序	263
19.4.3	转换顺序管理	263
19.4.3.1	序列模式	263
19.4.3.2	抢占自动转换模式	264
19.4.3.3	反复模式	264
19.4.3.4	分割模式	265
19.4.4	数据管理	265
19.4.4.1	数据内容处理	265
19.4.4.2	数据获取	265
19.4.5	电压监测	266
19.4.6	状态标志与中断	266
19.5	主从模式	266
19.5.1	数据管理	267
19.5.2	同时模式	267
19.5.3	抢占交错触发模式	267
19.5.4	普通位移模式	268
19.6	ADC寄存器	269
19.6.1	ADC状态寄存器 (ADC_STS)	270
19.6.2	ADC控制寄存器1 (ADC_CTRL1)	270
19.6.3	ADC控制寄存器2 (ADC_CTRL2)	272
19.6.4	ADC采样时间寄存器1 (ADC_SPT1)	275
19.6.5	ADC采样时间寄存器2 (ADC_SPT2)	277
19.6.6	ADC抢占通道数据偏移寄存器x (ADC_PCDTOx) (x=1..4)	279
19.6.7	ADC电压监测高边界寄存器 (ADC_VMHB)	279
19.6.8	ADC电压监测低边界寄存器 (ADC_VMLB)	279
19.6.9	ADC普通序列寄存器1 (ADC_OSQ1)	279
19.6.10	ADC普通序列寄存器2 (ADC_OSQ2)	280
19.6.11	ADC普通序列寄存器3 (ADC_OSQ3)	280
19.6.12	ADC抢占序列寄存器 (ADC_PSQ)	281

19.6.13	ADC抢占数据寄存器x (ADC_PDTx) (x= 1..4)	281
19.6.14	ADC普通数据寄存器 (ADC_ODT)	281
20	数字/模拟转换 (DAC)	282
20.1	简介	282
20.2	主要特性	282
20.3	设计提示	282
20.4	功能描述	283
20.4.1	触发事件	283
20.4.2	噪声/三角波生成	283
20.4.3	数据配置	284
20.5	DAC寄存器	285
20.5.1	DAC控制寄存器 (DAC_CTRL)	285
20.5.2	DAC软件触发寄存器 (DAC_SWTRG)	287
20.5.3	DAC1的12位右对齐数据保持寄存器 (DAC_D1DTH12R)	287
20.5.4	DAC1的12位左对齐数据保持寄存器 (DAC_D1DTH12L)	287
20.5.5	DAC1的8位右对齐数据保持寄存器 (DAC_D1DTH8R)	287
20.5.6	DAC2的12位右对齐数据保持寄存器 (DAC_D2DTH12R)	287
20.5.7	DAC2的12位左对齐数据保持寄存器 (DAC_D2DTH12L)	288
20.5.8	DAC2的8位右对齐数据保持寄存器 (DAC_D2DTH8R)	288
20.5.9	双DAC的12位右对齐数据保持寄存器 (DAC_DDTH12R)	288
20.5.10	双DAC的12位左对齐数据保持寄存器 (DAC_DDTH12L)	288
20.5.11	双DAC的8位右对齐数据保持寄存器 (DAC_DDTH8R)	288
20.5.12	DAC1数据输出寄存器 (DAC_D1ODT)	288
20.5.13	DAC2数据输出寄存器 (DAC_D2ODT)	288
21	CAN总线控制器	289
21.1	简介	289
21.2	主要特性	289
21.3	波特率设置	289
21.4	中断管理	292
21.5	设计提示	292
21.6	功能描述	293
21.6.1	整体功能描述	293
21.6.2	工作模式	293
21.6.3	测试方法	294
21.6.4	报文过滤	294
21.6.5	报文发送	297
21.6.6	报文接收	298
21.6.7	出错管理	298
21.7	CAN寄存器	299
21.7.1	CAN控制和状态寄存器	300
21.7.1.1	CAN主控制寄存器 (CAN_MCTRL)	300
21.7.1.2	CAN主状态寄存器 (CAN_MSTS)	301

21.7.1.3	CAN发送状态寄存器 (CAN_TSTS)	302
21.7.1.4	CAN接收FIFO 0寄存器 (CAN_RF0)	305
21.7.1.5	CAN接收FIFO 1寄存器 (CAN_RF1)	306
21.7.1.6	CAN中断使能寄存器 (CAN_INTEN)	306
21.7.1.7	CAN错误状态寄存器 (CAN_ESTS)	308
21.7.1.8	CAN位时序寄存器 (CAN_BTMG)	309
21.7.2	CAN邮箱寄存器	309
21.7.2.1	发送邮箱标识符寄存器 (CAN_TMIx) (x=0..2)	310
21.7.2.2	发送邮箱数据长度和时间戳寄存器 (CAN_TMCx) (x=0..2)	310
21.7.2.3	发送邮箱低字节数据寄存器 (CAN_TMDTLx) (x=0..2)	310
21.7.2.4	发送邮箱高字节数据寄存器 (CAN_TMDTHx) (x=0..2)	311
21.7.2.5	接收FIFO邮箱标识符寄存器 (CAN_RFIx) (x=0..1)	311
21.7.2.6	接收FIFO邮箱数据长度和时间戳寄存器 (CAN_RFCx) (x=0..1)	311
21.7.2.7	接收FIFO邮箱低字节数据寄存器 (CAN_RFDTLx) (x=0..1)	311
21.7.2.8	接收FIFO邮箱高字节数据寄存器 (CAN_RFDTHx) (x=0..1)	312
21.7.3	CAN过滤器寄存器	312
21.7.3.1	CAN过滤器控制寄存器 (CAN_FCTRL)	312
21.7.3.2	CAN过滤器模式配置寄存器 (CAN_FMCFG)	312
21.7.3.3	CAN过滤器位宽配置寄存器 (CAN_FBWCFG)	312
21.7.3.4	CAN过滤器FIFO关联寄存器 (CAN_FRF)	312
21.7.3.5	CAN过滤器激活控制寄存器 (CAN_FACFG)	313
21.7.3.6	CAN过滤器组i的过滤位寄存器x (CAN_FiFBx) (其中i=0..13; x=1..2)	313
22	外部存储控制器 (XMC)	314
22.1	XMC简介	314
22.2	XMC主要特征	314
22.3	XMC构造	315
22.3.1	框图	315
22.3.2	地址映射	316
22.4	NOR/PSRAM界面	317
22.4.1	操作方式	317
22.4.2	访问模式	318
22.4.2.1	读写相同时序的模式	318
22.4.2.2	读写不同时序的模式	321
22.4.2.3	复用模式	330
22.4.2.4	同步模式	331
22.5	NAND界面	334
22.5.1	操作方式	334
22.5.2	访问时序	334
22.5.3	ECC运算	336
22.6	XMC寄存器	336
22.6.1	NOR闪存和PSRAM控制器寄存器	337

22.6.1.1	SRAM/NOR闪存片选控制寄存器1 (XMC_BK1CTRL1)	337
22.6.1.2	SRAM/NOR闪存片选控制寄存器4 (XMC_BK1CTRL4)	338
22.6.1.3	SRAM/NOR闪存片选时序寄存器1,4 (XMC_BK1TMG1,4)	339
22.6.1.4	SRAM/NOR闪存写时序寄存器1,4 (XMC_BK1TMGWR1,4)	340
22.6.1.5	SRAM/NOR额外时序寄存器1,4 (XMC_EXT1,4)	341
22.6.2	NAND闪存控制器寄存器	341
22.6.2.1	NAND闪存控制寄存器2 (XMC_BK2CTRL)	341
22.6.2.2	中断使能和FIFO状态寄存器2 (XMC_BK2IS)	342
22.6.2.3	常规空间时序寄存器2 (XMC_BK2TMGRG)	342
22.6.2.4	特殊空间时序寄存器2 (XMC_BK2TMGSP)	343
22.6.2.5	ECC结果寄存器2 (XMC_BK2ECC)	343
23	SDIO接口	344
23.1	简介	344
23.2	主要特点	344
23.3	功能描述	346
23.3.1	卡功能描述	346
23.3.1.1	卡识别模式	346
23.3.1.2	数据传输模式	346
23.3.1.3	擦除	347
23.3.1.4	保护管理	348
23.3.2	命令与响应	350
23.3.2.1	命令	350
23.3.2.2	响应格式	353
23.3.3	SDIO功能描述	355
23.3.3.1	SDIO适配器	356
23.3.3.2	数据BUF	359
23.3.3.3	SDIO AHB接口	360
23.3.3.4	硬件流控制	360
23.3.4	SDIO I/O卡特定的操作	360
23.4	SDIO寄存器	361
23.4.1	SDIO电源控制寄存器 (SDIO_PWRCTRL)	362
23.4.2	SDIO时钟控制寄存器 (SDIO_CLKCTRL)	362
23.4.3	SDIO参数寄存器 (SDIO_ARG)	363
23.4.4	SDIO命令寄存器 (SDIO_CMD)	363
23.4.5	SDIO命令响应寄存器 (SDIO_RSPCMD)	364
23.4.6	SDIO响应1.4寄存器 (SDIO_RSPx)	364
23.4.7	SDIO数据定时器寄存器 (SDIO_DTTMR)	364
23.4.8	SDIO数据长度寄存器 (SDIO_DTLEN)	364
23.4.9	SDIO数据控制寄存器 (SDIO_DTCTRL)	365
23.4.10	SDIO数据计数器寄存器 (SDIO_DTCNTR)	366
23.4.11	SDIO状态寄存器 (SDIO_STS)	366
23.4.12	SDIO清除中断寄存器 (SDIO_INTCLR)	367



23.4.13	SDIO中断屏蔽寄存器 (SDIO_INTEN)	368
23.4.14	SDIOBUF计数器寄存器 (SDIO_BUFCNTR)	370
23.4.15	SDIO数据BUF寄存器 (SDIO_BUF)	370
24	通用串行总线全速设备接口 (USBFS)	371
24.1	简介	371
24.2	USBFS时钟与管脚配置	371
24.2.1	USB时钟配置	371
24.2.2	USB管脚配置	371
24.3	USBFS功能描述	371
24.3.1	USB初始化配置	371
24.3.2	端点配置	371
24.3.3	USB缓冲区	372
24.3.4	双缓冲端点配置	373
24.3.5	SOF输出	373
24.3.6	挂起/恢复	374
24.4	USB中断	374
24.5	USBFS寄存器	374
24.5.1	USBFS端点n寄存器 (USBFS_EPTn), n=[0..7]	375
24.5.2	USBFS控制寄存器 (USBFS_CTRL)	376
24.5.3	USBFS中断状态寄存器 (USBFS_INTSTS)	377
24.5.4	USBFS SOF帧编号寄存器 (USBFS_SOFRNUM)	377
24.5.5	USBFS设备地址寄存器 (USBFS_DEVADDR)	378
24.5.6	USBFS分组缓冲区描述表地址寄存器 (USBFS_BUFTBL)	378
24.5.7	USBFS CFG控制寄存器 (USBFS_CFG)	378
24.5.8	USBFS发送缓冲区首地址寄存器 n (USBFS_TnADDR)	378
24.5.9	USBFS发送数据长度寄存器 n (USBFS_TnLEN)	378
24.5.10	USBFS接收缓冲区首地址寄存器 n (USBFS_RnADDR)	378
24.5.11	USBFS接收数据字节数寄存器 n (USBFS_RnLEN)	379
25	HICK自动时钟校准 (ACC)	380
25.1	简介	380
25.2	主要特性	380
25.3	中断请求	380
25.4	功能概述	380
25.5	原理分析	381
25.6	寄存器描述	382
25.6.1	状态寄存器 (ACC_STS)	383
25.6.2	控制寄存器1 (ACC_CTRL1)	383
25.6.3	控制寄存器2 (ACC_CTRL2)	384
25.6.4	比较值1 (ACC_C1)	384
25.6.5	比较值2 (ACC_C2)	384
25.6.6	比较值3 (ACC_C3)	384
26	以太网控制器 (EMAC)	385
26.1	EMAC简介	385

26.1.1	EMAC总体结构图 .....	385
26.1.2	EMAC主要特征 .....	385
26.2	EMAC模块功能详述 .....	386
26.2.1	EMAC通信接口介绍 .....	386
26.2.2	EMAC帧通信 .....	390
26.2.3	专用DMA对以太网帧的传输调度 .....	396
26.2.4	EMAC掉电模式进入及唤醒 .....	405
26.2.5	IEEE1588定义的精确时间协议 .....	407
26.2.6	EMAC中断 .....	410
26.3	EMAC寄存器 .....	411
26.3.1	以太网MAC配置寄存器(EMAC_MACCTRL) .....	412
26.3.2	以太网MAC帧过滤器寄存器(EMAC_MACFRMF) .....	414
26.3.3	以太网MAC Hash列表高寄存器(EMAC_MACHTH) .....	416
26.3.4	以太网MAC Hash列表低寄存器(EMAC_MACHTL) .....	416
26.3.5	以太网MAC MII地址寄存器(EMAC_MACMIIADDR) .....	416
26.3.6	以太网MAC MII数据寄存器(EMAC_MACMIIDT) .....	417
26.3.7	以太网MAC流控寄存器(EMAC_MACFCTRL) .....	417
26.3.8	以太网MAC VLAN标签寄存器(EMAC_MACVLT) .....	419
26.3.9	以太网MAC远程唤醒帧过滤器寄存器(EMAC_MACRWFF) .....	419
26.3.10	以太网MAC PMT控制和状态寄存(EMAC_MACPMTCTRLSTS) .....	420
26.3.11	以太网MAC中断状态寄存器(EMAC_MACISTS) .....	420
26.3.12	以太网MAC中断屏蔽寄存器(EMAC_MAIMR) .....	421
26.3.13	以太网MAC地址0高寄存器(EMAC_MACA0H) .....	421
26.3.14	以太网MAC地址0低寄存器(EMAC_MACA0L) .....	421
26.3.15	以太网MAC地址1高寄存器(EMAC_MACA1H) .....	422
26.3.16	以太网MAC地址1低寄存器(EMAC_MACA1L) .....	422
26.3.17	以太网MAC地址2高寄存器(EMAC_MACA2H) .....	423
26.3.18	以太网MAC地址2低寄存器(EMAC_MACA2L) .....	423
26.3.19	以太网MAC地址3高寄存器(EMAC_MACA3H) .....	424
26.3.20	以太网MAC地址3低寄存器(EMAC_MACA3L) .....	424
26.3.21	以太网DMA总线模式寄存器(EMAC_DMABM) .....	425
26.3.22	以太网DMA发送轮询请求寄存器(EMAC_DMATPD) .....	426
26.3.23	以太网DMA接收轮询请求寄存器(EMAC_DMARPD) .....	426
26.3.24	以太网DMA接收描述符列表地址寄存器(EMAC_DMARDLADDR) .....	427
26.3.25	以太网DMA发送描述符列表地址寄存器(EMAC_DMATDLADDR) .....	427
26.3.26	以太网DMA状态寄存器(EMAC_DMASTS) .....	427
26.3.27	以太网DMA工作模式寄存器(EMAC_DMAOPM) .....	430
26.3.28	以太网DMA中断使能寄存器(EMAC_DMAIE) .....	432
26.3.29	以太网DMA丢失帧和缓存溢出计数器寄存器(EMAC_DMAMFBOCNT) .....	433
26.3.30	以太网DMA当前发送描述符寄存器(EMAC_DMACTD) .....	433
26.3.31	以太网DMA当前接收描述符寄存器(EMAC_DMACRD) .....	434
26.3.32	以太网DMA当前发送缓存地址寄存器(EMAC_DMACTBADDR) .....	434

26.3.33	以太网DMA当前接收缓存地址寄存器(EMAC_DMABADDR)	434
26.3.34	以太网MMC控制寄存器(EMAC_MMCTRL)	434
26.3.35	以太网MMC接收中断寄存器(EMAC_MMCR)	435
26.3.36	以太网MMC发送中断寄存器(EMAC_MMCTI)	435
26.3.37	以太网MMC接收中断屏蔽寄存器(EMAC_MMCRIM)	436
26.3.38	以太网MMC发送中断屏蔽寄存器(EMAC_MMCTIM)	436
26.3.39	以太网MMC 1次冲突后发送”好”帧的计数器寄存器(EMAC_MMCTFSCC)	436
26.3.40	以太网MMC 1次以上冲突后发送”好”帧的计数器寄存器(EMAC_MMCTFSCC)	437
26.3.41	以太网MMC发送”好”帧计数器寄存器(EMAC_MMCTFCNT)	437
26.3.42	以太网MMC CRC错误接收帧计数器寄存器(EMAC_MMCRFCECNT)	437
26.3.43	以太网MMC对齐错误接收帧计数器寄存器(EMAC_MMCRFAECNT)	437
26.3.44	以太网MMC 接收帧”好”单播帧计数器寄存器(EMAC_MMCRGUFCNT)	437
26.3.45	以太网PTP时间戳控制寄存器(EMAC_PTPTCTRL)	438
26.3.46	以太网PTP亚秒递增寄存器(EMAC_PTPSSINC)	440
26.3.47	以太网PTP时间戳高寄存器(EMAC_PTPTSH)	440
26.3.48	以太网PTP时间戳低寄存器(EMAC_PTPTSL)	440
26.3.49	以太网PTP时间戳高更新寄存器(EMAC_PTPTSHUD)	440
26.3.50	以太网PTP时间戳低更新寄存器(EMAC_PTPTSLUD)	441
26.3.51	以太网PTP时间戳低更新寄存器(EMAC_PTPTSLUD)	441
26.3.52	以太网PTP时间戳加数寄存器(EMAC_PTPTSAD)	441
26.3.53	以太网PTP目标时间高寄存器(EMAC_PTPTTH)	441
26.3.54	以太网PTP目标时间低寄存器(EMAC_PTPTTL)	441
26.3.55	以太网 PTP 时间戳状态寄存器 (EMAC_PTPTSSR)	442
26.3.56	以太网 PTP PPS 控制寄存器 (EMAC_PTPTPPSCR)	442
27	调试 (DEBUG)	443
27.1	简介	443
27.2	调试与跟踪功能	443
27.3	I/O控制	443
27.4	DEBUG寄存器	444
27.4.1	DEBUG设备ID (DEBUG_IDCODE)	444
27.4.2	DEBUG控制寄存器 (DEBUG_CTRL)	445
28	版本历史	447

## 图目录

图 1-1 AT32F403A/407/407A 系列微控制器系统架构.....	28
图 1-2 Cortex™-M4F 内部框图 .....	29
图 1-3 位带区与位带别名区的膨胀关系图 A.....	29
图 1-4 位带区与位带别名区的膨胀关系图 B.....	30
图 1-5 复位流程 .....	34
图 1-6 MSP 及 PC 初始化的一个范例 .....	35
图 2-1 AT32F403A/407/407A 地址映射 .....	37
图 3-1 各电源域框图.....	42
图 3-2 上电/低电压复位波形图 .....	43
图 3-3 PVM 的阈值与输出 .....	43
图 4-1 AT32F403A/407/407A 时钟结构图.....	47
图 4-2 系统复位电路.....	49
图 5-1 外部存储器密文保护 .....	67
图 5-2 外部存储器参考电路 .....	68
图 5-3 主存储器扇区擦除流程 .....	71
图 5-4 主存储器整片擦除流程 .....	72
图 5-5 主存储器编程流程 .....	73
图 5-6 系统数据区擦除 .....	75
图 5-7 系统数据区编程 .....	76
图 6-1 GPIO 基本结构 .....	87
图 7-1 IOMUX 基本结构 .....	92
图 8-1 外部中断/事件控制器框图.....	106
图 9-1 DMA 框图 .....	109
图 9-2 请求/应答对后重新仲裁 .....	110
图 9-3 PWIDTH: byte, MWIDTH: half-word .....	111
图 9-4 PWIDTH: half-word, MWIDTH: word .....	111
图 9-5 PWIDTH: word, MWIDTH: byte .....	111
图 11-1 I <sup>2</sup> C 总线协议 .....	124
图 11-2 I <sup>2</sup> C 的功能框图.....	125
图 11-3 从发送器的传送序列图 .....	127
图 11-4 从接收器的传送序列图 .....	128
图 11-5 主发送器传送序列图 .....	129
图 11-6 主接收器传送序列图 .....	130
图 11-7 N>2 主接收器传送序列图 .....	131
图 11-8 N=2 主接收器传送序列图 .....	132
图 11-9 N=1 主接收器传送序列图 .....	133
图 12-1 USART 框图 .....	143
图 12-2 USART 中断映像图 .....	150
图 13-1 SPI 框图 .....	156
图 13-2 SPI 双线单向全双工连接示意图 .....	157
图 13-3 SPI 作主机单线单向只收连接示意图.....	157
图 13-4 SPI 作从机单线单向只收连接示意图.....	158
图 13-5 SPI 作单线双向半双工连接示意图 .....	158
图 13-6 SPI 中断 .....	162
图 13-7 I <sup>2</sup> S 框图 .....	163
图 13-8 I <sup>2</sup> S 全双工结构图 .....	164
图 13-9 I <sup>2</sup> S 从设备发送连接示意图 .....	164
图 13-10 I <sup>2</sup> S 从设备接收连接示意图 .....	165
图 13-11 I <sup>2</sup> S 主设备发送连接示意图.....	165
图 13-12 I <sup>2</sup> S 主设备接收连接示意图 .....	165
图 13-13 SPI 作主机 CK & MCK 来源示意图 .....	167

图 13-14 I <sup>2</sup> S 中断 .....	170
图 14-1 基本定时器框图 .....	176
图 14-2 使用 CK_INT 且分频系数为 1 .....	176
图 14-3 PRBEN=0 时的溢出事件 .....	176
图 14-4 PRBEN=1 时的溢出事件 .....	177
图 14-5 计数器时序图，内部时钟分频因子为 4 .....	177
图 14-6 通用定时器框图 .....	180
图 14-7 使用 CK_INT 且分频系数为 1 .....	180
图 14-8 外部时钟模式 A 框图 .....	181
图 14-9 使用外部时钟模式 A 计数 .....	181
图 14-10 外部时钟模式 B 框图 .....	181
图 14-11 使用外部时钟模式 B 计数 .....	181
图 14-12 当预分频器的参数从 1 变到 4 时，计数器的时序图 .....	182
图 14-13 PRBEN=0 时的溢出事件 .....	182
图 14-14 PRBEN=1 时的溢出事件 .....	183
图 14-15 计数器时序图，内部时钟分频因子为 4 .....	183
图 14-16 计数器时序图，内部时钟分频因子为 1，TMRx_PR=0x32 .....	183
图 14-17 编码模式计数实例（编码器模式 C） .....	184
图 14-18 输入/输出通道 1 的主电路 .....	184
图 14-19 通道 1 输入部分 .....	184
图 14-20 捕获/比较通道的输出部分（通道 1 至 4） .....	185
图 14-21 计数值与 C1DT 值匹配时翻转 C1ORAW .....	186
图 14-22 向上计数下 PWM 模式 A .....	186
图 14-23 中央双向对齐计数下 PWM 模式 A .....	186
图 14-24 单周期模式 .....	187
图 14-25 EXT 清除 CxORAW(PWM 模式 A) .....	187
图 14-26 复位模式例子 .....	187
图 14-27 挂起模式下例子 .....	188
图 14-28 触发器模式例子 .....	188
图 14-29 主/次定时器连接框图 .....	188
图 14-30 主定时器启动次定时器例子 .....	189
图 14-31 外部触发同时启动主、次定时器 .....	189
图 14-32 通用定时器 TMR9/12 框图 .....	201
图 14-33 通用定时器 TMR10/11/13/14 框图 .....	202
图 14-34 使用 CK_INT 且分频系数为 1 .....	202
图 14-35 外部时钟模式 A 框图 .....	202
图 14-36 使用外部时钟模式 A 计数 .....	203
图 14-37 当预分频器的参数从 1 变到 4 时，计数器的时序图 .....	203
图 14-38 PRBEN=0 时的溢出事件 .....	204
图 14-39 PRBEN=1 时的溢出事件 .....	204
图 14-40 输入/输出通道 1 的主电路 .....	204
图 14-41 通道 1 输入部分 .....	204
图 14-42 捕获/比较通道的输出部分（通道 1） .....	205
图 14-43 计数值与 C1DT 值匹配时翻转 C1ORAW .....	206
图 14-44 向上计数下 PWM 模式 A .....	206
图 14-45 单周期模式 .....	206
图 14-46 复位模式例子 .....	207
图 14-47 挂起模式下例子 .....	207
图 14-48 触发器模式例子 .....	207
图 14-49 高级控制定时器框图 .....	219
图 14-50 使用 CK_INT 且分频系数为 1 .....	220
图 14-51 外部时钟模式 A 框图 .....	220

图 14-52 使用外部时钟模式 A 计数 .....	220
图 14-53 外部时钟模式 B 框图 .....	220
图 14-54 使用外部时钟模式 B 计数 .....	221
图 14-55 当预分频器的参数从 1 变到 4 时，计数器的时序图 .....	221
图 14-56 PRBEN=0 时的溢出事件 .....	222
图 14-57 PRBEN=1 时的溢出事件 .....	222
图 14-58 计数器时序图，内部时钟分频因子为 4 .....	222
图 14-59 计数器时序图，内部时钟分频因子为 1，TMRx_PR=0x32 .....	222
图 14-60 RPR=2 时的 OVFIIF .....	223
图 14-61 编码模式计数实例（编码器模式 C） .....	223
图 14-62 输入/输出通道 1 的主电路 .....	223
图 14-63 通道 1 输入部分 .....	224
图 14-64 通道 1 至 3 输出部分 .....	224
图 14-65 通道 4 输出部分 .....	224
图 14-66 计数值与 C1DT 值匹配时翻转 C1ORAW .....	225
图 14-67 向上计数下 PWM 模式 A .....	226
图 14-68 中央双向对齐计数下 PWM 模式 .....	226
图 14-69 单周期模式 .....	226
图 14-70 EXT 清除 CxORAW(PWM 模式 A) .....	227
图 14-71 带死区插入的互补输出 .....	227
图 14-72 TMR 刹车功能的例子 .....	228
图 14-73 复位模式例子 .....	228
图 14-74 挂起模式下例子 .....	229
图 14-75 触发器模式例子 .....	229
图 15-1 窗口看门狗框图 .....	244
图 15-2 窗口看门狗时序图 .....	245
图 16-1 看门狗框图 .....	247
图 17-1 简化的 RTC 框图 .....	250
图 17-2 RTC 秒和闹钟波形图示例，DIV=0004，TA=00003 .....	252
图 17-3 RTC 溢出波形图示例，DIV=0004 .....	252
图 19-1 ADC1 框图 .....	260
图 19-2 ADC 基础操作流程 .....	261
图 19-3 ADC 上电与校准 .....	262
图 19-4 序列模式 .....	264
图 19-5 抢占自动转换模式 .....	264
图 19-6 反复模式 .....	264
图 19-7 分割模式 .....	265
图 19-8 数据内容处理 .....	265
图 19-9 主从模式的 ADC 框图 .....	266
图 19-10 普通同时模式 .....	267
图 19-11 抢占同时模式 .....	267
图 19-12 抢占交错触发模式 .....	268
图 19-13 普通短位移模式 .....	268
图 19-14 普通长位移模式 .....	269
图 20-1 DAC1/DAC2 模块框图 .....	282
图 20-2 DAC LFSR 寄存器算法 .....	284
图 20-3 DAC 三角波生成 .....	284
图 21-1 位时序 .....	289
图 21-2 位时序 .....	291
图 21-3 发送中断的产生 .....	292
图 21-4 接收中断 0 的产生 .....	292
图 21-5 接收中断 1 的产生 .....	292



图 21-6 状态错误中断的产生 .....	292
图 21-7 CAN 框图 .....	293
图 21-8 32 位宽标识符掩码模式 .....	295
图 21-9 32 位宽标识符列表模式 .....	295
图 21-10 16 位宽标识符掩码模式 .....	295
图 21-11 16 位宽标识符列表模式 .....	295
图 21-12 发送邮箱状态转换 .....	297
图 21-13 接收 FIFO 状态 .....	298
图 21-14 发送和接收邮箱 .....	309
图 22-1XMC 框图 .....	315
图 22-2 XMC 存储块区 .....	316
图 22-3 NOR/PSRAM 界面模式 1 读 .....	319
图 22-4 NOR/PSRAM 界面模式 1 写 .....	320
图 22-5 NOR/PSRAM 界面模式 2 读 .....	321
图 22-6 NOR/PSRAM 界面模式 2 写 .....	321
图 22-7 NOR/PSRAM 界面模式 A 读 .....	323
图 22-8 NOR/PSRAM 界面模式 A 写 .....	323
图 22-9 NOR/PSRAM 界面模式 B 读 .....	325
图 22-10 NOR/PSRAM 界面模式 B 写 .....	325
图 22-11 NOR/PSRAM 界面模式 C 读 .....	327
图 22-12 NOR/PSRAM 界面模式 C 写 .....	327
图 22-13 NOR/PSRAM 界面模式 D 读 .....	329
图 22-14 NOR/PSRAM 界面模式 D 写 .....	329
图 22-15 NOR/PSRAM 界面复用模式读 .....	331
图 22-16 NOR/PSRAM 界面复用模式写 .....	331
图 22-17 NOR/PSRAM 界面同步模式复用读 .....	333
图 22-18 NOR/PSRAM 界面同步模式复用写 .....	333
图 22-19 NAND 界面读写 .....	335
图 22-20 NAND 界面等待功能 .....	336
图 23-1 SDIO“无响应”和“无数据”操作 .....	344
图 23-2 SDIO（多）数据块读操作 .....	345
图 23-3 SDIO（多）数据块写操作 .....	345
图 23-4 SDIO 连续读操作 .....	345
图 23-5 SDIO 连续写操作 .....	346
图 23-6 SDIO 框图 .....	356
图 23-7 命令通道状态机（CCSM） .....	358
图 23-8 SDIO 命令传输 .....	358
图 23-9 数据通道状态机（DCSM） .....	359
图 24-1 普通端点和双缓冲端点与缓冲描述表之间差异 .....	373
图 25-1 ACC 中断映像图 .....	380
图 25-2 ACC 框图 .....	381
图 25-3 cross-return 策略 .....	382
图 26-1 EMAC 框图 .....	385
图 26-2 SMI 接口信号 .....	386
图 26-3 MII 信号线 .....	387
图 26-4 精简的独立于媒体的接口信号 .....	388
图 26-5 MII 时钟源（CLKOUT 脚提供时钟） .....	389
图 26-6 MII 时钟源（外部晶振提供时钟） .....	389
图 26-7 RMII 时钟源（CLKOUT 脚提供时钟） .....	389
图 26-8 RMII 时钟源（外部晶振提供时钟） .....	389
图 26-9 MAC 帧格式 .....	390
图 26-10 带标签的 MAC 帧格式 .....	391



图 26-11 环结构描述符和链结构描述符 .....	397
图 26-12 传输描述符 .....	399
图 26-13 RXDMA 描述符 .....	403
图 26-14 唤醒帧过滤寄存器 .....	406
图 26-15 用精调的方式更新系统时间 .....	408
图 26-16 PTP 触发输出连接到 TMR2 的 ITR1 .....	409
图 26-17 PPS 输出 .....	410
图 26-18 以太网中断 .....	410
图 26-19 以太网 MAC 唤醒帧过滤器寄存器(EMAC_MACRWFF) .....	419

## 表目录

表 1-1 SRAM 区中的位带地址映射 .....	30
表 1-2 外设区中的位带地址映射 .....	31
表 1-3 AT32F403A/407/407A 产品的向量表 .....	31
表 1-4 寄存器描述缩写说明 .....	35
表 1-5 器件特征信息相关寄存器地址和复位值 .....	36
表 2-1 各外设起始地址 .....	39
表 3-1 PWC 寄存器的映像和复位值 .....	45
表 4-1 CRM 寄存器的映像和复位值 .....	50
表 5-1 闪存存储结构 (1024K) .....	66
表 5-2 闪存存储组织 (512K) .....	66
表 5-3 闪存存储组织 (256K) .....	67
表 5-4 外部存储器支持的指令集 .....	68
表 5-5 用户系统数据说明 .....	69
表 5-6 闪存访问权限 .....	77
表 5-7 闪存接口—寄存器映像和复位值 .....	78
表 6-1 GPIO 寄存器地址映射和复位值 .....	88
表 7-1 复用功能输入配置 .....	93
表 7-2 复用功能输出配置 .....	93
表 7-3 硬件抢占功能 .....	93
表 7-4 调试端口映射 .....	94
表 7-5 IOMUX 寄存器地址映射和复位值 .....	94
表 8-1 外部中断/事件控制器寄存器映像和复位值 .....	107
表 9-1 DMA 错误事件 .....	111
表 9-2 DMA 中断 .....	112
表 9-3 DMA1 各通道的外设请求 .....	112
表 9-4 DMA2 各通道的外设请求 .....	112
表 9-5 DMA 各通道的弹性请求 .....	112
表 9-6 DMA 寄存器的映像和复位值 .....	114
表 10-1 CRC 计算单元寄存器映像 .....	122
表 11-1 I <sup>2</sup> C 寄存器地址映射和复位值 .....	136
表 12-1 设置波特率时的误差计算 .....	146
表 12-2 检测起始位和噪声的数据采样 .....	149
表 12-3 检测有效数据和噪声的数据采样 .....	149
表 12-4 USART 中断请求 .....	149
表 12-5 USART 寄存器映像和复位值 .....	150
表 13-1 使用系统时钟得到精确的音频频率 .....	167
表 13-2 SPI 寄存器列表及其复位值 .....	170
表 14-1 TMR 功能对比 .....	175
表 14-2 TMR6 和 TMR7 寄存器和复位值 .....	177
表 14-3 TMRx 内部触发连接 .....	182
表 14-4 计数方向与编码器信号的关系 .....	184
表 14-5 TMRx – 寄存器图和复位值 .....	190
表 14-6 标准 CxOUT 通道的输出控制位 .....	198
表 14-7 TMRx 内部触发连接 .....	203
表 14-8 TMRx – 寄存器图和复位值 .....	208
表 14-9 标准 CxOUT 通道的输出控制位 .....	213
表 14-10 TMRx – 寄存器图和复位值 .....	214
表 14-11 标准 CxOUT 通道的输出控制位 .....	218
表 14-12 TMRx 内部触发连接 .....	221
表 14-13 计数方向与编码器信号的关系 .....	223
表 14-14 TMR1、TMR8 – 寄存器图和复位值 .....	229

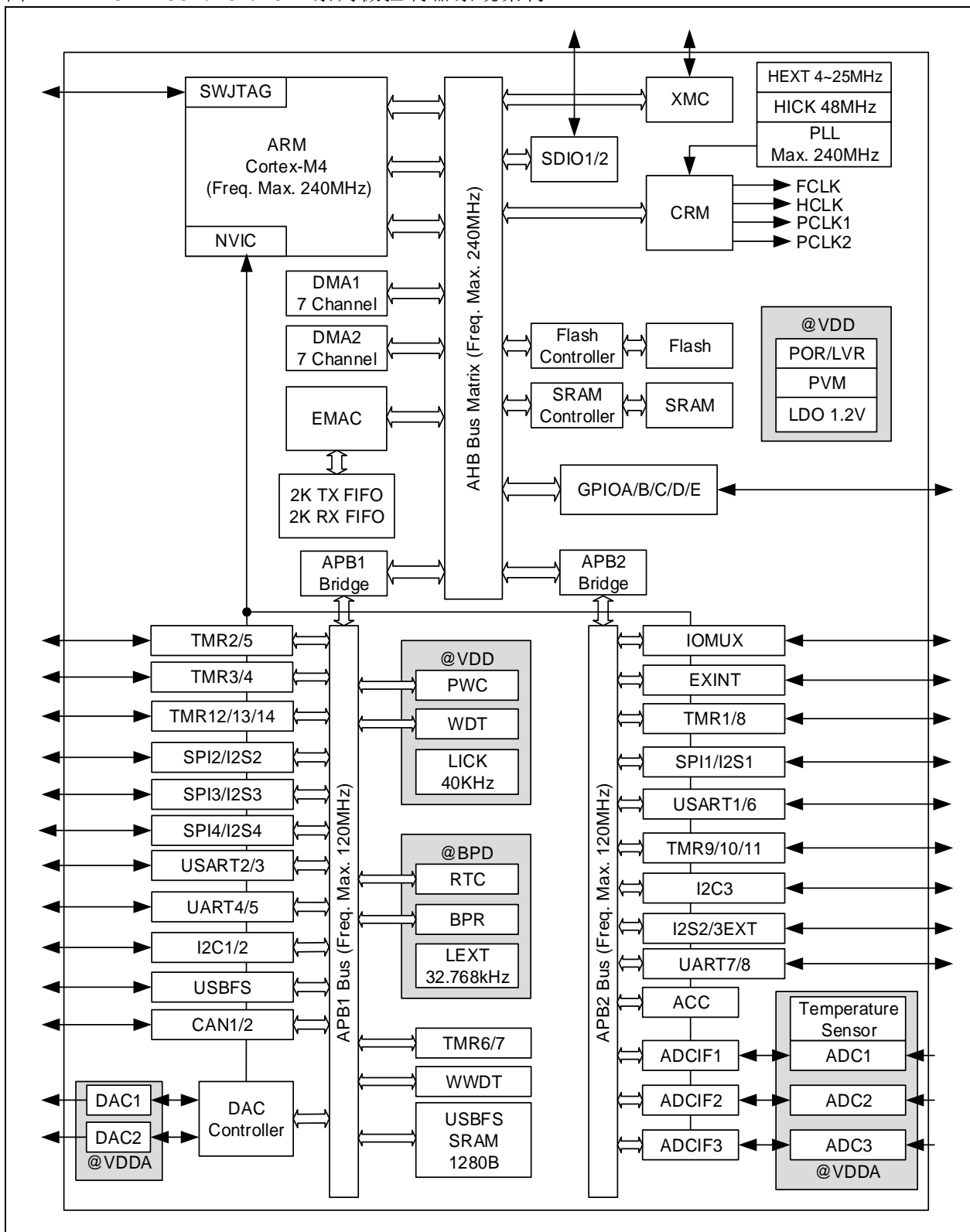
表 14-15 带刹车功能的互补输出通道 CxOUT 和 CxCOUT 的控制位 .....	240
表 15-1 PCLK1 频率为 72MHz 时, 最大和最小看门狗超时时间 .....	244
表 15-2 WWDT 寄存器的映像和复位值 .....	245
表 16-1 看门狗超时时间 (LICK=40kHz) .....	248
表 16-2 WDT 寄存器的映像和复位值 .....	248
表 17-1 RTC-寄存器映像和复位值 .....	252
表 18-1 BPR 寄存器映像和复位值 .....	255
表 19-1 ADC1 与 ADC2 的触发来源 .....	262
表 19-2 ADC3 的触发来源 .....	263
表 19-3 ADC 寄存器映像和复位值 .....	269
表 20-1 触发源选择 .....	283
表 20-2 DAC 寄存器映像和复位值 .....	285
表 21-1 CAN 寄存器映像和复位值 .....	299
表 22-1NOR/PSRAM 界面管脚 .....	315
表 22-2NAND 界面管脚 .....	315
表 22-3 存储区块选择 .....	317
表 22-4 NOR 闪存与 PSRAM 典型管脚信号 .....	317
表 22-5 HADDR 与外部存储器地址转换 .....	317
表 22-6 访问数据宽度与外部存储器数据宽度对照表 .....	318
表 22-7 NOR/PSRAM 参数寄存器 .....	318
表 22-8 模式 1 的 SRAM/NOR 闪存片选控制寄存器 (XMC_BK1CTRL) 配置 .....	319
表 22-9 模式 1 的 SRAM/NOR 闪存片选时序寄存器 (XMC_BK1TMG) 配置 .....	319
表 22-10 模式 2 的 SRAM/NOR 闪存片选控制寄存器 (XMC_BK1CTRL) 配置 .....	320
表 22-11 模式 2 的 SRAM/NOR 闪存片选时序寄存器 (XMC_BK1TMG) 配置 .....	320
表 22-12 模式 A 的 SRAM/NOR 闪存片选控制寄存器 (XMC_BK1CTRL) 配置 .....	322
表 22-13 模式 A 的 SRAM/NOR 闪存片选时序寄存器 (XMC_BK1TMG) 配置 .....	322
表 22-14 模式 A 的 SRAM/NOR 闪存写时序寄存器 (XMC_BK1TMGWR) 配置 .....	322
表 22-15 模式 B 的 SRAM/NOR 闪存片选控制寄存器 (XMC_BK1CTRL) 配置 .....	324
表 22-16 模式 B 的 SRAM/NOR 闪存片选时序寄存器 (XMC_BK1TMG) 配置 .....	324
表 22-17 模式 B 的 SRAM/NOR 闪存写时序寄存器 (XMC_BK1TMGWR) 配置 .....	324
表 22-18 模式 C 的 SRAM/NOR 闪存片选控制寄存器 (XMC_BK1CTRL) 配置 .....	326
表 22-19 模式 C 的 SRAM/NOR 闪存片选时序寄存器 (XMC_BK1TMG) 配置 .....	326
表 22-20 模式 C 的 SRAM/NOR 闪存写时序寄存器 (XMC_BK1TMGWR) 配置 .....	326
表 22-21 模式 D 的 SRAM/NOR 闪存片选控制寄存器 (XMC_BK1CTRL) 配置 .....	328
表 22-22 模式 D 的 SRAM/NOR 闪存片选时序寄存器 (XMC_BK1TMG) 配置 .....	328
表 22-23 模式 D 的 SRAM/NOR 闪存写时序寄存器 (XMC_BK1TMGWR) 配置 .....	328
表 22-24 复用模式的 SRAM/NOR 闪存片选控制寄存器 (XMC_BK1CTRL) 配置 .....	330
表 22-25 复用模式的 SRAM/NOR 闪存片选时序寄存器 (XMC_BK1TMG) 配置 .....	330
表 22-26 同步模式的 SRAM/NOR 闪存片选控制寄存器 (XMC_BK1CTRL) 配置 .....	332
表 22-27 同步模式的 SRAM/NOR 闪存片选时序寄存器 (XMC_BK1TMG) 配置 .....	332
表 22-28 NAND 闪存典型管脚信号 .....	334
表 22-29 访问数据宽度与 NAND 闪存数据宽度对照表 .....	334
表 22-30 NAND 参数寄存器 .....	334
表 22-31 ECC 结果有效位 .....	336
表 22-32 XMC 寄存器地址映像 .....	336
表 23-1 锁定/解锁命令的结构 .....	348
表 23-2 基于命令 .....	350
表 23-3 数据块读取命令 .....	351
表 23-4 数据流读取和写入命令 .....	351
表 23-5 数据块写入命令 .....	352
表 23-6 基于块传输的写保护命令 .....	352
表 23-7 擦除命令 .....	352

表 23-8 I/O 模式命令 .....	353
表 23-9 卡锁定命令 .....	353
表 23-10 应用相关命令 .....	353
表 23-11 R1 响应 .....	353
表 23-12 R2 响应 .....	354
表 23-13 R3 响应 .....	354
表 23-14 R4 响应 .....	354
表 23-15 R4b 响应 .....	354
表 23-16 R5 响应 .....	355
表 23-17 R6 响应 .....	355
表 23-18 SDIO 管脚定义 .....	356
表 23-19 命令格式 .....	357
表 23-20 短响应格式 .....	357
表 23-21 长响应格式 .....	357
表 23-22 命令通道状态标志 .....	357
表 23-23 数据令牌格式 .....	359
表 23-24 SDIO 寄存器映像 .....	361
表 23-25 响应类型和 SDIO_RSPx 寄存器 .....	364
表 24-1 缓冲区大小配置表 .....	372
表 24-2 USBFS 寄存器映像和复位值 .....	374
表 25-1 ACC 中断请求 .....	380
表 25-2 ACC 寄存器映像和复位值 .....	382
表 26-1 时钟范围 .....	387
表 26-2 发送接口信号编码 .....	388
表 26-3 接收接口信号编码 .....	388
表 26-4 以太网模块管脚配置（黑色：默认 红色：重映射信号） .....	390
表 26-5 目的地址过滤器结果列表 .....	392
表 26-6 源地址过滤器结果列表 .....	392
表 26-7 接收描述符 0 .....	404
表 26-8 以太网寄存器映像及其复位值 .....	411
表 27-1 跟踪功能使能 .....	443
表 27-2 跟踪功能模式 .....	444
表 27-3 DEBUG 寄存器地址和复位值 .....	444

# 1 系统架构

AT32F403A/407/407A 系列微控制器内部集成了：32 位 ARM®Cortex™-M4F 处理器，多个 16 位和 32 位的定时器，DMA 控制器，实时时钟 RTC，SPI 通信接口，I2C 通信接口，USART/UART 通信接口，SDIO 接口，CAN 总线控制器，外部存储控制器 XMC，USB2.0 全速设备接口，以太网 MAC，HICK 自动时钟校准 ACC，12 位 ADC，12 位 DAC 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 AT32F403A/407/407A系列微控制器系统架构



注意：AT32F403A 不支持 EMAC，AT32F407/407A 支持 EMAC。

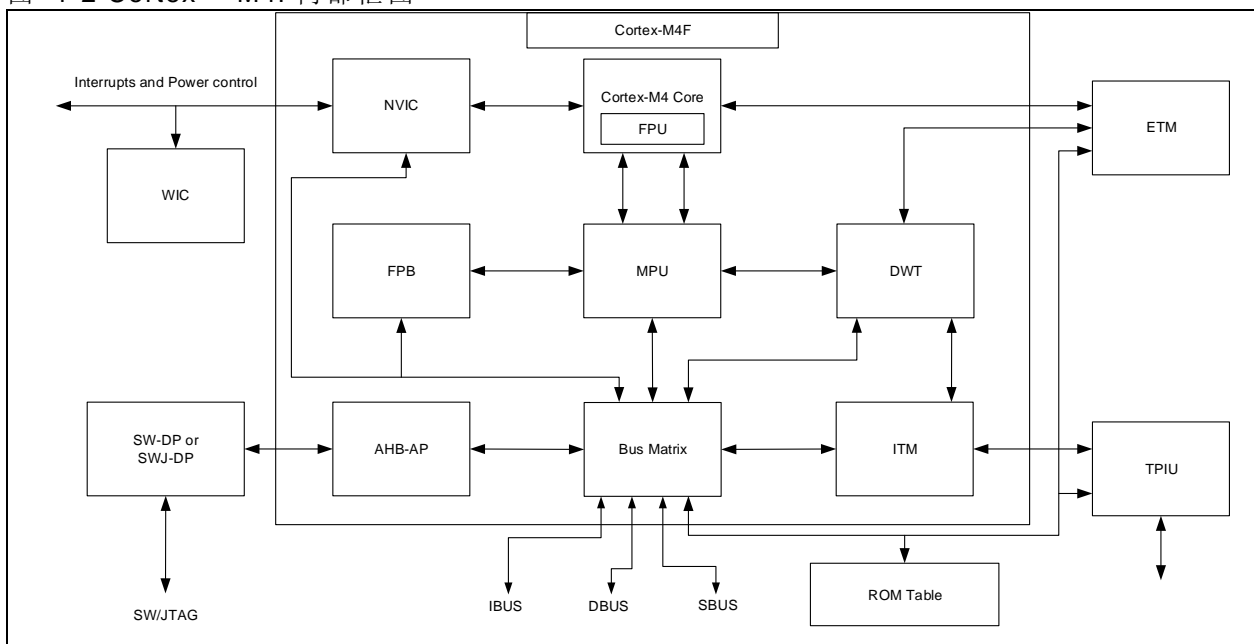
## 1.1 系统概述

### 1.1.1 ARM Cortex™-M4F处理器

Cortex™-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。Cortex™-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 Cortex™-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 Cortex™-M4F 更详尽信息。

图 1-2 Cortex™-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 Cortex™-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

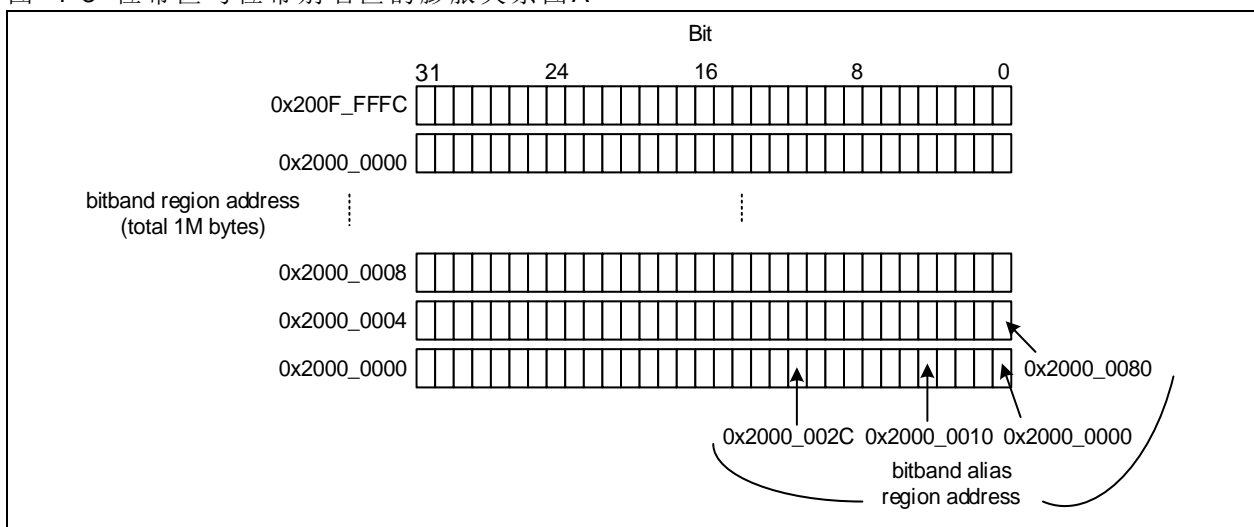
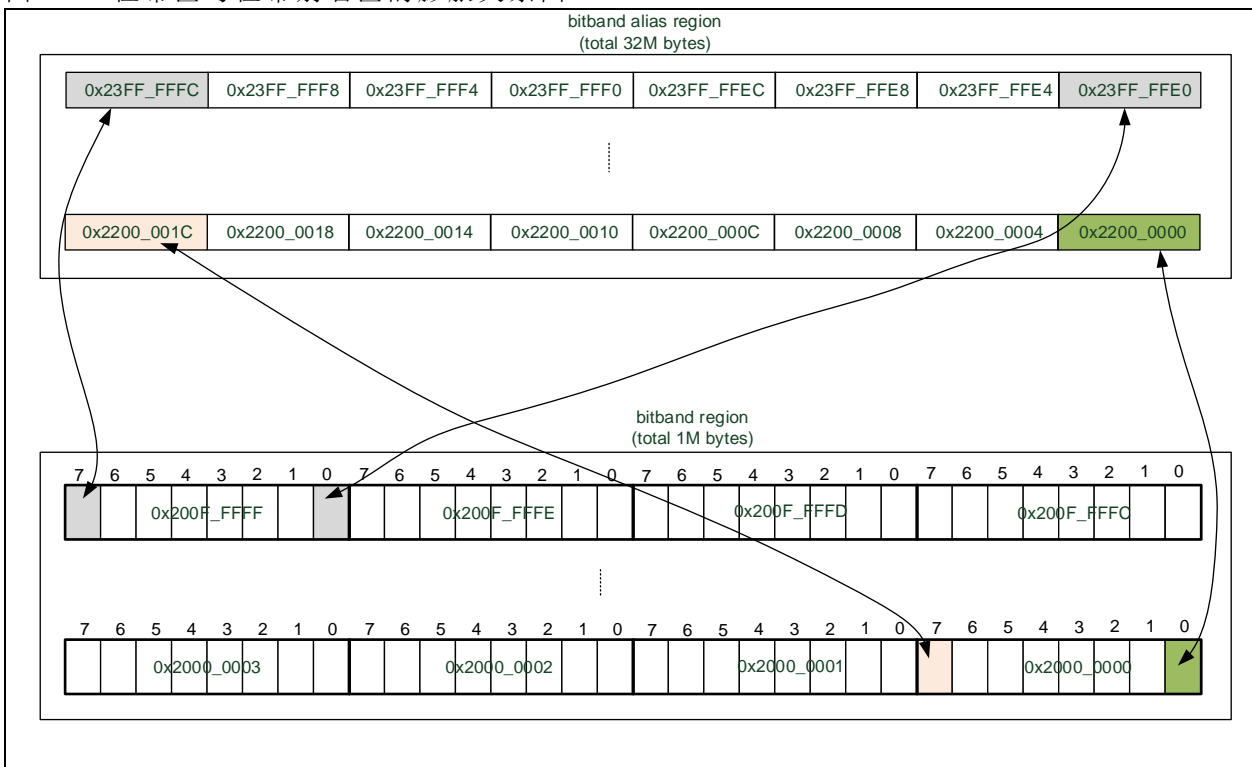




图 1-4 位带区与位带别名区的膨胀关系图B



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，会先把该地址变换成位带区地址。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，导致出现混乱的情况。

## 1.1.3 中断和异常向量

下面列出了 AT32F403A/407/407A 产品的向量表。

表 1-3 AT32F403A/407/407A产品的向量表

位置	优先级	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（Debug Monitor）	调试监控器		0x0000_0030
-	-		-	保留	0x0000_0034
5	可设置	PendSV	可挂起的系统服务		0x0000_0038
6	可设置	SysTick	系统嘀嗒定时器		0x0000_003C
0	7	可设置	WWDT	窗口定时器中断	0x0000_0040
1	8	可设置	PVM	连到 EXINT 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	RTC	实时时钟（RTC）全局中断	0x0000_004C
4	11	可设置	FLASH	闪存全局中断	0x0000_0050
5	12	可设置	CRM	时钟和复位控制（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	ADC1 和 ADC2 的全局中断	0x0000_0088
19	26	可设置	USBFS_H_CAN1_TX	USBFS 高优先级或 CAN1 发送中断	0x0000_008C
20	27	可设置	USBFS_L_CAN1_RX0	USBFS 低优先级或 CAN1 接收 0 中断	0x0000_0090
21	28	可设置	CAN1_RX1	CAN1 接收 1 中断	0x0000_0094
22	29	可设置	CAN1_SE	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_TMR11	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	可设置	TMR2	TMR2 全局中断	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 <sup>2</sup> C1 事件中断	0x0000_00BC
32	39	可设置	I2C1_ERR	I <sup>2</sup> C1 错误中断	0x0000_00C0
33	40	可设置	I2C2_EVT	I <sup>2</sup> C2 事件中断	0x0000_00C4
34	41	可设置	I2C2_ERR	I <sup>2</sup> 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_I <sup>2</sup> S2EXT	SPI2 全局中断和 I <sup>2</sup> S2EXT 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	可设置	USART3	USART3 全局中断	0x0000_00DC
40	47	可设置	EXINT15_10	EXINT 线[15: 10]中断	0x0000_00E0
41	48	可设置	RTCArm	连到 EXINT 的 RTC 闹钟中断	0x0000_00E4
42	49	可设置	USBFS_WAKEUP	连到 EXINT 的 USBFS 唤醒中断	0x0000_00E8
43	50	可设置	TMR8_BRK_TMR12	TMR8 停止中断和 TMR12 全局中断	0x0000_00EC
44	51	可设置	TMR8_OVF_TMR13	TMR8 溢出中断和 TMR13 全局中断	0x0000_00F0
45	52	可设置	TMR8_TRG_HALL_TMR14	TMR8 触发和 HALL 中断和 TMR14 全局中断	0x0000_00F4
46	53	可设置	TMR8_CH	TMR8 通道中断	0x0000_00F8
47	54	可设置	ADC3	ADC3 全局中断	0x0000_00FC
48	55	可设置	XMC	XMC 全局中断	0x0000_0100
49	56	可设置	SDIO	SDIO 全局中断	0x0000_0104
50	57	可设置	TMR5	TMR5 全局中断	0x0000_0108
51	58	可设置	SPI3_I <sup>2</sup> S3EXT	SPI3 全局中断和 I <sup>2</sup> S3EXT 全局中断	0x0000_010C
52	59	可设置	UART4	UART4 全局中断	0x0000_0110
53	60	可设置	UART5	UART5 全局中断	0x0000_0114
54	61	可设置	TMR6	TMR6 全局中断	0x0000_0118
55	62	可设置	TMR7	TMR7 全局中断	0x0000_011C
56	63	可设置	DMA2 通道 1	DMA2 通道 1 全局中断	0x0000_0120
57	64	可设置	DMA2 通道 2	DMA2 通道 2 全局中断	0x0000_0124
58	65	可设置	DMA2 通道 3	DMA2 通道 3 全局中断	0x0000_0128
59	66	可设置	DMA2 通道 4_5	DMA2 通道 4 和 DMA2 通道 5 全局中断	0x0000_012C
60	67	可设置	SDIO2	SDIO2 全局中断	0x0000_0130
61	68	可设置	I2C3_EVT	I2C3 事件中断	0x0000_0134
62	69	可设置	I2C3_ERR	I2C3 错误中断	0x0000_0138

63	70	可设置	SPI4	SPI4 全局中断	0x0000_013C
64	71	-	-	保留	0x0000_0140
65	72	-	-	保留	0x0000_0144
66	73	-	-	保留	0x0000_0148
67	74	-	-	保留	0x0000_014C
68	75	可设置	CAN2_TX	CAN2 发送中断	0x0000_0150
69	76	可设置	CAN2_RX0	CAN2 接收 0 中断	0x0000_0154
70	77	可设置	CAN2_RX1	CAN2 接收 1 中断	0x0000_0158
71	78	可设置	CAN2_SE	CAN2 状态错误中断	0x0000_015C
72	79	可设置	ACC	ACC 中断	0x0000_0160
73	80	可设置	USBFS_MAPH <sup>1</sup>	USBFS 重映射高优先级中断	0x0000_0164
74	81	可设置	USBFS_MAPL <sup>1</sup>	USBFS 重映射低优先级中断	0x0000_0168
75	82	可设置	DMA2 通道 6_7	DMA2 通道 6 和 DMA2 通道 7 全局中断	0x0000_016C
76	83	可设置	USART6	UART6 全局中断	0x0000_0170
77	84	可设置	UART7	UART7 全局中断	0x0000_0174
78	85	可设置	UART8	UART8 全局中断	0x0000_0178
79	86	可设置	EMAC <sup>2</sup>	以太网全局中断	0x0000_017C
80	87	可设置	EMAC_WKUP <sup>2</sup>	连接 EXINT 的以太网唤醒中断	0x0000_0180

注意：1、USBFS 模块中断支持重新映射，由中断映射寄存器（CRM\_INTMAP）的 USBINTMAP 位控制，当 USBINTMAP=0 时，使用 19 号 USBFS\_H 中断和 20 号 USBFS\_L 中断；当 USBINTMAP=1 时，使用 73 号 USB\_MAPH 中断和 74 号 USB\_MAPL 中断。

2、AT32F407/407A 支持 EMAC 和 EMAC\_WKUP 中断，AT32F403A 不支持这两个中断。

## 1.1.4 系统嘀嗒定时器（SysTick）

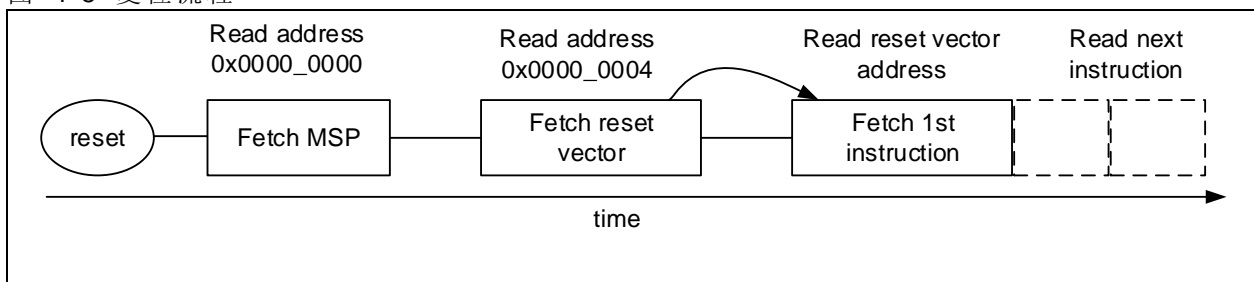
系统嘀嗒定时器是一个 24 位递减计数器，递减至零可自动重载计数初值。可产生周期性异常，用作嵌入式操作系统的多任务调度计数器，或对于无嵌入式操作系统，可用于调用需周期性执行的任务。系统嘀嗒定时器校准值固定值 9000，当系统嘀嗒时钟设定为 9MHz，产生 1ms 时间基准。

## 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。

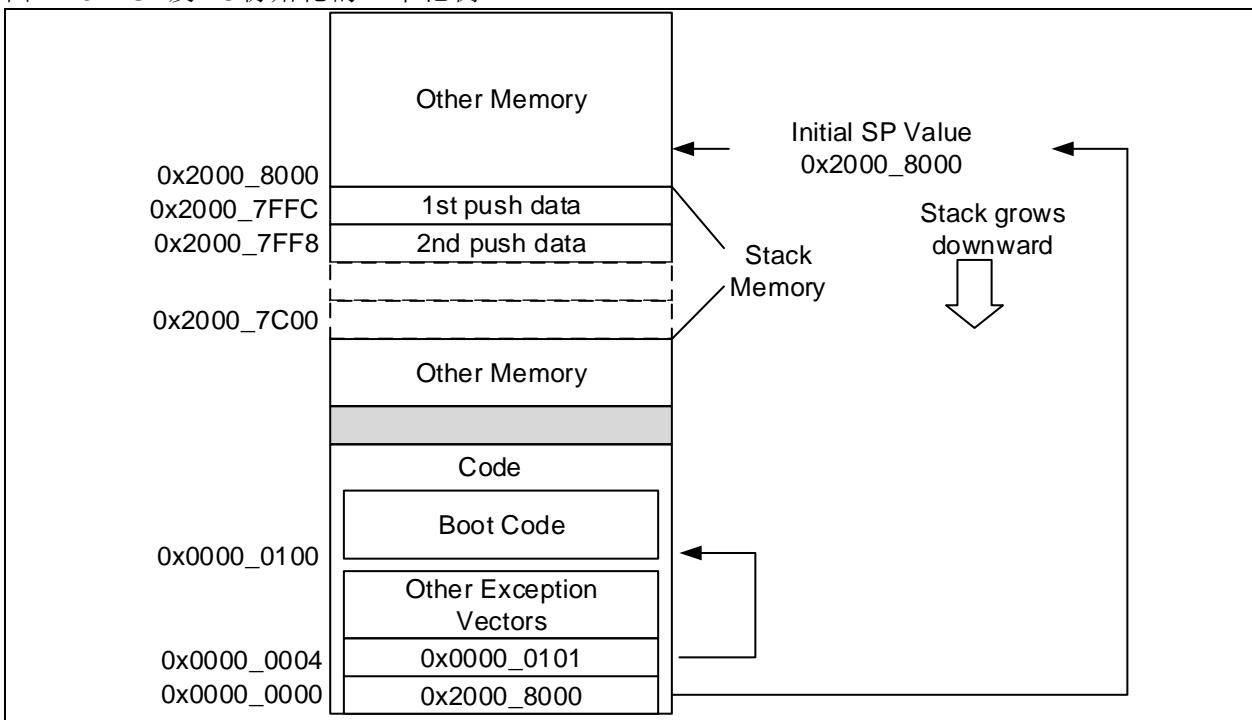
- 从地址 0x0000\_0000 处取出主栈指针（MSP）的初始值。
- 从地址 0x0000\_0004 处取出程序计数器（PC）的初始值，这个值是复位向量，LSB 必须是 1。然后从这个值所对应的地址处取指。

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 AT32F403A/407/407A 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，由 BOOT1 和 BOOT0 管脚来设定 CODE 从哪块存储器启动，当{BOOT1, BOOT0}=00/10 时，CODE 从主闪存存储器启动，当{BOOT1, BOOT0}=01 时，CODE 从启动程序存储器启动，当{BOOT1, BOOT0}=11 时，CODE 从片上 SRAM 启动。系统复位后或从待机模式退出时，BOOT1 和 BOOT0 管脚值都会被重新锁存。

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	0xXXXX
UID[31: 0]	0x1FFF F7E8	0xXXXX XXXX
UID[63: 32]	0x1FFF F7EC	0xXXXX XXXX
UID[95: 64]	0x1FFF F7F0	0xXXXX 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	0xXXXX	ro	闪存容量，以 KByte 为单位 例如：0x0080 = 128KByte

### 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 字符串序列
- 或者做为密钥的一部分

域	简称	复位值	类型	功能
位 31: 0	UID[31: 0]	0xXXXX XXXX	ro	UID 的 bit31 到 bit0 信息

域	简称	复位值	类型	功能
位 31: 0	UID[63: 32]	0xXXXX XXXX	ro	UID 的 bit63 到 bit32 信息

域	简称	复位值	类型	功能
位 31: 0	UID[95: 64]	0xXXXX XXXX	ro	UID 的 bit95 到 bit64 信息

注：UID[95: 88]为 Series ID，AT32F403A 为 0x07，AT32F407/407A 为 0x8。

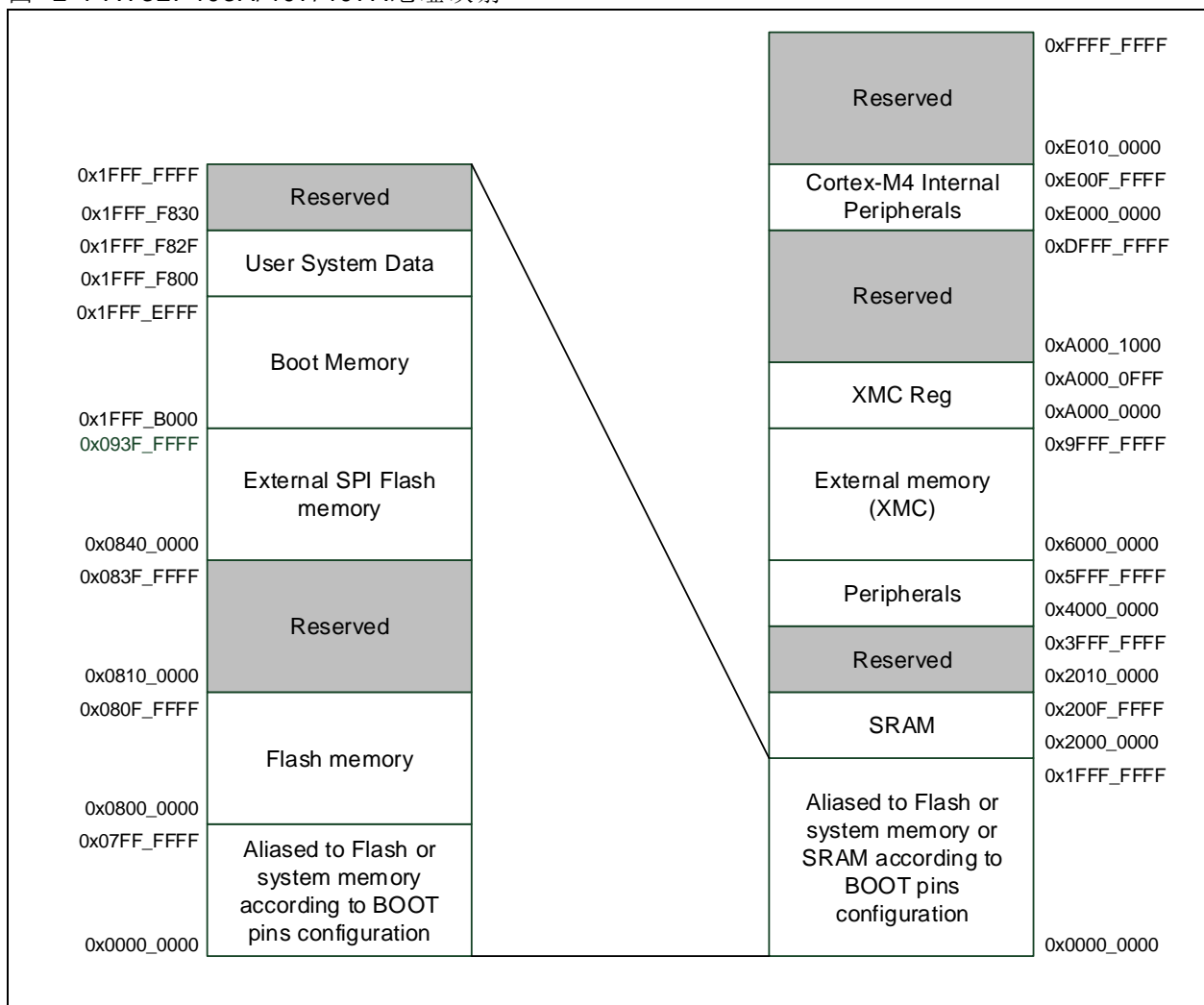


## 2 存储器资源

### 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 AT32F403A/407/407A地址映射



### 2.2 Flash存储器

AT32F403A/407/407A 系列提供最大 1024KB 的片上闪存，支持零等待延时的单周期最大 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（1024K）**

主存储器分为片 1 和片 2 闪存，每片闪存容量为 512K 字节，每片闪存包含 256 个扇区，每扇区大小为 2K 字节。

外部存储器容量可高达 16M 字节，包含 4096 个扇区，每扇区大小为 4K 字节。

结构	名称	地址范围
主存储器	片 1 (Bank1) 512KB	扇区 0
		扇区 1
		扇区 2
		...
	片 2 (Bank2) 512KB	扇区 255
		扇区 256
		扇区 257
		扇区 258
		...
		扇区 511
外部存储器	16MB	扇区 0
		扇区 1
		扇区 2
		...
		扇区 4095
信息块	启动程序代码区 16KB	0x0800 0000 – 0x0800 07FF
	用户系统数据区 48B	0x0800 0800 – 0x0800 0FFF

**闪存存储组织（512K）**

主存储器只有闪存容量为 512K 字节的片 1 闪存，包含 256 个扇区，每扇区大小为 2 K 字节。

外部存储器容量可高达 16M 字节，包含 4096 个扇区，每扇区大小为 4K 字节。

结构	名称	地址范围
主存储器	片 1 (Bank1) 512KB	扇区 0
		扇区 1
		扇区 2
		...
		扇区 255
外部存储器	16MB	扇区 0
		扇区 1
		扇区 2
		...
		扇区 4095
信息块	启动程序代码区 16KB	0x0800 0000 – 0x0800 07FF
	用户系统数据区 48B	0x0800 0800 – 0x0800 0FFF

**闪存存储组织（256K）**

主存储器只有闪存容量为 256K 字节的片 1 闪存，包含 128 个扇区，每扇区大小为 2K 字节。

外部存储器容量可高达 16M 字节，包含 4096 个扇区，每扇区大小为 4K 字节。

结构	名称	地址范围
主存储器	片 1 (Bank1) 256KB	扇区 0
		扇区 1
		扇区 2
		...
		扇区 127
外部存储器	16MB	扇区 0
		扇区 1
		扇区 2
		...
		扇区 4095
信息块	启动程序代码区 16KB	0x0800 0000 – 0x0800 07FF
	用户系统数据区 48B	0x0800 0800 – 0x0800 0FFF

## 2.3 SRAM存储器

AT32F403A/407/407A 系列内置 96K 字节的片上 SRAM，起始地址为 0x2000\_0000。它可以以字节、半字（16 位）或字（32 位）访问。AT32F403A/407/407A 系列另外提供一个特别的模式能使片上 SRAM 在 96K 字节和 224K 字节两个配置之间动态配置，用户可透过设定扩充的系统选项 EOPB0 位来使用此扩充模式。在 224KB 扩充模式下，零等待延迟(zero wait state)的闪存容量限制为 128K 字节。在 96KB 扩充模式下，零等待延迟(zero wait state)的闪存容量限制为 256K 字节。

## 2.4 外设地址映射

表 2-1 各外设起始地址

总线	起始地址	外设
AHB	0A000 1000 - 0xFFFF FFFF	保留
	0A000 0000 - 0xA000 0FFF	XMC_REG
	0x6000 0000 - 0x9FFF FFFF	XMC_MEM
	0x4002 A000 - 0x5FFF FFFF	保留
	0x4002 8000 - 0x4002 9FFF	EMAC
	0x4002 3400 - 0x4002 7FFF	SDIO2
	0x4002 3000 - 0x4002 33FF	CRC
	0x4002 2000 - 0x4002 23FF	闪存存储器接口（FLASH）
	0x4002 1400 - 0x4002 1FFF	保留
	0x4002 1000 - 0x4002 13FF	时钟和复位管理（CRM）
	0x4002 0800 - 0x4002 0FFF	保留
	0x4002 0400 - 0x4002 07FF	DMA2
	0x4002 0000 - 0x4002 03FF	DMA1
	0x4001 8400 - 0x4001 7FFF	保留
	0x4001 8000 - 0x4001 83FF	SDIO
	0x4001 7400 - 0x4001 7FFF	保留
	0x4001 7000 - 0x4001 73FF	I <sup>2</sup> S3EXT
	0x4001 6C00 - 0x4001 6FFF	I <sup>2</sup> S2EXT
APB2	0x4001 6800 - 0x4001 6BFF	UART8
	0x4001 6400 - 0x4001 67FF	UART7
	0x4001 6000 - 0x4001 63FF	USART6
	0x4001 5C00 - 0x4001 5FFF	I <sup>2</sup> C3
	0x4001 5800 - 0x4001 5BFF	ACC
	0x4001 5400 - 0x4001 57FF	TMR11 定时器
	0x4001 5000 - 0x4001 53FF	TMR10 定时器
	0x4001 4C00 - 0x4001 4FFF	TMR9 定时器
	0x4001 4400 - 0x4001 4BFF	保留
	0x4001 4000 - 0x4001 43FF	保留
	0x4001 3C00 - 0x4001 3FFF	ADC3
	0x4001 3800 - 0x4001 3BFF	USART1
	0x4001 3400 - 0x4001 37FF	TMR8 定时器
	0x4001 3000 - 0x4001 33FF	SPI1/I <sup>2</sup> S1

APB1	0x4001 2C00 - 0x4001 2FFF	TMR1 定时器
	0x4001 2800 - 0x4001 2BFF	ADC2
	0x4001 2400 - 0x4001 27FF	ADC1
	0x4001 2000 - 0x4001 23FF	保留
	0x4001 1C00 - 0x4001 1FFF	保留
	0x4001 1800 - 0x4001 1BFF	GPIO 端口 E
	0x4001 1400 - 0x4001 17FF	GPIO 端口 D
	0x4001 1000 - 0x4001 13FF	GPIO 端口 C
	0x4001 0C00 - 0x4001 0FFF	GPIO 端口 B
	0x4001 0800 - 0x4001 0BFF	GPIO 端口 A
	0x4001 0400 - 0x4001 07FF	EXINT
	0x4001 0000 - 0x4001 03FF	IOMUX
	0x4000 8400 - 0x4000 FFFF	保留
	0x4000 7800 - 0x4000 83FF	USBFS 1280 字节缓冲区 <sup>(1)</sup>
	0x4000 7400 - 0x4000 77FF	DAC
	0x4000 7000 - 0x4000 73FF	电源控制 (PWC)
	0x4000 6C00 - 0x4000 6FFF	后备寄存器 (BPR)
	0x4000 6800 - 0x4000 6BFF	CAN2
	0x4000 6400 - 0x4000 67FF	CAN1
	0x4000 6000 - 0x4000 63FF	USBFS 512 字节缓冲区 <sup>(1)</sup>
	0x4000 5C00 - 0x4000 5FFF	USBFS
	0x4000 5800 - 0x4000 5BFF	I <sup>2</sup> C2
	0x4000 5400 - 0x4000 57FF	I <sup>2</sup> C1
	0x4000 5000 - 0x4000 53FF	UART5
	0x4000 4C00 - 0x4000 4FFF	UART4
	0x4000 4800 - 0x4000 4BFF	USART3
	0x4000 4400 - 0x4000 47FF	USART2
	0x4000 4000 - 0x4000 43FF	SPI4/I <sup>2</sup> S4
	0x4000 3C00 - 0x4000 3FFF	SPI3/I <sup>2</sup> S3
	0x4000 3800 - 0x4000 3BFF	SPI2/I <sup>2</sup> S2
	0x4000 3400 - 0x4000 37FF	保留
	0x4000 3000 - 0x4000 33FF	看门狗 (WDT)
	0x4000 2C00 - 0x4000 2FFF	窗口看门狗 (WWDT)
	0x4000 2800 - 0x4000 2BFF	RTC
	0x4000 2400 - 0x4000 27FF	保留
	0x4000 2000 - 0x4000 23FF	TMR14 定时器
	0x4000 1C00 - 0x4000 1FFF	TMR13 定时器
	0x4000 1800 - 0x4000 1BFF	TMR12 定时器
	0x4000 1400 - 0x4000 17FF	TMR7 定时器
	0x4000 1000 - 0x4000 13FF	TMR6 定时器
	0x4000 0C00 - 0x4000 0FFF	TMR5 定时器

0x4000 0800 - 0x4000 0BFF	TMR4 定时器
0x4000 0400 - 0x4000 07FF	TMR3 定时器
0x4000 0000 - 0x4000 03FF	TMR2 定时器

注意：1、当 USBBUFS=0 时，USBFS 缓冲区大小为 512 字节，其地址范围为 0x4000 6000~0x4000 63FF。当 USBBUFS=1 时，USBFS 缓冲区大小为 768~1280 字节，其地址范围为 0x4000 7800~0x4000 83FF。CAN1 和 CAN2 都不使能时，USBFS 缓冲区最大可设定到 1280 字节，CAN1 或 CAN2 任意一个使能，USBFS 缓冲区最大可设定到 1024 字节，CAN1 和 CAN2 都使能时，USBFS 缓冲区最大可设定到 768 字节。

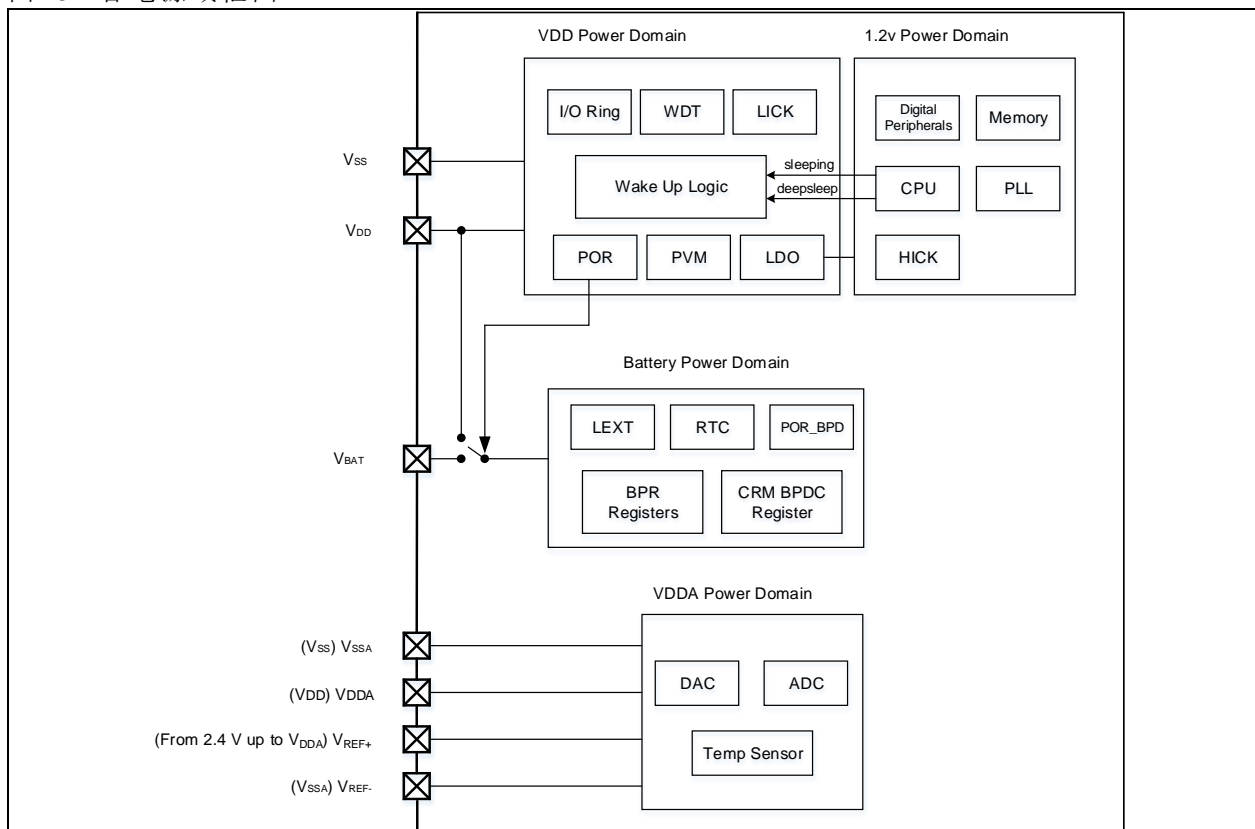
2、仅 AT32F407 和 A32F407A 支持 EMAC 模块。

## 3 电源控制（PWC）

### 3.1 简介

AT32F403A/407/407A 系列设备工作电压范围为 2.6V 至 3.6V，正常工作温度范围为-40~+105℃。AT32F403A/407/407A 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。AT32F403A/407/407A 系列设备有三个电源域——VDD/VDDA 域，1.2V 域和电池供电域。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，电池供电域由 V<sub>BAT</sub> 管脚供电。

图 3-1 各电源域框图



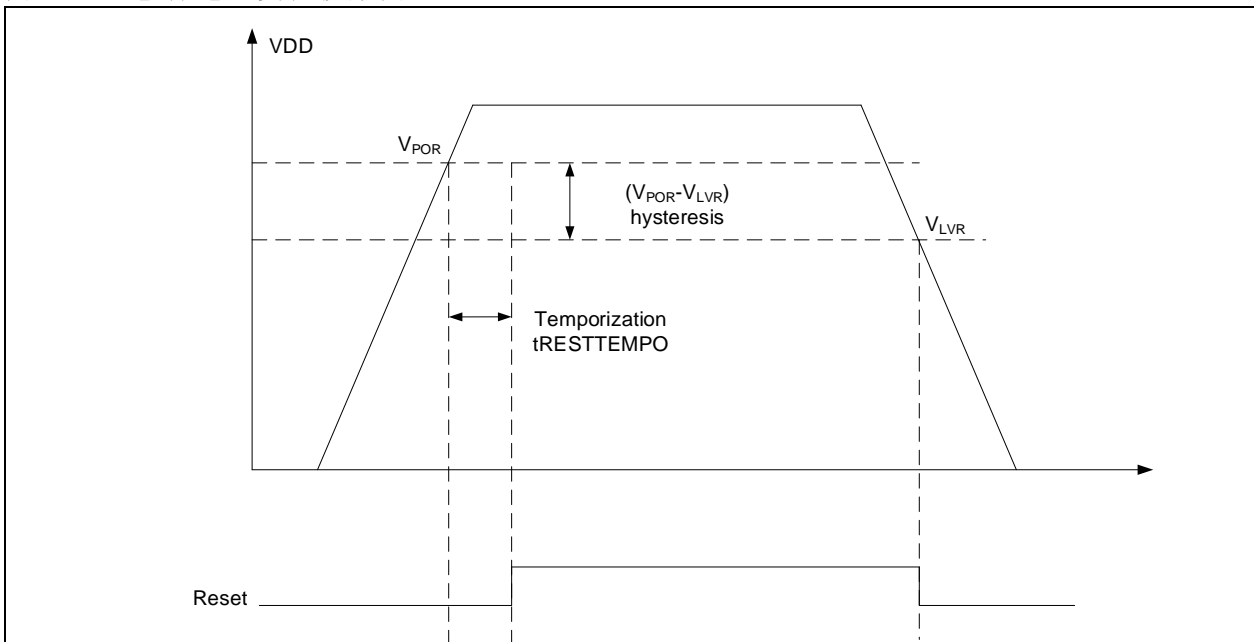
### 3.2 主要特点

- 具备三个电源域：VDD/VDDA 域、1.2V 内核域和电池供电域。
- 支持三种省电模式：睡眠模式、深度睡眠模式和待机模式。
- 内建电压调节器，提供 1.2V 给内核域。
- 提供电压监测器，能在电压低于或高于阈值时产生中断或事件。
- 当 VDD 供电关闭时，电池供电域由电池 V<sub>BAT</sub> 供电。
- VDD/VDDA 采用独立的数字和模拟地，用于隔离电源噪声。

### 3.3 上电低电压复位

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<sub>POR</sub> 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<sub>LVR</sub> 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

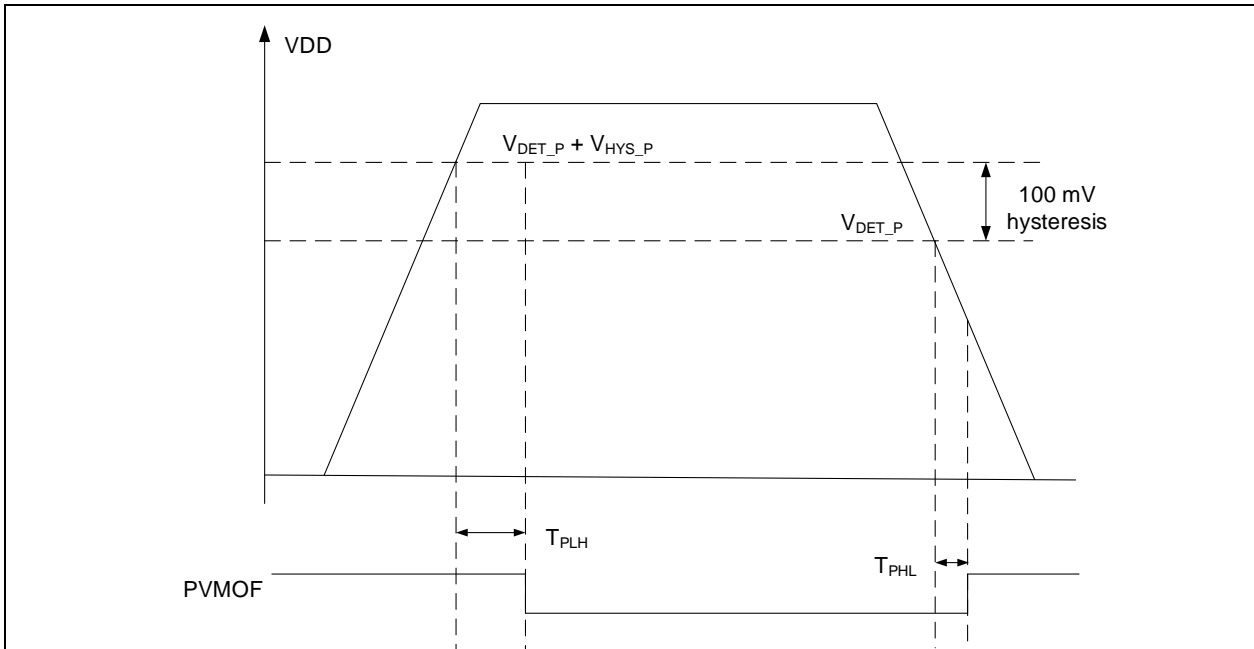


### 3.4 电压监测器（PVM）

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 以及除 PC13、PC14 和 PC15 之外的所有 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。

#### 电池供电域

电池供电域包括 RTC 电路、LEXT 振荡器、PC13、PC14 以及 PC15。电池供电域由 VDD 或 VBAT 管脚供电, 当 VDD 主电源被切断时, 电池供电域自动切换至 VBAT 管脚供电, RTC 可以正常工作。

- 1) 当电池供电域由 VDD 供电时, PC13 可以作为通用 I/O 口、TAMPER 管脚、RTC 校准时钟、RTC 闹钟或秒输出, PC14 和 PC15 可以用于 GPIO 或 LEXT 管脚。(PC13 至 PC15 作为 I/O 口的速度必须限制在 2MHz 以下, 最大负载为 30pF, 而且这些 I/O 口绝对不能当作电流源)。
- 2) 当电池供电域由 VBAT 管脚供电时, PC13 可以作为 TAMPER 管脚、RTC 闹钟或秒输出, PC14 和 PC15 只能用于 LEXT 管脚。

电池供电域的电源开关不会因为 VDD 在上升阶段或是因为 VDD 低电压复位而断开与 VBAT 的连接。当主电源上 VDD 上电较快, 电源开关还未切换至主电源 VDD 时, 为防止电流从 VDD 注入到 VBAT, 推荐在 VDD 与 VBAT 之间接一个低压降二极管。若应用中没有外部电池, VBAT 最好连接一个 100nF 的陶瓷滤波电容并在外部连接到 VDD。

## 3.6 省电模式

当 CPU 无需继续运行时, AT32F403A/407/407A 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 1.2V 电源 (CPU 内核、内存和内嵌外设)。

- 1) 执行 WFI 指令进入睡眠模式时, 只要产生外设中断, 都能使系统退出睡眠模式。
- 2) 执行 WFE 指令进入睡眠模式时, 存在两种方式的唤醒事件, 使系统退出睡眠模式:
  - 使能任一外设中断 (未在 NVIC 中使能) 且使能 SEVONPEND 位可以产生唤醒事件。系统唤醒后, 需清除外设中断挂起位和 NVIC 通道挂起位。
  - 配置内部 EXINT 线为事件模式来产生唤醒事件。

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

#### 深度睡眠模式 (Deepsleep Mode)

通过设置 Cortex™-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。

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

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

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

## 待机模式（Standby Mode）

待机模式可最大限度的降低系统功耗，在该模式下，电压调节器关闭，只有电池供电的寄存器和待机电路维持供电，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 SWEF 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 管脚的上升沿、RTC 闹钟事件的上升沿、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

必须以字（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

### 3.7.1 电源控制寄存器（PWC\_CTRL）

域	简称	复位值	类型	功能
位 31: 9	保留	0x000000	resd	保持默认值。
位 8	BPWEN	0x0	rw	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 7: 5	PVMSEL	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 4	PVMEN	0x0	rw	电压监测使能（Power voltage monitoring enable） 0：关闭； 1：开启。
位 3	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，且任何时刻读取该位返回值均是零。
				清除 SWEF 标志（Clear SWEF flag）
				0：无效；
				1：清除 SWEF 标志。
位 2	CLSWEF	0x0	wo	注： 实际 SWEF 标志的清除大约需要 2 个系统时钟周期； 该位在清除 SWEF 后由硬件将其清零，且任何时刻读取该位返回值均是零。
位 1	LPSEL	0x0	rw	SLEEPDEEP 状态下的低功耗模式选择位（Low power mode select when Cortex™-M4F sleepdeep） 0：进入 DEEPSLEEP 模式； 1：进入待机模式
位 0	VRSEL	0x0	rw	DEEPSLEEP 模式下电压调节器状态选择（Voltage regulator state select when deepsleep mode） 0：正常开启； 1：低功耗模式。

## 3.7.2 电源控制及状态寄存器（PWC\_CTRLSTS）

与标准的 APB 读相比，读此寄存器需要额外的 APB 周期

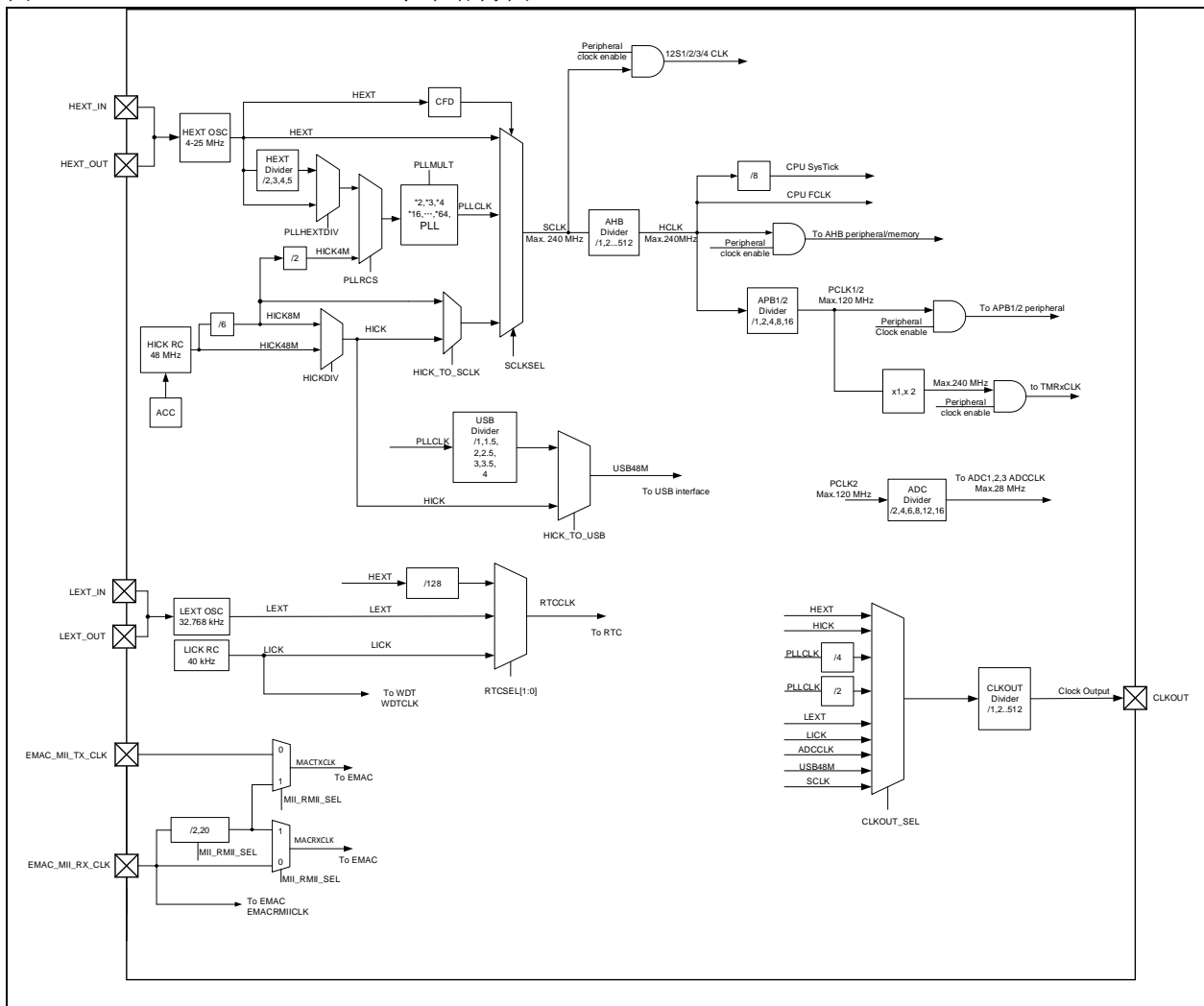
域	简称	复位值	类型	功能
位 31: 9	保留	0x000000	resd	保持默认值。
位 8	SWPEN	0x0	rw	待机唤醒管脚使能（Standby wake-up pin enable） 0：关闭（该管脚可用作通用 I/O）； 1：开启（该管脚被强置为输入下拉模式，且无法再用作通用 I/O）。 注：在系统复位时硬件将清除这一位。
位 7: 3	保留	0x00	resd	保持默认值。
位 2	PVMOF	0x0	ro	电源电压检测输出标志（Power voltage monitoring output flag） 0：电源电压高于临界值； 1：电源电压低于临界值。 注：待机模式下电压监测停止工作。
位 1	SEF	0x0	ro	进入待机模式标志（Standby mode entry flag） 0：未进过待机模式； 1：有进过待机模式。 注：该位被硬件置起（进入待机模式时），由 POR/LVR 或写 CLSEF 位将其清零。
位 0	SWEF	0x0	ro	待机唤醒事件标志（Standby wake-up event flag） 0：无唤醒事件产生； 1：有唤醒事件产生。 注： 该位被硬件置起（产生唤醒事件时），由 POR/LVR 或写 CLSWEF 位将其清零。 唤醒事件将由以下几种情况产生： 在待机唤醒管脚上出现上升沿时，将产生唤醒事件； 出现 RTC 闹钟事件时，将产生唤醒事件； 待机唤醒管脚保持高电平期间使能该待机唤醒管脚，将产生唤醒事件。

## 4 时钟和复位管理 (CRM)

### 4.1 时钟

AT32F403A/407/407A 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 AT32F403A/407/407A时钟结构图



AHB、APB1 和 APB2 的频率都支持多种分频。AHB 域的最大频率是 240MHz, APB1 和 APB2 域的最大允许频率是 120MHz。

#### 4.1.1 时钟源

##### ● HEXT 振荡器时钟

包括 HEXT 晶体/陶瓷谐振器和 HEXT 旁路时钟两个时钟源。

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

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

##### ● HICK 振荡器时钟

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

HICK 时钟直到稳定后才会被释放出来。

- PLL 时钟

PLL 的输入时钟源可以选择 HICK 时钟或 HEXT 时钟且输入时钟范围为 2M~16MHz 之间。使用 PLL 前，一定要先选择输入时钟源和倍频因子，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。

- LEXT 振荡器时钟

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

- LEXT 晶体/陶瓷谐振器

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

- LEXT 旁路时钟

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

- LICK 振荡器时钟

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

LICK 时钟直到稳定后，才会被释放出来。

## 4.1.2 系统时钟

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

## 4.1.3 外设时钟

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

系统嘀嗒定时器（SysTick）使用 HCLK 或 HCLK 的 8 分频作为时钟。

ADC 使用 APB2 时钟的 2、4、6、8、12、16 分频作为时钟。

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

USB 时钟可在 HICK 和 PLL 分频时钟之间切换。当选 HICK 时钟源时，需配置 USB 时钟为 48MHz 时钟；当选 PLL 分频时钟时，USB 分频器提供 48MHz 的 USBCLK 时钟，PLL 需设置为  $48 \times N \times 0.5$  MHz ( $N=2,3,4,5\dots$ )。

RTC 的时钟源有：HEXT 振荡器 128 分频时钟，LEXT 振荡器时钟 LICK 振荡器时钟。RTC 的时钟源一旦选择后就不可再更改，只有将电池供电域复位后才能重新配置 RTC 时钟源。当 VDD 掉电时，RTC 使用 LEXT 作为时钟的话，RTC 可以继续工作，但 RTC 使用 HEXT 或 LICK 作为时钟源时，由于 HEXT 和 LICK 均掉电，会导致 RTC 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 检测到失效后，将时钟失效事件送到 TMR1 和 TMR8 的刹车输入端，并产生 CFD 中断，此 CFD 中断直接连到 CPU 的 NMI 中断，供软件完成营救操作。NMI 中断将一直重复执行，直到 CFD 中断挂起位被清除为止，所以在 NMI 的处理程序中必须清除 CFD 中断。当 HEXT 时钟出现故障时，将导致系统时钟切换到 HICK 时钟，同时关闭 CFD，关闭 HEXT 时钟，如果 HEXT 时钟通过 PLL 做为系统时钟时，也会关闭 PLL 模块。

## 4.1.5 自动滑顺频率切换

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

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



### 4.1.6 内部时钟输出

微控制器允许输出内部时钟信号到外部 CLKOUT 管脚。ADC CLK、USB48M、SCLK、LICK、LEXT、HICK、HEXT、除 2 的 PLL 时钟以及除 4 的 PLL 时钟这 9 个时钟信号可输出到 CLKOUT。

### 4.1.7 中断

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

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

## 4.2 复位

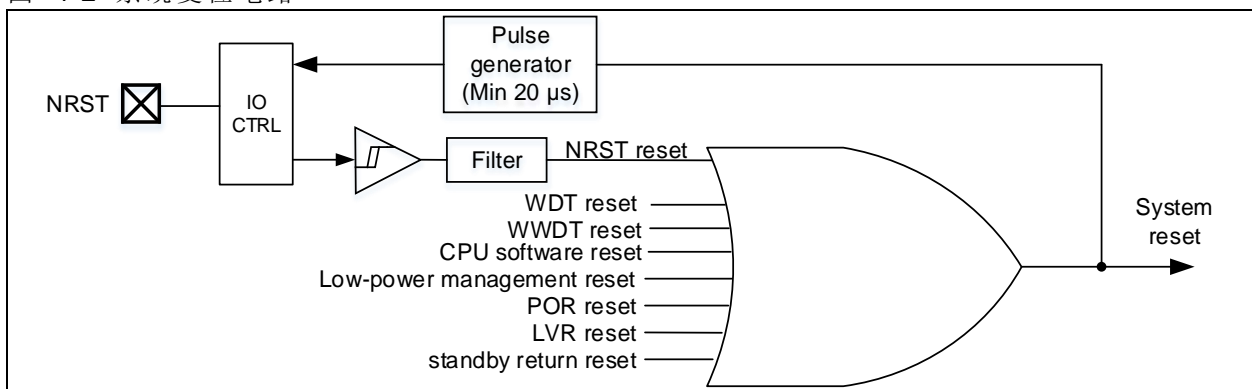
### 4.2.1 系统复位

AT32F403A/407/407A 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。
- POR 复位：上电复位
- LVR 复位：低电压复位
- 从待机模式中返回等事件产生复位。

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

图 4-2 系统复位电路



### 4.2.2 电池供电域复位

电池供电域复位包括以下复位源：

- 电池供电域软件复位：设置电池供电域控制寄存器（CRM\_BPDC）中的 BPDRST 位来产生复位
  - 在 VDD 和 VBAT 两者掉电的前提下，VDD 或 VBAT 再上电将产生复位。
- 电池供电域软件复位只影响电池供电域。

## 4.3 CRM寄存器

下表列出了 CRM 寄存器的映像和复位值。

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

表 4-1 CRM寄存器的映像和复位值

寄存器简称	基址偏移量	复位值
CRM_CTRL	0x000	0x0000 XX83
CRM_CFG	0x004	0x0000 0000
CRM_CLKINT	0x008	0x0000 0000
CRM_APB2RST	0x00C	0x0000 0000
CRM_APB1RST	0x010	0x0000 0000
CRM_AHBEN	0x014	0x0000 0014
CRM_APB2EN	0x018	0x0000 0000
CRM_APB1EN	0x01C	0x0000 0000
CRM_BPDC	0x020	0x0000 0000
CRM_CTRLSTS	0x024	0x0C00 0000
CRM_AHBRST	0x028	0x0000 0000
CRM_MISC1	0x030	0x0000 0000
CRM_MISC2	0x050	0x0000 0000
CRM_MISC3	0x054	0x0000 000D
CRM_INTMAP	0x05C	0x0000 0000

### 4.3.1 时钟控制寄存器（CRM\_CTRL）

域	简称	复位值	类型	功能
位 30: 26	保留	0x00	resd	请保持默认值。
位 25	PLLSTBL	0x0	ro	PLL 时钟稳定（PLL clock stable） 该位待 PLL 稳定后由硬件置起。 0: 未稳定； 1: 已稳定。
位 24	PLLEN	0x0	rw	PLL 使能（PLL enable） 该位可由软件置起或清除，也可在进入待机或深度睡眠模式时，由硬件清除。当系统时钟为 PLL 时钟时，该位无法清除。 0: 关闭； 1: 开启。
位 23: 20	保留	0x0	resd	保持默认值。
位 19	CFDEN	0x0	rw	时钟失效检测使能（Clock Failure Detection enable） 0: 关闭； 1: 开启。
位 18	HEXTBYPSS	0x0	rw	HEXT 旁路使能（High speed external crystal bypass） 只有在 HEXT 关闭时，软件才能操作该位。 0: 关闭； 1: 开启。
位 17	HEXTSTBL	0x0	ro	HEXT 时钟稳定（High speed external crystal stable） 该位待 HEXT 稳定后由硬件置起。 0: 未稳定； 1: 已稳定。



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

## 4.3.2 时钟配置寄存器 (CRM\_CFG)

访问：0 到 2 个等待周期，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	PLLRANGE	0x0	rw	PLL 倍频输出时钟范围选择位(PLL clock output range) 0: PLL 输出时钟范围 ≤ 72 MHz; 1: PLL 输出时钟范围 > 72 MHz。
位 26: 24	CLKOUT_SEL	0x0	rw	内部时钟输出选择 (Clock output selection) CLKOUT_SEL[3]在额外寄存器 1 (CRM_MISC1) 的位 16。 0000: 无; 0001: 保留; 0010: LICK; 0011: LEXT; 0100: SCLK; 0101: HICK; 0110: HEXT; 0111: PLL/2; 1100: PLL/4; 1101: USB; 1110: ADC。

位 27 位 23: 22	USBDIV	0x0	rw	<p>USB 分频因子 (USB division) PLL 时钟分频后作为 USB 时钟。</p> <p>000: 1.5 倍分频; 001: 不分频; 010: 2.5 倍分频; 011: 2 倍分频; 100: 3.5 倍分频; 101: 3 倍分频; 110: 4 倍分频; 111: 4 倍分频。</p>
位 30: 29 位 21: 18	PLLMULT	0x00	rw	<p>PLL 倍频系数 (PLL multiplication factor) { 位 30: 29, 位 21: 18}</p> <p>000000: 2 倍频      000001: 3 倍频; 000010: 4 倍频      000011: 5 倍频; ..... 001100: 14 倍频      001101: 15 倍频; 001110: 16 倍频      001111: 16 倍频; 010000: 17 倍频      010001: 18 倍频; 010010: 19 倍频      010011: 20 倍频; ..... 100000: 33 倍频      其他: 保留, 请勿使用。</p> <p>注意: PLLRANGE 位须搭配 PLL 倍频后的频率值进行设置</p>
位 17	PLLHEXTDIV	0x0	rw	<p>HEXT 分频后作为 PLL 输入时钟源 (HEXT division selection for PLL entry clock)</p> <p>0: 不分频; 1: 分频, 分频系数由 HEXTDIV 设定。</p>
位 16	PLLRCSC	0x0	rw	<p>PLL 输入时钟选择 (PLL reference clock select)</p> <p>0: HICK 分频时钟 (4MHz) 作为 PLL 输入时钟; 1: HEXT 时钟作为 PLL 输入时钟源。</p>
位 28 位 15: 14	ADCDIV	0x0	rw	<p>ADC 分频因子 (ADC division) PCLK 分频后作为 ADC 时钟。</p> <p>000: 2 分频; 001: 4 分频; 010: 6 分频; 011: 8 分频; 100: 2 分频; 101: 12 分频; 110: 8 分频; 111: 16 分频。</p>
位 13: 11	APB2DIV	0x0	rw	<p>APB2 分频因子 (APB2 division) HCLK 分频后作为 APB2 时钟。</p> <p>0xx: 不分频; 100: 2 分频; 101: 4 分频; 110: 8 分频; 111: 16 分频。</p> <p>注意: 软件必须保证 APB2 时钟频率不超过 120MHz。</p>
位 10: 8	APB1DIV	0x0	rw	<p>APB1 分频因子 (APB1 division) HCLK 分频后作为 APB1 时钟。</p> <p>0xx: 不分频; 100: 2 分频; 101: 4 分频; 110: 8 分频; 111: 16 分频。</p> <p>注意: 软件必须保证 APB1 时钟频率不超过 120MHz。</p>

位 7: 4	AHBDIV	0x0	rw	AHB 分频因子 (AHB division) SCLK 分频后作为 AHB 时钟。 0xxx: 不分频; 1000: 2 分频; 1001: 4 分频; 1010: 8 分频; 1011: 16 分频; 1100: 64 分频; 1101: 128 分频; 1110: 256 分频; 1111: 512 分频。
位 3: 2	SCLKSTS	0x0	ro	系统时钟选择状态位 (System clock select status) 00: HICK; 01: HEXT; 10: PLL; 11: 保留, 保持默认值。
位 1: 0	SCLKSEL	0x0	rw	系统时钟选择 (System clock select) 00: HICK; 01: HEXT; 10: PLL; 11: 保留, 保持默认值。

## 4.3.3 时钟中断寄存器 (CRM\_CLKINT)

域	简称	复位值	类型	功能
位 31: 24	保留	0x00	resd	保持默认值。
位 23	CFDFC	0x0	wo	清除时钟失效标志 (Clock failure detection flag clear) 由软件写'1'清除 CFDF。 0: 不清除; 1: 清除。
位 22: 21	保留	0x0	resd	保持默认值。
位 20	PLLSTBLFC	0x0	wo	清除 PLL 稳定标志 (PLL stable flag clear) 由软件写'1'清除 PLLSTBLF。 0: 不清除; 1: 清除。
位 19	HEXTSTBLFC	0x0	wo	清除 HEXT 稳定标志 (HEXT stable flag clear) 由软件写'1'清除 HEXTSTBLF。 0: 不清除; 1: 清除。
位 18	HICKSTBLFC	0x0	wo	清除 HICK 稳定标志 (HICK stable flag clear) 由软件写'1'清除 HICKSTBLF。 0: 不清除; 1: 清除。
位 17	LEXTSTBLFC	0x0	wo	清除 LEXT 稳定标志 (LEXT stable flag clear) 由软件写'1'清除 LEXTSTBLF。 0: 不清除; 1: 清除。
位 16	LICKSTBLFC	0x0	wo	清除 LICK 稳定标志 (LICK stable flag clear) 由软件写'1'清除 LICKSTBLF。 0: 不清除; 1: 清除。
位 15: 13	保留	0x0	resd	保持默认值。
位 12	PLLSTBLIEN	0x0	rw	PLL 稳定中断使能 (PLL stable interrupt enable) 0: 关闭; 1: 开启。

位 11	HEXTSTBLIEN	0x0	rw	HEXT 稳定中断使能 (HEXT stable interrupt enable) 0: 关闭; 1: 开启。
位 10	HICKSTBLIEN	0x0	rw	HICK 稳定中断使能 (HICK stable interrupt enable) 0: 关闭; 1: 开启。
位 9	LEXTSTBLIEN	0x0	rw	LEXT 稳定中断使能 (LEXT stable interrupt enable) 0: 关闭; 1: 开启。
位 8	LICKSTBLIEN	0x0	rw	LICK 稳定中断使能 (LICK stable interrupt enable) 0: 关闭; 1: 开启。
位 7	CFDF	0x0	ro	时钟失效标志 (Clock Failure Detection flag) 在 HEXT 时钟出现故障时, 由硬件置起。 0: 未出现; 1: 出现。
位 6: 5	保留	0x0	resd	保持默认值。
位 4	PLLSTBLF	0x0	ro	PLL 稳定标志 (PLL stable flag) 由硬件置起。 0: 未稳定; 1: 已稳定。
位 3	HEXTSTBLF	0x0	ro	HEXT 稳定标志 (HEXT stable flag) 由硬件置起。 0: 未稳定; 1: 已稳定。
位 2	HICKSTBLF	0x0	ro	HICK 稳定标志 (HICK stable flag) 由硬件置起。 0: 未稳定; 1: 已稳定。
位 1	LEXTSTBLF	0x0	ro	LEXT 稳定标志 (LEXT stable flag) 由硬件置起。 0: 未稳定; 1: 已稳定。
位 0	LICKSTBLF	0x0	ro	LICK 稳定中断标志 (LICK stable flag) 由硬件置起。 0: 未稳定; 1: 已稳定。

## 4.3.4 APB2外设复位寄存器 (CRM\_APB2RST)

域	简称	复位值	类型	功能
位 31: 27	保留	0x00	resd	保持默认值。
位 26	UART8RST	0x0	rw	UART8 复位 (UART8 reset) 0: 无复位; 1: 复位。
位 25	UART7RST	0x0	rw	UART7 复位 (UART7 reset) 0: 无复位; 1: 复位。
位 24	UART6RST	0x0	rw	UART6 复位 (UART6 reset) 0: 无复位; 1: 复位。
位 23	I2C3RST	0x0	rw	I2C3 复位 (I2C3 reset) 0: 无复位; 1: 复位。
位 22	ACCRST	0x0	rw	ACC 复位 (ACC reset) 0: 无复位; 1: 复位。

位 21	TMR11RST	0x0	rw	TMR11 复位 (TMR11 reset) 0: 无复位; 1: 复位。
位 20	TMR10RST	0x0	rw	TMR10 复位 (TMR10 reset) 0: 无复位; 1: 复位。
位 19	TMR9RST	0x0	rw	TMR9 复位 (TMR9 reset) 0: 无复位; 1: 复位。
位 18: 16	保留	0x0	resd	保持默认值。
位 15	ADC3RST	0x0	rw	ADC3 复位 (ADC3 reset) 0: 无复位; 1: 复位。
位 14	USART1RST	0x0	rw	USART1 复位 (USART1 reset) 0: 无复位; 1: 复位。
位 13	TMR8RST	0x0	rw	TMR 8 复位 (TMR8 reset) 0: 无复位; 1: 复位。
位 12	SPI1RST	0x0	rw	SPI1 复位 (SPI1 reset) 0: 无复位; 1: 复位。
位 11	TMR1RST	0x0	rw	TMR1 复位 (TMR1 reset) 0: 无复位; 1: 复位。
位 10	ADC2RST	0x0	rw	ADC2 复位 (ADC2 reset) 0: 无复位; 1: 复位。
位 9	ADC1RST	0x0	rw	ADC1 复位 (ADC1 reset) 0: 无复位; 1: 复位。
位 8: 7	保留	0x0	resd	保持默认值。
位 6	GPIOERST	0x0	rw	GPIOE 复位 (GPIOE reset) 0: 无复位; 1: 复位。
位 5	GIODRST	0x0	rw	GIOD 复位 (GIOD reset) 0: 无复位; 1: 复位。
位 4	GPIOCRST	0x0	rw	GPIOC 复位 (GPIOC reset) 0: 无复位; 1: 复位。
位 3	GPIOBRST	0x0	rw	GPIOB 复位 (GPIOB reset) 0: 无复位; 1: 复位。
位 2	GPIOARST	0x0	rw	GPIOA 复位 (GPIOA reset) 0: 无复位; 1: 复位。
位 1	EXINTRST	0x0	rw	EXINT 复位 (EXINT reset) 0: 无复位; 1: 复位。 注意软件读该位, 恒为 0。
位 0	IOMUXRST	0x0	rw	IOMUX 复位 (IOMUX reset) 0: 无复位; 1: 复位。

## 4.3.5 APB1外设复位寄存器（CRM\_APB1RST）

域	简称	复位值	类型	功能
位 31: 30 保留		0x0	resd	保持默认值。
位 29	DACRST	0x0	rw	DAC 复位（DAC reset） 0: 无复位； 1: 复位。
位 28	PWCRST	0x0	rw	PWC 复位（PWC reset） 0: 无复位； 1: 复位。
位 27	BPRRST	0x0	rw	电池供电寄存器接口复位（BPR reset） 0: 无复位； 1: 复位。
位 26	CAN2RST	0x0	rw	CAN2 复位（CAN2 reset） 0: 无复位； 1: 复位。
位 25	CAN1RST	0x0	rw	CAN1 复位（CAN1 reset） 0: 无复位； 1: 复位。
位 24	保留	0x0	resd	保持默认值。
位 23	USBRST	0x0	rw	USB 复位（USB reset） 0: 无复位； 1: 复位。
位 22	I2C2RST	0x0	rw	I2C2 复位（I2C2 reset） 0: 无复位； 1: 复位。
位 21	I2C1RST	0x0	rw	I2C1 复位（I2C1 reset） 0: 无复位； 1: 复位。
位 20	UART5RST	0x0	rw	UART5 复位（UART5 reset） 0: 无复位； 1: 复位。
位 19	UART4RST	0x0	rw	UART4 复位（UART4 reset） 0: 无复位； 1: 复位。
位 18	USART3RST	0x0	rw	USART3 复位（USART3 reset） 0: 无复位； 1: 复位。
位 17	USART2RST	0x0	rw	USART2 复位（USART2 reset） 0: 无复位； 1: 复位。
位 16	SPI4RST	0x0	rw	SPI4 复位（SPI4 reset） 0: 无复位； 1: 复位。
位 15	SPI3RST	0x0	rw	SPI3 复位（SPI3 reset） 0: 无作用； 1: 复位。
位 14	SPI2RST	0x0	rw	SPI2 复位（SPI2 reset） 0: 无复位； 1: 复位。
位 13: 12 保留		0x0	resd	保持默认值。
位 11	WWDTRST	0x0	rw	窗口看门狗复位（WWDT reset） 0: 无复位； 1: 复位。

位 10: 9 保留		0x0	resd	保持默认值。
位 8	TMR14RST	0x0	rw	TMR14 复位 (TMR14 reset) 0: 无复位; 1: 复位。
位 7	TMR13RST	0x0	rw	TMR13 复位 (TMR13 reset) 0: 无复位; 1: 复位。
位 6	TMR12RST	0x0	rw	TMR12 复位 (TMR12 reset) 0: 无复位; 1: 复位。
位 5	TMR7RST	0x0	rw	TMR7 复位 (TMR7 reset) 0: 无复位; 1: 复位。
位 4	TMR6RST	0x0	rw	TMR6 复位 (TMR6 reset) 0: 无复位; 1: 复位。
位 3	TMR5RST	0x0	rw	TMR5 复位 (TMR5 reset) 0: 无复位; 1: 复位。
位 2	TMR4RST	0x0	rw	TMR4 复位 (TMR4 reset) 0: 无复位; 1: 复位。
位 1	TMR3RST	0x0	rw	TMR3 复位 (TMR3 reset) 0: 无复位; 1: 复位。
位 0	TMR2RST	0x0	rw	TMR2 复位 (TMR2 reset) 0: 无复位; 1: 复位。

#### 4.3.6 AHB外设时钟使能寄存器 (CRM\_AHBEN)

域	简称	复位值	类型	功能
位 31: 29 保留		0x0	resd	保持默认值。
位 28	EMACPTPEN	0x0	rw	以太网 MAC PTP 时钟使能 (Ethernet MAC PTP clock enable) 0: 关闭; 1: 开启。
位 27: 17 保留		0x0	resd	保持默认值。
位 16	EMACRXEN	0x0	rw	以太网 MAC 接收时钟使能 (Ethernet MAC RX clock enable) 0: 关闭; 1: 开启。
位 15	EMACTXEN	0x0	rw	以太网 MAC 发送时钟使能 (Ethernet MAC TX clock enable) 0: 关闭; 1: 开启。
位 14	EMACEN	0x0	rw	以太网 MAC 时钟使能 (Ethernet MAC clock enable) 0: 关闭; 1: 开启。
位 13: 12 保留		0x0	resd	保持默认值。
位 11	SDIO2EN	0x0	rw	SDIO2 时钟使能 (SDIO2 clock enable) 0: 关闭; 1: 开启。
位 10	SDIO1EN	0x0	rw	SDIO1 时钟使能 (SDIO1 clock enable) 0: 关闭; 1: 开启。



位 9	保留	0x0	rw	保持默认值。
位 8	XMCEN	0x0	rw	XMC 时钟使能 (XMC clock enable) 0: 关闭; 1: 开启。
位 7	保留	0x0	resd	保持默认值。
位 6	CRCEN	0x0	rw	CRC 时钟使能 (CRC clock enable) 0: 关闭; 1: 开启。
位 5	保留	0x0	resd	保持默认值。
位 4	FLASHEN	0x1	rw	闪存时钟使能 (Flash clock enable) 该位配置睡眠或深度睡眠模式下闪存时钟使能。 0: 关闭; 1: 开启。
位 3	保留	0x0	resd	保持默认值。
位 2	SRAMEN	0x1	rw	SRAM 时钟使能 (SRAM clock enable) 该位配置睡眠或深度睡眠模式下 SRAM 时钟使能。 0: 关闭; 1: 开启。
位 1	DMA2EN	0x0	rw	DMA2 时钟使能 (DMA2 clock enable) 0: 关闭; 1: 开启。
位 0	DMA1EN	0x0	rw	DMA1 时钟使能 (DMA1 clock enable) 0: 关闭; 1: 开启。

## 4.3.7 APB2外设时钟使能寄存器 (CRM\_APB2EN)

域	简称	复位值	类型	功能
位 31: 27 保留		0x00	resd	保持默认值。
位 26	UART8EN	0x0	rw	UART8 时钟使能 (UART8 clock enable) 0: 关闭; 1: 开启。
位 25	UART7EN	0x0	rw	UART7 时钟使能 (UART7 clock enable) 0: 关闭; 1: 开启。
位 24	USART6EN	0x0	rw	USART6 时钟使能 (USART6 clock enable) 0: 关闭; 1: 开启。
位 23	I2C3EN	0x0	rw	I2C3 时钟使能 (I2C3 clock enable) 0: 关闭; 1: 开启。
位 22	ACCEN	0x0	rw	ACC 时钟使能 (ACC clock enable) 0: 关闭; 1: 开启。
位 21	TMR11EN	0x0	rw	TMR11 时钟使能 (TMR11 clock enable) 0: 关闭; 1: 开启。
位 20	TMR10EN	0x0	rw	TMR10 时钟使能 (TMR10 clock enable) 0: 关闭; 1: 开启。
位 19	TMR9EN	0x0	rw	TMR9 时钟使能 (TMR9 clock enable) 0: 关闭; 1: 开启。

位 18: 16 保留	0x0	resd	保持默认值。
位 15	ADC3EN	0x0	rw ADC3 使能 (ADC3 clock enable) 0: 关闭; 1: 开启。
位 14	USART1EN	0x0	rw USART1 时钟使能 (USART1 clock enable) 0: 关闭; 1: 开启。
位 13	TMR8EN	0x0	rw TMR8 时钟使能 (TMR8 clock enable) 0: 关闭; 1: 开启。
位 12	SPI1EN	0x0	rw SPI1 时钟使能 (SPI1 clock enable) 0: 关闭; 1: 开启。
位 11	TMR1EN	0x0	rw TMR1 时钟使能 (TMR1 clock enable) 0: 关闭; 1: 开启。
位 10	ADC2EN	0x0	rw ADC2 时钟使能 (ADC2 clock enable) 0: 关闭; 1: 开启。
位 9	ADC1EN	0x0	rw ADC1 时钟使能 (ADC 1 clock enable) 0: 关闭; 1: 开启。
位 8: 7 保留	0x0	rw	保持默认值。
位 6	GPIOEEN	0x0	rw GPIOE 时钟使能 (GPIOE clock enable) 0: 关闭; 1: 开启。
位 5	GPIODEN	0x0	rw GPIOD 时钟使能 (GPIOD clock enable) 0: 关闭; 1: 开启。
位 4	GPIOCEN	0x0	rw GPIOC 时钟使能 (GPIOC clock enable) 0: 关闭; 1: 开启。
位 3	GPIOBEN	0x0	rw GPIOB 时钟使能 (GPIOB clock enable) 0: 关闭; 1: 开启。
位 2	GPIOAEN	0x0	rw GPIOA 时钟使能 (GPIOA clock enable) 0: 关闭; 1: 开启。
位 1 保留	0x0	rw	保持默认值。
位 0	IOMUXEN	0x0	rw IOMUX 时钟使能 (IOMUX clock enable) 0: 关闭; 1: 开启。

## 4.3.8 APB1外设时钟使能寄存器（CRM\_APB1EN）

域	简称	复位值	类型	功能
位 31: 30 保留		0x0	resd	保持默认值。
位 29	DACEN	0x0	rw	DAC 时钟使能（DAC clock enable） 0: 关闭； 1: 开启。
位 28	PWCEN	0x0	rw	PWC 时钟使能（Power control clock enable） 0: 关闭； 1: 开启。
位 27	BPREN	0x0	rw	BPR 时钟使能（BPR clock enable） 0: 关闭； 1: 开启。
位 26	CAN2EN	0x0	rw	CAN2 时钟使能（CAN2 clock enable） 0: 关闭； 1: 开启。
位 25	CAN1EN	0x0	rw	CAN1 时钟使能（CAN1 clock enable） 0: 关闭； 1: 开启。
位 24	保留	0x0	resd	保持默认值。
位 23	USBEN	0x0	rw	USB 时钟使能（USB clock enable） 0: 关闭； 1: 开启。
位 22	I2C2EN	0x0	rw	I2C2 时钟使能（I2C2 clock enable） 0: 关闭； 1: 开启。
位 21	I2C1EN	0x0	rw	I2C1 时钟使能（I2C1 clock enable） 0: 关闭； 1: 开启。
位 20	UART5EN	0x0	rw	UART5 时钟使能（UART5 clock enable） 0: 关闭； 1: 开启。
位 19	UART4EN	0x0	rw	UART4 时钟使能（UART4 clock enable） 0: 关闭； 1: 开启。
位 18	USART3EN	0x0	rw	USART3 时钟使能（USART3 clock enable） 0: 关闭； 1: 开启。
位 17	USART2EN	0x0	rw	USART2 时钟使能（USART2 clock enable） 0: 关闭； 1: 开启。
位 16	SPI4EN	0x0	rw	SPI4 时钟使能（SPI4 clock enable） 0: 关闭； 1: 开启。
位 15	SPI3EN	0x0	rw	SPI3 时钟使能（SPI3 clock enable） 0: 关闭； 1: 开启。
位 14	SPI2EN	0x0	rw	SPI2 时钟使能（SPI2 clock enable） 0: 关闭； 1: 开启。
位 13: 12 保留		0x0	resd	保持默认值。

位 11	WWDTEN	0x0	rw	窗口看门狗时钟使能 (WWDT clock enable) 0: 关闭; 1: 开启。
位 10: 9	保留	0x0	resd	保持默认值。
位 8	TMR14EN	0x0	rw	TMR14 时钟使能 (TMR14 clock enable) 0: 关闭; 1: 开启。
位 7	TMR13EN	0x0	rw	TMR13 时钟使能 (TMR13 clock enable) 0: 关闭; 1: 开启。
位 6	TMR12EN	0x0	rw	TMR12 时钟使能 (TMR12 clock enable) 0: 关闭; 1: 开启。
位 5	TMR7EN	0x0	rw	TMR7 时钟使能 (TMR7 clock enable) 0: 关闭; 1: 开启。
位 4	TMR6EN	0x0	rw	TMR6 时钟使能 (TMR6 clock enable) 0: 关闭; 1: 开启。
位 3	TMR5EN	0x0	rw	TMR5 时钟使能 (TMR5 clock enable) 0: 关闭; 1: 开启。
位 2	TMR4EN	0x0	rw	TMR4 时钟使能 (TMR4 clock enable) 0: 关闭; 1: 开启。
位 1	TMR3EN	0x0	rw	TMR3 时钟使能 (TMR3 clock enable) 0: 关闭; 1: 开启。
位 0	TMR2EN	0x0	rw	TMR2 时钟使能 (TMR2 clock enable) 0: 关闭; 1: 开启。

## 4.3.9 电池供电域控制寄存器 (CRM\_BPDC)

访问: 0 到 3 等待周期; 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、LEXTBYPSS、RTCSEL 和 RTCEN 位处于电池供电域。因此, 这些位在复位后处于写保护状态, 只有在电源控制寄存器 (PWC\_CTRL) 中的 BPWEN 位置位后才能对这些位进行改动。这些位只能由电池供电域软件复位清除。任何内部或外部复位都不会影响这些位。

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

位 9: 8	RTCSEL	0x0	rw	RTC 时钟选择 (RTC clock selection) 确定了 RTC 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 RTC 时钟选择。 00: 无; 01: LEXT; 10: LICK; 11: HEXT/128。
位 7: 3	保留	0x00	resd	保持默认值。
位 2	LEXTBYP	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。

#### 4.3.10 控制/状态寄存器 (CRM\_CTRLSTS)

除复位标志外由系统复位清除, 复位标志能由电源复位或写 RSTFC 位进行清除。访问: 0 到 3 等待周期; 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 管脚复位标志 (NRST reset flag) 该位由硬件置起, 软件写 RSTFC 位清除。 0: 无; 1: 有。
位 25	保留	0x0	resd	保持默认值。
位 24	RSTFC	0x0	rw	清除复位标志 (Reset flag clear) 由软件写'1'来清除复位标志。 0: 无作用; 1: 清除复位标志。
位 23: 2	保留	0x000000	resd	保持默认值。

位 1	LICKSTBL	0x0	ro	LICK 稳定 (LICK stable) 0: 未稳定; 1: 已稳定。
位 0	LICKEN	0x0	rw	LICK 使能 (LICK enable) 0: 关闭; 1: 开启。

#### 4.3.11 AHB外设复位寄存器 (CRM\_AHBRST)

域	简称	复位值	类型	功能
位 31: 15 保留		0x00000	resd	保持默认值。
位 14	EMACRST	0x0	rw	以太网 MAC 复位 (Ethernet MAC reset) 0: 无作用; 1: 复位。
位 13: 0 保留		0x0000	resd	保持默认值。

#### 4.3.12 额外寄存器1 (CRM\_MISC1)

域	简称	复位值	类型	功能
位 31: 28 CLKOUTDIV		0x0	rw	CLKOUT 分频因子 (Clock output division) CLKOUT 输出频率的分频值设定。 0xxx: 不分频; 1000: 2 分频; 1001: 4 分频; 1010: 8 分频; 1011: 16 分频; 1100: 64 分频; 1101: 128 分频; 1110: 256 分频; 1111: 512 分频。
位 27: 26 保留		0x0	resd	保持默认值。
位 25	HICKDIV	0x0	rw	HICK 6 分频 (HICK 6 divider selection) 该位选择使用 HICK 时钟还是 HICK 的 6 分频时钟, 选择 HICK 的 6 分频时钟的话, 时钟频率为 8 MHz, 选择不分频的话, 时钟频率为 48 MHz。 0: 分频; 1: 不分频。 注意: 不论何种情况 HICK 输入到 PLL 时的频率都固定为 4 MHz。
位 24	USBBUFS	0x0	rw	USB 缓冲区大小 (USB buffer size selection) 0: 缓冲区为 512 字节; 1: 缓冲区为 768~1280 字节。
位 23: 17 保留		0x00	resd	保持默认值。
位 16	CLKOUT_SEL[3]	0x0	rw	内部时钟输出选择 (Clock output selection) 搭配时钟配置寄存器 (CRM_CFG) 位 26: 24 使用。
位 15: 8 保留		0x00	resd	保持默认值。
位 7: 0	HICKCAL_KEY	0x00	rw	HICKCAL 写入键值 (HICK calibration key) 此字段为 0x5A 时, HICKCAL [7: 0]才可被写入。

## 4.3.13 额外寄存器2 (CRM\_MISC2)

域	简称	复位值	类型	功能
位 31: 17 保留		0x0000	resd	保持默认值。
位 16	CLK_TO_TMR	0x0	rw	CLKOUT 连接到 TMR10 的 channel 1 选择 (CLKOUT internal connect to timer 10 channel 1) 0: 未连接; 1: 连接。
位 15: 0 保留		0x0000	resd	保持默认值。

## 4.3.14 额外寄存器3 (CRM\_MISC3)

域	简称	复位值	类型	功能
位 31: 16 保留		0x0000	resd	保持默认值。
位 15	EMAC_PPS_SEL	0x0	rw	EMAC PPS 输出脉宽选择位 (Ethernet MAC pulse width select) 0: 输出脉宽为 125 ms; 1: 输出脉宽为 1 个系统时钟 (1 个 SCLK)。
位 14	保留	0x0	resd	保持默认值。
位 13: 12 HEXTDIV		0x0	rw	HEXT 分频作为 PLL 时钟源时的分频因子 (HEXT division) 00: 2 分频; 01: 3 分频; 10: 4 分频; 11: 5 分频。 注意: 请在 HEXT 时钟稳定后再配置分频控制位, 否则会忽略当前这次的配置。
位 11: 10 保留		0x0	resd	保持默认值。
位 9	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: 根据 HICKDIV 设定可能为 48M 或 8M。
位 8	HICK_TO_USB	0x0	rw	USB 48M 时钟源选择位 (USB 48MHz clock source select) 0: USB 48M 时钟源是 PLL 或是其分频; 1: USB 48M 时钟源来自 HICK 或是其 6 分频。 注意: 由于 USB 必须工作在 48M, 此时必须保证 HICKDIV=1, 使 USB 48M 时钟选择 HICK 的 48MHz 输出。
位 7: 6 保留		0x0	resd	保持默认值。



自动滑顺频率切换（auto step system clock switch enable） 为使切换系统时钟源到 PLL 或是切换 AHB 预分频由大到小时平顺(系统频率由小变大)，建议系统频率操作目标大于 108MHz 时启动自动滑顺频率切换。 当自动滑顺频率切换功能作用时，硬件会暂停 AHB 总线，直到整个自动滑顺频率切换完成才恢复。此期间 DMA 仍正常工作，中断事件会被记忆并待 AHB 总线恢复后 NVIC 即可处理。 00：关闭； 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 5: 4	AUTO_STEP_EN	0x0	rw	
位 3: 0	保留	0xd	resd	固定为 0xd，请勿修改。

## 4.3.15 中断映射寄存器（CRM\_INTMAP）

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	USBINTMAP	0x0	rw	USBFS 模块中断重映射（USBFS interrupt remap） 0：USBFS 使用 19 号 USBFS_H 中断和 20 号 USBFS_L 中断； 1：USBDEV 使用 73 号 USBFS_MAPH 中断和 74 号 USBFS_MAPL 中断。

## 5 闪存控制器（FLASH）

### 5.1 FLASH介绍

闪存由主存储器、外部存储器、信息块、闪存寄存器这四个部分组成。

- 主存储器容量高达 1024K 字节，分为第 1 片闪存（bank1）和第 2 片闪存（bank2）
- 外部存储器容量最高可达 16M 字节
- 信息块由 16K 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 或者 USB（DFU）接口实现 ISP 编程

1024K 字节容量的主存储器分为片 1 和片 2 闪存，每片闪存容量为 512K 字节，每片闪存包含 256 扇区，每扇区大小为 2K 字节。

外部存储器容量可高达 16M 字节，包含 4096 扇区，每扇区大小为 4K 字节。

表 5-1 闪存存储结构（1024K）

结构	名称	地址范围
主存储器	片 1 (Bank1) 512KB	扇区 0
		0x0800 0000 – 0x0800 07FF
		扇区 1
		0x0800 0800 – 0x0800 0FFF
	片 2 (Bank2) 512KB	扇区 2
		0x0800 1000 – 0x0800 17FF
		...
		扇区 255
		0x0807 F800 – 0x0807 FFFF
		扇区 256
		0x0808 0000 – 0x0808 07FF
外部存储器	16MB	扇区 257
		0x0808 0800 – 0x0808 0FFF
		扇区 258
		0x0808 1000 – 0x0808 17FF
		...
		扇区 511
信息块		0x080F F800 – 0x080F FFFF
		扇区 0
		0x0840 0000 – 0x0840 0FFF
		扇区 1
		0x0840 1000 – 0x0840 1FFF
		扇区 2
		0x0840 2000 – 0x0840 2FFF
		...
		扇区 4095
		0x093F F000 – 0x093F FFFF
		启动程序代码区 16KB
		0x1FFF B000 – 0x1FFF EFFF
		用户系统数据区 48B
		0x1FFF F800 – 0x1FFF F82F

512K 字节容量的主存储器只有片 1 闪存，包含 256 扇区，每扇区大小为 2K 字节。

外部存储器容量可高达 16M 字节，包含 4096 扇区，每扇区大小为 4K 字节。

表 5-2 闪存存储组织（512K）

结构	名称	地址范围
主存储器	片 1 (Bank1) 512KB	扇区 0
		0x0800 0000 – 0x0800 07FF
		扇区 1
		0x0800 0800 – 0x0800 0FFF
		扇区 2
		0x0800 1000 – 0x0800 17FF
外部存储器	16MB	...
		扇区 255
		0x0807 F800 – 0x0807 FFFF
		扇区 0
		0x0840 0000 – 0x0840 0FFF
		扇区 1
信息块		0x0840 1000 – 0x0840 1FFF
		扇区 2
		0x0840 2000 – 0x0840 2FFF
		...
		扇区 4095
		0x093F F000 – 0x093F FFFF
		启动程序代码区 16KB
		0x1FFF B000 – 0x1FFF EFFF
		用户系统数据区 48B
		0x1FFF F800 – 0x1FFF F82F

256K 字节容量的主存储器只有片 1 闪存，包含 128 扇区，每扇区大小为 2K 字节。

外部存储器容量可高达 16M 字节，包含 4096 扇区，每扇区大小为 4K 字节。

表 5-3 闪存存储组织（256K）

结构	名称	地址范围
主存储器	扇区 0	0x0800 0000 – 0x0800 07FF
	扇区 1	0x0800 0800 – 0x0800 0FFF
	扇区 2	0x0800 1000 – 0x0800 17FF
	...	...
	扇区 127	0x0803 F800 – 0x0803 FFFF
外部存储器	扇区 0	0x0840 0000 – 0x0840 0FFF
	扇区 1	0x0840 1000 – 0x0840 1FFF
	扇区 2	0x0840 2000 – 0x0840 2FFF
	...	...
	扇区 4095	0x093F F000 – 0x093F FFFF
信息块	启动程序代码区 16KB	0x1FFF B000 – 0x1FFF EFFF
	用户系统数据区 48B	0x1FFF F800 – 0x1FFF F82F

**外部存储器**

外部存储器透过 SPIM 传输接口控制外部 SPI 闪存，支持密文保护功能，可通过用户系统数据区的 EXT\_FLASH\_KEYx 字节决定数据是否加密，并由闪存解密地址寄存器（FLASH\_DA）控制加密范围。AHB 时钟（HCLK）是 SPIM 的参考时钟。透过 SPIM 传输接口向外部 SPI 闪存提供 HCLK/2 的时钟。SPIM = 外部 SPI Flash memory 扩展（程序执行/数据储存/程序与数据可加密）。

**注意：** 外部闪存模块只支持字或半字的操作。

图 5-1 外部存储器密文保护

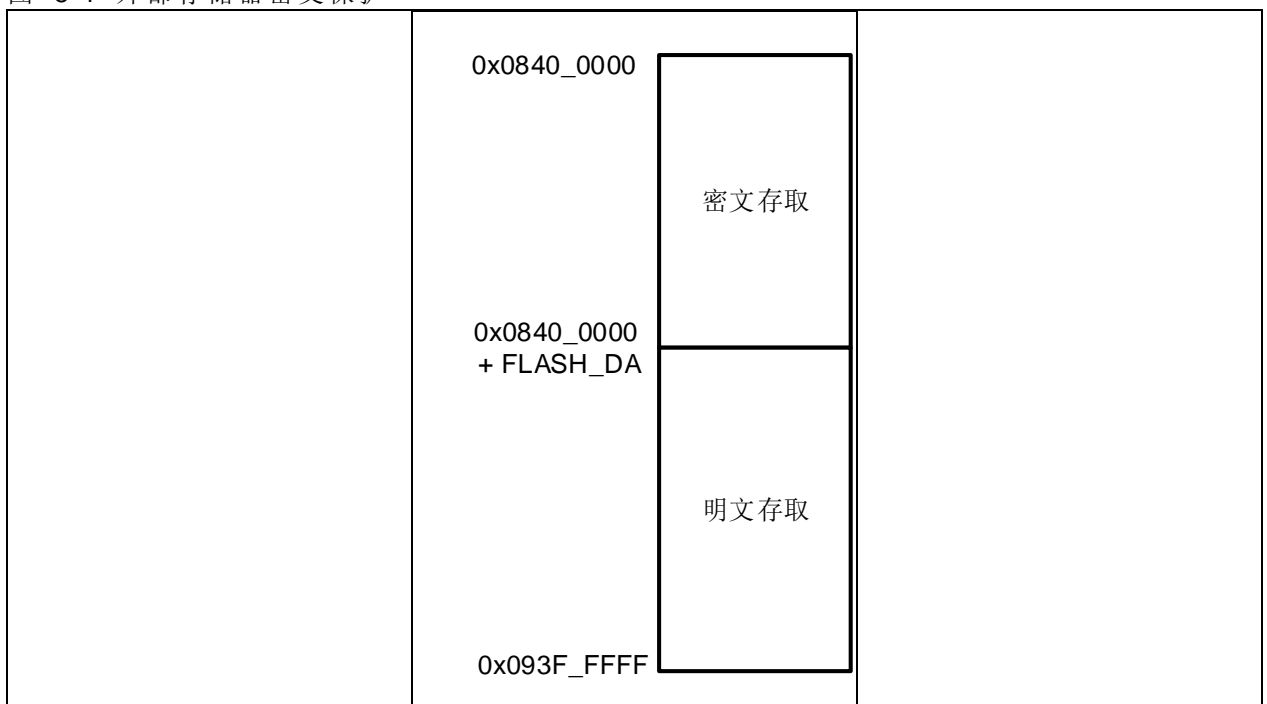


图 5-2 外部存储器参考电路

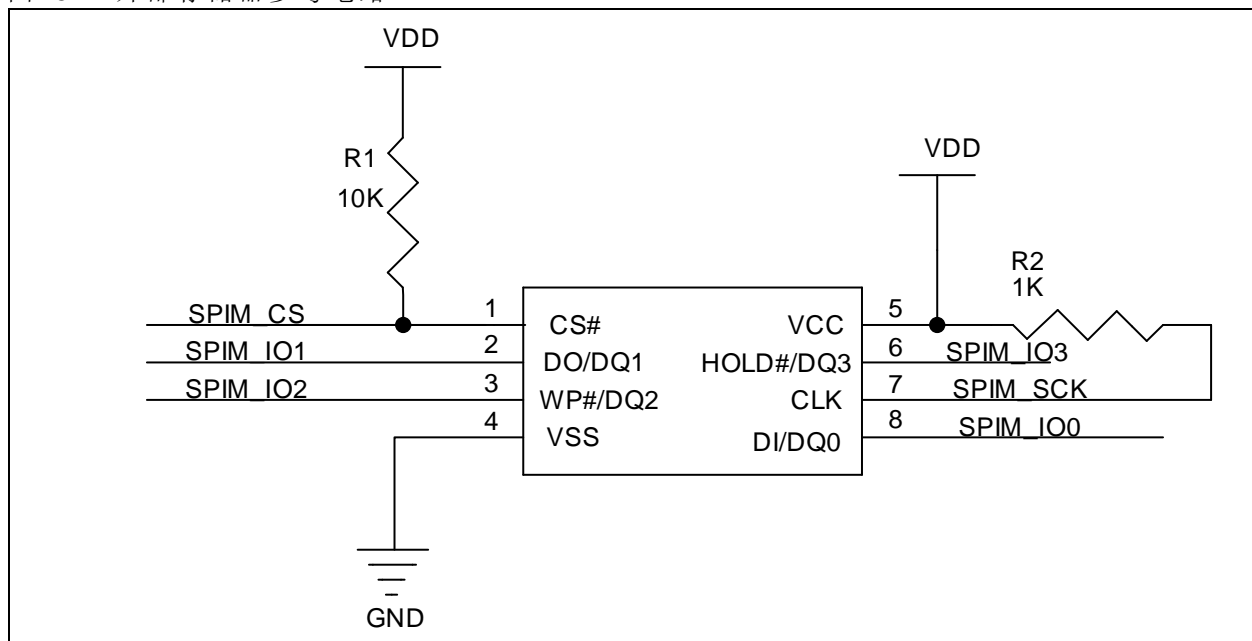


表 5-4 外部存储器支持的指令集

指令名称	指令码	FLASH_SELECT 寄存器配置	补充说明
Write Enable	0x06	0x1/0x2	2 类型号闪存均需要支持 0x06 指令
Quad Page Program	0x32	0x1/0x2	2 类型号闪存均需要支持 0x32 指令
Sector Erase	0x20	0x1/0x2	2 类型号闪存均需要支持 0x20 指令
Chip Erase	0xC7	0x1/0x2	2 类型号闪存均需要支持 0xC7 指令
Read Status Register	0x05	0x1/0x2	2 类型号闪存均需要支持 0x05 指令
Quad I/O Read	0xEB	0x1/0x2	2 类型号闪存均需要支持 0xEB 指令 24bit Addr + 6 个 Dummy cycle
Volatile status Register write enable	0x50	0x1	3 条指令用于当选择型号 1 闪存时，硬体自动发送指令配置闪存 Status Register 中的 Quad Enable（QE 位）  型号 1 闪存需要支持： 0x50 与 0x01 或是支持 0x50 与 0x31
Write Status Register-1	0x01		
Write Status Register-2	0x31		

注意：

- 1.当闪存在执行 0x32 以及 0xEB 指令前，强制要求必须设定闪存 QE 位，并且 QE 位是在 Status Register 的位 1。则可以配置闪存选择寄存器（FLASH\_SELECT）为 0x1，选择型号 1 闪存
- 2.当闪存在执行 0x32 以及 0xEB 指令前不要求设定闪存 QE 位，则可以配置闪存选择寄存器（FLASH\_SELECT）为 0x2，选择型号 2 闪存

举例：

闪存选择寄存器（FLASH\_SELECT）设定 0x1：

可支援 GD25Q127C、GD25Q64C、GD25Q32C、GD25Q16C、GD25Q80C 闪存、W25Q128V 闪存等

闪存选择寄存器（FLASH\_SELECT）设定 0x2：

可支援 EN25F20A、EN25QH128A 闪存

### 用户系统数据区

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

每个系统数据实际占用 2 个字节，低字节对应系统数据的内容，高字节对应系统数据的反码，用于验证选择位的正确性。当读出的高字节不等于低字节的反码时（高字节及低字节均为 0xFF 时除外），系统数据装载器会产生一个系统数据错误的标志（USDERR），并把对应的系统数据及其反码都强置为 0xFF。  
**注意：用户系统数据内容的更新需要一次系统复位才能真正实现。**

表 5-5 用户系统数据说明

地址	位	内容	
0x1FFF_F800	[7: 0]	FAP[7: 0]: 闪存访问保护（访问保护启动/解除结果存放在用户系统数据寄存器（FLASH_USD）[1]） 0xA5: 闪存访问保护解除 其他值: 闪存访问保护启动	
	[15: 8]	nFAP[7: 0]: FAP[7: 0]的反码	
	[23: 16]	SSB[7: 0]: 系统配置字节（存放在用户系统数据寄存器（FLASH_USD）[9: 2]）	
		位 7: 4	保留不用
		位 3（BTOPT）	0: 当配置从主闪存启动时，若片 2 中没有启动程序，将从片 1 启动，否则，从片 2 启动 1: 当配置从主闪存启动时，从片 1 启动
		位 2（nSTDBY_RST）	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 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（存放在用户系统数据寄存器（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（存放在用户系统数据寄存器（FLASH_USD）[25: 18]）	
	[31: 24]	nData1[7: 0]: Data1[7: 0]的反码	
0x1FFF_F808	[7: 0]	EPP0[7: 0]: 闪存擦写保护字节 0（存放在擦除编程保护状态寄存器（FLASH_EPPS）[7: 0]） 用于保护主闪存存储器的扇区 0 ~ 扇区 15，每个比特位保护 2 个扇区（2K 字节/扇区） 0: 擦写保护启动 1: 擦写保护解除	
	[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（存放在擦除编程保护状态寄存器（FLASH_EPPS）[15: 8]） 用于保护主闪存存储器的扇区 16 ~ 扇区 31，每个比特位保护 2 个扇区（2K 字节/扇区） 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	
0x1FFF_F80C	[7: 0]	EPP2[7: 0]: 闪存擦写保护字节 2（存放在擦除编程保护状态寄存器（FLASH_EPPS）[23: 16]） 用于保护主闪存存储器的扇区 32 ~ 扇区 47，每个比特位保护 2 个扇区（2K 字节/扇区） 0: 擦写保护启动 1: 擦写保护解除	
	[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（存放在擦除编程保护状态寄存器（FLASH_EPPS）[31: 24]） 其中位 6: 0 用于保护主闪存存储器的扇区 48 ~ 扇区 61，每个比特位保护 2 个扇区（2K 字节/扇区） 位 7 用于保护主闪存存储器的扇区 62 及之后的扇区，以及外部存储器 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]	EOPB0[7: 0]: 扩充的系统选项 位 7: 1: 保留不用 位 0: 0: 片上 SRAM 224K 字节 1: 片上 SRAM 96K 字节 注意: 由 1 改写成 0 只能在安全库区未启动的情况下实现
	[15: 8]	nEOPB0[7: 0]: EOPB0[7: 0]的反码
	[31: 16]	保留不用
0x1FFF_F814	[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_F818	[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_F81C	[7: 0]	Data6[7: 0]: 用户数据 6
	[15: 8]	nData6[7: 0]: Data6[7: 0]的反码
	[23: 16]	Data7[7: 0]: 用户数据 7
	[31: 24]	nData7[7: 0]: Data7[7: 0]的反码
0x1FFF_F820	[7: 0]	EXT_FLASH_KEY0[7: 0]: 外部存储器密文存取区加密键值字节 0 不加密的设定条件包括: EXT_FLASH_KEYx 以及 nEXT_FLASH_KEYx 均为 0xFF (即默认擦除状态) EXT_FLASH_KEYx 写入 0x00 EXT_FLASH_KEYx 写入 0xFF 即{nEXT_FLASH_KEYx, EXT_FLASH_KEYx }均设为 0xFFFF, 0xFF00, 0x00FF
	[15: 8]	nEXT_FLASH_KEY0[7: 0]: EXT_FLASH_KEY0[7: 0]的反码
	[23: 16]	EXT_FLASH_KEY1[7: 0]: 外部存储器密文存取区加密键值字节 1
	[31: 24]	nEXT_FLASH_KEY1[7: 0]: EXT_FLASH_KEY1[7: 0]的反码
0x1FFF_F824	[7: 0]	EXT_FLASH_KEY2[7: 0]: 外部存储器密文存取区加密键值字节 2
	[15: 8]	nEXT_FLASH_KEY2[7: 0]: EXT_FLASH_KEY2[7: 0]的反码
	[23: 16]	EXT_FLASH_KEY3[7: 0]: 外部存储器密文存取区加密键值字节 3
	[31: 24]	nEXT_FLASH_KEY3[7: 0]: EXT_FLASH_KEY3[7: 0]的反码
0x1FFF_F828	[7: 0]	EXT_FLASH_KEY4[7: 0]: 外部存储器密文存取区加密键值字节 4
	[15: 8]	nEXT_FLASH_KEY4[7: 0]: EXT_FLASH_KEY4[7: 0]的反码
	[23: 16]	EXT_FLASH_KEY5[7: 0]: 外部存储器密文存取区加密键值字节 5
	[31: 24]	nEXT_FLASH_KEY5[7: 0]: EXT_FLASH_KEY5[7: 0]的反码
0x1FFF_F82C	[7: 0]	EXT_FLASH_KEY6[7: 0]: 外部存储器密文存取区加密键值字节 6
	[15: 8]	nEXT_FLASH_KEY6[7: 0]: EXT_FLASH_KEY6[7: 0]的反码
	[23: 16]	EXT_FLASH_KEY7[7: 0]: 外部存储器密文存取区加密键值字节 7
	[31: 24]	nEXT_FLASH_KEY7[7: 0]: EXT_FLASH_KEY7[7: 0]的反码

## 5.2 主存储器操作

### 5.2.1 解锁/锁定

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

**解锁流程:**

对闪存解锁寄存器 x (FLASH\_UNLOCKx) 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 x (FLASH\_CTRLx) 中的 OPLK 位, 锁定对应区域闪存。

### 5.2.2 擦除

编程之前必须先进行擦除操作，主存储器有扇区擦除和整片擦除两种擦除方式。

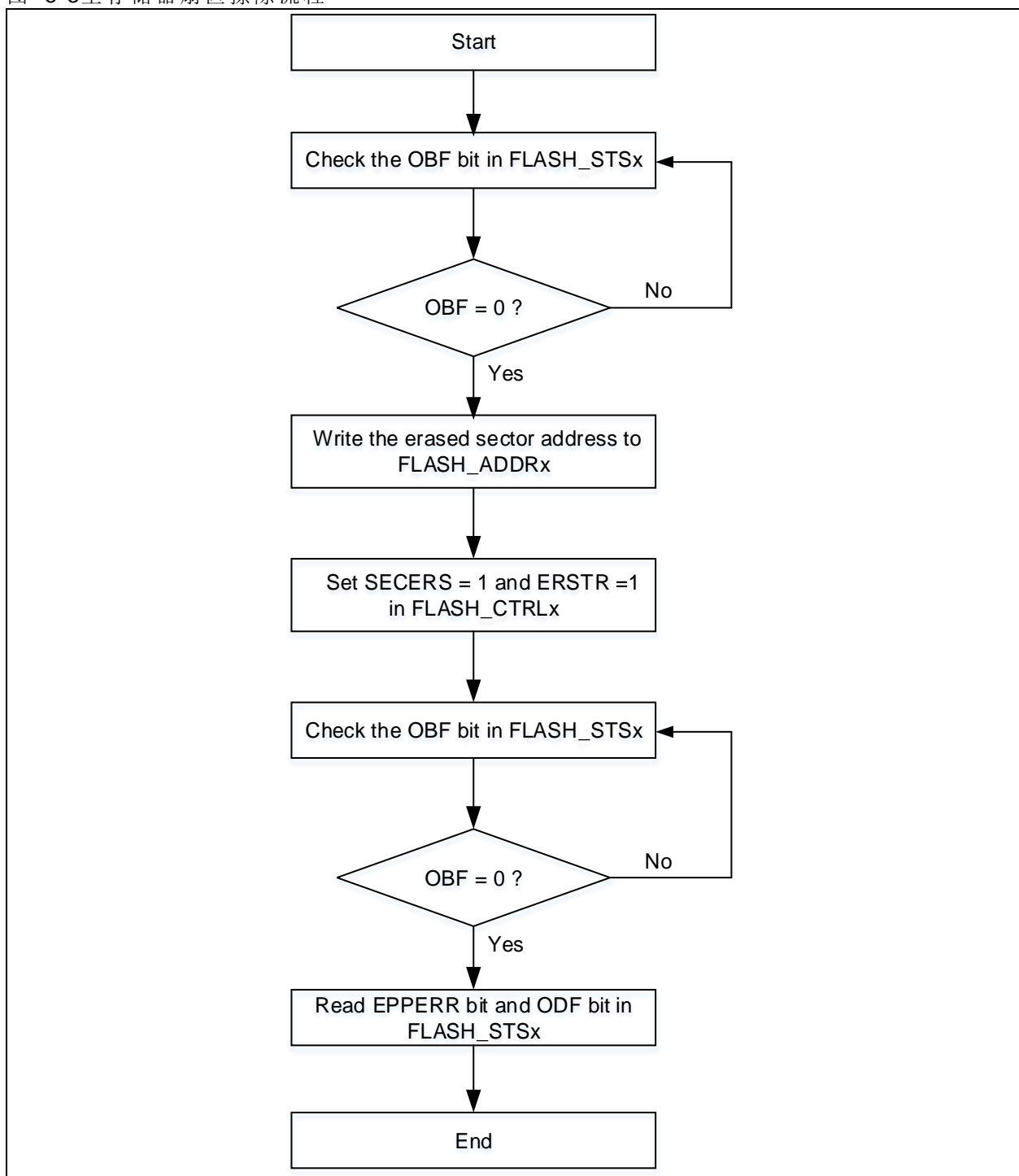
#### 扇区擦除

主闪存存储器的每一扇区都可以使用扇区擦除功能独立擦除。

擦除流程如下：

- 检查闪存状态寄存器  $x$  (FLASH\_STS $x$ ) 的 OBF 位，确认没有正在进行的闪存操作；
- 对闪存地址寄存器  $x$  (FLASH\_ADDR $x$ ) 写入要擦除的扇区地址；
- 对闪存控制寄存器  $x$  (FLASH\_CTRL $x$ ) 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  $x$  (FLASH\_STS $x$ ) 的 OBF 位变为‘0’，并查询闪存状态寄存器  $x$  (FLASH\_STS $x$ ) 的 EPPERR 位和 ODF 位，确认擦除结果。

图 5-3主存储器扇区擦除流程





### 整片擦除

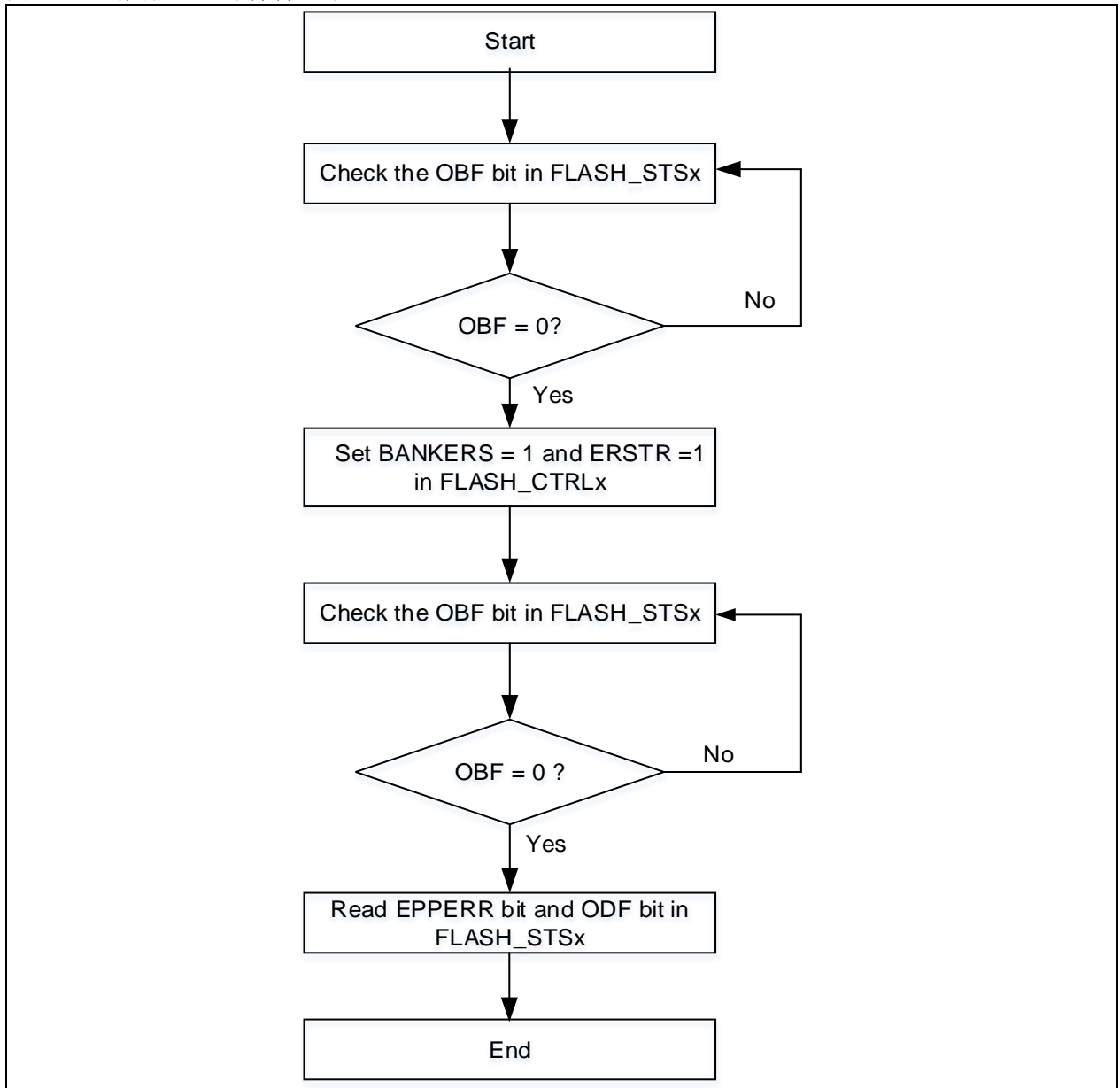
主闪存存储器的每一片闪存区都可以使用整片擦除功能独立擦除。

擦除流程如下：

- 检查闪存状态寄存器 x (FLASH\_STSx) 的 OBF 位，确认没有正在进行的闪存操作；
- 对闪存控制寄存器 x (FLASH\_CTRLx) 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 x (FLASH\_STSx) 的 OBF 位变为‘0’，并查询闪存状态寄存器 x (FLASH\_STSx) 的 EPPERR 位和 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。

图 5-4 主存储器整片擦除流程



### 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：

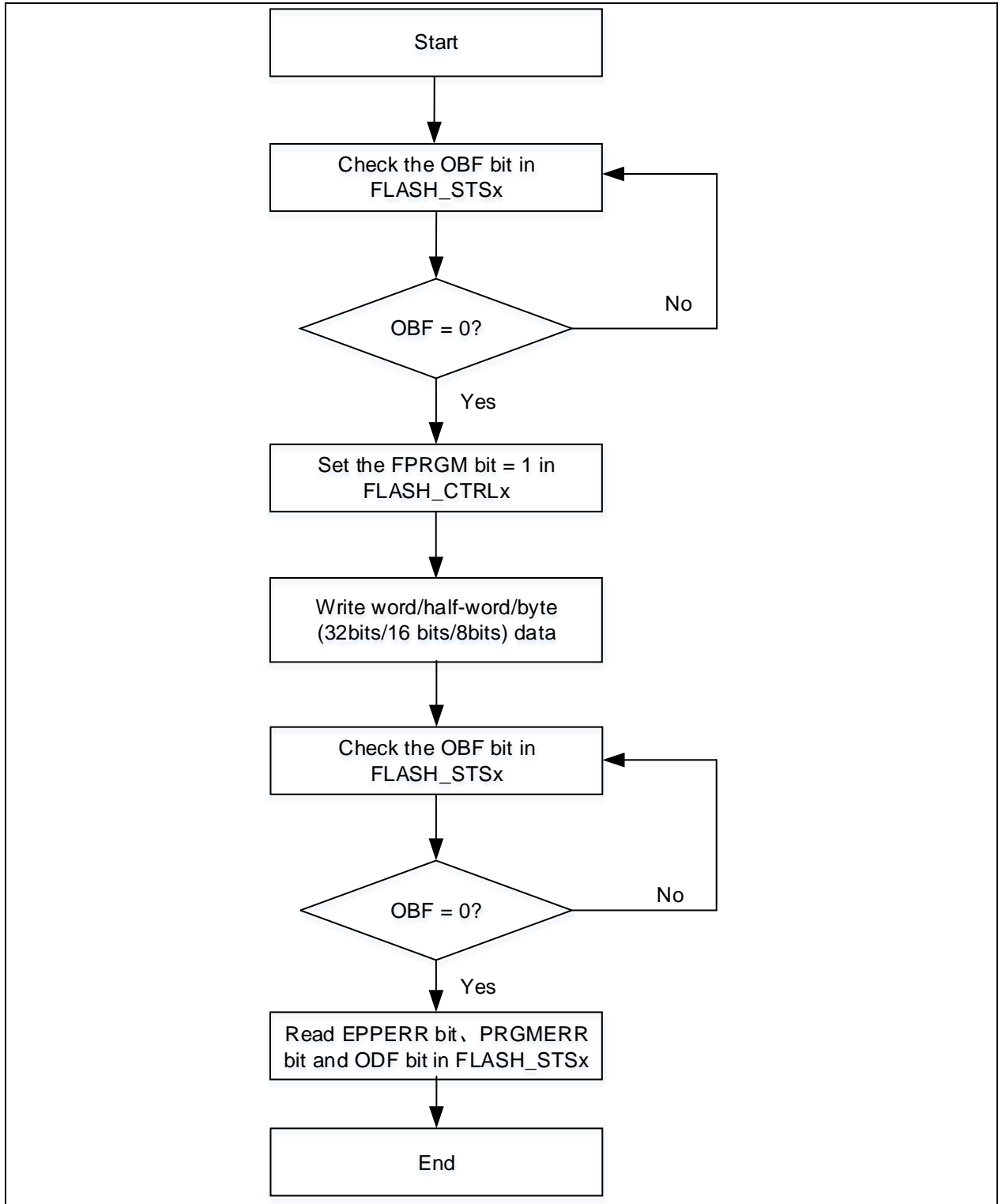
- 检查闪存状态寄存器 x (FLASH\_STSx) 的 OBF 位，确认没有正在进行的闪存操作；
- 对闪存控制寄存器 x (FLASH\_CTRLx) 的 FPRGM 位置 1，此时可以接受对主闪存的编程指令；
- 对指定的地址写入要编程的数据（任意字/半字/字节）；

- 等待闪存状态寄存器 x (FLASH\_STSx) 的 OBF 位变为‘0’，并查询闪存状态寄存器 x (FLASH\_STSx) 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 x (FLASH\_STSx) 的 PRGMERR 位来告知编程发生错误。

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

图 5-5 主存储器编程流程



## 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 外部存储器操作

外部存储器的操作方法，包括读取、解锁、擦除、编程都跟主存储器相同，唯一区别是外部存储器编程只支持 32 位和 16 位操作，不支持 8 位操作。

## 5.4 用户系统数据区操作

### 5.4.1 解锁/锁定

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

**解锁流程：**

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

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

**锁定流程：**

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

### 5.4.2 擦除

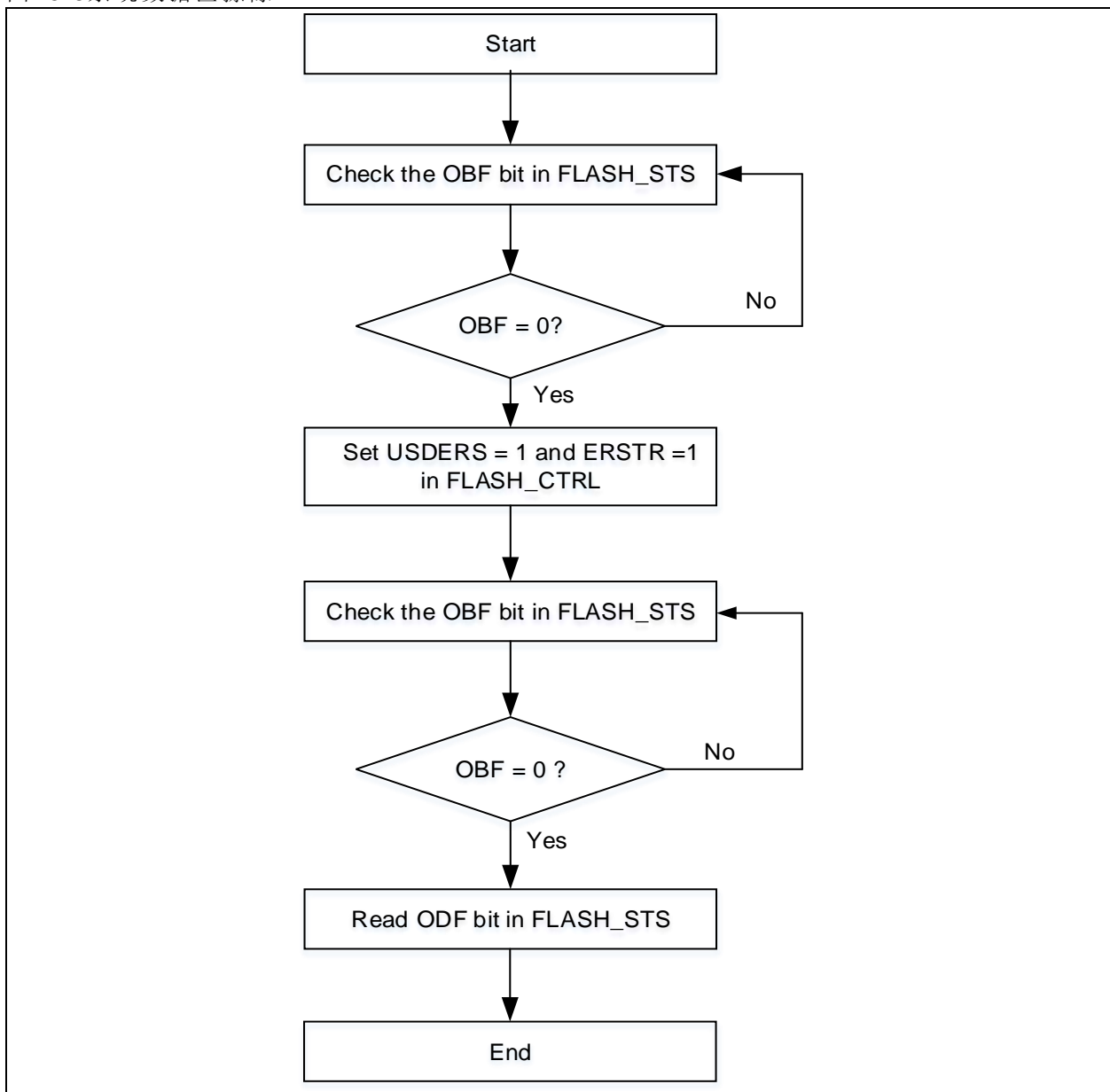
在编程之前必须先进行擦除操作，用户系统数据区域可单独实现擦除功能。

擦除流程如下：

- 检查闪存状态寄存器(FLASH\_STS)的 OBF 位，确认没有正在进行的闪存操作；
- 对闪存控制寄存器(FLASH\_CTRL)的 USDERS 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。*

图 5-6系统数据区擦除



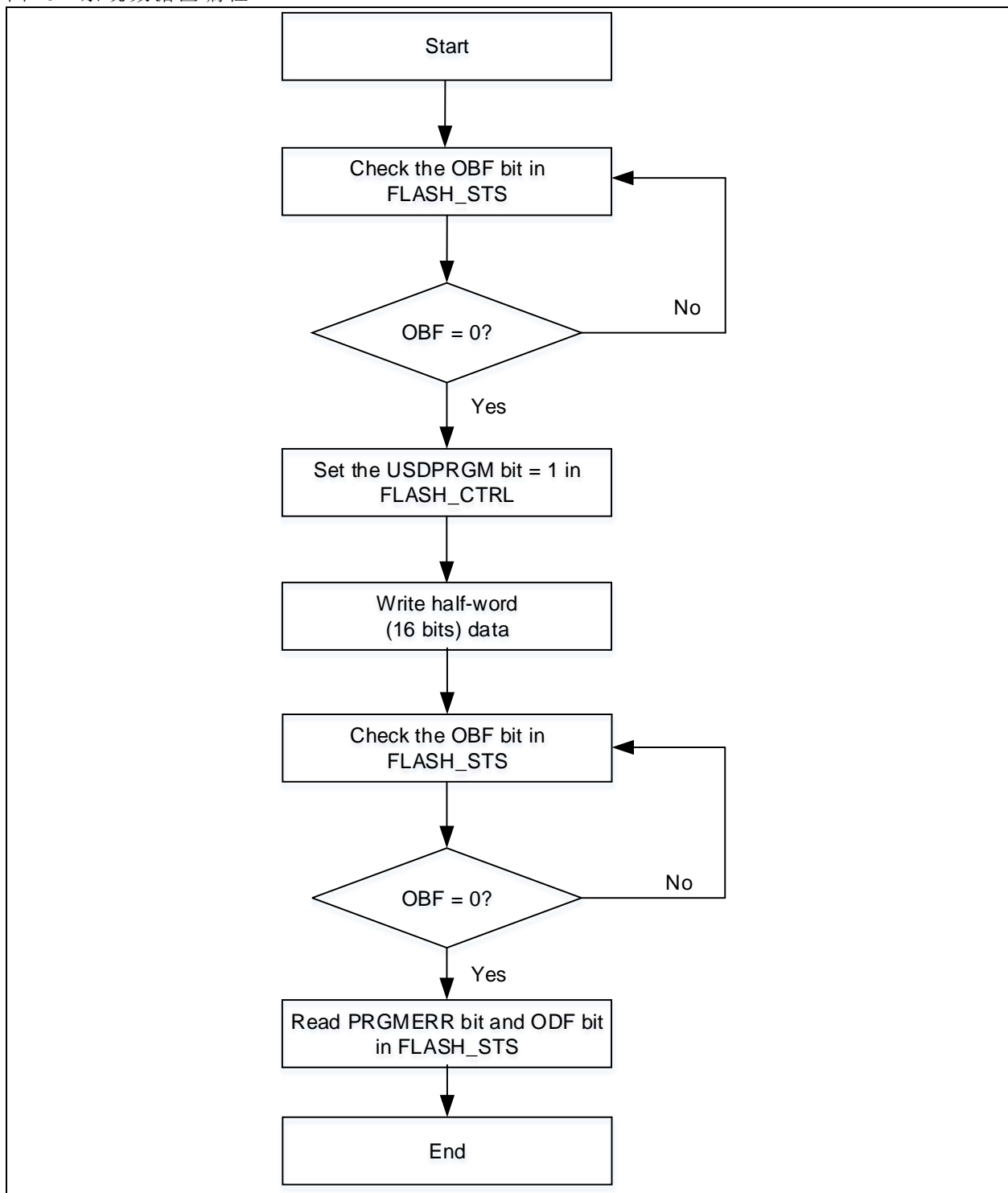
### 5.4.3 编程

当想要改写用户系统数据区域的内容时，可以通过用户系统数据区编程流程完成一次写入 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 USDEERS 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：编程期间进行读闪存的操作，将导致 CPU 会被暂停直到编程完成才处理读闪存操作。*

图 5-7 系统数据区编程



#### 5.4.4 读取

通过 CPU 的 AHB 总线可以直接寻址访问用户系统数据区。

### 5.5 闪存保护

闪存存储器有访问保护以及擦写保护两种保护方式。

#### 5.5.1 访问保护

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

当闪存访问保护启动后，用户可以重新擦除系统数据区，并对 FAP 字节写入 0xA5 解除闪存访问保护（从保护状态变为未保护状态，将自动产生对主存储器的整片擦除操作），最后进行系统复位，系统数据装载机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）。

**注意：**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 5-6 闪存访问权限

区域	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	禁止		禁止 (1) (2)	允许		
外部存储区	禁止		禁止 (2)	允许		
用户系统数据区	禁止	允许		允许		

(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，允许片1、片2擦除以及外部存储器全擦除

## 5.5.2 擦写保护

在 256K 及以上容量的闪存中，擦写保护的基本单位为 2 扇区。通过擦写保护可以防止程序在跑飞时闪存存储器的内容被意外更改。

在下面列出的情况下，擦写将不被允许，并会置位 EPPERR 位：

- 对被设置为擦写保护的扇区（主闪存以及外部存储器）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片 1、片 2 以及外部存储器做整片擦除将不被允许
- 闪存访问保护启动后，主闪存扇区 0 及扇区 1 将被自动擦写保护，不允许做扇区擦除操作以及编程操作
- 闪存访问保护启动后，主存储器在调试模式或是从非主闪存存储器启动下被自动擦写保护，不允许做扇区擦除操作以及编程操作

## 5.6 特殊功能

### 5.6.1 安全库区设定

设定以密码保护主存中指定范围的程式区，即安全库区，此区域仅能被执行，无法读取（I-Code, D-Code 总线除外），以及写入与删除，除非输入指定密码。安全库区划分为 指令安全库区 与 数据安全库区，并可选部分或是整个安全库区存放指令，但不支持整个安全库区存放数据。

**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：**只可在主存中设置安全库区；

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；

中断向量表会被放置在闪存的第一扇扇区(扇区 0)内，请勿将闪存的第一扇扇区设定为安全库区；

要被安全库区保护的程序代码，不可放置在闪存的第一扇区内；

仅允许 I-Code 总线读取指令安全库区；

仅允许 D-Code 总线读取数据安全库区；

写入或删除安全库区代码，将在闪存状态寄存器（FLASH\_STS）的 EPPERR 位置

‘1’ 提出警告；

执行主存的整片擦除时，将不会擦除安全库区。

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

以扇区为单位对安全库区代码进行可选的 CRC 校验。

启动主存安全库区的流程如下：

- 检查闪存状态寄存器（FLASH\_STS）的 OBF 位，以确认没有其他正在进行的编程操作；
- 对闪存安全库区解锁寄存器（SLIB\_UNLOCK）写入 0xA35F6D24，以进行安全库区解锁；
- 检查闪存安全库区额外状态寄存器（SLIB\_MISC\_STS）的 SLIB\_ULKF 位，以确认解锁成功；
- 在闪存安全库区地址设定寄存器（SLIB\_SET\_RANGE）设定要保护的区域，包含指令区与数据区的地址；
- 等待 OBF 位变为‘0’；
- 在闪存安全库区密码设定寄存器（SLIB\_SET\_PWD）设定安全区域密码；
- 等待 OBF 位变为‘0’；
- 烧录将存入安全库区的代码；
- 进行系统复位，重装载安全库区设定字；
- 读出闪存安全库区状态寄存器 0（SLIB\_STS0）/闪存安全库区状态寄存器 1（SLIB\_STS1）用于判断安全库区设定结果。

注意：启动安全库区的流程需要在闪存访问保护未启动时执行。

解除安全库区的流程是：

- 在闪存安全库区密码清除寄存器（SLIB\_PWD\_CLR）写入先前设置的安全区域密码；
- 等待 OBF 位变为‘0’；
- 进行系统复位，重装载安全库区设定字；
- 读出闪存安全库区状态寄存器 0（SLIB\_STS0）用于判断安全库区解除结果。

注意：解除安全库区将会自动执行主存储器的整片擦除，以及安全库设定块擦除。

## 5.7 FLASH寄存器

下表列出了 FLASH 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 5-7 闪存接口—寄存器映像和复位值

寄存器简称	基址偏移量	复位值
FLASH_PSR	0x00	0x0000 0030
FLASH_UNLOCK	0x04	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
FLASH_UNLOCK2	0x44	0xFFFF XXXX
FLASH_STS2	0x4C	0x0000 0000
FLASH_CTRL2	0x50	0x0000 0080



FLASH_ADDR2	0x54	0x0000 0000
FLASH_UNLOCK3	0x84	0xFFFF XXXX
FLASH_SELECT	0x88	0x0000 0000
FLASH_STS3	0x8C	0x0000 0000
FLASH_CTRL3	0x90	0x0000 0080
FLASH_ADDR3	0x94	0x0000 0000
FLASH_DA	0x98	0x0000 0000
SLIB_STS0	0xCC	0x0000 0000
SLIB_STS1	0xD0	0x0000 0000
SLIB_PWD_CLR	0xD4	0x0000 0000
SLIB_MISC_STS	0xD8	0x0100 0000
SLIB_SET_PWD	0xDC	0x0000 0000
SLIB_SET_RANGE	0xE0	0x0000 0000
SLIB_UNLOCK	0xF0	0x0000 0000
FLASH_CRC_CTRL	0xF4	0x0000 0000
FLASH_CRC_CHKR	0xF8	0x0000 0000

### 5.7.1 闪存性能选择寄存器 (FLASH\_PSR)

域	简称	复位值	类型	功能
位 31: 0	保留	0x0000 0030	resd	保持默认值。

### 5.7.2 闪存解锁寄存器 (FLASH\_UNLOCK)

专用于闪存片 1。

域	简称	复位值	类型	功能
位 31: 0	UKVAL	0xFFFF XXXX	wo	解锁键值。(Unlock key value) 该寄存器用于解锁片 1。

注意：所有这些位是只写的，读出时返回 0。

### 5.7.3 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

### 5.7.4 闪存状态寄存器 (FLASH\_STS)

专用于闪存片 1。

域	简称	复位值	类型	功能
位 31: 6	保留	0x00000000	resd	保持为默认值
位 5	ODF	0x0	rw1c	操作完成标志 (Operation done flag) 当闪存操作 (编程/擦除) 成功完成时，硬件会置起该位，软件写'1'可以清除。
位 4	EPPERR	0x0	rw1c	擦写保护错误 (Erase/Program protection error) 当擦除或编程的闪存地址在擦写保护设定范围内时，硬件会置起该位，软件写'1'可以清除。
位 3	保留	0x0	resd	保持为默认值
位 2	PRGMERR	0x0	rw1c	编程错误 (Program error) 当编程的闪存地址的值为非擦除状态时，硬件会置起该位，软件写'1'可以清除。
位 1	保留	0x0	resd	保持为默认值

位 0	OBF	0x0	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。
-----	-----	-----	----	--

### 5.7.5 闪存控制寄存器 (FLASH\_CTRL)

专用于闪存片 1。

域	简称	复位值	类型	功能
位 31: 13	保留	0x00000	resd	保持为默认值
位 12	ODFIE	0x0	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	0x0	resd	保持为默认值
位 10	ERRIE	0x0	rw	错误中断使能 (Error interrupt enable) 开启后 EPPERR 或 PROGERR 都会产生中断。 0: 关闭; 1: 开启
位 9	USDULKS	0x0	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	0x1	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	0x0	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	0x0	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	0x0	rw	用户系统数据编程 (User system data program) 用户系统数据编程。
位 2	BANKERS	0x0	rw	片擦除 (Bank erase) 擦除片操作。
位 1	SECERS	0x0	rw	扇区擦除 (Sector erase) 擦除扇区操作。
位 0	FPRGM	0x0	rw	闪存编程 (Flash program) 编程操作。

### 5.7.6 闪存地址寄存器 (FLASH\_ADDR)

专用于闪存片 1。

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。

### 5.7.7 用户系统数据寄存器 (FLASH\_USD)

域	简称	复位值	类型	功能
位 31: 26	保留	0x00	resd	保持为默认值
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: 6]: 未用 位 5: BTOPT 位 4: nSTDBY_RST 位 3: nDEPSLP_RST 位 2: nWDT_ATO_EN

位 1	FAP	0x0	ro	闪存访问保护（Flash access protection） 该位置起表示闪存存储器不允许访问。
位 0	USDERR	0x0	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

### 5.7.8 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。

### 5.7.9 闪存解锁寄存器2（FLASH\_UNLOCK2）

专用于闪存片 2。

域	简称	复位值	类型	功能
位 31: 0	UKVAL	0XXXXX XXXX	wo	解锁键值（Unlock key value） 该寄存器用于解锁片 2。

注意：所有这些位是只写的，读出时返回 0。

### 5.7.10 闪存状态寄存器2（FLASH\_STS2）

专用于闪存片 2。

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	0x0	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	0x0	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	0x0	resd	保持为默认值
位 2	PRGMERR	0x0	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	0x0	resd	保持为默认值
位 0	OBF	0x0	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。

### 5.7.11 闪存控制寄存器2（FLASH\_CTRL2）

专用于闪存片 2。

域	简称	复位值	类型	功能
位 31: 13	保留	0x00000	resd	保持为默认值
位 12	ODFIE	0x0	rw	操作完成中断使能（Operation done flag interrupt enable） 0：关闭； 1：开启
位 11	保留	0x0	resd	保持为默认值
位 10	ERRIE	0x0	rw	错误中断使能（Error interrupt enable） 开启后 EPPERR 或 PROGERR 都会产生中断。 0：关闭； 1：开启
位 9,8	保留	0x0	resd	保持为默认值
位 7	OPLK	0x1	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	0x0	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 5,4,3	保留	0x0	resd	保持为默认值
位 2	BANKERS	0x0	rw	片擦除 (Bank erase) 擦除片操作。
位 1	SECERS	0x0	rw	扇区擦除 (Sector erase) 擦除扇区操作。
位 0	FPRGM	0x0	rw	闪存编程 (Flash program) 编程操作。

### 5.7.12 闪存地址寄存器2 (FLASH\_ADDR2)

专用于闪存片 2。

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。

### 5.7.13 闪存解锁寄存器3 (FLASH\_UNLOCK3)

专用于外部存储器。

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 SPIM。

注意: 所有这些位是只写的, 读出时返回 0。

### 5.7.14 闪存选择寄存器 (FLASH\_SELECT)

专用于外部存储器。

域	简称	复位值	类型	功能
位 31: 0	SELECT	0x0000 0000	wo	SPIM 支持扩展 SPI Flash 芯片型号选择 0x0001: 参考表 5-4 0x0002: 参考表 5-4 其他: 保留

### 5.7.15 闪存状态寄存器3 (FLASH\_STS3)

专用于外部存储器。

域	简称	复位值	类型	功能
位 31: 6	保留	0x00000000	resd	保持为默认值
位 5	ODF	0x0	rw1	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	0x0	rw1	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	0x0	resd	保持为默认值
位 2	PRGMERR	0x0	rw1	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	0x0	resd	保持为默认值
位 0	OBF	0x0	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。

### 5.7.16 闪存控制寄存器3 (FLASH\_CTRL3)

专用于外部存储器。

域	简称	复位值	类型	功能
位 31: 13	保留	0x00000	resd	保持为默认值
位 12	ODFIE	0x0	rw	操作完成中断使能 (Operation done flag interrupt enable) 0: 关闭; 1: 开启
位 11	保留	0x0	resd	保持为默认值
位 10	ERRIE	0x0	rw	错误中断使能 (Error interrupt enable) 开启后 EPPERR 或 PROGERR 都会产生中断。 0: 关闭; 1: 开启
位 9,8	保留	0x0	resd	保持为默认值
位 7	OPLK	0x1	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	0x0	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 5,4,3	保留	0x0	resd	保持为默认值
位 2	CHPERS	0x0	rw	全擦除 (Chip erase) 外部存储器全擦除操作。
位 1	SECERS	0x0	rw	扇区擦除 (Sector erase) 擦除扇区操作。
位 0	FPRGM	0x0	rw	闪存编程 (Flash program) 编程操作。

### 5.7.17 闪存地址寄存器3 (FLASH\_ADDR3)

专用于外部存储器。

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外部存储器扇区地址。

### 5.7.18 闪存解密地址寄存器 (FLASH\_DA)

专用于外部存储器。

域	简称	复位值	类型	功能
位 31: 0	FDA	0x0000 0000	wo	闪存解密地址 (Flash decryption address) 在用户程序中需要设置 FLASH_DA 寄存器来设置外部存储器加密范围。 0x0840_0000 ~ (0x0840_0000+FDA-0x1)为外部存储器加密范围 (0x0840_0000 +FDA) ~ 0x093F FFFF 为外部存储器未加密范围 注意: FDA 的设定值必须是 4 的倍数, 按字对齐。

### 5.7.19 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: 4	保留	0x00000000	resd	保持为默认值
位 3	SLIB_ENF	0x0	ro	SLIB_ENF: 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: 0	保留	0x0	resd	保持为默认值

### 5.7.20 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	0x000	ro	主存安全库区结束扇区 (sLib end sector) 0: 扇区 0 1: 扇区 1 2: 扇区 2 ... 511: 扇区 511
位 21: 11	SLIB_DAT_SS	0x000	ro	主存安全库区数据区起始扇区 (sLib data start sector) 0: 无效扇区 1: 扇区 1 2: 扇区 2 ... 511: 扇区 511 0x7FF: 无安全库区数据区
位 10: 0	SLIB_SS	0x000	ro	主存安全库区起始扇区 (sLib start sector) 0: 扇区 0 1: 扇区 1 2: 扇区 2 ... 511: 扇区 511

### 5.7.21 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。

### 5.7.22 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: 25	保留	0x00	resd	保持为默认值
位 24: 16	SLIB_RCNT	0x100	ro	安全库区剩余的可使用次数 (sLib remaining count) 从 256 到 0
位 15: 3	保留	0x0000	resd	保持为默认值
位 2	SLIB_ULKF	0x0	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	0x0	ro	密码正确 (sLib password ok) 当密码正确, 该位被硬件置起。
位 0	SLIB_PWD_ERR	0x0	ro	密码错误 (sLib password error) 当密码错误, 并且设定的密码清除寄存器的值不等于 0xFFFF FFFF, 该位被硬件置起。 注意: 当该位置起后, 硬件将不再接受重新设定密码清除寄存器, 直到再次复位。

### 5.7.23 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。



### 5.7.24 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 0: 扇区 0 1: 扇区 1 2: 扇区 2 ... 511: 扇区 511
位 21: 11	SLIB_DSS_SET	0x000	wo	主存安全库区数据区起始扇区设定 (sLib data start sector setting) 用于设定启动安全库区时的数据区起始扇区位置 0: 无效扇区, 设定将导致安全库区无法启动 1: 扇区 1 2: 扇区 2 ... 511: 扇区 511 0x7FF: 无安全库区数据区
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 0: 扇区 0 1: 扇区 1 2: 扇区 2 ... 511: 扇区 511

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。

### 5.7.25 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。

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

专用于闪存主存区域。

域	简称	复位值	类型	功能
位 31	CRC_STRT	0x0	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。
位 30: 24	保留	0x00	wo	保持为默认值
位 23: 12	CRC_SN	0x000	wo	CRC 校验扇区数量 (CRC sector numbler) 设定本次 CRC 校验的数据量, 单位是扇区
位 11: 0	CRC_SS	0x000	wo	CRC 校验起始扇区 (CRC start sector) 设定本次 CRC 校验从哪一扇区开始 0x0: 扇区 0 0x1: 扇区 1 ...

注意: 所有这些位是只写的, 读出无反应。



## 5.7.27 闪存CRC校验结果寄存器（FLASH\_CRC\_CHKR）

专用于主存或是安全库区。

域	简称	复位值	类型	功能
位 31: 0	CRC_CHKR	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。

## 6 通用功能输入输出（GPIO）

### 6.1 简介

AT32F403A/407/407A 支持多达 80 个双向 I/O 管脚，这些管脚分为 5 组，分别为 PA、PB、PC、PD 和 PE，每组最多包含 16 个管脚，每个管脚都可以实现与外部的通讯、控制以及数据采集的功能。

每个管脚都支持通用功能输入输出（GPIO）或复用功能输入输出（IOMUX）。本章节详细介绍 GPIO 功能，IOMUX 功能详见复用功能输入输出章节。

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

每个管脚都可以软件配置输出驱动能力以及输出信号斜率。

每个管脚都可以配置为外部中断输入。

每个管脚都支持配置锁定功能。

### 6.2 功能描述

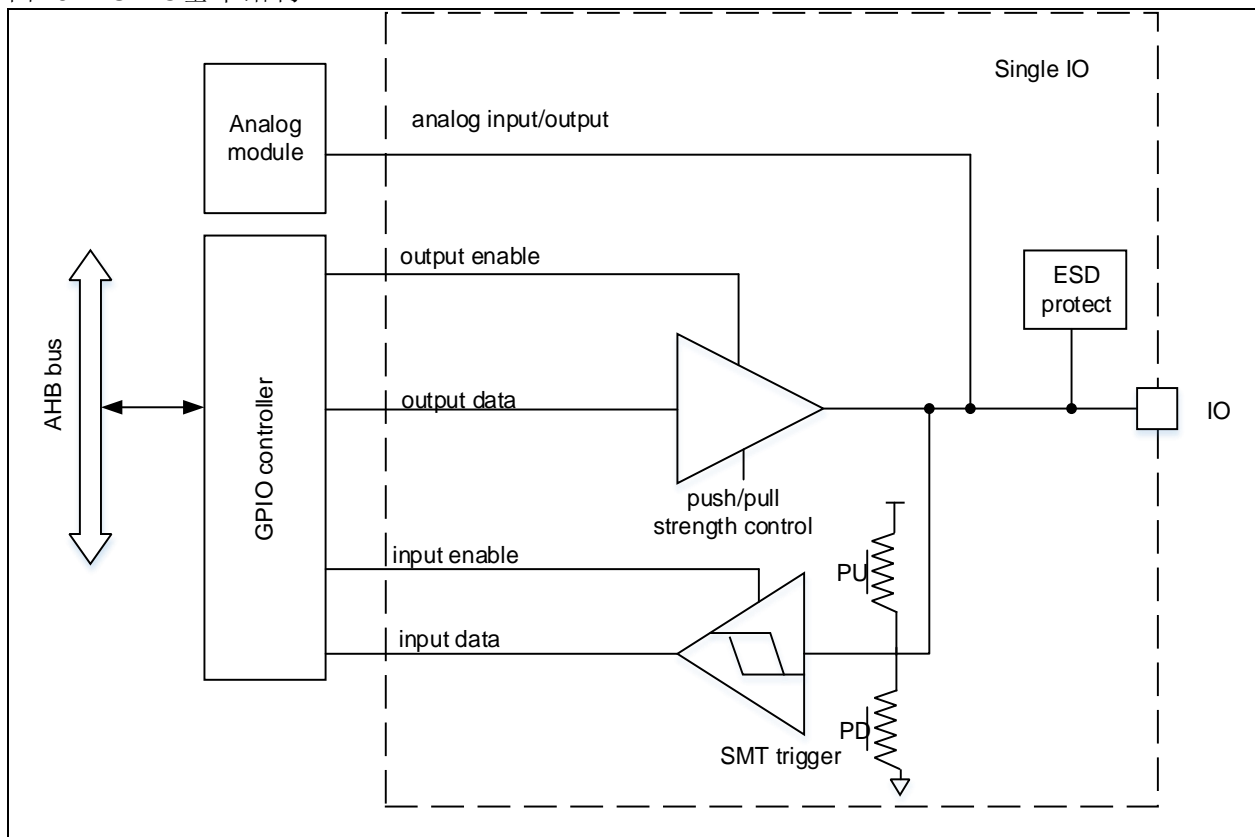
#### 6.2.1 GPIO结构

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

每个 I/O 端口对应的寄存器不允许半字或字节访问，必须按 32 位字被访问，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，所有管脚除了 JATG 相关管脚以外，都被配置为浮空输入模式，JTAG 相关管脚则配置为：PA15/JTDI、PA13/JTMS 和 PB4/JNTRST 为输入上拉模式，PA14/JTCK 为输入下拉模式，PB3/TDO 为浮空输入模式。

### 6.2.3 通用功能输入配置

配置模式	IOFC	HDRV	IOMC[1]	IOMC[0]	ODT 寄存器
浮空输入	01	000	000		不使用
下拉输入	10				0
上拉输入					1

当管脚配置为输入时：

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

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

### 6.2.4 模拟输入/输出配置

配置模式	IOFC	HDRV	IOMC[1]	IOMC[0]	ODT 寄存器
模拟输入输出	00		000		不使用

当 GPIO 端口被配置为模拟输入配置时：

- 施密特触发无效
- 不能对该管脚进行数字输入输出
- 对应的管脚，无任何上拉/下拉电阻。

### 6.2.5 通用功能输出配置

配置模式	IOFC	HDRV	IOMC[1]	IOMC[0]	ODT 寄存器
推挽 (Push-Pull)	00	000/100: 输入模式 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			0 或 1
开漏 (Open-Drain)	01				0 或 1

当 GPIO 端口被配置为输出时：

- 施密特触发器有效
- 可通过输出寄存器让对应管脚输出
- 上拉和下拉电阻不能被使用
- 在开漏模式时，可强输出 0，可用上拉电阻输出 1。
- 在推挽模式时，可通过输出寄存器输出数字 0/1。
- CONF = 10 或 11 时，为复用输出，详情请参考 IOMUX 章节

### 6.2.6 I/O端口保护

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

## 6.3 GPIO寄存器

下面列出了 GPIO 寄存器映射和复位数值。

必须以字（32 位）的方式操作这些外设寄存器。

表 6-1 GPIO寄存器地址映射和复位值

寄存器简称	基址偏移量	复位值
GPIOx_CFGLR	0x00	0x4444 4444
GPIOx_CFGHR	0x04	0x4444 4444
GPIOx_IDT	0x08	0x0000 XXXX
GPIOx_ODT	0x0C	0x0000 0000
GPIOx_SCR	0x10	0x0000 0000

GPIOx_CLR	0x14	0x0000 0000
GPIOx_WPR	0x18	0x0000 0000
GPIOx_HDRV	0x3C	0x0000 0000

## 6.3.1 GPIO配置低寄存器（GPIOx\_CFGLR）（x=A..E）

域	简称	复位值	类型	功能
位 31: 30 位 27: 26 位 23: 22 位 19: 18 位 15: 14 位 11: 10 位 7: 6 位 3: 2	IOFCy	0x1	rw	GPIOx 功能配置（y=0~7）（GPIOx function configure） 当 IO 模式配置为输入模式（IOMCy[1: 0]=00）： 00: 模拟； 01: 浮空（复位后的状态）； 10: 下拉或上拉； 11: 保留。 当 IO 模式配置为输出模式（IOMCy[1: 0]≠00）： 00: 通用推挽； 01: 通用开漏； 10: 复用推挽； 11: 复用开漏。
位 29: 28 位 25: 24 位 21: 20 位 17: 16 位 13: 12 位 9: 8 位 5: 4 位 1: 0	IOMCy	0x0	rw	GPIOx 模式配置（y=0~7）（GPIOx mode configure） 00: 输入模式（复位后的状态）； 01: 输出模式，较大电流推动/吸入能力 10: 输出模式，适中电流推动/吸入能力 11: 输出模式，适中电流推动/吸入能力

注意：有些端口寄存器复位值不同，比如 PA 有些管脚默认是 JTAG/SWD 有上拉输入管脚。

## 6.3.2 GPIO配置高寄存器（GPIOx\_CFGHR）（A..E）

域	简称	复位值	类型	功能
位 31: 30 位 27: 26 位 23: 22 位 19: 18 位 15: 14 位 11: 10 位 7: 6 位 3: 2	IOFCy	0x1	rw	GPIOx 功能配置（y=8~15）（GPIOx function configure） 当 IO 模式配置为输入模式（IOMCy[1: 0]=00）： 00: 模拟； 01: 浮空（复位后的状态）； 10: 下拉或上拉； 11: 保留。 当 IO 模式配置为输出模式（IOMCy[1: 0]≠00）： 00: 通用推挽； 01: 通用开漏； 10: 复用推挽； 11: 复用开漏。
位 29: 28 位 25: 24 位 21: 20 位 17: 16 位 13: 12 位 9: 8 位 5: 4 位 1: 0	IOMCy	0x0	rw	GPIOx 模式配置（y=8~15）（GPIOx mode configure） 00: 输入模式（复位后的状态）； 01: 输出模式，较大电流推动/吸入能力 10: 输出模式，适中电流推动/吸入能力 11: 输出模式，适中电流推动/吸入能力

注意：有些端口寄存器复位值不同，比如 PB 有些管脚默认是 JTAG/SWD 有上拉输入管脚。

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

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

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

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	ODT	0x0000	rw	GPIOx 输出的数据（GPIOx output data）。 每一位对应 GPIOx 的一个 IO。 做输出功能时：GPIOx 对应 IO 口的输出电平状态。 0：低电平； 1：高电平。 做输入功能时：GPIOx 对应 IO 口的上拉/下拉状态。 0：下拉； 1：上拉。

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

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。

## 6.3.6 GPIO清除寄存器（GPIOx\_CLR）（x=A..E）

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	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。

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

域	简称	复位值	类型	功能
位 31: 17	保留	0x0000	resd	保持默认值。
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 WPSEL 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。

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

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	HDRV	0x0000	rw	<p>端口 x 极大电流推动/吸入能力切换控制 (y=0...15) (Portx Huge Driven)</p> <p>每一位对应 GPIOx 的一个 IO。</p> <p>0: GPIO 配置为较大或适中电流推动/吸入能力, 由 IOMCy[1: 0]配置决定;</p> <p>1: GPIO 配置为极大电流推动/吸入能力, 忽略 IOMCy[1: 0]配置。</p>

## 7 复用功能输入输出（IOMUX）

### 7.1 简介

AT32F403A/407/407A 支持多达 80 个双向 I/O 管脚，这些管脚分为 5 组，分别为 PA、PB、PC、PD 和 PE，每组最多包含 16 个管脚，每个管脚都可以实现与外部的通讯、控制以及数据采集的功能。

每个管脚都支持通用功能输入输出（GPIO）或复用功能输入输出（IOMUX）。本章节详细介绍 IOMUX 功能，GPIO 功能详见通用功能输入输出章节。

每个管脚都通过软件配置 GPIO 配置低寄存器（GPIOx\_CFGLR）或 GPIO 配置高寄存器（GPIOx\_CFGHR）设定成复用功能输入输出端口。

大多数管脚支持多个外设的输出功能映射，可通过 IOMUX 章节寄存器来选择不同的外设输入输出功能。每个管脚都支持外部中断功能。

### 7.2 功能描述

#### 7.2.1 IOMUX结构

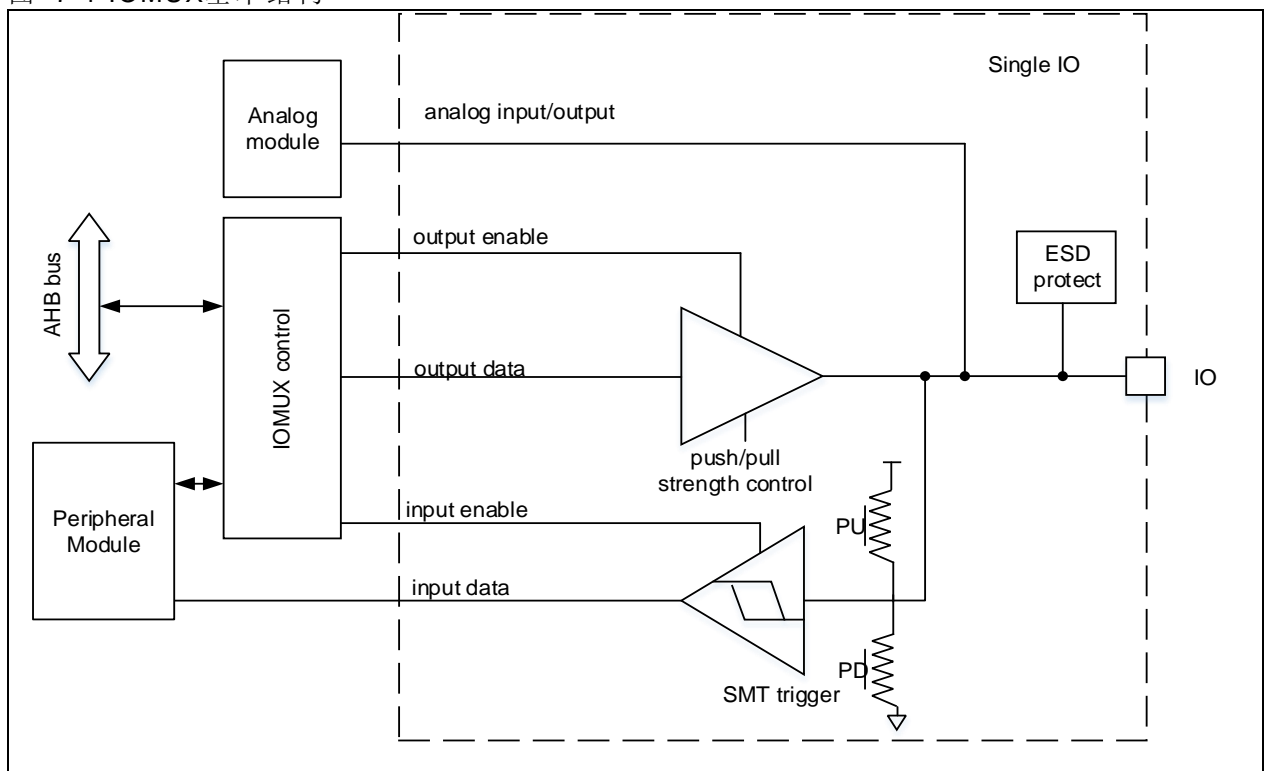
管脚作为复用输入功能时，与通用输入功能一样，端口配置成输入模式（浮空、上拉、下拉）。

要实现复用输出功能，必须配置 GPIO 配置低寄存器（GPIOx\_CFGLR）或 GPIO 配置高寄存器（GPIOx\_CFGHR）将该端口设定为复用功能输出模式（推挽或开漏）。此时管脚和 GPIO 控制器断开，由 IOMUX 控制器进行控制。

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

管脚作为复用输出功能时，一个管脚可能持多个外设的输出功能映射，需要通过配置 IOMUX 相关寄存器来选择复用输出功能。当管脚配置为复用输出功能但对应外设没有被激活的话，该管脚的输出将不确定。

图 7-1 IOMUX基本结构



#### 7.2.2 复用功能输入配置

当 I/O 端口配置为复用功能输入时：

- 管脚状态可通过对输入数据寄存器的读访问得到
- 可配置管脚为浮空输入、上拉输入或下拉输入



- 施密特触发器有效
- 不能对该管脚进行输出。

表 7-1 复用功能输入配置

配置模式	IOFC	HDRV	IOMC[1]	IOMC[0]	ODT 寄存器
浮空输入	01		000		不使用
下拉输入	10				0
上拉输入					1

## 7.2.3 复用功能输出或双向复用功能配置

当 I/O 端口配置为复用功能输出或双向复用功能时：

- 管脚输出由外设决定
- 施密特触发器有效
- 上拉和下拉电阻均关闭
- 如果管脚被误配成多个复用功能输出，管脚将按映射优先级输出，详见下面小节。
- 开漏模式时，读输入数据寄存器时可得到 I/O 口状态
- 推挽模式时，读输入数据寄存器时可得到 I/O 口状态

一些外设的输出功能可以重映射到不同的管脚，因此可以在不同封装中选择 I/O 外设复用功能的数量达到最优化。通过配置 IO 复用重映射寄存器(IOMUX\_REMAP)或复用重映射寄存器 x(IOMUX\_REMAPx) (x=2,3...8)来实现管脚的重新映射。

表 7-2 复用功能输出配置

配置模式	IOFC	HDRV	IOMC[1]	IOMC[0]
推挽（Push-Pull）	10	001: 输出模式，较大电流推动/吸入能力		
开漏（Open-Drain）	11	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		

注意：配置为复用功能输出或双向复用功能时，必须满足 IOMC[1: 0] > 00

## 7.2.4 外设复用功能管脚配置

当外设需要使用 IOMUX 复用功能时：

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

## 7.2.5 IOMUX映射优先级

单个管脚可能有多个外设复用映射，当多个外设复用映射到同一个管脚时，外设遵循以下优先级规则：

- 硬件抢占功能优先
- JTAG 调试端口优先
- 非 timer 外设复用映射优先于 timer 外设。
- 多个非 timer 外设之间复用映射无优先级关系，复用功能叠加到同一个管脚。

### 7.2.5.1 硬件抢占功能

某些管脚不管 GPIO 配置为任何模式，都会被特定的硬件功能占用。

表 7-3 硬件抢占功能

管脚名字	抢占使能位	说明
PA0	PWC_CTRLSTS[8] = 1	抢占使能位有效之后，PA0 管脚直接作为 PWC 的 WKUP 功能使用
PA4	DAC_CTRL[2] = 1	抢占使能位有效之后，PA4 作为 DAC1 模拟通道使用
PA5	DAC_CTRL[18] = 1	抢占使能位有效之后，PA5 作为 DAC2 模拟通道使用
PA11	CRM_APB1EN[23]=1	抢占使能位有效之后，PA11 作为 USB_DM 通道使用
PA12	CRM_APB1EN[23]=1	抢占使能位有效之后，PA12 作为 USB_DP 通道使用
PC13	CRM_APB1EN[27]=1 & (BPR_CTRL[0]=1   BPR_RTCCAL[ 8 ]= 1   BPR_RTCCAL[ 7 ]= 1 )	抢占使能位有效之后，PC13 作为 RTC 通道使用
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

抢占使能位有效之后, PC15 作为 LEXT 通道使用

## 7.2.5.2 调试端口优先

在进行芯片调试时, 为了防止其他外设对调试端口的干扰, 导致不能被调试, 配置好后的调试端口管脚, 不管对应管脚的 GPIO 寄存器被配置为任何模式, 都会一直保持为调试端口。

为了在调试期间可以使用更多管脚, 通过设置复用重映射和调试 I/O 配置寄存器 (IOMUX\_REMAP) 的 SWJTAG\_MUX [2: 0]位或复用重映射和调试 I/O 配置寄存器 7 (IOMUX\_REMAP7) 的 SWJTAG\_GMUX [2: 0]位, 可以改变上述重映射配置。

表 7-4 调试端口映射

SWJTAG_MUX [2: 0]或 SWJTAG_GMUX [2: 0]	SWJ/I/O 管脚分配				
	PA13/JTMS/ SWDIO	PA14/JTCK/ SWCLK	PA15/JTDI	PB3/JTDO/ TRACESWO	PB4/NJTRST
000	√	√	√	√	√
001	√	√	√	√	x
010	√	√	x	x	x
100	x	x	x	x	x
其它	-	-	-	-	-

注意: √ 表示该管脚被强制分配给调试端口, x 表示该管脚可以释放给其他外设使用。

## 7.2.5.3 其他外设输出优先级关系

除了硬件抢占功能和端口调试功能以外, 其他外设输出优先级如下:

- 非 timer 外设输出优先于 timer 外设, 即其他外设和 timer 同时映射到某一管脚, timer 不能输出。
- 多个非 timer 外设输出如果映射到同一管脚, 则这些外设输出会叠加输出到该管脚。

## 7.2.6 外部中断/唤醒线

每个管脚都支持作为外部中断的输入, 对应的管脚须配置为输入模式。

## 7.3 IOMUX寄存器

下面列出了 IOMUX 寄存器映射和复位数值。  
必须以字(32 位) 的方式操作这些外设寄存器。

表 7-5 IOMUX寄存器地址映射和复位值

寄存器简称	基址偏移量	复位值
IOMUX_EVTOUT	0x00	0x0000 0000
IOMUX_REMAP	0x04	0x0000 0000
IOMUX_EXINTC1	0x08	0x0000
IOMUX_EXINTC2	0x0C	0x0000
IOMUX_EXINTC3	0x10	0x0000
IOMUX_EXINTC4	0x14	0x0000
IOMUX_REMAP2	0x1C	0x0000 0000
IOMUX_REMAP3	0x20	0x0000 0000
IOMUX_REMAP4	0x24	0x0000 0000
IOMUX_REMAP5	0x28	0x0000 0000
IOMUX_REMAP6	0x2C	0x0000 0000
IOMUX_REMAP7	0x30	0x0000 0000
IOMUX_REMAP8	0x34	0x0000 0000

注意：对寄存器 IOMUX\_EVTOUT, IOMUX\_REMAPx 和 IOMUX\_EXINTCx 进行读写操作前，应当首先打开 IOMUX 的时钟。

### 7.3.1 事件输出控制寄存器 (IOMUX\_EVTOUT)

域	简称	复位值	类型	功能
位 31: 8	保留	0x000000	resd	保持默认值。
位 7	EVOEN	0x0	rw	事件输出使能 (Event output enable) 使能后, Cortex-M 的 EVENTOUT 信号将连接到配置的 I/O 口。
位 6: 4	SELPOR	0x0	rw	选择 IO 端口 (Selection IO port) 选择输出 EVENTOUT 信号的 GPIO 端口: 000: GPIOA; 001: GPIOB; 010: GPIOC; 011: GPIOD; 100: GPIOE。
位 3: 0	SELPIN	0x0	rw	选择 IO 管脚 (x=A...E) (Selection IO pin) 选择输出 EVENTOUT 信号的 GPIOx 的 I/O 管脚: 0000: 管脚 0    0001: 管脚 1 0010: 管脚 2    0011: 管脚 3 0100: 管脚 4    0101: 管脚 5 0110: 管脚 6    0111: 管脚 7 1000: 管脚 8    1001: 管脚 9 1010: 管脚 10    1011: 管脚 11 1100: 管脚 12    1101: 管脚 13 1110: 管脚 14    1111: 管脚 15

### 7.3.2 IO复用重映射寄存器 (IOMUX\_REMAP)

域	简称	复位值	类型	功能
位 31	SPI1_MUX	0x0	rw	SPI1 的 IO 复用 (SPI1 IO muxing) 具体定义参考位 0 的 SPI1_MUX[1: 0]。
位 30	PTP_PPS_MUX	0x0	rw	以太网 PTP PPS 的 IO 复用 (EMAC PTP PPS IO muxing) 选择 PPS_PTS 是否连接到 PB5。 0: PTP_PPS 不连接到 PB5 管脚; 1: PTP_PPS 连接到 PB5 管脚。 注: 该位只在 AT32F407 上有效。
位 29	TMR2ITR1_MUX	0x0	rw	TMR2 内部触发 1 复用 (TMR2 internal trigger 1 muxing) 选择 TMR2 ITR1 的内部复用。 0: 使用 TMR8_TRGO 作为 TMR2ITR1 的输入; 1: 使用以太网的 PTP 输出作为 TMR2 ITR1 的输入。 注: 该位只在 AT32F407 上有效。
位 28	SPI3_MUX	0x0	rw	SPI3 的 IO 复用 (SPI3 IO muxing) 选择 SPI3 的 CS、SCK、MISO、MOSI 的 IO 复用功能。 0: CS/PA15、SCK/PB3、MISO/PB4、MOSI/PB5; 1: CS/PA4、SCK/PC10、MISO/PC11、MOSI/PC12。 注: 该位只在 AT32F407 上有效。
位 27	保留	0x0	resd	保持默认值。
位 26: 24	SWJTAG_MUX	0x0	rw	SWD JTAG 复用 (SWD JTAG muxing) 配置 SWJTAG 接口相关的 IO 是否作为 GPIO 使用。 000: 支持 SWD 和 JTAG, 所有 SWJTAG 管脚不可作 GPIO; 001: 支持 SWD 和 JTAG, 禁用 NJTRST, PB4 可作 GPIO; 010: 支持 SWD, 禁用 JTAG, PA15/PB3/PB4 可作 GPIO;

				100: 禁用 SWD 和 JTAG, 所有 SWJTAG 管脚均可作 GPIO; 其它: 无作用。
位 23	MII_RMII_SEL_MUX	0x0	rw	MII 或 RMII 接口选择 (MII or RMII selection muxing) 选择以太网使用 MII 接口还是 RMII 接口。 0: MII; 1: RMII。 注: 该位只在 AT32F407 上有效。
位 22	CAN2_MUX	0x0	rw	CAN2 的 IO 复用 (CAN2 IO muxing) 选择 CAN2_TX 和 CAN2_RX 的 IO 复用功能。 0: RX/PB12、TX/PB13 1: RX/PB5、TX/PB6 注: 该位只在 AT32F407 上有效。
位 21	EMAC_MUX	0x0	rw	以太网 MAC 的 IO 复用 (EMAC IO muxing) 选择以太网 MAC 的 IO 复用功能。 0: RX_DV/CRS_DV/PA7、RXD0/PC4、RXD1/PC5、RXD2/PB0、RXD3/PB1 1: RX_DV/CRS_DV/PD8、RXD0/PD9、RXD1/PD10、RXD2/PD11、RXD3/PD12。 注: 该位只在 AT32F407 上有效。
位 20	ADC2_ETO_MUX	0x0	rw	ADC2 普通转换外部触发复用 (ADC2 external trigger ordinary conversion muxing) 选择 ADC2 普通转换的外部触发输入。 0: ADC2 普通转换外部触发连接到 EXINT11; 1: ADC2 普通转换外部触发连接到 TMR8_TRGO。
位 19	ADC2_ETP_MUX	0x0	rw	ADC2 抢占转换外部触发复用 (ADC2 external trigger preempted conversion muxing) 选择 ADC2 抢占转换外部触发输入。 0: ADC2 抢占转换外部触发连接到 EXINT15; 1: ADC2 抢占转换外部触发连接到 TMR8 通道 4。
位 18	ADC1_ETO_MUX	0x0	rw	ADC1 普通转换外部触发复用 (ADC1 external trigger ordinary conversion muxing) 选择 ADC1 普通转换外部触发输入。 0: ADC1 普通转换外部触发连接到 EXINT11; 1: ADC1 普通转换外部触发连接到 TMR8_TRGO。
位 17	ADC1_ETP_MUX	0x0	rw	ADC1 抢占转换外部触发复用 (ADC1 External trigger preempted conversion muxing) 选择 ADC1 抢占转换外部触发输入。 0: ADC1 抢占转换外部触发连接到 EXINT15; 1: ADC1 抢占转换外部触发连接到 TMR8 通道 4。
位 16	TMR5CH4_MUX	0x0	rw	TMR5 通道 4 复用 (TMR5 channel4 muxing) 选择 TMR5 通道 4 的内部映射。 0: TMR5_CH4 连接到 PA3; 1: TMR5_CH4 连接到 LICK 低速内部时钟, 可对 LICK 进行校准。
位 15	PD01_MUX	0x0	rw	PD0/PD1 映射到 HEXT_IN/HEXT_OUT (PD0/PD1 mapping on HEXT_IN / HEXT_OUT) 选择 PD0 和 PD1 的 GPIO 功能映射。 此功能只适用于 48 和 64 管脚的封装。 0: 无映射; 1: PD0 映射到 HEXT_IN, PD1 映射到 HEXT_OUT。
位 14: 13	CAN_MUX	0x0	rw	CAN 的 IO 复用 (CAN IO muxing) 选择 CAN_TX 和 CAN_RX 的 IO 复用功能。 00: RX/PA11、TX/PA12; 01: 不使用; 10: RX/ PB8、TX/ PB9; 11: RX/ PD0、TX/ PD1;
位 12	TMR4_MUX	0x0	rw	定时器 4 的 IO 复用 (TMR4 IO muxing) 选择 TMR4 的 IO 复用功能。 0: CH1/PB6, CH2/PB7, CH3/PB8, CH4/PB9;

				1: CH1/PD12, CH2/PD13, CH3/PD14 CH4/PD15。
位 11: 10	TMR3_MUX	0x0	rw	定时器 3 的 IO 复用 (TMR3 IO muxing) 选择 TMR3 的 IO 复用功能。 00: CH1/PA6, CH2/PA7, CH3/PB0, CH4/PB1; 01: 不使用; 10: CH1/PB4, CH2/PB5, CH3/PB0, CH4/PB1; 11: CH1/PC6, CH2/PC7, CH3/PC8, CH4/PC9。 注: IO 复用不影响在 PD2 上的 TMR3_EXT。
位 9: 8	TMR2_MUX	0x0	rw	定时器 2 的 IO 复用 (TMR2 IO muxing) 选择 TMR2 的 IO 复用功能。 00: CH1/EXT/PA0, CH2/PA1, CH3/PA2, CH4/PA3; 01: CH1/EXT/PA15, CH2/PB3, CH3/PA2, CH4/PA3; 10: CH1/EXT/PA0, CH2/PA1, CH3/PB10, CH4/PB11; 11: CH1/EXT/PA15, CH2/PB3, CH3/PB10, CH4/PB11。
位 7: 6	TMR1_MUX	0x0	rw	定时器 1 的 IO 复用 (TMR1 IO muxing) 选择 TMR1 的 IO 复用功能。 00: EXT/PA12, CH1/PA8, CH2/PA9, CH3/PA10, CH4/PA11, BRK/PB12, CH1C/PB13, CH2C/PB14, CH3C/PB15; 01: EXT/PA12, CH1/PA8, CH2/PA9, CH3/PA10, CH4/PA11, BRK/PA6, CH1C/PA7, CH2C/PB0, CH3C/PB1; 10: 不使用; 11: EXT/PE7, CH1/PE9, CH2/PE11, CH3/PE13, CH4/PE14, BRK/PE15, CH1C/PE8, CH2C/PE10, CH3C/PE12。
位 5: 4	USART3_MUX	0x0	rw	USART3 的 IO 复用 (USART3 IO muxing) 选择 USART3 的 IO 复用功能。 00: TX/PB10, RX/PB11, CK/PB12, CTS/PB13, RTS/PB14; 01: TX/PC10, RX/PC11, CK/PC12, CTS/PB13, RTS/PB14; 10: 不使用; 11: TX/PD8, RX/PD9, CK/PD10, CTS/PD11, RTS/PD12
位 3	USART2_MUX	0x0	rw	USART2 的 IO 复用 (USART2 IO muxing) 选择 USART2 的 IO 复用功能。 0: CTS/PA0, RTS/PA1, TX/PA2, RX/PA3, CK/PA4; 1: CTS/PD3, RTS/PD4, TX/PD5, RX/PD6, CK/PD7;
位 2	USART1_MUX	0x0	rw	USART1 的 IO 复用 (USART1 IO muxing) 选择 USART1 的 IO 复用功能。 0: TX/PA9, RX/PA10; 1: TX/PB6, RX/PB7。
位 1	I2C1_MUX	0x0	rw	I2C1 的 IO 复用 (I2C1 IO muxing) 选择 I2C1 的 IO 复用功能。 0: SCL/PB6, SDA/PB7 SMBA/PB5; 1: SCL/PB8, SDA/PB9 SMBA/PB5。
位 0	SPI1_MUX	0x0	rw	SPI1 的 IO 复用 (SPI1 IO muxing) 选择 SPI1 的 IO 复用功能。SPI1_MUX[1]设置于位 31。 00: CS/PA4, SCK/PA5, MISO/PA6, MOSI/PA7 MCK/PB0。 01: CS/PA15, SCK/PB3, MISO/PB4, MOSI/PB5 MCK/PB0。

10: CS/PA4, SCK/PA5, MISO/PA6, MOSI/PA7  
MCK/PB6 .  
11: CS/PA15, SCK/PB3, MISO/PB4, MOSI/PB5  
MCK/PB6.

## 7.3.3 复用外部中断配置寄存器1 (IOMUX\_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	0x0000	rw	配置 EXINT3 的输入源 (configure EXINT3 source) 选择 EXINT3 外部中断的输入源。 0000: GPIOA 管脚 3 0001: GPIOB 管脚 3 0010: GPIOC 管脚 3 0011: GPIOD 管脚 3 0100: GPIOE 管脚 3 其他: 保留
位 11: 8	EXINT2	0x0000	rw	配置 EXINT2 的输入源 (configure EXINT2 source) 选择 EXINT2 外部中断的输入源。 0000: GPIOA 管脚 2 0001: GPIOB 管脚 2 0010: GPIOC 管脚 2 0011: GPIOD 管脚 2 0100: GPIOE 管脚 2 其他: 保留
位 7: 4	EXINT1	0x0000	rw	配置 EXINT1 的输入源 (configure EXINT1 source) 选择 EXINT1 外部中断的输入源。 0000: GPIOA 管脚 1 0001: GPIOB 管脚 1 0010: GPIOC 管脚 1 0011: GPIOD 管脚 1 0100: GPIOE 管脚 1 其他: 保留
位 3: 0	EXINT0	0x0000	rw	配置 EXINT0 的输入源 (configure EXINT0 source) 选择 EXINT0 外部中断的输入源。 0000: GPIOA 管脚 0 0001: GPIOB 管脚 0 0010: GPIOC 管脚 0 0011: GPIOD 管脚 0 0100: GPIOE 管脚 0 其他: 保留

## 7.3.4 复用外部中断配置寄存器2 (IOMUX\_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	0x0000	rw	配置 EXINT7 的输入源 (configure EXINT7 source) 选择 EXINT7 外部中断的输入源。 0000: GPIOA 管脚 7 0001: GPIOB 管脚 7 0010: GPIOC 管脚 7 0011: GPIOD 管脚 7 0100: GPIOE 管脚 7 其他: 保留
位 11: 8	EXINT6	0x0000	rw	配置 EXINT6 的输入源 (configure EXINT6 source) 选择 EXINT6 外部中断的输入源。 0000: GPIOA 管脚 6 0001: GPIOB 管脚 6 0010: GPIOC 管脚 6 0011: GPIOD 管脚 6 0100: GPIOE 管脚 6 其他: 保留



位 7: 4	EXINT5	0x0000	rw	配置 EXINT5 的输入源 (configure EXINT5 source) 选择 EXINT5 外部中断的输入源。 0000: GPIOA 管脚 5 0001: GPIOB 管脚 5 0010: GPIOC 管脚 5 0011: GPIOD 管脚 5 0100: GPIOE 管脚 5 其他: 保留
位 3: 0	EXINT4	0x0000	rw	配置 EXINT4 的输入源 (configure EXINT4 source) 选择 EXINT4 外部中断的输入源。 0000: GPIOA 管脚 4 0001: GPIOB 管脚 4 0010: GPIOC 管脚 4 0011: GPIOD 管脚 4 0100: GPIOE 管脚 4 其他: 保留

## 7.3.5 复用外部中断配置寄存器3 (IOMUX\_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	0x0000	rw	配置 EXINT11 的输入源 (configure EXINT11 source) 选择 EXINT11 外部中断的输入源。 0000: GPIOA 管脚 11 0001: GPIOB 管脚 11 0010: GPIOC 管脚 11 0011: GPIOD 管脚 11 0100: GPIOE 管脚 11 其他: 保留
位 11: 8	EXINT10	0x0000	rw	配置 EXINT10 的输入源 (configure EXINT10 source) 选择 EXINT10 外部中断的输入源。 0000: GPIOA 管脚 10 0001: GPIOB 管脚 10 0010: GPIOC 管脚 10 0011: GPIOD 管脚 10 0100: GPIOE 管脚 10 其他: 保留
位 7: 4	EXINT9	0x0000	rw	配置 EXINT9 的输入源 (configure EXINT9 source) 选择 EXINT9 外部中断的输入源。 0000: GPIOA 管脚 9 0001: GPIOB 管脚 9 0010: GPIOC 管脚 9 0011: GPIOD 管脚 9 0100: GPIOE 管脚 9 其他: 保留
位 3: 0	EXINT8	0x0000	rw	配置 EXINT8 的输入源 (configure EXINT8 source) 选择 EXINT8 外部中断的输入源。 0000: GPIOA 管脚 8 0001: GPIOB 管脚 8 0010: GPIOC 管脚 8 0011: GPIOD 管脚 8 0100: GPIOE 管脚 8 其他: 保留



### 7.3.6 复用外部中断配置寄存器4 (IOMUX\_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	0x0000	rw	配置 EXINT15 的输入源 (configure EXINT15 source) 选择 EXINT15 外部中断的输入源。 0000: GPIOA 管脚 15 0001: GPIOB 管脚 15 0010: GPIOC 管脚 15 0011: GPIOD 管脚 15 0100: GPIOE 管脚 15 其他: 保留
位 11: 8	EXINT14	0x0000	rw	配置 EXINT14 的输入源 (configure EXINT14 source) 选择 EXINT14 外部中断的输入源。 0000: GPIOA 管脚 14 0001: GPIOB 管脚 14 0010: GPIOC 管脚 14 0011: GPIOD 管脚 14 0100: GPIOE 管脚 14 其他: 保留
位 7: 4	EXINT13	0x0000	rw	配置 EXINT13 的输入源 (configure EXINT13 source) 选择 EXINT13 外部中断的输入源。 0000: GPIOA 管脚 13 0001: GPIOB 管脚 13 0010: GPIOC 管脚 13 0011: GPIOD 管脚 13 0100: GPIOE 管脚 13 其他: 保留
位 3: 0	EXINT12	0x0000	rw	配置 EXINT12 的输入源 (configure EXINT12 source) 选择 EXINT12 外部中断的输入源。 0000: GPIOA 管脚 12 0001: GPIOB 管脚 12 0010: GPIOC 管脚 12 0011: GPIOD 管脚 12 0100: GPIOE 管脚 12 其他: 保留

### 7.3.7 IO复用重映射寄存器2 (IOMUX\_REMAP2)

域	简称	复位值	类型	功能
位 31: 22	保留	0x000	resd	保持默认值。
位 21	EXT_SPIM_EN_MUX	0x0	rw	使能 SPIM 接口 (SPIM enable muxing)。 选择是否使用外部 SPI Flash。
位 20: 19	SDIO2_MUX	0x0	rw	SDIO2_MUX[1: 0]: SDIO2 的 IO 复用 (SDIO2 IO muxing) 00: D0/PC0 D1/PC1 D2/PC3 D4/PA4 D5/PA5 D6/PA6 D7/PA7 CK/PC4 CMD/PC5 01: D0/PA4 D1/PA5 D2/PA6 D3/PA7 CK/PA2 CMD/PA3 10,11: 不使用
位 18	I2C3_MUX	0x0	rw	I2C3_MUX: I2C3 的 IO 复用 (I2C3 IO muxing) 选择 I2C3 的 IO 复用功能。 0: SCL/PA8 SDA/PC9 SMBA/PA9 1: SCL/PA8 SDA/PB4 SMBA/PA9
位 17	SPI4_MUX	0x0	rw	SPI4_MUX: SPI4 的 IO 复用 (SPI4 IO muxing) 选择 SPI4 的 IO 复用功能。 0: CS/PE4 SCK/PE2 MISO/PE5 MOSI/PE6 MCK/PC8 1: CS/PE12 SCK/PE11 MISO/PE13 MOSI/PE14 MCK/PC8
位 16: 11	保留	0x00	resd	保持默认值。
位 10	XMC_NADV_MUX	0x0	rw	XMC_NADV_MUX: XMC NADV 连接。

				选择是否使用 XMC_NADV 信号。 0: XMC_NADV 连接到 pin。(默认) 1: XMC_NADV 不使用, 对应的 pin 可被其他外设使用。
位 9: 6	保留	0x0	resd	保持默认值。
位 5	TMR9_MUX	0x0	rw	TMR9_MUX: TMR9 的 IO 复用 (TMR9 IO muxing) 选择 TMR9 的 IO 复用功能。 0: CH1/PA2, CH2/PA3; 1: CH1/PE5, CH2/PE6;
位 4: 0	保留	0x00	resd	保持默认值。

### 7.3.8 IO复用重映射寄存器3 (IOMUX\_REMAP3)

域	简称	复位值	类型	功能
位 31: 4	保留	0x0000000	resd	保持默认值。
位 3: 0	TMR9_GMUX	0x0	rw	TMR9 的 IO 全局复用 (TMR9 general muxing) 选择 TMR9 的 IO 复用功能。 0000: CH1/PA2 CH2/PA3 0001: CH1/PE5 CH2/PE6

### 7.3.9 IO复用重映射寄存器4 (IOMUX\_REMAP4)

域	简称	复位值	类型	功能
位 31: 20	保留	0x000	resd	保持默认值。
位 19	TMR5CH4_GMUX	0x0	rw	TMR5 通道 4 全局复用 (TMR5 channel4 general muxing) 选择 TMR5 通道 4 内部复用。 0: TMR5_CH4 与 PA3 相连; 1: LICK 内部振荡器与 TMR5_CH4 相连, 目的是对 LICK 进行校准。
位 18: 16	保留	0x0	resd	保持默认值。
位 15: 12	TMR4_GMUX	0x0	rw	TMR4 的 IO 全局复用 (TMR4 IO general muxing) 选择 TMR4 的 IO 复用功能。 0000: CH1/PB6 CH2/PB7 CH3/PB8 CH4/PB9 0001: CH1/PD12 CH2/PD13 CH3/PD14 CH4/PD15
位 11: 8	TMR3_GMUX	0x0	rw	TMR3 的 IO 全局复用 (TMR3 IO general muxing) 选择 TMR3 的 IO 复用功能。 0000: CH1/PA6 CH2/PA7 CH3/PB0 CH4/PB1 0010: CH1/PB4 CH2/PB5 CH3/PB0 CH4/PB1 0011: CH1/PC6 CH2/PC7 CH3/PC8 CH4/PC9
位 7: 6	TMR2ITR1_GMUX	0x0	rw	TMR2 内部触发 1 复用 (TMR2 internal trigger 1 general muxing) 选择 TMR2_ITR1 的内部复用。 00: 使用 TMR8_TRGO 作为 TMR2 ITR1 的输入 01: 保留, 请勿使用 10: 在内部连接以太网的 PTP 输出至 TMR2_ITR1; 11: 使用 USB 的 SOF 作为 TMR2_ITR1 的输入 (当 TMR2_ITR1 使用此选项时, 会导致 TMR2_GMUX/TMR2_MUX 功能失效, 请注意此使用限制)
位 5: 4	TMR2_GMUX	0x0	rw	TMR2 的 IO 全局复用 (TMR2 IO general muxing) 选择 TMR2 的 IO 复用功能。 00: CH1_EXT/PA0 CH2/PA1 CH3/PA2 CH4/PA3 01: CH1_EXT/PA15 CH2/PB3 CH3/PA2 CH4/PA3 10: CH1_EXT/PA0 CH2/PA1 CH3/PB10 CH4/PB11 11: CH1_EXT/PA15 CH2/PB3 CH3/PB10 CH4/PB11
位 3: 0	TMR1_GMUX	0x0	rw	TMR1 的 IO 全局复用 (TMR1 IO general muxing) 选择 TMR1 的 IO 复用功能。 0000: EXT/PA12, CH1/PA8, CH2/PA9, CH3/PA10, CH4/PA11, BRK/PB12, CH1C/PB13, CH2C/PB14, CH3C/PB15;

0001: EXT/PA12, CH1/PA8, CH2/PA9, CH3/PA10, CH4/PA11, BRK/PA6, CH1C/PA7, CH2C/PB0, CH3C/PB1;  
0011: EXT/PE7, CH1/PE9, CH2/PE11, CH3/PE13, CH4/PE14, BRK/PE15, CH1C/PE8, CH2C/PE10, CH3C/PE12。  
其他: 不使用

## 7.3.10 IO复用重映射寄存器5 (IOMUX\_REMAP5)

域	简称	复位值	类型	功能
位 31: 28	SPI4_GMUX	0x0	rw	<p>SPI4 的 IO 全局复用 (SPI4 IO general muxing) 选择 SPI4 的 IO 复用功能。 0000: CS/PE4 SCK/PE2 MISO/PE5 MOSI/PE6 MCK/PC8 0001: CS/PE12 SCK/PE11 MISO/PE13 MOSI/PE14 MCK/PC8 0010: CS/PB6 SCK/PB7 MISO/PB8 MOSI/PB9 MCK/PC8 0011: CS/PB6 SCK/PB7 MISO/PB8 MOSI/PB9 MCK/PA10 其他: 不使用</p>
位 27: 24	SPI3_GMUX	0x0	rw	<p>SPI3 的 IO 全局复用 (SPI3 IO general muxing) 选择 SPI3 的 IO 复用功能。 0000: CS/PA15、SCK/PB3、MISO/PB4、MOSI/PB5 MCK/PC7; 0001: CS/PA4、SCK/PC10、MISO/PC11、MOSI/PC12 MCK/PC7。 0010: CS/PA15、SCK/PB3、MISO/PB4、MOSI/PB5 MCK/PB10 0011: CS/PA4、SCK/PC10、MISO/PC11、MOSI/PC12 MCK/PB10 其他: 不使用</p>
位 23: 20	SPI2_GMUX	0x0	rw	<p>SPI2 的 IO 全局复用 (SPI2 IO general muxing) 选择 SPI2 的 IO 复用功能。 0000: MCK/PC6 0001: MCK/PA3 0010: MCK/PA6 其他: 不使用</p>
位 19: 16	SPI1_GMUX	0x0	rw	<p>SPI1 的 IO 全局复用 (SPI1 IO general muxing) 选择 SPI1 的 IO 复用功能。 0000: CS/PA4, SCK/PA5, MISO/PA6, MOSI/PA7 MCK/PB0 . 0001: CS/PA15, SCK/PB3, MISO/PB4, MOSI/PB5 MCK/PB0。 0010: CS/PA4, SCK/PA5, MISO/PA6, MOSI/PA7 MCK/PB6 . 0011: CS/PA15, SCK/PB3, MISO/PB4, MOSI/PB5 MCK/PB6。 其他: 不使用</p>
位 15: 12	I2C3_GMUX	0x0	rw	<p>I2C3 的 IO 全局复用 (I2C3 IO general muxing) 选择 I2C3 的 IO 复用功能。 0000: SCL/PA8 SDA/PC9 SMBA/PA9 0001: SCL/PA8 SDA/PB4 SMBA/PA9 其他: 不使用</p>
位 11: 8	保留	0x0	resd	保持默认值。
位 7: 4	I2C1_GMUX	0x0	rw	<p>I2C1 的 IO 全局复用 (I2C1 IO general muxing) 选择 I2C1 的 IO 复用功能。 0000: SCL/PB6, SDA/PB7 SMBA/PB5; 0001: SCL/PB8, SDA/PB9 SMBA/PB5。 其他: 不使用</p>
位 3: 0	USART5_GMUX	0x0	rw	USART5 的 IO 全局复用 (USART5 IO general muxing)

选择 USART5 的 IO 复用功能。  
0000: TX/PC12 RX/PD12  
0001: TX/PB9 RX/PB8  
其他: 不使用

## 7.3.11 IO复用重映射寄存器6 (IOMUX\_REMAP6)

域	简称	复位值	类型	功能
位 31: 28	UART4_GMUX	0x0	rw	UART4 的 IO 全局复用 (UART4 IO general muxing) 选择 UART4 的 IO 复用功能。 0000: TX/PC10 RX/PC11 0001: TX/PA0 RX/PA1 其他: 不使用
位 27: 24	USART3_GMUX	0x0	rw	USART3 的 IO 全局复用 (USART3 IO general muxing) 选择 USART3 的 IO 复用功能。 0000: TX/PB10, RX/PB11, CK/PB12, CTS/PB13, RTS/PB14; 0001: TX/PC10, RX/PC11, CK/PC12, CTS/PB13, RTS/PB14; 0011: TX/PD8, RX/PD9, CK/PD10, CTS/PD11, RTS/PD12 其他: 不使用
位 23: 20	USART2_GMUX	0x0	rw	USART1 的 IO 全局复用 (USART2 IO general muxing) 选择 USART2 的 IO 复用功能。 0000: CTS/PA0, RTS/PA1, TX/PA2, RX/PA3, CK/PA4; 0001: CTS/PD3, RTS/PD4, TX/PD5, RX/PD6, CK/PD7; 其他: 不使用
位 19: 16	USART1_GMUX	0x0	rw	USART1 的 IO 全局复用 (USART1 IO general muxing) 选择 USART1 的 IO 复用功能。 0000: TX/PA9, RX/PA10; 0001: TX/PB6, RX/PB7。 其他: 不使用
位 15: 12	SDIO2_GMUX	0x0	rw	SDIO2 的 IO 全局复用 (SDIO2 IO general muxing) 选择 SDIO2 的 IO 复用功能。 0000: D0/PC0 D1/PC1 D2/PC3 D4/PA4 D5/PA5 D6/PA6 D7/PA7 CK/PC4 CMD/PC5 0001: D0/PA4 D1/PA5 D2/PA6 D3/PA7 CK/PA2 CMD/PA3 其他: 不使用
位 11: 8	保留	0x0	resd	保持默认值。
位 7: 4	CAN2_GMUX	0x0	rw	CAN2 的 IO 全局复用 (CAN2 IO general muxing) 选择 CAN2 的 IO 复用功能。 0000: RX/PB12、TX/PB13 0001: RX/PB5、TX/PB6 其他: 不使用
位 3: 0	CAN1_GMUX	0x0	rw	CAN1 的 IO 全局复用 (CAN1 IO general muxing) 选择 CAN1 的 IO 复用功能。 00: RX/PA11、TX/PA12; 10: RX/ PB8、TX/ PB9; 11: RX/ PD0、TX/ PD1; 其他: 不使用

## 7.3.12 IO复用重映射寄存器7 (IOMUX\_REMAP7)

域	简称	复位值	类型	功能
位 31: 28	保留	0x0	resd	保持默认值。
位 27	XMC_NADV_GMUX	0x0	rw	XMC NADV 的 IO 全局复用 (XMC NADV IO general muxing)。 选择是否使用 XMC_NADV 信号。 0: XMC_NADV 连接到 pin。(默认) 1: XMC_NADV 不使用, 对应的 pin 可被其他外设使用。
位 26: 24	XMC_GMUX	0x0	rw	XMC 的 IO 全局复用 (XMC IO general muxing) 选择 XMC 的 IO 复用功能。 0000: NEW/PD5 D0/PD14 D1/PD15 D2/PD0 D3/PD1 D4/PE7 D5/PE8 D6/PE9 D7/PE10 D13/PD8 NOE/PD4 0001: NEW/PD2 D0/PB14 D1/PC6 D2/PC11 D3/PC12 D4/PA2 D5/PA3 D6/PA4 D7/PA5 D13/PB12 NOE/PC5 0010: NEW/PC2 D0/PB14 D1/PC6 D2/PC11 D3/PC12 D4/PA2 D5/PA3 D6/PA4 D7/PA5 D13/PB12 NOE/PC5 其他: 不使用
位 23: 21	保留	0x0	resd	保持默认值。
位 20	PD01_GMUX	0x0	rw	PD0/PD1 复用到 HEXT_IN/HEXT_OUT (PD0/PD1 mapping on HEXT_IN / HEXT_OUT) 选择 PD0 和 PD1 的 GPIO 功能复用。 此功能只适用于 48 和 64 管脚的封装。 0: 无复用; 1: PD0 复用到 HEXT_IN, PD1 复用到 HEXT_OUT。
位 19	保留	0x0	resd	保持默认值。
位 18: 16	SWJTAG_GMUX	0x0	rw	SWD JTAG 的 IO 全局复用 (SWD JTAG IO general muxing) 配置 SWJTAG 接口相关的 IO 是否作为 GPIO 使用。 000: 支持 SWD 和 JTAG, 所有 SWJTAG 管脚不可作 GPIO; 001: 支持 SWD 和 JTAG, 禁用 NJTRST, PB4 可作 GPIO; 010: 支持 SWD, 禁用 JTAG, PA15/PB3/PB4 可作 GPIO; 100: 禁用 SWD 和 JTAG, 所有 SWJTAG 管脚均可作 GPIO; 其它: 无作用。
位 15: 10	保留	0x00	resd	保持默认值。
位 9	ADC2_ETO_GMUX	0x0	rw	ADC2 普通转换外部触发重复用 (ADC2 external trigger ordinary conversion general muxing) 选择 ADC2 普通转换的外部触发输入。 0: ADC2 普通转换外部触发连接到 EXINT11; 1: ADC2 普通转换外部触发连接到 TMR8_TRGO。
位 8	ADC2_ETP_GMUX	0x0	rw	ADC2 抢占转换外部触发重复用 (ADC2 external trigger preempted conversion general muxing) 选择 ADC2 抢占转换外部触发输入。 0: ADC2 抢占转换外部触发连接到 EXINT15; 1: ADC2 抢占转换外部触发连接到 TMR8 通道 4。
位 7: 6	保留	0x0	resd	保持默认值。
位 5	ADC1_ETO_GMUX	0x0	rw	ADC1 普通转换外部触发重复用 (ADC1 external trigger ordinary conversion general muxing) 选择 ADC1 普通转换外部触发输入。 0: ADC1 普通转换外部触发连接到 EXINT11; 1: ADC1 普通转换外部触发连接到 TMR8_TRGO。
位 4	ADC1_ETP_GMUX	0x0	rw	ADC1 抢占转换外部触发重复用 (ADC1 external trigger preempted conversion general muxing) 选择 ADC1 抢占转换外部触发输入。 0: ADC1 抢占转换外部触发连接到 EXINT15; 1: ADC1 抢占转换外部触发连接到 TMR8 通道 4。

位 3	EXT_SPIM_GEN	0x0	rw	使能 SPIM 接口。 选择是否使用外部 SPI Flash。
位 2: 0	EXT_SPIM_GMUX	0x0	rw	选择 SPIM 接口的 IO 复用功能。 000: SCK/PB1 CS/PA8 IO0/PA11 IO1/PA12 IO2/PB7 IO3/PB6 001: SCK/PB1 CS/PA8 IO0/PB10 IO1/PB11 IO2/PB7 IO3/PB6 其他: 不使用

## 7.3.13 IO复用重映射寄存器8 (IOMUX\_REMAP8)

域	简称	复位值	类型	功能
位 31: 28	UART8_GMUX	0x0	rw	USART8 的 IO 全局复用 (UART8 IO general muxing) 选择 UART8 的 IO 复用功能。 0000: TX/PE1 RX/PE0 0001: TX/PC2 RX/PC3 其他: 不使用
位 27: 24	UART7_GMUX	0x0	rw	USART7 的 IO 全局复用 (UART7 IO general muxing) 选择 UART7 的 IO 复用功能。 0000: TX/PE8 RX/PE7 0001: TX/PB4 RX/PB3 其他: 不使用
位 23: 20	USART6_GMUX	0x0	rw	USART6 的 IO 全局复用 (USART6 IO general muxing) 选择 USART6 的 IO 复用功能。 0000: TX/PC6 RX/PC7 0001: TX/PA4 RX/PA5 其他: 不使用
位 19	PTP_PPS_GMUX	0x0	rw	以太网 PTP PPS 的 IO 全局复用 (EMAC PTP PPS IO general muxing) 选择 PPS_PTS 是否连接到 PB5。 0: PTP_PPS 不连接到 PB5 管脚; 1: PTP_PPS 连接到 PB5 管脚。 注: 该位只在 AT32F407 上有效。
位 18	MII_RMII_SEL GMUX	0x0	rw	MII 或 RMII 接口选择 (MII or RMII selection muxing) 选择以太网使用 MII 接口还是 RMII 接口。 0: MII; 1: RMII。 注: 该位只在 AT32F407 上有效。
位 17: 16	EMAC_GMUX	0x0	rw	以太网 MAC 的 IO 全局复用 (EMAC IO general muxing) 选择以太网 MAC 的 IO 复用功能。 0: RX_DV/CRS_DV/PA7、RXD0/PC4、RXD1/PC5、 RXD2/PB0、RXD3/PB1 1: RX_DV/CRS_DV/PD8、RXD0/PD9、RXD1/PD10、 RXD2/PD11、RXD3/PD12。 注: 该位只在 AT32F407 上有效。
位 15: 0	保留	0x0000	resd	保持默认值。

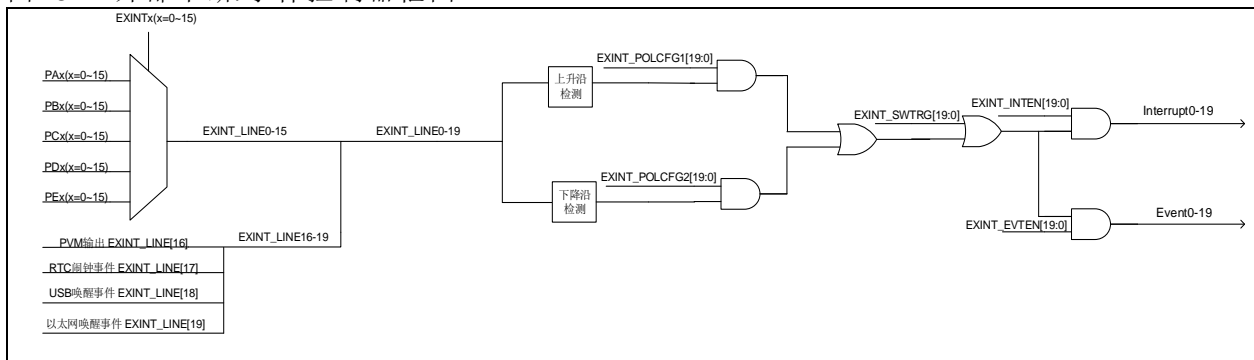


## 8 外部中断/事件控制器 (EXINT)

### 8.1 EXINT介绍

EXINT 共计有 20 条中断线 EXINT\_LINE[19:0]，每条中断线均支持通过边沿检测触发和软件触发来产生中断或事件。EXINT 可以根据软件配置，独立的使能或禁止中断或事件，并采取不同的边沿检测方式（检测上升沿或检测下降沿或同时检测上升沿和下降沿）以及触发方式（边沿检测触发或软件触发或边沿检测和软件同时触发）响应触发源独立的产生中断或事件。

图 8-1 外部中断/事件控制器框图



**EXINT 控制器的主要特性：**

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

### 8.2 功能描述和配置流程

EXINT 共计有 20 条中断线 EXINT\_LINE[19:0]，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，RTC 闹钟事件，USB 唤醒事件以及以太网唤醒事件共四种芯片内部的中断源，其中来自 GPIO 的中断源可以通过软件编程配置 IOMUX 中的复用外部中断配置寄存器 x（IOMUX\_EXINTCx）灵活的选择，需要注意的是这些输入源是互斥的，例如 EXINT\_LINE0 只能选择 PA0/PB0/PC0/PD0...中的某一个，而不能同时选择 PA0 和 PB0 作为输入源。

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

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

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

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

**中断初始化流程**

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



- 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）

#### 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

必须以字（32 位）的方式操作这些外设寄存器。

下表列出了 EXINT 寄存器的映像和复位值。

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: 20	保留	0x000	resd	硬件强制为 0。
位 19: 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 19 只适用于 AT32F407/407A，对于其它产品为保留位。

### 8.3.2 事件使能寄存器（EXINT\_EVTEN）

域	简称	复位值	类型	功能
位 31: 20	保留	0x000	resd	硬件强制为 0。
位 19: 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 19 只适用于 AT32F407/407A，对于其它产品为保留位。

### 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: 20	保留	0x000	resd	硬件强制为 0。
位 19: 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 19 只适用于 AT32F407/407A，对于其它产品为保留位。

## 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: 20	保留	0x000	resd	硬件强制为 0。
位 19: 0	FPx	0x00000	rw	<p>线 x 上的下降沿触发事件配置位 (Falling polarity event configuration bit of line x)</p> <p>这些位用于选择线 x 由下降沿触发中断和事件。</p> <p>0: 禁止下降沿触发;</p> <p>1: 允许下降沿触发。</p> <p>注: 位 19 只适用于 AT32F407/407A, 对于其它产品为保留位。</p>

## 8.3.5 软件触发寄存器 (EXINT\_SWTRG)

域	简称	复位值	类型	功能
位 31: 20	保留	0x000	resd	硬件强制为 0。
位 19: 0	SWTx	0x00000	rw	<p>软件触发线 x (Software trigger on line x)</p> <p>当中断使能寄存器 (EXINT_INTEN) 中的对应位为 1, 则软件写此位硬件将自动置起中断状态寄存器 (EXINT_INTSTS) 中的对应位并产生中断。</p> <p>当事件使能寄存器 (EXINT_EVTEN) 中的对应位为 1, 则软件写此位硬件将自动产生对应中断线上的事件。</p> <p>0: 默认值;</p> <p>1: 产生软件触发。</p> <p>注: 通过清除中断状态寄存器 (EXINT_INTSTS) 的对应位 (写入 1), 可以清除该位为 0。</p> <p>注: 位 19 只适用于 AT32F407/407A, 对于其它产品为保留位。</p>

## 8.3.6 中断状态寄存器 (EXINT\_INTSTS)

域	简称	复位值	类型	功能
位 31: 20	保留	0x000	resd	硬件强制为 0。
位 19: 0	LINEx	0x00000	rw	<p>线 x 状态位 (Line x state bit)</p> <p>0: 没有发生中断;</p> <p>1: 发生了中断。</p> <p>注: 在该位中写入 '1' 可以清除它。</p> <p>注: 位 19 只适用于 AT32F407/407A, 对于其它产品为保留位。</p>

## 9 DMA 控制器（DMA）

### 9.1 简介

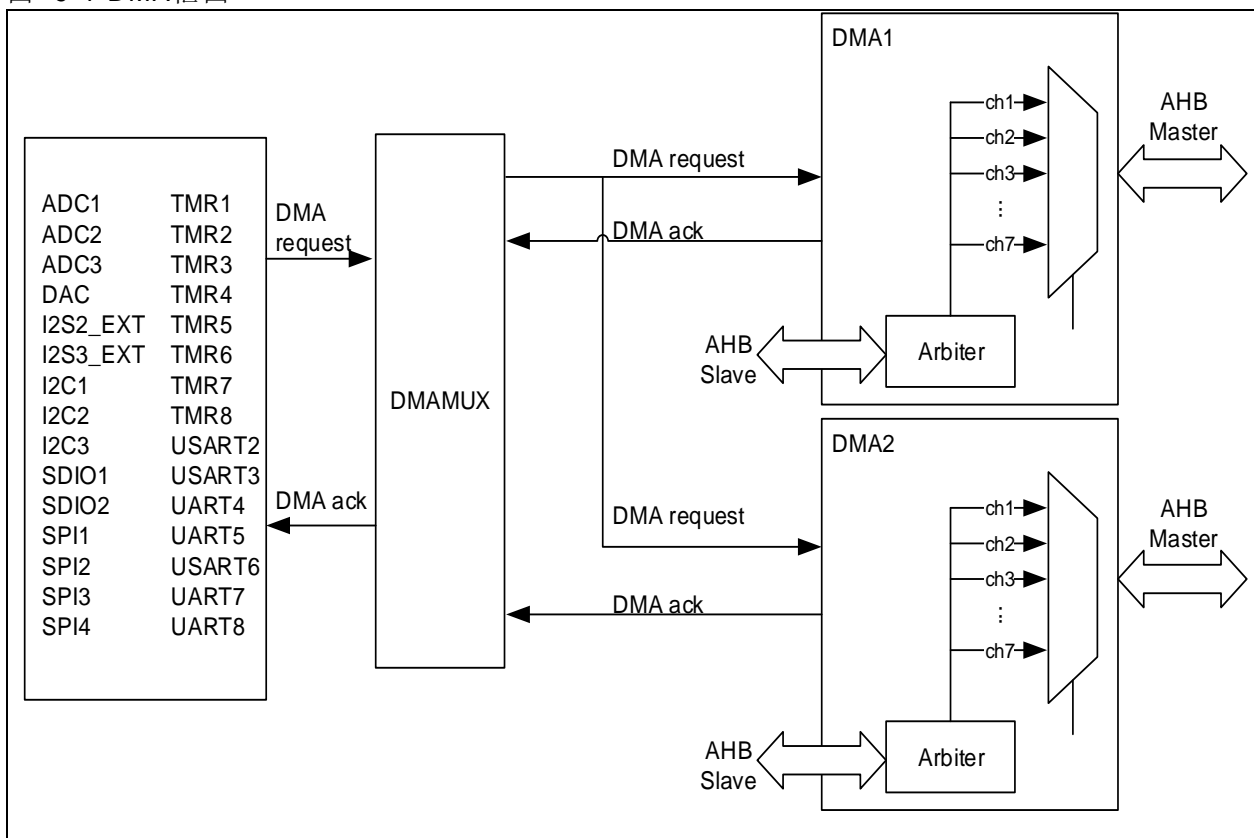
直接存储器访问（DMA）控制器，不仅旨在增强系统性能并减少处理器的中断生成，而且还针对 32 位 MCU 应用程序而设计。

一个处理器包含 2 个 DMA 控制器。每个控制器各有 7 个 DMA 通道，每个通道管理来自于外设对存储器访问的请求，并由仲裁器来协调各个 DMA 请求的优先权。

### 9.2 特性

- 符合 AMBA 规范（Rev. 2.0）
- 仅支持 AHB OKAY 和 ERROR 响应
- 不支持 AHB 主接口的 HBUSREQ 和 HGRANT
- 支持 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
- 支持弹性映射

图 9-1 DMA框图



注意： 根据不同型号，图中 DMA 外设可能会有所减少。

### 9.3 功能描述

#### 9.3.1 通道配置

1. 设置外设地址（DMA通道x外设地址寄存器（DMA\_CxPADDR））  
数据传输的初始外设地址，在传输过程中不会被改变。
2. 设置存储器地址（DMA通道x存储器地址寄存器（DMA\_CxMADDR））

数据传输的初始存储器地址，在传输过程中不会被改变。

### 3. 配置数据传输量（DMA通道x数据传输量寄存器（DMA\_CxDTCNT））

可编程的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通道x配置寄存器（DMA\_CxCTRL））

包含通道优先级，数据传输的方向、宽度，地址增量模式、循环模式和中断方式。

#### ● 通道优先级（CHPL）

分为4个等级，最高优先级、高优先级、中等优先级和低优先级。

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

#### ● 数据传输方向（DTD）

分为存储器到外设（M2P），外设到存储器（P2M）。

#### ● 地址增量模式（PINC/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\_CxDTCNT 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通道x配置寄存器（DMA\_CxCTRL）的CHEN位）

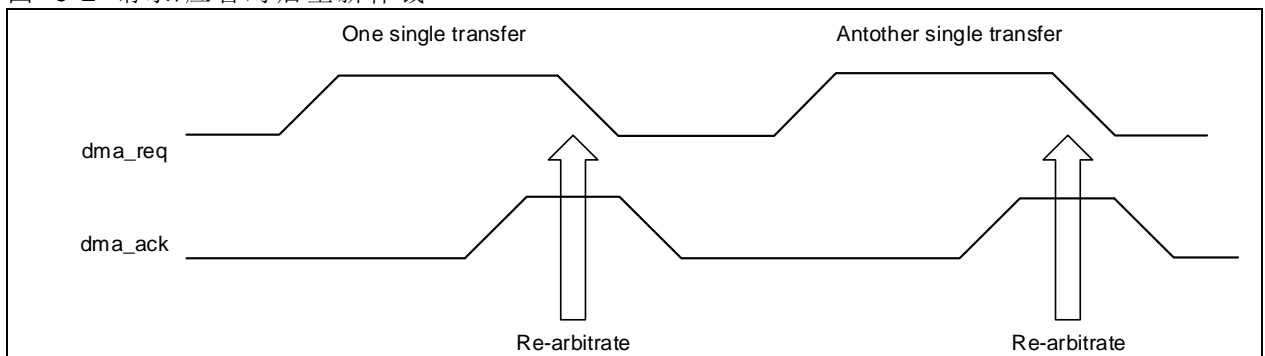
## 9.3.2 握手机制

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

## 9.3.3 仲裁

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

图 9-2 请求/应答对后重新仲裁



## 9.3.4 可编程数据传输宽度

通过 DMA 通道 x 配置寄存器（DMA\_CxCTRL）中的 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 PWIDTH: byte, MWIDTH: half-word

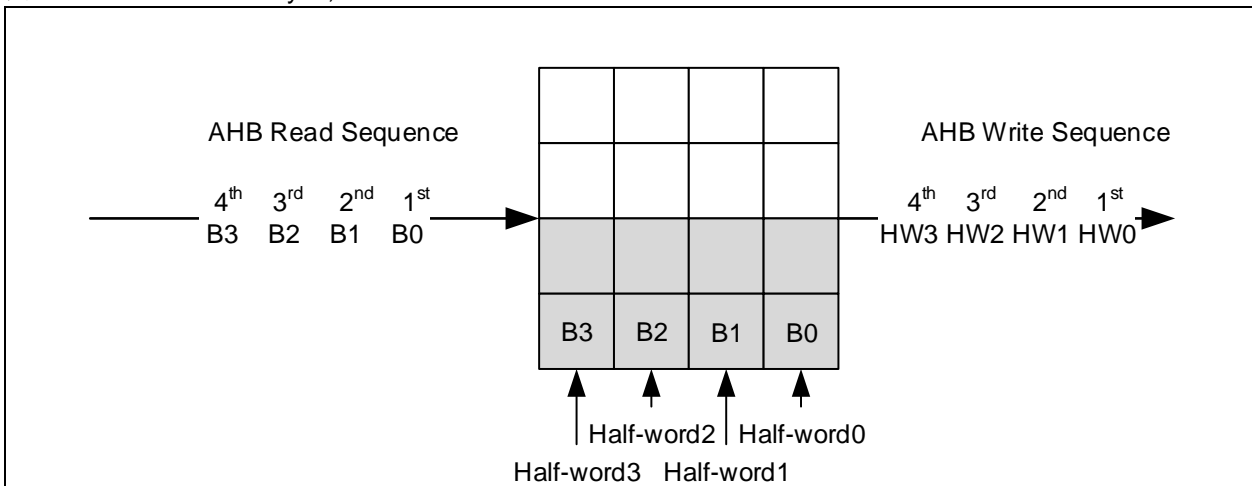


图 9-4 PWIDTH: half-word, MWIDTH: word

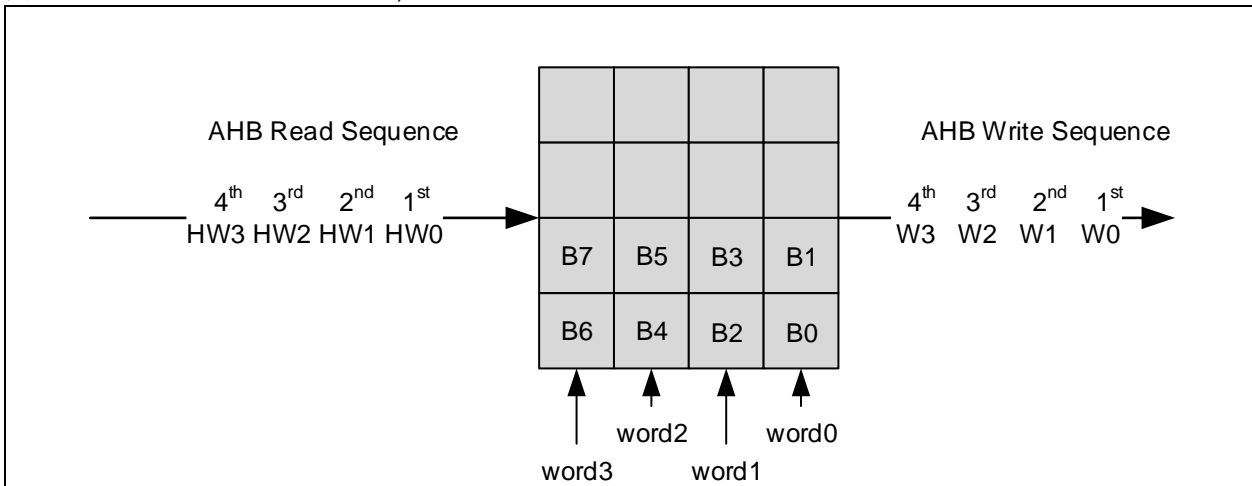
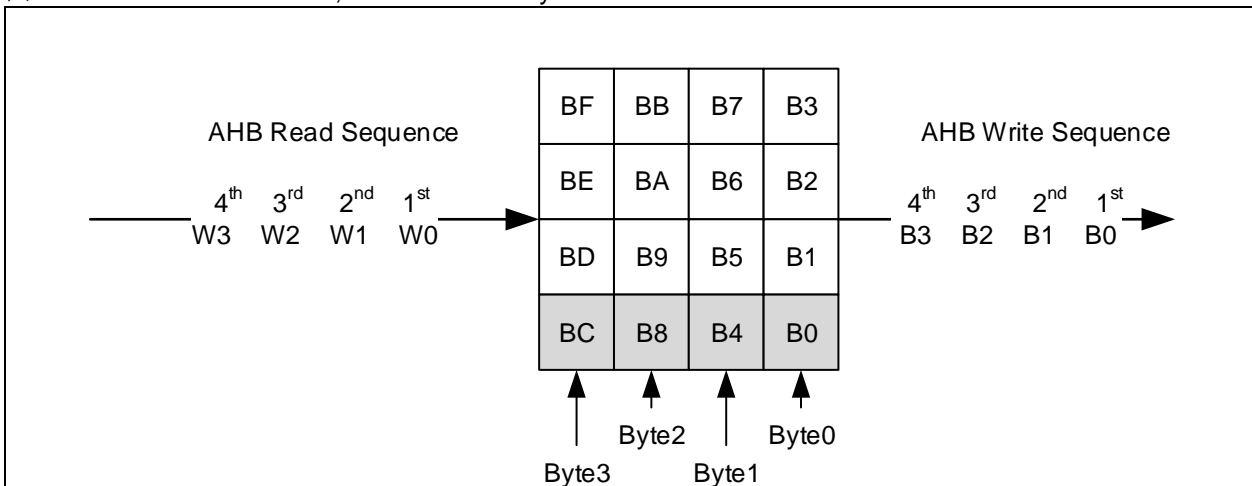


图 9-5 PWIDTH: word, MWIDTH: byte



### 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

注意：DMA2 通道 4/通道 5，通道 6/通道 7 的中断被映射在同一个中断向量上。

## 9.3.7 DMA固定请求映射

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

表 9-3 DMA1各通道的外设请求

外设	通道 1	通道 2	通道 3	通道 4	通道 5	通道 6	通道 7
ADC1	ADC1						
SPI/I <sup>2</sup> S		SPI1/I2S1_RX	SPI1/I2S1_TX	SPI2/I2S2_RX	SPI2/I2S2_TX		
USART		USART3_TX	USART3_RX	USART1_TX	USART1_RX	USART2_RX	USART2_TX
I <sup>2</sup> C		I2C3_TX	I2C3_RX	I2C2_TX	I2C2_RX	I2C1_TX	I2C1_RX
TMR1		TMR1_CH1	TMR1_CH2	TMR1_CH4 TMR1_TRIG TMR1_HALL	TMR1_UP	TMR1_CH3	
TMR2	TMR2_CH3	TMR2_UP			TMR2_CH1		TMR2_CH2 TMR2_CH4
TMR3		TMR3_CH3	TMR3_CH4 TMR3_UP			TMR3_CH1 TMR3_TRIG	
TMR4	TMR4_CH1			TMR4_CH2	TMR4_CH3		TMR4_UP

表 9-4 DMA2各通道的外设请求

外设	通道 1	通道 2	通道 3	通道 4	通道 5	通道 6	通道 7
ADC3					ADC3		
SPI/I <sup>2</sup> S	SPI3/I2S3_RX	SPI3/I2S3_TX	SPI4/I2S4_RX	SPI4/I2S4_TX			
UART4			UART4_RX		UART4_TX		
SDIO				SDIO			
SDIO2					SDIO2		
TMR5	TMR5_CH4 TMR5_TRIG	TMR5_CH3 TMR5_UP		TMR5_CH2	TMR5_CH1		
TMR6/ DAC CH1			TMR6_UP/ DAC CH1				
TMR7/ DAC CH2				TMR7_UP/ DAC CH2			
TMR8	TMR8_CH3 TMR8_UP	TMR8_CH4 TMR8_TRIG TMR8_HALL	TMR8_CH1		TMR8_CH2		

## 9.3.8 DMA弹性请求映射

当设定弹性模式时（DMA\_FLEX\_EN = 1），每个通道的请求来源由 CHx\_SRC 来设定[x=1~7]。

使用例子：假如 DMA 通道 1 指定成 USART3\_TX，通道 3 要指定成 USART3\_RX，其他不使用，则设定上必须是 DMA\_FLEX\_EN=1，CH1\_SRC=30，CH3\_SRC=29，CH[2/4/5/6/7]\_SRC=0。

CHx\_SRC 设定值对应请求来源见下表：

表 9-5 DMA各通道的弹性请求

CHx_SRC	请求来源	CHx_SRC	DMA 来源	CHx_SRC	请求来源	CHx_SRC	请求来源
0	No select	1	ADC1	2	reserved	3	ADC3
4	reserved	5	DAC1	6	DAC2	7	reserved
8	reserved	9	SPI1_RX	10	SPI1_TX	11	SPI2_RX
12	SPI2_TX	13	SPI3_RX	14	SPI3_TX	15	SPI4_RX
16	SPI4_TX	17	I2S2EXT_RX	18	I2S2EXT_TX	19	I2S3EXT_RX

20	I2S3EXT_TX	21	reserved	22	reserved	23	reserved
24	reserved	25	USART1_RX	26	USART1_TX	27	USART2_RX
28	USART2_TX	29	USART3_RX	30	USART3_TX	31	UART4_RX
32	UART4_TX	33	UART5_RX	34	UART5_TX	35	USART6_RX
36	USART6_TX	37	UART7_RX	38	UART7_TX	39	UART8_RX
40	UART8_TX	41	I2C1_RX	42	I2C1_TX	43	I2C2_RX
44	I2C2_TX	45	I2C3_RX	46	I2C3_TX	47	reserved
48	reserved	49	SDIO1	50	SDIO2	51	reserved
52	reserved	53	TMR1_TRIG	54	TMR1_HALL	55	TMR1_UP
56	TMR1_CH1	57	TMR1_CH2	58	TMR1_CH3	59	TMR1_CH4
60	reserved	61	TMR2_TRIG	62	reserved	63	TMR2_UP
64	TMR2_CH1	65	TMR2_CH2	66	TMR2_CH3	67	TMR2_CH4
68	reserved	69	TMR3_TRIG	70	reserved	71	TMR3_UP
72	TMR3_CH1	73	TMR3_CH2	74	TMR3_CH3	75	TMR3_CH4
76	reserved	77	TMR4_TRIG	78	reserved	79	TMR4_UP
80	TMR4_CH1	81	TMR4_CH2	82	TMR4_CH3	83	TMR4_CH4
84	reserved	85	TMR5_TRIG	86	reserved	87	TMR5_UP
88	TMR5_CH1	89	TMR5_CH2	90	TMR5_CH3	91	TMR5_CH4
92	reserved	93	reserved	94	reserved	95	TMR6_UP
96	reserved	97	reserved	98	reserved	99	reserved
100	reserved	101	reserved	102	reserved	103	TMR7_UP
104	reserved	105	reserved	106	reserved	107	reserved
108	reserved	109	TMR8_TRIG	110	TMR8_HALL	111	TMR8_UP
112	TMR8_CH1	113	TMR8_CH2	114	TMR8_CH3	115	TMR8_CH4
116	reserved	117	reserved	118	reserved	119	reserved
120	reserved	121	reserved	122	reserved	123	reserved
124	reserved	125	reserved	126	reserved	127	reserved
128	reserved	129	reserved	130	reserved	131	reserved
132	reserved	133	reserved	134	reserved	135	reserved
136	reserved	137	reserved	138	reserved	139	reserved
140	reserved	141	reserved	142	reserved	143	reserved
144	reserved	145	reserved	146	reserved	147	reserved
148	reserved	149	reserved	150	reserved	151	reserved
152	reserved	153	reserved	154	reserved	155	reserved
156	reserved	157	reserved	158	reserved	159	reserved
160	reserved	161	reserved	162	reserved	163	reserved
164	reserved	165	reserved	166	reserved	167	reserved
168	reserved	169	reserved	170	reserved	171	reserved
172	reserved	173	reserved	174	reserved	175	reserved

注意：DMA1、DMA2 的请求映射模式必须一致（DMA1、DMA2 的 DMA\_FLEX\_EN 必须同时设定为 1 或 0）。



## 9.4 DMA寄存器

下表列出了 DMA 寄存器的映像和复位值。

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

表 9-6 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_SRC_SEL0	0xA0	0x0000 0000
DMA_SRC_SEL1	0xA4	0x0000 0000

注意：在以下列举的所有寄存器中，所有与通道 6 和通道 7 相关的位，对 DMA2 固定请求映像不适用，因为 DMA2 固定请求映像只有 5 个通道。对于 DMA2 的弹性请求映像，则可支持到 7 个通道。

## 9.4.1 DMA状态寄存器（DMA\_STS）

域	简称	复位值	类型	功能
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

## 9.4.2 DMA状态清除寄存器 (DMA\_CLR)

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

位 0	GFC1	0x0	rw1c	清除通道 1 的全局中断标志（Global flag clear） 0：无效 1：清除 DMA_STS 寄存器中 DTERRF1、HDTF1、FDTF1 和 GF1 标志
-----	------	-----	------	---

## 9.4.3 DMA通道x配置寄存器（DMA\_CxCTRL）（x = 1…7）

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

#### 9.4.4 DMA通道x数据传输量寄存器（DMA\_CxDTCNT）（x = 1…7）

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

#### 9.4.5 DMA通道x外设地址寄存器（DMA\_CxPADDR）（x = 1…7）

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。 注：确保对应通道的 CHEN 位为 0，否则无法写入。

#### 9.4.6 DMA通道x存储器地址寄存器（DMA\_CxMADDR）（x = 1…7）

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

#### 9.4.7 通道来源寄存器0（DMA\_SRC\_SEL0）

域	简称	复位值	类型	功能
位 31: 24	CH4_SRC	0x00	rw	CH4 来源的选择位（CH4 source select） 当 DMA_FLEX_EN=1 时，由 CH4_SRC 选择通道 4 来源，详见 <a href="#">9.3.8DMA 弹性请求</a>
位 23: 16	CH3_SRC	0x00	rw	CH3 来源的选择位（CH3 source select） 当 DMA_FLEX_EN=1 时，由 CH3_SRC 选择通道 3 来源，详见 <a href="#">9.3.8DMA 弹性请求</a>
位 15: 8	CH2_SRC	0x00	rw	CH2 来源的选择位（CH2 source select） 当 DMA_FLEX_EN=1 时，由 CH2_SRC 选择通道 2 来源，详见 <a href="#">9.3.8DMA 弹性请求</a>
位 7: 0	CH1_SRC	0x00	rw	CH1 来源的选择位（CH1 source select） 当 DMA_FLEX_EN=1 时，由 CH1_SRC 选择通道 1 来源，详见 <a href="#">9.3.8DMA 弹性请求</a>



### 9.4.8 通道来源寄存器1 (DMA\_SRC\_SEL1)

域	简称	复位值	类型	功能
位 31: 25	保留	0x00	resd	保持默认值。
位 24	DMA_FLEX_EN:	0x0	rw	DMA 请求映像模式选择位。(DMA flexible mapping enable) 0: DMA 请求映像模式为固定模式 1: DMA 请求映像模式为弹性模式
位 23: 16	CH7_SRC	0x00	rw	CH7 来源的选择位 (CH7 source select) 当 DMA_FLEX_EN=1 时, 由 CH7_SRC 选择通道 7 来源, 详见 <a href="#">9.3.8DMA 弹性请求</a>
位 15: 8	CH6_SRC	0x00	rw	CH6 来源的选择位 (CH6 source select) 当 DMA_FLEX_EN=1 时, 由 CH6_SRC 选择通道 6 来源, 详见 <a href="#">9.3.8DMA 弹性请求</a>
位 7: 0	CH5_SRC	0x00	rw	CH5 来源的选择位 (CH5 source select) 当 DMA_FLEX_EN=1 时, 由 CH5_SRC 选择通道 5 来源, 详见 <a href="#">9.3.8DMA 弹性请求</a>

## 10 CRC 计算单元（CRC）

### 10.1 CRC 介绍

CRC 计算单元是一个独立的具备 CRC 计算功能的外设，CRC 计算单元采用 CRC32 标准。

用户可以通过软件编程配置控制寄存器（CRC\_CTRL）选择是否进行输入数据翻转（全字翻转，REVOD=1）或输出数据翻转（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 中的值搬入数据寄存器（CRC\_DT）。

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

**CRC 主要特性：**

- 采用 CRC-32 标准
- 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数据寄存器（CRC\_DT）实现
- 配置初始化寄存器（CRC\_IDT）写入初始化值，在每次 CRC 复位后该值会加载到数据寄存器（CRC\_DT）

### 10.2 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

#### 10.2.1 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。

#### 10.2.2 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。

## 10.2.3 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: 1	保留	0x0	resd	保持默认值。
位 0	RST	0x0	rw	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。

## 10.2.4 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。

## 11 I<sup>2</sup>C 接口

### 11.1 I<sup>2</sup>C 简介

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

### 11.2 I<sup>2</sup>C 主要特点

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

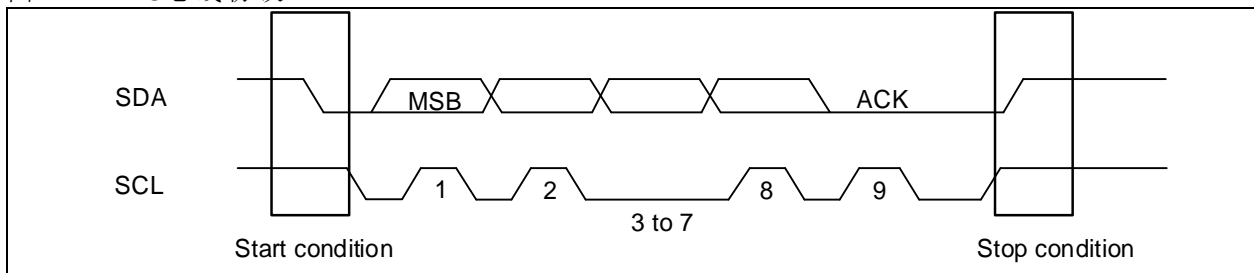
### 11.3 I<sup>2</sup>C 总线特性

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

开始信号: SCL 为高电平时, SDA 由高电平变为低电平。

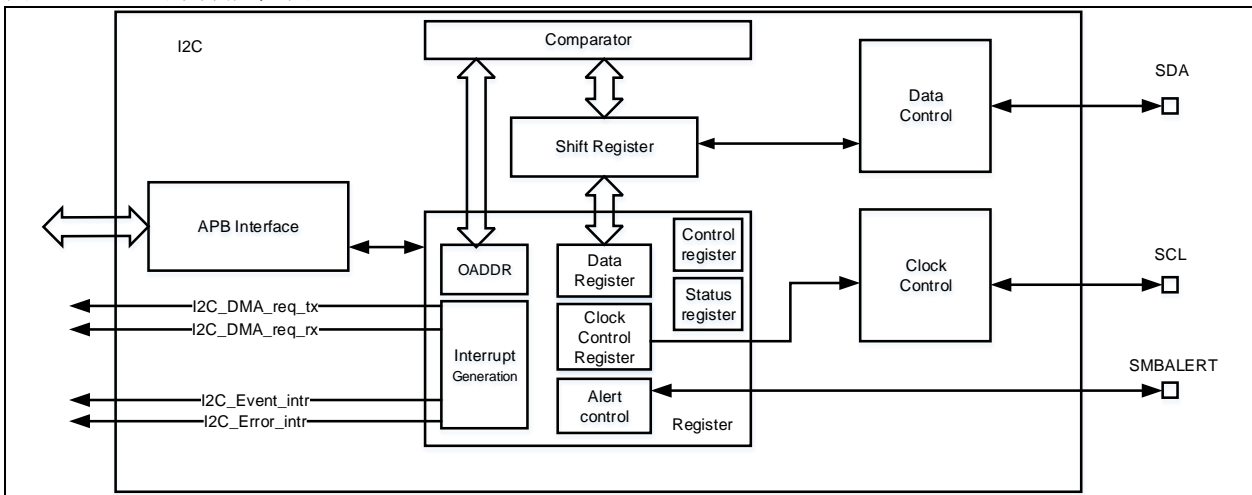
结束信号: SCL 为高电平时, SDA 由低电平变为高电平。

图 11-1 I<sup>2</sup>C 总线协议



### 11.4 I<sup>2</sup>C 接口

I<sup>2</sup>C 接口的功能框图示于下图。

图 11-2 I<sup>2</sup>C 的功能框图

## 1. I<sup>2</sup>C时钟

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

## 2. 接口工作模式

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

- 主机发送模式
- 主机接收模式
- 从机发送模式
- 从机接收模式

### 3. 通信流程

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

#### 4. 地址控制

### 主机和从机都支持 7 位和 10 位地址模式

从机地址模式:

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

从机特殊地址支持:

- 广播地址（0b00000000x）：当 GCAEN=1 时该地址启用

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

#### 从机地址匹配流程：

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

#### 5. 时钟延展功能

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

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

## 11.4.1 I<sup>2</sup>C从机通信流程

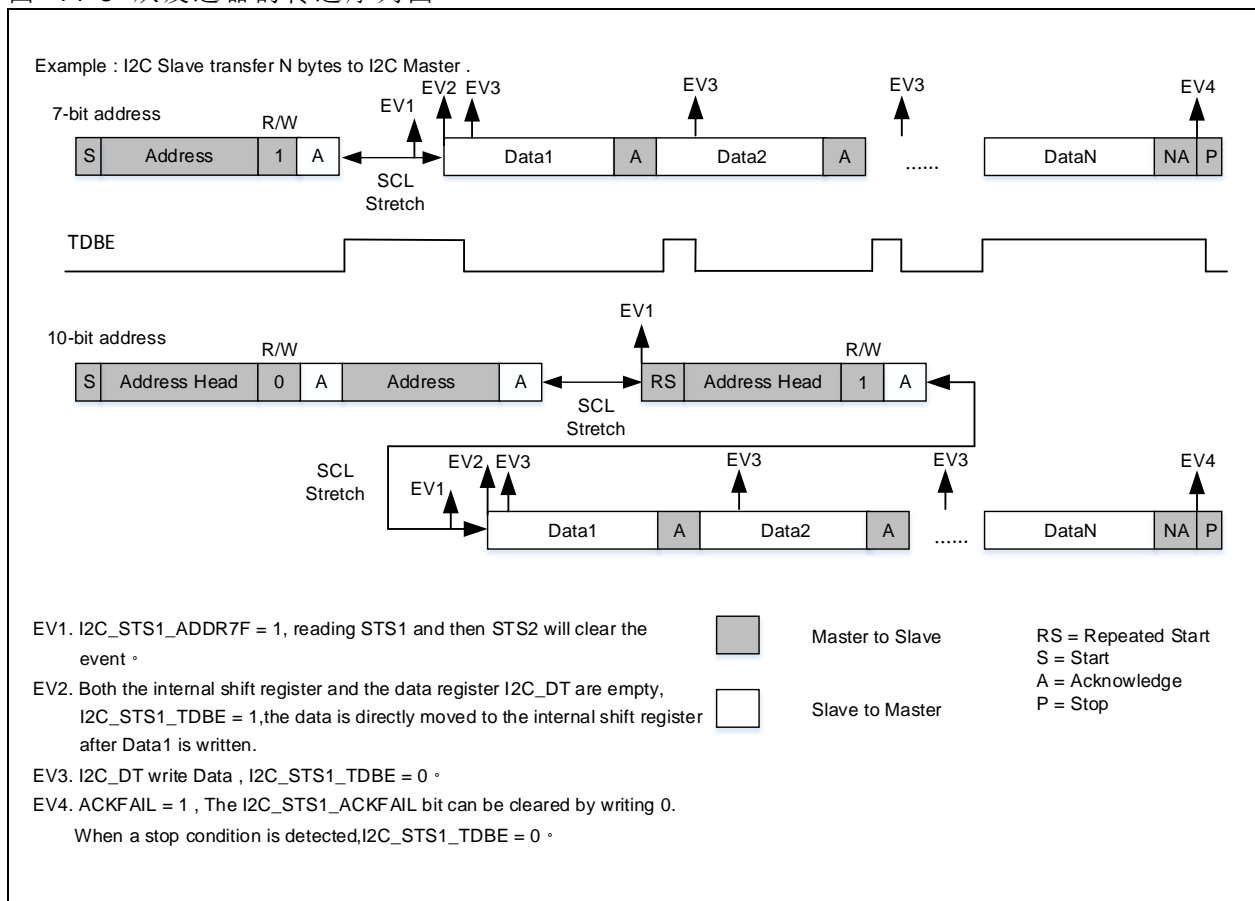
### 初始化

使能 I<sup>2</sup>C 外设时钟并配置控制寄存器 2 (I2C\_CTRL2) 中相关时钟位确保正确的时序，接着等待 I<sup>2</sup>C 主机发送开始讯号。

### 发送器

从机发送数据主要有以下操作流程，初始化后软件可以参照以下步骤进行操作：

图 11-3 从发送器的传送序列图



### 7位地址模式：

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

### 10位地址模式：

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

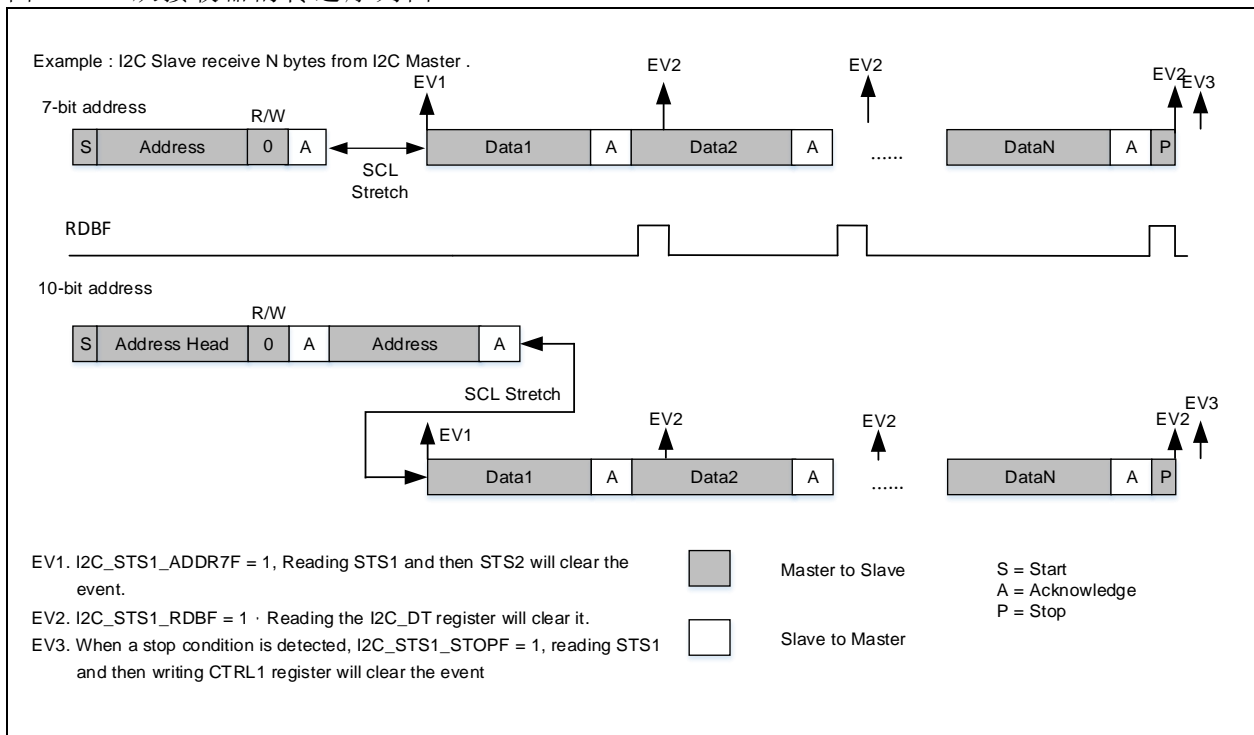


6. EV4: 收到主机发送的ACKFAIL事件, 此时ACKFIAL=1, 向ACKFIAL写0清除该事件
7. 通信结束

## 从接收器

从机接收数据主要有以下操作流程, 初始化后软件可以参照以下步骤进行操作 :

图 11-4 从接收器的传送序列图



## 7位地址模式:

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

## 10位地址模式:

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

## 11.4.2 I<sup>2</sup>C主机通信流程

### 主机模式初始化

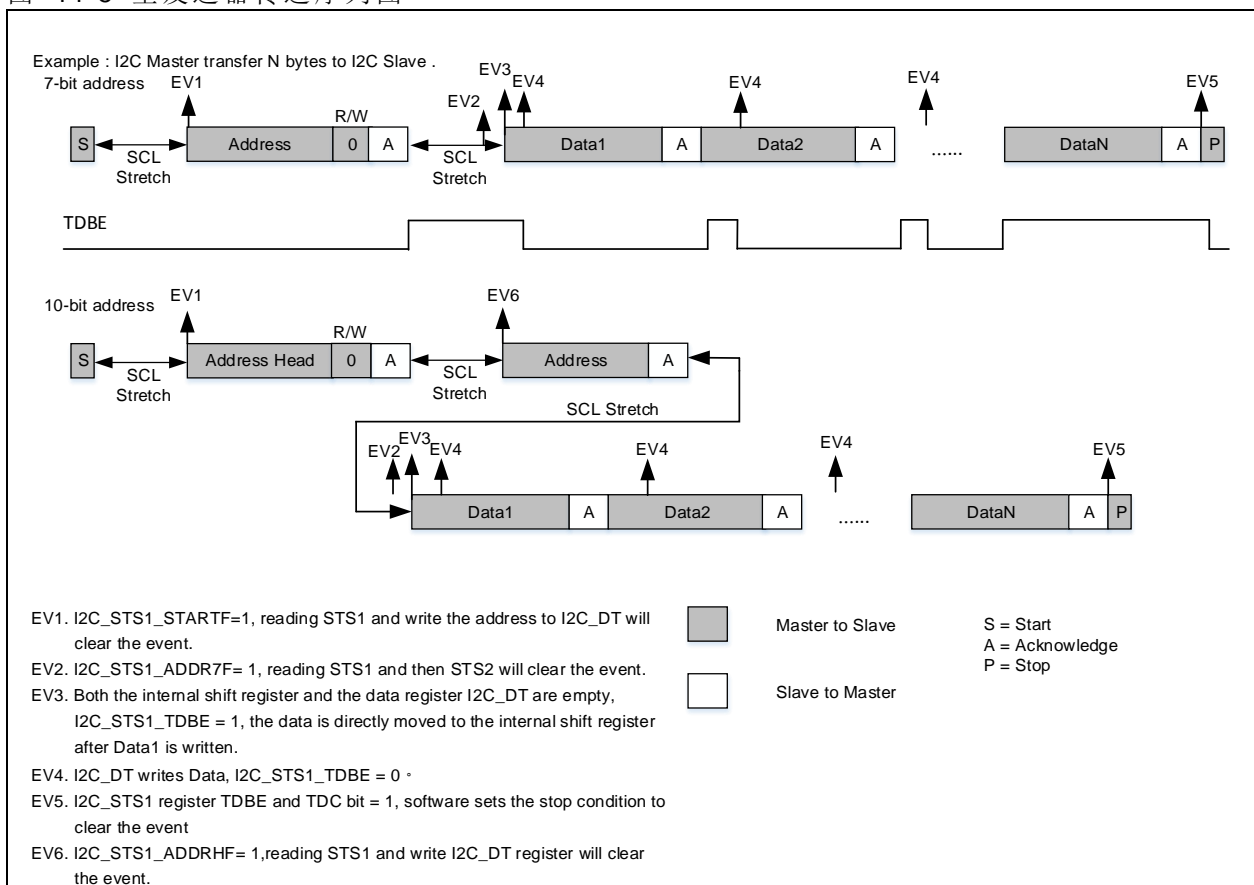
1. 设置输入时钟以产生正确的时序 (控制寄存器2 (I2C\_CTRL2) 中的CLKFREQ位);
2. 设置I<sup>2</sup>C的通信速度 (时钟控制寄存器 (I2C\_CLKCTRL));
3. 设置总线最大上升时间 (I2C\_TMRISIE寄存器);
4. 设置控制寄存器1 (I2C\_CTRL1);
5. 启动外设, 若设置GENSTART位在启动外设时会在总在线产生开始信号, 设备会进入主机模式从机地址发送

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

- 7 位地址模式：
  - 发送模式：发送的地址最低位为 0 时，进入发送器模式；
  - 接收模式：发送的地址最低位为 1 时，进入接收器模式。
- 10 位地址模式：
  - 发送模式：先发送从机地址头 0b11110xx0 (xx 为地址[9: 8])，再发送从机地址[7: 0]，主机进入发送器模式；
  - 接收模式：先发送从机地址头 0b11110xx0 (xx 为地址[9: 8])，再发送从机地址[7: 0]，再发送从机地址头 0b11110xx1 (xx 为地址[9: 8])，主机进入接收器模式。

## 主发送器

图 11-5 主发送器传送序列图



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

4. EV2: 成功匹配到地址 (ADDR7F=1), 软件先读取STS1, 再读取STS2清除ADDR7F位, 此时主机进入发送阶段, DT寄存器和内部移位寄存器皆为空, 硬件将TDBE位置1
5. EV3: 向DT寄存器写入数据, 此时数据会被立即送到移位寄存器并释放SCL总线, 此时TDBE仍然为1
6. EV4: 此时DT数据寄存器空, 移位寄存器非空, 向DT寄存器写入数据, 此时TDBE清零
7. TDBE位在倒数第二个字节发送完成后置起
8. EV5: TDC=1, 字节发送结束, 主机发送结束信号 (STOPF=1), 硬件自动清除TDBE位和TDC位
9. 通信结束

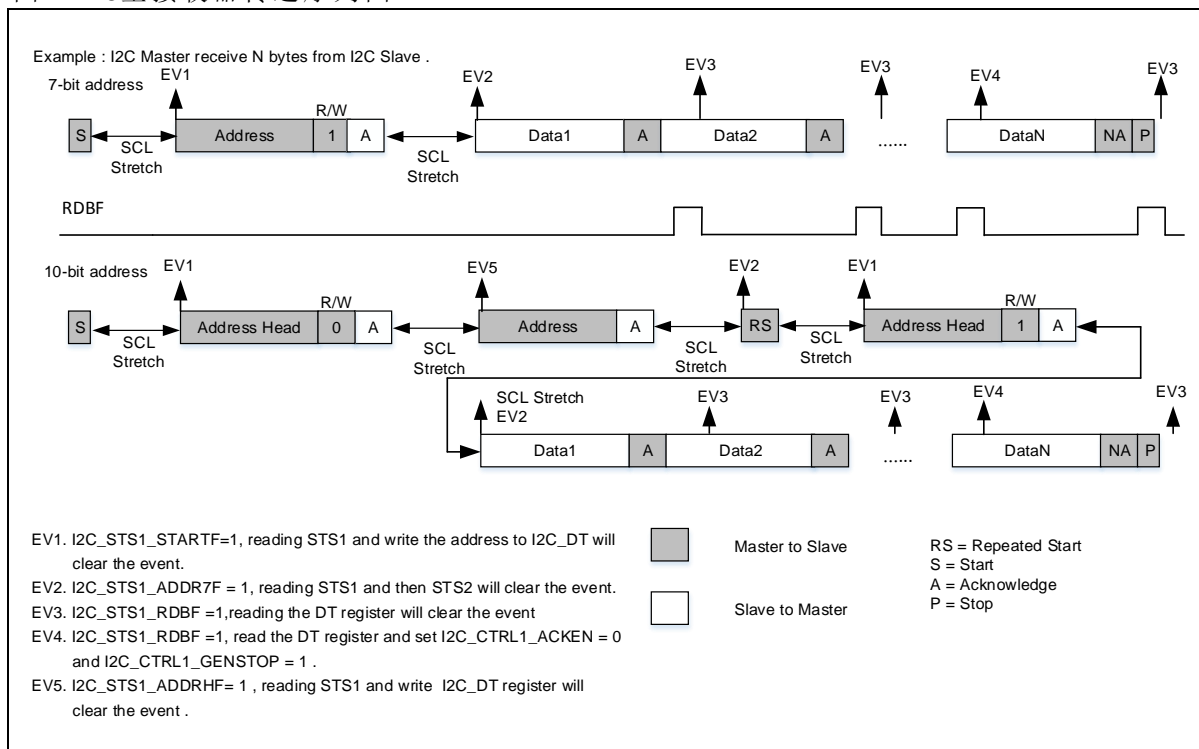
### 主接收器

主机接收数据可依 I<sup>2</sup>C 中断优先级分为几种情况:

#### 1. I<sup>2</sup>C中断为最高优先级

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

图 11-6主接收器传送序列图



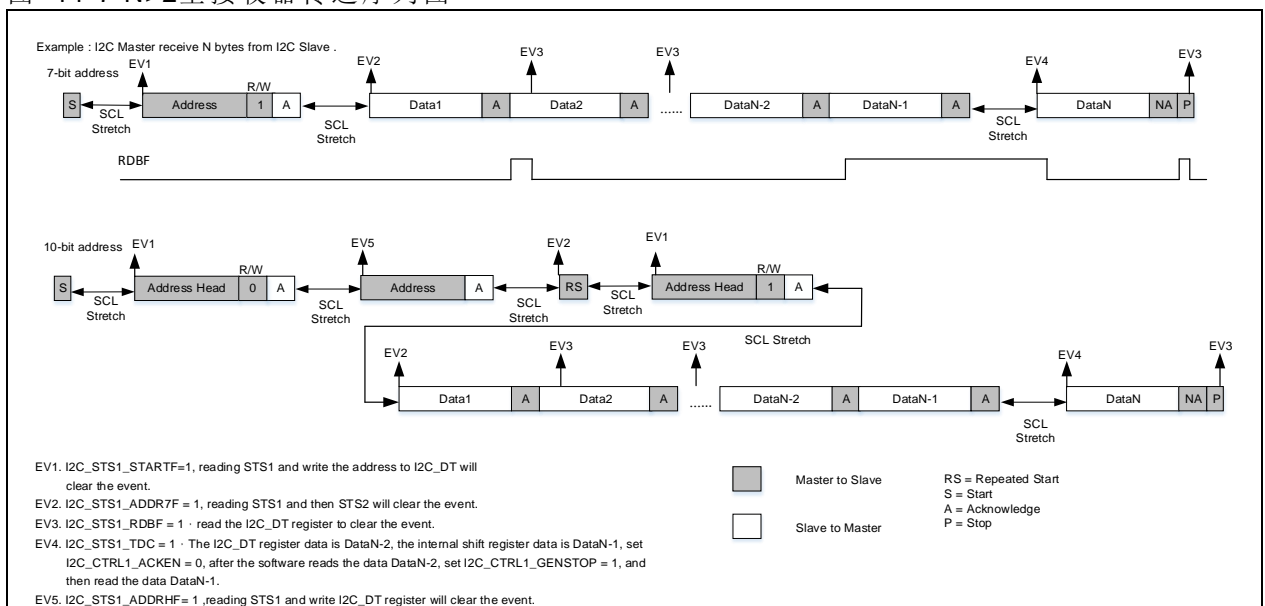
- 7 位地址模式:
  1. 发送开始信号 (GENSTART=1)
  2. EV1: 开始信号产生完成 (STARTF=1), 软件先读取STS1, 然后将地址写入DT寄存器
  3. EV2: 成功匹配到地址 (ADDR7F=1), 软件先读取STS1, 再读取STS2清除ADDR7F位, 此时主机进入接收阶段
  4. EV3: 在接收到字节后, RDBF位被置1, 软件读取数据寄存器 (I2C\_DT), RDBF位被清0
  5. EV4: 接收完倒数第二个字节后, 软件需立即将ACKEN位清0, GENSTOP位置1
  6. EV3: 在接收到字节后, RDBF位被置1, 软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

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

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

- 在接收到倒数第三个字节(N-2)时不进行读取, 待收到倒数第二个字节(N-1)时, 清除控制寄存器 1 (I2C\_CTRL1) 的 ACKEN 位, 接着读取倒数第三个字节(N-2), 设置控制寄存器 1 (I2C\_CTRL1) 的 GENSTOP 位后读取倒数第二个字节(N-1), 接着总线开始接收最后字节。

图 11-7 N>2主接收器传送序列图



## ● 7 位地址模式:

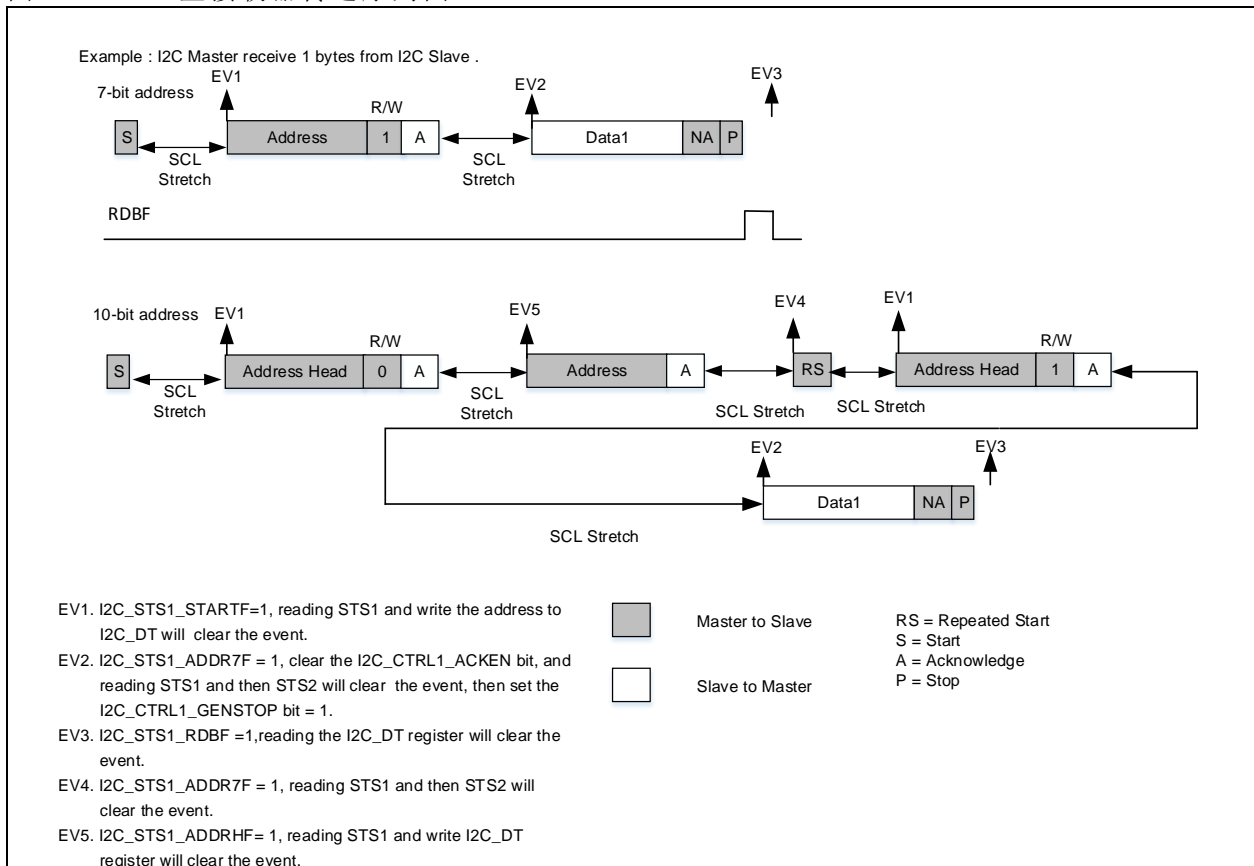
1. 发送开始信号 (GENSTART=1)
2. EV1: 开始信号产生完成 (STARTF=1), 软件先读取STS1, 然后将地址写入DT寄存器
3. EV2: 成功匹配到地址 (ADDR7F=1), 软件先读取STS1, 再读取STS2清除ADDR7F位, 此时主机进入接收阶段
4. EV3: 在接收到字节后, RDBF位被置1, 软件读取I2C\_DT寄存器, RDBF位被清0
5. EV4: TDC=1, 数据寄存器 (I2C\_DT) 内容为N-2, 移位寄存器内容为数据N-1, 软件将ACKEN位置0并读取数据N-2, 接着设置GENSTOP=1, 然后读数据N-1
6. EV3: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

## ● 10 位地址模式:

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

8. EV4: TDC=1, 数据寄存器(I2C\_DT)内容为N-2, 移位寄存器内容为数据N-1, 软件将ACKEN位置0并读取数据N-2, 接着设置GENSTOP=1, 然后读数据N-1
  9. EV3: 在接收到字节后, RDBF位被置1, 软件读取数据寄存器(I2C\_DT), RDBF位被清0
  10. 通信结束
3. I<sup>2</sup>C中断非最高优先级且要接收的字节数等于2
- 在接收数据前设置控制寄存器1(I2C\_CTRL1)的MACKCTRL位, 待地址匹配后, 先清除ACKEN位, 后清除ADDR7F位, 待TDC位置1后设置控制寄存器1(I2C\_CTRL1)的GENSTOP位, 接着读取DT寄存器。

图 11-8 N=2主接收器传送序列图



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



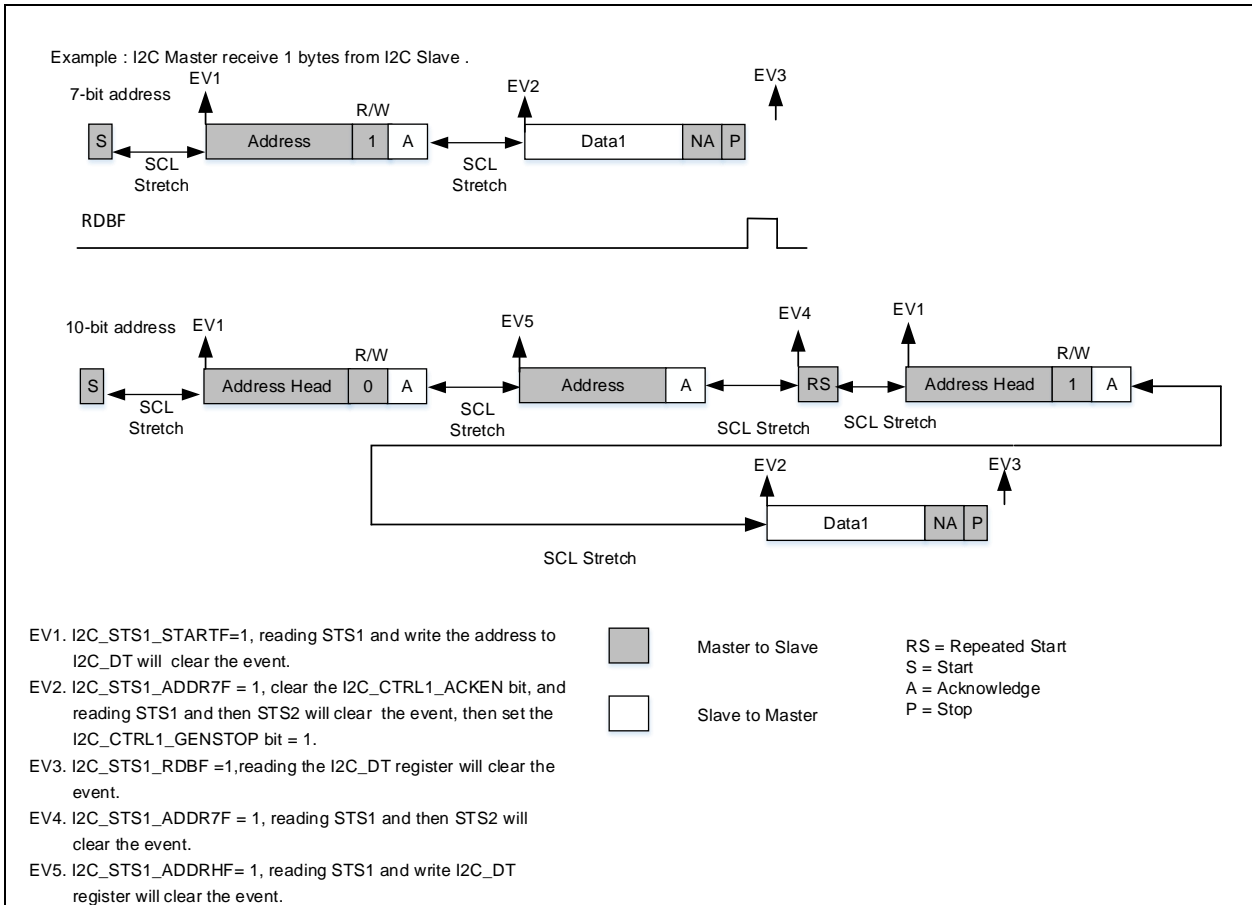
## 9. 通信结束

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

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

主机接收数据主要有以下操作流程，主机模式初始化后软件可以参照以下步骤进行操作：

图 11-9 N=1主接收器传送序列图



### ● 7 位地址模式：

1. 发送开始信号 (GENSTART=1)
2. EV1: 开始信号产生完成 (STARTF=1)，软件先读取STS1，然后将地址写入DT寄存器
3. EV2: 成功匹配到地址 (ADDR7F=1)，首先清除ACKEN位，然后先读取STS1，再读取STS2清除ADDR7F位，接着设置GENSTOP=1，此时主机进入接收阶段
4. EV3: RDBF=1，读取I2C\_DT寄存器，RDBF位被清0
5. 通信结束

### ● 10 位地址模式：

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

## 11.4.3 利用DMA传输

I<sup>2</sup>C 可以使用 DMA 进行数据传输，可通过使能传输完成中断位产生中断，当利用 DMA 进行传输时，控

制寄存器 2 (I2C\_CTRL2) 的 DATAIEN 位需设为 0，以下说明软件利用 DMA 进行数据传输的操作流程。

#### DMA 发送

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

#### DMA 接收

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

### 11.4.4 SMBus

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

#### SMBus 和 I<sup>2</sup>C 的差异

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

#### SMBus 使用流程

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

SMBMODE=1: SMBus主机

SMBMODE=0: SMBus设备

3. 其他配置和I<sup>2</sup>C使用配置一样

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

#### SMBus 地址解析协议(ARP)

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



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

#### **SMBus 主机通知协议**

通过 **SMBus** 主机通知协议，可让从设备发送数据到主设备，例如从机可以通过此协议通知主机进行 **ARP**。关于 **SMBus** 主机通知协议更详细的信息请参考 **SMBus2.0** 协议。

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

#### **SMBus 提醒协议（SMBus Alert）**

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

操作流程如下：

##### **SMBus 主机**

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

##### **SMBus 从机**

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

#### **包错误校验(PEC)**

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

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

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

**PEC** 发送：

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

**PEC** 接收：

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

在接收模式下，当 **PEC** 校验失败时，将产生一个 **NACK**。

## 11.4.5 I<sup>2</sup>C中断请求

下表列出了所有的 I<sup>2</sup>C 中断请求。

中断事件	事件标志	使能位
已发送起始条件(主机)	STARTF	EVTIEN
地址已发送(主机)或地址匹配(从机)	ADDR7F	
10 位地址头已发送(主机)	ADDRHF	
数据传输完成	TDC	
收到停止条件(从机)	STOPF	
发送缓冲区空	TDBE	EVTIEN 和 DATAIEN
接收缓冲区非空	RDBF	
SMBus 提醒	ALERTF	ERRIEN
超时错误	TMOUT	
PEC 错误	PECERR	
过载/欠载	OUF	
应答失败	ACKFAIL	
仲裁丢失	ARLOST	
总线错误	BUSERR	

## 11.4.6 I<sup>2</sup>C调试模式

当微控制器进入调试模式 (Cortex™-M4F 核心处于停止状态) 时, 根据 DEBUG 模块中的 I2Cx\_SMBUS\_TIMEOUT 配置位, SMBUS 超时控制或者继续正常工作或者可以停止。

## 11.5 I<sup>2</sup>C寄存器描述

必须以字(32 位)的方式操作这些外设寄存器。

表 11-1 I<sup>2</sup>C寄存器地址映像和复位值

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

## 11.5.1 控制寄存器1(I2C\_CTRL1)

域	简称	复位值	类型	功能
位 15	RESET	0x0	rw	I <sup>2</sup> C 外设复位 (I <sup>2</sup> C peripheral reset) 0: 不复位; 1: 复位。 注: 该位可以用于 BUSYF 位为'1', 在总线上又没有检测到停止条件时。
位 14	保留	0x0	resd	保持默认值。
位 13	SMBALERT	0x0	rw	SMBus 提醒引脚设置 (SMBus alert pin set) 软件可以使其置 1 或清为 0; 当 I2CEN=0 时, 由硬件清除。 0: 置高; 1: 置低。
位 12	PECTEN	0x0	rw	请求 PEC 传输使能 (Request PEC transmission enable) 软件可以使其置 1 或清为 0; 当传送 PECTEN 后, 开始或结束信号时, 由硬件清除。 0: 停止传输; 1: 启动传输。
位 11	MACKCTRL	0x0	rw	主机接收模式应答控制 (Master receiving mode acknowledge control) 0: ACKEN 位效果作用于当前传字节; 1: ACKEN 位效果作用于第二个传输字节。 该位只在主机接收两个字节模式下使用, 目的是为了让主机及时的回 ACK。
位 10	ACKEN	0x0	rw	应答使能(Acknowledge enable) 软件可以使其置 1 或清为 0; 0: 关闭, 不发送应答; 1: 开启。
位 9	GENSTOP	0x0	rw	产生停止条件 (Generate stop condition) 软件可以使其置 1 或清为 0; 或当检测到结束信号时, 由硬件清除; 当检测到超时错误时, 硬件将其置位。 0: 未产生; 1: 产生。 如果在从模式下当设置了此位, 从机将释放 SCL 和 SDA 总线。
位 8	GENSTART	0x0	rw	产生起始条件 (Generate start condition) 软件可以使其置 1 或清为 0; 或当起始条件发出后, 由硬件清除。 0: 未产生; 1: 产生。
位 7	STRETCH	0x0	rw	时钟延展模式 (Clock stretching mode) 0: 开启; 1: 关闭。
位 6	GCAEN	0x0	rw	广播地址使能 (General call address enable) 0: 开启; 1: 关闭。
位 5	PECEN	0x0	rw	PEC 计算使能 (PEC calculation enable) 0: 关闭; 1: 开启。
位 4	ARPEN	0x0	rw	SMBus ARP 协议使能 (SMBus address resolution protocol enable) 0: 关闭; 1: 开启。 SMBus 主机: 响应主机地址 0001000x; SMBus 设备: 响应设备默认地址 0001100x。
位 3	SMBMODE	0x0	rw	SMBus 设备模式 (SMBus device mode) 0: SMBus 设备; 1: SMBus 主机。

位 2	保留	0x0	resd	硬件强制为 0。
位 1	PERMODE	0x0	rw	I <sup>2</sup> C 外设模式 (I <sup>2</sup> C peripheral mode) 0: I <sup>2</sup> C 模式; 1: SMBus 模式。
位 0	I2CEN	0x0	rw	I <sup>2</sup> C 外设使能 (I <sup>2</sup> C peripheral enable) 0: 关闭; 1: 开启。 在通讯结束后发生 I2CEN=0, 所有的位被清除。 在主模式下, 通讯结束之前, 绝不能清除该位。

注意: 当 GENSTART、GENSTOP 或 PECTEN 设置后, 软件应该在相应位被硬件清零后写 I2C\_CTRL1, 否则有可能产生第二次 GENSTART、GENSTOP 或 PECTEN 请求。

## 11.5.2 控制寄存器2(I2C\_CTRL2)

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

### 11.5.3 自身地址寄存器1(I2C\_OADDR1)

域	简称	复位值	类型	功能
位 15	ADDR1MODE	0x0	rw	地址模式 (Address mode) 0: 7 位地址; 1: 10 位地址。
位 14: 10	保留	0x00	resd	保持默认值。
位 9: 0	ADDR1	0x000	rw	本机地址 1 (Own address) 当在 7 位地址模式时 BIT0 以及 BIT[9: 8]不关心。

### 11.5.4 自身地址寄存器2(I2C\_OADDR2)

域	简称	复位值	类型	功能
位 15: 8	保留	0x00	resd	保持默认值。
位 7: 1	ADDR2	0x00	rw	本机地址 2 (Own address 2) 7 位地址。
位 0	ADDR2EN	0x0	rw	本机地址 2 使能 (Own address 2 enable) 0: 在 7 位地址模式下, 只有 OADDR1 被识别; 1: 在 7 位地址模式下, OADDR1 和 OADDR2 都被识别。

### 11.5.5 数据寄存器(I2C\_DT)

域	简称	复位值	类型	功能
位 15: 8	保留	0x00	resd	保持默认值。
位 7: 0	DT[7: 0]	0x00	rw	用于存放接收到或待发送的数据 发送器模式: 当写一个字节至 DT 寄存器时, 自动启动数据传输。一旦传输开始(TDE=1), 如果能及时把下一个需传输的数据写入 DT 寄存器, I <sup>2</sup> C 模块将保持连续的数据流。 接收器模式: 接收到的字节被拷贝到 DT 寄存器 (RDNE=1)。在接收到下一个字(RDNE=1)之前读出数据寄存器, 即可实现连续的数据传送。 注: 如果在处理 ACK 脉冲时发生 ARLOST 事件, 接收到的字节不会被拷贝到数据寄存器里, 因此不能读到它。

### 11.5.6 状态寄存器1(I2C\_STS1)

域	简称	复位值	类型	功能
位 15	ALERTF	0x0	rw0c	SMBus 提醒标志 (SMBus alert flag) 在 SMBus 主机模式下: 0: 未收到; 1: 收到。 SMBus 从机: 指示设备默认地址接收状态 (0001100x) 0: 未收到; 1: 收到。 软件可以使其清为 0; 当 I2CEN=0 时, 由硬件清除。
位 14	TMOUT	0x0	rw0c	SMBus 超时标志 (SMBus timeout flag) 0: 无超时错误; 1: 超时。 软件可以使其清为 0; 当 I2CEN=0 时, 由硬件清除。 注: 这个功能仅在 SMBUS 模式下有效
位 13	保留	0x0	resd	保持默认值。
位 12	PECERR	0x0	rw0c	PEC 接收错误标志 (PEC receive error flag) 0: 正确; 1: 错误。 软件可以使其清为 0。
位 11	OUF	0x0	rw0c	溢出标志 (Overload / underload flag) 当传输方向为发送数据时: 0: 正常; 1: 欠载。 当传输方向为接收数据时:

				0: 正常; 1: 过载。 软件可以使其清为 0; 当 I2CEN=0 时, 由硬件清除。
位 10	ACKFAIL	0x0	rw0c	应答失败标志 (Acknowledge failure flag) 0: 正常; 1: 失败。 当没有返回应答时, 硬件将置该位为'1' 软件可以使其清为 0; 当 I2CEN=0 时, 由硬件清除。
位 9	ARLOST	0x0	rw0c	仲裁丢失标志 (Arbitration lost flag) 0: 正常; 1: 仲裁丢失。 软件可以使其清为 0; 当 I2CEN=0 时, 由硬件清除。 在 ARLOST 事件之后, I <sup>2</sup> C 接口自动切换回从模式。
位 8	BUSERR	0x0	rw0c	总线错误标志 (Bus error flag) 0: 正常; 1: 错误。 当接口检测到错误的起始或停止条件, 硬件将该位置'1'。 软件可以使其清为 0; 当 I2CEN=0 时, 由硬件清除。
位 7	TDBE	0x0	ro	发送缓冲器空标志 (Transmit data buffer empty flag) 0: 数据正在从数据寄存器 (DT) 发送到移位寄存器, 数据寄存器还装着数据; 1: 数据已经从数据寄存器 (DT) 发送到移位寄存器, 数据寄存器空。 当 DT 为空的时候, 该标志值起, 向 DT 写数据时, 此标志清除。 注: 在写入第 1 个要发送的数据后, 或设置了 TDC 时写入数据, 都不能清除 TDBE 位, 这是因为数据寄存器仍然为空。
位 6	RDBF	0x0	ro	接收数据缓冲器满标志 (Receive data buffer full flag) 0: 数据寄存器 (DT) 未接收到数据; 1: 数据寄存器 (DT) 接收到数据。 读取 DT 寄存器时, 此标志清除。 在发生 ARLOST 事件时, RDBF 不被置位。
位 5	保留	0x0	resd	保持默认值。
位 4	STOPF	0x0	ro	停止条件产生完成标志 (Stop condition generation complete flag) 0: 未产生; 1: 已产生。 当 ACKEN=1, 从设备在总线上检测到停止条件时, 硬件将该位置'1' 先读取 STS1 寄存器, 然后写 CTRL1 寄存器清除标志。
位 3	ADDRHF	0x0	ro	主机 9~8 位地址头匹配标志 (master 9~8 bit address header match flag) 0: 未匹配; 1: 已匹配。 在 10 位地址模式下, 当主设备已经将第一个字节发送出去时, 硬件将该位置'1' 软件读取 STS1 寄存器后, 对 CTRL1 寄存器的写操作将清除该位, 或当 PEN=0 时, 硬件清除该位 注: 收到一个 NACK 后, ADDR10F 位不被置位。
位 2	TDC	0x0	ro	数据传输完成标志 (Transmit data complete flag) 0: 未完成 (移位寄存器还有数据); 1: 已完成 (移位寄存器空闲) 读或写 DT 寄存器, 或者收到开始或结束信号自动清除。 当 STRETCH=0 接收时收到一个新字节(包括 ACK 脉冲)且数据寄存器还未被读取(RDBF=1) 发送时, 当一个新数据将被发送且数据寄存器还未被写入新的数据 (TDBE=1) 上述两种情况 TDC 位会置 1
位 1	ADDR7F	0x0	ro	0~7 位地址匹配标志 (0~7 bit address match flag)

				0: 未产生; 1: 地址在主机模式下被发送或从机模式下接收到匹配地址。 在软件读取 STS1 寄存器后, 对 STS2 寄存器的读操作将清除该位 注: 在收到 NACK 后, ADDR7F 位不会被置位。
位 0	STARTF	0x0	ro	起始条件产生完成标志 (Start condition generation complete flag) 0: 未产生; 1: 已产生。 先读取 STS1 寄存器, 然后写 DT 寄存器清除标志。

## 11.5.7 状态寄存器2(I2C\_STS2)

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



## 11.5.8 时钟控制寄存器(I2C\_CLKCTRL)

域	简称	复位值	类型	功能
位 15	SPEEDMODE	0x0	rw	速度模式选择 (Speed mode selection) 0: 标准模式 (最快 100 kHz); 1: 快速模式 (最快 400 kHz)。 在快速模式下, 当 I <sup>2</sup> C 时钟为 10MHz 整数倍时, 可以产生准确的 400kHz 时钟
位 14	DUTYMODE	0x0	rw	快速模式占空比 (Fast mode duty cycle) 0: 高电平与低电平比值为 1: 2; 1: 低电平与高电平比值为 9: 16。
位 13: 12	保留	0x0	resd	保持默认值。
位 11: 0	SPEED	0x000	rw	I <sup>2</sup> C 总线速度配置 (I <sup>2</sup> C bus speed config) 在标准模式下: 高电平 = SPEED x T <sub>I2C_CLK</sub> ; 低电平 = SPEED x T <sub>I2C_CLK</sub> ; 在快速模式下: DUTYMODE = 0: 高电平 = SPEED x T <sub>I2C_CLK</sub> x 1; 低电平 = SPEED x T <sub>I2C_CLK</sub> x 2; DUTYMODE = 1: 高电平 = SPEED x T <sub>I2C_CLK</sub> x 9; 低电平 = SPEED x T <sub>I2C_CLK</sub> x 16。 标准模式下最小值为 4, 快速模式下最小值为 1。 只有在关闭 I <sup>2</sup> C 时(I2CEN=0)才能设置 CLKCTRL 寄存器;

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

## 12 通用同步异步收发器（USART）

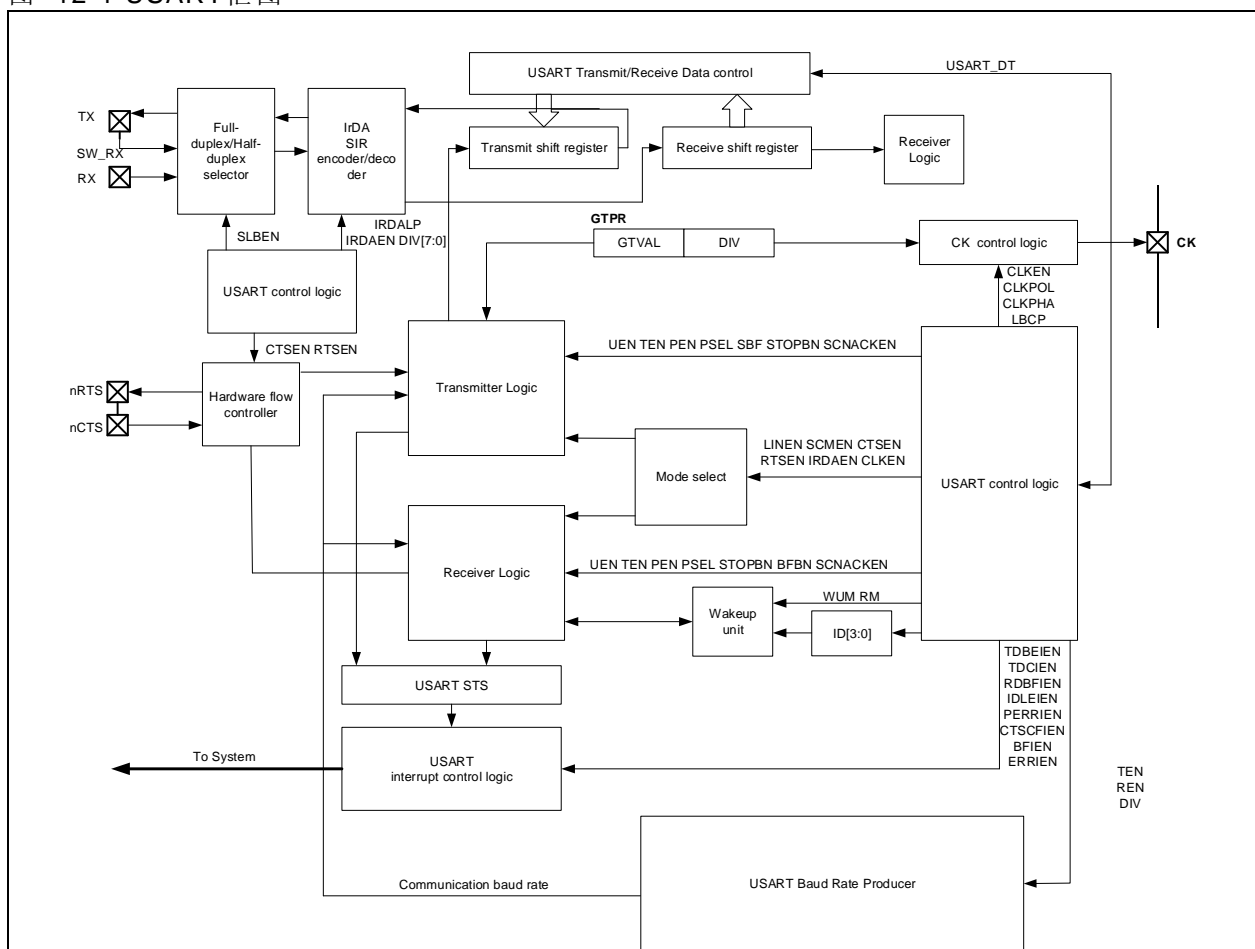
### 12.1 USART介绍

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

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

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

图 12-1 USART框图



USART 主要特性如下所列：

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

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

## 12.2 全双工半双工选择器简述和配置流程

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

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

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

## 12.3 模式选择器简述和配置流程

### 12.3.1 模式选择器简述

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

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

### 12.3.2 模式选择器配置方法

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

1. LIN模式：LINEN位置1，CLKEN位置0，STOPBN[1: 0]位置0，SCMEN位置0，SLHDEN位置0，IRDAEN位置0，DBN位置0。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，可以使用SBF位置1发送13位低电平 LIN同步间隔帧。
2. 智能卡模式：SCMEN位置1，LINEN位置0，SLHDEN位置0，IRDAEN位置0，CLKEN位置1，DBN位置1，PEN位置1，STOPBN[1: 0]=11。可以选择配置CLKPOL位和CLKPHA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。可以通过配置SCGT[7: 0]位选择保护时间。可以通过配置SCNACKEN位选择是否在校验出错时发送NACK。
3. 红外模式：IRDAEN位置1，CLKEN位置0，STOPBN[1: 0]位置0，SCMEN位置0，SLHDEN位置0。可以选择IRDALP位置1以开启红外低功耗模式，并配合ISDIV[7: 0]配置想要产生的低功耗频率。
4. 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。
5. 静默模式：RM位置1进入静默模式，根据WUM位置1和置0，可以分别通过ID匹配和空闲总线从静默模式中唤醒，其中ID号ID[3: 0]可编程配置，当选择ID匹配时，数据位的MSB为1表示当前数据是ID，4个LSB表示ID值。
6. 同步模式：CLKEN位置1开启同步模式并使能时钟管脚输出，通过配置CLKPOL位置1或0可以选择空闲状态下CK管脚上的电平为高或低，通过配置CLKPHA位置1或0可以选择在时钟的第二个或第一个边沿开始采样数据，通过配置LBCP位置1或0可以选择最后一位数据是否输出时钟，通过配置ISDIV[4: 0]可以选择想要输出的时钟频率。

## 12.4 USART帧格式简述和配置流程

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

USART 一笔空闲帧的长度等于当前配置下数据帧的长度，但所有位都为 1。

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

通过 DBN 位配置 8 位 (DBN=0) 或 9 位 (DBN=1) 数据位。

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。

## 12.5 DMA传输简述和配置流程

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

### 12.5.1 DMA发送配置流程

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

### 12.5.2 DMA接收配置流程

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

## 12.6 波特率发生器简述及配置流程

### 12.6.1 波特率发生器简述

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

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

### 12.6.2 波特率发生器配置方法

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

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

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

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

2. 关闭 USART 接收器或发送器会使内部计数器复位，波特率发生中断。

表 12-1 设置波特率时的误差计算

波特率		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	937.5	0%	2.4	1875	0%
2	9.6	9.6	234.375	0%	9.6	468.75	0%
3	19.2	19.2	117.1875	0%	19.2	234.375	0%
4	57.6	57.6	39.0625	0%	57.6	78.125	0%
5	115.2	115.384	19.5	0.15%	115.2	39.0625	0%
6	230.4	230.769	9.75	0.16%	230.769	19.5	0.16%
7	460.8	461.538	4.875	0.16%	461.538	9.75	0.16%
8	921.6	923.076	2.4375	0.16%	923.076	4.875	0.16%
9	2250	2250	1	0%	2250	2	0%
10	4500	不可能	不可能	不可能	4500	1	0%
11	6250	不可能	不可能	不可能	不可能	不可能	不可能
12	7500	不可能	不可能	不可能	不可能	不可能	不可能

波特率		fPCLK=100MHz			fPCLK=120MHz		
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	2.40004	2604.125	0%	2.4	3125	0%
2	9.6	9.60061	651	0%	9.6	781.25	0%
3	19.2	19.2012	325.5	0%	19.2	390.625	0%
4	57.6	57.6037	108.5	0%	57.61	130.1875	0%



5	115.2	115.207	54.25	0%	115.16	65.125	0%
6	230.4	230.415	27.125	0.01%	230.33	32.5625	0%
7	460.8	460.829	13.5625	0.01%	461.54	16.25	0.16%
8	921.6	925.926	6.75	0.47%	923.08	8.125	0.16%
9	2250	2272.73	2.75	1%	2264.15	3.3125	0.63%
10	4500	4545.45	1.375	1%	4444.44	1.6875	1.23%
11	6250	6250	1	0%	6315.79	1.1875	1.05%
12	7500	不可能	不可能	不可能	7500	1	0%

## 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 TDBEIEEN 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 管脚输入。

注意： 1. 在数据传输期间不能复位 **TEN** 位，否则将破坏 TX 脚上的数据。  
2. **TEN** 位被激活后，USART 将自动发送一个空闲帧。

### 12.7.2 发送器配置流程

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

## 12.8 接收器简述和配置流程

### 12.8.1 接收器简述

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

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

如若选择了硬件流控制，USART 接收器将控制信号将从 RTS 管脚输出。

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

## 12.8.2 接收器配置流程

配置步骤：

1. USART使能：UEN位置1。
2. 全双工半双工配置：具体参见全双工半双工选择器配置部分。
3. 模式配置：具体参见模式选择器配置部分。
4. 帧格式配置：具体参见帧格式配置部分。
5. 中断配置：具体参见中断发生器配置部分。
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。
8. 接收器使能：REN位置1。

当一个字符被接收到时：

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

当一个间隔帧被接收到时：

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

当一个空闲帧被接收到时：

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

当一个帧错误产生时：

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

当一个溢出错误产生时：

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

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

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

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

## 12.8.3 起始侦测和噪声检测

USART 接收器在 REN 位置位后便开始侦测起始位，USART 接收器通过过采样技术，在第 3、5、7、8、9、10 位共 6 个点进行数据采样，以此侦测有效起始位以及识别噪声，具体的噪声和有效起始位的判别方式可以参见下方表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	1	无效
任意值	111/110/101/011	1	无效

注意：如果在第 3、5、7、8、9、10 位的采样值满足不了上表任意一种组合，则 USART 接收器认为没有接受到正确的起始位，将退出起始位侦测并回到空闲状态等待下降沿。

USART 接收器具备噪声检测功能，在非同步模式时，使用过采样技术，在第 7、8、9 采样点，根据不同的采样值，区别有效输入数据和噪声，并恢复数据和置起噪声错误标志位 NERR。具体的采样方法以及噪声和有效数据的判别方式可以参见下表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

当 USART 接收器在数据帧中检测到噪声时：

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

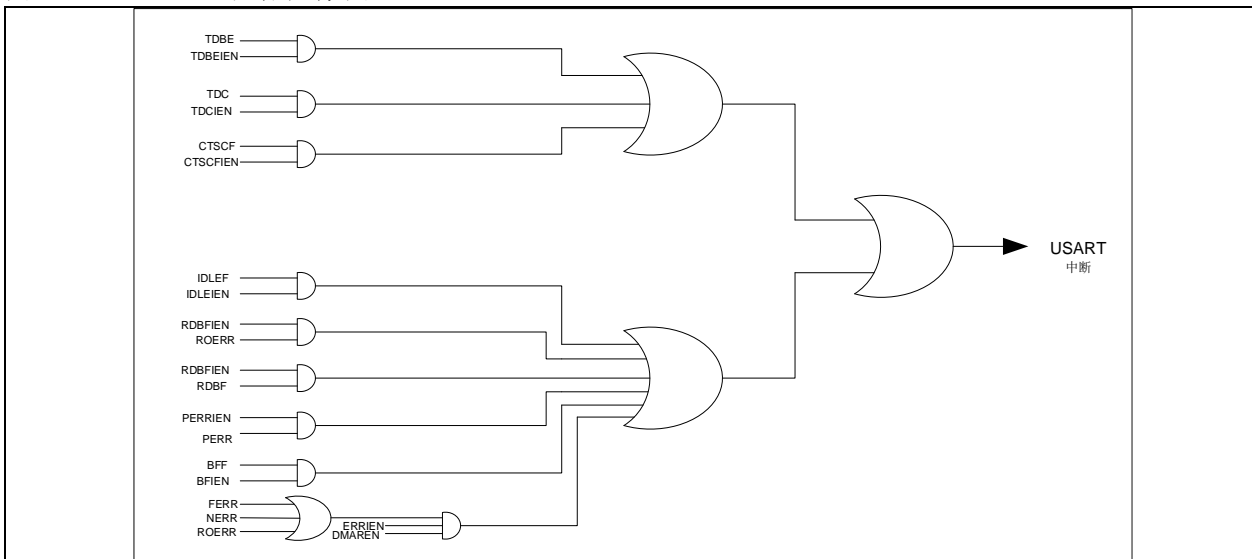
## 12.9 中断

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 12-4 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	TDBE1EN
CTS 标志	CTSCF	CTSCF1EN
发送完成	TDC	TDC1EN
接收数据就绪可读	RDBF	RDBF1EN
检测到数据溢出	ROERR	
检测到空闲线路	IDLEF	IDLE1EN
奇偶检验错	PERR	PERR1EN
断开标志	BFF	BFF1EN
噪声标志，多缓冲通信中的溢出错误和帧错误	NERR 或 ROERR 或 FERR	ERR1EN <sup>(1)</sup>

图 12-2 USART中断映像图



## 12.10 I/O管脚控制

USART 通过五个接口外部设备进行通信，管脚定义如下：

RX：串行数据输入端。

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

CK：发送器时钟输出。输出的 CLK 相位和极性以及频率均可编程配置。

CTS：发送器输入端，硬件流控制模式发送使能信号。

RTS：接收器输出端，硬件流控制模式发送请求信号。

## 12.11 USART寄存器描述

必须以字(32 位)的方式操作这些外设寄存器。

表 12-5 USART寄存器映像和复位值

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

## 12.11.1 状态寄存器 (USART\_STS)

域	简称	复位值	类型	功能
位 31: 10	保留	0x000000	resd	硬件强制为 0。
位 9	CTSCF	0x0	rw0c	CTS 变化标志 (CTS change flag) 当 CTS 线发送变化时, 该位被硬件置起, 由软件将其清零。 0: 无; 1: 有。
位 8	BFF	0x0	rw0c	间隔帧标志 (break frame flag) 当检测到间隔帧时, 该位被硬件置起, 由软件将其清零。 0: 无; 1: 有。
位 7	TDBE	0x1	ro	发送缓冲器空 (Transmit data buffer empty) 当发送缓冲器为空, 可以再次写入数据时, 该位被硬件置起。对 USART_DT 的写操作, 将清零该位。 0: 非空; 1: 空。
位 6	TDC	0x1	rw0c	发送数据完成 (Transmit data complete) 当发送数据完成, 该位被硬件置起, 由软件将其清零 (方式 1: 先读 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 RDNE 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	噪声错误 (Noise error) 接收到的数据有杂讯时, 该位被硬件置起, 由软件将其清零 (先读 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。

## 12.11.2 数据寄存器（USART\_DT）

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

## 12.11.3 波特比率寄存器（USART\_BAUDR）

注意：如果 TE 或 RE 被分别禁止，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。

## 12.11.4 控制寄存器1（USART\_CTRL1）

域	简称	复位值	类型	功能
位 31: 14	保留位	0x000000	resd	硬件强制为 0。
位 13	UEN	0x0	rw	USART 使能（USART enable） 0: 关闭； 1: 开启。
位 12	DBN	0x0	rw	数据位个数（Data bit num） 该位定义了数据位的个数。 0: 8 位； 1: 9 位。
位 11	WUM	0x0	rw	唤醒方式（Wake up mode） 该位定义静默状态下被唤醒的方式。 0: 空闲帧唤醒； 1: ID 匹配唤醒。
位 10	PEN	0x0	rw	奇偶校验使能（Parity enable） 该位定义使能硬件奇偶校验（对于发送来说就是校验位的产生；对于接收来说就是校验位的检测）。当使能了该位，硬件将发送数据的最高位替换成校验位；对接收到的数据检查其校验位是否正确。 0: 关闭； 1: 开启。
位 9	PSEL	0x0	rw	奇偶校验选择（Parity selection） 该位定义是采用奇校验还是偶校验。 0: 偶校验； 1: 奇校验。
位 8	PERRIEN	0x0	rw	PERR 中断使能（PERR interrupt enable） 0: 关闭； 1: 开启。
位 7	TDBEIEN	0x0	rw	发送数据缓冲器空中断使能（TDBE interrupt enable） 0: 关闭； 1: 开启。
位 6	TDCIEN	0x0	rw	发送数据完成中断使能（TDC interrupt enable） 0: 关闭； 1: 开启。
位 5	RDBFIEN	0x0	rw	接收数据缓冲器满中断使能（RDNE 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；

位 2	REN	0x0	rw	1: 开启。 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。

## 12.11.5 控制寄存器2 (USART\_CTRL2)

域	简称	复位值	类型	功能
位 31: 15	保留位	0x00000	resd	硬件强制为 0。
位 14	LINEN	0x0	rw	LIN 模式使能 (LIN mode enable) 0: 关闭; 1: 开启。
位 13: 12	STOPBN	0x0	rw	停止位个数 (STOP bit num) 这 2 位用来设置停止位的个数 00: 1 位; 01: 0.5 位; 10: 2 位; 11: 1.5 位;
位 11	CLKEN	0x0	rw	时钟使能 (Clock enable) 该位用来使能同步模式或智能卡模式的时钟管脚。 0: 关闭; 1: 开启。
位 10	CLKPOL	0x0	rw	时钟极性 (Clock polarity) 在同步模式或智能卡模式下, 可以用该位选择时钟管脚上总线空闲时时钟输出的极性。 0: 低电平; 1: 高电平。
位 9	CLKPHA	0x0	rw	时钟相位 (Clock phase) 在同步模式或智能卡模式下, 可以用该位选择时钟管脚上时钟输出的相位。 0: 第一个边沿进行数据捕获; 1: 第二个边沿进行数据捕获。
位 8	LBCP	0x0	rw	最后一位时钟脉冲 (Last bit clock pulse) 在同步模式下, 使用该位来控制是否在时钟管脚上输出数据的最后一位对应的时钟脉冲 0: 不输出; 1: 输出。
位 7	保留位	0x0	resd	保持默认值。
位 6	BFIEN	0x0	rw	间隔帧中断使能 (break frame interrupt enable) 0: 关闭; 1: 开启。
位 5	BFBN	0x0	rw	间隔帧位数 (break frame bit num) 该位用来选择是 11 位还是 10 位的间隔帧。 0: 10 位; 1: 11 位。
位 4	保留位	0x0	resd	保持默认值。
位 3: 0	ID	0x0	rw	USART 的 ID 号 (USART identification)

可配置的 USART 的 ID 号。

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

### 12.11.6 控制寄存器3（USART\_CTRL3）

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

## 12.11.7 保护时间和预分频寄存器（USART\_GDIV）

域	简称	复位值	类型	功能
位 31: 16	保留位	0x0000	resd	硬件强制为 0。
位 15: 8	SCGT	0x00	rw	<p>智能卡保护时间值（Smart card guard time）</p> <p>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。</p>
位 7: 0	ISDIV	0x00	rw	<p>红外或者智能卡分频系数（IrDA/smartcard division）</p> <p>红外（IrDA）模式：</p> <p>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；</p> <p>00000000：保留 - 不要写入该值；</p> <p>00000001：1 分频；</p> <p>00000010：2 分频；</p> <p>.....</p> <p>智能卡模式：</p> <p>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：</p> <p>00000：保留 - 不要写入该值；</p> <p>00001：2 分频；</p> <p>00010：4 分频；</p> <p>00011：6 分频；</p> <p>.....</p>



## 13 串行外设接口（SPI）

### 13.1 串行外设接口（SPI）简介

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

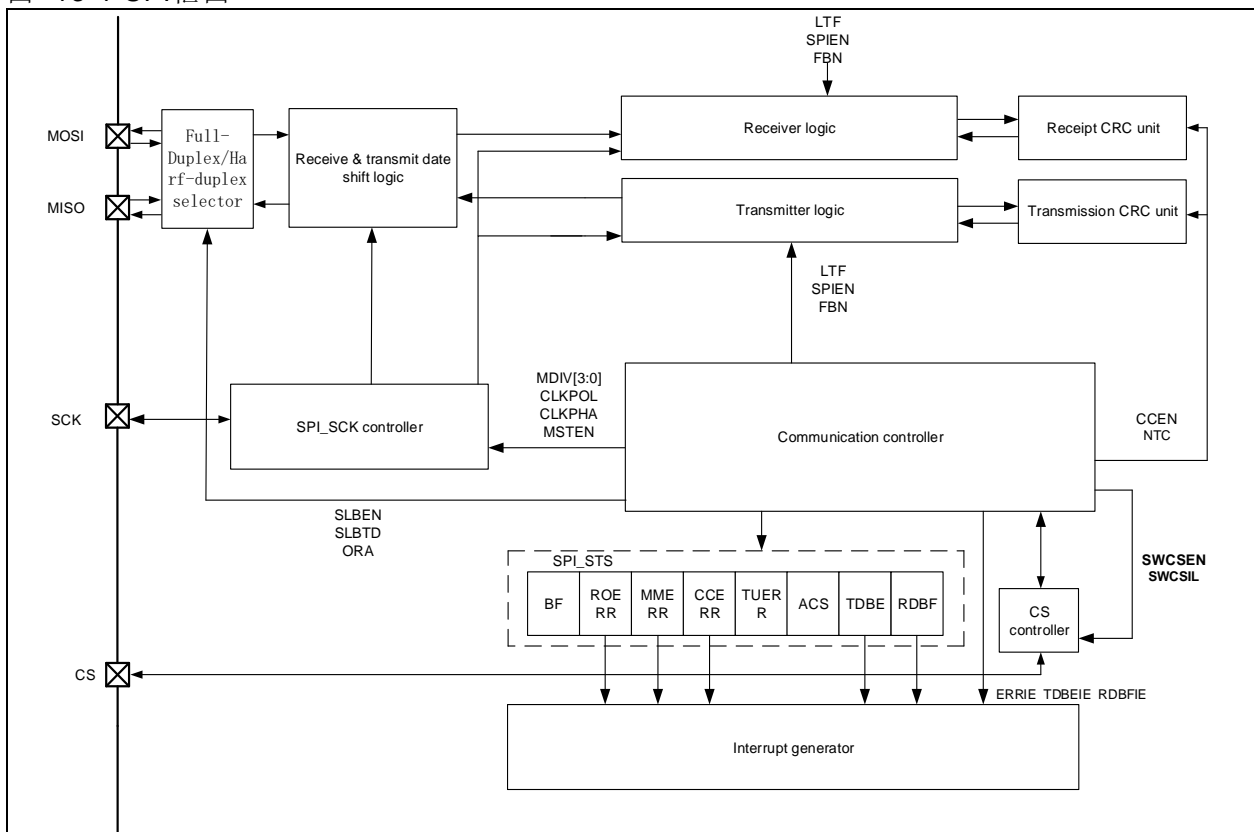
### 13.2 SPI功能描述

#### 13.2.1 SPI简述

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

SPI 的架构框图见下图：

图 13-1 SPI框图



SPI 接口作为 SPI 使用时主要特征如下：

- 可编程配置的全双工或半双工通信
  - 全双工同步通信（可以选择全双工只收以此释放用于发送的 IO）
  - 半双工同步通信（可以根据软件编程配置选择传输方向：发送或接收）
- 可编程配置主/从模式
- 可编程配置的 CS 信号处理方式
  - 硬件处理 CS
  - 软件处理 CS
- 可编程配置的 8 位或 16 位帧位数
- 可编程配置的通信频率以及分频系数（主从最大通信频率值为 48M，最大分频系数为  $f_{CLK}/2$ ）
- 可编程配置的时钟极性和相位
- 可编程配置的数据传输顺序(先发 MSB/LSB)
- 可编程配置的错误中断标志（接收器溢出错误，主模式错误，CRC 校验错误）

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

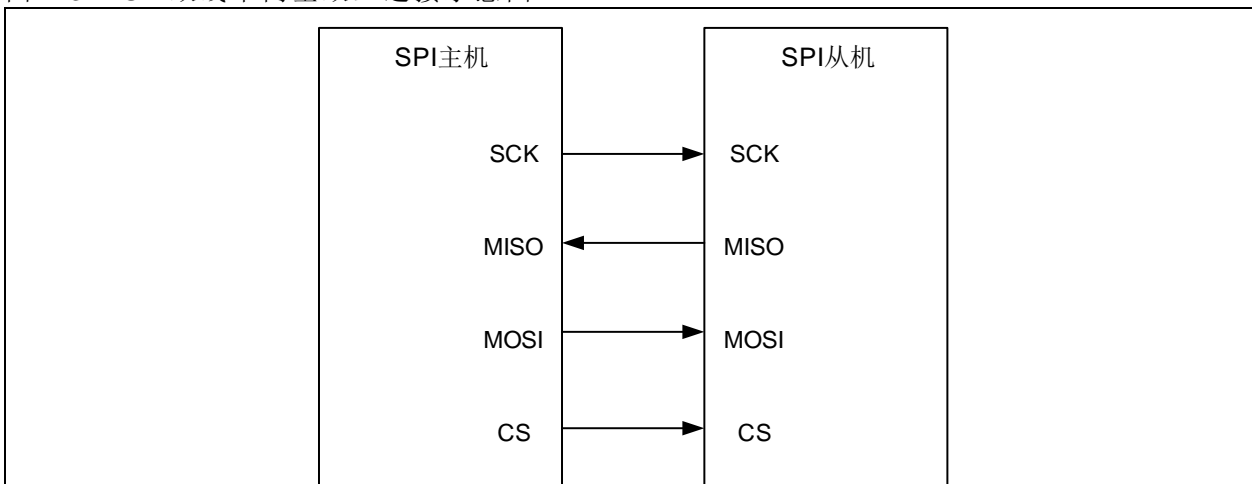
### 13.2.2 全双工半双工选择器简述和配置流程

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

**双线单向全双工模式配置方式以及 SPI IO 连接方式如下：**

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

图 13-2 SPI 双线单向全双工连接示意图



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

**单线单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 工作在单线单向只收模式，此时 SPI 只能作为数据接收方，无法发送数据。作为主机时使用 MISO 接收数据，MOSI 管脚所映射的 IO 释放。作为从机时使用 MOSI 接收数据，MISO 管脚所映射的 IO 释放。

图 13-3 SPI 作主机单线单向只收连接示意图

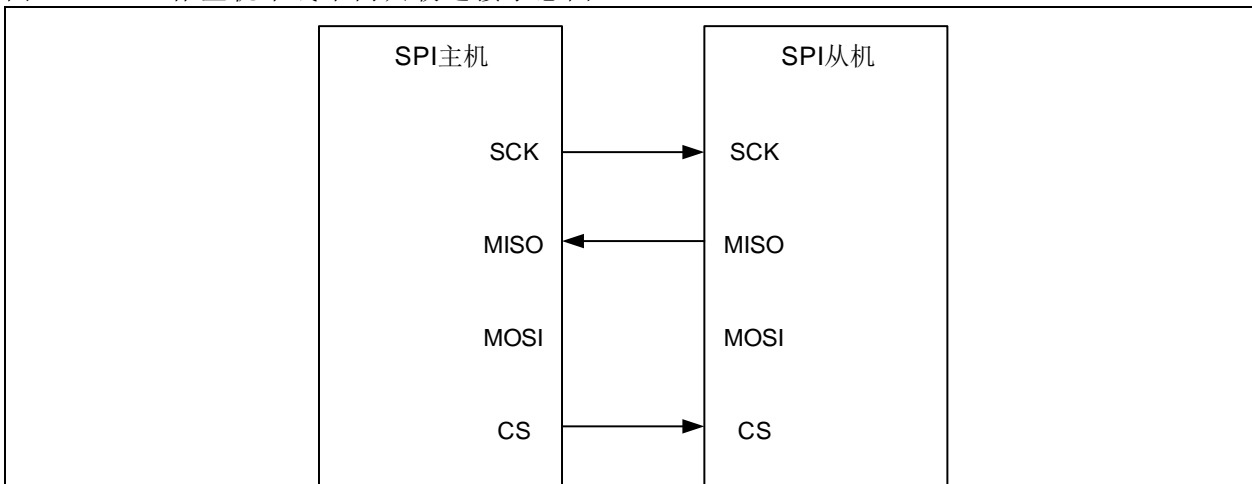
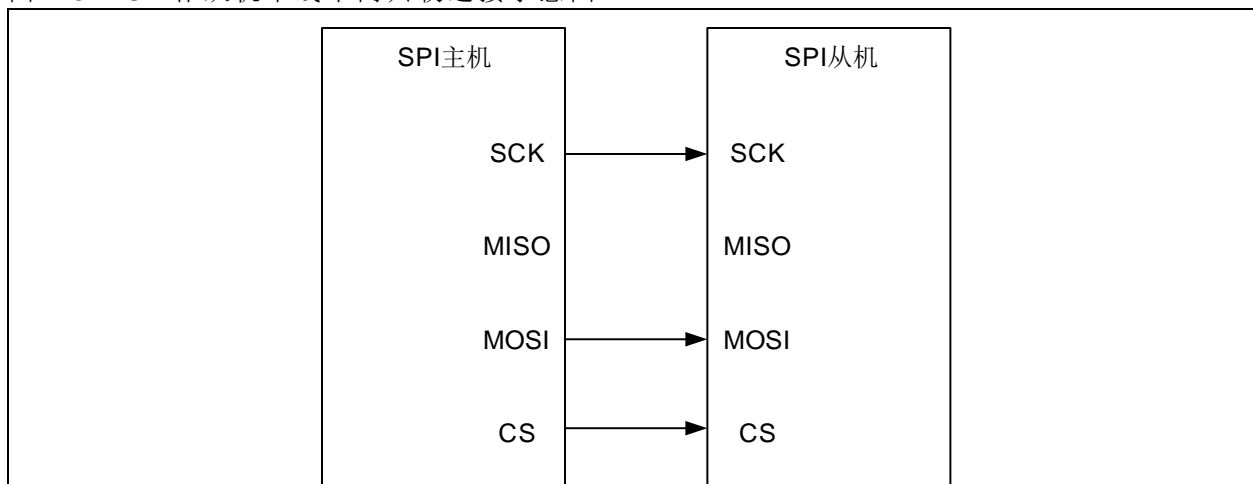


图 13-4 SPI作从机单线单向只收连接示意图



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

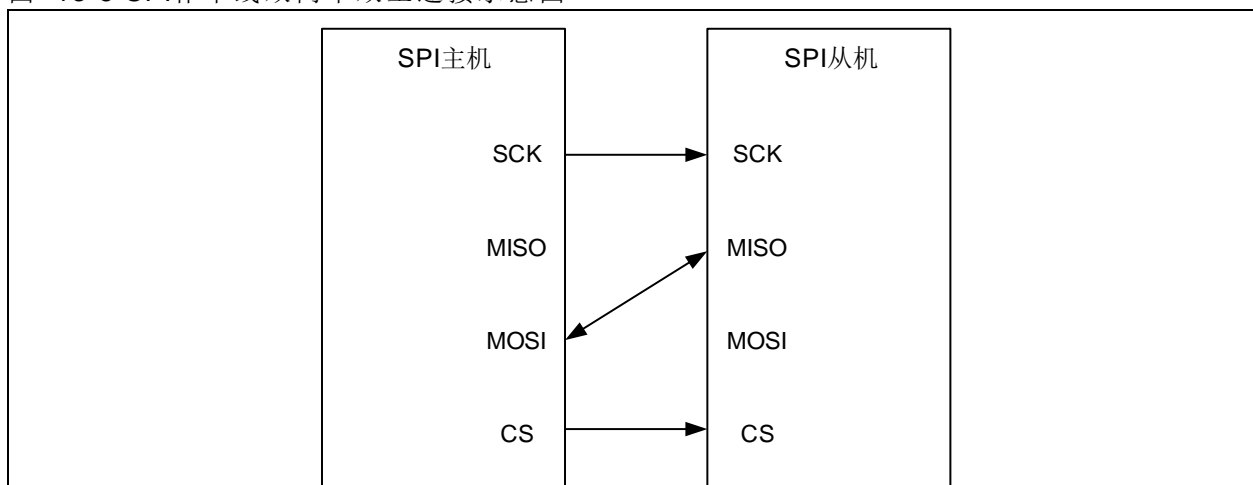
SPI 作从机时，在此模式下，关闭 SPI 无需判断任何标志，但是在进入省电模式(或关闭 SPI 系统时钟)之前需要等待 BF=0。

**单线双向半双工模式配置方式以及 SPI IO 连接方式如下：**

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

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

图 13-5 SPI作单线双向半双工连接示意图



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

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

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

### 13.2.3 CS控制器简述和配置流程

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

**硬件 CS 配置流程：**

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

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

#### 软件 CS 配置流程：

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

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

### 13.2.4 SPI\_SCK 控制器简述和配置流程

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

#### SPI\_SCK 控制器配置流程：

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

### 13.2.5 CRC 简述和配置流程

SPI 接口内部具有独立的发送和接收 CRC 计算单元，通过软件编程配置，SPI 接口在作为 SPI 使用时，可以同时在使用 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\_RXCRC 寄存器 (SPI\_RCR) 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 SPICRC 多项式寄存器 (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\_RXCRC 寄存器 (SPI\_RCR) 以及 SPITxCRC 寄存器 (SPI\_TCR)。
- 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 SPICRC 多项式寄存器 (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 SPITxCRC 寄存器 (SPI\_TCR) 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 SPICRC 多项式寄存器 (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\_RXCRC 寄存器 (SPI\_RCR) 中的值作比较，若校验出错，会置起 CCERR 标志，若使能了 ERRIE 位，则产生错误中断。

#### CPU 发送模式：

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

置起 NTC 位。

#### CPU 接收模式

在双线单向全双工模式下，按照 CPU 发送模式操作 NTC 位，CPU 接收模式的 CRC 计算和校验会自动完成，在单线单向只接收以及单线双向只接收模式下，相较于 DMA 接收需要软件在接收到倒数第二笔数据之后，接收到最后一笔数据之前置起 NTC 位。

### 13.2.6 DMA传输简述和配置流程

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

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

#### DMA 发送配置流程：

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

#### DMA接收配置流程：

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

### 13.2.7 发送器简述和配置流程

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

用户可以选择 DMA 或 CPU 来控制数据的写入，若选择 DMA 传输，详细配置请参见 DMA 传输章节，若



选择 CPU 传输，则用户需要判断 TDBE 位，该位复位值为 1，代表 SPI\_DT 为空，若 TDBEIE 置位，则产生中断，数据写入后，TDBE 拉低，直到数据被同步到发送器中的数据缓冲器后，TDBE 再次被拉起，即用户只可以在 TDBE 置位时写入待发送的数据。

发送器配置完成并使能 SPI 后，SPI 将进入数据发送状态，所以在此之前，应需要参考全双工半双工章节配置通信选用的是全双工或半双工等，并参考 CS 控制器章节配置选用的 CS 控制模式，还需要参考 SPI\_SCK 控制章节配置通信时钟，若使用了 CRC 以及 DMA，还需参考 CRC 以及 DMA 传输章节配置 CRC 以及 DMA，如下为推荐的发送器配置流程。

**发送器配置流程：**

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

### 13.2.8 接收器简述和配置流程

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

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

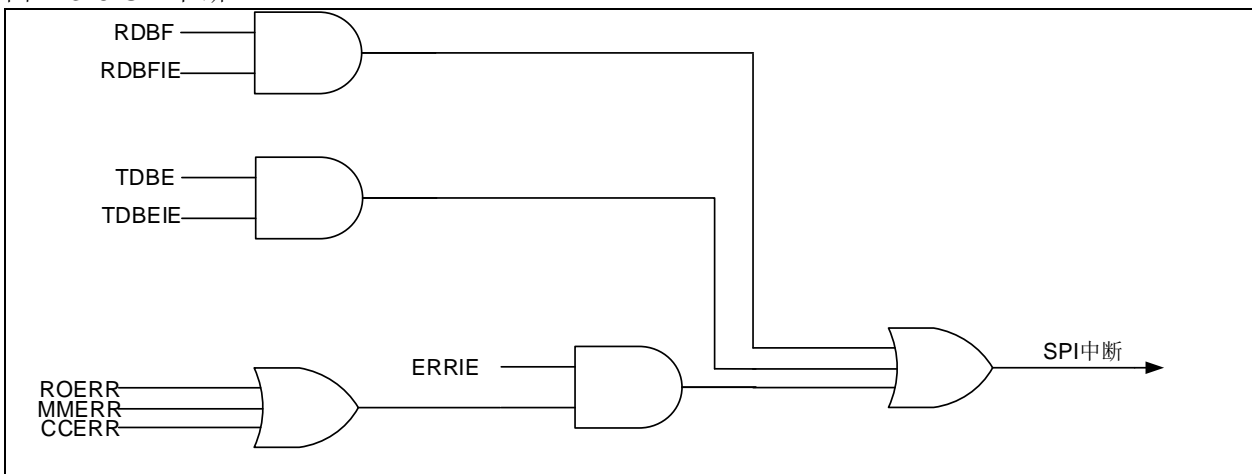
若在下一笔接收器接收到的数据准备压入 SPI 数据寄存器(SPI\_DT)时，之前接收到的数据仍未被读走，即 RDBF 仍为 1，则代表数据溢出，在此之前接收到的数据不会丢失，但之后的数据都将丢失，此时 ROERR 置起，若 ERRIE 置位，则产生错误中断，依次读 SPI 数据寄存器(SPI\_DT)和 SPI 状态寄存器(SPI\_STS)可将 ROERR 清除，如下为推荐的接收器配置流程。

**接收器配置流程：**

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

### 13.2.9 中断

图 13-6 SPI中断



### 13.2.10 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。

**警告：**由于 SPI1/I<sup>2</sup>S1、SPI3/I<sup>2</sup>S3 的部分管脚与 JTAG 管脚共享（SPIx\_CS/I<sup>2</sup>Sx\_WS 与 JTDI，SPIx\_SCK/I<sup>2</sup>Sx\_CK 与 JTDO），因此这些管脚不受 IO 控制器控制，他们（在每次复位后）被默认保留为 JTAG 用途。如果用户想把管脚配置给 SPIx/I<sup>2</sup>Sx，必须（在调试时）关闭 JTAG 并切换至 SWD 接口，或者（在标准应用时）同时关闭 JTAG 和 SWD 接口。

### 13.2.11 注意事项

CRC 接收完成后要软件读 DT 寄存器来读出 CRC 值。

## 13.3 I<sup>2</sup>S功能描述

### 13.3.1 I<sup>2</sup>S简述

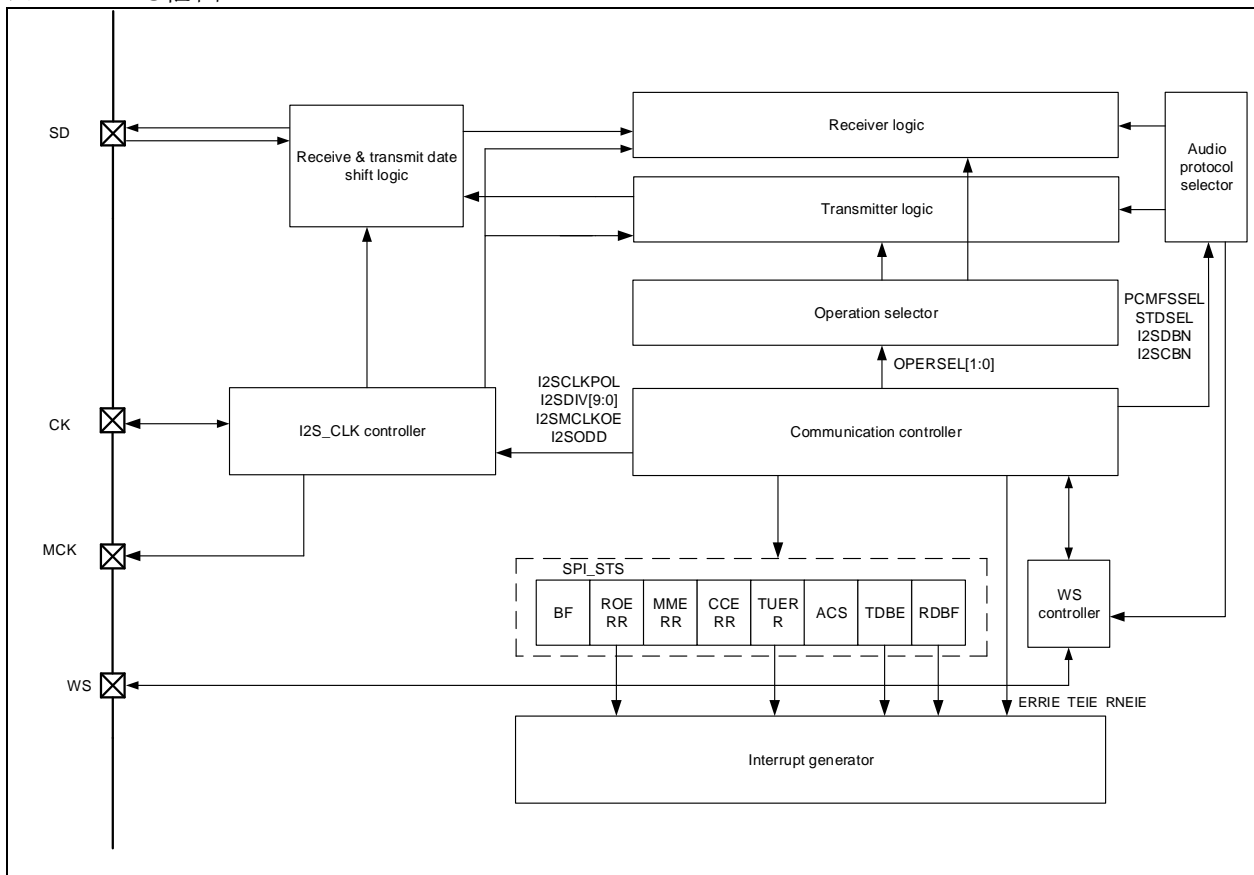
I<sup>2</sup>S 根据软件配置的不同，可以分别工作在主机接收，主机发送，从机接收，从机发送四种操作模式，并且可以分别支持包括飞利浦标准，高字节对齐标准，低字节对齐标准，PCM 标准在内的共四种音频标准，并同时支持 DMA 传输。

单个的 I<sup>2</sup>S 虽然只支持半双工，但结合额外例化的两个 I<sup>2</sup>S 模块（I<sup>2</sup>S2EXT，I<sup>2</sup>S3EXT），通过 I<sup>2</sup>S2 与 I<sup>2</sup>S2EXT 组合在一起可以使得 I<sup>2</sup>S2 支持全双工模式，或通过 I<sup>2</sup>S3 与 I<sup>2</sup>S3EXT 组合在一起使得 I<sup>2</sup>S3 支持全双工模式，具体配置和使用方式请参考 I<sup>2</sup>S 全双工章节。

I<sup>2</sup>S 的框图如下图所示：



图 13-7 I<sup>2</sup>S框图



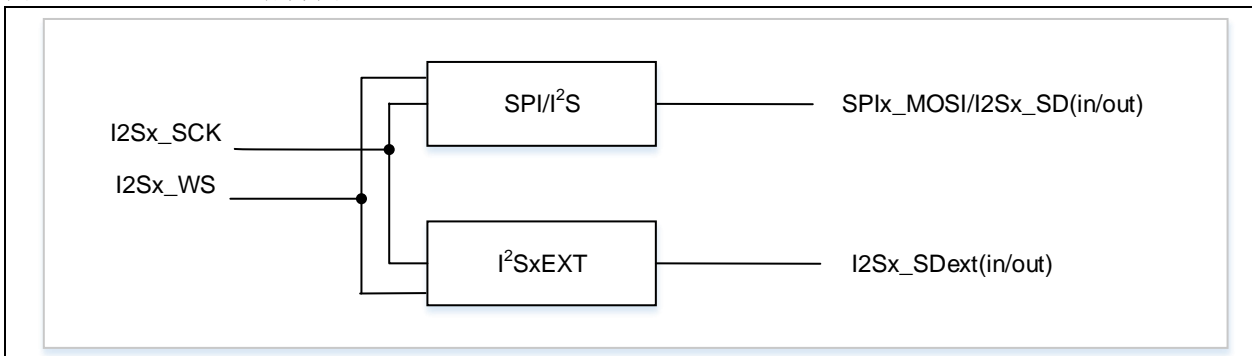
SPI 接口作为 I<sup>2</sup>S 使用时主要特征如下：

- 可编程配置的操作模式
  - 从设备发送
  - 从设备接收
  - 主设备发送
  - 主设备接收
- 可编程配置的时钟极性
- 可编程配置的时钟频率（8KHz 到 192KHz）
- 可编程配置的数据位数（16 位，24 位，32 位）
- 可编程配置的声道位数（16 位，32 位）
- 可编程配置的音频协议
  - I<sup>2</sup>S 飞利浦标准
  - 高字节对齐标准（左对齐）
  - 低字节对齐标准（右对齐）
  - PCM 标准（带长或短帧同步的通道帧）
- 支持 I<sup>2</sup>S 全双工
- 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 I<sup>2</sup>S 全双工

为了支持 I<sup>2</sup>S 全双工模式，额外例化了两个 I<sup>2</sup>S 模块（I<sup>2</sup>S2EXT，I<sup>2</sup>S3EXT）。I<sup>2</sup>S2 与 I<sup>2</sup>S2EXT 组合在一起支持全双工模式，I<sup>2</sup>S3 与 I<sup>2</sup>S3EXT 组合在一起支持全双工模式。

注意：I<sup>2</sup>S2EXT 和 I<sup>2</sup>S3EXT 只用于 I<sup>2</sup>S 全双工模式。

图 13-8 I<sup>2</sup>S全双工结构图

x可以是2或者3

I<sup>2</sup>Sx 可以作为主机

- 半双工模式下，只有 I<sup>2</sup>Sx 可以输出 SCK 和 WS。
- 全双工模式下，只有 I<sup>2</sup>Sx 可以输出 SCK 和 WS。

新增的 I<sup>2</sup>SxEXT 只能用 I<sup>2</sup>S 全双工模式下，I<sup>2</sup>SxEXT 只可以配置成从机模式。

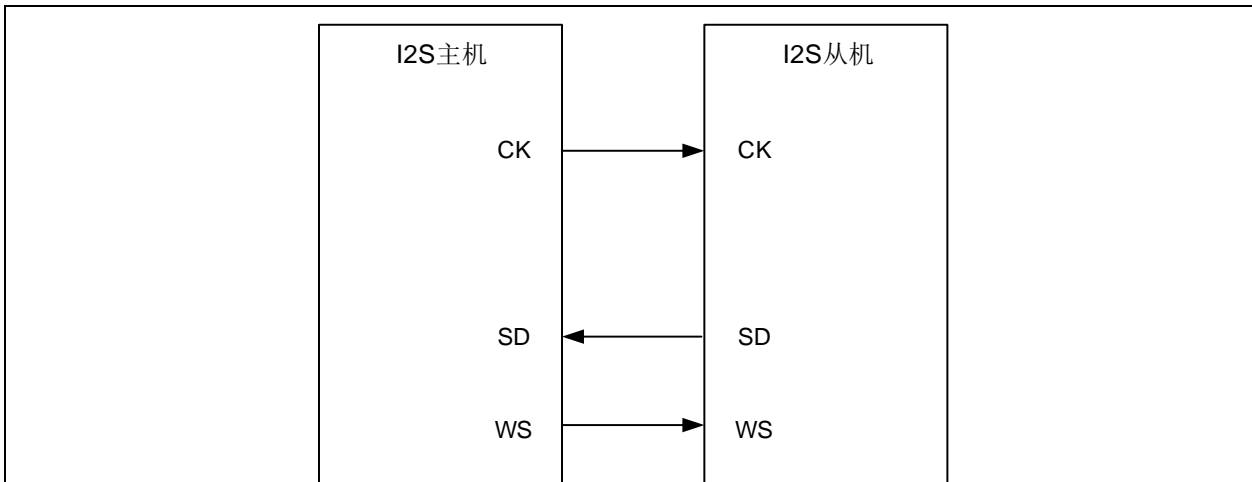
I<sup>2</sup>Sx 和 I<sup>2</sup>SxEXT 都可以做发送或者接收方。

### 13.3.3 操作模式选择器简述和配置流程

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

**从设备发送：**

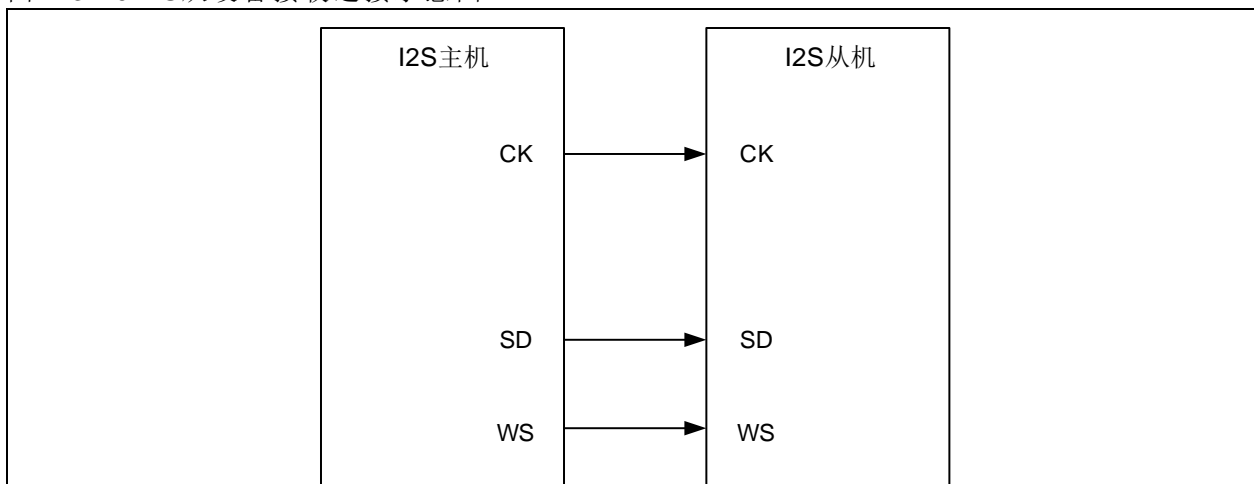
置位 I2SMSEL 位，配置 OPERSEL[1: 0]位为 00，I<sup>2</sup>S 将工作在从设备发送模式下

图 13-9 I<sup>2</sup>S从设备发送连接示意图

**从设备接收：**

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

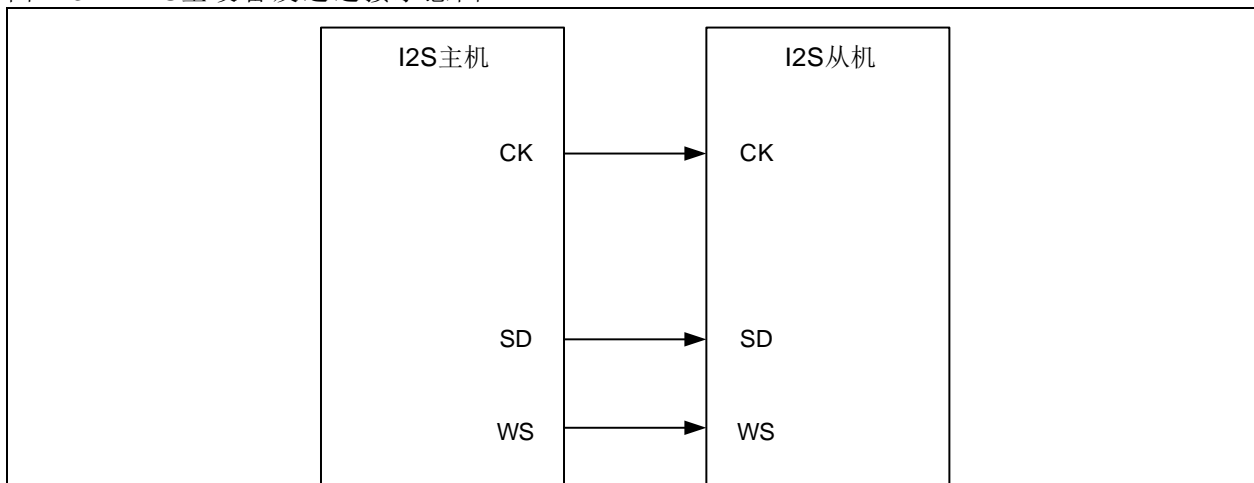
图 13-10 I<sup>2</sup>S从设备接收连接示意图



主设备发送:

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

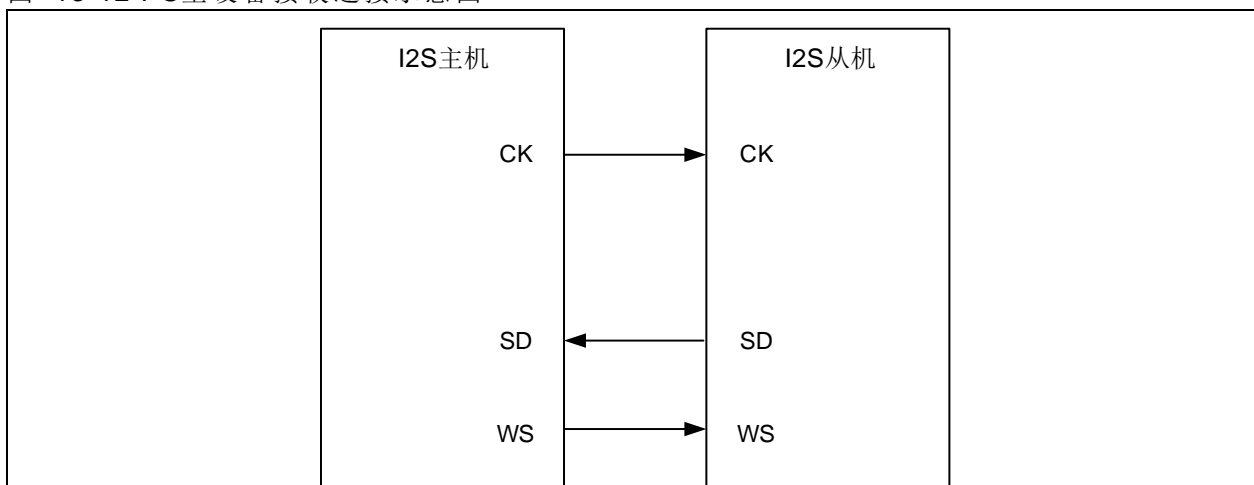
图 13-11 I<sup>2</sup>S主设备发送连接示意图



主设备接收:

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

图 13-12 I<sup>2</sup>S主设备接收连接示意图



## 13.3.4 音频协议选择器简述和配置流程

SPI接口作为I<sup>2</sup>S使用时支持多种音频协议, 用户可以通过软件编程控制音频协议选择器选择需要的音频协议, 数据位个数以及声道位个数同样由音频协议选择器控制, 用户同样可以通过软件编程配置的方式选

择需要的数据位个数以及声道位个数，同时，音频协议选择器会自动控制WS控制器，输出或检测符合协议要求的WS信号，具体的配置流程如下。

- 音频协议选择：配置 STDSEL 位选择需要的音频协议

STDSEL=00：飞利浦标准

STDSEL=01：高字节对齐标准（左对齐）

STDSEL=10：低字节对齐标准（右对齐）

STDSEL=11：PCM 标准

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

- 数据位个数选择：配置 I2SDBN 位选择需要的数据位个数

I2SDBN=00：16 位

I2SDBN =01：24 位

I2SDBN =10：32 位

- 声道位个数选择：配置 I2SCBN 位选择需要的声道位个数

I2SCBN =0：16 位

I2SCBN =1：32 位

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

- 飞利浦标准或 PCM 标准或高字节或低字节标准，16 位数据，16 位声道  
数据位数和声道位数一致，每个声道只需读写一次 SPI 数据寄存器（SPI\_DT），DMA 传输个数为 1。
- 飞利浦标准或 PCM 标准或高字节标准，16 位数据，32 位声道  
数据位数和声道位数不一致，每个声道只需读写一次 SPI 数据寄存器（SPI\_DT），DMA 传输个数为 1。只有前 16 位是有效数据，后 16 位数据硬件默认输出和接收 0。
- 飞利浦标准或 PCM 标准或高字节标准，24 位数据，32 位声道  
数据位数和声道位数不一致，每个声道需读写二次 SPI 数据寄存器（SPI\_DT），DMA 传输个数为 2。前 16 位发送和接收第一笔 16 位数据，后 16 位发送和接收高 8 位数据，低 8 位数据硬件默认输出和接收 0。
- 飞利浦标准或 PCM 标准或高字节或低字节标准，32 位数据，32 位声道  
数据位数和声道位数一致，每个声道需读写二次 SPI 数据寄存器（SPI\_DT），DMA 传输个数为 2。  
数据分两次，依次发送和接收 16 位数据。
- 低字节标准，16 位数据，32 位声道  
数据位数和声道位数不一致，每个声道只需读写一次 SPI 数据寄存器（SPI\_DT），DMA 传输个数为 1。只有后 16 位是有效数据，前 16 位数据硬件默认输出和接收 0。
- 低字节标准，24 位数据，32 位声道  
数据位数和声道位数不一致，每个声道需读写二次 SPI 数据寄存器（SPI\_DT），DMA 传输个数为 2。  
前 16 位数据只有低 8 位有效，高 8 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

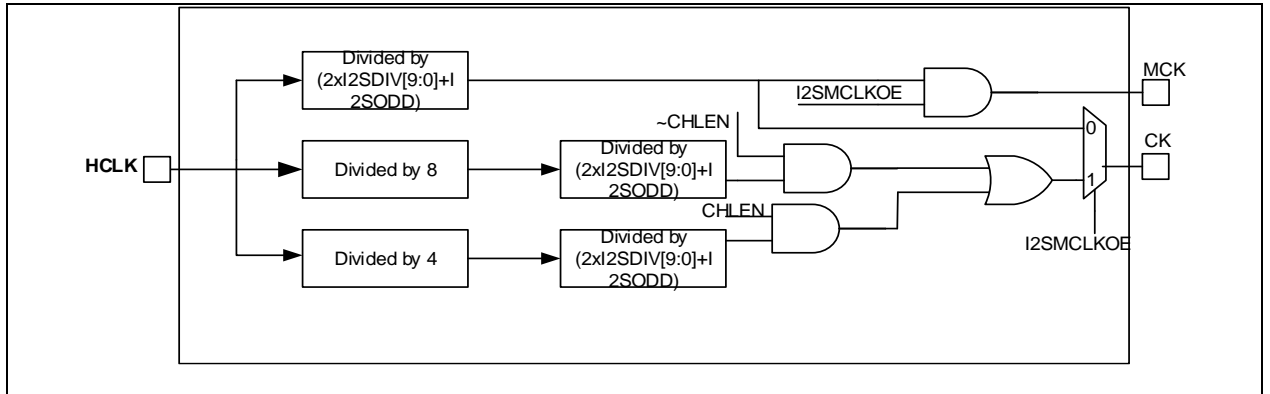
### 13.3.5 I2S\_CLK 控制器简述和配置流程

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 13-13 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 13-13。

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 13-13。

图 13-13 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

SysCLK (MHz)	MCL K	Target Fs (Hz)	16bit				32bit			
			I2S DIV	I2S_ODD	RealFs	Error	I2S DIV	I2S_ODD	RealFs	Error
240	NO	192000	19	1	192307	1.2%	10	0	187500	2.34%
240	NO	96000	39	0	96153	0.16%	19	1	96153	0.16%
240	NO	48000	78	0	48076	0.16%	39	0	48076	0.16%
240	NO	44100	85	0	44117	0.04%	42	1	44117	0.04%
240	NO	32000	117	0	32051	0.16%	58	1	32051	0.16%
240	NO	22050	170	0	22058	0.04%	85	0	22058	0.04%
240	NO	16000	234	1	15991	0.05%	117	0	16025	0.16%
240	NO	11025	340	0	11029	0.04%	170	0	11029	0.04%
240	NO	8000	469	0	7995	0.05%	234	1	7995	0.05%
240	YES	192000	2	1	187500	2.34%	2	1	187500	2.34%
240	YES	96000	5	0	93750	2.34%	5	0	93750	2.34%
240	YES	48000	10	0	46875	2.34%	10	0	46875	2.34%
240	YES	44100	10	1	44642	1.23%	10	1	44642	1.23%
240	YES	32000	14	1	32327	1.02%	14	1	32327	1.02%
240	YES	22050	21	1	21802	1.12%	21	1	21802	1.12%
240	YES	16000	29	1	15889	0.68%	29	1	15889	0.68%
240	YES	11025	42	1	11029	0.04%	42	1	11029	0.04%
240	YES	8000	58	1	8012	0.16%	58	1	8012	0.16%
200	No	192000	16	1	189393.9	1.36%	8	0	195312.5	1.73%
200	No	96000	32	1	96153.85	0.16%	16	1	94696.97	1.36%
200	No	48000	65	0	48076.92	0.16%	32	1	48076.92	0.16%
200	No	44100	71	0	44014.08	0.19%	35	1	44014.08	0.19%
200	No	32000	97	1	32051.28	0.16%	49	0	31887.76	0.35%
200	No	22050	141	1	22084.81	0.16%	71	0	22007.04	0.19%
200	No	16000	195	1	15984.65	0.10%	97	1	16025.64	0.16%
200	No	11025	283	1	11022.93	0.02%	141	1	11042.4	0.16%
200	No	8000	390	1	8002.561	0.03%	195	1	7992.327	0.10%
200	Yes	192000	3	0	130208.3	32.18%	3	0	130208.3	32.18%
200	Yes	96000	4	0	97656.25	1.73%	4	0	97656.25	1.73%
200	Yes	48000	8	0	48828.13	1.73%	8	0	48828.13	1.73%
200	Yes	44100	9	0	43402.78	1.58%	9	0	43402.78	1.58%
200	Yes	32000	12	0	32552.08	1.73%	12	0	32552.08	1.73%
200	Yes	22050	17	1	22321.43	1.23%	17	1	22321.43	1.23%

200	Yes	16000	24	1	15943.88	0.35%	24	1	15943.88	0.35%
200	Yes	11025	35	1	11003.52	0.19%	35	1	11003.52	0.19%
200	Yes	8000	49	0	7971.939	0.35%	49	0	7971.939	0.35%
100	No	192000	8	0	195312.5	1.73%	4	0	195312.5	1.73%
100	No	96000	16	1	94696.97	1.36%	8	0	97656.25	1.73%
100	No	48000	32	1	48076.92	0.16%	16	1	47348.48	1.36%
100	No	44100	35	1	44014.08	0.19%	17	1	44642.86	1.23%
100	No	32000	49	0	31887.76	0.35%	24	1	31887.76	0.35%
100	No	22050	71	0	22007.04	0.19%	35	1	22007.04	0.19%
100	No	16000	97	1	16025.64	0.16%	49	0	15943.88	0.35%
100	No	11025	141	1	11042.4	0.16%	71	0	11003.52	0.19%
100	No	8000	195	1	7992.327	0.10%	97	1	8012.821	0.16%
100	Yes	96000	2	0	97656.25	1.73%	2	0	97656.25	1.73%
100	Yes	48000	4	0	48828.13	1.73%	4	0	48828.13	1.73%
100	Yes	44100	4	1	43402.78	1.58%	4	1	43402.78	1.58%
100	Yes	32000	6	0	32552.08	1.73%	6	0	32552.08	1.73%
100	Yes	22050	9	0	21701.39	1.58%	9	0	21701.39	1.58%
100	Yes	16000	12	0	16276.04	1.73%	12	0	16276.04	1.73%
100	Yes	11025	17	1	11160.71	1.23%	17	1	11160.71	1.23%
100	Yes	8000	24	1	7971.939	0.35%	24	1	7971.939	0.35%
72	No	192000	6	0	187500	2.34%	3	0	187500	2.34%
72	No	96000	11	1	97826.09	1.90%	6	0	93750	2.34%
72	No	48000	32	1	34615.38	27.88%	11	1	48913.04	1.90%
72	No	44100	25	1	44117.65	0.04%	13	0	43269.23	1.88%
72	No	32000	35	0	32142.86	0.45%	17	1	32142.86	0.45%
72	No	22050	51	0	22058.82	0.04%	25	1	22058.82	0.04%
72	No	16000	70	1	15957.45	0.27%	35	0	16071.43	0.45%
72	No	11025	102	0	11029.41	0.04%	51	0	11029.41	0.04%
72	No	8000	140	1	8007.117	0.09%	70	1	7978.723	0.27%
72	Yes	96000	2	0	70312.5	26.76%	2	0	70312.5	26.76%
72	Yes	48000	3	0	46875	2.34%	3	0	46875	2.34%
72	Yes	44100	3	0	46875	6.29%	3	0	46875	6.29%
72	Yes	32000	4	1	31250	2.34%	4	1	31250	2.34%
72	Yes	22050	6	1	21634.62	1.88%	6	1	21634.62	1.88%
72	Yes	16000	9	0	15625	2.34%	9	0	15625	2.34%
72	Yes	11025	13	0	10817.31	1.88%	13	0	10817.31	1.88%
72	Yes	8000	17	1	8035.714	0.45%	17	1	8035.714	0.45%

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

#### DMA 发送配置流程：

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



- 配置 DMA 传输数据个数：在 DMA 控制寄存器相关位置配置期望传输的数据个数
- 配置 DMA 传输通道优先级：在 DMA 控制寄存器相关位置配置当前所使用通道的 SPI 的 DMA 传输通道优先级。
- 配置 DMA 中断产生时机：在 DMA 控制寄存器相关位置配置是在传输完成或传输完成一半时产生 DMA 中断。
- 使能 DMA 传输通道：在 DMA 控制寄存器相关位置使能当前所选用的 DMA 通道

#### DMA接收配置流程：

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

### 13.3.7 发送器接收器简述和配置流程

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

但需要特别注意的是：

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

下面给出发送器和接收器的配置流程

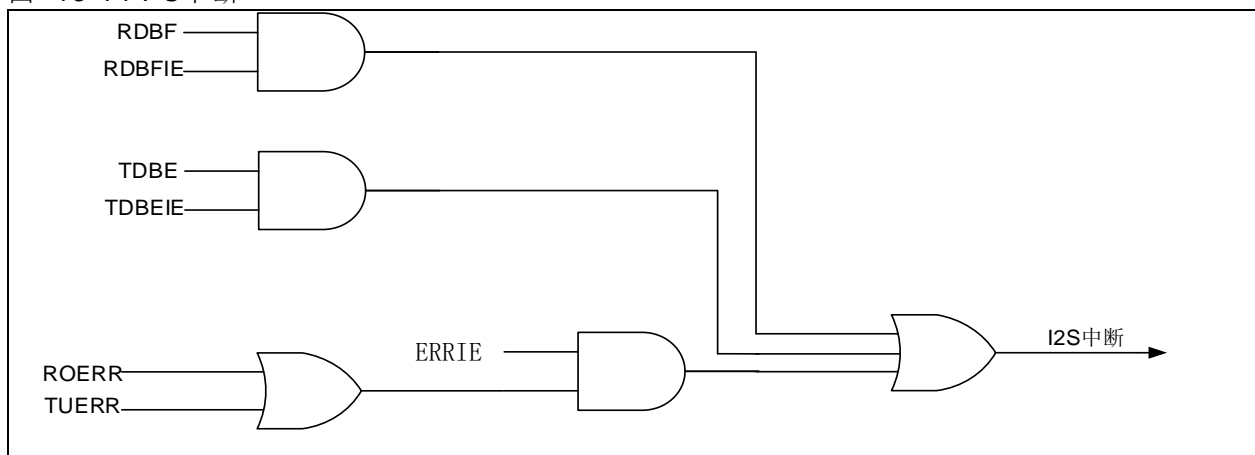
#### I<sup>2</sup>S 发送器配置流程：

- 配置操作模式选择器
- 配置音频协议选择器
- 配置 I2S\_CLK 控制器
- 配置 DMA(若需要开启 DMA 传输)
- 置位 I2SEN 位开启 I<sup>2</sup>S
- 按上述方式配置 I<sup>2</sup>SxEXT（若需要使用 I<sup>2</sup>S 全双工）



**I<sup>2</sup>S 接收器配置流程:**

- 配置操作模式选择器
- 配置音频协议选择器
- 配置 I2S\_CLK 控制器
- 配置 DMA(若需要开启 DMA 传输)
- 置位 I2SEN 位开启 I<sup>2</sup>S
- 按上述方式配置 I<sup>2</sup>SxEXT (若需要使用 I<sup>2</sup>S 全双工)

**13.3.8 中断**图 13-14 I<sup>2</sup>S 中断**13.3.9 I/O 管脚控制**

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

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

**13.4 SPI 寄存器**

必须以字（32 位）的方式操作这些外设寄存器。

表 13-2 SPI 寄存器列表及其复位值

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) (I<sup>2</sup>S模式下不使用)

域	简称	复位值	类型	功能
位 15	SLBEN	0x0	rw	单线双向半双工模式使能 (Single line bidirectional half-duplex enable) 0: 关闭; 1: 开启。
位 14	SLBTD	0x0	rw	单线双向半双工模式传输方向 (Single line bidirectional half-duplex transmission direction) 和 SLBEN 位一起决定在“单线双向半双工”模式下数据的传输方向 0: 接收模式; 1: 发送模式。
位 13	CCEN	0x0	rw	CRC 校验使能 (CRC calculation enable) 0: 关闭; 1: 开启。
位 12	NTC	0x0	rw	下一笔传输数据为 CRC (Next transmission CRC) 该位置起表示下一笔传输的数据为 CRC 数据。 0: 普通数据; 1: CRC 数据。
位 11	FBN	0x0	rw	帧位个数 (frame bit num) 该位配置发送/接收时数据帧位个数。 0: 8 位; 1: 16 位。
位 10	ORA	0x0	rw	仅接收有效 (Only receive active) 在“双线单向”模式时, 该位置起表示只有接收有效, 发送被禁止。 0: 发送和接收; 1: 仅接收。
位 9	SWCSEN	0x0	rw	软件 CS 模式使能 (Software CS enable) 当该位被置起时, CS 管脚上的电平由 SWCSIL 位的值决定, 此时在 CS 管脚上的 I/O 电平状态无效。 0: 关闭; 1: 开启。
位 8	SWCSIL	0x0	rw	软件 CS 内部电平 (Software CS internal level) 该位只在 SWCSEN 位置起时有意义, 它决定了 CS 上的内部电平状态。 做主设备时, 该位必须设置置起。 0: 低电平; 1: 高电平。
位 7	LTF	0x0	rw	LSB 先传输 (LSB transmit first) 该位用于选择数据先传输 MSB 还是 LSB。 0: MSB; 1: LSB。
位 6	SPIEN	0x0	rw	SPI 使能 (SPI enable) 0: 关闭; 1: 开启。
位 5: 3	MDIV	0x0	rw	主模式时钟频率分频系数 (Master clock frequency division) 作主模式时, 分频系数对外设时钟进行分频, 作为 SPI 时钟, MDIV[3]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 I<sup>2</sup>S 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: 9	保留位	0x00	resd	硬件强制为 0
位 8	MDIV	0x0	rw	主模式时钟频率分频系数 (Master clock frequency division) 详见 MDIV[2: 0]在 SPI_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: 3	保留位	0x0	resd	保持默认值。
位 2	HWCSE	0x0	rw	硬件 CS 输出使能 (Hardware CS output enable) 该位做主设备时才有意义, 设置为'1'时, CS 脚 I/O 口输出低电平, 设置为'0'时, 必须保证 CS 脚 I/O 口输入为高电平。 0: 关闭; 1: 开启。
位 1	DMATEN	0x0	rw	DMA 发送使能 (DMA transmit enable) 0: 关闭; 1: 开启。
位 0	DMAREN	0x0	rw	DMA 接收使能 (DMA receive enable) 0: 关闭; 1: 开启。

## 13.4.3 SPI状态寄存器 (SPI\_STS)

域	简称	复位值	类型	功能
位 15: 8	保留位	0x00	resd	硬件强制为 0
位 7	BF	0x0	ro	通信忙标志 (Busy flag) 0: 通信空闲; 1: 通信忙。
位 6	ROERR	0x0	ro	接收器溢出错误 (Receiver overflow error) 0: 无; 1: 有。
位 5	MMERR	0x0	ro	主模式错误 (Master mode error) 该位由硬件置位, 软件清除 (先读或写 SPI_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 <sup>2</sup> S 模式使用。
位 2	ACS	0x0	ro	音频通道状态 (Audio channel state) 该位表示当前传输的音频左右声道状态。 0: 左声道; 1: 右声道。 注: 该位只在 I <sup>2</sup> S 模式使用。
位 1	TDBE	0x1	ro	发送数据缓冲器空 (Transmit data buffer empty) 0: 非空; 1: 空。
位 0	RDBF	0x0	ro	接收数据缓冲器满 (Receive data buffer full) 0: 未滿; 1: 满。

#### 13.4.4 SPI数据寄存器 (SPI\_DT)

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

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

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

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

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

#### 13.4.7 SPITxCRC寄存器 (SPI\_TCRC)

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

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

域	简称	复位值	类型	功能
位 15: 12	保留位	0x0	resd	硬件强制为 0
位 11	I2SMSEL	0x0	rw	I <sup>2</sup> S 模式选择 (I <sup>2</sup> S mode select) 0: SPI 模式; 1: I <sup>2</sup> S 模式。
位 10	I2SEN	0x0	rw	I <sup>2</sup> S 使能 (I <sup>2</sup> S enable) 0: 关闭; 1: 开启。
位 9: 8	OPERSEL	0x0	rw	I <sup>2</sup> S 操作选择 (I <sup>2</sup> S operation select) 00: 从设备发送; 01: 从设备接收; 10: 主设备发送; 11: 主设备接收。
位 7	PCMFSSSEL	0x0	rw	PCM 帧同步 (PCM frame synchronization select) 该位只在使用 PCM 标准时才有意义。 0: 短帧同步; 1: 长帧同步。
位 6	保留位	0x0	resd	保持默认值。
位 5: 4	STDSEL	0x0	rw	I <sup>2</sup> S 标准选择 (I <sup>2</sup> S standard select) 00: 飞利浦标准; 01: 高字节对齐标准 (左对齐); 10: 低字节对齐标准 (右对齐); 11: PCM 标准。
位 3	I2SCLKPOL	0x0	rw	I <sup>2</sup> S 时钟极性 (I <sup>2</sup> S clock polarity) 时钟管脚上总线空闲时时钟输出的极性。 0: 低电平; 1: 高电平。
位 2: 1	I2SDBN	0x0	rw	I <sup>2</sup> S 数据位个数 (I <sup>2</sup> S data bit num) 00: 16 位; 01: 24 位; 10: 32 位; 11: 不允许。
位 0	I2SCBN	0x0	rw	I <sup>2</sup> S 声道位个数 (I <sup>2</sup> S channel bit num) 该位只有在 I <sup>2</sup> S 数据位个数为 16 位时配置才有意义, 否则都由硬件固定为 32 位。 0: 16 位宽; 1: 32 位宽。

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

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

## 14 定时器（TIMER）

AT32F403A/407/407A 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	EXT 输入	刹车输入
高级控制定时器	TMR1 TMR8	16	向上 向下 向上/向下	8 位	1~65535	支持	4	支持	支持	支持
	TMR2 TMR5	16/32	向上 向下 向上/向下	不支持	1~65535	支持	4	支持	支持	不支持
通用定时器	TMR3 TMR4	16	向上 向下 向上/向下	不支持	1~65535	支持	4	支持	支持	不支持
	TMR9 TMR12	16	向上	不支持	1~65535	不支持	2	支持	不支持	不支持
	TMR10 TMR11 TMR13 TMR14	16	向上	不支持	1~65535	不支持	1	不支持	不支持	不支持
	TMR6 TMR7	16	向上	不支持	1~65535	支持	不支持	不支持	不支持	不支持

Timer 类型	Timer	计数位数	计数方式	PWM 输出	单周期输出	互补输出	死区	编码器接口连接	霍尔传感器接口连接	连动外设
高级控制定时器	TMR1 TMR8	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/DAC
	TMR2 TMR5	16/32	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
通用定时器	TMR3 TMR4	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
	TMR9 TMR12	16	向上	支持	支持	不支持	不支持	不支持	不支持	定时器同步
	TMR10 TMR11 TMR13 TMR14	16	向上	支持	支持	不支持	不支持	不支持	不支持	无
	TMR6 TMR7	16	向上	不支持	不支持	不支持	不支持	不支持	不支持	DAC

### 14.1 基本定时器（TMR6和TMR7）

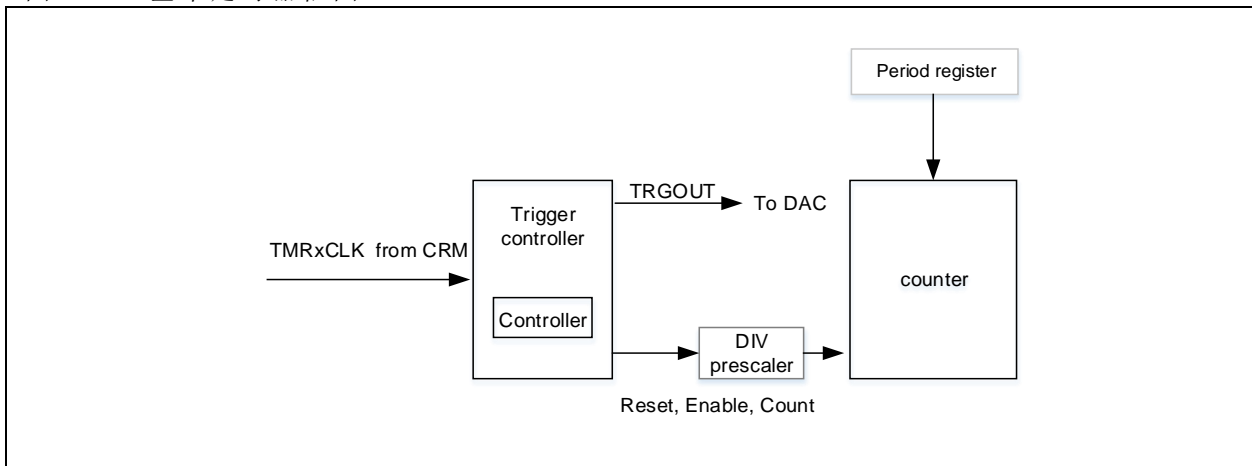
#### 14.1.1 TMR6和TMR7简介

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的主要特性

- 由内部时钟用作计数时钟
- 16 位向上计数器
- 触发 DAC 的同步电路（TMR6 和 TMR7 独有的特性）
- 支持溢出事件中断和 DMA 请求

图 14-1 基本定时器框图

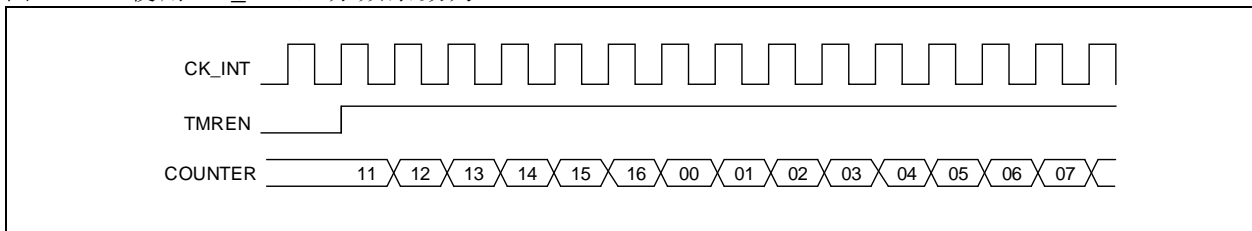


### 14.1.3 TMR6和TMR7的功能

#### 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，计数器的值可由周期寄存器（TMRx\_PR）载入。默认下，周期寄存器（TMRx\_PR）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），周期寄存器（TMRx\_PR）值会在溢出事件发生时传入它的影子寄存器。溢出事件由 OVFEN、OVFS 位配置。

将 TMREN 位置 1 可使能定时器计数，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。

向上计数模式中，当计数值达到周期寄存器（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，则计数器溢出后不再重载预分频值和重载值，否则预分频值和重载值将在溢出事件后更新。

图 14-3 PRBEN=0时的溢出事件

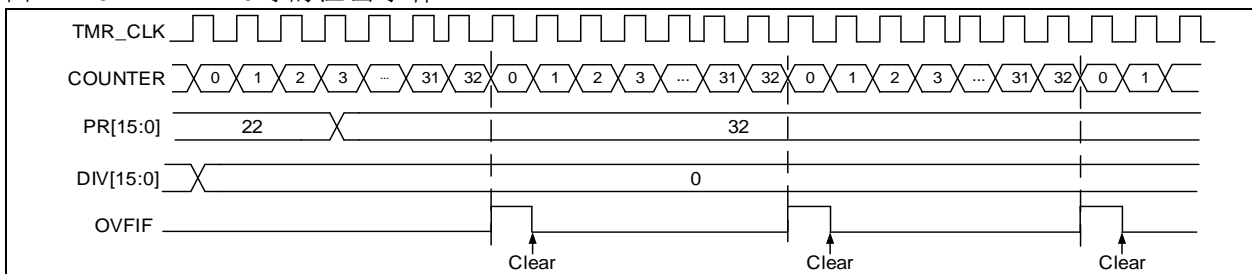




图 14-4 PRBEN=1时的溢出事件

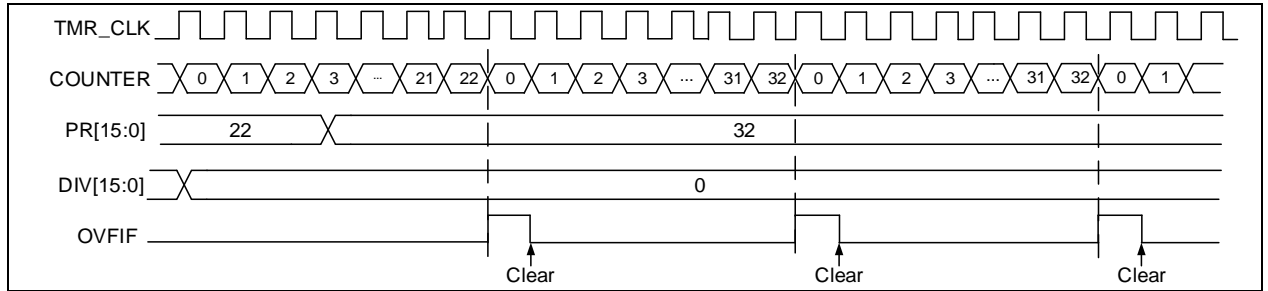
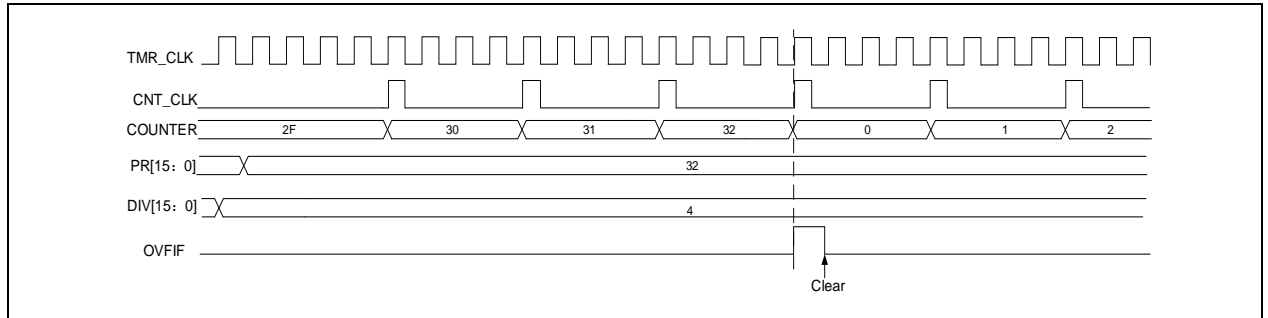


图 14-5 计数器时序图，内部时钟分频因子为4



### 14.1.3.3 调试模式

当微控制器进入调试模式（Cortex™-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

必须以字（32 位）的方式操作这些外设寄存器。

下表中将 TMRx 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
TMRx_ISTS	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: 8	保留	0x00	resd	保持默认值。
位 7	PRBEN	0x0	rw	周期缓冲使能（Period buffer enable） 0：缓冲关闭； 1：缓冲开启。
位 6: 4	保留	0x0	resd	保持默认值。
位 3	OCMEN	0x0	rw	单周期使能（One cycle mode enable） 该功能用于选择更新事件后，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。
				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 1	OVFEN	0x0	rw	
				使能定时器 (TMR enable) 0: 关闭; 1: 开启。
位 0	TMREN	0x0	rw	

## 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。
				主定时器输出信号选择 (Primary TMR output selection) TMRx 输出到次定时器的信号选择: 000: 复位; 001: 使能; 010: 更新;
位 6: 4	PTOS	0x0	rw	
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。
				溢出事件的 DMA 请求使能 (overflow event DMA request enable) 0: 关闭; 1: 开启。
位 8	OVFDEN	0x0	rw	
位 7: 1	保留	0x00	resd	保持默认值。
				溢出中断使能 (overflow interrupt enable) 0: 关闭; 1: 开启。
位 0	OVFIEN	0x0	rw	

## 14.1.4.4 TMR6和TMR7中断状态寄存器 (TMRx\_ISTS)

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

## 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。
				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;
位 0	OVFSWTR	0x0	rw0c	

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 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（TMR2到TMR5）

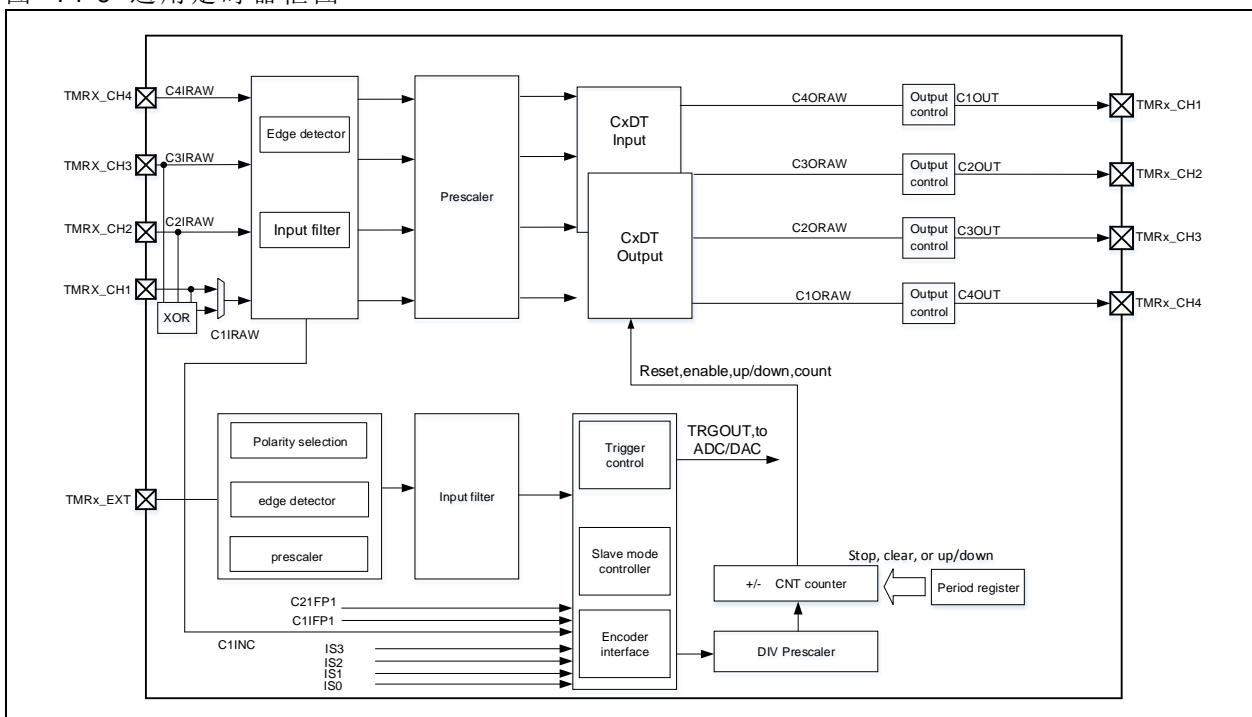
### 14.2.1 TMRx简介

通用定时器 TMR2 到 TMR5 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 TMRx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（TMR2/5 可扩展至 32 位）
- 4 组独立通道，支持输入捕获、输出比较、PWM 生成、单周期模式。
- 定时器之间可互联同步
- 支持溢出事件、触发事件、通道事件触发中断/DMA
- 支持 TMR burst DMA 传输

图 14-6 通用定时器框图



### 14.2.3 TMRx功能描述

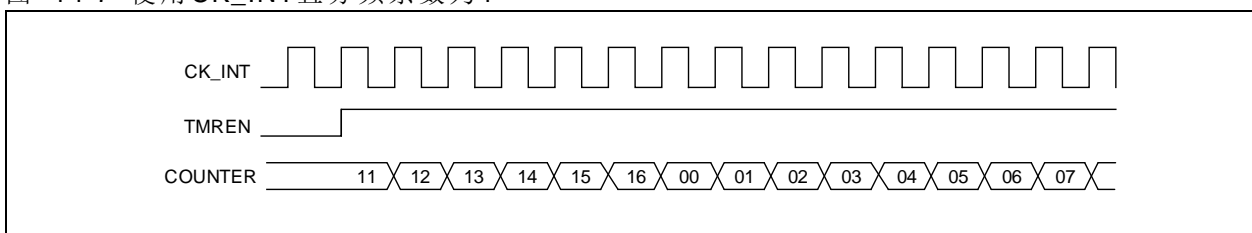
#### 14.2.3.1 计数时钟

TMR2 至 5 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。

##### 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。

图 14-7 使用CK\_INT且分频系数为1



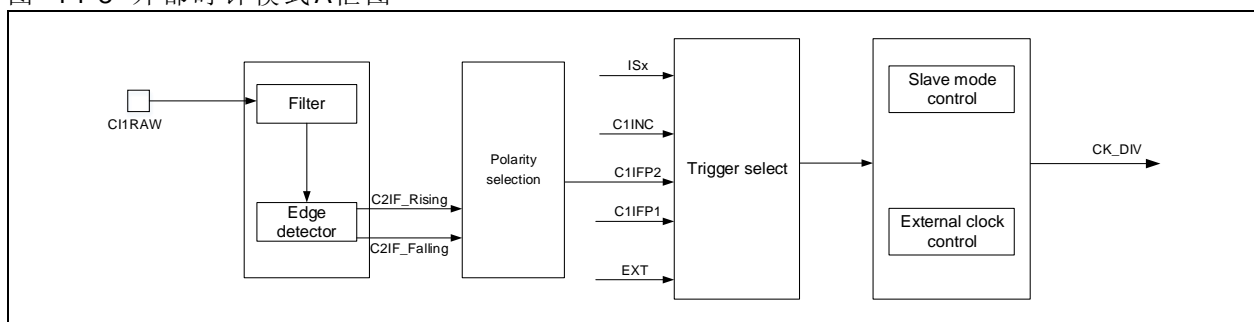
##### 外部时钟（TRGIN/EXT）

计数时钟可由两种外部时钟源提供，分别为 TRGIN 和 EXT 信号。

当 SMSEL=3'111 时，外部时钟模式 A 被选中，配置 STIS[2: 0]来选择 TRGIN 信号驱动计数器计数。

当  $ECMBEN=1$  时，外部时钟模式 B 被选中，计数器由 EXT 信号驱动计数。

图 14-8 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 14-9 使用外部时钟模式A计数

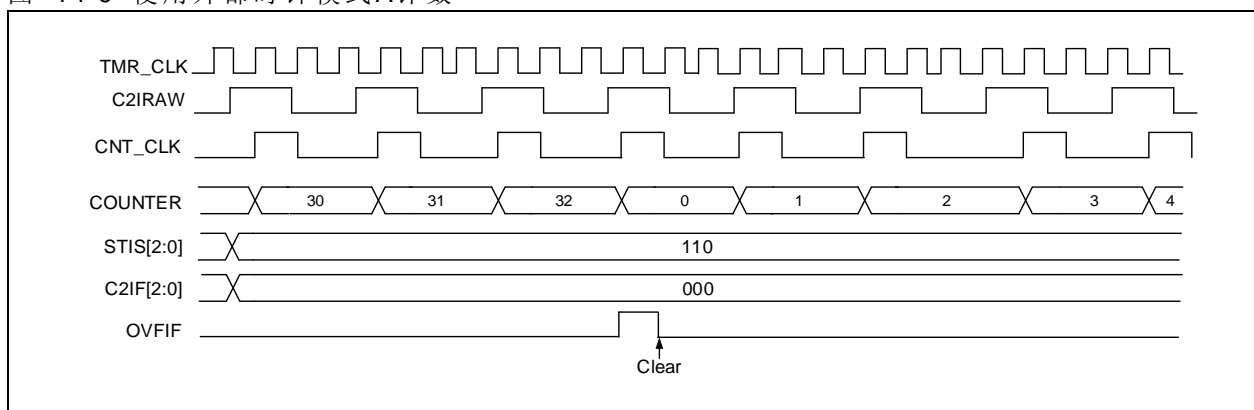
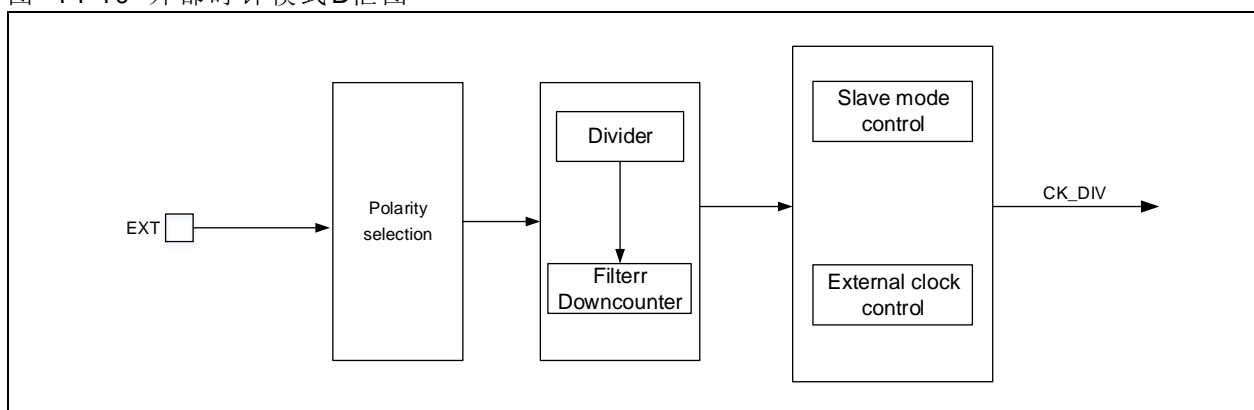
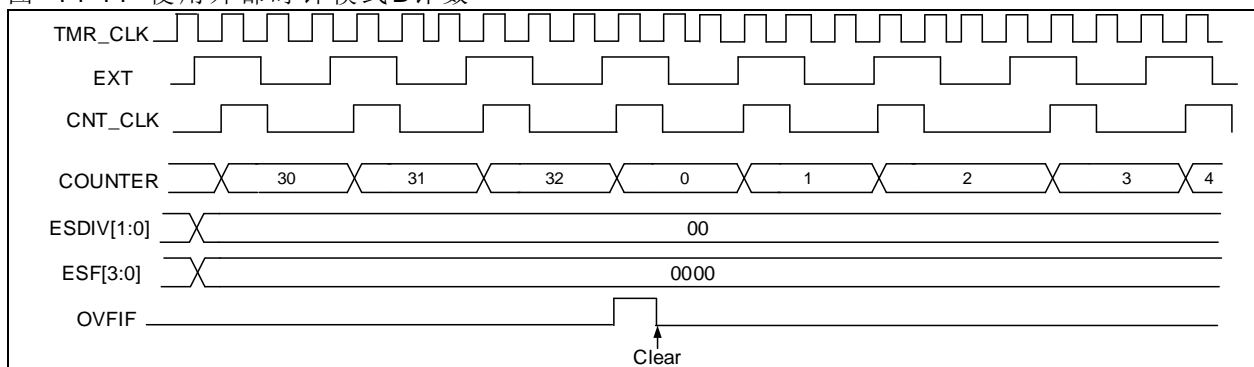


图 14-10 外部时钟模式B框图



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 14-11 使用外部时钟模式B计数



#### 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。

TMR2 至 5 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。

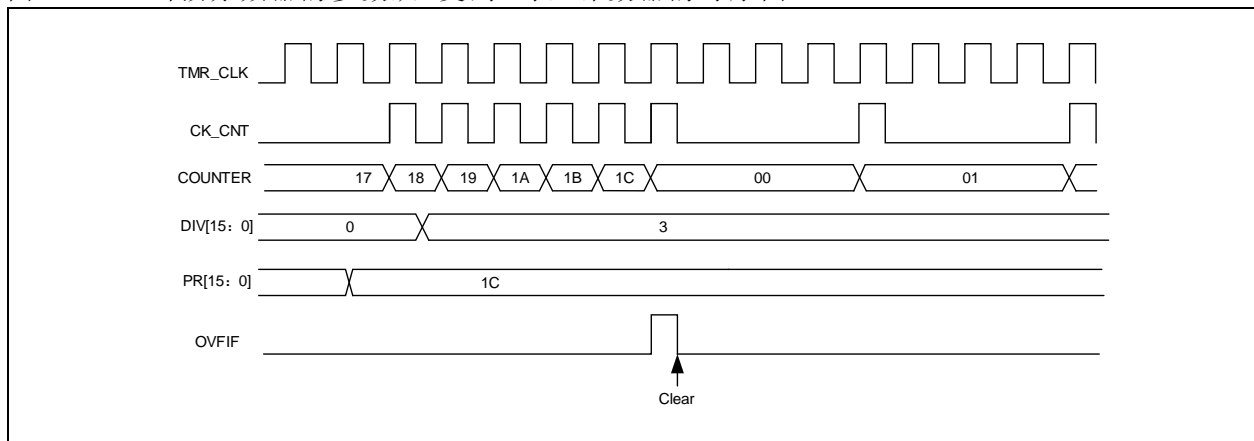
表 14-3 TMRx内部触发连接

次定时器	IS0 (STIS = 000)	IS1 (STIS = 001)	IS2 (STIS = 010)	IS3 (STIS = 011)
TMR2	TMR1	TMR8/USB_SOF <sup>(2)</sup>	TMR3	TMR4
TMR3	TMR1	TMR2	TMR5	TMR4
TMR4	TMR1	TMR2	TMR3	TMR8
TMR5	TMR2	TMR3	TMR4	TMR8

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。

注意 2：IS1 可以选择 TMR8 或 USB\_SOF，由 IOMUX\_MAP4 的 TMR2IS1\_IRMP 位控制。

图 14-12 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

TMR2 至 5 定时器提供了多种计数模式，用来满足不同的应用场景。其内部拥有一个支持 16 位向上计、向下、中央双向对齐计数的计数器，TMR2/5 可通过将 PMEN 位置 1 扩展至 32 位，计数器的值可由周期寄存器（TMRx\_PR）载入。默认下，周期寄存器（TMRx\_PR）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），周期寄存器（TMRx\_PR）值会在溢出事件发生时传入它的影子寄存器。溢出事件由 OVFEN、OVFS 位配置。

将 TMREN 位置 1 可使能定时器计数，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。

#### 向上计数模式

向上计数模式中，当计数值达到周期寄存器（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 OVFI 位置 1。若禁止产生溢出事件，则计数器溢出后不再重载预分频值和重载值，否则预分频值和重载值将在溢出事件后更新。

图 14-13 PRBEN=0时的溢出事件

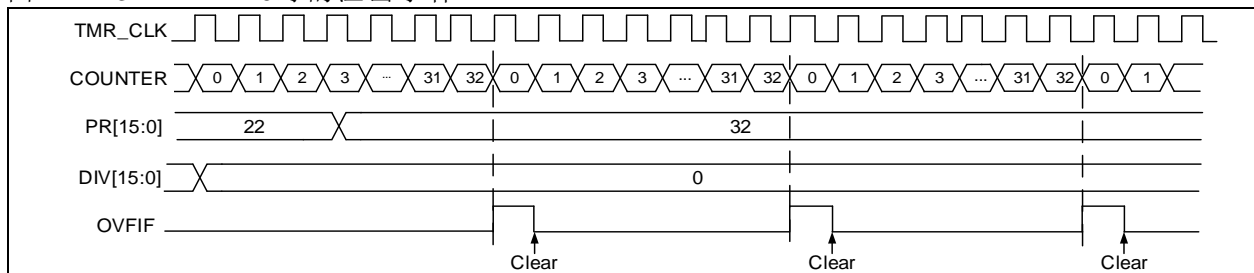
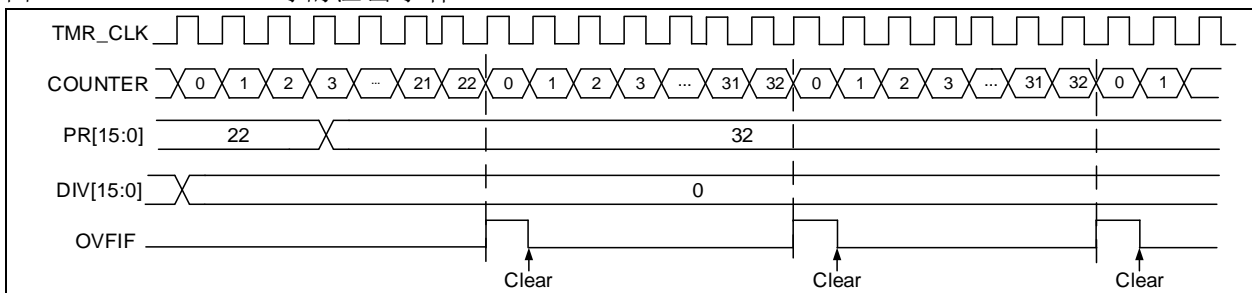


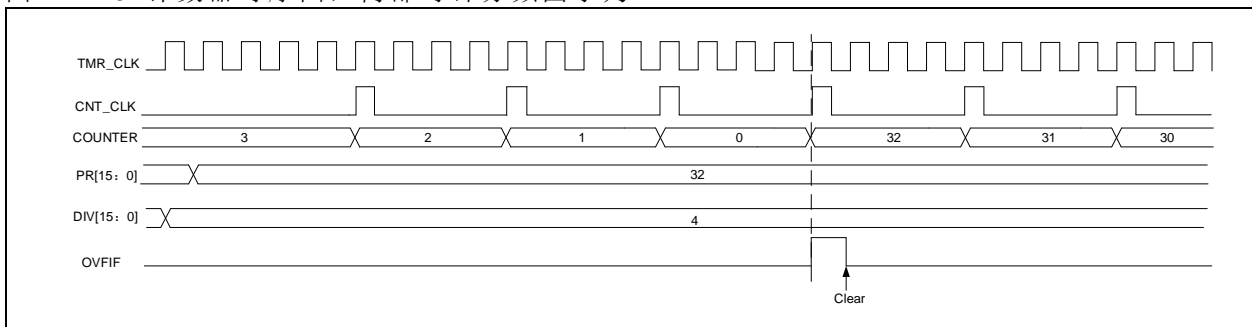
图 14-14 PRBEN=1时的溢出事件



### 向下计数模式

向下计数模式中，当计数值达到 0 值时，重新从周期寄存器（TMRx\_PR）向上下数，计数器下溢并产生溢出事件。

图 14-15 计数器时序图，内部时钟分频因子为 4

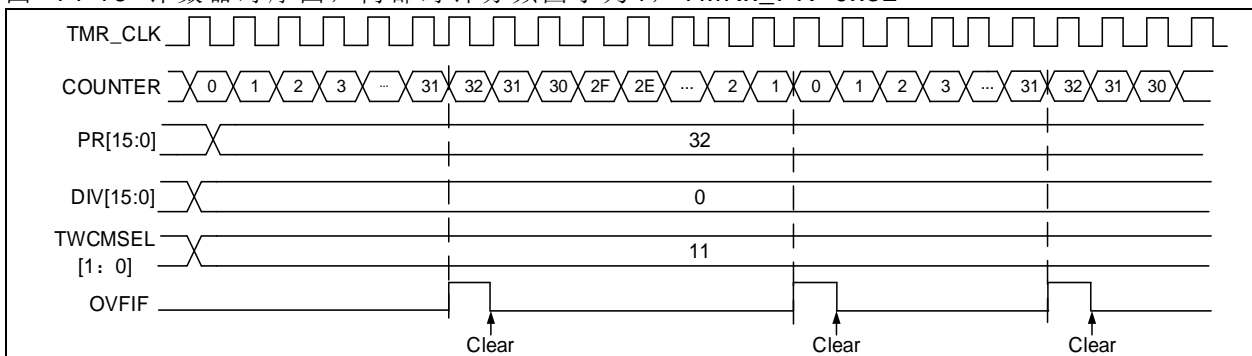


### 中央双向对齐计数模式

中央双向对齐计数模式中，计数器交替向上、向下计数。当计数值从周期寄存器（TMRx\_PR）值向下计数到 1 值时，产生下溢事件，然后从 0 开始向上计数；当向上计数到周期寄存器（TMRx\_PR）值-1 时，产生上溢事件，之后从周期寄存器（TMRx\_PR）值向下计数。计数器计数方向可由计数器方向控制位（OWCDIR）实时查看。

注意：中央双向对齐计数模式下，OWCDIR 位为只读位。

图 14-16 计数器时序图，内部时钟分频因子为 1，TMRx\_PR=0x32



### 编码器模式

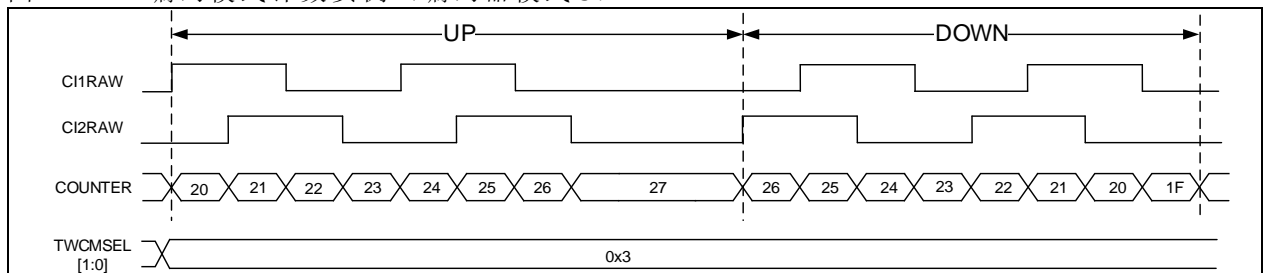
将 SMSEL[2:0]配置为 3'b001/3'b010/3'b011 可开启编码模式，编码模式下需提供两个输入（C1IN/C2IN），根据一个输入的电平值，计数器将在另一个输入的边沿向上或向下计数。计数方向将由 OWCDIR 值指示。编码模式下计数器计数方向如下表所示：



表 14-4 计数方向与编码器信号的关系

有效边沿	相对信号的电平 (C1INFP1 对应 C2IN, C2INFP2 对应 C1IN)	C1INFP1 信号		C2INFP2 信号	
		上升	下降	上升	下降
仅在 C1IN 计数	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
仅在 C2IN 计数	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
在 C1IN 和 C2IN 上计数	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 14-17 编码模式计数实例（编码器模式C）



### 14.2.3.3 TMR输入部分

TMR2 至 5 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，可用于对输入信号的滤波、选择、分频和输入捕获功能。

图 14-18 输入/输出通道1的主电路

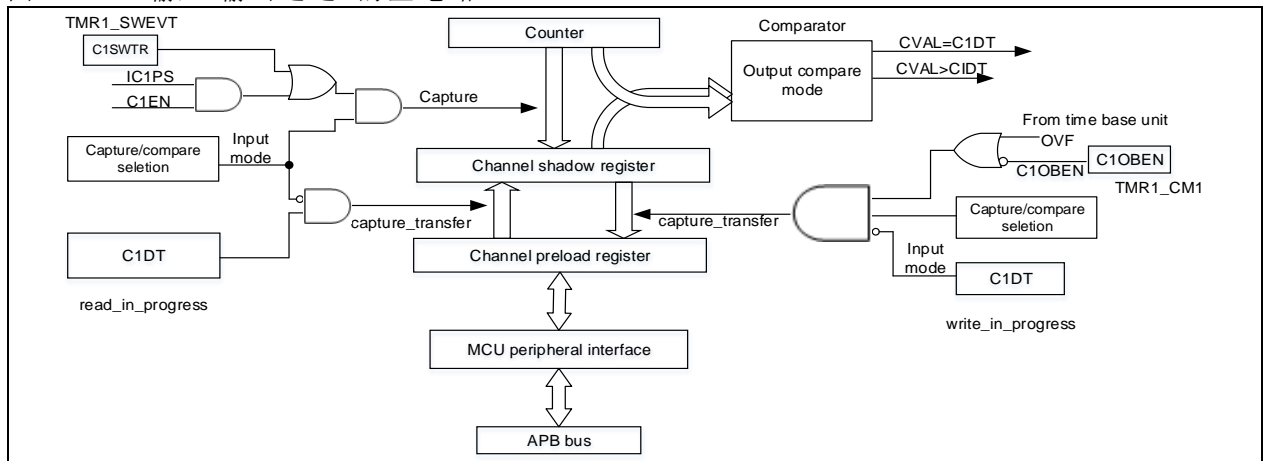
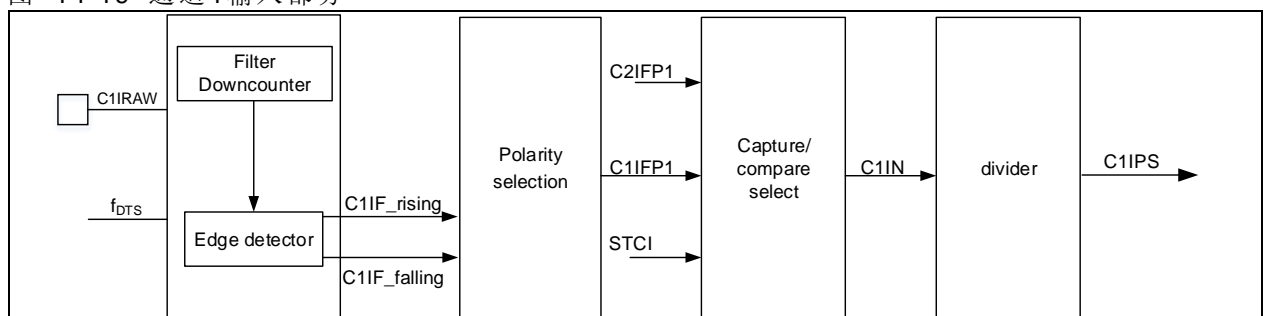


图 14-19 通道1输入部分



#### 输入模式

此模式下，当选中的触发信号被检测到时，通道寄存器（TMRx\_CxDT）会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 已置 1 后检测到选中的触发信号，则将 CxOF 位置 1。

若要捕获 C1IN 输入的上升沿，可按如下进行配置：

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

#### 多输入异或

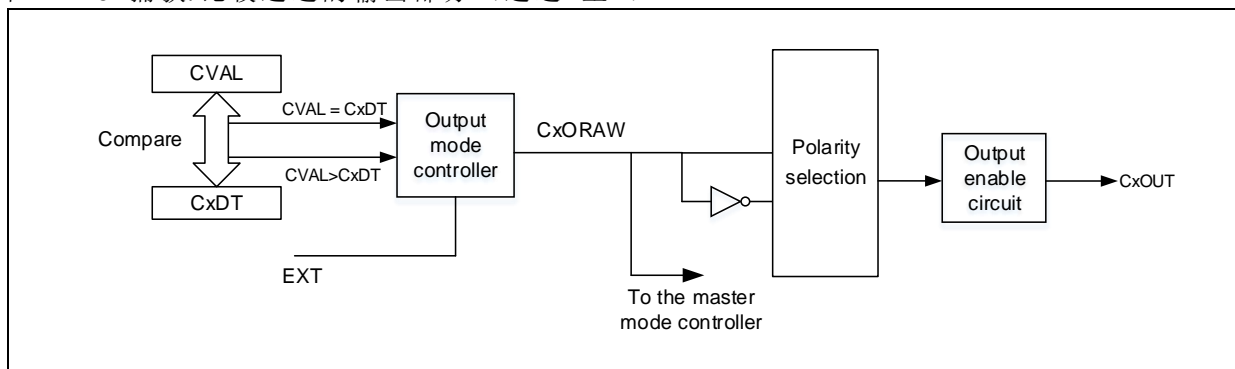
通道 1 的输入端可选择 TMRx\_CH1、TMRx\_CH2 和 TMRx\_CH3 经异或逻辑后输入。将控制寄存器 2（TMRx\_CTRL2）中的 C1INSEL 位置 1 可开启此功能。

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

### 14.2.3.4 TMR输出部分

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

图 14-20 捕获/比较通道的输出部分（通道 1 至 4）



#### 输出模式

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

输出比较模式有以下子类：

- **PWM 模式：**CxOCTRL=3'b110/111 时，开启 PWM 模式，每个通道可独立配置并输出一路 PWM 信号。此时，输出信号的周期由 TMRx\_PR 配置，占空比由 CxDT 值配置，计数器值与通道寄存器（TMRx\_CxDT）值进行比较，根据计数方向输出指定电平信号，关于 PWM 模式 A/B 详见 CxOCTRL[2: 0]位描述。当计数模式为中央双向对齐计数时，可根据 OWCDIR 位指示计数方向。
- **强制输出模式：**CxOCTRL=3'b100/101 时，开启强制输出模式。此时，CxORAW 信号的电平被强制输出为配置的电平，而与计数值无关。虽然输出信号不依赖于比较结果，但通道标志位和 DMA 请求仍依赖于比较结果。
- **输出比较模式：**CxOCTRL=3'b001/010/011 时，开启输出比较模式。此时，当计数值与 CxDT 值匹配时，CxORAW 被强制为高电平、低电平或进行电平翻转。
- **单周期模式：**PWM 模式的特例，将 OCMEN 位置 1 可开启单周期模式，此模式下，仅在当前计数周期中进行比较匹配，完成当前计数后，TMREN 位清 0，因此仅输出一个脉冲。当配置为向上计数模式时，需要严格配置 CVAL<CxDT≤PR；向下计数时，需严格配置 CVAL>CxDT。
- **快速输出模式：**将 CxOIEN 位置 1 可开启此功能，开启后 CxORAW 电平值不再在计数值与 CxDT 匹配时变化，而是在当前计数周期开始时，也就是说，比较结果被提前了，计数器值与通道寄存器（TMRx\_CxDT）的比较结果将会提前决定 CxORAW 的电平。

图 14-21 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 14-22 展示了计数器向上计数与 PWM 模式 A 配合的例子，PR=0x32，CxDT 配置为不同的值时输出时输出信号的翻转情况。

图 14-23 展示了计数器中央双向对齐计数与 PWM 模式 A 配合的例子，PR=0x32，CxDT 配置为不同的值时输出时输出信号的翻转情况。

图 14-24 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 14-21 计数值与 C1DT 值匹配时翻转 C1ORAW

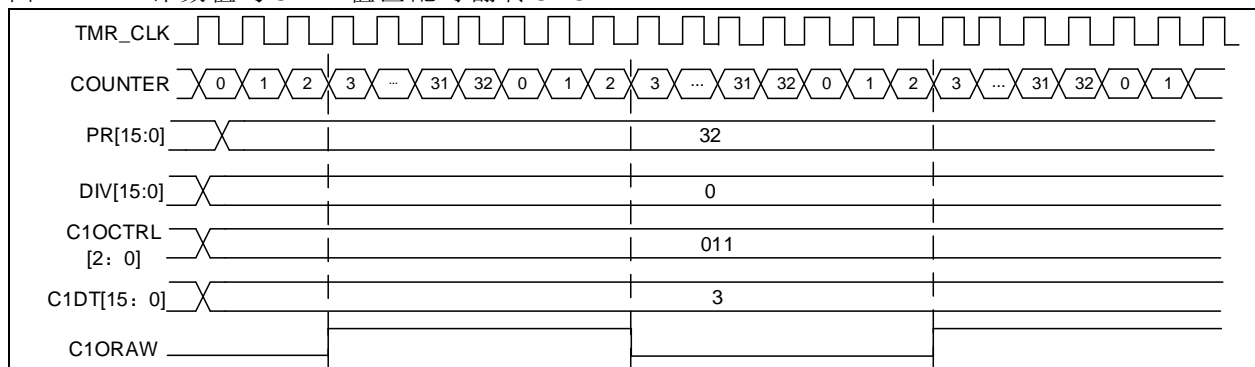


图 14-22 向上计数下 PWM 模式 A

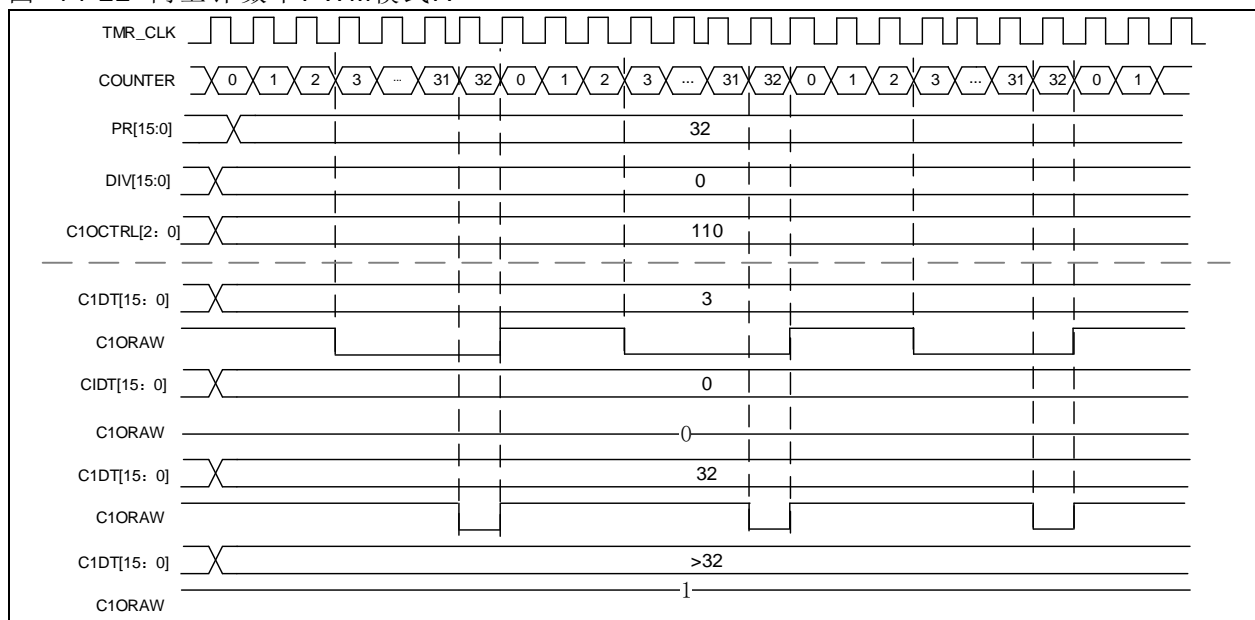


图 14-23 中央双向对齐计数下 PWM 模式 A

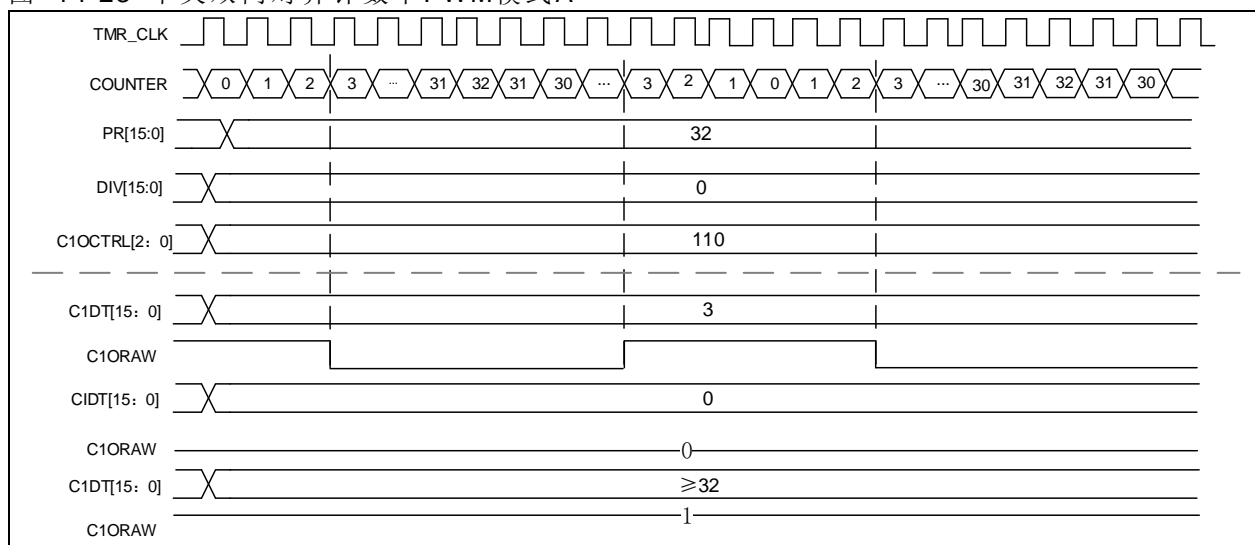
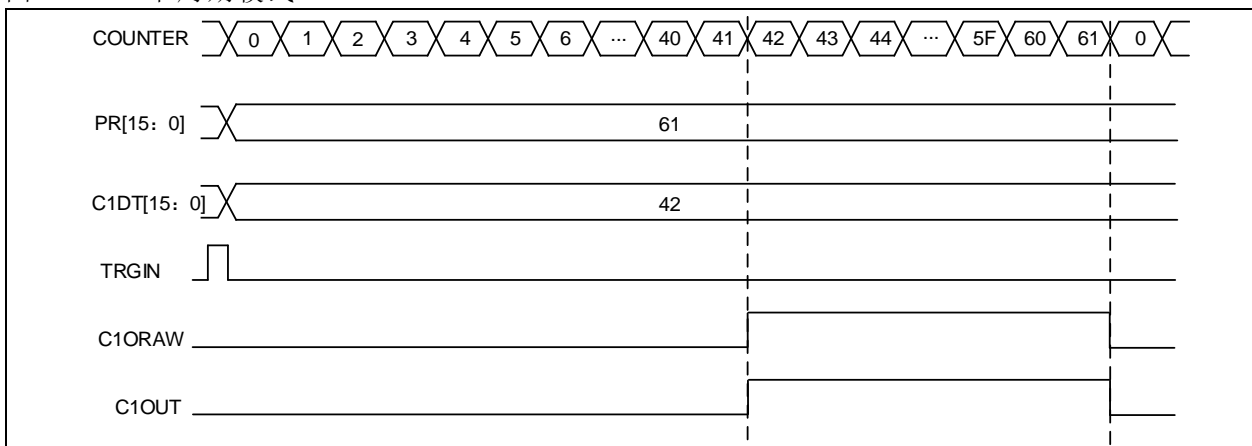


图 14-24 单周期模式

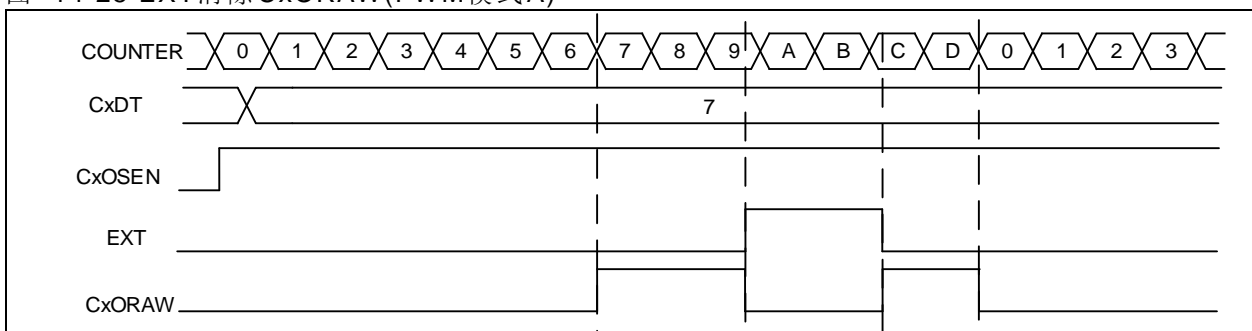


### CxORAW 信号清除

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

强制模式时, 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 14-25 EXT清除CxORAW(PWM模式A)



## 14.2.3.5 定时器同步

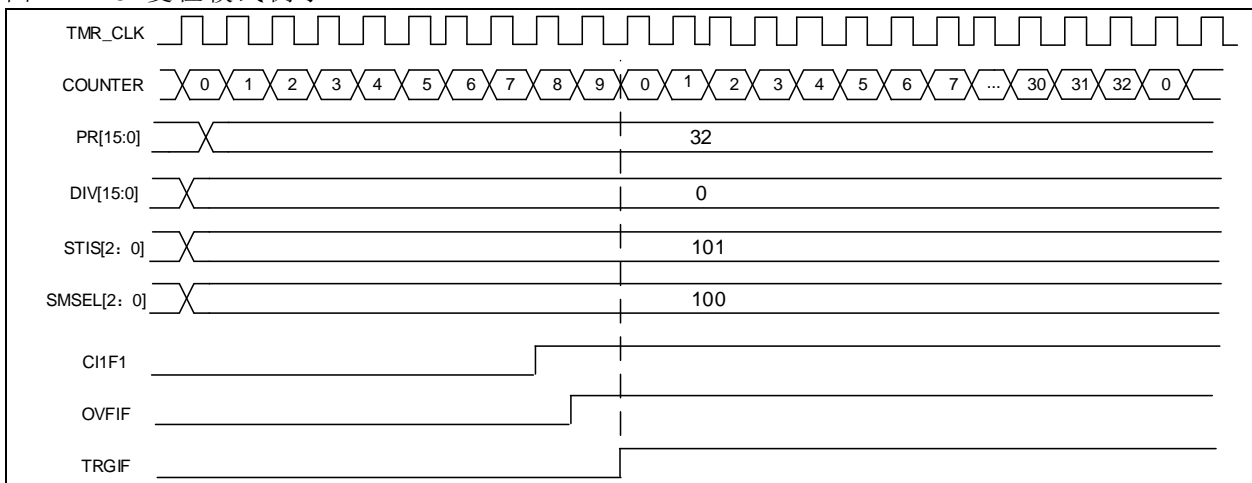
主次定时器之间可由内部连接信号进行同步。主定时器可由 PTOS[2: 0]位选择主定时器输出,即同步信息;次定时器由 SMSEL[2: 0]位选择从模式,即次定时器的工作模式。

定时器从模式有以下几种:

### 从模式: 复位模式

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

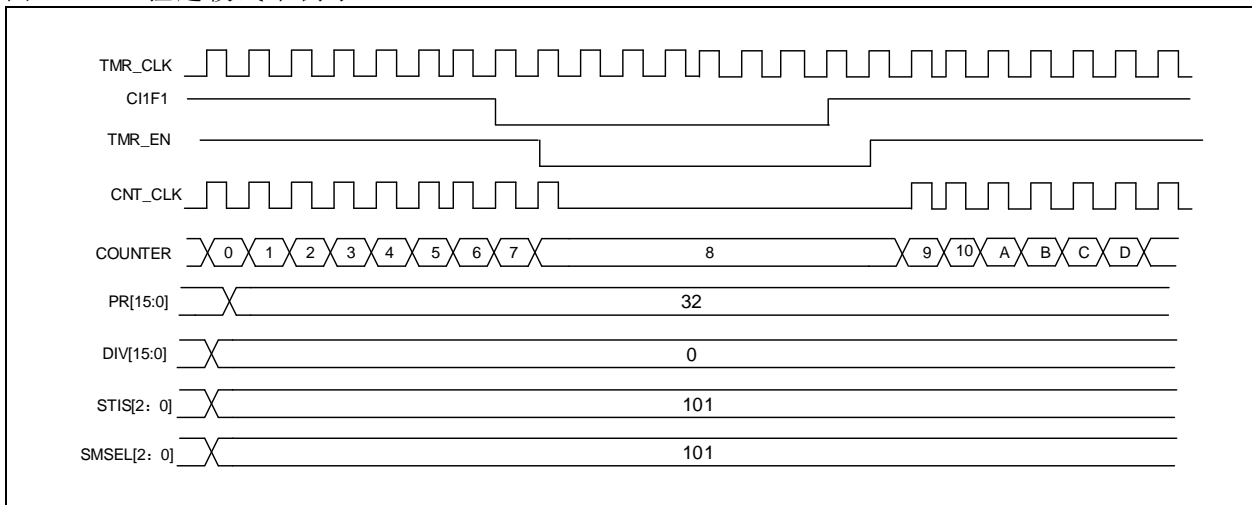
图 14-26 复位模式例子



### 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。

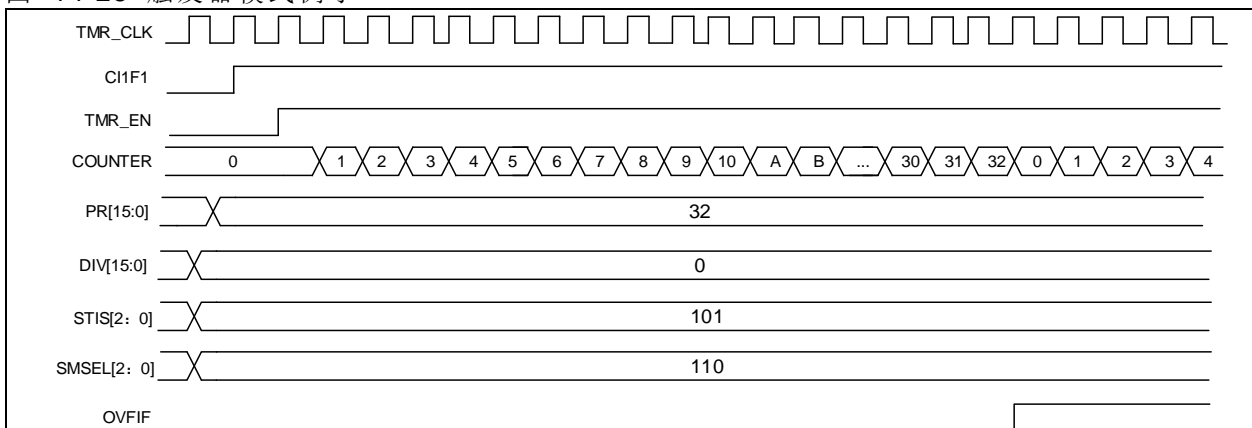
图 14-27 挂起模式下例子



### 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）。

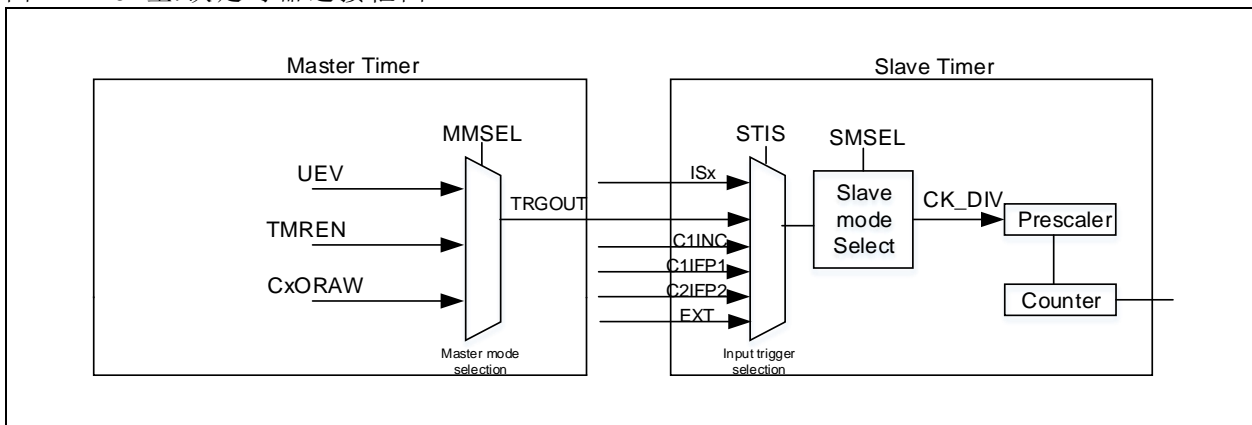
图 14-28 触发器模式例子



### 主/次定时器互联实例

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

图 14-29 主/次定时器连接框图



主定时器为次定时器提供时钟：

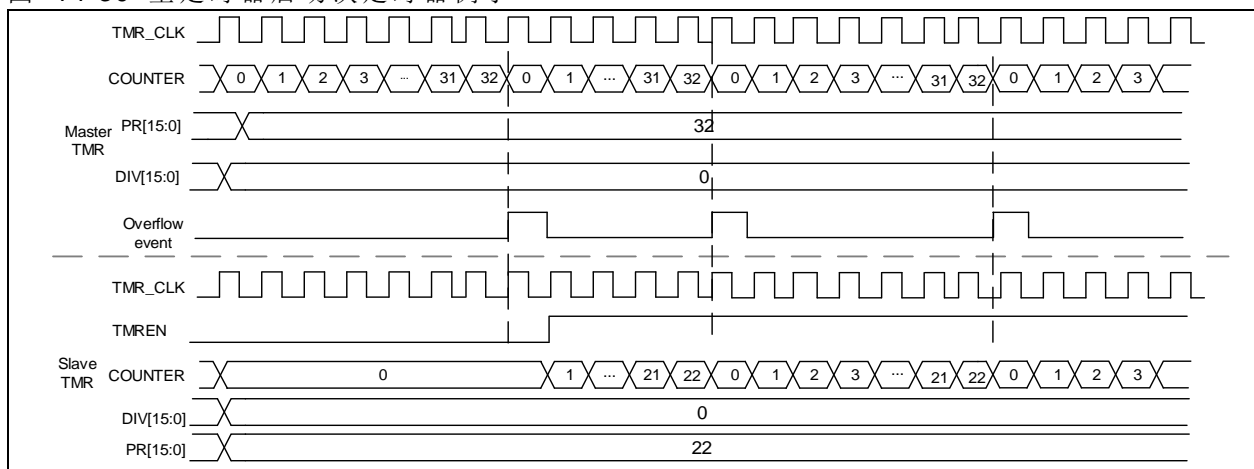
- 配置主定时器输出信号 TRGOUT 为溢出事件，配置 PTOS[2: 0]=3'b010，主定时器每次计数器溢出输出一个脉冲信号，用作次定时器计数时钟。
- 配置主定时器计数周期（周期寄存器（TMRx\_PR））。

- 配置次定时器触发输入信号 **TRGIN** 为主定时器输出（次定时器控制寄存器（**TMRx\_STCTRL**）的 **STIS[2: 0]**）。
- 配置次定时器使用外部时钟模式 **A**（次定时器控制寄存器（**TMRx\_STCTRL**）的 **SMSEL[2: 0]=3'b111**）。
- 将主定时器和次定时器的 **TMREN** 位置 1 启动定时器。

主定时器启动次定时器：

- 配置主定时器输出信号 **TRGOUT** 为溢出事件，配置 **PTOS[2: 0]=3'b010**，主定时器每次计数器溢出输出一个脉冲信号，用作次定时器计数时钟。
- 配置主定时器计数周期（周期寄存器（**TMRx\_PR**））。
- 配置次定时器触发输入 **TRGIN** 为主定时器输出。
- 配置次定时器为触发模式（**TMR2\_STCTRL** 寄存器的 **SMSEL=3'b110**）。
- 置主定时器 **TMREN=1** 以启动主定时器。

图 14-30 主定时器启动次定时器例子

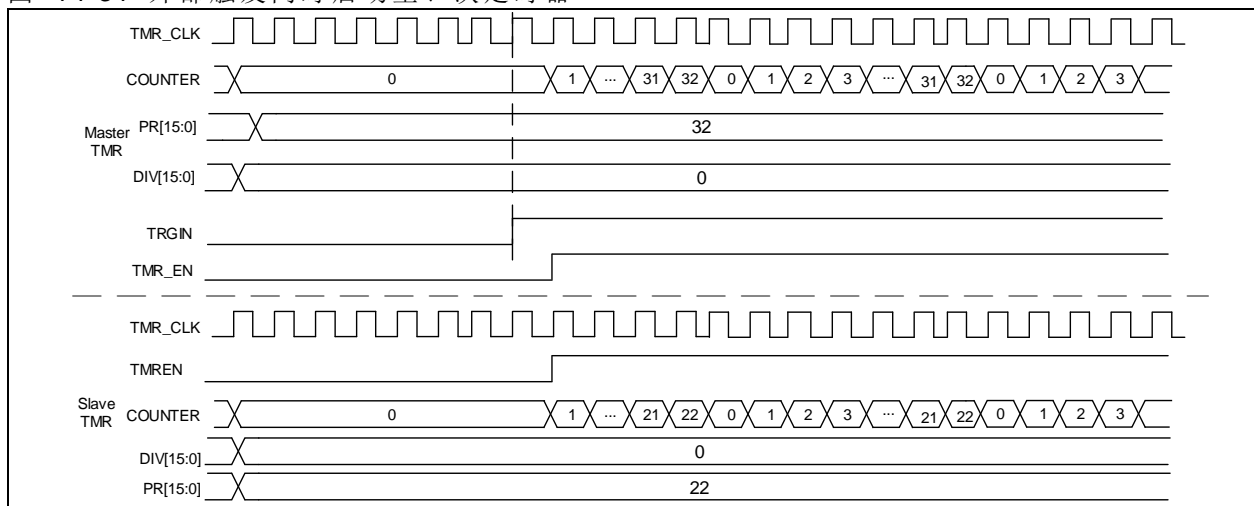


外部触发信号同步启动主、次定时器：

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（**TMR2\_STCTRL** 寄存器的 **SMSEL=3'b110**）。

图 14-31 外部触发同时启动主、次定时器



### 14.2.3.6 调试模式

当微控制器进入调试模式（Cortex™-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 TMRx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 TMRx 的所有寄存器映射到一个 16 位可寻址（编址）空间。

表 14-5 TMRx – 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
TMRx_CTRL2	0x04	0x0000
TMRx_STCTRL	0x08	0x0000
TMRx_IDEN	0x0C	0x0000
TMRx_ISTS	0x10	0x0000
TMRx_SWEVT	0x14	0x0000
TMRx_CM1	0x18	0x0000
TMRx_CM2	0x1C	0x0000
TMRx_CCTRL	0x20	0x0000
TMRx_CVAL	0x24	0x0000 0000
TMRx_DIV	0x28	0x0000
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
TMRx_DMA DT	0x4C	0x0000

#### 14.2.4.1 控制寄存器1（TMRx\_CTRL1）

域	简称	复位值	类型	功能
位 15: 11	保留	0x00	resd	保持默认值。
位 10	PMEN	0x0	rw	增强模式使能（Plus Mode Enable） 开启 TMRx 增强模式，该模式下 TMRx_CVAL，TMRx_PR，TMRx_CxDT 由 16 位扩展为 32 位。 0：关闭； 1：开启。 注：TMR2 和 TMR5 才具有此功能，其它 TMR 设置此位无效。
位 9: 8	CLKDIV	0x0	rw	时钟除频（Clock divider） 00：无除频； 01：2 除频； 10：4 除频； 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, 输出标志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, 输出标志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, 输出标志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) 该功能用于选择更新事件后, 计数器是否停止。 0: 关闭; 1: 开启。
位 2	OVFS	0x0	rw	溢出事件源选择 (Overflow event source) 配置溢出事件或 DMA 请求来源。 0: 来源于计数器溢出、设置 OVFSWTR 位或次定时器控制器产生的溢出事件; 1: 只能来源于计数器溢出。
位 1	OVFEN	0x0	rw	溢出事件使能 (Overflow event enable) 0: 开启; 1: 关闭。
位 0	TMREN	0x0	rw	使能定时器 (TMR enable) 0: 关闭; 1: 开启。

## 14.2.4.2 控制寄存器2 (TMRx\_CTRL2)

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

## 14.2.4.3 次定时器控制寄存器 (TMRx\_STCTRL)

域	简称	复位值	类型	功能
位 15	ESP	0x0	rw	外部信号极性 (External signal polarity) 用于选择外部方式。 0: 高电平或上升沿; 1: 低电平或下降沿。
位 14	ECMBEN	0x0	rw	外部时钟模式 B 使能 (External clock mode B enable) 用于启用外部时钟模式 B 0: 关闭; 1: 开启。
位 13: 12	ESDIV	0x0	rw	外部信号除频 (External signal divide) 用于选择降低外部触发频率的除频。 00: 关闭分频; 01: 2 分频; 10: 4 分频; 11: 8 分频。
位 11: 8	ESF	0x0	rw	外部信号滤波 (External signal filter) 用于过滤外部信号, 当外部信号产生了 N 次之后才能被采样。 0000: 无滤波器, 以 $f_{DTS}$ 采样 0001: $f_{SAMPLING} = f_{CK\_INT}$ , N=2; 0010: $f_{SAMPLING} = f_{CK\_INT}$ , N=4; 0011: $f_{SAMPLING} = f_{CK\_INT}$ , N=8; 0100: $f_{SAMPLING} = f_{DTS}/2$ , N=6; 0101: $f_{SAMPLING} = f_{DTS}/2$ , N=8; 0110: $f_{SAMPLING} = f_{DTS}/4$ , N=6; 0111: $f_{SAMPLING} = f_{DTS}/4$ , N=8; 1000: $f_{SAMPLING} = f_{DTS}/8$ , N=6; 1001: $f_{SAMPLING} = f_{DTS}/8$ , N=8; 1010: $f_{SAMPLING} = f_{DTS}/16$ , N=5; 1011: $f_{SAMPLING} = f_{DTS}/16$ , N=6; 1100: $f_{SAMPLING} = f_{DTS}/16$ , N=8; 1101: $f_{SAMPLING} = f_{DTS}/32$ , N=5; 1110: $f_{SAMPLING} = f_{DTS}/32$ , N=6; 1111: $f_{SAMPLING} = f_{DTS}/32$ , N=8。
位 7	STS	0x0	rw	次定时器同步 (Subordinate TMR synchronization) 该位开启后, 主次定时器可实现高度同步。 0: 关闭; 1: 开启。
位 6: 4	STIS	0x0	rw	次定时器输入选择 (Subordinate TMR input selection) 用于次定时器的输入选择。 000: 内部选择 0 (IS0); 001: 内部选择 1 (IS1); 010: 内部选择 2 (IS2); 011: 内部选择 3 (IS3); 100: C1IRAW 的输入检测器 (C1INC); 101: 滤波输入 1 (C1IF1); 110: 滤波输入 2 (C2IF2); 111: 外部输入 (EXT)。 关于每个定时器中 ISx 的细节, 参见表 14-3 <a href="#">表 14.5</a> 。
位 3	保留	0x0	resd	保持默认值。
位 2: 0	SMSEL	0x0	rw	次定时器模式选择 (Subordinate TMR mode selection) 000: 关闭从模式; 001: 编码模式 A; 010: 编码模式 B; 011: 编码模式 C; 100: 复位模式 - TRGIN 输入上升沿时, 重新初始化计数器;

101: 挂起模式 - TRGIN 输入高电平时, 计数器计数;  
 110: 触发模式 - TRGIN 输入上升沿时, 产生触发事件;  
 111: 外部时钟模式 A - TRGIN 输入上升沿时, 提供时钟;  
 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 DMA/中断使能寄存器 (TMRx\_IDEN)

域	简称	复位值	类型	功能
位 15	保留	0x0	resd	保持默认值。
位 14	TDEN	0x0	rw	触发 DMA 请求使能 (Trigger DMA request enable) 0: 关闭; 1: 开启。
位 13	保留	0x0	resd	保持默认值。
位 12	C4DEN	0x0	rw	通道 4 的 DMA 请求使能 (Channel 4 DMA request enable) 0: 关闭; 1: 开启。
位 11	C3DEN	0x0	rw	通道 3 的 DMA 请求使能 (Channel 3 DMA request enable) 0: 关闭; 1: 开启。
位 10	C2DEN	0x0	rw	通道 2 的 DMA 请求使能 (Channel 2 DMA request enable) 0: 关闭; 1: 开启。
位 9	C1DEN	0x0	rw	通道 1 的 DMA 请求使能 (Channel 1 DMA request enable) 0: 关闭; 1: 开启。
位 8	OVFDEN	0x0	rw	溢出事件的 DMA 请求使能 (overflow event DMA request enable) 0: 关闭; 1: 开启。
位 7	保留	0x0	resd	保持默认值。
位 6	TIEN	0x0	rw	触发中断使能 (Trigger interrupt enable) 0: 关闭; 1: 开启。
位 5	保留	0x0	resd	保持默认值。
位 4	C4IEN	0x0	rw	通道 4 中断使能 (Channel 4 interrupt enable) 0: 关闭; 1: 开启。
位 3	C3IEN	0x0	rw	通道 3 中断使能 (Channel 3 interrupt enable) 0: 关闭; 1: 开启。
位 2	C2IEN	0x0	rw	通道 2 中断使能 (Channel 2 interrupt enable) 0: 关闭; 1: 开启。
位 1	C1IEN	0x0	rw	通道 1 中断使能 (Channel 1 interrupt enable) 0: 关闭; 1: 开启。
位 0	OVFIEN	0x0	rw	溢出中断使能 (overflow interrupt enable) 0: 关闭; 1: 开启。

## 14.2.4.5 中断状态寄存器 (TMRx\_ISTS)

域	简称	复位值	类型	功能
位 15: 13	保留	0x0	resd	保持默认值。
位 12	C4RF	0x0	rw0c	通道 4 再捕获标记 (Channel 4 recapture flag) 见 C1RF 的描述。
位 11	C3RF	0x0	rw0c	通道 3 再捕获标记 (Channel 3 recapture flag) 见 C1RF 的描述。
位 10	C2RF	0x0	rw0c	通道 2 再捕获标记 (Channel 2 recapture flag) 见 C1RF 的描述。
位 9	C1RF	0x0	rw0c	通道 1 再捕获标记 (Channel 1 recapture flag) C1IF 的状态已经为'1'时是否再次发生了捕获, 由硬件置'1', 写'0'清除。 0: 无捕获发生; 1: 捕获发生。
位 8: 7	保留	0x0	resd	保持默认值。
位 6	TRGIF	0x0	rw0c	触发中断标记 (Trigger interrupt flag) 当发生触发事件时由硬件置'1', 写'0'清除。 0: 无触发事件发生; 1: 发生触发事件。 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 OVFE=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 软件事件寄存器 (TMRx\_SWEVT)

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

位 3	C3SWTR	0x0	wo	软件触发通道 3 事件 (Channel 3 event triggered by software) 见 C1M 的描述。
位 2	C2SWTR	0x0	wo	软件触发通道 2 事件 (Channel 2 event triggered by software) 见 C1M 的描述。
位 1	C1SWTR	0x0	wo	软件触发通道 1 事件 (Channel 1 event triggered by software) 通过软件触发一个通道 1 事件。 0: 无作用; 1: 制造一个通道 1 事件。
位 0	OVFSWTR	0x0	wo	软件触发溢出事件 (Overflow event triggered by software) 通过软件触发一个溢出事件。 0: 无作用; 1: 制造一个溢出事件。

## 14.2.4.7 通道模式寄存器1 (TMRx\_CM1)

输出比较模式:

域	简称	复位值	类型	功能
位 15	C2OSEN	0x0	rw	通道 2 输出开关使能 (Channel 2 output switch enable)
位 14: 12	C2OCTRL	0x0	rw	通道 2 输出控制 (Channel 2 output control)
位 11	C2OBEN	0x0	rw	通道 2 输出缓存使能 (Channel 2 output buffer enable)
位 10	C2OIEN	0x0	rw	通道 2 输出立即使能 (Channel 2 output immediately enable)
位 9: 8	C2C	0x0	rw	通道 2 配置 (Channel 2 configure) 当 C2EN='0'时, 这些位用于选择通道 2 为输出或输入, 以及输入时的映射选择: 00: 输出; 01: 输入, C2IN 映射在 C2IRAW 上; 10: 输入, C2IN 映射在 C1IRAW 上; 11: 输入, C2IN 映射在 STCI 上, 只有在 STIS 选择内部触发输入时才工作。
位 7	C1OSEN	0x0	rw	通道 1 输出开关使能 (Channel 1 output switch enable) 0: EXT 输入不影响 C1ORAW; 1: 当 EXT 输入高电平时, 将 C1ORAW 清 0。
位 6: 4	C1OCTRL	0x0	rw	通道 1 输出控制 (Channel 1 output control) 这些位用于设置原始信号 C1ORAW 的工作状态。 000: 断开。断开 C1ORAW 到 C1OUT 的输出; 001: 设置 C1ORAW 为高: TMRx_CVAL=TMRx_C1DT 时。 010: 设置 C1ORAW 为低: TMRx_CVAL=TMRx_C1DT 时。 011 : 切换 C1ORAW 的电平: 当 TMRx_CVAL=TMRx_C1DT 时。 100: 固定 C1ORAW 为低。 101: 固定 C1ORAW 为高。 110: PWM 模式 A — OWCDIR=0, 若 TMRx_C1DT>TMRx_CVAL 时设置 C1ORAW 为高, 否则为低; — OWCDIR=1, 若 TMRx_C1DT <TMRx_CVAL 时设置 C1ORAW 为低, 否则为高。 111: PWM 模式 B — OWCDIR=0, 若 TMRx_C1DT >TMRx_CVAL 时设置 C1ORAW 为低, 否则为高; — OWCDIR=1, 若 TMRx_C1DT <TMRx_CVAL 时设置 C1ORAW 为高, 否则为低。 注: 除'000'外, 其余配置下 C1OUT 将连接到 C1ORAW, C1OUT 的输出电平除了会根据 C1ORAW 变化外, 还与 CCTRL 所配置的输出极性有关。
位 3	C1OBEN	0x0	rw	通道 1 输出缓存使能 (Channel 1 output buffer enable)

				0: 关闭 TMRx_C1DT 的缓存功能, 写入 TMRx_C1DT 的内容会立即生效。 1: 启用 TMRx_C1DT 的缓存功能, 写入 TMRx_C1DT 的内容将保存到缓存寄存器中, 当发生溢出事件时再更新到 TMRx_C1DT 中。
位 2	C1OEN	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 C1IRAW 上; 10: 输入, C1IN 映射在 C2IRAW 上; 11: 输入, C1IN 映射在 STCI 上, 只有在 STIS 选择内部触发输入时才工作。
<b>输入模式:</b>				
域	简称	复位值	类型	功能
位 15: 12	C2DF	0x0	rw	通道 2 滤波器 (Channel 2 digital filter)
位 11: 10	C2IDIV	0x0	rw	通道 2 分频系数 (Channel 2 input divider)
位 9: 8	C2C	0x0	rw	通道 2 配置 (Channel 2 configure) 当 C2EN='0'时, 这些位用于选择通道 2 为输出或输入, 以及输入时的映射选择: 00: 输出; 01: 输入, C2IN 映射在 C2IRAW 上; 10: 输入, C2IN 映射在 C1IRAW 上; 11: 输入, C2IN 映射在 STCI 上, 只有在 STIS 选择内部触发输入时才工作。
位 7: 4	C1DF	0x0	rw	通道 1 滤波器 (Channel 1 digital filter) 这些位用于配置通道 1 的滤波器。滤波的个数为 N, 则表示发生了 N 次采样事件后输入边沿才能通过滤波器: 0000: 无滤波器, 以 $f_{DTS}$ 采样 1000: 采样频率 $f_{SAMPLING}=f_{DTS}/8$ , N=6 0001: 采样频率 $f_{SAMPLING}=f_{CK\_INT}$ , N=2 1001: 采样频率 $f_{SAMPLING}=f_{DTS}/8$ , N=8 0010: 采样频率 $f_{SAMPLING}=f_{CK\_INT}$ , N=4 1010: 采样频率 $f_{SAMPLING}=f_{DTS}/16$ , N=5 0011: 采样频率 $f_{SAMPLING}=f_{CK\_INT}$ , N=8 1011: 采样频率 $f_{SAMPLING}=f_{DTS}/16$ , N=6 0100: 采样频率 $f_{SAMPLING}=f_{DTS}/2$ , N=6 1100: 采样频率 $f_{SAMPLING}=f_{DTS}/16$ , N=8 0101: 采样频率 $f_{SAMPLING}=f_{DTS}/2$ , N=8 1101: 采样频率 $f_{SAMPLING}=f_{DTS}/32$ , N=5 0110: 采样频率 $f_{SAMPLING}=f_{DTS}/4$ , N=6 1110: 采样频率 $f_{SAMPLING}=f_{DTS}/32$ , N=6 0111: 采样频率 $f_{SAMPLING}=f_{DTS}/4$ , N=8 1111: 采样频率 $f_{SAMPLING}=f_{DTS}/32$ , N=8
位 3: 2	C1IDIV	0x0	rw	通道 1 分频系数 (Channel 1 input divider) 这些位定义了通道 1 的分频系数。 00: 不分频, 每一个有效的边沿都会产生一次输入; 01: 每 2 个有效的边沿产生一次输入; 10: 每 4 个有效的边沿产生一次输入; 11: 每 8 个有效的边沿产生一次输入。 注: C1EN='0'时, 分频系数复位。
位 1: 0	C1C	0x0	rw	通道 1 配置 (Channel 1 configure)

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

00：输出；

01：输入，C1IN 映射在 C1IRAW 上；

10：输入，C1IN 映射在 C2IRAW 上；

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

#### 14.2.4.8 通道模式寄存器2 (TMRx\_CM2)

输出比较模式：

域	简称	复位值	类型	功能
位 15	C4OSEN	0x0	rw	通道 4 输出开关使能 (Channel 4 output switch enable)
位 14: 12	C4OCTRL	0x0	rw	通道 4 输出控制 (Channel 4 output control)
位 11	C4OBEN	0x0	rw	通道 4 输出缓存使能 (Channel 4 output buffer enable)
位 10	C4OIEN	0x0	rw	通道 4 输出立即使能 (Channel 4 output immediately enable)
通道 4 配置 (Channel 4 configure)				
当 C4EN='0'时，这些位用于选择通道 4 为输出或输入，以及输入时的映射选择：				
00：输出；				
01：输入，C4IN 映射在 C4IRAW 上；				
10：输入，C4IN 映射在 C3IRAW 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 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 C3IRAW 上；				
10：输入，C3IN 映射在 C4IRAW 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 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 C4IRAW 上；				
10：输入，C4IN 映射在 C3IRAW 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 C3IRAW 上；				
10：输入，C3IN 映射在 C4IRAW 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	



## 14.2.4.9 通道控制寄存器 (TMRx\_CTRL)

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: 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: 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: 2	保留	0x0	resd	保持默认值。
位 1	C1P	0x0	rw	通道 1 极性 (Channel 1 polarity) 通道 1 配置为输出: 0: C1OUT 的有效电平为高 1: C1OUT 的有效电平为低 通道 1 配置为输入: 0: C1IN 的有效边沿为上升沿; 作为外部触发使用时, C1IN 不反相。 1: C1IN 的有效边沿为下降沿; 作为外部触发使用时, C1IN 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 14-6 标准CxOUT通道的输出控制位

CxEN 位	CxOUT 输出状态
0	禁止输出 (CxOUT=0, Cx_EN=0)
1	CxOUT = CxORAW + 极性, Cx_EN=1

注意: 连接到标准 CxOUT 通道的外部 I/O 管脚状态, 取决于 CxOUT 通道状态和 GPIO 以及 IOMUX 寄存器。

## 14.2.4.10 计数值 (TMRx\_CVAL)

域	简称	复位值	类型	功能
位 31: 16	CVAL	0x0000	rw	计数值 (Counter value) 当 TMR2 或 TMR5 开启增强模式时 (TMR_CTRL1 中的 PMEN 位), CVAL 被扩展为 32 位。
位 15: 0	CVAL	0x0000	rw	计数值 (Counter value)

## 14.2.4.11 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。

## 14.2.4.12 周期寄存器 (TMRx\_PR)

域	简称	复位值	类型	功能
位 31: 16	PR	0x0000	rw	周期值 (Period value) 当 TMR2 或 TMR5 开启增强模式时 (TMR_CTRL1 中的 PMEN 位), PR 被扩展为 32 位。
位 15: 0	PR	0x0000	rw	周期值 (Period value) 定时器计数的周期值。当周期值为 0 时, 定时器不工作。

## 14.2.4.13 通道1数据寄存器 (TMRx\_C1DT)

域	简称	复位值	类型	功能
位 31: 16	C1DT	0x0000	rw	通道 1 数据寄存器值 (Channel 1 data register) 当 TMR2 或 TMR5 开启增强模式时 (TMR_CTRL1 中的 PMEN 位), C1DT 被扩展为 32 位。
位 15: 0	C1DT	0x0000	rw	通道 1 数据寄存器值 (Channel 1 data register) 若通道 1 配置为输入: C1DT 是前一次通道 1 输入事件 (C1IN) 所保存的 CVAL。 若通道 1 配置为输出: C1DT 是将要和 CVAL 进行比较的值, 写入的值是否会立即生效取决于输出缓存使能位 (C1OBEN), 并根据设置在 C1OUT 上产生相应的输出。

## 14.2.4.14 通道2数据寄存器 (TMRx\_C2DT)

域	简称	复位值	类型	功能
位 31: 16	C2DT	0x0000	rw	通道 2 数据寄存器值 (Channel 2 data register) 当 TMR2 或 TMR5 开启增强模式时 (TMR_CTRL1 中的 PMEN 位), C2DT 被扩展为 32 位。
位 15: 0	C2DT	0x0000	rw	通道 2 数据寄存器值 (Channel 2 data register) 若通道 1 配置为输入: C2DT 是前一次通道 2 输入事件 (C2IN) 所保存的 CVAL。 若通道 2 配置为输出: C2DT 是将要和 CVAL 进行比较的值, 写入的值是否会立即生效取决于输出缓存使能位 (C2OBEN), 并根据设置在 C2OUT 上产生相应的输出。

## 14.2.4.15 通道3数据寄存器 (TMRx\_C3DT)

域	简称	复位值	类型	功能
位 31: 16	C3DT	0x0000	rw	通道 3 数据寄存器值 (Channel 3 data register) 当 TMR2 或 TMR5 开启增强模式时 (TMR_CTRL1 中的 PMEN 位), C3DT 被扩展为 32 位。
位 15: 0	C3DT	0x0000	rw	通道 3 数据寄存器值 (Channel 3 data register) 若通道 3 配置为输入: C3DT 是前一次通道 3 输入事件 (C3IN) 所保存的 CVAL。 若通道 3 配置为输出: C3DT 是将要和 CVAL 进行比较的值, 写入的值是否会立即生效取决于输出缓存使能位 (C3OBEN), 并根据设置在 C3OUT 上产生相应的输出。

## 14.2.4.16 通道4数据寄存器 (TMRx\_C4DT)

域	简称	复位值	类型	功能
位 31: 16	C4DT	0x0000	rw	通道 4 数据寄存器值 (Channel 4 data register) 当 TMR2 或 TMR5 开启增强模式时 (TMR_CTRL1 中的 PMEN 位), C4DT 被扩展为 32 位。
位 15: 0	C4DT	0x0000	rw	通道 4 数据寄存器值 (Channel 4 data register) 若通道 4 配置为输入: C4DT 是前一次通道 4 输入事件 (C4IN) 所保存的 CVAL。 若通道 4 配置为输出: C4DT 是将要和 CVAL 进行比较的值, 写入的值是否会立即生效取决于输出缓存使能位 (C4OBEN), 并根据设置在 C4OUT 上产生相应的输出。

## 14.2.4.17 DMA控制寄存器 (TMRx\_DMACTRL)

域	简称	复位值	类型	功能
位 15: 13	保留	0x0	resd	保持默认值。
位 12: 8	DTB	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 DMA数据寄存器 (TMRx\_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 通用定时器（TMR9到TMR14）

### 14.3.1 TMRx简介

通用定时器 TMR9 到 TMR14 支持 16 位向上计数，可通过同步功能进行互联。

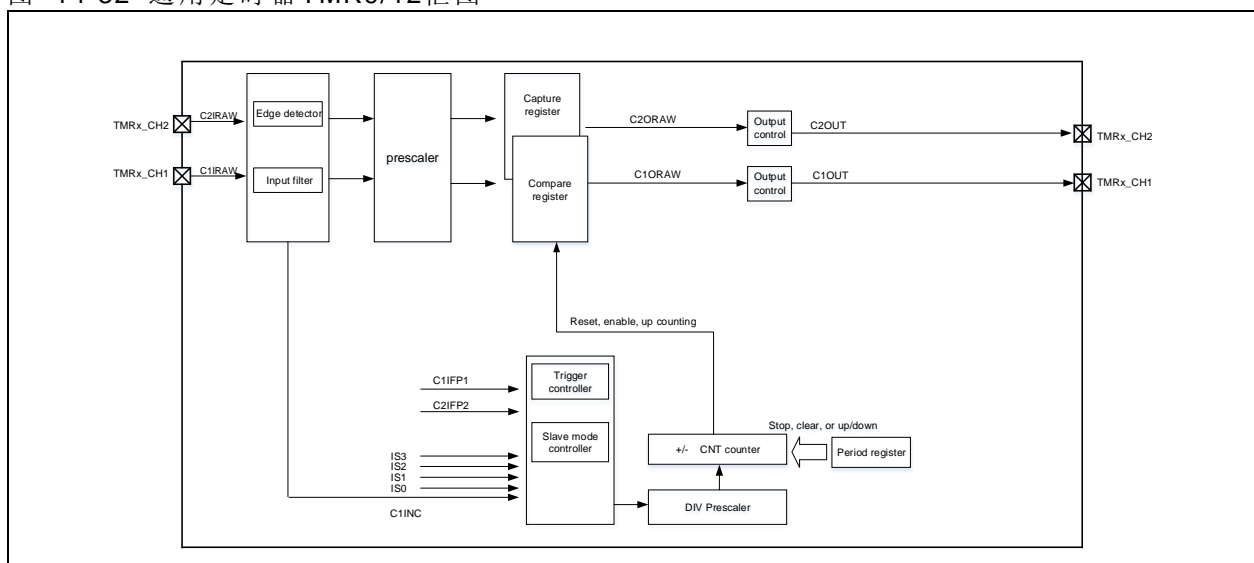
### 14.3.2 TMRx主要特性

#### 14.3.2.1 TMR9和TMR12主要特性

TMR9 和 TMR12 功能包括：

- 可选内部、外部输入用作计数时钟
- 16 位向上计数器
- 2 组独立通道，支持输入捕获、输出比较、PWM 生成、单周期模式
- 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

图 14-32 通用定时器TMR9/12框图

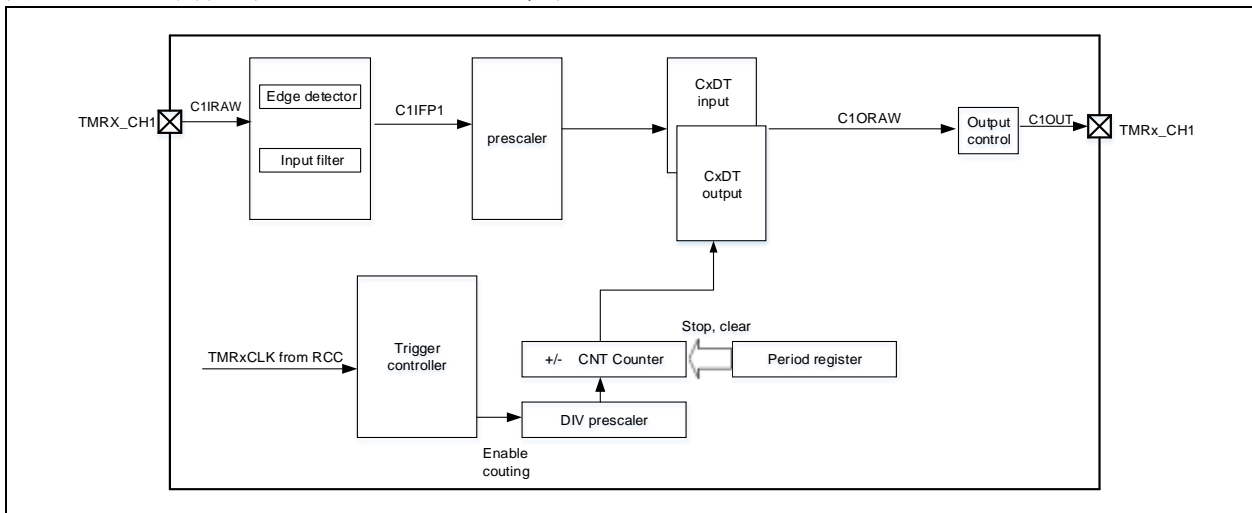


#### 14.3.2.2 TMR10、TMR11、TMR13和 TMR14主要特性

通用 TMRx（TMR10、TMR11、TMR13、TMR14）定时器功能包括：

- 由内部用作计数时钟
- 16 位向上计数器
- 1 组独立通道，支持输入捕获、输出比较、PWM 生成
- 定时器之间可互联同步
- 支持溢出事件、通道事件触发中断

图 14-33 通用定时器TMR10/11/13/14框图



### 14.3.3 TMRx功能描述

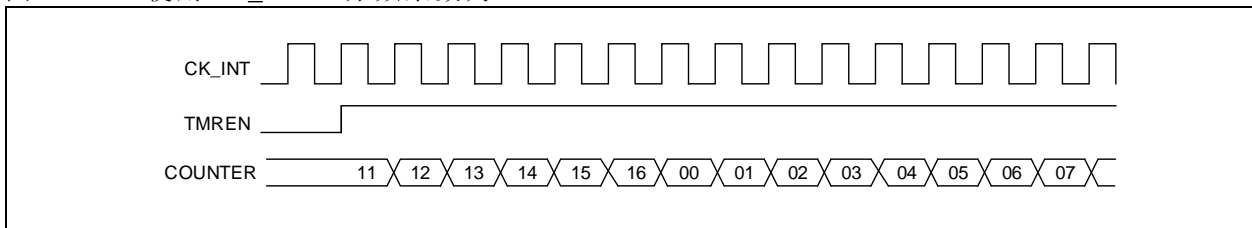
#### 14.3.3.1 计数时钟

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

##### 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。

图 14-34 使用CK\_INT且分频系数为1

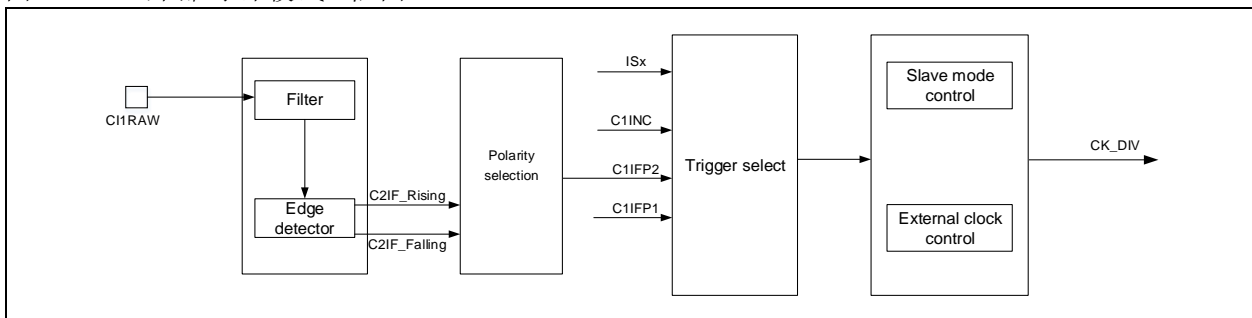


##### 外部时钟（仅 TMR9/12）

计数时钟可由选择的 TRGIN 信号提供。

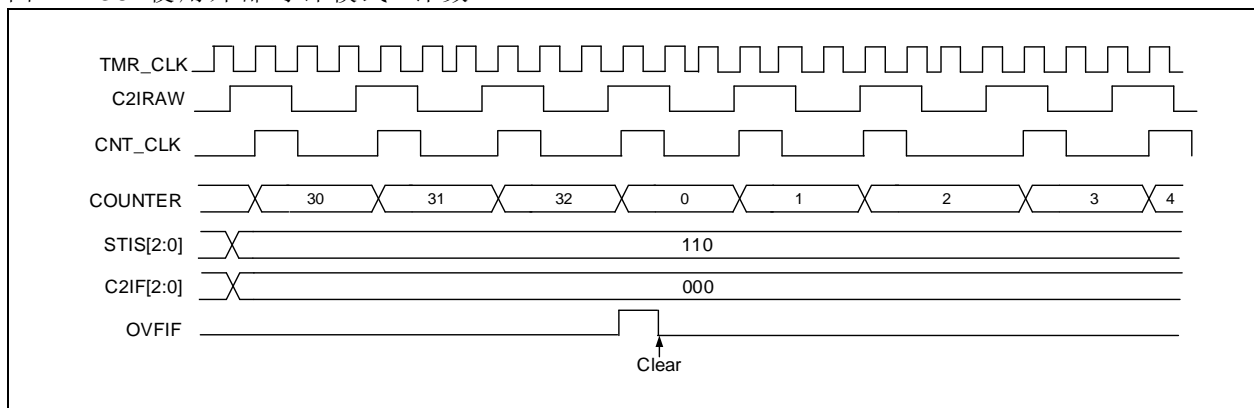
当 SMSEL=3'111 时，外部时钟模式 A 被选中，配置 STIS[2: 0]来选择 TRGIN 信号驱动计数器计数。

图 14-35 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 14-36 使用外部时钟模式A计数



### 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。

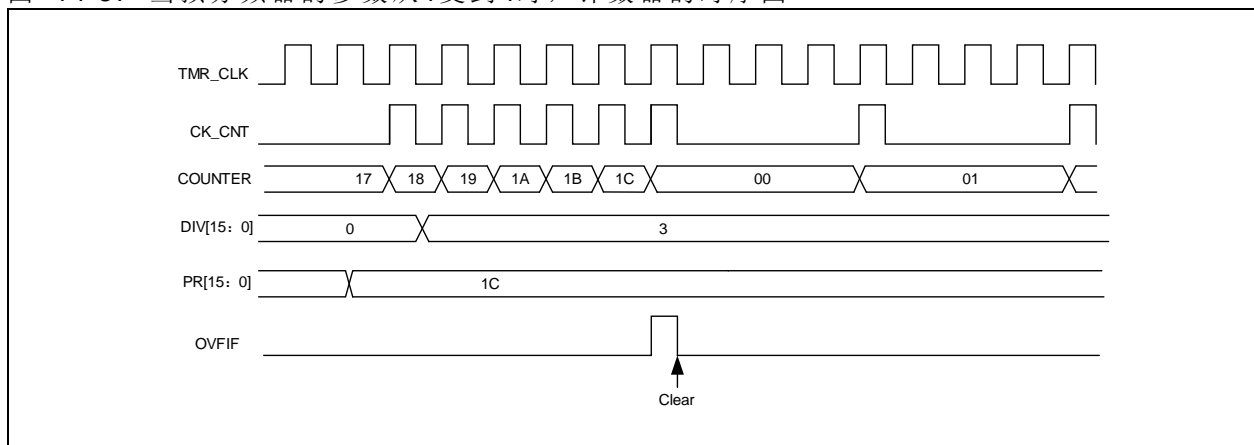
高级定时器内含一个 16 位预分频器, 用于产生驱动计数器计数的时钟 CK\_CNT, 通过配置 TMRx\_DIV 寄存器值, 可灵活调整 CK\_CNT 与 TMR\_CLK 之间的分频关系。预分频值可在任何时刻修改, 但只在下一个溢出事件发生时, 新值才会生效。

表 14-7 TMRx内部触发连接

次定时器	IS0 (STIS = 000)	IS1 (STIS = 001)	IS2 (STIS = 010)	IS3 (STIS = 011)
TMR9	TMR2_TRGOUT	TMR3_TRGOUT	TMR10_OC	TMR11_OC
TMR12	TMR4_TRGOUT	TMR5_TRGOUT	TMR13_OC	TMR14_OC

注意: 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 14-37 当预分频器的参数从1变到4时, 计数器的时序图



### 14.3.3.2 计数模式

通用定时器仅提供向上计数模式, 其内部拥有一个支持 16 位计数的计数器, 计数器的值可由周期寄存器 (TMRx\_PR) 载入。默认下, 周期寄存器 (TMRx\_PR) 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), 周期寄存器 (TMRx\_PR) 值会在溢出事件发生时传入它的影子寄存器。溢出事件由 OVFE、OVFS 位配置。

将 TMREN 位置 1 可使得定时器计数, 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。

#### 向上计数模式

向上计数模式中, 当计数值达到 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 OVFIIF 位置 1。若禁止产生溢出事件, 则计数器溢出后不再重载预分频值和重载值, 否则预分频值和重载值将在溢出事件后更新。

图 14-38 PRBEN=0时的溢出事件

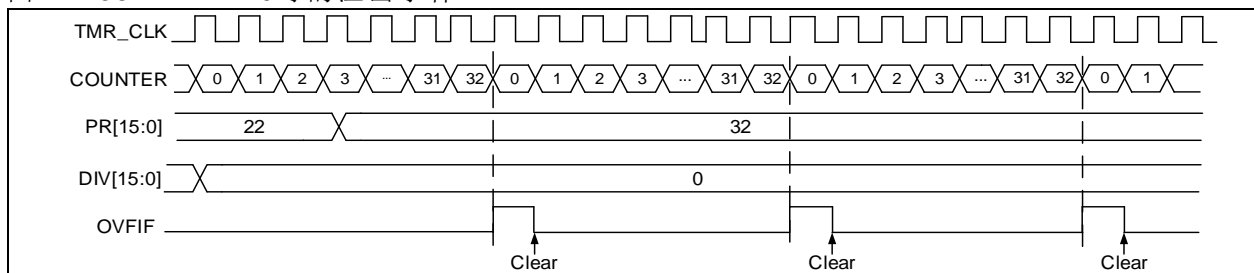
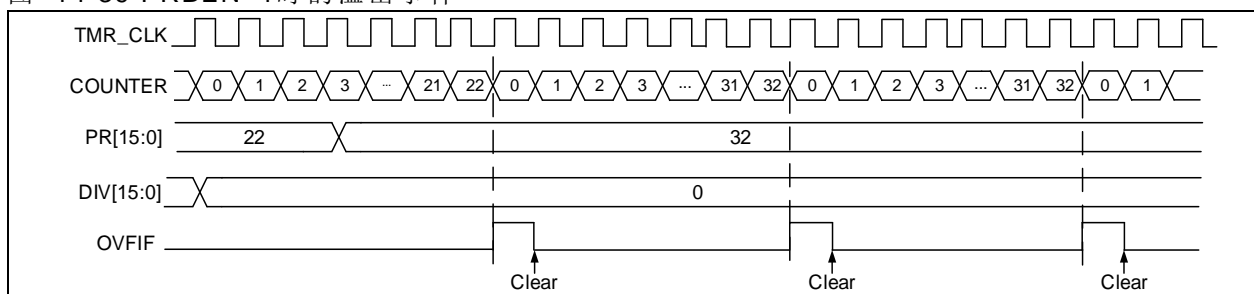


图 14-39 PRBEN=1时的溢出事件



### 14.3.3.3 TMR输入部分

TMR9 和 12 拥有两个独立通道，TMR10、11、13、14 拥有一个独立通道。每个通道可配置为输入或输出，当配置位输入时，可用于对输入信号的滤波、选择、分频和输入捕获功能。

图 14-40 输入/输出通道1的主电路

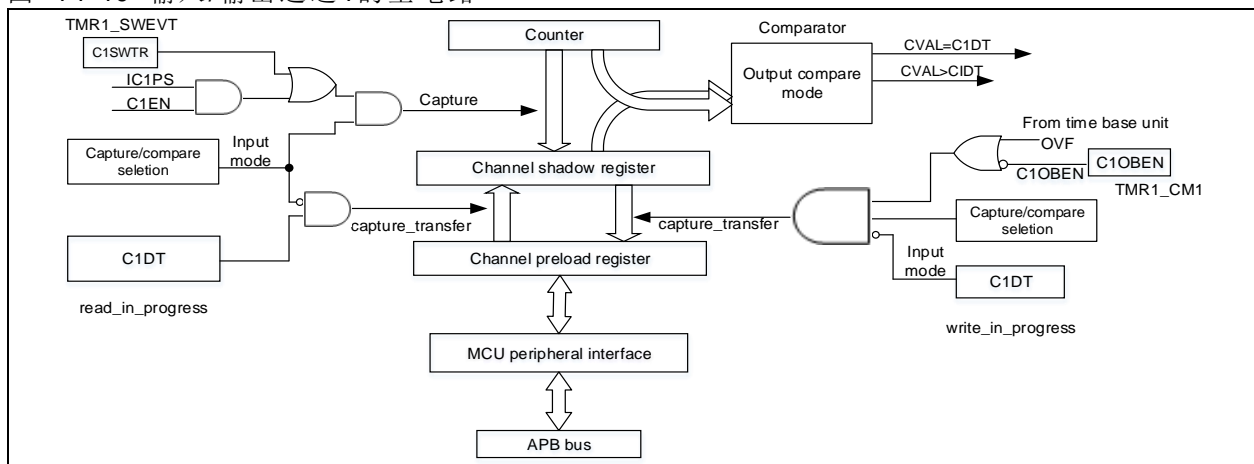
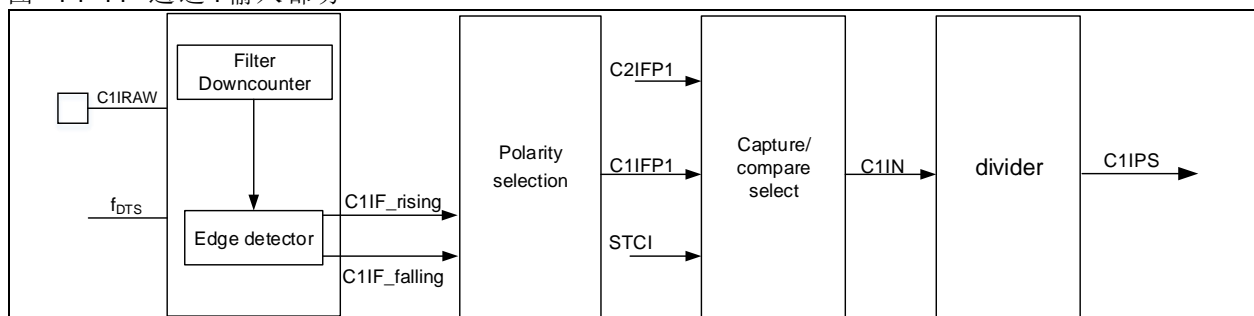


图 14-41 通道1输入部分



#### 输入模式

此模式下，当选中的触发信号被检测到时，通道寄存器（TMRx\_CxDT）会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）则产生相应的中断请求。若在 CxIF 已置 1 后检测到选中的触发信号，则将 CxOF 位置 1。

若要捕获 C1IN 输入的上升沿，可按如下进行配置：

- 将通道寄存器（TMRx\_CxDT）中的 C1C 位配置为 01，选择 C1IN 作为通道 1 输入。
- 配置 C1IN 信号滤波器带宽（CxDF[3: 0]）。

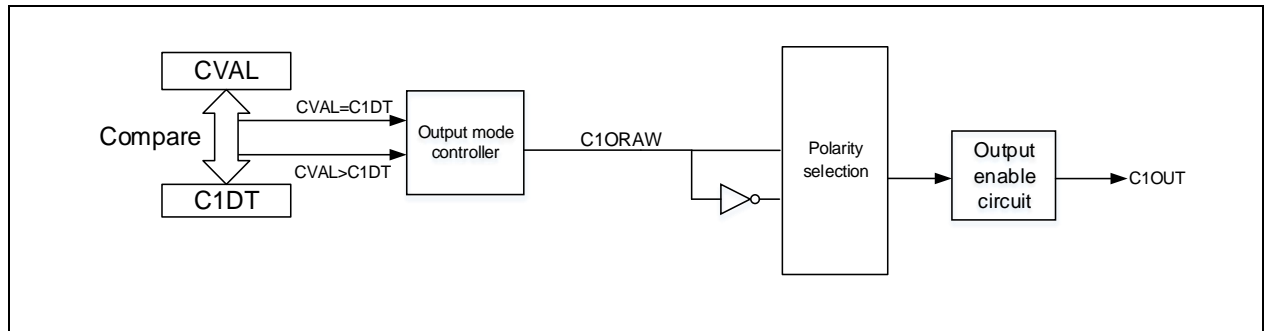


- 配置 C1IN 通道的有效沿，在通道控制寄存器（TMRx\_CCTRL）中写入 C1P=0（上升沿）。
- 配置 C1IN 信号捕获频率（C1DIV[1: 0]）。
- 使能通道 1 输入捕获（C1EN=1）。
- 根据需要设置 DMA/中断使能寄存器（TMRx\_IDEN）中的 C1IEN 位，选择中断请求。

#### 14.3.3.4 TMR输出部分

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

图 14-42 捕获/比较通道的输出部分（通道 1）



##### 输出模式

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

输出比较模式有以下子类：

- **PWM 模式：**CxOCTRL=3'b110/111 时，开启 PWM 模式，每个通道可独立配置并输出一路 PWM 信号。此时，输出信号的周期由 TMRx\_PR 配置，占空比由 CxDT 值配置，计数器值与通道寄存器（TMRx\_CxDT）值进行比较，根据计数方向输出指定电平信号，关于 PWM 模式 A/B 详见 CxOCTRL[2: 0]位描述。当计数模式为中央双向对齐计数时，可根据 OWCDIR 位指示计数方向。
- **强制输出模式：**CxOCTRL=3'b100/101 时，开启强制输出模式。此时，CxORAW 信号的电平被强制输出为配置的电平，而与计数值无关。虽然输出信号不依赖于比较结果，但通道标志位和 DMA 请求仍依赖于比较结果。
- **输出比较模式：**CxOCTRL=2'b001/010/011 时，开启输出比较模式。此时，当计数值与 CxDT 值匹配时，CxORAW 被强制为高电平、低电平或进行电平翻转。
- **单周期模式(仅 TMR9/12)：**PWM 模式的特例，将 OCMEN 位置 1 可开启单周期模式，此模式下，仅在当前计数周期中进行比较匹配，完成当前计数后，TMREN 位清 0，因此仅输出一个脉冲。当配置为向上计数模式时，需要严格配置  $CVAL < CxDT \leq PR$ ；向下计数时，需严格配置  $CVAL > CxDT$ 。
- **快速输出模式(仅 TMR9/12)：**将 CxOIEN 位置 1 可开启此功能，开启后 CxORAW 电平值不再在计数值与 CxDT 匹配时变化，而是在当前计数周期开始时，也就是说，比较结果被提前了，计数器值与通道寄存器（TMRx\_CxDT）的比较结果将会提前决定 CxORAW 的电平。

图 14-43 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 14-44 展示了计数器向上计数与 PWM 模式 A 配合的例子，PR=0x32，CxDT 配置为不同的值时输出时输出信号的翻转情况。

图 14-45 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 14-43 计数值与C1DT值匹配时翻转C1ORAW

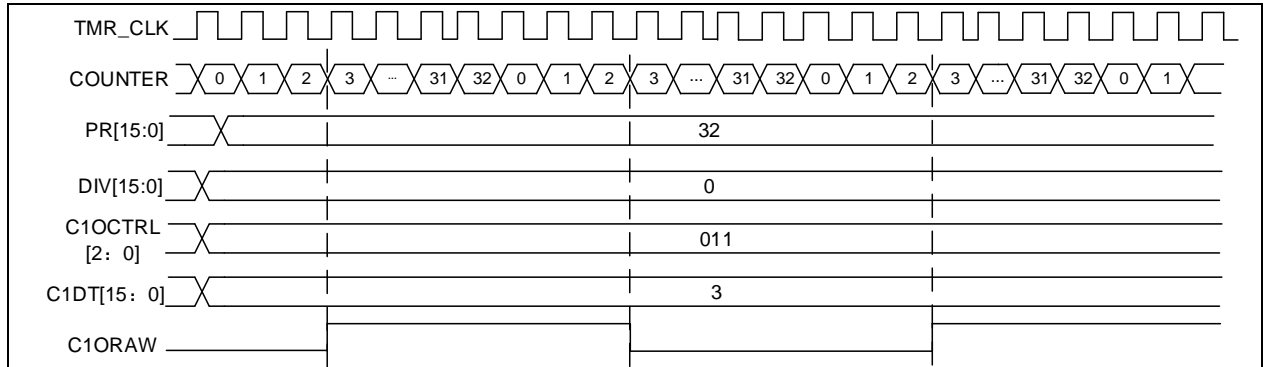


图 14-44 向上计数下PWM模式A

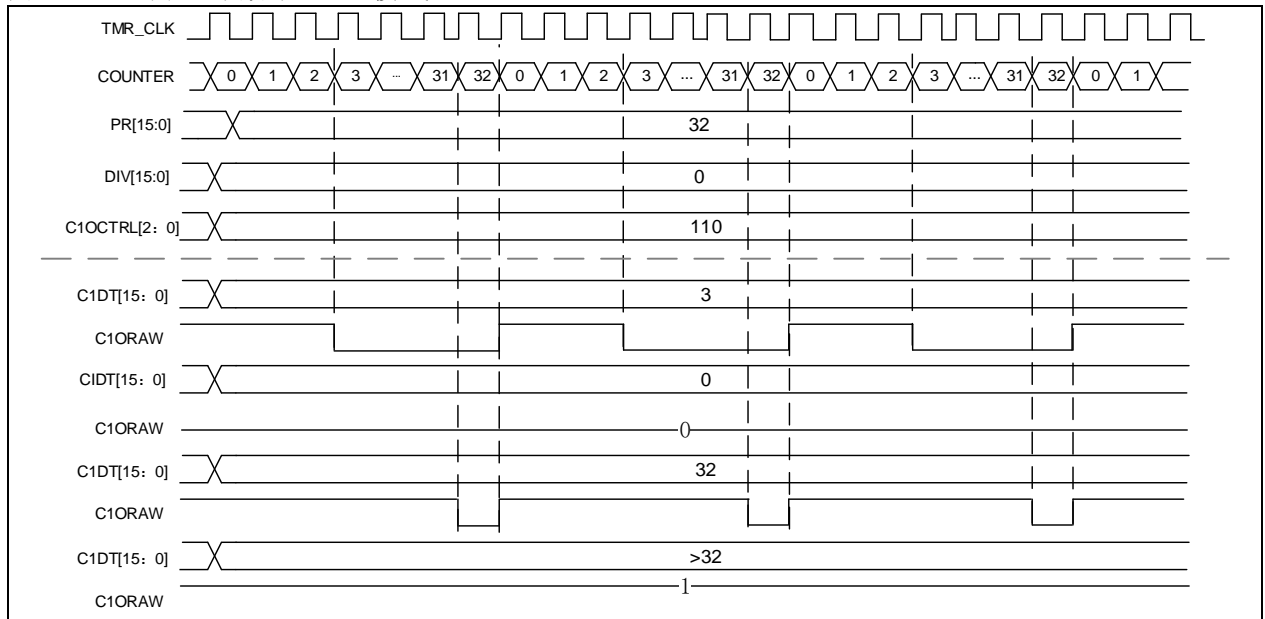
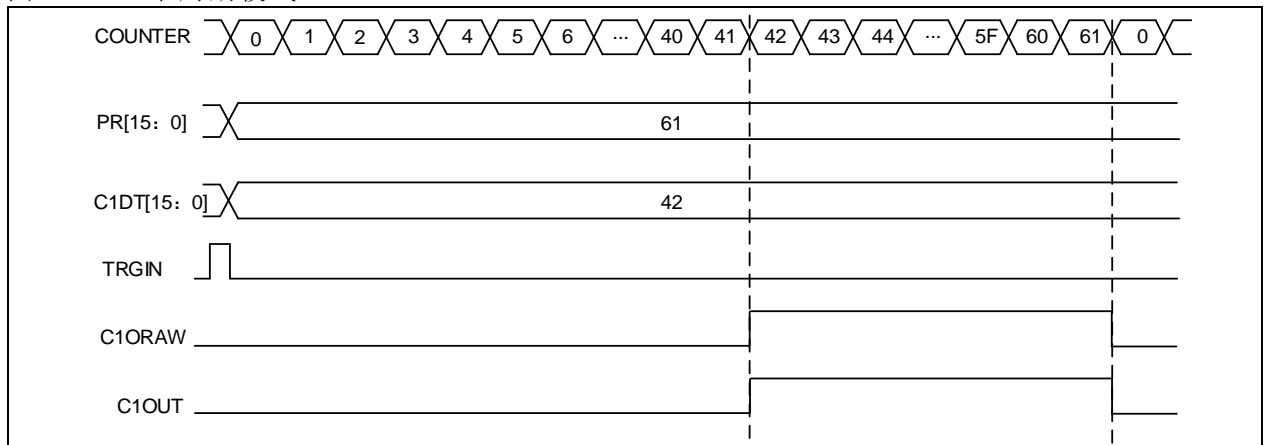


图 14-45 单周期模式



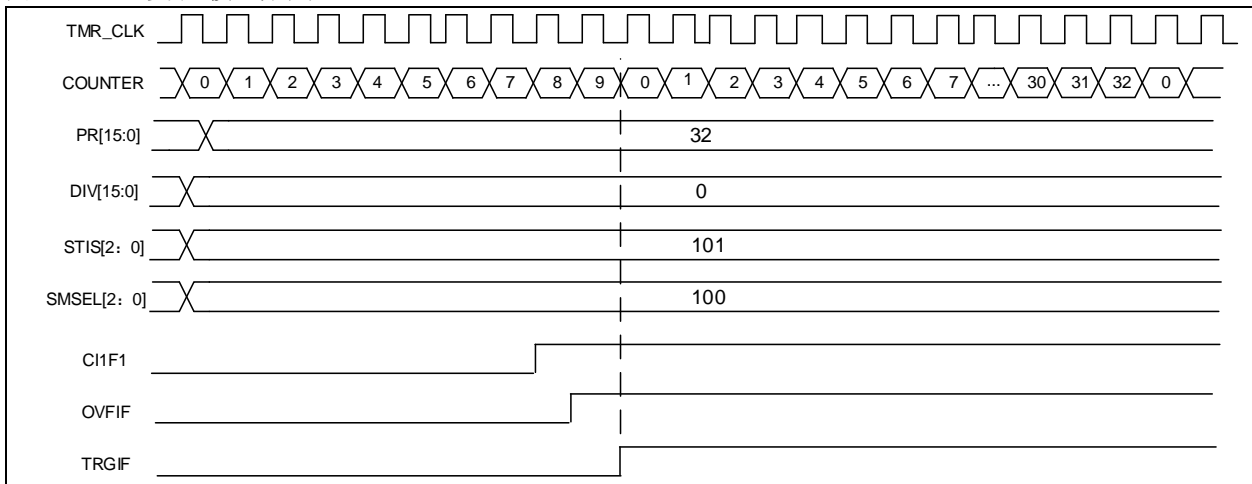
### 14.3.3.5 TMR同步

TMR9 和 12 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[2: 0]位选择从模式，即次定时器的工作模式。

从模式：复位模式

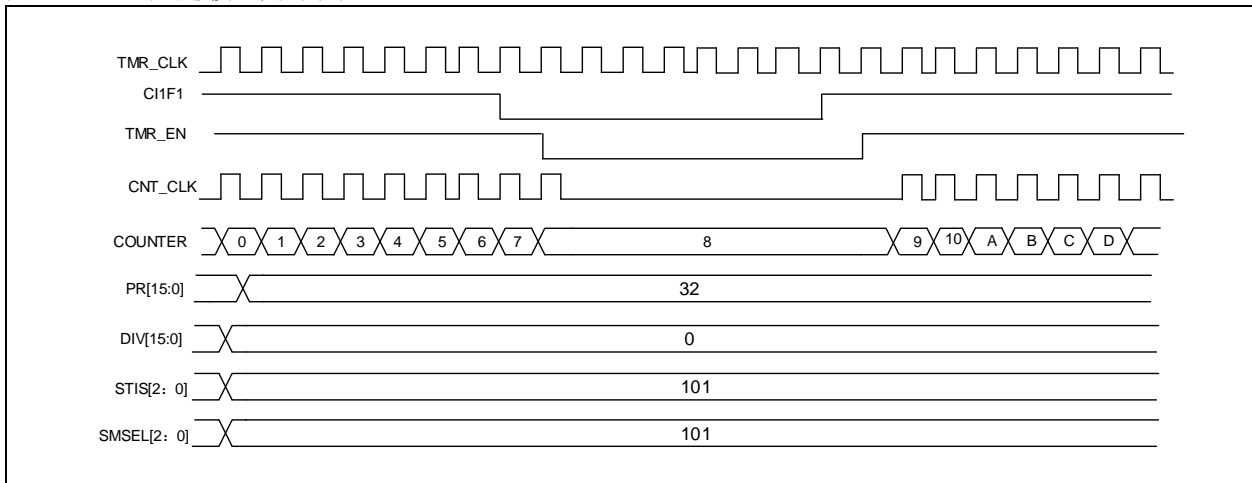
选中的触发信号将复位计数器和预分频器，若 OVFS 位为 0，将产生一个溢出事件。

图 14-46 复位模式例子

**从模式：挂起模式**

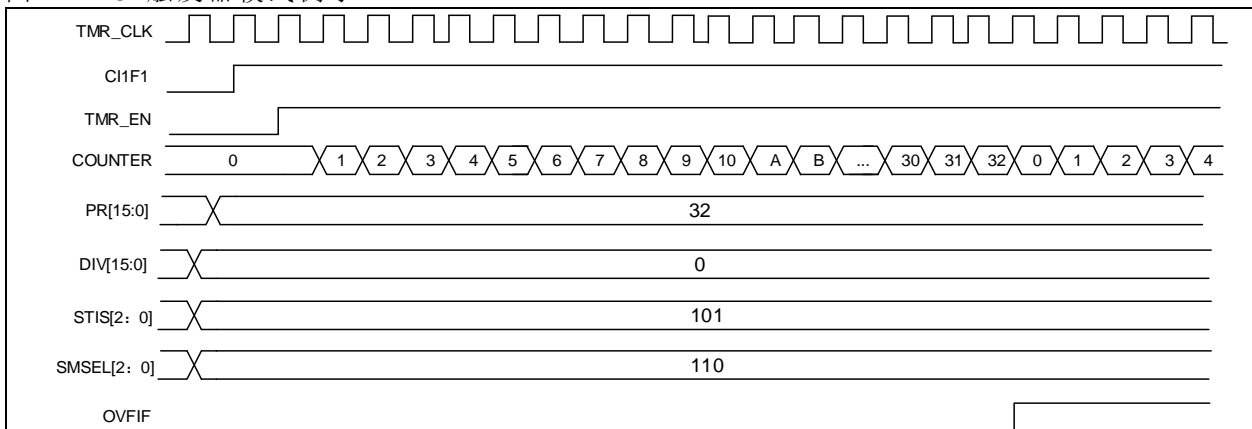
挂起模式下，计数的计数和停止受选中触发输入信号控制，当触发输入为高电平时计数器开始计数；当为低电平时，计数器暂停计数。

图 14-47 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 14-48 触发器模式例子



定时器的同步的更多实例详见 [14.2.3.5 节](#)。

**14.3.3.6 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和TMR12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 TMRx 的所有寄存器映射到一个 16 位可寻址（编址）空间。

表 14-8 TMRx – 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
TMRx_STCTRL	0x08	0x0000
TMRx_IDEN	0x0C	0x0000
TMRx_ISTS	0x10	0x0000
TMRx_SWEVT	0x14	0x0000
TMRx_CM1	0x18	0x0000
TMRx_CCTRL	0x20	0x0000
TMRx_CVAL	0x24	0x0000
TMRx_DIV	0x28	0x0000
TMRx_PR	0x2C	0x0000
TMRx_C1DT	0x34	0x0000 0000
TMRx_C2DT	0x38	0x0000 0000

### 14.3.4.1 控制寄存器 1（TMRx\_CTRL1）

域	简称	复位值	类型	功能
位 15: 10	保留	0x00	resd	保持默认值。
位 9: 8	CLKDIV	0x0	rw	时钟除频（Clock divider） 00: 无除频； 01: 2 除频； 10: 4 除频； 11: 保留。
位 7	PRBEN	0x0	rw	周期缓冲使能（Period buffer enable） 0: 缓冲关闭； 1: 缓冲开启。
位 6: 4	保留	0x0	resd	保持默认值。
位 3	OCMEN	0x0	rw	单周期使能（One cycle mode enable） 该功能用于选择溢出事件后，计数器是否停止。 0: 关闭； 1: 开启。
位 2	OVFS	0x0	rw	溢出事件源选择（Overflow event source） 配置溢出事件或 DMA 请求来源。 0: 来源于计数器溢出、设置 OVFSWTR 位或次定时器控制器产生的溢出事件； 1: 只能来源于计数器溢出。
位 1	OVFEN	0x0	rw	溢出事件使能（Overflow event enable） 0: 开启； 1: 关闭。
位 0	TMREN	0x0	rw	使能定时器（TMR enable） 0: 关闭； 1: 开启。

## 14.3.4.2 次定时器控制寄存器 (TMRx\_STCTRL)

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	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: 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	保持默认值。
位 2: 0	SMSEL	0x0	rw	次定时器模式选择 (Subordinate TMR mode selection) 000: 关闭从模式; 001: 编码模式 A; 010: 编码模式 B; 011: 编码模式 C; 100: 复位模式 - TRGIN 输入上升沿时, 重新初始化计数器; 101: 挂起模式 - TRGIN 输入高电平时, 计数器计数; 110: 触发模式 - TRGIN 输入上升沿时, 产生触发事件; 111: 外部时钟模式 A - TRGIN 输入上升沿时, 提供时钟; 注: 编码器模式 A/B/C 配置方法请查看计数模式章节。

## 14.3.4.3 DMA/中断使能寄存器 (TMRx\_IDEN)

域	简称	复位值	类型	功能
位 15: 7	保留	0x000	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: 3	保留	0x0	resd	保持默认值。
位 2	C2IEN	0x0	rw	通道 2 中断使能 (Channel 2 interrupt enable) 0: 关闭; 1: 开启。
位 1	C1IEN	0x0	rw	通道 1 中断使能 (Channel 1 interrupt enable) 0: 关闭; 1: 开启。
位 0	OVFIEN	0x0	rw	溢出中断使能 (overflow interrupt enable) 0: 关闭; 1: 开启。

## 14.3.4.4 中断状态寄存器 (TMRx\_ISTS)

域	简称	复位值	类型	功能
位 15: 11	保留	0x00	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: 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: 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) 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；

## 14.3.4.5 软件事件寄存器 (TMRx\_SWEVT)

域	简称	复位值	类型	功能
位 15: 7	保留	0x000	resd	保持默认值。
位 6	TRGSWTR	0x0	rw	软件触发触发事件 (Trigger event triggered by software) 通过软件触发一个触发事件。 0：无作用； 1：制造一个触发事件。
位 5: 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。

## 14.3.4.6 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 描述了通道在输出模式下的功能。因此必须注意，同一个位在输出模式和输入模式下的功能是不同的。

**输出比较模式：**

域	简称	复位值	类型	功能
位 15	保留	0x0	resd	保持默认值
位 14: 12	C2OCTRL	0x0	rw	通道 2 输出控制 (Channel 2 output control)
位 11	C2OBEN	0x0	rw	通道 2 输出缓存使能 (Channel 2 output buffer enable)
位 10	C2OIEN	0x0	rw	通道 2 输出立即使能 (Channel 2 output immediately enable)
位 9: 8	C2C	0x0	rw	通道 2 配置 (Channel 2 configure) 当 C2EN='0' 时，这些位用于选择通道 2 为输出或输入，以及输入时的映射选择： 00: 输出； 01: 输入，C2IN 映射在 C2IRAW 上； 10: 输入，C2IN 映射在 C1IRAW 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	保留	0x0	resd	保持默认值
位 6: 4	C1OCTRL	0x0	rw	通道 1 输出控制 (Channel 1 output control) 这些位用于设置原始信号 C1ORAW 的工作状态。 000: 断开。断开 C1ORAW 到 C1OUT 的输出； 001: 设置 C1ORAW 为高：TMRx_CVAL=TMRx_C1DT 时。 010: 设置 C1ORAW 为低：TMRx_CVAL=TMRx_C1DT 时。 011: 切换 C1ORAW 的电平：当 TMRx_CVAL=TMRx_C1DT 时。 100: 固定 C1ORAW 为低。 101: 固定 C1ORAW 为高。 110: PWM 模式 A —OWCDIR=0, 若 TMRx_C1DT>TMRx_CVAL 时设置 C1ORAW 为高，否则为低； —OWCDIR=1, 若 TMRx_C1DT <TMRx_CVAL 时设置 C1ORAW 为低，否则为高。 111: PWM 模式 B —OWCDIR=0, 若 TMRx_C1DT >TMRx_CVAL 时设置 C1ORAW 为低，否则为高； —OWCDIR=1, 若 TMRx_C1DT <TMRx_CVAL 时设置 C1ORAW 为高，否则为低。 注：除'000'外，其余配置下 C1OUT 将连接到 C1ORAW，C1OUT 的输出电平除了会根据 C1ORAW 变化外，还与 CCTRL 所配置的输出极性有关。
位 3	C1OBEN	0x0	rw	通道 1 输出缓存使能 (Channel 1 output buffer enable) 0: 关闭 TMRx_C1DT 的缓存功能，写入 TMRx_C1DT 的内容会立即生效。 1: 启用 TMRx_C1DT 的缓存功能，写入 TMRx_C1DT 的内容将保存到缓存寄存器中，当发生溢出事件时再更新到 TMRx_C1DT 中。
位 2	C1OIEN	0x0	rw	通道 1 输出立即使能 (Channel 1 output immediately enable) 在 PWM 模式 A 或模式 B 下，该位能够缩短触发事件到通道 1 的输出响应间的时间。 0: 需要比较 CVAL 与 C1DT 的值之后再产生输出。 1: 无需比较 CVAL 与 C1DT 的值，当发生触发事件时立即产生输出。
位 1: 0	C1C	0x0	rw	通道 1 配置 (Channel 1 configure)



<p>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：</p> <p>00: 输出；</p> <p>01: 输入，C1IN 映射在 C1IRAW 上；</p> <p>10: 输入，C1IN 映射在 C2IRAW 上；</p> <p>11: 输入，C1IN 映射在 STCI 上，只有在 STIS 选择内部触发输入时才工作。</p>				
<b>输入模式：</b>				
域	简称	复位值	类型	功能
位 15: 12	C2DF	0x0	rw	通道 2 滤波器 (Channel 2 digital filter)
位 11: 10	C2IDIV	0x0	rw	通道 2 分频系数 (Channel 2 input divider)
位 9: 8	C2C	0x0	rw	<p>通道 2 配置 (Channel 2 configure)</p> <p>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：</p> <p>00: 输出；</p> <p>01: 输入，C2IN 映射在 C2IRAW 上；</p> <p>10: 输入，C2IN 映射在 C1IRAW 上；</p> <p>11: 输入，C2IN 映射在 STCI 上，只有在 STIS 选择内部触发输入时才工作。</p>
位 7: 4	C1DF	0x0	rw	<p>通道 1 滤波器 (Channel 1 digital filter)</p> <p>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：</p> <p>0000: 无滤波器，以 <math>f_{DTS}</math> 采样</p> <p>1000: 采样频率 <math>f_{SAMPLING}=f_{DTS}/8</math>, N=6</p> <p>0001: 采样频率 <math>f_{SAMPLING}=f_{CK\_INT}</math>, N=2</p> <p>1001: 采样频率 <math>f_{SAMPLING}=f_{DTS}/8</math>, N=8</p> <p>0010: 采样频率 <math>f_{SAMPLING}=f_{CK\_INT}</math>, N=4</p> <p>1010: 采样频率 <math>f_{SAMPLING}=f_{DTS}/16</math>, N=5</p> <p>0011: 采样频率 <math>f_{SAMPLING}=f_{CK\_INT}</math>, N=8</p> <p>1011: 采样频率 <math>f_{SAMPLING}=f_{DTS}/16</math>, N=6</p> <p>0100: 采样频率 <math>f_{SAMPLING}=f_{DTS}/2</math>, N=6</p> <p>1100: 采样频率 <math>f_{SAMPLING}=f_{DTS}/16</math>, N=8</p> <p>0101: 采样频率 <math>f_{SAMPLING}=f_{DTS}/2</math>, N=8</p> <p>1101: 采样频率 <math>f_{SAMPLING}=f_{DTS}/32</math>, N=5</p> <p>0110: 采样频率 <math>f_{SAMPLING}=f_{DTS}/4</math>, N=6</p> <p>1110: 采样频率 <math>f_{SAMPLING}=f_{DTS}/32</math>, N=6</p> <p>0111: 采样频率 <math>f_{SAMPLING}=f_{DTS}/4</math>, N=8</p> <p>1111: 采样频率 <math>f_{SAMPLING}=f_{DTS}/32</math>, N=8</p>
位 3: 2	C1IDIV	0x0	rw	<p>通道 1 分频系数 (Channel 1 input divider)</p> <p>这些位定义了通道 1 的分频系数。</p> <p>00: 不分频，每一个有效的边沿都会产生一次输入；</p> <p>01: 每 2 个有效的边沿产生一次输入；</p> <p>10: 每 4 个有效的边沿产生一次输入；</p> <p>11: 每 8 个有效的边沿产生一次输入。</p> <p>注：C1EN='0'时，分频系数复位。</p>
位 1: 0	C1C	0x0	rw	<p>通道 1 配置 (Channel 1 configure)</p> <p>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：</p> <p>00: 输出；</p> <p>01: 输入，C1IN 映射在 C1IRAW 上；</p> <p>10: 输入，C1IN 映射在 C2IRAW 上；</p> <p>11: 输入，C1IN 映射在 STCI 上，只有在 STIS 选择内部触发输入时才工作。</p>

## 14.3.4.7 通道控制寄存器 (TMRx\_CTRL)

域	简称	复位值	类型	功能
位 15: 6	保留	0x00	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: 2	保留	0x0	resd	保持默认值。
位 1	C1P	0x0	rw	通道 1 极性 (Channel 1 polarity) 通道 1 配置为输出: 0: C1OUT 的有效电平为高 1: C1OUT 的有效电平为低 通道 1 配置为输入: 0: C1IN 的有效边沿为上升沿; 作为外部触发使用时, C1IN 不反相。 1: C1IN 的有效边沿为下降沿; 作为外部触发使用时, C1IN 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 标准CxOUT通道的输出控制位

CxEN 位	CxOUT 输出状态
0	禁止输出 (CxOUT=0)
1	CxOUT = CxORAW + 极性

注意: 连接到标准CxOUT通道的外部I/O管脚状态, 取决于CxOUT通道状态和GPIO以及IOMUX寄存器。

## 14.3.4.8 计数器 (TMRx\_CVAL)

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

## 14.3.4.9 预分频器 (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。

## 14.3.4.10 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。

## 14.3.4.11 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。

## 14.3.4.12 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 1 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。

## 14.3.5 TMR10、TMR11、TMR13和TMR14寄存器描述

必须以字 (32 位) 的方式操作这些外设寄存器。

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

表 14-10 TMRx – 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
TMRx_IDEN	0x0C	0x0000
TMRx_ISTS	0x10	0x0000
TMRx_SWEVT	0x14	0x0000
TMRx_CM1	0x18	0x0000
TMRx_CCTRL	0x20	0x0000
TMRx_CVAL	0x24	0x0000
TMRx_DIV	0x28	0x0000
TMRx_PR	0x2C	0x0000
TMRx_C1DT	0x34	0x0000

## 14.3.5.1 控制寄存器1 (TMRx\_CTRL1)

域	简称	复位值	类型	功能
位 15: 10	保留	0x00	resd	保持默认值。
位 9: 8	CLKDIV	0x0	rw	时钟除频 (Clock divider) 00: 无除频; 01: 2 除频; 10: 4 除频; 11: 保留。
位 7	PRBEN	0x0	rw	周期缓冲使能 (Period buffer enable) 0: 缓冲关闭; 1: 缓冲开启。
位 6: 4	保留	0x0	resd	保持默认值。
位 3	OCMEN	0x0	rw	单周期使能 (One cycle mode enable) 该功能用于选择更新事件后, 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 DMA/中断使能寄存器 (TMRx\_IDEN)

域	简称	复位值	类型	功能
位 15: 2	保留	0x0000	resd	保持默认值。
位 1	C1IEN	0x0	rw	通道 1 中断使能 (Channel 1 interrupt enable) 0: 关闭; 1: 开启。
位 0	OVFIEN	0x0	rw	溢出中断使能 (overflow interrupt enable) 0: 关闭; 1: 开启。

#### 14.3.5.3 中断状态寄存器 (TMRx\_ISTS)

域	简称	复位值	类型	功能
位 15: 10	保留	0x00	resd	保持默认值。
位 9	C1RF	0x0	rw0c	通道 1 再捕获标记 (Channel 1 recapture flag) C1IF 的状态已经为'1'时是否再次发生了捕获, 由硬件置'1', 写'0'清除。 0: 无捕获发生; 1: 捕获发生。
位 8: 2	保留	0x00	resd	保持默认值。
位 1	C1IF	0x0	rw0c	通道 1 中断标记 (Channel 1 interrupt flag) 若通道 1 为输入模式时: 捕获事件发生时由硬件置'1', 由软件清'0'或读 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 OVFE=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。

#### 14.3.5.4 软件事件寄存器 (TMRx\_SWEVT)

域	简称	复位值	类型	功能
位 15: 2	保留	0x0000	resd	保持默认值。
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。

### 14.3.5.5 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 描述了通道在输出模式下的功能。因此必须注意，同一个位在输出模式和输入模式下的功能是不同的。

输出比较模式：

域	简称	复位值	类型	功能
位 15: 7	保留	0x000	resd	保持默认值。
				通道 1 输出控制 (Channel 1 output control) 这些位用于设置原始信号 C1ORAW 的工作状态。 000: 断开。断开 C1ORAW 到 C1OUT 的输出; 001: 设置 C1ORAW 为高: TMRx_CVAL=TMRx_C1DT 时。 010: 设置 C1ORAW 为低: TMRx_CVAL=TMRx_C1DT 时。 011: 切换 C1ORAW 的电平: 当 TMRx_CVAL=TMRx_C1DT 时。 100: 固定 C1ORAW 为低。 101: 固定 C1ORAW 为高。 110: PWM 模式 A —OWCDIR=0, 若 TMRx_C1DT>TMRx_CVAL 时设置 C1ORAW 为高, 否则为低; —OWCDIR=1, 若 TMRx_C1DT <TMRx_CVAL 时设置 C1ORAW 为低, 否则为高。 111: PWM 模式 B —OWCDIR=0, 若 TMRx_C1DT >TMRx_CVAL 时设置 C1ORAW 为低, 否则为高; —OWCDIR=1, 若 TMRx_C1DT <TMRx_CVAL 时设置 C1ORAW 为高, 否则为低。 注: 除'000'外, 其余配置下 C1OUT 将连接到 C1ORAW, C1OUT 的输出电平除了会根据 C1ORAW 变化外, 还与 CCTRL 所配置的输出极性有关。
位 6: 4	C1OCTRL	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 3	C1OBEN	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 2	C1OIEN	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 C1IRAW 上; 10: 输入, C1IN 映射在 C2IRAW 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	

## 输入模式:

域	简称	复位值	类型	功能
位 15: 8	保留	0x00	resd	保持默认值
位 7: 4	C1DF	0x0	rw	通道 1 滤波器 (Channel 1 digital filter) 这些位用于配置通道 1 的滤波器。滤波的个数为 N, 则表示发生了 N 次采样事件后输入边沿才能通过滤波器: 0000: 无滤波器, 以 $f_{DTS}$ 采样 1000: 采样频率 $f_{SAMPLING}=f_{DTS}/8$ , N=6 0001: 采样频率 $f_{SAMPLING}=f_{CK\_INT}$ , N=2 1001: 采样频率 $f_{SAMPLING}=f_{DTS}/8$ , N=8 0010: 采样频率 $f_{SAMPLING}=f_{CK\_INT}$ , N=4 1010: 采样频率 $f_{SAMPLING}=f_{DTS}/16$ , N=5 0011: 采样频率 $f_{SAMPLING}=f_{CK\_INT}$ , N=8 1011: 采样频率 $f_{SAMPLING}=f_{DTS}/16$ , N=6 0100: 采样频率 $f_{SAMPLING}=f_{DTS}/2$ , N=6 1100: 采样频率 $f_{SAMPLING}=f_{DTS}/16$ , N=8 0101: 采样频率 $f_{SAMPLING}=f_{DTS}/2$ , N=8 1101: 采样频率 $f_{SAMPLING}=f_{DTS}/32$ , N=5 0110: 采样频率 $f_{SAMPLING}=f_{DTS}/4$ , N=6 1110: 采样频率 $f_{SAMPLING}=f_{DTS}/32$ , N=6 0111: 采样频率 $f_{SAMPLING}=f_{DTS}/4$ , N=8 1111: 采样频率 $f_{SAMPLING}=f_{DTS}/32$ , N=8
				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	
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 C1IRAW 上; 10: 输入, C1IN 映射在 C2IRAW 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。

## 14.3.5.6 通道控制寄存器 (TMRx\_CTRL)

域	简称	复位值	类型	功能
位 15: 2	保留	0x0	resd	保持默认值。
位 1	C1P	0x0	rw	通道 1 极性 (Channel 1 polarity) 通道 1 配置为输出: 0: C1OUT 的有效电平为高 1: C1OUT 的有效电平为低 通道 1 配置为输入: 0: C1IN 的有效边沿为上升沿; 作为外部触发使用时, C1IN 不反相。 1: C1IN 的有效边沿为下降沿; 作为外部触发使用时, C1IN 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 标准CxOUT通道的输出控制位

CxEN 位	CxOUT 输出状态
0	禁止输出 (CxOUT=0)
1	CxOUT = CxORAW + 极性

注意: 连接到标准 CxOUT 通道的外部 I/O 管脚状态, 取决于 CxOUT 通道状态和 GPIO 以及 IOMUX 寄存器。

## 14.3.5.7 计数值 (TMRx\_CVAL)

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

## 14.3.5.8 预分频器 (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 $f_{CK\_CNT} = f_{TMR\_CLK} / (DIV[15: 0] + 1)$ 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。

## 14.3.5.9 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。

## 14.3.5.10 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。



## 14.4 高级控制定时器 (TMR1、TMR8)

### 14.4.1 TMR1、TMR8简介

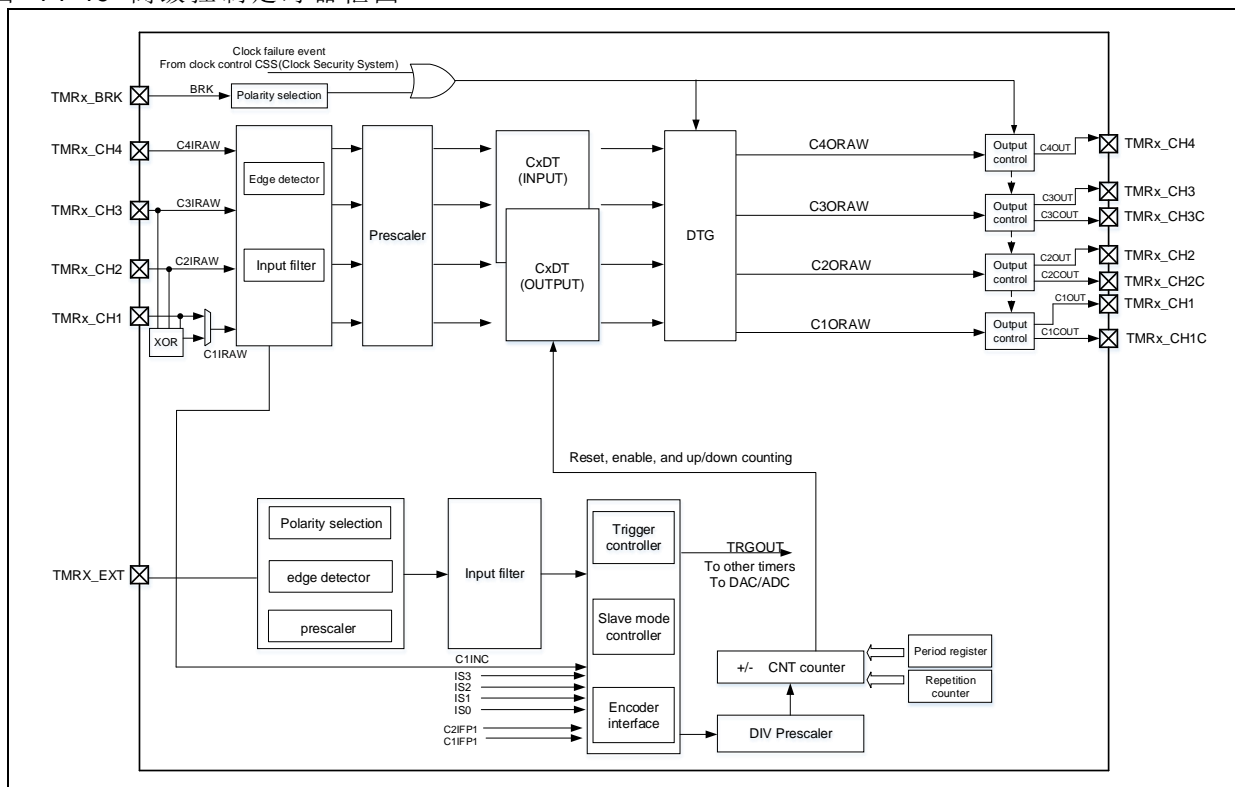
高级定时器 TMR1、TMR8 包含一个支持向上、向下、中央双向对齐计数的 16 位计数器、4 个通道寄存器、4 组独立的通道。可实现嵌入死区、输入捕获、可编程 PWM 输出。

#### 14.4.2 TMR1、TMR8主要特性

TMR1、TMR8 定时器的功能包括:

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

图 14-49 高级控制定时器框图



### 14.4.3 TMR1、TMR8功能描述

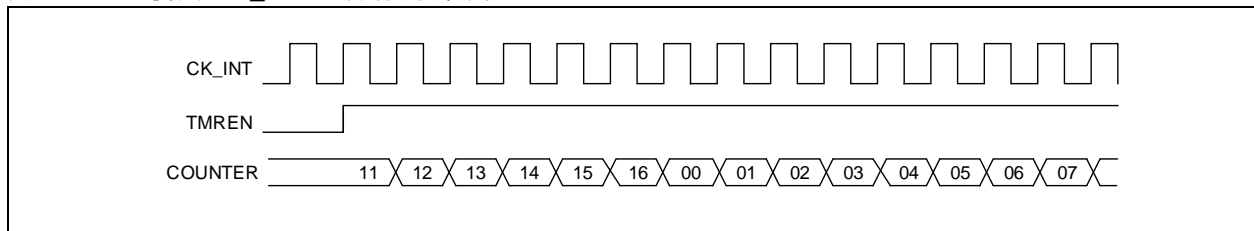
### 14.4.3.1 计数时钟

TMR1、TMR8 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。

### 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。

图 14-50 使用CK\_INT且分频系数为1



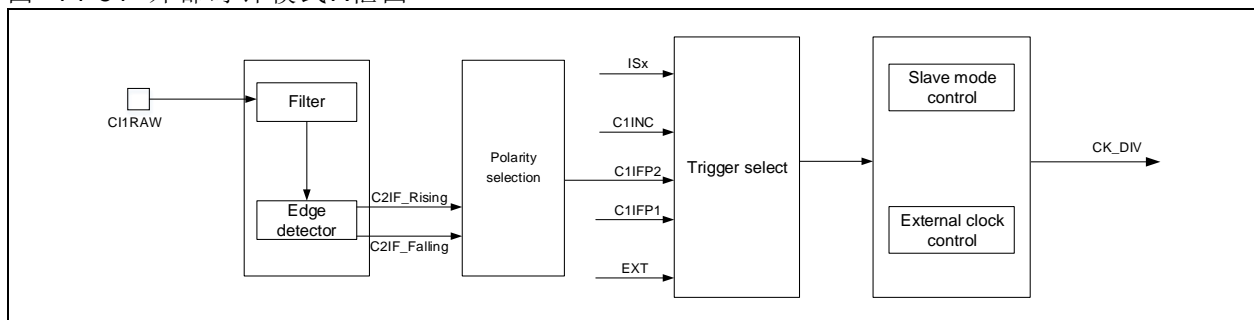
### 外部时钟（TRGIN/EXT）

计数时钟可由两种外部时钟源提供，分别为 TRGIN 和 EXT 信号。

当 SMSEL=3'111 时，外部时钟模式 A 被选中，配置 STIS[2: 0]来选择 TRGIN 信号驱动计数器计数。

当 ECMBEN=1 时，外部时钟模式 B 被选中，计数器由 EXT 信号驱动计数。

图 14-51 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 14-52 使用外部时钟模式A计数

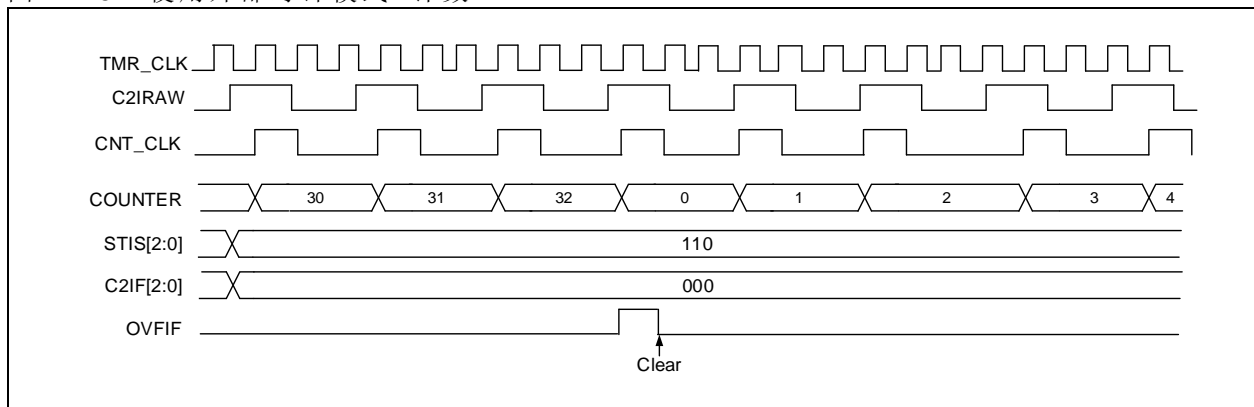
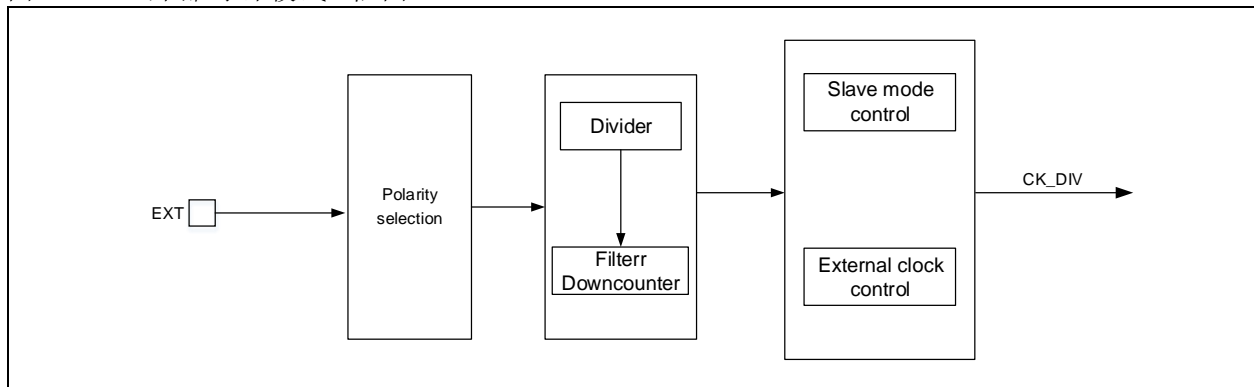
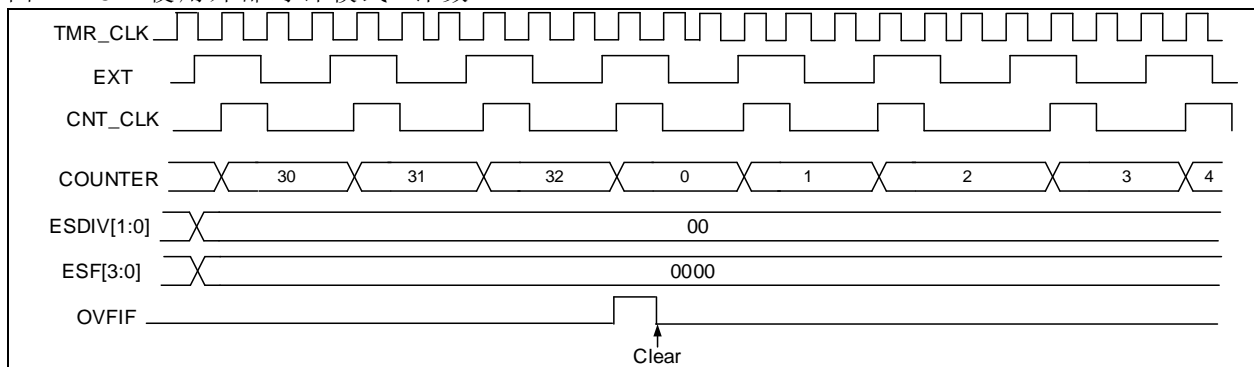


图 14-53 外部时钟模式B框图



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 14-54 使用外部时钟模式B计数



#### 内部触发输入 (ISx)

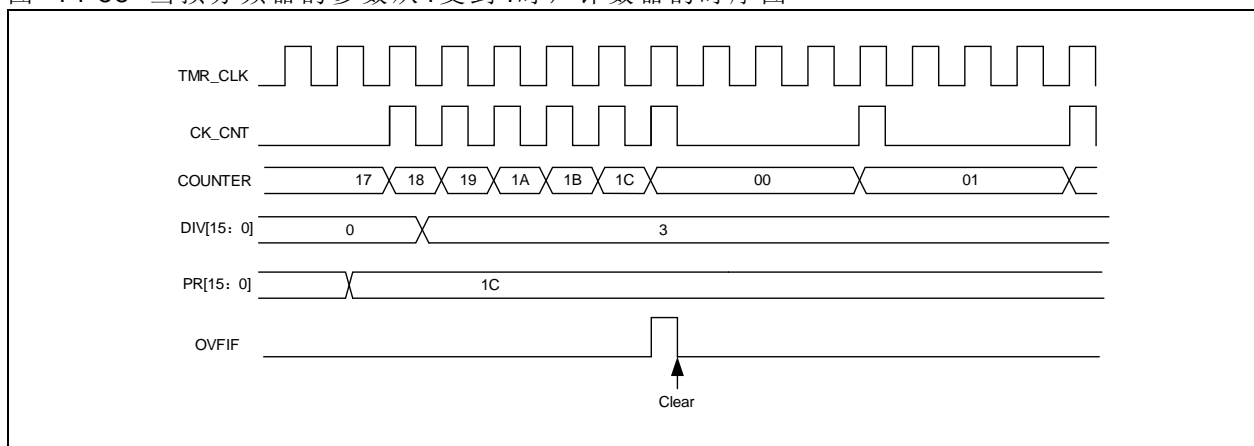
定时器之间支持互联同步，因此一个定时器的 TMR\_CLK 可由另一个定时器输出信号 TRGOUT 提供。配置 STIS[2:0]选择内部触发信号驱动计数器计数。

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 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。

表 14-12 TMRx内部触发连接

次定时器	IS0 (STIS=000)	IS1 (STIS=001)	IS2 (STIS=010)	IS3 (STIS=011)
TMR1	TMR5	TMR2	TMR3	TMR4
TMR8	TMR1	TMR2	TMR4	TMR5
保留	TMR8	TMR2	TMR4	TMR5

图 14-55 当预分频器的参数从1变到4时，计数器的时序图



### 14.4.3.2 计数模式

高级定时器提供了多种计数模式，用来满足不同的应用场景。其内部拥有一个支持 16 位向上计、向下、中央双向对齐计数的计数器，计数器的值可由周期寄存器 (TMRx\_PR) 载入。默认下，周期寄存器 (TMRx\_PR) 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)，周期寄存器 (TMRx\_PR) 值会在溢出事件发生时传入它的影子寄存器。溢出事件由 OVFEN、OVFS 位配置。

将 TMREN 位置 1 可使能定时器计数，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。

#### 向上计数模式

向上计数模式中，当计数值达到 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，则计数器溢出后不再重载预分频值和重载值，否则预分频值和重载值将在溢出事件后更新。

图 14-56 PRBEN=0时的溢出事件

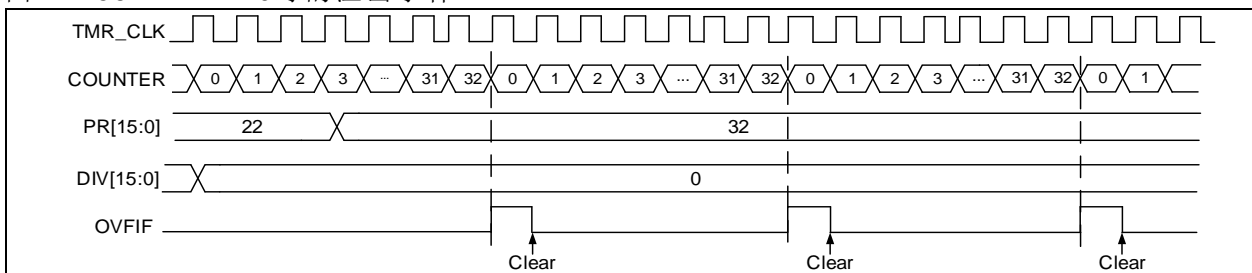
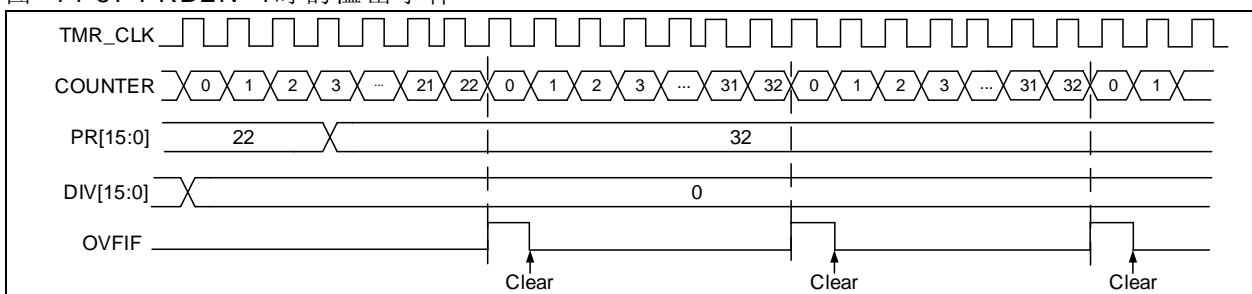


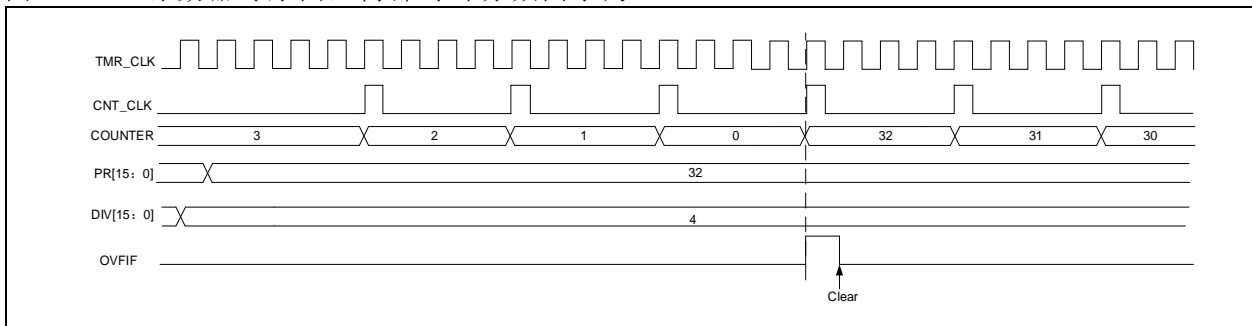
图 14-57 PRBEN=1时的溢出事件



### 向下计数模式

向下计数模式中，当计数值达到 0 值时，重新从 TMRx\_PR 向上下数，计数器下溢并产生溢出事件。

图 14-58 计数器时序图，内部时钟分频因子为 4

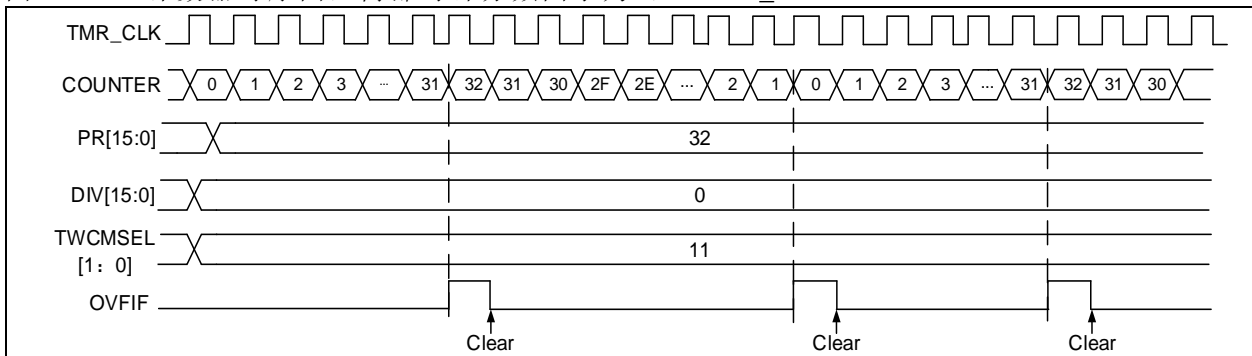


### 中央双向对齐计数模式

中央双向对齐计数模式中，计数器交替向上、向下计数。当计数值从 TMRx\_PR 值向下计数到 1 值时，产生下溢事件，然后从 0 开始向上计数；当向上计数到 TMRx\_PR 值-1 时，产生上溢事件，之后从 TMRx\_PR 值向下计数。计数器计数方向可由计数器方向控制位（OWCDIR）实时查看。

**注意：** 中央双向对齐计数模式下，OWCDIR 位为只读位。

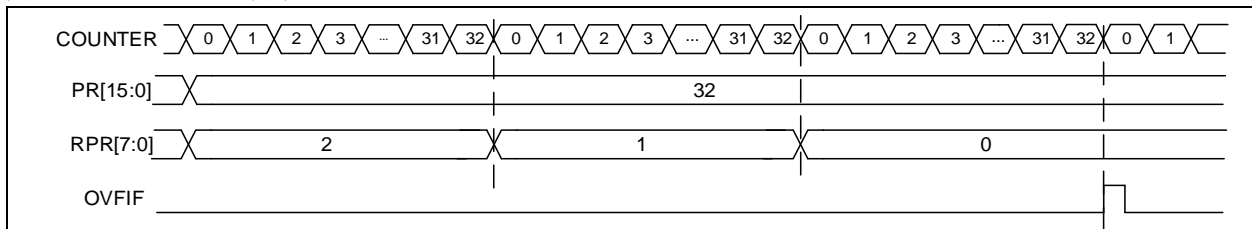
图 14-59 计数器时序图，内部时钟分频因子为 1，TMRx\_PR=0x32



### 重复计数模式：

重复计数器为非 0 值时，将开启重复计数功能。此时每次计数器溢出，重复计数器递减，当重复计数器达到 0 时，将产生溢出事件。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 14-60 RPR=2时的OVFIF



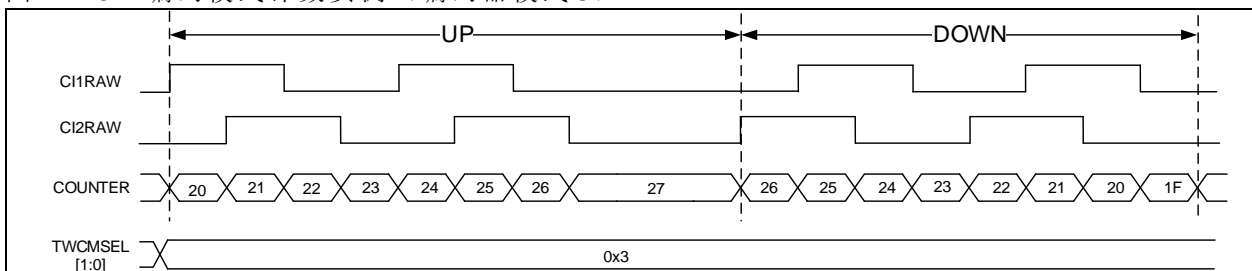
## 编码器模式

将 SMSEL[2:0]配置为 3'b001/3'b010/3'b011 可开启编码模式，编码模式下需提供两个输入(C1IN/C2IN)，根据一个输入的电平值，计数器将在另一个输入的边沿向上或向下计数。计数方向将由 OWCDIR 值指示。编码模式下计数器计数方向如下表所示：

表 14-13 计数方向与编码器信号的关系

有效边沿	相对信号的电平 (C1INFP1 对应 C2IN, C2INFP2 对应 C1IN)	C1INFP1 信号		C2INFP2 信号	
		上升	下降	上升	下降
仅在 C1IN 计数	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
仅在 C2IN 计数	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
在 C1IN 和 C2IN 上计数	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 14-61 编码模式计数实例 (编码器模式 C)



### 14.4.3.3 TMR输入部分

TMR1、TMR8 拥有 4 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，可用于对输入信号的滤波、选择、分频和输入捕获功能。

图 14-62 输入/输出通道1的主电路

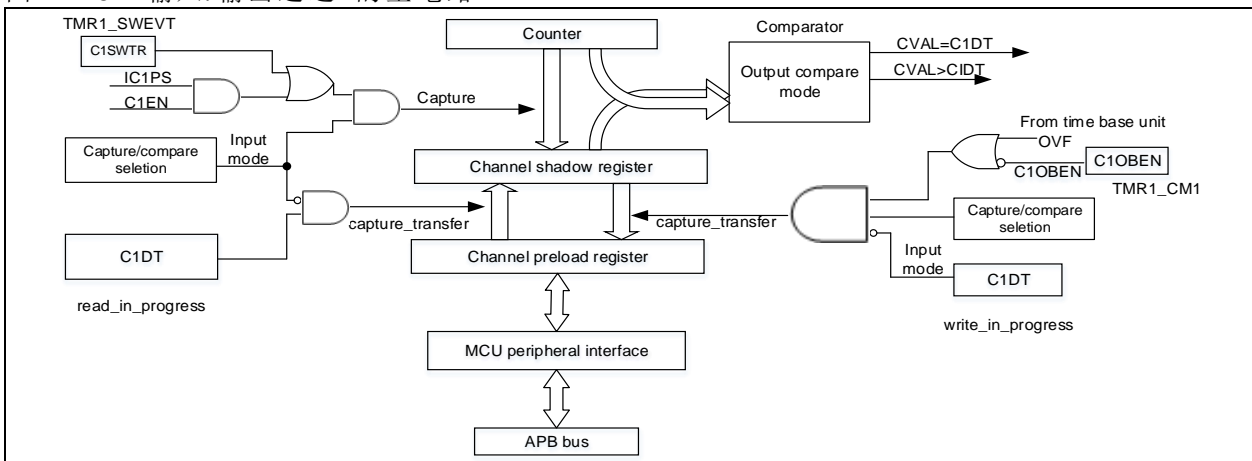
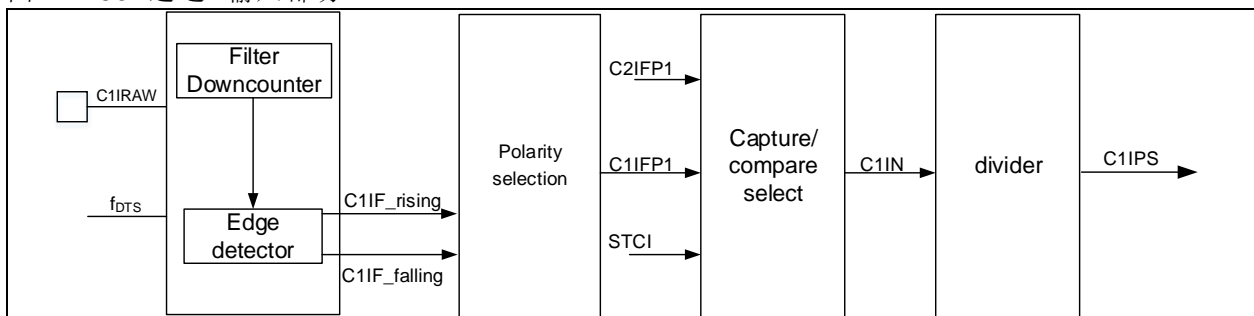


图 14-63 通道1输入部分



### 输入模式

此模式下，当选中的触发信号被检测到时，通道寄存器（TMRx\_CxDT）会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 已置 1 后检测到选中的触发信号，则将 CxOF 位置 1。

以若要捕获 C1IN 输入的上升沿，可按如下进行配置：

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

### 多输入异或

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

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

## 14.4.3.4 TMR输出部分

TMR 的输出部分由比较器和输出控制构成，用于编程输出信号的周期、占空比、极性。高级定时器的输出部分在不同通道上有所不同，如下图所示：

图 14-64 通道1至3输出部分

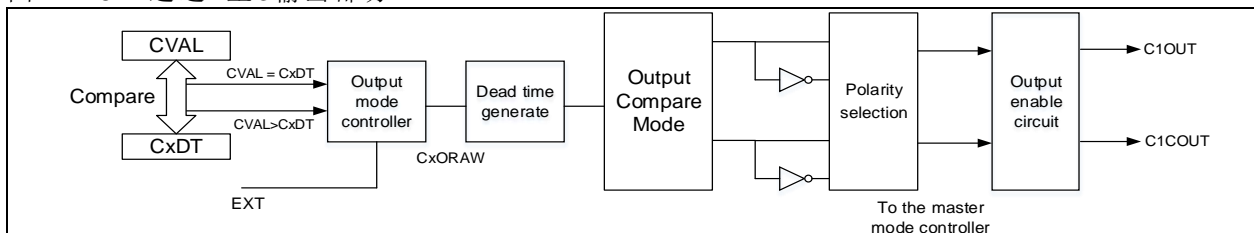
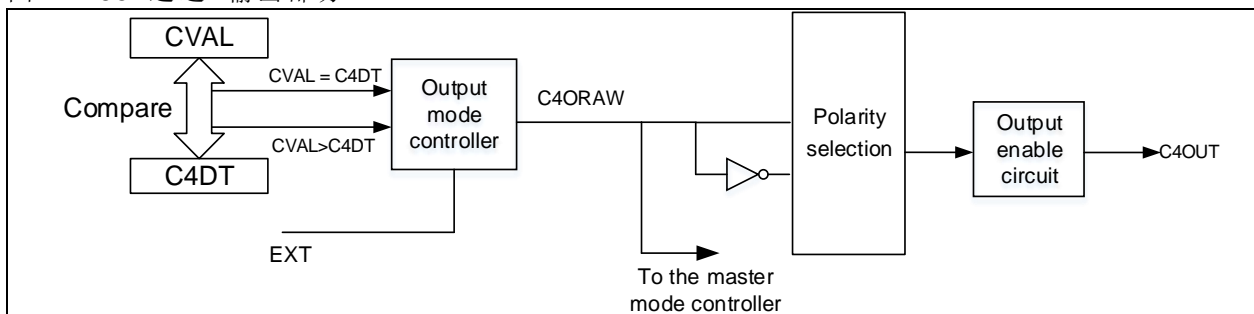


图 14-65 通道4输出部分



## 输出模式

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

输出比较模式有以下子类：

- **PWM 模式：** $CxOCTRL=2'b110/111$  时，开启 PWM 模式，每个通道可独立配置并输出一路 PWM 信号。此时，输出信号的周期由  $TMRx\_PR$  配置，占空比由  $CxDT$  值配置，计数器值与通道寄存器（ $TMRx\_CxDT$ ）值进行比较，根据计数方向输出指定电平信号，关于 PWM 模式 A/B 详见  $CxOCTRL[2:0]$  位描述。当计数模式为中央双向对齐计数时，可根据  $OWCDIR$  位指示计数方向。
- **强制输出模式：** $CxOCTRL=2'b100/101$  时，开启强制输出模式。此时， $CxORAW$  信号的电平被强制输出为配置的电平，而与计数值无关。虽然输出信号不依赖于比较结果，但通道标志位和 DMA 请求仍依赖于比较结果。
- **输出比较模式：** $CxOCTRL=2'b001/010/011$  时，开启输出比较模式。此时，当计数值与  $CxDT$  值匹配时， $CxORAW$  被强制为高电平、低电平或进行电平翻转。
- **单周期模式：**PWM 模式的特例，将  $OCMEN$  位置 1 可开启单周期模式，此模式下，仅在当前计数周期中进行比较匹配，完成当前计数后， $TMREN$  位清 0，因此仅输出一个脉冲。当配置为向上计数模式时，需要严格配置  $CVAL < CxDT \leq PR$ ；向下计数时，需严格配置  $CVAL > CxDT$ 。
- **快速输出模式：**将  $CxOIEN$  位置 1 可开启此功能，开启后  $CxORAW$  电平值不再在计数值与  $CxDT$  匹配时变化，而是在当前计数周期开始时，也就是说，比较结果被提前了，计数器值与通道寄存器（ $TMRx\_CxDT$ ）的比较结果将会提前决定  $CxORAW$  的电平。

图 14-66 展示了输出比较模式（翻转）的例子， $C1DT=0x3$ ，当计数值等于  $0x3$  时，输出电平  $C1OUT$  被翻转。

图 14-67 展示了计数器向上计数与 PWM 模式 A 配合的例子， $PR=0x32$ ， $CxDT$  配置为不同的值时输出时输出信号的翻转情况。

图 14-68 展示了计数器中央双向对齐计数与 PWM 模式 A 配合的例子， $PR=0x32$ ， $CxDT$  配置为不同的值时输出时输出信号的翻转情况。

图 14-69 展示了计数器向上计数与单周期模式下 PWM 模式 B 配合的例子，计数器仅计数了一个周期，输出信号在这个周期中只输出了一个脉冲。

图 14-66 计数值与  $C1DT$  值匹配时翻转  $C1ORAW$

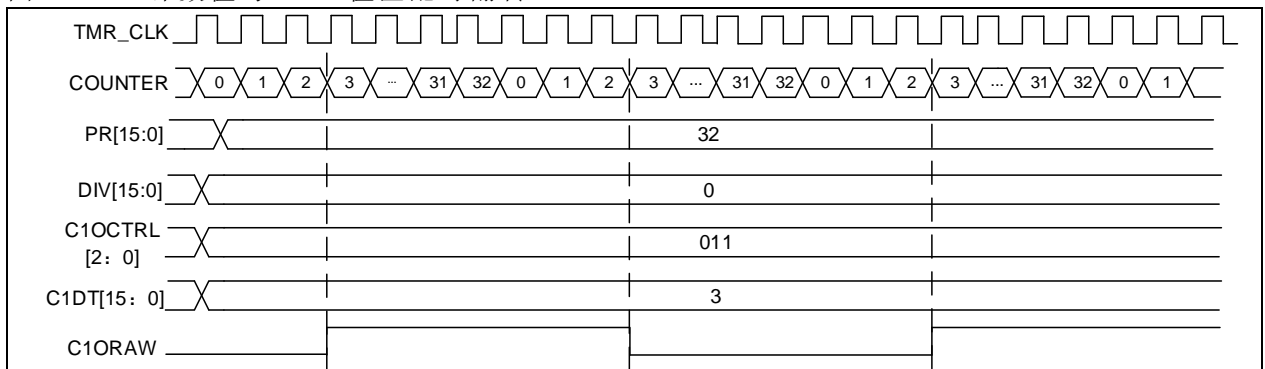




图 14-67 向上计数下PWM模式A

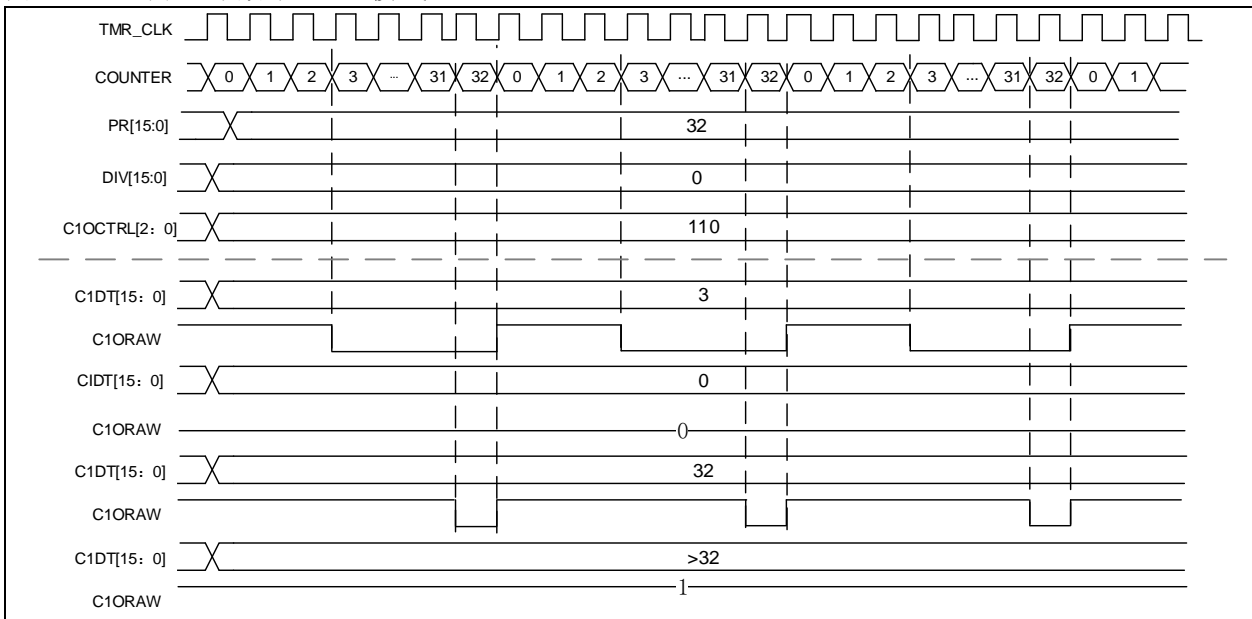


图 14-68 中央双向对齐计数下PWM模式

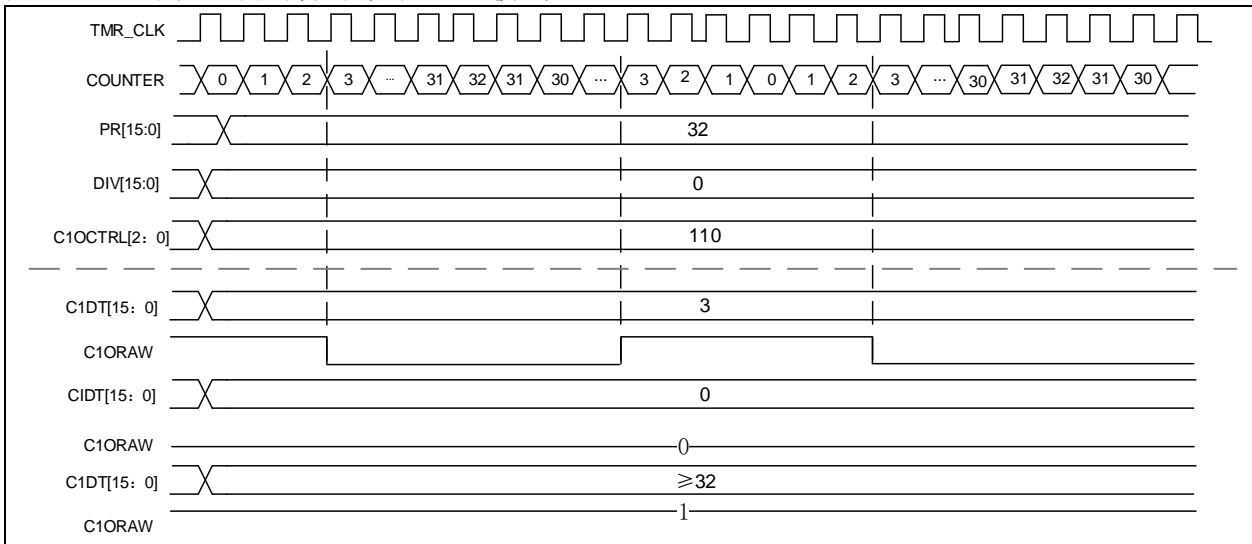
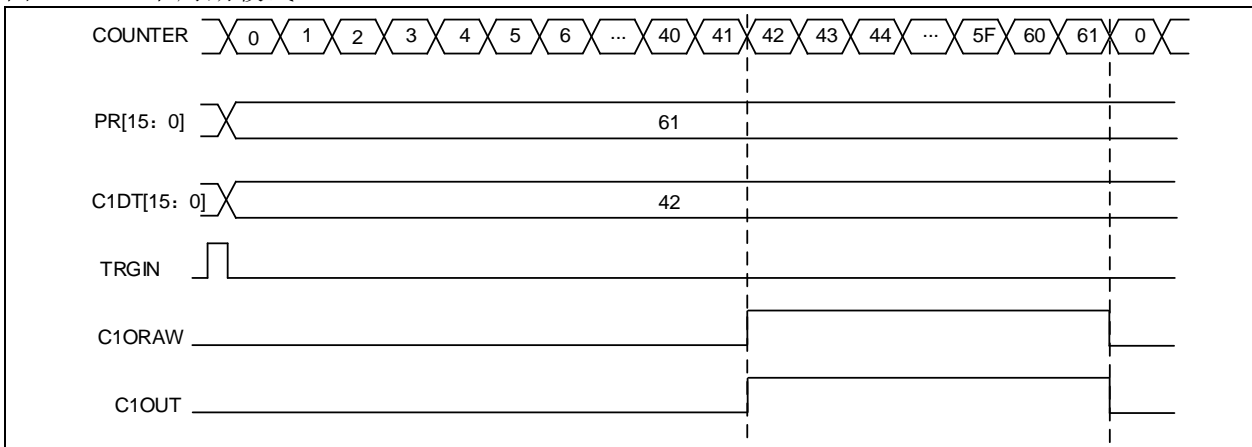


图 14-69 单周期模式

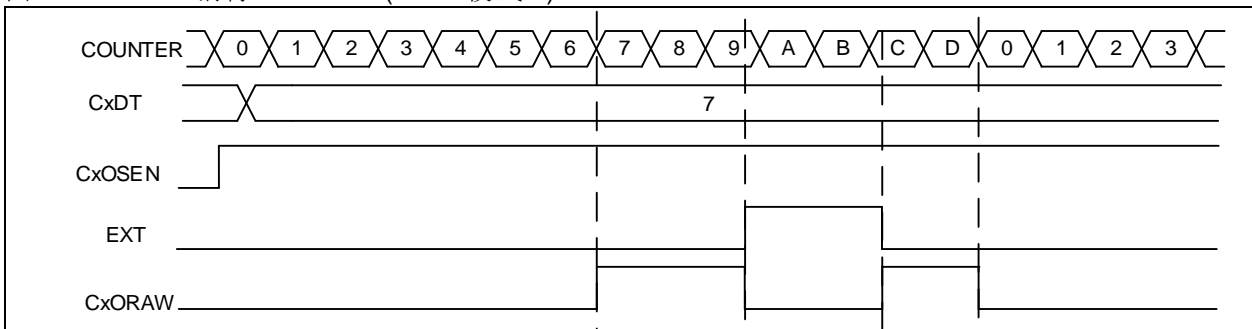


## CxORAW 信号清除

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

制模式时，CxORAW 信号清除功能不可用，只有在输出比较模式或 PWM 模式，此功能有效该功能只能用于输出比较和。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 14-70 EXT清除CxORAW(PWM模式A)



### 死区插入

高级定时器通道 1 至 3 包含一组反向通道输出，通过 CxCEN 使能，通过 CxCP 配置极性。CxOUT 和 CxCOUT 的输出状态见表 14-14。

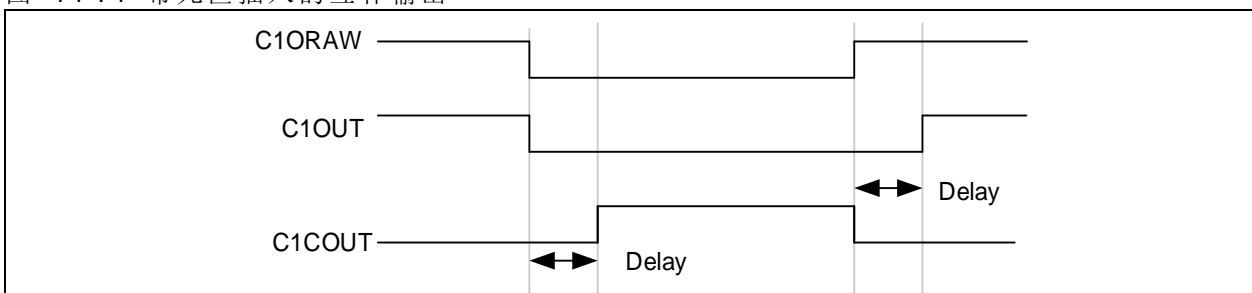
当转换为 IDLEF 状态，即 OEN 下降到 0，死区被激活。

将 CxEN 和 CxCEN 位置 1 后，通过配置 DTC[7: 0]死区发生器，可插入不同长时的死区。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如果 C1OUT 和 C1COUT 要产生相应的脉冲，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 14-71 带死区插入的互补输出



### 14.4.3.5 TMR刹车功能

开启刹车功能后（BRKEN 位置 1），CxOUT 和 CxCOUT 由 OEN、FCSODIS、FCOEN、CxIOS 和 CxCIOS 共同控制。但 CxOUT 和 CxCOUT 输出总是不能同时处于有效电平上的。详见表 14-15 带刹车功能的互补输出通道 CxOUT 和 CxCOUT 的控制位。

刹车信号来源可以是刹车输入管脚、时钟失效事件，刹车输入信号的极性由 BRKV 位控制。

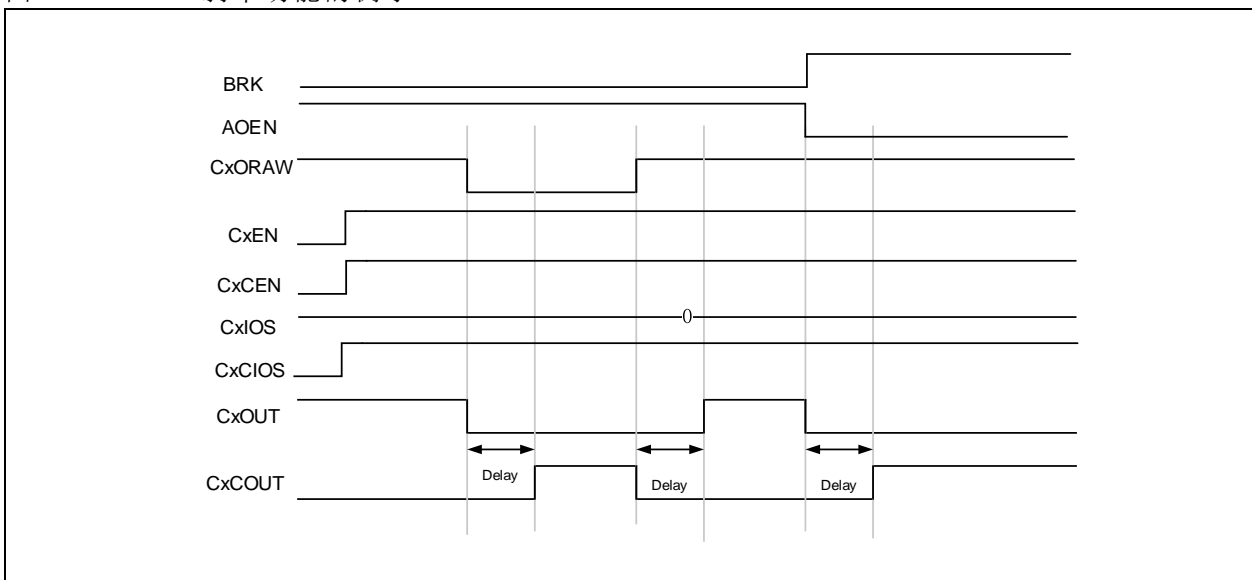
当发生刹车事件时，有下述动作：

- OEN 位异步清零，通道输出状态由 FCSODIS 位选择。关闭 MCU 的振荡器不影响该功能。
- OEN 被清零后，通道输出电平由 CxIOS 位设定。如果 FCSODIS=0，则定时器输出使能被禁止，否则输出使能始终为高。
- 当使用互补输出时：
  - 输出最开始处于复位状态，也就是无效的状态（取决于极性）。这是异步操作，定时器有无时钟并不影响此功能。
  - 定时器的时钟如果有效，会开启死区生成功能，CxIOS和CxCIOS位用来配置死区之后的电平。即使在这种情况下，CxOUT和CxCOUT也不能被同时驱动到有效的电平。  
注意，由于OEN位同步逻辑，死区时间较通常情况会延长一段时间（大约2个clk\_tmr的时钟周期）。

- 如果  $FCSODIS=0$ ，定时器释放使能输出，否则保持使能输出；或一旦  $CxEN$  与  $CxCEN$  之一变高时，使能输出变为高。
- 如果开启了刹车中断或 DMA 功能，刹车状态标志将置 1，并产生刹车中断或 DMA 请求。
- 如果将  $AOEN$  位置 1，在下一个溢出事件时  $OEN$  位被自动置 1。

注意：刹车输入电平有效时， $OEN$  不能被设置，状态标志  $BRKIF$  也不能被清除。

图 14-72 TMR刹车功能的例子



#### 14.4.3.6 TMR同步

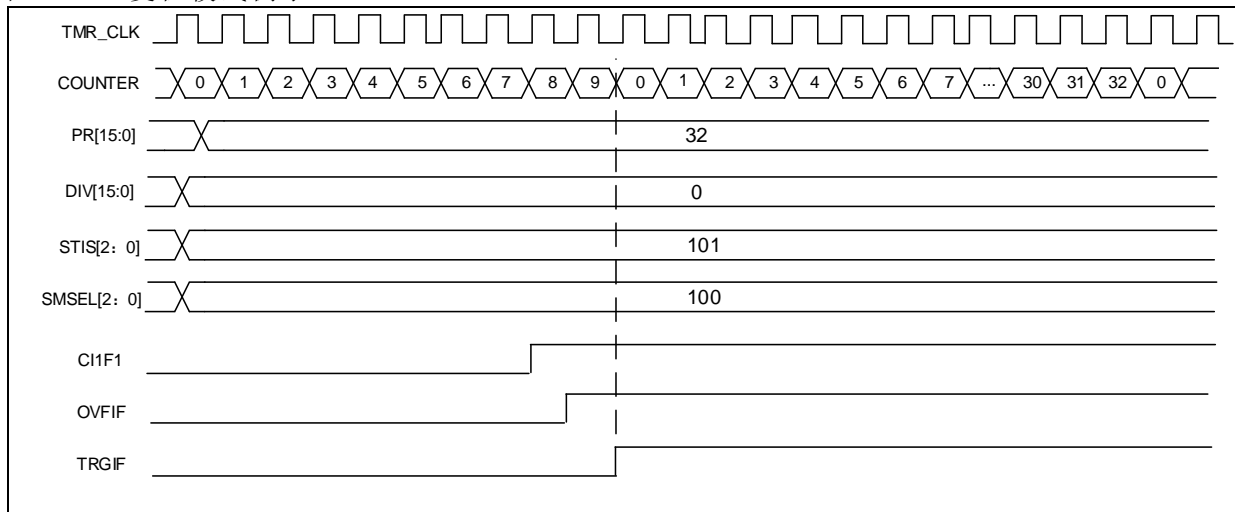
主次定时器之间可由内部连接信号进行同步。主定时器可由  $PTOS[2:0]$  位选择主定时器输出，即同步信息；次定时器由  $SMSEL[2:0]$  位选择从模式，即次定时器的工作模式。

定时器从模式有以下几种：

##### 从模式：复位模式

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

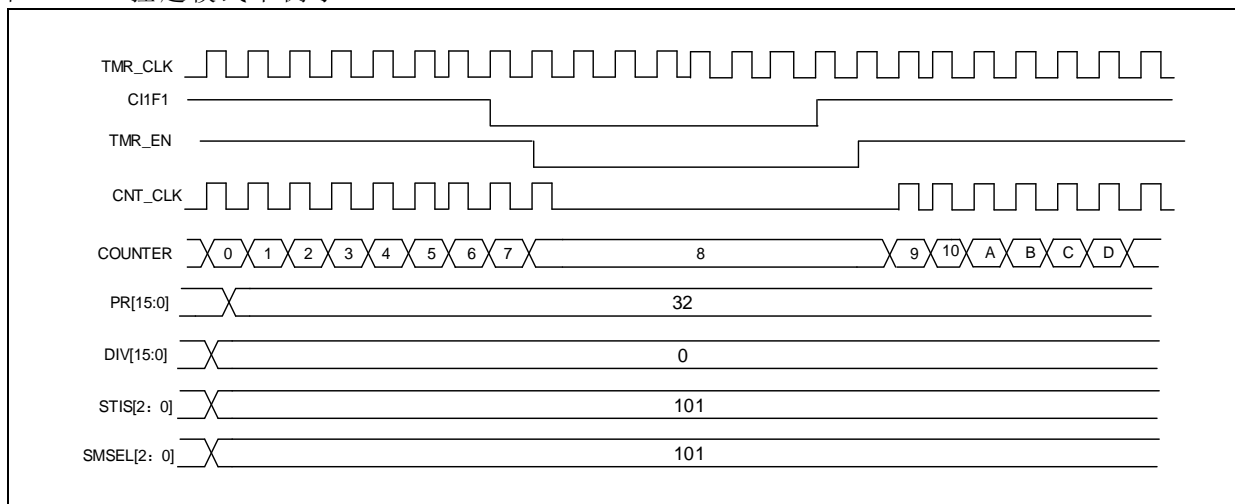
图 14-73 复位模式例子



##### 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。

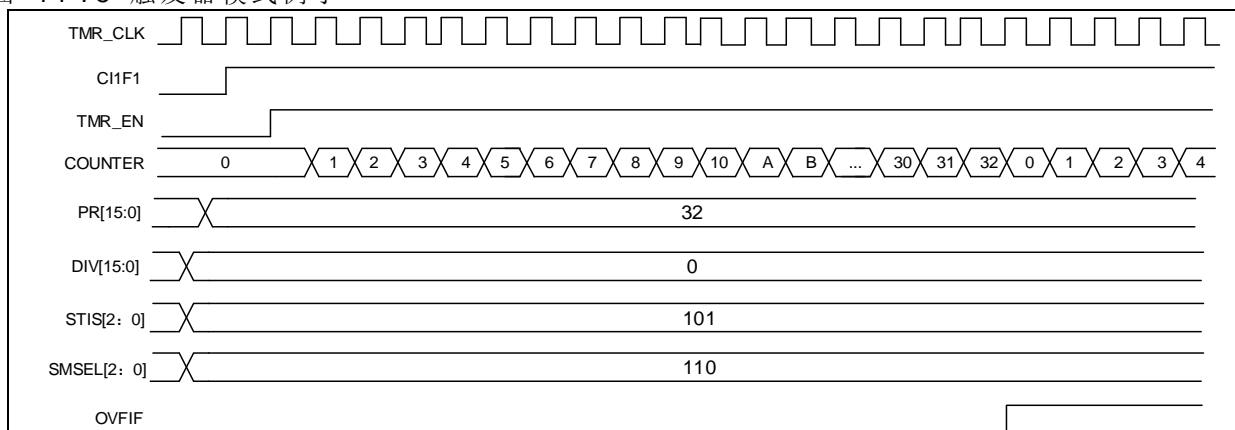
图 14-74 挂起模式下例子



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 14-75 触发器模式例子



定时器的同步的更多实例详见 [14.2.3.5](#) 节。

#### 14.4.3.7 调试模式

当微控制器进入调试模式（Cortex™-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。

#### 14.4.4 TMR1、TMR8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、TMR8 的所有寄存器映射到一个 16 位可寻址（编址）空间

表 14-14 TMR1、TMR8 – 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
TMRx_CTRL2	0x04	0x0000
TMRx_STCTRL	0x08	0x0000
TMRx_IDEN	0x0C	0x0000
TMRx_ISTS	0x10	0x0000
TMRx_SWEVT	0x14	0x0000
TMRx_CM1	0x18	0x0000
TMRx_CM2	0x1C	0x0000
TMRx_CCTRL	0x20	0x0000

TMRx_CVAL	0x24	0x0000
TMRx_DIV	0x28	0x0000
TMRx_PR	0x2C	0x0000
TMRx_RPR	0x30	0x0000
TMRx_C1DT	0x34	0x0000
TMRx_C2DT	0x38	0x0000
TMRx_C3DT	0x3C	0x0000
TMRx_C4DT	0x40	0x0000
TMRx_BRK	0x44	0x0000
TMRx_DMACTRL	0x48	0x0000
TMRx_DMA DT	0x4C	0x0000

## 14.4.4.1 TMR1、TMR8 控制寄存器1 (TMRx\_CTRL1)

域	简称	复位值	类型	功能
位 15: 10	保留	0x00	resd	保持默认值。
位 9: 8	CLKDIV	0x0	rw	时钟除频 (Clock divider) 00: 无除频; 01: 2 除频; 10: 4 除频; 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, 输出标志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, 输出标志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, 输出标志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。

## 14.4.4.2 TMR1、TMR8控制寄存器2 (TMRx\_CTRL2)

域	简称	复位值	类型	功能
位 15	保留	0x0	resd	保持默认值。
位 14	C4IOS	0x0	rw	通道 4 空闲输出状态 (Channel 4 idle output state)
位 13	C3CIOS	0x0	rw	通道 3 互补空闲输出状态 (Channel 3 complementary idle output state)
位 12	C3IOS	0x0	rw	通道 3 空闲输出状态 (Channel 3 idle output state)
位 11	C2CIOS	0x0	rw	通道 2 互补空闲输出状态 (Channel 2 complementary idle output state)
位 10	C2IOS	0x0	rw	通道 2 空闲输出状态 (Channel 2 idle output state)
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: C1OUTL=0; 1: C1OUTL=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 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: 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、TMR8次定时器控制寄存器 (TMRx\_STCTRL)

域	简称	复位值	类型	功能
位 15	ESP	0x0	rw	外部信号极性 (External signal polarity) 用于选择外部方式。 0: 高电平或上升沿; 1: 低电平或下降沿。
位 14	ECMBEN	0x0	rw	外部时钟模式 B 使能 (External clock mode B enable) 用于启用外部时钟模式 B 0: 关闭; 1: 开启。
位 13: 12	ESDIV	0x0	rw	外部信号除频 (External signal divide) 用于选择降低外部触发频率的除频。 00: 关闭分频;

				01: 2 分频; 10: 4 分频; 11: 8 分频。
				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 11: 8	ESF	0x0	rw	
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 <a href="#">表 14-12</a> 。
位 3	保留	0x0	resd	保留, 保持默认值。
位 2: 0	SMSEL	0x0	rw	次定时器模式选择 (Subordinate TMR mode selection) 000: 关闭从模式; 001: 编码模式 A; 010: 编码模式 B; 011: 编码模式 C; 100: 复位模式 - TRGIN 输入上升沿时, 重新初始化 CVAL; 101: 挂起模式 - TRGIN 输入高电平时, CVAL 计数; 110: 触发模式 - TRGIN 输入上升沿时, 产生触发事件; 111: 外部时钟模式 A - TRGIN 输入上升沿时, 提供时钟; 注: 编码器模式 A/B/C 配置方法请查看计数模式章节。



## 14.4.4.4 TMR1、TMR8 DMA/中断使能寄存器 (TMRx\_IDEN)

域	简称	复位值	类型	功能
位 15	保留	0x0	resd	保持默认值。
位 14	TDEN	0x0	rw	触发 DMA 请求使能 (Trigger DMA request enable) 0: 关闭; 1: 开启。
位 13	HALLDE	0x0	rw	HALL DMA 请求使能 (HALL DMA request enable) 0: 关闭; 1: 开启。
位 12	C4DEN	0x0	rw	通道 4 的 DMA 请求使能 (Channel 4 DMA request enable) 0: 关闭; 1: 开启。
位 11	C3DEN	0x0	rw	通道 3 的 DMA 请求使能 (Channel 3 DMA request enable) 0: 关闭; 1: 开启。
位 10	C2DEN	0x0	rw	通道 2 的 DMA 请求使能 (Channel 2 DMA request enable) 0: 关闭; 1: 开启。
位 9	C1DEN	0x0	rw	通道 1 的 DMA 请求使能 (Channel 1 DMA request enable) 0: 关闭; 1: 开启。
位 8	OVFDEN	0x0	rw	溢出事件的 DMA 请求使能 (overflow event DMA request enable) 0: 关闭; 1: 开启。
位 7	BRKIE	0x0	rw	刹车中断使能 (Brake interrupt enable) 0: 关闭; 1: 开启。
位 6	TIEN	0x0	rw	触发中断使能 (Trigger interrupt enable) 0: 关闭; 1: 开启。
位 5	HALLIEN	0x0	rw	HALL 中断使能 (HALL interrupt enable) 0: 关闭; 1: 开启。
位 4	C4IEN	0x0	rw	通道 4 中断使能 (Channel 4 interrupt enable) 0: 关闭; 1: 开启。
位 3	C3IEN	0x0	rw	通道 3 中断使能 (Channel 3 interrupt enable) 0: 关闭; 1: 开启。
位 2	C2IEN	0x0	rw	通道 2 中断使能 (Channel 2 interrupt enable) 0: 关闭; 1: 开启。
位 1	C1IEN	0x0	rw	通道 1 中断使能 (Channel 1 interrupt enable) 0: 关闭; 1: 开启。
位 0	OVFIEN	0x0	rw	更新中断使能 (Update interrupt enable) 0: 关闭; 1: 开启。

## 14.4.4.5 TMR1、TMR8中断状态寄存器（TMRx\_ISTS）

域	简称	复位值	类型	功能
位 15: 13	保留	0x0	resd	保持默认值。
位 12	C4RF	0x0	rw0c	通道 4 再捕获标记（Channel 4 recapture flag） 见 C1RF 的描述。
位 11	C3RF	0x0	rw0c	通道 3 再捕获标记（Channel 3 recapture flag） 见 C1RF 的描述。
位 10	C2RF	0x0	rw0c	通道 2 再捕获标记（Channel 2 recapture flag） 见 C1RF 的描述。
位 9	C1RF	0x0	rw0c	通道 1 再捕获标记（Channel 1 recapture flag） C1IF 的状态已经为'1'时是否再次发生了捕获，由硬件置'1'，写'0'清除。 0：无捕获发生； 1：捕获发生。
位 8	保留	0x0	resd	保持默认值。
位 7	BRKIF	0x0	rw0c	刹车中断标记（Brake interrupt flag） 用于标记刹车输入的电平是否有效，由硬件置'1'，写'0'清除。 0：无效； 1：有效。
位 6	TRGIF	0x0	rw0c	触发中断标记（Trigger interrupt flag） 当发生触发事件时由硬件置'1'，写'0'清除。 0：无触发事件发生； 1：发生触发事件。 触发事件：在 TRGIN 接收到有效边沿，或挂起模式下接收到任意边沿。
位 5	HALLIF	0x0	rw0c	HALL 中断标记（HALL interrupt flag） 当发生触发事件时由硬件置'1'，写'0'清除。 0：无 HALL 事件发生； 1：发生 HALL 事件。 HALL 事件：CxEN、CxSEN、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 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。

## 14.4.4.6 TMR1、TMR8软件事件寄存器 (TMRx\_SWEVT)

域	简称	复位值	类型	功能
位 15: 8	保留	0x00	resd	保持默认值。
位 7	BRKSWTR	0x0	wo	软件触发刹车事件 (Brake event triggered by software) 通过软件触发一个刹车事件。 0: 无作用; 1: 制造一个刹车事件。
位 6	TRGSWTR	0x0	wo	软件触发触发事件 (Trigger event triggered by software) 通过软件触发一个触发事件。 0: 无作用; 1: 制造一个触发事件。
位 5	HALLSWTR	0x0	wo	软件触发 HALL 事件 (HALL event triggered by software) 通过软件产生一个 HALL 事件。 0: 无作用; 1: 产生一个 HALL 事件。 注: 该位只对拥有互补输出的通道有效。
位 4	C4SWTR	0x0	wo	软件触发通道 4 事件 (Channel 4 event triggered by software) 见 C1M 的描述。
位 3	C3SWTR	0x0	wo	软件触发通道 3 事件 (Channel 3 event triggered by software) 见 C1M 的描述。
位 2	C2SWTR	0x0	wo	软件触发通道 2 事件 (Channel 2 event triggered by software) 见 C1M 的描述。
位 1	C1SWTR	0x0	wo	C1SWTR: 软件触发通道 1 事件 (Channel 1 event triggered by software) 通过软件触发一个通道 1 事件。 0: 无作用; 1: 制造一个通道 1 事件。
位 0	OVFSWTR	0x0	wo	软件触发溢出事件 (Overflow event triggered by software) 通过软件触发一个溢出事件。 0: 无作用; 1: 制造一个溢出事件。

## 14.4.4.7 TMR1、TMR8通道模式寄存器1 (TMRx\_CM1)

通道可用于输入 (捕获模式) 或输出 (比较模式), 通道的方向由相应的 CxC 位定义。该寄存器其它位的作用在输入和输出模式下不同。CxOx 描述了通道在输出模式下的功能, CxIx 描述了通道在输入模式下的功能。因此必须注意, 同一个位在输出模式和输入模式下的功能是不同的。

## 输出比较模式

域	简称	复位值	类型	功能
位 15	C2OSEN	0x0	rw	通道 2 输出开关使能 (Channel 2 output switch enable)
位 14: 12	C2OCTRL	0x0	rw	通道 2 输出控制 (Channel 2 output control)
位 11	C2OBEN	0x0	rw	通道 2 输出缓存使能 (Channel 2 output buffer enable)
位 10	C2OIEN	0x0	rw	通道 2 输出立即使能 (Channel 2 output immediately enable)
位 9: 8	C2C	0x0	rw	通道 2 配置 (Channel 2 configure) 当 C2EN='0' 时, 这些位用于选择通道 2 为输出或输入, 以及输入时的映射选择: 00: 输出; 01: 输入, C2IN 映射在 C2IRAW 上; 10: 输入, C2IN 映射在 C1IRAW 上; 11: 输入, C2IN 映射在 STI 上, 只有在 STIS 选择内部触发输入时才工作。
位 7	C1OSEN	0x0	rw	通道 1 输出开关使能 (Channel 1 output switch enable) 0: EXT 输入不影响 C1ORAW; 1: 当 EXT 输入高电平时, 将 C1ORAW 清 0。
位 6: 4	C1OCTRL	0x0	rw	通道 1 输出控制 (Channel 1 output control) 这些位用于设置原始信号 C1ORAW 的工作状态。

				<p>000: 断开。断开 C1ORAW 到 C1OUT 的输出;</p> <p>001: 设置 C1ORAW 为高: TMRx_CVAL=TMRx_C1DT 时。</p> <p>010: 设置 C1ORAW 为低: TMRx_CVAL=TMRx_C1DT 时。</p> <p>011 : 切换 C1ORAW 的电平: 当 TMRx_CVAL=TMRx_C1DT 时。</p> <p>100: 固定 C1ORAW 为低。</p> <p>101: 固定 C1ORAW 为高。</p> <p>110: PWM 模式 A</p> <p>— OWCDIR=0, 若 TMRx_C1DT&gt;TMRx_CVAL 时设置 C1ORAW 为高, 否则为低;</p> <p>— OWCDIR=1, 若 TMRx_C1DT &lt;TMRx_CVAL 时设置 C1ORAW 为低, 否则为高。</p> <p>111: PWM 模式 B</p> <p>— OWCDIR=0, 若 TMRx_C1DT &gt;TMRx_CVAL 时设置 C1ORAW 为低, 否则为高;</p> <p>— OWCDIR=1, 若 TMRx_C1DT &lt;TMRx_CVAL 时设置 C1ORAW 为高, 否则为低。</p> <p>注: 除'000'外, 其余配置下 C1OUT 将连接到 C1ORAW, C1OUT 的输出电平除了会根据 C1ORAW 变化外, 还与 CCTRL 所配置的输出极性有关。</p>
位 3	C1OBEN	0x0	rw	<p>通道 1 输出缓存使能 (Channel 1 output buffer enable)</p> <p>0: 关闭 TMRx_C1DT 的缓存功能, 写入 TMRx_C1DT 的内容会立即生效。</p> <p>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。</p>
位 2	C1OIEN	0x0	rw	<p>通道 1 输出立即使能 (Channel 1 output immediately enable)</p> <p>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。</p> <p>0: 需要比较 CVAL 与 C1DT 的值之后再产生输出。</p> <p>1: 无需比较 CVAL 与 C1DT 的值, 当发生触发事件时立即产生输出。</p>
位 1: 0	C1C	0x0	rw	<p>通道 1 配置 (Channel 1 configure)</p> <p>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:</p> <p>00: 输出;</p> <p>01: 输入, C1IN 映射在 C1IRAW 上;</p> <p>10: 输入, C1IN 映射在 C2IRAW 上;</p> <p>11: 输入, C1IN 映射在 STCI 上, 只有在 STIS 选择内部触发输入时才工作。</p>

## 输入模式

域	简称	复位值	类型	功能
位 15: 12	C2DF	0x0	rw	通道 2 滤波器 (Channel 2 digital filter)
位 11: 10	C2IDIV	0x0	rw	通道 2 分频系数 (Channel 2 input divider)
位 9: 8	C2C	0x0	rw	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 C2IRAW 上; 10: 输入, C2IN 映射在 C1IRAW 上; 11: 输入, C2IN 映射在 STCI 上, 只有在 STIS 选择内部触发输入时才工作。
位 7: 4	C1DF	0x0	rw	通道 1 滤波器 (Channel 1 digital filter) 这些位用于配置通道 1 的滤波器。滤波的个数为 N, 则表示发生了 N 次采样事件后输入边沿才能通过滤波器: 0000: 无滤波器, 以 $f_{DTS}$ 采样 1000: 采样频率 $f_{SAMPLING}=f_{DTS}/8$ , N=6 0001: 采样频率 $f_{SAMPLING}=f_{CK\_INT}$ , N=2 1001: 采样频率 $f_{SAMPLING}=f_{DTS}/8$ , N=8 0010: 采样频率 $f_{SAMPLING}=f_{CK\_INT}$ , N=4 1010: 采样频率 $f_{SAMPLING}=f_{DTS}/16$ , N=5 0011: 采样频率 $f_{SAMPLING}=f_{CK\_INT}$ , N=8 1011: 采样频率 $f_{SAMPLING}=f_{DTS}/16$ , N=6 0100: 采样频率 $f_{SAMPLING}=f_{DTS}/2$ , N=6 1100: 采样频率 $f_{SAMPLING}=f_{DTS}/16$ , N=8 0101: 采样频率 $f_{SAMPLING}=f_{DTS}/2$ , N=8 1101: 采样频率 $f_{SAMPLING}=f_{DTS}/32$ , N=5 0110: 采样频率 $f_{SAMPLING}=f_{DTS}/4$ , N=6 1110: 采样频率 $f_{SAMPLING}=f_{DTS}/32$ , N=6 0111: 采样频率 $f_{SAMPLING}=f_{DTS}/4$ , N=8 1111: 采样频率 $f_{SAMPLING}=f_{DTS}/32$ , N=8
位 3: 2	C1IDIV	0x0	rw	通道 1 分频系数 (Channel 1 input divider) 这些位定义了通道 1 的分频系数。 00: 不分频, 每一个有效的边沿都会产生一次输入; 01: 每 2 个有效的边沿产生一次输入; 10: 每 4 个有效的边沿产生一次输入; 11: 每 8 个有效的边沿产生一次输入。 注: C1EN='0' 时, 分频系数复位。
位 1: 0	C1C	0x0	rw	通道 1 配置 (Channel 1 configure) 当 C1EN='0' 时, 这些位用于选择通道 1 为输出或输入, 以及输入时的映射选择: 00: 输出; 01: 输入, C1IN 映射在 C1IRAW 上; 10: 输入, C1IN 映射在 C2IRAW 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、TMR8通道模式寄存器2 (TMRx\_CM2)

参看以上 CM1 寄存器描述

## 输出比较模式

域	简称	复位值	类型	功能
位 15	C4OSEN	0x0	rw	通道 4 输出开关使能 (Channel 4 output switch enable)
位 14: 12	C4OCTRL	0x0	rw	通道 4 输出控制 (Channel 4 output control)
位 11	C4OBEN	0x0	rw	通道 4 输出缓存使能 (Channel 4 output buffer enable)
位 10	C4OIEN	0x0	rw	通道 4 输出立即使能 (Channel 4 output immediately enable)
				通道 4 配置 (Channel 4 configure) 当 C4EN='0'时, 这些位用于选择通道 4 为输出或输入, 以及输入时的映射选择:
位 9: 8	C4C	0x0	rw	00: 输出; 01: 输入, C4IN 映射在 C4IRAW 上; 10: 输入, C4IN 映射在 C3IRAW 上; 11: 输入, C4IN 映射在 STCI 上, 只有在 STIS 选择内部触发输入时才工作。
位 7	C3OSEN	0x0	rw	通道 3 输出开关使能 (Channel 3 output switch enable)
位 6: 4	C3OCTRL	0x0	rw	通道 3 输出控制 (Channel 3 output control)
位 3	C3OBEN	0x0	rw	通道 3 输出缓存使能 (Channel 3 output buffer enable)
位 2	C3OIEN	0x0	rw	通道 3 输出立即使能 (Channel 3 output immediately enable)
				通道 3 配置 (Channel 3 configure) 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 C3IRAW 上; 10: 输入, C3IN 映射在 C4IRAW 上; 11: 输入, C3IN 映射在 STCI 上, 只有在 STIS 选择内部触发输入时才工作。

## 输入模式

域	简称	复位值	类型	功能
位 15: 12	C4DF	0x0	rw	通道 4 滤波器 (Channel 4 digital filter)
位 11: 10	C4IDIV	0x0	rw	通道 4 分频系数 (Channel 4 input divider)
				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 C4IRAW 上; 10: 输入, C4IN 映射在 C3IRAW 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 C3IRAW 上; 10: 输入, C3IN 映射在 C4IRAW 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。

## 14.4.4.9 TMR1、TMR8通道控制寄存器（TMRx\_CTRL）

域	简称	复位值	类型	功能
位 15: 14	保留	0x0	resd	保持默认值。
位 13	C4P	0x0	rw	通道 4 极性（Channel 4 polarity） 见 C1P 的描述。
位 12	C4EN	0x0	rw	通道 4 使能（Channel 4 enable） 见 C1EN 的描述。
位 11	C3CP	0x0	rw	通道 3 互补极性（Channel 3 complementary polarity） 见 C1P 的描述。
位 10	C3CEN	0x0	rw	通道 3 互补使能（Channel 3 complementary enable） 见 C1EN 的描述。
位 9	C3P	0x0	rw	通道 3 极性（Channel 3 polarity） 见 C1P 的描述。
位 8	C3EN	0x0	rw	通道 3 使能（Channel 3 enable） 见 C1EN 的描述。
位 7	C2CP	0x0	rw	通道 2 互补极性（Channel 2 complementary polarity） 见 C1P 的描述。
位 6	C2CEN	0x0	rw	通道 2 互补使能（Channel 2 complementary enable） 见 C1EN 的描述。
位 5	C2P	0x0	rw	通道 2 极性（Channel 2 polarity） 见 C1P 的描述。
位 4	C2EN	0x0	rw	通道 2 使能（Channel 2 enable） 见 C1EN 的描述。
位 3	C1CP	0x0	rw	通道 1 互补极性（Channel 1 complementary polarity） 0: C1COUT 的有效电平为高 1: C1COUT 的有效电平为低
位 2	C1CEN	0x0	rw	通道 1 互补使能（Channel 1 complementary enable） 0: 禁止输出； 1: 使能输出。
位 1	C1P	0x0	rw	通道 1 极性（Channel 1 polarity） 通道 1 配置为输出： 0: C1OUT 的有效电平为高 1: C1OUT 的有效电平为低 通道 1 配置为输入： 0: C1IN 的有效边沿为上升沿；作为外部触发使用时，C1IN 不反相。 1: C1IN 的有效边沿为下降沿；作为外部触发使用时，C1IN 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 带刹车功能的互补输出通道CxOUT和CxCOOUT的控制位

控制位					输出状态 (1)	
OEN 位	FCSODIS 位	FCSEN 位	CxEN 位	CxCEN 位	CxOUT 输出状态	CxCOOUT 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) CxCOOUT=0, CxCEN=0
		0	0	1	输出禁止 (与定时器断开) CxOUT=0, Cx_EN=0	CxORAW + 极性, CxCOOUT= 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) CxCOOUT=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) CxCOOUT=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, CxCOOUT= 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) CxCOOUT=CxCP, CxCEN=1
		1	1	1	CxORAW+极性+死区, Cx_EN=1	CxORAW 反相+极性+死区, CxCEN=1
0	0	X	0	0	输出禁止 (与定时器断开) 异步地: CxOUT=CxP, Cx_EN=0, CxCOOUT=CxCP, CxCEN=0; 若时钟存在: 经过一个死区时间后 CxOUT=CxIOS, CxCOOUT=CxCIOS, 假设 CxIOS 与 CxCIOS 并不都对应 CxOUT 和 CxCOOUT 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, CxCOOUT=CxCP, CxCEN=1; 若时钟存在: 经过一个死区 时间后 CxOUT =CxIOS, CxCOOUT=CxCIOS, 假设 CxIOS 与 CxCIOS 并不都对应 CxOUT 和 CxCOOUT 的有效电平。	
	0		0	1		
	0		1	0		
	0		1	1		
	1		0	0		
	1		0	1		
	1		1	0		
	1		1	1		

注意: 如果一个通道的 2 个输出都没有使用 (CxEN = CxCEN = 0), 那么 CxIOS, CxCIOS, CxP 和 CxCP 都必须清零。

注意: 管脚连接到互补的 CxOUT 和 CxCOOUT 通道的外部 I/O 管脚的状态, 取决于 CxOUT、CxCOOUT 通道状态和 GPIO 以及 IOMUX 寄存器。

## 14.4.4.10 TMR1、TMR8计数值 (TMRx\_CVAL)

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

## 14.4.4.11 TMR1、TMR8预分频器 (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 $f_{CK\_CNT} = f_{TMR\_CLK} / (DIV[15: 0] + 1)$ 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、TMR8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。

## 14.4.4.13 TMR1、TMR8重复周期寄存器 (TMRx\_RPR)

域	简称	复位值	类型	功能
位 15: 8	保留	0x00	resd	保持默认值。
位 7: 0	RPR	0x00	rw	重复周期的次数 (Repetition of period value) 这些位用于减慢溢出事件发生的速率，当重复周期的次数减为 0 时才会发生溢出事件。

## 14.4.4.14 TMR1、TMR8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。

## 14.4.4.15 TMR1、TMR8通道2数据寄存器 (TMRx\_C2DT)

域	简称	复位值	类型	功能
位 15: 0	C2DT	0x0000	rw	通道 2 数据寄存器值 (Channel 2 data register) 若通道 1 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、TMR8通道3数据寄存器 (TMRx\_C3DT)

域	简称	复位值	类型	功能
位 15: 0	C3DT	0x0000	rw	通道 3 数据寄存器值 (Channel 3 data register) 若通道 3 配置为输入： C3DT 是前一次通道 3 输入事件 (C3IN) 所保存的 CVAL。 若通道 3 配置为输出： C3DT 是将要和 CVAL 进行比较的值，写入的值是否会立即生效取决于输出缓存使能位 (C3OBEN)，并根据设置在 C3OUT 上产生相应的输出。

## 14.4.4.17 TMR1、TMR8通道4数据寄存器（TMRx\_C4DT）

域	简称	复位值	类型	功能
位 15: 0	C4DT	0x0000	rw	通道 4 数据寄存器值（Channel 4 data register） 若通道 4 配置为输入： C4DT 是前一次通道 4 输入事件（C4IN）所保存的 CVAL。 若通道 4 配置为输出： C4DT 是将要和 CVAL 进行比较的值，写入的值是否会立即生效取决于输出缓存使能位（C4OBEN），并根据设置在 C4OUT 上产生相应的输出。

## 14.4.4.18 TMR1、TMR8刹车寄存器（TMRx\_BRK）

域	简称	复位值	类型	功能
位 15	OEN	0x0	rw	输出使能（Output enable） 对配置为输出的通道，该位用于使能 CxOUT 和 CxCOUT 的输出。 0：关闭； 1：开启。
位 14	AOEN	0x0	rw	输出自动使能(Automatic output enable) 用于溢出事件时将 OEN 自动置'1' 0：关闭； 1：开启
位 13	BRKV	0x0	rw	刹车输入信号的有效性（Brake input validity） 用于选择刹车输入信号的输入有效电平： 0：低电平； 1：高电平。
位 12	BRKEN	0x0	rw	刹车功能使能（Brake enable） 用于开启刹车功能。 0：关闭； 1：开启。
位 11	FCSOEN	0x0	rw	总输出开时的冻结状态（Frozen channel status when holistic output enable） 该位用于配置具有互补输出的通道，在定时器不工作且 MOEN=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 MOEN=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：DTC、BRKEN、BRKV 和 AOEN TMRx_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： 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_CMx：C2OCTRL 和 C2OBEN 注：WPC>0 时将无法再次被修改，直到系统复位。
位 7: 0	DTC	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、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。

## 14.4.4.19 TMR1、TMR8 DMA控制寄存器（TMRx\_DMACTRL）

域	简称	复位值	类型	功能
位 15: 13	保留	0x0	resd	保持默认值。
位 12: 8	DTB	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, .....

## 14.4.4.20 TMR1、TMR8 DMA数据寄存器（TMRx\_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。

## 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载计数器。

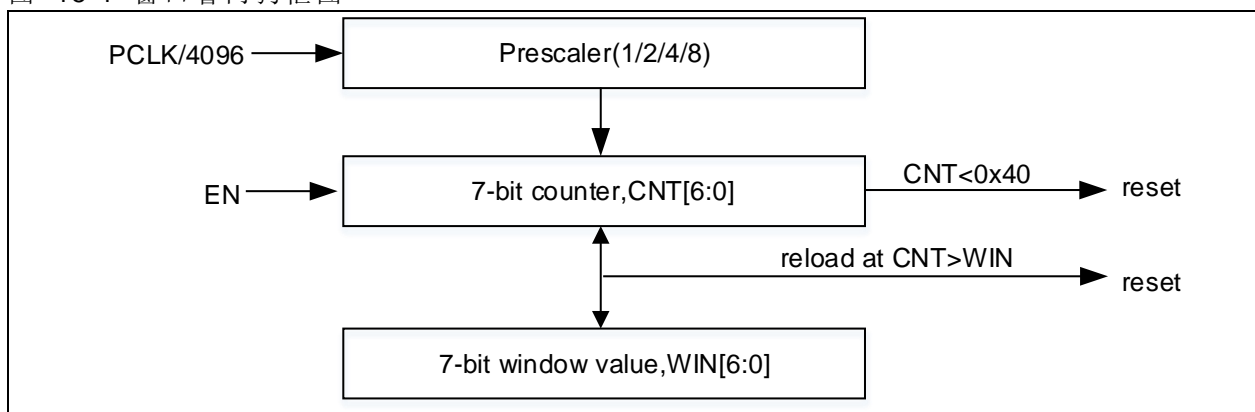
### 15.3 WWDT功能描述

启动窗口看门狗后，窗口看门狗可在以下两种情况下产生系统复位：

第一种，7 位递减计数器值由 0x40 变为 0x3F。

第二种，7 位递减计数器值大于 7 位窗口值时，重载计数器值。

图 15-1 窗口看门狗框图



为避免重载计数器值时产生复位，应在计数器值小于窗口值大于 0x40 时重载计数器值。

WWDT 计数器时钟由 APB1\_CLK 分频得到，分频系数可通过配置配置寄存器（WWDT\_CFG）DIV[1:0]改变。计数器值决定了 WWDT 复位前的最大计数周期数，结合 WIN[6:0]可灵活的调整重载窗口。

WWDT 提供了重载计数器中断功能，开启后，WWDT 将在计数值达到 0x40h 时将 RLDF 标志位置 1，同时产生重载计数器中断，可在中断服务程序中重载计数器值，以避免发生系统复位。需要注意的是，若在 CNT[6]为 0 时，将 WWDTEN 置 1 会产生一个系统复位，因此当写入控制寄存器（WWDT\_CTRL）时，应始终保持 CNT[6]为 1，避免使能窗口看门狗后立即产生一个系统复位。

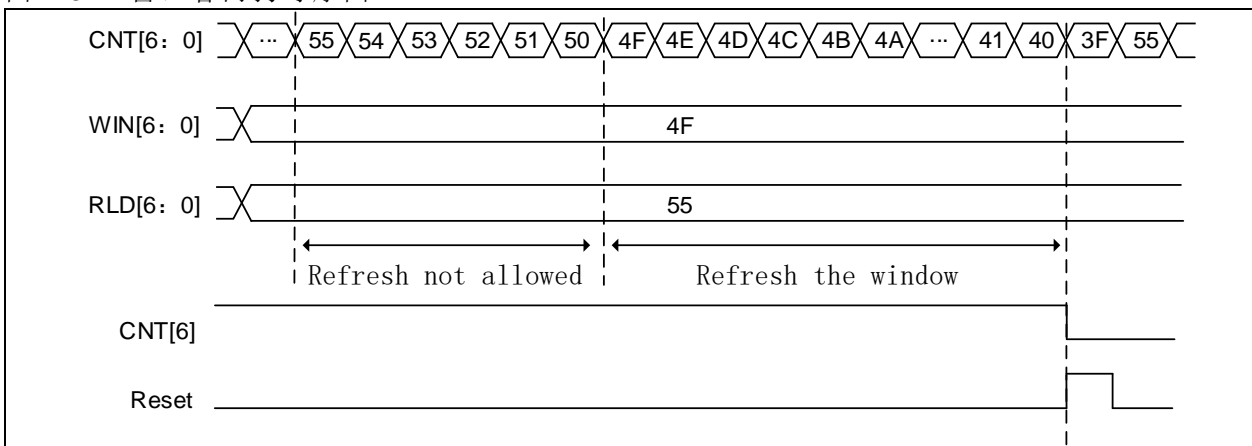
窗口看门狗超时时间  $T_{WWDT}$  可由一下公式计算，其中  $T_{PCLK1}$  为 APB1 时钟周期，单位为 ms：

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

表 15-1 PCLK1频率为72MHz时，最大和最小看门狗超时时间

时钟预分频值	最小超时时间	最大超时时间
0	56.5 $\mu s$	3.64ms
1	113.5 $\mu s$	7.28ms
2	227.5 $\mu s$	14.56ms
3	455 $\mu s$	29.12ms

图 15-2 窗口看门狗时序图



## 15.4 调试模式

微控制器处于调试模式时，意味着 Cortex™-M4F 核心停止。将 DEBUG 模块中 WWDT\_PAUSE 位置 1 可将 WWDT 计数器计数暂停。

## 15.5 WWDT寄存器

必须以字（32 位）的方式操作这些外设寄存器。

表 15-2 WWDT寄存器的映像和复位值

寄存器简称	基址偏移量	复位值
WWDT_CTRL	0x00	0x7F
WWDT_CFG	0x04	0x7F
WWDT_STS	0x08	0x00

### 15.5.1 控制寄存器（WWDT\_CTRL）

域	简称	复位值	类型	功能
位 31: 8	保留	0x000000	resd	保持默认值。
位 7	WWDTEN	0x0	rw1s	窗口看门狗使能（Window watchdog enable） 0：关闭； 1：开启。 该位由软件置起，只能在复位后自动清零。
位 6: 0	CNT	0x7F	rw	递减计数器（Decrement counter） 当计数器递减到 0x3F 时产生复位。

## 15.5.2 配置寄存器（WWDT\_CFG）

域	简称	复位值	类型	功能
位 31: 10	保留	0x000000	resd	保持默认值。
位 9	RLDIEN	0x0	rw	重载计数器中断（Reload counter interrupt） 0：关闭； 1：开启。
位 8: 7	DIV	0x0	rw	时钟预分频值（Clock division value） 00：PCLK1 除以 4096； 01：PCLK1 除以 8192； 10：PCLK1 除以 16384； 11：PCLK1 除以 32768。
位 6: 0	WIN	0x7F	rw	窗口值（Window value） 当计数器值大于窗口值时，此时重载计数器会产生复位， 重载计数器区间为 0x40~WIN[6: 0]

## 15.5.3 状态寄存器（WWDT\_STS）

域	简称	复位值	类型	功能
位 31: 1	保留	0x0000 0000	resd	保持默认值。
位 0	RLDF	0x0	rw0c	重载计数器中断标志（Reload counter interrupt flag） 当递减计数器为 0x40 时，该标志会置位。 该位被硬件置起，由软件将其清零。



## 16 看门狗（WDT）

### 16.1 WDT简介

看门狗由专用低速时钟（LICK）驱动，由于 LICK 时钟精度较低，因此看门狗适用于低时间精度、能够独立于主程序之外的应用

### 16.2 WDT主要特性

- 12位递减计数器
- 计数器由LICK时钟驱动（可在深睡眠和待机模式下工作）
- 看门狗使能后，将在计数器计数至0时产生WDT系统复位

### 16.3 WDT功能描述

#### WDT 启动方式：

WDT 的启动方式有两种，分别为软件启动和硬件启动。软件启动通过向命令寄存器（WDT\_CMD）写入 0xCCCC 实现；硬件启动则需通过配置用户系统数据区来实现，使能硬件看门狗后，看门狗将在上电复位后自动开始运行。

#### WDT 复位条件：

当 WDT 计数器值递减至 0 时将产生 WDT 系统复位，因此需定时向命令寄存器（WDT\_CMD）写入 0xAAAA 重载计数器值。

#### WDT 写保护：

预分频寄存器（WDT\_DIV）、重载寄存器（WDT\_RLD）受写保护，向命令寄存器（WDT\_CMD）写入 0x5555 可解锁寄存器写保护，之后可对其进行配置。这两个寄存器的更新状态分别由 WDT\_STS 寄存器中 DIVF、RLDF 指示。向命令寄存器（WDT\_CMD）写入其它值将重新启动预分频寄存器（WDT\_DIV）、重载寄存器（WDT\_RLD）写保护。向命令寄存器（WDT\_CMD）写入 0xAAAA 也会启动寄存器写保护。

#### WDT 时钟：

WDT 计数器由 LICK 时钟驱动，LICK 是内部 RC 时钟，其典型值为 40kHz，范围为 30kHz~60kHz 之间，所以超时时间也是在一定区间内，使用时应注意在超时时间配置上应该留有余量，如果需要获得较为精确的看门狗超时时间，可对 LICK 进行校准，有关 LICK 校准的问题，详见 [4.1.1 节](#)。

图 16-1 看门狗框图

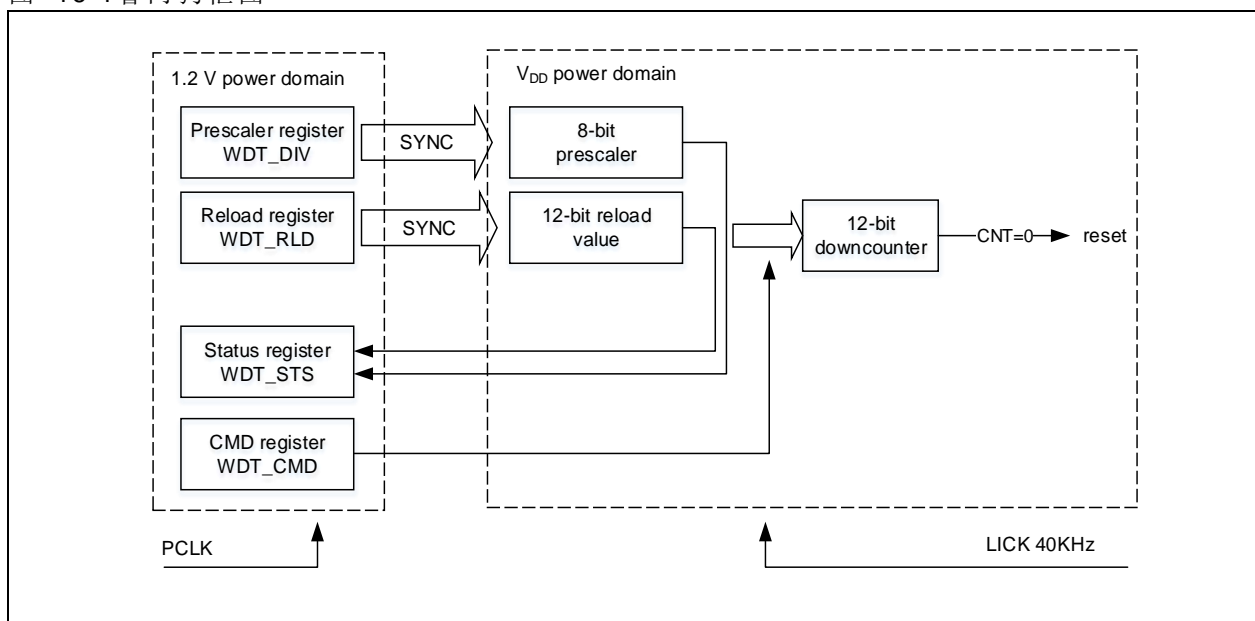


表 16-1 看门狗超时时间 (LICK=40kHz)

预分频系数	DIV[2: 0]位	最短时间 (ms) RLD[11: 0] = 0x000	最长时间 (ms) RLD[11: 0] = 0xFFFF
/4	0	0.1	409.6
/8	1	0.2	819.2
/16	2	0.4	1638.4
/32	3	0.8	3276.8
/64	4	1.6	6553.6
/128	5	3.2	13107.2
/256	(6 或 7)	6.4	26214.4

## 16.4 调试模式

微控制器处于调试模式时，意味着 Cortex™-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寄存器的映像和复位值

寄存器简称	基址偏移量	复位值
WDT_CMD	0x00	0x0000 0000
WDT_DIV	0x04	0x0000 0000
WDT_RLD	0x08	0x0000 0FFF
WDT_STS	0x0C	0x0000 0000

### 16.5.1 命令寄存器 (WDT\_CMD)

(在待机模式复位)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0000	resd	保持默认值。
位 15: 0	CMD	0x0000	wo	命令寄存器 (Command register) 0xAAAA: 重载计数器; 0x5555: 解锁 WDT_DIV 和 WDT_RLD 写保护; 0xCCCC: 启动看门狗, 如果使能了硬件看门狗, 则不需要执行此操作。

## 16.5.2 预分频寄存器 (WDT\_DIV)

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。
位 2: 0	DIV	0x0	rw	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。

## 16.5.3 重装载寄存器 (WDT\_RLD)

(待机模式时复位)

域	简称	复位值	类型	功能
位 31: 12	保留	0x00000	resd	保持默认值。
位 11: 0	RLD	0xFFFF	rw	重载值 (Reload value) 只有解锁写保护后才能写此寄存器，只有当 RLDF 为 0 时，才能读取此寄存器。

## 16.5.4 状态寄存器 (WDT\_STS)

域	简称	复位值	类型	功能
位 31: 2	保留	0x0000 0000	resd	保持默认值。
				重载值更新完成标志 (Reload value update complete flag)
				0: 更新完成;
				1: 正在更新。
				只有当 RLDF 为 0 时才能写重装载寄存器 (WDT_RLD)。
				分频值更新完成标志 (Division value update complete flag)
				0: 更新完成;
				1: 正在更新。
				只有当 DIVF 为 0 时才能写预分频寄存器 (WDT_DIV)。



## 17.4 RTC功能描述

### 17.4.1 RTC寄存器配置

上电复位后所有 RTC 寄存器处于写保护状态，需要先解除写保护，才能写 RTC 寄存器  
解锁步骤：

- 使能电源和电池供电域接口时钟：APB1 外设时钟使能寄存器（CRM\_APB1EN）的 PWCEN =1, BPREN=1

- 解锁电池供电域写保护：电源控制寄存器(PWC\_CTRL)的 BPWEN=1

DIV、CNT、ALA 寄存器配置：

需要先进入配置模式（CFGEN = 1），然后才能对寄存器进行写操作，当退出配置模式（CFGEN = 0）时，就会将寄存器值实际写到电池供电域，这个过程至少需要 3 个 RTCCLK 周期。

由于同步逻辑的存在，需要确保上一次的 RTC 寄存器配置完成后（CFGF = 1），才能进行新的写操作。

配置过程：

1. 等待寄存器同步完成（CFGF 位置 1）。
2. 进入配置模式（CFGEN 位置 1）。
3. 根据需要配置相关 RTC 寄存器。
4. 退出配置模式将（CFGEN 清 0），
5. 等待寄存器同步完成（CFGF 位置 1）。

DIV、ALA、CNT 和 DIVCNT 寄存器只能通过电池供电域复位信号复位，除此之外的所有系统寄存器都由系统复位或电源复位进行异步复位。

### 17.4.2 RTC寄存器读取

由于同步逻辑的存在，当在系统复位、电源复位、从待机、深度睡眠模式唤醒后，有可能在读取 RTC 寄存器的时候，正确的寄存器值还未从电池供电域更新到 APB1 接口，所以需等待寄存器更新标志(UPDF)位置 1 后，再读取 RTC 寄存器，否则可能会读出错误值。

### 17.4.3 RTC中断

RTC 支持以下中断：

- 秒中断：若秒中断使能（TSIEN=1），在每个 LN\_CLK 周期产生一个秒中断；
- 闹钟中断：若闹钟中断使能（寄存器 TAIEN=1），在 TA 寄存器值与 CNT 值相等时，产生闹钟中断；
- 溢出中断：若溢出中断使能（OVFIEN=1），当计数器计到 0xFFFFFFFF 时，产生溢出中断。

RTC 支持 RTC 全局中断向量（RTC\_IRQn）和 RTC 闹钟中断向量（RTCAAlarm\_IRQn）。若要使用 RTC 闹钟中断从 DEEPSLEEP 模式下唤醒，需使能 RTC 闹钟中断并使用 RTCAAlarm\_IRQn 向量，同时将 EXINT 线 17 配置为中断模式；若要使用 RTC 闹钟事件从 DEEPSLEEP 模式下唤醒，需要将 EXINT 线 17 配置为事件模式，但无需使能 RTC 闹钟中断，若要使用 RTC 闹钟事件从 standby 模式下唤醒，则无需配置闹钟中断和 EXINT 线 17。

对应的 RTC 标志位如下：

- RTC 秒标志（TSF）：RTC 计数器更新标志，RTC 计数器值更新前一个 RTC\_CLK 被置 1。
- RTC 闹钟标志（TAF）：计数器的值到达闹钟寄存器的值加 1（TA+1）前一个 RTC\_CLK 被置 1。
- RTC 溢出标志（OVFF）：RTC 计数器溢出标志，RTC 计数器值到达 0x00000000 前一个 RTC\_CLK 被置 1。

当 RTC 中断已产生，清除对应的标志位表示所请求的中断已被接受，且任何标志位仅能由硬件置 1 软件清 0。在复位后，所有的中断将被禁止；在 APB1 时钟停止运行时，标志位将不再更新。

图 17-2 RTC秒和闹钟波形图示例，DIV=0004，TA=00003

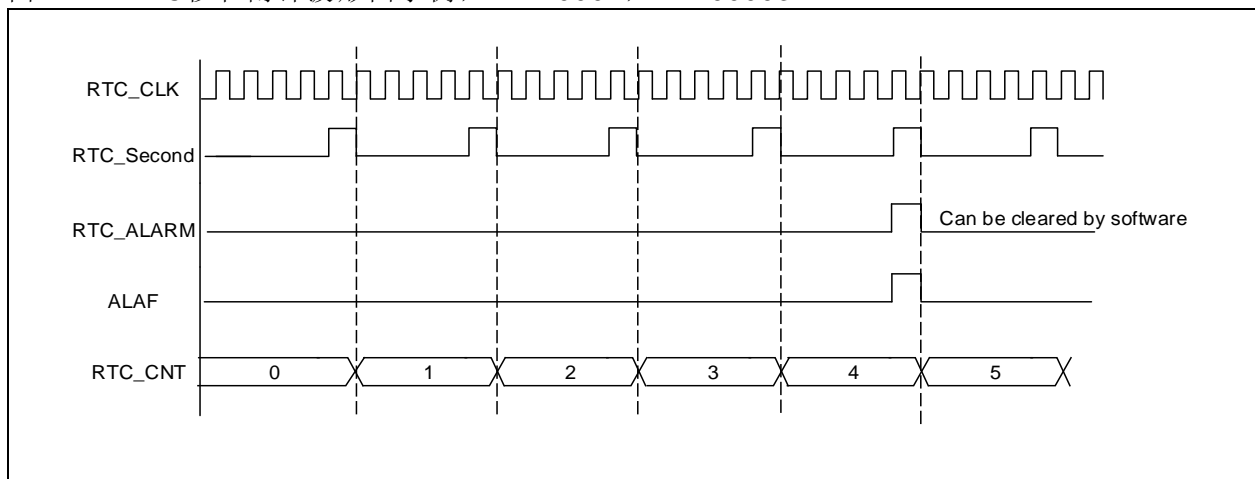
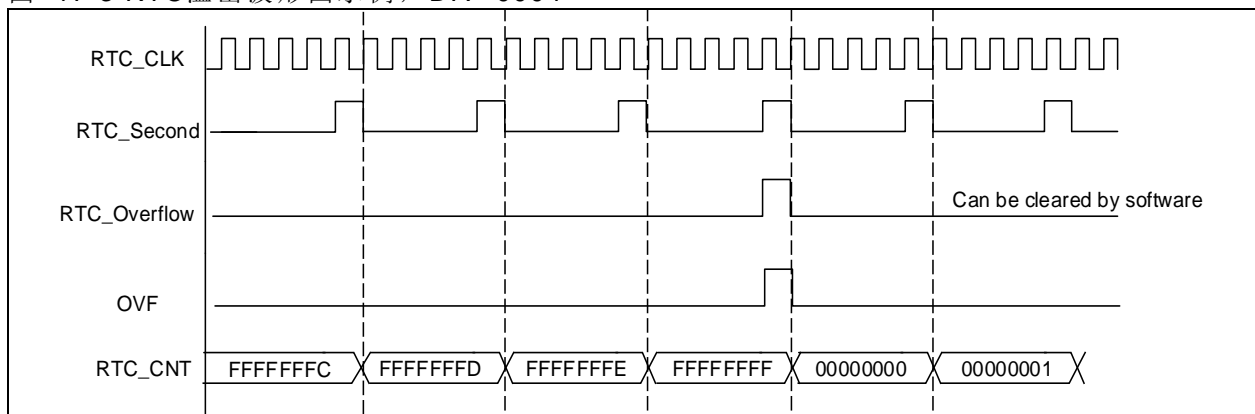


图 17-3 RTC溢出波形图示例，DIV=0004



## 17.5 RTC寄存器描述

必须以字（32 位）的方式操作这些外设寄存器。

RTC 寄存器是 16 位可寻址寄存器，具体描述如下：

表 17-1 RTC-寄存器映像和复位值

寄存器简称	基址偏移量	复位值
RTC_CTRLH	0x00	0x0000
RTC_CTRLH	0x04	0x0020
RTC_DIVH	0x08	0x0000
RTC_DIVL	0x0C	0x8000
RTC_DIVCNTH	0x10	0x0000
RTC_DIVCNTL	0x14	0x8000
RTC_CNTH	0x18	0x0000
RTC_CNTL	0x1C	0x0000
RTC_TAH	0x20	0xFFFF
RTC_TAL	0x24	0xFFFF

## 17.5.1 RTC控制寄存器高位 (RTC\_CTRLH)

域	简称	复位值	类型	功能
位 15: 3	保留	0x0000	resd	保持默认值。
位 2	OVFIEN	0x0	rw	溢出中断使能 (Overflow interrupt enable) 用于使能溢出中断。 0: 关闭; 1: 开启。
位 1	TAIEN	0x0	rw	闹钟中断使能 (Time alarm interrupt enable) 用于使能闹钟中断。 0: 关闭; 1: 开启。
位 0	TSIEN	0x0	rw	秒中断使能 (Time second interrupt enable) 用于使能秒中断。 0: 关闭; 1: 开启。

注意：系统复位后该寄存器被复位，关于 RTC 寄存器的配置过程见 [17.4.1](#) 节。

## 17.5.2 RTC控制寄存器低位 (RTC\_CTRL)

域	简称	复位值	类型	功能
位 15: 6	保留	0x000	resd	保持默认值。
位 5	CFGF	0x1	ro	RTC 配置完成 (RTC configuration finish) 该位用于检查前一次对 RTC 寄存器的写入是否完成，为'1'时，才能再次写入 RTC 寄存器。 0: 未完成; 1: 完成。
位 4	CFGEN	0x0	rw	RTC 配置使能 (RTC Configuration enable) 该位用于进入配置模式，置'1'时才能对 CNT、ALA、DIVCNT 进行写入。 0: 退出配置模式; 1: 进入配置模式。
位 3	UPDF	0x0	rw0c	RTC 更新标志 (RTC update finish) 该位用于指示 RTC 寄存器是否被更新完成。当 CNT 和 DIVCNT 被更新时，由硬件置'1'。在读取数据之前需要软件清除该位，然后等待该位置'1'后再进行读取。 0: 未更新; 1: 已更新。
位 2	OVFF	0x0	rw0c	溢出标志 (Overflow flag) 该位用于检查计数器是否发生溢出。如果 OVFIEN =1，则产生中断。 0: 未溢出; 1: 溢出。
位 1	TAF	0x0	rw0c	闹钟标志 (Time alarm flag) 该位用于检查是否发生了闹钟事件。如果 TAIEN =1，则产生中断。 0: 无闹钟; 1: 有闹钟。
位 0	TSF	0x0	rw0c	秒标志 (Time second flag) 该位用于检查是否发生了秒事件。如果 TSIEN =1，则产生中断。 0: 无秒事件; 1: 秒事件产生。



### 17.5.3 RTC分频系数寄存器（RTC\_DIVH/RTC\_DIVL）

#### RTC 分频系数寄存器高位（RTC\_DIVH）

域	简称	复位值	类型	功能
位 15: 4	保留	0x000	resd	保持默认值。
位 3: 0	DIV	0x0	wo	RTC 分频系数（RTC divider） 这些位用来配置计数器的时钟频率。时钟频率为： $f_{LN\_CLK} = f_{RTCCLK} / (DIV[19: 0] + 1)$

#### RTC 分频系数寄存器低位（RTC\_DIVL）

域	简称	复位值	类型	功能
位 15: 0	DIV	0x8000	wo	RTC 分频系数（RTC divider） 这些位用来配置计数器的分频系数。时钟频率为： $f_{LN\_CLK} = f_{RTCCLK} / (DIV[19: 0] + 1)$ 注：建议分频系数配置为非 0 值。

### 17.5.4 RTC分频计数寄存器（RTC\_DIVCNTH/RTC\_DIVCNTL）

#### RTC 分频计数寄存器高位（RTC\_DIVCNTH）

域	简称	复位值	类型	功能
位 15: 4	保留	0x000	resd	保持默认值。
位 3: 0	DIVCNT	0x0	ro	RTC 分频计数值（RTC clock divider counter）

#### RTC 分频计数寄存器低位（RTC\_DIVCNTL）

域	简称	复位值	类型	功能
位 15: 0	DIVCNT	0x8000	ro	RTC 分频计数值（RTC clock divider counter）

### 17.5.5 RTC计数值寄存器（RTC\_CNTH/RTC\_CNTL）

#### RTC 计数值寄存器高位（RTC\_CNTH）

域	简称	复位值	类型	功能
位 15: 0	CNT	0x0000	rw	RTC 计数值（RTC counter value） 这些位用来配置或读取 RTC 计数值的高位。

#### RTC 计数值寄存器低位（RTC\_CNTL）

域	简称	复位值	类型	功能
位 15: 0	CNT	0x0000	rw	RTC 计数值（RTC counter value） 这些位用来配置或读取 RTC 计数值的低位。

### 17.5.6 RTC闹钟寄存器（RTC\_TAH/RTC\_TAL）

#### RTC 闹钟值寄存器高位（RTC\_TAH）

域	简称	复位值	类型	功能
位 15: 0	TA	0xFFFF	wo	RTC 闹钟值（Time alarm clock value） 这些位用来配置闹钟值的高位。

#### RTC 闹钟寄存器低位（RTC\_TAL）

域	简称	复位值	类型	功能
位 15: 0	TA	0xFFFF	wo	RTC 闹钟值（Time alarm clock value） 这些位用来配置闹钟值的低位。

## 18 电池供电寄存器（BPR）

### 18.1 BPR简介

电池供电寄存器位于电池供电域中，由 VDD/VBAT 维持供电。电池供电寄存器有 42 个 16 位寄存器，当在发生入侵事件或电池供电域复位时，寄存器内容被清除，最大限度保证了数据的安全。

### 18.2 BPR特性

- 多达 42 个 16 位寄存器
- 支持入侵事件复位寄存器
- PC13 管脚复用功能输出配置

### 18.3 BPR功能描述

要解锁电池供电寄存器的访问，要将 PWCEN、BPREN、BPWEN 位置 1。

BPR 提供了入侵检测功能来保证数据的安全，使能 TAMPER 管脚后，通过 TPP 位配置有效的入侵电平极性。当检测到入侵事件后，TPEF 标志位将置 1，同时清除电池供电寄存器；若已使能入侵中断，将产生入侵中断，同时 TPIF 标志位置 1。

BPR 还提供了 RTC 校准功能，通过配置 CALVAL[6: 0]，最多可减慢 RTC 时钟 121ppm。若使能 RTC 校准输出，TAMPER 管脚将输出校准后的 64 分频 RTC 时钟（CCOS 置 1）。

注：当 TPP=0/1 时，设置 TPEN 位使能之前 TAMPER 管脚已经为高电平/低电平，TPEN 置 1 后会产生一个额外的侵入事件，尽管 TAMPER 管脚上没有上升/下降沿信号。

### 18.4 BPR寄存器描述

必须以字（32 位）的方式操作这些外设寄存器。

BPR 寄存器是 16 位的可寻址寄存器。

表 18-1 BPR寄存器映像和复位值

寄存器简称	基址偏移量	复位值
BPR_DT1	0x04	0x0000 0000
BPR_DT2	0x08	0x0000 0000
BPR_DT3	0x0C	0x0000 0000
BPR_DT4	0x10	0x0000 0000
BPR_DT5	0x14	0x0000 0000
BPR_DT6	0x18	0x0000 0000
BPR_DT7	0x1C	0x0000 0000
BPR_DT8	0x20	0x0000 0000
BPR_DT9	0x24	0x0000 0000
BPR_DT10	0x28	0x0000 0000
BPR_RTCCAL	0x2C	0x0000 0000
BPR_CTRL	0x30	0x0000 0000
BPR_CTRLSTS	0x34	0x0000 0000
BPR_DT11	0x40	0x0000 0000
BPR_DT12	0x44	0x0000 0000
BPR_DT13	0x48	0x0000 0000
BPR_DT14	0x4C	0x0000 0000
BPR_DT15	0x50	0x0000 0000

BPR_DT16	0x54	0x0000 0000
BPR_DT17	0x58	0x0000 0000
BPR_DT18	0x5C	0x0000 0000
BPR_DT19	0x60	0x0000 0000
BPR_DT20	0x64	0x0000 0000
BPR_DT21	0x68	0x0000 0000
BPR_DT22	0x6C	0x0000 0000
BPR_DT23	0x70	0x0000 0000
BPR_DT24	0x74	0x0000 0000
BPR_DT25	0x78	0x0000 0000
BPR_DT26	0x7C	0x0000 0000
BPR_DT27	0x80	0x0000 0000
BPR_DT28	0x84	0x0000 0000
BPR_DT29	0x88	0x0000 0000
BPR_DT30	0x8C	0x0000 0000
BPR_DT31	0x90	0x0000 0000
BPR_DT32	0x94	0x0000 0000
BPR_DT33	0x98	0x0000 0000
BPR_DT34	0x9C	0x0000 0000
BPR_DT35	0xA0	0x0000 0000
BPR_DT36	0xA4	0x0000 0000
BPR_DT37	0xA8	0x0000 0000
BPR_DT38	0xAC	0x0000 0000
BPR_DT39	0xB0	0x0000 0000
BPR_DT40	0xB4	0x0000 0000
BPR_DT41	0xB8	0x0000 0000
BPR_DT42	0xBC	0x0000 0000

## 18.4.1 电池供电数据寄存器x (BPR\_DTx) (x = 1 ... 42)

域	简称	复位值	类型	功能
位 15: 0	DT	0x0000	rw	电池供电域数据 (Battery powered domain data) 可用于保存数据。 BPR_DTx 寄存器只能通过电池供电域复位或入侵事件进行复位。

## 18.4.2 RTC校准寄存器 (BPR\_RTCCAL)

域	简称	复位值	类型	功能
位 15: 12	保留	0x0	resd	保持默认值。
位 11	OUTM	0x0	rw	输出模式 (Output mode) 用于选择闹钟或秒输出的输出模式: 0: 脉冲输出 (输出脉冲的宽度为一个 RTC 时钟的周期); 1: 翻转输出 (每来一次闹钟或秒输出事件, 相对应管脚翻转一次)。 注: 该位只能被电池供电域的复位所清除
位 10	CCOS	0x0	rw	校准时钟输出选择 (Calibration clock output selection) 0: 校准前的时钟 64 分频输出 1: 校准后的时钟 64 分频输出 注: 该位只能被电池供电域的复位所清除
位 9	OUTSEL	0x0	rw	输出选择 (Output selection) 该位用于选择在 TAMPER 管脚上输出的事件。 0: 输出 RTC 闹钟事件; 1: 输出秒事件。 注: 该位只能被电池供电域的复位所清除
位 8	OUTEN	0x0	rw	输出使能 (Output enable) 0: 关闭 1: 开启 注: 该位只能被电池供电域的复位所清除, 该位用于使能在 TAMPER 管脚上输出的事件。输出使能之后不能使用 TAMPER 功能。
位 7	CALOUT	0x0	rw	校准时钟输出 (Calibration clock output) 0: 无作用; 1: 在 TAMPER 管脚输出 64 分频后的 RTC 时钟。 校准时钟输出使能之后不能使用 TAMPER 功能。 注: 当 VDD 供电断开时, 该位被清除。
位 6: 0	CALVAL	0x00	rw	校准值 (Calibration value) 表示在一个周期内 ( $2^{20}$ 个时钟) 被过滤的时钟数量。 将会以 $1000000/2^{20}$ ppm 的最小精度来降低时钟频率, 降低的范围是 0~121ppm。

## 18.4.3 电池供电控制寄存器 (BPR\_CTRL)

域	简称	复位值	类型	功能
位 15: 2	保留	0x0000	resd	保持默认值。
位 1	TPP	0x0	rw	TAMPER 管脚极性 (TAMPER pin polarity) TAMPER 管脚极性选择, 检测到有效电平后会清除数据寄存器中的数据。 0: 高电平有效; 1: 低电平有效。 注: 为避免产生由于误操作引起的入侵事件, 建议在未使能 TAMPER 管脚时更改 TAMPER 管脚极性。
位 0	TPEN	0x0	rw	TAMPER 管脚使能 (TAMPER pin enable) 0: 关闭, TAMPER 管脚可做 GPIO 使用; 1: 开启。

## 18.4.4 电池供电控制/状态寄存器（BPR\_CTRLSTS）

域	简称	复位值	类型	功能
位 15: 10	保留	0x00	resd	保持默认值。
位 9	TPIF	0x0	ro	TAMPER 中断标志（Tamper interrupt flag） 当 TPIEN 位为 1 时检测到有入侵事件，则会被置 1。 0：未检测到入侵事件； 1：检测到入侵事件。 注：只有系统复位或退出待机模式后才被复位。
位 8	TPEF	0x0	ro	TAMPER 事件标志（Tamper event flag） 检测到入侵事件时会被置 1。 0：未检测到入侵事件； 1：检测到入侵事件。 注：入侵事件会导致 BPR_DT <sub>x</sub> 寄存器被复位，当 TPEF=1 时，请勿写入 BPR_DT <sub>x</sub> 寄存器。
位 7: 3	保留	0x00	resd	保持默认值。
位 2	TPIEN	0x0	rw	TAMPER 管脚中断使能（Tamper pin interrupt enable） 0：关闭； 1：开启。 注：入侵中断无法将系统内核从低功耗模式唤醒。
位 1	TPIFCLR	0x0	wo	TAMPER 中断清除（Tamper interrupt flag clear） 用于清除 TAMPER 中断 0：无效； 1：清除。
位 0	TPEFCLR	0x0	wo	TAMPER 事件清除（Tamper event flag clear） 用于清除 TAMPER 事件 0：无效； 1：清除。

## 19 模拟/数字转换（ADC）

### 19.1 ADC简介

ADC 是一个将模拟输入信号转换为 12 位数字信号的外设。采样率最高可达 2MSPS。多达 18 个通道源可进行采样及转换。

### 19.2 ADC主要特征

模拟方面有以下特征：

- 支持分辨率 12 位的转换
- 自校准时间：154 个 ADC 时钟周期
- ADC 转换时间
- ADC 时钟在最大频率 28MHz 时转换时间为 0.5  $\mu$ s（系统时钟为 240MHz 时，ADC 时钟最大频率为 20M,转换时间则为 0.7  $\mu$ s）
- ADC 供电要求：2.6V 到 3.6V
- ADC 输入范围： $V_{REF-} \leq V_{IN} \leq V_{REF+}$

数字控制方面有以下特征：

- 通道管理区分优先权不同的普通通道与抢占通道
- 普通通道与抢占通道具备各自独立的触发检测电路
- 各通道均可独立配置采样时间
- 转换顺序管理支持多种不同的多通道转换
- 可选择的数据对齐方式
- 可配置的电压监测边界
- 支持 DMA 传输的普通通道数据
- 可设定以下事件发生时响应中断
  - 抢占通道组转换结束
  - 通道转换结束
  - 电压监测超出范围
  - 联动多 ADC 的主从模式

### 19.3 ADC架构

ADC1的架构如图 19-1所示。

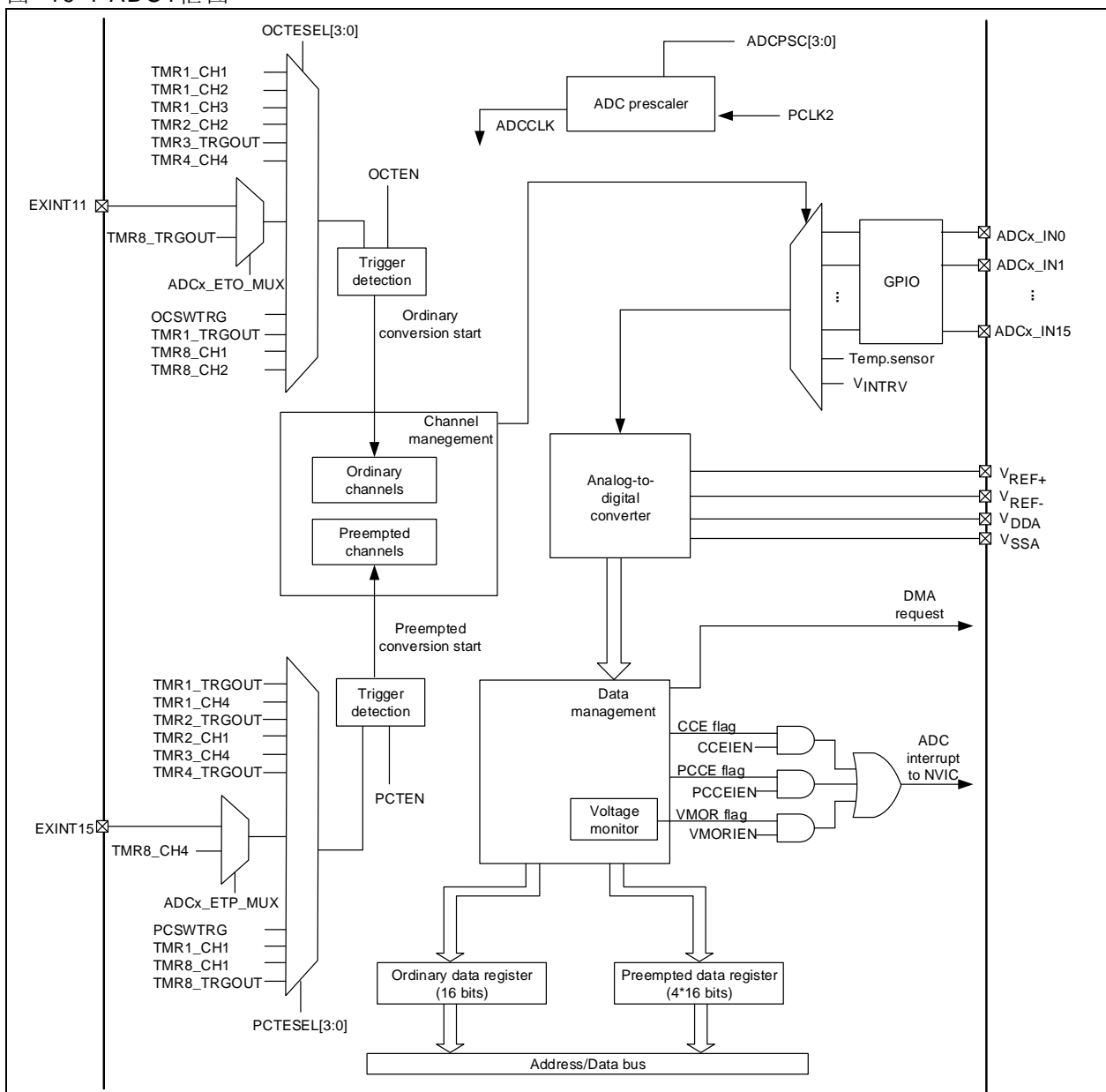
ADC2 与 ADC1 不同之处在于：

1. ADC2没有连接内部温度传感器（Temp. sensor）与内部参考电压（ $V_{INTRV}$ ）。
2. ADC2没有DMA request，可参考19.4.4.2章说明。

ADC3 与 ADC1 不同之处在于：

1. ADC3没有连接内部温度传感器（Temp. sensor）与内部参考电压（ $V_{INTRV}$ ）。
2. ADC3的外部模拟输入通道管脚数目不同，19.4.1章详述各个ADC的模拟通道。
3. ADC3的触发来源不同，19.4.2.2章详列各个ADC的触发来源。

图 19-1 ADC1框图



输入管脚介绍:

- $V_{DDA}$ : 模拟电源, ADC 模拟电源, 可与  $V_{DD}$  连接, 或  $2.6V \leq V_{DDA} \leq V_{DD}$  ( $3.6V$ )
- $V_{SSA}$ : 模拟电源地, ADC 模拟电源地, 必须与  $V_{SS}$  连接
- $V_{REF+}$ : 模拟参考正极, ADC 使用的高端/正极模拟参考电压,  $2.0V \leq V_{REF+} \leq V_{DDA}$
- $V_{REF-}$ : 模拟参考负极, ADC 使用的低端/负极参考电压, 必须与  $V_{SS}$  连接
- $ADCx\_IN$ : 模拟输入信号通道

## 19.4 ADC功能介绍

### 19.4.1 通道管理

模拟信号通道输入

每个 ADC 拥有多达 18 个模拟信号通道输入, 以  $ADC\_INx$  表示,  $x=0$  至 17。

- $ADC1\_IN0$  至  $ADC1\_IN15$  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。
- $ADC2\_IN0$  至  $ADC2\_IN15$  为外部模拟输入,  $ADC2\_IN16$  与  $ADC2\_IN17$  为  $V_{SS}$ 。
- $ADC3\_IN0$  至  $ADC3\_IN3$ 、 $ADC3\_IN10$  至  $ADC3\_IN13$  为外部模拟输入, 其余为  $V_{SS}$ 。



### 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。

#### 19.4.1.1 内部温度传感器

温度传感器接到 ADC1\_IN16，必须先使能 ADC 控制寄存器 2（ADC\_CTRL2）的 ITSRVEN 位并且等待上电时间后才可对温度传感通道进行转换。

转换后获得的数据，搭配数据手册的电气特性章节提供的 25° C 的电压值与数据对温度斜率(Avg\_Slope)，即可推算温度。

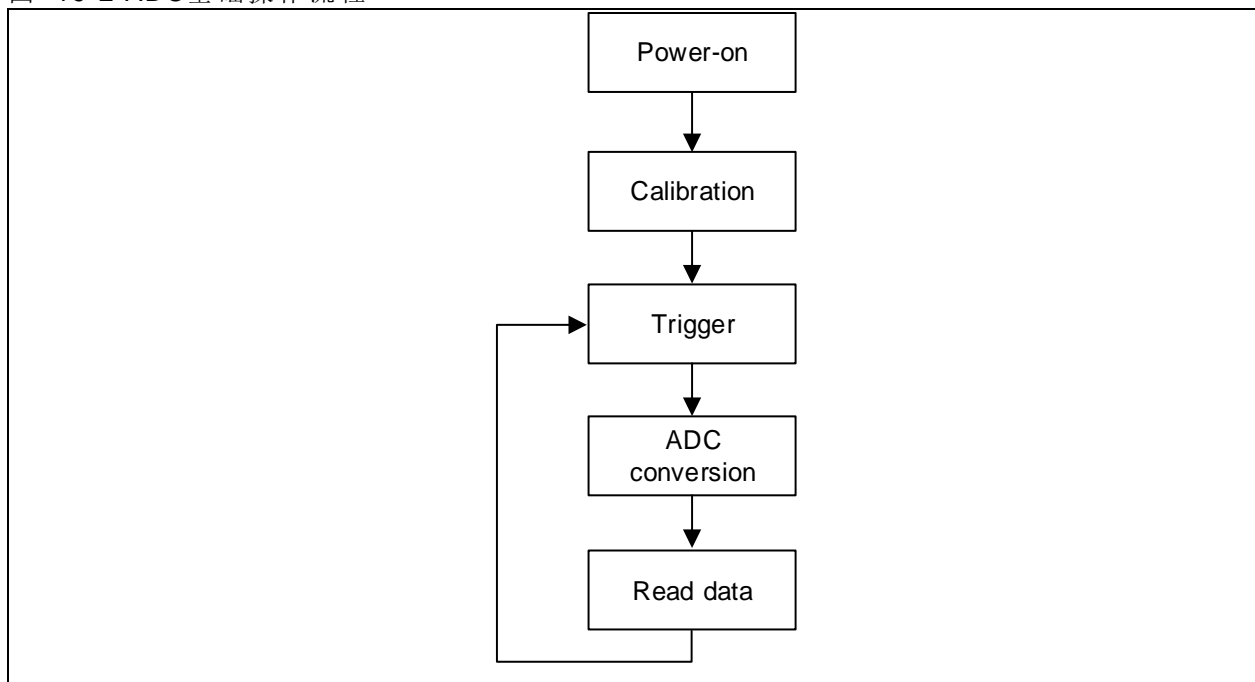
#### 19.4.1.2 内部参考电压

典型值 1.2V 的内部参考电压接到 ADC1\_IN17，必须先使能 ADC 控制寄存器 2（ADC\_CTRL2）的 ITSRVEN 位后才可对内部参考电压通道进行转换。此通道的转换数据可用于推算外部参考电压。

### 19.4.2 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 19-2 ADC基础操作流程



#### 19.4.2.1 上电与校准

##### 上电

用户须先使能 APB2 外设时钟使能寄存器（CRM\_APB2EN）的 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（时钟配置寄存器（CRM\_CFG）的 ADCDIV），将 ADCCLK 调整至需求的频率。ADCCLK 由 PCLK2 除频而来。

**注意：** ADCCLK 不可大于 28MHz。

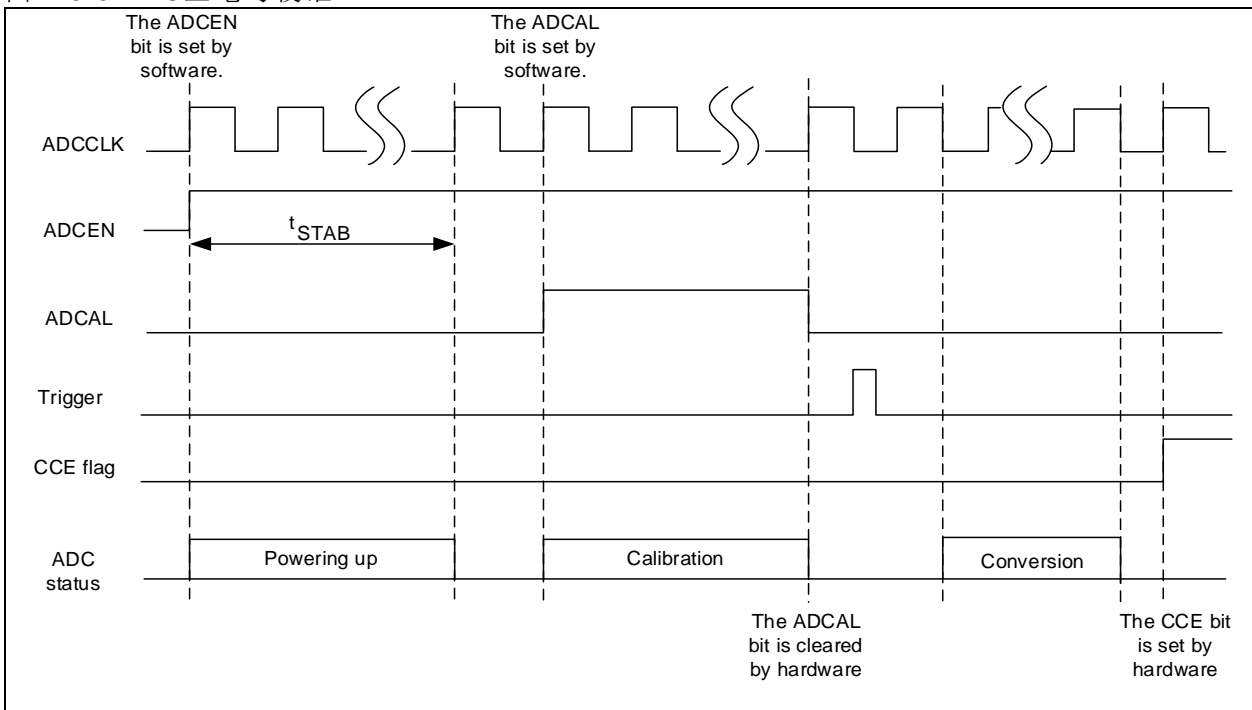
ADCCLK 频率调整完后，即可使能 ADC 控制寄存器 2（ADC\_CTRL2）的 ADCEN 位使 ADC 上电，等待 t<sub>STAB</sub> 后才可对 ADC 进行后续操作。清除 ADCEN 会使 ADC 的转换中止并复位，同时 ADC 被断电以达到省电的效果。

##### 校准

上电完成后可设置 ADC 控制寄存器 2 (ADC\_CTRL2) 的 ADCAL 使 ADC 进行校准, 校准完成后硬件清除 ADCAL 位, 软件即可触发以进行转换。

每次校准后, 校准值会被存放至 ADC 普通数据寄存器 (ADC\_ODT) 中, 这个校准值自动反馈回 ADC 内部, 以消除电容误差。该校准值的存放不会置位 CCE 标志, 不会产生中断或 DMA 请求。

图 19-3 ADC上电与校准



## 19.4.2.2 触发

ADC 触发分为普通通道触发与抢占通道触发, 普通通道触发引发普通通道转换, 抢占通道触发引发抢占通道转换。使能 ADC 控制寄存器 2 (ADC\_CTRL2) 的 OCTEN 或 PCTEN 后, ADC 才会检测触发来源的上升沿并响应转换。

触发来源可分为软件写寄存器触发 (ADC 控制寄存器 2 (ADC\_CTRL2) 的 OCSWTRG 与 PCSWTRG) 以及外部触发, 外部触发包含定时器触发与管脚触发, 由 ADC 控制寄存器 2 (ADC\_CTRL2) 的 OCTESEL 与 PCTESEL 选择触发来源, 如表 19-1 与

表 19-2 所示。

普通通道还有一种特殊的触发来源, 即重复使能 ADCEN 触发转换。此种情况下不需要使能 ADC 控制寄存器 2 (ADC\_CTRL2) 的 OCTEN 也可导致普通通道响应转换。

表 19-1 ADC1 与 ADC2 的触发来源

OCTESEL		触发来源	PCTESEL		触发来源
0000		TMR1_CH1 event	0000		TMR1_TRGOUT event
0001		TMR1_CH2 event	0001		TMR1_CH4 event
0010		TMR1_CH3 event	0010		TMR2_TRGOUT event
0011		TMR2_CH2 event	0011		TMR2_CH1 event
0100		TMR3_TRGOUT event	0100		TMR3_CH4 event
0101		TMR4_CH4 event	0101		TMR4_TRGOUT event
0110	ADCx_ETO_MU X=0	EXINT line11 external pin	0110	ADCx_ETP_MUX=0	EXINT line15 external pin
	ADCx_ETO_MU X=1	TMR8_TRGOUT event		ADCx_ETP_MUX=1	TMR8_CH4 event
0111		OCSWTRG bit	0111		PCSWTRG bit
1000		保留	1000		保留
1001		保留	1001		保留
1010		保留	1010		保留

1011	保留	1011	保留
1100	保留	1100	保留
1101	TMR1_TRGOUT event	1101	TMR1_CH1 event
1110	TMR8_CH1 event	1110	TMR8_CH1 event
1111	TMR8_CH2 event	1111	TMR8_TRGOUT event

表 19-2 ADC3的触发来源

OCTESEL	触发来源	PCTESEL	触发来源
0000	TMR3_CH1 event	0000	TMR1_TRGOUT event
0001	TMR2_CH3 event	0001	TMR1_CH4 event
0010	TMR1_CH3 event	0010	TMR4_CH3 event
0011	TMR8_CH1 event	0011	TMR8_CH2 event
0100	TMR8_TRGOUT event	0100	TMR8_CH4 event
0101	TMR5_CH1 event	0101	TMR5_TRGOUT event
0110	TMR5_CH3 event	0110	TMR5_CH4 event
0111	OCSWTRG bit	0111	PCSWTRG bit
1000	保留	1000	保留
1001	保留	1001	保留
1010	保留	1010	保留
1011	保留	1011	保留
1100	保留	1100	保留
1101	TMR1_TRGOUT event	1101	TMR1_CH1 event
1110	TMR1_CH1 event	1110	TMR1_CH2 event
1111	TMR8_CH3 event	1111	TMR8_TRGOUT event

### 19.4.2.3 采样与转换时序

用户可于 ADC 采样时间寄存器 1 (ADC\_SPT1) 与 ADC 采样时间寄存器 2 (ADC\_SPT2) 的 CSPTx 配置各个通道 (ADC\_INx) 的采样周期。一次转换所需的时间可利用以下公式推得：

$$\text{一次转换所需的时间(ADCCLK 的周期)} = \text{采样时间} + 12.5$$

示例：

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

CSPTx 选择 7.5 周期，一次转换需要  $7.5+12.5=20$  个 ADCCLK 周期。

### 19.4.3 转换顺序管理

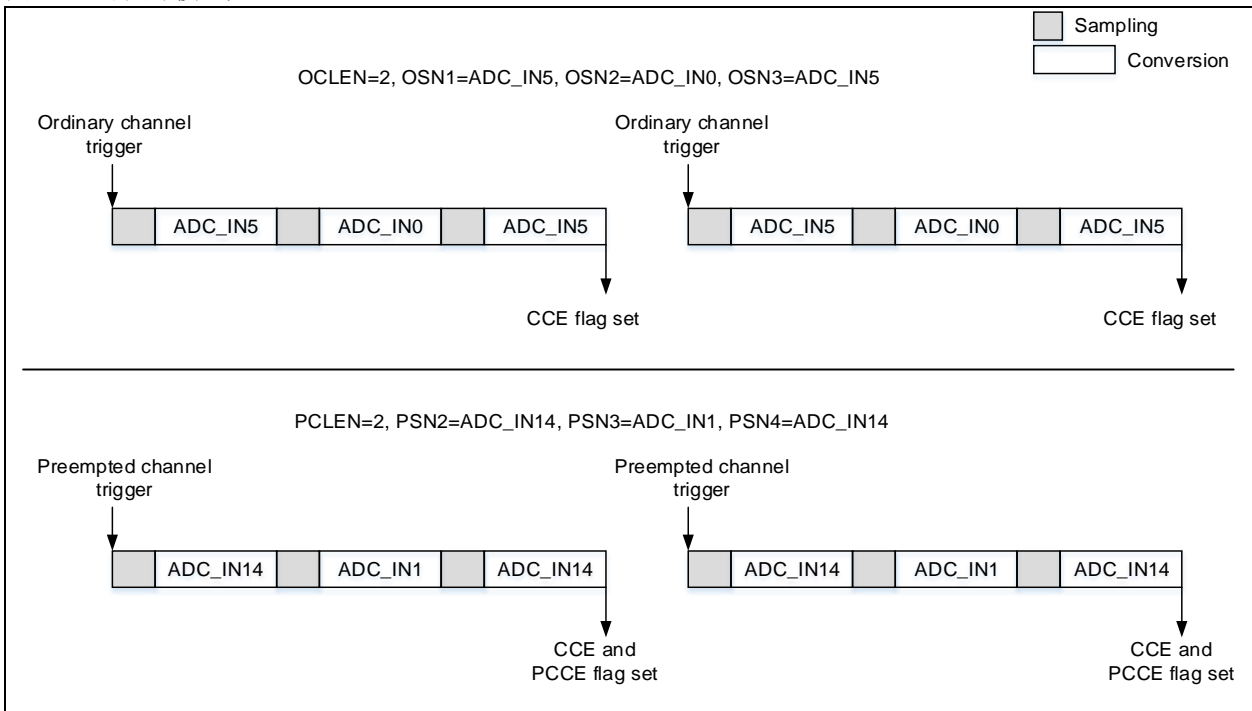
默认模式下，每次触发只会转换单个通道，即 OSN1（普通触发）或 PSN4（抢占触发）记录的通道。

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。

#### 19.4.3.1 序列模式

使能 ADC 控制寄存器 1 (ADC\_CTRL1) 的 SQEN，即开启序列模式，用户于 ADC\_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 PSNx 开始转换起， $x=4-PCLEN$ ，图 19-4 示范了序列模式的行为。

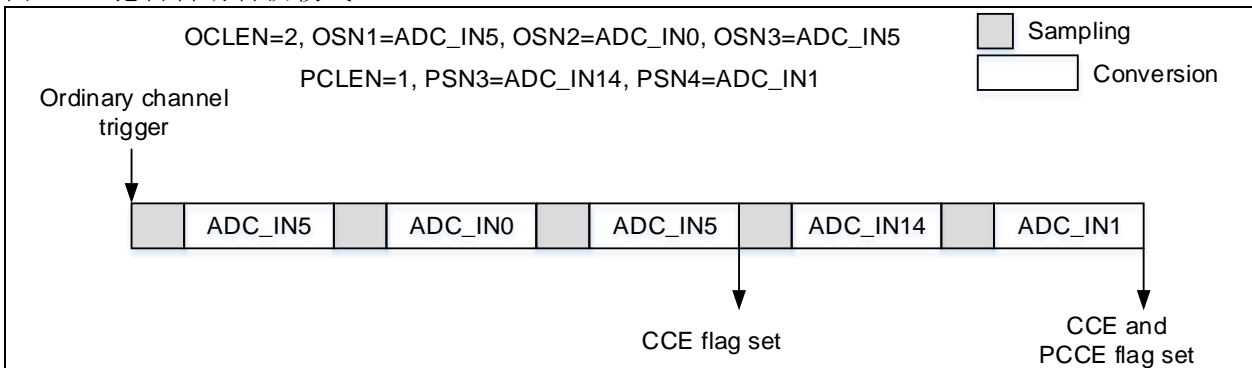
图 19-4 序列模式



### 19.4.3.2 抢占自动转换模式

使能 ADC 控制寄存器 1 (ADC\_CTRL1) 的 PCAUTOEN，即开启抢占自动转换模式，当普通通道转换完成后，抢占通道将自动接续着转换。可与序列模式共用，当普通通道序列完成后，即会自动开始抢占序列的转换。下图示范了与序列模式共用的抢占自动转换模式行为。

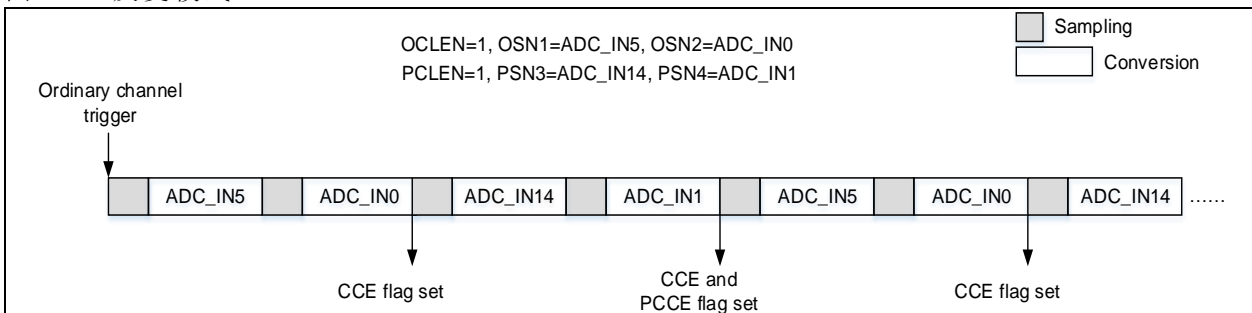
图 19-5 抢占自动转换模式



### 19.4.3.3 反复模式

使能 ADC 控制寄存器 2 (ADC\_CTRL2) 的 RPEN，即开启反复模式。当普通通道检测到触发后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 19-6 反复模式



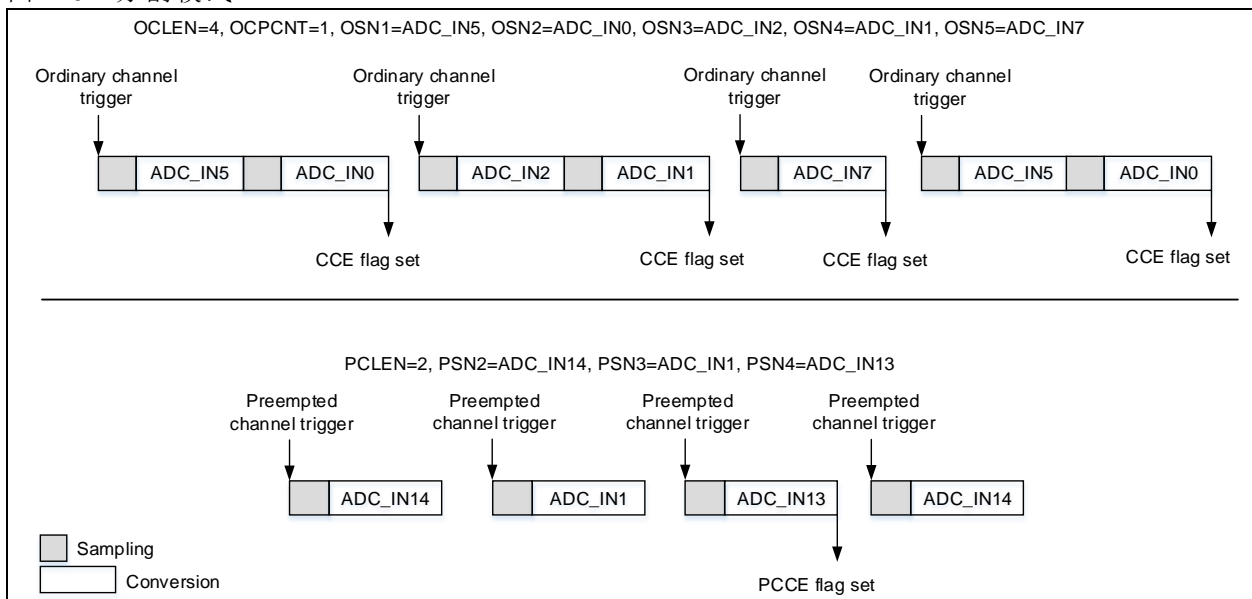
### 19.4.3.4 分割模式

使能 ADC 控制寄存器 1 (ADC\_CTRL1) 的 OCPEN，即开启普通通道的分割模式，此模式将 ADC 普通序列寄存器 1 (ADC\_OSQ1) 的 OCLEN 的序列长度分割成长度较小的子组别，子组别的通道数于 ADC 控制寄存器 1 (ADC\_CTRL1) 的 OCPCNT 配置，一次触发将转换子组别中的所有通道。每次触发会依序选择不同的子组别。

使能 ADC 控制寄存器 1 (ADC\_CTRL1) 的 PCPEN，即开启抢占通道的分割模式，此模式将 ADC 普通序列寄存器 1 (ADC\_OSQ1) 的 PCLEN 的序列长度分割成只有一个通道的子组别，一次触发将转换子组别中的通道。每次触发会依序选择不同的子组别。

分割模式与反复模式不可共用。下图分别示范了普通分割与抢占分割模式的行为。

图 19-7 分割模式



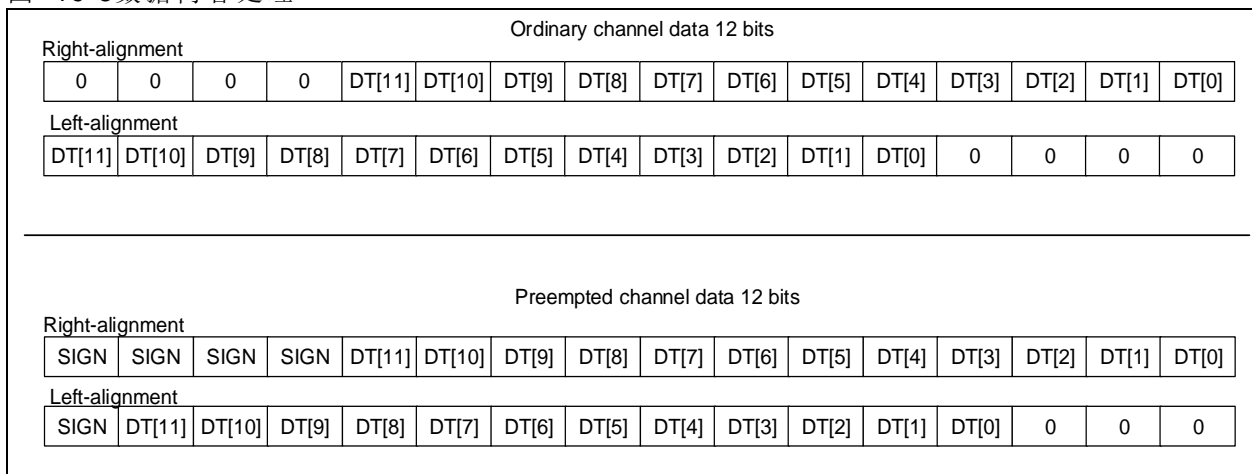
### 19.4.4 数据管理

普通通道转换完成后数据存储于 ADC 普通数据寄存器 (ADC\_ODT)，抢占通道转换完成后数据存储于 ADC 抢占数据寄存器 x (ADC\_PDTx)。

#### 19.4.4.1 数据内容处理

由 ADC 控制寄存器 2 (ADC\_CTRL2) 的 DTALIGN 选择转换数据靠右或是靠左对齐放置于数据寄存器，除此之外，抢占通道的数据还会减去 ADC 抢占通道数据偏移寄存器 x (ADC\_PCDTOx) 的偏移量，因此抢占通道数据有可能为负值，以 SIGN 作为符号。如下图所示。

图 19-8 数据内容处理



#### 19.4.4.2 数据获取

普通通道转换数据可藉由 CPU 或 DMA 读取 ADC 普通数据寄存器 (ADC\_ODT) 获得。抢占通道数据只

可藉由 CPU 读取 ADC 抢占数据寄存器 x (ADC\_PDTx) 获得。

使能 ADC 控制寄存器 2 (ADC\_CTRL2) 的 OCDMAEN 后, ADC 会在每次 ADC 普通数据寄存器 (ADC\_ODT) 更新时请求 DMA。

ADC1 与 ADC3 有各自的 DMA 通道, ADC2 可在主从模式下作为从机透过主机 ADC1 被 DMA 读取数据。

### 19.4.5 电压监测

使能 ADC 控制寄存器 1 (ADC\_CTRL1) 的 OCVMEN (普通通道) 或 PCVMEN (抢占通道) 即可通过对转换结果的判定来实现电压监测。当转换结果大于高边界 ADC 电压监测高边界寄存器 (ADC\_VMHB) 或是小于低边界 ADC 电压监测低边界寄存器 (ADC\_VMLB) 时, 电压监测超出标志 VMOR 会置起。透过 VMSGEN 选择对单一特定通道或是所有通道监测。对单一通道监测的话, 由 VMCSEL 配置通道。电压监测一律以转换的原始数据与 12 位边界寄存器做比较, 无视 PCDTOx 与 DTALIGN 位的设定。

### 19.4.6 状态标志与中断

每个 ADC 拥有自己的 ADC 状态寄存器 (ADCx\_STS): 普通通道转换开始标志 (OCCS)、抢占通道转换开始标志 (PCCS)、抢占通道组转换结束标志 (PCCE)、通道转换结束标志 (CCE) 及电压监测超出标志 (VMOR)。

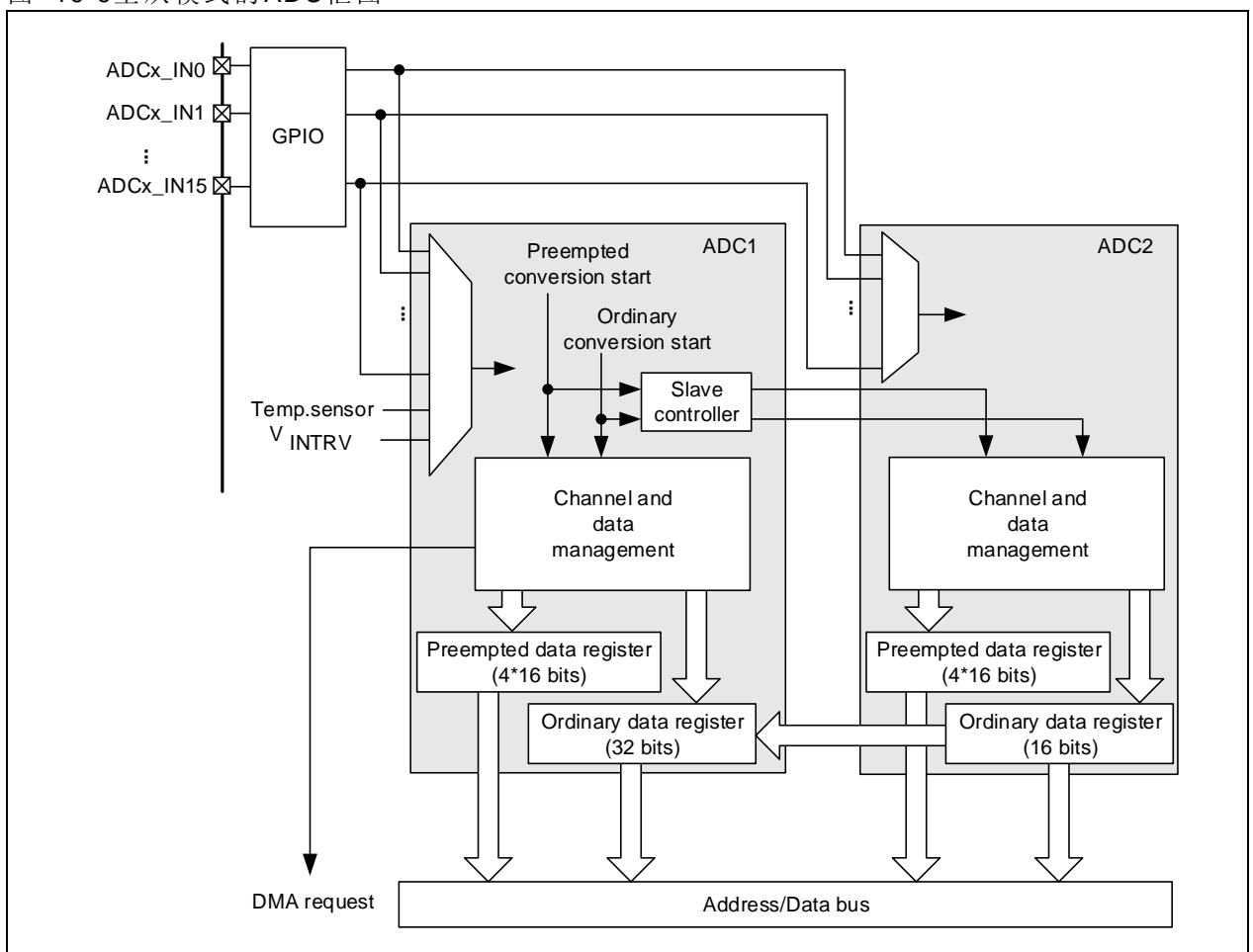
其中抢占通道组转换结束标志、通道转换结束标志及电压监测超出标志拥有对应中断使能位, 只要将中断使能, 标志置起时便会对 CPU 发出中断。ADC1 与 ADC2 共用一个中断向量, ADC3 独立一个中断向量。

## 19.5 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, 并且将主机的 ADC 普通数据寄存器 (ADC\_ODT) 作为获取主从 ADC 普通通道数据的单一接口。

主从模式以 ADC1 作为主机, ADC2 作为从机。

图 19-9 主从模式的 ADC 框图





### 19.5.1 数据管理

主从模式时，普通通道数据会共同存储于 ADC1 的 ADC 普通数据寄存器（ADC\_ODT）中，只要 ADC1\_CTRL2 的 OCDMAEN 位不为 0，就会在每次数据备齐时使用 ADC1 的 DMA 通道请求 DMA。

### 19.5.2 同时模式

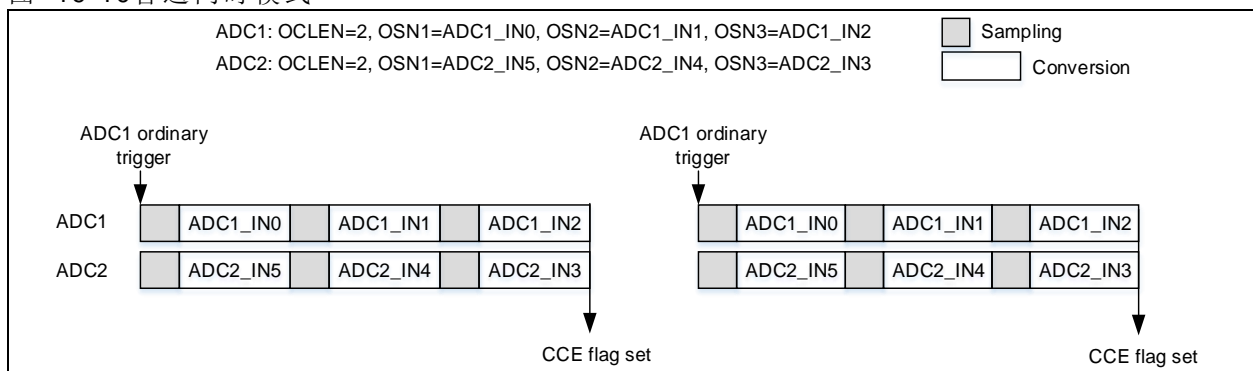
#### 普通同时模式

配置 ADC 控制寄存器 1（ADC\_CTRL1）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。

**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 19-10 普通同时模式

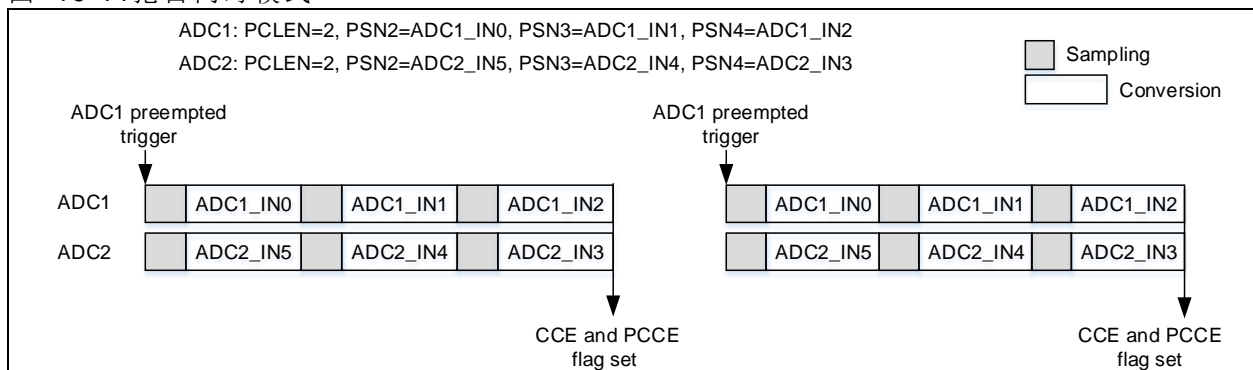


#### 抢占同时模式

配置 ADC 控制寄存器 1（ADC\_CTRL1）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。图 19-11 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 19-11 抢占同时模式



#### 混合的普通同时+抢占同时模式

配置 ADC 控制寄存器 1（ADC\_CTRL1）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。

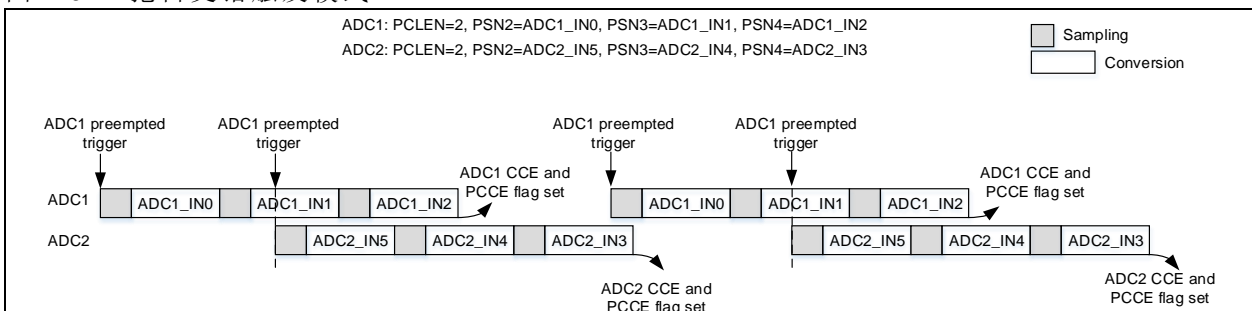
### 19.5.3 抢占交错触发模式

#### 抢占交错触发模式

配置 ADC 控制寄存器 1（ADC\_CTRL1）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 19-12 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 控制寄存器 1 (ADC\_CTRL1) 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。

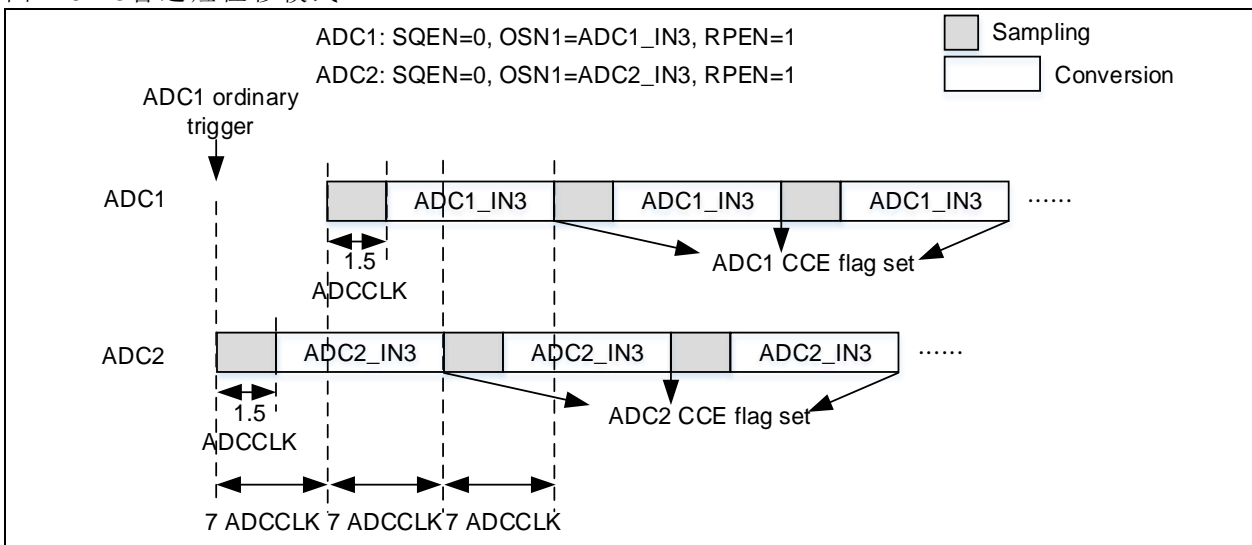
当普通通道转换被抢占通道触发打断, 所有的 ADC 停下普通通道转换, 其中一个 ADC 进入抢占通道转换, 此时主机将无视抢占通道触发, 直到普通通道恢复转换后才会再接受抢占通道触发。

**19.5.4 普通位移模式****普通短位移模式**

配置 ADC 控制寄存器 1 (ADC\_CTRL1) 的 MSSEL 至普通短位移模式后, 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 7 个 ADCCLK。在这个模式下, 采样时间只能选择 1.5 个 ADCCLK 周期。如下图所示。

注意: 此模式下禁止抢占通道触发。

图 19-13 普通短位移模式

**混合的抢占同时+普通短位移模式**

配置 ADC 控制寄存器 1 (ADC\_CTRL1) 的 MSSEL 至混合的抢占同时+普通短位移模式后, 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 7 个 ADCCLK, 也可触发主机抢占通道使主机与从机同时转换抢占通道。

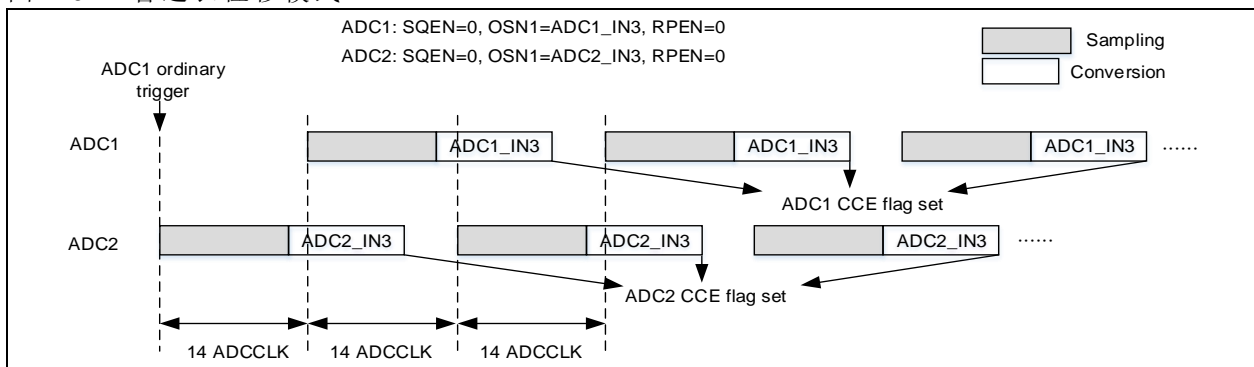
当普通通道转换被抢占通道触发打断, 等抢占通道转换完成后, 一律从 ADC2 开始恢复普通通道转换。

**普通长位移模式**

配置 ADC 控制寄存器 1 (ADC\_CTRL1) 的 MSSEL 至普通长位移模式后, 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 14 个 ADCCLK。在这个模式下, 采样时间只能选择小于 14 个 ADCCLK 周期。如下图所示。

注意: 此模式下禁止抢占通道触发。此模式下禁止与反复模式共用。

图 19-14普通长位移模式



### 混合的抢占同时+普通长位移模式

配置 ADC 控制寄存器 1 (ADC\_CTRL1) 的 MSSEL 至混合的抢占同时+普通长位移模式后, 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 14 个 ADCCLK, 也可触发主机抢占通道使主机与从机同时转换抢占通道。

当普通通道转换被抢占通道触发打断, 等抢占通道转换完成后, 一律从 ADC2 开始恢复普通通道转换。

## 19.6 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 19-3 ADC 寄存器映像和复位值

寄存器简称	基址偏移量	复位值
ADC_STS	0x000	0x0000 0000
ADC_CTRL1	0x004	0x0000 0000
ADC_CTRL2	0x008	0x0000 0000
ADC_SPT1	0x00C	0x0000 0000
ADC_SPT2	0x010	0x0000 0000
ADC_PCDTO1	0x014	0x0000 0000
ADC_PCDTO2	0x018	0x0000 0000
ADC_PCDTO3	0x01C	0x0000 0000
ADC_PCDTO4	0x020	0x0000 0000
ADC_VMHB	0x024	0x0000 0FFF
ADC_VMLB	0x028	0x0000 0000
ADC_OSQ1	0x02C	0x0000 0000
ADC_OSQ2	0x030	0x0000 0000
ADC_OSQ3	0x034	0x0000 0000
ADC_PSQ	0x038	0x0000 0000
ADC_PDT1	0x03C	0x0000 0000
ADC_PDT2	0x040	0x0000 0000
ADC_PDT3	0x044	0x0000 0000
ADC_PDT4	0x048	0x0000 0000
ADC_ODT	0x04C	0x0000 0000

### 19.6.1 ADC状态寄存器（ADC\_STS）

域	简称	复位值	类型	功能
位 31: 5	保留	0x0000000	resd	请保持默认值。
位 4	OCCS	0x0	rw0c	普通通道转换开始标志（Ordinary channel conversion start flag） 该位被硬件置起，由软件将其清零（对自身写零）。 0：未开始； 1：已开始。
位 3	PCCS	0x0	rw0c	抢占通道转换开始标志（Preempted channel conversion start flag） 该位被硬件置起，由软件将其清零（对自身写零）。 0：未开始； 1：已开始。
位 2	PCCE	0x0	rw0c	抢占通道组转换结束标志（Preempted channels conversion end flag） 该位被硬件置起，由软件将其清零（对自身写零）。 0：未结束； 1：已结束。
位 1	CCE	0x0	rw0c	通道转换结束标志（Channels conversion end flag） 该位被硬件置起，由软件将其清零（对自身写零），或由读取 ADC 普通数据寄存器（ADC_ODT）清零。 0：未结束； 1：已结束。 注：普通或抢占通道组转换结束均会置位此标志。
位 0	VMOR	0x0	rw0c	电压监测超出范围标志（Voltage monitoring out of range flag） 该位被硬件置起，由软件将其清零（对自身写零）。 0：无超出； 1：有超出。

### 19.6.2 ADC控制寄存器1（ADC\_CTRL1）

域	简称	复位值	类型	功能
位 31: 24	保留	0x00	resd	请保持默认值。
位 23	OCVMEN	0x0	rw	普通通道的电压监测使能（Voltage monitoring enable on ordinary channels） 0：关闭； 1：开启。
位 22	PCVMEN	0x0	rw	抢占通道的电压监测使能（Voltage monitoring enable on preempted channels） 0：关闭； 1：开启。
位 21: 20	保留	0x0	resd	请保持默认值。

位 19: 16	MSSEL	0x0	rw	<p>主从模式选择（Master slave mode select）</p> <p>0000：非主从模式；</p> <p>0001：混合的普通同时+抢占同时模式；</p> <p>0010：混合的普通同时+抢占交错触发模式；</p> <p>0011：混合的抢占同时+普通短位移模式；</p> <p>0100：混合的抢占同时+普通长位移模式；</p> <p>0101：抢占同时模式；</p> <p>0110：普通同时模式；</p> <p>0111：普通短位移模式；</p> <p>1000：普通长位移模式；</p> <p>1001：抢占交错触发模式；</p> <p>1010~1111：未用，禁止配置。</p> <p>注：</p> <p>在 ADC2 和 ADC3 中这些位为保留位，需保持默认值。</p> <p>在主从模式中，修改配置会导致主从时序丢失同步。建议在修改前先关闭主从模式。</p>
位 15: 13	OCPCNT	0x0	rw	<p>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）</p> <p>000：1 个通道；</p> <p>001：2 个通道；</p> <p>.....</p> <p>111：8 个通道。</p> <p>注： 抢占组在分割模式下每次触发固定只转换一个通道。</p>
位 12	PCPEN	0x0	rw	<p>抢占通道上的分割模式使能（Partitioned mode enable on preempted channels）</p> <p>0：关闭；</p> <p>1：开启。</p>
位 11	OCPEN	0x0	rw	<p>普通通道上的分割模式使能（Partitioned mode enable on ordinary channels）</p> <p>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</p> <p>0：关闭；</p> <p>1：开启。</p>
位 10	PCAUTOEN	0x0	rw	<p>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）</p> <p>0：关闭；</p> <p>1：开启。</p>
位 9	VMSGEN	0x0	rw	<p>单个通道的电压监测使能（Voltage monitoring enable on a single channel）</p> <p>0：关闭（电压监测所有通道）；</p> <p>1：开启（电压监测单一通道）。</p>
位 8	SQEN	0x0	rw	<p>序列模式使能（Sequence mode enable）</p> <p>0：关闭（转换选择的单一通道）；</p> <p>1：开启（转换设定的多个通道）。</p> <p>注： 如果开启多通道模式，且开启了 CCEIEN 或 PCCEIEN 位，则只在最后一个通道转换完毕后才会产生 CCE 或 PCCE 中断。</p>
位 7	PCCEIEN	0x0	rw	<p>抢占通道组转换结束中断使能（conversion end interrupt enable for Preempted channels）</p> <p>0：关闭；</p> <p>1：开启。</p>

位 6	VMORIEN	0x0	rw	电压监测超出范围中断使能（Voltage monitoring out of range interrupt enable） 0：关闭； 1：开启。
位 5	CCEIEN	0x0	rw	通道转换结束中断使能（Channel conversion end interrupt enable） 0：关闭； 1：开启。
位 4: 0	VMCSEL	0x00	rw	电压监测通道选择（Voltage monitoring channel select） 仅在 VMSGEN 开启时有效。 00000：ADC_IN0 通道； 00001：ADC_IN1 通道； ..... 01111：ADC_IN15 通道； 10000：ADC_IN16 通道； 10001：ADC_IN17 通道。 10010~11111：未用，禁止配置。

## 19.6.3 ADC控制寄存器2（ADC\_CTRL2）

域	简称	复位值	类型	功能
位 30: 26	保留	0x00	resd	请保持默认值。
位 23	ITSRVEN	0x0	rw	内部温度传感器及 VINTRV 使能（Internal temperature sensor and VINTRV enable） 0：关闭； 1：开启。 注：在 ADC2 和 ADC3 中此位为保留位，需保持默认值。
位 22	OCSWTRG	0x0	rw	软件触发普通通道转换（Conversion trigger by software of ordinary channels） 0：不触发； 1：触发转换（可由软件清除，或在转换开始后由硬件自动清除）。
位 21	PCSWTRG	0x0	rw	软件触发抢占通道转换（Conversion trigger by software of preempted channels） 0：不触发； 1：触发转换（可由软件清除，或在转换开始后由硬件自动清除）。
位 20	OCTEN	0x0	rw	普通通道组转换的触发模式使能（Trigger mode enable for ordinary channels conversion） 0：关闭； 1：开启。

				普通通道组转换的触发事件选择（trigger event select for ordinary channels conversion）		
				ADC1 和 ADC2 的触发配置如下		
				0000: 定时器 1 的 CH1 事件；		
				0001: 定时器 1 的 CH2 事件；		
				0010: 定时器 1 的 CH3 事件；		
				0011: 定时器 2 的 CH2 事件；		
				0100: 定时器 3 的 TRGOUT 事件；		
				0101: 定时器 4 的 CH4 事件；		
				0110: EXINT 线 11/ TMR8_TRGOUT 事件；		
				0111: OCSWTRG；		
				1000~1100: 未用，禁止配置。；		
				1101: 定时器 1 的 TRGOUT 事件；		
				1110: 定时器 8 的 CH1 事件；		
				1111: 定时器 8 的 CH2 事件。		
位 25	OCTESEL	0x0	rw			
位 19: 17						
ADC3 的触发配置如下						
0000: 定时器 3 的 CH1 事件；						
0001: 定时器 2 的 CH3 事件；						
0010: 定时器 1 的 CH3 事件；						
0011: 定时器 8 的 CH1 事件；						
0100: 定时器 8 的 TRGOUT 事件；						
0101: 定时器 5 的 CH1 事件；						
0110: 定时器 5 的 CH3 事件；						
0111: OCSWTRG；						
1000~1100: 未用，禁止配置。；						
1101: 定时器 1 的 TRGOUT 事件；						
1110: 定时器 1 的 CH1 事件；						
1111: 定时器 8 的 CH3 事件。						
位 16	保留	0x0	resd	请保持默认值。		
位 15	PCTEN	0x0	rw	抢占通道组转换的触发模式使能（Trigger mode enable for preempted channels conversion）		
				0: 关闭；		
				1: 开启。		

位 24 位 14: 12	PCTESEL	0x0	rw	<p>抢占通道组转换的触发事件选择（trigger event select for preempted channels conversion）</p> <p>ADC1 和 ADC2 的触发配置如下</p> <p>0000: 定时器 1 的 TRGOUT 事件;</p> <p>0001: 定时器 1 的 CH4 事件;</p> <p>0010: 定时器 2 的 TRGOUT 事件;</p> <p>0011: 定时器 2 的 CH1 事件;</p> <p>0100: 定时器 3 的 CH4 事件;</p> <p>0101: 定时器 4 的 TRGOUT 事件;</p> <p>0110: EXINT 线 15/TMR8_CH4 事件;</p> <p>0111: PCSWTRG;</p> <p>1000~1100: 未用, 禁止配置;</p> <p>1101: 定时器 1 的 CH1 事件;</p> <p>1110: 定时器 8 的 CH1 事件;</p> <p>1111: 定时器 8 的 TRGOUT 事件。</p> <p>ADC3 的触发配置如下</p> <p>0000: 定时器 1 的 TRGOUT 事件;</p> <p>0001: 定时器 1 的 CH4 事件;</p> <p>0010: 定时器 4 的 CH3 事件;</p> <p>0011: 定时器 8 的 CH2 事件;</p> <p>0100: 定时器 8 的 CH4 事件;</p> <p>0101: 定时器 5 的 TRGOUT 事件;</p> <p>0110: 定时器 5 的 CH4 事件;</p> <p>0111: PCSWTRG;</p> <p>1000~1100: 未用, 禁止配置;。</p> <p>1101: 定时器 1 的 CH1 事件;</p> <p>1110: 定时器 1 的 CH2 事件;</p> <p>1111: 定时器 8 的 TRGOUT 事件。</p>
位 11	DTALIGN	0x0	rw	<p>数据对齐方式（Data alignment）</p> <p>0: 右对齐;</p> <p>1: 左对齐。</p>
位 10: 9	保留	0x0	resd	请保持默认值。
位 8	OCDMAEN	0x0	rw	<p>普通通道转换数据的 DMA 传输使能（DMA transfer enable of ordinary channels）</p> <p>0: 关闭;</p> <p>1: 开启。</p> <p>注: ADC2 无自己的 DMA 功能, 其不可独立产生 DMA 请求。</p>
位 7: 4	保留	0x0	resd	请保持默认值。
位 3	ADCALINIT	0x0	rw	<p>A/D 初始化校准（initialize A/D calibration）</p> <p>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。</p> <p>0: 校准寄存器无初始化执行或初始化结束;</p> <p>1: 校准寄存器初始化或初始化进行中。</p>
位 2	ADCAL	0x0	rw	<p>A/D 校准（A/D Calibration）</p> <p>0: 无校准执行或校准结束;</p> <p>1: 开始校准或校准进行中。</p>



位 1	RPEN	0x0	rw	<p>反复模式使能（Repeat mode enable）</p> <p>0：关闭</p> <p>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；</p> <p>1：开启</p> <p>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。</p>
位 0	ADCEN	0x0	rw	<p>A/D 转换器使能（A/D converter enable）</p> <p>0：关闭（ADC 进入断电模式）；</p> <p>1：开启。</p> <p>注：</p> <p>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。</p> <p>当该位为开启状态时，再写入开启命令将启动普通通道组的转换。</p> <p>应用程序需注意，在转换器上电至转换开始有一个延迟 <math>t_{STAB}</math>。</p>

## 19.6.4 ADC采样时间寄存器1（ADC\_SPT1）

域	简称	复位值	类型	功能
位 31: 24	保留	0x00	resd	请保持默认值。
位 23: 21	CSPT17	0x0	rw	<p>选择 ADC_IN17 通道的采样时间（Selection sample time of channel ADC_IN17）</p> <p>000: 1.5 周期；</p> <p>001: 7.5 周期；</p> <p>010: 13.5 周期；</p> <p>011: 28.5 周期；</p> <p>100: 41.5 周期；</p> <p>101: 55.5 周期；</p> <p>110: 71.5 周期；</p> <p>111: 239.5 周期。</p>
位 20: 18	CSPT16	0x0	rw	<p>选择 ADC_IN16 通道的采样时间（Selection sample time of channel ADC_IN16）</p> <p>000: 1.5 周期；</p> <p>001: 7.5 周期；</p> <p>010: 13.5 周期；</p> <p>011: 28.5 周期；</p> <p>100: 41.5 周期；</p> <p>101: 55.5 周期；</p> <p>110: 71.5 周期；</p> <p>111: 239.5 周期。</p>
位 17: 15	CSPT15	0x0	rw	<p>选择 ADC_IN15 通道的采样时间（Selection sample time of channel ADC_IN15）</p> <p>000: 1.5 周期；</p> <p>001: 7.5 周期；</p> <p>010: 13.5 周期；</p> <p>011: 28.5 周期；</p> <p>100: 41.5 周期；</p> <p>101: 55.5 周期；</p> <p>110: 71.5 周期；</p> <p>111: 239.5 周期。</p>

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。

## 19.6.5 ADC采样时间寄存器2（ADC\_SPT2）

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。

## 19.6.6 ADC抢占通道数据偏移寄存器x（ADC\_PCDTOx）（x=1..4）

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

## 19.6.7 ADC电压监测高边界寄存器（ADC\_VMHB）

域	简称	复位值	类型	功能
位 31: 12	保留	0x00000	resd	请保持默认值。
位 11: 0	VMHB	0xFFFF	rw	电压监测高边界设定（Voltage monitoring high boundary）

## 19.6.8 ADC电压监测低边界寄存器（ADC\_VMLB）

域	简称	复位值	类型	功能
位 31: 12	保留	0x00000	resd	请保持默认值。
位 11: 0	VMLB	0x000	rw	电压监测低边界设定（Voltage monitoring low boundary）

## 19.6.9 ADC普通序列寄存器1（ADC\_OSQ1）

域	简称	复位值	类型	功能
位 31: 24	保留	0x00	resd	请保持默认值。
位 23: 20	OCLEN	0x0	rw	普通转换序列长度（Ordinary conversion sequence length） 0000: 1 个转换； 0001: 2 个转换； ..... 1111: 16 个转换。
位 19: 15	OSN16	0x00	rw	普通序列中第 16 个转换通道的编号（number of 16th conversion in ordinary sequence）
位 14: 10	OSN15	0x00	rw	普通序列中第 15 个转换通道的编号（number of 15th conversion in ordinary sequence）
位 9: 5	OSN14	0x00	rw	普通序列中第 14 个转换通道的编号（number of 14th conversion in ordinary sequence）
位 4: 0	OSN13	0x00	rw	普通序列中第 13 个转换通道的编号（number of 13th conversion in ordinary sequence） 注：编号可设定 0~17，示例：设定为 3 就代表第 13 个转换的是 ADC_IN3 通道。

## 19.6.10 ADC普通序列寄存器2 (ADC\_OSQ2)

域	简称	复位值	类型	功能
位 31: 30	保留	0x0	resd	请保持默认值。
位 29: 25	OSN12	0x00	rw	普通序列中第 12 个转换通道的编号 (number of 12th conversion in ordinary sequence)
位 24: 20	OSN11	0x00	rw	普通序列中第 11 个转换通道的编号 (number of 11th conversion in ordinary sequence)
位 19: 15	OSN10	0x00	rw	普通序列中第 10 个转换通道的编号 (number of 10th conversion in ordinary sequence)
位 14: 10	OSN9	0x00	rw	普通序列中第 9 个转换通道的编号 (number of 9th conversion in ordinary sequence)
位 9: 5	OSN8	0x00	rw	普通序列中第 8 个转换通道的编号 (number of 8th conversion in ordinary sequence)
位 4: 0	OSN7	0x00	rw	普通序列中第 7 个转换通道的编号 (number of 7th conversion in ordinary sequence) 注: 编号可设定 0~17, 示例: 设定为 8 就代表第 7 个转换的是 ADC_IN8 通道。

## 19.6.11 ADC普通序列寄存器3 (ADC\_OSQ3)

域	简称	复位值	类型	功能
位 31: 30	保留	0x0	resd	请保持默认值。
位 29: 25	OSN6	0x00	rw	普通序列中第 6 个转换通道的编号 (number of 6th conversion in ordinary sequence)
位 24: 20	OSN5	0x00	rw	普通序列中第 5 个转换通道的编号 (number of 5th conversion in ordinary sequence)
位 19: 15	OSN4	0x00	rw	普通序列中第 4 个转换通道的编号 (number of 4th conversion in ordinary sequence)
位 14: 10	OSN3	0x00	rw	普通序列中第 3 个转换通道的编号 (number of 3rd conversion in ordinary sequence)
位 9: 5	OSN2	0x00	rw	普通序列中第 2 个转换通道的编号 (number of 2nd conversion in ordinary sequence)
位 4: 0	OSN1	0x00	rw	普通序列中第 1 个转换通道的编号 (number of 1st conversion in ordinary sequence) 注: 编号可设定 0~17, 示例: 设定为 17 就代表第 1 个转换的是 ADC_IN17 通道。

## 19.6.12 ADC抢占序列寄存器 (ADC\_PSQ)

域	简称	复位值	类型	功能
位 31: 30	保留	0x0	resd	请保持默认值。
位 21: 20	PCLEN	0x0	rw	抢占转换序列长度 (Preempted conversion sequence length) 00: 1 个转换; 01: 2 个转换; 10: 3 个转换; 11: 4 个转换。
位 19: 15	PSN4	0x00	rw	抢占序列中第 4 个转换通道的编号 (number of 4th conversion in Preempted sequence)
位 14: 10	PSN3	0x00	rw	抢占序列中第 3 个转换通道的编号 (number of 3rd conversion in Preempted sequence)
位 9: 5	PSN2	0x00	rw	抢占序列中第 2 个转换通道的编号 (number of 2nd conversion in Preempted sequence)
位 4: 0	PSN1	0x00	rw	抢占序列中第 1 个转换通道的编号 (number of 1st conversion in Preempted sequence) 注: 编号可设定 0~17, 比如设定为 3 时其代表的就是 ADC_IN3 通道。 若 PCLEN 小于 4, 则转换的序列顺序是从 (4-PCLEN) 开始。例如: ADC 抢占序列寄存器 (ADC_PSQ) [21: 0] = 10 00110 00101 00100 00011, 意味着扫描转换将按下列通道顺序执行: 4、5、6, 而不是 3、4、5。

## 19.6.13 ADC抢占数据寄存器x (ADC\_PDTx) (x= 1..4)

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)

## 19.6.14 ADC普通数据寄存器 (ADC\_ODT)

域	简称	复位值	类型	功能
位 31: 16	ADC2ODT	0x0000	ro	ADC2 普通通道的转换数据 (ADC2 conversion data of ordinary channel) 注: 在 ADC2 和 ADC3 中这些位为保留位。 在 ADC1 中, 只有配置主从组合模式时这些位才有意义, 且这些位包含的是 ADC2 普通通道的转换数据。
位 15: 0	ODT	0x0000	ro	普通通道的转换数据 (Conversion data of ordinary channel)

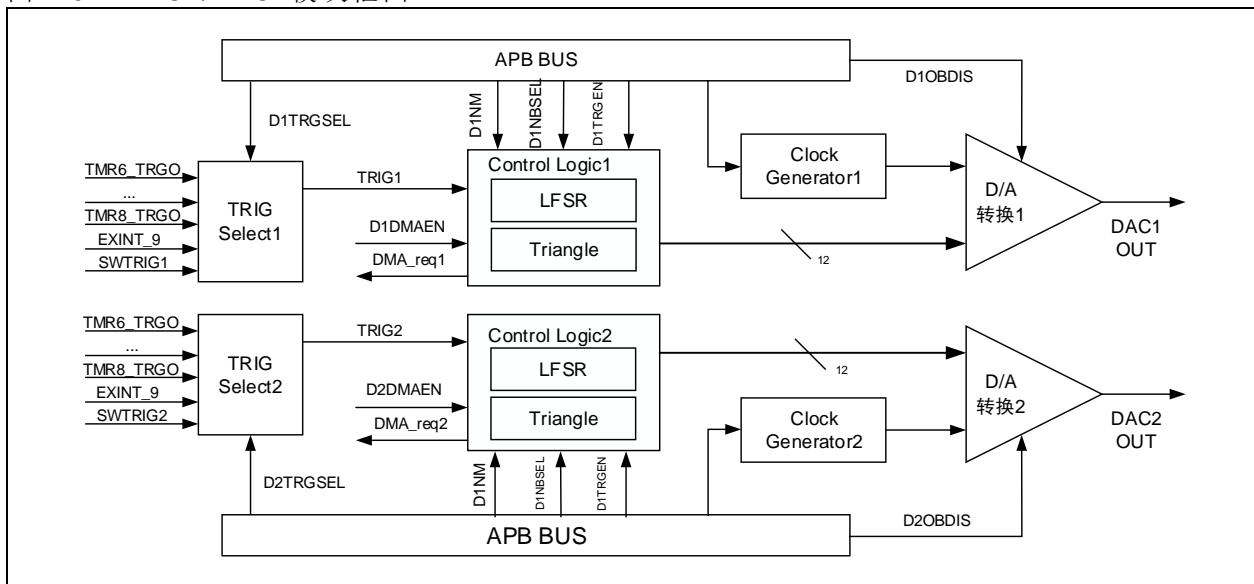


## 20 数字/模拟转换 (DAC)

### 20.1 简介

数模转换器 (DAC) 采用 12 位数字输入, 产生 0 至参考电压之间的模拟输出。数字部分可以配置为 8 位或者 12 位模式, 支持单/双 DAC 的左对齐或者右对齐, 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, 输入参考电压  $V_{REF+}$  可以使转换操作更加精确。

图 20-1 DAC1/DAC2 模块框图



### 20.2 主要特性

- 单/双 DAC 8 位或者 12 位数字输入
- 数据支持左对齐或者右对齐模式
- 支持噪声波/三角波产生
- 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
- 支持输入参考电压  $V_{REF+}$

### 20.3 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) ENx 位控制开启, 数字部分则不受该位控制。另外 DAC 集成了 2 个输出增益, 可以用来减少输出阻抗, 无需外部运放即可直接驱动外部负载。每个 DAC1/DAC2 输出增益可以通过设置 DAC 控制寄存器 (DAC\_CTRL) 的 DxOBDIS 位来使能或关闭。

- 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。

- 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  $V_{REF+}$ 。模拟 DAC 模块采

用 VDDA 供电，输入正模拟参考电压大小介于 2.0V 与 VDDA 之间，PA4 或者 PA5 作为模拟输出时，为避免寄生干扰和额外的功耗，需设置为模拟输入。

DAC 输出 =  $V_{REF+} \times (DxODT[11: 0] / 4095)$

## 20.4 功能描述

### 20.4.1 触发事件

如果 DAC 控制寄存器 (DAC\_CTRL) 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 20-1 触发源选择

触发源	DxTRGSEL [2: 0]	说明
TMR6_TRGOUT	000	片上信号
TMR8_TRGOUT	001	
TMR7_TRGOUT	010	
TMR5_TRGOUT	011	
TMR2_TRGOUT	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 检测到有效的触发事件，存放在 HDRx 中的数据就会被传送到 DAC1/DAC2 数据输出寄存器 (DAC\_DxODT) 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 DAC1/DAC2 数据输出寄存器 (DAC\_DxODT)，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 DAC1/DAC2 数据输出寄存器 (DAC\_DxODT) 中，无需等待触发事件。

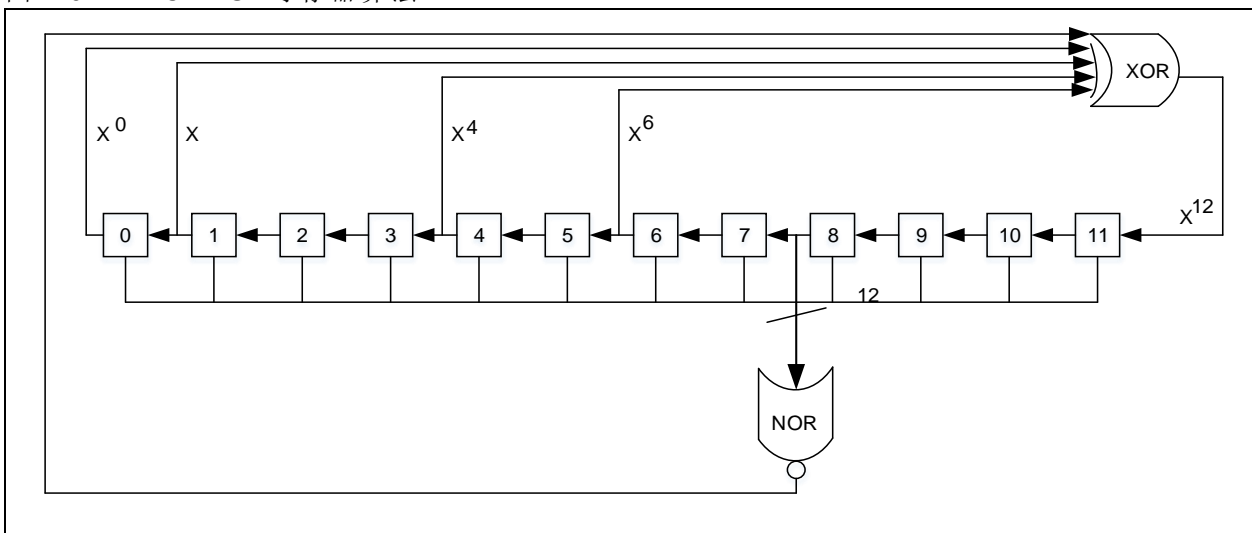
### 20.4.2 噪声/三角波生成

有噪声波和三角波两种波形可以叠加到 DAC 输出：分别利用线性反馈移位寄存器 (Linear Feedback Shift Register LFSR) 产生幅度变化的伪噪声和通过三角波发生器 (triangle) 产生三角波。当设置 DxNM[1: 0] 位为 '01' 使能 LFSR，输出幅度变化的伪噪声。当设置 DxNM[1: 0] 位为 '1x' 使能三角波发生器，输出三角波。

#### LFSR 原理

寄存器 LFSR 的预装入值为 0xAAA，按照特定算法，在每次触发事件之后更新该寄存器的值。

图 20-2 DAC LFSR 寄存器算法



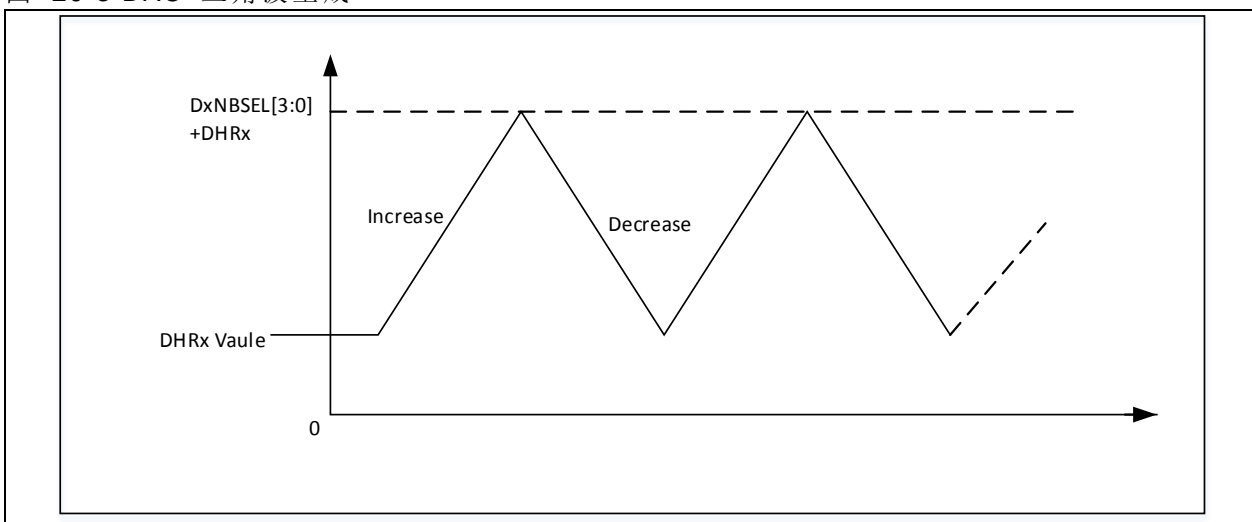
设置 DAC 控制寄存器（DAC\_CTRL）的 DxBSEL [3: 0]位可以屏蔽部分或者全部 LFSR 的数据，这样得到的 LFSR 值与 DHRx 的数值相加，去掉溢出位之后即被写入 DAC1/DAC2 数据输出寄存器（DAC\_DxODT）。将 DxBSEL [1: 0]位置‘00’可以关掉 LFSR 功能，同时复位 LFSR 波形的生成算法。

#### 三角波原理

当设置 DxBSEL [1: 0]位为‘1x’，则选择 DAC 的三角波生成功能。三角波的幅度由 DAC 控制寄存器（DAC\_CTRL）的 DxBSEL [3: 0]位设置，内部的三角波计数器每检测到一次触发事件累加 1，当达到 DxBSEL [3: 0]位定义的最大幅度时，则计数器开始递减，达到 0 后再开始累加，周而复始。

同时，计数器的值与 DHRx 寄存器的数值相加并丢弃溢出位后写入 DAC1/DAC2 数据输出寄存器（DAC\_DxODT）。将 DxBSEL [1: 0]设置‘00’，可以关掉三角波生成，同时复位三角波计数器。

图 20-3 DAC 三角波生成



### 20.4.3 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：**采用 8 位数据右对齐，或者 12 位数据左对齐，或者 12 位数据右对齐方式时，对寄存器 DAC\_DxDT8R [7: 0]，或者 DAC\_DxDT12L [15: 4]，或者 DAC\_DxDT12R [11: 0]位写入值。

**双 DAC 数据配置方式：**采用 8 位数据右对齐，或者 12 位数据左对齐，或者 12 位数据右对齐方式时，对双 DAC 的 8 位右对齐数据保持寄存器（DAC\_DDTH8R）[7: 0]和双 DAC 的 8 位右对齐数据保持寄存器（DAC\_DDTH8R）[15: 8]，或者双 DAC 的 12 位左对齐数据保持寄存器（DAC\_DDTH12L）[15: 4]和双 DAC 的 12 位左对齐数据保持寄存器（DAC\_DDTH12L）[31: 20]，或者双 DAC 的 12 位右对齐数据保持寄存器（DAC\_DDTH12R）[11: 0]和双 DAC 的 12 位右对齐数据保持寄存器（DAC\_DDTH12R）[27: 16]位写入值。

写入的 8 位数据对应 DHRx[11: 4]位，写入的 12 位数据对应 DHRx[11: 0]位。

## 20.5 DAC寄存器

必须以字（32 位）的方式操作这些外设寄存器。

下表列出了所有 DAC 寄存器。

表 20-2 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_D1DTH12R	008h	0x0000 0000
DAC_D1DTH12L	00Ch	0x0000 0000
DAC_D1DTH8R	010h	0x0000 0000
DAC_D2DTH12R	014h	0x0000 0000
DAC_D2DTH12L	018h	0x0000 0000
DAC_D2DTH8R	01Ch	0x0000 0000
DAC_DDTH12R	020h	0x0000 0000
DAC_DDTH12L	024h	0x0000 0000
DAC_DDTH8R	028h	0x0000 0000
DAC_D1ODT	02Ch	0x0000 0000
DAC_D2ODT	030h	0x0000 0000

### 20.5.1 DAC控制寄存器（DAC\_CTRL）

域	简称	复位值	类型	功能
位 31: 29	保留	0x0	resd	请保持默认值。
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 的 DMA 模式； 1：使能 DAC2 的 DMA 模式。
位 27: 24	D2NBSEL	0x0	rw	DAC2 噪声位宽选择（DAC2 noise bit select） 这些位用于在噪声生成模式下选择屏蔽位，在三角波生成模式下选择波形的幅值。 0000：不屏蔽 LSFR 位 0 / 三角波幅值等于 1； 0001：不屏蔽 LSFR 位[1: 0] / 三角波幅值等于 3； 0010：不屏蔽 LSFR 位[2: 0] / 三角波幅值等于 7； 0011：不屏蔽 LSFR 位[3: 0] / 三角波幅值等于 15； 0100：不屏蔽 LSFR 位[4: 0] / 三角波幅值等于 31； 0101：不屏蔽 LSFR 位[5: 0] / 三角波幅值等于 63； 0110：不屏蔽 LSFR 位[6: 0] / 三角波幅值等于 127； 0111：不屏蔽 LSFR 位[7: 0] / 三角波幅值等于 255； 1000：不屏蔽 LSFR 位[8: 0] / 三角波幅值等于 511； 1001：不屏蔽 LSFR 位[9: 0] / 三角波幅值等于 1023； 1010：不屏蔽 LSFR 位[10: 0] / 三角波幅值等于 2047； ≥1011：不屏蔽 LSFR 位[11: 0] / 三角波幅值等于 4095。
位 23: 22	D2NM	0x0	rw	DAC2 噪声/三角波生成选择（DAC2 noise mode） 00：关闭； 01：开启噪声波形发生器； 1x：开启三角波发生器。
位 21: 19	D2TRGSEL	0x0	rw	DAC2 的触发事件选择（DAC2 trigger select） 000：TMR6 TRGOUT 事件； 001：TMR8 TRGOUT 事件； 010：TMR7 TRGOUT 事件； 011：TMR5 TRGOUT 事件；

				100: TMR2 TRGOUT 事件; 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_D2DTHx 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_D2DTHx 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_D2DTHx 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	D2OBDIS	0x0	rw	关闭 DAC2 输出缓存 (DAC2 output buffer disable) 0: 开启输出缓存; 1: 关闭输出缓存。
位 16	D2EN	0x0	rw	DAC2 使能 (DAC2 enable) 0: 关闭; 1: 开启。
位 15: 13	保留	0x0	resd	请保持默认值。
位 12	D1DMAEN	0x0	rw	DAC1 的 DMA 传输使能 (DAC1 DMA transfer enable) 0: 关闭; 1: 开启。
位 11: 8	D1NBSEL	0x0	rw	DAC1 屏蔽/幅值选择 (DAC1 noise bit select) 这些位用于在噪声生成模式下选择屏蔽位, 在三角波生成模式下选择波形的幅值。 0000: 不屏蔽 LSFR 位 0 / 三角波幅值等于 1; 0001: 不屏蔽 LSFR 位[1: 0] / 三角波幅值等于 3; 0010: 不屏蔽 LSFR 位[2: 0] / 三角波幅值等于 7; 0011: 不屏蔽 LSFR 位[3: 0] / 三角波幅值等于 15; 0100: 不屏蔽 LSFR 位[4: 0] / 三角波幅值等于 31; 0101: 不屏蔽 LSFR 位[5: 0] / 三角波幅值等于 63; 0110: 不屏蔽 LSFR 位[6: 0] / 三角波幅值等于 127; 0111: 不屏蔽 LSFR 位[7: 0] / 三角波幅值等于 255; 1000: 不屏蔽 LSFR 位[8: 0] / 三角波幅值等于 511; 1001: 不屏蔽 LSFR 位[9: 0] / 三角波幅值等于 1023; 1010: 不屏蔽 LSFR 位[10: 0] / 三角波幅值等于 2047; ≥1011: 不屏蔽 LSFR 位[11: 0] / 三角波幅值等于 4095。
位 7: 6	D1NM	0x0	rw	DAC1 噪声/三角波生成选择 (DAC1 noise mode) 00: 关闭; 01: 开启噪声波形发生器; 1x: 开启三角波发生器。
位 5: 3	D1TRGSEL	0x0	rw	DAC1 的触发事件选择 (DAC1 trigger select) 000: TMR6 TRGOUT 事件; 001: TMR8 TRGOUT 事件; 010: TMR7 TRGOUT 事件; 011: TMR5 TRGOUT 事件; 100: TMR2 TRGOUT 事件; 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	D1OBDIS	0x0	rw	关闭 DAC1 输出缓存（DAC1 output buffer disable） 0：开启输出缓存； 1：关闭输出缓存。
位 0	D1EN	0x0	rw	DAC1 使能（DAC1 enable） 0：关闭； 1：开启。

## 20.5.2 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 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 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。

## 20.5.3 DAC1的12位右对齐数据保持寄存器（DAC\_D1DTH12R）

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	D1DT12R	0x000	rw	DAC1 的 12 位右对齐数据（DAC1 12-bit right-aligned data）

## 20.5.4 DAC1的12位左对齐数据保持寄存器（DAC\_D1DTH12L）

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: 4	D1DT12L	0x000	rw	DAC1 的 12 位左对齐数据（DAC1 12-bit left-aligned data）
位 3: 0	保留	0x0	resd	请保持默认值。

## 20.5.5 DAC1的8位右对齐数据保持寄存器（DAC\_D1DTH8R）

域	简称	复位值	类型	功能
位 31: 8	保留	0x000000	resd	请保持默认值。
位 7: 0	D1DT8R	0x00	rw	DAC1 的 8 位右对齐数据（DAC1 8-bit right-aligned data）

## 20.5.6 DAC2的12位右对齐数据保持寄存器（DAC\_D2DTH12R）

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	D2DT12R	0x000	rw	DAC2 的 12 位右对齐数据（DAC2 12-bit right-aligned data）



### 20.5.7 DAC2的12位左对齐数据保持寄存器（DAC\_D2DTH12L）

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: 4	D2DT12L	0x000	rw	DAC2 的 12 位左对齐数据（DAC2 12-bit left-aligned data）
位 3: 0	保留	0x0	resd	请保持默认值。

### 20.5.8 DAC2的8位右对齐数据保持寄存器（DAC\_D2DTH8R）

域	简称	复位值	类型	功能
位 31: 8	保留	0x000000	resd	请保持默认值。
位 7: 0	D2DT8R	0x00	rw	DAC2 的 8 位右对齐数据（DAC2 8-bit right-aligned data）

### 20.5.9 双DAC的12位右对齐数据保持寄存器（DAC\_DDTH12R）

域	简称	复位值	类型	功能
位 31: 28	保留	0x0	resd	请保持默认值。
位 27: 16	DD2DT12R	0x000	rw	DAC2 的 12 位右对齐数据（DAC2 12-bit right-aligned data）
位 15: 12	保留	0x0	resd	请保持默认值。
位 11: 0	DD1DT12R	0x000	rw	DAC1 的 12 位右对齐数据（DAC1 12-bit right-aligned data）

### 20.5.10 双DAC的12位左对齐数据保持寄存器（DAC\_DDTH12L）

域	简称	复位值	类型	功能
位 31: 20	DD2DT12L	0x000	rw	DAC2 的 12 位左对齐数据（DAC2 12-bit left-aligned data）
位 19: 16	保留	0x0	resd	请保持默认值。
位 15: 4	DD1DT12L	0x000	rw	DAC1 的 12 位左对齐数据（DAC1 12-bit left-aligned data）
位 3: 0	保留	0x0	resd	请保持默认值。

### 20.5.11 双DAC的8位右对齐数据保持寄存器（DAC\_DDTH8R）

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: 8	DD2DT8R	0x00	rw	DAC2 的 8 位右对齐数据（DAC2 8-bit right-aligned data）
位 7: 0	DD1DT8R	0x00	rw	DAC1 的 8 位右对齐数据（DAC1 8-bit right-aligned data）

### 20.5.12 DAC1数据输出寄存器（DAC\_D1ODT）

域	简称	复位值	类型	功能
位 31: 12	保留	0x000000	resd	请保持默认值。
位 11: 0	D1ODT	0x000	rw	DAC1 输出数据（DAC1 output data）

### 20.5.13 DAC2数据输出寄存器（DAC\_D2ODT）

域	简称	复位值	类型	功能
位 31: 12	保留	0x000000	resd	请保持默认值。
位 11: 0	D2ODT	0x000	rw	DAC2 输出数据（DAC2 output data）



## 21 CAN 总线控制器

### 21.1 简介

CAN（Controller Area Network）是一种实现各节点之间实时、可靠数据通信的分布式串行通信协议，支持 CAN 协议 2.0A 和 2.0B。

### 21.2 主要特性

- 波特率最高可达 1M bit/s/
- 支持时间触发通信
- 中断使能和屏蔽
- 自动重传功能可配

#### 发送

- 3 个发送邮箱
- 发送优先级可配置
- 支持发送时间戳

#### 接收

- 2 个深度为 3 的 FIFO
- 14 组过滤器组
- 支持标识符列表模式
- 支持标识符掩码模式
- 支持 FIFO 溢出管理

#### 时间触发通信模式

- 16 位定时器
- 发送时间戳

### 21.3 波特率设置

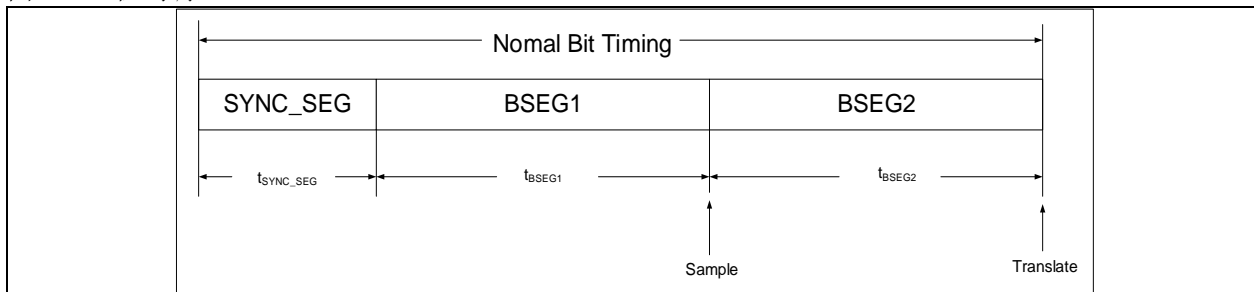
CAN 总线的额定位时间由 3 部分组成。

同步段(SYNC\_SEG)，该段占用 1 时间单元，时间长度由 CAN 位时序寄存器（CAN\_BTMG）的 BRDIV[11: 0]位定义。

位段 1（BIT SEGMENT 1），包括 CAN 标准里的 PROP\_SEG 和 PHASE\_SEG1，记为 BSEG1，该段占用 1 至 16 时间单元，时间单元个数由 BTS1[3: 0]位定义。

位段 2（BIT SEGMENT 2），包括 CAN 标准里的 PHASE\_SEG2，记为 BSEG2，该段占用 1 至 8 时间单元，时间单元个数由 BTS2[2: 0]位定义。

图 21-1 位时序



#### 波特率计算公式

$$BaudRate = \frac{1}{\text{Nomal Bit Timing}}$$

$$\text{Nomal Bit Timing} = t_{\text{SYNC\_SEG}} + t_{\text{BSEG1}} + t_{\text{BSEG2}}$$

其中

$$t_{\text{SYNC\_SEG}} = 1 \times t_q$$

$$t_{\text{BSEG1}} = (1 + \text{BTS1}[3: 0]) \times t_q$$

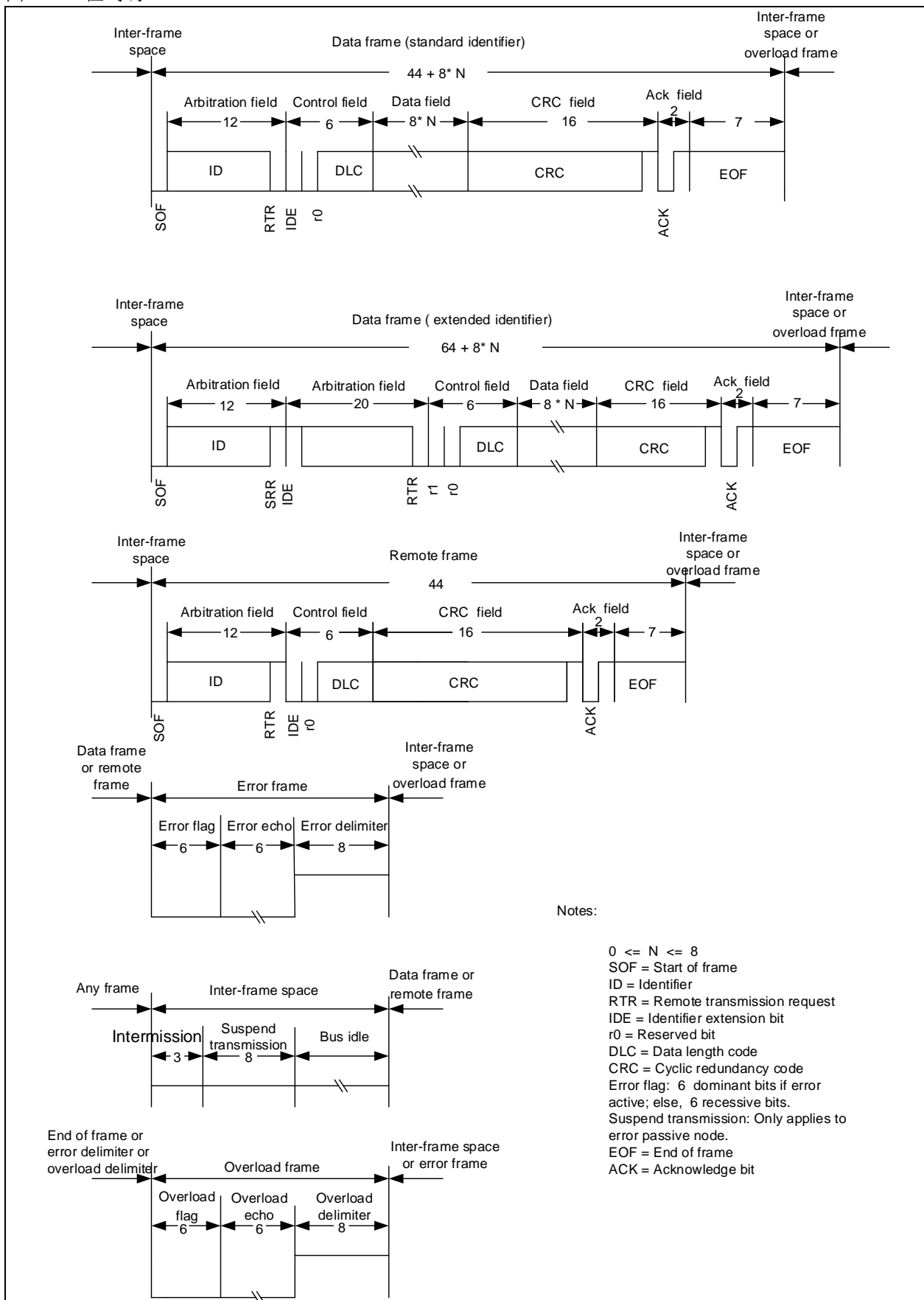
$$t_{BSEG2} = (1 + BTS2[2: 0]) \times t_q$$

$$t_q = (1 + BRDIV[11: 0]) \times t_{pclk}$$

## 硬同步和重同步

默认情况下,CAN 节点的每一位的起始位置总是在同步段内,同时在位段 1 和位段 2 临界位置进行采样。但是由于节点振荡器漂移,网络节点之间的传播延迟以及噪声干扰等,实际的传输过程中,CAN 节点的每一位会存在一定的相位误差。为避免相位误差对通讯造成影响,可以通过帧起始位置的边沿以及后面的下降沿进行硬同步或者重同步,同步补偿的时间长度最长不超过重新同步调整宽度(1 至 4 个时间单元,RSAW[1: 0]位设置)。

图 21-2位时序



## 21.4 中断管理

CAN 控制器具有 4 个中断向量，通过配置 CAN 中断使能寄存器（CAN\_INTEN），可以控制相应的中断开启或关闭。

图 21-3 发送中断的产生

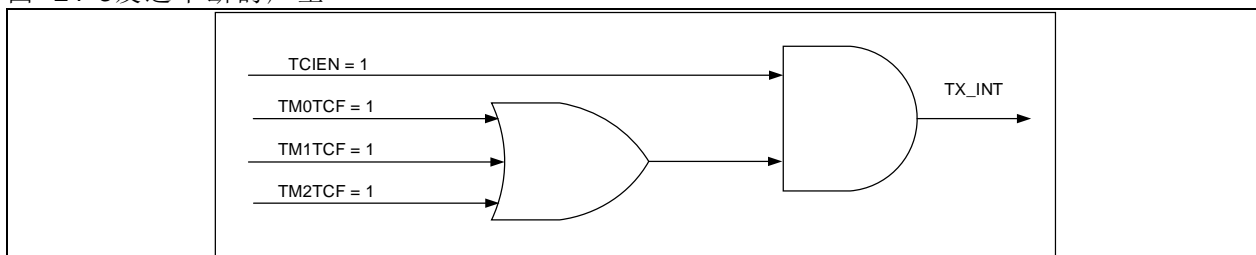


图 21-4 接收中断 0 的产生

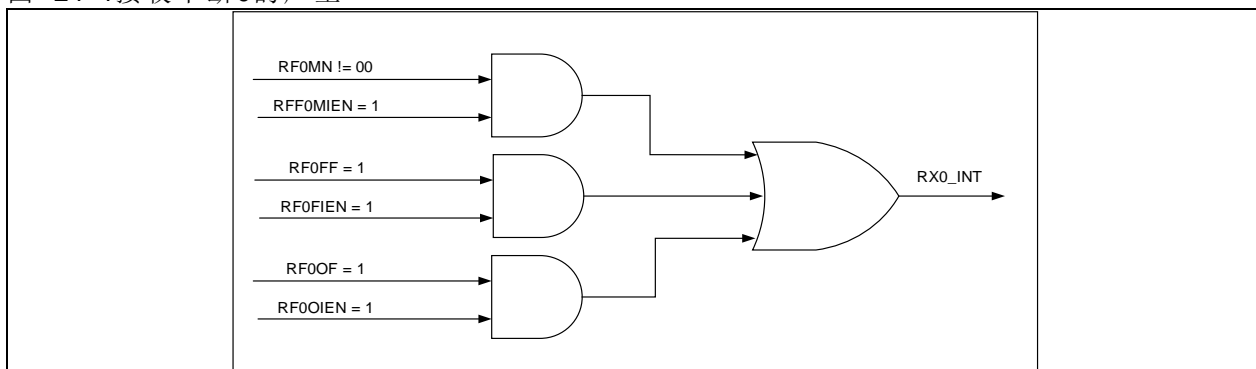


图 21-5 接收中断 1 的产生

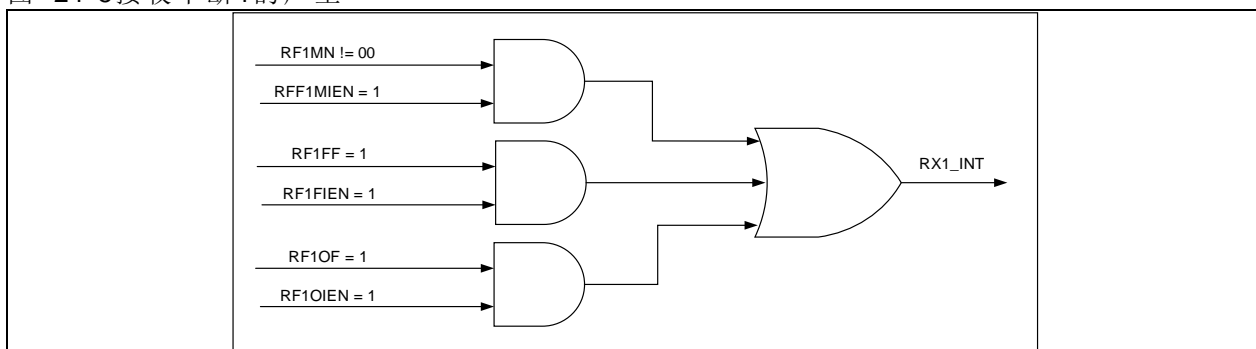
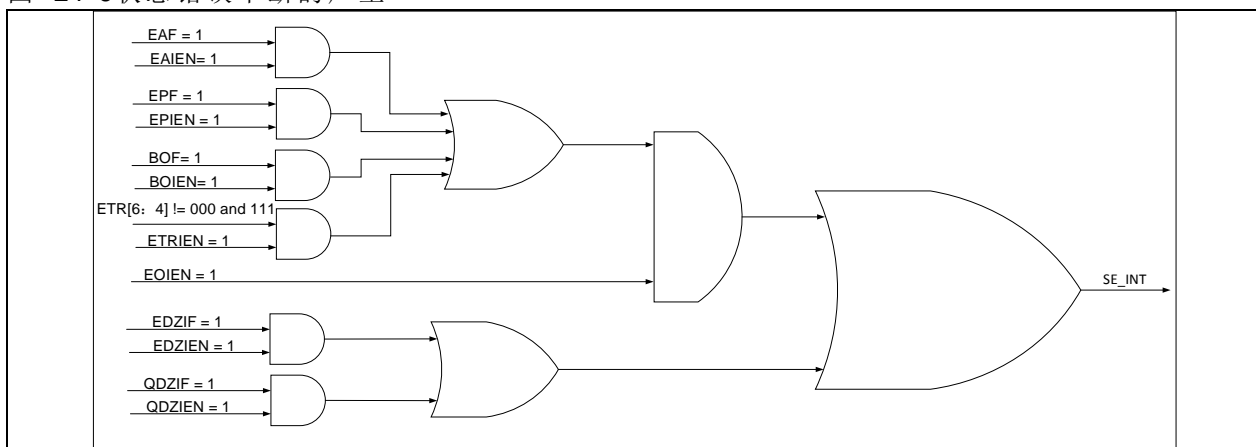


图 21-6 状态错误中断的产生



## 21.5 设计提示

为便于 CAN 应用开发，设计时建议参考如下提示。

- 调试控制

当系统进入调试模式时，可以通过控制 MCU 调试寄存器 DEBUG\_CTRL 的 CANx\_PAUSE

以及 CAN 主控制寄存器（CAN\_MCTRL）的 PTD 位控制 CAN 控制器处于停止状态或者正常发送接收状态。

- 时间触发通信

时间触发通信用于提高系统的实时性，避免总线竞争。当 CAN\_MCTRL 的 TTCEN 位置‘1’，CAN 控制器的时间触发通信即被激活。内部 16 位定时器在每个 CAN 位累加，在帧起始位置被采样，生成时间戳，存储在接收 FIFO 邮箱数据长度和时间戳寄存器（CAN\_RFCx）/发送邮箱数据长度和时间戳寄存器（CAN\_TMCx）中。

- 寄存器访问保护

CAN 位时序寄存器（CAN\_BTMG）只能在冻结工作模式下进行修改。

CAN 节点发送错误数据对网络层不会带来问题，但却会对应用程序造成严重影响，因此只能在发送邮箱为空时改变它。

只有在设置过滤器为配置模式下（即 FCS=1），才能修改过滤器的设置，即修改 CAN 过滤器模式配置寄存器（CAN\_FMCFG），CAN 过滤器位宽配置寄存器

（CAN\_FBWCFG），CAN 过滤器 FIFO 关联寄存器（CAN\_FRF）。过滤位寄存器 x（CAN\_FiFBx）只有在过滤器配置模式下（即 FCS=1）或者相应过滤器关闭情况下（即 FAENx=0）才能进行修改。

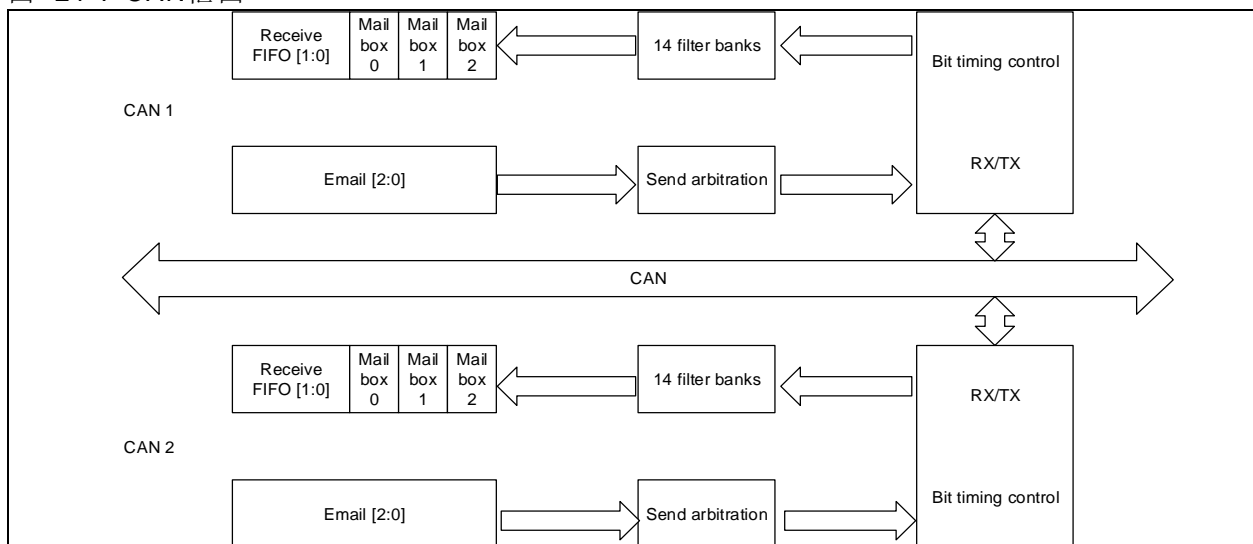
## 21.6 功能描述

### 21.6.1 整体功能描述

随着 CAN 网络节点和报文数量的增加，需要一个增强的过滤机制处理各种类型的报文，减少接收报文的处理时间，采用 FIFO 的方案，使得 CPU 可以长时间处理应用层任务而不会丢失报文。同时发送报文由硬件配置发送优先级顺序，并且完全支持标准标识符（11 位）和扩展标识符（29 位）。

基于以上考虑，CAN 控制器提供 14 组位宽可变/可配置的标识符过滤器组，2 个接收 FIFO，每个 FIFO 都可以存放 3 个完整的报文，且完全由硬件管理，共有 3 个发送邮箱，发送调度器决定发送优先级顺序。

图 21-7 CAN 框图



### 21.6.2 工作模式

CAN 控制器有 3 种工作模式：

- 睡眠模式

系统复位之后，CAN 控制器处于睡眠模式，该模式下 CAN 的时钟停止，因此可以节省电能，但软件仍然可以访问邮箱寄存器，同时内部上拉电阻被禁用。

软件通过对 CAN 主控制寄存器（CAN\_MCTRL）的 DZEN 位置‘1’，可以请求 bxCAN 进入睡眠模式，并且硬件对 CAN 主状态寄存器（CAN\_MSTS）的 DZC 位置‘1’进行确认。

有两种方式退出睡眠模式：配置 CAN 主控制寄存器（CAN\_MCTRL）的 AEDEN 位为‘1’，一旦检测到 CAN 总线的活动，硬件自动对 DZEN 位清‘0’来唤醒 CAN 控制器。或者软件对 DZEN 位清‘0’可以退出睡眠状态。

睡眠工作模式进入冻结工作模式：对 CAN 主控制寄存器（CAN\_MCTRL）的 FZEN 位置‘1’，并且同时

对 DZEN 位清 ‘0’，然后硬件对 CAN 主状态寄存器（CAN\_MSTS）的 FZC 位置‘1’来进行确认。  
睡眠工作模式进入通讯工作模式：对 CAN 主控制寄存器（CAN\_MCTRL）FZEN 和 DZEN 位清 ‘0’，并且 CAN 控制器必须跟总线取得同步，即在 CANRX 管脚上监测到 11 个连续的隐性位。

#### ● 冻结模式

软件对 CAN 控制器的初始化，只能在冻结模式下进行，包括位 CAN 位时序寄存器（CAN\_BTMG）和 CAN 主控制寄存器（CAN\_MCTRL）这 2 个寄存器。对 CAN 控制器的 14 组过滤器组（包括模式、位宽、FIFO 关联、激活和过滤器值）进行初始化可以在非冻结模式下进行。当 CAN 处于冻结模式时，禁止报文的接收和发送。

冻结工作模式进入通讯工作模式：对 CAN 主控制寄存器（CAN\_MCTRL）FZEN 位清 ‘0’，硬件对 CAN 主状态寄存器（CAN\_MSTS）的 FZC 位清‘0’就确认了冻结模式的退出，并且 CAN 控制器必须跟总线取得同步。

冻结工作模式进入睡眠工作模式：对 CAN 主控制寄存器（CAN\_MCTRL）DZEN 位置 ‘1’，CAN 主控制寄存器（CAN\_MCTRL）FZEN 位清 ‘0’，并且硬件对 CAN 主状态寄存器（CAN\_MSTS）的 DZC 位置 ‘1’ 进行确认。

#### ● 通讯模式

在冻结工作模式配置完成 CAN 位时序寄存器（CAN\_BTMG）和 CAN 主控制寄存器（CAN\_MCTRL）这两个寄存器后，控制 CAN 进入通讯工作模式，开始报文收发过程。

通讯工作模式进入睡眠工作模式：对 CAN 主控制寄存器（CAN\_MCTRL）DZEN 位置 ‘1’，并等待当前 CAN 总线传输完成。

通讯工作模式进入冻结工作模式：对 CAN 主控制寄存器（CAN\_MCTRL）FZEN 位置 ‘1’，并等待当前 CAN 总线传输完成。

### 21.6.3 测试方法

CAN 控制器定义了三种方法用于测试分析，包括只听方式、回环方式以及回环只听方式，可以通过 CAN 位时序寄存器（CAN\_BTMG）的 LOEN 位和 LBEN 位进行配置。

- 当 CAN 位时序寄存器（CAN\_BTMG）[31]位为 ‘1’ 时采用只听方式，此时 CAN 可以正常接收数据，但发送端 CANTX 固定隐性位输出。同时，发送端 CANTX 发出的显性位可以被接收端检测到，但是不会影响到 CAN 总线。

- 当 CAN 位时序寄存器（CAN\_BTMG）[30]位为 ‘1’ 时采用回环方式，此时 CAN 只会接收本节点发送端 CANTX 的电平信号，同时 CAN 可以发送数据至外部总线，回环方式主要用于本节点的自我检测。

- 当 CAN 位时序寄存器（CAN\_BTMG）[31: 30]位为 ‘11’ 时，只听方式和回环方式同时有效，此时 CAN 与总线网络断开，发送端 CANTX 固定隐性位输出，并且发送端直接与接收端相连。

### 21.6.4 报文过滤

在接收到的报文会根据其标识符（ID）进行过滤，通过过滤的报文会存储在对应的 FIFO 中，没有通过的报文则会被丢弃，整个过程由硬件自动完成，不会占用 CPU 开销。

#### 过滤器的位宽

每个 CAN 控制器提供 14 个位宽可变、可配置的过滤器组（0~13），每个过滤器组由 2 个 32 位寄存器，CAN\_FiFB1 和 CAN\_FiFB2 组成，通过配置 CAN 过滤器位宽配置寄存器（CAN\_FBWCFG）的对应位，设置过滤器位宽为 2 个 16 位或者单个 32 位。

32 位宽的过滤器寄存器 CAN\_FiFBx 包括：SID[10: 0]、EID[17: 0]、IDT 和 RTR 位。

CAN_FiFB1[31: 21]	CAN_FiFB1[20: 3]	CAN_FiFB1[2: 0]			
CAN_FiFB2[31: 21]	CAN_FiFB2[20: 3]	CAN_FiFB2[2: 0]			
SID[10: 0]/EID[28: 18]	EID[17: 0]	IDT	RTR	0	

2 个 16 位宽的过滤器寄存器 CAN\_FiFBx 包括：SID[10: 0]、IDT、RTR 和 EID[17: 15]位。

CAN_FiFB1[31: 21]	CAN_FiFB1 [20: 19]	CAN_FiFB1 [18: 16]	CAN_FiFB1[15: 5]	CAN_FiFB1 [4: 3]	CAN_FiFB1 [2: 0]
CAN_FiFB2[31: 21]	CAN_FiFB2 [20: 19]	CAN_FiFB2 [18: 16]	CAN_FiFB2[15: 5]	CAN_FiFB2 [4: 3]	CAN_FiFB2 [2: 0]
SID[10: 0]	IDT	RTR	EID[17: 15]	SID[10: 0]	IDT RTR EID[17: 15]

### 过滤器模式

通过设置 CAN 过滤器模式配置寄存器（CAN\_FMCFG）的 FMSELx 位可以设置过滤器寄存器工作在标识符掩码模式或者标识符列表模式，掩码模式用来指定哪些位与预设标识符相同，哪些位无需比较，列表模式表示标识符（ID 号）必须与预设标识符一致。两种模式与过滤器位宽配合使用，可以有以下四种过滤方式：

图 21-8 32位宽标识符掩码模式

ID	CAN_FiFB1[31:21]	CAN_FiFB1[20:3]	CAN_FiFB1 [2:0]	
Mask	CAN_FiFB2[31:21]	CAN_FiFB2[20:3]	CAN_FiFB2 [2:0]	
Mapping	SID[10:0]	EID[17:0]	IDT	RTR 0

图 21-9 32位宽标识符列表模式

ID	CAN_FiFB1[31:21]	CAN_FiFB1[20:3]	CAN_FiFB1 [2:0]	
ID	CAN_FiFB2[31:21]	CAN_FiFB2[20:3]	CAN_FiFB2 [2:0]	
Mapping	SID[10:0]	EID[17:0]	IDT	RTR 0

图 21-10 16位宽标识符掩码模式

ID	CAN_FiFB1[15:5]	CAN_FiFB1[4:0]	
Mask	CAN_FiFB1[31:21]	CAN_FiFB1[20:16]	
ID	CAN_FiFB2[15:5]	CAN_FiFB2[4:0]	
Mask	CAN_FiFB2[31:21]	CAN_FiFB2[20:16]	
Mapping	SID[10:0]	RTR	IDT EID[17:15]

图 21-11 16位宽标识符列表模式

ID	CAN_FiFB1[15:8]	CAN_FiFB1[7:0]	
ID	CAN_FiFB1[31:24]	CAN_FiFB1[23:16]	
ID	CAN_FiFB2[15:8]	CAN_FiFB2[7:0]	
ID	CAN_FiFB2[31:24]	CAN_FiFB2[23:16]	
Mapping	SID[10:0]	RTR	IDT EID[17:15]

### 过滤器匹配序号

14 组过滤器组根据位宽模式的不同，具有不同的过滤效果，例如 32 位宽标识符掩码模式包含序号为 n 的过滤器，而 16 位宽标识符列表模式包含序号为 n、n+1、n+2 以及 n+3 的过滤器。一帧报文通过了某个序号（Filter Nnumber）N 的过滤器，则该帧的接收 FIFO 邮箱数据长度和时间戳寄存器（CAN\_RFCx）RFFMN[7: 0]位存储该序号 N，过滤器序号的分配不关心对应的过滤器组是否处于激活状态。

下表为过滤器匹配序号的示例。



Filter bank	FIFO0	Active	Filter number	Filter bank	FIFO1	Active	Filter number
0	CAN_F0FB1[31: 0]-ID	Yes	0	3	CAN_F3FB1[15: 0]-ID	Yes	0
	CAN_F0FB2[31: 0]-ID		1		CAN_F3FB1[31: 16]-ID		1
1	CAN_F1FB1[15: 0]-ID	Yes	2		CAN_F3FB2[15: 0]-ID		2
	CAN_F1FB1[31: 16]-ID		3		CAN_F3FB2[31: 16]-ID		3
	CAN_F1FB2[15: 0]-ID		4	4	CAN_F4FB1[31: 0]-ID	Yes	4
	CAN_F1FB2[31: 16]-ID		5		CAN_F4FB2[31: 0]-Mask		
2	CAN_F2FB1[31: 0]-ID	Yes	6	5	CAN_F5FB1[15: 0]-ID	No	5
	CAN_F2FB2[31: 0]-Mask				CAN_F5FB1[31: 16]-Mask		
6	CAN_F6FB1[15: 0]-ID	No	7		CAN_F5FB2[15: 0]-ID		6
	CAN_F6FB1[31: 16]-Mask				CAN_F5FB2[31: 16]-Mask		
	CAN_F6FB2[15: 0]-ID		8	7	CAN_F7FB1[15: 0]-ID	No	7
	CAN_F6FB2[31: 16]-Mask				CAN_F7FB1[31: 16]-ID		8
9	CAN_F9FB1[31: 0]-ID	No	9		CAN_F7FB2[15: 0]-ID		9
	CAN_F9FB2[31: 0]-ID		10		CAN_F7FB2[31: 16]-ID		10
10	CAN_F10FB1[15: 0]-ID	Yes	11	8	CAN_F8FB1[31: 0]-ID	Yes	11
	CAN_F10FB1[31: 16]-Mask				CAN_F8FB2[31: 0]-Mask		
	CAN_F10FB2[15: 0]-ID		12	11	CAN_F11FB1[31: 0]-ID	Yes	12
	CAN_F10FB2[31: 16]-Mask				CAN_F11FB2[31: 0]-ID		13
12	CAN_F12FB1[15: 0]-ID	No	13	13	CAN_F13FB1[15: 0]-ID	Yes	14
	CAN_F12FB1[31: 16]-ID		14		CAN_F13FB1[31: 16]-ID		15
	CAN_F12FB2[15: 0]-ID		15		CAN_F13FB2[15: 0]-ID		16
	CAN_F12FB2[31: 16]-ID		16		CAN_F13FB2[31: 16]-ID		17

## 优先级匹配规则

CAN 控制器接收一帧报文，有可能能够通过多个过滤器的过滤，在这种情况下，存放在接收邮箱中的过滤器匹配序号，根据以下优先级规则确定。

- 位宽为 32 位的过滤器，优先级高于位宽为 16 位的过滤器。
- 在相同位宽的情况下，标识符列表模式的优先级高于标识符掩码模式。
- 在位宽和标识符模式都相同的情况下，标号越小的过滤器具有更高的优先级。

## 过滤器配置

- 将 CAN 过滤器控制寄存器（CAN\_FCTRL）FCS 位置 ‘1’，允许配置 CAN 过滤器。
- 写 CAN 过滤器模式配置寄存器（CAN\_FMCFG）FMSELx 位，控制过滤器工作模式为标识符掩码模式或者列表模式。
- 写 CAN 过滤器位宽配置寄存器（CAN\_FBWCFG）FBWSELx 位，控制过滤器位宽为 2 个 16 位或者单个 32 位。
- 写 CAN 过滤器 FIFO 关联寄存器（CAN\_FRF）FRFSELx 位，关联过滤器 x 到 FIFO0 或者 FIFO1。

- 将 CAN 过滤器激活控制寄存器（CAN\_FACFG）FAENx 位置 ‘1’，激活对应的过滤器组 x。
- 写 CAN\_FiFBx（其中 i=0...13；x=1,2），配置 0~13 组过滤器组。
- 将 CAN 过滤器控制寄存器（CAN\_FCTRL）FCS 位置 ‘0’，完成 CAN 过滤器配置过程。

## 21.6.5 报文发送

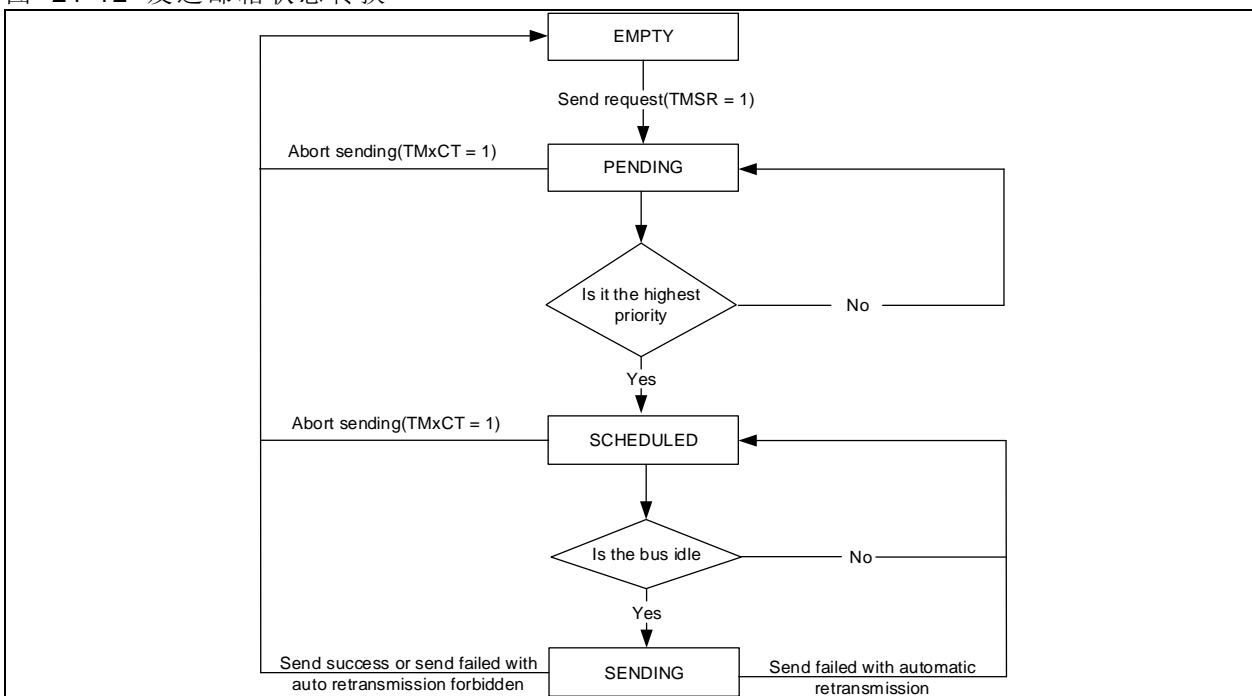
### 寄存器配置

数据发送首先需要选择发送邮箱进行配置，对应的寄存器为发送邮箱标识符寄存器（CAN\_TMIx）、发送邮箱数据长度和时间戳寄存器（CAN\_TMCx）、发送邮箱低字节数据寄存器（CAN\_TMDTLx）以及发送邮箱高字节数据寄存器（CAN\_TMDTHx）。当邮箱配置完成后，对发送邮箱标识符寄存器（CAN\_TMIx）TMSR 位置 ‘1’ 控制 CAN 启动发送流程。

### 报文发送

当对应的邮箱配置完成且 CAN 控制器接收到发送请求后，该邮箱进入 PENDING 状态，此时 CAN 控制器会检查该邮箱是否处于最高优先级状态，如果是则进入 SCHEDULED 状态，否则停下等待该邮箱获取最高优先级。处于 SCHEDULED 状态的邮箱会实时监控 CAN 总线状态，只要总线空闲，预定发送邮箱中的报文就马上被发送。发送完成，该邮箱进入 EMPTY 状态。

图 21-12 发送邮箱状态转换



### 发送优先级配置

当有两个及以上发送邮箱处于 PENDING 状态时，需要决定邮箱的发送优先级。

由标识符决定：当 CAN 主控制寄存器（CAN\_MCTRL）的 MMSSR 位置 ‘0’，发送顺序由邮箱中报文的标识符决定。标识符数值低的报文具有更高优先级，相同标识符的，邮箱号小的报文优先发送。

由发送请求顺序决定：当 CAN 主控制寄存器（CAN\_MCTRL）的 MMSSR 位置 ‘1’，发送优先级由各邮箱的发送请求次序决定。

### 发送状态及错误信息

CAN 发送状态寄存器（CAN\_TSTS）中的 TMxTCF、TMxTSF、TMxALF、TMxTEF 以及 TMxEF 用于显示发送状态和错误信息。

TMxTCF 位：发送完成标志。表示本次数据发送完成，置 ‘1’ 有效。

TMxTSF 位：无错误发送完成标志。表示本次数据发送完成且无错误，置 ‘1’ 有效。

TMxALF 位：发送仲裁丢失标志。表示本次数据发送仲裁失败，置 ‘1’ 有效。

TMxTEF 位：发送错误标志。表示本次数据发送检测到总线错误，且发送错误帧，置 ‘1’ 有效。

TMxEF 位：邮箱空标志。表示本次数据发送完成，邮箱变为空状态，置 ‘1’ 有效。

### 数据发送中止

可以通过将 CAN 发送状态寄存器（CAN\_TSTS）的 TMxCT 位置 ‘1’ 中止当前邮箱的发送，具体情况

需要分类讨论。

当前邮箱发送失败或者丢失仲裁，假如报文自动重传功能被禁止，则发送邮箱进入 **EMPTY** 状态；假如报文自动重传功能被使能，则发送邮箱进入 **SCHEDULED** 状态，接着邮箱发送被中止，进入 **EMPTY** 状态。当前邮箱本次数据发送完成且无错误，邮箱进入 **EMPTY** 状态。

## 21.6.6 报文接收

### 寄存器配置

用户程序通过读接收 FIFO 邮箱标识符寄存器 (**CAN\_RFIx**)、接收 FIFO 邮箱数据长度和时间戳寄存器 (**CAN\_RFCx**)、接收 FIFO 邮箱低字节数据寄存器 (**CAN\_RFDTLx**) 以及接收 FIFO 邮箱高字节数据寄存器 (**CAN\_RFDTHx**) 获取接收到的有效报文。

### 报文接收

**CAN** 控制器具有两个深度为 3 的 FIFO 用于接收报文，采用先进先出的原则。当报文被正确接收且通过了标识符过滤，则被认为是有效报文并存储在对应的 FIFO 中。接收 FIFO 每接收到一帧有效报文，**CAN\_RFx** 寄存器中的报文数目 **RFxMN[1: 0]** 就加 1，当 **RFxMN[1: 0]** 等于 3 的同时又接收到一帧有效报文，此时控制器会根据 **CAN** 主控制寄存器 (**CAN\_MCTRL**) 的 **MDRSEL** 位选择覆盖接收到的原有的报文或者丢弃该报文。

同时，当用户每次读出一帧报文且控制 **CAN\_RFx** 寄存器 **RFxR** 位置 ‘1’，则对应 FIFO 释放一个深度空间，并且 **CAN\_RFx** 寄存器中的报文数目 **RFxMN[1: 0]** 减 1。

### 接收 FIFO 状态

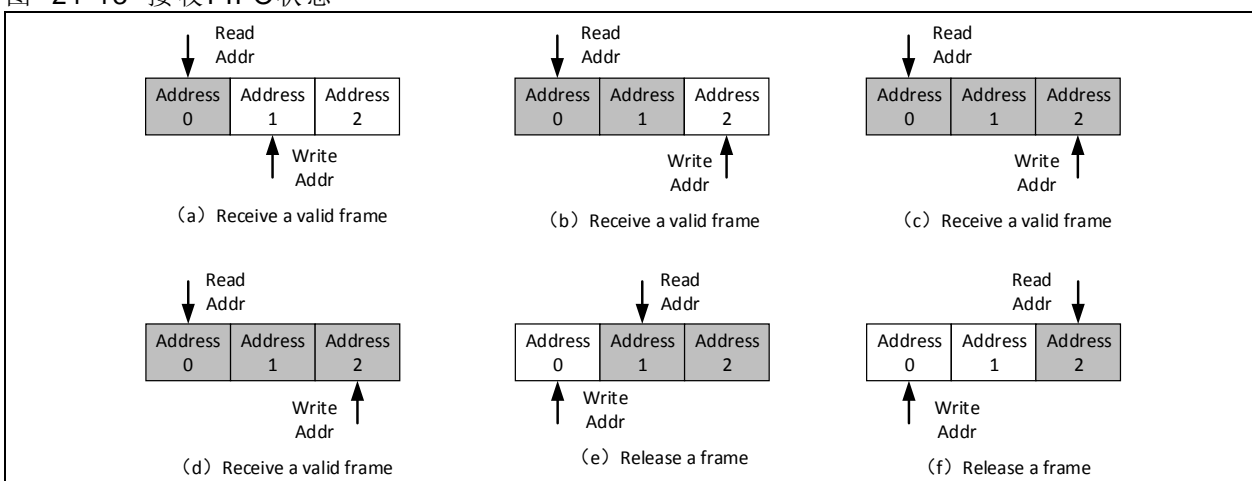
寄存器 **RFx** 中的 **RFxMN[1: 0]**、**RFxFF** 以及 **RFxOF** 用于显示接收 FIFO 的状态信息。

**RFxMN[1: 0]**：表示 FIFOx 中当前存储有效报文的数目。

**RFxFF**：表示 FIFOx 中当前存储 3 帧有效报文处于 ‘满’ 状态，如图 (c) 所示。

**RFxOF**：表示 FIFOx 中当前有 3 帧有效报文同时又接收到一帧有效报文，处于溢出状态，如图 (d) 所示。

图 21-13 接收 FIFO 状态



## 21.6.7 出错管理

**CAN** 总线的状态可以根据发送错误计数值 **TEC** 和接收错误计数值 **REC** 表明当前 **CAN** 节点所处的状态，同时 **CAN** 错误状态寄存器 (**CAN\_ESTS**) 的 **ETR[6: 4]** 位用于记录上次错误的原因，这些错误状态在 **CAN** 中断使能寄存器 (**CAN\_INTEN**) 控制下产生中断。

- 主动错误状态：当 **TEC** 且 **REC** 计数值都小于 128 时，系统处于主动错误状态，当检测到错误时发送主动错误标志。
- 被动错误状态：当 **TEC** 或 **REC** 计数值大于 127 时，系统处于被动错误状态，当检测到错误时发送被动错误标志。
- 离线状态：当 **TEC** 大于 255 时，系统进入离线状态，处于离线状态的节点不能发送接收报文，从离线状态恢复分两种情况。当 **CAN** 主控制寄存器 (**CAN\_MCTRL**) **AEBOEN** 位为 ‘0’ 时，通信模式下 **CAN** 节点 **RX** 检测到 128 次 11 个连续隐性位，接着软件请求进入冻结模式，随后从离线状态恢复。当 **AEBOEN** 位为 ‘1’ 时，通信模式下 **CAN** 节点 **RX** 检测到 128 次 11 个连续隐性位，随后自动从离线状态恢复。

## 21.7 CAN寄存器

必须以字（32 位）的方式操作这些外设寄存器。

表 21-1 CAN寄存器映像和复位值

寄存器简称	基址偏移量	复位值
MCTRL	000h	0x0001 0002
MSTS	004h	0x0000 0C02
TSTS	008h	0x1C00 0000
RF0	00Ch	0x0000 0000
FR1	010h	0x0000 0000
INTEN	014h	0x0000 0000
ESTS	018h	0x0000 0000
BTMG	01Ch	0x0123 0000
保留	020h~17Fh	xx
TMI0	180h	0xFFFF XXXX
TMC0	184h	0xFFFF XXXX
TMDTL0	188h	0xFFFF XXXX
TMDTH0	18Ch	0xFFFF XXXX
TMI1	190h	0xFFFF XXXX
TMC1	194h	0xFFFF XXXX
TMDTL1	198h	0xFFFF XXXX
TMDTH1	19Ch	0xFFFF XXXX
TMI2	1A0h	0xFFFF XXXX
TMC2	1A4h	0xFFFF XXXX
TMDTL2	1A8h	0xFFFF XXXX
TMDTH2	1ACh	0xFFFF XXXX
RFI0	1B0h	0xFFFF XXXX
RFC0	1B4h	0xFFFF XXXX
RFDTL0	1B8h	0xFFFF XXXX
RFDTL0	1BCh	0xFFFF XXXX
RFI1	1C0h	0xFFFF XXXX
RFC1	1C4h	0xFFFF XXXX
RFDTL1	1C8h	0xFFFF XXXX
RFDTL1	1CCh	0xFFFF XXXX
保留	1D0h~1FFh	xx
FCTRL	200h	0x2A1C 0E01
FMCFG	204h	0x0000 0000
保留	208h	xx
FSCFG	20Ch	0x0000 0000
保留	210h	xx
FRF	214h	0x0000 0000
保留	218h	xx

FACFG	21Ch	0x0000 0000
保留	220h~23Fh	xx
FB0F1	240h	0xFFFF XXXX
FB0F2	244h	0xFFFF XXXX
FB1F1	248h	0xFFFF XXXX
FB1F2	24Ch	0xFFFF XXXX
...	...	...
FB13F1	2A8h	0xFFFF XXXX
FB13F2	2ACh	0xFFFF XXXX

## 21.7.1 CAN控制和状态寄存器

### 21.7.1.1 CAN主控制寄存器（CAN\_MCTRL）

域	简称	复位值	类型	功能
位 31: 17	保留	0x0000	resd	请保持默认值。
位 16	PTD	0x1	rw	调试时禁止收发（Prohibit trans when debug） 0：不禁止； 1：禁止。仍然可以正常地读写和控制接收 FIFO。 注：仅 PTD 及 DEBUG 控制寄存器（DEBUG_CTRL）的 CANx_PAUSE 同时置位时，才会实现禁止收发的效果。
位 15	SPRST	0x0	rw1s	部分软复位（Software partial reset） 0：不复位； 1：部分复位。 注： SPRST 指只复位接收 FIFO 及 MCTRL 寄存器。 复位后 CAN 进入睡眠模式。此后硬件自动对该位清零。
位 14: 8	保留	0x00	resd	请保持默认值。
位 7	TTCEN	0x0	rw	时间触发通信模式使能（Time triggered communication mode enable） 0：关闭； 1：开启。
位 6	AEBOEN	0x0	rw	自动退出离线状态使能（Automatic exit bus-off enable） 0：关闭； 1：开启。 注： 当开启时，硬件只需检测到 CAN 总线上出现退出时序就自动退出； 当关闭时，需要软件执行一次额外的冻结模式的进入以及退出动作，接着在 CAN 总线上检测到退出时序时才会退出离线状态。
位 5	AEDEN	0x0	rw	自动退出睡眠模式使能（Automatic exit doze mode enable） 0：关闭； 1：开启。 注： 当关闭时，需软件写清睡眠请求命令来退出； 当开启时，无需软件干预，只要检测到 CAN 总线上出现报文时就立即退出睡眠模式。
位 4	PRSFEN	0x0	rw	发送失败时禁止重传使能（Prohibit retransmission when sending fails enable） 0：关闭； 1：开启。
位 3	MDRSEL	0x0	rw	接收溢出时报文丢弃规则选择（Message discarding rule select when overflow） 0：最先收到的报文被丢弃； 1：最新收到的报文被丢弃。

位 2	MMSSR	0x0	rw	多报文发送顺序规则选择（Multiple message sending sequence rule） 0：标识符最小的最先被发送； 1：最先请求的最先被发送。
位 1	DZEN	0x1	rw	睡眠模式使能（Doze mode enable） 0：关闭； 1：开启。 注： 当设置了 AEDEN，且检测到 CAN 总线上出现报文时，硬件会自动退出睡眠模式； 在 CAN 复位或部分软复位后，该位被硬件强制置位，即 CAN 默认将处于睡眠模式。
位 0	FZEN	0x0	rw	冻结模式使能（Freeze mode enable） 0：关闭； 1：开启。 注： 当写关闭命令时，会在检测到接收管脚上出现连续的 11 个隐性位才会实际退出。因此软件需等待 FZC 被硬件清零来确认。 当写开启命令时，会在当前的 CAN 活动（发送或接收）结束后才会实际进入。因此软件需等待 FZC 被硬件置位来确认。

## 21.7.1.2 CAN主状态寄存器（CAN\_MSTS）

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	REALRX	0x1	ro	接收管脚实时电平（Real time level of RX pin） 0：低电平； 1：高电平。
位 10	LSAMPRX	0x1	ro	接收管脚上次采样电平（Last sample level of RX pin） 0：低电平； 1：高电平。 注：此值会跟随 REALRX 实时更新。
位 9	CURS	0x0	ro	当前的接收状态（Currently receiving status） 0：未接收； 1：正在接收。 注：在 CAN 开始接收时硬件置位此标志，接收完毕后硬件自动清除。
位 8	CUSS	0x0	ro	当前的发送状态（Currently sending status） 0：未发送； 1：正在发送。 注：在 CAN 开始发送时硬件置位此标志，发送完毕后硬件自动清除。
位 7: 5	保留	0x0	resd	请保持默认值。
位 4	EDZIF	0x0	rw1c	进入睡眠模式的中断标志（Enter doze mode interrupt flag） 0：未进入或无标志置位条件； 1：已进入。 注： 只有当 EDZIEN=1，且 CAN 进入睡眠模式时才会由硬件置位此标志。此标志的置位将会产生一个状态改变中断。此标志可由软件清零（对自身写一），或当 DZC 位被清零时硬件自动对本标志清零。
位 3	QDZIF	0x0	rw1c	退出睡眠模式的中断标志（Quit doze mode interrupt flag） 0：未退出或无退出条件； 1：已退出或产生了退出条件。 注： 该位由软件将其清零（对自身写一）。 退出条件为检测到总线上出现帧起始位（SOF）。



				如果 QDZIEN=1，此标志的置位将会产生一个状态改变中断。
位 2	EOIF	0x0	rw1c	<p>出现错误的中断标志（Error occur Interrupt flag）</p> <p>0：未出现或无标志置位条件；</p> <p>1：已出现。</p> <p>注：</p> <p>该位由软件将其清零（对自身写一）。</p> <p>仅当 CAN 错误状态寄存器（CAN_ESTS）中的某位被置位，且其对应的 CAN 中断使能寄存器（CAN_INTEN）的相应中断使能位处于使能状态时，该标志才会被硬件置位。此标志的置位将会产生一个状态改变中断。</p>
位 1	DZC	0x1	ro	<p>睡眠模式确认（Doze mode confirm）</p> <p>0：未处于睡眠模式；</p> <p>1：正处于睡眠模式中。</p> <p>注：</p> <p>该位用于确定 CAN 当前是否处于睡眠模式，是对软件请求进入睡眠模式的确认。</p> <p>当进入睡眠模式时，会在当前的 CAN 活动（发送或接收）结束后才会实际进入。因此软件需等待本标志被硬件置位来确认进入睡眠模式。</p> <p>当退出睡眠模式（即软件写关闭睡眠模式命令，或者自动退出睡眠模式使能状态下检测到 CAN 总线上出现报文）时，会在检测到 CAN 的 RX 管脚上出现连续的 11 位隐性位时才会实际退出。因此软件需等待本标志被硬件清零来确认退出睡眠模式。</p>
位 0	FZC	0x0	ro	<p>冻结模式确认（Freeze mode confirm）</p> <p>0：未处于冻结模式；</p> <p>1：正处于冻结模式中。</p> <p>注：</p> <p>该位用于确定 CAN 当前是否处于冻结模式，是对软件请求进入冻结模式的确认。</p> <p>当进入冻结模式时，会在当前的 CAN 活动（发送或接收）结束后才会实际进入。因此软件需等待本标志被硬件置位来确认进入冻结模式。</p> <p>当退出冻结模式时，会在检测到 CAN 的 RX 管脚上出现连续的 11 位隐性位时才会实际退出。因此软件需等待本标志被硬件清零来确认退出冻结模式。</p>

### 21.7.1.3 CAN发送状态寄存器（CAN\_TSTS）

域	简称	复位值	类型	功能
位 31	TM2LPF	0x0	ro	<p>邮箱 2 优先级最低标志（Transmit mailbox 2 lowest priority flag）</p> <p>0：非最低优先级；</p> <p>1：最低优先级（表明多个邮箱在等待发送报文时，邮箱 2 的优先级最低）。</p>
位 30	TM1LPF	0x0	ro	<p>邮箱 1 优先级最低标志（Transmit mailbox 1 lowest priority flag）</p> <p>0：非最低优先级；</p> <p>1：最低优先级（表明多个邮箱在等待发送报文时，邮箱 1 的优先级最低）。</p>
位 29	TM0LPF	0x0	ro	<p>邮箱 0 最低优先级标志（Transmit mailbox 0 lowest priority flag）</p> <p>0：非最低优先级；</p> <p>1：最低优先级（表明多个邮箱在等待发送报文时，邮箱 0 的优先级最低）。</p>
位 28	TM2EF	0x1	ro	<p>发送邮箱 2 空标志（Transmit mailbox 2 empty flag）</p> <p>当发送邮箱 2 中没有等待发送的报文时，硬件置位该位。</p>
位 27	TM1EF	0x1	ro	<p>发送邮箱 1 空标志（Transmit mailbox 1 empty flag）</p> <p>当发送邮箱 1 中没有等待发送的报文时，硬件置位该位。</p>
位 26	TM0EF	0x1	ro	<p>发送邮箱 0 空标志（Transmit mailbox 0 empty flag）</p>



				当发送邮箱 0 中没有等待发送的报文时，硬件置位该位。
				发送邮箱号记录（Transmit Mailbox number record）
				注： 当有发送邮箱为空时，这两位表示接下来将要使用的空置邮箱号。 示例：CAN 空闲状态下，写一个报文的发送命令后，这 2 位的值将变为 01。 当没有发送邮箱为空时，这两位表示优先级最低的那个发送邮箱号。 示例：3 个报文待发，报文标识符依次为，邮箱 0 为 0x400，邮箱 1 为 0x433，邮箱 2 为 0x411，此时这 2 位的值将变为 01。
位 25: 24	TMNR	0x0	ro	
				邮箱 2 取消发送（Transmit mailbox 2 cancel transmit）
				0：无意义； 1：取消发送。 注：软件设置此位可中断邮箱 2 的发送，硬件清除邮箱 2 的发送报文后同步清除该位。若邮箱 2 为空置邮箱时，软件置位该位没有任何意义。
位 23	TM2CT	0x0	rw1s	
				保持默认值。
位 22: 20	保留	0x0	resd	
				邮箱 2 发送错误标志（Transmit mailbox 2 transmission error flag）
				0：无错误； 1：出现错误。 注： 当邮箱 2 出现发送错误时置位该位。 可由软件对该位写一清零。或在启动下一次发送时由硬件清除此标志。
位 19	TM2TEF	0x0	rw1c	
				邮箱 2 仲裁丢失标志（Transmit mailbox 2 arbitration lost flag）
				0：无仲裁问题； 1：出现仲裁丢失。 注： 当邮箱 2 因仲裁丢失导致发送失败时置位该位。 可由软件对该位写一清零。或在启动下一次发送时由硬件清除此标志。
位 18	TM2ALF	0x0	rw1c	
				邮箱 2 发送成功标志（Transmit mailbox 2 transmission success flag）
				0：发送失败； 1：发送成功。 注： 该位实时指示每次邮箱 2 的发送结果。可由软件对该位写一清零。
位 17	TM2TSF	0x0	rw1c	
				邮箱 2 发送完成标志（transmit mailbox 2 transmission completed flag）
				0：正在发送； 1：发送完成。 注： 每次对邮箱 2 的请求（发送或中止）完成后，由硬件置位该位。 该位可由软件写一清零。或当接收到新的发送请求时由硬件自动清除。 当该位被清除时，邮箱 2 的 TSMF2、ALMF2、TEMF2 也会同步被硬件清除。
位 16	TM2TCF	0x0	rw1c	
				邮箱 1 取消发送（Transmit mailbox 1 cancel transmit）
				0：无意义； 1：取消发送。 注：软件设置此位可禁止邮箱 1 的发送，硬件清除邮箱 1 的发送报文后同步清除该位。若邮箱 1 为空置邮箱时，软件置位该位没有任何意义。
位 15	TM1CT	0x0	rw1s	
				保持默认值。
位 14: 12	保留	0x0	resd	

位 11	TM1TEF	0x0	rw1c	<p>邮箱 1 发送错误标志 (Transmit mailbox 1 transmission error flag)</p> <p>0: 无错误; 1: 出现错误。</p> <p>注: 当邮箱 1 出现发送错误时置位该位。 可由软件对该位写一清零。或在启动下一次发送时由硬件清除此标志。</p>
位 10	TM1ALF	0x0	rw1c	<p>邮箱 1 仲裁丢失标志 (Transmit mailbox 1 arbitration lost flag)</p> <p>0: 无仲裁问题; 1: 出现仲裁丢失。</p> <p>注: 当邮箱 1 因仲裁丢失导致发送失败时置位该位。 可由软件对该位写一清零。或在启动下一次发送时由硬件清除此标志。</p>
位 9	TM1TSF	0x0	rw1c	<p>邮箱 1 发送成功标志 (Transmit mailbox 1 transmission success flag)</p> <p>0: 发送失败; 1: 发送成功。</p> <p>注: 该位实时指示每次邮箱 1 的发送结果。可由软件对该位写一清零。</p>
位 8	TM1TCF	0x0	rw1c	<p>邮箱 1 发送完成标志 (Transmit mailbox 1 transmission completed flag)</p> <p>0: 正在发送; 1: 发送完成。</p> <p>注: 每次对邮箱 1 的请求 (发送或中止) 完成后, 由硬件置位该位。 该位可由软件写一清零。或当接收到新的发送请求时由硬件自动清除。 当该位被清除时, 邮箱 1 的 TSMF1、ALMF1、TEMF1 也会同步被硬件清除。</p>
位 7	TM0CT	0x0	rw1s	<p>邮箱 0 取消发送 (Transmit mailbox 0 cancel transmit)</p> <p>0: 无意义; 1: 取消发送。</p> <p>注: 软件设置此位可禁止邮箱 0 的发送, 硬件清除邮箱 0 的发送报文后同步清除该位。若邮箱 0 为空置邮箱时, 软件置位该位没有任何意义。</p>
位 6: 4	保留	0x0	resd	保持默认值。
位 3	TM0TEF	0x0	rw1c	<p>邮箱 0 发送错误标志 (Transmit mailbox 0 transmission error flag)</p> <p>0: 无错误; 1: 出现错误。</p> <p>注: 当邮箱 0 出现发送错误时置位该位。 可由软件对该位写一清零。或在启动下一次发送时由硬件清除此标志。</p>
位 2	TM0ALF	0x0	rw1c	<p>邮箱 0 仲裁丢失标志 (Transmit mailbox 0 arbitration lost flag)</p> <p>0: 无仲裁问题; 1: 出现仲裁丢失。</p> <p>注: 当邮箱 0 因仲裁丢失导致发送失败时置位该位。 可由软件对该位写一清零。或在启动下一次发送时由硬件清除此标志。</p>
位 1	TM0TSF	0x0	rw1c	<p>邮箱 0 发送成功标志 (Transmit mailbox 0 transmission success flag)</p> <p>0: 发送失败; 1: 发送成功。</p>

				注： 该位实时指示每次邮箱 0 的发送结果。可由软件对该位写一清零。
				邮箱 0 发送完成标志（Transmit mailbox 0 transmission completed flag） 0：正在发送； 1：发送完成。
位 0	TM0TCF	0x0	rw1c	注： 每次对邮箱 0 的请求（发送或中止）完成后，由硬件置位该位。 该位可由软件写一清零。或当接收到新的发送请求时由硬件自动清除。 当该位被清除时，邮箱 0 的 TSMF0、ALMF0、TEMF0 也会同步被硬件清除。

#### 21.7.1.4 CAN接收FIFO 0寄存器（CAN\_RF0）

域	简称	复位值	类型	功能
位 31: 6	保留	0x0000000	resd	保持默认值。
位 5	RF0R	0x0	rw1s	释放接收 FIFO 0（Receive FIFO 0 release） 0：无意义； 1：释放 FIFO。 注： 软件设置此位可释放接收 FIFO 0，当 FIFO 0 被释放时，硬件对该位清零。 接收 FIFO 0 为空时，软件置位该位没有任何意义。 若 FIFO 0 中有 2 个以上的报文时，软件需要执行一次释放命令后才能访问第 2 个报文。
				接收 FIFO 0 溢出标志（Receive FIFO 0 overflow flag） 0：无溢出； 1：有溢出。 注： 当 FIFO 0 已满时，又收到了新的符合过滤条件的报文，硬件将置位该位。 该位由软件写一清零。
位 4	RF0OF	0x0	rw1c	接收 FIFO 0 满标志（Receive FIFO 0 full flag） 0：未滿； 1：已滿。 注： 当 FIFO 0 中存储 3 笔待读取的报文时，硬件将置位该位。 该位由软件写一清零。
位 3	RF0FF	0x0	rw1c	保持默认值。
位 2	保留	0x0	resd	FIFO 0 报文数目（Receive FIFO 0 message num） 注： 这 2 位表示存储在 FIFO 0 中的待读取或者处理的报文数目。 当 FIFO 0 未滿时，每收到了一笔新的符合过滤条件的报文，硬件就对 RF0ML 加 1。 每当软件对 RF0R 位写一来释放接收 FIFO 0 时，RF0ML 就会被减 1，直到其值为 0。
位 1: 0	RF0MN	0x0	ro	

## 21.7.1.5 CAN接收FIFO 1寄存器 (CAN\_RF1)

域	简称	复位值	类型	功能
位 31: 6	保留	0x0000000	resd	保持默认值。
位 5	RF1R	0x0	rw1s	释放接收 FIFO 1 (Receive FIFO 1 release) 0: 无意义; 1: 释放 FIFO。 注: 软件设置此位可释放接收 FIFO 1, 当 FIFO 1 被释放时, 硬件对该位清零。 接收 FIFO 1 为空时, 软件置位该位没有任何意义。 若 FIFO 1 中有 2 个以上的报文时, 软件需要执行一次释放命令后才能访问第 2 个报文。
位 4	RF1OF	0x0	rw1c	接收 FIFO 1 溢出标志 (Receive FIFO 1 overflow flag) 0: 无溢出; 1: 有溢出。 注: 当 FIFO 1 已满时, 又收到了新的符合过滤条件的报文, 硬件将置位该位。 该位由软件写一清零。
位 3	RF1FF	0x0	rw1c	接收 FIFO 1 满标志 (Receive FIFO 1 full flag) 0: 未滿; 1: 已滿。 注: 当 FIFO 1 中存储 3 笔待读取的报文时, 硬件将置位该位。 该位由软件写一清零。
位 2	保留	0x0	resd	保持默认值。
位 1: 0	RF1MN	0x0	ro	FIFO 1 报文数目 (Receive FIFO 1 message num) 注: 这 2 位表示存储在 FIFO 1 中的待读取或者处理的报文数目。 当 FIFO 1 未滿时, 每收到了一笔新的符合过滤条件的报文, 硬件就对 RF1ML 加 1。 每当软件对 RF1R 位写一来释放接收 FIFO 1 时, RF1ML 就会被硬件减 1, 直到其值为 0。

## 21.7.1.6 CAN中断使能寄存器 (CAN\_INTEN)

域	简称	复位值	类型	功能
位 31: 18	保留	0x0000	resd	保持默认值。
位 17	EDZIEN	0x0	rw	进入睡眠模式的中断使能 (Enter doze mode interrupt enable) 0: 关闭; 1: 开启。 注: 此中断对应的标志位为 EDZIF, 故仅本中断使能且 EDZIF 被置位时才会产生中断。
位 16	QDZIEN	0x0	rw	退出睡眠模式的中断使能 (Quit doze mode interrupt enable) 0: 关闭; 1: 开启。 注: 此中断对应的标志位为 QDZIF, 故仅本中断使能且 QDZIF 被置位时才会产生中断。
位 15	EOIEN	0x0	rw	出现错误的中断使能 (Error occur interrupt enable) 0: 关闭; 1: 开启。 注: 此中断对应的标志位为 EOIF, 故仅本中断使能且 EOIF 被置位时才会产生中断。
位 14: 12	保留	0x0	resd	保持默认值。
位 11	ETRIEN	0x0	rw	错误类型记录中断使能 (Error type record interrupt enable)

				0: 关闭; 1: 开启。 注: 只有此中断使能后, 硬件设置 ETR[2: 0]时, 才会同步设置 EOIF 位为'1'。
位 10	BOIEN	0x0	rw	总线关闭中断使能 (Bus-off interrupt enable) 0: 关闭; 1: 开启。 注: 只有此中断使能后, 硬件设置 BOF 时, 才会同步设置 EOIF 位为'1'。
位 9	EPIEN	0x0	rw	错误被动中断使能 (Error passive interrupt enable) 0: 关闭; 1: 开启。 注: 只有此中断使能后, 硬件设置 EPF 时, 才会同步设置 EOIF 位为'1'。
位 8	EAIEN	0x0	rw	错误警告中断使能 (Error active interrupt enable) 0: 关闭; 1: 开启。 注: 只有此中断使能后, 硬件设置 EAF 时, 才会同步设置 EOIF 位为'1'。
位 7	保留	0x0	resd	保持默认值。
位 6	RF1OIEN	0x0	rw	接收 FIFO 1 溢出中断使能 (Receive FIFO 1 overflow interrupt enable) 0: 关闭; 1: 开启。 注: 此中断对应的标志位为 RF1OF, 故仅本中断使能且 RF1OF 被置位时才会产生中断。
位 5	RF1FIEN	0x0	rw	接收 FIFO 1 满中断使能 (Receive FIFO 1 full interrupt enable) 0: 关闭; 1: 开启。 注: 此中断对应的标志位为 RF1FF, 故仅本中断使能且 RF1FF 被置位时才会产生中断。
位 4	RF1MIEN	0x0	rw	接收 FIFO 1 报文接收中断使能 (FIFO 1 receive message interrupt enable) 0: 关闭; 1: 开启。 注: 此中断对应的标志位为 RF1MN, 故仅本中断使能且 RF1MN 为非零时才会产生中断。
位 3	RF0OIEN	0x0	rw	接收 FIFO 0 溢出中断使能 (Receive FIFO 0 overflow interrupt enable) 0: 关闭; 1: 开启。 注: 此中断对应的标志位为 RF0OF, 故仅本中断使能且 RF0OF 被置位时才会产生中断。
位 2	RF0FIEN	0x0	rw	接收 FIFO 0 满中断使能 (Receive FIFO 0 full interrupt enable) 0: 关闭; 1: 开启。 注: 此中断对应的标志位为 RF0FF, 故仅本中断使能且 RF0FF 被置位时才会产生中断。
位 1	RF0MIEN	0x0	rw	接收 FIFO 0 报文接收中断使能 (FIFO 0 receive message interrupt enable) 0: 关闭; 1: 开启。 注: 此中断对应的标志位为 RF0MN, 故仅本中断使能且 RF0MN 为非零时才会产生中断。
位 0	TCIEN	0x0	rw	发送邮箱发送完成中断使能 (Transmit mailbox empty interrupt enable) 0: 关闭; 1: 开启。

注：此中断对应的标志位为 TMxTCF，故仅本中断使能且 TMxTCF 被置位时才会产生中断。

## 21.7.1.7 CAN错误状态寄存器（CAN\_ESTS）

域	简称	复位值	类型	功能
位 31: 24	REC	0x00	ro	接收错误计数器（Receive error counter） 这个计数器按照 CAN 协议的故障界定机制的接收部分实现。
位 23: 16	TEC	0x00	ro	发送错误计数器（Transmit error counter） 这个计数器按照 CAN 协议的故障界定机制的发送部分实现。
位 15: 7	保留	0x00	resd	保持默认值。
位 6: 4	ETR	0x0	rw	错误类型记录（Error type record） 000：没有错误； 001：位填充错误； 010：格式错误； 011：确认错误； 100：隐性位错误； 101：显性位错误； 110：CRC 错误； 111：由软件设置。 注： 这三位用于记录最新错误类型，由硬件根据 CAN 总线上的出错情况设置。当报文被正确发送或接收后，硬件自动将这三位清零。 硬件没有使用错误代码 7，软件可以设置该值，从而可以检测代码的更新。
位 3	保留	0x0	resd	保持默认值。
位 2	BOF	0x0	ro	总线关闭标志（Bus-off flag） 0：未处于总线关闭状态； 1：处于总线关闭状态。 注：当发送错误计数器 TEC 溢出（即大于 255）时，CAN 进入总线关闭状态，硬件对该位置'1'。
位 1	EPF	0x0	ro	错误被动标志（Error passive flag） 0：未处于错误被动状态； 1：处于错误被动状态。 注：当前记录的出错次数达到错误被动状态（即接收错误计数器或发送错误计数器的值>127）时，硬件对该位置'1'。
位 0	EAF	0x0	ro	错误主动标志（Error active flag） 0：未处于错误主动状态； 1：处于错误主动状态。 注：当前记录的出错次数达到错误主动状态（即接收错误计数器或发送错误计数器的值≥96）时，硬件对该位置'1'。

### 21.7.1.8 CAN位时序寄存器 (CAN\_BTMG)

域	简称	复位值	类型	功能
位 31	LOEN	0x0	rw	只听模式使能 (Listen-Only mode) 0: 关闭; 1: 开启。
位 30	LBEN	0x0	rw	回环模式使能 (Loop back mode) 0: 关闭; 1: 开启。
位 29: 26	保留	0x0	resd	保持默认值。
位 25: 24	RSAW	0x1	rw	重新同步调整宽度 (Resynchronization width) $t_{RSAW} = t_{CAN} \times (RSAW[1: 0] + 1)$ 。 注: 该位域定义了 CAN 硬件在每位中可以延长或缩短多少个时间单元的上限。
位 23	保留	0x0	resd	保持默认值。
位 22: 20	BTS2	0x2	rw	位时间段 2 (Bit time segment 2) $t_{BTS2} = t_{CAN} \times (BTS2[2: 0] + 1)$ 。 注: 该位域定义了位时间段 2 占用了多少个时间单元。
位 19: 16	BTS1	0x3	rw	位时间段 1 (Bit time segment 1) $t_{BTS1} = t_{CAN} \times (BTS1[3: 0] + 1)$ 注: 该位域定义了位时间段 1 占用了多少个时间单元。
位 15: 12	保留	0x0	resd	保持默认值。
位 11: 0	BRDIV	0x000	rw	波特率分频器 (Baud rate division) $t_q = (BRDIV[11: 0] + 1) \times t_{PCLK}$ 注: 该位域定义了时间单元 (tq) 的时间长度。

### 21.7.2 CAN邮箱寄存器

本节描述发送和接收邮箱寄存器。关于寄存器映像的详细信息, 请参考 21.6.5 节报文。

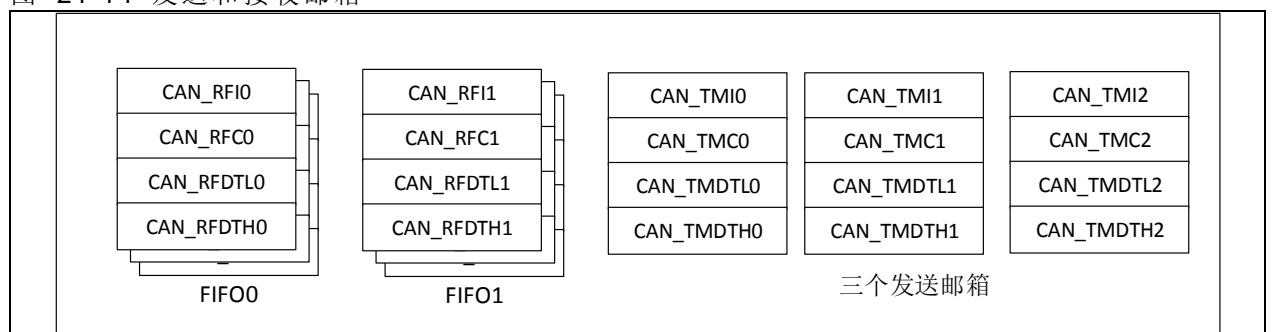
除了下述例外, 发送和接收邮箱几乎一样:

- 接收 FIFO 邮箱数据长度和时间戳寄存器 (CAN\_RFCx) 的 RFFMN 域;
- 接收邮箱是只读的;
- 发送邮箱只有在它为空时才是可写的, CAN 发送状态寄存器 (CAN\_TSTS) 的相应 TM2S 位为 '1', 表示发送邮箱为空。

共有 3 个发送邮箱和 2 个接收邮箱。每个接收邮箱为 3 级深度的 FIFO, 并且只能访问 FIFO 中最先收到的报文。

每个邮箱包含 4 个寄存器。

图 21-14 发送和接收邮箱





### 21.7.2.1 发送邮箱标识符寄存器 (CAN\_TMIx) (x=0..2)

注意：1. 当其所属的邮箱处在等待发送的状态时，该寄存器是写保护的。  
2. 该寄存器实现了发送请求控制功能（第0位）—复位值为0。

域	简称	复位值	类型	功能
位 31: 21	TMSID/ TMEID	0xFFFF	rw	发送邮箱标准标识符或扩展标识符高字节 (Transmit mailbox standard identifier or extended identifier high bytes) 注：这 11 位为标准标识符或扩展标识符的高 11 位。
位 20: 3	TMEID	0xFFFF	rw	发送邮箱扩展标识符低字节 (Transmit mailbox extended identifier) 注：这 18 位为扩展标识符的低 18 位。
位 2	TMIDSEL	0xX	rw	标识符类型选择 (Transmit mailbox identifier type select) 0: 标准标识符; 1: 扩展标识符。
位 1	TMFRSEL	0xX	rw	发送邮箱帧类型选择 (Transmit mailbox frame type select) 0: 数据帧; 1: 远程帧。
位 0	TMSR	0x0	rw	发送邮箱的数据发送请求 (transmit mailbox send request) 0: 无意义; 1: 请求发送。 注：当数据发送完成，邮箱为空时，硬件对其清'0'。

### 21.7.2.2 发送邮箱数据长度和时间戳寄存器 (CAN\_TMCx) (x=0..2)

当邮箱不在空置状态时，该寄存器的所有位为写保护。

域	简称	复位值	类型	功能
位 31: 16	TMTS	0xFFFF	rw	发送邮箱的报文时间戳 (Transmit mailbox time stamp) 注：该时间戳为报文发送帧起始时刻采样到的 CAN 内部定时器的值。
位 15: 9	保留	0xFFFF	resd	保持默认值。
位 8	TMTSTEN	0xX	rw	时间戳的发送使能 (Transmit mailbox time stamp transmit enable) 0: 不发送; 1: 发送。 注： 只有时间触发通信模式使能后，该位才有意义。 时间戳 MTS[15: 0]中，MTS[7: 0]存放于 TMDT7，MTS[15: 8]存放于 TMDT6。故为发送时间戳，发送数据长度需要被设定为 8。
位 7: 4	保留	0xX	resd	保持默认值。
位 3: 0	TMDTBL	0xX	rw	发送数据长度 (Transmit mailbox data byte length) 注：该域指定了发送报文的数据长度。其中 1 个报文可包含 0 到 8 个字节数据。

### 21.7.2.3 发送邮箱低字节数据寄存器 (CAN\_TMDTLx) (x=0..2)

当邮箱不在空置状态时，该寄存器的所有位为写保护。

域	简称	复位值	类型	功能
位 31: 24	TMDT3	0xFF	rw	发送邮箱数据字节 3 (Transmit mailbox data byte 3)
位 23: 16	TMDT2	0xFF	rw	发送邮箱数据字节 2 (Transmit mailbox data byte 2)
位 15: 8	TMDT1	0xFF	rw	发送邮箱数据字节 1 (Transmit mailbox data byte 1)
位 7: 0	TMDT0	0xFF	rw	发送邮箱数据字节 0 (Transmit mailbox data byte 0)

#### 21.7.2.4 发送邮箱高字节数据寄存器 (CAN\_TMDTHx) (x=0..2)

当邮箱不在空置状态时，该寄存器的所有位为写保护。

域	简称	复位值	类型	功能
位 31: 24	TMDT7	0xXX	rw	发送邮箱数据字节 7 (Transmit mailbox data byte 7)
位 23: 16	TMDT6	0xXX	rw	发送邮箱数据字节 6 (Transmit mailbox data byte 6) 注：若时间触发通信模式使能，且对应的时间戳的发送使能，则此位将被 MTS[15: 8]替代。
位 15: 8	TMDT5	0xXX	rw	发送邮箱数据字节 5 (Transmit mailbox data byte 5)
位 7: 0	TMDT4	0xXX	rw	发送邮箱数据字节 4 (Transmit mailbox data byte 4)

#### 21.7.2.5 接收FIFO邮箱标识符寄存器 (CAN\_RFIx) (x=0..1)

注意：所有接收邮箱寄存器都是只读的。

域	简称	复位值	类型	功能
位 31: 21	RFSID/RFEID	0xXXX	ro	接收 FIFO 的标准标识符或扩展标识符 (Receive FIFO standard identifier or receive FIFO extended identifier) 注：这 11 位为标准标识符或扩展标识符的高 11 位。
位 20: 3	RFEID	0xxxxxx	ro	接收 FIFO 的扩展标识符 (Receive FIFO extended identifier) 注：这 18 位为扩展标识符的低 18 位。
位 2	RFIDI	0xX	ro	接收 FIFO 的标识符类型指示 (Receive FIFO identifier type indication) 0：使用标准标识符； 1：使用扩展标识符。
位 1	RFFRI	0xX	ro	接收 FIFO 的帧类型指示 (Receive FIFO frame type indication) 0：数据帧； 1：远程帧。
位 0	保留	0x0	resd	保持默认值。

#### 21.7.2.6 接收FIFO邮箱数据长度和时间戳寄存器 (CAN\_RFCx) (x=0..1)

注意：有接收邮箱寄存器都是只读的。

域	简称	复位值	类型	功能
位 31: 16	RFTS	0xxxxx	ro	接收邮箱的报文时间戳 (Receive FIFO time stamp) 注：该时间戳为报文接收帧起始时刻采样到的 CAN 内部定时器的值。
位 15: 8	RFFMN	0xXX	ro	过滤器匹配序号 (Receive FIFO filter match number) 注：此处存放的是报文通过的那个过滤器序号。
位 7: 4	保留	0xX	resd	保持默认值。
位 3: 0	RFDTL	0xX	ro	接收数据长度 (Receive FIFO data length) 注：该域指定了接收报文的数据长度。其中 1 个报文可包含 0 到 8 个字节数据。对于远程帧，数据长度 RFDTL 固定为 0。

#### 21.7.2.7 接收FIFO邮箱低字节数据寄存器 (CAN\_RFDTLx) (x=0..1)

注意：所有接收邮箱寄存器都是只读的。

域	简称	复位值	类型	功能
位 31: 24	RFDT3	0xXX	ro	接收 FIFO 数据字节 3 (Receive FIFO data byte 3)
位 23: 16	RFDT2	0xXX	ro	接收 FIFO 数据字节 2 (Receive FIFO data byte 2)
位 15: 8	RFDT1	0xXX	ro	接收 FIFO 数据字节 1 (Receive FIFO data byte 1)
位 7: 0	RFDT0	0xXX	ro	接收 FIFO 数据字节 0 (Receive FIFO data byte 0)

### 21.7.2.8 接收FIFO邮箱高字节数据寄存器 (CAN\_RFDTHx) (x=0..1)

注意：所有接收邮箱寄存器都是只读的。

域	简称	复位值	类型	功能
位 31: 24	RFDT7	0xXX	ro	接收 FIFO 数据字节 7 (Receive FIFO data byte 7)
位 23: 16	RFDT6	0xXX	ro	接收 FIFO 数据字节 6 (Receive FIFO data byte 6)
位 15: 8	RFDT5	0xXX	ro	接收 FIFO 数据字节 5 (Receive FIFO data byte 5)
位 7: 0	RFDT4	0xXX	ro	接收 FIFO 数据字节 4 (Receive FIFO data byte 4)

## 21.7.3 CAN过滤器寄存器

### 21.7.3.1 CAN过滤器控制寄存器 (CAN\_FCTRL)

注意：该寄存器的非保留位完全由软件控制。

域	简称	复位值	类型	功能
位 31: 1	保留	0x160E0700	resd	保持默认值。
位 0	FCS	0x1	rw	过滤器组配置控制开关 (Filters configure switch) 0: 关闭 (过滤器组处于工作状态); 1: 开启 (过滤器组处于配置状态)。 注: 过滤器组的初始化配置必须要在过滤器组工作在配置状态下进行。

### 21.7.3.2 CAN过滤器模式配置寄存器 (CAN\_FMCFG)

注意：只有在设置 CAN 过滤器控制寄存器 (CAN\_FCTRL) (FCS=1)，使过滤器处于配置模式下，才能对该寄存器写入。

域	简称	复位值	类型	功能
位 31: 14	保留	0x00000	resd	保持默认值。
位 13: 0	FMSELx	0x0000	rw	过滤器组的模式选择 (Filter mode select) 每一位对应于一个过滤器组 0: 掩码模式; 1: 列表模式。

### 21.7.3.3 CAN过滤器位宽配置寄存器 (CAN\_FBWCFG)

注意：只有在设置 CAN 过滤器控制寄存器 (CAN\_FCTRL) (FCS=1)，使过滤器处于配置模式下，才能对该寄存器写入。

域	简称	复位值	类型	功能
位 31: 14	保留	0x00000	resd	保持默认值。
位 13: 0	FBWSELx	0x0000	rw	过滤器组的位宽选择 (Filter bit width select) 每一位对应于一个过滤器组 0: 2 个 16 位; 1: 单个 32 位。

### 21.7.3.4 CAN过滤器FIFO关联寄存器 (CAN\_FRF)

注意：只有在设置 CAN 过滤器控制寄存器 (CAN\_FCTRL) (FCS=1)，使过滤器处于初始化模式下，才能对该寄存器写入。

域	简称	复位值	类型	功能
位 31: 14	保留	0x00000	resd	保持默认值。
位 13: 0	FRFSELx	0x0000	rw	过滤器组关联 FIFO 选择 (Filter relation FIFO select) 每一位对应于一个过滤器组 0: 关联 FIFO0; 1: 关联 FIFO1。

### 21.7.3.5 CAN过滤器激活控制寄存器 (CAN\_FACFG)

域	简称	复位值	类型	功能
位 31: 14	保留	0x00000	resd	保持默认值。
位 13: 0	FAENx	0x0000	rw	过滤器组激活使能 (Filter active enable) 每一位对应于一个过滤器组 0: 关闭; 1: 开启。

### 21.7.3.6 CAN过滤器组i的过滤位寄存器x (CAN\_FiFBx) (其中i=0..13; x=1..2)

注意: 共有 14 组过滤器:  $i=0..13$ 。每组过滤器由 2 个 32 位的寄存器,  $CAN\_FiFB[2: 1]$  组成。

只有在 CAN 过滤器激活控制寄存器 (CAN\_FACFG) 相应的 FAENx 位清'0', 或 CAN 过滤器控制寄存器 (CAN\_FCTRL) 的 FCS 位为'1'时, 才能修改相应的过滤器寄存器。

域	简称	复位值	类型	功能
位 31: 0	FFDB	0x0000 0000	rw	过滤器过滤数据位 (Filters filter data bit) 列表模式: 寄存器配置值跟总线上接收到的数据对应位的电平完全一致 (如果标准帧则忽略扩展帧对应位数值)。 掩码模式: 只有寄存器配置值为'1'的位才跟总线上接收到的数据对应位的电平一致, 寄存器配置值为'0'的位不关心。

## 22 外部存储控制器（XMC）

### 22.1 XMC简介

XMC 是一个将 AHB 传输信号转换与外部存储器信号相互转换的外设。拥有两个在不同脚位的片选信号，最高可以一次接两个外部存储器。支持的外部存储器有 NAND 闪存，以及具复用信号或有额外增加地址锁存功能的静态存储器件，包含静态随机存储器（SRAM）、NOR 闪存与 PSRAM。

### 22.2 XMC主要特征

NOR/PSRAM 界面有以下特征：

- 支持两个片选信号，拥有各自的控制寄存器
- 支持访问具复用信号或有额外增加地址锁存功能的静态存储器件，包括：
  - 静态随机存储器（SRAM）
  - NOR闪存
  - PSRAM
- 支持 8 位与 16 位数据宽度存储器
- 提供多种时序模式选择
  - 读写相同时序的2种模式
  - 读写不同时序的4种模式
  - 地址数据复用的模式
  - 同步模式
- 具可编程的时序控制寄存器
- 支持将 AHB 数据宽度转换为外部存储器适用的数据宽度

NAND 界面有以下特征：

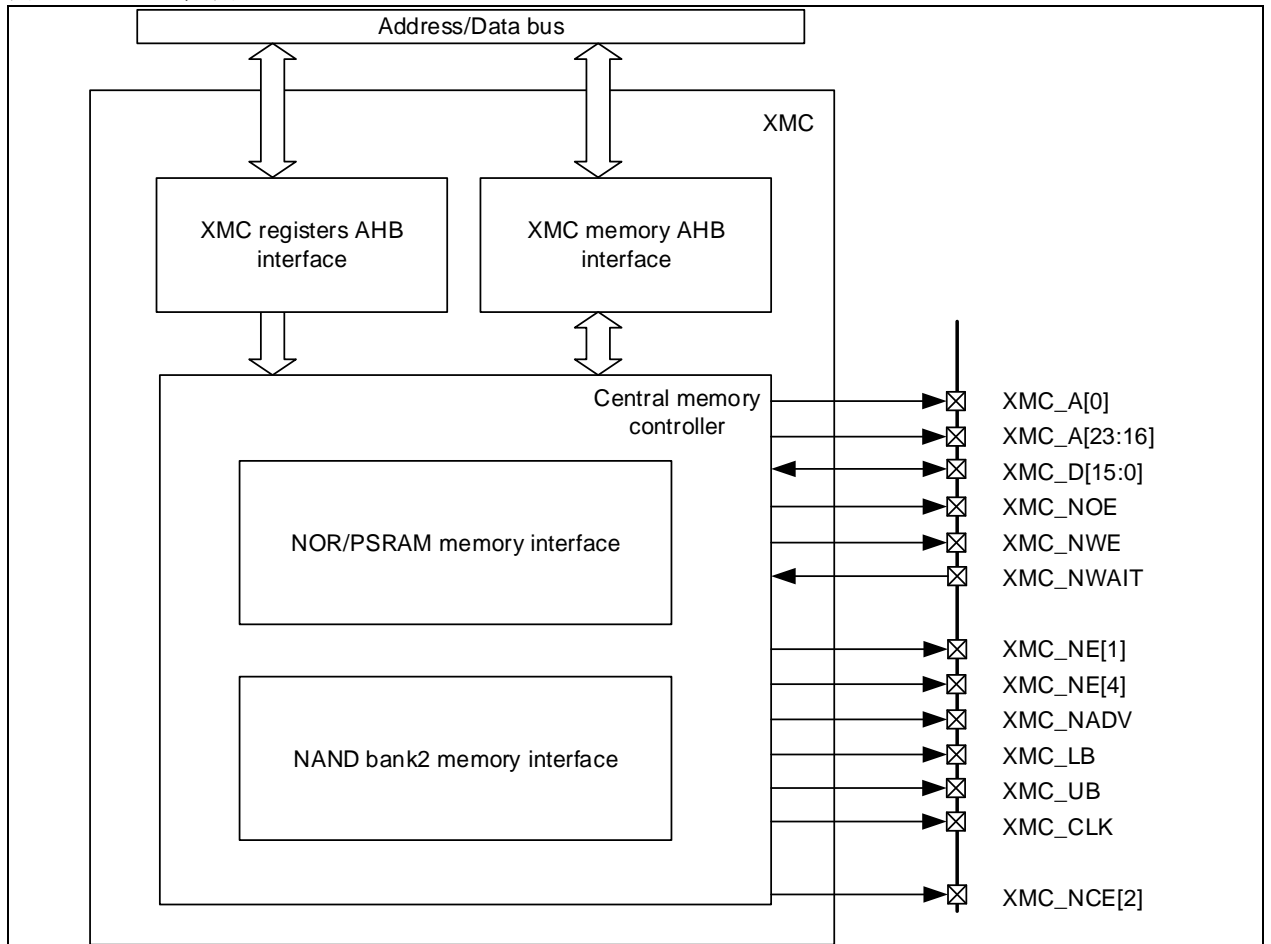
- 支持一个片选信号
- 支持 8 位与 16 位数据宽度 NAND 闪存
- 区分两个存储空间，各自具备可编程的时序控制寄存器
- 支持将 AHB 数据宽度转换为外部存储器适用的数据宽度
- 支持 ECC 运算

## 22.3 XMC构造

### 22.3.1 框图

XMC 的架构如下图所示

图 22-1 XMC框图



与外部存储器沟通时，透过 NOR/PSRAM 界面与透过 NAND 界面所需要使用到的管脚不同，如表 22-1 与

表 22-2 所列。

表 22-1 NOR/PSRAM 界面管脚

管脚	方向	介绍
XMC_CLK	输出	时钟
XMC_NE[x], x=1,4	输出	片选
XMC_NADV	输出	地址锁存或地址有效 (NL) 信号
XMC_A[x]	输出	地址总线
XMC_NOE	输出	输出使能信号
XMC_NWE	输出	写使能信号
XMC_LB、XMC_UB	输出	字节选择信号
XMC_D[15: 0]	读输入/写输出	数据总线/地址数据复用总线
XMC_NWAIT	输入	等待信号

表 22-2 NAND 界面管脚

管脚	方向	介绍
XMC_NCE[2]	输出	片选
XMC_A[17]	输出	地址锁存 (ALE) 信号
XMC_A[16]	输出	命令锁存 (CLE) 信号
XMC_NOE	输出	输出使能 (NRE) 信号
XMC_NWE	输出	写使能信号

XMC_D[15: 0]	读输入/写输出	数据总线
XMC_NWAIT	输入	就绪/忙碌 (R/B) 信号

## 22.3.2 地址映射

XMC 地址分为多个存储块区，如下图所示。

图 22-2 XMC存储块区

Address	Memory banks	Memory chip select signals
6000 0000h	NOR/PSRAM bank1 16 MB	XMC_NE[1]
60FF FFFFh		
	Reserved	
6C00 0000h	NOR/PSRAM bank4 16 MB	XMC_NE[4]
6CFF FFFFh		
	Reserved	
7000 0000h	NAND bank2 common memory 16 MB	XMC_NCE[2]
70FF FFFFh		
	Reserved	
7800 0000h	NAND bank2 attribute memory 16 MB	
78FF FFFFh		

透过 HADDR 的部份特定位数，选择对哪个存储区块读写，如下表所示。



表 22-3 存储区块选择

HADDR[31: 28]		HADDR[27: 26]	
0110: NOR/PSRAM		00: bank1	
		11: bank4	
HADDR[31: 28]	HADDR[27]	HADDR[17: 16]	
0111: NAND bank2	0: 常规空间	00: 数据区	
		01: 命令区	
		1x: 地址区	
	1: 特殊空间	00: 数据区	
		01: 命令区	
		1x: 地址区	

## 22.4 NOR/PSRAM界面

NOR/PSRAM 界面提供多种具有不同时序的访问模式，利用这些模式，可驱动多种存储器：NOR 闪存、SRAM、PSRAM 或 Cellular RAM。

两个存储区块 bank1 与 bank4 有分开的控制寄存器，可使用不同的时序及不同的片选信号访问这两个存储区块。

### 22.4.1 操作方式

#### 管脚使用

不同的外部存储器所需的信号不同，下表列出了典型信号。

表 22-4 NOR闪存与PSRAM典型管脚信号

XMC 管脚信号	NOR 闪存	PSRAM
XMC_CLK	时钟（同步模式）	时钟（同步模式）
XMC_NE[x]	片选信号	片选信号
XMC_NADV	地址锁存或地址有效信号	地址锁存或地址有效信号
XMC_A[23: 16]、 XMC_A[0]	地址总线	地址总线
XMC_NOE	输出使能信号	输出使能信号
XMC_NWE	写使能信号	写使能信号
XMC_LB、XMC_UB	无使用 XMC_LB、XMC_UB 信号	XMC_LB: 低字节选信号 XMC_UB: 高字节选信号
XMC_D[15: 0]	数据总线 地址数据复用总线（复用与同步模式）	数据总线 地址数据复用总线（复用与同步模式）
XMC_NWAIT	NOR 闪存要求等待信号	PSRAM 要求等待信号

注意：若存储器数据宽度为 8 位，典型数据总线为 XMC\_D[7: 0]。

#### 访问地址

HADDR 的高地址用来选择存储区块，低地址选择数据存储地址。HADDR 是字节地址，XMC 可支持字节与半字地址的存储器，地址转换如表 22-5 所示。只要对特定地址作读写，XMC 即可根据 HADDR 启动片选信号并对外部存储器的地址做读写。

表 22-5 HADDR与外部存储器地址转换

外部存储器数据宽度	地址线连接	最大可访问存储器空间（位）
8 位	HADDR[23: 16]与 XMC_A[23: 16]相连。 HADDR[0]与 XMC_A[0]相连。	16M 字节 x8=128 M 位
	复用与同步模式时 HADDR[15: 0]与 XMC_D[15: 0]在地址锁存时间相连。	
16 位	HADDR[23: 17]与 XMC_A[22: 16]相连。 HADDR[1]与 XMC_A[0]相连。	(16M 字节 x16)/2=128 M 位
	复用与同步模式时 HADDR[16: 1]与 XMC_D[15: 0]在地址锁存时间相连。	

#### 访问数据

在 AHB 数据宽度与存储器数据宽度不同时，XMC 针对外部存储器拥有的典型信号可做适度的处理，下表列出 XMC 支持的操作。

表 22-6 访问数据宽度与外部存储器数据宽度对照表

存储器	模式	AHB 数据宽度	存储器数据宽度	说明
SRAM	异步读写	8/16/32	8	1 次、分 2 次或 4 次 XMC 访问
	异步读写	8/16/32	16	使用字节信号 XMC_LB、XMC_UB、1 次或分 2 次 XMC 访问
NOR 闪存	异步读	8	16	
	异步读写	16	16	
	异步读写	32	16	分 2 次 XMC 访问
	同步读	16	16	
	同步读	32	16	分 2 次 XMC 访问
PSRAM	异步读	8	16	
	异步写	8	16	使用字节信号 XMC_LB、XMC_UB
	异步读写	16	16	
	异步读写	32	16	分 2 次 XMC 访问
	同步写	8	16	使用字节信号 XMC_LB、XMC_UB
	同步读写	16	16	
	同步读写	32	16	分 2 次 XMC 访问

## 22.4.2 访问模式

XMC 提供多种行为不同的访问模式，每种访问会依据时序参数动作，如表 22-7 所示，用户需依照外部存储器的规格与应用需求进行编程。

XMC 提供的访问模式有：

- 读写相同时序的模式：模式 1 与模式 2
- 读写不同时序的模式：模式 A、B、C 与 D
- 地址数据线复用的复用模式
- 有时钟的同步模式

表 22-7 NOR/PSRAM 参数寄存器

参数寄存器	意义	访问模式	单位
ADDRST	地址建立时间	1、2、A、B、C、D、复用	HCLK 周期
ADDRHT	地址保持时间	D、复用	HCLK 周期
DTST	数据建立时间	1、2、A、B、C、D、复用	HCLK 周期
DTLAT	数据延迟时间	同步	XMC_CLK 周期
CLKPSC	时钟分频系数	同步	HCLK 周期

时序控制除了时序参数寄存器外，若是开启等待使能位（NWASEN 或 NWSEN），XMC 会在数据建立其间检查 XMC\_NWAIT 信号，若是 XMC\_NWAIT 信号处在请求等待状态，XMC 便会等待 XMC\_NWAIT 回到就绪状态再进行数据传输。

### 22.4.2.1 读写相同时序的模式

模式 1 与模式 2 读与写的时序皆是参照 SRAM/NOR 闪存片选时序寄存器（XMC\_BK1TMG）的配置。

#### 模式 1

如表 22-8 与

表 22-9 配置，XMC 即会使用模式 1 访问外部存储器。读时序如图 22-3 所示，写时序如图 22-4 所示。

表 22-8 模式1的SRAM/NOR闪存片选控制寄存器 (XMC\_BK1CTRL) 配置

域	名称	配置方式
位 31: 20	保留	0x0
位 19	MWMC: 对存储器写操作位	0x0
位 18: 16	CRPGS: CRAM 页大小选择	0x0
位 15	NWASEN: 异步传输等待信号使能	根据存储器规格配置
位 14	RWTD: 读写时序不同控制	0x0
位 13	NWSEN: 同步传输等待信号使能	0x0
位 12	WEN: 写使能	根据需求配置
位 11	NWTCFG: 等待时序配置	0x0
位 10	WRAPEN: 支持非对齐的成组模式	0x0
位 9	NWPOL: 等待信号极性	根据存储器规格配置
位 8	SYNCBEN: 同步突发模式使能	0x0
位 7	保留	0x1
位 6	NOREN: NOR 闪存访问使能	0x0
位 5: 4	EXTMDBW: 外部存储器数据宽度	根据存储器规格配置
位 3: 2	DEV: 存储器类型	根据存储器规格配置, 除 0x2 (NOR 闪存) 外有效
位 1	ADMUXEN: 地址/数据复用使能	0x0
位 0	EN: 存储器块使能	0x1

表 22-9 模式1的SRAM/NOR闪存片选时序寄存器 (XMC\_BK1TMG) 配置

域	名称	配置方式
位 31: 30	保留	0x0
位 29: 28	ASYNCM: 异步访问模式选择	0x0
位 27: 24	DTLAT: 数据延迟时间	0x0
位 23: 20	CLKPSC: 时钟分频系数	0x0
位 19: 16	BUSLAT: 总线延迟时间	XMC_NE[x]由上升沿到下降沿的时间, 根据需求与存储器规格配置
位 15: 8	DTST: 数据建立时间	参照图 22-3 与图 22-4, 根据需求与存储器规格配置
位 7: 4	ADDRHT: 地址保持时间	0x0
位 3: 0	ADDRST: 地址建立时间	参照图 22-3 与图 22-4, 根据需求与存储器规格配置

图 22-3 NOR/PSRAM界面模式1读

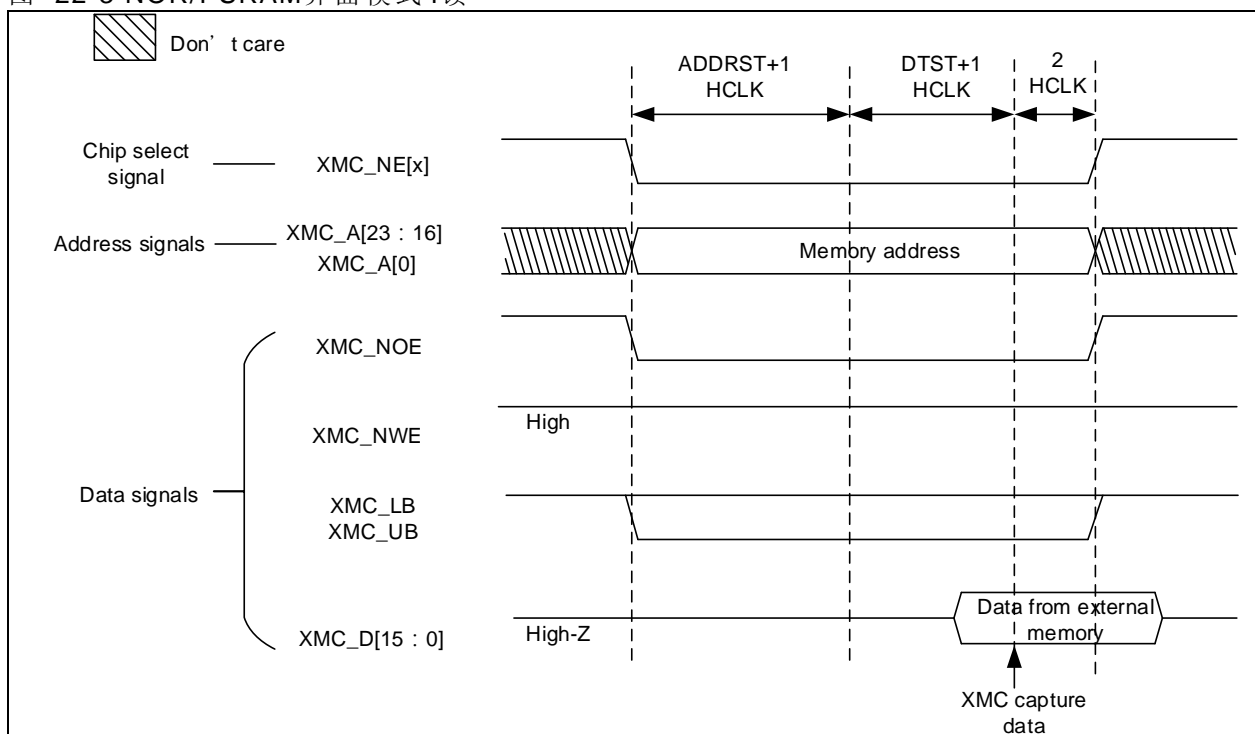
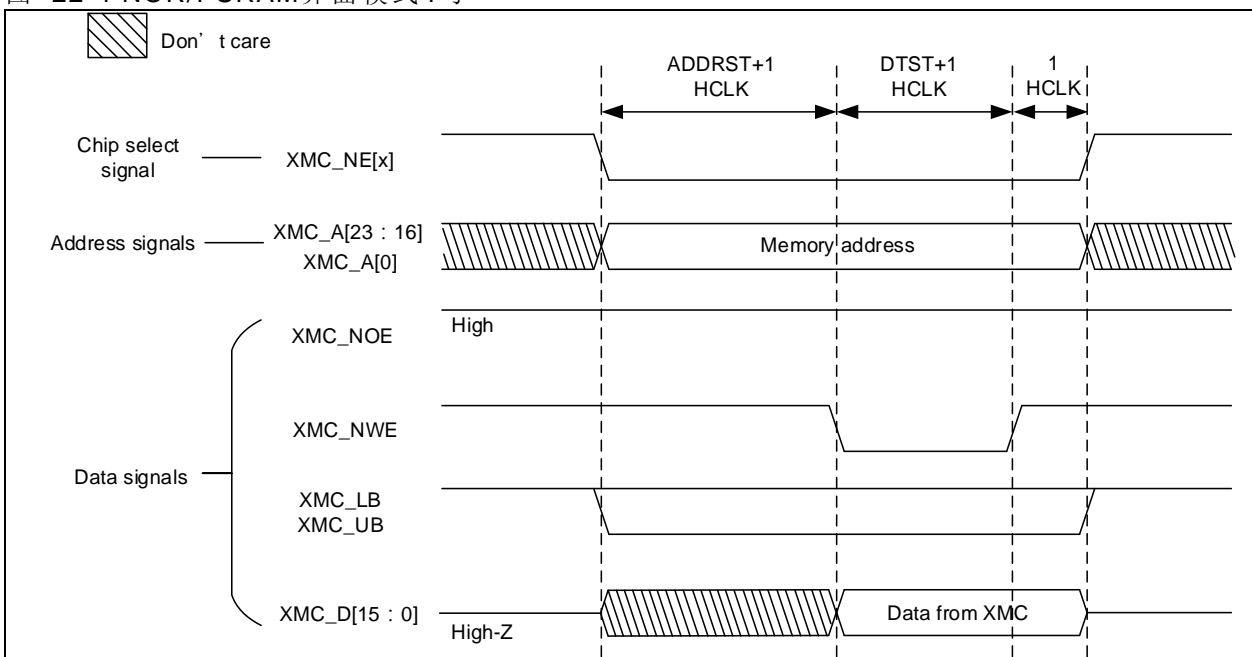


图 22-4 NOR/PSRAM界面模式1写



## 模式 2

如表 22-10与

表 22-11配置，XMC即会使用模式2访问外部存储器。读时序如

图 22-5所示，写时序如图 22-6所示。

表 22-10 模式2的SRAM/NOR闪存片选控制寄存器（XMC\_BK1CTRL）配置

域	名称	配置方式
位 31: 20	保留	0x0
位 19	MWMC: 对存储器写操作位	0x0
位 18: 16	CRPGS: CRAM 页大小选择	0x0
位 15	NWASEN: 异步传输等待信号使能	根据存储器规格配置
位 14	RWTD: 读写时序不同控制	0x0
位 13	NWSEN: 同步传输等待信号使能	0x0
位 12	WEN: 写使能	根据需求配置
位 11	NWTCFG: 等待时序配置	0x0
位 10	WRAPEN: 支持非对齐的成组模式	0x0
位 9	NWPOL: 等待信号极性	根据存储器规格配置
位 8	SYNCBEN: 同步突发模式使能	0x0
位 7	保留	0x1
位 6	NOREN: NOR 闪存访问使能	0x1
位 5: 4	EXTMDBW: 外部存储器数据宽度	根据存储器规格配置
位 3: 2	DEV: 存储器类型	0x2 (NOR 闪存)
位 1	ADMUXEN: 地址/数据复用使能	0x0
位 0	EN: 存储器块使能	0x1

表 22-11 模式2的SRAM/NOR闪存片选时序寄存器（XMC\_BK1TMG）配置

域	名称	配置方式
位 31: 30	保留	0x0
位 29: 28	ASYNM: 异步访问模式选择	0x0
位 27: 24	DTLAT: 数据延迟时间	0x0
位 23: 20	CLKPSC: 时钟分频系数	0x0
位 19: 16	BUSLAT: 总线延迟时间	XMC_NE[x]由上升沿到下降沿的时间，根据需求与存储器规格配置
位 15: 8	DTST: 数据建立时间	参照 图 22-5 与图 22-6，根据需求与存储器规格配置

位 7: 4	ADDRHT: 地址保持时间	0x0
位 3: 0	ADDRST: 地址建立时间	参照
图 22-5 与图 22-6, 根据需求与存储器规格配置		

图 22-5 NOR/PSRAM界面 模式2读

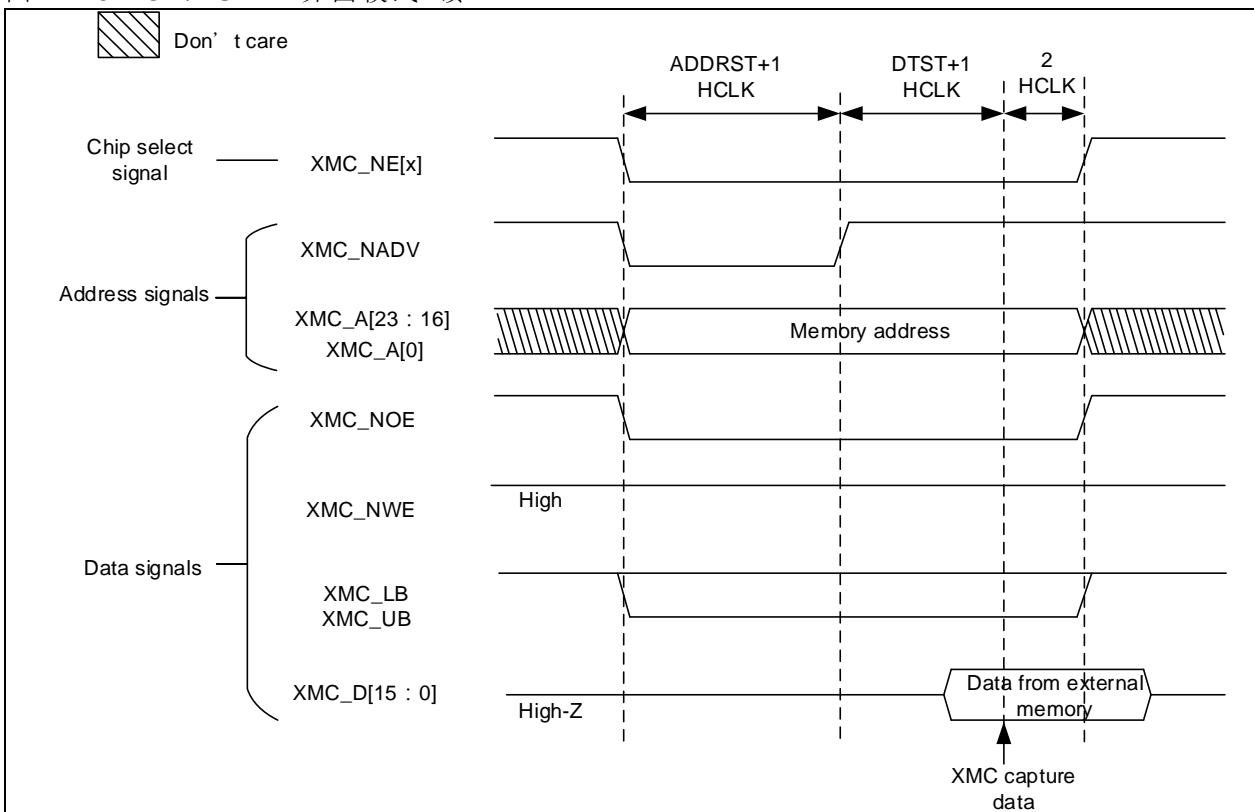
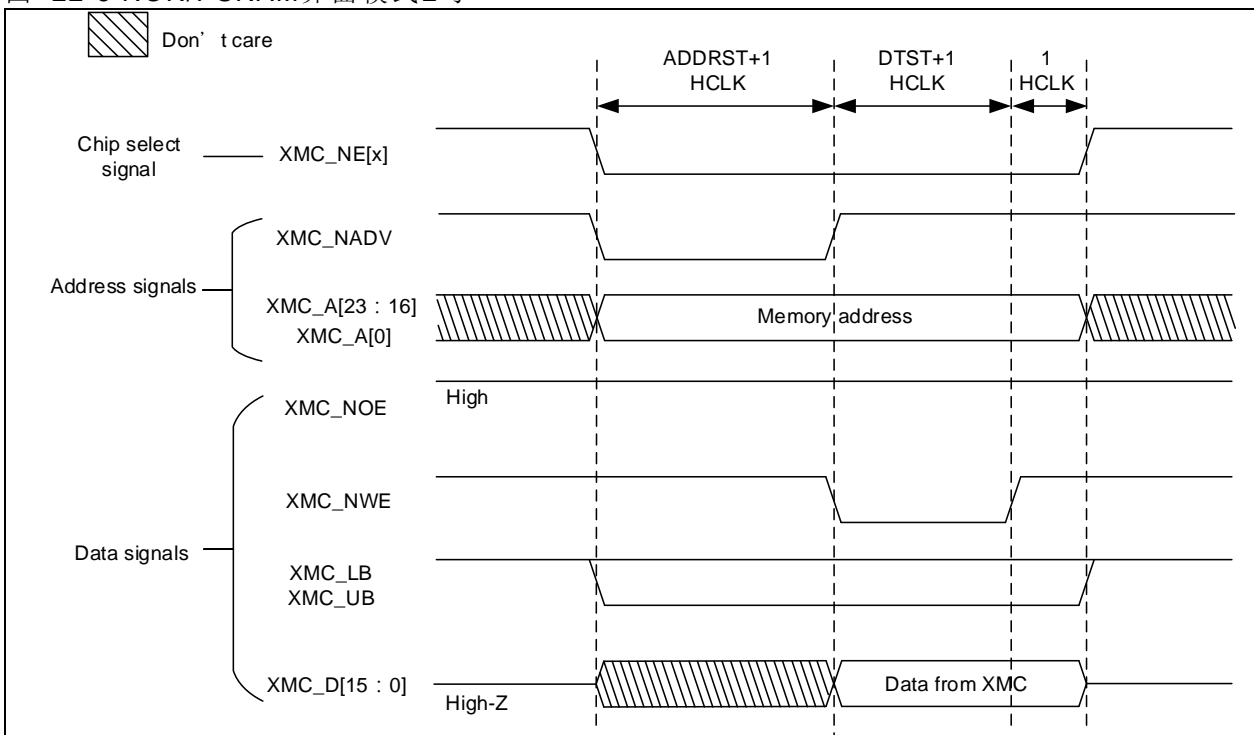


图 22-6 NOR/PSRAM界面 模式2写



## 22.4.2.2 读写不同时序的模式

模式 A、模式 B、模式 C 与模式 D 读时序是参照 SRAM/NOR 闪存片选时序寄存器 (XMC\_BK1TMG) 的配置, 写时序是参照 SRAM/NOR 闪存写时序寄存器 (XMC\_BK1TMGWR) 的配置。除此之外, 读与

写可混合搭配不同的模式。

#### 模式 A

如表 22-12、

表 22-13与

表 22-14配置，XMC即会使用模式A访问外部存储器。读时序如

图 22-7所示，写时序如图 22-8所示。

表 22-12 模式A的SRAM/NOR闪存片选控制寄存器（XMC\_BK1CTRL）配置

域	名称	配置方式
位 31: 20	保留	0x0
位 19	MWMC: 对存储器写操作位	0x0
位 18: 16	CRPGS: CRAM 页大小选择	0x0
位 15	NWASEN: 异步传输等待信号使能	根据存储器规格配置
位 14	RWTD: 读写时序不同控制	0x1
位 13	NWSEN: 同步传输等待信号使能	0x0
位 12	WEN: 写使能	根据需求配置
位 11	NWTCFG: 等待时序配置	0x0
位 10	WRAPEN: 支持非对齐的成组模式	0x0
位 9	NWPOL: 等待信号极性	根据存储器规格配置
位 8	SYNCBEN: 同步突发模式使能	0x0
位 7	保留	0x1
位 6	NOREN: NOR 闪存访问使能	0x0
位 5: 4	EXTMDBW: 外部存储器数据宽度	根据存储器规格配置
位 3: 2	DEV: 存储器类型	根据存储器规格配置, 除 0x2 (NOR 闪存) 有效
位 1	ADMUXEN: 地址/数据复用使能	0x0
位 0	EN: 存储器块使能	0x1

表 22-13 模式A的SRAM/NOR闪存片选时序寄存器（XMC\_BK1TMG）配置

域	名称	配置方式
位 31: 30	保留	0x0
位 29: 28	ASYNM: 异步访问模式选择	0x0 (模式 A)
位 27: 24	DTLAT: 数据延迟时间	0x0
位 23: 20	CLKPSC: 时钟分频系数	0x0
位 19: 16	BUSLAT: 总线延迟时间	XMC_NE[x]由上升沿到下降沿的时间, 根据需求与存储器规格配置
位 15: 8	DTST: 数据建立时间	参照 图 22-7, 根据需求与存储器规格配置
位 7: 4	ADDRHT: 地址保持时间	0x0
位 3: 0	ADDRST: 地址建立时间	参照 图 22-7, 根据需求与存储器规格配置

表 22-14 模式A的SRAM/NOR闪存写时序寄存器（XMC\_BK1TMGWR）配置

域	名称	配置方式
位 31: 30	保留	0x0
位 29: 28	ASYNM: 异步访问模式选择	0x0 (模式 A)
位 27: 20	保留	0x0
位 19: 16	BUSLAT: 总线延迟时间	XMC_NE[x]由上升沿到下降沿的时间, 根据需求与存储器规格配置
位 15: 8	DTST: 数据建立时间	参照图 22-8, 根据需求与存储器规格配置
位 7: 4	ADDRHT: 地址保持时间	0x0
位 3: 0	ADDRST: 地址建立时间	参照图 22-8, 根据需求与存储器规格配置

图 22-7 NOR/PSRAM界面模式A读

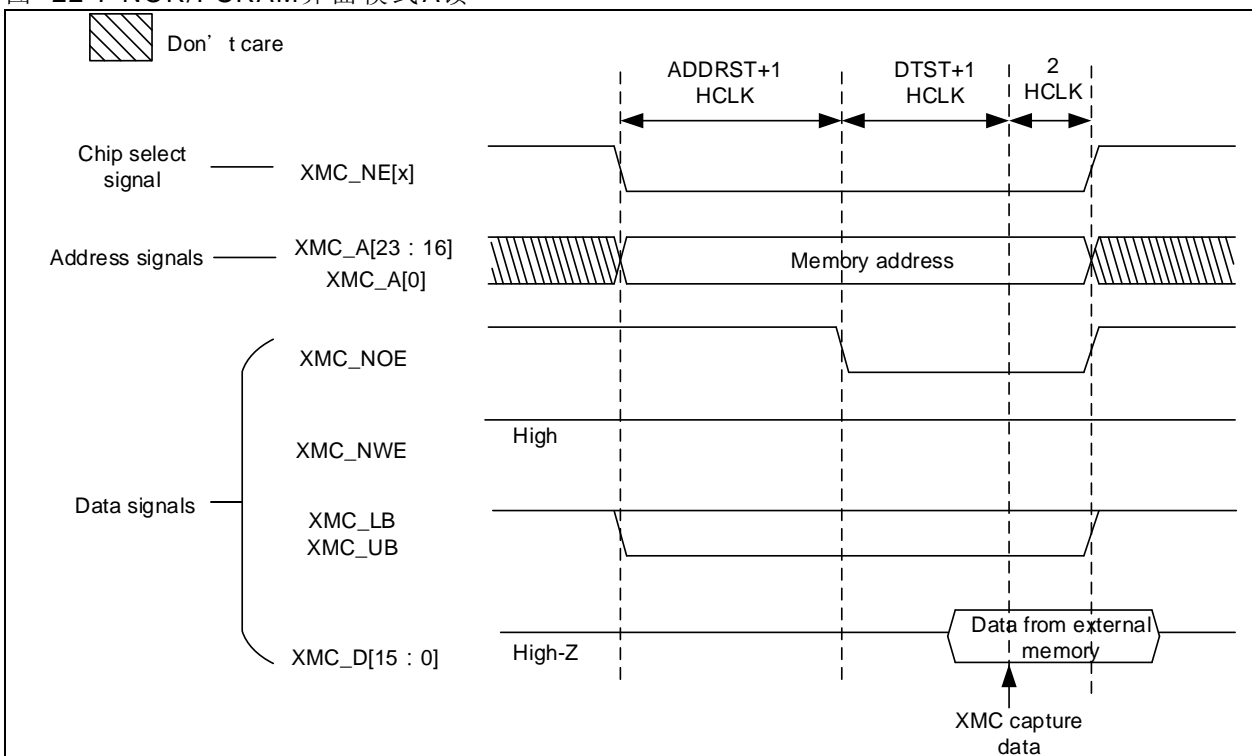
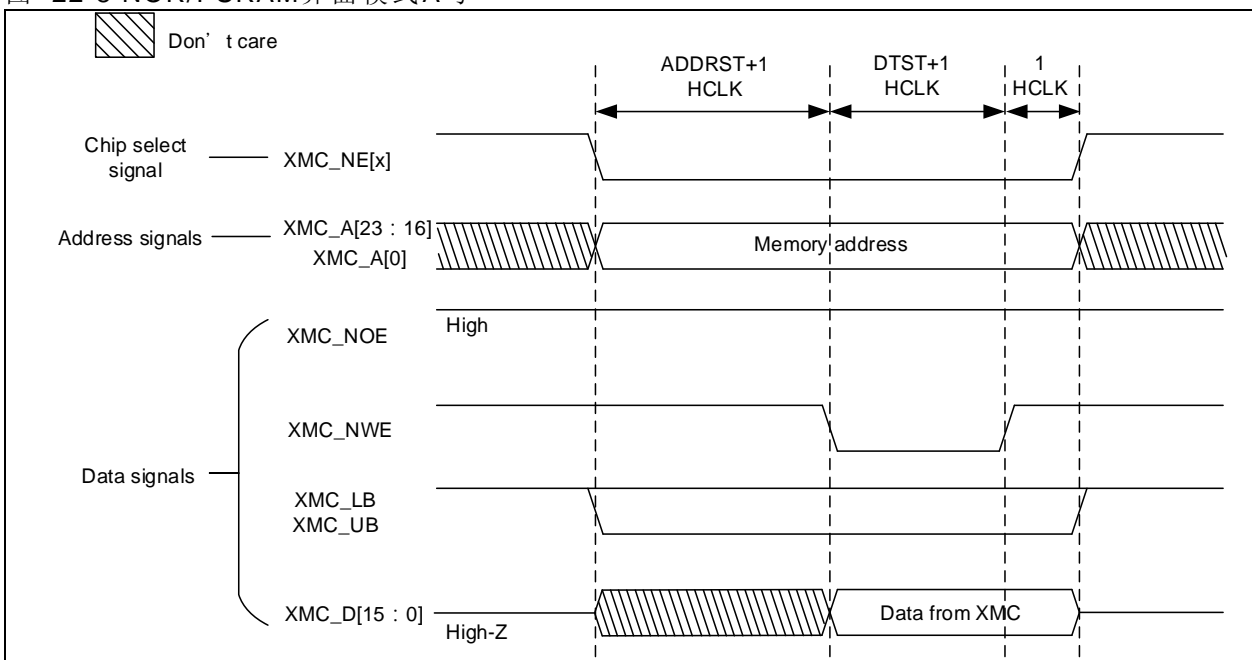


图 22-8 NOR/PSRAM界面模式A写





**模式 B**

如表 22-15、

表 22-16与

表 22-17配置，XMC即会使用模式B访问外部存储器。读时序如

图 22-9所示，写时序如图 22-10所示。

表 22-15 模式B的SRAM/NOR闪存片选控制寄存器（XMC\_BK1CTRL）配置

域	名称	配置方式
位 31: 20	保留	0x0
位 19	MWMC: 对存储器写操作位	0x0
位 18: 16	CRPGS: CRAM 页大小选择	0x0
位 15	NWASEN: 异步传输等待信号使能	根据存储器规格配置
位 14	RWTD: 读写时序不同控制	0x1
位 13	NWSEN: 同步传输等待信号使能	0x0
位 12	WEN: 写使能	根据需求配置
位 11	NWTCFG: 等待时序配置	0x0
位 10	WRAPEN: 支持非对齐的成组模式	0x0
位 9	NWPOL: 等待信号极性	根据存储器规格配置
位 8	SYNCBEN: 同步突发模式使能	0x0
位 7	保留	0x1
位 6	NOREN: NOR 闪存访问使能	0x1
位 5: 4	EXTMDBW: 外部存储器数据宽度	根据存储器规格配置
位 3: 2	DEV: 存储器类型	0x2 (NOR 闪存)
位 1	ADMUXEN: 地址/数据复用使能	0x0
位 0	EN: 存储器块使能	0x1

表 22-16 模式B的SRAM/NOR闪存片选时序寄存器（XMC\_BK1TMG）配置

域	名称	配置方式
位 31: 30	保留	0x0
位 29: 28	ASYNCM: 异步访问模式选择	0x1 (模式 B)
位 27: 24	DTLAT: 数据延迟时间	0x0
位 23: 20	CLKPSC: 时钟分频系数	0x0
位 19: 16	BUSLAT: 总线延迟时间	XMC_NE[x]由上升沿到下降沿的时间，根据需求与存储器规格配置
位 15: 8	DTST: 数据建立时间	参照图 22-9, 根据需求与存储器规格配置
位 7: 4	ADDRHT: 地址保持时间	0x0
位 3: 0	ADDRST: 地址建立时间	参照图 22-9, 根据需求与存储器规格配置

表 22-17 模式B的SRAM/NOR闪存写时序寄存器（XMC\_BK1TMGWR）配置

域	名称	配置方式
位 31: 30	保留	0x0
位 29: 28	ASYNCM: 异步访问模式选择	0x1 (模式 B)
位 27: 20	保留	0x0
位 19: 16	BUSLAT: 总线延迟时间	XMC_NE[x]由上升沿到下降沿的时间，根据需求与存储器规格配置
位 15: 8	DTST: 数据建立时间	参照图 22-10, 根据需求与存储器规格配置
位 7: 4	ADDRHT: 地址保持时间	0x0
位 3: 0	ADDRST: 地址建立时间	参照图 22-10, 根据需求与存储器规格配置

图 22-9 NOR/PSRAM界面模式B读

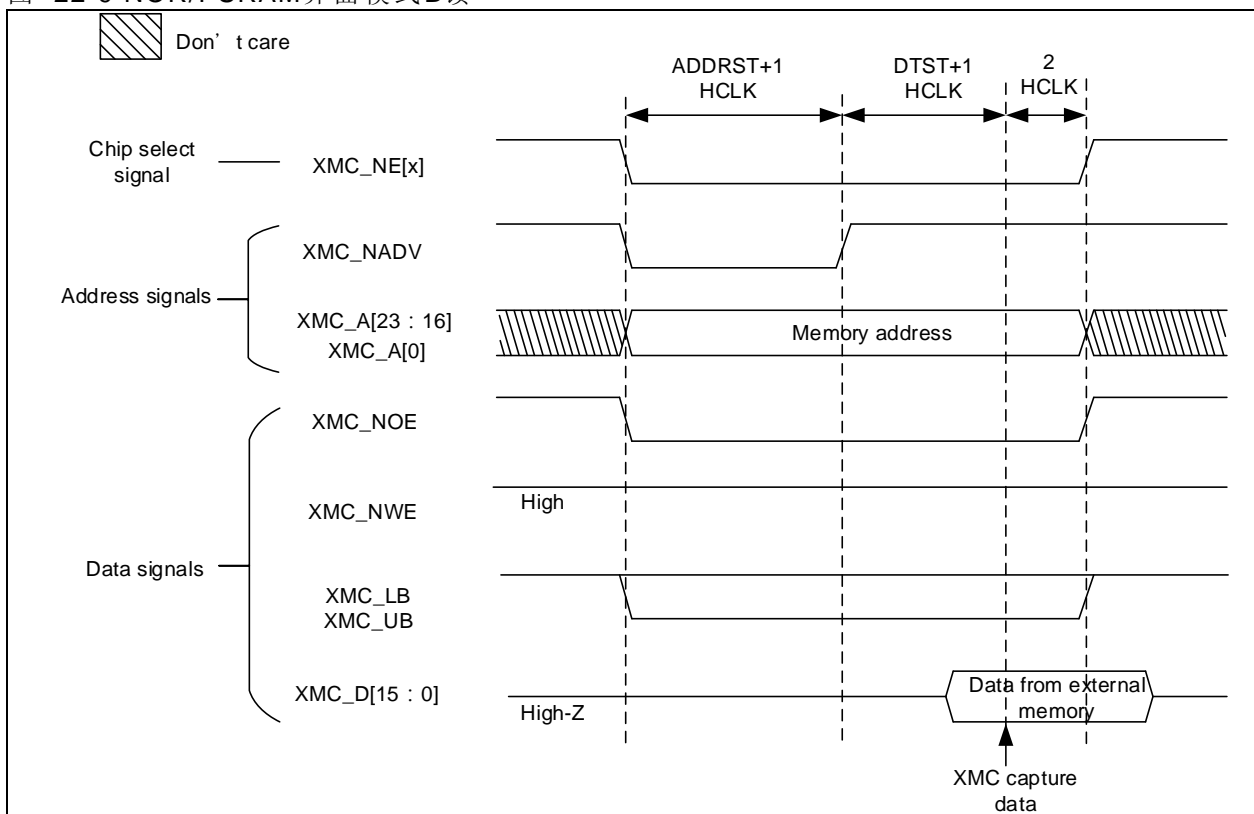
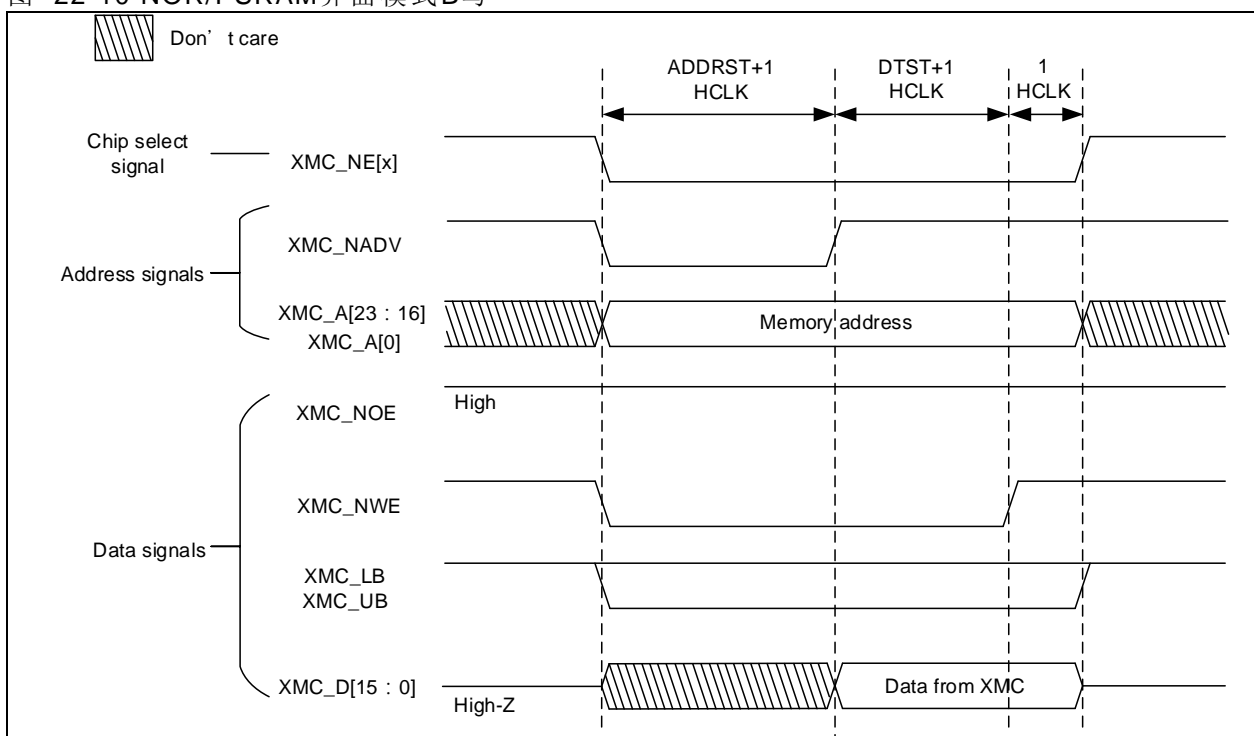


图 22-10 NOR/PSRAM界面模式B写



**模式 C**

如表 22-18、

表 22-19与

表 22-20配置，XMC即会使用模式C访问外部存储器。读时序如

图 22-11所示，写时序如图 22-12所示。

表 22-18 模式C的SRAM/NOR闪存片选控制寄存器（XMC\_BK1CTRL）配置

域	名称	配置方式
位 31: 20	保留	0x0
位 19	MWMC: 对存储器写操作位	0x0
位 18: 16	CRPGS: CRAM 页大小选择	0x0
位 15	NWASEN: 异步传输等待信号使能	根据存储器规格配置
位 14	RWTD: 读写时序不同控制	0x1
位 13	NWSEN: 同步传输等待信号使能	0x0
位 12	WEN: 写使能	根据需求配置
位 11	NWTCFG: 等待时序配置	0x0
位 10	WRAPEN: 支持非对齐的成组模式	0x0
位 9	NWPOL: 等待信号极性	根据存储器规格配置
位 8	SYNCBEN: 同步突发模式使能	0x0
位 7	保留	0x1
位 6	NOREN: NOR 闪存访问使能	0x1
位 5: 4	EXTMDBW: 外部存储器数据宽度	根据存储器规格配置
位 3: 2	DEV: 存储器类型	0x2 (NOR 闪存)
位 1	ADMUXEN: 地址/数据复用使能	0x0
位 0	EN: 存储器块使能	0x1

表 22-19 模式C的SRAM/NOR闪存片选时序寄存器（XMC\_BK1TMG）配置

域	名称	配置方式
位 31: 30	保留	0x0
位 29: 28	ASYNCM: 异步访问模式选择	0x2 (模式 C)
位 27: 24	DTLAT: 数据延迟时间	0x0
位 23: 20	CLKPSC: 时钟分频系数	0x0
位 19: 16	BUSLAT: 总线延迟时间	XMC_NE[x]由上升沿到下降沿的时间，根据需求与存储器规格配置
位 15: 8	DTST: 数据建立时间	参照图 22-11，根据需求与存储器规格配置
位 7: 4	ADDRHT: 地址保持时间	0x0
位 3: 0	ADDRST: 地址建立时间	参照图 22-11，根据需求与存储器规格配置

表 22-20 模式C的SRAM/NOR闪存写时序寄存器（XMC\_BK1TMGWR）配置

域	名称	配置方式
位 31: 30	保留	0x0
位 29: 28	ASYNCM: 异步访问模式选择	0x2 (模式 C)
位 27: 20	保留	0x0
位 19: 16	BUSLAT: 总线延迟时间	XMC_NE[x]由上升沿到下降沿的时间，根据需求与存储器规格配置
位 15: 8	DTST: 数据建立时间	参照图 22-12，根据需求与存储器规格配置
位 7: 4	ADDRHT: 地址保持时间	0x0
位 3: 0	ADDRST: 地址建立时间	参照图 22-12，根据需求与存储器规格配置

图 22-11 NOR/PSRAM界面模式C读

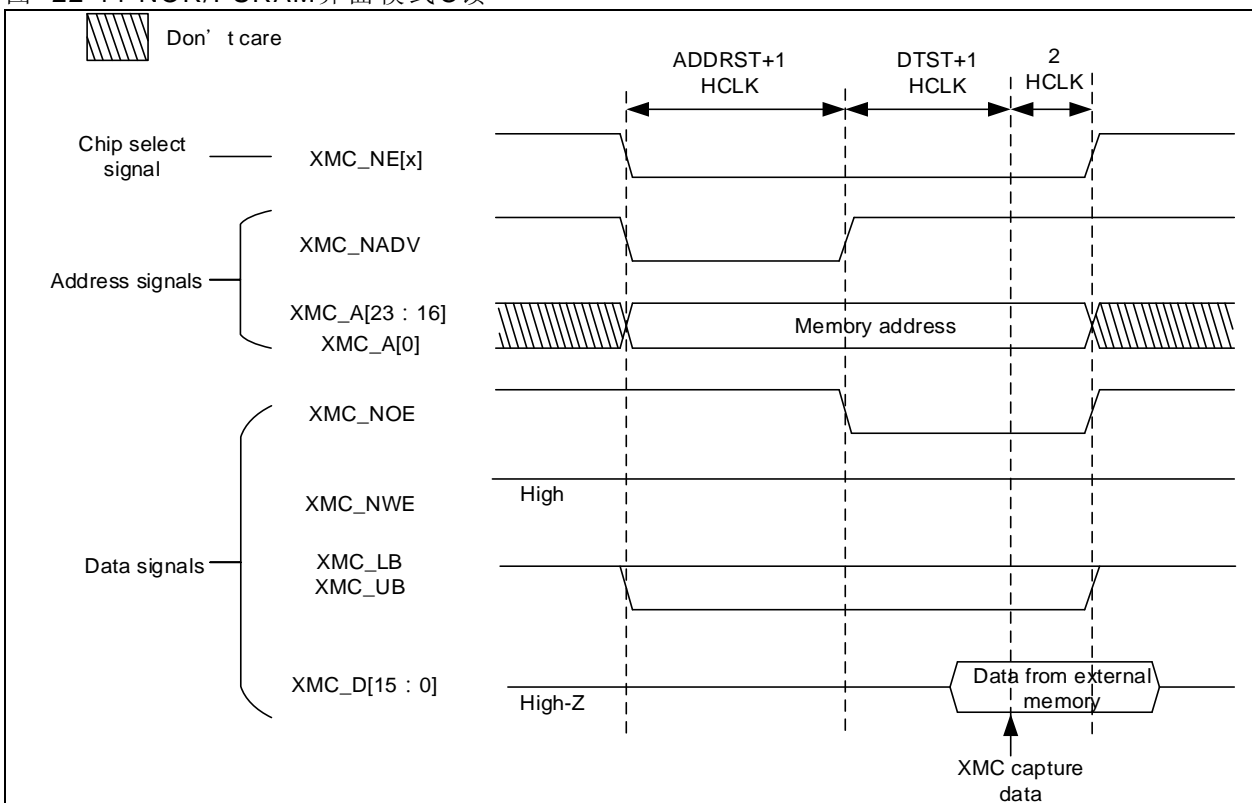
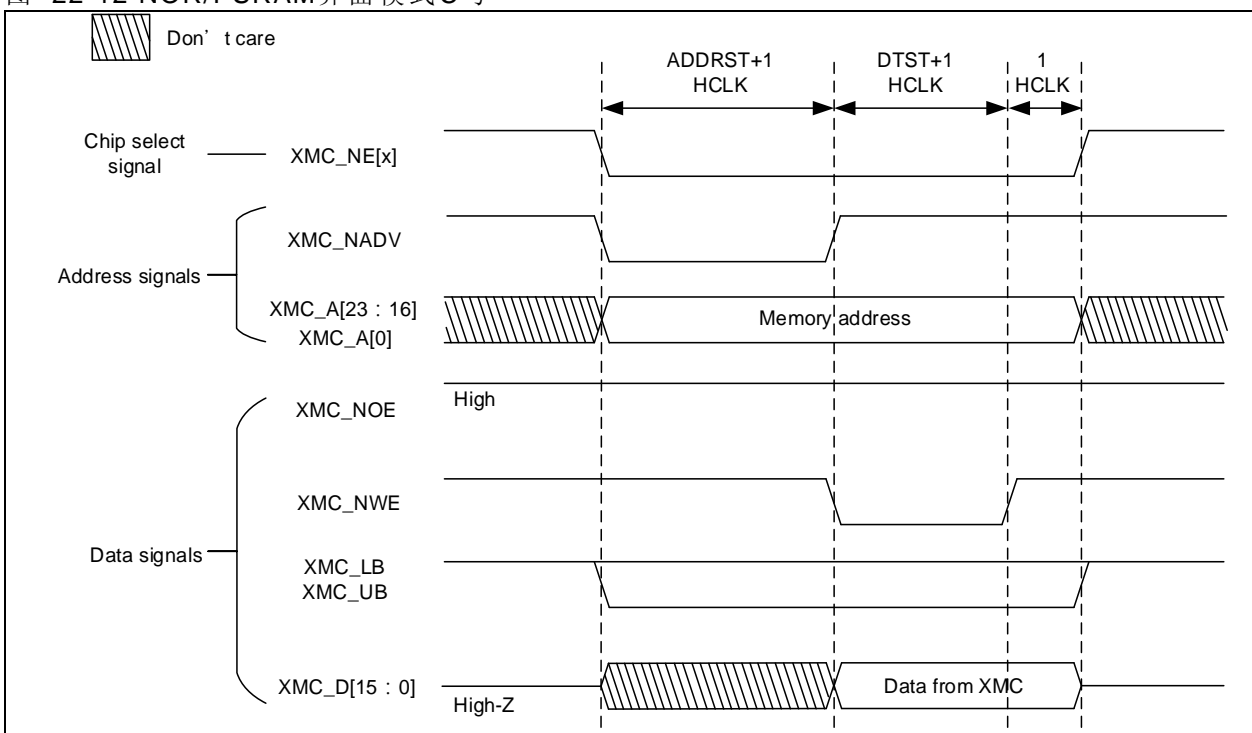


图 22-12 NOR/PSRAM界面模式C写



**模式 D**

如表 22-21、

表 22-22与

表 22-23配置，XMC即会使用模式D访问外部存储器。读时序如

图 22-13所示，写时序如图 22-14所示。

表 22-21 模式D的SRAM/NOR闪存片选控制寄存器（XMC\_BK1CTRL）配置

域	名称	配置方式
位 31: 20	保留	0x0
位 19	MWMC: 对存储器写操作位	0x0
位 18: 16	CRPGS: CRAM 页大小选择	0x0
位 15	NWASEN: 异步传输等待信号使能	根据存储器规格配置
位 14	RWTD: 读写时序不同控制	0x1
位 13	NWSEN: 同步传输等待信号使能	0x0
位 12	WEN: 写使能	根据需求配置
位 11	NWTCFG: 等待时序配置	0x0
位 10	WRAPEN: 支持非对齐的成组模式	0x0
位 9	NWPOL: 等待信号极性	根据存储器规格配置
位 8	SYNCBEN: 同步突发模式使能	0x0
位 7	保留	0x1
位 6	NOREN: NOR 闪存访问使能	根据存储器规格配置
位 5: 4	EXTMDBW: 外部存储器数据宽度	根据存储器规格配置
位 3: 2	DEV: 存储器类型	根据存储器规格配置
位 1	ADMUXEN: 地址/数据复用使能	0x0
位 0	EN: 存储器块使能	0x1

表 22-22 模式D的SRAM/NOR闪存片选时序寄存器（XMC\_BK1TMG）配置

域	名称	配置方式
位 31: 30	保留	0x0
位 29: 28	ASYNCM: 异步访问模式选择	0x3（模式 D）
位 27: 24	DTLAT: 数据延迟时间	0x0
位 23: 20	CLKPSC: 时钟分频系数	0x0
位 19: 16	BUSLAT: 总线延迟时间	XMC_NE[x]由上升沿到下降沿的时间，根据需求与存储器规格配置
位 15: 8	DTST: 数据建立时间	参照图 22-13，根据需求与存储器规格配置
位 7: 4	ADDRHT: 地址保持时间	参照图 22-13，根据需求与存储器规格配置
位 3: 0	ADDRST: 地址建立时间	参照图 22-13，根据需求与存储器规格配置

表 22-23 模式D的SRAM/NOR闪存写时序寄存器（XMC\_BK1TMGWR）配置

域	名称	配置方式
位 31: 30	保留	0x0
位 29: 28	ASYNCM: 异步访问模式选择	0x3（模式 D）
位 27: 20	保留	0x0
位 19: 16	BUSLAT: 总线延迟时间	XMC_NE[x]由上升沿到下降沿的时间，根据需求与存储器规格配置
位 15: 8	DTST: 数据建立时间	参照图 22-14，根据需求与存储器规格配置
位 7: 4	ADDRHT: 地址保持时间	参照图 22-14，根据需求与存储器规格配置
位 3: 0	ADDRST: 地址建立时间	参照图 22-14，根据需求与存储器规格配置

图 22-13 NOR/PSRAM界面模式D读

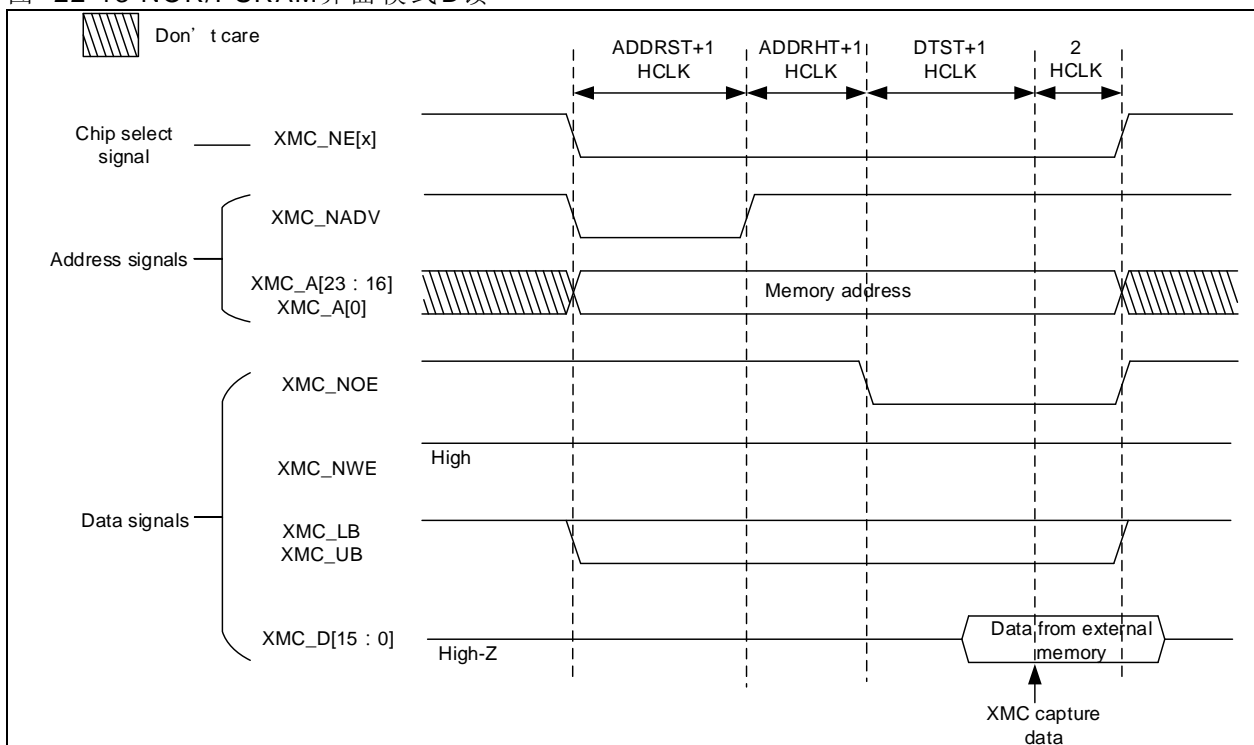
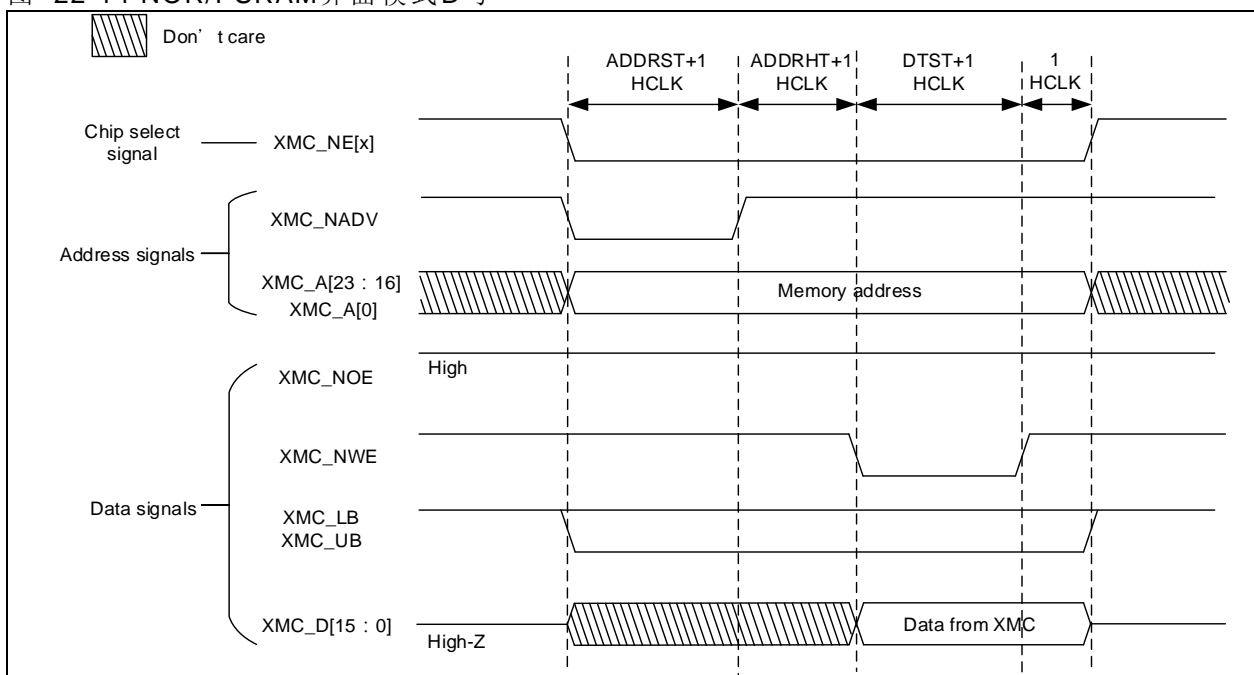


图 22-14 NOR/PSRAM界面模式D写



## 22.4.2.3 复用模式

如表 22-24 与表 22-25 配置，XMC 即会使用复用模式访问外部存储器。读时序如图 22-15 所示，写时序如图 22-16 所示。

表 22-24 复用模式的SRAM/NOR闪存片选控制寄存器（XMC\_BK1CTRL）配置

域	名称	配置方式
位 31: 20	保留	0x0
位 19	MWMC: 对存储器写操作位	0x0
位 18: 16	CRPGS: CRAM 页大小选择	0x0
位 15	NWASEN: 异步传输等待信号使能	根据存储器规格配置
位 14	RWTD: 读写时序不同控制	0x0
位 13	NWSEN: 同步传输等待信号使能	0x0
位 12	WEN: 写使能	根据需求配置
位 11	NWTCFG: 等待时序配置	0x0
位 10	WRAPEN: 支持非对齐的成组模式	0x0
位 9	NWPOL: 等待信号极性	根据存储器规格配置
位 8	SYNCBEN: 同步突发模式使能	0x0
位 7	保留	0x1
位 6	NOREN: NOR 闪存访问使能	根据存储器规格配置
位 5: 4	EXTMDBW: 外部存储器数据宽度	根据存储器规格配置
位 3: 2	DEV: 存储器类型	根据存储器规格配置，除 0x0（SRAM）外有效
位 1	ADMUXEN: 地址/数据复用使能	0x1
位 0	EN: 存储器块使能	0x1

表 22-25 复用模式的SRAM/NOR闪存片选时序寄存器（XMC\_BK1TMG）配置

域	名称	配置方式
位 31: 30	保留	0x0
位 29: 28	ASYNCM: 异步访问模式选择	0x0
位 27: 24	DTLAT: 数据延迟时间	0x0
位 23: 20	CLKPSC: 时钟分频系数	0x0
位 19: 16	BUSLAT: 总线延迟时间	XMC_NE[x]由上升沿到下降沿的时间，根据需求与存储器规格配置
位 15: 8	DTST: 数据建立时间	参照图 22-15 与图 22-16，根据需求与存储器规格配置
位 7: 4	ADDRHT: 地址保持时间	参照图 22-15 与图 22-16，根据需求与存储器规格配置
位 3: 0	ADDRST: 地址建立时间	参照图 22-15 与图 22-16，根据需求与存储器规格配置



图 22-15 NOR/PSRAM界面复用模式读

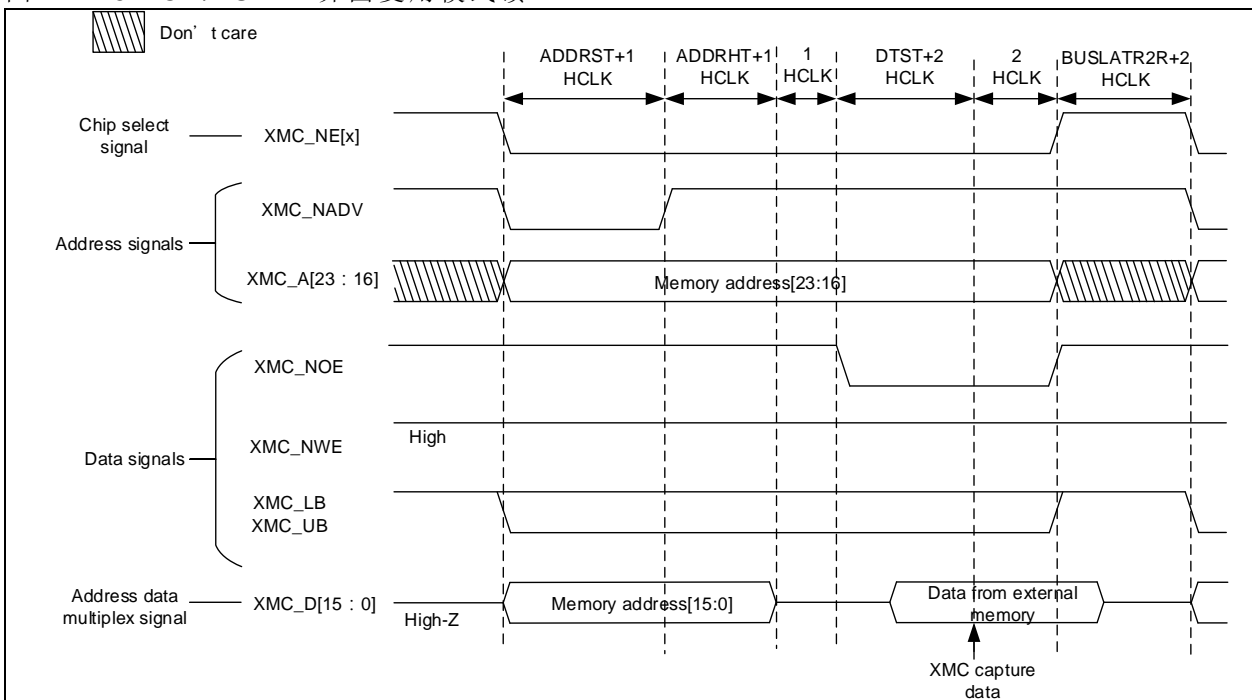
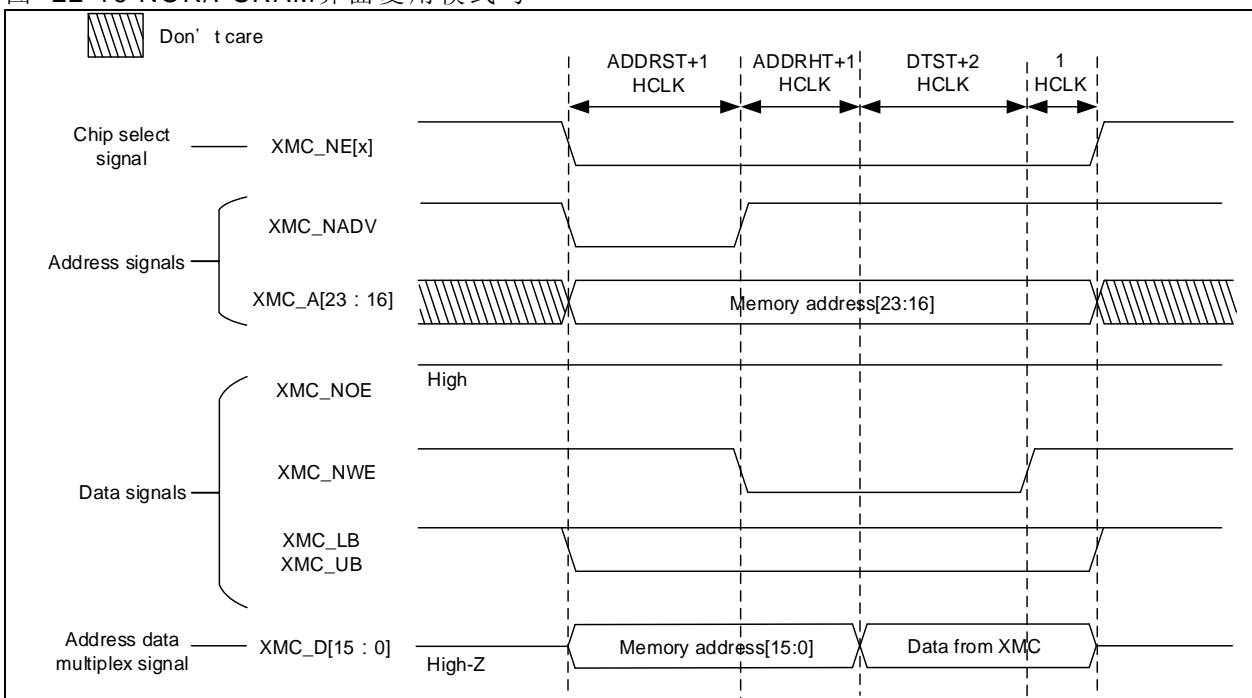


图 22-16 NOR/PSRAM界面复用模式写



## 22.4.2.4 同步模式

如表 22-26与

表 22-27配置，XMC即会使用同步模式访问外部存储器。

若存储器在地址锁存与数据传输之间插入 XMC\_NWAIT 信号，XMC 除了等待 DTLAT+1 个 XMC\_CLK 外，也会根据 XMC\_NWAIT 进行等待。在数据传输途中，XMC 会根据 NWTCFG 的配置在 XMC\_NWAIT 信号的下一个周期等待或是当个周期等待。

读时序如图 22-17 所示，写时序如图 22-18 所示。图 22-17 与图 22-18 皆是 XMC\_NWAIT 信号的下一个周期等待（NWTCFG=0）做示范。

表 22-26 同步模式的SRAM/NOR闪存片选控制寄存器（XMC\_BK1CTRL）配置

域	名称	配置方式
位 31: 20	保留	0x0
位 19	MWMC: 对存储器写操作位	0x1
位 18: 16	CRPGS: CRAM 页大小选择	根据存储器规格配置
位 15	NWASEN: 异步传输等待信号使能	0x0
位 14	RWTD: 读写时序不同控制	0x0
位 13	NWSEN: 同步传输等待信号使能	根据存储器规格配置
位 12	WEN: 写使能	根据需求配置
位 11	NWTCFG: 等待时序配置	根据存储器规格配置
位 10	WRAPEN: 支持非对齐的成组模式	根据需求配置
位 9	NWPOL: 等待信号极性	根据存储器规格配置
位 8	SYNCBEN: 同步突发模式使能	0x1
位 7	保留	0x1
位 6	NOREN: NOR 闪存访问使能	同步写: 0x0 同步读: 根据存储器规格配置
位 5: 4	EXTMDBW: 外部存储器数据宽度	根据存储器规格配置
位 3: 2	DEV: 存储器类型	同步写: 0x1 同步读: 根据存储器规格配置, 除 0x0 (SRAM) 外有效
位 1	ADMUXEN: 地址/数据复用使能	根据需求配置
位 0	EN: 存储器块使能	0x1

表 22-27 同步模式的SRAM/NOR闪存片选时序寄存器（XMC\_BK1TMG）配置

域	名称	配置方式
位 31: 30	保留	0x0
位 29: 28	ASYNCM: 异步访问模式选择	0x0
位 27: 24	DTLAT: 数据延迟时间	参照图 22-17 与图 22-18, 根据需求与存储器规格配置
位 23: 20	CLKPSC: 时钟分频系数	XMC_CLK 周期为 HCLK 周期*CLKPSC。参照图 22-17 与图 22-18, 根据需求与存储器规格配置
位 19: 16	BUSLAT: 总线延迟时间	XMC_NE[x]由上升沿到下降沿的时间, 根据需求与存储器规格配置
位 15: 8	DTST: 数据建立时间	0x0
位 7: 4	ADDRHT: 地址保持时间	0x0
位 3: 0	ADDRST: 地址建立时间	0x0

图 22-17 NOR/PSRAM界面同步模式复用读

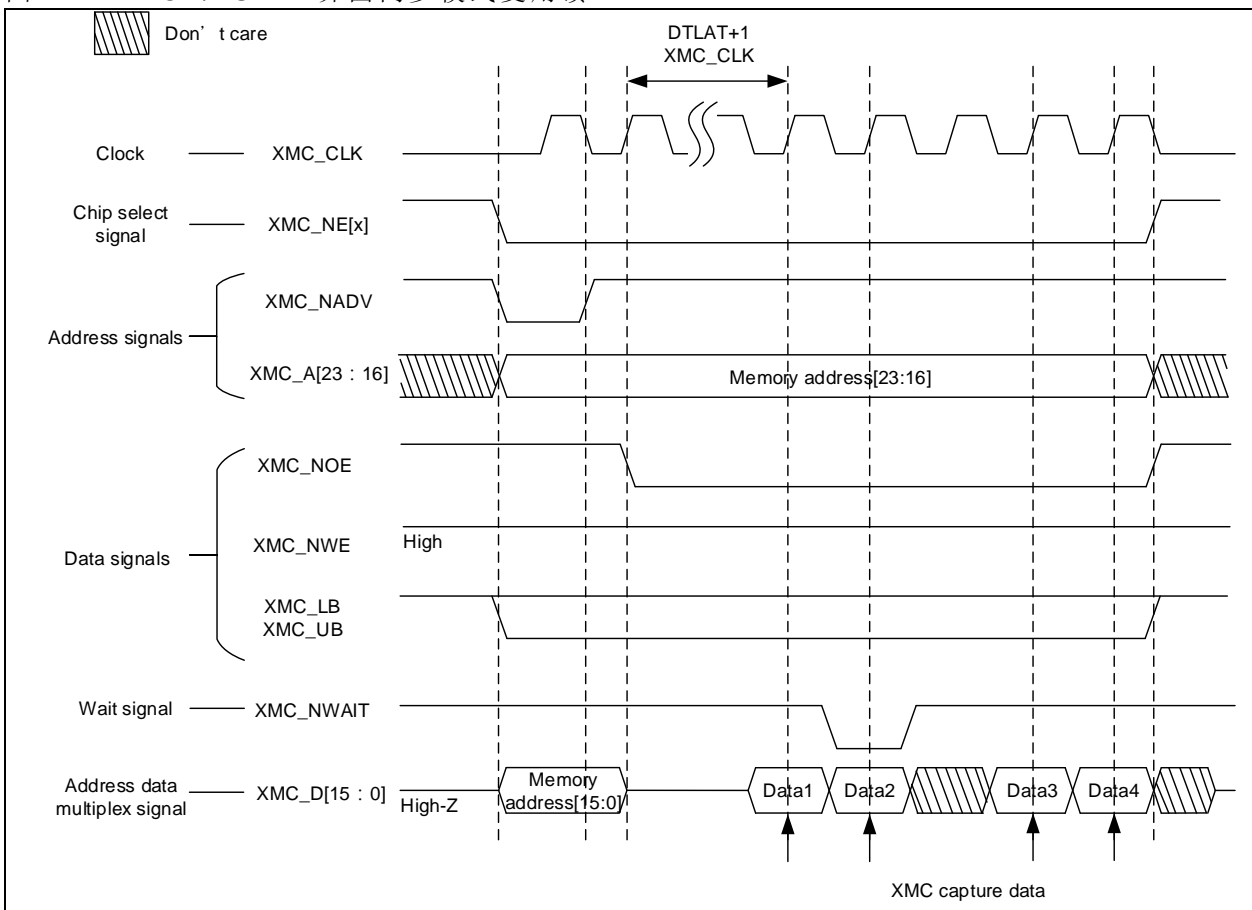
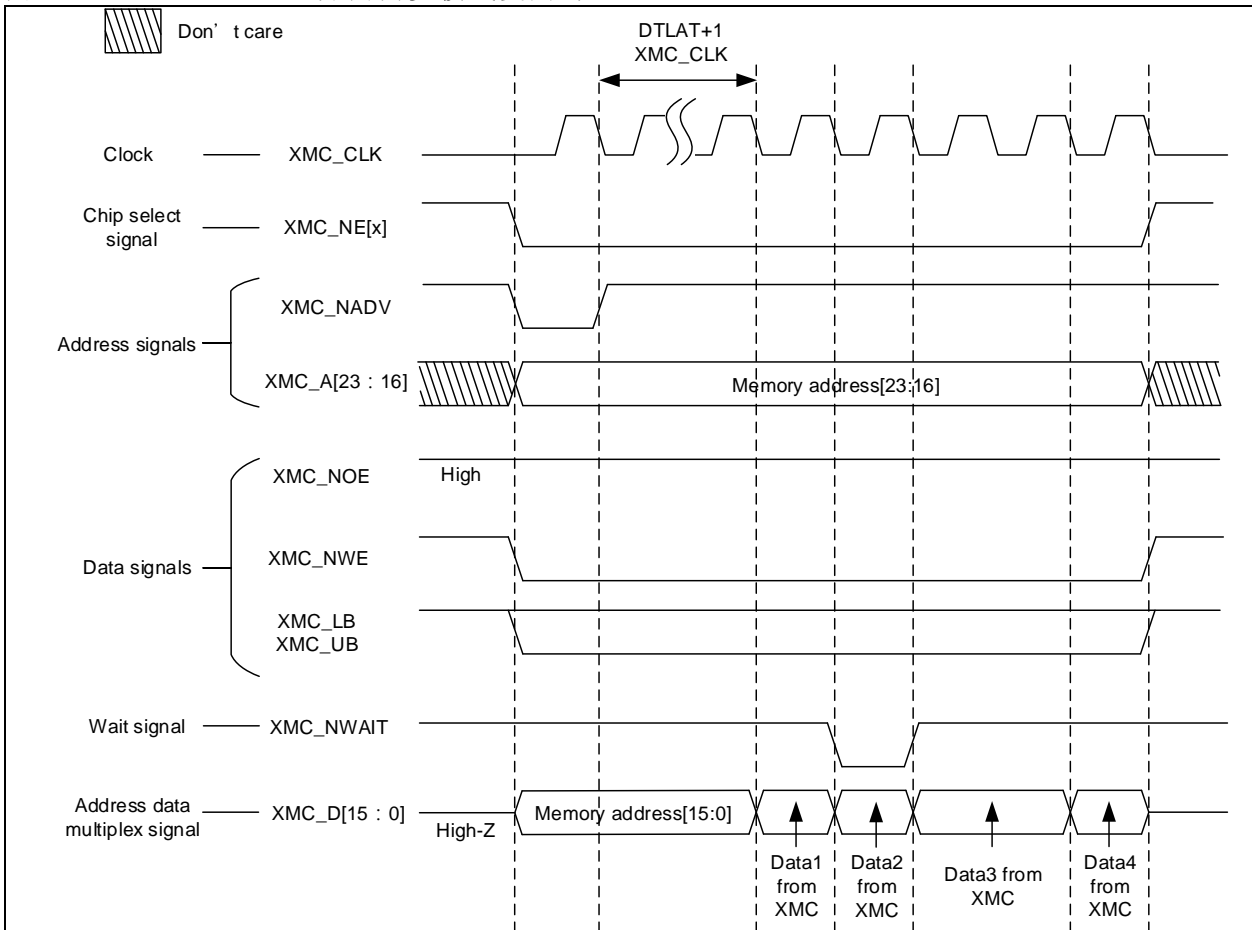


图 22-18 NOR/PSRAM界面同步模式复用写



## 22.5 NAND界面

NAND 界面可驱动 NAND 闪存，存储区块分为两个空间区块：常规空间与特殊空间，各自有分开的时序寄存器，可使用不同的时序访问这两个空间区块。

### 22.5.1 操作方式

#### 管脚使用

不同的外部存储器所需的信号不同，表 22-28 列出了典型信号。

表 22-28 NAND 闪存典型管脚信号

XMC 管脚信号	8 位 NAND 闪存	16 位 NAND 闪存
XMC_NCE[2]	片选信号	片选
XMC_A[17]	地址锁存 (ALE) 信号	地址锁存 (ALE) 信号
XMC_A[16]	命令锁存 (CLE) 信号	命令锁存 (CLE) 信号
XMC_NOE	输出使能 (NRE) 信号	输出使能 (NRE) 信号
XMC_NWE	写使能信号	写使能信号
XMC_D[15: 0]	数据总线	未使用 XMC_D[15: 8] XMC_D[7: 0]为数据总线
XMC_NWAIT	就绪/忙碌 (R/B) 信号	就绪/忙碌 (R/B) 信号

#### 访问地址

HADDR 只用来选择存储区块，存储区块规则可参照表 22-3，用户于命令区写入命令，于地址区写入目标地址，于数据区进行数据的读写。由于访问地址是以数据总线传输的，HADDR 实际上不与 NAND 容量有关联，因此理论上 XMC 不限制可以访问的 NAND 闪存容量。

#### 访问数据

在 AHB 数据宽度与存储器数据宽度不同时，XMC 针对外部存储器拥有的典型信号可做适度的处理，表 22-29 列出 XMC 支持的操作。

表 22-29 访问数据宽度与 NAND 闪存数据宽度对照表

存储器	模式	AHB 数据宽度	存储器数据宽度	说明
8 位 NAND 闪存	读写	8	8	
	读写	16	8	分 2 次 XMC 访问
	读写	32	8	分 4 次 XMC 访问
16 位 NAND 闪存	读	8	16	
	读写	16	16	
	读写	32	16	分 2 次 XMC 访问

### 22.5.2 访问时序

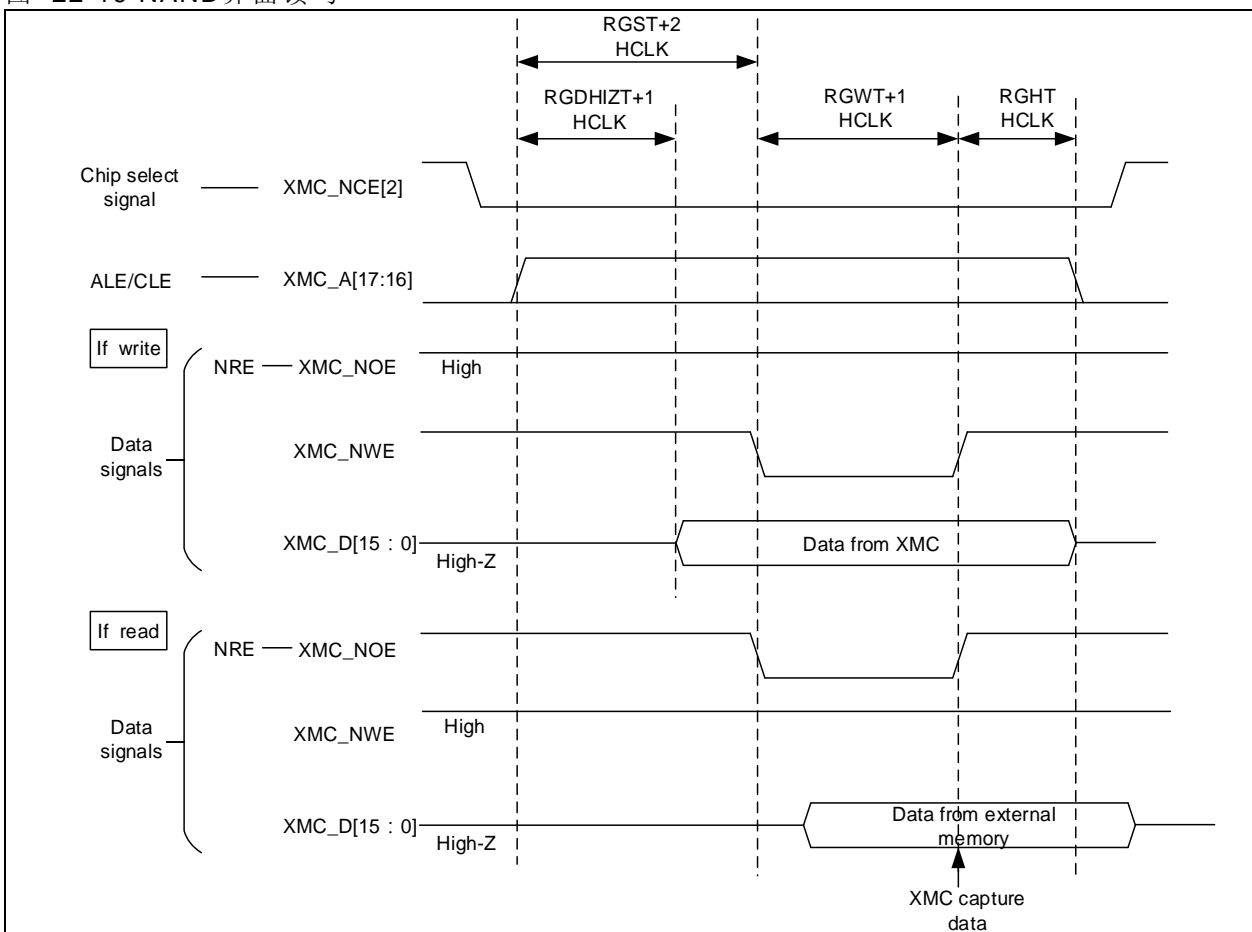
XMC 对 NAND 闪存的访问会依据时序参数动作，如表 22-30 与

图 22-19 所示，用户需依照 NAND 闪存的规格与应用需求进行编程。

表 22-30 NAND 参数寄存器

参数寄存器	意义	访问模式	单位
RGDHIZT/SPDHIZT	存储器数据总线高阻时间	写	HCLK 周期
RGST/SPST	存储器建立时间	读写	HCLK 周期
RGWT/SPWT	存储器等待时间	读写	HCLK 周期
RGHT/SPHT	存储器保持时间	读写	HCLK 周期

图 22-19 NAND界面读写

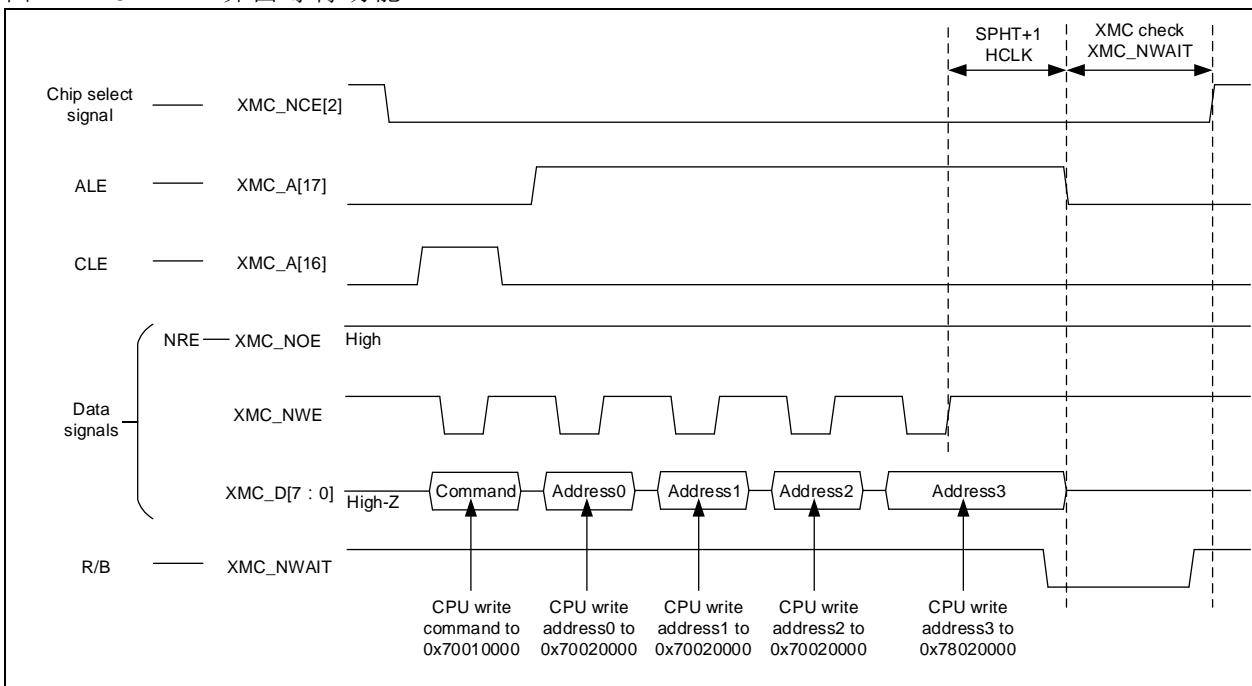


当 **NWEN** 位被使能，XMC 会在存储器保持时间结束时检查 **XMC\_NWAIT** 信号是否被拉低，若是 **XMC\_NWAIT** 信号为低，XMC 会持续将 **XMC\_NCE[2]** 拉低，直到 **XMC\_NWAIT** 回到高电平。

部份 NAND 闪存在收到最后一个地址字节后特别要求 **XMC\_NCE[2]** 保持低电平直到 NAND 闪存回到就绪状态，利用特殊时序寄存器与 **NWEN** 位，即可达到这个要求：

用户只需要根据 NAND 闪存从 **XMC\_NWE** 上升沿到 **XMC\_NWAIT** 下降沿所需的时间配置于 **SPHT** 寄存器，并且将最后一个地址字节写在特殊空间地址区，如此 XMC 就会以特殊空间时序寄存器的时序进行写操作。如图 22-20 的 Address3 所示。

图 22-20 NAND界面等待功能



### 22.5.3 ECC运算

NAND 界面包含了 ECC 运算模块，可在 NAND 界面访问 NAND 闪存时，对数据进行 ECC 的运算，计算完的 ECC 数值存入 ECC 结果寄存器 2 (XMC\_BK2ECC)。

以下示范步骤：

1. 配置ECC页面大小ECCPGS以选择每次计算ECC的字节数目：256、512、1024、2048、4096或8192个字节。
2. 开启ECC使能位ECCEN。
3. 进行数据区的读写。
4. XMC在收到/送出与ECCPGS相同数目的字节后将ECC运算结果存入ECC结果寄存器2 (XMC\_BK2ECC)。
5. 软件读取最后一个字节/写出最后一个字节并且等待FIFOE标志位被置起
6. 软件读取ECC结果寄存器2 (XMC\_BK2ECC) 并进行对应的错误更正流程。
7. 软件清除ECCEN位。重复2~6的步骤。

下表列出字节数目对应的 ECC 结果位数。

表 22-31 ECC结果有效位

ECCPGS	000	001	010	011	100	101
字节数目	256	512	1024	2048	4096	8192
ECC 结果位数	ECC[21: 0]	ECC[23: 0]	ECC[25: 0]	ECC[27: 0]	ECC[29: 0]	ECC[31: 0]

### 22.6 XMC寄存器

必须以字（32 位）的方式操作这些外设寄存器。

表 22-32 XMC寄存器地址映像

寄存器简称	基址偏移量	复位值
XMC_BK1CTRL1	0x000	0x0000 30DB
XMC_BK1TMG1	0x004	0x0FFF FFFF
XMC_BK1CTRL4	0x018	0x0000 30D2
XMC_BK1TMG4	0x01C	0x0FFF FFFF
XMC_BK2CTRL	0x060	0x0000 0018
XMC_BK2IS	0x064	0x0000 0040

XMC_BK2TMGRG	0x068	0xFCFC FCFC
XMC_BK2TMGSP	0x06C	0xFCFC FCFC
XMC_BK2ECC	0x074	0x0000 0000
XMC_BK1TMGWR1	0x104	0x0FFF FFFF
XMC_BK1TMGWR4	0x11C	0x0FFF FFFF
XMC_EXT1	0x220	0x0000 0808
XMC_EXT4	0x22C	0x0000 0808

## 22.6.1 NOR闪存和PSRAM控制器寄存器

### 22.6.1.1 SRAM/NOR闪存片选控制寄存器1（XMC\_BK1CTRL1）

域	简称	复位值	类型	功能
位 31: 20	保留	0x000	resd	保持默认值。
位 19	MWMC	0x0	rw	对存储器写操作位（Memory write mode control） 0：写操作为异步模式； 1：写操作为同步模式。
位 18: 16	CRPGS	0x0	rw	CRAM 页大小选择位（CRAM page size） Cellular RAM 1.5 不允许跨页地址边界的同步访问。同步模式配置这些位时，遇到跨页时，XMC 会自动拆开访问。 000：当跨页地址边界时不会拆开访问（默认值）； 001：128 字节； 010：256 字节； 011：512 字节； 100：1024 字节； 其他：保留。
位 15	NWASEN	0x0	rw	异步传输期间等待信号使能位（NWAIT in asynchronous transfer enable） 0：禁止 NWAIT 信号； 1：使能 NWAIT 信号。
位 14	RWTD	0x0	rw	读写时序不同控制位（Read-write timing different） 读存储器与写存储器使用不同的时序进行操作，即 SRAM/NOR 闪存写时序寄存器（XMC_BK1TMGWR）被开放。 0：读写时序相同； 1：读写时序不同。
位 13	NWSEN	0x1	rw	同步传输期间等待信号使能位（NWAIT in synchronous transfer enable） 0：禁用 NWAIT 信号； 1：使能 NWAIT 信号。
位 12	WEN	0x1	rw	写使能位（Write enable） 0：禁止； 1：使能。
位 11	NWTCFG	0x0	rw	等待时序配置（NWAIT timing configuration） 仅在同步模式有效。 0：NWAIT 信号在等待状态前的一个数据周期有效； 1：NWAIT 信号在等待状态期间有效。
位 10	WRAPEN	0x0	rw	支持非对齐的成组模式（Wrapped enable） XMC 于同步模式时是否支持将非对齐的 AHB 成组操作拆成 2 次操作； 0：不允许直接的非对齐成组操作； 1：允许直接的非对齐成组操作。
位 9	NWPOL	0x0	rw	等待信号极性（NWAIT polarity） 在同步模式下，此位设置 NWAIT 信号极性。 0：低有效； 1：高有效。



位 8	SYNCBEN	0x0	rw	同步突发模式使能（Synchronous burst enable） 允许对闪存存储器进行同步模式访问。 0：禁用； 1：使能。
位 7	保留	0x1	resd	保持默认值。
位 6	NOREN	0x1	rw	NOR 闪存访问使能（Nor flash access enable） 0：禁止 NOR 闪存的访问操作； 1：使能 NOR 闪存的访问操作。
位 5：4	EXTMDBW	0x1	rw	外部存储器数据宽度（External memory data bus width） 外部存储器的数据总线宽度。 00：8 位； 01：16 位； 10：保留； 11：保留。
位 3：2	DEV	0x2	rw	存储器类型（Memory device type） 00：SRAM/ROM； 01：PSRAM（Cellular RAM 或 CRAM）； 10：NOR 闪存； 11：保留。
位 1	ADMUXEN	0x1	rw	地址/数据复用使能位（Address/data multiplexing enable） 0：地址/数据不复用； 1：地址/数据复用数据总线。
位 0	EN	0x1	rw	存储器块使能位（Memory bank enable） 0：禁用存储器块； 1：启用存储器块。

## 22.6.1.2 SRAM/NOR 闪存片选控制寄存器 4（XMC\_BK1CTRL4）

域	简称	复位值	类型	功能
位 31：20	保留	0x000	resd	保持默认值。
位 19	MWMC	0x0	rw	对存储器写操作位（Memory write mode control） 0：写操作为异步模式； 1：写操作为同步模式。
位 18：16	CRPGS	0x0	rw	CRAM 页大小选择位（CRAM page size） Cellular RAM 1.5 不允许跨页地址边界的同步访问。同步模式配置这些位时，遇到跨页时，XMC 会自动拆开访问。 000：当跨页地址边界时不会拆开访问（默认值）； 001：128 字节； 010：256 字节； 011：512 字节； 100：1024 字节； 其他：保留。
位 15	NWASEN	0x0	rw	异步传输期间等待信号使能位（NWAIT in asynchronous transfer enable） 0：禁止 NWAIT 信号； 1：使能 NWAIT 信号。
位 14	RWTD	0x0	rw	读写时序不同控制位（Read-write timing different） 读存储器与写存储器使用不同的时序进行操作，即 SRAM/NOR 闪存写时序寄存器（XMC_BK1TMGWR）被开放。 0：读写时序相同； 1：读写时序不同。
位 13	NWSEN	0x1	rw	同步传输期间等待信号使能位（NWAIT in synchronous transfer enable） 0：禁用 NWAIT 信号； 1：使能 NWAIT 信号。
位 12	WEN	0x1	rw	写使能位（Write enable） 0：禁止； 1：使能。

位 11	NWTCFG	0x0	rw	等待时序配置 (NWAIT timing configuration) 仅在同步模式有效。 0: NWAIT 信号在等待状态前的一个数据周期有效; 1: NWAIT 信号在等待状态期间有效。
位 10	WRAPEN	0x0	rw	支持非对齐的成组模式 (Wrapped enable) XMC 于同步模式时是否支持将非对齐的 AHB 成组操作拆成 2 次操作; 0: 不允许直接的非对齐成组操作; 1: 允许直接的非对齐成组操作。
位 9	NWPOL	0x0	rw	等待信号极性 (NWAIT polarity) 在同步模式下, 此位设置 NWAIT 信号极性。 0: 低有效; 1: 高有效。
位 8	SYNCBEN	0x0	rw	同步突发模式使能 (Synchronous burst enable) 允许对闪存存储器进行同步模式访问。 0: 禁用; 1: 使能。
位 7	保留	0x1	resd	保持默认值。
位 6	NOREN	0x1	rw	NOR 闪存访问使能 (Nor flash access enable) 0: 禁止 NOR 闪存的访问操作; 1: 使能 NOR 闪存的访问操作。
位 5: 4	EXTMDBW	0x1	rw	外部存储器数据宽度 (External memory data bus width) 外部存储器的数据总线宽度。 00: 8 位; 01: 16 位; 10: 保留; 11: 保留。
位 3: 2	DEV	0x0	rw	存储器类型 (Memory device type) 00: SRAM/ROM; 01: PSRAM (Cellular RAM 或 CRAM); 10: NOR 闪存; 11: 保留。
位 1	ADMUXEN	0x1	rw	地址/数据复用使能位 (Address/data multiplexing enable) 0: 地址/数据不复用; 1: 地址/数据复用数据总线。
位 0	EN	0x0	rw	存储器块使能位 (Memory bank enable) 0: 禁用存储器块; 1: 启用存储器块。

## 22.6.1.3 SRAM/NOR 闪存片选时序寄存器 1,4 (XMC\_BK1TMG1,4)

域	简称	复位值	类型	功能
位 31: 30	保留	0x0	resd	保持默认值。
位 29: 28	ASYNM	0x0	rw	异步访问模式选择位 (Asynchronous mode) 只在 RWTD 位使能时有效。 00: 模式 A; 01: 模式 B; 10: 模式 C; 11: 模式 D。
位 27: 24	DTLAT	0xF	rw	数据延迟 (Data latency) 仅在同步模式有效。 0000: 额外插入 0 个 XMC_CLK 周期; 0001: 额外插入 1 个 XMC_CLK 周期; ..... 1111: 额外插入 15 个 XMC_CLK 周期。
位 23: 20	CLKPSC	0xF	rw	时钟分频系数 (Clock prescale) 仅在同步模式有效, 定义 XMC_CLK 时钟的频率。 0000: 保留; 0001: XMC_CLK 周期为 HCLK 周期的 2 倍; 0010: XMC_CLK 周期为 HCLK 周期的 3 倍;

				..... 1111: XMC_CLK 周期为 HCLK 周期的 16 倍。
位 19: 16	BUSLAT	0xF	rw	总线延迟时间 (Bus latency) 为了防止数据总线发生冲突, 在复用模式或同步模式时, 一次读操作之后 XMC 在数据总线上插入延迟。 0000: 插入 1 个 HCLK 周期; 0001: 插入 2 个 HCLK 周期; ..... 1111: 插入 16 个 HCLK 周期。
位 15: 8	DTST	0xFF	rw	数据建立时间 (Data setup time) 0000: 额外插入 0 个 HCLK 周期; 0001: 额外插入 1 个 HCLK 周期; ..... 1111: 额外插入 15 个 HCLK 周期。
位 7: 4	ADDRHT	0xF	rw	地址保持时间 (Address-hold time) 0000: 额外插入 0 个 HCLK 周期; 0001: 额外插入 1 个 HCLK 周期; ..... 1111: 额外插入 15 个 HCLK 周期。
位 3: 0	ADDRST	0xF	rw	地址建立时间 (Address setup time) 0000: 额外插入 0 个 HCLK 周期; 0001: 额外插入 1 个 HCLK 周期; ..... 1111: 额外插入 15 个 HCLK 周期。

## 22.6.1.4 SRAM/NOR 闪存写时序寄存器1,4 (XMC\_BK1TMGWR1,4)

域	简称	复位值	类型	功能
位 31: 30	保留	0x0	resd	保持默认值。
位 29: 28	ASYNCM	0x0	rw	异步访问模式选择位 (Asynchronous mode) 只在 RWTB 位使能识有效。 00: 模式 A; 01: 模式 B; 10: 模式 C; 11: 模式 D。
位 27: 20	保留	0xFF	resd	保持默认值。
位 19: 16	BUSLAT	0xF	rw	总线延迟时间 (Bus latency) 为了防止数据总线发生冲突, 在复用模式或同步模式时, 一次读操作之后 XMC 在数据总线上插入延迟。 0000: 插入 1 个 HCLK 周期; 0001: 插入 2 个 HCLK 周期; ..... 1111: 插入 16 个 HCLK 周期。
位 15: 8	DTST	0xFF	rw	数据建立时间 (Data setup time) 0000: 额外插入 0 个 HCLK 周期; 0001: 额外插入 1 个 HCLK 周期; ..... 1111: 额外插入 15 个 HCLK 周期。
位 7: 4	ADDRHT	0xF	rw	地址保持时间 (Address-hold time) 0000: 额外插入 0 个 HCLK 周期; 0001: 额外插入 1 个 HCLK 周期; ..... 1111: 额外插入 15 个 HCLK 周期。
位 3: 0	ADDRST	0xF	rw	地址建立时间 (Address setup time) 0000: 额外插入 0 个 HCLK 周期; 0001: 额外插入 1 个 HCLK 周期; ..... 1111: 额外插入 15 个 HCLK 周期。

## 22.6.1.5 SRAM/NOR额外时序寄存器1,4 (XMC\_EXT1,4)

域	简称	复位值	类型	功能
位 31: 16	保留	0x0000	resd	保持默认值。
位 15: 8	BUSLATR2R	0x08	rw	连续读操作恢复时间 (Bus turnaround phase for consecutive read duration) 用于定义之连续读操作间总线恢复时间。 00000000: 连续读操作额外插入 1 个 HCLK 周期; 00000001: 连续读操作额外插入 2 个 HCLK 周期; ..... 00001000: 连续读操作额外插入 9 个 HCLK 周期 (默认值)。 ..... 11111111: 连续读操作额外插入 256 个 HCLK 周期
位 7: 0	BUSLATW2W	0x08	rw	连续写操作恢复时间 (Bus turnaround phase for consecutive write duration) 用于定义之连续写操作间总线恢复时间。 00000000: 连续写操作额外插入 1 个 HCLK 周期; 00000001: 连续写操作额外插入 2 个 HCLK 周期; ..... 00001000: 连续写操作额外插入 9 个 HCLK 周期 (默认值); ..... 11111111: 连续写操作额外插入 256 个 HCLK 周期。

## 22.6.2 NAND闪存控制器寄存器

### 22.6.2.1 NAND闪存控制寄存器2 (XMC\_BK2CTRL)

域	简称	复位值	类型	功能
位 31: 20	保留	0x000	resd	保持默认值。
位 19: 17	ECCPGS	0x0	rw	ECC 页面大小 (ECC size) 000: 256 字节; 001: 512 字节; 010: 1024 字节; 011: 2048 字节; 100: 4096 字节; 101: 8192 字节。
位 16: 13	TAR	0x0	rw	ALE 至 RE 的延迟时间 (ALE to RE delay) 从 ALE 下降沿至 RE 下降沿的时间。 0000: 1 个 HCLK 周期; ..... 1111: 16 个 HCLK 周期。
位 12: 9	TCR	0x0	rw	CLE 至 RE 的延迟 (CLE to RE delay) 从 CLE 下降沿至 RE 下降沿的时间。 0000: 1 个 HCLK 周期; ..... 1111: 16 个 HCLK 周期。
位 8: 7	保留	0x0	resd	保持默认值。
位 6	ECCEN	0x0	rw	ECC 使能位 (ECC enable) 0: 禁用 ECC 运算; 1: 使能 ECC 运算。
位 5: 4	EXTMDBW	0x1	rw	外部存储器数据宽度 (External memory data bus width) 定义外部 NAND 闪存数据总线的宽度。 00: 8 位; 01: 16 位; 10: 保留; 11: 保留。
位 3	DEV	0x1	rw	存储器类型 (Memory device type) 0: 保留; 1: NAND 闪存。
位 2	EN	0x0	rw	存储器块使能位 (Memory bank enable)

				0: 禁用存储器块; 1: 启用存储器块。
位 1	NWEN:	0x0	rw	等待功能使能位 (Wait feature enable) 使能 NAND 闪存存储器块的等待功能 0: 关闭; 1: 使能。
位 0	保留	0x0	resd	保持默认值。

## 22.6.2.2 中断使能和FIFO状态寄存器2 (XMC\_BK2IS)

域	简称	复位值	类型	功能
位 31: 7	保留	0x000000	resd	保持默认值。
位 6	FIFOE	0x1	ro	FIFO 空标志 (FIFO empty) 当 FIFO 为空时, 硬件自动置起。 0: FIFO 非空; 1: FIFO 空。 XMC 的 FIFO 大小为 16 个字, 用于储存从 AHB 送来的数据。
位 5	FEIEN	0x0	rw	下降沿中断使能 (Falling edge interrupt enable) 0: 关闭; 1: 开启。
位 4	HLIEN	0x0	rw	高电平中断使能 (High-level interrupt enable) 0: 关闭; 1: 开启。
位 3	REIEN	0x0	rw	上升沿中断使能 (Rising edge interrupt enable) 0: 关闭; 1: 开启。
位 2	FES	0x0	rw	下降沿状态 (Falling edge status) 该位由硬件设置, 软件清除。 0: 没有产生下降沿中断; 1: 产生下降沿中断。
位 1	HLS	0x0	rw	高电平状态 (High-level status) 该位由硬件设置, 软件清除。 0: 没有产生高电平中断; 1: 产生高电平中断。
位 0	RES	0x0	rw	上升沿状态 (Rising edge status) 该位由硬件设置, 软件清除。 0: 没有产生上升中断; 1: 产生上升中断。

## 22.6.2.3 常规空间时序寄存器2 (XMC\_BK2TMGRG)

域	简称	复位值	类型	功能
位 31: 24	RGDHI2T	0xFC	rw	在常规空间数据总线的高阻时间 (Regular memory databus High resistance time) 定义在常规空间开始执行对 NAND 闪存的写操作后数据总线的高阻时间 00000000: 额外插入 0 个 HCLK 周期; 00000001: 额外插入 1 个 HCLK 周期; ..... 11111111: 额外插入 255 个 HCLK 周期。
位 23: 16	RGHT	0xFC	rw	在常规空间的保持时间 (Regular memory hold time) 定义在在常规空间对进行访问时, 数据总线保持的时间。 00000000: 保留; 00000001: 插入 1 个 HCLK 周期; ..... 11111111: 插入 255 个 HCLK 周期。
位 15: 8	RGWT	0xFC	rw	在常规空间的等待时间 (Regular memory wait time) 定义在在常规空间对进行访问时, XMC_NWE、XMC_NOE 为低的时间。 00000000: 额外插入 0 个 HCLK 周期; 00000001: 额外插入 1 个 HCLK 周期;

				..... 11111111: 额外插入 255 个 HCLK 周期。
				在常规空间的建立时间 (Regular memory setup time) 定义在在常规空间对进行访问时, 地址线的建立时间。
位 7: 0	RGST	0xFC	rw	00000000: 额外插入 0 个 HCLK 周期; 00000001: 额外插入 1 个 HCLK 周期;
				..... 11111111: 额外插入 255 个 HCLK 周期。

#### 22.6.2.4 特殊空间时序寄存器2 (XMC\_BK2TMGSP)

域	简称	复位值	类型	功能
				在特殊空间数据总线的高阻时间 (Special memory databus High resistance time) 定义在特殊空间开始执行对 NAND 闪存的写操作后数据总线的高阻时间
位 31: 24	SPDHIZT	0xFC	rw	00000000: 额外插入 0 个 HCLK 周期; 00000001: 额外插入 1 个 HCLK 周期;
				..... 11111111: 额外插入 255 个 HCLK 周期。
				在特殊空间的保持时间 (Special memory hold time) 定义在在特殊空间对进行访问时, 数据总线保持的时间。
位 23: 16	SPHT	0xFC	rw	00000000: 保留; 00000001: 插入 1 个 HCLK 周期;
				..... 11111111: 插入 255 个 HCLK 周期。
				在特殊空间的等待时间 (Special memory wait time) 定义在在特殊空间对进行访问时, XMC_NWE、XMC_NOE 为低的时间。
位 15: 8	SPWT	0xFC	rw	00000000: 额外插入 0 个 HCLK 周期; 00000001: 额外插入 1 个 HCLK 周期;
				..... 11111111: 额外插入 255 个 HCLK 周期。
				在特殊空间的建立时间 (Special memory setup time) 定义在在特殊空间对进行访问时, 地址线的建立时间。
位 7: 0	SPST	0xFC	rw	00000000: 额外插入 0 个 HCLK 周期; 00000001: 额外插入 1 个 HCLK 周期;
				..... 11111111: 额外插入 255 个 HCLK 周期。

#### 22.6.2.5 ECC结果寄存器2 (XMC\_BK2ECC)

域	简称	复位值	类型	功能
位 31: 0	ECC	0x0000 0000	ro	ECC 结果 (ECC value) 硬件计算的 ECC 结果。

## 23 SDIO 接口

### 23.1 简介

SD/SDIO MMC 卡主机模块（SDIO）在 AHB 外设总线和多媒体卡（MMC）、SD 存储卡、SDIO 卡间提供了操作接口。

SD 储存卡和 SDIO 卡的系统规格书可以通过 SD 卡协议网站([www.sdcard.org](http://www.sdcard.org))

多媒体卡系统规格书由 MMCA 技术委员会发布,可以在多媒体卡协会的网站([www.mmca.org](http://www.mmca.org))获得。

### 23.2 主要特点

- 与 SD 储存卡 2.0 规格版本全兼容
- 与 SDIO 卡 2.0 规格版本全兼容并支持 1 位和 4 位数据总线模式
- 与多媒体卡 4.2 规格版本全兼容并支持 1 位、4 位和 8 位数据总线模式
- 与较早的多媒体卡规格版本全兼容
- 支持 DMA 传输
- 8 位总线模式下数据传输速率可达 50 MHz
- 中断请求

**注意：**SDIO 并不兼容 SPI 的通信模式，并且在同一时间内只能支持一个 SD/SDIO/MMC 4.2 卡

总线上的通信是通过传送命令和数据实现。

- 命令：命令是启动操作的令牌。命令从主机发送到单个卡（寻址命令）或所有连接的卡（广播命令），命令在 CMD 总线上串行传输。
- 响应：响应是从卡发送到主机，作为对先前命令的答复，响应在 CMD 总线上串行传输
- 数据：数据可以从卡传送到主机或是主机传送到卡端，数据通过 SDIO\_D 数据总线进行传送

MMC 卡/SD 卡/SDIO 卡在总线上的基本操作是命令/响应结构，这样的总线操作在命令或总线机制下实现信息交换；另外，某些操作还具有数据令牌。

在 SD/SDIO 存储器卡上传送的数据是以数据块的形式传输，数据块总是以 CRC 位为后，定义了单个和多个块操作，在 MMC 上传送的数据是以数据块或数据流的形式传输，详细可参考下列图示。

图 23-1 SDIO“无响应”和“无数据”操作

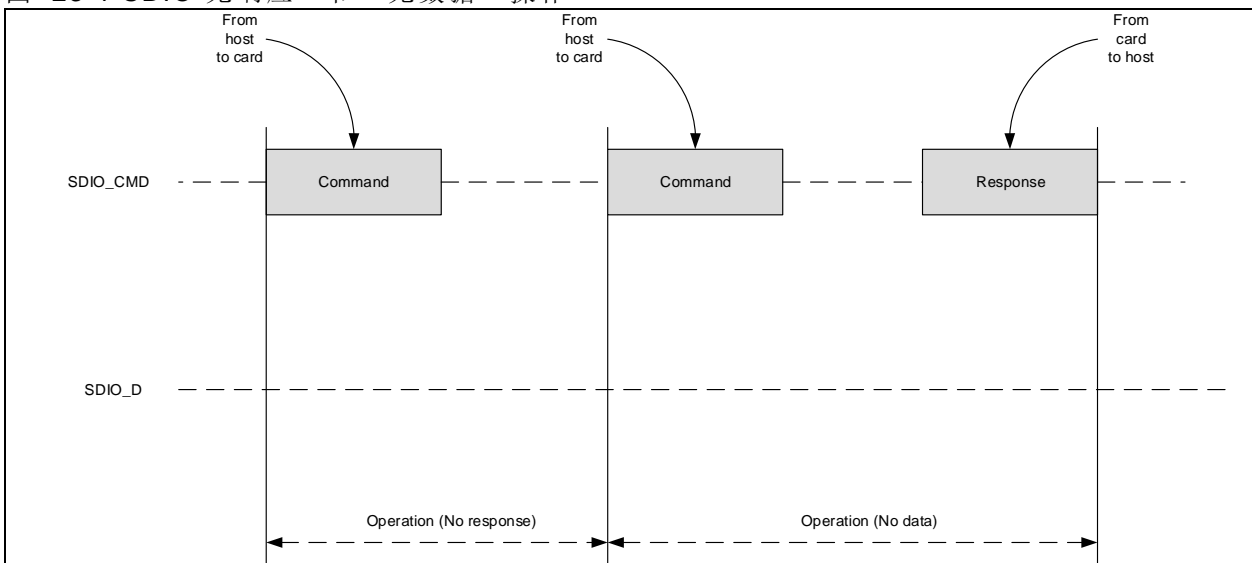




图 23-2 SDIO（多）数据块读操作

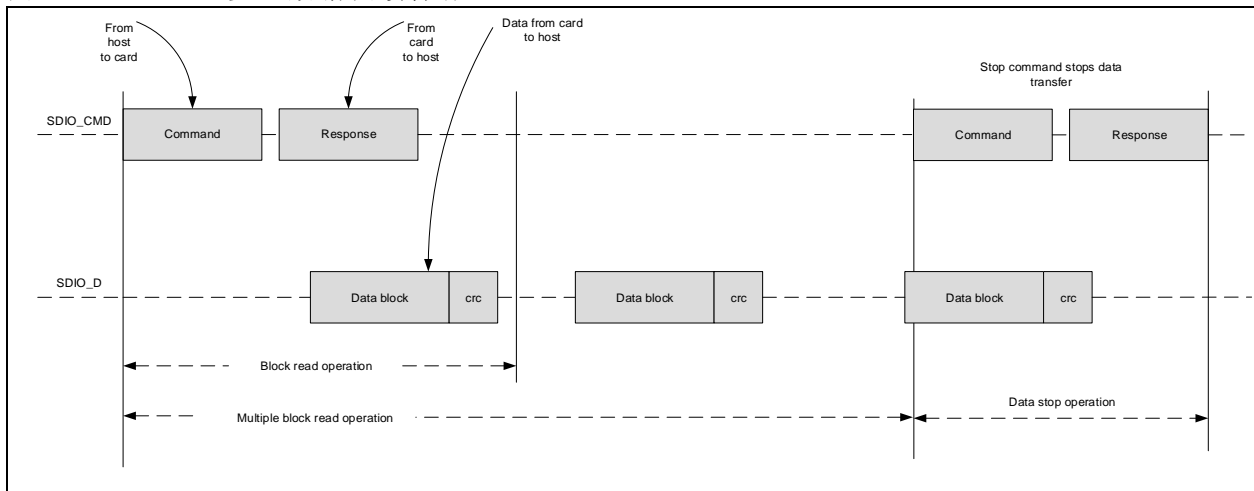
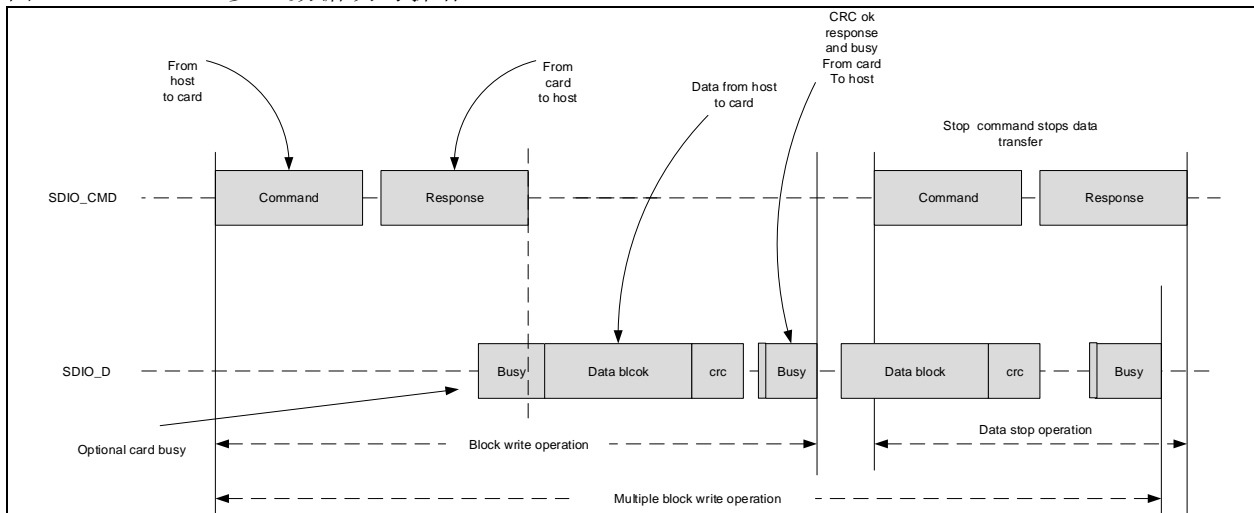


图 23-3 SDIO（多）数据块写操作



注意：当有 *Busy*（繁忙）信号时，*SDIO*（*SDIO\_D0* 被拉低）将不会发送任何数据。

图 23-4 SDIO连续读操作

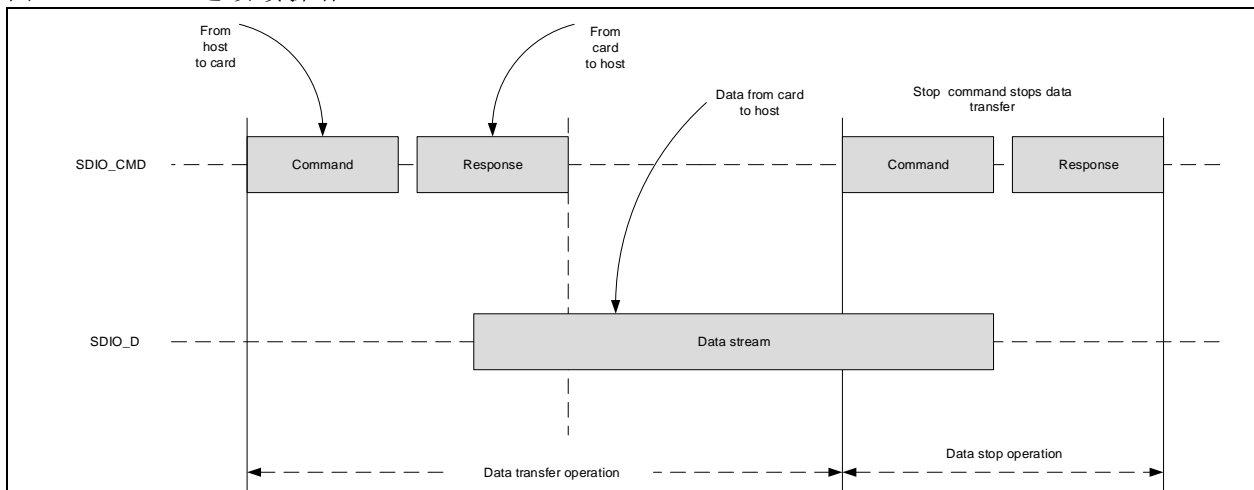
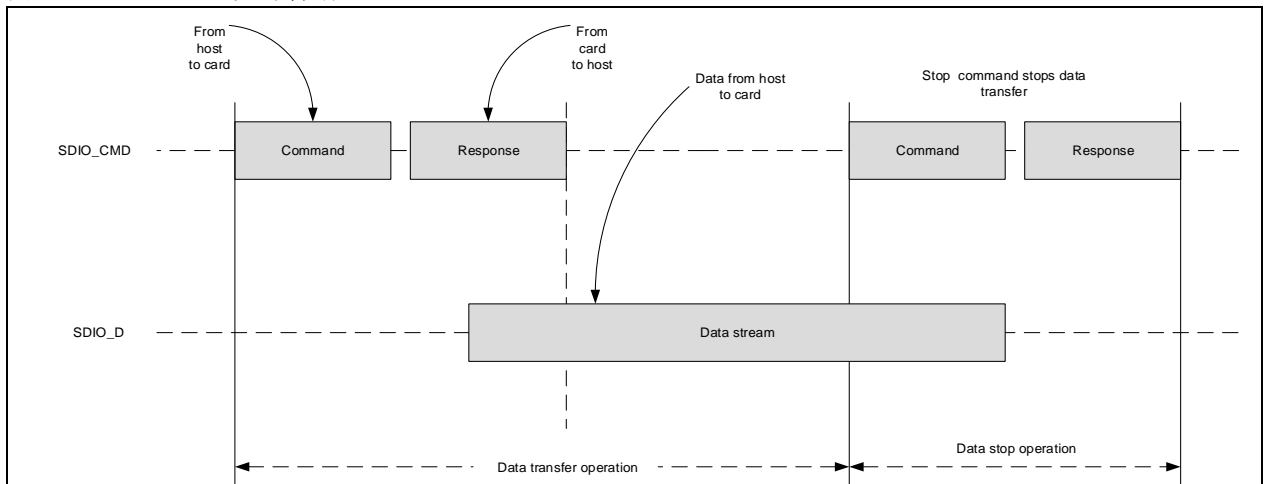


图 23-5 SDIO连续写操作



## 23.3 功能描述

### 23.3.1 卡功能描述

主机与卡之间的所有通信都由卡来控制，主机发送两种类型的命令：广播和寻址（点对点）命令。

- 广播命令：适用于所有卡，其中一些命令需要响应
- 寻址命令：发送到寻址的卡，并引起卡的响应

存储卡定义了两种操作模式：

- 卡识别模式
- 数据传输模式

#### 23.3.1.1 卡识别模式

在卡识别模式下，主机会重置处于卡识别模式的所有卡和检测工作电压范围，识别卡并要求它们发布相对卡地址(RCA)，对每个卡在其自己的 CMD 在线分别执行此操作，在卡识别模式下的所有通信都使用命令信号线(CMD)。

##### 卡识别过程

不同的卡有不同的识别过程，主机需要发送不同的命令，卡的类型可以分为 SD 卡、SDI/O 卡和 MMC 卡，要识别卡的类型可以发送 CMD5 命令，如果主机接收到响应，卡的类型就是 SDI/O 卡，若没有响应那接着发送 ACMD41 命令，如果主机接收到响应就是 SD 卡，否则就是 MMC 卡。

以下描述卡的识别过程：

1. 总线被激活，检测卡是否连接，卡识别过程中时钟频率为0-400kHz。
2. SDIO主机发送命令识别卡的类型是SD卡、SDI/O卡或是MMC卡。
3. 根据卡的类型进行初始化
  - SD卡：SDIO主机发送CMD2(ALL\_SEND\_CID)，以获得其唯一的卡标志(CID号)，卡发送CID号作为响应后主机发送CMD3(SEND\_RELATIVE\_ADDR)要求卡发布新的相对卡地址(RCA)，该地址比CID短并且用于之后的数据传输模式下寻址卡。
  - SDI/O卡：SDIO主机发送CMD3(SEND\_RELATIVE\_ADDR)要求卡发布新的相对卡地址(RCA)，该地址比CID短并且用于之后的数据传输模式下寻址卡。
  - MMC卡：SDIO主机发送CMD1(SEND\_OP\_COND)，接着发送CMD2和CMD3。
4. 如果主机要分配另一个RCA号码，则可以通过向该卡发送另一个CMD3命令来要求该卡发布新的号码，最后发布的RCA是卡的实际RCA编号，主机重复识别过程即系统中每个卡的CMD2和CMD3循环。

#### 23.3.1.2 数据传输模式

主机在识别总在线的所有卡后将进入数据传输模式，在数据传输模式下主机可以在 0 - 50MHz 频率范围内操作卡，主机可以发出 CMD9 (SEND\_CSD)以获取卡特定数据(CSD 寄存器)，例如块长度和卡储存容

量等。在数据传输模式下的所有数据通信都是主机与所选卡之间的点对点传输，CMD 总在线的响应会确认所有以寻址的命令，数据传输读写可分为数据块模式和数据流模式，可以在 SDIO 数据控制寄存器（SDIO\_DTCTRL）的 TFRMODE 位做设置，在数据流模式，数据按字节传输，同时每个数据块后没有 CRC。

#### 宽总线选择/解除

对于 SD 卡可以使用 ACMD6(SET\_BUS\_WIDTH)命令选择或解除宽总线(4 位总线宽度)操作模式，上电或 CMD0(GO\_IDLE\_STATE)后默认总线宽度为 1 位，ACMD6 命令仅在传输状态时有效也就是在经过 CMD7 选择卡之后才可以改变总线宽度。

#### 数据流读写(只适用于多媒体卡)

读取：

1. 主机发送过 CMD11(READ\_DAT\_UNTIL\_STOP)进行数据流读取。
2. 直到主机发送 CMD12 ( STOP\_TRANSMISSION )，由于串行命令的发送，停止命令具有执行延迟，在停止命令的结束位后数据传输停止。

写入：

1. 主机发送 CMD20 (WRITE\_DAT\_UNTIL\_STOP)进行数据流写入。
2. 直到主机发送 CMD12(STOP\_TRANSMISSION)，由于未预先确定要传输的数据量，因此无法使用 CRC，如果主机提供超出范围的地址作为 CMD20 的参数，则卡将拒绝该命令，保持在传输状态，并通过将 ADDRESS\_OUT\_OF\_RANGE 位置 1 进行响应。

#### 数据块读取

在数据块读取的模式下，数据传输的基本单位是块，最大块大小在 CSD(READ\_BL\_LEN)定义，其最大大小始终为 512 字节，如果设置了 READ\_BL\_PARTIAL，可以发送其起始和结束地址完全包含在 512 字节边界内较小的数据块，CRC 会附加到每个块的末尾用以确保数据传输的正确，数据块读取有几个相关的命令操作如下：

- CMD17 ( READ\_SINGLE\_BLOCK )：启动数据块读取，完成传输后卡返回到传输状态。
- CMD18 ( READ\_MULTIPLE\_BLOCK )：开始传输几个连续的数据块。

数据块将连续传输直到主机发出 CMD12(STOP\_TRANSMISSION)，由于串行命令的发送，停止命令具有执行延迟，在停止命令的结束位后数据传输停止。

#### 数据块写入

在执行数据块写入命令(CMD24-27)时，一个或多个数据块从主机传输到卡，CRC 会附加到每个数据块的末尾，如果 CRC 检测失败，卡通过 SDIO\_D 信号线指示错误，传送数据被丢弃而不写入，并且发送的数据块将被忽略。

如果主机使用的部分块的累积长度未对齐，并且不允许块未对齐（未设置 CSD 参数 WRITE\_BLK\_MISALIGN），则卡应检测到块未对齐错误，并在第一个未对齐块开始之前中止编程。卡片应当在 SDIO 状态寄存器（SDIO\_STS）中设置 ADDRESS\_ERROR 错误位，并且在忽略所有进一步的数据传输的同时，在接收数据状态中等待停止命令，如果主机试图在写保护区域上进行写操作，则写操作也应中止。但是，在这种情况下，卡应将 WP\_VIOLATION 位置 1。

设置 CID 和 CSD 寄存器不需要事先设置块长度，传送的数据也受 CRC 保护的，如果 CSD 或 CID 寄存器的部分是存储在 ROM 中，则该不可更改的部分应与接收缓冲区的一部分匹配，若匹配失败，则卡将报告错误并且不会更改任何寄存器的内容，某些卡可能需要很长且不可预测的时间来写入数据块。接收到数据块并完成 CRC 检查后，如果卡的写缓冲区已满并且无法从新的 WRITE\_BLOCK 命令接受新数据，则该卡将开始写并保持 SDIO\_D 信号线为低电平，主机可以随时使用 SEND\_STATUS 命令（CMD13）查询卡的状态，卡将以其状态进行响应。状态位 READY\_FOR\_DATA 指示卡是否可以接受新数据或写入过程是否仍在进行中，主机可以通过发出 CMD7（选择另一张卡）来取消选择卡，这将使卡进入断开状态并释放 SDIO\_D 信号线而不会中断写入操作。重新选择卡时，如果编程仍在进行且写缓冲区不可用，它将通过将 SDIO\_D 信号线拉至低电平来重新激活忙碌指示。

### 23.3.1.3 擦除

多媒体卡和 SD 卡的擦除单位是擦除组，以写数据块计算，写数据块是卡的基本写入单位，擦除组的大小是卡的特定参数，在 CSD 中定义。

主机能擦除一个连续范围的擦除组，开始擦除操作有三个步骤，而多媒体卡和 SD 卡发送的命令有所不

同。

1. 主机发送命令定义连续范围的开始地址
  - SD卡：发送CMD32 (ERASE\_WR\_BLK\_START)
  - MMC卡：发送CMD35 (ERASE\_GROUP\_START)
2. 主机发送命令定义连续范围的结束地址
  - SD卡：发送CMD33(ERASE\_WR\_BLK\_END)
  - MMC卡：发送CMD36(ERASE\_GROUP\_END)
3. 主机发送擦除命令CMD38(ERASE)，开始擦除操作

### 23.3.1.4 保护管理

SDIO 卡主机模块支持三种保护方式，使主机保护数据不被擦除或改写，如下所示：

#### 机械写保护开关

在卡的侧边有一个机械的滑动开关，使用户设置是否对卡进行写保护，如果滑动平板计算机以窗口打开的方式放置，则表示卡已被写保护。如果窗口关闭，则该卡不受写保护。

#### 卡的内部写保护

卡数据可以受到保护，不被擦除或写入。通过设置 CSD 中的永久或临时写保护位，制造商或内容提供商可以对整个卡进行永久性写保护。支持通过设置扇区组写保护的卡可以设置 CSD 中的 WP\_GRP\_ENABLE 位以可保护部分数据，并且写保护可由应用程序更改。SET\_WRITE\_PROT 命令设置寻址的写保护组的写保护，CLR\_WRITE\_PROT 命令清除寻址的写保护组的写保护。

SEND\_WRITE\_PROT 命令类似于单个块读取命令。卡应发送一个数据块，该数据块包含 32 个写保护位（代表从指定地址开始的 32 个写保护组），后跟 16 个 CRC 位。写保护命令中的地址域是一个字节为单位的组地址。

#### 密码保护卡锁定

SDIO 卡主机可以使用密码保护功能对卡锁定或解锁，密码储存在 128 位的 PWD 寄存器中，密码长度设置储存在 8 位的 PWD\_LEN 寄存器中，这些寄存器是非挥发性的，掉电后不会清除寄存器的内容。

已锁定的卡能够支持基本的命令，主机可以对卡进行复位、初始化和状态查询等操作，但无法获取卡中的数据，当设置了密码后(PWD\_LEN 不为 0)，上电后卡自动锁定。

与 CSD 和 CID 寄存器写入命令相似，锁定/解锁命令仅在传输状态下有效，锁定/解锁命令不包含地址参数所以在使用前卡必须要被选中。

卡的锁定/解锁命令具有单数据块写命令的结构和总线操作类型，传输的数据块包含所有命令所需要的信息（密码设置模式、PWD 内容和上锁/解锁指示）。在发送卡的锁定/解锁命令之前，命令数据块的长度由 SDIO 卡主机模块定义，锁定/解锁命令的结构如下表：

表 23-1 锁定/解锁命令的结构

Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	保留(需设为 0)				ERASE	LOCK_UNLOCK	CLR_PWD	SET_PWD
1	PWDS_LEN							
2	密码数据							
...								
PWDS_LEN+1								

- ERASE：将该位置 1 将执行强制擦除，所有其它位必须为 0，只发送命令字节
- LOCK\_UNLOCK：将该位置 1 锁住卡，置 0 解锁卡，LOCK\_UNLOCK 与 SET\_PWD 可以同时设置，但不能与 CLR\_PWD 同时设置
- CLR\_PWD：将该位置 1 清除密码数据
- SET\_PWD：将该位置 1 将密码数据保存至存储器
- PWD\_LEN：以字节为单位定义密码的长度，在改变密码的情况下，长度应该是新旧密码长度和
- PWD：密码（依不同的命令，新的密码或正在使用的密码）

数据块大小应由主机在发送卡锁定/解锁命令之前定义。块长度应设置为大于或等于锁定/解锁命令所需的数据结构。

以下几节列出了设置/清除密码、上锁/解锁和强制擦除的命令序列。

#### 设置密码

1. 如果先前未选择卡，先使用CMD7 (SELECT/DESELECT\_CARD) 选择一个卡
2. 使用CMD16(SET\_BLOCKLEN)定义要在8位的卡锁定/解锁模式下发送的数据块长度，8位的PWD\_LEN，新密码的字节数目。当更换了密码后，发送命令的数据块长



度必须同时考虑新旧密码的长度。

3. 在数据线上以合适的块长度发送CMD42(LOCK/UNLOCK)命令,并包含16位的CRC码。数据块包含了操作模式(SET\_PWD=1)、长度(PWD\_LEN)和密码(PWD)。在完成密码替换的情况下,长度数值(PWD\_LEN)包含了新旧两个密码的长度,PWD域包含了旧的密码(正在使用的)和新的密码。
4. 当旧的密码匹配后,新的密码和它的长度被分别存储在PWD和PWD\_LEN域。如果发送的旧密码不正确(大小和内容不相等),则SDIO状态寄存器(SDIO\_STS)中的LOCK\_UNLOCK\_FAILED错误位将被设置,并且旧密码不会更改。如果发送的旧密码正确(大小和内容相等),则给定的新密码及其大小将分别保存在PWD和PWD\_LEN寄存器中。

密码长度域(PWD\_LEN)指示当前是否设置了密码,等于0时,未设置密码。如果PWD\_LEN的值不等于零,则卡在上电后会自行锁定。如果该域为非零,则表示使用了密码,卡在上电时自动上锁。在不断电的情况下,如果设置了密码,可以通过设置LOCK\_UNLOCK位或发送一个额外的锁定命令,立即锁住卡。

#### 清除密码

1. 如果先前未选择卡,先使用CMD7(SELECT/DESELECT\_CARD)选择一个卡
2. 使用CMD16(SET\_BLOCKLEN)定义要在8位的卡锁定/解锁模式下发送的数据块长度,8位的PWD\_LEN,当前密码的字节数目。
3. 在数据线上以合适的块长度发送CMD42(LOCK/UNLOCK)命令,并包含16位的CRC码。数据块包含了操作模式(SET\_PWD=1)、长度(PWD\_LEN)和密码(PWD)。当密码匹配后,PWD域被清除同时PWD\_LEN被设为0。如果送出的密码与期望的密码(长度或内容)不吻合,则设置SDIO状态寄存器(SDIO\_STS)中的LOCK\_UNLOCK\_FAILED错误位,同时密码不变。

#### 卡锁定

1. 如果先前未选择卡,先使用CMD7(SELECT/DESELECT\_CARD)选择一个卡
2. 使用CMD16(SET\_BLOCKLEN)定义要在8位的卡锁定/解锁模式下发送的数据块长度,8位的PWD\_LEN,当前密码的字节数目。
3. 在数据线上以合适的块长度发送CMD42(LOCK/UNLOCK)命令,并包含16位的CRC码。数据块包含了操作模式(SET\_PWD=1)、长度(PWD\_LEN)和密码(PWD)。
4. 当密码匹配后,卡被锁定且设置SDIO状态寄存器(SDIO\_STS)中的CARD\_IS\_LOCKED状态位。如果送出的密码与期望的密码(长度或内容)不吻合,则设置SDIO状态寄存器(SDIO\_STS)中的LOCK\_UNLOCK\_FAILED错误位,同时锁定操作失败。

如果曾经设置过密码(PWD\_LEN不为0),卡会在上电复位时自动地上锁。对已经上锁的卡执行上锁操作或对没有密码的卡执行上锁操作会导致失败,并设置SDIO状态寄存器(SDIO\_STS)中的LOCK\_UNLOCK\_FAILED错误位。

#### 卡解锁

1. 如果先前未选择卡,先使用CMD7(SELECT/DESELECT\_CARD)选择一个卡
2. 使用CMD16(SET\_BLOCKLEN)定义要在8位的卡锁定/解锁模式下发送的数据块长度,8位的PWD\_LEN,当前密码的字节数目。
3. 在数据线上以合适的块长度发送CMD42(LOCK/UNLOCK)命令,并包含16位的CRC码。数据块包含了操作模式(SET\_PWD=1)、长度(PWD\_LEN)和密码(PWD)。
4. 当密码匹配后,卡锁被解除,同时SDIO状态寄存器(SDIO\_STS)中的CARD\_IS\_LOCKED位被清除。如果送出的密码与期望的密码(长度或内容)不吻合,则设置SDIO状态寄存器(SDIO\_STS)中的LOCK\_UNLOCK\_FAILED错误位,同时卡仍保持上锁状态。

解锁状态只在当前的供电过程中有效,只要不清除PWD域,下次上电后卡会被自动上锁。

试图对已经解了锁的卡执行解锁操作会导致操作失败,并设置SDIO状态寄存器(SDIO\_STS)中的LOCK\_UNLOCK\_FAILED错误位。

#### 强制擦除

如果用户忘记了密码(PWD的内容),可以在清除卡中的所有内容后使用卡。强制擦除操作擦除所有卡中的数据和密码。

1. 如果先前未选择卡，先使用CMD7（SELECT/DESELECT\_CARD）选择一个卡
2. 使用CMD16(SET\_BLOCKLEN)定义要在8位的卡锁定/解锁模式下发送的数据块长度，8位的PWD\_LEN，当前密码的字节数目。
3. 在数据线上以合适的的数据块长度发送CMD42(LOCK/UNLOCK)命令，并包含16位的CRC码。数据块包含了操作模式（ERASE=1）所有其它位为0。
4. 当ERASE位是数据域中仅有的位时，卡中的所有内容将被擦除，包括PWD和PWD\_LEN域，同时卡不再被上锁。如果有任何其它位不为0，则设置SDIO状态寄存器（SDIO\_STS）中的LOCK\_UNLOCK\_FAILED错误位，卡中的数据保持不变，同时卡仍保持上锁状态。

试图对已经解了锁的卡执行擦除操作会导致操作失败，并设置 SDIO 状态寄存器（SDIO\_STS）中的 LOCK\_UNLOCK\_FAILED 错误位。

## 23.3.2 命令与响应

### 23.3.2.1 命令

#### 命令类型

四种命令来控制 SD 储存卡：

1. 广播命令：发送到所有卡无响应
2. 带有响应的广播命令：发送到所有卡，收到来自所有卡的同时响应
3. 寻址命令：发送到已选定的卡，SDIO\_D数据在线没有数据传输
4. 已寻址数据传输命令：发送到已选定的卡，SDIO\_D数据在线有数据传输

#### 详细命令描述

SDIO 主机模块系统是用于提供一个适用于多种应用类型的标准接口，但同时又要兼顾特定用户和应用的功能，因此标准中定义了两类通用命令：通用命令（GEN\_CMD）和应用相关命令（ACMD）。

若要使用应用相关命令，SDIO 主机需先发送 CMD55(APP\_CMD)，待卡响应给主机指示设置了 APP\_CMD 位并等待 ACMD 命令，接着再发送 ACMD 命令。

表 23-2基于命令

CMD 索引	类型	参数	响应格式	缩写	说明
CMD0	bc	[31: 0]=填充位	-	GO_IDLE_STATE	复位所有的卡到空闲状态
CMD1	bc	[31: 0]=OCR	R3	SEND_OP_COND	在空闲状态请求卡通过 CMD 总线发送 OCR 寄存器的内容
CMD2	bcr	[31: 0]=填充位	R2	ALL_Send_CID	请求所有卡通过 CMD 总线发送 CID 数据
CMD3	bcr	[31: 0]=填充位	R6	SEND_RELATIVE_ADDR	请求卡发布新的相对卡地址(RCA)
CMD4	bc	[31: 16]=DSR [15: 0]=填充位	-	SET_DSR	设置所有卡的 DSR 寄存器
CMD5	bcr	[31: 24]保留位 [23: 0] I/O OCR	R4	IO_SEND_OP_COND	仅用于 SDIO 卡，查询所需要的 I/O 卡电压范围
CMD6	ac	[31: 26] 设为 0 [25: 24] 访问 [23: 16] 索引 [15: 8] 值 [7: 3] 设为 0 [2: 0] 命令集	R1b	SWITCH	仅用于 MMC 卡，切换选择卡的操作模式或是修改 EXT_CSD 寄存器
CMD7	ac	[31: 16]=RCA [15: 0]=填充位	R1b	SELECT/DESELECT_CARD	这个命令用于卡在待机状态和发送状态之间切换，或是编成和断开状态间切换，若要选择该卡则用他自己的相对地址，地址 0 用于取消选择该卡

CMD8 (SD)	bcr	[31: 12]保留位 [11: 8]工作电压 (VHS) [7: 0]检查模式	R7	SEND_IF_ COND	向 SD 卡发送主机供电电压讯息和询问 卡是否支持电压
CMD8 (MMC)	adtc	[31: 0]=填充位	R1	SEND_EXT_ CSD	仅用于 MMC 卡, 卡发送自己的 EXT_CSD 寄存器作为数据块
CMD9	ac	[31: 16]=RCA [15: 0]=填充位	R2	SEND_CSD	被选择的卡通过 CMD 总线发送 CSD(卡 特定数据)
CMD10	ac	[31: 16]=RCA [15: 0]=填充位	R2	SEND_CID	被选择的卡通过 CMD 总线发送 CID(卡 标志)
CMD12	ac	[31: 0]=填充位	R1b	STOP_TRANSMI SSION	强制卡停止传输
CMD13	ac	[31: 16]=RCA [15: 0]=填充位	R1	SEND_ STATUS	被选择的卡发送状态寄存器
CMD15	ac	[31: 16]=RCA [15: 0]=填充位	-	GO_INACTIVE_ STATE	被选择的卡切换到非激活状态

表 23-3 数据块读取命令

CMD 索引	类型	参数	响应格式	缩写	说明
CMD16	ac	[31: 0]=数据块长度	R1	SET_ BLOCKLEN	该命令为所有后续块命令设置数据块长 度(以字节为单位), 默认是 512 字节
CMD17	adtc	[31: 0]=数据地址	R1	READ_ SINGLE_ BLOCK	读取由 CMD16 设置大小的数据块
CMD18	adtc	[31: 0]=数据地址	R1	READ_ MULTIPLE_ BLOCK	不断从卡读取数据到主机, 直到收到 STOP_TRANSMISSION 命令

表 23-4 数据流读取和写入命令

CMD 索引	类型	参数	响应格式	缩写	说明
CMD11	adtc	[31: 0]=数据地址	R1	READ_DAT_ UNTIL_STOP	从卡中读取数据流, 从给定的地址开 始, 直到收到 STOP_TRANSMISSION 命令
CMD20	adtc	[31: 0]=数据地址	R1	WRITE_DAT_ UNTIL_STOP	从主机写数据流, 从给定的地址开始, 直到收到 STOP_TRANSMISSION 命令



表 23-5数据块写入命令

CMD 索引	类型	参数	响应格式	缩写	说明
CMD16	ac	[31: 0]=数据块长度	R1	SET_BLOCKLEN	该命令为所有后续块命令设置数据块长度(以字节为单位), 默认是 512 字节
CMD23	ac	[31: 16]=设为 0 [15: 0]=数据块数	R1	SET_BLOCK_COUNT	定义后续数据块读写的块数目
CMD24	adtc	[31: 0]=数据地址	R1	WRITE_BLOCK	写入由 CMD16 设置大小的数据块
CMD25	adtc	[31: 0]=数据地址	R1	WRITE_MULTIPLE_BLOCK	连续写入数据块, 直到收到 STOP_TRANSMISSION 命令
CMD26	adtc	[31: 0]=填充位	R1	PROGRAM_CID	对卡识别寄存器进行编程
CMD27	adtc	[31: 0]=填充位	R1	PROGRAM_CSD	对 CSD 的可编程位编程

表 23-6基于块传输的写保护命令

CMD 索引	类型	参数	响应格式	缩写	说明
CMD28	ac	[31: 0]=数据地址	R1b	SET_WRITE_PROT	如果卡具有写保护的功能, 该命令设置指定组的写保护位。写保护的属性设置在卡的特定数据域 (WP_GRP_SIZE)。
CMD29	ac	[31: 0]=数据地址	R1b	CLR_WRITE_PROT	如果卡具有写保护的功能, 该命令清除指定组的写保护位。
CMD30	adtc	[31: 0]=写保护数据地址	R1	SEND_WRITE_PROT	如果卡具有写保护的功能, 该命令要求卡发送写保护位的状态。

表 23-7擦除命令

CMD 索引	类型	参数	响应格式	缩写	说明
CMD32 ... CMD34		保留。为了与旧版本的对媒体卡协议向后兼容, 不能使用这些命令代码。			
CMD35	ac	[31: 0]=数据地址	R1	ERASE_GROUP_START	在选择的擦除范围内, 设置第一个擦除组的地址。
CMD36	ac	[31: 0]=数据地址	R1	ERASE_GROUP_END	在选择的连续擦除范围内, 设置最后一个擦除组的地址。
CMD37		保留。为了与旧版本的对媒体卡协议向后兼容, 不能使用这个命令代码。			
CMD38	ac	[31: 0]=填充位	R1b	ERASE	擦除之前选择的数据块。

表 23-8 I/O模式命令

CMD 索引	类型	参数	响应格式	缩写	说明
CMD39	ac	[31: 16]=RCA [15]=寄存器写标志 [14: 8]=寄存器地址 [7: 0]=寄存器数据	R4	FAST_IO	用于写入和读取 8 位（寄存器）数据域。该命令指定一个卡和寄存器，如果设置了写标志还提供写入的数据。R4 响应包含从指定寄存器读出的数据。该命令访问未多媒体卡标准中定义的与应用相关的寄存器。
CMD40	bcr	[31: 0]=填充位	R5	GO_IRQ_STATE	置系统于中断模式。

表 23-9 卡锁定命令

CMD 索引	类型	参数	响应格式	缩写	说明
CMD42	adtc	[31: 0]=填充位	R1	LOCK_UNLOCK	设置/清除密码，又或是对卡锁定/解锁，数据块的长度由 CMD16 定义

表 23-10 应用相关命令

CMD 索引	类型	参数	响应格式	缩写	说明
CMD55	ac	[31: 16]=RCA [15: 0]=填充位	R1	APP_CMD	指示卡下一个命令是应用相关命令而不是一个标准命令。
CMD56	adtc	[31: 1]=填充位 [0]=RD/WR	R1	GEN_CMD	在通用或应用相关命令中，或者用于向卡中传输一个数据块，或者用于从卡中读取一个数据块。数据块的长度由 SET_BLOCK_LEN 命令设置。
CMD57 ... CMD59	保留。				
CMD60 ... CMD63	保留给生产厂商。				

### 23.3.2.2 响应格式

所有的响应都是通过 CMD 总线发送，响应传输总是从与响应对应字相对应的位串的左位开始，响应字的长度取决于响应类型。

响应总是以起始位(始终为 0)开始，然后是指示传输方向的传输位(卡=0)，下表中标示为 - 的数值表示为可变的，除了 R3 响应类型外，所有的响应均受 CRC 保护，每个命令码字都以结束位(始终为 1)终止。

#### 23.3.2.2.1 R1（普通响应命令）

编码长度为 48 位，位 45：40 指示指示要响应的命令的索引，该值被解释为二进制编码的数字（介于 0 和 63 之间）。卡的状态以 32 位编码。请注意，如果涉及到向卡的数据传输，则在传输每个数据块后，数据在线可能会出现繁忙信号。数据块传输后，主机应检查是否忙碌。

表 23-11 R1 响应

位	47	46	[45: 40]	[39: 8]	[7: 1]	0
域宽度	1	1	6	32	7	1
数值	0	0	-	-	-	1
说明	开始位	传输位	命令索引	卡状态	CRC7	结束位

**23.3.2.2.2 R1b**

R1b 与 R1 相同，只是在数据在线传输了可选的忙信号。根据命令接收之前的状态，卡在接收到这些命令后可能会变得很忙。主机应检查响应是否忙碌

**23.3.2.2.3 R2(CID、CSD寄存器)**

编码长度为 136 位，CID 寄存器的内容作为对命令 CMD2 和 CMD10 的响应发送。CSD 寄存器的内容作为对 CMD9 的响应发送。仅传送 CID 和 CSD 的位[127 ... 1]，这些寄存器的保留位[0]被响应的结束位替换。

表 23-12 R2响应

位	135	134	[133 : 128]	[127 : 1]	0
域宽度	1	1	6	127	1
数值	1	0	111111	-	1
说明	开始位	传输位	保留	CID 或 CSD 寄存器	结束位

**23.3.2.2.4 R3(OCR寄存器)**

编码长度为 48 位，该 OCR 寄存器内容作为 ACMD41 的响应被发送。

表 23-13 R3响应

位	47	46	[45: 40]	[39: 8]	[7: 1]	0
域宽度	1	1	6	32	7	1
数值	1	0	111111	-	111111	1
说明	开始位	传输位	保留	OCR 寄存器	保留	结束位

**23.3.2.2.5 R4(快速I/O)**

编码长度为 48 位，参数域包含指定卡的 RCA、需要读出或写入寄存器的地址、和它的内容。

表 23-14 R4响应

位	47	46	[45: 40]		[39: 8]		[7: 1]	0
域宽度	1	1	6	16	8	8	7	1
数值	1	0	100111	-	-	-	-	1
说明	开始位	传输位	CMD39	RCA	寄存器地址	读寄存器的内容	CRC7	结束位

**23.3.2.2.6 R4b**

仅适合 SD I/O 卡：一个 SDIO 卡收到 CMD5 后将返回一个唯一的 SDIO 响应 R4

表 23-15 R4b响应

位	47	46	[45: 40]		[39: 8]			[7: 1]		0
域宽度	1	1	6	1	3	1	3	24	7	1
数值	1	0	-	-	-	-	-	-	-	1
说明	开始位	传输位	保留	卡就绪	I/O 功能数目	当前寄存器	填充位	I/O OCR	保留	结束位

**23.3.2.2.7 R5(中断请求)**

仅适用于多媒体卡。代码长度=48 位。如果这个响应由主机产生，则参数中的 RCA 域为 0x0。

表 23-16 R5响应

位	47	46	[45: 40]	[39: 8]	[7: 1]	0	
域宽度	1	1	6	16	16	7	1
数值	1	0	101000	-	-	-	1
说明	开始位	传输位	CMD40	成功的卡或主机的 RCA[31: 16]	未定义可以 作为中断数据。	CRC7	结束位

**23.3.2.2.8 R6(中断请求)**

仅适用于 SD I/O 卡。这是一个存储器设备对 CMD3 命令的正常响应

表 23-17 R6响应

位	47	46	[45: 40]	[39: 8]	[7: 1]	0	
域宽度	1	1	6	16	16	7	1
数值	1	0	000011	-	-	-	1
说明	开始位	传输位	CMD3	成功的卡或主机的 RCA[31: 16]	卡状态	CRC7	结束位

当发送 CMD3 命令到只有 I/O 功能的卡时，卡的状态位[23: 8]会改变；此时，响应中的 16 位将是只有 I/O 功能的 SD 卡中的数值：

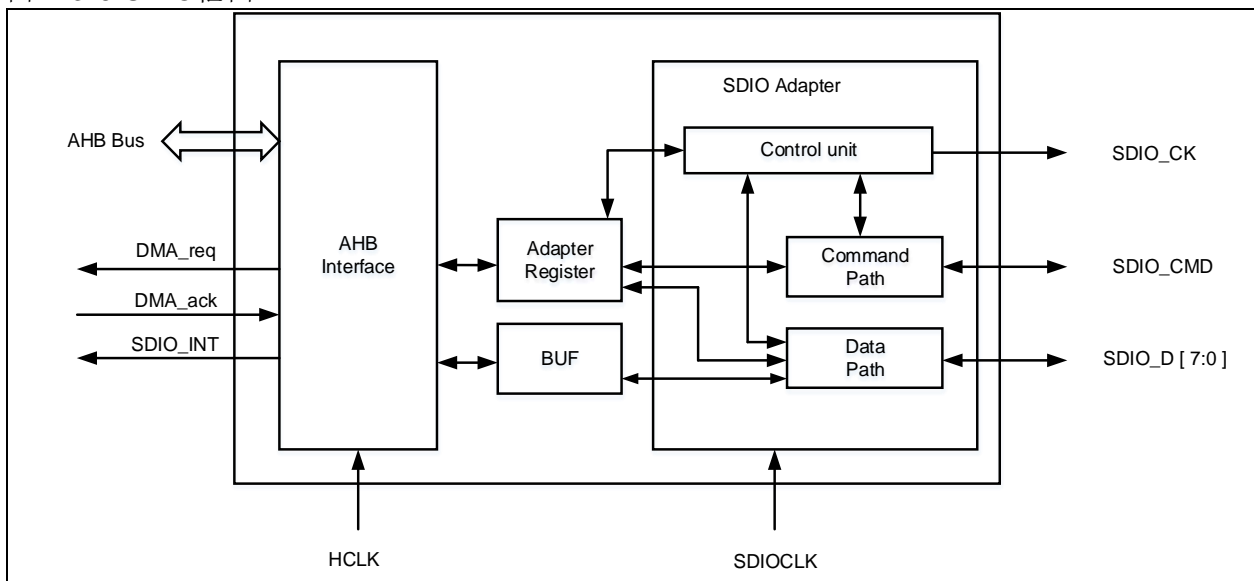
- 位 15=COM\_CRC\_ERROR
- 位 14=ILLEGAL\_COMMAND
- 位 13=ERROR
- 位[12: 0]=保留

**23.3.3 SDIO功能描述**

SDIO 包含 4 个部分：

- SDIO 适配器模块：由控制单元、命令单元和数据单元所组成，实现所有 MMC/SD/SD I/O 卡的相关功能，如时钟的产生、命令和数据的传送
  - 控制单元：管理并产生时钟信号
  - 命令单元：管理命令的传输
  - 数据单元：管理数据的传输
- AHB 接口：产生中断和 DMA 请求信号
- 适配器寄存器：系统寄存器
- BUF：用于数据传输

图 23-6 SDIO框图



### 23.3.3.1 SDIO适配器

**SDIO\_CK** 是主机给多媒体/SD/SDIO 卡的时钟，每个时钟周期再命令和数据在线传输 1 位命令或数据，不同的卡和协议之间有不同的时钟频率限制

- 多媒体卡
  - V3.31协议 0 – 20MHz
  - V4.0/4.2协议 0 – 50MHz
- SD 卡
  - 0 – 50MHz
- SD I/O 卡
  - 0 – 50MHz

**SDIO\_CMD** 信号是双向命令通道，用于卡的初始化和命令传输，主机发送命令至卡端后，卡送出响应至主机端，**SDIO\_CMD** 信号有两种操作模式：

- 用于初始化时的开路模式（仅用于 MMC 版本 V3.31 或之前版本）
- 用于命令传输的推挽模式（SD/SD I/O 卡和 MMC V4.2 在初始化时也使用推挽驱动）

**SDIO\_D [7:0]** 信号是双向数据信道，初始化后主机可以改变数据总线的宽度，在复位后默认情况下 **SDIO\_D0** 用于数据传输，MMC 卡在 V3.31 和之前版本的协议只支持一位数据线，只能使用 **SDIO\_D0**。

下表适用于多媒体卡/SD/SD I/O 卡总线：

表 23-18 SDIO管脚定义

管脚	方向	说明
SDIO_CK	输出	多媒体卡/SD/SDIO 卡时钟。这是从主机至卡的时钟线。
SDIO_CMD	双向	多媒体卡/SD/SDIO 卡命令。这是双向的命令/响应信号线。
SDIO_D[7: 0]	双向	多媒体卡/SD/SDIO 卡数据。这些是双向的数据总线。

#### 控制单元

控制单元包含电源和时钟管理功能，电源管理的部分主要由 **SDIO\_PWRCTRL** 寄存器控制，**PS** 位控制上下电，在电源关闭和电源启动阶段，电源管理子单元会关闭卡总线上的输出信号，时钟管理则是由 **SDIO** 时钟控制寄存器（**SDIO\_CLKCTRL**）控制，**CLKDIV** 位定义了 **SDIO** 时钟(**SDIOCLK**)与 **SDIO** 输出至卡端的时钟(**SDIO\_CK**)间的分频系数关系，若 **BYPSEN** 位为 0，**SDIO\_CK** 输出信号由 **SDIOCLK** 依据 **CLKDIV** 位分频后驱动，若 **BYPSEN** 位被置 1，则 **SDIO\_CL** 输出信号直接由 **SDIOCLK** 驱动，将 **HFCEN** 位置 1 可以开启硬件流控制功能，避免在发送模式出现下溢和接收模式出现上溢的错误，软件可通过设置 **PWRSVEN** 位开启省电模式，仅有在总线活动时才会输出 **SDIO\_CK**。

#### 命令通道

命令通道负责向卡发送和接收命令，将 **SDIO\_CMDCTRL** 寄存器的 **CCSMEN** 位置 1 后，命令传输开始，

首先向卡发送一个命令，这个命令共 48 位，通过 SDIO\_CMD 发出，SDIO\_CMD 上的数据与 SDIO\_CK 的上升沿同步，每个 SDIO\_CK 传输一笔数据，包含开始位、传输位、由 SDIO\_CMDCTRL\_CMDIDX 位定义的命令索引、SDIO 参数寄存器 (SDIO\_ARG) 定义的参数、7 位的 CRC 和停止位，然后接收卡端的响应，响应可分为 48 位的短响应和 136 位的长响应，2 种类型都有 CRC 错误检测，收到的响应回存在 SDIO\_RSP1 到 SDIO\_RSP4 中，命令通道可以产生命令状态标志并由 SDIO 状态寄存器 (SDIO\_STS) 定义。

表 23-19 命令格式

位	47	46	[45 : 40]	[39 : 8]	[7 : 1]	0
宽度	1	1	6	32	7	1
数值	0	1	-	-	-	1
说明	开始位	传输位	命令索引	参数	CRC7	结束位

— 响应：响应是由一个被指定地址的卡发送到主机，对于 MMCV3.31 或以前版本所有的卡同时发送响应；响应是对先前接收到命令的一个应答。响应在 CMD 线上串行传送。

表 23-20 短响应格式

位	47	46	[45 : 40]	[39 : 8]	[7 : 1]	0
宽度	1	1	6	32	7	1
数值	0	0	-	-	-	1
说明	开始位	传输位	命令索引	参数	CRC7(或 1111111)	结束位

表 23-21 长响应格式

位	135	134	[133: 128]	[127 : 1]	0
宽度	1	1	6	127	1
数值	0	0	111111	-	1
说明	开始位	传输位	保留	CID 或 CSD(包含内部 CRC7)	结束位

表 23-22 命令通道状态标志

标志	说明
CMDRSPCMPL	已接受到响应(CRC 检测成功)
CMDFAIL	已收到命令响应(CRC 检测失败)
CMDCMPL	命令（不需要响应的命令）已发送
CMDTIMEOUT	命令响应超时(64 个 SDIO_CK 时钟周期)
DOCMD	正在发送命令

#### 命令通道状态机 (CCSM)

当设置 SDIO\_CMDCTRL 寄存器的 CCSMEN 位，控制器开始发送命令。命令发送完成时，命令通道状态机 (CCSM) 设置命令通道状态标志并在不需要响应时进入空闲状态。当收到响应后，接收到的 CRC 码将会与内部产生的 CRC 码比较，然后设置相应的状态标志。

- CCSM 在空闲状态至少保持 8 个 SDIO\_CK 周期以满足  $N_{cc}$  (两个主机命令之间的最小时间间隔) 和  $N_{rc}$  (主机命令与卡响应之间的最小时间间隔) 的时序限制，
- 在发送完成后进入等待状态时，会启动命令通道内的定时器，若进入接收状态前违反  $N_{cr}$  (指令响应时间) 时序，超过了 64 个 SDIO\_CK 的时间，会设置超时标志

(CMDTIMEOUT)并回到空闲状态。

如果在命令寄存器设置了中断位，则关闭定时器，CCSM 等待某一个卡发出的中断请求。如果命令寄存器中设置挂起位，CCSM 进入挂起（Pend）状态并等待数据通道子单元发出的 CmdPend 信号，在检测到 CmdPend 信号时，CCSM 进入发送（Send）状态，这将触发数据计数器发送停止命令的功能。

图 23-7 命令通道状态机 (CCSM)

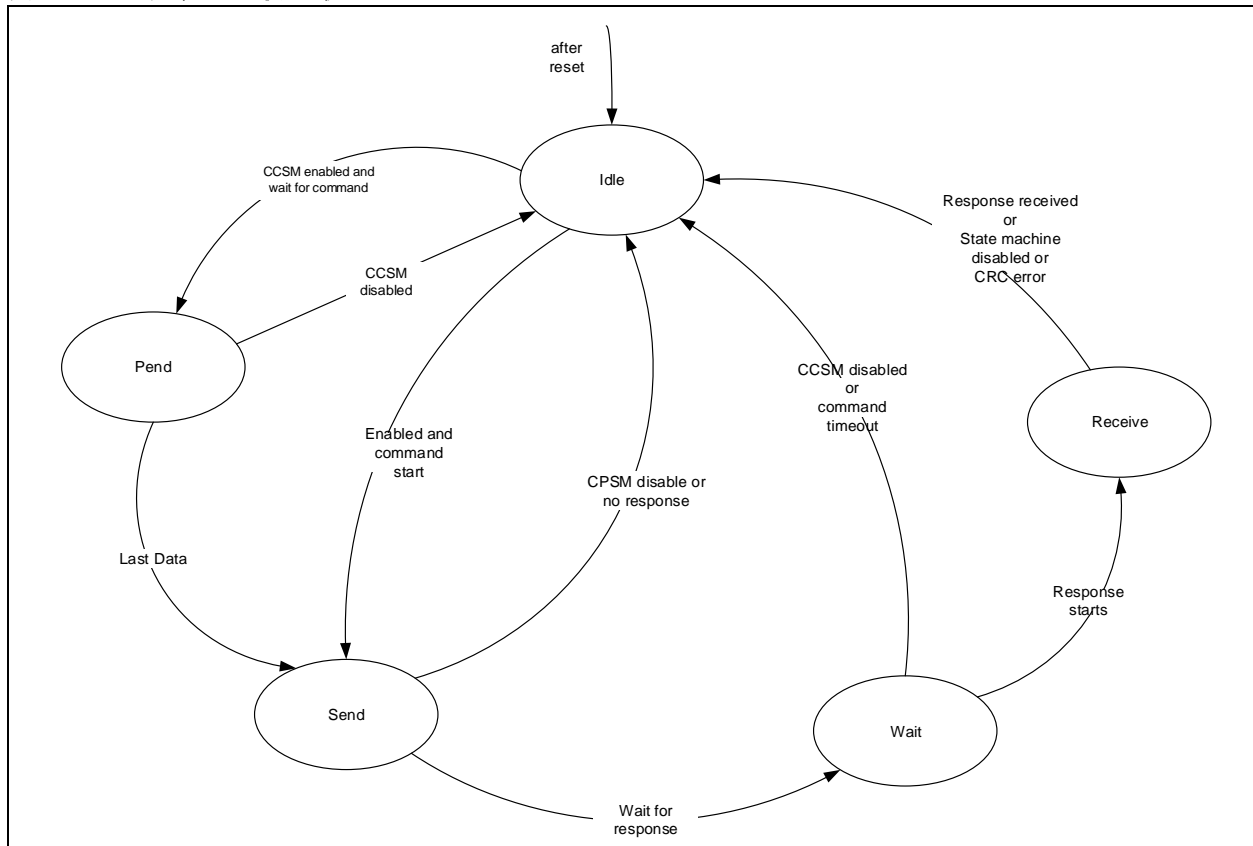
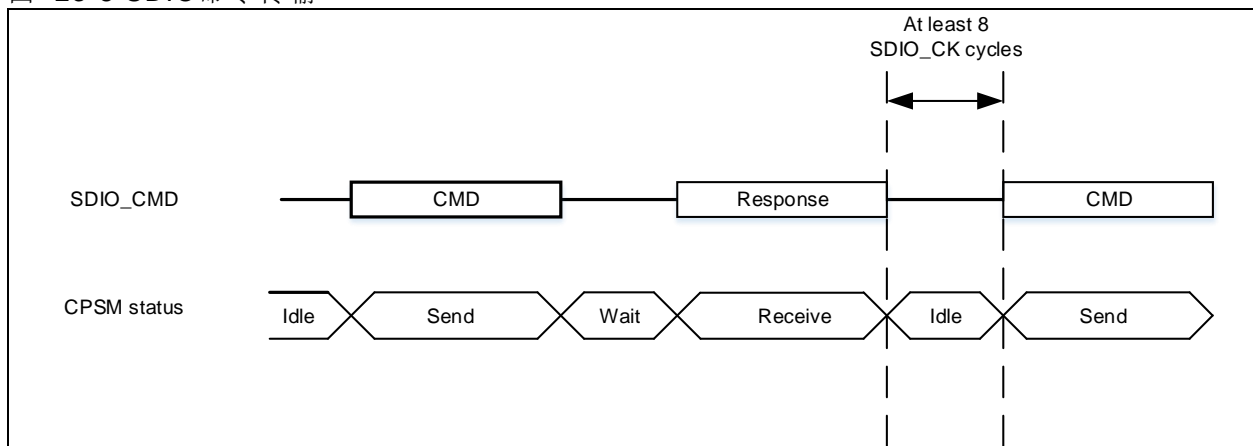


图 23-8 SDIO命令传输



## 数据通道

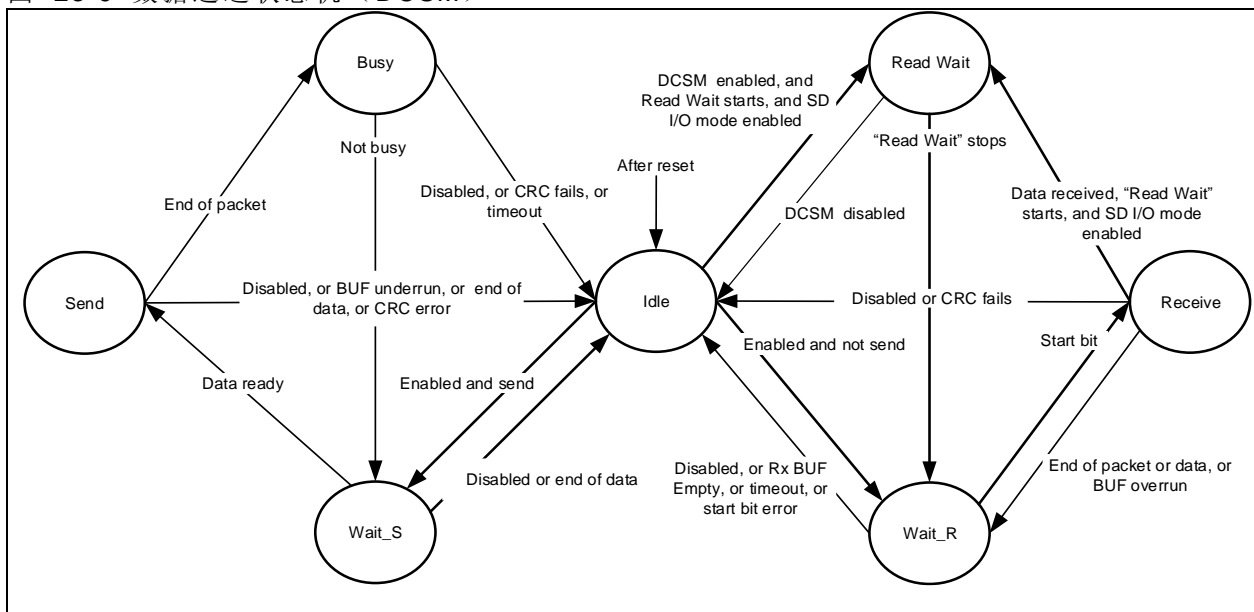
数据通道负责实现主机与卡之间数据传输，可以透过设定 SDIO 时钟控制寄存器 (SDIO\_CLKCTRL) 的 BUSWS 位选择数据总线宽度，默认情况下只会使用 SDIO\_D0 信号线传输，每个时钟周期传输 1 位的数据，可以选择设定 4 位总线模式，每个时钟周期传输 4 位数据并使用 SDIO\_D[3:0] 信号线，若设定 8 位总线模式，每个时钟周期传输 8 位数据并使用 SDIO\_D[7:0] 信号线，设定 SDIO\_DTCTRL 寄存器的 TFRDIR 位可以传输方向，当 TFRDIR 位为 0 时表示传输方向是从控制器至卡端，若为 1 则表示传输方向是从卡至控制器，TFRMODE 位可配合多媒体卡选择块数据传输或是流数据传输，如果将 TFREN 位置 1，则开始传输数据，根据 TFRDIR 位决定传输的方向(发送或接收)，使能时数据通道状态机(DCSM)将进入 Wait S 或 Wait R 状态。



## 数据通道状态机 (DCSM)

DCSM 有 7 个状态，可分为发送和接收模式来看，如下图所示：

图 23-9 数据通道状态机 (DCSM)



## 发送模式

- 空闲 (Idle)：数据通道不工作，等待发送(进入 Wait\_S 状态)或接收数据(进入 Wait\_R 状态)
- 等待发送 (Wait\_S)：等待数据 Buf 为空标志或是数据传输结束，须至少保持两个时钟周期，以满足  $N_{WR}$  的时序要求， $N_{WR}$  是接收到卡的响应置主机开始传输数据的间隔。
- 发送 (Send)：发送数据到卡，根据 `SDIO_DTCTRL_TFRMODE` 位决定是块数据或是流数据传输，若发生下溢错误则会回到空闲状态。
- 繁忙 (Busy)：等待 CRC 标志，若接收正确且卡不繁忙时回到 Wait\_S 状态，若不正确或是繁忙状态超时则回到空闲状态，并产生 CRC 失败标志或是超时标志。
- 等待接收 (Wait\_R)：等待数据接收的起始位，若在检测到起始位前发生超时错误，则会回到空闲状态并产生超时标志。
- 接收 (Receive)：接收来自卡端的数据并写入数据 Buf，根据 `SDIO_DTCTRL_TFRMODE` 位决定是块数据或是流数据传输，若发生上溢错误则会回到等待接收状态。

表 23-23 数据令牌格式

说明	开始位	数据	CRC16	结束位
块数据	0	-	有	1
流数据	0	-	无	1

## 23.3.3.2 数据 BUF

数据 BUF 是一个具有发送和接收单元，每字 32 位宽、共 32 个字的数据缓冲区，因为数据 BUF 工作在 AHB 时钟区域 (HCLK)，所有与 SDIO 时钟区域 (SDIOCLK) 连接的信号都进行了重新同步。

- 发送 BUF：当使能了 SDIO 的发送功能，数据可以通过 AHB 接口写入发送 BUF。发送 BUF 有 32 个连续的地址。发送 BUF 中有一个数据输出寄存器，包含读指针指向的数据字。当数据通道装填了移位寄存器后，它移动读指针至下个数据并传输出数据。如果未使能发送 BUF，所有的状态标志均处于无效状态。当发送数据时，数据通道设置 DOTX 为有效。
- 接收 BUF：当数据通道接收到一个数据字，它会把数据写入 BUF，写操作结束后，写指针自动加一；在另一端，有一个读指针始终指向 BUF 中的当前数据。如果关闭了接收 BUF，所有的状态标志会被清除，读写指针也被复位。在接收到数据时数据通道设置 DORX。

### 23.3.3.3 SDIO AHB接口

AHB 接口产生中断和 DMA 请求，并访问 SDIO 接口寄存器和数据 BUF。

#### SDIO 中断

当有任一个选中的状态标志为高时，中断控制逻辑产生中断请求。SDIO 中断屏蔽寄存器 (SDIO\_INTEN) 可以选择产生中断的条件。

#### SDIO/DMA 接口：在 SDIO 和存储器之间数据传输的过程

在下面的例子中，从主机传送数据到卡端，DMA 控制器用于从存储器向 SDIO 的 BUF 填充数据。

1. 卡识别过程
2. 提高SDIO\_CK频率
3. 发送CMD7命令选择卡
4. 使能DMA2控制器并清除所有的中断标志位，设置DMA2信道4的源地址寄存器为存储器缓冲区的基地址，DMA2信道4的目标地址寄存器为SDIO\_BUF寄存器的地址，接着设置DMA2通道4控制寄存器（存储器递增，非外设递增，外设和源的数据宽度为字宽度），最后使能DMA2通道4。
5. 发送CMD24 (WRITE\_BLOCK)，操作如下：  
设置 SDIO 数据长度寄存器 (SDIO\_DTLLEN)，接着设置 SDIO 数据控制寄存器 (SDIO\_DTCTRL) 的 BLKSIZE 位，之后设置 SDIO 参数寄存器 (SDIO\_ARG) 写入需要传送数据的地址，配置 SDIO 命令寄存器 (SDIO\_CMD)，使能 CCSMEN 位，等待 SDIO 状态寄存器 (SDIO\_STS) [6]=CMDRSPCMPL 中断，然后设置 SDIO 数据控制寄存器 (SDIO\_DTCTRL)：TFREN 置为 1 (使能 SDIO 卡主机发送数据)；TFRDIR 置为 0 (控制器至卡方向)；TFRMODE 置为 0 (块数据传送)；DMAEN 置为 1 (DMA 使能)；BLKSIZE 置为 9 (512 字节)，等待 SDIO 状态寄存器 (SDIO\_STS) [10]=DTBLKCMPL。
6. 查询SDIO已使能DMA通道的状态寄存器，确认没有通道仍处于传输状态

### 23.3.3.4 硬件流控制

可以透过设置 SDIO 时钟控制寄存器 (SDIO\_CLKCTRL) 的 HFCEN 位开启功能，避免 BUF 下溢和上溢的错误，在流控制功能开启时仍进行读出或写入 BUF 操作。

### 23.3.4 SDIO I/O卡特定的操作

当设置了 SDIO 数据控制寄存器 (SDIO\_DTCTRL) [11]位时，SDIO 可支持这些操作(读暂停除外，因为它不需要特殊的硬件操作)

#### 由 SDIO\_D2 信号线实现的 SDIO 读等待操作

可选的读等待 (RW) 操作只适用于 SD 卡的 1 位或 4 位模式，读等待操作允许主机正在读多个寄存器 (IO\_RW\_EXTENDED, CMD53) 时，要求它暂时停止数据传输，同时允许主机发送命令到 SD I/O 设备中的其他功能，以收到第一个数据块之前即可以开始读等待过程，下描述详细过程。

- 使能数据通道 (SDIO 数据控制寄存器 (SDIO\_DTCTRL) [0] = 1)
- 使能 SDIO 特定操作 (SDIO 数据控制寄存器 (SDIO\_DTCTRL) [11] = 1)
- 开始读等待 (SDIO 数据控制寄存器 (SDIO\_DTCTRL) [10]=0 且 SDIO 数据控制寄存器 (SDIO\_DTCTRL) [8]=1)
- 数据传输方向是从卡至 SDIO 主机 (SDIO 数据控制寄存器 (SDIO\_DTCTRL) [1]=1)
- SDIO 适配器的数据单元将进入读等待状态，待 2 个 SDIO\_CK 后驱动 SDIO\_D2 为 '0'
- 数据单元开始等待从卡里接收数据。在接收数据块时，即使设置了开始读等待，DCSM 也不会进入读等待，读等待过程将在收到 CRC 后开始。必须清除 RDWTSTOP 才能开始新的读等待操作。

在读等待期间，SDIO 主机可以在 SDIO\_D1 上监测 SDIO 中断。

#### 通过停止时钟实现的 SDIO 读等待操作

如果 SDIO 卡不能支持前述的读等待操作，SDIO 可以停止 SDIO\_CK 进入读等待，详细操作如下：

- 使能数据通道 (SDIO 数据控制寄存器 (SDIO\_DTCTRL) [0] = 1)
- 使能 SDIO 特定操作 (SDIO 数据控制寄存器 (SDIO\_DTCTRL) [11] = 1)
- 开始读等待 (SDIO 数据控制寄存器 (SDIO\_DTCTRL) [10]=且 SDIO 数据控制寄存器 (SDIO\_DTCTRL) [8]=1)

在接收当前数据块结束位之后的 2 个 SDIO\_CK 周期后，DCSM 停止时钟，在设置了读等待结束位后恢复时钟。

需注意因为 SDIO\_CK 停止，SDIO 主机不可以向卡发送任何命令，并且 SDIO 主机可以在 SDIO\_D1 上监测 SDIO 中断

#### SDIO 暂停/恢复操作（写和读暂停）

为了给其他的功能或者存储器提供更高优先级的传输而释放总线，主机可以暂停某个功能或者存储器的数据传输。一旦高优先级的传输完成后，原来的传输在暂停处重新开始。

在向卡发送数据时，SDIO 可以暂停写操作。设置 SDIO 命令寄存器（SDIO\_CMD）[11]位并指示 CCSM 当前的命令是一个暂停命令。CCSM 分析响应，在从卡收到响应时（暂停被接受），它确认在收到当前数据块的 CRC 后 DCSM 进入空闲状态，而暂停读操作的部分，CCSM 会在 Wait\_R 状态等待，若再暂停前已发送数据，应用程序会继续读出 BUF 直到 BUF 为空，之后 CCSM 进入 IDLE 状态。

#### SDIO 中断

为了让 SD I/O 卡能够中断 SDIO 模块，在 SD 接口上有一个具有中断功能的管脚，在 4 位 SD 模式下这个脚是 SDIO\_D1，SD I/O 的中断是电平有效，即在识别并得到 SDIO 模块的响应之前，中断信号线必须保持有效电平（低），在中断过程结束后保持无效电平（高）。

当设置了 SDIO 数据控制寄存器（SDIO\_DTCTRL）[11]位，SDIO 主机在 SDIO\_D1 信号线上监测 SDIO 中断。

## 23.4 SDIO 寄存器

设备可以通过在 AHB 上操作的 32 位控制寄存器与系统通信。

必须以字（32 位）的方式操作这些外设寄存器。

下表是 SDIO 寄存器的总结。

表 23-24 SDIO 寄存器映像

寄存器简称	基址偏移量	复位值
SDIO_PWRCTRL	0x00	0x0000 0000
SDIO_CLKCTRL	0x04	0x0000 0000
SDIO_ARG	0x08	0x0000 0000
SDIO_CMD	0x0C	0x0000 0000
SDIO_RSPCMD	0x10	0x0000 0000
SDIO_RSP1	0x14	0x0000 0000
SDIO_RSP2	0x18	0x0000 0000
SDIO_RSP3	0x1C	0x0000 0000
SDIO_RSP4	0x20	0x0000 0000
SDIO_DTTMR	0x24	0x0000 0000
SDIO_DTLEN	0x28	0x0000 0000
SDIO_DTCTRL	0x2C	0x0000 0000
SDIO_DTCNTR	0x30	0x0000 0000
SDIO_STS	0x34	0x0000 0000
SDIO_INTCLR	0x38	0x0000 0000
SDIO_INTEN	0x3C	0x0000 0000
SDIO_BUFCNTR	0x48	0x0000 0000
SDIO_BUF	0x80	0x0000 0000

## 23.4.1 SDIO电源控制寄存器（SDIO\_PWRCTRL）

域	简称	复位值	类型	功能
位 31: 2	保留	0x0000 0000	resd	保持默认值。
位 1: 0	PS	0x0	rw	电源开关位（Power switch） 由软件置起或清零。该位用于定义卡时钟的当前状态。 00: 关闭，卡的时钟停止； 01: 保留； 10: 保留； 11: 开启，卡的时钟开启。

注意：写数据后的 7 个 HCLK 时钟周期内，不能写入这个寄存器。

## 23.4.2 SDIO时钟控制寄存器（SDIO\_CLKCTRL）

SDIO 时钟控制寄存器（SDIO\_CLKCTRL）控制 SDIO\_CK 输出时钟。

域	简称	复位值	类型	功能
位 31: 17	保留	0x0000	resd	保持默认值。
位 16: 15	CLKDIV	0x0	rw	时钟分频系数（Clock division） 由软件置起或清零。该位定义了 SDIO 时钟（SDIOCLK）与 SDIO 总线时钟（SDIO_CK）间的分频系数关系： $SDIO\_CK \text{ 频率} = SDIOCLK / [CLKDIV[9: 0] + 2]$ 。
位 14	HFCEN	0x0	rw	硬件流控制使能（Hardware flow control enable） 由软件置起或清零。 0: 关闭； 1: 开启。 注：当开启硬件流控制后，关于 TXBUF_E 和 RXBUF_F 中断信号的意义请参考 SDIO 状态寄存器（SDIO_STS）的定义。
位 13	CLKEGS	0x0	rw	SDIO_CK 边沿选择（SDIO_CK edge selection） 由软件置起或清零。 0: 在主时钟 SDIOCLK 上升沿产生 SDIO_CK； 1: 在主时钟 SDIOCLK 下降沿产生 SDIO_CK。
位 12: 11	BUSWS	0x0	rw	总线宽度选择（bus width selection） 由软件置起或清零。 00: 默认总线模式，使用 SDIO_D0； 01: 4 位总线模式，使用 SDIO_D[3: 0]； 10: 8 位总线模式，使用 SDIO_D[7: 0]。
位 10	BYPSEN	0x0	rw	旁路时钟分频器（Clock divider bypass enable bit） 由软件置起或清零。关闭表示 SDIO_CK 输出信号由 SDIOCLK 依据 CLKDIV 数值分频后驱动，开启表示 SDIO_CK 输出信号直接由 SDIOCLK 驱动。 0: 关闭； 1: 开启。
位 9	PWRSVEN	0x0	rw	省电模式使能（Power saving mode enable） 由软件置起或清零。关闭表示始终都会输出 SDIO_CK，开启表示仅在总线活动时才会输出 SDIO_CK。 0: 关闭； 1: 开启。
位 8	CLKOEN	0x0	rw	时钟输出使能（Clock output enable） 由软件置起或清零。 0: 关闭； 1: 开启。
位 7: 0	CLKDIV	0x00	rw	时钟分频系数（Clock division） 由软件置起或清零。该位定义了 SDIO 时钟（SDIOCLK）与 SDIO 总线时钟（SDIO_CK）间的分频系数关系： $SDIO\_CK \text{ 频率} = SDIOCLK / [CLKDIV[9: 0] + 2]$ 。

注意：1. 当 SD/SDIO 卡或多媒体卡在识别模式，SDIO\_CK 的频率必须低于 400kHz。

2. 当所有卡都被赋予了相应的地址后，时钟频率可以改变到卡总线允许的最大频率。

3.写数据后的 7 个 HCLK 时钟周期内不能写入这个寄存器。对于 SD I/O 卡，在读等待期间可以停止 SDIO\_CK，此时 SDIO 时钟控制寄存器（SDIO\_CLKCTRL）不控制 SDIO\_CK。

### 23.4.3 SDIO参数寄存器（SDIO\_ARG）

SDIO 参数寄存器（SDIO\_ARG）包含 32 位命令参数，它将作为命令的一部分发送到卡中。

域	简称	复位值	类型	功能
位 31: 0	ARGU	0x0000 0000	rw	命令参数（Command argument） 命令参数是发送到卡中命令的一部分，如果一个命令包含一个参数，必须在写命令到命令寄存器之前加载这个寄存器。

### 23.4.4 SDIO命令寄存器（SDIO\_CMD）

SDIO 命令寄存器（SDIO\_CMD）包含命令索引和命令类型位。命令索引是作为命令的一部分发送到卡中。命令类型位控制命令通道状态机（CPSM）。

域	简称	复位值	类型	功能
位 31: 12	保留	0x00000	resd	保持默认值。
位 11	IOSUSP	0x0	rw	SD I/O 暂停命令（SD I/O suspend command） 由软件置起或清零。如果该位被置起，则将要发送的命令是一个暂停命令（只能用于 SDIO 卡）。 0：关闭； 1：开启。
位 10	CCSMEN	0x0	rw	命令通道状态机使能（Command channel state machine（CCSM） enable bit） 由软件置起或清零。 0：关闭； 1：开启。
位 9	PNDWT	0x0	rw	CCSM 等待数据传输结束（CmdPend 内部信号）（CCSM Waits for ends of data transfer（CmdPend internal signal）） 由软件置起或清零。如果该位被置起，则 CCSM 在开始发送一个命令之前会等待数据传输结束。 0：关闭； 1：开启。
位 8	INTWT	0x0	rw	CCSM 等待中断请求（CCSM waits for interrupt request） 由软件置起或清零。如果该位被置起，则 CCSM 关闭命令超时控制并等待中断请求。 0：关闭； 1：开启。
位 7: 6	RSPWT	0x0	rw	等待响应位（Wait for response bits） 由软件置起或清零。该位指示 CCSM 是否需要等待响应，如果需要等待响应，则指示响应类型。 00：无响应； 01：短响应； 10：无响应； 11：长响应。
位 5: 0	CMDIDX	0x00	rw	命令索引（Command index） 命令索引是作为命令的一部分发送到卡中。

注意：1.写数据后的 7 个 HCLK 时钟周期内不能写入这个寄存器。

2.多媒体卡可以发送 2 种响应：48 位长的短响应，或 136 位长的长响应。SD 卡和 SD I/O 卡只能发送短响应，参数可以根据响应的类型而变化，软件将根据发送的命令区分响应的类型。



### 23.4.5 SDIO命令响应寄存器（SDIO\_RSPCMD）

SDIO 命令响应寄存器（SDIO\_RSPCMD）包含最后收到的命令响应中的命令索引。如果传输的命令响应不包含命令索引（长响应或 OCR 响应），尽管它应该包含 111111b（响应中的保留域值），但 RSPCMD 域的内容未知。

域	简称	复位值	类型	功能
位 31: 6	保留	0x00000000	resd	保持默认值。
位 5: 0	RSPCMD	0x00	ro	响应的命令索引（Response command index） 收到的命令响应中的命令索引。

### 23.4.6 SDIO响应1..4寄存器（SDIO\_RSPx）

SDIO 响应 1..4 寄存器（SDIO\_RSPx）包含卡的状态，即收到响应的部分信息。

域	简称	复位值	类型	功能
位 31: 0	CARDSTSx	0x0000 0000	ro	见下表

根据响应状态，卡的状态长度是 32 位或 127 位。

表 23-25 响应类型和 SDIO\_RSPx 寄存器

寄存器	短响应	长响应
SDIO_RSP1	卡状态[31: 0]	卡状态[127: 96]
SDIO_RSP2	不用	卡状态[95: 64]
SDIO_RSP3	不用	卡状态[63: 32]
SDIO_RSP4	不用	卡状态[31: 1]

总是先收到卡状态的最高位，SDIO\_RSP4 寄存器的最低位始终为 0。

### 23.4.7 SDIO数据定时器寄存器（SDIO\_DTTMR）

SDIO 数据定时器寄存器（SDIO\_DTTMR）包含以卡总线时钟周期为单位的数据超时时间。

一个计数器从 SDIO 数据定时器寄存器（SDIO\_DTTMR）加载数值，并在数据通道状态机（DCSM）进入 Wait\_R 或繁忙状态时进行递减计数，当 DCSM 处在这些状态时，如果计数器减为 0，则设置超时标志。

域	简称	复位值	类型	功能
位 31: 0	TIMEOUT	0x0000 0000	rw	数据超时时间（Data timeout period） 以卡总线时钟周期为单位的数据超时时间。

注意：在写入 SDIO 数据控制寄存器（SDIO\_DTCTRL）进行数据传输之前，必须先写入 SDIO 数据计数器寄存器（SDIO\_DTCNTR）和 SDIO 数据长度寄存器（SDIO\_DTLEN）。

### 23.4.8 SDIO数据长度寄存器（SDIO\_DTLEN）

SDIO 数据长度寄存器（SDIO\_DTLEN）包含需要传输的数据字节长度。当数据传输开始时，这个数值被加载到数据计数器中。

域	简称	复位值	类型	功能
位 31: 25	保留	0x00	resd	保持默认值。
位 24: 0	DTLEN	0x00000000	rw	数据长度（Data length value） 要传输的数据字节数目。

注意：对于块数据传输，SDIO 数据长度寄存器（SDIO\_DTLEN）中的数值必须是数据块长度的倍数。在写入 SDIO 数据控制寄存器（SDIO\_DTCTRL）进行数据传输之前，必须先写入 SDIO 数据定时器寄存器（SDIO\_DTTMR）和 SDIO 数据长度寄存器（SDIO\_DTLEN）。

### 23.4.9 SDIO数据控制寄存器（SDIO\_DTCTRL）

SDIO 数据控制寄存器（SDIO\_DTCTRL）控制数据通道状态机（DCSM）。

域	简称	复位值	类型	功能
位 31: 12	保留	0x00000	resd	保持默认值。
位 11	IOEN	0x0	rw	SD I/O 使能功能（SD I/O enable functions） 由软件置起或清零。如果该位被置起，则 DCSM 执行 SD I/O 卡特定的操作。 0: 关闭; 1: 开启。
位 10	RDWTMODE	0x0	rw	读等待模式（Read wait mode） 由软件置起或清零。关闭表示使用 SDIO_D2 控制读等待，开始表示使用 SDIO_CK 控制读等待。 0: 关闭; 1: 开启。
位 9	RDWTSTOP	0x0	rw	读等待停止（Read wait stop） 由软件置起或清零。如果设置了 RDWTSTART，关闭表示执行读等待，开启表示关闭读等待。 0: 关闭; 1: 开启。
位 8	RDWTSTART	0x0	rw	读等待开始（Read wait start） 由软件置起或清零。关闭表示无动作，开启表示开始读等待。 0: 关闭; 1: 开启。
位 7: 4	BLKSIZE	0x0	rw	数据块长度（Data block size） 由软件置起或清零。当选择了块数据传输模式，该域定义数据块长度。 0000: 块长度= $2^0$ =1 字节; 0001: 块长度= $2^1$ =2 字节; 0010: 块长度= $2^2$ =4 字节; 0011: 块长度= $2^3$ =8 字节; 0100: 块长度= $2^4$ =16 字节; 0101: 块长度= $2^5$ =32 字节; 0110: 块长度= $2^6$ =64 字节; 0111: 块长度= $2^7$ =128 字节; 1000: 块长度= $2^8$ =256 字节; 1001: 块长度= $2^9$ =512 字节; 1010: 块长度= $2^{10}$ =1024 字节; 1011: 块长度= $2^{11}$ =2048 字节; 1100: 块长度= $2^{12}$ =4096 字节; 1101: 块长度= $2^{13}$ =8192 字节; 1110: 块长度= $2^{14}$ =16384 字节; 1111: 保留。
位 3	DMAEN	0x0	rw	DMA 使能位（DMA enable bit） 由软件置起或清零。 0: 关闭; 1: 开启。
位 2	TFRMODE	0x0	rw	数据传输模式（Data transfer mode selection） 由软件置起或清零。关闭表示块数据传输，开启表示流数据传输。 0: 关闭; 1: 开启。
位 1	TFRDIR	0x0	rw	数据传输方向（Data transfer direction selection） 由软件置起或清零。关闭表示控制器至卡，开启表示卡至控制器。 0: 关闭; 1: 开启。
位 0	TFREN	0x0	rw	数据传输使能位（Data transfer enabled bit） 由软件置起或清零。如果设置该位为 1，则开始数据传输。根据 TFRDIR 方向位，DCSM 进入 Wait_S 或



Wait\_R 状态，如果在传输的一开始就设置了 RDWTSTART 位，则 DCSM 进入读等待状态。不需要在数据传输结束后清除使能位，但必须更新 SDIO\_DTCNTR 以允许新的数据传输。

0: 关闭;

1: 开启。

注意：写数据后的 7 个 HCLK 时钟周期内不能写入这个寄存器。

### 23.4.10 SDIO 数据计数器寄存器 (SDIO\_DTCNTR)

当 DCSM 从空闲状态进入 Wait\_R 或 Wait\_S 状态时，SDIO 数据计数器寄存器 (SDIO\_DTCNTR) 从 SDIO 数据长度寄存器 (SDIO\_DTLN) 加载数值 (见 23.4.8 节 SDIO 数据长度寄存器 (SDIO\_DTLN))，在数据传输过程中，该计数器的数值递减直到减为 0，然后 DCSM 进入空闲状态并设置数据状态结束标志 DTCMPL。

域	简称	复位值	类型	功能
位 31: 25	保留	0x00	resd	保持默认值。
位 24: 0	CNT	0x00000000	ro	数据计数数值 (Data count value) 读这个寄存器时返回待传输的数据字节数，写这个寄存器无作用。

注意：只能在数据传输结束时读这个寄存器。

### 23.4.11 SDIO 状态寄存器 (SDIO\_STS)

SDIO\_STS 是一个只读寄存器，它包含两类标志：

- 静态标志 (位[23: 22、10: 0])：写入清除中断寄存器 (SDIO\_INTCLR) (见 23.4.12 节 SDIO 清除中断寄存器 (SDIO\_INTCLR))，可以清除这些位。
- 动态标志 (位[21: 11])：这些位的状态变化根据它们对应的那部分逻辑而变化 (例如：BUF 满和空标志变高或变低随 BUF 的数据写入变化)。

域	简称	复位值	类型	功能
位 31: 23	保留	0x000	resd	保持默认值。
位 22	IOIF	0x0	ro	收到 SD I/O 接口中断 (SD I/O interrupt received)
位 21	RXBUF	0x0	ro	在接收 BUF 中的数据可用 (Data available in receive BUF)
位 20	TXBUF	0x0	ro	在发送 BUF 中的数据可用 (Data available in transmit BUF)
位 19	RXBUFE	0x0	ro	接收 BUF 空 (Receive BUF empty)
位 18	TXBUFE	0x0	ro	发送 BUF 空 (Transmit BUF empty) 若使用了硬件流控制，当 BUF 包含 2 个字时，TXBUF_E 信号变为有效。
位 17	RXBUFF	0x0	ro	接收 BUF 满 (Receive BUF full) 若使用了硬件流控制，当 BUF 还差 2 个字满时，RXBUF_F 信号变为有效。
位 16	TXBUFF	0x0	ro	发送 BUF 满 (Transmit BUF full)
位 15	RXBUFH	0x0	ro	接收 BUF 半满 (Receive BUF half full)：BUF 中至少还有 8 个字，该标志位可以作为 DMA 请求。
位 14	TXBUFH	0x0	ro	发送 BUF 半空 (Transmit BUF half empty)：BUF 中至少还可以写入 8 个字，该标志位可以作为 DMA 请求。
位 13	DORX	0x0	ro	正在接收数据 (Data receive in progress)
位 12	DOTX	0x0	ro	正在发送数据 (Data transmit in progress)
位 11	DOCMD	0x0	ro	正在传输命令 (Command transfer in progress)
位 10	DTBLKCMPL	0x0	ro	已发送/接收数据块 (CRC 检测成功) (Data block sent/received (CRC check passed))
位 9	SBITERR	0x0	ro	在宽总线模式，没有在所有数据信号上检测到起始位 (Start bit not detected on all data signals in wide bus mode)
位 8	DTCMPL	0x0	ro	数据结束 (数据计数器，SDIO_DTCNTR=0) (Data end (data counter, SDIO CNT, is zero))
位 7	CMDCMPL	0x0	ro	命令已发送 (不需要响应) (Command sent (no response required))

位 6	CMDRSPCMPL	0x0	ro	已接收到响应（CRC 检测成功）（Command response）
位 5	RXERRO	0x0	ro	接收 BUF 上溢错误（Received BUF overrun error）
位 4	TXERRU	0x0	ro	发送 BUF 下溢错误（Transmit BUF underrun error）
位 3	DTTIMEOUT	0x0	ro	数据超时（Data timeout）
位 2	CMDTIMEOUT	0x0	ro	命令响应超时（Command response timeout） 命令超时时间是一个固定的值，为 64 个 SDIO_CK 时钟周期。
位 1	DTFAIL	0x0	ro	已发送/接收数据块（CRC 检测失败）（Data block sent/received）
位 0	CMDFAIL	0x0	ro	已收到命令响应（CRC 检测失败）（Command response received）

## 23.4.12 SDIO清除中断寄存器（SDIO\_INTCLR）

清除中断寄存器（SDIO\_INTCLR）是一个只写寄存器，在对应寄存器位写‘1’将清除 SDIO 状态寄存器（SDIO\_STS）中的对应位。

域	简称	复位值	类型	功能
位 31: 23	保留	0x000	resd	保持默认值。
位 22	IOIF	0x0	rw	SD I/O 接口标志清除位（SD I/O interface flag clear bit） 由软件置起以清除 IOIF 标志。
位 21: 11	保留	0x000	resd	保持默认值。
位 10	DTBLKCMPL	0x0	rw	DTBLKCMPL 标志清除位（DTBLKCMPL flag clear bit） 由软件置起以清除 DTBLKCMPL 标志。
位 9	SBITERR	0x0	rw	SBITERR 标志清除位（SBITERR flag clear bit） 由软件置起以清除 SBITERR 标志。
位 8	DTCMPL	0x0	rw	DTCMPL 标志清除位（DTCMPL flag clear bit） 由软件置起以清除 DTCMPL 标志。
位 7	CMDCMPL	0x0	rw	CMDCMPL 标志清除位（CMDCMPL flag clear bit） 由软件置起以清除 CMDCMPL 标志。
位 6	CMDRSPCMPL	0x0	rw	CMDRSPCMPL 标志清除位（CMDRSPCMPL flag clear bit） 由软件置起以清除 CMDRSPCMPL 标志。
位 5	RXERRO	0x0	rw	RXERRO 标志清除位（RXERRO flag clear bit） 由软件置起以清除 RXERRO 标志。
位 4	TXERRU	0x0	rw	TXERRU 标志清除位（TXERRU flag clear bit） 由软件置起以清除 TXERRU 标志。
位 3	DTTIMEOUT	0x0	rw	DTTIMEOUT 标志清除位（DTTIMEOUT flag clear bit） 由软件置起以清除 DTTIMEOUT 标志。
位 2	CMDTIMEOUT	0x0	rw	CMDTIMEOUT 标志清除位（CMDTIMEOUT flag clear bit） 由软件置起以清除 CMDTIMEOUT 标志。
位 1	DTFAIL	0x0	rw	DTFAIL 标志清除位（DTFAIL flag clear bit） 由软件置起以清除 DTFAIL 标志。
位 0	CMDFAIL	0x0	rw	CMDFAIL 标志清除位。（CMDFAIL flag clear bit） 由软件置起以清除 CMDFAIL 标志。

## 23.4.13 SDIO中断屏蔽寄存器 (SDIO\_INTEN)

在对应位置'1'，SDIO中断屏蔽寄存器 (SDIO\_INTEN) 决定哪一个状态位产生中断。

域	简称	复位值	类型	功能
位 31: 23	保留	0x000	resd	保持默认值。
位 22	IOIFIEN	0x0	rw	SD I/O 模式接收中断使能 (SD I/O mode received interrupt enable) 由软件置起或清零。开关 SD I/O 模式接收中断功能。 0: 关闭; 1: 开启。
位 21	RXBUFIEN	0x0	rw	接收 BUF 中的数据有效产生中断 (Data available in RxBUF interrupt enable) 由软件置起或清零。开关接收 BUF 中的数据有效中断。 0: 关闭; 1: 开启。
位 20	TXBUFIEN	0x0	rw	发送 BUF 中的数据有效产生中断 (Data available in TxBUF interrupt enable) 由软件置起或清零。开关发送 BUF 中的数据有效中断。 0: 关闭; 1: 开启。
位 19	RXBUFEIEN	0x0	rw	接收 BUF 空产生中断 (RxBUF empty interrupt enable) 由软件置起或清零。开关接收 BUF 空中断。 0: 关闭; 1: 开启。
位 18	TXBUFEIEN	0x0	rw	发送 BUF 空产生中断 (TxBUF empty interrupt enable) 由软件置起或清零。开关发送 BUF 空中断。 0: 关闭; 1: 开启。
位 17	RXBUFFIEN	0x0	rw	接收 BUF 满产生中断 (RxBUF full interrupt enable) 由软件置起或清零。开关接收 BUF 满中断。 0: 关闭; 1: 开启。
位 16	TXBUFFIEN	0x0	rw	发送 BUF 满产生中断 (TxBUF full interrupt enable) 由软件置起或清零。开关发送 BUF 满中断。 0: 关闭; 1: 开启。
位 15	RXBUFHIEN	0x0	rw	接收 BUF 半满产生中断 (RxBUF half full interrupt enable) 由软件置起或清零。开关接收 BUF 半满中断。 0: 关闭; 1: 开启。
位 14	TXBUFHIEN	0x0	rw	发送 BUF 半空产生中断 (TxBUF half empty interrupt enable) 由软件置起或清零。开关发送 BUF 半空中断。 0: 关闭; 1: 开启。
位 13	DORXIEN	0x0	rw	正在接收数据产生中断 (Data receive acting interrupt enable) 由软件置起或清零。开关正在接收数据中断。 0: 关闭; 1: 开启。
位 12	DOTXIEN	0x0	rw	正在发送数据产生中断 (Data transmit acting interrupt enable) 由软件置起或清零。开关正在发送数据中断。 0: 关闭; 1: 开启。
位 11	DOCMDIEN	0x0	rw	正在传输命令产生中断 (Command acting interrupt enable) 由软件置起或清零。开关正在传输命令中断。 0: 关闭; 1: 开启。

位 10	DTBLKCMPIEN	0x0	rw	数据块传输结束产生中断（Data block end interrupt enable） 由软件置起或清零。开关数据块传输结束中断。 0：关闭； 1：开启。
位 9	SBITERRIEN	0x0	rw	起始位错误产生中断（Start bit error interrupt enable） 由软件置起或清零。开关起始位错误中断。 0：关闭； 1：开启。
位 8	DTCMPIEN	0x0	rw	数据传输结束产生中断（Data end interrupt enable） 由软件置起或清零。开关数据传输结束中断。 0：关闭； 1：开启。
位 7	CMDCMPIEN	0x0	rw	命令已发送产生中断（Command sent interrupt enable） 由软件置起或清零。开关命令已发送中断。 0：关闭； 1：开启。
位 6	CMDRSPCMPIEN	0x0	rw	接收到响应产生中断（Command response received interrupt enable） 由软件置起或清零。开关接收到响应中断。 0：关闭； 1：开启。
位 5	RXERROIEN	0x0	rw	接收 BUF 上溢错误产生中断（RxBUF overrun error interrupt enable） 由软件置起或清零。开关接收 BUF 上溢错误中断。 0：关闭； 1：开启。
位 4	TXERRUIEN	0x0	rw	发送 BUF 下溢错误产生中断（TxBUF underrun error interrupt enable） 由软件置起或清零。开关发送 BUF 下溢错误中断。 0：关闭； 1：开启。
位 3	DTTIMEOUTIEN	0x0	rw	数据超时产生中断（Data timeout interrupt enable） 由软件置起或清零。开关数据超时中断。 0：关闭； 1：开启。
位 2	CMDTIMEOUTIEN	0x0	rw	命令超时产生中断（Command timeout interrupt enable） 由软件置起或清零。开关命令超时中断。 0：关闭； 1：开启。
位 1	DTFAILIEN	0x0	rw	数据块 CRC 检测失败产生中断（Data CRC fail interrupt enable） 由软件置起或清零。开关数据块 CRC 检测失败中断。 0：关闭； 1：开启。
位 0	CMDFAILIEN	0x0	rw	命令 CRC 检测失败产生中断（Command CRC fail interrupt enable） 由软件置起或清零。开关命令 CRC 检测失败中断。 0：关闭； 1：开启。

## 23.4.14 SDIOBUF计数器寄存器（SDIO\_BUFCNTR）

SDIOBUF 计数器寄存器（SDIO\_BUFCNTR）包含还未写入 BUF 或还未从 BUF 读出的数据字数目。当在 SDIO 数据控制寄存器（SDIO\_DTCTRL）中设置了数据传输使能位 TFREN，并且 DCSM 处于空闲状态时，BUF 计数器从 SDIO 数据长度寄存器（SDIO\_DTLEN）（见 23.4.8 节 SDIO 数据长度寄存器（SDIO\_DTLEN））加载数值。如果数据长度未与字对齐（4 的倍数），则最后剩下的 1~3 个字节被当成一个字处理。

域	简称	复位值	类型	功能
位 31: 24	保留	0x00	resd	保持默认值。
位 23: 0	CNT	0x000000	ro	将要写入 BUF 或将要从 BUF 读出数据字的数目。

## 23.4.15 SDIO数据BUF寄存器（SDIO\_BUF）

接收和发送 BUF 是一组可读或可写的 32 位宽的寄存器，它在连续的 32 个地址上包含 32 个寄存器，CPU 可以使用 BUF 读写多个操作数。

域	简称	复位值	类型	功能
位 31: 0	DT	0x0000 0000	rw	接收或发送 BUF 数据（Receive and transmit BUF data） BUF 数据占据 32 个 32 位的字，地址为： （SDIO 基址 + 0x80）至（SDIO 基址 + 0xFC）

## 24 通用串行总线全速设备接口（USBFS）

### 24.1 简介

USBFS 实现了 USB2.0 全速设备协议，总线速度 12Mb/s，支持控制传输（Control）、批量传输（Bulk）、同步传输（Isochronous）、中断传输（Interrupt），同时支持 USB 挂起/恢复操作。

USBFS 设计有 8 个可配置双向端点，每个端点可根据具体需求配置为不同的传输类型，USBFS 有一块双端口的 SRAM 用于端点与用户程序的数据交互，同时为了提高传输效率，还实现了批量端点/同步端点的双缓冲机制。USBFS PHY 内置了 DP 上拉电阻，以满足设备需求。

### 24.2 USBFS 时钟与管脚配置

#### 24.2.1 USB 时钟配置

USB 全速设备模块接口中存在两个时钟：USB 控制时钟和 APB1 总线时钟。USB 全速设备总线速度标准为  $12\text{Mb/s} \pm 0.25\%$ 。因此需要给 USBFS 提供  $48\text{MHz} \pm 0.25\%$  的时钟频率用于 USB 总线采样。

USBFS 48M 时钟有两种配置来源：

- HICK 48M  
使用 HICK 48M 时钟作为 USB 控制时钟时，建议开启 ACC 功能。
- 通过 PLL 分频  
注意 PLL 的输出频率要满足 USBDIV（参考时钟配置寄存器（CRM\_CFG））能够正常分频到 48MHz。

注意：使用 USBFS 时，APB1 时钟频率必须大于 12Mhz

#### 24.2.2 USB 管脚配置

PA11 和 PA12 可复用为 DP/DM，复用条件为在 CRM 中使能 USB 模块；PA8 可复用为 SOF 输出功能，复用条件为在 CRM 中使能 USB 模块，并配置 PA8 为推挽复用输出功能。

名称	GPIO	条件
USB_DM	PA11	CRM 中使能 USB 模块
USB_DP	PA12	CRM 中使能 USB 模块
USB_SOF	PA8	可选配置，CRM 中使能 USB 模块，打开 SOF 输出功能，并配置 PA8 为推挽复用输出

### 24.3 USBFS 功能描述

#### 24.3.1 USB 初始化配置

使能 USB 模块之后（在 CRM 中开启 USBFS 时钟），在主机枚举之前需要对 USBFS 进行部分初始化，初始化步骤如下：

1. 清除软件复位 CSRST = 0（设置 CTRL.CSRST=0）
2. 清除所有状态标志（设置 INTSTS=0）
3. 使能 USB Core（设置 DEVADDR.CEN=1）
4. 配置各中断使能位
5. 开启 USB PHY（设置 CTRL.DISUSB=0）

#### 24.3.2 端点配置

USBFS 设计最多支持 8 个双向端点和 16 个单向端点（8 个 IN 端点和 8 个 OUT 端点），每个端点都有与之对应的 USBFS 端点 n 寄存器（USBFS\_EPTn），用于存储端点的各种状态信息，如下是端点需要配置的内容：

- 端点号（配置 EPTADDR，每个端点寄存器的端点号都是可配置的）
- 传输类型（控制传输，批量传输，同步传输，中断传输）
- IN/OUT 端点需配置缓冲区（下节介绍缓冲分配）



- IN/OUT Toggle 状态(对应 DATA0/DATA1)
- IN/OUT 状态 (VALID, NAK, STALL, DISABLE)

注意：端点 0 默认作为控制端点使用，端点 0 的配置通常是在收到主机发送的 reset 信号之后进行配置。

### 24.3.3 USB缓冲区

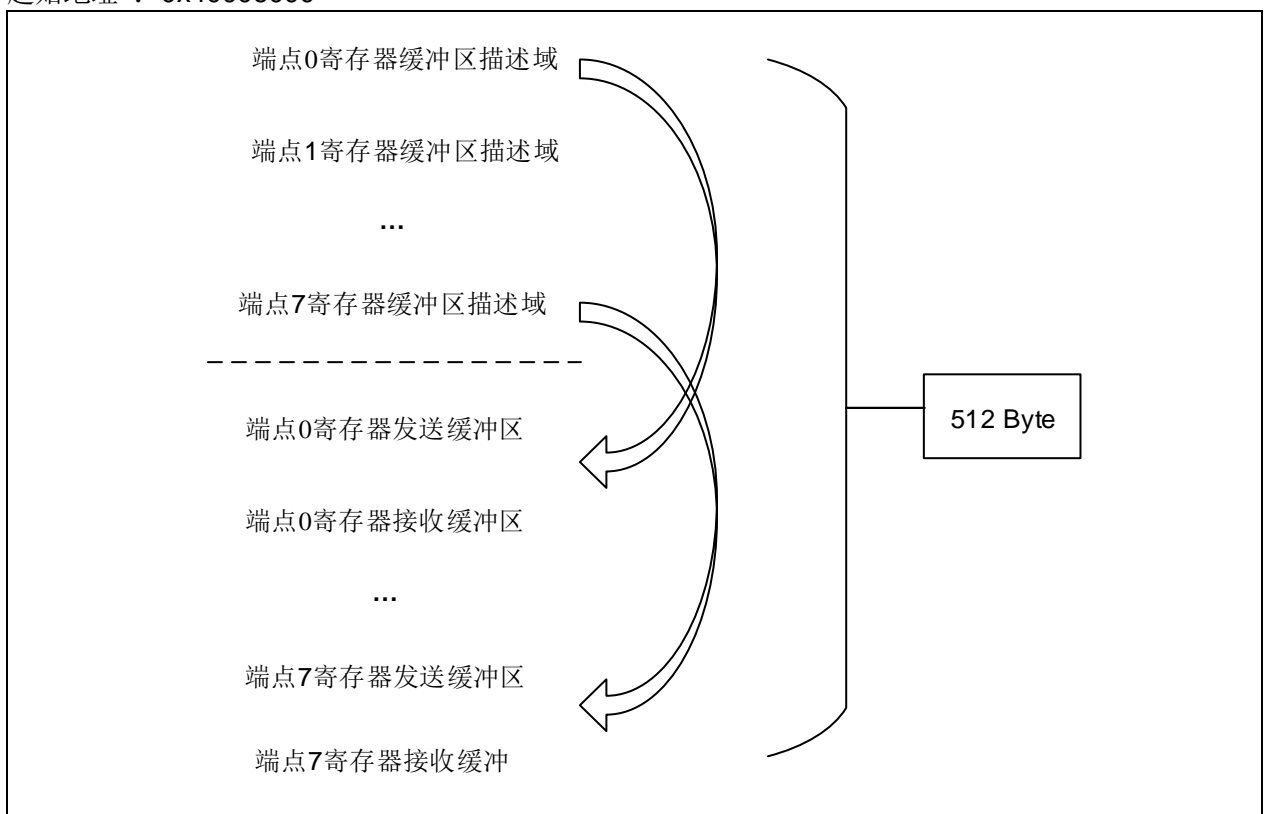
USB 有一块双端口的 SRAM 缓冲区用于端点与用户程序进行数据交互，用户程序和 USBFS 可以同时访问此缓冲区，此缓冲区大小会根据 CAN 的使用情况自动调整，缓冲区映射地址及大小，由下表所示：

表 24-1 缓冲区大小配置表

	USBBUFS	0	1			
工作条件	CAN1 状态	使能/不使能	不使能	不使能	使能	使能
	CAN2 状态	使能/不使能	不使能	使能	不使能	使能
缓冲区大小		512 Byte	1280 Byte	1024 Byte	1024 Byte	768 Byte
缓冲区地址范围		0x4000 6000~ 0x4000 63FF	0x4000 7800~ 0x4000 81FF	0x4000 7800~ 0x4000 7FFF	0x4000 7800~ 0x4000 7FFF	0x4000 7800~ 0x4000 7DFF

缓冲区的结构由端点寄存器的缓冲区描述域和端点的缓冲区组成，端点寄存器描述表里面描述端点接收/发送相对于缓冲区的偏移地址。如下是一个缓冲区结构表示例：（以 512 Byte 大小举例）

起始地址：0x40006000



USBFS 的架构中，每个端点寄存器需对应一个缓冲区描述域，用于描述该端点的接收/发送时的缓冲区，数据长度等信息。常规端点寄存器缓冲区描述域结构如下图所示（双缓冲描述表在下节介绍）：

端点 n:

0	2	4	6	8	10	12	14
TnADDR	保留	TnLEN	保留	RnADDR	保留	RnLEN	保留
发送缓冲区描述				接收缓冲描述			



缓冲区描述域的起始地址等于缓冲区地址+BTADDR\*2，用户程序可根据具体需求将端点寄存器的缓冲描述放到不同位置，默认 BTADDR 为 0。

USBFS 有 8 个端点寄存器，每个端点寄存器描述表实际占用 8 Byte 空间，用户在编程过程中，使用端点寄存器的不同，需要为缓冲区描述域留下足够空间。在给端点分配发送/接收缓冲区时，要注意偏移地址不能占用缓冲区描述和其它端点的发送/接收缓冲区。

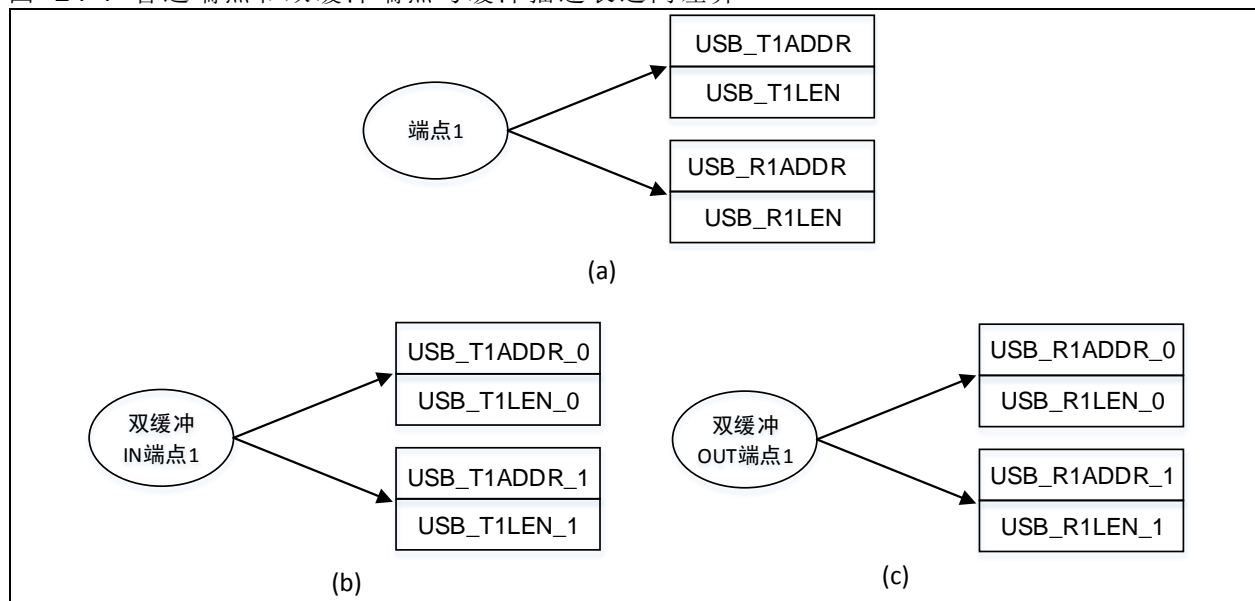
注意：APB1 总线宽度为 32 位，但是由于缓冲区为 16 位宽存储器，APB1 总线对分组缓冲区的一次写操作只能写入 2 字节数据，所以写入 16 字节数据需要 8 次写操作。

#### 24.3.4 双缓冲端点配置

为提高批量传输和同步传输的传输效率，USBFS 设计了双缓冲模式。

即一个 IN 端点（或者 OUT 端点）对应两个缓冲区，普通端点和双缓冲端点与缓冲描述表之间的差异如下图所示：

图 24-1 普通端点和双缓冲端点与缓冲描述表之间差异



两种端点传输类型支持双缓冲功能：

- 批量传输端点（Bulk）  
配置 TRANS\_TPYE=00 和 EXF=1 开启双缓冲功能
- 同步传输端点（Isochronous）  
TRANS\_TPYE=10，同步传输默认使用双缓冲功能

双缓冲提高效率的方法是使用两个缓冲区，USBFS 和用户程序同时访问不同缓冲区，达到同时处理数据的功能。USBFS 和用户程序需要确定当前哪个缓冲区是可以访问的，因此 USBFS 引入一个标志 SBUF 用于判断。

- OUT 双缓冲端点：SBUF 对应 USB\_EPTn 的 bit6  
SBUF=1, USBFS 使用 RnADDR\_0 和 RnLEN\_0, 用户程序使用 RnADDR\_1 和 RnLEN\_1  
SBUF=0, USBFS 使用 RnADDR\_1 和 RnLEN\_1, 用户程序使用 RnADDR\_0 和 RnLEN\_0
- IN 双缓冲端点：SBUF 对应 USB\_EPTn 的 bit14  
SBUF=1, USBFS 使用 TnADDR\_0 和 TnLEN\_0, 用户程序使用 TnADDR\_1 和 TnLEN\_1  
SBUF=0, USBFS 使用 TnADDR\_1 和 TnLEN\_1, 用户程序使用 TnADDR\_0 和 TnLEN\_0

注意：端点 0 不能作为双缓冲端点。

#### 24.3.5 SOF输出

USBFS 收到主机发送的 SOF，会产生 SOF 标志，同时可以从 USBFS SOF 帧编号寄存器（USBFS\_SOFNUM）读出当前帧号。通过使能 USBFS CFG 控制寄存器（USBFS\_CFG）的

SOFOUTEN 位，将产生一个宽度为 24 个 APB1 时钟周期的 SOF 脉冲信号输出到管脚上。

### 24.3.6 挂起/恢复

在 USB2.0 全速协议中规定了一种低功耗状态，即挂起状态，协议规定当设备检测到总线上连续 3ms 处于空闲状态，便进入挂起状态。USBFS 可以通过两种方式进入挂起状态：

- 1、当检测到连续三个 SOF 丢失
- 2、应用程序置位控制寄存器 SSP

当设备检测到连续三个 SOF 丢失后，需要应用程序置位控制寄存器 SSP 和 LPM，从而达到禁止对 SOF 的检查，并关闭 USB 物理收发器的静态功耗的作用。

USB 设备从挂起状态回到正常工作状态的过程叫恢复。当 USB 设备处于挂起状态时，其上行端口的任何非空闲状态（如包开始信号 SOP）都将使其得到恢复；另外 USB 设备自己也可以通过配置 GRESUME 寄存器申请启动恢复操作，这被称为远程唤醒。

## 24.4 USB 中断

低优先级中断：中断号 20、74，当没有开启高优先中断是，所有 USBFS 中断都可通过低优先级中断进行处理。

高优先级中断：中断号 19、73，只有同步传输端点和双缓冲批量端点可触发高优先级中断。

USB 唤醒中断：中断号 42，当 USB 进入挂起状态，芯片进入 Deep Sleep 模式之后，可通过此中断唤醒。

USBFS 默认中断号(19,20)与 CAN1 共用，导致 USBFS 和 CAN1 不能同时使用，因此引入 USB 新的中断号(73,74)，用户程序可以通过设置中断映射寄存器（CRM\_INTMAP）中的 USBINTMAP=1 使 USBFS 的中断号映射到 73、74。

## 24.5 USBFS 寄存器

必须以字（32 位）的方式操作这些外设寄存器。

表 24-2 USBFS 寄存器映像和复位值

寄存器简称	基址偏移量	复位值
USBFS_EPT0	0x00	0x0000
USBFS_EPT1	0x04	0x0000
USBFS_EPT2	0x008	0x0000
USBFS_EPT3	0x0C	0x0000
USBFS_EPT4	0x10	0x0000
USBFS_EPT5	0x14	0x0000
USBFS_EPT6	0x18	0x0000
USBFS_EPT7	0x1C	0x0000
USBFS_CTRL	0x40	0x0003
USBFS_INTSTS	0x44	0x0000
USBFS_SOFNUM	0x48	0x0XXX
USBFS_DEVADDR	0x4C	0x0000
USBFS_BUFTBL	0x50	0x0000
USBFS_CFG	0x60	0x0000
USBFS_TnADDR	[USB_BUFTBL] x 2 + n x 16	0xFFFF
USBFS_TnLEN	[USB_BUFTBL] x 2 + n x 16 + 4	0xFFFF
USBFS_RnADDR	[USB_BUFTBL] x 2 + n x 16 + 8	0xFFFF

USBFS\_RnLEN

[USB\_BTABLE] x 2 + n x 16 + 12

0xFFFF

### 24.5.1 USBFS端点n寄存器（USBFS\_EPTn），n=[0..7]

域	简称	复位值	类型	功能
位 15	RXTC	0x0	rw0c	接收完成标志位（Rx transaction completed） 在接收完 OUT 或者 SETUP 事务时，此位将置起，表示接收完成。 0：应用程序清除接收完成标志 1：OUT/SETUP 事务接收完成
位 14	RXDTS	0x0	tog	接收数据翻转同步位（Rx Data Toggle (DAT0/DATA1) Synchronization） 非 ISO 传输，此位表示当前事务是 DATA0/DATA1。 0：DATA0 1：DATA1
位 13: 12	RXSTS	0x0	tog	接收状态（Rx Status） 表示当前响应主机 OUT 传输的状态，存在 4 种状态，DISABLE, NAK, STALL, ACK 00: DISABLED, 端点忽略所有的接收请求 01: STALL, 端点以 STALL 分组响应所有的接收请求 10: NAK, 端点以 NAK 分组响应所有的接收请求 11: VALID, 端点可用于接收
位 11	SETUPTC	0x0	rog	SETUP 传输完成标志位（Setup transaction completed） 此位表示当前 RXTC 置位后，用于区分是 OUT 还是 SETUP 接收完成。 0：OUT 事务传输完成 1：SETUP 事务传输完成
位 10: 9	TRANS_TYPE	0x0	rw	传输类型（Transfer types） 此位用于描述 USB 的 4 种传输类型，Control, Bulk, Interrupt, ISO。 00: BULK, 批量端点；可搭配 EXF 位寄存器 01: CTRL, 控制端点；可搭配 EXF 位寄存器 10: ISO, 同步端点 11: INT, 中断端点
位 8	EXF	0x0	rw	端点扩展功能（Endpoint Extend function） USB 端点的扩展功能，主要用于 Bulk 和 Control 传输。 Bulk 传输时，如果此位置起，表示使用双缓冲功能。 Control 传输时，如果此位置起，表示会检测 SETUP 传输的状态阶段数据长度是否为 0，如果不为 0，则返回 STALL。
位 7	TXTC	0x0	rw0c	发送完成标志位（Tx transaction completed） IN 事务完成之后，会置起此位，表示发送已经完成。 0：应用程序清除发送完成标志 1：IN 事务接收完成
位 6	TXDTS	0x0	tog	发送数据翻转同步位（Tx Data Toggle (DAT0/DATA1) Synchronization） 非 ISO 端点，表示当前 IN 事务是 DATA0/DATA1。 0：DATA0 1：DATA1
位 5: 4	TXSTS	0x0	tog	发送状态（Tx Status） 表示当前响应主机 IN 传输的状态，存在 3 种状态，DISABLE, NAK, STALL, ACK 00: DISABLED, 端点忽略所有的发送请求 01: STALL, 端点以 STALL 分组响应所有的发送请求 10: NAK, 端点以 NAK 分组响应所有的发送请求 11: VALID, 端点可用于发送
位 3: 0	EPTADDR	0x0	rw	端点地址（Endpoint address） 表示端点地址。

## 24.5.2 USBFS控制寄存器（USBFS\_CTRL）

域	简称	复位值	类型	功能
位 15	TCIEN	0x0	rw	传输完成中断使能位（Transmission completed interrupt enable） 0：关闭 1：开启
位 14	UCFORIEN	0x0	rw	USB Core 缓冲区溢出中断使能位（USB Core fifo overrun interrupt enable） 0：关闭 1：开启
位 13	BEIEN	0x0	rw	总线错误中断使能位（Bus error interrupt enable） 0：关闭 1：开启
位 12	WKIEN	0x0	rw	唤醒/远程唤醒中断使能位（Wakeup/Remote wakeup interrupt enable） 0：关闭 1：开启。
位 11	SPIEN	0x0	rw	总线挂起使能位（Bus suspend interrupt enable） 0：关闭 1：开启
位 10	RSTIEN	0x0	rw	总线复位中断使能位（Bus reset interrupt enable） 0：关闭 1：开启
位 9	SOFIEN	0x0	rw	帧起始中断使能位（Start of frame interrupt enable） 0：关闭 1：开启
位 8	LSOFIEN	0x0	rw	丢失帧起始中断使能位（Lost start of frame interrupt enable） 0：关闭 1：开启
位 7: 5	保留	0x0	resd	保持默认值。
位 4	GRESUME	0x0	rw	产生 Resume 请求（Generate Resume request） 在挂起状态下软件设置为 1，将向主机发送 resume 信号，用于唤醒主机，软件必须在 10ms 到 15ms 内清 0 此位。
位 3	SSP	0x0	rw	软件挂起配置（soft suspend config） 此位由软件在检测到挂起标志时置 1，当退出挂起状态时，软件必须设置此位为 0。 0：软件退出挂起状态 1：软件进入挂起模式
位 2	LPM	0x0	rw	低功耗模式（Low power mode） 当进入挂起状态时，可以设置此位为 1 以减少功耗，此位由 USB Core 唤醒是自动清除。 0：非低功耗模式 1：低功耗模式
位 1	DISUSB	0x1	rw	关闭 USB PHY（Disble USB PHY） 0：USB PHY 开启 1：USB PHY 关闭
位 0	CSRST	0x1	rw	软件复位（Core soft Reset） 0：软件清除复位 1：软件复位 usb core，将产生一个 reset 中断

### 24.5.3 USBFS中断状态寄存器（USBFS\_INTSTS）

域	简称	复位值	类型	功能
位 15	TC	0x0	ro	传输完成（Transaction completed） 0：复位值 1：USB 完成一次 IN/OUT 事务后置起此位。
位 14	UCFOR	0x0	rw0c	USB Core 缓冲区溢出（USB Core fifo overrun） 0：复位值 1：USB Core 缓冲区溢出。
位 13	BE	0x0	rw0c	总线出错（Bus error） 0：复位值 1：检测到总线数据有错，如 crc 错误、位填充错误、应答超时错误、帧格式错误等。
位 12	WK	0x0	rw0c	唤醒信号（Wakeup） 0：复位值 1：USB 挂起状态时，USB 收到唤醒信号。
位 11	SP	0x0	rw0c	总线挂起（Bus Suspend） 0：复位值 1：检测到总线在 3ms 没有数据传输，总线进入挂起状态
位 10	RST	0x0	rw0c	总线复位（Bus reset） 0：复位值 1：USB 检测到总线有 USB 复位信号
位 9	SOF	0x0	rw0c	帧起始（Start of frame） 0：复位值 1：检测到总线有 SOF 事务时将置起此位。
位 8	LSOF	0x0	rw0c	丢失帧起始（Lost start of frame） 0：复位值 1：当总线超过 1ms 没有检测到 SOF。
位 7: 5	保留	0x0	resd	保持默认值。
位 4	INOUT	0x0	ro	IN/OUT 事务（In /Out transaction） 当产生 TC 完成中断时，通过此位判断当前是 IN 事务完成还是 OUT 事务完成。 0：IN 事务 1：OUT 事务
位 3: 0	EPT_NUM	0x0	ro	端点号（Endpoint number） 当产生 TC 完成中断时，通过此位判断当前是哪个端点传输完成。

### 24.5.4 USBFS SOF帧编号寄存器（USBFS\_SOFRNUM）

域	简称	复位值	类型	功能
位 15	DPSTS	0x0	ro	D+ 状态位（D+ status） 表示 D+ 状态。
位 14	DMSTS	0x0	ro	D- 状态位（D- status） 表示 D- 的状态。
位 13	CLCK	0x0	ro	连接锁定（Connect Locked） 连续收到两个 SOF，此位将置起。
位 12: 11	LSOFNUM	0x0	ro	起始帧丢失个数（Lost SOF number） LSOF 后，表示当前丢失 SOF 个数；收到 SOF 事务后，硬件清除此位。
位 10: 0	SOFNUM	0xFFFF	ro	起始帧编号（Start of Frame number） 记录当前 SOF 帧编号

### 24.5.5 USBFS设备地址寄存器（USBFS\_DEVADDR）

域	简称	复位值	类型	功能
位 15: 8	保留	0x00	resd	保持默认值。
位 7	CEN	0x0	rw	USB Core 使能位（USB Core Enable） 0: USB Core 停止工作 1: USB Core 开始工作
位 6: 0	ADDR	0x00	rw	主机分配给设备地址（Host assign Device address） 记录在主机在枚举过程中分配给设备的地址。

### 24.5.6 USBFS分组缓冲区描述表地址寄存器（USBFS\_BUFTBL）

域	简称	复位值	类型	功能
位 15: 3	BTADDR	0x0000	rw	端点缓冲区描述表的起始位置（Endpoint buffer table start address） 此位表示端点的缓冲区描述表的起始位置，默认为 0。
位 2: 0	保留	0x0	resd	保留位，由硬件置为 0

### 24.5.7 USBFS CFG控制寄存器（USBFS\_CFG）

域	简称	复位值	类型	功能
位 15: 2	保留	0x0000	resd	保持默认值。
位 1	PUO	0x0	rw	DP 上拉电阻关闭位（DP pullup off） 0: DP 上拉电阻使能； 1: DP 上拉电阻关闭。
位 0	SOFOUTEN	0x0	rw	SOF 输出使能位（SOF output enable） 0: 不输出 SOF 脉冲； 1: 输出 SOF 脉冲到管脚上。

### 24.5.8 USBFS发送缓冲区首地址寄存器 n（USBFS\_TnADDR）

域	简称	复位值	类型	功能
位 15: 1	TnADDR	0xFFFF	rw	发送缓冲区首地址（Transmission buffer first address） 此位记录了收到下一个 IN 事务请求时，需要发送的数据所在的缓冲区起始地址。
位 0	保留	0x0	resd	因为分组缓冲区的地址必须按字对齐，所以此位必须为'0'。

### 24.5.9 USBFS发送数据长度寄存器 n（USBFS\_TnLEN）

域	简称	复位值	类型	功能
位 15: 10	保留	0xFFFF	resd	保持默认值。
位 9: 0	TnLEN	0xFFFF	rw	发送数据长度（Transmission length） 此位记录了收到下一个 IN 事务请求时要传输的数据字节数。

### 24.5.10 USBFS接收缓冲区首地址寄存器 n（USBFS\_RnADDR）

域	简称	复位值	类型	功能
位 15: 1	RnADDR	0xFFFF	rw	接收缓冲区首地址（Reception buffer first address） 此位记录了收到下一个 OUT 或者 SETUP 事务请求时，用于保存数据的缓冲区起始地址。
位 0	保留	0x0	resd	因为分组缓冲区的地址按字对齐，所以此位必需为'0'。

## 24.5.11 USBFS接收数据字节数寄存器 n (USBFS\_RnLEN)

域	简称	复位值	类型	功能
位 15	BSIZE	0xX	rw	存储区块的大小 (Block size) 表示当前端点接收缓冲块大小 如果 BSIZE=0, 块的大小为 2 字节, 分组缓冲区的大小范围为 2—62 个字节。 如果 BSIZE=1, 块的大小为 32 字节, 分组缓冲区的大小范围为 32—1024 字节
位 14: 10	NBLK	0xXX	rw	存储区块的数目 (Number of blocks) 表示当前端点接收缓冲区用到几个块。
位 9: 0	RnLEN	0xXXX	rw	接收到的字节数 (Reception Length) 表示当前收到数据的长度。



## 25 HICK 自动时钟校准（ACC）

### 25.1 简介

HICK 自动时钟校准器（HICK ACC）利用 USB 模块产生的 SOF 信号（周期为 1 毫秒）作为参考信号，实现对 HICK 时钟的采样和校准。

本模块主要功能就是实现对 USB 设备提供  $48\text{MHz} \pm 0.25\%$  精度的时钟。

它采取“跨越回归”算法，可以将校准后的频率尽可能地靠近目标频率。

### 25.2 主要特性

- 可配置的中心频率
- 可配置的触发校准功能的边界频率
- 满足中心频率  $\pm 0.25\%$  的精度要求
- 状态检测标志
  - 校准就绪标志
- 一个错误检测标志
  - 参考信号丢失错误
- 2 个带标志的中断源
  - 校准就绪标志
  - 参考信号丢失错误
- 两种校验方式：粗校验和精校验。

### 25.3 中断请求

表 25-1 ACC中断请求

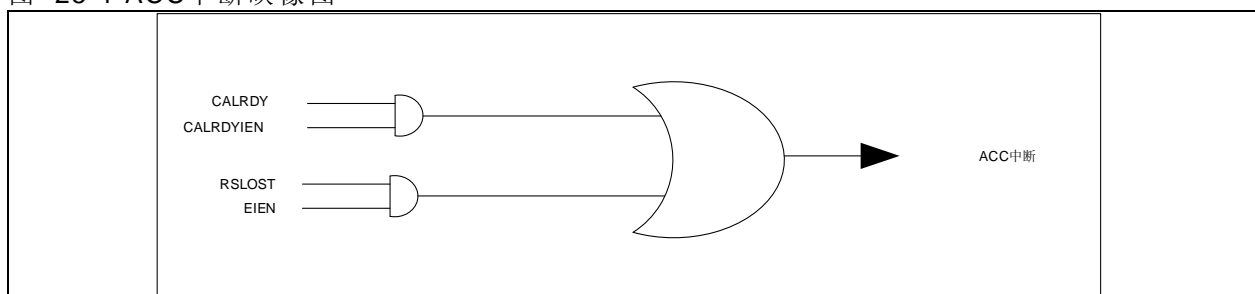
中断事件	事件标志	使能位
校准就绪	CALRDY	CALRDYIEN
参考信号丢失错误	RSLOST	EIEN

ACC 的各种中断事件被连接到同一个中断向量（见下图），有以下各种中断事件：

- 校准期间：当校准就绪或者参考信号丢失错误。

如果设置了对应的使能控制位，这些事件就可以产生各自的中断。

图 25-1 ACC中断映像图



### 25.4 功能概述

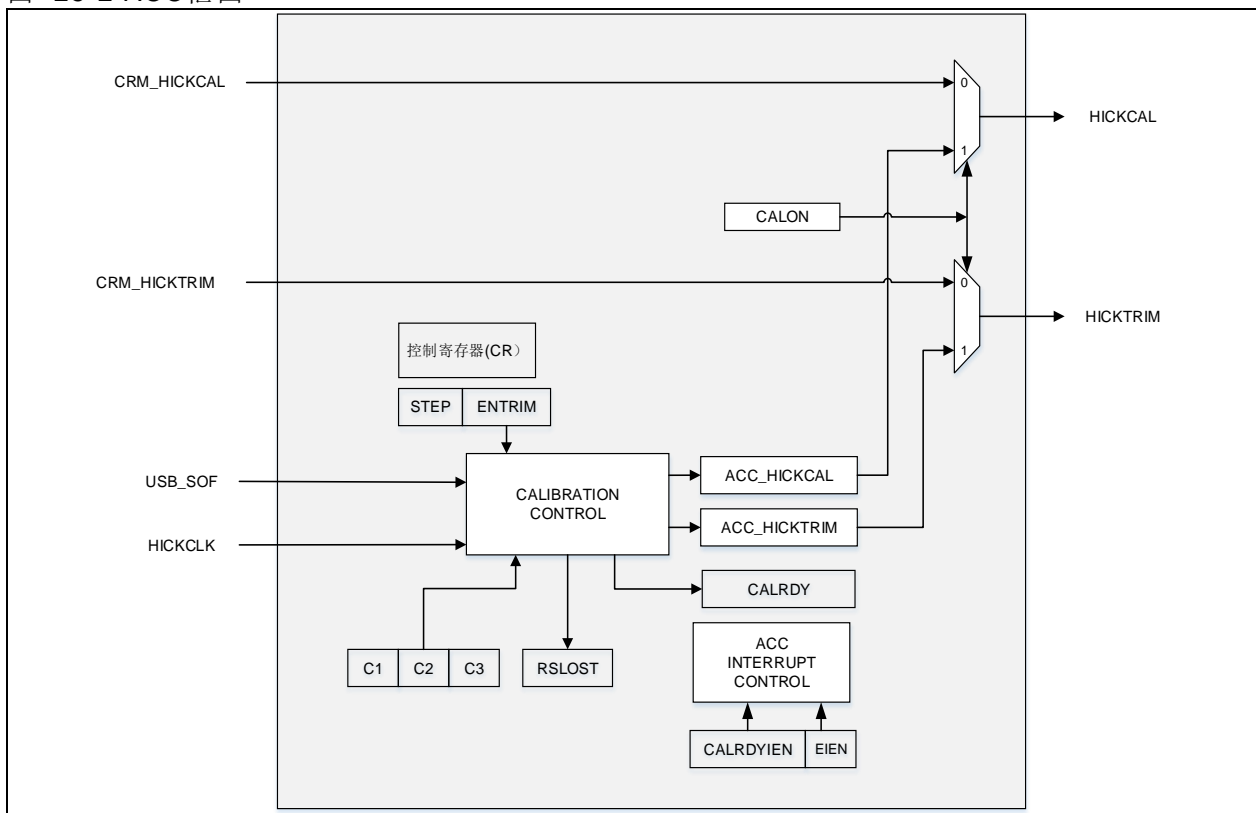
ACC 模块的功能：利用 USB 模块产生的 SOF 信号（周期为 1 毫秒）作为参考信号，实现对 HICK 时钟的采样和校准。特别的是，可以将 HICK 时钟的频率精度校准到  $\pm 0.25\%$  以内的精度，从而满足高精度时钟要求的应用场景，例如 USB 应用。

本模块的信号均未外接到芯片管脚，而是和芯片内部的 CRM、HICK 等模块相连。

- **CRM\_HICKCAL**: 复位和时钟控制 (CRM) 模块之 HICKCAL。此信号用于 bypass 模式下对内部高速时钟 (HICK 模块) 的校准, 其值的大小由时钟控制寄存器 (CRM\_CTRL) 中的 HICKCAL[7: 0]定义。
  - **CRM\_HICKTRIM**: 复位和时钟控制 (CRM) 模块之 HICKTRIM。此信号用于 bypass 模式下对内部高速时钟校准 (HICK) 的校准, 其值的大小由时钟控制寄存器 (CRM\_CTRL) 中的 HICKTRIM[5: 0]定义。
- 默认数值为 32, 可以把 HICK 调整到  $8\text{MHz} \pm 0.25\%$ ; 每步 CRM\_HICKTRIM 的变化调整 HICK 的频率 20kHz (设计值)。
- **USB\_SOF**: USB 设备解析给出的帧开始信号 (USB Start-of-Frame)。其高电平宽度为 12 个系统时钟周期, 周期为 1 毫秒的脉冲信号。
  - **HICKCLK**: HICK 时钟。本系列的 HICK 模块输出的原始时钟频率为 48MHz, 但是 HICK 校准模块使用的采样时钟是除频 (1/6) 电路输出的时钟, 频率约 8MHz。
  - **HICKCAL**: HICK 模块的校验信号。对于除频 (1/6) 后的 HICK 时钟来讲, HICKCAL 每改变一步, 除频 (1/6) 后 HICK 时钟频率改变 40KHz (设计值), 且为正相关。换句话说, HICKCAL 每增加一, 除频 (1/6) 后 HICK 时钟频率会增加 40KHz (设计值); HICKCAL 每减少一, 除频 (1/6) 后 HICK 时钟频率会减少 40KHz。
  - **HICKTRIM**: HICK 模块的校验信号。对于除频 (1/6) 后的 HICK 时钟来讲, HICKTRIM 每改变一步, 除频 (1/6) 后 HICK 时钟频率改变 20KHz (设计值), 且为正相关。

关于以上寄存器中每个位的具体定义, 请参考寄存器描述第 25.6 节: 寄存器描述。

图 25-2 ACC框图

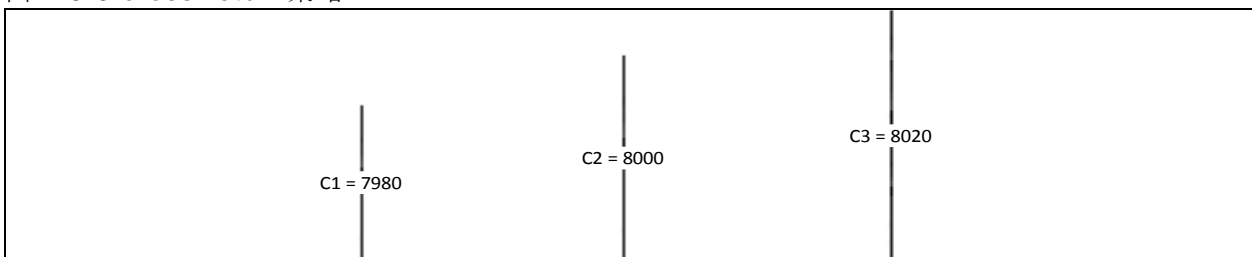


## 25.5 原理分析

**USB\_SOF 周期信号**: 1 毫秒的周期性必须是准确的, 是自动校准模块能够正常工作的前提条件。

**cross-return 策略**: 以计算出离理论值最近的校准值。从理论上来说, 可以将校准后的实际频率调校到离目标频率 (8MHz) 约 0.5 个 step 的精度范围以内。

图 25-3 cross-return策略



如上图所示，一旦触发自动校准的条件满足，自动校准就会按照 **step** 所规定的步长调整 **HICKCAL** 或者 **HICKTRIM**。

**跨越 (cross):**

在满足自动校准的条件后的第一个 1 毫秒采样周期内的实际采样值要么小于 **C2**，要么大于 **C2**。

当这个值小于 **C2**，自动校准按照 **step** 的定义，增加 **HICKCAL** 或者 **HICKTRIM**，直到实际采样值比 **C2** 大，实现实际采样值由小到大对 **C2** 的跨越。

当这个值大于 **C2**，自动校准按照 **step** 的定义，减少 **HICKCAL** 或者 **HICKTRIM**，直到实际采样值比 **C1** 小，实现实际采样值由大到小对 **C2** 的跨越。

**回归 (return):**

在跨越完成后，比较在跨越前后的实际采样值和 **C2** 之间的差值（按绝对值计算），得到离 **C2** 最近的实际采样值，从而得到最佳的校验值 **HICKCAL** 或者 **HICKTRIM**。

若跨越后的实际采样值和 **C2** 之间的差值小于跨越前的实际采样值和 **C2** 之间的差值，则以跨越后的校验值为准，并结束校验流程，直到满足下一个满足自动校准的条件。

若跨越后的实际采样值和 **C2** 之间的差值大于跨越前的实际采样值和 **C2** 之间的差值，则以跨越前的校验值为准，那么校验值会退回一个 **step**，并返回到跨越前的那个校验值，并结束校验流程，直到满足下一个满足自动校准的条件。

按照 **cross-return** 策略，在理论上，可以得到离中心频率约 0.5 个 **step** 所对应的频率精度。

如下四种情形会启动自动校准：

第一， **CALON** 的上升沿（从 0 到 1）；

第二， 当 **CALON**=1 时，参考信号丢失之后又恢复；

第三， 当采样计数器的值小于 **C1**；

第四， 当采样计数器的值大于 **C3**。

在 **CALON** 的上升沿，即便采样计数器的值大于 **C1** 并小于 **C3**，也会启动自动校准，其目的在于，在 **CALON** 之后，能够尽快将 **HICK** 的频率调整到中心频率的 0.5 个 **step** 以内。

以上四种情形的自动校准的结果均能将 **HICK** 的频率调整到中心频率的 0.5 个 **step** 以内。所以为了获得最佳的校准精度，建议将 **step** 保持为默认值 1。若将 **step** 设为 0，则 **HICKCAL** 或者 **HICKTRIM** 将无法改变，也即，无法校准。

## 25.6 寄存器描述

有关寄存器描述里所使用的缩写，请参考“寄存器描述表中使用的缩写列表”。

必须以字（32 位）的方式操作这些外设寄存器。

表 25-2 ACC寄存器映像和复位值

寄存器简称	基址偏移量	复位值
ACC_STS	0x00	0x0000 000
ACC_CTRL1	0x04	0x0000 0100
ACC_CTRL2	0x08	0x0000 2080
ACC_C1	0x0C	0x0000 1F2C
ACC_C2	0x10	0x0000 1F40
ACC_C3	0x14	0x00000 1F54

## 25.6.1 状态寄存器 (ACC\_STS)

域	简称	复位值	类型	功能
位 31: 2	保留	0x0000000	resd	保持默认值。
位 1	RSLOST	0x0	ro	参考信号丢失 (Reference Signal Lost) 0: 参考信号未丢失; 1: 参考信号丢失。 注: 在校验过程中, 当校准模块的采样计数器的值为 C2 的 2 倍时, 还未检测到 SOF 参考信号, 则意味着参考信号丢失。内部状态机回归到 idle 状态, 除非再次检测到 SOF 信号, 否则内部时钟采样计数器保持为 0。在 CALON 位清零后, 或者向 RSLOST 写入 0, 则 RSLOST 立即被清零。仅仅在 CALON=1 时, 才会检测参考信号。
位 0	CALRDY	0x0	ro	内部高速时钟就绪 (Internal high-speed clock calibration ready) 0: 内部 8MHz 振荡器校验没有就绪; 1: 内部 8MHz 振荡器校验就绪。 注: 由硬件置'1'来指示内部 8MHz 振荡器已经校验到离 8MHz 最近的频率上。在 CALON 位清零后, 或者向 CALRDY 写入 0, 则 CALRDY 立即被清零。

## 25.6.2 控制寄存器1 (ACC\_CTRL1)

域	简称	复位值	类型	功能
位 31: 12	保留位	0x00000	resd	硬件强制为 0
位 11: 8	STEP	0x1	rw	校准的步长 这 4 位定义了每次校准改变的值。 备注: 为了获得更高的校准精度, 建议将 step 设为 1。 当 ENTRIM=0, 仅校准 HICKCAL, 若 step 改变 1, 对应的 HICKCAL 也改变 1, HICK 频率改变 40KHz (设计值), 为正相关关系。 当 ENTRIM=1, 仅校准 HICKTRIM, 若 step 改变 1, 对应的 HICKTRIM 也改变 1, HICK 频率改变 20KHz (设计值), 为正相关关系。
位 7: 6	保留位	0x0	rw	硬件强制为 0
位 5	CALRDYIEN	0x0	rw	CALRDY 中断使能 (CALRDY interrupt enable) 该位由软件设置或清除。 0: 禁止产生中断; 1: 当状态寄存器 (ACC_STS) 中的 CALRDY 为'1'时, 产生 ACC 中断。
位 4	EIEN	0x0	rw	RSLOST 中断使能 (RSLOST error interrupt enable) 该位由软件设置或清除。 0: 禁止产生中断; 1: 当状态寄存器 (ACC_STS) 中的 RSLOST 为'1'时, 产生 ACC 中断。
位 3: 2	保留位	0x0	rw	硬件强制为 0
位 1	ENTRIM	0x0	rw	TWK 使能 (Enable trim) 该位由软件设置或清除。 0: 仅校准 HICKCAL; 1: 仅校准 HICKTRIM。 注: 为了获得更高的校准精度, 建议将 ENTRIM 设为 1。
位 0	CALON	0x0	rw	Calibration 使能 (Calibration on) 该位由软件设置或清除。 0: 禁止校验; 1: 使能校验, 并开始搜寻 USB_SOF 上的脉冲。 注: 如果没有 USB_SOF 参考信号, 则本模块无法使用。若对 HICK 时钟的精度没有要求, 也无需开启本模块以节省功耗。

### 25.6.3 控制寄存器2 (ACC\_CTRL2)

域	简称	复位值	类型	功能
位 31: 14	保留位	0x00000	resd	硬件强制为 0
位 13: 8	HICKTRIM	0x00	ro	<p>内部高速时钟自动调整 (Internal high-speed auto clock trimming)</p> <p>该位由软件读取, 不可写。</p> <p>由 ACC 自动校准模块来调整内部高速时钟, 它们被叠加在 ACC_HICKCAL[7: 0]数值上。这些位在 ACC_HICKCAL[7: 0]的基础上, 让用户可以输入一个调整数值, 根据电压和温度的变化调整内部 HICK RC 振荡器的频率。</p> <p>默认数值为 32, 可以把 HICK 调整到 <math>8\text{MHz} \pm 0.25\%</math>; 每步 ACC_HICKTRIM 的变化调整 20kHz (设计值)。</p>
位 7: 0	HICKCAL	0x00	ro	<p>内部高速时钟自动校准 (Internal high-speed auto clock calibration)</p> <p>该位由软件读取, 不可写。</p> <p>由 ACC 自动校准模块来调整内部高速时钟, 让用户可以输入一个调整数值, 根据电压和温度的变化调整内部 HICK RC 振荡器的频率。</p> <p>默认数值为 128, 可以把 HICK 调整到 <math>8\text{MHz} \pm 0.25\%</math>; 每步 ACC_HICKCAL 的变化调整 40kHz (设计值)。</p>

### 25.6.4 比较值1 (ACC\_C1)

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	C1	0x1F2C	rw	<p>比较值 1 (Compare 1)</p> <p>该值是触发校准的下边界, 默认值为 7980。当自动校验模块在 1 毫秒的周期内采样的时钟个数小于或等于 C1, 则会触发自动校验;</p> <p>当实际采样值 (1 毫秒内的时钟个数) 大于 C1, 且小于 C3, 则不会触发自动校验。</p>

### 25.6.5 比较值2 (ACC\_C2)

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	C2	0x1F40	rw	<p>比较值 2 (Compare 2)</p> <p>该值确定了理想频率 (8MHz) 时钟在 1 毫秒为采样周期的时钟个数, 默认值为 8000, 也是其理论值。</p> <p>该值是 cross-return 策略的中心点, 以计算出离理论值最近的校准值。从理论上来说, 可以将校准后的实际频率调教到离目标频率 (8MHz) 约 0.5 个 step 的精度范围以内。</p>

### 25.6.6 比较值3 (ACC\_C3)

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	C3	0x1F54	rw	<p>比较值 3 (Compare 3)</p> <p>该值是触发校准的上边界。当自动校验模块在 1 毫秒的周期内采样的时钟个数大于或等于 C3, 则会触发自动校验;</p> <p>当实际采样值 (1 毫秒内的时钟个数) 大于 C1, 且小于 C3, 则不会触发自动校验。</p>

## 26 以太网控制器（EMAC）

仅 AT32F407 支持该模块，AT32F403A 则不支持。

### 26.1 EMAC简介

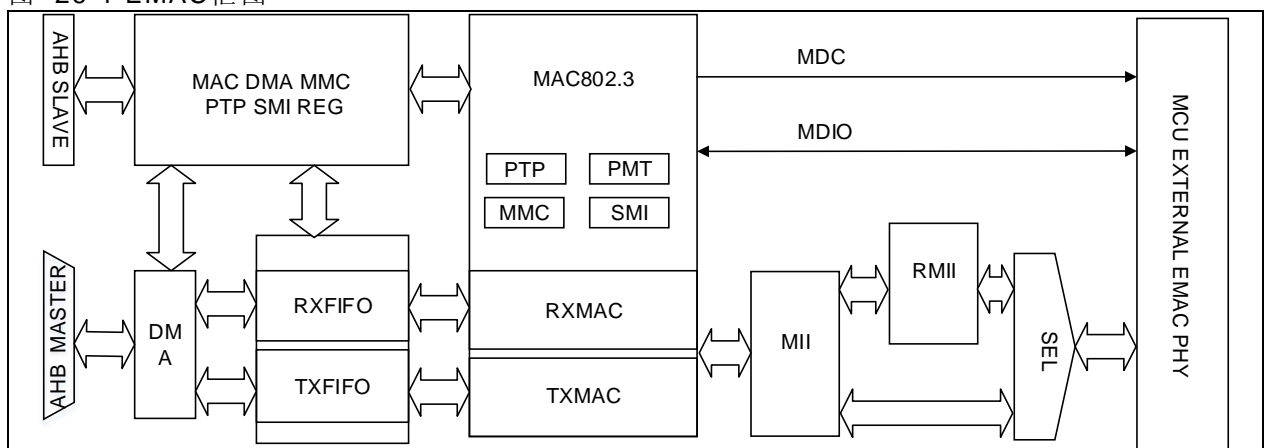
部分版权归 Synopsys 所有，经许可使用。

AT32F407 的以太网模块支持通过以太网收发数据(10/100Mbps)，符合 IEEE 802.3-2002 标准。

AT32F407 以太网模块支持两种标准接口连接到外接的 PHY：IEEE 802.3 协议定义的独立于媒体的接口 (MII)和简化的独立于媒体的接口(RMII)。

#### 26.1.1 EMAC总体结构图

图 26-1 EMAC框图



#### 26.1.2 EMAC主要特征

整个模块包括一个 EMAC CORE 和 DMA 控制器，帧的发送和接收通过 DMA 来调度。

##### DMA 特征

- AHB 突发传输的类型可通过软件配置
- 支持环结构或者链结构的描述符
- 每个描述符可以传输最高达 8K 字节的数据
- 发送或者接收通过轮询或者固定优先级的方式进行仲裁
- 多种工作状态下的对应的可配置中断
- 每一次传输都会有状态信息报告

##### EMAC CORE 特征

- 支持 10/100Mbps 的数据传输速率
- 通过兼容 IEEE 802.3 标准的 MII 接口，外接高速以太网 PHY
- 支持流控的全双工操作以及符合 CSMA/CD 协议的半双工操作
- 发送帧时可配置自动计算 CRC 和产生可控制的填充位
- 接收帧时可配置自动去除填充位/CRC
- 帧长度可配，最长为 16K 字节
- 帧间隙可配(40~96 位)
- 支持多种地址过滤模式以及混杂模式
- 支持检测接收到帧的 IEEE 802.1Q VLAN 标签
- 使用 RMON/MIB 计数器(RFC2819/RFC2665)进行强制网络统计
- 检测 LAN 远程唤醒帧和 AMD 的 Magic Packet™ 帧
- 支持对 IPv4 报头校验和以及对 IPv4 或 TCP、UDP 或 ICMP（IPv4 或 IPv6 数据格式封装）的校验和检查
- 支持由 IEEE 1588-2008 标准定义的以太网帧时间戳，在帧的接收或发送状态中记录 64 位的时间戳。



- 2 个 2K 字节的 FIFO，一个用来发送，一个用来接收，阈值均可独立配置
- 可以滤掉接收到的错误帧，并在存储-转发模式下，不向应用程序转发错误的帧
- 对于 MAC 控制器的数据传输，支持存储-转发机制
- 在延迟冲突、冲突过多、顺延过多和下溢情况下丢弃帧
- 可通过软件清空发送 FIFO
- 在存储-转发模式下，可对发送帧计算并插入 IPv4 的报头校验和及 TCP、UDP 或 ICMP 的校验和
- MII 接口下支持 loopback 模式，方便调试定位
- IEEE 1588-2008 标准定义的精确时间协议，可以设置接收和发送帧的时间戳
- 支持粗调和精调两种校正方法
- 输出秒脉冲（可配置）
- 系统时间达到预定时间时会触发中断

## 26.2 EMAC模块功能详述

以太网模块由 MAC 802.3(媒体访问控制器)、独立于媒体的接口(MII)和一个专用的 DMA 控制器组成。主要实现以下功能：

- 数据传送和接收
  - 帧的组装(帧间隔和帧同步)
  - 源地址和目的地址管理
  - 错误检测
- 半双工模式下介质访问管理
  - 介质分配(防止冲突)
  - 冲突解决(处理冲突)
 通常有两种模式可以操作 MAC 子层：
- 半双工模式：站点通过 CSMA/CD 算法来抢占对物理介质的访问，同一时间只有一个传输方向的两个站点可以进行信息传输
- 全双工模式：不使用 CSMA/CD 算法，但是全双工传输需要满足下面要求：
  - 物理介质支持同时进行收发操作
  - 只有两个站点接入LAN
  - 接入LAN的两个站点都配置为全双工模式

### 26.2.1 EMAC通信接口介绍

EMAC 支持配置 PHY 的站点管理接口 SMI，以及用于以太网帧通信的媒体接口 MII 和精简的媒体接口 RMII。

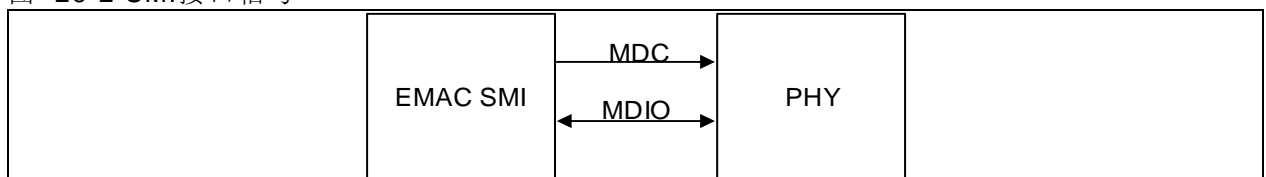
#### 站点管理接口(SMI)

PHY 管理接口(SMI)通过时钟和数据两根线来访问 PHY 寄存器。可以支持最多 32 个 PHY。

MDC: PHY 配置时钟信号，最高频率为 2.5MHz。时钟高电平和低电平的最小维持时间为 160ns，最小周期为 400ns。在空闲状态下 MDC 时钟信号保持在低电平状态。

MDIO: 双向口，数据的输入/输出线。

图 26-2 SMI接口信号



执行写操作前需要先配置 PHY 地址和 MII 寄存器以及以太网 MAC MII 数据寄存器(EMAC\_MACMIIDT)，然后设置 MII MW 和 MB 位，SMI 接口会向 PHY 传送 PHY 地址和 PHY 寄存器地址，然后传输数据。传输过程中 MB 位始终为 1，当完成写操作时，SMI 接口将清除 MB 位。需要注意的是，在传输过程中，以太网 MAC MII 地址寄存器(EMAC\_MACMIIADDR)和以太网 MAC MII 数据寄存器(EMAC\_MACMIIDT)信息是不能被更改的。



执行读操作前需要配置 PHY 地址和 MII 寄存器，然后配置以太网 MAC MII 地址寄存器（EMAC\_MACMIIADDR）MB 为 1，MW 位为 0，

SMI 接口则发送 PHY 地址和 PHY 寄存器地址，开始读 PHY 寄存器数据。传输过程中 MB 位始终为 1，当完成读操作时，SMI 接口将清除 MB 位。需要注意的是，在传输过程中，以太网 MAC MII 地址寄存器 (EMAC\_MACMIIADDR) 和以太网 MAC MII 数据寄存器 (EMAC\_MACMIIDT) 信息是不能被更改的（不能被应用程序写，EMAC\_MACMIIDT 在传输完成后会被自动更新为 SMI 读回的值）。

SMI 接口的时钟源由 AHB 时钟分频得到。根据 AHB 时钟频率的不同，需要选择不同的分频系数。注意必须要配置合适的分频系数，因为 MDC 的频率不能大于 2.5MHz。

下表列出了这个时钟范围。

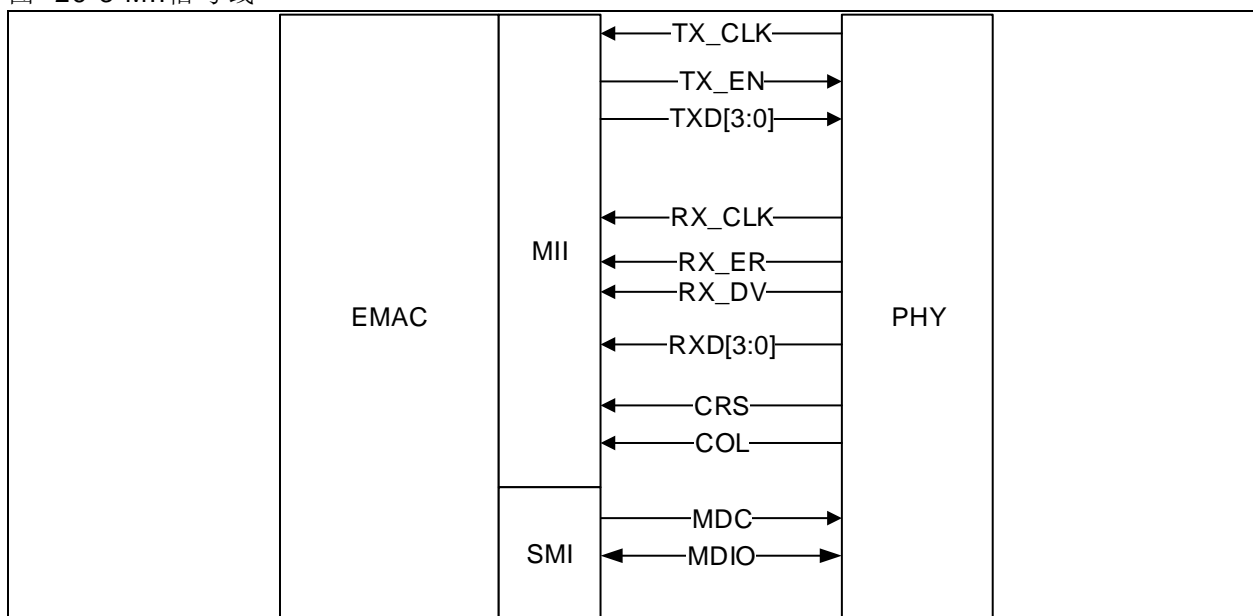
表 26-1 时钟范围

选择位	AHB 时钟	MDC 时钟
0000	60~100MHz	AHB 时钟/42
0001	100~150MHz	AHB 时钟/62
0010	20~35MHz	AHB 时钟/16
0011	35~60MHz	AHB 时钟/26
0100	150~240MHz	AHB 时钟/102
0101,0110,0111	保留	—

## 媒体接口 MII

独立媒体接口(MII)用于 MAC 子层和 PHY 之间的互联，允许 10M 位/s 和 100M 位/s 数据传输。

图 26-3 MII信号线



MII\_TX\_CLK: 发送数据时钟信号，对于 10Mbps 的数据传输，此时钟为 2.5MHz，对于 100Mbps 的数据传输，此时钟为 25MHz

MII\_RX\_CLK: 接收数据时钟信号，对于 10Mbps 的数据传输，此时钟为 2.5MHz，对于 100Mbps 的数据传输，此时钟为 25MHz

MII\_TX\_EN: 发送使能信号，此信号必需与数据前导符的起始位同步出现，并且必需一直保持到所有需要传输的位都传输完为止

MII\_TXD[3: 0]: 发送数据，每次同步地传输 4 位数据，数据在 MII\_TX\_EN 信号有效时有效。MII\_TXD[0] 是数据的最低位，MII\_TXD[3]是最高位。

MII\_CRS: CSMA/CD 定义的载波侦听信号，由 PHY 控制，只在半双工模式下有意义。当发送或接收的介质非空闲时，此信号有效。PHY 必需保证 MII\_CS 信号在发生冲突的整个时间段内都保持有效。不需要此信号与发送/接收的时钟同步。

MII\_COL: CSMA/CD 定义的冲突检测信号，由 PHY 控制，只在半双工模式下有意义。当检测到介质发生冲突时，使能此信号，并且在整个冲突的持续时间内，保持此信号有效。此信号不需要和发送/接收的

时钟同步。

**MII\_RXD[3: 0]:** 由 PHY 控制，每次同步地发送 4 位需要接收的数据，数据在 MII\_RX\_DV 信号有效时有效。MII\_RXD[0]是数据的最低位，MII\_RXD[3]是最高位。当 MII\_RX\_DV 无效而 MII\_RX\_ER 有效时，PHY 会传送一组特殊的 MII\_RXD[3: 0]数据来表示一些信息。

**MII\_RX\_DV:** 接收数据有效信号，由 PHY 控制，当 PHY 准备好数据供 MII 接收时，该信号有效。此信号必需和帧数据的首位同步(MII\_RX\_CLK)出现，并在数据完全传输完毕之前，都保持有效。在传送最后 4 位数据后的第一个时钟之前，此信号必需变为无效状态。为了正确的接收一个帧，MII\_RX\_DV 信号必需在整个帧传输期间内都保持有效，有效电平不能晚于数据线上的 SFD 位

**MII\_RX\_ER:** 接收出错信号，保持一个或多个时钟周期(MII\_RX\_CLK)的有效状态，指示 MAC 在帧内检测到了错误。具体错误信息需配合 MII\_RX\_DV 的状态以及 MII\_RXD[3: 0]的数据来指示。

表 26-2 发送接口信号编码

MII_TX_EN	MII_TXD[3: 0]	描述
0	0000 到 1111	正常的帧间隔
1	0000 到 1111	正常的数据传输

表 26-3 接收接口信号编码

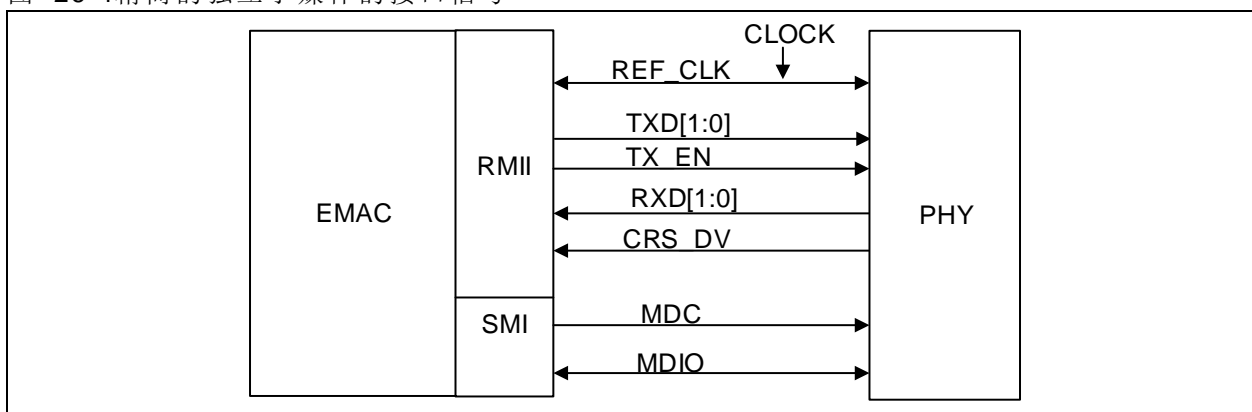
MII_RX_DV	MII_RX_ER	MII_RXD[3: 0]	描述
0	0	0000 到 1111	正常的帧间隔
0	1	0000	正常的帧间隔
0	1	0001 到 1101	保留
0	1	1110	失败的载波指示
0	1	1111	保留
1	0	0000 到 1111	正常的数据接收
1	1	0000 到 1111	数据接收出错

### 精简的媒体接口：RMII

精简的独立媒体接口(RMII)减少了 AT32F407xx 以太网模块和外部以太网之间的管脚数。根据 IEEE802.3u 标准，MII 接口需要 16 个数据和控制信号管脚，而 RMII 则将管脚数减少到了 7 个(减少了 62.5%的管脚数目)。

RMII 接口用于连接 EMAC 和 PHY，该模块将 MAC 的 MII 信号转换到 RMII 接口上。

图 26-4精简的独立于媒体的接口信号



### MII/RMII 的选择及时钟源

通过 IO 复用重映射寄存器 (IOMUX\_REMAP) 的第 23 位 MII\_RMII\_SEL 位，可以选择使用 MII 或者 RMII 模式。必需在以太网控制器处于复位状态时、或者在使能时钟前，选择好 MII/RMII 模式。

### MII 时钟源

EMAC TX\_CLK 和 RX\_CLK 时钟信号由 PHY 提供，外接的 PHY 模块需要由外部的 25MHz 时钟驱动。这个 25MHz 时钟可以由晶振或者 MCU CLKOUT 脚提供。使用 CLKOUT 脚时，需要正确的配置 PLL 或

者直接输出 HEXT，具体请参照 CRM 章节相关描述。

图 26-5 MII时钟源（CLKOUT脚提供时钟）

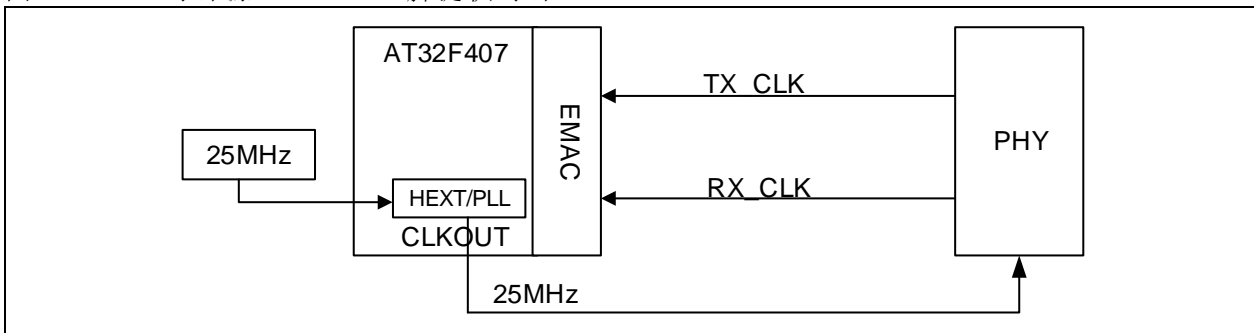
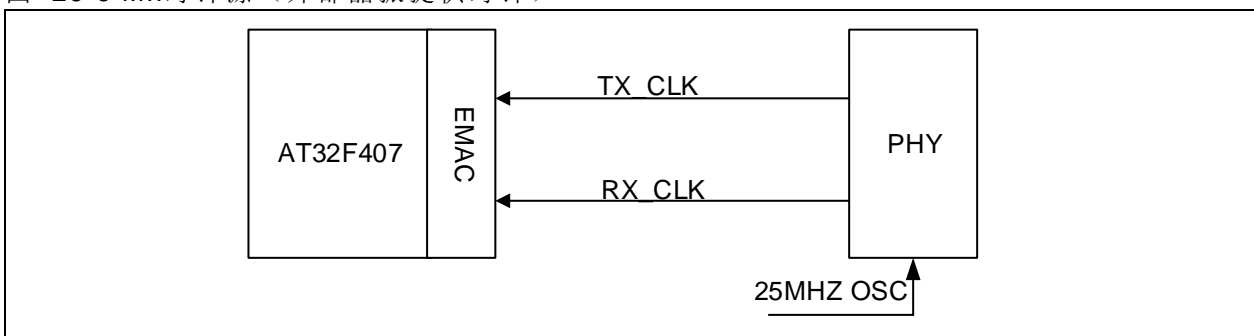


图 26-6 MII时钟源（外部晶振提供时钟）



## RMII 时钟源

如下图所示，EMAC 和 PHY 都需要 50MHz 时钟源，可以使用外部晶体振荡器或者 CLKOUT 脚提供 50MHz 参考时钟，具体请参考 CRM 相关章节。

图 26-7 RMII时钟源（CLKOUT脚提供时钟）

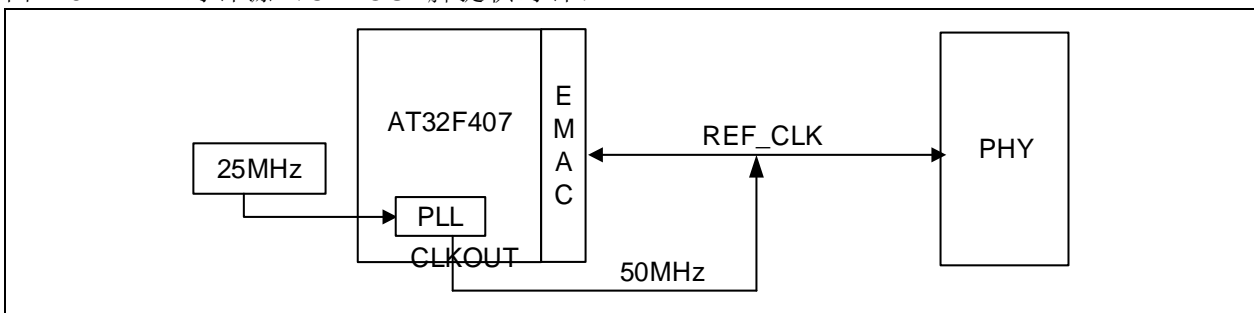
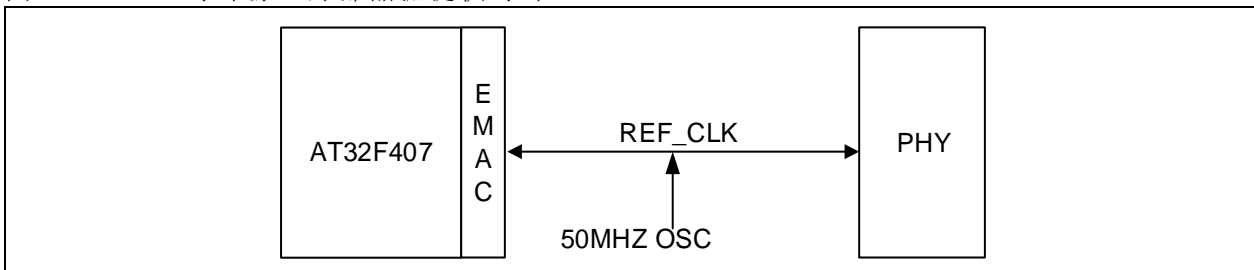


图 26-8 RMII时钟源（外部晶振提供时钟）



**EMAC 管脚分配及复用**

表 26-4 以太网模块管脚配置（黑色：默认 红色：重映射信号）

EMAC 信号	MII	RMII	管脚	管脚配置
EMAC_MDC	MDC	MDC	PC1	推挽复用输出
EMAC_MII_TXD2	TXD2	-	PC2	推挽复用输出
EMAC_MII_TX_CLK	TX_CLK	-	PC3	浮空输入(复位状态)
EMAC_MII_CRS	CRS	-	PA0	浮空输入(复位状态)
EMAC_MII_RX_CLK EMAC_RMII_REF_CLK	RX_CLK	REF_CLK	PA1	浮空输入(复位状态)
EMAC_MDIO	MDIO	MDIO	PA2	推挽复用输出
EMAC_MII_COL	COL	-	PA3	浮空输入(复位状态)
EMAC_MII_RX_DV EMAC_RMII_CRS_DV	RX_DV	CRS_DV	PA7	浮空输入(复位状态)
EMAC_MII_RXD0 EMAC_RMII_RXD0	RXD0	RXD0	PC4	浮空输入(复位状态)
EMAC_MII_RXD1 EMAC_RMII_RXD1	RXD1	RXD1	PC5	浮空输入(复位状态)
EMAC_MII_RXD2	RXD2	-	PB0	浮空输入(复位状态)
EMAC_MII_RXD3	RXD3	-	PB1	浮空输入(复位状态)
EMAC_MII_RX_ER	RX_ER	-	PB10	浮空输入(复位状态)
EMAC_MII_TX_EN EMAC_RMII_TX_EN	TX_EN	TX_EN	PB11	推挽复用输出
EMAC_MII_TXD0 EMAC_RMII_TXD0	TXD0	TXD0	PB12	推挽复用输出
EMAC_MII_TXD1 EMAC_RMII_TXD1	TXD1	TXD1	PB13	推挽复用输出
EMAC_PPS_OUT	PPS_OUT	PPS_OUT	PB5	推挽复用输出
EMAC_MII_TXD3	TXD3	-	PB8	推挽复用输出
EMAC_RMII_CRS_DV	RX_DV	CRS_DV	PD8	浮空输入(复位状态)
EMAC_MII_RXD0 EMAC_RMII_RXD0	RXD0	RXD0	PD9	浮空输入(复位状态)
EMAC_MII_RXD1 EMAC_RMII_RXD1	RXD1	RXD1	PD10	浮空输入(复位状态)
EMAC_MII_RXD2	RXD2	-	PD11	浮空输入(复位状态)
EMAC_MII_RXD3	RXD3	-	PD12	浮空输入(复位状态)

**26.2.2 EMAC帧通信****帧格式**

MAC 帧格式和带标签的 MAC 帧格式如下图所示(有关详细的 MAC 帧格式定义请见 IEEE 802.C-2002):

图 26-9 MAC帧格式

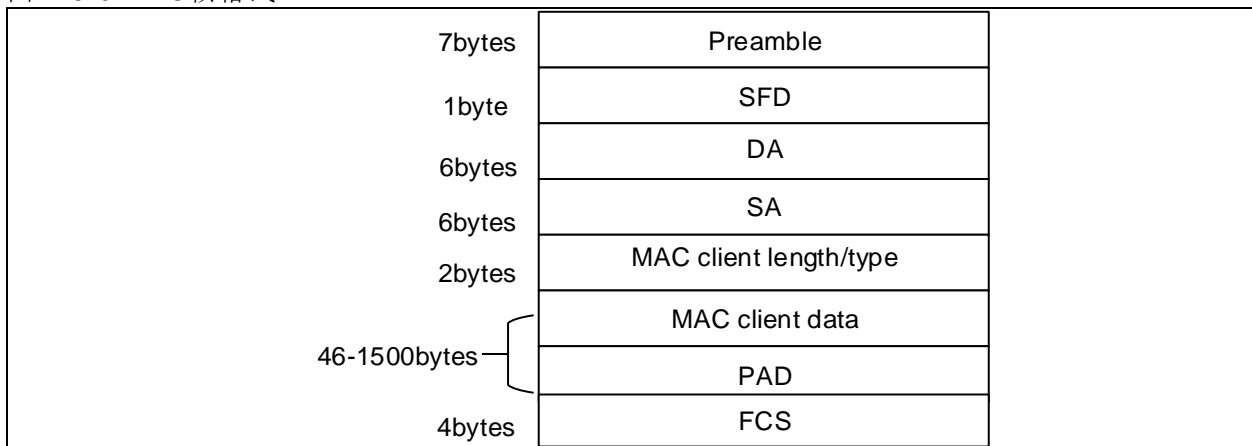
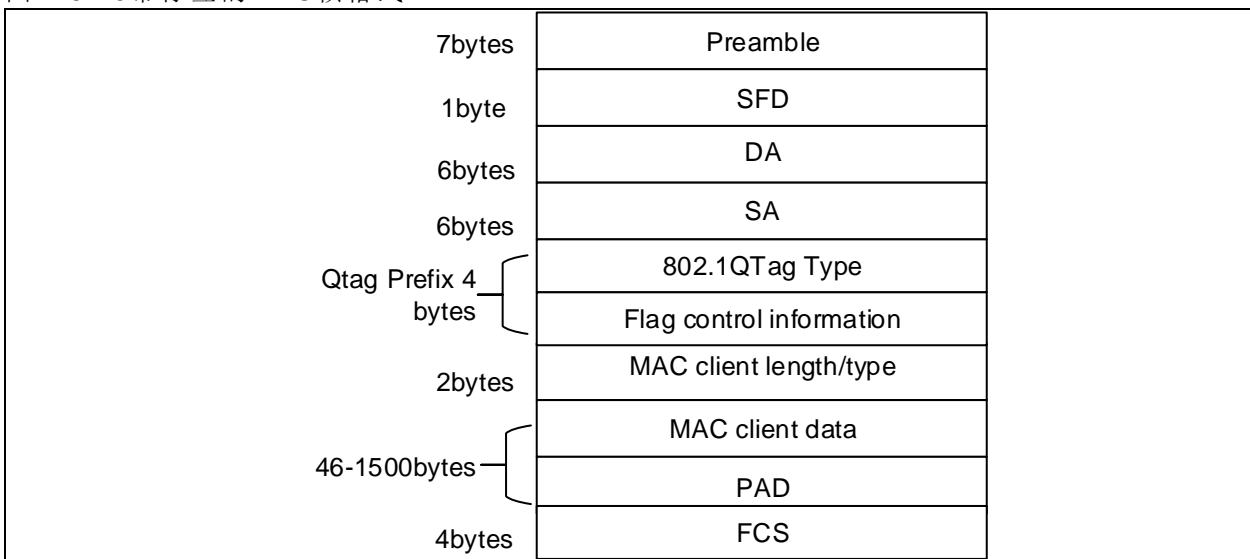


图 26-10带标签的MAC帧格式



### EMAC 帧过滤

EMAC 支持多种源地址和目的地址过滤。

#### 地址过滤

利用 MAC 地址和多播 HASH 列表对接收到的帧进行地址过滤，EMAC 根据应用程序设定的帧过滤寄存器对目的地址和源地址进行检验，并报告相应的地址过滤结果。

#### 单播目的地址滤波器

有完美地址过滤和 HASH 列表过滤两种方式。

1：完美地址过滤：完美地址过滤必须配置帧过滤寄存器 HUC 位 0。有 4 组 MAC 地址可以用来进行完美地址过滤，MACADDR0 始终是使能的，MACADDR1,MACADDR2, MACADDR3 可以配置相应的 AE 位来使能。可以通过配置地址寄存器 MBC 位来屏蔽相应字节的地址比对，如果 MBC 位全为 0，则 EMAC 会被接收到的帧的 48 位单播地址与设定好的 MAC 地址逐位进行比较，必须全部匹配，才能通过过滤。

2：HASH 列表过滤模式：需配置帧过滤寄存器 HUC 位为 1。EMAC 利用 64 位的 HASH 列表对单播地址进行不完美过滤。HASH 过滤时，MAC 计算接收帧的目的地址的 CRC 值(见下面的注释)，并取高 6 位作为索引检索 HASH 列表。CRC 值为'000000'对应 HASH 列表寄存器的位 0,CRC 值为'111111'对应 HASH 列表寄存器的位 63。如果 CRC 值对应的 HASH 列表上的相应位为'1'，说明该帧能通过 HASH 过滤器，否则该帧不能通过 HASH 过滤器。

#### 多播目的地址滤波器

1：如果帧过滤寄存器 PMC 位为 1，则 MAC 可以接收所有的多播帧

2：如果 PMC 位为 0，HMC 为 0，利用地址 MACADDR0, 1, 2, 3 进行完美地址过滤

3：如果 PMC 位为 0，HMC 位为 1，利用 64 位 HASH 列表进行不完美过滤

#### 广播地址滤波器

在帧过滤寄存器的 DBF 位为 1 的情况下，EMAC 拒收所有的多播帧，DBF 为 0 时，EMAC 接收任何广播帧。

#### 单播源地址过滤器

设置 MAC 地址寄存器 1,2,3 的位 30 为'1'，则在接收帧时过滤器将会比较源地址而不是目的地址。

如果帧过滤寄存器的 SAF 位为'1'，EMAC 会丢弃没能通过源地址(SA)过滤器的帧；而且在 SAF 位为'1'时，只有源地址过滤和目的地址过滤都通过，帧才会转发给应用程序。否则，帧将会被丢弃。

#### 颠倒过滤操作

通过设置帧过滤寄存器 DAIF 位和 SAIF 位为 1，可以对目的地址过滤和源地址过滤的输出结果进行颠倒。

DAIF 位作用于单播和多播帧的目的地址，SAIF 作用于单播帧的源地址。

表 26-5 目的地址过滤器结果列表

帧类型	PMC	HPF	HUC	DAIF	HMC	PMC	DBF	目的地址(DA)过滤器操作
广播帧	1	X	X	X	X	X	X	通过
	0	X	X	X	X	X	0	通过
	0	X	X	X	X	X	1	不通过
单播帧	1	X	X	X	X	X	X	所有帧通过
	0	X	0	0	X	X	X	完美/组过滤器匹配时通过
	0	X	0	1	X	X	X	完美/组过滤器匹配时不通过
	0	0	1	0	X	X	X	HASH 过滤器匹配时通过
	0	0	1	1	X	X	X	HASH 过滤器匹配时不通过
	0	1	1	0	X	X	X	HASH 或者完美/组过滤器匹配时通过
	0	1	1	1	X	X	X	HASH 或者完美/组过滤器匹配时不通过
多播帧	1	X	X	X	X	X	X	通过所有帧
	X	X	X	X	X	1	X	通过所有帧
	0	X	X	0	0	0	X	完美/组过滤器匹配时通过, 如果 PCF=0x, 丢弃暂停帧
	0	0	X	0	1	0	X	HASH 过滤器匹配时通过, 如果 PCF = 0x, 丢弃暂停帧
	0	1	X	0	1	0	X	HASH 或者完美/组过滤器匹配时通过, 如果 PCF = 0x, 丢弃暂停帧
	0	X	X	1	0	0	X	完美/组过滤器匹配时不通过, 如果 PCF = 0x, 丢弃暂停帧
	0	0	X	1	1	0	X	HASH 过滤器匹配时不通过, 如果 PCF = 0x, 丢弃暂停帧
	0	1	X	1	1	0	X	HASH 或者完美/组过滤器匹配时不通过, 如果 PCF = 0x, 丢弃暂停帧

表 26-6 源地址过滤器结果列表

帧类型	PR	SAIF	SAF	源(SA)过滤器操作
单播帧	1	X	X	所有帧通过
	0	0	0	传送完美/组过滤器匹配状态, 但不丢弃不通过的帧
	0	1	0	传送完美/组过滤器不匹配状态, 但不丢弃帧
	0	0	1	完美/组过滤器匹配时通过, 丢弃不通过的帧
	0	1	1	完美/组过滤器匹配时不通过, 丢弃不通过的帧

## EMAC 帧的发送

DMA 控制器和 MAC 控制以太网帧的发送。应用程序发出发送指令后, DMA 将以以太网帧从用户数据缓冲区 (比如 SRAM) 读出并存入 TXFIFO 中, 帧再弹出并传入 MAC 控制器, MAC 控制器通过 MII/RMII 接口传给外部的以太网 PHY 进而和外部的站点进行通信, 发送帧结束后, MAC 控制器会生成发送状态并返回到 DMA 控制器中。

当检测到 SOF 信号后, MAC 接收数据, 并开始传输数据至 MII/RMII。由于各种延迟因素, 比如帧间隙、发送前导码/起始定界符的时间、和半双工模式下由于 CSMA/CD 算法导致的回退等待等, 从应用程序启动传输, 到数据帧发送到接口的时间是不定的。当 EOF 信号传输到 MAC 控制器后, 控制器完成一次传输, 并生成发送状态到 DMA。

从发送 FIFO 弹出数据到 MAC 控制器的操作有两种模式:



- **直通模式：**当 FIFO 中的数据达到了设置好的阈值时(或者在达到阈值之前写入了 EOF)，数据会弹出 FIFO 并送入到 MAC 控制器中。直通模式下的阈值可以通过以太网 DMA 工作模式寄存器(EMAC\_DMAOPM)的 TTC 位来设置
- **存储—转发模式：**只有当一个完整的帧写入 FIFO 之后，数据才会被送入 MAC 控制器。如果发送 FIFO 的长度小于要发送的以太网帧，那么在发送 FIFO 即将全满时，数据也会被送入到 MAC 控制器

可以通过设置 FTF 位即以太网 DMA 工作模式寄存器(EMAC\_DMAOPM)的第 20 位来清空发送 FIFO 中的内容。异常情况下如果在帧数据传送到 MAC 控制器的过程中误设置了 FTF 位，FIFO 会被清空，发送过程会被强制终止。EMAC 发送器会标记发生了下溢事件，并生成相应的状态信息给 DMA 控制器。

#### 自动计算 CRC 和填充字节

IEEE802.3 协议规定以太网帧的最小数据长度为 46 字节，如果发送帧的数据长度不够，EMAC 可以配置成自动填充（填充数据是 0），那么在发送短帧时，数据长度会被填充达到 46 字节。

在帧发送时，EMAC 控制器可以配置成自动添加 CRC 到帧校验序列，也可以配置成不添加 CRC。但是如果 EMAC 控制器配置为当帧(DA+SA+LT+数据)少于 60 字节，自动添加填充字节时，不管是否配置了自动添加 CRC，都会添加 CRC 值到填充过的帧结尾。

#### CSMA/CD 定义的冲突检测

冲突检测仅在半双工模式下会起作用，全双工模式下无需冲突检测（详细原因请查以太网协议）。

在 MII 模式下，如果在帧传送的任意时刻外部 PHY 检测到了(从帧头到 CRC 的末位)冲突，冲突信号会发送给 EMAC 控制器，EMAC 控制器会发送一个 0x5555 5555 的 32 位堵塞信号，通知 LAN 上其他站点发生了冲突。如果冲突发生在传送前导码期间，EMAC 控制器会完成前导码和起始定界符的发送，然后立刻发送堵塞信号。

#### Jabber 定时器

在一个 LAN 上有可能一个站点会长时间的占用 LAN，为了避免这种情况，可以使能 EMAC jabber 定时器（设置 EMAC\_MACCTRL[20]位为 0）。使能后，如果应用程序试图发送超过 2048 字节的帧，EMAC 会停止发送。

#### 帧间隙管理

EMAC 会在符合帧间隙和退让延迟的情况下启动传输。在两个传输的帧之间，MAC 在帧间隙（以太网 MAC 配置寄存器(EMAC\_MACCTRL)的 IFG 位）内会保持空闲状态。一旦 MII 的载波信号消失，EMAC 即启动它的 IFG 计数器。如果一个帧早于配置好的 IFG 时间到达，则这个帧将会被延迟发送，直到达到帧间隙时间才能开始传输。一旦 MII 的载波信号消失，MAC 即启动它的 IFG 计数器。全双工模式下，配置好的 IFG 时间到时后，MAC 则使能传输。在半双工模式下，如果 IFG 时间配置为 96 个比特的时间，MAC 将遵循 IEEE 802.3 规范的 4.2.3.2.1 章节定义的规则。如果在整个 IFG 时间间隔的前三分之二(64 个比特时间)的时间段里检测到了载波，MAC 将复位 IFG 计数器。如果在 IFG 时间间隔的后三分之一的时段里检测到了载波，MAC 将继续 IFG 计数，并在达到 IFG 间隔时间后使能传输。在半双工模式下，MAC 遵循截断二进制指数退让算法。

#### 发送流控

全双工模式下 EMAC 通过 pulse 帧来实现发送流控。有两种情况 EMAC 可以通过发送暂停帧来请求发送端暂停发送以太网帧。

应用程序设置以太网 MAC 流控寄存器(EMAC\_MACFCTRL)的 FCB 位为'1'时，EMAC 会产生并发送一个单独的 Pause 帧。这个 Pause 帧指定的暂停时间为以太网 MAC 流控寄存器(EMAC\_MACFCTRL)中配置好的暂停时间值 PT。EMAC 可以取消之前剩余的暂停时间或者延长暂停时间，只需要再发送一个 pulse 帧即可（取消剩余的暂停时间需要设置 PT 为 0）

在使能流控时（EMAC\_MACFCTRL 位 ETF 为 1），当 RXFIFO 满时，EMAC 会产生并传输一个 Pause 帧。如果在达到配置好的暂停时间极限(以太网 MAC 流控寄存器(EMAC\_MACFCTRL)的 PLT)时，RXFIFO 仍然为满，MAC 会再传输一个 Pause 帧。如果接收 FIFO 一直为满，则将一直重复这个过程。当 RXFIFO 不为满而暂停时间仍未到时，MAC 会传输一个暂停时间为 0 的 Pause 帧，指示远程站点结束暂停，本地缓存区已经准备好接收新的数据帧

#### 遇到冲突时重新发送

半双工模式下，在帧发往 MAC 的时候有可能在 MAC 线路上发生冲突事件。这样 MAC 有可能在 FIFO 接收完这帧数据前就尝试重新发送。这时，如果重新发送开始，就需要再次把数据从 FIFO 中弹出。但是在 FIFO 控制器把 96 字节的数据弹出供 MAC 发送以后，就会释放这些数据的空间，允许 DMA 往那里推进新的数据。这意味着如果帧发出的数据数超过该门限(96 字节)或者 MAC 控制器提示迟到冲突事件时，不



能进行重新发送。

#### 清空发送 FIFO 操作

EMAC 可以通过设置以太网 DMA 工作模式寄存器(EMAC\_DMAOPM)的 FTF 位(位 20)来清空 TXFIFO。当配置 FTF 位后, EMAC 会立刻执行 TXFIFO 清空操作, 复位相应的指针为初始状态, 即使 TXFIFO 正在把一帧数据发送给 MAC 控制器。清空操作发生时 EMAC 会中止发送当前帧, 标记下溢事件, 生成状态信息 (TDES0 位 1 和位 13)。在应用程序(DMA)接收到全部被清空帧的状态信息字以后, 清空操作完成。随后以太网 DMA 工作模式寄存器(EMAC\_DMAOPM)的 FTF 位被清' 0'。这时, 允许 TXFIFO 从应用程序(DMA)接收新帧。清空操作后, FIFO 会忽略所有不以 SOF 起始符开头的的数据。

#### 发送状态信息以及时间戳

EMAC 控制器把以太网帧发送出去以后, 会把发送状态信息返回到应用程序。如果使能了 IEEE 1588 时间戳功能, 一个 64 位的时间戳会和发送状态信息一起写回到发送描述符中。

#### 发送校验和模块

以太网被广泛应用在收发封装有 TCP 或者 UDP 数据的 IP 数据包, 因此以太网控制器理所应当具有发送校验和的功能, 支持计算校验和, 并在发送时插入计算结果, 以及在接收时检测校验和错误。下面将详细介绍。

注意: 只有在 TXFIFO 设置成存储-转发模式的时候(以太网 DMA 工作模式寄存器(EMAC\_DMAOPM)的 TSF 位为' 1' )才能使用此功能。在数据帧发送到 EMAC 控制器发送端前, 必须保证发送 FIFO 的深度足够容纳一个完整的帧。如果 TXFIFO 的深度小于输入的以太网帧长度, 即便在存储-转发模式下, 校验和功能也是无效的, EMAC 只会计算和修改 IPv4 报头的校验和域。

想要了解 IPv4、TCP、UDP、ICMP、IPv6 和 ICMPv6 报头规范, 请分别查阅 IETF 规范 RFC791、RFC 793、RFC 768、RFC 792、RFC 2460 和 RFC 4443。

#### IP 头校验和

如果以太网帧的类型域值为 0x0800 并且 IP 数据包的版本域值为 0x4, 校验和模块就检测到 IPv4 数据包。EMAC 不管输入帧的校验和域内容, 直接以计算结果取而代之。IPv6 的报头不包含校验和域, 因此校验和模块不会改变 IPv6 报头的值, 而是通过发送状态信息的 IP 报头错误状态标志位(TDES0 位 16)反映这个报头的校验和计算结果; 在以太网类型域以及 IP 报头版本域取值不匹配, 或者以太网帧数据数目不够 IP 报头的的数据长度域时, 该标志位被置为' 1'。换言之, 该位在 IP 报头出现下列错误时被置为' 1':

- 对于 IPv4 数据包
  - 接收到的以太网类型域 0x0800, 但 IP 报头的版本域不等于 0x4
  - IPv4 报头长度域取值小于 0x5(20 字节)
  - 帧的总长度小于 IPv4 报头长度域的值
- 对于 IPv6 数据包
  - 接收到的以太网类型域为 0x86DD, 但 IP 报头的版本域不等于 0x6
  - 帧在完整接收 IPv6 报头(40 字节)或者扩展报头(扩展报头包含报头长度域)之前结束。即使校验和模块检测到这样的报头错误, 如果以太网类型域提示载荷数据为 IPv4 类型, 仍然会插入一个 IPv4 的报头校验和

#### TCP/UDP/ICMP 校验和

TCP/UDP/ICMP 校验和通过分析 IPv4 或 IPv6 报头(包括扩展报头)来判断封装的数据是 TCP 还是 UDP 或者 ICMP。

注意: 1: 对于非 TCP、UDP 或者 ICMP/ICMPv6 的有效数据, 校验和功能无效, 不改动帧的数据

2: 对于不完整的 IP 帧(IPv4 或者 IPv6), 包含安全功能的 IP 帧(如验证报头或者封装有安全数据), 以及带路由报头的 IPv6 帧, 校验和功能无效。

校验和模块会对 TCP、UDP 或者 ICMP 的数据进行计算, 并插入报头的相应域。它有以下 2 种工作模式:

1: 校验和计算不包括 TCP、UDP 或者 ICMPv6 的伪报头, 假设报头的校验和值就是输入帧的校验和域的数值。校验和域的内容会被包含在校验和的计算中, 并最终被计算的结果取代

2: 不管输入帧的校验和域的内容, 校验和的计算包括 TCP、UDP 或者 ICMPv6 的伪报头数据, 最终的计算结果将取代校验和域的原有内容

校验和计算完毕后, 结果会记录到发送状态信息的校验和错误状态位(TDES0 位 12)。当发生下述情况时, 数据校验和错误状态位置' 1':

1: 在存储-转发模式下, 在帧结尾没有写入 TXFIFO 的情况下, 帧就被转发给 MAC 发射端

2: 接收到的数据包少于 IP 报头数据长度域指示的字节数目

如果数据包长度大于给出的数据长度, 多余的数据会被当成充填字节而丢弃, 而不会报告错误。如果检测到第一类错误, 则不改变 TCP、UDP 或者 ICMP 报头。如果检测到第二类错误, 仍然会把校验和计算结果插入报头的相应域。

### EMAC 帧的接收

MAC 接收到的帧都会被送入 RXFIFO 中, 然后由 DMA 通过 AHB 接口送出。接收有 2 中模式, 分别是直通模式和存储-转发模式。

在直通模式(默认模式)下, 如果 RXFIFO 接收到帧数据长度大于门限设定值(可通过以太网 DMA 工作模式寄存器(EMAC\_DMAOPM)的 RTC 设置) 数据或者完整的帧, 就开始从 RXFIFO 中弹出数据, 并通知 DMA 接收。DMA 会持续从 RXFIFO 中弹出数据, 知道传输完整个数据。RXFIFO 转发完 EOF 后, 就会生成接收状态信息并将其发送给 DMA 控制器。

在存储-转发模式(通过以太网 DMA 工作模式寄存器(EMAC\_DMAOPM)的 RSF 位设置)下, 只有在 RXFIFO 接收到一个完整的帧后, DMA 才能将其读出。

如果 EMAC 设置成丢弃所有错误帧, 那么在两种不同的接收模式下将会有不同的行为:

1: 存储-转发模式下 DMA 只会读出合法的帧, 并转发给应用程序。

2: 直通模式下, 因为在帧结尾才会收到错误状态信息, 因此一部分错误的帧并没有被丢弃。

一旦 EMAC 在 MII 上检测到起始定界符就会启动接收过程。MAC 控制器在处理帧之前会剥离前导码和起始定界符。会通过过滤器检查帧的报头, 并用 FCS 域核对帧的 CRC 值。如果帧没能通过地址滤波器, EMAC 控制器就会丢弃该帧。

如果使能了 IEEE1588 时间戳, 会在检测到帧的起始定界符的时候记录下系统的当前时间。EMAC 会在接收完当前帧后把这个时间戳转交给应用程序。

如果接收到的“帧长度/类型”域的值小于 0x600, 并且把 EMAC 配置成 CRC/充填自动剥离, 那么 EMAC 会在向 RXFIFO 发送完“帧长度/类型”域规定数目的帧数据后, 开始丢弃剩下的字节(包括 FCS 域)。如果长度/类型域的值大于或等于 0x600, 不管自动 CRC 剥离选项怎么设置, MAC 都会把所有接收到的以太网帧数据发送到 RXFIFO。

EMAC 默认打开看门狗定时器, 长度大于 2048 字节的帧(DA + SA + LT + 数据 + 填充 + FCS)会被切断。可以通过以太网 MAC 配置寄存器(EMAC\_MACCTRL)看门狗使能位(WD)来关闭这项功能。不过如果帧的长度大于 16KB, 即使关闭了看门狗定时器, EMAC 仍然会切断当前以太网帧, 并报告一个看门狗超时事件。

### 接收校验和模块

通过设置以太网 MAC 配置寄存器(EMAC\_MACCTRL)的 IPC 位, EMAC 会检测 IPv4 和 IPv6 数据帧的完整性。EMAC 通过以太网帧的类型域值来判别当前帧是 IPv4 帧还是 IPv6 帧。接收校验和模块会计算 IPv4 报头的校验和, 检查它是否与 IPv4 报头的校验和域的内容匹配。对报头的校验和结果体现在接收描述符 RDES0 的第 7 位, 有三种情况会导致 IP 报头错误为被置为 1:

1: 以太网类型域值指示的数据类型与 IP 报头版本域不匹配

2: 接收到的帧长少于 IPv4 报头长度域指示的长度

3: IPv4 或 IPv6 报头少于 20 字节

接收校验和模块还能识别 IP 数据包的数据类型是 TCP、UDP 还是 ICMP, 并按照 TCP、UDP 或 ICMP 的规范计算它们的校验和。计算包括了 TCP/UDP/ICMPv6 伪报头的数据, 数据的校验结果体现在接收描述符 RDES0 第 0 位。该位在下面两种情况下会被置 1:

1: 收到的 TCP、UDP 或者 ICMP 数据长度与 IP 报头给出的长度不符

2: 计算的校验和和 TCP、UDP 或者 ICMP 校验和域的值不相等

### 接收流控

在全双工模式下, MAC 能够检测 PAUSE 帧, 并按照 PAUSE 帧中的参数, 在一定时间内暂停 发送数据。

设置以太网 MAC 流控寄存器(EMAC\_MACFCTRL)的 ERF 位, 可以使能或者取消 PAUSE 帧检测功能。

如果使能了接收流控功能, EMAC 就会对暂停帧进行解码;

在暂停期间, 如果接收到另一个 PAUSE 时间为 0 的 PAUSE 帧, MAC 会清除 PAUSE 的时间, 重新开始发送数据。如果不为 0, 则帧中的暂停时间会立刻加载到暂停时间计数器中。

### 多个帧的接收处理

由于帧的状态信息紧随在帧数据之后。因此只要 RXFIFO 未满, 就可以存放任意数量的帧。

### 接收状态信息字

在接收以太网帧结束时，EMAC 会把接收状态信息发给应用程序(DMA)。接收状态信息的具体含义详见 RDES0。

#### EMAC 自循环模式

EMAC 支持 loopback 模式：发射端把帧发送到自己的接收端上。loopback 模式可以非常方便的用来调试以太网通信，需要通过设置以太网 MAC 配置寄存器(EMAC\_MACCTRL)的 LM 位置'1'来打开这个功能。注意 loopback 模式只针对 MII 接口。

#### EMAC 帧计数器

MMC 是收集发送和接收到帧的统计寄存器。主要包括以太网 MMC 控制寄存器(EMAC\_MMCTRL)，以太网 MMC 接收中断寄存器(EMAC\_MMCRIR)和以太网 MMC 接收中断屏蔽寄存器(EMAC\_MMCRIM)，以太网 MMC 发送中断寄存器(EMAC\_MMCTIR)和以太网 MMC 发送中断屏蔽寄存器(EMAC\_MMCTIM)。

如果一个帧的发送过程因为发生错误被中止，则这个帧将不会被统计，错误主要有以下几种：

- 1: 没有载波或者载波丢失
- 2: Jabber 超时
- 3: 迟到冲突
- 4: 冲突过多
- 5: 顺延(Deferral)过多
- 6: 帧溢出

如果一个帧的接收过程因为发生错误被中止，则这个帧将不会被统计，错误主要有以下几种：

- 1: CRC 错误
- 2: 过短帧(少于 64 字节)
- 3: 对齐错误
- 4: 长度错误(长度域值域接收到帧长不符)
- 5: 超出范围(帧长超过上限，非标签帧最大为 1518 字节，标签帧最大为 1522 字节)
- 6: MII\_RXER 输入错误

### 26.2.3 专用DMA对以太网帧的传输调度

EMAC 通过内部专用的 DMA 来调度以太网帧的发送和接收。

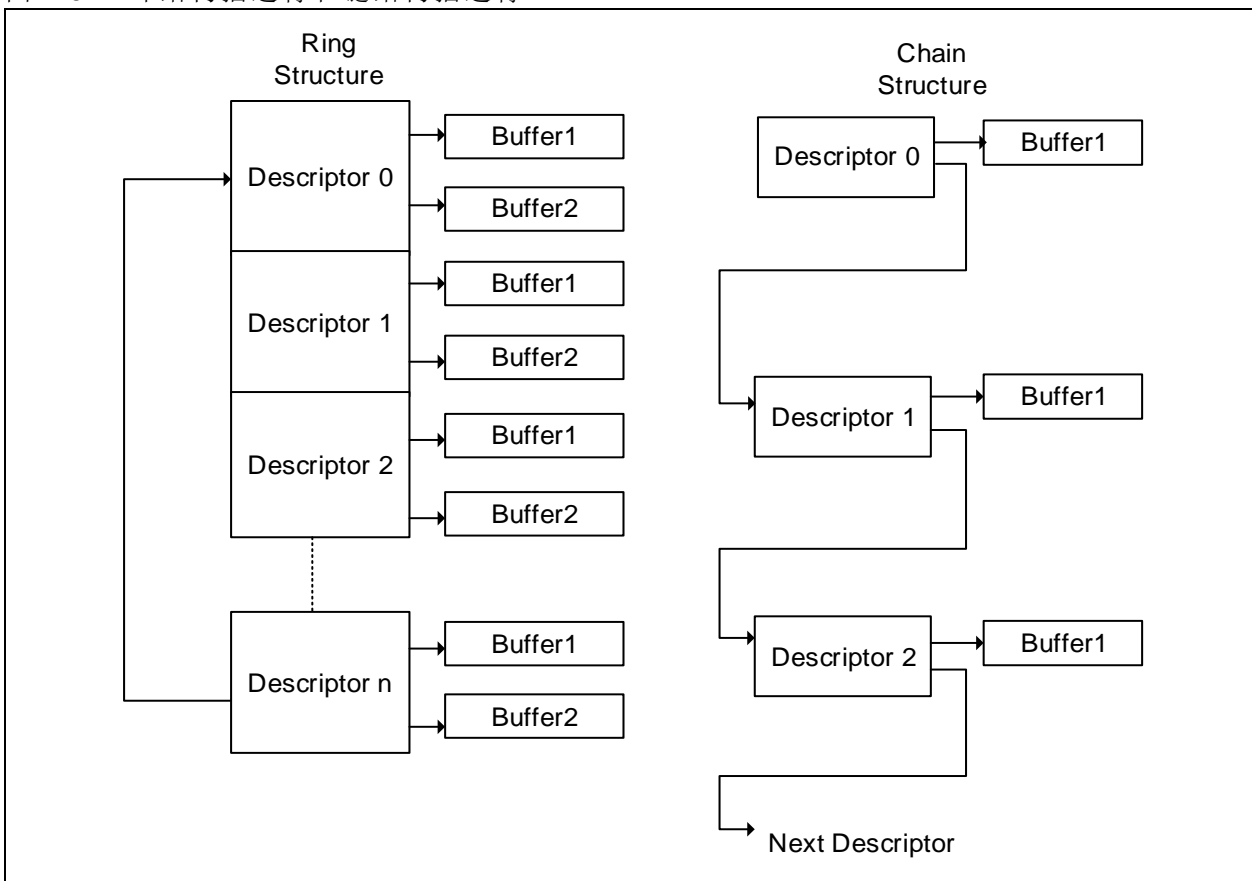
对于发送流程，DMA 会通过 AHB 主接口从用户数据缓冲区(比如 SRAM)读出以太网帧并送给 TXFIFO，EMAC 控制器根据配置把 TXFIFO 的数据成帧的发送到 MII/RMII 接口。

对于接收流程，EMAC 控制器会把从 MII/RMII 接口接收的以太网帧送到 RXFIFO，然后 DMA 会把 RXFIFO 中的以太网帧读出并通过 AHB 接口发送给用户数据缓冲区(比如 SRAM)。

DMA 通过 DMA 控制和状态寄存器以及描述符列表来统筹整个发送和接收流程。发送和接收流程均有对应的描述符列表。描述符列表一般存放在系统缓存中(SRAM)，DMA 在发送和接收启动时会通过发送和接收查询寄存器轮询描述符列表以开始发送和接收过程。发送和接收的描述符列表的基地址存放在发送描述符列表寄存器和接收描述符列表地址寄存器。

描述符有环结构和链结构两种形式。描述符组成环结构时，每个描述符可以指向两个帧的缓冲区。在链结构(对于发送需配置 TDES0[20]为 1，对于接收需配置 RDES1[14]为 1)下，每个描述符只能指向一个缓冲区，当前帧的 TDES3 和 RDES3 的内容代表下一个发送和接收描述符的地址。

图 26-11 环结构描述符和链结构描述符



### DMA AHB 主机突发访问

当设置以太网 DMA 总线模式寄存器(EMAC\_DMABM)的 FB 位为'1' 时, DMA 会在 AHB 主接口上执行固定长度的突发访问。最大突发长度由 PBL 域(以太网 DMA 总线模式寄存器(EMAC\_DMABM)位[13: 8])定义。对接收和发送描述符这 16 字节数据的访问总是以 PBL 限定的最大可能突发长度进行。

DMA 会向 AHB 主接口提供起始地址和要传送的字节数量, 然后开始一次传输。

注意, 对于发送流程还要满足下面条件之一:

- 1: TXFIFO 空间大于预设的突发长度
- 2: 帧结束前的数据字节数小于突发长度而且 TXFIFO 能容纳这些字节数

对于接收流程要满足下面条件之一:

- 1: RXFIFO 里的可用数据多于预设的突发长度
- 2: 帧结束之前的数据字节数小于突发长度且在接收 FIFO 里检测到帧结束

### AHB 主机数据缓存对齐

DMA 发起传输的时候, 总是从与总线宽度对齐的地址开始, 但是缓存的起始地址可以对齐到 4 个字节中的任意一个。

- 读缓存示例: 如果发送缓存的地址为 0x2000 0AA3, 并需要传输 15 字节。开始读操作后 DMA 实际会从地址 0x2000 0AA0 开始读 5 个字(32 位), 但是在往 TXFIFO 发送数据的时候, 会忽略头 3 个字节和后 2 个字节。除非 DMA 传输的是帧的结尾, 否则 DMA 总是保证向 TXFIFO 传输 32 位的数据。

- 写缓存示例: 如果接收缓存的地址为 0x2000 0BB2, 并需要传输 16 字节长的帧。开始写操作后 DMA 实际会从地址 0x2000 0BB0 开始写 5 个 32 位数据。但是传输的头 2 个字节和末尾的 2 个字节是虚拟字节。

### 缓冲区大小计算

针对发送过程, 软件需要计算缓存的大小。TXDMA 会传输与 TDES1 缓存大小域相等数目的字节给 EMAC 控制器。如果当前描述符 TDES0 的 FS 位为'1', 那么 DMA 就标记从这个缓存的第一次发送是帧首。如果当前描述符 TDES0 的 LS 位为'1', 则 DMA 就标记从这个缓存的最后一次发送是帧的帧尾。

接收帧的过程中, 如果接收缓存地址是字对齐的, 缓存的有效长度为 RDES1 中配置的值。如果接收缓存地址是非字对齐的, 则缓存的有效长度将小于 RDES1 中配置的值。缓存有效长度值应为 RDES1 中配置



的值减去缓存地址的低 2 位值。例如，假设缓存的总大小为 1024 字节，缓存地址为 0x2000 0001，地址的低 2 位值为 0x01，那么缓存有效长度为 1023 个字节。

当收到了一个帧起始 SOF，则 DMA 控制器将 FS 位置位，当收到一个帧结束 EOF 时，则 LS 位被置位。如果接收缓存长度域值配置的足够大，能放下整个帧，则 FS 和 LS 位将在同一个描述符中被置位。实际接收的帧长度可从 RDES0 的 FL 位域获取。

### DMA 仲裁器

有两种仲裁方式来仲裁 DMA 的发送和接收控制器：轮询和固定优先级。选择轮询仲裁方式时(以太网 DMA 总线模式寄存器(EMAC\_DMABM)DA 位为'0')，在发送和 RXDMA 同时要求访问 AHB 总线的时候，按照以太网 DMA 总线模式寄存器(EMAC\_DMABM)PR 位设定的比例分配总线。在 DA 位为'1'时，RXDMA 总是比 TXDMA 对总线拥有更高的访问优先级。

### DMA 错误响应

如果 DMA 在传输过程中接收到了错误的总线响应，那么 DMA 会停止所有操作，并更新以太网 DMA 状态寄存器(EMAC\_DMASTS)的错误位和总线致命错误位。要想 DMA 重新工作，必须通过软件或者硬件复位以太网外设并重新初始化 DMA。

### DMA 初始化

- 1: 在以太网 DMA 总线模式寄存器(EMAC\_DMABM)中设置 AT32F407xx 总线访问相关参数。
- 2: 在以太网 DMA 中断使能寄存器(EMAC\_DMAIE)中屏蔽不需要的中断源。
- 3: 应用程序生成发送和接收描述符列表并写入系统缓存，然后把描述符列表的起始地址写入以太网 DMA 接收描述符列表地址寄存器(EMAC\_DMARDLADDR)和以太网 DMA 发送描述符列表地址寄存器(EMAC\_DMATDLADDR)。
- 4: 配置地址过滤寄存器。
- 5: 在 MAC 的以太网 MAC 配置寄存器(EMAC\_MACCTRL)中设置并使能发送和接收操作。根据自动协商(auto-negotiation)的结果，设置 PS 位和 DM 位的值。
- 6: 设置以太网 DMA 工作模式寄存器(EMAC\_DMAOPM)的位 13 和位 1 为'1'，开始发送和接收。
- 7: DMA 发送和接收控制器开始从对应的描述符列表里读取描述符并开始处理接收和发送操作。

### TXDMA 操作：非 OSF 模式

在默认模式下，TXDMA 的工作流程如下：

1. 应用程序建立以太网帧数据缓存以及发送描述符列表(TDES0-TDES3)，并置OWN位(TDES0[31])为'1'。
2. 设置SSTC位(以太网DMA工作模式寄存器(EMAC\_DMAOPM)位[13])为'1'后，DMA启动发送。
3. DMA查询发送描述符来获取待发送的帧，如果DMA检测到标志位提示CPU正在操作描述符或者发生了错误，就会终止传输进入挂起状态，并设置发送缓存不可用位(以太网DMA状态寄存器(EMAC\_DMASTS)位2)和正常中断总结位(以太网DMA状态寄存器(EMAC\_DMASTS)位16)为'1'。发送控制器操作跳至步骤8。
4. DMA根据发送描述符的指示从AT32F407xx存储器里取数据，并把数据发送给TXFIFO。
5. 如果以太网帧的数据存放在属于不同描述符的不同缓存里，DMA会关闭存放中间数据的描述符，并取下一个缓存对应的描述符，重复步骤3、4、5直到发送完帧结尾的数据。
6. 在帧发送完成以后，如果发送状态信息显示这个帧使能了IEEE1588时间戳功能，那么DMA在更新发送状态信息给TDES0的同时会把时间戳的值写入存放帧尾的缓存对应的发送描述符(TDES2和TDES3)。然后把OWN位清'0'，关闭当前描述符。如果这个帧没有打开时间戳功能，那么DMA只会更新状态信息到TDES0，不会记录时间戳。
7. 在帧发送完成以后，如果描述符的完成中断位(TDES0[30])为'1'，则发送中断位(以太网DMA状态寄存器(EMAC\_DMASTS)位[0])会被置'1'。然后DMA控制器返回步骤3，准备发送下一帧数据。
8. 在暂停状态下，如果DMA收到发送查询的请求并且溢出中断标志位为'0'，DMA就会尝试重新获取描述符(因此会回到步骤3)。

### TXDMA 操作：OSF 模式

此模式中需要设置 OSF 位(以太网 DMA 工作模式寄存器(EMAC\_DMAOPM)位 2)为'1'。在这种模式下，DMA 在发送完当前帧数据后，不必等状态信息写回就去直接查询第二帧数据的发送描述符。

1. 按照TXDMA默认模式的步骤1-6操作。
2. DMA不等前一帧最后一个描述符状态信息更新，直接取下一个描述符。

- 如果DMA占有描述符，那么就解析发送缓存的地址。如果DMA不占有这个描述符，则进入挂起状态并跳到步骤7。
- DMA从AT32F407xx的存储器中取出发送帧的数据并发送直到帧尾发送完毕，如果帧数据分散在多个缓存中，DMA关闭存放在中间数据缓存对应的描述符。
- DMA等待前一帧的发送状态信息和时间戳，在接到状态信息后，如果标志位提示记录了帧的时间戳，DMA把时间戳写入TDES2和TDES3。DMA再清零OWN位关闭该描述符。如果这个帧没有打开时间戳功能，那么DMA不会更改TDES2和TDES3的内容。
- 如果使能中断，发送中断位置'1'，DMA在状态信息正常的情况下继续取下一个描述符，并跳到步骤3。如果前一个发送状态信息显示有数据下溢错误，DMA进入暂停状态，并跳到步骤7。
- 在暂停状态下，如果DMA收到一个待处理的状态信息和时间戳，那么如果这个帧使能了时间戳，DMA就把时间戳写入TDES2和TDES3，然后把状态信息写入TDES0。随后，设置相关的中断标志位并回到暂停状态。
- 只有在收到发送查询请求以太网DMA当前发送描述符寄存器(EMAC\_DMACTD)后，DMA才会退出暂停状态并进入运行状态，并根据是否有待处理的状态信息跳到步骤1或者步骤2。

### 发送帧处理

发送缓存里存放的以太网帧数据必须包括目的地址，源地址，正确的类型/长度域值，以及数据。至于是否包括CRC值，要看发送描述符的设置。如果描述符要求EMAC控制器关闭CRC和插入填充功能，那么缓存里还应当包含CRC。

一个帧可以存放在不同缓存里，用数据链的形式连接起来。传输开始时，第一个描述符的TDES0位28应为'1'，DMA则会把数据从内存送到TXFIFO中。如果TDES0位29是'1'，表示这个缓存存放的是帧结尾的数据，是这个帧的最后一个缓存。在这最后一个缓存的数据发送完毕以后，DMA写回发送状态信息到TDES0。如果发送完成中断位(TDES0[30])为'1'，则设置发送中断位(以太网DMA状态寄存器(EMAC\_DMASTS)位0)为'1'。DMA随后取下一个描述符，重复上述步骤。实际的帧发送还需要看配置的是存储-转发模式或者是阈值模式。在DMA传输完帧以后，则会关闭描述符(TDES0[31]清'0')。

### 发送查询暂停

在下列情况下，可以暂停DMA查询描述符：

DMA检测到CPU占有描述符(TDES0[31]=0)，DMA会进入挂起状态。

在检测到数据下溢错误时，DMA会中止帧的传输，并设置异常中断总结位(以太网DMA状态寄存器(EMAC\_DMASTS)位15)和发送数据下溢位(以太网DMA状态寄存器(EMAC\_DMASTS)位5)为'1'，并更新TDES0里的相应错误位为'1'。

### TXDMA 描述符

描述符结构体包含4个32位字，TDES0、TDES1、TDES2和TDES3的位定义如下图：

图 26-12 传输描述符

31										0														
TDES0	OWN	控制 [30:26]				T T S E	保留 24	控制 [23:20]				保留 [19:18]				T T S S	状态[16:0]							
TDES1	保留 [31:29]				缓冲区2字节计数[28:16]										保留 [15:13]				缓冲区1字节计数 [12:0]					
TDES2	缓冲区1地址[31:0]/ 时间戳低位[31:0]																							
TDES3	缓冲区2地址[31:0]或下一个描述符地址[31:0]/ 时间戳高位[31:0]																							

## TDES0: 发送描述符字 0

在初始化描述符的时候需要编写控制位[30: 26]+位[23: 20]和 OWN 位。在 DMA 更新, 或者说写回描述符的时候, 会清除所有的控制位和 OWN 位, 只报告状态信息标志位。

域	简称	类型	功能
位 31	OWN	rw	占有位 (Own bit) 0: CPU 占有描述符。 1: DMA 占有描述符。 DMA 在传输完整个帧或者这个缓存里的数据全部读出这两种情况下会把该位清'0'。每个帧的第一个缓存描述符的占有位, 必须在后面缓存描述符的占有位全部置'1'以后, 才能置'1'。
位 30	IC	rw	完成时中断 (Interrupt on completion) 配置为'1'时, 在当前帧发送完成以后, 会把传输中断位(以太网 DMA 状态寄存器(EMAC_DMASTS)位[0])置'1'。该位仅在 LS 位为 1 时有效。
位 29	LS	rw	最后分块 (Last segment) 置'1'时, 表示当前描述符指向的缓存存放着帧的最后一个分块。此位为 1 时, TDES1 中 TBS1 或者 TBS2 的值不能为 0。
位 28	FS	rw	第一分块 (First segment) 置'1'时, 表示当前描述符指向的缓存存放着帧的第一个分块。
位 27	DC	rw	不计算 CRC (Disable CRC) 置'1'时, MAC 不再在发送帧结束时插入 CRC 域。该位只有在 FS 位(TDES0[28])为'1'时才有效。
位 26	DP	rw	不填充 (Disable pad) 0: MAC 对帧长不足 64 字节的帧自动填充字节, 而且无论 DC 位(TDES0[27])为何值 MAC 都会在帧结尾插入 CRC 数值。该位只在 FS 位(TDES0[28])为'1'时有效。 1: MAC 不对帧长不足 64 字节的帧自动填充字节。
位 25	TTSE	rw	发送时间戳使能 (Transmit time stamp enable) 置'1'时, 这个描述符对应的发送帧会打开 IEEE1588 硬件时间戳功能。该位只在 TSE 位(EMAC_PTPTSCTRL[0])和 FS 位(TDES0[28])为'1'时有效。
位 24	保留	resd	请保持默认值。
位 23: 22	CIC	rw	校验和插入控制 (Checksum insertion control) 这 2 位控制校验和的计算和插入, 详述如下: 00: 不使能插入校验和功能; 01: 仅使能 IP 报头的校验和计算和插入; 10: 使能 IP 报头和数据域的校验和计算和插入, 但是不计算伪报头的校验和; 11: 使能 IP 报头和数据域的校验和计算和插入, 同时也计算伪报头的校验和。
位 21	TER	rw	环形发送结束 (Transmit end of ring) 置'1'时, 表示到达描述符列表的最后一个描述符。DMA 返回描述符列表的起始地址, 形成描述符环。
位 20	TCH	rw	第二地址链表 (Second address chained) 置'1'时, 表示 TDES1 中 TBS2 代表的是下一个描述符的地址, 而不是第二个缓存的地址。TDES0[21]的功能优先于本位(TDES0[20])。该位只在 FS 位(TDES0[28])为'1'时才有效。
位 19: 18	保留	resd	请保持默认值。
位 17	TTSS	rw	发送时间戳状态 (Transmit time stamp status) 此位作为状态位用来表明发送帧的时间戳是否被记录。置'1'时表示记录下了描述符对应的发送帧的时间戳, 记录的时间戳放在 TDES2 和 TDES3 处。该位只在 LS 位(TDES0[29])为'1'时才有效。
位 16	IHE	rw	IP 报头错误 (IP header error) 该位置'1'时表示 MAC 发送端检测到了 IP 数据包报头的错误。对于 IPv4 数据包, MAC 会检查数据包中的长度域和接收到的 IPV4 数据字节数是否相等, 如果不相等则报错。对 IPv6 数据包, 如果主报头长度不是 40 字节时则报错。此外, IPv4 或者 IPv6 帧的长度/类型域值和报头的版本信息必须匹配。对 IPv4 帧, 如果报头长度域值小于 0x5, 该位也置'1'报错。
位 15	ES	rw	错误汇总 (Error summary) 该位为下列位的逻辑“或” TDES0[14]: 啰嗦(Jabber)超时; TDES0[13]: 帧清空; TDES0[11]: 载波丢失; TDES0[10]: 无载波; TDES0[9]: 迟到冲突;



			TDES0[8]: 冲突过多; TDES0[2]: 顺延(Deferral)过多; TDES0[1]: 下溢错误; TDES0[16]: IP 报头错误; TDES0[12]: IP 数据错误。
位 14	JT	rw	啰嗦超时 (Jabber timeout) 置'1'时, 该位表示 MAC 发送端发生了啰嗦超时。该位只在以太网 MAC 配置寄存器(EMAC_MACCTRL)的 JAD 位不为'1'时才会发生在发生啰嗦超时时被置'1'。
位 13	FF	rw	帧清空 (Frame flushed) 置'1'时, 该位表明由于 CPU 发出帧清空命令, DMA 或者 MTL 把帧从 FIFO 中清空。
位 12	IPE	rw	IP 数据错误 (IP payload error) 置'1'时, 该位表明 MAC 发送端检测到了 TCP、UDP 或者 ICMP 的 IP 数据错误。发送端会把接收到的 IPv4 或者 IPv6 报头中的长度域和实际接收到的 TCP、UDP 和 ICMP 字节数目做比对, 如果不相等就置'1'报错。
位 11	LOC	rw	载波丢失 (Loss of carrier) 置'1'时, 该位表明在帧发送的过程中发生了载波丢失(MII_CRS 信号在发送过程中存在一个或多个以上发送时钟周期的无效状态)。该位只有在半双工模式下且发送帧没有冲突时有效。
位 10	NC	rw	无载波 (No carrier) 置'1'时, 表示在帧发送的过程中从 PHY 发过来的载波侦听信号没有置起。
位 9	LC	rw	迟到冲突 (Late collision) 置'1'时, 该位表明因为在发送过程中在冲突检测窗口检测到冲突(MII 模式下, 包括前导符的 64 个字节时间)而中止发送。在下溢错误位置'1'的情况下该位无效。
位 8	EC	rw	冲突过多 (Excessive collision) 置'1'时, 该位表明 MAC 在尝试发送当前帧的过程中因为连续发生了 16 次冲突而中止发送。如果以太网 MAC 配置寄存器(EMAC_MACCTRL)的 RD(不进行重试)位为'1', 那么在发生第一次冲突后, 该位就置'1', 并中止发送。
位 7	VF	rw	VLAN 帧 (VLAN frame) 置'1'时, 该位表明发送的帧是 VLAN 类型帧。
位 6: 3	CC	rw	冲突计数 (Collision count) 该 4 位值记录了帧发送出去前出现的冲突次数。在 EC 位(TDES0[8])为'1'时, 该位无效。这位仅在半双工模式下有效。
位 2	ED	rw	顺延过多 (Excessive deferral) 置'1'时, 表示在以太网 MAC 配置寄存器(EMAC_MACCTRL)的 DC 位为'1'时, 因为顺延超过 24288 位时而结束发送。
位 1	UF	rw	数据下溢错误 (Underflow error) 置'1'时, 该位表示从系统缓存送到 MAC 的数据过迟, 导致 MAC 中止帧发送过程。下溢错误表示 DMA 在发送帧的过程中遇到了空的发送缓存。发送进程进入挂起状态, 并设置发送下溢位(以太网 DMA 状态寄存器(EMAC_DMASTS)位 5 TU)和发送中断位(以太网 DMA 状态寄存器(EMAC_DMASTS)位 0 TX)置'1'。
位 0	DB	rw	顺延位 (Deferred bit) 置'1'时, 该位表示 MAC 在发送前因为监测到载波信号而推迟帧发送。该位只在半双工模式下有效。

## TDES1: 发送描述符字 1

域	简称	类型	功能
位 31: 29	保留	resd	请保持默认值。
位 28: 16	TBS2	rw	发送缓存 2 大小 (Transmit buffer 2 size) 第二个数据缓存的大小(以字节记)。如果 TDES0[20]位为'1'时, 这些位表示第二个描述符的地址。
位 15: 13	保留	resd	请保持默认值。
位 12: 0	TBS1	rw	发送缓存 1 大小 (Transmit buffer 1 size) 第一个数据缓存的大小(以字节记)。如果它的值是 0, 那么 DMA 跳过这个缓存, 根据 TDES0[20]位配置使用缓存 2 或者下一个缓存。

**TDES2: 发送描述符字 2**

TDES2 包含描述符的第一个缓冲区的地址指针，或时间戳低 32 位数据。

域	简称	类型	功能
位 31: 0	TBAP1/T TSL	rw	发送缓存 1 地址指针/发送帧时间戳低 32 位 (Transmit buffer 1 address pointer / Transmit frame time stamp low) 这些位有 2 个功能: 1: 应用程序用它们指示 DMA 以太网帧数据在系统缓存中的位置。 2: 发送帧过程结束时, DMA 可以用它们存放发送帧的时间戳
	TBAP1		DMA 占有当前描述符时, 这些位表示缓存 1 的物理地址。
	TTSL		DMA 释放描述符给 CPU 前, DMA 会把对应发送帧的时间戳低 32 位写入这些位。只有在该帧的时间戳使能(TDES0 位 25 的 TTSE 位为'1')和帧的 LS 位为'1'时(表明当前帧传输结束), DMA 才会把时间戳写入这些位。

**TDES3: 发送描述符字 3**

TDES3 包含描述符的第二个缓冲区的地址指针或下一个描述符，或时间戳数据。

域	简称	类型	功能
位 31: 0	TBAP2/T TSH	rw	发送缓存 2 地址指针(下个描述符地址)/发送帧时间戳高位 (Transmit buffer 2 address pointer (Next descriptor address) / Transmit frame time stamp high) 这些位有 2 个功能: 1: 程序用它们指示 DMA 以太网数据在系统缓存中的位置。 2: 帧发送结束后, DMA 可以用它们存放帧的时间戳高 32 位
	TBAP2		DMA 占有当前描述符时, 如果描述符队列为环式时, 这些位表示缓存 2 的物理地址。如果描述符为链式的, 这些位表示下一个描述符的物理地址。
	TTSH		发送帧时间戳高 32 位 (Transmit frame time stamp high) DMA 会把对应帧的时间戳高 32 位写入这些位。只有在帧的时间戳使能(TDES0 位 25 的 TTSE 位为'1')和 LS 位(帧的最后一部分)为'1'时(表明当前帧发送结束), DMA 才会把时间戳写入这些位

**RXDMA 设置**

1. 应用程序建立接收描述符(RDES0~RDES3), 并置OWN位为'1' 将描述符释放给DMA。
2. 配置SSR位(EMAC\_DMAOPM[1])为'1', DMA进入接收运行状态, DMA尝试获取接收描述符, 如果取到的描述符不可用(被CPU占有), 则DMA进入暂停状态, 并跳到步骤9。
3. DMA从取得的接收描述符中解析出接收缓存地址。
4. DMA将帧数据从RXFIFO写入到接收缓存。
5. 如果缓存被填满或者帧传输结束, 接收控制器会取描述符队列中的下一个接收描述符。
6. 如果当前帧传输结束, DMA跳到步骤7。如果当前帧传输没有结束(未接收到帧尾EOF)而下一个接收描述符的OWN位为0, 这时, 如果使能了帧清空功能, 则DMA置位RDES0的描述符错误位并关闭当前描述符(OWN位清'0')同时置位RDES1的LS位, 然后跳到步骤8(注意如果没有打开帧清空功能, 则不会置位RDES1的LS位)。如果当前帧传输没有结束, 而下一个接收描述符的OWN位为1, 那么DMA关闭当前描述符, 标记其为中间描述符, 然后退回步骤4。
7. 如果使能了IEEE1588时间戳功能, DMA在写回状态信息给RDES0的同时也会把时间戳写入当前描述符的RDES2和RDES3。然后清除当前描述符RDES0 OWN位同时把LS位置'1'。
8. 接收控制器检查新获取的接收描述符, 如果描述符被DMA占有则返回到步骤4。如果描述符被CPU占有, RXDMA会进入挂起状态, 并置位接收缓存不可用位, 如果使能了接收帧清空功能, 控制器会清空接收帧。
9. DMA收到接收查询命令或者接收FIFO里有下一个接收到帧的帧头时, DMA退出挂起状态, 并取下一个描述符, 然后跳到步骤2。

**获取接收描述符**

RXDMA 总是会尝试获取另一个接收描述符。只要满足下列条件中的任意一个, RXDMA 就会尝试获取接收描述符:

- 在 DMA 进入运行状态 (EMAC\_DMAOPM SSR 位被置为'1')
- 当前描述符指定的缓存不足以容纳整个帧或者剩余帧数据
- 控制器已经完成当前帧的接收, 但还没有关闭当前的接收描述符
- 接收到新的帧但是接收流程由于描述符被 CPU 占有而挂起
- 接到接收轮询命令

## 接收帧处理

MII/RMII 接口接收帧数据并把帧存入 RXFIFO，当达到设定的条件后 RXDMA 开始把数据转发给当前描述符对应的接收缓存。DMA 会把当前描述符 RDES0 的 FS 位置 1，表示缓存里是帧的第一部分。在 DMA 填满缓存，或者帧接收完毕后，DMA 把 OWN 位清 0 以释放描述符。如果当前描述符缓存足以容纳完整帧数据，则当前描述符 RDES0 LS 位和 FS 位都会被置 1。

DMA 取下一个描述符，把前一个描述符的 LS 位置为 '1'，写回接收状态信息到前一个描述符再关闭前一个描述符。然后，DMA 把接收中断位(EMAC\_DMASTS[6])置为 '1'。上面的步骤会一直重复直到 DMA 发现描述符被 CPU 占有，此时接收流程会把接收缓存不可用位(EMAC\_DMASTS[7])置为 '1'，并进入挂起状态。在进入挂起状态时会记录描述符列表地址指针当前值，并在退出挂起状态后作为描述符开始的地址。

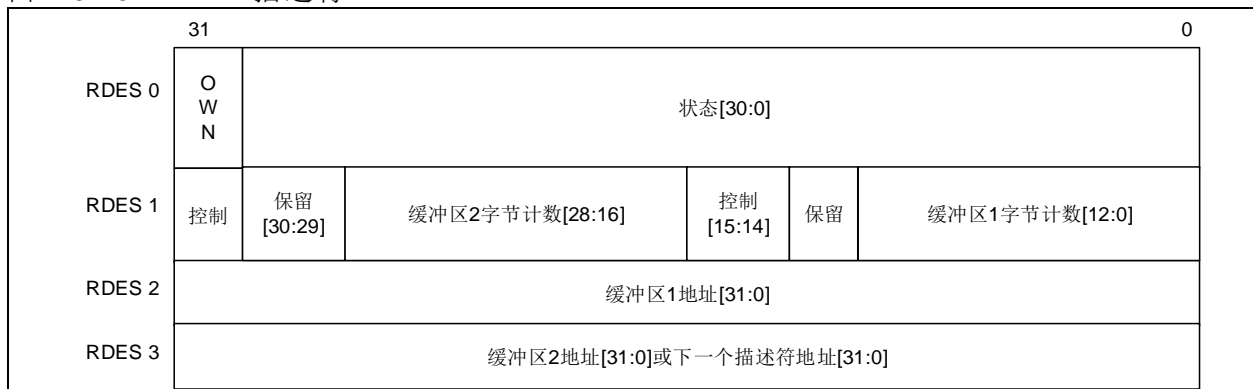
## 接收流程暂停

在 RXDMA 挂起的情况下，如果 RXFIFO 接收到新的帧，DMA 会重新从 AT32F407xx 内存里取当前描述符。如果取的描述符 OWN 位为 1，接收流程就会重新进入运行模式。如果 OWN 位为 0，那么 DMA 默认会丢弃在接收 FIFO 顶部的这个帧，并且丢失帧计数器加 1。如果 RXFIFO 里有一个以上的帧，则重复上述的步骤。把以太网 DMA 工作模式寄存器(EMAC\_DMAOPM)的位 24(DFRF 位)置为 '1' 可以阻止清空或者丢弃接收 FIFO 顶部的帧；这时，接收流程会把接收缓存不可用位置为 '1'，并回到挂起状态。

## RXDMA 描述符

如下图，描述符结构体包含 4 个 32 位字，RDES0、RDES1、RDES2 和 RDES3。

图 26-13 RXDMA 描述符



## RDES0: 接收描述符字 0

RDES0 包含了接收帧的状态信息，帧长和描述符的占有信息。

域	简称	类型	功能
位 31	OWN	rw	占有位 (Own bit) 0: 表示 CPU 占有描述符; 1: 表示 DMA 占有描述符。 DMA 在传输完整个帧或者填满描述符对应的缓存以后把该位清 '0'，释放描述符给 CPU。
位 30	AFM	rw	目的地址过滤器未通过 (Destination address filter fail) 置 '1' 时，该位表示帧没有通过 MAC 控制器的目的地址(DA)过滤器。
位 29: 16	FL	rw	帧长 (Frame length) 这些位表示 DMA 发送到系统缓存的接收帧的字节数目。只有在 LS 位(RDES0[8])为 '1' 且描述符错误位(RDES0[14])为 '0' 时，这些位才有效。如果 LS 位为 0，则这些位的值表示帧当前传送到系统缓存里的的累计字节数目。帧长是否包括 CRC 部分取决于以太网 MAC 配置寄存器(EMAC_MACCTRL)中 bit7 和 bit25 的配置。
位 15	ES	rw	错误汇总 (Error summary) 该位为下列位的逻辑“或”: RDES0[1]: CRC 错误; RDES0[3]: 接收错误; RDES0[4]: 看门狗超时; RDES0[6]: 迟到冲突; RDES0[7]: 巨大帧或者 IP 校验和错误; RDES0[11]: 溢出错误; RDES0[14]: 描述符错误。 该位在 LS 位(RDES0[8])为 '1' 时才有效。

位 14	DE	rw	描述符错误 (Descriptor error) 置'1'时, 该位表示由于描述符的缓存装不下当前接收帧, 而 DMA 又不占有下一个描述符, 导致当前帧被切断。该位只有在 LS 位(RDES0[8])为'1'时才有效。
位 13	SAF	rw	源地址过滤器未通过 (Source address filter fail) 置'1'时, 该位表示接收帧没有通过 MAC 控制器的源地址(SA)过滤器。
位 12	LE	rw	长度错误 (Length error) 置'1'时, 表示接收到帧的实际长度与以太网帧头长度/类型域的值不相符。该位只有在 RDES0[5]位为'0'时才有效。该位在有 CRC 错误时无效。
位 11	OE	rw	溢出错误 (Overflow error) 置'1'时, 表示由于接收 FIFO 溢出, 接收到的帧被破坏。
位 10	VLAN	rw	VLAN 标签 (VLAN tag) 置'1'时, 表示当前描述符指向的帧被 MAC 标记为 VLAN 帧。
位 9	FS	rw	第一个描述符 (First descriptor) 置'1'时, 表示这个描述符包含帧的第一个缓冲区, 如果缓冲区的长度为 0, 则第 2 个缓冲区包含帧的开头; 如果第 2 个缓冲区的长度也是 0, 则下一个描述符指示的缓冲区包含帧的开头。
位 8	LS	rw	最后一个描述符 (Last descriptor) 置'1'时, 表示这个描述符指向的缓存区是帧的最后一个缓冲区。
位 7	IPHCE	rw	IPv 报头校验和错误 (IPv header checksum error) 置'1'时, 表示 IPv4 或者 IPv6 的报头错误。错误可能是由于以太网类型域和 IP 版本域值不匹配, IPv4 报头校验和不对或者以太网帧的 IP 报头字节数不足造成的。
位 6	LC	rw	迟到冲突 (Late collision) 置'1'时, 该位表示在半双工模式下接收帧的时候发生了迟到冲突。
位 5	FT	rw	帧类型 (Frame type) 置 '1' 时表示接收到的帧是以太网帧 (LT 域大于等于 1536)。清 '0' 时表示接收到的帧是 IEEE802.3 帧。在接收到帧是小于 14 字节的过短帧时, 该位无效。
位 4	RWT	rw	接收看门狗超时 (Receive watchdog timeout) 置'1'时表示在接收当前帧的时候, 看门狗超时。看门狗超时后, 当前接收帧被截断。
位 3	RE	rw	接收错误 (Receive error) 置'1'时表示在接收帧过程中 RX_DV 有效时, RX_ER 信号有效。
位 2	DE	rw	Dribble 位错误 (Dribble bit error) 置'1'时表示接收到的帧长度不是字节的整数倍(奇数个 4 位数据)。该位只在 MII 模式下有效。
位 1	CE	rw	CRC 错误 (CRC error) 置'1'时表示接收到的帧发生 CRC 错误。该位只在 LS 位(RDES0[8])为'1'时才有效。
位 0	PCE	rw	数据校验和错误 (Payload checksum error) 置'1'时, 表示 MAC 控制器计算的 TCP、UDP 或 ICMP 校验与接收到帧的 TCP、UDP 或 ICMP 的校验和域值不相符。在接收到以太网帧的数据长度和 IPv4 或 IPv6 数据包长度域的值不符时, 该位也会置'1'。

下表显示了位 5、7、0 取值的含义

表 26-7 接收描述符 0

位 5: 帧类型	位 7: 校验和错误	位 0: 数据校验和错误	帧状态
0	0	0	IEEE802.3 类型帧(长度域值小于 0x0600)
1	0	0	IPv4/IPv6 类型帧, 未检测到校验和错误
1	0	1	IPv4/IPv6 类型帧, 检测到数据校验和错误(PCE 位)
1	1	0	IPv4/IPv6 类型帧, 检测到 IP 报头校验和错误(IPC CE 位)
1	1	1	IPv4/IPv6 类型帧, 检测到 IP 报头校验和错误和数据校验和错误
0	0	1	IPv4/IPv6 类型帧, 未检测到 IP 报头校验和错误; 由于不支持的数据格式, 未执行数据校验和检测
0	1	1	帧类型即不是 IPv4 也不是 IPv6(校验和模块不执行校验和检测)
0	1	0	保留



## RDES1: 接收描述符字 1

域	简称	类型	功能
位 31	DIC	rw	关闭接收完成中断 (Disable interrupt on completion) 置'1'时, 在帧接收完成的时候, 对于当前描述符指示的帧, 不会设置以太网 DMA 状态寄存器(EMAC_DMASTS)的 RECV 位(位 6)。这样也就不会发生 RECV 位触发的中断。该位只在 RDES0[8]为 1 时有效。
位 30: 29	保留	resd	请保持默认值。
位 28: 16	RBS2	rw	接收缓存 2 大小 (Receive buffer 2 size) 这些位的值给出接收缓存 2 的大小, 以字节为单位。如果 RDES1[14]位为'1', 这些位取值无意义。
位 15	RER	rw	接收描述符环形结构结尾 (Receive end of ring) 置'1'时, 该位表示当前描述符是描述符队列中的最后一个。DMA 会回到描述符队列基地址去取下一个描述符, 使描述符队列形成环形结构。
位 14	RCH	rw	第二地址链表 (Second address chained) 置'1'时, 该位表示描述符里的第二地址是下一个描述符的地址, 而不是第二个缓存的地址。该位为'1'时, RBS2(TDES1[28: 16])的值可以忽略。RDES0[15]的作用优先于本位 RDES0[14]。
位 13	保留	resd	请保持默认值。
位 12: 0	RBS1	rw	接收缓存 1 大小 (Receive buffer 1 size) 这些位的值表示接收缓存 1 的大小, 以字节为单位。如果这些位取值为 0, DMA 忽略接收缓存 1, 按照 RDES[14]的取值, 直接取接收缓存 2 或者下一个描述符。

## RDES2: 接收描述符字 2

RDES2 包含描述符第一个数据缓存的地址指针, 或者时间戳数据。

域	简称	类型	功能
位 31: 0	RBAP1/RTSL	rw	接收缓存 1 地址指针/接收帧时间戳低位 (Receive buffer 1 address pointer / Receive frame time stamp low) 这些位有 2 个功能: 应用程序用它们表示 DMA 数据在内存中的位置。等到数据接收完后, DMA 可以用它们存放帧的时间戳。
	RBAP1		DMA 占有描述符时, 这些位表示缓存 1 的物理地址。
	RTSL		DMA 释放描述符之前, DMA 会把帧的时间戳低 32 位写入这些位。只有在使能时间戳功能和 LS 位为'1'(表示缓存存放的是帧的最后分块)时, DMA 才会在这些位写入时间戳。

## RDES3: 接收描述符字 3

RDES3 包含描述符第二个数据缓存的地址指针或者指向下一个描述符, 或者时间戳数据。

域	简称	类型	功能
位 31: 0	RBAP2/RTSH	rw	接收缓存 2 地址指针(或者下一个描述符地址指针)/接收帧时间戳高位 (Receive buffer 2 address pointer (next descriptor address) / Receive frame time stamp high) 位 31: 0 这些位有 2 个功能: 程序用它们表示 DMA 数据在内存中的位置。等到数据接收完后, DMA 可以用它们存放帧的时间戳
	RBAP2		DMA 占有当前描述符时, 在描述符队列为环形结构的时候, 这些位表示缓存 2 的物理地址。如果第二地址链表位(RDES0[14])为'1', 并且下一个描述符存在, 则这些位指向下一个描述符的物理地址。
	RTSH		DMA 释放描述符之前, DMA 会把帧的时间戳高 32 位写入这些位。只有在时间戳使能和 LS 位为'1'(表示缓存存放的是帧的最后分块)时, DMA 才会在这些位写入时间戳。

## 26.2.4 EMAC掉电模式进入及唤醒

在使能了以太网 MAC PMT 控制和状态寄存器(EMAC\_MACPMTCTRLSTS)的 PD 位时, EMAC 会进入掉电模式, 在掉电模式下 EMAC 会丢弃所有的帧而不把它们转发给应用程序。

PMT 模块支持接收远程唤醒帧和 AMD Magic Packet 帧用以把 EMAC 从掉电模式下唤醒, 需要设置相应的使能位 ERWF 和 EMP (以太网 MAC PMT 控制和状态寄存器(EMAC\_MACPMTCTRLSTS)位 6 和位 5)。

## 远程唤醒帧过滤器寄存器

唤醒帧寄存器一共有 8 个，需要逐一配置帧过滤器寄存器。连续写 8 次唤醒帧过滤器寄存器，就可以把需要的值分别写入唤醒帧过滤器寄存器 1-8。对这些寄存器的读操作流程和写操作流程一致，需要连续读 8 次唤醒帧过滤器寄存器，才能读出唤醒帧过滤器寄存器 1-8 的值。

图 26-14 唤醒帧过滤寄存器

Wkuppktfilter_reg0	Filter 0 Byte Mask							
Wkuppktfilter_reg1	Filter 1 Byte Mask							
Wkuppktfilter_reg2	Filter 2 Byte Mask							
Wkuppktfilter_reg3	Filter 3 Byte Mask							
Wkuppktfilter_reg4	RES0	Filter 3 Cmd	RES0	Filter 2 Cmd	RES0	Filter 1 Cmd	RES0	Filter 0 Cmd
Wkuppktfilter_reg5	Filter 3 Offset		Filter 2 Offset		Filter 1 Offset		Filter 0 Offset	
Wkuppktfilter_reg6	Filter 1 CRC-16				Filter 0 CRC-16			
Wkuppktfilter_reg7	Filter 3 CRC-16				Filter 2 CRC-16			

### 过滤器 i 字节屏蔽

该寄存器定义了使用过滤器 i(i=0~3)的哪些字节检查判断帧是否为唤醒帧。第 31 位必须为'0'；位 j[30:0]是字节屏蔽位，如果位 j 为'1'，则 CRC 模块会处理输入帧的第[过滤器 i 偏移+ j]字节，否则忽略之过滤器 i 命令

4bit 命令。位 3 选择地址类型，如果该位为'1'，则只检测多播地址；如果该位为'0'，则只检测单播地址。位 2 和位 1 是保留位。位 0 是过滤器的使能位，如果为'0'，则该过滤器不工作

### 过滤器 i 偏移

这个 8bit 寄存器定义了过滤器 i 要检查的首字节在帧内的起始偏移。最小允许取值是 12，代表了帧的第 13 个字节(偏移值为 0 表示帧的第 1 个字节)

### 过滤器 i CRC-16

该寄存器包含了过滤器计算出的 CRC\_16 值，也包含了唤醒帧过滤器寄存器模块预先写好的字节屏蔽值

### 远程唤醒帧检测

设置以太网 MAC PMT 控制和状态寄存器(EMAC\_MACPMTCTRLSTS)的 RRWF 为'1'时可以使能远程唤醒帧检测。

PMT 支持 4 个可编程的过滤器，如果输入帧通过了过滤器的地址过滤，而且过滤器 CRC\_16 与被检查的输入帧匹配，则认为接收到唤醒帧。PMT 只会检查唤醒帧是否有长度错误、FCS 错误、Dribble 位错误、MII 错误、冲突，并确保唤醒帧不是过短帧。

当接收到远程唤醒帧后 EMAC 将从睡眠模式时恢复到正常工作状态。同时 EMAC\_MACPMTCTRLSTS 位 6 (RRWF) 将会被置 1，表示接收到了远程唤醒帧。如果使能了远程唤醒中断，那么 PMT 在接收到远程唤醒帧时会产生中断。

### Magic Packet 检测

通过配置 EMAC\_MACPMTCTRLSTS 位 1 (EMP) 可以使能 Magic Packet 帧检测。Magic Packet 帧包含特殊信息的数据包，专门用来唤醒 LAN 上的站点。

Magic Packet 帧的数据格式：6 字节全 1，紧接着重复 16 次的 MAC 地址。比如某设备的 MAC 地址是 0x11aabb22cc33,那么用于唤醒这个设备的 Magic Packet 帧如下：

```

目的地址 源地址 .....FFFF FFFF FFFF
11aa bb22 cc33 11aa bb22 cc33 11aa bb22 cc33 11aa bb22 cc33
11aa bb22 cc33 11aa bb22 cc33 11aa bb22 cc33 11aa bb22 cc33
11aa bb22 cc33 11aa bb22 cc33 11aa bb22 cc33 11aa bb22 cc33
11aa bb22 cc33 11aa bb22 cc33 11aa bb22 cc33 11aa bb22 cc33
... CRC
    
```

EMAC 睡眠模式下, PMT 模块会持续检测每一个发向本站点的帧, 检测其是否符合 Magic Packet 的格式。

接收到 Magic Packet 时会更新以太网 MAC PMT 控制和状态寄存器(EMAC\_MACPMTCTRLSTS)的位 5 (RMP)。如果使能了 Magic Packet 中断, 则 PMT 在接收到 Magic Packet 时会产生中断。

#### 系统在深度睡眠模式下注意事项

在 MCU 处于深度睡眠模式时, 使能外部中断线 19, 以太网的 PMT 模块仍能够检测帧。但是要注意以太网 MAC 配置寄存器(EMAC\_MACCTRL)的 RE 位需要保持为'1', 因为 EMAC 需要检测 Magic Packet 或者远程唤醒帧。

推荐的进入深度睡眠模式以及唤醒顺序如下:

1. 关闭TXDMA并等待之前所有的帧发送完毕。可以通过轮询以太网DMA状态寄存器(EMAC\_DMASTS)位0(TI)来检查发送是否完成。
2. 把以太网MAC配置寄存器(EMAC\_MACCTRL)的TE位和RE位清'0'来关闭MAC发送模块和MAC接收模块。
3. 轮询以太网DMA状态寄存器(EMAC\_DMASTS)位6(RI), 等待RXDMA把RXFIFO里的所有帧读出, 关闭RXDMA。
4. 配置并使能外部中断线19, 使其或者能产生事件或者能产生中断。
5. 设置以太网MAC PMT控制和状态寄存器(EMAC\_MACPMTCTRLSTS)的EMP/ERWF位为'1', 使能Magic Packet/远程唤醒帧检测。
6. 设置以太网MAC PMT控制和状态寄存器(EMAC\_MACPMTCTRLSTS)的PD位为'1', 使能掉电模式。
7. 设置以太网MAC配置寄存器(EMAC\_MACCTRL)的RE位为'1', 打开EMAC接收器。
8. MCU进入深度睡眠模式。
9. 接收到Magic Packet/远程唤醒帧后, 以太网模块退出掉电模式。
10. 读取以太网MAC PMT控制和状态寄存器(EMAC\_MACPMTCTRLSTS)来清除RRWF/RMP, 打开EMAC发送状态机, 以及发送和RXDMA。
11. 配置MCU系统时钟。

## 26.2.5 IEEE1588定义的精确时间协议

PTP 协议主要用来同步拥有不同精度, 分辨率和稳定度的系统, 不局限于以太网。关于详细的 PTP 协议请参考 IEEE 1588 相关文档。

EMAC 的 PTP 模块主要是记录 PTP 包从以太网端口发出和收到的准确时间, 并将其返回给应用程序。

#### 基准时钟源

IEEE1588 协议规定, 系统需要 64 位格式的基准时间来获得当前时间记录, 其中高 32 位提供以秒为单位的时间信息, 低 32 位提供以纳秒为单位的时间信息。

PTP 基准时钟主要用来在内部生成系统时间和记录时间戳。这个基准时钟的频率必须大于或等于时间戳计数器的分辨率。主节点和从节点之间的时间同步精度在 100ns 左右。

时间同步的精度主要取决于 PTP 基准时钟输入的频率, 晶体振荡器的频漂以及同步流程的执行频度。

#### 带 PTP 功能的帧的发送与接收

使能时间戳功能(EMAC\_PTPTSCTRL 位 0 配置为 1)以及当前帧的发送时间戳功能也使能时(发送描述符 TDES0 位 25 配置为 1), 当前帧的起始定界符从 MII 接口输出的时候, EMAC 会记录当前帧的发送时间戳。时间戳高 32 位和低 32 位分别存放在发送描述符 TDES3 和 TDES2 里面, 这样时间戳和发送状态信息会一同被返回给应用程序。

使能时间戳功能(EMAC\_PTPTSCTRL 位 0 配置为 1)以及接收帧的时间戳使能后(EMAC\_PTPTSCTRL 位 8 配置为 1), EMAC 会在 MII 端记录下所有收到帧的时间戳。时间戳高 32 位和低 32 位分别存放在接收描述符 RDES3 和 RDES2 里面, 这样时间戳和接收状态信息会一同被返回给应用程序。

#### 系统时间校准方法

PTP 输入基准时钟是系统时钟 SYSCLK, 通过 SYSCLK 来更新发送和接收的以太网帧的 64 位时间戳。系统时间的校准有两种方式: 粗略校准和精密校准。

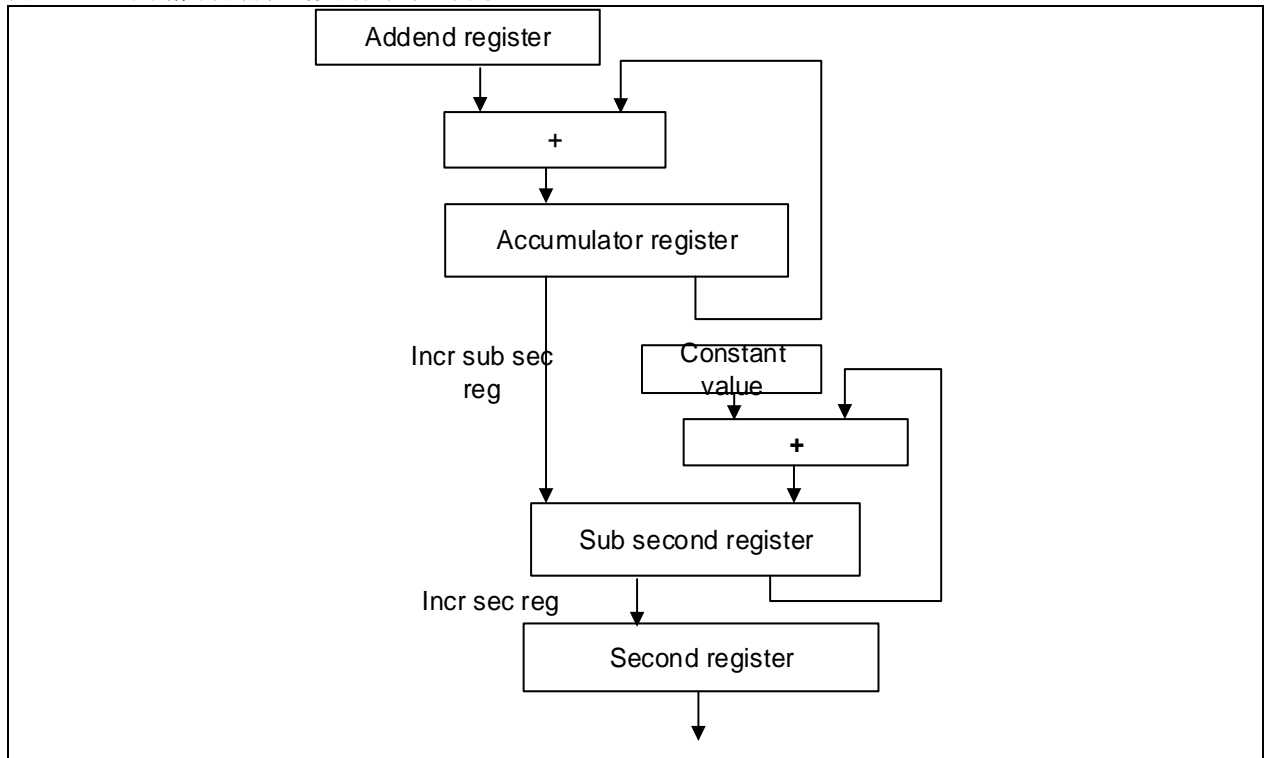
粗略校准: 把初始值/偏移值写入时间戳更新寄存器(以太网 PTP 时间戳高更新寄存器(EMAC\_PTPTSHUD)和以太网 PTP 时间戳低更新寄存器(EMAC\_PTPTSLUD))。如果 EMAC\_PTPTSCTRL 位 2(TI)被设置为 1, 则进行初始化流程, 时间戳更新寄存器的值会被写入系统时钟计数器。如果 EMAC\_PTPTSCTRL 位 3(TU)被设置为 1, 则进行校准流程, 时间戳更新寄存器



值会被作为偏移值，系统时间会加上或者从中减去这个偏移值。

精密校准：在一段时间里纠正从时钟(基准时钟)相对主时钟(如 IEEE1588 定义)的频率偏移。采用这种方法，如下图所示，把加数寄存器的值逐渐加入累加器，累加器的算术进位产生的脉冲令系统时间计数器值增加，加数寄存器的值取决于系统时钟频率。累加器和加数寄存器都是 32 位寄存器。

图 26-15用精调的方式更新系统时间



假设系统时钟更新电路的精度需要达到 20ns，则亚秒寄存器的更新频率需要是 50MHZ。因此，假设系统时钟频率是 70MHZ，计算频率比得  $70/50=1.4$ 。因此，写入加数寄存器的值是  $2^{32}/1.4$ ，等于 0XB6DB6DB6。

加数寄存器中的值要随着系统时钟频率的漂移做相应的改动。亚秒寄存器更新的频率是 50MHZ，每次更新增加的值也即是图中的常量数值为  $2^{31}/(50 \times 10^6)=43$ 。

软件需要利用 Sync 消息计算出频率的漂移，并相应地更新累加寄存器的值。开始时，设从时钟加数寄存器值为 FreqCompensationValue0：

$\text{FreqCompensationValue0} = 2^{32}/\text{频率比}$  假设对于连续的 Sync 消息，主从节点之间的时延 MasterToSlaveDelay 是恒定的，那么使用下文

描述的算法，在若干个 Sync 周期后，就能锁定频率。从时钟就可以确定精确的 MasterToSlaveDelay 值，并用新的值把从时钟与主时钟同步。算法如下：

在主时钟为 MasterSyncTime<sub>n</sub> 时，主节点向从节点发 Sync 消息。从节点在它的时钟为 SlaveClockTime<sub>n</sub> 时，收到 Sync 消息，并计算出 MasterClockTime<sub>n</sub> 为

$$\text{MasterClockTime}_n = \text{MasterSyncTime}_n + \text{MasterToSlaveDelay}_n$$

当前 Sync 周期的主时钟计数数目 MasterClockCount<sub>n</sub> 为

$$\text{MasterClockCount}_n = \text{MasterClockTime}_n - \text{MasterClockTime}_{n-1}, \text{ 假设 MasterToSlaveDelay 的数值在 Sync 周期 } n \text{ 与 Sync 周期 } n-1 \text{ 是相同的}$$

当前 Sync 周期的从时钟计数数目 SlaveClockCount<sub>n</sub> 为

$$\text{SlaveClockCount}_n = \text{SlaveClockTime}_n - \text{SlaveClockTime}_{n-1}$$

当前 Sync 周期，主时钟计数数目和从时钟计数数目的差别，ClockDiffCount<sub>n</sub> 为

$$\text{ClockDiffCount}_n = \text{MasterClockCount}_n - \text{SlaveClockCount}_n$$

从时钟的频率比系数，FreqScaleFactor<sub>n</sub> 为

$$\text{FreqScaleFactor}_n = (\text{MasterClockCount}_n + \text{ClockDiffCount}_n) / \text{SlaveClockCount}_n$$

加数寄存器的频率补偿值，FreqCompensationValue<sub>n</sub> 为

$$\text{FreqCompensationValue}_n = \text{FreqScaleFactor}_n \times \text{FreqCompensationValue}_{n-1}$$

该算法具有自校准功能，理论上可以在一个同步周期内锁定频率，不过实际上可能需要多个同步周期才

能同步从设备。

#### 系统时间初始化流程

1. 设置以太网MAC中断屏蔽寄存器(EMAC\_MAIMR)的位9为'1'，屏蔽时间戳触发中断。
2. 设置EMAC\_PTPTSCTRL位0为'1'，使能时间戳。
3. 根据系统时间更新精度要求编程亚秒递增寄存器。
4. 如果使用精调校准方式，则设置以太网PTP时间戳加数寄存器(EMAC\_PTPTSAD)，并设置以太网PTP时间戳控制寄存器(EMAC\_PTPTSCTRL)的位5为'1' (更新加数寄存器)，并轮询以太网PTP时间戳控制寄存器(EMAC\_PTPTSCTRL)直到位5变为'0'。如果使用粗略校准方式，请忽略此步骤并直接跳到第6步。
5. 如果使用精调校准方式，设置以太网PTP时间戳控制寄存器(EMAC\_PTPTSCTRL)的位1为'1'。
6. 把要设置的系统时间值写入以太网PTP时间戳高更新寄存器(EMAC\_PTPTSHUD)和以太网PTP时间戳低更新寄存器(EMAC\_PTPTSLUD)。
7. 设置以太网PTP时间戳控制寄存器(EMAC\_PTPTSCTRL)的位2为'1'，开始初始化时间戳并轮询此位，此位为0时，初始化完成。
8. 初始化完成，时间戳计数器开始工作。
9. 使能MAC的接收端和发送端，即可以正确地记录时间戳。

#### 用粗调方式更新系统时间的编程步骤

1. 在以太网PTP时间戳高更新寄存器(EMAC\_PTPTSHUD)和以太网PTP时间戳低更新寄存器(EMAC\_PTPTSLUD)里写入偏移值(正值或者负值)。
2. 设置以太网PTP时间戳控制寄存器(EMAC\_PTPTSCTRL)的位3(TU)为'1'。
3. 在TU位清'0'时，系统时间就会加上或者从中减去时间戳更新寄存器的值

#### 用精调方式更新系统时间的编程步骤

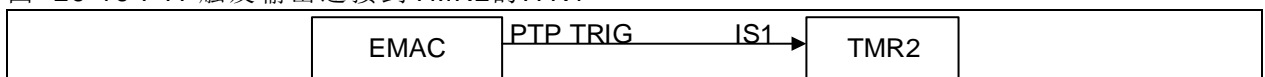
1. 采用前述“[错误!未找到引用源。](#)”介绍的算法，计算出加数寄存器的值。
2. 更新加数寄存器。
3. 把要求的目标时间写入目标时间高和目标时间低寄存器，并设置以太网MAC中断屏蔽寄存器(EMAC\_MAIMR)的位9为'0'来允许时间戳中断。
4. 设置以太网PTP时间戳控制寄存器(EMAC\_PTPTSCTRL)的位4(TITE)为'1'。
5. 在这个事件产生中断时，读以太网MAC中断屏蔽寄存器(EMAC\_MAIMR)清除相应的中断标志位。
6. 重新用旧值编写以太网PTP时间戳加数寄存器(EMAC\_PTPTSAD)并设置以太网PTP时间戳控制寄存器(EMAC\_PTPTSCTRL)位5为'1'来更新加数寄存器。

#### PTP 触发与 TMR2 的内部连接

EMAC 可以在系统时间大于目标值的时候提供触发中断。使用中断会引入中断时延，为了得到精确的中断延时时间，在系统时间大于目标值的时候，PTP 输出高信号给 TMR2。由于定时器的时钟与 PTP 基准时钟是同步的，因此可以精确的计算中断延时。

设置 IO 复用重映射寄存器(IOMUX\_REMAP)的位 29 为'1'可以使能 PTP 输出和 TIM2 IS1 的连接。

图 26-16 PTP触发输出连接到TMR2的ITR1



#### PTP 秒脉冲输出信号

PTP 脉冲输出详见以太网 PTP PPS 控制寄存器 (EMAC\_PTPPPSCR)描述，下面只介绍 emac\_pps\_sel 位(额外寄存器 3 (CRM\_MISC3) 位 15) 为 0 时的情况。

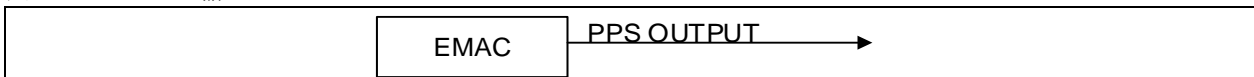
PPS 输出的默认频率为 1 Hz。可使用 PPSFREQ[3: 0] (位于 EMAC\_PTPPPSCR 中) 将 PPS 输出的频率设置为 2PPSFREQ Hz。

如果设置为 1 Hz，则使用二进制翻转(TSSSR=0，以太网 PTP 时间戳控制寄存器(EMAC\_PTPTSCTRL) 中的位 9) 时，PPS 脉冲宽度为 125 ms；使用数字翻转 (TSSSR=1) 时，PPS 脉冲宽度为 100 ms。

如果设置为 2 Hz 或更高频率，则使用二进制翻转时，PPS 输出的占空比为 50%。

设置 IO 复用重映射寄存器 (IOMUX\_REMAP) 的位 30 为'1'，可以使能 PPS 输出功能。

图 26-17 PPS输出



## 26.2.6 EMAC中断

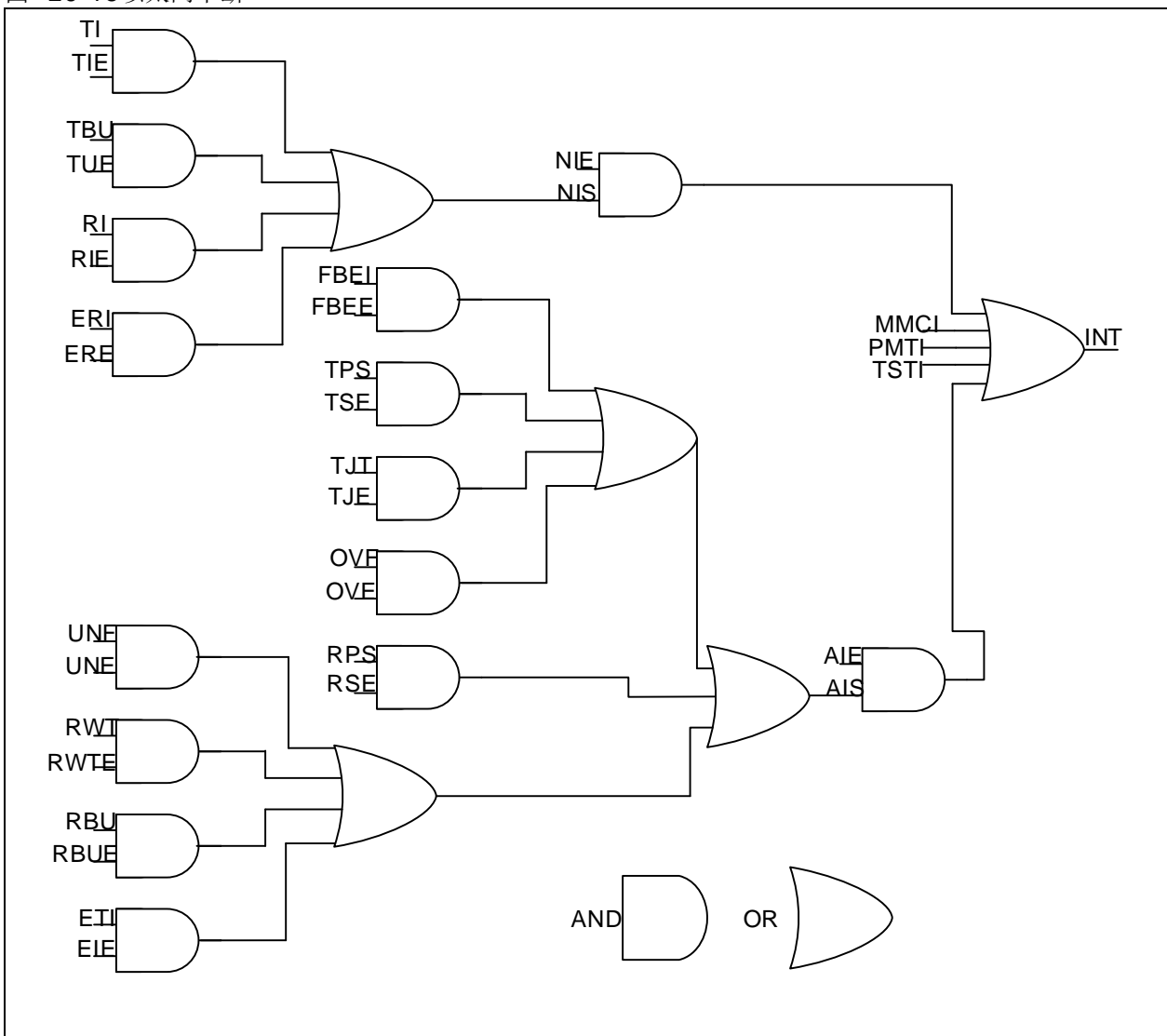
EMAC 有 2 个中断向量，一个用于以太网正常操作，另一个用于映射到 EXINT 线 19 的以太网唤醒事件 (检测远程唤醒帧或者 Magic Packet)。

第一个中断向量用于由 MAC 和 DMA 产生的中断。

第二个中断向量用于 PMT 模块在唤醒事件时产生的中断。唤醒事件映射到 EXINT 线 19 上，它可以使 AT32F407xx 微控制器退出低功耗模式，并产生中断。

在以太网唤醒事件映射到 EXINT 线 19 时，一旦发生唤醒事件，如果使能了 EMAC 的 PMT 中断，又使能了 EXINT 线 19 检测到上升沿时产生中断，那么 2 个中断会同时发生。

图 26-18 以太网中断



## 26.3 EMAC寄存器

下表给出了以太网寄存器映像及其复位值。

可以通过字节(8位)、半字(16位)和字(32位)的形式对本外设的寄存器进行访问。

表 26-8 以太网寄存器映像及其复位值

寄存器简称	基址偏移量	复位值
EMAC_MACCTRL	0x00	0x0000 8000
EMAC_MACFRMF	0x04	0x0000 0000
EMAC_MACHTH	0x08	0x0000 0000
EMAC_MACHTL	0x0C	0x0000 0000
EMAC_MACMIIADDR	0x10	0x0000 0000
EMAC_MACMIIDT	0x14	0x0000 0000
EMAC_MACFCTRL	0x18	0x0000 0000
EMAC_MACVLT	0x1C	0x0000 0000
EMAC_MACRWFF	0x28	0x0000 0000
EMAC_MACPMTCTRLSTS	0x2C	0x0000 0000
EMAC_MACISTS	0x38	0x0000 0000
EMAC_MAIMR	0x3C	0x0000 0000
EMAC_MACA0H	0x40	0x0010 FFFF
EMAC_MACA0L	0x44	0xFFFF FFFF
EMAC_MACA1H	0x48	0x0000 FFFF
EMAC_MACA1L	0x4C	0xFFFF FFFF
EMAC_MACA2H	0x50	0x0000 FFFF
EMAC_MACA2L	0x54	0xFFFF FFFF
EMAC_MACA3H	0x58	0x0000 FFFF
EMAC_MACA3L	0x5C	0xFFFF FFFF
EMAC_MMCTRL	0x100	0x0000 0000
EMAC_MMCRIM	0x104	0x0000 0000
EMAC_MMCTI	0x108	0x0000 0000
EMAC_MMCRIM	0x10C	0x0000 0000
EMAC_MMCTIM	0x110	0x0000 0000
EMAC_MMCTFSCC	0x14C	0x0000 0000
EMAC_MMCTFMSCC	0x150	0x0000 0000
EMAC_MMCTFCNT	0x168	0x0000 0000
EMAC_MMCRFCECNT	0x194	0x0000 0000
EMAC_MMCRFAECNT	0x198	0x0000 0000
EMAC_MMCRGUFCNT	0x1C4	0x0000 0000
EMAC_PTPTSCCTRL	0x700	0x0000 2000
EMAC_PTPSSINC	0x704	0x0000 0000
EMAC_PTPTSH	0x708	0x0000 0000
EMAC_PTPTSL	0x70C	0x0000 0000
EMAC_PTPTSH	0x708	0x0000 0000

EMAC_PTPTSL	0x70C	0x0000 0000
EMAC_PTPTSHUD	0x710	0x0000 0000
EMAC_PTPTSLUD	0x714	0x0000 0000
EMAC_PTPTSAD	0x718	0x0000 0000
EMAC_PTPTTH	0x71C	0x0000 0000
EMAC_PTP TTL	0x720	0x0000 0000
EMAC_PTPTSSR	0x728	0x0000 0000
EMAC_PT PPPSCR	0x72c	0x0000 0000
EMAC_DMABM	0x1000	0x0002 0101
EMAC_DMATPD	0x1004	0x0000 0000
EMAC_DMARPD	0x1008	0x0000 0000
EMAC_DMARDLADDR	0x100C	0x0000 0000
EMAC_DMATDLADDR	0x1010	0x0000 0000
EMAC_DMASTS	0x1014	0x0000 0000
EMAC_DMAOPM	0x1018	0x0000 0000
EMAC_DMAIE	0x101C	0x0000 0000
EMAC_DMAMFBOCNT	0x1020	0x0000 0000
EMAC_DMACTD	0x1048	0x0000 0000
EMAC_DMACRD	0x104C	0x0000 0000
EMAC_DMACTBADDR	0x1050	0x0000 0000
EMAC_DMACRBADDR	0x1054	0x0000 0000

## 26.3.1 以太网MAC配置寄存器(EMAC\_MACCTRL)

以太网 MAC 配置寄存器(EMAC\_MACCTRL)定义了接收和发送的工作模式。对此寄存器连续的写之间要插入延时，延时值大于 4us。

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	WD	0x0	rw	关闭看门狗 (Watchdog Disable) 关闭看门狗 该位被置起后，MAC 关闭接收器上的看门狗定时器，能够接收多达 16,384 字节的帧。 该位被复位后，MAC 不允许接收超过 2048 字节的帧。
位 22	JD	0x0	rw	关闭超长帧 (Jabber Disable) 该位被置起后，MAC 关闭发送器上的 Jabber (超长帧) 定时器，能够传输多达 16384 字节的帧。 该位被复位后，如果应用程序在传输过程中发送的数据超过 2048 字节时 MAC 会切断发送器。
位 21: 20	保留	0x0	resd	请保持默认值。
位 19: 17	IFG	0x0	rw	帧间间隙 (InterFrame Gap) 这些位是用来控制发送过程中帧与帧之间的最小间隙。 000: 96 bit times 96 位时间 001: 88 bit times 88 位时间 010: 80 bit times 80 位时间 ... 111: 40 bit times 40 位时间 在半双工模式下，IFG 最小值只能设置为 64 位时间(IFG = 100)，不允许更小的值。
位 16	DCS	0x0	rw	关闭载波监听 (Disable Carrier Sense) 该位被置起后，在半双工模式下，MAC 发送器在发送帧

				过程中会忽略 MII CRS 信号，发送过程中不会因为载波丢失或无载波而报错。 该位复位后，因为有载波监听，MAC 发送器会报错，甚至会中止发送。 在全双工模式配置中，此位保留。
位 15	保留	0x1	resd	请保持默认值。
位 14	FES	0x0	rw	快速 EMAC (Fast EMAC Speed) 此位用于选择 MII, RMII 接口速度。 0: 10 Mbps 1: 100 Mbps
位 13	DRO	0x0	rw	关闭自接收 (Disable Receive Own) 该位被置起后，如果在半双工模式下将 phy_txen_o 置为有效，则 MAC 不接受帧。 该位被复位后，MAC 在发送过程中接受 PHY 给出的所有数据包。 该位不适用于 MAC 全双工模式。 当 MAC 被设置为“仅适用于全双工”操作模式时，该位是保留位（有默认值的 RO）
位 12	LM	0x0	rw	环回模式 (Loopback Mode) 该位被置起后，MAC 的 MII 运行在环回模式。环回模式需要 MII 接收时钟输入(即 clk_rx_i)以确保能正常工作，因为发送时钟内部没有环回模式。
位 11	DM	0x0	rw	双工模式 (Duplex Mode) 该位被置起后，MAC 运行于全双工模式，可以同时接收和发送。
位 10	IPC	0x0	rw	IPv4 校验和 (IPv4 Checksum) 该位被置起后，MAC 计算所接收到的所有以太网帧的 16 位补码和，并检验所收到的帧的 IPv4 报头校验和（假设是所收到的以太网帧的字节 25-26 或者 29-30 (VLANtagged)）是否正确，并在接收状态信息中给出状态。 MAC 还附加了所计算得出的 IP 报头数据包（即 IPv4header 之后的字节）的 16 位校验和，并将这个校验和添加到已传输给应用程序的以太网帧（当取消选择 Type 2 COE 时）。 当该位被复位后，此功能关闭。 如果选择了 Type 2 COE，该位被置起时，会使能 IPv4 报头检验和检验功能以及 IPv4 或 IPv6 TCP, UDP 或 ICMP 有效载荷检验和检验功能。被复位后，接收器中的 COE 函数被禁用，相应的 PCE 和 IP HCE 状态位始终为 0。 如果在内核配置过程中，IP 校验和机制未使能，该位是保留位（有默认值 RO）
位 9	DR	0x0	rw	关闭重试 (Disable Retry) 该位被置起后，MAC 只尝试发送 1 次。如果在 MII 接口上发生冲突，MAC 忽略当前正在进行的发送帧，并且在发送帧状态里反馈有过多冲突而引发帧中止。 被复位后，MAC 根据 BL 字段（位[6: 5]）的设定尝试重发。该位仅适用于半双工模式，在“仅适用于全双工模式”配置中，该位是保留位（有默认值 RO）。
位 8	保留	0x0	resd	请保持默认值。
位 7	ACS	0x0	rw	自动填充/CRC 剥离 (Automatic pad/CRC Stripping) 该位被置起后，只有在帧的长度小于 1536 字节时，MAC 才会剥离所接收到的帧的 Pad 和 FCS 域。当接收到的帧的长度等于或大于 1536 字节时，MAC 会保留其 Pad 或 FCS 域并将帧传送给应用程序。 被复位后，MAC 会将所有接收到的帧转发给主机，而不改变其内容。
位 6: 5	BL	0x0	rw	后退限制 (Back-off Limit) 后退限制定义了 MAC 在发生冲突后，重新发送前需要等待的随机时间隙整数（对于 10/100 Mbps 是 512 位时



				<p>间)。该位仅适用于半双工模式，在“仅适用于全双工模式”的配置中，该位是保留位（RO）</p> <p>00: <math>k = \min(n, 10)</math></p> <p>01: <math>k = \min(n, 8)</math></p> <p>10: <math>k = \min(n, 4)</math></p> <p>11: <math>k = \min(n, 1)</math></p> <p>其中，<math>n</math> = 重发需要等待的时间间隙数量。<math>r</math> 取的是 <math>0 \leq r &lt; 2k</math> 之间的随机整数值。</p>
位 4	DC	0x0	rw	<p><b>递延检验 (Deferral Check)</b></p> <p>该位被置起后，MAC 使能递延检验功能。在 10/100 位模式下，如果发送被延迟超过 24288 位时间时，MAC 会发出帧中止状态，并且在发送帧状态里面将过多延迟错误标志位置起。</p> <p>如果 MAC 在 10 或 100 Mbps 模式中使能了 Jumbo 帧模式，则递延的阈值就是 155680 位时间。发送器准备发送时就开始递延，但是当检测到 MII 上有有效的载波监听信号时，递延就不会开始。递延时间不会累计。例如，如果因为 CRS 信号先是有有效的，而后又变为无效而使发送器延迟了 10000 位时间，然后发送器才开始发送，又发生冲突，因为冲突，发送器需要后退，并在完成后退动作之后又重新递延。那么这种情况下，递延定时器被重置为 0，并重新启动递延。</p> <p>当该位被重置后，递延检验功能被关闭，MAC 会延迟直到 CRS 信号失效。该位仅适用于半双工模式，在“仅适用于全双工模式”配置中，该位是保留位（RO）。</p>
位 3	TE	0x0	rw	<p><b>使能发送器 (Transmitter Enable)</b></p> <p>该位被置起后，MAC 的发送状态机被使能。该位被复位后，MAC 在发送完当前帧后，关闭发送状态机，不会再发送任何帧。(如果要连续通过指令更改此位，则连续的两次操作之间要插入延时，延时值大于 4us)</p>
位 2	RE	0x0	rw	<p><b>使能接收器 (Receiver Enable)</b></p> <p>该位被置起后，MAC 的接收状态机被使能。该位被复位后，MAC 在接收完当前帧后，关闭接收状态机，不会再接收任何帧。(如果要连续通过指令更改此位，则连续的两次操作之间要插入延时，延时值大于 4us)</p>
位 1: 0	保留	0x0	resd	请保持默认值。

### 26.3.2 以太网MAC帧过滤器寄存器(EMAC\_MACFRMF)

以太网 MAC 帧过滤器寄存器(EMAC\_MACFRMF)包含接收帧的过滤器控制位。一些控制位前往 MAC 的地址检验模块进行第一层级的地址过滤。

第二层级的过滤是根据其它控制位（比如传递坏帧和传递控制帧）针对传入的帧执行。

域	简称	复位值	类型	功能
位 31	RA	0x0	rw	<p><b>接收全部 (Receive All)</b></p> <p>该位被置起后，MAC 将接收到的所有帧传递给应用程序，无论这些帧是否通过了地址过滤。源地址或目的地址的过滤结果(pass or fail)会在接收状态的相应位进行更新。该位被复位后，接收模块只会将那些通过了源地址/目的地址过滤器的帧传递给应用程序。</p>
位 30: 11	保留	0x00000	resd	请保持默认值。
位 10	HPF	0x0	rw	<p><b>Hash 或完美过滤器 (Hash or Perfect Filter)</b></p> <p>该位被置起后，如果帧与 HMC 或 HUC 位设定的完美过滤器或 hash 过滤器相符，地址过滤器就能通过此帧。该位被复位后，如果 HUC 或 HMC 位被置起，则只有当帧符合 hash 过滤器时，地址过滤器才能通过此帧。</p>
位 9	SAF	0x0	rw	<p><b>源地址过滤器 (Source Address Filter)</b></p> <p>该位被置起后，MAC 会将接收到的帧的源地址域与所使能的源地址寄存器的值进行比较。如果不符，MAC 会丢弃此帧。</p>



				该位被复位后，MAC 会将接收到的帧转给应用程序，并根据源地址比较结果更新接收状态中的源地址过滤器位（SAF）。
位 8	SAIF	0x0	rw	<p>源地址逆过滤（Source Address Inverse Filtering）</p> <p>该位被置起后，地址检验模块运行于逆过滤模式。源地址与源地址寄存器相符的帧会被标记为未通过源地址过滤器。</p> <p>该位被复位后，源地址与源地址寄存器不相符的帧被标记为未通过源地址过滤器。</p>
位 7: 6	PCF	0x0	rw	<p>通过控制帧（Pass Control Frames）</p> <p>这些位用于管理所有控制帧的转发（包括单波和多波 Pause 帧）。</p> <p>00: MAC 过滤所有控制帧使其无法转到应用程序。</p> <p>01: MAC 将除 Pause 帧以外的所有控制帧转发给应用程序，即便这些帧未能通过地址过滤器。</p> <p>10: MAC 将所有控制帧转给应用程序，即便这些帧未能通过地址过滤器。</p> <p>11: MAC 转发通过了地址过滤器的控制帧给应用程序。</p> <p>在处理 Pause 帧时应满足以下条件：</p> <p>条件 1: MAC 处于全双工模式下，通过设置寄存器 6（流控制寄存器）中的位 2（REF）来使能流控。</p> <p>条件 2: 当寄存器 6（流控制寄存器）中的位 3（UP）被置起时，接收的帧的目的地址与特定的多播地址或 MAC 地址 0 相匹配。</p> <p>条件 3: 接收的帧的类型域（Type）是 0x8808，OPCODE 域是 0x0001。</p>
位 5	DBF	0x0	rw	<p>关闭广播帧（Disable Broadcast Frames）</p> <p>该位被置起后，地址过滤模拟将阻止所有传入的广播帧。此外，还会覆盖所有其它的过滤器设置。</p> <p>当该位被复位后，地址过滤器模块能允许通过所有接收到的广播帧。</p>
位 4	PMC	0x0	rw	<p>通过多播帧（Pass MultiCast）</p> <p>该位被置起后，所有带多播目的地址（目的地址的第一位被置 1）的帧都能通过地址过滤器。</p> <p>该位被复位后，多播帧是否能够过滤取决于 HMC 位。</p>
位 3	DAIF	0x0	rw	<p>目的地址逆过滤（Destination Address Inverse Filtering）</p> <p>该位被置起后，地址检验模块运行在逆过滤模式用于比较单播帧和多播帧的目的地址。</p> <p>该位被复位后，过滤器将正常工作。</p>
位 2	HMC	0x0	rw	<p>Hash 多播帧（Hash MultiCast）</p> <p>该位被置起后，MAC 根据 hash 列表对接收到的多播帧进行目的地址过滤。</p> <p>该位被复位后，MAC 对多播帧进行目的地址完美过滤，即将目的地址域与目的地址寄存器的设定值进行比较。如果在内核配置过程中没有选择 Hash 过滤器，则此位是保留位（及 RO）</p>
位 1	HUC	0x0	rw	<p>Hash 单播帧（Hash UniCast）</p> <p>该位被置起后，MAC 根据 hash 列表对单播帧的目的地址进行过滤。</p> <p>该位被复位后，MAC 对单播帧的目的地址进行完美过滤，即将目的地址域与目的地址寄存器的设定值进行比较。</p>
位 0	PR	0x0	rw	<p>混杂模式（Promiscuous Mode）</p> <p>该位被置起后，无论是目的地址还是源地址，地址过滤器能通过所有传入的帧。当 PR 被置起时，接收状态的源地址或目的地址错误位总是为 0。</p>

### 26.3.3 以太网MAC Hash列表高寄存器(EMAC\_MACHTH)

64 位的 Hash 列表用于成组的地址过滤。进行 hash 过滤时,传入帧的目的地址的内容要通过 CRC 逻辑, CRC 寄存器的高 6 位用于检索 Hash 列表。CRC 的最高位决定使用哪个寄存器 (以太网 MAC Hash 列表高寄存器(EMAC\_MACHTH)或以太网 MAC Hash 列表低寄存器(EMAC\_MACHTL)), 剩余的 5 位决定使用寄存器的哪一位。 Hash 值 5b'00000 使用的是所选中寄存器的位 0, 而 Hash 值 5b'11111 使用的是所选中寄存器的位 31。

目的地址的 hash 值按照以下方法计算:

1. 计算目的地址的32位CRC值 (见IEEE 802.3, 参考3.2.8章节获得CRC32计算方法)
2. 对步骤1获取的值进行位反转
3. 从步骤2获取的值中取其高6位

例如, 如果传入帧的目的地址是 0x1F52419CB6AF (MII 接口收到的第一个字节是 0x1F), 那么所计算出的 6 位 Hash 值是 0x2C, 并检查以太网 MAC Hash 列表高寄存器(EMAC\_MACHTH)的位 12 是否过滤。如果传入帧的目的地址是 0xA00A98000045, 那么所计算的 6 位 Hash 值是 0x07, 并检查以太网 MAC Hash 列表低寄存器(EMAC\_MACHTL)的位 7 是否过滤。

域	简称	复位值	类型	功能
位 31	HTH	0x0000 0000	rw	该位包含了 Hash 列表的高 32 位。

### 26.3.4 以太网MAC Hash列表低寄存器(EMAC\_MACHTL)

以太网 MAC Hash 列表低寄存器(EMAC\_MACHTL)包含了 Hash 列表的低 32 位。在内核配置过程中, 如果 Hash 过滤器功能关闭或者选择了 128 位或 256 位 Hash 列表, 则寄存器 2 和寄存器 3 都是保留位。

域	简称	复位值	类型	功能
位 31	HTL	0x0000 0000	rw	Hash 列表低 (Hash Table Low) 此位包含了 Hash 列表的低 32 位。

### 26.3.5 以太网MAC MII地址寄存器(EMAC\_MACMIIADDR)

以太网 MAC MII 地址寄存器(EMAC\_MACMIIADDR)通过管理接口控制外部 PHY。

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: 11	PA	0x00	rw	PHY 地址 (PHY Address) 这些位表示的是这 32 个可能的 PHY 设备中的哪一些正在被访问。
位 10: 6	MII	0x00	rw	MII 寄存器 (MII Register) 这些位表示在 PHY 中选择所需的 MII 寄存器。
位 5: 2	CR	0x0	rw	时钟范围 (Clock Range) CSR 时钟范围选择是指根据用户所使用的 CSR 时钟频率来确定 MDC 时钟频率。 针对每个值 (当位 5=0 时) 均推荐了 CSR 时钟频率范围, 以确保 MDC 时钟频率大致位于 1.0 MHz~2.5 MHz 以内。 0000: CSR 时钟频率为 60 - 100 MHz, MDC 时钟频率为 CSR 时钟除以 42。 0001: CSR 时钟频率为 100 - 150 MHz, MDC 时钟频率为 CSR 时钟除以 62 0010: CSR 时钟频率为 20 - 35 MHz, MDC 时钟频率为 CSR 时钟除以 16. 0011: CSR 时钟频率为 35 - 60 MHz, MDC 时钟频率为 CSR 时钟除以 26. 0100: CSR 时钟频率为 150 - 250 MHz, MDC 时钟频率为 CSR 时钟除以 102. 0101: CSR 时钟频率为 250 - 300 MHz, MDC 时钟频率为 CSR 时钟除以 124. 0110, 0111: 保留。

位 1	MW	0x0	rw	<p><b>MII 写操作 (MII Write)</b> 当该位被置起时，表示将使用以太网 MAC MII 数据寄存器(EMAC_MACMIIDT)对 PHY 进行写操作。 当该位未被置起时，表示读操作，读的数据将加载到以太网 MAC MII 数据寄存器(EMAC_MACMIIDT)。</p>
位 0	MB	0x0	rw	<p><b>MII 繁忙 (MII Busy)</b> 在写入以太网 MAC MII 地址寄存器(EMAC_MACMIIADDR)和以太网 MAC MII 数据寄存器(EMAC_MACMIIDT)之前，该位应该先读逻辑 0。在访问 PHY 寄存器期间，软件将该位设置为 1'b1，以此表示正在进行读操作或写操作。 在 MAC 清除该位之前，以太网 MAC MII 数据寄存器(EMAC_MACMIIDT)无效。因此，在对 PHY 写操作时，必须保持 MII 数据于有效状态直到 MAC 清除该位。同样地，在对 PHY 读操作时，以太网 MAC MII 数据寄存器(EMAC_MACMIIDT)的值是无效的，直到该位被清除。 只有在完成前一项操作之后，才能进行随后的读操作或写操作。因为在完成读操作或者写操作后不会有从 PHY 到 MAC 的确认，所以即使 PHY 不存在，该位的功能也不会发生变化。</p>

### 26.3.6 以太网MAC MII数据寄存器(EMAC\_MACMIIDT)

以太网 MAC MII 数据寄存器(EMAC\_MACMIIDT)存放的是要写入 PHY 寄存器的数据，该 PHY 寄存器位于以太网 MAC MII 地址寄存器(EMAC\_MACMIIADDR)所指定的地址内。该寄存器也存放着从该 PHY 寄存器读出的值。

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	MD	0x0000	rw	<p><b>MII 数据 (MII Data)</b> 这些位包含了在读操作之后从 PHY 读出的 16 位数据，或者在进行写操作之前要写入 PHY 的 16 位数据。</p>

### 26.3.7 以太网MAC流控寄存器(EMAC\_MACFCTRL)

以太网 MAC 流控寄存器(EMAC\_MACFCTRL)通过 MAC 的流控模块管理着控制帧（Pause 命令）的生成和接收。对 Busy 位写 1 会触发流控模块生成 Pause 帧。按照 802.3x 规范选择控制帧的位域，控制帧的 Pause Time 域会使用该寄存器的 Pause Time 值。在控制帧被发送到电缆上之前，Busy 位保持置起状态。主机必须确保在写寄存器之前，Busy 位已经清除。

域	简称	复位值	类型	功能
位 31: 16	PT	0x0000	rw	<p><b>Pause 时间 (Pause Time)</b> 这些位域存放着要用在控制帧 Pause 时间域的值。如果 Pause 时间位被设置为与 MII 时钟域双重同步，那么至少要 4 个目的时钟域周期后，才允许对该寄存器连续写操作。</p>
位 15: 8	保留	0x00	resd	请保持默认值。
位 7	DZQP	0x0	rw	<p><b>关闭零值 Pause (Disable Zero-Quanta Pause)</b> 该位被置起后，在撤销 FIFO 层流控信号时，它会关闭零值 Pause 帧的自动生成功能。 当该位被复位后，可正常操作，零值 Pause 帧的自动生成功能被启用。</p>
位 6	保留	0x0	resd	请保持默认值。
位 5: 4	PLT	0x0	rw	<p><b>Pause 低阈值 (Pause Low Threshold)</b> 这些位设置了 Pause 定时器的阈值。 阈值应该始终小于位[31: 16]定义的 Pause 时间。例如，如果 PT = 100H (256 个时间间隔)，PLT = 01，那么如果在发送第一个 Pause 帧之后的 228 (256-28) 个时间间隔后会自动发送第二个 Pause 帧。 下表给出了不同值所对应的阈值：</p>

				<p>00: 阈值是 Pause 时间- 4 个时间间隙 (PT - 4 个时间间隙).</p> <p>01: 阈值是 Pause 时间- 28 个时间间隙 (PT - 28 个时间间隙).</p> <p>10: 阈值是 Pause 时间- 144 个时间间隙 (PT - 144 个时间间隙).</p> <p>11: 阈值是 Pause 时间- 256 个时间间隙 (PT - 256 个时间间隙).</p> <p>时间间隙是 MII 接口发送 512 位 (64 字节) 所需要的时间。</p>
位 3	DUP	0x0	rw	<p>单播 Pause 帧监测 (Detect Unicast Pause Frame)</p> <p>当 pause 帧具有 IEEE 802.3 规定的唯一多播地址时, 将对其进行处理。该位被置起后, MAC 也能检测到带有单播地址的 Pause 帧。MAC 地址 0 高寄存器和 MAC 地址 0 低寄存器指定了单播地址。</p> <p>该位被复位后, MAC 只能检测带有唯一多播地址的 Pause 帧。</p> <p>备注: 如果接收帧的多播地址与唯一多播地址不一致, MAC 就不会处理 Pause 帧。</p>
位 2	ERF	0x0	rw	<p>使能接收流控 (Enable Receive Flow control)</p> <p>该位被置起后, MAC 会解译所收到的 Pause 帧, 且发送器被关闭一段时间 (Pause)。该位被复位后, Pause 帧的解译功能被关闭。</p>
位 1	ETF	0x0	rw	<p>使能发送流控 (Enable Transmit Flow control)</p> <p>在全双工模式下, 该位被置起后, MAC 使能流控功能而发送 Pause 帧。该位被复位后, MAC 的流控功能被关闭, 而且 MAC 不再发送任何 Pause 帧。</p> <p>在半双工模式下, 该位被置起后, MAC 使能背压功能。该位被复位后, 背压功能被关闭。</p>
位 0	FCB/BPA	0x0	rw1c/rw	<p>流控繁忙/背压激活 (Flow Control Busy/Back Pressure Activate)</p> <p>在全双工模式下, 该位启动 Pause 帧; 在半双工模式下, 当 TFE 位被置起后, 背压功能被激活。</p> <p>在全双工模式下, 在写以太网 MAC 流控寄存器 (EMAC_MACFCTRL) 之前, 该位读出值为 1'b0。应用程序必须将该位置为 1'b1 才能启动 Pause 帧。在发送控制帧期间, 该位始终处于置起状态, 表示正在发送帧。在 Pause 帧发送完成后, MAC 将该位重置为 1'b0。在清除该位之前, 不允许写以太网 MAC 流控寄存器 (EMAC_MACFCTRL),</p> <p>在半双工模式下, 当该位被置起时 (以及 TFE 被置起), MAC 激活背压功能。在背压工作期间, 如果 MAC 接收到新的帧, 发送器就开始发送 JAM 模式, 发生冲突。当 MAC 被设置为全双工模式时, 背压 (BPA) 功能自动关闭。</p>

### 26.3.8 以太网MAC VLAN标签寄存器(EMAC\_MACVLT)

以太网 MAC VLAN 标签寄存器(EMAC\_MACVLT)包含了用来识别 VLAN 帧的 IEEE 802.1Q VLAN 标签。MAC 会将接收帧(长度/类型)的第 13 和第 14 字节与 16'h8100 进行比较, 再将随后的两个字节与 VLAN 标签进行比较。如果比较结果相符, MAC 会在接收帧的状态栏里置起 VLAN 位。VLAN 帧的合法长度从 1518 字节增加到 1522 字节。

如果以太网 MAC VLAN 标签寄存器(EMAC\_MACVLT)被设置成与(G)MII 时钟域双重同步, 那么至少在 4 个目的时钟域周期之后才允许连续写入该寄存器。

域	简称	复位值	类型	功能
位 31: 17	保留	0x0000	resd	请保持默认值。
位 16	ETV	0x0	ro	<p>使能 12 位 VLAN 标签比较 (Enable 12-bit VLAN tag comparison)</p> <p>该位被置起后, 使用 12 位而不是 16 位 VLAN 标识符进行比较和过滤。将 VLAN 标签的位[11: 0]与接收到的 VLAN 标签帧的相应域进行比较。同样地, 被使能后, 只使用接收帧的 12 位 VLAN 标签来进行 hash VLAN 过滤。该位被复位后, 使用接收的 VLAN 帧的第 15 和第 16 字节的 16 位进行比较和 VLAN hash 过滤。</p>
位 15: 0	VTI	0x0000	rw	<p>VLAN 标签标识符 (针对接收帧) (VLAN Tag Identifier (for receive frames))</p> <p>这些位包含了用于识别 VLAN 帧的 802.1Q VLAN 标签, 并与接收到的 VLAN 帧的第 15 和第 16 字节进行比较。下表对这些位进行了详细描述:</p> <p>位[15: 13]: 用户优先级</p> <p>位 12: 标准格式指示器 (CFI) or 丢弃适当性指示符 (DEI)</p> <p>位[11: 0]: VLAN 标签的 VLAN 指示符字段</p> <p>当 ETV 位被置起时, 只使用 VID(位[11: 0])来进行比较。</p> <p>如果 VL (如果是 ETV 被置起, 则为 VL[11: 0])全为 0, MAC 不会检验用于 VLAN 标签比较的第 15 和第 16 字节, 只要帧的类型域值是 0x8100 或 0x88a8, 则均被视为 VLAN 帧。</p>

### 26.3.9 以太网MAC远程唤醒帧过滤器寄存器(EMAC\_MACRWFF)

PMT CSR 设置请求唤醒事件并监测唤醒事件。

图 26-19 以太网MAC唤醒帧过滤器寄存器(EMAC\_MACRWFF)

Wkuppktfilter_reg0	Filter 0 Byte Mask							
Wkuppktfilter_reg1	Filter 1 Byte Mask							
Wkuppktfilter_reg2	Filter 2 Byte Mask							
Wkuppktfilter_reg3	Filter 3 Byte Mask							
Wkuppktfilter_reg4	RESD	Filter 3 Cmd	RESD	Filter 2 Cmd	RESD	Filter 1 Cmd	RESD	Filter 0 Cmd
Wkuppktfilter_reg5	Filter 3 Offset		Filter 2 Offset		Filter 1 Offset		Filter 0 Offset	
Wkuppktfilter_reg6	Filter 1 CRC-16				Filter 0 CRC-16			
Wkuppktfilter_reg7	Filter 3 CRC-16				Filter 2 CRC-16			



### 26.3.10 以太网MAC PMT控制和状态寄存(EMAC\_MACPMTCTRLSTS)

以太网 MAC PMT 控制和状态寄存器(EMAC\_MACPMTCTRLSTS)设置并监控唤醒事件。

域	简称	复位值	类型	功能
位 31	RWFFPR	0x0	rw1s	远程唤醒帧过滤器寄存器指针复位 (Remote Wakeup Frame Filter Register Pointer Reset) 被置起后, 该位会将远程唤醒帧过滤器寄存器指针重置为 3'b000。该位在 1 个时钟周期后自动被清除。
位 30: 10	保留	0x000000	resd	请保持默认值。
位 9	GUC	0x0	rw	全局单播 (Global UniCast) 该位被置起后, 所有通过了 MAC 地址过滤的单播包均被使能为远程唤醒帧。
位 8: 7	保留	0x0	resd	请保持默认值。
位 6	RRWF	0x0	rrc	接收到远程唤醒帧 (Received Remote Wakeup Frame) 被置起后, 该位表示的是因为收到远程唤醒帧而生成了电源管理事件。读该寄存器可以将该位清除。
位 5	RMP	0x0	rrc	接收到 Magic Packet (Received Magic Packet) 被置起后, 该位表示的是, 因为接收到 Magic packet 而生成了电源管理事件。读该寄存器可以将该位清除。
位 4: 3	保留	0x0	resd	请保持默认值。
位 2	ERWF	0x0	rw	使能远程唤醒帧 (Enable Remote Wakeup Frame) 被置起后, 表示在收到远程唤醒帧时会生成电源管理事件。
位 1	EMP	0x0	rw	使能 Magic Packet Enable Magic Packet) 被置起后, 表示在收到 magic packet 时会生成电源管理事件。
位 0	PD	0x0	rw1s	掉电 (Power Down) 被置起后, MAC 接收器在收到预期的 magic packet 或远程唤醒帧之前会丢弃所有接收到的帧。然后该位会自动清除并关闭掉电模式。在收到预期的 magic packet 或远程唤醒帧之前, 软件也会清除该位。该位被清除后, MAC 将接收到的帧转发给应用程序。只能当 Magic Packet 使能位, 全局单播位或远程唤醒帧使能位被置高时, 才能将该位置起。

### 26.3.11 以太网MAC中断状态寄存器(EMAC\_MACISTS)

以太网 MAC 中断状态寄存器(EMAC\_MACISTS)用于识别 MAC 中能产生中断的事件。

域	简称	复位值	类型	功能
位 15: 10	保留	0x00	resd	请保持默认值。
位 9	TIS	0x0	rrc	时间戳中断状态 (Timestamp Interrupt Status) 如果该位被置起, 表示系统时间值等于或超过目的时间寄存器的设定值。完成读取此位的动作后, 该位被清除。
位 8: 7	保留	0x0	resd	请保持默认值。
位 6	MTIS	0x0	ro	MMC 发送中断状态 (MMC Transmit Interrupt Status) 以太网 MMC 发送中断寄存器(EMAC_MMCTI)产生中断事件后, 该位被置起。清除发送中断寄存器的所有位即清除该位。
位 5	MRIS	0x0	ro	MMC 接收中断状态 (MMC Receive Interrupt Status) 当以太网 MMC 接收中断寄存器(EMAC_MMCTI)产生了中断时, 此位被置起。清除接收中断寄存器的所有位即清除此位。
位 4	MIS	0x0	ro	MMC 中断状态 (MMC Interrupt Status) 位[7: 5]中的任一位被置高时, 该位被置起。只有当这些位都被置低时, 该位才被清除。
位 3	PIS	0x0	ro	PMT 中断状态 (PMT Interrupt Status) 在掉电模式下 (详见以太网 MAC PMT 控制和状态寄存器(EMAC_MACPMTCTRLSTS)的位 5 和位 6) 接收到 magic packet 或远程唤醒帧时, 该位置起。

				如果因为读取以太网 MAC PMT 控制和状态寄存器 (EMAC_MACPMTCTRLSTS)而清除了位[6: 5], 则该位被清除。
位 2: 0	保留	0x0	resd	请保持默认值。

### 26.3.12 以太网MAC中断屏蔽寄存器(EMAC\_MAIMR)

以太网 MAC 中断屏蔽寄存器(EMAC\_MAIMR)可以用来屏蔽因以太网 MAC 中断状态寄存器(EMAC\_MACISTS)中对应事件而产生的中断信号。

域	简称	复位值	类型	功能
位 15: 10	保留	0x00	resd	请保持默认值。
位 9	TIM	0x0	rw	时间戳中断屏蔽 (Timestamp Interrupt Mask) 置起后, 该位会屏蔽以太网 MAC 中断状态寄存器 (EMAC_MACISTS)中时间戳中断状态位生成的中断信号。 只有使能了 IEEE1588 时间戳, 该位才有效。在其它模式中, 该位保留。
位 8: 4	保留	0x00	resd	请保持默认值。
位 3	PIM	0x0	rw	PMT 中断屏蔽 (PMT Interrupt Mask) 置起后, 该位会屏蔽以太网 MAC 中断状态寄存器 (EMAC_MACISTS)中 PMT 中断状态位生成的中断信号。
位 2: 0	保留	0x0	resd	请保持默认值。

### 26.3.13 以太网MAC地址0高寄存器(EMAC\_MACA0H)

以太网 MAC 地址 0 高寄存器(EMAC\_MACA0H)包含了设备的前 6 字节 MAC 地址的高 16 位。MII 接口收到的第一个 DA 字节对应于 MAC 地址低寄存器的 LS 字节(位[7: 0])。例如, 如果将 MII 收到的 (第一列的通道 0 中的 0x11) 0x112233445566 视为目的地址, 那么 MacAddress0 寄存器[47: 0]用来与 0x665544332211 作比较。

如果 MAC 地址寄存器被设置成与 MII 域双重同步, 那么只有向以太网 MAC 地址 0 低寄存器(EMAC\_MACA0L)写入位[31: 24] (在小端模式中)或位[7: 0] (在大端模式中), 才能触发同步功能。为了进行正确的同步更新, 必须至少在 4 个目的时钟域周期之后, 才允许对地址低寄存器进行连续写操作。

域	简称	复位值	类型	功能
位 31	AE	0x0	rrc	使能 (Adress) 该位始终为 1。
位 30: 16	保留	0x0010	resd	请保持默认值。
位 15: 0	MA0H	0xFFFF	rw	MAC 地址高 (MAC Address0 [47: 32]) 该位域包含了前 6 字节 MAC 地址的高 16 位。MAC 使用这个位域来过滤接收帧, 并将 MAC 地址插入到发送流控帧 (Pause)。

### 26.3.14 以太网MAC地址0低寄存器(EMAC\_MACA0L)

以太网 MAC 地址 0 低寄存器(EMAC\_MACA0L)包含了 6 字节的第一个 MAC 地址的低 32 位。

域	简称	复位值	类型	功能
位 31: 0	MA0L	0xFFFF FFFF	rw	MAC 地址低 (MAC Address0 [31: 0]) 此位域包含了前 6 字节 MAC 地址的低 32 位。MAC 使用这个位域来过滤所接收帧, 并将 MAC 地址插入到发送流控帧 (Pause)。



### 26.3.15 以太网MAC地址1高寄存器(EMAC\_MACA1H)

以太网 MAC 地址 1 高寄存器(EMAC\_MACA1H)包含了第二个 6 字节 MAC 地址的高 16 位。

如果 MAC 地址寄存器被设置成与 MII 时钟域双重同步，只有当以太网 MAC 地址 1 低寄存器(EMAC\_MACA1L)的位[31: 24]（在小端模式下）或位[7: 0]（在大端模式下）被写入时，才能触发同步功能。为了进行正确的同步更新，必须在至少 4 个目的时钟域周期之后，才允许向地址低寄存器连续写操作。

域	简称	复位值	类型	功能
位 31	AE	0x0	rw	地址使能 (Address Enable) 该位置起后，地址过滤器使用第二个 MAC 地址进行完美过滤。 复位后，地址过滤器将忽略要过滤的地址。
位 30	SA	0x0	rw	源地址 (Source Address) 该位置起后，MAC 地址 1[47: 0]用来与接收帧的源地址域进行比较。 复位后，MAC 地址 1[47: 0]用来与接收帧的目的地址域进行比较。
位 29: 24	MBC	0x00	rw	屏蔽字节控制 (Mask Byte Control) 这些位是指用于比较每个 MAC 地址字节的屏蔽控制位。 置起时，MAC 不会将接收到的目的地址或源地址的对应字节与 MAC 地址 1 寄存器的内容进行比较。每个控制位用来控制字节的屏蔽，如下所示： 位 29: EMAC_MACA1H [15: 8] 位 28: EMAC_MACA1H [7: 0] 位 27: EMAC_MACA1L[31: 24] ... 位 24: EMAC_MACA1L[7: 0] 可以通过屏蔽该地址的一个或多个字节来过滤一组地址（称为成组地址过滤）
位 23: 16	保留	0x00	resd	请保持默认值。
位 15: 0	MA1H	0xFFFF	rw	MAC Address1 [47: 32] 这些位包含了第二个 6 字节 MAC 地址的高 16 位(47: 32)。

### 26.3.16 以太网MAC地址1低寄存器(EMAC\_MACA1L)

以太网 MAC 地址 1 低寄存器(EMAC\_MACA1L)包含了第二个 6 字节 MAC 地址的低 32 位。

域	简称	复位值	类型	功能
位 31: 0	MA1L	0xFFFF FFFF	rw	MAC Address1 [31: 0] 这些位包含了第二个 6 字节 MAC 地址的低 32 位。在初始化流程后需要应用程序加载，才会定义这个位域的内容。

### 26.3.17 以太网MAC地址2高寄存器(EMAC\_MACA2H)

以太网 MAC 地址 2 高寄存器(EMAC\_MACA2H)包含了第二个 6 字节 MAC 地址的高 16 位。

如果 MAC 地址寄存器被设置为与 MII 时钟域双重同步，那么当以太网 MAC 地址 2 低寄存器(EMAC\_MACA2L)的位[31: 24]（小端模式）或位[7: 0]（大端模式）被写入时，才会触发同步功能。为了进行正确地同步更新，至少在 4 个目的时钟域周期之后，才允许对地址低寄存器进行连续写操作。

域	简称	复位值	类型	功能
位 31	AE	0x0	rw	地址使能 (Address Enable) 该位置起后，地址过滤器模块使用第二个 MAC 地址进行完美过滤。 复位后，地址过滤器模块会忽略要过滤的地址。
位 30	SA	0x0	rw	源地址 (Source Address) 该位置起后，使用 MAC 地址 1[47: 0]与接收帧的源地址域进行比较。 复位后，使用 MAC 地址 1[47: 0]与接收帧的目的地址域进行比较。
位 29: 24	MBC	0x00	rw	屏蔽字节控制 (Mask Byte Control) 这些位是指用于比较每个 MAC 地址的屏蔽控制位。置起时，MAC 不会将接收到的源地址或目的地址的对应字节与 MAC 地址 1 寄存器的内容进行比较。每个位用来控制字节的屏蔽，如下所示： 位 29: EMAC_MACA2H [15: 8] 位 28: EMAC_MACA2H [7: 0] 位 27: EMAC_MACA2L [31: 24] ... 位 24: EMAC_MACA2L [7: 0] 可以通过屏蔽该地址的一个或多个字节来过滤一组地址（称为成组地址过滤）
位 23: 16	保留	0x00	resd	请保持默认值。
位 15: 0	MA2H	0xFFFF	rw	(MAC Address2 High [47: 32]) 这些位包含了第二个 6 字节 MAC 地址的高 16 位 (47: 32)。

### 26.3.18 以太网MAC地址2低寄存器(EMAC\_MACA2L)

以太网 MAC 地址 2 低寄存器(EMAC\_MACA2L)包含了第二个 6 字节 MAC 地址的低 32 位。

域	简称	复位值	类型	功能
位 31: 0	MA2L	0xFFFF FFFF	rw	(MAC Address2 Low [31: 0]) 这些位包含了第二个 6 字节 MAC 地址的低 32 位。 在初始化流程之后，需要应用程序加载，才会定义这些位的内容。

### 26.3.19 以太网MAC地址3高寄存器(EMAC\_MACA3H)

以太网 MAC 地址 3 高寄存器(EMAC\_MACA3H)包含了第二个 6 字节 MAC 地址的高 16 位。

如果 MAC 地址寄存器被设置为与 MII 时钟域双重同步，那么当以太网 MAC 地址 3 低寄存器(EMAC\_MACA3L)的位[31: 24]（小端模式）或位[7: 0]（大端模式）被写入时，才会触发同步功能。为了进行正确地同步更新，至少在 4 个目的时钟域周期之后，才允许对地址低寄存器进行连续写操作。

域	简称	复位值	类型	功能
位 31	AE	0x0	rw	地址使能 (Address Enable) 该位置起后，地址过滤器使用第二个 MAC 地址进行完美过滤。 复位后，地址过滤器将忽略要过滤的地址。
位 30	SA	0x0	rw	源地址 (Source Address) 该位置起后，使用 MAC 地址 1[47: 0]与接收帧的源地址域进行比较。 复位后，使用 MAC 地址 1[47: 0]与接收帧的目的地址域进行比较。
位 29: 24	MBC	0x00	rw	屏蔽字节控制 (Mask Byte Control) 这些位是指用于比较每个 MAC 地址的屏蔽控制位。置起时，MAC 不会将接收到的源地址或目的地址的对应字节与 MAC 地址 1 寄存器的内容进行比较。每个位用来控制字节的屏蔽，如下所示： 位 29: EMAC_MACA3H [15: 8] 位 28: EMAC_MACA3H [7: 0] 位 27: EMAC_MACA3L [31: 24] ... 位 24: EMAC_MACA3L [7: 0] 可以通过屏蔽该地址的一个或多个字节来过滤一组地址（称为成组地址过滤）
位 23: 16	保留	0x00	resd	请保持默认值。
位 15: 0	MA3H	0xFFFF	rw	(MAC Address3 High [47: 32]) 这些位包含了第二个 6 字节 MAC 地址的低 16 位(47: 32)。

### 26.3.20 以太网MAC地址3低寄存器(EMAC\_MACA3L)

以太网 MAC 地址 3 低寄存器(EMAC\_MACA3L)包含了第二个 6 字节 MAC 地址的低 32 位。

域	简称	复位值	类型	功能
位 31: 0	MA3L	0xFFFF FFFF	rw	(MAC Address3 Low [31: 0]) 这些位包含了第二个 6 字节 MAC 地址的低 32 位。在初始化流程后需要应用程序加载，才会定义这个位域的内容。

### 26.3.21 以太网DMA总线模式寄存器(EMAC\_DMABM)

以太网 DMA 总线模式寄存器(EMAC\_DMABM)设定了 DMA 的总线工作模式。

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	AAB	0x0	rw	地址对齐 (Address-Aligned Beats) 当该位被置起且 FB 位等于 1 时, AHB 接口会生成与起始地址 LS 位对齐的突发传输。如果 FB 位等于 0, 第一次突发传输 (即访问数据缓冲器的起始地址) 可能不会与地址对齐, 但是随后的突发传输会与地址对齐。 该位仅适用于 GMAC-AHB 和 GMAC-AXI 配置, 在其它配置中, 该位是保留的。
位 24	PBLx8	0x0	rw	PBLx8 模式 (PBLx8 Mode) 被置起后, 该位将设置好的 PBL 值 (位 [22: 17] 和位 [13: 8]) 乘以 8。所以, DMA 根据 PBL 的值以 8, 16, 32, 64, 128, 和 256 节拍转发数据,
位 23	USP	0x0	rw	使用分散的 PBL (Use Separate PBL) 被置起后, Rx DMA 使用位[22: 17] 的设定值作为 PBL。位[13: 8]中的 PBL 值仅适用于 DMA 发送操作。被复位后, 位[13: 8]的 PBL 值既适用于 DMA 接收操作也适用于 DMA 发送操作。
位 22: 17	RDP	0x01	rw	接收 DMA PBL (Rx DMA PBL) 这些位域表示的是一次 DMA 接收操作所传输的最大节拍数。这个最大值用于单次读或写操作。 接收 DMA 每次在主机总线上进行突发传输时, 总会尝试按照 RPBL 的设定值进行突发传输。可以使用 1, 2, 4, 8, 16 和 32 来设置 RPBL。除此以外的其它值均会导致意外情况发生。 该位域仅适用于 USP 被置起时的情况。
位 16	FB	0x0	rw	固定的突发传输 (Fixed Burst) 该位控制着 AHB 主机接口是否进行固定突发传输。该位被置起后, AHB 接口在开始正常的突发传输过程中, 仅使用 SINGLE, INCR4, INCR8 或 INCR16。复位后, AHB 或 AXI 接口使用 SINGLE 和 INCR 突发传输。
位 15: 14	PR	0x0	rw	优先级比率 (Priority Ratio) 这些位控制着接收 DMA 和发送 DMA 之间的加权循环仲裁的优先级比率。这些位仅在位 1 (目的地址) 被复位时才有效。优先级比率为 Rx: Tx 或者 Tx: Rx, 具体取决于 27(TXPR)是置起还是复位。 00: 优先级比率为 1: 1。 01: 优先级比率为 2: 1。 10: 优先级比率为 3: 1。 11: 优先级比率为 4: 1。
位 13: 8	PBL	0x01	rw	可编程的突发长度 (Programmable Burst Length) 这些位定义了一次 DMA 操作中所传输的最大节拍数。这个最大值用于单次读写操作。 DMA 每次在主机总线上进行突发传输时, 总会尝试按照 PBL 的设定值进行突发传输。PBL 允许设置为 1, 2, 4, 8, 16 和 32。除此以外的其它值均会导致意外情况发生。当 USP 置起时, PBL 值仅适用于 DMA 发送操作。 如果要传输的节拍数量大于 32, 需按以下步骤操作: 1. 设置 PBLx8 模式。 2. 设置 PBL。 例如, 如果待传输的最大值为 64, 那么首先需要将 PBLx8 置 1, 再将 PBL 设置为 8。PBL 的值有以下限制: 可能的最大传输数会受到 MTL 层的发送 FIFO 和接收 FIFO 大小的限制, 以及 DMA 上的数据总线宽度的限制。FIFO 的限制: FIFO 所支持的最大传输次数是 FIFO 深度的一半, 除非另有规定
位 7	保留	0x0	resd	请保持默认值。
位 6: 2	DSL	0x00	rw	描述符跳跃长度 (Descriptor Skip Length)

				这些位定义了两个未链接的描述符之间要跳过的字距离。地址跳跃是指从当前描述符的末尾跳到下一个描述符的开头。 当 DSL 值等于 0 时，在环形模式下 DMA 将描述符视为连续的。
位 1	DA	0x0	rw	<b>DMA 仲裁 (DMA Arbitration)</b> 该位定义了通道 0 的发送路径和接收路径之间的仲裁方案。 0: Rx: Tx 或者 Tx: Rx 加权循环 通道之间的优先级取决于位[15: 14] (PR)所定义的优先级以及位 27(TXPR)所定义的优先级权重。 1: 固定的优先级 当位 27(TXPR)被置起时，发送路径的优先级高于接收路径。反之，接收路径高于发送路径。
位 0	SWR	0x1	rw	<b>软件复位 (SoftWare Reset)</b> 该位被置起后，MAC DMA 控制器将 MAC 逻辑电路和所有内部寄存器进行复位。在完成了所有复位操作之后，该位将自动清除。

### 26.3.22 以太网DMA发送轮询请求寄存器(EMAC\_DMATPD)

以太网 DMA 发送轮询请求寄存器(EMAC\_DMATPD)使能发送 DMA 来查询 DMA 是否拥有当前描述符。发送轮询请求可以用来唤醒处于暂停模式下的发送 DMA。发送 DMA 会因为发送的帧出现下溢错误或者未获得可用的描述符而进入暂停模式。可以在任何时刻发出轮询指令，发送 DMA 在重新开始从主机内存中取回当前描述符时，会将这一轮询指令复位。该寄存器始终读 0。

域	简称	复位值	类型	功能
位 31: 0	TPD	0x0000 0000	rrc	<b>发送轮询请求 (Transmit Poll Demand)</b> 当这些位被写入任意值时，DMA 读取 EMAC_DMACRD 指向的当前描述符。如果描述符不可用（被主机占用），发送动作暂停，状态寄存器的位 2（TU）会置起。如果描述符可用，则恢复发送动作。

### 26.3.23 以太网DMA接收轮询请求寄存器(EMAC\_DMARPDP)

以太网 DMA 接收轮询请求寄存器(EMAC\_DMARPDP)使能接收 DMA 来查询新的描述符。接收轮询请求可以用来唤醒处于暂停模式下的接收 DMA。接收 DMA 会因为描述符不可用而进入暂停模式。该寄存器始终读 0。

域	简称	复位值	类型	功能
位 31: 0	RPD	0x0000 0000	rrc	<b>接收轮询请求 (Receive Poll Demand)</b> 当这些位被写入任意值时，DMA 读取 EMAC_DMACRD 指向的当前描述符。如果描述符不可用（被主机占用），接收动作暂停，状态寄存器的位 7（RU）会置起。如果描述符可用，则恢复接收动作。

### 26.3.24 以太网DMA接收描述符列表地址寄存器 (EMAC\_DMARDLADDR)

以太网 DMA 接收描述符列表地址寄存器(EMAC\_DMARDLADDR)指向接收描述符列表的开头。描述符列表位于主机物理内存内，并且必须与字对齐。DMA 通过将相应的 LS 位置低将地址对齐总线宽度。只有当接收 DMA 停止的时候，才允许对该寄存器进行写操作。在接收 DMA 停止后，必须先写入该寄存器，再发出接收开始指令。

只有在接收 DMA 停止后，才能对该寄存器进行写操作，即将操作模式寄存器的位 1(SR)置 0。在接收 DMA 停止之后，该寄存器可以写入一个新的描述符列表地址。

当 SR 位被置 1 后，DMA 将使用最新编写的描述符基地址。

如果在 SR 位被置 0 后，该寄存器没有更改，那么 DMA 将使用先前 DMA 接收停止时的描述符地址。

域	简称	复位值	类型	功能
位 31: 0	SRL	0x0000 0000	rw	接收描述符基地址 (Start of Receive List) 这些位包含了接收描述符列表中第一个描述符的基地址。 32 位, 64 位或 128 位总线宽度的 LSB 位(1: 0, 2: 0 或 3: 0)被忽略, 被 DMA 视为 0。因此, 这些 LSB 位是只读的。

### 26.3.25 以太网DMA发送描述符列表地址寄存器 (EMAC\_DMATDLADDR)

以太网 DMA 发送描述符列表地址寄存器(EMAC\_DMATDLADDR)指向发送描述符列表的开头。描述符列表位于主机物理内存内，并且必须与字对齐。DMA 通过将相应的 LS 位置低将地址对齐总线宽度。

只有当发送 DMA 停止的时候，才允许写入该寄存器，即，将寄存器 6（操作模式寄存器）中的位 13(ST)置为 0。在发送 DMA 停止后，可以向该寄存器写入一个新的描述符列表地址。

当 SR 位被置 1 后，DMA 将使用最新编写的描述符基地址。

如果在 SR 位被置 0 后，该寄存器没有更改，那么 DMA 将使用先前 DMA 发送停止时的描述符地址。

域	简称	复位值	类型	功能
位 31: 0	STL	0x0000 0000	rw	发送描述符基地址 (Start of Transmit List) 这些位包含了发送描述符列表中第一个描述符的基地址。 32 位, 64 位或 128 位总线宽度的 LSB 位(1: 0, 2: 0 或 3: 0)被忽略, 被 DMA 视为 0。因此, 这些 LSB 位是只读的。

### 26.3.26 以太网DMA状态寄存器(EMAC\_DMASTS)

以太网 DMA 状态寄存器(EMAC\_DMASTS)包含了 DMA 报告给主机的所有状态位。软件驱动在中断服务程序或者轮询过程中会读取该寄存器。该寄存器中的大多数位都能中断主机。读取该寄存器不能清除其中的位。向该寄存器中的位[16: 0]（非保留位）写入 1'b1 才能清除这些位，写入 1'b0 不起作用。通过中断使能屏蔽寄存器中的相应位，可以屏蔽每一个位 (位[16: 0])。

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	TTI	0x0	ro	时间戳触发中断 (Timestamp Trigger Interrupt) 该位表示时间戳生成模块发生了中断事件。软件必须读取相应寄存器才能获取中断源。 该位仅适用于当 IEEE1588 时间戳功能被使能的情况。反之，该位是保留的。
位 28	MPI	0x0	ro	MAC PMT 中断 (MAC PMT Interrupt) 该位表示 PMT 模块发生了中断事件。软件必须读取以太网 MAC PMT 控制和状态寄存器 (EMAC_MACPMTCTRLSTS)才能获取中断源并清除中断源，将此位复位为 1'b0。 该位仅适用于当电源管理功能被使能的情况。反之，该位是保留的。
位 27	MMI	0x0	ro	MAC MMC 中断 (MAC MMC Interrupt)



				该位表示 MMC 模块发生了中断事件。软件必须读取对应寄存器才能获取中断源并清除中断源，以将此位复位为 1'b0。 该位仅适用于当 MAC 管理计数器 MMC 被使能的情况。反之，该位是保留的。
位 26	保留	0x0	resd	请保持默认值。
位 25: 23	EB	0x0	ro	<p><b>错误位 (Error Bits)</b>            这些位表示的是造成总线错误的错误类型。仅适用于当位 13 (FBI)被置起的情况。该位域不会产生中断。            000: 在 Rx DMA 传输写数据时出错            011: 在 Tx DMA 传输读数据时出错            100: 在 Rx DMA 描述符写访问时出错            101: 在 Tx DMA 描述符写访问时出错            110: 在 Rx DMA 描述符读访问时出错            111: 在 Tx DMA 描述符读访问时出错            注意: 001 和 010 表示保留。</p>
位 22: 20	TS	0x0	ro	<p><b>发送流程状态 (Transmit Process State)</b>            该位域表示发送 DMA 的 FSM 状态。这些位不会产生中断。            3' b000: 停止; 发出复位或停止发送命令            3' b001: 运行; 正在提取发送描述符            3' b010: 运行; 正在等待状态信息            3' b011: 运行; 正在读取主机内存缓存数据并将其排队到发送缓存(Tx FIFO)            3' b100: 时间戳写状态            3' b101: 保留备用            3' b110: 暂停; 发送描述符不可用或发送缓冲器下溢            3' b111: 运行; 正在关闭发送描述符</p>
位 19: 17	RS	0x0	ro	<p><b>接收流程状态 (Receive Process State)</b>            该位域表示接收 DMA 的 FSM 状态。这些位不会产生中断。            3' b000: 停止; 发出复位或停止发送命令            3' b001: 运行; 正在提取接收描述符            3' b010: 保留备用            3' b011: 运行; 正在等待接收数据包            3' b100: 暂停; 接收描述符不可用            3' b101: 运行; 正在关闭描述符            3' b110: 时间戳写状态            3' b111: 运行; 将接收缓冲器的数据包转发到主机内存</p>
位 16	NIS	0x0	rw1c	<p><b>正常中断汇总 (Normal Interrupt Summary)</b>            当中断使能寄存器中的相应中断位被使能时，正常中断汇总的值是下列位的逻辑或：            EMAC_DMASTS[0]: 发送中断            EMAC_DMASTS[2]: 发送缓冲不可用            EMAC_DMASTS[6]: 接收中断            EMAC_DMASTS[14]: 早接收中断            只有未被屏蔽的位才会影响到正常中断汇总位。            该位是粘滞位，必须在每次清除相应位（导致 NIS 被置起）时被清除（向该位写 1）。</p>
位 15	AIS	0x0	rw1c	<p><b>异常中断汇总 (Abnormal Interrupt Summary)</b>            当中断使能寄存器中的相应中断位被使能时，异常中断汇总的值是下列位的逻辑或            EMAC_DMASTS[1]: 发送流程停止            EMAC_DMASTS[3]: 发送 Jabber 超时            EMAC_DMASTS[4]: 接收 FIFO 溢出            EMAC_DMASTS[5]: 发送数据下溢            EMAC_DMASTS[7]: 接收缓冲不可用            EMAC_DMASTS[8]: 接收流程停止            EMAC_DMASTS[9]: 接收看门狗超时            EMAC_DMASTS[10]: 早发送中断            EMAC_DMASTS[13]: 线致命错误</p>



				只有未被屏蔽的位才会影响到异常中断汇总位。 该位是粘滞位，必须在每次清除相应位（导致 AIS 被置起）时被清除（向该位写 1）。
位 14	ERI	0x0	rw1c	早接收中断 (Early Receive Interrupt) 该位表示 DMA 填充了数据包的第一个数据缓存。当软件向该位写 1 或者置起该寄存器的位 6(RI)时（以先发生的为准），即可清除该位。
位 13	FBEI	0x0	rw1c	总线致命错误中断 (Fatal Bus Error Interrupt) 该位表示发生了位[25: 23]所描述的总线错误。该位被置起后，对应的 DMA 将关闭全部的总线访问。
位 12: 11	保留	0x0	resd	请保持默认值。
位 10	ETI	0x0	rw1c	早发送中断 (Early Transmit Interrupt) 该位表示要发送的帧已经全部被转送到 MTL 发送 FIFO。
位 9	RWT	0x0	rw1c	接收看门狗超时 (Receive Watchdog Timeout) 该位被置起后，该位表示在接收看门狗定时器在接收当前帧时到期，并且在看门狗发生超时后，当前帧被截断。
位 8	RPS	0x0	rw1c	接收流程停止 (Receive Process Stopped) 当接收流程进入停止状态时，该位置起。
位 7	RBU	0x0	rw1c	接收缓冲不可用 (Receive Buffer Unavailable) 该位表示主机占用了接收列表中的下一个描述符，使得 DMA 无法获取。因此接收流程暂停。主机需要更改描述符的所有权，并释放接收轮询请求指令才能恢复接收流程。如果未释放接收轮询请求指令，那么 DMA 在收到下一个输入帧的时候，会恢复接收流程。只有在 DMA 占用了前一个接收描述符的情况下，该位才会置起。
位 6	RI	0x0	rw1c	接收中断 (Receive Interrupt) 该位表示帧接收已完成。接收完成后，将在最后一个描述符中复位 RDES1(完成后关闭中断)的位 31，具体的帧状态信息会更新到描述符中。 接收流程仍处于运行状态。
位 5	UNF	0x0	rw1c	发送数据下溢 (Transmit Underflow) 该位表示发送缓存在发送帧期间发生数据下溢。发送流程暂停，并且将下溢错误 TDES0[1]位置起。
位 4	OVF	0x0	rw1c	接收溢出 (Receive Overflow) 该位表示接收缓存在接收帧期间发生数据溢出。如果部分帧已转发到应用程序，RDES0[11]位会置起溢出状态
位 3	TJT	0x0	rw1c	发送超长帧（即 Jabber）超时 (Transmit Jabber Timeout) 该位表示当帧大小超过 2048 字节（若 Jumbo 帧被使能，则为 10240 字节）时会发生发送 Jabber 定时器超时。Jabber 发生超时后，发送流程中止并进入停止状态。这将导致发送 Jabber 超时标志位 TDES0[14]被置起。
位 2	TBU	0x0	rw1c	发送缓冲不可用 (Transmit Buffer Unavailable) 该位表示主机占用了发送列表中的下一个描述符，使得 DMA 无法获取，因此，发送流程暂停。位[22: 20]解释了发送流程状态。如果需要恢复发送流程，可以通过设置 TDES0[31]更改描述符的所有权并释放发送轮询请求指令。
位 1	TPS	0x0	rw1c	发送流程停止 (Transmit Process Stopped) 当发送流程停止时，该位置起。
位 0	TI	0x0	rw1c	(Transmit Interrupt)发送中断 该位表示帧发送已完成。发送完成后，TDES0 的位 Bit 31 (OWN)会复位，具体的帧状态信息会更新到描述符。

### 26.3.27 以太网DMA工作模式寄存器(EMAC\_DMAOPM)

以太网 DMA 工作模式寄存器(EMAC\_DMAOPM)定义了发送和接收的工作模式和指令。该寄存器应该是 DMA 初始化过程中最后需要写入的 CSR。该寄存器也适用于 GMAC-MTL 配置，其中，未使用和保留的位是 24, 13, 2 和 1。对此寄存器连续的写之间要插入延时，延时值大于 4us。

域	简称	复位值	类型	功能
位 31: 27	保留	0x00	resd	请保持默认值。
位 26	DT	0x0	rw	不丢弃 TCP/IP 校验和错误帧 (Disable Dropping of TCP/IP Checksum Error Frames) 该位置起后，对于仅由接收校验和减负引擎检测到的有错误的帧，MAC 不会丢弃该帧。这些帧仅在封装数据里会出错，而在 MAC 接收到的以太网帧中不会出错（包含 FCS 错误）。该位复位后，如果 FEF 位复位，则所有错误帧均被丢弃。
位 25	RSF	0x0	rw	接收存储和转发 (Receive Store and Forward) 该位置起后，仅在一个完整的帧写入 Rx FIFO 后，MTL 才会读取 Rx FIFO 的帧，并忽略 RTC 位。该位复位后，Rx FIFO 工作在直通模式下，会受 RTC 指定阈值的限制。
位 24	DFRF	0x0	rw	不清除接收帧 (Disable Flushing of Received Frames) 该位置起后，接收 DMA 不会因为接收描述符或接收缓存不可用而清除接收帧。当该位复位后，遭遇上述情况时，接收 DMA 会清除接收帧。
位 23: 22	保留	0x000	resd	请保持默认值。
位 21	TSF	0x0	rw	发送存储和转发 (Transmit Store and Forward) 该位置起后，当一个完整的帧位于 MTL 发送 FIFO 时，就会开始发送，并且位[16: 14]设定的 TTC 值被忽略。该位的状态仅在发送流程停止时才允许更改。
位 20	FTF	0x0	rw	清空发送 FIFO (Flush Transmit FIFO) 该位被置起后，发送 FIFO 控制器逻辑电路被恢复至默认值，发送 FIFO 中的所有数据要么丢失，要么被清空。在完成清空动作后，该位即被清除。在该位被清除之前，不允许向工作模式寄存器进行写操作。MAC 发送器已接受的数据是不会被清空的，会被安排发送，并导致数据下溢和超短帧传输。(如果要连续通过指令更改此位，则连续的两次操作之间要插入延时，延时值大于 4us)
位 19: 17	保留	0x0	resd	请保持默认值。
位 16: 14	TTC	0x0	rw	发送阈值控制 (Transmit Threshold Control) 这些位控制的是 MTL 发送 FIFO 的阈值。当 MTL 发送 FIFO 中的帧超过阈值时，开始发送。另外，长度小于该阈值的完整帧也会被发送。这些位仅适用于当位 21(TSF)复位的情况下。 000: 64 001: 128 010: 192 011: 256 100: 40 101: 32 110: 24 111: 16
位 13	SSTC	0x0	rw	开始或停止发送指令 (Start or Stop Transmission Command) 该位被置起后，发送流程处于运行状态，DMA 会检查当前位置的发送列表，确定待发送的帧。DMA 要么从当前列表位置获取描述符（即发送描述符列表地址寄存器所设定的发送列表基地址），要么从之前发送流程中止的位置获取描述符。如果 DMA 未占用当前描述符，发送流程进入暂停状态，并且状态寄存器的位 2（发送缓冲不可用）会被置起。发送开始指令仅在发送停止的时候才有效。如果在未设置发送描述符列表地址寄存器之前就发出了发送开始指令，则 DMA 将会出现意想不到的后果。

				复位后，在发送完当前帧后，发送流程进入停止状态。保存发送列表中的下一个描述符位置，并在重新开始发送时，将该描述符位置作为当前位置。若要修改列表地址，需要在该位被复位时向发送描述符列表地址寄存器编写一个新值。该位被重新置起时，这个写入的新值会生效。只有在当前帧已发送完成或者发送流程进入暂停状态时，停止发送指令才会生效。
位 12: 8	保留	0x00	resd	请保持默认值。
				转发错误帧 (Forward Error Frames) 1: 表示除了过短帧以外，所有的帧都会转发给 DMA。 0: 接收 FIFO 会丢弃有错误的帧(CRC 错误、冲突错误、巨人帧、看门狗超时、溢出)。然而，如果在阈值模式下，已经把帧的起始字节指针转发给应用程序，则就不会丢弃该帧。接收 FIFO 会丢弃那些帧的起始字节还没有发送到 AHB 总线的出错帧。
位 7	FEF	0x0	rw	
				转发长度偏小的好帧 (Forward Undersized Good Frames) 该位被置起后，接收 FIFO 会转发填充字节和 CRC 等长度偏小的好帧（即没有错误且长度小于 64 字节的帧） 复位后，接收 FIFO 会丢弃所有长度小于 64 字节的帧，除非这个帧因为小于接收阈值而被传输给了应用程序，比如 RTC = 01
位 6	FUGF	0x0	rw	
位 5	保留	0x0	resd	请保持默认值。
				接收阈值控制 (Receive Threshold Control) 这两个位是用来控制 MTL 接收 FIFO 的阈值。当 MTL 接收 FIFO 里的帧大于阈值时，开始向 DMA 发送请求。此外，长度小于阈值的完整帧也会自动发送。 如果接收 FIFO 大小被设定为 128 字节，则值 11 不适用。这些位仅适用于当 RSF 位等于 0 的情况，当 RSF 位置 1 时，这些位被忽略。 00: 64 01: 32 10: 96 11: 128
位 4: 3	RTC	0x0	rw	
				操作第二帧 (Operate on Second Frame) 该位被置起后，它将指导 DMA 在获得第一帧的状态之前就开始处理发送数据的第二帧。
位 2	OSF	0x0	rw	
				发送或停止接收 (Start or Stop Receive) 该位被置起后，接收流程处于运行状态，DMA 会尝试从接收列表获取描述符并处理输入的帧。DMA 要么从当前列表位置获取描述符（即接收描述符列表地址寄存器所设定的地址），要么从之前接收流程中止的位置获取描述符。如果 DMA 未占用当前描述符，接收流程进入暂停状态，并且状态寄存器的位 7（接收缓冲不可用）会被置起。接收开始指令仅在接收停止的时候才有效。如果在未设置接收描述符列表地址寄存器之前就发出了接收开始指令，则 DMA 将会出现意想不到的后果。 该位被清除后，在发送完当前帧后，接收 DMA 操作停止。保存接收列表中的下一个描述符位置，并在重新开始接收时，将该描述符位置作为当前位置。只有在接收流程进入运行状态（正等着接收数据包）或者暂停状态时，停止接收指令才会生效。
位 1	SSR	0x0	rw	
位 0	保留	0x0	resd	请保持默认值。

### 26.3.28 以太网DMA中断使能寄存器(EMAC\_DMAIE)

以太网 DMA 中断使能寄存器(EMAC\_DMAIE)将使能由状态寄存器反馈的中断。把相应的位置 1'b1 可以使能相应的中断。在软件或硬件复位后，所有的中断关闭。

域	简称	复位值	类型	功能
位 31: 17	保留	0x0000	resd	请保持默认值。
位 16	NIE	0x0	rw	<p>使能正常中断 (Normal Interrupt enable)</p> <p>该位被置起后，正常中断汇总被启用。该位复位后，正常中断汇总关闭。该位可以使能寄存器 5（状态寄存器）中的下列中断：</p> <p>EMAC_DMASTS[0]: 发送中断</p> <p>EMAC_DMASTS[2]: 发送缓冲不可用</p> <p>EMAC_DMASTS[6]: 接收中断</p> <p>EMAC_DMASTS[14]: 早接收中断</p>
位 15	AIE	0x0	rw	<p>使能异常中断 (Abnormal interrupt enable)</p> <p>该位被置起后，异常中断汇总被启用。复位后，异常中断汇总关闭。该位可以使能状态寄存器中的下列中断：</p> <p>EMAC_DMASTS[1]: 发送流程停止</p> <p>EMAC_DMASTS[3]: 发送 Jabber 超时</p> <p>EMAC_DMASTS[4]: 接收溢出</p> <p>EMAC_DMASTS[5]: 发送数据下溢</p> <p>EMAC_DMASTS[7]: 发送缓冲不可用</p> <p>EMAC_DMASTS[8]: 接收流程停止</p> <p>EMAC_DMASTS[9]: 接收看门狗超时</p> <p>EMAC_DMASTS[10]: 早发送中断</p> <p>EMAC_DMASTS[13]: 总线致命错误</p>
位 14	ERE	0x0	rw	<p>早接收中断使能 (Early Receive interrupt Enable)</p> <p>通过正常中断汇总使能（位 16）将该位置起后，早接收中断被使能。复位后，早接收中断关闭。</p>
位 13	FBEE	0x0	rw	<p>总线致命错误使能 (Fatal Bus Error Enable)</p> <p>通过异常中断汇总使能（位 15）将该位置起后，总线致命错误被使能。复位后，总线致命错误使能中断关闭。</p>
位 12: 11	保留	0x0	resd	请保持默认值。
位 10	EIE	0x0	rw	<p>早发送中断使能 (Early transmit Interrupt Enable)</p> <p>通过异常中断汇总使能（位 15）将该位置起后，早发送中断被使能。复位后，早发送中断关闭。</p>
位 9	RWTE	0x0	rw	<p>接收看门狗超时使能 (Receive Watchdog Timeout Enable)</p> <p>通过异常中断汇总使能（位 15）将该位置起后，接收看门狗超时中断被使能。复位后，接收看门狗超时中断关闭。</p>
位 8	RSE	0x0	rw	<p>接收停止使能 (Receive Stopped Enable)</p> <p>通过异常中断汇总使能（位 15）将该位置起后，接收停止中断被使能。复位后，接收停止中断关闭。</p>
位 7	RBUE	0x0	rw	<p>接收缓存不可用使能 (Receive Buffer Unavailable Enable)</p> <p>通过异常中断汇总使能（位 15）将该位置起后，接收缓存不可用中断被使能。复位后，接收缓存不可用中断关闭。</p>
位 6	RIE	0x0	rw	<p>接收中断使能 (Receive Interrupt Enable)</p> <p>通过正常中断汇总使能（位 16）将该位置起后，接收中断被使能。复位后，接收中断关闭。</p>
位 5	UNE	0x0	rw	<p>下溢中断使能 (Underflow Interrupt Enable)</p> <p>通过异常中断汇总使能（位 15）将该位置起后，下溢中断被使能。复位后，下溢中断关闭。</p>
位 4	OVE	0x0	rw	<p>溢出中断使能 (Overflow Interrupt Enable)</p> <p>通过异常中断汇总使能（位 15）将该位置起后，溢出中断被使能。复位后，溢出中断关闭。</p>
位 3	TJE	0x0	rw	<p>发送 Jabber 超时使能 (Transmit Jabber Timeout Enable)</p>

				通过异常中断汇总使能（位 15）将该位置起后，发送 Jabber 超时中断被使能。复位后，发送 Jabber 超时中断关闭。
位 2	TUE	0x0	rw	发送缓存不可用使能（Transmit Buffer Unavailable Enable） 通过正常中断汇总使能（位 16）将该位置起后，发送缓存不可用中断被使能。复位后，发送缓存不可用中断关闭。
位 1	TSE	0x0	rw	发送停止使能（Transmit Stopped Enable） 通过异常中断汇总使能（位 15）将该位置起后，发送停止中断被使能。复位后，发送停止中断关闭。
位 0	TIE	0x0	rw	发送中断使能（Transmit Interrupt Enable） 通过正常中断汇总使能（位 16）将该位置起后，发送中断被使能。复位后，发送中断关闭。

对于以太网中断，只有在 DMA 状态寄存器的 TST 位或者 PMT 位为 '1'，并且相应中断没有被屏蔽时，或者在 NIS/AIS 位置 '1'，且相应中断被使能时，中断才会发生。

### 26.3.29 以太网 DMA 丢失帧和缓存溢出计数器寄存器 (EMAC\_DMAMFBOCNT)

DMA 包含两个计数器来追踪接收流程中丢失的帧数目。该寄存器反馈计数器的当前值。计数器用于诊断故障。位[15: 0]表示由于主机缓存不可用而丢失的帧数目。位[27: 17]表示由于缓存溢出(MTL 和 MAC)及被 MTL 丢弃的超短帧（长度小于 64 字节的好帧）而丢失的帧数目。

域	简称	复位值	类型	功能
位 31: 29	保留	0x0	resd	请保持默认值。
位 28	OBFOC	0x0	rrc	FIFO 溢出计数器溢出位（Overflow Bit for FIFO Overflow Counter） 每一次溢出帧计数器(位[27: 17])发生溢出时，该位就会被置起，即接收 FIFO 发生溢出，溢出帧计数器达到最大值。在这种情况下，溢出帧计数器被重置为全 0，该位表示已发生翻转。
位 27: 17	OFC	0x000	rrc	溢出帧计数器（Overflow Frame Counter） 这些位域表示应用程序所丢失的帧数目。
位 16	OBMFC	0x0	rrc	丢失帧计数器溢出位（Overflow Bit for Missed Frame Counter） 每当丢失帧计数器（位[15: 0]）发生溢出时，该位被置起，即 DMA 会忽略由于主机接收缓存不可用而传入的帧，且丢失帧计数器达到最大值。在这种情况下，丢失帧计数器被重置为全 0，该位代表翻转已发生。
位 15: 0	MFC	0x0000	rrc	丢失帧计数器（Missed Frame Counter） 这些位域表示由于主机接收缓存不可用而被控制器丢失的帧数目。每当 DMA 忽略一个传入的帧时，该计数器就会递增一次。当寄存器的 mci_be_i[0]位被置为 1'b1 时，该计数器被清除。

### 26.3.30 以太网 DMA 当前发送描述符寄存器(EMAC\_DMACTD)

当前主机发送描述符寄存器指向 DMA 正在读取的发送描述符的起始地址。

域	简称	复位值	类型	功能
位 31: 0	HTDAP	0x0000 0000	ro	主机发送描述符地址指针（Host Transmit Descriptor Address Pointer） 这些位在复位时被清 0，DMA 在运行过程中对指针进行更新。



### 26.3.31 以太网DMA当前接收描述符寄存器(EMAC\_DMACRD)

当前主机接收描述符寄存器指向 DMA 正在读取的接收描述符的起始地址。

域	简称	复位值	类型	功能
位 31: 0	HRDAP	0x0000 0000	ro	主机接收描述符地址指针 (Host Receive Descriptor Address Pointer) 这些位在复位时被清0, DMA 在运行过程中对指针进行更新。

### 26.3.32 以太网DMA当前发送缓存地址寄存器(EMAC\_DMACTBADDR)

当前主机发送缓存地址寄存器指向 DMA 正在读取的发送缓存地址。

域	简称	复位值	类型	功能
位 31: 0	HTBAP	0x0000 0000	ro	主机发送缓存地址指针 (Host Transmit Buffer Address Pointer) 这些位在复位时被清0, DMA 在运行过程中对指针进行更新。

### 26.3.33 以太网DMA当前接收缓存地址寄存器(EMAC\_DMACRBADDR)

当前主机接收缓存地址寄存器指向 DMA 正在读取的接收缓存地址。

域	简称	复位值	类型	功能
位 31: 0	HRBAP	0x0000 0000	ro	主机接收缓存地址指针 (Host Receive Buffer Address Pointer) 这些位在复位时被清0, DMA 在运行过程中对指针进行更新。

### 26.3.34 以太网MMC控制寄存器(EMAC\_MMCCTRL)

以太网 MMC 控制寄存器(EMAC\_MMCCTRL)定义了各管理计数器的工作模式。

域	简称	复位值	类型	功能
位 31: 4	保留	0x00000000	resd	请保持默认值。
位 3	FMC	0x0	rw	冻结 MMC 计数器 (Freeze MMC Counter) 置起后, 该位会冻结所有 MMC 计数器使它们保持当前值。除非将该位置 0, 否则, 不会因为接收帧或发送帧而更新 MMC 计数器。如果在读取 MMC 计数器时将 Read 位设置成了 Reset, 那么计数器也同样被清除。
位 2	RR	0x0	rw	读操作时复位 (Reset on Read) 该位置起后, 在读(复位后自动清除) MMC 计数器之后, 该计数器被重置为 0。读取最低字节位(Bits[7: 0])将清除计数器。
位 1	SCR	0x0	rw	计数器停止滚动 (Stop Counter Rollover) 该位置起后, 计数器在计数到最大值后不会滚动为 0
位 0	RC	0x0	rw	计数器复位 (Reset Counter) 该位置起后, 所有计数器被复位。在 1 个时钟周期后, 该位自动清除。

### 26.3.35 以太网MMC接收中断寄存器(EMAC\_MMCR1)

以太网 MMC 接收中断寄存器(EMAC\_MMCR1)包含着以下情况下所产生的中断。

- 接收统计计数器计数到其最大值的一半时（32 位计数器对应的是 0x8000\_0000，16 位计数器对应的是 0x8000）
- 接收统计计数器的计数超过其最大值（32 位计数器对应的是 0xFFFF\_FFFF，16 位计数器对应的是 0xFFFF）

当计数器停止滚动时，中断被置起，但是计数器仍为全 1。以太网 MMC 接收中断寄存器(EMAC\_MMCR1)是一个 32 位寄存器。读取产生中断的 MMC 计数器时会清除中断位。必须读取相应的计数器的低字节（Bits[7: 0]）才能清除中断位。

域	简称	复位值	类型	功能
位 31: 18	保留	0x0000	resd	请保持默认值。
位 17	RGUF	0x0	rrc	接收到好的单播帧 (Received Good Unicast Frames) 接收好的单播帧计数器计数达到最大值或者最大值的一半时，该位被置起。
位 16: 7	保留	0x000	resd	请保持默认值。
位 6	RFAE	0x0	rrc	接收帧对齐错误 (Received Frames Alignment Error) 对齐错误接收帧计数器计数到最大值的一半或最大值时，该位被置起。
位 5	RFCE	0x0	rrc	接收帧 CRC 错误 (Received Frames CRC Error) 接收 CRC 错误帧计数器计数到最大值的一半或最大值时，该位被置起。
位 4: 0	保留	0x00	resd	请保持默认值。

### 26.3.36 以太网MMC发送中断寄存器(EMAC\_MMCTI)

以太网 MMC 发送中断寄存器(EMAC\_MMCTI)包含着以下情况所产生的中断：发送统计计数器计数达到其最大值的一半（32 位计数器对应的是 0x8000\_0000，16 位计数器对应的是 0x8000），以及当发送计数器计数超过其最大值时（32 位计数器对应的是 0xFFFF\_FFFF，16 位计数器对应的是 0xFFFF）。当计数器停止滚动时，中断被置起，但计数器仍为全 1。以太网 MMC 发送中断寄存器(EMAC\_MMCTI)是一个 32 位寄存器。当读取产生中断的 MMC 计数器时，中断位被清除。必须读取相应的计数器的低字节（位[7: 0]）才能清除中断位。

域	简称	复位值	类型	功能
位 31: 22	保留	0x000	resd	请保持默认值。
位 21	TGF	0x0	rrc	发送好的帧 (Transmitted Good Frames) 发送好帧计数器计数达到最大值的一半或最大值时，该位被置起。
位 20: 16	保留	0x00	resd	请保持默认值。
位 15	TGFMSC	0x0	rrc	发送好的帧时遇到不止 1 个冲突 (Transmitted Good Frames More Single Collision) 发送时 1 次以上冲突后好帧计数器计数达到最大值的一半或最大值时，该位被置起。
位 14	TSCGFCI	0x0	rrc	发送单次冲突好帧计数器中断 (Transmit Single Collision Good Frame Counter Interrupt) 发送单次冲突好帧计数器计数达到最大值的一半或最大值时，该位被置起。
位 13: 0	保留	0x0000	resd	请保持默认值。



### 26.3.37 以太网MMC接收中断屏蔽寄存器(EMAC\_MMCRIM)

以太网 MMC 接收中断屏蔽寄存器(EMAC\_MMCRIM)包含着：当接收统计计数器计数达到其最大值的一半或最大值时所产生的中断的屏蔽。该寄存器是一个 32 位寄存器。

域	简称	复位值	类型	功能
位 31: 18	保留	0x0000	resd	请保持默认值。
位 17	RUGFCIM	0x0	rw	接收的好单播帧计数器中断屏蔽 (Received Unicast Good Frame Counter Interrupt Mask) 接收的好的单播帧计数器计数达到最大值的一半或最大值时，置起该位将屏蔽中断。
位 16: 7	保留	0x000	resd	请保持默认值。
位 6	RAEFACIM	0x0	rw	接收的对齐错误帧对齐计数器中断屏蔽 (Received Alignment Error Frame Alignment Counter Interrupt Mask) 接收的对齐错误帧计数器计数达到最大值的一半或最大值时，置起该位将屏蔽中断。
位 5	RCEFCIM	0x0	rw	接收的 CRC 错误帧计数器中断屏蔽 (Received CRC Error Frame Counter Interrupt Mask) 接收的 CRC 错误帧计数器计数达到最大值的一半或最大值时，置起该位将屏蔽中断。
位 4: 0	保留	0x00	resd	请保持默认值。

### 26.3.38 以太网MMC发送中断屏蔽寄存器(EMAC\_MMCTIM)

以太网 MMC 发送中断屏蔽寄存器(EMAC\_MMCTIM)包含着：当发送统计计数器计数达到其最大值的一半或最大值时所产生的中断的屏蔽。该寄存器是一个 32 位寄存器。

域	简称	复位值	类型	功能
位 31: 22	保留	0x000	resd	请保持默认值。
位 21	TGFCIM	0x0	rw	发送好帧计数器中断屏蔽 (Transmit Good Frame Counter Interrupt Mask) 发送好帧计数器计数达到最大值的一半或最大值时，置起该位将屏蔽中断。
位 20: 16	保留	0x00	resd	请保持默认值。
位 15	TMCGFCIM	0x0	rw	发送多冲突好帧计数器中断屏蔽 (Transmit Multiple Collision Good Frame Counter Interrupt Mask) 发送多冲突好帧计数器计数达到最大值的一半或最大值时，置起该位将屏蔽中断。
位 14	TSCGFCIM	0x0	rw	发送单冲突好帧计数器中断屏蔽 (Transmit Single Collision Good Frame Counter Interrupt Mask) 发送单冲突好帧计数器计数达到最大值的一半或最大值时，置起该位将屏蔽中断。
位 13: 0	保留	0x0000	resd	请保持默认值。

### 26.3.39 以太网MMC 1次冲突后发送”好”帧的计数器寄存器(EMAC\_MMCTFSCC)

该寄存器统计在半双工模式下，发送帧成功时只遇到一次冲突的帧的数目。

域	简称	复位值	类型	功能
位 31: 0	TGFSCC	0x0000 0000	ro	发送好帧单冲突计数器 (Transmitted Good Frames Single Collision Counter) 发送好帧计数器，帧在发送时遭遇一次冲突

### 26.3.40 以太网MMC 1次以上冲突后发送”好”帧的计数器寄存器 (EMAC\_MMCTFSCC)

该寄存器统计在半双工模式下，发送帧成功时遇到一次以上冲突的帧的数目。

域	简称	复位值	类型	功能
位 31: 0	TGFMSCC	0x0000 0000	ro	发送好帧多冲突计数器 (Transmitted Good Frame More Single Collision Counter) 发送好帧计数器,帧在发送时遭遇多次冲突

### 26.3.41 以太网MMC发送”好”帧计数器寄存器(EMAC\_MMCTFCNT)

该寄存器统计发送”好”帧的数目。

域	简称	复位值	类型	功能
位 31: 0	TGFC	0x0000 0000	ro	发送好帧计数器 (Transmitted Good Frames Counter) 发送的好帧计数器。

### 26.3.42 以太网MMC CRC错误接收帧计数器寄存器(EMAC\_MMCRFCECNT)

该寄存器统计接收到有CRC错误帧的数目。

域	简称	复位值	类型	功能
位 31: 0	RFCEC	0x0000 0000	ro	接收帧CRC错误计数器 (Received Frames CRC Error Counter) 接收到有CRC错误的帧计数器

### 26.3.43 以太网MMC对齐错误接收帧计数器寄存器(EMAC\_MMCRFAECNT)

该寄存器统计接收到有对齐错误帧的数目。

域	简称	复位值	类型	功能
位 31: 0	RFAEC	0x0000 0000	ro	接收到帧对齐错误计数器 (Received Frames Alignment Error Counter) 接收到有对齐错误的帧计数器。

### 26.3.44 以太网MMC 接收帧”好”单播帧计数器寄存器(EMAC\_MMCRGUFCNT)

该寄存器统计接收到”好”单播帧的数目。

域	简称	复位值	类型	功能
位 31: 0	RGUFC	0x0000 0000	ro	接收到好的单播帧计数器 (Received Good Unicast Frames Counter) 接收到好的单播帧的帧计数器。

### 26.3.45 以太网PTP时间戳控制寄存器(EMAC\_PTPTSCTRL)

该寄存器控制着接收器内系统时间的生成以及为 PTP 数据包加盖时间戳。

域	简称	复位值	类型	功能
位 31: 19	保留	0x0000	resd	请保持默认值。
位 18	EMAFPF	0x0	rw	<p>使能 MAC 地址来过滤 PTP 帧 (Enable MAC Address For PTP Frame Filtering)</p> <p>该位被置起后, 当通过以太网直接发送 PTP 时, 目的 MAC 地址 (与任意一个 MAC 地址寄存器匹配) 将用于过滤 PTP 帧。</p>
位 17: 16	SPPFTS	0x0	rw	<p>选择需要拍摄快照的 PTP 数据包 (Select PTP Packets For Taking Snapshot)</p> <p>00: 普通时钟 01: 边界时钟 10: 端对端透明时钟 11: 点对点透明时钟</p>
位 15	ESFMRTM	0x0	rw	<p>使能与主节点相关的消息的快照拍摄 (Enable Snapshot For Message Relevant To Master)</p> <p>该位被置起后, 仅对与主机节点相关的消息进行拍照。反之, 对与从节点相关的消息进行拍照。</p>
位 14	ETSFEM	0x0	rw	<p>使能时间消息的时间戳快照 (Enable Timestamp Snapshot For Event Messages)</p> <p>该位被置起后, 时间戳快照仅用于事件消息(SYNC, Delay_Req, Pdelay_Req, or Pdelay_Resp). 该位被复位后, 时间戳快照适用于除了 Announce、Management 和 Signaling 以外的所有消息。</p>
位 13	EPPFSIP4U	0x1	rw	<p>使能处理经由 IPv4-UDP 发送的 PTP 帧 (Enable Processing of PTP Frames Sent over IPv4-UDP)</p> <p>该位被置起后, MAC 接收器将处理通过 IPv4 数据包封装在 UDP 内的 PTP 数据包。该位被清除后, MAC 将忽略经由 UDP-IPv4 数据包传输的 PTP。该位是默认置起。</p>
位 12	EPPFSIP6U	0x0	rw	<p>使能处理经由 IPv6-UDP 发送的 PTP 帧 (Enable Processing of PTP Frames Sent over IPv6-UDP)</p> <p>该位被置起后, MAC 接收器将处理经由 IPv6 数据包封装在 UDP 内的 PTP 数据包。该位被清除后, MAC 将忽略经由 UDP-IPv6 数据包发送的 PTP。</p>
位 11	EPPEF	0x0	rw	<p>使能处理 EMAC 帧上的 PTP (Enable Processing of PTP over EMAC Frames)</p> <p>该位被置起后, MAC 接收器将处理直接封装在以太网帧内的 PTP 数据包。该位被清除后, MAC 会忽略以太网上的 PTP。</p>
位 10	EPPV2F	0x0	rw	<p>使用 V2 格式处理 PTP 数据包 (Enable PTP packet Processing for Version 2 Format)</p> <p>该位被置起后, 使用 1588 V2 格式来处理 PTP 数据包。反之, 使用 V1 格式处理 PTP 数据包。关于 IEEE 1588 V1 和 V2 格式, 请参考第 155 页的“PTP 处理和控制”章节。</p>
位 9	TDBRC	0x0	rw	<p>时间戳数字滚动或二进制滚动控制 (Timestamp Digital or Binary Rollover Control)</p> <p>该位被置起后, 时间戳低寄存器将在 0x3B9A_C9FF 值 (即 1 纳秒精度) 之后滚动, 并递增时间戳 (高) 秒。该位被复位后, 亚秒寄存器的滚动值为 0x7FFF_FFFF。必须根据 PTP 参考时钟频率和该位的值来正确设置亚秒递增量。</p>
位 8	ETAF	0x0	rw	<p>使能所有帧的时间戳功能 (Enable Timestamp for All Frames)</p> <p>该位被置起后, MAC 接收到的所有帧的时间戳快照功能被启用。</p>
位 7: 6	保留	0x0	resd	请保持默认值。
位 5	ARU	0x0	rw	<p>更新加数寄存器 (Addend Reg Update)</p> <p>该位被置起后, 以太网 PTP 时间戳加数寄存器 (EMAC_PTPTSAD)的内容会更新在 PTP 模块进行精细修</p>

				正。完成内容更新后，清除此位。该寄存器位在被置起之前应该为 0。
位 4	TITE	0x0	rw	使能时间戳中断触发 (Timestamp Interrupt Trigger Enable) 该位被置起后，当系统时间大于目标时间寄存器中写入的值时，产生时间戳中断。产生了时间戳触发中断后，该位被清除。
位 3	TU	0x0	rw	时间戳更新 (Timestamp Update) 该位被置起后，系统时间将根据系统时间-秒更新寄存器和系统时间-纳秒更新寄存器的设定值进行更新（即加或减）。 在被更新之前，该位应该为 0。硬件完成更新后，该位被复位。“时间戳高字”寄存器（如果已使能）不会被更新。
位 2	TI	0x0	rw	时间戳初始化 (Timestamp Initialize) 该位被置起后，将根据系统时间-秒更新寄存器（和寄存器系统时间-纳秒更新寄存器）的设定值来初始化（即覆盖）系统时间。 在被更新之前，该位应该为 0。完成初始化后，该位被复位。“时间戳高字”寄存器（如果已使能）才能被初始化。
位 1	TFCU	0x0	rw	时间戳精细更新或粗略更新 (Timestamp Fine or Coarse Update) 该位被置起后，该位表示使用精细更新方法来更新系统时间戳。被复位后，表示使用粗略的更新方法来更新系统时间戳。
位 0	TE	0x0	rw	使能时间戳 (Timestamp Enable) 该位被置起后，将会发送帧和接收帧添加时间戳。被关闭后，不再为发送帧和接收帧添加时间戳，时间戳生成器也会被挂起。被使能后，需要初始化时间戳（系统时间）。在接收端，只有当该位被置起后，MAC 才会处理 1588 帧。

## 时间戳快照对寄存器位的相依性

SPPFTS 位 17: 16	ESFMRTM 位 15	ETSFEM 位 14	PTP 消息
00 或 01	X	0	SYNC、Follow_Up、Delay_Req、Delay_Resp
00 或 01	1	1	Delay_Req
00 或 01	0	1	SYNC
10	N/A	0	SYNC、Follow_Up、Delay_Req、Delay_Resp
10	N/A	1	SYNC、Follow_Up
11	N/A	0	SYNC、Follow_Up、Delay_Req、Delay_Resp、Pdelay_Req、Pdelay_Resp
11	N/A	1	SYNC、Pdelay_Req、Pdelay_Resp

1 : N/A=不适用

2 : X=无关

### 26.3.46 以太网PTP亚秒递增寄存器(EMAC\_PTPSSINC)

只有在没有外部时间戳输入的情况下选择了 IEEE1588 时间戳功能，该寄存器才会存在。在粗略更新模式下（寄存器中的 TSCFUPDT 位），每个 clk\_ptp\_ref\_i 时间周期都会将该寄存器的值添加到系统时间。在精细更新模式下，每当累加器产生溢出时，就会向系统时间添加该寄存器的值。

域	简称	复位值	类型	功能
位 31: 8	保留	0x000000	resd	请保持默认值。
位 7: 0	SSIV	0x00	rw	亚秒递增值 (Sub-Second Increment Value) 该位域的设定值会在每个时钟周期(of clk_ptp_i)随着亚秒寄存器的值累加。例如，如果 PTP 时钟是 50 MHz (周期 20 ns)，当系统时间纳秒寄存器精度为 1 ns[通过在寄存器 448 即以太网 PTP 时间戳控制寄存器 (EMAC_PTPTCTRL)中设置位 9(TSCTRLSSR)]时，则需要将这些位的值设为 20 (0x14)。当 TSCTRLSSR 被清除后，纳秒寄存器分辨精度为~0.465ns。此时，需要将这些位的值设为 43 (0x2B)，即 20ns/0.465。

### 26.3.47 以太网PTP时间戳高寄存器(EMAC\_PTPTSH)

系统时间-秒寄存器以及系统时间-纳秒寄存器表示的是由 MAC 维护的系统时间的当前值。该值会持续更新。

域	简称	复位值	类型	功能
位 31: 0	TS	0x0000 0000	ro	时间戳秒 (Timestamp Second) 这些位域的值表示的是由 MAC 维护的当前系统时间的秒值。

### 26.3.48 以太网PTP时间戳低寄存器(EMAC\_PTPTSL)

该寄存器包含时间信息的低 32 位。该寄存器为只读，包含了系统时间的亚秒值。

域	简称	复位值	类型	功能
位 31	AST	0x0	ro	时间的加或减 (Add or Subtract Time) 该位被置起后，时间值将与更新寄存器的值相减。被复位后，时间值将与更新寄存器的值相加。
位 30: 0	TSS	0x0000 0000	ro	时间戳亚秒 (Timestamp Sub Seconds) 这个位域的值表示的是亚秒时间，精度为 0.46 ns。将寄存器 448（以太网 PTP 时间戳控制寄存器 (EMAC_PTPTCTRL)）的位 9(TSCTRLSSR)置起后，每个位代表 1ns，所编程的值不能超过 0x3B9A_C9FF。

### 26.3.49 以太网PTP时间戳高更新寄存器(EMAC\_PTPTSHUD)

系统时间-秒更新寄存器和系统时间-纳秒更新寄存器将初始化或更新由 MAC 维护的系统时间。在设置以太网 PTP 时间戳控制寄存器(EMAC\_PTPTCTRL)内的 TSINIT 或 TSUPDT 位之前，必须先向这两个寄存器进行写操作。

域	简称	复位值	类型	功能
位 31: 0	TS	0x0000 0000	rw	时间戳秒 (Timestamp Second) 这个位域表示的是将被初始化或者将被添加到系统时间的秒时间值。



### 26.3.50 以太网PTP时间戳低更新寄存器(EMAC\_PTPTSLUD)

只有在没有外部时间戳输入的情况下选择了 IEEE1588 时间戳功能，该寄存器才会存在。

域	简称	复位值	类型	功能
位 31	AST	0x0	rw	时间的加与减 (Add or Subtract Time) 该位被置起后，时间值将与更新寄存器的值相减。被复位后，时间值将与更新寄存器的值相加。
位 30: 0	TSS	0x0000 0000	rw	时间戳亚秒 (Timestamp Sub Seconds) 这个位域的值表示的是亚秒时间，精度为 0.46 ns。将寄存器 448 (以太网 PTP 时间戳控制寄存器 (EMAC_PTPTSCTRL)) 的位 9(TSCTRLSSR)置起后，每个位代表 1ns，所编程的值不能超过 0x3B9A_C9FF。

### 26.3.51 以太网PTP时间戳低更新寄存器(EMAC\_PTPTSLUD)

只有在没有外部时间戳输入的情况下选择了 IEEE1588 时间戳功能，该寄存器才会存在。

域	简称	复位值	类型	功能
位 31	AST	0x0	rw	时间的加与减 (Add or Subtract Time) 该位被置起后，时间值将与更新寄存器的值相减。被复位后，时间值将与更新寄存器的值相加。
位 30: 0	TSS	0x0000 0000	rw	时间戳亚秒 (Timestamp Sub Seconds) 这个位域的值表示的是亚秒时间，精度为 0.46 ns。将寄存器 448 (以太网 PTP 时间戳控制寄存器 (EMAC_PTPTSCTRL)) 的位 9(TSCTRLSSR)置起后，每个位代表 1ns，所编程的值不能超过 0x3B9A_C9FF。

### 26.3.52 以太网PTP时间戳加数寄存器(EMAC\_PTPTSAD)

该寄存器的值仅用于当系统时间被设置为精细更新模式的情况。该寄存器的值会在每个时钟周期(of clk\_ptp\_ref\_i)被添加到 32 位累加器，每当累加器溢出时，系统时间就被更新。

域	简称	复位值	类型	功能
位 31: 0	TAR	0x0000 0000	rw	时间戳加数寄存器 (Timestamp Addend Register) 这个位域表示的是为了实现时间同步，将要被添加到累加器的 32 位时间值。

### 26.3.53 以太网PTP目标时间高寄存器(EMAC\_PTPTTH)

目标时间秒寄存器和目标时间亚秒寄存器用于调度当系统时间超过寄存器的设定值时的中断事件。

域	简称	复位值	类型	功能
位 31: 0	TTSR	0x0000 0000	rw	目标时间秒寄存器 (Target Time Seconds Register) 该寄存器存放着秒时间值。当时间戳值等于或超过这两个目标时间戳寄存器的值时，MAC 会根据 PPS 控制寄存器的位[6: 5]，开始或中断 PPS 信号输出，并产生中断（如果已使能）

### 26.3.54 以太网PTP目标时间低寄存器(EMAC\_PTPTTL)

域	简称	复位值	类型	功能
位 31: 0	TTLR	0x0000 0000	rw	目标时间戳低寄存器 (Target Timestamp Low Register) 该寄存器存放着纳秒时间值(有符号的)。当时间戳的值等于这两个目标时间戳寄存器的值时，MAC 会根据 PPS 控制寄存器的 TRGTMODSEL0(位[6: 5])，开始或中断 PPS 信号输出，并产生中断（如果已使能） 当以太网 PTP 时间戳控制寄存器(EMAC_PTPTSCTRL)的位 9(TSCTRLSSR)被置起时，该位域的值不能超过 0x3B9A_C9FF PPS 信号输出的实际开始时间或中断时间可能会有高达 1 个亚秒增量值的误差。

## 26.3.55 以太网 PTP 时间戳状态寄存器 (EMAC\_PTPTSSR)

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	TTTR	0x0	ro	达到时间戳目标时间 (Timestamp Target Time Reached) 被置起后, 该位表示系统时间大于或等于目标时间秒寄存器和目标时间纳秒寄存器的设定值
位 0	TSO	0x0	ro	时间戳秒溢出 (Timestamp Seconds Overflow) 被置起后, 该位表示时间戳秒值 (支持 V2 格式) 已经溢出并超过了 32'hFFFF_FFFF。

## 26.3.56 以太网 PTP PPS 控制寄存器 (EMAC\_PTPPPSCR)

域	简称	复位值	类型	功能
位 31: 4	保留	0x00000000	resd	请保持默认值。
				PPS 频率选择 (PPS0 Output Frequency Control) 此位输出受 emac_pps_sel 位控制 (额外寄存器 2 (CRM_MISC2) 位 15) Emac_pps_sel=0: 0000: 1 Hz, 使用二进制翻转时, 脉冲宽度为 125 ms, 使用数字翻转时, 脉冲宽度为 100 ms 0001: 2 hz, 使用二进制翻转时, 占空比为 50% (不建议使用数字翻转) 0010: 4 hz, 使用二进制翻转时, 占空比为 50% (不建议使用数字翻转) 0011: 8 hz, 使用二进制翻转时, 占空比为 50% (不建议使用数字翻转) 0100: 16 hz, 使用二进制翻转时, 占空比为 50% (不建议使用数字翻转) ... 1111: 32.768 khz, 使用二进制翻转时, 占空比为 50% (不建议使用数字翻转)
位 3: 0	POFC	0x0	rw	Emac_pps_sel=1: 0000: 1 Hz, 脉冲宽度为一个 clk_ptp 周期 0001: 二进制翻转是 2hz, 占空比 50%, 数字翻转是 1hz (不建议使用数字翻转) 0010: 二进制翻转是 4hz, 占空比 50%, 数字翻转是 2hz (不建议使用数字翻转) 0011: 二进制翻转是 8hz, 占空比 50%, 数字翻转是 4hz (不建议使用数字翻转) ... 1111: 二进制翻转是 32.768khz, 占空比 50%, 数字翻转是 16.384khz (不建议使用数字翻转)  在 PPS 为非 0 值时, 不建议使用数字翻转, 因为这些情况下 PPS 输出的波形会不规则 (尽管其平均频率在任何一秒窗口内始终正确)



## 27 调试 (DEBUG)

### 27.1 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两种接口实现：串行调试接口 (SWD) 与 JTAG 调试接口。跟踪信息可以由单线的串行线查看接口收集，或者若需要的跟踪带宽较大时，也可以使用 TRACE 接口。跟踪和调试接口可以合并到一个接口中。

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

### 27.2 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。在深度休眠模式下，HICK 振荡器将开启并为 FCLK 和 HCLK 提供时钟。

MCU 内部有多个 ID 编码，调试器可通过地址为 0xE0042000 的 DEBUG\_IDCODE 来访问。它是 DEBUG 的一个组成部分，并且映射到外部 PPB 总线上。使用 JTAG 调试口或 SWD 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。

支持两种跟踪接口模式：串行线查看的单针模式和多针跟踪接口。

### 27.3 I/O 控制

AT32F403A/407/407A 在所有的封装里都支持 SWJ-DP 调试，该调试共使用 5 个普通 I/O 口。复位以后，SWJ-DP 作为默认功能可立即供调试器使用。为了防止 JTAG 的输入管脚悬空（尤其是 SWCLK/JTCK 这些时钟管脚），JTAG 输入脚硬件开启了内部上拉或下拉功能。NJTRST、JTDI 和 JTMS/SWDIO 硬件开启了内部上拉功能，JTCK/SWCLK 硬件开启了内部下拉功能。

当用户切换调试接口或不使用调试功能时，可配置 IOMUX\_MAPR 或者 IOMUX\_MAPR7 寄存器来释放这些专用 I/O 口。用户释放相应的调试 I/O 后，GPIO 控制器将取得控制，这些 I/O 口可作为普通的 I/O 口使用。

当用户需要使用跟踪功能时，可以通过设置 DEBUG 控制寄存器 (DEBUG\_CTRL) 的 TRACE\_IOEN 和 TRACE\_MODE 位来使能跟踪功能以及选择跟踪模式。

表 27-1 跟踪功能使能

TRACE_IOEN	功能说明
0	无跟踪（默认状态）
1	开启跟踪功能

表 27-2跟踪功能模式

TRACE _MODE[1: 0]		PB3/JTDO/TR ACESWO	PE2/TRAC ECK	PE3/TRAC ED[0]	PE4/TRAC ED[1]	PE5/TRAC D[2]	PE6/TRAC ED[3]
00	异步跟踪	TRACES WO	释放（可用作普通 I/O 口）				
01	同步跟踪	释放（可用作 普通 I/O 口）	TRAC ECK	TRAC ED[0]	释放 （可用作普通 I/O 口）		
10	同步跟踪		TRAC ECK	TRAC ED[0]	TRAC ED[1]	释放（可用作 普通 I/O 口）	
11	同步跟踪		TRACE CK	TRACE D[0]	TRACE D[1]	TRACE D[2]	TRACE D[3]

## 27.4DEBUG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 27-3 DEBUG 寄存器地址和复位值

寄存器简称	基地址	复位值
DEBUG_IDCODE	0xE004 2000	0xFFFF XXXX
DEBUG_CTRL	0xE004 2004	0x0000 0000

### 27.4.1 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
0x7005_0240	AT32F403AVCT7	256KB	LQFP100
0x7005_0241	AT32F403ARCT7	256KB	LQFP64
0x7005_0242	AT32F403ACCT7	256KB	LQFP48
0x7005_0243	AT32F403ACCU7	256KB	QFN48
0x7005_0344	AT32F403AVGT7	1024KB	LQFP100
0x7005_0345	AT32F403ARGT7	1024KB	LQFP64
0x7005_0346	AT32F403ACGT7	1024KB	LQFP48
0x7005_0347	AT32F403ACGU7	1024KB	QFN48
0x7005_0249	AT32F407VCT7	256KB	LQFP100
0x7005_024A	AT32F407RCT7	256KB	LQFP64
0x7005_034B	AT32F407VGT7	1024KB	LQFP100
0x7005_034C	AT32F407RGT7	1024KB	LQFP64
0x7005_02CD	AT32F403AVET7	512KB	LQFP100
0x7005_02CE	AT32F403ARET7	512KB	LQFP64
0x7005_02CF	AT32F403ACET7	512KB	LQFP48
0x7005_02D0	AT32F403ACEU7	512KB	QFN48
0x7005_02D1	AT32F407VET7	512KB	LQFP100
0x7005_02D2	AT32F407RET7	512KB	LQFP64
0x7005_0353	AT32F407AVGT7	1024KB	LQFP100
0x7005_0254	AT32F407AVCT7	256KB	LQFP100

### 27.4.2 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	I2C3_SMBUS_TIMEOUT	0x0	rw	I2C3 暂停控制位。 0: 正常工作; 1: I2C3 SMBUS 的超时控制停止工作。
位 30	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 29	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 28	TMR9_PAUSE	0x0	rw	TMR9 暂停控制位 0: 定时器正常工作; 1: 定时器停止工作。
位 27	TMR14_PAUSE	0x0	rw	TMR14 暂停控制位 0: 定时器正常工作; 1: 定时器停止工作。
位 26	TMR13_PAUSE	0x0	rw	TMR13 暂停控制位 0: 定时器正常工作; 1: 定时器停止工作。
位 25	TMR12_PAUSE	0x0	rw	TMR12 暂停控制位 0: 定时器正常工作; 1: 定时器停止工作。
位 24: 22	保留	0x0	resd	必须保持为 0。
位 21	CAN2_PAUSE	0x0	rw	CAN2 暂停控制位。 0: CAN2 正常运行; 1: CAN2 的接收寄存器不继续接收数据。
位 20	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 19	TMR6_PAUSE	0x0	rw	TMR6 暂停控制位 0: 定时器正常工作; 1: 定时器停止工作。
位 18	TMR5_PAUSE	0x0	rw	TMR5 暂停控制位 0: 定时器正常工作; 1: 定时器停止工作。
位 17	TMR8_PAUSE	0x0	rw	TMR8 暂停控制位 0: 定时器正常工作; 1: 定时器停止工作。
位 16	I2C2_SMBUS_TIMEOUT	0x0	rw	I2C2 暂停控制位。 0: 正常工作; 1: I2C2 SMBUS 的超时控制停止工作。
位 15	I2C1_SMBUS_TIMEOUT	0x0	rw	I2C1 暂停控制位。 0: 正常工作; 1: I2C1 SMBUS 的超时控制停止工作。
位 14	CAN1_PAUSE	0x0	rw	CAN1 暂停控制位。 0: CAN1 正常运行; 1: CAN1 的接收寄存器不继续接收数据。

位 13	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 12	TMR3_PAUSE	0x0	rw	TMR3 暂停控制位。 0: 定时器正常工作; 1: 定时器停止工作。
位 11	TMR2_PAUSE	0x0	rw	TMR2 暂停控制位。 0: 定时器正常工作; 1: 定时器停止工作。
位 10	TMR1_PAUSE	0x0	rw	TMR1 暂停控制位。 0: 定时器正常工作; 1: 定时器停止工作。
位 9	WWDT_PAUSE	0x0	rw	窗口看门狗暂停控制位。 0: 窗口看门狗正常工作; 1: 窗口看门狗停止工作。
位 8	WDT_PAUSE	0x0	rw	看门狗暂停控制位 0: 看门狗正常工作; 1: 看门狗停止工作。
位 7: 6	TRACE_MODE	0x0	rw	跟踪管脚分配控制 00: 异步模式; 01: 同步模式, 数据长度为 1; 10: 同步模式, 数据长度为 2; 11: 同步模式, 数据长度为 4。
位 5	TRACE_IOEN	0x0	rw	跟踪管脚分配使能 0: 无跟踪功能 (默认状态); 1: 开启跟踪功能。
位 4: 3	保留	0x0	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。

## 28 版本历史

文档版本历史

日期	版本	变更
2021.06.30	2.00	新版本发布
2021.10.22	2.01	1、更新图 4-1，便于阅读 2、修正部分章节文字笔误
2021.12.01	2.02	修正部分章节文字笔误

**重要通知 - 请仔细阅读**

买方自行负责对本文所述雅特力产品和服务的选择和使用，雅特力概不承担与选择或使用本文所述雅特力产品和服务相关的任何责任。

无论之前是否有过任何形式的表示，本文档不以任何方式对任何知识产权进行任何明示或默示的授权或许可。如果本文档任何部分涉及任何第三方产品或服务，不应被视为雅特力授权使用此类第三方产品或服务，或许可其中的任何知识产权，或者被视为涉及以任何方式使用任何此类第三方产品或服务或其中任何知识产权的保证。

除非在雅特力的销售条款中另有说明，否则，雅特力对雅特力产品的使用和 / 或销售不做任何明示或默示的保证，包括但不限于有关适销性、适合特定用途（及其依据任何司法管辖区的法律的对应情况），或侵犯任何专利、版权或其他知识产权的默示保证。

雅特力产品并非设计或专门用于下列用途的产品：（A）对安全性有特别要求的应用，例如：生命支持、主动植入设备或对产品功能安全有要求的系统；（B）航空应用；（C）汽车应用或汽车环境；（D）航天应用或航天环境，且/或（E）武器。如果雅特力产品不是为前述应用设计的，而采购商擅自将其用于前述应用，即使采购商向雅特力发出了书面通知，采购商仍将独自承担因此而导致的任何风险，雅特力的产品设计规格明确指定的汽车、汽车安全或医疗工业领域专用产品除外。根据相关政府主管部门的规定，ESCC、QML 或 JAN 正式认证产品适用于航天应用。

经销的雅特力产品如有不同于本文档中提出的声明和 / 或技术特点的规定，将立即导致雅特力针对本文所述雅特力产品或服务授予的任何保证失效，并且不应以任何形式造成或扩大雅特力的任何责任。

© 2019 雅特力科技（重庆）有限公司 保留所有权利