

GigaDevice Semiconductor Inc.

**GD32F3x0
ARM® Cortex™-M4 32-bit MCU**

用户手册

1.0 版本

(2017 年 6 月)

目录

目录	2
图索引	15
表索引	21
1. 系统及存储器架构	23
1.1. ARM Cortex-M4 处理器	23
1.2. 系统架构	24
1.3. 存储器映射	25
1.3.1. 位带操作	28
1.3.2. 片上 SRAM 存储器	28
1.3.3. 片上闪存	29
1.4. 引导配置	29
1.5. I/O 补偿单元	30
1.6. 系统配置寄存器 (SYSCFG)	30
1.6.1. 配置寄存器 0 (SYSCFG_CFG0)	30
1.6.2. EXTI 源选择寄存器 0 (SYSCFG_EXTISSL0)	31
1.6.3. EXTI 源选择寄存器 1 (SYSCFG_EXTISSL1)	32
1.6.4. EXTI 源选择寄存器 2 (SYSCFG_EXTISSL2)	34
1.6.5. EXTI 源选择寄存器 3 (SYSCFG_EXTISSL3)	35
1.6.6. 系统配置寄存器 2 (SYSCFG_CFG2)	36
1.6.7. I/O 补偿控制寄存器 (SYSCFG_CPSCTL)	37
1.7. 设备电子签名	38
1.7.1. 存储容量信息	38
1.7.2. 设备唯一 ID (96 位/位域)	38
2. 闪存控制器 (FMC)	40
2.1. 简介	40
2.2. 主要特性	40
2.3. 功能描述	40
2.3.1. 闪存结构	40
2.3.2. 读操作	41
2.3.3. FMC_CTL 寄存器解锁	41
2.3.4. 页擦除	41
2.3.5. 整片擦除	42
2.3.6. 主存储闪存块编程	43
2.3.7. 选项字节擦除	44
2.3.8. 选项字节编程	45

2.3.9. 选项字节说明	45
2.3.10. 页擦除/编程保护	46
2.3.11. 安全保护	47
2.4. FMC 寄存器	47
2.4.1. 等待状态寄存器 (FMC_WS)	47
2.4.2. 解锁寄存器 (FMC_KEY)	48
2.4.3. 选项字节解锁寄存器 (FMC_OBKEY)	48
2.4.4. 状态寄存器 (FMC_STAT)	49
2.4.5. 控制寄存器 (FMC_CTL)	49
2.4.6. 地址寄存器 0 (FMC_ADDR)	51
2.4.7. 选项字节状态寄存器 (FMC_OBSTAT)	51
2.4.8. 写保护寄存器 (FMC_WP)	52
2.4.9. 等待状态使能寄存器 (FMC_WSEN)	52
2.4.10. 产品 ID 寄存器 (FMC_PID)	53
3. 电源管理单元 (PMU)	54
3.1. 简介	54
3.2. 主要特性	54
3.3. 功能描述	54
3.3.1. 电池备份域	55
3.3.2. VDD/VDDA 电源域	56
3.3.3. 1.2V 电源域	57
3.3.4. 省电模式	58
3.4. PMU 寄存器	60
3.4.1. 控制寄存器 (PMU_CTL)	60
3.4.2. 电源控制和状态寄存器 (PMU_CS)	62
4. 复位和时钟单元 (RCU)	65
4.1. 复位控制单元 (RCTL)	65
4.1.1. 简介	65
4.1.2. 功能描述	65
4.2. 时钟控制单元 (CCTL)	66
4.2.1. 简介	66
4.2.2. 主要特性	68
4.2.3. 功能描述	68
4.3. RCU 寄存器	71
4.3.1. 控制寄存器 0 (RCU_CTL0)	71
4.3.2. 配置寄存器 0 (RCU_CFG0)	73
4.3.3. 中断寄存器 (RCU_INT)	76
4.3.4. APB2 复位寄存器 (RCU_APB2RST)	79
4.3.5. APB1 复位寄存器 (RCU_APB1RST)	80
4.3.6. AHB 使能寄存器 (RCU_AHBEN)	82

4.3.7. APB2 使能寄存器 (RCU_APB2EN).....	84
4.3.8. APB1 使能寄存器 (RCU_APB1EN).....	85
4.3.9. 备份域控制寄存器 (RCU_BDCTL).....	87
4.3.10. 复位源/时钟寄存器 (RCU_RSTSCK)	89
4.3.11. AHB 复位寄存器 (RCU_AHBRST)	90
4.3.12. 配置寄存器 1 (RCU_CFG1).....	92
4.3.13. 配置寄存器 2 (RCU_CFG2).....	93
4.3.14. 控制寄存器 1 (RCU_CTL1)	94
4.3.15. 附加时钟控制寄存器(RCU_ADDCTL).....	94
4.3.16. 附加时钟中断寄存器(RCU_ADDINT).....	95
4.3.17. APB1 附加使能寄存器 (RCU_ADDAPPB1EN)	96
4.3.18. APB1 附加复位寄存器 (RCU_ADD APB1RST)	97
4.3.19. 电源解锁寄存器 (RCU_VKEY)	97
4.3.20. 深度睡眠模式电压寄存器 (RCU_DSV).....	98
5. 时钟校准控制器 (CTC)	99
5.1. 简介.....	99
5.2. 主要特性	99
5.3. 功能描述	99
5.3.1. REF 同步脉冲发生器	100
5.3.2. CTC 校准计数器	100
5.3.3. 频率评估和自动校准过程	101
5.3.4. 软件编程指南	102
5.4. CTC 寄存器	102
5.4.1. 控制寄存器 0 (CTC_CTL0).....	102
5.4.2. 控制寄存器 1 (CTC_CTL1).....	104
5.4.3. 状态寄存器 (CTC_STAT)	105
5.4.4. 中断清除寄存器 (CTC_INTC).....	107
6. 中断/事件控制器 (EXTI)	108
6.1. 简介.....	108
6.2. 主要特性	108
6.3. 中断功能描述	108
6.4. 外部中断及事件(EXTI)结构框图.....	111
6.5. 外部中断及事件功能概述.....	111
6.6. 寄存器定义	112
6.6.1. 中断使能寄存器 (EXTI_INTEN)	112
6.6.2. 事件使能寄存器 (EXTI_EVENT)	113
6.6.3. 上升沿触发使能寄存器 (EXTI_RTEN)	113
6.6.4. 下降沿触发使能寄存器 (EXTI_FTEN)	114
6.6.5. 软件中断事件寄存器 (EXTI_SWIEV)	114

6.6.6. 挂起寄存器 (EXTI_PD)	115
7. 通用和备用输入/输出接口 (GPIO 和 AFIO)	116
7.1. 简介.....	116
7.2. 主要特性	116
7.3. 功能描述	116
7.3.1. GPIO 引脚配置	117
7.3.2. 备用功能 (AF)	118
7.3.3. 附加功能.....	118
7.3.4. 输入配置.....	118
7.3.5. 输出配置.....	119
7.3.6. 模拟配置.....	119
7.3.7. 备用功能 (AF) 配置	120
7.3.8. GPIO 锁定功能	120
7.3.9. GPIO 单周期输出翻转功能	121
7.3.10. GPIO 超高速驱动能力	121
7.4. GPIO 寄存器.....	121
7.4.1. 端口控制寄存器 (GPIOx_CTL, x=A..D,F)	121
7.4.2. 端口输出模式寄存器 (GPIOx_OMODE, x=A..D,F)	123
7.4.3. 端口输出速度寄存器 0 (GPIOx_OSPD0, x=A..D,F)	124
7.4.4. 端口上拉/下拉寄存器 (GPIOx_PUD, x=A..D,F)	126
7.4.5. 端口输入状态寄存器 (GPIOx_ISTAT, x=A..D,F)	128
7.4.6. 端口输出控制寄存器 (GPIOx_OCTL, x=A..D,F)	128
7.4.7. 端口位操作寄存器 (GPIOx_BOP, x=A..D,F)	129
7.4.8. 端口配置锁定寄存器 (GPIOx_LOCK, x=A,B)	129
7.4.9. 备用功能选择寄存器 0 (GPIOx_AFSEL0, x=A,B,C)	130
7.4.10. 备用功能选择寄存器 1 (GPIOx_AFSEL1, x=A,B,C)	131
7.4.11. 位清除寄存器 (GPIOx_BC, x=A..D,F)	132
7.4.12. 端口位翻转寄存器 (GPIOx_TG, x=A..D,F)	133
7.4.13. 端口输出速度寄存器 1 (GPIOx_OSPD1, x=A..D,F)	133
8. 循环冗余校验计算单元 (CRC).....	135
8.1. 简介.....	135
8.2. 主要特性	135
8.3. 功能描述	136
8.4. CRC 寄存器.....	136
8.4.1. 数据寄存器 (CRC_DATA).....	136
8.4.2. 独立数据寄存器 (CRC_FDATA).....	137
8.4.3. 控制寄存器 (CRC_CTL).....	137
8.4.4. 初值寄存器 (CRC_IDATA)	138
8.4.5. 多项式寄存器 (CRC_POLY)	138

9. 直接存储器访问控制器 (DMA)	140
9.1. 简介.....	140
9.2. 主要特性	140
9.3. 结构框图	141
9.4. 功能描述	141
9.4.1. DMA 操作	141
9.4.2. 外设握手	143
9.4.3. 仲裁	143
9.4.4. 地址生成	143
9.4.5. 循环模式	143
9.4.6. 存储器到存储器模式	144
9.4.7. 通道配置	144
9.4.8. 中断	144
9.4.9. DMA 请求映射	145
9.5. DMA 寄存器	148
9.5.1. 中断标志位寄存器 (DMA_INTF).....	148
9.5.2. 中断标志位清除寄存器 (DMA_INTC)	148
9.5.3. 通道 x 控制寄存器 (DMA_CHxCTL)	149
9.5.4. 通道 x 计数寄存器 (DMA_CHxCNT).....	151
9.5.5. 通道 x 外设基地址寄存器 (DMA_CHxPADDR)	152
9.5.6. 通道 x 存储器基地址寄存器 (DMA_CHxMADDR)	152
10. 调试 (DBG)	153
10.1. 简介	153
10.2. 串行调试接口简介	153
10.2.1. 引脚分配	153
10.2.2. JEDEC-106 ID code	153
10.3. 调试保持功能描述	153
10.3.1. 低功耗模式调试支持	153
10.3.2. TIMER, I2C, RTC, WWDGT 和 FWDGT 外设调试支持	154
10.4. DBG 寄存器	155
10.4.1. ID 寄存器 (DBG_ID).....	155
10.4.2. 控制寄存器 0 (DBG_CTL0).....	155
10.4.3. 控制寄存器 1 (DBG_CTL1)	157
11. 模数数字转换器(ADC).....	159
11.1. 简介	159
11.2. 主要特性	159
11.3. 引脚和内部信号	160
11.4. 功能描述	161

11.4.1. 校准 (ADC_CLB).....	161
11.4.2. 双时钟域架构	162
11.4.3. ADCON 开关.....	162
11.4.4. 规则组和注入组	162
11.4.5. 转换模式.....	162
11.4.6. 注入通道管理.....	166
11.4.7. 模拟看门狗	167
11.4.8. 数据对齐	167
11.4.9. 可编程的采样时间	168
11.4.10. 外部触发	169
11.4.11. DMA 请求	169
11.4.12. 温度传感器和内部参考电压 VREFINT	170
11.4.13. 电池电压监测	170
11.4.14. ADC 中断	170
11.4.15. 可编程分辨率(DRES) - 快速转换模式	171
11.4.16. 片上硬件过采样	171
11.5. ADC 寄存器.....	173
11.5.1. 状态寄存器 (ADC_STAT)	173
11.5.2. 控制寄存器 0 (ADC_CTL0).....	174
11.5.3. 控制寄存器 1 (ADC_CTL1).....	176
11.5.4. 采样时间寄存器 0 (ADC_SAMPT0).....	178
11.5.5. 采样时间寄存器 1 (ADC_SAMPT1).....	179
11.5.6. 注入通道数据偏移寄存器 x (ADC_IOFFx)(x=0..3)	179
11.5.7. 看门狗高阈值寄存器 (ADC_WDHT)	180
11.5.8. 看门狗低阈值寄存器 (ADC_WDLT)	180
11.5.9. 规则序列寄存器 0 (ADC_RSQ0).....	181
11.5.10. 规则序列寄存器 1 (ADC_RSQ1).....	181
11.5.11. 规则序列寄存器 2 (ADC_RSQ2).....	182
11.5.12. 注入序列寄存器 (ADC_ISQ)	183
11.5.13. 注入数据寄存器 x (ADC_IDATAx) (x= 0..3).....	183
11.5.14. 规则数据寄存器 (ADC_RDATA).....	184
11.5.15. 过采样控制寄存器(ADC_OVSAMPCTL)	184
12. 数模转换器 (DAC)	186
12.1. 简介	186
12.2. 主要特性	186
12.3. 功能描述	187
12.3.1. DAC 使能	187
12.3.2. DAC 输出缓冲	188
12.3.3. DAC 数据配置	188
12.3.4. DAC 触发	188
12.3.5. DAC 转换	188
12.3.6. DAC 噪声波	188

12.3.7. DAC 输出电压.....	189
12.3.8. DMA 请求	190
12.4. DAC 寄存器.....	190
12.4.1. 控制寄存器 (DAC_CTL)	190
12.4.2. 软件触发寄存器 (DAC_SWT)	191
12.4.3. DAC 12 位右对齐数据保持寄存器 (DAC_R12DH)	192
12.4.4. DAC 12 位左对齐数据保持寄存器 (DAC_L12DH).....	192
12.4.5. DAC 8 位右对齐数据保持寄存器 (DAC_R8DH)	193
12.4.6. DAC 数据输出寄存器 (DAC_DO)	193
12.4.7. 状态寄存器 (DAC_STAT)	193
13. 比较器 (CMP).....	195
13.1. 简介	195
13.2. 主要特性	195
13.3. 功能描述	195
13.3.1. 比机器时钟和复位	196
13.3.2. 比较器的输入和输出	196
13.3.3. 比较器供电模式	197
13.3.4. 比较器迟滞	197
13.3.5. 比较器寄存器写保护	197
13.4. 比较器寄存器	197
13.4.1. 控制状态寄存器(CMP_CS).....	197
14. 看门狗定时器 (WDGT)	201
14.1. 自由看门狗定时器 (FWDGT)	201
14.1.1. 简介	201
14.1.2. 主要特性	201
14.1.3. 功能描述	201
14.1.4. FWDGT 寄存器	203
14.2. 窗口看门狗定时器 (WWDT)	206
14.2.1. 简介	206
14.2.2. 主要特性	206
14.2.3. 功能描述	206
14.2.4. WWDT 寄存器	208
15. 实时时钟 (RTC)	210
15.1. 简介	210
15.2. 主要特性	210
15.3. 功能描述	211
15.3.1. 结构框图	211
15.3.2. 时钟源和预分频	211
15.3.3. 影子寄存器	212

15.3.4. 位域可屏蔽可配置的闹钟	212
15.3.5. RTC 初始化和配置	212
15.3.6. 读取日历	214
15.3.7. RTC 复位	215
15.3.8. RTC 移位功能	215
15.3.9. RTC 参考时钟检测	216
15.3.10. RTC 数字平滑校准	216
15.3.11. 时间戳功能	218
15.3.12. 侵入检测	218
15.3.13. 校准时钟输出	219
15.3.14. 闹钟输出	219
15.3.15. RTC 省电模式管理	220
15.3.16. RTC 中断	220
15.4. RTC 寄存器	220
15.4.1. 时间寄存器 (RTC_TIME)	220
15.4.2. 日期寄存器 (RTC_DATE)	221
15.4.3. 控制寄存器 (RTC_CTL)	222
15.4.4. 状态寄存器 (RTC_STAT)	224
15.4.5. 预分频寄存器 (RTC_PSC)	226
15.4.6. 闹钟 0 时间日期寄存器 (RTC_ALRM0TD)	226
15.4.7. 写保护钥匙寄存器 (RTC_WPK)	227
15.4.8. 亚秒寄存器 (RTC_SS)	228
15.4.9. 移位控制寄存器 (RTC_SHIFTCTL)	228
15.4.10. 时间戳时间寄存器 (RTC_TTS)	229
15.4.11. 时间戳日期寄存器 (RTC_DTS)	230
15.4.12. 时间戳亚秒寄存器 (RTC_SSTS)	230
15.4.13. 高精度频率补偿寄存器 (RTC_HRFC)	231
15.4.14. 侵入寄存器 (RTC_TAMP)	232
15.4.15. 闹钟 0 亚秒寄存器 (RTC_ALRM0SS)	234
15.4.16. 备份寄存器 (RTC_BKPx) (x=0..4)	235
16. 定时器 (TIMER)	236
16.1. 高级定时器 (TIMERx, x=0)	237
16.1.1. 简介	237
16.1.2. 主要特性	237
16.1.3. 结构框图	238
16.1.4. 功能描述	238
16.1.5. TIMERx 寄存器(x=0)	265
16.2. 通用定时器 L0 (TIMERx, x=1, 2)	288
16.2.1. 简介	288
16.2.2. 主要特性	288
16.2.3. 结构框图	289
16.2.4. 功能描述	289

16.2.5. TIMER _x 寄存器 ($x=1,2$)	305
16.3. 通用定时器 L2 (TIMER_x, $x=13$)	327
16.3.1. 简介	327
16.3.2. 主要特性	327
16.3.3. 结构框图	328
16.3.4. 功能描述	328
16.3.5. TIMER _x 寄存器($x=13$)	334
16.4. 通用定时器 L3 (TIMER_x,$x=14$)	343
16.4.1. 简介	343
16.4.2. 主要特性	343
16.4.3. 结构框图	344
16.4.4. 功能描述	344
16.4.5. TIMER _x 寄存器($x=14$)	359
16.5. 通用定时器 L4 (TIMER_x,$x=15,16$)	376
16.5.1. 简介	376
16.5.2. 主要特性	376
16.5.3. 结构框图	377
16.5.4. 功能描述	377
16.5.5. TIMER _x 寄存器($x=15,16$)	389
16.6. 基本定时器 (TIMER_x, $x=5$)	403
16.6.1. 简介	403
16.6.2. 主要特性	403
16.6.3. 结构框图	403
16.6.4. 功能描述	403
16.6.5. TIMER _x 寄存器($x=5$)	406
17. 红外线接口 (IFRP)	411
17.1. 简介	411
17.2. 主要特性	411
17.3. 功能描述	411
18. 通用同步异步收发器 (USART)	413
18.1. 简介	413
18.2. 主要特性	413
18.3. 功能描述	414
18.3.1. USART 帧格式	415
18.3.2. 波特率发生	416
18.3.3. USART 发送器	416
18.3.4. USART 接收器	417
18.3.5. DMA 方式访问数据缓冲区	419
18.3.6. 硬件流控制	420

18.3.7.	多处理器通信	421
18.3.8.	LIN 模式	422
18.3.9.	同步通信模式	422
18.3.10.	串行红外(IrDA SIR)编解码功能模块	423
18.3.11.	半双工通信模式	424
18.3.12.	智能卡(ISO7816)模式	425
18.3.13.	自动波特率检测	426
18.3.14.	ModBus 通信	427
18.3.15.	接收 FIFO	427
18.3.16.	从 DeepSleep 模式唤醒	427
18.3.17.	USART 中断	428
18.4.	寄存器定义	429
18.4.1.	USART 控制寄存器 0 (USART_CTL0)	429
18.4.2.	USART 控制寄存器 1 (USART_CTL1)	432
18.4.3.	USART 控制寄存器 2 (USART_CTL2)	434
18.4.4.	USART 波特率寄存器 (USART_BAUD)	437
18.4.5.	USART 保护时间和预分频器寄存器 (USART_GP)	437
18.4.6.	USART 接收超时寄存器 (USART_RT)	438
18.4.7.	USART 请求寄存器 (USART_CMD)	439
18.4.8.	USART 状态寄存器 (USART_STAT)	440
18.4.9.	USART 中断标志清除寄存器 (USART_INTC)	443
18.4.10.	USART 数据接收寄存器 (USART_RDATA)	445
18.4.11.	USART 数据发送寄存器 (USART_TDATA)	445
18.4.12.	USART 接收 FIFO 控制和状态寄存器 (USART_RFCS)	446
19.	内部集成电路总线接口 (I2C)	448
19.1.	简介	448
19.2.	主要特性	448
19.3.	功能描述	449
19.3.1.	SDA 线和 SCL 线	449
19.3.2.	数据有效性	450
19.3.3.	开始和停止状态	450
19.3.4.	时钟同步	450
19.3.5.	仲裁	451
19.3.6.	I2C 通讯流程	451
19.3.7.	软件编程模型	452
19.3.8.	SCL 线控制	460
19.3.9.	DMA 模式下数据传输	461
19.3.10.	报文错误校验	461
19.3.11.	SMBus 支持	461
19.3.12.	状态、错误和中断	463
19.4.	I2C 寄存器	463

19.4.1. 控制寄存器 0 (I2C_CTL0)	463
19.4.2. 控制寄存器 1 (I2C_CTL1)	465
19.4.3. 从机地址寄存器 0 (I2C_SADDR0)	466
19.4.4. 从机地址寄存器 1 (I2C_SADDR1)	466
19.4.5. 传输缓冲区寄存器 (I2C_DATA)	467
19.4.6. 传输状态寄存器 0 (I2C_STAT0)	467
19.4.7. 传输状态寄存器 1 (I2C_STAT1)	469
19.4.8. 时钟配置寄存器 (I2C_CKCFG)	470
19.4.9. 上升时间寄存器 (I2C_RT)	471
19.4.10. 快速+ 模式配置寄存器 (I2C_FMPCFG)	471
20. 串行外设接口/片上音频接口 (SPI/I2S)	472
20.1. 简介	472
20.2. 主要特性	472
20.2.1. SPI 主要特性	472
20.2.2. I2S 主要特性	472
20.3. SPI 结构框图	473
20.4. SPI 信号线描述	473
20.4.1. 常规配置 (非 SPI 四线模式)	473
20.4.2. SPI 四线配置	473
20.5. SPI 功能描述	474
20.5.1. SPI 时序和数据帧格式	474
20.5.2. NSS 功能	475
20.5.3. SPI 运行模式	476
20.5.4. DMA 功能	484
20.5.5. CRC 功能	484
20.6. SPI 中断	485
20.6.1. 状态标志位	485
20.6.2. 错误标志	485
20.7. I2S 结构框图	486
20.8. I2S 信号线描述	486
20.9. I2S 功能描述	487
20.9.1. I2S 音频标准	487
20.9.2. I2S 时钟	494
20.9.3. 运行	495
20.9.4. DMA 功能	497
20.10. I2S 中断	497
20.10.1. 状态标志位	497
20.10.2. 错误标志	498
20.11. SPI/I2S 寄存器	498

20.11.1.	控制寄存器 0 (SPI_CTL0).....	498
20.11.2.	控制寄存器 1 (SPI_CTL1).....	500
20.11.3.	状态寄存器 (SPI_STAT).....	501
20.11.4.	数据寄存器 (SPI_DATA).....	503
20.11.5.	CRC 多项式寄存器 (SPI_CRCPOLY)	503
20.11.6.	接收 CRC 寄存器 (SPI_RCRC)	504
20.11.7.	发送 CRC 寄存器 (SPI_TCRC).....	504
20.11.8.	I2S 控制寄存器 (SPI_I2SCTL)	505
20.11.9.	I2S 时钟预分频寄存器 (SPI_I2SPSC).....	506
20.11.10.	SPI1 四线 SPI 控制寄存器 (SPI_QCTL)	507
21.	HDMI-CEC 控制器(HDMI-CEC)	509
21.1.	简介	509
21.2.	主要特性	509
21.3.	功能描述	509
21.3.1.	CEC 总线引脚.....	509
21.3.2.	信息说明	510
21.3.3.	位时序说明	511
21.3.4.	仲裁	511
21.3.5.	SFTOPT 位说明	512
21.3.6.	错误定义	513
21.3.7.	HDMI-CEC 中断	516
21.4.	HDMI-CEC 寄存器	516
21.4.1.	控制寄存器 (CEC_CTL)	516
21.4.2.	配置寄存器 (CEC_CFG)	517
21.4.3.	数据发送寄存器 (CEC_TDATA)	518
21.4.4.	数据接收寄存器 (CEC_RDATA)	519
21.4.5.	中断标志寄存器 (CEC_INTF)	519
21.4.6.	中断使能寄存器 (CEC_INTEN)	521
22.	触摸传感控制器 (TSI)	524
22.1.	简介	524
22.2.	主要特性	524
22.3.	功能描述	524
22.3.1.	TSI 框图.....	524
22.3.2.	触摸传感技术概述	525
22.3.3.	电荷转移序列	525
22.3.4.	电荷转移序列状态机	527
22.3.5.	状态时钟和持续时间	528
22.3.6.	PIN 模式和 TSI 控制	529
22.3.7.	ASW 和 I/O 迟滞模式.....	529
22.3.8.	TSI 操作流	530

22.3.9. TSI 标志和中断.....	530
22.3.10. TSI GPIOs	530
22.4. TSI 寄存器.....	532
22.4.1. 控制寄存器 0 (TSI_CTL0).....	532
22.4.2. 中断使能寄存器 (TSI_INTEN)	534
22.4.3. 中断标志位清除寄存器 (TSI_INTC).....	535
22.4.4. 中断标志位寄存器 (TSI_INTF)	535
22.4.5. 引脚迟滞模式寄存器 (TSI_PHM)	536
22.4.6. 模拟开关寄存器 (TSI_ASW).....	536
22.4.7. 采样配置寄存器 (TSI_SAMPCFG)	537
22.4.8. 通道配置寄存器 (TSI_CHCFG).....	537
22.4.9. 组控制寄存器 (TSI_GCTL)	538
22.4.10. 组 x 周期数寄存器 (TSI_GxCYCN) (x= 0..5).....	539
22.4.11. 控制寄存器 1(TSI_CTL1).....	539
23. 通用串行总线全速接口 (USBFS)	541
23.1. 简介	541
23.2. 主要特性	541
23.3. 结构框图	542
23.4. 信号线描述.....	542
23.5. 功能描述	542
23.5.1. USBFS 时钟及工作模式.....	542
23.5.2. USB 主机功能.....	544
23.5.3. USB 设备功能.....	546
23.5.4. OTG 功能概述	547
23.5.5. 数据 FIFO.....	548
23.5.6. 操作手册	550
23.6. 中断.....	553
23.7. USBFS 寄存器.....	555
23.7.1. 全局控制与状态寄存器组	555
23.7.2. 主机控制和状态寄存器	576
23.7.3. 设备控制和状态寄存器	587
23.7.4. 电源和时钟控制寄存器 (USBFS_PWRCLKCTL).....	609
24. 版本历史	610

图索引

图 1-1. Cortex™-M4 处理器结构框图.....	24
图 1-2. GD32F3x0 系列器件的系统架构示意图.....	25
图 2-1. 页擦除操作流程	42
图 2-2. 整片擦除操作流程.....	43
图 2-3. 字编程操作流程	44
图 3-1. 电源域概览	55
图 3-2. 上电/掉电复位波形图	56
图 3-3. LVD 阈值波形图	57
图 4-1. 系统复位电路	66
图 4-2. 时钟树	67
图 4-3. HXTAL 时钟源	68
图 5-1. CTC 简介	100
图 5-2. CTC 校准计数器	101
图 6-1. EXTI 结构框图	111
图 7-1. 标准 I/O 端口位的基本结构.....	117
图 7-2. 浮空/上拉/下拉输入配置	118
图 7-3. 输出配置.....	119
图 7-4. 模拟高阻配置	120
图 7-5. 备用功能配置	120
图 8-1. CRC 计算单元框图	135
图 9-1. DMA 结构框图	141
图 9-2. 握手机制.....	143
图 9-3. DMA 中断逻辑图	145
图 9-4. DMA 请求映射	146
图 11-1. ADC 模块框图.....	161
图 11-2. 单次转换模式	162
图 11-3. 连续转换模式	163
图 11-4. 扫描转换模式, 连续转换模式禁用	164
图 11-5. 扫描转换模式, 连续转换模式使能	165
图 11-6. 间断转换模式	165
图 11-7. 自动注入, CTN=1	166
图 11-8. 触发注入	167
图 11-9. 12 位分辨率的数据对齐	167
图 11-10. 10 位分辨率的数据对齐	168
图 11-11. 8 位分辨率的数据对齐	168
图 11-12. 6 位分辨率的数据对齐	168
图 11-13. 20 位到 16 位的结果截断	172
图 11-14. 右移 5 位和取整的数例	172
图 12-1. DAC 结构框图.....	187
图 12-2. DAC LFSR 算法	189

图 12-3. DAC 三角噪声模式生成的波形	189
图 13-1. GD32F3x0 芯片的比较器框图.....	196
图 14-1. 自由看门狗定时器框图	202
图 14-2. 窗口看门狗定时器框图	206
图 14-3. 窗口看门狗定时器时序图.....	207
图 15-1. RTC 结构框图.....	211
图 16-1. 高级定时器结构框图	238
图 16-2. 内部时钟分频为 1 时正常模式下的控制电路.....	239
图 16-3. 当预分频器的参数从 1 变到 2 时，计数器的时序图	240
图 16-4. 向上计数时序图， $PSC=0/1$	241
图 16-5. 向上计数时序图，在运行时改变 $TIMERx_CAR$ 寄存器的值	242
图 16-6. 向下计数时序图， $PSC=0/1$	243
图 16-7. 向下计数时序图，在运行时改变 $TIMERx_CAR$ 寄存器值	244
图 16-8. 中央计数模式计数器时序图	245
图 16-9. 中央计数模式下计数器重复时序图	246
图 16-10. 在向上计数模式下计数器重复时序图	246
图 16-11. 在向下计数模式下计数器重复时序图	247
图 16-12. 输入捕获逻辑	248
图 16-13. 三种输出比较模式.....	250
图 16-14. EAPWM 时序图	251
图 16-15. CAPWM 时序图	251
图 16-16. 带死区时间的互补输出	254
图 16-17. 通道响应中止输入（高电平有效）时，输出信号的行为	255
图 16-18. 编码器接口模式下计数器运行例子.....	256
图 16-19. CIOFE0 极性反相的编码器接口模式下的例子	256
图 16-20. 霍尔传感器用在 BLDC 电机控制中	257
图 16-21. 两个定时器之间的霍尔传感器时序图	258
图 16-22. 复位模式下的控制电路	259
图 16-23. 暂停模式下的控制电路	259
图 16-24. 事件模式下的控制电路	260
图 16-25. 单脉冲模式， $TIMERx_CHxCV = 0x04$ $TIMERx_CAR=0x60$	260
图 16-26. 定时器 0 主/从模式的例子	261
图 16-27. 用定时器 1 的使能信号触发定时器 0	262
图 16-28. 用定时器 1 的更新事件来触发定时器 0	262
图 16-29. 用定时器 1 的使能来选通定时器 0	263
图 16-30. 用定时器 1 的 O0CPRE 信号选通定时器 1	264
图 16-31. 用定时器 1 的 CIO 输入来触发定时器 0 和定时器 1	264
图 16-32. 通用定时器 L0 结构框图.....	289
图 16-33. 内部时钟分频为 1 时正常模式下的控制电路.....	290
图 16-34. 当预分频器的参数从 1 变到 2 时，计数器的时序图	291
图 16-35. 向上计数时序图， $PSC=0/1$	292
图 16-36. 向上计数时序图，在运行时改变 $TIMERx_CAR$ 寄存器的值	292
图 16-37. 向下计数时序图， $PSC=0/1$	293
图 16-38. 向下计数时序图，在运行时改变 $TIMERx_CAR$ 寄存器值	294

图 16-39. 中央计数模式计数器时序图	295
图 16-40. 输入捕获逻辑	296
图 16-41. 三种输出比较模式	298
图 16-42. EAPWM 时序图	299
图 16-43. CAPWM 时序图	299
图 16-44. 编码器接口模式下计数器运行例子	301
图 16-45. CI0FE0 极性反相的编码器接口模式下的例子	301
图 16-46. 复位模式下的控制电路	302
图 16-47. 暂停模式下的控制电路	302
图 16-48. 事件模式下的控制电路	303
图 16-49. 单脉冲模式, $\text{TIMERx_CHxCV} = 0x04$ $\text{TIMERx_CAR}=0x60$	304
图 16-50. 通用定时器 L2 结构框图	328
图 16-51. 内部时钟分频为 1 时正常模式下的控制电路	328
图 16-52. 当预分频器的参数从 1 变到 2 时, 计数器的时序图	329
图 16-53. 向上计数时序图, $\text{PSC}=0/1$	330
图 16-54. 向上计数时序图, 在运行时改变 TIMERx_CAR 寄存器的值	330
图 16-55. 输入捕获逻辑	331
图 16-56. 三种输出比较模式	333
图 16-57. PWM 时序图	334
图 16-58. 通用定时器 L3 结构框图	344
图 16-59. 内部时钟分频为 1 时正常模式下的控制电路	345
图 16-60. 当预分频器的参数从 1 变到 2 时, 计数器的时序图	346
图 16-61. 向上计数时序图, $\text{PSC}=0/1$	347
图 16-62. 向上计数时序图, 在运行时改变 TIMERx_CAR 寄存器的值	348
图 16-63. 在向上计数模式下计数器重复时序图	349
图 16-64. 输入捕获逻辑	350
图 16-65. 三种输出比较模式	352
图 16-66. PWM 时序图	353
图 16-67. 带死区时间的互补输出	355
图 16-68. 通道响应中止输入 (高电平有效) 时, 输出信号的行为	356
图 16-69. 复位模式下的控制电路	357
图 16-70. 暂停模式下的控制电路	357
图 16-71. 事件模式下的控制电路	358
图 16-72. 单脉冲模式, $\text{TIMERx_CHxCV} = 0x04$ $\text{TIMERx_CAR}=0x60$	358
图 16-73. 通用定时器 L4 结构框图	377
图 16-74. 内部时钟分频为 1 时正常模式下的控制电路	378
图 16-75. 当预分频器的参数从 1 变到 2 时, 计数器的时序图	378
图 16-76. 向上计数时序图, $\text{PSC}=0/1$	379
图 16-77. 向上计数时序图, 在运行时改变 TIMERx_CAR 寄存器的值	380
图 16-78. 在向上计数模式下计数器重复时序图	381
图 16-79. 输入捕获逻辑	382
图 16-80. 三种输出比较模式	383
图 16-81. PWM 时序图	384
图 16-82. 带死区时间的互补输出	387

图 16-83. 通道响应中止输入（高电平有效）时，输出信号的行为	388
图 16-84. 单脉冲模式， $\text{TIMERx_CHxCV} = 0x04$ $\text{TIMERx_CAR}=0x60$	389
图 16-85. 基本定时器结构框图	403
图 16-86. 内部时钟分频为 1 时正常模式下的控制电路.....	404
图 16-87. 当预分频器的参数从 1 变到 2 时，计数器的时序图	404
图 16-88. 向上计数时序图， $\text{PSC}=0/1$	405
图 16-89. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值	406
图 17-1. IFRP 输出时序图 1	411
图 17-2. IFRP 输出时序图 2	412
图 17-3. IFRP 输出时序图 3	412
图 18-1. USART 模块内部框图	415
图 18-2. USART 字符帧 (8 数据位和 1 停止位)	415
图 18-3. USART 发送步骤	417
图 18-4. 过采样方式接收一个数据位($\text{OSB}=0$)	418
图 18-5. 采用 DMA 方式实现 USART 数据发送配置步骤.....	419
图 18-6. 采用 DMA 方式实现 USART 数据接收配置步骤.....	420
图 18-7. 两个 USART 之间的硬件流控制.....	420
图 18-8. 硬件流控制	421
图 18-9. 空闲状态下检测断开帧	422
图 18-10. 数据传输过程中检测断开帧	422
图 18-11. 同步模式下的 USART 示例	423
图 18-12. 8-bit 格式的 USART 同步通信波形($\text{CLEN}=1$).....	423
图 18-13. IrDA SIR ENDEC 模块	424
图 18-14. IrDA 数据调制	424
图 18-15. ISO7816-3 数据帧格式	425
图 18-16. USART 接收 FIFO 结构	427
图 18-17. USART 中断映射框图	429
图 19-1. I2C 模块框图	449
图 19-2. 数据有效性	450
图 19-3. 开始和停止状态	450
图 19-4. 时钟同步	451
图 19-5. SDA 线仲裁	451
图 19-6. 7 位地址的 I2C 通讯流程	452
图 19-7. 10 位地址的 I2C 通讯流程	452
图 19-8. 从机发送模式	454
图 19-9. 从机接收模式	455
图 19-10. 主机发送模式	456
图 19-11. 主机接收使用方案 A 模式	458
图 19-12. 主机接收使用方案 B 模式	460
图 20-1. SPI 结构框图	473
图 20-2. 常规模式下的 SPI 时序图	474
图 20-3. SPI 四线模式下的 SPI 时序图($\text{CKPL}=1$, $\text{CKPH}=1$, $\text{LF}=0$)	475
图 20-4. 典型的全双工模式连接	477
图 20-5. 典型的单工模式连接（主机：接收，从机：发送）	477

图 20-6. 典型的单工模式连接（主机：只发送，从机：接收）	477
图 20-7. 典型的双向线连接.....	478
图 20-8. 主机 TI 模式在不连续发送时的时序图.....	479
图 20-9. 主机 TI 模式在连续发送时的时序图	479
图 20-10. 从机 TI 模式时序图	480
图 20-11. NSS 脉冲模式时序图（主机连续发送）	481
图 20-12. SPI 四线模式四线写操作时序图.....	482
图 20-13. SPI 四线模式四线读操作时序图.....	483
图 20-14. I2S 结构框图.....	486
图 20-15. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=0, CKPL=0).....	487
图 20-16. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=0, CKPL=1).....	487
图 20-17. I2S 飞利浦标准时序图(DTLEN=10, CHLEN=1, CKPL=0).....	488
图 20-18. I2S 飞利浦标准时序图(DTLEN=10, CHLEN=1, CKPL=1).....	488
图 20-19. I2S 飞利浦标准时序图(DTLEN=01, CHLEN=1, CKPL=0).....	488
图 20-20. I2S 飞利浦标准时序图(DTLEN=01, CHLEN=1, CKPL=1).....	488
图 20-21. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=1, CKPL=0).....	488
图 20-22. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=1, CKPL=1).....	489
图 20-23. MSB 对齐标准时序图(DTLEN=00, CHLEN=0, CKPL=0).....	489
图 20-24. MSB 对齐标准时序图(DTLEN=00, CHLEN=0, CKPL=1).....	489
图 20-25. MSB 对齐标准时序图(DTLEN=10, CHLEN=1, CKPL=0).....	489
图 20-26. MSB 对齐标准时序图(DTLEN=10, CHLEN=1, CKPL=1).....	489
图 20-27. MSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=0).....	490
图 20-28. MSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=1).....	490
图 20-29. MSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=0).....	490
图 20-30. MSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=1).....	490
图 20-31. LSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=0)	490
图 20-32. LSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=1)	491
图 20-33. LSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=0)	491
图 20-34. LSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=1)	491
图 20-35. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=0).....	491
图 20-36. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=1).....	492
图 20-37. PCM 标准短帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=0).....	492
图 20-38. PCM 标准短帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=1).....	492
图 20-39. PCM 标准短帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=0).....	492
图 20-40. PCM 标准短帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=1).....	492
图 20-41. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=0).....	492
图 20-42. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=1).....	493
图 20-43. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=0).....	493
图 20-44. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=1).....	493
图 20-45. PCM 标准长帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=0).....	493
图 20-46. PCM 标准长帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=1).....	493
图 20-47. PCM 标准长帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=0).....	493
图 20-48. PCM 标准长帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=1).....	494
图 20-49. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=0).....	494

图 20-50. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=1).....	494
图 20-51. I2S 时钟生成结构框图	494
图 21-1. HDMI-CEC 控制器框图	510
图 22-1. TSI 模块框图.....	524
图 22-2. 一个通道引脚的采样引脚的框图	525
图 22-3. 电荷序列传输期间的采样引脚的电压.....	527
图 22-4. 电荷转移序列的有限状态机的状态转移图.....	527
图 23-1. USBFS 结构框图	542
图 23-2. 在主机或设备模式下连接示意图	543
图 23-3. OTG 模式下连接示意图.....	544
图 23-4. 主机端口状态转移图	544
图 23-5. 主机模式 FIFO 空间	548
图 23-6. 主机模式 FIFO 访问寄存器映射表.....	549
图 23-7. 设备模式 FIFO 空间	549
图 23-8. 设备模式 FIFO 访问寄存器映射表.....	550

表索引

表 1-1. GD32F3x0 系列器件的存储器映射表	26
表 1-2. Flash 模块组织	29
表 1-3. 引导模式	29
表 2-1. 闪存的基地址和大小	40
表 2-2. 选项字节	45
表 2-3. OB_WP 位对应页保护	46
表 3-1. 节电模式总结	59
表 4-1. 时钟源的选择	70
表 4-2. 深度睡眠模式下内核电压选择	71
表 6-1. Cortex-M4 中的 NVIC 异常类型	108
表 6-2. 中断向量表	109
表 6-3. EXTI 触发源	111
表 7-1. GPIO 配置表	116
表 9-1. DMA 传输操作	142
表 9-2. 中断事件	144
表 9-3. DMA 各通道请求表	147
表 11-1. ADC 内部信号	160
表 11-2. ADC 引脚定义	160
表 11-3. 用于 ADC 规则通道的外部触发	169
表 11-4. 用于 ADC 注入组通道的外部触发	169
表 11-5. 不同分辨率对应的 t _{CONV} 时间	171
表 11-6. N 和 M 的最大输出值（灰色部分表示截断）	172
表 12-1. DAC 引脚	187
表 12-2. DAC 外部触发	188
表 14-1. 自由看门狗定时器在 40kHz (IRC40K) 时的最小/最大超时周期	202
表 14-2. 在 54MHz (f _{PCLK1}) 时的最大/最小超时值	207
表 15-1. 省电模式管理	220
表 15-2. 中断控制	220
表 16-1. 定时器 (TIMERx) 分为六种类型	236
表 16-2. 由参数控制的互补输出表	252
表 16-3. 计数方向与编码器信号之间的关系	255
表 16-4. 从模式例子列表	258
表 16-5. 计数方向与编码器信号之间的关系	300
表 16-6. 从模式列表和举例 (通用定时器 L0)	301
表 16-7. TIMERx(x=1,2) 定时器内部互连	304
表 16-8. 由参数控制的互补输出表	353
表 16-9. 从模式例子列表	356
表 16-10. TIMERx(x=14) 定时器内部互连	359
表 16-11. 由参数控制的互补输出表	385
表 18-1. USART 重要引脚描述	414

表 18-2. 停止位配置	416
表 18-3. USART 中断请求	428
表 19-1. I2C 总线术语说明（参考飞利浦 I2C 规范）	449
表 19-2. 事件状态标志位	463
表 19-3. I2C 错误标志位	463
表 20-1. SPI 信号描述	473
表 20-2. SPI 四线信号描述	474
表 20-3. SPI 运行模式	476
表 20-4. SPI 中断请求	486
表 20-5. I2S 比特率计算公式	494
表 20-6. 音频采样频率计算公式	495
表 20-7. 各种运行模式下 I2S 接口信号的方向	495
表 20-8. I2S 中断	498
表 21-1. 数据位时序参数表	511
表 21-2. 错误处理时序参数表	514
表 21-3. 时序参数表	515
表 22-1. 电荷转移序列的详细步骤以及引脚和开关状态	525
表 22-2. 充电扩展状态的持续时间	528
表 22-3 基于 HCLK 周期的扩频偏差	529
表 22-4. TSI 错误和标志位	530
表 22-5. TSI 引脚	530
表 23-1. USBFS 信号线描述	542
表 23-2. USBFS 全局中断	553
表 24-1. 版本历史	610

1. 系统及存储器架构

GD32F3x0系列器件是基于ARM® Cortex™-M4处理器的32位通用微控制器。ARM® Cortex™-M4处理器包括三条AHB总线分别称为I-CODE总线、D-Code总线和系统总线。Cortex™-M4处理器的所有存储访问，根据不同的目的和目标存储空间，都会在这三条总线上执行。存储器的组织采用了哈佛结构，预先定义的存储器映射和高达4 GB的存储空间，充分保证了系统的灵活性和可扩展性。

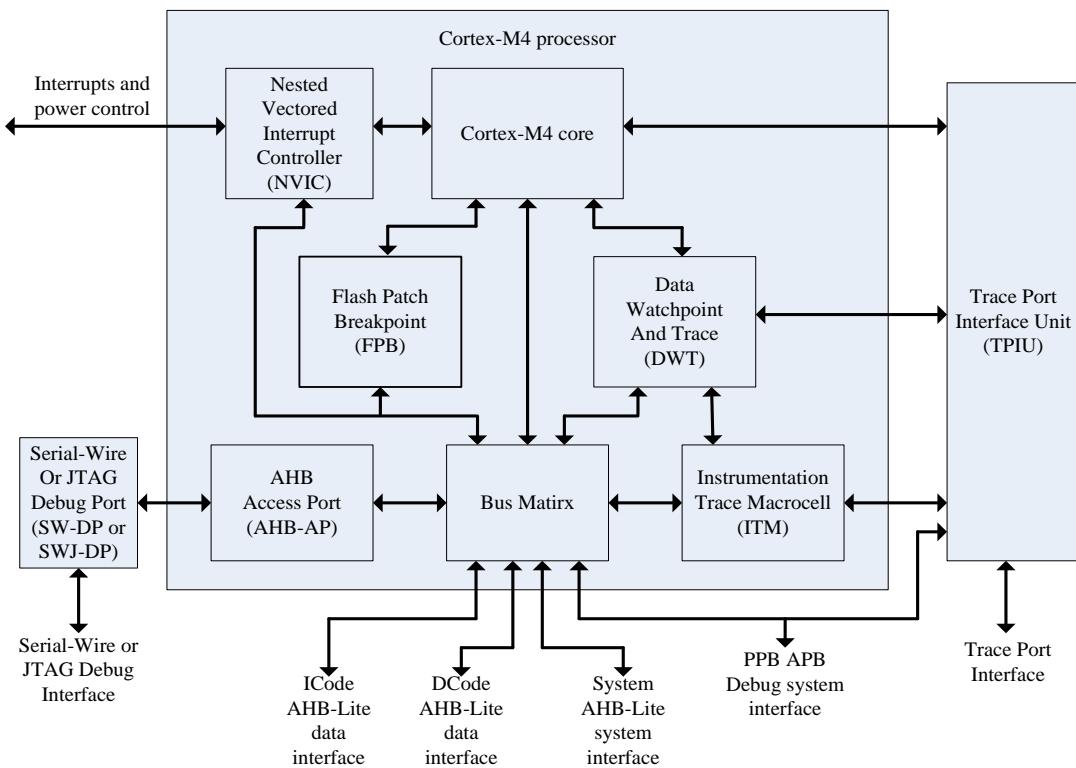
1.1. ARM Cortex-M4 处理器

Cortex™-M4处理器是一个具有浮点运算功能、低中断延迟时间和低成本调试特性的32位处理器。高集成度和增强的特性使Cortex™-M4处理器适合于那些需要高性能和低功耗微控制器的市场领域。Cortex™-M4处理器基于ARMv7架构，并且支持一个强大且可扩展的指令集，包括通用数据处理I/O控制任务、增强的数据处理位域操作。下面列出由Cortex™-M4提供的一些系统外设：

- 内部总线矩阵，用于实现I-Code总线、D-Code总线、系统总线、专用总线(PPB)以及调试专用总线(AHB-AP)的互联；
- 嵌套式向量型中断控制器 (NVIC)；
- 闪存地址重载及断点单元 (FPB)；
- 数据观测点及跟踪单元 (DWT)；
- 指令跟踪宏单元 (ITM)；
- 串行线和JTAG调试接口 (SWJ-DP)；
- 跟踪端口接口单元 (TPIU)；

下图显示了 Cortex™-M4 处理器结构框图。欲了解更多信息，请参阅 ARM® Cortex™-M4 技术参考手册。

图 1-1. Cortex™-M4 处理器结构框图

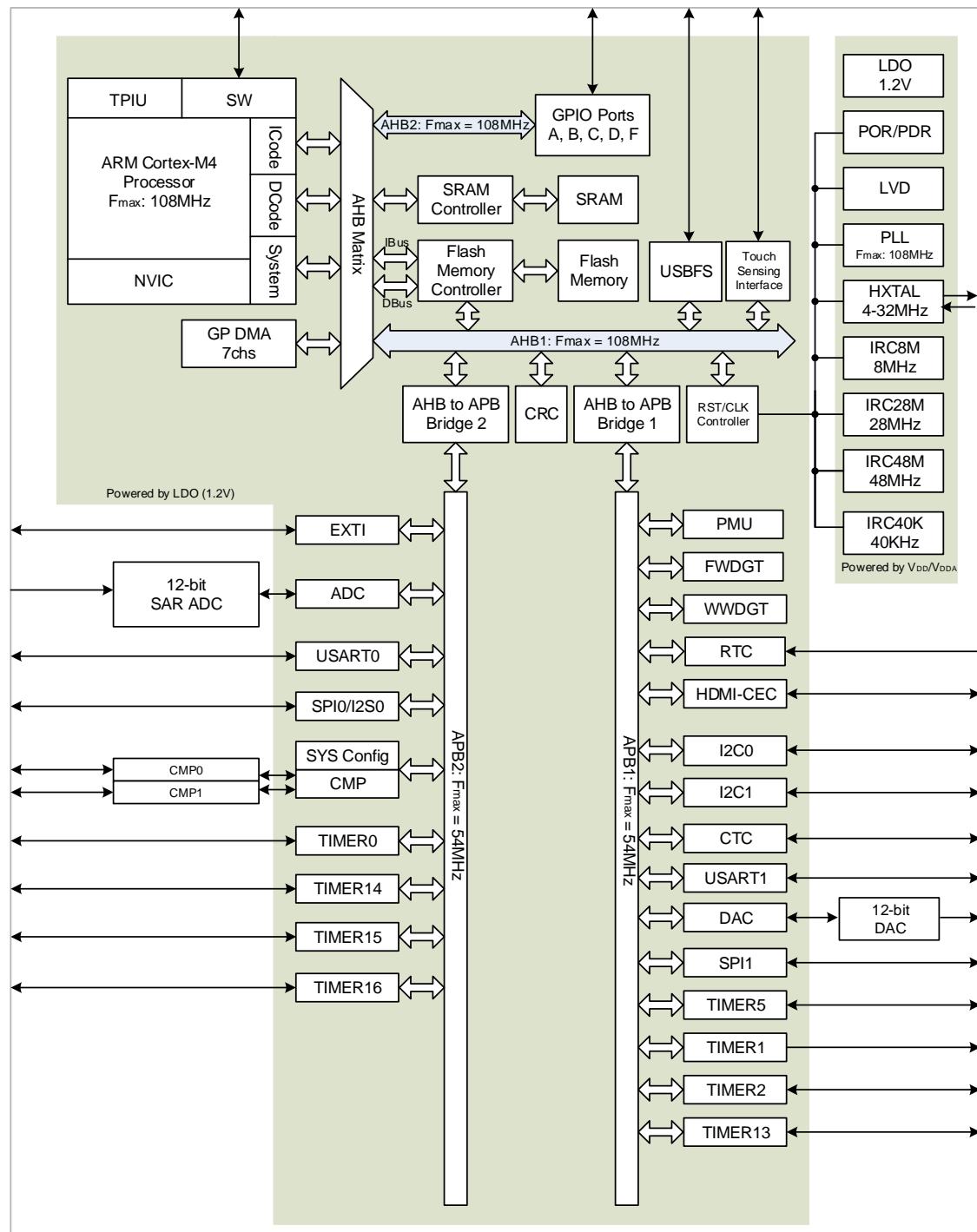


1.2. 系统架构

GD32F3x0 系列器件的系统架构如下图所示。该 AHB 矩阵是一个基于 AMBA 3.0 AHB-LITE 的多层总线，这个结构使得系统中的多个主机和从机之间的并行通信成为可能。该 AHB 矩阵中包含属于 Cortex™-M4 内核的 I-Code 总线、D-Code 总线和系统总线，以及内核外的 DMA 共 4 个主机。I-Code 总线是 Cortex™-M4 内核的指令总线，用于从代码区域(0x0000 0000~0x1FFF FFFF)中获取向量。D-Code 总线用于加载和存储数据，以及代码区域的调试访问。类似的，系统总线用于指令和向量获取、数据加载和存储以及系统区域的调试访问。系统区域包括内部 SRAM 区域和外设区域。该 AHB 矩阵还连接了 5 个从机，分别为：FMC 的 I-Code、FMC 的 D-Code、内部 SRAM、AHB1 和 AHB2。

AHB2 连接 GPIO 端口。AHB1 连接 AHB 外设，包括 2 个 AHB-APB 总线桥。AHB-APB 总线桥提供了 AHB1 和两条 APB 总线之间的全同步连接。两条 APB 总线连接了所有的 APB 外设。

图 1-2. GD32F3x0 系列器件的系统架构示意图



1.3. 存储器映射

ARM® Cortex™-M4 处理器采用哈佛结构，可以使用相互独立的总线来读取指令和加载/存储数据。指令代码和数据都位于相同的存储器地址空间，但在不同的地址范围。程序存储器，数据存储器，寄存器和 I/O 端口都在同一个线性的 4 GB 的地址空间之内。这是 Cortex™-M4 的

最大地址范围，因为它的地址总线宽度是 32 位。此外，为了降低不同客户在相同应用时的软件复杂度，存储映射是按 Cortex™-M4 处理器提供的规则预先定义的。同时，一部分地址空间由 ARM® Cortex™-M4 的系统外设所占用。下表显示了 GD32F3x0 系列器件的存储器映射，包括代码、SRAM、外设和其他预先定义的区域。几乎每个外设都分配了 1KB 的地址空间，这样可以简化每个外设的地址译码。

表 1-1. GD32F3x0 系列器件的存储器映射表

预先定义的地址空间	总线	地址范围	外设
		0xE000 0000 - 0xE00F FFFF	Cortex M4 内部外设
外部设备		0xA000 0000 - 0xDFFF FFFF	保留
外部 RAM		0x6000 0000 - 0x9FFF FFFF	保留
外设	AHB1	0x5004 0000 - 0x5FFF FFFF	保留
		0x5000 0000 - 0x5003 FFFF	USBFS
	AHB2	0x4800 1800 - 0x4FFF FFFF	保留
		0x4800 1400 - 0x4800 17FF	GPIOF
		0x4800 1000 - 0x4800 13FF	保留
		0x4800 0C00 - 0x4800 0FFF	GPIOD
		0x4800 0800 - 0x4800 0BFF	GPIOC
		0x4800 0400 - 0x4800 07FF	GPIOB
		0x4800 0000 - 0x4800 03FF	GPIOA
	AHB1	0x4002 4400 - 0x47FF FFFF	保留
		0x4002 4000 - 0x4002 43FF	TSI
		0x4002 3400 - 0x4002 3FFF	保留
		0x4002 3000 - 0x4002 33FF	CRC
		0x4002 2400 - 0x4002 2FFF	保留
		0x4002 2000 - 0x4002 23FF	FMC
		0x4002 1400 - 0x4002 1FFF	保留
		0x4002 1000 - 0x4002 13FF	RCU
		0x4002 0400 - 0x4002 0FFF	保留
		0x4002 0000 - 0x4002 03FF	DMA
	APB2	0x4001 8000 - 0x4001 FFFF	保留
		0x4001 5C00 - 0x4001 7FFF	保留
		0x4001 5800 - 0x4001 5BFF	保留
		0x4001 4C00 - 0x4001 57FF	保留
		0x4001 4800 - 0x4001 4BFF	TIMER16
		0x4001 4400 - 0x4001 47FF	TIMER15
		0x4001 4000 - 0x4001 43FF	TIMER14
		0x4001 3C00 - 0x4001 3FFF	保留
		0x4001 3800 - 0x4001 3BFF	USART0
		0x4001 3400 - 0x4001 37FF	保留
		0x4001 3000 - 0x4001 33FF	SPI0/I2S0
		0x4001 2C00 - 0x4001 2FFF	TIMER0

预先定义的地址空间	总线	地址范围	外设
APB1		0x4001 2800 - 0x4001 2BFF	保留
		0x4001 2400 - 0x4001 27FF	ADC
		0x4001 0800 - 0x4001 23FF	保留
		0x4001 0400 - 0x4001 07FF	EXTI
		0x4001 0000 - 0x4001 03FF	SYSCFG + CMP
		0x4000 CC00 - 0x4000 FFFF	保留
		0x4000 C800 - 0x4000 CBFF	CTC
		0x4000 C400 - 0x4000 C7FF	保留
		0x4000 C000 - 0x4000 C3FF	保留
		0x4000 8000 - 0x4000 BFFF	保留
		0x4000 7C00 - 0x4000 7FFF	保留
		0x4000 7800 - 0x4000 7BFF	CEC
		0x4000 7400 - 0x4000 77FF	DAC
		0x4000 7000 - 0x4000 73FF	PMU
		0x4000 6400 - 0x4000 6FFF	保留
		0x4000 6000 - 0x4000 63FF	保留
		0x4000 5C00 - 0x4000 5FFF	保留
		0x4000 5800 - 0x4000 5BFF	I2C1
		0x4000 5400 - 0x4000 57FF	I2C0
		0x4000 4800 - 0x4000 53FF	保留
		0x4000 4400 - 0x4000 47FF	USART1
		0x4000 4000 - 0x4000 43FF	保留
		0x4000 3C00 - 0x4000 3FFF	保留
		0x4000 3800 - 0x4000 3BFF	SPI1
		0x4000 3400 - 0x4000 37FF	保留
		0x4000 3000 - 0x4000 33FF	FWDGT
		0x4000 2C00 - 0x4000 2FFF	WWDGT
		0x4000 2800 - 0x4000 2BFF	RTC
		0x4000 2400 - 0x4000 27FF	保留
		0x4000 2000 - 0x4000 23FF	TIMER13
		0x4000 1400 - 0x4000 1FFF	保留
		0x4000 1000 - 0x4000 13FF	TIMER5
		0x4000 0800 - 0x4000 0FFF	保留
		0x4000 0400 - 0x4000 07FF	TIMER2
		0x4000 0000 - 0x4000 03FF	TIMER1
SRAM		0x2000 5000 - 0x3FFF FFFF	保留
		0x2000 0000 - 0x2000 4FFF	SRAM
代码		0x1FFF FC00 - 0x1FFF FFFF	保留
		0x1FFF F800 - 0x1FFF FBFF	Option bytes
		0x1FFF EC00 - 0x1FFF F7FF	System memory

预先定义的地址空间	总线	地址范围	外设
		0x0810 0000 - 0x1FFF EBFF	保留
		0x0800 0000 - 0x080F FFFF	Main Flash memory
		0x0010 0000 - 0x07FF FFFF	保留
		0x0000 0000 - 0x000F FFFF	Aliased to Flash or system memory

1.3.1. 位带操作

为了减少“读-改-写”操作的次数，Cortex™-M4处理器提供了一个可以执行单原子比特操作的位带功能。存储器映射包含了两个支持位带操作的区域，分别位于**SRAM**和外设中。位带区域将存储器别名区的每个字映射到存储器位带区的某个位上。

下面的映射公式表明了别名区中的每个字如何对应位带区的相应比特或目标比特。

$$\text{bit_word_addr} = \text{bit_band_base} + (\text{byte_offset} \times 32) + (\text{bit_number} \times 4) \quad (\text{式1-1})$$

其中：

- `bit_word_addr`指的是映射到位带区目标比特的别名存储器区字地址；
- `bit_band_base`指的是别名区的起始地址；
- `byte_offset`指的是位带区目标比特所在的字节的字节地址偏移量；
- `bit_number`指的是目标比特在对应字节中的位置(0-7)。

例如，要想访问0x2000 0200地址的第7位，可访问的位带别名区地址是：

$$\text{bit_word_addr} = 0x2200 0000 + (0x200 * 32) + (7 * 4) = 0x2200 401C \quad (\text{式1-2})$$

如果对0x2200 401C进行写操作，那么0x2000 0200的第7位将会相应变化；如果对0x2200 401C进行读操作，那么视0x2000 0200的第7位状态而返回0x01或0x00。

1.3.2. 片上 **SRAM** 存储器

GD32F3x0系列微控制器含有高达16KB的片上**SRAM**，起始地址为0x2000 0000，支持字节、半字(16比特)和整字(32比特)访问。存储器支持奇偶校验来提高鲁棒性。用户可以通过用户选项字节(请参考第2.3.9章节 [选项字节说明](#))的**SRAM_PARITY_CHECK**位来启用奇偶校验功能。当启用时，如果校验失败，产生一个NMI中断。**SRAM**奇偶校验错误标志在系统配置寄存器2(SYSCFG_CFG2)之中。如果系统配置寄存器2(SYSCFG_CFG2)的**SRAM_PARITY_ERROR_LOCK**位置1，错误标志将被连接到定时器0/定时器14/定时器15/定时器16的break输入端。

SRAM的真实数据宽度为36位，包括32位数据和4位奇偶校验(每字节1位)位。在写入时，奇偶校验位被计算并存储到**SRAM**。当读取时，奇偶校验位会用**SRAM**读出的数据再计算一遍。计算出的奇偶校验位将与读出的奇偶校验位(写入访问期间计算并存储的奇偶校验位)进行比较。如果它们不相同，则奇偶校验失败。

注意：如果启用了SRAM奇偶校验，建议通过软件在代码的开始初始化整个SRAM存储器，以避免读取未初始化的位置时，得到的奇偶校验错误。

1.3.3. 片上闪存

该系列微控制器提供高达128KB的片上闪存。片上闪存包括高达128KB的主存储块和3KB容量的用于存储引导装载程序(boot loader)的信息块。主存储块分为128页，每页的容量为1KB。下表显示详细信息。

表 1-2. Flash 模块组织

闪存块	名称	地址范围	大小
主存储块	Page 0	0x0800 0000 - 0x0800 03FF	1 Kbytes
	Page 1	0x0800 0400 - 0x0800 07FF	1 Kbytes
	Page 2	0x0800 0800 - 0x0800 0BFF	1 Kbytes
	.	.	.
	.	.	.
	Page 127	0x0801 FC00 - 0x0801 FFFF	1 Kbytes
信息块	系统存储器	0x1FFF EC00 - 0x1FFF F7FF	3 Kbytes
	选项字节	0x1FFF F800 - 0x1FFF F80F	16 bytes

针对前32页的读访问可实现每个周期无任何等待状态读出32比特数据的效率。读访问支持字节、半字(16比特)和整字(32比特)；写访问（编程）只支持半字(16比特)和整字(32比特)。片上闪存的每一页都可以单独被擦除，整个主存模块也可以同时被擦除。

1.4. 引导配置

GD32F3x0 系列微控制器提供了三种引导源，可以通过用户选项字节 BOOT1_n 位(请参考第 2.3.9 章节 [选项字节说明](#))和 BOOT0 引脚进行选择。BOOT0 引脚的电平值是在复位后第 4 个系统时钟上升沿锁存的。用户可自行选择所需要的引导源，通过设置上电复位或系统复位后的 BOOT1_n 位和 BOOT0 的引脚电平值。下表描述了详细的引导模式信息。

表 1-3. 引导模式

引导源选择	启动模式选择引脚	
	Boot1	Boot0
主FLASH存储器	x	0
系统存储器	0	1
片上SRAM	1	1

1. Boot1值与BOOT1_n值相反

上电序列或系统复位后，ARM® Cortex™-M4处理器先从0x0000 0000地址获取栈顶值，再从0x0000 0004地址获得引导代码的基址，然后从引导代码的基址开始执行程序。

根据所选的引导源，片上闪存的主存（开始于0x0800 0000的存储空间）或系统存储器（开始于0x1FFF EC00的存储空间）会被映射到引导空间，即从0x0000 0000开始的地址空间。如果

片上SRAM（开始于0x2000 0000的存储空间）被选为引导源，用户必须在应用程序初始化代码中通过修改NVIC异常向量表和偏移寄存器将向量表重置到SRAM中。

芯片内嵌的引导装载程序位于系统存储器中，用来对片上闪存的主存进行重编程。该引导装载程序可通过以下串行接口之一工作：USART0或USART1。

1.5. I/O 补偿单元

通常情况下是不使用I/O补偿单元的，但是，当I/O端口输出速度大于50MHz时，建议使用I/O补偿单元对I/O端口进行斜率控制，从而降低I/O端口噪声对电源的影响。

在使能I/O补偿单元后，SYSCFG_CPSCTL寄存器中的准备完成标志位CPS_RDY将会被置位，用于指示补偿单元已经准备好，可以使用。

1.6. 系统配置寄存器 (SYSCFG)

1.6.1. 配置寄存器 0 (SYSCFG_CFG0)

地址偏移：0x00

复位值：0x0000 000X (根据BOOT0引脚的状态和用户选项字节的BOOT1_n的值，X表示BOOT_MODE[1:0]可能为任意值)

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留													PB9_HCCE	保留	
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TIMER16_DMA_RMP	TIMER15_DMA_RMP	USART0_RX_DMA_RMP	USART0_TX_DMA_RMP	ADC_DMA_RMP	保留					BOOT_MODE[1:0]				
	rw	rw	rw	rw	rw										

位/位域	名称	描述
31:20	保留	必须保持复位值
19	PB9_HCCE	PB9引脚大电流能力使能 当该位为1时，PB9引脚可以直接用来控制红外发光二极管。 0: PB9引脚大电流能力关闭 1: PB9引脚大电流能力开启，同时该引脚的速度控制被忽略
18:13	保留	必须保持复位值
12	TIMER16_DMA_RMP	TIMER 16 DMA请求重映射使能 0: 不重映射 (TIMER16_CH0和TIMER16_UP DMA被映射在DMA通道0) 1: 重映射 (TIMER16_CH0和TIMER16_UP DMA被映射在DMA通道1)

11	TIMER15_DMA_RMP	TIMER 15 DMA请求重映射使能 0: 不重映射 (TIMER15_CH0和TIMER15_UP DMA被映射在DMA通道2) 1: 重映射 (TIMER15_CH0和TIMER15_UP DMA被映射在DMA通道3)
10	USART0_RX_DMA_RMP	USART0_RX DMA请求重映射使能 0: 不重映射 (USART0_RX DMA被映射在DMA通道2) 1: 重映射 (USART0_RX DMA被映射在DMA通道4)
9	USART0_TX_DMA_RMP	USART0_TX DMA请求重映射使能 0: 不重映射 (USART0_TX DMA被映射在DMA通道1) 1: 重映射 (USART0_TX DMA被映射在DMA通道3)
8	ADC_DMA_RMP	ADC DMA请求重映射使能 0: 不重映射 (ADC DMA被映射在DMA通道0) 1: 重映射 (ADC DMA被映射在DMA通道1)
7:2	保留	必须保持复位值
1:0	BOOT_MODE[1:0]	引导模式(详细请参考第1.4章节 引导配置) bit0 映射到BOOT0引脚; bit1 的值与BOOT1_n的值相反 x0: 从片上闪存的主存引导启动 01: 从片上闪存的系统存储器引导启动 11: 从片上SRAM引导启动

1.6.2. EXTI 源选择寄存器 0 (SYSCFG_EXTISS0)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI3_SS [3:0]				EXTI2_SS [3:0]				EXTI1_SS [3:0]				EXTI0_SS [3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:16	保留	必须保持复位值
15:12	EXTI3_SS[3:0]	EXTI 3源选择 X000: PA3引脚 X001: PB3引脚 X010: PC3引脚 X011: 保留 X100: 保留

		X101: 保留
		X110: 保留
		X111: 保留
11:8	EXTI2_SS[3:0]	EXTI 2源选择
		X000: PA2引脚
		X001: PB2引脚
		X010: PC2引脚
		X011: PD2引脚
		X100: 保留
		X101: 保留
		X110: 保留
		X111: 保留
7:4	EXTI1_SS[3:0]	EXTI 1源选择
		X000: PA1引脚
		X001: PB1引脚
		X010: PC1引脚
		X011: 保留
		X100: 保留
		X101: PF1引脚
		X110: 保留
		X111: 保留
3:0	EXTI0_SS[3:0]	EXTI 0 源选择
		X000: PA0引脚
		X001: PB0引脚
		X010: PC0引脚
		X011: 保留
		X100: 保留
		X101: PF0引脚
		X110: 保留
		X111: 保留

1.6.3. EXTI 源选择寄存器 1 (SYSCFG_EXTISS1)

地址偏移: 0x0C

复位值: 0x0000 0000

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI7_SS [3:0]				EXTI6_SS [3:0]				EXTI5_SS [3:0]				EXTI4_SS [3:0]			

rw

rw

rw

rw

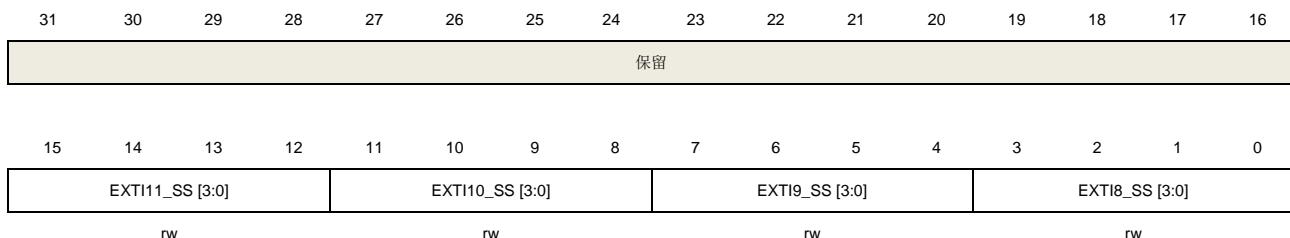
位/位域	名称	描述
31:16	保留	必须保持复位值
15:12	EXTI7_SS[3:0]	EXTI 7源选择 X000: PA7引脚 X001: PB7引脚 X010: PC7引脚 X011: 保留 X100: 保留 X101: PF7引脚 X110: 保留 X111: 保留
11:8	EXTI6_SS[3:0]	EXTI 6源选择 X000: PA6引脚 X001: PB6引脚 X010: PC6引脚 X011: 保留 X100: 保留 X101: PF6引脚 X110: 保留 X111: 保留
7:4	EXTI5_SS[3:0]	EXTI 5源选择 X000: PA5引脚 X001: PB5引脚 X010: PC5引脚 X011: 保留 X100: 保留 X101: PF5引脚 X110: 保留 X111: 保留
3:0	EXTI4_SS[3:0]	EXTI 4源选择 X000: PA4引脚 X001: PB4引脚 X010: PC4引脚 X011: 保留 X100: 保留 X101: PF4引脚 X110: 保留 X111: 保留

1.6.4. EXTI 源选择寄存器 2 (SYSCFG_EXTISS2)

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器只能按字（32位）访问



位/位域	名称	描述
31:16	保留	必须保持复位值
15:12	EXTI11_SS[3:0]	EXTI 11源选择 X000: PA11引脚 X001: PB11引脚 X010: PC11引脚 X011: 保留 X100: 保留 X101: 保留 X110: 保留 X111: 保留
11:8	EXTI10_SS[3:0]	EXTI 10源选择 X000: PA10引脚 X001: PB10引脚 X010: PC10引脚 X011: 保留 X100: 保留 X101: 保留 X110: 保留 X111: 保留
7:4	EXTI9_SS[3:0]	EXTI 9源选择 X000: PA9引脚 X001: PB9引脚 X010: PC9引脚 X011: 保留 X100: 保留 X101: 保留 X110: 保留

X111: 保留

3:0	EXTI8_SS[3:0]	EXTI 8源选择
		X000: PA8引脚
		X001: PB8引脚
		X010: PC8引脚
		X011: 保留
		X100: 保留
		X101: 保留
		X110: 保留
		X111: 保留

1.6.5. EXTI 源选择寄存器 3 (SYSCFG_EXTISS3)

地址偏移: 0x14

复位值: 0x0000 0000

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI15_SS [3:0]				EXTI14_SS [3:0]				EXTI13_SS [3:0]				EXTI12_SS [3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:16	保留	必须保持复位值
15:12	EXTI15_SS[3:0]	EXTI 15源选择 X000: PA15引脚 X001: PB15引脚 X010: PC15引脚 X011: 保留 X100: 保留 X101: 保留 X110: 保留 X111: 保留
11:8	EXTI14_SS[3:0]	EXTI 14源选择 X000: PA14引脚 X001: PB14引脚 X010: PC14引脚 X011: 保留 X100: 保留

		X101: 保留
		X110: 保留
		X111: 保留
7:4	EXTI13_SS[3:0]	EXTI 13源选择
		X000: PA13引脚
		X001: PB13引脚
		X010: PC13引脚
		X011: 保留
		X100: 保留
		X101: 保留
		X110: 保留
		X111: 保留
3:0	EXTI12_SS[3:0]	EXTI 12源选择
		X000: PA12引脚
		X001: PB12引脚
		X010: PC12引脚
		X011: 保留
		X100: 保留
		X101: 保留
		X110: 保留
		X111: 保留

1.6.6. 系统配置寄存器 2 (SYSCFG_CFG2)

地址偏移: 0x18

复位值: 0x0000 0000

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					SRAM_P CEF	保留					LVD_ LOCK	SRAM_ PARITY_ ERROR_ LOCK	LOCK UP_ LOCK		

位/位域	名称	描述
31:9	保留	必须保持复位值
8	SRAM_PCEF	<p>SRAM奇偶校验错误标志</p> <p>当SRAM奇偶校验错误发生时，该位由硬件置1。该位由软件写1清零。</p> <p>0: 未检测到SRAM奇偶校验错误</p>

1: 检测到SRAM奇偶校验错误

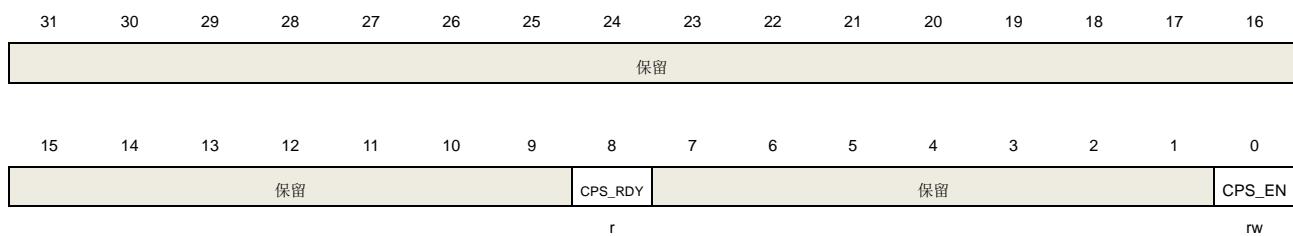
7:3	保留	必须保持复位值
2	LVD_LOCK	<p>LVD锁定</p> <p>该位由软件置1，在系统复位时才能清零。</p> <p>0: LVD中断从TIMER0/14/15/16的break输入端断开。PMU_CTL寄存器的LVDEN和LVDT[2:0]可以被设置</p> <p>1: LVD中断与TIMER0/14/15/16的break输入端连接。PMU_CTL寄存器的LVDEN和LVDT[2:0]仅仅可读</p>
1	SRAM_PARITY_ERROR_LOCK	<p>SRAM奇偶校验错误锁定</p> <p>该位由软件置1，在系统复位时才能清零。</p> <p>0: SRAM奇偶校验错误从TIMER0/14/15/16的break输入端断开</p> <p>1: SRAM奇偶校验错误与TIMER0/14/15/16的break输入端连接</p>
0	LOCKUP_LOCK	<p>Cortex-M4 LOCKUP输出锁定</p> <p>该位由软件置1，在系统复位时才能清零。</p> <p>0: Cortex-M4 LOCKUP输出从TIMER0/14/15/16的break输入端断开</p> <p>1: Cortex-M4 LOCKUP输出与TIMER0/14/15/16的break输入端连接</p>

1.6.7. I/O 补偿控制寄存器 (SYSCFG_CPSCTL)

地址偏移: 0x20

复位值: 0x0000 0000

该寄存器只能按字（32位）访问



位/位域	名称	描述
31:9	保留	必须保持复位值
8	CPS_RDY	<p>I/O补偿单元是否准备好</p> <p>该位只读</p> <p>0: I/O补偿单元没有准备好</p> <p>1: I/O补偿单元准备好</p>
7:1	保留	必须保持复位值
0	CPS_EN	<p>I/O补偿单元使能</p> <p>0: I/O补偿单元掉电</p>

1: I/O补偿单元使能

1.7. 设备电子签名

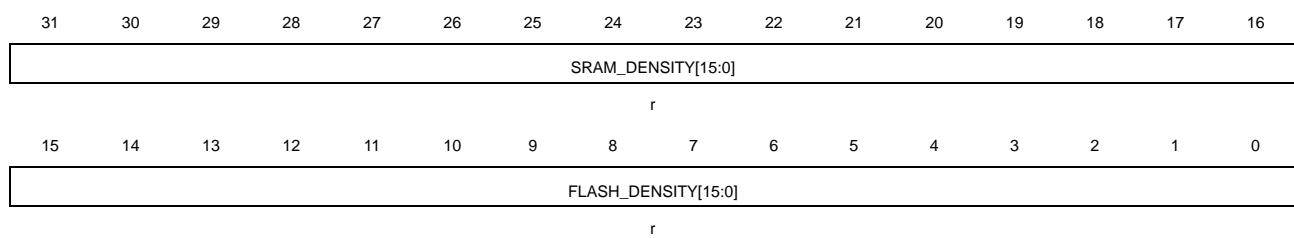
设备的电子签名中包含的存储容量信息和96位的唯一设备ID。它被存储在片上闪存的信息模块中。96位唯一设备ID对于每颗芯片而言都是唯一的。它可以用作序列号，或安全密钥的一部分，等等。

1.7.1. 存储容量信息

基地址: 0xFFFF F7E0

该值是原厂设定的，不能由用户修改。

该寄存器只能按字（32位）访问



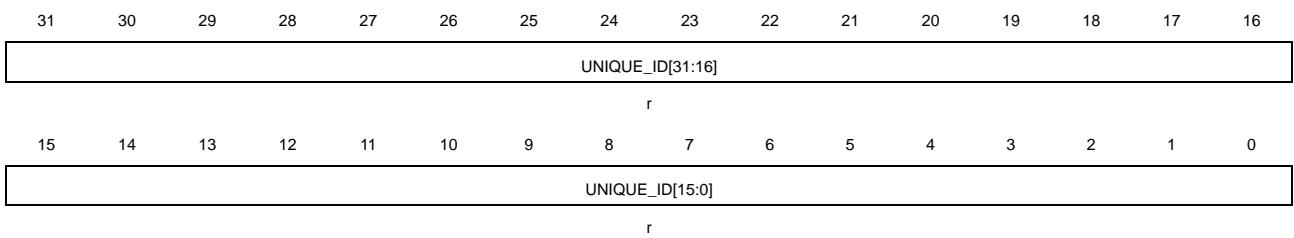
位/位域	名称	描述
31:16	SRAM_DENSITY	SRAM存储器容量
[15:0]		该值表明芯片的片上SRAM存储器容量，以Kbytes为单位 例如：0x0008表示8Kbytes。
15:0	FLASH_DENSITY	Flash存储器容量
[15:0]		该值表明芯片的片上Flash容量，以Kbytes为单位 例如：0x0020表示32Kbytes。

1.7.2. 设备唯一 ID (96 位/位域)

基地址: 0xFFFF F7AC

该值是原厂设定的，不能由用户修改。

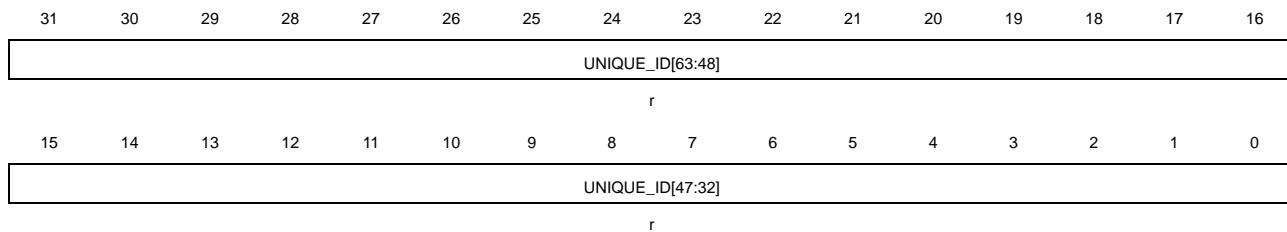
该寄存器只能按字（32位）访问



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

基地址: 0x1FFF F7B0
 该值是原厂设定的, 不能由用户修改。

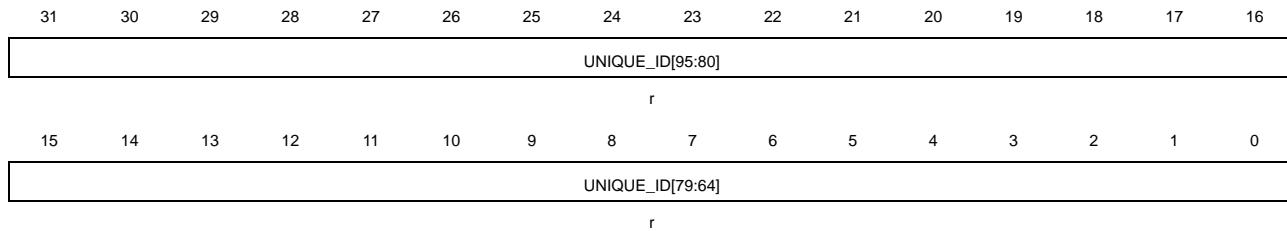
该寄存器只能按字 (32位) 访问



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

基地址: 0x1FFF F7B4
 该值是原厂设定的, 不能由用户修改。

该寄存器只能按字 (32位) 访问



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

2. 闪存控制器 (FMC)

2.1. 简介

闪存控制器 (FMC)，提供了片上闪存需要的所有功能。在闪存的前64K字节空间内，CPU执行指令零等待。FMC也提供了页擦除、整片擦除以及32位整字或16位半字编程等操作。

2.2. 主要特性

- 高达128KB字节的片上闪存可用于存储指令或数据；
- 在闪存的前64K字节空间内，CPU执行指令零等待；
- 从闪存的64K ~ 128K地址空间内取数据有比较长的延迟；
- 3K字节引导装载程序信息块；
- 16字节的选项字节块用于用户需求；
- 每页大小为1K字节；
- 32位字或16位半字编程，支持页擦除和整片擦除；
- 有闪存读保护功能，阻止非法代码或数据进入；
- 有页擦除和页编程保护功能，阻止意外操作；

2.3. 功能描述

2.3.1. 闪存结构

闪存存储器包括一个高达128K字节的主闪存(按128页每页1K字节分块)和一个3K字节的用于引导装载程序的信息块。主闪存存储器的128页每页都可以单独擦除。闪存存储器的基址和大小见下表。

表 2-1. 闪存的基址和大小

闪存块	名称	地址范围	大小(字节)
主闪存块	Page 0	0x0800 0000 - 0x0800 03FF	1KB
	Page 1	0x0800 0400 - 0x0800 07FF	1KB
	Page 2	0x0800 0800 - 0x0800 0BFF	1KB
	.	.	.
	.	.	.
	.	.	.
	Page 127	0x0801 FC00 - 0x0801 FFFF	1KB
信息块	引导装载程序	0x1FFF EC00 - 0x1FFF F7FF	3KB
选项字节块	选项字节	0x1FFF F800 - 0x1FFF F80F	16B

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

2.3.2. 读操作

闪存可以像普通存储空间一样直接寻址访问。对闪存取指令和取数据分别使用CPU的IBUS或DBUS总线。

2.3.3. FMC_CTL 寄存器解锁

复位后，FMC_CTL寄存器不可以用写模式进行访问（OBRLD位除外，此位用于重加载选项字节）并且FMC_CTL寄存器中的LK位被置为1。一个包含对FMC_KEY寄存器进行两次写操作的解锁序列可以解锁FMC_CTL寄存器的访问，分别是先后写入0x45670123和0xCDEF89AB。两次写操作后，FMC_CTL寄存器的LK位将被硬件清0。可以通过软件设置FMC_CTL寄存器的LK位为1再次锁定FMC_CTL寄存器。任何对FMC_KEY寄存器的错误操作都会将LK位置1，从而锁定FMC_CTL寄存器，并且引发一个总线错误。

FMC_CTL寄存器的OBPG位和OBER位可以被FMC_OBKEY寄存器锁定。解锁序列是向FMC_OBKEY寄存器先后写入0x45670123和0xCDEF89AB，然后将FMC_CTL寄存器的OBWEN位置1。软件可以将FMC_CTL的OBWEN位清0来锁定FMC_CTL的OBPG位和OBER位。

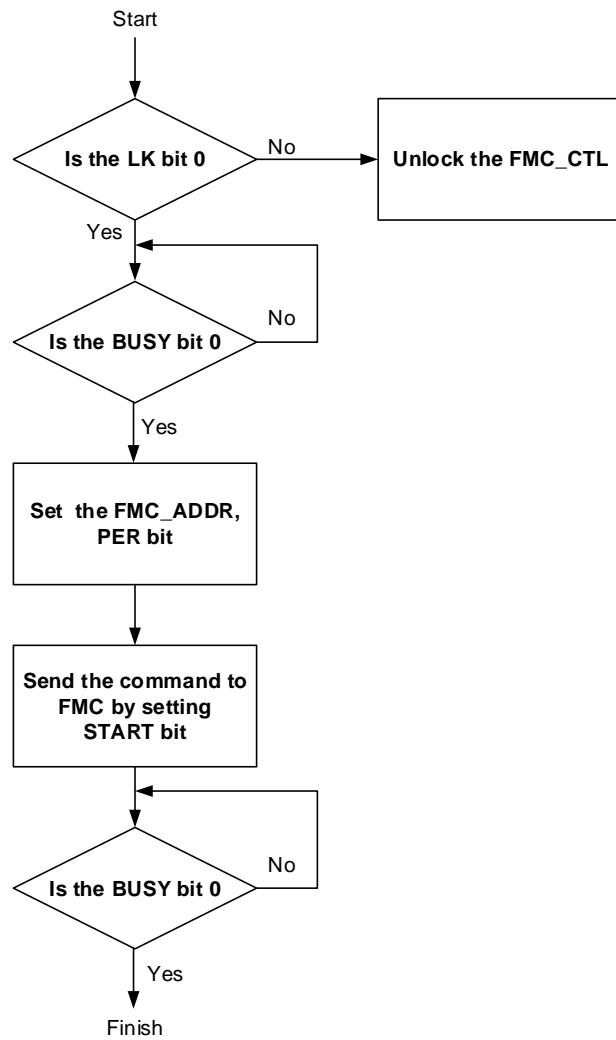
2.3.4. 页擦除

FMC的页擦除功能将一个主闪存存储页的内容初始化为高电平。每一页都可以被独立擦除，不影响其他页的内容。FMC页擦除的操作步骤如下：

- 确保FMC_CTL寄存器不处于锁定状态；
- 检查FMC_STAT寄存器的BUSY位来确保闪存存储器没有正在进行中的操作，即BUSY位为0。否则等待该操作完成；
- 写页地址到FMC_ADDR寄存器；
- 写页擦除命令到FMC_CTL寄存器的PER位；
- 通过将FMC_CTL寄存器的START位置1来发送页擦除命令到FMC；
- 通过检查FMC_STAT寄存器的BUSY位来判断擦除指令是否执行完毕，若未完成则需等待BUSY位为0；
- 如果有需要，可以使用DBUS读和验证该页的内容。

当页擦除成功执行，FMC_STAT寄存器的ENDF位将会被置1，并且如果FMC_CTL寄存器的ENDIE位之前已经被置1，那么FMC将触发一个中断。需要注意的是，一定要确保目标页地址的正确性。否则当错误的目标擦除页被用来取指令或访问数据时，软件可能失去控制。另一方面，在擦除/编程保护的页进行页擦除操作将会无效。如果FMC_CTL寄存器的ERRIE位被置1，FMC将触发一次闪存操作错误中断。软件可以检查FMC_STAT寄存器的PGERR位来监测中断处理器的状况。FMC_STAT寄存器的ENDF位指示该操作的结束。下图显示了页擦除操作的流程：

图 2-1. 页擦除操作流程



2.3.5. 整片擦除

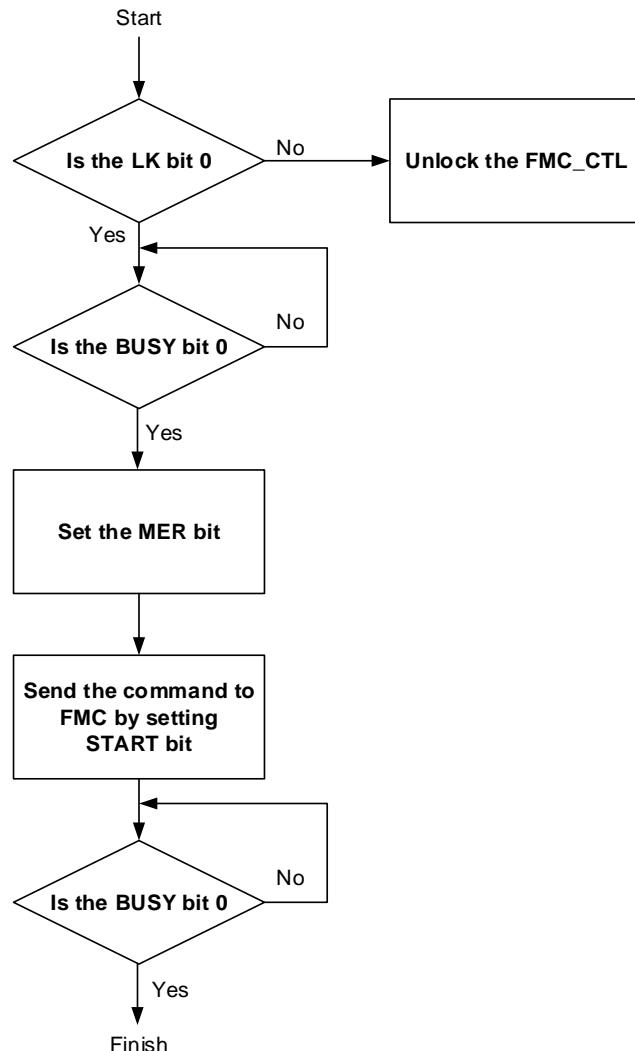
FMC提供了整片擦除功能可以初始化主闪存块的内容。整片擦除操作的寄存器设置具体步骤如下：

- 确保FMC_CTL寄存器不处于锁定状态；
- 检查FMC_STAT寄存器的BUSY位来确保闪存存储器没有正在进行中的操作，即BUSY位为0。否则等待该操作完成；
- 写整片擦除命令到FMC_CTL寄存器的MER位（置1）；
- 通过将FMC_CTL寄存器的START位置1来发送整片擦除命令到FMC；
- 通过检查FMC_STAT寄存器的BUSY位来判断擦除指令是否执行完毕，若未完成则需等待BUSY位为0；
- 如果有需要，可以使用DBUS读和验证该闪存的内容。

当整片擦除成功执行，FMC_STAT寄存器的ENDF位将会被置1，并且如果FMC_CTL寄存器的ENDIE位之前已经被置1，那么FMC将触发一个中断。由于所有的闪存数据都将被复位为0xFFFF FFFF，可以通过运行在SRAM中的程序或使用调试工具直接访问FMC寄存器来实现

整片擦除操作。FMC_STAT寄存器的ENDF位指示该操作的结束。（编程操作的起始地址需要是0x0800 0000）。下图显示了整片擦除操作的流程：

图 2-2. 整片擦除操作流程



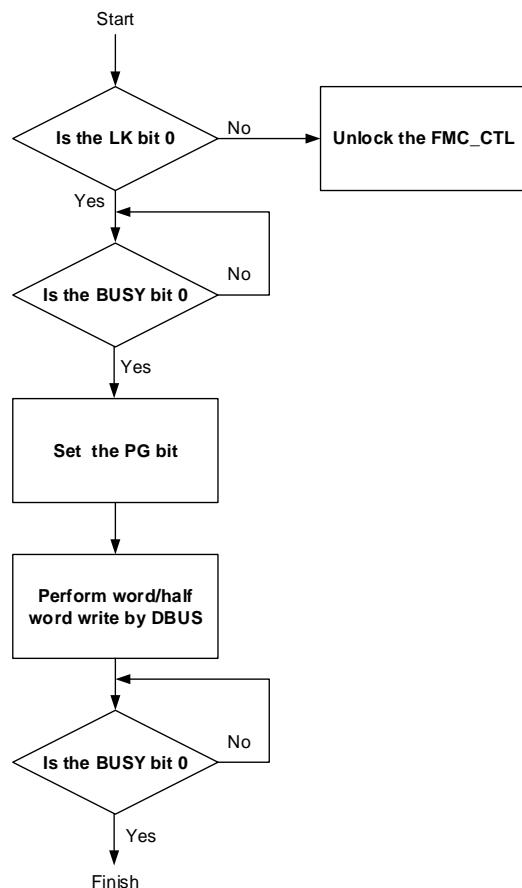
2.3.6. 主存储闪存块编程

FMC提供了一个32位整字/16位半字编程功能，用来修改主闪存存储器的内容。以下步骤显示了字编程操作寄存器的过程：

- 确保FMC_CTL寄存器不处于锁定状态；
- 检查FMC_STAT寄存器的BUSY位来确保闪存存储器没有正在进行中的操作，即BUSY位为0。否则等待该操作完成；
- 写字编程命令到FMC_CTL寄存器的PG位；
- DBUS写一个32位字/16位半字到预期地址；
- 通过检查FMC_STAT寄存器的BUSY位来判断擦除指令是否执行完毕，若未完成则需等待BUSY位为0；
- 如果有需要，可以使用DBUS读并验证该闪存存储器是否编程成功。

当主存储块编程成功执行, FMC_STAT 寄存器的 ENDF 位将会被置 1, 并且如果 FMC_CTL 寄存器的 ENDIE 位之前已经被置 1, 那么 FMC 将触发一个中断。需要注意的是在执行字/半字编程操作之前需要检查目的地址是否已经被擦除过, 如果没有被擦除过, PGERR 位将被置 1, 该页上的编程操作无效(编程内容为 0x0 的情况除外)。另一方面, 在擦除/编程保护的页进行编程操作将会无效。如果 FMC_CTL 寄存器的 ERRIE 位被置 1, FMC 将触发一次闪存操作错误中断。软件可以检查 FMC_STAT 寄存器的 PGERR 位来监测中断处理器的状况。FMC_STAT 寄存器的 ENDF 位指示操作的结束。下图显示了主存储块字编程操作的流程:

图 2-3. 字编程操作流程



2.3.7. 选项字节擦除

FMC提供了一个擦除功能用来初始化闪存中的选项字节块。下面的步骤显示了选项字节块的擦除过程:

- 确保FMC_CTL寄存器不处于锁定状态;
- 确保FMC_CTL寄存器的OBWEN位处于使能状态;
- 检查FMC_STAT寄存器的BUSY位来确保闪存存储器没有正在进行中的操作, 即BUSY位为0。否则等待该操作完成;
- 写选项字节擦除命令到FMC_CTL寄存器的OBER位(置1);
- 通过将FMC_CTL寄存器的START位置1来发送选项字节擦除命令到FMC;
- 通过检查FMC_STAT寄存器的BUSY位来判断擦除指令是否执行完毕, 若未完成则需等待

- BUSY位为0;
- 如果有需要，可以使用DBUS读并验证是否擦除成功。

当选项字节擦除成功执行，FMC_STAT寄存器的ENDF位将会被置1，并且如果FMC_CTL寄存器的ENDIE位之前已经被置1，那么FMC将触发一个中断。FMC_STAT寄存器的ENDF位置1指示该操作的结束。

2.3.8. 选项字节编程

FMC提供了一个32位字/16位半字编程功能，用来修改选项字节块的内容。下面的步骤显示了选项字节编程的操作过程：

- 确保FMC_CTL寄存器不处于锁定状态；
- 确保FMC_CTL寄存器的OBWEN位处于使能状态；
- 检查FMC_STAT寄存器的BUSY位来确保闪存存储器没有正在进行中的操作，即BUSY位为0。否则等待该操作完成；
- 写选项字节编程命令到FMC_CTL寄存器的OBPG位；
- DBUS写一个32位字/16位半字到预期地址；
- 通过检查FMC_STAT寄存器的BUSY位来判断擦除指令是否执行完毕，若未完成则需等待BUSY位为0；
- 如果有需要，可以使用DBUS读并验证是否编程成功。

当选项字节编程操作成功执行，FMC_STAT寄存器的ENDF位将会被置1，并且如果FMC_CTL寄存器的ENDIE位之前已经被置1，那么FMC将触发一个中断。需要注意的是在执行字/半字编程操作之前需要检查目的地址是否已经被擦除过，如果没有被擦除过，PGERR位将被置1，该页上的编程操作无效（编程内容为0x0的情况除外）。FMC_STAT寄存器的ENDF位指示操作的结束。

2.3.9. 选项字节说明

每次系统复位或将 FMC_CTL 寄存器的 OBRLD 位置 1 时，闪存存储器的选项字节块会被重加载到 FMC_OBSTAT 和 FMC_WP 寄存器，之后选项字节生效。选项字节的补码和选项字节相反。当选项字节被重加载，如果选项字节的补码和选项字节不匹配，FMC_OBSTAT 寄存器的OBERR 位将被置 1，选项字节将被设置为 0xFF。选项字节详情见下表：

表 2-2. 选项字节

地址	名称	说明
0x1fff f800	OB_SPC	选项字节安全保护码 0xA5: 无保护 除0xA5或0xCC之外的任意值: 低级别保护 0xCC: 高级别保护
0x1fff f801	OB_SPC_N	OB_SPC补码值
0x1fff f802	OB_USER	用户定义的选项字节 [7]: 保留 [6]: SRAM_PARITY_CHECK

地址	名称	说明
		0: 使能SRAM奇偶校验 1: 去使能SRAM奇偶校验 [5]: VDDA_VISOR 0: 去使能V _{DDA} 监视器 1: 使能V _{DDA} 监视器 [4]: BOOT1_n 0: BOOT1位是1 1: BOOT1位是0 [3]: 保留 [2]: nRST_STDBY 0: 设置待机模式时产生复位而不是进入待机模式 1: 设置待机模式时进入待机模式而不是产生复位 [1]: nRST_DPSLP 0: 设置深度睡眠模式时产生复位而不进入深度睡眠模式 1: 设置深度睡眠模式时进入深度睡眠模式而不产生复位 [0]: nWDG_SW 0: 硬件自动设置独立看门狗定时器 1: 软件设置独立看门狗定时器
0x1fff f803	OB_USER_N	OB_USER补码值
0x1fff f804	OB_DATA[7:0]	用户定义数据位7到0位
0x1fff f805	OB_DATA_N[7:0]	OB_DATA补码值的7到0位
0x1fff f806	OB_DATA[15:8]	用户定义数据位15到8位
0x1fff f807	OB_DATA_N[15:8]	OB_DATA补码值的15到8位
0x1fff f808	OB_WP[7:0]	页擦除/编程保护位的7到0位
0x1fff f809	OB_WP_N[7:0]	OB_WP补码值的7到0位
0x1fff f80a	OB_WP[15:8]	页擦除/编程保护位的15到8位
0x1fff f80b	OB_WP_N[15:8]	OB_WP补码值的15到8位

2.3.10. 页擦除/编程保护

FMC的页擦除/编程保护功可以阻止对闪存存储器的意外操作。当FMC对被保护的页进行页擦除或编程操作时，操作本身无效且FMC_STAT寄存器的WPERR位将被置1。如果WPERR位被置1且ERRIE位也被置1来使能相应的中断，FMC将触发闪存操作错误中断来引起CPU重视。配置选项字节的OB_WP [15:0]位为0可以单独使能每个页的保护功能。如果在选项字节区域执行了页擦除操作，所有的闪存存储器页保护功能都将被禁止。当对选项字节的OB_WP位置1或清0时，需要将FMC_CTL寄存器的OBRLD位置1或系统复位来重加载OB_WP位。下表显示了通过设置OB_WP [15:0]寄存器可以保护哪些页：

表 2-3. OB_WP 位对应页保护

OB_WP位	页保护
--------	-----

OB_WP[0]	页0 ~ 页3
OB_WP[1]	页4 ~ 页7
OB_WP[2]	页8 ~ 页11
.	.
OB_WP[14]	页56 ~ 页59
OB_WP[15]	页60 ~ 页127

2.3.11. 安全保护

FMC提供了一个安全保护功能来阻止闪存上非法的代码或数据访问。此功能可以很好地保护软件和固件免受非法用户的操作。安全保护等级划分为以下三等：

无保护状态：当将OB_SPC字节和它的补码值设置为0xA55A，则不执行保护。主存储块和选项字节块可以被所有操作访问。

保护等级低：当设置OB_SPC字节和它的补码值为除了0xA55A或0xCC33之外的任意值，则执行低级别保护。主闪存仅仅能被用户代码访问。在调试模式下，从SRAM或boot loader模式启动时，所有对主存储块的操作都被禁止，并且如果是读操作将会产生一个总线错误，如果是编程或擦除操作将会导致FMC_STAT寄存器的PGERR位被置1。在低级别保护状态下，选项字节块可以被任意操作访问。如果将OB_SPC字节和它的补码值设置为0xA55A，进入无保护级别，那么将执行一次主存储块整片擦除操作。

保护等级高：当设置OB_SPC字节和它的补码值为0xCC33，则执行高级别保护。在调试模式下，从SRAM或boot loader模式启动都将被禁止。主闪存块可由用户代码的所有操作进行访问。选项字节不能被擦除，OB_SPC字节和它的补码值不能被重编程。因此，如果设置保护级别为高，不能转换为低级别保护和无保护。

2.4. FMC 寄存器

2.4.1. 等待状态寄存器 (FMC_WS)

地址偏移： 0x00

复位值： 0x0000 0000

该寄存器只能按字(32位)访问



rw

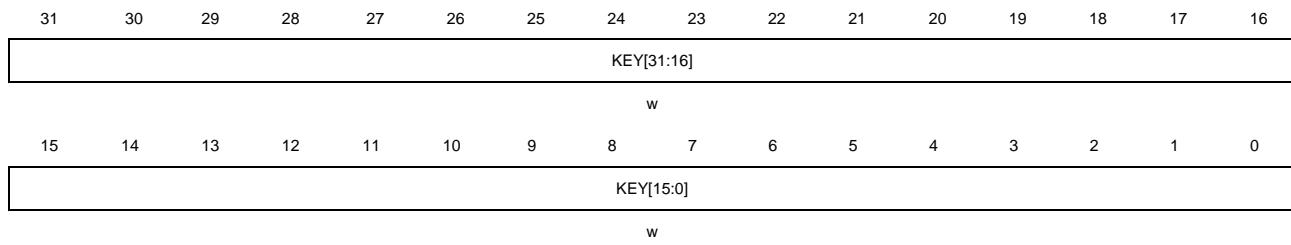
位/位域	名称	描述
31:3	保留	必须保持复位值
2:0	WSCNT[2:0]	等待状态计数寄存器 硬件置1和清0。WSEN位被置1时WSCNT位有效。 000: 不增加等待状态 001: 增加1个等待状态 010: 增加2个等待状态 011 ~ 111: 保留

2.4.2. 解锁寄存器 (FMC_KEY)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器只能按字(32位)访问



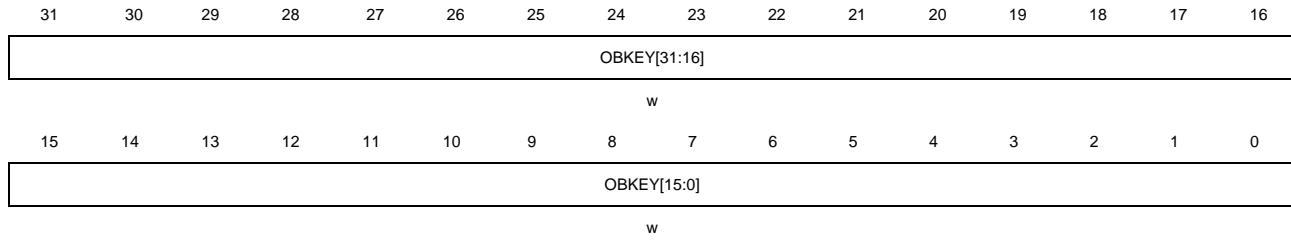
位/位域	名称	描述
31:0	KEY[31:0]	FMC_CTL解锁寄存器 这些位仅能被软件写。 写解锁值到KEY[31:0]来解锁FMC_CTL寄存器。

2.4.3. 选项字节解锁寄存器 (FMC_OBKEY)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器只能按字(32位)访问



位/位域	名称	描述
------	----	----

31:0 OBKEY[31:0] FMC_CTL选项字节解锁寄存器
 这些位仅能被软件写
 写解锁值到OBKEY[31:0]来解锁FMC_CTL寄存器的选项字节命令

2.4.4. 状态寄存器 (FMC_STAT)

地址偏移: 0x0C

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

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

位/位域	名称	描述
31:6	保留	必须保持复位值
5	ENDF	操作结束标志位 操作成功执行后，此位被硬件置1。软件可以通过写1来清0该位。
4	WPERR	擦除/编程保护错误标志位 在受保护的页上进行擦除/编程操作时，此位会被硬件置1。软件可以通过写1来清0该位。
3	保留	必须保持复位值
2	PGERR	编程错误标志位 当在值不为0xFFFF的闪存上进行编程时，此位会被硬件置1。软件可以通过写1来清0该位。
1	保留	必须保持复位值
0	BUSY	闪存忙标志位 当有闪存操作正在进行时，此位被置1。当操作结束或者出错时，此位被清0。

2.4.5. 控制寄存器 (FMC_CTL)

地址偏移: 0x10

复位值: 0x0000 0080

该寄存器只能按字(32位)访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	OBRLD	ENDIE	保留	ERRIE	OBWEN	保留	LK	START	OBER	OBPG	保留	MER	PER	PG	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	名称	描述
31:14	保留	必须保持复位值
13	OBRLD	选项字节重加载位 软件置1。 0: 没有作用 1: 强制选项字节重装载，并产生一次系统复位
12	ENDIE	操作结束中断使能位 软件置1或清0。 0: 无硬件中断产生 1: 使能操作结束中断
11	保留	必须保持复位值
10	ERRIE	错误中断使能位 软件置1或清0。 0: 无硬件中断产生 1: 使能错误中断
9	OBWEN	选项字节擦除/编程使能位 当正确的序列写入FMC_OBKEY寄存器，此位由硬件置1。此位可以被软件清0。
8	保留	必须保持复位值
7	LK	锁FMC_CTL寄存器标志位 当正确的序列写入FMC_KEY寄存器，此位由硬件清0。此位可以由软件置1。
6	START	发送擦除命令到FMC位 软件置1后将发送擦除命令到FMC。当BUSY位被清0后，此位将被硬件清0。
5	OBER	选项字节擦除命令位 软件置1和清0。 0: 无作用 1: 选项字节擦除命令
4	OBPG	选项字节编程命令位 软件置1和清0。 0: 无作用 1: 选项字节编程命令
3	保留	必须保持复位值
2	MER	主存储块整片擦除命令位 软件置1和清0。

		0: 无作用 1: 主存储块整片擦除命令
1	PER	主存储块页擦除命令位 软件置1和清0。 0: 无作用 1: 主存储块页擦除命令
0	PG	主存储块编程命令位 软件置1和清0 0: 无作用 1: 主存储块编程命令

2.4.6. 地址寄存器 0 (FMC_ADDR)

地址偏移: 0x14

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

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

位/位域	名称	描述
31:0	ADDR[31:0]	闪存命令地址位 软件置位 ADDR位是闪存擦除命令的地址。

2.4.7. 选项字节状态寄存器 (FMC_OBSTAT)

地址偏移: 0x1C

复位值: 0xXXXX XX0X

该寄存器只能按字(32位)访问

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

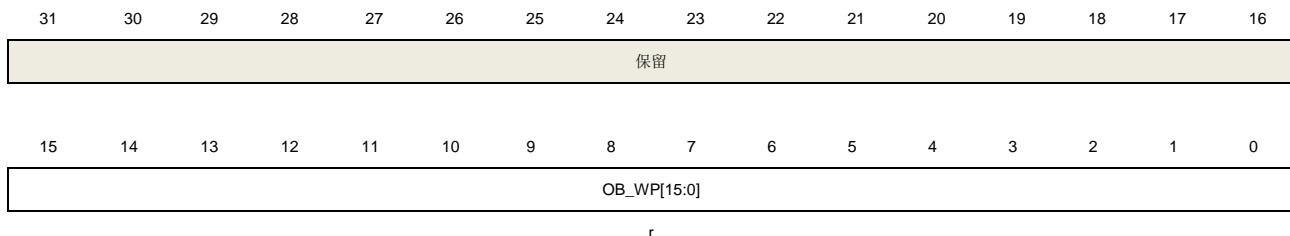
位/位域	名称	描述
31:16	OB_DATA[15:0]	系统复位后保存选项字节块的OB_DATA[15:0]部分
15:8	OB_USER[7:0]	系统复位后保存选项字节块的OB_USER字节
7:3	保留	必须保持复位值
2:1	PLEVEL[1:0]	安全保护级别 00: 无保护 01: 低保护级别 11: 高保护级别
0	OBERR	选项字节读错误位 当选项字节和它的补码不匹配时此位由硬件置1，并且选项字节被设置为0xFF。

2.4.8. 写保护寄存器 (FMC_WP)

地址偏移: 0x20

复位值: 0x0000 XXXX

该寄存器只能按字(32位)访问



r

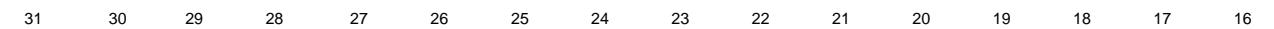
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	OB_WP[15:0]	系统复位后保存选项字节块的OB_WP[15:0]部分 0: 保护生效 1: 未保护

2.4.9. 等待状态使能寄存器 (FMC_WSEN)

地址偏移: 0xFC

复位值: 0x0000 0000

该寄存器只能按字(32位)访问



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

位/位域	名称	描述
31:2	保留	必须保持复位值
1	BPEN	<p>FMC位编程使能寄存器</p> <p>此位由软件置1和清0。</p> <p>0: 无影响, 页编程操作前必须检查闪存值是否是“FF”</p> <p>1: 页编程前不需要检查闪存值是否是“FF”, FMC能够编程每一位</p>
0	WSEN	<p>FMC等待状态使能寄存器</p> <p>此位由软件置1和清0。此位也被FMC_KEY寄存器保护。软件需要写0x45670123和0xCDEF89AB到FMC_KEY寄存器。</p> <p>0: 从闪存取指无等待状态增加</p> <p>1: 从闪存取指增加等待状态</p>

2.4.10. 产品 ID 寄存器 (FMC_PID)

地址偏移: 0x100

复位值: 0xXXXX XXXX

该寄存器只能按字(32位)访问

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

位/位域	名称	描述
31:0	PID1[31:0]	<p>产品保留ID寄存器</p> <p>该寄存器为只读</p> <p>上电后这些位始终不会改变, 该寄存器在生产过程中被一次性编程。</p>

3. 电源管理单元（PMU）

3.1. 简介

功耗设计是GD32F3x0系列产品比较注重的问题之一。电源管理单元提供了三种省电模式，包括睡眠模式，深度睡眠模式和待机模式。这些模式能减少电源能耗，且使得应用程序可以在CPU运行时间要求、速度和功耗的相互冲突中获得最佳折衷。如[图3-1. 电源域概览](#)所示，GD32F3x0系列设备有三个电源域，包括 V_{DD}/V_{DDA} 域，1.2V域和备份域。 V_{DD}/V_{DDA} 域由电源直接供电。在 V_{DD}/V_{DDA} 域中嵌入了一个LDO，用来为1.2V域供电。在备份域中有一个电源切换器，当 V_{DD} 电源关闭时，电源切换器可以将备份域的电源切换到 V_{BAT} 引脚，此时备份域由 V_{BAT} 引脚（电池）供电。

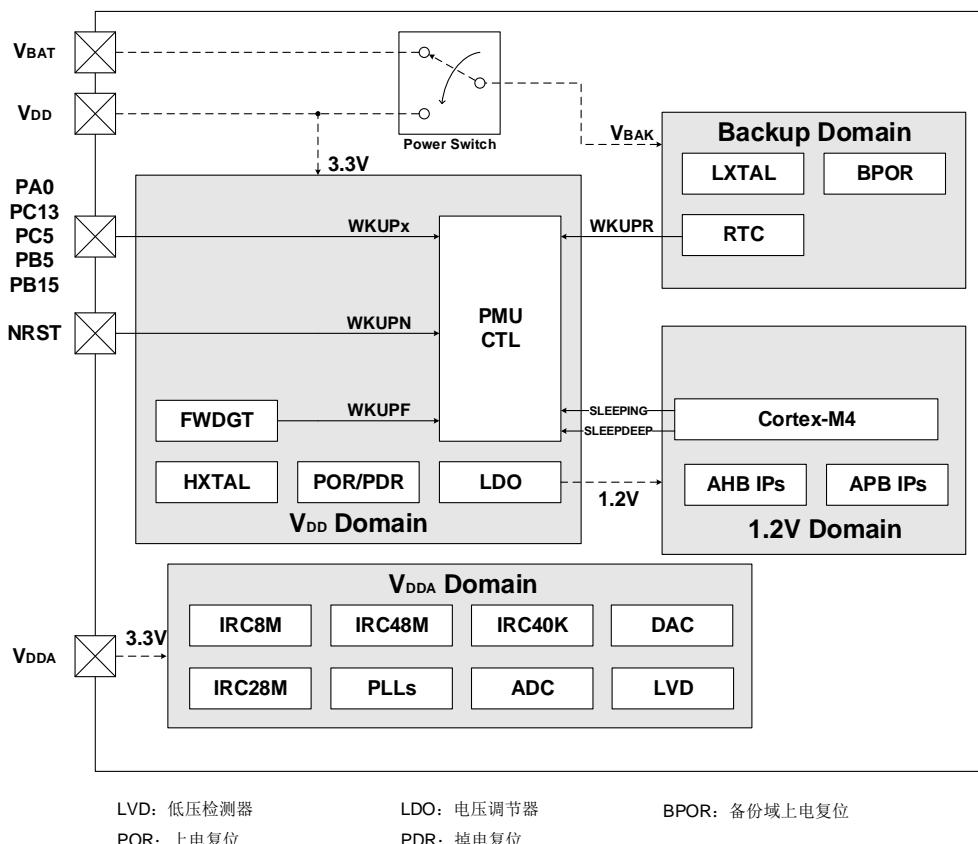
3.2. 主要特性

- 三个电源域：备份域、 V_{DD}/V_{DDA} 域和1.2V电源域；
- 三种省电模式：睡眠模式、深度睡眠模式和待机模式；
- 内部电压调节器（LDO）提供1.2V电源；
- 提供低电压检测器，当电压低于所设定的阈值时能发出中断或事件；
- 当 V_{DD} 供电关闭时，由 V_{BAT} （电池）为备份域供电；
- LDO输出电压用于节约能耗；
- 低驱动模式用于在深入睡眠模式下超低功耗。高驱动模式用在高频模式中。

3.3. 功能描述

[图3-1. 电源域概览](#)提供了PMU及相关电源域的内部结构框图。

图3-1. 电源域概览



3.3.1. 电池备份域

电池备份域由内部电源切换器来选择V_{DD}供电或V_{BAT}（电池）供电，然后由V_{BAK}为备份域供电，该备份域包含RTC（实时时钟）、LXTAL（低速外部晶体振荡器）、BPOR（备份域上电复位），以及PC13至PC15共3个PAD。为了确保备份域中寄存器的内容及RTC正常工作，当V_{DD}关闭时，V_{BAT}引脚可以连接至电池或其他电源等备份源供电。电源切换器是由V_{DD}/V_{DDA}域掉电复位电路控制的。对于没有外部电池的应用，建议将V_{BAT}引脚通过100nF的外部陶瓷去耦电容连接到V_{DD}引脚上。

备份域的复位源包括备份域上电复位和备份域软件复位。在V_{BAK}没有完全上电前，BPOR信号强制设备处于复位状态。应用软件可以通过设置RCU_BDCTL寄存器BKPRST位来触发备份域软件复位。

RTC的时钟源可以是低速内部RC振荡器（IRC40K）或低速外部晶体振荡器（LXTAL），或高速外部晶体振荡器（HXTAL）时钟32分频。当V_{DD}被关闭时，RTC只能选择LXTAL作为时钟源。在通过WFI/WFE指令进入省电模式之前，Cortex™-M4需要通过RTC寄存器设置预期的闹钟时间并启用闹钟功能，以实现RTC定时器唤醒事件。进入省电模式一定时间之后，当经过的时间与预设的唤醒时间匹配时，RTC将唤醒设备。RTC的配置和操作的细节将在RTC章节来描述。

当备份域由V_{DD}供电（V_{BAK}连接至V_{DD}）时，以下功能可用：

- PC13可以作为通用I/O口或RTC功能引脚（参见[实时时钟 \(RTC\)](#)）；
- PC14和PC15可以作为通用I/O口或LXTAL晶振引脚。

当备份域由 V_{BAT} 电源供电时 (V_{BAK} 连接至 V_{BAT})，以下功能可用：

- PC13 仅可以作为 RTC 功能引脚（参见 [实时时钟 \(RTC\)](#)）；
- PC14 和 PC15 仅可作为 LXTAL 晶振引脚。

注意：由于 PC13 至 PC15 引脚是通过电源切换器供电的，电源切换器仅可通过小电流，因此当 PC13 至 PC15 的 GPIO 口在输出模式时，其工作的速度不能超过 2MHz(最大负载为 30pF)。

3.3.2. VDD/VDDA 电源域

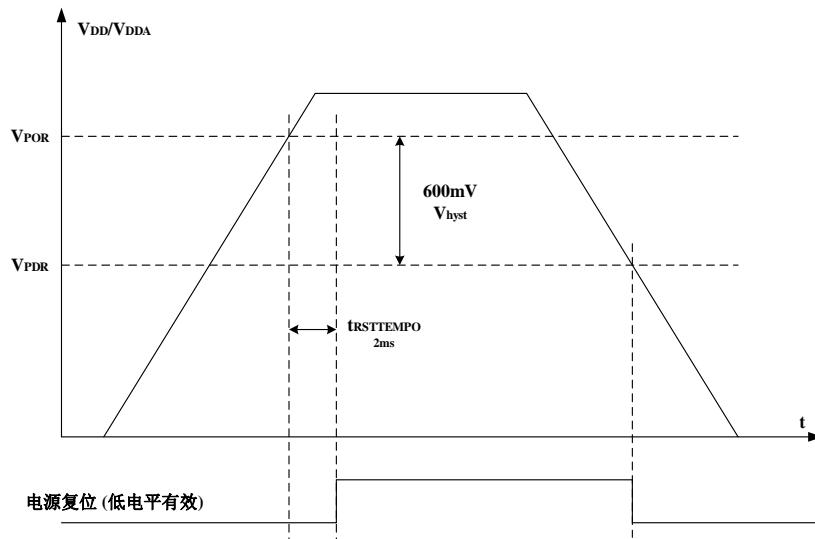
V_{DD}/V_{DDA} 域包括 V_{DD} 域和 V_{DDA} 域两部分。 V_{DD} 域包括 HXTAL (高速外部晶体振荡器)、LDO (电压调节器)、POR/PDR (上电/掉电复位)、FWDGT (独立看门狗定时器) 和除 PC13、PC14 和 PC15 之外的所有 PAD 等等。 V_{DDA} 域包括 ADC/DAC (AD/DA 转换器)、IRC8M (内部 8M RC 振荡器)、IRC28M (内部 28M RC 振荡器)、IRC48M (内部 48M RC 振荡器)、IRC40K (内部 40KHz RC 振荡器) PLLs (锁相环) 和 LVD (低电压检测器) 等等。

V_{DD} 域

为 1.2V 域供电的 LDO (电压调节器)，其复位后保持使能。可以被配置为三种不同的工作状态：包括睡眠模式 (全供电状态)、深度睡眠模式 (全供电或低功耗状态) 和待机模式 (关闭状态)。

POR/ PDR (上电/掉电复位) 电路检测 V_{DD}/V_{DDA} 并在电压低于特定阈值时产生电源复位信号复位除备份域之外的整个芯片。[图 3-2. 上电/掉电复位波形图](#) 显示了供电电压和电源复位信号之间的关系。 V_{POR} 表示上电复位的阈值电压，典型值约为 2.40V， V_{PDR} 表示掉电复位的阈值电压，典型值约为 1.8V。迟滞电压 V_{hyst} 值约为 600mV。

图3-2. 上电/掉电复位波形图

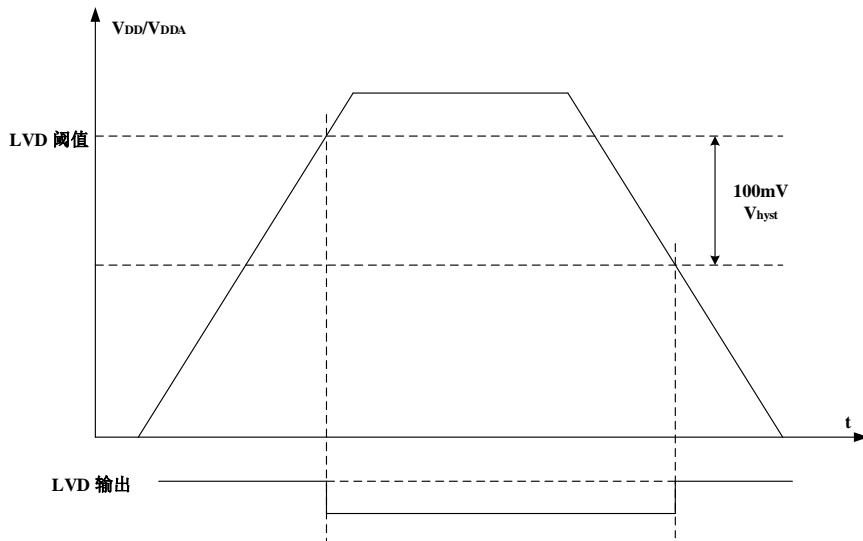


V_{DDA} 域

LVD 的功能是检测 V_{DD}/V_{DDA} 供电电压是否低于低电压检测阈值，该阈值由电源控制寄存器 (PMU_CTL) 中的 LVDT[2:0] 位进行配置。LVD 通过 LVDEN 置位使能，位于电源状态寄存器

(PMU_CS)中的 LVDF 位表示低电压事件是否出现，该事件连接至 EXTI 的第 16 线，用户可以通过配置 EXTI 的第 16 线产生相应的中断。[图 3-3. LVD 阈值波形图](#)显示了 V_{DD}/V_{DDA} 供电电压和 LVD 输出信号的关系。(LVD 中断信号依赖于 EXTI 第 16 线的上升或下降沿配置)。迟滞电压 V_{hyst} 值为 100mV。

图3-3. LVD阈值波形图



一般来说，数字电路由 V_{DD} 供电，而大多数的模拟电路由 V_{DDA} 供电。为了提高 ADC 和 DAC 的转换精度，为 V_{DDA} 独立供电可使模拟电路达到更好的特性。为避免噪声，V_{DDA} 通过外部滤波电路连接至 V_{DD}，相应的 V_{SSA} 通过特定电路连接至 V_{SS}。否则，如果 V_{DDA} 和 V_{DD} 不同时，V_{DDA} 须高于 V_{DD}，但压差不超过 0.2V。

为保证低电压时 ADC 和 DAC 的高精度，独立的外部参考电压 V_{REF} 被连接至 ADC/DAC 引脚。根据不同的封装，V_{REF+} 须被连接至 V_{DDA} 引脚，V_{REF-} 须被连接至 V_{SSA} 引脚，V_{REF+} 引脚仅存在于不小于 100-pin 封装上，而在 64-pin 或更少引脚封装不存在，因其内部已经连接至 V_{DDA} 和 V_{SSA}。V_{REF-} 仅存在于 176-pin 封装上，在其他封装里，其内部连接至 V_{SSA}。

3.3.3. 1.2V 电源域

主要功能包括 Cortex™-M4 内核逻辑、AHB/APB 外设、备份域和 V_{DD}/V_{DDA} 域的 APB 接口等。当 1.2V 电压上电后，POR 将在 1.2V 域中产生一个复位序列，复位完成后，要进入指定的省电模式，须先配置相关的控制位，之后一旦执行 WFI 或 WFE 指令，设备便进入该省电模式。关于这方面的详细内容，将在以下章节予以说明。

高驱动模式

如果 1.2V 电源域需要工作在高频状态下，并同时打开多种功能，建议进入高驱动模式。使用高驱动模式有以下步骤：

- 选择系统时钟为IRC8M或HXTAL；
- 将PMU_CTL寄存器的HDEN置1，使能高驱动模式；
- 等待PMU_CS寄存器的HDRF被置位；

- 将PMU_CTL寄存器的HDS置1，将LDO切换到高驱动模式；
- 等待PMU_CS寄存器的HDSRF被置位。进入高驱动模式；
- PLL配置工作在高频状态。

在选择IRC8M或HXTAL作为系统时钟后，可以通过将PMU_CTL寄存器的HDEN和HDS清0退出高驱动模式。当系统退出深度睡眠模式，将会自动退出高驱动模式。

3.3.4. 省电模式

系统复位或电源复位后，GD32F3x0 MCU处于全功能状态且电源域全部处于供电状态。实现较低的功耗的方法有三种：减慢系统时钟（HCLK, PCLK1, PCLK2），关闭未使用的外设的时钟或通过PMU_CTL寄存器的LDOVS来配置LDO输出电压。LDOVS只有在PLL关闭情况下才可以配置，在PLL打开时，被配置的LDO输出电压才会被用来驱动1.2V电源域。此外，三种省电模式可以实现更低的功耗，它们是睡眠模式、深度睡眠模式和待机模式。

睡眠模式

睡眠模式与Cortex™-M4的SLEEPING模式相对应。在睡眠模式下，仅关闭Cortex™-M4的时钟。如需进入睡眠模式，只要清除Cortex™-M4系统控制寄存器中的SLEEPDEEP位，并执行一条WFI或WFE指令即可。如果睡眠模式是通过执行WFI指令进入的，任何中断都可以唤醒系统。如果睡眠模式是通过执行WFE指令进入的，任何唤醒事件都可以唤醒系统（如果SEVONPEND为1，任何来自EXTI的中断都可以唤醒系统，请参考Cortex™-M4技术手册）。由于无需在进入或退出中断上消耗时间，该模式所需的唤醒时间最短。

根据Cortex™-M4中SCR（系统控制寄存器）的SLEEPONEXIT位，有两种睡眠进入机制可选：

- Sleep-now: 如果SLEEPONEXIT位被清零，一旦执行WFI或WFE指令，MCU立即进入睡眠模式；
- Sleep-on-exit: 如果SLEEPONEXIT位被置位，当系统从最低优先级的中断处理程序离开后，MCU立即进入睡眠模式。

深度睡眠模式

深度睡眠模式与Cortex™-M4的SLEEPDEEP模式相对应。在深度睡眠模式下，1.2V域中的所有时钟全部关闭，IRC8M、IRC28M、IRC48M、HXTAL及PLLs也全部被禁用。SRAM和寄存器中的内容被保留。根据PMU_CTL寄存器的LDOLP位的配置，可控制LDO工作在正常模式或低功耗模式。进入深度睡眠模式之前，先将Cortex™-M4系统控制寄存器的SLEEPDEEP位置1，再清除PMU_CTL寄存器的STBMOD位，然后执行WFI或WFE指令即可进入深度睡眠模式。如果睡眠模式是通过执行WFI指令进入的，任何来自EXTI的中断可以将系统从深度睡眠模式中唤醒。如果睡眠模式是通过执行WFE指令进入的，任何来自EXTI的事件可以将系统从深度睡眠模式中唤醒（如果SEVONPEND为1，任何来自EXTI的中断都可以唤醒系统，请参考Cortex™-M4技术手册）。刚退出深度睡眠模式时，IRC8M被选中作为系统时钟。请注意，如果LDO工作在低功耗模式，那么唤醒时需额外的延时时间。

在深度睡眠模式下，通过配置PMU_CTL寄存器的LDEN, LDNP, LDLP, LDOLP位可以进入低驱动模式。低驱动模式具有低驱动能力，低能耗。

正常驱动/正常功耗：将 PMU_CTL 寄存器的 LDEN 位配置为 00，深度睡眠模式就工作在正常驱动模式下。将 PMU_CTL 寄存器的 LDOLP 清 0 可以退出低功耗模式。

正常驱动/低功耗：将 PMU_CTL 寄存器的 LDEN 位配置为 00，深度睡眠模式就工作在正常驱动模式下。将 PMU_CTL 寄存器的 LDOLP 置 1 可以进入低功耗模式。

低驱动/正常功耗：将 PMU_CTL 寄存器的 LDEN 设置为 0b11，LDNP 置 1 可以进入深度睡眠模式的低驱动模式。将 PMU_CTL 寄存器的 LDOLP 清 0 可以使 LDO 处于正常功耗模式。

低驱动/低功耗：将 PMU_CTL 寄存器的 LDEN 设置为 0b11，LDLP 置 1 可以进入深度睡眠模式的低驱动模式。将 PMU_CTL 寄存器的 LDOLP 置 1 可以使 LDO 处于低功耗模式。

非低驱动：将 PMU_CTL 寄存器的 LDEN 设置为 00，深度睡眠模式将不会处在低驱动模式。

注意：为了顺利进入深度睡眠模式，所有 EXTI 线上的挂起状态（在 EXTI_PD 寄存器中）和 RTC 闹钟/时间戳/侵入标志必须被复位。否则，程序将直接跳过深度睡眠模式进入过程而继续执行下面的程序。

待机模式

待机模式是基于 Cortex™-M4 的 SLEEPDEEP 模式实现的。在待机模式下，整个 1.2V 域全部停止供电，同时 LDO 和包括 IRC8M、IRC28M、IRC48M、HXTAL 和 PLL 也会被关闭。进入待机模式前，先将 Cortex™-M4 系统控制寄存器的 SLEEPDEEP 位置 1，再将 PMU_CTL 寄存器的 STBMOD 位置 1，再清除 PMU_CS 寄存器的 WUF 位，然后执行 WFI 或 WFE 指令，系统进入待机模式，PMU_CS 寄存器的 STBF 位状态表示 MCU 是否曾进入过待机模式。待机模式有四个唤醒源，包括来自 NRST 引脚的外部复位，RTC 闹钟/时间戳/侵入事件，FWDGT 复位，WKUP 引脚的上升沿。待机模式可以达到最低的功耗，但唤醒时间最长。另外，一旦进入待机模式，SRAM 和 1.2V 电源域寄存器的内容都会丢失。退出待机模式时，会发生上电复位，复位之后 Cortex™-M4 将从 0x00000000 地址开始执行指令代码。

表3-1. 节电模式总结

模式	睡眠	深度睡眠	待机
描述	仅关闭 CPU 时钟	1、关闭 1.2V 电源域的所有时钟 2、关闭 IRC8M、IRC28M、IRC48M、HXTAL 和 PLL	1、关闭 1.2V 电源域的供电 2、关闭 IRC8M、IRC28M、IRC48M、HXTAL 和 PLL
LDO 状态	开启	开启、低功耗模式或低驱动模式	关闭
配置	SLEEPDEEP = 0	SLEEPDEEP = 1 STBMOD = 0	SLEEPDEEP = 1 STBMOD = 1, WURST=1
进入指令	WFI 或 WFE	WFI 或 WFE	WFI 或 WFE
唤醒	若通过 WFI 进入，则任何中断均可唤醒；若通过 WFE 进入，则任何事件（或 SEVONPEND=1 时的	若通过 WFI 进入，来自 EXTI 的任何中断可唤醒；若通过 WFE 进入，来自 EXTI 的任何事件（或 SEVONPEND=1 时的中	1、NRST 引脚 2、WKUP 引脚 3、FWDGT 复位 4、RTC

模式	睡眠	深度睡眠	待机
	中断) 均可唤醒	断) 可唤醒	
唤醒延迟	无	IRC8M 唤醒时间 如果 LDO 处于低功耗模式, 需增加 LDO 唤醒时间	上电序列

3.4. PMU 寄存器

3.4.1. 控制寄存器 (PMU_CTL)

地址偏移: 0x00

复位值: 0x0000 C000 (从待机模式唤醒后复位)

该寄存器可以按半字 (16位) 或字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留										LDEN[1:0]		HDS		HDEN	
										rw		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LDOVS[1:0]	保留	保留	LDNP	LDLP	保留	BKPWEN	LVDT[2:0]		LVDEN	STBRST	WURST	STBMOD	LDOLP	rw	rw
rs			rw	rw		rw		rw		rw	rc_w1	rc_w1	rw	rw	

位/位域	名称	描述
31:20	保留	必须保持复位值
19:18	LDEN[1:0]	深度睡眠模式下, 低驱动模式使能 00: 在深度睡眠模式下, 禁用低驱动模式 01: 保留 10: 保留 11: 在深度睡眠模式下, 使能低驱动模式
17	HDS	高驱动模式切换器 选择IRC8M或HXTAL作为系统时钟, 当HDRF被置位时, 由软件将该位置1。该位被置位后, 系统进入高驱动模式。可由软件清0, 也可在退出深度睡眠模式或HDEN被清0时由硬件清0。 0: 没有高驱动模式切换器 1: 有高驱动模式切换器
16	HDEN	高驱动模式使能 当系统时钟为IRC8M或HXTAL时, 该位由软件置位。当系统退出深度睡眠模式时, 该位由软件或硬件清零。 0: 禁用高驱动模式 1: 使能高驱动模式
15:14	LDOVS[1:0]	选择LDO输出

在PLL关闭时，这些位由软件配置，在主PLL使能后，LLDOVS设置的值生效。如果主PLL关闭，LDO输出低电压模式被选中(不改变该位域的值)。

00: 保留 (LDO输出低电压模式)

01: LDO输出低电压模式

10: LDO输出中电压模式

11: LDO输出高电压模式

13:12	保留	必须保持复位值
11	LDNP	使用正常功耗LDO时，工作在低驱动模式 0: 使用正常功耗LDO时，工作在正常驱动模式 1: 使用正常功耗LDO且LDEN为0b'11时，低驱动模式被使能
10	LDLP	使用低功耗LDO时，工作在低驱动模式 0: 使用低功耗LDO时，工作在正常驱动模式 1: 使用低功耗LDO且LDEN为0b'11时，低驱动模式被使能
9	保留	必须保持复位值
8	BKPWEN	备份域写使能 0: 禁止对备份域寄存器的写访问 1: 允许对备份域寄存器的写访问 复位之后，任何对备份域寄存器的写访问都将被禁止。如需对备份域寄存器做写访问，需先将该位置1。
7:5	LVDT[2:0]	低电压检测器阈值 000: 2.1V 001: 2.3V 010: 2.4V 011: 2.6V 100: 2.7V 101: 2.9V 110: 3.0V 111: 3.1V
4	LVDEN	低电压检测器使能 0: 关闭低电压检测器 1: 开启低电压检测器
3	STBRST	待机标志复位 0: 无影响 1: 复位待机标志 读该位，始终返回0
2	WURST	唤醒标志复位 0: 无影响 1: 复位唤醒标志 读该位，始终返回0

1	STBMOD	待机模式 0: 当Cortex™-M4进入SLEEPDEEP模式时, 系统进入深度睡眠模式 1: 当Cortex™-M4进入SLEEPDEEP模式时, 系统进入待机模式
0	LDOLP	LDO低功耗模式 0: 当系统进入深度睡眠模式时, LDO仍正常工作 1: 当系统进入深度睡眠模式时, LDO进入低功耗模式

3.4.2. 电源控制和状态寄存器 (PMU_CS)

地址偏移: 0x04

复位值: 0x0000 0000 (从待机模式唤醒后不复位)

该寄存器可以按半字 (16位) 或字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留														LDRF[1:0]	HDSRF	HDRF
														RC_W1	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
LDOVSR F	WUPEN6	WUPEN5	WUPEN4	保留	WUPEN1	WUPENO	保留				LVDF	STBF	WUF			
r	rw	rw	rw		rw	rw					r	r	r			

位/位域	名称	描述
31:20	保留	必须保持复位值
19:18	LDRF[1:0]	低驱动模式就绪标志 在深度睡眠模式下, 且LDO处于低驱动模式, 这些位由硬件设置。软件对这些位写11可以清0。 00: 深度睡眠模式下, 普通驱动模式 01: 保留 10: 保留 11: 深度睡眠模式下, 低驱动模式
17	HDSRF	高驱动切换器就绪标志 0: 高驱动切换器未就绪 1: 高驱动切换器就绪
16	HDRF	高驱动准备就绪标志 0: 高驱动未就绪 1: 高驱动就绪
15	LDOVSRF	LDO电压选择就绪标志 0: LDO电压选择未就绪 1: LDO电压选择就绪
14	WUPEN6	WKUP引脚6 (PB15) 唤醒使能

		0: 关闭WKUP引脚6唤醒功能 1: 开启WKUP引脚6唤醒功能 如果WUPEN6在进入省电模式之前置1, WKUP引脚6的上升沿会将系统从省电模式唤醒。 由于WKUP引脚6为高电平有效, WKUP引脚6内部被配置为输入下拉模式。当置位该控制位后, 当输入为高的时候, 将会触发一个唤醒事件。
13	WUPEN5	WKUP引脚5 (PB5) 唤醒使能 0: 关闭WKUP引脚5唤醒功能 1: 开启WKUP引脚5唤醒功能 如果WUPEN5在进入省电模式之前置1, WKUP引脚5的上升沿会将系统从省电模式唤醒。 由于WKUP引脚5为高电平有效, WKUP引脚5内部被配置为输入下拉模式。当置位该控制位后, 当输入为高的时候, 将会触发一个唤醒事件。
12	WUPEN4	WKUP引脚4 (PC5) 唤醒使能 0: 关闭WKUP引脚4唤醒功能 1: 开启WKUP引脚4唤醒功能 如果WUPEN4在进入省电模式之前置1, WKUP引脚4的上升沿会将系统从省电模式唤醒。 由于WKUP引脚4为高电平有效, WKUP引脚4内部被配置为输入下拉模式。当置位该控制位后, 当输入为高的时候, 将会触发一个唤醒事件。
11:10	保留	必须保持复位值
9	WUPEN1	WKUP引脚1 (PC13) 唤醒使能 0: 关闭WKUP引脚1唤醒功能 1: 开启WKUP引脚1唤醒功能 如果WUPEN1在进入省电模式之前置1, WKUP引脚1的上升沿会将系统从省电模式唤醒。 由于WKUP引脚1为高电平有效, WKUP引脚1内部被配置为输入下拉模式。当置位该控制位后, 当输入为高的时候, 将会触发一个唤醒事件。
8	WUPEN0	WKUP引脚0 (PA0) 唤醒使能 0: 关闭WKUP引脚0唤醒功能 1: 开启WKUP引脚0唤醒功能 如果WUPEN0在进入省电模式之前置1, WKUP引脚0的上升沿会将系统从省电模式唤醒。 由于WKUP引脚0为高电平有效, WKUP引脚0内部被配置为输入下拉模式。当置位该控制位后, 当输入为高的时候, 将会触发一个唤醒事件。
7:3	保留	必须保持复位值
2	LVDF	低电压状态标志 0: 低电压事件没出现 (V_{DD} 高于设定的LVD阈值) 1: 低电压事件出现 (V_{DD} 等于或低于LVD阈值) 注意: LVD功能在待机模式被禁用。
1	STBF	待机标志 0: 设备没进入过待机模式 1: 设备曾进入过待机模式 该位只能由POR/PDR或通过设置PMU_CTL寄存器的STBRST位来清零。

0

WUF

唤醒标志

0: 没有收到唤醒事件

1: 收到来自WKUP引脚或RTC唤醒事件，包括RTC侵入事件、RTC闹钟事件、RTC时间戳事件。

该位只能由POR/PDR或通过设置PMU_CTL寄存器的STBRST位来清零。

4. 复位和时钟单元 (RCU)

4.1. 复位控制单元 (RCTL)

4.1.1. 简介

GD32F3x0复位控制包括三种复位控制：电源复位、系统复位和备份域复位。电源复位又称为冷复位，电源启动时复位除了备份域的所有系统。除了SW-DP控制器和备份域，系统复位将复位处理器内核和外设IP部分。备份域复位复位备份区域。复位被外部信号、内部事件和复位发生器触发。接下章节将详细介绍这些复位。

4.1.2. 功能描述

电源复位

当以下事件中之一发生时，产生电源复位：1、上电/掉电复位(POR/PDR 复位) 2、从待机模式中返回后由内部复位发生器产生。电源复位复位所有的寄存器除了备份域。电源复位为低电平有效，当内部LDO电源基准准备好提供1.2V电压给GD32F3x0产品时，电源复位电平将变为无效。复位入口向量被固定在存储器映射地址0x0000_0004。

系统复位

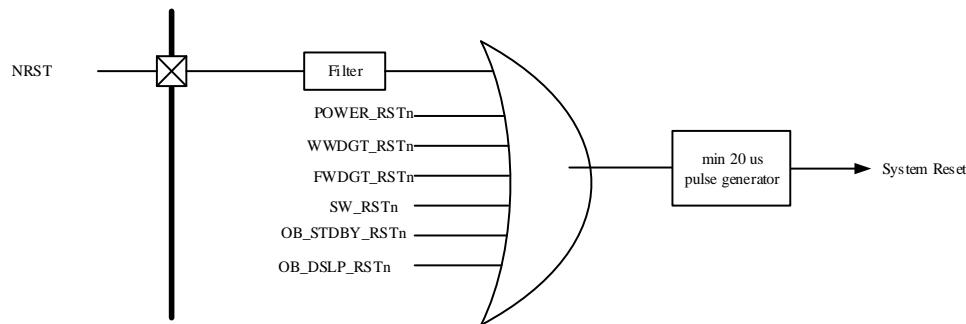
当发生以下任一事件时，产生一个系统复位：

- 电源复位 (POWER_RSTn);
- 外部引脚复位 (NRST);
- 窗口看门狗定时器计数终止 (WWDGT_RSTn);
- 独立看门狗定时器计数终止 (FWDGT_RSTn);
- Cortex™-M4的中断应用和复位控制寄存器中的SYSRESETREQ位置‘1’ (SW_RSTn);
- 用户选择字节寄存器nRST_STDBY设置为0，并且进入待机模式时(STDBY_RSTn);
- 用户选择字节寄存器nRST_DPSLP设置为0，并且进入深度睡眠模式时(DPSLP_RSTn)。

除了SW-DP控制器和备份域，系统复位将复位处理器内核和外设IP部分。

系统复位脉冲发生器保证每一个复位源(外部或内部)都能有至少20μs的低电平脉冲延时。

图 4-1. 系统复位电路



备份域复位

当以下事件之一发生时，产生备份域复位。1、设置备份域控制寄存器中的BKPRST位为‘1’；
2、备份域电源上电复位(在V_{DD}和V_{BAT}两者掉电的前提下，V_{DD}或V_{BAT}上电)。

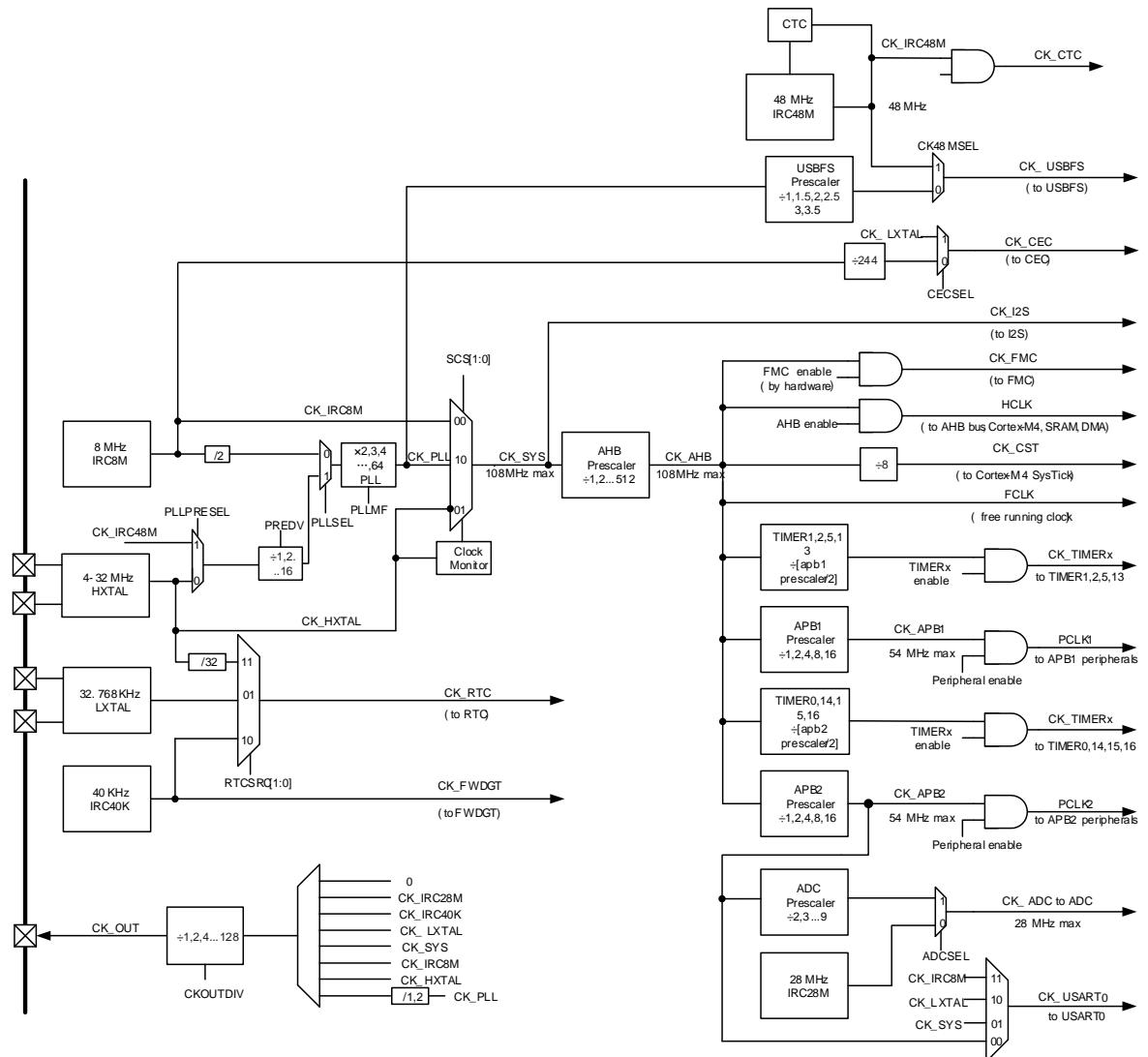
4.2. 时钟控制单元 (CCTL)

4.2.1. 简介

时钟控制单元提供了一系列频率和时钟功能，包括一个内部8M RC振荡器时钟(IRC8M)、一个内部48M RC 振荡器时钟(IRC48M)、一个内部28M RC振荡器时钟(IRC28M)、一个外部高速晶体振荡器时钟(HXTAL)、一个内部低速RC振荡器时钟(IRC40K)、一个外部低速晶体振荡器时钟(LXTAL)、一个锁相环(PLL)、一个HXTAL时钟监视器、时钟预分频器、时钟多路复用器和时钟选通电路。

AHB、APB和Cortex™-M4时钟都源自系统时钟(CK_SYS)，系统时钟的时钟源为IRC8M、HXTAL或PLL。系统时钟的最大运行时钟频率可以达到108MHz。独立看门狗定时器定时器有独立的时钟源(IRC40K)，实时时钟(RTC)使用IRC40K、LXTAL或HXTAL/32作为时钟源。

图4-2. 时钟树



预分频器可以配置AHB、APB2和APB1域的时钟频率。AHB、APB2和APB1域的最高时钟频率分别为108MHz、54MHz和54MHz。RCU通过AHB时钟(HCLK)8分频后作为Cortex系统定时器(SysTick)的外部时钟。通过对SysTick控制与状态寄存器的设置，可选择上述时钟或Cortex(HCLK)时钟作为SysTick时钟。

在GD32F3x0产品中ADC时钟由APB2时钟经2到9分频或IRC28M/2时钟获得，它们是通过设置配置寄存器2(RCU_CFG2)的ADCSEL位来选择ADC时钟源的。USART0的时钟可以选择IRC8M时钟、LXTAL时钟、系统时钟或APB2时钟，通过设置配置寄存器2(RCU_CFG2)的USART0SEL位来选择。CEC时钟可以选择IRC8M时钟的244分频或LXTAL时钟，通过设置配置寄存器2(RCU_CFG2)的CECSEL位来选择。

RTC时钟可以选择LXTAL时钟、IRC40K时钟或HXTAL时钟32分频，通过设置备用域控制寄存器(RCU_BDCTL)的RTCSR位来选择。

USBFS时钟由CK48M提供，CK48M可以选择CK_PLL或者IRC48M时钟，通过设置RCU_ADDCTL寄存器的CK48MSEL位来选择IRC48M。

FWDGT时钟可以选择IRC40K时钟，当FWDGT启动时强制选择。

如果APB时钟分频系数为1，定时器的时钟频率与所在APB总线频率一致。否则，定时器的时钟频率被设为与其相连的APB总线频率的2倍。

4.2.2. 主要特性

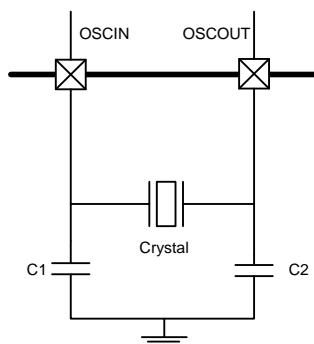
- 4到32 MHz外部高速晶体振荡器 (HXTAL)
- 8 MHz内部高速RC振荡器 (IRC8M)
- 48 MHz内部高速RC振荡器 (IRC48M)
- 28 MHz内部高速RC振荡器 (IRC28M)
- 32,768 Hz外部低速晶体振荡器 (LXTAL)
- 40 kHz内部低速RC振荡器 (IRC40K)
- PLL时钟源可以是HXTAL或IRC8M或IRC48M
- HXTAL时钟监视

4.2.3. 功能描述

高速外部晶体振荡器时钟(HXTAL)

4到32MHz的外部振荡器可为系统提供更为精确的主时钟。带有特定频率的晶体必须靠近两个HXTAL的引脚。和晶体连接的外部电阻和电容必须根据所选择的振荡器来调整。

图4-3. HXTAL时钟源



HXTAL晶体可以通过设置时钟控制寄存器RCU_CTL0的HXTALEN位来启动或关闭，在时钟控制寄存器RCU_CTL0中的HXTALSTB位用来指示高速外部振荡器是否已稳定。在启动时，直到这一位被硬件置‘1’，时钟才被释放出来。这个特定的延迟时间又称启动时间。当HXTAL时钟稳定后，如果在时钟中断寄存器RCU_INT中的相应中断使能位HXTALSTBIE位被置‘1’，将会产生相应中断。在这一点上，HXTAL时钟可以被直接用作系统时钟源或者PLL输入时钟。

将时钟控制寄存器RCU_CTL0的HXTALBPS和HXTALEN位置‘1’可以设置外部时钟旁路模式。CK_HXTAL等于驱动OSCIN管脚的外部时钟。

高速内部8MHz RC振荡器时钟 (IRC8M)

高速内部8MHz RC振荡器时钟，简称IRC8M时钟，拥有8MHz的固定频率，设备上电后CPU默认选择的时钟源就是IRC8M时钟。IRC8M RC振荡器能够在不需要任何外部器件的条件下提供更低成本类型的时钟源。IRC8M晶体可以通过设置时钟控制寄存器(RCU_CTL0)中的IRC8MEN位被启动和关闭。时钟控制寄存器RCU_CTL0中的IRC8MSTB位用来指示IRC8M内部RC振荡器是否稳定。IRC8M振荡器的启动时间比HXTAL晶体振荡器要短。如果时钟中断寄存器RCU_INT中的相应中断使能位IRC8MSTBIE被置‘1’，在IRC8M稳定以后，将产生一个中断。IRC8M时钟也可用作PLL输入时钟。

工厂会校准IRC8M时钟频率的精度，但是它的精度仍然比HXTAL时钟要差。用户需求、环境条件和成本将决定选择哪个时钟作为系统时钟源。

如果HXTAL或者PLL是系统时钟源，为了最大程度减小系统从深度睡眠模式启动的时间，系统从深度睡眠模式初始唤醒的时候硬件强制IRC8M时钟作为系统时钟。

锁相环 (PLL)

内部锁相环PLL通过对输入参考频率为4~32MHz的时钟基准2~64倍频，可以提供16~108 MHz的时钟输出。

PLL可以通过设置时钟控制寄存器0(RCU_CTL0)中的PLLEN位被启动和关闭。时钟控制寄存器RCU_CTL0中的PLLSTB位用来指示PLL时钟是否稳定。如果时钟中断寄存器RCU_INT中的相应中断使能位PLLSTBIE被置‘1’，在PLL稳定以后，将产生一个中断。

高速内部28MHz RC振荡器时钟 (IRC28M)

高速内部28MHz RC振荡器时钟 (IRC28M)有一个固定的频率28MHz，专门用作ADC时钟。IRC28M可以通过设置时钟控制寄存器1(RCU_CTL1)中的IRC28MEN位被启动和关闭。时钟控制寄存器1(RCU_CTL1)中的IRC28MSTB位用来指示IRC28M时钟是否已稳定。如果时钟中断寄存器RCU_INT中的相应中断使能位IRC28MSTBIE被置‘1’，在IRC28M稳定以后，将产生一个中断。

高速内部48MHz RC振荡器时钟 (IRC48M)

高速内部48M RC振荡器时钟(IRC48M)有一个固定的频率48MHz，用作USB时钟或者PLL时钟源。IRC48M可以通过设置附加控制寄存器(RCU_ADDCTL)中的IRC48MEN位被启动和关闭。附加控制寄存器(RCU_ADDCTL)中的IRC48MSTB位用来指示IRC48M时钟是否已稳定。如果附加控制寄存器RCU_ADDCTL中的相应中断使能位IRC48MSTBIE被置‘1’，在IRC48M稳定以后，将产生一个中断。

工厂会校准IRC48M时钟频率的精度，但其工作频率仍不够准确，因为USB需要在48MHz波动500ppm之内的频率精度。在CTC单元中硬件自动动态调整IRC48M来满足频率需求。

低速外部晶体振荡器时钟(LXTAL)

LXTAL晶体是一个32.768kHz的低速外部晶体或陶瓷谐振器。它为实时时钟电路提供一个低功耗且精确的时钟源。LXTAL时钟可以通过设置备份域控制寄存器(RCU_BDCTL)中的LXTALEN位被启动和关闭。备份域控制寄存器RCU_BDCTL中的LXTALSTB位用来指示LXTAL时钟是否稳定。如果时钟中断寄存器RCU_INT中的相应中断使能位LXTALSTBIE被置‘1’，在LXTAL稳定

以后，将产生一个中断。

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

低速内部RC振荡器时钟(IRC40K)

IRC40K RC振荡器时钟担当一个低功耗时钟源的角色，它的时钟频率大约40 kHz，为独立看门狗定时器和实时时钟电路提供时钟。IRC40K提供低成本的时钟源，因为不需要外部器件。IRC40K RC振荡器可以通过设置控制/状态寄存器RCU_RSTSCK中的IRC40KEN位被启动和关闭。控制/状态寄存器RCU_RSTSCK中的IRC40KSTB位用来指示IRC40K时钟是否已稳定。如果时钟中断寄存器RCU_INT中的相应中断使能位IRC40KSTBIE被置‘1’，在IRC40K稳定以后，将产生一个中断。

系统时钟 (CK_SYS) 选择

系统复位后，IRC8M时钟被选为系统时钟，改变时钟配置寄存器RCU_CFG0 中的系统时钟转换位SCS可以切换系统时钟源为HXTAL或PLL。当SCS的值改变，系统时钟将使用原来的时钟源继续运行直到转换的目标时钟源稳定。当一个时钟源被直接或通过PLL间接作为系统时钟时，它将不能被停止。

HXTAL 时钟监视器(CKM)

设置时钟控制寄存器RCU_CTL0中的HXTAL时钟监视使能位CKMEN，HXTAL可以使能时钟监视功能。该功能必须在HXTAL启动延迟完毕后使能，在HXTAL停止后禁止。一旦监测到HXTAL故障，HXTAL将自动被禁止，时钟中断寄存器RCU_INT中的HXTAL时钟阻塞标志位CKMIF将被置‘1’，产生HXTAL故障事件。这个故障引发的中断和Cortex-M4的不可屏蔽中断相连。如果HXTAL被选作系统或PLL的时钟源，HXTAL故障将促使选择IRC8M为系统时钟源且PLL将被自动禁止。

时钟输出功能

时钟输出功能输出从32kHz到108MHz的时钟。通过设置时钟配置寄存器RCU_CFG0中的CK_OUT时钟源选择位CKOUTSEL能够选择不同的时钟信号。相应的GPIO引脚应该被配置成复用功能I/O(AFIO)模式来输出选择的时钟信号。

表 4-1. 时钟源的选择

时钟源选择位	时钟源
000	无时钟
001	CK_IRC28M
010	CK_IRC40K
011	CK_LXTAL
100	CK_SYS
101	CK_IRC8M
110	CK_HXTAL
111	CK_PLL 或 CK_PLL/2

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

深度睡眠模式时钟控制

当MCU工作在深度睡眠模式时，HDMI CEC或USART0能唤醒MCU，前提是它们的时钟是由LXTAL时钟提供且LXTAL时钟被使能。

如果HDMI CEC或USART0时钟选择IRC8M时钟并且工作在深度睡眠模式，它们有能力开启或关闭IRC8M时钟，HDMI CEC或USART0采用IRC8M时钟作为工作时钟来唤醒深度睡眠模式。

电压控制

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

表 4-2. 深度睡眠模式下内核电压选择

DSLPVS[1:0]	深度睡眠模式电压(V)
00	1.0
01	0.9
10	0.8
11	0.7

RCU_DSV 寄存器被电源解锁寄存器(RCU_VKEY)保护。只有在写0x1A2B3C4D到RCU_VKEY后，RCU_DSV寄存器才能被写入。

4.3. RCU 寄存器

4.3.1. 控制寄存器 0 (RCU_CTL0)

地址偏移: 0x00

复位值: 0x0000 XX83 X表示未定义。

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		PLLSTB		PLLEN		保留		CKMEN		HXTALBPS		HXTALSTB		HXTALEN	
						r	rw				rw	rw		r	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRC8MCALIB[7:0]						IRC8MADJ[4:0]				保留.		IRC8MSTB		IRC8MEN	
													r		rw

位/位域	名称	描述
31:26	保留	必须保持复位值
25	PLLSTB	<p>PLL 时钟稳定标志位</p> <p>硬件置‘1’来指示PLL输出时钟是否稳定待用。</p> <p>0: PLL没稳定</p> <p>1: PLL稳定</p>

24	PLLEN	PLL 使能 软件置位或复位。如果PLL时钟作为系统时钟的时候该位不能被复位。进入深度睡眠或待机模式时硬件自动复位。 0: PLL被关闭 1: PLL被打开
23:20	保留	必须保持复位值.
19	CKMEN	HXTAL时钟监视使能 0: 禁止外部4 ~ 32 MHz 晶体振荡器(HXTAL) 时钟监视器 1: 使能外部4 ~ 32 MHz 晶体振荡器(HXTAL) 时钟监视器 当硬件监测到HXTAL时钟一直停留在低或者高的状态，内部硬件将切换系统时钟到IRC8M RC时钟。恢复原来系统时钟的方式有以下几种：外部复位，上电复位，软件清CKMIF位。 注意： 使能HXTAL时钟监视器以后，硬件无视控制位IRC8MEN的状态，自动使能IRC8M时钟。
18	HXTALBPS	外部晶体振荡器(HXTAL)时钟旁路模式使能 只有在HXTALEN 位为0时HXTALBPS位才可写。 0: 禁止HXTAL旁路模式 1: 使能HXTAL旁路模式 HXTAL输出时钟等于输入时钟
17	HXTALSTB	外部晶体振荡器(HXTAL)时钟稳定状态标志位 硬件置‘1’来指示HXTAL振荡器时钟是否稳定待用。 0: HXTAL 振荡器未稳定 1: HXTAL 振荡器已稳定
16	HXTALEN	外部高速振荡器时钟使能 软件置‘1’或清‘0’。如果HXTAL时钟或者PLL输入时钟作为系统时钟，该位不能被复位。进入深度睡眠或待机模式时硬件自动复位。 0: 禁止外部4 ~ 32 MHz晶体振荡器 1: 使能外部4 ~ 32 MHz 晶体振荡器
15:8	IRC8MCALIB[7:0]	高速内部振荡器校准值寄存器 上电时自动加载这些位
7:3	IRC8MADJ[4:0]	高速内部振荡器时钟调整值 这些位由软件置位，最终调整值为IRC8MADJ当前值加上IRC8MCALIB[7:0]位的值。最终调整值应该调整IRC8M到8 MHz ± 1%。
2	保留	必须保持复位值.
1	IRC8MSTB	高速内部(IRC8M)时钟稳定状态标志位 硬件置‘1’来指示IRC8M振荡器时钟是否稳定待用。 0: IRC8M 振荡器未稳定 1: IRC8M 振荡器已稳定
0	IRC8MEN	高速内部振荡器使能 软件复位置位。如果IRC8M时钟用作系统时钟时该位不能被复位。当从待机和深

度睡眠模式返回或用作系统时钟的HXTAL振荡器发生故障时，该位由硬件置1来启动IRC8M振荡器。

- 0: 内部8 MHz RC振荡器关闭
- 1: 内部8 MHz RC振荡器开启

4.3.2. 配置寄存器 0 (RCU_CFG0)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PLLDV	CKOUTDIV[2:0]	PLLMF[4]	CKOUTSEL[2:0]	USBFS_PSC[1:0]	PLLMF[3:0]	PLL_PREDV	PLLSEL								
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCPSC[1:0]	APB2PSC[2:0]	APB1PSC[2:0]	AHB_PSC[3:0]	SCSS[1:0]	SCS[1:0]										
rw	rw	rw	rw	rw	rw	r	rw								

位/位域	名称	描述
31	PLLDV	CK_PLL 1或2分频来用作CK_OUT 0: CK_PLL 2分频用作CK_OUT 1: CK_PLL用作CK_OUT
30:28	CKOUTDIV[2:0]	CK_OUT分频器, 来降低CK_OUT频率 CK_OUT的选择参考RCU_CFG0的26:24位。 000: CK_OUT不分频 001: CK_OUT 2分频 010: CK_OUT 4分频 011: CK_OUT 8分频 100: CK_OUT 16分频 101: CK_OUT 32分频 110: CK_OUT 64分频 111: CK_OUT 128分频
27	PLLMF[4]	PLLMF 寄存器第4位 见RCU_CFG0的21:18位
26:24	CKOUTSEL[2:0]	CK_OUT时钟源选择 软件置位或清零。 000: 没有时钟被选择 001: 选择内部28M RC振荡器时钟 010: 选择内部40K RC振荡器时钟 011: 选择外部低速振荡器时钟 100: 选择系统时钟

		101: 选择内部8M RC振荡器时钟 110: 选择外部高速振荡器时钟 111: 依赖于PLLDV选择(CK_PLL / 2)或CK_PLL
23:22	USBFSPSC[1:0]	USBFS时钟预分频选择 软件写这两位和RCU_CFG2的第30位来确定USBFS的时钟分频。软件置位或清零来控制USBFS时钟预分频值。USBFS时钟必须为48MHz.。如果USBFS时钟使能这些位不能被复位。 000: 选择(CK_PLL / 1.5) 001: 选择CK_PLL 010: 选择(CK_PLL / 2.5) 011: 选择(CK_PLL / 2) 100: 选择(CK_PLL / 3) 101/110/111:选择(CK_PLL / 3.5)
21:18	PLLMF[3:0]	PLL倍频因子 软件写这些位包括RCU_CFG0的27位和RCU_CFG1的31位来确定PLL的倍频因子。 00000: (PLL时钟源 x 2) 00001: (PLL时钟源 x 3) 00010: (PLL时钟源 x 4) 00011: (PLL时钟源 x 5) 00100: (PLL时钟源 x 6) 00101: (PLL时钟源 x 7) 00110: (PLL时钟源 x 8) 00111: (PLL时钟源 x 9) 01000: (PLL时钟源 x 10) 01001: (PLL时钟源 x 11) 01010: (PLL时钟源 x 12) 01011: (PLL时钟源 x 13) 01100: (PLL时钟源 x 14) 01101: (PLL时钟源 x 15) 01110: (PLL时钟源 x 16) 01111: (PLL时钟源 x 16) 10000: (PLL时钟源 x 17) 10001: (PLL时钟源 x 18) 10010: (PLL时钟源 x 19) 10011: (PLL时钟源 x 20) 10100: (PLL时钟源 x 21) 10101: (PLL时钟源 x 22) 10110: (PLL时钟源 x 23) 10111: (PLL时钟源 x 24) 11000: (PLL时钟源 x 25) 11001: (PLL时钟源 x 26) 11010: (PLL时钟源 x 27) 11011: (PLL时钟源 x 28)

		11100: (PLL时钟源 x 29) 11101: (PLL时钟源 x 30) 11110: (PLL时钟源 x 31) 11111: (PLL时钟源 x 32) 100000: (PLL时钟源 x 33) ... 111110: (PLL时钟源 x 63) 111111: (PLL时钟源 x 64) 注意: PLL输出频率不能超过108MHz。
17	PLLPREDV	HXTAL或CK_IRC48M作为PLL输入。该位与时钟配置寄存器1(RCU_CFG1)中的PREDV[0]位是一样的。参考RCU_CFG1的PREDV位说明。 由软件置1或清0来分频或不分频后，当作PLL输入时钟源。 0: 选择HXTAL或CK_IRC48M时钟 1: HXTAL或CK_IRC48M时钟二分频
16	PLLSEL	PLL 时钟源选择 软件置1或清0来控制PLL时钟源 0: 选择IRC8M二分频为PLL时钟源 1: 选择HXTAL或者IRC48M(RCU_CFG1寄存器的PLLPRESEL位)为PLL时钟源
15:14	ADCPSC[1:0]	ADC时钟预分频选择 软件写三位包括RCU_CFG2的31位来确定ADC时钟分频。 软件清0和置1。 000: 选择APB2时钟2分频 001: 选择APB2时钟4分频 010: 选择APB2时钟6分频 011: 选择APB2时钟8分频 100: 选择APB2时钟3分频 101: 选择APB2时钟5分频 110: 选择APB2时钟7分频 111: 选择APB2时钟9分频
13:11	APB2PSC[2:0]	APB2预分频选择 软件置1和清0来控制APB2时钟分频因子。 0xx: 选择AHB时钟不分频 100: 选择AHB时钟2分频 101: 选择AHB时钟4分频 110: 选择AHB时钟8分频 111: 选择AHB时钟16分频
10:8	APB1PSC[2:0]	APB1 预分频选择 软件设置和清除来控制APB1时钟分频因子。 0xx: 选择AHB时钟不分频 100: 选择AHB时钟2分频 101: 选择AHB时钟4分频

		110: 选择AHB时钟8分频
		111: 选择AHB时钟16分频
7:4	AHBPSC[3:0]	<p>AHB预分频选择</p> <p>软件设置和清除来控制AHB时钟分频因子。</p> <p>0xxx: 选择CK_SYS系统时钟不分频</p> <p>1000: 选择CK_SYS系统时钟2分频</p> <p>1001: 选择CK_SYS系统时钟4分频</p> <p>1010: 选择CK_SYS系统时钟8分频</p> <p>1011: 选择CK_SYS系统时钟16分频</p> <p>1100: 选择CK_SYS系统时钟64分频</p> <p>1101: 选择CK_SYS系统时钟128分频</p> <p>1110: 选择CK_SYS系统时钟256分频</p> <p>1111: 选择CK_SYS系统时钟512分频</p>
3:2	SCSS[1:0]	<p>系统时钟转换状态</p> <p>硬件设置和清除指示系统当前时钟源</p> <p>00: 选择CK_IRC8M作为CK_SYS 系统时钟源</p> <p>01: 选择CK_HXTAL作为CK_SYS 系统时钟源</p> <p>10: 选择CK_PLL作为CK_SYS 系统时钟源</p> <p>11: 保留</p>
1:0	SCS[1:0]	<p>系统时钟转换</p> <p>软件设置选择系统时钟源。由于CK_SYS的改变有固有的延迟，需要软件读SCSS位来确保转换是否结束。在从深度睡眠或待机模式中返回时，或作为系统时钟或PLL时钟源的HXTAL出现故障时，强制选择IRC8M作为系统时钟或PLL时钟。</p> <p>00: 选择IRC8M时钟作为CK_SYS系统时钟源</p> <p>01: 选择HXTAL时钟作为CK_SYS系统时钟源</p> <p>10: 选择PLL作为CK_SYS系统时钟源</p> <p>11: 保留</p>

4.3.3. 中断寄存器 (RCU_INT)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								CKMIC	保留	IRC28M	PLL	HXTAL	IRC8M	LXTAL	IRC40K
									STBIC	STBIC	STBIC	STBIC	STBIC	STBIC	STBIC
										w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	IRC28M STBIE	PLL STBIE	HXTAL STBIE	IRC8M STBIE	LXTAL STBIE	IRC40K STBIE	CKMIF	保留	IRC28M STBIF	PLL STBIF	HXTAL STBIF	IRC8M STBIF	LXTAL STBIF	IRC40K STBIF	
	rw	rw	rw	rw	rw	rw	r		r	r	r	r	r	r	r

位/位域	名称	描述
31:24	保留	必须保持复位值
23	CKMIC	HXTAL时钟阻塞中断清除 软件写1复位CKMIF标志位。 0: 不复位CKMIF标志位 1: 复位CKMIF标志位
22	保留	必须保持复位值
21	IRC28MSTBIC	IRC28M时钟稳定中断清除 软件写1复位IRC28MSTBIF标志位。 0: 不复位IRC28MSTBIF标志位 1: 复位IRC28MSTBIF标志位
20	PLLSTBIC	PLL稳定中断清除 软件写1复位PLLSTBIF标志位。 0: 不复位PLLSTBIF标志位 1: 复位PLLSTBIF标志位
19	HXTALSTBIC	HXTAL时钟稳定中断清除 软件写1复位HXTALSTBIF标志位。 0: 不复位HXTALSTBIF标志位 1: 复位HXTALSTBIF标志位
18	IRC8MSTBIC	IRC8M时钟稳定中断清除 软件写1复位IRC8MSTBIF标志位。 0: 不复位IRC8MSTBIF标志位 1: 复位IRC8MSTBIF标志位
17	LXTALSTBIC	LXTAL时钟稳定中断清除 软件写1复位LXTALSTBIF标志位。 0: 不复位LXTALSTBIF标志位 1: 复位LXTALRDYF标志位
16	IRC40KSTBIC	IRC40K时钟稳定中断清除 软件写1复位IRC40KSTBIF标志位。 0: 不复位IRC40KSTBIF标志位 1: 复位IRC40KSTBIF标志位
15:14	保留	必须保持复位值
13	IRC28MSTBIE	IRC28M时钟稳定中断使能 软件置1和清0来使能/禁止IRC28M时钟稳定中断。 0: 禁止IRC28M时钟稳定中断 1: 使能IRC28M时钟稳定中断
12	PLLSTBIE	PLL时钟稳定中断使能

		软件置1和清0来使能/禁止PLL时钟稳定中断。 0: 禁止PLL时钟稳定中断 1: 使能PLL时钟稳定中断
11	HXTALSTBIE	HXTAL时钟稳定中断使能 软件置1和清0来使能/禁止HXTAL时钟稳定中断。 0: 禁止HXTAL时钟稳定中断 1: 使能HXTAL时钟稳定中断
10	IRC8MSTBIE	IRC8M时钟稳定中断使能 软件置1和清0来使能/禁止IRC8M时钟稳定中断。 0: 禁止IRC8M时钟稳定中断 1: 使能IRC8M时钟稳定中断
9	LXTALSTBIE	LXTAL时钟稳定中断使能 LXTAL时钟稳定中断使能/禁止控制。 0: 禁止LXTAL时钟稳定中断 1: 使能LXTAL时钟稳定中断
8	IRC40KSTBIE	IRC40K时钟稳定中断使能 IRC40K时钟稳定中断使能/禁止控制。 0: 禁止IRC40K时钟稳定中断 1: 使能IRC40K时钟稳定中断
7	CKMIF	HXTAL时钟阻塞中断标志位 当HXTAL时钟阻塞时硬件置1。 软件置CKMIC=1时清除该位。 0: 时钟运行正常 1: HXTAL时钟阻塞
6	保留	必须保持复位值.
5	IRC28MSTBIF	IRC28M 时钟稳定中断标志位 当IRC28M时钟稳定且IRC28MSTBIE位被置1时由硬件置1。 软件置IRC14MSTBIC=1时清除该位。 0: 无IRC28M时钟稳定中断产生 1: IRC28M时钟稳定中断发生
4	PLLSTBIF	PLL 时钟稳定中断标志位 当PLL时钟稳定且PLLSTBIE位被置1时由硬件置1。 软件置PLLSTBIC=1时清除该位。 0: 无PLL时钟稳定中断产生 1: 产生PLL时钟稳定中断
3	HXTALSTBIF	HXTAL时钟稳定中断标志位 当外部4 ~ 32 MHz晶体振荡器时钟稳定且HXTALSTBIE位被置1时由硬件置1。 软件置HXTALSTBIC=1时清除该位。 0: 无HXTAL时钟稳定中断发生

		1: 发生HXTAL时钟稳定中断
2	IRC8MSTBIF	IRC8M时钟稳定中断标志位 当内部8 MHz RC振荡器时钟稳定且IRC8MSTBIE位被置1时由硬件置1。 软件置IRC8MSTBIC=1时清除该位。 0: 无IRC8M时钟稳定中断产生 1: 产生IRC8M时钟稳定中断
1	LXTALSTBIF	LXTAL 时钟稳定中断标志位 当外部32.768KHz晶体振荡器时钟稳定且LXTALSTBIE为被置1时由硬件置1。 软件置LXTALSTBIC=1时清除该位。 0: 无LXTAL时钟稳定中断发生 1: 发生LXTAL时钟稳定中断
0	IRC40KSTBIF	IRC40K时钟稳定中断标志位 当内部32kHz RC振荡器时钟稳定且IRC40KSTBIE位被置1时由硬件置1。 软件置IRC40KSTBIC =1时清除该位。 0: 无IRC40K时钟稳定中断产生 1: 产生IRC40K时钟稳定中断

4.3.4. APB2 复位寄存器 (RCU_APB2RST)

地址偏移: 0x0C

复位值: 0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留										TIMER16	TIMER15	TIMER14			
										RST	RST	RST			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	USART0 RST	保留.	SPI0 RST	TIMER0 RST	保留.	ADC RST				保留			CFGCMP RST		
	rw		rw	rw		rw							rw		

位/位域	名称	描述
31:19	保留	必须保持复位值
18	TIMER16RST	TIMER 16定时器复位 由软件置1或清0。 0: 无复位 1: 复位TIMER16定时器
17	TIMER15RST	TIMER 15定时器复位 由软件置1或清0。

		0: 无复位 1: 复位TIMER15定时器
16	TIMER14RST	TIMER 14定时器复位 由软件置1或清0。 0: 无复位 1: 复位TIMER14定时器
14	USART0RST	USART0 复位 由软件置1或清0。 0: 无复位 1: 复位USART0
13	保留	必须保持复位值
12	SPI0RST	SPI0复位 由软件置1或清0。 0: 无复位 1: 复位SPI0
11	TIMER0RST	TIMER0定时器复位 由软件置1或清0。 0: 无复位 1: 复位TIMER0定时器
10	保留	必须保持复位值
9	ADCRST	ADC复位 由软件置1或清0。 0: 无复位 1: 复位ADC
8:1	保留	必须保持复位值
0	CFGCMPRST	系统配置和比较器复位 由软件置1或清0。 0: 无复位 1: 复位系统配置和比较器模块

4.3.5. APB1 复位寄存器 (RCU_APB1RST)

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	CEC RST	DAC RST	PMU RST	保留	I2C1 RST	I2C0 RST	保留	USART1 RST	保留.						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
保留	SPI1 RST	保留	WWDGT RST	保留	TIMER13 RST	保留	TIMER5 RST	保留	TIMER2 RST	TIMER1 RST							
	rw		rw		rw		rw		rw		rw		rw		rw		

位/位域	名称	描述
31	保留	必须保持复位值
30	CECRST	HDMI CEC复位 由软件置1或清0。 0: 无复位 1: 复位HDMI CEC单元
29	DACRST	DAC复位 由软件置1或清0。 0: 无复位 1: 复位DAC单元
28	PMURST	电源控制复位 由软件置1或清0。 0: 无复位 1: 复位电源控制单元
27:23	保留	必须保持复位值
22	I2C1RST	I2C1复位 由软件置1或清0。 0: 无复位 1: 复位I2C1
21	I2C0RST	I2C0复位 由软件置1或清0。 0: 无复位 1: 复位I2C0
20:18	保留	必须保持复位值
17	USART1RST	USART1复位 由软件置1或清0。 0: 无复位 1: 复位USART1
16:15	保留	必须保持复位值
14	SPI1RST	SPI1复位 由软件置1或清0。 0: 无复位

		1: 复位SPI1
13:12	保留	必须保持复位值
11	WWDGTRST	窗口看门狗定时器复位 由软件置1或清0。 0: 无复位 1: 复位窗口看门狗定时器
10:9	保留	必须保持复位值
8	TIMER13RST	TIMER13定时器复位 由软件置1或清0。 0: 无复位 1: 复位TIMER13定时器
7:5	保留	必须保持复位值
4	TIMER5RST	TIMER5定时器复位 由软件置1或清0。 0: 无复位 1: 复位TIMER5定时器
3:2	保留	必须保持复位值
1	TIMER2RST	TIMER2定时器复位 由软件置1或清0。 0: 无复位 1: 复位TIMER2定时器
0	TIMER1RST	TIMER1定时器复位 由软件置1或清0。 0: 无复位 1: 复位TIMER1定时器

4.3.6. AHB 使能寄存器 (RCU_AHBEN)

地址偏移: 0x14

复位值: 0x0000 0014

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			保留				TSIEN	保留.	PFEN	保留.	PDEN	PCEN	PBEN	PAEN	保留.
								rw		rw		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
.	保留	USBFSEN		保留			CRCEN	保留.	FMCEN	保留.	SRAMEN	保留.	DMAEN		
								rw		rw		rw		rw	

位/位域	名称	描述
31:25	保留	必须保持复位值
24	TSIEN	TSI时钟使能 由软件置1或清0。 0: TSI时钟关闭 1: TSI时钟开启
23	保留	必须保持复位值
22	PFEN	GPIOF时钟使能 由软件置1或清0。 0: GPIOF时钟关闭 1: GPIOF时钟开启
21	保留	必须保持复位值
20	PDEN	GPIOD时钟使能 由软件置1或清0。 0: GPIOD时钟关闭 1: GPIOD时钟开启
19	PCEN	GPIOC时钟使能 由软件置1或清0。 0: GPIOC时钟关闭 1: GPIOC时钟开启
18	PBEN	GPIOB时钟使能 由软件置1或清0。 0: GPIOB时钟关闭 1: GPIOB时钟开启
17	PAEN	GPIOA时钟使能 由软件置1或清0。 0: GPIOA时钟关闭 1: GPIOA时钟开启
16:13	保留	必须保持复位值
12	USBFSEN	USBFS 时钟使能 由软件置1或清0。 0: USBFS时钟关闭 1: USBFS时钟开启
11:7	保留	必须保持复位值
6	CRCEN	CRC 时钟使能 由软件置1或清0。 0: CRC时钟关闭

		1: CRC时钟开启
5	保留	必须保持复位值
4	FMCSPEN	<p>FMC时钟使能 由软件置1或清0来开启/关闭在睡眠模式下的FMC时钟。</p> <p>0: 关闭睡眠模式下的FMC时钟 1: 开启睡眠模式下的FMC时钟</p>
3	保留	必须保持复位值
2	SRAMSPEN	<p>SRAM接口时钟使能 由软件置1或清0来开启/关闭在睡眠模式下的SRAM时钟。</p> <p>0: 关闭睡眠模式下的SRAM接口时钟 1: 开启睡眠模式下的SRAM接口时钟</p>
1	保留	必须保持复位值
0	DMAEN	<p>DMA时钟使能 由软件置1或清0。</p> <p>0: 关闭DMA时钟 1: 开启DMA时钟</p>

4.3.7. APB2 使能寄存器 (RCU_APB2EN)

地址偏移: 0x18

复位值: 0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留														TIMER16EN	TIMER15EN	TIMER14EN
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
保留	USART0EN	保留	SPI0EN	TIMEROEN	保留	ADCEN	保留						CFGCMP EN			
	rw		rw	rw		rw										

位/位域	名称	描述
31:19	保留	必须保持复位值
18	TIMER16EN	<p>TIMER16定时器时钟使能 由软件置1或清0。</p> <p>0: 关闭TIMER16定时器时钟 1: 开启TIMER16定时器时钟</p>
17	TIMER15EN	<p>TIMER15定时器时钟使能 由软件置1或清0。</p>

		0: 关闭TIMER15定时器时钟 1: 开启TIMER15定时器时钟
16	TIMER14EN	TIMER14定时器时钟使能 由软件置1或清0。 0: 关闭TIMER14定时器时钟 1: 开启TIMER14定时器时钟
15	保留	必须保持复位值
14	USART0EN	USART0时钟使能 由软件置1或清0。 0: 关闭USART0时钟 1: 开启USART0时钟
13	保留	必须保持复位值
12	SPI0EN	SPI0时钟使能 由软件置1或清0。 0: 关闭SPI0时钟 1: 开启SPI0时钟
11	TIMER0EN	TIMER0定时器时钟使能 由软件置1或清0。 0: 关闭TIMER0定时器时钟 1: 开启TIMER0定时器时钟
10	保留	必须保持复位值
9	ADCEN	ADC 接口时钟使能 由软件置1或清0。 0: 关闭ADC接口时钟 1: 开启ADC接口时钟
8:1	保留	必须保持复位值
0	CFGCMPEN	系统配置与比较器时钟使能 由软件置1或清0。 0: 关闭系统配置与比较器模块时钟 1: 开启系统配置与比较器模块时钟

4.3.8. APB1 使能寄存器 (RCU_APB1EN)

地址偏移: 0x1C

复位值: 0x0000 0000

该寄存器可以按字节 (8位)、半字 (16位) 或字 (32位) 访问

保留	CECEN	DACEN	PMUEN	保留						I2C1EN	I2C0EN	保留			USART1EN	保留
	rw	rw	rw							rw	rw				rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
保留	SPI1EN	保留		WWDGTE	保留	TIMER13EN	保留			TIMER5EN	保留			TIMER2EN	TIMER1EN	
	rw	rw			rw		rw			rw	rw			rw	rw	

位/位域	名称	描述
31	保留	必须保持复位值
30	CECEN	HDMI CEC接口时钟使能 由软件置1或清0。 0: 关闭HDMI CEC接口时钟 1: 开启HDMI CEC接口时钟
29	DACEN	DAC接口时钟使能 由软件置1或清0。 0: 关闭DAC接口时钟 1: 开启DAC接口时钟
28	PMUEN	电源接口时钟使能 由软件置1或清0。 0: 关闭电源接口时钟 1: 开启电源接口时钟
27:23	保留	必须保持复位值
22	I2C1EN	I2C1时钟使能 由软件置1或清0。 0: 关闭I2C1时钟 1: 开启I2C1时钟
21	I2C0EN	I2C0时钟使能 由软件置1或清0。 0: 关闭I2C0时钟 1: 开启I2C0时钟
20:18	保留	必须保持复位值
17	USART1EN	USART1时钟使能 由软件置1或清0。 0:关闭USART1时钟 1:开启USART1时钟
16:15	保留	必须保持复位值
14	SPI1EN	SPI1时钟使能 由软件置1或清0。 0: 关闭SPI1时钟

		1: 开启SPI1时钟
13:12	保留	必须保持复位值
11	WWDGTE	窗口看门狗定时器时钟使能 由软件置1或清0。 0: 关闭窗口看门狗定时器时钟 1: 开启窗口看门狗定时器时钟
10:9	保留	必须保持复位值
8	TIMER13EN	TIMER13定时器时钟使能 由软件置1或清0。 0: 关闭TIMER13定时器时钟 1: 开启TIMER13定时器时钟
7:5	保留	必须保持复位值
4	TIMER5EN	TIMER5定时器时钟使能 由软件置1或清0。 0: 关闭TIMER5定时器时钟 1: 开启TIMER5定时器时钟
3:2	保留	必须保持复位值
1	TIMER2EN	TIMER2定时器时钟使能 由软件置1或清0。 0: 关闭TIMER2定时器时钟 1: 开启TIMER2定时器时钟
0	TIMER1EN	TIMER1定时器时钟使能 由软件置1或清0。 0: 关闭TIMER1定时器时钟 1: 开启TIMER1定时器时钟

4.3.9. 备份域控制寄存器 (RCU_BDCTL)

地址偏移: 0x20

复位值: 0x0000 0018, 由备份域复位电路复位

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

注意: 备份域控制寄存器(BDCTL)的LXTALEN, LXTALBPS, RTCSRC和RTCEN位仅在备份域复位后才清0。只有在电源控制寄存器(PMU_CTL)中的BKPRST位置1后才能对这些位进行改动。

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

rw

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTCEN	保留				RTCSRC[1:0]		保留				LXTALDRI[1:0]	LXTALBPS	LXTALSTB	LXTALEN	
		rw					rw					rw	rw	r	rw	

位/位域	名称	描述
31:17	保留	必须保持复位值
16	BKPRST	<p>备份域复位 由软件置1或清0。</p> <p>0: 无复位 1: 复位备份域</p>
15	RTCEN	<p>RTC时钟使能 由软件置1或清0。</p> <p>0: 关闭RTC时钟 1: 开启RTC时钟</p>
14:10	保留	必须保持复位值
9:8	RTCSRC[1:0]	<p>RTC时钟入口选择 软件置位或清除来控制RTC时钟源。</p> <p>00: 没有时钟 01: 选择LXTAL时钟作为RTC时钟源 10: 选择IRC40K时钟作为RTC时钟源 11: 选择HXTAL时钟32分频作为RTC时钟源</p>
7:5	保留	必须保持复位值
4:3	LXTALDRI[1:0]	<p>LXTAL驱动能力 软件置位或清除。当复位备份域时，会重装载缺省值。</p> <p>00: 弱驱动能力 01: 中低驱动能力 10: 中高驱动能力 11: 强驱动能力 (复位后的缺省值)</p> <p>注: LXTALDRI在旁路模式下无效</p>
2	LXTALBPS	<p>LXTAL旁路模式使能 软件置1和清0。</p> <p>0: 禁止LXTAL旁路模式 1: 使能LXTAL旁路模式</p>
1	LXTALSTB	<p>外部低速振荡器稳定状态位 硬件置1来指示LXTAL输出时钟是否稳定待用。</p> <p>0: LXTAL未稳定 1: LXTAL已稳定</p>
0	LXTALEN	LXTAL使能

软件置1和清0。

0: 关闭LXTAL

1: 开启LXTAL

4.3.10. 复位源/时钟寄存器 (RCU_RSTSCK)

地址偏移: 0x24

复位值: 0x0C00 0000, 除复位标志外由系统复位清除, 复位标志只能由电源复位清除。

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LP RSTF	WWDGT RSTF	FWDGT RSTF	SW RSTF	POR RSTF	EP RSTF	OBL RSTF	RSTFC RSTF	V12 RSTF	保留	保留	保留	保留	保留	保留	保留
r	r	r	r	r	r	r	rw	r							rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												IRC40K STB	IRC40K EN		
												r	rw		

位/位域	名称	描述
31	LPRSTF	低功耗复位标志位 深度睡眠/待机复位发生时由硬件置1。 由软件通过写RSTFC位来清除该位。 0: 无低功耗管理复位发生 1: 发生低功耗管理复位
30	WWDGTRSTF	窗口看门狗定时器复位标志位 窗口看门狗定时器复位发生时由硬件置1。 由软件通过写RSTFC位来清除该位。 0: 无窗口看门狗定时器复位发生 1: 发生窗口看门狗定时器复位
29	FWDGTRSTF	独立看门狗定时器复位标志位 独立看门狗复位发生时由硬件置1。 由软件通过写RSTFC位来清除该位。 0: 无独立看门狗定时器复位发生 1: 发生独立看门狗定时器复位
28	SWRSTF	软件复位标志位 软件复位发生时由硬件置1。 由软件通过写RSTFC位来清除该位。 0: 无软件复位发生 1: 发生软件复位
27	PORRSTF	电源复位标志位

			电源复位发生时由硬件置1。 由软件通过写RSTFC位来清除该位。 0: 无电源复位发生 1: 发生电源复位
26	EPRSTF	外部引脚复位标志位	当有外部引脚复位发生时由硬件置1。 由软件通过写RSTFC位来清除该位。 0: 无外部引脚复位发生 1: 发生外部引脚复位
25	OBLRSTF	可选字节装载器复位标志位	可选字节装载器装载字节时由硬件置1。 由软件通过写RSTFC位来清除该位。 0: 无选项字节装载器复位发生 1: 发生选项字节装载器复位
24	RSTFC	清除复位标志位	由软件置1来清除所有复位标志位。 0: 无作用 1: 清除复位标志位
23	V12RSTF	V12域电源复位标志位	当有V12域电源复位发生时由硬件置1。 由软件通过写RSTFC位来清除该位。 0: 无V12域电源复位发生 1: 发生V12域电源复位
22:2	保留	必须保持复位值.	
1	IRC40KSTB	IRC40K时钟稳定状态位	该位由硬件置1指示IRC40K输出时钟是否稳定待用。 0: IRC40K时钟未稳定 1: IRC40K时钟已稳定
0	IRC40KEN	IRC40K时钟使能	软件置1和清0。 0: 关闭IRC40K时钟 1: 开启IRC40K时钟

4.3.11. AHB 复位寄存器 (RCU_AHBRST)

地址偏移: 0x28

复位值: 0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问

保留								TSIRST	保留	PFRST	保留	PDRST	PCRST	PBRST	PARST	保留
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
保留	USBFSRST	保留														
rw																

位/位域	名称	描述
31:25	保留	必须保持复位值
24	TSIRST	TSI单元复位 由软件置1或清0。 0: 无作用 1: 复位TSI单元
23	保留	必须保持复位值
22	PFRST	GPIOF复位 由软件置1或清0。 0: 无作用 1: 复位GPIOF口
21	保留	必须保持复位值
20	PDRST	GPIOD复位 由软件置1或清0。 0: 无作用 1: 复位GPIOD口
19	PCRST	GPIOC复位 由软件置1或清0。 0: 无作用 1: 复位GPIOC口
18	PBRST	GPIOB复位 由软件置1或清0。 0: 无作用 1: 复位GPIOB口
17	PARST	GPIOA复位 由软件置1或清0。 0: 无作用 1: 复位GPIOA口
16:3	保留	必须保持复位值
12	USBFSRST	USBFS复位 由软件置1或清0。 0: 无作用

1: 复位USBFS口

11:0 保留 必须保持复位值

4.3.12. 配置寄存器 1 (RCU_CFG1)

地址偏移: 0x2c

复位值: 0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问

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

位/位域	名称	描述
31	PLLMF[5]	PLLMF的第五位 参考RCU_CFG0的位27,21:18
30	PLLPRESEL	PLL时钟源预选 0: HXTAL选为PLL时钟源 1: IRC48M选为PLL时钟源
29:4	保留	必须保持复位值
3:0	PREDV[3:0]	HXTAL或CK_IRC48M作为PLL输入源分频因子 由软件置1或清0。这些位仅能在PLL关闭时改写。 注: PREDV的位0 与 RCU_CFG0的位17的功能相同, 修改RCU_CFG0的位17 同时改变这里的位0, 时钟分频因子为 (PREDV + 1) 0000: PLL的输入, 不分频 0001: PLL的输入2分频 0010: PLL的输入3分频 0011: PLL的输入4分频 0100: PLL的输入5分频 0101: PLL的输入6分频 0110: PLL的输入7分频 0111: PLL的输入8分频 1000: PLL的输入9分频 1001: PLL的输入10分频 1010: PLL的输入11分频 1011: PLL的输入12分频 1100: PLL的输入13分频 1101: PLL的输入14分频

1110: PLL的输入15分频

1111: PLL的输入16分频

4.3.13. 配置寄存器 2 (RCU_CFG2)

地址偏移: 0x30

复位值: 0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADCPS C[2]	USBFSPS C[2]														IRC28MDIV
rw	rw														rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		保留				ADCSEL	保留	CECSEL		保留		保留		USART0SEL[1:0]	
						rw		rw							rw

位/位域	名称	描述
31	ADCPSC[2]	ADCPSC第二位 参考RCU_CFG0的位15: 14
30	USBFSPSC[2]	USBFSPSC第二位 参考RCU_CFG0的位23: 22
29:17	保留	必须保持复位值
16	IRC28MDIV	IRC28M是否分频 0: IRC28M/2作为ADC时钟 1: IRC28M作为ADC时钟
15:9	保留	必须保持复位值
8	ADCSEL	ADC时钟源选择 由软件置1或清0。 0: ADC时钟源选择IRC28M时钟 1: ADC时钟源选择APB2时钟2到9分频
7	保留	必须保持复位值
6	CECSEL	CEC时钟源选择 由软件置1或清0。 0: CEC时钟选择IRC8M时钟244分频 1: CEC时钟选择LXTAL时钟
5:2	保留	必须保持复位值
1:0	USART0SEL[1:0]	USART0时钟源选择 由软件置1或清0。

- 00: USART0时钟选择APB2时钟
 - 01: USART0时钟选择系统时钟
 - 10: USART0时钟选择LXTAL时钟
 - 11: USART0时钟选择IRC8M时钟

4.3.14. 控制寄存器 1 (RCU_CTL1)

地址偏移: 0x34

复位值: 0x0000 XX80 X表示未定义

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRC28MCALIB[7:0]								IRC28MADJ[4:0]							
r				rw				r				rw			

位/位域	名称	描述
31:16	保留	必须保持复位值
15:8	IRC28MCALIB[7:0]	IRC28M时钟校准 启动时这些位会被自动初始化为出厂校准参数。
7:3	IRC28MADJ[4:0]	IRC28M时钟调整 这些位由软件设定，最终调整值为(IRC28MADJ)位的值加上IRC28MCALIB[7:0]位的值。最终调整值应该把IRC28M调整到28MHz ±1%。
2	保留	必须保持复位值.
1	IRC28MSTB	IRC28M时钟稳定标志位 由硬件置1来指示内部IRC28M振荡器已经稳定可用。 0: IRC28M振荡器未稳定 1: IRC28M振荡器已稳定
0	IRC28MEN	IRC28M时钟使能 软件置1和清0。 0: 关闭IRC28M时钟 1: 开启IRC28M时钟

4.3.15. 附加时钟控制寄存器(RCU_ADDCTL)

地址偏移： 0xC0

复位值: 0x8000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRC48MCALIB[7:0]								保留							
														r	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								保留							
														CK48M	
														SEL	

rw

位/位域	名称	描述
31:24	IRC48MCALIB [7:0]	内部48MHz RC 振荡器校准值寄存器 上电时自动加载这些位
23:18	保留	必须保持复位值
17	IRC48MSTB	内部48MHz RC振荡器时钟稳定标志位 硬件置‘1’来指示IRC48M振荡器时钟是否稳定待用 0: IRC48M未稳定 1: IRC48M 已稳定
16	IRC48MEN	内部48MHz RC振荡器使能 由软件置位和复位。当进入深度睡眠或待机模式后由硬件复位 0: 关闭IRC48M时钟 1: 打开IRC48M时钟
15:2	保留	必须保持复位值
0	CK48MSEL	48MHz时钟源选择 由软件置位和复位。该位用于选择IRC48M时钟或PLL48M时钟作为CK48M时钟源。 0: 不选择IRC48M时钟(使用USBFSPSC来对CK_PLL分频) 1: 选择IRC48M时钟

4.3.16. 附加时钟中断寄存器(RCU_ADDINT)

地址偏移: 0xCC

复位值: 0x0000 0000

该寄存器可以按字节 (8位)、半字 (16位) 或字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16			
保留								IRC48M	保留									
														w				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
保留	IRC48M	保留								IRC48M	保留							

	STBIE		STBIF	
--	-------	--	-------	--

rw

r

位/位域	名称	描述
31:23	保留	必须保持复位值
22	IRC48MSTBIC	内部48MHz RC振荡器稳定中断清零 软件写1复位IRC48MSTBIF标志位 0: 不复位IRC48MSTBIF标志位 1: 复位IRC48MSTBIF标志位
21:15	保留	必须保持复位值
14	IRC48MSTBIE	内部48 MHz RC振荡器稳定中断使能 由软件置位和复位来使能/禁止IRC48M时钟稳定中断 0: 禁止IRC48M时钟稳定中断 1: 使能IRC48M时钟稳定中断
13:7	保留	必须保持复位值
6	IRC48MSTBIF	IRC48M时钟稳定中断标志位 当内部48 MHz RC振荡器时钟稳定且IRC48MSTBIE位被置1时由硬件置1 软件置位IRC48MSTBIC位时清除该位 0: 无IRC48M时钟稳定中断产生 1: 产生IRC48M时钟稳定中断
5:0	保留	必须保持复位值

4.3.17. APB1 附加使能寄存器 (RCU_ADDAPB1EN)

地址偏移: 0xF8

复位值: 0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

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

位/位域	名称	描述
31:26	保留	必须保持复位值
27	CTCEN	CTC时钟使能

26:0 保留 必须保持复位值

由软件置1或清0。

0: 关闭CTC时钟单元

1: 开启CTC时钟单元

4.3.18. APB1 附加复位寄存器 (RCU_ADD_APB1RST)

地址偏移: 0xFC

复位值: 0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问

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

位/位域	名称	描述
31:28	保留	必须保持复位值
27	CTCRST	<p>CTC复位</p> <p>由软件置1或清0。</p> <p>0: 无作用</p> <p>1: 复位CTC单元</p>
26:0	保留	必须保持复位值

4.3.19. 电源解锁寄存器 (RCU_VKEY)

地址偏移: 0x100

复位值: 0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问

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

位/位域	名称	描述
31:0	KEY[31:0]	RCU_DSV寄存器解锁

这些位只能被软件写，读的话全是0。只有在向RCU_VKEY 寄存器写0x1A2B3C4D

后，RCU_DSV 寄存器才能被写。

4.3.20. 深度睡眠模式电压寄存器 (RCU_DSV)

地址偏移: 0x134

复位值: 0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

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

rw

位/位域	名称	描述
31:2	保留	必须保持复位值
1:0	DSLPVS[1:0]	<p>深度睡眠模式电压选择 这些位由软件置位和清除。</p> <p>00: 在深度睡眠模式下内核电压为1.0V 01: 在深度睡眠模式下内核电压为0.9V 10: 在深度睡眠模式下内核电压为0.8V 11: 在深度睡眠模式下内核电压为0.7V</p>

5. 时钟校准控制器（CTC）

5.1. 简介

时钟校准控制器（CTC）采用硬件的方式，自动校准内部48MHz RC晶振（IRC48M）。当USBFS模块使用IRC48M时钟作为时钟源时，必须要求IRC48M时钟频率在 $48\text{MHz}\pm500\text{ppm}$ 的范围内，但是没有经过校准的内部晶振无法满足这么高的精度。CTC模块基于外部高精度的参考信号源来校准IRC48M的时钟频率，通过自动的或手动的调整校准值，以得到一个精准的IRC48M时钟。

5.2. 主要特性

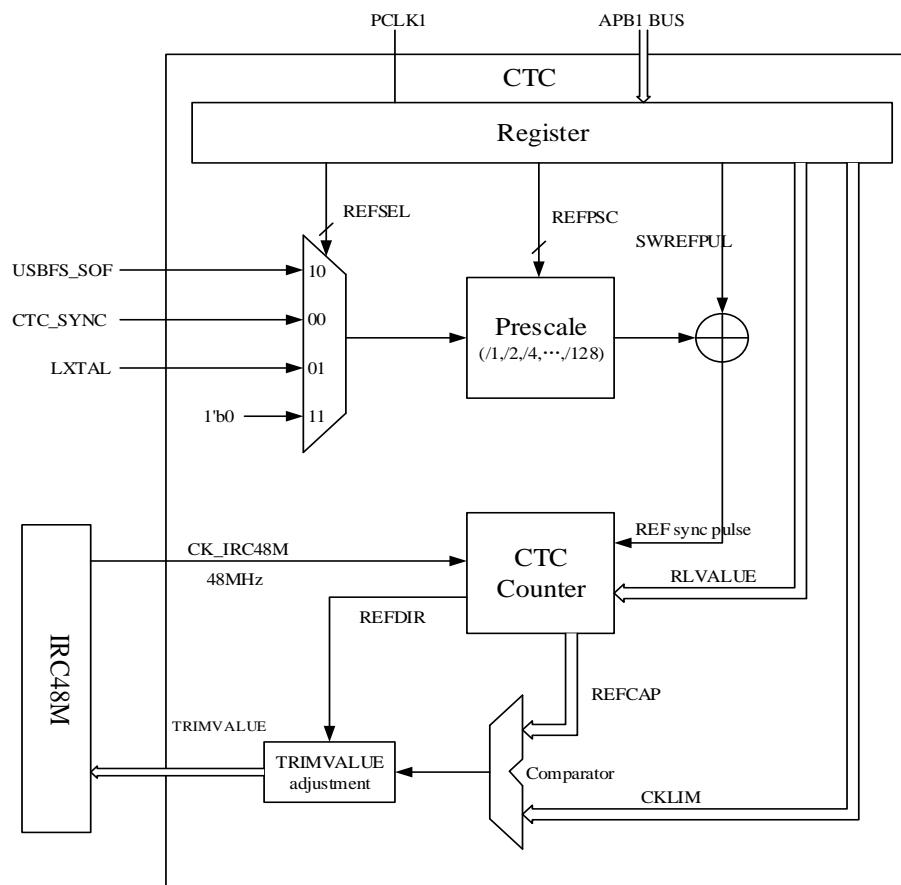
- 三个外部参考信号源：GPIO，LXTAL时钟，USBFS SOF；
- 提供软件参考同步脉冲；
- 硬件自动校准，无需软件操作；
- 具有参考信号源捕获和重载功能的16 bits校准计数器；
- 用于频率评估和自动校准的8 bits时钟校准基值；
- 标志位和中断，用于指示时钟校准的状态：校准成功状态（CKOKIF），警告状态（CKWARNIF）和错误状态（ERRIF）。

注意： 如果使用USBFS SOF作为外部参考信号源，USBFS_GCCFG寄存器的SOFOEN位必须要置位。

5.3. 功能描述

CTC模块的内部结构图如[图5-1. CTC简介](#)。

图 5-1. CTC 简介



5.3.1. REF 同步脉冲发生器

首先，通过设置**CTC_CTL1**寄存器（CTC控制寄存器1）中的**REFSEL**位来选择参考信号源：GPIO，LXTAL时钟输出或者USBFSOF。

然后，可以通过设置**CTC_CTL1**寄存器中的**REFPOL**位来配置参考信号源同步时的信号极性，通过设置**CTC_CTL1**寄存器中的**REFPSC**位来产生一个合适的同步时钟频率信号。

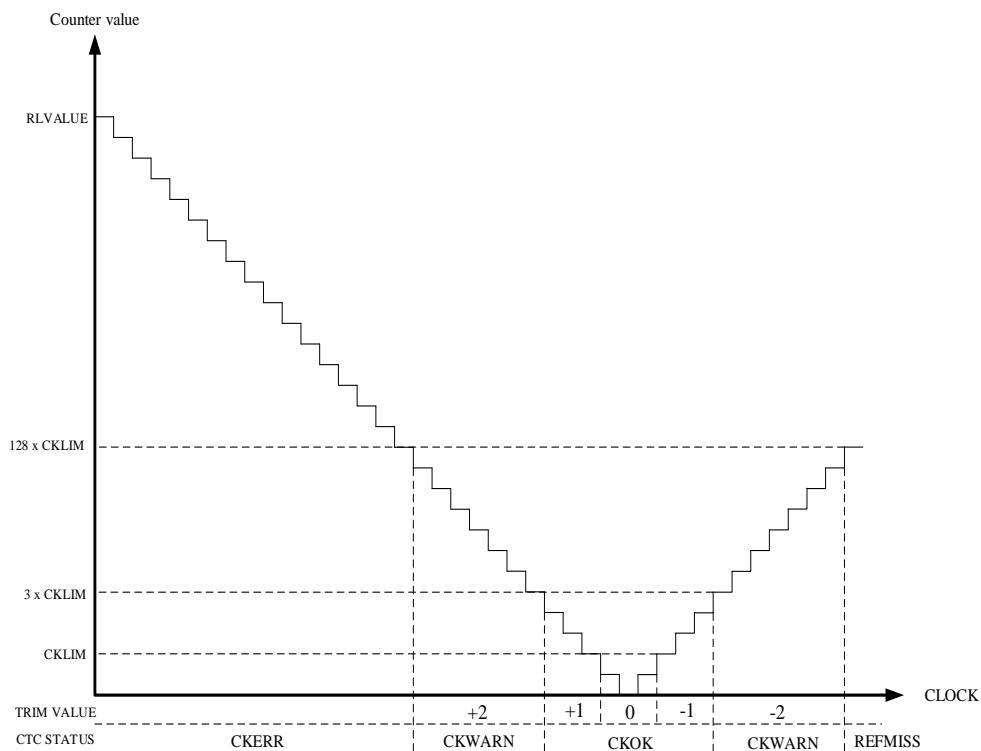
如果需要使用软件参考脉冲信号，则需要设置**CTC_CTL0**寄存器（CTC控制寄存器0）中的**SWREFPUL**位为1。软件参考脉冲信号与外部参考脉冲信号最后进行逻辑或操作。

5.3.2. CTC 校准计数器

CTC 时钟校准计数器由 **CK_IRC48M** 提供时钟。在置位 **CTC_CTL0** 寄存器中的 **CNTEN** 位后，当检测到第一个 REF 同步脉冲信号，计数器开始从 **RLVALUE** 值（**RLVALUE** 在 **CTC_CTL1** 寄存器中定义）开始向下计数。每次检测到 REF 同步脉冲信号时，计数器重载 **RLVALUE** 值，同时重新开始向下计数。如果始终检测不到 REF 同步脉冲信号，计数器会向下计数到零，然后再向上计数到 $128 \times \text{CKLIM}$ （**CKLIM** 在 **CTC_CTL1** 中定义），最后停止，直到检测到下一个 REF 同步脉冲信号。一旦检测到 REF 同步脉冲信号，当前 CTC 校准计数器的计数值被捕获存入 **CTC_STAT**（CTC 状态寄存器）中的 **REFCAP** 位，同时，当前计数器的计数方向被存

入 CTC_STAT 中的 REFDIR 位。详细内容如[图 5-2. CTC 校准计数器](#)所示。

图 5-2. CTC 校准计数器



5.3.3. 频率评估和自动校准过程

当REF同步脉冲信号出现时，时钟频率评估功能开始执行。如果REF同步脉冲信号出现在计数器向下计数的过程中，说明当前时钟频率比期望时钟频率（频率为48M）慢，需要增大CTC_CTL0中的TRIMVALUE值（时钟校准值）。如果REF同步脉冲信号出现在计数器向上计数的过程中，说明当前时钟频率比期望时钟频率快，需要减小TRIMVALUE值。CTC_STAT中的CKOKIF位，CKWARNIF位，CKERR位和REFMISS位反映了频率评估的状态。

如果CTC_CTL0中的AUTOTRIM（硬件自动校准模式）位置1，硬件自动校准模式使能。在这个模式中，如果REF同步脉冲信号出现在计数器向下计数的过程中，说明当前时钟频率比期望时钟频率慢，CTC_CTL0中的TRIMVALUE值会自动增大，来提高当前的时钟频率。反之，如果REF同步脉冲信号出现在计数器向上计数的过程中，说明当前时钟频率比期望时钟频率快，TRIMVALUE值会自动减小，从而减小当前的时钟频率。

- Counter < CKLIM 时，检测到REF同步脉冲信号；
CTC_STAT中的CKOKIF位（时钟校准成功标志位）被置位，同时，如果CTC_CTL0中的CKOKIE位（时钟校准完成中断使能位）置1，将会产生一个中断。
如果CTC_CTL0中的AUTOTRIM置1，CTC_CTL0中的TRIMVALUE值不变。
- CKLIM ≤ Counter < 3 × CKLIM 时，检测到REF同步脉冲信号；
CTC_STAT中的CKOKIF位被置位，同时，如果CTC_CTL0中的CKOKIE位置1，将会产生一个中断。

如果CTC_CTL0中的AUTOTRIM位置1，在计数器向下计数过程中，CTC_CTL0中的TRIMVALUE值将加1，而在向上计数过程中将减1。

- $3 \times CKLIM \leq Counter < 128 \times CKLIM$ 时，检测到REF同步脉冲信号；
CTC_STAT中的CKWARNIF位（时钟校准警告中断位）被置位，同时，如果CTC_CTL0中的CKWARNIE位（时钟校准警告中断使能位）置1，将会产生一个中断。
如果CTC_CTL0中的AUTOTRIM位置1，在计数器向下计数过程中，CTC_CTL0中的TRIMVALUE值将加2，而在向上计数过程中将减2。
- $Counter \geq 128 \times CKLIM$ ，计数器在向下计数过程中，检测到REF同步脉冲信号；
CTC_STAT中的CKERR位（时钟校准错误位）被置位，同时，如果CTC_CTL0中的ERRIE位（错误中断使能位）置1，将会产生一个中断。
CTC_CTL0中的TRIMVALUE值不变。
- $Counter = 128 \times CKLIM$ ，计数器在向上计数过程中；
CTC_STAT中的REFMISS位（REF同步脉冲丢失位）被置位，同时，如果CTC_CTL0中的ERRIE位置1，将会产生一个中断。
CTC_CTL0中的TRIMVALUE值不变。

如果CTC_CTL0中的TRIMVALUE的校准值大于63，将会发生上溢事件，同时，若TRIMVALUE的校准值小于0，将会发生下溢事件。TRIMVALUE的取值范围为0~63（上溢事件发生时，TRIMVALUE值为63；下溢事件发生时，TRIMVALUE值为0）。然后，CTC_STAT中的TRIMERR位（校准值错误位）将会被置位，如果CTC_CTL0中的ERRIE位置1，将会产生一个中断。

5.3.4. 软件编程指南

CTC_CTL1中RLVALUE位和CKLIM位是时钟频率评估和硬件自动校准的关键。它们的数值由期望时钟的频率（IRC48M: 48 MHz）和REF同步脉冲信号的频率计算得到。理想状态是REF同步脉冲信号在CTC计数器计数到零时出现，所以RLVALUE的值为：

$$RLVALUE = (F_{clock} \div F_{REF}) - 1 \quad (式5-1)$$

CKLIM的值由用户根据时钟的精度来设置，一般建议设为步长的一半，所以CKLIM的值为：

$$CKLIM = (F_{clock} \div F_{REF}) \times 0.12\% \div 2 \quad (式5-2)$$

典型的步长值是0.12%， F_{clock} 是期望时钟的频率（IRC48M）， F_{REF} 是REF同步脉冲信号的频率。

5.4. CTC 寄存器

5.4.1. 控制寄存器 0 (CTC_CTL0)

地址偏移：0x00

复位值：0x0000 2000

该寄存器只能按字（32位）访问。



GD32F3x0 用户手册

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TRIMVALUE[5:0]					SWREF PUL	AUTO TRIM	CNTEN	保留	EREFIE	ERRIE	CKWARN IE	CKOKIE		

位/位域	名称	描述
31:14	保留	必须保持复位值
13:8	TRIMVALUE[5:0]	<p>IRC48M校准值</p> <p>当 CTC_CTL0 中的 AUTOTRIM 值为 0 时，该位由软件置位和清除，该模式用于软件校准过程。</p> <p>当 CTC_CTL0 中的 AUTOTRIM 值为 1 时，该位只读，由硬件自动修改，该模式用于硬件校准过程。</p> <p>TRIMVALUE 的中间值是 32，当 TRIMVALUE 值加 1 时，IRC48M 时钟频率增加大约 57KHz。当 TRIMVALUE 值减 1 时，IRC48M 时钟频率的减少大约 57KHz。</p>
7	SWREFPUL	<p>软件生成同步参考信号脉冲</p> <p>该位由软件置位，并为CTC计数器提供一个同步参考脉冲信号。该位由硬件自动清除，读操作时返回0。</p> <p>0：没有影响</p> <p>1：软件产生一个同步参考脉冲信号</p>
6	AUTOTRIM	<p>硬件自动校准模式</p> <p>该位由软件置位或清除。当该位置1时，硬件自动校准模式使能，通过硬件不断的自动修改CTC_CTL0中的TRIMVALUE值，直到IRC48M的时钟频率达到48MHz。</p> <p>0：禁止硬件自动校准模式</p> <p>1：使能硬件自动校准模式</p>
5	CNTEN	<p>CTC计数器使能</p> <p>该位由软件置位或清除，用于使能或禁止CTC计数器。当该位置1时，不能修改CTC_CTL1的值。</p> <p>0：禁止CTC计数器</p> <p>1：使能 CTC 计数器</p>
4	保留	必须保持复位值
3	EREFIE	<p>期望参考信号中断使能</p> <p>0：禁止期望参考信号产生中断</p> <p>1：使能期望参考信号产生中断</p>
2	ERRIE	<p>错误中断使能</p> <p>0：禁止错误中断</p> <p>1：使能错误中断</p>

1	CKWARNIE	时钟校准警告中断使能 0: 禁止时钟校准警告中断 1: 使能时钟校准警告中断
0	CKOKIE	时钟校准完成中断使能 0: 禁止时钟校准完成中断 1: 使能时钟校准完成中断

5.4.2. 控制寄存器 1 (CTC_CTL1)

地址偏移: 0x04

复位值: 0x2022 BB7F

该寄存器只能按字(32位)访问。当CNTEN为1时,不能修改该寄存器的值。

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

位/位域	名称	描述
31	REFPOL	参考信号源极性 该位由软件置位或清除, 用于选择参考信号源的同步极性 0: 选择上升沿 1: 选择下降沿
30	保留	必须保持复位值
29:28	REFSEL[1:0]	参考信号源选择 该位由软件置位或清除, 用于选择参考信号源 00: 选择GPIO输入信号 01: 选择LXTAL时钟 10: 选择USBFS SOF信号 11: 保留, 选择0
27	保留	必须保持复位值
26:24	REFPSC[2:0]	参考信号源预分频 该位由软件置位或清除 000: 参考信号不分频 001: 参考信号2分频 010: 参考信号4分频 011: 参考信号8分频

- 100: 参考信号16分频
- 101: 参考信号32分频
- 110: 参考信号64分频
- 111: 参考信号128分频

23:16	CKLIM[7:0]	时钟校准时基限值 该位由软件置位或清除，用于定义时钟校准时基限值。该位用于频率评估和自动校准过程，详细情况请参考“ 频率评估和自动校准过程 ”。
15:0	RLVALUE[15:0]	CTC计数器重载值 该位由软件置位或清除，用于定义CTC计数器的重载值，当检测到一个同步参考脉冲时，该值将重载到CTC校准计数器中。

5.4.3. 状态寄存器 (CTC_STAT)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REFCAP[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REFDIR	保留				TRIM ERR	REF MISS	CKERR	保留				EREFIF	ERRIF	CKWARN IF	CKOK IF
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:16	REFCAP[15:0]	CTC计数器捕获值 当检测到一个同步参考脉冲信号时，CTC校准计数器中的计数值被存入到REFCAP位中。
15	REFDIR	CTC校准时钟计数方向 当检测到一个同步参考脉冲信号时，CTC校准计数器的计数方向被存入REFDIR位中。 0: 向上计数 1: 向下计数
14:11	保留	必须保持复位值
10	TRIMERR	校准值错误位 当CTC_CTL0中的TRIMVALUE值发生上溢或下溢时，该位由硬件置位。若CTC_CTL0中的ERRIE位置1，则会产生一个中断。通过写1到CTC_INTC中的ERRIC位，可以将TRIMERR位清零。 0: 无校准值错误发生

	1:	发生校准值错误
9	REFMISS	<p>同步参考脉冲信号丢失</p> <p>当同步参考脉冲信号丢失时，该位由硬件置位。当CTC校准计数器在增计数的过程中计数到$128 \times CKLIM$都没有检测到同步参考脉冲信号时，REFMISS位置位。说明当前时钟太快，无法校准到期望频率值，或者有其他错误产生。通过写1到CTC_INTC中的ERRIC位，可以将REFMISS位清零。</p> <p>0: 无同步参考脉冲信号丢失 1: 同步参考脉冲信号丢失</p>
8	CKERR	<p>时钟校准错误位</p> <p>当时钟校准错误产生时，该位由硬件置位。当CTC校准计数器计数值在减计数的过程中大于或等于$128 \times CKLIM$，并检测到同步参考脉冲信号时，CKERR置位，说明当前时钟太慢，无法校准到期望频率值。当CTC_CTL0中的ERRIE置1时，产生一个中断。通过写1到CTC_INTC中的ERRIC位，可以将CKERR位清零。</p> <p>0: 无时钟校准错误发生 1: 发生时钟校准错误</p>
7:4	保留	必须保持复位值
3	EREFIF	<p>期望参考中断标志位</p> <p>当CTC校准时钟计数器计数到0时，该位由硬件置位。当CTC_CTL0中的EREFIE置1时，产生一个中断。通过写1到CTC_INTC中的EREFIC位，可以将EREFIF位清零。</p> <p>0: 无期望参考信号产生 1: 期望参考信号产生</p>
2	ERRIF	<p>错误中断标志位</p> <p>当发生一个错误时，该位由硬件置位。只要有TRIMERR，REFMISS或者CKERR错误发生时，该位置位。当CTC_CTL0中的ERRIE置位时，产生一个中断。通过写1到CTC_INTC中的ERRIC位，可以将ERRIF位清零。</p> <p>0: 无错误发生 1: 发生错误</p>
1	CKWARNIF	<p>时钟校准警告中断标志位</p> <p>当时钟校准警告产生时，该位由硬件置位。当CTC校准计数器计数值大于或等于$3 \times CKLIM$且小于$128 \times CKLIM$，并检测到同步参考脉冲信号时，CKWARNIF置位。这说明当前时钟频率太慢或者太快，但可以通过校准达到期望频率值。当时钟校准警告产生时，TRIMVALUE值加2或者减2。当CTC_CTL0中的CKWARNIE置1时，产生一个中断。通过写1到CTC_INTC中的CKWARNIC位，可以将CKWARNIF位清零。</p> <p>0: 无时钟校准警告发生 1: 有时钟校准警告发生</p>
0	CKOKIF	<p>时钟校准成功中断标志位</p> <p>当时钟校准成功时，该位由硬件置位。若在CTC校准计数器计数值小于$3 \times CKLIM$时，检测当同步参考脉冲信号，CKOKIF置位。说明当前时钟频率正常，可以使用，不需要通过TRIMVALUE值进行时钟校准。当CTC_CTL0中的CKOKIE置1时，产生一个中断。通过写1到CTC_INTC中的CKOKIC位，可以将CKOKIF位清零。</p>

0: 时钟校准未成功

1: 时钟校准成功

5.4.4. 中断清除寄存器 (CTC_INTC)

地址偏移: 0x0C

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												EREFIC	ERRIC	CKWARN IC	CKOK IC
												w	w	w	w

位/位域	名称	描述
31:4	保留	必须保持复位值
3	EREFIC	EREFIF 中断清除位 该位只能由软件写, 读操作返回 0。写 1 可以清除 CTC_STAT 中的 EREFIF 位, 写 0 没影响。
2	ERRIC	ERRIF 中断清除位 该位只能由软件写, 读操作返回 0。写 1 可以清除 CTC_STAT 中的 ERRIF 位, TRIMERR 位, REFMIS 位和 CKERR 位, 写 0 没影响。
1	CKWARNIC	CKWARNIF 中断清除位 该位只能由软件写, 读操作返回 0。写 1 可以清除 CTC_STAT 中的 CKWARNIF 位, 写 0 没影响。
0	CKOKIC	CKOKIF 中断清除位 该位只能由软件写, 读操作返回 0。写 1 可以清除 CTC_STAT 中的 CKOKIF 位, 写 0 没影响。

6. 中断/事件控制器 (EXTI)

6.1. 简介

Cortex-M4集成了嵌套式矢量型中断控制器(Nested Vectored Interrupt Controller (NVIC))来实现高效的异常和中断处理。NVIC实现了低延迟的异常和中断处理，以及电源管理控制。它和内核是紧密耦合的。更多关于NVIC的说明请参考《Cortex-M4技术参考手册》。

EXTI（中断/事件控制器）包括24个相互独立的边沿检测电路并且能够向处理器内核产生中断请求或唤醒事件。EXTI有三种触发类型：上升沿触发、下降沿触发和任意沿触发。EXTI中的每一个边沿检测电路都可以独立配置和屏蔽。

6.2. 主要特性

- Cortex-M4系统异常；
- 68种可屏蔽的外设中断（GD32F3x0产品）；
- 4位中断优先级配置位——共提供16个中断优先等级；
- 高效的中断处理；
- 支持异常抢占和咬尾中断；
- 将系统从省电模式唤醒；
- EXTI中有24个相互独立的边沿检测电路；
- 3种触发类型：上升沿触发，下降沿触发和任意沿触发；
- 软件中断或事件触发；
- 可配置的触发源。

6.3. 中断功能描述

ARM Cortex-M4处理器和嵌套式矢量型中断控制器(NVIC)在处理(Handler)模式下对所有异常进行优先级区分以及处理。当异常发生时，系统自动将当前处理器工作状态压栈，在执行完中断服务子程序(ISR)后自动将其出栈。

取向量是和当前工作态压栈并行进行的，从而提高了中断入口效率。处理器支持咬尾中断，可实现背靠背中断，大大削减了反复切换工作态所带来的开销。下表列出了所有的异常类型。

表 6-1. Cortex-M4 中的 NVIC 异常类型

异常类型	向量编号	优先级(a)	向量地址	描述
-	0	-	0x0000_0000	保留
复位	1	-3	0x0000_0004	复位
NMI	2	-2	0x0000_0008	不可屏蔽中断
硬件故障	3	-1	0x0000_000C	各种硬件级别的故障

异常类型	向量编号	优先级(a)	向量地址	描述
存储器管理	4	可编程设置	0x0000_0010	存储器管理
总线故障	5	可编程设置	0x0000_0014	预取指故障, 存储器访问故障
用法故障	6	可编程设置	0x0000_0018	未定义的指令或非法状态
-	7-10	-	0x0000_001C - 0x0000_002B	保留
SVCALL 服务调用	11	可编程设置	0x0000_002C	通过 SWI 指令实现系统服务调用
调试监控	12	可编程设置	0x0000_0030	调试监视器
-	13	-	0x0000_0034	保留
PendSV 挂起服务	14	可编程设置	0x0000_0038	可挂起的系统服务请求
系统节拍	15	可编程设置	0x0000_003C	系统节拍定时器

SysTick校准值设为13500, SysTick时钟频率配置为HCLK/8, 此时若HCLK时钟被配置为108MHz, 则SysTick中断会1ms响应一次。

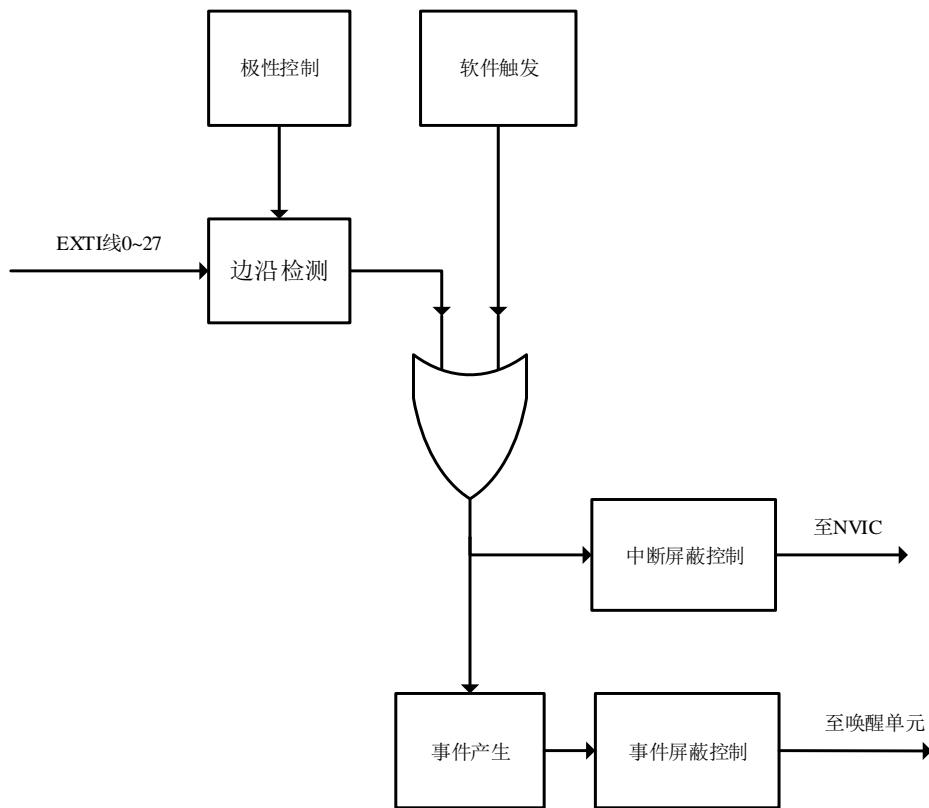
表 6-2. 中断向量表

中断编号	向量编号	外设中断描述	向量地址
IRQ 0	16	窗口看门狗中断	0x0000_0040
IRQ 1	17	连接到 EXTI 线的 LVD 中断	0x0000_0044
IRQ 2	18	RTC 全局中断	0x0000_0048
IRQ 3	19	FMC 全局中断	0x0000_004C
IRQ 4	20	RCU 全局中断	0x0000_0050
IRQ 5	21	EXTI 线 0-1 中断	0x0000_0054
IRQ 6	22	EXTI 线 2-3 中断	0x0000_0058
IRQ 7	23	EXTI 线 4-15 中断	0x0000_005C
IRQ 8	24	TSI 全局中断	0x0000_0060
IRQ 9	25	DMA 通道 0 全局中断	0x0000_0064
IRQ 10	26	DMA 通道 1-2 全局中断	0x0000_0068
IRQ 11	27	DMA 通道 3-4 全局中断	0x0000_006C
IRQ 12	28	ADC 和 CMP0-1 中断	0x0000_0070
IRQ 13	29	TIMER0 中止, 更新, 触发与通道换相中断	0x0000_0074
IRQ 14	30	TIMER0 捕获比较中断	0x0000_0078
IRQ 15	31	TIMER1 全局中断	0x0000_007C
IRQ 16	32	TIMER2 全局中断	0x0000_0080
IRQ 17	33	TIMER5 和 DAC 全局中断	0x0000_0084
IRQ 18	34	保留	0x0000_0088
IRQ 19	35	TIMER13 全局中断	0x0000_008C
IRQ 20	36	TIMER14 全局中断	0x0000_0090
IRQ 21	37	TIMER15 全局中断	0x0000_0094
IRQ 22	38	TIMER16 全局中断	0x0000_0098
IRQ 23	39	I2C0 事件中断	0x0000_009C

中断编号	向量编号	外设中断描述	向量地址
IRQ 24	40	I2C1 事件中断	0x0000_00A0
IRQ 25	41	SPI0 全局中断	0x0000_00A4
IRQ 26	42	SPI1 全局中断	0x0000_00A8
IRQ 27	43	USART0 全局中断	0x0000_00AC
IRQ 28	44	USART1 全局中断	0x0000_00B0
IRQ 29	45	保留	0x0000_00B4
IRQ 30	46	CEC 全局中断	0x0000_00B8
IRQ 31	47	保留	0x0000_00BC
IRQ 32	48	I2C0 错误中断	0x0000_00C0
IRQ 33	49	保留	0x0000_00C4
IRQ 34	50	I2C1 错误中断	0x0000_00C8
IRQ 35	51	保留	0x0000_00CC
IRQ 36	52	保留	0x0000_00D0
IRQ 37	53	保留	0x0000_00D4
IRQ 38	54	保留	0x0000_00D8
IRQ 39-41	55-57	保留	0x0000_00DC- 0x0000_00E4
IRQ 42	58	通过 EXTI 线 18 产生的 USBFS 唤醒中断	0x0000_00E8
IRQ 43-47	59-63	保留	0x0000_00EC- 0x0000_00FC
IRQ 48	64	DMA 通道 5-6 全局中断	0x0000_0100
IRQ 49-50	65-66	保留	0x0000_0104- 0x0000_0108
IRQ 51	67	保留	0x0000_010C
IRQ52-66	68-82	保留	0x0000_0110- 0x0000_0148
IRQ67	83	USBFS 全局中断	0x0000_014C

6.4. 外部中断及事件(EXTI)结构框图

图 6-1. EXTI 结构框图



6.5. 外部中断及事件功能概述

EXTI 包含 24 个相互独立的边沿检测电路并且可以向处理器产生中断请求或事件唤醒。EXTI 提供 3 种触发类型：上升沿触发，下降沿触发和任意沿触发。EXTI 中每个边沿检测电路都可以分别予以配置或屏蔽。

EXTI 触发源包括来自 I/O 管脚的 16 根线以及来自内部模块的 8 根线，包括 LVD、RTC、USB、USART、CEC 和 CMP。通过配置 SYSCFG 模块的 SYSCFG_EXTISSx 寄存器，所有的 GPIO 管脚都可以被选作 EXTI 的触发源，具体细节请参考 [系统配置寄存器 \(SYSCFG\)](#)。

除了中断，EXTI 还可以向处理器提供事件信号。Cortex-M4 内核完全支持等待中断(WFI)，等待事件(WFE)和发送事件(SEV)指令。唤醒中断控制器(WIC)可以让处理器和 NVIC 进入功耗极低的省电模式，由 WIC 来识别中断和事件以及判断优先级。当某些预期的事件发生时，EXTI 能唤醒处理器及整个系统，例如一个特定的 I/O 管脚电平翻转或者 RTC 闹钟。

表 6-3. EXTI 触发源

EXTI 线编号	触发源
0	PA0 / PB0 / PC0 / PF0

EXTI 线编号	触发源
1	PA1 / PB1 / PC1 / PF1
2	PA2 / PB2 / PC2 / PD2
3	PA3 / PB3 / PC3
4	PA4 / PB4 / PC4 / PF4
5	PA5 / PB5 / PC5 / PF5
6	PA6 / PB6 / PC6 / PF6
7	PA7 / PB7 / PC7 / PF7
8	PA8 / PB8 / PC8
9	PA9 / PB9 / PC9
10	PA10 / PB10 / PC10
11	PA11 / PB11 / PC11
12	PA12 / PB12 / PC12
13	PA13 / PB13 / PC13
14	PA14 / PB14 / PC14
15	PA15 / PB15 / PC15
16	LVD
17	RTC 闹钟
18	USBFS 唤醒
19	RTC 干预和时间戳
20	保留
21	CMP0 输出
22	CMP1 输出
23	保留
24	保留
25	USART0 唤醒
26	保留
27	CEC 唤醒

6.6. 寄存器定义

6.6.1. 中断使能寄存器 (EXTI_INTEN)

地址偏移: 0x00

复位值: 0x0F94 0000

该寄存器只能按字 (32 位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留			INTEN27	INTEN26	INTEN25	INTEN24	INTEN23	INTEN22	INTEN21	INTEN20	INTEN19	INTEN18	INTEN17	INTEN16	
rw	rw														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTEN15	INTEN14	INTEN13	INTEN12	INTEN11	INTEN10	INTEN9	INTEN8	INTEN7	INTEN6	INTEN5	INTEN4	INTEN3	INTEN2	INTEN1	INTENO

rw rw

位/位域	名称	描述
31:28	保留	必须保持复位值
27:0	INTENx	中断使能位x(x=0..27) 0: 第x线中断被禁止 1: 第x线中断被使能

6.6.2. 事件使能寄存器 (EXTI_EVEN)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				EVEN27	EVEN26	EVEN25	EVEN24	EVEN23	EVEN22	EVEN21	EVEN20	EVEN19	EVEN18	EVEN17	EVEN16
				rw											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVEN15	EVEN14	EVEN13	EVEN12	EVEN11	EVEN10	EVEN9	EVEN8	EVEN7	EVEN6	EVEN5	EVEN4	EVEN3	EVEN2	EVEN1	EVEN0
rw															

位/位域	名称	描述
31:28	保留	必须保持复位值
27:0	EVENx	事件使能位x(x=0..27) 0: 第x线事件被禁止 1: 第x线事件被使能

6.6.3. 上升沿触发使能寄存器 (EXTI_RTEN)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								RTEN22	RTEN21	保留	RTEN19	RTEN18	RTEN17	RTEN16	
								rw	rw		rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTEN15	RTEN14	RTEN13	RTEN12	RTEN11	RTEN10	RTEN9	RTEN8	RTEN7	RTEN6	RTEN5	RTEN4	RTEN3	RTEN2	RTEN1	RTEN0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:23	保留	必须保持复位值
22:21	RTENx	上升沿触发使能(x=21,22) 0: 第x线上升沿触发无效 1: 第x线上升沿触发有效 (中断/事件请求)
20	保留	必须保持复位值
19:0	RTENx	上升沿触发使能(x=0..19) 0: 第x线上升沿触发无效 1: 第x线上升沿触发有效 (中断/事件请求)

6.6.4. 下降沿触发使能寄存器 (**EXTI_FTEN**)

地址偏移: 0x0C

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留										FTEN22	FTEN21	保留	FTEN19	FTEN18	FTEN17	FTEN16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
FTEN15	FTEN14	FTEN13	FTEN12	FTEN11	FTEN10	FTEN9	FTEN8	FTEN7	FTEN6	FTEN5	FTEN4	FTEN3	FTEN2	FTEN1	FTEN0	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

位/位域	名称	描述
31:23	保留	必须保持复位值
22:21	FTENx	下降沿触发使能(x=21,22) 0: 第x线下降沿触发无效 1: 第x线下降沿触发有效 (中断/事件请求)
20	保留	必须保持复位值
19:0	FTENx	下降沿触发使能(x=0,19) 0: 第x线下降沿触发无效 1: 第x线下降沿触发有效 (中断/事件请求)

6.6.5. 软件中断事件寄存器 (**EXTI_SWIEV**)

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问

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

保留															SWIEV22	SWIEV21	保留	SWIEV19	SWIEV18	SWIEV17	SWIEV16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
SWIEV15	SWIEV14	SWIEV13	SWIEV12	SWIEV11	SWIEV10	SWIEV9	SWIEV8	SWIEV7	SWIEV6	SWIEV5	SWIEV4	SWIEV3	SWIEV2	SWIEV1	SWIEV0						
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW						

位/位域	名称	描述
31:23	保留	必须保持复位值
22:21	SWIEVx	中断/事件软件触发(x=21,22) 0: 禁用EXTI线x软件中断/事件请求 1: 激活EXTI线x软件中断/事件请求
20	保留	必须保持复位值
19:0	SWIEVx	中断/事件软件触发(x=0,19) 0: 禁用EXTI线x软件中断/事件请求 1: 激活EXTI线x软件中断/事件请求

6.6.6. 挂起寄存器 (EXTI_PD)

地址偏移: 0x14

复位值: 未定义

该寄存器只能按字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								PD22	PD21	保留	PD19	PD18	PD17	PD16	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
rc_w1															

位/位域	名称	描述
31:23	保留	必须保持复位值
22:21	PDx	中断挂起状态(x=21,22) 0: EXTI线x没有被触发 1: EXTI线x被触发, 对这些位写1, 可将其清0
20	保留	必须保持复位值
19:0	PDx	中断挂起状态(x=0,19) 0: EXTI线x没有被触发 1: EXTI线x被触发, 对这些位写1, 可将其清0

7. 通用和备用输入/输出接口（GPIO 和 AFIO）

7.1. 简介

最多可支持 55 个通用 I/O 引脚 (GPIO)，分别为 PA0 ~ PA15，PB0 ~ PB15，PC0 ~ PC15，PD2，PF0/PF1，PF4 ~ PF7，各片上设备用其来实现逻辑输入/输出功能。每个 GPIO 端口有相关的控制和配置寄存器以满足特定应用的需求。

GPIO 端口和其他的备用功能 (AFs) 备用引脚，在特定的封装下获得最大的灵活性。GPIO 引脚通过配置相关的寄存器可以用作备用功能引脚，备用功能输入/输出都可以。

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

7.2. 主要特性

- 输入/输出方向控制；
- 施密特触发输入功能使能控制；
- 每个引脚都具有弱上拉/下拉功能；
- 推挽/开漏输出使能控制；
- 置位/复位输出使能；
- 输出驱动速度选择；
- 模拟输入/输出配置；
- 备用功能输入/输出配置；
- 端口锁定配置；
- 单周期输出翻转功能。

7.3. 功能描述

每个通用I/O端口都可以通过32位控制寄存器 (GPIOx_CTL) 配置为GPIO输入，GPIO输出，AF功能或模拟模式。引脚AFIO输入/输出是通过AFIO功能使能来选择。当端口配置为输出 (GPIO输出或AFIO输出) 时，可以通过GPIO输出模式寄存器 (GPIOx_OMODE) 配置为推挽或开漏模式。输出端口的最大速度可以通过GPIO输出速度寄存器 (GPIOx_OSPEED(y=0,1)) 配置。每个端口可以通过GPIO上/下拉寄存器 (GPIOx_PUD) 配置为浮空 (无上拉或下拉)，上拉或下拉功能。

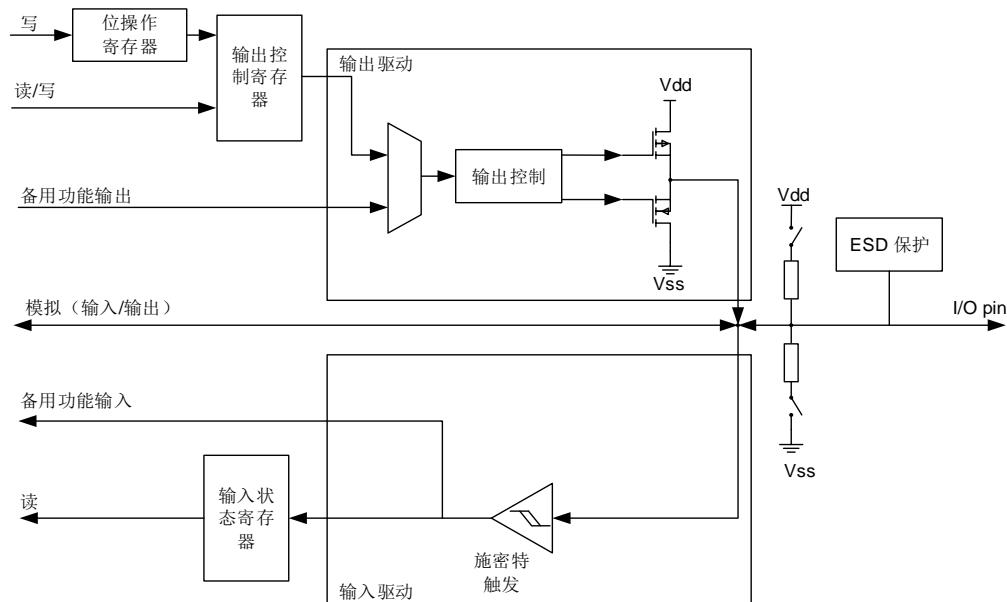
表 7-1. GPIO 配置表

PAD TYPE		CTLn	OMn	PUDn
GPIO 输入	X	悬空	X	00
		上拉		01
		下拉		10

PAD TYPE			CTLn	OMn	PUDn
GPIO 输出	推挽	悬空	01	00	
		上拉		01	
		下拉		10	
	开漏	悬空	1	00	
		上拉		01	
		下拉		10	
AFIO 输入	X	悬空	10	00	
		上拉		01	
		下拉		10	
AFIO 输出	推挽	悬空	10	00	
		上拉		01	
		下拉		10	
	开漏	悬空	1	00	
		上拉		01	
		下拉		10	
ANALOG	X	X	11	X	XX

图7-1. 标准I/O端口位的基本结构为标准I/O端口位的基本结构图。

图 7-1. 标准 I/O 端口位的基本结构



7.3.1. GPIO 引脚配置

在复位期间或复位之后，备用功能并未激活，所有 GPIO 端口都被配置成输入浮空模式，这种输入模式禁用上拉(PU)/下拉(PD)电阻。但是复位后，串行线调试为输入 PU/PD 模式。

PA14: SWCLK为AF下拉模式

PA13: SWDIO为AF上拉模式

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

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

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

7.3.2. 备用功能（AF）

当端口配置为AFIO（设置**GPIOx_CTL**寄存器中的**CTLy**值为“10”）时，该端口用作外设备用功能。通过配置GPIO备用功能选择寄存器（**GPIOx_AFSELy(y=0..1)**），每个端口可以配置16个备用功能。端口备用功能分配的详细介绍见芯片数据手册。

7.3.3. 附加功能

有些引脚具有附加功能，它们优先于标准GPIO寄存器中的配置。当用作ADC或DAC附加功能时，引脚必须配置成模拟模式。当引脚用作RTC、WKUPx和振荡器附加功能时，端口类型通过相关的RTC、PMU和RCU寄存器自动设置。当附加功能禁用时，这些端口可用作普通GPIO。

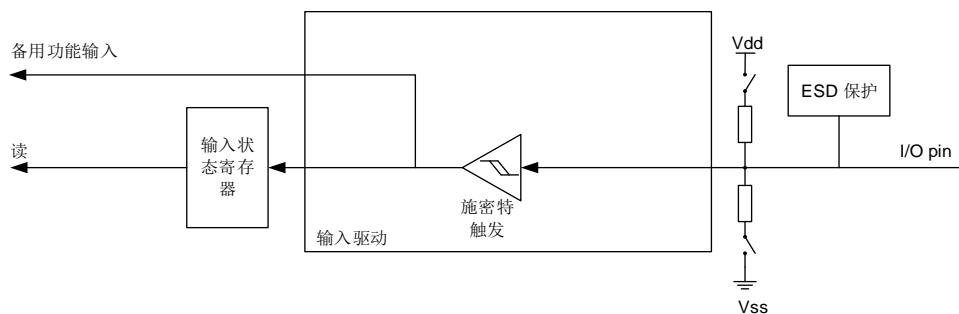
7.3.4. 输入配置

当GPIO引脚配置为输入时：

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

图7-2. 浮空/上拉/下拉输入配置是I/O引脚的输入配置。

图 7-2. 浮空/上拉/下拉输入配置



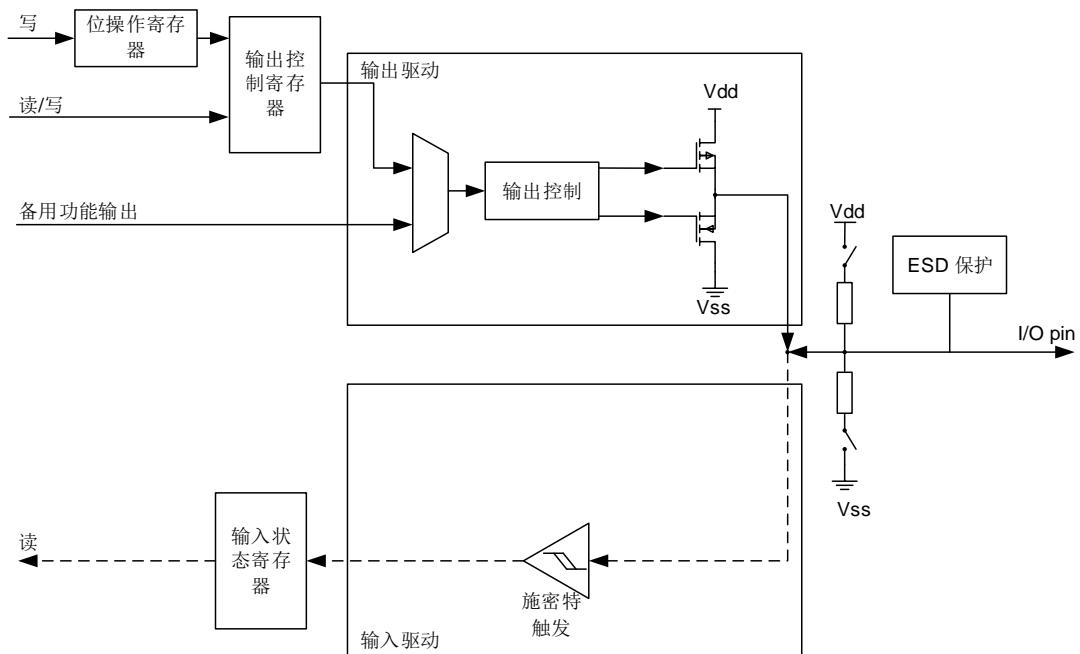
7.3.5. 输出配置

当GPIO配置为输出时：

- 施密特触发输入使能；
- 可选择的弱上拉和下拉电阻；
- 输出缓冲器使能：
 - 开漏模式：输出控制寄存器设置为“0”时，相应引脚输出“0”；输出控制寄存器设置为“1”，相应管脚处于高阻状态；
 - 推挽模式：输出控制寄存器设置为“0”时，相应引脚输出“0”；输出控制寄存器设置为“1”，相应引脚输出“1”。
- 在推挽模式下，对端口输出控制寄存器的读访问将返回上次写入的值；
- 在开漏模式下，对端口输入状态寄存器的读访问将返回I/O的状态。

图 7-3. 输出配置 是 I/O 端口的输出配置。

图 7-3. 输出配置



7.3.6. 模拟配置

当GPIO引脚用于模拟模式时：

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

图7-4. 模拟高阻配置 是 I/O 端口的模拟高阻配置。

图 7-4. 模拟高阻配置



7.3.7. 备用功能（AF）配置

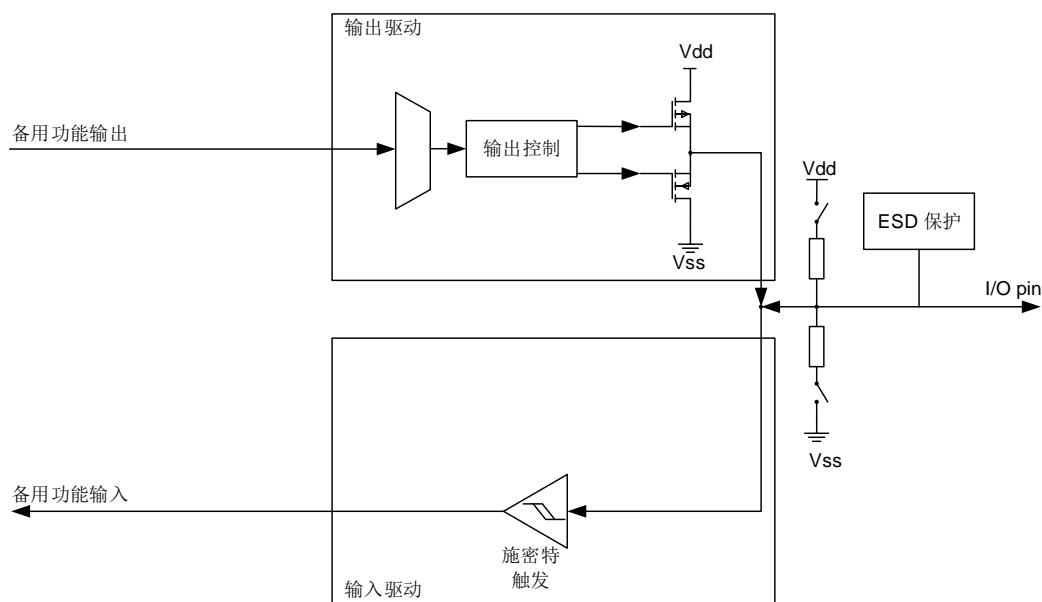
为了适应不同的器件封装，GPIO端口支持软件配置将一些备用功能应用到其他引脚上。

当引脚配置为备用功能时：

- 输出缓冲器启用开漏或者推挽功能；
- 输出缓冲器由外设驱动；
- 施密特触发输入使能；
- 可选择的弱上拉/下拉电阻；
- I/O引脚上的数据在每个AHB时钟周期采样并存入端口输入状态寄存器；
- 在开漏模式下对端口输入状态寄存器进行读操作，将获得I/O口的状态；
- 在推挽输出模式下对端口输出控制寄存器进行读操作，将返回上次写入的值。

[图7-5. 备用功能配置](#)是I/O端口备用功能配置图。

图 7-5. 备用功能配置



7.3.8. GPIO 锁定功能

GPIO的锁定机制可以保护I/O端口的配置。

被保护的寄存器有：GPIOx_CTL, GPIOx_OMODE, GPIOx_OSPEED(y=0,1), GPIOx_PUD和

GPIO_x_AFSEL_y(y=0..1)。通过配置32位锁定寄存器(GPIO_x_LOCK)可以锁定I/O端口的配置。通过32位锁定寄存器(GPIO_x_LOCK)可以锁定I/O配置。当LOCK序列已经被应用在相应端口位上，直到下一次复位前，不能改变锁定寄存器的值。建议在电源驱动模块驱动的配置时使用锁定功能。

7.3.9. GPIO 单周期输出翻转功能

通过将GPIO_x_TG寄存器中对应的位写1，GPIO可以在一个AHB时钟周期内翻转I/O的输出电平。输出信号的频率可以达到AHB时钟的一半。

7.3.10. GPIO 超高速驱动能力

当GPIO工作在输出方向时，它可以在输出模式或AF模式下驱动超过50MHz的超高速信号。配置GPIO_x_OSPD0寄存器相应的OSPD_y位为“11”同时将GPIO_x_OSPD1寄存器响应的SPD_y位置1可以启用该功能。当使用到该功能时，需要使能I/O补偿功能（详细信息请参考[I/O补偿单元](#)）。

7.4. GPIO 寄存器

7.4.1. 端口控制寄存器 (GPIO_x_CTL, x=A..D,F)

地址偏移：0x00

复位值：端口A 0x2800 0000；其他端口 0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTL15[1:0]	CTL14[1:0]	CTL13[1:0]	CTL12[1:0]	CTL11[1:0]	CTL10[1:0]	CTL9[1:0]	CTL8[1:0]								
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTL7[1:0]	CTL6[1:0]	CTL5[1:0]	CTL4[1:0]	CTL3[1:0]	CTL2[1:0]	CTL1[1:0]	CTL0[1:0]								
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	CTL15[1:0]	Pin 15配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
29:28	CTL14[1:0]	Pin 14配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
27:26	CTL13[1:0]	Pin 13配置位 该位由软件置位和清除。

参照CTL0[1:0]的描述

25:24	CTL12[1:0]	Pin 12配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
23:22	CTL11[1:0]	Pin 11配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
21:20	CTL10[1:0]	Pin 10配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
19:18	CTL9[1:0]	Pin 9配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
17:16	CTL8[1:0]	Pin 8配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
15:14	CTL7[1:0]	Pin 7配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
13:12	CTL6[1:0]	Pin 6配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
11:10	CTL5[1:0]	Pin 5配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
9:8	CTL4[1:0]	Pin 4配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
7:6	CTL3[1:0]	Pin 3配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
5:4	CTL2[1:0]	Pin 2配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
3:2	CTL1[1:0]	Pin 1配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述

1:0 CTL0[1:0] Pin 0配置位

该位由软件置位和清除。

00: 输入模式（复位值）

01: GPIO输出模式

10: 备用功能模式

11: 模拟模式

7.4.2. 端口输出模式寄存器 (**GPIOx_Omode**, x=A..D,F)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OM15	OM14	OM13	OM12	OM11	OM10	OM9	OM8	OM7	OM6	OM5	OM4	OM3	OM2	OM1	OM0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

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

9	OM9	Pin 9输出模式位 该位由软件置位和清除。 参考OM0的描述
8	OM8	Pin 8输出模式位 该位由软件置位和清除。 参考OM0的描述
7	OM7	Pin 7输出模式位 该位由软件置位和清除。 参考OM0的描述
6	OM6	Pin 6输出模式位 该位由软件置位和清除。 参考OM0的描述
5	OM5	Pin 5输出模式位 该位由软件置位和清除。 参考OM0的描述
4	OM4	Pin 4输出模式位 该位由软件置位和清除。 参考OM0的描述
3	OM3	Pin 3输出模式位 该位由软件置位和清除。 参考OM0的描述
2	OM2	Pin 2输出模式位 该位由软件置位和清除。 参考OM0的描述
1	OM1	Pin 1输出模式位 该位由软件置位和清除。 参考OM0的描述
0	OM0	Pin 0输出模式位 该位由软件置位和清除。 0: 输出推挽模式（复位值） 1: 输出开漏模式

7.4.3. 端口输出速度寄存器 0 (GPIOx_OSPD0, x=A..D,F)

地址偏移: 0x08

复位值: 端口 A 0x0C00 0000; 其他端口 0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问。

OSPD15[1:0]	OSPD14[1:0]	OSPD13[1:0]	OSPD12[1:0]	OSPD11[1:0]	OSPD10[1:0]	OSPD9[1:0]	OSPD8[1:0]
rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8
OSPD7[1:0]	OSPD6[1:0]	OSPD5[1:0]	OSPD4[1:0]	OSPD3[1:0]	OSPD2[1:0]	OSPD1[1:0]	OSPD0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	OSPD15[1:0]	Pin 15输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
29:28	OSPD14[1:0]	Pin 14输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
27:26	OSPD13[1:0]	Pin 13输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
25:24	OSPD12[1:0]	Pin 12输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
23:22	OSPD11[1:0]	Pin 11输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
21:20	OSPD10[1:0]	Pin 10输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
19:18	OSPD9[1:0]	Pin 9输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
17:16	OSPD8[1:0]	Pin 8输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
15:14	OSPD7[1:0]	Pin 7输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
13:12	OSPD6[1:0]	Pin 6输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
11:10	OSPD5[1:0]	Pin 5输出最大速度位

		该位由软件置位和清除。 参考OSPD0[1:0]的描述
9:8	OSPD4[1:0]	Pin 4输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
7:6	OSPD3[1:0]	Pin 3输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
5:4	OSPD2[1:0]	Pin 2输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
3:2	OSPD1[1:0]	Pin 1输出最大速度位 该位由软件置位和清除。 参考OSPD0[1:0]的描述
1:0	OSPD0[1:0]	Pin 0输出最大速度位 该位由软件置位和清除。 x0: 输出最大速度2M (复位值) 01: 输出最大速度10M 11: 输出最大速度50M

7.4.4. 端口上拉/下拉寄存器 (**GPIOx_PUD**, x=A..D,F)

地址偏移: 0x0C

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

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

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

位/位域	名称	描述
31:30	PUD15[1:0]	Pin 15上拉或下拉位 该位由软件置位和清除。 参照PUDO[1:0]的描述
29:28	PUD14[1:0]	Pin 14上拉或下拉位 该位由软件置位和清除。 参照PUDO[1:0]的描述

27:26	PUD13[1:0]	Pin 13上拉或下拉位 该位由软件置位和清除。 参照PUDO[1:0]的描述
25:24	PUD12[1:0]	Pin 12上拉或下拉位 该位由软件置位和清除。 参照PUDO[1:0]的描述
23:22	PUD11[1:0]	Pin 11上拉或下拉位 该位由软件置位和清除。 参照PUDO[1:0]的描述
21:20	PUD10[1:0]	Pin 10上拉或下拉位 该位由软件置位和清除。 参照PUDO[1:0]的描述
19:18	PUD9[1:0]	Pin 9上拉或下拉位 该位由软件置位和清除。 参照PUDO[1:0]的描述
17:16	PUD8[1:0]	Pin 8上拉或下拉位 该位由软件置位和清除。 参照PUDO[1:0]的描述
15:14	PUD7[1:0]	Pin 7上拉或下拉位 该位由软件置位和清除。 参照PUDO[1:0]的描述
13:12	PUD6[1:0]	Pin 6上拉或下拉位 该位由软件置位和清除。 参照PUDO[1:0]的描述
11:10	PUD5[1:0]	Pin 5上拉或下拉位 该位由软件置位和清除。 参照PUDO[1:0]的描述
9:8	PUD4[1:0]	Pin 4上拉或下拉位 该位由软件置位和清除。 参照PUDO[1:0]的描述
7:6	PUD3[1:0]	Pin 3上拉或下拉位 该位由软件置位和清除。 参照PUDO[1:0]的描述
5:4	PUD2[1:0]	Pin 2上拉或下拉位 该位由软件置位和清除。 参照PUDO[1:0]的描述
3:2	PUD1[1:0]	Pin 1上拉或下拉位

该位由软件置位和清除。

参照PUDO[1:0]的描述

1:0 PUDO[1:0] Pin 0上拉或下拉位

该位由软件置位和清除。

00: 悬空模式，无上拉和下拉（复位值）

01: 端口上拉模式

10: 端口下拉模式

11: 保留

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

地址偏移: 0x10

复位值: 0x0000 XXXX

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISTAT15	ISTAT14	ISTAT13	ISTAT12	ISTAT11	ISTAT10	ISTAT9	ISTAT8	ISTAT7	ISTAT6	ISTAT5	ISTAT4	ISTAT3	ISTAT2	ISTAT1	ISTAT0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

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

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

地址偏移: 0x14

复位值: 0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OCTL15	OCTL14	OCTL13	OCTL12	OCTL11	OCTL10	OCTL9	OCTL8	OCTL7	OCTL6	OCTL5	OCTL4	OCTL3	OCTL2	OCTL1	OCTL0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

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

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

地址偏移: 0x18

复位值: 0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOP15	BOP14	BOP13	BOP12	BOP11	BOP10	BOP9	BOP8	BOP7	BOP6	BOP5	BOP4	BOP3	BOP2	BOP1	BOP0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位/位域	名称	描述
31:16	CRy	端口清除位y(y=0..15) 该位由软件置位和清除。 0: 相应的OCTLy位没有改变 1: 清除相应的OCTLy位为0
15:0	BOPy[15:0]	端口置位位y(y=0..15) 该位由软件置位和清除。 0: 相应的OCTLy位没有改变 1: 设置相应的OCTLy位为1

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

地址偏移: 0x1C

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LK15	LK14	LK13	LK12	LK11	LK10	LK9	LK8	LK7	LK6	LK5	LK4	LK3	LK2	LK1	LK0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

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

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

地址偏移: 0x20

复位值: 0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL7[3:0]				SEL6[3:0]				SEL5[3:0]				SEL4[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEL3[3:0]				SEL2[3:0]				SEL1[3:0]				SEL0[3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:28	SEL7[3:0]	<p>Pin 7选择备用功能</p> <p>该位由软件置位和清除。</p> <p>参照SEL0 [3:0]的描述</p>
27:24	SEL6[3:0]	<p>Pin 6选择备用功能</p> <p>该位由软件置位和清除。</p> <p>参照SEL0 [3:0]的描述</p>
23:20	SEL5[3:0]	<p>Pin 5选择备用功能</p> <p>该位由软件置位和清除。</p> <p>参照SEL0 [3:0]的描述</p>
19:16	SEL4[3:0]	<p>Pin 4选择备用功能</p> <p>该位由软件置位和清除。</p>

参照SEL0 [3:0]的描述

15:12	SEL3[3:0]	Pin 3选择备用功能 该位由软件置位和清除。 参照SEL0 [3:0]的描述
11:8	SEL2[3:0]	Pin 2选择备用功能 该位由软件置位和清除。 参照SEL0 [3:0]的描述
7:4	SEL1[3:0]	Pin 1选择备用功能 该位由软件置位和清除。 参照SEL0 [3:0]的描述
3:0	SEL0[3:0]	Pin 0选择备用功能 该位由软件置位和清除。 0000: 选择AF0功能 (复位值) 0001: 选择AF1功能 0010: 选择AF2功能 0011: 选择AF3功能 0100: 选择AF4功能 (Port A,B only) 0101: 选择AF5功能 (Port A,B only) 0110: 选择AF6功能 (Port A,B only) 0111: 选择AF7功能 (Port A,B only)

1000 ~ 1111: 保留

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

地址偏移: 0x24

复位值: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL15[3:0]				SEL14[3:0]				SEL13[3:0]				SEL12[3:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEL11[3:0]				SEL10[3:0]				SEL9[3:0]				SEL8[3:0]			

位/位域	名称	描述
31:28	SEL15[3:0]	Pin 15选择备用功能 该位由软件置位和清除。

参照SEL8[3:0]的描述

27:24	SEL14[3:0]	Pin 14选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
23:20	SEL13[3:0]	Pin 13选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
19:16	SEL12[3:0]	Pin 12选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
15:12	SEL11[3:0]	Pin 1选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
11:8	SEL10[3:0]	Pin 10选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
7:4	SEL9[3:0]	Pin 9选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
3:0	SEL8[3:0]	Pin 8选择备用功能 该位由软件置位和清除。 0000: 选择AF0功能 (复位值) 0001: 选择AF1功能 0010: 选择AF2功能 0011: 选择AF3功能 0100: 选择AF4功能 (Port A,B only) 0101: 选择AF5功能 (Port A,B only) 0110: 选择AF6功能 (Port A,B only) 0111: 选择AF7功能 (Port A,B only) 1000 ~ 1111: 保留

7.4.11. 位清除寄存器 (**GPIOx_BC**, x=A..D,F)

地址偏移: 0x28

复位值: 0x0000 0000

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0

w w w w w w w w w w w w w w w

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

7.4.12. 端口位翻转寄存器 (GPIOx_TG, x=A..D,F)

地址偏移: 0x2C

复位值: 0x0000 0000

该寄存器可以按字节(8位)、半字(16位)或字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
TG15	TG14	TG13	TG12	TG11	TG10	TG9	TG8	TG7	TG6	TG5	TG4	TG3	TG2	TG1	TG0

w w w w w w w w w w w w w w w

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

7.4.13. 端口输出速度寄存器 1 (GPIOx_OSPD1, x=A..D,F)

地址偏移: 0x3C

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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



GD32F3x0 用户手册

SPD15	SPD14	SPD13	SPD12	SPD11	SPD10	SPD9	SPD8	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	SPDy	<p>当相应的OSPDy(y=0..15)为"11"时，开启超高速输出功能 如果输出速度超过50MHz，将该位设为1同时设OSPDy为"11"。 该位由软件置位和清除。</p> <p>0: 无影响 1: 最大速度超过50MHz。应设OSPDy为"11"</p>

8. 循环冗余校验计算单元 (CRC)

8.1. 简介

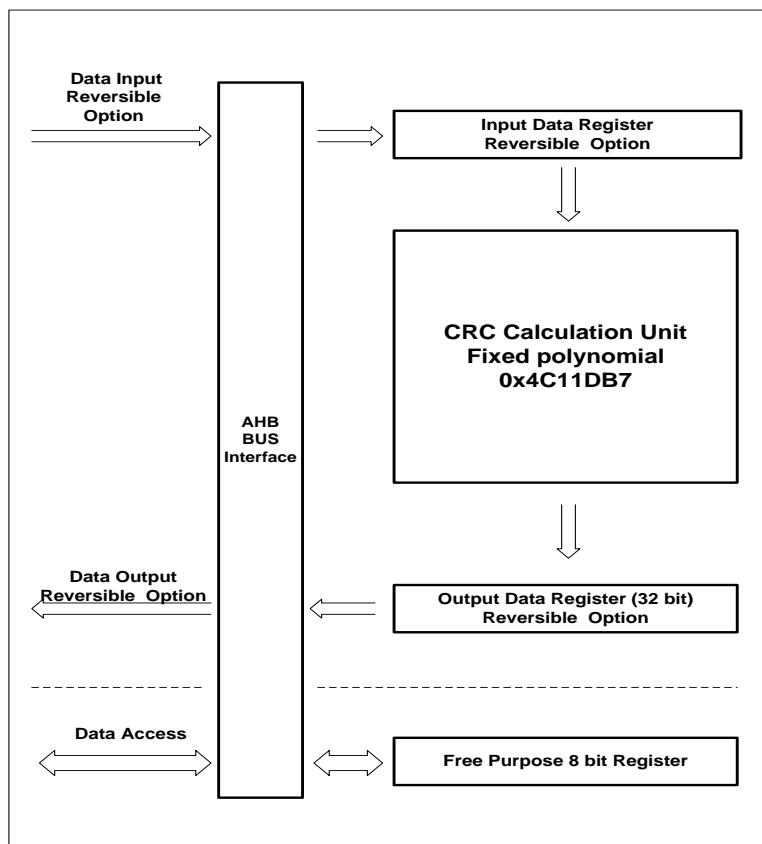
循环冗余校验码是一种用在数字网络和存储设备上的差错校验码，可以校验原始数据的偶然误差。

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

8.2. 主要特性

- 支持7/8/16/32位数据输入；
- 对于7(8)/16/32位的输入数据长度，计算周期分别为1/2/4个AHB时钟周期；
- 用户可以配置多项式长度；
- CRC复位后，用户可以配置计算初值；
- 配有与计算无关的独立8位寄存器，可以供其他任何外设使用；
- 用户可设置的计算初值，可使计算更灵活。

图 8-1. CRC 计算单元框图



8.3. 功能描述

- CRC计算单元可以用来计算32位的原始数据，**CRC_DATA**寄存器接收原始数据并存储计算结果；
如果不通过软件设置**CRC_CTL**寄存器的方式来清除**CRC_DATA**寄存器，新输入的原始数据将会基于前一次**CRC_DATA**寄存器中的结果进行计算；
对于32/16/8(7)位的数据长度，CRC的计算分别要花费4/2/1个AHB的时钟周期。在此期间，因为32位输入缓存的原因，AHB总线将不会被挂起。
- 此模块提供了一个8位的独立寄存器**CRC_FDATA**，**CRC_FDATA**与CRC计算无关，任何时候都可以进行独立的读写操作；
- 逆序功能可以交换输入输出数据的位序。
输入数据可选择三种逆序形式。
以原始数据0x1A2B3C4D为例：
 - 1) 按字节逆序：
32位数据被分成四组，组内完成颠倒。逆序后的数据为：0x58D43CB2
 - 2) 按半字逆序：
32位数据被分成两组，组内完成颠倒。逆序后的数据为：0xD458B23C
 - 3) 按字逆序：
32位数据被分成一组，组内完成颠倒，逆序后的数据为：0xB23CD458
对于输出数据来说，逆序形式为按字逆序。
例如：当REV_O=1，计算结果0x22CC4488将被逆序成0x11223344。
- 用户可配置的初始计算数据。
当RST位置位或对**CRC_IDATA**寄存器进行写操作时，**CRC_DATA**寄存器将自动初始化为**CRC_IDATA**寄存器中的值。
- 用户配置多项式。
通过配置PS[1:0]，用户可以选择有效多项式和输出数据位宽。如果多项式少于32位，那么输入和输出数据的高位无效。当PS[1:0]或多项式改变后，需要复位CRC。

8.4. CRC 寄存器

8.4.1. 数据寄存器 (**CRC_DATA**)

地址偏移: 0x00

复位值: 0xFFFF FFFF

该寄存器只能按字(32位)访问

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

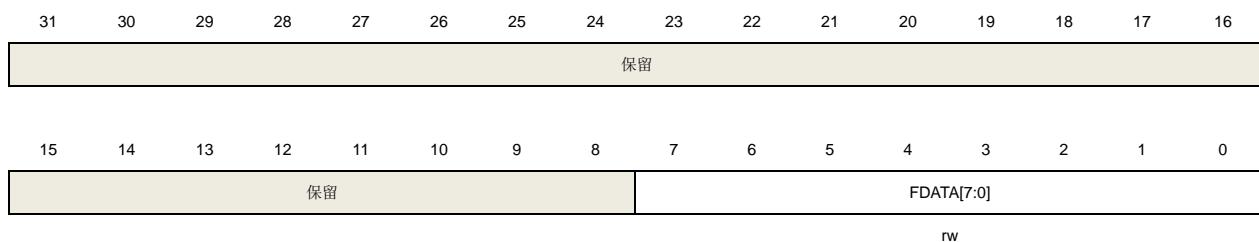
位/位域	名称	描述
31:0	DATA[31:0]	CRC 计算结果位 软件可读可写。 该寄存器用于接收待计算的新数据，直接将其写入即可。刚写入的数据不能被读出来因为读取该寄存器得到的是上次CRC计算的结果。

8.4.2. 独立数据寄存器 (CRC_FDATA)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器只能按字(32位)访问



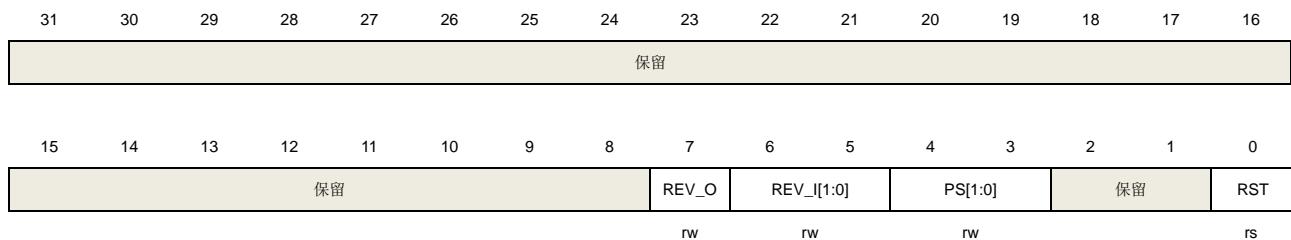
位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	FDATA[7:0]	独立数据寄存器位 软件可读可写。 这些位与CRC计算无关。该字节能被任何其他外设用于其他任何目的。该字节不受CRC_CTL寄存器的影响。

8.4.3. 控制寄存器 (CRC_CTL)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器只能按字(32位)访问



位/位域	名称	描述
------	----	----

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

8.4.4. 初值寄存器 (CRC_IDATA)

地址偏移: 0x10

复位值: 0xFFFF FFFF

该寄存器只能按字(32位)访问

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

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

8.4.5. 多项式寄存器 (CRC_POLY)

地址偏移: 0x14

复位值: 0x04C1 1DB7

该寄存器只能按字(32位)访问

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

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

9. 直接存储器访问控制器（DMA）

9.1. 简介

DMA 控制器提供了一种硬件的方式在外设和存储器之间或者存储器和存储器之间传输数据，而无需 CPU 的介入，从而使 CPU 可以专注在处理其他系统功能上。DMA 控制器有 7 个通道。每个通道都是专门用来处理一个或多个外设的存储器访问请求的。DMA 控制器内部实现了一个仲裁器，用来仲裁多个 DMA 请求的优先级。

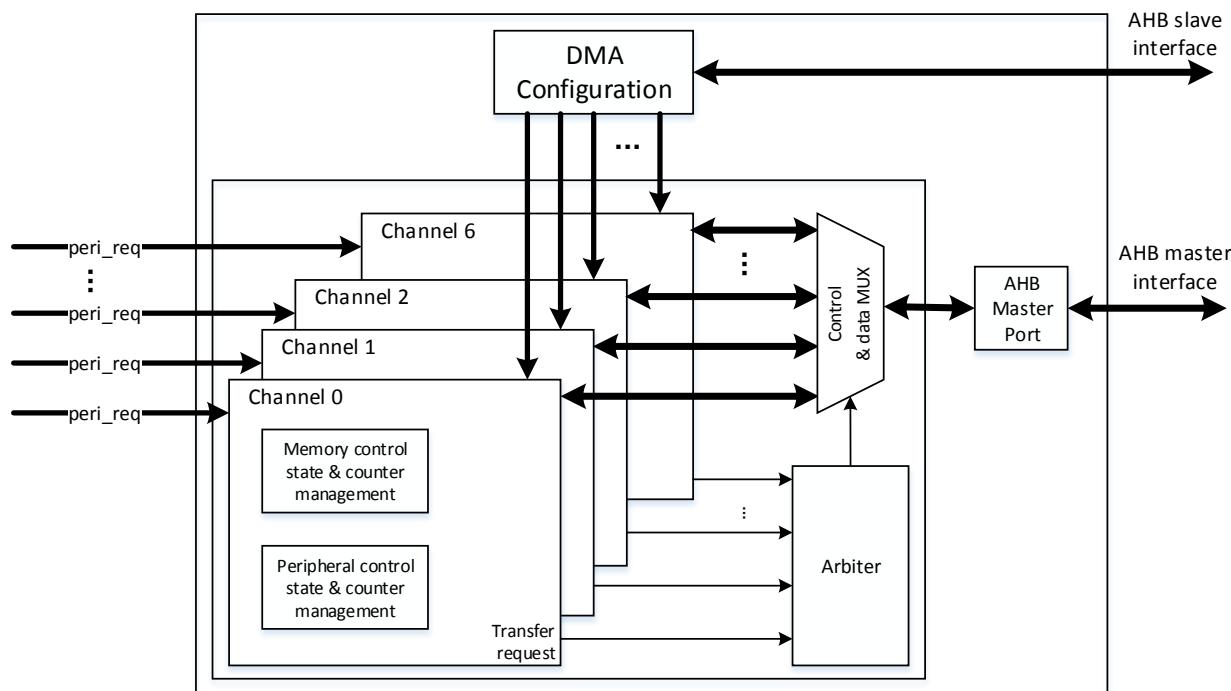
DMA 控制器和 CortexTM-M4 内核共享系统总线。当 DMA 和 CPU 访问同样的地址空间时，DMA 访问可能会阻挡 CPU 访问系统总线几个总线周期。总线矩阵中实现了循环仲裁算法来分配 DMA 与 CPU 的访问权，它可以确保 CPU 得到至少一半的系统总线带宽。

9.2. 主要特性

- 传输数据长度可编程配置，最大到 65536；
- 7 个通道，并且每个通道都可配置；
- AHB 和 APB 外设，片上闪存和 SRAM 都可以作为访问的源端和目的端；
- 每个通道连接固定的硬件 DMA 请求；
- 支持软件优先级（低、中、高、极高）和硬件优先级（通道号越低，优先级越高）；
- 存储器和外设的数据传输宽度可配置：字节，半字，字；
- 存储器和外设的数据传输支持固定寻址和增量式寻址；
- 支持循环传输模式；
- 支持外设到存储器，存储器到外设，存储器到存储器的数据传输；
- 每个通道有 3 种类型的事件标志和独立的中断，支持中断的使能和清除；
- 支持中断使能和清除。

9.3. 结构框图

图 9-1. DMA 结构框图



由 [图 9-1. DMA 结构框图](#) 所示，DMA 控制器由 4 部分组成：

- AHB 从接口配置 DMA
- AHB 主接口进行数据传输
- 仲裁器进行 DMA 请求的优先级管理
- 数据处理和计数

9.4. 功能描述

9.4.1. DMA 操作

DMA 传输分为两步操作：从源地址读取数据，之后将读取的数据存储到目的地址。DMA 控制器基于 DMA_CHxPADDR、DMA_CHxMADDR、DMA_CHxCTL 寄存器的值计算下一次操作的源/目的地址。DMA_CHxCNT 寄存器用于控制传输的次数。DMA_CHxCTL 寄存器的 PWIDHT 和 MWIDHT 位域决定每次发送和接收的字节数（字节/半字/字）。

假设 DMA_CHxCNT 寄存器的值为 4，并且 PNAGA 和 MNAGA 位均置位。结合 PWIDHT 和 MWIDHT 的各种配置，DMA 传输的操作详见 [表 9-1. DMA 传输操作](#)。

表 9-1. DMA 传输操作

传输宽度		传输操作	
源	目标	源	目标
32 bits	32 bits	1: Read B3B2B1B0[31:0] @0x0 2: Read B7B6B5B4[31:0] @0x4 3: Read BBBAB9B8[31:0] @0x8 4: Read BFBEBCDB[31:0] @0xC	1: Write B3B2B1B0[31:0] @0x0 2: Write B7B6B5B4[31:0] @0x4 3: Write BBBAB9B8[31:0] @0x8 4: Write BFBEBCDB[31:0] @0xC
32 bits	16 bits	1: Read B3B2B1B0[31:0] @0x0 2: Read B7B6B5B4[31:0] @0x4 3: Read BBBAB9B8[31:0] @0x8 4: Read BFBEBCDB[31:0] @0xC	1: Write B1B0[7:0] @0x0 2: Write B5B4[7:0] @0x2 3: Write B9B8[7:0] @0x4 4: Write BDBC[7:0] @0x6
32 bits	8 bits	1: Read B3B2B1B0[31:0] @0x0 2: Read B7B6B5B4[31:0] @0x4 3: Read BBBAB9B8[31:0] @0x8 4: Read BFBEBCDB[31:0] @0xC	1: Write B0[7:0] @0x0 2: Write B4[7:0] @0x1 3: Write B8[7:0] @0x2 4: Write BC[7:0] @0x3
16 bits	32 bits	1: Read B1B0[15:0] @0x0 2: Read B3B2[15:0] @0x2 3: Read B5B4[15:0] @0x4 4: Read B7B6[15:0] @0x6	1: Write 0000B1B0[31:0] @0x0 2: Write 0000B3B2[31:0] @0x4 3: Write 0000B5B4[31:0] @0x8 4: Write 0000B7B6[31:0] @0xC
16 bits	16 bits	1: Read B1B0[15:0] @0x0 2: Read B3B2[15:0] @0x2 3: Read B5B4[15:0] @0x4 4: Read B7B6[15:0] @0x6	1: Write B1B0[15:0] @0x0 2: Write B3B2[15:0] @0x2 3: Write B5B4[15:0] @0x4 4: Write B7B6[15:0] @0x6
16 bits	8 bits	1: Read B1B0[15:0] @0x0 2: Read B3B2[15:0] @0x2 3: Read B5B4[15:0] @0x4 4: Read B7B6[15:0] @0x6	1: Write B0[7:0] @0x0 2: Write B2[7:0] @0x1 3: Write B4[7:0] @0x2 4: Write B6[7:0] @0x3
8 bits	32 bits	1: Read B0[7:0] @0x0 2: Read B1[7:0] @0x1 3: Read B2[7:0] @0x2 4: Read B3[7:0] @0x3	1: Write 000000B0[31:0] @0x0 2: Write 000000B1[31:0] @0x4 3: Write 000000B2[31:0] @0x8 4: Write 000000B3[31:0] @0xC
8 bits	16 bits	1: Read B0[7:0] @0x0 2: Read B1[7:0] @0x1 3: Read B2[7:0] @0x2 4: Read B3[7:0] @0x3	1, Write 00B0[15:0] @0x0 2, Write 00B1[15:0] @0x2 3, Write 00B2[15:0] @0x4 4, Write 00B3[15:0] @0x6
8 bits	8 bits	1: Read B0[7:0] @0x0 2: Read B1[7:0] @0x1 3: Read B2[7:0] @0x2 4: Read B3[7:0] @0x3	1, Write B0[7:0] @0x0 2, Write B1[7:0] @0x1 3, Write B2[7:0] @0x2 4, Write B3[7:0] @0x3

DMA_CHxCNT 寄存器的 CNT 位域必须在 CHEN 位置位前被配置，其控制传输的次数。在传输过程中，CNT 位域的值表示还有多少次数据传输将被执行。

将 DMA_CHxCTL 寄存器的 CHEN 位清零，可以停止 DMA 传输。

- 若 CHEN 位被清零时 DMA 传输还未完成，重新使能 CHEN 位将分两种情况：
 - 在重新使能 DMA 通道前，未对该通道的相关寄存器进行操作，则 DMA 将继续完成上次的传输。
 - 在重新使能 DMA 通道前，对任意相关寄存器进行了操作，则 DMA 将开始一次新的传输。
- 若清零 CHEN 位时，DMA 传输已经完成，之后未对任意寄存器进行操作前便使能 DMA 通道，则不会触发任何 DMA 传输。

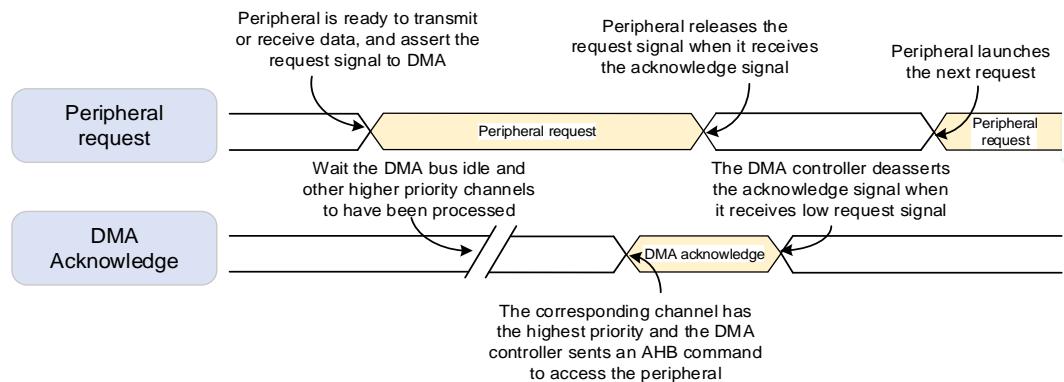
9.4.2. 外设握手

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

- 请求信号：由外设发出，表明外设已经准备好发送或接收数据；
- 应答信号：由 DMA 控制器响应，表明 DMA 控制器已经发送 AHB 命令去访问外设。

[图9-2. 握手机制](#)中详细描述了DMA控制器与外设之间的握手机制。

图 9-2. 握手机制



9.4.3. 仲裁

当DMA控制器在同一时间接收到多个外设请求时，仲裁器将根据外设请求的优先级来决定响应哪一个外设请求。优先级包括软件优先级和硬件优先级，优先级规则如下：

- 软件优先级：分为4级，低，中，高和极高。可以通过寄存器DMA_CHxCTL的PRIO位域来配置。
- 硬件优先级：当通道具有相同的软件优先级时，编号低的通道优先级高。例：通道0和通道2配置为相同的软件优先级时，通道0的优先级高于通道2。

9.4.4. 地址生成

存储器和外设都独立的支持两种地址生成算法：固定模式和增量模式。寄存器DMA_CHxCTL的PNAGA和MNAGA位用来设置存储器和外设的地址生成算法。

在固定模式中，地址一直固定为初始化的基地址（DMA_CHxPADDR, DMA_CHxMADDR）。

在增量模式中，下一次传输数据的地址是当前地址加1（或者2, 4），这个值取决于数据传输宽度。

9.4.5. 循环模式

循环模式用来处理连续的外设请求(如ADC扫描模式)。将DMA_CHxCTL寄存器的CMEN位置

位可以使能循环模式。

在循环模式中，当每次DMA传输完成后，CNT值会被重新载入，且传输完成标志位会被置1。DMA会一直响应外设的请求，直到通道使能位（DMA_CHxCTL寄存器的CHEN位）被清0。

9.4.6. 存储器到存储器模式

将DMA_CHxCTL寄存器的M2M位置位可以使能存储器到存储器模式。在此模式下，DMA通道传输数据时不依赖外设的请求信号。一旦DMA_CHxCTL寄存器的CHEN位被置1，DMA通道就立即开始传输数据，直到DMA_CHxCNT寄存器达到0，DMA通道才会停止。

9.4.7. 通道配置

要启动一次新的DMA数据传输，建议遵循以下步骤进行操作：

1. 读取 CHEN 位，如果为 1（通道已使能），清零该位。当 CHEN 为 0 时，请按照下列步骤配置 DMA 开始新的传输。
2. 配置 DMA_CHxCTL 寄存器的 M2M 及 DIR 位，选择传输模式。
3. 配置 DMA_CHxCTL 寄存器的 CMEN 位，选择是否使能循环模式。
4. 配置 DMA_CHxCTL 寄存器的 PRIO 位域，选择该通道的软件优先级。
5. 通过 DMA_CHxCTL 寄存器配置存储器和外设的传输宽度以及存储器和外设地址生成算法。
6. 通过 DMA_CHxCTL 寄存器配置传输完成中断，半传输完成中断，传输错误中断的使能位。
7. 通过 DMA_CHxPADDR 寄存器配置外设地址。
8. 通过 DMA_CHxMADDR 寄存器配置存储器地址。
9. 通过 DMA_CHxCNT 寄存器配置数据传输总量。
10. 将 DMA_CHxCTL 寄存器的 CHEN 位置 1，使能 DMA 通道。

9.4.8. 中断

每个DMA通道都有一个专用的中断。中断事件有三种类型：传输完成，半传输完成和传输错误。

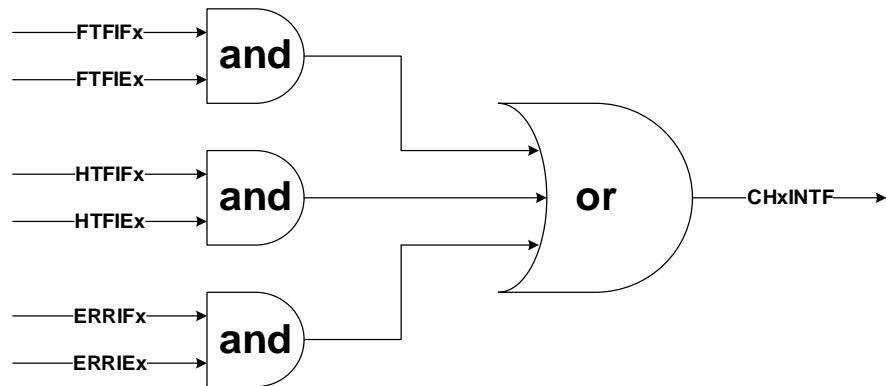
每一个中断事件在DMA_INTF寄存器中有专用的标志位，在DMA_INTC寄存器中有专用的清除位，在DMA_CHxCTL寄存器中有专用的使能位。[表9-2. 中断事件](#)描述了其对应关系。

表 9-2. 中断事件

中断事件	标志位	清除位	使能位
	DMA_INTF	DMA_INTC	DMA_CHxCTL
传输完成	FTFIF	FTFIFC	FTFIE
传输半完成	HTFIF	HTFIFC	HTFIE
传输错误	ERRIF	ERRIFC	ERRIE

DMA中断逻辑如[表9-3. DMA各通道请求表](#)所示，任何类型中断使能时，产生了相应中断事件均会产生中断。

图 9-3. DMA 中断逻辑图



注意：“x”表示通道数（对应 $x=0\ldots 6$ ）

9.4.9. DMA 请求映射

多个外设请求被映射到同一个 DMA 通道。这些请求信号在经过逻辑或后进入 DMA。详情可见[图 9-4. DMA 请求映射](#)。通过配置对应外设的寄存器，每个外设的请求均可以独立的开启或关闭。用户必须确保同一时间，在同一个通道上仅有 一个外设的请求被开启。[表 9-3. DMA 各通道请求表](#)列举了 DMA 的每个通道所支持的外设请求。

图 9-4. DMA 请求映射

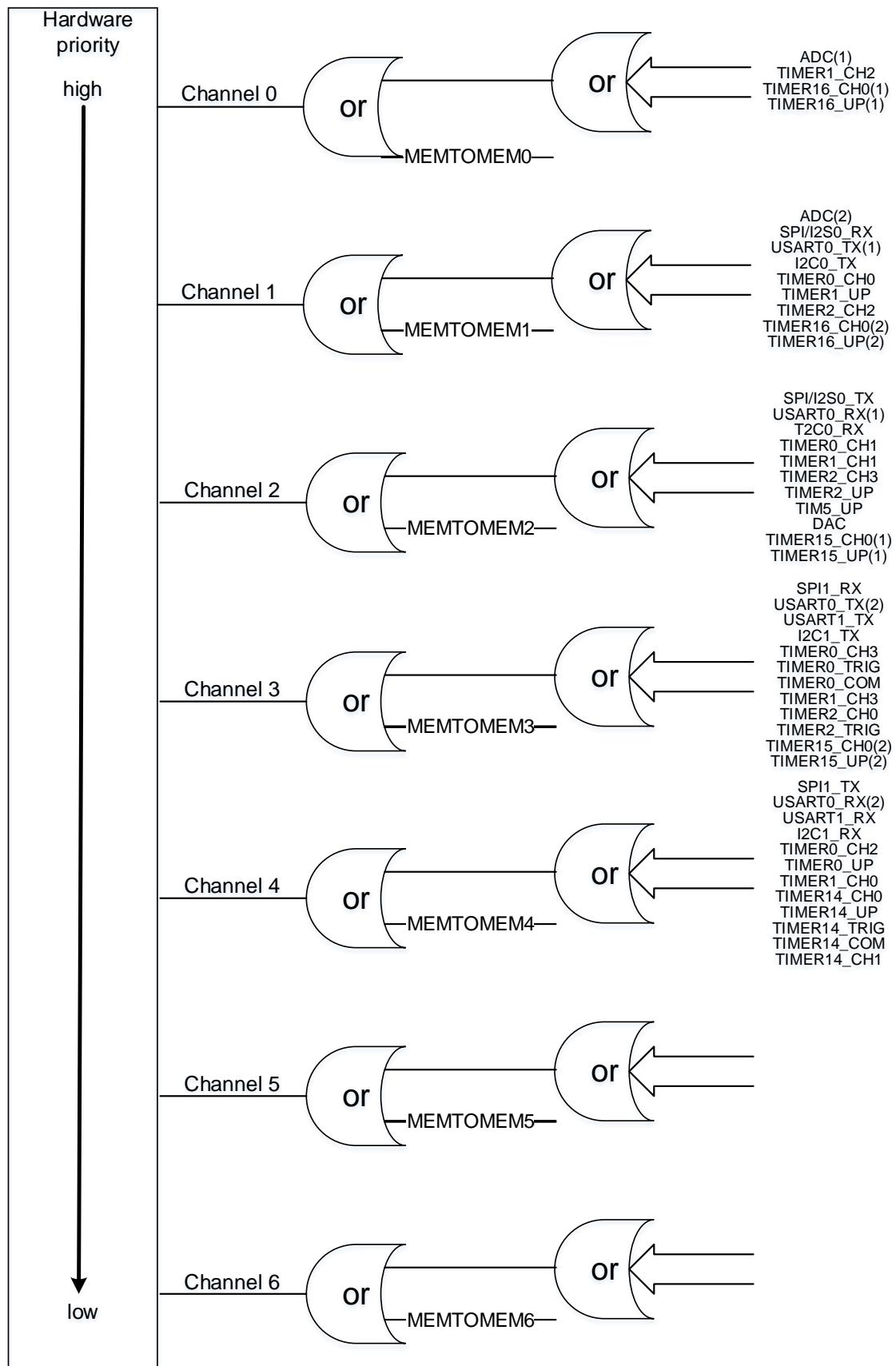


表9-3. DMA各通道请求表

外设	通道 0	通道 1	通道 2	通道 3	通道 4	通道 5	通道 6
ADC	ADC(1)	ADC(2)	•	•	•	•	•
SPI/I2S	•	SPI/I2S0_RX	SPI/I2S0_TX	SPI1_RX	SPI1_TX	•	•
USART	•	USART0_TX(1)	USART0_RX(1)	USART0_RX(2) USART1_TX	USART0_RX(2) USART1_RX	•	•
I ² C	•	I2C0_TX	I2C0_RX	I2C1_TX	I2C1_RX	•	•
TIMER0	•	TIMER0_CH0	TIMER0_CH1	TIMER0_CH3 TIMER0_TRIG TIMER0_COM	TIMER0_CH2 TIMER0_UP	•	•
TIMER1	TIMER1_CH2	TIMER1_UP	TIMER1_CH1	TIMER1_CH3	TIMER1_CH0	•	•
TIMER2	•	TIMER2_CH2	TIMER2_CH3 TIMER2_UP	TIMER2_CH0 TIMER2_TRIG	•	•	•
TIMER5/ DAC	•	•	TIMER5_UP DAC	•	•	•	•
TIMER14	•	•	•	•	TIMER14_CH0 TIMER14_UP TIMER14_TRIG TIMER14_COM TIMER14_CH1	•	•
TIMER15	•	•	TIMER15_CH0(1) TIMER15_UP(1)	TIMER15_CH0(2) TIMER15_UP(2)	•	•	•
TIMER16	TIMER16_CH0(1) TIMER16_UP(1)	TIMER16_CH0(2) TIMER16_UP(2)	•	•	•	•	•

(1). 当 SYSCFG_CFGR0 寄存器的相应重映射位被清零时，请求被映射到该通道。

(2). 当 SYSCFG_CFGR0 寄存器的相应重映射位被置位时，请求被映射到该通道。

9.5. DMA 寄存器

9.5.1. 中断标志位寄存器 (DMA_INTF)

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		保留		ERRIF6	HTFIF6	FTFIF6	GIF6	ERRIF5	HTFIF5	FTFIF5	GIF5	ERRIF4	HTFIF4	FTFIF4	GIF4
				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIF3	HTFIF3	FTFIF3	GIF3	ERRIF2	HTFIF2	FTFIF2	GIF2	ERRIF1	HTFIF1	FTFIF1	GIF1	ERRIFO	HTFIFO	FTFIFO	GIF0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:28	保留	必须保持复位值
27/23/19/	ERRIFx	通道x错误标志位(x=0...6) 硬件置位, 软件写DMA_INTC相应位为1清零
15/11/7/3		0: 通道x未发生传输错误 1: 通道x发生传输错误
26/22/18/	HTFIFx	通道x半传输完成标志位(x=0...6) 硬件置位, 软件写DMA_INTC相应位为1清零
14/10/6/2		0: 通道x半传输未完成 1: 通道x半传输完成
25/21/17/	FTFIFx	通道x传输完成标志位(x=0...6) 硬件置位, 软件写DMA_INTC相应位为1清零
13/9/5/1		0: 通道x传输未完成 1: 通道x传输完成
24/20/16/	GIFx	通道x全局中断标志位(x=0...6) 硬件置位, 软件写DMA_INTC相应位为1清零
12/8/4/0		0: 通道x ERRIF, HTFIF或FTFIF标志位未置位 1: 通道x至少发生ERRIF, HTFIF或FTFIF之一置位

9.5.2. 中断标志位清除寄存器 (DMA_INTC)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		保留		ERRIFC6	HTFIFC6	FTFIFC6	GIFC6	ERRIFC5	HTFIFC5	FTFIFC5	GIFC5	ERRIFC4	HTFIFC4	FTFIFC4	GIFC4
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIFC3	HTFIFC3	FTFIFC3	GIFC3	ERRIFC2	HTFIC2	FTFIFC2	GIFC2	ERRIFC1	HTFIFC1	FTFIFC1	GIFC1	ERRIFC0	HTFIFC0	FTFIFC0	GIFC0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位/位域	名称	描述
31:28	保留	必须保持复位值
27/23/19/15/11/7/3	ERRIFCx	清除通道x(x=0...6)的错误标志位 0: 无影响 1: 清零DMA_INTF寄存器的ERRIFx位
26/22/18/14/10/6/2	HTFIFCx	清除通道x(x=0...6)的半传输完成标志位 0: 无影响 1: 清零DMA_INTF寄存器的HTFIFx位
25/21/17/13/9/5/1	FTFIFCx	清除通道x(x=0...6)的传输完成标志位 0: 无影响 1: 清零DMA_INTF寄存器的FTFIFx位
24/20/16/12/8/4/0	GIFCx	清除通道x(x=0...6)的全局中断标志位 0: 无影响 1: 清零DMA_INTF寄存器的GIFx, ERRIFx, HTFIFx和FTFIFx位

9.5.3. 通道 x 控制寄存器 (DMA_CHxCTL)

x = 0...6, x 为通道序号

地址偏移: 0x08 + 0x14 × x

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问

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

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

		1: 使能存储器到存储器模式 CHEN位为1时，该位不能被配置
13:12	PRIOR[1:0]	软件优先级 软件置位和清零 00: 低 01: 中 10: 高 11: 极高 CHEN位为1时，该位域不能被配置
11:10	MWIDTH[1:0]	存储器的传输数据宽度 软件置位和清零 00: 8-bit 01: 16-bit 10: 32-bit 11: 保留 CHEN位为1时，该位域不能被配置
9:8	PWIDTH[1:0]	外设的传输数据宽度 软件置位和清零 00: 8-bit 01: 16-bit 10: 32-bit 11: 保留 CHEN位为1时，该位域不能被配置
7	MNAGA	存储器的地址生成算法 软件置位和清零 0: 固定地址模式 1: 增量地址模式 CHEN位为1时，该位不能被配置
6	PNAGA	外设的地址生成算法 软件置位和清零 0: 固定地址模式 1: 增量地址模式 CHEN位为1时，该位不能被配置
5	CMEN	循环模式使能 软件置位和清零 0: 禁止循环模式 1: 使能循环模式 CHEN位为1时，该位不能被配置
4	DIR	传输方向 软件置位和清零

		0: 从外设读出并写入存储器 1: 从存储器读出并写入外设 CHEN位为1时，该位不能被配置
3	ERRIE	通道错误中断使能位 软件置位和清零 0: 禁止通道错误中断 1: 使能通道错误中断
2	HTFIE	通道半传输完成中断使能位 软件置位和清零 0: 禁止通道半传输完成中断 1: 使能通道半传输完成中断
1	FTFIE	通道传输完成中断使能位 软件置位和清零 0: 禁止通道传输完成中断 1: 使能通道传输完成中断
0	CHEN	通道使能 软件置位和清零 0: 禁止该通道 1: 使能该通道

9.5.4. 通道 x 计数寄存器 (DMA_CHxCNT)

x = 0...6, x 为通道序号

地址偏移: 0x0C + 0x14 × x

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问



位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CNT[15:0]	传输计数 CHEN位为1时，该位域不能被配置 该寄存器标明还有多少数据等待被传输。一旦通道使能，该寄存器为只读的，并在每个 DMA 传输之后值减1。如果该寄存器的值为0，无论通道开启与否，都不会有数据传输。 如果该通道工作在循环模式下，一旦通道的传输任务完成，该寄存器会被自动重装载为初

始设置值。

9.5.5. 通道 x 外设基地址寄存器 (DMA_CHxPADDR)

$x = 0 \dots 6$, x 为通道序号

地址偏移: $0x10 + 0x14 \times x$

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问

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

位/位域	名称	描述
31:0	PADDR[31:0]	外设基地址 CHEN位为1时，该位域不能被配置 当PWIDTH位域的值为01 (16-bit),, PADDR[0]被忽略，访问自动与16位地址对齐。 当PWIDTH位域的值为10 (32-bit), PADDR [1:0]被忽略，访问自动与32位地址对齐。

9.5.6. 通道 x 存储器基地址寄存器 (DMA_CHxMADDR)

$x = 0 \dots 6$, x 为通道序号

地址偏移: $0x14 + 0x14 \times x$

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问

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

位/位域	名称	描述
31:0	MADDR[31:0]	存储器基地址 CHEN位为1时，该位域不能被配置 当MWIDTH位域的值为01 (16-bit)时，MADDR [0]被忽略，访问自动与16位地址对齐。 当MWIDTH位域的值为10 (32-bit)时，MADDR [1:0]被忽略，访问自动与32位地址对齐。

10. 调试 (DBG)

10.1. 简介

GD32F3x0系列产品提供了各种各样的调试，跟踪和测试功能。这些功能通过ARM CoreSight™组件的标准配置和链状连接的TAP控制器来实现的。调试和跟踪功能集成在ARM Cortex-M4内核中。调试系统支持串行调试（SWD）和跟踪功能。调试和跟踪功能请参考下列文档：

- Cortex-M4技术参考手册；
- ARM调试接口v5结构规范。

调试系统帮助调试者在低功耗模式下调试以及一些外设调试，包括：TIMER、I2C、RTC、WWDGT和FWDGT。当相应的位被置1，调试系统会在低功耗模式下提供时钟，或者为一些外设保持当前状态，这些外设包括：TIMER、I2C、RTC、WWDGT和FWDGT。

10.2. 串行调试接口简介

调试工具可以通过串行调试接口（SWD）来访问调试功能。

10.2.1. 引脚分配

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

调试引脚分配：

PA14 : SWCLK

PA13 : SWDIO

如果SWD没有使用，这两个引脚均释放作为普通GPIO功能。两个引脚具体配置请参考[通用和备用输入输出接口（GPIO和AFIO）](#)。

10.2.2. JEDEC-106 ID code

Cortex-M4集成了JEDEC-106 ID代码。位于ROM表中，映射地址为0xE00FF000_0xE00FFFF。

10.3. 调试保持功能描述

10.3.1. 低功耗模式调试支持

当DBG控制寄存器0（DBG_CTL0）的STB_HOLD位置1并且进入待机模式，AHB总线时钟和系统时钟由CK_IRC8M提供，可以在待机模式下调试。当退出待机模式后，产生系统复位。

当DBG控制寄存器0（DBG_CTL0）的DSLP_HOLD位置1并且进入深度睡眠模式，AHB总线时

钟和系统时钟由CK_IRC8M提供，可以在深度睡眠模式下调试。

当DBG控制寄存器0（DBG_CTL0）的SLP_HOLD位置1并且进入睡眠模式，AHB总线时钟没有关闭，可以在睡眠模式下调试。

10.3.2. TIMER, I2C, RTC, WWDGT 和 FWDGT 外设调试支持

当内核停止，并且DBG控制寄存器0（DBG_CTL0）或DBG控制寄存器1（DBG_CTL1）中的相应位置1。对于不同外设，有不同动作：

对于**TIMER**外设，**TIMER**计数器停止并进行调试；

对于**I2C**外设，**SMBUS**保持状态并进行调试；

对于**RTC**外设，计数器停止并进行调试；

对于**WWDGT**或者**FWDGT**外设，计数器时钟停止并进行调试。

10.4. DBG 寄存器

10.4.1. ID 寄存器 (DBG_ID)

地址: 0xE004 2000

只读寄存器

该寄存器只能按字(32位)访问

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

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

10.4.2. 控制寄存器 0 (DBG_CTL0)

地址偏移: 0x04

复位值: 0x0000 0000, 仅上电复位

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留.				TIMER13_HOLD	保留				TIMER5_HOLD	保留		I2C1_HOL	LD		
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C0_HOL	保留	TIMER2_HOLD	TIMER1_HOLD	TIMER0_HOLD	WWDGTHOLD	FWDGTHOLD	保留				STB_HOLD	DSLP_HOLD	SLP_HOLD		
rw				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

位/位域	名称	描述
31:28	保留	必须保持复位值
27	TIMER13_HOLD	TIMER 13 保持位 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持 TIMER 13 计数器不变，用于调试
26:20	保留	必须保持复位值
19	TIMER5_HOLD	TIMER 5 保持位 该位由软件置位和复位

		0: 无影响 1: 当内核停止时保持 TIMER 5 计数器不变, 用于调试
18:17	保留	必须保持复位值
16	I2C1_HOLD	I2C1 保持位 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持 I2C1 的 SMBUS 状态不变, 用于调试
15	I2C0_HOLD	I2C0 保持位 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持 I2C0 的 SMBUS 状态不变, 用于调试
14:13	保留	必须保持复位值
12	TIMER2_HOLD	TIMER 2 保持位 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持 TIMER 2 计数器不变, 用于调试
11	TIMER1_HOLD	TIMER 1 保持位 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持 TIMER 1 计数器不变, 用于调试
10	TIMER0_HOLD	TIMER 0 保持位 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持 TIMER 0 计数器不变, 用于调试
9	WWDGT_HOLD	WWDGT 保持位 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持 WWDGT 计数器时钟, 用于调试
8	FWDGT_HOLD	FWDGT 保持位 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持 FWDGT 计数器时钟, 用于调试
7:3	保留	必须保持复位值
2	STB_HOLD	待机模式保持位 该位由软件置位和复位 0: 无影响 1: 在待机模式下系统时钟和 AHB 时钟由 CK_IRC8M 提供, 当退出待机模式时, 产生系统复位

1	DSLP_HOLD	深度睡眠模式保持位 该位由软件置位和复位 0: 无影响 1: 在待机模式下系统时钟和 AHB 时钟由 CK_IRC8M 提供, 当退出待机模式时, 产生系统复位
0	SLP_HOLD	睡眠模式保持位 该位由软件置位和复位 0: 无影响 1: 在睡眠模式下, AHB 时钟继续运行

10.4.3. 控制寄存器 1 (DBG_CTL1)

地址偏移: 0x08

复位值: 0x0000 0000, 仅上电复位

该寄存器只能按字(32 位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留.														TIMER16_HOLD	TIMER15_HOLD	TIMER14_HOLD
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
保留														保留		

rw rw rw

位/位域	名称	描述
31:19	保留	必须保持复位值
18	TIMER16_HOLD	TIMER 16 保持位 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持 TIMER 16 计数器不变, 用于调试
17	TIMER15_HOLD	TIMER 15 保持位 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持 TIMER 15 计数器不变, 用于调试
16	TIMER14_HOLD	TIMER 14 保持位 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持 TIMER 14 计数器不变, 用于调试
15:11	保留	必须保持复位值
10	RTC_HOLD	RTC 保持位

该位由软件置位和复位

0: 无影响

1: 当内核停止时保持 RTC 计数器不变，用于调试

9:0

保留

必须保持复位值

11. 模数数字转换器(ADC)

11.1. 简介

12 位 ADC 是一种采用逐次逼近方式的模拟数字转换器。它有 19 个多路复用通道，可以测量来自 16 个外部通道，2 个内部通道和电池电压 (V_{BAT}) 通道的模拟信号。模拟看门狗允许应用程序来检测输入电压是否超出用户设定的高低阈值。每个通道的 A/D 转换可以配置成单次、连续、扫描或间断转换模式。ADC 转换的结果可以按照左对齐或右对齐的方式存储在 16 位数据寄存器中。片上的硬件过采样机制可以通过减少 MCU 的相关计算负担来提高性能。

11.2. 主要特性

- 高性能:
 - 可配置 12 位、10 位、8 位或 6 位分辨率
 - 自校准
 - 可编程的采样时间
 - 数据寄存器可配置数据对齐方式
 - 支持规则通道数据转换的 DMA 请求
- 双时钟域架构 (APB 时钟和 ADC 时钟)
- 模拟输入通道:
 - 16 个外部模拟输入通道
 - 1 个内部温度传感通道 (V_{SENSE})
 - 1 个内部参考电压输入通道 (V_{REFINT})
 - 1 个监测外部 V_{BAT} 供电引脚的内部输入通道
- 转换开始的触发:
 - 软件方式
 - 配置极性进行硬件触发 (来自 TIMER0、TIMER1、TIMER2 和 TIMER14 的内部定时器事件)
 - 规则通道和注入通道的外部触发
- 转换模式:
 - 转换单个通道，或者扫描一序列的通道
 - 单次模式，每次触发转换一个选择的输入通道

- 连续模式，连续转换所选择的输入通道
- 间断模式
- 规则组转换完成，注入组转换完成和模拟看门狗事件都可以产生中断
- 模拟看门狗
- ADC供电要求：2.6V到3.6V，一般电源电压为3.3V
- 过采样：
 - 16位的数据寄存器
 - 可调整的过采样率，范围从2x到256x
 - 高达8位的可编程数据移位
- ADC输入范围： $V_{SSA} \leq V_{IN} \leq V_{DDA}$

11.3. 引脚和内部信号

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

表 11-1. ADC 内部信号

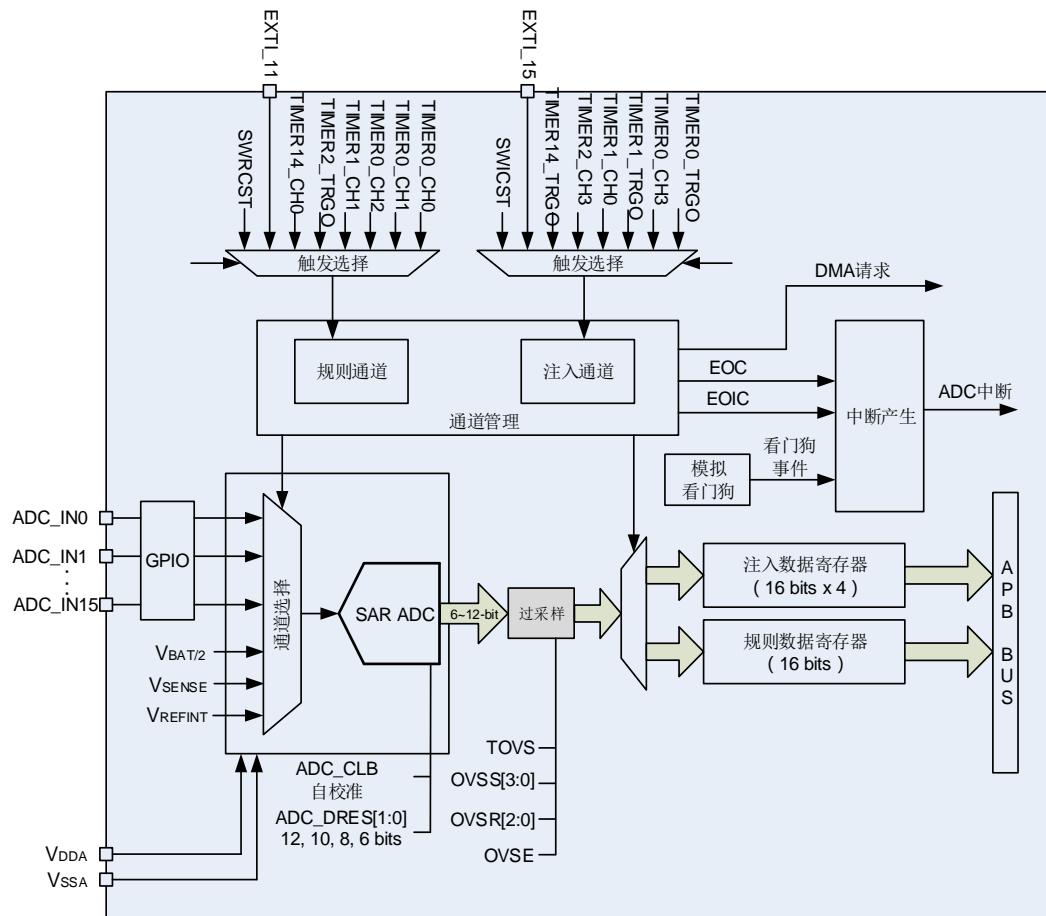
内部信号名称	信号类型	说明
V_{SENSE}	输入	内部温度传感器输出电压
V_{REFINT}	输入	内部电压参考输出电压
$V_{BAT}/2$	输入	硬件输入电压除以2

表 11-2. ADC 引脚定义

名称	信号类型	注释
$V_{DDA}^{(1)}$	输入，模拟供电电源	模拟电源输入等于 V_{DD} , $2.6\text{ V} \leq V_{DDA} \leq 3.6\text{ V}$
$V_{SSA}^{(1)}$	输入，模拟电源地	模拟地，等于 V_{SS}
ADCx_IN [15:0]	输入，模拟信号	多达16路外部通道

11.4. 功能描述

图 11-1. ADC 模块框图



11.4.1. 校准 (ADC_CLB)

ADC带有一个前置校准功能。在校准期间，ADC计算一个校准系数，这个系数应用于ADC内部的，直到ADC下次掉电才无效。在校准期间，应用程序不能使用ADC，必须等到ADC校准完成。在A/D开始转换前应执行校准操作。通过软件设置CLB=1，来对初始化校准功能，在整个校准期间，CLB位一直保持为1，直到校准完成，该位由硬件清0。

当ADC运行条件改变(例如供电电压V_{DDA}、温度等)时，建议重新执行一次校准操作。

内部的模拟校准可以通过复位ADC_CTL1寄存器的RSTCLB位来重置。

软件校准过程：

1. 确保 ADCON=1
2. 延迟 14 个 ADCCLK 以等待 ADC 稳定；
3. 设置 RSTCLB (可选的)

4. 设置 CLB=1
5. 等待直到 CLB =0

11.4.2. 双时钟域架构

除了APB接口时钟，ADC的子模块时钟还可以由ADC时钟提供。ADC时钟和APB时钟异步，并独立于APB时钟。

应用程序能够在低功耗运行时，降低PLCK时钟频率，同时ADC仍能保持最佳运行状态。

想要更多ADC时钟产生的信息，可以参考RCU章节的[4.2.1](#)部分。

11.4.3. ADCON 开关

ADC_CTL1寄存器中的ADCON位是ADC模块的使能开关。如果ADCON位为0，则ADC模块保持复位状态。为了省电，当ADCON位为0时，ADC模拟子模块将会进入掉电模式。

11.4.4. 规则组和注入组

ADC支持19个多路复用通道，并且把转换通道分成两组：规则组通道和注入组通道。

规则组中，可以按照特定的序列组成多达16个转换的序列。ADC_RSQ0~ADC_RSQ2寄存器规定了规则组的通道选择。ADC_RSQ0寄存器中的RL[3:0]位规定了整个规则组转换序列的长度。

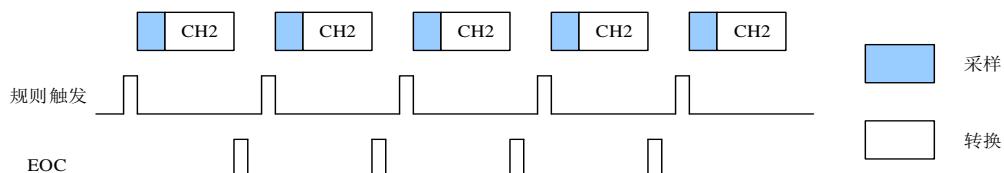
注入组中，可以按照特定的序列组成多达4个转换的序列。ADC_ISQ寄存器规定了注入组的通道选择。ADC_ISQ寄存器的IL[1:0]位规定了整个注入组转换序列的长度。

11.4.5. 转换模式

单次转换模式

该模式能够运行在规则组和注入组。单次转换模式下，ADC_RSQ2寄存器的RSQ0[4:0]位（规则组）或者ADC_ISQ寄存器的ISQ3[4:0]位（注入组）规定了ADC的转换通道。当ADCON位被置1时，一旦相应的软件触发或者外部触发产生，ADC就会采样和转换一个通道。

图 11-2. 单次转换模式



规则通道一次单次转换结束后，转换数据将被存放于ADC_RDATA寄存器中，EOC位置1。如果EOCIE位被置1，将产生一个中断。

注入通道一次单次转换结束后，转换数据将被存放于ADC_IDATA0寄存器中，EOC位和EOIC

位将会置1。如果EOCIE或EOICIE位被置1，将产生一个中断。

规则组单次转换模式的软件流程：

1. 确保ADC_CTL0寄存器中的DISRC位和SM位以及ADC_CTL1寄存器中的CTN位为0
2. 根据模拟通道编号来配置RSQ0
3. 配置ADC_SAMPTx寄存器
4. 如果有需要，可以配置ADC_CTL1寄存器中的ETERC位和ETSRC位
5. 设置SWRCST位，或者为规则组产生一个外部触发信号
6. 等待EOC标志位置1
7. 从ADC_RDATA寄存器中读ADC转换结果
8. 写0清除EOC标志位

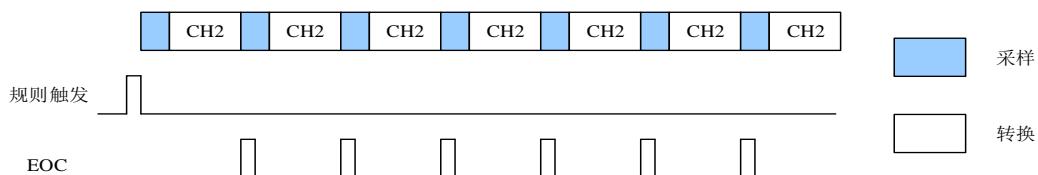
注入组单次转换模式的软件流程：

1. 确保ADC_CTL0寄存器的DISRC位和SM位为0
2. 根据模拟通道编号来配置ISQ3
3. 配置ADC_SAMPTx寄存器
4. 如果有需要，可以配置ADC_CTL1寄存器的ETEIC和ETSIC位
5. 设置SWICST位，或者为注入组产生一个外部触发信号
6. 等待EOC/EOIC标志位置1
7. 从ADC_IDATA0寄存器中读ADC转换结果
8. 写0清除EOC/EOIC标志位

连续转换模式

该模式可以运行在规则组通道上。当ADC_CTL1寄存器中的CTN位置1时，可以使能连续转换模式。该模式下，ADC执行由RSQ0[4:0]规定的转换通道。当ADCON位被置1，一旦相应的软件触发或者外部触发产生，ADC就会采样和转换规定的通道。转换数据保存在ADC_RDATA寄存器中。

图 11-3. 连续转换模式



规则组连续转换模式的软件流程：

1. 设置ADC_CTL1寄存器中的CTN位为1
2. 根据模拟通道编号来配置RSQ0
3. 配置ADC_SAMPTx寄存器
4. 如果有需要，配置ADC_CTL1寄存器的ETERC和ETSRC位
5. 设置SWRCST位，或者给规则组产生一个外部触发信号
6. 等待EOC标志位置1
7. 从ADC_RDATA寄存器中读ADC转换结果
8. 写0清除EOC标志位

9. 如果需要进行连续转换，重复步骤6~8

由于要循环查询EOC标志位，可以使用DMA来传输转换数据，软件流程如下：

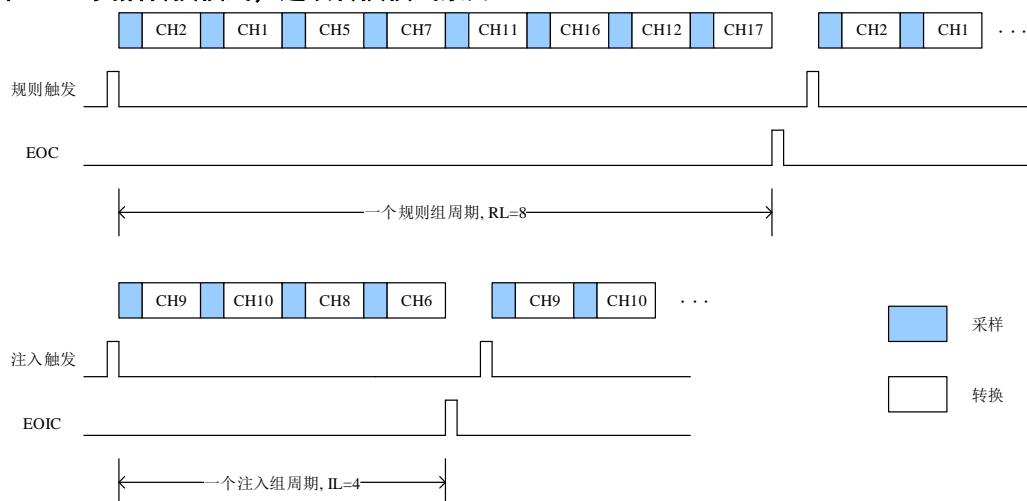
1. 设置ADC_CTL1寄存器的CTN位为1
2. 根据模拟通道编号配置RSQ0
3. 配置ADC_SAMPTx寄存器
4. 如果有需要，配置ADC_CTL1寄存器的ETERC位和ETSRC位
5. 准备DMA模块，用于传输ADC_RDATA寄存器的数据
6. 设置SWRCST位，或者给规则组产生一个外部触发

扫描转换模式

扫描转换模式可以通过将ADC_CTL0寄存器的SM位置1来使能。该模式下，ADC扫描转换所有被ADC_RSQ0~ADC_RSQ2寄存器或ADC_ISQ寄存器选中的通道。一旦ADCON位被置1，当相应的软件触发或者外部触发产生，ADC就会一个接一个的采样和转换规则组或注入组通道。转换数据存储在ADC_RDATA或ADC_IDATAx寄存器中。规则组或注入组转换结束后，EOC位或EOIC位将被置1。如果EOCIE或EOICIE位被置1，将产生一个中断。当规则组通道工作在扫描模式下时，ADC_CTL1寄存器的DMA位必须设置为1。

如果ADC_CTL1寄存器的CTN位也被置1，则在规则通道转换完之后，这个转换自动重新开始。

图 11-4. 扫描转换模式，连续转换模式禁用



规则组扫描转换模式的软件流程：

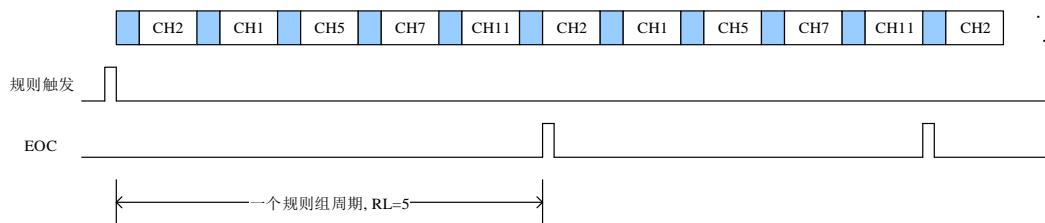
1. 设置 ADC_CTL0 寄存器的 SM 位和 ADC_CTL1 寄存器的 DMA 位为 1
2. 配置 ADC_RSQx 和 ADC_SAMPTx 寄存器
3. 如果有需要，配置 ADC_CTL1 寄存器中的 ETERC 和 ETSRC 位
4. 准备 DMA 模块，用于传输 ADC_RDATA 寄存器的数据
5. 设置 SWRCST 位，或者给规则组产生一个外部触发
6. 等待 EOC 标志位置 1
7. 写 0 清除 EOC 标志位

注入组扫描转换模式的软件流程：

1. 设置 ADC_CTL0 寄存器的 SM 位为 1
2. 配置 ADC_ISQ 和 ADC_SAMPTx 寄存器

3. 如果有需要，配置 ADC_CTL1 寄存器中的 ETEIC 和 ETSIC 位
4. 设置 SWRCST 位，或者给注入组产生一个外部触发
5. 等待 EOC/EOIC 标志位置 1
6. 读 ADC_IDATAx 寄存器中的转换结果
7. 写 0 清除 EOC/EOIC 标志位

图 11-5. 扫描转换模式，连续转换模式使能



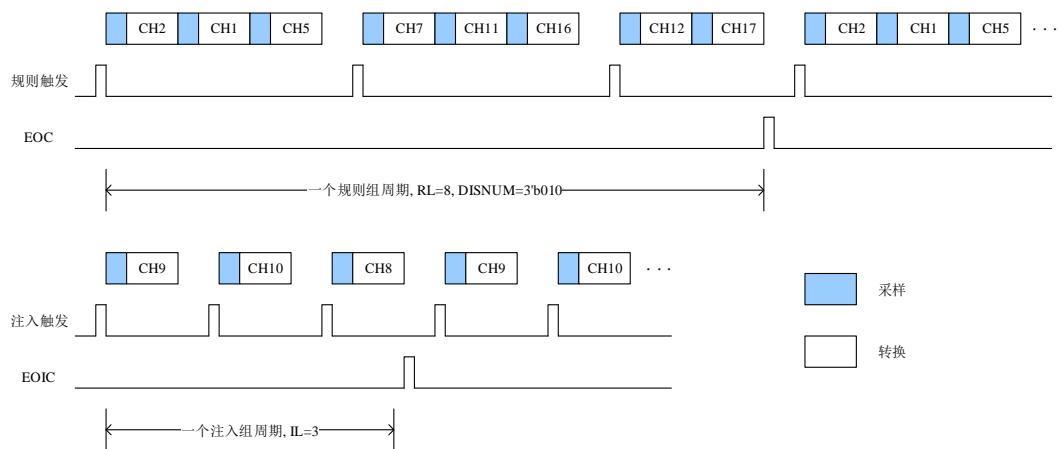
间断模式

对于规则组，将 ADC_CTL0 寄存器的 DISRC 位置 1 可以使能间断转换模式。该模式下，可以执行一次 n 个通道的短序列转换($n \leq 8$)，该转换是由 ADC_RSQ0~RSQ2 寄存器所选择的转换序列的一部分。数值 n 由 ADC_CTL0 寄存器的 DISNUM[2:0]位给出。当相应的软件触发或外部触发产生时，ADC 就会采样和转换在 ADC_RSQ0~RSQ2 寄存器所选择通道中剩下的 n 个通道，直到规则序列中所有的通道转换完成。每个规则组短序列转换周期结束后，EOC 位将被置 1。如果 EOICIE 位被置 1，将产生一个中断。

对于注入组，将 ADC_CTL0 寄存器的 DISIC 位置 1 可以使能间断转换模式。该模式下，可以对 ADC_ISQ 寄存器所选择的转换序列中的一个通道进行转换。当相应的软件触发或外部触发产生时，ADC 就会采样和转换 ADC_ISQ 寄存器中所选择通道中的下一个通道，直到注入组序列中所有通道转换完成。每个注入组通道转换周期结束后，EOIC 位将被置 1。如果 EOICIE 位被置 1，将产生一个中断。

规则组和注入组不能同时工作在间断模式，同一时刻只能有一组被设置成间断模式。

图 11-6. 间断转换模式



规则组间断模式的软件流程：

1. 设置 ADC_CTL0 寄存器的 DISRC 位和 ADC_CTL1 寄存器的 DMA 位为 1
2. 配置 ADC_CTL0 寄存器的 DISNUM[2:0]位

3. 配置 ADC_RSQx 和 ADC_SAMPTx 寄存器
4. 如果有需要，配置 ADC_CTL1 寄存器中的 ETERC 位和 ETSRC 位
5. 准备 DMA 模块，用于传输 ADC_RDATA 寄存器中的数据
6. 设置 SWRCST 位，或者给规则组产生一个外部触发
7. 如果有需要，重复步骤 6
8. 等待 EOC 标志位置 1
9. 写 0 清除 EOC 标志位

注入组间断模式的软件流程：

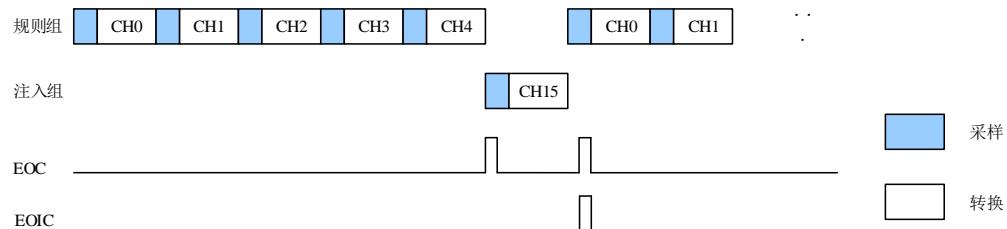
1. 设置 ADC_CTL0 寄存器的 DISRC 位为 1
2. 配置 ADC_ISQ 和 ADC_SAMPTx 寄存器
3. 如果有需要，配置 ADC_CTL1 寄存器中的 ETEIC 位和 ETSIC 位
4. 设置 SWICST 位，或者给注入组产生一个外部触发
5. 如果需要，重复步骤 4
6. 等待 EOC/EOIC 标志位置 1
7. 读 ADC_IDATAx 寄存器中的转换结果
8. 写 0 清除 EOC、EOIC 标志位

11.4.6. 注入通道管理

自动注入

如果将 ADC_CTL0 寄存器的 ICA 位置 1，在规则组通道转换之后，注入组通道将自动转换。该模式下，注入组通道的外部触发不能被使能。该模式可以转换 ADC_RSQ0~ADC_RSQ2 寄存器和 ADC_ISQ 寄存器中设置的多达 20 个转换序列。除了 ICA 位之外，如果 CTN 位也被置 1，注入组通道将在规则组通道转换之后自动转换。

图 11-7. 自动注入，CTN=1

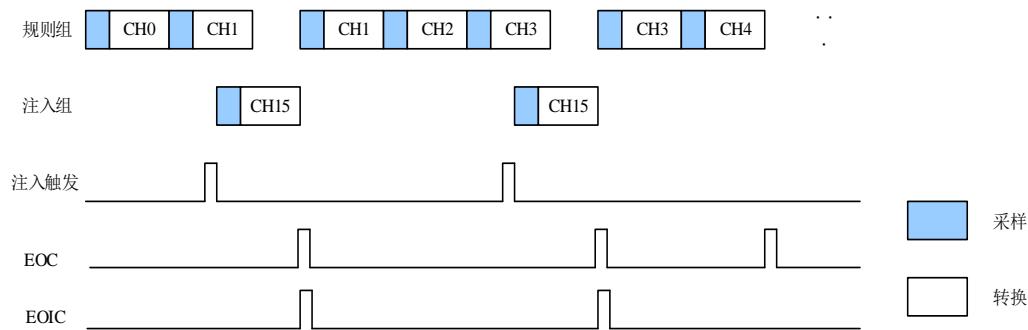


不能同时使用自动注入和间断模式。

触发注入

清除 ICA 位，在规则组通道转换期间，如果软件触发或者外部触发产生，则启动触发注入转换。这种情况下，ADC 取消当前转换，注入通道序列进行扫描模式转换。注入通道组转换结束后，规则通道组转换从上次被取消的转换处重新开始。

图 11-8. 触发注入



11.4.7. 模拟看门狗

ADC_CTL0 寄存器中的 RWDEN 位和 IWDEN 位置 1 时，将分别使能规则组和注入组的模拟看门狗功能。如果 ADC 转换的模拟电压低于低阈值或高于高阈值时，ADC_STAT 状态寄存器的 WDE 位将被置 1。如果 WDEIE 位被置 1，将产生中断。ADC_WDHT 和 ADC_WDLT 寄存器用来设定高低阈值。内部数据的比较在对齐之前完成，因此阀值与 ADC_CTL1 寄存器中 DAL 位确定的对齐方式无关。ADC_CTL0 寄存器的 RWDEN, IWDEN, WDSC 和 WDCHSEL[4:0] 位可以用来选择模拟看门狗监控单一通道或多个通道。

11.4.8. 数据对齐

ADC_CTL1 寄存器的 DAL 位确定转换后数据存储的对齐方式。

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

在左对齐中，12/10/8 位数据按半字方式对齐，而 6 位数据按照字节的方式对齐的，如下 [图 11-9. 12 位分辨率的数据对齐](#), [图 11-10. 10 位分辨率的数据对齐](#), [图 11-11. 8 位分辨率的数据对齐](#) 和 [图 11-12. 6 位分辨率的数据对齐](#) 所示。

图 11-9. 12 位分辨率的数据对齐

规则组数据															
0	0	0	0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

注入组数据															
Sign	Sign	Sign	Sign	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

DAL=0

规则组数据															
D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0

注入组数据															
Sign	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0

DAL=1

图 11-10. 10 位分辨率的数据对齐

规则组数据

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

注入组数据

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

DAL=0

规则组数据

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

注入组数据

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

DAL=1

图 11-11. 8 位分辨率的数据对齐

规则组数据

0	0	0	0	0	0	0	0	D7	D6	D5	D4	D3	D2	D1	D0
---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----

注入组数据

Sign	D7	D6	D5	D4	D3	D2	D1	D0							
------	------	------	------	------	------	------	------	----	----	----	----	----	----	----	----

DAL=0

规则组数据

D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0	0	0
----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---

注入组数据

Sign	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0	0
------	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---

DAL=1

图 11-12. 6 位分辨率的数据对齐

规则组数据

0	0	0	0	0	0	0	0	0	D5	D4	D3	D2	D1	D0
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----

注入组数据

Sign	D5	D4	D3	D2	D1	D0								
------	------	------	------	------	------	------	------	------	----	----	----	----	----	----

DAL=0

规则组数据

0	0	0	0	0	0	0	0	D5	D4	D3	D2	D1	D0	0	0
---	---	---	---	---	---	---	---	----	----	----	----	----	----	---	---

注入组数据

Sign	D5	D4	D3	D2	D1	D0	0								
------	------	------	------	------	------	------	------	------	----	----	----	----	----	----	---

DAL=1

11.4.9. 可编程的采样时间

ADC 使用若干个 ADCCLK 周期对输入电压采样，采样周期数目可以通过 ADC_SAMPT0 和

ADC_SAMPT1 寄存器的 SPTn[2:0]位设置。每个通道可以使用不同的采样周期。例如，在 12 位分辨率的情况下，总转换时间=采样时间+12.5 个 ADCCLK 周期。

例如：

ADCCLK = 28MHz，采样时间为 1.5 个周期，那么总的转换时间为：“1.5+12.5”个 ADCCLK 周期，即 0.500us。

11.4.10. 外部触发

外部触发输入的上升沿可以触发规则组或注入组的转换。规则组的外部触发源由 ADC_CTL1 寄存器的 ETSRC[2:0]位控制，注入组的外部触发源由 ADC_CTL1 寄存器的 ETSIC[2:0]位控制。

ETSRC[2:0]和 ETSIC[2:0]控制位可以用来确定 8 个可能事件中的哪一个可以触发规则和注入组的转换。

表 11-3. 用于 ADC 规则通道的外部触发

ETSRC[2:0]	触发源	触发类型
000	TIMER0_CH0	来自片上定时器的内部信号
001	TIMER0_CH1	
010	TIMER0_CH2	
011	TIMER1_CH1	
100	TIMER2_TRGO	
101	TIMER14_CH0	
110	EXTI_11	外部信号
111	SWRCST	软件触发

表 11-4. 用于 ADC 注入组通道的外部触发

ETSIC[2:0]	触发源	触发类型
000	TIMER0_TRGO	来自片上定时器的内部信号
001	TIMER0_CH3	
010	TIMER1_TRGO	
011	TIMER1_CH0	
100	TIMER2_CH3	
101	TIMER14_TRGO	
110	EXTI_15	外部信号
111	SWICST	软件触发

11.4.11. DMA 请求

因为规则通道转换的值储存在一个仅有的数据寄存器中，所以当转换多个规则通道时需要使用 DMA。DMA 请求，可以通过设置ADC_CTL1寄存器的DMA位来使能。ADC在规则组一个通道转换结束后产生一个DMA请求，DMA接受到请求后可以将转换的数据从ADC_RDATA寄存器传输到用户指定的目的地址。

11.4.12. 温度传感器和内部参考电压 V_{REFINT}

将 ADC_CTL1 寄存器的 TSVREN 位置 1，可以使能温度传感器通道(ADC 通道 16)和 V_{REFINT} 通道(ADC 通道 17)。温度传感器可以用来测量器件周围的温度。传感器输出电压能被 ADC 转换成数字量。建议温度传感器的采样时间为 17.1μs。温度传感器不用时，复位 TSVREN 位可以将其置于掉电模式。

温度传感器的输出电压随温度线性变化，由于生产过程的多样化，温度变化曲线的偏移在不同的芯片上会有不同(最多相差 45°C)。内部温度传感器更适合于检测温度的变化，而不是测量绝对温度。如果需要测量精确的温度，应该使用一个外置的温度传感器来校准这个偏移错误。

内部电压参考(V_{REFINT})提供了一个稳定的（带隙基准）电压输出给 ADC 和比较器。V_{REFINT} 内部连接到 ADC 通道 17 输入通道。

使用温度传感器：

1. 配置温度传感器通道 (ADC_IN16) 的转换序列和采样时间为 17.1us
2. 置位 ADC_CTL1 寄存器中的 TSVREN 位，使能温度传感器
3. 置位 ADC_CTL1 寄存器的 ADCON 位，或者由外部触发启动 ADC 转换
4. 从 ADC 数据寄存器中读取并计算温度传感器数据 V_{temperature}，并由下面公式计算出实际温度：

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

V₂₅: 温度传感器在 25°C 下的电压，典型值为 1.43V。

Avg_Slope: 温度与温度传感器电压曲线的均值斜率，典型值为 4.3mV/°C

11.4.13. 电池电压监测

V_{BAT}通道用于监测从V_{BAT}引脚过来的备份电池电压。当ADC_CTL1寄存器中的VBATEN位置1时，使能V_{BAT}通道 (ADC_IN18)，同时一个集成在V_{BAT}引脚上的2分压桥也随之自动被使能。由于V_{BAT}可能比V_{DDA}高，所以使用这个2分压桥来确保ADC正确操作。它将ADC_IN18输入通道连接到V_{BAT}/2，所以，ADC_IN18输入通道转换的值是V_{BAT}/2。为了防止不必要的电池能量消耗，推荐仅在需要时才使能2分压桥。

11.4.14. ADC 中断

以下任一个事件发生都可以产生中断：

- 规则组和注入组转换结束；
- 模拟看门狗事件（模拟看门狗状态位置1）；

单独的中断使能位用于灵活设置ADC中断。

11.4.15. 可编程分辨率(DRES) - 快速转换模式

通过降低 ADC 的分辨率，可能获得较快的转换时间 (t_{ADC})。

对寄存器 ADC_CTL0 中的 DRES[1:0]位进行编程，可配置 ADC 分辨率为 12、10、8 或 6 位。对于那些不需要高精度数据的应用，可以使用较低的分辨率来实现更快速地转换。

只有在 ADCON 位为 0 时，才能修改 DRES[1:0]的值。

如[表 11-5. 不同分辨率对应的 tCONV 时间](#)所示，较低的分辨率能够减少逐次逼近步骤所需的转换时间 t_{ADC} 。

表 11-5. 不同分辨率对应的 tCONV 时间

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

11.4.16. 片上硬件过采样

片上硬件过采样单元可以通过设置 ADC_OVSAMPCTL 寄存器中的 OVSE 位来使能，它是以降低数据输出率为代价，换取较高的数据分辨率。

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

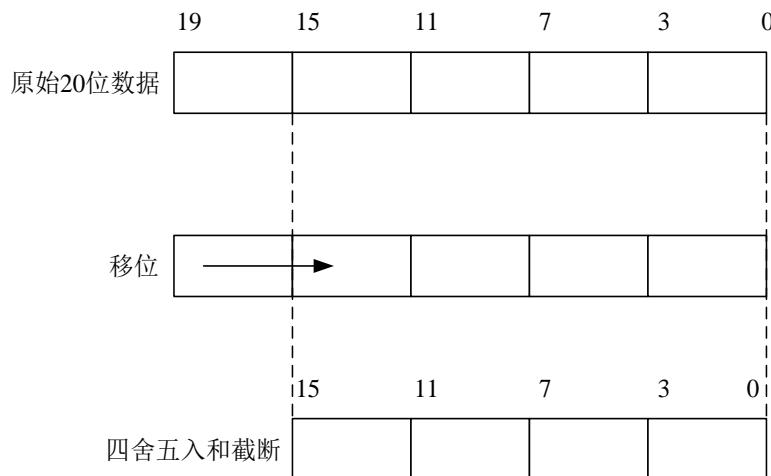
其结果值根据如下公式计算得出，其中，N 和 M 的值可以被调整， $D_{out}(n)$ 是指 ADC 输出的第 n 个数字信号：

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

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

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

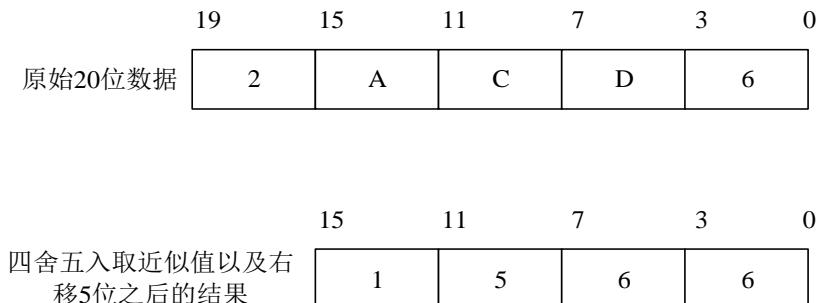
图 11-13. 20 位到 16 位的结果截断



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

[图 11-14. 右移 5 位和取整的数例](#)描述了一个从原始 20 位数据的累积数值处理成 16 位结果值的例子。

图 11-14. 右移 5 位和取整的数例



[表 11-6. N 和 M 的最大输出值 \(灰色部分表示截断\)](#)给出了 N 和 M 的各种组合的数据格式, 初始转换值为 0xFFFF。

表 11-6. N 和 M 的最大输出值 (灰色部分表示截断)

过采样率	最大原始数据	无移位 OVSS= 0000	1 位移位 OVSS= 0001	2 位移位 OVSS= 0010	3 位移位 OVSS= 0011	4 位移位 OVSS= 0100	5 位移位 OVSS= 0101	6 位移位 OVSS= 0110	7 位移位 OVSS= 0111	8 位移位 OVSS= 1000
2x	0x1FFE	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100	0x0080	0x0040	0x0020
4x	0x3FFC	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100	0x0080	0x0040
8x	0x7FF8	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100	0x0080
16x	0xFFFF0	0xFFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100
32x	0x1FFE0	0xFFE0	0xFFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200

过采样率	最大原始数据	无移位 OVSS=0000	1位移位 OVSS=0001	2位移位 OVSS=0010	3位移位 OVSS=0011	4位移位 OVSS=0100	5位移位 OVSS=0101	6位移位 OVSS=0110	7位移位 OVSS=0111	8位移位 OVSS=1000
64x	0x3FFC0	0xFFC0	0xFFE0	0xFFFF	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400
128x	0x7FF80	0xFF80	0xFFC0	0xFFE0	0xFFFF	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800
256x	0xFFFF00	0xFF00	0xFF80	0xFFC0	0xFFE0	0xFFFF	0x7FF8	0x3FFC	0x1FFE	0x0FFF

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

过采样配合 ADC 工作模式

当过采样使能时，大多数 ADC 工作模式都是可用的。

- 规则组通道或注入组通道
- 由软件触发或外部触发开始 ADC 转换
- 单次或扫描模式，连续或间断模式
- 可编程的采样时间
- 模拟看门狗

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

11.5. ADC 寄存器

11.5.1. 状态寄存器 (ADC_STAT)

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问。

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

位/位域	名称	描述
------	----	----

31:5	保留	必须保持复位值
4	STRC	规则组转换开始标志 0: 规则组转换没有开始 1: 规则组转换开始 规则组转换开始时，该位由硬件置位。软件写0清除。
3	STIC	注入组通道转换开始标志 0: 注入通道组转换没有开始 1: 注入通道组转换开始 注入组转换开始时，该位由硬件置位。软件写0清除。
2	EOIC	注入组转换结束标志 0: 注入组转换没有结束 1: 注入组转换结束 所有的注入组通道转换结束时，该位硬件置位。软件写0清除。
1	EOC	组转换结束标志 0: 组转换没有结束 1: 组转换结束 注入组或规则组转换结束时，该位由硬件置位。 软件写0或读ADC_RDATA寄存器清除。
0	WDE	模拟看门狗事件标志 0: 没有模拟看门狗事件 1: 产生模拟看门狗事件 转换电压超过ADC_WDLT和ADC_WDHT寄存器中设定的阈值时，该位由硬件置1，软件写0清除。

11.5.2. 控制寄存器 0 (ADC_CTL0)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				DRES [1:0]	RWDEN	IWDEN	保留								
					rw		rw		rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISNUM [2:0]				DISIC	DISRC	ICA	WDSC	SM	EOICIE	WDEIE	EOCIE	WDCHSEL [4:0]			
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw		

位/位域	名称	描述
31:26	保留	必须保持复位值

25:24	DRES [1:0]	ADC分辨率 00: 12位 01: 10位 10: 8位 11: 6位
23	RWDEN	规则组模拟看门狗使能 0: 规则组模拟看门狗禁止 1: 规则组模拟看门狗使能
22	IWDEN	注入组模拟看门狗使能 0: 注入组模拟看门狗禁止 1: 注入组模拟看门狗使能
21:16	保留	必须保持复位值
15:13	DISNUM [2:0]	间断模式下的转换通道数目 触发后即将被转换的通道数目将变成DISNUM[2:0]+1。
12	DISIC	注入组间断模式 0: 注入组间断模式禁止 1: 注入组间断模式使能
11	DISRC	规则组间断模式 0: 规则组间断模式禁止 1: 规则组间断模式使能
10	ICA	注入组自动转换 0: 注入组自动转换禁止 1: 注入组自动转换使能
9	WDSC	扫描模式下，模拟看门狗在单通道有效 0: 模拟看门狗在所有通道有效 1: 模拟看门狗在单通道有效
8	SM	扫描模式 0: 扫描模式禁止 1: 扫描模式使能
7	EOICIE	EOIC中断使能 0: EOIC中断禁止 1: EOIC中断使能
6	WDEIE	WDE中断使能 0: WDE中断禁止 1: WDE中断使能
5	EOCIE	EOC中断使能 0: EOC中断禁止

1: EOC中断使能

- 4:0 WDCHSEL [4:0] 模拟看门狗通道选择
 00000: ADC 通道0
 00001: ADC 通道1
 00010: ADC 通道2

 01111: ADC 通道15
 10000: ADC 通道16
 10001: ADC 通道17
 10010: ADC 通道18
- 注意:** ADC的模拟输入通道16, 通道17和通道18分别连接到温度传感器, V_{REFINT} 和 V_{BAT}模拟输入。

11.5.3. 控制寄存器 1 (ADC_CTL1)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问。

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

位/位域	名称	描述
31:25	保留	必须保持复位值
24	VBATEN	该位由软件置位和清零, 用于使能和禁止V _{BAT} 通道 0: V _{BAT} 通道禁止 1: V _{BAT} 通道使能
23	TSVREN	ADC的通道16和17使能 0: ADC的通道16和17禁止 1: ADC的通道16和17使能
22	SWRCST	规则组通道软件启动转换 如果ETSIC是111, 该位置‘1’时开启规则组转换。该位由软件置位, 软件清零或转换开始由硬件清零。
21	SWICST	注入组转换开始 如果ETSIC是111, 该位置‘1’时开启规则组转换。该位由软件置位, 软件清零或转换开始由硬件清零。
20	ETERC	规则组通道外部触发使能

		0: 规则组通道外部触发禁止 1: 规则组通道外部触发使能
19:17	ETSRC [2:0]	规则组通道外部触发选择000: TIMER0 CC0 001: TIMER0 CC1 010: TIMER0 CC2 011: TIMER1 CC1 100: TIMER2 TRGO 101: TIMER14 CC1 110: 中断线 11 111: 软件触发SWRCST
16	保留	必须保持复位值
15	ETEIC	注入组通道外部触发使能 0: 注入组通道外部触发禁止 1: 注入组通道外部触发使能
14:12	ETSIC [2:0]	注入组通道外部触发选择000: TIMER0 TRGO 001: TIMER0 CC3 010: TIMER1 TRGO 011: TIMER1 CC0 100: TIMER2 CC3 101: TIMER14 TRGO 110: 中断线15 111: 软件触发SWICST
11	DAL	数据对齐 0: 最低有效位对齐 1: 最高有效位对齐
10:9	保留	必须保持复位值
8	DMA	DMA请求使能 0: DMA请求禁止 1: DMA请求使能
7:4	保留	必须保持复位值
3	RSTCLB	校准复位 该位由软件置位，在校准寄存器初始化后由硬件清零。 0: 校准寄存器初始化结束. 1: 校准寄存器初始化开始
2	CLB	ADC 校准 0: 校准结束 1: 校准开始
1	CTN	连续模式

0: 连续模式禁止

1: 连续模式使能

- 0 ADCON 开启ADC。该位从'0'变成'1'将在稳定时间结束后唤醒ADC。当该位被置位以后，不改变寄存器的其他位仅仅对该位写'1'，将开启转换。
 0: 关闭ADC，关闭电源
 1: 使能ADC

11.5.4. 采样时间寄存器 0 (ADC_SAMPT0)

地址偏移: 0x0C

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				SPT18[2:0]			SPT17[2:0]			SPT16[2:0]			SPT15[2:1]		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPT15[0]		SPT14[2:0]		SPT13[2:0]		SPT12[2:0]		SPT11[2:0]		SPT10[2:0]					
rw		rw		rw		rw		rw		rw		rw			

位/位域	名称	描述
31:27	保留	必须保持复位值
26:24	SPT18[2:0]	参考SPT10[2:0]的描述
23:21	SPT17[2:0]	参考SPT10[2:0]的描述
20:18	SPT16[2:0]	参考SPT10[2:0]的描述
17:15	SPT15[2:0]	参考SPT10[2:0]的描述
14:12	SPT14[2:0]	参考SPT10[2:0]的描述
11:9	SPT13[2:0]	参考SPT10[2:0]的描述
8:6	SPT12[2:0]	参考SPT10[2:0]的描述
5:3	SPT11[2:0]	参考SPT10[2:0]的描述
2:0	SPT10[2:0]	通道采样时间 000: 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

11.5.5. 采样时间寄存器 1 (ADC_SAMPT1)

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留		SPT9[2:0]		SPT8[2:0]		SPT7[2:0]		SPT6[2:0]		SPT5[2:1]						
rw															rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
SPT5[0]	SPT4[2:0]		SPT3[2:0]		SPT2[2:0]		SPT1[2:0]		SPT0[2:0]							
rw															rw	

位/位域	名称	描述
31:30	保留	必须保持复位值
29:27	SPT9[2:0]	参考SPT0[2:0]的描述
26:24	SPT8[2:0]	参考SPT0[2:0]的描述
23:21	SPT7[2:0]	参考SPT0[2:0]的描述
20:18	SPT6[2:0]	参考SPT0[2:0]的描述
17:15	SPT5[2:0]	参考SPT0[2:0]的描述
14:12	SPT4[2:0]	参考SPT0[2:0]的描述
11:9	SPT3[2:0]	参考SPT0[2:0]的描述
8:6	SPT2[2:0]	参考SPT0[2:0]的描述
5:3	SPT1[2:0]	参考SPT0[2:0]的描述
2:0	SPT0[2:0]	通道采样时间 000: 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

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

地址偏移: 0x14-0x20

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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

位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	IOFF [11:0]	注入通道x的数据偏移 当转换注入通道时, 这些位定义了用于从原始转换数据中减去的数值。转换的结果可以在ADC_IDATAx寄存器中读出。

11.5.7. 看门狗高阈值寄存器 (ADC_WDHT)

地址偏移: 0x24

复位值: 0x0000 0FFF

该寄存器只能按字(32位)访问。

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

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

11.5.8. 看门狗低阈值寄存器 (ADC_WDLT)

地址偏移: 0x28

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

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

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

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

11.5.9. 规则序列寄存器 0 (ADC_RSQ0)

地址偏移: 0x2C

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留						RL [3:0]				RSQ15[4:1]					
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSQ15[0]	RSQ14[4:0]				RSQ13[4:0]				RSQ12[4:0]				rw		
rw	rw				rw				rw				rw		

位/位域	名称	描述
31:24	保留	必须保持复位值
23:20	RL [3:0]	规则组通道长度 规则组通道转换序列中的总通道数目为 RL[3:0]+1.
19:15	RSQ15[4:0]	参考RSQ0[4:0]的描述
14:10	RSQ14[4:0]	参考RSQ0[4:0]的描述
9:5	RSQ13[4:0]	参考RSQ0[4:0]的描述
4:0	RSQ12[4:0]	参考RSQ0[4:0]的描述

11.5.10. 规则序列寄存器 1 (ADC_RSQ1)

地址偏移: 0x30

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		RSQ11[4:0]				RSQ10[4:0]				RSQ9[4:1]					
rw								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSQ9[0]		RSQ8[4:0]				RSQ7[4:0]				RSQ6[4:0]					
rw				rw				rw				rw			

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

11.5.11. 规则序列寄存器 2 (ADC_RSQ2)

地址偏移: 0x34

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		RSQ5[4:0]				RSQ4[4:0]				RSQ3[4:1]					
rw								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSQ3[0]		RSQ2[4:0]				RSQ1[4:0]				RSQ0[4:0]					
rw				rw				rw				rw			

位/位域	名称	描述
31:30	保留	必须保持复位值
29:25	RSQ5[4:0]	参考RSQ0[4:0]的描述
24:20	RSQ4[4:0]	参考RSQ0[4:0]的描述
19:15	RSQ3[4:0]	参考RSQ0[4:0]的描述
14:10	RSQ2[4:0]	参考RSQ0[4:0]的描述
9:5	RSQ1[4:0]	参考RSQ0[4:0]的描述

4:0 RSQ0[4:0] 通道编号(0..18)写入这些位来选择规则通道的第n个转换的通道

11.5.12. 注入序列寄存器 (ADC_ISQ)

地址偏移: 0x38

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留										IL[1:0]	ISQ3[4:1]				
rw											rw				rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISQ3[0]	ISQ2[4:0]				ISQ1[4:0]				ISQ0[4:0]						
rw					rw				rw				rw		

位/位域	名称	描述
31:22	保留	必须保持复位值
21:20	IL [1:0]	注入组通道长度 注入通道转换序列中的总通道数目为IL[1:0]+1.
19:15	ISQ3[4:0]	参考ISQ0[4:0]的描述
14:10	ISQ2[4:0]	参考ISQ0[4:0]的描述
9:5	ISQ1[4:0]	参考ISQ0[4:0]的描述
4:0	ISQ0[4:0]	通道编号 (0..18)写入这些位来选择注入组的第n个转换通道。 和规则通道转换序列不同的是, 如果IL[1:0]长度不足4, 注入通道转换从(4-IL[1:0]-1)开始。 IL 注入通道转换顺序 11 ISQ0>>ISQ1>>ISQ2>>ISQ3 10 ISQ1>>ISQ2>>ISQ3 01 ISQ2>>ISQ3 00 ISQ3

11.5.13. 注入数据寄存器 x (ADC_IDATAx) (x= 0..3)

地址偏移: 0x3C - 0x48

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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

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

r

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

11.5.14. 规则数据寄存器 (ADC_RDATA)

地址偏移: 0x4C

复位值: 0x0000_0000

该寄存器只能按字(32位)访问。

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

r

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

r

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	RDATA [15:0]	规则组通道转换数据 这些位包含了规则组通道的转换结果，只读。

11.5.15. 过采样控制寄存器(ADC_OVSAMPCTL)

地址偏移: 0x80

复位值: 0x0000_0000

该寄存器只能按字(32位)访问。

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

r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留 TOVS OVSS [3:0] OVSR [2:0] 保留 OVSEN															

rw

rw

rw

rw

位/位域	名称	描述

31:10	保留	必须保持复位值
9	TOVS	<p>过采样触发 该位通过软件设置和清除。 0: 一个触发后，连续执行通道的所有过采样转换 1: 通道的每次过采样转换都需要一个触发 注意: 当ADCON=0时软件才允许写该位(确定没有转换正在进行)</p>
8:5	OVSS [3:0]	<p>过采样移位 该位通过软件设置和清除。 0000: 不移位 0001: 移1位 0010: 移2位 0011: 移3位 0100: 移4位 0101: 移5位 0110: 移6位 0111: 移7位 1000: 移8位 其余位保留 注意: 只有在ADCON=0的时候，才允许通过软件对该位进行写操作(确保没有转换正在执行)。</p>
4:2	OVSR [2:0]	<p>过采样率 这些位定义了过采样率的大小。000: 2x 001: 4x 010: 8x 011: 16x 100: 32x 101: 64x 110: 128x 111: 256x 注意: 只有在ADCON=0的时候，才允许通过软件对该位进行写操作(确保没有转换正在执行)。</p>
1	保留	必须保持复位值
0	OVSEN	<p>过采样使能 该位通过软件置位和清除。 0: 过采样失能 1: 过采样使能 注意: 只有在ADCON=0的时候，才允许通过软件对该位进行写操作(确保没有转换正在执行)。</p>

12. 数模转换器 (DAC)

12.1. 简介

数字/模拟转换器可以将 12 位的数字数据转换为外部引脚上的电压输出。数据可以采用 8 位或 12 位模式，左对齐或右对齐模式。当使能了外部触发，DMA 可被用于更新数字数据。在输出电压时，可以利用 DAC 输出缓冲区来获得更高的驱动能力。

12.2. 主要特性

DAC 的主要特征如下：

- 12 位分辨率，数据左对齐或右对齐
- DMA 功能并带有下溢错误检测
- 同步更新转换
- 外部事件触发转换
- 可配置的内部缓冲区
- 输入参考电压， V_{DDA}
- 噪声波发生器

[图 12-1. DAC 结构框图](#)为 DAC 的结构框图, [表 12-1. DAC 引脚](#)给出了引脚描述。

图 12-1. DAC 结构框图

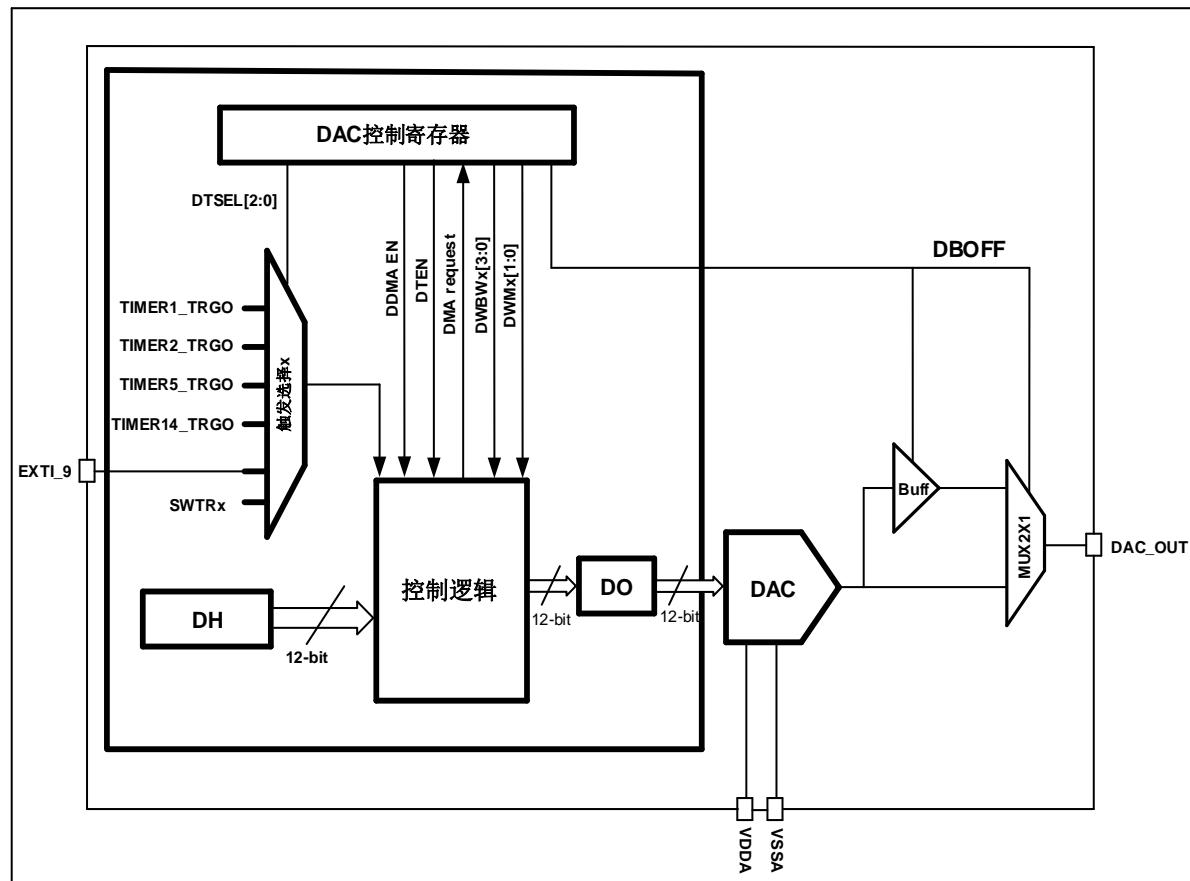


表 12-1. DAC 引脚

名称	描述	信号类型
V _{DDA}	模拟电源	输入, 模拟电源
V _{SSA}	模拟电源地	输入, 模拟电源地
DAC_OUT	DAC 模拟输出	模拟输出信号

在使能 DAC 模块前, GPIO 口 (PA4) 应配置为模拟模式。

12.3. 功能描述

12.3.1. DAC 使能

将 DAC_CTL 寄存器中的 DEN 位置 1 可以给 DAC 上电, 需要 t_{WAKEUP} 时间来启动模拟 DAC 子模块。

12.3.2. DAC 输出缓冲

为了降低输出阻抗，并在没有外部运算放大器的情况下驱动外部负载，DAC 模块内集成了一个输出缓冲区。

这个输出缓冲区默认情况是开启的，可以通过设置 DAC_CTL 寄存器的 DBOFF 位来关闭。

12.3.3. DAC 数据配置

对于 12 位的 DAC 保持数据（DAC_DH），通过对 DAC_R12DH、DAC_L12DH 和 DAC_R8DH 中的任意一个寄存器写入数据来配置。当数据被加载到 DAC_R8DH 寄存器时，只有 8 位最高有效位是可被配置的，4 位最低有效位被强制置为 0。

12.3.4. DAC 触发

通过设置 DAC_CTL 寄存器中 DTEN 位来使能 DAC 外部触发。触发源可以通过 DAC_CTL 寄存器中 DTSEL 位来进行选择。如[表 12-2. DAC 外部触发](#)所示。

表 12-2. DAC 外部触发

DTSELx[2:0]	触发源	触发类型
000	TIMER5_TRGO	内部片上信号
001	TIMER2_TRGO	
010	保留	
011	TIMER14_TRGO	
100	TIMER1_TRGO	
101	保留	
110	EXTI_9	外部信号
111	SWTR	软件触发

TIMERx_TRGO 信号是由定时器生成的，而软件触发是通过设置 DAC_SWT 寄存器的 SWTR 位生成的。

12.3.5. DAC 转换

如果使能了外部触发（通过设置 DAC_CTL 寄存器的 DTEN 位），根据已经选择的触发事件，DAC 保持数据（DAC_DH）会被转移到 DAC 数据输出寄存器（DAC_DO）。否则，在外部触发没有使能的情况下，DAC 保持数据（DAC_DH）会被自动转移到 DAC 数据输出寄存器（DAC_DO）。

当 DAC 保持数据（DAC_DH）加载到 DAC_DO 寄存器时，经过 tSETTLING 时间之后，模拟输出变得有效，tSETTLING 的值与电源电压和模拟输出负载有关。

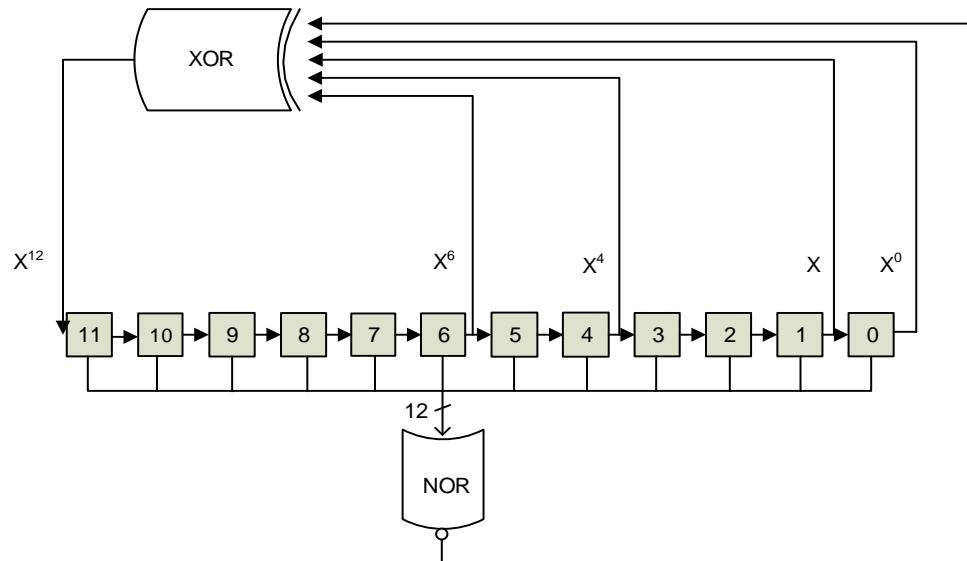
12.3.6. DAC 噪声波

有两种方式可以将噪声波加载到 DAC 输出数据：LFSR 噪声波和三角波。噪声波模式可以通

过 DAC_CTL 寄存器的 DWM 位来进行选择。噪声的幅值可以通过配置 DAC_CTL 寄存器的 DAC 噪声波位宽 (DWBW) 位来进行设置。

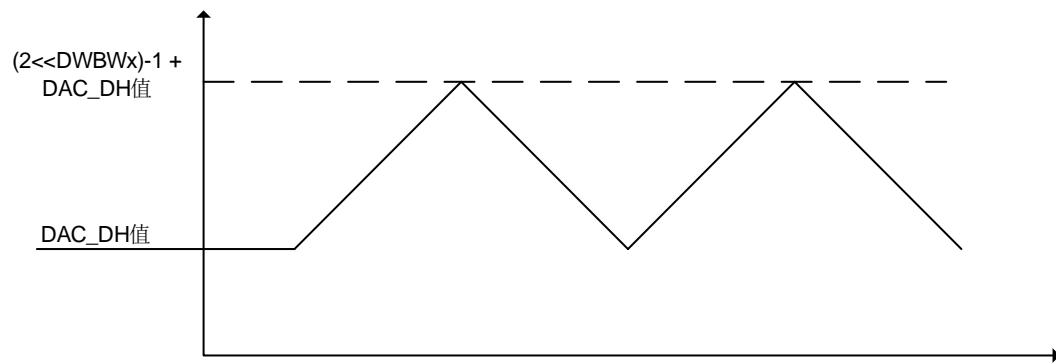
在 DAC 控制逻辑中有一个线性反馈移位寄存器 (LFSR)。在 LFSR 噪声模式下，LFSR 的值与 DAC_DH 值相加后，被写入到 DAC 数据输出寄存器 (DAC_DO)。当配置的 DAC 噪声波位宽小于 12 时，LFSR 的值等于 LFSR 寄存器最低的 DWBW 位，DWBW 位决定了不屏蔽 LFSR 的哪些位。

图 12-2. DAC LFSR 算法



在三角噪声模式下，三角波幅值与 DAC_DH 值相加后，被写入到 DAC 数据输出寄存器 (DAC_DO)。三角波幅值的最小值为 0，最大值为 $(2 \ll \text{DWBW}) - 1$ 。

图 12-3. DAC 三角噪声模式生成的波形



12.3.7. DAC 输出电压

DAC 引脚上的模拟输出电压取决于下面的等式：

$$\text{DAC}_{\text{output}} = V_{\text{DDA}} * \text{DAC}_D / 4096$$

数字输入被线性地转换成模拟输出电压，输出范围为 0 到 V_{DDA} 。

12.3.8. DMA 请求

在外部触发使能的情况下，通过设置 DAC_CTL 寄存器的 DDMAEN 位来使能 DMA 请求。当有外部硬件触发的时候（不是软件触发），则产生一个 DMA 请求。

如果第二个外部触发在上一次请求得到应答之前到达，这个新的请求将不会被处理，并且发生一次下溢错误事件。此时 DAC_STAT 寄存器中的 DDUDR 位被置位，如果 DAC_CTL 寄存器中的 DDUDRIE 位被置位，将产生一次中断。DMA 请求将停滞直到 DDUDR 位被清除。

12.4. DAC 寄存器

12.4.1. 控制寄存器 (DAC_CTL)

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	DDUDRIE	DDMAEN	DWBW [3:0]	DWM [1:0]	DTSEL [2:0]	DTEN	DBOFF	DEN							
	rw	rw	rw	rw	rw	rw	rw	rw							

位/位域	名称	描述
31:14	保留	必须保持复位值
13	DDUDRIE	DAC DMA下溢中断使能 0: DAC DMA下溢中断失能 1: DAC DMA下溢中断使能
12	DDMAEN	DAC DMA使能 0: DAC DMA模式失能 1: DAC DMA模式使能
11:8	DWBW[3:0]	DAC 噪声波位宽 这些位指定了 DAC 的噪声波信号的位宽。LFSR 噪声模式下，这些位表示不屏蔽 LFSR 的位[n-1, 0]；三角噪声模式下，这些位表示三角波幅值为($(2^{n-1}) - 1$)。其中，n 为噪声波位宽。 0000: 波形信号的位宽为 1 0001: 波形信号的位宽为 2 0010: 波形信号的位宽为 3 0011: 波形信号的位宽为 4 0100: 波形信号的位宽为 5 0101: 波形信号的位宽为 6

0110: 波形信号的位宽为 7
 0111: 波形信号的位宽为 8
 1000: 波形信号的位宽为 9
 1001: 波形信号的位宽为 10
 1010: 波形信号的位宽为 11
 ≥1011: 波形信号的位宽为12

7:6	DWM[1:0]	DAC 噪声波模式 这些位指定了在 DAC 外部触发使能(DTEN=1)的情况下，DAC 的噪声波模式的选择。 00: 波形生成失能 01: LFSR 噪声模式 1x: 三角噪声模式
5:3	DTSEL[2:0]	DAC触发选择 这些位用于在DAC外部触发使能(DTEN=1)的情况下，DAC外部触发的选择。 000: TIMER 5 TRGO 001: TIMER 2 TRGO 010: 保留 011: TIMER 14 TRGO 100: TIMER 1 TRGO 101: 保留 110: 外部中断线9 111: 软件触发
2	DTEN	DAC触发使能 0: DAC触发失能 1: DAC触发使能
1	DBOFF	DAC输出缓冲区关闭 0: DAC输出缓冲区打开 1: DAC输出缓冲区关闭
0	DEN	DAC使能 0: DAC失能 1: DAC使能

12.4.2. 软件触发寄存器 (DAC_SWT)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问

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

保留	SWTR
----	------

w

位/位域	名称	描述
31:1	保留	必须保持复位值
0	SWTR	DAC软件触发，由硬件清除 0: 软件触发失能 1: 软件触发使能

12.4.3. DAC 12 位右对齐数据保持寄存器 (DAC_R12DH)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问

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

rw

位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	DAC_DH[11:0]	DAC 12位右对齐数据 这些位指定了数据将由DAC转换。

12.4.4. DAC 12 位左对齐数据保持寄存器 (DAC_L12DH)

地址偏移: 0x0C

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问

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

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:4	DAC_DH[11:0]	DAC 12位左对齐数据 这些位指定了数据将由DAC转换。

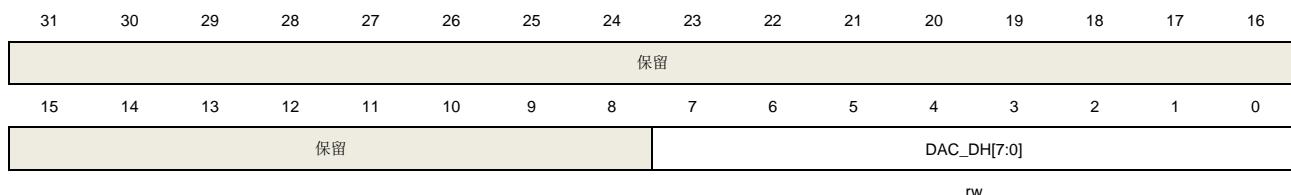
3:0 保留 必须保持复位值

12.4.5. DAC 8 位右对齐数据保持寄存器 (DAC_R8DH)

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问



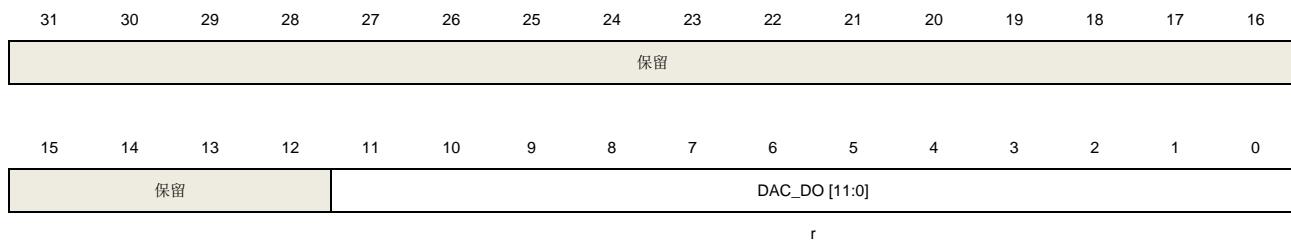
位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	DAC_DH[7:0]	DAC 8位右对齐数据 这些位指定了数据的8位最高有效位将由DAC转换。

12.4.6. DAC 数据输出寄存器 (DAC_DO)

地址偏移: 0x2C

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问



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

12.4.7. 状态寄存器 (DAC_STAT)

地址偏移: 0x34

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

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

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

rc_w1

位/位域	名称	描述
31:14	保留	必须保持复位值
13	DDUDR	<p>DAC DMA 下溢标志</p> <p>该位由硬件置位，写 1 清零。</p> <p>如果当前所选触发的频率在高于 DMA 服务能力的情况下驱动 DAC 转换，则下溢错误发生。</p> <p>0: DMA 下溢错误未发生 1: DMA 下溢错误发生</p>
12:0	保留	必须保持复位值

13. 比较器 (CMP)

13.1. 简介

通用比较器包含 CMP0 和 CMP1，可独立工作，其输出端可用于 I/O 口，也可和定时器结合使用。

比较器可通过模拟信号将 MCU 从低功耗模式中唤醒，在一定的条件下，可将模拟信号作为触发源，结合 DAC 和定时器的 PWM 输出，可以实现电流控制。

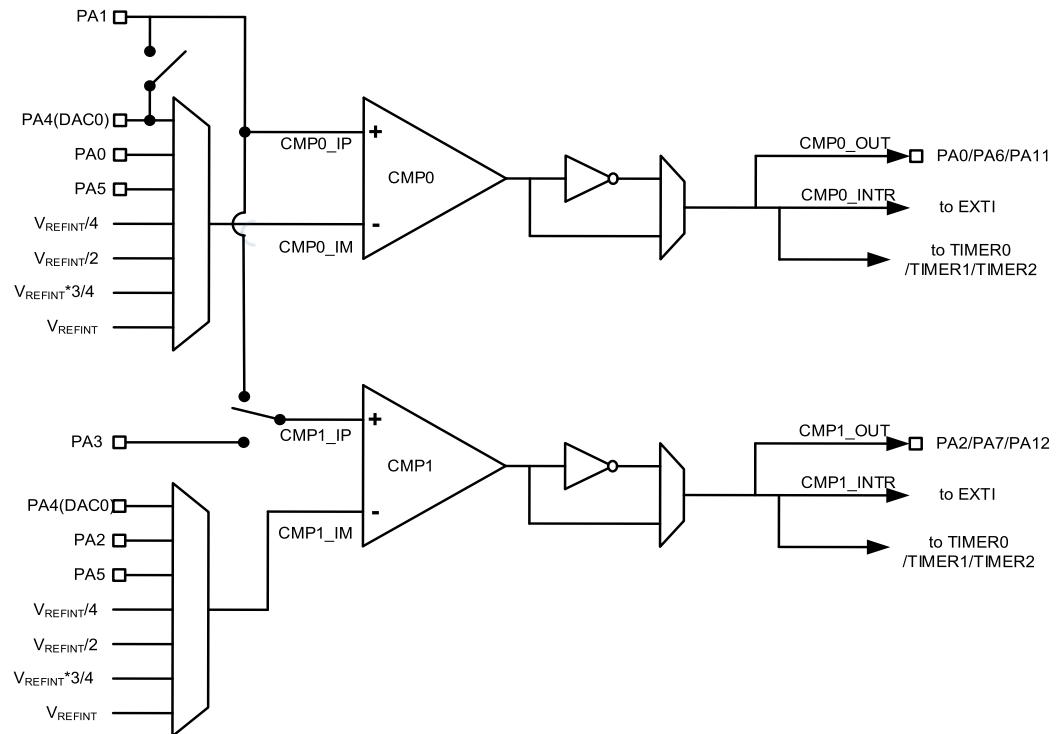
13.2. 主要特性

- 轨对轨比较器
- 迟滞可配置
- 速度、功耗可配置
- 每个比较器可配置以下模拟信号作为输入源
 - DAC 输出
 - 三个 I/O 引脚
 - 0.25、0.5、0.75、1 倍的内部参考电压
- 窗口比较器
- 输出到 I/O 口
- 作为触发源输出到定时器
- 输出到 EXTI

13.3. 功能描述

比较器的框图展示如下：

图 13-1. GD32F3x0 芯片的比较器框图



注意 : V_{REFINT} 是 1.2V.

13.3.1. 比机器时钟和复位

比较器的时钟与 PCLK 同步，它与 SYSCFG 分享共同的复位和时钟使能位。

13.3.2. 比较器的输入和输出

在 I/O 口被选为比较器输入端之前，这些 I/O 口必须在 GPIO 的寄存器中配置为模拟模式。

参考 Datasheet 的引脚定义，比较器输出必须连接到对应的复用 I/O 口。

不同的定时器通过内部连接到比较器的输出端来实现下述功能：

- 输入捕获
- 通过中止功能紧急关掉 PWM 信号
- 通过 OCPRE_CLR 输入逐周期地电流控制

为了在深度睡眠模式下工作，比较器端口的极性选择和输出重定向不会因为 PCLK 关闭。

比较器的输出可实现同步更新内部和外部输入。

比较器的输出端内连到外部中断和事件控制器，每个比较器都有其 EXTI 线，可用于产生中断或事件，同样的机制可运用于从省电模式中退出。

13.3.3. 比较器供电模式

对于给定的程序，在比较器功耗和传输迟滞之间存在着协定，可通过寄存器 **CMP_CS** 的位 **CMPxM** 的配置进行调整。当 **CMPxM** 为 $2'b\ 00$ 时，比较器以最快和功耗最大模式工作，但当 **CMPxM** 位 $2'b\ 11$ 时，比较器以最慢和功耗最小的模式工作。

13.3.4. 比较器迟滞

为了避免噪声信号所引起的假输出，比较器的迟滞可配置，该功能可以在无需要时关闭

13.3.5. 比较器寄存器写保护

比较器的控制状态寄存器（**CMP_CS**）可通过设置 **CMPxLK** 位为 1 来进行保护，**CMP_CS** 寄存器，包含 **CMPxLK** 位，就会变为只读位，只有在 MCU 复位时才可以复位。该写保护功能可运用于某些应用，如热保护和过流保护。

13.4. 比较器寄存器

13.4.1. 控制状态寄存器(**CMP_CS**)

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器通过字访问(32 位)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMP1LK	CMP1O	CMP1HST[1:0]	CMP1PL	CMP1OSEL[2:0]	WNDEN	CMP1MSEL[2:0]	CMP1M	保留	CMP1EN						
rwo	r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP0LK	CMP0O	CMP0HST[1:0]	CMP0PL	CMP0OSEL[2:0]	保留	CMP0MSEL[2:0]	CMP0M[1:0]	CMP0SW	CMP0EN						
rwo	r	rw/r	rw/r	rw/r		rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r

位/位域	名称	描述
31	CMP1LK	CMP1 锁定 该位可将 CMP1 的各控制位设为只读，该位可写一次，通过系统复位清除，可通过软件置位 0: CMP_CS[31:16] 是可读可写位 1: CMP_CS[31:16] 是只读位
30	CMP1O	CMP1 输出 该位即 CMP1 输出状态，是只读位 0: 相输入端低于反相输入端，输出为低电平 1: 相输入端高于反相输入端，输出为高电平

29:28	CMP1HST[1:0]	CMP1 迟滞 该域用于控制迟滞水平 00: 无迟滞 01: 低迟滞 10: 中迟滞 11: 高迟滞
27	CMP1PL	CMP1 输出极性 该位是用来控制输出极性 0: 输出是正相的 1: 输出是反相的
26:24	CMP1OSEL[2:0]	CMP1 输出选择 该域用于控制 CMP1 输出选择 000: 无选择 001: 定时器 0 中止输入 010: 定时器 0 通道 0 输入捕获 011: 定时器 0 OCPRE_CLR 输入 100: 定时器 1 通道 3 输入捕获 101: 定时器 1 OCPRE_CLR 输入 110: 定时器 2 通道 0 输入捕获 111: 定时器 2 OCPRE_CLR 输入
23	WNDEN	窗口模式使能 该位用来断开 CMP1 的 CMP1_IP 输入端与 PA3 的连接，让其与 CMP0 的 CMP0_IP 相连接 0: CMP1_IP 与 PA3 相连 1: CMP1_IP 与 CMP0_IP 相连
22:20	CMP1MSEL[2:0]	CMP1_IM 输入选择 该位用来选择 CMP1 的输入端 CMP1_IM 的输入源 000: V _{REFINT} /4 001: V _{REFINT} /2 010: V _{REFINT} *3/4 011: V _{REFINT} 100: PA4 (DAC) 101: PA5 110: PA2 111: 保留
19:18	CMP1M[1:0]	CMP1 模式 该位用来控制 CMP1 的运行模式以调整速度和功耗 00: 高速/全功耗 01: 中速/中功耗 10: 低速/低功耗 11: 超低速/超低功耗

17	保留	必须为保留值
16	CMP1EN	CMP1 使能 0: CMP1 除能 1: CMP1 使能
15	CMP0LK	CMP0 lock 该位可将 CMP0 的各控制位设为只读，该位可写一次，通过系统复位清除，可通过软件置位。 0: CMP_CS[15:0] 是可读可写位 1: CMP_CS[15:0] 是只读位
14	CMP0O	CMP0 输出 该位即 CMP0 输出状态，是只读位。 0: 相输入端低于反相输入端，输出为低电平 1: 相输入端高于反相输入端，输出为高电平
13:12	CMP0HST[1:0]	CMP0 迟滞 该域用于控制迟滞水平 00: 无迟滞 01: 低迟滞 10: 中迟滞 11: 高迟滞
11	CMP0PL	CMP0 输出极性 该位是用来控制输出极性 0: 输出是正相的 1: 输出是反相的
10:8	CMP0OSEL[2:0]	比较器 0 的输出选择 该域用于控制 CMP0 输出选择 000: 无选择 001: 定时器 0 中止输入 010: 定时器 0 通道 0 输入捕获 011: 定时器 0 OCPRE_CLR 输入 100: 定时器 1 通道 3 输入捕获 101: 定时器 1 OCPRE_CLR 输入 110: 定时器 2 通道 0 输入捕获 111: 定时器 2 OCPRE_CLR 输入
7	保留	必须为保留值
6:4	CMP0MSEL[2:0]	CMP0_IM 输入选择 该位用来选择 CMP0 的输入端 CMP0_IM 的输入源 000: VREFINT/4 001: VREFINT/2 010: VREFINT*3/4 011: VREFINT

100: PA4 (DAC)

101: PA5

110: PA0

111: 保留

3:2 CMP0M[1:0] CMP0 模式

该位用来控制 CMP0 的运行模式以调整速度和功耗

00: 高速/全功耗

01: 中速/中功耗

10: 低速/低功耗

11: 超低速/超低功耗

1 CMP0SW CMP0 开关

该位用于开关 CMP0 相输入端 PA1 与 PA4 之间的连接

0: 开关打开

1: 开关关闭

0 CMP0EN CMP0 使能

0: CMP0 除能

1: CMP0 使能

14. 看门狗定时器 (WDGT)

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

看门狗定时器在内部计数值达到了预设的门限的时候，会触发一个复位(对于窗口看门狗定时器来说，会产生一个中断)。当处理器工作在调试模式的时候看门狗定时器定时计数器可以停止计数。

14.1. 自由看门狗定时器 (FWDGT)

14.1.1. 简介

自由看门狗定时器(FWDGT)有独立的时钟源(IRC40K)。因此就算是主时钟失效了，它仍然能保持工作状态，这非常适合于需要独立环境且对计时精度要求不高的场合。

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

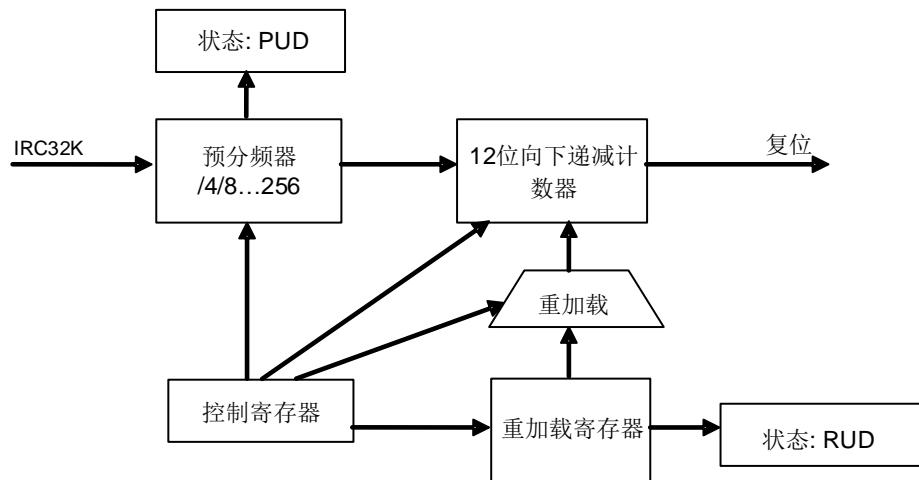
14.1.2. 主要特性

- 自由运行的12位向下计数器；
- 如果看门狗定时器被使能，那么当向下计数器的值达到0时产生系统复位；
- 独立时钟源，自由看门狗定时器在主时钟故障(例如待机和深度睡眠模式下)时仍能工作；
- 自由看门狗定时器硬件控制位，可以用来控制是否在上电时自动启动自由看门狗定时器；
- 可以配置自由看门狗定时器在调试模式下选择停止还是继续工作。

14.1.3. 功能描述

自由看门狗定时器带有一个8级预分频器和一个12位的向下递减计数器。参考[图14-1. 自由看门狗定时器框图](#)的自由看门狗定时器的功能模块。

图 14-1. 自由看门狗定时器框图



向控制寄存器(FWDGT_CTL)中写0xCCCC可以开启自由看门狗定时器，计数器开始向下计数。当计数器记到0x000，产生一次复位。

在任何时候向控制寄存器(FWDGT_CTL)中写0xAAAA都可以重装载计数器，重装载值来源于FWDGT_RLD寄存器。软件可以在计数器计数值达到0x000之前可以通过重装载计数器来阻止看门狗定时器复位。

如果在选项字节中打开了“硬件看门狗定时器”功能，那么在上电的时候看门狗定时器就被自动打开。为了避免复位，软件应该在计数器达到0x000之前重装载计数器。

FWDGT_PSC寄存器和FWDGT_RLD寄存器都有写保护功能。在写数据到这些寄存器之前，需要写0x5555到控制寄存器(FWDGT_CTL)中。写其他任何值到控制寄存器中将会再次启动对这些寄存器的写保护。当预分频寄存器(FWDGT_PSC)或者重装载寄存器(FWDGT_RLD)更新时，FWDGT_STAT寄存器的状态位应该被置1。

如果在MCU调试模块中的FWDGT_HOLD位被清0，即使Cortex™-M4内核停止(调试模式下)自由看门狗定时器依然工作。如果FWDGT_HOLD位被置1，自由看门狗定时器将在调试模式下停止工作。

表 14-1. 自由看门狗定时器在 40kHz (IRC40K)时的最小/最大超时周期

预分频系数	PSC[2:0] 位	最小超时(ms)		最大超时(ms)	
		RLD[11:0]=0x000	RLD[11:0]=0xFFFF	RLD[11:0]=0x000	RLD[11:0]=0xFFFF
1/4	000	0.1		409.6	
1/8	001	0.2		819.2	
1/16	010	0.4		1638.4	
1/32	011	0.8		3276.8	
1/64	100	1.6		6553.6	
1/128	101	3.2		13107.2	
1/256	110 or 111	6.4		26214.4	

通过IRC40K校准可以使自由看门狗定时器超时更加精确。

14.1.4. FWDGT 寄存器

控制寄存器 (FWDGT_CTL)

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

wo

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

预分频寄存器 (FWDGT_PSC)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

rw

位/位域	名称	说明
31:3	保留	必须保持复位值
2:0	PSC[2:0]	自由看门狗定时器计时预分频选择。写这些位之前要通过向FWDGT_CTL寄存器写0x5555去除写保护。在改写这个寄存器的过程中, FWDGT_STAT寄存器的PUD位被置1, 此时读取此寄存器的值都是无效的。 000: 1/4 001: 1/8 010: 1/16

011: 1/32 100: 1/64 101: 1/128

110: 1/256 111: 1/256

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

重装载寄存器 (FWDGT_RLD)

地址偏移: 0x08

复位值: 0x0000 0FFF

该寄存器可以按半字（16位）或字（32位）访问

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

rw

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

状态寄存器 (FWDGT_STAT)

地址偏移: 0x0C

复位值: 0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问

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

ro ro ro

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

窗口寄存器 (FWDGT_WND)

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问

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

位/位域	名称	说明
31:12	保留	必须保持复位值
11:0	WND[11:0]	自由看门狗定时器计数器窗口值。这些位用来将窗口值的上限值与向下递减计数器进行比较。当计数值大于WND[11:0]中值，重装载操作会引起复位。若要改变重装载值，FWDGT_STAT寄存器中的WUD位必须保持复位状态。 这些位有写保护功能。在写这些位之前需向FWDGT_CTL寄存器中写0x5555。 如果应用需要使用几个窗口值，改变窗口值之前必须等到WUD位被清0。除了在进入低功耗模式下，更新了窗口值后，在代码持续执行之前不必等待WUD值被清零。

14.2. 窗口看门狗定时器 (WWDGT)

14.2.1. 简介

窗口看门狗定时器(WWDGT)用来监测由软件故障导致的系统故障。窗口看门狗定时器开启后，向下递减计数器值逐渐减小。计数值达到0x3F时会产生复位(CNT[6]位被清0)。在计数器计数值达到窗口寄存器值之前，计数器的更新也会产生复位。因此软件需要在给定的区间内更新计数器。窗口看门狗定时器在计数器计数值达到0x40或者在计数值达到窗口寄存器之前更新计数器，都会产生一个提前唤醒标志，如果使能中断也将会产生中断。

窗口看门狗定时器时钟是由APB1时钟预分频而来。窗口看门狗定时器适用于需要精确计时的场合。

14.2.2. 主要特性

- 可编程的7位自由运行向下递减计数器；
- 当窗口看门狗使能后，有以下两种情况会产生复位：
 - 当计数器达到0x3F时产生复位；
 - 当计数器的值大于窗口寄存器的值时，更新计数器会产生复位。
- 提前唤醒中断(EWI)：如果看门狗定时器打开，中断使能，计数值达到0x40或者在计数值达到窗口寄存器之前更新计数器的时候会产生中断；
- 可以配置窗口看门狗定时器在调试模式下选择停止还是继续工作。

14.2.3. 功能描述

如果窗口看门狗定时器使能(将WWDGT_CTL寄存器的WDGTEN位置1)，计数值达到0x3F的时候产生复位(CNT[6]位被清0)。或是在计数值达到窗口寄存器值之前，更新计数器也会产生复位。

图 14-2. 窗口看门狗定时器框图



复位之后看门狗定时器总是关闭的。软件可以向WWDGT_CTL的WDGTEN写1开启看门狗定时器。窗口看门狗定时器打开后，计数器始终递减计数，计数器配置的值应该大于0x3F，也就是说CNT[6]位应该被置1。CNT[5:0]决定了两次重装载之间的最大间隔时间。计数器的递减速

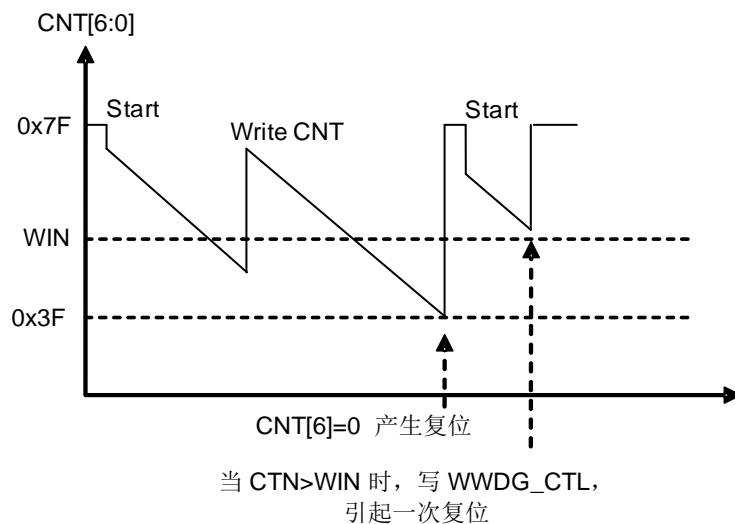
度取决于APB1时钟和预分频器(WWDGT_CFG寄存器的PSC[1:0]位)。

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

对WWDGT_CFG寄存器的EWIE位置1可以使能提前唤醒中断(EWI)，当计数值达到0x40或者在计数值达到窗口寄存器之前更新计数器的时候该中断产生。同时可以用相应的中断服务程序(ISR) 来触发特定的行为(例如通信或数据记录)，来分析软件故障的原因以及在器件复位的时候挽救重要数据。此外，在ISR中软件可以重装载计数器来管理软件系统检查等。在这种情况下，窗口看门狗定时器将永远不会复位但是可以用于其他地方。

通过将WWDGT_STAT寄存器的EWIF位写0可以清除EWI中断。

图 14-3. 窗口看门狗定时器时序图



窗口看门狗定时器超时的计算公式如下：

$$t_{WWDGT} = t_{PCLK1} \times 4096 \times 2^{PSC} \times (CNT[5:0] + 1) \quad (ms) \quad (\text{式14-1})$$

其中：

t_{WWDGT} : 窗口看门狗定时器的超时时间

t_{PCLK1} : APB1以ms为单位的时钟周期

t_{WWDGT} 的最大值和最小值请参考[表 14-2. 在 54MHz \(f_{PCLK1}\) 时的最大/最小超时值](#)。

表 14-2. 在 54MHz (f_{PCLK1}) 时的最大/最小超时值

预分频系数	PSC[1:0]	最小超时	
		CNT[6:0]=0x40	CNT[6:0]=0x7F
1/1	00	75 μs	4.85 ms
1/2	01	151 μs	9.71 ms
1/4	10	303 μs	19.41 ms
1/8	11	606 μs	38.84 ms

如果MCU调试模块中的WWDGT_HOLD位被清0，即使Cortex™-M4内核停止工作(调试模式

(下), 窗口看门狗定时器也可以继续工作。当WWDT_HOLD位被置1时, 窗口看门狗定时器在调试模式下停止。

14.2.4. WWDGT 寄存器

控制寄存器 (WWDGT_CTL)

地址偏移: 0x00

复位值: 0x0000 007F

该寄存器可以按半字(16位)或字(32位)访问

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

配置寄存器 (WWDGT_CFG)

地址偏移: 0x04

复位值: 0x0000 007F

该寄存器可以按半字（16位）或字（32位）访问

位/位域	名称	说明
31:10	保留	必须保持复位值

9	EWIE	提前唤醒中断使能。如果该位被置1，计数值达到0x40时触发中断，或者在计数值达到窗口寄存器之前更新计数器也能触发中断。该位由硬件复位清0，或通过软件时钟复位来清0(参考 4.3.5. APB1 复位寄存器)。写0没有任何作用。
8:7	PSC[1:0]	预分频器，看门狗定时器计数器的时间基准 00: PCLK1/4096/1 01: PCLK1/4096/2 10: PCLK1/4096/4 11: PCLK1/4096/8
6:0	WIN[6:0]	窗口值，当看门狗定时器计数器的值大于窗口值时，写看门狗定时器计数器(WWDGT_CTL的CNT位)会产生复位。

状态寄存器 (WWDGT_STAT)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问

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

rw

位/位域	名称	说明
31:1	保留	必须保持复位值
0	EWIF	提前唤醒中断标志位。当计数值达到0x40或者在计数值达到窗口寄存器之前更新计数器，即使中断没有被使能(WWDGT_CFG中的EWIE位为0)该位也会被硬件置1。这个位可以通过写0清零，写1无效。

15. 实时时钟 (RTC)

15.1. 简介

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

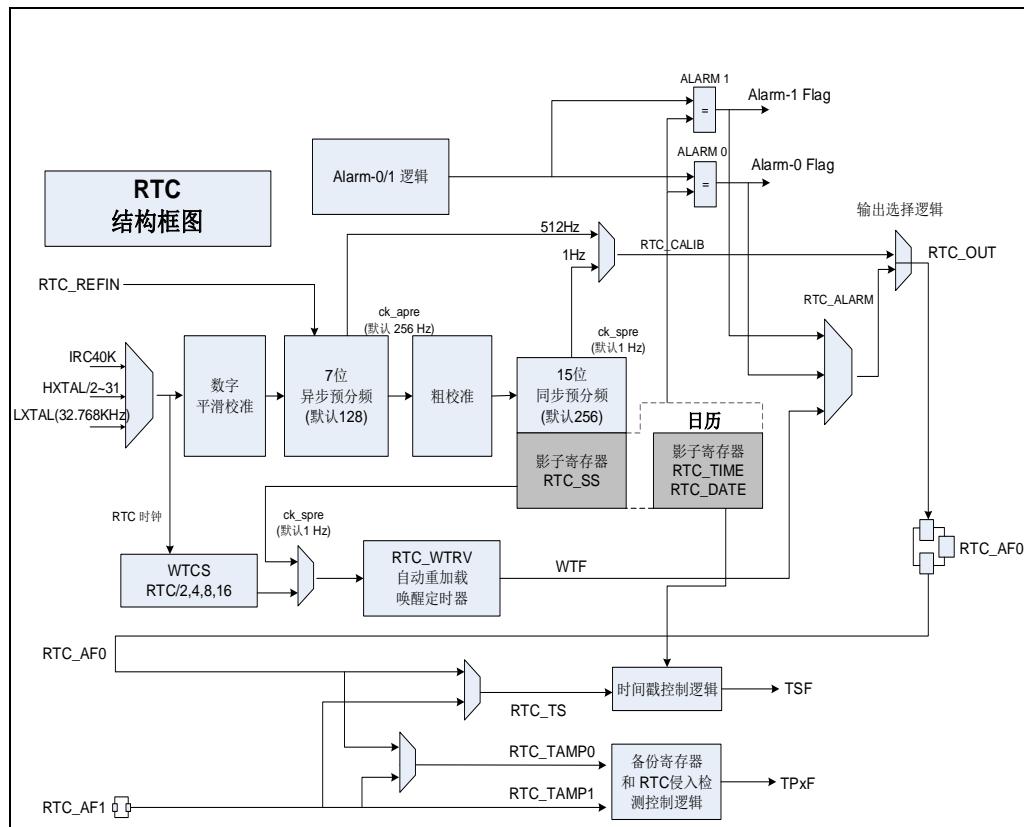
15.2. 主要特性

- 通过软件设置来实现夏令时补偿。
- 参考时钟检测功能：通过外接更高精度的低频率时钟源(50Hz或60Hz)来提高日历精度。
- 数字校准功能：通过调整最小时间单位（最大可调精度0.95ppm）来进行日历校准。
- 通过移位功能进行亚秒级调整。
- 记录事件时间的时间戳功能。
- 两个模式可配置的独立的侵入检测。
- 可编程的日历和一个位域可屏蔽的闹钟。
- 可屏蔽的中断源：
 - 闹钟 0；
 - 时间戳检测；
 - 侵入检测；
- 5个32位 (共20字节) 通用备份寄存器，能够在省电模式下保存数据。当有外部事件侵入时，备份寄存器将会复位。

15.3. 功能描述

15.3.1. 结构框图

图 15-1. RTC 结构框图



RTC 单元包括:

- 闹钟事件/中断。
- 侵入事件/中断。
- 32位备份寄存器。
- 可选的RTC输出功能:
 - 512Hz (默认预分频值): RTC_OUT;
 - 1Hz(默认预分频值) : RTC_OUT;
 - 闹钟事件(极性可配置) : RTC_OUT;
- 可选的RTC输入功能:
 - 时间戳事件检测(RTC_TS);
 - 侵入事件检测 0(RTC_TAMP0);
 - 侵入事件检测 1(RTC_TAMP1);
 - 参考时钟输入 RTC_REFIN(50 或 60Hz)。

15.3.2. 时钟源和预分频

RTC 单元有三个可选的独立时钟源: LXTAL、IRC40K和HXTAL的32分频后的时钟。

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

两个预分频器的频率计算公式如下：

$$f_{\text{ck_apre}} = \frac{f_{\text{rtcclk}}}{\text{FACTOR_A} + 1} \quad (\text{式15-1})$$

$$f_{\text{ck_spre}} = \frac{f_{\text{ck_apre}}}{\text{FACTOR_S} + 1} = \frac{f_{\text{rtcclk}}}{(\text{FACTOR_A} + 1) * (\text{FACTOR_S} + 1)} \quad (\text{式15-2})$$

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

15.3.3. 影子寄存器

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

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

系统复位将复位影子寄存器。

15.3.4. 位域可屏蔽可配置的闹钟

RTC闹钟功能被划分为多个位域并且每一个位域有一个该域的可屏蔽位。

RTC闹钟功能的使能由RTC_CTL寄存器中的ALRMxEN(x=0, 1)位控制。当ALRMxEN=1(x=0, 1) 并且闹钟所有位域的值与对应的日历时间值匹配，ALRMxF (x=0, 1) 标志位将会置位。

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

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

15.3.5. RTC 初始化和配置

RTC 寄存器写保护

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

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

通过下面的步骤，可以解锁这些保护：

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

写一个错误的值到RTC_WPK会使写保护再次生效。写保护状态不受系统复位的影响。被写保护的寄存器如下：

RTC_TIME, RTC_DATE, RTC_CTL, RTC_STAT, RTC_PSC, RTC_ALRM0TD,
RTC_SHIFTCTL, RTC_HRFC, RTC_ALRM0SS。

日历初始化和配置

通过以下步骤可以设置日历和预分频器的值：

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

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

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

YCM标志表明日历是否完成初始化，该标志会硬件检查日历的年份值。

夏令时

通过S1H, A1H和DSM位配置，RTC模块可以支持夏令时补偿调节功能。

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

闹钟功能操作步骤

为了避免意外的闹钟标记置位和亚稳态，闹钟功能的操作应遵循如下流程：

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

15.3.6. 读取日历

当 BPSHAD=0 时，读日历寄存器

当BPSHAD=0，从影子寄存器读日历的值。由于同步机制的存在，正常读取日历需要满足一个基本要求：APB1总线时钟频率必须大于或等于RTC时钟频率的7倍。在任何情况下 APB1总线时钟的频率都不能低于RTC的时钟频率。

当APB1总线时钟频率低于7倍RTC时钟频率时，日历的读取应该遵守以下流程：

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

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

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

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

如果想在一个很短的时间间隔内（少于2个RTCCLK）读取日历，应先清除RSYNF位并等待其置位后再读取。

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

1. 系统复位之后；
2. 日历初始化之后；
3. 一次移位操作之后。

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

当 BPSHAD=1 时，读日历寄存器

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

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

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

值是不准确的。

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

15.3.7. RTC 复位

在RTC单元，有两个复位源可用：系统复位和备份域复位。

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

备份域复位将会影响下面的寄存器，但系统复位不会对它们产生影响：

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

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

15.3.8. RTC 移位功能

当用户有一个高精度的远程时钟而且RTC 1hz 时钟 (ck_spre) 和远程时钟只有一个亚秒级的偏差，RTC单元提供一个称作移位的功能去消除这个偏差来提高秒钟的精确性。

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

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

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

注意：在使用移位功能之前，软件必须检查 RTC_SS 中 SSC 的第 15 位(SSC[15])并确保该位为 0。写 RTC_SHIFTCTL 寄存器之后，RTC_STAT 寄存器的 SOPF 位将会再次置位。当同步移位操作完成时，SOPF 位被硬件清 0。系统复位不影响 SOPF 位。当 REFEN=0 时，移位操作才能正确的工作。如果 REFEN=1，软件禁止写入 RTC_SHIFTCTL。

15.3.9. RTC 参考时钟检测

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

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

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

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

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

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

待机模式下和数字粗校准时，参考时钟检测功能不可用。

15.3.10. RTC 数字平滑校准

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

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

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

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

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

因此使用FREQI可以使RTC频率增加488.5ppm。

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

当数字平滑校准功能正在运行时，按如下公式计算输出校准频率：

$$f_{cal} = f_{rtcclk} \times \left(1 + \frac{FREQI \times 512 - CMSK}{2^N + CMSK - FREQI \times 512}\right) \quad (\text{式15-3})$$

注意： N=20/19 /18 (32/16/8 秒)校准时间周期。

当 FACTOR_A < 3 时校准：

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

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

FACTOR_A = 2: FACTOR_S减少2(8189)

FACTOR_A = 1: FACTOR_S减少4(16379)

FACTOR_A = 0: FACTOR_S减少8(32759)

当FACTOR_A小于3， CMSK为0x100，校准频率公式如下：

$$f_{cal} = f_{rtcclk} \times \left(1 + \frac{256 - CMSK}{2^N + CMSK - 256}\right) \quad (\text{式 15-4})$$

注意： N=20/19 /18 (32/16/8 秒)校准时间周期。

验证 RTC 校准

提供1Hz校准时钟的输出用于协助软件测量并验证RTC的精度。

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

为了消除这一测量误差，测量周期应该和校准周期一致。

■ 校准周期设为32秒(默认配置)

用准确的32秒周期去测量1Hz校准输出的准确性能保证这个测量误差在0.477ppm(在32秒周期内0.5个RTCCLK)之内。

■ 校准周期设为16秒(通过设置CWND16位)

使用此配置，CMSK[0]被硬件置0。

用准确的16秒周期去测量1Hz校准输出的准确性能保证这个测量误差在0.954ppm(在16秒周期内0.5个RTCCLK)之内。

■ 校准周期设为8秒(通过设置CWND8位)

使用此配置，CMSK[1: 0]被硬件置0。

用准确的8秒周期去测量1Hz校准输出的准确性能保证这个测量误差在1.907ppm(在8秒周期内0.5个RTCCLK)之内。

运行中重校准

当INITF位是0，用下面的步骤，软件可以更新RTC_HRFC:

- 1). 等待SCPF位置0;
- 2). 写一个新的值到RTC_HRFC寄存器;
- 3). 3个ck_apre时钟周期之后，新的校准设置开始生效。

15.3.11. 时间戳功能

时间戳功能由RTC_TS管脚输入，通过配置TSEN位来使能。

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

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

RTC模块提供了一个可选的功能特性，来增加时间戳事件的触发源：设置TPTS=1，使得侵入检测功能的侵入事件同时也作为时间戳事件的输入源。

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

15.3.12. 侵入检测

RTC_TAMPx管脚可以作为侵入事件检测功能输入管脚，检测模式有两种可供用户选择：边沿检测模式或者是带可配置滤波功能的电平检测模式。

RTC 备份寄存器((RTC_BKPx)

RTC备份寄存器处于VDD备份域中，即使VDD电源被切断，该区域的寄存器的电源还可由VBAT提供。从待机模式唤醒或系统复位操作都不会影响这些寄存器。

只有当被检测到有侵入事件和备份域复位时，这些寄存器复位。.

初始化侵入检测功能

TPxEN位可以独立使能对应于不同管脚上的RTC侵入检测功能。使能TPxEN位启动侵入检测功能之前，需要设置好侵入检测的配置。当检测到侵入事件，相应的标志位(TPxF)将会置位。如果侵入事件中断使能被启用(TPIE)，侵入事件会产生一个中断。任何侵入事件都会导致备份寄存器(RTC_BKPx)复位。

侵入事件源的时间戳

使能TPTS位，能让侵入检测功能被用作时间戳功能。如果这位被设置为1，当检测到侵入事件时，TSF也将被置位，如同使能了时间戳功能。当检测到侵入事件时，无论TPTS位的值如何，TPxF位将置位。

侵入事件检测为边沿检测模式

当FLT位为0x0时，侵入检测被设置成边沿检测模式，TPxE位决定检测沿是上升沿还是下降沿。当侵入检测配置为边沿检测模式时，侵入检测输入管脚上的上拉电阻将会被禁用。

由于检测侵入事件会复位备份寄存器 (RTC_BKPx)，因此对备份寄存器写操作时应该确保侵入事件导致的复位和写操作不会同时发生。避免这种情形的推荐方法是先关闭侵入检测功能，在完成写操作后再重新启动该功能。

注意：PC13 上的侵入检测功能即使 VDD 电源被关掉也依然可以运行。

侵入事件检测为带可配置滤波功能的电平检测模式

当FLT位没有被设置成0x0时，侵入检测被设置成电平检测模式，FLT位决定有效电平需连续采样的次数(2, 4或者 8)。

当DISPU被设置成0(默认值)，内部的上拉电阻将会在每一次采样前预充电侵入管脚，这样侵入事件的输入管脚上就允许连接更大的电容。预充电的时间可以通过PRCH位来配置。越大的电容，所需的充电时间越长。

电平检测模式下每次采样之间的时间间隔是可配置的。通过调整采样频率(FREQ)，软件能在功耗和检测延迟之间取得一个平衡。

15.3.13. 校准时钟输出

如果COEN位设置为1，RTC_OUT管脚会输出参考校准时钟。

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

当COS位设置为1时，RTC_CALIB的频率计算公式为：

$$f_{rtc_calib} = \frac{f_{rtcclk}}{(FACTOR_A+1) \times (FACTOR_S+1)} \quad (\text{式 15-5})$$

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

15.3.14. 闹钟输出

当OS控制位被设置为0x01时，RTC_ALARM复用输出功能被启用。这个功能将直接输出RTC_STAT寄存器的ALRMxF值。

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

15.3.15. RTC 省电模式管理

表 15-1. 省电模式管理

模式	模式下能否工作	退出该模式的方法
睡眠模式	是	RTC 中断
深度睡眠模式	当时钟源是LXTAL或IRC40K时可以工作	RTC闹钟/侵入事件/时间戳事件
待机模式	当时钟源是LXTAL或IRC40K时可以工作	RTC闹钟/侵入事件/时间戳事件

15.3.16. RTC 中断

所有的RTC中断都被连接到EXTI控制器。

如果想使用RTC闹钟/侵入事件/时间戳中断，应按下面步骤操作：

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

表 15-2. 中断控制

中断	事件标志	控制位	退出睡眠模式	退出深度睡眠模式	退出待机模式
闹钟 0	ALRMOF	ALRMOIE	Y	Y(*)	Y(*)
时间戳	TSF	TSIE	Y	Y(*)	Y(*)
侵入 0	TP0F	TPIE	Y	Y(*)	Y(*)
侵入 1	TP1F	TPIE	Y	Y(*)	Y(*)

*：仅当RTC时钟源是LXTAL或IRC40K时有效。

15.4. RTC 寄存器

15.4.1. 时间寄存器 (RTC_TIME)

偏移地址: 0x00

系统复位值: 当BPSHAD = 0, 0x0000 0000

当BPSHAD = 1, 无影响

写保护寄存器，仅在初始化状态可以进行写操作

该寄存器只能按字(32位)访问

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

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

15.4.2. 日期寄存器 (RTC_DATE)

偏移地址: 0x04

系统复位值: 当 BPSHAD = 0, 0x0000 2101

当 BPSHAD = 1, 无影响

写保护寄存器，仅在初始化状态可以进行写操作

该寄存器只能按字(32位)访问

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

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

15.4.3. 控制寄存器 (RTC_CTL)

偏移地址: 0x08

系统复位: 无影响

备份域复位值: 0x0000 0000

写保护寄存器

该寄存器只能按字(32位)访问

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

位/位域	名称	描述
31:24	保留	必须保持复位值
23	COEN	校准输出使能 0: 关闭校准输出 1: 使能校准输出
22:21	OS[1:0]	输出选择 该位用来选择输出的标志源。 0x0: 禁用 RTC_ALARM 输出

		0x1: 启用闹钟 0 标志输出
20	OPOL	输出极性 该位用来反转 RTC_ALARM 输出。 0: 禁用反转 RTC_ALARM 输出 1: 启用反转 RTC_ALARM 输出
19	COS	校准输出选择 仅当 COEN=1 并且预分频器是默认值时有效。 0: 校准输出是 512Hz 1: 校准输出是 1Hz
18	DSM	夏令时屏蔽位 该位可以通过软件灵活使用。常用来记录夏令时调整。
17	S1H	减 1 小时(冬季时间变化) 当前时间非零的情况下，将当前时间减去一个小时。 0: 没有影响 1: 在下一个秒改变时，将减少一个小时
16	A1H	增加 1 小时(夏季时间变化) 将当前时间增加一个小时。 0: 没有影响 1: 在下一个秒改变时，将增加一个小时
15	TSIE	时间戳中断使能 0: 禁用时间戳中断 1: 启用时间戳中断
14:13	保留	必须保持复位值
13	ALRM1IE	RTC 闹钟 1 中断使能 0: 禁用闹钟中断 1: 启用闹钟中断
12	ALRM0IE	RTC 闹钟 0 中断使能 0: 禁用闹钟中断 1: 启用闹钟中断
11	TSEN	时间戳功能使能 0: 禁用时间戳功能 1: 启用时间戳功能
10:9	保留	必须保持复位值
8	ALRM0EN	闹钟 0 功能使能 0: 禁用闹钟功能 1: 启用闹钟功能
7	保留	必须保持复位值

6	CS	时间格式 0: 24 小时制 1: 12 小时制 注意: 仅能在初始化状态进行写入
5	BPSHAD	禁止影子寄存器 0: 读取的日历的值来自影子日历寄存器 1: 读取的日历的值来自真正日历寄存器 注意: 如果 APB1 时钟的频率小于 RTCCLK 频率的 7 倍, 该位必须设为 1
4	REFEN	参考时钟检测功能使能 0: 禁用参考时钟检测功能 1: 启用参考时钟检测功能 注意: 仅能在初始化状态进行写入并且 FACTOR_S 必须为 0x00FF
3	TSEG	时间戳事件有效检测边沿 0: 上升沿是时间戳事件有效检测沿 1: 下降沿是时间戳事件有效检测沿
2:0	保留	必须保持复位值

15.4.4. 状态寄存器 (RTC_STAT)

偏移地址: 0x0C

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

备份域复位值: 0x0000 0007

写保护寄存器, 除 RTC_STAT[14: 8]外

该寄存器只能按字(32 位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留														SCPF	
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TP1F	TP0F	TSOVRF	TSF	保留	ALRMOF	INITM	INITF	RSYNF	YCM	SOPF	保留	ALRMOWF		
rc_w0	rc_w0	rc_w0	rc_w0	rc_w0		rc_w0	rw	r	rc_w0	r	r			r	

位/位域	名称	描述
31:17	保留	必须保持复位值
16	SCPF	平滑校准挂起标志 在未进入初始化模式时向 RTC_HRFC 进行软件写操作, 该位被硬件置 1。当平滑校准设置开始执行后, 该位被硬件清零 0。
15	保留	必须保持复位值
14	TP1F	RTC_TAMP1 事件标志 当在 tamper1 输入管脚检测到侵入事件时, 该位硬件置 1。可以通过向该位软件写 0 来清

		除。
13	TP0F	RTC_TAMPO 事件标志 当在 tamper0 输入管脚检测到侵入事件时，该位硬件置 1。可以通过向该位软件写 0 来清除。
12	TSOVRF	时间戳事件溢出标志 如果 TSF 位已经置位，当再次检测到时间戳事件时，该位会通过硬件置 1。 可以通过向该位软件写 0 来清除。
11	TSF	时间戳事件标志 当检测到一个时间戳事件时，该位会通过硬件置 1。可以通过向该位软件写 0 来清除。
10:9	保留	必须保持复位值
9	ALRM1F	Alarm1 发生标志 当现在的时间/日期与闹钟 1 设置的时间/日期匹配的时候，该位会通过硬件置 1。可以通过向该位软件写 0 来清除。
8	ALRM0F	Alarm0 发生标志 当现在的时间/日期与闹钟 0 设置的时间/日期匹配的时候，该位会通过硬件置 1。可以通过向该位软件写 0 来清除。
7	INITM	进入初始化模式 0：自由运行模式 1：进入初始化模式设置时间/日期和预分频，计数器将停止运行
6	INITF	初始化状态标志 该位被硬件置 1， 初始化状态时可以设置日历寄存器和预分频器。 0：日历寄存器和预分频器的值不能改变 1：日历寄存器和预分频器的值可以改变
5	RSYNF	寄存器同步标志 每 2 个 RTCCLK 将会由硬件置 1 一次，同时会复制当前日历时间/日期到影子日历寄存器。初始化模式(INITM)，移位操作挂起标志(SOPF)或者禁止影子寄存器模式(BPSHAD = 1)会清除该位。该位也可以通过软件写 0 清除。 0：影子寄存器未同步 1：影子寄存器已同步
4	YCM	年份配置标志 当日历寄存器的年份值不为 0 时硬件置 1 0：日历尚未初始化 1：日历已经初始化
3	SOPF	移位功能操作挂起标志 0：移位操作没有挂起 1：移位操作挂起
2:1	保留	必须保持复位值

1	ALRM1WF	Alarm1 配置可写标志 硬件置位和清零。ALRM1EN=0 时，标记 alarm 是否可写。 0: 不允许修改 Alarm 寄存器设置 1: 允许修改 Alarm 寄存器设置
0	ALRM0WF	Alarm0 配置可写标志 硬件置位和清零。ALRM0EN=0 时，标记 alarm 是否可写。 0: 不允许修改 Alarm 寄存器设置 1: 允许修改 Alarm 寄存器设置

15.4.5. 预分频寄存器 (RTC_PSC)

偏移地址: 0x10

系统复位: 无影响

备份域复位值: 0x007F 00FF

写保护寄存器，仅在初始化状态可以进行写操作

该寄存器只能按字(32位)访问

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

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

15.4.6. 闹钟 0 时间日期寄存器 (RTC_ALRM0TD)

偏移地址: 0x1C

系统复位: 无影响

备份域复位值: 0x0000 0000

写保护寄存器，仅在初始化状态可以进行写操作

该寄存器只能按字(32位)访问

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

rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSKM	MNT[2:0]		MNU[3:0]		MSKS	SCT[2:0]		SCU[3:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域

名称

描述

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

15.4.7. 写保护钥匙寄存器 (RTC_WPK)

偏移地址: 0x24

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	WPK[7:0]	写保护的解锁值

15.4.8. 亚秒寄存器(RTC_SS)

偏移地址: 0x28

系统复位值：当BPSHAD = 0, 0x0000 0000。

当BPSHAD = 1, 无影响。

该寄存器只能按字(32位)访问

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

15.4.9. 移位控制寄存器 (RTC_SHIFTCTL)

偏移地址： 0x2C

系统复位：无影响

备份域复位值: 0x0000 0000

写保护寄存器，仅当SOPF=0，该寄存器可写

该寄存器只能按字(32位)访问

A1S	保留
W	
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
保留	SFS[14:0]

W

位/位域	名称	描述
31	A1S	增加一秒 0: 无影响 1: 增加一秒到时钟/日历 该位与 SFS 位一起使用，增加小于一秒到当前时间。
30:15	保留	必须保持复位值
14:0	SFS[14:0]	减去小于一秒的一段时间 这位的值将增加到同步预分频计数器 当仅用 SFS 时，由于同步预分频器是一个递减计数器，所以时钟将会延迟。 $\text{延迟(秒)} = \text{SFS} / (\text{FACTOR_S} + 1)$ 当 A1S 和 SFS 一起使用时，时钟将会提前 $\text{提前(秒)} = (1 - (\text{SFS} / (\text{FACTOR_S} + 1)))$

注意：写入此寄存器会导致 RSYNF 位被清 0。

15.4.10. 时间戳时间寄存器 (RTC_TTS)

偏移地址: 0x30

备份域复位值: 0x0000 0000

系统复位: 无影响

当 TSF 被置 1，该位用来记录日历时间

清除 TSF 位也会清除此寄存器

该寄存器只能按字(32 位)访问

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

位/位域	名称	描述
31:23	保留	必须保持复位值
22	PM	AM/PM 标记 0: AM 或 24 小时制 1: PM

21:20	HRT[1:0]	小时十位值, 以 BCD 码形式存储
19:16	HRU[3:0]	小时个位值, 以 BCD 码形式存储
15	保留	必须保持复位值
14:12	MNT[2:0]	分钟十位值, 以 BCD 码形式存储
11:8	MNU[3:0]	分钟个位值, 以 BCD 码形式存储
7	保留	必须保持复位值
6:4	SCT[2:0]	秒钟十位值, 以 BCD 码形式存储
3:0	SCU[3:0]	秒钟个位值, 以 BCD 码形式存储

15.4.11. 时间戳日期寄存器 (RTC_DTS)

偏移地址: 0x34

备份域复位值: 0x0000 0000

系统复位: 无影响

当TSF被置1, 该位用来记录日历日期

清除TSF位也会清除此寄存器

该寄存器只能按字(32位)访问

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

r r r r r r

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

15.4.12. 时间戳亚秒寄存器 (RTC_SSTS)

偏移地址: 0x38

备份域复位: 0x0000 0000

系统复位: 无影响

当TSF被置1, 该位用来记录日历时间

清除TSF位也会清除此寄存器

该寄存器只能按字(32位)访问

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

r

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	SSC[15:0]	亚秒值 TSF 置 1 时记录当时的同步预分频计数器的值。

15.4.13. 高精度频率补偿寄存器 (RTC_HRFC)

偏移地址: 0x3C

备份域复位: 0x0000 0000

系统复位: 无影响

写保护寄存器

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FREQI	CWND8	CWND16	保留				CMSK[8:0]								rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15	FREQI	RTC 频率增加 488.5ppm 0: 无影响 1: 每 2^{11} 个脉冲增加一个 RTCCLK 脉冲 该位需与 CMSK 位一起使用。如果输入时钟频率是 32.768KHz, 在 32s 校准窗期间, 增加的 RTCCLK 脉冲数是 $(512 * FREQI) - CMSK$
14	CWND8	采用 8 秒校准周期 0: 无影响

1: 采用 8 秒校准周期
注意: 当 CWND8=1, CMSK[1:0] 被锁定在“00”。

13	CWND16	采用 16 秒校准周期 0: 无影响 1: 采用 16 秒校准周期 注意: 当 CWND16=1, CMSK[0] 被锁定在“0”。
12:9	保留	必须保持复位值
8:0	CMSK[8:0]	校准周期 RTCCLK 脉冲屏蔽数 在 2^{20} RTCCLK 脉冲之内屏蔽的脉冲数 此项功能可以以 0.9537 ppm 的分辨率来降低日历频率

15.4.14. 侵入寄存器 (RTC_TAMP)

偏移地址: 0x40
备份域复位: 0x0000 0000
系统复位: 无影响

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								PC15MDE	PC15VAL	PC14MDE	PC14VAL	PC13MDE	PC13VAL	保留	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISPU	PRCH[1:0]	FLT[1:0]	FREQ[2:0]	TPTS	保留		TP1EG	TP1EN	TPIE	TP0EG	TP0EN	rw	rw	rw	rw
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:24	保留	必须保持复位值
23	PC15MDE	PC15 模式 0: 无影响 1: 如果 LXTAL 禁用, 强制 PC15 推挽输出
22	PC15VAL	PC15 值 当 LXTAL 禁用且 PC15MDE=1 时, PC15 输出该位数据
21	PC14MDE	PC14 模式 0: 无影响 1: 如果 LXTAL 禁用, 强制 PC14 推挽输出
20	PC14VAL	PC14 值 当 LXTAL 禁用且 PC14MDE=1 时, PC14 输出该位数据
19	PC13MDE	PC13 模式 0: 无影响

		1: 当 RTC 所有备用功能禁用时, PC13 输出该位数据
18	PC13VAL	<p>PC13 值或闹钟输出类型值</p> <p>PC13 输出闹钟</p> <p>0: PC13 开漏输出</p> <p>1: PC13 推挽输出</p> <p>当 RTC 所有备用功能禁用且 PC13MDE=1 时</p> <p>0: PC13 输出 0</p> <p>1: PC13 输出 1</p>
17:16	保留	必须保持复位值
15	DISPU	<p>RTC_TAMPx 上拉禁用位</p> <p>0: 使能内部 RTC_TAMPx 引脚上的上拉电阻并在采样前进行预充电</p> <p>1: 禁用预充电功能</p>
14:13	PRCH[1:0]	<p>RTC_TAMPx 的预充电时间</p> <p>该位设置决定了每次采样前的预充电时间</p> <p>0x0: 1 个 RTC 时钟</p> <p>0x1: 2 个 RTC 时钟</p> <p>0x2: 4 个 RTC 时钟</p> <p>0x3: 8 个 RTC 时钟</p>
12:11	FLT[1:0]	<p>RTC_TAMPx 过滤器计数设置</p> <p>该位决定了侵入事件检测模式和在电平检测模式下连续采样的次数。</p> <p>0x0: 用边沿模式检测侵入事件, 预充电功能被自动禁用。</p> <p>0x1: 用电平模式检测侵入事件。连续采样到 2 个有效电平时认为发生侵入事件</p> <p>0x2: 用电平模式检测侵入事件。连续采样到 4 个有效电平时认为发生侵入事件</p> <p>0x3: 用电平模式检测侵入事件。连续采样到 8 个有效电平时认为发生侵入事件</p>
10:8	FREQ[2:0]	<p>侵入事件电平模式检测的采样频率</p> <p>0x0: 每次采样间隔 32768 个 RTCCLK(若 RTCCLK=32.768KHz, 频率为 1Hz)</p> <p>0x1: 每次采样间隔 16384 个 RTCCLK(若 RTCCLK=32.768KHz, 频率为 2Hz)</p> <p>0x2: 每次采样间隔 8192 个 RTCCLK(若 RTCCLK=32.768KHz, 频率为 4Hz)</p> <p>0x3: 每次采样间隔 4096 个 RTCCLK(若 RTCCLK=32.768KHz, 频率为 8Hz)</p> <p>0x4: 每次采样间隔 2048 个 RTCCLK(若 RTCCLK=32.768KHz, 频率为 16Hz)</p> <p>0x5: 每次采样间隔 1024 个 RTCCLK(若 RTCCLK=32.768KHz, 频率为 32Hz)</p> <p>0x6: 每次采样间隔 512 个 RTCCLK(若 RTCCLK=32.768KHz, 频率为 64Hz)</p> <p>0x7: 每次采样间隔 256 个 RTCCLK(若 RTCCLK=32.768KHz, 频率为 128Hz)</p>
7	TPTS	<p>侵入事件时触发时间戳</p> <p>0: 无影响</p> <p>1: 当检测到侵入事件时, 即使 TSEN=0, TSF 也会被置位</p>
6:5	保留	必须保持复位值
4	TP1EG	<p>TAMP1 输入管脚的侵入事件检测触发沿</p> <p>如果侵入检测处于边沿模式(FLT =0):</p>

		0: 上升沿触发一个侵入检测事件 1: 下降沿触发一个侵入检测事件 如果侵入检测处于电平模式(FLT !=0): 0: 低电平触发一个侵入检测事件 1: 高电平触发一个侵入检测事件
3	TP1EN	Tamper1 检测使能位 0: 禁用 Tamper1 检测功能 1: 启用 Tamper1 检测功能
2	TPIE	侵入检测中断使能 0: 禁用侵入中断 1: 启用侵入中断
1	TP0EG	TAMP0 输入管脚的侵入事件检测触发沿 如果侵入检测处于边沿模式(FLT =0): 0: 上升沿触发一个侵入检测事件 1: 下降沿触发一个侵入检测事件 如果侵入检测处于电平模式(FLT !=0): 0: 低电平触发一个侵入检测事件 1: 高电平触发一个侵入检测事件
0	TP0EN	Tamper0 检测使能位 0: 禁用 Tamper0 检测功能 1: 启用 Tamper0 检测功能

注意: 强烈建议在改变侵入检测配置之前, 应该复位 TPxEN 位。

15.4.15. 闹钟 0 亚秒寄存器 (RTC_ALRM0SS)

偏移地址: 0x44

备份域复位: 0x0000 0000

系统复位: 无影响

写保护寄存器, 仅当ALRM0EN=0或INITM=1, 可以进行写操作

该寄存器只能按字(32 位)访问

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

位/位域	名称	描述
31:28	保留	必须保持复位值
27:24	MSKSSC[3:0]	亚秒位域的屏蔽控制位

0x0: 屏蔽闹钟亚秒设置。当所有其他的闹钟位域匹配的时候，闹钟将会在每一秒钟到达的时刻置 1。

0x1: SSC[0]位用于时间匹配，其他位被忽略。

0x2: SSC[1: 0] 位用于时间匹配，其他位被忽略。

0x3: SSC[2: 0] 位用于时间匹配，其他位被忽略。

0x4: SSC[3: 0] 位用于时间匹配，其他位被忽略。

0x5: SSC[4: 0] 位用于时间匹配，其他位被忽略。

0x6: SSC[5: 0] 位用于时间匹配，其他位被忽略。

0x7: SSC[6: 0] 位用于时间匹配，其他位被忽略。

0x8: SSC[7: 0] 位用于时间匹配，其他位被忽略。

0x9: SSC[8: 0] 位用于时间匹配，其他位被忽略。

0x10: SSC[9: 0] 位用于时间匹配，其他位被忽略。

0x11: SSC[10: 0] 位用于时间匹配，其他位被忽略。

0x12: SSC[11: 0] 位用于时间匹配，其他位被忽略。

0x13: SSC[12: 0] 位用于时间匹配，其他位被忽略。

0x14: SSC[13: 0] 位用于时间匹配，其他位被忽略。

0x15: SSC[14: 0] 位用于时间匹配，其他位被忽略。

注意：同步预分频计数器的第 15 位(RTC_SS 寄存器中的 SSC[15])从不被匹配。

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

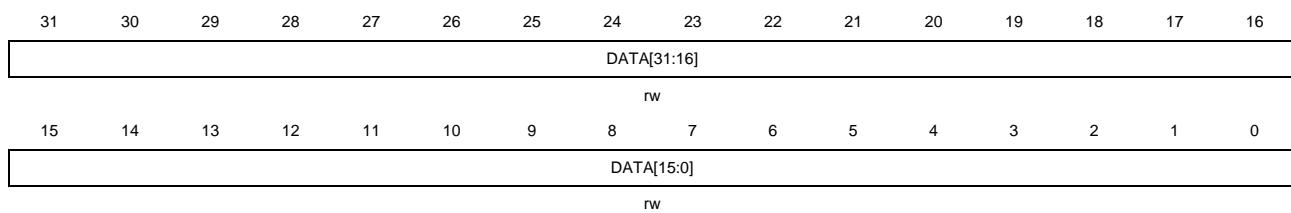
15.4.16. 备份寄存器 (RTC_BKPx) (x=0..4)

偏移地址: 0x50 到 0x60

备份域复位: 0x0000 0000

系统复位: 无影响

该寄存器只能按字(32位)访问



位/位域	名称	描述
31:0	DATA[31:0]	数据 软件可读写寄存器。由于此寄存器可由 V _{BAT} 供电，因此寄存器值在省电模式下依然保持有效。当侵入检测标志位 TPxF 置 1，这些寄存器会被复位。当 FMC 读保护功能禁用时，这些寄存器会被复位。

16. 定时器 (TIMER)

表 16-1. 定时器 (TIMERx) 分为六种类型

定时器	定时器 0	定时器 1/2	定时器 13	定时器 14	定时器 15/16	定时器 5
类型	高级	通用 L0	通用 L2	通用 L3	通用 L4	基本
预分频器	16 位	16 位	16 位	16 位	16 位	16 位
计数器	16 位	32 位(定时器 1) 16 位(定时器 2)	16 位	16 位	16 位	16 位
计数模式	向上, 向下, 中央对齐	向上, 向下, 中央对齐	只有向上	只有向上	只有向上	只有向上
可重复性	●	×	×	●	●	×
捕获/比较通道数	4	4	1	2	1	0
互补和死区时间	●	×	×	●	●	×
中止输入	●	×	×	●	●	×
单脉冲	●	●	×	●	●	●
正交译码器	●	●	×	×	×	×
从设备控制器	●	●	×	●	×	×
内部连接	● ⁽¹⁾	● ⁽²⁾	×	● ⁽³⁾	×	TRGO TO DAC
DMA	●	●	×	●	●	● ⁽⁴⁾
Debug 模式	●	●	●	●	●	●

(1) TIMER0 **ITI0: TIMER14_TRGO** **ITI1: TIMER1_TRGO** **ITI2: TIMER2_TRGO** **ITI3: 0**

(2) TIMER1 **ITI0: TIMER0_TRGO** **ITI1: TIMER14_TRGO** **ITI2: TIMER2_TRGO** **ITI3: 0**
 TIMER2 **ITI0: TIMER0_TRGO** **ITI1: TIMER1_TRGO** **ITI2: TIMER14_TRGO** **ITI3: 0**

(3) TIMER14 **ITI0: TIMER1_TRGO** **ITI1: TIMER2_TRGO** **ITI2: 0** **ITI3: 0**

(4) 只有更新事件可以产生 DMA 请求。但是定时器 5 中没有 DMA 配置寄存器。

16.1. 高级定时器 (TIMERx,x=0)

16.1.1. 简介

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

高级定时器是可编程的，可以被用来计数，其外部事件可以驱动其他定时器

高级定时器包含了一个死区时间插入模块，非常适合电机控制。

定时器和定时器之间是相互独立，但是他们可以被同步在一起形成一个更大的定时器，这些定时器的计数器一致地增加。

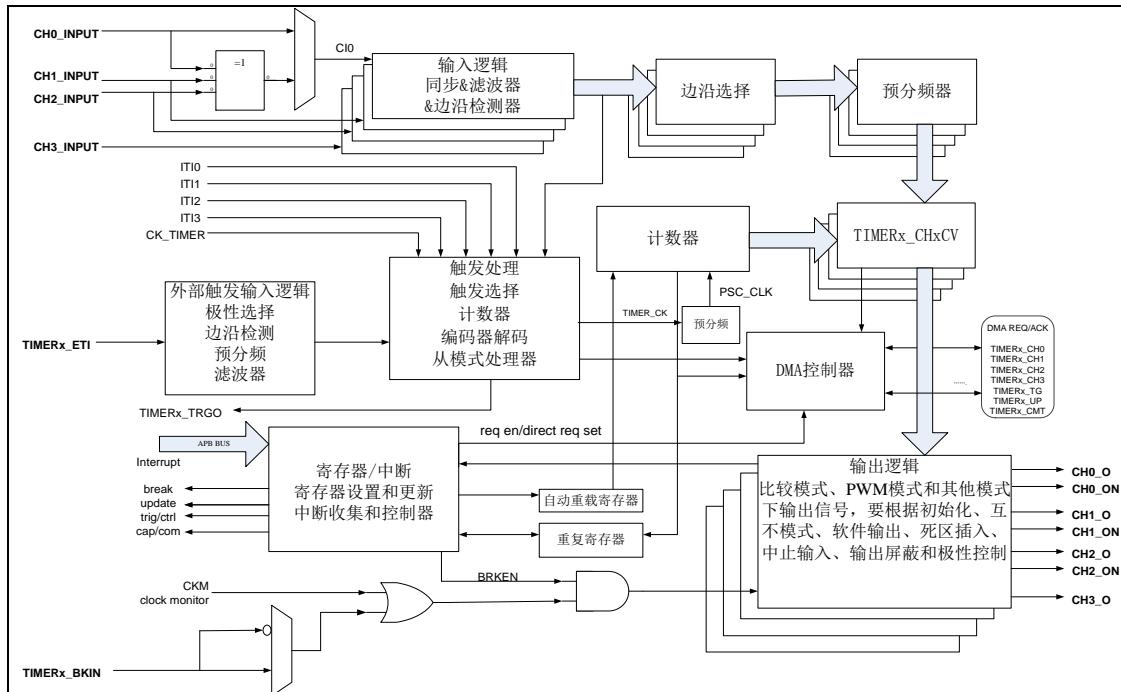
16.1.2. 主要特性

- 总通道数： 4;
- 计数器宽度： 16位;
- 时钟源可选： 内部时钟，内部触发，外部输入，外部触发;
- 多种计数模式： 向上计数，向下计数和中央计数;
- 正交编码器接口： 被用来追踪运动和分辨旋转方向和位置;
- 霍尔传感器接口： 用来做三相电机控制;
- 可编程的预分频器： 16位，运行时可以被改变;
- 每个通道可配置： 输入捕获模式，输出比较模式，可编程的PWM模式，单脉冲模式;
- 可编程的死区时间;
- 自动重装载功能;
- 可编程的计数器重复功能;
- 中止输入功能;
- 中断输出和DMA请求： 更新事件，触发事件，比较/捕获事件和中止事件;
- 多个定时器的菊花链使得一个定时器可以同时启动多个定时器;
- 定时器的同步允许被选择的定时器在同一个时钟周期开始计数;
- 定时器主/从模式控制器。

16.1.3. 结构框图

[图 16-1. 高级定时器结构框图](#)提供了高级定时器的内部配置细节

图 16-1. 高级定时器结构框图



16.1.4. 功能描述

时钟源选择

高级定时器可以由内部时钟源TIMER_CK或者由SMC (TIMERx_SMCFG寄存器位[2:0]) 控制的复用时钟源驱动。

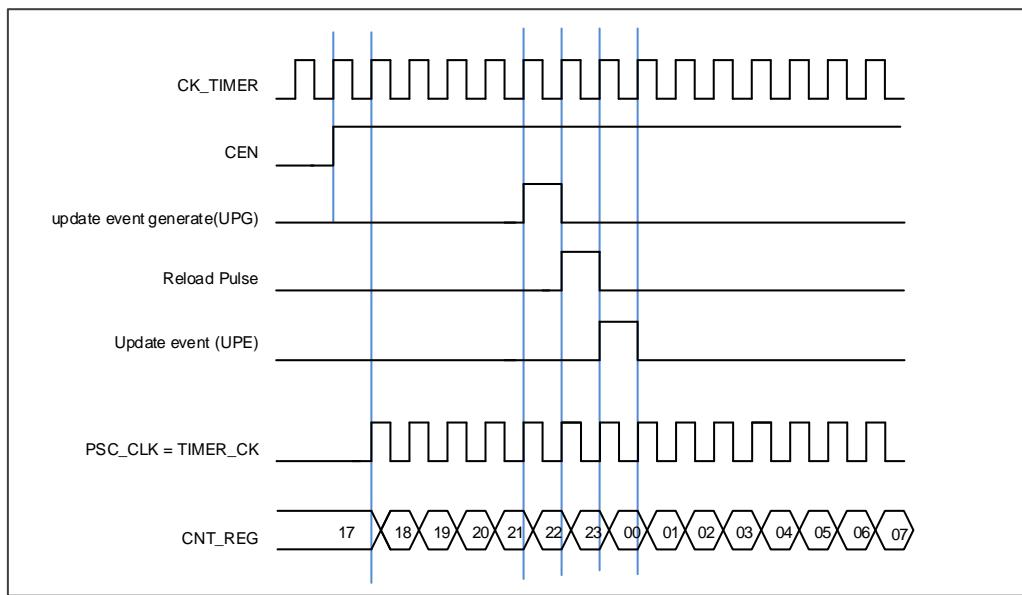
- SMC[2:0]==3'b000, 定时器选择内部时钟源 (连接到RCU模块的CK_TIMER)

如果禁能从模式 (SMC[2:0]==3'b000), 默认用来驱动计数器预分频器的是内部时钟源 CK_TIMER。当CEN置位, CK_TIMER经过预分频器 (预分频值由TIMERx_PSC寄存器确定) 产生PSC_CLK。

这种模式下, 驱动预分频器计数的TIMER_CK等于来自于RCU模块的CK_TIMER

如果使能从模式控制器(将TIMERx_SMCFG寄存器的SMC[2:0]设置为包括0x1、0x2、0x3和0x7), 预分频器被其他时钟源(由TIMERx_SMCFG寄存器的TRGS [2:0]区域选择)驱动, 在下文说明。当从模式选择位SMC被设置为0x4、0x5和0x6, 计数器预分频器时钟源由内部时钟 TIMER_CK驱动。

图 16-2. 内部时钟分频为 1 时正常模式下的控制电路



- SMC[2:0]==3'b111(外部时钟模式0)，定时器选择外部输入引脚作为时钟源

计数器预分频器可以在TIMERx_CI0/TIMERx_CI1引脚的每个上升沿或下降沿计数。这种模式可以通过设置SMC [2:0]为0x7同时设置TRGS[2:0]为0x4, 0x5或0x6来选择。

计数器预分频器也可以在内部触发信号ITI0/1/2/3的上升沿计数。这种模式可以通过设置SMC [2:0]为0x7同时设置TRGS [2:0]为0x0, 0x1, 0x2 或者0x3。

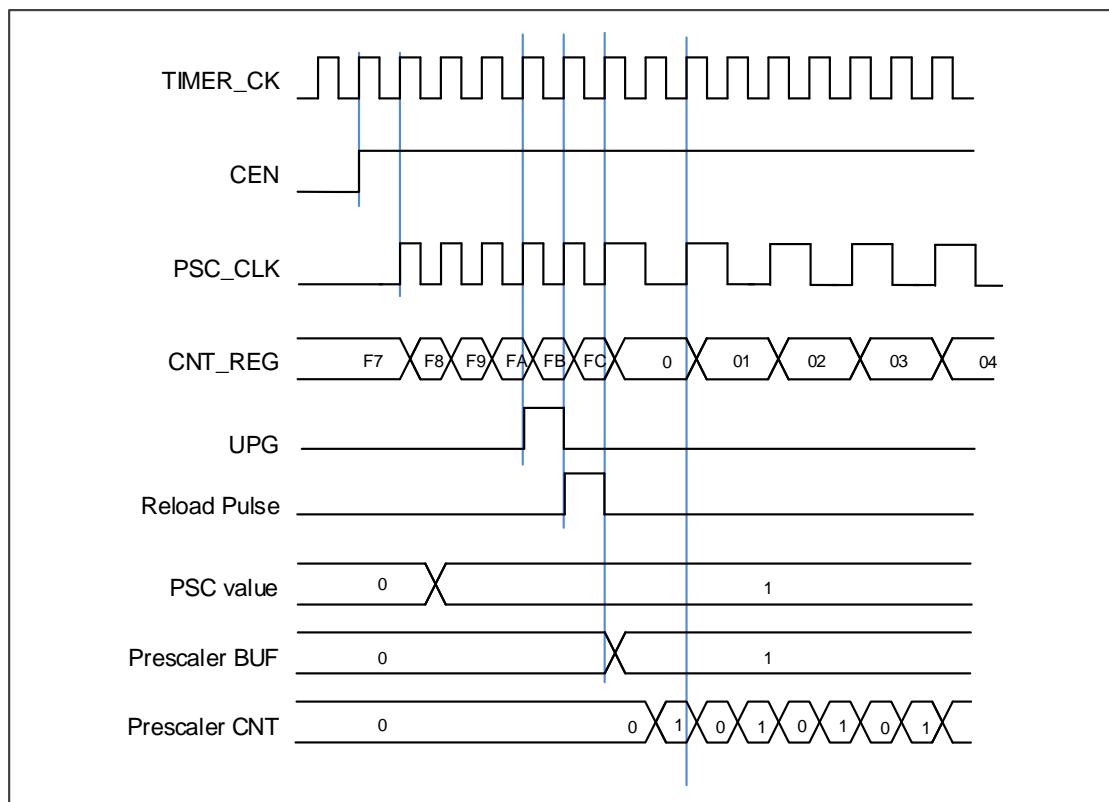
- SMC1==1'b1(外部时钟模式1)，定时器选择外部输入引脚ETI作为时钟源

计数器预分频器可以在外部引脚ETI的每个上升沿或下降沿计数。这种模式可以通过设置TIMERx_SMCFG寄存器中的SMC1位为1来选择。另一种选择ETI信号作为时钟源方式是，设置SMC [2:0]为0x7同时设置TRGS [2:0]为0x7。注意ETI信号是通过数字滤波器采样ETI引脚得到的。如果选择ETI信号为时钟源，触发控制器包括边沿监测电路将在每个ETI信号上升沿产生一个时钟脉冲来为计数器预分频器提供时钟。

预分频器

预分频器可以将定时器的时钟 (TIMER_CK) 频率按1到65536之间的任意值分频，分频后的时钟PSC_CLK驱动计数器计数。分频系数受预分频寄存器TIMERx_PSC控制，这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 16-3. 当预分频器的参数从 1 变到 2 时，计数器的时序图



向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从0开始向上连续计数到自动加载值（定义在**TIMERx_CAR**寄存器中），一旦计数器计数到自动加载值，会重新从0开始向上计数。如果设置了重复计数器，在(**TIMERx_CREP+1**)次上溢后产生更新事件，否则在每次上溢时都会产生更新事件。在向上计数模式中，**TIMERx_CTL0**寄存器中的计数方向控制位**DIR**应该被设置成0。

当通过**TIMERx_SWEVG**寄存器的**UPG**位置1来设置更新事件时，计数值会被清0，并产生更新事件。

如果**TIMERx_CTL0**寄存器的**UPDIS**置1，则禁止更新事件。

当发生更新事件时，所有的寄存器(重复计数器，自动重载寄存器，预分频寄存器)都将被更新。

下面这些图给出了一些例子，当**TIMERx_CAR=0x63**时，计数器在不同预分频因子下的行为。

图 16-4. 向上计数时序图, PSC=0/1

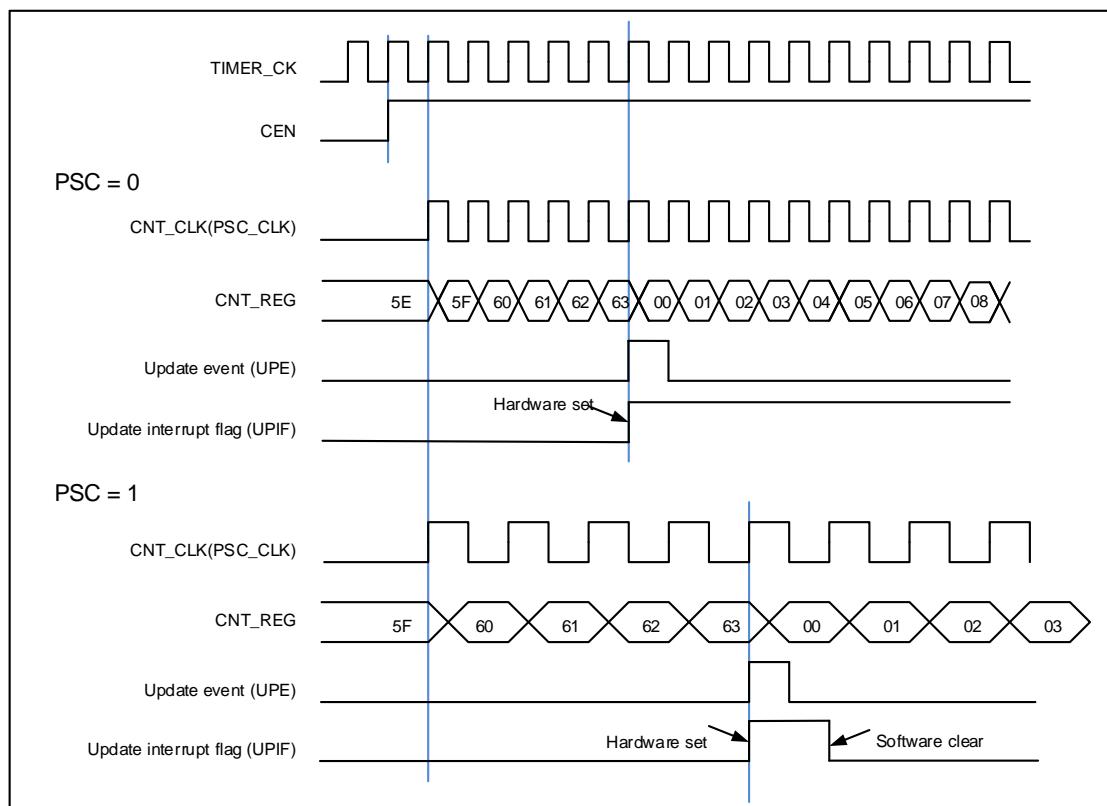
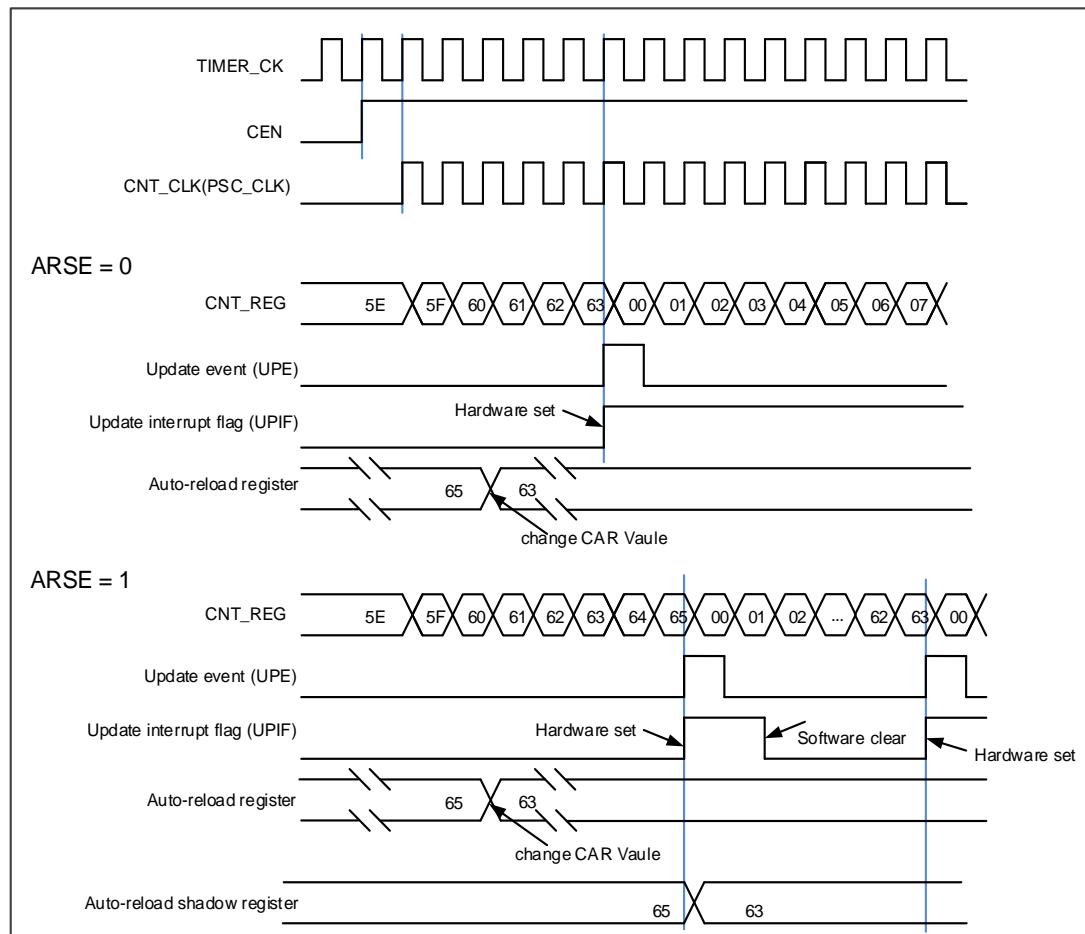


图 16-5. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



向下计数模式

在这种模式，计数器的计数方向是向下计数。计数器从自动加载值（定义在**TIMERx_CAR**寄存器中）向下连续计数到0。一旦计数器计数到0，计数器会重新从自动加载值开始计数。如果设置了重复计数器，在(**TIMERx_CREP+1**)次下溢后产生更新事件，否则在每次下溢时都会产生更新事件。在向下计数模式中，**TIMERx_CTL0**寄存器中的计数方向控制位**DIR**应该被设置成1。

当通过**TIMERx_SWEVG**寄存器的**UPG**位置1来设置更新事件时，计数值会被初始化为自动加载值，并产生更新事件。

如果**TIMERx_CTL0**寄存器的**UPDIS**置1，则禁止更新事件。

当发生更新事件时，所有的寄存器(重复计数器，自动重载寄存器，预分频寄存器)都将被更新。

下面这些图给出了一些例子，当**TIMERx_CAR=0x63**时，计数器在不同时钟频率下的行为。

图 16-6. 向下计数时序图, PSC=0/1

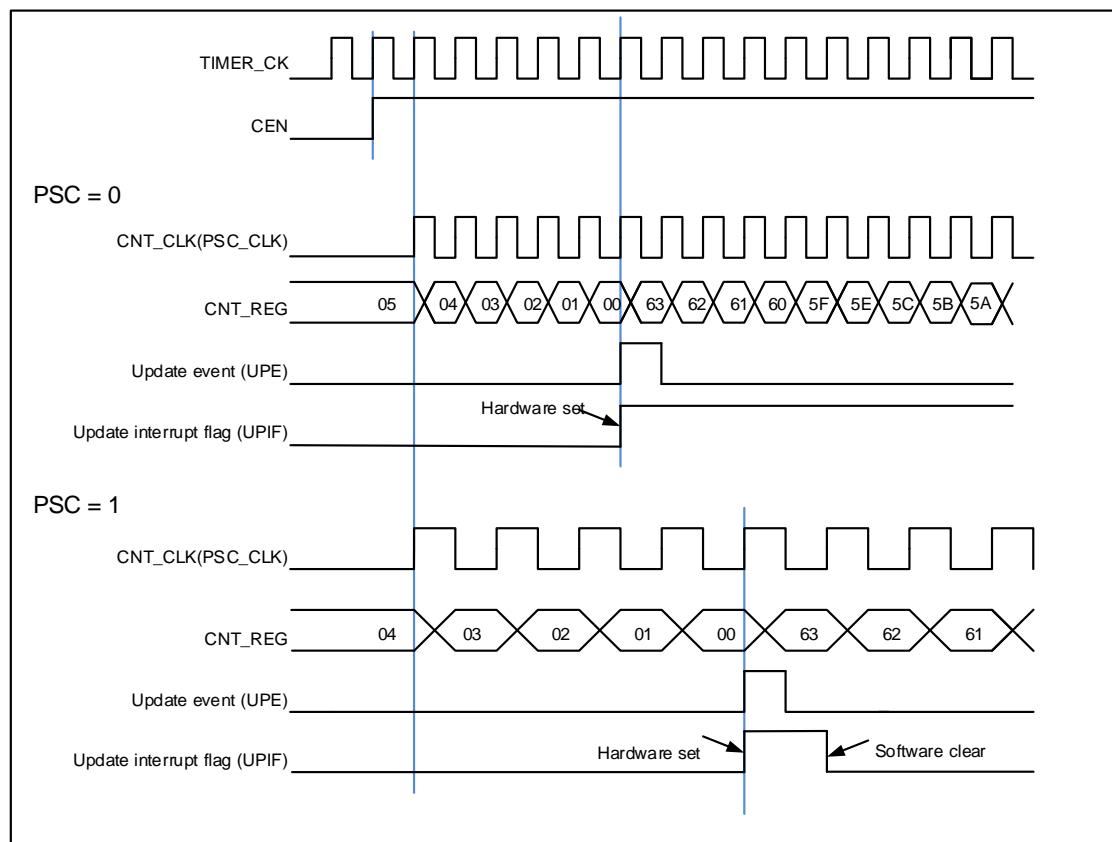
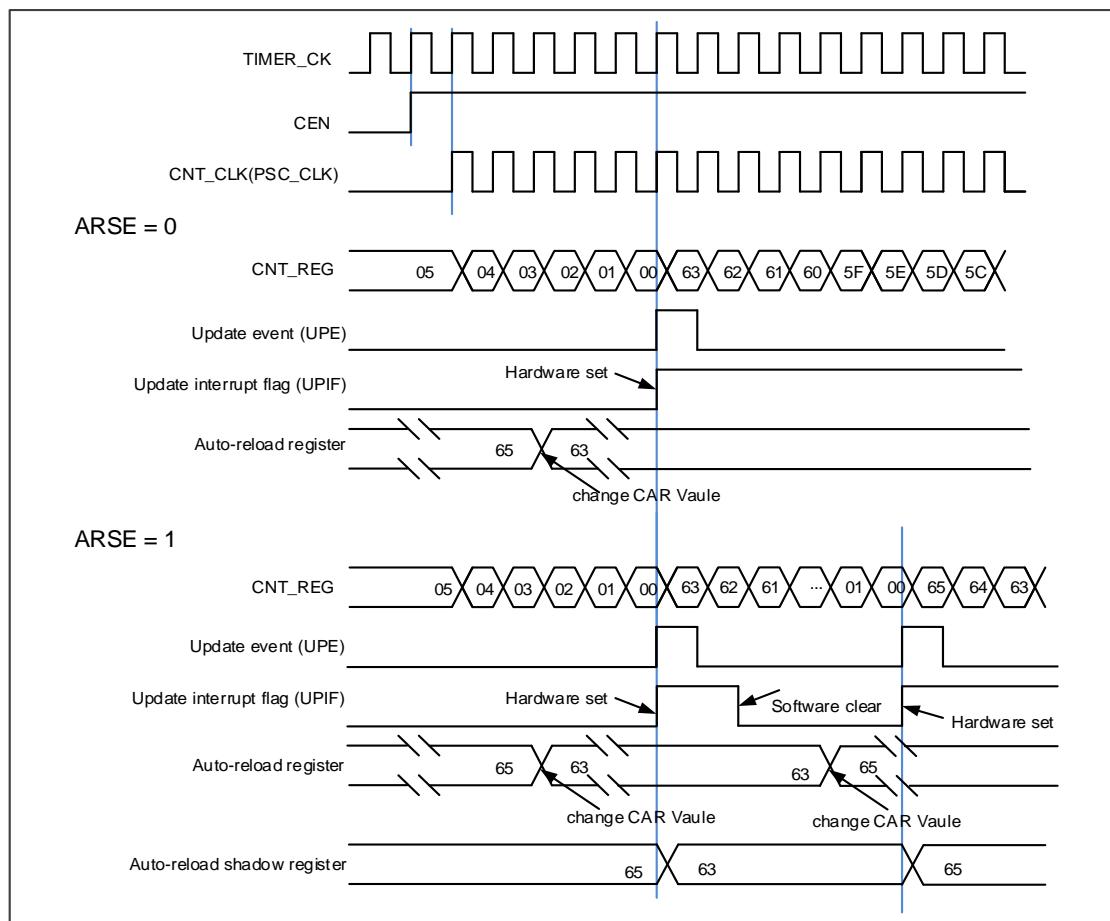


图 16-7. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值



中央对齐模式

在中央对齐模式下，计数器交替的从0开始向上计数到自动加载值，然后再向下计数到0。在向上计数模式中，定时器模块在计数器计数到自动加载值-1产生一个上溢事件；向下计数模式中，定时器模块在计数器计数到1时产生一个下溢事件。在中央计数模式中，TIMERx_CTL0寄存器中的计数方向控制位DIR只读，表明了的计数方向。计数方向被硬件自动更新。

将TIMERx_SWEVG寄存器的UPG位置1可以初始化计数值为0，并产生一个更新事件，而无需考虑计数器在中央模式下是向上计数还是向下计数。

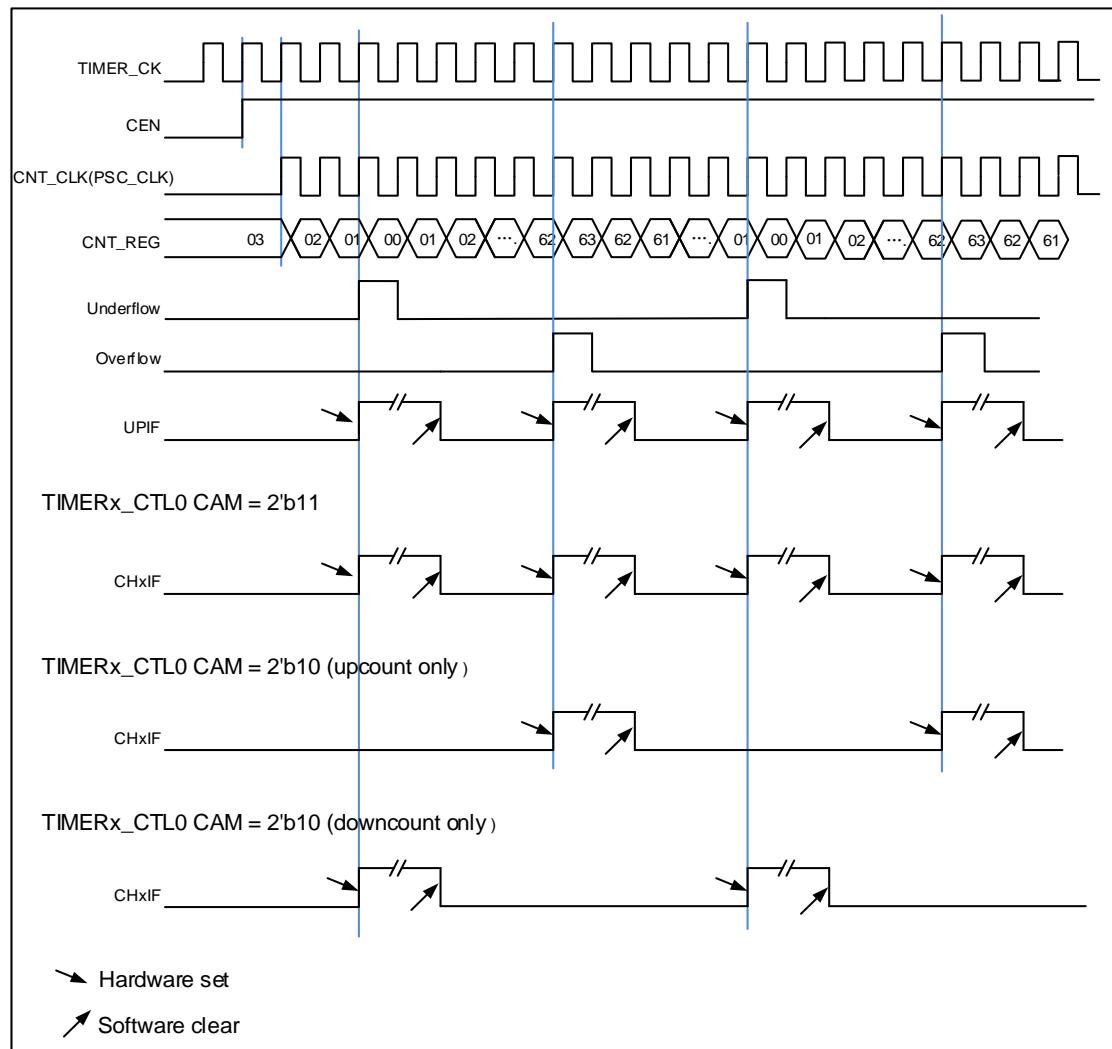
上溢或者下溢时，TIMERx_INTF寄存器中的UPIF位都会被置1，然而CHxIF位置1与TIMERx_CTL0寄存器中CAM的值有关。具体细节参考图18-8。

如果TIMERx_CTL0寄存器的UPDIS置1，则禁止更新事件。

当发生更新事件时，所有的寄存器(重复计数器，自动重载寄存器，预分频寄存器)都将被更新。

[图 16-8. 中央计数模式计数器时序图](#)给出了一些例子，当 TIMERx_CAR=0x63，TIMERx_PSC=0x0 时，计数器的行为

图 16-8. 中央计数模式计数器时序图



重复计数器

重复计数器是用来在 $N+1$ 个计数周期之后产生更新事件，更新定时器的寄存器， N 为TIMERx_CREP寄存器的CREP。向上计数模式下，重复计数器在每次计数器上溢时递减；向下计数模式下，重复计数器在每次计数器下溢时递减；在中央对齐模式下，重复计数器在计数器上溢和下溢时递减。

将TIMERx_SWEVG寄存器的UPG位置1可以重载TIMERx_CREP寄存器中CREP的值并产生一个更新事件。

在中央对齐模式下，对于CREP为奇数值，更新事件发生在上溢或下溢的时刻取决于奇数值写入CREP寄存器和计数器启动的时刻。如果在计数器启动前写入CREP寄存器，则在上溢时产生更新事件。如果在计数器启动后写入CREP寄存器，则在下溢时产生更新事件。

图 16-9. 中央计数模式下计数器重复时序图

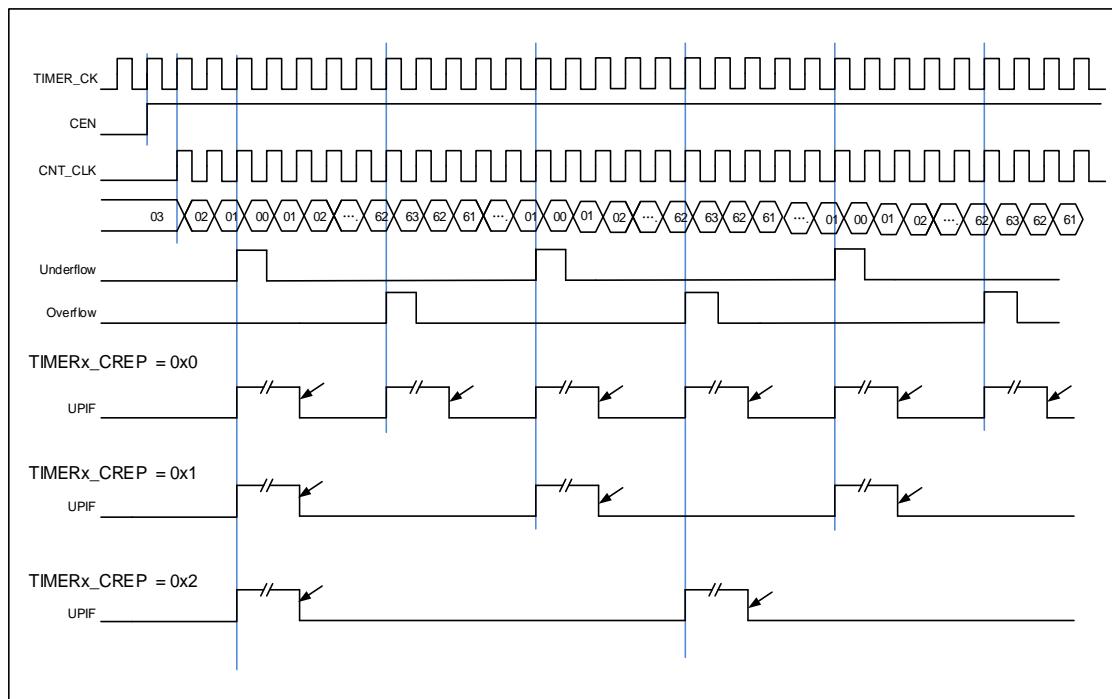


图 16-10. 在向上计数模式下计数器重复时序图

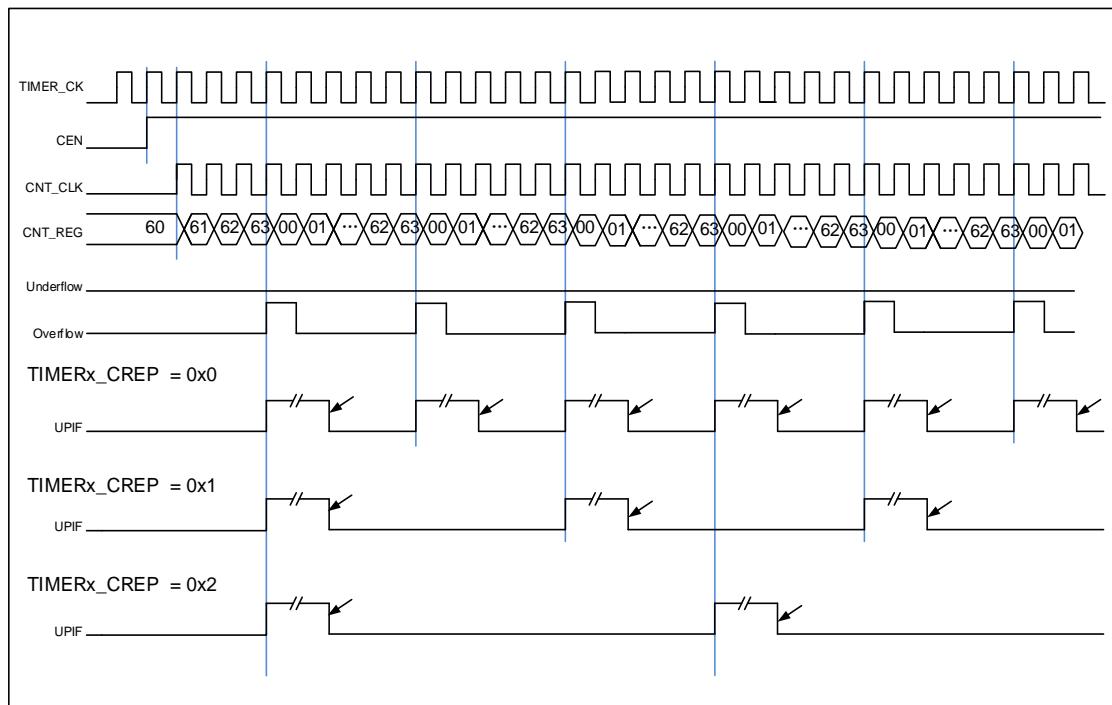
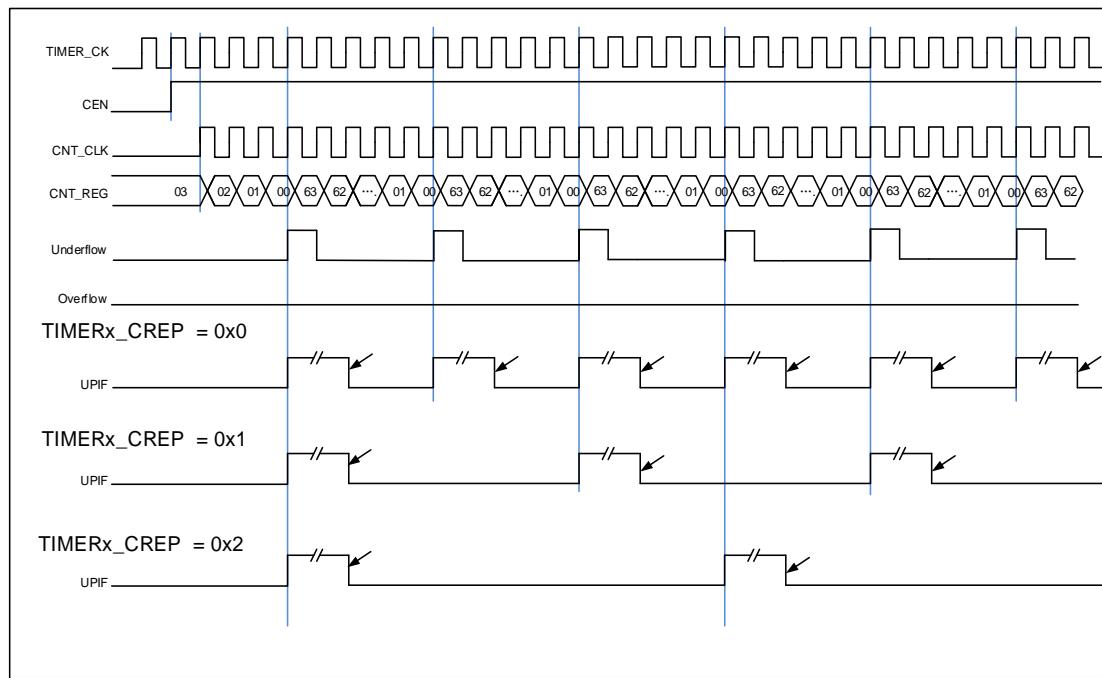


图 16-11. 在向下计数模式下计数器重复时序图



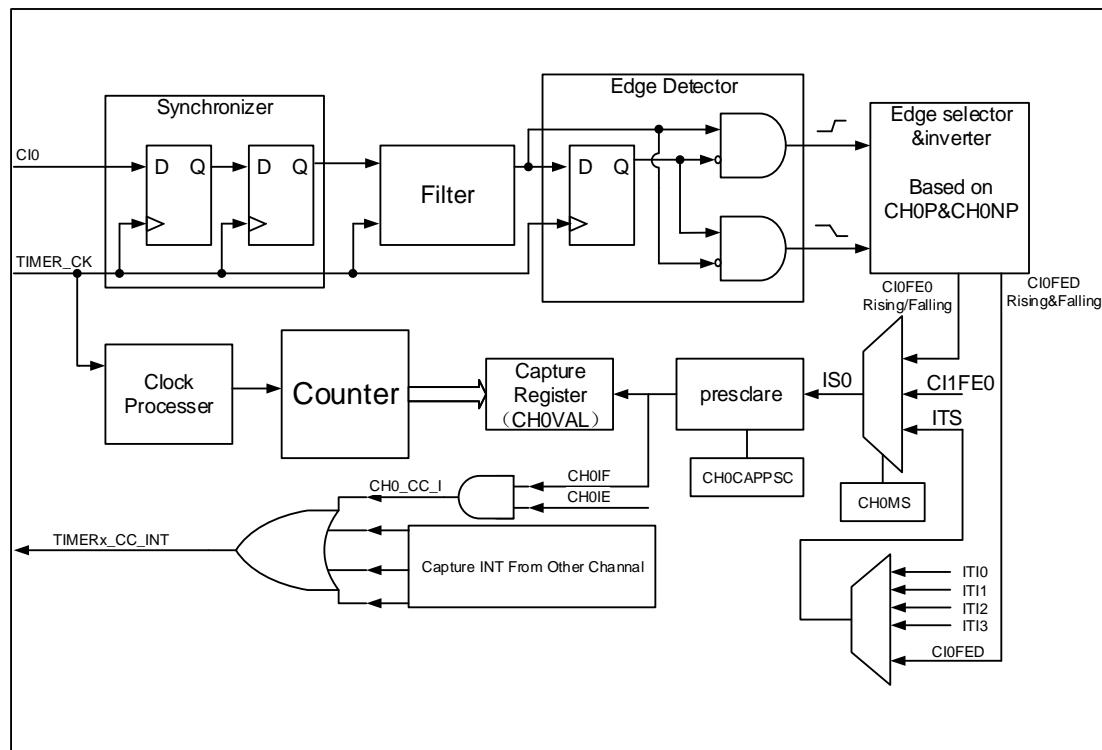
捕获/比较通道

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

■ 输入捕获模式

捕获模式允许通道测量一个波形时序，频率，周期，占空比等。输入级包括一个数字滤波器，一个通道极性选择，边沿检测和一个通道预分频器。如果在输入引脚上出现被选择的边沿，**TIMERx_CHxCV**寄存器会捕获计数器当前的值，同时**CHxIF**位被置1，如果**CHxIE = 1**则产生通道中断。

图 16-12. 输入捕获逻辑



通道输入信号 CIx 有两种选择，一种是 $TIMERx_CHx$ 信号，另一种是 $TIMERx_CH0, TIMERx_CH1$ 和 $TIMERx_CH2$ 异或之后的信号。通道输入信号 CIx 先被 $TIMER_CK$ 信号同步，然后经过数字滤波器采样，产生一个被滤波后的信号。通过边沿检测器，可以选择检测上升沿或者下降沿。通过配置 $CHxP$ 选择使用上升沿或者下降沿。配置 $CHxMS$ ，可以选择其他通道的输入信号，内部触发信号。配置 IC 预分频器，使得若干个输入事件后才产生一个有效的捕获事件。捕获事件发生， $CHxVAL$ 存储计数器的值。

配置步骤如下：

第一步：滤波器配置（ $TIMERx_CHCTL0$ 寄存器中 $CHxCAPFLT$ ）：

根据输入信号和请求信号的质量，配置相应的 $CHxCAPFLT$ 。

第二步：边沿选择（ $TIMERx_CHCTL2$ 寄存器中 $CHxP/CHxNP$ ）：

配置 $CHxP/CHxNP$ 选择上升沿或者下降沿。

第三步：捕获源选择（ $TIMERx_CHCTL0$ 寄存器中 $CHxMS$ ）：

一旦通过配置 $CHxMS$ 选择输入捕获源，必须确保通道配置在输入模式 ($CHxMS \neq 0x0$)，而且 $TIMERx_CHxCV$ 寄存器不能再被写。

第四步：中断使能（ $TIMERx_DMAINTEN$ 寄存器中 $CHxIE$ 和 $CHxDEN$ ）：

使能相应中断，可以获得中断和 DMA 请求。

第五步：捕获使能（ $TIMERx_CHCTL2$ 寄存器中 $CHxEN$ ）。

结果：当期望的输入信号发生时， $TIMERx_CHxCV$ 被设置成当前计数器的值， $CHxIF$ 为置 1。

如果 $CHxIF$ 位已经为 1，则 $CHxOF$ 位置 1。根据 $TIMERx_DMAINTEN$ 寄存器中 $CHxIE$ 和 $CHxDEN$ 的配置，相应的中断和 DMA 请求会被提出。

直接产生: 软件设置CHxG位，会直接产生中断和DMA请求。

输入捕获模式也可用来测量TIMERx_CHx引脚上信号的脉冲波宽度。例如，一个PWM波连接到CI0。配置TIMERx_CHCTL0寄存器中CH0MS为2'b01，选择通道0的捕获信号为CI0并设置上升沿捕获。配置TIMERx_CHCTL0寄存器中CH1MS为2'b10，选择通道1捕获信号为CI0并设置下降沿捕获。计数器配置为复位模式，在通道0的上升沿复位。TIMERX_CH0CV寄存器测量PWM的周期值，TIMERx_CH1CV寄存器测量PWM占空比值。

■ 输出比较模式

在输出比较模式，TIMERx可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的CHxVAL寄存器与计数器的值匹配时，根据CHxCOMCTL的配置，这个通道的输出可以被置高电平，被置低电平或者反转。当计数器的值与CHxVAL寄存器的值匹配时，CHxIF位被置1，如果CHxIE = 1则会产生中断，如果CHxDEN=1则会产生DMA请求。

配置步骤如下：

第一步：时钟配置：

配置定时器时钟源，预分频器等。

第二步：比较模式配置：

设置CHxCOMSEN位来配置输出比较影子寄存器；

设置CHxCOMCTL位来配置输出模式（置高电平/置低电平/反转）；

设置CHxP/CHxNP位来选择有效电平的极性；

设置CHxEN使能输出。

第三步：通过CHxIE/CHxDEN位配置中断/DMA请求使能。

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

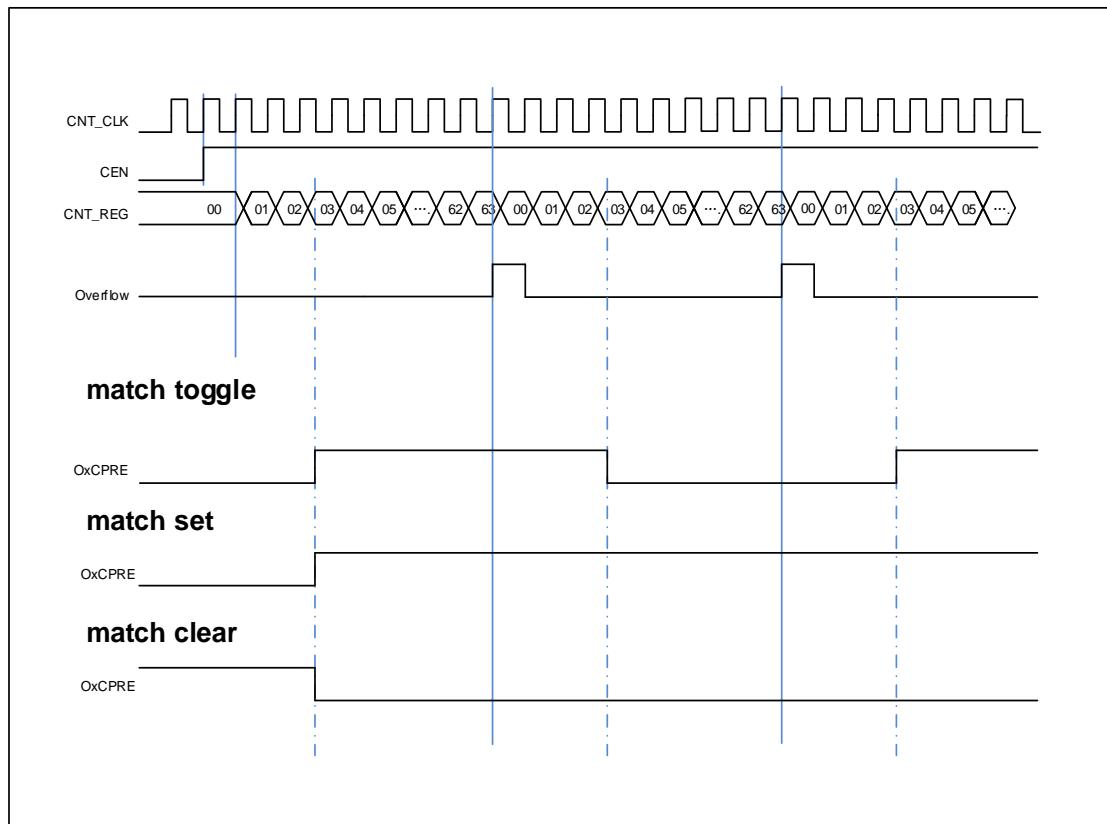
CHxVAL可以在运行时根据你所期望的波形而改变。

第五步：设置CEN位使能定时器。

[**图 16-13. 三种输出比较模式**](#)显示了三种比较输出模式：反转/置高电平/置低电平，CAR=0x63，

CHxVAL=0x3。

图 16-13. 三种输出比较模式



PWM 模式

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

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

EAPWM 的周期由 TIMERx_CAR 寄存器值决定，占空比由 TIMERx_CHxCV 寄存器值决定。[图 16-14. EAPWM 时序图](#)显示了 EAPWM 的输出波形和中断。

CAPWM 的周期由 (2*TIMERx_CAR 寄存器值) 决定，占空比由 (2*TIMERx_CHxCV 寄存器值) 决定。[图 16-15. CAPWM 时序图](#)显示了 CAPWM 的输出波形和中断。

在 PWM0 模式下 (CHxCOMCTL==3'b110)，如果 TIMERx_CHxCV 寄存器的值大于 TIMERx_CAR 寄存器的值，通道输出一直为有效电平。

在 PWM0 模式下 (CHxCOMCTL==3'b110)，如果 TIMERx_CHxCV 寄存器的值等于 0，通道输出一直为无效电平。

图 16-14. EAPWM 时序图

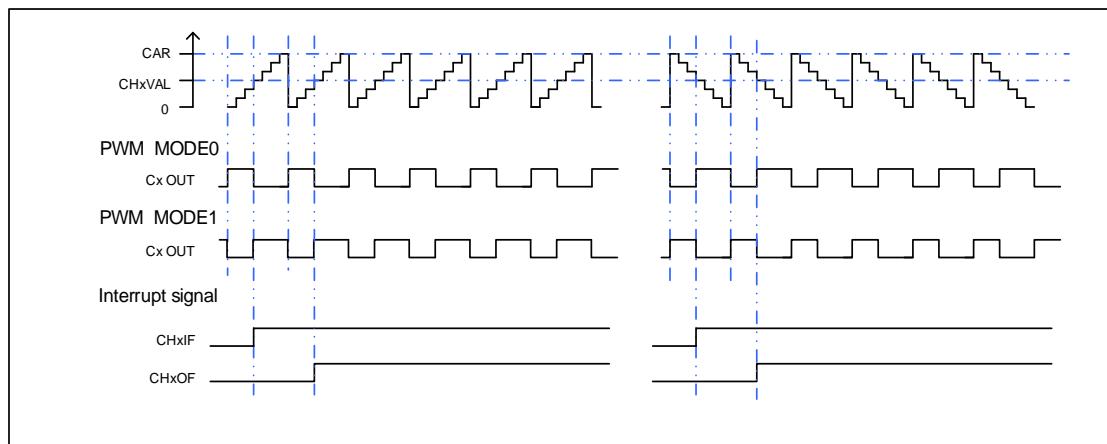
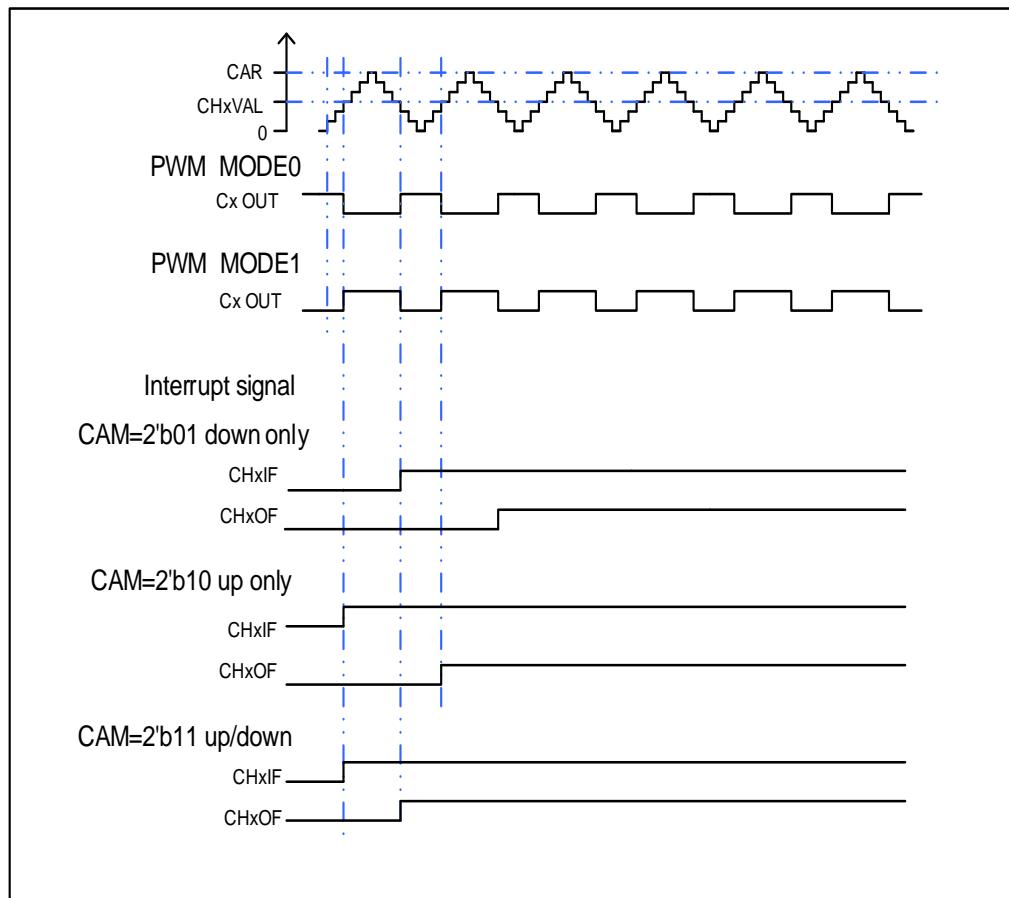


图 16-15. CAPWM 时序图



通道输出参考信号

当TIMERx用于输出匹配比较模式下，设置CHxCOMCTL位可以定义OxCPR_E信号(通道x准备信号)类型。OxCPR_E信号有若干类型的输出功能，包括，设置CHxCOMCTL=0x00可以保持原始电平；设置CHxCOMCTL=0x01可以将OxCPR_E信号设置为高电平；设置

CHxCOMCTL=0x02可以将**OxCPR**E信号设置为低电平；设置**CHxCOMCTL=0x03**，在计数器值和**TIMERx_CHxCV**寄存器的值匹配时，可以翻转输出信号。

PWM模式0和PWM模式1是**OxCPR**E的另一种输出类型，设置**CHxCOMCTL**位域位**0x06**或**0x07**可以配置PWM模式0/PWM模式1。在这些模式中，根据计数器值和**TIMERx_CHxCV**寄存器值的关系以及计数方向，**OxCPR**E信号改变其电平。具体细节描述，请参考相应的位。

设置**CHxCOMCTL =0x04**或**0x05**可以实现**OxCPR**E信号的强制输出功能。输出比较信号能够直接由软件强置为有效或无效状态，而不依赖于**TIMERx_CHxCV**的值和计数器值之间间的比较结果。

设置**CHxCOMCEN=1**，当由外部**ETI**引脚信号产生的**ETIFE**信号为高电平时，**OxCPR**E被强制为低电平。在下一次更新事件到来时，**OxCPR**E信号才会回到有效电平状态。

互补输出

CHx_O和**CHx_ON**是一对互补输出通道，这两个信号不能同时有效。**TIMERx**有四路通道，只有前三路有互补输出通道。互补信号**CHx_O** 和 **CHx_ON** 是由一组参数来决定：**TIMERx_CHCTL2** 寄存器中的 **CHxEN** 和 **CHxNEN** 位，**TIMERx_CCHP** 寄存器中和 **TIMERx_CTL1** 寄存器中的 **POEN**, **ROS**, **IOS**, **ISOx** 和 **ISOxN** 位。输出极性由 **TIMERx_CHCTL2**寄存器中的**CHxP**和**CHxNP**位来决定。

表 16-2. 由参数控制的互补输出表

互补参数					输出状态	
POEN	ROS	IOS	CHxEN	CHxNEN	CHx_O	CHx_ON
0	0/1	0	0	0	CHx_O / CHx_ON = LOW CHx_O / CHx_ON 输出禁用.	
				1	CHx_O = CHxP CHx_ON = CHxNP CHx_O/CHx_ON 输出禁用.	
			1	0	如果时钟使能: CHx_O = ISOx CHx_ON = ISOxN	
				1	CHx_O = CHxP CHx_ON = CHxNP CHx_O/CHx_ON输出禁用.	
		1	0	0	CHx_O = CHxP CHx_ON = CHxNP CHx_O/CHx_ON输出禁用.	
				1	CHx_O = CHxP CHx_ON = CHxNP CHx_O/CHx_ON 输出使能.	
			1	0	如果时钟使能: CHx_O = ISOx CHx_ON = ISOxN	
				1	CHx_O = CHxP CHx_ON = CHxNP CHx_O/CHx_ON输出禁用.	
1	0	0/1	0	0	CHx_O/CHx_ON = LOW	

互补参数					输出状态	
POEN	ROS	IOS	CHxEN	CHxNEN	CHx_O	CHx_ON
					CHx_O/CHx_ON 输出禁用.	
					1	CHx_O = LOW CHx_O 输出禁用.
					0	CHx_O=OxCPR _E ⊕ CHxP CHx_O输出使能
					1	CHx_O=OxCPR _E ⊕ CHxP CHx_O输出使能
				0	0	CHx_O = CHxP CHx_O输出禁用.
					1	CHx_O = CHxP CHx_O输出使能
				1	0	CHx_O=OxCPR _E ⊕ CHxP CHx_O输出使能
					1	CHx_O=OxCPR _E ⊕ CHxP CHx_O输出使能
				0	CHx_O = CHxP CHx_O输出使能	CHx_ON = CHxNP CHx_ON输出禁用.
					1	CHx_ON=OxCPR _E ⊕ CHxNP CHx_ON输出使能
				1	CHx_O = CHxP CHx_O输出使能	CHx_ON = CHxNP CHx_ON输出使能
					1	CHx_ON=OxCPR _E ⊕ CHxNP CHx_ON输出使能

死区时间插入

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

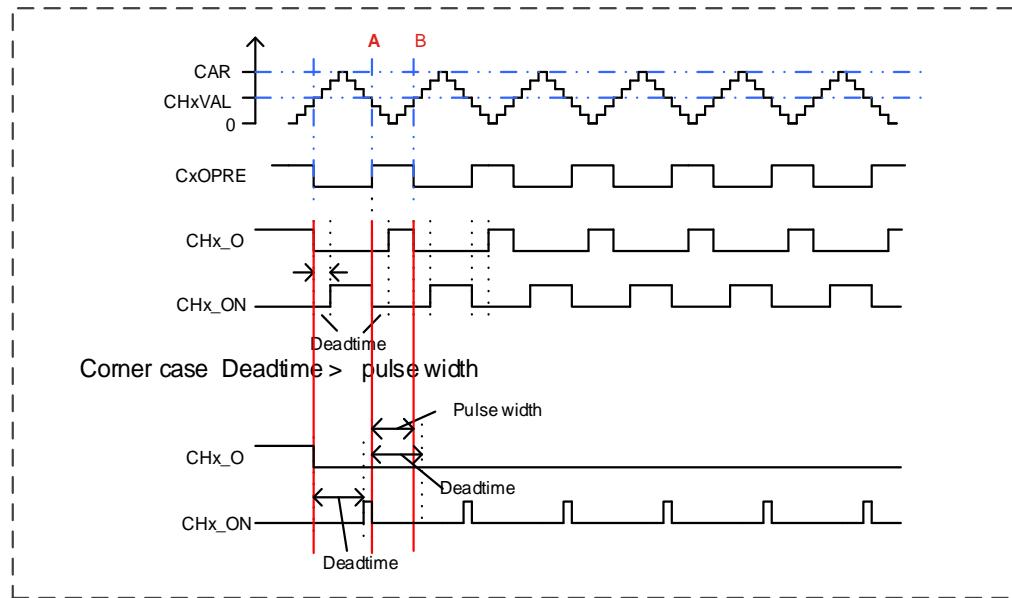
死区时间的插入，确保了通道互补的两路信号不会同时有效。

在 PWM0 模式，当通道 x 匹配发生时 (TIMERx 计数器=CHxVAL)，OxCPR_E 反转。在 [图 16-16. 带死区时间的互补输出](#) 中的 A 点，CHx_O 信号在死区时间内为低电平，直到死区时间过后才变为高电平，而 CHx_ON 信号立刻变为低电平。同样，在 B 点，计数器再次匹配 (TIMERx 计数器=CHxVAL)，OxCPR_E 信号被清 0，CHx_O 信号被立即清零，CHx_ON 信号在死区时间内仍然是低电平，在死区时间过后才变为高电平。

有时会有一些死角事件发生，例如：

- 如果死区延时大于或者等于CHx_O信号的占空比, CHx_O信号一直为无效值(如图16-16. 带死区时间的互补输出)。
- 如果死区延时大于或者等于CHx_ON信号的占空比, CHx_ON信号一直为无效值。

图 16-16. 带死区时间的互补输出



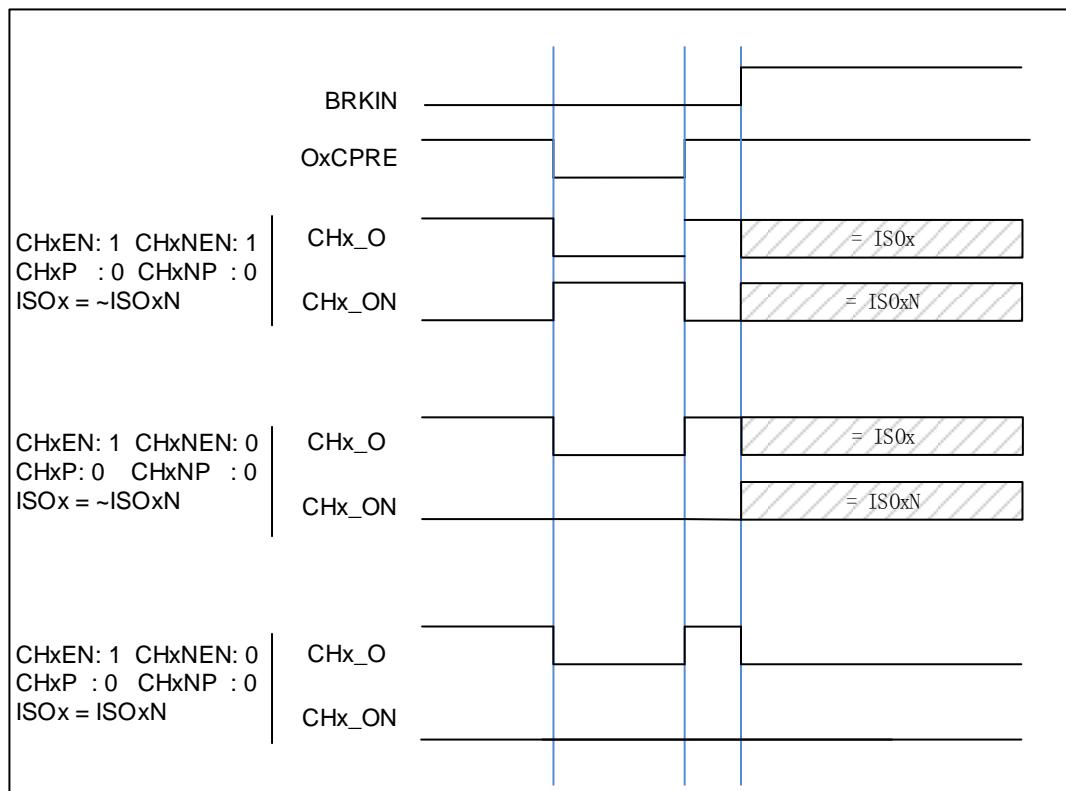
中止功能

使用中止功能时, 输出CHx_O和CHx_ON信号电平被以下位控制, TIMERx_CCHP寄存器的POEN, IOS和ROS位, TIMERx_CTL1寄存器的ISOx和ISOxN位。当中止事件发生时, CHx_O和CHx_ON信号输出不能同时设置为有效电平。中止源可以选择中止输入引脚, 也可以选择HXTAL时钟失效事件。时钟失效事件由RCU中的时钟监视器(CKM)产生。将TIMERx_CCHP寄存器的BRKEN位置1可以使能中止功能。TIMERx_CCHP寄存器的BRKP位决定了中止输入极性。

发生中止时, POEN位被异步清除, 一旦POEN位为0, CHx_O和CHx_ON被TIMERx_CTL1寄存器中的ISOx位和ISOxN驱动。如果IOS=0, 定时器释放输出使能, 否则输出使能仍然为高。起初互补输出被置于复位状态, 然后死区时间产生器重新被激活, 以便在一个死区时间后驱动输出, 输出电平由ISOx和ISOxN位配置。

发生中止时, TIMERx_INTF寄存器的BRKIF位被置1。如果BRKIE=1, 中断产生。

图 16-17. 通道响应中止输入（高电平有效）时，输出信号的行为



正交译码器

正交译码器功能使用由TIMERx_CH0和TIMERx_CH1引脚生成的CI0和CI1正交信号各自相互作用产生计数值。在每个输入源改变期间，DIR位被硬件自动改变。输入源可以是只有CI0，可以只有CI1，或者可以同时有TI1和TI2，通过设置SMC=0x01, 0x02或0x03来选择使用哪种模式。计数器计数方向改变的机制[表16-3. 计数方向与编码器信号之间的关系](#)所示。正交译码器可以当作一个带有方向选择的外部时钟，这意味着计数器会在0和自动加载值之间连续的计数。因此，用户必须在计数器开始计数前配置TIMERx_CAR寄存器。

表 16-3. 计数方向与编码器信号之间的关系

计数模式	电平	TI1FP1		TI2FP2	
		上升	下降	上升	下降
只有CI0	CI1FE1=1	向下	向上	-	-
	CI1FE1=0	向上	向下	-	-
只有CI1	CI0FE0=1	-	-	向上	向下
	CI0FE0=0	-	-	向下	向上
CI0和CI1	CI1FE1=1	向下	向上	X	X
	CI1FE1=0	向上	向下	X	X
	CI0FE0=1	X	X	向上	向下
	CI0FE0=0	X	X	向下	向上

注意: “-”意思是“无计数”; “X”意思是不可能。

图 16-18. 编码器接口模式下计数器运行例子

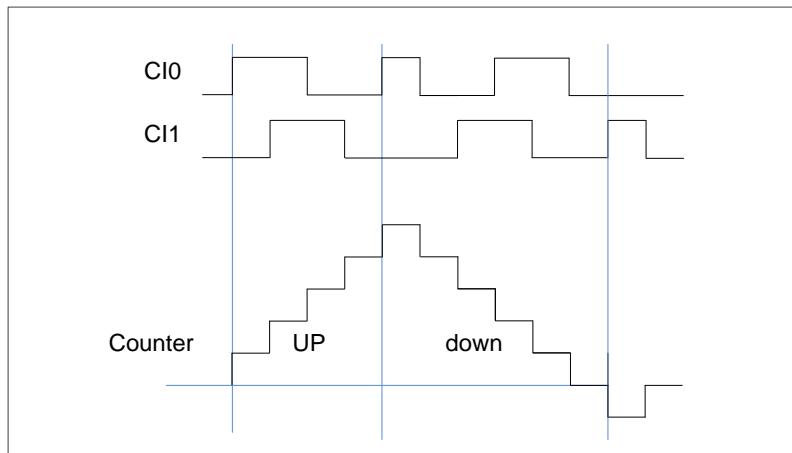
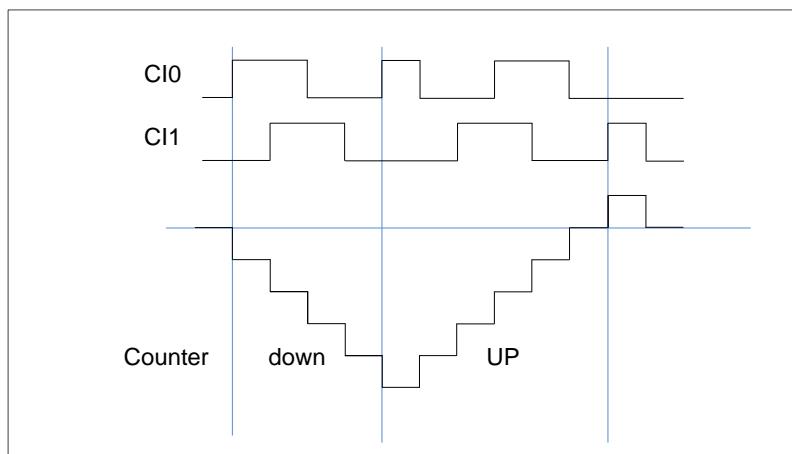


图 16-19. CI0/FE0 极性反相的编码器接口模式下的例子



霍尔传感器接口功能

高级定时器支持霍尔传感器接口功能，该功能可以用来控制BLDC电机。

图 16-20. 霍尔传感器用在 BLDC 电机控制中是定时器和电机的连接示意图。TIMER_in 定时器（可以是高级定时器或者通用 L0 定时器）接收来自电机霍尔传感器的三路信号，这三路信号是电机转子的位置信号。

三个霍尔传感器与TIMER_in定时器的三路输入捕获引脚一一对应连接，每个霍尔传感器输入一路波形到输入引脚，分析三路霍尔信号可以计算出转子的位置和速度。

通过定时器内部连接，例如TRGO-ITIx，TIMER_in定时器和TIMER_out定时器可以连接在一起。TIMER_out定时器根据ITIx触发信号输出PWM波，驱动BLDC电机，控制BLDC电机的速度。这样，TIMER_in定时器和TIMER_out定时器的连接形成了一个反馈电路，可以根据需求改变配置。

TIMER_in定时器需要具备输入异或功能，所以可以选择高级定时器和通用L0定时器。

TIMER_out定时器需要具备互补输出和死区插入功能，所以可以选择高级定时器。另外，根据定时器的内部互连关系，可以选择成对的互连定时器，例如：

TIMER_in (TIMER1) -> TIMER_out (TIMER0 ITI1)

等等。

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

- 设置TIOS，使能异或功能。三路输入信号的任何一路发生变化，CI0都会反转，CH0VAL此时会捕获计数器的当前值。
- 设置CCUC和CCSE，使能ITIx直接连接到换相功能。
- 根据需求配置PWM参数。

图 16-20. 霍尔传感器用在 BLDC 电机控制中

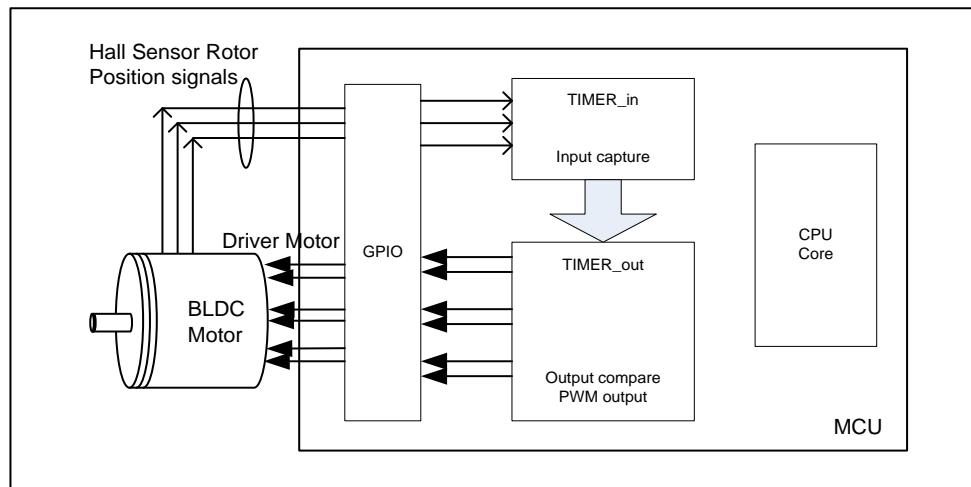
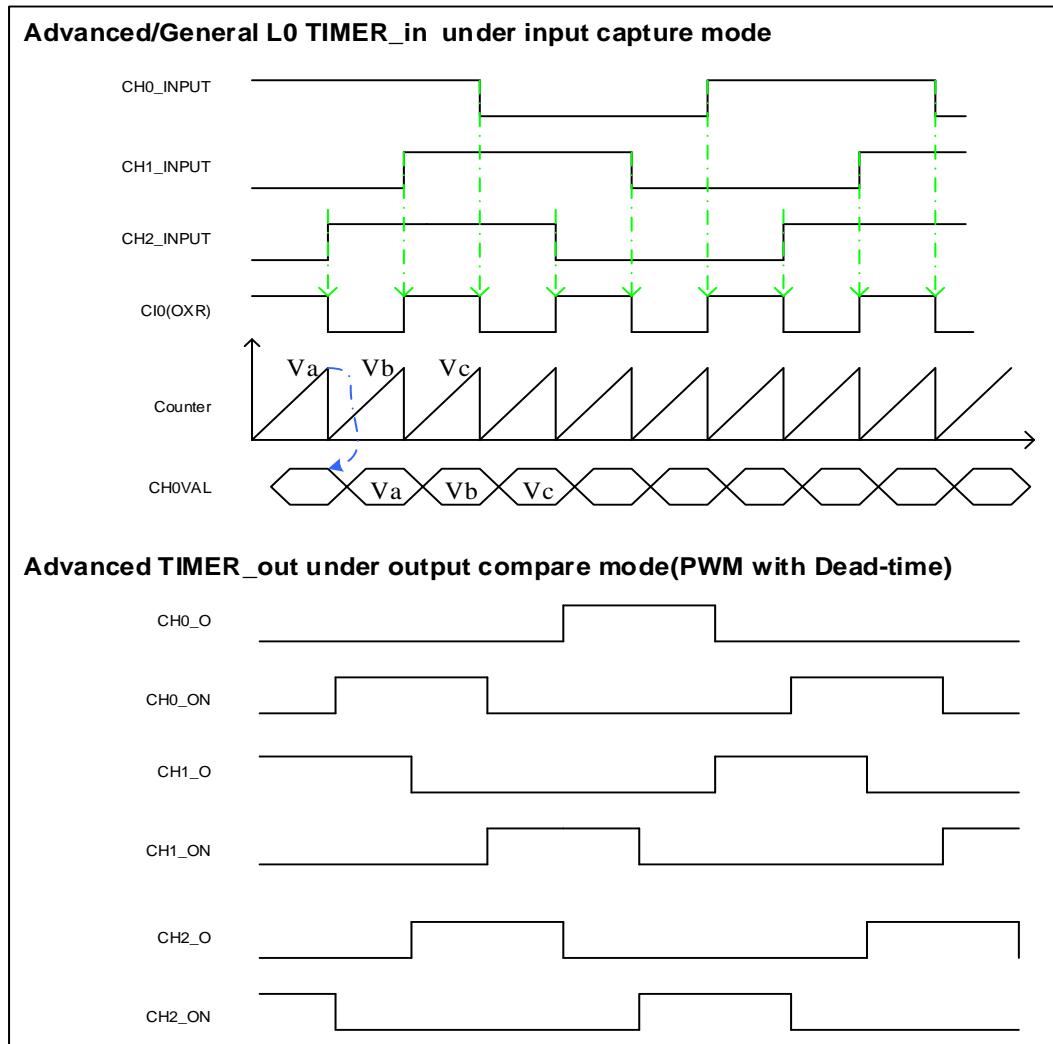


图 16-21. 两个定时器之间的霍尔传感器时序图

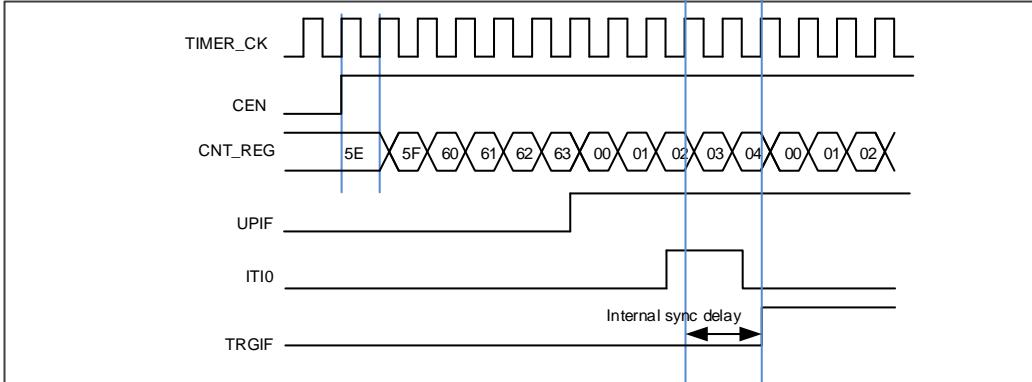
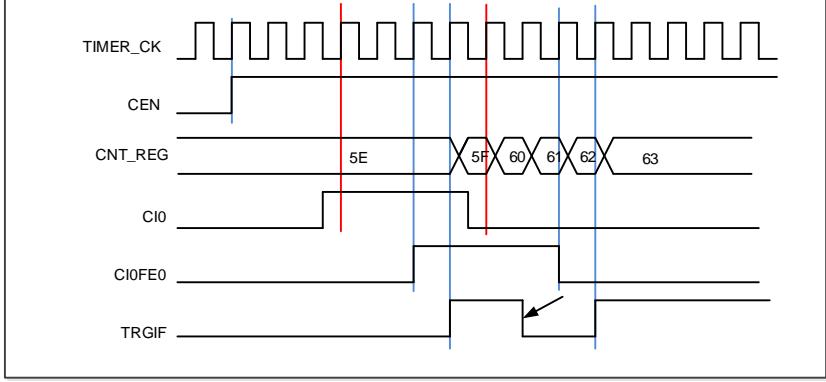


从控制器

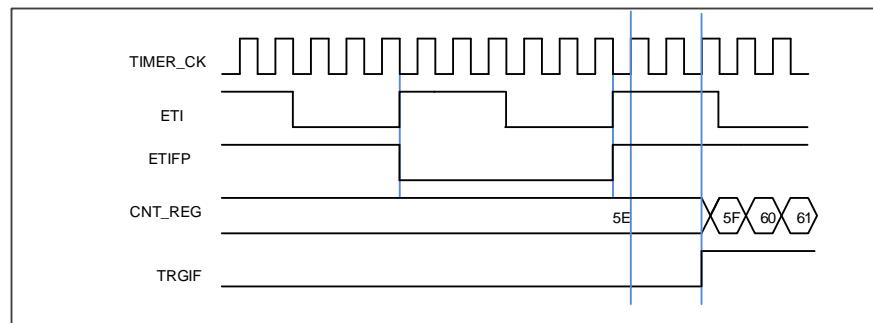
TIMERx能在多种模式下同步外部触发，包括复位模式，暂停模式和事件模式，可以通过设置TIMERx_SMCFG寄存器中的SMC [2:0]配置这些模式。这些模式的输入触发源可以通过设置TIMERx_SMCFG寄存器中的TRGS [2:0]来选择。

表 16-4. 从模式例子列表

	模式选择	触发源选择	极性选择	滤波和预分频
列举	SMC[2:0] 3'b100 (复位模式) 3'b101 (暂停模式) 3'b110 (事件模式)	TRGS[2:0] 000: ITI0 001: ITI1 010: ITI2 011: ITI3 100: CI0F_ED 101: CI0FE0 110: CI1FE1	如果触发源是CI0FE0或者CI1FE1，配置CHxP和CHxNP来选择极性和反相 如果触发源是ETIF，配置ETP选择极性和反相	触发源ITIx，滤波和预分频不可用 触发源CIx，配置CHxCAPFLT设置滤波，分频不可用 触发源是ETIF，滤波

	模式选择	触发源选择	极性选择	滤波和预分频
		111: ETIFPP		和预分频不可用
例1	复位模式 当触发输入上升沿, 计数器清零重启	TRGIS[2:0]=3'b000 选择ITIO为触发源	触发源是ITIO, 极性选 择不可用	触发源是 ITIO, 滤波 和预分频不可用
图 16-22. 复位模式下的控制电路				
				
例2	暂停模式 当触发输入为低的 时候, 计数器暂停计 数	TRGIS[2:0]=3'b101 选择CI0FE0为触发源	TIOS=0. (非异或) [CH0NP==0, CH0P==0]不反相. 在 上升沿捕获	在这个例子中滤波被 旁路
图 16-23. 暂停模式下的控制电路				
				
例3	事件模式 触发输入的上升沿 计数器开始计数	TRGIS[2:0]=3'b111 选择ETIF为触发源.	ETP = 0 没有极性改变	ETPSC = 1, 2分频. ETFC = 0, 无滤波

	模式选择	触发源选择	极性选择	滤波和预分频
图 16-24. 事件模式下的控制电路				



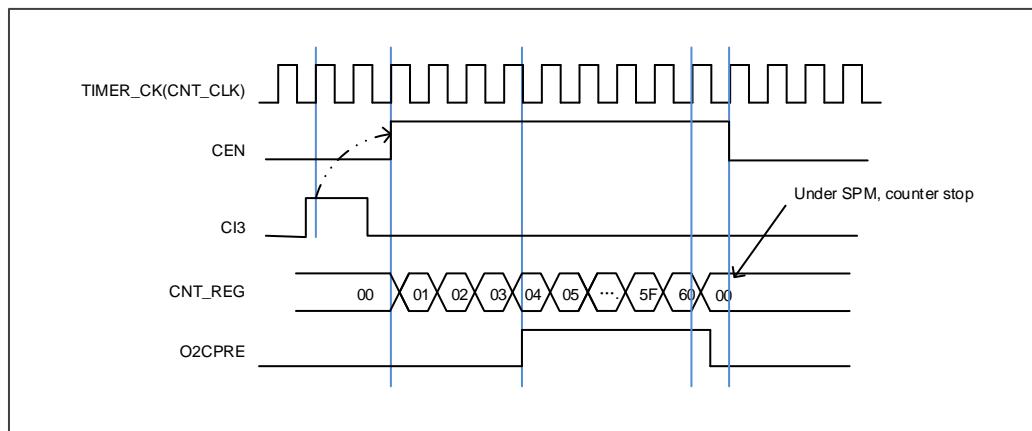
单脉冲模式

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

一旦设置定时器运行在单脉冲模式下，没有必要设置TIMERx_CTL0寄存器的定时器使能位CEN=1来使能计数器。触发信号沿或者软件写CEN=1都可以产生一个脉冲，此后CEN位一直保持为1直到更新事件发生或者CEN位被软件写0。如果CEN位被软件清0，计数器停止工作，计数值被保持。如果CEN值被硬件更新事件自动清0，计数器将被再次初始化。

在单脉冲模式下，有效的外部触发边沿会将CEN位置1，使能计数器。然而，执行计数值和TIMERx_CHxCV寄存器值的比较结果依然存在一些时钟延迟。为了最大限度减少延迟，用户可以将TIMERx_CHCTL0/1寄存器的CHxCOMFEN位置1。单脉冲模式下，触发上升沿产生之后，OxCPRE信号将被立即强制转换为与发生比较匹配时相同的电平，但是不用考虑比较结果。只有输出通道配置为PWM0或PWM1输出运行模式下时CHxCOMFEN位才可用，触发源来源于触发信号

图 16-25. 单脉冲模式，TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60

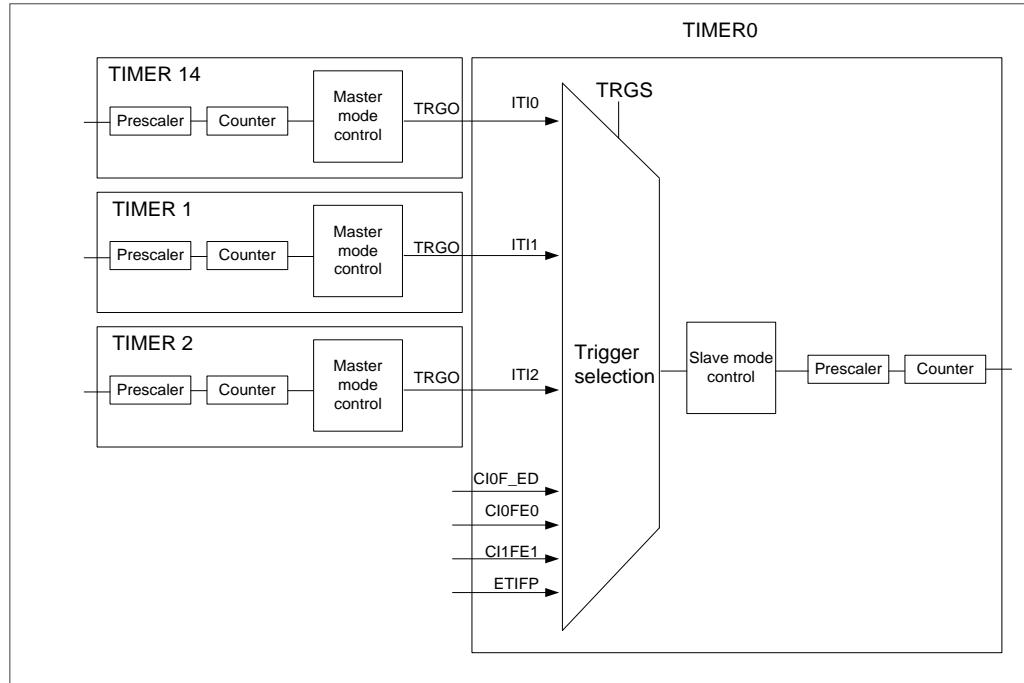


定时器互连

定时器之间相互连接可以定时器级联或者同步。可以通过配置一个定时器工作在主模式另一个定时器工作在从模式来实现。[图 16-26. 定时器 0 主/从模式的例子](#)显示了一些主从模式触发选择的例子。

[图 16-26. 定时器 0 主/从模式的例子](#) 显示了当定时器 0 配置为从模式时的触发选择

图 16-26. 定时器 0 主/从模式的例子



其他定时器互连的例子：

- 定时器1作为定时器0的预分频器

参考[图 16-26. 定时器 0 主/从模式的例子](#)连接配置定时器 1 为定时器 0 的预分频器，步骤如下：

1. 配置定时器1为主模式，选择其更新事件(UPE)为触发输出(配置TIMER1_CTL1寄存器的MMC=3'b010)。定时器2在每次计数器溢出时，输出一个周期信号；
2. 配置定时器2周期(TIMER1_CAR寄存器)；
3. 选择定时器0输入触发源为定时器1 (配置TIMERx_SMCFG寄存器的TRGS=3'b001)；
4. 配置定时器0在外部时钟模式1 (配置TIMERx_SMCFG寄存器的SMC=3'b111)；
5. 写1到CEN位启动定时器0 (TIMER0_CTL0寄存器)；
6. 写1到CEN位启动定时器1 (TIMER1_CTL0寄存器)。

- 用定时器1的使能/更新信号来启动定时器0

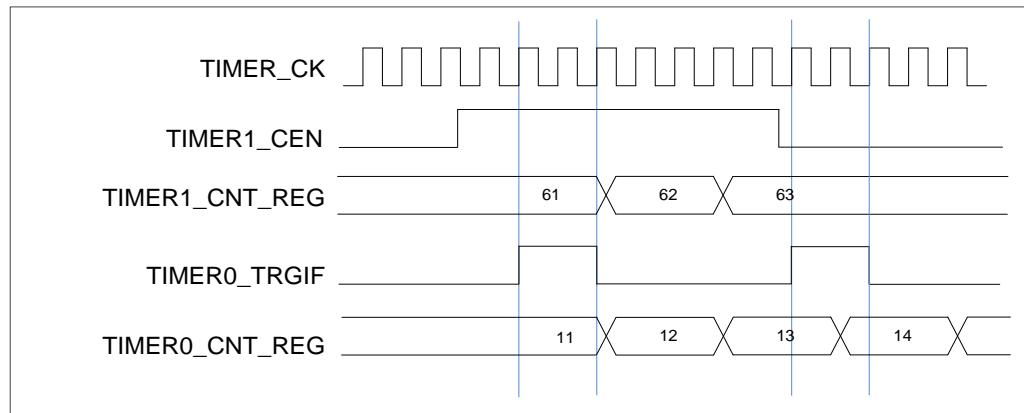
用定时器 1 的使能信号来启动定时器 0，见[图 16-27. 用定时器 1 的使能信号触发定时器 0](#)。在定时器 1 使能信号输出后，定时器 0 按照分频后的内部时钟从当前值开始计数。

当定时器0接收到触发信号，它的CEN位被自动置1，计数器计数直到禁能定时器0。两个定时

器的计数器频率都是TIMER_CK经过预分频器3分频后频率($f_{CNT_CLK} = f_{TIMER_CK}/3$)。步骤如下：

1. 配置定时器1为主模式，发送它的使能信号作为触发输出(配置TIMER1_CTL1寄存器的MMC=3'b001);
2. 配置定时器0选择输入触发来自定时器1 (配置TIMERx_SMCFG寄存器的TRGS=3'b001);
3. 配置定时器0在事件模式 (配置TIMERx_SMCFG寄存器的SMC=3'b 110);
4. 写1到CEN来开启定时器1 (TIMER1_CTL0寄存器)。

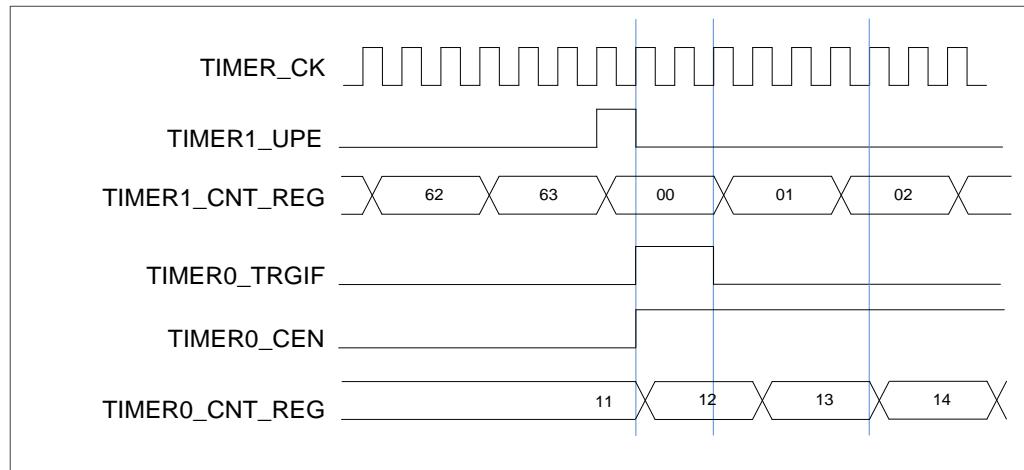
图 16-27. 用定时器 1 的使能信号触发定时器 0



在这个例子中，我们也可以使用更新事件代替使能信号作为触发源。见[图 16-28. 用定时器 1 的更新事件来触发定时器 0](#)，按以下步骤进行：

1. 配置定时器1为主模式，发送它的更新事件 (UPE) 作为触发输出(配置TIMER1_CTL1寄存器的MMC=3'b010)
2. 配置定时器1的周期 (TIMER1_CARL寄存器);
3. 配置定时器0选择输入触发来自定时器1 (配置TIMERx_SMCFG寄存器的TRGS=3'b001);
4. 配置定时器0在事件模式 (配置TIMERx_SMCFG寄存器的SMC=3'b 110);
5. 写1到CEN来开启定时器1 (TIMER1_CTL0寄存器)。

图 16-28. 用定时器 1 的更新事件来触发定时器 0

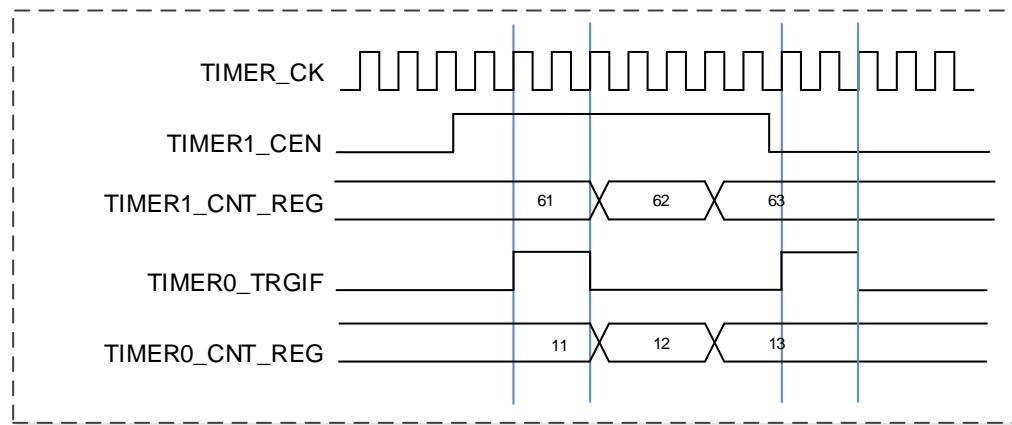


- 使用定时器1的使能/O0CPRE参考信号来使能定时器0计数。

在这个例子中，我们使用定时器1的使能输出来控制定时器0的使能。如[图16-29. 用定时器1的使能来选通定时器0](#)，在定时器1被使能后，定时器0在内部分频的时钟上开始计数。两个计数器的时钟频率都是由TIMER_CK时钟3分频得来($f_{CNT_CLK} = f_{TIMER_CK}/3$)，步骤如下：

1. 配置定时器1在主模式，配置其输出使能信号作为触发输出(配置TIMER1_CTL1寄存器的MMC=3'b001);
2. 配置定时器0从定时器1获取输入触发(配置TIMERx_SMCFG寄存器的TRGS=3'b001);
3. 配置定时器0工作在暂停模式(配置TIMERx_SMCFG寄存器的SMC=3'b101);
4. 写1到CEN位来使能定时器0 (TIMER0_CTL0寄存器);
5. 写1到CEN位来启动定时器1 (TIMER1_CTL0寄存器);
6. 写0到CEN位来停止定时器1 (TIMER1_CTL0寄存器)。

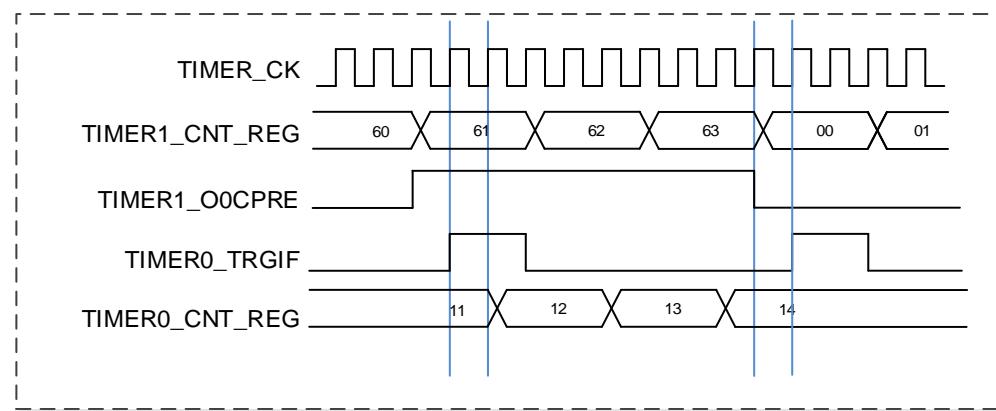
图 16-29. 用定时器 1 的使能来选通定时器 0



这个例子中，我们也可以使用定时器1的O0CPRE信号代替其使能信号输出作为触发源。步骤如下：

1. 配置定时器1在主模式下，配置O0CPRE信号为触发输出(配置TIMER1_CTL1寄存器的MMS=3'b100);
2. 配置定时器1的O0CPRE波形(TIMER1_CH0CTL寄存器);
3. 配置定时器0获取来自定时器1的输入触发(配置TIMERx_SMCFG寄存器TRGS=3'b010);
4. 配置定时器0工作在暂停模式(配置TIMERx_SMCFG寄存器的SMC=3'b101);
5. 写1到CEN位来使能定时器0 (TIMER0_CTL0寄存器);
6. 写1到CEN位来开启定时器1 (TIMER1_CTL0寄存器)。

图 16-30. 用定时器 1 的 O0CPRE 信号选通定时器 1



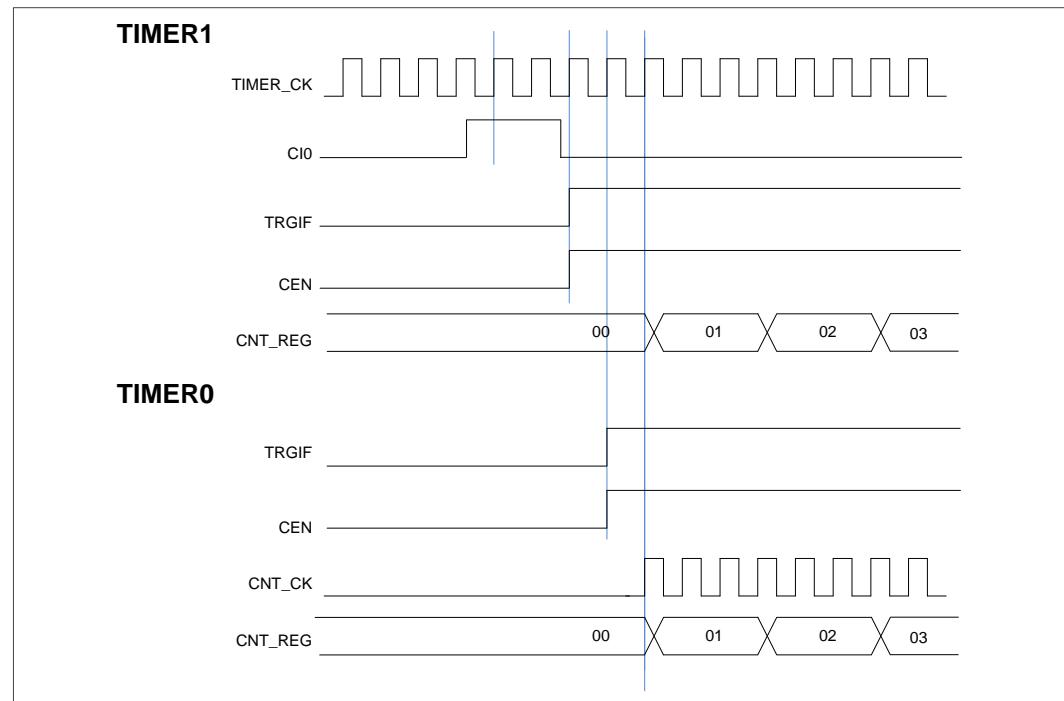
■ 使用一个外部触发来同步两个定时器

配置定时器1的使能信号触发定时器0的开启，配置定时器1的Cl0输入信号上升沿来触发定时器1。为了确保两个定时器同步开启，定时器1必须配置在主/从模式。步骤如下：

1. 配置定时器1工作在从模式来获取来自Cl0的触发输入(配置TIMER1_SMCFG寄存器的TRGS=3'b100);
2. 配置定时器1工作在事件模式(配置TIMER1_SMCFG寄存器的SMC=3'b110);
3. 写MSM=1(TIMER1_SMCFG寄存器)来配置定时器1工作在主/从模式;
4. 配置定时器0的触发输入来自定时器1 (配置TIMERx_SMCFG 寄存器的TRGS=3'b001);
5. 配置定时器0工作在事件模式(配置TIMER0_SMCFG寄存器的SMC=3'b110)。

当定时器1的Cl0信号产生上升沿时，两个定时器的计数器在内部时钟下开始同步计数，二者的TRGIF标志位都被置1。

图 16-31. 用定时器 1 的 Cl0 输入来触发定时器 0 和定时器 1



定时器 DMA 模式

定时器DMA模式是指通过DMA模块配置定时器的寄存器。有两个跟定时器DMA模式相关的寄存器：TIMERx_DMACFG和TIMERx_DMATB。当然，必须要使能DMA请求，一些内部中断事件可以产生DMA请求。当中断事件发生，TIMERx会给DMA发送请求。DMA配置成M2P模式，PADDR是TIMERx_DMATB寄存器地址，DMA就会访问TIMERx_DMATB寄存器。实际上，TIMERx_DMATB寄存器只是一个缓冲，定时器会将TIMERx_DMATB映射到一个内部寄存器，这个内部寄存器由TIMERx_DMACFG寄存器中的DMATA来指定。如果TIMERx_DMACFG寄存器的DMATC位域值为0，表示1次传输，定时器的发送1个DMA请求就可以完成。如果TIMERx_DMACFG寄存器的DMATC位域值不为1，例如其值为3，表示4次传输，定时器就需要再多发3次DMA请求。在这3次请求下，DMA对TIMERx_DMATB寄存器的访问会映射到访问定时器的DMATA+0x4, DMATA+0x8, DMATA+0xc寄存器。总之，发生一次DMA内部中断请求，定时器会连续发送(DMATC+1)次请求。

如果再来1次DMA请求事件，TIMERx将会重复上面的过程。

定时器调试模式

当Cortex™-M4内核停止，DBG_CTL0寄存器中的TIMERx_HOLD配置位被置1，定时器计数器停止。

16.1.5. TIMERx 寄存器(x=0)

控制寄存器 0 (TIMERx_CTL0)

地址偏移: 0x00

复位值: 0x0000

该寄存器可以按半字(16位)或字(32位)访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					CKDIV[1:0]	ARSE	CAM[1:0]	DIR	SPM	UPS	UPDIS	CEN			

rw rw rw rw rw rw rw rw

位/位域	名称	描述
15:10	保留	必须保持复位值
9:8	CKDIV[1:0]	时钟分频 通过软件配置CKDIV，规定定时器时钟(TIMER_CK)与死区时间和采样时钟(DTS)之间的分频系数，死区发生器和数字滤波器会用到DTS时间。 00: f _{DTS} =f _{TIMER_CK} 01: f _{DTS} =f _{TIMER_CK} /2 10: f _{DTS} =f _{TIMER_CK} /4 11: 保留
7	ARSE	自动重载影子使能 0: 禁能TIMERx_CAR寄存器的影子寄存器

1: 使能TIMERx_CAR寄存器的影子寄存器

6:5	CAM[1:0]	计数器对齐模式选择 00: 无中央对齐模式(边沿对齐模式). DIR位指定了计数方向 01: 中央对齐向下计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式（TIMERx_CHCTL0寄存器中CHxMS=00），只有在向下计数时，通道的比较中断标志置1 10: 中央对齐向上计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式（TIMERx_CHCTL0寄存器中CHxMS=00），只有在向上计数时，通道的比较中断标志置1 11: 中央对齐上下计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式（TIMERx_CHCTL0寄存器中CHxMS=00），在向上和向下计数时，通道的比较中断标志都会置1 当计数器使能以后，改为不能从0x00切换到非0x00
4	DIR	方向 0: 向上计数 1: 向下计数 当计数器配置为中央对齐模式或编码器模式时，该位为只读
3	SPM	单脉冲模式 0: 更新事件发生后，计数器继续计数 1: 在下一次更新事件发生时，CEN硬件清零并且计数器停止计数
2	UPS	更新请求源 软件配置该为，选择更新事件源。 0: 使能后，下述任一事件产生更新中断或DMA请求： – UPG位被置1 – 计数器溢出/下溢 – 从模式控制器产生的更新 1: 使能后只有计数器溢出/ 下溢才产生更新中断或DMA请求
1	UPDIS	禁止更新。 该位用来使能或禁能更新事件的产生。 0: 更新事件使能.当以下事件之一发生时，更新事件产生，具有缓存的寄存器被装入它们的预装载值： – UPG位被置1 – 计数器溢出/下溢 – 从模式控制器产生一个更新事件 1: 更新事件禁能. 带有缓存的寄存器保持原有值，如果UPG位被置1或者从模式控制器产生一个硬件复位事件，计数器和预分频器被重新初始化
0	CEN	计数器使能 0: 计数器禁能 1: 计数器使能 在软件将CEN位置1后，外部时钟、暂停模式和编码器模式才能工作。触发模式可以

自动地通过硬件设置CEN位。

控制寄存器 1 (TIMERx_CTL1)

地址偏移: 0x04

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留.	ISO3	ISO2N	ISO2	ISO1N	ISO1	ISOON	ISO0	TIOS		MMC[2:0]	DMAS	CCUC	保留.	CCSE	

rw rw

位/位域	名称	描述
15	保留	必须保持复位值
14	ISO3	通道3的空闲状态输出 参考ISO0位
13	ISO2N	通道2的互补通道空闲状态输出 参考ISOON位
12	ISO2	通道2的空闲状态输出 参考ISO0位
11	ISO1N	通道1的互补通道空闲状态输出 参考ISOON位
10	ISO1	通道1的空闲状态输出 参考ISO0位
9	ISOON	通道0的互补通道空闲状态输出 0: 当POEN复位, CH0_ON设置低电平. 1: 当POEN复位, CH0_ON设置高电平 此位只有在TIMERx_CCHP寄存器的PROT [1:0]位为00的时候可以被更改.
8	ISO0	通道0的空闲状态输出 0: 当POEN复位, CH0_O设置低电平 1: 当POEN复位, CH0_O设置高电平 如果CH0_ON生效, 一个死区时间后CH0_O输出改变。此位只有在TIMERx_CCHP 寄存器的PROT [1:0]位为00的时候可以被更改.
7	TIOS	通道0触发输入选择 0: 选择TIMERx_CH0 引脚作为通道0的触发输入 1: 选择TIMERx_CH0, CH1和CH2 引脚异或的结果作为通道0的触发输入
6:4	MMC[2:0]	主模式控制 这些位控制TRGO信号的选择, TRGO信号由主定时器发给从定时器用于同步功能 000: 复位。TIMERx_SWEVG寄存器的UPG位被置1或从模式控制器产生复位触发

一次TRGO脉冲，后一种情况下，TRGO上的信号相对实际的复位会有一个延迟。

001：使能。此模式可用于同时启动多个定时器或控制在一段时间内使能从定时器。主模式控制器选择计数器使能信号作为触发输出TRGO。当CEN控制位被置1或者暂停模式下触发输入为高电平时，计数器使能信号被置1。在暂停模式下，计数器使能信

号受控于触发输入，在触发输入和TRGO 上会有一个延迟，除非选择了主/从模式。

010：更新。主模式控制器选择更新事件作为TRGO。

011：捕获/比较脉冲.通道0在发生一次捕获或一次比较成功时，主模式控制器产生一个TRGO脉冲

100：比较。在这种模式下主模式控制器选择O0CPRE信号被用于作为触发输出TRGO

101：比较。在这种模式下主模式控制器选择O1CPRE信号被用于作为触发输出TRGO

110：比较。在这种模式下主模式控制器选择O2CPRE信号被用于作为触发输出TRGO

111：比较。在这种模式下主模式控制器选择O3CPRE信号被用于作为触发输出

TRGO

3 DMAS DMA请求源选择

0：当通道捕获/比较事件发生时，发送通道x的DMA请求。

1：当更新事件发生，发送通道x的DMA请求

2 CCUC 换相控制影子寄存器更新控制

当换相控制影子寄存器 (CHxEN, CHxNEN和CHxCOMCTL位) 使能(CCSE=1)，这些影子寄存器更新控制如下：

0：CMTG位被置1时更新影子寄存器

1：当CMTG位被置1或检测到TRIGI上升沿时，影子寄存器更新
当通道没有互补输出时，此位无效。

1 保留 必须保持复位值。

0 CCSE 换相控制影子使能

0：影子寄存器CHxEN, CHxNEN和CHxCOMCTL位禁能。

1：影子寄存器CHxEN, CHxNEN和CHxCOMCTL位使能。

如果这些位已经被写入了，换相事件到来时这些位才被更新

当通道没有互补输出时，此位无效

从模式配置寄存器 (TIMERx_SMCFG)

地址偏移：0x08

复位值：0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	SMC1	ETPSC[1:0]		ETFC[3:0]	MSM		TRGS[2:0]	OCRC		SMC[2:0]					

位/位域	名称	描述
15	ETP	<p>外部触发极性</p> <p>该位指定ETI信号的极性</p> <p>0: ETI高电平或上升沿有效 .</p> <p>1: ETI低电平或下降沿有效 .</p>
14	SMC1	<p>SMC的一部分为了使能外部时钟模式1</p> <p>在外部时钟模式1, 计数器由ETIF信号上的任意有效边沿驱动</p> <p>0: 外部时钟模式1禁能</p> <p>1: 外部时钟模式1使能</p> <p>复位模式, 暂停模式和事件模式可以与外部时钟模式1同时使用。但是TRGS必须不能为3'b111。</p> <p>如果外部时钟模式0和外部时钟模式1同时被使能, 外部时钟的输入是ETIF</p> <p>注意: 外部时钟模式0使能在寄存器的SMC位域。</p>
13:12	ETPSC[1:0]	<p>外部触发预分频</p> <p>外部触发信号ETI的频率不能超过TIMER_CK频率的1/4。当输入较快的外部时钟时, 可以使用预分频降低ETIP的频率。</p> <p>00: 预分频禁能</p> <p>01: ETI 频率被2分频</p> <p>10: ETI 频率被4分频</p> <p>11: ETI 频率被8分频</p>
11:8	ETFC[3:0]	<p>外部触发滤波控制</p> <p>数字滤波器是一个事件计数器, 它记录到N个事件后会产生一个输出的跳变。这些位定义了对ETI信号采样的频率和对ETI数字滤波的带宽。</p> <p>0000: 滤波器禁能 $f_{SAMP} = f_{DTS}$, $N=1$.</p> <p>0001: $f_{SAMP} = f_{TIMER_CK}$, $N=2$.</p> <p>0010: $f_{SAMP} = f_{TIMER_CK}$, $N=4$.</p> <p>0011: $f_{SAMP} = f_{TIMER_CK}$, $N=8$.</p> <p>0100: $f_{SAMP} = f_{DTS}/2$, $N=6$.</p> <p>0101: $f_{SAMP} = f_{DTS}/2$, $N=8$.</p> <p>0110: $f_{SAMP} = f_{DTS}/4$, $N=6$.</p> <p>0111: $f_{SAMP} = f_{DTS}/4$, $N=8$.</p> <p>1000: $f_{SAMP} = f_{DTS}/8$, $N=6$.</p> <p>1001: $f_{SAMP} = f_{DTS}/8$, $N=8$.</p> <p>1010: $f_{SAMP} = f_{DTS}/16$, $N=5$.</p> <p>1011: $f_{SAMP} = f_{DTS}/16$, $N=6$.</p> <p>1100: $f_{SAMP} = f_{DTS}/16$, $N=8$.</p> <p>1101: $f_{SAMP} = f_{DTS}/32$, $N=5$.</p> <p>1110: $f_{SAMP} = f_{DTS}/32$, $N=6$.</p> <p>1111: $f_{SAMP} = f_{DTS}/32$, $N=8$.</p>
7	MSM	主-从模式

该位被用来同步被选择的定时器同时开始计数。通过TRIGI和TRGO，定时器被连接在一起，TRGO用做启动事件。

0: 主从模式禁能

1: 主从模式使能

6:4	TRGS[2:0]	触发选择 该位域用来指定选择哪一个信号作为用来同步计数器的触发输入源 000: 内部触发输入0 (ITIO) 001: 内部触发输入1 (ITI1) 010: 内部触发输入2 (ITI2) 011: 保留 100: CI0的边沿标志位 (CI0F_ED) 101: 滤波后的通道0输入 (CI0FE0) 110: 滤波后的通道1输入(CI1FE1) 111: 滤波后的外部触发输入(ETIFP) 从模式被使能后这些位不能改
3	OCRC	OCPRE清除源选择 0: OCPRE_CLR_INT连接到OCPRE_CLR输入 1: OCPRE_CLR_INT连接到ETIF
2:0	SMC[2:0]	从模式控制 000: 关闭从模式. 如果CEN=1, 则预分频器直接由内部时钟驱动 001: 编码器模式0. 根据CI0FE0的电平, 计数器在CI1FE1的边沿向上/下计数 010: 编码器模式1. 根据CI1FE1的电平, 计数器在CI0FE0的边沿向上/下计数 011: 编码器模式2. 根据另一个信号的输入电平, 计数器在CI0FE0和CI1FE1的边沿向上/下计数 100: 复位模式. 选中的触发输入的上升沿重新初始化计数器, 并且更新影子寄存器. 101: 暂停模式. 当触发输入为高时, 计数器的时钟开启. 一旦触发输入变为低, 则计数器停止 110: 事件模式. 计数器在触发输入的上升沿启动. 计数器不能被从模式控制器关闭. 111: 外部时钟模式0. 选中的触发输入的上升沿驱动计数器 由于CI0F_ED是一个脉冲波形, 而暂停模式是检测触发信号的电平, 所以, 当 CI0F_ED用作触发输入时, 暂停模式必须禁能。

DMA 和中断使能寄存器 (TIMERx_DMAINTEN)

地址偏移: 0x0C

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TRGDEN	CMTDEN	CH3DEN	CH2DEN	CH1DEN	CH0DEN	UPDEN	BRKIE	TRGIE	CMTIE	CH3IE	CH2IE	CH1IE	CHOIE	UPIE

位/位域	名称	描述
15	保留	必须保持复位值.
14	TRGDEN	触发DMA请求使能 0: 禁止触发DMA请求 1: 使能触发DMA请求
13	CMTDEN	换相DMA更新请求使能 0: 禁止换相DMA更新请求 1: 使能换相DMA更新请求
12	CH3DEN	通道3比较/捕获 DMA请求使能 0: 禁止通道3比较/捕获DMA请求 1: 使能通道3比较/捕获DMA请求
11	CH2DEN	通道2比较/捕获 DMA请求使能 0: 禁止通道2比较/捕获DMA请求 1: 使能通道2比较/捕获DMA请求
10	CH1DEN	通道1比较/捕获 DMA请求使能 0: 禁止通道1比较/捕获DMA请求 1: 使能通道1比较/捕获DMA请求
9	CH0DEN	通道0比较/捕获 DMA请求使能 0: 禁止通道0比较/捕获DMA请求 1: 使能通道0比较/捕获DMA请求
8	UPDEN	更新DMA请求使能 0: 禁止更新DMA请求 1: 使能更新DMA请求
7	BRKIE	中止中断使能 0: 禁止中止中断 1: 使能中止中断
6	TRGIE	触发中断使能 0: 禁止触发中断 1: 使能触发中断
5	CMTIE	换相更新中断使能 0: 禁止换相更新中断 1: 使能换相更新中断
4	CH3IE	通道3比较/捕获中断使能 0: 禁止通道3中断 1: 使能通道3中断
3	CH2IE	通道2比较/捕获中断使能 0: 禁止通道2中断

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

中断标志寄存器 (TIMERx_INTF)

地址偏移: 0x10

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	CH3OF	CH2OF	CH1OF	CH0OF	保留	保留	BRKIF	TRGIF	CMTIF	CH3IF	CH2IF	CH1IF	CHOIF	UPIF	

rc_w0 rc_w0 rc_w0 rc_w0 . rc_w0 rc_w0 rc_w0 rc_w0 rc_w0 rc_w0 rc_w0 rc_w0 rc_w0 rc_w0

位/位域	名称	描述
15:13	保留	必须保持复位值.
12	CH3OF	通道3捕获溢出标志 参见CH0OF描述
11	CH2OF	通道2捕获溢出标志 参见CH0OF描述
10	CH1OF	通道1捕获溢出标志 参见CH0OF描述
9	CH0OF	通道0捕获溢出标志 当通道0被配置为输入模式时, 在CHOIF标志位已经被置1后, 捕获事件再次发生时, 该标志位可以由硬件置1。该标志位由软件清0. 0: 无捕获溢出中断发生 1: 发生了捕获溢出中断
8	保留	必须保持复位值.
7	BRKIF	中止中断标志位 一旦中止输入有效, 由硬件对该位置'1'。如果中止输入无效, 则该位可由软件清'0'。 0: 无中止事件产生 1: 中止输入上检测到有效电平

6	TRGIF	触发中断标志 当发生触发事件时，此标志由硬件置1。此位由软件清0。当从模式控制器处于除暂停模式外的其它模式时，在触发输入端检测到有效边沿，产生触发事件。当从模式控制器处于暂停模式时，触发输入的任意边沿都可以产生触发事件。 0: 无触发事件产生 1: 触发中断产生
5	CMTIF	通道换相更新中断标志 当通道换相更新事件发生时此标志位被硬件置1，此位由软件清0。 0: 无通道换相更新中断发生 1: 通道换相更新中断发生
4	CH3IF	通道3比较/捕获中断标志 参见CH0IF描述
3	CH2IF	通道2比较/捕获中断标志 参见CH0IF描述
2	CH1IF	通道1比较/捕获中断标志 参见CH0IF描述
1	CH0IF	通道0比较/捕获中断标志 此标志由硬件置1软件清0。当通道0在输入模式下时，捕获事件发生时此标志位被置1；当通道0在输出模式下时，此标志位在一个比较事件发生时被置1。 0: 无通道0中断发生 1: 通道0中断发生
0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置1，软件清0。 0: 无更新中断发生 1: 发生更新中断

软件事件产生寄存器 (**TIMERx_SWEVG**)

地址偏移: 0x14

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								BRKG	TRGG	CMTG	CH3G	CH2G	CH1G	CH0G	UPG
								w	w	w	w	w	w	w	w

位/位域	名称	描述
15:8	保留	必须保持复位值。
7	BRKG	产生中止事件 该位由软件置1，用于产生一个中止事件，由硬件自动清0。当此位被置1时，POEN

		位被清0且BRKIF位被置1，若开启对应的中断和DMA，则产生相应的中断和DMA传输。
		0: 不产生中止事件 1: 产生中止事件
6	TRGG	触发事件产生 此位由软件置1，由硬件自动清0。当此位被置1，TIMERx_INTF寄存器的TRGIF标志位被置1，若开启对应的中断和DMA，则产生相应的中断和DMA传输。 0: 无触发事件产生 1: 产生触发事件
5	CMTG	通道换相更新事件发生 此位由软件置1，由硬件自动清0。当此位被置1，通道捕获/比较控制寄存器(CHxEN, CHxNEN 和CHxCOMCTL) 的互补输出被更新。 0: 不产生通道控制更新事件 1: 产生通道控制更新事件
4	CH3G	通道3捕获或比较事件发生 参见CH0G描述
3	CH2G	通道2捕获或比较事件发生 参见CH0G描述
2	CH1G	通道1捕获或比较事件发生 参见CH0G描述
1	CH0G	通道0捕获或比较事件发生 该位由软件置1，用于在通道0产生一个捕获/比较事件，由硬件自动清0。当此位被置1，CH0IF标志位被置1，若开启对应的中断和DMA，则发出相应的中断和DMA请求。此外，如果通道0配置为输入模式，计数器的当前值被TIMERx_CH0CV寄存器捕获，如果CH0IF标志位已经为1，则CH0OF标志位被置1。 0: 不产生通道0捕获或比较事件 1: 发生通道0捕获或比较事件
0	UPG	更新事件产生 此位由软件置1，被硬件自动清0。当此位被置1，如果选择了中央对齐或向上计数模式，计数器被清0。否则(向下计数模式)计数器将载入自动重载值，预分频计数器将同时被清除。 0: 无更新事件产生 1: 产生更新事件

通道控制寄存器 0 (TIMERx_CHCTL0)

地址偏移: 0x18

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1COM CEN	CH1COMCTL[2:0]	CH1COM SEN	CH1COM FEN	CH1MS[1:0]	CH0COM CEN	CH0COMCTL[2:0]	CH0COM SEN	CH0COM FEN	CH0MS[1:0]						
CH1CAPFLT[3:0]	CH1CAPPSC[1:0]				CH0CAPFLT[3:0]	CH0CAPPSC[1:0]									

rw

rw

rw

rw

rw

rw

输出比较模式:

位/位域	名称	描述
15	CH1COMCEN	通道1输出比较清0使能 参见CH0COMCEN描述
14:12	CH1COMCTL[2:0]	通道1输出比较模式 参见CH0COMCTL描述
11	CH1COMSEN	通道1输出比较影子寄存器使能 参见CH0COMSEN描述
10	CH1COMFEN	通道1输出比较快速使能 参见CH0COMFEN描述
9:8	CH1MS[1:0]	通道1模式选择 这些位定义了通道的方向和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2寄存器的CH1EN位被清0)时这些位才可以写。 00: 通道1配置为输出 01: 通道1配置为输入, IS1映射在CI1FE1上 10: 通道1配置为输入, IS1映射在CI0FE1上 11: 通道1配置为输入, IS1映射在ITS上, 此模式仅工作在内部触发器输入被选中时(由 TIMERx_SMCFGFG寄存器的TRGS位选择)。
7	CH0COMCEN	通道0输出比较清0使能 当此位被置1, 当检测到ETIF输入高电平时, O0CPRE参考信号被清0 0: 禁止通道0输出比较清零 1: 使能通道0输出比较清零
6:4	CH0COMCTL[2:0]	通道0输出比较模式 此位定义了输出参考信号O0CPRE的动作, 而O0CPRE决定了CH0_O、CH0_ON的值。O0CPRE高电平有效, 而CH0_O、CH0_ON的有效电平取决于CH0P、CH0NP位。 000: 冻结。输出比较寄存器TIMERx_CH0CV与计数器TIMERx_CNT间的比较对O0CPRE不起作用 001: 匹配时设置为高。当计数器的值与捕获/比较值寄存器TIMERx_CH0CV相同时, 强制O0CPRE为高。 010: 匹配时设置为低。当计数器的值与捕获/比较值寄存器TIMERx_CH0CV相同时, 强制O0CPRE为低。 011: 匹配时翻转。当计数器的值与捕获/比较值寄存器TIMERx_CH0CV相同时, 强制O0CPRE翻转。 100: 强制为低。强制O0CPRE为低电平

101: 强制为高。强制O0CPRE为高电平
 110: PWM 模式0 (EAPWM)。在向上计数时，一旦计数器值小于 TIMERx_CH0CV时，O0CPRE为高电平，否则为低电平。在向下计数时，一旦计数器的值大于TIMERx_CH0CV时，O0CPRE 为低电平，否则为高电平。
 111: PWM 模式1(CAPWM)。在向上计数时，一旦计数器值小于 TIMERx_CH0CV时，O0CPRE为低电平，否则为高电平。在向下计数时，一旦计数器的值大于TIMERx_CH0CV时，O0CPRE为高电平，否则为低电平。
 在PWM模式0或PWM模式1中，只有当比较结果改变了或者输出比较模式中从冻结模式切换到PWM模式时，O0CPRE电平才改变。
 当TIMERx_CCHP寄存器的PROT [1:0]=11且CH0MS =00（比较模式）时此位不能被改变。

3	CH0COMSEN	<p>通道0输出比较影子寄存器使能 当此位被置1，TIMERx_CH0CV寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0: 禁止通道0输出/比较影子寄存器 1: 使能通道0输出/比较影子寄存器 仅在单脉冲模式下(TIMERx_CTL0寄存器的SPM =1)，可以在未确认预装载寄存器情况下使用PWM 模式 当TIMERx_CCHP寄存器的PROT [1:0]=11且CH0MS =00时此位不能被改变。</p>
2	CH0COMFEN	<p>通道0输出比较快速使能 当该位为1时，如果通道配置为PWM0模式或者PWM1模式，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH0_O被设置为比较电平而与比较结果无关。 0: 禁止通道0输出比较快速。当触发器的输入有一个有效沿时，激活CH0_O输出的最小延时为5个时钟周期 1: 使能通道0输出比较快速。当触发器的输入有一个有效沿时，激活CH0_O输出的最小延时为3个时钟周期</p>
1:0	CH0MS[1:0]	<p>通道0 I/O模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭 (TIMERx_CHCTL2寄存器的CH0EN位被清0)时这些位才可写。 00: 通道0配置为输出 01: 通道0配置为输入，IS0映射在CI0FE0上 10: 通道0配置为输入，IS0映射在CI1FE0上 11: 通道0配置为输入，IS0映射在ITS上。此模式仅工作在内部触发输入被选中时(通过设置TIMERx_SMCFGFG寄存器的TRGS位)</p>

输入捕获模式:

位/位域	名称	描述
15:12	CH1CAPFLT[3:0]	通道1输入捕获滤波控制 参见CH0CAPFLT描述

11:10	CH1CAPPSC[1:0]	通道1输入捕获预分频器 参见CH0CAPPSC描述
9:8	CH1MS[1:0]	通道1模式选择 与输出模式相同
7:4	CH0CAPFLT[3:0]	通道0输入捕获滤波控制 数字滤波器由一个事件计数器组成，它记录N个输入事件后会产生一个输出的跳变。这些位定义了CI0输入信号的采样频率和数字滤波器的长度。 0000: 无滤波器, $f_{SAMP} = f_{DTS}$, N=1 0001: $f_{SAMP} = f_{PCLK}$, N=2 0010: $f_{SAMP} = f_{PCLK}$, N=4 0011: $f_{SAMP} = f_{PCLK}$, N=8 0100: $f_{SAMP} = f_{DTS}/2$, N=6 0101: $f_{SAMP} = f_{DTS}/2$, N=8 0110: $f_{SAMP} = f_{DTS}/4$, N=6 0111: $f_{SAMP} = f_{DTS}/4$, N=8 1000: $f_{SAMP} = f_{DTS}/8$, N=6 1001: $f_{SAMP} = f_{DTS}/8$, N=8 1010: $f_{SAMP} = f_{DTS}/16$, N=5 1011: $f_{SAMP} = f_{DTS}/16$, N=6 1100: $f_{SAMP} = f_{DTS}/16$, N=8 1101: $f_{SAMP} = f_{DTS}/32$, N=5 1110: $f_{SAMP} = f_{DTS}/32$, N=6 1111: $f_{SAMP} = f_{DTS}/32$, N=8
3:2	CH0CAPPSC[1:0]	通道0输入捕获预分频器 这2位定义了通道0输入的预分频系数。当TIMERx_CHCTL2寄存器中的CH0EN =0时，则预分频器复位。 00: 无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获 01: 每2个事件触发一次捕获 10: 每4个事件触发一次捕获 11: 每8个事件触发一次捕获
1:0	CH0MS[1:0]	通道0模式选择 与输出比较模式相同

通道控制寄存器 1 (TIMERx_CHCTL1)

地址偏移: 0x1C

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3COM CEN	CH3COMCTL[2:0]	CH3COM SEN	CH3COM FEN	CH3MS[1:0]	CH2COM CEN	CH2COMCTL[2:0]	CH2COM SEN	CH2COM FEN	CH2MS[1:0]						

CH3CAPFLT[3:0]	CH3CAPPSC[1:0]		CH2CAPFLT[3:0]	CH2CAPPSC[1:0]	
rw	rw	rw	rw	rw	rw

输出比较模式:

位/位域	名称	描述
15	CH3COMCEN	通道3输出比较清0使能 参见CH0COMCEN描述
14:12	CH3COMCTL[2:0]	通道3输出比较模式 参见CH0COMCTL描述
11	CH3COMSEN	通道3输出比较影子寄存器使能 参见CH0COMSEN描述
10	CH3COMFEN	通道3输出比较快速使能 参见CH0COMSEN描述
9:8	CH3MS[1:0]	通道3模式选择 这些位定义了通道的方向和输入信号的选择。只有当通道关闭 (TIMERx_CHCTL2寄存器的CH3EN位被清0)时这些位才可以写。 00: 通道3配置为输出 01: 通道3配置为输入, IS3映射在CI3FE3上 10: 通道3配置为输入, IS3映射在CI2FE3上 11: 通道3配置为输入, IS3映射在ITS上, 此模式仅工作在内部触发器输入被 选中时(由 TIMERx_SMCFGFG寄存器的TRGS位选择)。
7	CH2COMCEN	通道2输出比较清0使能 当此位被置1, 当检测到ETIF输入高电平时, O2CPRE参考信号被清0 0: 使能通道2输出比较清零 1: 禁止通道2输出比较清零
6:4	CH2COMCTL[2:0]	通道2输出比较模式 此位定义了输出参考信号O2CPRE的动作, 而O2CPRE决定了CH2_O、 CH2_ON的值。O2CPRE高电平有效, 而CH2_O、CH2_ON的有效电平取决于 CH2P、CH2NP位。 000: 冻结。输出比较寄存器TIMERx_CH2CV与计数器TIMERx_CNT间的比 较对O2CPRE不起作用 001: 匹配时设置为高。当计数器的值与捕获/比较值寄存器TIMERx_CH2CV 相同时, 强制O2CPRE为高。 010: 匹配时设置为低。当计数器的值与捕获/比较值寄存器TIMERx_CH2CV 相同时, 强制O2CPRE为低。 011: 匹配时翻转。当计数器的值与捕获/比较值寄存器TIMERx_CH2CV相同 时, 强制O2CPRE翻转。 100: 强制为低。强制O2CPRE为低电平 101: 强制为高。强制O2CPRE为高电平 110: PWM 模式0 (EAPWM)。在向上计数时, 一旦计数器值小于 TIMERx_CH2CV时, O2CPRE为高电平, 否则为低电平。在向下计数时, 一

一旦计数器的值大于TIMERx_CH2CV时，O2CPRE 为低电平，否则为高电平。

111: PWM 模式1(CAPWM)。在向上计数时，一旦计数器值小于

TIMERx_CH2CV时，O2CPRE为低电平，否则为高电平。在向下计数时，一

旦计数器的值大于TIMERx_CH2CV时，O2CPRE为高电平，否则为低电平。

在PWM模式0或PWM模式1中，只有当比较结果改变了或者输出比较模式中从冻结模式切换到PWM模式时，CxCOMR电平才改变。

当TIMERx_CCHP寄存器的PROT [1:0]=11且CH2MS =00（比较模式）时此位不能被改变。

3	CH2COMSEN	<p>通道0输出比较影子寄存器使能</p> <p>当此位被置1，TIMERx_CH2CV寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。</p> <p>0: 禁止通道2输出/比较影子寄存器</p> <p>1: 使能通道2输出/比较影子寄存器</p> <p>仅在单脉冲模式下(TIMERx_CTL0寄存器的SPM =1)，可以在未确认预装载寄存器情况下使用PWM 模式</p> <p>当TIMERx_CCHP寄存器的PROT [1:0]=11且CH2MS =00时此位不能被改变。</p>
2	CH2COMFEN	<p>通道2输出比较快速使能</p> <p>当该位为1时，如果通道配置为PWM0模式或者PWM1模式，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH2_O被设置为比较电平而与比较结果无关。</p> <p>0: 禁止通道2输出比较快速. 当触发器的输入有一个有效沿时，激活CH2_O输出的最小延时为5个时钟周期</p> <p>1: 使能通道2输出比较快速。当触发器的输入有一个有效沿时，激活CH2_O输出的最小延时为3个时钟周期</p>
1:0	CH2MS[1:0]	<p>通道2 I/O模式选择</p> <p>这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2寄存器的CH2EN位被清0)时这些位才可写。</p> <p>00: 通道2配置为输出</p> <p>01: 通道2配置为输入，IS2映射在CI2FE2上</p> <p>10: 通道2配置为输入，IS2映射在CI3FE2上</p> <p>11: 通道2配置为输入，IS2映射在ITS上. 此模式仅工作在内部触发输入被选中时(通过设置TIMERx_SMCFGFG寄存器的TRGS位)</p>

输入捕获模式:

位/位域	名称	描述
15:12	CH3CAPFLT[3:0]	通道3输入捕获滤波控制 参见CH0CAPFLT描述
11:10	CH3CAPPSC[1:0]	通道3输入捕获预分频器 参见CH0CAPPSC描述
9:8	CH3MS[1:0]	通道3模式选择

与输出模式相同

7:4	CH2CAPFLT[3:0]	通道2输入捕获滤波控制 数字滤波器由一个事件计数器组成，它记录N个输入事件后会产生一个输出的跳变。这些位定义了CI2输入信号的采样频率和数字滤波器的长度。 0000: 无滤波器, $f_{SAMP} = f_{DTS}$, N=1 0001: $f_{SAMP} = f_{PCLK}$, N=2 0010: $f_{SAMP} = f_{PCLK}$, N=4 0011: $f_{SAMP} = f_{PCLK}$, N=8 0100: $f_{SAMP} = f_{DTS}/2$, N=6 0101: $f_{SAMP} = f_{DTS}/2$, N=8 0110: $f_{SAMP} = f_{DTS}/4$, N=6 0111: $f_{SAMP} = f_{DTS}/4$, N=8 1000: $f_{SAMP} = f_{DTS}/8$, N=6 1001: $f_{SAMP} = f_{DTS}/8$, N=8 1010: $f_{SAMP} = f_{DTS}/16$, N=5 1011: $f_{SAMP} = f_{DTS}/16$, N=6 1100: $f_{SAMP} = f_{DTS}/16$, N=8 1101: $f_{SAMP} = f_{DTS}/32$, N=5 1110: $f_{SAMP} = f_{DTS}/32$, N=6 1111: $f_{SAMP} = f_{DTS}/32$, N=8
3:2	CH2CAPPSC[1:0]	通道2输入捕获预分频器 这2位定义了通道2输入的预分频系数。当TIMERx_CHCTL2寄存器中的CH2EN =0时，则预分频器复位。 00: 无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获 01: 每2个事件触发一次捕获 10: 每4个事件触发一次捕获 11: 每8个事件触发一次捕获
1:0	CH2MS[1:0]	通道2模式选择 与输出比较模式相同

通道控制寄存器 2 (TIMERx_CHCTL2)

地址偏移: 0x20

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留.	CH3P	CH3EN	CH2NP	CH2NEN	CH2P	CH2EN	CH1NP	CH1NEN	CH1P	CH1EN	CH0NP	CH0NEN	CH0P	CH0EN	

位/位域	名称	描述
15:14	保留	必须保持复位值
13	CH3P	通道3极性

参考CH0P描述		
12	CH3EN	通道3使能 参考 CH0EN 描述
11	CH2NP	通道2互补输出极性 参考 CH0NP 描述
10	CH2NEN	通道2互补输出使能 参考 CH0NEN 描述
9	CH2P	通道2极性 参考 CH0P 描述
8	CH2EN	通道2使能 参考 CH0EN 描述
7	CH1NP	通道1互补输出极性 参考 CH0NP 描述
6	CH1NEN	通道1互补输出使能 参考 CH0NEN 描述
5	CH1P	通道1极性 参考 CH0P 描述
4	CH1EN	通道1使能 参考 CH0EN 描述
3	CH0NP	通道0互补输出极性 当通道0配置为输出模式，此位定义了互补输出信号的极性。 0: 通道0高电平有效 1: 通道0低电平有效 当通道0配置为输入模式时，此位和 CH0P 联合使用，作为输入信号 CI0 的极性选择控制信号。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11或10 时此位不能被更改。
2	CH0NEN	通道0互补输出使能 当通道0配置为输出模式时，将此位置1使能通道0的互补输出。 0: 禁止通道0互补输出 1: 使能通道0互补输出
1	CH0P	通道0极性 当通道0配置为输出模式时，此位定义了输出信号极性。 0: 通道0高电平有效 1: 通道0低电平有效 当通道0配置为输入模式时，此位定义了 CI0 信号极性 [CH0NP, CH0P] 将选择 CI0FE0 或者 CI1FE0 的有效边沿或者捕获极性 [CH0NP==0, CH0P==0]: 把 CIxFE0 的上升沿作为捕获或者从模式下触发的有效信号，并且 CIxFE0 不会被翻转。

[CH0NP==0, CH0P==1]: 把ClxFE0的下降沿作为捕获或者从模式下触发的有效信号，并且ClxFE0会被翻转。

[CH0NP==1, CH0P==0]: 保留。

[CH0NP==1, CH0P==1]: 把ClxFE0的上升沿和下降沿都作为捕获或者从模式下触发的有效信号，并且ClxFE0不会被翻转。

当TIMERx_CCHP寄存器的PROT [1:0]=11或10 时此位不能被更改。

0	CH0EN	通道0捕获/比较使能
		当通道0配置为输出模式时，将此位置1使能CH0_O信号有效。当通道0配置为输入模式时，将此位置1使能通道0上的捕获事件。
	0:	禁止通道0
	1:	使能通道0

计数器寄存器 (TIMERx_CNT)

地址偏移: 0x24

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

位/位域	名称	描述
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器 (TIMERx_PSC)

地址偏移: 0x28

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

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

计数器自动重载寄存器 (TIMERx_CAR)

地址偏移: 0x2C

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

rw

位/位域	名称	描述
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

重复计数寄存器 (TIMERx_CREP)

地址偏移: 0x30

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

rw

位/位域	名称	描述
15:8	保留	必须保持复位值.
7:0	CREP[7:0]	重复计数器的值 这些位定义了更新事件的产生速率。重复计数器计数值减为0时产生更新事件。影子寄存器的更新速率也会受这些位影响(前提是影子寄存器被使能)。

通道 0 捕获/比较寄存器 (TIMERx_CH0CV)

地址偏移: 0x34

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															

rw

位/位域	名称	描述
15:0	CH0VAL[15:0]	通道0的捕获或比较值 当通道0配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道0配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 1 捕获/比较寄存器 (TIMERx_CH1CV)

地址偏移: 0x38

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															

rw

位/位域	名称	描述
15:0	CH1VAL[15:0]	<p>通道1的捕获或比较值</p> <p>当通道1配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。</p> <p>当通道1配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。</p>

通道 2 捕获/比较寄存器 (TIMERx_CH2CV)

地址偏移: 0x3C

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2VAL[15:0]															

rw

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

通道 3 捕获/比较寄存器 (TIMERx_CH3CV)

地址偏移: 0x40

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3VAL[15:0]															

rw

位/位域	名称	描述
15:0	CH3VAL[15:0]	<p>通道3的捕获或比较值</p> <p>当通道3配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。</p> <p>当通道3配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。</p>

互补通道保护寄存器 (TIMERx_CCHP)

地址偏移: 0x44

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POEN	OAEN	BRKP	BRKEN	ROS	IOS	PROT[1:0]						DTCFG[7:0]			

位/位域	名称	描述
15	POEN	<p>所有的通道输出使能</p> <p>根据OAEN位，该位可以软件设置或者硬件自动设置。一旦中止输入有效，该位被硬件异步清0。如果一个通道配置为输出模式，如果设置了相应的使能位 (TIMERx_CHCTL2寄存器的CHxEN, CHxNEN位)，则开启CHx_O 和CHx_ON 输出。</p> <p>0: 禁止通道输出或强制为空闲状态</p> <p>1: 使能通道输出</p>
14	OAEN	<p>自动输出使能</p> <p>此位定义了POEN位是否可以被硬件自动置1。</p> <p>0: POEN位不能被硬件置1</p> <p>1: 如果中止输入无效，下一次更新事件发生时，POEN位能被硬件自动置1</p> <p>此位只有在TIMERx_CCHP寄存器的PROT [1:0] =00时才可修改。</p>
13	BRKP	<p>中止极性</p> <p>此位定义了中止输入信号BKIN的极性。</p> <p>0: 中止输入低电平有效</p> <p>1: 中止输入高电平有效</p>
12	BRKEN	<p>中止使能</p> <p>此位置1使能中止事件和CCS时钟失败事件输入。</p> <p>0: 禁能中止输入</p> <p>1: 使能中止输入</p> <p>此位只有在TIMERx_CCHP寄存器的PROT [1:0] =00时才可修改。.</p>
11	ROS	<p>运行模式下“关闭状态”配置</p> <p>当POEN位被置1，此位定义了通道(带有互补输出且配置为输出模式)的输出状态。</p> <p>0: 当POEN位被置1，通道输出信号 (CHx_O/ CHx_ON)被禁止</p> <p>1: 当POEN位被置1，通道输出信号 (CHx_O / CHx_ON)被使能，和</p>

TIMER0_CHCTL2寄存器CHxEN/CHxNEN位有关

此位在TIMERx_CCHP寄存器的PROT [1:0]=10或11时不能被更改。

10	IOS	空闲模式下“关闭状态”配置 当POEN位被清0，此位定义了已经配置为输出模式的通道的输出状态。 0: 当POEN位被清0，通道输出信号(CHx_O/ CHx_ON)被禁止 1: 当POEN位被清0，通道输出信号(CHx_O/ CHx_ON)被使能，和 TIMERx_CHCTL2寄存器CHxEN/CHxNEN位有关 此位在TIMERx_CCHP寄存器的PROT [1:0]=10或11时不能被更改。.
9:8	PROT[1:0]	互补寄存器保护控制 这两位定义了寄存器的写保护特性。 00: 禁能保护模式。无写保护. 01: PROT模式0。TIMERx_CTL1寄存器中ISOx/ISOxN 位，TIMERx_CCHP 寄存器中BRKEN/BRKP/OAEN/DTCFG 位写保护 10: PROT模式1。除了PROT模式0下的寄存器写保护外，还有TIMERx_CHCTL2 寄存器中CHxP/CHxNP位（如果相应通道配置为输出模式），TIMERx_CCHP 寄存器中ROS/IOS 位。 11: PROT模式2。除了PROT模式1下的寄存器写保护外，还有 TIMERx_CHCTRLR0/1 中CHxCOMCTL/CHxCOMSEN位（如果相关通道配置为输出模式）写保护。 系统复位后这两位只能被写一次，一旦TIMERx_CCHP寄存器被写入，这两位被写保护
7:0	DTCFG[7:0]	死区时间控制 这些位定义了插入互补输出之间的死区持续时间。DTCFG值和死区时间的关系如下： DTCFG [7:5] =3'b0xx: DTvalue =DTCFG [7:0]x t _{DT} , t _{DT} =t _{DTS} . DTCFG [7:5] =3'b 10x: DTvalue = (64+DTCFG [5:0])x t _{DT} , t _{DT} =t _{DTS} *2. DTCFG [7:5] =3'b 110: DTvalue = (32+DTCFG [4:0])x t _{DT} , t _{DT} =t _{DTS} *8. DTCFG [7:5] =3'b 111: DTvalue = (32+DTCFG [4:0])x t _{DT} , t _{DT} =t _{DTS} *16. 此位只有在TIMERx_CCHP寄存器的PROT [1:0]=00时才可修改。

DMA 配置寄存器 (TIMERx_DMACFG)

地址偏移: 0x48

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

位/位域	名称	描述
15:13	保留	必须保持复位值.
12:8	DMATC [4:0]	DMA传输计数

该位域定义了DMA访问（读写）**TIMERx_DMABV**寄存器的数量

7:5	保留	必须保持复位值
4:0	DMATA [4:0]	DMA传输起始地址 该位域定义了DMA访问 TIMERx_DMABV 寄存器的第一个地址。当通过 TIMERx_DMA 第一次访问时，访问的就是该位域指定的地址。第二次访问 TIMERx_DMABV 时，将访问起始地址+0x4。
5'b0_0000: TIMERx_CTL0		
5'b0_0001: TIMERx_CTL1		
...		
总之： 起始地址 = TIMERx_CTL0 + DMATA *4		

DMA 发送缓冲区寄存器 (**TIMERx_DMATB**)

地址偏移: 0x4C

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

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

配置寄存器 (**TIMERx_CFG**)

地址偏移: 0xFC

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

域	位 / 位 域	名称	描述
	15:2	保留	必须保持复位值
	1	CHVSEL	写捕获比较寄存器选择位 此位由软件写1或清0。

1: 当写入捕获比较寄存器的值与寄存器当前值相等时，写入操作无效

0: 无影响

0 OUTSEL 输出值选择位

此位由软件写1或清0。

1: 如果POEN位与IOS位均为0，则输出无效

0: 无影响

16.2. 通用定时器 L0 (TIMERx, x=1, 2)

16.2.1. 简介

通用定时器L0是4通道定时器，支持输入捕获，输出比较，产生PWM信号控制电机和电源管理。通用定时器L0计数器是16位无符号计数器。

通用定时器L0是可编程的，可以被用来计数，其外部事件可以驱动其他定时器。

定时器和定时器之间是相互独立，但是他们可以被同步在一起形成一个更大的定时器，这些定时器的计数器一致地增加。

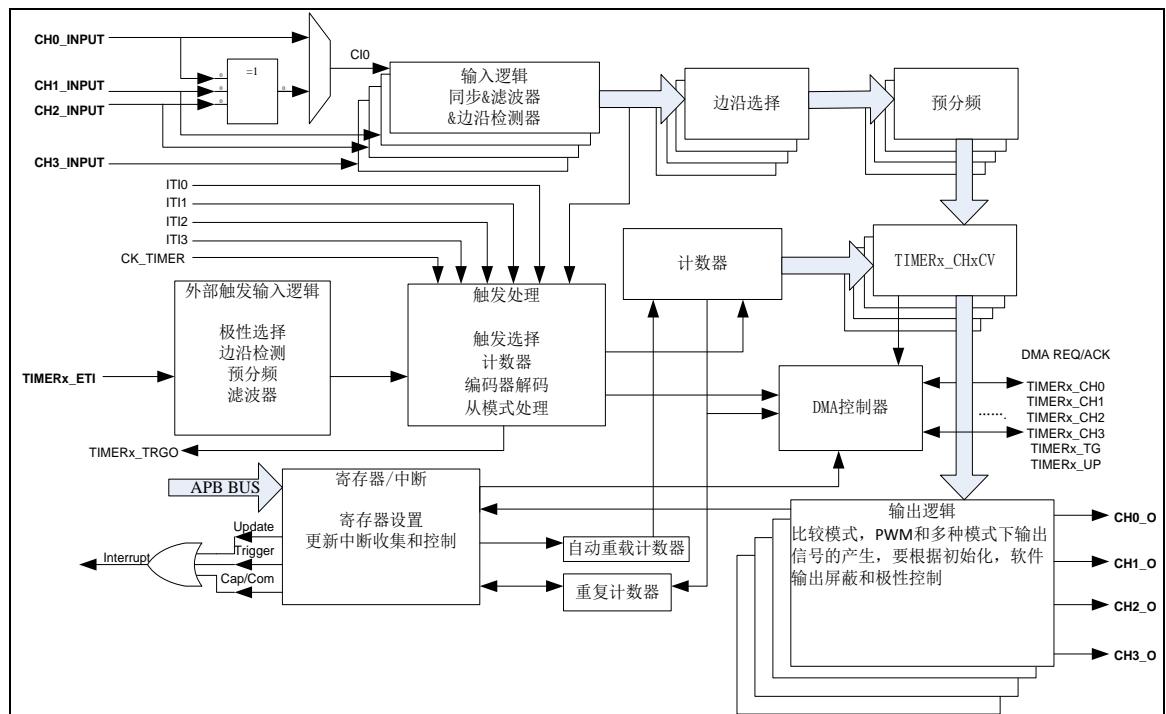
16.2.2. 主要特性

- 总通道数: 4;
- 计数器宽度: 16位(TIMER2), 32位(TIMER1);
- 时钟源可选: 内部时钟, 内部触发, 外部输入, 外部触发;
- 多种计数模式: 向上计数, 向下计数和中央计数;
- 正交编码器接口: 被用来追踪运动和分辨旋转方向和位置;
- 霍尔传感器接口: 用来做三相电机控制;
- 可编程的预分频器: 16位, 运行时可以被改变;
- 每个通道可配置: 输入捕获模式, 输出比较模式, 可编程的PWM模式, 单脉冲模式;
- 自动重装载功能;
- 中断输出和DMA请求: 更新事件, 触发事件, 比较/捕获事件;
- 多个定时器的菊花链使得一个定时器可以同时启动多个定时器;
- 定时器的同步允许被选择的定时器在同一个时钟周期开始计数;
- 定时器主/从模式控制器。

16.2.3. 结构框图

[图 16-32. 通用定时器 L0 结构框图](#)提供了通用定时器 L0 的内部细节

图 16-32. 通用定时器 L0 结构框图



16.2.4. 功能描述

时钟源选择

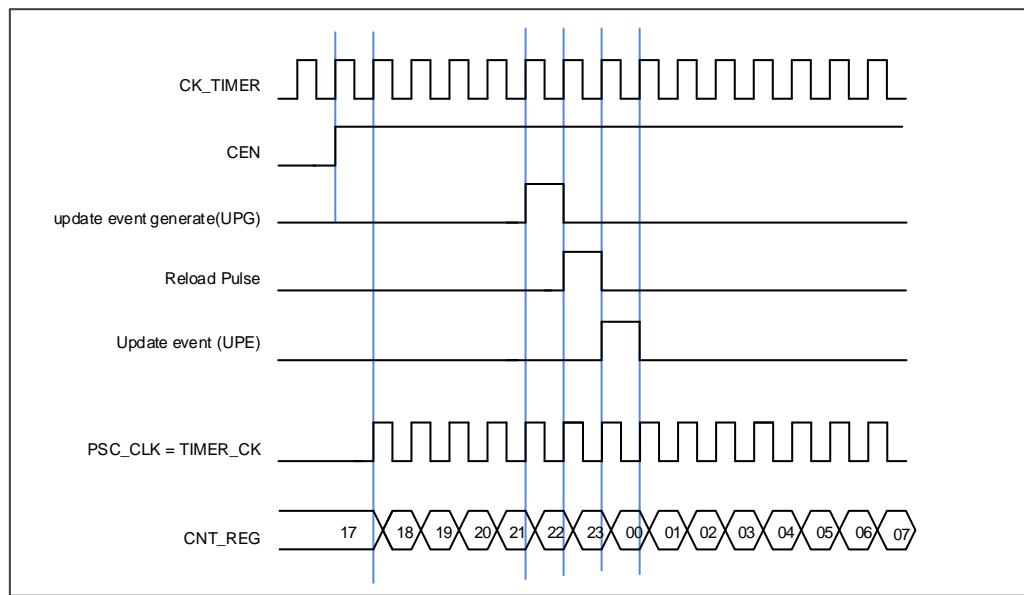
通用定时器L0可以由内部时钟源TIMER_CK或者由SMC (TIMERx_SMCFG寄存器位[2:0]) 控制的复用时钟源驱动。

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

如果禁能从模式 (SMC[2:0]==3'b000), 默认用来驱动计数器预分频器的是内部时钟源 CK_TIMER。当CEN置位, CK_TIMER经过预分频器 (预分频值由TIMERx_PSC寄存器确定) 产生PSC_CLK。

如果使能从模式控制器(将TIMERx_SMCFG寄存器的SMC[2:0]设置为包括0x1、0x2、0x3和0x7), 预分频器被其他时钟源(由TIMERx_SMCFG寄存器的TRGS [2:0]区域选择)驱动, 在下文说明。当从模式选择位SMC被设置为0x4、0x5和0x6, 计数器预分频器时钟源由内部时钟 TIMER_CK驱动。

图 16-33. 内部时钟分频为 1 时正常模式下的控制电路



- SMC[2:0]==3'b111(外部时钟模式0), 定时器选择外部输入引脚作为时钟源

计数器预分频器可以在TIMERx_CI0/ TIMERx_CI1引脚的每个上升沿或下降沿计数。这种模式可以通过设置SMC [2:0]为0x7同时设置TRGS [2:0]为0x4, 0x5或0x6来选择。Clx 是TIMERx_Cl通过数字滤波器采样后的信号。

计数器预分频器也可以在内部触发信号ITI0/1/2/3的上升沿计数。这种模式可以通过设置SMC [2:0]为0x7同时设置TRGS [2:0]为0x0, 0x1, 0x2 或者0x3。

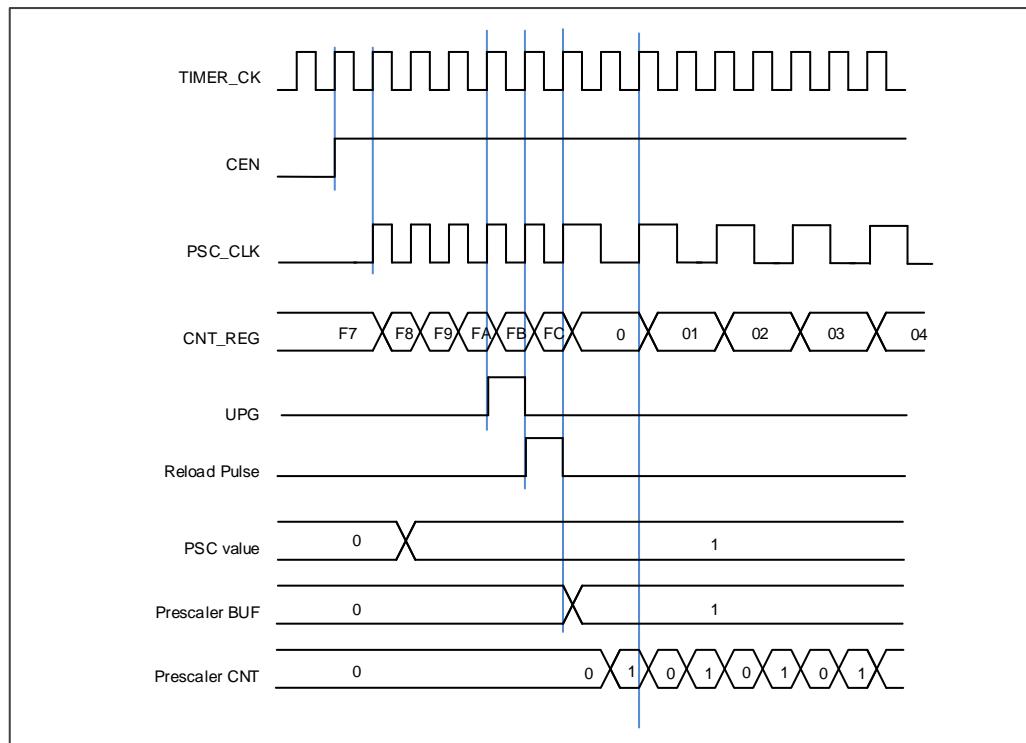
- SMC==1'b1(外部时钟模式1), 定时器选择外部输入引脚ETI作为时钟源

计数器预分频器可以在外部引脚ETI的每个上升沿或下降沿计数。这种模式可以通过设置TIMERx_SMCFG寄存器中的SMC1位为1来选择。另一种选择ETI信号作为时钟源方式是，设置SMC [2:0]为0x7同时设置TRGS [2:0]为0x7。注意ETI信号是通过数字滤波器采样ETI引脚得到的。。如果选择ETIF信号为时钟源，触发控制器包括边沿监测电路将在每个ETI信号上升沿产生一个时钟脉冲来为计数器预分频器提供时钟。

预分频器

预分频器可以将定时器的时钟 (TIMER_CK) 频率按1到65536之间的任意值分频，分频后的时钟PSC_CLK驱动计数器计数。分频系数受预分频寄存器TIMERx_PSC控制，这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 16-34. 当预分频器的参数从 1 变到 2 时，计数器的时序图



向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从0开始向上连续计数到自动加载值（定义在**TIMERx_CAR**寄存器中），一旦计数器计数到自动加载值，会重新从0开始向上计数并产生上溢事件。在向上计数模式中，**TIMERx_CTL0**寄存器中的计数方向控制位**DIR**应该被设置成0。

当通过**TIMERx_SWEVG**寄存器的UPG位置1来设置更新事件时，计数值会被清0，并产生更新事件。

如果**TIMERx_CTL0**寄存器的UPDIS置1，则禁止更新事件。

当发生更新事件时，所有的寄存器(自动重载寄存器，预分频寄存器)都将被更新。

下面的这些图给出了一些例子，当**TIMERx_CAR=0x63**时，计数器在不同预分频因子下的行为。

图 16-35. 向上计数时序图, PSC=0/1

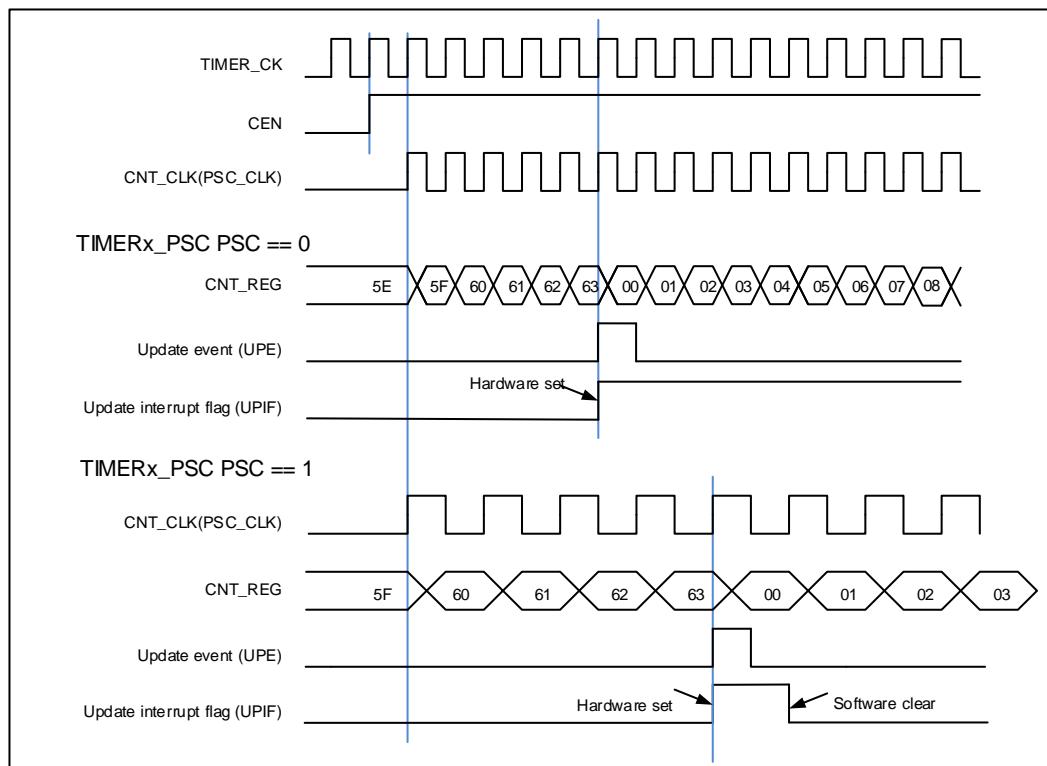
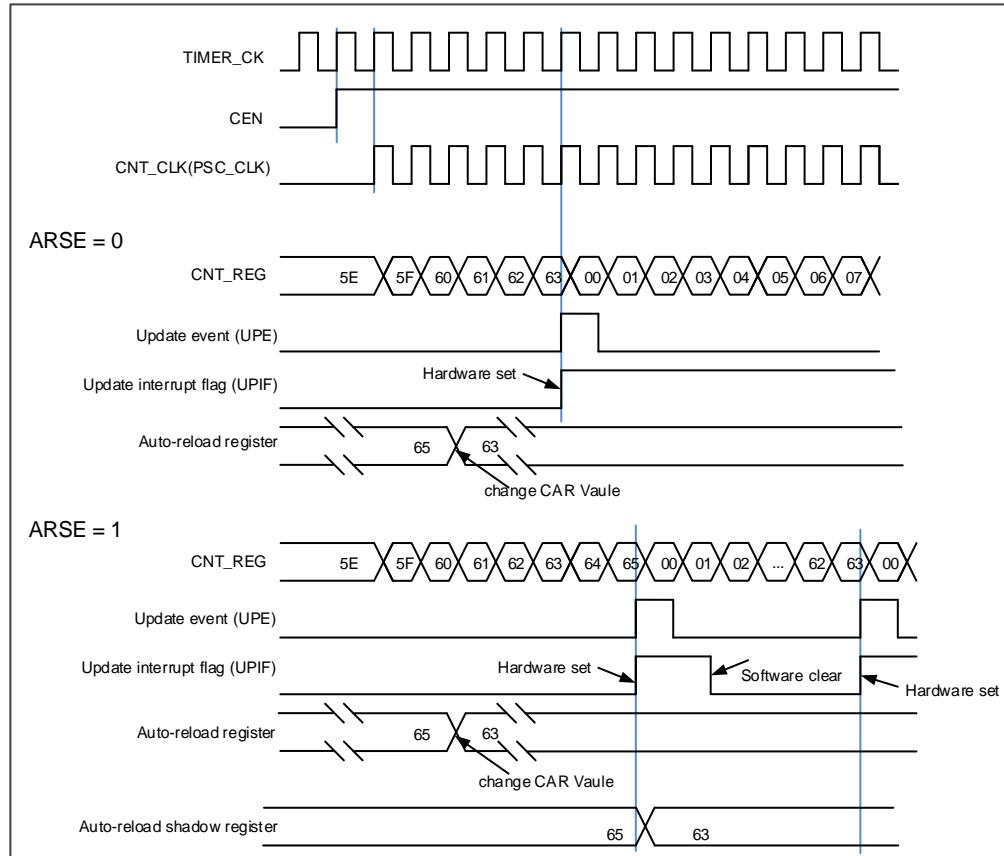


图 16-36. 向上计数时序图, 在运行时改变 TIMERx_CAR 寄存器的值



向下计数模式

在这种模式，计数器的计数方向是向下计数。计数器从自动加载值（定义在TIMERx_CAR寄存器中）向下连续计数到0。一旦计数器计数到0，计数器会重新从自动加载值开始计数并产生下溢事件。在向下计数模式中，TIMERx_CTL0寄存器中的计数方向控制位DIR应该被设置成1。

当通过TIMERx_SWEVG寄存器的UPG位置1来设置更新事件时，计数值会被初始化为自动加载值，并产生更新事件。

如果TIMERx_CTL0寄存器的UPDIS置1，则禁止更新事件。

当发生更新事件时，所有的寄存器(自动重载寄存器，预分频寄存器)都将被更新。

下面这些图给出了一些例子，当TIMERx_CAR=0x63时，计数器在不同时钟频率下的行为。

图 16-37. 向下计数时序图，PSC=0/1

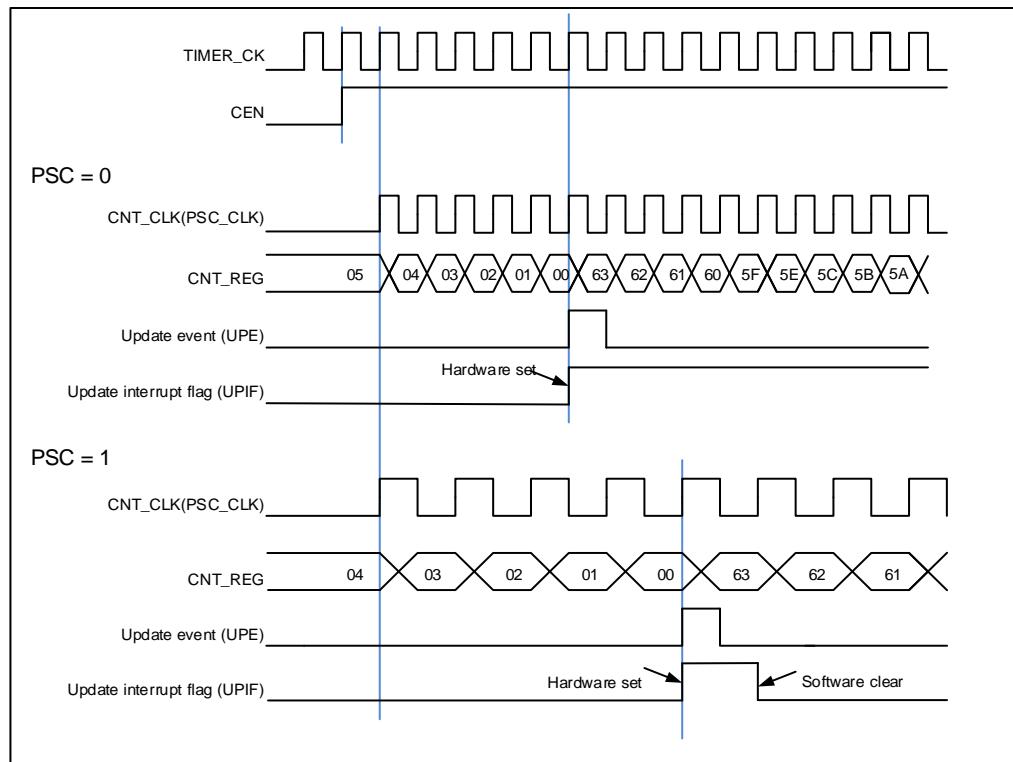
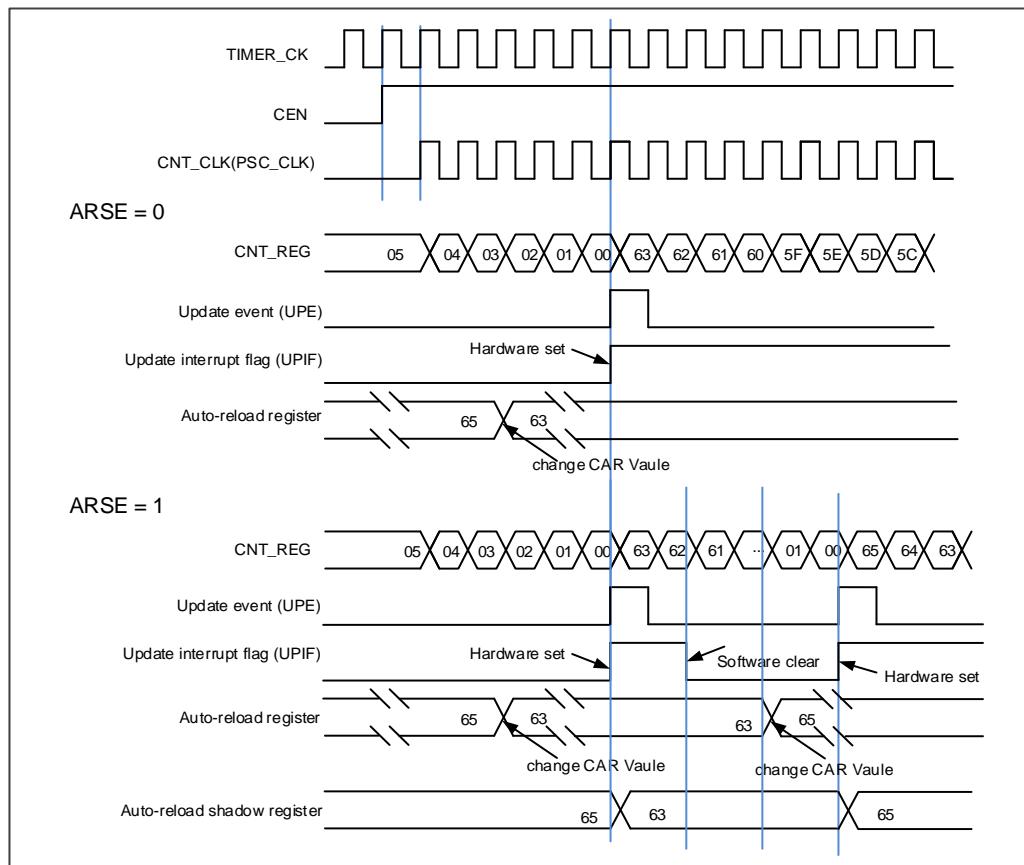


图 16-38. 向下计数时序图, 在运行时改变 TIMERx_CAR 寄存器值



中央对齐模式

在中央对齐模式下，计数器交替的从0开始向上计数到自动加载值，然后再向下计数到0。在向上计数模式中，定时器模块在计数器计数到（自动加载值-1）产生一个上溢事件；向下计数模式中，定时器模块在计数器计数到1时产生一个下溢事件。在中央计数模式中，TIMERx_CTL0 寄存器中的计数方向控制位DIR只读，表明了的计数方向。计数方向被硬件自动更新。

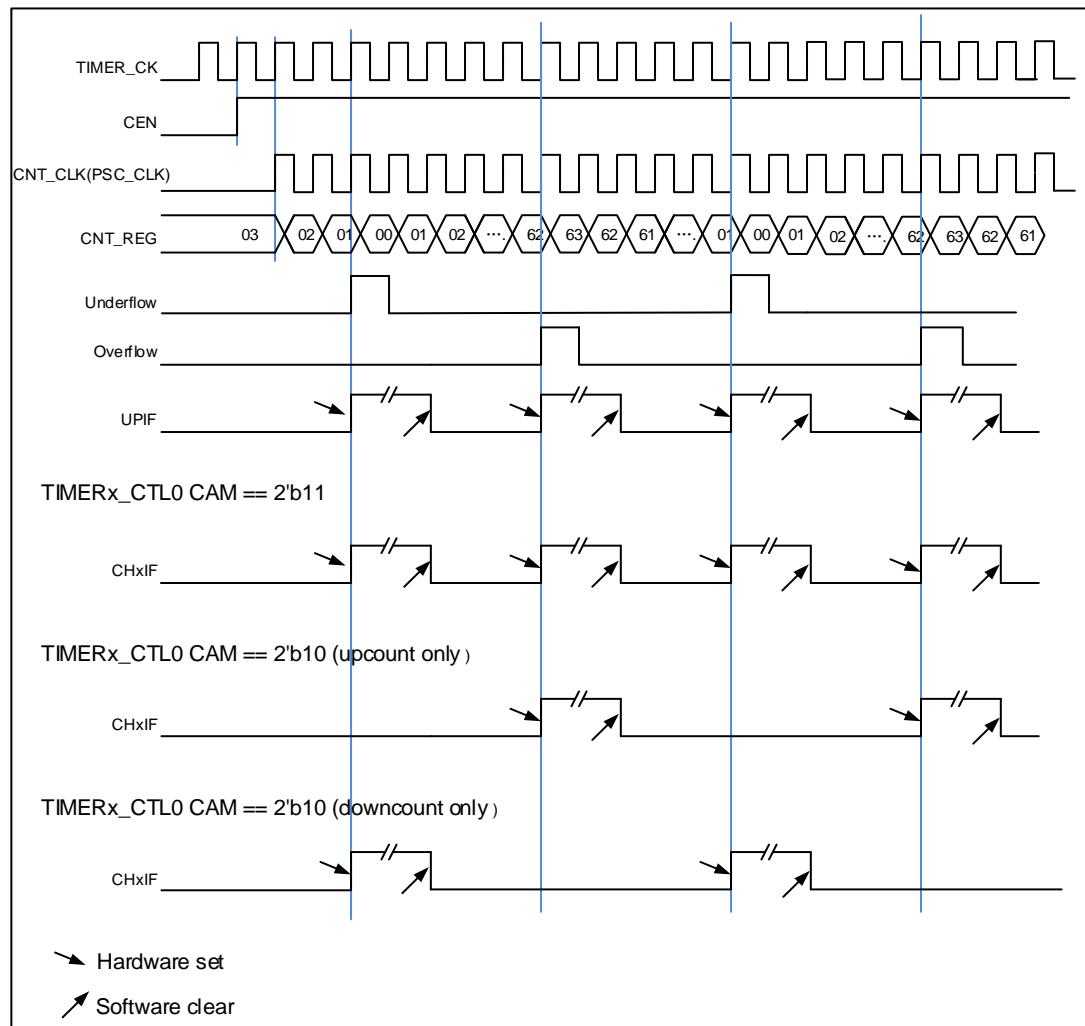
将TIMERx_SWEVG寄存器的UPG位置1可以初始化计数值为0，并产生一个更新事件，而无需考虑计数器在中央模式下是向上计数还是向下计数。

上溢或者下溢时，TIMERx_INTF 寄存器中的 UPIF 位都会被置 1，然而 CHxIF 位置 1 与 TIMERx_CTL0 寄存器中 CAM 的值有关。具体细节参考[图 16-39. 中央计数模式计数器时序图](#)。如果 TIMERx_CTL0 寄存器的 UPDIS 置 1，则禁止更新事件。

当发生更新事件时，所有的寄存器(自动重载寄存器，预分频寄存器)都将被更新。

下面这些图给出了一些例子，当TIMERx_CAR=0x63，TIMERx_PSC=0x0时，计数器的行为

图 16-39. 中央计数模式计数器时序图



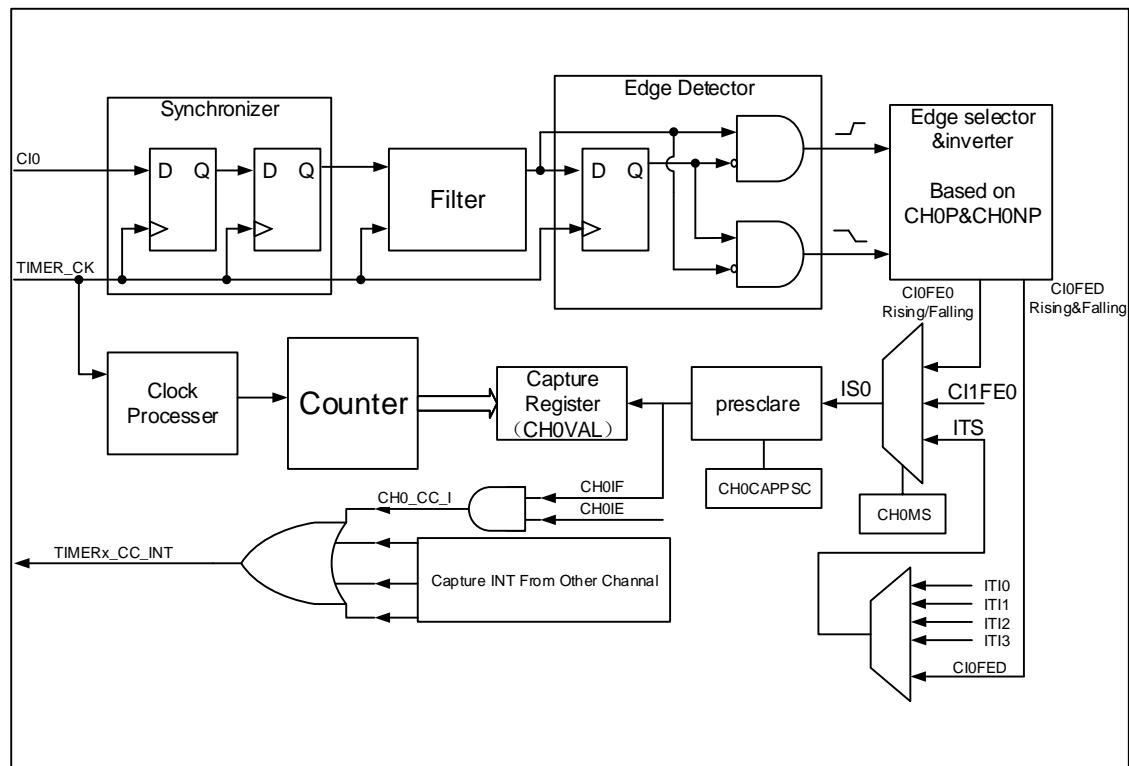
捕获/比较通道

通用定时器L0拥有四个独立的通道用于捕获输入或比较输出是否匹配。每个通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级。

■ 输入捕获模式

捕获模式允许通道测量一个波形时序，频率，周期，占空比等。输入级包括一个数字滤波器，一个通道极性选择，边沿检测和一个通道预分频器。如果在输入引脚上出现被选择的边沿，**TIMERx_CHxCV**寄存器会捕获计数器当前的值，同时**CHxIF**位被置1，如果**CHxIE = 1**则产生通道中断。

图 16-40. 输入捕获逻辑



通道输入信号 CI_x 有两种选择，一种是 $TIMERx_CHx$ 信号，另一种是 $TIMERx_CH0, TIMERx_CH1$ 和 $TIMERx_CH2$ 异或之后的信号。通道输入信号 CI_x 先被 $TIMER_CK$ 信号同步，然后经过数字滤波器采样，产生一个被滤波后的信号。通过边沿检测器，可以选择检测上升沿或者下降沿。通过配置 $CHxP$ 选择使用上升沿或者下降沿。配置 $CHxMS$ ，可以选择其他通道的输入信号，内部触发信号。配置 IC 预分频器，使得若干个输入事件后才产生一个有效的捕获事件。捕获事件发生， $CHxVAL$ 存储计数器的值。

配置步骤如下：

第一步：滤波器配置（ $TIMERx_CHCTL0$ 寄存器中 $CHxCAPFLT$ ）：

根据输入信号和请求信号的质量，配置相应的 $CHxCAPFLT$ 。

第二步：边沿选择（ $TIMERx_CHCTL2$ 寄存器中 $CHxP/CHxNP$ ）：

配置 $CHxP/CHxNP$ 选择上升沿或者下降沿。

第三步：捕获源选择（ $TIMERx_CHCTL0$ 寄存器中 $CHxMS$ ）：

一旦通过配置 $CHxMS$ 选择输入捕获源，必须确保通道配置在输入模式 ($CHxMS \neq 0x0$)，而且 $TIMERx_CHxCV$ 寄存器不能再被写。

第四步：中断使能（ $TIMERx_DMAINTEN$ 寄存器中 $CHxIE$ 和 $CHxDEN$ ）：

使能相应中断，可以获得中断和 DMA 请求。

第五步：捕获使能（ $TIMERx_CHCTL2$ 寄存器中 $CHxEN$ ）。

结果：当期望的输入信号发生时， $TIMERx_CHxCV$ 被设置成当前计数器的值， $CHxIF$ 为置 1。

如果 $CHxIF$ 位已经为 1，则 $CHxOF$ 位置 1。根据 $TIMERx_DMAINTEN$ 寄存器中 $CHxIE$ 和 $CHxDEN$ 的配置，相应的中断和 DMA 请求会被提出。

直接产生: 软件设置CHxG位，会直接产生中断和DMA请求。

输入捕获模式也可用来测量TIMERx_CHx引脚上信号的脉冲波宽度。例如，一个PWM波连接到CI0。配置TIMERx_CHCTL0寄存器中CH0MS为2'b01，选择通道0的捕获信号为CI0并设置上升沿捕获。配置TIMERx_CHCTL0寄存器中CH1MS为2'b10，选择通道1捕获信号为CI0并设置下降沿捕获。计数器配置为复位模式，在通道0的上升沿复位。TIMERX_CH0CV寄存器测量PWM的周期值，TIMERx_CH1CV寄存器测量PWM占空比值。

■ 输出比较模式

在输出比较模式，TIMERx可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的CHxVAL寄存器与计数器的值匹配时，根据CHxCOMCTL的配置，这个通道的输出可以被置高电平，被置低电平或者反转。当计数器的值与CHxVAL寄存器的值匹配时，CHxIF位被置1，如果CHxIE = 1则会产生中断，如果CHxDEN=1则会产生DMA请求。

配置步骤如下：

第一步：时钟配置：

配置定时器时钟源，预分频器等。

第二步：比较模式配置：

设置CHxCOMSEN位来配置输出比较影子寄存器；

设置CHxCOMCTL位来配置输出模式（置高电平/置低电平/反转）；

设置CHxP/CHxNP位来选择有效电平的极性；

设置CHxEN使能输出。

第三步：通过CHxIE/CHxDEN位配置中断/DMA请求使能。

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

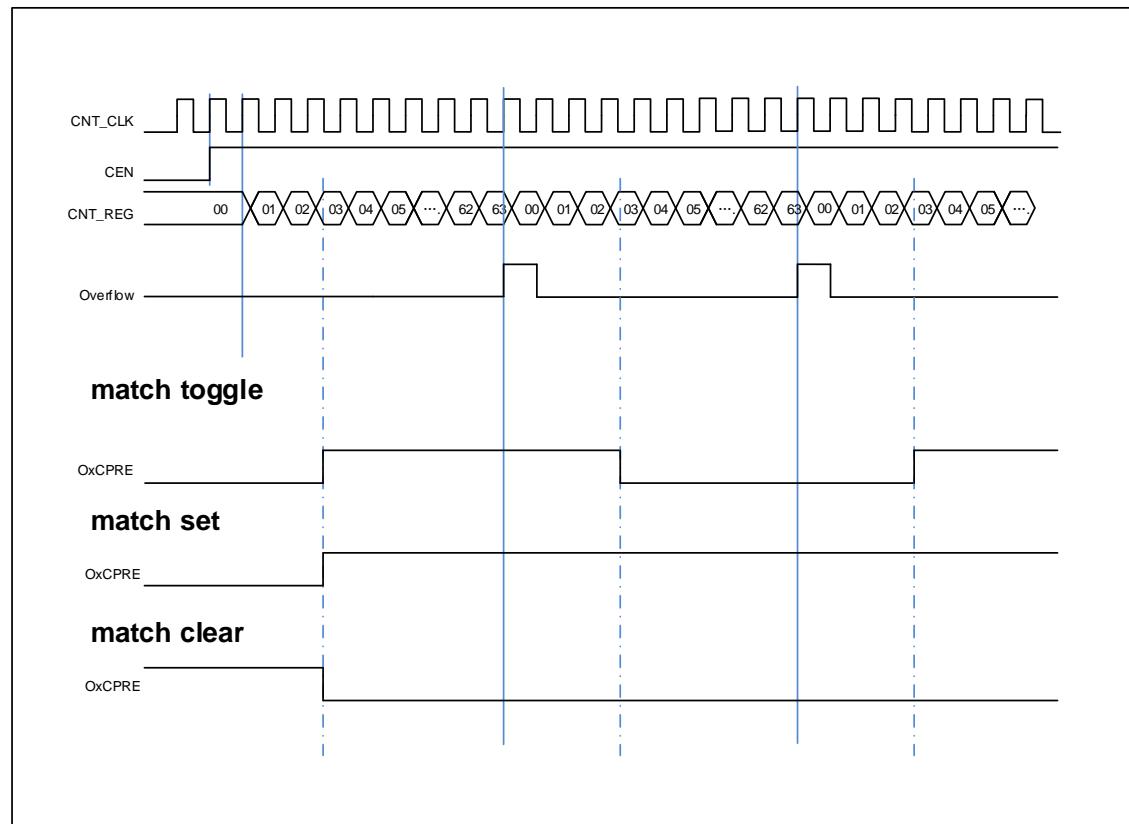
CHxVAL可以在运行时根据你所期望的波形而改变。

第五步：设置CEN位使能定时器。

图 16-41. 三种输出比较模式显示了三种比较输出模式：反转/置高电平/置低电平，CAR=0x63，

CHxVAL=0x3。

图 16-41. 三种输出比较模式



PWM 模式

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

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

EAPWM 的周期由 TIMERx_CAR 寄存器值决定，占空比由 TIMERx_CHxCV 寄存器值决定。[图 16-42. EAPWM 时序图](#)显示了 EAPWM 的输出波形和中断。

CAPWM 的周期由 (2*TIMERx_CAR 寄存器值) 决定，占空比由 (2*TIMERx_CHxCV 寄存器值) 决定。[图 16-43. CAPWM 时序图](#)显示了 CAPWM 的输出波形和中断。

在 PWM0 模式下 (CHxCOMCTL==3'b110)，如果 TIMERx_CHxCV 寄存器的值大于 TIMERx_CAR 寄存器的值，通道输出一直为有效电平。

在 PWM0 模式下 (CHxCOMCTL==3'b110)，如果 TIMERx_CHxCV 寄存器的值等于 0，通道输出一直为无效电平。

图 16-42. EAPWM 时序图

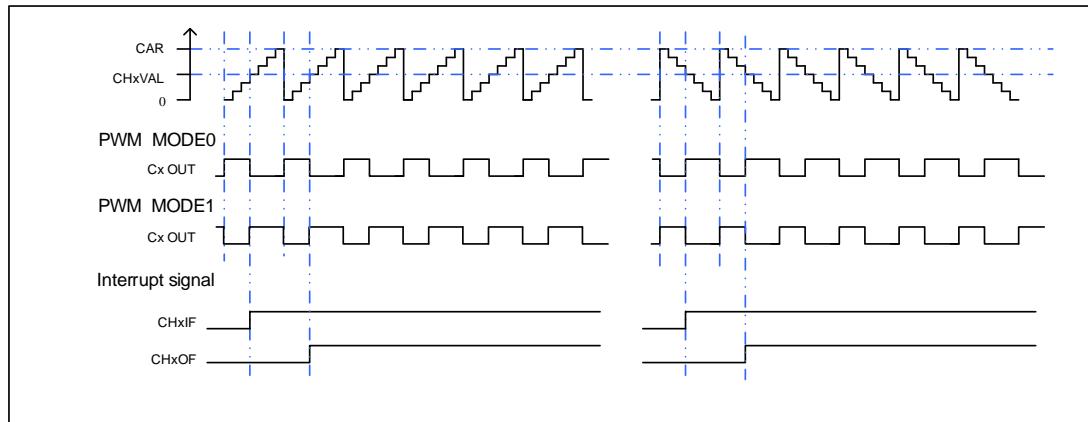
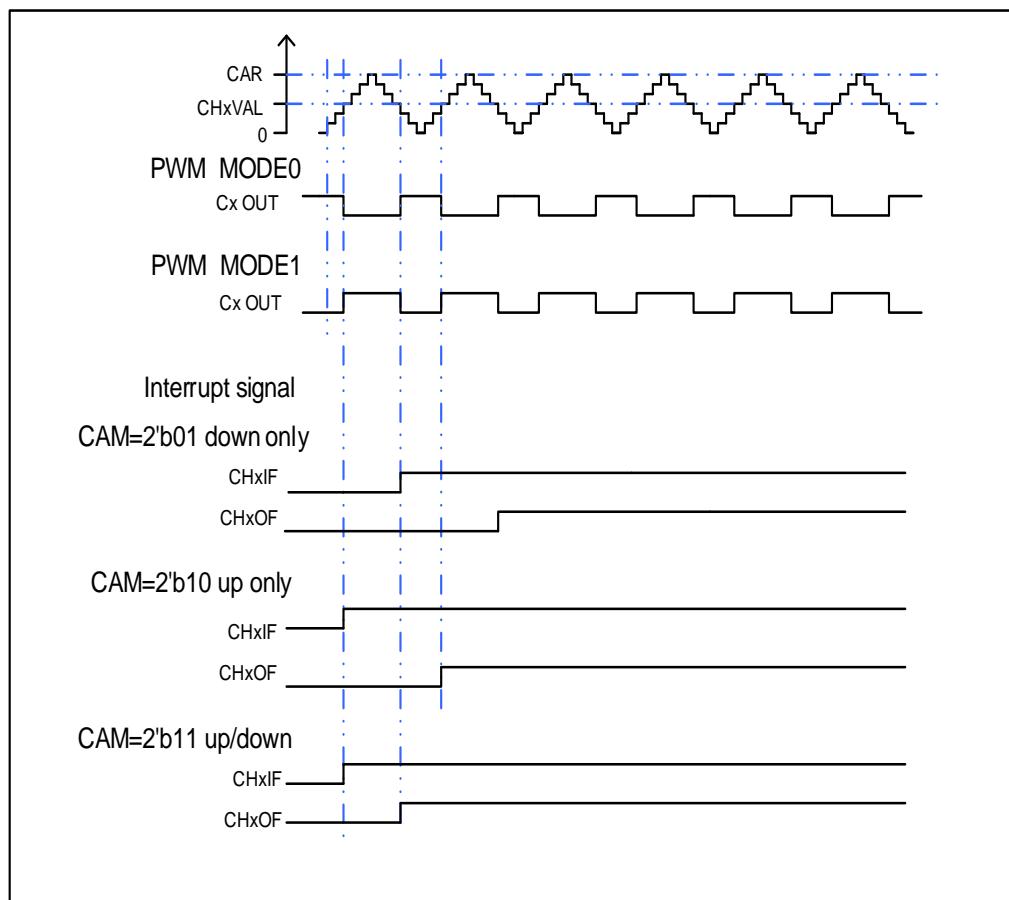


图 16-43. CAPWM 时序图



通道输出参考信号

当TIMERx用于输出匹配比较模式下，设置CHxCOMCTL位可以定义OxCPR_E信号(通道x准备信号)类型。OxCPR_E信号有若干类型的输出功能，包括，设置CHxCOMCTL=0x00可以保持原始电平；设置CHxCOMCTL=0x01可以将OxCPR_E信号设置为高电平；设置

CHxCOMCTL=0x02可以将OxCPR信号设置为低电平；设置CHxCOMCTL=0x03，在计数器值和TIMERx_CHxCV寄存器的值匹配时，可以翻转输出信号。

PWM模式0和PWM模式1是OxCPR的另一种输出类型，设置CHxCOMCTL位域位0x06或0x07可以配置PWM模式0/PWM模式1。在这些模式中，根据计数器值和TIMERx_CHxCV寄存器值的关系以及计数方向，OxCPR信号改变其电平。具体细节描述，请参考相应的位。

设置CHxCOMCTL =0x04或0x05可以实现OxCPR信号的强制输出功能。输出比较信号能够直接由软件强置为有效或无效状态，而不依赖于TIMERx_CHxCV的值和计数器值之间间的比较结果。

设置CHxCOMCEN=1，当由外部ETI引脚信号产生的ETIFE信号为高电平时，OxCPR被强制为低电平。在下一次更新事件到来时，OxCPR信号才会回到有效电平状态。

正交译码器

正交译码器功能使用由 TIMERx_CH0 和 TIMERx_CH1 引脚生成的 CI0 和 CI1 正交信号各自相互作用产生计数值。在每个输入源改变期间，DIR 位被硬件自动改变。输入源可以是只有 CI0，可以只有 CI1，或者可以同时有 TI1 和 TI2，通过设置 SMC=0x01, 0x02 或 0x03 来选择使用哪种模式。计数器计数方向改变的机制如[表 16-5. 计数方向与编码器信号之间的关系](#)所示。正交译码器可以当作一个带有方向选择的外部时钟，这意味着计数器会在 0 和自动加载值之间连续的计数。因此，用户必须在计数器开始计数前配置 TIMERx_CAR 寄存器。

表 16-5. 计数方向与编码器信号之间的关系

计数模式	电平	TI1FP1		TI2FP2	
		上升	下降	上升	下降
只有CI0	CI1FE1=1	向下	向上	-	-
	CI1FE1=0	向上	向下	-	-
只有CI1	CI0FE0=1	-	-	向上	向下
	CI0FE0=0	-	-	向下	向上
CI0和CI1	CI1FE1=1	向下	向上	X	X
	CI1FE1=0	向上	向下	X	X
	CI0FE0=1	X	X	向上	向下
	CI0FE0=0	X	X	向下	向上

注意：“-”意思是“无计数”；“X”意思是不可能。

图 16-44. 编码器接口模式下计数器运行例子

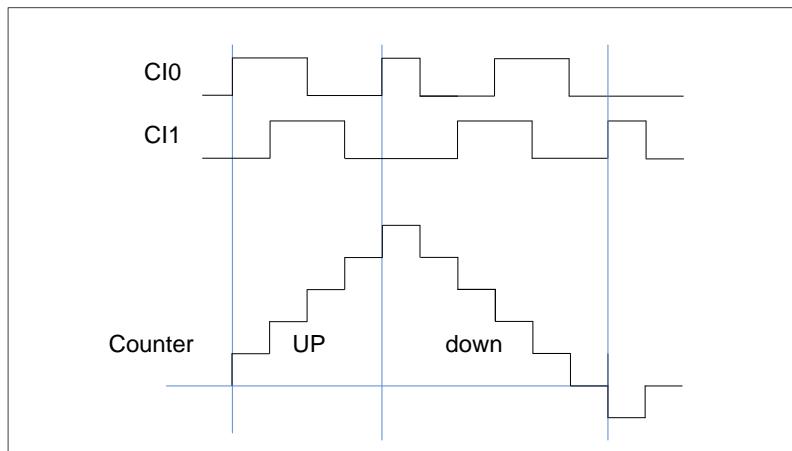
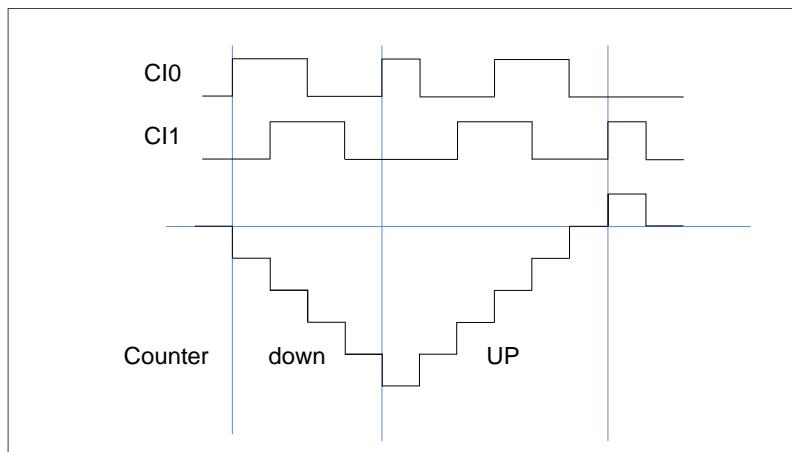


图 16-45. CI0FE0 极性反相的编码器接口模式下的例子

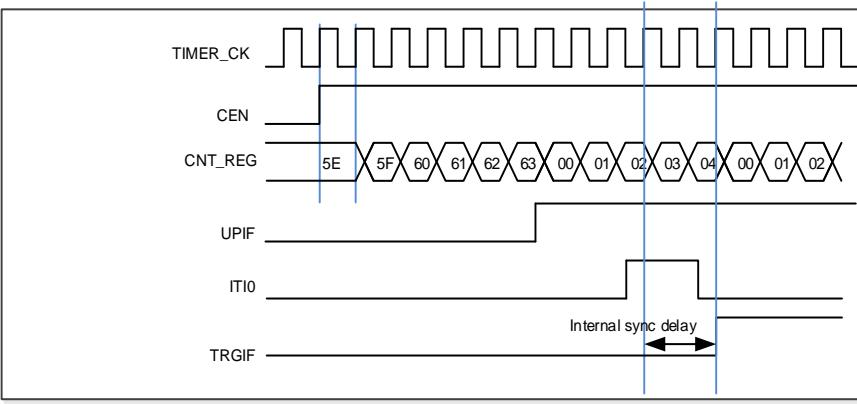
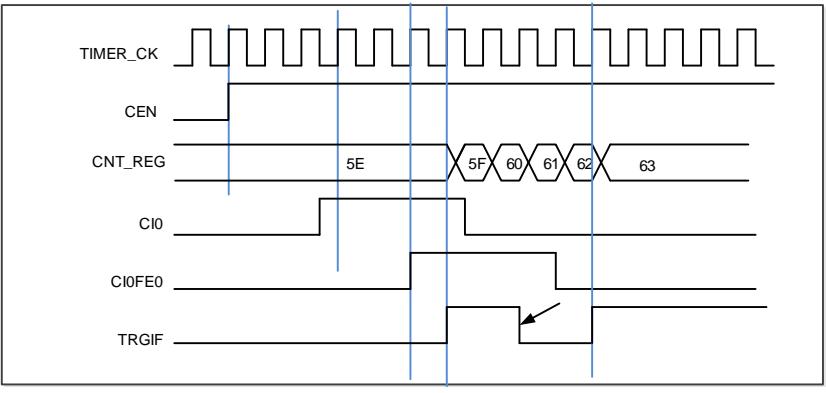


从控制器

TIMERx能在多种模式下同步外部触发，包括复位模式，暂停模式和事件模式，可以通过设置 TIMERx_SMCFG 寄存器中的 SMC [2:0] 配置这些模式。这些模式的输入触发源可以通过设置 TIMERx_SMCFG 寄存器中的 TRGS [2:0] 来选择。

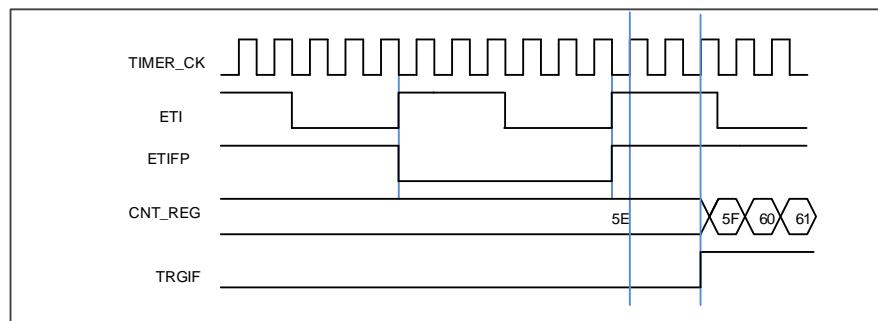
表 16-6. 从模式列表和举例（通用定时器 L0）

	模式选择	触发源选择	极性选择	滤波和预分频
列举	SMC[2:0] 3'b100 (复位模式) 3'b101 (暂停模式) 3'b110 (事件模式)	TRGS[2:0] 000: ITI0 001: ITI1 010: ITI2 011: ITI3 100: CI0F_ED 101: CI0FE0 110: CI1FE1	如果触发源是 CI0FE0 或者 CI1FE1，配置 CHxP 和 CHxNP 来选择极性和反相 如果触发源是 ETIF，配置 ETP 选	触发源 ITIx，滤波和预分频不可用 触发源 CIx，配置 CHxCAPFLT 设置滤波，分频不可用 触发源是 ETIF，滤波和预分频不可用

	模式选择	触发源选择	极性选择	滤波和预分频
		111: ETIFP	择极性和反相	
例1	复位模式 当触发输入上升沿， 计数器清零重启	TRGIS[2:0]=3'b000 选择ITIO为触发源	触发源是 ITIO, 极性选择不可用	触发源是 ITIO, 滤波和预分频不可用
图 16-46. 复位模式下的控制电路				
				
例2	暂停模式 当触发输入为低的时候, 计数器暂停计数	TRGIS[2:0]=3'b101 选择CI0FE0为触发源	TIOS=0. (非异或) [CH0NP==0, CH0P==0]不反相. 在上升沿捕获	在这个例子中滤波被旁路
			图 16-47. 暂停模式下的控制电路	
				

	模式选择	触发源选择	极性选择		滤波和预分频
例3	事件模式 触发输入的上升沿 计数器开始计数	TRGIS[2:0]=3'b111 选择ETIF为触发源.	ETP = 0 没有极性改变		ETPSC = 1, 2分频. ETFC = 0, 无滤波

图 16-48. 事件模式下的控制电路



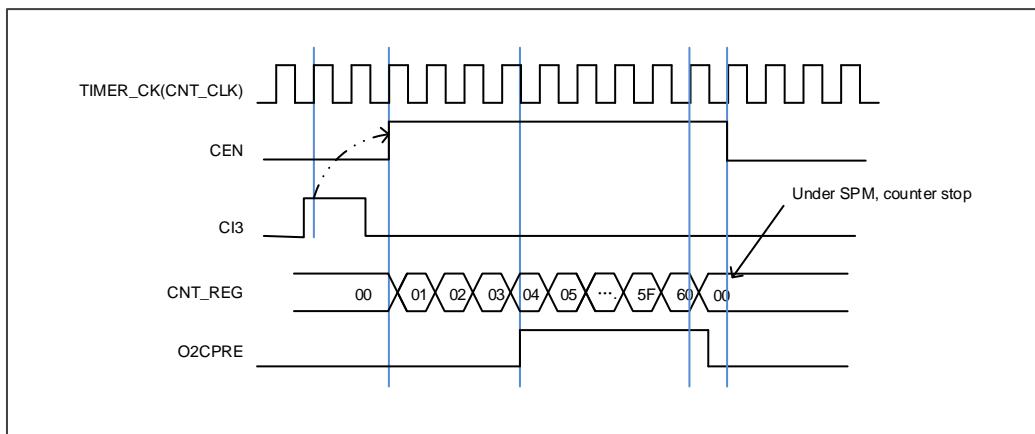
单脉冲模式

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

一旦设置定时器运行在单脉冲模式下，没有必要设置TIMERx_CTL0寄存器的定时器使能位CEN=1来使能计数器。触发信号沿或者软件写CEN=1都可以产生一个脉冲，此后CEN位一直保持为1直到更新事件发生或者CEN位被软件写0。如果CEN位被软件清0，计数器停止工作，计数值被保持。如果CEN值被硬件更新事件自动清0，计数器将被再次初始化。

在单脉冲模式下，有效的外部触发边沿会将CEN位置1，使能计数器。然而，执行计数值和TIMERx_CHxCV寄存器值的比较结果依然存在一些时钟延迟。为了最大限度减少延迟，用户可以将TIMERx_CHCTL0/1寄存器的CHxCOMFEN位置1。单脉冲模式下，触发上升沿产生之后，OxCPRE信号将被立即强制转换为与发生比较匹配时相同的电平，但是不用考虑比较结果。只有输出通道配置为PWM0或PWM1输出运行模式下时CHxCOMFEN位才可用，触发源来源于触发信号。

图 16-49. 单脉冲模式, TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60



定时器互连

参考[高级定时器互连 \(TIMERx,x=0\)](#)

表 16-7. TIMERx(x=1,2)定时器内部互连

Slave TIMER	ITI0(TRGS = 000)	ITI1(TRGS = 001)	ITI2(TRGS = 010)	ITI3(TRGS = 011)
TIMER1	TIMER0	TIMER14	TIMER2	保留
TIMER2	TIMER0	TIMER1	TIMER14	保留

定时器 DMA 模式

定时器DMA模式是指通过DMA模块配置定时器的寄存器。有两个跟定时器DMA模式相关的寄存器: TIMERx_DMACFG和TIMERx_DMATB。当然, 必须要使能DMA请求, 一些内部中断事件可以产生DMA请求。当中断事件发生, TIMERx会给DMA发送请求。DMA配置成M2P模式, PADDR是TIMERx_DMATB寄存器地址, DMA就会访问TIMERx_DMATB寄存器。实际上, TIMERx_DMATB寄存器只是一个缓冲, 定时器会将TIMERx_DMATB映射到一个内部寄存器, 这个内部寄存器由TIMERx_DMACFG寄存器中的DMATA来指定。如果TIMERx_DMACFG寄存器的DMATC位域值为0, 表示1次传输, 定时器的发送1个DMA请求就可以完成。如果TIMERx_DMACFG寄存器的DMATC位域值不为1, 例如其值为3, 表示4次传输, 定时器就需要再多发3次DMA请求。在这3次请求下, DMA对TIMERx_DMATB寄存器的访问会映射到访问定时器的DMATA+0x4, DMATA+0x8, DMATA+0xc寄存器。总之, 发生一次DMA内部中断请求, 定时器会连续发送(DMATC+1)次请求。

如果再来1次DMA请求事件, TIMERx将会重复上面的过程。

定时器调试模式

当Cortex™-M4内核停止, DBG_CTL0寄存器中的TIMERx_HOLD配置位被置1, 定时器计数器停止。

16.2.5. TIMERx 寄存器 ($x=1,2$)

控制寄存器 0 (TIMERx_CTL0)

地址偏移: 0x00

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					CKDIV[1:0]	ARSE	CAM[1:0]	DIR	SPM	UPS	UPDIS	CEN			
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
15:10	保留	必须保持复位值
9:8	CKDIV[1:0]	<p>时钟分频</p> <p>通过软件配置CKDIV，规定定时器时钟(TIMER_CK) 与死区时间和采样时钟(DTS) 之间的分频系数，死区发生器和数字滤波器会用到DTS时间。</p> <p>00: $f_{DTS}=f_{\text{TIMER_CK}}$ 01: $f_{DTS}=f_{\text{TIMER_CK}}/2$ 10: $f_{DTS}=f_{\text{TIMER_CK}}/4$ 11: 保留</p>
7	ARSE	<p>自动重载影子使能</p> <p>0: 禁能TIMERx_CAR寄存器的影子寄存器 1: 使能TIMERx_CAR寄存器的影子寄存器</p>
6:5	CAM[1:0]	<p>计数器对齐模式选择</p> <p>00: 无中央对齐模式(边沿对齐模式). DIR位指定了计数方向 01: 中央对齐向下计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式 (TIMERx_CHCTL0寄存器中CHxMS=00)，只有在向下计数时，通道的比较中断标志置1 10: 中央对齐向上计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式 (TIMERx_CHCTL0寄存器中CHxMS=00)，只有在向上计数时，通道的比较中断标志置1 11: 中央对齐上下计数置1模式。计数器在中央计数模式计数，通道被配置在输出模式 (TIMERx_CHCTL0寄存器中CHxMS=00)，在向上和向下计数时，通道的比较中断标志都会置1</p> <p>当计数器使能以后，改为不能从0x00切换到非0x00</p>
4	DIR	<p>方向</p> <p>0: 向上计数 1: 向下计数</p> <p>当计数器配置为中央对齐模式或编码器模式时，该位为只读</p>
3	SPM	单脉冲模式

		0: 更新事件发生后，计数器继续计数 1: 在下一次更新事件发生时，CEN硬件清零并且计数器停止计数
2	UPS	<p>更新请求源 软件配置该为，选择更新事件源.</p> <p>0:使能后，下述任一事件产生更新中断或DMA请求： <ul style="list-style-type: none"> - UPG位被置1 - 计数器溢出/下溢 - 从模式控制器产生的更新 <p>1:使能后只有计数器溢出/ 下溢才产生更新中断或DMA请求。</p> </p>
1	UPDIS	<p>禁止更新. 该位用来使能或禁能更新事件的产生 .</p> <p>0: 更新事件使能.当以下事件之一发生时，更新事件产生，具有缓存的寄存器被装入它们的预装载值： <ul style="list-style-type: none"> - UPG位被置1 - 计数器溢出/下溢 - 从模式控制器产生一个更新事件 <p>1: 更新事件禁能. 带有缓存的寄存器保持原有值，如果UPG位被置1或者从模式控制器产生一个硬件复位事件，计数器和预分频器被重新初始化</p> </p>
0	CEN	<p>计数器使能 0: 计数器禁能 1: 计数器使能 在软件将CEN位置1后，外部时钟、暂停模式和编码器模式才能工作。触发模式可以自动地通过硬件设置CEN位。</p>

控制寄存器 1 (TIMERx_CTL1)

地址偏移: 0x04

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					TIOS		MMC[2:0]		DMAS		保留.				

位/位域	名称	描述
15:8	保留	必须保持复位值
7	TIOS	<p>通道0触发输入选择</p> <p>0: 选择TIMERx_CH0 引脚作为通道0的触发输入 1: 选择TIMERx_CH0, CH1和CH2 引脚异或的结果作为通道0的触发输入</p>
6:4	MMC[2:0]	<p>主模式控制</p> <p>这些位控制TRGO信号的选择，TRGO信号由主定时器发给从定时器用于同步功能</p>

000: 复位。TIMERx_SWEVG寄存器的UPG位被置1或从模式控制器产生复位触发一次TRGO脉冲，后一种情况下，TRGO上的信号相对实际的复位会有一个延迟。

001: 使能。此模式可用于同时启动多个定时器或控制在一段时间内使能从定时器。主模式控制器选择计数器使能信号作为触发输出TRGO。当CEN控制位被置1或者暂停模式下触发输入为高电平时，计数器使能信号被置1。在暂停模式下，计数器使能信号受控于触发输入，在触发输入和TRGO 上会有一个延迟，除非选择了主/从模式。

010: 更新。主模式控制器选择更新事件作为TRGO。

011: 捕获/比较脉冲.通道0在发生一次捕获或一次比较成功时，主模式控制器产生一个TRGO脉冲

100: 比较。在这种模式下主模式控制器选择O0CPRE信号被用于作为触发输出TRGO

101: 比较。在这种模式下主模式控制器选择O1CPRE信号被用于作为触发输出TRGO

110: 比较。在这种模式下主模式控制器选择O2CPRE信号被用于作为触发输出TRGO

111: 比较。在这种模式下主模式控制器选择O3CPRE信号被用于作为触发输出TRGO

3	DMAS	DMA请求源选择
		0: 当通道捕获/比较事件发生时，发送通道x的DMA请求 . 1: 当更新事件发生，发送通道x的DMA请求
2:0	保留	必须保持复位值

从模式配置寄存器 (TIMERx_SMCFG)

地址偏移: 0x08

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	SMC1	ETPSC[1:0]		ETFC[3:0]		MSM		TRGS[2:0]		OCRC		SMC[2:0]			

位/位域	名称	描述
15	ETP	外部触发极性 该位指定ETI信号的极性 0: ETI高电平或上升沿有效 . 1: ETI低电平或下降沿有效 .
14	SMC1	SMC的一部分为了使能外部时钟模式1 在外部时钟模式1，计数器由ETIF信号上的任意有效边沿驱动 0: 外部时钟模式1禁能 1: 外部时钟模式1使能

复位模式，暂停模式和事件模式可以与外部时钟模式1同时使用。但是TRGS必须不能为3'b111。

如果外部时钟模式0和外部时钟模式1同时被使能，外部时钟的输入是ETIF
注意：外部时钟模式0使能在寄存器的SMC位域。

13:12	ETPSC[1:0]	外部触发预分频 外部触发信号ETI的频率不能超过TIMER_CK频率的1/4。当输入较快的外部时钟时， 可以使用预分频降低ETIP的频率。 00: 预分频禁能 01: ETI 频率被2分频 10: ETI 频率被4分频 11: ETI 频率被8分频
11:8	ETFC[3:0]	外部触发滤波控制 数字滤波器是一个事件计数器，它记录到N个事件后会产生一个输出的跳变。这些位定义了对ETI信号采样的频率和对ETI数字滤波的带宽。 0000: 滤波器禁能 $f_{SAMP} = f_{DTS}$, N=1. 0001: $f_{SAMP} = f_{TIMER_CK}$, N=2. 0010: $f_{SAMP} = f_{TIMER_CK}$, N=4. 0011: $f_{SAMP} = f_{TIMER_CK}$, N=8. 0100: $f_{SAMP} = f_{DTS}/2$, N=6. 0101: $f_{SAMP} = f_{DTS}/2$, N=8. 0110: $f_{SAMP} = f_{DTS}/4$, N=6. 0111: $f_{SAMP} = f_{DTS}/4$, N=8. 1000: $f_{SAMP} = f_{DTS}/8$, N=6. 1001: $f_{SAMP} = f_{DTS}/8$, N=8. 1010: $f_{SAMP} = f_{DTS}/16$, N=5. 1011: $f_{SAMP} = f_{DTS}/16$, N=6. 1100: $f_{SAMP} = f_{DTS}/16$, N=8. 1101: $f_{SAMP} = f_{DTS}/32$, N=5. 1110: $f_{SAMP} = f_{DTS}/32$, N=6. 1111: $f_{SAMP} = f_{DTS}/32$, N=8.
7	MSM	主-从模式 该位被用来同步被选择的定时器同时开始计数。通过TRIGI和TRGO，定时器被连接在一起，TRGO用做启动事件。 0: 主从模式禁能 1: 主从模式使能
6:4	TRGS[2:0]	触发选择 该位域用来指定选择哪一个信号作为用来同步计数器的触发输入源 000: 内部触发输入0 (ITI0) 001: 内部触发输入1 (ITI1) 010: 内部触发输入2 (ITI2) 011: 保留

100: CI0的边沿标志位 (CI0F_ED)

101: 滤波后的通道0输入 (CI0FE0)

110: 滤波后的通道1输入(CI1FE1)

111: 滤波后的外部触发输入(ETIFP)

从模式被使能后这些位不能改

3	OCRC	OCPRE清除源选择 0: OCPRE_CLR_INT连接到OCPRE_CLR输入 1: OCPRE_CLR_INT连接到ETIF
2:0	SMC[2:0]	从模式控制 000: 关闭从模式. 如果CEN=1, 则预分频器直接由内部时钟驱动 001: 编码器模式0. 根据CI0FE0的电平, 计数器在CI1FE1的边沿向上/下计数 010: 编码器模式1. 根据CI1FE1的电平, 计数器在CI0FE0的边沿向上/下计数 011: 编码器模式2. 根据另一个信号的输入电平, 计数器在CI0FE0和CI1FE1的边沿向上/ 下计数 100: 复位模式. 选中的触发输入的上升沿重新初始化计数器, 并且更新影子寄存器. 101: 暂停模式. 当触发输入为高时, 计数器的时钟开启。一旦触发输入变为低, 则计数器停止 110: 事件模式.计数器在触发输入的上升沿启动。计数器不能被从模式控制器关闭。 111: 外部时钟模式0. 选中的触发输入的上升沿驱动计数器

DMA 和中断使能寄存器 (TIMERx_DMAINTEN)

地址偏移: 0x0C

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TRGDEN	保留	CH3DEN	CH2DEN	CH1DEN	CH0DEN	UPDEN	保留	TRGIE	保留	CH3IE	CH2IE	CH1IE	CHOIE	UPIE

位/位域	名称	描述
15	保留	必须保持复位值.
14	TRGDEN	触发DMA请求使能 0: 禁止触发DMA请求 1: 使能触发DMA请求
13	保留	必须保持复位值.
12	CH3DEN	通道3比较/捕获 DMA请求使能 0: 禁止通道3比较/捕获DMA请求 1: 使能通道3比较/捕获DMA请求
11	CH2DEN	通道2比较/捕获 DMA请求使能 0: 禁止通道2比较/捕获DMA请求

		1: 使能通道2比较/捕获DMA请求
10	CH1DEN	通道1比较/捕获 DMA请求使能 0: 禁止通道1比较/捕获DMA请求 1: 使能通道1比较/捕获DMA请求
9	CH0DEN	通道0比较/捕获 DMA请求使能 0: 禁止通道0比较/捕获DMA请求 1: 使能通道0比较/捕获DMA请求
8	UPDEN	更新DMA请求使能 0: 禁止更新DMA请求 1: 使能更新DMA请求
7	保留	必须保持复位值.
6	TRGIE	触发中断使能 0: 禁止触发中断 1: 使能触发中断
5	保留	必须保持复位值.
4	CH3IE	通道3比较/捕获中断使能 0: 禁止通道3中断 1: 使能通道3中断
3	CH2IE	通道2比较/捕获中断使能 0: 禁止通道2中断 1: 使能通道2中断
2	CH1IE	通道1比较/捕获中断使能 0: 禁止通道1中断 1: 使能通道1中断
1	CH0IE	通道0比较/捕获中断使能 0: 禁止通道0中断 1: 使能通道0中断
0	UPIE	更新中断使能 0: 禁止更新中断 1: 使能更新中断

中断标志寄存器 (**TIMERx_INTF**)

地址偏移: 0x10

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

保留	CH3OF	CH2OF	CH1OF	CH0OF	保留	TRGIF	保留	CH3IF	CH2IF	CH1IF	CHOIF	UPIF
	rc_w0	rc_w0	rc_w0	rc_w0	.		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位/位域	名称	描述
15:13	保留	必须保持复位值.
12	CH3OF	通道3捕获溢出标志 参见CH0OF描述
11	CH2OF	通道2捕获溢出标志 参见CH0OF描述
10	CH1OF	通道1捕获溢出标志 参见CH0OF描述
9	CH0OF	通道1捕获溢出标志 当通道0被配置为输入模式时，在CH0IF标志位已经被置1后，捕获事件再次发生时，该标志位可以由硬件置1。该标志位由软件清0. 0: 无捕获溢出中断发生 1: 发生了捕获溢出中断
8:7	保留	必须保持复位值.
6	TRGIF	触发中断标志 当发生触发事件时，此标志由硬件置1。此位由软件清0。当从模式控制器处于除暂停模式外的其它模式时，在TRGI 输入端检测到有效边沿，产生触发事件。当从模式控制器处于暂停模式时，TRGI的任意边沿都可以产生触发事件。 0: 无触发事件产生 1: 触发中断产生
5	保留	必须保持复位值.
4	CH3IF	通道3比较/捕获中断标志 参见CHOIF描述
3	CH2IF	通道2比较/捕获中断标志 参见CHOIF描述
2	CH1IF	通道1比较/捕获中断标志 参见CHOIF描述
1	CH0IF	通道0比较/捕获中断标志 此标志由硬件置1软件清0。当通道0在输入模式下时，捕获事件发生时此标志位被置1；当通道0在输出模式下时，此标志位在一个比较事件发生时被置1。 0: 无通道0中断发生 1: 通道0中断发生
0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置1，软件清0。

0: 无更新中断发生

1: 发生更新中断

软件事件产生寄存器 (**TIMERx_SWEVG**)

地址偏移: 0x14

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								TRGG	保留.	CH3G	CH2G	CH1G	CH0G	UPG	

w w w w w w

位/位域	名称	描述
15:7	保留	必须保持复位值.
6	TRGG	<p>触发事件产生</p> <p>此位由软件置1, 由硬件自动清0. 当此位被置1, TIMERx_INTF寄存器的TRGIF标志位被置1, 若开启对应的中断和DMA, 则产生相应的中断和DMA传输。</p> <p>0: 无触发事件产生</p> <p>1: 产生触发事件</p>
5	保留	必须保持复位值.
4	CH3G	<p>通道3捕获或比较事件发生</p> <p>参见CH0G描述</p>
3	CH2G	<p>通道2捕获或比较事件发生</p> <p>参见CH0G描述</p>
2	CH1G	<p>通道1捕获或比较事件发生</p> <p>参见CH0G描述</p>
1	CH0G	<p>通道0捕获或比较事件发生</p> <p>该位由软件置1, 用于在通道0产生一个捕获/比较事件, 由硬件自动清0。当此位被置1, CH0IF标志位被置1, 若开启对应的中断和DMA, 则发出相应的中断和DMA请求。此外, 如果通道0配置为输入模式, 计数器的当前值被TIMERx_CH0CV寄存器捕获, 如果CH0IF标志位已经为1, 则CH0OF标志位被置1。</p> <p>0: 不产生通道0捕获或比较事件</p> <p>1: 发生通道0捕获或比较事件</p>
0	UPG	<p>更新事件产生</p> <p>此位由软件置1, 被硬件自动清0。当此位被置1, 如果选择了中央对齐或向上计数模式, 计数器被清0。否则(向下计数模式)计数器将载入自动重载值, 预分频计数器将同时被清除。</p>

0: 无更新事件产生

1: 产生更新事件

通道控制寄存器 0 (TIMERx_CHCTL0)

地址偏移: 0x18

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1COMCEN	CH1COMCTL[2:0]	CH1COMSEN	CH1COMFEN	CH1MS[1:0]	CH0COMCEN	CH0COMCTL[2:0]	CH0COMSEN	CH0COMFEN	CH0MS[1:0]						
CH1CAPFLT[3:0]	CH1CAPPSC[1:0]				CH0CAPFLT[3:0]	CH0CAPPSC[1:0]									

rw rw rw rw rw rw

输出比较模式:

位/位域	名称	描述
15	CH1COMCEN	通道1输出比较清0使能 参见CH0COMCEN描述
14:12	CH1COMCTL[2:0]	通道1输出比较模式 参见CH0COMCTL描述
11	CH1COMSEN	通道1输出比较影子寄存器使能 参见CH0COMSEN描述
10	CH1COMFEN	通道1输出比较快速使能 参见CH0COMFEN描述
9:8	CH1MS[1:0]	通道1模式选择 这些位定义了通道的方向和输入信号的选择。只有当通道关闭 (TIMERx_CHCTL2寄存器的CH1EN位被清0)时这些位才可以写。 00: 通道1配置为输出 01: 通道1配置为输入, IS1映射在CI1FE1上 10: 通道1配置为输入, IS1映射在CI0FE1上 11: 通道1配置为输入, IS1映射在ITS上, 此模式仅工作在内部触发器输入被选中时(由 TIMERx_SMCFGFG寄存器的TRGS位选择)。
7	CH0COMCEN	通道0输出比较清0使能 当此位被置1, 当检测到ETIF输入高电平时, O0CPRE参考信号被清0 0: 禁止通道0输出比较清零 1: 使能通道0输出比较清零
6:4	CH0COMCTL[2:0]	通道0输出比较模式 此位定义了输出参考信号O0CPRE的动作, 而O0CPRE决定了CH0_O、CH0_ON的值。O0CPRE高电平有效, 而CH0_O、CH0_ON的有效电平取决于

		于CH0P、CH0NP位。
		000: 冻结。输出比较寄存器TIMERx_CH0CV与计数器TIMERx_CNT间的比较对O0CPRE不起作用
		001: 匹配时设置为高。当计数器的值与捕获/比较值寄存器TIMERx_CH0CV相同时，强制O0CPRE为高。
		010: 匹配时设置为低。当计数器的值与捕获/比较值寄存器TIMERx_CH0CV相同时，强制O0CPRE为低。
		011: 匹配时翻转。当计数器的值与捕获/比较值寄存器TIMERx_CH0CV相同时，强制O0CPRE翻转。
	100:	强制为低。强制O0CPRE为低电平
	101:	强制为高。强制O0CPRE为高电平
	110:	PWM 模式0 (EAPWM)。在向上计数时，一旦计数器值小于TIMERx_CH0CV时，O0CPRE为高电平，否则为低电平。在向下计数时，一旦计数器的值大于TIMERx_CH0CV时，O0CPRE 为低电平，否则为高电平。
	111:	PWM 模式1(CAPWM)。在向上计数时，一旦计数器值小于TIMERx_CH0CV时，O0CPRE为低电平，否则为高电平。在向下计数时，一旦计数器的值大于TIMERx_CH0CV时，O0CPRE为高电平，否则为低电平。在PWM模式0或PWM模式1中，只有当比较结果改变了或者输出比较模式中从冻结模式切换到PWM模式时，CxCOMR电平才改变。 当TIMERx_CCHP寄存器的PROT [1:0]=11且CH0MS =00（比较模式）时此位不能被改变。
3	CH0COMSEN	通道0输出比较影子寄存器使能 当此位被置1，TIMERx_CH0CV寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0: 禁止通道0输出/比较影子寄存器 1: 使能通道0输出/比较影子寄存器 仅在单脉冲模式下(TIMERx_CTL0寄存器的SPM =1)，可以在未确认预装载寄存器情况下使用PWM 模式 当TIMERx_CCHP寄存器的PROT [1:0]=11且CH0MS =00时此位不能被改变。
2	CH0COMFEN	通道0输出比较快速使能 当该位为1时，如果通道配置为PWM0模式或者PWM1模式，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH0_O被设置为比较电平而与比较结果无关。 0: 禁止通道0输出比较快速。当触发器的输入有一个有效沿时，激活CH0_O输出的最小延时为5个时钟周期 1: 使能通道0输出比较快速。当触发器的输入有一个有效沿时，激活CH0_O输出的最小延时为3个时钟周期
1:0	CH0MS[1:0]	通道0 I/O模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2寄存器的CH0EN位被清0)时这些位才可写。 00: 通道0配置为输出

- 01: 通道0配置为输入, ISO映射在CI0FE0上
 10: 通道0配置为输入, ISO映射在CI1FE0上
 11: 通道0配置为输入, ISO映射在ITS上。此模式仅工作在内部触发输入被选中时(通过设置TIMERx_SMCFGFG寄存器的TRGS位)

输入捕获模式:

位/位域	名称	描述
15:12	CH1CAPFLT[3:0]	通道1输入捕获滤波控制 参见CH0CAPFLT描述
11:10	CH1CAPPSC[1:0]	通道1输入捕获预分频器 参见CH0CAPPSC描述
9:8	CH1MS[1:0]	通道1模式选择 与输出模式相同
7:4	CH0CAPFLT[3:0]	通道0输入捕获滤波控制 数字滤波器由一个事件计数器组成, 它记录N个输入事件后会产生一个输出的跳变。这些位定义了CI0输入信号的采样频率和数字滤波器的长度。 0000: 无滤波器, $f_{SAMP} = f_{DTS}$, N=1 0001: $f_{SAMP} = f_{PCLK}$, N=2 0010: $f_{SAMP} = f_{PCLK}$, N=4 0011: $f_{SAMP} = f_{PCLK}$, N=8 0100: $f_{SAMP} = f_{DTS}/2$, N=6 0101: $f_{SAMP} = f_{DTS}/2$, N=8 0110: $f_{SAMP} = f_{DTS}/4$, N=6 0111: $f_{SAMP} = f_{DTS}/4$, N=8 1000: $f_{SAMP} = f_{DTS}/8$, N=6 1001: $f_{SAMP} = f_{DTS}/8$, N=8 1010: $f_{SAMP} = f_{DTS}/16$, N=5 1011: $f_{SAMP} = f_{DTS}/16$, N=6 1100: $f_{SAMP} = f_{DTS}/16$, N=8 1101: $f_{SAMP} = f_{DTS}/32$, N=5 1110: $f_{SAMP} = f_{DTS}/32$, N=6 1111: $f_{SAMP} = f_{DTS}/32$, N=8
3:2	CH0CAPPSC[1:0]	通道0输入捕获预分频器 这2位定义了通道0输入的预分频系数。当TIMERx_CHCTL2寄存器中的CH0EN =0时, 则预分频器复位。 00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获 01: 每2个事件触发一次捕获 10: 每4个事件触发一次捕获 11: 每8个事件触发一次捕获
1:0	CH0MS[1:0]	通道0模式选择

与输出比较模式相同

通道控制寄存器 1 (TIMERx_CHCTL1)

地址偏移: 0x1C

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3COM CEN	CH3COMCTL[2:0]	CH3COM SEN	CH3COM FEN		CH3MS[1:0]	CH2COM CEN	CH2COMCTL[2:0]		CH2COM SEN	CH2COM FEN			CH2MS[1:0]		
	CH3CAPFLT[3:0]		CH3CAPPSC[1:0]				CH2CAPFLT[3:0]			CH2CAPPSC[1:0]					

rw rw rw rw rw rw

输出比较模式:

位/位域	名称	描述
15	CH3COMCEN	通道3输出比较清0使能 参见CH0COMCEN描述
14:12	CH3COMCTL[2:0]	通道3输出比较模式 参见CH0COMCTL描述
11	CH3COMSEN	通道3输出比较影子寄存器使能 参见CH0COMSEN描述
10	CH3COMFEN	通道3输出比较快速使能 参见CH0COMSEN描述
9:8	CH3MS[1:0]	通道3模式选择 这些位定义了通道的方向和输入信号的选择。只有当通道关闭 (TIMERx_CHCTL2寄存器的CH3EN位被清0)时这些位才可以写。 00: 通道3配置为输出 01: 通道3配置为输入, IS3映射在CI3FE3上 10: 通道3配置为输入, IS3映射在CI2FE3上 11: 通道3配置为输入, IS3映射在ITS上, 此模式仅工作在内部触发器输入被选中时(由 TIMERx_SMCFGFG寄存器的TRGS位选择)。
7	CH2COMCEN	通道2输出比较清0使能 当此位被置1, 当检测到ETIF输入高电平时, O2CPRE参考信号被清0 0: 使能通道2输出比较清零 1: 禁止通道2输出比较清零
6:4	CH2COMCTL[2:0]	通道2输出比较模式 此位定义了输出参考信号O2CPRE的动作, 而O2CPRE决定了CH2_O、CH2_ON的值。O2CPRE高电平有效, 而CH2_O、CH2_ON的有效电平取决于CH2P、CH2NP位。 000: 冻结。输出比较寄存器TIMERx_CH2CV与计数器TIMERx_CNT间的比

较对O2CPRE不起作用

001: 匹配时设置为高。当计数器的值与捕获/比较值寄存器TIMERx_CH2CV相同时，强制O2CPRE为高。

010: 匹配时设置为低。当计数器的值与捕获/比较值寄存器TIMERx_CH2CV相同时，强制O2CPRE为低。

011: 匹配时翻转。当计数器的值与捕获/比较值寄存器TIMERx_CH2CV相同时，强制O2CPRE翻转。

100: 强制为低。强制O2CPRE为低电平

101: 强制为高。强制O2CPRE为高电平

110: PWM 模式0 (EAPWM)。在向上计数时，一旦计数器值小于 TIMERx_CH2CV时，O2CPRE为高电平，否则为低电平。在向下计数时，一旦计数器的值大于TIMERx_CH2CV时，O2CPRE 为低电平，否则为高电平。

111: PWM 模式1(CAPWM)。在向上计数时，一旦计数器值小于 TIMERx_CH2CV时，O2CPRE为低电平，否则为高电平。在向下计数时，一旦计数器的值大于TIMERx_CH2CV时，O2CPRE为高电平，否则为低电平。在PWM模式0或PWM模式1中，只有当比较结果改变了或者输出比较模式中从冻结模式切换到PWM模式时，CxCOMR电平才改变。

当TIMERx_CCHP寄存器的PROT [1:0]=11且CH2MS =00（比较模式）时此位不能被改变。

3 CH2COMSEN

通道0输出比较影子寄存器使能

当此位被置1，TIMERx_CH2CV寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。

0: 禁止通道2输出/比较影子寄存器

1: 使能通道2输出/比较影子寄存器

仅在单脉冲模式下(TIMERx_CTL0寄存器的SPM =1)，可以在未确认预装载寄存器情况下使用PWM 模式

当TIMERx_CCHP寄存器的PROT [1:0]=11且CH2MS =00时此位不能被改变。

2 CH2COMFEN

通道2输出比较快速使能

当该位为1时，如果通道配置为PWM0模式或者PWM1模式，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH2_O被设置为比较电平而与比较结果无关。

0: 禁止通道2输出比较快速。当触发器的输入有一个有效沿时，激活CH2_O输出的最小延时为5个时钟周期

1: 使能通道2输出比较快速。当触发器的输入有一个有效沿时，激活CH2_O输出的最小延时为3个时钟周期

1:0 CH2MS[1:0]

通道2 I/O模式选择

这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭 (TIMERx_CHCTL2寄存器的CH2EN位被清0)时这些位才可写。

00: 通道2配置为输出

01: 通道2配置为输入，IS2映射在CI2FE2上

10: 通道2配置为输入，IS2映射在CI3FE2上

11: 通道2配置为输入, IS2映射在ITS上。此模式仅工作在内部触发输入被选中时(通过设置TIMERx_SMCFGFG寄存器的TRGS位)

输入捕获模式:

位/位域	名称	描述
15:12	CH3CAPFLT[3:0]	通道3输入捕获滤波控制 参见CH0CAPFLT描述
11:10	CH3CAPPSC[1:0]	通道3输入捕获预分频器 参见CH0CAPPSC描述
9:8	CH3MS[1:0]	通道3模式选择 与输出模式相同
7:4	CH2CAPFLT[3:0]	通道2输入捕获滤波控制 数字滤波器由一个事件计数器组成, 它记录N个输入事件后会产生一个输出的跳变。这些位定义了CI2输入信号的采样频率和数字滤波器的长度。 0000: 无滤波器, $f_{SAMP} = f_{DTS}$, N=1 0001: $f_{SAMP} = f_{PCLK}$, N=2 0010: $f_{SAMP} = f_{PCLK}$, N=4 0011: $f_{SAMP} = f_{PCLK}$, N=8 0100: $f_{SAMP} = f_{DTS}/2$, N=6 0101: $f_{SAMP} = f_{DTS}/2$, N=8 0110: $f_{SAMP} = f_{DTS}/4$, N=6 0111: $f_{SAMP} = f_{DTS}/4$, N=8 1000: $f_{SAMP} = f_{DTS}/8$, N=6 1001: $f_{SAMP} = f_{DTS}/8$, N=8 1010: $f_{SAMP} = f_{DTS}/16$, N=5 1011: $f_{SAMP} = f_{DTS}/16$, N=6 1100: $f_{SAMP} = f_{DTS}/16$, N=8 1101: $f_{SAMP} = f_{DTS}/32$, N=5 1110: $f_{SAMP} = f_{DTS}/32$, N=6 1111: $f_{SAMP} = f_{DTS}/32$, N=8
3:2	CH2CAPPSC[1:0]	通道2输入捕获预分频器 这2位定义了通道2输入的预分频系数。当TIMERx_CHCTL2寄存器中的CH2EN =0时, 则预分频器复位。 00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获 01: 每2个事件触发一次捕获 10: 每4个事件触发一次捕获 11: 每8个事件触发一次捕获
1:0	CH2MS[1:0]	通道2模式选择 与输出比较模式相同

通道控制寄存器 2 (TIMERx_CHCTL2)

地址偏移: 0x20

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3NP	保留	CH3P	CH3EN	CH2NP	保留	CH2P	CH2EN	CH1NP	保留	CH1P	CH1EN	CH0NP	保留	CH0P	CH0EN

rw rw

位/位域	名称	描述
15	CH3NP	通道3互补输出极性 参考CH0NP 描述
14	保留	必须保持复位值
13	CH3P	通道3极性 参考CH0P描述
12	CH3EN	通道3使能 参考CH0EN 描述
11	CH2NP	通道2互补输出极性 参考CH0NP 描述
10	保留	必须保持复位值
9	CH2P	通道2极性 参考CH0P 描述
8	CH2EN	通道2使能 参考CH0EN描述
7	CH1NP	通道1互补输出极性 参考CH0NP描述
6	保留	必须保持复位值
5	CH1P	通道1极性 参考CH0P 描述
4	CH1EN	通道1使能 参考CH0EN 描述
3	CH0NP	通道0互补输出极性 当通道0配置为输出模式，该位保持0。 当通道0配置为输入模式时，此位和CH0P联合使用，作为输入信号CI0的极性选择控制信号。 当TIMERx_CCHP寄存器的PROT [1:0]=11或10 时此位不能被更改。

2	保留	必须保持复位值
1	CH0P	<p>通道0极性</p> <p>当通道0配置为输出模式时，此位定义了输出信号极性。</p> <p>0：通道0高电平有效</p> <p>1：通道0低电平有效</p> <p>当通道0配置为输入模式时，此位定义了CI0信号极性</p> <p>[CH0NP, CH0P] 将选择CI0FE0或者CI1FE0的有效边沿或者捕获极性</p> <p>[CH0NP==0, CH0P==0]: 把CIxFE0的上升沿作为捕获或者从模式下触发的有效信号，并且CIxFE0不会被翻转。</p> <p>[CH0NP==0, CH0P==1]: 把CIxFE0的下降沿作为捕获或者从模式下触发的有效信号，并且CIxFE0会被翻转。</p> <p>[CH0NP==1, CH0P==0]: 保留。</p> <p>[CH0NP==1, CH0P==1]: 把CIxFE0的上升沿和下降沿都作为捕获或者从模式下触发的有效信号，并且CIxFE0不会被翻转。</p> <p>当TIMERx_CCHP寄存器的PROT [1:0]=11或10 时此位不能被更改。</p>
0	CH0EN	<p>通道0捕获/比较使能</p> <p>当通道0配置为输出模式时，将此位置1使能CH0_O信号有效。当通道0配置为输入模式时，将此位置1使能通道0上的捕获事件。</p> <p>0：禁止通道0</p> <p>1：使能通道0</p>

计数器寄存器 (TIMERx_CNT)(x=1)

地址偏移: 0x24

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

位/位域	名称	描述
31:0	CNT[31:0]	这些位是当前的计数值。写操作能改变计数器值。

计数器寄存器 (TIMERx_CNT)(x=2)

地址偏移: 0x24

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

rw

位/位域	名称	描述
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器 (TIMERx_PSC)

地址偏移: 0x28

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

rw

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

计数器自动重载寄存器 (TIMERx_CAR)(x=1)

地址偏移: 0x2C

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

CARL[15:0]

rw

位/位域	名称	描述
31:0	CARL[31:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

计数器自动重载寄存器 (TIMERx_CAR)(x=2)

地址偏移: 0x2C

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

位/位域	名称	描述
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

通道 0 捕获/比较值寄存器 (TIMERx_CH0CV) (x=1)

地址偏移: 0x34

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH0VAL[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															
rw															

位/位域	名称	描述
31:0	CH0VAL[31:0]	通道0的捕获或比较值 当通道0配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道0配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 0 捕获/比较值寄存器 (TIMERx_CH0CV)(x=2)

地址偏移: 0x34

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															
rw															

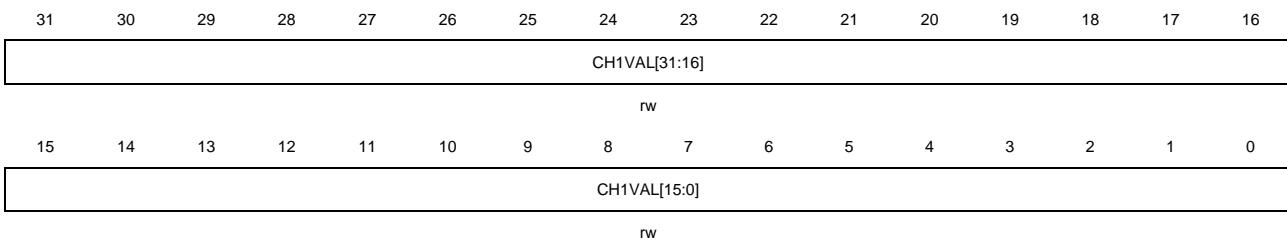
位/位域	名称	描述
15:0	CH0VAL[15:0]	<p>通道0的捕获或比较值</p> <p>当通道0配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。</p> <p>当通道0配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。</p>

通道 1 捕获/比较值寄存器 (**TIMERx_CH1CV**) (x=1)

地址偏移: 0x38

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问



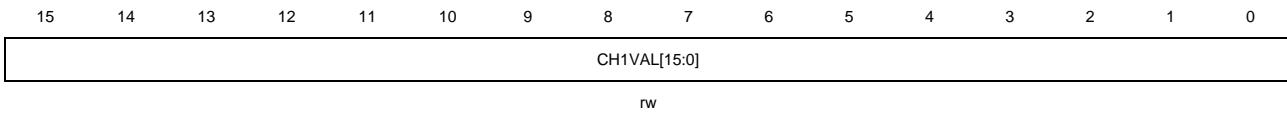
位/位域	名称	描述
31:0	CH1VAL[31:0]	<p>通道1的捕获或比较值</p> <p>当通道1配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。</p> <p>当通道1配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。</p>

通道 1 捕获/比较值寄存器 (**TIMERx_CH1CV**)(x=2)

地址偏移: 0x38

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问



位/位域	名称	描述
15:0	CH1VAL[15:0]	<p>通道1的捕获或比较值</p> <p>当通道1配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。</p> <p>当通道1配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器</p>

后，影子寄存器值随每次更新事件更新。

通道 2 捕获/比较值寄存器 (**TIMERx_CH2CV**) (x=1)

地址偏移: 0x3C

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH2VAL[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2VAL[15:0]															
rw															

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

通道 2 捕获/比较值寄存器 (**TIMERx_CH2CV**)(x=2)

地址偏移: 0x3C

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2VAL[15:0]															
rw															

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

通道 3 捕获/比较值寄存器 (**TIMERx_CH3CV**) (x=1)

地址偏移: 0x40

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3VAL[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3VAL[15:0]															
rw															

位/位域	名称	描述
31:0	CH3VAL[31:0]	<p>通道3的捕获或比较值</p> <p>当通道3配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。</p> <p>当通道3配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。</p>

通道 3 捕获/比较值寄存器 (TIMERx_CH3CV)(x=2)

地址偏移: 0x40

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3VAL[15:0]															
rw															

位/位域	名称	描述
15:0	CH3VAL[15:0]	<p>通道3的捕获或比较值</p> <p>当通道3配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。</p> <p>当通道3配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。</p>

DMA 配置寄存器 (TIMERx_DMACFG)

地址偏移: 0x48

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

位/位域	名称	描述
15:13	保留	必须保持复位值.
12:8	DMATC [4:0]	DMA传输计数 该位域定义了DMA访问（读写）TIMERx_DMAB寄存器的数量
7:5	保留	必须保持复位值
4:0	DMATA [4:0]	DMA传输起始地址 该位域定义了DMA访问TIMERx_DMAB寄存器的第一个地址。当通过TIMERx_DMA第一次访问时，访问的就是该位域指定的地址。第二次访问TIMERx_DMAB时，将访问起始地址+0x4。 5'b0_0000: TIMERx_CTL0 5'b0_0001: TIMERx_CTL1 ... 总之： 起始地址 = TIMERx_CTL0 + DMATA*4

DMA 发送缓冲区寄存器 (TIMERx_DMATB)

地址偏移: 0x4C

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

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

配置寄存器 (TIMERx_CFG)

地址偏移: 0xFC

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

位/位域	名称	描述
------	----	----

15:2	保留	必须保持复位值
1	CHVSEL	写捕获比较寄存器选择位 此位由软件写1或清0。 1: 当写入捕获比较寄存器的值与寄存器当前值相等时, 写入操作无效 0: 无影响
0	保留	必须保持复位值

16.3. 通用定时器 L2 (TIMERx, x=13)

16.3.1. 简介

通用定时器L2 (TIMERx, x=13)是单通道定时器, 支持输入捕获和输出比较, 产生PWM信号控制电机和电源管理。通用定时器L2含有一个16位无符号计数器。

通用定时器L2是可编程的, 可以被用来计数, 其外部事件可以驱动其他定时器

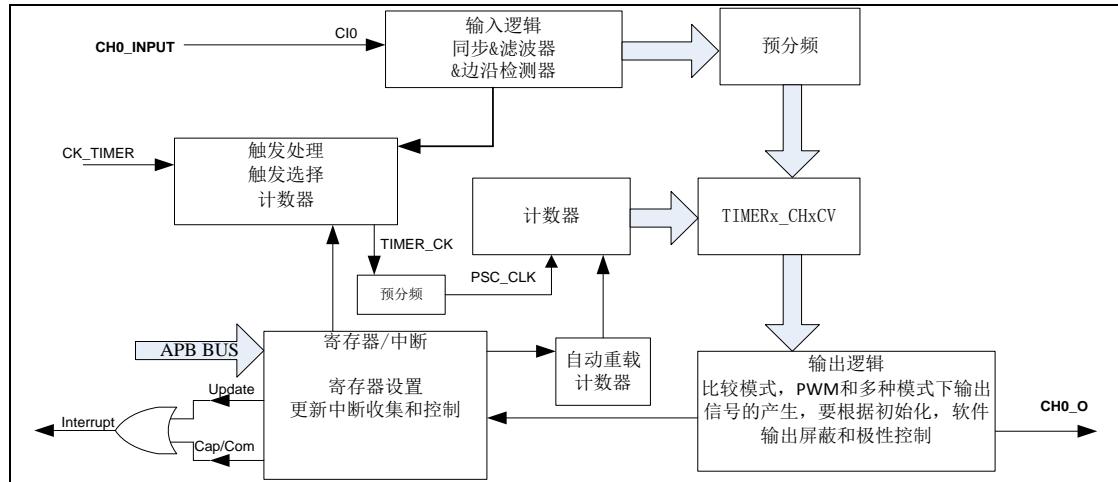
16.3.2. 主要特性

- 总通道数: 1
- 计数器宽度: 16位
- 时钟源可选: 内部时钟
- 计数模式: 向上计数
- 可编程的预分频器: 16位, 运行时可以被改变
- 每个通道可配置: 输入捕获模式, 输出比较模式, 可编程的PWM模式
- 自动重装载功能.
- 中断输出: 更新事件, 比较/捕获事件

16.3.3. 结构框图

[图 16-50. 通用定时器 L2 结构框图](#)提供了通用定时器 L2 的内部配置细节

图 16-50. 通用定时器 L2 结构框图



16.3.4. 功能描述

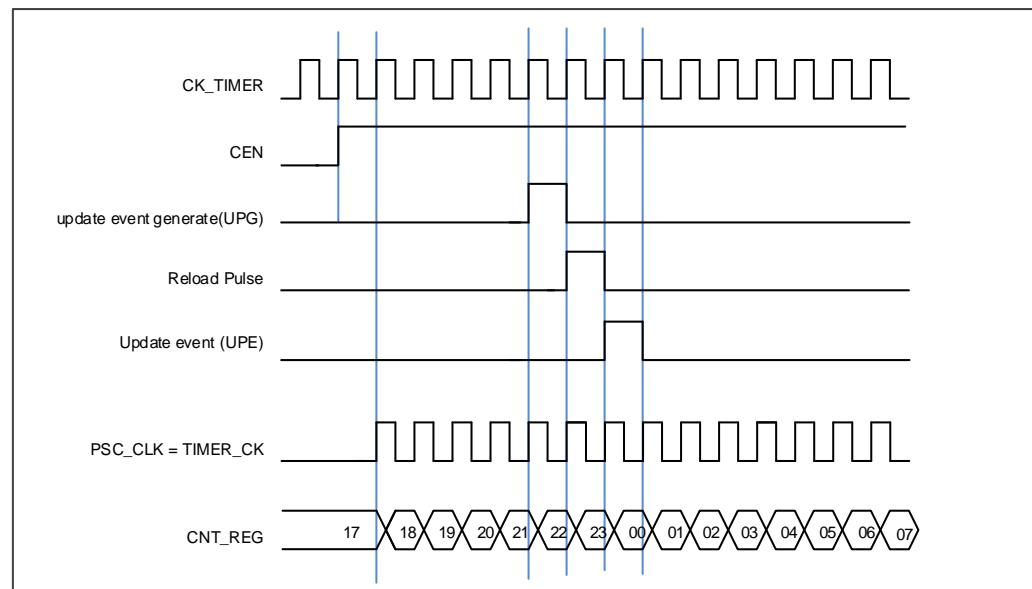
时钟源选择

通用定时器L2由内部时钟源TIMER_CK驱动

定时器内部时钟源TIMER_CK连接到RCU模块的CK_TIMER

通用定时器L2仅有一个时钟源CK_TIMER，用来驱动计数器预分频器。当CEN置位，CK_TIMER经过预分频器（预分频值由TIMERx_PSC寄存器确定）产生PSC_CLK。

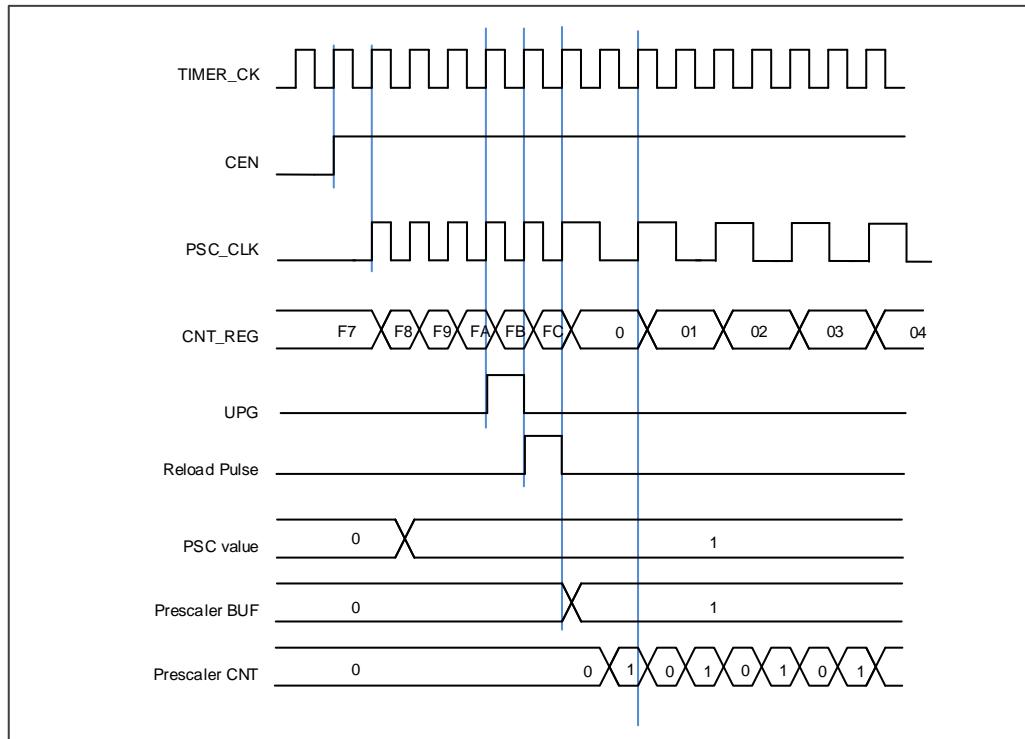
图 16-51. 内部时钟分频为 1 时正常模式下的控制电路



预分频器

预分频器可以将定时器的时钟 (TIMER_CK) 频率按1到65536之间的任意值分频，分频后的时钟PSC_CLK驱动计数器计数。分频系数受预分频寄存器TIMERx_PSC控制，这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 16-52. 当预分频器的参数从 1 变到 2 时，计数器的时序图



向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从0开始向上连续计数到自动加载值（定义在TIMERx_CAR寄存器中），一旦计数器计数到自动加载值，会重新从0开始向上计数并产生上溢事件。在向上计数模式中，TIMERx_CTL0寄存器中的计数方向控制位DIR应该被设置成0。

当通过TIMERx_SWEVG寄存器的UPG位置1来设置更新事件时，计数值会被清0，并产生更新事件。

如果TIMERx_CTL0寄存器的UPDIS置1，则禁止更新事件。

当发生更新事件时，所有的寄存器(重复计数器，自动重载寄存器，预分频寄存器)都将被更新。

[图 16-53. 向上计数时序图, PSC=0/1](#) 和 [图 16-54. 向上计数时序图, 在运行时改变 TIMERx_CAR 寄存器的值](#)给出了一些例子，当 TIMERx_CAR=0x63 时，计数器在不同预分频

因子下的行为。

图 16-53. 向上计数时序图, PSC=0/1

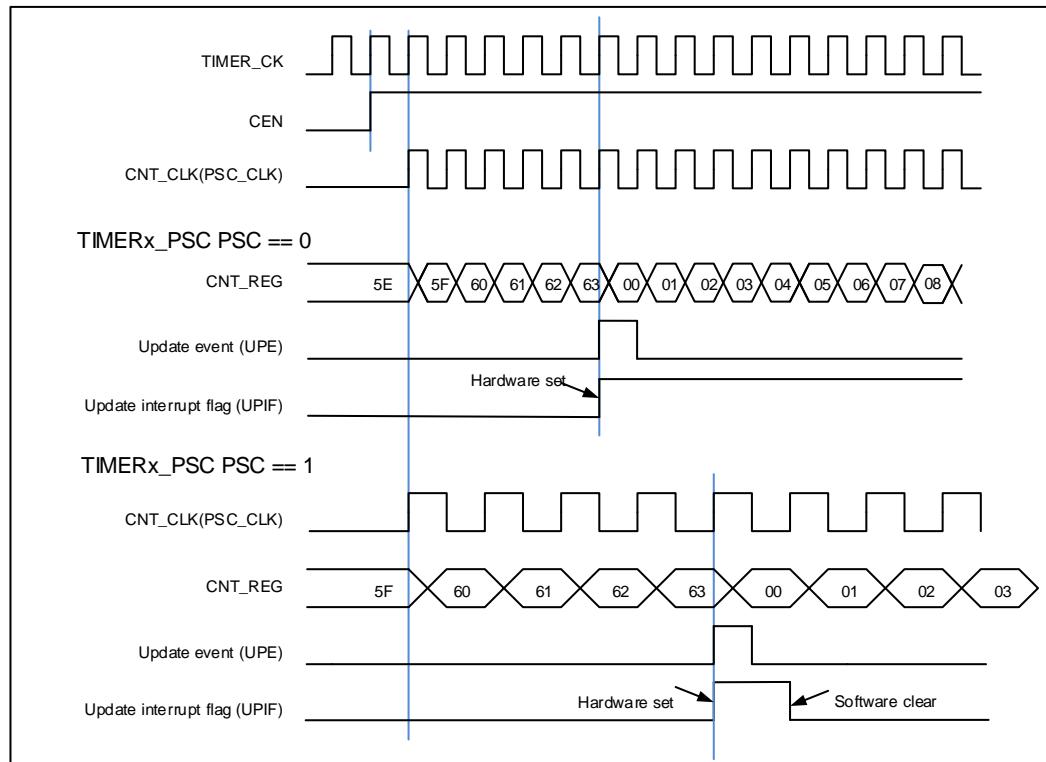
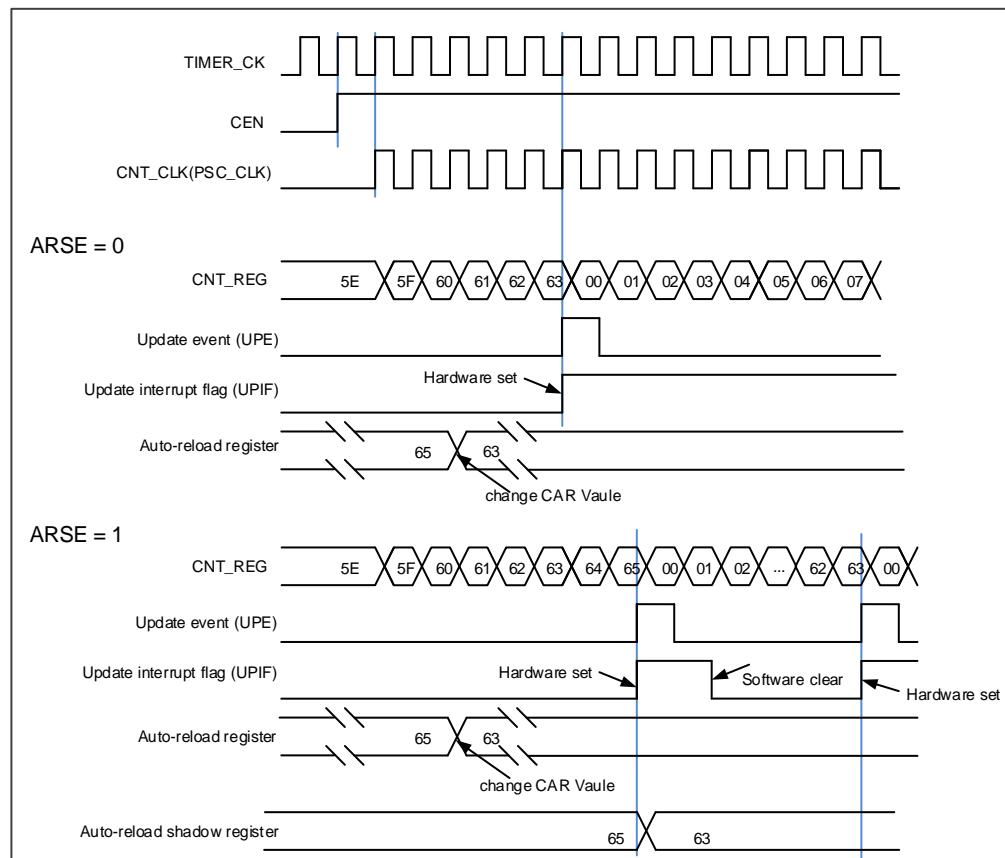


图 16-54. 向上计数时序图, 在运行时改变 TIMERx_CAR 寄存器的值



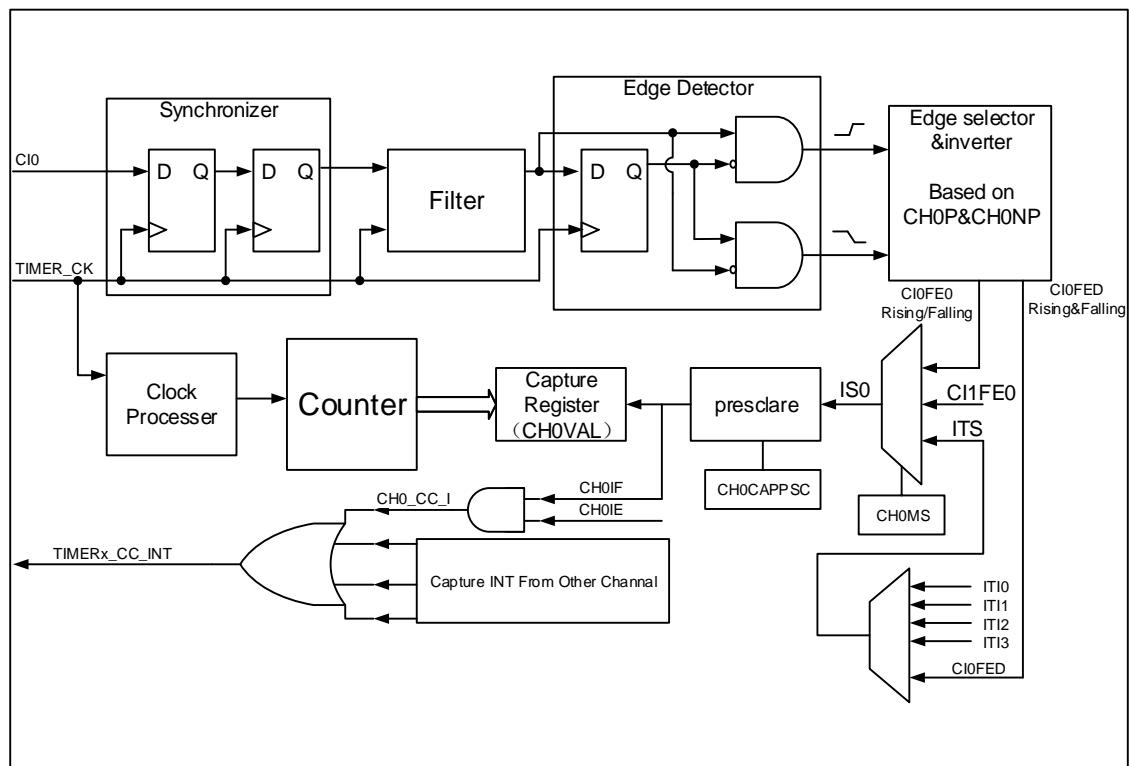
捕获/比较通道

通用定时器L2只有一个独立的通道用于捕获输入或比较输出是否匹配。该通道通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级。

■ 输入捕获模式

捕获模式允许通道测量一个波形时序，频率，周期，占空比等。输入级包括一个数字滤波器，一个通道极性选择，边沿检测和一个通道预分频器。如果在输入引脚上出现被选择的边沿，TIMERx_CHxCV寄存器会捕获计数器当前的值，同时CHxIF位被置1，如果CHxIE = 1则产生通道中断。

图 16-55. 输入捕获逻辑



通道输入信号CIx先被TIMER_CK信号同步，然后经过数字滤波器采样，产生一个被滤波后的信号。通过边沿检测器，可以选择检测上升沿或者下降沿。通过配置CHxP选择使用上升沿或者下降沿。配置CHxMS.，可以选择其他通道的输入信号，内部触发信号。配置IC预分频器，使得若干个输入事件后才产生一个有效的捕获事件。捕获事件发生，CHxVAL存储计数器的值。

配置步骤如下：

第一步：滤波器配置 (TIMERx_CHCTL0寄存器中CHxCAPFLT):

根据输入信号和请求信号的质量，配置相应的CHxCAPFLT。

第二步：边沿选择 (TIMERx_CHCTL2寄存器中CHxP/CHxNP):

配置CHxP/CHxNP选择上升沿或者下降沿。

第三步：捕获源选择 (TIMERx_CHCTL0寄存器中CHxMS):

一旦通过配置CHxMS选择输入捕获源，必须确保通道配置在输入模式 (CHxMS!=0x0)，而且

TIMERx_CHxCV寄存器不能再被写。

第四步：中断使能 (TIMERx_DMAINTEN寄存器中CHxIE):

使能相应中断，可以获得中断。

第五步：捕获使能 (TIMERx_CHCTL2寄存器中CHxEN)。

结果：当期望的输入信号发生时，TIMERx_CHxCV被设置成当前计数器的值，CHxIF为置1。如果CHxIF位已经为1，则CHxOF位置1。根据TIMERx_DMAINTEN寄存器中CHxIE的配置，相应的中断会被提出。

直接产生：软件设置CHxG位，会直接产生中断。

■ 输出比较模式

在输出比较模式，TIMERx可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的CHxVAL寄存器与计数器的值匹配时，根据CHxCOMCTL的配置，这个通道的输出可以被置高电平，被置低电平或者反转。当计数器的值与CHxVAL寄存器的值匹配时，CHxIF位被置1，如果CHxIE = 1则会产生中断，如果CHxDEN=1则会产生DMA请求。

配置步骤如下：

第一步：时钟配置：

配置定时器时钟源，预分频器等。

第二步：比较模式配置：

设置CHxCOMSEN位来配置输出比较影子寄存器；

设置CHxCOMCTL位来配置输出模式（置高电平/置低电平/反转）；

设置CHxP/CHxNP位来选择有效电平的极性；

设置CHxEN使能输出。

第三步：通过CHxIE位配置中断使能。

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

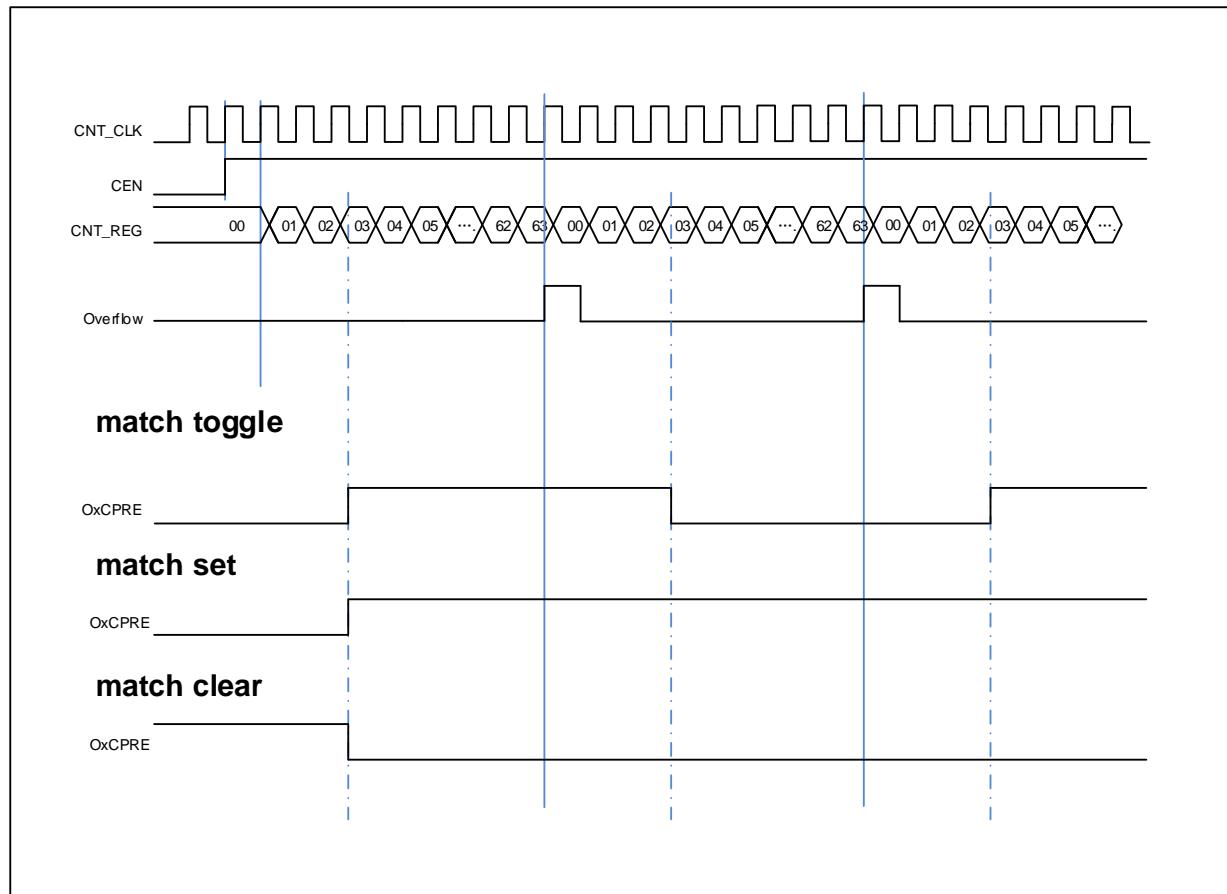
CHxVAL可以在运行时根据你所期望的波形而改变。

第五步：设置CEN位使能定时器。

[**图 16-56. 三种输出比较模式**](#)显示了三种比较输出模式：反转/置高电平/置低电平，

CAR=0x63, CHxVAL=0x3。

图 16-56. 三种输出比较模式



PWM 模式

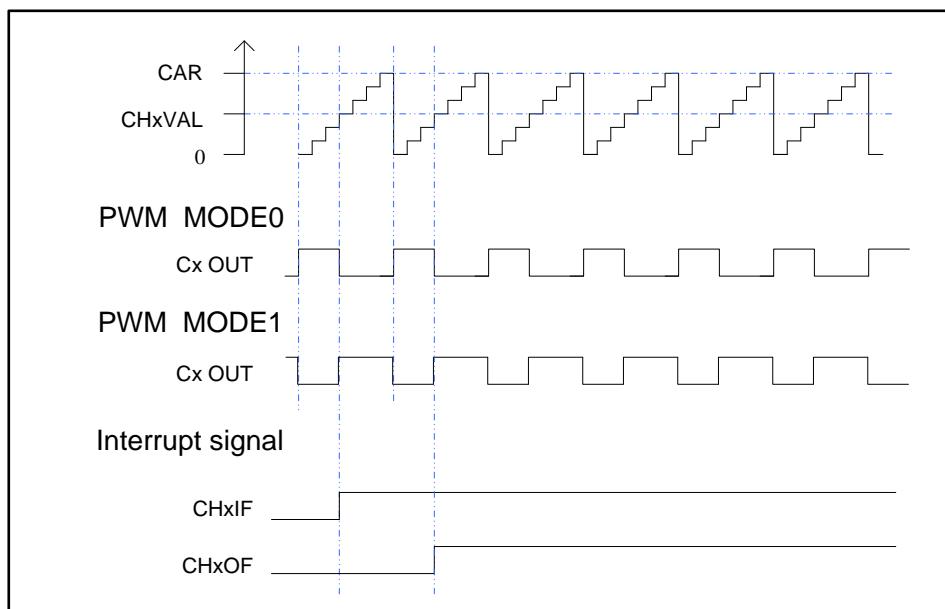
在 PWM 输出模式下 (PWM 模式 0 是配置 CHxCOMCTL 为 3'b110, PWM 模式 1 是配置 CHxCOMCTL 为 3'b111), 通道根据 TIMERx_CAR 寄存器和 TIMERx_CHxCV 寄存器的值, 输出 PWM 波形。

PWM 的周期由 TIMERx_CAR 寄存器值决定, 占空比由 TIMERx_CHxCV 寄存器值决定。图 16-57. PWM 时序图 显示了 EAPWM 的输出波形和中断。

在 PWM0 模式下 (CHxCOMCTL==3'b110), 如果 TIMERx_CHxCV 寄存器的值大于 TIMERx_CAR 寄存器的值, 通道输出一直为有效电平。

在 PWM0 模式下 (CHxCOMCTL==3'b110), 如果 TIMERx_CHxCV 寄存器的值等于 0, 通道输出一直为无效电平。

图 16-57. PWM 时序图



通道输出参考信号

当TIMERx用于输出匹配比较模式下，设置CHxCOMCTL位可以定义OxCPRE信号(通道x准备信号)类型。OxCPRE信号有若干类型的输出功能，包括，设置CHxCOMCTL=0x00可以保持原始电平；设置CHxCOMCTL=0x01可以将OxCPRE信号设置为高电平；设置CHxCOMCTL=0x02可以将OxCPRE信号设置为低电平；设置CHxCOMCTL=0x03，在计数器值和TIMERx_CHxCV寄存器的值匹配时，可以翻转输出信号。

PWM模式0和PWM模式1是OxCPRE的另一种输出类型，设置CHxCOMCTL位域位0x06或0x07可以配置PWM模式0/PWM模式1。在这些模式中，根据计数器值和TIMERx_CHxCV寄存器值的关系以及计数方向，OxCPRE信号改变其电平。具体细节描述，请参考相应的位。

设置CHxCOMCTL =0x04或0x05可以实现OxCPRE信号的强制输出功能。输出比较信号能够直接由软件强置为有效或无效状态，而不依赖于TIMERx_CHxCV的值和计数器值之间间的比较结果。

设置CHxCOMCEN=1，当由外部ETI引脚信号产生的ETIFE信号为高电平时，OxCPRE被强制为低电平。在下一次更新事件到来时，OxCPRE信号才会回到有效电平状态。

定时器调试模式

当Cortex™-M4内核停止，DBG_CTL0寄存器中的TIMERx_HOLD配置位被置1，定时器计数器停止。

16.3.5. TIMERx 寄存器(x=13)

控制寄存器 0 (TIMERx_CTL0)

地址偏移: 0x00

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				CKDIV[1:0]		ARSE		保留		UPS		UPDIS		CEN	
						rw		rw			rw		rw		rw

位/位域	名称	描述
15:10	保留	必须保持复位值
9:8	CKDIV[1:0]	<p>时钟分频</p> <p>通过软件配置CKDIV，规定定时器时钟(TIMER_CK) 与死区时间和采样时钟(DTS)之间的分频系数，死区发生器和数字滤波器会用到DTS时间。</p> <ul style="list-style-type: none"> 00: $f_{DTS}=f_{TIMER_CK}$ 01: $f_{DTS}= f_{TIMER_CK} /2$ 10: $f_{DTS}= f_{TIMER_CK} /4$ 11: 保留
7	ARSE	<p>自动重载影子使能</p> <p>0: 禁能TIMERx_CAR寄存器的影子寄存器</p> <p>1: 使能TIMERx_CAR寄存器的影子寄存器</p>
6:3	保留	必须保持复位值
2	UPS	<p>更新请求源</p> <p>软件配置该为，选择更新事件源.</p> <p>0: 使能后，下述任一事件产生更新中断或DMA请求:</p> <ul style="list-style-type: none"> - UPG位被置1 - 计数器溢出/下溢 - 溢从模式控制器产生的更新 <p>1: 使能后只有计数器溢出/ 下溢才产生更新中断或DMA请求。</p>
1	UPDIS	<p>禁止更新.</p> <p>该位用来使能或禁能更新事件的产生 .</p> <p>0: 更新事件使能.当以下事件之一发生时，更新事件产生，具有缓存的寄存器被装入它们的预装载值:</p> <ul style="list-style-type: none"> - 以下事件位被置1 - 计数器溢出/下溢 - 溢从模式控制器产生一个更新事件 <p>1: 更新事件禁能. 带有缓存的寄存器保持原有值，如果UPG位被置1或者从模式控制器产生一个硬件复位事件，计数器和预分频器被重新初始化</p>
0	CEN	<p>计数器使能</p> <p>0: 计数器禁能</p> <p>1: 计数器使能</p> <p>在软件将CEN位置1后，外部时钟、暂停模式和编码器模式才能工作。触发模式可以自动</p>

地通过硬件设置CEN位。

DMA 和中断使能寄存器 (TIMERx_DMAINTEN)

地址偏移: 0x0C

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

位/位域	名称	描述
15:2	保留	必须保持复位值
1	CHOIE	通道0比较/捕获中断使能 0: 禁止通道0中断 1: 使能通道0中断
0	UPIE	更新中断使能 0: 禁止更新中断 1: 使能更新中断

中断标志寄存器 (TIMERx_INTF)

地址偏移: 0x10

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				CH0OF		保留				CHOIF				UPIF	
rc_w0														rc_w0	rc_w0

位/位域	名称	描述
15:10	保留	必须保持复位值
9	CH0OF	通道0捕获溢出标志 当通道0被配置为输入模式时，在CHOIF标志位已经被置1后，捕获事件再次发生时，该标志位可以由硬件置1。该标志位由软件清0。 0: 无捕获溢出中断发生 1: 发生了捕获溢出中断
8:2	保留	必须保持复位值
1	CHOIF	通道0比较/捕获中断标志 此标志由硬件置1软件清0。当通道0在输入模式下时，捕获事件发生时此标志位被置1；

当通道0在输出模式下时，此标志位在一个比较事件发生时被置1。

0: 无通道0中断发生

1: 通道0中断发生

0	UPIF	更新中断标志
---	------	--------

此位在任何更新事件发生时由硬件置1，软件清0。

0: 无更新中断发生

1: 发生更新中断

软件事件产生寄存器 (TIMERx_SWEVG)

地址偏移: 0x14

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														CH0G	UPG
w														w	w

位/位域	名称	描述
15:2	保留	必须保持复位值.
1	CH0G	<p>通道0捕获或比较事件发生</p> <p>该位由软件置1，用于在通道0产生一个捕获/比较事件，由硬件自动清0。当此位被置1，CH0IF标志位被置1，若开启对应的中断和DMA，则发出相应的中断和DMA请求。此外，如果通道0配置为输入模式，计数器的当前值被TIMERx_CH0CV寄存器捕获，如果CH0OF标志位已经为1，则CH0OF标志位被置1。</p> <p>0: 不产生通道0捕获或比较事件</p> <p>1: 发生通道0捕获或比较事件</p>
0	UPG	<p>更新事件产生</p> <p>此位由软件置1，被硬件自动清0。当此位被置1并且向上计数模式，计数器被清0，预分频计数器将同时被清除。</p> <p>0: 无更新事件产生</p> <p>1: 产生更新事件</p>

通道控制寄存器 0 (TIMERx_CHCTL0)

地址偏移: 0x18

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				保留				CH0COMCTL[2:0]		CH0COMSEN		CH0COMFEN		CH0MS[1:0]	

	CH0CAPFLT[3:0]	CH0CAPPSC[1:0]
	RW	RW

输出比较模式:

位/位域	名称	描述
15:7	保留	必须保持复位值.
6:4	CH0COMCTL[2:0]	<p>通道0输出比较模式</p> <p>此位定义了输出参考信号O0CPRE的动作，而O0CPRE决定了CH0_O、CH0_ON的值。O0CPRE高电平有效，而CH0_O、CH0_ON的有效电平取决于CH0P、CH0NP位。</p> <p>000: 冻结。输出比较寄存器TIMERx_CH0CV与计数器TIMERx_CNT间的比较对O0CPRE不起作用</p> <p>001: 匹配时设置为高。当计数器的值与捕获/比较值寄存器TIMERx_CH0CV相同时，强制O0CPRE为高。</p> <p>010: 匹配时设置为低。当计数器的值与捕获/比较值寄存器TIMERx_CH0CV相同时，强制O0CPRE为低。</p> <p>011: 匹配时翻转。当计数器的值与捕获/比较值寄存器TIMERx_CH0CV相同时，强制O0CPRE翻转。</p> <p>100: 强制为低。强制O0CPRE为低电平</p> <p>101: 强制为高。强制O0CPRE为高电平</p> <p>110: PWM 模式0 (EAPWM)。在向上计数时，一旦计数器值小于TIMERx_CH0CV时，O0CPRE为高电平，否则为低电平。在向下计数时，一旦计数器的值大于TIMERx_CH0CV时，O0CPRE 为低电平，否则为高电平。</p> <p>111: PWM 模式1(CAPWM)。在向上计数时，一旦计数器值小于TIMERx_CH0CV时，O0CPRE为低电平，否则为高电平。在向下计数时，一旦计数器的值大于TIMERx_CH0CV时，O0CPRE为高电平，否则为低电平。</p> <p>在PWM模式0或PWM模式1中，只有当比较结果改变了或者输出比较模式中从冻结模式切换到PWM模式时，CxCOMR电平才改变。</p> <p>当TIMERx_CCHP寄存器的PROT [1:0]=11且CH0MS =00（比较模式）时此位不能被改变。</p>
3	CH0COMSEN	<p>通道0输出比较影子寄存器使能</p> <p>当此位被置1，TIMERx_CH0CV寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。</p> <p>0: 禁止通道0输出/比较影子寄存器</p> <p>1: 使能通道0输出/比较影子寄存器</p> <p>仅在单脉冲模式下(TIMERx_CTL0寄存器的SPM =1)，可以在未确认预装载寄存器情况下使用PWM 模式</p> <p>当TIMERx_CCHP寄存器的PROT [1:0]=11且CH0MS =00时此位不能被改变。</p>
2	CH0COMFEN	<p>通道0输出比较快速使能</p> <p>当该位为1时，如果通道配置为PWM0模式或者PWM1模式，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，</p>

CH0_O被设置为比较电平而与比较结果无关。

0: 禁止通道0输出比较快速。当触发器的输入有一个有效沿时，激活**CH0_O**输出的最小延时为5个时钟周期

1: 使能通道0输出比较快速。当触发器的输入有一个有效沿时，激活**CH0_O**输出的最小延时为3个时钟周期

1:0	CH0MS[1:0]	<p>通道0 I/O模式选择</p> <p>这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2寄存器的CH0EN位被清0)时这些位才可写。</p> <p>00: 通道0配置为输出</p> <p>01: 通道0配置为输入，ISO映射在CIOFE0上</p> <p>10: 通道0配置为输入，ISO映射在CIOFE1上</p> <p>11: 通道0配置为输入，ISO映射在ITS上。此模式仅工作在内部触发输入被选中时(通过设置TIMERx_SMCFGFG寄存器的TRGS位)</p>
-----	-------------------	---

输入捕获模式:

位/位域	名称	描述
15:8	保留	必须保持复位值.
7:4	CH0CAPFLT[3:0]	<p>通道0输入捕获滤波控制</p> <p>数字滤波器由一个事件计数器组成，它记录N个输入事件后会产生一个输出的跳变。这些位定义了CIO输入信号的采样频率和数字滤波器的长度。</p> <p>0000: 无滤波器, fSAMP= fDTS, N=1</p> <p>0001: fSAMP= fPCLK, N=2</p> <p>0010: fSAMP= fPCLK, N=4</p> <p>0011: fSAMP= fPCLK, N=8</p> <p>0100: fSAMP=fDTS/2, N=6</p> <p>0101: fSAMP=fDTS/2, N=8</p> <p>0110: fSAMP=fDTS/4, N=6</p> <p>0111: fSAMP=fDTS/4, N=8</p> <p>1000: fSAMP=fDTS/8, N=6</p> <p>1001: fSAMP=fDTS/8, N=8</p> <p>1010: fSAMP=fDTS/16, N=5</p> <p>1011: fSAMP=fDTS/16, N=6</p> <p>1100: fSAMP=fDTS/16, N=8</p> <p>1101: fSAMP=fDTS/32, N=5</p> <p>1110: fSAMP=fDTS/32, N=6</p> <p>1111: fSAMP=fDTS/32, N=8</p>
3:2	CH0CAPPSC[1:0]	<p>通道0输入捕获预分频器</p> <p>这2位定义了通道0输入的预分频系数。当TIMERx_CHCTL2寄存器中的CH0EN =0时，则预分频器复位。</p> <p>00: 无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获</p> <p>01: 每2个事件触发一次捕获</p> <p>10: 每4个事件触发一次捕获</p>

11: 每8个事件触发一次捕获

1:0 CH0MS[1:0] 通道0模式选择
与输出比较模式相同

通道控制寄存器 2 (TIMERx_CHCTL2)

地址偏移: 0x20

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留..												CH0NP	保留	CH0P	CH0EN
rw												rw	rw	rw	

位/位域	名称	描述
15:4	保留	必须保持复位值.
3	CH0NP	<p>通道0互补输出极性</p> <p>当通道0配置为输出模式，此位定义了互补输出信号的极性。</p> <p>0: 通道0高电平有效</p> <p>1: 通道0低电平有效</p> <p>当通道0配置为输入模式时，此位和CH0P联合使用，作为输入信号CIO的极性选择控制信号。</p> <p>当TIMERx_CCHP寄存器的PROT [1:0]=11或10 时此位不能被更改。</p>
2	保留	必须保持复位值.
1	CH0P	<p>通道0极性</p> <p>当通道0配置为输出模式时，此位定义了输出信号极性。</p> <p>0: 通道0高电平有效</p> <p>1: 通道0低电平有效</p> <p>当通道0配置为输入模式时，此位定义了CIO信号极性</p> <p>[CH0NP, CH0P] 将选择CI0FE0或者CI1FE0的有效边沿或者捕获极性</p> <p>[CH0NP==0, CH0P==0]: 把CIxFEO的上升沿作为捕获或者从模式下触发的有效信号，并且CIxFEO不会被翻转。</p> <p>[CH0NP==0, CH0P==1]: 把CIxFEO的下降沿作为捕获或者从模式下触发的有效信号，并且CIxFEO会被翻转。</p> <p>[CH0NP==1, CH0P==0]: 保留。</p> <p>[CH0NP==1, CH0P==1]: 把CIxFEO的上升沿和下降沿都作为捕获或者从模式下触发的有效信号，并且CIxFEO不会被翻转。</p> <p>当TIMERx_CCHP寄存器的PROT [1:0]=11或10 时此位不能被更改。</p>
0	CH0EN	<p>通道0捕获/比较使能</p> <p>当通道0配置为输出模式时，将此位置1使能CH0_O信号有效。当通道0配置为输入模式时，将此位置1使能通道0上的捕获事件。</p>

0: 禁止通道0

1: 使能通道0

计数器寄存器 (TIMERx_CNT)

地址偏移: 0x24

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

位/位域	名称	描述
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器 (TIMERx_PSC)

地址偏移: 0x28

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

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

计数器自动重载寄存器 (TIMERx_CAR)

地址偏移: 0x2C

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

位/位域	名称	描述
15:0	CARL[15:0]	计数器自动重载值

这些位定义了计数器的自动重载值。

通道 0 捕获/比较值寄存器 (TIMERx_CH0CV)

地址偏移: 0x34

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															
rw															

位/位域	名称	描述
15:0	CH0VAL[15:0]	<p>通道0的捕获或比较值</p> <p>当通道0配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。</p> <p>当通道0配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。</p>

通道输入重映射寄存器(TIMERx_IRMP)

地址偏移: 0x50

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														CI0_RMP[1:0]	
rw															

位/位域	名称	描述
15:2	保留	必须保持复位值.
1:0	CI0_RMP[1:0]	<p>通道0输入重映射</p> <p>00:通道0输入连接到 GPIO(TIMER13_CH0)</p> <p>01: 通道0输入连接到 RTCCLK</p> <p>10: 通道0输入连接到 HXTAL/32 clock</p> <p>11: 通道0输入连接到 CKOUTSEL</p>

配置寄存器 (TIMERx_CFG)

地址偏移: 0xFC

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

位/位域	名称	描述
15:2	保留	必须保持复位值
1	CHVSEL	<p>写捕获比较寄存器选择位 此位由软件写1或清0。</p> <p>1: 当写入捕获比较寄存器的值与寄存器当前值相等时, 写入操作无效</p> <p>0: 无影响</p>
0	保留	必须保持复位值

16.4. 通用定时器 L3 (TIMERx,x=14)

16.4.1. 简介

通用定时器L3 (TIMER14) 是两通道定时器, 支持输入捕获和输出比较。可以产生PWM信号控制电机和电源管理。通用定时器L3含有一个16位无符号计数器。

通用定时器L3是可编程的, 可以被用来计数, 其外部事件可以驱动其他定时器

通用定时器L3包含了一个死区时间插入模块, 非常适合电机控制。

定时器和定时器之间是相互独立, 但是他们可以被同步在一起形成一个更大的定时器, 这些定时器的计数器一致地增加。

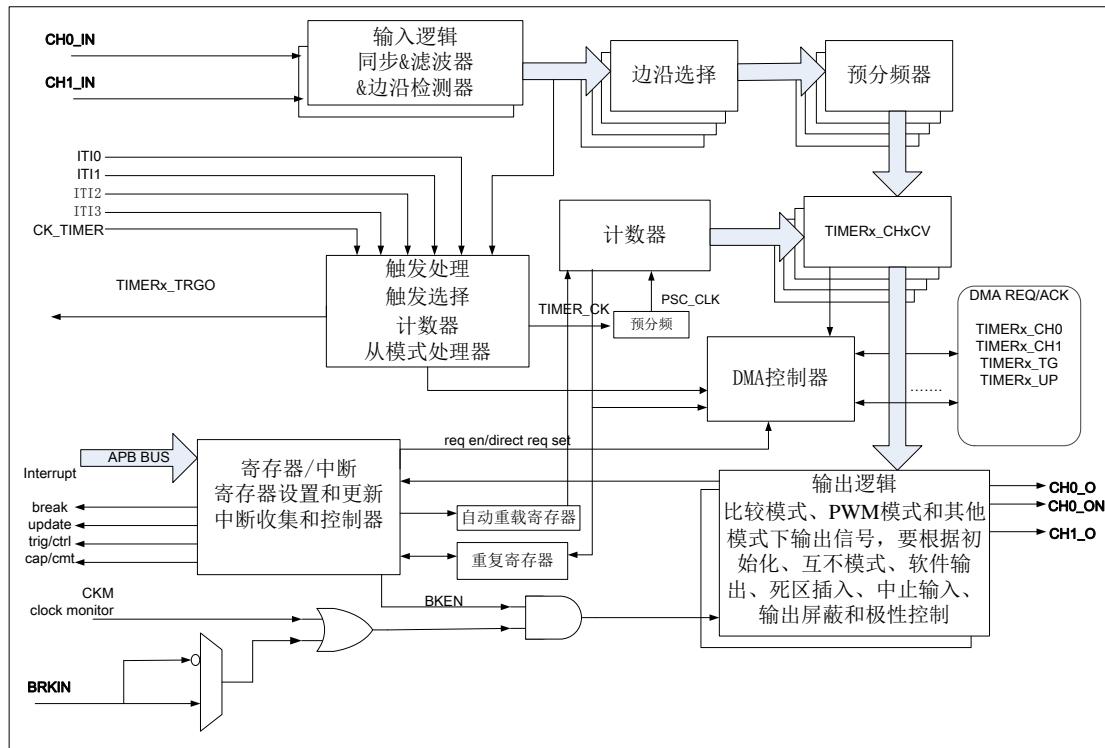
16.4.2. 主要特性

- 总通道数: 2;
- 计数器宽度: 16位;
- 时钟源可选: 内部时钟, 内部触发, 外部输入;
- 计数模式: 向上计数;
- 可编程的预分频器: 16位, 运行时可以被改变;
- 每个通道可配置: 输入捕获模式, 输出比较模式, 可编程的PWM模式, 单脉冲模式;
- 可编程的死区时间;
- 自动重装载功能;
- 可编程的计数器重复功能;
- 中止输入功能;
- 中断输出和DMA请求: 更新事件, 比较/捕获事件和中止事件;
- 多个定时器的菊花链使得一个定时器可以同时启动多个定时器;
- 定时器的同步允许被选择的定时器在同一个时钟周期开始计数;
- 定时器主/从模式控制器。

16.4.3. 结构框图

[图 16-58. 通用定时器 L3 结构框图](#)提供了通用定时器 L3 的内部配置细节

图 16-58. 通用定时器 L3 结构框图



16.4.4. 功能描述

时钟源选择

通用定时器L3可以由内部时钟源TIMER_CK或者由SMC (TIMERx_SMCFG寄存器位[2:0]) 控制的复用时钟源驱动。

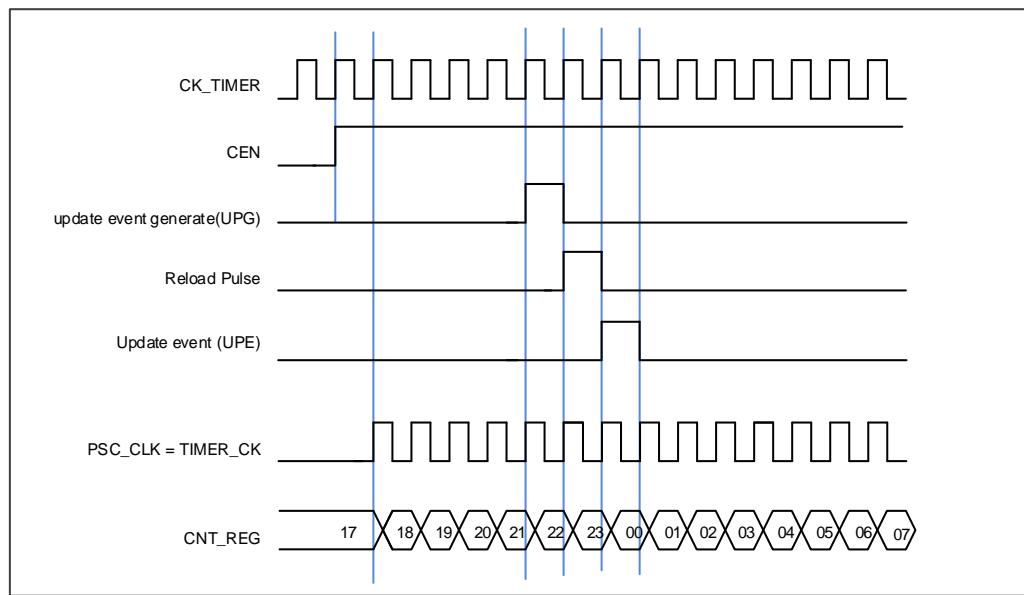
- SMC[2:0]==3'b000, 定时器选择内部时钟源 (连接到RCU模块的CK_TIMER)

如果禁能从模式 (SMC[2:0]==3'b000), 默认用来驱动计数器预分频器的是内部时钟源 CK_TIMER。当CEN置位, CK_TIMER经过预分频器 (预分频值由TIMERx_PSC寄存器确定) 产生PSC_CLK。

这种模式下, 驱动预分频器计数的TIMER_CK等于来自于RCU模块的CK_TIMER

如果使能从模式控制器(将TIMERx_SMCFG寄存器的SMC[2:0]设置为包括0x7, 预分频器被其他时钟源(由TIMERx_SMCFG寄存器的TRGS [2:0]区域选择)驱动, 在下文说明。当从模式选择位SMC被设置为0x4、0x5和0x6, 计数器预分频器时钟源由内部时钟TIMER_CK驱动。

图 16-59. 内部时钟分频为 1 时正常模式下的控制电路



- SMC[2:0]==3'b111(外部时钟模式0)，定时器选择外部输入引脚作为时钟源

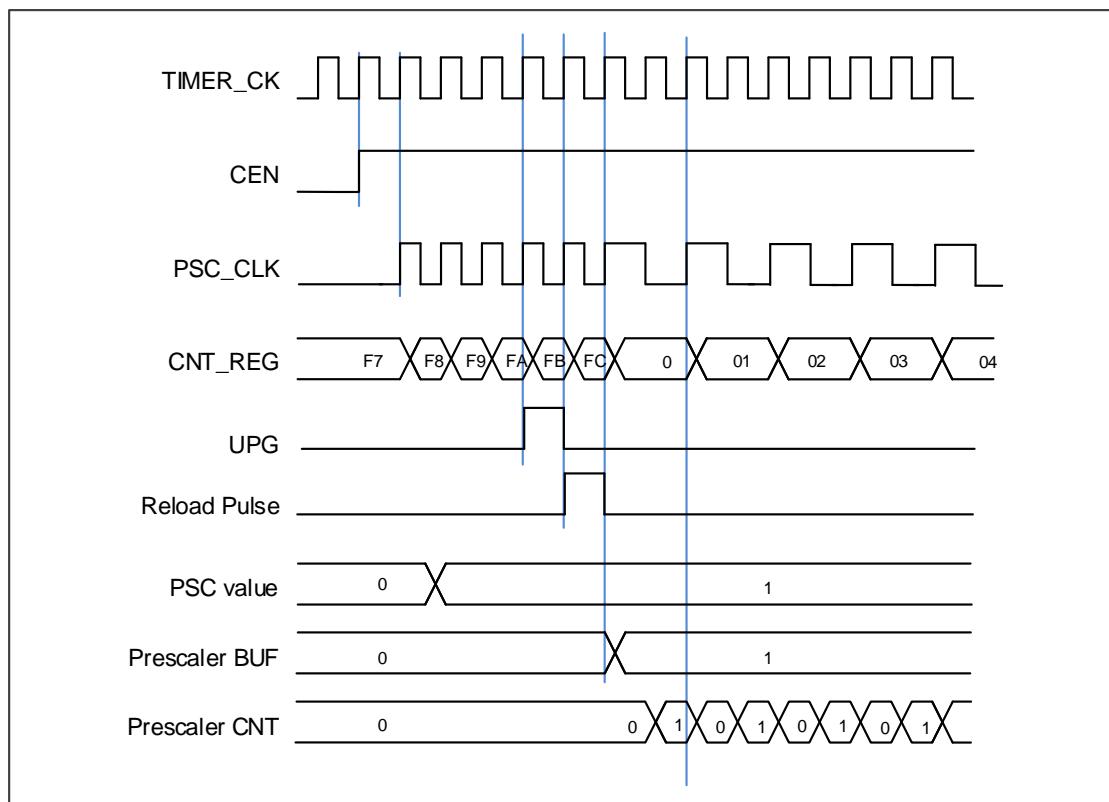
计数器预分频器可以在TIMERx_CI0/TIMERx_CI1引脚的每个上升沿或下降沿计数。这种模式可以通过设置SMC [2:0]为0x7同时设置TRGS[2:0]为0x4, 0x5或0x6来选择。

计数器预分频器也可以在内部触发信号ITI0/1/2/3的上升沿计数。这种模式可以通过设置SMC [2:0]为0x7同时设置TRGS [2:0]为0x0, 0x1, 0x2 或者0x3。

预分频器

预分频器可以将定时器的时钟 (TIMER_CK) 频率按1到65536之间的任意值分频，分频后的时钟PSC_CLK驱动计数器计数。分频系数受预分频寄存器TIMERx_PSC控制，这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 16-60. 当预分频器的参数从 1 变到 2 时，计数器的时序图



向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从0开始向上连续计数到自动加载值（定义在**TIMERx_CAR**寄存器中），一旦计数器计数到自动加载值，会重新从0开始向上计数。如果设置了重复计数器，在(**TIMERx_CREP+1**)次上溢后产生更新事件，否则在每次上溢时都会产生更新事件。在向上计数模式中，**TIMERx_CTL0**寄存器中的计数方向控制位**DIR**应该被设置成0。

当通过**TIMERx_SWEVG**寄存器的**UPG**位置1来设置更新事件时，计数值会被清0，并产生更新事件。

如果**TIMERx_CTL0**寄存器的**UPDIS**置1，则禁止更新事件。

当发生更新事件时，所有的寄存器(重复计数器，自动重载寄存器，预分频寄存器)都将被更新。

下面这些图给出了一些例子，当**TIMERx_CAR=0x63**时，计数器在不同预分频因子下的行为。

图 16-61. 向上计数时序图, PSC=0/1

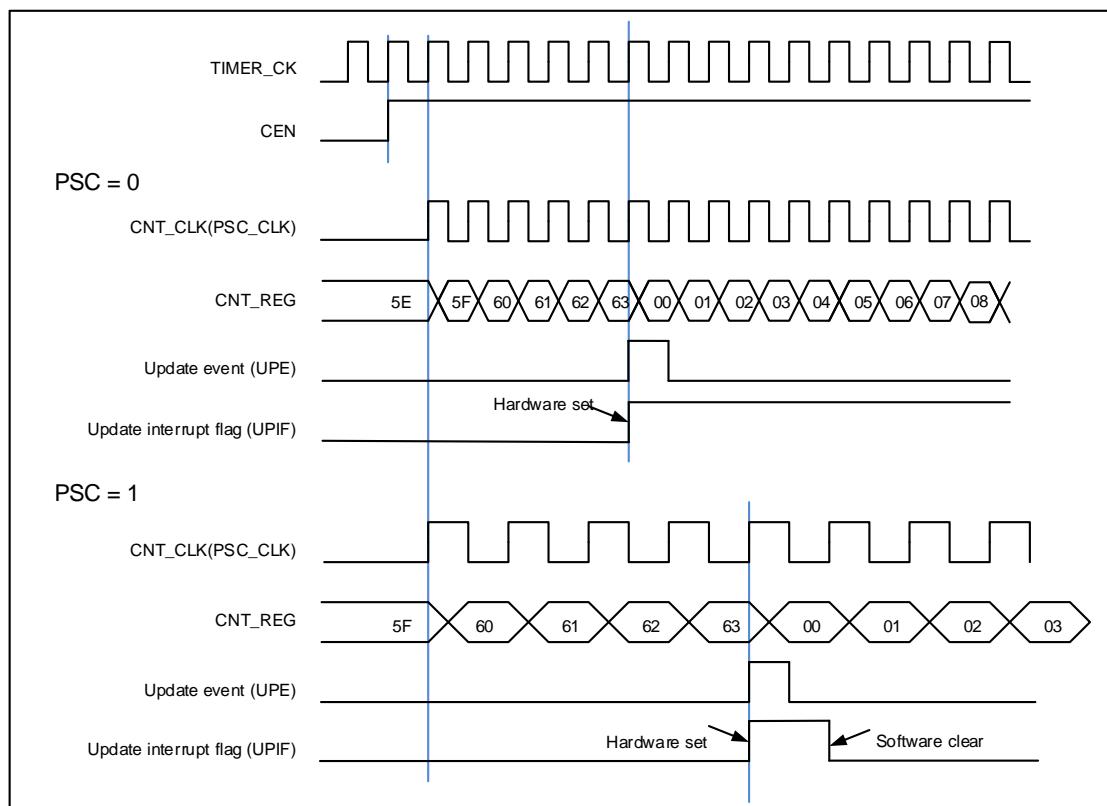
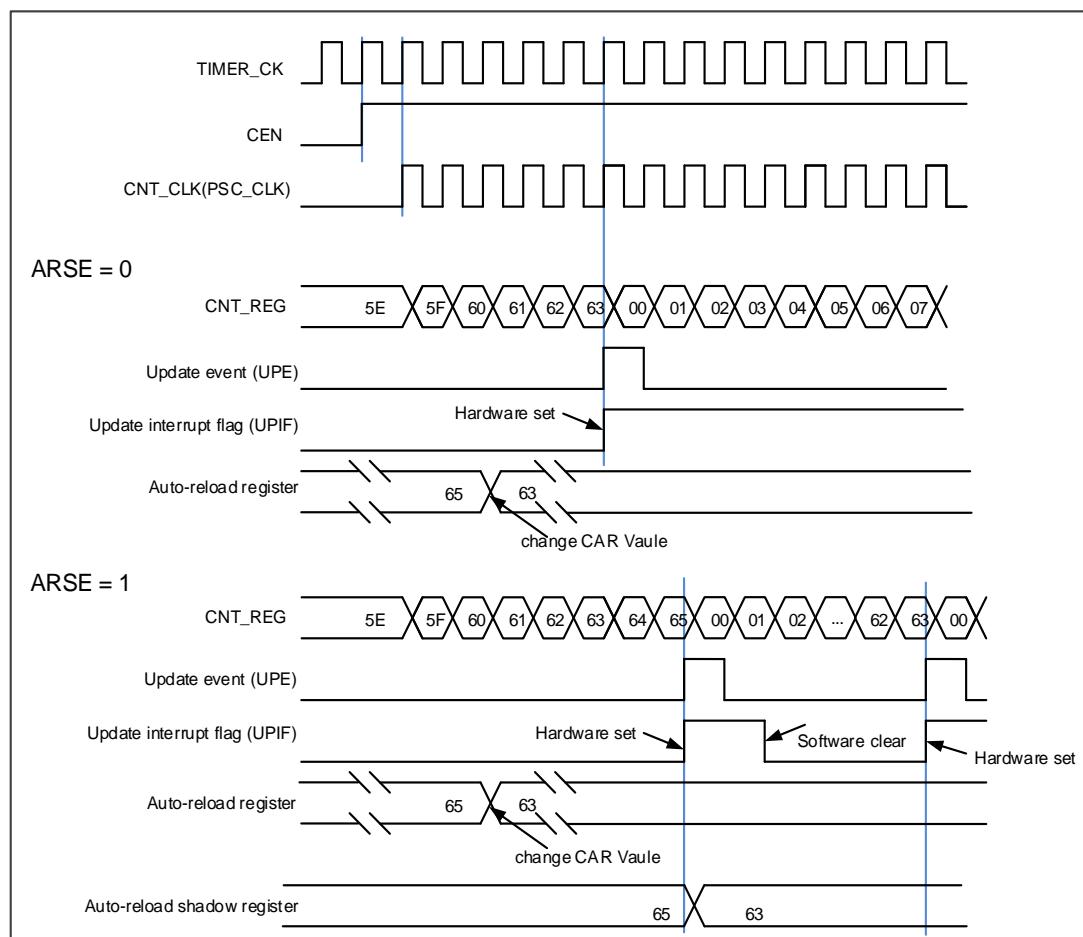


图 16-62. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值

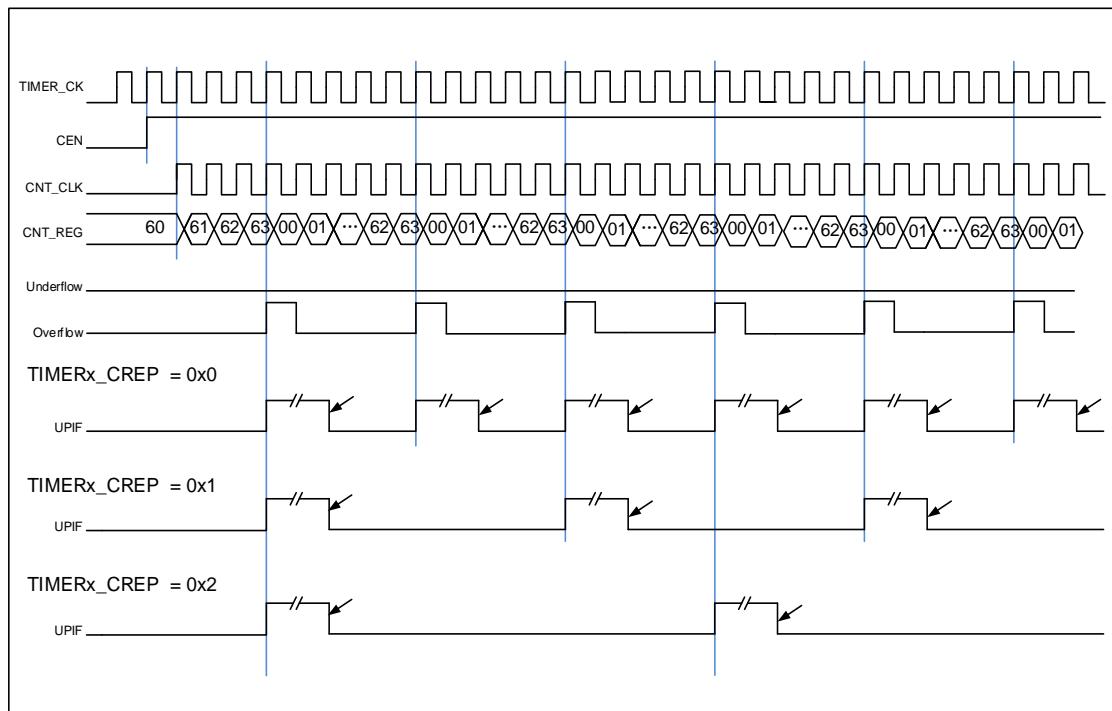


重复计数器

重复计数器是用来在N+1个计数周期之后产生更新事件，更新定时器的寄存器，N为TIMERx_CREP寄存器的CREP。向上计数模式下，重复计数器在每次计数器上溢时递减。

将TIMERx_SWEVG寄存器的UPG位置1可以重载TIMERx_CREP寄存器中CREP的值并产生一个更新事件。

图 16-63. 在向上计数模式下计数器重复时序图



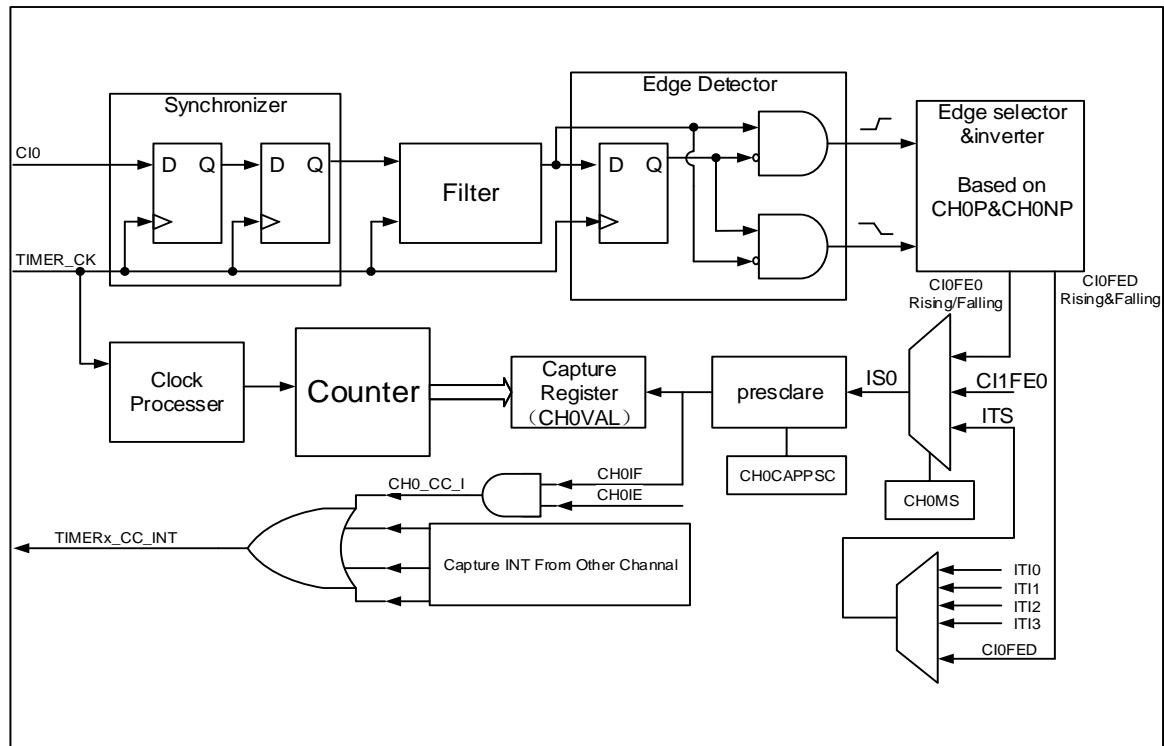
捕获/比较通道

通用定时器L3拥有两个独立的通道用于捕获输入或比较输出是否匹配。每个通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级。

■ 输入捕获模式

捕获模式允许通道测量一个波形时序，频率，周期，占空比等。输入级包括一个数字滤波器，一个通道极性选择，边沿检测和一个通道预分频器。如果在输入引脚上出现被选择的边沿，**TIMERx_CHxCV**寄存器会捕获计数器当前的值，同时**CHxIF**位被置1，如果**CHxIE = 1**则产生通道中断。

图 16-64. 输入捕获逻辑



通道输入信号 CI_x 有两种选择，一种是 $TIMER_{x_CHx}$ 信号，另一种是 $TIMER_{x_CH0}, TIMER_{x_CH1}$ 和 $TIMER_{x_CH2}$ 异或之后的信号。通道输入信号 CI_x 先被 $TIMER_{CK}$ 信号同步，然后经过数字滤波器采样，产生一个被滤波后的信号。通过边沿检测器，可以选择检测上升沿或者下降沿。通过配置 CH_{xP} 选择使用上升沿或者下降沿。配置 CH_{xMS} ，可以选择其他通道的输入信号，内部触发信号。配置 IC 预分频器，使得若干个输入事件后才产生一个有效的捕获事件。捕获事件发生， CH_{xVAL} 存储计数器的值。

配置步骤如下：

第一步：滤波器配置（ $TIMER_{x_CHCTL0}$ 寄存器中 $CH_{xCAPFLT}$ ）：

根据输入信号和请求信号的质量，配置相应的 $CH_{xCAPFLT}$ 。

第二步：边沿选择（ $TIMER_{x_CHCTL2}$ 寄存器中 CH_{xP}/CH_{xNP} ）：

配置 CH_{xP}/CH_{xNP} 选择上升沿或者下降沿。

第三步：捕获源选择（ $TIMER_{x_CHCTL0}$ 寄存器中 CH_{xMS} ）：

一旦通过配置 CH_{xMS} 选择输入捕获源，必须确保通道配置在输入模式 ($CH_{xMS} \neq 0x0$)，而且 $TIMER_{x_CHxCV}$ 寄存器不能再被写。

第四步：中断使能（ $TIMER_{x_DMAINTEN}$ 寄存器中 CH_{xIE} 和 CH_{xDEN} ）：

使能相应中断，可以获得中断和 DMA 请求。

第五步：捕获使能（ $TIMER_{x_CHCTL2}$ 寄存器中 CH_{xEN} ）。

结果：当期望的输入信号发生时， $TIMER_{x_CHxCV}$ 被设置成当前计数器的值， CH_{xIF} 为置 1。

如果 CH_{xIF} 位已经为 1，则 CH_{xOF} 位置 1。根据 $TIMER_{x_DMAINTEN}$ 寄存器中 CH_{xIE} 和 CH_{xDEN} 的配置，相应的中断和 DMA 请求会被提出。

直接产生: 软件设置CHxG位，会直接产生中断和DMA请求。

输入捕获模式也可用来测量TIMERx_CHx引脚上信号的脉冲波宽度。例如，一个PWM波连接到CI0。配置TIMERx_CHCTL0寄存器中CH0MS为2'b01，选择通道0的捕获信号为CI0并设置上升沿捕获。配置TIMERx_CHCTL0寄存器中CH1MS为2'b10，选择通道1捕获信号为CI0并设置下降沿捕获。计数器配置为复位模式，在通道0的上升沿复位。TIMERX_CH0CV寄存器测量PWM的周期值，TIMERx_CH1CV寄存器测量PWM占空比值。

■ 输出比较模式

在输出比较模式，TIMERx可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的CHxVAL寄存器与计数器的值匹配时，根据CHxCOMCTL的配置，这个通道的输出可以被置高电平，被置低电平或者反转。当计数器的值与CHxVAL寄存器的值匹配时，CHxIF位被置1，如果CHxIE = 1则会产生中断，如果CHxDEN=1则会产生DMA请求。

配置步骤如下：

第一步：时钟配置：

配置定时器时钟源，预分频器等。

第二步：比较模式配置：

设置CHxCOMSEN位来配置输出比较影子寄存器；

设置CHxCOMCTL位来配置输出模式（置高电平/置低电平/反转）；

设置CHxP/CHxNP位来选择有效电平的极性；

设置CHxEN使能输出。

第三步：通过CHxIE/CHxDEN位配置中断/DMA请求使能。

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

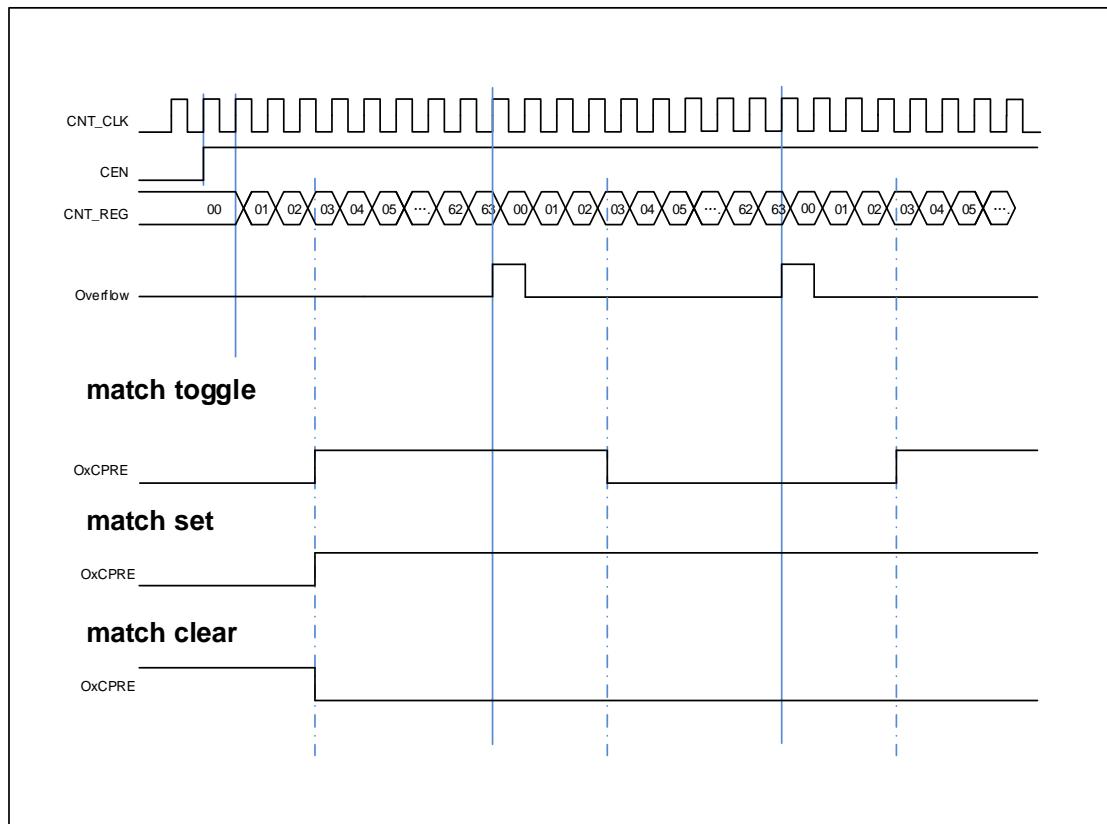
CHxVAL可以在运行时根据你所期望的波形而改变。

第五步：设置CEN位使能定时器。

图 16-65. 三种输出比较模式显示了三种比较输出模式：反转/置高电平/置低电平，CAR=0x63，

CHxVAL=0x3。

图 16-65. 三种输出比较模式



PWM 模式

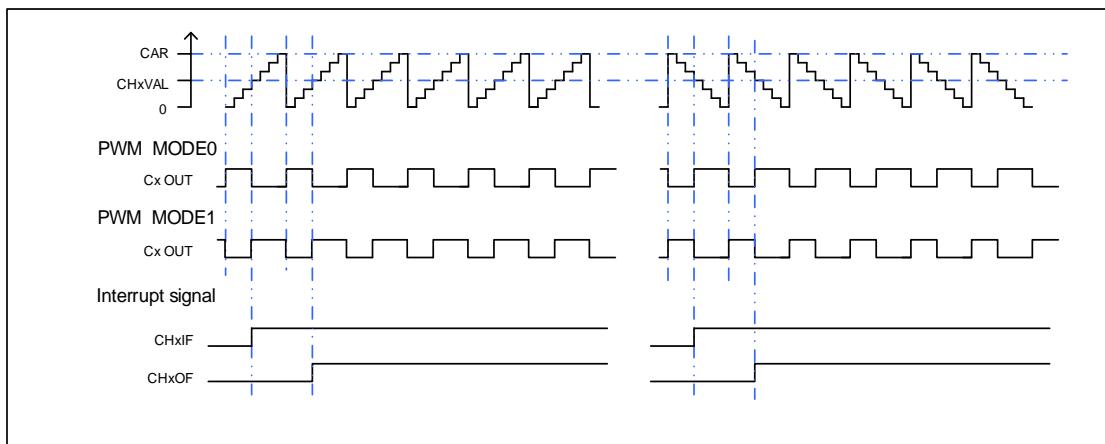
在 PWM 输出模式下 (PWM 模式 0 是配置 CHxCOMCTL 为 3'b110， PWM 模式 1 是配置 CHxCOMCTL 为 3'b111)，通道根据 TIMERx_CAR 寄存器和 TIMERx_CHxCV 寄存器的值，输出 PWM 波形。

PWM 的周期由 TIMERx_CAR 寄存器值决定，占空比由 TIMERx_CHxCV 寄存器值决定。[图 16-66. PWM 时序图](#) 显示了 PWM 的输出波形和中断。

在 PWM0 模式下 (CHxCOMCTL==3'b110)，如果 TIMERx_CHxCV 寄存器的值大于 TIMERx_CAR 寄存器的值，通道输出一直为有效电平。

在 PWM0 模式下 (CHxCOMCTL==3'b110)，如果 TIMERx_CHxCV 寄存器的值等于 0，通道输出一直为无效电平。

图 16-66. PWM 时序图



通道输出参考信号

当TIMERx用于输出匹配比较模式下，设置CHxCOMCTL位可以定义OxCPRE信号(通道x准备信号)类型。OxCPRE信号有若干类型的输出功能，包括，设置CHxCOMCTL=0x00可以保持原始电平；设置CHxCOMCTL=0x01可以将OxCPRE信号设置为高电平；设置CHxCOMCTL=0x02可以将OxCPRE信号设置为低电平；设置CHxCOMCTL=0x03，在计数器值和TIMERx_CHxCV寄存器的值匹配时，可以翻转输出信号。

PWM模式0和PWM模式1是OxCPRE的另一种输出类型，设置CHxCOMCTL位域位0x06或0x07可以配置PWM模式0/PWM模式1。在这些模式中，根据计数器值和TIMERx_CHxCV寄存器值的关系以及计数方向，OxCPRE信号改变其电平。具体细节描述，请参考相应的位。

设置CHxCOMCTL =0x04或0x05可以实现OxCPRE信号的强制输出功能。输出比较信号能够直接由软件强置为有效或无效状态，而不依赖于TIMERx_CHxCV的值和计数器值之间间的比较结果。

互补输出

CHx_O和CHx_ON是一对互补输出通道，这两个信号不能同时有效。TIMERx有两路通道，只有第一路有互补输出通道。互补信号CHx_O和CHx_ON是由一组参数来决定：TIMERx_CHCTL2寄存器中的CHxEN和CHxNEN位，TIMERx_CCHP寄存器中和TIMERx_CTL1寄存器中的POEN, ROS, IOS, ISOx和ISOxN位。输出极性由TIMERx_CHCTL2寄存器中的CHxP和CHxNP位来决定。

表 16-8. 由参数控制的互补输出表

互补参数					输出状态	
POEN	ROS	IOS	CHxEN	CHxNEN	CHx_O	CHx_ON
0	0/1	0	0	0	CHx_O / CHx_ON = LOW	

互补参数					输出状态	
POEN	ROS	IOS	CHxEN	CHxNEN	CHx_O	CHx_ON
1	0/1	0	1	1	CHx_O / CHx_ON 输出禁用.	
					CHx_O = CHxP CHx_ON = CHxNP	
				0	CHx_O/CHx_ON 输出禁用.	
					如果时钟使能: CHx_O = ISOx CHx_ON = ISOxN	
			1	0	CHx_O = CHxP CHx_ON = CHxNP CHx_O/CHx_ON输出禁用.	
					CHx_O = CHxP CHx_ON = CHxNP	
				1	CHx_O/CHx_ON 输出使能. 如果时钟使能: CHx_O = ISOx CHx_ON = ISOxN	
					CHx_O = ISOx CHx_ON = ISOxN	
			0	0	CHx_O/CHx_ON = LOW CHx_O/CHx_ON 输出禁用.	
					CHx_O = LOW	CHx_ON=OxCPRE \oplus CHxNP
				1	CHx_O 输出禁用.	CHx_ON 输出使能
					CHx_O=OxCPRE \oplus CHxP CHx_O输出使能	CHx_ON = LOW CHx_ON输出禁用.
			1	0	CHx_O=OxCPRE \oplus CHxP CHx_O输出使能	
					CHx_O = CHxP CHx_O输出禁用.	CHx_ON = CHxNP CHx_ON输出禁用.
				1	CHx_O = CHxP CHx_O输出使能	CHx_ON=OxCPRE \oplus CHxNP CHx_ON输出使能
					CHx_O=OxCPRE \oplus CHxP CHx_O输出使能	CHx_ON = CHxNP CHx_ON输出使能
			1	0	CHx_O=OxCPRE \oplus CHxP CHx_O输出使能	
					CHx_O = CHxP CHx_O输出使能	CHx_ON=OxCPRE \oplus CHxNP CHx_ON输出使能

死区时间插入

设置 CHxEN 和 CHxNEN 为1'b1同时设置 POEN , 死区插入就会被使能。 DTCFG 位域定义了死区时间, 死区时间对通道0有效。死区时间的细节, 请参考 TIMERx_CCHP 寄存器。

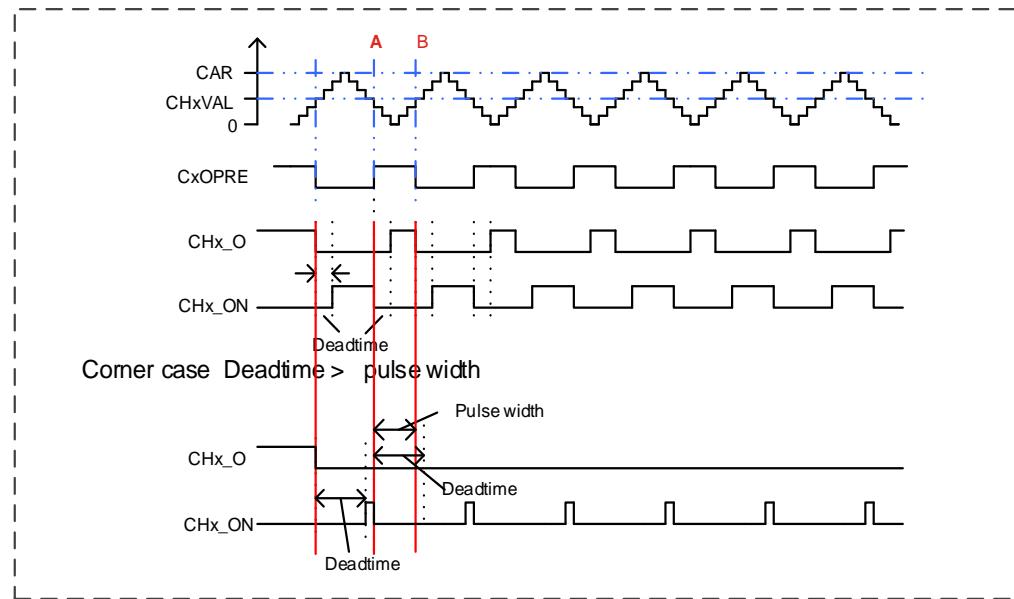
死区时间的插入, 确保了通道互补的两路信号不会同时有效。

在 PWM0 模式, 当通道 x 匹配发生时 (TIMERx 计数器= CHxVAL), OxCOPRE 反转。在 [图 16-67. 带死区时间的互补输出](#) 中的 A 点, CHx_O 信号在死区时间内为低电平, 直到死区时间过后才变为高电平, 而 CHx_ON 信号立刻变为低电平。同样, 在 B 点, 计数器再次匹配 (TIMERx 计数器= CHxVAL), OxCOPRE 信号被清 0, CHx_O 信号被立即清零, CHx_ON 信号在死区时间内仍然是低电平, 在死区时间过后才变为高电平。

有时会有一些死角事件发生, 例如:

- 如果死区延时大于或者等于 CHx_O 信号的占空比, CHx_O 信号一直为无效值 (如 [图 16-67. 带死区时间的互补输出](#))。
- 如果死区延时大于或者等于 CHx_ON 信号的占空比, CHx_ON 信号一直为无效值。

图 16-67. 带死区时间的互补输出



中止功能

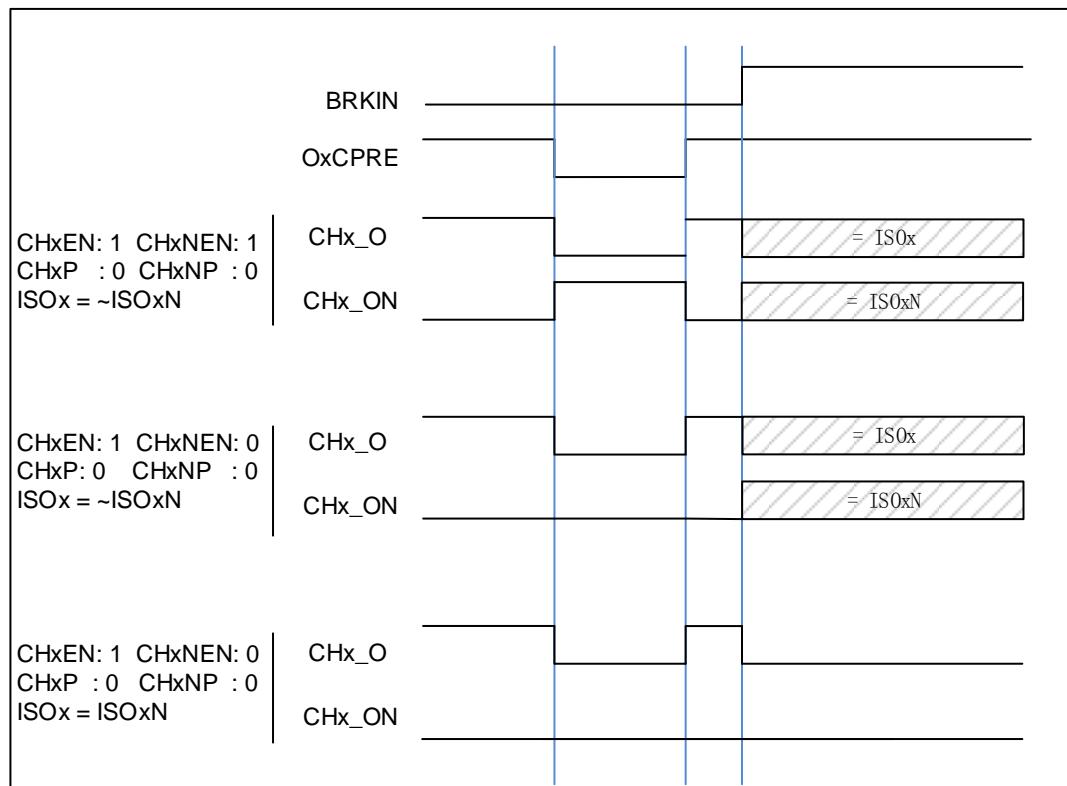
使用中止功能时, 输出 CHx_O 和 CHx_ON 信号电平被以下位控制, TIMERx_CCHP 寄存器的 POEN , IOS 和 ROS 位, TIMERx_CTL1 寄存器的 ISOx 和 ISOxN 位。当中止事件发生时, CHx_O 和 CHx_ON 信号输出不能同时设置为有效电平。中止源可以选择中止输入引脚, 也可以选择 HXTAL 时钟失效事件。时钟失效事件由 RCU 中的时钟监视器 (CKM) 产生。将 TIMERx_CCHP 寄存器的 BRKEN 位置1可以使能中止功能。 TIMERx_CCHP 寄存器的 BRKP 位决定了中止输入极性。

发生中止时, POEN 位被异步清除, 一旦 POEN 位为0, CHx_O 和 CHx_ON 被 TIMERx_CTL1 寄存器中的 ISOx 位和 ISOxN 驱动。如果 $\text{IOS}=0$, 定时器释放输出使能, 否则输出使能仍然为高。

起初互补输出被置于复位状态，然后死区时间产生器重新被激活，以便在一个死区时间后驱动输出，输出电平由ISOx和ISOxN位配置。

发生中止时，TIMERx_INTF寄存器的BRKIF位被置1。如果BRKIE=1，中断产生。

图 16-68. 通道响应中止输入（高电平有效）时，输出信号的行为



从控制器

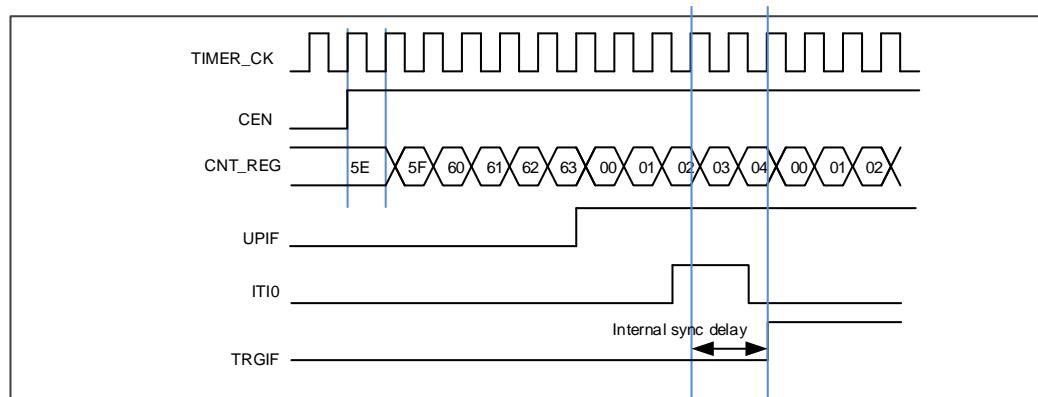
TIMERx能在多种模式下同步外部触发，包括复位模式，暂停模式和事件模式，可以通过设置TIMERx_SMCFG寄存器中的SMC [2:0]配置这些模式。这些模式的输入触发源可以通过设置TIMERx_SMCFG寄存器中的TRGS [2:0]来选择。

表 16-9. 从模式例子列表

	模式选择	触发源选择	极性选择	滤波和预分频
列举	SMC[2:0] 3'b100 (复位模式) 3'b101 (暂停模式) 3'b110 (事件模式)	TRGS[2:0] 000: ITI0 001: ITI1 010: ITI2 011: ITI3 100: CI0F_ED 101: CI0FE0 110: CI1FE1 111: 保留	如果触发源是CI0FE0 或者 CI1FE1，配置 CHxP 和 CHxNP 来选择极性和反相	触发源 ITIx，滤波和预分频不可用 触发源 CIx，配置 CHxCAPFLT 设置滤波，分频不可用

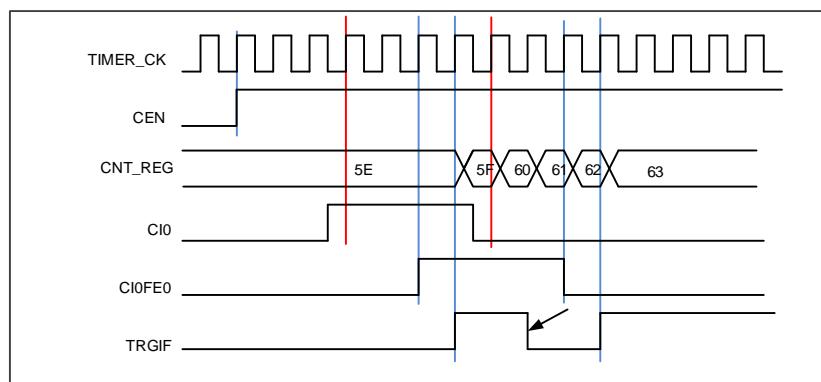
	模式选择	触发源选择	极性选择	滤波和预分频
例1	复位模式 当触发输入上升沿，计数器清零重启	TRGIS[2:0]=3'b000 选择ITIO为触发源	触发源是ITIO, 极性选择不可用	触发源是 ITIO, 滤波和预分频不可用

图 16-69. 复位模式下的控制电路

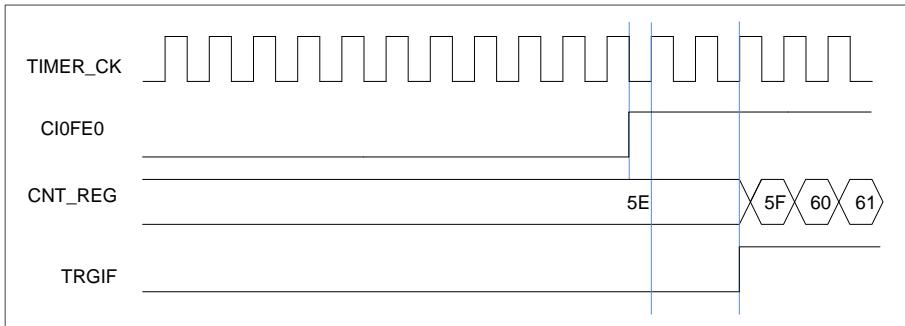


例2	暂停模式 当触发输入为低的时候, 计数器暂停计数	TRGIS[2:0]=3'b101 选择CI0FE0为触发源	TIOS=0. (非异或) [CH0NP==0, CH0P==0]不反相. 在上升沿捕获	在这个例子中滤波被旁路
----	-----------------------------	-----------------------------------	--	-------------

图 16-70. 暂停模式下的控制电路



例3	事件模式 触发输入的上升沿 计数器开始计数	TRGIS[2:0]=3'b101 选择 CI0FE0 为触发源.	TIOS=0. (非异或) [CH0NP==0,CH0P==0] 不反相	在这个例子中滤波被旁路
----	-----------------------------	--------------------------------------	--	-------------

	模式选择	触发源选择	极性选择	滤波和预分频
图 16-71. 事件模式下的控制电路				
	 <p>The timing diagram illustrates the control signals in event mode. The TIMER_CK signal shows a continuous square wave. The CI0FE0 signal is a long pulse starting at the beginning of the diagram. The CNT_REG signal shows a sequence of values: 5E, 5F, 60, 61, and back to 5E. The TRGIF signal is a pulse that occurs when the counter reaches the value of the compare register (60).</p>			

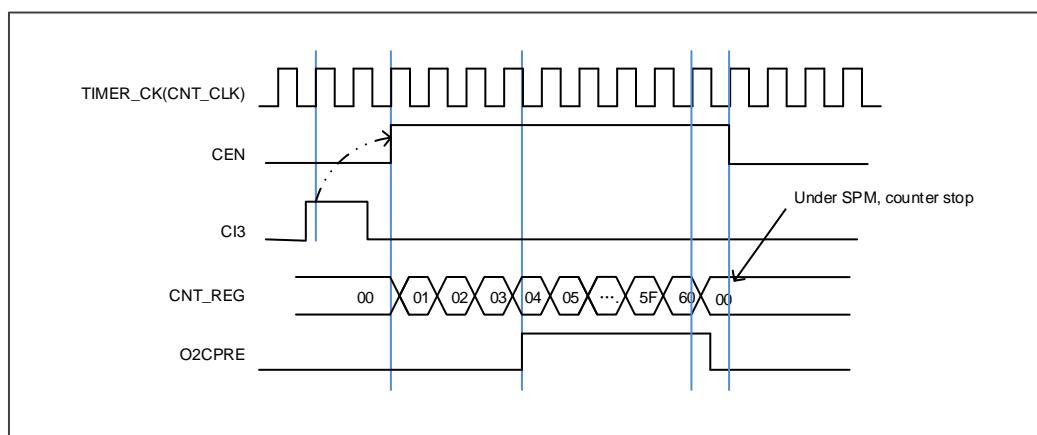
单脉冲模式

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

一旦设置定时器运行在单脉冲模式下，没有必要设置TIMERx_CTL0寄存器的定时器使能位CEN=1来使能计数器。触发信号沿或者软件写CEN=1都可以产生一个脉冲，此后CEN位一直保持为1直到更新事件发生或者CEN位被软件写0。如果CEN位被软件清0，计数器停止工作，计数值被保持。如果CEN值被硬件更新事件自动清0，计数器将被再次初始化。

在单脉冲模式下，有效的外部触发边沿会将CEN位置1，使能计数器。然而，执行计数值和TIMERx_CHxCV寄存器值的比较结果依然存在一些时钟延迟。为了最大限度减少延迟，用户可以将TIMERx_CHCTL0/1寄存器的CHxCOMFEN位置1。单脉冲模式下，触发上升沿产生之后，OxCPRE信号将被立即强制转换为与发生比较匹配时相同的电平，但是不用考虑比较结果。只有输出通道配置为PWM0或PWM1输出运行模式下时CHxCOMFEN位才可用，触发源来源于触发信号。

图 16-72. 单脉冲模式，TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60



定时器互连

参考 [高级定时器 \(TIMERx,x=0\)](#)

表 16-10. TIMERx(x=14)定时器内部互连

Slave TIMER	ITI0(TRGS = 000)	ITI1(TRGS = 001)	ITI2(TRGS = 010)	ITI3(TRGS = 011)
TIMER14	TIMER1	TIMER2	保留	保留

定时器 DMA 模式

定时器DMA模式是指通过DMA模块配置定时器的寄存器。有两个跟定时器DMA模式相关的寄存器：TIMERx_DMACFG和TIMERx_DMATB。当然，必须要使能DMA请求，一些内部中断事件可以产生DMA请求。当中断事件发生，TIMERx会给DMA发送请求。DMA配置成M2P模式，PADDR是TIMERx_DMATB寄存器地址，DMA就会访问TIMERx_DMATB寄存器。实际上，TIMERx_DMATB寄存器只是一个缓冲，定时器会将TIMERx_DMATB映射到一个内部寄存器，这个内部寄存器由TIMERx_DMACFG寄存器中的DMATA来指定。如果TIMERx_DMACFG寄存器的DMATC位域值为0，表示1次传输，定时器的发送1个DMA请求就可以完成。如果TIMERx_DMACFG寄存器的DMATC位域值不为1，例如其值为3，表示4次传输，定时器就需要再多发3次DMA请求。在这3次请求下，DMA对TIMERx_DMATB寄存器的访问会映射到访问定时器的DMATA+0x4, DMATA+0x8, DMATA+0xc寄存器。总之，发生一次DMA内部中断请求，定时器会连续发送(DMATC+1)次请求。

如果再来1次DMA请求事件，TIMERx将会重复上面的过程。

定时器调试模式

当Cortex™-M4内核停止，DBG_CTL1寄存器中的TIMERx_HOLD配置位被置1，定时器计数器停止。

16.4.5. TIMERx 寄存器(x=14)

控制寄存器 0 (TIMERx_CTL0)

地址偏移: 0x00

复位值: 0x0000

该寄存器可以按半字(16位)或字(32位)访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
保留				CKDIV[1:0]		ARSE		保留				SPM	UPS	UPDIS	CEN								
rw rw rw rw rw rw																							
位/位域	名称	描述																					
15:10	保留	必须保持复位值																					
9:8	CKDIV[1:0]	时钟分频																					

通过软件配置CKDIV，规定定时器时钟(TIMER_CK)与死区时间和采样时钟(DTS)之间的分频系数，死区发生器和数字滤波器会用到DTS时间。

00: fDTS=fTIMER_CK

01: fDTS= fTIMER_CK /2

10: fDTS= fTIMER_CK /4

11: 保留

7	ARSE	自动重载影子使能 0: 禁能TIMERx_CAR寄存器的影子寄存器 1: 使能TIMERx_CAR寄存器的影子寄存器
6:4	保留	必须保持复位值
3	SPM	单脉冲模式 0: 更新事件发生后，计数器继续计数 1: 在下一次更新事件发生时，CEN硬件清零并且计数器停止计数
2	UPS	更新请求源 软件配置该为，选择更新事件源。 0: 使能后，下述任一事件产生更新中断或DMA请求： – UPG位被置1 – 计数器溢出/下溢 – 从模式控制器产生的更新 1: 使能后只有计数器溢出/下溢才产生更新中断或DMA请求
1	UPDIS	禁止更新。 该位用来使能或禁能更新事件的产生。 0: 更新事件使能.当以下事件之一发生时，更新事件产生，具有缓存的寄存器被装入它们的预装载值： – UPG位被置1 – 计数器溢出/下溢 – 从模式控制器产生一个更新事件 1: 更新事件禁能. 带有缓存的寄存器保持原有值，如果UPG位被置1或者从模式控制器产生一个硬件复位事件，计数器和预分频器被重新初始化
0	CEN	计数器使能 0: 计数器禁能 1: 计数器使能 在软件将CEN位置1后，外部时钟、暂停模式和编码器模式才能工作。触发模式可以自动地通过硬件设置CEN位。

控制寄存器 1 (TIMERx_CTL1)

地址偏移: 0x04

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				ISO1	ISO0N	ISO0	保留	MMC[2:0]			DMAS	CCUC	保留	CCSE	
				rw	rw	rw			rw		rw	rw		rw	

位/位域	名称	描述
15:11	保留	必须保持复位值
10	ISO1	通道1的空闲状态输出 参考ISO0位
9	ISO0N	通道0的互补通道空闲状态输出 0: 当POEN复位, CH0_ON设置低电平. 1: 当POEN复位, CH0_ON设置高电平 此位只有在TIMERx_CCHP寄存器的PROT [1:0]位为00的时候可以被更改.
8	ISO0	通道0的空闲状态输出 0: 当POEN复位, CH0_O设置低电平 1: 当POEN复位, CH0_O设置高电平 如果CH0_ON生效, 一个死区时间后CH0_O输出改变。此位只有在TIMERx_CCHP寄存器的PROT [1:0]位为00的时候可以被更改.
7	保留	必须保持复位值
6:4	MMC[2:0]	主模式控制 这些位控制TRGO信号的选择, TRGO信号由主定时器发给从定时器用于同步功能 000: 复位。TIMERx_SWEVG寄存器的UPG位被置1或从模式控制器产生复位触发一次TRGO脉冲, 后一种情况下, TRGO上的信号相对实际的复位会有一个延迟。 001: 使能。此模式可用于同时启动多个定时器或控制在一段时间内使能从定时器。主模式控制器选择计数器使能信号作为触发输出TRGO。当CEN控制位被置1或者暂停模式下触发输入为高电平时, 计数器使能信号被置1。在暂停模式下, 计数器使能信号受控于触发输入, 在触发输入和TRGO 上会有一个延迟, 除非选择了主/ 从模式。 010: 更新。主模式控制器选择更新事件作为TRGO。 011: 捕获/比较脉冲.通道0在发生一次捕获或一次比较成功时, 主模式控制器产生一个TRGO脉冲 100: 比较。在这种模式下主模式控制器选择O0CPRE信号被用于作为触发输出TRGO 101: 比较。在这种模式下主模式控制器选择O1CPRE信号被用于作为触发输出TRGO 110: 比较。在这种模式下主模式控制器选择O2CPRE信号被用于作为触发输出TRGO 111: 比较。在这种模式下主模式控制器选择O3CPRE信号被用于作为触发输出TRGO
3	DMAS	DMA请求源选择 0: 当通道捕获/比较事件发生时, 发送通道x的DMA请求 . 1: 当更新事件发生, 发送通道x的DMA请求
2	CCUC	换相控制影子寄存器更新控制 当换相控制影子寄存器 (CHxEN, CHxNEN和CHxCOMCTL位) 使能(CCSE=1), 这些影子寄存器更新控制如下: 0: CMTG位被置1时更新影子寄存器 1: 当CMTG位被置1或检测到TRIGI上升沿时, 影子寄存器更新

当通道没有互补输出时，此位无效。

1	保留	必须保持复位值.
0	CCSE	换相控制影子使能 0: 影子寄存器CHxEN, CHxNEN和CHxCOMCTL位禁能. 1: 影子寄存器CHxEN, CHxNEN和CHxCOMCTL位使能. 如果这些位已经被写入了，换相事件到来时这些位才被更新 当通道没有互补输出时，此位无效

从模式配置寄存器 (TIMERx_SMCFG)

地址偏移: 0x08

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				MSM		TRGS[2:0]			保留.		SMC[2:0]				
				rw		rw			rw		rw				

位/位域	名称	描述
15:8	保留	必须保持复位值.
7	MSM	主-从模式 该位被用来同步被选择的定时器同时开始计数。通过TRIGI和TRGO，定时器被连接在一起，TRGO用做启动事件。 0: 主从模式禁能 1: 主从模式使能
6:4	TRGS[2:0]	触发选择 该位域用来指定选择哪一个信号作为用来同步计数器的触发输入源 000: 内部触发输入0 (ITI0) TIMER1 001: 内部触发输入1 (ITI1) TIMER2 010: 保留 011: 保留 100: CI0的边沿标志位 (CI0F_ED) 101: 滤波后的通道0输入 (CI0FE0) 110: 滤波后的通道1输入(CI1FE1) 111: 保留 从模式被使能后这些位不能改
3	保留	必须保持复位值.
2:0	SMC[2:0]	从模式控制 000: 关闭从模式. 如果CEN=1，则预分频器直接由内部时钟驱动 001: 保留 010: 保留 011: 保留

- 100: 复位模式. 选中的触发输入的上升沿重新初始化计数器，并且更新影子寄存器。
 101: 暂停模式. 当触发输入为高时，计数器的时钟开启。一旦触发输入变为低，则计数器停止。
 110: 事件模式. 计数器在触发输入的上升沿启动。计数器不能被从模式控制器关闭。
 111: 外部时钟模式0. 选中的触发输入的上升沿驱动计数器。

由于CI0F_ED是一个脉冲波形，而暂停模式是检测触发信号的电平，所以，当CI0F_ED用作触发输入时，暂停模式必须禁能。

DMA 和中断使能寄存器 (**TIMERx_DMAINTEN**)

地址偏移: 0x0C

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TRGDEN	CMTDEN	保留	CH1DEN	CH0DEN	UPDEN	BRKIE	TRGIE	CMTIE	保留	CH1IE	CH0IE	UPIE		

rw rw

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

		0: 禁止触发中断 1: 使能触发中断
5	CMTIE	换相更新中断使能 0: 禁止换相更新中断 1: 使能换相更新中断
4:3	保留	必须保持复位值.
2	CH1IE	通道1比较/捕获中断使能 0: 禁止通道1中断 1: 使能通道1中断
1	CHOIE	通道0比较/捕获中断使能 0: 禁止通道0中断 1: 使能通道0中断
0	UPIE	更新中断使能 0: 禁止更新中断 1: 使能更新中断

中断标志寄存器 (TIMERx_INTF)

地址偏移: 0x10

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				保留	CH1OF	CH0OF	保留.	BRKIF	TRGIF	CMTIF	保留.	CH1IF	CHOIF	UPIF	

位/位域	名称	描述
15:11	保留	必须保持复位值.
10	CH1OF	通道1捕获溢出标志 参见CH0OF描述
9	CH0OF	通道0捕获溢出标志 当通道0被配置为输入模式时, 在CHOIF标志位已经被置1后, 捕获事件再次发生时, 该标志位可以由硬件置1。该标志位由软件清0. 0: 无捕获溢出中断发生 1: 发生了捕获溢出中断
8	保留	必须保持复位值.
7	BRKIF	中止中断标志位 一旦中止输入有效, 由硬件对该位置'1'。如果中止输入无效, 则该位可由软件清'0'。 0: 无中止事件产生 1: 中止输入上检测到有效电平

6	TRGIF	触发中断标志 当发生触发事件时，此标志由硬件置1。此位由软件清0。当从模式控制器处于除暂停模式外的其它模式时，在触发输入端检测到有效边沿，产生触发事件。当从模式控制器处于暂停模式时，触发输入的任意边沿都可以产生触发事件。 0: 无触发事件产生 1: 触发中断产生
5	CMTIF	通道换相更新中断标志 当通道换相更新事件发生时此标志位被硬件置1，此位由软件清0。 0: 无通道换相更新中断发生 1: 通道换相更新中断发生
4:3	保留	必须保持复位值。
2	CH1IF	通道1比较/捕获中断标志 参见CH0IF描述
1	CH0IF	通道0比较/捕获中断标志 此标志由硬件置1软件清0。当通道0在输入模式下时，捕获事件发生时此标志位被置1；当通道0在输出模式下时，此标志位在一个比较事件发生时被置1。 0: 无通道0中断发生 1: 通道0中断发生
0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置1，软件清0。 0: 无更新中断发生 1: 发生更新中断

软件事件产生寄存器 (**TIMERx_SWEVG**)

地址偏移: 0x14

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					保留			BRKG	TRGG	CMTG		保留	CH1G	CH0G	UPG

w w w w w w w

位/位域	名称	描述
15:8	保留	必须保持复位值。
7	BRKG	产生中止事件 该位由软件置1，用于产生一个中止事件，由硬件自动清0。当此位被置1时，POEN位被清0且BRKIF位被置1，若开启对应的中断和DMA，则产生相应的中断和DMA传输。 0: 不产生中止事件 1: 产生中止事件

6	TRGG	触发事件产生
		此位由软件置1, 由硬件自动清0. 当此位被置1, TIMERx_INTF寄存器的TRGIF标志位被置1, 若开启对应的中断和DMA, 则产生相应的中断和DMA传输。
	0: 无触发事件产生	
	1: 产生触发事件	
5	CMTG	通道换相更新事件发生
		此位由软件置1, 由硬件自动清0. 当此位被置1, 通道捕获/比较控制寄存器 (CHxEN, CHxNEN 和CHxCOMCTL) 的互补输出被更新 (根据TIMERx_CTL1中CCSE值)。
	0: 不产生通道控制更新事件	
	1: 产生通道控制更新事件	
4:3	保留	必须保持复位值.
2	CH1G	通道1捕获或比较事件发生 参见CH0G描述
1	CH0G	通道0捕获或比较事件发生 该位由软件置1, 用于在通道0产生一个捕获/比较事件, 由硬件自动清0。当此位被置1, CH0IF标志位被置1, 若开启对应的中断和DMA, 则发出相应的中断和DMA请求。此外, 如果通道0配置为输入模式, 计数器的当前值被TIMERx_CH0CV寄存器捕获, 如果CH0IF标志位已经为1, 则CH0OF标志位被置1。 0: 不产生通道0捕获或比较事件 1: 发生通道0捕获或比较事件
0	UPG	更新事件产生 此位由软件置1, 被硬件自动清0。当此位被置1, 如果选择了中央对齐或向上计数模式, 计数器被清0。否则(向下计数模式)计数器将载入自动重载值, 预分频计数器将同时被清除。 0: 无更新事件产生 1: 产生更新事件

通道控制寄存器 0 (TIMERx_CHCTL0)

地址偏移: 0x18

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	CH1COMCTL[2:0]	CH1COM SEN	CH1COM FEN	CH1MS[1:0]	保留	CH0COMCTL[2:0]	CH0COM SEN	CH0COM FEN	CH0MS[1:0]	CH0CAPFLT[3:0]	CH0CAPPSC[1:0]	CH0CAPFLT[3:0]	CH0CAPPSC[1:0]	CH0MS[1:0]	CH0MS[1:0]
CH1CAPFLT[3:0]	CH1CAPPSC[1:0]														

rw

rw

rw

rw

rw

rw

输出比较模式:

位/位域	名称	描述
------	----	----

15	保留	必须保持复位值.
14:12	CH1COMCTL[2:0]	通道1输出比较模式 参见CH0COMCTL描述
11	CH1COMSEN	通道1输出比较影子寄存器使能 参见CH0COMSEN描述
10	CH1COMFEN	通道1输出比较快速使能 参见CH0COMFEN描述
9:8	CH1MS[1:0]	通道1模式选择 这些位定义了通道的方向和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2寄存器的CH1EN位被清0)时这些位才可以写。 00: 通道1配置为输出 01: 通道1配置为输入, IS1映射在CI1FE1上 10: 通道1配置为输入, IS1映射在CI0FE1上 11: 通道1配置为输入, IS1映射在ITS上, 此模式仅工作在内部触发器输入被选中时(由 TIMERx_SMCFGFG寄存器的TRGS位选择)。
7	保留	必须保持复位值.
6:4	CH0COMCTL[2:0]	通道0输出比较模式 此位定义了输出参考信号O0CPRE的动作, 而O0CPRE决定了CH0_O、CH0_ON的值。O0CPRE高电平有效, 而CH0_O、CH0_ON的有效电平取决于CH0P、CH0NP位。 000: 冻结。输出比较寄存器TIMERx_CH0CV与计数器TIMERx_CNT间的比较对O0CPRE不起作用 001: 匹配时设置为高。当计数器的值与捕获/比较值寄存器TIMERx_CH0CV相同时, 强制O0CPRE为高。 010: 匹配时设置为低。当计数器的值与捕获/比较值寄存器TIMERx_CH0CV相同时, 强制O0CPRE为低。 011: 匹配时翻转。当计数器的值与捕获/比较值寄存器TIMERx_CH0CV相同时, 强制O0CPRE翻转。 100: 强制为低。强制O0CPRE为低电平 101: 强制为高。强制O0CPRE为高电平 110: PWM 模式0(EAPWM)。在向上计数时, 一旦计数器值小于TIMERx_CH0CV时, O0CPRE为高电平, 否则为低电平。在向下计数时, 一旦计数器的值大于TIMERx_CH0CV时, O0CPRE 为低电平, 否则为高电平。 111: PWM 模式1(CAPWM)。在向上计数时, 一旦计数器值小于TIMERx_CH0CV时, O0CPRE为低电平, 否则为高电平。在向下计数时, 一旦计数器的值大于TIMERx_CH0CV时, O0CPRE为高电平, 否则为低电平。 在PWM模式0或PWM模式1中, 只有当比较结果改变了或者输出比较模式中从冻结模式切换到PWM模式时, O0CPRE电平才改变。 当TIMERx_CCHP寄存器的PROT [1:0]=11且CH0MS =00 (比较模式) 时此位不能被改变。

3	CH0COMSEN	通道0输出比较影子寄存器使能 当此位被置1, TIMERx_CH0CV寄存器的影子寄存器被使能, 影子寄存器在每次更新事件时都会被更新。 0: 禁止通道0输出/比较影子寄存器 1: 使能通道0输出/比较影子寄存器 仅在单脉冲模式下(TIMERx_CTL0寄存器的SPM =1), 可以在未确认预装载寄存器情况下使用PWM 模式 当TIMERx_CCHP寄存器的PROT [1:0]=11且CH0MS =00时此位不能被改变。
2	CH0COMFEN	通道0输出比较快速使能 当该位为1时, 如果通道配置为PWM0模式或者PWM1模式, 会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配, CH0_O被设置为比较电平而与比较结果无关。 0: 禁止通道0输出比较快速. 当触发器的输入有一个有效沿时, 激活CH0_O输出的最小延时为5个时钟周期 1: 使能通道0输出比较快速。当触发器的输入有一个有效沿时, 激活CH0_O输出的最小延时为3个时钟周期
1:0	CH0MS[1:0]	通道0 I/O模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2寄存器的CH0EN位被清0)时这些位才可写。 00: 通道0配置为输出 01: 通道0配置为输入, ISO映射在CI0FE0上 10: 通道0配置为输入, ISO映射在CI1FE0上 11: 通道0配置为输入, ISO映射在ITS上. 此模式仅工作在内部触发输入被选中时(通过设置TIMERx_SMCFGFG寄存器的TRGS位)

输入捕获模式:

位/位域	名称	描述
15:12	CH1CAPFLT[3:0]	通道1输入捕获滤波控制 参见CH0CAPFLT描述
11:10	CH1CAPPSC[1:0]	通道1输入捕获预分频器 参见CH0CAPPSC描述
9:8	CH1MS[1:0]	通道1模式选择 与输出模式相同
7:4	CH0CAPFLT[3:0]	通道0输入捕获滤波控制 数字滤波器由一个事件计数器组成, 它记录N个输入事件后会产生一个输出的跳变。这些位定义了CI0输入信号的采样频率和数字滤波器的长度。 0000: 无滤波器, fSAMP= fDTS, N=1 0001: fSAMP= fPCLK, N=2 0010: fSAMP= fPCLK, N=4 0011: fSAMP= fPCLK, N=8 0100: fSAMP=fDTS/2, N=6

0101: fSAMP=fDTS/2, N=8
 0110: fSAMP=fDTS/4, N=6
 0111: fSAMP=fDTS/4, N=8
 1000: fSAMP=fDTS/8, N=6
 1001: fSAMP=fDTS/8, N=8
 1010: fSAMP=fDTS/16, N=5
 1011: fSAMP=fDTS/16, N=6
 1100: fSAMP=fDTS/16, N=8
 1101: fSAMP=fDTS/32, N=5
 1110: fSAMP=fDTS/32, N=6
 1111: fSAMP=fDTS/32, N=8

3:2 CH0CAPPSC[1:0] 通道0输入捕获预分频器

这2位定义了通道0输入的预分频系数。当TIMERx_CHCTL2寄存器中的CH0EN =0时，则预分频器复位。

- 00: 无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获
- 01: 每2个事件触发一次捕获
- 10: 每4个事件触发一次捕获
- 11: 每8个事件触发一次捕获

1:0 CH0MS[1:0] 通道0模式选择
与输出比较模式相同

通道控制寄存器 2 (TIMERx_CHCTL2)

地址偏移: 0x20

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留.								CH1NP	保留	CH1P	CH1EN	CH0NP	CH0NEN	CH0P	CH0EN

位/位域	名称	描述
15:8	保留	必须保持复位值
7	CH1NP	通道1互补输出极性 参考CH0NP描述
6	保留	必须保持复位值
5	CH1P	通道1极性 参考CH0P 描述
4	CH1EN	通道1使能 参考CH0EN 描述
3	CH0NP	通道0互补输出极性

当通道0配置为输出模式，此位定义了互补输出信号的极性。

0: 通道0高电平有效

1: 通道0低电平有效

当通道0配置为输入模式时，此位和CH0P联合使用，作为输入信号Cl0的极性选择控制信号。

当TIMERx_CCHP寄存器的PROT [1:0]=11或10 时此位不能被更改。

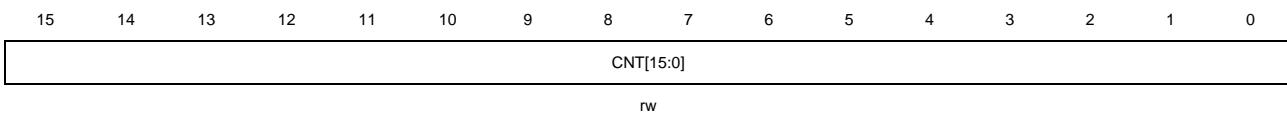
2	CH0NEN	通道0互补输出使能
		当通道0配置为输出模式时，将此位置1使能通道0的互补输出。
0		0: 禁止通道0互补输出
1		1: 使能通道0互补输出
1	CH0P	通道0极性
		当通道0配置为输出模式时，此位定义了输出信号极性。
0		0: 通道0高电平有效
1		1: 通道0低电平有效
		当通道0配置为输入模式时，此位定义了Cl0信号极性
		[CH0NP, CH0P] 将选择Cl0FE0或者Cl1FE0的有效边沿或者捕获极性
		[CH0NP==0, CH0P==0]: 把ClxFE0的上升沿作为捕获或者从模式下触发的有效信号，并且ClxFE0不会被翻转。
		[CH0NP==0, CH0P==1]: 把ClxFE0的下降沿作为捕获或者从模式下触发的有效信号，并且ClxFE0会被翻转。
		[CH0NP==1, CH0P==0]: 保留。
		[CH0NP==1, CH0P==1]: 把ClxFE0的上升沿和下降沿都作为捕获或者从模式下触发的有效信号，并且ClxFE0不会被翻转。
		当TIMERx_CCHP寄存器的PROT [1:0]=11或10 时此位不能被更改。
0	CH0EN	通道0捕获/比较使能
		当通道0配置为输出模式时，将此位置1使能CH0_O信号有效。当通道0配置为输入模式时，将此位置1使能通道0上的捕获事件。
0		0: 禁止通道0
1		1: 使能通道0

计数器寄存器 (TIMERx_CNT)

地址偏移: 0x24

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问



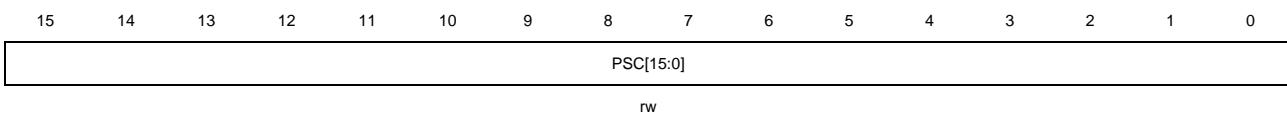
位/位域	名称	描述
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器 (TIMERx_PSC)

地址偏移: 0x28

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问



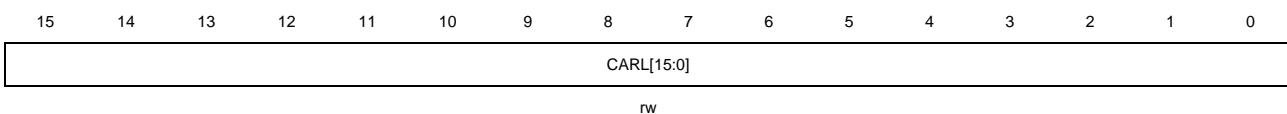
位/位域	名称	描述
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于PSC时钟除以 (PSC+1)，每次当更新事件产生时，PSC 的值被装入当前预分频寄存器。

计数器自动重载寄存器 (TIMERx_CAR)

地址偏移: 0x2C

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问



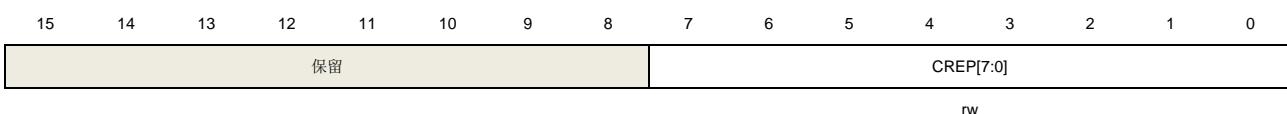
位/位域	名称	描述
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

重复计数寄存器 (TIMERx_CREP)

地址偏移: 0x30

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问



位/位域	名称	描述
15:8	保留	必须保持复位值。
7:0	CREP[7:0]	重复计数器的值 这些位定义了更新事件的产生速率。重复计数器计数值减为0时产生更新事件。影子寄

存器的更新速率也会受这些位影响(前提是影子寄存器被使能)。

通道 0 捕获/比较寄存器 (**TIMERx_CH0CV**)

地址偏移: 0x34

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															
rw															

位/位域	名称	描述
15:0	CH0VAL[15:0]	<p>通道0的捕获或比较值</p> <p>当通道0配置为输入模式时, 这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。</p> <p>当通道0配置为输出模式时, 这些位包含了即将和计数器比较的值。使能相应影子寄存器后, 影子寄存器值随每次更新事件更新。</p>

通道 1 捕获/比较寄存器 (**TIMERx_CH1CV**)

地址偏移: 0x38

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															
rw															

位/位域	名称	描述
15:0	CH1VAL[15:0]	<p>通道1的捕获或比较值</p> <p>当通道1配置为输入模式时, 这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。</p> <p>当通道1配置为输出模式时, 这些位包含了即将和计数器比较的值。使能相应影子寄存器后, 影子寄存器值随每次更新事件更新。</p>

互补通道保护寄存器 (**TIMERx_CCHP**)

地址偏移: 0x44

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POEN	OAEN	BRKP	BRKEN	ROS	IOS	PROT[1:0]									DTCFG[7:0]

位/位域	名称	描述
15	POEN	<p>所有的通道输出使能</p> <p>根据OAEN位，该位可以软件设置或者硬件自动设置。一旦中止输入有效，该位被硬件异步清0。如果一个通道配置为输出模式，如果设置了相应的使能位(TIMERx_CHCTL2寄存器的CHxEN, CHxNEN位)，则开启CHx_O 和CHx_ON 输出。</p> <p>0: 禁止通道输出或强制为空闲状态</p> <p>1: 使能通道输出</p>
14	OAEN	<p>自动输出使能</p> <p>此位定义了POEN位是否可以被硬件自动置1。</p> <p>0: POEN位不能被硬件置1</p> <p>1: 如果中止输入无效，下一次更新事件发生时，POEN位能被硬件自动置1</p> <p>此位只有在TIMERx_CCHP寄存器的PROT [1:0] =00时才可修改。</p>
13	BRKP	<p>中止极性</p> <p>此位定义了中止输入信号BKIN的极性。</p> <p>0: 中止输入低电平有效</p> <p>1: 中止输入高电平有效</p>
12	BRKEN	<p>中止使能</p> <p>此位置1使能中止事件和CCS时钟失败事件输入。</p> <p>0: 禁能中止输入</p> <p>1: 使能中止输入</p> <p>此位只有在TIMERx_CCHP寄存器的PROT [1:0] =00时才可修改。.</p>
11	ROS	<p>运行模式下“关闭状态”配置</p> <p>当POEN位被置1，此位定义了通道(带有互补输出且配置为输出模式)的输出状态。</p> <p>0: 当POEN位被置1，通道输出信号 (CHx_O/ CHx_ON)被禁止</p> <p>1: 当POEN位被置1，通道输出信号 (CHx_O/ CHx_ON)被使能，和TIMER0_CHCTL2寄存器CHxEN/CHxNEN位有关</p> <p>此位在TIMERx_CCHP寄存器的PROT [1:0]=10或11时不能被更改。</p>
10	IOS	<p>空闲模式下“关闭状态”配置</p> <p>当POEN位被清0，此位定义了已经配置为输出模式的通道的输出状态。</p> <p>0: 当POEN位被清0，通道输出信号(CHx_O/ CHx_ON)被禁止</p> <p>1: 当POEN位被清0，通道输出信号(CHx_O/ CHx_ON)被使能，和TIMERx_CHCTL2寄存器CHxEN/CHxNEN位有关</p> <p>此位在TIMERx_CCHP寄存器的PROT [1:0]=10或11时不能被更改。.</p>
9:8	PROT[1:0]	<p>互补寄存器保护控制</p> <p>这两位定义了寄存器的写保护特性。</p> <p>00: 禁能保护模式。无写保护.</p> <p>01: PROT模式0。TIMERx_CTL1寄存器中ISOx/ISOxN 位，TIMERx_CCHP 寄存器中BRKEN/BRKP/OAEN/DTCFG 位写保护</p> <p>10: PROT模式1。除了PROT模式0下的寄存器写保护外，还有TIMERx_CHCTL2寄存器</p>

中CHxP/CHxNP位（如果相应通道配置为输出模式），**TIMERx_CCHP** 寄存器中ROS/IOS 位。

11: PROT模式2.除了PROT模式1下的寄存器写保护外，还有**TIMERx_CHCTRLR0/1** 中CHxCOMCTL/ CHxCOMSEN位（如果相关通道配置为输出模式）写保护。

系统复位后这两位只能被写一次，一旦**TIMERx_CCHP**寄存器被写入，这两位被写保护

7:0 DTCFG[7:0] 死区时间控制

这些位定义了插入互补输出之间的死区持续时间。DTCFG值和死区时间的关系如下：

DTCFG [7:5] =3'b0xx: DTvalue =DTCFG [7:0]x tDT, tDT=tDTS.

DTCFG [7:5] =3'b 10x: DTvalue = (64+DTCFG [5:0])xtDT, tDT=tDTS*2.

DTCFG [7:5] =3'b 110: DTvalue = (32+DTCFG [4:0])xtDT, tDT=tDTS*8.

DTCFG [7:5] =3'b 111: DTvalue = (32+DTCFG [4:0])xtDT, tDT=tDTS*16.

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

DMA 配置寄存器 (**TIMERx_DMACFG**)

地址偏移: 0x48

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

位/位域	名称	描述
15:13	保留	必须保持复位值.
12:8	DMATC [4:0]	DMA传输计数 该位域定义了DMA访问（读写） TIMERx_DMAVB 寄存器的数量
7:5	保留	必须保持复位值
4:0	DMATA [4:0]	DMA传输起始地址 该位域定义了DMA访问 TIMERx_DMAVB 寄存器的第一个地址。当通过 TIMERx_DMA 第一次访问时，访问的就是该位域指定的地址。第二次访问 TIMERx_DMAVB 时，将访问起始地址+0x4。 5'b0_0000: TIMERx_CTL0 5'b0_0001: TIMERx_CTL1 ... 总之： 起始地址 = TIMERx_CTL0 + DMATA*4

DMA 发送缓冲区寄存器 (**TIMERx_DMATB**)

地址偏移: 0x4C

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

rw

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

配置寄存器 (TIMERx_CFG)

地址偏移: 0xFC

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

rw rw

位/位域	名称	描述
15:2	保留	必须保持复位值
1	CHVSEL	写捕获比较寄存器选择位 此位由软件写1或清0。 1: 当写入捕获比较寄存器的值与寄存器当前值相等时，写入操作无效 0: 无影响
0	OUTSEL	输出值选择位 此位由软件写1或清0。 1: 如果POEN位与IOS位均为0，则输出无效 0: 无影响

16.5. 通用定时器 L4 (TIMERx,x=15,16)

16.5.1. 简介

通用定时器L4 (TIMER15/16) 是单通道定时器，支持输入捕获和输出比较。可以产生**PWM**信号控制电机和电源管理。通用定时器L4含有一个16位无符号计数器。

通用定时器L4是可编程的，可以被用来计数，其外部事件可以驱动其他定时器

通用定时器L4包含了一个死区时间插入模块，非常适合电机控制。

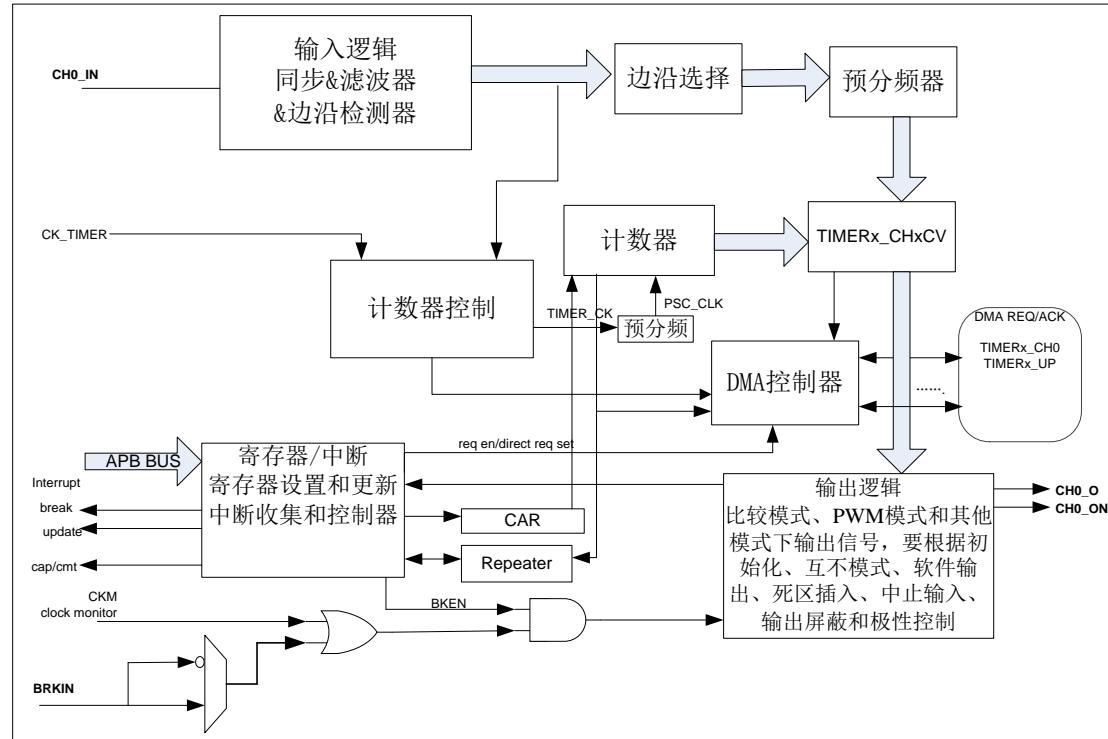
16.5.2. 主要特性

- 总通道数: 1;
- 计数器宽度: 16位;
- 时钟源可选: 内部时钟;
- 计数模式: 向上计数;
- 可编程的预分频器: 16位, 运行时可以被改变;
- 每个通道可配置: 输入捕获模式, 输出比较模式, 可编程的**PWM**模式, 单脉冲模式;
- 可编程的死区时间;
- 自动重装载功能;
- 可编程的计数器重复功能;
- 中止输入功能;
- 中断输出和**DMA**请求: 更新事件, 比较/捕获事件和中止事件;

16.5.3. 结构框图

[图 16-73. 通用定时器 L4 结构框图](#)提供了通用定时器 L4 的内部配置细节

图 16-73. 通用定时器 L4 结构框图



16.5.4. 功能描述

时钟源选择

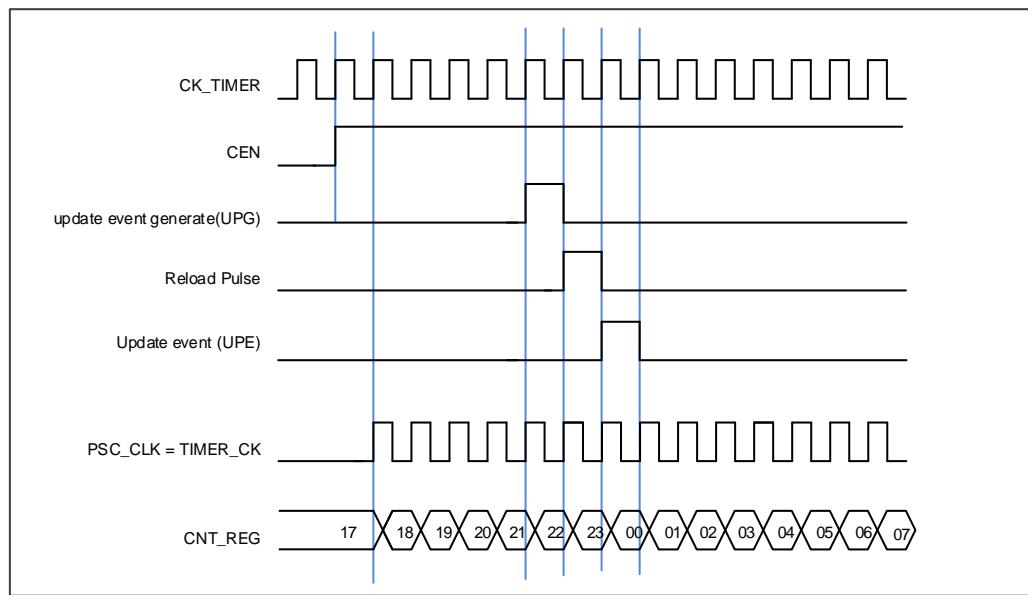
通用定时器L4由内部时钟源TIMER_CK.

- 定时器选择内部时钟源（连接到RCU模块的CK_TIMER）

通用定时器L4只有一个时钟源：内部时钟源。用来驱动计数器预分频器的是内部时钟源CK_TIMER。当CEN置位，CK_TIMER经过预分频器（预分频值由TIMERx_PSC寄存器确定）产生PSC_CLK。

驱动预分频器计数的TIMER_CK等于来自于RCU模块的CK_TIMER

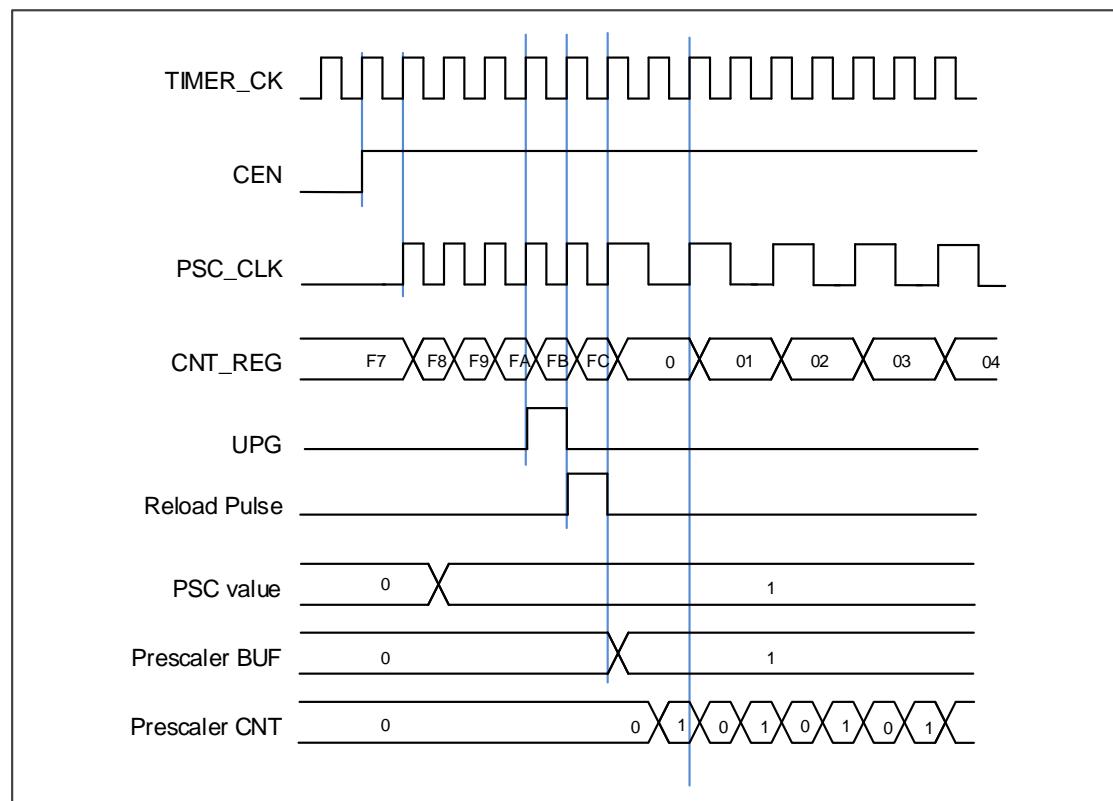
图 16-74. 内部时钟分频为 1 时正常模式下的控制电路



预分频器

预分频器可以将定时器的时钟 (TIMER_CK) 频率按1到65536之间的任意值分频，分频后的时钟PSC_CLK驱动计数器计数。分频系数受预分频寄存器TIMERx_PSC控制，这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 16-75. 当预分频器的参数从 1 变到 2 时，计数器的时序图



向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从0开始向上连续计数到自动加载值（定义在TIMERx_CAR寄存器中），一旦计数器计数到自动加载值，会重新从0开始向上计数。如果设置了重复计数器，在(TIMERx_CREP+1)次上溢后产生更新事件，否则在每次上溢时都会产生更新事件。在向上计数模式中，TIMERx_CTL0寄存器中的计数方向控制位DIR应该被设置成0。

当通过TIMERx_SWEVG寄存器的UPG位置1来设置更新事件时，计数值会被清0，并产生更新事件。

如果TIMERx_CTL0寄存器的UPDIS置1，则禁止更新事件。

当发生更新事件时，所有的寄存器(重复计数器，自动重载寄存器，预分频寄存器)都将被更新。

图16-76. 向上计数时序图, PSC=0/1给出了一些例子，当TIMERx_CAR=0x63时，计数器在不同预分频因子下的行为。

图 16-76. 向上计数时序图, PSC=0/1

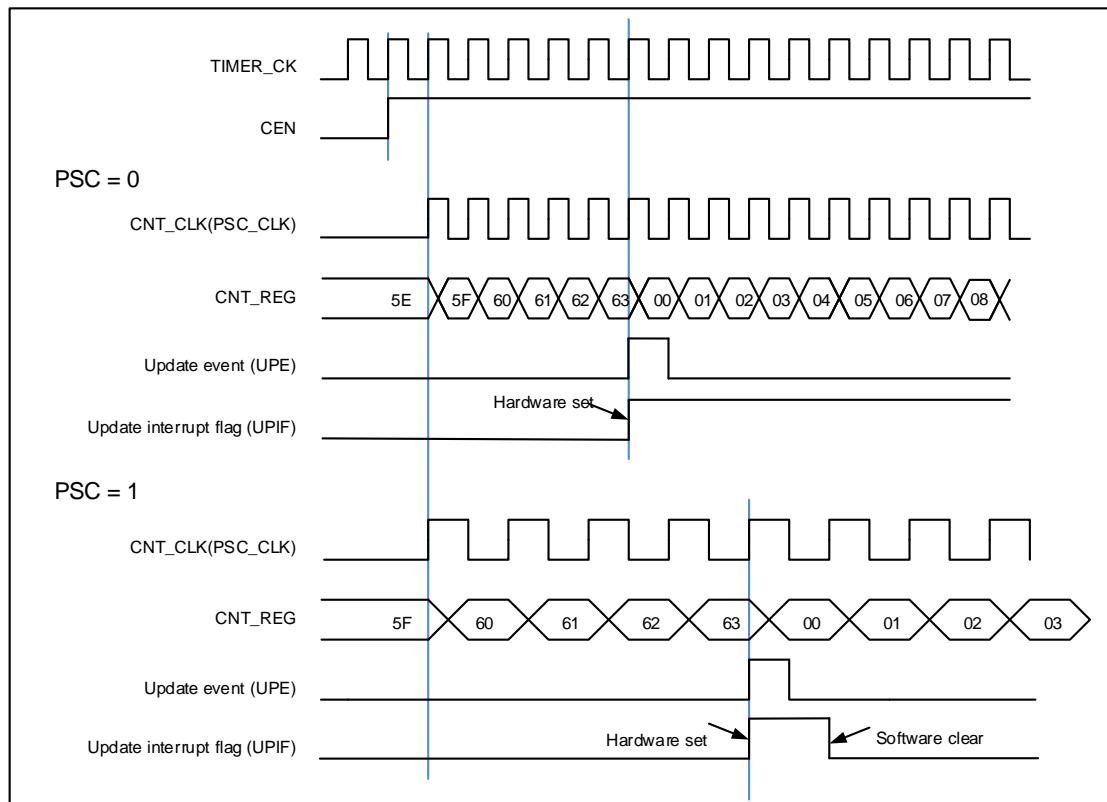
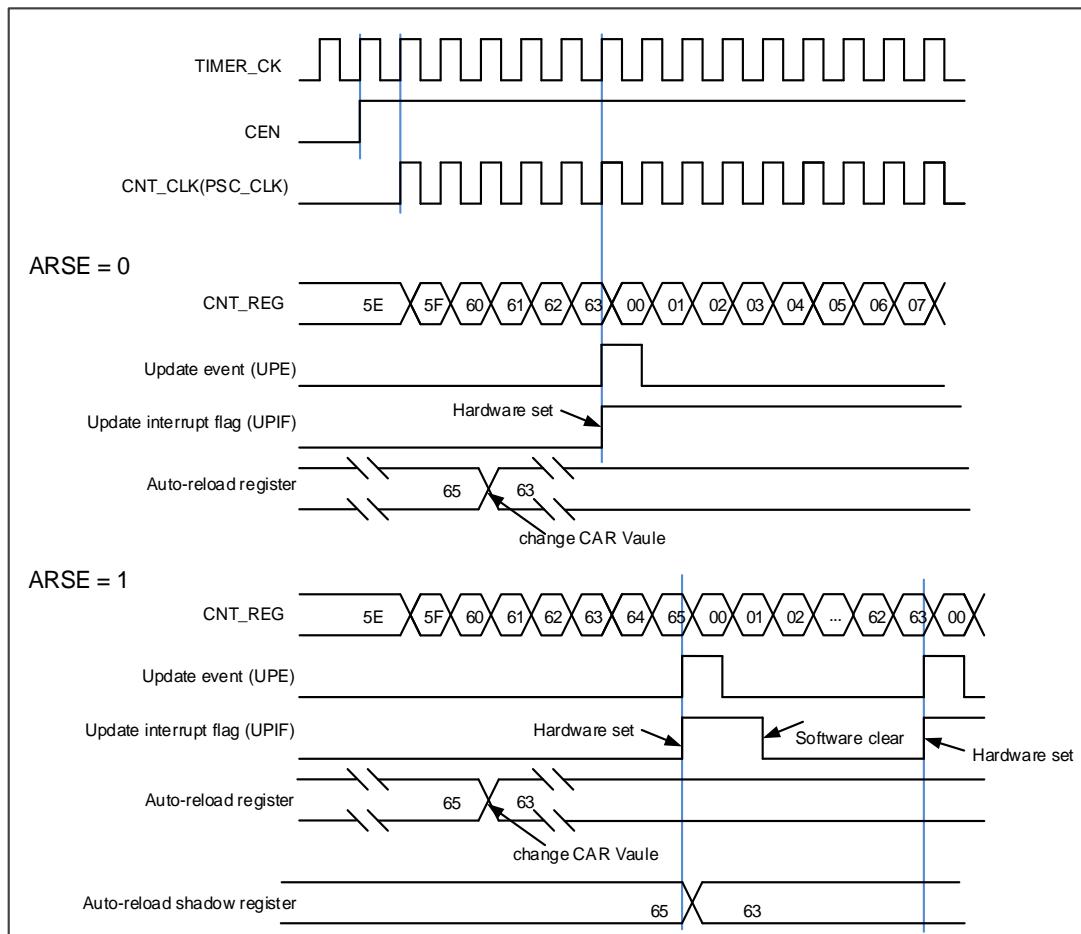


图 16-77. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值

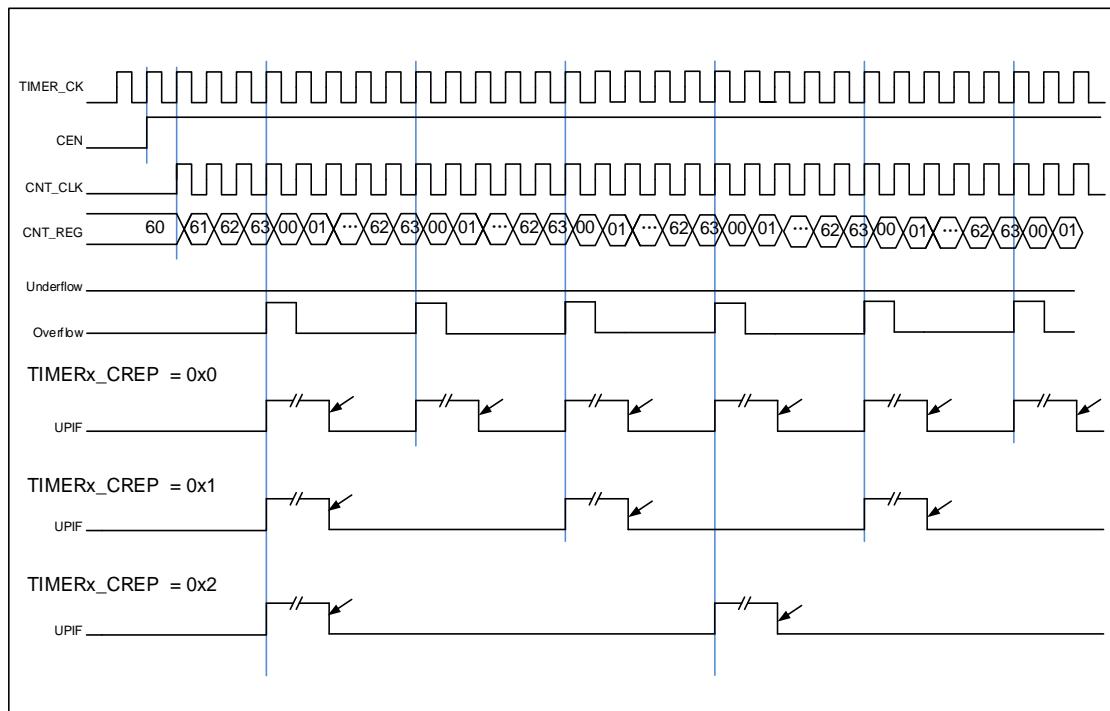


重复计数器

重复计数器是用来在 $N+1$ 个计数周期之后产生更新事件，更新定时器的寄存器， N 为 TIMERx_CREP 寄存器的 CREP。向上计数模式下，重复计数器在每次计数器上溢时递减。

将 TIMERx_SWEVG 寄存器的 UPG 位置 1 可以重载 TIMERx_CREP 寄存器中 CREP 的值并产生一个更新事件。

图 16-78. 在向上计数模式下计数器重复时序图



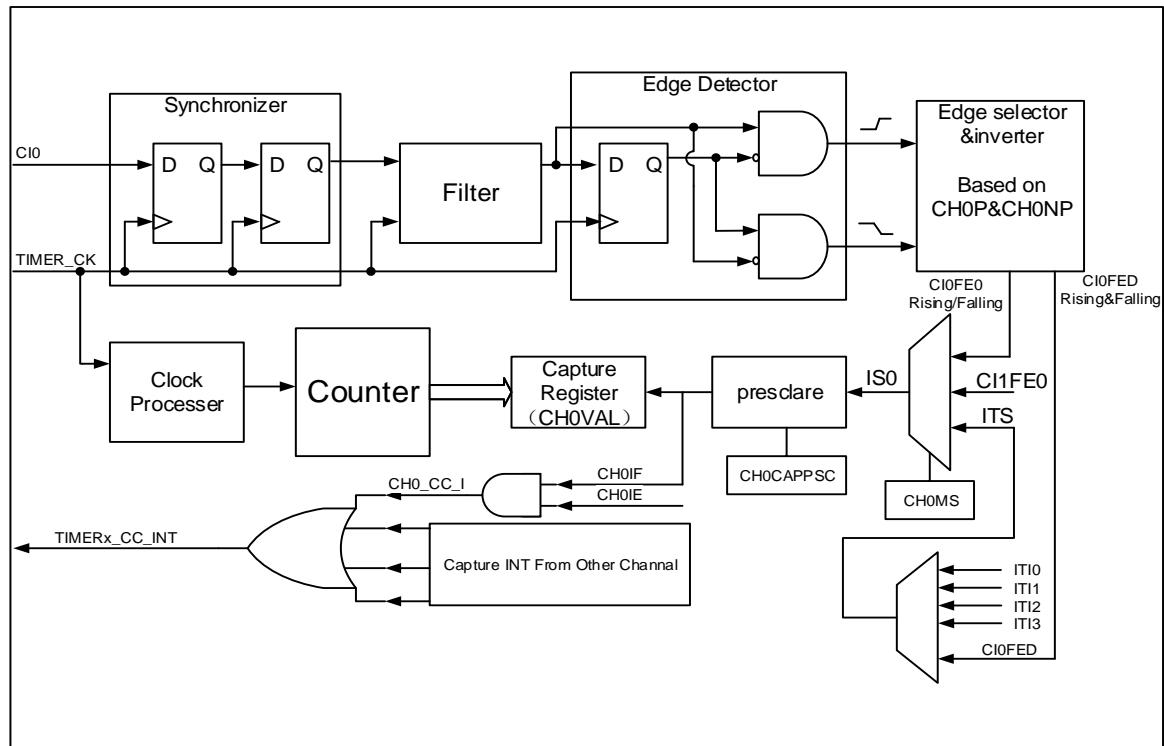
捕获/比较通道

通用定时器L4拥有一个独立的通道用于捕获输入或比较输出是否匹配。每个通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级。

■ 输入捕获模式

捕获模式允许通道测量一个波形时序，频率，周期，占空比等。输入级包括一个数字滤波器，一个通道极性选择，边沿检测和一个通道预分频器。如果在输入引脚上出现被选择的边沿，**TIMERx_CHxCV**寄存器会捕获计数器当前的值，同时**CHxIF**位被置1，如果**CHxIE = 1**则产生通道中断。

图 16-79. 输入捕获逻辑



通道输入信号 CI_x 有两种选择，一种是 $TIMERx_CHx$ 信号，另一种是 $TIMERx_CH0, TIMERx_CH1$ 和 $TIMERx_CH2$ 异或之后的信号。通道输入信号 CI_x 先被 $TIMER_CK$ 信号同步，然后经过数字滤波器采样，产生一个被滤波后的信号。通过边沿检测器，可以选择检测上升沿或者下降沿。通过配置 $CHxP$ 选择使用上升沿或者下降沿。配置 $CHxMS$ ，可以选择其他通道的输入信号，内部触发信号。配置 IC 预分频器，使得若干个输入事件后才产生一个有效的捕获事件。捕获事件发生， $CHxVAL$ 存储计数器的值。

配置步骤如下：

第一步：滤波器配置（ $TIMERx_CHCTL0$ 寄存器中 $CHxCAPFLT$ ）：

根据输入信号和请求信号的质量，配置相应的 $CHxCAPFLT$ 。

第二步：边沿选择（ $TIMERx_CHCTL2$ 寄存器中 $CHxP/CHxNP$ ）：

配置 $CHxP/CHxNP$ 选择上升沿或者下降沿。

第三步：捕获源选择（ $TIMERx_CHCTL0$ 寄存器中 $CHxMS$ ）：

一旦通过配置 $CHxMS$ 选择输入捕获源，必须确保通道配置在输入模式 ($CHxMS \neq 0x0$)，而且 $TIMERx_CHxCV$ 寄存器不能再被写。

第四步：中断使能（ $TIMERx_DMAINTEN$ 寄存器中 $CHxIE$ 和 $CHxDEN$ ）：

使能相应中断，可以获得中断和 DMA 请求。

第五步：捕获使能（ $TIMERx_CHCTL2$ 寄存器中 $CHxEN$ ）。

结果：当期望的输入信号发生时， $TIMERx_CHxCV$ 被设置成当前计数器的值， $CHxIF$ 为置 1。

如果 $CHxIF$ 位已经为 1，则 $CHxOF$ 位置 1。根据 $TIMERx_DMAINTEN$ 寄存器中 $CHxIE$ 和 $CHxDEN$ 的配置，相应的中断和 DMA 请求会被提出。

直接产生: 软件设置CHxG位，会直接产生中断和DMA请求。

■ 输出比较模式

在输出比较模式，TIMERx可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的CHxVAL寄存器与计数器的值匹配时，根据CHxCOMCTL的配置，这个通道的输出可以被置高电平，被置低电平或者反转。当计数器的值与CHxVAL寄存器的值匹配时，CHxFIF位被置1，如果CHxIE = 1则会产生中断，如果CHxDEN=1则会产生DMA请求。

配置步骤如下：

第一步：时钟配置：

配置定时器时钟源，预分频器等。

第二步：比较模式配置：

设置CHxCOMSEN位来配置输出比较影子寄存器；

设置CHxCOMCTL位来配置输出模式（置高电平/置低电平/反转）；

设置CHxP/CHxNP位来选择有效电平的极性；

设置CHxEN使能输出。

第三步：通过CHxIE/CHxDEN位配置中断/DMA请求使能。

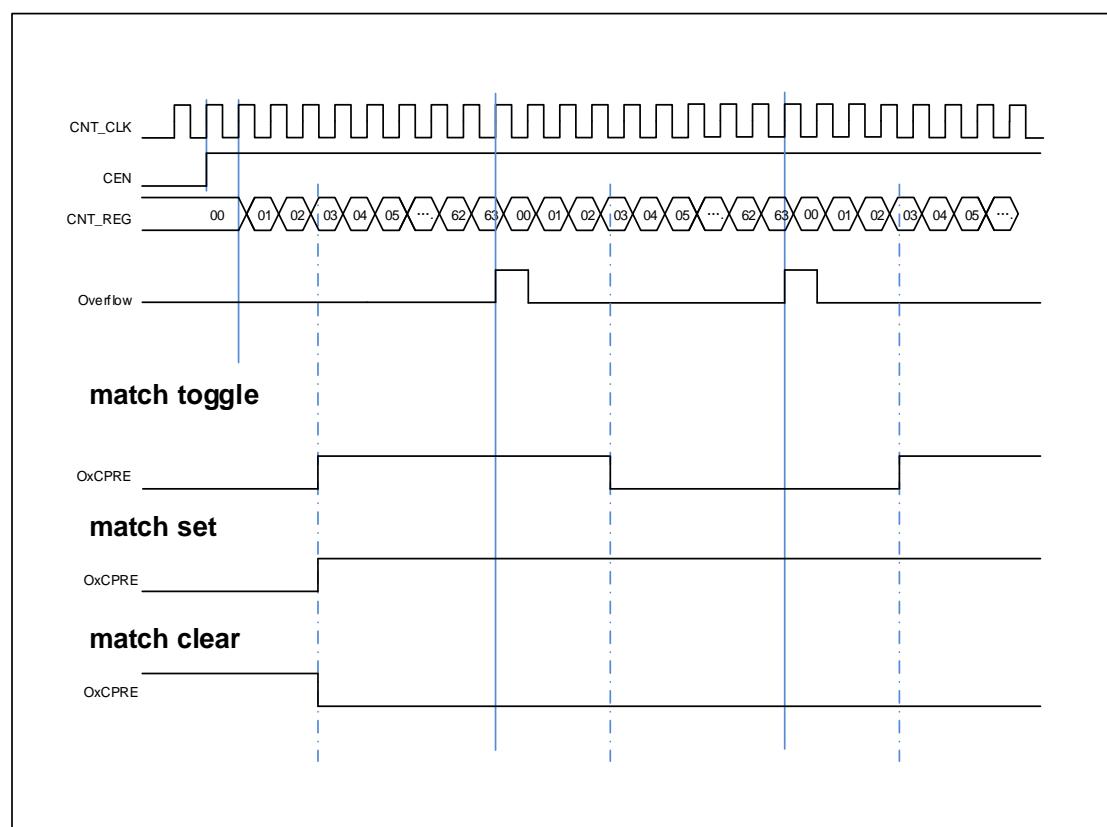
第四步：通过TIMERx_CAR寄存器和TIMERx_CHxCV寄存器配置输出比较时基：

CHxVAL可以在运行时根据你所期望的波形而改变。

第五步：设置CEN位使能定时器。

图 16-80. 三种输出比较模式显示了三种比较输出模式：反转/置高电平/置低电平，CAR=0x63，CHxVAL=0x3。

图 16-80. 三种输出比较模式



PWM 模式

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

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

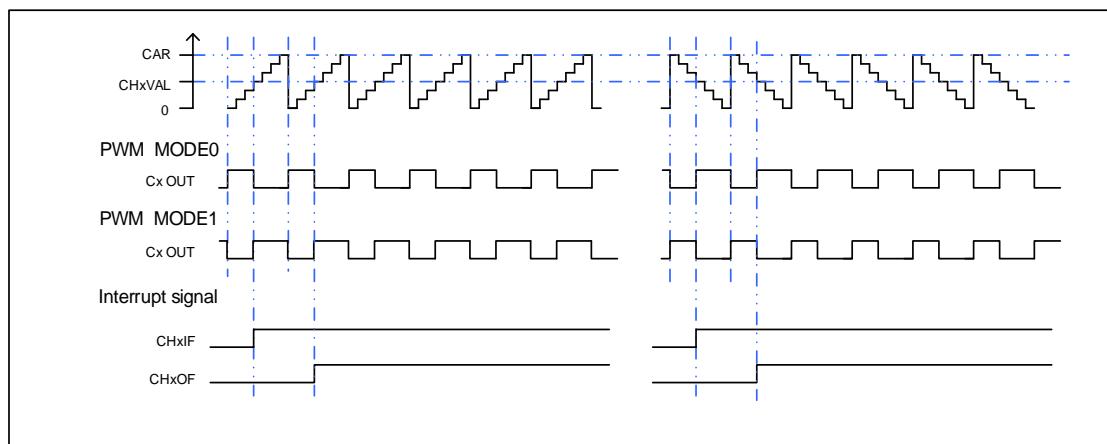
EAPWM 的周期由 TIMERx_CAR 寄存器值决定，占空比由 TIMERx_CHxCV 寄存器值决定。

[图 16-81. PWM 时序图](#)显示了 PWM 的输出波形和中断。

在 PWM0 模式下 (CHxCOMCTL==3'b110)，如果 TIMERx_CHxCV 寄存器的值大于 TIMERx_CAR 寄存器的值，通道输出一直为有效电平。

在 PWM0 模式下 (CHxCOMCTL==3'b110)，如果 TIMERx_CHxCV 寄存器的值等于 0，通道输出一直为无效电平。

图 16-81. PWM 时序图



通道输出参考信号

当 TIMERx 用于输出匹配比较模式下，设置 CHxCOMCTL 位可以定义 OxCPR 信号 (通道 x 准备信号) 类型。OxCPR 信号有若干类型的输出功能，包括，设置 CHxCOMCTL=0x00 可以保持原始电平；设置 CHxCOMCTL=0x01 可以将 OxCPR 信号设置为高电平；设置 CHxCOMCTL=0x02 可以将 OxCPR 信号设置为低电平；设置 CHxCOMCTL=0x03，在计数器值和 TIMERx_CHxCV 寄存器的值匹配时，可以翻转输出信号。

PWM 模式 0 和 PWM 模式 1 是 OxCPR 的另一种输出类型，设置 CHxCOMCTL 位域位 0x06 或 0x07 可以配置 PWM 模式 0/PWM 模式 1。在这些模式中，根据计数器值和 TIMERx_CHxCV 寄存器值的关系以及计数方向，OxCPR 信号改变其电平。具体细节描述，请参考相应的位。

设置 CHxCOMCTL = 0x04 或 0x05 可以实现 OxCPR 信号的强制输出功能。输出比较信号能够

直接由软件强置为有效或无效状态，而不依赖于TIMERx_CHxCV的值和计数器值之间间的比较结果。

互补输出

CHx_O和CHx_ON是一对互补输出通道，这两个信号不能同时有效。TIMERx有两路通道，只有一路有互补输出通道。互补信号CHx_O和CHx_ON是由一组参数来决定：TIMERx_CHCTL2寄存器中的CHxEN和CHxNEN位，TIMERx_CCHP寄存器中和TIMERx_CTL1寄存器中的POEN, ROS, IOS, ISOx和ISOxN位。输出极性由TIMERx_CHCTL2寄存器中的CHxP和CHxNP位来决定。

表 16-11. 由参数控制的互补输出表

互补参数					输出状态	
POEN	ROS	IOS	CHxEN	CHxNEN	CHx_O	CHx_ON
0	0/1	0	0	0	CHx_O / CHx_ON = LOW CHx_O / CHx_ON 输出禁用.	
				1	CHx_O = CHxP CHx_ON = CHxNP	
			1	0	CHx_O/CHx_ON 输出禁用.	
				1	如果时钟使能: CHx_O = ISOx CHx_ON = ISOxN	
		1	0	0	CHx_O = CHxP CHx_ON = CHxNP CHx_O/CHx_ON输出禁用.	
				1	CHx_O = CHxP CHx_ON = CHxNP	
			1	0	CHx_O/CHx_ON 输出使能.	
				1	如果时钟使能: CHx_O = ISOx CHx_ON = ISOxN	
1	0	0/1	0	0	CHx_O/CHx_ON = LOW CHx_O/CHx_ON 输出禁用.	
				1	CHx_O = LOW CHx_O 输出禁用.	CHx_ON=OxCPRE⊕CHxNP CHx_ON 输出使能
		0/1	1	0	CHx_O=OxCPRE ⊕ CHxP CHx_O输出使能	CHx_ON = LOW CHx_ON输出禁用.
				1	CHx_O=OxCPRE ⊕ CHxP CHx_O输出使能	CHx_ON=OxCPRE⊕CHxNP CHx_ON输出使能

互补参数					输出状态		
POEN	ROS	IOS	CHxEN	CHxNEN	CHx_O	CHx_ON	
			0	0	CHx_O = CHxP CHx_O输出禁用.	CHx_ON = CHxNP CHx_ON输出禁用.	
				1	CHx_O = CHxP CHx_O输出使能	CHx_ON=OxCPRE⊕CHxNP CHx_ON输出使能	
	1		1	0	CHx_O=OxCPRE ⊕ CHxP CHx_O输出使能	CHx_ON = CHxNP CHx_ON输出使能	
				1	CHx_O=OxCPRE ⊕ CHxP CHx_O输出使能	CHx_ON=OxCPRE⊕CHxNP CHx_ON输出使能	

死区时间插入

设置CHxEN和CHxNEN为1'b1同时设置POEN，死区插入就会被使能。DTCFG位域定义了死区时间，死区时间对通道0有效。死区时间的细节，请参考TIMERx_CCHP寄存器。

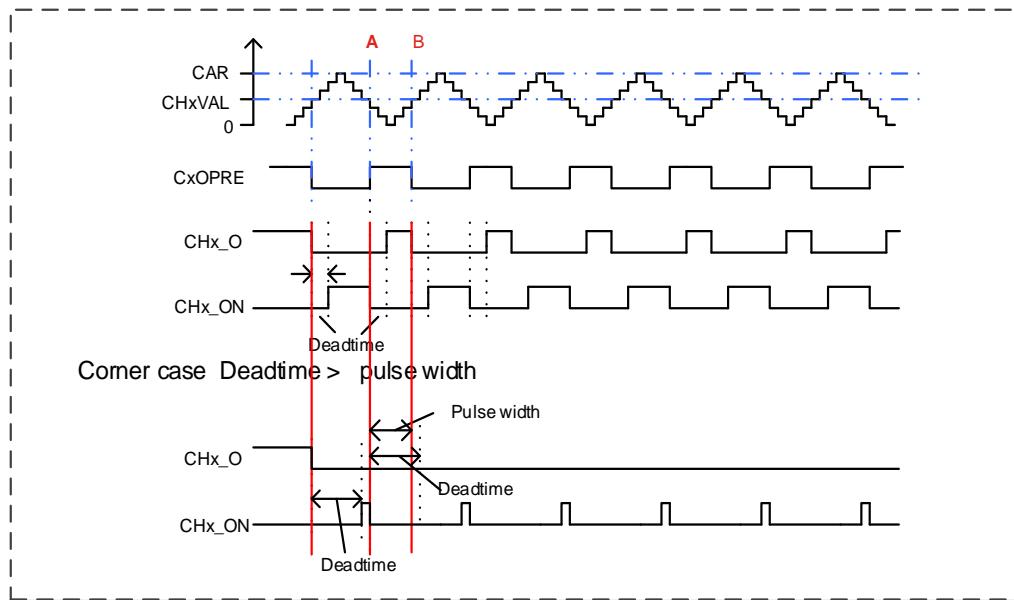
死区时间的插入，确保了通道互补的两路信号不会同时有效。

在 PWM0 模式，当通道 x 匹配发生时 (TIMERx 计数器= CHxVAL)，OxC PRE 反转。在 [图 16-82. 带死区时间的互补输出](#) 中的 A 点，CHx_O 信号在死区时间内为低电平，直到死区时间过后才变为高电平，而 CHx_ON 信号立刻变为低电平。同样，在 B 点，计数器再次匹配(TIMERx 计数器= CHxVAL)，OxC PRE 信号被清 0，CHx_O 信号被立即清零，CHx_ON 信号在死区时间内仍然是低电平，在死区时间过后才变为高电平。

有时会有一些死角事件发生，例如：

- 如果死区延时大于或者等于CHx_O信号的占空比，CHx_O信号一直为无效值(如[图 16-82. 带死区时间的互补输出](#))。
- 如果死区延时大于或者等于CHx_ON信号的占空比，CHx_ON信号一直为无效值。

图 16-82. 带死区时间的互补输出



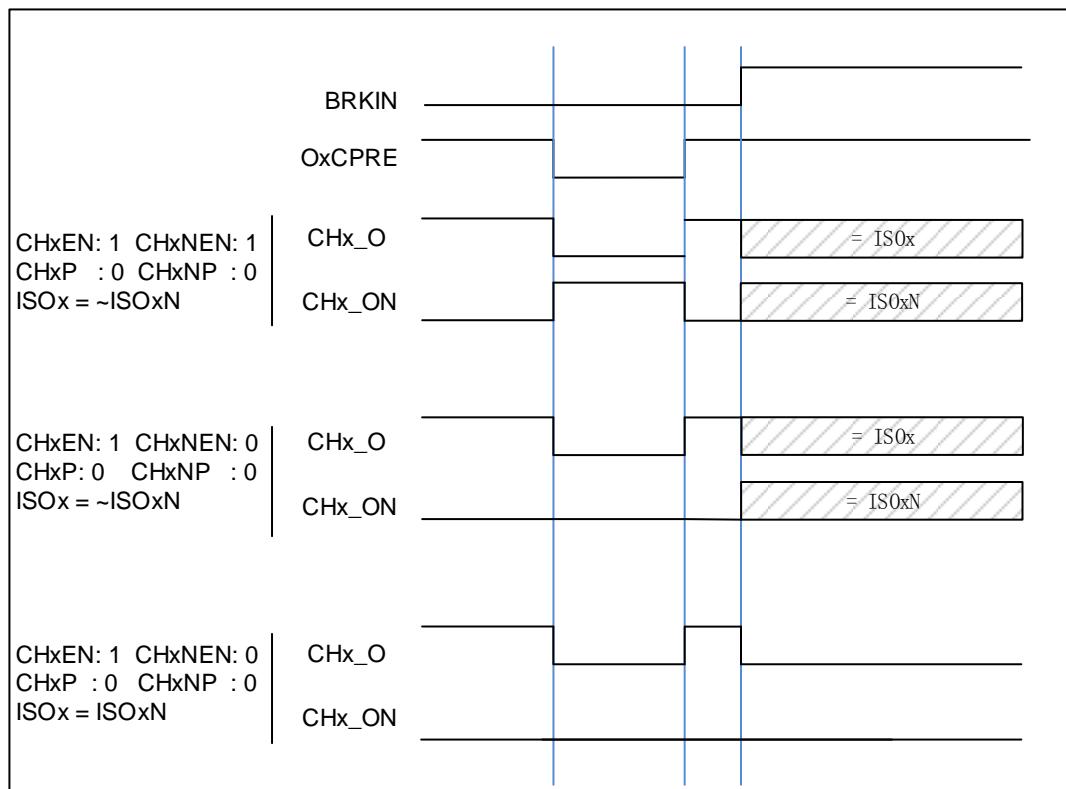
中止功能

使用中止功能时，输出CHx_O和CHx_ON信号电平被以下位控制，**TIMERx_CCHP**寄存器的POEN, IOS和ROS位，**TIMERx_CTL1**寄存器的ISOx和ISOxN位。当中止事件发生时，CHx_O和CHx_ON信号输出不能同时设置为有效电平。中止源可以选择中止输入引脚，也可以选择HXTAL时钟失效事件。时钟失效事件由RCU中的时钟监视器(CKM)产生。将**TIMERx_CCHP**寄存器的BRKEN位置1可以使能中止功能。**TIMERx_CCHP**寄存器的BRKP位决定了中止输入极性。

发生中止时，POEN位被异步清除，一旦POEN位为0，CHx_O和CHx_ON被**TIMERx_CTL1**寄存器中的ISOx位和ISOxN驱动。如果IOS=0，定时器释放输出使能，否则输出使能仍然为高。起初互补输出被置于复位状态，然后死区时间产生器重新被激活，以便在一个死区时间后驱动输出，输出电平由ISOx和ISOxN位配置。

发生中止时，**TIMERx_INTF**寄存器的BRKIF位被置1。如果BRKIE=1，中断产生。

图 16-83. 通道响应中止输入（高电平有效）时，输出信号的行为



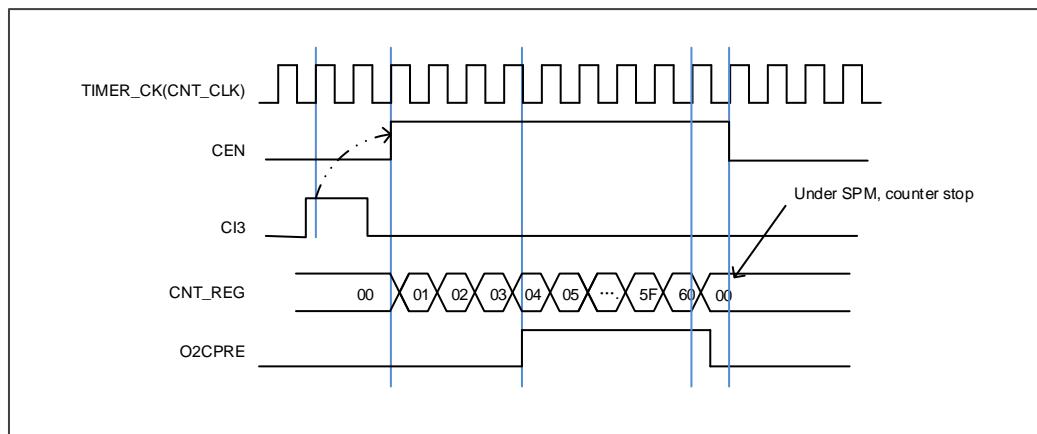
单脉冲模式

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

一旦设置定时器运行在单脉冲模式下，没有必要设置TIMERx_CTL0寄存器的定时器使能位CEN=1来使能计数器。触发信号沿或者软件写CEN=1都可以产生一个脉冲，此后CEN位一直保持为1直到更新事件发生或者CEN位被软件写0。如果CEN位被软件清0，计数器停止工作，计数值被保持。如果CEN值被硬件更新事件自动清0，计数器将被再次初始化。

在单脉冲模式下，有效的外部触发边沿会将CEN位置1，使能计数器。然而，执行计数值和TIMERx_CHxCV寄存器值的比较结果依然存在一些时钟延迟。为了最大限度减少延迟，用户可以将TIMERx_CHCTL0/1寄存器的CHxCOMFEN位置1。单脉冲模式下，触发上升沿产生之后，OxCPRE信号将被立即强制转换为与发生比较匹配时相同的电平，但是不用考虑比较结果。只有输出通道配置为PWM0或PWM1输出运行模式下时CHxCOMFEN位才可用，触发源来源于触发信号。

图 16-84. 单脉冲模式, TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60



定时器 DMA 模式

定时器DMA模式是指通过DMA模块配置定时器的寄存器。有两个跟定时器DMA模式相关的寄存器: **TIMERx_DMACFG**和**TIMERx_DMATB**。当然, 必须要使能DMA请求, 一些内部中断事件可以产生DMA请求。当中断事件发生, TIMERx会给DMA发送请求。DMA配置成M2P模式, PADDR是TIMERx_DMATB寄存器地址, DMA就会访问TIMERx_DMATB寄存器。实际上, TIMERx_DMATB寄存器只是一个缓冲, 定时器会将TIMERx_DMATB映射到一个内部寄存器, 这个内部寄存器由TIMERx_DMACFG寄存器中的DMATA来指定。如果TIMERx_DMACFG寄存器的DMATC位域值为0, 表示1次传输, 定时器的发送1个DMA请求就可以完成。如果TIMERx_DMACFG寄存器的DMATC位域值不为1, 例如其值为3, 表示4次传输, 定时器就需要再多发3次DMA请求。在这3次请求下, DMA对TIMERx_DMATB寄存器的访问会映射到访问定时器的DMATA+0x4, DMATA+0x8, DMATA+0xc寄存器。总之, 发生一次DMA内部中断请求, 定时器会连续发送(DMATC+1)次请求。

如果再来1次DMA请求事件, TIMERx将会重复上面的过程。

定时器调试模式

当Cortex™-M4内核停止, DBG_CTL1寄存器中的TIMERx_HOLD配置位被置1, 定时器计数器停止。

16.5.5. TIMERx 寄存器(x=15,16)

控制寄存器 0 (TIMERx_CTL0)

地址偏移: 0x00

复位值: 0x0000

该寄存器可以按半字(16位)或字(32位)访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					CKDIV[1:0]	ARSE	保留	SPM	UPS	UPDIS	CEN				

位/位域	名称	描述
15:10	保留	必须保持复位值
9:8	CKDIV[1:0]	<p>时钟分频</p> <p>通过软件配置CKDIV，规定定时器时钟(TIMER_CK) 与死区时间和采样时钟(DTS)之间的分频系数，死区发生器和数字滤波器会用到DTS时间。</p> <ul style="list-style-type: none"> 00: fDTS=fTIMER_CK 01: fDTS= fTIMER_CK /2 10: fDTS= fTIMER_CK /4 11: 保留
7	ARSE	<p>自动重载影子使能</p> <p>0: 禁能TIMERx_CAR寄存器的影子寄存器</p> <p>1: 使能TIMERx_CAR寄存器的影子寄存器</p>
6:4	保留	必须保持复位值
3	SPM	<p>单脉冲模式</p> <p>0: 更新事件发生后，计数器继续计数</p> <p>1: 在下一次更新事件发生时，CEN硬件清零并且计数器停止计数</p>
2	UPS	<p>更新请求源</p> <p>软件配置该为，选择更新事件源.</p> <p>0: 使能后，下述任一事件产生更新中断或DMA请求:</p> <ul style="list-style-type: none"> - UPG位被置1 - 计数器溢出/下溢 - 从模式控制器产生的更新 <p>1: 使能后只有计数器溢出/ 下溢才产生更新中断或DMA请求</p>
1	UPDIS	<p>禁止更新.</p> <p>该位用来使能或禁能更新事件的产生 .</p> <p>0: 更新事件使能.当以下事件之一发生时，更新事件产生，具有缓存的寄存器被装入它们的预装载值:</p> <ul style="list-style-type: none"> - UPG位被置1 - 计数器溢出/下溢 - 从模式控制器产生一个更新事件 <p>1: 更新事件禁能. 带有缓存的寄存器保持原有值，如果UPG位被置1或者从模式控制器产生一个硬件复位事件，计数器和预分频器被重新初始化</p>
0	CEN	<p>计数器使能</p> <p>0: 计数器禁能</p> <p>1: 计数器使能</p> <p>在软件将CEN位置1后，外部时钟、暂停模式和编码器模式才能工作。触发模式可以自动地通过硬件设置CEN位。</p>

控制寄存器 1 (TIMERx_CTL1)

地址偏移: 0x04

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				ISOON		ISOO		保留		DMAS		CCUC		保留	CCSE
						rw	rw				rw		rw		rw

位/位域	名称	描述
15:10	保留	必须保持复位值
9	ISOON	<p>通道0的互补通道空闲状态输出</p> <p>0: 当POEN复位, CH0_ON设置低电平.</p> <p>1: 当POEN复位, CH0_ON设置高电平</p> <p>此位只有在TIMERx_CCHP寄存器的PROT [1:0]位为00的时候可以被更改.</p>
8	ISOO	<p>通道0的空闲状态输出</p> <p>0: 当POEN复位, CH0_O设置低电平</p> <p>1: 当POEN复位, CH0_O设置高电平</p> <p>如果CH0_ON生效, 一个死区时间后CH0_O输出改变。此位只有在TIMERx_CCHP寄存器的PROT [1:0]位为00的时候可以被更改.</p>
7:4	保留	必须保持复位值
3	DMAS	<p>DMA请求源选择</p> <p>0: 当通道捕获/比较事件发生时, 发送通道x的DMA请求 .</p> <p>1: 当更新事件发生, 发送通道x的DMA请求</p>
2	CCUC	<p>换相控制影子寄存器更新控制</p> <p>当换相控制影子寄存器 (CHxEN, CHxNEN和CHxCOMCTL位) 使能(CCSE=1), 这些影子寄存器更新控制如下:</p> <p>0: CMTG位被置1时更新影子寄存器</p> <p>1: 当CMTG位被置1或检测到TRIGI上升沿时, 影子寄存器更新</p> <p>当通道没有互补输出时, 此位无效。</p>
1	保留	必须保持复位值.
0	CCSE	<p>换相控制影子使能</p> <p>0: 影子寄存器CHxEN, CHxNEN和CHxCOMCTL位禁能.</p> <p>1: 影子寄存器CHxEN, CHxNEN和CHxCOMCTL位使能.</p> <p>如果这些位已经被写入了, 换相事件到来时这些位才被更新</p> <p>当通道没有互补输出时, 此位无效</p>

DMA 和中断使能寄存器 (TIMERx_DMAINTEN)

地址偏移: 0x0C

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		保留				CH0DEN	UPDEN	BRKIE	保留	CMTIE		保留		CHOIE	UPIE

rw rw rw rw rw rw rw

位/位域	名称	描述
15:10	保留	必须保持复位值.
9	CH0DEN	通道0比较/捕获 DMA请求使能 0: 禁止通道0比较/捕获DMA请求 1: 使能通道0比较/捕获DMA请求
8	UPDEN	更新DMA请求使能 0: 禁止更新DMA请求 1: 使能更新DMA请求
7	BRKIE	中止中断使能 0: 禁止中止中断 1: 使能中止中断
5	CMTIE	换相更新中断使能 0: 禁止换相更新中断 1: 使能换相更新中断
4:2	保留	必须保持复位值.
1	CHOIE	通道0比较/捕获中断使能 0: 禁止通道0中断 1: 使能通道0中断
0	UPIE	更新中断使能 0: 禁止更新中断 1: 使能更新中断

中断标志寄存器 (TIMERx_INTF)

地址偏移: 0x10

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		保留				CH0OF	保留.	BRKIF	保留	CMTIF		保留.		CHOIF	UPIF

rc_w0 rc_w0 rc_w0 rc_w0 rc_w0 rc_w0

位/位域	名称	描述
15:10	保留	必须保持复位值.
9	CH0OF	通道0捕获溢出标志 当通道0被配置为输入模式时, 在CHOIF标志位已经被置1后, 捕获事件再次发生时, 该标志位可以由硬件置1。该标志位由软件清0.

		0: 无捕获溢出中断发生 1: 发生了捕获溢出中断
8	保留	必须保持复位值.
7	BRKIF	中止中断标志位 一旦中止输入有效，由硬件对该位置‘1’。如果中止输入无效，则该位可由软件清‘0’。 0: 无中止事件产生 1: 中止输入上检测到有效电平
5	CMTIF	通道换相更新中断标志 当通道换相更新事件发生时此标志位被硬件置1，此位由软件清0。 0: 无通道换相更新中断发生 1: 通道换相更新中断发生
4:2	保留	必须保持复位值.
1	CH0IF	通道0比较/捕获中断标志 此标志由硬件置1软件清0。当通道0在输入模式下时，捕获事件发生时此标志位被置1； 当通道0在输出模式下时，此标志位在一个比较事件发生时被置1。 0: 无通道0中断发生 1: 通道0中断发生
0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置1，软件清0。 0: 无更新中断发生 1: 发生更新中断

软件事件产生寄存器 (**TIMERx_SWEVG**)

地址偏移: 0x14

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问



位/位域	名称	描述
15:8	保留	必须保持复位值.
7	BRKG	产生中止事件 该位由软件置1，用于产生一个中止事件，由硬件自动清0。当此位被置1时，POEN位被清0且BRKIF位被置1，若开启对应的中断和DMA，则产生相应的中断和DMA传输。 0: 不产生中止事件 1: 产生中止事件
5	CMTG	通道换相更新事件发生

此位由软件置1，由硬件自动清0。当此位被置1，通道捕获/比较控制寄存器（CHxEN, CHxNEN 和CHxCOMCTL）的互补输出被更新（根据TIMERx_CTL1中CCSE值）。

0：不产生通道控制更新事件

1：产生通道控制更新事件

4:2 保留 必须保持复位值。

1 CH0G 通道0捕获或比较事件发生

该位由软件置1，用于在通道0产生一个捕获/比较事件，由硬件自动清0。当此位被置1，CH0IF标志位被置1，若开启对应的中断和DMA，则发出相应的中断和DMA请求。此外，如果通道0配置为输入模式，计数器的当前值被TIMERx_CH0CV寄存器捕获，如果CH0IF标志位已经为1，则CH0OF标志位被置1。

0：不产生通道0捕获或比较事件

1：发生通道0捕获或比较事件

0 UPG 更新事件产生

此位由软件置1，被硬件自动清0。当此位被置1，如果选择了中央对齐或向上计数模式，计数器被清0。否则(向下计数模式)计数器将载入自动重载值，预分频计数器将同时被清除。

0：无更新事件产生

1：产生更新事件

通道控制寄存器 0 (TIMERx_CHCTL0)

地址偏移：0x18

复位值：0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								保留	CH0COMCTL[2:0]	CH0COMSEN	CH0COMFEN	CH0MS[1:0]			
CH0CAPFLT[3:0]								CH0CAPPSC[1:0]							

rw

rw

rw

输出比较模式：

位/位域	名称	描述
15:7	保留	必须保持复位值。
6:4	CH0COMCTL[2:0]	<p>通道0输出比较模式</p> <p>此位定义了输出参考信号O0CPRE的动作，而O0CPRE决定了CH0_O、CH0_ON的值。O0CPRE高电平有效，而CH0_O、CH0_ON的有效电平取决于CH0P、CH0NP位。</p> <p>000：冻结。输出比较寄存器TIMERx_CH0CV与计数器TIMERx_CNT间的比较对O0CPRE不起作用</p> <p>001：匹配时设置为高。当计数器的值与捕获/比较值寄存器TIMERx_CH0CV相同时，强制O0CPRE为高。</p>

010: 匹配时设置为低。当计数器的值与捕获/比较值寄存器TIMERx_CH0CV相同时，强制O0CPRE为低。

011: 匹配时翻转。当计数器的值与捕获/比较值寄存器TIMERx_CH0CV相同时，强制O0CPRE翻转。

100: 强制为低。强制O0CPRE为低电平

101: 强制为高。强制O0CPRE为高电平

110: PWM 模式0(EAPWM)。在向上计数时，一旦计数器值小于TIMERx_CH0CV时，O0CPRE为高电平，否则为低电平。在向下计数时，一旦计数器的值大于TIMERx_CH0CV时，O0CPRE 为低电平，否则为高电平。

111: PWM 模式1(CAPWM)。在向上计数时，一旦计数器值小于TIMERx_CH0CV时，O0CPRE为低电平，否则为高电平。在向下计数时，一旦计数器的值大于TIMERx_CH0CV时，O0CPRE为高电平，否则为低电平。

在PWM模式0或PWM模式1中，只有当比较结果改变了或者输出比较模式中从冻结模式切换到PWM模式时，O0CPRE电平才改变。

当TIMERx_CCHP寄存器的PROT [1:0]=11且CH0MS =00（比较模式）时此位不能被改变。

3	CH0COMSEN	通道0输出比较影子寄存器使能 当此位被置1，TIMERx_CH0CV寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0: 禁止通道0输出/比较影子寄存器 1: 使能通道0输出/比较影子寄存器 仅在单脉冲模式下(TIMERx_CTL0寄存器的SPM =1)，可以在未确认预装载寄存器情况下使用PWM 模式 当TIMERx_CCHP寄存器的PROT [1:0]=11且CH0MS =00时此位不能被改变。
2	CH0COMFEN	通道0输出比较快速使能 当该位为1时，如果通道配置为PWM0模式或者PWM1模式，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH0_O被设置为比较电平而与比较结果无关。 0: 禁止通道0输出比较快速. 当触发器的输入有一个有效沿时，激活CH0_O输出的最小延时为5个时钟周期 1: 使能通道0输出比较快速。当触发器的输入有一个有效沿时，激活CH0_O输出的最小延时为3个时钟周期
1:0	CH0MS[1:0]	通道0 I/O模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭(TIMERx_CHCTL2寄存器的CH0EN位被清0)时这些位才可写。 00: 通道0配置为输出 01: 通道0配置为输入，ISO映射在CI0FE0上 10: 通道0配置为输入，ISO映射在CI1FE0上 11: 通道0配置为输入，ISO映射在ITS上. 此模式仅工作在内部触发输入被选中时(通过设置TIMERx_SMCFGFG寄存器的TRGS位)

输入捕获模式:

位/位域	名称	描述
15:8	保留	必须保持复位值.
7:4	CH0CAPFLT[3:0]	<p>通道0输入捕获滤波控制</p> <p>数字滤波器由一个事件计数器组成，它记录N个输入事件后会产生一个输出的跳变。这些位定义了C10输入信号的采样频率和数字滤波器的长度。</p> <p>0000: 无滤波器, fSAMP= fDTS, N=1</p> <p>0001: fSAMP= fPCLK, N=2</p> <p>0010: fSAMP= fPCLK, N=4</p> <p>0011: fSAMP= fPCLK, N=8</p> <p>0100: fSAMP=fDTS/2, N=6</p> <p>0101: fSAMP=fDTS/2, N=8</p> <p>0110: fSAMP=fDTS/4, N=6</p> <p>0111: fSAMP=fDTS/4, N=8</p> <p>1000: fSAMP=fDTS/8, N=6</p> <p>1001: fSAMP=fDTS/8, N=8</p> <p>1010: fSAMP=fDTS/16, N=5</p> <p>1011: fSAMP=fDTS/16, N=6</p> <p>1100: fSAMP=fDTS/16, N=8</p> <p>1101: fSAMP=fDTS/32, N=5</p> <p>1110: fSAMP=fDTS/32, N=6</p> <p>1111: fSAMP=fDTS/32, N=8</p>
3:2	CH0CAPPSC[1:0]	<p>通道0输入捕获预分频器</p> <p>这2位定义了通道0输入的预分频系数。当TIMERx_CHCTL2寄存器中的CH0EN =0时，则预分频器复位。</p> <p>00: 无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获</p> <p>01: 每2个事件触发一次捕获</p> <p>10: 每4个事件触发一次捕获</p> <p>11: 每8个事件触发一次捕获</p>
1:0	CH0MS[1:0]	<p>通道0模式选择</p> <p>与输出比较模式相同</p>

通道控制寄存器 2 (TIMERx_CHCTL2)

地址偏移: 0x20

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
保留.														CH0NP	CH0NEN	CH0P	CH0EN

位/位域	名称	描述
15:4	保留	必须保持复位值

3	CH0NP	通道0互补输出极性 当通道0配置为输出模式，此位定义了互补输出信号的极性。 0：通道0高电平有效 1：通道0低电平有效 当通道0配置为输入模式时，此位和CH0P联合使用，作为输入信号Cl0的极性选择控制信号。 当TIMERx_CCHP寄存器的PROT [1:0]=11或10 时此位不能被更改。
2	CH0NEN	通道0互补输出使能 当通道0配置为输出模式时，将此位置1使能通道0的互补输出。 0：禁止通道0互补输出 1：使能通道0互补输出
1	CH0P	通道0极性 当通道0配置为输出模式时，此位定义了输出信号极性。 0：通道0高电平有效 1：通道0低电平有效 当通道0配置为输入模式时，此位定义了Cl0信号极性 [CH0NP, CH0P] 将选择Cl0FE0或者Cl1FE0的有效边沿或者捕获极性 [CH0NP==0, CH0P==0]: 把ClxFE0的上升沿作为捕获或者从模式下触发的有效信号，并且ClxFE0不会被翻转。 [CH0NP==0, CH0P==1]: 把ClxFE0的下降沿作为捕获或者从模式下触发的有效信号，并且ClxFE0会被翻转。 [CH0NP==1, CH0P==0]: 保留。 [CH0NP==1, CH0P==1]: 把ClxFE0的上升沿和下降沿都作为捕获或者从模式下触发的有效信号，并且ClxFE0不会被翻转。 当TIMERx_CCHP寄存器的PROT [1:0]=11或10 时此位不能被更改。
0	CH0EN	通道0捕获/比较使能 当通道0配置为输出模式时，将此位置1使能CH0_O信号有效。当通道0配置为输入模式时，将此位置1使能通道0上的捕获事件。 0：禁止通道0 1：使能通道0

计数器寄存器 (TIMERx_CNT)

地址偏移: 0x24

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

位/位域	名称	描述
------	----	----

15:0 CNT[15:0] 这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器 (TIMERx_PSC)

地址偏移: 0x28

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

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

计数器自动重载寄存器 (TIMERx_CAR)

地址偏移: 0x2C

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

位/位域	名称	描述
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

重复计数寄存器 (TIMERx_CREP)

地址偏移: 0x30

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

位/位域	名称	描述
15:8	保留	必须保持复位值。

7:0 CREP[7:0] 重复计数器的值

这些位定义了更新事件的产生速率。重复计数器计数值减为0时产生更新事件。影子寄存器的更新速率也会受这些位影响(前提是影子寄存器被使能)。

通道 0 捕获/比较寄存器 (**TIMERx_CH0CV**)

地址偏移: 0x34

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															

rw

位/位域	名称	描述
15:0	CH0VAL[15:0]	<p>通道0的捕获或比较值</p> <p>当通道0配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。</p> <p>当通道0配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。</p>

互补通道保护寄存器 (**TIMERx_CCHP**)

地址偏移: 0x44

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POEN	OAEN	BRKP	BRKEN	ROS	IOS	PROT[1:0]									DTCFG[7:0]

rw rw

位/位域	名称	描述
15	POEN	<p>所有的通道输出使能</p> <p>根据OAEN位，该位可以软件设置或者硬件自动设置。一旦中止输入有效，该位被硬件异步清0。如果一个通道配置为输出模式，如果设置了相应的使能位(TIMERx_CHCTL2寄存器的CHxEN, CHxNEN位)，则开启CHx_O 和CHx_ON 输出。</p> <p>0: 禁止通道输出或强制为空闲状态</p> <p>1: 使能通道输出</p>
14	OAEN	<p>自动输出使能</p> <p>此位定义了POEN位是否可以被硬件自动置1。</p> <p>0: POEN位不能被硬件置1</p> <p>1: 如果中止输入无效，下一次更新事件发生时，POEN位能被硬件自动置1</p> <p>此位只有在TIMERx_CCHP寄存器的PROT [1:0] =00时才可修改。</p>
13	BRKP	中止极性

		此位定义了中止输入信号BKIN的极性。
		0: 中止输入低电平有效 1: 中止输入高电平有效
12	BRKEN	中止使能 此位置1使能中止事件和CCS时钟失败事件输入。 0: 禁能中止输入 1: 使能中止输入 此位只有在TIMERx_CCHP寄存器的PROT [1:0] =00时才可修改。.
11	ROS	运行模式下“关闭状态”配置 当POEN位被置1，此位定义了通道(带有互补输出且配置为输出模式)的输出状态。 0: 当POEN位被置1，通道输出信号 (CHx_O/ CHx_ON)被禁止 1: 当POEN位被置1，通道输出信号 (CHx_O/ CHx_ON)被使能，和TIMER0_CHCTL2 寄存器CHxEN/CHxNEN位有关 此位在TIMERx_CCHP寄存器的PROT [1:0]=10或11时不能被更改。
10	IOS	空闲模式下“关闭状态”配置 当POEN位被清0，此位定义了已经配置为输出模式的通道的输出状态。 0: 当POEN位被清0，通道输出信号(CHx_O/ CHx_ON)被禁止 1: 当POEN位被清0，通道输出信号(CHx_O/ CHx_ON)被使能，和TIMERx_CHCTL2寄存器CHxEN/CHxNEN位有关 此位在TIMERx_CCHP寄存器的PROT [1:0]=10或11时不能被更改。.
9:8	PROT[1:0]	互补寄存器保护控制 这两位定义了寄存器的写保护特性。 00: 禁能保护模式。无写保护. 01: PROT模式0。TIMERx_CTL1寄存器中ISOx/ISOxN 位，TIMERx_CCHP 寄存器中 BRKEN/BRKP/OAEN/DTCFG 位写保护 10: PROT模式1。除了PROT模式0下的寄存器写保护外，还有TIMERx_CHCTL2寄存器中CHxP/CHxNP位（如果相应通道配置为输出模式），TIMERx_CCHP 寄存器中ROS/IOS 位。 11: PROT模式2。除了PROT模式1下的寄存器写保护外，还有TIMERx_CHCTRL0/1 中 CHxCOMCTL/ CHxCOMSEN位（如果相关通道配置为输出模式）写保护。 系统复位后这两位只能被写一次，一旦TIMERx_CCHP寄存器被写入，这两位被写保护
7:0	DTCFG[7:0]	死区时间控制 这些位定义了插入互补输出之间的死区持续时间。DTCFG值和死区时间的关系如下： DTCFG [7:5] =3'b0xx: DTvalue =DTCFG [7:0]x tDT, tDT=tDTS. DTCFG [7:5] =3'b 10x: DTvalue = (64+DTCFG [5:0])xtDT, tDT =tDTS*2. DTCFG [7:5] =3'b 110: DTvalue = (32+DTCFG [4:0])xtDT, tDT=tDTS*8. DTCFG [7:5] =3'b 111: DTvalue = (32+DTCFG [4:0])xtDT, tDT =tDTS*16. 此位只有在TIMERx_CCHP寄存器的PROT [1:0]=00时才可修改。

DMA 配置寄存器 (TIMERx_DMACFG)

地址偏移: 0x48

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				DMATC[4:0]				保留				DMATA [4:0]			

rw

rw

位/位域	名称	描述
15:13	保留	必须保持复位值.
12:8	DMATC [4:0]	DMA传输计数 该位域定义了DMA访问（读写）TIMERx_DMAVB寄存器的数量
7:5	保留	必须保持复位值
4:0	DMATA [4:0]	DMA传输起始地址 该位域定义了DMA访问TIMERx_DMAVB寄存器的第一个地址。当通过TIMERx_DMA第一次访问时，访问的就是该位域指定的地址。第二次访问TIMERx_DMAVB时，将访问起始地址+0x4。 5'b0_0000: TIMERx_CTL0 5'b0_0001: TIMERx_CTL1 ... 总之： 起始地址 = TIMERx_CTL0 + DMATA*4

DMA 发送缓冲区寄存器 (TIMERx_DMATB)

地址偏移: 0x4C

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

rw

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

配置寄存器 (TIMERx_CFG)

地址偏移: 0xFC

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

rw rw

位/位域	名称	描述
15:2	保留	必须保持复位值
1	CHVSEL	<p>写捕获比较寄存器选择位 此位由软件写1或清0。 1: 当写入捕获比较寄存器的值与寄存器当前值相等时，写入操作无效 0: 无影响</p>
0	OUTSEL	<p>输出值选择位 此位由软件写1或清0。 1: 如果POEN位与IOS位均为0，则输出无效 0: 无影响</p>

16.6. 基本定时器 (TIMERx, x=5)

16.6.1. 简介

基本定时器(Timer5)包含一个无符号16位计数器。可以被用作通用定时器和为DAC (数字到模拟转换器)提供时钟。基本定时器可以配置产生DMA请求，TRGO触发连接到DAC。

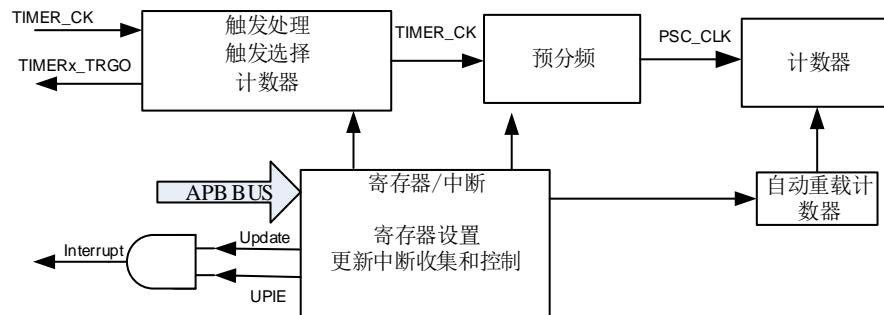
16.6.2. 主要特性

- 计数器宽度: 16位
- 时钟源只有内部时钟
- 计数模式: 向上计数
- 可编程的预分频器: 16位, 运行时可以被改变
- 自动重装载功能.
- 中断输出和DMA请求: 更新事件

16.6.3. 结构框图

[图 16-85. 基本定时器结构框图](#)提供了基本定时器内部配置的细节

图 16-85. 基本定时器结构框图



16.6.4. 功能描述

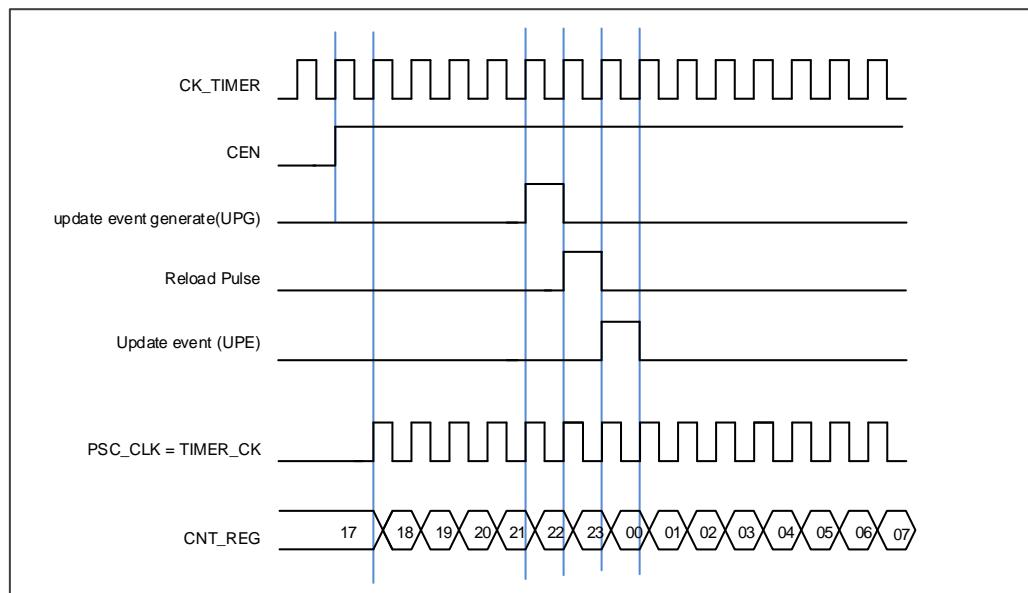
时钟源选择

基本定时器可以由内部时钟源CK_TIMER驱动。

基本定时器时钟内部连接到TIMER_CK。

基本定时器仅有一个时钟源TIMER_CK, 用来驱动计数器预分频器。当CEN置位, TIMER_CK 经过预分频器 (预分频值由TIMERx_PSC寄存器确定) 产生PSC_CLK。

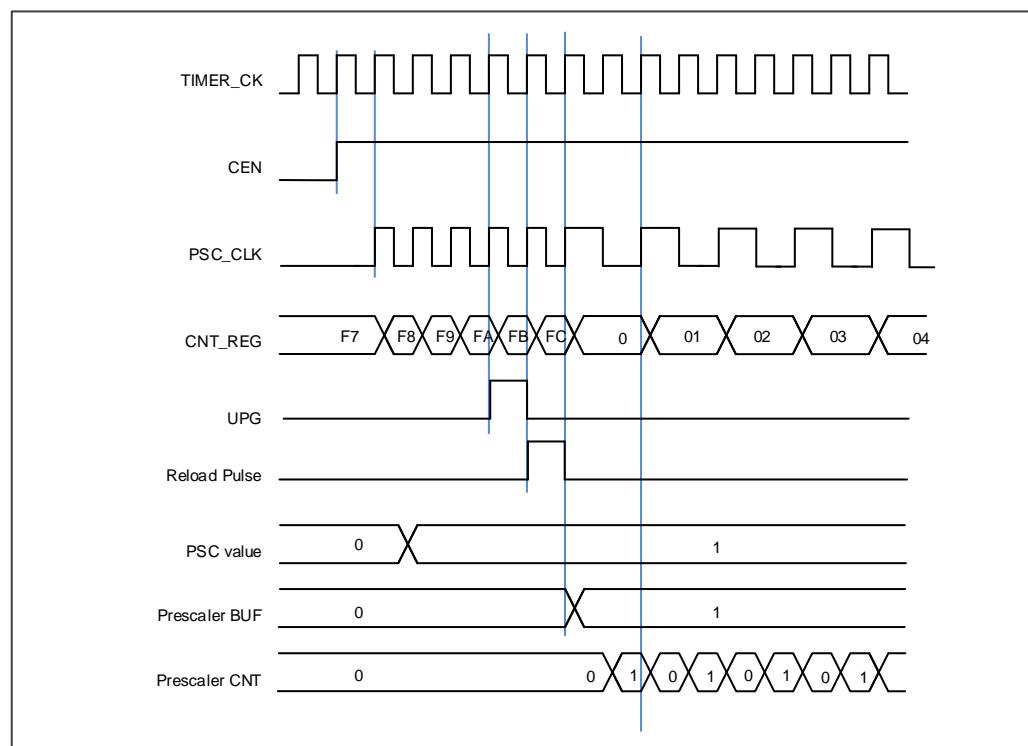
图 16-86. 内部时钟分频为 1 时正常模式下的控制电路



预分频

预分频器可以将定时器的时钟 (TIMER_CK) 频率按 1 到 65536 之间的任意值分频，分频后的时钟 PSC_CLK 驱动计数器计数。分频系数受预分频寄存器 TIMERx_PSC 控制，这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下一次更新事件到来时被采用。

图 16-87. 当预分频器的参数从 1 变到 2 时，计数器的时序图



向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从0开始向上连续计数到自动加载值（定义在TIMERx_CAR寄存器中），一旦计数器计数到自动加载值，会重新从0开始向上计数并产生上溢事件。在向上计数模式中，TIMERx_CTL0寄存器中的计数方向控制位DIR应该被设置成0。

当通过TIMERx_SWEVG寄存器的UPG位置1来设置更新事件时，计数值会被清0，并产生更新事件。

如果TIMERx_CTL0寄存器的UPDIS置1，则禁止更新事件。

当发生更新事件时，所有的寄存器(重复计数器，自动重载寄存器，预分频寄存器)都将被更新。

下面这些图给出了一些例子，当TIMERx_CAR=0x63时，计数器在不同预分频因子下的行为。

图 16-88. 向上计数时序图，PSC=0/1

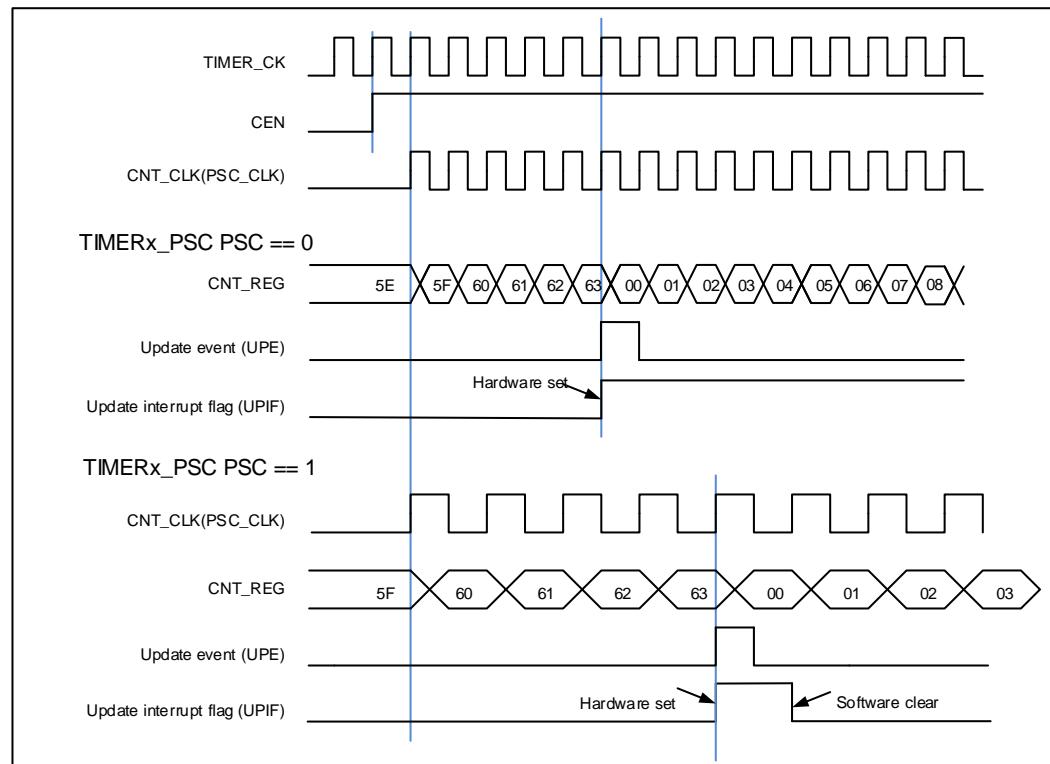
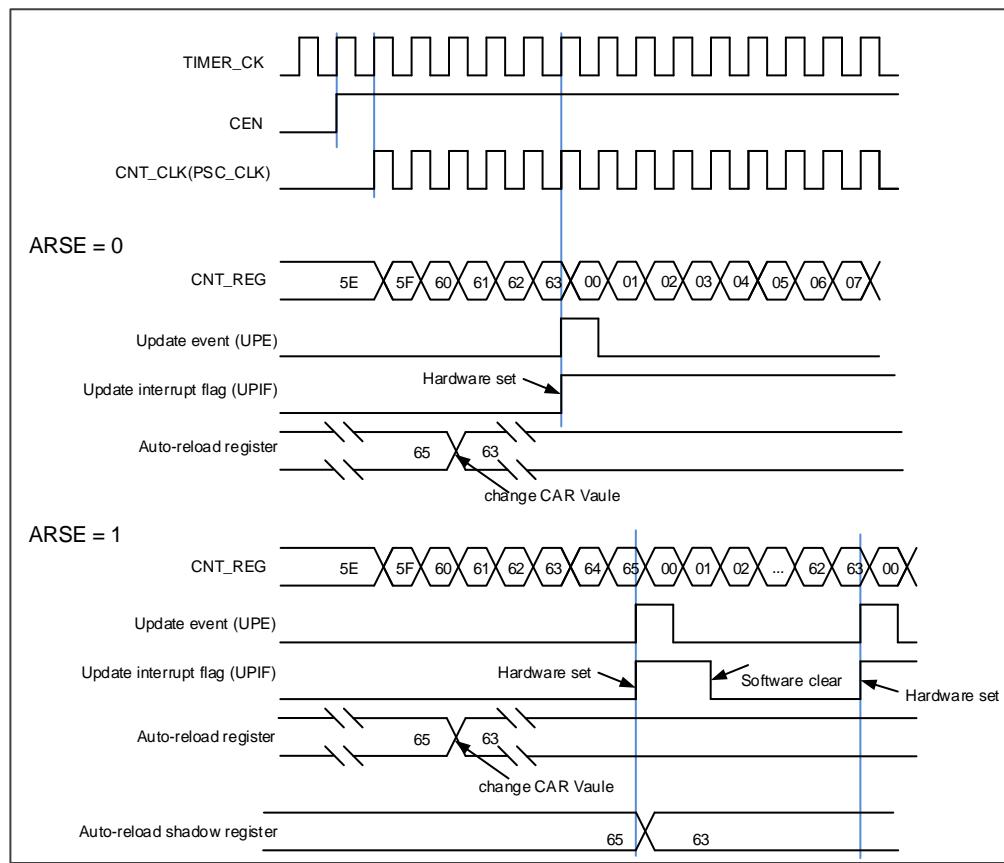


图 16-89. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



定时器调试模式

当Cortex™-M4内核停止，DBG_CTL0寄存器中的TIMERx_HOLD配置位被置1，定时器计数器停止。

16.6.5. TIMERx 寄存器(x=5)

控制寄存器 0 (TIMERx_CTL0)

地址偏移: 0x00

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问



位/位域	名称	描述
15:8	保留	必须保持复位值
7	ARSE	自动重载影子使能 0: 禁能TIMERx_CAR寄存器的影子寄存器

		1: 使能TIMERx_CAR寄存器的影子寄存器
6:4	保留	必须保持复位值
3	SPM	<p>单脉冲模式</p> <p>0: 更新事件发生后, 计数器继续计数</p> <p>1: 在下一次更新事件发生时, CEN硬件清零并且计数器停止计数</p>
2	UPS	<p>更新请求源</p> <p>软件配置该为, 选择更新事件源.</p> <p>0: 使能后, 下述任一事件产生更新中断或DMA请求:</p> <ul style="list-style-type: none"> - UPG位被置1 - 计数器溢出/下溢 - 溢从模式控制器产生的更新 <p>1: 使能后只有计数器溢出/下溢才产生更新中断或DMA请求。</p>
1	UPDIS	<p>禁止更新.</p> <p>该位用来使能或禁能更新事件的产生 .</p> <p>0: 更新事件使能. 当以下事件之一发生时, 更新事件产生, 具有缓存的寄存器被装入它们的预装载值:</p> <ul style="list-style-type: none"> - UPG位被置1 - 计数器溢出/下溢 - 从模式控制器产生一个更新事件 <p>1: 更新事件禁能. 带有缓存的寄存器保持原有值, 如果UPG位被置1或者从模式控制器产生一个硬件复位事件, 计数器和预分频器被重新初始化</p>
0	CEN	<p>计数器使能</p> <p>0: 计数器禁能</p> <p>1: 计数器使能</p> <p>在软件将CEN位置1后, 外部时钟、暂停模式和编码器模式才能工作。触发模式可以自动地通过硬件设置CEN位。</p>

控制寄存器 1 (TIMERx_CTL1)

地址偏移: 0x04

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

rw

位/位域	名称	描述
15:7	保留	必须保持复位值
6:4	MMC[2:0]	这些位控制TRGO信号的选择, TRGO信号由主定时器发给从定时器用于同步功能

000: 复位。TIMERx_SWEVG寄存器的UPG位被置1或从模式控制器产生复位触发一次TRGO脉冲，后一种情况下，TRGO上的信号相对实际的复位会有一个延迟。

001: 使能。此模式可用于同时启动多个定时器或控制在一段时间内使能从定时器。主模式控制器选择计数器使能信号作为触发输出TRGO。当CEN控制位被置1或者暂停模式下触发输入为高电平时，计数器使能信号被置1。在暂停模式下，计数器使能信号受控于触发输入，在触发输入和TRGO 上会有一个延迟，除非选择了主/从模式。

010: 更新。主模式控制器选择更新事件作为TRGO。

3:0 保留 必须保持复位值

DMA 和中断使能寄存器 (TIMERx_DMAINTEN)

地址偏移: 0x0C

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

位/位域	名称	描述
15:9	保留	必须保持复位值
8	UPDEN	更新DMA请求使能 0: 禁止更新DMA请求 1: 使能更新DMA请求
7:1	保留	必须保持复位值
0	UPIE	更新中断使能 0: 禁止更新中断 1: 使能更新中断

中断标志寄存器 (TIMERx_INTF)

地址偏移: 0x10

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								UPIF							
rc_w0															

位/位域	名称	描述
15:1	保留	必须保持复位值

0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置1，软件清0。 0：无更新中断发生 1：发生更新中断
---	------	--

软件事件产生寄存器 (TIMERx_SWEVG)

地址偏移: 0x14

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

位/位域	名称	描述
15:1	保留	必须保持复位值
0	UPG	<p>更新事件产生</p> <p>此位由软件置1，被硬件自动清0。当此位被置1并且向上计数模式，计数器被清0，预分频计数器将同时被清除。</p> <p>0：无更新事件产生 1：产生更新事件</p>

计数器寄存器 (TIMERx_CNT)

地址偏移: 0x24

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

位/位域	名称	描述
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器 (TIMERx_PSC)

地址偏移: 0x28

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问

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

PSC[15:0]

rw

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

计数器自动重载寄存器 (TIMERx_CAR)

地址偏移: 0x2C

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问

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

rw

位/位域	名称	描述
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

17. 红外线接口 (IFRP)

17.1. 简介

红外线接口 (IFRP) 用来控制发射红外光的LED，该LED可发射红外数据来实现红外遥控。

该模块无寄存器，受TIMER15定时器和TIMER16定时器控制。通过设置GPIO引脚为高速模式，可以提高模块的输出电流能力。

17.2. 主要特性

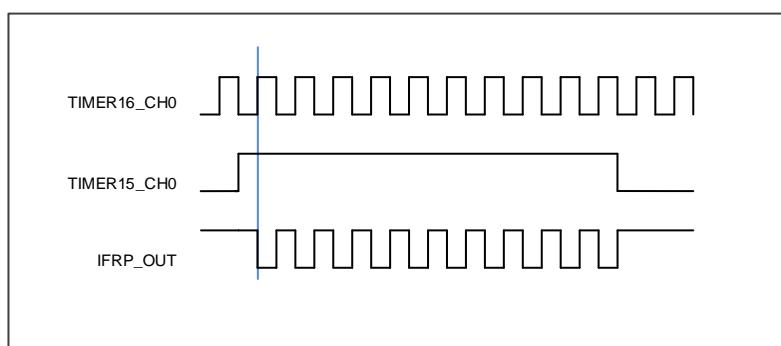
- IFRP输出信号由TIMER15定时器的通道0和TIMER16定时器的通道0决定
- 为了获取正确的红外信号，TIMER15定时器应该产生低频调制包络信号，TIMER16应该产生高频载波信号
- 通过设置SYSCFG_CFG0中的PB9_HCCE，红外线接口输出（PB9）能够提供高电流输出驱动LED接口

17.3. 功能描述

IFRP 模块整合了 TIMER15 定时器和 TIMER16 定时器的输出来产生红外信号。

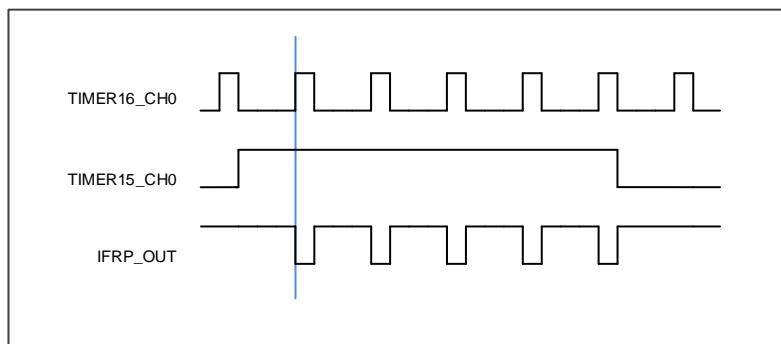
1. 通过对 TIMER15 定时器的通道 0 编程输出低频 PWM 信号来产生调制包络信号，对 TIMER16 定时器的通道 0 编程输出高频 PWM 信号来产生载波信号。产生信号之前需要开启这些通道。
2. 配置 GPIO 口为复用，并使能这些引脚。
3. 如果你想获得更高的驱动电流输出，需要将 IFRP_OUT 映射到 PB9 口上，并且在 SYS_CFG 模式下通过相应寄存器设置 PB9 口为高速模式。

图 17-1. IFRP 输出时序图 1



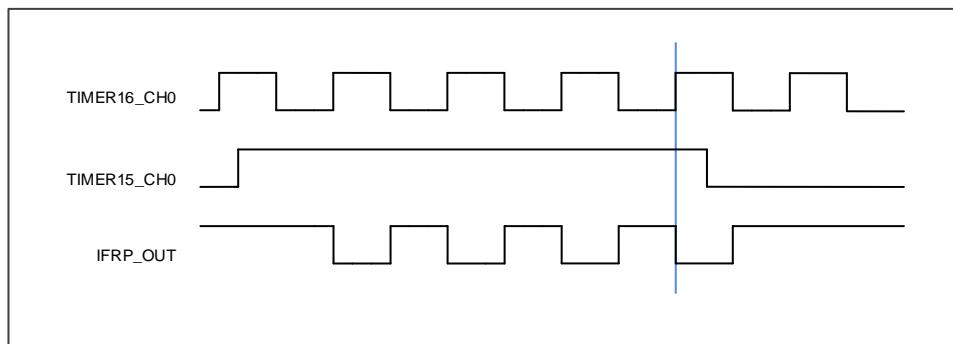
注：IFRP_OUT比TIMER16_CH0时钟通道1输出信号有一个APB时钟延迟。

图 17-2. IFRP 输出时序图 2



注：载波（TIMER15_CH0）的占空比可以被改变，在TIMER15定时器的通道0输出电平为高时，IFRP_OUT和TIMER16定时器的通道0输出信号反相。

图 17-3. IFRP 输出时序图 3



注：IFRP_OUT将保持TIMER16定时器通道0输出信号的完整，即使TIMER15定时器的包络信号无效。

18. 通用同步异步收发器 (USART)

18.1. 简介

通用同步异步收发器(USART)提供了一个灵活方便的串行数据交换接口。数据帧可以通过全双工或半双工，同步或异步的方式进行传输。USART提供了可编程的波特率发生器，能对系统时钟进行分频产生USART发送和接收所需的特定频率。

USART不仅支持标准的异步收发模式，还实现了一些其他类型的串行数据交换模式，如红外编码规范，SIR，智能卡协议，LIN，以及同步单双工模式。它还支持多处理器通信和Modem流控操作(CTS/RTS)。数据帧支持从LSB或者MSB开始传输。数据位的极性和TX/RX引脚都可以灵活配置。

所有USART都支持DMA功能，来实现高速率的数据通信。

18.2. 主要特性

- NRZ标准格式(Mark/Space)
- 全双工异步通信
- 半双工单线通信
- 接收FIFO功能
- 双时钟域：
 - 互为异步关系的APB时钟和USART时钟
 - 不依赖PCLK设置的波特率设置
- 可编程的波特率产生器，当时钟频率为108MHz，过采样为8，最高速度可达13.5 MBits/s
- 完全可编程的串口特性：
 - 数据位(8或9位)低位或高位在前
 - 偶校验位，奇校验位，无校验位的生成或检测
 - 产生0.5, 1, 1.5或者2个停止位
- 可互换的Tx/Rx引脚
- 可配置的数据极性
- 自动检测波特率
- 支持硬件Modem流控操作(CTS/RTS)和RS485驱动使能
- 借助集中式DMA，实现可配置的多级缓存通信
- 发送器和接收器可分别使能
- 奇偶校验位控制：
 - 发送奇偶校验位
 - 检测接收的数据字节的奇偶校验位
- LIN断开帧的产生和检测
- 支持红外数据协议(IrDA)
- 同步传输模式以及为同步传输输出发送时钟

- 支持兼容ISO7816-3的智能卡接口:
 - 字节模式($T=0$)
 - 块模式($T=1$)
 - 直接和反向转换
- 多处理器通信:
 - 如果地址不匹配，则进入静默模式
 - 通过线路空闲检测或者地址掩码检测从静默模式唤醒
- 支持ModBus通信:
 - 超时功能
 - CR/LF字符识别
- 从深度睡眠模式唤醒:
 - 通过标准的RBNE中断
 - 通过WUF中断
- 多种状态标志:
 - 传输检测标志: 接收缓冲区不为空(RBNE), 接收FIFO满(RFF), 发送缓冲区为空(TBE), 传输完成(TC)
 - 错误检测标志: 过载错误(ORERR), 噪声错误(NERR), 帧格式错误(FERR), 奇偶校验错误(PERR)
 - 硬件流控操作标志: CTS变化(CTSF)
 - LIN模式标志: LIN断开检测(LBDF)
 - 多处理器通信模式标志: IDLE帧检测(IDLEF)
 - ModBus通信标志: 地址/字符匹配(AMF), 接收超时(RTF)
 - 智能卡模式标志: 块结束(EBF)和接收超时(RTF)
 - 从深度睡眠模式唤醒标志
 - 若相应的中断使能，这些事件发生将会触发中断

USART0完全实现上述功能，但是USART1只实现了上面所介绍的部分功能，下面这些功能在USART1中没有实现：

- 自动波特率检测
- 智能卡模式
- IrDA SIR ENDEC模块
- LIN模式
- 双时钟域和从深度睡眠模式唤醒
- 接收超时中断
- ModBus通信

18.3. 功能描述

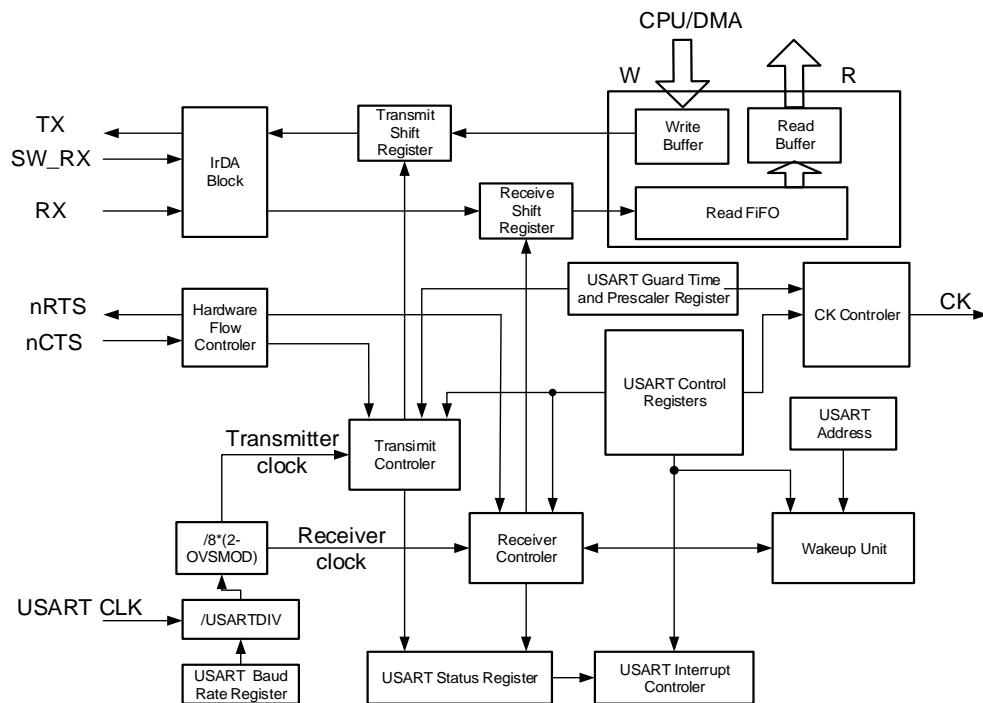
USART 接口通过[表 18-1. USART 重要引脚描述](#)中主要引脚从外部连接到其他设备。

表 18-1. USART 重要引脚描述

引脚	类型	描述
RX	输入	接收数据
TX	输出	发送数据。当 USART 使能后，若无数据发

引脚	类型	描述
	I/O (单线模式/智能卡模式)	送, 默认为高电平
CK	输出	用于同步通信的串行时钟信号
nCTS	输入	硬件流控模式发送使能信号
nRTS	输出	硬件流控模式发送请求信号

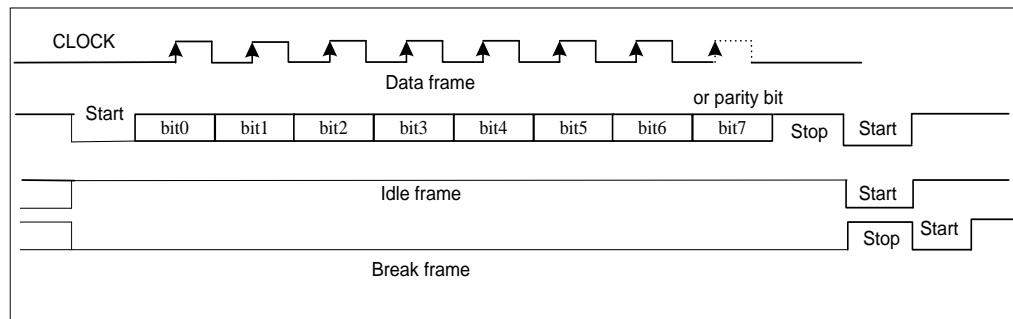
图 18-1. USART 模块内部框图



18.3.1. USART 帧格式

USART数据帧开始于起始位,结束于停止位。USART_CTL0寄存器中WL位可以设置数据长度。将USART_CTL0寄存器中PCEN置位,最后一个数据位可以用作校验位。若WL位为0,第七位为校验位。若WL位置1,第八位为校验位。USART_CTL0寄存器中PM位用于选择校验位的计算方法。

图18-2. USART字符帧 (8数据位和1停止位)



在发送和接收中, 停止位可以在USART_CTL1寄存器中STB[1:0]位域中配置。

表 18-2. 停止位配置

STB[1:0]	停止位长度(位)	功能描述
00	1	默认值
01	0.5	智能卡模式接收
10	2	标准 USART 和单线模式
11	1.5	智能卡模式发送和接收

在一个空闲帧中，所有位都为1。数据帧长度与正常USART数据帧长度相同。

紧随停止位后多个低电平为中断帧。USART数据帧的传输速度由PCLK时钟频率，波特率发生器的配置，以及过采样模式共同决定。

18.3.2. 波特率发生

波特率分频系数是一个16位的数字，包含12位整数部分和4位小数部分。波特率发生器使用这两部分组合所得的数值来确定波特率。由于具有小数部分的波特率分频系数，将使USART能够产生所有标准波特率。

波特率分频系数 (USARTDIV) 与系统时钟具有如下关系：

如果过采样率是16，公式为：

$$\text{USARTDIV} = \frac{\text{UCLK}}{16 \times \text{Baud Rate}} \quad (\text{式18-1})$$

如果过采样是8，公式为：

$$\text{USARTDIV} = \frac{\text{UCLK}}{8 \times \text{Baud Rate}} \quad (\text{式18-2})$$

USART时钟(UCLK)的选择是通过时钟控制系统来实现的(参考RCU部分)。在使能USART之前，必须选好时钟源(通过设置UEN位)。

18.3.3. USART 发送器

如果USART_CTL0寄存器的发送使能位(TEN)被置位，当发送数据缓冲区不为空时，发送器将会通过TX引脚发送数据帧。TX引脚的极性可以通过USART_CTL3寄存器中TINV位来配置。时钟脉冲通过CK引脚输出。

在传输被破坏的情况下，只要传输在继续，便不得禁用TEN位。

系统上电后，TBE默认为高电平。在USART_STAT寄存器中TBE置位时，数据可以在不覆盖前一个数据的情况下写入USART_TDATA寄存器。当数据写入USART_TDATA寄存器，TBE位将被清0。在数据由USART_TDATA移入移位寄存器后，该位由硬件置1。如果数据在一个发送过程正在进行时被写入USART_TDATA寄存器，它将首先被存入发送缓冲区，在当前发送过程完成时传输到发送移位寄存器中。如果数据在写入USART_TDATA寄存器时，没有发送过程正在进行，TBE位将被清零然后迅速置位，原因是数据被立刻传输到发送移位寄存器。

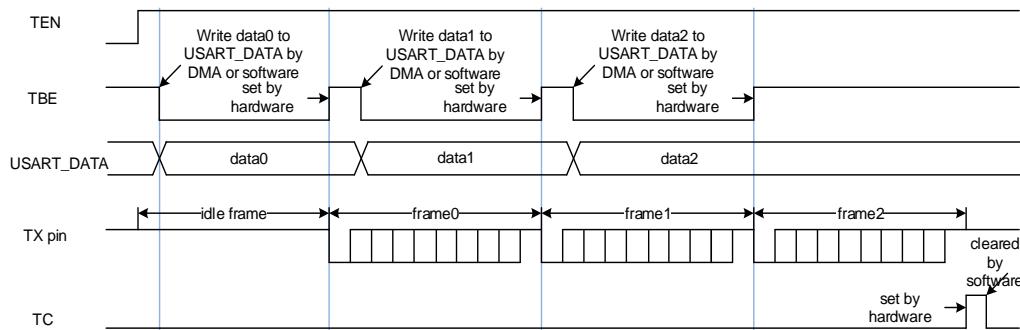
假如一帧数据已经被发送出去，并且TBE位已被置位，那么USART_STAT寄存器中TC位将被

置1。如果USART_CTL0寄存器中的中断使能位(TCIE)为1，将会产生中断。

USART 发送按下列步骤进行：

1. 通过USART_CTL0寄存器的WL设置字长；
2. 在USART_CTL1寄存器中写STB[1:0]位来设置停止位的长度；
3. 如果选择了多级缓存通信方式，应该在USART_CTL2寄存器中使能DMA (DENT位)；
4. 在USART_BAUD寄存器中设置波特率；
5. 在USART_CTL0寄存器中置位UEN位，使能USART；
6. 在USART_CTL0寄存器中设置TEN位；
7. 等待TBE置位；
8. 向USART_TDATA寄存器写数据；
9. 等待TC=1，发送完成。

图 18-3. USART 发送步骤



在禁用USART或进入低功耗状态之前，必须等待TC置位。

先读USART_STAT然后写USART_TDATA可将TC位清0。在多级缓存通信方式(DENT=1)下，直接向TC写0，也能清TC。

当SBKCMD置位时，会发送一个断开帧，发送完成后，SBKCMD将被清0。

18.3.4. USART 接收器

上电后，按以下步骤使能USART接收器：

1. 在USART_CTL0寄存器中置位UEN位，使能USART；
2. 写USART_CTL0寄存器的WL位去设置字长；
3. 在USART_CTL1寄存器中写STB[1:0]位来设置停止位的长度；
4. 如果选择了多级缓存通信方式，应该在USART_CTL2寄存器中使能DMA(DENR位)；
5. 在USART_BAUD寄存器中设置波特率；
6. 在USART_CTL0中设置REN位。

接收器在使能后，若检测到一个有效的起始脉冲，接收器开始接收码流。在接收一个数据帧的过程中会进行噪声错误，奇偶校验错误，帧错误和过载错误的检测。

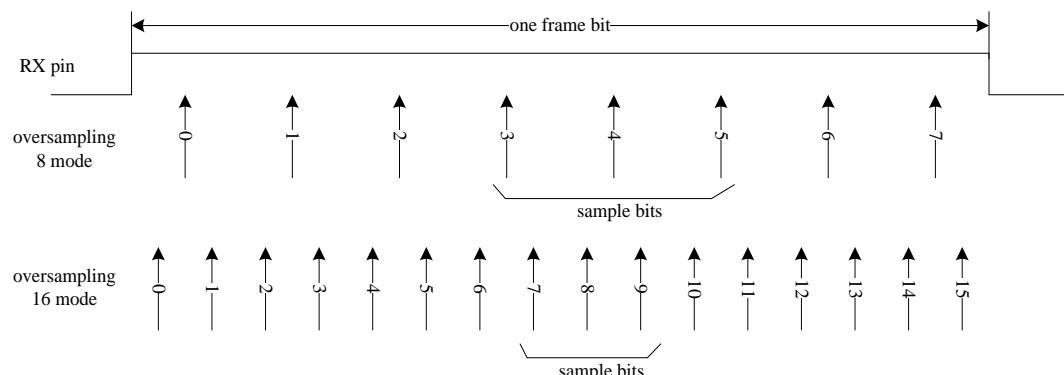
当一个数据帧接收完成，**USART_STAT**寄存器中的**RBNE**置位，如果设置了**USART_CTL0**寄存器中相应的中断使能位**RBNEIE**，将会产生中断。在**USART_STAT**寄存器中可以观察接收状态标志。

软件可以通过读**USART_RDATA**寄存器或者DMA方式获取接收到的数据。不管是直接读寄存器还是通过DMA，只要是对**USART_RDATA**寄存器的一个读操作都可以清除**RBNE**位。

在接收过程中，需使能**REN**位，不然当前的数据帧将会丢失。

在默认情况下，接收器通过获取三个采样点的值来估计该位的值。如果是8倍过采样模式，选择第3、4、5个采样点；如果是16倍过采样模式，选择第7、8、9个采样点。如果在3个采样点中有2个或3个为0，该数据位被视为0，否则为1。如果3个采样点中有一个采样点的值与其他两个不同，不管是起始位，数据位，奇偶校验位或者停止位，都将产生噪声错误(NERR)。如果使能DMA，并置位**USART_CTL2**寄存器中**ERRIE**，将会产生中断。如果在**USART_CTL2**中置位**OSB**，接收器将仅获取一个采样点来估计一个数据位的值。在这种情况下将不会检测到噪声错误。

图 18-4. 过采样方式接收一个数据位(OSB=0)



通过置位**USART_CTL0**寄存器中的**PCEN**位，奇偶校验功能被使能，接收器在接收一个数据帧时计算预期奇偶校验值，并将其与接收到的奇偶校验位进行比较。如果不相等，**USART_STAT**寄存器中**PERR**被置位。如果设置了**USART_CTL0**寄存器中的**PERRIE**位，将产生中断。

如果在停止位传输过程中RX引脚为0，将产生帧错误，**USART_STAT**寄存器中**FERR**置位。如果使能DMA并置位**USART_CTL2**寄存器中**ERRIE**位，将产生中断。

当接收到一帧数据，而**RBNE**位还没有被清零，随后的数据帧将不会存储在数据接收缓冲区中。**USART_STAT**寄存器中的溢出错误标志位**ORERR**将置位。如果使能DMA并置位**USART_CTL2**寄存器中**ERRIE**位或者置位**RBNEIE**，将产生中断。

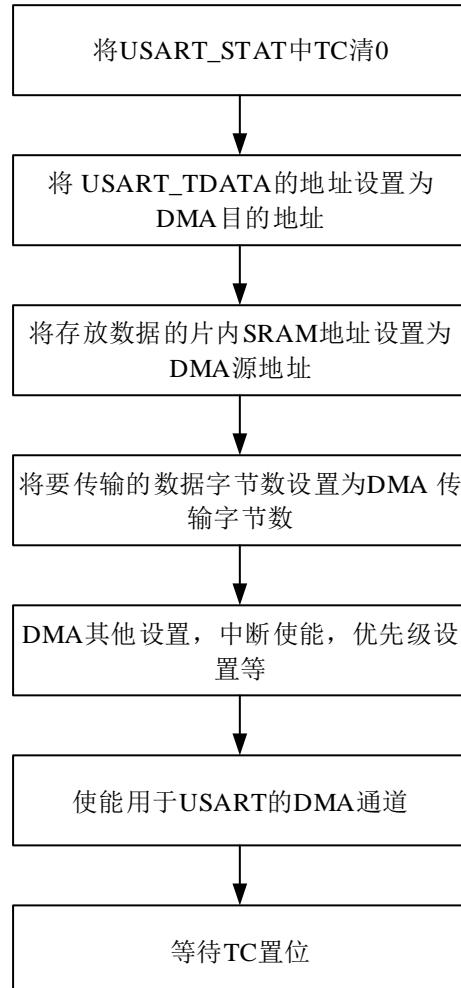
在一个接收过程中，**RBNE**、**NERR**、**PERR**、**FERR**和**ORERR**总是同时置位。如果没有使能DMA，软件需检查**RBNE**中断是否由**NERR**、**PERR**、**FERR**或者**ORERR**置位产生。

18.3.5. DMA 方式访问数据缓冲区

为减轻处理器的负担，可以采用DMA访问发送缓冲区或者接收缓冲区。置位USART_CTL2寄存器中DENT位可以使能DMA发送，置位USART_CTL2寄存器中DENR位可以使能DMA接收。

当 DMA 用于 USART 发送时，DMA 将数据从片内 SRAM 传送到 USART 的数据缓冲区。配置步骤如[图 18-5. 采用 DMA 方式实现 USART 数据发送配置步骤](#)所示。

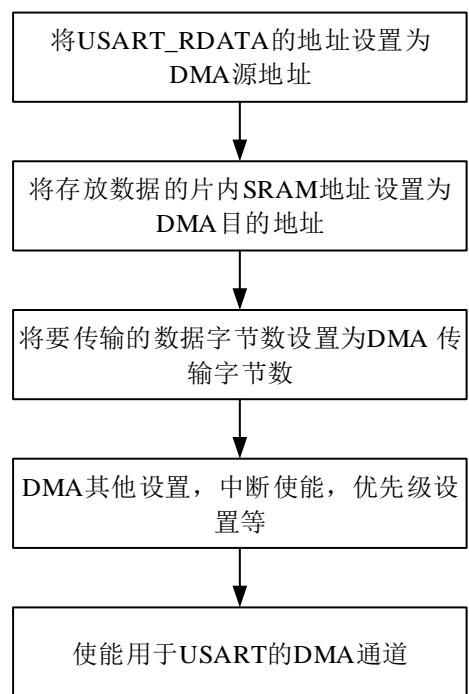
图 18-5. 采用 DMA 方式实现 USART 数据发送配置步骤



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

当 DMA 用于 USART 接收时，DMA 将数据从接收缓冲区传送到片内 SRAM。配置步骤如[图 18-6. 采用 DMA 方式实现 USART 数据接收配置步骤](#)所示。如果将 USART_CTL2 寄存器中ERRIE 位置 1，USART_STAT 寄存器中的错误标志位(FERR、ORERR 和 NERR)被置位时将产生中断。

图 18-6. 采用 DMA 方式实现 USART 数据接收配置步骤

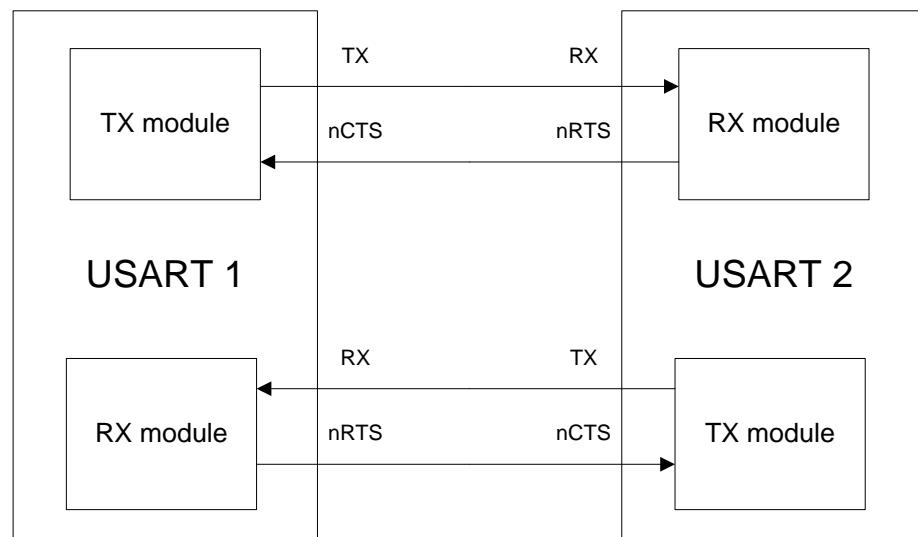


当USART接收到的数据数量达到了DMA传输数据数量，DMA模块将产生传输完成中断。

18.3.6. 硬件流控制

硬件流控制功能通过nCTS和nRTS引脚来实现。通过将USART_CTL2寄存器中RTSEN位置1来使能RTS流控，将USART_CTL2寄存器中CTSEN位置1来使能CTS流控。

图 18-7. 两个 USART 之间的硬件流控制



RTS 流控

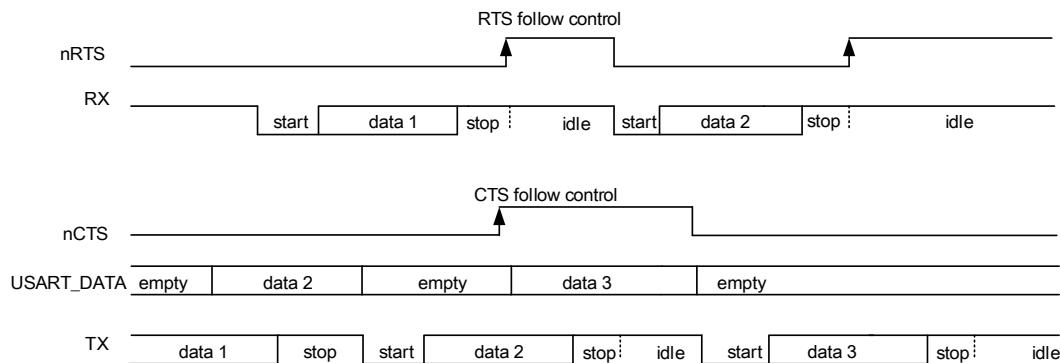
仅当nRTS信号低电平时，USART接收器才能接收数据。在接收期间，信号保持低电平。接收完成之后，电平变高。当nRTS信号再次变低时，开始下一次接收。如果接收寄存器已满，信号

将保持高电平。

CTS 流控

如果USART_STAT寄存器的TBE位是'0'，且nCTS信号是低电平，发送器发送数据帧。在发送期间，若nCTS信号变为高电平，发送器在发送完当前数据帧后停止发送。

图 18-8. 硬件流控制



如果CTS流控制被使能，在nCTS引脚信号发生变化时，USART_STAT寄存器中CTSF位会置1。如果USART_CTL2寄存器中的CTSIE位被置位，将会产生中断。

RS485 驱动使能

驱动使能功能通过设置USART_CTL2控制寄存器的DEM位来打开。它允许用户通过DE(Driver Enable)信号激活外部收发器控制。提前时间是驱动使能信号和第一个字节的起始位之间的时间间隔。这个时间可以在USART_CTL0控制器的DEA[4:0]位域中进行设置。滞后时间是一个发送信息最后一个字节的停止位与释放DE信号之间的时间间隔。这个时间可以在USART_CTL0控制寄存器的DED[4:0]位域中进行设置。DE信号的极性可以通过USART_CTL2控制寄存器的DEP位进行设置。

18.3.7. 多处理器通信

在多处理器通信中，多个USART被连接成一个网络。对于一个设备来说，监视所有来自RX引脚的消息，是一种巨大的负担。为减轻设备负担，软件可以通过将USART_CMD寄存器中MMCMD位置1使USART进入静默模式。

如果USART处于静默模式，所有的接收状态标志位将不会被置位。此外，USART可以由硬件用以下两种方式中的一种来唤醒：空闲总线检测和地址掩码检测。

设备默认使用空闲总线检测方法唤醒USART。当在RX引脚检测到空闲帧时，硬件会将RWU清零，从而退出静默模式，但USART_STAT寄存器中IDLEF位不会被置1。

当USART_CTL0寄存器中WM被置位，数据最高位会被认为是地址标志位。如果地址标志位为1，该字节被认为是地址字节。如果地址标志位是0，该字节被认为是数据字节。如果地址字节的低4位或低7位与USART_CTL1寄存器中的ADDR位相同，硬件会将RWU清零，并退出静默模式。接收到将USART唤醒的数据帧，RBNE将置位。状态标志可以从USART_STAT寄存器中

获取。如果地址字节的低4位或低7位与USART_CTL1寄存器中的ADDR位不相同，硬件会置位RWU并自动进入静默模式。在这种情况下，RBNE不会被置位。

如果USART_CTL0寄存器中PCEN位被置位，地址字节最高位被视为校验位，其余位被视为地址。如果ADDM位被置位，且接收帧为7位的数据，其中最低的6位将与ADDR[5:0]比较。如果ADDM位被置位，且接收帧为9位的数据，其中最低8位将与ADDR[7:0]进行比较。

18.3.8. LIN 模式

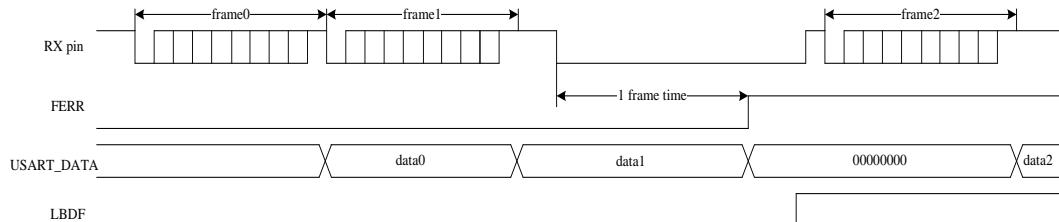
将USART_CTL1寄存器的LMEN置位即可使能本地互联网络模式。在LIN模式下，USART_CTL1寄存器中CKEN，STB[1:0]和USART_CTL2的SCEN，HDEN，IREN位须被清0。

LIN发送过程与普通发送过程基本相同。数据位的长度只能为8。一个停止位后连续13个0为断开帧。

断开检测功能完全独立于普通USART接收器。因此，在空闲状态下或帧传输状态下都可以进行断开帧检测。USART_CTL1寄存器中LBLEN位可以选择断开帧的长度。如果在RX引脚检测到大于或等于与预期的断开帧长度的0(LBLEN=0时，10个0；LBLEN=1时，11个0)，USART_STAT寄存器中LBDF置位。如果USART_CTL1寄存器中LBDIE被置位，将产生中断。

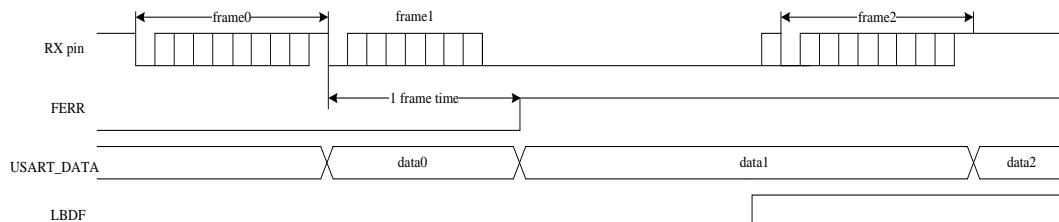
如[图 18-9. 空闲状态下检测断开帧](#)所示，如果断开帧发生在空闲状态下，USART接收器会接收到一个全0数据帧，同时FERR置位。

图 18-9. 空闲状态下检测断开帧



如[图 18-10. 数据传输过程中检测断开帧](#)所示，如果断开帧发生在数据传输过程中，当前传输帧发生错误，FERR置位。

图 18-10. 数据传输过程中检测断开帧



18.3.9. 同步通信模式

USART支持主机模式下的全双工同步串行通信，可以通过置位USART_CTL1的CKEN位来使能。在同步模式下，USART_CTL1的LMEN和USART_CTL2的SCEN，HDEN，IREN位应被清0。CK引脚作为USART同步发送器的时钟输出，仅仅当TEN位被使能时，它才被激活。在起始

位和停止位传送期间，不会从CK引脚输出时钟脉冲。USART_CTL1的CLEN位用来决定在最低位（地址索引位）发送期间是否有时钟信号输出。在空闲状态和断开帧的发送过程中，也不会有时钟信号产生。USART_CTL1的CPH位用来决定数据在第一个时钟沿被采样还是在第二个时钟沿被采样。USART_CTL1的CPL位用来决定在USART同步模式空闲状态下，时钟引脚的电平。

在发送器或接收器使能的情况下，不能改变这三位(CPL, CPH, CLEN)的值。

时钟与已发送的数据同步。同步模式下的接收器按照发送器的时钟进行采样，没有任何过采样。

图 18-11. 同步模式下的 USART 示例

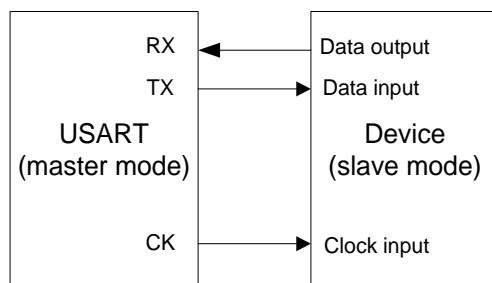
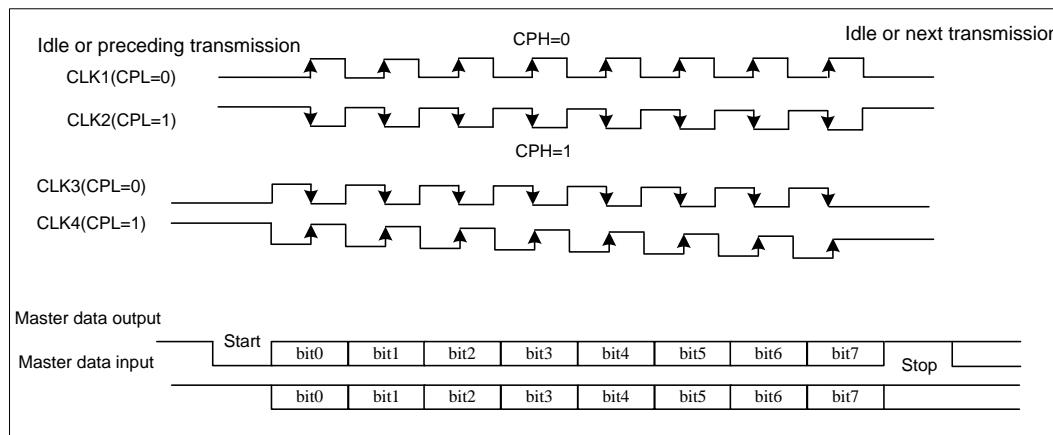


图 18-12. 8-bit 格式的 USART 同步通信波形(CLEN=1)

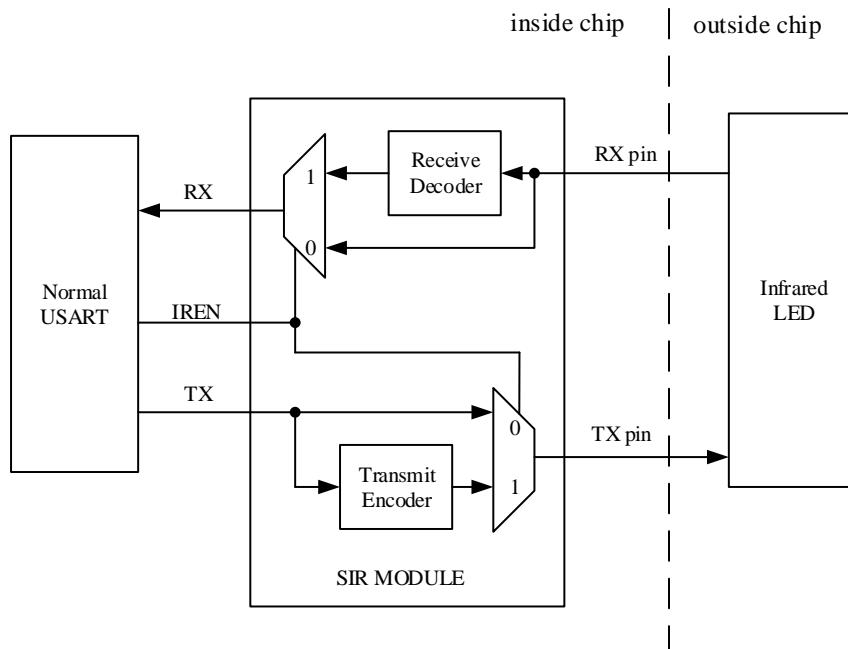


18.3.10. 串行红外(IrDA SIR)编解码功能模块

通过置位USART_CTL2寄存器中IREN，使能IrDA模式。在IrDA模式下，USART_CTL1寄存器的LMEN，STB[1:0]，CKEN位和USART_CTL2寄存器的HDEN，SCEN位将被清0。

在IrDA模式下，USART数据帧由SIR发送编码器进行调制，再被发送至红外LED，调制后的信号经由红外LED进行发送，经解调后将数据发送至USART接收器。对于编码器而言，波特率应小于115200。

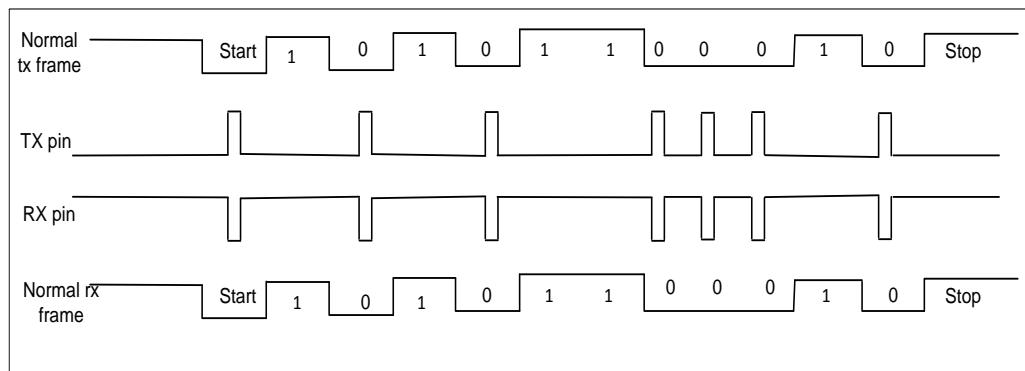
图 18-13. IrDA SIR ENDEC 模块



在IrDA模式下，TX引脚与RX引脚电平不同。TX引脚通常为低电平，RX引脚通常为高电平。IrDA引脚电平保持稳定代表逻辑‘1’，红外光源脉冲(RTZ信号)代表逻辑‘0’。其脉冲宽度通常占一个位时间的3/16。IrDA无法检测到宽度小于一个PSC时钟的脉冲。如果脉冲宽度大于1但是小于2倍PSC时钟，IrDA则无法可靠地检测到。

由于IrDA是一种半双工协议，因此在IrDA SIR ENDEC模块中，发送和接收不得同时进行。

图 18-14. IrDA 数据调制



将USART_CTL2寄存器中IRLP置位可以使SIR子模块工作在低功耗模式下。发送编码器由PCLK分频得到的低速时钟来驱动。分频系数在USART_GP寄存器中PSC[7:0]位配置。TX引脚脉冲宽度是低速时钟的3个周期。接收解码器工作模式与正常IrDA模式相同。

18.3.11. 半双工通信模式

通过设置USART_CTL2寄存器的HDEN位，可以使能半双工模式。在半双工通信模式下，USART_CTL1寄存器的LMEN，CKEN位和USART_CTL2寄存器的SCEN，IREN位清零。

半双工模式下仅用单线通信。TX引脚和RX引脚从内部连接到一起，TX引脚应被配置为IO管脚。通信冲突应由软件处理。当TEN被置位时，在数据寄存器中的数据将会被发送。

18.3.12. 智能卡(ISO7816)模式

智能卡模式是一种异步通信模式，将USART_CTL2寄存器的SCEN位置位可使能智能卡模式。在智能卡模式下，USART_CTL1的LMEN位和USART_CTL2的HDEN, IREN位应该清零。

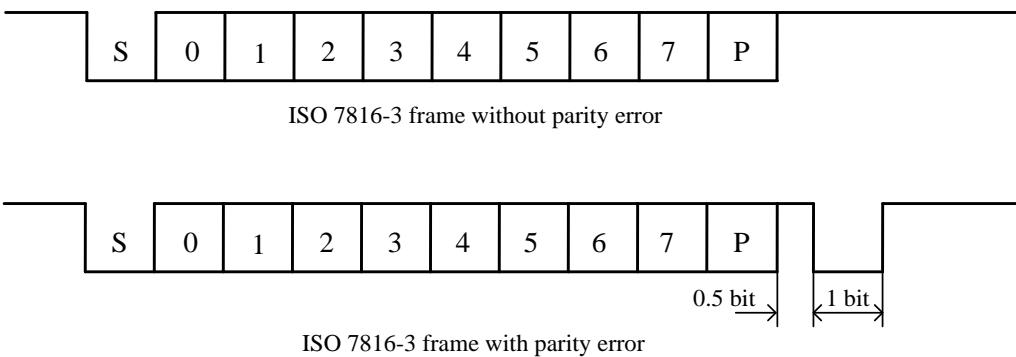
如果CKEN位被置位，USART将向智能卡提供一个时钟。该时钟可以分频用于其他用途。

智能卡模式下的帧格式为：1起始位+9数据位(包括1个奇偶校验位)+1.5停止位。

智能卡模式是一种半双工通信协议模式。当与智能卡连接时，TX引脚须被设置成开漏模式，这个引脚将会与智能卡驱动同一条双向连线。

T=0 模式

图 18-15. ISO7816-3 数据帧格式



相较于正常操作模式下的时序，从发送移位寄存器到TX引脚的传递时间延迟了半个波特率时钟，并且TC标志的置位将根据USART_GP寄存器的GUAT[7:0]设置延迟某一特定时间。根据协议，USART能自动重发数据，重发次数在SCRTNUM中设置。在上一个重复的字节接收结束，TC位将立即被置位而没有一个保护时间。如果USART连续收到NACK信号，那么在设定好的重发次数后，它将停止发送并将该错误标记为帧格式错误。USART_CMD寄存器的TXFCMD位可将TBE位清0。

在USART接收期间，在一帧数据的接收过程中，如果检测到有奇偶校验错误，TX引脚将拉低一个波特率时钟。这个信号是发送‘NACK’信号到智能卡。然后智能卡一侧会产生一个帧错误。如果接收到的字节是错误的，RBNE中断和接收DMA请求都不会被激活。根据协议，智能卡将重新发送数据。如果在最大的重新发送次数后(这个次数的具体值在SCRTNUM位域)，接收到的字符仍然是错误的，USART停止发送NACK信号并标注这个错误为奇偶校验错误。

若USART_CTL2中NKEN被置位，‘NACK’信号将被发送到USART。USART不会将‘NACK’误当作起始位。

空闲帧和断开帧在智能卡模式下不适用。

T=1 模式 (块模式)

在T=1(块模式)下，USART_CTL2寄存器的NKEN位应该清零来关闭校验错误发送。

当要从智能卡读取数据时，软件必须将USART_RT寄存器设置成BWT (块等待)-11的值并将RBNEIE置位。如果到了这个时间，还没有从智能卡收到应答，将引起超时中断。如果在超时之前收到了第一个字节，则会引起RBNE中断。块模式下，如果用DMA从智能卡读取数据，也只能在第一个字节接收完后去使能DMA。

在接收到第一个字节之后(RBNE中断)必须将USART_RT寄存器设置为CWT (字节等待时间)-11的值(这个时间以波特时间作为单位)，这是为了在两个连续的字符之间自动检测最大等待时间。如果智能卡在前一个字符发送结束后到设定的CWT周期之间没有发送字符，USART会通过RTF标志提醒软件，当RTIE被置位时，会引起中断。

USART用一个块长度计数器去统计收到的所有字符的长度。这个计数器在USART开始发送的时候自动清0(TBE=0)。这个块长度信息位于智能卡发出数据的第三个字节(序言部分)。这个值必须写入USART_RT寄存器的BL。当使用DMA模式时，在块开始之前，这个寄存器必须被设定为最小值(0x0)。有了这个值，在收到第四个字节后，会引起一个中断。软件必须从接收缓冲区读取第三个字节作为块长度。

在中断驱动接收模式，块的长度可以由软件提取出来并做检测或者通过设置BL的值得到。但是在块开始之前，BL(0xFF)可以被设置为最大值。实际值则要在接收到第三个字节后写到寄存器中。

整个块的长度(包括序言区，收尾区和信息区)等于BL+4。块尾通过EBF标志和相应中断提醒给软件(当EBIE位置1时)。如果块长度出错，将会引起一个RT中断。

直接和反向转换

智能卡协议定义了两种转换方式：直接转换和反向转换。

如果选择直接转换，从数据帧的最低位开始传输，TX引脚高电平代表逻辑‘1’，偶校验。在这种情况下，MSBF位和DINV位都设置为0(默认值)。

如果选择反向转换，从数据帧的最高位开始传输，TX引脚低电平代表逻辑‘1’，偶校验。在这种情况下，MSBF位和DINV位都设置为1。

18.3.13. 自动波特率检测

USART能够基于接收到的一个字符自动检测和设置USART_BAUD寄存器的值。通过设置USART_CTL1寄存器的ABDM位，有两种自动波特率检测方法可以选择。分别是：

1. 以1开始的任一字符。在这种情况下，USART将开始测量起始位的长度(下降沿到上升沿)。
2. 以10xx开头的任何字符，这种情况下USART将测量起始位和第一个数据位的总长度，通过下降沿到下降沿，以减少信号斜率对测量精度的影响。

18.3.14. ModBus 通信

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

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

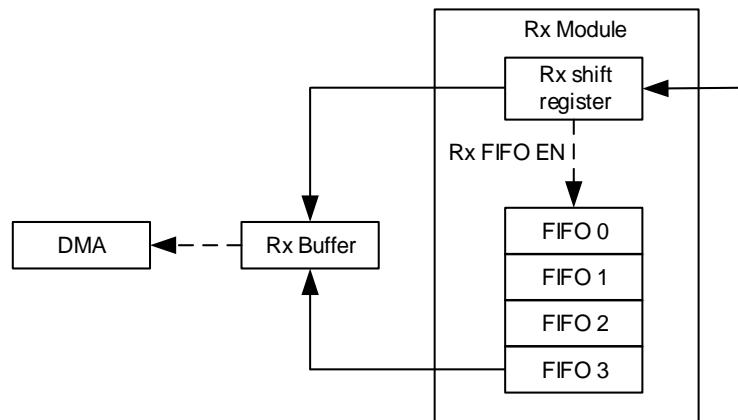
为了检测空闲状态，USART_CTL1寄存器的RTEN位和USART_CTL0寄存器的RTIE位必须置位。USART_RT寄存器必须被设置成与2个字节超时所对应的值。在最后一个停止位被接收后，当接收线在这期间是空闲的，将产生一个中断，通知软件当前块接收已经完成。

在ModBus/ASCII模式下，块尾被认为是一个特定的字符(CR/LF)串。USART用字符匹配机制实现这个功能。具体是通过将LF的ASCII码配置到ADDR区域并激活地址匹配中断(AMIE=1)来实现。软件将在收到LF或可以在DMA缓存中查找到CR/LF时得到提示。

18.3.15. 接收 FIFO

将USART_RFCS寄存器的RFEN置位，接收FIFO被使能，并当CPU无法迅速响应RBNE中断时，能避免发生过载错误。接收FIFO和接收缓存区可储存多至5帧的数据。若接收FIFO满，RFFINT位将被置位。如果RFFIE被置位，此时产生中断。

图 18-16. USART 接收 FIFO 结构



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

18.3.16. 从 DeepSleep 模式唤醒

通过标准RBNE中断或WUM中断USART能从深度睡眠模式唤醒MCU。

UESM位必须置1并且USART时钟必须设置为IRC8M或LXTAL (请参考RCU部分)。

当使用RBNE标准中断时，必须在进入深度睡眠模式前将RBNEIE位置位。

当使用WUIE中断时，WUIE中断源可以通过WUM位来选择。

在进入深度睡眠模式前，必须禁用DMA。在进入深度睡眠模式前，软件必须检测USART没有正在传送数据。这可以通过USART_STAT寄存器中的BSY标志来判断。REA位必须被检测以确保USART是使能的。

当检测到唤醒事件时，无论MCU工作在深度睡眠模式还是正常模式，WUF标志位通过硬件被置1，并且在WUIE被置位的情况下，触发一个唤醒中断。

18.3.17. USART 中断

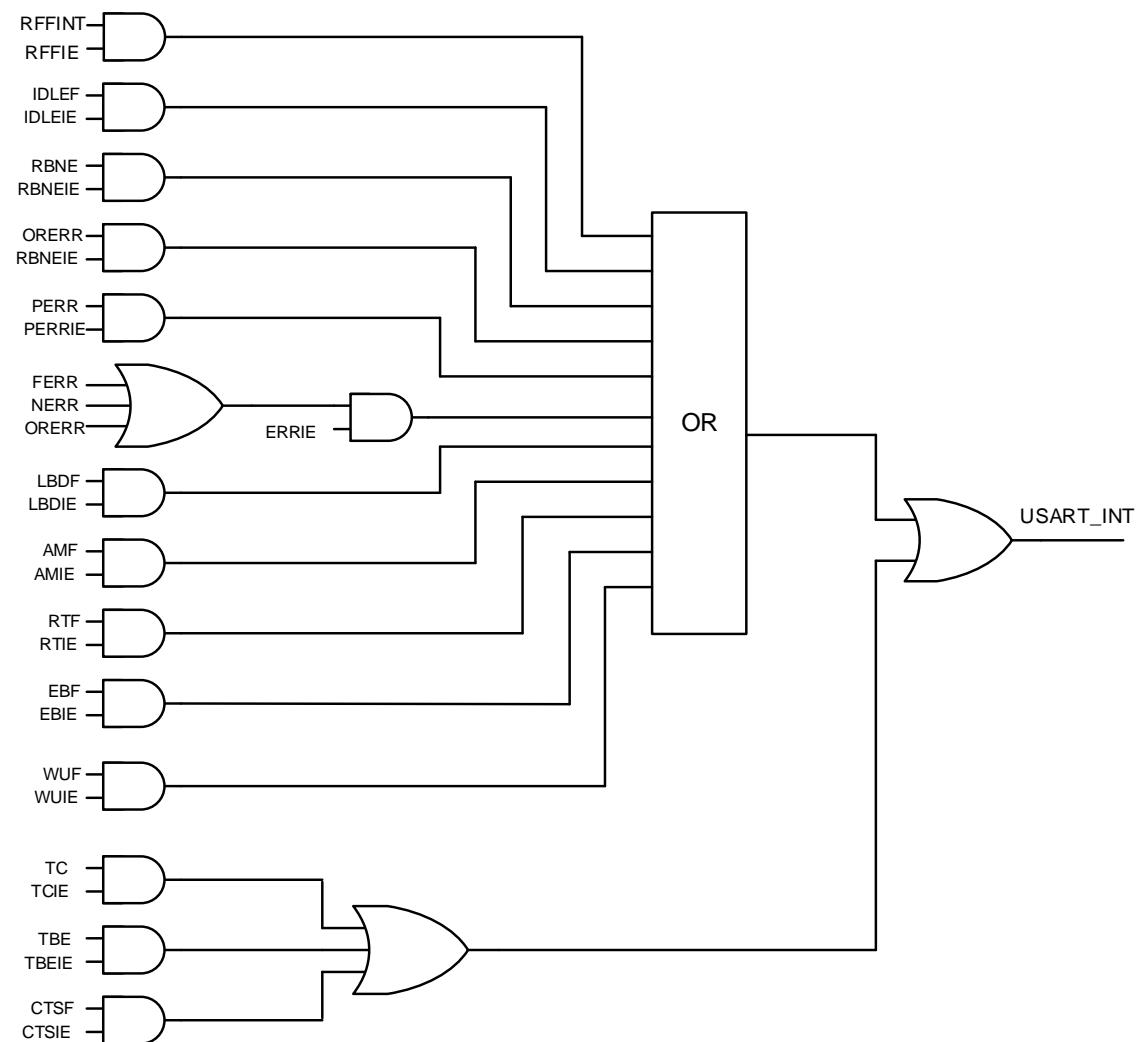
USART 中断事件和标志如[表 18-3. USART 中断请求](#)所示：

表 18-3. USART 中断请求

中断事件	事件标志	使能控制位
发送数据寄存器空	TBE	TBEIE
CTS标志	CTSF	CTSIE
发送结束	TC	TCIE
接收到的数据可以读取	RBNE	RBNEIE
检测到过载错误	ORERR	
接收FIFO满	RFFINT	RFFIE
检测到线路空闲	IDLEF	IDLEIE
奇偶校验错误	PERR	PERRIE
LIN模式下，检测到断开标志	LBDF	LBDIE
接收错误(噪声错误、溢出错误、帧错误)当DMA接收使能时	NERR or ORERR or FERR	ERRIE
字符匹配	AMF	AMIE
接收超时错误	RTF	RTIE
发现块尾	EBF	EBIE
从Deepsleep模式唤醒	WUF	WUIE

在发送给中断控制器之前，所有的中断事件是逻辑或的关系。因此在任何时候 USART 只能向控制器产生一个中断请求。不过软件可以在一个中断服务程序里处理多个中断事件。

图 18-17. USART 中断映射框图



18.4. 寄存器定义

18.4.1. USART 控制寄存器 0 (USART_CTL0)

地址偏移: 0x00

复位值: 0x0000_0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				EBIE	RTIE	DEA[4:0]				DED[4:0]					
						rw				rw				rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVSMOD	AMIE	MEN	WL	WM	PCEN	PM	PERRIE	TBEIE	TCIE	RBNEIE	IDLEIE	TEN	REN	UESM	UEN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:28	保留	必须保持复位值
27	EBIE	块尾中断使能 0: 中断禁止 1: 中断使能 在USART1, 该位保留。
26	RTIE	接收超时中断使能 0: 中断禁止 1: 中断使能 在USART1, 该位保留。
25:21	DEA[4:0]	驱动使能置位时间 这些数字用来定义DE(驱动使能)信号的置位与第一个字节的起始位之间的时间间隔。它以采样时间为单位(1/8或1/16位时间),可以通过OVSMOD位来配置。 当USART被使能(UEN=1)时,该位域不能被改写。
20:16	DED[4:0]	驱动使能置低时间 这些位用来定义一个发送信息最后一个字节的停止位与置低DE(驱动使能)信号之间的时间间隔。它以采样时间为单位(1/8或1/16位时间),可以通过OVSMOD位来配置。 当USART被使能(UEN=1)时,该位域不能被改写。
15	OVSMOD	过采样模式 0: 16倍过采样 1: 8倍过采样 在LIN, IrDA 和智能卡模式, 该位保持清0。 当USART被使能(UEN=1)时,该位域不能被改写。
14	AMIE	ADDR字符匹配中断使能 0: ADDR字符匹配中断禁用 1: ADDR字符匹配中断使能
13	MEN	静默模式使能 0: 静默模式禁用 1: 静默模式被使能
12	WL	字长 0: 8 数据位 1: 9 数据位 当USART被使能(UEN=1)时,该位域不能被改写。
11	WM	从静默模式唤醒方法 0: 空闲线 1: 地址标记 当USART被使能(UEN=1)时,该位域不能被改写。
10	PCEN	校验控制使能

		0: 校验控制禁用 1: 校验控制被使能 当USART被使能(UEN=1)时，该位域不能被改写。
9	PM	校验模式 0: 偶校验 1: 奇校验 当USART被使能(UEN=1)时，该位域不能被改写。
8	PERRIE	校验错误中断使能 0: 校验错误中断禁用 1: 当USART_STAT寄存器的PERR位置位时，将触发中断。
7	TBEIE	发送寄存器空中断使能 0: 中断禁止 1: 当USART_STAT寄存器的TBE位置位时，将触发中断。
6	TCIE	发送完成中断使能 0: 发送完成中断禁用 1: 当USART_STAT寄存器的TC位置位时，将触发中断。
5	RBNEIE	读数据缓冲区非空中断和过载错误中断使能 0: 读数据缓冲区非空中断和过载错误中断禁用 1: 当USART_STAT寄存器的ORERR或RBNE位置位时，将触发中断。
4	IDLEIE	IDLE线检测中断使能 0: IDLE线检测中断禁用 1: 当USART_STAT寄存器的IDLEF位置位时，将触发中断。
3	TEN	发送器使能 0: 发送器关闭 1: 发送器打开
2	REN	接收器使能 0: 接收器关闭 1: 接收器打开并且开始搜索起始位。
1	UESM	USART在深度睡眠模式下使能 0: USART不能从深度睡眠模式唤醒MCU 1: USART能从深度睡眠模式唤醒MCU。条件是USART的时钟源必须是IRC8M或LXTAL。 在USART1，该位保留。
0	UEN	USART使能 0: USART预分频器和输出禁用 1: USART预分频器和输出被使能

18.4.2. USART 控制寄存器 1 (USART_CTL1)

地址偏移: 0x04

复位值: 0x0000_0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR[7:0]								RTEN	ABDM[1:0]	ABDEN	MSBF	DINV	TINV	RINV	
rw								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STRP	LMEN	STB[1:0]		CKEN	CPL	CPH	CLEN	保留	LBDIE	LBLEN	ADDM	保留			
rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw				

位/位域	名称	描述
31:24	ADDR[7:0]	<p>USART的节点地址</p> <p>这些位给出USART的节点地址。</p> <p>在多处理器通信并且静默模式或者深度睡眠模式期间，这些位用来唤醒进行地址标记的检测。接收到的最高位为1的数据帧将和这些位进行比较。当ADDM位被清零时，仅仅ADDR[3:0]被用来比较。</p> <p>在正常的接收期间，这些位也用来进行字符检测。所有接收到的字符(8位)与ADDR[7:0]的值进行比较，如果匹配，AMF标志将被置位。</p> <p>当接收器(REN=1)和USART(UEN=1)被使能时，该位域不能被改写。</p>
23	RTEN	<p>接收器超时使能</p> <p>0: 接收器超时功能禁用</p> <p>1: 接收器超时功能被使能</p> <p>在USART1，该位保留。</p>
22:21	ABDM[1:0]	<p>自动波特率检测模式</p> <p>00: 下降沿到上升沿的测量 (测量起始位)</p> <p>01: 下降沿对下降沿的测量(接收到的帧必须是一个这种格式的帧10xxxxxx)</p> <p>10: 保留</p> <p>11: 保留</p> <p>当USART被使能(UEN=1)时，该位域不能改写。</p> <p>在USART1，该位保留。</p>
20	ABDEN	<p>自动波特率检测使能</p> <p>0: 自动波特率检测禁用</p> <p>1: 自动波特率检测被使能</p> <p>在USART1，该位保留。</p>
19	MSBF	<p>高位在前</p> <p>0: 数据发送/接收，采用低位在前</p> <p>1: 数据发送/接收，采用高位在前</p>

		USART被使能(UEN=1)时，该位域不能被改写。
18	DINV	数据位反转 0: 数据位信号值没有反转 1: 数据位信号值被反转 USART被使能(UEN=1)时，该位域不能被改写。
17	TINV	TX管脚电平反转 0: TX管脚信号值没有反转 1: TX管脚信号值被反转。 USART被使能(UEN=1)时，该位域不能被改写。
16	RINV	RX管脚电平反转 0: RX管脚信号值没有反转. 1: RX管脚信号值被反转 USART被使能(UEN=1)时，该位域不能被改写。
15	STRP	交换TX/RX管脚 0: TX和RX管脚功能不被交换 1: TX和RX管脚功能被交换 当USART被使能(UEN=1)时，该位域不能改写。
14	LMEN	LIN模式使能 0: LIN模式关闭 1: LIN模式开启 USART被使能(UEN=1)时，该位域不能被改写。 在USART1，该位保留。
13:12	STB[1:0]	STOP位长 00: 1停止位 01: 0.5停止位 10: 2停止位 11: 1.5停止位 USART被使能(UEN=1)时，该位域不能被改写。
11	CKEN	CK管脚使能 0: CK管脚禁用 1: CK管脚被使能 USART被使能(UEN=1)时，该位域不能被改写。 在USART1，该位保留。
10	CPL	时钟极性 0: 在同步模式下，CK管脚不对外发送时保持为低电平 1: 在同步模式下，CK管脚不对外发送时保持为高电平 USART被使能(UEN=1)时，该位域不能被改写。
9	CPH	时钟相位 0: 在同步模式下，在首个时钟边沿采样第一个数据

			1: 在同步模式下, 在第二个时钟边沿采样第一个数据 USART被使能(UEN=1)时, 该位域不能被改写。
8	CLEN	CK长度	0: 在同步模式下, 最后一位(MSB)的时钟脉冲不输出到CK管脚 1: 在同步模式下, 最后一位(MSB)的时钟脉冲输出到CK管脚 USART被使能(UEN=1)时, 该位域不能被改写。
7	保留	必须保持复位值	
6	LBDIE	LIN断开信号检测中断使能	0: 断开信号检测中断禁用 1: 当USART_STAT的LBDF位置位, 将产生中断。 在USART1, 该位保留。
5	LBDL	LIN断开帧长度	0: 检测10位断开帧 1: 检测11位断开帧 USART被使能(UEN=1)时, 该位域不能被改写。 在USART1, 该位保留
4	ADDM	地址检测模式	这位用来选择4位地址检测还是全位地址检测。 0: 4位地址检测 1: 全位地址检测。在7位, 8位和9位数据模式下, 地址检测分别按6位, 7位和8位地址(ADDR[5:0], ADDR[6:0] 和ADDR[7:0])执行。 USART被使能(UEN=1)时, 该位域不能被改写。
3:0	保留	必须保持复位值	

18.4.3. USART 控制寄存器 2 (USART_CTL2)

地址偏移: 0x08

复位值: 0x0000_0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	保留
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
DEP	DEM	DDRE	OVRD	OSB	CTSIE	CTSEN	RTSEN	DENT	DENR	SCEN	NKEN	HDEN	IRLP	IREN	ERRIE	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

位/位域	名称	描述
31:23	保留	必须保持复位值
22	WUIE	从深度睡眠模式唤醒中断使能

		0: 从深度睡眠模式唤醒中断禁用 1: 从深度睡眠模式唤醒中断被使能 在USART1, 该位保留。
21:20	WUM[1:0]	从深度睡眠模式唤醒模式 这个位域指定什么事件可以置位USART_STAT寄存器中的WUF(从深度睡眠唤醒标志) 标志。 00: WUF在地址匹配的时候置位。如何实现地址匹配在ADDR和ADDM中定义。 01:保留 10: WUF在检测到起始位时置位 11: WUF在检测到RBNE时置位 USART被使能(UEN=1)时, 该位域不能被改写。 在USART1, 该位保留。
19:17	SCRTNUM[2:0]	智能卡自动重试数目 在智能卡模式下, 这些位用来指定在发送和接收时重试的次数。在发送模式下, 它指的是在产生发送错误(FERR位置位) 之前自动重试的发送次数。 在接收模式下, 它指的是在产生接收错误(RBNE位和PERR位置位)之前自动重试的接收次数。 当这些位被设置为0x0时, 在发送模式下这些位将不会自动发送。 USART被使能(UEN=1)时, 该位域被清零, 并停止重发。 在USART1, 该位保留。
16	保留	必须保持复位值
15	DEP	驱动使能的极性选择模式 0: DE信号高有效 1: DE信号低有效 USART被使能(UEN=1)时, 该位域不能被改写。
14	DEM	驱动使能模式 用户使能该位以后, 可以通过DE信号对外部收发器进行控制。DE信号是从RTS 管脚输出的。 0: DE功能禁用 1: DE功能开启 USART被使能(UEN=1)时, 该位域不能被改写。
13	DDRE	在接收错误时禁止DMA 0: 在发生接收错误的情况下, 不禁用DMA。所有的错误数据不会产生DMA请求, 以确保错误的数据不会被传输, 但是下一个接收到的正确的数据会被传输。RBNE位保持0以阻止过载错误, 但是相应错误标志位会被置位。这种模式可用于智能卡模式。 1: 在接收错误的情况下, DMA被关闭。 DMA请求会被屏蔽, 直到相应的标志位被清0。 RBNE标志和相应的错误标志位会被置位。软件在清除错误标志前, 必须首先关DMA请求(DMAR = 0) 或清RBNE。 USART被使能(UEN=1)时, 该位域不能被改写。
12	OVRD	溢出禁止

0: 溢出功能被使能。当接收到的数据在新数据到达前没有被读走, ORERR错误标志位将被置位, 并且新数据将会丢失。

1: 溢出功能禁止。当接收到的数据在新数据到达前没有被读走, ORERR错误标志位将不会被置位, 新数据会将USART_RDATA寄存器以前的内容覆盖。

USART被使能(UEN=1)时, 该位域不能被改写。

11	OSB	单次采样方式 0: 三次采样方法 1: 一次采样方法 USART被使能(UEN=1)时, 该位域不能被改写。
10	CTSIE	CTS中断使能 0: CTS中断屏蔽 1: 当USART_STAT的CTS位置位时, 会产生中断。
9	CTSEN	CTS使能 0: CTS 硬件流控禁用 1: CTS 硬件流控被使能 USART被使能(UEN=1)时, 该位域不能被改写。
8	RTSEN	RTS使能 0: RTS硬件流控禁用 1: RTS硬件流控被使能, 只有当接收缓冲区有空间的时候, 才会请求下一个数据。 USART被使能(UEN=1)时, 该位域不能被改写。
7	DENT	DMA发送使能 0: 关闭DMA发送模式 1: 开启DMA发送模式
6	DENR	DMA接收使能 0: 关闭DMA接收模式 1: 开启DMA接收模式
5	SCEN	智能卡模式使能 0: 智能卡模式禁用 1: 智能卡模式使能 USART被使能(UEN=1)时, 该位域不能被改写。 在USART1中, 该位保留。
4	NKEN	在智能卡模式NACK使能 0: 当出现校验错误时不发送NACK 1: 当出现校验错误时发送NACK USART被使能(UEN=1)时, 该位域不能被改写。 在USART1中, 该位保留。
3	HDEN	半双工使能 0: 禁用半双工模式 1: 开启半双工模式

		USART被使能(UEN=1)时，该位域不能被改写。
2	IRLP	IrDA低功耗模式 0: 正常模式 1: 低功耗模式 USART被使能(UEN=1)时，该位域不能被改写。
1	IREN	IrDA模式使能 0: IrDA禁用 1: IrDA被使能 USART被使能(UEN=1)时，该位域不能被改写。 在USART1中，该位保留。
0	ERRIE	多级缓存通信模式的错误中断使能 0: 禁用错误中断 1: 在多级缓存通信时，当USART_STAT寄存器的FERR位，ORERR位或NERR位被置位时，会产生中断。

18.4.4. USART 波特率寄存器 (USART_BAUD)

地址偏移: 0x0C

复位值: 0x0000_0000

该寄存器只能按字(32位)访问

当USART(UEN=1)被使能时，该寄存器不能被改写。

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

位/位域	名称	描述
31:16	保留	必须保持复位值
15:4	BRR[15:4]	波特率分频系数的整数部分 $DIV_INT[11:0] = BRR[15:4]$
3:0	BRR [3:0]	波特率分频系数的小数部分 如果OVSMOD = 0, USARTDIV [3:0] = BRR [3:0]; 如果OVSMOD = 1, USARTDIV [3:1] = BRR [2:0], BRR [3]必须被置0。

18.4.5. USART 保护时间和预分频器寄存器 (USART_GP)

地址偏移: 0x10

复位值: 0x0000_0000

该寄存器只能按字(32位)访问

USART被使能(UEN=1)时，该寄存器不能被改写。

在USART1中，该寄存器保留。

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

位/位域	名称	描述
31:16	保留	必须保持复位值
15:8	GUAT[7:0]	在智能卡模式下的保护时间值 USART被使能(UEN=1)时，该位域不能被改写。
7:0	PSC[7:0]	预分频器值 在红外低功耗模式下，对系统时钟进行分频已获得低功耗模式下的频率。寄存器的值是分频系数 00000000: 保留 - 不设置这个值 00000001: 1分频 00000010: 2分频 ... 在IrDA正常模式下的分频值 00000001: 仅仅能设为这个值 在智能卡模式下，对系统时钟进行分频的值存于PSC[4:0]位域中。PSC[7:5]位保持为复位值。分频系数是寄存器中值的两倍。 0000: 保留 -不设置这个值 00001: 2分频 00010: 4分频 00011: 6分频 ... USART被使能(UEN=1)时，该位域不能被改写。

18.4.6. USART 接收超时寄存器 (USART_RT)

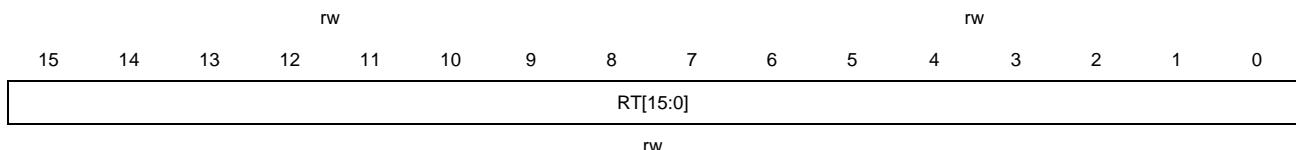
地址偏移: 0x14

复位值: 0x0000_0000

该寄存器只能按字(32位)访问

在USART1中，该寄存器保留。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BL[7:0]								RT[23:16]							



位/位域	名称	描述
31:24	BL[7:0]	<p>块长度</p> <p>这些位给出了智能卡T=1的接收时块的长度。它的值等于信息字节的长度+结束部分的长度(1-LEC/2-CRC) - 1。</p> <p>这个值可以在块接收开始时设置(用于需要从块的序言提取块的长度的情形)，这个只在每一个接收时钟周期只能设置一次。在智能卡模式下，当TBE=0时，块的长度计数器被清0。</p> <p>在其他模式下，当REN=0(禁用接收器)并且/或者当EBC位被写1时块的长度计数器被清0。</p>
23:0	RT[23:0]	<p>接收器超时门限</p> <p>该位域指定接收超时值，单位是波特时钟的时长</p> <p>标准模式下，如果在最后一个字节接收后，在RT规定的时长内，没有检测到新的起始位，RTF标志被置位。</p> <p>在智能卡模式，这个值被用来实现CWT和BWT。在这种情况下，超时检测是从最后一个接收字节的起始位开始。</p> <p>这些位可以在工作时改写。假如一个新数据到来的时间比RT规定的晚，RTF标志会被置位。对于每个接收字符，这个值只能改写一次。</p>

18.4.7. USART 请求寄存器 (USART_CMD)

地址偏移: 0x18

复位值: 0x0000_0000

该寄存器只能按字(32位)访问



位/位域	名称	描述
31:5	保留	必须保持复位值
4	TXFCMD	<p>发送数据清空请求</p> <p>向该位写1去置位TBE标志位，以取消发送数据。</p> <p>在USART1中，该位保留。</p>
3	RXFCMD	接收数据清空请求

向该位写1来清除RBNE标志位，以丢弃未读的接收数据。

2	MMCMD	静默模式请求 向该位写1使USART进入静默模式并且置位RWU标志位。
1	SBKCMD	发送断开帧请求 向该位写1置位SBKF标志并使USART在空闲时发送一个断开帧。
0	ABDCMD	自动波特率检测请求 向该位写1会清除USART_STAT寄存器的ABDF位，并且在下一个数据接收帧开始自动检测波特率。 在USART1中，该位保留。

18.4.8. USART 状态寄存器 (USART_STAT)

地址偏移: 0x1C

复位值: 0x0000_00C0

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留										REA	TEA	WUF	RWU	SBF	AMF	BSY
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ABDF	ABDE	保留	EBF	RTF	CTS	CTSF	LBDF	TBE	TC	RBNE	IDLEF	ORERR	NERR	FERR	PERR	
r	r		r	r	r	r	r	r	r	r	r	r	r	r	r	

位/位域	名称	描述
31:23	保留	必须保持复位值
22	REA	接收使能通知标志 这位反映了USART核心逻辑的接收使能状态，该位可以通过硬件设置。 0: USART核心接收逻辑禁用 1: USART核心接收逻辑被使能
21	TEA	发送使能通知标志 这位反映了USART核心逻辑的发送使能状态，该位可以通过硬件设置。 0: USART核心发送逻辑禁用 1: USART核心发送逻辑被使能
20	WUF	从深度睡眠模式唤醒标志 0: 没有从深度睡眠模式唤醒 1: 已从深度睡眠模式唤醒，如果在USART_CTL2寄存器的WUFIE=1并且MCU处于深度睡眠模式，将引发一个中断。 当检测到一个唤醒事件时，该位通过硬件置位，这个事件在WUM位域被定义。 向USART_INTC寄存器中的WUC写1，该位被清0。 当UESM被清0时，该位清0。

		在USART1中，该位保留。
19	RWU	<p>接收器从静默模式唤醒</p> <p>这位表示USART处于静默模式。</p> <p>0: 接收器在工作状态</p> <p>1: 接收器在静默状态</p> <p>当在唤醒和静默模式切换时，它通过硬件清0或者置1。静默模式控制 (地址帧还是空闲帧) 是用通过USART_CTL0寄存器的WAKE位选择。</p> <p>如果选择空闲信号唤醒，只能通过向USART_CMD寄存器的MMCMD位写1来将该位置位。</p>
18	SBF	<p>断开信号发送标识</p> <p>0: 没发送断开字符</p> <p>1: 将要发送断开字符</p> <p>该位表示一个断开发送信号被请求。</p> <p>通过向USART_CMD寄存器的SBKCMD写1来置位。</p> <p>在断开帧的停止位发送期间，硬件清0。</p>
17	AMF	<p>ADDR匹配标志</p> <p>0: ADDR和接收到的字符不匹配</p> <p>1: ADDR和接收到的字符匹配，如果USART_CTL0寄存器的AMIE=1，将引发一个中断。</p> <p>当接收到ADDR [7:0]中定义的字符时，硬件置位。</p> <p>通过向USART_INTC寄存器的AMC写1清0。</p>
16	BSY	<p>忙标志</p> <p>0: USART处于空闲</p> <p>1: USART正在接收</p>
15	ABDF	<p>自动波特率检测标志</p> <p>0: 没有自动波特率检测完成</p> <p>1: 自动波特率检测完成</p> <p>当自动波特率检测完成时，硬件置位。</p> <p>通过向USART_CMD寄存器的ABDCMD写1，该位清0，并请求新的波特率检测。</p> <p>在USART1中，该位保留。</p>
14	ABDE	<p>自动波特率检测错误</p> <p>0: 没有自动波特率检测错误出现</p> <p>1: 自动波特率检测错误出现</p> <p>如果波特率超出范围或者字符对比失败，硬件置位。</p> <p>通过向USART_CTL2寄存器的ABDRQ位写1清0。</p> <p>在USART1中，该位保留。</p>
13	保留	必须保持复位值
12	EBF	<p>块结束标志</p> <p>0: 块没有结束</p> <p>1: 块结束已到 (足够的字节数)，如果USART_CTL1寄存器的EBIE=1，将引发一个中断。</p> <p>当接收到的字节数 (从块开始，包括序言部分) 等于或大于BLEN + 4，硬件置位。</p>

		通过向USART_INTC寄存器的EBC写1清0。 在USART1中，该位保留。
11	RTF	接收超时标志 0: 尚未超时 1: 已经超时，如果USART_CTL1寄存器的RTIE被置位，将会引发中断。 如果空闲的时间已经超过了在USART_RT寄存器中设定的RT值，通过硬件置1。 通过向USART_INTC寄存器的RTC位写1清0。 在智能卡模式，这个超时相当于CWT或BWT计时。 在USART1中，该位保留。
10	CTS	CTS电平 这个值等于nCTS输入引脚电平的反向拷贝。 0: nCTS输入引脚高电平 1: nCTS输入引脚低电平
9	CTSF	CTS变化标志 0: nCTS状态线没有变化 1: nCTS状态线发生变化 如果USART_CTL2寄存器的CTSIE位置位，将引发中断。 当nCTS输入变化时，由硬件置位。 通过向USART_INTC寄存器的CTSC位写1，清零该位。
8	LBDF	LIN断开检测标志 0: 没有检测到LIN断开字符 1: 检测到LIN断开字符。当USART_CTL1寄存器的LBDIE位被置位时，将会有中断产生。 当LIN断开帧被检测到的时候，硬件置位。 通过向USART_INTC寄存器的LBDC位写1，清零该位。 在USART1中，该位保留。
7	TBE	发送数据寄存器空 0: 数据没有发送到移位寄存器 1: 数据发送到移位寄存器。如果USART_CTL0寄存器的TBEIE位置位，将会有中断产生。 当USART_TDATA寄存器的内容已经被转移到移位寄存器或者向USART_CMD寄存器的TXFCMD位写1时，由硬件置位。 通过向USART_TDATA寄存器中写数据来清0。
6	TC	发送完成 0: 发送没有完成 1: 发送完成。如果USART_CTL0寄存器的TCIE被置位，将会有中断产生。 如果一个包含数据的帧的发送完成且TBE被置位，该位由硬件置位。 通过向USART_INTC寄存器的TCC位写1清0。
5	RBNE	读数据缓冲区非空 0: 没有接收到数据 1: 已接收到数据并且可以读取。当寄存器USART_CTL0的RBNEIE位被置位，将会有中断产生。

当接收移位寄存器的内容已经被转移到寄存器USART_RDATA，由硬件置位。

通过读USART_RDATA寄存器或向USART_CMD寄存器的RXFCMD位写1清0。

4	IDLEF	空闲线检测标志 0: 没检测到空闲线 1: 检测到空闲线。,如果USART_CTL0寄存器的IDLEIE位置1, 将会有中断产生。 当检测到空闲线时, 通过硬件置位。直到RBNE位置位, 否则它不会被再次置位。 向USART_INTC寄存器的IDLEC位写1清0。
3	ORERR	溢出错误 0: 未检测到溢出错误 1: 检测到溢出错误。在多级缓存通信中, 如果寄存器USART_CTL0的RBNEIE位置位, 将会引发中断。如果寄存器USART_CTL2的ERRIE位置位也会引发中断。 在RBNE置位的情况下, 如果接收移位寄存器的数据传递给USART_RDATA寄存器, 将会由硬件置位。 向USART_INTC寄存器的OREC位写1清0。
2	NERR	噪声错误标志 0: 未检测到噪声错误 1: 检测到噪声错误。在多级缓存通信中, 如果寄存器USART_CTL2的ERRIE位置位, 将会有中断产生。 在接收帧的时候检测到噪声错误, 将会由硬件置位。 向寄存器USART_INTC的NEC位写1清0。
1	FERR	帧错误 0: 未检测到帧错误 1: 检测到帧错误或者断开字符。在多级缓存通信中, 如果寄存器USART_CTL2的ERRIE位置位, 将会有中断产生。 当一个不同步, 强噪声或者断开字符被检测到时, 硬件置位。在智能卡模式下, 当发送次数达到上限, 仍然没有收到成功发送应答(卡一直响应NACKs), 该位也将被置位。 向USART_INTC寄存器的FEC位写1清0。
0	PERR	校验错误 0: 未检测到校验错误 1: 检测到校验错误 在多级缓存通信中, 如果寄存器USART_CTL0的PERIE位置位, 将会有中断产生。 当在接收模式的时候检测到校验错误, 将会由硬件置位。 向USART_INTC寄存器的PEC位写1清0。

18.4.9. USART 中断标志清除寄存器 (USART_INTC)

地址偏移: 0x20

复位值: 0x0000_0000

该寄存器只能按字(32位)访问

保留														WUC	保留	AMC	保留
														W	W		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
保留	EBC	RTC	保留	CTSC	LBDC	保留	TCC	保留	IDLEC	OREC	NEC	FEC	PEC				
	W	W		W	W		W	W	W	W	W	W	W	W	W	W	W

位/位域	名称	描述
31:21	保留	必须保持复位值
20	WUC	从深度睡眠模式唤醒标志的清除 向该位写1清除USART_STAT寄存器的WUF位。 在USART1中，该位保留。
19:18	保留	必须保持复位值
17	AMC	ADDR匹配标志清除 向该位写1清除USART_STAT寄存器的AMF位。
16:13	保留	必须保持复位值
12	EBC	块结束标志清除 向该位写1清除USART_STAT寄存器的EBF位。 在USART1中，该位保留。
11	RTC	接收超时标志清除 向该位写1清除USART_STAT寄存器的RTF标志。 在USART1中，该位保留。
10	保留	必须保持复位值
9	CTSC	CTS变化标志清除 向该位写1清除USART_STAT寄存器的CTSF位。
8	LBDC	LIN断开字符检测标志清除 向该位写1清除USART_STAT寄存器的LBDF标志位。 在USART1中，该位保留。
7	保留	必须保持复位值
6	TCC	发送完成标志清除 向该位写1清除USART_STAT寄存器的TC位。
5	保留	必须保持复位值
4	IDLEC	空闲线检测标志清除 向该位写1清除USART_STAT寄存器的IDLEF位。
3	OREC	溢出标志清除 向该位写1清除USART_STAT寄存器的ORERR位。

2	NEC	噪声检测清除 向该位写1清除USART_STAT寄存器的NERR位。
1	FEC	帧格式错误标志清除 向该位写1清除USART_STAT寄存器的FERR位。
0	PEC	校验错误标志清除 向该位写1清除USART_STAT寄存器的PERR位。

18.4.10. USART 数据接收寄存器 (USART_RDATA)

地址偏移: 0x24

复位值: 未定义

该寄存器只能按字(32位)访问

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

r

位/位域	名称	描述
31:9	保留	必须保持复位值
8:0	RDATA[8:0]	接收数据的值 包含接收到的数据字节 如果接收到的数据打开了奇偶校验位(USART_CTL0寄存器的PCEN置1), 那么接收到的数据的最高位(第7位或8位, 取决于数据的长度)是奇偶校验位。

18.4.11. USART 数据发送寄存器 (USART_TDATA)

地址偏移: 0x28

复位值: 未定义

该寄存器只能按字(32位)访问

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

rw

位/位域	名称	描述
------	----	----

31:9	保留	必须保持复位值
8:0	TDATA[8:0]	发送数据的值 包含发送的数据字节 如果发送到的数据打开了奇偶校验位(USART_CTL0寄存器的PCEN置1)，那么发送的数据的最高位(第7位或8位取决于数据的长度)将会被奇偶校验位替代。 只有当USART_STAT寄存器的TBE位被置位时，这个寄存器才可以改写。

18.4.12. USART 接收 FIFO 控制和状态寄存器 (USART_RFCS)

地址偏移: 0xD0

复位值: 0x0000_0400

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RFFINT	RFCNT[2:0]	RFF	RFE	RFFIE	RFEN	保留								ELNACK	

位/位域	名称	描述
31:16	保留	必须保持复位值
15	RFFINT	接收FIFO满中断标志
14:12	RFCNT[2:0]	接收FIFO计数值
11	RFF	接收FIFO满标志 0: 接收FIFO不为满 1: 接收FIFO满
10	RFE	接收FIFO空标志 0: 接收FIFO不为空 1: 接收FIFO空
9	RFFIE	接收FIFO满中断使能 0: 禁止接收FIFO满中断 1: 使能接收FIFO满中断
8	RFEN	接收FIFO使能 当UESM=1, 该位置位。 0: 禁止使用接收FIFO 1: 使能接收FIFO
7:1	保留	必须保持复位值
0	ELNACK	若选择了智能卡模式, 提前NACK 如果检测到校验位错误, NACK脉冲提前1/16位的时间。

0: 若选择了智能卡模式，禁止提前NACK

1: 若选择了智能卡模式，使能提前NACK

在USART1中，该位保留。

19. 内部集成电路总线接口 (I2C)

19.1. 简介

I2C (内部集成电路总线)模块提供了符合工业标准的两线串行制接口，可用于 MCU 和外部 I2C 设备的通讯。I2C 总线使用两条串行线：串行数据线 SDA 和串行时钟线 SCL。

I2C 接口模块实现了 I2C 协议的标速模式，快速模式以及快速+ 模式，具备 CRC 计算和校验功能、支持 SMBus(系统管理总线)和 PMBus (电源管理总线)，此外还支持多主机 I2C 总线架构。I2C 接口模块也支持 DMA 模式，可有效减轻 CPU 的负担。

19.2. 主要特性

- 并行总线至 I2C 总线协议的转换及接口；
- 同一接口既可实现主机功能又可实现从机功能；
- 主从机之间的双向数据传输；
- 支持 7 位和 10 位的地址模式和广播寻址；
- 支持 I2C 多主机模式；
- 支持标速(最高 100 kHz)，快速(最高 400 kHz)和快速+ 模式(最高 1MHz)；
- 从机模式下可配置的 SCL 主动拉低；
- 支持 DMA 模式；
- 兼容 SMBus 2.0 和 PMBus；
- 两个中断：字节成功发送中断和错误事件中断；
- 可选择的 PEC (报文错误校验)生成和校验；

19.3. 功能描述

I2C 接口的内部结构如 [图 19-1. I2C 模块框图](#) 所示。

图 19-1. I2C 模块框图

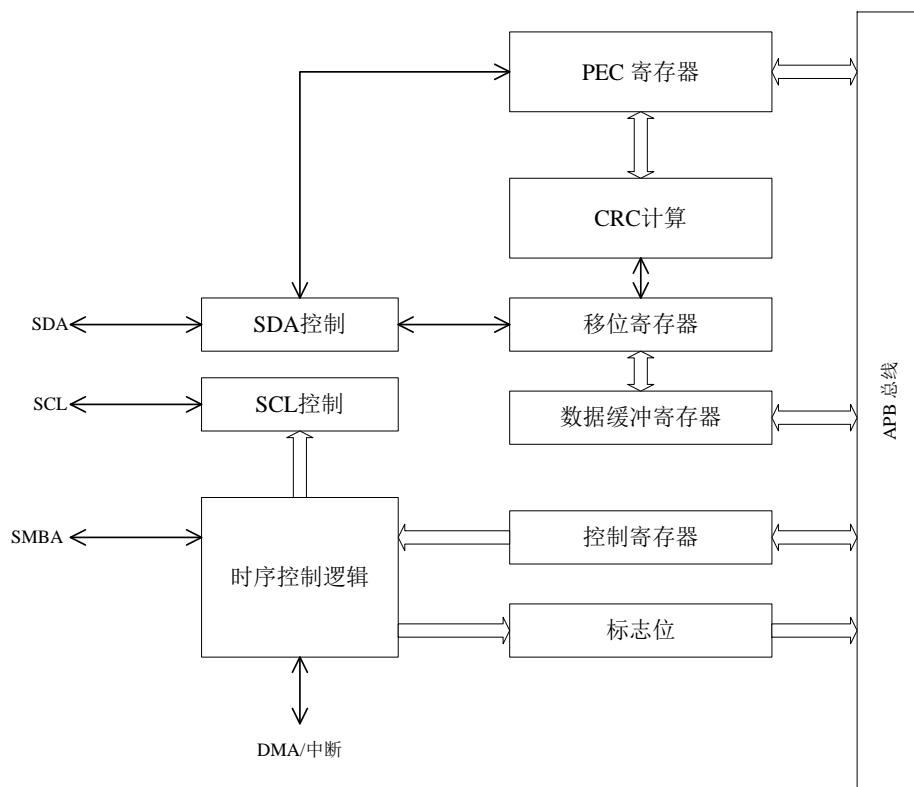


表 19-1. I2C 总线术语说明（参考飞利浦 I2C 规范）

术语	说明
发送器	发送数据到总线的设备
接收器	从总线接收数据的设备
主机	初始化数据传输，产生时钟信号和结束数据传输的设备
从机	由主机寻址的设备
多主	不破坏信息的前提下同时控制总线的多个主机
同步	同步两个或更多设备之间的时钟信号的过程
仲裁	如果超过一个主机同时试图控制总线，只有一个主机被允许，且获胜主机的信息不被破坏，保证上述的过程叫仲裁

19.3.1. SDA 线和 SCL 线

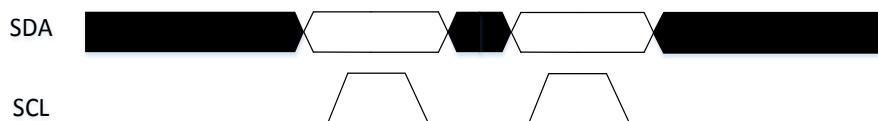
I2C 模块有两条接口线：串行数据 SDA 线和串行时钟 SCL 线。连接到总线上的设备通过这两根线互相传递信息。SDA 和 SCL 都是双向线，通过一个电流源或者上拉电阻接到电源正极。当总线空闲时，两条线都是高电平。连接到总线的设备输出极必须是开漏或者开集，以提供线与功能。I2C 总线上的数据在标准模式下可以达到 100Kbit/s，在快速模式下可以达到 400Kbit/s。

当 I2C_FMPCFG 中 FMPEN 被置位时，在快速+ 模式下可达 1Mbit/s。由于 I2C 总线上可能会连接不同工艺的设备，逻辑‘0’和逻辑‘1’的电平并不是固定的，取决于 V_{DD} 的实际电平。

19.3.2. 数据有效性

时钟信号的高电平期间 SDA 线上的数据必须稳定。只有在时钟信号 SCL 变低的时候数据线 SDA 的电平状态才能跳变(如 [图 19-2. 数据有效性](#))。每个数据比特传输需要一个时钟脉冲。

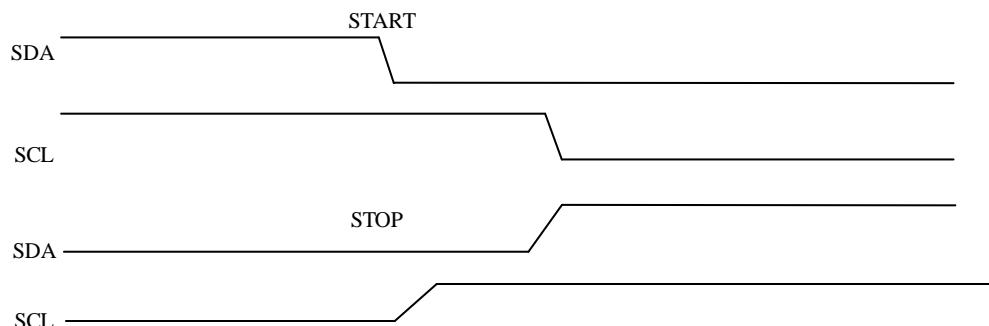
图 19-2. 数据有效性



19.3.3. 开始和停止状态

所有的数据传输起始于一个 START(S)结束于一个 STOP(P)(参见 [图 19-3. 开始和停止状态](#))。START 起始位定义为，在 SCL 为高时，SDA 线上出现一个从高到低的电平转换。STOP 结束位定义为，在 SCL 为高时，SDA 线上出现一个从低到高的电平转换。

图 19-3. 开始和停止状态



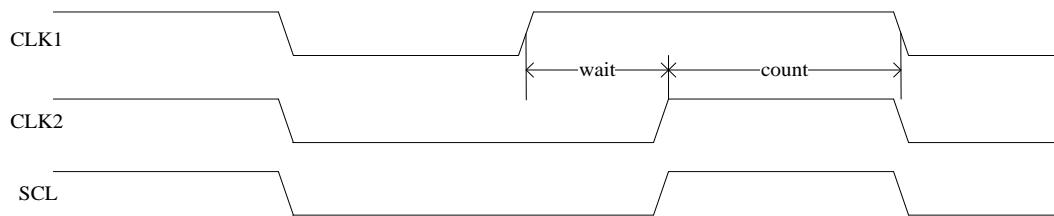
19.3.4. 时钟同步

两个主机可以同时在空闲总线上开始传送数据，因此必须通过一些机制来决定哪个主机获取总线的控制权，这一般是通过时钟同步和仲裁来完成的。单主机系统下不需要时钟同步和仲裁机制。

时钟同步通过 SCL 线的线与来实现。这就是说 SCL 线的高到低切换会使器件开始数它们的低电平周期，而且一旦主器件的时钟变低电平，它会使 SCL 线保持这种状态直到到达时钟的高电平(参见 [图 19-4. 时钟同步](#))。但是如果另一个时钟仍处于低电平周期，这个时钟的低到高切换不会改变 SCL 线的状态。因此 SCL 线被有最长低电平周期的器件保持低电平。此时低电平

周期短的器件会进入高电平的等待状态。

图 19-4. 时钟同步



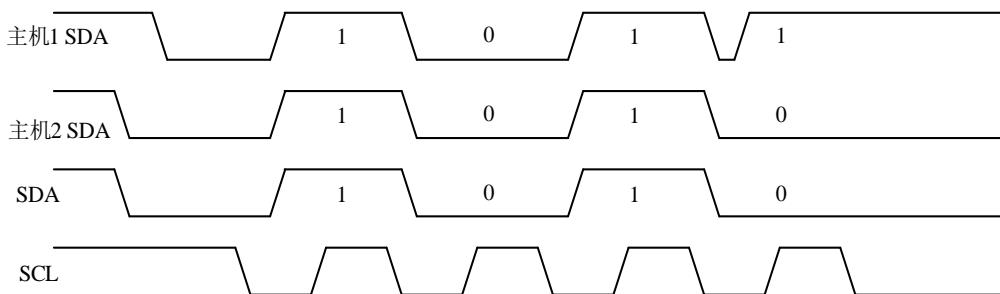
19.3.5. 仲裁

仲裁和同步一样，都是为了解决多主机情况下的总线控制冲突。仲裁的过程与从机无关。

只有在总线空闲的时候主机才可以启动传输。两个主机可能在START起始位的最短保持时间内在总线上产生一个有效的START起始位，这种情况下需要仲裁来决定由哪个主机来完成传输。

仲裁逐位进行，在每一位的仲裁期间，当SCL为高时，每个主机都检查SDA电平是否和自己发送的相同。仲裁的过程需要持续很多位。理论上讲，如果两个主机所传输的内容完全相同，那么他们能够成功传输而不出现错误。如果一个主机发送高电平但检测到SDA电平为低，则认为自己仲裁失败并关闭自己的SDA输出驱动，而另一个主机则继续完成自己的传输。

图 19-5. SDA 线仲裁



19.3.6. I2C 通讯流程

每个I2C设备(不管是微控制器，LCD驱动，存储器或者键盘接口)都通过唯一的地址进行识别，根据设备功能，他们既可以是发送器也可作为接收器。

I2C从机检测到I2C总线上的START起始位之后，就开始从总线上接收地址，之后会把从总线接收到的地址和自身的地址（通过软件编程）进行比较，一旦两个地址相同，I2C从机将发送一个确认应答(ACK)，并响应总线的后续命令：发送或接收所要求的数据。此外，如果软件开启了广播呼叫，则I2C从机始终对一个广播地址 (0x00)发送确认应答。I2C模块始终支持7位和10位的地址。

I2C主机负责产生START起始位和STOP结束位来开始和结束一次传输，并且负责产生SCL时钟。

图 19-6. 7 位地址的 I2C 通讯流程

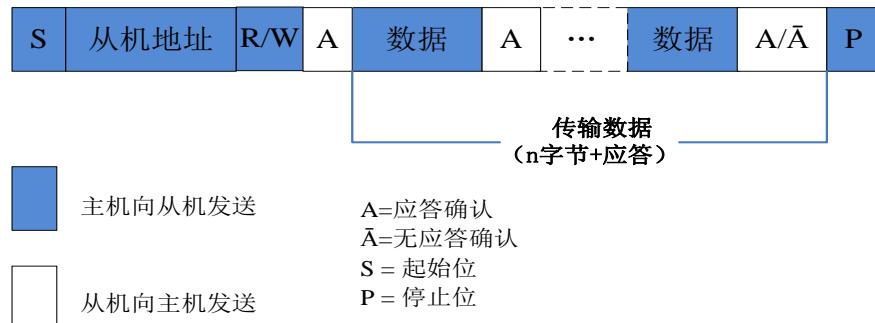
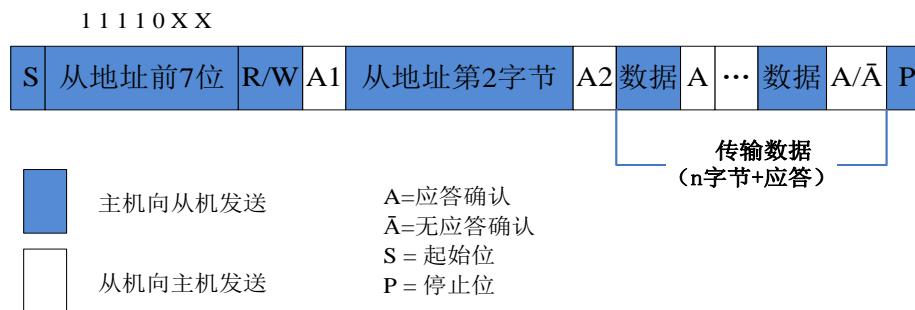


图 19-7. 10 位地址的 I2C 通讯流程



19.3.7. 软件编程模型

一个I2C设备例如LCD驱动器可能只是作为一个接收器，但是一个存储器既可以接收数据，也能发送数据。除了按照发送/接收方来区分，I2C设备也分为数据传输的主机和从机。主机是指负责初始化总线上数据的传输并产生时钟信号的设备，此时任何被寻址的设备都是从机。

不管I2C设备是主机还是从机，都可以发送或接收数据，因此，I2C设备有以下4种运行模式：

- 主机发送方；
- 主机接收方；
- 从机发送方；
- 从机接收方。

I2C模块支持以上四种模式。系统复位以后，I2C默认工作在从机模式下。通过软件配置使I2C在总线上发送一个START起始位之后，I2C变为主机模式，软件配置在I2C总线上发送STOP结束位后，I2C又变回从机模式。

从机发送模式下的软件流程

在从机模式下要发送数据到I2C总线，软件应该按照下面的步骤来运行操作：

1. 首先，软件应该使能I2C外设时钟，以及配置I2C_CTL1中时钟相关寄存器来确保正确的I2C时序。使能和配置以后，I2C运行在默认的从机模式状态，等待START起始位和地址。
2. 接收一个START起始位及随后的地址，地址可以是7位格式也可以是10位格式，I2C硬件

将I2C_STAT0寄存器的ADDSEND位置1，此位应该被软件查询或者中断监视，发现置位后，软件应该读I2C_STAT0寄存器然后读I2C_STAT1寄存器来清除ADDSEND位。如果地址是10位格式，I2C主机应该接着再产生一个START (Sr) 并发送一个地址头到I2C总线。

从机在检测到START (Sr) 和紧接着的地址头之后会继续将ADDSEND位置1。软件也可以通过读I2C_STAT0寄存器和接着读I2C_STAT1寄存器来第二次清除ADDSEND位。

3. 现在I2C进入数据发送状态，由于移位寄存器和数据寄存器I2C_DATA都是空的，硬件将TBE位置1。软件此时可以写入第一个字节数据到I2C_DATA寄存器，但是TBE位并没有被清0，因为写入I2C_DATA寄存器的字节被立即移入内部移位寄存器。当移位寄存器非空的时候，I2C开始发送数据到I2C总线。
4. 第一个字节的发送期间，软件可以写第二个字节到I2C_DATA，此时TBE位被清0，因为I2C_DATA寄存器和移位寄存器都不是空。
5. 第一个字节的发送完成之后，TBE被再次置起，软件可以写第三个字节到I2C_DATA，同时TBE位被清0。在此之后，任何时候TBE被置1，只要依然有数据待被发送，软件都可以写入一个字节到I2C_DATA寄存器。
6. 倒数第二个字节发送期间，软件写最后一个数据到I2C_DATA寄存器来清除TBE标志位，之后就再不用关心TBE的状态。TBE位会在倒数第二个字节发送完成后置起，直到检测到STOP结束位时被清0。
7. 根据I2C协议，I2C主机将不会对接收到的最后一个字节发送应答，所以在最后一个字节发送结束后，I2C从机的AEER会置起以通知软件发送结束。软件写0到AEER位可以清除此位。

图 19-8. 从机发送模式


从机接收模式下的软件流程

如[图 19-9. 从机接收模式](#)所示，在从机模式下接收数据时，软件应该遵循这些步骤来操作：

1. 首先，软件应该使能I2C外设时钟，以及配置I2C_CTL1中时钟相关寄存器来确保正确的I2C时序。使能和配置以后，I2C运行在默认的从机模式状态，等待START起始位以及地址。
2. 在接收到START起始条件和匹配的7位或10地址之后，I2C硬件将I2C状态寄存器的ADDSEND位置1，此位应该通过软件轮询或者中断来检测，发现置起后，软件通过先读I2C_STAT0寄存器然后读I2C_STAT1寄存器来清除ADDSEND位。一旦ADDSEND位被清0，I2C就开始接收来自I2C总线的数据。
3. 一旦接收到第一个字节，RBNE位被硬件置1，软件可以读取I2C_DATA寄存器的第一个字节，此时RBNE位也被清0。
4. 任何时候RBNE被置1，软件可以从I2C_DATA寄存器读取一个字节。
5. 接收到最后一个字节后，RBNE被置1，软件可以读取最后的字节。
6. 当I2C检测到I2C总线上一个STOP结束位，STPDET位被置1，软件通过先读I2C_STAT0寄存器再写I2C_CTL0寄存器来清除STPDET位。

图 19-9. 从机接收模式


主机发送模式下的软件流程

如[图19-10. 主机发送模式](#)所示，在主机模式下发送数据到I2C总线时，软件应该遵循这些步骤来运行I2C模块：

1. 首先，软件应该使能I2C外设时钟，以及配置I2C_CTL1中时钟相关寄存器来确保正确的I2C时序。使能和配置以后，I2C运行在默认的从机模式状态，等待START起始位，随后等待I2C总线寻址。
2. 软件将START位置1，在I2C总线上产生一个START起始位。
3. 发送一个START起始位后，I2C硬件将I2C_STAT0的SBSEND位置1然后进入主机模式。现在软件应该读I2C_STAT0寄存器然后写一个7位地址位或带有地址头的10位地址位到I2C_DATA寄存器来清除SBSEND位。一旦SBSEND位被清0，I2C就开始发送地址或者地址头到I2C总线。如果发送的地址是10位带地址头的地址，硬件在发送地址头的时候会将ADD10SEND位置1，软件应该通过读I2C_STAT0寄存器然后写10位低地址到I2C_DATA来清除ADD10SEND位。
4. 7位或10位的地址位发送出去之后，I2C硬件将ADDSEND位置1，软件应该清除ADDSEND位（通过读I2C_STAT0寄存器然后读I2C_STAT1寄存器）。
5. I2C进入数据发送状态，因为移位寄存器和数据寄存器I2C_DATA都是空的，所以硬件将TBE位置1。此时软件可以写第一个字节数据到I2C_DATA寄存器，但是TBE位此时不会被清零，因为写入I2C_DATA寄存器的字节被立即移入内部移位寄存器。一旦移位寄存器非空，I2C就开始发送数据到总线。
6. 在第一个字节的发送过程中，软件可以写第二个字节到I2C_DATA，此时TBE会被清零。
7. 任何时候TBE被置1，软件都可以向I2C_DATA寄存器写入一个字节，只要还有数据待发送。

8. 在倒数第二个字节发送过程中，软件写入最后一个字节数据到I2C_DATA来清除TBE标志位，此后就不用关心TBE位的状态。TBE位会在倒数第二个字节发送完成后被置起，直到发送STOP结束位时被清零。
9. 最后一个字节发送结束后，I2C主机将BTC位置起，因为移位寄存器和I2C_DATA寄存器此时都为空。软件此时应该配置STOP来发送一个STOP结束位，此后TBE和BTC状态位都将被清0。

图 19-10. 主机发送模式



主机接收模式下的软件流程

在主机接收模式下，主机需要为最后一个字节接收产生NACK，然后发送STOP结束位。因此，需要额外注意以确保最后接收到数据的正确性。下面提供了两种针对主机接收模式的软件编程方案，方案A需要保证软件能对I2C的中断进行快速响应，方案B则不需要。

方案 A

1. 首先，软件应该使能I2C外设时钟，以及配置I2C_CTL1中时钟相关寄存器来确保正确的I2C时序。使能和配置以后，I2C运行在默认的从机模式状态，等待START起始位，随后等待I2C总线寻址。
2. 软件将START位置1，从而在I2C在总线上产生一个START起始位

3. 发送一个START起始位后, I2C硬件将I2C_STAT0的SBSEND位置1然后进入主机模式。现在软件应该读I2C_STAT0寄存器然后写一个7位地址位或带有地址头的10位地址位到I2C_DATA寄存器来清除SBSEND位。一旦SBSEND位被清0, I2C就开始发送地址或者地址头到I2C总线。如果发送的地址是10位带地址头的地址, 硬件在发送地址头的时候会先将ADD10SEND位置1, 软件应该通过读I2C_STAT0寄存器然后写10位低地址到I2C_DATA来清除ADD10SEND位
4. 7位或10位的地址位发送出去之后, I2C硬件将ADDSEND位置1, 软件应该清除ADDSEND位, 通过读I2C_STAT0寄存器然后读I2C_STAT1寄存器。如果地址是10位格式, 软件应该再次将GENSTA位置1来重新产生一个START(Sr)。在START产生后, SBSEND位会被置1。软件应该通过先读I2C_STAT0然后写地址头到I2C_DATA来清除SBSEND位, 然后地址头被发到I2C总线, ADDSEND再次被置1。软件应该再次通过先读I2C_STAT0然后读I2C_STAT1来清除ADDSEND位。
5. 一旦接收到第一个字节, 硬件会将RBNE位置1。此时软件可以从I2C_DATA寄存器读取第一个字节, 之后RBNE位被清0。
6. 此后任何时候RBNE被置1, 软件就可以从I2C_DATA寄存器读取一个字节。
7. 接收完倒数第二个字节(N-1)数据之后, 软件应该立即将ACKEN位清0, 并将STOP位置1, 这一过程需要在最后一个字节接收完毕之前完成, 以确保NACK发送给最后一个字节。
8. 最后一个字节接收完毕后, RBNE位被置1, 软件可以读取最后一个字节。由于ACKEN已经在前一步骤中被清0, I2C不再为最后一个字节发送ACK, 并在最后一个字节发送完毕后产生一个STOP结束位。

以上步骤要求字节数目N>1, 如果N=1, 步骤7应该在步骤4之后就执行, 且需要在字节接收完成之前完成。

图 19-11. 主机接收使用方案 A 模式


方案 B

1. 首先，软件应该使能I2C外设时钟，配置I2C_CTL1中时钟相关寄存器来确保正确的I2C时序。初始化完成之后，I2C运行在默认的从机模式状态，等待START起始位和地址。
2. 软件将GENSTA位置1从而产生一个起始位
3. 发送一个START起始位后，I2C硬件将I2C_STAT0的SBSEND位置1然后进入主机模式。
现在软件应该读I2C_STAT0寄存器然后写一个7位地址位或带有地址头的10位地址位到I2C_DATA寄存器来清除SBSEND位。一旦SBSEND位被清0，I2C就开始发送地址或者地址头到I2C总线。如果发送的地址是10位带地址头的地址，硬件在发送地址头的时候会先将ADD10SEND位置1，软件应该通过读I2C_STAT0寄存器然后写10位低地址到I2C_DATA来清除ADD10SEND位。
4. 7位或10位的地址位发送出去之后，I2C硬件将ADDSEND位置1，软件应该清除ADDSEND位，通过读I2C_STAT0寄存器然后读I2C_STAT1寄存器。如果地址是10位格式，软件应该接着将GENSTA位再次置1来产生一个开始条件(Sr)，Sr被发送出去以后SBSEND位被再

次置1。软件应该通过先读I2C_STAT0然后写地址头到I2C_DATA来清除SBSEND位，然后地址头被发到I2C总线，ADDSEND再次被置1。软件应该再次通过先读I2C_STAT0然后读I2C_STAT1来清除ADDSEND位。

5. 一旦第一个字节被接收，RBNE位会被硬件置1。此时软件可从I2C_DATA寄存器读取出第一个字节，同时RBNE位被清0。
6. 此后任何时候，一旦RBNE位被置1，软件就可以从I2C_DATA寄存器读取一个字节的数据，直到主机接收了N-3个字节。

如[图19-12. 主机接收使用方案B模式](#)所示，第N-2个字节还没被软件读出，之后第N-1个字节被接收，此时BTC和RBNE都被置位，总线就会被主机锁死以阻止最后一个字节的接收。然后软件应该清除ACKEN位。

7. 软件从I2C_DATA读出倒数第三个(N-2)字节数据，同时也将BTC位清0。此后第N-1个字节从移位寄存器被移到I2C_DATA，总线得到释放然后开始接收最后一个字节，由于ACKEN已经被清除，因此主机不会给最后一个字节数据发送ACK响应。
8. 最后一个字节接收完毕后，硬件再次把BTC位和RBNE置1，并拉低SCL，软件将STOP位置1，主机发出一个STOP结束位。
9. 软件读取第N-1个字节，清除BTC。此后最后一个字节从移位寄存器被移动到I2C_DATA。
10. 软件读取最后一个字节，清除RBNE。

以上步骤需要字节数字N>2，N=1和N=2的情况近似。

N=1

在第4步，软件应该在清除ADDSEND位之前将ACK位清0，在清除ADDSEND位之后将STOP位置1。当N=1时步骤5是最后一步。

N=2

在第2步，软件应该在START置1之前将POAP置1。在第4步，软件应该在清除ADDSEND位之前将ACK位清0。在第5步，软件应该一直等到BTC位被置1然后将STOP位置1且读取I2C_DATA两次。

图 19-12. 主机接收使用方案 B 模式


19.3.8. SCL 线控制

SCL 线拉低功能是为了避免在接收时发生上溢错误以及在发送时发生下溢错误。如在软件编程模型中所示，在发送模式，当 TBE 和 BTC 被置位，发送器保持 SCL 线为低电平直到下一个发送数据写入传输缓冲区寄存器。在接收模式，当 RBNE 和 BTC 被置位，发送器保持 SCL 线为低电平直到传输缓冲区寄存器里的数据被读出。

当工作在从模式的时候，可以通过置位 I2C_CTL0 寄存器的 DISSTRC 位禁止 SCL 线拉低功能。如果该位置位，软件要能足够快的处理 TBE，RBNE 和 BTC 状态，否则上溢或下溢的情

况可能会发生。

19.3.9. DMA 模式下数据传输

按照前面的软件流程，每当 TBE 位和 RBNE 位被置 1 之后，软件都应该写或读一个字节，这样将导致 CPU 的负荷较重。I2C 的 DMA 功能可以在 TBE 或 RBNE 位置 1 时，自动进行一次写或读操作，从而减轻了 CPU 的负荷，具体 DMA 的配置请参看 DMA 相关章节。

DMA 请求通过 I2C_CTL1 寄存器的 DMAON 位使能。该位应该在清除 ADDSEND 状态位前被置位。如果一个从机的 SCL 线延长功能被禁止，DMAON 位应该在 ADDSENT 事件前被置位。

参考 DMA 控制器的关于 DMA 流的配置方法说明。DMA 必须在 I2C 传输开始之前配置和使能。当指定个数的字节已经传输完成，DMA 控制器将产生一个传输结束（EOC）中断。

当主机接收到两个或两个以上字节时，I2C_CTL1 寄存器的 DMALST 位应该置位。在接收到最后一个字节之后，I2C 主机不发送 NACK。在 DMA EOC 中断 ISR 中，软件置位 STOP 位，产生一个停止状态。

当主机仅接收到一个字节时，清除 ADDSEND 状态前 ACKEN 位必须被清除。在清除 ADDSEN 状态后或在 DMA EOC 中断 ISR 中，软件置位 STOP 位，产生一个停止状态。

19.3.10. 报文错误校验

I2C 模块中有一个 PEC 模块，它使用 CRC-8 计算器来执行 I2C 数据的报文校验，CRC 多项式为 $x^8 + x^2 + x + 1$ ，和 SMBus 协议兼容。将 PECEN 位置 1 就可以使能 PEC 功能。PEC 将会计算所有通过 I2C 总线发送的数据（包括地址）。在非 DMA 模式下，软件可以通过配置 PECTRANS 来控制 I2C 在最后一个字节发送完毕后发送 PEC 值，或者在接收完成后检查接收到的 PEC 值是否正确。在 DMA 模式下，如果 PECEN 位被置 1，I2C 将自动发送或者检查 PEC 值。

19.3.11. SMBus 支持

系统管理总线（System Management Bus，简写为 SMBus 或 SMB）是一种结构简单的单端双线制总线，可实现轻量级的通信需求。一般来说，SMBus 最常见于计算机主板，主要用于电源传输 ON/OFF 指令的通信。SMBus 是 I2C 的一种衍生总线形式，主要用于计算机主板上的低带宽设备间通信，尤其是与电源相关的芯片，例如笔记本电脑的可充电电池子系统（参见 Smart Battery Data）。

SMBus 协议

SMBus 上每个报文交互都遵从 SMBus 协议中预定义的格式。SMBus 是 I2C 规范中数据传输格式的子集。只要 I2C 设备可通过 SMBus 协议之一进行访问，便视为兼容 SMBus 规范。不符合这些协议的 I2C 设备，将无法被 SMBus 和 ACPI 规范所定义的标准方法访问。

地址解析协议

SMBus采用了I2C硬件以及I2C的硬件寻址方式，但在I2C的基础上增加了二级软件处理，建立自己独特的系统。比较特别的是SMBus规范包含一个地址解析协议，可用于实现动态地址分配。动态识别硬件和软件使得总线设备能够支持热插拔，无需重启系统便能即插即用。总线中的设备将被自动识别并分配唯一地址。这个优点非常有利于实现即插即用的用户界面。协议中有个非常特别之处在于：系统的主机和所有其它设备能够有定义其名称和功能。

超时特性

SMBus有一种超时特性：假如某个通信耗时太久，便会自动复位设备。这就解释了为什么最小时钟周期为10kHz——为了防止长时间锁死总线。I2C在本质上可以视为一个“直流”总线，也就是说当主机正在访问从机的时候，假如从机正在执行一些子程序无法及时响应，从机可以拉住主机的时钟。这样便可以提醒主机：从机正忙，但并不想放弃当前的通信。从机的当前任务结束后，将可以继续I2C会话。I2C总线协议中并没有限制这个延时的上限，但在SMBus系统中，这个时间被限定为35ms。按照SMBus协议的假定，如果某个会话耗时太久，就意味着总线出了问题，此时所有设备都应当复位以消除这种（问题）状态。这样就并不允许从设备将时钟拉低太长时间。

报文错误校验

SMBus 2.0以及1.1都采用了报文错误校验（Packet Error Checking，缩写为PEC）。在这种模式中，每次会话最后都将传输PEC（报文错误码）字节。该字节是按照CRC-8校验和的方式计算的，计算范围包括整个报文，包括地址以及读/写位。所采用的多项式为 $x^8 + x^2 + x + 1$ (CRC-8-ATMHEC算法，初始化为0)。

SMBus 警报

SMBus还有一个额外的共享的中断信号，称为SMBALERT#。从机上发生事件后，可通过这个信号通知主机来访问从机。SMBus中还定义了较少见的“主机提醒协议（Host Notify Protocol）”，基于I2C 多主模式实现类似的提醒功能，但是可以传递更多数据。

SMBus 通讯流程

SMBus的通讯流程和标准I2C的流程相似。如果一个应用要使用SMBus模式，那么在程序中需要配置几个SMBus特定的寄存器、响应一些SMBus特定标志位、实现那些在SMBus手册中介绍的上层协议。

1. 在通信之前，需要设置I2C_CTL0中SMBEN=1，并且根据需求，配置SMBSEL和ARPEN的值。
2. 为了支持ARP协议（ARPEN=1），在SMBus主机模式下（SMBTYPE=1），软件需要响应标志位HSTSMB（在SMBus从机模式下，响应DEFSMB标志位），并实现ARP协议中的功能。
3. 为了支持SMBus警告模式，软件应该响应SMBALTS标志位，并实现相应的功能。

19.3.12. 状态、错误和中断

I2C有一些状态和错误标志位和中断，通过设置一些寄存器位，便可以从这些标志触发中断（详情参见[I2C寄存器](#)）。

表 19-2. 事件状态标志位

事件标志位名称	说明
SBSEND	主机发送 START 起始位
ADDSEND	地址发送和接收
ADD10SEND	10 位地址模式中地址头发送
STPDET	监测到 STOP 结束位
BTC	字节发送结束
TBE	发送时 I2C_DATA 为空
RBNE	接收时 I2C_DATA 非空

表 19-3. I2C 错误标志位

I2C 错误名称	说明
BERR	总线错误
LOSTARB	仲裁丢失
OUERR	当禁用 SCL 拉低后，发生了溢出或下溢
AERR	没有接收到应答
PECERR	CRC 值不相同
SMBTO	SMBus 模式下总线超时
SMBALT	SMBus 警报

19.4. I2C 寄存器

19.4.1. 控制寄存器 0 (I2C_CTL0)

地址偏移: 0x00

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRESET	保留	SALT	PECTRANS	POAP	ACKEN	STOP	START	DISSTRG	GCEN	PECEN	ARPEN	SMBSEL	保留	SMBEN	I2CEN

位/位域	名称	描述
15	SRESET	软件复位I2C，软件应该在I2C总线被释放时复位I2C 0: I2C未复位 1: I2C复位
14	保留	必须保持复位值

13	SALT	软件置1和清0，硬件清0。 0: 不通过SMBA发布警告 1: 通过SMBA引脚发送警告
12	PECTRANS	PEC传输 软件置1和清0，硬件在以下条件下清除此位：PEC传输完成，监测到START/STOP结束位，I2CEN=0。 0: 不传输PEC值 1: 传输的PEC值
11	POAP	ACK/PEC的位置含义 软件置1和清0，当I2CEN=0时，硬件清0。 0: ACKEN位决定对目前正在接收的字节是否发送ACK；PECTRANS位表明PEC是否处于移位寄存器中 1: ACKEN位决定是否对下一个字节发送ACK，PECTRANS位表明下一个即将被接收的字节是PEC
10	ACKEN	是否发送ACK 软件置1和清0，当I2CEN=0时硬件清0。 0: 不发送ACK 1: 发送ACK
9	STOP	I2C总线上产生一个STOP结束位 软件置1和清0，SMBus超时时，硬件置1，监测到STOP结束位时，硬件清0。 0: 不发送STOP 1: 发送STOP
8	START	I2C总线上产生一个START起始位 软件置1和清0，当监测到START起始位或I2CEN=0时由硬件清0。 0: 不发送START 1: 发送START
7	DISSTRC	在从机模式下数据未就绪是否将SCL拉低 软件置1和清0。 0: 拉低SCL 1: 不拉低SCL
6	GCEN	是否响应对地址(0x00)的广播呼叫 0: 从机不响应广播呼叫 1: 从机将响应广播呼叫
5	PECEN	PEC计算开关 0: PEC计算关闭 1: PEC计算打开
4	ARPEN	SMBus下ARP协议开关 0: 关闭ARP 1: 开启ARP

3	SMBSEL	SMBus类型选择 0: 从机 1: 主机
2	保留	必须保持复位值
1	SMBEN	SMBus/I2C模式开关 0: I2C模式 1: SMBus模式
0	I2CEN	I2C外设使能 0: 禁用I2C 1: 使能I2C

19.4.2. 控制寄存器 1 (I2C_CTL1)

地址偏移: 0x04

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		DMALST	DMAON	BUFIE	EVIE	ERRIE	保留					I2CCLK[6:0]			

rw rw rw rw rw

rw

位/位域	名称	描述
15:13	保留	必须保持复位值
12	DMALST	DMA最后传输标志位 0: 下一个DMA EOT不是最后传输 1: 下一个DMA EOT是最后传输
11	DMAON	DMA模式开关 0: DMA模式关 1: DMA模式开
10	BUFIE	缓冲区中断使能 0: 当TBE = 1或RBNE = 1时无中断 1: 如果ITEVTEN=1, 当TBE = 1或RBNE = 1时产生中断
9	EVIE	事件中断使能 0: 禁用事件中断 1: 使能事件中断, 意味着当SBSEND、ADDSEND、ADD10SEND、STPDET或BTC标志位有效或当BIE=1且TBE=1或RBNE=1时产生中断。
8	ERRIE	错误中断使能 0: 禁用错误中断 1: 使能错误中断, 意味着当BE、LOSTARB、AE、RXORE、PECE、SMBTO或SMBALTS

标志位生效时产生中断。

7	保留	必须保持复位值
6:0	I2CCLK[6:0]	I2C外设时钟频率 I2CCLK[6:0]应该是输入APB1时钟频率，最低2MHz。 0h - 1h: 无时钟 2h - 84h: 2 MHz~84MHz 85h - 127h: 由于APB1时钟限制，无时钟

19.4.3. 从机地址寄存器 0 (I2C_SADDR0)

地址偏移: 0x08

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDFORMAT				保留		ADDRESS[9:8]		ADDRESS[7:1]		ADDRESS0					
	rw					rw				rw					rw

位/位域	名称	描述
15	ADDFORMAT	I2C从机地址模式 0: 7位地址 1: 10位地址
14:10	保留	必须保持复位值
9:8	ADDRESS[9:8]	10位地址的最高两位
7:1	ADDRESS[7:1]	7位地址或者10位地址的第7-1位
0	ADDRESS0	10位地址的第0位

19.4.4. 从机地址寄存器 1 (I2C_SADDR1)

地址偏移: 0x0C

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				保留				ADDRESS2[7:1]		DUADEN					
										rw					rw

位/位域	名称	描述
15:8	保留	必须保持复位值
7:1	ADDRESS2[7:1]	从机在双重地址模式下第二个I2C地址

0	DUADEN	双重地址模式开关
		0: 双重地址模式关
		1: 双重地址模式开

19.4.5. 传输缓冲区寄存器 (I2C_DATA)

地址偏移: 0x10

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问。

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

位/位域	名称	描述
15:8	保留	必须保持复位值
7:0	TRB[7:0]	数据发送接收缓冲区

19.4.6. 传输状态寄存器 0 (I2C_STAT0)

地址偏移: 0x14

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMBALT	SMBTO	保留	PECERR	OUERR	AERR	LOSTARB	BERR	TBE	RBNE	保留	STPDET	ADD10SEND	BTC	ADDSEND	SBSEND
rc_w0	rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	r	r		r	r	r	r	r

位/位域	名称	描述
15	SMBALT	SMBus警报状态 硬件置1, 软件写0清0。 0: SMBA引脚未被拉低(从机模式)或未监测到警报(主机模式) 1: SMBA引脚被拉低(从机模式)或监测到警报(主机模式)
14	SMBTO	SMBus模式下超时信号 硬件置1, 软件写0清0。 0: 无超时错误 1: 超时事件发生(SCL被拉低达25ms)
13	保留	必须保持复位值
12	PECERR	接收数据时PEC错误 硬件置1, 软件写0清0。 0: 接收到PEC且校验正确

		1: 接收到PEC但检验错误，此时I2C将无视ACKEN位直接发送NACK
11	OUERR	<p>当禁用SCL 拉低功能后，在从机模式下发生了过载或欠载事件。在从机接收模式下，假如DATA 中的最后一字节并未被读出，并且后续字节又接收完成，就会发生过载。在从机发送模式下，假如当前字节已经发送完成，而DATA 仍然为空，就会发生欠载。</p> <p>硬件置1，软件写0清0。</p> <p>0: 无溢出和欠载错误发生</p> <p>1: 发生溢出或欠载错误</p>
10	AERR	<p>应答错误</p> <p>硬件置1，软件写0清0。</p> <p>0: 未发生应答错误</p> <p>1: 发生了应答错误</p>
9	LOSTARB	<p>主机模式下仲裁丢失</p> <p>硬件置1，软件写0清0。</p> <p>0: 无仲裁丢失</p> <p>1: 发生仲裁丢失，I2C模块返回从机模式。</p>
8	BERR	<p>总线错误，表示I2C总线上发生了预料之外的START起始位t或STOP结束位。</p> <p>硬件置1，软件写0清0。</p> <p>0: 无总线错误</p> <p>1: 发生了总线错误</p>
7	TBE	<p>发送期间I2C_DATA为空</p> <p>硬件从I2C_DATA寄存器移动一个字节到移位寄存器之后将此位置1，软件写一个字节到I2C_DATA 寄存器清除该位。如果移位寄存器和I2C_DATA寄存器都是空的，写I2C_DATA 寄存器将不会清除TBE位（详见主机/从机发送模式下的软件操作流程）</p> <p>0: I2C_DATA非空</p> <p>1: I2C_DATA空，软件可以写</p>
6	RBNE	<p>接收期间I2C_DATA非空</p> <p>硬件从移位寄存器移动一个字节到I2C_DATA寄存器之后将此位置1，读此位可以清除此位。如果BTC和RBNE都被置1，读I2C_DATA将不会清除RBNE，因为移位寄存器的字节已经被立即移到I2C_DATA.（详见主机/从机接收模式下的软件操作流程）</p> <p>0: I2C_DATA为空</p> <p>1: I2C_DATA非空，软件可以读</p>
5	保留	必须保持复位值
4	STPDET	<p>从机模式下监测到STOP结束位</p> <p>此位被硬件置1，先读I2C_STAT0然后写I2C_CTL0可以清除此位。</p> <p>0: 从机模式下未监测到STOP结束位</p> <p>1: 从机模式下监测到STOP结束位</p>
3	ADD10SEND	<p>主机模式下10位地址的地址头被发送</p> <p>该位由硬件置1，软件读I2C_STAT0和写I2C_DATA清除此位。</p> <p>0: 主机模式下未发送10位地址的地址头</p>

		1: 主机模式下发送10位地址的地址头
2	BTC	<p>字节发送结束</p> <p>接收模式下，如果一个字节已经被移位寄存器接收但是此时I2C_DATA寄存器仍然是满的；或者发送模式下，一个字节已经被移位寄存器发送但是I2C_DATA寄存器仍然是空的，如果使能了SCL拉低功能，硬件就会置起BTC标志位。</p> <p>此位由硬件置1。</p> <p>可由以下三种方式清除：</p> <ol style="list-style-type: none"> 1、软件清除：读I2C_STAT0，然后读或写I2C_DATA寄存器清除此位 2、硬件清除：发送一个STOP或START信号，或者寄存器I2C_CTL0中I2CEN=0 <p>0: 未发生BTC 1: 发生了BTC</p>
1	ADDSEND	<p>主机模式下：成功发送了地址</p> <p>从机模式下：接收到地址并且和自身的地址匹配</p> <p>此位由硬件置1，软件读I2C_STAT0寄存器和读I2C_STAT1清0。</p> <p>0: 无地址被发送或接收 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</p>
0	SBSEND	<p>主机模式下发送START起始位</p> <p>此位由硬件置1，软件读I2C_STAT0和写I2C_DATA清0。</p> <p>0: 未发送START条件 1: START条件被发送</p>

19.4.7. 传输状态寄存器 1 (I2C_STAT1)

地址偏移: 0x18

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				ECV[7:0]				DUMODF	HSTSMB	DEFSMB	RXGC	保留	TRS	I2CBSY	MASTER

位/位域	名称	描述
15:8	ECV[7:0]	当PEC使能后硬件计算出的PEC值。
7	DUMODF	<p>从机模式下双标志位表明哪个地址和双地址模式匹配</p> <p>STOP或START起始位产生后或I2CEN=0时此位由硬件清0。</p> <p>0: 地址和I2C_SADDR0匹配 1: 地址和I2C_SADDR1匹配</p>
6	HSTSMB	<p>从机模式下监测到SMBus主机地址头</p> <p>STOP或START起始位产生后或I2CEN=0时此位由硬件清0。</p> <p>0: 未监测到SMBus主机地址头 1: 监测到SMBus主机地址头</p>

5	DEFSMB	从机模式下SMBus主机地址头 STOP或START起始位产生后或I2CEN=0时此位由硬件清0。 0: SMBus设备没有缺省地址 1: 从SMBus设备接收到一个缺省的地址
4	RXGC	是否接收到广播地址(00h) STOP或START起始位产生后或I2CEN=0时此位由硬件清0。 0: 未接收到广播呼叫地址 1: 接收到广播呼叫地址(00h)
3	保留	必须保持复位值
2	TRS	I2C作发送端还是接收端 STOP或START起始位产生后或I2CEN或LOSTARB=0时此位由硬件清0。 0: 接收端 1: 发送端
1	I2CBSY	忙标志 STOP结束位后硬件清0。 0: 无I2C通讯 1: I2C正在通讯
0	MASTER	表明I2C时钟在主机模式还是从机模式的标志位 STOP或START起始位产生后或I2CEN或LOSTARB=0时此位由硬件清0。 0: 从机模式 1: 主机模式

19.4.8. 时钟配置寄存器 (I2C_CKCFG)

地址偏移: 0x1C

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAST	DTCY	保留										CLKC[11:0]			

位/位域	名称	描述
15	FAST	主机模式下I2C速度选择 0: 标准速度 1: 快速
14	DTCY	快速模式下占空比 0: $T_{low}/T_{high} = 2$ 1: $T_{low}/T_{high} = 16/9$
13:12	保留	必须保持复位值

11:0	CLKC[11:0]	主机模式下I2C时钟控制 标准速度模式下: $\lceil T_{high} = T \rceil _{low} = CLKC * T_{PCLK1}$ 如果DTCY=0, 快速模式或快速+ 模式下: $T_{high} = CLKC * T_{PCLK1}$, $T_{low} = 2 * CLKC * T_{PCLK1}$ 如果DTCY=1, 快速模式或快速+ 模式下: $T_{high} = 9 * CLKC * T_{PCLK1}$, $T_{low} = 16 * CLKC * T_{PCLK1}$ 注意: 如果 DTCY=0, 当 PCLK1 为 3 的整数倍时, 波特率会比较准确。如果 DTCY=1, 当 PCLK1 为 25 的整数倍时, 波特率会比较准确。
------	------------	--

19.4.9. 上升时间寄存器 (I2C_RT)

地址偏移: 0x20

复位值: 0x0002

该寄存器可以按半字 (16位) 或字 (32位) 访问。

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

rw

位/位域	名称	描述
15:7	保留	必须保持复位值
6:0	RISETIME[6:0]	主机模式下最大上升时间 RISETIME值应该为SCL最大上升时间加1

19.4.10. 快速+ 模式配置寄存器 (I2C_FMPCFG)

地址偏移: 0x90

复位值: 0x0002

该寄存器可以按半字 (16位) 或字 (32位) 访问。

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

rw

位/位域	名称	描述
15:1	保留	必须保持复位值
0	FMPCEN	快速+ 模式使能 当该位被置1时, I2C设备支持高达1MHz

20. 串行外设接口/I₂S音频接口（SPI/I₂S）

20.1. 简介

SPI/I₂S模块可以通过SPI协议或I₂S音频协议与外部设备进行通信。

串行外设接口（Serial Peripheral Interface，缩写为SPI）提供了基于SPI协议的数据发送和接收功能，可以工作于主机或从机模式。SPI接口支持具有硬件CRC计算和校验的全双工和单工模式。SPI1还支持SPI四线主机模式。

片上音频接口（Inter-IC Sound，缩写为I₂S）支持四种音频标准，分别是I₂S飞利浦标准，MSB对齐标准，LSB对齐标准和PCM标准。它可以在四种模式下运行，包括主机发送模式，主机接收模式，从机发送模式和从机接收模式。

20.2. 主要特性

20.2.1. SPI 主要特性

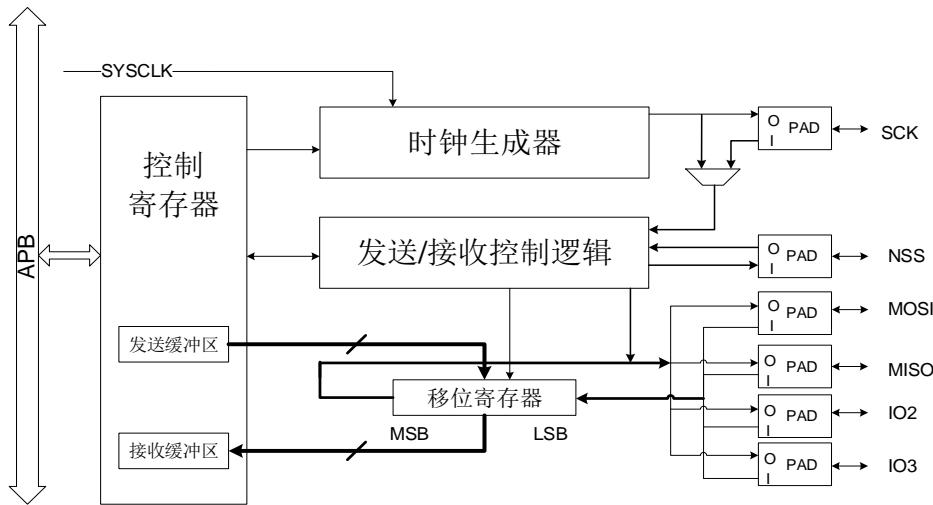
- 具有全双工和单工模式的主从操作；
- 16位宽度，独立的发送和接收缓冲区；
- 8位或16位数据帧格式；
- 低位在前或高位在前的数据位顺序；
- 软件和硬件NSS管理；
- 硬件CRC计算、发送和校验；
- 发送和接收支持DMA模式；
- 支持SPI TI模式；
- 支持SPI NSS脉冲模式
- 支持SPI四线功能的主机模式（只有SPI1）。

20.2.2. I₂S 主要特性

- 具有发送和接收功能的主从操作；
- 支持四种I₂S音频标准：飞利浦标准，MSB对齐标准，LSB对齐标准和PCM标准；
- 数据长度可以为16位，24位和32位；
- 通道长度为16位或32位；
- 16位缓冲区用于发送和接收；
- 通过I₂S时钟分频器，可以得到8 kHz到192 kHz的音频采样频率；
- 可编程空闲状态时钟极性；
- 可以输出主时钟（MCK）；
- 发送和接收支持DMA功能。

20.3. SPI 结构框图

图 20-1. SPI 结构框图



20.4. SPI 信号线描述

20.4.1. 常规配置（非 SPI 四线模式）

表 20-1. SPI 信号描述

引脚名称	方向	描述
SCK	I/O	主机: SPI 时钟输出 从机: SPI 时钟输入
MISO	I/O	主机: 数据接收线 从机: 数据发送线 主机双向线模式: 不使用 从机双向性模式: 数据发送和接收线
MOSI	I/O	主机: 数据发送线 从机: 数据接收线 主机双向线模式: 数据发送和接收线 从机双向线模式: 不使用
NSS	I/O	软件 NSS 模式: 不使用 主机硬件 NSS 模式: 为 NSS 输出, NSSDRV=1 时, 为单主机模式, NSSDRV=0 时, 为多主机模式。 从机硬件 NSS 模式: 为 NSS 输入, 作为从机的片选信号。

20.4.2. SPI 四线配置

SPI默认配置为单线模式，当SPI_QCTL中的QM0D位置1时，配置为SPI四线模式（只适用于

SPI1)。SPI四线模式只能工作在主机模式。

通过配置SPI_QCTL中的IO23_DRV位，在常规非四线SPI模式下，软件可以驱动IO2引脚和IO3引脚为高电平。

在SPI四线模式下，SPI通过以下6个引脚与外部设备连接：

表 20-2. SPI 四线信号描述

引脚名称	方向	描述
SCK	O	SPI 时钟输出
MOSI	I/O	发送或接收数据 0 线
MISO	I/O	发送或接收数据 1 线
IO2	I/O	发送或接收数据 2 线
IO3	I/O	发送或接收数据 3 线
NSS	O	NSS 输出

20.5. SPI 功能描述

20.5.1. SPI 时序和数据帧格式

SPI_CTL0寄存器中的CKPL位和CKPH位决定了SPI时钟和数据信号的时序。CKPL位决定了空闲状态时SCK的电平，CKPH位决定了第一个或第二个时钟跳变沿为有效采样边沿。在TI模式下，这两位没有意义。

图 20-2. 常规模式下的 SPI 时序图

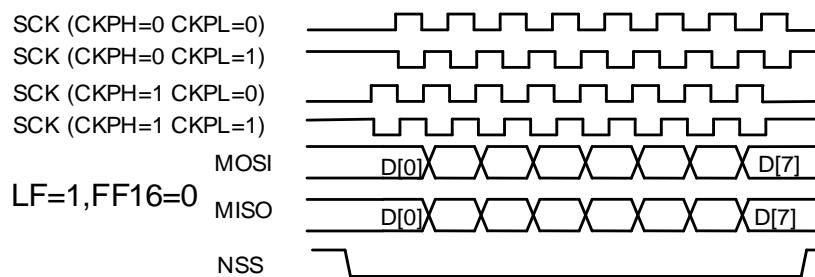
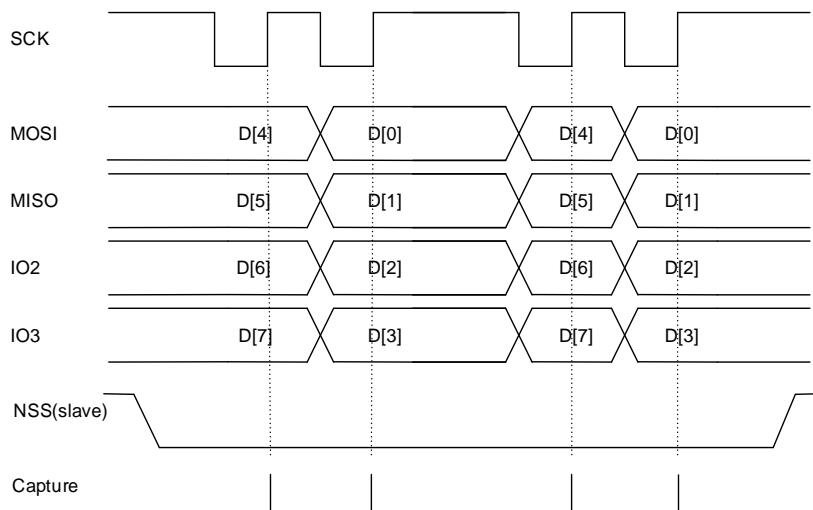


图 20-3. SPI 四线模式下的 SPI 时序图(CKPL=1, CKPH=1, LF=0)



在常规模式中，通过SPI_CTL0中的FF16位配置数据长度，当FF16=1时，数据长度为16位，否则为8位。在SPI四线模式下，数据帧长度固定为8位。

通过设置SPI_CTL0中的LF位可以配置数据顺序，当LF=1时，SPI先发送LSB位，当LF=0时，则先发送MSB位。在TI模式中，数据顺序固定为先发MSB位。

20.5.2. NSS 功能

从机模式

当配置为从机模式（MSTMOD=0）时，在硬件NSS模式（SWNSSEN = 0）下，SPI从NSS引脚获取NSS电平，在软件NSS（SWNSSEN = 1）下，SPI根据SWNNS位得到NSS电平。只有当NSS为低电平时，发送或接收数据。在软件NSS模式下，不使用NSS引脚。

主机模式

在主机模式（MSTMOD=1）下，如果应用程序使用多主机连接方式，NSS可以配置为硬件输入模式（SWNSSEN=0, NSSDRV=0）或者软件模式（SWNSSEN=1）。一旦NSS引脚（在硬件NSS模式下）或SWNNS位（在软件NSS模式下）被拉低，SPI将自动进入从机模式，并且产生主机配置错误，CONFERR位置1。

如果应用程序希望使用 NSS 引脚控制 SPI 从设备，NSS 应该配置为硬件输出模式（SWNSSEN=0, NSSDRV=1）。使能SPI之后，NSS保持高电平，当发送或接收过程开始时，NSS变为低电平。

应用程序可以使用一个通用I/O口作为NSS引脚，以实现更加灵活的NSS应用。

20.5.3. SPI 运行模式

表 20-3. SPI 运行模式

模式	描述	寄存器配置	使用的数据引脚
MFD	全双工主机模式	MSTMOD = 1 RO = 0 BDEN = 0 BDOEN: 不要求	MOSI: 发送 MISO: 接收
MTU	单向线连接主机发送模式	MSTMOD = 1 RO = 0 BDEN = 0 BDOEN: 不要求	MOSI: 发送 MISO: 不使用
MRU	单向线连接主机接收模式	MSTMOD = 1 RO = 1 BDEN = 0 BDOEN: 不要求	MOSI: 不使用 MISO: 接收
MTB	双向线连接主机发送模式	MSTMOD = 1 RO = 0 BDEN = 1 BDOEN = 1	MOSI: 发送 MISO: 不使用
MRB	双向线连接主机接收模式	MSTMOD = 1 RO = 0 BDEN = 1 BDOEN = 0	MOSI: 接收 MISO: 不使用
SFD	全双工从机模式	MSTMOD = 0 RO = 0 BDEN = 0 BDOEN: 不要求	MOSI: 接收 MISO: 发送
STU	单向线连接从机发送模式	MSTMOD = 0 RO = 0 BDEN = 0 BDOEN: 不要求	MOSI: 不使用 MISO: 发送
SRU	单向线连接从机接收模式	MSTMOD = 0 RO = 1 BDEN = 0 BDOEN: 不要求	MOSI: 接收 MISO: 不使用
STB	双向线连接从机发送模式	MSTMOD = 0 RO = 0 BDEN = 1 BDOEN = 1	MOSI: 不使用 MISO: 发送
SRB	双向线连接从机接收模式	MSTMOD = 0 RO = 0 BDEN = 1	MOSI: 不使用 MISO: 接收

模式	描述	寄存器配置	使用的数据引脚
		BDOEN = 0	

图 20-4. 典型的全双工模式连接

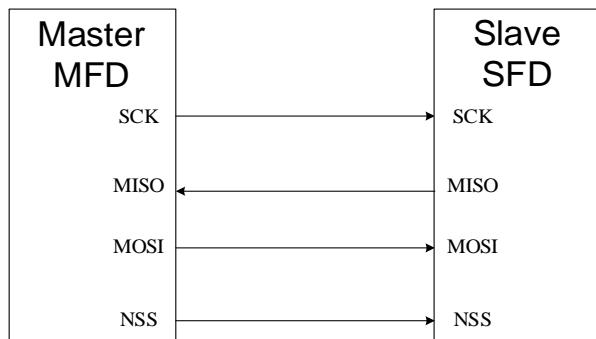


图 20-5. 典型的单工模式连接（主机：接收，从机：发送）

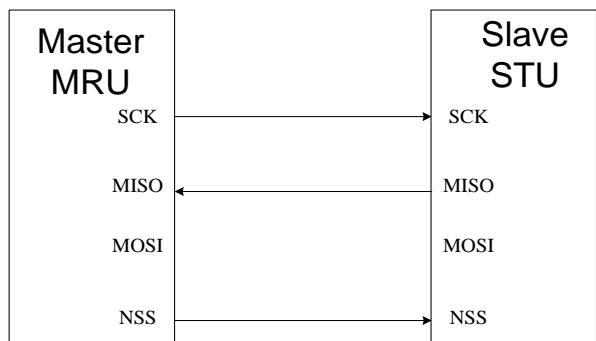


图 20-6. 典型的单工模式连接（主机：只发送，从机：接收）

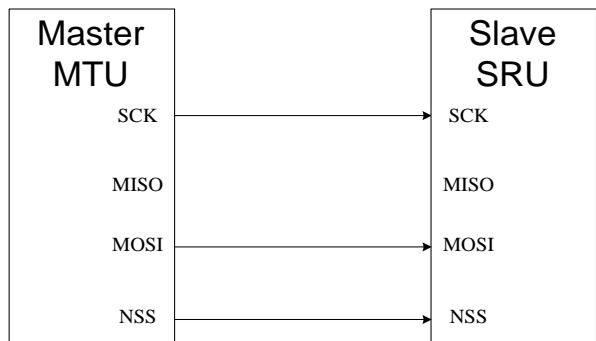
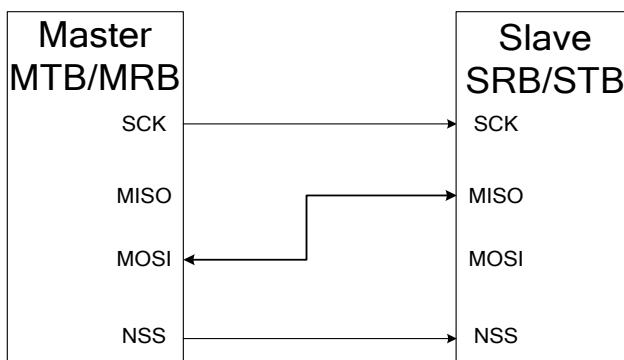


图 20-7. 典型的双向线连接



SPI 初始化流程

在发送或接收数据之前，应用程序应遵循如下的SPI初始化流程：

1. 如果工作在主机模式或从机TI模式，配置SPI_CTL0中的PSC[2:0]位来生成预期波特率的SCK信号，或配置TI模式下的Td时间。否则，忽略此步骤。
2. 配置数据格式（SPI_CTL0中的FF16位）。
3. 配置时钟时序（SPI_CTL0中的CKPL位和CKPH位）。
4. 配置帧格式（SPI_CTL0中的LF位）。
5. 按照上文[NSS功能](#)的描述，根据应用程序的需求，配置NSS模式（SPI_CTL0中的SWNSSEN位和NSSDRV位）。
6. 如果工作在TI模式，需要将SPI_CTL1中的TMOD位置1，否则，忽略此步骤。
7. 根据上面描述的运行模式，配置MSTMOD位、RO位、BDEN位和BDOEN位。
8. 如果工作在SPI四线模式，需要将SPI_QCTL中的QMOD位置1，如果不是，则忽略此步骤。
9. 使能SPI（将SPIEN位置1）。

SPI 基本发送和接收流程

发送流程

在完成初始化过程之后，SPI模块使能并保持在空闲状态。在主机模式下，当软件写一个数据到发送缓冲区时，发送过程开始。在从机模式下，当SCK引脚上的SCK信号开始翻转，且NSS引脚电平为低，发送过程开始。所以，在从机模式下，应用程序必须确保在数据发送开始前，数据已经写入发送缓冲区中。

当SPI开始发送一个数据帧时，首先将这个数据帧从数据缓冲区加载到移位寄存器中，然后开始发送加载的数据。在数据帧的第一位发送之后，TBE（发送缓冲区空）位置1。TBE标志位置1，说明发送缓冲区为空，此时如果需要发送更多数据，软件应该继续写SPI_DATA寄存器。

在主机模式下，若想要实现连续发送功能，那么在当前数据帧发送完成前，软件应该将下一个数据写入SPI_DATA寄存器中。

接收流程

在最后一个采样时钟边沿之后，接收到的数据将从移位寄存器存入到接收缓冲区，且RBNE（接收缓冲区非空）位置1。软件通过读SPI_DATA寄存器获得接收的数据，此操作会自动清除RBNE

标志位。在MRU和MRB模式中，为了接收下一个数据帧，硬件需要连续发送时钟信号，而在全双工主机模式（MFD）中，当发送缓冲区非空时，硬件只接收下一个数据帧。

SPI 不同模式下的操作流程（非 SPI 四线模式，TI 模式或 NSSP 模式）

在全双工模式下，无论是MFD模式或者SFD模式，应用程序都应该监视RBNE标志位和TBE标志位，并且遵循上文描述的操作流程。

除了忽略RBNE位和OVRE位，且只执行上述的发送流程之外，发送模式（MTU，MTB，STU和STB）与全双工模式类似。

在主机接收模式（MRU或MRB）下，全双工模式和发送模式是不同的。在MRU模式或MRB模式下，在SPI使能后，SPI产生连续的SCK信号，直到SPI失能。所以，软件应该忽略TBE标志位，并且在RBNE位置1后，读出接收缓冲区内的数据，否则，将会产生接收过载错误。

除了忽略TBE标志位，且只执行上述的接收流程之外，从机接收模式(SRU或SRB)与全双工模式类似。

SPI TI 模式

SPI TI模式将NSS作为一种特殊的帧头标志信号，它的操作流程与上文描述的常规模式类似。上文描述的模式（MFD，MTU，MRU，MTB，MRB，SFD，STU，SRU，STB和SRB）都支持TI模式。但是，在TI模式中，SPI_CTL0中的CKPL位和CKPH位是没有意义的，SCK信号的采样边沿为下降沿。

图 20-8. 主机 TI 模式在不连续发送时的时序图

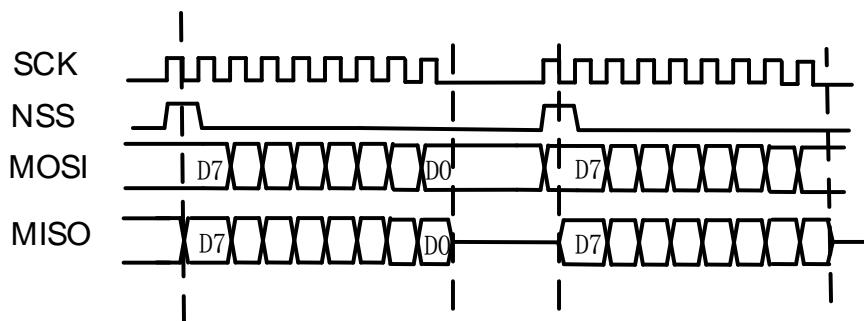
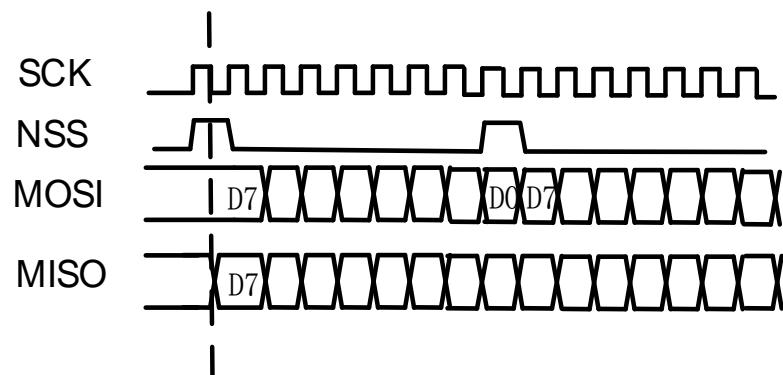
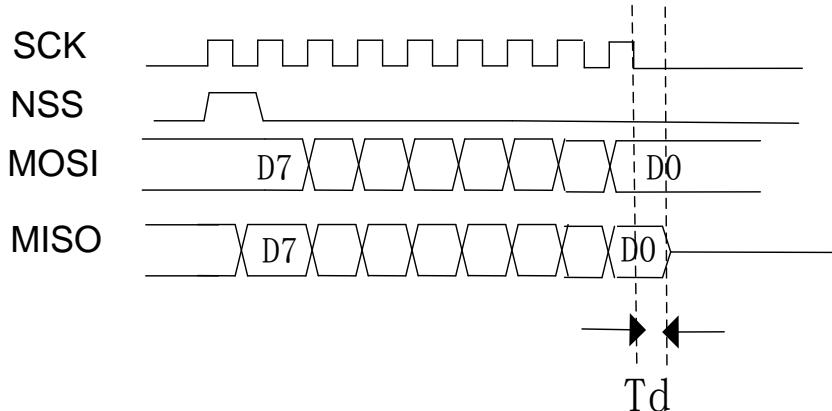


图 20-9. 主机 TI 模式在连续发送时的时序图



在主机TI模式下，**SPI**模块可实现连续传输或者不连续传输。如果主机写**SPI_DATA**的速度很快，那么就是连续传输，否则，为不连续传输。在不连续传输中，在每个字节传输前需要一个额外的时钟周期。在连续传输中，额外的时钟周期只存在于第一个字节之前，随后字节的起始时钟周期被前一个字节的最后一一位的时钟周期覆盖。

图 20-10. 从机 TI 模式时序图



在从机TI模式中，在SCK信号的最后一个上升沿，从机开始发送最后一个字节的LSB位，在半位的时间之后，主机开始采集数据。为了确保主机采集到正确的数据，在释放该引脚之前，从机需要在SCK信号的下降沿之后继续驱动该位一段时间，这段时间称为Td。Td通过SPI_CTL0寄存器中的PSC[2:0]位来设置。

$$T_d = \frac{T_{\text{bit}}}{2} + 5 * T_{\text{pclk}} \quad (20-1)$$

例如，如果PSC[2:0] = 010，那么 T_d 数值为9*Tpclk。

在从机模式下，从机需要监视NSS信号，如果检测到错误的NSS信号，将会置位FERR标志位。例如，NSS信号在一个字节的中间位发生翻转。

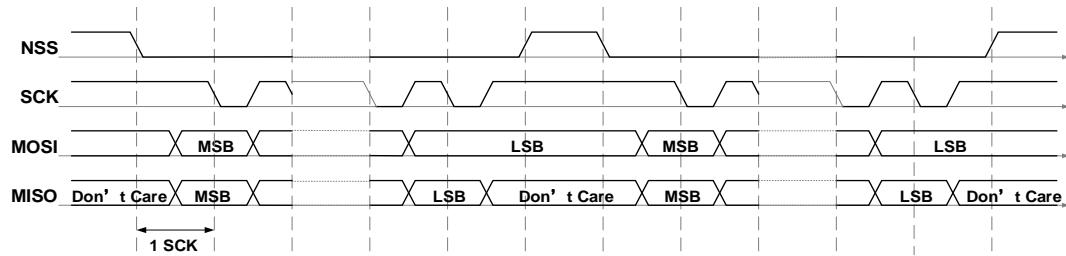
NSS 脉冲模式操作流程

配置SPI_CTL1寄存器中的NSSP位使能该功能，为了确保使用该功能实现，需满足以下几个条件：配置设备为主机模式，使用普通SPI协议的数据帧格式，同时在第一个时钟跳变沿采样数据。

总之：NSSP = 1， MSTMOD = 1， CKPH = 0。

当使用NSS脉冲模式时，根据内部数据发送缓冲区的状态，NSS脉冲会在两个连续的数据帧之间产生，且持续时间至少为1个SCK时钟周期。如果数据发送缓冲区保持为空，可能会持续多个SCK时钟周期。NSS脉冲功能专为单一的主从应用设计，支持从机锁存数据。下图描述了NSS脉冲模式在主机连续发送时的时序图。

图 20-11. NSS 脉冲模式时序图（主机连续发送）



SPI 四线模式操作流程

SPI四线模式用于控制四线SPI flash外设。

要配置成SPI四线模式，首先要确认TBE位置1，且TRANS位清零，然后将SPI_QCTL寄存器中的QMOD位置1。在SPI四线模式，SPI_CTL0寄存器中BDEN位、BDOEN位、CRCEN位、CRCNT位、FF16位、RO位和LF位保持清零，且MSTMOD位置1，以保证SPI工作于主机模式。SPIEN位、PSC位、CKPL位和CKPH位根据需要进行配置。

SPI四线模式有两种运行模式：四线写模式和四线读模式，通过SPI_QCTL寄存器中的QRD位进行配置。

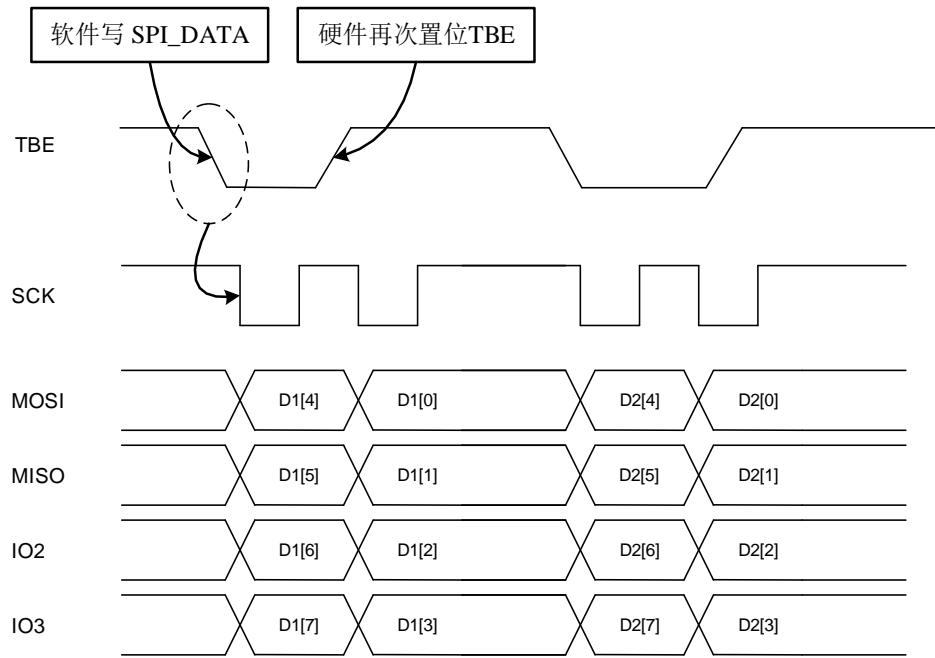
四线写模式

当SPI_QCTL寄存器中的QMOD位置1且QRD位清零时，SPI工作在四线写模式。在四线写模式中，MOSI、MISO、IO2和IO3都用作输出引脚，在SCK产生时钟信号后，一旦数据写入SPI_DATA寄存器（TBE位清零）且SPIEN位置1时，将会通过这四个引脚发送写入的数据。SPI开始数据传输之后，每发送一个数据帧都要检测TBE标志位，若不能满足条件则停止传输。

四线模式下发送操作流程：

1. 根据应用需求，配置SPI_CTL0和SPI_CTL1中的时钟预分频、时钟极性、相位等参数；
2. 将SPI_QCTL中的QMOD位置1，然后将SPI_CTL0中的SPIEN位置1来使能SPI功能；
3. 向SPI_DATA寄存器中写入一个字节的数据，TBE标志位将会清零；
4. 等待硬件将TBE位重新置位，然后写入下一个字节数据。

图 20-12. SPI 四线模式四线写操作时序图



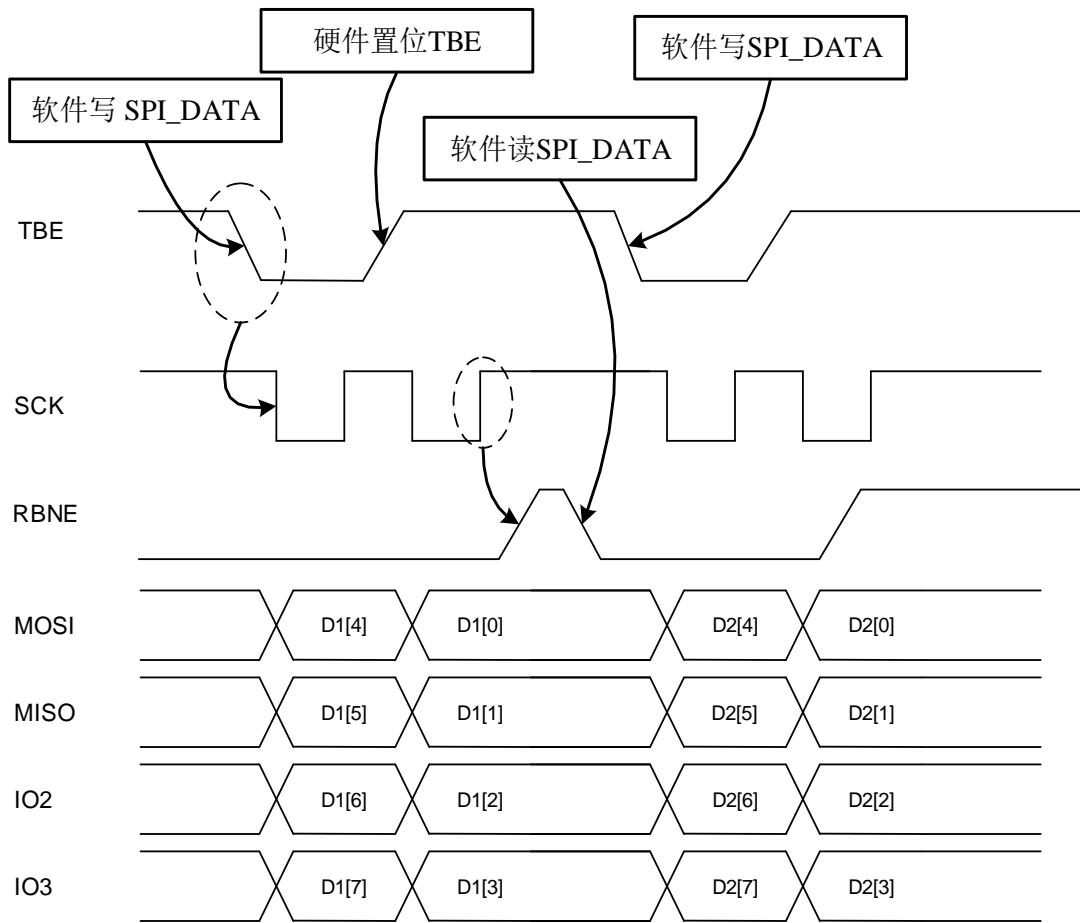
四线读模式

当SPI_QCTL寄存器中的QMOD位都置1时，SPI工作在四线读模式。在四线读模式中，MOSI、MISO、IO2和IO3都用作输入引脚，一旦数据写入SPI_DATA寄存器（TBE位清零）且SPIEN位置1时，在SCK信号线产生时钟信号。写数据到SPI_DATA寄存器只是为了产生SCK时钟信号，所以可以写入任何数据。SPI开始数据传输之后，每发送一个数据帧都要检测SPIEN位和TBE位，若条件不满足则停止传输。所以软件需要一直向SPI_DATA写空闲数据，以产生SCK时钟信号。

四线模式下接收操作流程：

1. 根据应用需求，配置SPI_CTL0和SPI_CTL1中时钟预分频、时钟极性、相位等参数；
2. 将SPI_QCTL中的QMOD位和QRD位置1，然后将SPI_CTL0中的SPIEN位置1来使能SPI功能；
3. 写任意数据（例如0xFF）到SPI_DATA寄存器；
4. 等待RBNE位置1，然后读SPI_DATA寄存器来获取接收的数据；
5. 写任意数据（例如0xFF）到SPI_DATA寄存器，以接收下一个字节数据。

图 20-13. SPI 四线模式四线读操作时序图



SPI 停止流程

不同运行模式下采用不同的流程来停止SPI功能：

MFD SFD

等待最后一个RBNE位并接收最后一个数据，等待TBE=1和TRANS=0，最后，通过清零SPIEN位关闭SPI。

MTU MTB STU STB

将最后一个数据写入SPI_DATA寄存器，等待TBE位置1，等待TRANS位清零，通过清零SPIEN位关闭SPI。

MRU MRB

等待倒数第二个RBNE位置1，从SPI_DATA寄存器读数据，等待一个SCK时钟周期，然后通过清零SPIEN位关闭SPI。等待最后一个RBNE位置1，并从SPI_DATA读数据。

SRU SRB

应用程序可以在任何时候关闭SPI功能，然后等待TRANS=0以确保当前通信过程结束。

TI模式

TI模式的停止流程与上面描述过程相同。

NSS脉冲模式

NSS脉冲模式的停止流程与上面描述过程相同。

SPI四线模式

在禁用SPI四线模式和关闭SPI功能之前，软件应该先检查：TBE位置1，TRANS位清零，SPI_QCTL中的QMOD位和SPI_CTL0中的SPIEN位清零。

20.5.4. DMA 功能

DMA功能在传输过程中将应用程序从数据读写过程中释放出来，从而提高了系统效率。

通过置位SPI_CTL1寄存器中的DMATEN位和DMAREN位，使能SPI模式的DMA功能。为了使用DMA功能，软件首先应当正确配置DMA模块，然后通过初始化流程配置SPI模块，最后使能SPI。

SPI使能后，如果DMATEN位置1，每当TBE=1时，SPI将会发出一个DMA请求，然后DMA应答该请求，并自动写数据到SPI_DATA寄存器。如果DMAREN位置1，每当RBNE=1时，发出一个DMA请求，然后DMA应答该请求，并自动从SPI_DATA寄存器读取数据。

20.5.5. CRC 功能

SPI模块包含两个CRC计算单元：分别用于发送数据和接收数据。CRC计算单元使用SPI_CRCPOLY寄存器中定义的多项式。

通过配置SPI_CTL0中的CRCEEN位使能CRC功能。对于数据线上每个发送和接收的数据，CRC单元逐位计算CRC值，计算得到的CRC值可以从SPI_TCRC寄存器和SPI_RCRC寄存器中读取。

为了传输计算得到的CRC值，应用程序需要在最后一个数据写入发送缓冲区之后，设置SPI_CTL0中的CRCNT位。在全双工模式（MFD或SFD），当SPI发送一个CRC值并且准备校验接收到的CRC值时，会将最新接收到的数据当作CRC值。在接收模式（MRB，MRU，SRU和SRB）下，在倒数第二个数据帧被接收后，软件将CRCNT位置1。在CRC校验失败时，CRCERR错误标志位将会置1。

如果使能了DMA功能，软件不需要设置CRCNT位，硬件将会自动处理CRC传输和校验。

20.6. SPI 中断

20.6.1. 状态标志位

■ 发送缓冲区空标志位(TBE)

当发送缓冲区为空时，TBE置位。软件可以通过写SPI_DATA寄存器将下一个待发送数据写入发送缓冲区。

■ 接收缓冲区非空标志位(RBNE)

当接收缓冲区非空时，RBNE置位，表示此时接收到一个数据，并已存入到接收缓冲区中，软件可以通过读SPI_DATA寄存器来读取此数据。

■ SPI 通信进行中标志位 (TRANS)

TRANS位是用来指示当前传输是否正在进行或结束的状态标志位，它由内部硬件置位和清除，无法通过软件控制。该标志位不会产生任何中断。

20.6.2. 错误标志

■ 配置错误标志(CONFERR)

在主机模式中，CONFERR位是一个错误标志位。在硬件NSS模式中，如果NSSDRV没有使能，当NSS被拉低时，CONFERR位被置1。在软件NSS模式中，当SWNSS位为0时，CONFERR位置1。当CONFERR位置1时，SPIEN位和MSTMOD位由硬件清除，SPI关闭，设备强制进入从机模式。

在CONFERR位清零之前，SPIEN位和MSTMOD位保持写保护，从机的CONFERR位不能置1。在多主机配置中，设备可以在CONFERR位置1时进入从机模式，这意味着发生了系统控制的多主冲突。

■ 接收过载错误(RXORERR)

在RBNE位为1时，如果再有数据被接收，RXORERR位将会置1。这说明，上一帧数据还未被读出而新的数据已经接收了。接收缓冲区的内容不会被新接收的数据覆盖，所以新接收的数据丢失。

■ 帧错误(FERR)

在TI从机模式下，从机也要监视NSS信号，如果检测到错误的NSS信号，将会置位FERR标志位。例如，NSS信号在一个字节的中间位发生翻转。

■ CRC错误(CRCERR)

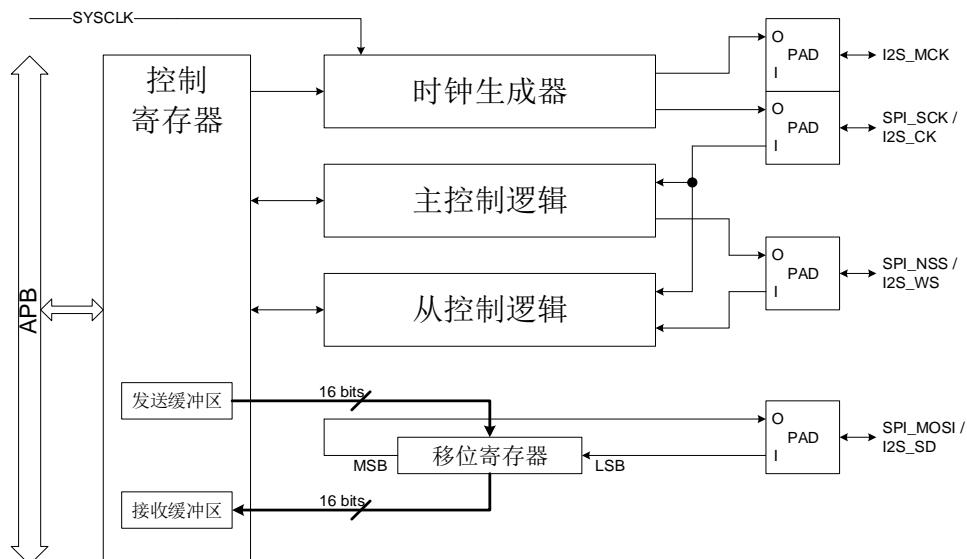
当CRCEN位置1时，SPI_RCRC寄存器中接收到的数据的CRC计算值将会和紧随着最后一帧数据接收到的CRC值进行比较，当两者不同时，CRCERR位将会置1。

表 20-4. SPI 中断请求

中断事件	描述	清除方式	中断使能位
TBE	发送缓冲区空	写SPI_DATA寄存器	TBEIE
RBNE	接收缓冲区非空	读SPI_DATA寄存器	RBNEIE
CONFERR	配置错误	读或写SPI_STAT寄存器, 然后写SPI_CTL0寄存器	ERRIE
RXORERR	接收过载错误	读SPI_DATA寄存器, 然后读SPI_STAT寄存器	
CRCERR	CRC错误	写0到CRCERR位	
FERR	TI模式帧错误	写0到FERR位	

20.7. I2S 结构框图

图 20-14. I2S 结构框图



I2S功能有5个子模块，分别是控制寄存器、时钟生成器、主机控制逻辑、从机控制逻辑和移位寄存器。所有的用户可配置寄存器都在控制寄存器模块实现，其中包括发送缓冲区和接收缓冲区。时钟生成器用来在主机模式下生成I2S通信时钟。主机控制逻辑用来在主机模式下生成I2S_WS信号并控制通信。从机控制逻辑根据接收到的I2SCK和I2S_WS信号来控制从机模式的通信。移位寄存器控制I2S_SD上的串行数据发送和接收。

20.8. I2S 信号线描述

I2S接口有4个引脚，分别是I2S_CK、I2S_WS、I2S_SD和I2S_MCK。I2S_CK是串行时钟信号，与SPI_SCK共享引脚。I2S_WS是数据帧控制信号，与SPI_NSS共享引脚。I2S_SD是串行数据信号，与SPI_MOSI共享引脚。I2S_MCK是主时钟信号，它提供了一个256倍于Fs的时钟频率，其中Fs是音频采样率。

20.9. I2S 功能描述

20.9.1. I2S 音频标准

I2S音频标准是通过设置SPI_I2SCTL寄存器中的I2SSSTD位来选择的，可以选择四种音频标准：I2S飞利浦标准，MSB对齐标准，LSB对齐标准和PCM标准。除PCM之外的所有标准都是两个通道(左通道和右通道)的音频数据分时复用I2S接口的，并通过I2S_WS信号来区分当前数据属于哪个通道。对于PCM标准，I2S_WS信号表示帧同步信息。

数据长度和通道长度可以通过SPI_I2SCTL寄存器中的DTLEN位和CHLEN位来设置。由于通道长度必须大于或等于数据长度，所以有四种数据包类型可供选择。它们分别是：16位数据打包成16位数据帧格式，16位数据打包成32位数据帧格式，24位数据打包成32位数据帧格式，32位数据打包成32位数据帧格式。用于发送和接收的数据缓冲区都是16位宽度。所以，要完成数据长度为24位或32位的数据帧传输，SPI_DATA寄存器需要被访问2次；而要完成数据长度为16位的数据帧传输，SPI_DATA寄存器只需被访问1次。如需将16位数据打包成32位数据帧，硬件会自动插入16位0将16位数据扩展为32位格式。

对于所有标准和数据包类型来说，数据的最高有效位总是最先被发送的。对于所有基于两通道分时复用的标准来说，总是先发送左通道，然后是右通道。

I2S 飞利浦标准

对于I2S飞利浦标准，I2S_WS和I2S_SD在I2S_CK的下降沿变化，各种配置情况的时序图如下所示。

图 20-15. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=0, CKPL=0)

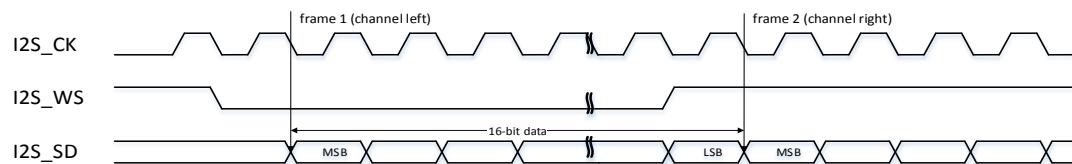
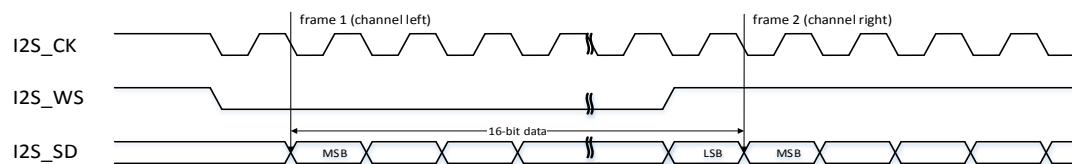


图 20-16. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=0, CKPL=1)



当 16 位数据打包成 16 位数据帧时，每完成一帧数据的传输只需要访问 SPI_DATA 寄存器一次。

图 20-17. I2S 飞利浦标准时序图(DTLEN=10, CHLEN=1, CKPL=0)

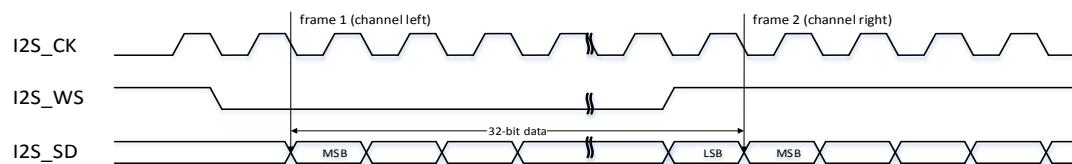
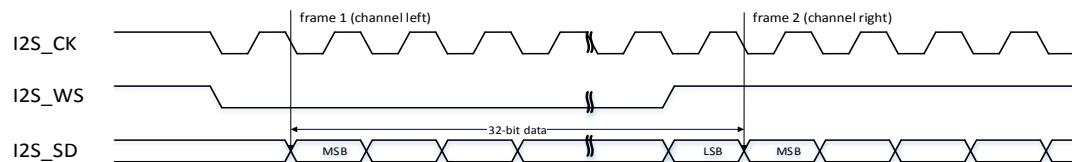


图 20-18. I2S 飞利浦标准时序图(DTLEN=10, CHLEN=1, CKPL=1)



当32位数据打包成32位数据帧的帧格式时，每完成1帧数据的传输需要访问SPI_DATA寄存器2次。在发送模式下，如果要发送一个32位数据，第一个写入SPI_DATA寄存器的数据应该是高16位数据，第二个数据应该是低16位数据。在接收模式下，如果要接收一个32位数据，第一个从SPI_DATA寄存器读到的数据应该是高16位数据，第二个数据应该是低16位数据。

图 20-19. I2S 飞利浦标准时序图(DTLEN=01, CHLEN=1, CKPL=0)

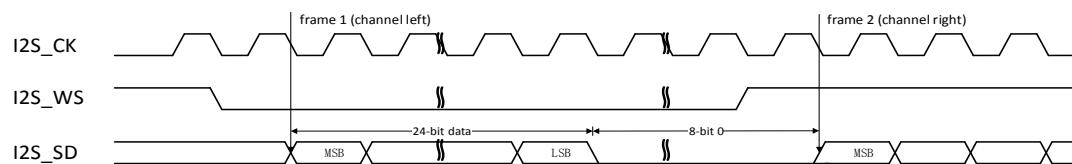
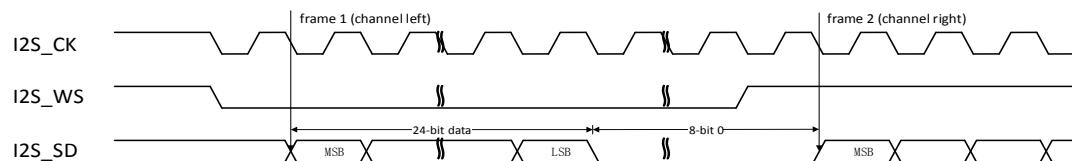


图 20-20. I2S 飞利浦标准时序图(DTLEN=01, CHLEN=1, CKPL=1)



当24位数据打包成32位数据帧的帧格式时，每完成1帧数据的传输需要访问SPI_DATA寄存器2次。在发送模式下，如果要发送一个24位数据D[23:0]，第一个写入SPI_DATA寄存器的数据应该是高16位数据D[23:8]，第二个数据应该是一个16位数据，该16位数据的高8位是D[7:0]，低8位数据可以是任意值。在接收模式下，如果要接收一个24位数据D[23:0]，第一个从SPI_DATA寄存器读到的数据应该是高16位数据D[23:8]，第二个数据应该是一个16位数据，该16位数据的高8位是D[7:0]，低8位数据全为0。

图 20-21. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=1, CKPL=0)

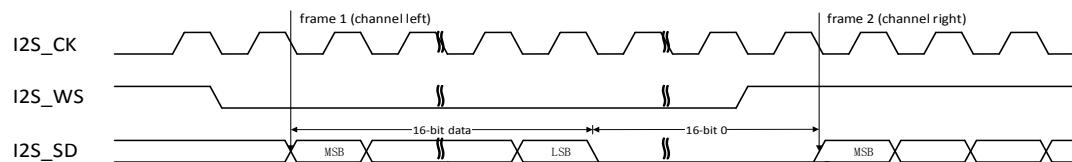
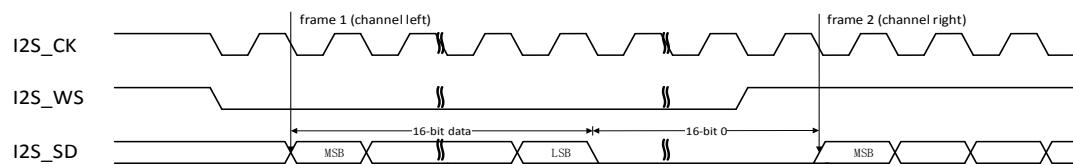


图 20-22. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=1, CKPL=1)


当16位数据打包成32位数据帧时，每完成一帧数据的传输只需要访问SPI_DATA寄存器一次。为了将该16位数据扩展成32位数据，剩下的16位被硬件强制填充为0x0000。

MSB 对齐标准

对于MSB对齐标准，I2S_WS和I2S_SD在I2S_CK的下降沿变化。SPI_DATA 寄存器的处理方式与I2S飞利浦标准完全相同。各个配置情况的时序图如下所示。

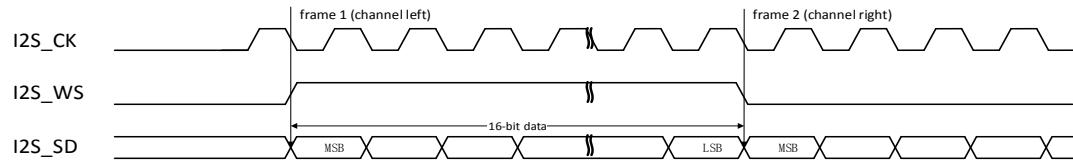
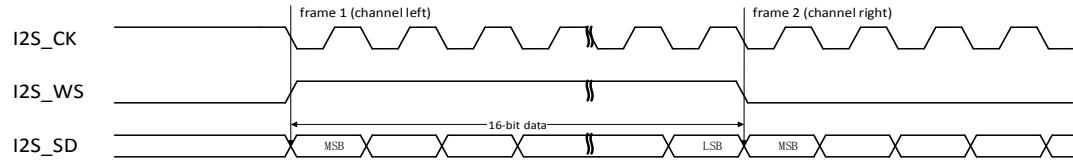
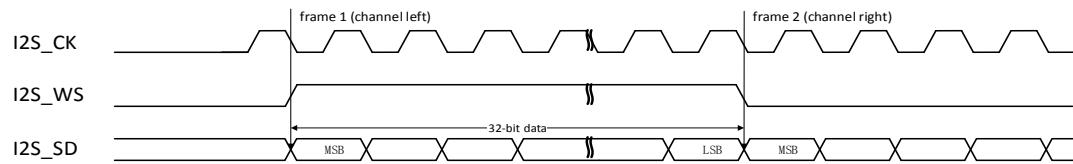
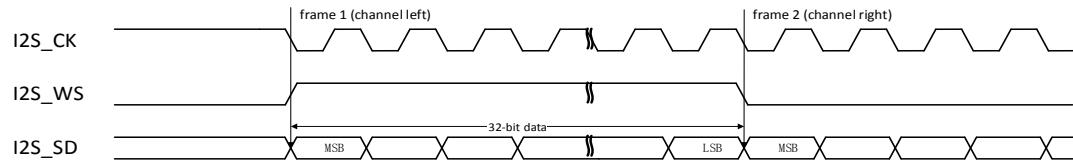
图 20-23. MSB 对齐标准时序图(DTLEN=00, CHLEN=0, CKPL=0)

图 20-24. MSB 对齐标准时序图(DTLEN=00, CHLEN=0, CKPL=1)

图 20-25. MSB 对齐标准时序图(DTLEN=10, CHLEN=1, CKPL=0)

图 20-26. MSB 对齐标准时序图(DTLEN=10, CHLEN=1, CKPL=1)


图 20-27. MSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=0)

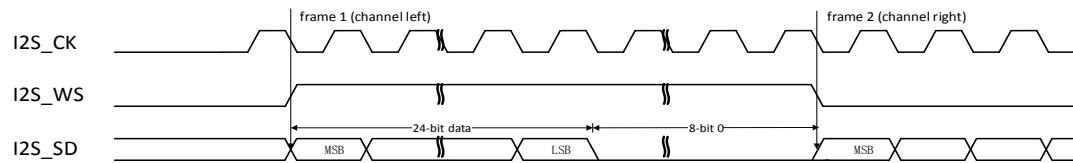


图 20-28. MSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=1)

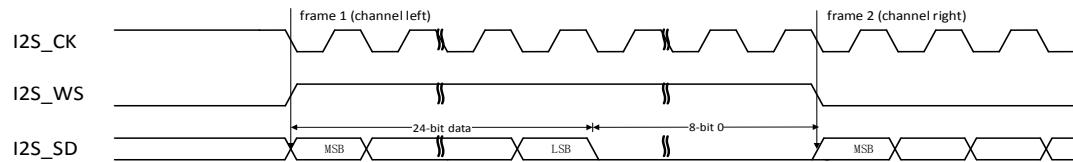


图 20-29. MSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=0)

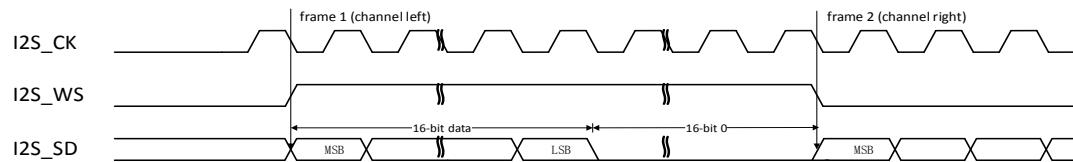
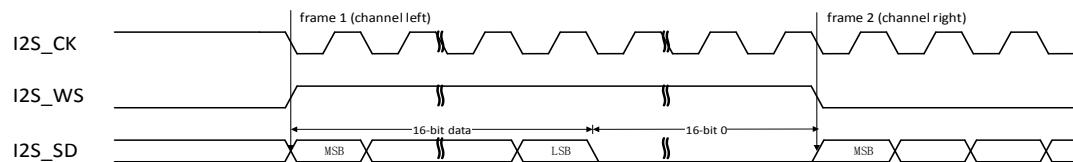


图 20-30. MSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=1)



LSB 对齐标准

对于LSB对齐标准，I2S_WS和I2S_SD在I2S_CK的下降沿变化。在通道长度与数据长度相同的情况下，LSB对齐标准和MSB对齐标准是完全相同的。对于通道长度大于数据长度的情况，LSB对齐标准的有效数据与最低位对齐，而MSB对齐标准的有效数据与最高位对齐。通道长度大于数据长度的各种配置情况时序图如下所示。

图 20-31. LSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=0)

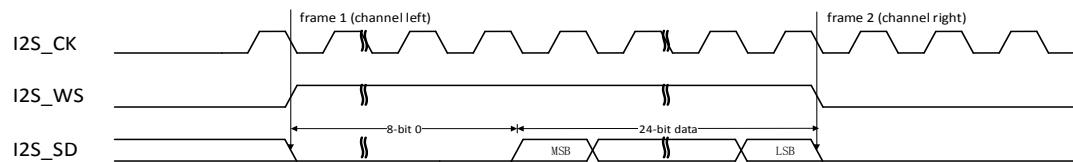
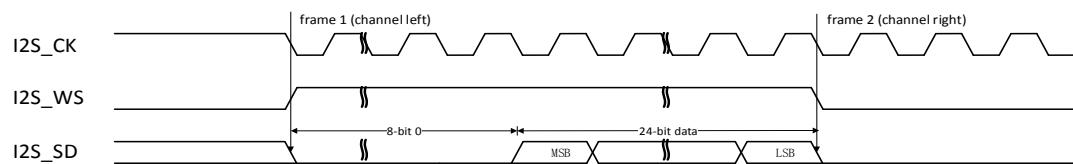


图 20-32. LSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=1)



当24位数据打包成32位数据帧的帧格式时，每完成1帧数据的传输需要访问SPI_DATA寄存器2次。在发送模式下，如果要发送一个24位数据D[23:0]，第一个写入SPI_DATA寄存器的数据应该是一个16位数据，该16位数据的高8位可以是任意值，低8位是D[23:16]，第二个数据应该是低16位数据D[15:0]。在接收模式下，如果要接收一个24位数据D[23:0]，第一个从SPI_DATA寄存器读到的数据应该是一个16位数据，该16位数据的高8位是0，低8位是D[23:16]，第二个数据应该是低16位数据D[15:0]。

图 20-33. LSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=0)

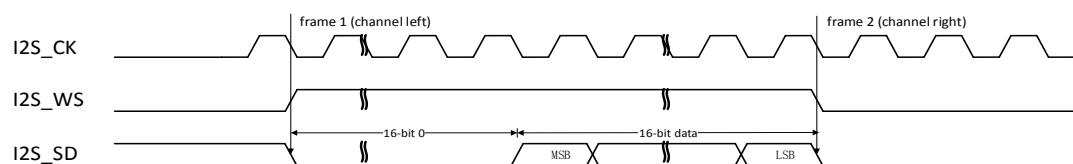
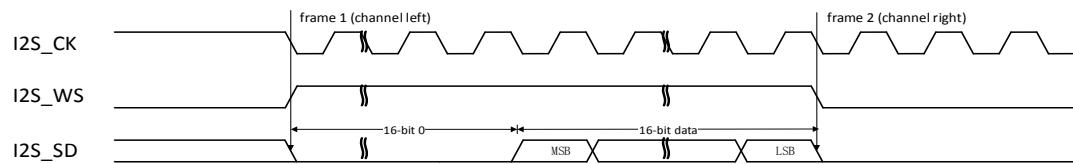


图 20-34. LSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=1)



当16位数据打包成32位数据帧时，每完成一帧数据的传输只需要访问SPI_DATA寄存器一次。为了将该16位数据扩展成32位数据，剩下的16位被硬件强制填充为0x0000。

PCM 标准

对于PCM标准，I2S_WS和I2S_SD在I2S_CK的上升沿变化，I2S_WS信号表示帧同步信息。可以通过SPI_I2SCTL寄存器的PCMSMOD位来选择短帧同步模式和长帧同步模式。SPI_DATA寄存器的处理方式与I2S飞利浦标准完全相同。短帧同步模式的各种配置情况时序图如下所示。

图 20-35. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=0)

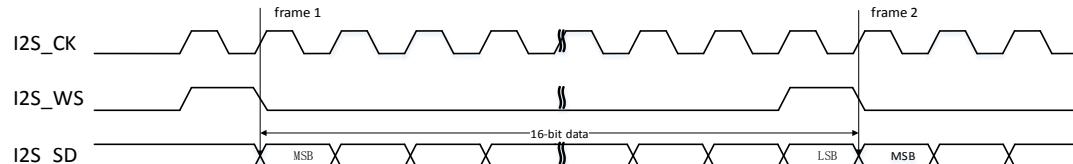


图 20-36. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=1)

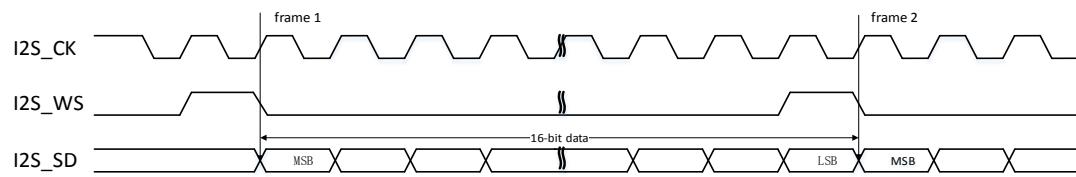


图 20-37. PCM 标准短帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=0)

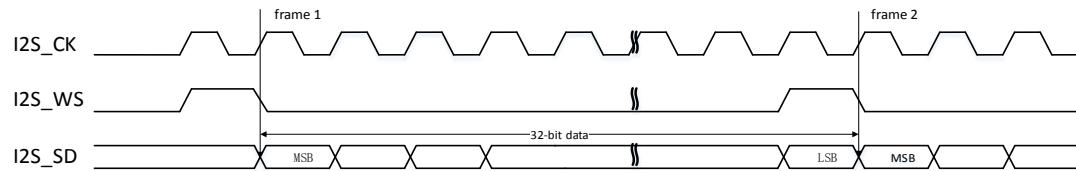


图 20-38. PCM 标准短帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=1)

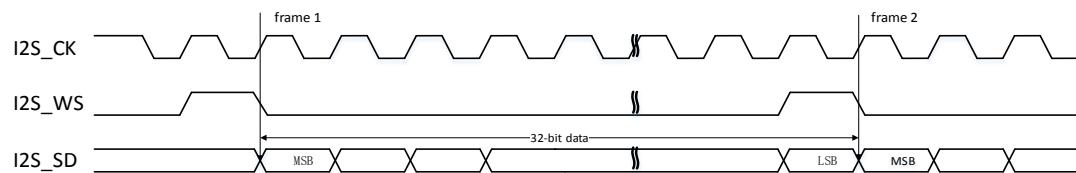


图 20-39. PCM 标准短帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=0)

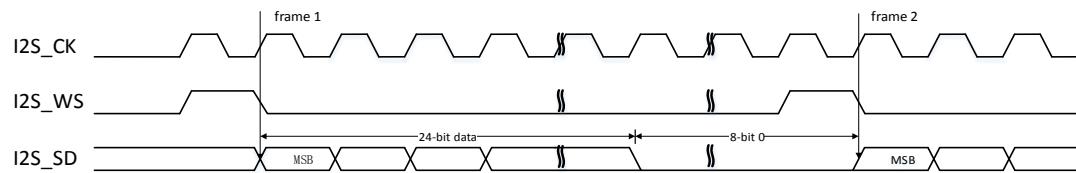


图 20-40. PCM 标准短帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=1)

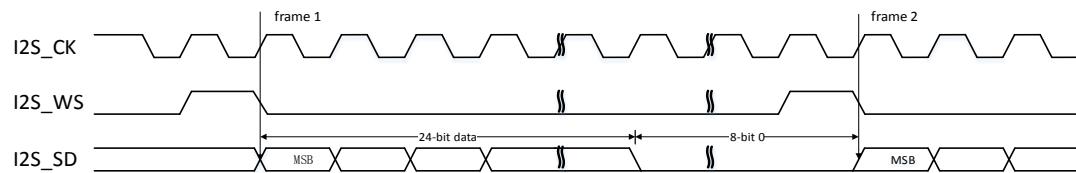


图 20-41. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=0)

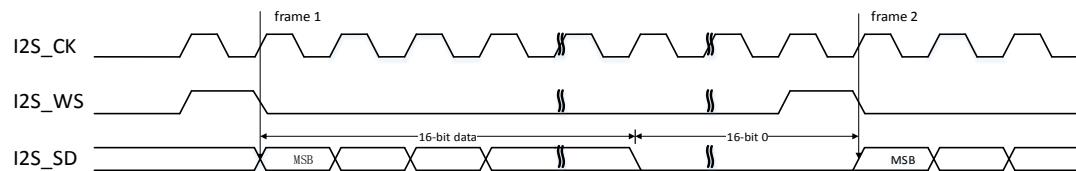
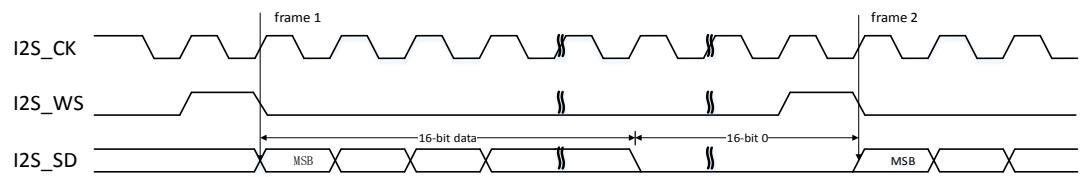


图 20-42. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=1)



长帧同步模式的各种配置情况时序图如下所示。

图 20-43. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=0)

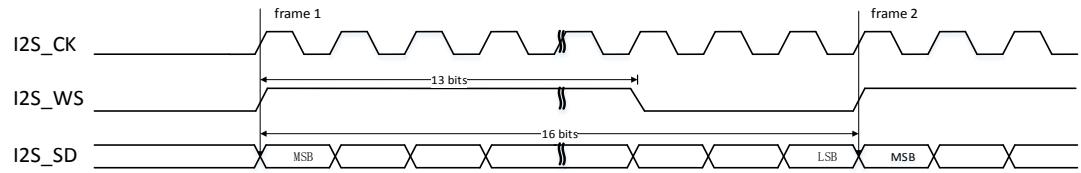


图 20-44. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=1)

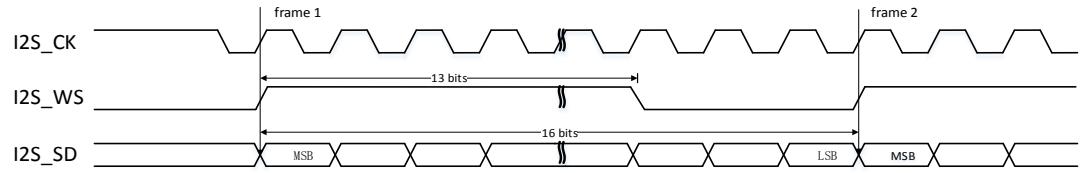


图 20-45. PCM 标准长帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=0)

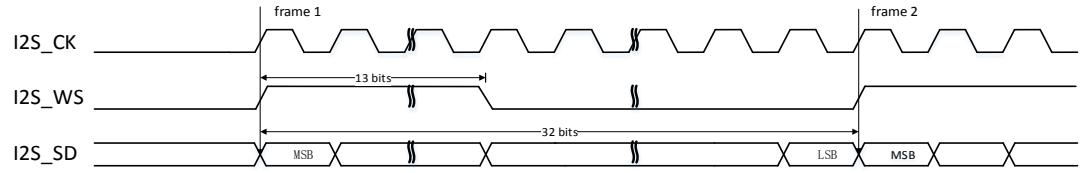


图 20-46. PCM 标准长帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=1)

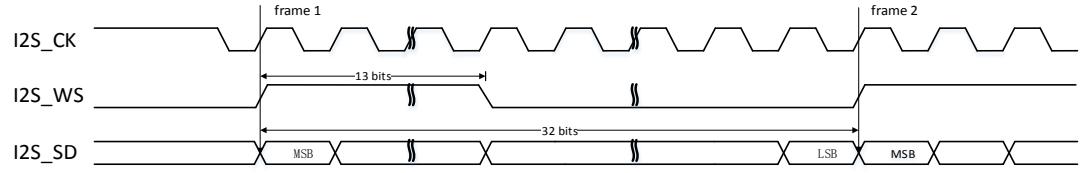


图 20-47. PCM 标准长帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=0)

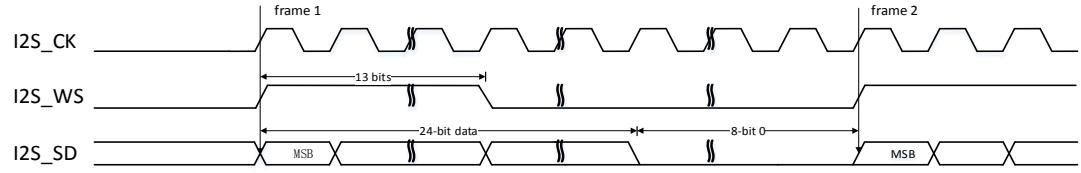


图 20-48. PCM 标准长帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=1)

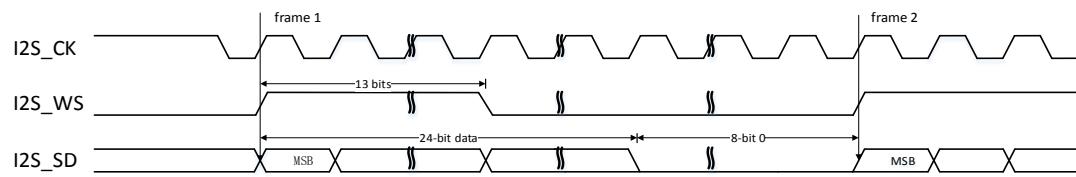


图 20-49. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=0)

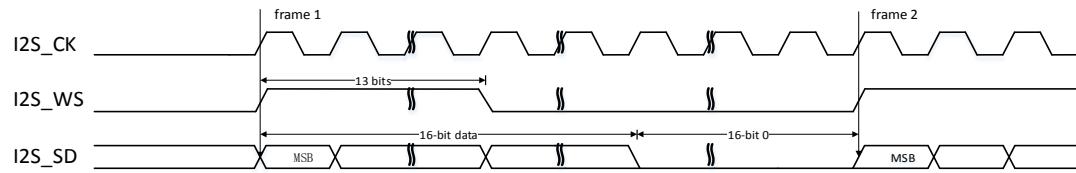
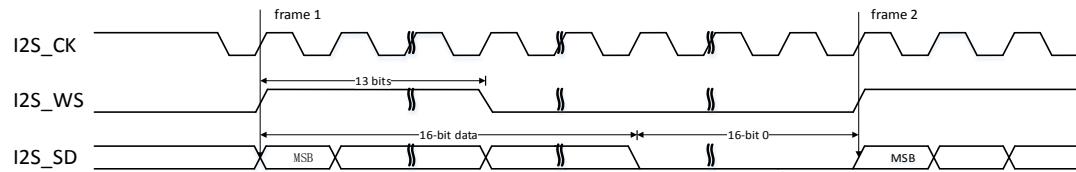
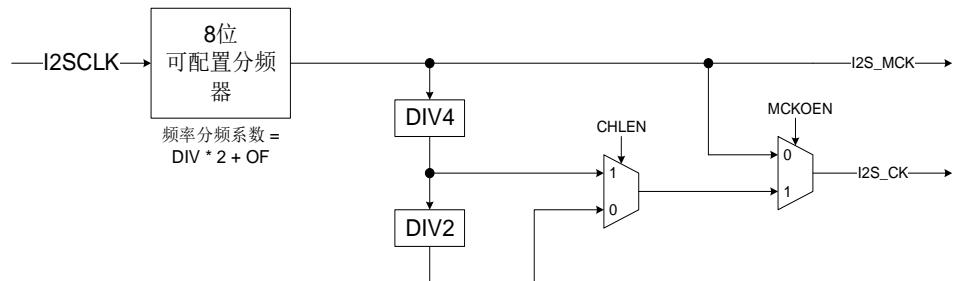


图 20-50. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=1)



20.9.2. I2S 时钟

图 20-51. I2S 时钟生成结构框图



I2S 时钟生成器框图如图 20-51. I2S 时钟生成结构框图所示。I2S 接口时钟是通过 SPI_I2SPSC 寄存器的 DIV 位, OF 位和 MCKOEN 位以及 SPI_I2SCTL 寄存器的 CHLEN 位来配置的。I2S 的时钟源是系统时钟 (CK_SYS)。I2S 比特率可以通过表 20-5. I2S 比特率计算公式所示的公式计算。

表 20-5. I2S 比特率计算公式

MCKOEN	CHLEN	公式
0	0	$I2SCLK / (DIV * 2 + OF)$
0	1	$I2SCLK / (DIV * 2 + OF)$
1	0	$I2SCLK / (8 * (DIV * 2 + OF))$
1	1	$I2SCLK / (4 * (DIV * 2 + OF))$

音频采样率(F_s)和I2S比特率的关系由如下公式定义：

$$F_s = \text{I2S比特率} / (\text{通道长度} * \text{通道数})$$

所以，为了得到期望的音频采样率，时钟生成器需要按[表 20-6. 音频采样频率计算公式](#)所列的公式进行配置。

表 20-6. 音频采样频率计算公式

MCKOEN	CHLEN	公式
0	0	I2SCLK / (32 * (DIV * 2 + OF))
0	1	I2SCLK / (64 * (DIV * 2 + OF))
1	0	I2SCLK / (256 * (DIV * 2 + OF))
1	1	I2SCLK / (256 * (DIV * 2 + OF))

20.9.3. 运行

运行模式

运行模式是通过 SPI_I2SCTL 寄存器的 I2SOPMOD 位来选择的。共有四种运行模式可供选择：主机发送模式，主机接收模式，从机发送模式和从机接收模式。各种运行模式下 I2S 接口信号的方向如[表 20-7. 各种运行模式下 I2S 接口信号的方向](#)所示。

表 20-7. 各种运行模式下 I2S 接口信号的方向

运行模式	I2S_MCK	I2S_CK	I2S_WS	I2S_SD
主机发送	输出或 NU(1)	输出	输出	输出
主机接收	输出或 NU(1)	输出	输出	输入
从机发送	输出或 NU(1)	输入	输入	输出
从机接收	输出或 NU(1)	输入	输入	输入

- NU表示该引脚没有被I2S使用，可以用于其他功能。

I2S 初始化流程

I2S初始化过程包括以下五个步骤。如果要初始化I2S工作在主机模式，五个步骤都要执行，如果要初始化I2S工作在从机模式，只需要执行步骤2、3、4、5。

- 步骤1：配置SPI_I2SPSC寄存器的DIV[7:0]位，OF位和MCKOEN位，定义I2S的比特率和选择是否需要提供I2S_MCK信号。
- 步骤2：配置SPI_I2SCTL寄存器的CKPL位，定义空闲状态的时钟极性。
- 步骤3：配置 SPI_I2SCTL 寄存器的 I2SSEL 位，I2SSTD[1:0] 位，PCMSMOD 位，I2SOPMOD[1:0] 位，DTLEN[1:0] 位和 CHLEN 位，定义 I2S 的特性。
- 步骤4：配置SPI_CTL1寄存器的TBEIE位，RBNEIE位，ERRIE位，DMATEN位和DMAREN位，选择中断源和DMA功能。此步骤可选。
- 步骤5：将SPI_I2SCTL寄存器的I2SEN位置1，来启动I2S。

I2S 主机发送流程

TBE标志位被用来控制发送流程。如前文所述，TBE标志位表示发送缓冲区空，此时，如果SPI_CTL1寄存器的TBEIE位为1，将产生中断。首先，发送缓冲区为空(TBE为1)，且移位寄存器中没有发送序列。当16位数据被写入SPI_DATA寄存器时(TBE变为0)，数据立即从发送缓冲区装载到移位寄存器中(TBE变为1)。此时，发送序列开始。

数据是并行地装载到16位移位寄存器中的，然后串行地从I2S_SD引脚发出（高位先发）。下一个数据应该在TBE为1时写入SPI_DATA寄存器。数据写入SPI_DATA寄存器之后，TBE变为0。当前发送序列结束时，发送缓冲区的数据会自动装载到移位寄存器中，然后TBE标志变回1。为了保证连续的音频数据发送，下一个将要发送的数据必须在当前发送序列结束之前写入SPI_DATA寄存器。

对于除PCM标准外的所有标准，I2SCH标志用来区别当前传输数据所属的通道。I2SCH标志在每次TBE标志由0变1的时候更新。刚开始I2SCH标志为0，表示左通道的数据应该被写入SPI_DATA寄存器。

为了关闭I2S，I2SEN位必须在TBE标志为1且TRANS标志为0之后清零。

I2S 主机接收流程

RBNE标志被用来控制接收序列。如前文所述，RBNE标志表示接收缓冲区非空，如果SPI_CTL1寄存器的RBNEIE位为1，将产生中断。当SPI_I2SCTL寄存器的I2SEN位被置1时，接收流程立即开始。首先，接收缓冲区为空(RBNE为0)。当一个接收流程结束时，接收到的数据将从移位寄存器装载到接收缓冲区(RBNE变为1)。当RBNE为1时，用户应该将数据从SPI_DATA寄存器中读走。读操作完成后，RBNE变为0。必须在下一次接收结束之前读走SPI_DATA寄存器中的数据，否则将发生接收过载错误。此时RXOVERR标志位会被置1，如果SPI_CTL1寄存器的ERRIE位为1，将会产生中断。这种情况下，必须先关闭I2S再打开I2S，然后再恢复通讯。

对于除PCM之外的所有标准来说，I2SCH标志用来区分当前传输数据所属的通道。I2SCH标志在每次RBNE标志由0变1时更新。

为了关闭I2S，不同的音频标准，数据长度和通道长度采用不同的操作步骤。每种情况的操作步骤如下所示。

- 数据长度为16位，通道长度为32位，LSB对齐标准 (DTLEN = 00, CHLEN = 1，且I2SSTD = 0b10)：
 1. 等待倒数第二个RBNE；
 2. 等待17个I2S时钟周期 (I2S_CK引脚上的时钟)；
 3. 清除I2SEN位。
- 数据长度为16位，通道长度为32位，除LSB对齐标准之外的其他标准 (DTLEN = 00, CHLEN = 1，且 I2SSTD不等于0b10)：
 1. 等待最后一个RBNE；
 2. 等待1个I2S时钟周期；
 3. 清除I2SEN位。
- 其他所有情况：

1. 等待倒数第二个RBNE;
2. 等待1个I2S时钟周期;
3. 清除I2SEN位。

I2S 从机发送流程

从机发送流程和主机发送流程相似，不同之处如下：

在从机模式下，从机需要在外部主机开始通讯之前使能。当外部主机开始发送时钟信号且I2S_WS信号请求传输数据时，发送流程开始。数据需要在外部主机发起通讯之前写入SPI_DATA寄存器。为了确保音频数据的连续传输，必须在当前发送序列结束之前将下一个待发送的数据写入SPI_DATA寄存器，否则会产生发送欠载错误。此时TXURERR标志会置1，如果SPI_CTL1寄存器的ERRIE位为1，将会产生中断。这种情况下，必须先关闭I2S再打开I2S来恢复通讯。从机模式下，I2SCH标志是根据外部主机发送的I2S_WS信号而变化的。

为关闭I2S，必须在TBE标志变为1且TRANS标志变为0之后，才能清除I2SEN位。

I2S 从机接收流程

从机接收流程与主机接收流程类似。不同之处如下。

在从机模式下，从机需要在外部主机开始通讯之前使能。当外部主机开始发送时钟信号且I2S_WS信号指示数据开始时，接收流程开始。从机模式下，I2SCH标志是根据外部主机发送的I2S_WS信号而变化的。

为了关闭I2S，必须在收到最后一个RBNE之后立即清除I2SEN位。

20.9.4. DMA 功能

DMA功能与SPI模式完全一样，唯一不同的地方就是I2S模式不支持CRC功能。

20.10. I2S 中断

20.10.1. 状态标志位

SPI_STAT寄存器中有4个可用的标志位，分别是TBE、RBNE、TRANS和I2SCH，用户通过这些标志位可以全面监视I2S总线的状态。

- **发生缓冲区空标志(TBE):**
当发送缓冲区为空时，TBE置位。软件可以通过写SPI_DATA寄存器将下一个数据写入发送缓冲区。
- **接收缓冲区非空标志(RBNE):**
接收缓冲区非空时，RBNE置位，表示此时接收到一个数据，并已存入接收缓冲区中，软件可以通过读SPI_DATA寄存器来读取此数据。
- **I2S通信进行中标志(TRANS):**
TRANS是用来指示当前传输是否正在进行或结束的状态标志，它由内部硬件置位和清除，

无法进行软件操作。该标志位不会产生任何中断。

■ I2S通道标志(I2SCH):

I2SCH用来表明当前传输数据的通道信息,对PCM音频标准来说没有意义。在发送模式下, I2SCH标志在每次TBE由0变1时更新,在接收模式下, I2SCH标志在每次RBNE由0变1时更新。该标志位不会产生任何中断。

20.10.2. 错误标志

有三个错误标志:

■ 发送欠载错误标志(TXURERR):

在从发送模式下, 有效的SCK信号开始发送, 当发送缓冲区为空时, 发送欠载错误标志TXURERR置位。

■ 接收过载错误标志(RXORERR):

当接收缓冲区已满且又接收到一个新的数据时, 接收过载错误标志RXORERR置位。当接收过载发生时, 接收缓冲区中的数据没有更新, 新接收的数据丢失。

■ 帧错误(FERR):

在从I2S模式下, I2S模块监视I2S_WS信号, 如果I2S_WS信号在一个错误的位置发生翻转, 将会置位FERR帧错误标志位。

表 20-8. I2S 中断 总结了 I2S 中断事件和相应的使能位。

表 20-8. I2S 中断

中断事件	描述	清除方式	中断使能位
TBE	发送缓冲区空	写 SPI_DATA 寄存器	TBEIE
RBNE	接收缓冲区非空	读 SPI_DATA 寄存器	RBNEIE
TXURERR	发送欠载错误	读 SPI_STAT 寄存器	ERRIE
RXORERR	接收过载错误	读 SPI_DATA 寄存器, 然后再读 SPI_STAT 寄存器	
FERR	I2S 帧错误	读 SPI_STAT 寄存器	

20.11. SPI/I2S 寄存器

20.11.1. 控制寄存器 0 (SPI_CTL0)

地址偏移: 0x00

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问。

该寄存器在I2S模式下没有意义。

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

BDEN	BDOEN	CRCEN	CRCNT	FF16	RO	SWNSS EN	SWNSS	LF	SPIEN	PSC [2:0]	MSTMOD	CKPL	CKPH
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15	BDEN	双向数据模式 0: 2线单向传输模式 1: 1线双向传输模式。数据在主机的MOSI引脚和从机的MISO引脚之间传输。
14	BDOEN	双向传输输出使能 当BDEN置位时，该位决定了数据的传输方向。 0: 工作在只接收模式 1: 工作在只发送模式
13	CRCEN	CRC计算使能 0: CRC计算禁止 1: CRC计算使能
12	CRCNT	下一次传输CRC 0: 下一次传输值为数据 1: 下一次传输值为CRC值（TCR） 当数据传输由DMA管理时，CRC值由硬件传输，该位应该被清零。 在全双工和只发送模式下，当最后一个数据写入SPI_DATA寄存器后应将该位置1。在只接收模式下，在接收完倒数第二个数据后应将该位置1。
11	FF16	数据帧格式 0: 8位数据帧格式 1: 16位数据帧格式
10	RO	只接收模式 当BDEN清零时，该位决定了数据的传输方向。 0: 全双工模式 1: 只接收模式
9	SWNSSEN	NSS软件模式选择 0: NSS硬件模式，NSS电平取决于NSS引脚 1: NSS软件模式，NSS电平取决于SWNSS位 该位在SPI TI模式下没有意义。
8	SWNSS	NSS软件模式下NSS引脚选择 0: NSS引脚拉低 1: NSS引脚拉高 只有在SWNSSEN置位时，该位有效。 该位在SPI TI模式下没有意义。

7	LF	最低有效位先发模式 0: 先发送最高有效位 1: 先发送最低有效位 该位在SPI TI模式下没有意义。
6	SPIEN	SPI使能 0: SPI设备禁止 1: SPI设备使能
5:3	PSC[2:0]	主时钟预分频选择 000: PCLK/2 100: PCLK/32 001: PCLK/4 101: PCLK/64 010: PCLK/8 110: PCLK/128 011: PCLK/16 111: PCLK/256 当使用SPI0时, PCLK=PCLK2, 当使用SPI1时, PCLK=PCLK1。
2	MSTMOD	主从模式使能 0: 从机模式 1: 主机模式
1	CKPL	时钟极性选择 0: SPI为空闲状态时, CLK引脚拉低 1: SPI为空闲状态时, CLK引脚拉高
0	CKPH	时钟相位选择 0: 在第一个时钟跳变沿采集第一个数据 1: 在第二个时钟跳变沿时钟跳变沿采集第一个数据

20.11.2. 控制寄存器 1 (SPI_CTL1)

地址偏移: 0x04

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								TBEIE	RBNEIE	ERRIE	TMOD	NSSP	NSSDRV	DMATEN	DMAREN

位/位域	名称	描述
31:8	保留	必须保持复位值
7	TBEIE	发送缓冲区空中断使能

		0: TBE中断禁止 1: TBE中断使能。当TBE置位时，产生中断。
6	RBNEIE	接收缓冲区非空中断使能 0: RBNE中断禁止。 1: RBNE中断使能。当RBNE置位时，产生中断。
5	ERRIE	错误中断使能 0: 错误中断禁止 1: 错误中断使能。当CRCERR位，CONFERR位，RXORERR位或者TXURERR位置1时，产生中断。
4	TMOD	SPI TI模式使能 0: SPI TI模式禁止 1: SPI TI模式使能
3	NSSP	SPI NSS 脉冲模式使能 0: SPI NSS 脉冲模式禁止 1: SPI NSS 脉冲模式使能
2	NSSDRV	NSS输出使能 0: NSS输出禁止 1: NSS输出使能。 当SPI使能时，如果NSS引脚配置为输出模式，NSS引脚在主模式时被拉低。如果NSS引脚配置为输入模式，NSS引脚在主模式时被拉高，此时该位无效。
1	DMATEN	发送缓冲区DMA使能 0: 发送缓冲区DMA禁止 1: 发送缓冲区DMA使能。当SPI_STAT中的TBE置位时，将会在相应的DMA通道上产生一个DMA请求。
0	DMAREN	接收缓冲区DMA使能 0: 接收缓冲区DMA禁止 1: 接收缓冲区DMA使能。当SPI_STAT中的RBNE置位时，将会在相应的DMA通道上产生一个DMA请求。

20.11.3. 状态寄存器 (SPI_STAT)

地址偏移: 0x08

复位值0x0002

该寄存器可以按半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
保留								FERR	TRANS	RXORERR	CONFERR	CRCERR	TXURERR	I2SCH	TBE	RBNE

位/位域	名称	描述
31:9	保留	必须保持复位值
8	FERR	<p>帧错误</p> <p>SPI TI模式:</p> <p>0: 没有TI模式帧错误发生</p> <p>1: TI模式帧错误发生</p> <p>I2S模式:</p> <p>0: 没有I2S帧错误发生</p> <p>1: I2S帧错误发生</p> <p>该位由硬件置位，可以通过写0清除。</p>
7	TRANS	<p>通信进行中标志</p> <p>0: SPI或I2S空闲</p> <p>1: SPI或I2S当前正在发送或接收数据</p> <p>该位由硬件置位和清除。</p>
6	RXORERR	<p>接收过载错误标志</p> <p>0: 没有接收过载错误发生</p> <p>1: 接收过载错误发生</p> <p>该位由硬件置位，软件序列清零。软件序列为：先读SPI_DATA寄存器，然后读SPI_STAT寄存器。</p>
5	CONFERR	<p>SPI配置错误</p> <p>0: 无配置错误发生</p> <p>1: 配置错误发生（主机模式下，在硬件NSS模式时NSS引脚被拉低，或者软件NSS模式时SWNSS位为0，都会产生CONFERR错误）</p> <p>该位由硬件置位，软件序列清零。软件序列为：先写SPI_STAT寄存器，然后读或写SPI_CTL0寄存器。</p> <p>I2S模式下不使用该位。</p>
4	CRCERR	<p>SPI CRC错误标志</p> <p>0: SPI_RCRC值等于最后接收到的CRC值</p> <p>1: SPI_RCRC值不等于最后接收到的CRC值</p> <p>该位由硬件置位，可以通过写0清除。</p> <p>I2S模式下不使用该位。</p>
3	TXURERR	<p>发送欠载错误标志</p> <p>0: 无发送欠载错误发生</p> <p>1: 发送欠载错误发生</p> <p>该位由硬件置位，通过写SPI_STAT寄存器清除。</p> <p>SPI模式下不使用该位。</p>
2	I2SCH	I2S通道标志

0: 下一个将要发送或接收的数据属于左通道
 1: 下一个要发送或接收的数据属于右通道
 该位由硬件置位和清除。
 SPI模式下该位无用, I2S PCM模式下该位没有意义。

1	TBE	发送缓冲区空 0: 发送缓冲区非空 1: 发送缓冲区空
0	RBNE	接收缓冲区非空 0: 接收缓冲区空 1: 接收缓冲区非空

20.11.4. 数据寄存器 (SPI_DATA)

地址偏移: 0x0C

复位值: 0x0000

该寄存器可以按半字 (16位) 或字 (32位) 访问。

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

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	SPI_DATA[15:0]	<p>数据传输寄存器值</p> <p>硬件有两个缓冲区：发送缓冲区和接收缓冲区。向SPI_DATA写数据将会把数据存入发送缓冲区，从SPI_DATA读数据，将从接收缓冲区获得数据。</p> <p>当数据帧格式为8位时，SPI_DATA[15:8]强制为0，SPI_DATA[7:0]用来发送和接收数据，发送和接收缓冲区都是8位。如果数据帧格式为16位，SPI_DATA[15:0]用于发送和接收数据，发送和接收缓冲区也是16位。</p>

20.11.5. CRC 多项式寄存器 (SPI_CRCPOLY)

地址偏移: 0x10

复位值: 0x0007

该寄存器可以按半字 (16位) 或字 (32位) 访问。

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

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

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CPR[15:0]	CRC多项式寄存器值 该值包含了CRC多项式，用于CRC计算，默认值为0007h。

20.11.6. 接收 CRC 寄存器 (SPI_RCRC)

地址偏移: 0x14

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问。

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

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	RCR[15:0]	接收CRC寄存器值 当SPI_CTL0中的CRCEN置位时，硬件计算接收数据的CRC值，并保存到RCR寄存器中。如果是8位数据帧格式，CRC计算基于CRC8标准进行，保存数据到RCR [7:0]。如果是16位数据帧格式，CRC计算基于CRC16标准进行，保存数据到RCR [15:0]。 硬件在接收到每个数据位后都会计算CRC值，当TRANS置位时，读该寄存器将返回一个中间值。 当SPI_CTL0寄存器中的CRCEN位置1或RCU复位寄存器中的SPIxRST位置1时，该寄存器复位。

20.11.7. 发送 CRC 寄存器 (SPI_TCRC)

地址偏移: 0x18

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问。

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

r

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	TCR[15:0]	<p>发送CRC寄存器值</p> <p>当SPI_CTL0中的CRCEN置位时，硬件计算发送数据的CRC值，并保存到TCR寄存器中。如果是8位数据帧格式，CRC计算基于CRC8标准进行，保存数据到TCR[7:0]。如果是16位数据帧格式，CRC计算基于CRC16标准进行，保存数据到TCR[15:0]。</p> <p>硬件在发送出每个数据位后都会计算CRC值，当TRANS置位时，读该寄存器将返回一个中间值。不同的数据帧格式（SPI_CTL0中的LF位决定）将会得到不同的CRC值。</p> <p>当SPI_CTL0寄存器中的CRCEN位置1或RCU复位寄存器中的SPIxRST位置1时，该寄存器复位。</p>

20.11.8. I2S 控制寄存器 (SPI_I2SCTL)

地址偏移: 0x1C

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16										
保留																									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0										
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 12.5%;">保留</td><td style="width: 12.5%;">I2SEL</td><td style="width: 12.5%;">I2SEN</td><td style="width: 12.5%;">I2SOPMOD[1:0]</td><td style="width: 12.5%;">PCMS MOD</td><td style="width: 12.5%;">保留</td><td style="width: 12.5%;">I2STD[1:0]</td><td style="width: 12.5%;">CKPL</td><td style="width: 12.5%;">DTLEN[1:0]</td><td style="width: 12.5%;">CHLEN</td></tr> </table>																保留	I2SEL	I2SEN	I2SOPMOD[1:0]	PCMS MOD	保留	I2STD[1:0]	CKPL	DTLEN[1:0]	CHLEN
保留	I2SEL	I2SEN	I2SOPMOD[1:0]	PCMS MOD	保留	I2STD[1:0]	CKPL	DTLEN[1:0]	CHLEN																

rw rw rw rw rw rw rw rw rw rw

位/位域	名称	描述
31:12	保留	必须保持复位值
11	I2SEL	<p>I2S模式选择</p> <p>0: SPI模式</p> <p>1: I2S模式</p> <p>当SPI模式或I2S模式关闭时配置该位。</p>
10	I2SEN	I2S使能

		0: I2S禁止 1: I2S使能 SPI模式不使用该位。
9:8	I2SOPMOD[1:0]	I2S运行模式 00: 从机发送模式 01: 从机接收模式 10: 主机发送模式 11: 主机接收模式 当I2S模式关闭时配置该位。SPI模式不使用该位。
7	PCMSMOD	PCM帧同步模式 0: 短帧同步 1: 长帧同步 只有在PCM标准下，该位才有意义。 当I2S模式关闭时配置该位。SPI模式不使用该位。
6	保留	必须保持复位值
5:4	I2SSSTD[1:0]	I2S标准选择 00: I2S飞利浦标准 01: MSB对齐标准 10: LSB对齐标准 11: PCM标准 当I2S模式关闭时配置该位。SPI模式不使用该位。
3	CKPL	空闲状态时钟极性 0: I2S_CK空闲状态为低电平 1: I2S_CK空闲状态为高电平 当I2S模式关闭时配置该位。SPI模式不使用该位。
2:1	DTLEN[1:0]	数据长度 00: 16位 01: 24位 10: 32位 11: 保留 当I2S模式关闭时配置该位。SPI模式不使用该位。
0	CHLEN	通道长度 0: 16位 1: 32位 通道长度必须大于或等于数据长度。 当I2S模式关闭时配置该位。SPI模式不使用该位。

20.11.9. I2S 时钟预分频寄存器 (SPI_I2SPSC)

地址偏移: 0x20

复位值: 0x0002

该寄存器可以按半字（16位）或字（32位）访问。

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

位/位域

名称

描述

31:10 保留 必须保持复位值

9 MCKOEN I2S_MCK输出使能

0: I2S_MCK输出禁止

1: I2S_MCK输出使能

当I2S模式关闭时配置该位。

SPI模式不使用该位。

8 OF 预分频器的奇系数

0: 实际分频系数为DIV * 2

1: 实际分频系数为DIV * 2 + 1

当I2S模式关闭时配置该位。SPI模式下不使用该位。

7:0 DIV[7:0] 预分频器的分频系数

实际分频系数是DIV * 2 + OF。

DIV不能为0.

当I2S模式关闭时配置该位。SPI模式下不使用该位。

20.11.10. SPI1 四线 SPI 控制寄存器 (SPI_QCTL)

地址偏移: 0x80

复位值: 0x0000

该寄存器可以按半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												IO23_DR V	QRD	QMOD	

位/位域	名称	描述
31:3	保留	必须保持复位值
2	IO23_DRV	IO2和IO3输出使能 0: 单线模式下IO2和IO3输出关闭 1: 单线模式下IO2和IO3输出高电平 该位仅适用于 SPI1 。
1	QRD	四线SPI模式读选择 0: SPI四线模式写操作 1: SPI四线模式读操作 该位仅能在 SPI 未通信时配置（TRANS位清零）。 该位仅适用于 SPI1 。
0	QMOD	四线SPI模式使能 0: SPI工作在单线模式 1: SPI工作在四线模式 该位仅能在 SPI 未通信时配置（TRANS位清零）。 该位仅适用于 SPI1 。

21. HDMI-CEC 控制器(HDMI-CEC)

21.1. 简介

消费电子控制（CEC）是 HDMI（高清多媒体接口）标准的一部分。CEC 作为一种协议，提供了在用户环境中各种音像制品之间的高级控制功能。CEC 协议从 HDMI-CEC 控制器获得硬件支持。

21.2. 主要特性

- HDMI-CEC 控制器符合 HDMI-CEC1.4 规范
- 32.768KHz 的 CEC 时钟具有两个时钟源可供选择：
 - 1) LXTAL 振荡器
 - 2) 固定分频后的 IRC8M 振荡器 (IRC8M/244)
- 为了超低功耗应用，CEC 控制器可工作在 Deepsleep 模式
- 可配置信号空闲时间的仲裁优先级
 - 1) 用户配置
 - 2) 控制器根据 HDMI-CEC 协议规格自动配置
- 可编程的私有地址 (OAD)
- 支持监听模式，在不干扰 CEC 总线情况下接收 CEC 总线上的数据
- 接收位宽容度功能支持更高的兼容性
- 位错误检测
 - 1) 短位错误 (BPSE)
 - 2) 长位错误 (BPLE)
 - 3) 位上升沿错误 (BRE)
- 可配置的错误位生成条件
 - 1) BPSE 检测将总会生成错误位
 - 2) BPLE 检测只有在 BPLEG=1 时会生成错误位
 - 3) BRE 检测只有在 BREG =1 时会生成错误位
- 传输错误检测 (TERR)
- 传输欠载检测 (TU)
- 接收过载检测 (RO)
- 仲裁失败检测 (ARBF)，如果 ARBF 有效，控制器会自动重传

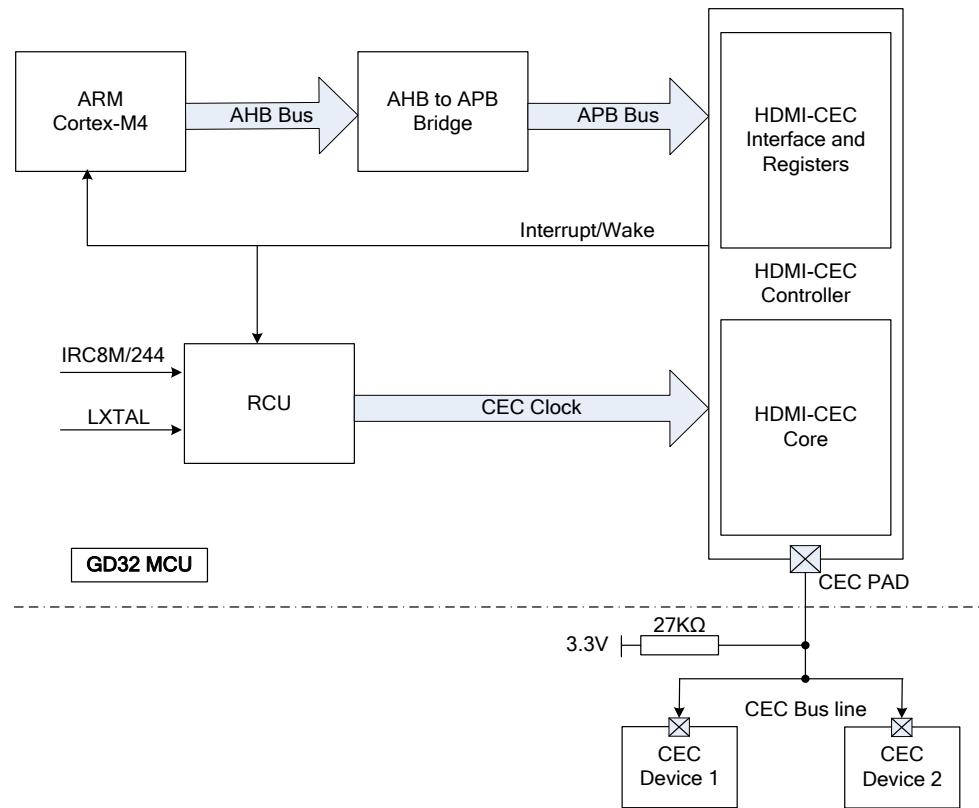
21.3. 功能描述

21.3.1. CEC 总线引脚

CEC 控制器用单根双向线来发送和接收数据。控制器的输出级，必须用漏极开路或集电极开路方式，以实现逻辑与的线路连接。CEC 引脚通过外接一个 $27\text{K}\Omega$ 的上拉电阻连接到+3.3V 电

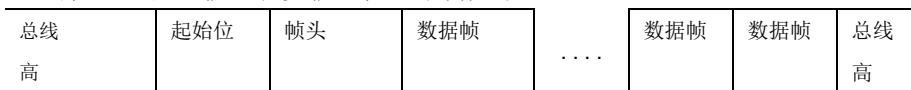
源电压。

图 21-1. HDMI-CEC 控制器框图



21.3.2. 信息说明

完整的信息包括一帧或者多帧，信息结构如下：



帧有两种类型：

- 1) **帧头**: 信息中紧接着起始位的第一帧，包含信号源的逻辑地址和信号目的地的逻辑地址。帧头必须存在。
- 2) **数据帧**: 帧结构中紧跟着帧头的部分。数据帧是可选的。

所有的帧长度都是 10 位并且有相同的基本结构如下：

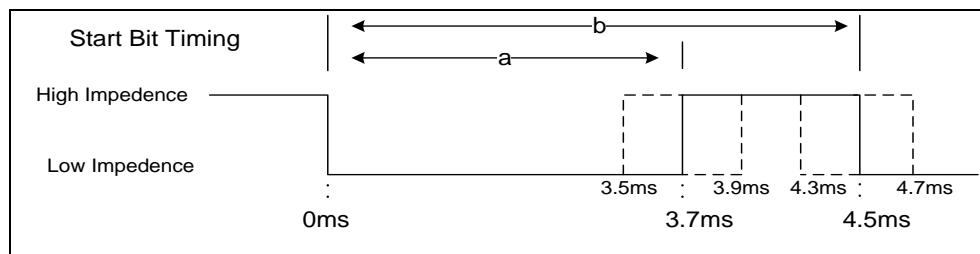
帧结构									
7	6	5	4	3	2	1	0		
信息位							ENDOM	ACK	

信息位是数据、指令还是地址取决于应用环境。控制位 ENDOM 和 ACK 在一帧中会始终出现，并且所代表的意义相同。

21.3.3. 位时序说明

信息中所有位的时序都被分成两种类型：起始位和数据位

1) 起始位：起始位通过一段低电平持续时间(a)来表示有效，其总周期(b)如下：



2) 数据位：有效数据位的时序约束如下：

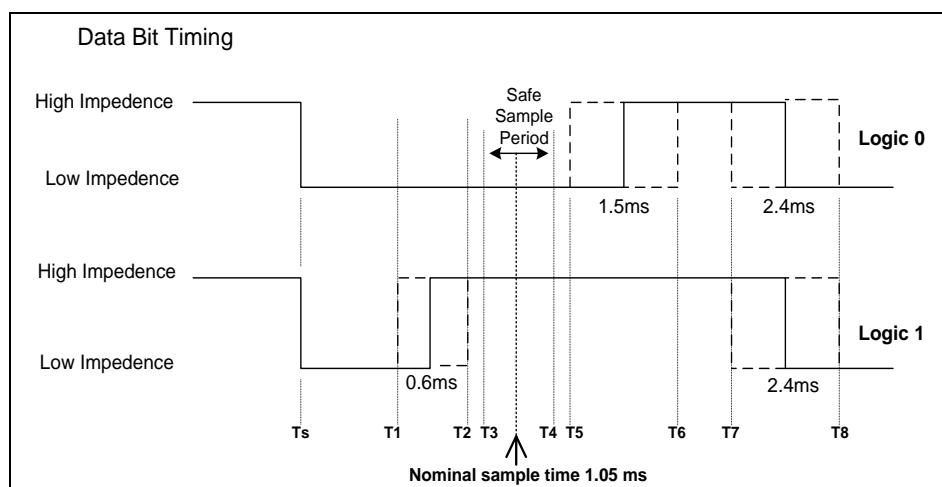


表 21-1. 数据位时序参数表

T _s	时间 (ms)	位起始
T ₁	0.4ms	传输逻辑1时, T ₁ 为从低电平到高电平过渡的最早时间
T ₂	0.8ms	传输逻辑1时, T ₂ 为从低电平到高电平过渡的最晚时间
T ₃	0.85ms	可以对信号线安全采样的最早时间
T ₄	1.25ms	可以对信号线安全采样的最晚时间
T ₅	1.3ms	T ₅ 为逻辑0时序中允许设备返回高阻态的最早时间
T ₆	1.7ms	T ₆ 为逻辑0时序中允许设备返回高阻态的最晚时间
	2.4ms	额定数据位周期总长
T ₈	2.75ms	T ₈ 为下一位开始的最晚时间

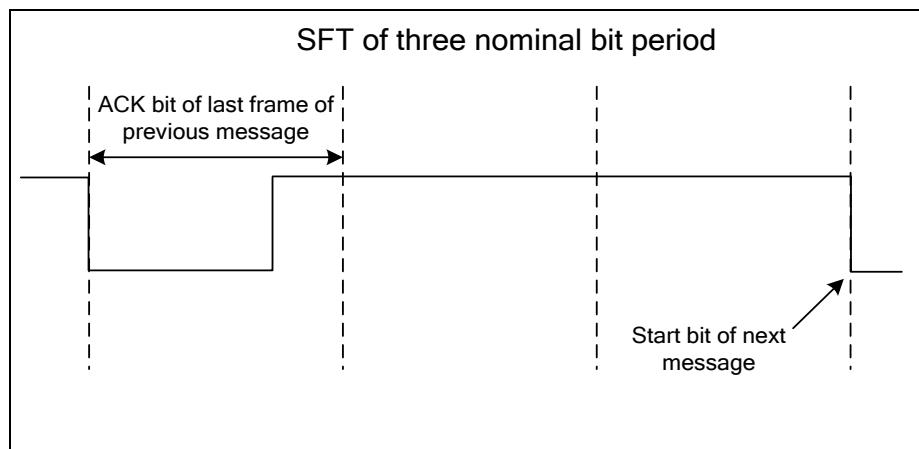
21.3.4. 仲裁

CEC 线仲裁阶段起始于起始位的前沿，结束于启动器地址位的末端。在此期间，启动器监视 CEC 线。如果此时它监测到非自身驱动的低阻态，那么就认为失去了仲裁。



在试图传输或者再次传输一帧之前，CEC 设备应该确保 CEC 总线信号已经空闲了一段时间。

这个信号空闲时间的起点为先前帧结束位的开始时刻。



需要的信号空闲时间长度取决于当前控制信号线的状态和设备的初始值。如果 $SFT=0x0$, HDMI-CEC 控制器的 SFT 将有如下表现:

应用场景	信号空闲时间（额定信号位周期）
先前的启动器在发送之前信息后想立即发送另一个信息	≥ 7
新的启动器想发送一个信息	≥ 5
先前的信息发送不成功	≥ 3

这意味着在当前设备完成发送当前信息之后，其他设备有机会在以上空闲时间内进入 CEC 线并发送自己的信息。

如果 SFT 不是 $0x0$, 会执行相应的用户所配置的 SFT 时间。

21.3.5. SFTOPT 位说明

SFT 选项位可通过设置更多的 **SFT** 计数器的启动时间点来节省总线闲置时间。

当 $SFTOPT = 0$ 时，**SOM** 位生效时，**SFT** 定时器将在控制器处于闲置状态下启动。

当 $SFTOPT = 1$ 时，**SFT** 定时器将在 CEC 总线处于空闲状态下启动。此时如果在 **SFT** 结束后配置 **SOM**, 会节省 **SFT** 时间，因为控制器启动传输的时候不再需要等待时间。

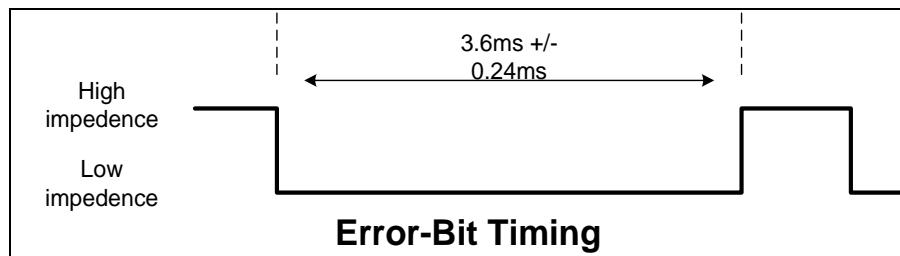
当 $SFTOPT = 1$ 时，某些事件也可能启动 **SFT** 计数器:

- 常规 TX/RX 收发结束 (**TEND/REND** 有效)
- 传输没有结束例如 **TE**, **TAE** 或 **TU** 有效
- 接收过程中，如果监测到错误且产生错误位，**SFT** 定时器在输出错误位结束后启动

21.3.6. 错误定义

错误位

如果发生错误且相应产生配置使能，HDMI-CEC 控制器将在 CEC 引脚上产生一个错误位来提示错误。错误位周期定义如下：



帧错误

CEC 协议规定信息的每一帧需要受到确定信号来确保传输成功。对于广播（目的地址=0xF）来说，ACK 位应该为逻辑 1；对于单播（目的地址<0xF）来说，ACK 位应该为逻辑 0，否则将产生帧错误（TAERR/RAE 标志位生效）。

另一个帧错误产生的原因是当 HDMI-CEC 控制器处于启动器阶段时，CEC 总线电平和 CEC 控制器输出电平不同(TERR 会置位)。

位上升错误 (BRE)

如果在 BRE 检查窗口的时候监测到上升沿，BRE 标志位会置位；如果 BREIE=1，BRE 标志位也会产生 CEC 中断。

如果 BRES=1，控制器将停止接受信息，BREG=1，错误位将产生。

如果 BRES=1，在广播时 BRE 标志位会置位，并且会产生错误位来通知启动器有错误发生。如果不产生监测 BRE 的错误位，可以配置 BREG=0 且 BCNG=1。

注：不可同时配置 BRES=0 和 BREG=1

短位错误 (BPSE)

当相邻的下降沿周期比预期短的时候，BPSE 被置 1。如果 BPSEIE=1，BPSE 标志位也会产生 CEC 中断。

如果 BPSE 错误标志置位，一定会输出错误位，除了以下情况：

- 1) BCNG = 1
- 2) LMEN = 1
- 3) 接收广播

长位错误 (BPLE)

当相邻下降沿周期比预期长时, BPLE 位被置 1。如果 BPLEIE =1, LBPE 标志位也会产生 CEC 中断。

BPLE 置位时, 如果有以下一种情况出现, 控制器将会停止接收信息并产生错误位:

- 1) 单次播放和广播模式下 BPLEG=1
- 2) 广播模式下 BCNG=0

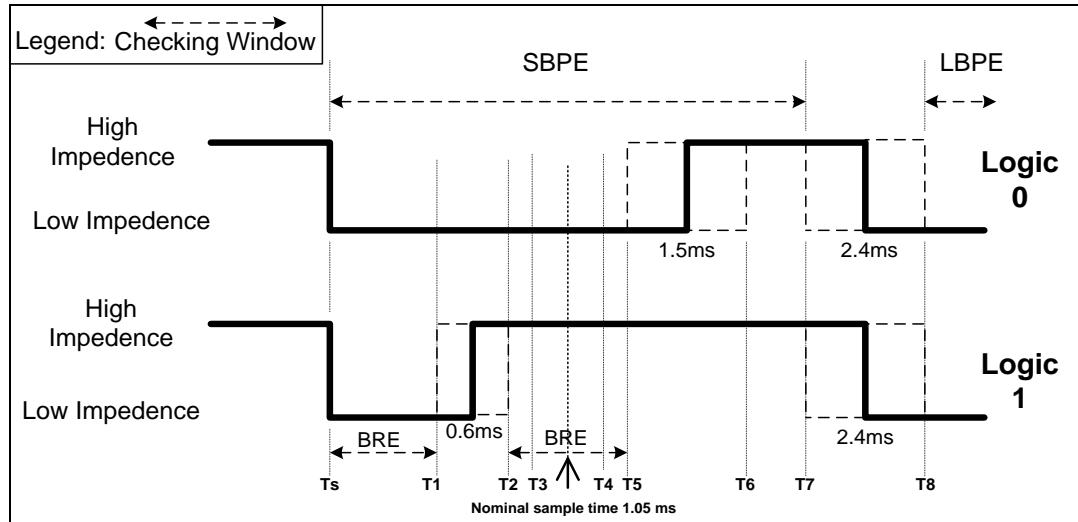


表 21-2. 错误处理时序参数表

标记	RTOL	时长(ms)	说明
Ts	-	0ms	开始传输时刻.
T1	1	0.3ms	传输逻辑1时低电平到高电平的转换最早时间
	0	0.4ms	
T2	0	0.8ms	传输逻辑1时低电平到高电平的转换最晚时间
	1	0.9ms	
T3	-	0.85ms	对信号线安全采样的最早时间
T4	-	1.25ms	对信号线安全采样的最晚时间
T5	1	1.2ms	逻辑0时允许设备返回到高阻态的最早时间
	0	1.3ms	
T6	0	1.7ms	逻辑0时允许设备返回到高阻态的最晚时间
	1	1.8ms	
T7	1	1.85ms	下一位开始传输的最早时间
	0	2.05ms	
		2.4ms	额定数据位周期
T8	0	2.75ms	下一位开始传输的最晚时间
	1	2.95ms	

传输错误监测 (TERR)

当启动器在 CEC 总线传输高组态的时候监测到总线呈低阻态时，TERR 被置 1。如果 TERRIE=1 时，TERR 也会产生 CEC 中断。

当 TERR 置位时，传输停止，软件可以重启传输。

TERR 检查窗口依赖于帧的不同位状态，显示如下：

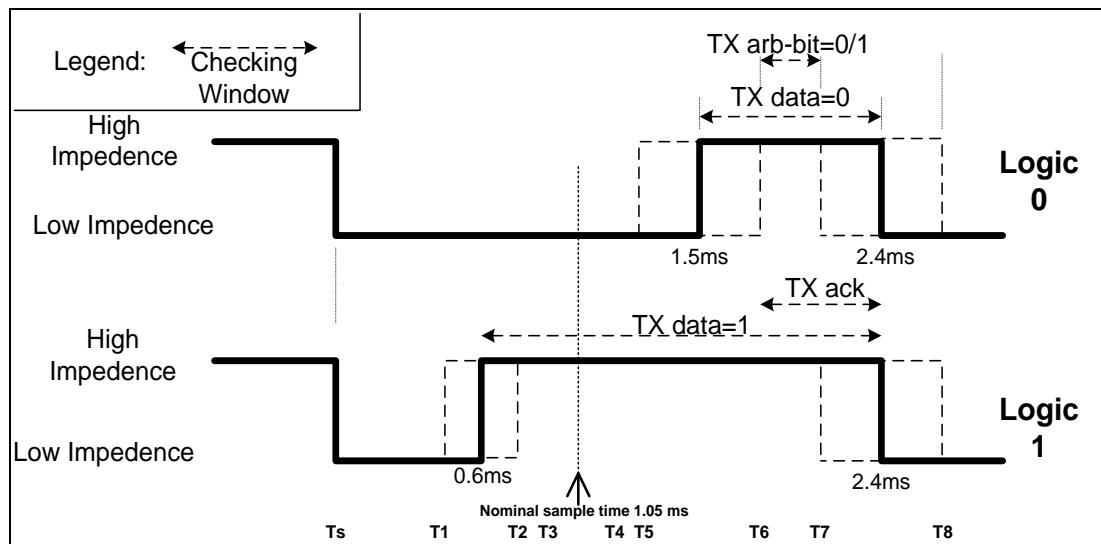


表 21-3. 时序参数表

标记	RTOL	时长(ms)	说明
Ts	-	0ms	开始传输时刻
T1	1	0.3ms	传输逻辑1时低电平到高电平的转换最早时间
	0	0.4ms	
T2	0	0.8ms	传输逻辑1时低电平到高电平的转换最晚时间
	1	0.9ms	
T3	-	0.85ms	对信号线安全采样的最早时间
T4	-	1.25ms	对信号线安全采样的最晚时间
T5	1	1.2ms	逻辑0时允许设备返回到高阻态的最早时间
	0	1.3ms	
T6	0	1.7ms	逻辑0时允许设备返回到高阻态的最晚时间
	1	1.8ms	
T7	1	1.85ms	下一位开始传输的最早时间
	0	2.05ms	
		2.4ms	额定数据位周期
T8	0	2.75ms	下一位开始传输的最晚时间
	1	2.95ms	

21.3.7. HDMI-CEC 中断

HDMI-CEC 控制器中有 13 个中断，每个中断由相应的标志位和中断使能位组成。

编号	HDMI-CEC中断事件	事件标志位	中断使能位
1	仲裁失败	ARBF	ARBFIE
2	传输字节请求	TBR	TBRIE
3	传输结束	TEND	TENDIE
4	传输字节缓冲区欠载	TU	TUIE
5	传输错误	TERR	TERRIE
6	传输确认错误	TAERR	TAERRIE
7	字节接收	BR	BRIE
8	接收结束	REND	RENDIE
9	接收过载	RO	ROIE
10	位上升沿错误	BRE	BREIE
11	短位错误	BPSE	BPSEIE
12	长位错误	BPLE	BPLEIE
13	接收位确认错误	RAE	RAEIE

注：HDMI-CEC 的任何中断都将使芯片从深度睡眠模式唤醒。

21.4. HDMI-CEC 寄存器

21.4.1. 控制寄存器 (CEC_CTL)

地址偏移: 0x00

复位值: 0x0000 0000

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

位/位域	名称	描述
31:3	保留	必须保持复位值
2	ENDOM	发送模式下下一帧的ENDOM位值 当CECEN=1时ENDOM只能被软件写入。ENDOM和STAOM一样被硬件清0。 0: 下一帧的ENDOM位发送0 1: 下一帧的ENDOM位发送1
1	STAOM	启动发送一帧信息 当CECEN=1时STAOM只能被软件写入。在TEND、TU、TAERR、TERR中的任意一位

置位或者CECEN位清零时，STAOM位被硬件清零。

如果信息只有一帧，配置TDATA之前ENDOM应该被置1。STAOM被置1以后，SFT计数器启动，当SFT计数结束，Start-bit将输出到CEC总线上。软件可以在STAOM =1的时候通过清除CECEN位来取消发送。

0: 无CEC传输

1: CEC传输挂起或者正在进行

0	CECEN	使能/禁止HDMI-CEC控制位 CECEN位由软件配置。 0: 禁止HDMI-CEC控制器，取消任何信息发送并清除ENDOM/STAOM位 1: 使能CEC控制器，如果SOM=0，进入接收状态
---	-------	--

21.4.2. 配置寄存器 (CEC_CFG)

地址偏移: 0x04

复位值: 0x0000 0000

注意：该寄存器只有在 CECEN=0 的时候才能写入

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LMEN	OAD [14:0]														
	rw														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							SFTOPT	BCNG	BPLEG	BREG	BRES	RTOL	SFT[2:0]		
							rw	rw	rw	rw	rw	rw	rw		

位/位域	名称	描述
31	LMEN	监听模式使能位 由软件置1和清0 0: 只接收OAD中的单播信息或广播信息，并返回正确ACK 1: 接收OAD的信息时给出正确ACK；接收不是OAD的信息，不发送ACK
30:16	OAD[14:0]	自身地址 OAD的每一位代表一个目的地址。例如，如果OAD[0]=1，控制器将接收发送目的地址为0x0的信息。这意味着控制器可以配置为多个私有地址。广播信息始终被接收。 接收到目的地址（帧头的后4位）以后，如果在OAD中声明了，控制器将收到正反馈信息，如果目的地址不在OAD中但LMEN=1，控制器将接收信息但不反馈ACK。如果在OAD中没有声明且LMEN=0，控制器将不会接收该信息。
15:9	保留	必须保持复位值
8	SFTOPT	SFT开始选项位 软件置1和清0 0: STAOM置1后SFT计数器开始计数 1: 发收/接收结束后SFT计数器自动启动

7	BCNG	广播信息模式下不产生错误位 软件置1和清0 0: 广播模式下, BRE和BPLE将在CEC总线上产生错误位, 当LMEN =1时BPSE也将产生错误位 1: 以上条件下不产生错误位
6	BPLEG	在单播模式下监测到BPLE的时候产生错误位 软件置1和清0 0: 在单播模式下监测到BPLE时不产生错误位 1: 在单播模式下监测到BPLE时产生错误位
5	BREG	在单次传播模式下监测到BRE的时候产生错误位 软件置1和清0 0: 在单播模式下监测到BRE的时候不产生错误位 1: 在单播模式下监测到BRE的时候产生错误位
4	BRES	监测到BRE时是否停止接收信息 软件置1和清0 0: 不停止接收BRE下的信息, 数据位按额定时间采样 (1.05ms) 1: 停止接收BRE下的信息
3	RTOL	接收位时间宽容度 软件置1和清0 0: 标准接收位时间宽容度 1: 扩展接收位时间宽容度
2:0	SFT[2:0]	信号空闲时间 软件置1和清0 如果SFT=0x0, SFT时间跟HDMI-CEC协议描述的一样, 否则, SFT时间被软件固定配置。ACK位的下降沿启动计时。 0x0: - 3个额定数据位周期, 如果SFT计数器是由于传输不成功而启动 (ARBF=1, TERR=1, TU=1或TAERR=1) - 5个额定数据位周期, 如果CEC控制器是一次新的传输 - 7个额定数据位周期, 如果CEC控制器成功完成传输 0x1: 1.5个额定数据位周期 0x2: 2.5个额定数据位周期 0x3: 3.5个额定数据位周期 0x4: 4.5个额定数据位周期 0x5: 5.5个额定数据位周期 0x6: 6.5个额定数据位周期 0x7: 7.5个额定数据位周期

21.4.3. 数据发送寄存器 (**CEC_TDATA**)

地址偏移: 0x08

复位值: 0x0000 0000

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

w

位域	名称	描述
31:8	保留	必须保持复位值
7:0	TDATA[7:0]	发送数据寄存器 包含将要传输的数据位, 只写

21.4.4. 数据接收寄存器 (CEC_RDATA)

地址偏移: 0xC

复位值: 0x0000 0000

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

r

位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	RDATA[7:0]	接收数据寄存器 只读, 该位包含从CEC线接收的最后数据字节。

21.4.5. 中断标志寄存器 (CEC_INTF)

地址偏移: 0x10

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TAERR	TERR	TU	TEND	TBR	ARBF	RAE	BPLE	BPSE	BRE	RO	REND	BR	rc_w1	rc_w1

rc_w1 rc_w1

位/位域	名称	描述
31:13	保留	必须保持复位值
12	TAERR	<p>发送ACK错误标志位 硬件置1，软件写1清0 单播模式下ACK位作为1被接收及广播模式下作为0被接收将使该标志位生效。TAERR将停止发送信息并清除STAOM和ENDOM。</p>
11	TERR	<p>发送错误 该位由硬件置1，软件写1清零 如果控制器在启动器阶段下且监测到CEC线上非控制器输出的低阻态，TERR置位。 TERR将停止发送信息且清除STAOM和ENDOM。</p>
10	TU	<p>发送数据缓冲区欠载 该位由硬件置1，软件写1清零 在发送下一字节之前如果软件没有写数据，TU生效。TU将停止发送信息并且清除STAOM和ENDOM。</p>
9	TEND	<p>发送成功结束 该位由硬件置1，软件写1清零 如果信息的所有帧成功传输，TEND位有效，TEND将清除STAOM和ENDOM位。</p>
8	TBR	<p>发送字节数据请求 该位由硬件置1，软件写1清零 当前帧的第四位传输结束后，TBR位生效，软件需要在6个额定数据位周期内写数据到txdata。</p>
7	ARBF	<p>仲裁失败 该位由硬件置1，软件写1清零 以下任一情况发生时ARBF位生效：外部CEC设备在控制器处于SFT状态时为了发送起始位拉低CEC线，或者控制器和CEC设备同时发送起始位但是控制器的启动器地址优先级更低。 如果ARBF生效，控制器将进入接收状态，接收信息结束后控制器将重新尝试发送消息。发送和接收期间，STAOM始终被置1。</p>
6	RAE	<p>接收ACK错误 该位由硬件置1，软件写1清零 以下情况RAE生效，在广播模式下如果ACK=0；或者LMEN=1且目的地址不是OAD的单播模式下，ACK=1。RAE将停止接收信息。</p>
5	BPLE	<p>长位周期错误 该位由硬件置1，软件写1清零 数据位周期超过最大时间范围的时候BPLE生效。BPLE将停止接收信息且在以下情况产生错误位：单次传播模式下BPLEG=1，或广播模式下BCNG=0。</p>
4	BPSE	<p>短位周期错误 该位由硬件置1，软件写1清零 数据位周期小于最小周期的时候BPSE位生效。</p>

3	BRE	位上升错误 该位由硬件置1，软件写1清零 期望时间外产生上升沿时，BRE生效。
2	RO	接收过载 该位由硬件置1，软件写1清零 接收新的字节且BR被置1时RO仍有效。 RO将停止接收信息并发送错误ACK位。
1	REND	接收结束 该位由硬件置1，软件写1清零 当控制器完整接收到带有成功ACK反馈的全部信息的时候REND置位。REND和BR同时生效。
0	BR	字节接收 该位由硬件置1，软件写1清零 当控制器接收到带有成功ACK反馈的数据的时候BR生效，此时RDATA有效。

21.4.6. 中断使能寄存器 (CEC_INTEN)

地址偏移: 0x14

复位值: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		TAERRIE	TERRIE	TUIE	TXENDIE	TBRIE	ARBFIG	RAEIE	BPLEIE	BPSEIE	BREIE	ROIE	RENDIE	BRIE	
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

位/位域	名称	描述
31:13	保留	必须保持复位值
12	TAERRIE	TAERR中断使能 软件置1和清0 0: TAERR中断禁止 1: TAERR中断允许
11	TERRIE	TERR中断使能 软件置1和清0 0: TERR中断禁止 1: TERR中断允许
10	TUIE	TU中断使能 软件置1和清0 0: TU中断禁止

		1: TU中断允许
9	TENDIE	TEND中断使能 软件置1和清0 0: TEND中断禁止 1: TEND中断允许
8	TBRIE	TBR中断使能 软件置1和清0 0: TBR中断禁止 1: TBR中断允许
7	ARBFIE	ARBF中断使能 软件置1和清0 0: ARBF中断禁止 1: ARBF中断允许
6	RAEIE	RAE中断使能 软件置1和清0 0: RAE中断禁止 1: RAE中断允许
5	BPLEIE	BPLE中断使能 软件置1和清0 0: BPLE中断禁止 1: BPLE中断允许
4	BPSEIE	BPSE中断使能 软件置1和清0 0: BPSE中断禁止 1: BPSE中断允许
3	BREIE	BRE中断使能 软件置1和清0 0: BRE中断禁止 1: BRE中断允许
2	ROIE	RO中断使能 软件置1和清0 0: RO中断禁止 1: RO中断允许
1	RENDIE	REND中断使能 软件置1和清0 0: REND中断禁止 1: REND中断允许
0	BRIE	BR中断使能 软件置1和清0

0: BR中断禁止

1: BR中断允许

22. 触摸传感控制器 (TSI)

22.1. 简介

触摸传感控制器 (TSI) 为触摸按键、滑块、电容近距感测等应用提供了简易的解决方案。控制器基于电荷转移方法，当一个手指接近电极时会引起整个系统的电容变化，而TSI可以通过电荷转移的方法来检测到这种变化从而感知到手指接近这一行为。

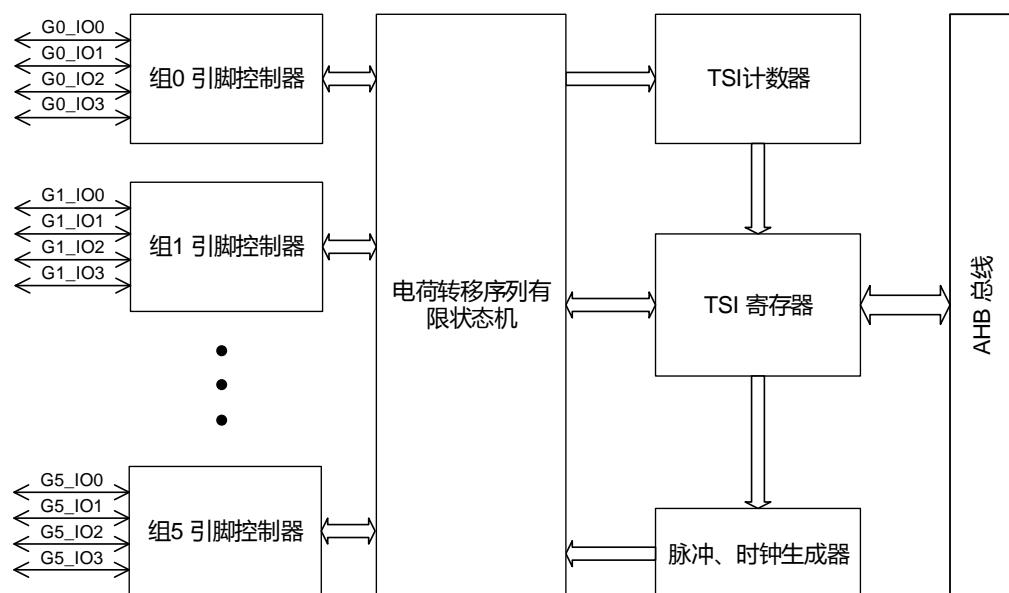
22.2. 主要特性

- 电荷转移序列完全由硬件控制；
- 包含6个完全并行的引脚组；
- 18个引脚可配置为电容感应通道引脚、6个引脚可配置为采样引脚；
- 电荷转移序列的频率可配置；
- 能够实现用户特定电荷转移序列；
- 序列结束、错误标志和可配置中断；
- 扩频功能可应用。

22.3. 功能描述

22.3.1. TSI 框图

图 22-1. TSI 模块框图

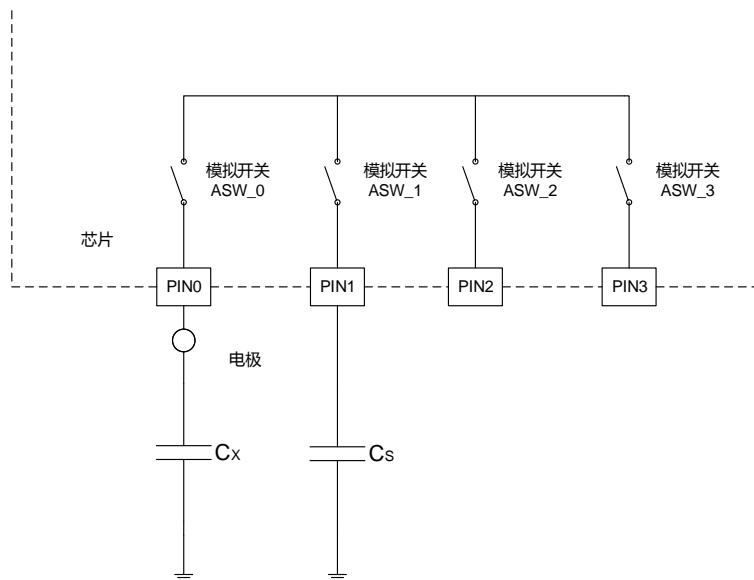


22.3.2. 触摸传感技术概述

针对触摸感知有很多不同的技术手段，例如光学、电阻式、电容式及应变式等等，其中检测系统的变化是这些技术的关键问题和目标。TSI模块使用电荷转移的方法来检测由于触摸或手指靠近带来的系统的变化，尤其是触摸电极上电容的变化。为了检测到这种变化，TSI会运行电荷转移序列，这个序列包括数次充电、转移的步骤直至满足终止条件，这一过程中的充电、转移次数就表征了系统的电容特性，因此应用程序可以通过监测这一次数值来感知系统的变化。

如[图22-2. 一个通道引脚的采样引脚的框图](#)中，由4个引脚组成了一个引脚组，4个引脚各自通过一个模拟开关连接到一个公共点上，这些模拟开关是实现电荷转移序列的关键组成部分。每一个引脚组中应该配置一个引脚为采样引脚，其他的3个引脚中应该至少配置一个为通道引脚。例如在[图22-2. 一个通道引脚的采样引脚的框图](#)中，PIN0就被配置为通道引脚，PIN1则配置为采样引脚，PIN2和PIN3没有使用。PIN0作为通道引脚，它会连接一个电极，这个电极通常设计在PCB板上，而采样引脚PIN1上需要连接一个采样电容 C_s 。现在通道引脚PIN0的电容包括 C_x 和通过电极引入的电容，所以当手指触摸电极时，PIN0的电容会增大而PIN1的电容保持不变。TSI模块会使用电荷转移序列来检测到这一变化，具体实现方法会在下一节中详细说明。

图 22-2. 一个通道引脚的采样引脚的框图



22.3.3. 电荷转移序列

TSI模块会执行电荷转移序列来测量一个通道引脚的电容变化。整个序列的详细步骤如[表22-1. 电荷转移序列的详细步骤以及引脚和开关状态](#)所示（[表22-1. 电荷转移序列的详细步骤以及引脚和开关状态](#)是按照[图22-2. 一个通道引脚的采样引脚的框图](#)来配置的，即PIN0是通道引脚，PIN1是采样引脚）。

表 22-1. 电荷转移序列的详细步骤以及引脚和开关状态

步骤序号	名称	ASW_0 状态	ASW_1 状态	PIN0 状态	PIN1 状态
1	放电	闭合	闭合	浮空输入	下拉
2	缓冲时间 1	断开	断开	浮空输入	浮空输入
3	充电	断开	断开	输出高电平	浮空输入

4	充电扩展	断开	断开	输出高电平	浮空输入
5	缓冲时间 2	断开	断开	浮空输入	浮空输入
6	电荷转移	闭合	闭合	浮空输入	浮空输入
7	缓冲时间 3	断开	断开	浮空输入	浮空输入
8	对比	断开	断开	浮空输入	浮空输入

1. 放电

闭合 ASW_0 和 ASW_1 并且将PIN1配置成下拉，使C_x和 C_s放电。这个步骤是初始化的操作，应该由软件在传输序列开始之前通过配置寄存器来实现。该步骤的放电时间需要保证C_x 和 C_s 的电压被彻底放电至0。

2. 缓冲时间 1

ASW_0 和 ASW_1 断开， PIN0 被设置成浮空输入。

3. 充电

充电过程中，通道引脚PIN0被配置成输出高电平，给 C_x充电，ASW_0 和 ASW_1仍然断开。充电时间应该配置为可确保（细节参看寄存器部分）C_x 充电到VDD。

4. 充电扩展

这是在电荷转移序列中的一个可选步骤，目的是实现扩频。此步骤中所有引脚和模拟开关的状态与步骤3完全相同，唯一差别是持续时间，该步骤的持续时间在每次循环都会变化，最大值可以TSI寄存器配置。

5. 缓冲时间 2

ASW_0 和 ASW_1 断开， PIN0 被设置成浮空输入。

6. 电荷转移

这一步骤中ASW_0和ASW_1闭合，而PIN0被设置成浮空输入，电荷将从 C_x向 C_s 转移。这个转移时间应该被合理配置（细节参看寄存器部分）以确保转移后采样引脚和通道引脚的电压值相等。

7. 缓冲时间 3

ASW_0 和ASW_1 断开， PIN0 被设置成浮空输入。

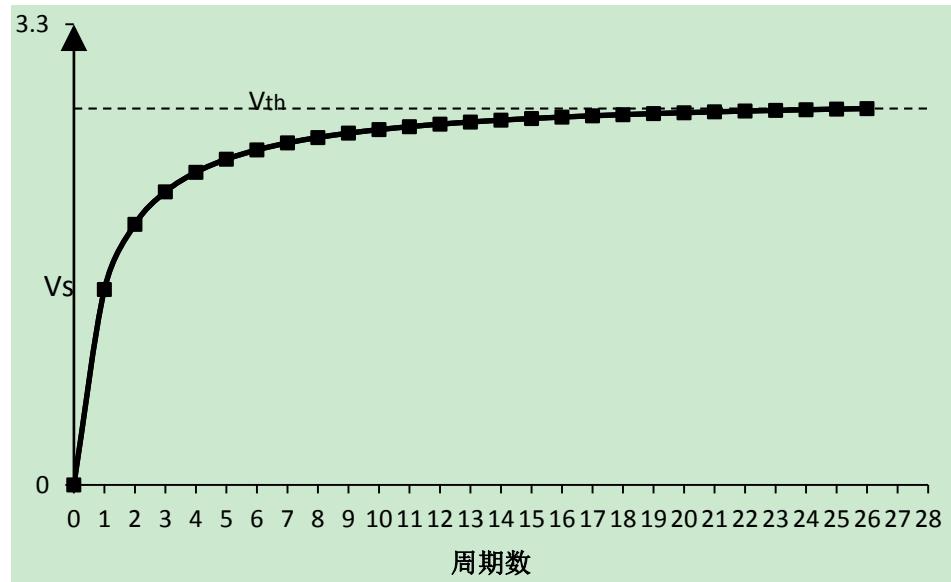
8. 对比

ASW_0、ASW_1 和 PIN0 仍然保留上一步骤的配置。在这一步将采样引脚PIN1 的电压 V_s 和一个门限电压 V_{th}进行比较。如果 V_s 比 V_{th}低，那么整个序列返回到步骤2开始新的循环，否则整个序列结束。

在初始化步骤1完成之后，采样引脚的电压 V_s 是0，之后每经过一个充电、转移周期都会增大一些，如[图22-3. 电荷序列传输期间的采样引脚的电压](#)所示。根据电气学知识，如果电极上的电容越大，则每个周期 V_s 增大的值就会越大。电荷转移序列的停止条件是 V_s 达到 V_{th}。

每个序列都有一个计数器用来记录达到 V_{th} 的周期次数。在电荷转移序列的结束，序列计数器会被读出用于评估 C_x 的电容，例如，一个比较小的计数器值代表一个比较大的 C_x 电容值。

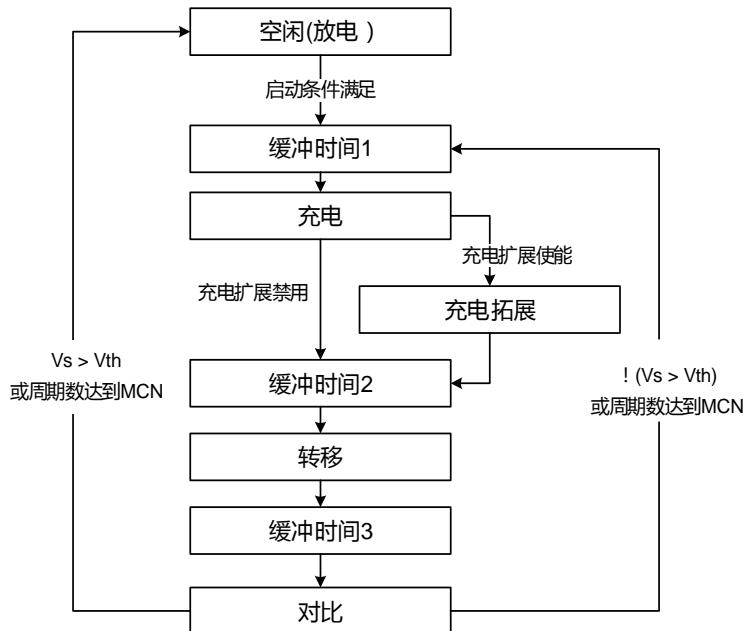
图 22-3. 电荷序列传输期间的采样引脚的电压



22.3.4. 电荷转移序列状态机

TSI 内部有一个硬件实现的有限状态机，来执行如前一节中描述的电荷转移序列，如 [图 22-4. 电荷转移序列的有限状态机的状态转移图](#) 所示。

图 22-4. 电荷转移序列的有限状态机的状态转移图



复位之后，状态机处于默认的 IDLE 状态。状态机有 2 种启动条件（由 TSI_CTL0 寄存器的 TRGMOD 位定义）：

TRGMOD = 0: 软件触发模式。在这一模式下，软件通过将 TSI_CTL0 寄存器的 TSIS 位写 1 来

启动状态机。

TRGMOD = 1: 硬件触发模式。在这种模式下，状态机会监测TSITG引脚，当发现上升沿或下降沿（极性可由寄存器配置）时自动启动。

状态机一旦启动就会按照上图所描述的跳转流程来运行。状态机如果在当前状态下的持续时间一旦达到配置值就会离开当前状态，并根据当前的条件进入下一个状态。

只有当TSI_CTL0寄存器中的ECEN位被置1，才会出现充电扩展这一状态，这个状态用于实现扩频功能，该状态会根据当前FSM循环次数扩展脉冲高电平持续时间。换言之，充电频率将变得动态且不固定。在有干扰的应用环境下，启用该功能能够提高TSI的鲁棒性，同时系统的电磁干扰将会降低。

在比较状态，该状态机会比较每个已启用的引脚组中的采样引脚电压和阈值电压。如果所有采样引脚的电压都达到了阈值，状态机返回空闲状态并停止，否则，状态机返回到缓冲时间1状态，继续下一个周期。如[图22-3. 电荷序列传输期间的采样引脚的电压](#)所示，在27个周期之后， V_s (采样引脚的电压) 到达 V_{th} (阈值电压)。

在TSI_CTL0寄存器的MCN位也定义了一个最大周期数，当周期数达到 MCN，状态机将会在比较状态之后返回 IDLE状态并停止，不管此时 V_s 是否达到了 V_{th} 。

22.3.5. 状态时钟和持续时间

TSI模块里有3个时钟：HCLK、CTCLK和ECCLK。 HCLK为系统时钟，它驱动TSI的寄存器和状态机。 CTCLK是从HCLK分频得到的（分频系数由CTCDIV定义），它是用来计算充电状态和电荷转移状态持续时间的时钟。 ECCLK也是从HCLK分频得到（分频系数由ECCDIV定义），它用来计算扩展充电状态的最大持续时间。 ECCLK和CTCLK是相互独立的。

除扩展充电状态外，其他状态的持续时间在每一次循环周期中都是固定的(根据寄存器配置)。

缓冲时间1、缓冲时间2和缓冲时间3的持续时间固定为2 HCLK周期。充电状态、电荷传输状态的持续时间由CDT与CTDT寄存器定义（详细见TSI_CTL0寄存器部分）。

通常，扩展充电频率的变化范围限制在10%到50%之间，在电荷转移状态机的每个周期中扩展充电状态的持续时间由TSI_CTL0寄存器的ECDT[6:0]控制位定义。如果扩展充电状态被使能，

最长的充电时间是当周期数为ECDT+2时。扩展充电状态在每个周期的持续时间变化规律如[表22-2. 充电扩展状态的持续时间](#)所示：

表 22-2. 充电扩展状态的持续时间

周期数	充电扩展状态持续时间
1	0
2	1
...	
ECDT	ECDT-1
ECDT+1	ECDT
ECDT+2	ECDT+1
ECDT+3	ECDT
ECDT+4	ECDT-1

周期数	充电扩展状态持续时间
...	...
2*ECDT+1	2
2*ECDT+2	1
2*ECDT+3	0
2*ECDT+4	1
2*ECDT+5	2
...	...

表 22-3 基于 HCLK 周期的扩频偏差

HCLK 周期	不同 ECDIV 值的扩频偏差(ECDT=0x7F)		
	ECDIV[2:0]=0x0 (最小)	ECDIV[2:0]=0x1	ECDIV[2:0]=0x7(最大)
41.6ns (24MHz)	5333.3ns	10666.6ns	42666.6ns
20.8ns (48MHz)	2666.6ns	5333.3ns	21333.3ns
13.8ns (72MHz)	1777.7ns	3555.5ns	14222.2ns
11.9ns (84MHz)	1523.8ns	3047.6ns	12190.4ns
9.26ns(108MHz)	1185.18ns	2370.37ns	9481.48ns

22.3.6. PIN 模式和 TSI 控制

每一个引脚组包括4个引脚，每个这些引脚任意一个都能够被用作采样引脚或通道引脚。但是在一个引脚组内，只有一个引脚应配置为采样引脚，通道引脚可以是多于一个，但任何情况下采样引脚和通道引脚都不能配置为同一个引脚。

一旦在GPIO中将一个引脚配置为TSI引脚，并在TSI中将其配置为采样或通道引脚，则在电荷转移序列中，该引脚的模式就由TSI控制。一般情况下，每个引脚有3种模式：输入、输出高和输出低。

一个通道引脚或采样引脚在电荷转移序列期间的模式已在[表22-1. 电荷转移序列的详细步骤以及引脚和开关状态](#)中具体描述，其中，PIN0代表一个通道引脚，PIN1代表一个采样引脚，即当序列在进行时，电荷转移状态机控制这些通道和采样引脚的模式以及相关模拟开关的状态。当序列处于空闲状态时，TSI_CTL0寄存器中的PINMOD控制位定义了这些引脚的模式。此外，如果一个引脚在GPIO中配置为TSI引脚，但在TSI模块配置中它既不是采样引脚，也不是通道引脚，我们称之为自由引脚，该引脚的模式也由TSI_CTL0寄存器的PINMOD位定义。

22.3.7. ASW 和 I/O 迟滞模式

当状态机正在运行时，一个通道或采样引脚的模拟开关由电荷转移序列控制，如[表22-1. 电荷转移序列的详细步骤以及引脚和开关状态](#)所示。当状态机处于空闲状态时，这些引脚的模拟开关由TSI_ASW寄存器中对应的GxPy位控制，此外所有的自由引脚的模拟开关也由GxPy位控制（自由引脚的定义见上一节）。

值得注意的是，TSI模块会始终控制引脚的模拟开关，即使这些引脚在GPIO中没有配置成TSI引脚。用户可以利用这个特性，通过软件读写GxPy位来控制这些模拟开关，同时通过GPIO方

式直接配置引脚的输出/输入模式，这样就可以实现用户自定义的电荷转移序列。

TSI控制器具有GPIO最高的优先级，当TSI被使能，该配置变为可用，而不受GPIO寄存器、模式或其他外设的控制。

通过清除TSI_PHM寄存器中的GxPy控制位可以禁用TSI引脚的施密特触发器迟滞模式，该操作能够提高系统的抗干扰性。

22.3.8. TSI 操作流

TSI的正常软件运行流程如下所示：

系统初始化，如系统时钟配置，TSI相关的GPIO配置等等。

按要求编程TSI_CTL0, TSI_CHCFG, TSI_INTEN, TSI_SAMPCFG和TSI_GCTL寄存器的GEx位/位域。

通过设置TSI_CTL0寄存器的TSIEN位使能TSI。

通过设置TSI_INTC寄存器清除CCTCF和CMNERR中断标志位。

可选软件触发模式：设置TSIS位以启动充电传输序列。在硬件触发模式中，TSI由触发引脚的下降沿/上升沿启动。

等待TSI_INTF寄存器中的CTCF或MNERR中断标志位置位，并且通过写TSI_INTC控制位清除这些标志位。

读出TSI_GxCYCN寄存器的CYCN位。

22.3.9. TSI 标志和中断

表 22-4. TSI 错误和标志位

标识名称	描述	清零
CTCF	TSI 停止因为所有引脚组的采样电压 V_s 都达到了阈值电压 V_{th} 。	TSI_INTC 寄存器的 CCTCF 位
MNERR	TSI 停止由于周期数到达最大值	TSI_INTC 寄存器的 CMNERR 位

22.3.10. TSI GPIOs

表 22-5. TSI 引脚

TSI 组	TSI 引脚	GPIO 引脚
第 0 引脚组	PIN0	PA0
	PIN1	PA1
	PIN2	PA2
	PIN3	PA3
第 1 引脚组	PIN0	PA4
	PIN1	PA5

TSI 组	TSI 引脚	GPIO 引脚
	PIN2	PA6
	PIN3	PA7
第 2 引脚组	PIN0	PC5
	PIN1	PB0
	PIN2	PB1
	PIN3	PB2
第 3 引脚组	PIN0	PA9
	PIN1	PA10
	PIN2	PA11
	PIN3	PA12
第 4 引脚组	PIN0	PB3
	PIN1	PB4
	PIN2	PB6
	PIN3	PB7
第 5 引脚组	PIN0	PB11
	PIN1	PB12
	PIN2	PB13
	PIN3	PB14

22.4. TSI 寄存器

22.4.1. 控制寄存器 0 (TSI_CTL0)

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CDT[3:0]				CTDT[3:0]				ECDT[6:0]				ECEN			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECDIV	CTCDIV[2:0]			保留				MCN[2:0]		PINMOD	EGSEL	TRGMOD	TSIS	TSIEN	
rw				rw				rw				rw			

位/位域	名称	描述
31:28	CDT[3:0]	充电状态持续时间 CDT[3:0] 由软件置位和清除。这些位控制在电荷转移序列中充电状态的持续时间。 0000: $1 \times t_{CTCLK}$ 0001: $2 \times t_{CTCLK}$ 0010: $3 \times t_{CTCLK}$ 1111: $16 \times t_{CTCLK}$
27:24	CTDT[3:0]	电荷转移状态持续时间 CTDT[3:0] 由软件置位和清除。这些位控制在电荷转移序列中电荷转移状态的持续时间。 0000: $1 \times t_{CTCLK}$ 0001: $2 \times t_{CTCLK}$ 0010: $3 \times t_{CTCLK}$ 1111: $16 \times t_{CTCLK}$
23:17	ECDT[6:0]	扩展充电状态最大持续时间 ECDT[6:0] 由软件置位和清除。这些位控制在电荷转移序列中充电扩展状态的最大持续时间。注：仅当 TSI_CTL0 寄存器的 ECEN 位置 1 时才有充电扩展状态出现。 0000000: $1 \times t_{ECCLK}$ 0000001: $2 \times t_{ECCLK}$ 0000010: $3 \times t_{ECCLK}$ 1111111: $128 \times t_{ECCLK}$

16	ECEN	扩展充电状态使能 0: 充电扩展禁用 1: 充电扩展启用
15	ECDIV	ECCLK时钟分频系数 ECCLK由HCLK分频得到， ECDIV定义了分频系数。 0x0: $f_{ECCLK} = f_{HCLK}$ 0x1: $f_{ECCLK} = f_{HCLK}/2$ 0x2: $f_{ECCLK} = f_{HCLK}/3$ 0x3: $f_{ECCLK} = f_{HCLK}/4$ 0x4: $f_{ECCLK} = f_{HCLK}/5$ 0x5: $f_{ECCLK} = f_{HCLK}/6$ 0x6: $f_{ECCLK} = f_{HCLK}/7$ 0x7: $f_{ECCLK} = f_{HCLK}/8$
		注意： ECDIV[2:1]定义在TSI_CTL1中，并且ECDIV[0]定义在TSI_CTL0中。
14:12	CTCDIV[2:0]	CTCLK 时钟分频系数 CTCLK由HCLK分频得到， CTCDIV定义了该分频系数。 0000: $f_{CTCLK} = f_{HCLK}$ 0001: $f_{CTCLK} = f_{HCLK}/2$ 0010: $f_{CTCLK} = f_{HCLK}/4$ 0011: $f_{CTCLK} = f_{HCLK}/8$ 0111: $f_{CTCLK} = f_{HCLK}/128$ 1000: $f_{CTCLK} = f_{HCLK}/256$ 1001: $f_{CTCLK} = f_{HCLK}/512$ 1110: $f_{CTCLK} = f_{HCLK}/16384$ 1111: $f_{CTCLK} = f_{HCLK}/32768$
		注意： CTCDIV[3]定义在TSI_CTL1中，并且CTCDIV[2:0]定义在TSI_CTL0中。
11:8	保留	必须保持复位值
7:5	MCN[2:0]	一个序列的最大周期数 MCN[2:0] 定义电荷转移序列的最大充电、转移周期数，一旦达到这个值，不论采样电压是否达到阈值电压序列都会停止。 000: 255 001: 511 010: 1023 011: 2047 100: 4096 101: 8191

		110: 16383
		111: 保留
4	PINMOD	引脚模式 该位定义在状态机为空闲状态时TSI引脚的模式。 0: TSI引脚输出低电平 1: TSI引脚保持悬空输入模式
3	EGSEL	边沿类型选择 该位定义硬件触发模式下的边沿类型。 0: 下降沿触发 1: 上升沿触发
2	TRGMOD	触发模式选择 0: 软件触发模式, 当TSIS位置位后, 序列将启动 1: 硬件触发模式, 在触发引脚检测到一个上升沿或下降沿时, 序列将启动
1	TSIS	TSI 启动 在软件触发模式下, 由软件置1, 用于启动一次电荷转移序列, 序列结束后硬件自动清零。置位该位后, 软件也可以手动清零来停止已启动的序列。 0: TSI不启动 1: TSI启动
0	TSIEN	TSI 使能 0: TSI模块使能 1: TSI模块禁用

22.4.2. 中断使能寄存器 (TSI_INTEN)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问

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

位/位域	名称	描述
31:2	保留	必须保持复位值
1	MNERRIE	最大循环次数错误中断使能 0: MNERR 中断禁用

1: MNERR 中断使能

0	CTCFIE	电荷转移完成标志中断使能
		0: CTCF 中断禁用
		1: CTCF 中断使能

22.4.3. 中断标志位清除寄存器 (TSI_INTC)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问

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

位/位域	名称	描述
31:2	保留	必须保持复位值
1	CMNERR	清除最大循环次数错误 0: 保留 1: 清MNERR
0	CCTCF	清除电荷转移完成标志 0: 保留 1: 清CTCF

22.4.4. 中断标志位寄存器 (TSI_INTF)

地址偏移: 0x0C

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问

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

r r

位/位域	名称	描述
31:2	保留	必须保持复位值
1	MNERR	<p>最大循环次数错误</p> <p>因为到达MCN[2:0]定义的最大周期数，在电荷转移序列停止后，该位由硬件置位。软件可通过向TSI_INTC寄存器的CMNERR位写1来清零。</p> <p>0: 没有最大循环次数错误 1: 最大循环次数错误</p>
0	CTCF	<p>电荷转移完成标志</p> <p>因为所有的使能的组的采样引脚达到电压阈值或到达MCN[2:0]定义的最大周期数，在电荷转移序列停止后，该位由硬件置位。软件可通过向TSI_INTC寄存器的CCTCF位写1来清零。</p> <p>0: 电荷转移没有完成 1: 电荷转移完成</p>

22.4.5. 引脚迟滞模式寄存器 (TSI_PHM)

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16							
保留														G5P3	G5P2	G5P1	G5P0	G4P3	G4P2	G4P1	G4P0	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	rw	rw	rw	rw	rw	rw	rw
G3P3	G3P2	G3P1	G3P0	G2P3	G2P2	G2P1	G2P0	G1P3	G1P2	G1P1	G1P0	G0P3	G0P2	G0P1	G0P0	rw	rw	rw	rw	rw	rw	rw
rw	rw																					

位/位域	名称	描述
31:24	保留	必须保持复位值
23:0	GxPy	<p>引脚迟滞模式</p> <p>该位由软件置位和清除。</p> <p>0: Pin GxPy 施密特触发迟滞模式禁用 1: Pin GxPy 施密特触发迟滞模式使能</p>

22.4.6. 模拟开关寄存器 (TSI_ASW)

地址偏移: 0x18

复位值: 0x0000 0000

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16						
保留														G5P3	G5P2	G5P1	G5P0	G4P3	G4P2	G4P1	G4P0

地址偏移: 0x20

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

位/位域	名称	描述
31:24	保留	必须保持复位值
23:0	GxPy	<p>采样引脚模式</p> <p>该位由软件置位和清除。</p> <p>0: GxPy 引脚不是采样引脚</p> <p>1: GxPy 引脚是采样引脚</p>

22.4.8. 通道配置寄存器 (TSI_CHCFG)

地址偏移: 0x28

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16
保留 G5P3 G5P2 G5P1 G5P0 G4P3 G4P2 G4P1 G4P0



位/位域	名称	描述
31:24	保留	必须保持复位值
23:0	GxPy	<p>通道引脚模式 该位由软件置位和清除。</p> <p>0: GxPy 引脚不是通道引脚 1: GxPy 引脚是通道引脚</p>

22.4.9. 组控制寄存器 (TSI_GCTL)

地址偏移: 0x30

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								GC5	GC4	GC3	GC2	GC1	GC0		
								r	r	r	r	r	r		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								GE5	GE4	GE3	GE2	GE1	GE0		
								rW	rW	rW	rW	rW	rW		

位/位域	名称	描述
31:24	保留	必须保持复位值
21:16	GCx	<p>组完成</p> <p>当一个使能组的电荷转移序列完成的时候，该位由硬件置位。当一个新的电荷转移序列开始的时候，由硬件清0。</p> <p>0: 组 x 电荷转移没有完成 1: 组 x 电荷转移完成</p>
15:6	保留	必须保持复位值
5:0	GEx	<p>引脚组使能</p> <p>该位由软件置位和清除。</p> <p>0: 引脚组 x 禁用 1: 引脚组 x 使能</p>

22.4.10. 组 x 周期数寄存器 (**TSI_GxCYCN**) (x=0..5)

地址偏移: 0x30 + 0x04 * (x + 1) Group Number x(0-5)

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问

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

rw

位/位域	名称	描述
31:14	保留	必须保持复位值
13:0	CYCN[13:0]	周期数目 这些位反映了电荷转移序列完成时组x执行的周期数。当一个新的电荷转移序列开始的时候，由硬件清0。

22.4.11. 控制寄存器 1(**TSI_CTL1**)

地址偏移: 0x300

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问

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

保留

位/位域	名称	描述
31:30	保留	必须保持复位值
29:28	ECDIV[2:1]	ECCLK 时钟分频系数 TSI中的ECCLK时钟是由HCLK分频而来，并且ECDIV定义了该分频系数 0x0: $f_{ECCLK} = f_{HCLK}$ 0x1: $f_{ECCLK} = f_{HCLK}/2$ 0x2: $f_{ECCLK} = f_{HCLK}/3$ 0x3: $f_{ECCLK} = f_{HCLK}/4$ 0x4: $f_{ECCLK} = f_{HCLK}/5$ 0x5: $f_{ECCLK} = f_{HCLK}/6$ 0x6: $f_{ECCLK} = f_{HCLK}/7$

0x7: $f_{ECCLK} = f_{HCLK}/8$

注意: ECDIV[2:1]定义在TSI_CTL1中，并且ECDIV[0]定义在TSI_CTL0中。

27:25 保留 必须保持复位值

24 CTCDIV[3] CTCLOCK时钟分频系数

TSI中的CTCLOCK时钟是由HCLK分频而来，并且CTCDIV定义了该分频系数

0000: $f_{CTCLK} = f_{HCLK}$

0001: $f_{CTCLK} = f_{HCLK}/2$

0010: $f_{CTCLK} = f_{HCLK}/4$

0011: $f_{CTCLK} = f_{HCLK}/8$

....

0111: $f_{CTCLK} = f_{HCLK}/128$

1000: $f_{CTCLK} = f_{HCLK}/256$

1001: $f_{CTCLK} = f_{HCLK}/512$

....

1110: $f_{CTCLK} = f_{HCLK}/16384$

1111: $f_{CTCLK} = f_{HCLK}/32768$

注意: CTCDIV[3]定义在TSI_CTL1中，并且CTCDIV[2:0]定义在TSI_CTL0中。.

23:0 保留 必须保持复位值

23. 通用串行总线全速接口（USBFS）

23.1. 简介

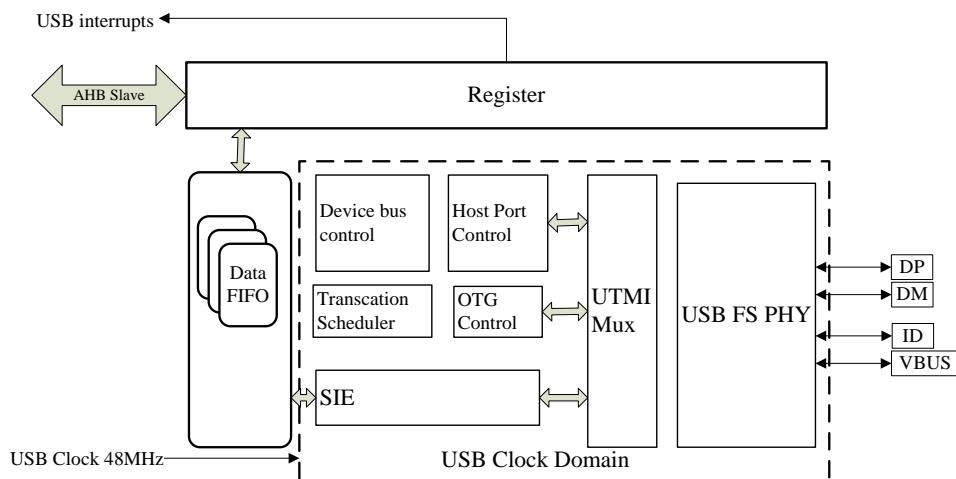
USB全速（USBFS）控制器为便携式设备提供了一套USB通信解决方案。USBFS不仅提供了主机模式和设备模式，也提供了遵循HNP（主机协商协议）和SRP（会话请求协议）的OTG模式。USBFS包含了一个内部的全速USB PHY，并且不再需要外部PHY芯片。USBFS可提供USB 2.0协议所定义的所有四种传输方式（控制传输、批量传输、中断传输和同步传输）。

23.2. 主要特性

- 支持USB 2.0全速（12Mb/s）/低速（1.5Mb/s）主机模式；
- 支持USB 2.0全速（12Mb/s）设备模式；
- 支持遵循HNP（主机协商协议）和SRP（会话请求协议）的OTG协议；
- 支持所有的4种传输方式：控制传输、批量传输、中断传输和同步传输；
- 在主机模式下，包含USB事务调度器，用于有效地处理USB事务请求；
- 包含一个1.25KB的FIFO RAM；
- 在主机模式下，支持8个通道；
- 在主机模式下，包含2个发送FIFO（周期性发送FIFO和非周期性发送FIFO）和1个接收FIFO（由所有的通道共享）；
- 在设备模式下，包含4个发送FIFO（每个IN端点一个发送FIFO）和1个接收FIFO（由所有的OUT端点共享）；
- 在设备模式下，支持4个OUT端点和4个IN端点；
- 在设备模式下，支持远程唤醒功能；
- 包含一个支持USB协议的全速USB PHY；
- 在主机模式下，SOF的时间间隔可动态调节；
- 可将SOF脉冲输出到PAD；
- 可检测ID引脚电平和VBUS电压；
- 在主机模式或者OTG A设备模式下，需要外部部件为连接的USB设备提供电源。

23.3. 结构框图

图 23-1. USBFS 结构框图



23.4. 信号线描述

表 23-1. USBFS 信号线描述

I/O 端口	类型	描述
VBUS	输入	总线电源端口
DM	输入/输出	差分信号 D-端口
DP	输入/输出	差分信号 D+端口
ID	输入	USB 识别: 微连接器识别接口

23.5. 功能描述

23.5.1. USBFS 时钟及工作模式

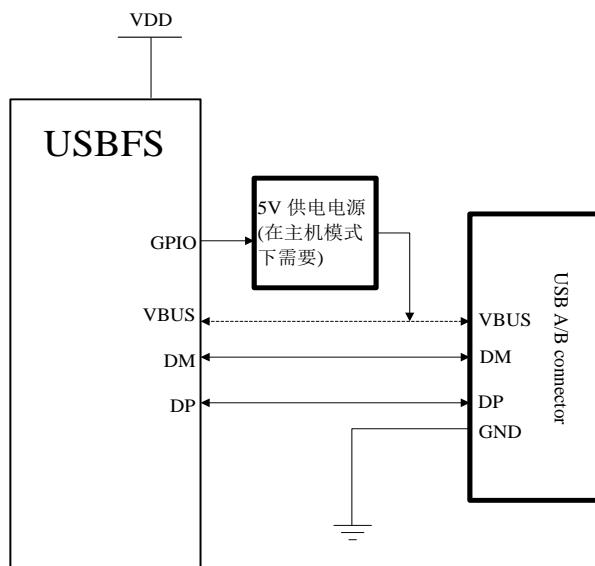
USBFS可以作为一个主机、一个设备或者一个DRD（双角色设备），并且包含一个内部全速PHY。USBFS可支持的最大速率为全速。

内部PHY支持全速和低速的主机模式、全速的设备模式以及具备HNP和SRP的OTG模式。USBFS所使用的USB时钟需要配置为48MHz。该48MHz USB时钟从系统内部时钟产生，并且其时钟源和分频器需要在RCU模块中配置。

上拉或下拉电阻已经集成在内部全速PHY的内部，并且USBFS可根据当前模式（主机、设备或OTG模式）和连接状态进行自动选择。一个利用内部全速PHY的典型连接示意图如图23-2. 在

[主机或设备模式下连接示意图](#)所示。

图 23-2. 在主机或设备模式下连接示意图

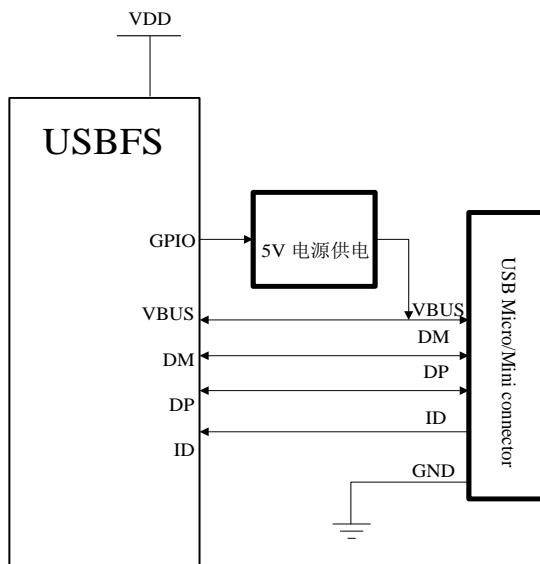


当USBFS工作在主机模式下时（FHM控制位置位、PDM控制位清除），VBUS为USB协议所定义的5V电源检测引脚。内部PHY不能提供5V VBUS电源，仅在VBUS信号线上具有电压比较器和充电放电电路。所以，如果应用需要提供VBUS电源，那么则需要一个外部的供电电源IC。在主机模式下，USBFS和USB连接头之间的VBUS连接可以被忽略，这是由于USBFS并不检测VBUS引脚的电平状态，并假定5V供电电源一直存在。

当USBFS工作在设备模式下时（FHM控制位清除、FDM控制位置位），VBUS检测电路由USBFS_GCCFG寄存器中的VBUSIG控制位所确定。所以，如果设备不需要检测VBUS引脚电压，可以置位VBUSIG控制位，并可释放VBUS引脚作为其他用途。否则，VBUS引脚的连接不能被忽略，并且USBFS需要不断的检测VBUS电平状态，一旦VBUS电压降至所需有效值以下，需要立即关闭DP信号线上的上拉电阻，这样则会产生一个断开状态。

OTG 模式连接示意图如[图 23-3. OTG 模式下连接示意图](#)所示。当 USBFS 工作在 OTG 模式下时，USBFS_GUSBCS 寄存器内的 FHM、FDM 控制位和 VBUSIG 位都应该被清除。在这种模式下，USBFS 需要以下四个引脚：DM、DP、VBUS 和 ID，并且需要使用若干个电压比较器检测这些引脚的电压。USBFS 也包含 VBUS 充电和放电电路，用以完成 OTG 协议中所描述的 SRP 请求。OTG A 设备或 B 设备由 ID 引脚的电平状态所决定。在实现 HNP 协议的过程中，USBFS 控制上拉和下拉电阻。

图 23-3. OTG 模式下连接示意图

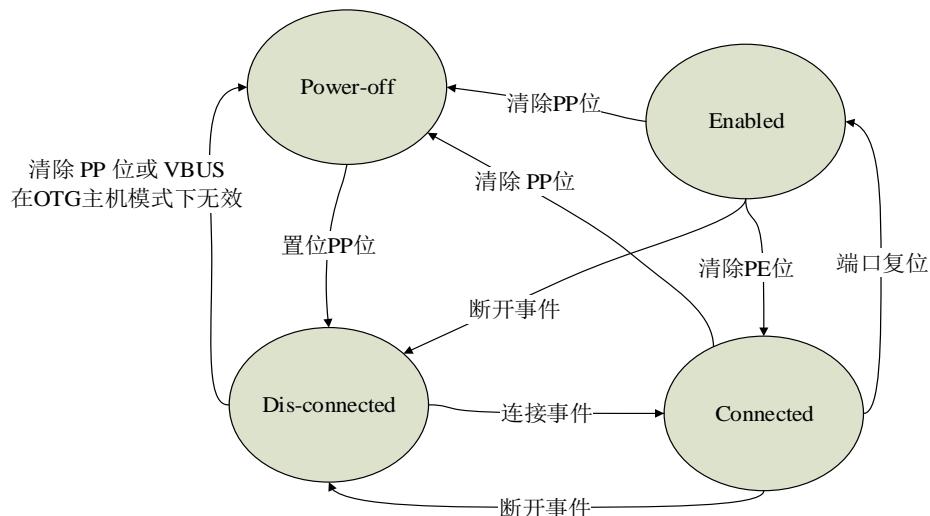


23.5.2. USB 主机功能

USB 主机端口状态

主机应用可以通过USBFS_HPCS寄存器控制USB端口状态。系统初始化之后，USB端口保持掉电状态。通过软件置位PP控制位后，内部USB PHY将被上电，并且USB端口变为断开状态。检测到连接后，USB端口变为连接状态。在USB总线上产生一个复位后，USB端口将变为使能状态。

图 23-4. 主机端口状态转移图



连接、复位和速度识别

作为USB主机，在检测到一个连接事件后，USBFS会为应用触发一个连接标志；同样，若检测到一个断开事件后，将会触发一个断开标志。

PRST控制位用于实现USB复位序列。应用可以置位该控制位以启动一个USB复位序列，或者清除该控制位以结束USB复位序列。仅当端口在连接或使能状态时，该控制位有效。

USBFS在对设备连接和复位时执行速度检测，并且速度检测的结果会反馈在USBFS_HPCS寄存器的PS位域中。USBFS以DM或DP的电平状态确定设备速度，如USB协议所描述，全速设备上拉DP信号线，而低速设备上拉DM信号线。

挂起和复位

USBFS支持挂起和复位状态，当USBFS端口在使能状态时，向USBFS_HPCS寄存器的PSP控制位写1，USBFS会进入到挂起状态。在挂起状态时，USBFS停止在USB总线上发送SOF，并且这样会让连接的USB设备在3ms后进入挂起状态。应用程序能够置位USBFS_HPCS寄存器中的PREM控制位以启动一个恢复序列，从而唤醒挂起的设备；清除该控制位可以停止恢复序列。如果主机在挂起状态检测到一个远程唤醒信号，将会置位USBFS_GINTF寄存器的WKUPIF标志位，并且触发USBFS唤醒中断。

SOF产生器

在主机模式下，USBFS向USB总线发送SOF令牌包。如USB 2.0协议所描述，全速连接下，每毫秒产生一次SOF令牌包（由主机控制器或者HUB事务转换器产生）。

每当USBFS进入到使能状态后，它将会按照USB2.0所定义的周期产生SOF令牌包。然而，应用程序可以通过写USBFS_HFT寄存器中的FRI位来调整一帧的间隔。FRI位定义了在一帧中的USB时钟周期个数，并且应用程序应该基于USBFS所使用的USB时钟频率计算该值。FRT位指示当前帧剩余的时钟周期个数，并且在挂起状态时，该值将停止改变。

USBFS能够在每个SOF令牌包中产生一个脉冲信号，并且将其输出至一个引脚。该脉冲信号长度为16个HCLK周期。如果应用程序希望使用该功能，需要置位USBFS_GCCFG寄存器的SOFOEN控制位，并且配置相应的引脚寄存器为GPIO功能。

USB通道和事务

USBFS在主机模式下包含8个独立的通道。每个通道能够与一个USB设备端点通信。通道的传输类型、方向、数据包长和其他信息都在通道相应的寄存器中配置，例如USBFS_HCHxCTL和USBFS_HCHxLEN寄存器。

USBFS支持所有的四种传输类型：控制、批量、中断和同步。USB 2.0协议将这些传输类型划分为两类：非周期性传输（控制和批量）和周期性传输（中断和同步）。基于此，为了有效地进行事务调度，USBFS包含两种请求队列：周期性请求队列和非周期性请求队列。在上述请求队列中的请求条目可能代表一个USB事务请求或者一个通道操作请求。

如果应用程序想要在USB总线上启动一个OUT事务，需要通过AHB寄存器接口向数据FIFO中写入数据包。USBFS硬件会在整包数据写完后，自动产生一个事务请求并进入请求队列。

请求队列中的请求条目通过事务控制模块按顺序处理。USBFS通常首先尝试处理周期性请求队列，然后处理非周期性请求队列。

帧起始后，USBFS首先开始处理周期性队列，直到队列为空抑或当前周期性请求队列所需时间不够，然后处理非周期性队列。这种做法保证了一帧中周期性传输的带宽。每次USBFS从请求队列中读取并取出一个请求条目。如果取出的是通道禁用请求，这将直接禁用通道并准备处理

下个条目。

如果当前请求是一个事务请求并且USB总线时间能够处理这个请求，USBFS会使用SIE在USB总线上产生该事务。

在当前帧内，当前请求所需的总线时间不足时，如果当前请求为周期性请求，USBFS停止处理该周期性请求队列，并启动处理非周期性请求。如果当前请求为非周期性请求，USBFS会停止处理任何队列，并等待直到当前帧结束。

23.5.3. USB 设备功能

USB设备连接

在设备模式下，USBFS在初始化后保持掉电状态。利用VBUS引脚上的5V电源连接USB主机后或者置位USBFS_GCCFG寄存器中VBUSIG控制位，USBFS将进入供电状态。USBFS首先打开DP信号线上的上拉电阻，之后主机将会检测到一个连接事件。

复位和速度识别

USB主机在检测到设备连接之后，总是会启动一个USB复位序列，并且在设备模式下，检测到USB总线复位事件后，USBFS会为软件触发一个复位中断。

在复位序列后，USBFS将会触发USBFS_GINTF寄存器中的ENUMF中断，并且利用USBFS_DSTAT寄存器内的ES标志位指示当前枚举设备速度，该位总是为11（全速）。

如USB 2.0协议所描述，USBFS在外设模式下不支持低速。

挂起和唤醒

USB总线保持IDLE状态并且数据线3ms无变化，USB设备将会进入挂起状态。当USB设备在挂起状态时，软件能够关闭大部分的时钟以节省电能。USB主机可以通过在USB总线上产生恢复信号来唤醒挂起的设备。USBFS检测到恢复信号后，将置位USBFS_GINTF寄存器的WKUPIF标志位并且触发USBFS唤醒中断。

在挂起设备模式，USBFS也能够远程唤醒USB总线。软件可以通过置位USBFS_DCTL寄存器的RWKUP控制位来发送一个远程唤醒信号，并且如果USB主机支持远程唤醒，主机会在USB总线上启动发送一个恢复信号。

软件断开

USBFS支持软件断开。设备进入到供电状态后，USBFS会打开DP信号线的上拉电阻，并且这样主机会检测到设备连接。然后，软件可以通过置位USBFS_DCTL寄存器中SD控制位进行强制断开。在SD控制位被置位后，USBFS将会直接关闭上拉电阻。这样，USB主机将会在USB总线上检测到设备断开。

SOF跟踪

当USBFS在USB总线上接收到一个SOF令牌包时，将触发一个SOF中断，并且开始利用本地USB时钟计算总线时间。当前帧的帧号将会反应在USBFS_DSTAT寄存器的FNRSOF位域中。当USB总线时间达到EOF1或EOF2点（帧结束，在USB 2.0协议中描述），USBFS会触发USBFS_GINTF寄存器中的EOPFIF中断。软件能够使用这些标志位和寄存器以获得当前总线

时间和位置信息。

23.5.4. OTG 功能概述

USBFS支持OTG协议1.3中所描述的OTG功能，OTG功能包括SRP和HNP。

A设备和B设备

当标准A或微型A插头插入相应的插座时，具有OTG能力的USB设备为A设备。A设备向VBUS供电，并且在会话开始时默认为主机。当标准B、微型B、迷你B插头插入相应的插座或采用一端为标准A插头的不可分离电缆时，具有OTG能力的USB设备为B设备。B设备在会话开始时默认为外设。**USBFS**使用ID引脚电平状态决定A设备或B设备。ID引脚状态反馈在**USBFS_GOTGCS**寄存器的**IDPS**状态位。为了了解A设备和B设备之间传输的详细状态，请参考OTG1.3协议。

HNP

主机协商协议（**HNP**）允许主机功能在两个直接连接的OTG设备之间转换，并且用户不需要为了设备之间通信控制的改变而切换电缆线的连接。典型地，**HNP**协议是由B设备上的用户或应用启动，**HNP**只能通过设备上的微型AB插座执行。

一旦OTG设备具有一个微型AB插座，该OTG设备可通过插入的插头类型决定默认为主机或设备（微型A插头插入为主机，微型B插头插入为设备）。通过使用主机协商协议（**HNP**），一个默认为外设的OTG设备可以请求成为主机。主机角色切换的过程在下段中描述。此协议使用户不需要为了更改连接设备的角色而切换电缆线的连接。

当**USBFS**工作在OTG A主机模式时，并且其想放弃主机角色，可以首先置位**USBFS_HPCS**寄存器的**PSP**控制位来使USB总线进入挂起状态，然后B设备在3ms后进入挂起状态。如果B设备想要变为主机，软件需要置位**USBFS_GOTGCS**寄存器的**HNPREQ**控制位，然后**USBFS**会开始在总线上执行**HNP**协议，最后，**HNP**的结果会反馈在**USBFS_GOTGCS**寄存器的**HNPS**状态位。另外，软件总能从**USBFS_GINTF**寄存器的**COPM**状态位获取当前设备角色（主机或外设）。

SRP

会话请求协议（**SRP**）允许B设备请求A设备打开VBUS并启动一个会话。该协议允许A设备（或许是电池供电）当总线无活动时通过关闭VBUS以节省电能，并为B设备启动总线活动提供了一种方法。如OTG协议中所描述，OTG设备必须和几个阈值比较VBUS电压，并且将比较结果反馈在**USBFS_GOTGCS**寄存器的**ASV**和**BSV**状态位中。

当**USBFS**工作在B设备OTG模式时，软件可以通过置位**USBFS_GOTGCS**寄存器的**SRPREQ**控制位来启动一个SRP请求，并且如果SRP请求成功，**USBFS**会在**USBFS_GOTGCS**寄存器中产生一个成功标志位**SRPS**。

当**USBFS**工作在OTG A设备模式且从B设备检测到一个SRP请求时，**USBFS**将会置位**USBFS_GINTF**寄存器中的**SESIF**标志位。软件获取该标志位后，需要准备为VBUS引脚打开5V供电电源。

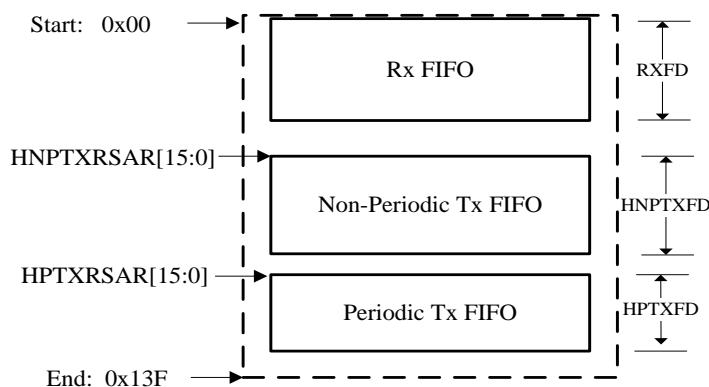
23.5.5. 数据 FIFO

USBFS中采用1.25K字节数据FIFO存储包数据, 数据FIFO是通过USBFS的内部SRAM实现的。

主机模式

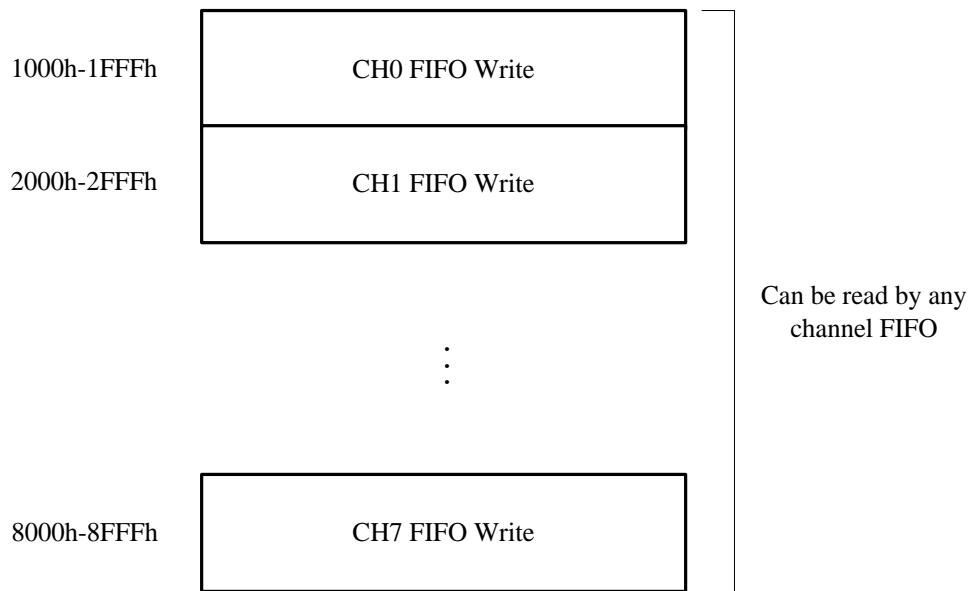
主机模式下, 数据 FIFO 空间分为三个部分, 分别是: 用于接收数据包的 Rx FIFO、用于非周期性发送数据包的非周期性 Tx FIFO 和用于周期性发送数据包的周期性 Tx FIFO。所有的 IN 通道通过共享 Rx FIFO 接收数据。所有的周期性 OUT 通道通过共享周期性 Tx FIFO 来发送数据, 所有的非周期性 OUT 通道通过共享非周期性 Tx FIFO 来发送数据。通过寄存器 USBFS_GRFLEN、USBFS_HNPTFLEN 和 USBFS_HPTFLEN, 软件可以配置以上数据 FIFO 的大小和起始偏移地址。[图 23-5. 主机模式 FIFO 空间](#)所描述的是 SRAM 中各 FIFO 的结构, 图中的数值是按照 32 位为单位写的。

图 23-5. 主机模式 FIFO 空间



USBFS 为程序提供了专有寄存器空间来读写数据 FIFO。[图 23-6. 主机模式 FIFO 访问寄存器映射表](#)所描述的是数据 FIFO 所访问的寄存器存储空间, 此区域可被任意主机通道读取。图中的数值是以字节为单位寻址。尽管所有的非周期通道共享相同的 FIFO 以及所有的周期通道共享相同的 FIFO, 每个通道都拥有它们的 FIFO 访问寄存器空间。对 USBFS 而言, 获知当前压入数据包的通道号是非常重要的, 通过寄存器 USBFS_GRXTATR/USBFS_GRSTATP 来访问数据包所从属的 Rx FIFO。

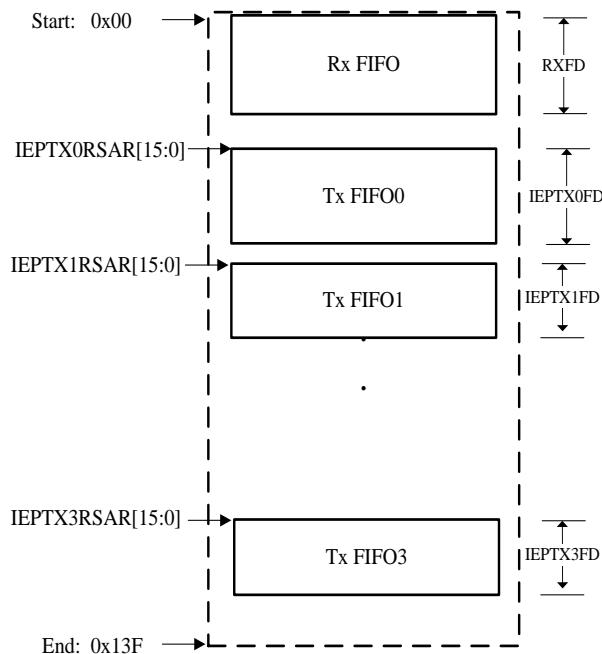
图 23-6. 主机模式 FIFO 访问寄存器映射表



设备模式

在设备模式下，数据 FIFO 分为多个部分，其中包含 1 个 Rx FIFO 和 4 个 Tx FIFO，每个 Tx FIFO 对应着一个 IN 端点，所有的 OUT 端点通过共享 Rx FIFO 接收数据包。通过寄存器 USBFS_GRFLEN 和 USBFS_DIEPxTFLEN ($x=0\ldots3$)，程序可配置数据 FIFO 的大小和起始偏移地址。[图 23-7. 设备模式 FIFO 空间](#)所描述的是 SRAM 中各 FIFO 的结构，图中的数值是以按照 32 位写的。

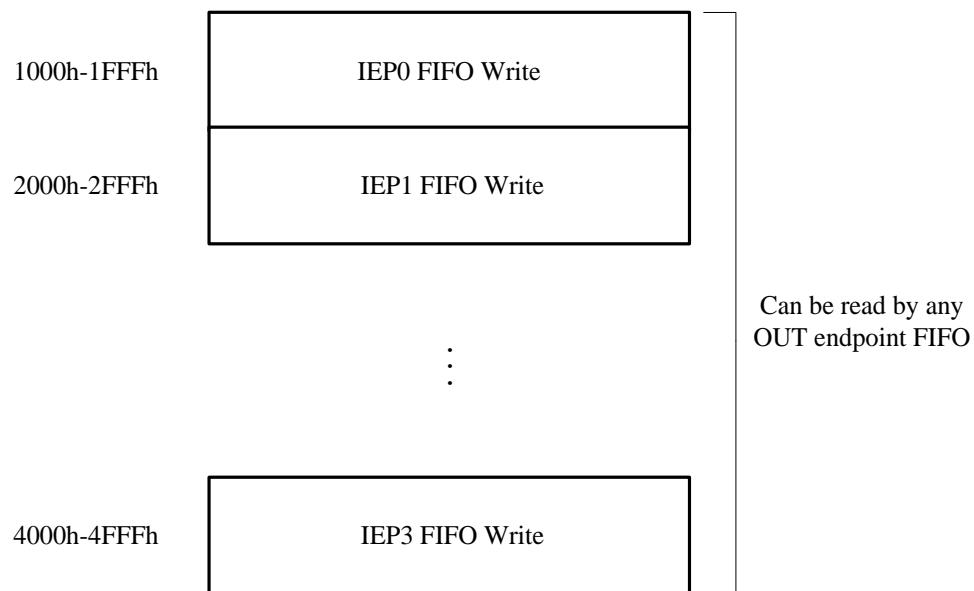
图 23-7. 设备模式 FIFO 空间



USBFS 为程序提供了专有寄存器空间来读写数据 FIFO。图 23-8. 设备模式 FIFO 访问寄存器

映射表所描述的是数据 FIFO 所访问的寄存器存储空间，此区域可被任意端点 FIFO 读取。图中的数值是以字节为单位寻址。每个端点都拥有它们的 FIFO 访问寄存器空间。通过寄存器 USBFS_GRXTATTR/USBFS_GRSTATP 来访问 Rx FIFO。

图 23-8. 设备模式 FIFO 访问寄存器映射表



23.5.6. 操作手册

该部分描述的是USBFS的操作手册。

主机模式

全局寄存器初始化顺序:

- 1、根据应用的需求，如Tx FIFO的空阈值等，设置寄存器USBFS_GAHBCS，此时，GINTEN位需要保持清零状态；
- 2、根据应用的需求，如操作模式（主机、设备或OTG）、某些OTG参数和USB协议，设置寄存器USBFS_GUSBCS；
- 3、根据应用的需求，设置寄存器USBFS_GCCFG；
- 4、根据应用的需求，设置寄存器USBFS_GRFLEN、USBFS_HNPTFLEN_DIEP0TFLEN、USBFS_HPTFLEN，配置数据FIFO；
- 5、通过设置寄存器USBFS_GINTEN使能模式错误和主机端口中断，置位USBFS_GAHBCS寄存器的GINTEN位使能全局中断；
- 6、设置寄存器USBFS_HPCS，置位PP位；
- 7、等待设备连接，当设备连接后，触发寄存器USBFS_HPCS的PCD位，然后置位PRST位，执行一次端口复位，等待至少10毫秒后，清除PRST位；
- 8、等待USBFS_HPCS寄存器的PEDC中断，然后读取PE位以确认端口被成功地使能，读取PS位以获取连接的设备速度，之后，如果软件需要改变SOF间隔，设置USBFS_HFT寄存器。

通道初始化和使能顺序:

- 1、根据期望的传输类型、方向、包大小等信息，设置寄存器USBFS_HCHxCTL，在设置期间，要保证位CEN和CDIS保持清除；
- 2、设置寄存器USBFS_HCHxINTEN，设置期望的中断使能位；
- 3、设置寄存器USBFS_HCHxLEN，PCNT表示一次传输中的包数，TLEN表示一次传输中发送或接收的包数据的总字节数；
- 4、对于OUT通道，如果PCNT为1，单包的大小等于TLEN。如果PCNT大于1，前PCNT-1个包被认定为最大包长度的包，其大小是由寄存器USBFS_HCHxCTL的位MPL所定义。最后一包的大小可通过PCNT、TLEN和MPL计算得到。如果程序想要发出一个零长度的包，应该设定TLEN为0，PCNT位1；
- 5、对于IN通道，因为在IN事务结束之前，程序不知道实际接收的数据大小，程序可将TLEN设定为Rx FIFO所支持的最大值；
- 6、置位寄存器USBFS_HCHxCTL中的CEN位以使能通道。

通道除能顺序:

程序可以通过同时置位CEN和CDIS除能通道。在寄存器操作后，USBFS将在请求队列中产生一个通道除能请求条目。当这个请求条目到达请求队列的顶部时，USBFS立即进行处理。

对于OUT通道而言，特定的通道将被立即除能。然后，会产生CH标志，USBFS将清除CEN和CDIS位。

对于IN通道而言，USBFS将通道除能状态条目压入Rx FIFO，然后，程序应该处理Rx FIFO非空事件：读和取出该状态条目，然后会产生CH标志，USBFS将清除CEN和CDIS位。

IN传输操作顺序:

- 1、初始化USBFS全局寄存器；
- 2、初始化相应的通道；
- 3、使能相应的通道；
- 4、通过软件使能IN通道后，USBFS在相应请求队列中生成一个Rx请求条目；
- 5、当Rx请求条目到达请求队列的顶部时，USBFS开始执行该请求条目。对于由请求条目所指示的事务而言，如果总线时间足够，USBFS在USB总线上开始IN事务；
- 6、当IN事务结束时（收到ACK握手包），USBFS将接收到的数据包压入Rx FIFO，ACK标志位被触发，否则，状态标志（NAK）会指示事务结果；
- 7、如果步骤5所描述的IN事务完成后，步骤2的PCNT的数值比1大，程序将会返回步骤3，继续接收剩下的数据包。如果步骤5中描述的IN事务没有成功完成，程序将会返回步骤3来再次发送该数据包；
- 8、在所有的传输中的所有事务都被成功接收后，USBFS将TF状态条目压入Rx FIFO的最后的数据包的顶部，这样，软件在读取所有接收的数据包后，再读取TF状态条目。USBFS生成TF标志来指示传输成功结束；
- 9、除能通道，当通道处于空闲状态，即可为其他传输做准备。

OUT传输操作顺序:

- 1、初始化USBFS全局寄存器；
- 2、初始化及使能相应通道；

- 3、将数据包写入通道的Tx FIFO（周期性Tx FIFO或非周期性Tx FIFO）。在所有的数据包都被写入FIFO后，USBFS在相应的请求队列中产生一个Tx请求条目，并且将USBFS_HCHxTLEN中的TLEN值减少，减少的数值等于已写的包大小；
- 4、当请求条目到达请求队列的顶部时，USBFS开始执行该请求条目。如果请求条目对应的服务的总线时间足够，USBFS在USB总线上开展OUT事务；
- 5、当由请求条目所指示的OUT事务结束时，寄存器USBFS_HCHnTLEN的位PCNT减1。如果该事务完成（收到ACK握手包），ACK标志位被触发，否则，状态标志（NAK）会指示事务结果；
- 6、如果步骤5所描述的OUT事务完成后且步骤2的PCNT的数值比1大，程序将会返回步骤3，继续发送剩下的数据包。如果步骤5中描述的OUT事务没有成功完成，程序将会返回步骤3来再次发送该包；
- 7、在所有的传输中的所有事务都被成功送达后，USBFS生成TF标志来指示传输成功结束；
- 8、除能通道，当通道处于空闲状态，即可为其他传输做准备。

设备模式

全局寄存器初始化顺序：

- 1、根据应用的需求，如Tx FIFO的空阈值等，设置寄存器USBFS_GAHBCS，此时，GINTEN位需要保持清零状态；
- 2、根据应用的需求，如操作模式（主机、设备或OTG）、某些OTG参数、USB协议，设置寄存器USBFS_GUSBCS；
- 3、根据应用的需求，设置寄存器USBFS_GCCFG；
- 4、根据应用的需求，设置寄存器USBFS_GRFLEN、USBFS_HNPTFLEN_DIEP0TFLEN、USBFS_HPTFLEN，配置数据FIFO；
- 5、通过设置寄存器USBFS_GINTEN使能模式错误、挂起、SOF、枚举完成和USB复位中断，置位USBFS_GAHBCS寄存器的GINTEN位使能全局中断；
- 6、根据应用的需求，如设备的地址等，设置寄存器USBFS_DCFG；
- 7、在设备连接上主机上后，主机在USB总线上执行端口复位，触发寄存器USBFS_GINTF的RST中断；
- 8、等待寄存器USBFS_GINTF的ENUMF中断。

端点初始化和使能顺序：

- 1、根据预期的传输类型、包大小等信息，设置寄存器USBFS_DIEPnCTL 或 USBFS_DOEPxCTL；
- 2、设定寄存器USBFS_DIEPINTEN 或 USBFS_DOEPINTEN，置位相应中断使能位；
- 3、设定寄存器USBFS_DIEPxLEN 或 USBFS_DOEPxLEN，PCNT表示一次传输中的包数，TLEN 表示一次传输中发送或接收的数据包的总字节数；
- 4、对于 IN 端点，如果 PCNT 等于 1，单数据包的大小等于 TLEN。如果 PCNT 大于 1，前 PCNT-1 个包被认定为最大包长度的包，其大小是由寄存器 USBFS_DIEPxCTL 的位 MPL 所定义。最后一包的大小可通过 PCNT、TLEN 和 MPL 计算得到。如果程序想要发出一个零长度的包，应该设定 TLEN 为 0，PCNT 位 1；
- 5、对于 OUT 端点，因为在 IN 事务结束之前，程序不知道实际接收的数据大小，程序可将 TLEN 设定为 Rx FIFO 所支持的最大值；
- 6、置位 USBFS_DIEPxCTL 或 USBFS_DOEPxCTL 寄存器 EPEN 位使能端点。

端点除能顺序

当USBFS_DIEPnCTL或USBFS_DOEPnCTL寄存器的EPEN位被清除时，程序可以在任何时候除能端点。

IN传输操作顺序：

- 1、初始化USBFS全局寄存器；
- 2、初始化和使能IN端点；
- 3、将数据包写入端点的Tx FIFO，每当数据包写入FIFO，USBFS减少USBFS_DIEPxLEN寄存器的TLEN域的数值，其减少的数值等于已写的数据包大小；
- 4、当IN令牌接收后，USBFS发送数据包，在USB总线上的事务完成后，USBFS_DIEPxLEN寄存器的PCNT值减1。如果事务成功完成（接收到ACK握手包），ACK标志被触发，或者，其他状态标志表示事务的结果；
- 5、在一次传输的所有数据包都被成功发送，USBFS生成一个TF标志位以表明传输成功结束，除能相应IN端点。

OUT传输操作顺序（DMA除能）：

- 1、初始化USBFS全局寄存器；
- 2、初始化和使能端点；
- 3、当OUT令牌接收后，USBFS接收数据包或基于Rx FIFO状态和寄存器配置回复NAK握手包。如果事务成功完成（USBFS接收并保存数据到Rx FIFO，发送ACK握手包），USBFS_DOEPxLEN寄存器的PCNT值减1。如果事务成功完成（接收到ACK握手包），ACK标志被触发，或者，其他状态标志表示事务的结果；
- 4、在一次传输的所有数据包都被成功接收，USBFS将TF状态条目压入Rx FIFO的最后的数据包的顶部，这样，软件在读取所有接收的数据包后，再读取TF状态条目。USBFS生成TF标志来指示传输成功结束。USBFS生成一个TF标志位以表明传输成功结束，除能相应OUT端点。

23.6. 中断

OTG有两种中断：全局中断、唤醒中断。

全局中断是软件需要处理的主要中断，全局中断的标志位可在 USBFS_GINTF 寄存器读取，列举在 [表 23-2. USBFS 全局中断](#) 中。

表 23-2. USBFS 全局中断

中断标志	描述	运行模式
SEIF	会话中断	主机或设备模式
DISCIF	断开连接中断标志	主机模式
IDPSC	ID 引脚状态变化	主机或设备模式
PTXFEIF	周期性 Tx FIFO 空中断标志	主机模式
HCIF	主机通道中断标志	主机模式

中断标志	描述	运行模式
HPIF	主机端口中断	主机模式
ISOONCIF/PXNCIF	周期性传输未完成中断标志 / 同步OUT传输未完成中断标志	主机或设备模式
ISOINCIF	同步 IN 传输未完成中断标志	设备模式
OEPIF	OUT 端点中断标志	设备模式
IEPIF	IN 端点中断标志	设备模式
EOPFIF	周期性帧尾中断标志	设备模式
ISOOPDIF	同步 OUT 丢包中断标志	设备模式
ENUMF	枚举完成	设备模式
RST	USB 复位	设备模式
SP	USB挂起	设备模式
ESP	早挂起	设备模式
GONAK	全局OUT NAK有效	设备模式
GNPINAK	全局非周期IN NAK有效	设备模式
NPTXFEIF	非周期Tx FIFO空中断标志	主机模式
RXFNEIF	Rx FIFO非空中断标志	主机或设备模式
SOF	帧首	主机或设备模式
OTGIF	OTG 中断标志	主机或设备模式
MFIF	模式错误中断标志	主机或设备模式

唤醒中断可以在 USBFS 处于挂起状态时触发，即使 USBFS 的时钟停止。寄存器 USBFS_GINTF 的位 WKUPIF 是唤醒源。

23.7. USBFS 寄存器

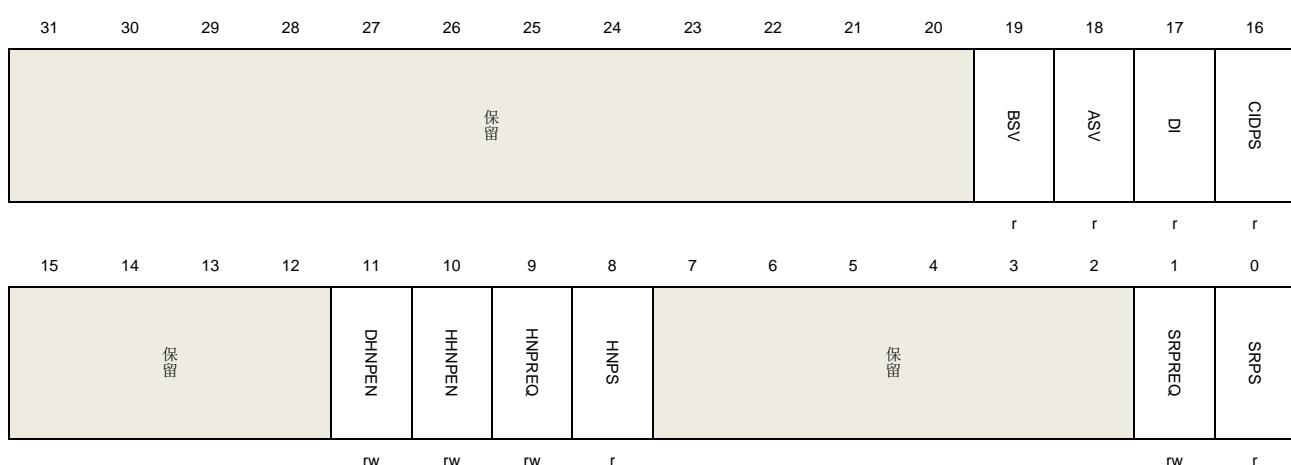
23.7.1. 全局控制与状态寄存器组

全局 OTG 控制和状态寄存器 (USBFS_GOTGCS)

地址偏移: 0x0000

复位值: 0x0000 0800

该寄存器只能按字 (32位) 访问



位/位域	名称	描述
31:20	保留	必须保持复位值
19	BSV	B会话有效 (在OTG协议中描述) 0: OTG B设备VBUS电压水平低于VBSESSVLD 1: OTG B设备VBUS电压水平不低于VBSESSVLD 注意: 仅在OTB B设备模式下可访问
18	ASV	A会话有效 A主机模式收发器状态 0: OTG A设备VBUS电压水平低于VASESSVLD 1: OTG A设备VBUS电压水平不低于VASESSVLD 在会话的开始, A设备默认是主机。 注意: 仅在OTG A设备模式下可访问
17	DI	去抖动间隔 检测到连接的去抖动间隔。 0: 当USB总线上发生插入和连接时, 表示长去抖动间隔 1: 当HNP协议中使用一个软连接时, 指示短去抖动间隔 注意: 仅在主机模式下可访问
16	CIDPS	ID引脚状态

		连接器ID引脚的电压水平 0: USBFS工作在A设备模式 1: USBFS工作在B设备模式 注意: 在设备和主机模式下均可访问
15:12	保留	必须保持复位值
11	DHNPEN	设备HNP使能 使能B设备HNP功能。如果该控制位清除，当应用置位USBFS_GOTGCS寄存器中的HNPREQ控制位c时，USBFS并不启动HNP协议。 0: HNP功能不使能 1: HNP功能使能 注意: 仅在设备模式下访问
10	HHNPEN	主机HNP使能 使能A设备HNP功能。如果该控制位清除，USBFS不能够响应B设备的HNP请求。 0: HNP功能不使能 1: HNP功能使能 注意: 仅在主机模式下访问
9	HNPREQ	HNP请求 软件通过置位该控制位在USB总线上启动一个HNP。当USBFS_GOTGINTF寄存器中HNPEND控制位置位时，软件可以通过向该控制位写0或者清除USBFS_GOTGINTF寄存器中的HNPEND控制位来清除该控制位。 0: 不发送HNP请求 1: 发送HNP请求 注意: 仅在设备模式下访问
8	HNPS	HNP成功标志位 当HNP成功时，该标志位由内核置位。当HNPREQ置位时，该控制位被清除。 0: HNP失败 1: HNP成功 注意: 仅在设备模式下访问
7:2	保留	必须保持复位值
1	SRPREQ	SRP请求 软件通过置位该控制位在USB总线上启动一个SRP会话请求。当USBFS_GOTGINTF寄存器中的SRPEND控制位置位时，软件可以通过向该控制位写0或者清除USBFS_GOTGINTF寄存器中的SRPEND控制位来清除该控制位。 0: 没有会话请求 1: 会话请求 注意: 仅在设备模式下访问
0	SRPS	SRP会话请求成功 当SRP会话请求成功时，该标志位由内核置位。当SREQ控制位被置位时，该标志位被清除。 0: SRP会话请求失败 1: SRP会话请求成功

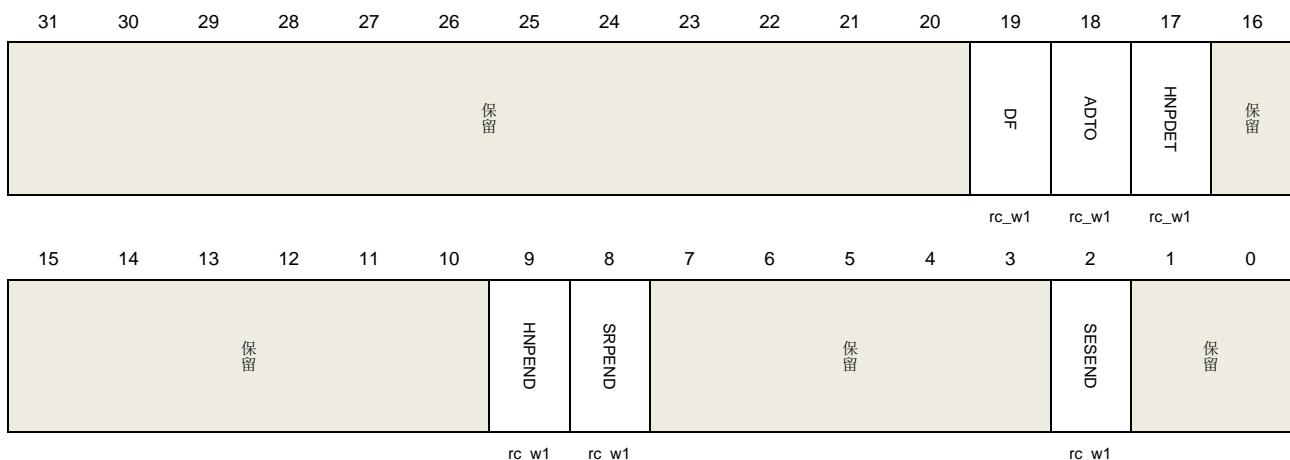
注意：仅在设备模式下访问

全局 OTG 中断状态寄存器 (USBFS_GOTGINTR)

地址偏移: 0x0004

复位值: 0x0000 0000

该寄存器只能按字（32位）访问



位/位域	名称	描述
31:20	保留	必须保持复位值
19	DF	去抖动完成 当设备连接去抖动完成时，USBFS置位该控制位 注意： 仅在主机模式下可访问
18	ADTO	A设备超时 当A设备等待B设备连接发生超时，USBFS置位该控制位 注意： 在设备和主机模式下，均可访问
17	HNPDET	检测到主机协商请求 当A设备检测到一个HNP请求时，USBFS置位该标志位 注意： 在设备和主机模式下，均可访问
16:10	保留	必须保持复位值
9	HNPEND	HNP结束 当一个HNP结束时，内核置位该标志位。软件应该读取USBFS_GOTGCS寄存器中HNPS标志位，以获取HNP结果。 注意： 在设备和主机模式下，均可访问。
8	SRPEND	SRPEND 当一个SRP结束时，内核置位该标志位。软件应该读取USBFS_GOTGCS寄存器中SRPS标志位，以获取SRP结果。

注意：在设备和主机模式下，均可访问。

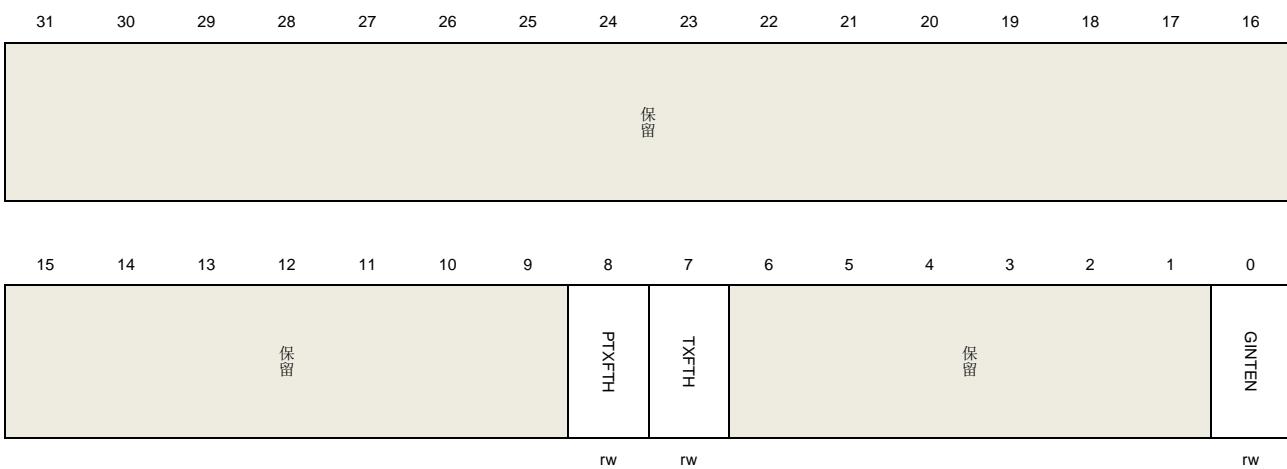
7:3	保留	必须保持复位值
2	SESEND	会话结束 当VBUS电压低于Vb_ses_vld时，内核置位该标志位。
1:0	保留	必须保留复位值

全局 AHB 控制和状态寄存器 (USBFS_GAHBCS)

地址偏移: 0x0008

复位值: 0x0000 0000

该寄存器只能按字（32位）访问



位/位域	名称	描述
31:9	保留	必须保持复位值
8	PTXFTH	周期性Tx FIFO阈值 0: 当周期性发送FIFO半空时，将触发PTXFEIF标志位 1: 当周期性发送FIFO全空时，将触发PTXFEIF标志位 注意： 只在主机模式下访问
7	TXFTH	Tx FIFO 阈值 设备模式： 0: 当IN端点发送FIFO半空时，将触发TXFEIF标志位 1: 当IN端点发送FIFO全空时，将触发TXFEIF标志位 主机模式： 0: 当非周期性发送FIFO半空时，将触发NPTXFEIF标志位 1: 当非周期性发送FIFO全空时，将触发NPTXFEIF标志位
6:1	保留	必须保持复位值
0	GINTEN	全局中断使能 0: 全局中断不使能

1: 全局中断使能

注意：在主机和设备模式下，均可访问

全局 USB 控制和状态寄存器 (USBFS_GUSBCS)

地址偏移: 0x000C

复位值: 0x0000 0A80

该寄存器只能按字(32位)访问

位/位域	名称	描述
31	保留	必须保持复位值
30	FDM	<p>强制设备模式</p> <p>通过置位该控制位，可强制USB内核为设备模式，并且忽略USBFS ID引脚的输入状态</p> <p>0: 正常模式 1: 设备模式</p> <p>设置该控制位后，应用必须等待至少25ms，让变化产生作用。</p> <p>注意：在设备和主机模式下，均可访问。</p>
29	FHM	<p>强制主机模式</p> <p>通过置位该控制位，可强制USB内核为主机模式，并且忽略USBFS ID引脚的输入状态</p> <p>0: 正常模式 1: 主机模式</p> <p>设置该控制位后，应用必须等待至少25ms，让变化产生作用。</p> <p>注意：在设备和主机模式下，均可访问</p>
28:14	保留	必须保持复位值
13:10	UTT[3:0]	<p>USB运转时间</p> <p>以物理时钟数来设定运转时间</p> <p>注意：仅在设备模式下访问</p>

9	HNPCEN	HNP能力使能 控制HNP能力是否使能 0: HNP能力禁用 1: HNP能力使能 注意: 在设备和主机模式下, 均可访问
8	SRPCEN	SRP能力使能 控制SRP能力是否使能 0: SRP能力禁用 1: SRP能力使能 注意: 在设备和主机模式下, 均可访问
7:3	保留	必须保持复位值
2:0	TOC[2:0]	超时校准 当等待一个包时, USBFS需要使用USB2.0协议中需要的超时数值。应用可以使用TOC[2:0]增加该数值(以PHY时钟为单位)。PHY时钟频率为48MHz。

全局复位控制寄存器 (**USBFS_GRSTCTL**)

地址偏移: 0x0010

复位值: 0x8000 0000

应用通过该寄存器来复位内核的不同硬件特性。

该寄存器只能按字(32位)访问



位/位域	名称	描述
31:11	保留	必须保持复位值
10:6	TXFNUM[4:0]	Tx FIFO数目 当本寄存器中TXFF控制位置位时, 该标志位决定那个Tx FIFO会被冲刷 主机模式: 00000: 仅非周期性Tx FIFO被冲刷

00001: 仅周期性Tx FIFO被冲刷

1xxxx: 周期性和非周期性Tx FIFO均被冲刷

其他: 没有数据被冲刷

设备模式:

00000: 仅Tx FIFO0被冲刷

00001: 仅Tx FIFO1被冲刷

...

00011: 仅Tx FIFO3被冲刷

1XXXX: 所有的Tx FIFO均被冲刷

其他: 没有数据被冲刷

5	TXFF	Tx FIFO冲刷控制位 应用通过置位该控制位来冲刷Tx FIFO数据，并且TXFNUM[4:0]决定冲刷的FIFO数目。当冲刷完成后，硬件自动清除该控制位。置位该控制位后，应用应该等待该控制位清除，并且，在此之前USBFS不应有其他操作。 注意： 在设备和主机模式下，均可访问
4	RXFF	Rx FIFO冲刷控制位 应用通过置位该控制位来冲刷Rx FIFO数据。当冲刷完成后，硬件自动清除该控制位。置位该控制位后，应用应该等待该控制位清除，并且，在此之前USBFS不应有其他操作。 注意： 在设备和主机模式下，均可访问
3	保留	必须保持复位值
2	HFCRST	主机帧计数器复位 应用通过置位该控制位来复位USBFS内的帧计数器。该控制位置位后，接下来SOF的帧计数器将变为0。当复位操作完成后，硬件自动清除该控制位。置位该控制位后，应用应该等待该控制位清除，并且，在此之前USBFS不应有其他操作。 注意： 仅在主机模式下访问
1	HCSRST	HCLK软件复位 应用通过置位该控制位来复位ABH时钟域电路 在复位操作完成后，硬件自动清除该控制位。置位该控制位后，应用应该等待该控制位清除，并且，在此之前USBFS不应有其他操作。 注意： 在设备和主机模式下，均可访问
0	CSRST	USB内核软件复位 复位AHB和USB时钟域电路，以及大多数的寄存器。

全局中断标志寄存器 (USBFS_GINTF)

地址偏移: 0x0014

复位值: 0x0400 0021

该寄存器只能按字（32位）访问

WKUPIF	SESIF	DISCIF	IDPSC	保留	PTXFEIF	HCIF	HPIF	保留	PXNCIF/ ISOONCF	ISOINCF	OEPIF	IEPIF	保留	
rc_w1	rc_w1	rc_w1	rc_w1		r	r	r		rc_w1	rc_w1	r	r		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	
EOPEIF	ISOOPDF	ENUMF	RST	SP	ESP	保留	GONAK	GNPINAK	NPTXFEIF	RXFNEIF	SOF	OTGIF	MIFIF	COPM
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1		r	r	r	r	rc_w1	r	rc_w1	r

位/位域	名称	描述
31	WKUPIF	唤醒中断标志位 当在USB总线上检测到一个恢复信号（在设备模式下）或者一个远程唤醒信号（在主机模式下），硬件将置位该中断标志位。 注意： 在设备和主机模式下，均可访问
30	SESIF	会话中断标志位 当在A设备模式下检测到一个SRP会话请求或在B设备模式下B设备的Vbus变为可用时，硬件将置位该中断标志位 注意： 在设备和主机模式下，均可访问
29	DISCIF	断开中断标志位 当设备断开后，将触发该标志位。 注意： 仅在主机模式下访问
28	IDPSC	ID引脚状态改变中断标志位 当ID引脚状态改变时，内核将置位该标志位 注意： 在设备和主机模式下，均可访问
27	保留	必须保持复位值
26	PTXFEIF	周期性Tx FIFO空中断标志位 当周期性发送FIFO半空或全空时，将触发该标志位。空阈值由USBFS_GAHBCS寄存器中周期性Tx FIFO空等级控制位（PTXFTH）决定。 注意： 仅在主机模式下访问
25	HCIF	主机通道中断标志位 当在主机模式下其中一个通道挂起一个中断时，USBFS将置位该标志位。软件应该首先读取USBFS_HACHINT寄存器以获取通道号，然后读取相应的USBFS_HCHxINTF寄存器以获取产生中断的通道标志位。当产生通道中断的独立通道标志位被清除后，该中断标志位将自动清除。 注意： 仅在主机模式下访问
24	HPIF	主机端口中断标志位 当USBFS在主机模式下检测到端口状态改变时，USB内核将置位该标志位。软件应

该读取USBFS_HPCSR寄存器以获取该中断源。当产生端口中断的标志被清除后，该中断标志位将自动清除。

注意：仅在主机模式下访问

23:22	保留	必须保持复位值
21	PXNCIF	周期性传输未完成中断标志位 在当前帧内，当帧结束时，周期性传输未完成，USBFS将置位该标志位（主机模式）。 同步OUT传输未完成中断标志位 在周期性帧结束时（由USBFS_DCFG寄存器的EOPFT控制位定义），如果仍有同步OUT端点未完成传输，USBFS将置位该标志位（设备模式）。
20	ISOINCIF	同步IN传输未完成中断标志位 在周期性帧结束时（由USBFS_DCFG寄存器的EOPFT控制位定义），如果仍有同步IN端点未完成传输，USBFS将置位该标志位（设备模式）。 注意： 仅在设备模式下访问
19	OEIF	OUT端点中断标志位 当在设备模式下，其中一个OUT端点挂起一个中断时，USBFS将置位该中断标志位。软件应该首先读取USBFS_DAEPINT寄存器以获取设备号，然后读取相应的USBFS_DOEPxINTF寄存器以获取产生中断的端点标志位。当产生中断的相应端点标志位被清除后，该中断标志位被自动清除。 注意： 仅在设备模式下访问
18	IEIF	IN端点中断标志位 当在设备模式下，其中一个IN端点挂起一个中断时，USBFS将置位该标志位。软件应该首先读取USBFS_DAEPINT寄存器以获取设备号，然后读取相应的USBFS_DIEPxINTF寄存器以获取产生中断的端点标志位。当相应产生中断的端点标志位被清除后，该中断标志位被自动清除。
17:16	保留	必须保持复位值
15	EOPIF	周期性帧结束中断标志位 当一帧内USB总线时间已经达到USBFS_DCFG寄存器中EOPFT控制位所定义的数值时，USBFS将置位该中断标志位。 注意： 仅在设备模式下访问
14	ISOOPDIF	同步OUT包丢失中断标志位 如果USBFS接收到一个同步OUT包，但是Rx FIFO没有足够的空间来接收该OUT包，USBFS将置位该标志位。 注意： 仅在设备模式下访问
13	ENUMF	枚举完成中断标志位 在速度枚举完成后，USBFS将置位该中断标志位。软件能够读取USBFS_DSTAT寄存器以获取当前设备速度。 注意： 仅在设备模式下访问
12	RST	USB复位中断标志位 当USBFS在USB总线上检测到一个USB复位信号后，USBFS将置位该中断标志位。

注意：仅在设备模式下访问

11	SP	USB挂起中断标志位 当USBFS检测到USB总线空闲3ms并且进入挂起状态， USBFS将置位该中断标志位。 注意： 仅在设备模式下访问
10	ESP	早期挂起中断标志位 当USBFS检测到USB总线空闲3ms时， USBFS将置位该中断标志位。
9:8	保留	必须保持复位值
7	GONAK	全局OUT NAK有效标志位 软件能够向USBFS_DCTL寄存器的SGONAK控制位写1，并且USBFS将会在SGONAK写入有效后，置位GONAK标志位。 注意： 仅在设备模式下可访问
6	GNPINAK	全局非周期性IN NAK有效标志位 软件能够向USBFS_DCTL寄存器中的SGINAK控制位写1，并且USBFS将会在SGINAK写入有效后，置位GNPINAK标志位 注意： 仅在设备模式下可访问
5	NPTXFEIF	非周期性Tx FIFO空中断标志位 当非周期性Tx FIFO为半空或全空时，将置位该中断标志位。该阈值由USBFS_GAHBCS寄存器中的非周期Tx FIFO空等级控制位(TXFTH)决定。 注意： 仅在主机模式下访问
4	RXFNEIF	Rx FIFO非空中断标志位 当至少有一个包或状态条目在Rx FIFO中时，USBFS将置位该标志位。 注意： 在主机和设备模式下，均可访问
3	SOF	帧起始中断标志位 主机模式： 当准备在USB总线上发送一个SOF或保持有效信号，USBFS将置位该中断标志位。 软件可以通过写1清除该中断标志位。 设备模式： 当USBFS接收到一个SOF令牌包后，USBFS置位该标志位。应用可以读取设备状态寄存器以获取当前帧号。软件可以通过写1清除该中断标志位。 注意： 在设备和主机模式下，均可访问
2	OTGIF	OTG中断标志位 当USBFS_GOTGINTE寄存器中标志位产生一个中断时，USBFS置位该中断标志位。软件应该读取USBFS_GOTGINTE寄存器以获取产生该中断的信号源，当USBFS_GOTGINTE寄存器中产生该中断的标志位被清除后，该中断标志位也被自动清除。 注意： 在设备和主机模式下，均可访问
1	MFIF	模式错误中断标志位 如果软件在设备模式下操作仅主机可访问的寄存器或者在主机模式下操作仅设备

可访问的寄存器，USBFS将置位该中断标志位。这些错误操作不会产生作用。

注意：在主机和设备模式下，均可访问

0 COPM 当前操作模式

0: 设备模式

1: 主机模式

注意：在主机和设备模式下，均可访问

全局中断使能寄存器 (USBFS_GINTEN)

地址偏移: 0x0018

复位值: 0x0000 0000

这个寄存器同全局中断标志寄存器 (USBFS_GINTF) 一起工作来中断应用程序。当中断使能位被禁止后，相应的中断就不会产生。然而，相应的全局中断标志位依然会被置位。

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WKUPIE	SEIE	DISCIE	IDPSCLIE	保留	PTXFIE	HCEIE	HPIE	保留	ISOONCIE	PXNCIE/	ISOINCIE	OEPIE	IEPIE	保留	保留
rw	rw	rw	rw		rw	rw	r		rw	rw	rw	rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOPFIE	ISOOPDIE	ENUMFIE	RSTIE	SPIE	ESPIE	保留	GONAKIE	GPNPINAKIE	NPTXFIE	RXFNIE	SOFIE	OTGIE	MFIE	保留	保留
rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw		

位/位域	名称	描述
31	WKUPIE	唤醒中断使能 0: 禁用唤醒中断 1: 使能唤醒中断 注意： 在主机和设备模式下，均可访问
30	SEIE	会话中断使能 0: 禁用会话中断 1: 使能会话中断 注意： 在主机和设备模式下，均可访问
29	DISCIE	断开中断使能 0: 禁用断开中断 1: 使能断开中断 注意： 仅在设备模式下使用

28	IDPSCIE	ID引脚状态改变中断使能 0: 禁用连接器ID引脚状态中断 1: 使能连接器ID引脚状态中断 注意: 在主机和设备模式下, 均可访问
27	保留	必须保持复位值
26	PTXFEIE	周期性Tx FIFO空中断使能 0: 禁用周期性Tx FIFO空中断 1: 使能周期性Tx FIFO空中断 注意: 仅在主机模式下访问
25	HCIE	主机通道中断使能 0: 禁用主机通道中断 1: 使能主机通道中断 注意: 仅在主机模式下访问
24	HPIE	主机端口中断使能 0: 禁止主机端口中断 1: 使能主机端口中断 注意: 仅在主机模式下访问
23:22	保留	必须保持复位值
21	PXNCIE	周期性传输未完成中断使能 0: 禁止周期性未完成传输中断 1: 使能周期性未完成传输中断 注意: 仅在主机模式下访问
	ISOONCIE	同步OUT传输未完成中断使能 0: 禁止同步OUT传输未完成中断 1: 使能同步OUT传输未完成中断 注意: 仅在设备模式下访问
20	ISOINCIE	同步IN传输未完成中断使能 0: 禁止同步IN传输未完成中断 1: 使能同步IN传输未完成中断 注意: 仅在设备模式下访问
19	OEPIE	OUT端点中断使能 0: 禁止OUT端点中断 1: 使能OUT端点中断 注意: 仅在设备模式下访问
18	IEPIE	IN端点中断使能 0: 禁止IN端点中断 1: 使能IN端点中断 注意: 仅在设备模式下访问

17:16	保留	必须保持复位值
15	EOPFIE	周期性帧结束中断使能 0: 禁止周期性帧结束中断 1: 使能周期性帧结束中断 注意: 仅在设备模式下访问
14	ISOOPDIE	同步OUT包丢失中断使能 0: 禁止同步OUT包丢失中断 1: 使能同步OUT包丢失中断 注意: 仅在设备模式下访问
13	ENUMFIE	枚举完成中断使能 0: 禁止枚举完成中断 1: 使能枚举完成中断 注意: 仅在设备模式下访问
12	RSTIE	USB复位中断使能 0: 禁止USB复位中断 1: 使能USB复位中断 注意: 仅在设备模式下访问
11	SPIE	USB挂起中断使能 0: 禁止USB挂起中断 1: 使能USB挂起中断 注意: 仅在设备模式下访问
10	ESPIE	早期挂起中断使能 0: 禁止早期挂起中断 1: 使能早期挂起中断 注意: 仅在设备模式下访问
9:8	保留	必须保持复位值
7	GONAKIE	全局OUT NAK有效中断使能 0: 禁止全局OUT NAK有效中断 1: 使能全局OUT NAK有效中断 注意: 仅在设备模式下访问
6	GNPINAKIE	全局非周期性IN NAK有效中断使能 0: 禁止全局非周期性IN NAK有效中断 1: 使能全局非周期性IN NAK有效中断 注意: 仅在设备模式下访问
5	NPTXFEIE	非周期性发送FIFO空中断使能 0: 禁止非周期性发送FIFO空中断 1: 使能非周期性发送FIFO空中断 注意: 仅在主机模式下访问

4	RXFNEIE	接收FIFO非空中断使能 0: 禁止接收FIFO非空中断 1: 使能接收FIFO非空中断 注意: 在设备模式与主机模式下, 均可访问
3	SOFIE	帧首中断使能 0: 禁止帧首中断 1: 使能帧首中断 注意: 在设备模式下与主机模式下, 均可访问
2	OTGIE	OTG中断使能 0: 禁止OTG中断 1: 使能OTG中断 注意: 在设备模式下与主机模式下, 均可访问
1	MFIE	模式错误中断使能 0: 禁止模式错误中断 1: 使能模式错误中断 注意: 在设备模式下与主机模式下, 均可访问
0	保留	必须保持复位值

全局接收状态读取 / 接收状态读取和弹出寄存器 (USBFS_GRSTATR/USBFS_GRSTATP)

读地址偏移: 0x001C

弹出地址偏移: 0x0020

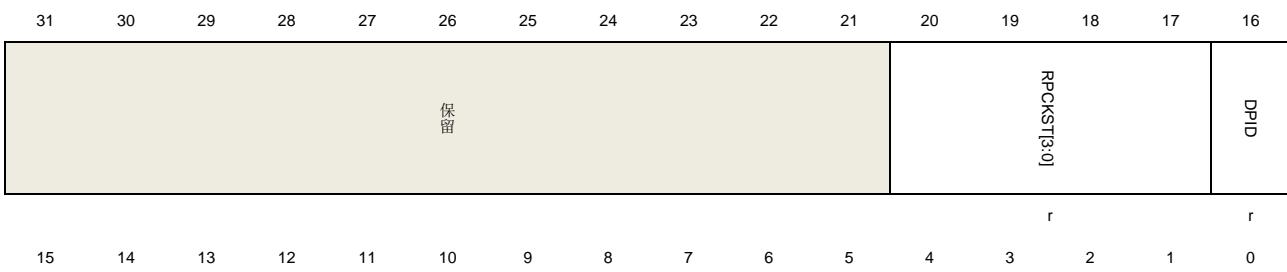
复位值: 0x0000 0000

对接收状态读寄存器的读操作, 将返回接收FIFO中顶部的条目。对接收状态读取和弹出寄存器的读操作, 将额外的弹出Rx FIFO的顶部条目。

在主机模式和设备模式下, Rx FIFO中的条目具有不同的含义。当全局中断标志寄存器 (USBFS_GINTF) 中的接收FIFO非空中断标志位 (RXFNEIF) 置位后, 软件应该读取该寄存器。

该寄存器只能按字(32位)访问

主机模式:



DPIID		BCOUNT[10:0]	CNUM[3:0]
-------	--	--------------	-----------

位/位域	名称	描述
31:21	保留	必须保持复位值
20:17	RPCKST[3:0]	接收包状态 0010: 接收到IN数据包 0011: IN传输完成（如果取出，触发一个中断） 0101: 数据翻转错误（如果取出，触发一个中断） 0111: 通道中止（如果取出，触发一个中断） 其他: 保留
16:15	DPIID[1:0]	数据PID 接收包的数据PID 00: DATA0 10: DATA1 01: DATA2 11: MDATA
14:4	BCOUNT[10:0]	字节数 接收IN数据包字节数。
3:0	CNUM[3:0]	通道数 当前接收包所属通道编号。

设备模式:

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

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

位/位域	名称	描述
31:21	保留	必须保持复位值

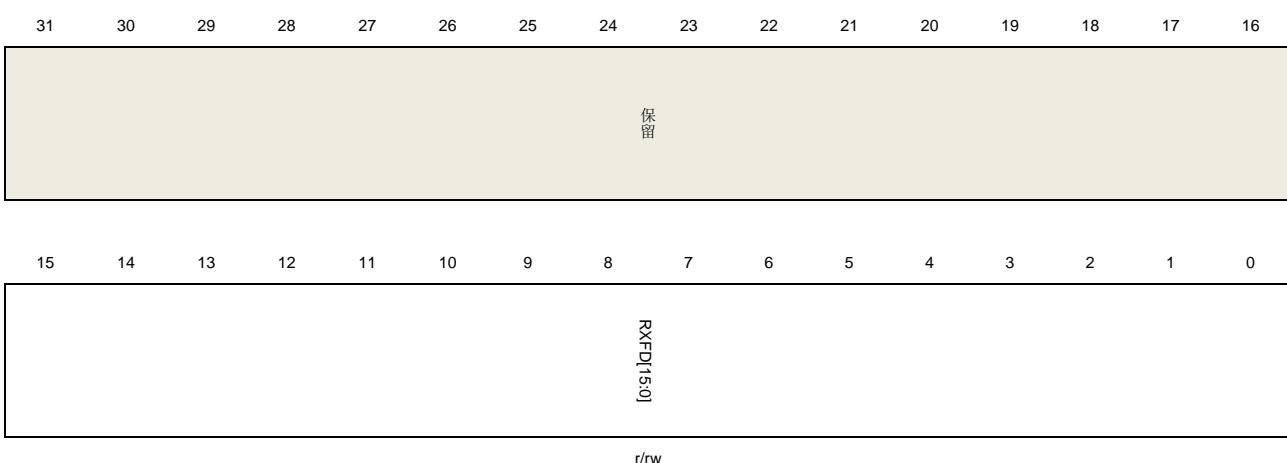
20:17	RPCKST[3:0]	接收包状态
		0001: 全局OUT NAK (产生一个中断)
		0010: 接收到OUT数据包
		0011: OUT传输完成 (产生一个中断)
		0100: SETUP传输完成 (产生一个中断)
		0110: 接收到SETUP数据包
		其他: 保留
16:15	DPID[1:0]	数据PID
		接收到OUT数据包的数据PID
		00: DATA0
		10: DATA1
		01: DATA2
		11: MDATA
14:4	BCOUNT[10:0]	字节数
		接收数据包的字节数
3:0	EPNUM[3:0]	端点号
		当前接收包所属端点编号

全局接收 FIFO 长度寄存器 (USBFS_GRFLEN)

地址偏移: 0x0024

复位值: 0x0000 0200

该寄存器只能按字 (32位) 访问



位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	RxFD[15:0]	Rx FIFO 深度 以32位字计数

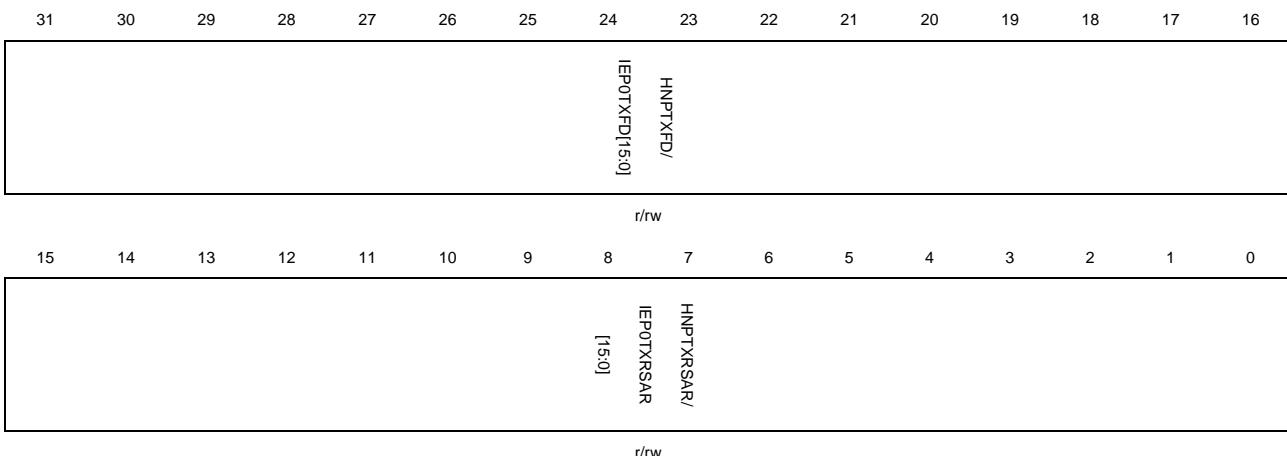
1≤RXFD≤1024

**主机非周期性发送 FIFO 长度寄存器/设备 IN 端点 0 发送 FIFO 长度寄存器
(USBFS_HNPTFLEN_DIEP0TFLEN)**

地址偏移: 0x0028

复位值: 0x0200 0200

该寄存器只能按字（32位）访问

**主机模式下:**

位/位域	名称	描述
31:16	HNPTXFD[15:0]	主机非周期性Tx FIFO深度 以32位字计数 1≤HNPTXFD≤1024
15:0	HNPTXRSAR[15:0]	主机非周期性Tx RAM起始地址 非周期性发送FIFO RAM的起始地址

设备模式下:

位/位域	名称	描述
31:16	IEP0TXFD[15:0]	输入端点0 Tx FIFO深度 以32位字计数 16≤IEP0TXFD≤140
15:0	IEP0TXRSAR[15:0]	输入端点0 TX RAM起始地址 端点0发送FIFO RAM的起始地址

主机非周期性发送 FIFO/队列状态寄存器 (USBFS_HNPTFQSTAT)

地址偏移: 0x002C

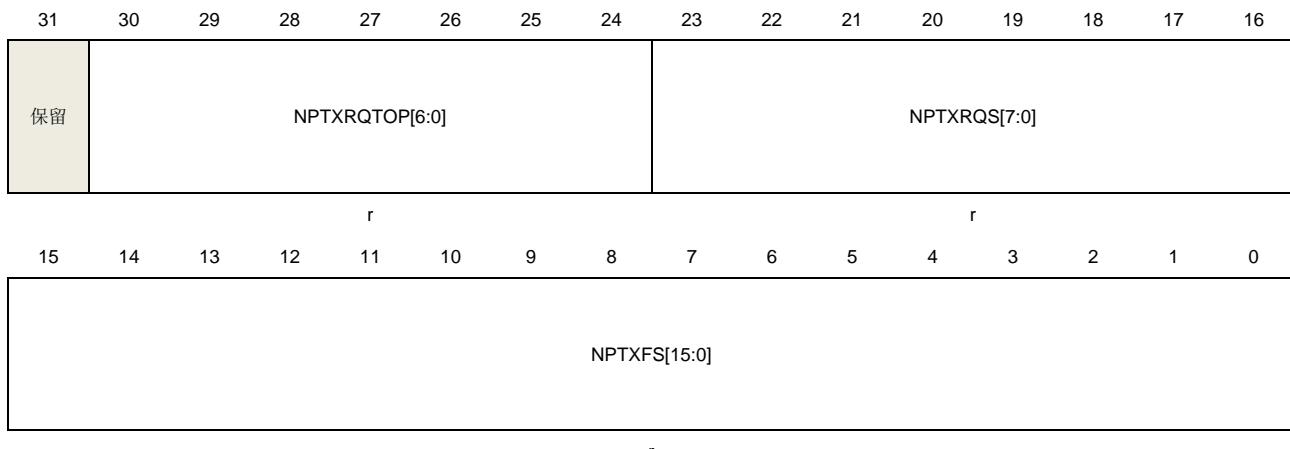
复位值: 0x0008 0200

该寄存器反映了非周期性Tx FIFO和请求队列的当前状态。

请求队列包括在主机模式下的IN、OUT或其他请求条目。

注意：在设备模式下，该寄存器不可用。

该寄存器只能按字（32位）访问



位/位域	名称	描述
31	保留	必须保持复位值
30:24	NPTXRQTOP[6:0]	非周期性发送请求队列的顶部条目 在非周期性传输请求队列中的条目。 位30:27: 通道号 位26:25: <ul style="list-style-type: none">- 00: IN/OUT令牌- 01: 0长度OUT包- 11: 通道中止请求 位24: 结束标志位, 表明所选通道的最后一个条目
23:16	NPTXRQS[7:0]	非周期性发送请求队列空间 非周期性请求队列的剩余空间 0: 请求队列空 1: 1个条目 2: 2个条目 ... n: n个条目 ($0 \leq n \leq 8$) 其他: 保留
15:0	NPTXFS[15:0]	非周期性Tx FIFO空间 非周期性发送FIFO剩余空间 以32位字计数 0: 非周期性Tx FIFO为空 1: 1个字 2: 2个字 n: n个字($0 \leq n \leq NPTXFD$)

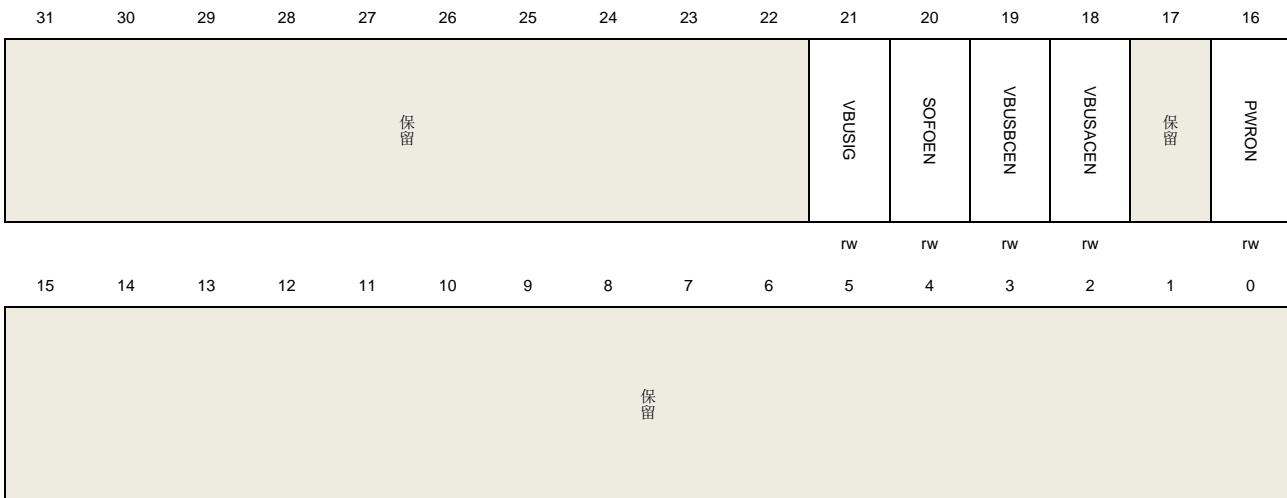
其他: 保留

全局内核配置寄存器 (USBFS_GCCFG)

地址偏移: 0x0038

复位值: 0x0000 0000

该寄存器只能按字（32位）访问



位/位域	名称	描述
31:22	保留	必须保持复位值
21	VBUSIG	V _{BUS} 忽略 当该控制位被置位，USBFS并不监测V _{BUS} 引脚电压，并且认为在主机和设备模式下，V _{BUS} 电压一直有效，然后可释放V _{BUS} 引脚作为其他用途。 0: V _{BUS} 不被忽略 1: V _{BUS} 被忽略，并认为V _{BUS} 电压一直有效
20	SOFOEN	SOF输出使能 0: SOF脉冲输出禁止 1: SOF脉冲输出使能
19	VBUSBCEN	V _{BUS} B设备比较器使能 0: V _{BUS} B设备比较器禁止 1: V _{BUS} B设备比较器使能
18	VBUSACEN	V _{BUS} A设备比较器使能 0: V _{BUS} A设备比较器禁止 1: V _{BUS} A设备比较器使能
17	保留	必须保持复位值
16	PWRON	上电

该控制位为内部嵌入式全速PHY的电源开关

0: 嵌入式全速PHY掉电

1: 嵌入式全速PHY上电

15:0 保留 必须保持复位值

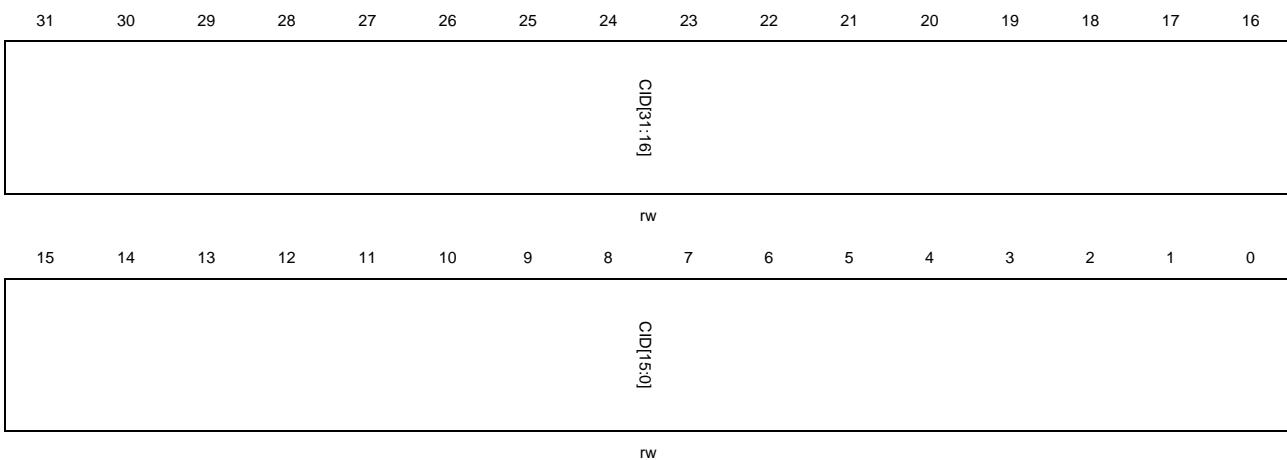
内核 ID 寄存器 (USBFS_CID)

地址偏移: 0x003C

复位值: 0x0000 1000

该寄存器包含产品ID

该寄存器只能按字（32位）访问



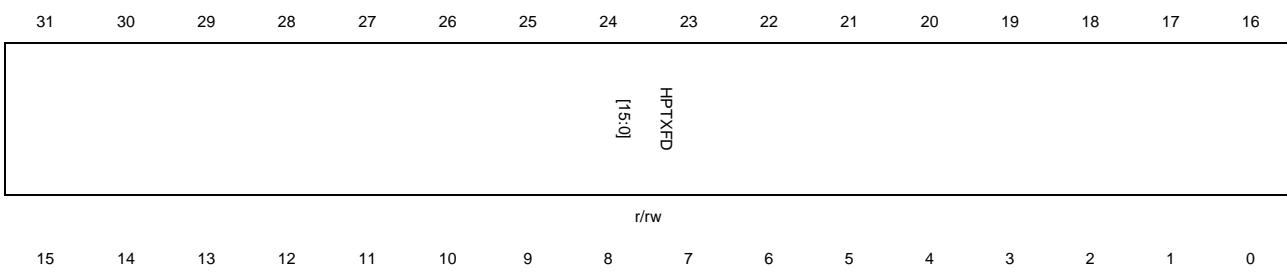
位/位域	名称	描述
31:0	CID[31:0]	内核ID 软件能够写入或读取该域值，并利用该域值为应用产生一个唯一ID。

主机周期性发送 FIFO 长度寄存器 (USBFS_HPTFLEN)

地址偏移: 0x0100

复位值: 0x0200 0600

该寄存器只能按字（32位）访问





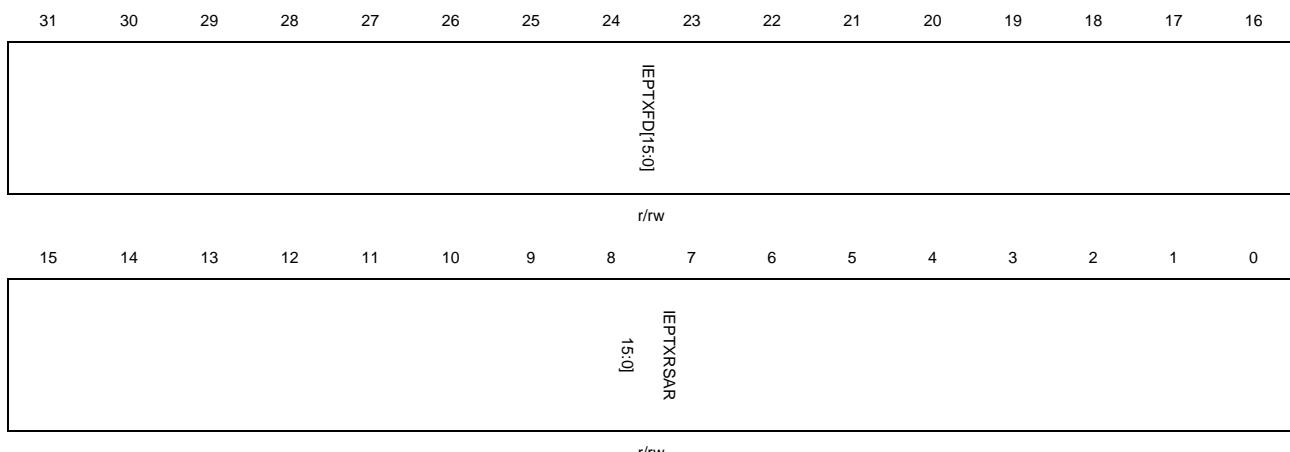
位/位域	名称	描述
31:16	HPTXFD[15:0]	主机周期性Tx FIFO深度 以32位字计数 $1 \leq HPTXFD \leq 1024$
15:0	HPTXFSAR[15:0]	主机周期性Tx RAM起始地址 主机周期性发送FIFO RAM起始地址

设备 IN 端点发送 FIFO 长度寄存器 (USBFS_DIEPxTFLEN**) (x = 1..3, 其中 x 为 FIFO 编号)**

地址偏移: $0x0104 + (\text{FIFO_number} - 1) \times 0x04$

复位值: 0x0200 0400

该寄存器只能按字（32位）访问



位/位域	名称	描述
31:16	IEPTXFD[15:0]	IN端点Tx FIFO深度 以32位字计数 $1 \leq IEPTXFD \leq 1024$
15:0	IEPTXRSAR[15:0]	IN端点FIFOx Tx RAM起始地址 以32位字为单位的IN端点发送FIFOx起始地址

23.7.2. 主机控制和状态寄存器

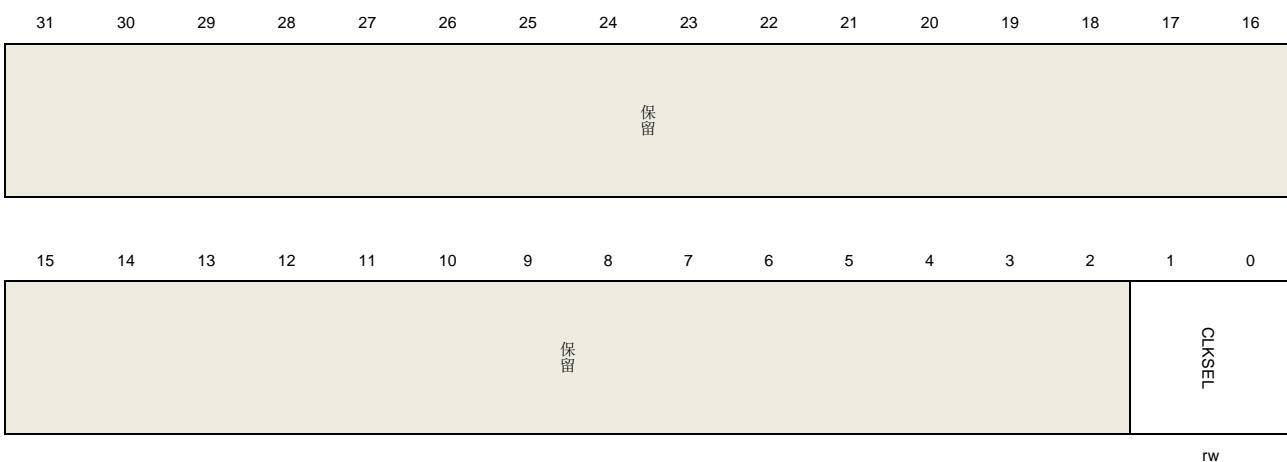
主机控制寄存器 (USBFS_HCTL)

地址偏移: 0x0400

复位值: 0x0000 0000

在主机模式下, 上电后, 该寄存器有USB内核配置。主机初始化后, 无需修改。

该寄存器只能按字 (32位) 访问



位/位域	名称	描述
31:2	保留	必须保持复位值
1:0	CLKSEL[1:0]	USB时钟选择 01: 48MHz时钟 其他: 保留

主机帧间隔寄存器 (USBFS_HFT)

地址偏移: 0x0404

复位值: 0x0000 BB80

当USBFS控制器正在枚举中时, 该寄存器为当前枚举速度设置帧间隔。

该寄存器只能按字 (32位) 访问





rw

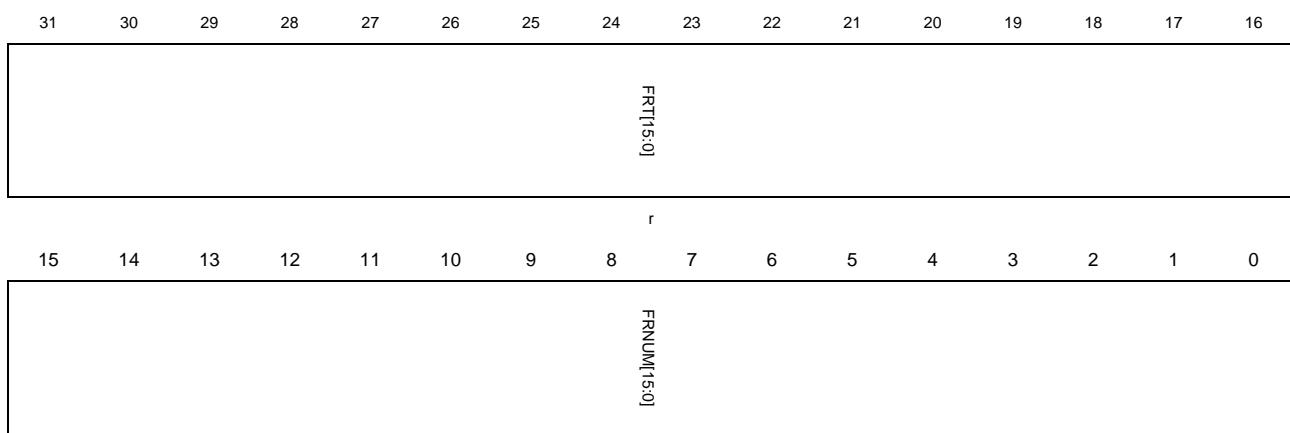
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	FRI[15:0]	<p>帧间隔</p> <p>该值描述了以PHY时钟为单位的帧周期。每次端口复位操作后，端口被使能，USBFS根据当前速度，采用一个固有值，并且软件可以向该位域写值以改变该固有值。该值需要采用以下描述的频率来进行计算：</p> <p>全速：48MHz</p> <p>低速：6MHz</p>

主机帧信息保持寄存器 (USBFS_HFINFR)

地址偏移: 0x0408

复位值: 0xBB80 0000

该寄存器只能按字（32位）访问



位/位域	名称	描述
31:16	FRT[15:0]	帧剩余时间
		该位域以PHY时钟为单位反映了当前帧剩余时间。
15:0	FRNUM[15:0]	帧号
		该位域反映了当前帧的帧号，当其增加到0x3FFF后，其值变为0。

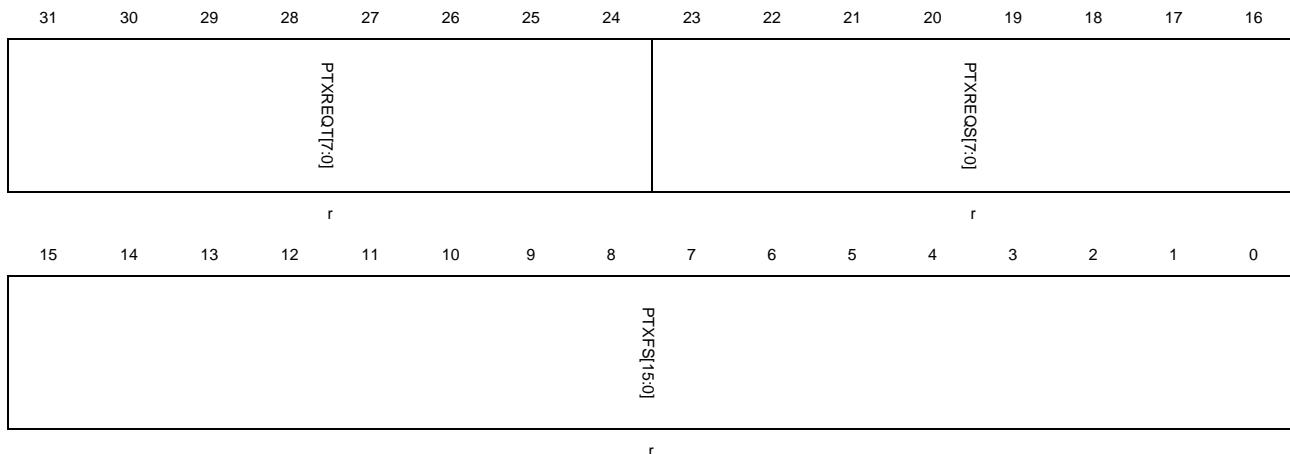
主机周期性发送 FIFO/队列状态寄存器 (USBFS_HPTFQSTAT)

地址偏移: 0x0410

复位值: 0x0008 0200

该寄存器反映了主机周期性Tx FIFO和请求队列的当前状态。请求队列包括在主机模式下的IN、OUT或其他请求条目。

该寄存器只能按字（32位）访问



位/位域	名称	描述
31:24	PTXREQT[7:0] 周期性Tx 请求队列的顶部条目	在周期性发送请求队列中的条目 位30:27: 通道号 位26:25: <ul style="list-style-type: none"> - 00: IN/OUT 令牌 - 01: 0长度OUT包 - 11: 通道中止请求 位24: 中止标志, 指示所选通道的最后一个条目
23:16	PTXREQS[7:0] 周期性发送请求队列空间	周期性发送请求队列剩余空间 0: 请求队列为空 1: 1个条目 2: 2个条目 ... n: n个条目 (0≤n≤8) 其他: 保留
15:0	PTXFS[15:0] 周期性发送FIFO空间	周期性发送FIFO剩余空间 以32位字计数 0: 周期性发送FIFO为空 1: 1个字 2: 2个字 n: n个字 (0≤n≤PTXFD))

其他：保留

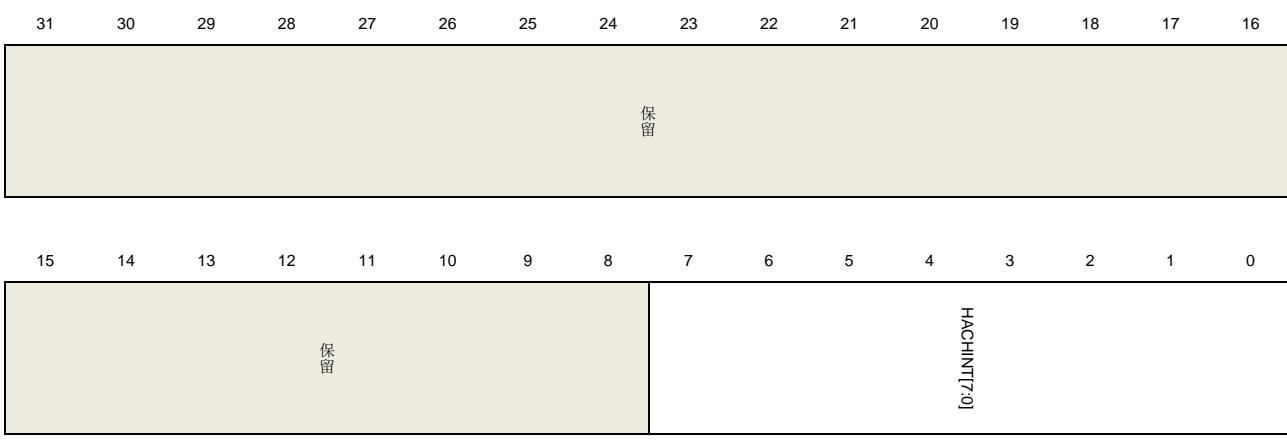
主机所有通道中断寄存器 (**USBFS_HACHINT**)

地址偏移: 0x0414

复位值: 0x0000 0000

当触发一个通道中断时，**USBFS**在该寄存器中置位相应的位，并且软件可以读取该寄存器以获取产生中断的通道。

该寄存器只能按字（32位）访问



位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	HACHINT[7:0]	主机所有通道中断 每一位表示一个通道：位0代表通道0，位7表示通道7

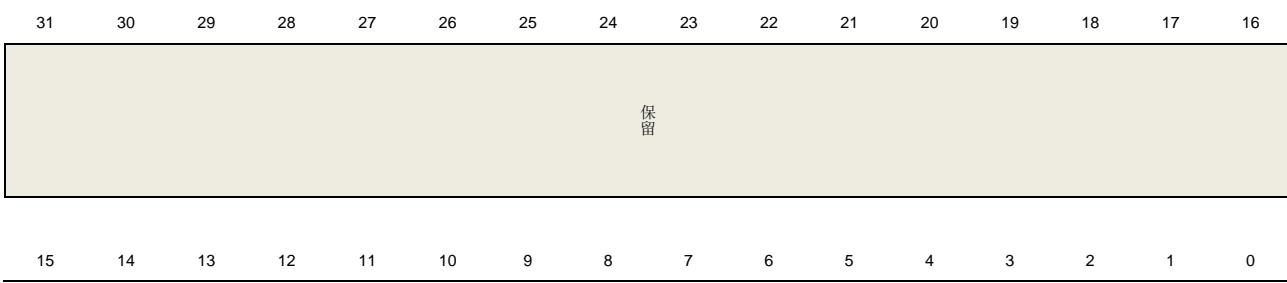
主机所有通道中断使能寄存器 (**USBFS_HACHINTEN**)

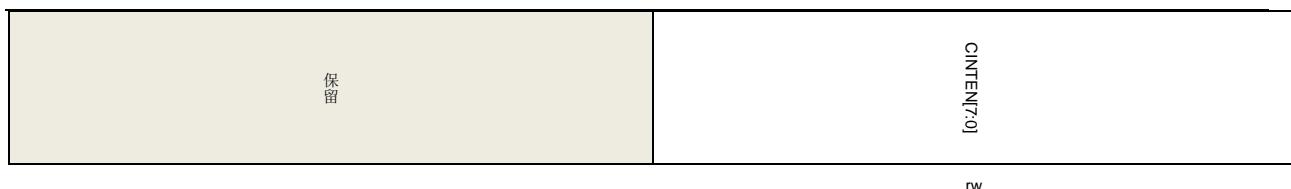
地址偏移: 0x0418

复位值: 0x0000 0000

软件可以使用该寄存器使能或禁用一个通道的中断。只有该寄存器中相应通道的中断使能控制位被置位，**USBFS_GINTF**寄存器中的通道中断标志位**HCIIF**标志位才可产生。

该寄存器只能按字（32位）访问





位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	CINTEN[7:0]	通道中断使能 0: 禁用通道n中断 1: 使能通道n中断 每一位表示一个通道: 位0代表通道0, 位7代表通道7

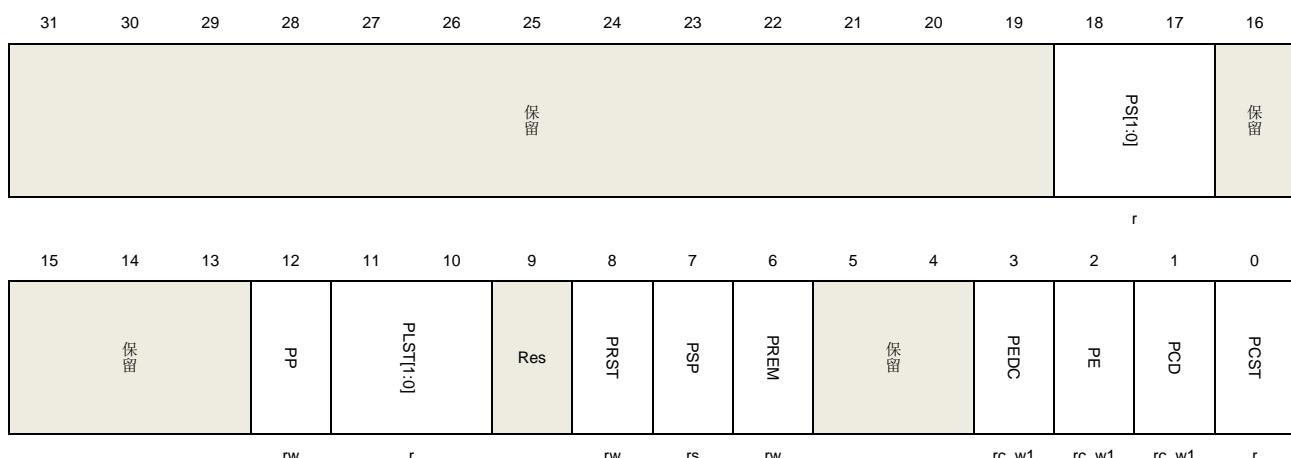
主机端口控制和状态寄存器 (USBFS_HPCS)

地址偏移: 0x0440

复位值: 0x0000 0000

该寄存器控制端口行为, 并且也包含一些反映端口状态的标志位。如果本寄存器中的PRST、PEDC和PCD标志位被USBFS置位的话, USBFS_GINTF寄存器中的HPIF标志位会被置位。

该寄存器只能按字 (32位) 访问



位/位域	名称	描述
31:19	保留	必须保持复位值
18:17	PS[1:0]	端口速度 反映连接到该端口的设备的枚举速度。 01: 全速 10: 低速 其他: 保留
16:13	保留	必须保持复位值

12	PP	端口供电 在端口被使用后，该控制位应该被置位。由于 USBFS 不具有电源供应能力，它只能使用该控制位以获取该端口是否在供电状态。软件应该在设置该控制位之前，保证在 V_{Bus} 引脚上具有电源供应。 0: 端口掉电 1: 端口供电
11:10	PLST[1:0]	端口线状态 反映 USB 数据线当前状态 位10: DP线状态 位11: DM线状态
9	保留	必须保持复位值
8	PRST	端口复位 应用通过设置该控制位以在 USB 端口上启动一个复位信号。当应用希望停止复位信号时，应用应该清除该控制位。 0: 端口不在复位状态 1: 端口处于复位状态
7	PSP	端口挂起 应用设置该控制位来将端口进入挂起状态。当该控制位被置位后，端口停止发送 SOF 令牌包。该控制位只能通过以下操作清除。 <ul style="list-style-type: none">- 应用置位该寄存器中的PRST控制位- 置位该寄存器中的PREM控制位- 检测到一个远程唤醒信号- 检测到一个设备断开 0: 端口不在挂起状态 1: 端口处于挂起状态
6	PREM	端口恢复 应用通过置位该控制位以在 USB 端口上启动一个恢复信号。当应用希望停止恢复信号时，应用可以清除该控制位。 0: 无恢复驱动 1: 恢复驱动
5:4	保留	必须保持复位值
3	PEDC	端口使能/禁止更改 当该寄存器中的位2端口使能控制位更改时， USB 内核置位该标志位。
2	PE	端口使能 当 USB 复位信号完成后， USBFS 自动置位该位，并且该位不可由软件置位。 该位可通过以下事件清除： <ul style="list-style-type: none">- 一个断开状态- 软件清除该位 0: 端口禁止

1: 端口使能

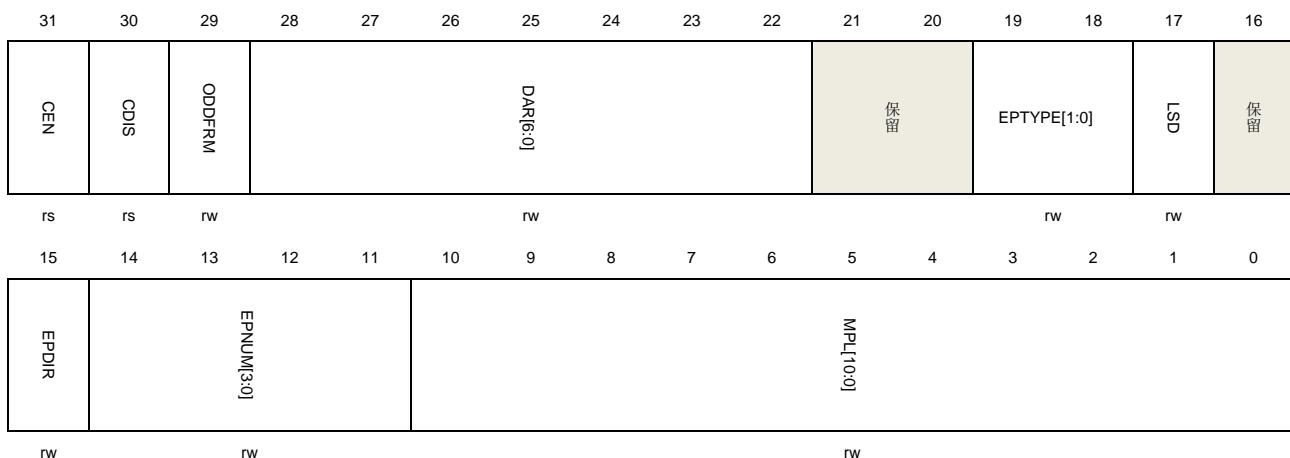
1	PCD	端口连接检测 当检测到设备连接时，USBFS置位该标志位。可通过向该位写1清除该标志位。
0	PCST	端口连接状态 0: 设备没有连接到该端口 1: 设备连接到该端口

主机通道 x 控制寄存器 (USBFS_HCHxCTL) (x = 0...7, 其中 x 为通道号)

地址偏移: 0x0500 + (通道号 × 0x20)

复位值: 0x0000 0000

该寄存器只能按字（32位）访问



位/位域	名称	描述
31	CEN	通道使能 由应用设置，并且由USBFS清除 0: 通道禁止 1: 通道使能 软件应该遵循操作指南来禁用或者使能一个通道
30	CDIS	通道禁止 软件可以置位该控制位，来从处理事务中禁用该通道。软件应该遵循操作指南来禁用或者使能一个通道。
29	ODDFRM	奇偶帧控制 对于周期性传输（中断或同步传输），该位控制将要处理的通道事务为奇数帧还是偶数帧。
28:22	DAR[6:0]	设备地址 与该通道通信的USB设备地址。

21:20	保留	必须保持复位值
19:18	EPTYPE[1:0]	端点类型 与该通道通信的端点的传输类型 00: 控制 01: 同步 10: 批量 11: 中断
17	LSD	低速设备 与该通道通信的设备是一个低速设备。
16	保留	必须保持复位值
15	EPDIR	端点方向 与该通道通信的端点的传输方向 0: OUT 1: IN
14:11	EPNUM[3:0]	端点号 与该通道通信的端点号
10:0	MPL[10:0]	最大包长 目标端点的最大包长

主机通道 x 中断标志寄存器 (USBFS_HCHxINTF) (x = 0...7, 其中 x = 通道号)

地址偏移: 0x0508 + (通道号 x 0x20)

复位值: 0x0000 0000

该寄存器包含一个通道的状态和事件，当软件获取一个通道中断时，软件需要为相应通道读取该寄存器以获取产生中断的中断源。该寄存器中的标志位均由硬件置位，并且写1清除。

该寄存器只能按字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
保留				DTER	REQOVR	BBER	UBER	保留		ACK	NAK	STALL	保留		CH	TF
rc_w1				rc_w1	rc_w1	rc_w1	rc_w1	rc_w1		rc_w1	rc_w1	rc_w1	rc_w1		rc_w1	

位/位域	名称	描述
31:11	保留	必须保持复位值
10	DTER	数据切换错误 IN事务获取一个数据包，但是该包的PID和USBFS_HCHxLEN寄存器中的DPID[1:0]控制位不匹配。
9	REQOVR	请求队列上溢 当软件启动新的传输时，请求队列上溢。
8	BBER	串扰错误 USB总线上发生一个串扰事件。产生串扰事件的典型原因是端点发送了一个数据包，但是数据包长度超过了端点的最大包长。
7	USBER	USB总线错误 当在接收一个数据包的过程中，发生以下事件时，将置位USB总线错误标志位： – 接收包有一个错误的CRC域 – 在USB总线上检测到填充错误 – 当等待一个响应包时，超时
6	保留	必须保持复位值
5	ACK	ACK 接收或者发送一个ACK响应包
4	NAK	NAK 接收到一个NAK响应包
3	STALL	STALL 接收到一个STALL响应包
2	保留	必须保持复位值
1	CH	通道中止 通道被当前请求所禁用，在当前请求处理的过程中，并不响应其他请求处理。
0	TF	发送完成 该通道所有的事务成功完成并且无错误发生。 对于IN通道，在USBFS_HCHxLEN寄存器的PCNT位减到0后，该标志位被置位。 对于OUT通道，当软件从Rx FIFO中读取和取出一个TF状态条目时，该标志位被置位。

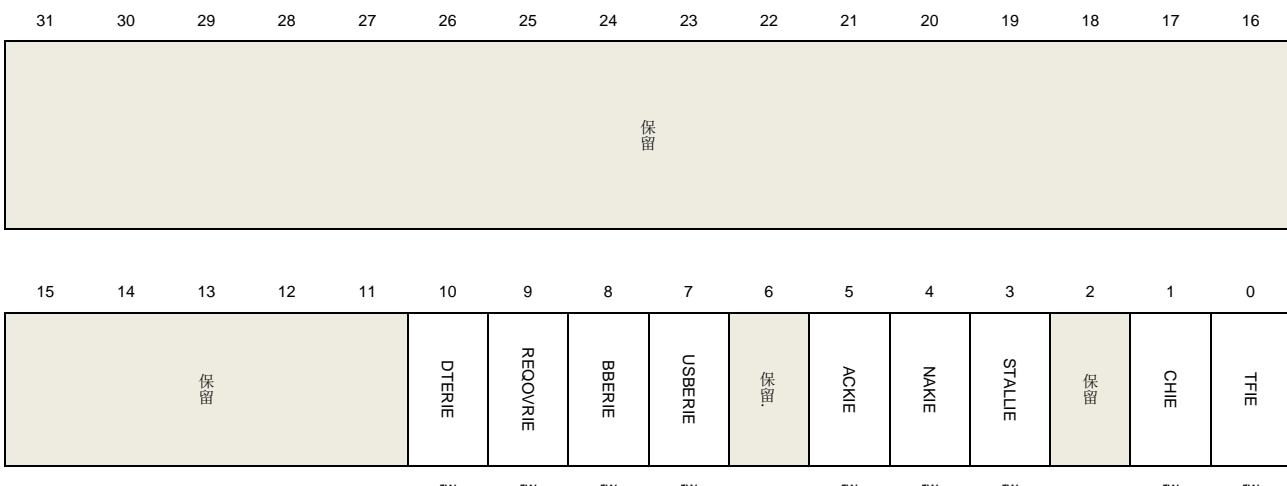
主机通道 x 中断使能寄存器 (USBFS_HCHxINTEN) (x = 0...7, 其中 x = 通道号)

地址偏移: 0x050C + (通道号 x 0x20)

复位值: 0x0000 0000

该寄存器包含USBFS_HCHxINTF寄存器内中断标志位的中断使能位。如果该寄存器的某位被软件置位，USBFS_HCHxINTF寄存器内的相应位能够触发一个通道中断。该寄存器内的位可由软件置位和清除。

该寄存器只能按字（32位）访问



位/位域	名称	描述
31:11	保留	必须保持复位值
10	DTERIE	数据切换错误中断使能 0: 禁用数据切换错误中断 1: 使能数据切换错误中断
9	REQOVRIE	请求队列上溢中断使能 0: 禁用请求队列上溢中断 1: 使能请求队列上溢中断
8	BBERIE	串扰错误中断使能 0: 禁用串扰错误中断 1: 使能串扰错误中断
7	USBERIE	USB总线错误中断使能 0: 禁用USB总线错误中断 1: 使能USB总线错误中断
6	保留	必须保持复位值
5	ACKIE	ACK中断使能 0: 禁用ACK中断 1: 使能ACK中断
4	NAKIE	NAK中断使能 0: 禁用NAK中断 1: 使能NAK中断
3	STALLIE	STALL中断使能 0: 禁用STALL中断 1: 使能STALL中断

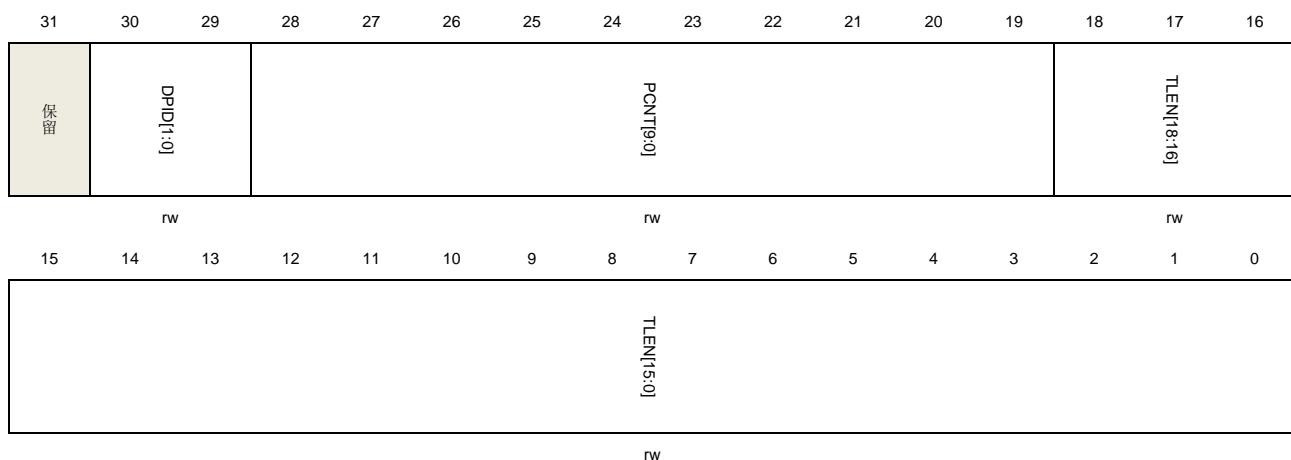
2	保留	必须保持复位值
1	CHIE	通道中止中断使能 0: 禁用通道中止中断 1: 使能通道中止中断
0	TFIE	传输完成中断使能 0: 禁用传输完成中断 1: 使能传输完成中断

主机通道 x 长度寄存器 (**USBFS_HCHxLEN**) (x = 0...7, 其中 x = 通道号)

地址偏移: 0x0510 + (通道号 × 0x20)

复位值: 0x0000 0000

该寄存器只能按字（32位）访问



位/位域	名称	描述
31	保留	必须保持复位值
30:29	DPID[1:0]	数据PID 软件应该在传输起始之前写该段位域。对于OUT传输，该位域包含第一个传输包的数据PID。对于IN传输，该位域包含第一个接收包的数据PID，并且如果数据PID不匹配的话，将会触发DTER标志位。在传输开始之后，USBFS遵循USB协议自动改变和切换该位域。 00: DATA0 10: DATA1 11: SETUP (仅对于控制传输) 01: 保留
28:19	PCNT[9:0]	包计数 在一个传输中希望发送(OUT)或接收(IN)的数据包个数。 软件应该在通道使能之前写该位域。在传输启动之后，该位域在USBFS正确传输每个数

据包后，自动减少。

18:0	TLEN[18:0]	传输长度 一次传输的总数据字节数。 对于OUT传输，该位域为OUT传输中期望发送的所有数据包总数据字节数。软件应该在通道使能之前写该位域。当软件或DMA正确向通道的数据FIFO中写入一个包时，该位域以包中字节大小进行减少。 对于IN传输，每次软件或DMA从Rx FIFO中读取一个包后，该位域也以包中字节大小进行减少。
------	------------	--

23.7.3. 设备控制和状态寄存器

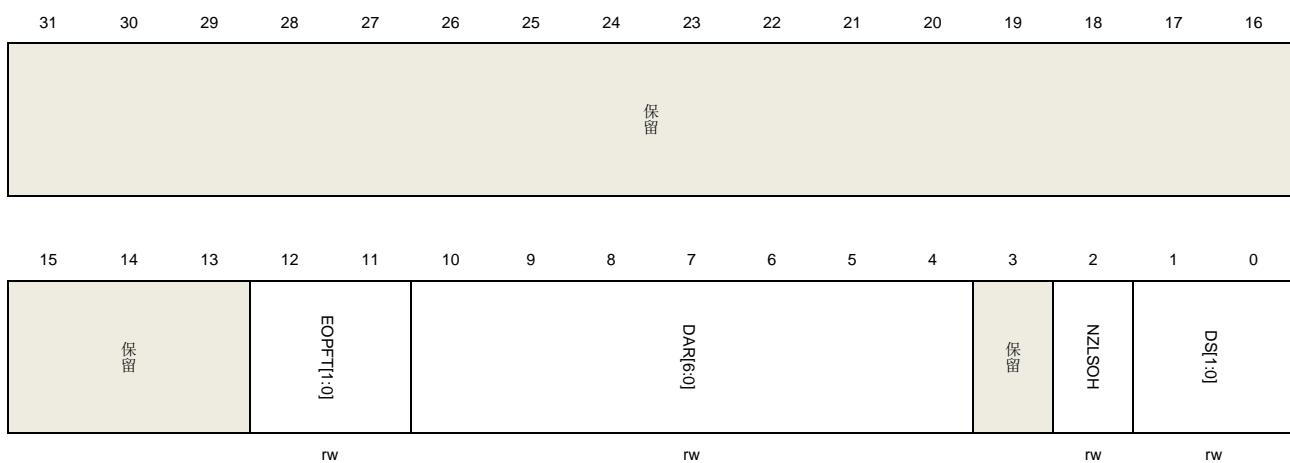
设备配置寄存器 (USBFS_DCFG)

地址偏移: 0x0800

复位值: 0x0000 0000

在上电、枚举或执行某些控制命令后，该寄存器配置内核为设备模式。在设备初始化后，不可以改变该寄存器值。

该寄存器采用字（32位）访问



位/位域	名称	描述
31:13	保留	必须保留为复位值
12:11	EOPFT[1:0]	周期性帧尾时间 该域定义周期性帧时间的帧尾标志触发的时间点 00: 80%的帧时间 01: 85%的帧时间 10: 90%的帧时间 11: 95%的帧时间
10:4	DAR[6:0]	设备地址

该位定义USB设备地址，**USBFS**采用该位匹配接收的设备令牌地址域，在接收到来自主机的设置地址的命令后，软件设置该域

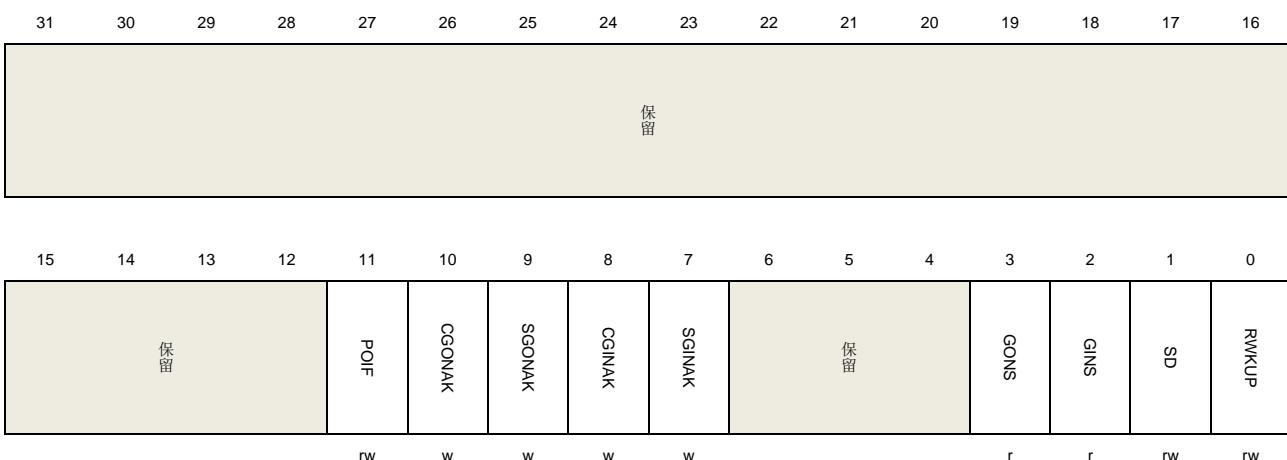
3	保留	必须保留为复位值
2	NZLSOH	非零长度OUT状态阶段握手 在控制传输的OUT状态阶段，当 USB 设备接收到一个非零长度数据包时，该域控制 USBFS 是接收该包，还是用 STALL 握手信号拒绝该包。 0: 把该包视为正常包，根据设备OUT端点控制寄存器的 NAKS 和 STALL 位，回复握手相应握手包 1: 发送 STALL 握手，不保存接收到的OUT数据包
1:0	DS[1:0]	设备速度 该域控制设备连入主机后的设备速度 11: 全速 其他: 保留

设备控制寄存器 (**USBFS_DCTL**)

地址偏移: 0x0804

复位值: 0x0000 0000

该寄存器采用字（32位）访问



位/位域	名称	描述
31:12	保留	必须保留为复位值
11	POIF	上电初始化完成 软件通过设置该位，通知 USBFS 寄存器在从掉电模式下唤醒，然后完成初始化。
10	CGONAK	清零全局OUT NAK 软件设置该位从而清零该寄存器的 GONS 位
9	SGONAK	设置全局OUT NAK 软件设置该位从而实现该寄存器的位 GONS 置位。

当GONS位为零，设置该位会引起USBFS_GINTF寄存器的GONAK标志触发，软件应该在再写该位前清除GONAK标志。

8	CGINAK	清零全局IN NAK 软件设置该位从而清零该寄存器的GINS位
7	SGINAK	设置全局IN NAK 软件设置该位从而实现该寄存器的位GINS置位 当GINS位为零, 设置该位会引起USBFS_GINTF寄存器的GINAK标志触发, 软件应该在再写该位前清除GINAK标志。
6:4	保留	必须保留为复位值
3	GONS	全局OUT NAK状态 0: USBFS回复OUT事务的握手信号以及是否保存OUT数据包由Rx FIFO状态、端点的NAKS、STALL位确定。 1: USBFS回复OUT事务NAK握手信号, 不保存接收的OUT数据包。
2	GINS	全局IN NAK状态 0: USBFS回复IN事务的握手信号由Tx FIFO状态、端点的NAKS、STALL位确定。 1: USBFS通常回复IN事务NAK握手信号
1	SD	软断开 软件可实现USB总线上的软断开, 在置1该位后, 关掉DP线上的上拉电阻, 从而引起主机检测设备的断开。 0: 没有软断开生成 1: 生成软断开
0	RWKUP	远程唤醒 在挂起状态, 软件可通过该位来生成一个远程唤醒信号来通知主机恢复USB总线 0: 没有远程唤醒信号生成 1: 生成远程唤醒信号

设备状态寄存器 (USBFS_DSTAT)

地址偏移: 0x0808

复位值: 0x0000 0000

该寄存器包含设备模式下的 USBFS 的状态和信息。

该寄存器采用字(32位)访问

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

FNRSOF[7:0]	保留	ES[1:0]	SPST
r	r	r	r

位/位域	名称	描述
31:22	保留	必须保留为复位值
21:8	FNRSOF[13:0]	所接收的SOF帧编号 USBFS会在接收到一个SOF令牌后更新该域。
7:3	保留	必须保留为复位值
2:1	ES[1:0]	枚举速度 该域指示所枚举的设备速度，在寄存器USBFS_GINTF的ENUMF标志触发后，软件可以读取该域。 01: 全速 其他: 保留
0	SPST	挂起状态 该位指示设备是否处于挂起状态。 0: 设备在挂起状态 1: 设备不在挂起状态

设备 IN 端点通用中断使能寄存器 (USBFS_DIEPINTEN)

地址偏移: 0x810

复位值: 0x0000 0000

该寄存器包含寄存器 USBFS_DIEPxINTF 中的标志的中断使能位，如果软件置 1 某位，其在寄存器 USBFS_DIEPxINTF 中对应的位可以触发一个寄存器 USBFS_DAEPINT 端点中断。该位可以通过软件置位和清零。

该寄存器采用字（32位）访问

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

位/位域	名称	描述
31:7	保留	必须保留为复位值
6	IEPNEEN	IN端点NAK有效中断使能位 0: 除能中断 1: 使能中断
5	保留	必须保留为复位值
4	EPTXFUDEN	端点Tx FIFO下溢中断使能位 0: 除能中断 1: 使能中断
3	CITOEN	控制IN事务超时中断使能位 0: 除能中断 1: 使能中断
2	保留	必须保留为复位值
1	EPDISEN	端点除能中断使能位 0: 除能中断 1: 使能中断
0	TFEN	传输完成中断使能位 0: 除能中断 1: 使能中断

设备 OUT 端点通用中断使能寄存器 (USBFS_DOEPINTEN)

地址偏移: 0x0814

复位值: 0x0000 0000

该寄存器包含寄存器 USBFS_DOEPxINTF 中的标志的中断使能位，如果软件置 1 某位，其在寄存器 USBFS_DOEPxINTF 中对应的位可以触发一个寄存器 USBFS_DAEPINT 端点中断。该位可以通过软件置位和清零。

该寄存器采用字（32位）访问

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

保留	BTBSTPEN	保留	EPRXFOVREN	STPFEN	保留	EPDISEN	TFEN
		rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:6	保留	必须保留为复位值
6	BTBSTPEN	连续SETUP包中断使能位（仅适用于控制OUT端点） 0: 除能中断 1: 使能中断
5	保留	必须保留为复位值
4	EPRXFOVREN	端点Rx FIFO上溢中断使能位 0: 除能中断 1: 使能中断
3	STPFEN	SETUP阶段完成中断使能位（仅适用于控制OUT端点） 0: 除能中断 1: 使能中断
2	保留	必须保留为复位值
1	EPDISEN	端点除能中断使能位 0: 除能中断 1: 使能中断
0	TFEN	传输完成中断使能位 0: 除能中断 1: 使能中断

设备端点中断寄存器 (USBFS_DAEPINT)

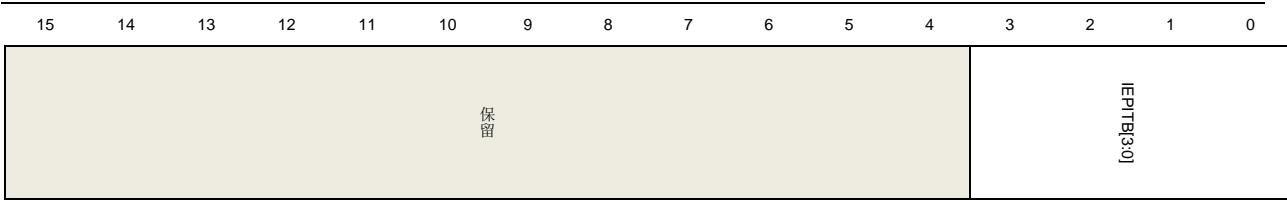
地址偏移: 0x0818

复位值: 0x0000 0000

当一个端点的中断被触发, USBFS 置 1 该寄存器的相应位, 软件可通过该寄存器知道在本次中断中的端点号。

该寄存器采用字 (32位) 访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留														OEPTRB[3:0]	



r

位/位域	名称	描述
31:20	保留	必须保留为复位值
19:16	OEPITB[3:0]	设备OUT端点中断位 每个位代表一个OUT端点: Bit16代表OUT端点0, Bit19代表OUT端点3
15:4	保留	必须保留为复位值
3:0	IEPITB[3:0]	设备IN端点中断位 每个位代表一个IN端点: Bit0代表IN端点0, Bit3代表IN端点3

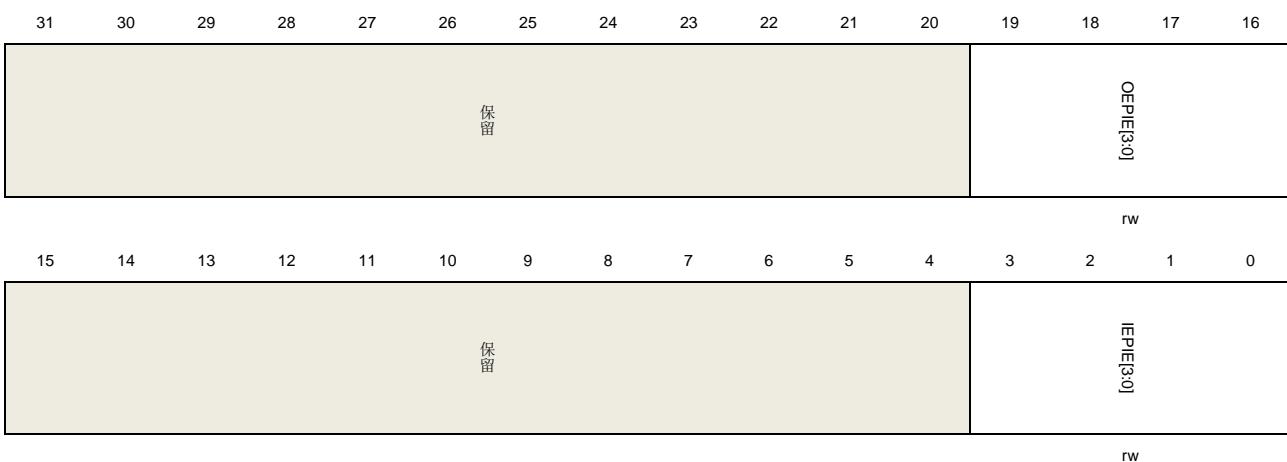
设备端点中断使能寄存器 (USBFS_DAEPISEN)

地址偏移: 0x081C

复位值: 0x0000 0000

该寄存器可通过软件使能或除能端点的中断，只有当端点在该寄存器中相应位被置 1 才能触发寄存器 USBFS_GINTF 的端点中断标志 OEPIF 或 IEPIF。

该寄存器采用字（32位）访问



rw

rw

位/位域	名称	描述
31:20	保留	必须保留为复位值
19:16	OEPIE[3:0]	OUT端点中断使能位 0: 除能OUT端点n中断 1: 使能OUT端点n中断

每个位代表一个OUT端点: Bit16对应OUT端点0, Bit19对应OUT端点3

15:4 保留 必须保留为复位值

3:0 IEPIE[3:0] IN端点中断使能位

0: 除能IN端点n中断

1: 使能IN端点n中断

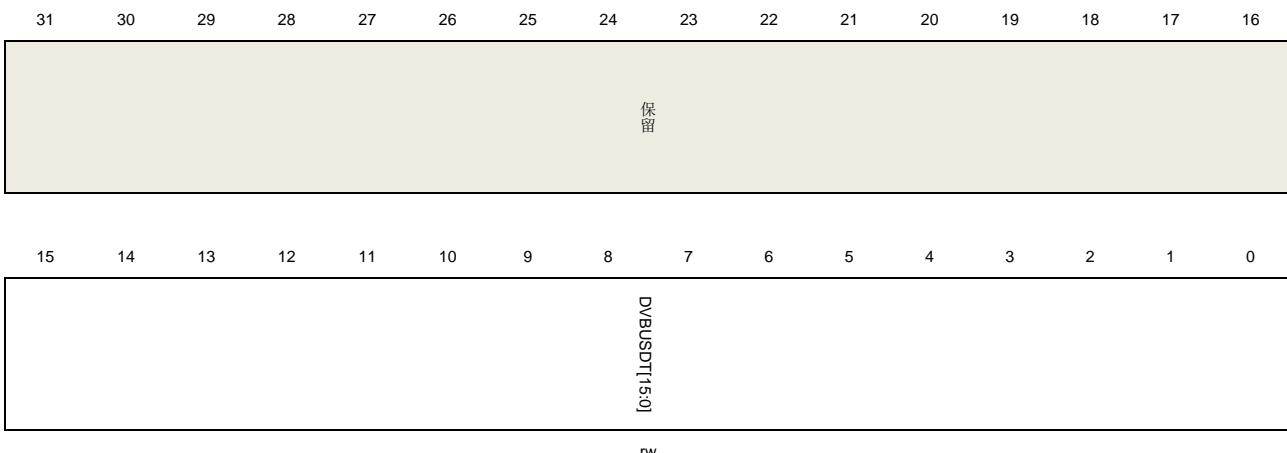
每个位代表一个IN端点: Bit0对应IN端点0, Bit3对应IN端点3

设备 VBUS 放电时间寄存器 (**USBFS_DVBUSDT**)

地址偏移: 0x0828

复位值: 0x0000 17D7

该寄存器采用字(32位)访问



位/位域	名称	描述
31:16	保留	必须保留为复位值
15:0	DVBUSDT[15:0] 设备V _{BUS} 放电时间	在SRP协议中, 在V _{BUS} 脉冲产生后, 有一个放电过程, 该域定义了V _{BUS} 的放电时间, 真正的放电时间是1024*DVBUSDT[15:0] *T _{USBCLOCK} , T _{USBCLOCK} 是USB时钟周期时间。

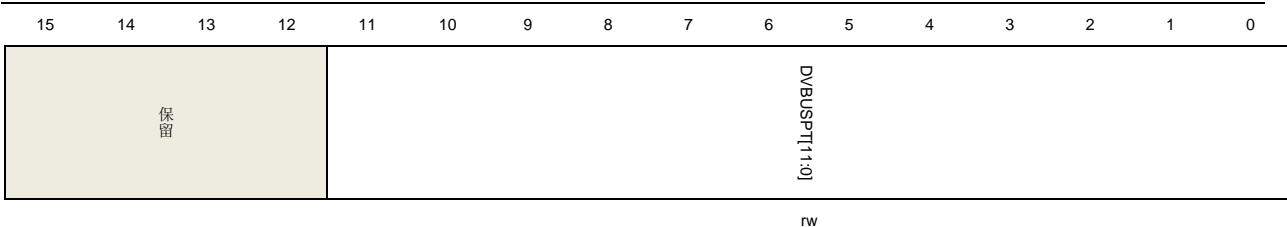
设备 VBUS 脉冲时间寄存器 (**USBFS_DVBUSPT**)

地址偏移: 0x082C

复位值: 0x0000 05B8

该寄存器采用字(32位)访问





位/位域	名称	描述
31:12	保留	必须保留为复位值
11:0	DVBUSPT[11:0]	设备V _{BUS} 脉冲时间 该域定义V _{BUS} 的脉冲时间，真正的充电时间是1024*DVBUSPT[15:0]*T _{USBCLOCK} ， T _{USBCLOCK} 是USB时钟周期时间

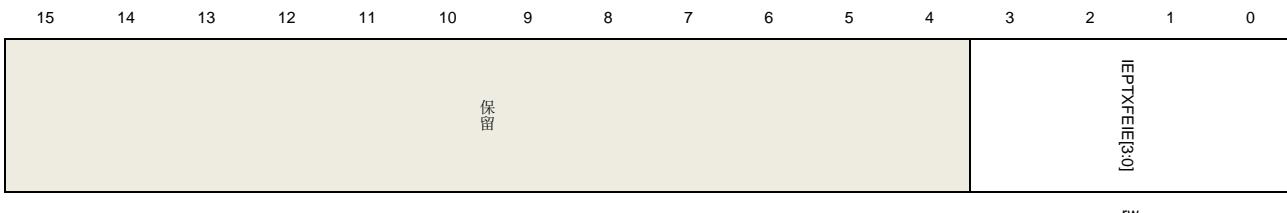
设备 IN 端点 FIFO 空中断使能寄存器 (**USBFS_DIEPFEINTEN**)

地址偏移: 0x0834

复位值: 0x0000 0000

该寄存器包含 IN 端点 Tx FIFO 空中断的使能位

寄存器采用字（32位）访问



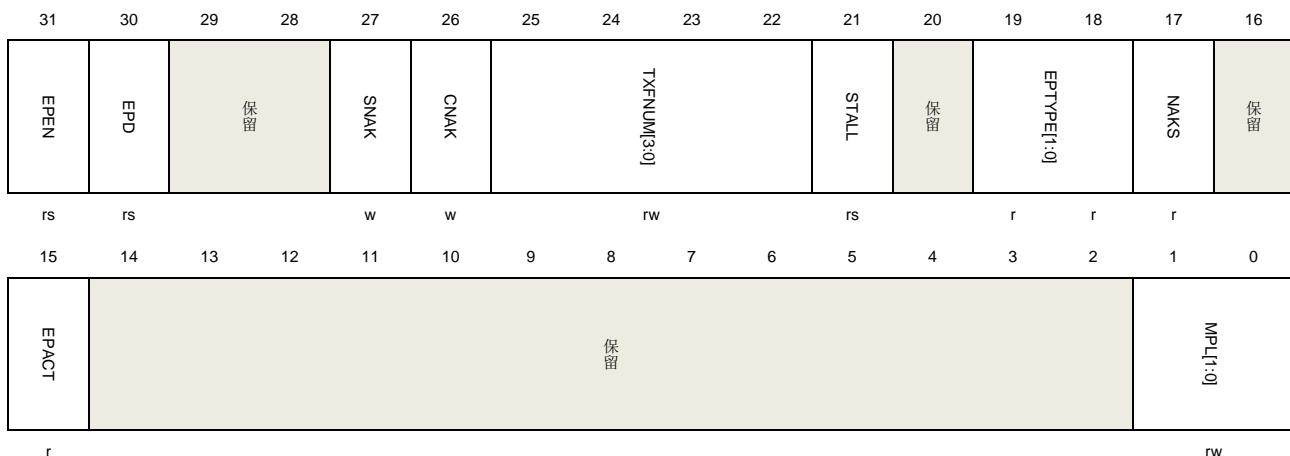
位/位域	名称	描述
31:4	保留	必须保留为复位值
3:0	IEPTXFEIE[3:0]	IN端点Tx FIFO空中断的使能位 该域控制着 USBFS_DIEPxINTF 寄存器的 TXFE 位能否生成一个寄存器 USBFS_DAEPINT的端点中断位 Bit0对应IN端点0, Bit5对应IN端点5 0: 除能FIFO空中断 1: 使能FIFO空中断

设备 IN 端点 0 控制寄存器 (USBFS_DIEP0CTL)

地址偏移: 0x0900

复位值: 0x0000 8000

该寄存器采用字 (32位) 访问



位/位域	名称	描述
31	EPEN	端点使能 软件置位、USBFS清零 0: 端点除能 1: 端点使能 软件应该按照操作指南使能或除能端点
30	EPD	端点除能 软件可通过置位该位从而除能端点，软件应该按照操作指南使能或除能端点。
29:28	保留	必须保留为复位值
27	SNAK	置位NAK 软件置位该位来设置该寄存器的NAKS位
26	CNAK	清零NAK 软件置位该位来清零该寄存器的NAKS位
25:22	TXFNUM[3:0]	Tx FIFO编号 定义IN端点0的Tx FIFO编号
21	STALL	STALL握手 当接收IN令牌时，软件可以通过置1该位发送STALL握手包，对于相应的OUT端点0，在接收SETUP令牌后，USBFS清除此位。该位比该寄存器的NAKS位和寄存器USBFS_DCTL的GINS位优先级要高，如果STALL和NAKS位都被置位，STALL位生效。
20	保留	必须保留为复位值
19:18	EPTYPE[1:0]	端点类型

该域固定为'00',控制端点。

17	NAKS	NAK状态 当该寄存器的STALL位和寄存器USBFS_DCTL的位GINS被清零，该位控制USBFS的NAK状态。 0: 根据端点Tx FIFO的状态，USBFS发送数据或握手包 1: USBFS总为IN令牌发送NAK握手包 该位是只读位，可以通过该寄存器的位CNAK和位SNAK控制该位
16	保留	必须保留为复位值
15	EPACT	端点激活 对于端点0来说，该域固定为'1'
14:2	保留	必须保留为复位值
1:0	MPL[1:0]	最大包长 域定义了控制数据包的最大包长，如USB 2.0协议所描述，对控制传输而言，有四种包长度： 00: 64字节 01: 32字节 10: 16字节 11: 8字节

设备 IN 端点 x 控制寄存器 (USBFS_DIEPxCTL) (x = 1..3, 3 是端点编号)

地址偏移: 0x0900 + (x * 0x20)

复位值: 0x0000 0000

该寄存器采用字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPEN	EPD	SODDFRM/MSD1 PID	SDOPID/ SEVENFRM	SNAK	CNAK		TXFNUM[3:0]		STALL	保留	EPTYPE[1:0]	NAKS		EOF/RM/DPID	
rs	rs	w	w	w	w		rw		rw/rs		rw	r	r		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPACT	保留									MPL[10:0]					
rw										rw					

位/位域	名称	描述
31	EPEN	端点使能 软件置位，USBFS清零

		0: 端点除能 1: 端点使能 软件应该按照操作指南使能或除能端点
30	EPD	端点除能 软件可通过置位该位从而除能端点，软件应该按照操作指南使能或除能端点。
29	SODDFRM	设置奇数帧（适用于同步IN端点） 软件通过置1该位置1该寄存器的EOFRM位
	SD1PID	设置DATA1 PID(适用于中断和大容量IN端点) 软件可通过置1该位置1该寄存器的DPID位
28	SEVENFRM	设置偶数帧(适用于同步IN端点) 软件通过置1该位清零该寄存器的EOFRM位
	SD0PID	设置DATA1(适用于中断和大容量IN端点) 软件可通过置1该位清零该寄存器的DPID位
27	SNAK	设置NAK 软件置1该位置1该寄存器的NAKS位
26	CNAK	清零NAK 软件置1该位清零该寄存器的NAKS位
25:22	TXFNUM[3:0]	Tx FIFO编号 该位定义了IN端点的Tx FIFO编号
21	STALL	STALL握手 当接收IN令牌时，软件可以通过置1该位发送STALL握手包。该位比该寄存器的NAKS位和寄存器USBFS_DCTL的GINS位优先级要高，如果STALL和NAKS位都被置位，STALL位生效。 对于控制IN端点： 当对应的OUT端点接收到SETUP令牌时，只有USBFS可以清零此位，软件不可清除此位。 对于中断或大容量IN端点： 只有软件可以清零此位。
20	保留	必须保留为复位值
19:18	EPTYPE[1:0]	端点类型 该域定义端点的传输类型： 00: 控制 01: 同步 10: 大容量 11: 中断
17	NAKS	NAK状态 当该寄存器的STALL位和寄存器USBFS_DCTL的位GINS被清零，该位控制USBFS的NAK状态： 0: 根据端点Tx FIFO的状态，USBFS发送数据或握手包

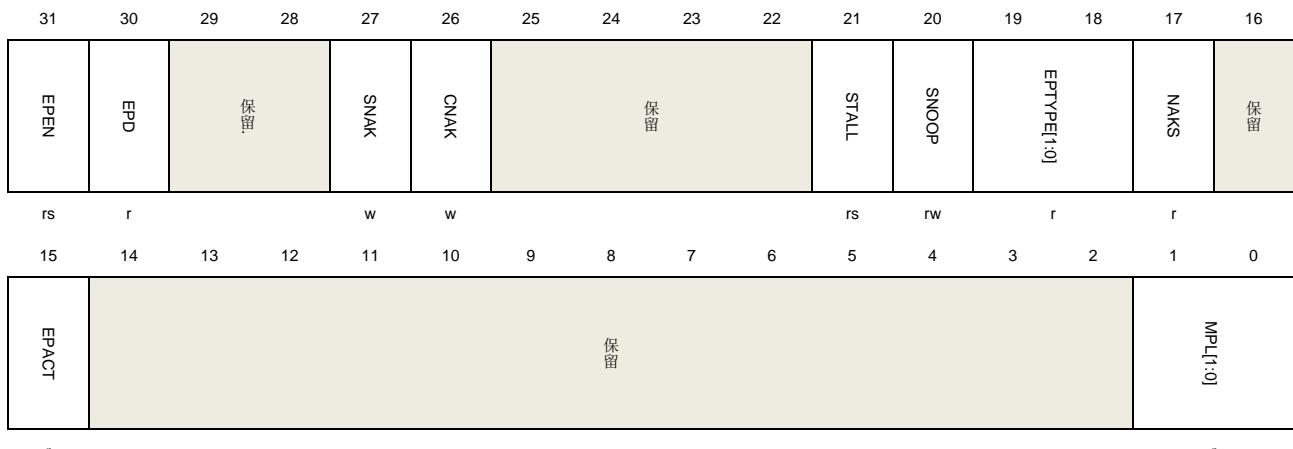
		1: USBFS总为IN令牌发送NAK握手包 该位是只读位，可以通过该寄存器的位CNAK和位SNAK控制该位
16	EOFRM	奇偶帧（适用于同步IN端点） 对于同步传输，软件通过使用该位控制USBFS只在奇数帧或偶数帧为IN事务发送数据包，如果当前帧号的奇偶性不匹配该位，USBFS回复一个零长度的包： 0: 只在偶数帧发送数据 1: 只在奇数帧发送数据
	DPID	端点数据PID（适用于中断或大容量IN端点） 在端点或大容量传输中，有数据PID翻转机制，在传输开始之前，软件通过设定SD0PID来设置此位，按照USB协议中描述的数据PID翻转机制，USBFS在传输过程中保持该位。 0: 数据包的PID是DATA0 1: 数据包的PID是DATA1
15	EPACT	端点激活 该位控制端点是否激活，当端点没有激活，忽略任何令牌，不做任何回复。
14:11	保留	必须保留为复位值
10:0	MPL[10:0]	该域定义最大包长

设备 OUT 端点 0 控制寄存器 (USBFS_DOEP0CTL)

地址偏移: 0x0B00

复位值: 0x0000 8000

该寄存器采用字（32位）访问



位/位域	名称	描述
31	EPEN	端点使能 软件置位，USBFS清零 0: 端点除能 1: 端点使能 软件应该按照操作指南使能或除能端点。

30	EPD	端点除能 对于OUT端点0, 该位固定为0
29:28	保留	必须保留为复位值
27	SNAK	设置NAK 软件置1该位置1该寄存器的NAKS位
26	CNAK	清零NAK 软件置1该位清零该寄存器的NAKS位
25:22	保留	必须保留为复位值
21	STALL	STALL握手 在OUT事务中, 软件可以通过置1该位发送STALL握手包, 对于OUT端点0, 在接收SETUP令牌后, USBFS清除此位。该位比该寄存器的NAKS位和寄存器USBFS_DCTL的GINS位优先级要高, 即如果STALL和NAKS位都被置位, STALL位生效。
20	SNOOP	调查模式 该位控制OUT端点的调查模式, 在调查模式中, USBFS不再检查接收数据包的CRC值 0: 调查模式除能 1: 调查模式使能
19:18	EPTYPE[1:0]	端点类型 对于控制端点, 该位固定为"00"
17	NAKS	NAK状态 当该寄存器的STALL位和寄存器USBFS_DCTL的位GINS被清零, 该位控制USBFS的NAK状态: 0: 根据端点Rx FIFO的状态, USBFS发送数据或握手包 1: USBFS为OUT事务发NAK握手包 该位是只读位, 通过该寄存器的CNAK和SNAK位控制该位
16	保留	必须保留为复位值
15	EPACT	端点激活 对于端点0, 该域固定为1
14:2	保留	必须保留为复位值
1:0	MPL[1:0]	最大包长 该位是只读位, 其数值来自于寄存器USBFS_DIEP0CTL的位MPL: 00: 64字节 01: 32字节 10: 16字节 11: 8字节

设备 OUT 端点 x 控制寄存器 (USBFS_DOEPxCTL) (x=1..3, x 是端点编号)

地址偏移: 0x0B00 + (x * 0x20)

复位值: 0x0000 0000

软件用该寄存器控制 OUT 端点 0 以外的每个逻辑 OUT 端点

该寄存器采用字（32位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPEN	EPD	SODDFRM/SD1 PID	SEVENFRM/ SD0PID	SNAK	CNAK	保留			STALL	SNOOP	EPTYPE[1:0]	NAKS			EOFRM/DPID
rs	rs	w	w	w	w				rw/rs	rw	rw	r		r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPACT	保留									MPL[10:0]					

位/位域	名称	描述
31	EPEN	端点使能 软件置位, USBFS清零 0: 端点除能 1: 端点使能 软件应该按照操作指南使能或除能端点。
30	EPD	端点除能 软件通过置1该位除能端点, 软件应该按照操作指南使能或除能端点。
29	SODDFRM	设置奇数帧 (适用于同步OUT端点) 该位只针对同步OUT端点有效 软件置1该位来置位该寄存器的EOFRM位
	SD1PID	设置DATA1 PID(适用于中断和大容量OUT端点) 软件置1该位来置位该寄存器的DPID位
28	SEVENFRM	设置偶数帧 (适用于同步OUT端点) 软件置1该位来清零该寄存器的EOFRM位
	SD0PID	设置DATA0 PID(适用于中断和大容量OUT端点) 软件置1该位来清零该寄存器的DPID位
27	SNAK	设置NAK 软件置1该位从而置1该寄存器的NAKS位
26	CNAK	清零NAK 软件置1该位从而清零该寄存器的NAKS位
25:22	保留	必须保留为复位值
21	STALL	STALL握手

在OUT事务中，软件可以通过置1该位发送STALL握手包。该位比该寄存器的NAKS位和寄存器USBFS_DCTL的GINS位优先级要高，如果STALL和NAKS位都被置位，STALL位生效。

对于控制OUT端点：

当OUT端点接收SETUP令牌时，只有USBFS可以清零该位，软件不可清零此位。

对于中断或大容量OUT端点

只有软件可以清零该位

20	SNOOP	调查模式 该位控制OUT端点的调查模式，在调查模式中，USBFS不再检查接收数据包的CRC值 0：调查模式除能 1：调查模式使能
19:18	EPTYPE[1:0]	端点类型 该域定义端点的传输类型 00：控制 01：同步 10：大容量 11：中断
17	NAKS	NAK状态 当该寄存器的STALL位和寄存器USBFS_DCTL的位GONS被清零，该位控制USBFS的NAK状态： 0：根据端点的Rx FIFO的状态，发送握手包 1：USBFS为OUT事务发送NAK握手 该位是只读位，通过该寄存器的CNAK和SNAK位控制该位
16	EOFRM	奇偶帧（适用于同步OUT端点） 对于同步传输，软件通过使用该位控制USBFS只在奇数帧或偶数帧发送数据包给OUT事务，如果当前帧号的奇偶性不匹配该位，USBFS不保存数据包 0：只在偶数帧发送数据 1：只在奇数帧发送数据
	DPID	端点数据PID（适用于中断或大容量端点） 在端点或大容量传输中，有数据PID翻转机制，在传输开始之前，软件通过设定SD0PID来设置此位，按照USB协议中描述的数据PID翻转机制，USBFS在传输过程中保持该位。 0：数据包PID是DATA0 1：数据包PID是DATA1
15	EPACT	端点激活 位控制端点是否激活，当端点没有激活，忽略任何令牌，不做任何回复
14:11	保留	必须保留为复位值
10:0	MPL[10:0]	该位定义最大包长

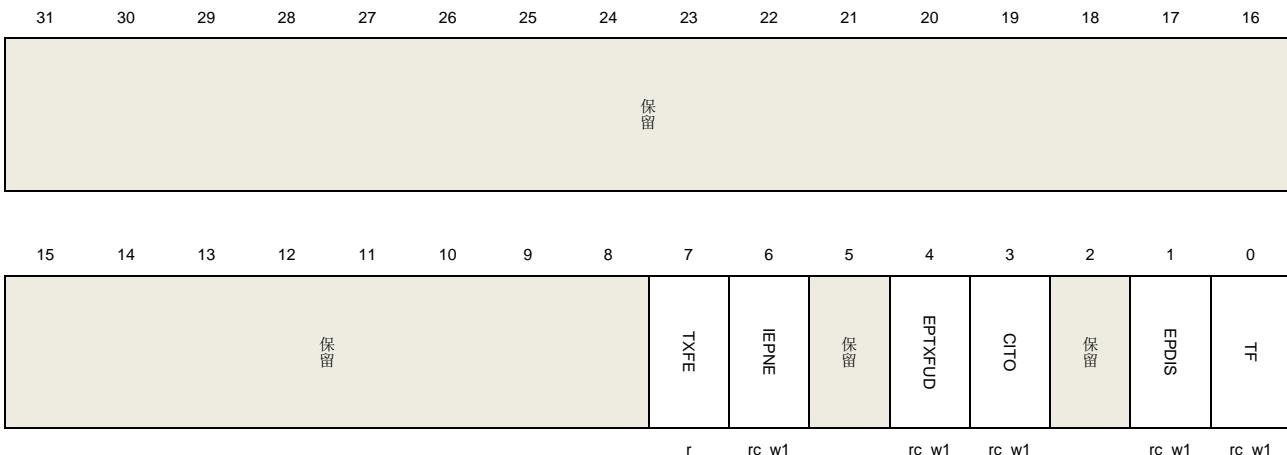
设备 IN 端点 x 中断标志寄存器 (USBFS_DIEPxINTF) (x = 0..3, x 是端点编号)

地址偏移: 0x0908 + (x * 0x20)

复位值: 0x0000 0080

该寄存器包含 IN 端点的状态和事件，当获得一个 IN 端点的中断时，应该读取该端点的中断标志寄存器，从而获知中断源。该寄存器的标志位通常硬件置位，除了 TXFE 位，各位写 1 清零。

该寄存器采用字（32位）访问



位/位域	名称	描述
31:8	保留	必须保留为复位值
7	TXFE	发送FIFO空 端点的Tx FIFO达到寄存器USBFS_GAHBCS的位TXFTH定义的空阈值。
6	IEPNE	IN端点NAK有效 寄存器USBFS_DIEPxCTL的位SNAK的设置生效，该位可以通过写1清零或设置CNAK位
5	保留	必须保留为复位值
4	EPTXFUD	端点Tx FIFO下溢 如果当IN令牌被接收后，Tx FIFO没有包数据，该标志被触发。
3	CITO	控制IN事务超时中断 在控制IN事务中，如果设备等待的握手包超时，该标志位被触发
2	保留	必须保留为复位值
1	EPDIS	端点除能 端点除能时，该标志位被触发
0	TF	传输完成 当该端点的所有IN事务完成，该标志位被触发。

设备 OUT 端点 x 中断标志寄存器 (USBFS_DOEPxINTF) (x = 0..3, x 是端点编号)

地址偏移: 0x0B08 + (x * 0x20)

复位值: 0x0000 0000

该寄存器包含 OUT 端点的状态和事件，当获得一个 OUT 端点的中断时，应该读取该端点的中断标志寄存器，从而获知中断源。该寄存器的标志位通常硬件置位，各位写 1 清零。

该寄存器采用字（32位）访问



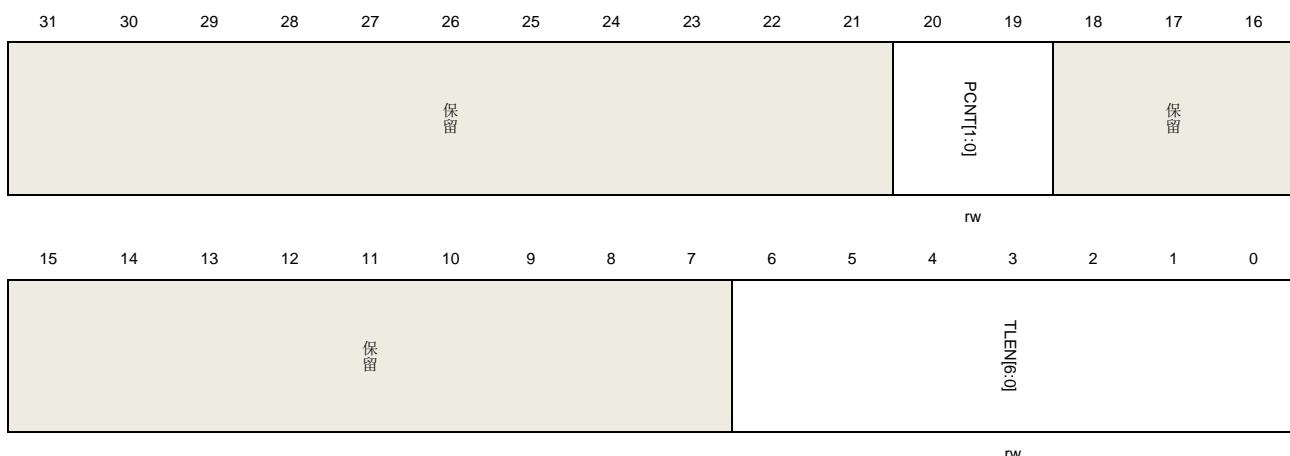
位/位域	名称	描述
31:7	保留	必须保留为复位值
6	BTBSTOP	连续SETUP包（适用于控制OUT端点） 当一个控制OUT端点接收超过连续3个SETUP包时，该标志被触发。
5	保留	必须保留为复位值
4	EPRXFOVR	端点Rx FIFO上溢 当OUT令牌被接收时，如果OUT端点的Rx FIFO没有足够的空间存放数据包，该位被触发。 在这种情况下，USBFS不能接收OUT数据包，发送一个NAK握手包。
3	STPF	SETUP阶段完成（适用于控制OUT端点） 当一个SETUP阶段完成，也就是USBFS在一个setup令牌后接收了一个IN或OUT令牌，该位被置位。
2	保留	必须保留为复位值
1	EPDIS	端点除能 端点除能时，该标志位被触发
0	TF	传输完成 当该端点的所有OUT事务完成，该标志位被触发

设备 IN 端点 0 传输长度寄存器 (**USBFS_DIEP0LEN**)

地址偏移: 0x0910

复位值: 0x0000 0000

该寄存器采用字 (32位) 访问



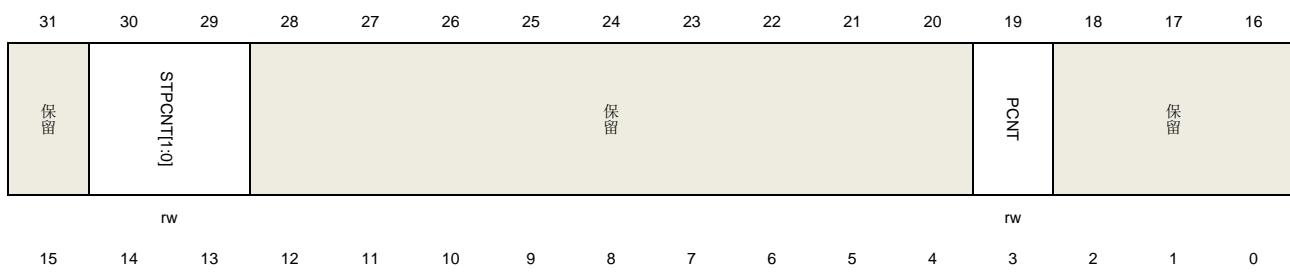
位/位域	名称	描述
31:21	保留	必须保留为复位值
20:19	PCNT[1:0]	包数 传输中被发送的数据包数量 在端点使能之前，软件设置该位，在传输开始后，该域在每次数据包成功发送后自动减少。
18:7	保留	必须保留为复位值
6:0	TLEN[6:0]	传输长度 一次传输的数据总字节数 该域是IN传输中需要发送的包数据的总字节数，在端点使能之前，软件设置该位，在软件或DMA成功地将包数据写入端点的Tx FIFO中，该域减少与包数据大小相同的数值。

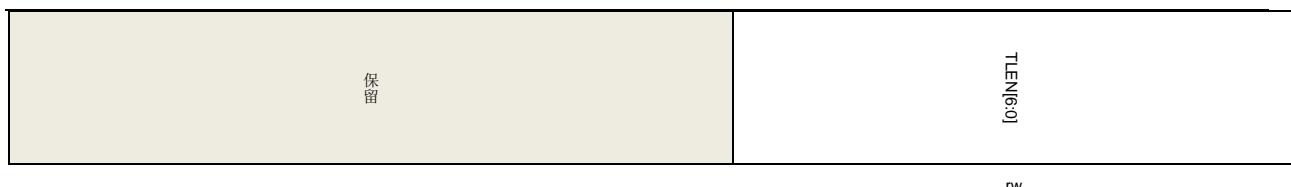
设备 OUT 端点 0 传输长度寄存器 (**USBFS_DOEP0LEN**)

地址偏移: 0x0B10

复位值: 0x0000 0000

该寄存器采用字 (32位) 访问





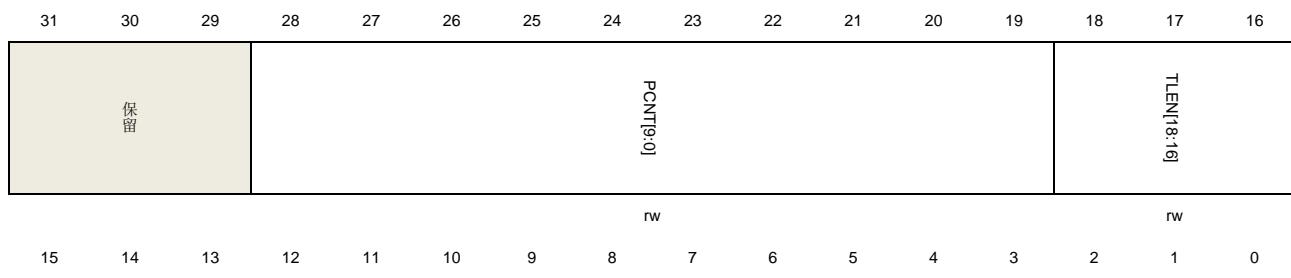
位/位域	名称	描述
31	保留	必须保留为复位值
30:29	STPCNT[1:0]	SETUP包计数 该域定义端点可以接受的最大连续SETUP包数量 在SETUP传输之前，设置该域，每当连续SETUP包接收到时，该域值减1，当该域达到0时，寄存器USBFS_DOEP0INTF的BTBSTOP标志被触发。 00: 0个包 01: 1个包 10: 2个包 11: 3个包
28:20	保留	必须保留为复位值
19	PCNT	包计数 一次传输中应该接收到包数量。 在端点使能前，软件设置该位，在传输开始后，每当数据包接收到后，该域数值自动减少。
18:7	保留	必须保留为复位值
6:0	TLEN[6:0]	传输长度 传输中数据总字数。 该域是OUT传输中需要接收的包数据的总字节数，在端点使能之前，软件设置该位，在软件或DMA成功地将包数据读取端点的Rx FIFO中，该域减少与包数据大小相同的数值。

设备 IN 端点 x 传输长度寄存器 (**USBFS_DIEPxLEN**) (x = 1..3, x 是端点编号)

地址偏移: 0x910 + (x * 0x20)

复位值: 0x0000 0000

该寄存器采用字（32位）访问





rw

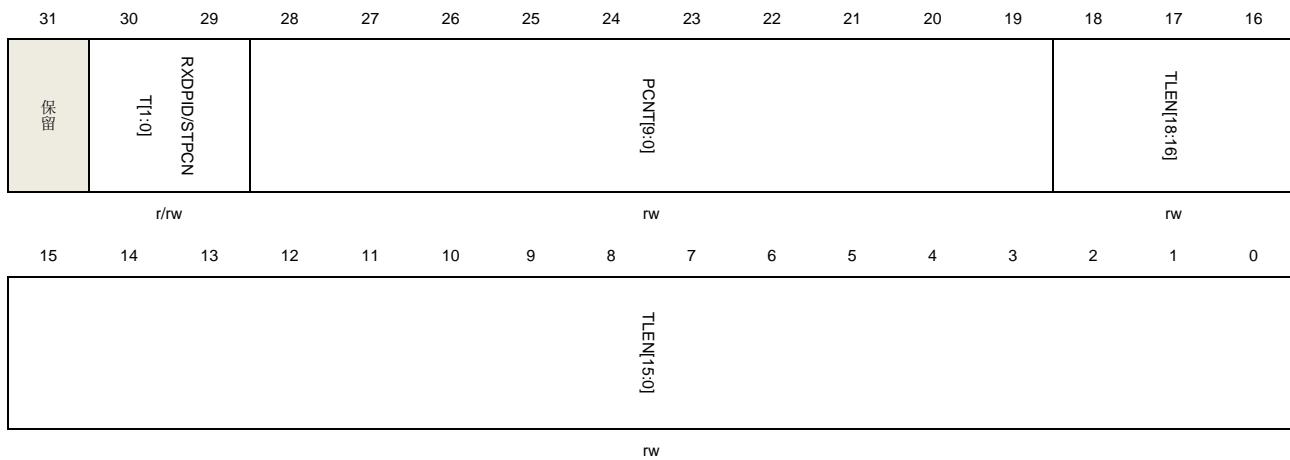
位/位域	名称	描述
31:29	保留	必须保留为复位值
28:19	PCNT[9:0]	包数量 传输中被发送的数据包数量 在端点使能之前，软件设置该位，在传输开始后，该域在每次数据包成功发送后自动减少。
18:0	TLEN[18:0]	传输长度 传输的数据总字节数 该域是IN传输中需要发送的包数据的总字节数，在端点使能之前，软件设置该位，在软件或DMA成功地将包数据写入端点的Tx FIFO中，该域减少与包数据大小相同的数值。

设备 OUT 端点 x 传输长度寄存器 (USBFS_DOEPxLEN) (x = 1..3, x 是端点编号)

地址偏移: 0x0B10 + (x * 0x20)

复位值: 0x0000 0000

该寄存器采用字（32位）访问



位/位域	名称	描述
31	保留	必须保留为复位值
30:29	RXDPID[1:0]	接收数据PID（适用于同步OUT端点）
		该域保存该端点该数据包所接受的最后一个数据包的PID
	00: DATA0	
	10: DATA1	

其他：保留

SETUP包数（适用于控制OUT端点）

STPCNT[1:0] 该位定义该端点可以接受连续SETUP最大包数

在SETUP传输之前，设置该域，每当连续SETUP包接收到时，该域值减1，当该域达到0时，寄存器USBFS_DOEP0INTF的BTBSTOP标志被触发。

00: 0个包

01: 1个包

10: 2个包

11: 3个包

28:19 **PCNT[9:0]** 包数

传输中应该接收到包数量

在端点使能前，软件设置该位，在传输开始后，每当数据包接收到后，该域数值自动减少。

18:0 **TLEN[18:0]** 传输长度

传输中数据总字数

该域是IN传输中需要接收的包数据的总字节数，在端点使能之前，软件设置该位，在软件或DMA成功地将包数据读取端点的Rx FIFO中，该域减少与包数据大小相同的数值。

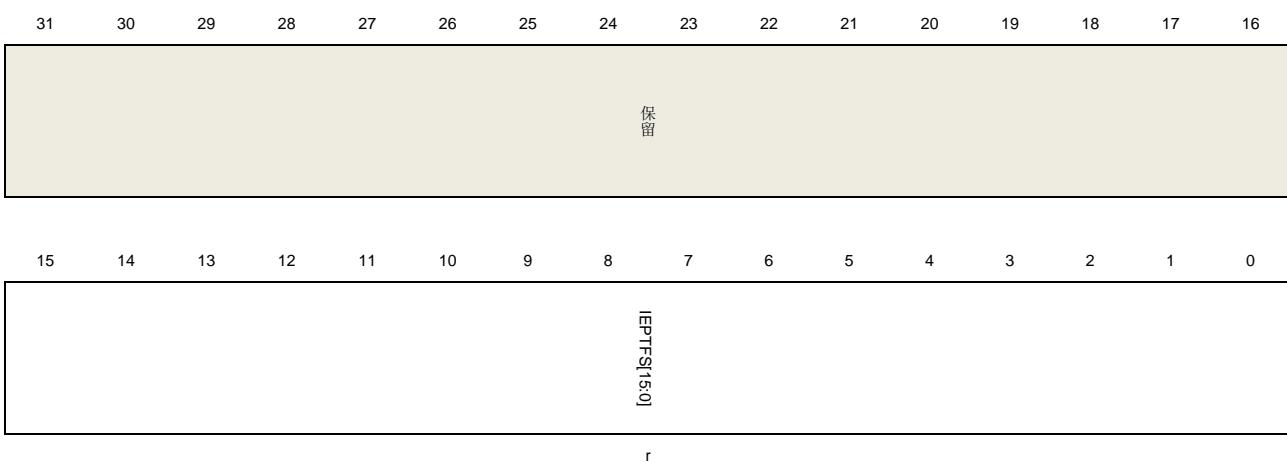
设备 IN 端点 x 发送 FIFO 状态寄存器 (USBFS_DIEPxTFSTAT) (x = 0..3, x 是端点编号)

地址偏移: 0x0918 + (x * 0x20)

复位值: 0x0000 0200

该寄存器包含每个端点的 Tx FIFO 的信息。

该寄存器采用字（32位）访问



位/位域	名称	描述
31:16	保留	必须保留为复位值
15:0	IEPTFS[15:0]	IN端点的Tx FIFO可用空间

IN端点的Tx FIFO可用空间用32位字为单位

0: FIFO是满的

1: 1字可用

...

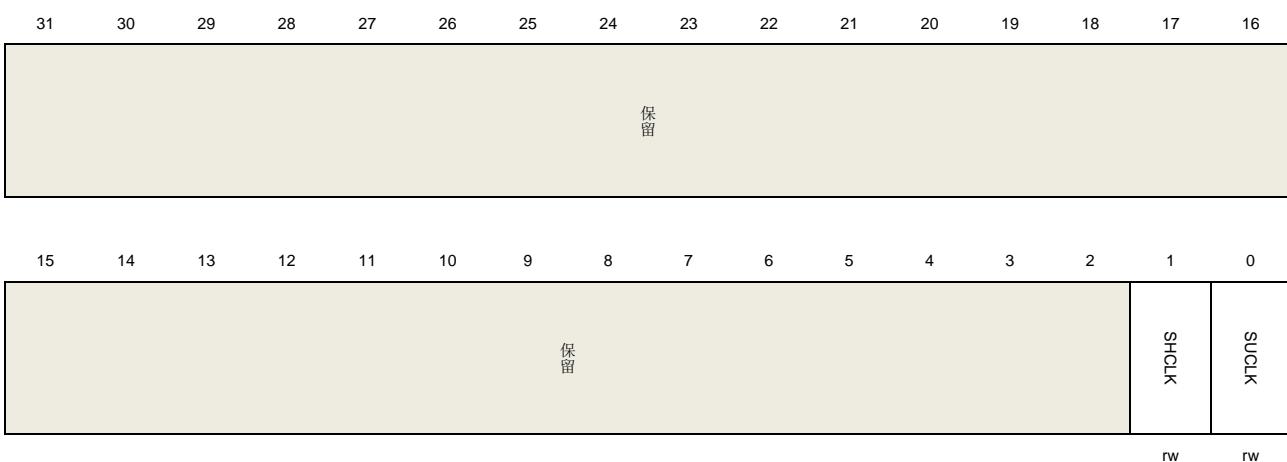
n: n字可用

23.7.4. 电源和时钟控制寄存器 (**USBFS_PWRCLKCTL**)

地址偏移: 0x0E00

复位值: 0x0000 0000

该寄存器采用字（32位）访问



位/位域	名称	描述
31:2	保留	必须保留为复位值
1	SHCLK	停止HCLK 停止HCLK, 节省电量 0: HCLK未停止 1: HCLK停止
0	SUCLK	停止USB时钟 停止USB时钟, 节省电量 0: USB时钟未停止 1: USB时钟停止

24. 版本历史

表 24-1. 版本历史

版本号	描述	日期
1.0	初稿发布	2017 年 6 月 6 日