

GigaDevice Semiconductor Inc.

**GD32C10x
ARM® Cortex®-M4 32-bit MCU**

用户手册

1.1 版本

(2021 年 12 月)

目 录

目录	2
图索引	15
表索引	21
1. 系统及存储器架构	24
1.1. ARM Cortex-M4 处理器	24
1.2. 系统架构	25
1.3. 存储器映射	27
1.3.1. 位带操作	31
1.3.2. 片上 SRAM 存储器	32
1.3.3. 片上 FLASH 存储器概述	32
1.4. 引导配置	32
1.5. 设备电子签名	33
1.5.1. 存储器容量信息	33
1.5.2. 设备唯一 ID (96 位)	33
1.6. 系统配置寄存器	34
2. 闪存控制器 (FMC)	36
2.1. 简介	36
2.2. 主要特征	36
2.3. 功能说明	36
2.3.1. 闪存结构	36
2.3.2. 读操作	37
2.3.3. FMC_CTL 寄存器解锁	38
2.3.4. 页擦除	39
2.3.5. 整片擦除	40
2.3.6. 主存储闪存块编程	41
2.3.7. OTP 编程	43
2.3.8. 选项字节擦除	43
2.3.9. 选项字节编程	44
2.3.10. 选项字节说明	44
2.3.11. 页擦除/编程保护	45
2.3.12. 安全保护	45
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. 选项字节操作解锁寄存器 (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 (FMC_ADDR).....	51
2.4.7. 选项字节状态寄存器 (FMC_OBSTAT)	51
2.4.8. 擦除/编程保护寄存器 (FMC_WP).....	52
2.4.9. 产品 ID 寄存器 (FMC_PID)	52
3. 电源管理单元 (PMU)	54
3.1. 简介	54
3.2. 主要特征	54
3.3. 功能说明	55
3.3.1. 电池备份域	55
3.3.2. V _{DD} / V _{DDA} 电源域	56
3.3.3. 1.2V 电源域	58
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).....	61
4. 备份寄存器 (BKP)	63
4.1. 简介	63
4.2. 主要特征特性	63
4.3. 功能说明	63
4.3.1. RTC 时钟校准	63
4.3.2. 侵入检测.....	63
4.4. BKP 寄存器	64
4.4.1. 备份数据寄存器 (BKP_DATAx) (x= 0..41)	64
4.4.2. RTC 信号输出控制寄存器 (BKP_OCTL)	64
4.4.3. 侵入引脚控制寄存器 (BKP_TPCTL).....	65
4.4.4. 侵入控制状态寄存器 (BKP_TPCS)	65
5. 复位和时钟单元 (RCU)	67
5.1. 复位控制单元(RCTL)	67
5.1.1. 简介	67
5.1.2. 功能说明.....	67
5.2. 时钟控制单元(CCTL)	68
5.2.1. 简介	68
5.2.2. 主要特征	70
5.2.3. 功能说明	70
5.3. RCU 寄存器	74
5.3.1. 控制寄存器 (RCU_CTL)	74

5.3.2.	时钟配置寄存器 0 (RCU_CFG0).....	76
5.3.3.	时钟中断寄存器 (RCU_INT).....	79
5.3.4.	APB2 复位寄存器 (RCU_APB2RST).....	82
5.3.5.	APB1 复位寄存器 (RCU_APB1RST).....	84
5.3.6.	AHB 使能寄存器 (RCU_AHBEN).....	87
5.3.7.	APB2 使能寄存器 (RCU_APB2EN)	89
5.3.8.	APB1 使能寄存器 (RCU_APB1EN)	91
5.3.9.	备份域控制寄存器 (RCU_BDCTL).....	94
5.3.10.	复位源/时钟寄存器 (RCU_RSTSCK).....	95
5.3.11.	AHB 复位寄存器 (RCU_AHBRST).....	97
5.3.12.	时钟配置寄存器 1 (RCU_CFG1).....	97
5.3.13.	深度睡眠模式电压寄存器 (RCU_DSV).....	100
5.3.14.	附加时钟控制寄存器 (RCU_ADDCTL).....	100
5.3.15.	附加时钟中断寄存器 (RCU_ADDINT).....	101
5.3.16.	APB1 附加复位寄存器 (RCU_ADDAPB1RST)	102
5.3.17.	APB1 附加使能寄存器 (RCU_ADDAPB1EN).....	102
6.	时钟校准控制器 (CTC)	104
6.1.	简介	104
6.2.	主要特征	104
6.3.	功能说明	105
6.3.1.	参考同步脉冲发生器	105
6.3.2.	CTC 校准计数器	105
6.3.3.	频率评估和自动校准过程	106
6.3.4.	软件编程指南	107
6.4.	CTC 寄存器	108
6.4.1.	控制寄存器 0 (CTC_CTL0)	108
6.4.2.	控制寄存器 1 (CTC_CTL1)	109
6.4.3.	状态寄存器 (CTC_STAT).....	110
6.4.4.	中断清除寄存器 (CTC_INTC).....	112
7.	中断/事件控制器 (EXTI)	113
7.1.	简介	113
7.2.	主要特征	113
7.3.	功能说明	113
7.4.	外部中断及事件(EXTI) 框图	116
7.5.	外部中断及事件功能概述	116
7.6.	EXTI 寄存器	118
7.6.1.	中断使能寄存器 (EXTI_INTEN).....	118
7.6.2.	事件使能寄存器 (EXTI_EVENT).....	118
7.6.3.	上升沿触发使能寄存器 (EXTI_RTEN).....	119

7.6.4.	下降沿触发使能寄存器 (EXTI_FTEN)	119
7.6.5.	软件中断事件寄存器 (EXTI_SWIEV)	119
7.6.6.	挂起寄存器 (EXTI_PD).....	120
8.	通用和备用输入/输出接口（GPIO 和 AFIO）	121
8.1.	简介	121
8.2.	主要特征	121
8.3.	功能说明	121
8.3.1.	GPIO 引脚配置	122
8.3.2.	外部中断/事件线	123
8.3.3.	备用功能(AF).....	123
8.3.4.	输入配置.....	123
8.3.5.	输出配置.....	123
8.3.6.	模拟配置.....	124
8.3.7.	备用功能(AF)配置	125
8.3.8.	GPIO 锁定功能	125
8.3.9.	GPIO I/O 补偿单元	126
8.4.	I/O 重映射功能和调试配置	126
8.4.1.	介绍	126
8.4.2.	主要特征	126
8.4.3.	JTAG/SWD 备用功能重映射	126
8.4.4.	ADC AF 重映射	127
8.4.5.	TIMER AF 重映射	128
8.4.6.	USART AF 重映射	129
8.4.7.	I2C0 备用功能重映射	130
8.4.8.	SPI0 备用功能重映射	130
8.4.9.	SPI2/I2S2 备用功能重映射	130
8.4.10.	CAN0 备用功能重映射	130
8.4.11.	CAN1 备用功能重映射	131
8.4.12.	CTC 备用功能重映射	131
8.4.13.	CLK 引脚 AF 重映射	131
8.5.	GPIO 寄存器	132
8.5.1.	端口控制寄存器 0 (GPIOx_CTL0, x=A..E).....	132
8.5.2.	端口控制寄存器 1 (GPIOx_CTL1, x=A..E).....	134
8.5.3.	端口输入状态寄存器 (GPIOx_ISTAT, x=A..E)	135
8.5.4.	端口输出控制寄存器 (GPIOx_OCTL, x=A..E).....	136
8.5.5.	端口位操作寄存器 (GPIOx_BOP, x=A..E)	136
8.5.6.	位清除寄存器 (GPIOx_BC, x=A..E)	137
8.5.7.	端口配置锁定寄存器 (GPIOx_LOCK, x=A..E).....	137
8.5.8.	端口位速度寄存器 (GPIOx_SPD, x=A..E).....	138
8.5.9.	事件控制寄存器 (AFIO_EC).....	139
8.5.10.	AFIO 端口配置寄存器 0 (AFIO_PCF0)	139

8.5.11. EXTI 源选择寄存器 0 寄存器 (AFIO_EXTISSL)	143
8.5.12. EXTI 源选择寄存器 1 寄存器 (AFIO_EXTISSL1)	144
8.5.13. EXTI 源选择寄存器 2 寄存器 (AFIO_EXTISSL2)	145
8.5.14. EXTI 源选择寄存器 3 寄存器 (AFIO_EXTISSL3)	146
8.5.15. AFIO 端口配置寄存器 1 (AFIO_PCF1)	147
8.5.16. IO 补偿控制寄存器 (AFIO_CPSCTL)	148
9. 循环冗余校验计算单元 (CRC)	150
9.1. 简介	150
9.2. 主要特征	150
9.3. 功能说明	151
9.4. CRC 寄存器	152
9.4.1. 数据寄存器 (CRC_DATA)	152
9.4.2. 独立数据寄存器 (CRC_FDATA)	152
9.4.3. 控制寄存器 (CRC_CTL)	153
10. 直接存储器访问控制器 (DMA)	154
10.1. 简介	154
10.2. 主要特征	154
10.3. 结构框图	155
10.4. 功能说明	155
10.4.1. DMA 操作	155
10.4.2. 外设握手	156
10.4.3. 仲裁	157
10.4.4. 地址生成	157
10.4.5. 循环模式	157
10.4.6. 存储器到存储器模式	158
10.4.7. 通道配置	158
10.4.8. 中断	158
10.4.9. DMA 请求映射	159
10.5. DMA 寄存器	162
10.5.1. 中断标志位寄存器 (DMA_INTF)	162
10.5.2. 中断标志位清除寄存器 (DMA_INTC)	163
10.5.3. 通道 x 控制寄存器 (DMA_CHxCTL)	163
10.5.4. 通道 x 计数寄存器 (DMA_CHxCNT)	165
10.5.5. 通道 x 外设基址寄存器 (DMA_CHxPADDR)	166
10.5.6. 通道 x 存储器基址寄存器 (DMA_CHxMADDR)	166
11. 调试 (DBG)	168
11.1. 简介	168
11.2. JTAG/SW 功能说明	168

11.2.1.	切换 JTAG/ SW 接口	168
11.2.2.	引脚分配.....	168
11.2.3.	JTAG 链状结构.....	169
11.2.4.	调试复位.....	169
11.2.5.	JEDEC-106 ID 代码	169
11.3.	调试保持功能说明	169
11.3.1.	低功耗模式调试支持	169
11.3.2.	TIMER, I2C, WWDGT, FWDGT 和 CAN 外设调试支持	170
11.4.	DBG 寄存器	171
11.4.1.	ID 寄存器 (DBG_ID).....	171
11.4.2.	控制寄存器 (DBG_CTL).....	171
12.	模数转换器 (ADC)	175
12.1.	简介.....	175
12.2.	主要特征.....	175
12.3.	引脚和内部信号	176
12.4.	功能说明.....	177
12.4.1.	校准 (CLB)	177
12.4.2.	ADC 时钟	178
12.4.3.	ADCON 开关.....	178
12.4.4.	规则组和注入组.....	178
12.4.5.	转换模式.....	178
12.4.6.	注入通道管理	182
12.4.7.	模拟看门狗	183
12.4.8.	数据对齐	183
12.4.9.	可编程的采样时间	184
12.4.10.	外部触发	184
12.4.11.	DMA 请求	185
12.4.12.	温度传感器和内部参考电压 VREFINT	185
12.4.13.	可编程分辨率(DRES) – 快速转换模式	185
12.4.14.	片上硬件过采样	186
12.5.	ADC 同步模式	187
12.5.1.	独立模式.....	188
12.5.2.	规则并行模式	188
12.5.3.	注入并行模式	189
12.5.4.	快速交叉模式	189
12.5.5.	慢速交叉模式	190
12.5.6.	交替触发模式	191
12.5.7.	规则并行和注入并行组合模式	192
12.5.8.	规则并行和交替触发组合模式	192
12.5.9.	注入并行和交叉组合模式	193

12.6. 中断.....	193
12.7. ADC 寄存器.....	194
12.7.1. 状态寄存器 (ADC_STAT)	194
12.7.2. 控制寄存器 0 (ADC_CTL0)	195
12.7.3. 控制寄存器 1 (ADC_CTL1)	197
12.7.4. 采样时间寄存器 0 (ADC_SAMPT0)	199
12.7.5. 采样时间寄存器 1 (ADC_SAMPT1)	200
12.7.6. 注入通道数据偏移寄存器 x (ADC_IOFFx) (x=0..3)	201
12.7.7. 看门狗高阈值寄存器 (ADC_WDHT).....	201
12.7.8. 看门狗低阈值寄存器 (ADC_WDLT).....	201
12.7.9. 规则序列寄存器 0 (ADC_RSQ0).....	202
12.7.10. 规则序列寄存器 1 (ADC_RSQ1)	202
12.7.11. 规则序列寄存器 2 (ADC_RSQ2)	203
12.7.12. 注入序列寄存器 (ADC_ISQ).....	204
12.7.13. 注入数据寄存器 x (ADC_IDATAx) (x= 0..3).....	204
12.7.14. 规则数据寄存器 (ADC_RDATA)	205
12.7.15. 过采样控制寄存器 (ADC_OVSAMPCTL)	205
13. 数模转换器 (DAC)	208
13.1. 简介.....	208
13.2. 主要特征.....	208
13.3. 功能说明.....	209
13.3.1. DAC 使能	209
13.3.2. DAC 输出缓冲	209
13.3.3. DAC 数据配置	209
13.3.4. DAC 触发	209
13.3.5. DAC 转换	210
13.3.6. DAC 噪声波.....	210
13.3.7. DAC 输出电压	211
13.3.8. DMA 请求	211
13.3.9. DAC 并发转换	211
13.4. DAC 寄存器.....	212
13.4.1. 控制寄存器 (DAC_CTL)	212
13.4.2. 软件触发寄存器 (DAC_SWT).....	214
13.4.3. DAC0 12 位右对齐数据保持寄存器 (DAC0_R12DH).....	215
13.4.4. DAC0 12 位左对齐数据保持寄存器 (DAC0_L12DH)	215
13.4.5. DAC0 8 位右对齐数据保持寄存器 (DAC0_R8DH).....	216
13.4.6. DAC1 12 位右对齐数据保持寄存器 (DAC1_R12DH).....	216
13.4.7. DAC1 12 位左对齐数据保持寄存器 (DAC1_L12DH)	217
13.4.8. DAC1 8 位右对齐数据保持寄存器 (DAC1_R8DH).....	217
13.4.9. DAC 并发模式 12 位右对齐数据保持寄存器 (DACC_R12DH)	218
13.4.10. DAC 并发模式 12 位左对齐数据保持寄存器 (DACC_L12DH).....	218

13.4.11.	DAC 并发模式 8 位右对齐数据保持寄存器 (DACC_R8DH)	219
13.4.12.	DAC0 数据输出寄存器 (DAC0_DO)	219
13.4.13.	DAC1 数据输出寄存器 (DAC1_DO)	220
14.	看门狗定时器 (WDGT)	221
14.1.	独立看门狗定时器 (FWDGT)	221
14.1.1.	简介	221
14.1.2.	主要特征	221
14.1.3.	功能说明	221
14.1.4.	FWDGT 寄存器	224
14.2.	窗口看门狗定时器 (WWDT)	227
14.2.1.	简介	227
14.2.2.	主要特征	227
14.2.3.	功能说明	227
14.2.4.	WWDT 寄存器	230
15.	实时时钟 (RTC)	232
15.1.	简介	232
15.2.	主要特征	232
15.3.	功能说明	232
15.3.1.	RTC 复位	233
15.3.2.	RTC 读取	233
15.3.3.	RTC 配置	233
15.3.4.	RTC 标志位	234
15.4.	RTC 寄存器	235
15.4.1.	RTC 中断使能寄存器 (RTC_INTEN)	235
15.4.2.	RTC 控制寄存器 (RTC_CTL)	235
15.4.3.	RTC 预分频寄存器高位 (RTC_PSCH)	236
15.4.4.	RTC 预分频寄存器低位 (RTC_PSCL)	236
15.4.5.	RTC 分频器高位 (RTC_DIVH)	237
15.4.6.	RTC 分频器低位 (RTC_DIVL)	237
15.4.7.	RTC 计数寄存器高位 (RTC_CNTH)	237
15.4.8.	RTC 计数寄存器低位 (RTC_CNTL)	238
15.4.9.	RTC 闹钟寄存器高位 (RTC_ALRMH)	238
15.4.10.	RTC 闹钟寄存器低位 (RTC_ALRML)	238
16.	定时器 (TIMER)	239
16.1.	高级定时器 (TIMERx,x=0,7)	240
16.1.1.	简介	240
16.1.2.	主要特征	240
16.1.3.	结构框图	241
16.1.4.	功能说明	241
16.1.5.	TIMERx 寄存器(x=0,7)	268

16.2. 通用定时器 L0 (TIMERx, x=1,2,3,4)	292
16.2.1. 简介	292
16.2.2. 主要特征.....	292
16.2.3. 结构框图.....	293
16.2.4. 功能说明.....	293
16.2.5. TIMERx 寄存器 (x=1,2,3,4)	309
16.3. 通用定时器 L1 (TIMERx, x=8,11)	329
16.3.1. 简介	329
16.3.2. 主要特征.....	329
16.3.3. 结构框图.....	330
16.3.4. 功能说明.....	330
16.3.5. TIMERx 寄存器(x=8,11)	341
16.4. 通用定时器 L2 (TIMERx, x=9,10,12,13)	353
16.4.1. 简介	353
16.4.2. 主要特征.....	353
16.4.3. 结构框图.....	353
16.4.4. 功能说明.....	353
16.4.5. TIMERx 寄存器(x=9,10,12,13).....	361
16.5. 基本定时器 (TIMERx, x=5,6)	371
16.5.1. 简介	371
16.5.2. 主要特征.....	371
16.5.3. 结构框图.....	371
16.5.4. 功能说明.....	371
16.5.5. TIMERx 寄存器(x=5,6).....	375
17. 通用同步异步收发器 (USART)	380
 17.1. 简介.....	380
 17.2. 主要特征.....	380
 17.3. 功能说明.....	381
17.3.1. USART 帧格式	382
17.3.2. 波特率发生	383
17.3.3. USART 发送器	383
17.3.4. USART 接收器	384
17.3.5. DMA 方式访问数据缓冲区.....	385
17.3.6. 硬件流控制	387
17.3.7. 多处理器通信	388
17.3.8. LIN 模式	388
17.3.9. 同步通信模式	389
17.3.10. 串行红外(IrDA SIR)编解码功能模块.....	390
17.3.11. 半双工通信模式.....	391
17.3.12. 智能卡(ISO7816-3)模式	391
17.3.13. USART 中断.....	393

17.4. USART 寄存器	395
17.4.1. 状态寄存器 0 (USART_STAT0).....	395
17.4.2. 数据寄存器 (USART_DATA)	397
17.4.3. 波特率寄存器 (USART_BAUD).....	397
17.4.4. 控制寄存器 0 (USART_CTL0).....	398
17.4.5. 控制寄存器 1 (USART_CTL1).....	399
17.4.6. 控制寄存器 2 (USART_CTL2).....	401
17.4.7. 保护时间和预分频器寄存器 (USART_GP).....	402
17.4.8. 控制寄存器 3 (USART_CTL3).....	403
17.4.9. 接收超时寄存器 (USART_RT).....	405
17.4.10. 状态寄存器 1 (USART_STAT1).....	405
17.4.11. 兼容性控制寄存器 (USART_CHC).....	406
18. 内部集成电路总线接口 (I2C)	408
18.1. 简介.....	408
18.2. 主要特征.....	408
18.3. 功能说明.....	409
18.3.1. SDA 线和 SCL 线	409
18.3.2. 数据有效性	410
18.3.3. 开始和停止状态	410
18.3.4. 时钟同步	410
18.3.5. 仲裁	411
18.3.6. I2C 通讯流程	411
18.3.7. 软件编程模型	412
18.3.8. SCL 线控制	419
18.3.9. DMA 模式下数据传输	420
18.3.10. 报文错误校验.....	420
18.3.11. SMBus 支持	420
18.3.12. SAM_V 支持	421
18.3.13. 状态、错误和中断	422
18.4. I2C 寄存器	423
18.4.1. 控制寄存器 0 (I2C_CTL0)	423
18.4.2. 控制寄存器 1 (I2C_CTL1)	424
18.4.3. 从机地址寄存器 0 (I2C_SADDR0)	425
18.4.4. 从机地址寄存器 1 (I2C_SADDR1)	426
18.4.5. 传输缓冲区寄存器 (I2C_DATA)	426
18.4.6. 传输状态寄存器 0 (I2C_STAT0)	427
18.4.7. 传输状态寄存器 1 (I2C_STAT1)	429
18.4.8. 时钟配置寄存器 (I2C_CKCFG)	430
18.4.9. 上升时间寄存器 (I2C_RT)	431
18.4.10. SAM 控制状态寄存器 (I2C_SAMCS)	431
18.4.11. 快速+ 模式配置寄存器 (I2C_FMPCFG).....	432

19. 串行外设接口/片上音频接口 (SPI/I2S)	433
19.1. 简介.....	433
19.2. 主要特征.....	433
19.2.1. SPI 主要特征.....	433
19.2.2. I2S 主要特征	433
19.3. SPI 结构框图.....	434
19.4. SPI 信号线描述	434
19.4.1. 常规配置 (非 SPI 四线模式)	434
19.4.2. SPI 四线配置.....	434
19.5. SPI 功能说明	435
19.5.1. SPI 时序和数据帧格式	435
19.5.2. NSS 功能	436
19.5.3. SPI 运行模式.....	437
19.5.4. DMA 功能	445
19.5.5. CRC 功能	445
19.6. SPI 中断.....	445
19.6.1. 状态标志位	445
19.6.2. 错误标志.....	446
19.7. I2S 结构框图	447
19.8. I2S 信号线描述.....	447
19.9. I2S 功能说明	447
19.9.1. I2S 音频标准	447
19.9.2. I2S 时钟	455
19.9.3. 运行	456
19.9.4. DMA 功能	458
19.10. I2S 中断	458
19.10.1. 状态标志位	458
19.10.2. 错误标志.....	458
19.11. SPI/I2S 寄存器	460
19.11.1. 控制寄存器 0 (SPI_CTL0).....	460
19.11.2. 控制寄存器 1 (SPI_CTL1).....	462
19.11.3. 状态寄存器 (SPI_STAT)	463
19.11.4. 数据寄存器 (SPI_DATA).....	464
19.11.5. CRC 多项式寄存器 (SPI_CRCPOLY)	465
19.11.6. 接收 CRC 寄存器 (SPI_RCRC).....	465
19.11.7. 发送 CRC 寄存器 (SPI_TCRC)	466
19.11.8. I2S 控制寄存器 (SPI_I2SCTL)	466
19.11.9. I2S 时钟预分频寄存器 (SPI_I2SPSC).....	468
19.11.10. SPI0 四路 SPI 控制寄存器 (SPI_QCTL).....	468

20. 外部存储器控制器 (EXMC)	470
20.1. 简介.....	470
20.2. 主要特征.....	470
20.3. 功能说明.....	470
20.3.1. 结构框图.....	470
20.3.2. EXMC 访问基本规范	471
20.3.3. NOR/PSRAM 控制器	472
20.4. EXMC 寄存器	491
20.4.1. SRAM/NOR Flash 控制寄存器 (EXMC_SNCTL)	491
20.4.2. SRAM/NOR Flash 时序配置寄存器 (EXMC_SNTCFG).....	492
20.4.3. SRAM/NOR Flash 写时序寄存器 (EXMC_SNWTCFG)	494
21. 控制器局域网络 (CAN)	496
21.1. 简介.....	496
21.2. 主要特征.....	496
21.3. 功能说明.....	497
21.3.1. 工作模式.....	497
21.3.2. 通信模式.....	498
21.3.3. 数据发送.....	499
21.3.4. 数据接收.....	501
21.3.5. 过滤功能.....	502
21.3.6. 时间触发通信	505
21.3.7. 通信参数.....	506
21.3.8. CAN FD 操作	507
21.3.9. 传输延迟补偿	508
21.3.10. 错误标志	508
21.3.11. 中断	509
21.4. CAN 寄存器.....	511
21.4.1. 控制寄存器 (CAN_CTL)	511
21.4.2. 状态寄存器 (CAN_STAT)	512
21.4.3. 发送状态寄存器 (CAN_TSTAT).....	514
21.4.4. 接收 FIFO0 寄存器 (CAN_RFIFO0)	516
21.4.5. 接收 FIFO1 寄存器 (CAN_RFIFO1)	517
21.4.6. 中断使能寄存器 (CAN_INTEN)	518
21.4.7. 错误寄存器 (CAN_ERR)	519
21.4.8. 位时序寄存器 (CAN_BT).....	520
21.4.9. FD 控制寄存器 (CAN_FDCTL).....	521
21.4.10. FD 状态寄存器 (CAN_FDSTAT).....	522
21.4.11. FD 传输延迟补偿寄存器 (CAN_FDTDC).....	522
21.4.12. 数据位时序寄存器 (CAN_DBT)	523
21.4.13. 发送邮箱标识符寄存器 (CAN_TMIx) (x=0..2).....	524

21.4.14. 发送邮箱属性寄存器 (CAN_TMPx) (x=0..2).....	524
21.4.15. 发送邮箱 data0 寄存器 (CAN_TMDATA0x) (x=0..2).....	525
21.4.16. 发送邮箱 data1 寄存器 (CAN_TMDATA1x) (x=0..2).....	526
21.4.17. 接收 FIFO 邮箱标识符寄存器 (CAN_RFIFOMIx) (x=0,1).....	526
21.4.18. 接收 FIFO 邮箱属性寄存器 (CAN_RFIFOMPx) (x=0,1).....	527
21.4.19. 接收 FIFO 邮箱 data0 寄存器 (CAN_RFIFOMDATA0x) (x=0,1).....	528
21.4.20. 接收 FIFO 邮箱 data1 寄存器 (CAN_RFIFOMDATA1x) (x=0,1).....	528
21.4.21. 过滤器控制寄存器 (CAN_FCTL)	529
21.4.22. 过滤器模式配置寄存器 (CAN_FMCFG)	529
21.4.23. 过滤器位宽配置寄存器 (CAN_FSCFG).....	530
21.4.24. 过滤器关联 FIFO 寄存器 (CAN_FA FIFO).....	530
21.4.25. 过滤器激活寄存器 (CAN_FW).....	531
21.4.26. 过滤器(x)数据(y)寄存器 (CAN_FxDATAy) (x=0..27, y=0,1).....	531
22. 通用串行总线全速接口 (USBFS)	533
22.1. 概述.....	533
22.2. 主要特征.....	533
22.3. 结构框图.....	534
22.4. 信号线描述.....	534
22.5. 功能说明.....	534
22.5.1. USBFS 时钟及工作模式.....	534
22.5.2. USB 主机功能	536
22.5.3. USB 设备功能	538
22.5.4. OTG 功能概述.....	539
22.5.5. 数据 FIFO.....	539
22.5.6. 操作手册.....	542
22.6. 中断.....	545
22.7. USBFS 寄存器	547
22.7.1. 全局控制与状态寄存器组	547
22.7.2. 主机控制和状态寄存器	568
22.7.3. 设备控制和状态寄存器	579
22.7.4. 电源和时钟控制寄存器 (USBFS_PWRCLKCTL)	602
23. 文档附录	603
23.1. 寄存器表中使用的缩写列表	603
23.2. 术语表	603
23.3. 可用外设	603
24. 版本历史	604

图索引

图 1-1. Cortex®-M4 结构框图	25
图 1-2. GD32C10x 器件的系统架构示意图.....	27
图 2-1. 页擦除操作流程	40
图 2-2. 整片擦除操作流程	41
图 2-3. 字编程操作流程	43
图 3-1. 电源域概览.....	55
图 3-2. 上电 / 掉电复位波形图.....	57
图 3-3. LVD 阈值波形图	57
图 5-1. 系统复位电路.....	68
图 5-2. 时钟树	69
图 5-3. HXTAL 时钟源.....	70
图 6-1. CTC 结构框图.....	105
图 6-2. CTC 校准计数器.....	106
图 7-1. EXTI 框图.....	116
图 8-1. 标准 I/O 端口位的基本结构.....	122
图 8-2. 输入配置	123
图 8-3. 输出配置	124
图 8-4. 模拟配置	125
图 8-5. 备用功能配置.....	125
图 9-1. CRC 计算单元框图.....	151
图 10-1. DMA 结构框图.....	155
图 10-2. 握手机制	157
图 10-3. DMA 中断逻辑图	159
图 10-4. DMA0 请求映射.....	160
图 10-5. DMA1 请求映射.....	161
图 12-1. ADC 模块框图	177
图 12-2. 单次转换模式.....	178
图 12-3. 连续转换模式.....	179
图 12-4. 扫描转换模式，且连续转换模式失能.....	180
图 12-5. 扫描转换模式，连续转换模式使能.....	181
图 12-6. 间断转换模式.....	181
图 12-7. 自动注入， $CNT=1$	182
图 12-8. 触发注入	183
图 12-9. 12 位数据对齐	183
图 12-10. 6 位数据对齐	184
图 12-11. 20 位到 16 位的结果截断.....	186
图 12-12. 右移 5 位和取整的数例	187
图 12-13. ADC 同步框图	188
图 12-14. 基于 16 个通道的规则并行模式.....	189
图 12-15. 4 个通道的注入并行模式	189
图 12-16. 一个采用连续转换模式通道上的快速交叉模式	190

图 12-17. 一个通道上的慢速交叉模式	191
图 12-18. 交替触发: 注入通道组	191
图 12-19. 交替触发: 间断模式下注入通道组	192
图 12-20. 规则并行和交替触发组合模式	192
图 12-21. 在注入转换过程中触发出现	193
图 12-22. 交叉的单通道转换被注入序列 CH1 和 CH2 中断	193
图 13-1. DAC 结构框图	208
图 13-2. DAC LFSR 算法	210
图 13-3. DAC 三角噪声模式生成的波形	211
图 14-1. 独立看门狗定时器框图	222
图 14-2. 窗口看门狗定时器框图	227
图 14-3. 窗口看门狗定时器时序图	228
图 15-1. RTC 框图	233
图 15-2. RTC 秒信号及闹钟信号的波形 (RTC_PSC = 3, RTC_ALRM = 2)	234
图 15-3. RTC 秒信号及溢出信号的波形 (RTC_PSC = 3)	234
图 16-1. 高级定时器结构框图	241
图 16-2. 内部时钟分频为 1 时, 正常模式下的控制电路	242
图 16-3. 当预分频器的参数从 1 变到 2 时, 计数器的时序图	243
图 16-4. 向上计数时序图, PSC=0/1	244
图 16-5. 向上计数时序图, 在运行时改变 TIMERx_CAR 寄存器的值	244
图 16-6. 向下计数时序图, PSC=0/1	245
图 16-7. 向下计数时序图, 在运行时改变 TIMERx_CAR 寄存器值	246
图 16-8. 中央计数模式计数器时序图	247
图 16-9. 中央计数模式下计数器重复时序图	248
图 16-10. 在向上计数模式下计数器重复时序图	248
图 16-11. 在向下计数模式下计数器重复时序图	249
图 16-12. 输入捕获逻辑	249
图 16-13. 输出比较逻辑 (带有互补输出的通道, x=0,1,2)	250
图 16-14. 输出比较逻辑	251
图 16-15. 三种输出比较模式	252
图 16-16. EAPWM 时序图	253
图 16-17. CAPWM 时序图	253
图 16-18. 带死区时间的互补输出	256
图 16-19. 通道响应中止输入 (高电平有效) 时, 输出信号的行为	257
图 16-20. 编码器接口模式下计数器运行例子	258
图 16-21. CI0FE0 极性反相的编码器接口模式下的例子	258
图 16-22. 霍尔传感器用在 BLDC 电机控制中	259
图 16-23. 两个定时器之间的霍尔传感器时序图	260
图 16-24. 复位模式	261
图 16-25. 暂停模式	261
图 16-26. 事件模式	262
图 16-27. 单脉冲模式, TIMERx_CHxCV = 0x04, TIMERx_CAR=0x60	262
图 16-28. 定时器 0 主/从模式的例子	263
图 16-29. 用定时器 2 的使能信号触发定时器 0	264

图 16-30. 用定时器 2 的更新事件来触发定时器 0	265
图 16-31. 用定时器 2 的使能信号来控制定时器 0 的暂停模式	265
图 16-32. 用定时器 2 的 O0CPRE 信号控制定时器 0 的暂停模式	266
图 16-33. 用定时器 2 的 CI0 信号来触发定时器 0 和定时器 2	267
图 16-34. 通用定时器 L0 结构框图	293
图 16-35. 内部时钟分频为 1 时正常模式下的控制电路	294
图 16-36. 当预分频器的参数从 1 变到 2 时，计数器的时序图	295
图 16-37. 向上计数时序图，PSC=0/1	296
图 16-38. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值	296
图 16-39. 向下计数时序图，PSC=0/1	297
图 16-40. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值	298
图 16-41. 中央计数模式计数器时序图	299
图 16-42. 输入捕获逻辑	300
图 16-43. 输出比较逻辑 (x=0,1,2,3)	301
图 16-44. 三种输出比较模式	302
图 16-45. EAPWM 时序图	303
图 16-46. CAPWM 时序图	303
图 16-47. 编码器接口模式下计数器运行例子	304
图 16-48. CI0FE0 极性反相的编码器接口模式下的例子	305
图 16-49. 复位模式	306
图 16-50. 暂停模式	306
图 16-51. 事件模式	307
图 16-52. 单脉冲模式，TIMERx_CHxCV = 0x04, TIMERx_CAR=0x60	307
图 16-53. 通用定时器 L1 结构框图	330
图 16-54. 内部时钟分频为 1 时正常模式下的控制电路	331
图 16-55. 当预分频器的参数从 1 变到 2 时，计数器的时序图	332
图 16-56. 向上计数时序图，PSC=0/1	333
图 16-57. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值	333
图 16-58. 输入捕获逻辑	334
图 16-59. 输出比较逻辑 (x=0,1)	335
图 16-60. 三种输出比较模式	336
图 16-61. EAPWM 时序图	337
图 16-62. CAPWM 时序图	337
图 16-63. 复位模式	338
图 16-64. 暂停模式	339
图 16-65. 事件模式	339
图 16-66. 单脉冲模式，TIMERx_CHxCV = 0x04, TIMERx_CAR=0x60	340
图 16-67. 通用定时器 L2 结构框图	353
图 16-68. 内部时钟分频为 1 时正常模式下的控制电路	354
图 16-69. 当预分频器的参数从 1 变到 2 时，计数器的时序图	355
图 16-70. 向上计数时序图，PSC=0/1	356
图 16-71. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值	356
图 16-72. 输入捕获逻辑	357
图 16-73. 输出比较逻辑	358

图 16-74. 三种输出比较模式	359
图 16-75. 基本定时器结构框图	371
图 16-76. 内部时钟分频为 1 时正常模式下的控制电路	372
图 16-77. 当预分频器的参数从 1 变到 2 时，计数器的时序图	372
图 16-78. 向上计数时序图， $PSC=0/1$	373
图 16-79. 向上计数时序图，在运行时改变 $TIMERx_CAR$ 寄存器的值	374
图 17-1. USART 模块内部框图	382
图 17-2. USART 字符帧 (8 数据位和 1 停止位)	382
图 17-3. USART 发送步骤	384
图 17-4. 过采样方式接收一个数据位	385
图 17-5. 采用 DMA 方式实现 USART 数据发送配置步骤	386
图 17-6. 采用 DMA 方式实现 USART 数据接收配置步骤	387
图 17-7. 两个 USART 之间的硬件流控制	387
图 17-8. 硬件流控制	388
图 17-9. 空闲状态下检测断开帧	389
图 17-10. 数据传输过程中检测断开帧	389
图 17-11. 同步模式下的 USART 示例	390
图 17-12. 8-bit 格式的 USART 同步通信波形($CLEN=1$)	390
图 17-13. IrDA SIR ENDEC 模块	390
图 17-14. IrDA 数据调制	391
图 17-15. ISO7816-3 数据帧格式	392
图 17-16. USART 中断映射框图	394
图 18-1. I2C 模块框图	409
图 18-2. 数据有效性	410
图 18-3. 开始和停止状态	410
图 18-4. 时钟同步	410
图 18-5. SDA 线仲裁	411
图 18-6. 7 位地址的 I2C 通讯流程	411
图 18-7. 10 位地址的 I2C 通讯流程（主机发送）	412
图 18-8. 10 位地址的 I2C 通讯流程（主机接收）	412
图 18-9. 从机发送模式	413
图 18-10. 从机接收模式	414
图 18-11. 主机发送模式	416
图 18-12. 主机接收使用方案 A 模式	417
图 18-13. 主机接收使用方案 B 模式	419
图 19-1. SPI 结构框图	434
图 19-2. 常规模式下的 SPI 时序图	435
图 19-3. SPI 四线模式下的 SPI 时序图($CKPL=1, CKPH=1, LF=0$)	436
图 19-4. 典型的全双工模式连接	438
图 19-5. 典型的单工模式连接（主机：接收，从机：发送）	438
图 19-6. 典型的单工模式连接（主机：只发送，从机：接收）	438
图 19-7. 典型的双向线连接	438
图 19-8. 主机 TI 模式在不连续发送时的时序图	440
图 19-9. 主机 TI 模式在连续发送时的时序图	440

图 19-10. 从机 TI 模式时序图.....	441
图 19-11. NSS 脉冲模式时序图（主机连续发送）.....	442
图 19-12. SPI 四线模式四线写操作时序图.....	443
图 19-13. SPI 四路模式四路读操作时序图.....	444
图 19-14. I2S 结构框图	447
图 19-15. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=0, CKPL=0）	448
图 19-16. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=0, CKPL=1）	448
图 19-17. I2S 飞利浦标准时序图（DTLEN=10, CHLEN=1, CKPL=0）	448
图 19-18. I2S 飞利浦标准时序图（DTLEN=10, CHLEN=1, CKPL=1）	448
图 19-19. I2S 飞利浦标准时序图（DTLEN=01, CHLEN=1, CKPL=0）	449
图 19-20. I2S 飞利浦标准时序图（DTLEN=01, CHLEN=1, CKPL=1）	449
图 19-21. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=1, CKPL=0）	449
图 19-22. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=1, CKPL=1）	449
图 19-23. MSB 对齐标准时序图（DTLEN=00, CHLEN=0, CKPL=0）	450
图 19-24. MSB 对齐标准时序图（DTLEN=00, CHLEN=0, CKPL=1）	450
图 19-25. MSB 对齐标准时序图（DTLEN=10, CHLEN=1, CKPL=0）	450
图 19-26. MSB 对齐标准时序图（DTLEN=10, CHLEN=1, CKPL=1）	450
图 19-27. MSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=0）	450
图 19-28. MSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=1）	450
图 19-29. MSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=0）	451
图 19-30. MSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=1）	451
图 19-31. LSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=0）	451
图 19-32. LSB 对齐标准时序图（DTLEN=01, CHLEN=1, CKPL=1）	451
图 19-33. LSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=0）	452
图 19-34. LSB 对齐标准时序图（DTLEN=00, CHLEN=1, CKPL=1）	452
图 19-35. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=0）	452
图 19-36. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=1）	452
图 19-37. PCM 标准短帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=0）	452
图 19-38. PCM 标准短帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=1）	453
图 19-39. PCM 标准短帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=0）	453
图 19-40. PCM 标准短帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=1）	453
图 19-41. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=0）	453
图 19-42. PCM 标准短帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=1）	453
图 19-43. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=0）	453
图 19-44. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=0, CKPL=1）	454
图 19-45. PCM 标准长帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=0）	454
图 19-46. PCM 标准长帧同步模式时序图（DTLEN=10, CHLEN=1, CKPL=1）	454
图 19-47. PCM 标准长帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=0）	454
图 19-48. PCM 标准长帧同步模式时序图（DTLEN=01, CHLEN=1, CKPL=1）	454
图 19-49. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=0）	454
图 19-50. PCM 标准长帧同步模式时序图（DTLEN=00, CHLEN=1, CKPL=1）	455
图 19-51. I2S 时钟生成结构框图	455
图 20-1. EXMC 结构框图	471
图 20-2. EXMC Bank 划分	472

图 20-3. 模式 1 读访问.....	475
图 20-4. 模式 1 写访问.....	475
图 20-5. 模式 A 读访问.....	477
图 20-6. 模式 A 写访问.....	477
图 20-7. 模式 2/B 读访问.....	478
图 20-8. 模式 2 写访问.....	479
图 20-9. 模式 B 写访问.....	479
图 20-10. 模式 C 读访问.....	480
图 20-11. 模式 C 写访问.....	481
图 20-12. 模式 D 读访问.....	482
图 20-13. 模式 D 写访问.....	482
图 20-14. 复用模式读访问.....	484
图 20-15. 复用模式写访问.....	484
图 20-16. 异步等待有效时的读时序.....	486
图 20-17. 异步等待有效时的写时序.....	486
图 20-18. 同步复用突发传输读时序.....	488
图 20-19. 同步复用突发传输写时序.....	489
图 21-1. CAN 模块结构框图	497
图 21-2. 发送寄存器.....	499
图 21-3. 发送邮箱状态转换	500
图 21-4. 接收寄存器.....	501
图 21-5. 32-bit 位宽过滤器	502
图 21-6. 16-bit 位宽过滤器	502
图 21-7. 32-bit 位宽掩码模式过滤器	503
图 21-8. 16-bit 位宽掩码模式过滤器	503
图 21-9. 32-bit 位宽列表模式过滤器	503
图 21-10. 16-bit 位宽列表模式过滤器	503
图 21-11. 位时序	506
图 21-12 传输延迟测量.....	508
图 22-1. USBFS 结构框图	534
图 22-2. 在主机或设备模式下连接示意图	535
图 22-3. OTG 模式下连接示意图	536
图 22-4. 主机端口状态转移图	536
图 22-5. 主机模式 FIFO 空间	540
图 22-6. 主机模式 FIFO 访问寄存器映射表	541
图 22-7. 设备模式 FIFO 空间	541
图 22-8. 设备模式 FIFO 访问寄存器映射表	542

表索引

表 1-1. AHB 互联矩阵的互联关系列表	25
表 1-2. GD32C10x 系列器件的存储器映射表	28
表 1-3. 引导模式.....	32
表 2-1. GD32C10x 闪存基地址和构成	36
表 2-2. WSCNT 与 AHB 时钟频率对应关系	37
表 2-3. 选项字节.....	44
表 3-1. 节电模式总结	59
表 5-1. 时钟输出 0 的时钟源选择	72
表 5-2. 深度睡眠模式下 1.2V 域电压选择	73
表 7-1. Cortex-M3 中的 NVIC 异常类型	113
表 7-2. 中断向量表	114
表 7-3. EXTI 触发源	117
表 8-1. GPIO 配置表	121
表 8-2. 调试接口信号	126
表 8-3. 调试端口映射 ⁽¹⁾	127
表 8-4. ADC0 注入转换外部触发备用功能重映射	127
表 8-5. ADC0 规则转换外部触发备用功能重映射	127
表 8-6. ADC1 注入转换外部触发备用功能重映射	127
表 8-7. ADC1 规则转换外部触发备用功能重映射	127
表 8-8. TIMER0 备用功能重映射	128
表 8-9. TIMER1 备用功能重映射.....	128
表 8-10. TIMER2 备用功能重映射 ⁽¹⁾	128
表 8-11. TIMER3 备用功能重映射 ⁽¹⁾	128
表 8-12. TIMER4 备用功能重映射	129
表 8-13. TIMER8 备用功能重映射 ⁽¹⁾	129
表 8-14. USART0 备用功能重映射	129
表 8-15. USART1 备用功能重映射	129
表 8-16. USART2 备用功能重映射	129
表 8-17. I2C0 备用功能重映射	130
表 8-18. SPI0 备用功能重映射	130
表 8-19. SPI2/I2S2 备用功能重映射	130
表 8-20. CAN0 备用功能重映射	130
表 8-21. CAN1 备用功能重映射	131
表 8-22. CTC 备用功能重映射	131
表 8-23. OSC32 引脚配置.....	131
表 8-24. OSC 引脚配置.....	131
表 10-1. DMA 传输操作	155
表 10-2. 中断事件.....	158
表 10-3. DMA0 各通道请求表	160
表 10-4. DMA1 各通道请求表	161
表 12-1. ADC 内部信号	176

表 12-2. ADC 引脚定义	176
表 12-3. ADC0 和 ADC1 的规则通道的外部触发	184
表 12-4. ADC0 和 ADC1 的注入通道的外部触发	185
表 12-5. 不同分辨率对应的 t_{CONV} 时间	185
表 12-6. 不同 N 和 M 组合的最大输出值（灰色值表示截断）	187
表 13-1. DAC 引脚	209
表 13-2. DAC 外部触发	209
表 14-1. 独立看门狗定时器在 40kHz (IRC40K)时的最小/最大超时周期	222
表 14-2. 在 60MHz (f_{PCLK1})时的最大/最小超时值	228
表 16-1. 定时器 (TIMERx) 分为五种类型	239
表 16-2. 由参数控制的互补输出表	254
表 16-3. 计数方向与编码器信号之间的关系	257
表 16-4. 从模式示例	260
表 16-5. 计数方向与编码器信号之间的关系	304
表 16-6. 从模式示例 (通用定时器 L0)	305
表 16-7. 从机模式示例 (通用定时器 L1)	338
表 17-1. USART 重要引脚描述	381
表 17-2. 停止位配置	382
表 17-3. USART 中断请求	393
表 18-1. I2C 总线术语说明 (参考飞利浦 I2C 规范)	409
表 18-2. 事件状态标志位	422
表 18-3. I2C 错误标志位	422
表 19-1. SPI 信号描述	434
表 19-2. SPI 四线信号描述	435
表 19-3. SPI 运行模式	437
表 19-4. SPI 中断请求	446
表 19-5. I2S 比特率计算公式	455
表 19-6. 音频采样频率计算公式	455
表 19-7. 各种运行模式下 I2S 接口信号的方向	456
表 19-8. I2S 中断	459
表 20-1. NOR Flash 接口信号描述	472
表 20-2. PSRAM 非复用接口信号描述	473
表 20-3. EXMC 的 Bank0 支持的所有传输	473
表 20-4. NOR/PSRAM 控制时序参数	474
表 20-5. EXMC 时序模型	474
表 20-6. 模式 1 相关寄存器配置	476
表 20-7. 模式 A 相关寄存器配置	477
表 20-8. 模式 2/B 相关寄存器配置	479
表 20-9. 模式 C 相关寄存器配置	481
表 20-10. 模式 D 相关寄存器配置	483
表 20-11. 复用模式相关寄存器配置	484
表 20-12. 同步复用模式读时序配置	488
表 20-13. 同步复用模式写时序配置	489
表 21-1. 32-bit 过滤序号	503

表 21-2. 过滤索引.....	504
表 22-1. USBFS 信号线描述.....	534
表 22-2. USBFS 全局中断	545
表 23-1. 寄存器功能位访问属性.....	603
表 23-2. 术语	603
表 24-1. 版本历史.....	604

1. 系统及存储器架构

GD32C10x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。

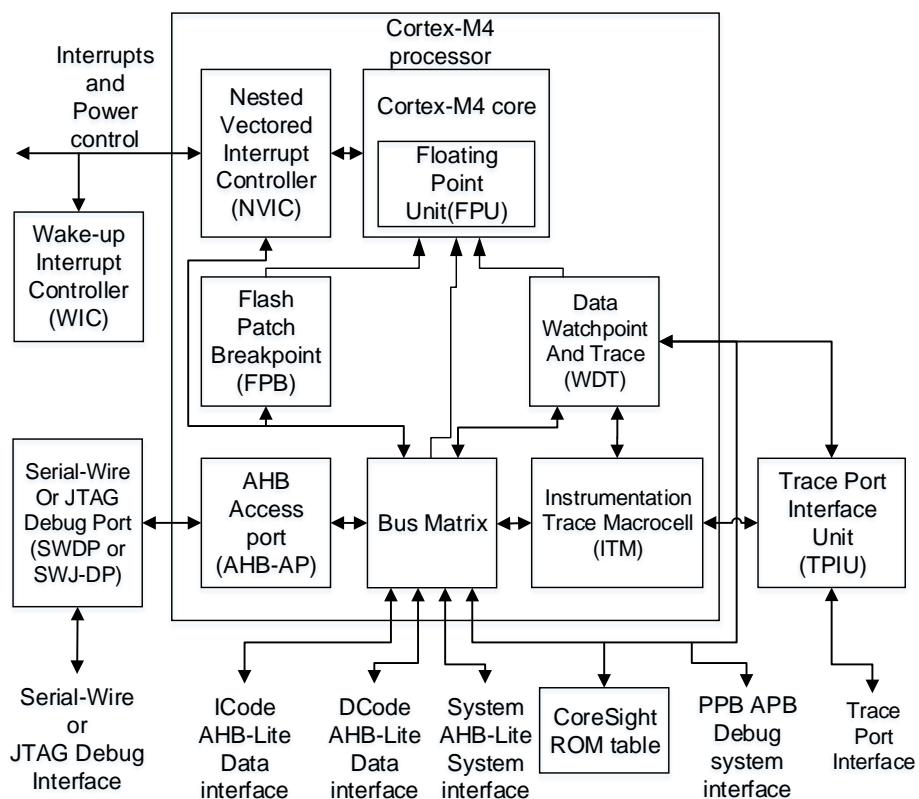
1.1. ARM Cortex-M4 处理器

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、DSP(数字信号处理)和浮点运算指令。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)；
- 浮点运算单元 (FPU)。

[图 1-1. Cortex®-M4 结构框图](#)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 结构框图



1.2. 系统架构

GD32C10x 系列器件采用 32 位多层次总线结构，该结构可使系统中的多个主机和从机之间的并行通信成为可能。多层次总线结构包括一个 AHB 互联矩阵、两个 AHB 总线和两个 APB 总线。
AHB 互联矩阵的互联关系接下来将进行说明。在 [表 1-1. AHB 互联矩阵的互联关系列表](#)中，“1”表示相应的主机可以通过 AHB 互联矩阵访问对应的从机，空白的单元格表示相应的主机不可通过 AHB 互联矩阵访问对应的从机。

表 1-1. AHB 互联矩阵的互联关系列表

	IBUS	DBUS	SBUS	DMA0	DMA1
FMC-I	1				
FMC-D		1		1	1
SRAM	1	1	1	1	1
EXMC	1	1	1	1	1
AHB			1	1	1
APB1			1	1	1
APB2			1	1	1

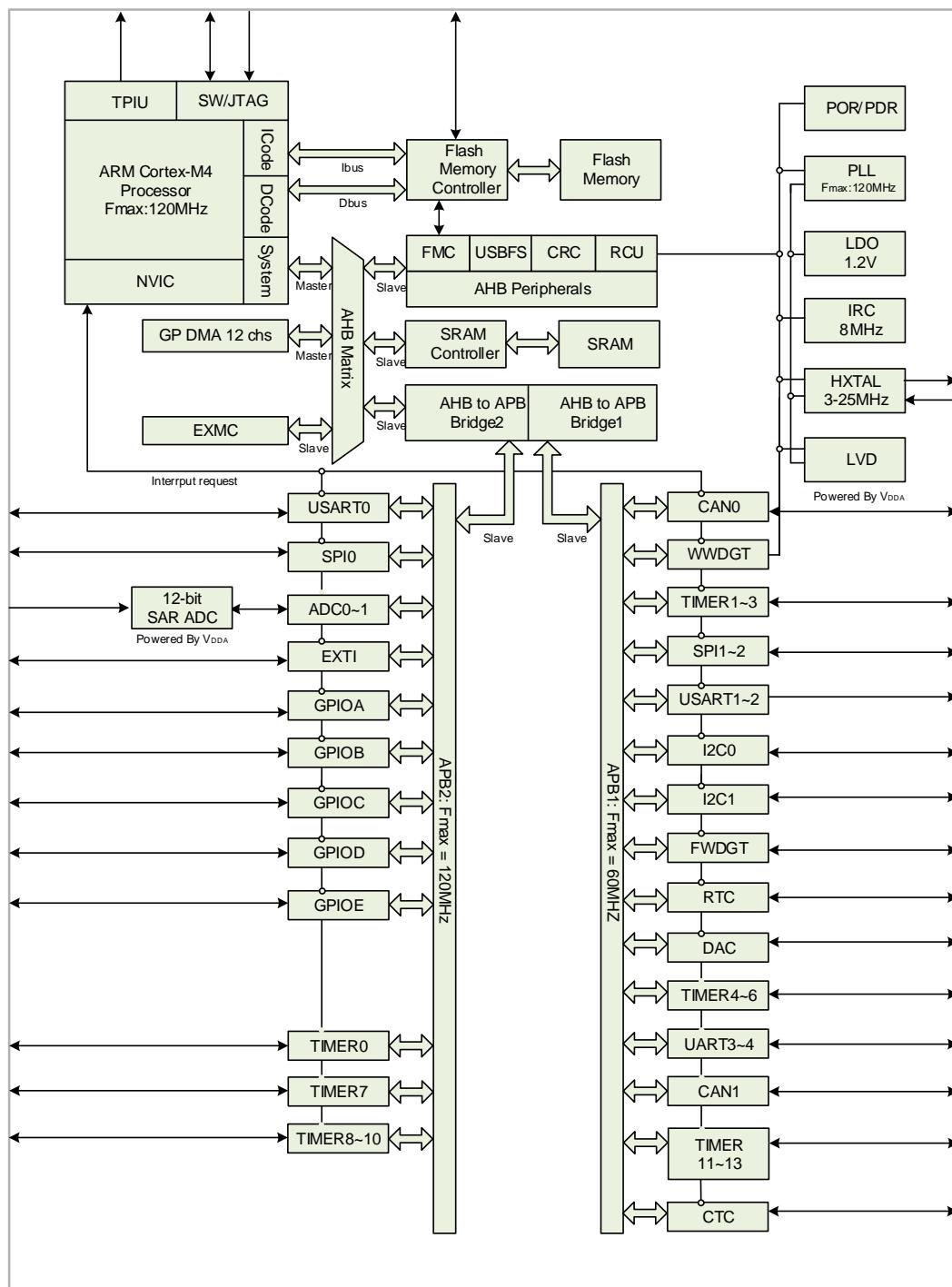
如上表所示，AHB互联矩阵共连接11个主机，分别为：IBUS、DBUS、SBUS、DMA0、DMA1。

IBUS是Cortex®-M4内核的指令总线，用于从代码区域(0x0000 0000~0x1FFF FFFF)中取指令和向量。DBUS是Cortex®-M4内核的数据总线，用于加载和存储数据，以及代码区域的调试访问。同样，SBUS是Cortex®-M4内核的系统总线，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。DMA0和DMA1分别是DMA0和DMA1的存储器总线。

AHB互联矩阵也连接了12个从机，分别为：FMC-I、FMC-D、SRAM、EXMC、AHB、APB1和APB2。FMC-I是闪存存储器控制器的指令总线，而FMC-D是闪存存储器的数据总线。SRAM是片上静态随机存取存储器。EXMC是外部存储器控制器。AHB是连接所有AHB从机的AHB总线，APB1和APB2是连接所有APB从机的两条APB总线。APB1速度限制为60MHz，APB2可以全速运行（最高可到120MHz）。

GD32C10x 系列器件的系统架构如 [图 1-2. GD32C10x 器件的系统架构示意图](#) 所示。

图 1-2. GD32C10x 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。在存储器映射表中，一部分地址空间由 ARM® Cortex®-M4 的系统外设所占用，且不可更改。此外，其余部分地址空间可由芯片供应商定义使用。[表 1-2. GD32C10x 系列器件的存储器映射表](#)显示了 GD32C10x 系列器件的存储器映射，包括代码、SRAM、外设和其他预先定义的区域。几乎每个外设都分配了 1KB 的地址空间，这样可以简化每个外设的地址译码。

表 1-2. GD32C10x 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
外部设备	AHB3	0xA000 0000 - 0xA000 0FFF	EXMC - SWREG
外部 RAM		0x9000 0000 - 0x9FFF FFFF	保留
		0x7000 0000 - 0x8FFF FFFF	保留
		0x6000 0000 - 0x63FF FFFF	EXMC - NOR/PSRAM/SRAM
Peripheral	AHB1	0x5000 0000 - 0x5003 FFFF	USBFS
		0x4008 0000 - 0x4FFF FFFF	保留
		0x4004 0000 - 0x4007 FFFF	保留
		0x4002 BC00 - 0x4003 FFFF	保留
		0x4002 B000 - 0x4002 BBFF	保留
		0x4002 A000 - 0x4002 AFFF	保留
		0x4002 8000 - 0x4002 9FFF	保留
		0x4002 6800 - 0x4002 7FFF	保留
		0x4002 6400 - 0x4002 67FF	保留
		0x4002 6000 - 0x4002 63FF	保留
		0x4002 5000 - 0x4002 5FFF	保留
		0x4002 4000 - 0x4002 4FFF	保留
		0x4002 3C00 - 0x4002 3FFF	保留
		0x4002 3800 - 0x4002 3BFF	保留
		0x4002 3400 - 0x4002 37FF	保留
		0x4002 3000 - 0x4002 33FF	CRC
		0x4002 2C00 - 0x4002 2FFF	保留
		0x4002 2800 - 0x4002 2BFF	保留
		0x4002 2400 - 0x4002 27FF	保留
		0x4002 2000 - 0x4002 23FF	FMC
		0x4002 1C00 - 0x4002 1FFF	保留
		0x4002 1800 - 0x4002 1BFF	保留
		0x4002 1400 - 0x4002 17FF	保留
		0x4002 1000 - 0x4002 13FF	RCU
		0x4002 0C00 - 0x4002 0FFF	保留
		0x4002 0800 - 0x4002 0BFF	保留
		0x4002 0400 - 0x4002 07FF	DMA1
		0x4002 0000 - 0x4002 03FF	DMA0
		0x4001 8400 - 0x4001 FFFF	保留

预先定义的地址空间	总线	地址范围	外设
APB2		0x4001 8000 - 0x4001 83FF	保留
		0x4001 7C00 - 0x4001 7FFF	保留
		0x4001 7800 - 0x4001 7BFF	保留
		0x4001 7400 - 0x4001 77FF	保留
		0x4001 7000 - 0x4001 73FF	保留
		0x4001 6C00 - 0x4001 6FFF	保留
		0x4001 6800 - 0x4001 6BFF	保留
		0x4001 5C00 - 0x4001 67FF	保留
		0x4001 5800 - 0x4001 5BFF	保留
		0x4001 5400 - 0x4001 57FF	TIMER10
		0x4001 5000 - 0x4001 53FF	TIMER9
		0x4001 4C00 - 0x4001 4FFF	TIMER8
		0x4001 4800 - 0x4001 4BFF	保留
		0x4001 4400 - 0x4001 47FF	保留
		0x4001 4000 - 0x4001 43FF	保留
		0x4001 3C00 - 0x4001 3FFF	保留
		0x4001 3800 - 0x4001 3BFF	USART0
		0x4001 3400 - 0x4001 37FF	TIMER7
		0x4001 3000 - 0x4001 33FF	SPI0
		0x4001 2C00 - 0x4001 2FFF	TIMER0
		0x4001 2800 - 0x4001 2BFF	ADC1
		0x4001 2400 - 0x4001 27FF	ADC0
		0x4001 2000 - 0x4001 23FF	保留
		0x4001 1C00 - 0x4001 1FFF	保留
APB1		0x4001 1800 - 0x4001 1BFF	GPIOE
		0x4001 1400 - 0x4001 17FF	GPIOD
		0x4001 1000 - 0x4001 13FF	GPIOC
		0x4001 0C00 - 0x4001 0FFF	GPIOB
		0x4001 0800 - 0x4001 0BFF	GPIOA
		0x4001 0400 - 0x4001 07FF	EXTI
		0x4001 0000 - 0x4001 03FF	AFIO
		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	保留
		0x4000 7400 - 0x4000 77FF	DAC

预先定义的地址空间	总线	地址范围	外设
		0x4000 7000 - 0x4000 73FF	PMU
		0x4000 6C00 - 0x4000 6FFF	BKP
		0x4000 6800 - 0x4000 6BFF	CAN1
		0x4000 6400 - 0x4000 67FF	CAN0
		0x4000 6000 - 0x4000 63FF	CAN SRAM 1K bytes
		0x4000 5C00 - 0x4000 5FFF	保留
		0x4000 5800 - 0x4000 5BFF	I2C1
		0x4000 5400 - 0x4000 57FF	I2C0
		0x4000 5000 - 0x4000 53FF	UART4
		0x4000 4C00 - 0x4000 4FFF	UART3
		0x4000 4800 - 0x4000 4BFF	USART2
		0x4000 4400 - 0x4000 47FF	USART1
		0x4000 4000 - 0x4000 43FF	保留
		0x4000 3C00 - 0x4000 3FFF	SPI2/I2S2
		0x4000 3800 - 0x4000 3BFF	SPI1/I2S1
		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 1C00 - 0x4000 1FFF	TIMER12
		0x4000 1800 - 0x4000 1BFF	TIMER11
		0x4000 1400 - 0x4000 17FF	TIMER6
		0x4000 1000 - 0x4000 13FF	TIMER5
		0x4000 0C00 - 0x4000 0FFF	TIMER4
		0x4000 0800 - 0x4000 0BFF	TIMER3
		0x4000 0400 - 0x4000 07FF	TIMER2
		0x4000 0000 - 0x4000 03FF	TIMER1
SRAM	AHB	0x2007 0000 - 0x3FFF FFFF	保留
		0x2006 0000 - 0x2006 FFFF	保留
		0x2003 0000 - 0x2005 FFFF	保留
		0x2002 0000 - 0x2002 FFFF	保留
		0x2001 C000 - 0x2001 FFFF	保留
		0x2001 8000 - 0x2001 BFFF	保留
		0x2000 8000 - 0x2001 7FFF	保留
		0x2000 0000 - 0x2000 7FFF	SRAM
Code	AHB	0x1FFF F810 - 0x1FFF FFFF	保留
		0x1FFF F800 - 0x1FFF F80F	Option Bytes

预先定义的地址空间	总线	地址范围	外设
		0x1FFF F000 - 0x1FFF F7FF	Boot loader
		0x1FFF C010 - 0x1FFF EFFF	
		0x1FFF C000 - 0x1FFF C00F	
		0x1FFF B000 - 0x1FFF BFFF	
		0x1FFF 7A10 - 0x1FFF AFFF	
		0x1FFF 7800 - 0x1FFF 7A0F	
		0x1FFF 0000 - 0x1FFF 77FF	
		0x1FFE C010 - 0x1FFE FFFF	
		0x1FFE C000 - 0x1FFE C00F	
		0x1001 0000 - 0x1FFE BFFF	
		0x1000 0000 - 0x1000 FFFF	
		0x083C 0000 - 0x0FFF FFFF	
		0x0830 0000 - 0x083B FFFF	
		0x0810 0000 - 0x082F FFFF	
		0x0802 0000 - 0x080F FFFF	
		0x0800 0000 - 0x0801 FFFF	Main Flash
		0x0030 0000 - 0x07FF FFFF	保留
		0x0010 0000 - 0x002F FFFF	别名为 Main Flash 或 Boot loader
		0x0002 0000 - 0x000F FFFF	
		0x0000 0000 - 0x0001 FFFF	

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。一个是**SRAM**，另一个是片内外设区。这两个区域中的地址除了普通应用外，还有自己的“位带别名区”。位带别名区把每个比特扩展成一个32位的字。

下面的公式表明了位带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。

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

其中：

- `bit_word_addr`指的是位带区目标比特对应在位带别名区的地址；
- `bit_band_base`指的是位带别名区的起始地址；
- `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0x2000 401C：

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

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

GD32C10x系列微控制器含有高达32KB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。

1.3.3. 片上 FLASH 存储器概述

GD32C10x系列微控制器可以提供高密度片上FLASH存储器，按以下分类进行组织：

- 高达128KB主FLASH存储器；
- 高达18KB引导装载程序(boot loader)信息块存储器；
- 器件配置的选项字节。

更多详细说明请参考[闪存控制器\(FMC\)](#)章节。

1.4. 引导配置

GD32C10X 系列微控制器提供了三种引导源，可以通过 BOOT0 和 BOOT1 引脚来进行选择，详细说明见[表 1-3. 引导模式](#)。该两个引脚的电平状态会在复位后的第四个 CK_SYS(系统时钟)的上升沿进行锁存。用户可自行选择所需要的引导源，通过设置上电复位和系统复位后的 BOOT0 和 BOOT1 的引脚电平。一旦这两个引脚电平被采样，它们可以被释放并用于其他用途。

表 1-3. 引导模式

引导源选择	启动模式选择引脚	
	Boot1	Boot0
主FLASH存储器	x	0
引导装载程序	0	1
片上SRAM	1	1

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。

根据所选择的引导源，主FLASH存储器（开始于0x0800 0000的原始存储空间）或系统存储器（开始于0x1FFF F000的原始存储空间）被映射到引导存储空间（起始于0x0000 0000）。片上SRAM存储空间的起始地址是0x2000 0000，当它被选择为引导源时，在应用初始化代码中，你必须使用NVIC异常表和偏移寄存器来将向量表重定向到SRAM中。

嵌入式的Bootloader存放在系统存储空间，用于对FLASH存储器进行重新编程。在GD32C10x设备中，Bootloader可以通过USART0接口和外界交互。

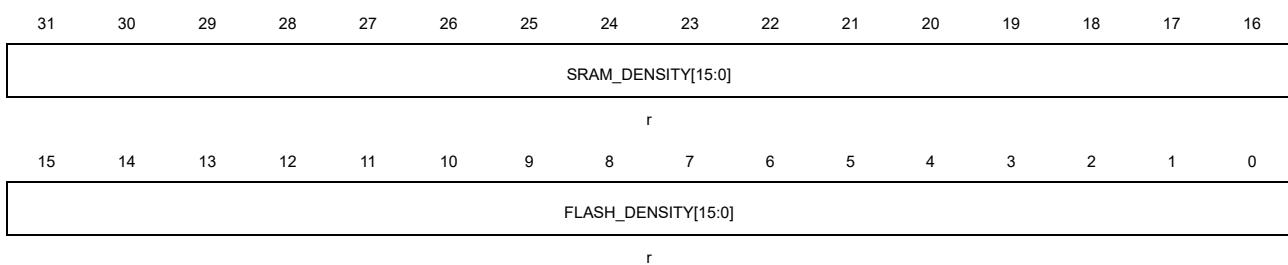
1.5. 设备电子签名

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，它位于Flash存储器的信息块中。96位设备唯一ID对任何设备来说都是独一无二的，可以用作序列号，或密钥的一部分，等等。

1.5.1. 存储器容量信息

基址址: 0x1FFF F7E0

其值出厂已设置，不可由用户更改。

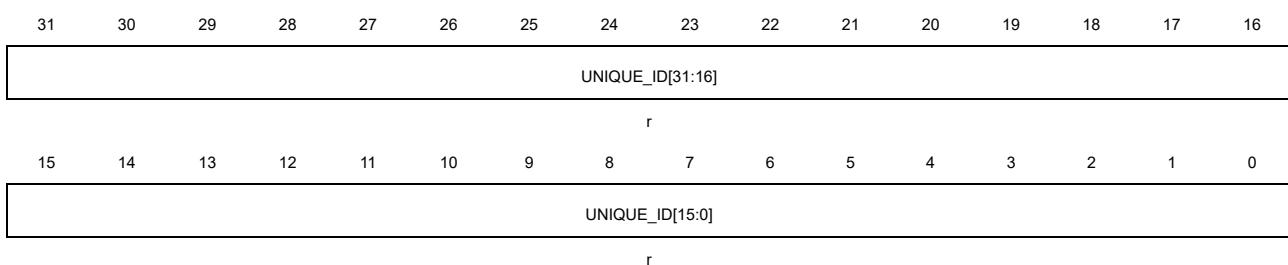


位/位域	名称	描述
31:16	SRAM_DENSITY[15:0] SRAM容量	该值表示器件的片上SRAM容量，以Kbytes为单位。 例如：0x0008表示8Kbytes。
15:0	FLASH_DENSITY[15:0] FLASH存储器容量	该值表示器件的FLASH存储器容量，以Kbytes为单位。 例如：0x0020表示32Kbytes。

1.5.2. 设备唯一 ID (96 位)

基址址: 0x1FFF F7E8

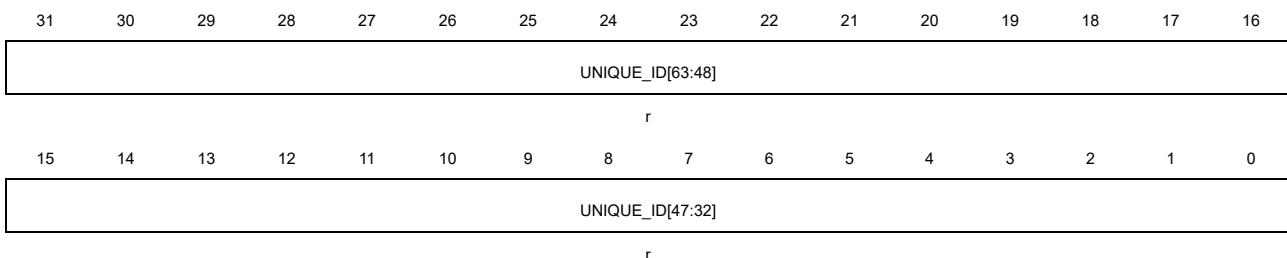
其值出厂已设置，不可由用户更改。



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

基地址: 0x1FFF F7EC

其值出厂已设置, 不可由用户更改。

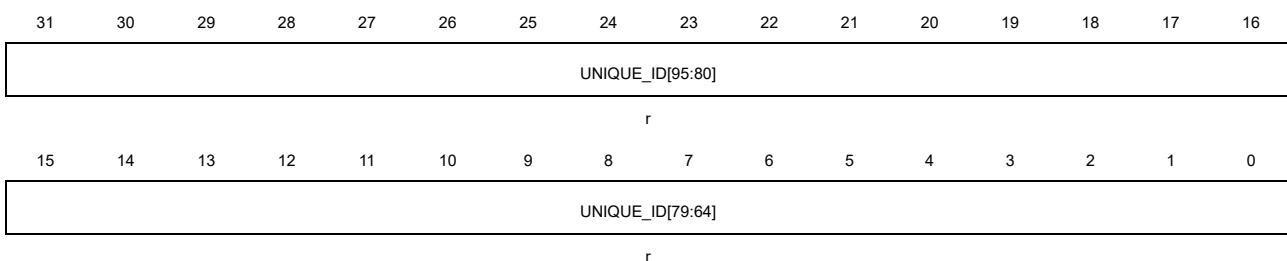


位/位域	名称	描述
------	----	----

31:0	UNIQUE_ID[63:32]	设备唯一ID
------	------------------	--------

基地址: 0x1FFF F7F0

其值出厂已设置, 不可由用户更改。

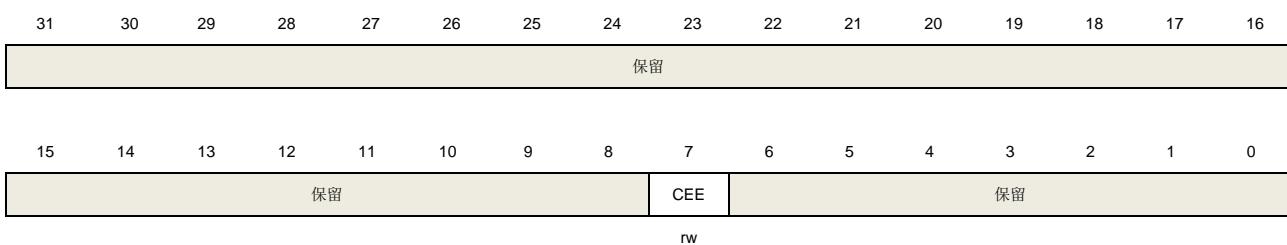


位/位域	名称	描述
------	----	----

31:0	UNIQUE_ID[95:64]	设备唯一ID
------	------------------	--------

1.6. 系统配置寄存器

基地址: 0x4002 103C



位/位域	名称	描述
------	----	----

7	CEE	代码执行效率
		0: 默认的代码执行效率
		1: 代码执行效率降低

注意：只有bit[7]可以被读-改-写，其他的位都不允许。

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128K字节空间内，CPU执行指令需要少量等待时间。FMC也提供了页擦除，整片擦除，以及编程操作。

2.2. 主要特征

- 高达128KB的片上闪存可用于存储指令或数据；
- 在闪存的前128K字节空间内，CPU执行指令需要0~3个等待时间；
- 预取缓冲区以加速读操作；
- IBUS高速缓存区为512字节，由32条缓存线组成，每条缓存线为2*64位；
- DBUS高速缓存区为256字节，由8条缓存线组成，每条缓存线为4*64位；
- 闪存页大小为1KB；
- 支持32位整字或双16位半字编程，页擦除和整片擦除操作；
- 大小为16字节的选项字节可根据用户需求配置；
- 512字节OTP块（一次性编程），用于存储用户数据；
- 当系统复位时，选项字节被上载到选项字节控制寄存器；
- 具有安全保护状态，可阻止对代码或数据的非法读访问；
- 具有擦除和编程保护状态，可阻止意外写操作。

2.3. 功能说明

2.3.1. 闪存结构

闪存页大小为1KB。主存储闪存的每页都可以单独擦除。闪存结构细节见下表。

表 2-1. GD32C10x 闪存基址和构成

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

	第127页	0x0801 FC00 - 0x0801 FFFF	1KB
信息块	引导装载程序	0x1FFF B000- 0x1FFF F7FF	18KB
选项字节块	选项字节	0x1FFF F800 - 0x1FFF F80F	16B
一次性编程块	OTP字节	0x1FFF_7000~0x1FFF_71FF	512B

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。

增加等待状态:

根据AHB时钟频率，正确配置FMC_WS寄存器中的WSCNT位。WSCNT位和AHB时钟频率对应关系见下表。

表 2-2. WSCNT 与 AHB 时钟频率对应关系

AHB 时钟频率	WSCNT 配置
<= 30MHz	0 (0等待状态增加)
<= 60MHz	1 (1等待状态增加)
<= 90MHz	2 (2等待状态增加)
<= 120MHz	3 (3等待状态增加)

如果发生系统复位，AHB时钟频率为8MHz，此时WSCNT置为0。

注意:

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

当前缓存区：

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

预取缓存区：

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

IBUS 高速缓存区：

置位FMC_WS寄存器中ICEN位来使能IBUS高速缓存区。IBUS高速缓存区仅在IBUS取数据时使用。IBUS快速缓存区为512字节，由32条缓存线组成，每条缓存线为 2^*64 位。

当IBUS数据存在于IBUS高速缓存区时，CPU从IBUS高速缓存区读取数据时无延迟。当IBUS数据不存在于IBUS高速缓存区并且也不存在于当前缓存区/预取缓存区时，高速缓存线从flash闪存中读取数据并复制到IBUS高速缓存区。当所有高速缓存线被填充，LRU（最近最少使用）策略被用于转移高速缓存线中数据。

DBUS 高速缓存区：

置位FMC_WS寄存器中DCEN位来使能IBUS高速缓存区。DBUS高速缓存区仅在CPU通过DBUS取数据（不是通过DMA）时使用，此时选项字节不可缓存。DBUS快速缓存区为256字节，由8条缓存线组成，每条缓存线为 4^*64 位。

当DBUS数据存在于DBUS高速缓存区时，CPU从DBUS高速缓存区读取数据时无延迟。当DBUS数据不存在于DBUS高速缓存区并且也不存在于当前缓存区时，高速缓存线从flash闪存中读取数据并复制到DBUS高速缓存区。当所有高速缓存线被填充，LRU（最近最少使用）策略被用于转移高速缓存线中数据。

2.3.3. FMC_CTL 寄存器解锁

复位后，FMC_CTL寄存器进入写操作锁定状态，LK位复位后置为1。通过先后向FMC_KEY寄存器写入0x45670123和0xCDEF89AB，可以使得FMC_CTL寄存器解锁。两次写操作后，FMC_CTL寄存器的LK位被硬件清0。可以通过软件设置FMC_CTL寄存器的LK位为1再次锁定FMC_CTL寄存器。任何对FMC_KEY寄存器的错误操作都会将LK位置1，从而锁定FMC_CTL寄存器，并引发一个总线错误。

FMC_CTL寄存器的OBPG位和OBER位在FMC_CTL寄存器解锁后，仍然被保护。向FMC_OBKEY寄存器先后写入0x45670123和0xCDEF89AB，然后硬件将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。

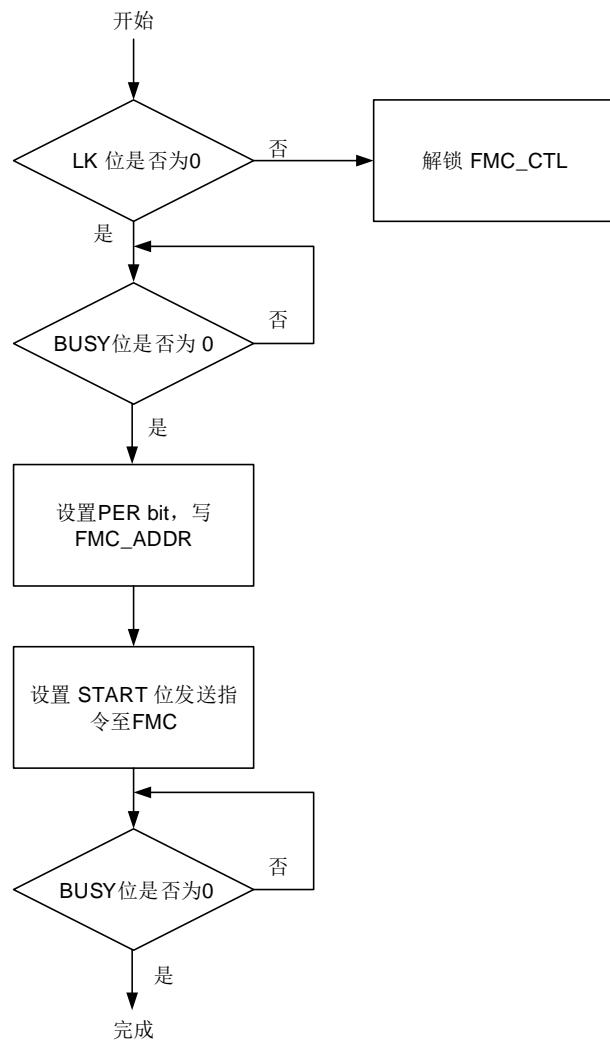
2.3.4. 页擦除

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

- 确保FMC_CTL寄存器不处于锁定状态；
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0；
- 置位FMC_CTL寄存器的PER位；
- 将待擦除页的绝对地址（0x08XX XXXX）写到FMC_ADDR寄存器；
- 通过将FMC_CTL寄存器的START位置1来发送页擦除命令到FMC；
- 等待擦除指令执行完毕，FMC_STAT寄存器的BUSY位清0；
- 如果需要，使用DBUS读并验证该页是否擦除成功。

当页擦除成功执行，FMC_STAT寄存器的ENDF位将置位。若FMC_CTL寄存器的ENDIE位被置1，则FMC将触发一个中断。需要注意的是，用户需确保写入的是正确的擦除地址。否则当待擦除页的地址被用来取指令或访问数据时，软件将会“跑飞”。该情况下，FMC不会提供任何出错通知。另一方面，对擦写保护的页进行擦除操作将无效。如果FMC_CTL寄存器的ERRIE位被置位，该操作将触发操作出错中断。中断服务程序可通过检测FMC_STAT寄存器的WPERR位来判断该中断是否发生。下图显示了页擦除操作流程。

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存储闪存块的内容。当设置FMC_CTL寄存器中MER为1时，擦除过程作用于整片闪存。整片擦除操作，寄存器设置具体步骤如下：

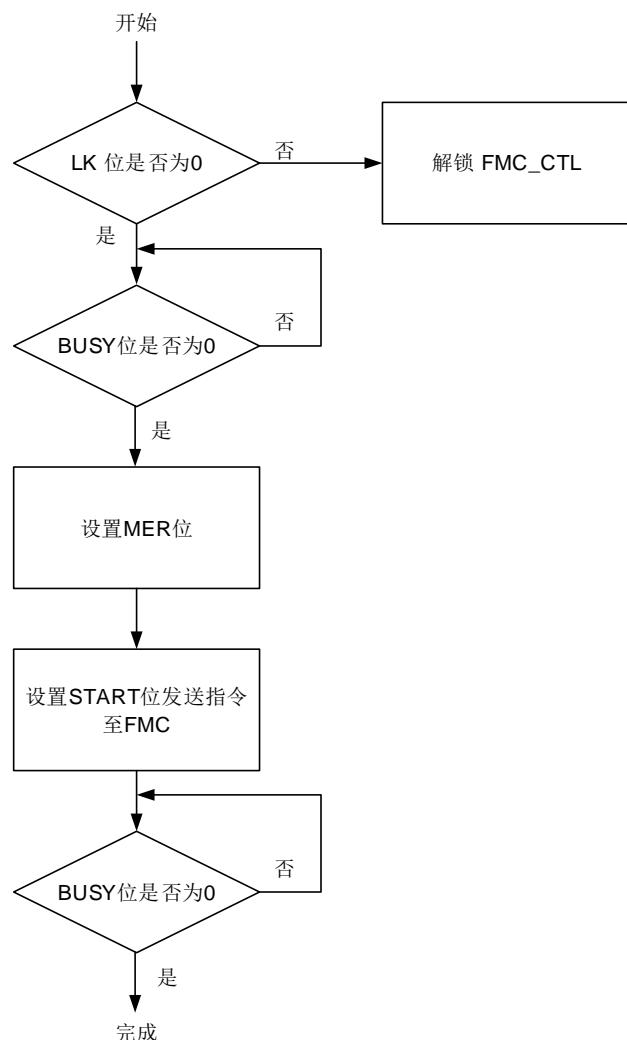
- 确保FMC_CTL寄存器不处于锁定状态；
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0；
- 如果整片擦除闪存，置位FMC_CTL寄存器的MER位；
- 通过将FMC_CTL寄存器的START位置1来发送整片擦除命令到FMC；
- 等待擦除指令执行完毕，FMC_STAT寄存器的BUSY位清0；
- 如果需要，使用DBUS读并验证是否擦除成功。

当整片擦除成功执行，FMC_STAT寄存器的ENDF位置位。若FMC_CTL寄存器的ENDIE位被置1，FMC将触发一个中断。由于所有的闪存数据都将被复位为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。此外，如果任何闪存页处于擦除/编程保护下，整片擦除操作会被忽略。在这种情况下，如果FMC_CTL寄存器

的ERRIE位被置位，该操作将触发操作出错中断。在中断服务程序中，软件可以通过检查FMC_STAT寄存器中的WPERR位来检测这种情况。

下图显示了整片擦除操作流程。

图 2-2. 整片擦除操作流程



2.3.6. 主存储闪存块编程

FMC提供了一个通过DBUS修改主存储闪存内容的32位整字/16位半字编程功能。实际上，主存储闪存编程为32位或64位，由FMC_WS寄存器中的PGW位定义。

编程操作，寄存器设置具体步骤如下：

- 确保FMC_CTL寄存器不处于锁定状态；
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0；
- 设置FMC_WS寄存器的PGW位；
- 置位FMC_CTL寄存器的PG位；
- DBUS写一个32位整字/16位半字到目的绝对地址（0x08XX XXXX）；

如果DBUS按32位字编程并且PGW位设置为0（32位编程），DBUS写1次即可将数据编程入闪存。待编程数据必须字对齐。

如果DBUS按32位字编程并且PGW位设置为1（64位编程），DBUS写2次即可将数据编程入闪存。待编程数据必须双字对齐。

如果DBUS按16位字编程并且PGW位设置为0（32位编程），DBUS写2次即可将数据编程入闪存。待编程数据必须字对齐。

如果DBUS按16位字编程并且PGW位设置为1（64位编程），DBUS写4次即可将数据编程入闪存。待编程数据必须双字对齐。

为了缩短编程时间，建议DBUS采用按32位编程。当待编程数据按双字对齐时，设置PGW位为1。当待编程数据按字对齐时，设置PGW位为0。

- 等待编程指令执行完毕，FMC_STAT寄存器的BUSY位清0；
- 如果需要，使用DBUS读并验证是否编程成功。

当主存储块编程成功执行，FMC_STAT寄存器的ENDF位置位。若FMC_CTL寄存器的ENDIE位被置1，FMC将触发一个中断。有一些编程错误需要注意：

编程操作时需要检查目的地址是否已经被擦除。如果该地址没有被擦除，即使对该地址写0x0值，FMC_STAT寄存器的PGERR位也将被置1。每个字在擦除后和下次擦除前只能编程一次。注意PG位必须在字/半字编程操作前被置位。

此外，在正被擦除/保护页上的编程操作会被忽略，FMC_STAT寄存器中的WPERR位会被置位。

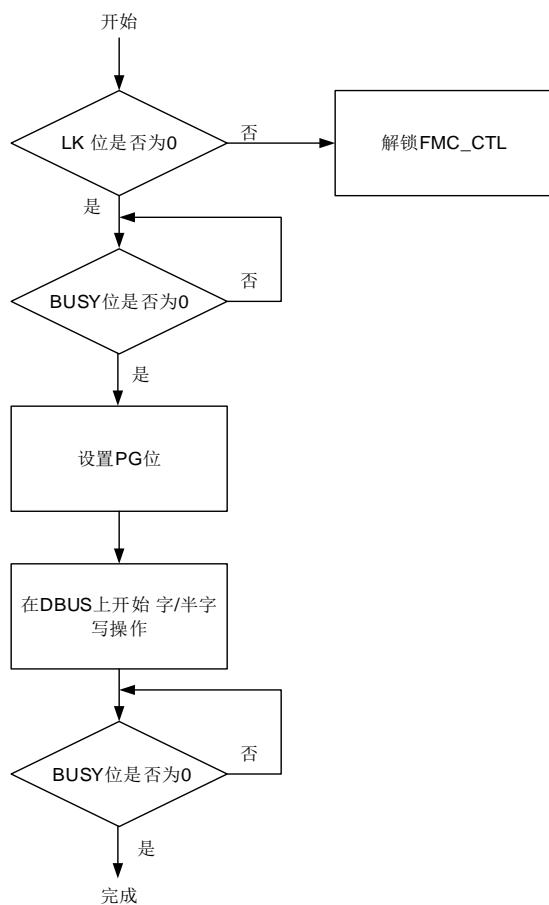
在下列情况下，FMC_STAT寄存器中的PGAERR位会被置位。

- DBUS按字节编程（非32位或16位）
- DBUS混合编程。不允许混合32位和16位编程。
- DBUS编程未对齐。如果DBUS按32位编程并且PGW设置为1（64位编程），下一次DBUS必须在对应的双字地址按双字对齐写操作。如果DBUS按16位编程并且PGW设置为0（32位编程），下一次DBUS必须在对应的字地址按字对齐写操作。如果DBUS按16位编程并且PGW设置为1（64位编程），第2nd/3rd/4th DBUS写操作必须在对应的双字地址按双字对齐。

注意：如果编程数据未能写满64位/32位，这些数据不会被编程入flash闪存，并且不会有任何提示。

在这些情况下，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。在中断服务程序中，软件可以通过检查FMC_STAT寄存器中的PGERR位，PGAERR位或者WPERR位来检测发生了哪种错误。下图显示了主存储块字编程操作流程。

图 2-3. 字编程操作流程



注意：避免在同一个bank中既进行读操作，又进行擦除/编程操作。当CPU进入省电模式时，对闪存的操作将失败。

2.3.7. OTP 编程

OTP编程方法与主储存闪存编程相同。OTP块只能被编程一次并且不能被擦除。

注意：必须确保在OTP编程操作时不会发生任何意外中断，例如系统复位或掉电。如果发生意外中断，flash闪存中的数据有很小可能性会出错。

2.3.8. 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。选项字节擦除过程如下所示。

- 确保FMC_CTL寄存器不处于锁定状态；
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0；
- 解锁FMC_CTL寄存器的选项字节操作位；
- 等待FMC_CTL寄存器的OBWEN位置1；
- 置位FMC_CTL寄存器的OBER位；

- 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;
- 等待擦除指令执行完毕, FMC_STAT寄存器的BUSY位清0;
- 如果需要, 使用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位置位。若FMC_CTL寄存器的ENDIE位被置1, FMC将触发一个中断。

2.3.9. 选项字节编程

FMC提供了擦除/编程功能, 可用来修改选项字节内容。选项字节共有8对选项字节。每对选项字节的高字节是低字节的补字节。当选项字节被修改时, FMC自动生成该选项字节的高字节。擦除操作过程如下。

- 确保FMC_CTL寄存器不处于锁定状态;
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态, 若BUSY位为1, 则需等待该操作结束, BUSY位变为0;
- 解锁FMC_CTL寄存器的选项字节操作位;
- 等待FMC_CTL寄存器的OBWEN位置1;
- 置位FMC_CTL寄存器的OBPG位;
- DBUS写一个32位整字/16位半字到目的地址。根据不同的DBUS写数据大小和FMC_WS寄存器中的PGW位数值, 可能需要写1次, 2次或4次。写操作方法与主存储闪存编程操作类似;
- 等待编程指令执行完毕, FMC_STAT寄存器的BUSY位清0;
- 如果需要, 使用DBUS读并验证是否编程成功。

当选项字节编程成功执行, FMC_STAT寄存器的ENDF位置位。若FMC_CTL寄存器的ENDIE位被置1, FMC将触发一个中断。注意可能发生编程错误, 类似主存储闪存编程操作, PGERR位和PGAERR位可能被置位。

选项字节修改后只有在系统复位后才生效。

2.3.10. 选项字节说明

每次系统复位后, 选项字节被重加载到FMC_OBSTAT和FMC_WP寄存器后, 选项字节生效。选项字节的补字节具体为选项字节取反。当选项字节被重装载时, 如果选项字节的补字节和选项字节不匹配, FMC_OBSTAT寄存器的OBERR位将被置1, 选项字节被强制设置为0xFF。若选项字节和其补字节同为0xFF, 则OBERR位不置位。选项字节详情见下表。

表 2-3. 选项字节

地址	名称	说明
0x1fff f800	SPC	选项字节安全保护值 0xA5: 未保护状态 除0xA5外的任何值: 已保护状态
0x1fff f801	SPC_N	SPC补字节
0x1fff f802	USER	[7:3]: 保留 [2]: nRST_STDBY

地址	名称	说明
		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_HW
		0: 硬件使能独立看门狗功能 1: 软件使能独立看门狗功能
0x1fff f803	USER_N	USER补字节值
0x1fff f804	DATA[7:0]	用户定义数据7到0位
0x1fff f805	DATA_N[7:0]	DATA补字节值的7到0位
0x1fff f806	DATA[15:8]	用户定义数据15到8位
0x1fff f807	DATA_N[15:8]	DATA补字节值的15到8位
0x1fff f808	WP[7:0]	页擦除/编程保护值的7到0位 0: 保护生效 1: 未保护
0x1fff f809	WP_N[7:0]	WP补字节值的7到0位
0x1fff f80a	WP[15:8]	页擦除/编程的保护值的15到8位
0x1fff f80b	WP_N[15:8]	WP补字节值的15到8位
0x1fff f80c	WP[23:16]	页擦除/编程的保护值的23到16位
0x1fff f80d	WP_N[23:16]	WP补字节值的23到16位
0x1fff f80e	WP[31:24]	页擦除/编程的保护值的31到24位 WP[30:24]: 每个bit可设置4KB闪存的保护状态。第0位设置前4KB闪存的保护状态，以此类推。这31位总计可设置前124KB的闪存保护状态。 WP[31]: 第31位可设置闪存剩下部分的保护状态。
0x1fff f80f	WP_N[31:24]	WP补字节值的31到24位

2.3.11. 页擦除/编程保护

FMC的页擦除/编程保护功能可以阻止对闪存的意外操作。当FMC对被保护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FMC_CTL寄存器的ERRIE位也被置1来使能相应的中断，FMC将触发闪存操作出错中断，等待CPU处理。配置选项字节的WP[31:0]某位为0可以单独使能某几页的保护功能。如果在选项字节块执行了擦除操作，所有的闪存页擦除和编程保护功能都将失效。当选项字节的WP被改变时，需要系统复位使之生效。

2.3.12. 安全保护

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

未保护状态：当将SPC字节和它的补字节被设置为0x5AA5，系统复位以后，闪存将处于非安

全保护状态。主存储块和选项字节可以被所有操作模式访问。

已保护状态：当设置SPC字节和它的补字节值为任何除0x5AA5外的值，系统复位以后，安全保护状态生效。需要注意的是，若该修改过程中，MCU的调试模块依然和外部JTAG/SWD设备相连，需要用上电复位代替系统复位以使得修改后的保护状态生效。在安全保护状态下，主存储闪存块仅能被用户代码访问且前4KB的闪存自动处于页擦除/编程保护状态下。在调试模式下，或从SRAM中启动时，以及从boot loader区启动时，这些模式下对主存储块的操作都被禁止。如果在这些模式下读主存储块，将产生总线错误。如果在这些模式下，对主存储块进行编程或擦除操作，**FMC_STAT**寄存器的WPERR位将被置1。但这些模式下都可以对选项字节进行操作，从而可以通过该方式失能安全保护功能。如果将SPC字节和它的补字节设置为0x5AA5，安全保护功能将失效，并自动触发一次整片擦除操作。

2.4. FMC 寄存器

FMC基地址: 0x4002 2000

2.4.1. 等待状态寄存器 (FMC_WS)

地址偏移: 0x00

复位值: 0x0000 0630

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
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PGW	Reserved	DCRST	ICRST	DCEN	ICEN		Reserved			PFEN	Reserved		WSCNT[2:0]		
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	PGW	编程宽度 0: 32位编程 1: 64位编程
14:13	保留	必须保持复位值
12	DCRST	复位DBUS高速缓存区。该位仅可在DCEN位置0时可写。 0: 无效果 1: 复位DBUS高速缓存区
11	ICRST	复位IBUS高速缓存区。该位仅可在ICEN位置0时可写。 0: 无效果 1: 复位IBUS高速缓存区
10	DCEN	DBUS高速缓存区使能位 0: 失能DBUS高速缓存区 1: 使能DBUS高速缓存区
9	ICEN	IBUS高速缓存区使能位 0: 失能IBUS高速缓存区 1: 使能IBUS高速缓存区
8:5	保留	必须保持复位值。
4	PFEN	预取功能使能位 0: 失能预取功能

1: 使能预取功能

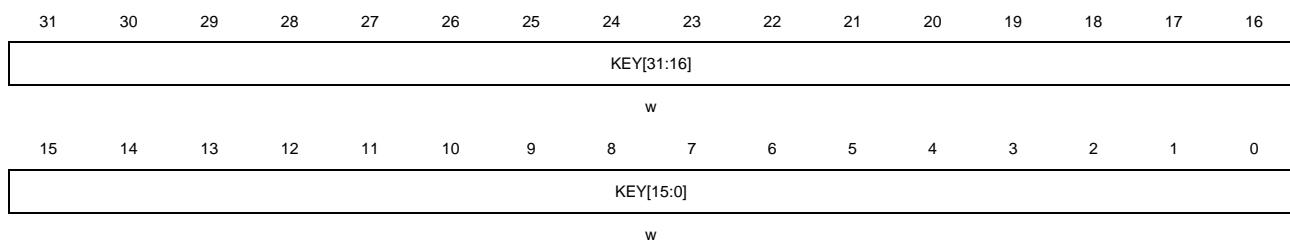
3	保留	必须保持复位值
2:0	WSCNT[2:0]	等待状态计数寄存器 软件置1和清0。 000: 不增加等待状态 001: 增加1个等待状态 010: 增加2个等待状态 011: 增加3个等待状态 100 ~ 111: 保留

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



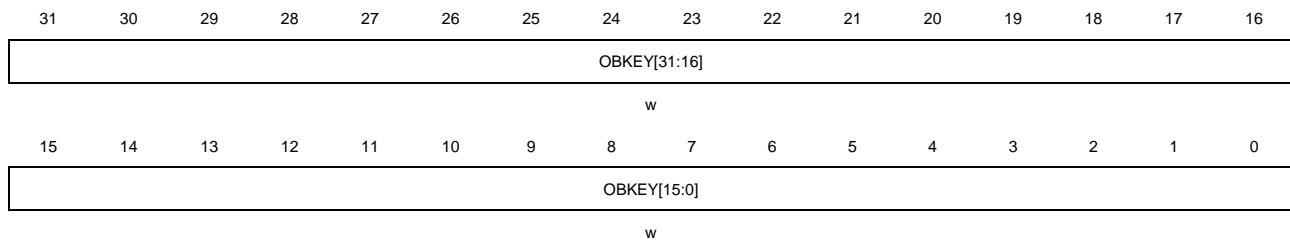
位/位域	名称	描述
31:0	KEY[31:0]	FMC_CTL解锁寄存器 这些位仅能被软件写。 写解锁值到KEY[31:0]可以解锁 FMC_CTL寄存器。

2.4.3. 选项字节操作解锁寄存器 (**FMC_OBKEY**)

地址偏移: 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

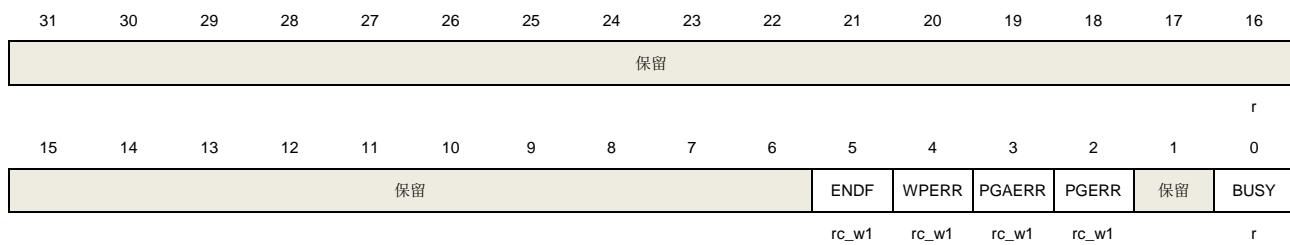
31:0	OBKEY [31:0]	FMC_CTL选项字节操作解锁寄存器 这些位仅能被软件写 写解锁值到OBKEY[31:0]解锁FMC_CTL寄存器的选项字节命令。
------	--------------	---

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



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。软件写1清0。
4	WPERR	擦除/编程保护错误标志位 在受保护的页上擦除/编程操作时，此位被硬件置1。软件写1清0。
3	PGAERR	编程对齐错误标志位 当DBUS写数据不对齐时，此位被硬件置1。软件写1清0。
2	PGERR	编程错误标志位 当被编程区域状态不为0xFFFF时，对闪存编程，此位被硬件置1。软件写1清0。
1	保留	必须保持复位值
0	BUSY	闪存忙标志 当闪存操作正在进行时，此位被置1。当操作结束或者出错，此位被清0。

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



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

rw rw rw rs rs 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	ENDIE	操作结束中断使能位 软件置1和清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	出错中断使能位 软件置1和清0 0: 无硬件中断产生. 1: 使能出错中断
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	FMC_CTL0寄存器锁定标志位 当正确的序列写入FMC_KEY0寄存器，此位由硬件清0。此位可以由软件置1。
6	START	向FMC发送擦除命令位 软件置1可以发送擦除命令到FMC。当BUSY位被清0时，此位由硬件清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

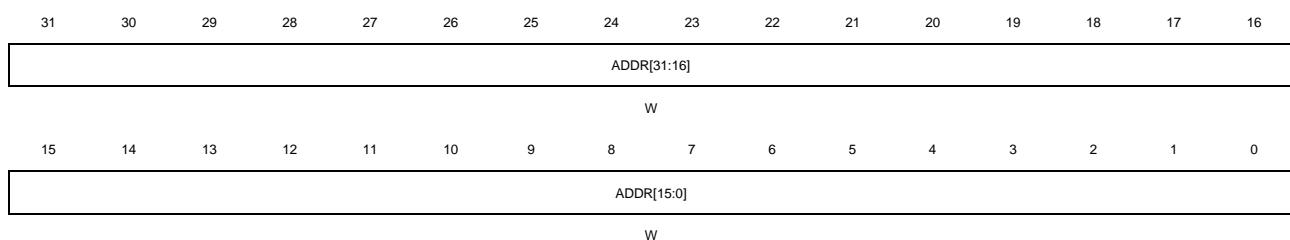
注意：当相应闪存操作完成后，该寄存器需处于复位状态。

2.4.6. 地址寄存器 (FMC_ADDR)

地址偏移: 0x14

复位值: 0x0000 0000

该寄存器只能按字(32位)访问



位/位域	名称	描述
31:0	ADDR[31:0]	闪存擦除或编程地址 该位通过软件设置。 ADDR位是闪存擦除/编程命令的地址

2.4.7. 选项字节状态寄存器 (FMC_OBSTAT)

地址偏移: 0x1C

复位值: 0x0XXX XXXX

该寄存器只能按字(32位)访问



位/位域	名称	描述
------	----	----

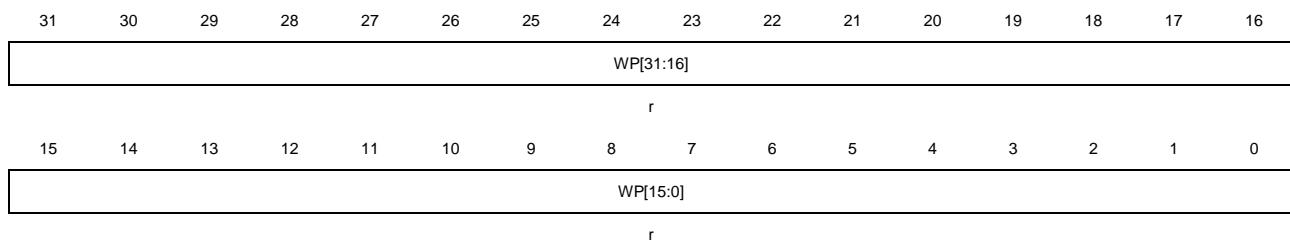
31:26	保留	必须保持复位值
25:10	DATA[15:0]	系统复位后保存选项字节的DATA[15:0]部分
9:2	USER[7:0]	系统复位后保存选项字节块的USER字节
1	SPC	安全保护状态 0: 未保护 1: 已保护
0	OBERR	选项字节读错误位 当选项字节和它的补字节不匹配时此位由硬件置1, 选项字节被强制设置为0xFF。

2.4.8. 擦除/编程保护寄存器 (FMC_WP)

地址偏移: 0x20

复位值: 0xXXXX XXXX

该寄存器只能按字(32位)访问



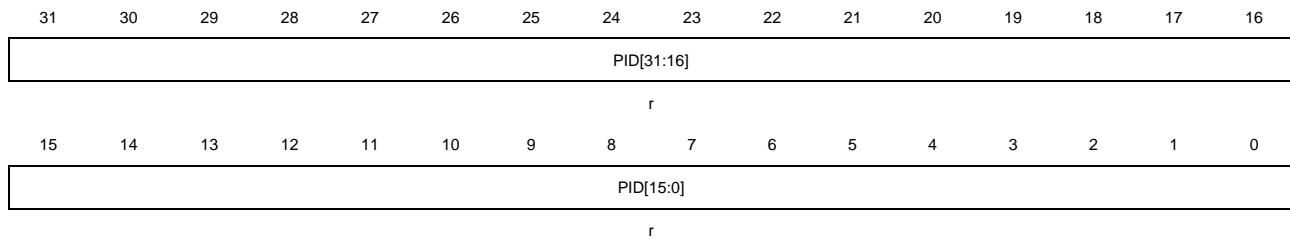
位/位域	名称	描述
31:0	WP[31:0]	系统复位后保存选项字节块的WP[31:0]部分

2.4.9. 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



位/位域	名称	描述
31:0	PID[31:0]	产品保留 ID 寄存器 该寄存器为只读

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。

3. 电源管理单元（PMU）

3.1. 简介

功耗设计是GD32C10x系列产品最注重的问题之一。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，GD32C10x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。

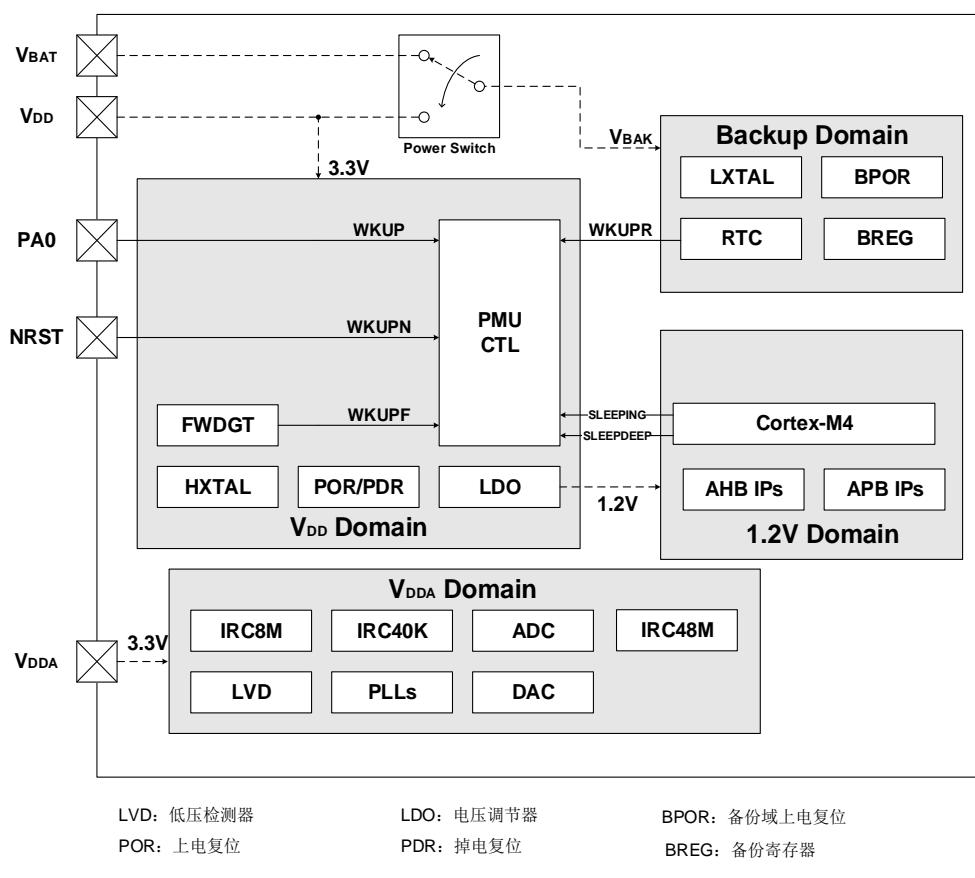
3.2. 主要特征

- 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（LVD），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，由电池（ V_{BAT} ）为备份域供电；
- LDO输出电压用于节约能耗；

3.3. 功能说明

[图3-1. 电源域概览](#)提供了PMU及相关电源域的内部结构框图。

图3-1. 电源域概览



3.3.1. 电池备份域

电池备份域由内部电源切换器来选择VDD供电或VBAT(电池)供电，然后由VBAK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)、BREG(备份寄存器)，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VDD关闭时，VBAT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VDD/VDDA域掉电复位电路控制的。对于没有外部电池的应用，建议将VBAT引脚通过100nF的外部陶瓷去耦电容连接到VDD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VBAK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)时钟的128分频。当VDD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寄存器设置预期的唤醒时间并启用唤醒功能，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[实时时钟 \(RTC\)](#)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. V_{DD} / V_{DDA} 电源域

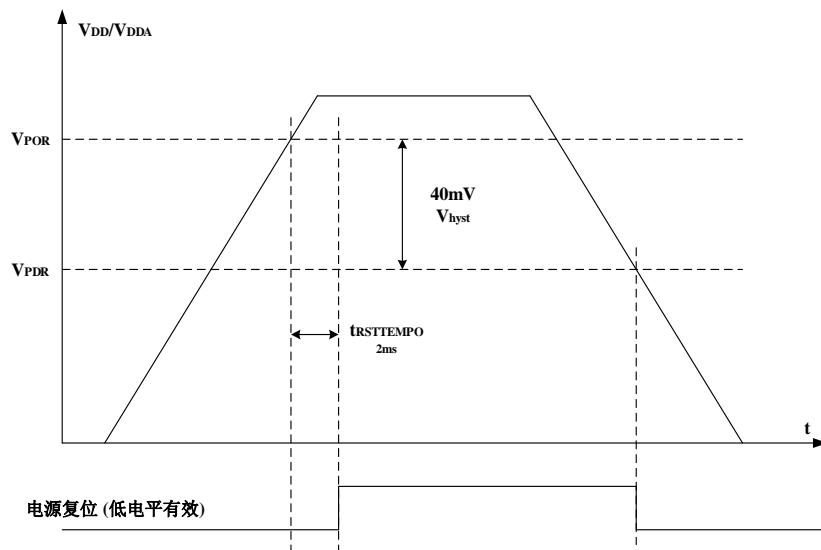
V_{DD}/V_{DDA} 域包括 V_{DD} 域和 V_{DDA} 域两部分。V_{DD} 域包括 HXTAL（高速外部晶体振荡器）、LDO（电压调节器）、POR / PDR（上电/掉电复位）、FWDGT（独立看门狗定时器）和除 PC13、PC14 和 PC15 之外的所有 PAD 等等。V_{DDA} 域包括 ADC / DAC（AD / DA 转换器）、IRC8M（内部 8M RC 振荡器）、IRC48M（内部 48M RC 振荡器）、IRC40K（内部 40KHz RC 振荡器）、PLLs（锁相环）和 LVD（低电压检测器）等等。

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 1.66V，V_{PDR} 表示掉电复位的阈值电压，典型值约为 1.62V。迟滞电压 V_{hyst} 值约为 40mV。

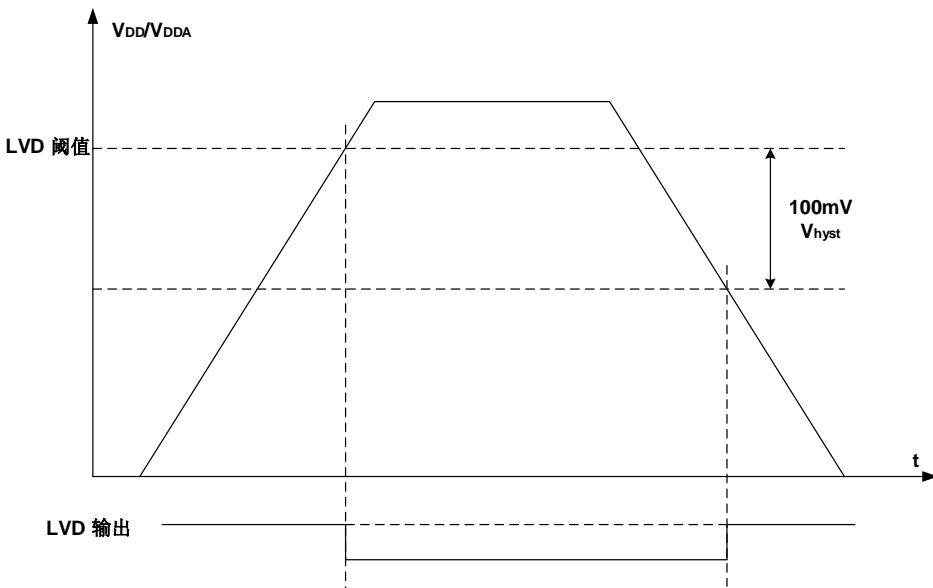
图3-2. 上电 / 掉电复位波形图



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，位于电源控制和状态寄存器（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 0.3V。

为提高 ADC 和 DAC 的精度，可将独立的外部参考电压连接至 ADC / DAC 引脚 V_{REF+} / V_{REF-} 。根据不同的封装， V_{REF+} 可被连接至 V_{DDA} 引脚，或者外部参考电压，外部参考电压的范围请参考 [表 12-2. ADC 引脚定义](#) 和 [表 13-1. DAC 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 的封装上，而在 64-pin 或更少引脚的封装上不存在，因其内部已经连接至 V_{DDA} 。 V_{REF-} 内部直接连接至 V_{SSA} 。

3.3.3. 1.2V 电源域

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

3.3.4. 省电模式

系统复位或电源复位后，GD32C10x 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、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。

注意：为了顺利进入深度睡眠模式，所有 EXTI 线上的挂起状态（在 EXTI_PD 寄存器中）和相关外设标志位必须被复位，参考[表 7-3. EXTI 触发源](#)。否则，程序将直接跳过深度睡眠模式进入过程而继续执行下面的程序。

待机模式

待机模式也是基于 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、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 是否已进入待机模式。待机模式有四个唤醒源，包括来自 NRST 引脚的外部复位，RTC 闹钟，FWDGT 复位，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 0x0000 0000 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、IRC48M、HXTAL 和 PLL	1. 关闭 1.2V 电源域的供电 2. 关闭 IRC8M、IRC48M、HXTAL 和 PLL
LDO 状态	开启（正常功耗模式）	开启（正常功耗模式或者低功耗模式）	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 时的中断）均可唤醒	若通过 WFI 进入，来自 EXTI 的任何中断可唤醒； 若通过 WFE 进入，来自 EXTI 的任何事件（或 SEVONPEND=1 时的中断）可唤醒	1. NRST 引脚 2. WKUP 引脚 3. FWDGT 复位 4. RTC 闹钟
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

注意：在待机模式下，除了 RESET 引脚，配置为 RTC 功能的 PC13，用作 LXTAL 晶振引脚的 PC14 和 PC15，使能的 WKUP 引脚，其他所有 I/O 都处于高阻态。

3.4. PMU 寄存器

PMU基地址: 0x4000 7000

3.4.1. 控制寄存器 (PMU_CTL)

地址偏移: 0x00

复位值: 0x0000 4000 (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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LDOVS[1:0]		保留		BKPWEN		LVDT[2:0]		LVDEN	STBRST	WURST	STBMOD	LDOLP			

rs rw rw rc_w1 rc_w1 rw rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:14	LDOVS[1:0]	选择LDO输出 在主PLL关闭时，这些位由软件配置。在主PLL使能后，LDOVS设置的值生效。如果主PLL关闭，LDO输出低电压模式被选中。 00: 保留 (LDO输出正常电压模式) 01: LDO输出正常电压模式 10: 保留 (LDO输出低电压模式) 11: LDO输出低电压模式
13:9	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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								WUPEN	保留				LVDF	STBF	WUF
rw r r															

位/位域	名称	描述
31:9	保留	必须保持复位值
8	WUPEN	WKUP引脚唤醒使能 0: 关闭WKUP引脚唤醒功能 1: 开启WKUP引脚唤醒功能 如果WUPEN在进入省电模式之前置1, WKUP引脚的上升沿会将系统从省电模式唤醒。由于WKUP引脚为高电平有效, WKUP引脚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	低电压检测器状态标志 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闹钟事件。 该位只能由POR / PDR或通过设置PMU_CTL寄存器的WURST位来清零。

4. 备份寄存器 (BKP)

4.1. 简介

位于备份域中的备份寄存器可在 V_{DD} 电源关闭时由 V_{BAT} 供电，备份寄存器有 42 个 16 位（84 字节）寄存器可用来存储并保护用户应用数据，从待机模式唤醒或系统复位也不会对这些寄存器造成影响。

此外，BKP 寄存器也可实现侵入检测和 RTC 校准功能。

在复位之后，任何对备份域寄存器的写操作都将被禁止，也就是说，备份寄存器和 RTC 不允许写访问。为使能对备份寄存器和 RTC 的写访问，首先通过设置 RCU_APB1EN 寄存器的 PMUEN 和 BKPIEN 位来打开电源和备份接口时钟，然后再通过设置 PMU_CTL 寄存器的 BKPWEN 位来使能对备份域中寄存器的写访问。

4.2. 主要特征特性

- 84字节备份寄存器用来在省电模式下保护数据。如果侵入事件发生，备份寄存器会被复位
- 侵入源 (PC13) 的有效电平可配置
- RTC时钟校准寄存器可提供RTC闹钟或秒输出选择，及设置校准值的功能
- 侵入控制状态寄存器 (BKP_TPCS) 可实现对侵入检测的中断或事件的控制

4.3. 功能说明

4.3.1. RTC 时钟校准

为提高RTC时钟精度，MCU提供时钟输出校准功能。RTC时钟或者RTC时钟经64分频后作为输出至PC13。通过设置BKP_OCTL寄存器中的COEN位来使能此功能。

校准值通过BKP_OCTL寄存器中的RCCV[6:0]设置，校准功能可实现以 $1000000/2^{20}\text{ppm}$ 的比例减慢RTC时钟。

4.3.2. 侵入检测

MCU提供侵入检测功能以保护重要的用户数据，可通过设置BKP_TPCTL寄存器中的TPEN位来使能TAMPER引脚对应的功能。为防止侵入事件的丢失，边沿检测信号与TPEN位的逻辑与作为侵入检测信号的输入，因此在TAMPER引脚使能之前，侵入检测应该被配置。当侵入事件被检测到，对应的BKP_TPCS寄存器中的TEF位被置位。如果侵入中断被使能，侵入事件可以产生一个中断。任何侵入事件将会复位所有备份数据寄存器。

注意：当TPAL=0/1，如果TAMPER引脚在使能（通过设置TPEN位）之前已经为高/低，尽管TAMPER引脚上没有上升/下降沿信号，一个额外的侵入事件将会发生。

4.4. BKP 寄存器

BKP基地址: 0x4000 6C00

4.4.1. 备份数据寄存器 (BKP_DATAx) (x= 0..41)

地址偏移: 0x04到0x28, 0x40到0xBC

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
DATA [15:0]															
rw															

位/位域	名称	描述
15:0	DATA[15:0]	备份数据 这些位用来存储一般用户数据。即使从待机模式唤醒或系统复位或电源复位后，BKP_DATAx寄存器的内容仍旧不会丢失。

4.4.2. RTC 信号输出控制寄存器 (BKP_OCTL)

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
CALDIR	CCOSEL	保留	ROSEL	A SOEN	COEN	RCCV[6:0]									
rw	rw		rw	rw	rw	rw									

位/位域	名称	描述
15	CALDIR	RTC时钟校准方向 0: 变慢 1: 变快 该位只能被备份域复位清除。
14	CCOSEL	RTC时钟输出选择 0: RTC时钟64分频 1: RTC时钟 该位只能被上电复位 (POR) 清除。
13:10	保留	必须保持复位值
9	ROSEL	RTC输出选择 0: RTC输出为闹钟脉冲

		1: RTC输出为秒脉冲 该位只能被备份域复位清除。
8	ASOEN	RTC闹钟或秒信号输出使能 0: RTC闹钟或秒信号输出禁止 1: RTC闹钟或秒信号输出使能 使能后, TAMPER引脚可作为RTC输出。 该位只能被备份域复位清除。
7	COEN	RTC时钟校准输出使能 0: RTC时钟校准输出禁止 1: RTC时钟校准输出使能 使能后, TAMPER引脚输出RTC时钟或RTC时钟的64分频。ASOEN位优先于COEN位, 当ASOEN位置位时, 不管COEN置位与否, TAMPER引脚作为RTC闹钟或秒信号输出。 该位只能被上电复位(POR)清除。
6:0	RCCV[6:0]	RTC时钟校准值 该值表示在每 2^{20} 个时钟脉冲内将有多少个时钟脉冲被忽略。 该位只能被备份域复位清除。

4.4.3. 侵入引脚控制寄存器 (**BKP_TPCTL**)

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														TPAL	TPEN
rw rw															

位/位域	名称	描述
15:2	保留	必须保持复位值
1	TPAL	TAMPER引脚有效电平 0: TAMPER引脚高电平有效 1: TAMPER引脚低电平有效
0	TPEN	TAMPER引脚使能 0: TAMPER引脚作为GPIO口使用 1: TAMPER引脚可实现备份复位功能。TAMPER引脚上的有效电平将复位BKP_DATAx寄存器中所有数据。

4.4.4. 侵入控制状态寄存器 (**BKP_TPCS**)

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
保留				TIF	TEF		保留			TPIE	TIR	TER			
				r	r					rw	w	w			

位/位域	名称	描述
15:10	保留	必须保持复位值
9	TIF	侵入中断标志 0: 没有侵入中断发生 1: 有侵入中断发生 该位可通过TIR位置1或TPIE位置0来清零。
8	TEF	侵入事件标志 0: 没有侵入事件发生 1: 有侵入事件发生 该位可通过对TER为写1来清零。
7:3	保留	必须保持复位值
2	TPIE	侵入中断使能 0: 侵入中断禁用 1: 侵入中断使能 该位仅可通过系统复位或待机模式唤醒后复位。
1	TIR	侵入中断复位 0: 不影响 1: 复位TIF位 该位一直读为0。
0	TER	侵入事件复位 0: 不影响 1: 复位TEF位 该位一直读为0。

5. 复位和时钟单元 (RCU)

5.1. 复位控制单元(RCTL)

5.1.1. 简介

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

5.1.2. 功能说明

电源复位

当发生以下任一事件时，产生电源复位：上电/掉电复位（POR/PDR 复位），从待机模式中返回后由内部复位发生器产生。电源复位复位所有的寄存器除了备份域。电源复位为低电平有效，当内部LDO电源基准准备好提供1.2V电压时，电源复位电平将变为无效。复位入口向量被固定在存储器映射的地址0x0000 0004。

系统复位

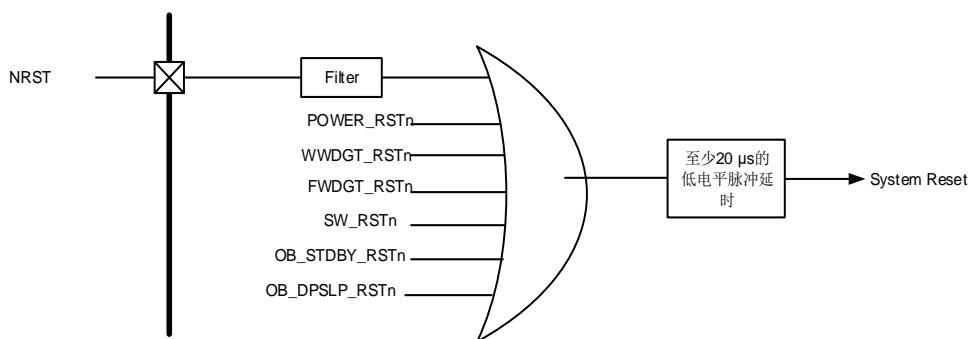
当发生以下任一事件时，产生一个系统复位：

- 上电复位(POWER_RSTn)
- 外部引脚复位(NRST)
- 窗口看门狗计数终止(WWDGT_RSTn)
- 独立看门狗计数终止(FWDGT_RSTn)
- Cortex™-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将产生复位(OB_STDBY_RSTn)
- 用户选择字节寄存器nRST_DPSLP设置为0，并且进入深度睡眠模式时(OB_DPSLP_RSTn)

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

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

图5-1. 系统复位电路



备份域复位

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

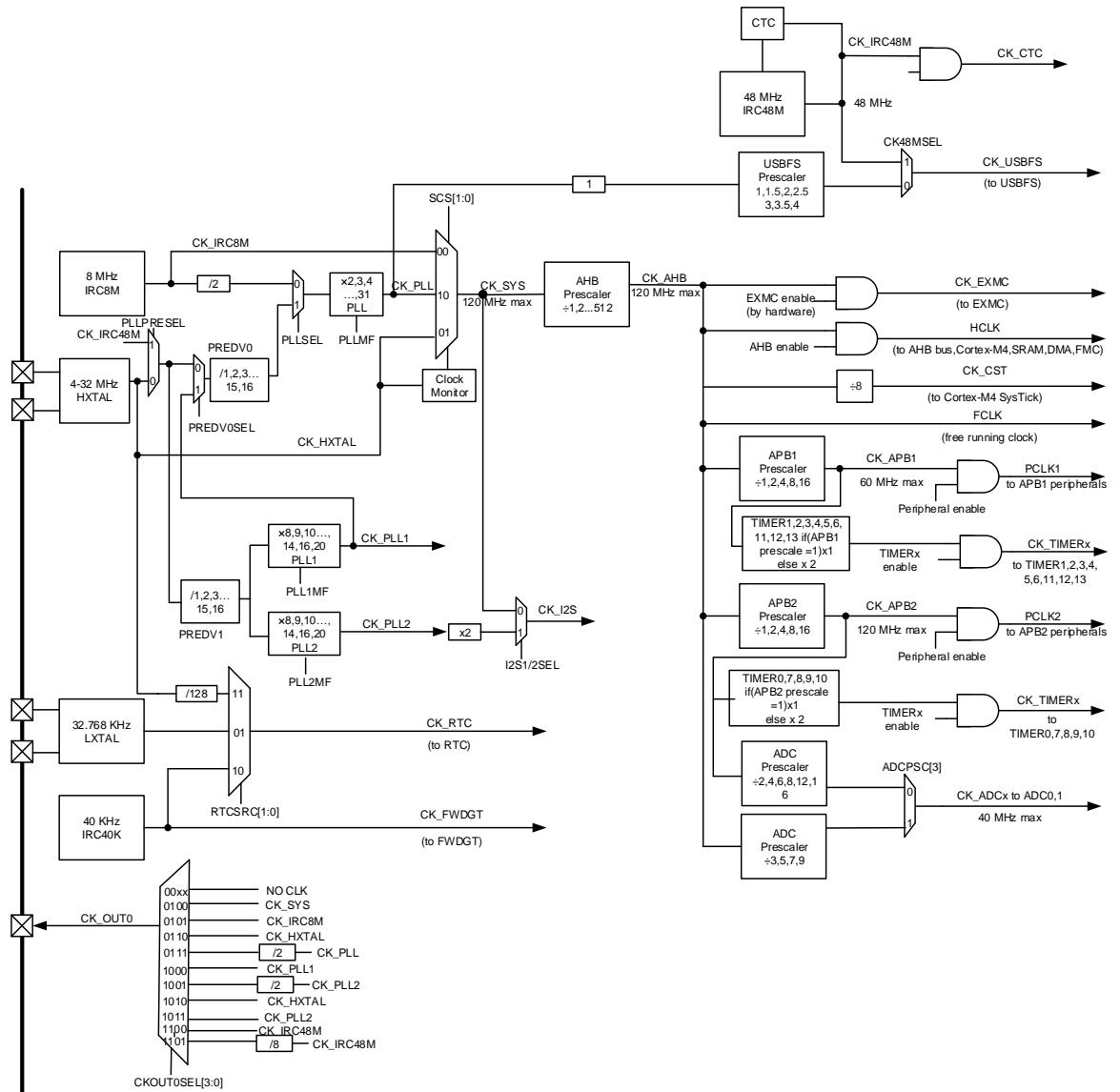
5.2. 时钟控制单元(CCTL)

5.2.1. 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)、一个外部高速晶体振荡器时钟(HXTAL)、一个内部40K RC振荡器时钟(IRC40K)、一个外部低速晶体振荡器时钟(LXTAL)、三个锁相环(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门控电路。

AHB、APB和Cortex™-M4时钟都源自系统时钟(CK_SYS)，系统时钟的时钟源可以选择IRC8M、HXTAL或PLL。系统时钟的最大运行时钟频率可以达到120MHz。独立看门狗定时器有独立的时钟源(IRC40K)，实时时钟(RTC)使用IRC40K、LXTAL或HXTAL的128分频作为时钟源。

图5-2. 时钟树



预分频器可以配置AHB、APB2和APB1域的时钟频率。AHB、APB2、APB1域的最高时钟频率分别为120MHz、120MHz、60MHz。RCU通过AHB时钟(HCLK)8分频后作为Cortex系统定时器(SysTick)的外部时钟。通过对SysTick控制和状态寄存器的设置，可选择上述时钟或AHB(HCLK)时钟作为SysTick时钟。

ADC时钟由APB2时钟经2、4、6、8、12、16分频或由AHB时钟经3、5、7、9分频获得，它们是通过设置RCU_CFG0和RCU_CFG1寄存器的ADCPSC位来选择。

TIMER时钟由CK_APB1和CK_APB2时钟分频获得，如果APBx(x=1,2)的分频系数不为1，则TIMER时钟为CK_APBx(x=1,2)的两倍，如果APBx(x=1,2)的分频系数为1，则TIMER时钟等于CK_APBx(x=1,2)。

USBFS的时钟由CK48M时钟提供。通过配置RCU_ADDCTL寄存器的CK48MSEL位可以选择CK_PLL时钟或IRC48M时钟做为CK48M的时钟源。

CTC时钟由IRC48M时钟提供，通过CTC单元，可以实现IRC48M时钟精度的自动调整。

I2S的时钟由CK_SYS或PLL2*2提供，通过配置RCU_CFG1寄存器的I2SxSEL来选择。

通过配置RCU_BDCTL寄存器的RTCSRC位，RTC时钟可以选择由LXTAL时钟、IRC40K时钟或HXTAL时钟的128分频提供。RTC时钟选择HXTAL时钟的128分频做为时钟源后，当1.2V内核电压域掉电时，时钟将停止。RTC时钟选择IRC40K时钟做为时钟源后，当V_{DD}掉电时，时钟将停止。RTC时钟选择LXTAL时钟做为时钟源后，当V_{DD}和V_{BAT}都掉电时，时钟将停止。

当FWDGT启动时，FWDGT时钟被强制选择由IRC40K时钟做为时钟源。

5.2.2. 主要特征

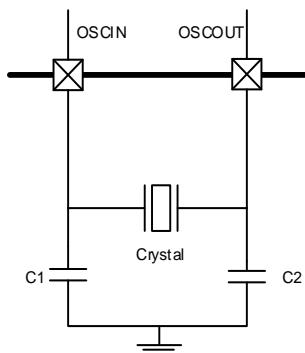
- 4到32MHz外部高速晶体振荡器(HXTAL);
- 内部8MHz RC振荡器(IRC8M);
- 内部48MHz RC振荡器;
- 32,768 Hz外部低速晶体振荡器 (LXTAL);
- 内部40KHz RC振荡器(IRC40K);
- PLL时钟源可选HXTAL、IRC8M或IRC48M;
- HXTAL时钟监视器。

5.2.3. 功能说明

外部高速晶体振荡时钟(HXTAL)

4到32M的外部高速晶体振荡器可为系统时钟提供更为精确时钟源。带有特定频率的晶体必须靠近两个HXTAL的引脚连接。和晶体连接的外部电阻和电容必须根据所选择的振荡器来调整。

图5-3. HXTAL时钟源



HXTAL晶体振荡器可以通过设置控制寄存器RCU_CTL的HXTALEN位来启动或关闭，在控制寄存器RCU_CTL中的HXTALSTB位用来指示外部高速振荡器是否已稳定。在启动时，直到这一位被硬件置‘1’，时钟才被释放出来。这个特定的延迟时间被称为振荡器的启动时间。当HXTAL时钟稳定后，如果在中断寄存器RCU_INT中的相应中断使能位HXTALSTBIE位被置‘1’，将会产生相应中断。此时，HXTAL时钟可以被直接用作系统时钟源或者PLL输入时钟。

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

内部 8M RC 振荡器时钟(IRC8M)

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默认选择其做为系统时钟源。IRC8M RC振荡器能够在不需要任何外部器件的条件下为用户提供更低成本类型的时钟源。IRC8M RC振荡器可以通过设置控制寄存器(RCU_CTL)中的IRC8MEN位被启动和关闭。控制寄存器RCU_CTL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晶体振荡器要更短。如果中断寄存器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时钟也可用作系统时钟源或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。用户可以根据需求、环境条件和成本决定选择哪个时钟作为系统时钟源。

如果HXTAL或者PLL被选择为系统时钟源，为了最大程度减小系统从深度睡眠模式恢复的时间，当系统从深度睡眠模式初始唤醒时，硬件会强制IRC8M时钟作为系统时钟。

内部 48M RC 振荡器时钟(IRC48M)

内部48MHz RC振荡器时钟，简称IRC48M时钟，拥有48MHz的固定频率，当使用USBFS模块时，IRC48M振荡器在不需要任何外部器件的条件下为用户提供了一种成本更低的时钟源选择。IRC48M RC振荡器可以通过设置RCU_ADDCTL寄存器中的IRC48MEN位被启动和关闭。RCU_ADDCTL寄存器中的IRC48MSTB位用来指示内部48MHz RC振荡器是否稳定。如果RCU_ADDINT寄存器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。IRC48M时钟可做为USBFS的系统时钟。

工厂会校准IRC48M时钟频率的精度，但是它的精度仍然不够精准。因为USB模块需要的时钟频率必须满足48MHz(500ppm)。CTC单元提供了一种硬件自动执行动态调整的功能将IRC48M时钟调整到需要的频率。

Phase locked loop (PLL)

内部有三个锁相环，PLL，PLL1，PLL2。

PLL可以通过设置RCU_CTL寄存器中的PLLEN位被启动和关闭。RCU_CTL寄存器中的PLLSTB位用来指示PLL时钟是否稳定。如果RCU_INT寄存器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。

PLL1可以通过设置RCU_CTL寄存器中的PLL1EN位被启动和关闭。RCU_CTL寄存器中的PLL1STB位用来指示PLL1时钟是否稳定。如果RCU_INT寄存器中的相应中断使能位PLL1STBIE被置‘1’，在PLL1稳定以后，将产生一个中断。

PLL2可以通过设置RCU_CTL寄存器中的PLL2EN位被启动和关闭。RCU_CTL寄存器中的PLL2STB位用来指示PLL2时钟是否稳定。如果RCU_INT寄存器中的相应中断使能位PLL2STBIE被置‘1’，在PLL2稳定以后，将产生一个中断。

当进入深度睡眠模式或待机模式或者HXTAL监视器检测到时钟阻塞时（HXTAL做为锁相环的输入时钟），三个PLL将被关闭。

外部低速晶体振荡器时钟(LXTAL)

LXTAL是一个频率为32.768kHz的外部低速晶体或陶瓷谐振器。它为实时时钟电路提供一个低功耗且高精准的时钟源。LXTAL振荡器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。

内部 40K RC 振荡器时钟(IRC40K)

IRC40K内部RC振荡器时钟担当一个低功耗时钟源的角色，不需要外部器件，它的时钟频率大约40kHz，为独立看门狗和实时时钟电路提供时钟。IRC40K RC振荡器可以通过设置复位源/时钟寄存器RCU_RSTSCK中的IRC40KEN位被启动和关闭。复位源/时钟寄存器RCU_RSTSCK中的IRC40KSTB位用来指示IRC40K时钟是否已稳定。如果复位源/时钟寄存器RCU_RSTSCK中的相应中断使能位IRC40KSTBIE被置‘1’，在IRC40K稳定以后，将产生一个中断。

TIMER4_CH3可以捕获IRC40K的时钟，进而对RTC和FWDGT的计数器进行校准，详细的信息可以参考AFIO_PCF0寄存器的位TIMER4CH3_IREMAP。

系统时钟(CK_SYS)选择

系统复位后，IRC8M时钟默认做为CK_SYS的时钟源，改变配置寄存器0(RCU_CFG0)中的系统时钟变换位SCS可以切换系统时钟源为HXTAL或CK_PLL。当SCS的值被改变，系统时钟将使用原来的时钟源继续运行直到转换的目标时钟源稳定。当一个时钟源被直接或通过PLL间接作为系统时钟时，它将不能被停止。

HXTAL 时钟监视器(CKM)

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

时钟输出能力

时钟输出功能输出从0.09375MHz到120MHz的时钟。通过设置时钟配置寄存器0(RCU_CFG0)中的CK_OUT0时钟源选择位域CKOUT0SEL能够选择不同的时钟信号。相应的GPIO引脚应该被配置成备用功能I/O(AFIO)模式来输出选择的时钟信号。

表5-1. 时钟输出0的时钟源选择

时钟输出0的时钟源选择位域	时钟源
00xx	NO CLK

时钟输出 0 的时钟源选择位域	时钟源
0100	CK_SYS
0101	CK_IRC8M
0110	CK_HXTAL
0111	CK_PLL/2
1000	CK_PLL1
1001	CK_PLL2/2
1010	CK_HXTAL
1011	CK_PLL2

电压控制

深度睡眠模式电压寄存器(RCU_DSV)中的DSLPVS[1:0]位域可以控制1.2V域在深度睡眠模式下的电压。

表5-2. 深度睡眠模式下1.2V域电压选择

DSLPVS[1:0]	深度睡眠模式电压(V)
00	1.0
01	0.9
10	0.8
11	1.2

5.3. RCU 寄存器

RCU基地址: 0x4002 1000

5.3.1. 控制寄存器 (RCU_CTL)

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	PLL2STB	PLL2EN	PLL1STB	PLL1EN	PLLSTB	PLL EN	保留	CKMEN	HXTALB	HXTALST	HXTALE	PS	B	N	
r	rw	r	rw	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	IRC8MST B	IRC8MEN	
r								rw				r	rw		

位/位域	名称	描述
31:30	保留	必须保持复位值
29	PLL2STB	PLL2时钟稳定标志位 硬件置1来表示PLL2输出时钟是否稳定待用 0: PLL2未稳定 1: PLL2已稳定
28	PLL2EN	PLL2使能 软件置位或复位，当PLL2时钟做为系统时钟时该位不能被复位。当进入深度睡眠或待机模式时由硬件复位 0: PLL2被关闭 1: PLL2被打开
27	PLL1STB	PLL1时钟稳定标志位 硬件置1来表示PLL1输出时钟是否稳定待用 0: PLL1未稳定 1: PLL1已稳定
26	PLL1EN	PLL1使能 软件置位或复位，当PLL1时钟做为系统时钟时该位不能被复位。当进入深度睡眠或待机模式时由硬件复位 0: PLL1被关闭 1: PLL1被打开
25	PLLSTB	PLL时钟稳定标志位

		硬件置1来表示PLL输出时钟是否稳定待用 0: PLL未稳定 1: PLL已稳定
24	PLLEN	PLL使能 软件置位或复位，当PLL时钟做为系统时钟时该位不能被复位。当进入深度睡眠或待机模式时由硬件复位 0: PLL被关闭 1: PLL被打开
23:20	保留	必须保持复位值
19	CKMEN	HXTAL时钟监视器使能 0: 禁止高速4 ~ 32 MHz晶体振荡器(HXTAL)时钟监视器 1: 使能高速4 ~ 32 MHz晶体振荡器(HXTAL)时钟监视器 当硬件检测到HXTAL时钟被阻塞在低或高状态时，内部硬件自动切换系统时钟到IRC8M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	高速晶体振荡器(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	高速晶体振荡器(HXTAL)时钟稳定标志位 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	高速晶体振荡器(HXTAL)使能 软件置位或复位，如果HXTAL时钟作为系统时钟或者当PLL时钟做为系统时钟时，其做为PLL的输入时钟，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: 高速4 ~ 32 MHz晶体振荡器被关闭 1: 高速4 ~ 32 MHz晶体振荡器被打开
15:8	IRC8MCALIB[7:0]	内部8MHz RC振荡器校准值寄存器 上电时自动加载这些位
7:3	IRC8MADJ[4:0]	内部8MHz RC振荡器时钟调整值 这些位由软件置位，最终调整值为IRC8MADJ[4:0]位域的当前值加上IRC8MCALIB[7:0]位域的值。最终调整值应该调整IRC8M到8MHz ± 1%
2	保留	必须保持复位值
1	IRC8MSTB	IRC8M内部8MHz RC振荡器稳定标志位 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	内部8MHz RC振荡器使能 软件置位或复位，如果IRC8M时钟做为系统时钟时，该位不能被复位。当从深度睡眠或待机模式返回，或当CKMEN置位同时用作系统时钟的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8MHz RC振荡器被关闭 1: 内部8MHz RC振荡器被打开
---	---------	--

5.3.2. 时钟配置寄存器 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
USBFSPSC[2]	保留	PLLMF[4]	ADC PSC[2]	CKOUT0SEL[3:0]	USBFSPSC[1:0]	PLLMF[3:0]	PREDV0_LSB	PLLSEL							
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]		AHPBSC[3:0]	SCSS[1:0]	SCS[1:0]							
rw		rw		rw		rw	r	rw							

位/位域	名称	描述
31	USBFSPSC[2]	USBFSPSC的第2位 参考寄存器RCU_CFG0的22到23位
30	保留	必须保持复位值
29	PLLMF[4]	PLLMF的第4位 参考寄存器RCU_CFG0的18到21位
28	ADCPSC[2]	ADCPSC的第2位 参考寄存器RCU_CFG0的14到15位
27:24	CKOUT0SEL[3:0]	CKOUT0时钟源选择 由软件置位或清零 00xx: 无时钟输出 0100: 选择系统时钟CK_SYS 0101: 选择内部8M RC振荡器时钟 0110: 选择高速晶体振荡器时钟 (HXTAL) 0111: 选择 (CK_PLL / 2) 时钟 1000: 选择CK_PLL1时钟 1001: 选择 (CK_PLL2 / 2) 时钟 1010: 选择CK_HXTAL时钟 1011: 选择CK_PLL2时钟

		1100: 选择CK_IRC48M时钟 1101: 选择 (CK_IRC48M / 8) 时钟
23:22	USBFSPSC[1:0]	USBFS的时钟分频系数 由软件置位或清零。USBFS的时钟必须为48MHz, 当USBFS时钟使能的时候, 这些位无法修改 000: CK_USBFS = CK_PLL / 1.5 001: CK_USBFS = CK_PLL 010: CK_USBFS = CK_PLL / 2.5 011: CK_USBFS = CK_PLL / 2 100: CK_USBFS = CK_PLL / 3 101: CK_USBFS = CK_PLL / 3.5 11x: CK_USBFS = CK_PLL / 4
21:18	PLLMF[3:0]	PLL时钟倍频因子 与寄存器RCU_CFG0的29位共同构成倍频因子, 由软件置位或清零 注意: PLL输出时钟频率不能超过120MHz 00000: (PLL源时钟 x 2) 00001: (PLL源时钟 x 3) 00010: (PLL源时钟 x 4) 00011: (PLL源时钟 x 5) 00100: (PLL源时钟 x 6) 00101: (PLL源时钟 x 7) 00110: (PLL源时钟 x 8) 00111: (PLL源时钟 x 9) 01000: (PLL源时钟 x 10) 01001: (PLL源时钟 x 11) 01010: (PLL源时钟 x 12) 01011: (PLL源时钟 x 13) 01100: (PLL源时钟 x 14) 01101: (PLL源时钟 x 6.5) 01110: (PLL源时钟 x 16) 01111: (PLL源时钟 x 16) 10000: (PLL源时钟 x 17) 10001: (PLL源时钟 x 18) 10010: (PLL源时钟 x 19) 10011: (PLL源时钟 x 20) 10100: (PLL源时钟 x 21) 10101: (PLL源时钟 x 22) 10110: (PLL源时钟 x 23) 10111: (PLL源时钟 x 24) 11000: (PLL源时钟 x 25) 11001: (PLL源时钟 x 26) 11010: (PLL源时钟 x 27) 11011: (PLL源时钟 x 28)

		11100: (PLL源时钟 x 29) 11101: (PLL源时钟 x 30) 11110: (PLL源时钟 x 31) 11111: (PLL源时钟 x 31)
17	PREDV0_LSB	PREDV0分频因子的最低位 与寄存器RCU_CFG1位PREDV0第0位相同，通过寄存器RCU_CFG1来改变PREDV0的值，此位也会一同改。当PREDV0的第1到3位未修改时，此位决定PREDV0的输入时钟是否二分频。
16	PLLSEL	PLL时钟源选择 由软件置位或复位，控制PLL时钟源 0: (IRC8M / 2)被选择为PLL时钟的时钟源 1: HXTAL时钟或者IRC48M时钟(寄存器RCU_CFG1位PLLPRESEL决定)被选择为PLL时钟的时钟源
15:14	ADCPSC[1:0]	ADC的时钟分频系数 与寄存器RCU_CFG0的28位，寄存器RCU_CFG1的29位共同构成分频因子，由软件置位或清零 0000: CK_ADC = CK_APB2 / 2 0001: CK_ADC = CK_APB2 / 4 0010: CK_ADC = CK_APB2 / 6 0011: CK_ADC = CK_APB2 / 8 0100: CK_ADC = CK_APB2 / 2 0101: CK_ADC = CK_APB2 / 12 0110: CK_ADC = CK_APB2 / 8 0111: CK_ADC = CK_APB2 / 16 1x00: CK_ADC = CK_AHB / 3 1x01: CK_ADC = CK_AHB / 5 1x10: CK_ADC = CK_AHB / 7 1x11: CK_ADC = CK_AHB / 9
13:11	APB2PSC[2:0]	APB2预分频选择 由软件置位或清零，控制APB2时钟分频因子 0xx: CK_APB2 = CK_AHB 100: CK_APB2 = CK_AHB / 2 101: CK_APB2 = CK_AHB / 4 110: CK_APB2 = CK_AHB / 8 111: CK_APB2 = CK_AHB / 16
10:8	APB1PSC[2:0]	APB1预分频选择 由软件置位或清零，控制APB1时钟分频因子. 0xx: CK_APB1 = CK_AHB 100: CK_APB1 = CK_AHB / 2 101: CK_APB1 = CK_AHB / 4 110: CK_APB1 = CK_AHB / 8

111: CK_APB1 = CK_AHB / 16

7:4	AHBPSC[3:0]	AHB预分频选择 由软件置位或清零，控制AHB时钟分频因子。 0xxx: CK_AHB = CK_SYS 1000: CK_AHB = CK_SYS / 2 1001: CK_AHB = CK_SYS / 4 1010: CK_AHB = CK_SYS / 8 1011: CK_AHB = CK_SYS / 16 1100: CK_AHB = CK_SYS / 64 1101: CK_AHB = CK_SYS / 128 1110: CK_AHB = CK_SYS / 256 1111: CK_AHB = CK_SYS / 512
3:2	SCSS[1:0]	系统时钟选择状态 由硬件置位或清零，标识当前系统时钟的时钟源 00: 选择CK_IRC8M时钟作为CK_SYS时钟源 01: 选择CK_HXTAL时钟作为CK_SYS时钟源 10: 选择CK_PLL时钟作为CK_SYS时钟源 11: 保留
1:0	SCS[1:0]	系统时钟选择 由软件配置选择系统时钟源。由于CK_SYS的改变存在固有的延迟，因此软件应当读SCSS位来确保时钟源切换是否结束。在从深度睡眠或待机模式中返回时，以及当HXTAL直接或间接作为系统时钟同时HXTAL时钟监视器检测到HXTAL故障时，强制选择IRC8M作为系统时钟。 00: 选择CK_IRC8M时钟作为CK_SYS时钟源 01: 选择CK_HXTAL时钟作为CK_SYS时钟源 10: 选择CK_PLL时钟作为CK_SYS时钟源 11: 保留

5.3.3. 时钟中断寄存器 (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
保留								CKMIC	PLL2	PLL1	PLL	HXTAL	IRC8M	LXTAL	IRC40K
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	PLL2	PLL1	PLL	HXTAL	IRC8M	LXTAL	IRC40K	CKMIF	PLL2	PLL1	PLL	HXTAL	IRC8M	LXTAL	IRC40K
	STBIE	STBIF													
	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时钟阻塞中断清零 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	PLL2STBIC	PLL2时钟稳定中断清零 软件写1复位PLL2STBIF标志位 0: 不复位PLL2STBIF标志位 1: 复位PLL2STBIF标志位
21	PLL1STBIC	PLL1时钟稳定中断清零 软件写1复位PLL1STBIF标志位 0: 不复位PLL1STBIF标志位 1: 复位PLL1STBIF标志位
20	PLLSTBIC	PLL时钟稳定中断清零 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时钟稳定中断清零 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时钟稳定中断清零 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时钟稳定中断清零 软件写1复位LXTALSTBIF标志位 0: 不复位LXTALSTBIF标志位 1: 复位LXTALSTBIF标志位
16	IRC40KSTBIC	IRC40K时钟稳定中断清零 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	保留	必须保持复位值
14	PLL2STBIE	PLL2时钟稳定中断使能 软件置位和复位来使能/禁止PLL2时钟稳定中断 0: 禁止PLL2时钟稳定中断

		1: 使能PLL2时钟稳定中断
13	PLL1STBIE	<p>PLL1时钟稳定中断使能 软件置位和复位来使能/禁止PLL1时钟稳定中断</p> <p>0: 禁止PLL1时钟稳定中断 1: 使能PLL1时钟稳定中断</p>
12	PLLSTBIE	<p>PLL时钟稳定中断使能 软件置位和复位来使能/禁止PLL时钟稳定中断</p> <p>0: 禁止PLL时钟稳定中断 1: 使能PLL时钟稳定中断</p>
11	HXTALSTBIE	<p>HXTAL时钟稳定中断使能 软件置位和复位来使能/禁止HXTAL时钟稳定中断</p> <p>0: 禁止HXTAL时钟稳定中断 1: 使能HXTAL时钟稳定中断</p>
10	IRC8MSTBIE	<p>IRC8M时钟稳定中断使能 软件置位和复位来使能/禁止IRC8M时钟稳定中断</p> <p>0: 禁止IRC8M时钟稳定中断 1: 使能IRC8M时钟稳定中断</p>
9	LXTALSTBIE	<p>LXTAL时钟稳定中断使能 软件置位和复位来使能/禁止LXTAL时钟稳定中断</p> <p>0: 禁止LXTAL时钟稳定中断 1: 使能LXTAL时钟稳定中断</p>
8	IRC40KSTBIE	<p>IRC40K时钟稳定中断使能 软件置位和复位来使能/禁止IRC40K时钟稳定中断</p> <p>0: 禁止IRC40K时钟稳定中断 1: 使能IRC40K时钟稳定中断</p>
7	CKMIF	<p>HXTAL时钟阻塞中断标志位 当HXTAL时钟被阻塞时由硬件置位. 软件置位CKMIC位时清除该位</p> <p>0: 时钟正常运行 1: HXTAL时钟阻塞</p>
6	PLL2STBIF	<p>PLL2时钟稳定中断标志位 当PLL时钟稳定且PLL2STBIE位被置1时由硬件置1 软件置位PLL2STBIC位时清除该位</p> <p>0: 无PLL2时钟稳定中断产生 1: 产生PLL2时钟稳定中断</p>
5	PLL1STBIF	<p>PLL1时钟稳定中断标志位 当PLL时钟稳定且PLL1STBIE位被置1时由硬件置1 软件置位PLL1STBIC位时清除该位</p> <p>0: 无PLL1时钟稳定中断产生</p>

1: 产生PLL时钟稳定中断

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 软件置位 PLLSTBIC 位时清除该位 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 当高速4~32 MHz晶体振荡器时钟稳定且 HXTALSTBIE 位被置1时由硬件置1 软件置位 HXTALSTBIC 位时清除该位 0: 无HXTAL时钟稳定中断产生 1: 产生HXTAL时钟稳定中断
2	IRC8MSTBIF	IRC8M时钟稳定中断标志位 当内部8MHz RC振荡器时钟稳定且 IRC8MSTBIE 位被置1时由硬件置1 软件置位 IRC8MSTBIC 位时清除该位 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 当低速晶体振荡器时钟稳定且 LXTALSTBIE 位被置1时由硬件置1 软件置位 LXTALSTBIC 位时清除该位 0: 无LXTAL时钟稳定中断产生 1: 产生LXTAL时钟稳定中断
0	IRC40KSTBIF	IRC40K时钟稳定中断标志位 当内部40kHz RC振荡器时钟稳定且 IRC40KSTBIE 位被置1时由硬件置1 软件置位 IRC40KSTBIC 位时清除该位 0: 无IRC40K时钟稳定中断产生 1: 产生IRC40K时钟稳定中断

5.3.4. 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												TIMER10	TIMER9	TIMER8	保留
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	USART0 RST	TIMER7R ST	SPI0RST	TIMER0R ST	ADC1RS T	ADC0RS T	保留	PERST	PDRST	PCRST	PBRST	PARST	保留	AFRST	
	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	

位/位域	名称	描述
31:22	保留	必须保持复位值
21	TIMER10RST	TIMER10复位 由软件置位或复位 0: 无作用 1: 复位TIMER10
20	TIMER9RST	TIMER9复位 由软件置位或复位 0: 无作用 1: 复位TIMER9
19	TIMER8RST	TIMER8复位 由软件置位或复位 0: 无作用 1: 复位TIMER8
18:15	保留	必须保持复位值
14	USART0RST	USART0复位 由软件置位或复位 0: 无作用 1: 复位USART0
13	TIMER7RST	TIMER7复位 由软件置位或复位 0: 无作用 1: 复位TIMER7
12	SPI0RST	SPI0复位 由软件置位或复位 0: 无作用 1: 复位SPI0
11	TIMER0RST	TIMER0复位 由软件置位或复位 0: 无作用 1: 复位TIMER0
10	ADC1RST	ADC1复位 由软件置位或复位 0: 无作用 1: 复位所有ADC1
9	ADC0RST	ADC0复位 由软件置位或复位 0: 无作用

1: 复位所有ADC0

8:7	保留	必须保持复位值
6	PERST	GPIO端口E复位 由软件置位或复位 0: 无作用 1: 复位GPIO端口E
5	PDRST	GPIO端口D复位 由软件置位或复位 0: 无作用 1: 复位GPIO端口D
4	PCRST	GPIO端口C复位 由软件置位或复位 0: 无作用 1: 复位GPIO端口C
3	PBRST	GPIO端口B复位 由软件置位或复位 0: 无作用 1: 复位GPIO端口B
2	PARST	GPIO端口A复位 由软件置位或复位 0: 无作用 1: 复位GPIO端口A
1	保留	必须保持复位值
0	AFRST	复用功能I/O复位 由软件置位或复位 0: 无作用 1: 复位复用功能I/O

5.3.5. 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	DACRST	PMURST	BKPIRST	CAN1RS T	CAN0RS T	保留	I2C1RST	I2C0RST ST	UART4R ST	UART3R ST	USART2 RST	USART1 RST	保留	保留	保留
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

SPI2RST	SPI1RST	保留	WWWDGT	保留	TIMER13	TIMER12	TIMER11	TIMER6R	TIMER5R	TIMER4R	TIMER3R	TIMER2R	TIMER1R
rw	rw		rw		rw								

位/位域	名称	描述
31:30	保留	必须保持复位值
29	DACRST	<p>DAC复位 由软件置位或复位 0: 无作用 1: 复位DAC</p>
28	PMURST	<p>PMU复位 由软件置位或复位 0: 无作用 1: 复位PMU</p>
27	BKPIRST	<p>BKPI复位 由软件置位或复位 0: 无作用 1: 复位BKP</p>
26	CAN1RST	<p>CAN1复位 由软件置位或复位 0: 无作用 1: 复位CAN1</p>
25	CAN0RST	<p>CAN0复位 由软件置位或复位 0: 无作用 1: 复位CAN0</p>
24:23	保留	必须保持复位值
22	I2C1RST	<p>I2C1复位 由软件置位或复位 0: 无作用 1: 复位I2C1</p>
21	I2C0RST	<p>I2C0复位 由软件置位或复位 0: 无作用 1: 复位I2C0</p>
20	UART4RST	<p>UART4复位 由软件置位或复位 0: 无作用</p>

		1: 复位UART4
19	UART3RST	UART3复位 由软件置位或复位 0: 无作用 1: 复位UART3
18	USART2RST	USART2复位 由软件置位或复位 0: 无作用 1: 复位USART2
17	USART1RST	USART1复位 由软件置位或复位 0: 无作用 1: 复位USART1
16	保留	必须保持复位值
15	SPI2RST	SPI2复位 由软件置位或复位 0: 无作用 1: 复位SPI2
14	SPI1RST	SPI1复位 由软件置位或复位 0: 无作用 1: 复位SPI1
13:12	保留	必须保持复位值
11	WWDGTRST	WWDGT复位 由软件置位或复位 0: 无作用 1: 复位WWDGT
10:9	保留	必须保持复位值
8	TIMER13RST	TIMER13复位 由软件置位或复位 0: 无作用 1: 复位TIMER13
7	TIMER12RST	TIMER12复位 由软件置位或复位 0: 无作用 1: 复位TIMER12
6	TIMER11RST	TIMER11复位 由软件置位或复位

		0: 无作用 1: 复位TIMER11
5	TIMER6RST	TIMER6复位 由软件置位或复位 0: 无作用 1: 复位TIMER6
4	TIMER5RST	TIMER5复位 由软件置位或复位 0: 无作用 1: 复位TIMER5
3	TIMER4RST	TIMER4复位 由软件置位或复位 0: 无作用 1: 复位TIMER4
2	TIMER3RST	TIMER3复位 由软件置位或复位 0: 无作用 1: 复位TIMER3
1	TIMER2RST	TIMER2复位 由软件置位或复位 0: 无作用 1: 复位TIMER2
0	TIMER1RST	TIMER1复位 由软件置位或复位 0: 无作用 1: 复位TIMER1

5.3.6. 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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	USBFSE N	保留	EXMCEN	保留	CRCEN	保留	FMCSPE N	保留	SRAMSP EN	DMA1EN	DMA0EN				
	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	USBFSEN	USBFS时钟使能 由软件置位或复位 0: 关闭USBFS时钟 1: 开启USBFS时钟
11:9	保留	必须保持复位值
8	EXMCEN	EXMC时钟使能 由软件置位或复位 0: 关闭EXMC时钟 1: 开启EXMC时钟
7	保留	必须保持复位值
6	CRCEN	CRC时钟使能 由软件置位或复位 0: 关闭CRC时钟 1: 开启CRC时钟
5	保留	必须保持复位值
4	FMCSPEN	在睡眠模式下FMC时钟使能 由软件置位或复位 0: 在睡眠模式下关闭FMC时钟 1: 在睡眠模式下开启FMC时钟
3	保留	必须保持复位值
2	SRAMSPEN	在睡眠模式下SRAM时钟使能 由软件置位或复位 0: 在睡眠模式下关闭SRAM时钟 1: 在睡眠模式下开启SRAM时钟
1	DMA1EN	DMA1时钟使能 由软件置位或复位 0: 关闭DMA1时钟 1: 开启DMA1时钟
0	DMA0EN	DMA0时钟使能 由软件置位或复位 0: 关闭DMA0时钟 1: 开启DMA0时钟

5.3.7. 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								TIMER10 EN	TIMER9E N	TIMER8E N	保留				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留 RW	USART0 EN RW	TIMER7E N RW	SPI0EN RW	TIMER0E N RW	ADC1EN RW	ADC0EN RW	保留 RW	PEEN RW	PDEN RW	PCEN RW	PBEN RW	PAEN RW	保留 RW	AFEN RW	

位/位域	名称	描述
31:22	保留	必须保持复位值
21	TIMER10EN	TIMER10时钟使能 由软件置位或复位 0: 关闭TIMER10时钟 1: 开启TIMER10时钟
20	TIMER9EN	TIMER9时钟使能 由软件置位或复位 0: 关闭TIMER9时钟 1: 开启TIMER9时钟
19	TIMER8EN	TIMER8时钟使能 由软件置位或复位 0: 关闭TIMER8时钟 1: 开启TIMER8时钟
18:15	保留	必须保持复位值
14	USART0EN	USART0时钟使能 由软件置位或复位 0: 关闭USART0时钟 1: 开启USART0时钟
13	TIMER7EN	TIMER7复位 由软件置位或复位 0: 无作用 1: 复位TIMER7
12	SPI0EN	SPI0复位

		由软件置位或复位 0: 无作用 1: 复位SPI0
11	TIMER0EN	TIMER0 复位 由软件置位或复位 0: 无作用 1: 复位 TIMER0
10	ADC1EN	ADC1 时钟使能 由软件置位或复位 0: 关闭 ADC1 时钟 1: 开启 ADC1 时钟
9	ADC0EN	ADC0 时钟使能 由软件置位或复位 0: 关闭 ADC0 时钟 1: 开启 ADC0 时钟
8:7	保留	必须保持复位值
6	PEEN	GPIO 端口E时钟使能 由软件置位或复位 0: 关闭 GPIO 端口E时钟 1: 开启 GPIO 端口E时钟
5	PDEN	GPIO 端口D时钟使能 由软件置位或复位 0: 关闭 GPIO 端口D时钟 1: 开启 GPIO 端口D时钟
4	PCEN	GPIO 端口C时钟使能 由软件置位或复位 0: 关闭 GPIO 端口C时钟 1: 开启 GPIO 端口C时钟
3	PBEN	GPIO 端口B时钟使能 由软件置位或复位 0: 关闭 GPIO 端口B时钟 1: 开启 GPIO 端口B时钟
2	PAEN	GPIO 端口A时钟使能 由软件置位或复位 0: 关闭 GPIO 端口A时钟 1: 开启 GPIO 端口A时钟
1	保留	必须保持复位值
0	AFEN	复用功能IO时钟使能

由软件置位或复位

0: 关闭复用功能IO时钟

1: 开启复用功能IO时钟

5.3.8. 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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	DACEN	PMUEN	BKPIEN	CAN1EN	CAN0EN	保留	I2C1EN	I2C0EN	UART4EN	UART3EN	USART2EN	USART1EN	保留	保留	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
SPI2EN	SPI1EN	保留	WWDGT	保留	TIMER13EN	TIMER12EN	TIMER11EN	TIMER6EN	TIMER5EN	TIMER4EN	TIMER3EN	TIMER2EN	TIMER1EN	保留	保留
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	保留	必须保持复位值
29	DACEN	DAC时钟使能 由软件置位或复位 0: 关闭DAC时钟 1: 开启DAC时钟
28	PMUEN	PMU时钟使能 由软件置位或复位 0: 关闭PMU时钟 1: 开启PMU时钟
27	BKPIEN	BKP时钟使能 由软件置位或复位 0: 关闭BKP时钟 1: 开启BKP时钟
26	CAN1EN	CAN1时钟使能 由软件置位或复位 0: 关闭CAN1时钟 1: 开启CAN1时钟
25	CAN0EN	CAN0时钟使能 由软件置位或复位 0: 关闭CAN0时钟

		1: 开启CAN0时钟
24:23	保留	必须保持复位值
22	I2C1EN	<p>I2C1时钟使能</p> <p>由软件置位或复位</p> <p>0: 关闭I2C1时钟</p> <p>1: 开启I2C1时钟</p>
21	I2C0EN	<p>I2C0时钟使能</p> <p>由软件置位或复位</p> <p>0: 关闭I2C0时钟</p> <p>1: 开启I2C0时钟</p>
20	UART4EN	<p>UART4时钟使能</p> <p>由软件置位或复位</p> <p>0: 关闭UART4时钟</p> <p>1: 开启UART4时钟</p>
19	UART3EN	<p>UART3时钟使能</p> <p>由软件置位或复位</p> <p>0: 关闭UART3时钟</p> <p>1: 开启UART3时钟</p>
18	USART2EN	<p>USART2时钟使能</p> <p>由软件置位或复位</p> <p>0: 关闭USART2时钟</p> <p>1: 开启USART2时钟</p>
17	USART1EN	<p>USART1时钟使能</p> <p>由软件置位或复位</p> <p>0: 关闭USART1时钟</p> <p>1: 开启USART1时钟</p>
16	保留	必须保持复位值
15	SPI2EN	<p>SPI2时钟使能</p> <p>由软件置位或复位</p> <p>0: 关闭SPI2时钟</p> <p>1: 开启SPI2时钟</p>
14	SPI1EN	<p>SPI1时钟使能</p> <p>由软件置位或复位</p> <p>0: 关闭SPI1时钟</p> <p>1: 开启SPI1时钟</p>
13:12	保留	必须保持复位值
11	WWDGTEN	<p>WWDGT时钟使能</p> <p>由软件置位或复位</p>

		0: 关闭WWDGT时钟 1: 开启WWDGT时钟
10:9	保留	必须保持复位值
8	TIMER13EN	TIMER13时钟使能 由软件置位或复位 0: 关闭TIMER13时钟 1: 开启TIMER13时钟
7	TIMER12EN	TIMER12时钟使能 由软件置位或复位 0: 关闭TIMER12时钟 1: 开启TIMER12时钟
6	TIMER11EN	TIMER11时钟使能 由软件置位或复位 0: 关闭TIMER11时钟 1: 开启TIMER11时钟
5	TIMER6EN	TIMER6时钟使能 由软件置位或复位 0: 关闭TIMER6时钟 1: 开启TIMER6时钟
4	TIMER5EN	TIMER5时钟使能 由软件置位或复位 0: 关闭TIMER5时钟 1: 开启TIMER5时钟
3	TIMER4EN	TIMER4时钟使能 由软件置位或复位 0: 关闭TIMER4时钟 1: 开启TIMER4时钟
2	TIMER3EN	TIMER3时钟使能 由软件置位或复位 0: 关闭TIMER3时钟 1: 开启TIMER3时钟
1	TIMER2EN	TIMER2时钟使能 由软件置位或复位 0: 关闭TIMER2时钟 1: 开启TIMER2时钟
0	TIMER1EN	TIMER1时钟使能 由软件置位或复位 0: 关闭TIMER1时钟

1: 开启TIMER1时钟

5.3.9. 备份域控制寄存器 (RCU_BDCTL)

地址偏移: 0x20

复位值: 0x0000 0018, 只能由备份域复位进行复位

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(RCU_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BKPWEN位置1后才能对这些位进行改动。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															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]	LXTALBP S	LXTALST B	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	备份域复位 由软件置位或复位 0: 无作用 1: 复位备份域
15	RTCEN	RTC时钟使能 由软件置位或复位 0: 关闭RTC时钟 1: 开启RTC时钟
14:10	保留	必须保持复位值
9:8	RTCSRC[1:0]	RTC时钟源选择 由软件置位或清零来控制RTC的时钟源。一旦RTC的时钟源选择后，除了将备份域复位否则时钟源不能被改变。 00: 没有时钟 01: 选择CK_LXTAL时钟作为RTC的时钟源 10: 选择CK_IRC40K时钟作为RTC的时钟源 11: 选择CK_HXTAL / 128时钟作为RTC的时钟源
7:5	保留	必须保持复位值
4:3	LXTALDRI[1:0]	LXTAL驱动能力

由软件置位或复位。当备份域复位时将复位该值

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)

注意: LXTALDRI位在旁路模式下无效

2	LXTALBPS	LXTAL旁路模式使能 由软件置位或复位 0: 禁止LXTAL旁路模式 1: 使能LXTAL旁路模式
1	LXTALSTB	低速晶体振荡器稳定标志位 硬件置‘1’来指示LXTAL振荡器时钟是否稳定待用 0: LXTAL未稳定 1: LXTAL已稳定
0	LXTALEN	LXTAL时钟使能 由软件置位或复位 0: 关闭LXTAL时钟 1: 使能LXTAL时钟

5.3.10. 复位源/时钟寄存器 (RCU_RSTSCK)

地址偏移: 0x24

复位值: 0x0C00 0000, 所有复位标志位仅在电源复位时被清零, RSTFC/IRC40KEN在系统复位时被清零。

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	WWDGTRSTF	FWDGTRSTF	SW RSTF	POR RSTF	EP RSTF	保留	RSTFC					保留			
r	r	r	r	r	r			rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						保留							IRC40K STB	IRC40KE N	
													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位 向RSTFC位写1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	<p>窗口看门狗定时器复位标志位</p> <p>窗口看门狗定时器复位发生时由硬件置1</p> <p>向RSTFC位写1来清除该位</p> <p>0: 无窗口看门狗复位发生</p> <p>1: 发生窗口看门狗复位</p>
29	FWDGTRSTF	<p>独立看门狗定时器复位标志位</p> <p>独立看门狗复位发生时由硬件置1</p> <p>向RSTFC位写1来清除该位</p> <p>0: 无独立看门狗定时器复位发生</p> <p>1: 发生独立看门狗定时器复位</p>
28	SWRSTF	<p>软件复位标志位</p> <p>软件复位发生时由硬件置1</p> <p>向RSTFC位写1来清除该位</p> <p>0: 无软件复位发生</p> <p>1: 发生软件复位</p>
27	PORRSTF	<p>电源复位标志位</p> <p>电源复位发生时由硬件置1</p> <p>向RSTFC位写1来清除该位</p> <p>0: 无电源复位发生</p> <p>1: 发生电源复位</p>
26	EPRSTF	<p>外部引脚复位标志位</p> <p>外部引脚复位发生时由硬件置1</p> <p>向RSTFC位写1来清除该位</p> <p>0: 无外部引脚复位发生</p> <p>1: 发生外部引脚复位</p>
25	保留	必须保持复位值
24	RSTFC	<p>清除复位标志位</p> <p>由软件置1来清除所有复位标志位</p> <p>0: 无作用</p> <p>1: 清除所有复位标志位</p>
23:2	保留	必须保持复位值
1	IRC40KSTB	<p>IRC40K时钟稳定标志位</p> <p>该位由硬件置1指示IRC40K输出时钟是否稳定待用</p> <p>0: IRC40K时钟未稳定</p> <p>1: IRC40K已稳定</p>
0	IRC40KEN	<p>IRC40K使能</p> <p>由软件置位和复位</p> <p>0: 关闭IRC40K时钟</p>

1: 开启IRC40K时钟

5.3.11. 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

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	描述
31:13	保留	必须保持复位值
12	USBFSRST	USBFS复位 由软件置位或复位 0: 无作用 1: 复位USBFS
11:0	保留	必须保持复位值

5.3.12. 时钟配置寄存器 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
保留															
保留	PLLREPS EL	ADC PSC[3]	保留								I2S2SEL	I2S1SEL	PREDV0 SEL		
rw rw rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLL2MF[3:0]				PLL1MF[3:0]				PREDV1[3:0]				PREDV0[3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31	保留	必须保持复位值

30	PLLPRESEL	PLL时钟源预选择 由软件置位或复位，控制PLL时钟源 0: HXTAL被选择为PLL时钟的时钟源 1: CK_IRC48M被选择为PLL时钟的时钟源
29	ADCPSC[3]	ADCPSC的第3位 参考寄存器RCU_CFG0的14到15位
28:19	保留	必须保持复位值
18	I2S2SEL	I2S2时钟源选择 由软件置位或复位，控制I2S2时钟源 0: 系统时钟被选择为I2S2时钟的时钟源 1: (CK_PLL2 x 2) 被选择为I2S2时钟的时钟源
17	I2S1SEL	I2S1时钟源选择 由软件置位或复位，控制I2S1时钟源 0: 系统时钟被选择为I2S1时钟的时钟源 1: (CK_PLL2 x 2) 被选择为I2S1时钟的时钟源
16	PREDV0SEL	PREDV0时钟源选择 由软件置位或复位 0: HXTAL或IRC48M被选择为PREDV0的时钟源 1: CK_PLL1被选择为PREDV0的时钟源
15:12	PLL2MF[3:0]	PLL2时钟倍频因子 倍频因子，由软件置位或清零 00xx: 保留 010x: 保留 0110: (PLL2源时钟 x 8) 0111: (PLL2源时钟 x 9) 1000: (PLL2源时钟 x 10) 1001: (PLL2源时钟 x 11) 1010: (PLL2源时钟 x 12) 1011: (PLL2源时钟 x 13) 1100: (PLL2源时钟 x 14) 1101: 保留 1110: (PLL2源时钟 x 16) 1111: (PLL2源时钟 x 20)
11:8	PLL1MF[3:0]	PLL1时钟倍频因子 由软件置位或清零 00xx: 保留 010x: 保留 0110: (PLL1源时钟 x 8) 0111: (PLL1源时钟 x 9)

1000 :(PLL1源时钟 x 10)

1001: (PLL1源时钟 x 11)

1010: (PLL1源时钟 x 12)

1011: (PLL1源时钟 x 13)

1100: (PLL1源时钟 x 14)

1101: 保留

1110: (PLL1源时钟 x 16)

1111: (PLL1源时钟 x 20)

7:4

PREDV1[3:0]

PREDV1分频因子

由软件置位或清零，PLL1和PLL2未使能时，可以修改这些位

0000: PREDV1输入源时钟未分频

0001: PREDV1输入源时钟2分频

0010: PREDV1输入源时钟3分频

0011: PREDV1输入源时钟4分频

0100: PREDV1输入源时钟5分频

0101: PREDV1输入源时钟6分频

0110: PREDV1输入源时钟7分频

0111: PREDV1输入源时钟8分频

1000: PREDV1输入源时钟9分频

1001: PREDV1输入源时钟10分频

1010: PREDV1输入源时钟11分频

1011: PREDV1输入源时钟12分频

1100: PREDV1输入源时钟13分频

1101: PREDV1输入源时钟14分频

1110: PREDV1输入源时钟15分频

1111: PREDV1输入源时钟16分频

3:0

PREDV0[3:0]

PREDV0分频因子

由软件置位或清零，PLL未使能时，可以修改这些位

注意：PREDV0的第0位与RCU_CFG0寄存器的17位相同，修改RCU_CFG0寄存器的17位，PREDV0的第0位也会进行相同的修改

0000: PREDV0输入源时钟未分频

0001: PREDV0输入源时钟2分频

0010: PREDV0输入源时钟3分频

0011: PREDV0输入源时钟4分频

0100: PREDV0输入源时钟5分频

0101: PREDV0输入源时钟6分频

0110: PREDV0输入源时钟7分频

0111: PREDV0输入源时钟8分频

1000: PREDV0输入源时钟9分频

1001: PREDV0输入源时钟10分频

1010: PREDV0输入源时钟11分频

1011: PREDV0输入源时钟12分频

1100: PREDV0输入源时钟13分频

- 1101: PREDV0输入源时钟14分频
- 1110: PREDV0输入源时钟15分频
- 1111: PREDV0输入源时钟16分频

5.3.13. 深度睡眠模式电压寄存器 (RCU_DSV)

地址偏移: 0x34

复位值: 0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问

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

rw

位/位域	名称	描述
31:2	保留	必须保持复位值
1:0	DSLPVS[1:0]	深度睡眠模式电压选择 由软件置位和清零这些位 000: 在深度睡眠模式下内核电压为1.0V 001: 在深度睡眠模式下内核电压为0.9V 010: 在深度睡眠模式下内核电压为0.8V 011: 在深度睡眠模式下内核电压为1.2V

5.3.14. 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					IRC48MS		IRC48ME			
										TB		N			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留														CK48MS	
														EL	

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
15:1	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。 CK48M时钟用于： 0: 不选择IRC48M时钟(使用CK_PLL/USBFSPSC时钟) 1: 选择IRC48M时钟

5.3.15. 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										IRC48MS	保留					
W																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
保留	IRC48MS TBIE	保留										IRC48MS TBIF	保留			
rw								r								

位/位域	名称	描述
31:23	保留	必须保持复位值
22	IRC48MSTBIC	内部48 MHz 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

5.3.16. APB1 附加复位寄存器 (RCU_ADDAPB1RST)

地址偏移: 0xE0

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

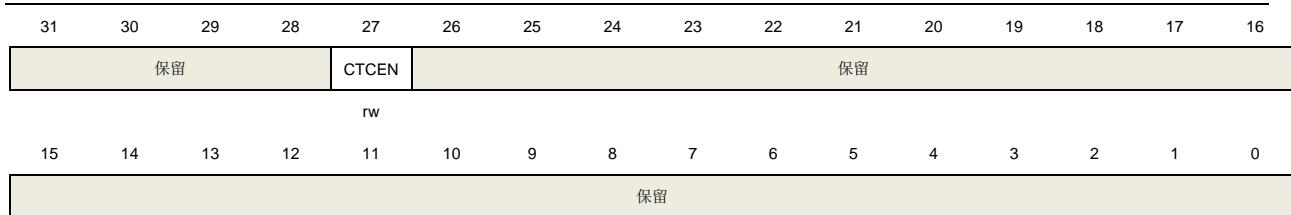
位/位域	名称	描述
31:28	保留	必须保持复位值
27	CTCRST	CTC复位 由软件置位或复位 0: 无作用 1: 复位CTC
26:0	保留	必须保持复位值

5.3.17. APB1 附加使能寄存器 (RCU_ADDAPB1EN)

地址偏移: 0xE4

复位值: 0x0000 0000

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问



位/位域	名称	描述
31:28	保留	必须保持复位值
27	CTCEN	<p>CTC时钟使能</p> <p>由软件置位或复位</p> <p>0: 关闭CTC时钟</p> <p>1: 开启CTC时钟</p>
26:0	保留	必须保持复位值

6. 时钟校准控制器（CTC）

6.1. 简介

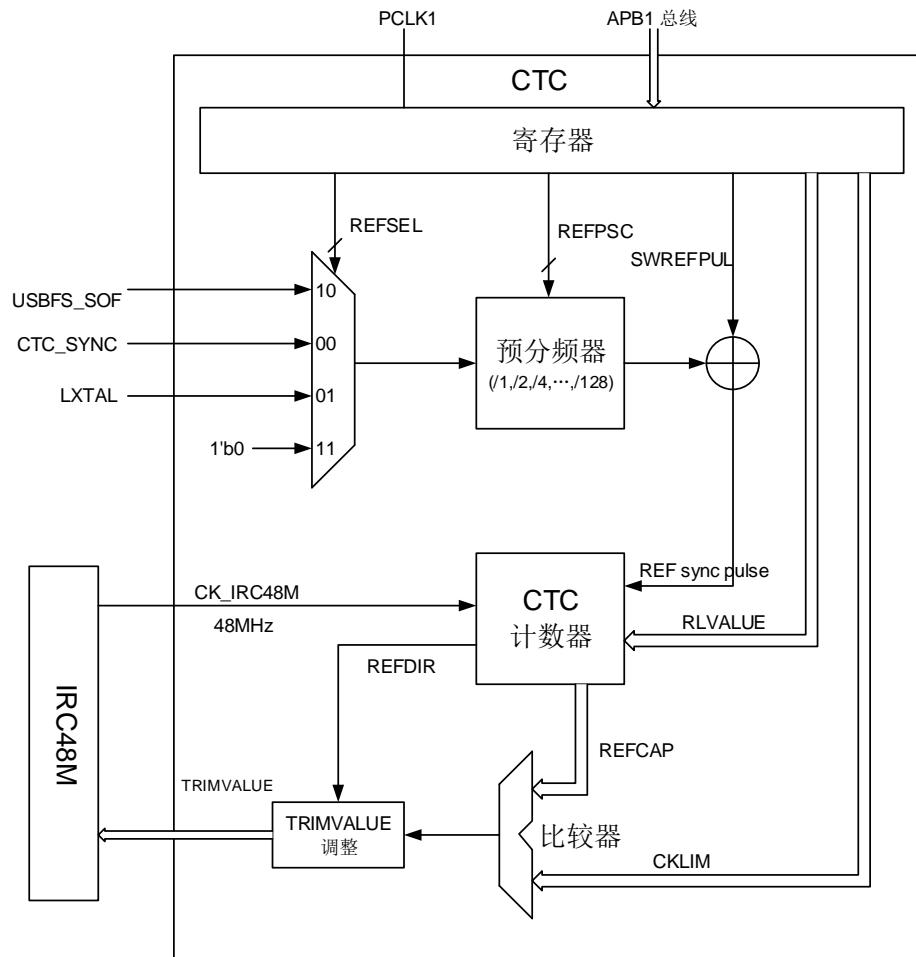
时钟校准控制器（CTC）采用硬件的方式，自动校准内部48MHz RC晶振（IRC48M）。当USBFS模块使用IRC48M时钟作为时钟源时，必须要求IRC48M时钟频率在 $48\text{MHz}\pm500\text{ppm}$ 的范围内。内部晶振无法满足这么高的精度，需要校准IRC48M时钟。CTC模块基于外部高精度的参考信号源来校准IRC48M的时钟频率，通过自动的或手动的调整校准值，以得到一个精准的IRC48M时钟。

6.2. 主要特征

- 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位校准计数器；
- 用于频率评估和自动校准的8位时钟校准基值；
- 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）。

6.3. 功能说明

图 6-1. CTC 结构框图



6.3.1. 参考同步脉冲发生器

首先，通过设置CTC_CTL1寄存器中的REFSEL位来选择参考信号源：GPIO，LXTAL时钟输出或者USBFS SOF。

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，并且通过设置CTC_CTL1寄存器中的REFPSC位进行分频从而产生一个合适的同步时钟频率。

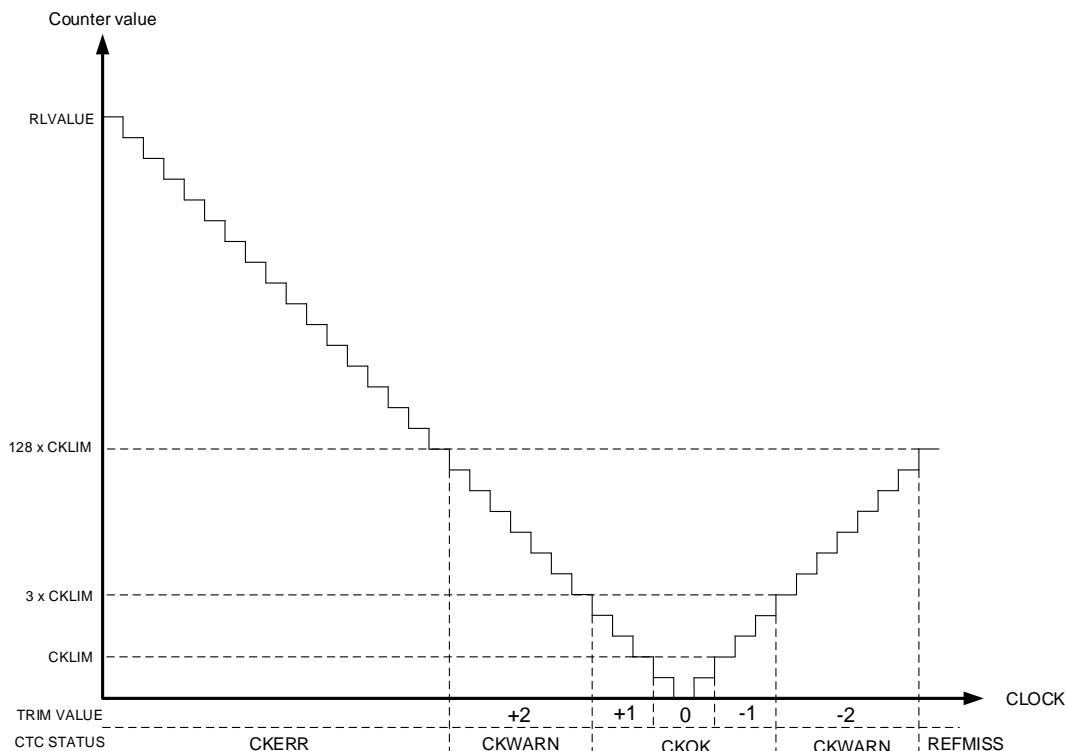
如果需要使用软件参考脉冲信号，则需要设置CTC_CTL0寄存器（CTC控制寄存器0）中的SWREFPUL位为1。软件参考脉冲信号与外部参考脉冲信号最后进行逻辑或操作。

6.3.2. CTC 校准计数器

CTC 时钟校准计数器由 CK_IRC48M 提供时钟。在 CTC_CTL0 寄存器中的 CNTEN 置位后，当检测到第一个参考同步脉冲信号，计数器开始从 RLVALUE 值（RLVALUE 在 CTC_CTL1 寄存器中定义）向下计数。每次检测到参考同步脉冲信号时，计数器重载 RLVALUE 值，同时重

新开始向下计数。如果始终检测不到参考同步脉冲信号，计数器会向下计数到零，然后再向上计数到 $128 \times CKLIM$ ($CKLIM$ 在 CTC_CTL1 中定义)，最后停止，直到检测到下一个参考同步脉冲信号。一旦检测到参考同步脉冲信号，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。详细内容如下图所示。

图 6-2. CTC 校准计数器



6.3.3. 频率评估和自动校准过程

当参考同步脉冲信号出现时，时钟频率评估功能开始执行。如果参考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）。如果参考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 寄存器中的 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，如果参考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，如果参考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，检测到参考同步脉冲信号；

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。

- $CKLIM \leq \text{Counter} < 3 \times CKLIM$ 时，检测到参考同步脉冲信号；
 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。
 - $3 \times CKLIM \leq \text{Counter} < 128 \times CKLIM$ 时，检测到参考同步脉冲信号；
 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。
 - $\text{Counter} \geq 128 \times CKLIM$ ，计数器在向下计数过程中，检测到参考同步脉冲信号；
 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。
 - $\text{Counter} = 128 \times CKLIM$ ，计数器在向上计数过程中；
 CTC_STAT中的REFMISS位（参考同步脉冲丢失位）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。

6.3.4. 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）和参考同步脉冲信号的频率计算得到。理想状态是参考同步脉冲信号在CTC计数器计数到零时出现，所以RLVALUE的值为：

$$\text{RLVALUE} = (\text{F}_{\text{clock}} \div \text{F}_{\text{REF}}) - 1 \quad (6-1)$$

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：

$$\text{CKLIM} = (\text{F}_{\text{clock}} \div \text{F}_{\text{REF}}) \times 0.12\% \div 2 \quad (6-2)$$

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

6.4. CTC 寄存器

CTC基地址: 0x4000 C800

6.4.1. 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。

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>软件生成参考同步脉冲信号</p> <p>该位由软件置位，并为CTC计数器提供一个参考同步脉冲信号。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>

		1: 使能 CTC 计数器
4	保留	必须保持复位值
3	EREFIE	期望参考信号中断使能 0: 禁止期望参考信号产生中断 1: 使能期望参考信号产生中断
2	ERRIE	错误中断使能 0: 禁止错误中断 1: 使能错误中断
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

6.4.2. 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
REFPOL	保留	REFSEL[1:0]	保留	REFPSC[2:0]							CKLIM[7:0]				
rw															
5	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
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，当检测到一个参考同步脉冲时，该值将重载到CTC校准计数器中。

6.4.3. 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			TRIMERR	REFMISS	CKERR	保留			EREFIF	ERRIF	CKWARN	CKOK		
	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	<p>校准值错误位</p> <p>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。</p> <p>0: 无校准值错误发生 1: 发生校准值错误</p>
9	REFMISS	<p>参考同步脉冲信号丢失</p> <p>当参考同步脉冲信号丢失时，该位由硬件置位。当CTC校准计数器在向上增加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: 无参考同步脉冲信号丢失 1: 参考同步脉冲信号丢失</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向下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>

0: 无时钟校准警告发生

1: 有时钟校准警告发生

0	CKOKIF	时钟校准成功中断标志位 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。
		0: 时钟校准未成功 1: 时钟校准成功

6.4.4. 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
保留															

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，REFMISS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。

7. 中断/事件控制器 (EXTI)

7.1. 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19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。

7.2. 主要特征

- Cortex-M4 系统异常；
- 多达 66 种可屏蔽的外设中断；
- 4 位中断优先级配置位——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 中有多达 19 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。

7.3. 功能说明

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。如下表列出了Cortex-M4中的 NVIC 异常类型。

表 7-1. Cortex-M3 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
存储器管理	4	可编程设置	0x0000_0010	存储器管理

异常类型	向量编号	优先级(a)	向量地址	描述
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
SysTick	15	可编程设置	0x0000_003C	系统节拍定时器

SysTick标定值为15000，并且SysTick时钟频率固定为HCLK*0.125。当HCLK配置为120MHz时，会产生1ms的SysTick中断。

表 7-2. 中断向量表

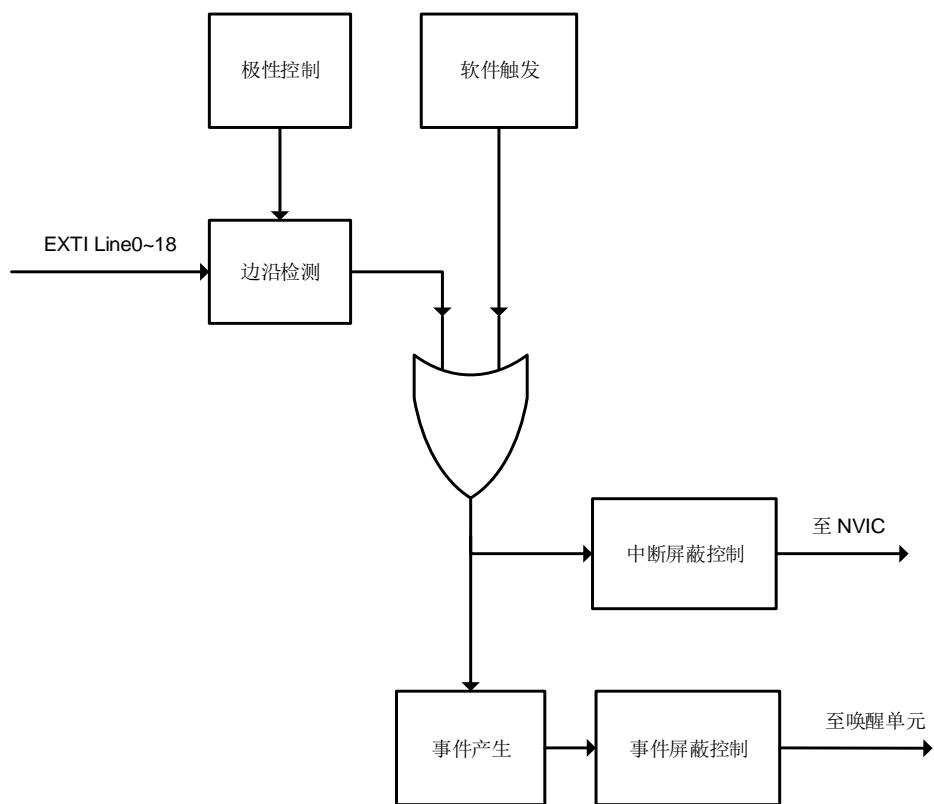
中断编号	向量编号	外设中断描述	向量地址
IRQ 0	16	窗口看门狗中断	0x0000_0040
IRQ 1	17	连接到 EXTI 线的 LVD 中断	0x0000_0044
IRQ 2	18	侵入检测中断	0x0000_0048
IRQ 3	19	RTC 全局中断	0x0000_004C
IRQ 4	20	FMC 全局中断	0x0000_0050
IRQ 5	21	RCU 和 CTC 中断	0x0000_0054
IRQ 6	22	EXTI 线 0 中断	0x0000_0058
IRQ 7	23	EXTI 线 1 中断	0x0000_005C
IRQ 8	24	EXTI 线 2 中断	0x0000_0060
IRQ 9	25	EXTI 线 3 中断	0x0000_0064
IRQ 10	26	EXTI 线 4 中断	0x0000_0068
IRQ 11	27	DMA0 通道 0 全局中断	0x0000_006C
IRQ 12	28	DMA0 通道 1 全局中断	0x0000_0070
IRQ 13	29	DMA0 通道 2 全局中断	0x0000_0074
IRQ 14	30	DMA0 通道 3 全局中断	0x0000_0078
IRQ 15	31	DMA0 通道 4 全局中断	0x0000_007C
IRQ 16	32	DMA0 通道 5 全局中断	0x0000_0080
IRQ 17	33	DMA0 通道 6 全局中断	0x0000_0084
IRQ 18	34	ADC0 和 ADC1 全局中断	0x0000_0088
IRQ 19	35	CAN0 发送中断	0x0000_008C
IRQ 20	36	CAN0 接收 0 中断	0x0000_0090
IRQ 21	37	CAN0 接收 1 中断	0x0000_0094
IRQ 22	38	CAN0 EWMC 中断	0x0000_0098
IRQ 23	39	EXTI 线[9:5] 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	TIMER0 中止中断和 TIMER8 全局中断	0x0000_00A0
IRQ 25	41	TIMER0 更新中断和 TIMER9 全局中断	0x0000_00A4
IRQ 26	42	TIMER0 触发与通道换相中断 和 TIMER10 全局中断	0x0000_00A8
IRQ 27	43	TIMER0 通道捕获比较中断	0x0000_00AC
IRQ 28	44	TIMER1 全局中断	0x0000_00B0
IRQ 29	45	TIMER2 全局中断	0x0000_00B4
IRQ 30	46	TIMER3 全局中断	0x0000_00B8
IRQ 31	47	I2C0 事件中断	0x0000_00BC
IRQ 32	48	I2C0 错误中断	0x0000_00C0
IRQ 33	49	I2C1 事件中断	0x0000_00C4
IRQ 34	50	I2C1 错误中断	0x0000_00C8
IRQ 35	51	SPI0 全局中断	0x0000_00CC
IRQ 36	52	SPI1 全局中断	0x0000_00D0
IRQ 37	53	USART0 全局中断	0x0000_00D4
IRQ 38	54	USART1 全局中断	0x0000_00D8
IRQ 39	55	USART2 全局中断	0x0000_00DC
IRQ 40	56	EXTI 线[15:10] 中断	0x0000_00E0
IRQ 41	57	连接 EXTI 线的 RTC 闹钟中断	0x0000_00E4
IRQ 42	58	连接 EXTI 线的 USBFS 唤醒 中断	0x0000_00E8
IRQ 43	59	TIMER7 中止中断和 TIMER11 全局中断	0x0000_00EC
IRQ 44	60	TIMER7 更新中断和 TIMER12 全局中断	0x0000_00F0
IRQ 45	61	TIMER7 触发与通道换相中断 和 TIMER13 全局中断	0x0000_00F4
IRQ 46	62	TIMER7 通道捕获比较中断	0x0000_00F8
IRQ 47	63	保留	0x0000_00FC
IRQ 48	64	EXMC 全局中断	0x0000_0100
IRQ 49	65	保留	0x0000_0104
IRQ50	66	TIMER4 全局中断	0x0000_0108
IRQ51	67	SPI2 全局中断	0x0000_010C
IRQ52	68	UART3 全局中断	0x0000_0110
IRQ53	69	UART4 全局中断	0x0000_0114
IRQ54	70	TIMER5 全局中断	0x0000_0118
IRQ55	71	TIMER6 全局中断	0x0000_011C
IRQ56	72	DMA1 通道 0 全局中断	0x0000_0120

中断编号	向量编号	外设中断描述	向量地址
IRQ57	73	DMA1 通道 1 全局中断	0x0000_0124
IRQ58	74	DMA1 通道 2 全局中断	0x0000_0128
IRQ59	75	DMA1 通道 3 全局中断	0x0000_012C
IRQ60	76	DMA1 通道 4 全局中断	0x0000_0130
IRQ61	77	保留	0x0000_0134
IRQ62	78	保留	0x0000_0138
IRQ63	79	CAN1 发送中断	0x0000_013C
IRQ64	80	CAN1 接收 0 中断	0x0000_0140
IRQ65	81	CAN1 接收 1 中断	0x0000_0144
IRQ66	82	CAN1 EWMC 中断	0x0000_0148
IRQ67	83	USBFS 全局中断	0x0000_014C

7.4. 外部中断及事件(EXTI) 框图

图 7-1. EXTI 框图



7.5. 外部中断及事件功能概述

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3根线。(包括LVD、RTC闹钟、USBFS唤醒)。通过配置GPIO模块的AFIO_EXTISx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GPIO章节AFIO部分。

除了中断，EXTI还可以向处理器提供事件信号。The 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。芯片内部有一个唤醒中断控制器(WIC)，用户可以放心的让处理器和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闹钟动作。

表 7-3. EXTI 触发源

EXTI 线编号	触发源
0	PA0/PB0/PC0/PD0/PE0
1	PA1/PB1/PC1/PD1/PE1
2	PA2/PB2/PC2/PD2/PE2
3	PA3/PB3/PC3/PD3/PE3
4	PA4/PB4/PC4/PD4/PE4
5	PA5/PB5/PC5/PD5/PE5
6	PA6/PB6/PC6/PD6/PE6
7	PA7/PB7/PC7/PD7/PE7
8	PA8/PB8/PC8/PD8/PE8
9	PA9/PB9/PC9/PD9/PE9
10	PA10/PB10/PC10/PD10/PE10
11	PA11/PB11/PC11/PD11/PE11
12	PA12/PB12/PC12/PD12/PE12
13	PA13/PB13/PC13/PD13/PE13
14	PA14/PB14/PC14/PD14/PE14
15	PA15/PB15/PC15/PD15/PE15
16	LVD
17	RTC 闹钟
18	USBFS 唤醒

7.6. EXTI 寄存器

EXTI基地址: 0x4001 0400

7.6.1. 中断使能寄存器 (EXTI_INTEN)

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															INTEN18 INTEN17 INTEN16
															rW 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	INTEN0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

位/位域	名称	描述
31:19	保留	必须保持复位值
18: 0	INTEN x	中断使能位 0: 第 x 线中断被禁用 1: 第 x 线中断被使能

7.6.2. 事件使能寄存器 (EXTI_EVENT)

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															EVEN18 EVEN17 EVEN16
															rW rW 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	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

位/位域	名称	描述
31:19	保留	必须保持复位值
18: 0	EVEN x	事件使能位 0: 第 x 线事件被禁用. 1: 第 x 线事件被使能

7.6.3. 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															RTEN18 RTEN17 RTEN16
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
31:19	保留	必须保持复位值
18:0	RTENx	上升沿触发使能 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)

7.6.4. 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															FTEN18 FTEN17 FTEN16
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: 19	保留	必须保持复位值
18: 0	FTENx	下降沿触发使能 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)

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

位/位域	名称	描述
31: 19	保留	必须保持复位值
18: 0	SWIEV x	中断/事件软件触发 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

7.6.6. 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														PD18	PD17	PD16
rc_w1 rc_w1 rc_w1														rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	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: 19	保留	必须保持复位值
18: 0	PD x	中断挂起状态 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。

8. 通用和备用输入/输出接口（GPIO 和 AFIO）

8.1. 简介

最多可支持80个通用I/O引脚(GPIO)，分别为PA0 ~ PA15，PB0 ~ PB15，PC0 ~ PC15，PD0 ~ PD15，PE0 ~ PE15，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引脚上的外部中断在中断/事件控制器(EXTI)中有相关的控制和配置寄存器。

GPIO端口和其他的备用功能(AFs)共用引脚，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输入/输出引脚。

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。

8.2. 主要特征

- 输入/输出方向控制；
- 施密特触发器输入功能使能控制；
- 每个引脚都具有弱上拉/下拉功能；
- 推挽/开漏输出使能控制；
- 置位/复位输出使能；
- 可编程触发沿的外部中断—使用EXTI配置寄存器
- 模拟输入/输出配置；
- 备用功能输入/输出配置；
- 端口配置锁定。

8.3. 功能说明

每个通用I/O端口都可以通过两个32位的控制寄存器(GPIOx_CTL0/ GPIOx_CTL1)和一个32位寄存器(GPIOx_OCTL)配置为8种模式：模拟输入，浮空输入，上拉输入，下拉输入，GPIO推挽输出，GPIO开漏输出，备用推挽输出和备用开漏输出。详情请见[表8-1. GPIO配置表](#)。

表 8-1. GPIO 配置表

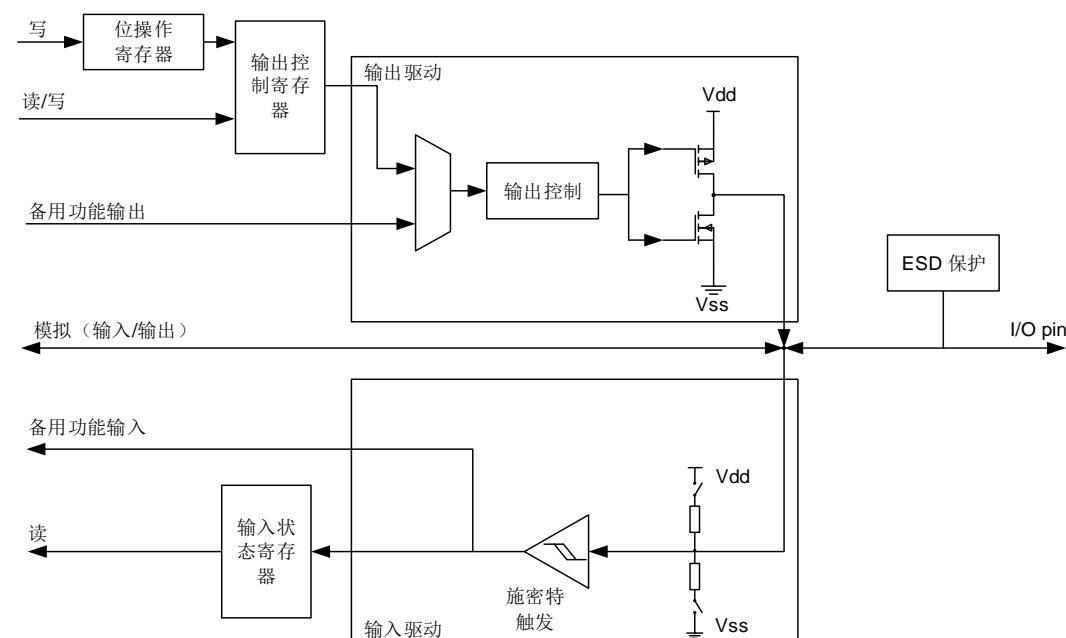
配置模式		CTL[1:0]	MD[1:0]	OCTL
输入	模拟	00	00	不使用
	浮空输入	01		不使用
	下拉输入	10		0

	上拉输入	10		1
	配置模式	CTL[1:0]	SPDy: MD[1:0]	OCTL
普通输出 (GPIO)	推挽	00	00: 保留 01: 最大速度到 10MHz 10: 最大速度到 2MHz 11: 最大速度到 50MHz	0 或 1 0 或 1 不使用
	开漏	01	11: 最大速度到 120MHz ⁽¹⁾ (同时设置SPDy 值为1)	不使用
备用功能输出 (AFIO)	推挽	10		
	开漏	11		

1、当GPIO输出速度超过50MHz时，需要使能GPIO的补偿单元，参考IO补偿控制寄存器（AFIO_CPSCTL）。

图 8-1. 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 8-1. 标准 I/O 端口位的基本结构



8.3.1. 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，串行线调试端口（JTAG/Serial-Wired Debug pins）为输入PU/PD模式：

- PA15: JTDI为上拉模式；
- PA14: JTCK / SWCLK为下拉模式；
- PA13: JTMS / SWDIO为上拉模式；
- PB4: NJTRST为上拉模式；
- PB3: JTDO为浮空模式。

GPIO引脚可以配置为输入或输出模式，当GPIO引脚可配置为输入引脚时，所有的GPIO引脚内部都有一个可选择的弱上拉和弱下拉电阻。外部引脚上的数据在每个APB2时钟周期时都会装载到数据输入寄存器(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，输出寄存器(GPIOx_OCTL)的值将会从相应I/O引脚上输出。

当对GPIOx_OCTL进行位操作时，不需要先读再写，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)修改一位或几位，该过程仅需要一个最小的APB2写访问周期，而其他位不受影响。

8.3.2. 外部中断/事件线

所有的端口都有外部中断能力，为了使用外部中断线，端口必须配置为输入模式。

8.3.3. 备用功能(AF)

当端口配置为AFIO（设置GPIOx_CTL0/GPIOx_CTL1寄存器中的CTLy值为“0b10”或“0b11”，MDy位值为“0b01”，“0b10”或“0b11”）时，该端口用作外设备用功能。端口备用功能分配的详细介绍见芯片数据手册。

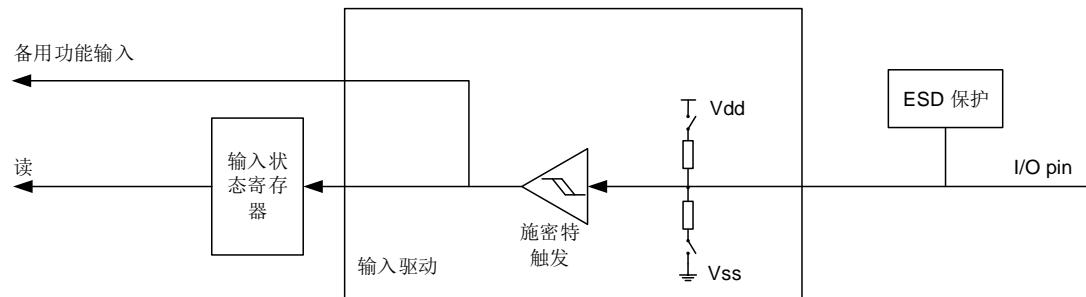
8.3.4. 输入配置

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APB2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 8-2. 输入配置显示 I/O 引脚的输入配置。

图 8-2. 输入配置



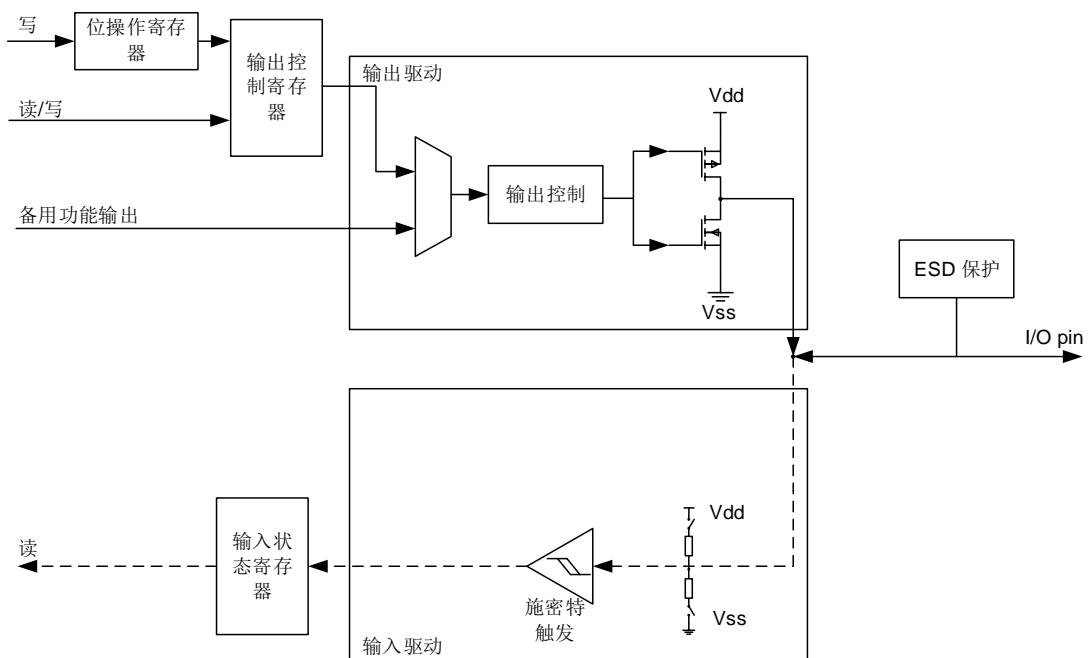
8.3.5. 输出配置

当GPIO配置为输出时：

- 施密特触发输入使能；
- 弱上拉和下拉电阻禁用；
- 输出缓冲器使能；
- 开漏模式：输出控制寄存器设置为“0”时，相应引脚输出低电平；输出控制寄存器设置为“1”，相应管脚处于高阻状态；
- 推挽模式：输出控制寄存器设置为“0”时，相应引脚输出低电平；输出控制寄存器设置为“1”，相应引脚输出高电平；
- 对端口输出控制寄存器进行读操作，将返回上次写入的值；
- 对端口输入状态寄存器进行读操作，将获得当前I/O口的状态。

图 8-3. 输出配置是 I/O 端口的输出配置。

图 8-3. 输出配置



8.3.6. 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；
- 端口输入状态寄存器相应位读取为“0”。

[图 8-4. 模拟配置](#)是 I/O 端口的模拟模式配置。

图 8-4. 模拟配置



8.3.7. 备用功能(AF)配置

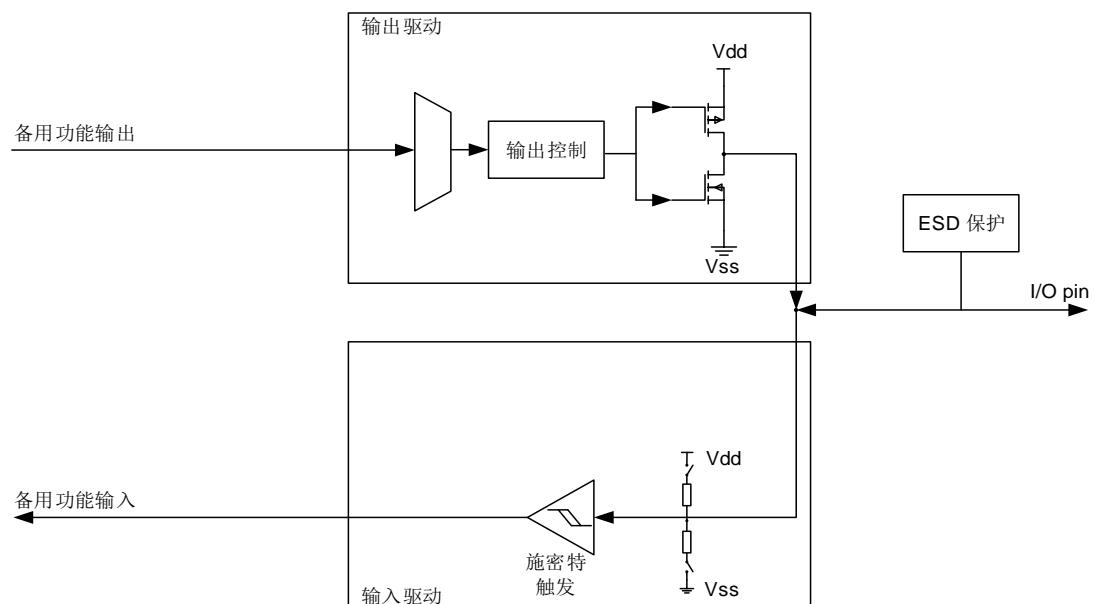
为了适应不同的器件封装，GPIO端口支持软件配置将一些备用功能应用到其他引脚上。

当引脚配置为备用功能时：

- 使用开漏或推挽功能时，可使能输出缓冲器；
- 输出缓冲器由外设驱动；
- 施密特触发输入使能；
- 在输入配置时，可选择弱上拉/下拉电阻；
- I/O引脚上的数据在每个APB2时钟周期采样并存入端口输入状态寄存器；
- 对端口输入状态寄存器进行读操作，将获得I/O口的状态；
- 对端口输出控制寄存器进行读操作，将返回上次写入的值。

[图 8-5. 备用功能配置](#)是 I/O 端口备用功能配置图。

图 8-5. 备用功能配置



8.3.8. 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_CTL0和GPIOx_CTL1。通过配置32位锁定寄存器(GPIOx_LOCK)可以锁定I/O端口的配置。通过特定的锁定序列配置GPIOx_LOCK中的LKK位和LKy位，相应的端口位被锁定，直到下一个复位前，相应端口位的配置都不能修改。建议在电源驱动模块的配置中使用锁定功能。

8.3.9. GPIO I/O 补偿单元

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端口噪声对工作电源的影响。

I/O补偿单元在系统复位后，默认是关闭状态，需要用户根据需要开启。

在使能I/O补偿单元后，将产生一个准备完成标志位CPS_RDY，用于指示补偿单元已经准备好，可以使用。如果电源电压超过2.4V~3.6V，不能使用I/O补偿单元，必须关闭该功能。

8.4. I/O 重映射功能和调试配置

8.4.1. 介绍

为了扩展GPIO的灵活性或外设功能使用，通过配置AFIO端口配置寄存器(AFIO_PCF0/AFIO_PCF1)，每个I/O引脚都可以配置多达4种不同的功能。通过使用外设IO的重映射功能可以选择合适的引脚位置。另外，通过配置相应的EXTI源选择寄存器(AFIO_EXTISSLx)选择触发中断或事件，GPIO引脚可以用作EXTI中断线。

8.4.2. 主要特征

- EXTI 源选择
- 每个引脚具有多达4种备用功能的配置

8.4.3. JTAG/SWD 备用功能重映射

调试接口信号映射在GPIO端口的情况如下表所示。

表 8-2. 调试接口信号

备用功能	GPIO 端口
JTMS / SWDIO	PA13
JTCK / SWCLK	PA14
JTDI	PA15
JTDO / TRACESWO	PB3
NJTRST	PB4
TRACECK	PE2
TRACECK0	PE3
TRACECK1	PE4

备用功能	GPIO 端口
TRACECK2	PE5
TRACECK3	PE6

为了减少用于调试的GPIO端口，用户可以配置AFIO_PCF0寄存器中的SWJ_CFG [2:0]位为不同的值。具体情况参照下表。

表 8-3. 调试端口映射⁽¹⁾

SWJ_CFG [2:0]	可用的调试端口	SWJ I/O 引脚分配				
		PA13/ JTMS/ SWDIO	PA14/ JTCK/S WCLK	PA15/ JTDI	PB3/ JTDO/ TRACE SWO	PB4/ NJTRST
000	完全 SWJ (JTAG-DP + SW-DP) (复位状态)	●	●	●	●	●
001	完全 SWJ (JTAG-DP + SW-DP) 但没有 NJTRST	●	●	●	●	X
010	JTAG-DP 关闭 SW-DP 开启	●	●	X	X ⁽¹⁾	X
100	JTAG-DP 关闭 SW-DP 关闭	X	X	X	X	X
其他	禁用					

- 只有在不使用异步跟踪时，I/O才能使用。

8.4.4. ADC AF 重映射

表 8-4. ADC0 注入转换外部触发备用功能重映射

备用功能	ADC0_ETRGINS_REMAP = 0	ADC0_ETRGINS_REMAP = 1
ADC0注入转换外部触发	ADC0注入转换外部触发与EXTI15相连	ADC0注入转换外部触发与TIMER7_CH3相连

表 8-5. ADC0 规则转换外部触发备用功能重映射

备用功能	ADC0_ETRGREG_REMAP = 0	ADC0_ETRGREG_REMAP = 1
ADC0规则转换外部触发	ADC0规则转换外部触发与EXTI11相连	ADC0 规则转换外部触发与TIMER7_TRGO 相连

表 8-6. ADC1 注入转换外部触发备用功能重映射

备用功能	ADC1_ETRGINS_REMAP = 0	ADC1_ETRGINS_REMAP = 1
ADC1注入转换外部触发	ADC1 注入转换外部触发与EXTI15 相连	ADC1 注入转换外部触发与TIMER7_CH3 相连

表 8-7. ADC1 规则转换外部触发备用功能重映射

备用功能	ADC1_ETRGREG_REMAP = 0	ADC1_ETRGREG_REMAP = 1
ADC1规则转换外部触发	ADC1 规则转换外部触发与EXTI11 相连	ADC1 规则转换外部触发与TIMER7_TRGO 相连

8.4.5. TIMER AF 重映射

表 8-8. TIMER0 备用功能重映射

备用功能	TIMER0_REMAP[1:0] =“00” (没有映射)	TIMER0_REMAP[1:0] =“01” (部分映射)	TIMER0_REMAP[1:0] =“11” (全映射) ⁽¹⁾
TIMER0_ETI	PA12		PE7
TIMER0_CH0	PA8		PE9
TIMER0_CH1	PA9		PE11
TIMER0_CH2	PA10		PE13
TIMER0_CH3	PA11		PE14
TIMER0_BKIN	PB12 ⁽²⁾	PA6	PE15
TIMER0_CH0_ON	PB13	PA7	PE8
TIMER0_CH1_ON	PB14 ⁽²⁾	PB0	PE10
TIMER0_CH2_ON	PB15 ⁽²⁾	PB1	PE12

1. 重映射仅适用于100引脚的封装。

2. 重映射在36引脚的封装下不可用

表 8-9. TIMER1 备用功能重映射

备用功能	TIMER1_REMAP [1:0] = “00” (没有映射)	TIMER1_REMAP [1:0] = “01” (部分映射)	TIMER1_REMAP [1:0] = “10” (部分映射)	TIMER1_REMAP [1:0] = “11” (全映射) ⁽¹⁾
TIMER1_CH0/TIMER 1 ETI ⁽²⁾	PA0	PA15	PA0	PA15
TIMER1_CH1	PA1	PB3	PA1	PB3
TIMER1_CH2	PA2		PB10	
TIMER1_CH3	PA3		PB11	

1. 重映射在 36 引脚的封装下不可用

2. TIMER1_CH0 和 TIMER1_ETI 共用一个引脚，但不能同时使用。

表 8-10. TIMER2 备用功能重映射⁽¹⁾

备用功能	TIMER2_REMAP[1:0] =“00” (没有映射)	TIMER2_REMAP[1:0] =“10” (部分映射)	TIMER2_REMAP[1:0] =“11” (全映射) ⁽¹⁾
TIMER2_CH0	PA6	PB4	PC6
TIMER2_CH1	PA7	PB5	PC7
TIMER2_CH2	PB0		PC8
TIMER2_CH3	PB1		PC9

1. 重映射仅仅适用于64引脚，100引脚的封装。

表 8-11. TIMER3 备用功能重映射⁽¹⁾

备用功能	TIMER3_REMAP = 0	TIMER3_REMAP = 1 ⁽¹⁾
TIMER3_CH0	PB6	PD12
TIMER3_CH1	PB7	PD13

TIMER3_CH2	PB8	PD14
TIMER3_CH3	PB9	PD15

1. 重映射仅仅适用于100引脚的封装。

表 8-12. TIMER4 备用功能重映射

备用功能	TIMER4CH3_REMAP = 0	TIMER4CH3_REMAP = 1
TIMER4_CH3	TIMER4_CH3 与 PA3 相连	IRC40K 内部时钟与 TIMER4_CH3 输入相连，用于 校正

表 8-13. TIMER8 备用功能重映射⁽¹⁾

备用功能	TIMER8_REMAP = 0	TIMER8_REMAP = 1
TIMER8_CH0	PA2	PE5
TIMER8_CH1	PA3	PE6

1. 参考备用功能映射和调试I/O配置寄存器1 (AFIO_PCF1)。

8.4.6. USART AF 重映射

参考AFIO端口配置寄存器0 (AFIO_PCF0)。

表 8-14. USART0 备用功能重映射

备用功能	USART0_REMAP = 0	USART0_REMAP = 1
USART0_TX	PA9	PB6
USART0_RX	PA10	PB7

表 8-15. USART1 备用功能重映射

备用功能	USART1_REMAP = 0	USART1_REMAP = 1⁽¹⁾
USART1_CTS	PA0	PD3
USART1_RTS	PA1	PD4
USART1_TX	PA2	PD5
USART1_RX	PA3	PD6
USART1_CK	PA4	PD7

1. 重映射仅仅适用于100引脚的封装。

表 8-16. USART2 备用功能重映射

备用功能	USART2_REMAP[1:0] =“00” (没有映射)	USART2_REMAP[1:0] =“01” (部分映射) ⁽¹⁾	USART2_REMAP[1:0] =“11” (全映射) ⁽²⁾
USART2_TX	PB10	PC10	PD8
USART2_RX	PB11	PC11	PD9
USART2_CK	PB12	PC12	PD10
USART2_CTS	PB13		PD11
USART2_RTS	PB14		PD12

1. 重映射仅仅适用于64引脚和100引脚的封装。

2. 重映射仅仅适用于100引脚的封装。

8.4.7. I2C0 备用功能重映射

参考 AFIO 端口配置寄存器 0 (AFIO_PCF0)。

表 8-17. I2C0 备用功能重映射

备用功能	I2C0_REMAP = 0	I2C0_REMAP = 1
I2C0_SCL	PB6	PB8
I2C0_SDA	PB7	PB9

8.4.8. SPI0 备用功能重映射

参考 AFIO 端口配置寄存器 0 (AFIO_PCF0)。

表 8-18. SPI0 备用功能重映射

备用功能	SPI0_REMAP = 0	SPI0_REMAP = 1
SPI0_NSS	PA4	PA15
SPI0_SCK	PA5	PB3
SPI0_MISO	PA6	PB4
SPI0_MOSI	PA7	PB5
SPI0_IO2	PA2	PB6
SPI0_IO3	PA3	PB7

8.4.9. SPI2/I2S2 备用功能重映射

参考AFIO端口配置寄存器0(AFIO_PCF0)。

表 8-19. SPI2/I2S2 备用功能重映射

备用功能	SPI2_REMAP = 0	SPI2_REMAP = 1
SPI2_NSS/ I2S2_WS	PA15	PA4
SPI2_SCK/ I2S2_CK	PB3	PC10
SPI2_MISO	PB4	PC11
SPI2_MOSI/I2S2_SD	PB5	PC12

8.4.10. CAN0 备用功能重映射

如下表所示，CAN0的信号引脚可以映射到端口A，端口B或端口D。对于端口D，重映射不适用36引脚、48引脚、64引脚的封装中。

表 8-20. CAN0 备用功能重映射

备用功能 ⁽¹⁾	CAN0_REMAP[1:0] =“00”	CAN0_REMAP[1:0] =“10” ⁽²⁾	CAN0_REMAP[1:0] =“11” ⁽³⁾
CAN0_RX	PA11	PB8	PD0

CAN0_TX	PA12	PB9	PD1
---------	------	-----	-----

1. 重映射在36引脚封装下不可用
2. 当PD0和PD1不重映射到OSC_IN和OSC_OUT时，该重映射仅适用于100引脚的封装。

8.4.11. CAN1 备用功能重映射

表 8-21. CAN1 备用功能重映射

备用功能	CAN_REMAP =“0”	CAN_REMAP =“1”
CAN1_RX	PB12	PB5
CAN1_TX	PB13	PB6

8.4.12. CTC 备用功能重映射

参考AFIO端口配置寄存器1(AFIO_PCF1)。

表 8-22. CTC 备用功能重映射

备用功能	CTC_REMAP [1:0] = “00”	CTC_REMAP [1:0] = “01”
CTC_SYNC	PA8	PD15

8.4.13. CLK 引脚 AF 重映射

当LXTAL关闭的时候，OSC32_IN和OSC32_OUT分别可以用做普通的I/O端口PC14和PC15。LXTAL的优先级比其他普通IO功能高。

- 注意：
1. 当1.8V区域关掉（进入待机模式）或备份区域由VBAT供电（不使用VDD供电），PC14/PC15 不能用于普通IO功能，将会被设置为模拟模式。
 2. 参考3.3.1电池备份域章节中的IO口用法。

表 8-23. OSC32 引脚配置

备用功能	LXTAL= ON	LXTAL= OFF
PC14	OSC32_IN	PC14
PC15	OSC32_OUT	PC15

HXTAL晶振引脚OSC_IN/OSC_OUT可以用做普通的I/O端口PD0/PD1。对于36、48和64脚的封装，PD0和PD1不能用来产生外部中断/事件。

表 8-24. OSC 引脚配置

备用功能	HXTAL= ON	HXTAL= OFF
PD0	OSC_IN	PD0
PD1	OSC_OUT	PD1

8.5. GPIO 寄存器

GPIOA 基地址: 0x4001 0800

GPIOB 基地址: 0x4001 0C00

GPIOC 基地址: 0x4001 1000

GPIOD 基地址: 0x4001 1400

GPIOE 基地址: 0x4001 1800

AFIO 基地址: 0x4001 0000

8.5.1. 端口控制寄存器 0 (GPIOx_CTL0, x=A..E)

地址偏移: 0x00

复位值: 0x4444 4444

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
CTL7[1:0]	MD7[1:0]	CTL6[1:0]	MD6[1:0]	CTL5[1:0]	MD5[1:0]	CTL4[1:0]	MD4[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
CTL3[1:0]	MD3[1:0]	CTL2[1:0]	MD2[1:0]	CTL1[1:0]	MD1[1:0]	CTL0[1:0]	MD0[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	CTL7[1:0]	Port 7 配置位 该位由软件置位和清除。 参考CTL0[1:0]的描述
29:28	MD7[1:0]	Port 7 模式位 该位由软件置位和清除。 参考MD0 [1:0]的描述
27:26	CTL6[1:0]	Port 6 配置位 该位由软件置位和清除。 参考CTL0[1:0]的描述
25:24	MD6[1:0]	Port 6 模式位 该位由软件置位和清除。 参考MD0 [1:0]的描述
23:22	CTL5[1:0]	Port 5 配置位 该位由软件置位和清除。 参考CTL0[1:0]的描述
21:20	MD5[1:0]	Port 5 模式位 该位由软件置位和清除。

参考MD0 [1:0]的描述

19:18	CTL4[1:0]	Port 4配置位 该位由软件置位和清除。 参考CTL0[1:0]的描述
17:16	MD4[1:0]	Port 4模式位 该位由软件置位和清除。 参考MD0 [1:0]的描述
15:14	CTL3[1:0]	Port 3配置位 该位由软件置位和清除。 参考CTL0[1:0]的描述
13:12	MD3[1:0]	Port 3模式位 该位由软件置位和清除。 参考MD0 [1:0]的描述
11:10	CTL2[1:0]	Port 2配置位 该位由软件置位和清除。 参考CTL0[1:0]的描述
9:8	MD2[1:0]	Port 2模式位 该位由软件置位和清除。 参考MD0 [1:0]的描述
7:6	CTL1[1:0]	Port 1配置位 该位由软件置位和清除。 参考CTL0[1:0]的描述
5:4	MD1[1:0]	Port 1模式位 该位由软件置位和清除。 参考MD0 [1:0]的描述
3:2	CTL0[1:0]	Pin 0配置位 该位由软件置位和清除。 输入模式 (MD[1:0] =00) 00: 模拟输入 01: 浮空输入 10: 上拉输入 /下拉输入 11: 保留 输出模式 (MD[1:0] >00) 00: GPIO 推挽输出 01: GPIO 开漏输出 10: AFIO 推挽输出 11: AFIO 开漏输出
1:0	MD0[1:0]	Port 0模式位

该位由软件置位和清除。

- 00: 输入模式 (复位状态)
- 01: 输出模式, 最大速度10MHz
- 10: 输出模式, 最大速度2 MHz
- 11: 输出模式, 最大速度50MHz

8.5.2. 端口控制寄存器 1 (GPIOx_CTL1, x=A..E)

地址偏移: 0x04

复位值: 0x4444 4444

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
CTL15[1:0]	MD15[1:0]	CTL14[1:0]	MD14[1:0]	CTL13[1:0]	MD13[1:0]	CTL12[1:0]	MD12[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
CTL11[1:0]	MD11[1:0]	CTL10[1:0]	MD10[1:0]	CTL9[1:0]	MD9[1:0]	CTL8[1:0]	MD8[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]	Port 15配置位 该位由软件置位和清除。 参考CTL0[1:0]的描述
29:28	MD15[1:0]	Port 15模式位 该位由软件置位和清除。 参考MD0 [1:0]的描述
27:26	CTL14[1:0]	Port 14配置位 该位由软件置位和清除。 参考CTL0[1:0]的描述
25:24	MD14[1:0]	Port 14模式位 该位由软件置位和清除。 参考MD0 [1:0]的描述
23:22	CTL13[1:0]	Port13配置位 该位由软件置位和清除。 参考CTL0[1:0]的描述
21:20	MD13[1:0]	Port 13模式位 该位由软件置位和清除。 参考MD0 [1:0]的描述

19:18	CTL12[1:0]	Port 12配置位 该位由软件置位和清除。 参考CTL0[1:0]的描述
17:16	MD12[1:0]	Port 12模式位 该位由软件置位和清除。 参考MD0 [1:0]的描述
15:14	CTL11[1:0]	Port 11配置位 该位由软件置位和清除。 参考CTL0[1:0]的描述
13:12	MD11[1:0]	Port 11模式位 该位由软件置位和清除。 参考MD0 [1:0]的描述
11:10	CTL10[1:0]	Port 10配置位 该位由软件置位和清除。 参考CTL0[1:0]的描述
9:8	MD10[1:0]	Port10模式位 该位由软件置位和清除。 参考MD0 [1:0]的描述
7:6	CTL9[1:0]	Port 9配置位 该位由软件置位和清除。 参考CTL0[1:0]的描述
5:4	MD9[1:0]	Port 9模式位 该位由软件置位和清除。 参考MD0 [1:0]的描述
3:2	CTL8[1:0]	Pin8配置位 该位由软件置位和清除。 参考CTL0[1:0]的描述
1:0	MD8[1:0]	Port 8模式位 该位由软件置位和清除。 参考MD0 [1:0]的描述

8.5.3. 端口输入状态寄存器 (**GPIOx_ISTAT, x=A..E**)

地址偏移: 0x08

复位值: 0x0000 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
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

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	ISTATy	<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>

8.5.4. 端口输出控制寄存器 (GPIOx_OCTL, x=A..E)

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
OCTL15	OCTL14	OCTL13	OCTL12	OCTL11	OCTL10	OCTL9	OCTL8	OCTL7	OCTL6	OCTL5	OCTL4	OCTL3	OCTL2	OCTL1	OCTL0

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

8.5.5. 端口位操作寄存器 (GPIOx_BOP, x=A..E)

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
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) 这些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	端口置位位y(y=0..15) 这些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

8.5.6. 位清除寄存器 (**GPIOx_BC**, x=A..E)

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
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

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) 这些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

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

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															

位/位域	名称	描述
31:17	保留	必须保持复位值
16	LKK	<p>锁定序列键 该位只能通过使用Lock Key写序列设置，始终可读。</p> <p>0: GPIO_LOCK寄存器和端口配置没有锁定 1: 直到下一次MCU复位前，GPIO_LOCK寄存器被锁定 LOCK Key写序列： 写1→写0→写1→ 读0→ 读1 注意：在LOCK Key写序列期间，LK[15:0]的值必须保持。</p>
15:0	LKy	<p>端口锁定位y(y=0..15) 这些位由软件置位和清除。 0: 相应的端口位配置没有锁定 1: 当LKK位置1时，相应的端口位配置被锁定</p>

8.5.8. 端口位速度寄存器 (GPIOx_SPD, x=A..E)

地址偏移: 0x3C

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	SPDy	<p>当MDx值为0b11时，设置相应端口速度为高速（120MHz）。</p> <p>如果端口输出速度大于50MHz，该位置1，同时设置MDx值为0b11。这些位由软件置位和清除。</p> <p>0: 没有影响</p> <p>1: 最大输出速度大于50MHz（同时，需要设置MDx值为0b11）</p> <p>注意：当端口输出速度大于50MHz时，需要使能I/O补偿单元。详见AFIO_CPSCTL寄存器中的CPS_EN位说明。</p>

8.5.9. 事件控制寄存器 (AFIO_EC)

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								EOE	PORT[2:0]	PIN[3:0]					
								rw	rw	rw					

位/位域	名称	描述
31:8	保留	必须保持复位值
7	EOE	事件输出使能 该位由软件置位和清除。当设置该位后, Cortex 的EVENTOUT输出将连接到由PORT[2:0]和PIN[3:0]位选择的I/O口。
6:4	PORT[2:0]	事件输出端口选择 这些位由软件置位和清除。选择用于输出Cortex的EVENTOUT信号的端口。 000: 选择端口A 001: 选择端口B 010: 选择端口C 011: 选择端口D 100: 选择端口E
3:0	PIN[3:0]	事件输出引脚选择 这些位由软件置位和清除。选择用于输出Cortex的EVENTOUT信号的引脚。 0000: 选择引脚0 0001: 选择引脚1 0010: 选择引脚2 ... 1111: 选择引脚15

8.5.10. AFIO 端口配置寄存器 0 (AFIO_PCF0)

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	TIMER1ITR0 _REMAP	SPI2_ REMAP	保留	SWJ_CFG[2:0]	保留	CAN1_ REMAP	保留	ADC1_ ETRGREG _REMAP	ADC1_ ETRGINS _REMAP	ADC0_ ETRGREG _REMAP	ADC0_ ETRGINS _REMAP	TIMER4CH3 _IREMAP			
	RW	RW		W		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
PD01_ REMAP	CANO_REMAP [1:0]	TIMER3_ REMAP	TIMER2_REMAP [1:0]	TIMER1_REMAP [1:0]	TIMER0_REMAP [1:0]	USART2_REMAP[1:0]	USART1_ REMAP	USART0_ REMAP	I2C0_ REMAP	SPI0_ REMAP					
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	名称	描述
31:30	保留	必须保持复位值
29	TIMER1ITI1_REMAP	TIMER1内部触发1重映射 该位由软件置位和清除，用于控制TIMER1_ITI1内部重映射。 0: 没有重映射 1: TIMER1_ITI1 内部连接到USBFS SOF (起始帧) 输出，用于校准
28	SPI2_REMAP	SPI2/I2S2重映射 该位由软件置位和清除。 0: 没有重映射 (SPI2_NSS-I2S2_WS/PA15, SPI2_SCK-I2S2_CK/PB3, SPI2_MISO/PB4, SPI2_MOSI-I2S_SD/PB5) 1: 完全重映射 (SPI2_NSS-I2S2_WS/PA4, SPI2_SCK-I2S2_CK/PC10, SPI2_MISO/PC11, SPI2_MOSI-I2S_SD/PC12)
27	保留	必须保持复位值
26:24	SWJ_CFG[2:0]	串行线JTAG配置 这些位只写（读这些位，将返回未定义值）。用于配置SWJ和跟踪复用功能的I/O口。SWJ（串行线JTAG）支持JTAG或SWD访问Cortex调试端口。系统复位后的默认状态是启用SWJ但没有跟踪功能，这种状态下，可以通过在JTMS/JTCK引脚上的发送特定的信号使能JTAG或SW（串行线）模式。 000: 完全SWJ(JTAG-DP +SW-DP): 复位状态 001: 完全SWJ(JTAG-DP +SW-DP): 没有NJTRST 010: JTAG-DP禁用和SW-DP使能 100: JTAG-DP禁用和SW-DP禁用 其他组合: 无作用 注意: 每次只能设置三个比特中的一个。
23	保留	必须保持复位值
22	CAN1_REMAP	CAN1 I/O重映射 该位由软件置位和清除，控制着CAN1_RX和CAN1_TX引脚。 0: 没有重映射 (CAN1_RX/PB12,CAN1_TX/PB13) 1: 重映射 (CAN1_RX/PB5, CAN1_TX/PB6)
21	保留	必须保持复位值
20	ADC1_ETRGREG	ADC 1规则转换外部触发重映射

	<u>_REMAP</u>	该位由软件置位和清除。该位控制着触发输入与ADC1规则转换外部触发连接。当该位复位时，ADC1规则转换外部触发与EXTI11相连。当该位置位时，ADC1规则转换外部触发与TIMER7_TRGO相连。
19	ADC1_ETRGINS	ADC 1注入转换外部触发重映射
	<u>_REMAP</u>	该位由软件置位和清除。该位控制着触发输入与ADC1注入转换外部触发连接。当该位复位时，ADC1规则转换外部触发与EXTI15相连。当该位置位时，ADC1注入转换外部触发与TIMER7_CH3相连。
18	ADC0_ETRGREG_R	ADC 0规则转换外部触发重映射
	<u>EMAP</u>	该位由软件置位和清除。该位控制着触发输入与ADC0规则转换外部触发连接。当该位复位时，ADC0规则转换外部触发与EXTI11相连。当该位置位时，ADC0规则转换外部触发与TIMER7_TRGO相连。
17	ADC0_ETRGINS	ADC 0注入转换外部触发重映射
	<u>_REMAP</u>	该位由软件置位和清除。该位控制着触发输入与ADC0注入转换外部触发连接。当该位复位时，ADC0规则转换外部触发与EXTI15相连。当该位置位时，ADC0注入转换外部触发与TIMER7_CH3相连。
16	TIMER4CH3_IREMA	TIMER4通道3内部重映射
	P	该位由软件置位和清除，控制着TIMER4_CH3的内部重映射。当该位复位时TIMER4_CH3与PA3相连。当该位置位时，IRC40K内部振荡器与TIMER4_CH3相连，用于对IRC40K校准。 0: 没有重映射 1: 重映射
15	PD01_REMAP	Port D0/Port D1重映射到OSC_IN/OSC_OUT
		该位由软件置位和清除。 0: 没有重映射 1: PD0重映射到OSC_IN, PD1重映射到OSC_OUT
14:13	CAN0_REMAP[1:0]	CAN0备用功能接口重映射 这些位由软件置位和清除。 00: 没有重映射(CAN0_RX/PA11, CAN0_TX/PA12) 01: 没有使用 10: 部分重映射(CAN0_RX/PB8, CAN0_TX/PB9) 11: 完全重映射(CAN0_RX/PD0, CAN0_TX/PD1)
12	TIMER3_REMAP	TIMER3 重映射 该位由软件置位和清除。 0: 没有重映射 (TIMER3_CH0/PB6, TIMER3_CH1/PB7, TIMER3_CH2/PB8, TIMER3_CH3/PB9) 1: 完全重映射(TIMER3_CH0/PD12, TIMER3_CH1/PD13, TIMER3_CH2/PD14, TIMER3_CH3/PD15)
11:10	TIMER2_REMAP [1:0]	TIMER2 重映射 这些位由软件置位和清除。 00: 没有重映射(TIMER2_CH0/PA6, TIMER2_CH1/PA7, TIMER2_CH2/PB0, TIMER2_CH3/PB1)

		TIMER2_CH3/PB1)
		01: 没有使用
		10: 部分重映射(TIMER2_CH0/PB4, TIMER2_CH1/PB5, TIMER2_CH2/PB0, TIMER2_CH3/PB1)
		11: 完全重映射(TIMER2_CH0/PC6, TIMER2_CH1/PC7, TIMER2_CH2/PC8, TIMER2_CH3/PC9)
9:8	TIMER1_REMAP	TIMER1 重映射
	[1:0]	这些位由软件置位和清除。
		00: 没有重映射 (TIMER1_CH0-TIMER1_ETI/PA0, TIMER1_CH1/PA1, TIMER1_CH2/PA2, TIMER1_CH3/PA3)
		01: 部分重映射 0 (TIMER1_CH0-TIMER1_ETI/PA15, TIMER1_CH1/PB3, TIMER1_CH2/PA2, TIMER1_CH3/PA3)
		10: 部分重映射 1 (TIMER1_CH0-TIMER1_ETI/PA0, TIMER1_CH1/PA1, TIMER1_CH2/PB10, TIMER1_CH3/PB11)
		11: 完全重映射 (TIMER1_CH0-TIMER1_ETI/PA15, TIMER1_CH1/PB3, TIMER1_CH2/PB10, TIMER1_CH3/PB11)
7:6	TIMER0_REMAP	TIMER0 重映射
	[1:0]	这些位由软件置位和清除。
		00: 没有重映射(TIMER0_ETI/PA12, TIMER0_CH0/ PA8, TIMER0_CH1/PA9, TIMER0_CH2/PA10, TIMER0_CH3/PA11, TIMER0_BKIN/PB12, TIMER0_CH0_ON/PB13, TIMER0_CH1_ON/PB14, TIMER0_CH2_ON/PB15)
		01: 部分重映射 (TIMER0_ETI/PA12, TIMER0_CH0/ PA8, TIMER0_CH1/PA9, TIMER0_CH2/PA10, TIMER0_CH3/PA11, TIMER0_BKIN/PA6, TIMER0_CH0_ON/PA7, TIMER0_CH1_ON/PB0, TIMER0_CH2_ON/PB1)
		10: 没有使用
		11: 完全重映射 (TIMER0_ETI/PE7, TIMER0_CH0/ PE9, TIMER0_CH1/PE11, TIMER0_CH2/PE13, TIMER0_CH3/PE14, TIMER0_BKIN/PE15, TIMER0_CH0_ON/PE8, TIMER0_CH1_ON/PE10, TIMER0_CH2_ON/PE12)
5:4	USART2_REMAP	USART2 重映射
	[1:0]	这些位由软件置位和清除。
		00: 没有重映射(USART2_TX/PB10, USART2_RX /PB11, USART2_CK/PB12, USART2_CTS/PB13, USART2_RTS/PB14)
		01: 部分重映射 (USART2_TX/PC10, USART2_RX /PC11, USART2_CK/PC12, USART2_CTS/PB13, USART2_RTS/PB14)
		10: 没有使用
		11: 完全重映射 (USART2_TX/PD8, USART2_RX /PD9, USART2_CK/PD10, USART2_CTS/PD11, USART2_RTS/PD12)
3	USART1_REMAP	USART1 重映射
		该位由软件置位和清除。
		0: 没有重映射 (USART1_CTS/PA0, USART1_RTS/PA1, USART1_TX/PA2, USART1_RX /PA3, USART1_CK/PA4)

1: 重映射 (USART1_CTS/PD3, USART1_RTS/PD4, USART1_TX/PD5,
USART1_RX /PD6, USART1_CK/PD7)

2	USART0_REMAP	USART0 重映射
该位由软件置位和清除。		
0:	没有重映射 (USART0_TX/PA9, USART0_RX /PA10)	
1:	重映射 (USART0_TX/PB6, USART0_RX /PB7)	
1	I2C0_REMAP	I2C0 重映射
该位由软件置位和清除。		
0:	没有重映射 (I2C0_SCL/PB6, I2C0_SDA /PB7)	
1:	重映射 (I2C0_SCL/PB8, I2C0_SDA /PB9)	
0	SPI0_REMAP	SPI0 重映射
该位由软件置位和清除。		
0:	没有重映射 (SPI0_NSS/PA4, SPI0_SCK /PA5, SPI0_MISO /PA6, SPI0_MOSI /PA7, SPI0_IO2 /PA2, SPI0_IO3 /PA3)	
1:	重映射 (SPI0_NSS/PA15, SPI0_SCK /PB3, SPI0_MISO /PB4, SPI0_MOSI /PB5, SPI0_IO2 /PB6, SPI0_IO3 /PB7)	

8.5.11. EXTI 源选择寄存器 0 寄存器 (AFIO_EXTI0_SS0)

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 0000: PA3 引脚 0001: PB3 引脚 0010: PC3 引脚 0011: PD3 引脚 0100: PE3 引脚 其他配置保留。
11:8	EXTI2_SS[3:0]	EXTI 2 源选择

0000: PA2 引脚
 0001: PB2 引脚
 0010: PC2 引脚
 0011: PD2 引脚
 0100: PE2 引脚
 其他配置保留。

7:4 EXTI1_SS[3:0] EXTI 1 源选择
 0000: PA1 引脚
 0001: PB1 引脚
 0010: PC1 引脚
 0011: PD1 引脚
 0100: PE1 引脚
 其他配置保留。

3:0 EXTI0_SS[3:0] EXTI 0 源选择
 0000: PA0 引脚
 0001: PB0 引脚
 0010: PC0 引脚
 0011: PD0 引脚
 0100: PE0 引脚
 其他配置保留。

8.5.12. EXTI 源选择寄存器 1 寄存器 (AFIO_EXTI_SS1)

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		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 0000: PA7 引脚 0001: PB7 引脚 0010: PC7 引脚

		0011: PD7 引脚
		0100: PE7 引脚
其他配置保留。		
11:8	EXTI6_SS[3:0]	EXTI 6 源选择
		0000: PA6 引脚
		0001: PB6 引脚
		0010: PC6 引脚
		0011: PD6 引脚
		0100: PE6 引脚
其他配置保留。		
7:4	EXTI5_SS[3:0]	EXTI 5 源选择
		0000: PA5 引脚
		0001: PB5 引脚
		0010: PC5 引脚
		0011: PD5 引脚
		0100: PE5 引脚
其他配置保留。		
3:0	EXTI4_SS[3:0]	EXTI 4 源选择
		0000: PA4 引脚
		0001: PB4 引脚
		0010: PC4 引脚
		0011: PD4 引脚
		0100: PE4 引脚
其他配置保留。		

8.5.13. EXTI 源选择寄存器 2 寄存器 (AFIO_EXTI_SS2)

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI11_SS [3:0]				EXTI10_SS [3:0]				EXTI9_SS [3:0]				EXTI8_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	EXTI11_SS[3:0]	EXTI 11 源选择 0000: PA11 引脚 0001: PB11 引脚 0010: PC11 引脚 0011: PD11 引脚 0100: PE11 引脚 其他配置保留。
11:8	EXTI10_SS[3:0]	EXTI 10 源选择 0000: PA10 引脚 0001: PB10 引脚 0010: PC10 引脚 0011: PD10 引脚 0100: PE10 引脚 其他配置保留。
7:4	EXTI9_SS[3:0]	EXTI 9 源选择 0000: PA9 引脚 0001: PB9 引脚 0010: PC9 引脚 0011: PD9 引脚 0100: PE9 引脚 其他配置保留。
3:0	EXTI8_SS[3:0]	EXTI 8 源选择 0000: PA8 引脚 0001: PB8 引脚 0010: PC8 引脚 0011: PD8 引脚 0100: PE8 引脚 其他配置保留。

8.5.14. EXTI 源选择寄存器 3 寄存器 (AFIO_EXTI_SS3)

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 0000: PA15 引脚 0001: PB15 引脚 0010: PC15 引脚 0011: PD15 引脚 0100: PE15 引脚 其他配置保留。
11:8	EXTI14_SS[3:0]	EXTI 14 源选择 0000: PA14 引脚 0001: PB14 引脚 0010: PC14 引脚 0011: PD14 引脚 0100: PE14 引脚 其他配置保留。
7:4	EXTI13_SS[3:0]	EXTI 13 源选择 0000: PA13 引脚 0001: PB13 引脚 0010: PC13 引脚 0011: PD13 引脚 0100: PE13 引脚 其他配置保留。
3:0	EXTI12_SS[3:0]	EXTI 12 源选择 0000: PA12 引脚 0001: PB12 引脚 0010: PC12 引脚 0011: PD12 引脚 0100: PE12 引脚 其他配置保留。

8.5.15. AFIO 端口配置寄存器 1 (AFIO_PCF1)

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
保留		CTC_REMAP[1:0]		EXMC_NA DV		保留		TIMER8_ REMAP		保留					

rw rw rw

位/位域	名称	描述
31:13	保留	必须保持复位值
12:11	CTC_REMAP [1:0]	CTC重映射 这些位由软件置位和清除，控制着将CTC_SYNC备用功能重映射到GPIO端口。 00: 没有重映射 (PA8) 01: 重映射0 (PD15) 10/11: 保留
10	EXMC_NADV	EXMC_NADV 连接/不连接 该位由软件置位和清除，控制着可选的EXMC_NADV信号 0: NADV信号连接到输出(默认值) 1: NADV信号没有连接，I/O引脚可以用于其他外设。
9:6	保留	必须保持复位值
5	TIMER8_REMAP	TIMER8 重映射 该位由软件置位和清除，控制着将TIMER8_CH0和TIMER8_CH1备用功能重映射到GPIO端口。 0: 没有重映射 (TIMER8_CH0连接到PA2和TIMER8_CH1连接到 PA3) 1: 重映射(TIMER8_CH0重映射到PE5和TIMER8_CH1重映射到PE6)
4:0	保留	必须保持复位值

8.5.16. IO 补偿控制寄存器 (AFIO_CPSCTL)

地址偏移: 0x20

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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

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

r rw

位/位域	名称	描述
31:9	保留	必须保持复位值
8	CPS_RDY	I/O补偿单元是否准备好，该位只读。

0: I/O补偿单元没有准备好

1: I/O补偿单元准备好

7:1 保留 必须保持复位值

0 CPS_EN I/O补偿单元使能

当端口输出速度大于50MHz时，需要使能I/O补偿单元。

0: I/O补偿单元掉电

1: I/O补偿单元使能

9. 循环冗余校验计算单元 (CRC)

9.1. 简介

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

CRC 计算单元使用固定多项式计算 32 位 CRC 校验码。

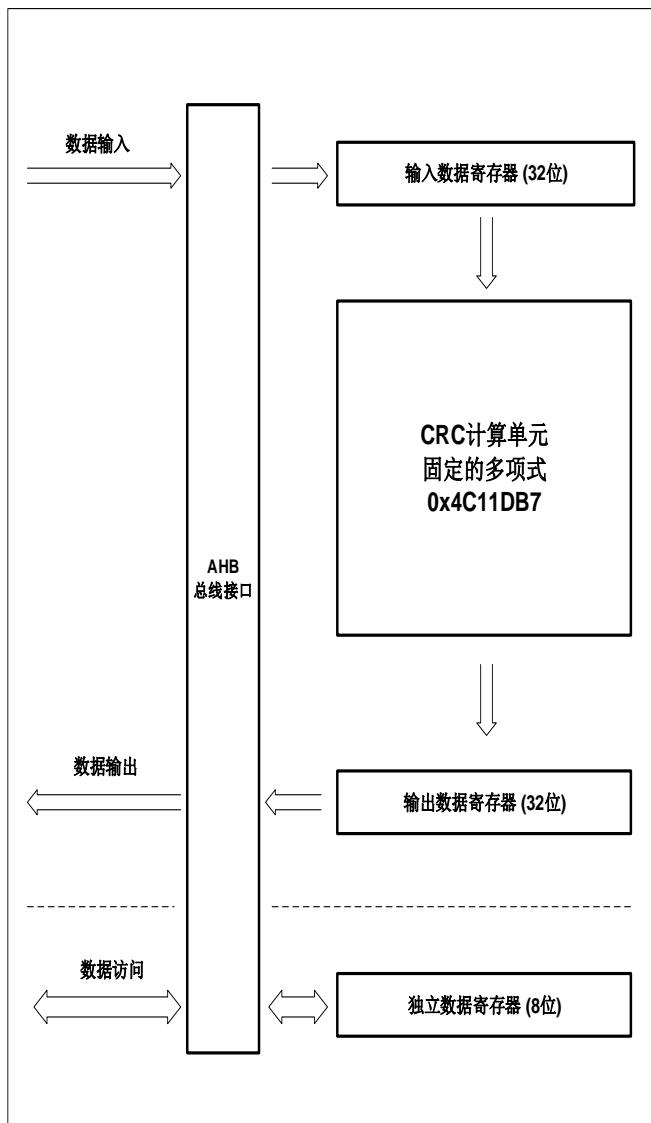
9.2. 主要特征

- 32位数据输入/输出寄存器。对于32位的输入数据，从数据输入到得出计算结果，需要4个AHB的时钟周期；
- 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；
- 固定的计算多项式：0x4C11DB7；

$$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$$

该 32 位 CRC 多项式与以太网 CRC 计算多项式相同。

图 9-1. CRC 计算单元框图



9.3. 功能说明

- CRC计算单元可以用来计算32位的原始数据，**CRC_DATA**寄存器接收原始数据并存储计算结果。

如果不通过软件设置**CRC_CTL**寄存器的方式来清除**CRC_DATA**寄存器，CRC计算单元将基于新输入的原始数据和前一次**CRC_DATA**寄存器中的结果进行计算。

对于32位的数据，CRC的计算需要4个AHB的时钟周期。在此期间，因为32位输入缓存的存在，AHB总线将不会被挂起。

- 此模块提供了一个8位的独立数据寄存器**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。

9.4. CRC 寄存器

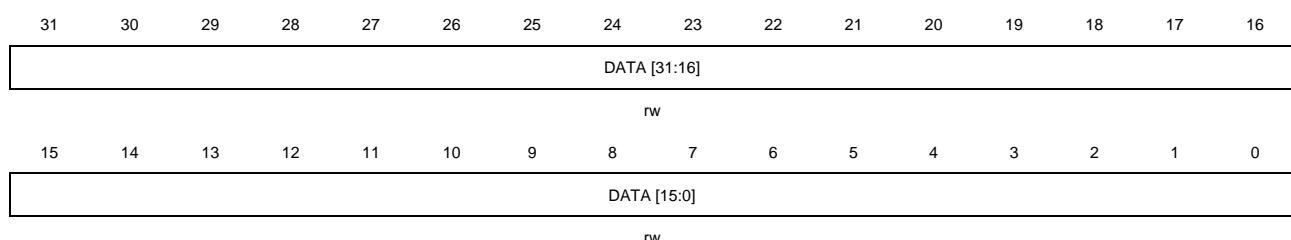
CRC基地址: 0x4002 3000

9.4.1. 数据寄存器 (CRC_DATA)

地址偏移: 0x00

复位值: 0xFFFF FFFF

该寄存器只能按字 (32位) 访问



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

9.4.2. 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	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。

9.4.3. 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

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															RST

rs

位/位域	名称	描述
31:1	保留	保持复位值
0	RST	将该位置1可以复位CRC_DATA寄存器，并设置其值为0xFFFFFFFF，然后该位被硬件自动清零。该位对CRC_FDATA寄存器没有影响。 软件可读可写。

10. 直接存储器访问控制器（DMA）

10.1. 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，从而释放带宽，使 CPU 可以专注在处理其他系统功能上。数据可以通过 DMA 在外设和内存以及内存与内存之间快速移动，而无需任何 CPU 操作。DMA 控制器有 12 个通道（DMA0 有 7 个通道，DMA1 有 5 个通道）。每个通道都是专门用来处理一个或多个外设的存储器访问请求的。DMA 控制器内部实现了一个仲裁器，用来仲裁多个 DMA 请求的优先级。

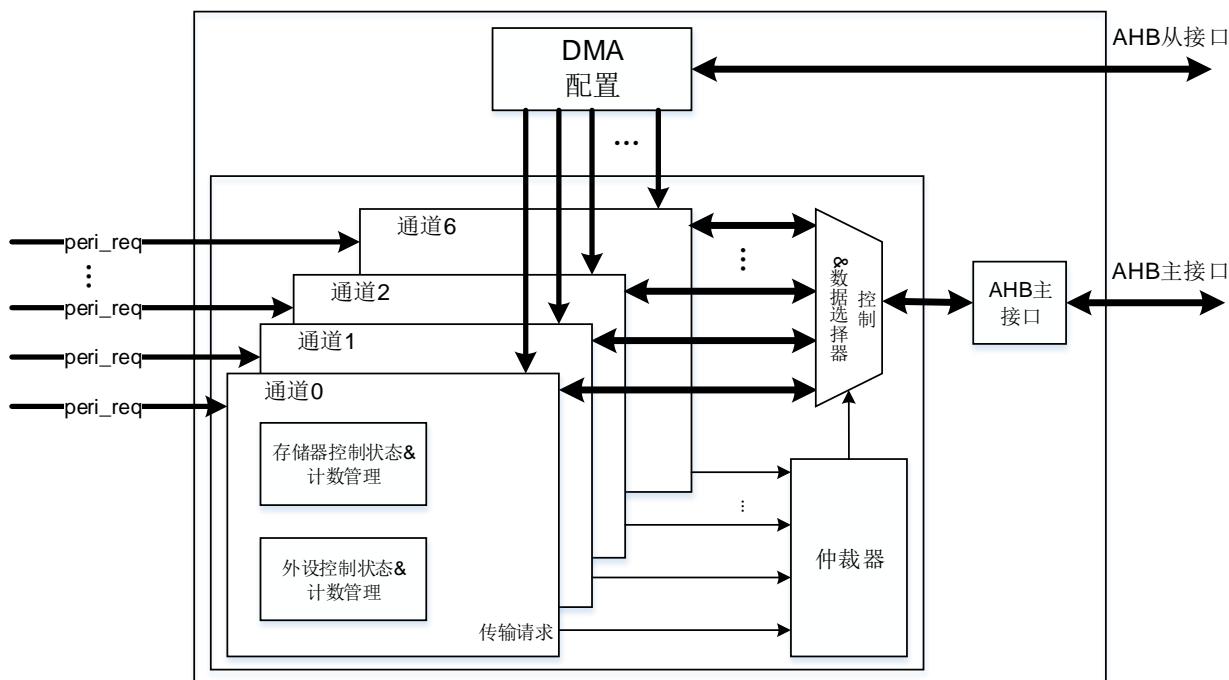
DMA 控制器和 Cortex®-M4 内核共享系统总线。当 DMA 和 CPU 访问同样的地址空间时，DMA 访问可能会阻挡 CPU 访问系统总线几个总线周期。总线矩阵中实现了循环仲裁算法来分配 DMA 与 CPU 的访问权，它可以确保 CPU 得到至少一半的系统总线带宽。

10.2. 主要特征

- 传输数据长度可编程配置，最大到 65536；
- 12 个通道，并且每个通道都可配置（DMA0 有 7 个通道，DMA1 有 5 个通道）；
- 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。

10.3. 结构框图

图 10-1. DMA 结构框图



由 [图 10-1. DMA 结构框图](#) 所示，DMA 控制器由 4 部分组成：

- AHB 从接口配置 DMA
- AHB 主接口进行数据传输，用于存储器访问和外设访问
- 仲裁器 DMA 请求的优先级管理
- 通道管理用于控制数据/地址选择和数据计数

10.4. 功能说明

10.4.1. 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 10-1. DMA 传输操作](#)。

表 10-1. 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	1: Write B3B2B1B0[31:0] @0x0

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

DMA_CHxCNT寄存器的CNT位域必须在CHEN位置位前被配置，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，对任意配置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。

10.4.2. 外设握手

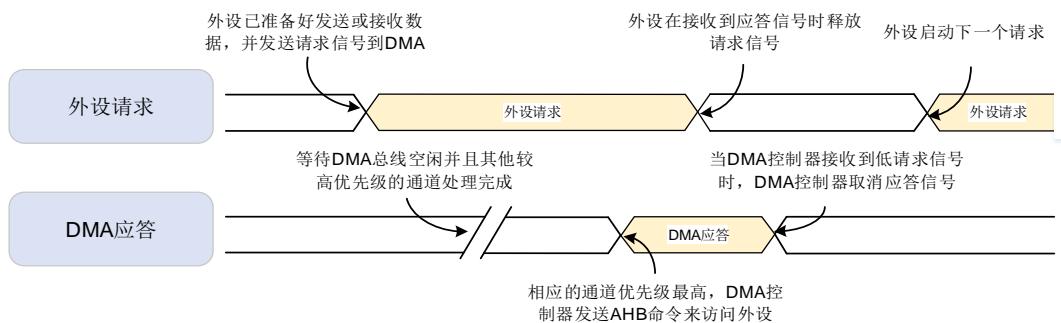
为了保证数据的有效传输，DMA控制器中引入了外设和存储器的握手机制，包括请求信号和应答信号：

- 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10-2. 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 10-2. 握手机制



10.4.3. 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。

10.4.4. 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。

10.4.5. 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。

10.4.6. 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。

10.4.7. 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，判断通道是否使能。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，启动新的传输。
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。

10.4.8. 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。

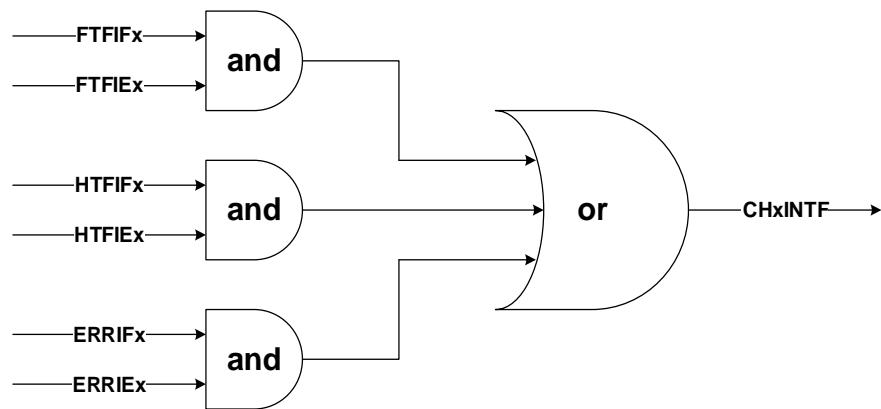
每一个中断事件在DMA_INTF寄存器中有专用的标志位，在DMA_INTC寄存器中有专用的清除位，在DMA_CHxCTL寄存器中有专用的使能位。[表10-2. 中断事件](#)描述了其对应关系。

表 10-2. 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（Full transfer finish）	FTFIF	FTFIFC	FTFIE
半传输完成（Half transfer finish）	HTFIF	HTFIFC	HTFIE
传输错误（Transfer error）	ERRIF	ERRIFC	ERRIE

DMA中断逻辑如[图10-3. DMA中断逻辑图](#)所示，任何类型中断使能时，产生了相应中断事件均会产生中断。

图 10-3. DMA 中断逻辑图



注意：“x”表示通道数（DMA0对应x=0...6； DMA1对应x=0...4）

10.4.9. 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 10-4. DMA0 请求映射](#) 和 [图 10-5. DMA1 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 10-3. DMA0 各通道请求表](#)列举了 DMA0 的每个通道所支持的外设请求，[表 10-4. DMA1 各通道请求表](#)列举了 DMA1 的每个通道所支持的外设请求。

图 10-4. DMA0 请求映射

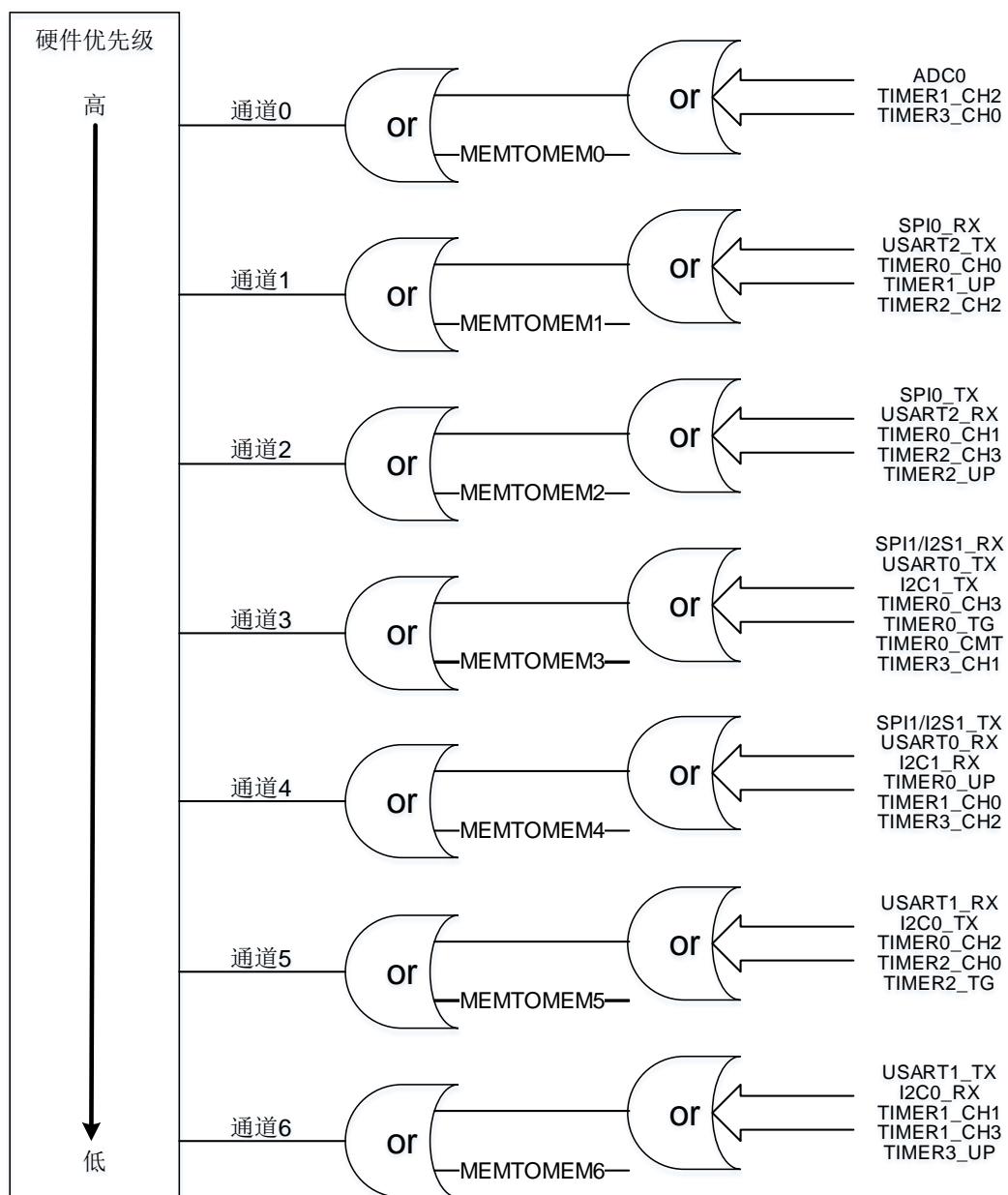


表10-3. DMA0各通道请求表

外设	通道0	通道1	通道2	通道3	通道4	通道5	通道6
TIMER0	•	TIMER0_CH0	TIMER0_CH1	TIMER0_CH3 TIMER0_TG TIMER0_CMT	TIMER0_UP	TIMER0_CH2	•
TIMER1	TIMER1_CH2	TIMER1_UP	•	•	TIMER1_CH0	•	TIMER1_CH1 TIMER1_CH3
TIMER2	•	TIMER2_CH2	TIMER2_CH3 TIMER2_UP	•	•	TIMER2_CH0 TIMER2_TG	•
TIMER3	TIMER3_CH0	•	•	TIMER3_CH1	TIMER3_CH2	•	TIMER3_UP
ADC0	ADC0	•	•	•	•	•	•

外设	通道0	通道1	通道2	通道3	通道4	通道5	通道6
SPI/I2S	•	SPI0_RX	SPI0_TX	SPI1/I2S1_RX	SPI1/I2S1_TX	•	•
USART	•	USART2_TX	USART2_RX	USART0_TX	USART0_RX	USART1_RX	USART1_TX
I2C	•	•	•	I2C1_TX	I2C1_RX	I2C0_TX	I2C0_RX

图10-5. DMA1请求映射

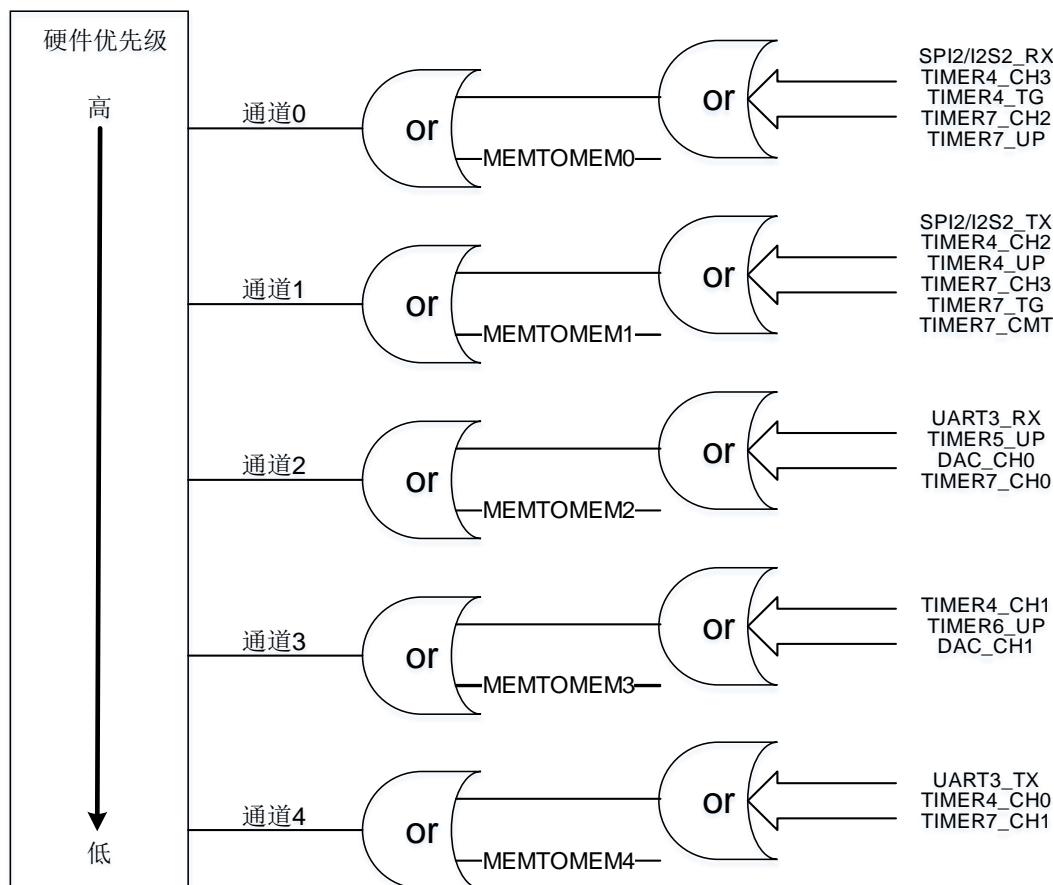


表10-4. DMA1各通道请求表

外设	通道0	通道1	通道2	通道3	通道4
TIMER4	TIMER4_CH3 TIMER4_TG	TIMER4_CH2 TIMER4_UP	•	TIMER4_CH1	TIMER4_CH0
TIMER5	•	•	TIMER5_UP	•	•
TIMER6	•	•	•	TIMER6_UP	•
TIMER7	TIMER7_CH2 TIMER7_UP	TIMER7_CH3 TIMER7_TG TIMER7_CMT	TIMER7_CH0	•	TIMER7_CH1
DAC	•	•	DAC_CH0	DAC_CH1	•
SPI/I2S	SPI2/I2S2_RX	SPI2/I2S2_TX	•	•	•
USART	•	•	UART3_RX	•	UART3_TX

10.5. DMA 寄存器

DMA0 基地址: 0x4002 0000

DMA1 基地址: 0x4002 0400

注意: DMA1 仅有五个通道 (0 到 4 通道), 所有相关寄存器中通道 5 和通道 6 相关标志位不适用于 DMA1。

10.5.1. 中断标志位寄存器 (DMA_INTF)

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		ERRIF6	HTFIF6	FTFIF6	GIF6	ERRIF5	HTFIF5	FTFIF5	GIF5	ERRIF4	HTFIF4	FTFIF4	GIF4
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	ERRIF0	HTFIF0	FTFIF0	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

10.5.2. 中断标志位清除寄存器 (DMA_INTC)

地址偏移: 0x04

复位值: 0x0000 0000

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
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIFC3	HTFIFC3	FTFIFC3	GIFC3	ERRIFC2	HTFIFC2	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/	ERRIFCx	清除通道x(x=0...6)的错误标志位
15/11/7/3		0: 无影响 1: 清零DMA_INTF寄存器的ERRIFx位
26/22/18/	HTFIFCx	清除通道x(x=0...6)的半传输完成标志位
14/10/6/2		0: 无影响 1: 清零DMA_INTF寄存器的HTFIFx位
25/21/17/	FTFIFCx	清除通道x(x=0...6)的传输完成标志位
13/9/5/1		0: 无影响 1: 清零DMA_INTF寄存器的FTFIFx位
24/20/16/	GIFCx	清除通道x(x=0...6)的全局中断标志位
12/8/4/0		0: 无影响 1: 清零DMA_INTF寄存器的GIFx, ERRIFx, HTFIFx和FTFIFx位

10.5.3. 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

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	<p>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</p>
13:12	PRIO[1:0]	<p>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</p>
11:10	MWIDTH[1:0]	<p>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</p>
9:8	PWIDTH[1:0]	<p>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</p>
7	MNAGA	<p>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</p>
6	PNAGA	<p>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</p>
5	CMEN	<p>循环模式使能 软件置位和清零</p>

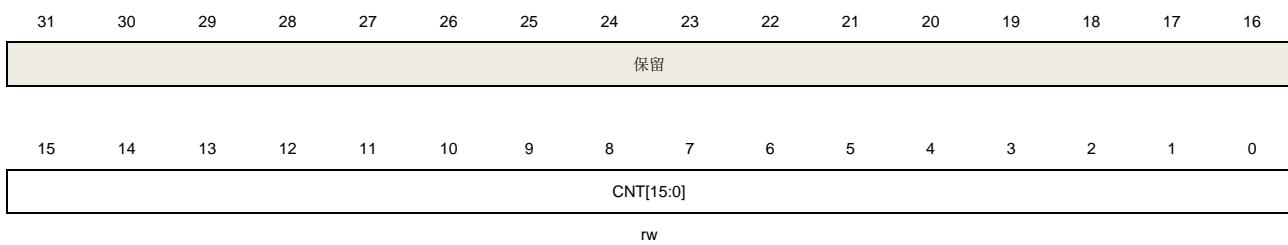
		0: 禁止循环模式 1: 使能循环模式 CHEN 位为1时, 该位不能被配置
4	DIR	传输方向 软件置位和清零 0: 从外设读出并写入存储器 1: 从存储器读出并写入外设 CHEN 位为1时, 该位不能被配置
3	ERRIE	通道错误中断使能位 软件置位和清零 0: 禁止通道错误中断 1: 使能通道错误中断
2	HTFIE	通道半传输完成中断使能位 软件置位和清零 0: 禁止通道半传输完成中断 1: 使能通道半传输完成中断
1	FTFIE	通道传输完成中断使能位 软件置位和清零 0: 禁止通道传输完成中断 1: 使能通道传输完成中断
0	CHEN	通道使能 软件置位和清零 0: 禁止该通道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

10.5.4. 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



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。
------	-----------	---

10.5.5. 通道 x 外设地址寄存器 (**DMA_CHxPADDR**)

x = 0...6, x 为通道序号

地址偏移: 0x10 + 0x14 × x

复位值: 0x0000 0000

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。

10.5.6. 通道 x 存储器地址寄存器 (**DMA_CHxMADDR**)

x = 0..6, x 为通道序号

地址偏移: 0x14 + 0x14 × x

复位值: 0x0000 0000

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。

11. 调试 (DBG)

11.1. 简介

GD32C10x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（SW）调试和跟踪功能，也支持JTAG调试。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实现低功耗模式调试和TIMER、I2C、WWDT、FWDGT和CAN等一些外设调试。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、WWDT、FWDGT和CAN。

11.2. JTAG/SW 功能说明

调试工具可以通过串行（SW）调试接口或者JTAG调试接口来访问调试功能。

11.2.1. 切换 JTAG/ SW 接口

默认使用JTAG调试接口，可以通过下列软件序列从JTAG调试切换到SW调试：

- 发送50个以上TCK周期的TMS=1信号；
- 发送16位TMS = 111001110011110 (0xE79E LSB)信号；
- 发送50个以上TCK周期的TMS=1信号。

切换SW调试到JTAG调试的软件序列：

- 发送50个以上TCK周期的TMS=1信号；
- 发送16位TMS = 1110011100111100 (0xE73C LSB)信号；
- 发送50个以上TCK周期的TMS=1信号。

11.2.2. 引脚分配

JTAG调试提供五个引脚的接口：JTAG时钟引脚（JTCK），JTAG 模式选择引脚（JTMS），JTAG 数据输入引脚（JTDI），JTAG数据输出引脚（JTDO），JTAG复位引脚（NJTRST,低电平有效）。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）。SW调试接口的两个引脚与JTAG调试接口的两个引脚复用，SWDIO和JTMS复用，SWCLK和JTCK复用。

当异步跟踪功能开启时，JTDO引脚也用作异步跟踪数据输出（TRACESWO）。

调试引脚分配：

PA15 : JTDI

PA14 : JTCK/SWCLK
PA13 : JTMS/SWDIO
PB4 : NJTRST
PB3 : JTDO

默认复位后使用五个引脚的JTAG调试，用户可以在不使用NJTRST引脚情况下正常使用JTAG功能，此时PB4可以用作普通GPIO功能（NJTRST硬件拉高）。如果切换到SW调试模式，PA15/PB4/PB3释放作为普通GPIO功能。如果JTAG和SW调试功能都没有使用，这五个引脚都释放作为普通GPIO功能。五个引脚具体配置请参考[GPIO引脚配置](#)。

11.2.3. JTAG 链状结构

Cortex-M4内核的JTAG TAP和边界扫描(BSD) TAP串行连接。边界扫描(BSD)JTAG的IR（指令寄存器）是5位，而Cortex-M4内核的JTAG的IR（指令寄存器）是4位。所以当JTAG进行IR移位输入时，首先移位5位BYPASS指令给BSD JTAG，然后移位4位标准指令给Cortex®-M4 JTAG。当进行数据移位时，数据链只需要额外添加一位，因为BSD JTAG已处在BYPASS模式。

BSD JTAG ID代码是0x790007A3。

11.2.4. 调试复位

JTAG-DP和SW-DP寄存器位于上电复位域。系统复位初始化了Cortex®-M4的绝大部分组件，除了NVIC和调试逻辑（FPB，DWT，ITM）。NJTRST能复位JTAG TAP控制器。所以，可以在系统复位下实现调试功能。例如：复位后停止，用户在系统复位后配置相应停止位，系统复位释放后处理器会立即停止。

11.2.5. JEDEC-106 ID 代码

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

11.3. 调试保持功能说明

11.3.1. 低功耗模式调试支持

当DBG控制寄存器（DBG_CTL）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（DBG_CTL）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（DBG_CTL）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。

11.3.2. TIMER, I2C, WWDGT, FWDGT 和 CAN 外设调试支持

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

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**WWDGT**或者**FWDGT**外设，计数器时钟停止并进行调试；

对于**CAN**外设，接收寄存器停止计数并进行调试。

11.4. DBG 寄存器

DBG 基地址: 0xE004 2000

11.4.1. 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

11.4.2. 控制寄存器 (DBG_CTL)

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. _HOLD	TIMER10 HOLD	TIMER9_ HOLD	TIMER8_ HOLD	TIMER13 _HOLD	TIMER12 _HOLD	TIMER11 _HOLD	保留	CAN1_H OLD	TIMER7_ HOLD	TIMER6_ HOLD	TIMER5_ HOLD	TIMER4_ HOLD	I2C1_HO LD		
	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
I2C0_HO LD	CAN0_H OLD	TIMER3_ HOLD	TIMER2_ HOLD	TIMER1_ HOLD	TIMER0_ HOLD	WWDGT_ HOLD	FWDGT_ HOLD	TRACE_MODE[1:0] _IOEN	TRACE _IOEN	保留	STB_ HOLD	DSLP_ HOLD	SLP_ HOLD		
	rw	rw	rw	rw	rw	rw	rw	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	TIMER10_HOLD	TIMER10 保持寄存器 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持定时器 10 计数器不变, 用于调试

29	TIMER9_HOLD	TIMER9 保持寄存器 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持定时器 9 计数器不变，用于调试
28	TIMER8_HOLD	TIMER8 保持寄存器 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持定时器 8 计数器不变，用于调试
27	TIMER13_HOLD	TIMER13 保持寄存器 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持定时器 13 计数器不变，用于调试
26	TIMER12_HOLD	TIMER12 保持寄存器 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持定时器 13 计数器不变，用于调试
25	TIMER11_HOLD	TIMER11 保持寄存器 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持定时器 13 计数器不变，用于调试
24:22	保留	必须保持复位值
21	CAN1_HOLD	CAN1 保持寄存器 该位由软件置位和复位 0: 无影响 1: 当内核停止时 CAN1 接收寄存器停止接收数据
20	TIMER7_HOLD	TIMER7 保持寄存器 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持定时器 7 计数器不变，用于调试
19	TIMER6_HOLD	TIMER 6 保持寄存器 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持定时器 6 计数器不变，用于调试
18	TIMER5_HOLD	TIMER 5 保持寄存器 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持定时器 5 计数器不变，用于调试
17	TIMER4_HOLD	TIMER 4 保持寄存器 该位由软件置位和复位

		0: 无影响 1: 当内核停止时保持定时器 4 计数器不变，用于调试
16	I2C1_HOLD	I2C1 保持寄存器 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持 I2C1 的状态不变以避免 SMBUS 超时，用于调试
15	I2C0_HOLD	I2C0 保持寄存器 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持 I2C0 的状态不变以避免 SMBUS 超时，用于调试
14	CAN0_HOLD	CAN0 保持寄存器 该位由软件置位和复位 0: 无影响 1: 当内核停止时 CAN0 接收寄存器停止接收数据
13	TIMER3_HOLD	TIMER 3 保持寄存器 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持定时器 3 计数器不变，用于调试
12	TIMER2_HOLD	TIMER 2 保持寄存器 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持定时器 2 计数器不变，用于调试
11	TIMER1_HOLD	TIMER 1 保持寄存器 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持定时器 1 计数器不变，用于调试
10	TIMER0_HOLD	TIMER 0 保持寄存器 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持定时器 0 计数器不变，用于调试
9	WWDGT_HOLD	WWDG 保持寄存器 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 保持寄存器 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
7:6	TRACE_MODE[1:0]	跟踪引脚分配模式

		该位由软件置位和复位 00: 跟踪引脚用于异步模式 01: 跟踪引脚用于同步模式且数据长度为 1 10: 跟踪引脚用于同步模式且数据长度为 2 11: 跟踪引脚用于同步模式且数据长度为 4
5	TRACE_IOEN	跟踪引脚分配使能 该位由软件置位和复位 0: 跟踪引脚分配禁用 1: 跟踪引脚分配使能
4:3	保留	必须保持复位值
2	STB_HOLD	待机模式保持寄存器 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	深度睡眠模式保持寄存器 该位由软件置位和复位 0: 无影响 1: 在深度睡眠模式下, 系统时钟和 AHB 时钟由 CK_IRC8M 提供
0	SLP_HOLD	睡眠模式保持寄存器 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

12. 模数转换器 (ADC)

12.1. 简介

12位ADC是一种采用逐次逼近方式的模拟数字转换器。ADC可以转换16个外部通道和2个内部通道的模拟信号。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阈值。各种通道的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来自MCU的相关计算负担来提高性能。

12.2. 主要特征

- 高性能:
 - 可配置12位、10位、8位、或者6位分辨率;
 - 自校准;
 - 可编程采样时间;
 - 数据寄存器可配置数据对齐方式;
 - 支持规则数据转换的DMA请求。
- 模拟输入通道:
 - 16个外部模拟输入通道;
 - 1个内部温度传感器输入通道(V_{SENSE});
 - 1个内部参考电压输入通道(V_{REFINT})。
- 转换开始的发起:
 - 软件;
 - 硬件触发。
- 转换模式:
 - 转换单个通道，或者扫描一组通道;
 - 单次模式，每次触发转换一次选择的输入通道;
 - 连续模式，连续转换所选择的输入通道;
 - 间断模式;
 - 同步模式（适用于具有两个或多个ADC的设备）。
- 模拟看门狗。
- 中断的产生:
 - 规则组或注入组转换结束;
 - 模拟看门狗事件。
- 过采样:
 - 16位的数据寄存器;
 - 可调整的过采样率，从 $2\times$ 到 $256\times$;
 - 高达8位的可编程数据移位。
- ADC供电要求:
 - 2.4V到3.6V，一般供电电压为3.3V。

- ADC输入范围: $V_{REF-} \leq V_{IN} \leq V_{REF+}$ 。

12.3. 引脚和内部信号

[图12-1. ADC模块框图](#) 给出了ADC模块框图。[表12-2. ADC引脚定义](#)给出了ADC引脚说明。

表 12-1. ADC 内部信号

内部信号名称	信号类型	说明
V_{SENSE}	输入	内部温度传感器电压输出
V_{REFINT}	输入	内部参考电压输出

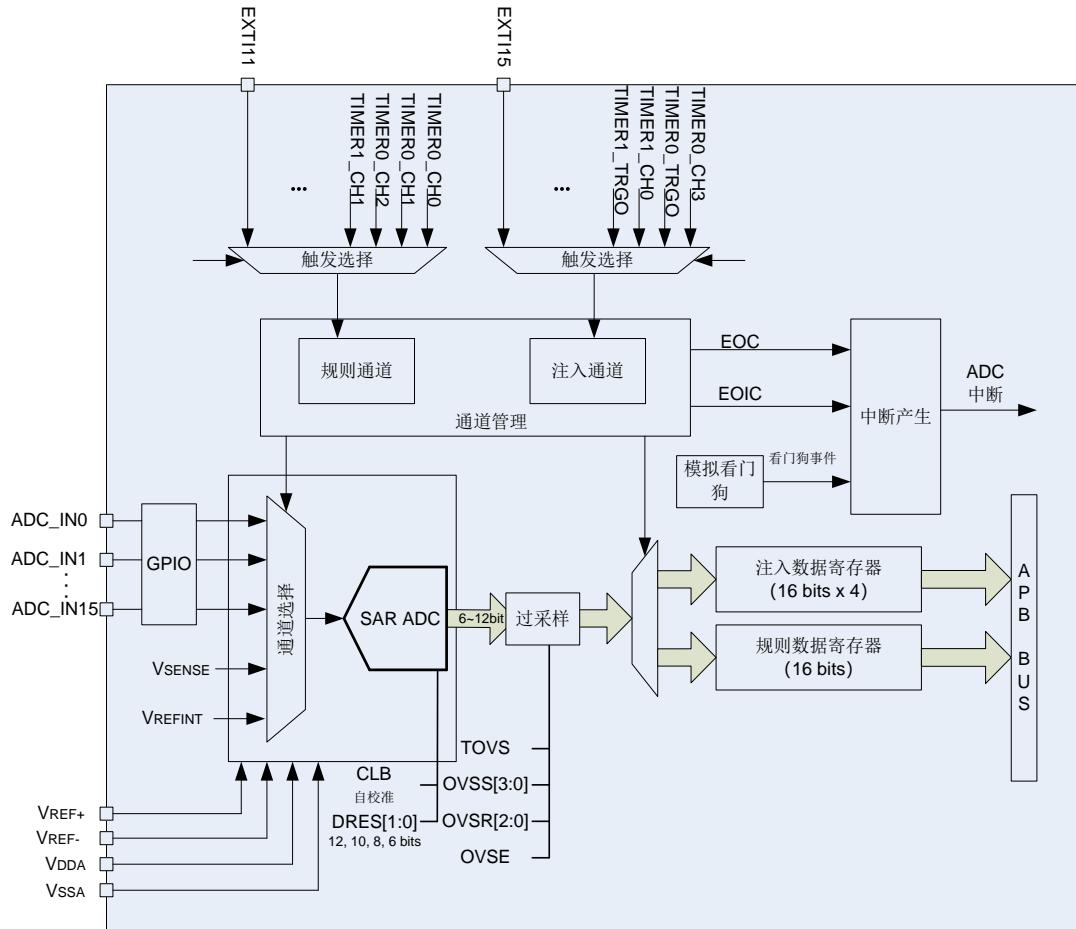
表 12-2. ADC 引脚定义

名称	信号类型	注释
V_{DDA}	输入, 模拟供电电源	模拟电源输入等于 V_{DD} , $2.4V \leq V_{DDA} \leq 3.6V$
V_{SSA}	输入, 模拟电源地	模拟地, 等于 V_{SS}
V_{REF+}	输入, 正模拟参考电压	ADC正参考电压, $2.4V \leq V_{REF+} \leq V_{DDA}$
V_{REF-}	输入, 负模拟参考电压	ADC负参考电压, $V_{REF-} = V_{SSA}$
$ADCx_IN[15:0]$	输入, 模拟信号	多达16路外部通道

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

12.4. 功能说明

图 12-1. ADC 模块框图



12.4.1. 校准 (CLB)

ADC带有一个前置校准功能。在校准期间，ADC计算一个校准因子，这个因子是应用于ADC内部的，它直到ADC下次掉电才无效。在校准期间，应用不能使用ADC，必须等到校准完成。在开始A/D转换前应执行校准操作。通过软件设置CLB=1启动校准。在校准期间CLB位会一直保持1。一旦校准完成，CLB位由硬件清0。

当ADC运行条件改变(例如，V_{DDA}、V_{REF+} 以及温度等)，建议重新执行一次校准操作。

内部的模拟校准可以通过设置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 (该步骤是可选的)；
4. 设置CLB=1；
5. 等待直到CLB=0。

12.4.2. ADC 时钟

ADCCLK时钟是由RCU时钟控制器提供的，它和AHB、APB2时钟保持同步。在RCU时钟控制器中，有一个专门用于ADC时钟的可编程分频器。

12.4.3. ADCON 开关

ADC_CTL1寄存器中的ADCON位是ADC模块的使能开关。如果该位为0，则ADC模块保持复位状态。为了省电，当ADCON位为0时，ADC模拟子模块将会进入掉电模式。ADC使能后需等待tsu时间后才能采样，tsu数值详见芯片数据手册。

12.4.4. 规则组和注入组

ADC支持18个多路通道，可以把转换结果组织成两组：一个规则组通道组和一个注入组通道组。

规则组，可以按照特定的序列组织成多达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寄存器的RL[3:0]位规定了整个规则组转换序列的长度。

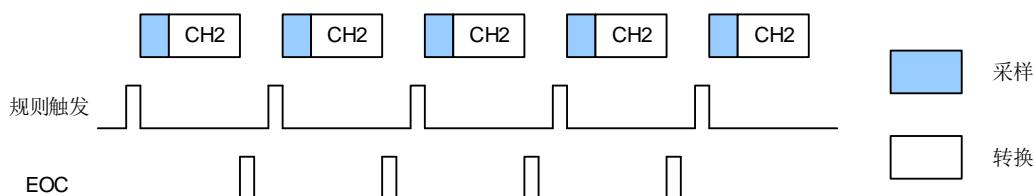
注入组，可以按照特定的序列组织成多达4个转换的序列。ADC_ISQ寄存器规定了注入组的通道选择。ADC_ISQ寄存器的IL[1:0]位规定了整个注入组转换序列的长度。

12.4.5. 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]位或者ADC_ISQ寄存器的ISQ3[4:0]位规定了ADC的转换通道。当ADCON位被置1，一旦相应软件触发或者外部触发发生，ADC就会采样和转换一个通道。

图 12-2. 单次转换模式



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

注入通道单次转换结束后，转换数据将被存放于ADC_IDATA0寄存器中，EOC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寄存器的DISRC和SM位以及ADC_CTL1寄存器的CTN位为0；
2. 用模拟通道编号来配置RSQ0；
3. 配置ADC_SAMPTx寄存器；

4. 如果有需要，可以配置ADC_CTL1寄存器的ETERC和ETSRC位；
5. 设置SWRCST位，或者为规则组产生一个外部触发信号；
6. 等到EOC置1；
7. 从ADC_RDATA寄存器中读ADC转换结果；
8. 写0清除EOC标志位。

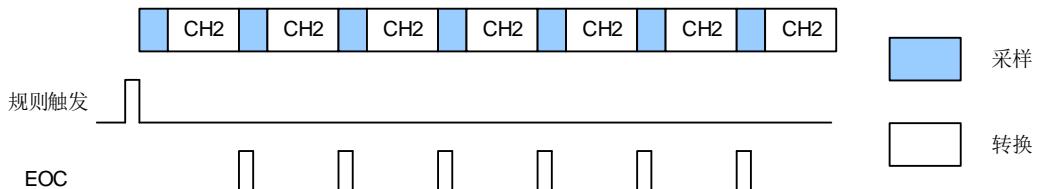
注入组单次转换模式的软件流程：

1. 确保ADC_CTL0寄存器的DISRC和SM位为0；
2. 用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. 等到EOC、EOIC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。对ADC_CTL1寄存器的CTN位置1可以使能连续转换模式。在此模式下，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 12-3. 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. 只要还需要进行连续转换，重复步骤6~8。

可以使用DMA来传输转换数据，这样就不用循环查询EOC标志位了：

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和ETSRC位；

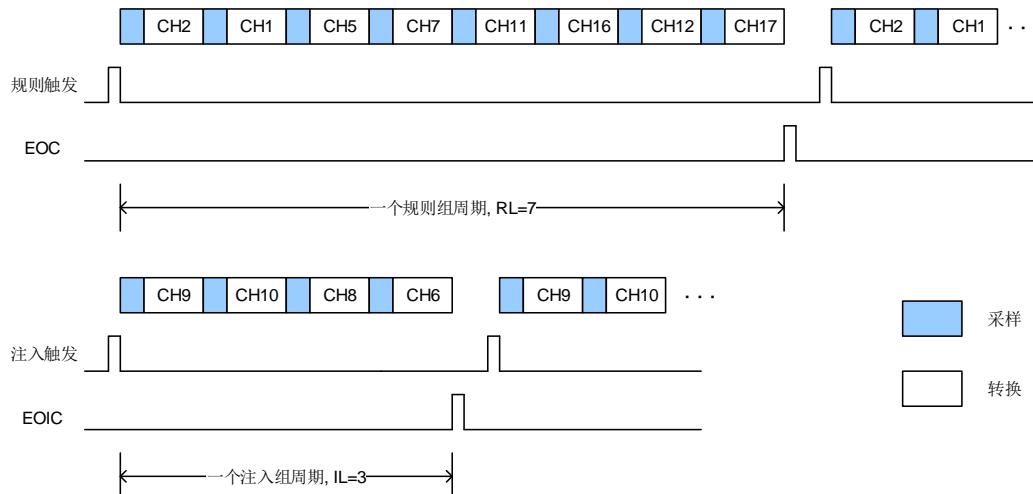
5. 准备DMA模块，用于传输来自ADC_RDATA的数据；
6. 设置SWRCST位，或者给规则组产生一个外部触发。

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。在此模式下，ADC扫描转换所有被ADC_RSQ0~ADC_RSQ2寄存器或ADC_ISQ寄存器选中的所有通道。一旦ADCON位被置1，当相应软件触发或者外部触发产生，ADC就会一个接一个的采样和转换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或者EOIC位将被置1。如果EOCIE或EOICIE位被置1，将产生中断。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，则在规则组通道转换完之后，转换会自动重新开始。

图 12-4. 扫描转换模式，且连续转换模式失能

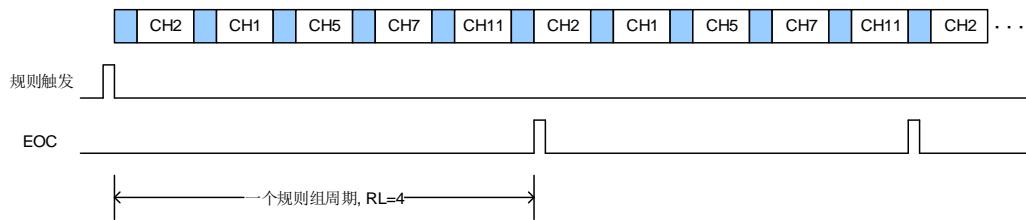


规则组扫描转换模式的软件流程：

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，用于传输来自 ADC_RDATA 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 SWICST 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。

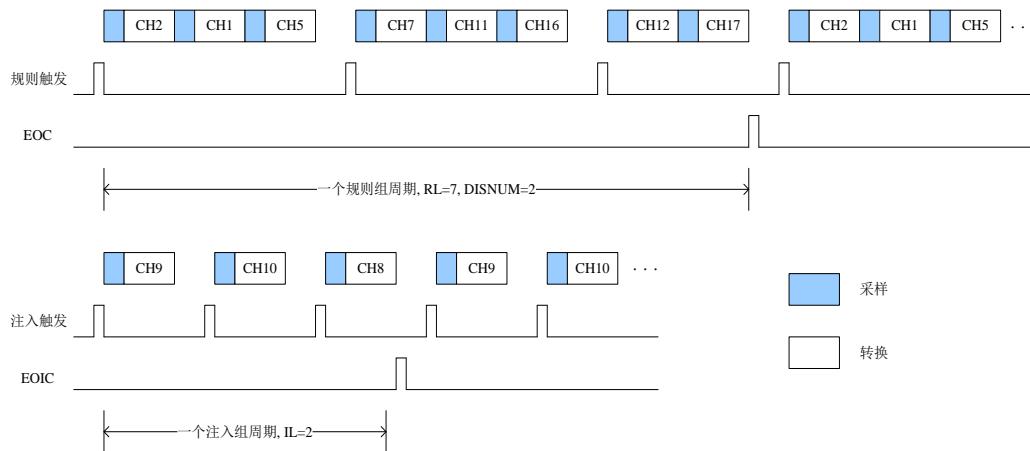
图 12-5. 扫描转换模式，连续转换模式使能


间断模式

对于规则组，当 ADC_CTL0 寄存器的 DISRC 位被置 1 时，规则组间断转换模式被使能。该模式下可以执行一次 n 个通道的短序列转换($n \leq 8$)，这个短序列是 ADC_RSQ0~RSQ2 寄存器所选择的转换序列的一部分。数值 n 由 ADC_CTL0 寄存器的 DISCNUM[2:0]位给出。当相应的软件触发或外部触发发生，ADC 就会采样和转换在 ADC_RSQ0~RSQ2 寄存器所选择通道中接下来的 n 个通道，直到规则序列中所有的通道转换完成。每个规则组转换周期结束后，EOC 位将被置 1。如果 EOIC 位被置 1 将产生一个中断。

对于注入组，当 ADC_CTL0 寄存器的 DISIC 位被置 1 时，注入组间断转换模式被使能。该模式下可以执行 ADC_ISQ 寄存器所选择的转换序列的一个通道进行转换。当相应的软件触发或外部触发发生，ADC 就会采样和转换 ADC_ISQ 寄存器中所选择通道的下一个通道，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 12-6. 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 和 ETSRC 位;
5. 准备 DMA 模块，用于传输来自 ADC_RDATA 的数据;
6. 设置 SWRCST 位，或者给规则组产生一个外部触发;
7. 如果需要，重复步骤 6;

8. 等待 EOC 标志位置 1;
9. 写 0 清除 EOC 标志位。

注入组间断模式的软件流程:

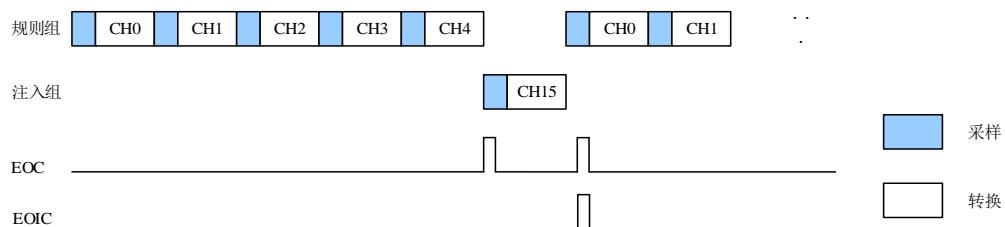
1. 设置 ADC_CTL0 寄存器的 DISRC 位为 1;
2. 配置 ADC_ISQ 和 ADC_SAMPTx 寄存器;
3. 如果有需要, 配置 ADC_CTL1 寄存器中的 ETEIC 和 ETSIC 位;
4. 设置 SWICST 位, 或者给注入组产生一个外部触发;
5. 如果需要, 重复步骤 4;
6. 等待 EOC、EOIC 标志位置 1;
7. 读 ADC_IDATAx 寄存器中的转换结果;
8. 写 0 清除 EOC、EOIC 标志位。

12.4.6. 注入通道管理

自动注入

如果将 ADC_CTL0 寄存器的 ICA 位置 1, 在规则组通道之后, 注入组通道被自动转换。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 和 ADC_ISQ 寄存器中设置的多至 20 个转换序列。除了 ICA 位之外, 如果 CTN 位也被置 1, 规则组通道将在注入组通道之后被连续转换。

图 12-7. 自动注入, CNT=1

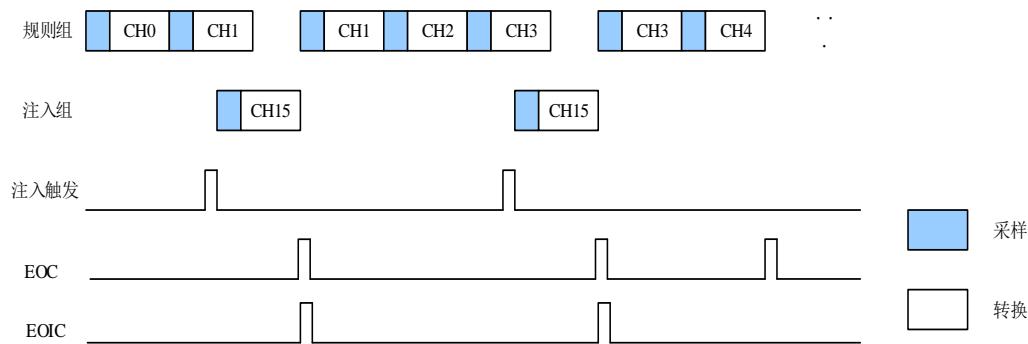


不能同时使用自动注入和间断模式。

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发生, 则启动触发注入转换。这种情况下, ADC 取消当前转换, 注入通道序列进行转换。注入通道组转换结束后, 规则组转换从上次被取消的转换处重新开始。

图 12-8. 触发注入



12.4.7. 模拟看门狗

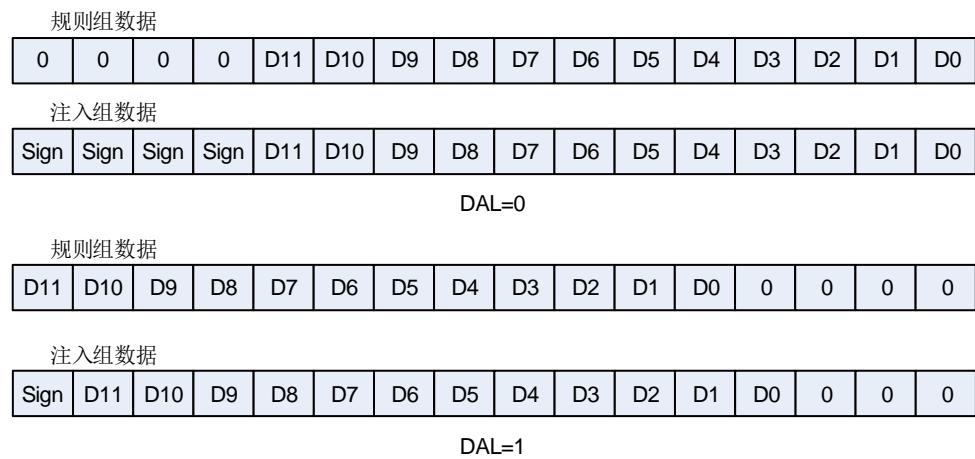
ADC_CTL0 寄存器的 RWDEN 和 IWDEN 位置 1 将分别使能规则组和注入组的模拟看门狗功能。如果 ADC 的模拟转换电压低于低阈值或高于高阈值时，ADC_STAT 状态寄存器的 WDE 位将被置 1。如果 WDEIE 位被置 1，将产生中断。ADC_WDHT 和 ADC_WDLT 寄存器用来设定高低阈值。内部数据的比较在对齐之前完成，因此阀值与 ADC_CTL1 寄存器的 DAL 位确定的对齐方式无关。ADC_CTL0 寄存器的 RWDEN, IWDEN, WDSC 和 WDCHSEL[4:0]位可以用来选择模拟看门狗监控单一通道或者多通道。

12.4.8. 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。

图 12-9. 12 位数据对齐



6 位分辨率的数据对齐不同于 12 位/10 位/8 位分辨率数据对齐，如 [图 12-10. 6 位数据对齐](#) 所示。

图 12-10. 6 位数据对齐

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	Sign	Sign	Sign	Sign	Sign	Sign	Sign	Sign	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	Sign	Sign	Sign	Sign	Sign	Sign	Sign	Sign	Sign	D5	D4	D3	D2	D1	D0
DAL=1															

12.4.9. 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]位更改。每个通道可以用不同的时间采样。在 12 位分辨率的情况下，总转换时间=采样时间+12.5 个 ADCCLK 周期。

例如：

ADCCLK = 30MHz，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 0.467us。

12.4.10. 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 12-3. ADC0 和 ADC1 的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	TIMER3_CH3	
110	EXTI11/TIMER7_TRGO	外部信号/内部信号
111	SWRCST	软件触发

表 12-4. ADC0 和 ADC1 的注入通道的外部触发

ETDIC[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	TIMER3_TRGO	
110	EXTI15/TIMER7_CH3	外部信号/内部信号
111	SWICST	软件触发

12.4.11. DMA 请求

DMA 请求，可以通过设置 ADC_CTL1 寄存器的 DMA 位来使能，用来传输规则组多个通道的转换结果。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。

12.4.12. 温度传感器和内部参考电压 VREFINT

将 ADC_CTL1 寄存器的 TSVREN 位置 1 可以使能温度传感器通道(ADC0_CH16)和 V_{REFINT} 通道(ADC0_CH17)。温度传感器可以用来测量器件周围的温度。传感器输出电压能被 ADC 转换成数字量。建议温度传感器的采样时间至少设置为 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 ADC0_CH17 输入通道。

12.4.13. 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_OVSAMPCTL 中的 DRES[1:0]位进行编程即可配置分辨率为 6、8、10、12 位。对于那些不需要高精度数据的应用，可以使用较低的分辨率来实现更快速地转换。只有在 ADCON 比特为 0 时，才能修改 DRES[1:0]的值。较低的分辨率能够减少转换时间。如[表 12-5. 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 12-5. 不同分辨率对应的 t_{CONV} 时间

DRES[1:0] bits	t _{CONV} (ADC clock cycles)	t _{CONV(ns)} at f _{ADC} =30MHz	t _{SMPL(min)} (ADC clock cycles)	t _{ADC} (ADC clock	t _{ADC(us)} at f _{ADC} =30M Hz
6	16	16	16	16	16

				(cycles)	
12	12.5	417 ns	1.5	14	467 ns
10	10.5	350 ns	1.5	12	400 ns
8	8.5	283 ns	1.5	10	333 ns
6	6.5	217 ns	1.5	8	267 ns

12.4.14. 片上硬件过采样

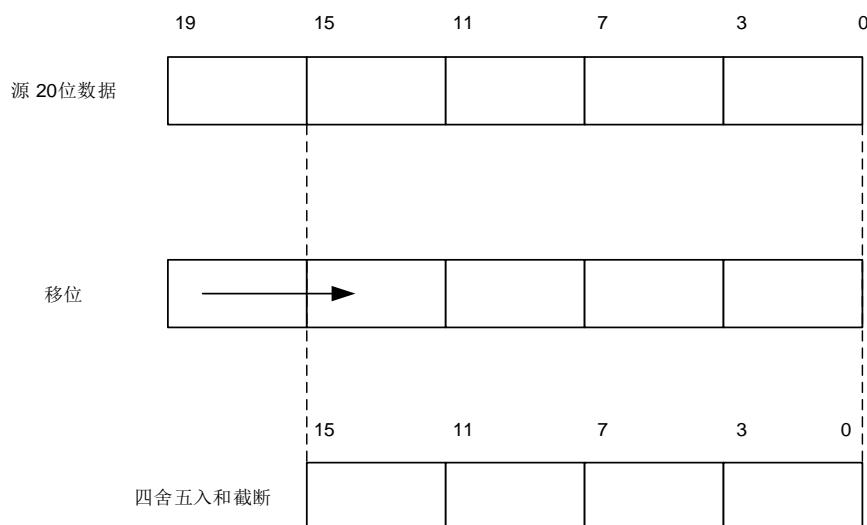
片上硬件过采样单元执行数据预处理以减轻 CPU 负担。它能够处理多个转换，并将多个转换的结果取平均，得出一个 16 位宽的数据。其结果根据如下公式计算得出，其中 N 和 M 的值可以被调整，过采样单元可以通过设置 ADC_OVSAMPCTL 寄存器的 OVSE 位来使能，它是以降低数据输出率为代价，换取较高的数据分辨率。D_{out}(n)是指 ADC 输出的第 n 个数字信号：

$$\text{Result} = \frac{1}{M} * \sum_{n=0}^{N-1} D_{\text{out}}(n) \quad (12-1)$$

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，将这个值要进行右移，将移位后剩余的部分再通过取整转化一个近似值，最后将高位会被截断，仅保留最低 16 位有效位作为最终值传入对应的数据寄存器中。

图 12-11. 20 位到 16 位的结果截断



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

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

图 12-12. 右移 5 位和取整的数例

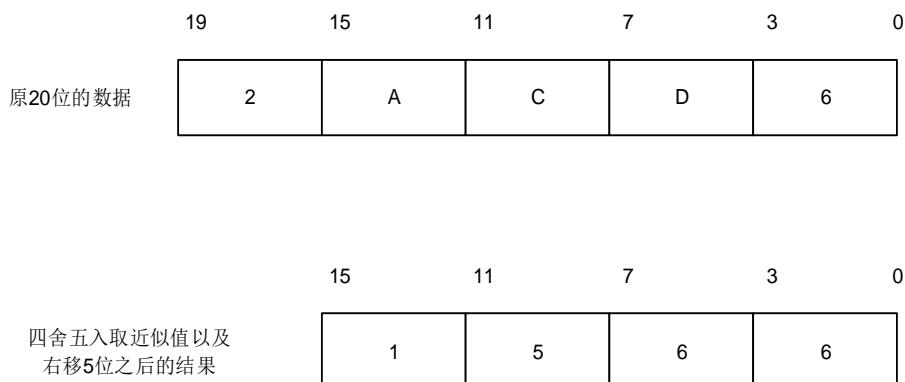


表 12-6. 不同 N 和 M 组合的最大输出值 (灰色值表示截断) 给出了 N 和 M 各种组合的数据格式, 初始转换值为 0xFFFF。

表 12-6. 不同 N 和 M 组合的最大输出值 (灰色值表示截断)

Oversampling ratio	Max Raw data	No-shift OVSS= 0000	1-bit shift OVSS= 0001	2-bit shift OVSS= 0010	3-bit shift OVSS= 0011	4-bit shift OVSS= 0100	5-bit shift OVSS= 0101	6-bit shift OVSS= 0110	7-bit shift OVSS= 0111	8-bit shift OVSS= 1000
2x	0x1FFE	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F	0x003F	0x001F
4x	0x3FFC	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F	0x003F
8x	0x7FF8	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F
16x	0xFFFF0	0xFFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF
32x	0x1FFE0	0xFFE0	0xFFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF
64x	0x3FFC0	0xFFC0	0xFFE0	0xFFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF
128x	0x7FF80	0xFF80	0xFFC0	0xFFE0	0xFFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF
256x	0xFFFF00	0xFF00	0xFF80	0xFFC0	0xFFE0	0xFFFF0	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:

$$N \times t_{ADC} = N \times (t_{SMPL} + t_{CONV}) \quad (12-2)$$

12.5. ADC 同步模式

在有两个或者两个以上的ADC模块的产品中, 可以使用ADC同步模式。

在ADC同步模式下, 根据ADC0_CTL0寄存器中SYNCM[3:0]位所选的模式, 转换可以被ADC0(主机)和ADC1(从机)的交替触发或同步触发。

在同步模式下, 当主ADC配置由外部事件触发的转换时, 从ADC必须配置为软件触发, 从而避免错误的触发引起的不必要的转换。此外, 对于主ADC和从ADC的外部触发必须被使能。

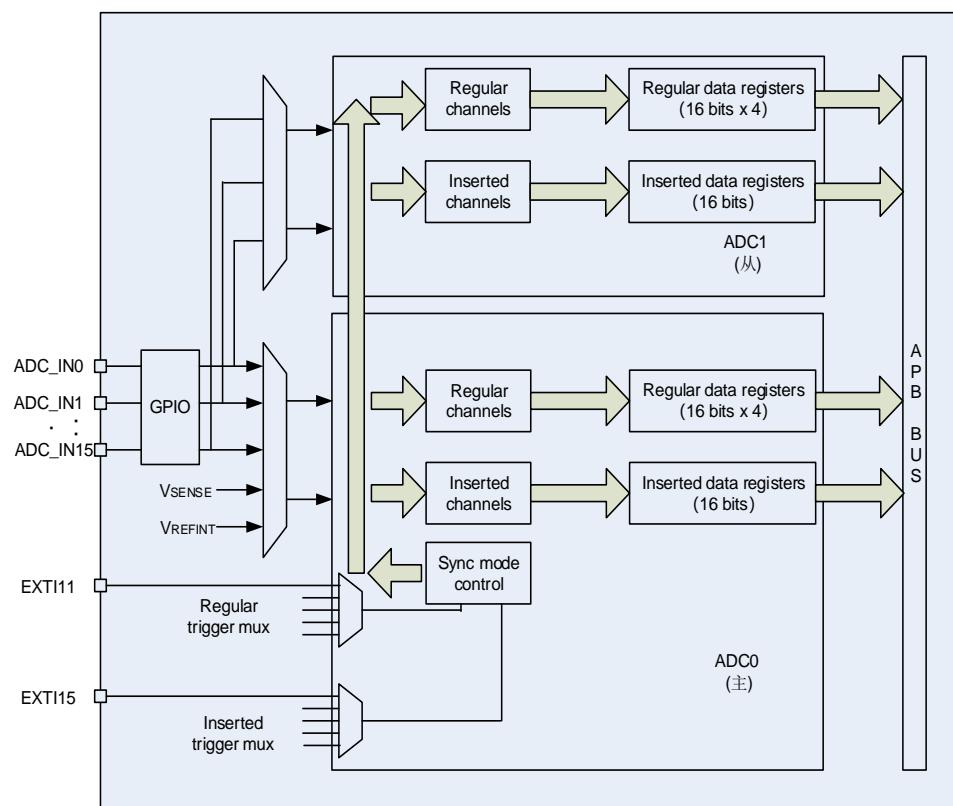
共有以下几种模式:

- 独立模式
- 规则并行模式
- 注入并行模式
- 快速交叉模式
- 慢速交叉模式
- 交替触发模式
- 注入并行模式+规则并行模式
- 规则并行模式+交替触发模式
- 注入并行模式+快速交叉模式
- 注入并行模式+慢速交叉模式

在 ADC 同步模式下，即使 DMA 不用，也要将 DMA 置位，从 ADC 的转换数据可以通过主 ADC 数据寄存器读取。

ADC 同步框图如 [图 12-13. ADC 同步框图](#) 所示。

图 12-13. ADC 同步框图



12.5.1. 独立模式

在这种模式下，ADC 同步是忽略的，每个 ADC 都独立工作。

12.5.2. 规则并行模式

此模式可并行转换规则通道组，外部触发来源于规则通道组的 ADC0 的多路开关（由 ADC0_CTL1 寄存器的 ETSRC[2:0]决定），ADC1 被并行触发。

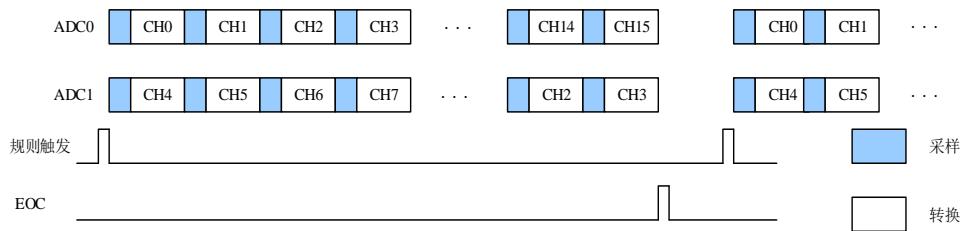
在 ADC0 或 ADC1 的转换结束时，即 ADC0 或 ADC1 的规则通道组转换完毕，会产生一个 EOIC 中断。规则并行模式请参考 [图 12-14. 基于 16 个通道的规则并行模式](#)。

32 位 ADC_RDATA 寄存器包含上半字（由 ADC1 转换的数据构成）和下半字（由 ADC0 转换的数据构成），32 位的 DMA 被用来将 ADC_RDATA 中的数据传送到 SRAM。

注意：

- 1.不要在两路 ADC 上转换相同的通道（两路 ADC 在同一通道转换时采样时间不可重叠）。
- 2.在并行模式下，ADC0 和 ADC1 并行采样的两个通道的需要设置为准确的相同采样时间。

图 12-14. 基于 16 个通道的规则并行模式



12.5.3. 注入并行模式

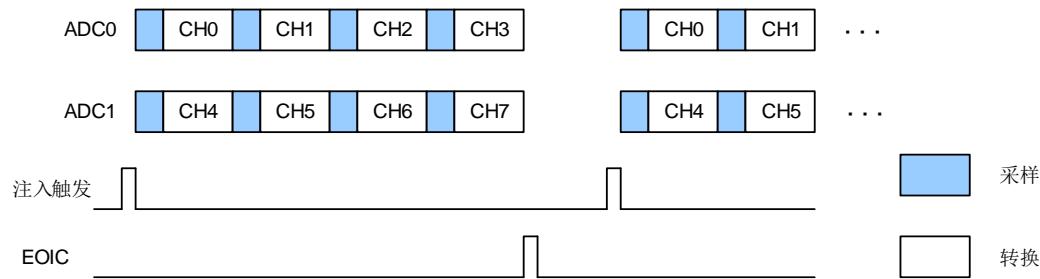
此模式同时转换一个注入通道组,外部触发来源于注入通道组的多路开关（由 ADC_CTL1 寄存器的 ETSIC[2:0]决定），ADC1 被并行触发。

在 ADC0 或 ADC1 的转换事件结束时，即 ADC0 和 ADC1 的注入通道组被转换完，会产生一个 EOIC 中断,其转换的数据被存储在各自的 ADC_IDATAx 寄存器中.如 [图 12-15. 4 个通道的注入并行模式](#)。

注意：

- 1.不要在两路 ADC 上转换相同的通道（两路 ADC 在同一通道转换时采样时间不可重叠）。
- 2.在并行模式下，ADC0 和 ADC1 并行采样的两个通道的需要设置为准确的相同采样时间。

图 12-15. 4 个通道的注入并行模式



12.5.4. 快速交叉模式

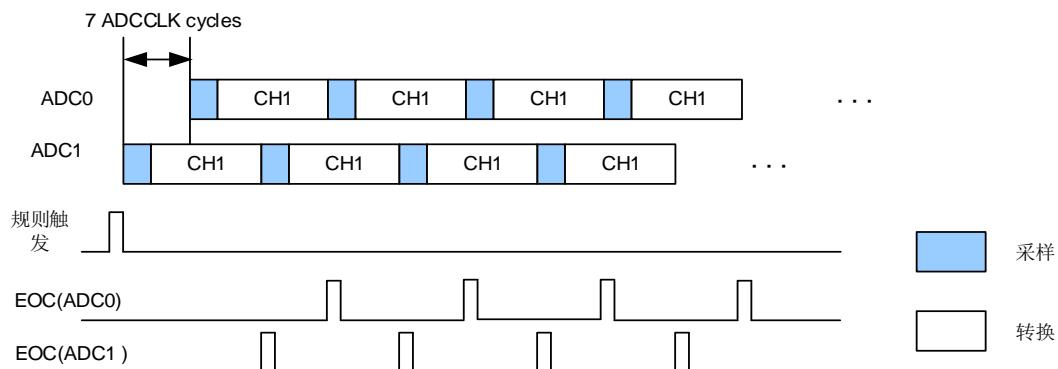
此模式应用于规则通道组（通常一个通道），外部触发来源于规则通道组的 ADC0 的多路开关（由 ADC_CTL1 寄存器的 ETSRC[2:0]决定）。当触发产生时，ADC1 立刻启动，而 ADC0 在 7 个 ADC 时钟周期后启动。

如果 ADC0 和 ADC1 的 CTN 位被置位，所选的规则通道组在两个 ADC 中被不停的转换，如 [图 12-16. 一个采用连续转换模式通道上的快速交叉模式](#)。

32 位 ADC_RDATA 寄存器包含上半字（由 ADC1 转换的数据构成）和下半字（由 ADC0 转换的数据构成）。在 ADC0 产生 EOC 中断后（可通过置位 EOCIE 位），可通过 32 位 DMA 将 ADC_RDATA 中数据传送到 SRAM。

注意：可允许的最大采样时间必须小于 7 个 ADC 采样时钟，从而避免 ADC0 和 ADC1 在转换相同通道时出现采样时钟重叠。

图 12-16. 一个采用连续转换模式通道上的快速交叉模式



12.5.5. 慢速交叉模式

此模式应用于规则通道组（通常一个通道），外部触发来源于规则通道组的 ADC0 的多路开关（由 ADC_CTL1 寄存器的 ETSRC[2:0]决定）。当触发产生时，ADC1 立刻启动，而 ADC0 在 14 个 ADC 时钟周期后启动，在 ADC0 启动后的 14 个时钟周期，ADC1 再次启动。

在这种模式下，不能使用连续转换模式，因为在这种模式下所选的规则通道组在两个 ADC 中被不停的转换，如 [图 12-17. 一个通道上的慢速交叉模式](#)。

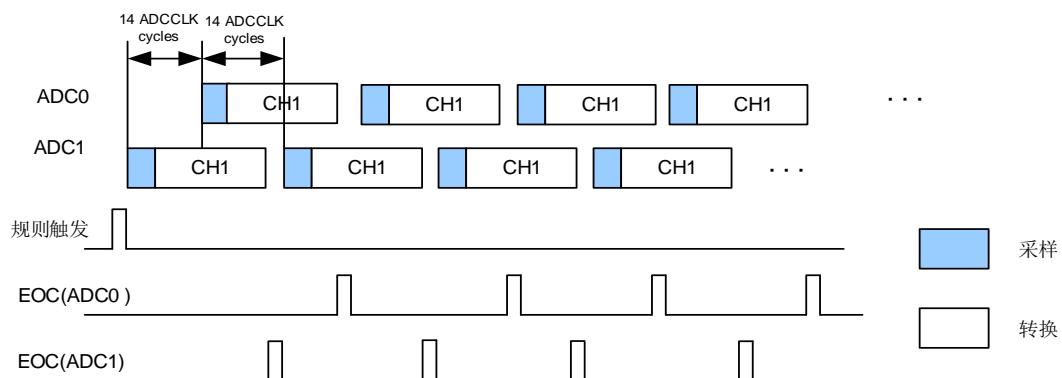
32 位 ADC_RDATA 寄存器包含上半字（由 ADC1 转换的数据构成）和下半字（由 ADC0 转换的数据构成）。在 ADC0 产生 EOC 中断后（可通过置位 EOCIE 位），可通过 32 位 DMA 将 ADC_RDATA 中数据传送到 SRAM。

注意：

1. 可允许的最大采样时间必须小于 14 个 ADC 采样时钟，从而避免 ADC0 和 ADC1 在转换相同通道时出现采样时钟重叠。

2. 在快速交叉模式和慢速交叉模式下，必须确保没有注入通道事件被外部触发。

图 12-17. 一个通道上的慢速交叉模式



12.5.6. 交替触发模式

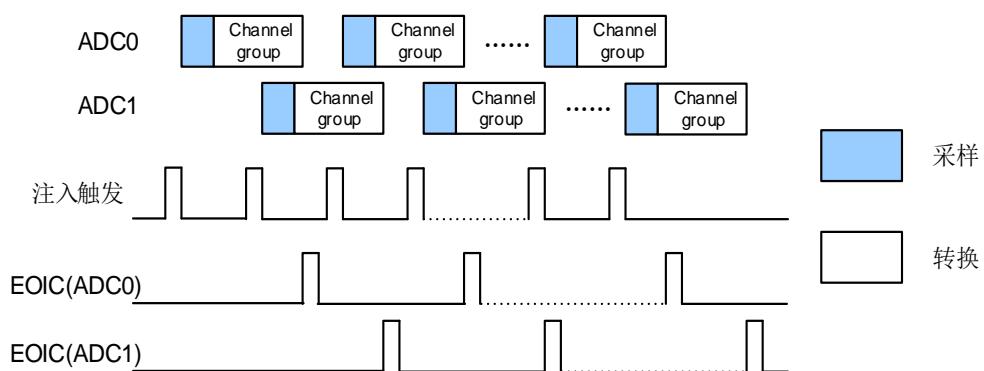
此模式应用于注入通道组，外部触发来源于 ADC0 的注入通道组多路开关（由 ADC_CTL1 寄存器的 ETSIC[2:0]决定）。

当第一次触发发生，ADC0 所有的注入通道被转换，当第二次触发发生，ADC1 所有的注入通道被转换。

如果 ADC0 和 ADC1 的 EOIC 中断被使能，二者的通道被转换完毕，其相应的中断就会被触发。

如果所有的注入通道组转换完成后另一个外部触发发生，那么交替触发进程将通过转换 ADC0 注入通道组实现重启。

图 12-18. 交替触发：注入通道组

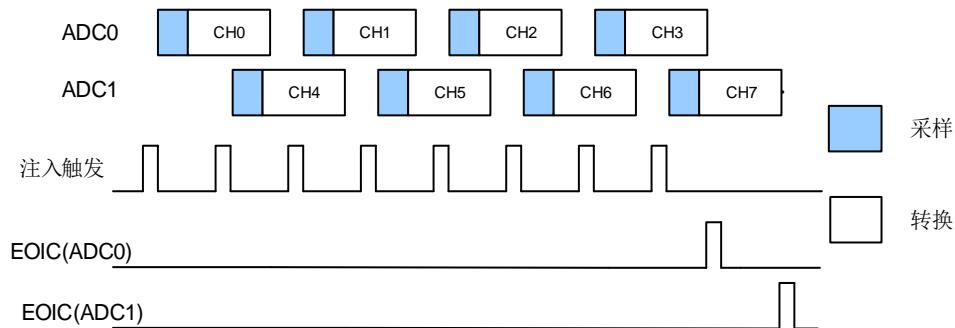


如果使能 ADC0 和 ADC1 间断模式，当第一次触发发生，ADC0 转换第一个注入通道。当第二次触发发生，ADC1 转换第一个注入通道。然后，ADC0 转换第二个注入通道，ADC1 转换第二个通道，以此类推。

如果 ADC0 和 ADC1 的 EOIC 中断被使能，二者的通道被转换完毕，其相应的中断就会被触发。

如果所有的注入通道组转换完成后另一个外部触发发生，那么交替触发进程将实现重启。

图 12-19. 交替触发:间断模式下注入通道组



12.5.7. 规则并行和注入并行组合模式

在独立模式下，规则通道组的转换可以被注入通道组的转换所中断。在 ADC 同步模式下，注入并行通道组的转换也可以中断规则并行通道组的转换。

注意：在规则并行和注入并行组合模式下，两路通道的采样时间应该被设置成相同值。

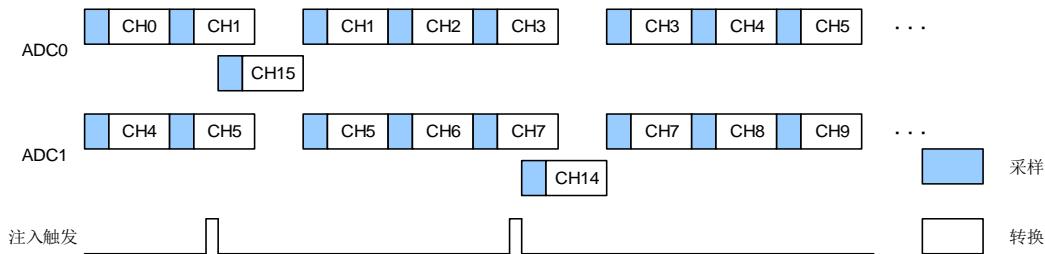
12.5.8. 规则并行和交替触发组合模式

注入通道组的交替触发转换的启动可以中断规则通道组的并行转换。如图所示，交替触发中断规则并行转换。

当注入通道事件发生时，注入交替转换立刻启动。如果规则转换此刻正在运行，为保证注入通道组转换启动后的同步，规则通道组在两路 ADC 的转换停止。当注入通道组转换完成后，规则通道组的转换同步启动。

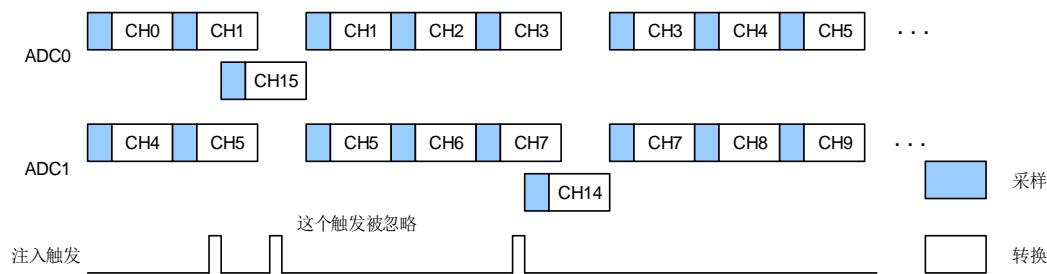
注意：在规则并行和交替触发组合模式下，两路通道的采样时间应该被设置成相同值。

图 12-20. 规则并行和交替触发组合模式



如果在一个注入转换期间，另一个注入触发出现，那么后面的这个触发将会被忽略，如[图 12-21. 在注入转换过程中触发现象](#)所示。

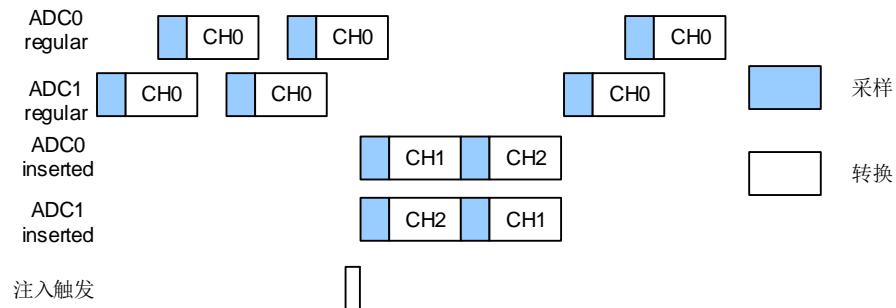
图 12-21. 在注入转换过程中触发出现



12.5.9. 注入并行和交叉组合模式

通过注入转换可以中断交叉转换，当注入触发发生时，交叉转换被中断，注入转换启动。在注入转换完成后，交叉转换恢复，如[图12-22. 交叉的单通道转换被注入序列CH1和CH2中断](#)所示。

图 12-22. 交叉的单通道转换被注入序列 CH1 和 CH2 中断



12.6. 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；

ADC0、ADC1都被映射到同一个中断向量IRQ18。

12.7. ADC 寄存器

ADC0地址: 0x4001 2400

ADC1地址: 0x4001 2800

12.7.1. 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	

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 规则组转换开始时硬件置位。 软件写0清除。
3	STIC	注入通道转换开始标志 0: 注入通道组转换没有开始 0: 注入通道组转换开始 注入通道组转换开始时硬件置位。 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寄存器设定的阈值时由硬件置1，软件写0清除。
---	-----	--

12.7.2. 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					RWDEN	IWDEN	保留		SYNCM[3:0]							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	
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	rw	rw	rw	rw	

位/位域	名称	描述
31:24	保留	必须保持复位值
23	RWDEN	规则组看门狗使能 0: 规则组看门狗禁止 1: 规则组看门狗使能
22	IWDEN	注入组看门狗使能 0: 注入组看门狗禁止 1: 注入组看门狗使能
21:20	保留	必须保持复位值
19:16	SYNCM[2:0]	同步模式选择 这些位用于运行模式选择 0000: 独立模式 0001: 规则并行+注入并行组合模式 0010: 规则并行+交替触发组合模式 0011: 注入并行+快速交叉组合模式 0100: 注入并行+慢速交叉组合模式 0101: 注入并行模式 0110: 规则并行模式 0111: 快速交叉模式 1000: 慢速交叉模式 1001: 交替触发模式

注意: 在ADC1中这些位为保留位。在同步模式下, 改变通道的配置会产生一个重新

开始的条件，会导致同步丢失，建议在任何配置之前关闭同步模式。

15:13	DISNUM[2:0]	间断模式下的转换数目 触发后即将被转换的通道数目将变成 DISNUM[2:0]+1
12	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 00011: ADC 通道3 00100: ADC 通道4 00101: ADC 通道5 00110: ADC 通道6 00111: ADC 通道7 01000: ADC 通道8 01001: ADC 通道9 01010: ADC 通道10

01011: ADC 通道11

01100: ADC 通道12

01101: ADC 通道13

01110: ADC 通道14

01111: ADC 通道15

10000: ADC 通道16

10001: ADC 通道17

其他值保留。

注意: ADC0的模拟输入通道16和通道17分别连接到温度传感器和V_{REFINT}。 ADC1的模拟输入通道16和通道17内部都连接到V_{SSA}。

12.7.3. 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					TSVREN	SWRCST	SWICST	ETERC	ETSRC[2: 0]				保留.	保留	
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	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	TSVREN	ADC0的通道16和17使能 0: ADC0的通道16和17禁止 1: ADC0的通道16和17使能
22	SWRCST	规则组转换开始。 如果ETSRC是111, 该位置'1'开启规则组转换。软件置位, 软件清零, 或转换开始后, 立刻由硬件清零。
21	SWICST	注入组转换开始。 如果ETSIC是111, 该位置'1'开启注入组转换。软件置位, 软件清零, 或转换开始后, 立刻由硬件清零。
20	ETERC	规则组外部触发使能 0: 规则组外部触发禁止 1: 规则组外部触发使能
19:17	ETSRC[2:0]	规则组外部触发选择 对于 ADC0与ADC1: 000: 定时器0 CH0

		001: 定时器0 CH1 010: 定时器0 CH2 011: 定时器1 CH1 100: 定时器2 TRGO 101: 定时器3 CH3 110: 中断线11/定时器7 TRGO 111: 软件触发
16	保留	必须保持复位值
15	ETEIC	注入组外部触发使能 0: 注入组外部触发禁止 1: 注入组外部触发使能
14:12	ETSIC[2:0]	注入组外部触发选择 对于 ADC0与ADC1: 000: 定时器0 TRGO 001: 定时器0 CH3 010: 定时器1 TRGO 011: 定时器1 CH0 100: 定时器2 CH3 101: 定时器3 TRGO 110: 中断线15/定时器7 CH3 111: 软件触发
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 软件置位，在校准寄存器初始化后该位硬件清零。 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: 禁止连续模式

1: 使能连续模式

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

12.7.4. 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						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	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
23:21	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

12.7.5. 采样时间寄存器 1 (ADC_SAMPT1)

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

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

位/位域	名称	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

12.7.6. 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。

12.7.7. 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。

12.7.8. 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。

12.7.9. 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		

位/位域	名称	描述
31:24	保留	必须保持复位值
23:20	RL[3:0]	规则通道序列长度 规则通道转换序列中的总的通道数目为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

12.7.10. 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]					
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]					

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

12.7.11. 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]					
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]					

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..17)写入这些位来选择规则通道的第n个转换的通道

12.7.12. 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]				
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					

位/位域	名称	描述
31:22	保留	必须保持复位值
21:20	IL[1:0]	注入通道组长度 注入组总的通道数目为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..17)写入这些位来选择注入组的第n个转换的通道 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 3 ISQ0 >> ISQ1 >> ISQ2 >> ISQ3 2 ISQ1 >> ISQ2 >> ISQ3 1 ISQ2 >> ISQ3 0 ISQ3

12.7.13. 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
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

IDATA_n[15:0]

r

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

12.7.14. 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
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

ADC1RDTR[15:0]

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	ADC1RDTR[15:0]	ADC1规则通道数据 ADC0: 在同步模式下，这些位包含着ADC1的规则通道数据 ADC1: 这些位没有使用。
15:0	RDATA[15:0]	规则通道数据 这些位包含了规则通道的转换结果，只读。

12.7.15. 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

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

保留	DRES[1:0]	保留	TOVS	OVSS[3:0]	OVSR[2:0]	保留	OVSEN
----	-----------	----	------	-----------	-----------	----	-------

rw rw rw rw rw

位/位域	名称	描述
31:14	保留	必须保持复位值
13:12	DRES[1:0]	ADC分辨率 00: 12位 01: 10位 10: 8位 11: 6位
11:10	保留	必须保持复位值
9	TOVS	触发过滤采样 该位通过软件设置和清除. 0: 在一次触发后连续执行过采样通道的所有转换 1: 对于过采样通道的每次转换都需要一次触发, 触发次数由过采样率 (OVSR[2:0]) 决定。 注意: 当ADCON=0时软件才允许写该位(确定没有转换正在进行)。
8:5	OVSS[3:0]	过滤采样移位 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 其他保留 注意: 当ADCON=0时软件才允许写该位(确定没有转换正在进行)。
4:2	OVSR[2:0]	过采样率 这些位定义了过采样率的大小. 000: 2x 001: 4x 010: 8x 011: 16x 100: 32x 101: 64x 110: 128x 111: 256x 注意: 当ADCON=0时软件才允许写该位(确定没有转换正在进行)。

1	保留	必须保持复位值
0	OVSEN	<p>过滤采样使能 该位通过软件和设置和清除 0: 过滤采样失能 1: 过滤采样使能 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>

13. 数模转换器 (DAC)

13.1. 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 可被用于更新输入端数字数据。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。

两个 DAC 可以独立或并发工作。

13.2. 主要特征

DAC 的主要特征如下：

- 8 位或 12 位分辨率，数据左对齐或右对齐；
- 每个通道带有 DMA 功能；
- 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_{REF+} ；
- 噪声波生成(LSFR 噪声模式和三角噪声模式)；
- 双 DAC 并发模式。

[图 13-1. DAC 结构框图](#) 为 DAC 的结构框图，[表 13-1. DAC 引脚](#) 给出了引脚描述。

图 13-1. DAC 结构框图

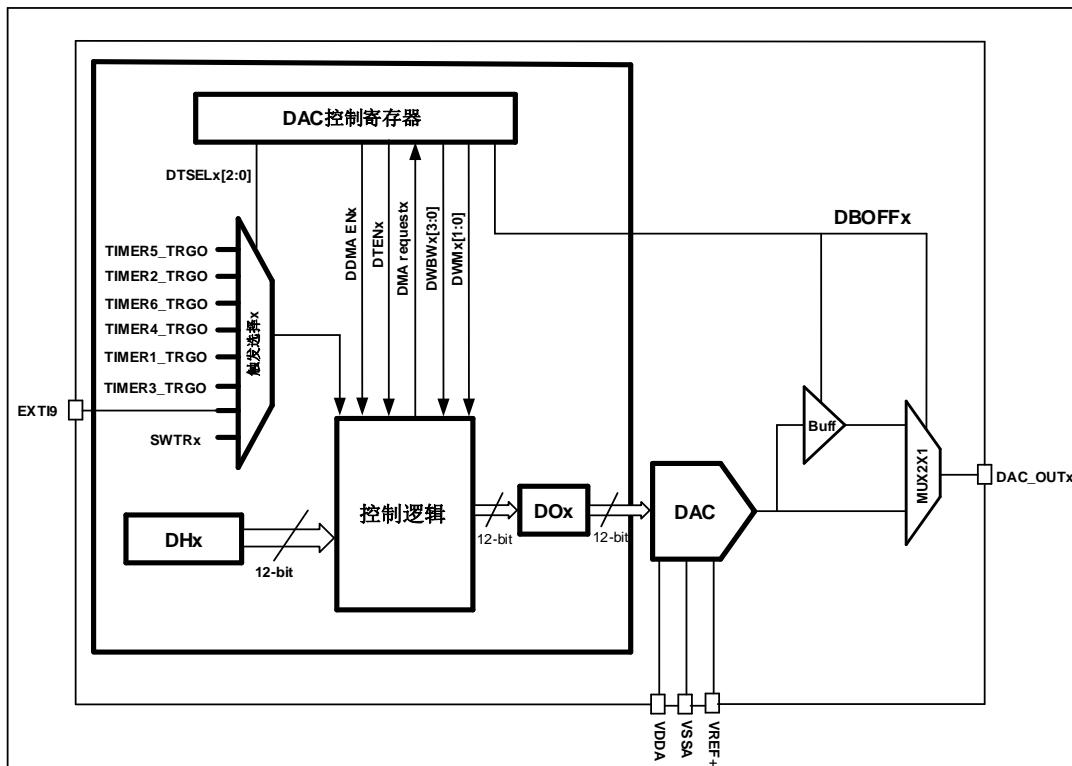


表 13-1. DAC 引脚

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
V _{REF+}	DAC 正参考电压, $2.4\text{ V} \leq V_{\text{REF}+} \leq V_{\text{DDA}}$	输入, 模拟正参考电压
DAC_OUTx	DACx 模拟输出	模拟输出信号

在使能 DAC 模块前, GPIO 口 (PA4 对应 DAC0, PA5 对应 DAC1) 应配置为模拟模式。

13.3. 功能说明

13.3.1. DAC 使能

将 DAC_CTL 寄存器中的 DENx 位置 1 可以给 DAC 上电, DAC 子模块完全启动需要等待 t_{WAKEUP}。

13.3.2. 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, 每个 DAC 模块内部各自集成了一个输出缓冲区。

缺省情况下, 输出缓冲区是开启的, 可以通过设置 DAC_CTL 寄存器的 DBOFFx 位来开启或者关闭缓冲区。

13.3.3. DAC 数据配置

对于 12 位的 DAC 保持数据 (DACx_DH), 可以通过对 DACx_R12DH、DACx_L12DH 和 DACx_R8DH 中的任意一个寄存器写入数据来配置。当数据被写入到 DACx_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。

13.3.4. DAC 触发

通过设置 DAC_CTL 寄存器中 DTENx 位来使能 DAC 外部触发。触发源可以通过 DAC_CTL 寄存器中 DTSELx 位来对外部触发信号进行选择。如 [表 13-2. DAC 外部触发](#) 所示。

表 13-2. 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	TIMER6_TRGO	
011	TIMER4_TRGO	
100	TIMER1_TRGO	
101	TIMER3_TRGO	

DTSELx[2:0]	触发源	触发类型
110	EXTI9	外部信号
111	SWTRIG	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 SWTRx 位生成的。

13.3.5. DAC 转换

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

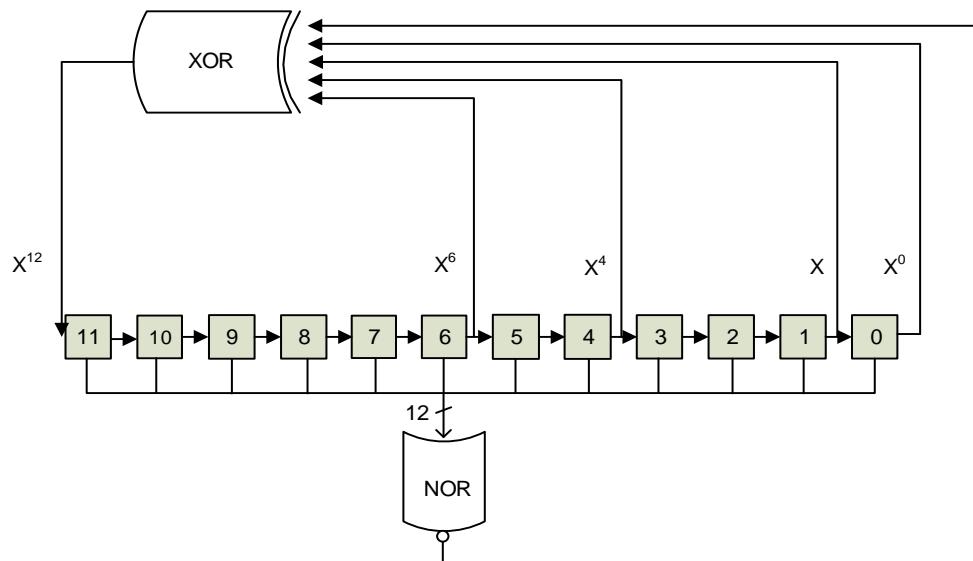
当 DAC 保持数据（DACx_DH）加载到 DACx_DO 寄存器时，经过 $t_{SETTLING}$ 时间之后，模拟输出变得有效， $t_{SETTLING}$ 的值与电源电压和模拟输出负载有关。

13.3.6. 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 DWMx 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（DWBWx）位来进行设置。

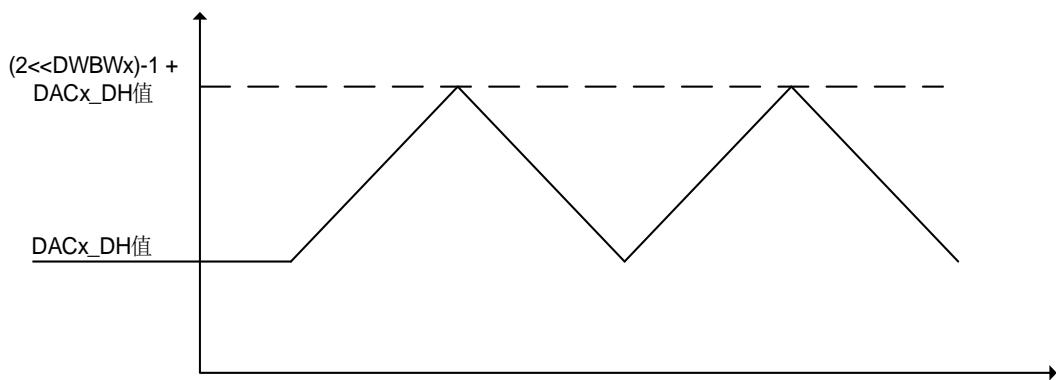
LFSR 噪声模式：在 DAC 控制逻辑中有一个线性反馈移位寄存器（LFSR）。在此模式下，LFSR 的值与 DACx_DH 值相加后，被写入到 DAC 数据输出寄存器（DACx_DO）。当配置的 DAC 噪声波位宽小于 12 时，LFSR 的值等于 LFSR 寄存器最低的 DWBWx 位，DWBWx 位决定了不屏蔽 LFSR 的哪些位。

图 13-2. DAC LFSR 算法



三角噪声模式：三角波信号与 DACx_DH 值相加后，被写入到 DAC 数据输出寄存器（DACx_DO）。三角波信号的最小值为 0，最大值为 $(2 << DWBWx) - 1$ 。

图 13-3. DAC 三角噪声模式生成的波形



13.3.7. DAC 输出电压

DAC 引脚上的模拟输出电压取决于下面的公式：

$$DAC_{output} = V_{REF+} * DAC_DO / 4096 \quad (13-1)$$

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_{REF+} 。

13.3.8. DMA 请求

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

13.3.9. DAC 并发转换

当两个 DAC 同时工作时，为了在特定应用中最大限度利用总线带宽，两个 DAC 可以被配置为并发模式。在并发模式中，两个 DAC 的 DACx_DO 的值将同时被更新。

有 3 个寄存器可被用于加载 DACx_DH 的值，分别是：DACC_R8DH、DACC_R12DH 和 DACC_L12DH。配置其中的一个寄存器来实现同时驱动两个 DAC。

当使能了外部触发时，两个 DAC 模块的 DTENx 位都应被置位。DTSEL0 和 DTSEL1 位应被配置为相同的值。

当使能了 DMA 功能时，任一 DAC 的 DDMAENx 位被置位即可。

噪声模式和噪声位宽可以根据使用情况配置为相同或不同。

13.4. DAC 寄存器

DAC 基地址: 0x4000 7400

13.4.1. 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	DDMAEN1		DWBW1[3:0]		DWM1[1:0]		DTSEL1[2:0]		DTEN1	DBOFF1	DEN1				
	rw		rw		rw		rw		rw	rw	rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	DDMAEN0		DWBW0[3:0]		DWM0[1:0]		DTSEL0[2:0]		DTEN0	DBOFF0	DEN0				
	rw		rw		rw		rw		rw	rw	rw				

位/位域	名称	描述
31:29	保留	必须保持复位值
28	DDMAEN1	DAC1 DMA使能 0: DAC1 DMA模式失能 1: DAC1 DMA模式使能
27:24	DWBW1[3:0]	DAC1 噪声波位宽 这些位指定了 DAC1 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
23:22	DWM1[1:0]	DAC1 噪声波模式 这些位指定了在 DAC1 外部触发使能(DTEN1=1)的情况下，DAC1 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
21:19	DTSEL1[2:0]	DAC1触发选择 这些位用于在DAC1外部触发使能(DTEN1=1)的情况下，DAC1外部触发的选择。 000: Timer 5 TRGO 001: Timer 2 TRGO 010: Timer 6 TRGO 011: Timer 4 TRGO 100: Timer 1 TRGO 101: Timer 3 TRGO 110: 外部中断线9 111: 软件触发
18	DTEN1	DAC1触发使能 0: DAC1触发失能 1: DAC1触发使能
17	DBOFF1	DAC1输出缓冲区关闭 0: DAC1输出缓冲区打开,以降低输出阻抗, 提高驱动能力 1: DAC1输出缓冲区关闭
16	DEN1	DAC1使能 0: DAC1失能 1: DAC1使能
15:13	保留	必须保持复位值
12	DDMAEN0	DAC0 DMA使能 0: DAC0 DMA模式失能 1: DAC0 DMA模式使能
11:8	DWBW0[3:0]	DAC0 噪声波位宽 这些位指定了 DAC0 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	DWM0[1:0]	DAC0 噪声波模式 这些位指定了在 DAC0 外部触发使能(DTEN0=1)的情况下，DAC0 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	DTSEL0[2:0]	DAC0触发选择 这些位用于在DAC0外部触发使能(DTEN0=1)的情况下，DAC0外部触发的选择。 000: Timer 5 TRGO 001: Timer 2 TRGO 010: Timer 6 TRGO 011: Timer 4 TRGO 100: Timer 1 TRGO 101: Timer 3 TRGO 110: 外部中断线9 111: 软件触发
2	DTEN0	DAC0触发使能 0: DAC0触发失能 1: DAC0触发使能
1	DBOFF0	DAC0输出缓冲区关闭 0: DAC输出缓冲区打开,以降低输出阻抗, 提高驱动能力 1: DAC0输出缓冲区关闭
0	DEN0	DAC0使能 0: DAC0失能 1: DAC0使能

13.4.2. 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															

w w

位/位域	名称	描述
31:2	保留	必须保持复位值
1	SWTR1	DAC1软件触发，由硬件清除 0: 软件触发失能 1: 软件触发使能
0	SWTR0	DAC0软件触发，由硬件清除 0: 软件触发失能 1: 软件触发使能

13.4.3. DAC0 12 位右对齐数据保持寄存器 (DAC0_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															DAC0_DH[11:0]
rw															

位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	DAC0_DH[11:0]	DAC0 12位右对齐数据 这些位指定了将由DAC0转换的数据。

13.4.4. DAC0 12 位左对齐数据保持寄存器 (DAC0_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
															DAC0_DH[11:0]
rw															

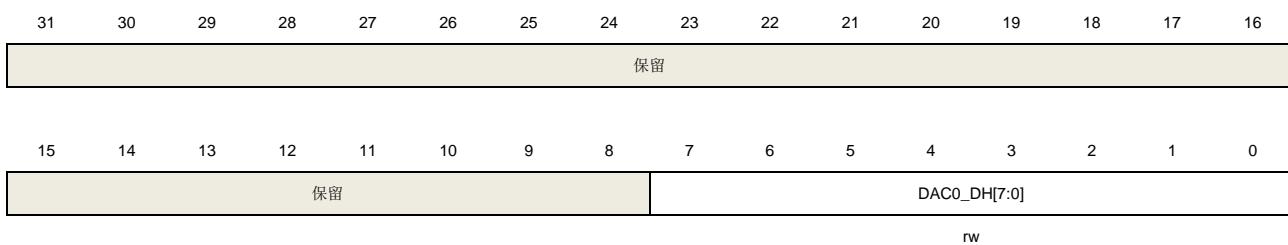
位/位域	名称	描述
31:16	保留	必须保持复位值
15:4	DAC0_DH[11:0]	DAC0 12位左对齐数据 这些位指定了将由DAC0转换的数据。
3:0	保留	必须保持复位值

13.4.5. DAC0 8 位右对齐数据保持寄存器 (DAC0_R8DH)

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器只能按字(32位)访问



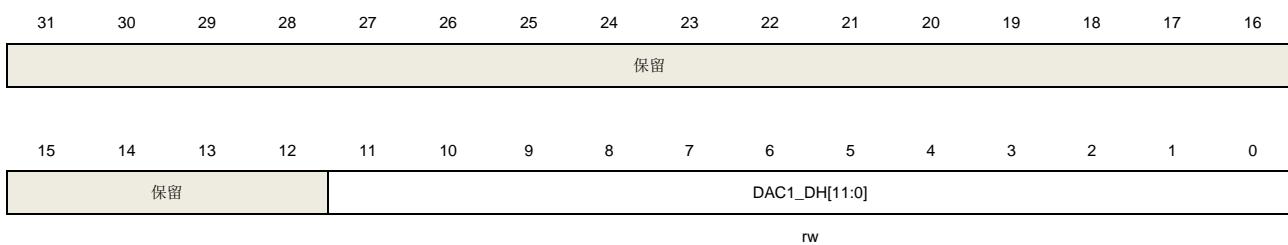
位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	DAC0_DH[7:0]	DAC0 8位右对齐数据 这些位指定了将由DAC0转换的数据的最高8位有效位。

13.4.6. DAC1 12 位右对齐数据保持寄存器 (DAC1_R12DH)

地址偏移: 0x14

复位值: 0x0000 0000

该寄存器只能按字(32位)访问



位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	DAC1_DH[11:0]	DAC1 12位右对齐数据

这些位指定了将由DAC1转换的数据。

13.4.7. DAC1 12 位左对齐数据保持寄存器 (DAC1_L12DH)

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
DAC1_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	DAC1_DH[11:0]	DAC1 12位左对齐数据 这些位指定了将由DAC1转换的数据。
3:0	保留	必须保持复位值

13.4.8. DAC1 8 位右对齐数据保持寄存器 (DAC1_R8DH)

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
保留								DAC1_DH[7:0]							

rw

位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	DAC1_DH[7:0]	DAC1 8位右对齐数据 这些位指定了将由 DAC1 转换的数据的 8 位最高有效位。

13.4.9. DAC 并发模式 12 位右对齐数据保持寄存器 (DACC_R12DH)

地址偏移: 0x20

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				DAC1_DH[11:0]											
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				DAC0_DH[11:0]											
rw															

位/位域	名称	描述
31:28	保留	必须保持复位值
27:16	DAC1_DH[11:0]	DAC1 12 位右对齐数据 这些位指定了将由 DAC1 转换的数据。
15:12	保留	必须保持复位值
11:0	DAC0_DH[11:0]	DAC0 12 位右对齐数据 这些位指定了将由 DAC0 转换的数据。

13.4.10. DAC 并发模式 12 位左对齐数据保持寄存器 (DACC_L12DH)

地址偏移: 0x24

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
DAC1_DH[11:0]								保留							
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAC0_DH[11:0]								保留							
rw															

位/位域	名称	描述
31:20	DAC1_DH[11:0]	DAC1 12 位左对齐数据 这些位指定了将由 DAC1 转换的数据。
19:16	保留	必须保持复位值
15:4	DAC0_DH[11:0]	DAC0 12 位左对齐数据 这些位指定了将由 DAC0 转换的数据。

3:0 保留 必须保持复位值

13.4.11. DAC 并发模式 8 位右对齐数据保持寄存器 (DACC_R8DH)

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
DAC1_DH [7:0]								DAC0_DH [7:0]							

rw

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:8	DAC1_DH[7:0]	DAC1 8 位右对齐数据 这些位指定了将由 DAC1 转换的数据的 8 位最高有效位。
7:0	DAC0_DH[7:0]	DAC0 8 位右对齐数据 这些位指定了将由 DAC0 转换的数据的 8 位最高有效位。

13.4.12. DAC0 数据输出寄存器 (DAC0_DO)

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								DAC0_DO [11:0]							

r

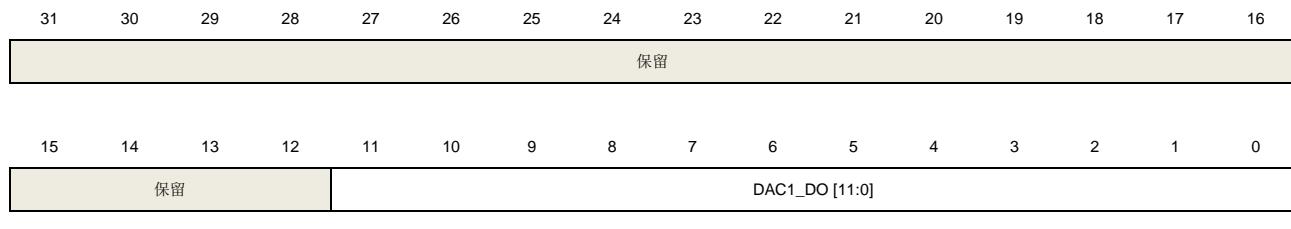
位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	DAC0_DO [11:0]	DAC0 数据输出 这些位为只读类型，存储由 DAC0 转换的数据。

13.4.13. DAC1 数据输出寄存器 (DAC1_DO)

地址偏移: 0x30

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问



位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	DAC1_DO [11:0]	DAC1 数据输出 这些位为只读类型，存储由 DAC1 转换的数据。

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，独立看门狗定时器 (FWDGT) 和窗口看门狗定时器 (WWDGT)。它们使用灵活，并提供了很高的安全水平和精准的时间控制。两个看门狗定时器都是用来解决软件故障问题的。

看门狗定时器在内部计数值达到了预设的门限的时候，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. 独立看门狗定时器 (FWDGT)

14.1.1. 简介

独立看门狗定时器(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，独立看门狗会产生一个复位。使能独立看门狗的寄存器写保护功能可以避免寄存器的值被意外的配置篡改。

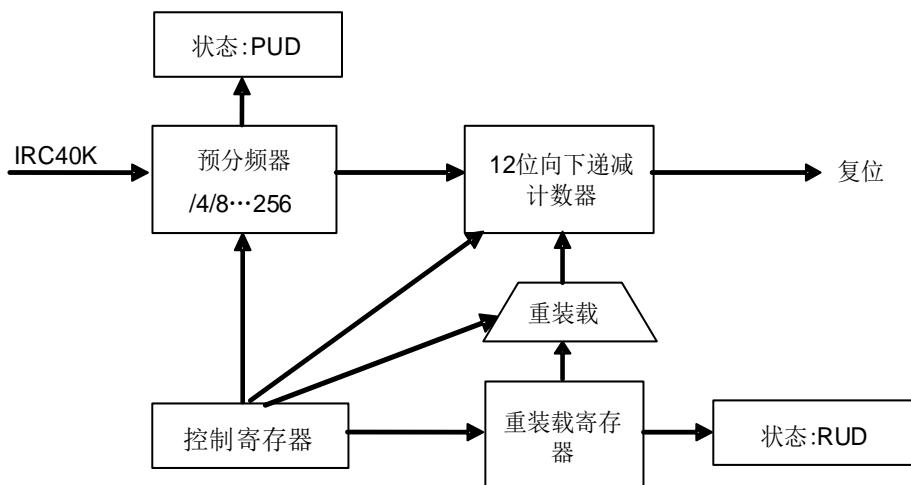
14.1.2. 主要特征

- 自由运行的12位向下计数器；
- 如果看门狗定时器被使能，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，独立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；
- 独立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独立看门狗定时器；
- 可以配置独立看门狗定时器在调试模式下选择停止还是继续工作。

14.1.3. 功能说明

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

图 14-1. 独立看门狗定时器框图



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

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寄存器相应的状态位被置1。

如果在MCU调试模块中的FWDGT_HOLD位被清0，即使Cortex™-M3内核停止(调试模式下)独立看门狗定时器依然工作。如果FWDGT_HOLD位被置1，独立看门狗定时器将在调试模式下停止工作。

表 14-1. 独立看门狗定时器在 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)	
		RLD[11:0]=0x000	RLD[11:0]=0xFFFF
1/4	000	0.025	409.525
1/8	001	0.025	819.025
1/16	010	0.025	1638.025
1/32	011	0.025	3276.025
1/64	100	0.025	6552.025
1/128	101	0.025	13104.025
1/256	110 or 111	0.025	26208.025

通过IRC40K校准可以使独立看门狗定时器超时更加精确。

注意：当执行完喂狗reload操作之后，如需要立即进入deepsleep/standby模式时，必须通过软

件设置，在reload命令及deepsleep/standby模式命令中间插入（3个以上）IRC40K时钟间隔。

14.1.4. FWDGT 寄存器

FWDGT 基地址: 0x4000 3000

控制寄存器 (FWDGT_CTL)

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器可以按半字 (16 位) 或字 (32 位) 访问

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]															

w

位/位域	名称	描述
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和FWDGT_RLD的写保护 0xCCCC: 开启独立看门狗定时器计数器。计数减到0时产生复位 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														PSC[2:0]	

rw

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

001: 1/8
 010: 1/16
 011: 1/32
 100: 1/64
 101: 1/128
 110: 1/256
 111: 1/256

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

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	描述
31:12	保留	必须保持复位值
11:0	RLD[11:0]	<p>独立看门狗定时器计数器重装载值，向FWDGT_CTL寄存器写入0xFFFF的时候，这个值会被更新到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，改变重装载值之前必须等到RUD位被清0。更新了重装载寄存器的值后，在代码持续执行之前不必等待RUD值被清零（在进入省电模式前需等待RUD值清零）。</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														RUD	PUD
r r															

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

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，7位向下递减计数值逐渐减小。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。

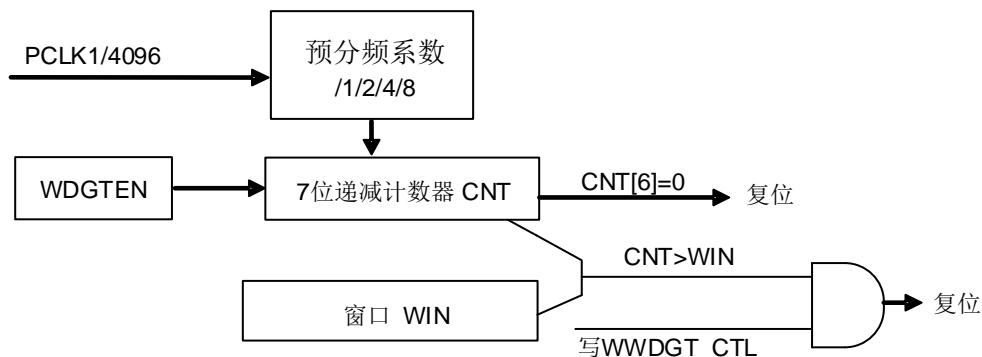
14.2.2. 主要特征

- 可编程的7位自由运行向下递减计数器。
- 当窗口看门狗使能后，有以下两种情况会产生复位：
 - 当计数器达到0x3F时产生复位；
 - 当计数器的值大于窗口寄存器的值时，更新计数器会产生复位。
- 提前唤醒中断(EWI)：如果看门狗定时器打开，中断允许，计数值达到0x40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. 功能说明

如果窗口看门狗定时器使能(将WWDGT_CTL寄存器的WDGTEN位置1)，计数值达到0x3F的时候产生复位(CNT[6]位被清0)。或者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



上电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

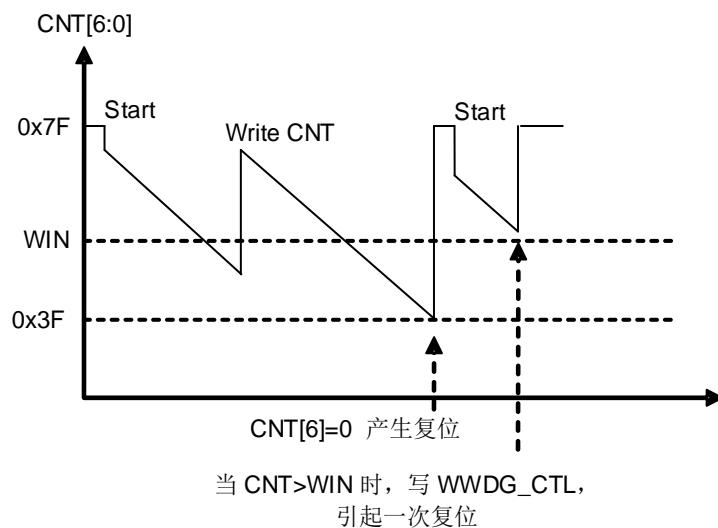
减速度取决于APB1时钟和预分频器(WWDGT_CFG寄存器的PSC[1:0]位)。

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，否则在其他时候进行重加载就会引起复位。

对WWDGT_CFG寄存器的EWIE位置1可以使能提前唤醒中断(EWI)，当计数值达到0x40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：

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

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60MHz \(f_{PCLK1}\)时的最大/最小超时值](#)。

表 14-2. 在 60MHz (f_{PCLK1})时的最大/最小超时值

预分频系数	PSC[1:0]	最小超时	最大超时
		CNT[6:0] =0x40	CNT[6:0]=0x7F
1/1	00	68.2 μs	4.3ms
1/2	01	136.4 μs	8.6 ms
1/4	10	272.8μs	17.2 ms
1/8	11	545.6 μs	34.4 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WWDGT_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

WWDGT 基地址: 0x4000 2C00

控制寄存器 (WWDGT_CTL)

地址偏移: 0x00

复位值: 0x0000 007F

该寄存器可以按半字 (16 位) 或字 (32 位) 访问

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								WDGTEN	CNT[6:0]						
rs								rw							

位/位域	名称	描述
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

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								EWIE	PSC[1:0]	WIN[6:0]						
rs								rw								

位/位域	名称	描述
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RCU模块

的WWDGT软件复位来清0。写0没有任何作用。

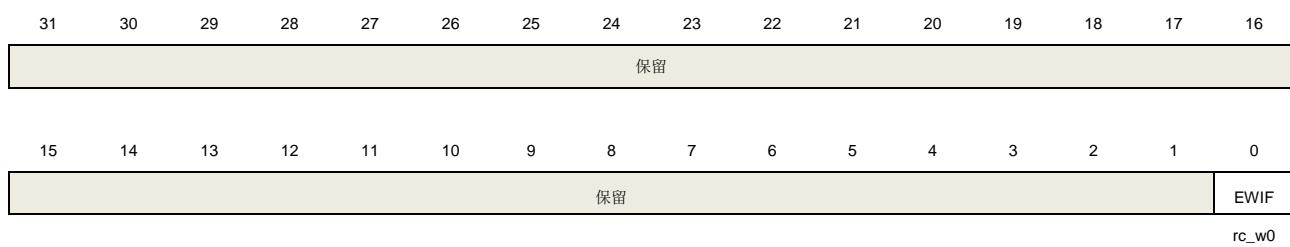
8:7	PSC[1:0]	预分频器，看门狗定时器计数器的时间基准 00: PCLK1/4096/1 01: PCLK1/4096/2 10: PCLK1/4096/4 11: PCLK1/4096/8
6:0	WIN[6:0]	窗口值，当看门狗定时器计数器的值大于窗口值时，写看门狗定时器计数器(WWDGT_CTL的CNT位)会产生复位。

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



位/位域	名称	描述
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bit可以通过写0清零，写1无效。

15. 实时时钟 (RTC)

15.1. 简介

实时时钟RTC通常用于日历时钟。RTC电路分属于两个电源域。一部分位于备份域中，该部分包括一个32位的累加计数器、一个闹钟、一个预分频器、一个分频器以及RTC时钟配置寄存器。这表明系统复位或者从待机模式唤醒时，RTC的设置和时间都保持不变。另一部分位于V_{DD}电源域中，该部分只包括APB接口以及一组控制寄存器。在本章接下来的部分，将详细介绍RTC的功能。

15.2. 主要特征

- 32位可编程计数器，用于计数运行时间
 - 可编程的预分频器：分频系数最高可达2²⁰
- 独立时钟域：
 - PCLK1时钟域
 - RTC时钟域（该时钟必须比PCLK1时钟至少慢4倍）
- RTC时钟源：
 - HXTAL时钟除以128
 - LXTAL振荡电路时钟
 - IRC40K振荡电路时钟
- 可屏蔽的中断源：
 - 闹钟中断
 - 秒中断
 - 溢出中断

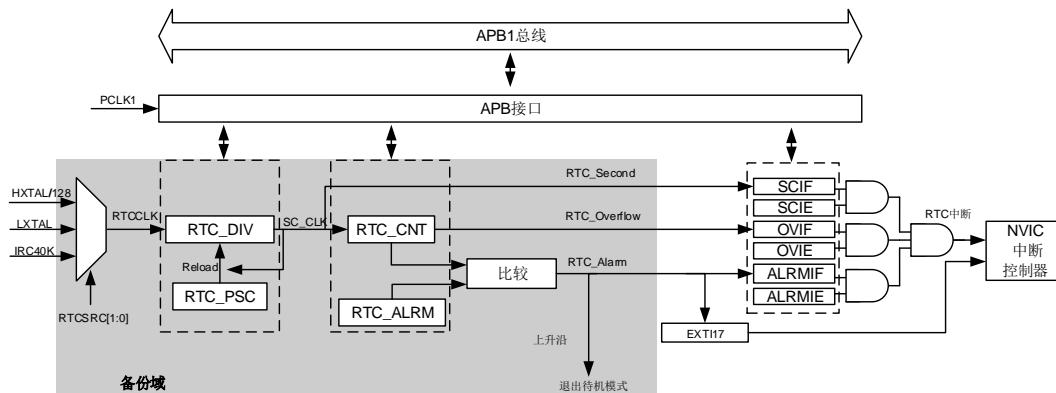
15.3. 功能说明

RTC电路包含两个主要部分，位于PCLK1时钟域的APB接口和位于RTC时钟域的RTC内核。

APB接口与APB1总线相连，包含一组寄存器，通过APB1总线可以对其进行读或写。

RTC内核包含两个主要模块。一个是RTC预分频模块，用来产生RTC时间基准SC_CLK，RTC预分频模块包含一个20位可编程分频器（RTC预分频器）。该分频器可以通过对RTC时钟源分频产生SC_CLK。如果对RTC_INTEN寄存器中秒中断位进行使能，RTC会在每个SC_CLK上升沿产生一个秒中断。另外一个模块是一个32位可编程计数器，其数值可以被初始化为当前系统时间。如果对RTC_INTEN寄存器的闹钟中断位进行使能，RTC会在系统时间等于闹钟时间（存储于RTC_ALRMH/L寄存器）时产生一个闹钟中断。

图 15-1. RTC 框图



15.3.1. RTC 复位

APB接口和RTC_INTEN寄存器会随着系统复位进行复位。RTC内核（预分频器、分频器、计数器以及闹钟）只会随备份域复位进行复位。

通过下面的步骤，可以在复位后访问备份域寄存器以及RTC寄存器：

1. 通过对RCU_APB1EN寄存器中的PMUEN和BKPN位进行置位，使能电源以及备份接口时钟。
2. 通过对PMU_CTL中的BKPWEN位进行置位，使能对备份域寄存器和RTC的访问。

15.3.2. RTC 读取

APB接口和RTC内核分属于两个不同电源域。

在RTC内核中，只有计数器和分频器寄存器为可读寄存器。这两个寄存器的值以及RTC标志会在每个RTC时钟的上升沿进行内部更新，并与APB1时钟进行重新同步。

当APB接口从禁用状态使能后，建议不要立即进行读操作，因为这些寄存器的首次内部更新可能尚未完成。这表明，在系统复位、电源复位、从待机/深度睡眠模式下唤醒时，APB接口是被禁用的，但是RTC内核仍然保持运行。在这类情况下，正确的读操作应该先将RTC_CTL寄存器的RSYNF清零并等待其被硬件置位。WFI和WFE指令对于RTC的APB接口没有影响。

15.3.3. RTC 配置

RTC内核中的RTC_PSC、RTC_CNT和RTC_ALARM寄存器都是可写的。只有在外设进入配置模式后，这些寄存器的值才能进行设置。通过查询RTC_CTL寄存器的CMF位，可以检测配置模式的状态。只有在外设退出配置模式后，之前对这些寄存器的写操作才能生效，且至少需要三个RTCCCLK周期才能完成。当写操作完成后，RTC_CTL寄存器中的LWOFF位的值变为‘1’。下一个写操作必须等待上次写操作完成之后才能进行。

配置过程如下：

1. 等待RTC_CTL寄存器中的LWOFF位的值变为1；

2. 通过设置RTC_CTL寄存器中的CMF位来进入配置模式;
3. 对RTC寄存器进行写操作;
4. 通过将RTC_CTL寄存器中的CMF清零来退出配置模式;
5. 等待RTC_CTL寄存器中的LWOFF位的值变为1。

15.3.4. RTC 标志位

RTC秒中断标志（SCIF）在RTC计数器更新之前的最后一个RTCCLK周期进行置位。

RTC闹钟中断标志（ALRMIF）在计数器达到存储于闹钟寄存器中的RTC闹钟值加1的前一个RTCCLK周期进行置位。

RTC溢出中断标志（OVIF）在计数器值达到0x00的前一个RTCCLK周期置位。

RTC闹钟的写操作需要按照下列方法之一来进行，以达到和秒中断标志保持同步：

- 启用RTC闹钟中断，在RTC内部中断服务程序内更新RTC闹钟寄存器及/或RTC计数器寄存器的内容。
- RTC闹钟寄存器及/或RTC计数器寄存器的内容必须等待RTC控制寄存器中SCIF置位后才能更新。

图 15-2. RTC 秒信号及闹钟信号的波形 (RTC_PSC = 3, RTC_ALARM = 2)

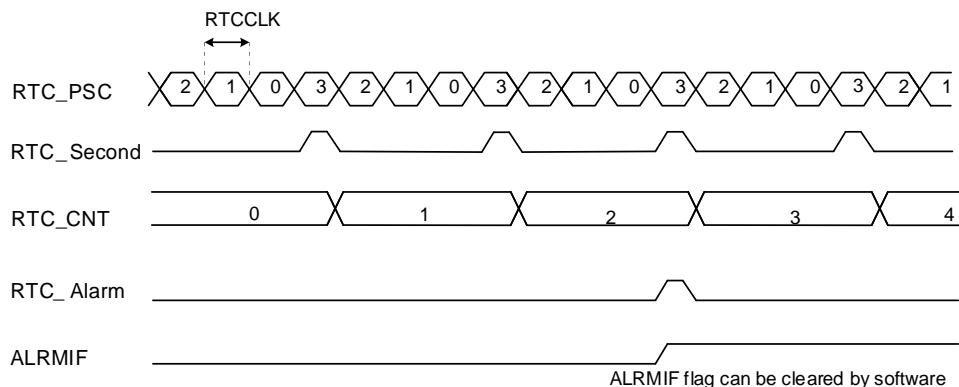
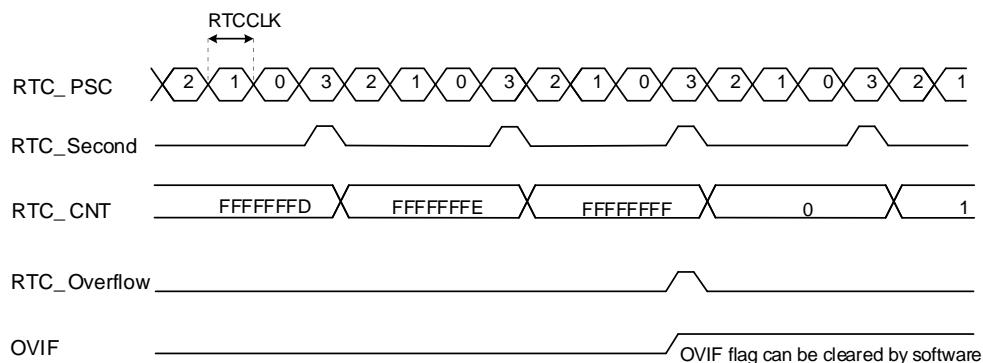


图 15-3. RTC 秒信号及溢出信号的波形(RTC_PSC = 3)



15.4. RTC 寄存器

RTC基地址: 0x4000 2800

15.4.1. RTC 中断使能寄存器 (RTC_INTEN)

偏移地址: 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														OVIE	ALRMIE	SCIE
													rw	rw	rw	

位/位域	名称	描述
15:3	保留	必须保持复位值
2	OVIE	溢出中断使能 0: 禁用溢出中断 1: 使能溢出中断
1	ALRMIE	闹钟中断使能 0: 禁用闹钟中断 1: 使能闹钟中断
0	SCIE	秒中断使能 0: 禁用秒中断 1: 使能秒中断

15.4.2. RTC 控制寄存器 (RTC_CTL)

偏移地址: 0x04

复位值: 0x0020

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														LWOFF	CMF	RSYNF	OVIF	ALRMIF	SCIF
													r	rw	rc_w0	rc_w0	rc_w0	rc_w0	

位/位域	名称	描述
15:6	保留	必须保持复位值
5	LWOFF	上次对RTC寄存器写操作标志 0: 上次对RTC寄存器写操作没有完成 1: 上次对RTC寄存器写操作已经完成

4	CMF	配置模式标志 0: 退出配置模式 1: 进入配置模式
3	RSYNF	寄存器同步标志 0: 寄存器没有与APB1时钟同步 1: 寄存器已经与APB1时钟同步
2	OVIF	溢出中断标志 0: 没有检测到溢出事件 1: 检测到溢出事件。当RTC_INTEN寄存器的OVIE位被置1, 中断发生。
1	ALRMIF	闹钟中断标志 0: 没有检测到闹钟事件 1: 检测到闹钟事件。当RTC_INTEN寄存器的ALRMIE位被置1, RTC全局中断发生。 并且当EXTI17被使能中断模式, 发生RTC闹钟中断。
0	SCIF	秒中断标志 0: 没有检测到秒事件 1: 检测到秒事件。当RTC_INTEN寄存器的SCIE位被置1, 中断发生。 当分频器重加载RTC_PSC值时, 硬件将该位置1, 从而累加RTC计数器。

15.4.3. RTC 预分频寄存器高位 (RTC_PSCH)

偏移地址: 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														PSC[19:16]	

w

位/位域	名称	描述
15:4	保留	必须保持复位值
3:0	PSC[19:16]	RTC预分频器高位值

15.4.4. RTC 预分频寄存器低位 (RTC_PSCL)

偏移地址: 0x0C

复位值: 0x8000

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]															

w

位/位域	名称	描述
15:0	PSC[15:0]	RTC预分频器低位值 SC_CLK的频率是RTCCLK的频率除以(PSC[19:0]+1)

15.4.5. RTC 分频器高位 (RTC_DIVH)

偏移地址: 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														DIV[19:16]	

r

位/位域	名称	描述
15:4	保留	必须保持复位值
3:0	DIV[19:16]	RTC分频器高位

15.4.6. RTC 分频器低位 (RTC_DIVL)

偏移地址: 0x14

复位值: 0x8000

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
DIV[15:0]														r	

位/位域	名称	描述
15:0	DIV[15:0]	RTC分频器低位 当RTC预分频寄存器或者RTC计数寄存器更新时，RTC分频器寄存器会由硬件自动加载

15.4.7. RTC 计数寄存器高位 (RTC_CNTH)

偏移地址: 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
CNT[31:16]														rw	

位/位域	名称	描述
15:0	CNT[31:16]	RTC计数寄存器高位

15.4.8. RTC 计数寄存器低位 (RTC_CNTL)

偏移地址: 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
CNT[15:0]															
rw															

位/位域	名称	描述
15:0	CNT[15:0]	RTC计数寄存器低位

15.4.9. RTC 闹钟寄存器高位 (RTC_ALRMH)

偏移地址: 0x20

复位值: 0xFFFF

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
ALRM[31:16]															
w															

位/位域	名称	描述
15:0	ALRM[31:16]	RTC闹钟值高位

15.4.10. RTC 闹钟寄存器低位 (RTC_ALRML)

偏移地址: 0x24

复位值: 0xFFFF

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
ALRM[15:0]															
w															

位/位域	名称	描述
15:0	ALRM[15:0]	RTC闹钟值低位

16. 定时器 (TIMER)

表 16-1. 定时器 (TIMERx) 分为五种类型

定时器	定时器 0/7	定时器 1/2/3/4	定时器 8/11	定时器 9/10/12/13	定时器 5/6
类型	高级	通用 (L0)	通用 (L1)	通用 (L2)	基本
预分频器	16 位	16 位	16 位	16 位	16 位
计数器	16 位	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
可重复性	•	✗	✗	✗	✗
捕获/比较 通道数	4	4	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 **IT10: TIMER4_TRGO** **IT11: TIMER1_TRGO** **IT12: TIMER2_TRGO** **IT13: TIMER3_TRGO**
 TIMER7 **IT10: TIMER0_TRGO** **IT11: TIMER1_TRGO** **IT12: TIMER3_TRGO** **IT13: TIMER4_TRGO**

TIMER1 **IT10: TIMER0_TRGO** **IT11: 0** **IT12: TIMER2_TRGO** **IT13: TIMER3_TRGO**

(2) TIMER2 **IT10: TIMER0_TRGO** **IT11: TIMER1_TRGO** **IT12: TIMER4_TRGO** **IT13: TIMER3_TRGO**

TIMER3 **IT10: TIMER0_TRGO** **IT11: TIMER1_TRGO** **IT12: TIMER2_TRGO** **IT13: TIMER7_TRGO**

TIMER4 **IT10: TIMER1_TRGO** **IT11: TIMER2_TRGO** **IT12: TIMER3_TRGO** **IT13: TIMER7_TRGO**

(3) TIMER8 **IT10: TIMER1_TRGO** **IT11: TIMER2_TRGO** **IT12: TIMER9_TRGO** **IT13: TIMER10_TRGO**

TIMER11 **IT10: TIMER3_TRGO** **IT11: TIMER4_TRGO** **IT12: TIMER12_TRGO** **IT13: TIMER13_TRGO**

(4) 只有更新事件可以产生 DMA 请求, 定时器 5 和定时 6 中没有 DMAS 位(DMA 请求源选择位)。

16.1. 高级定时器 (TIMERx,x=0,7)

16.1.1. 简介

高级定时器 (TIMER0和TIMER7) 是四通道定时器，支持输入捕获和输出比较。可以产生PWM信号控制电机和电源管理。高级定时器含有一个16位无符号计数器。

高级定时器是可编程的，可以用来计数，其外部事件可以驱动其他定时器。

高级定时器包含了一个死区时间插入模块，非常适合电机控制。

定时器和定时器之间是相互独立，但是它们的计数器可以被同步在一起形成一个更大的定时器。

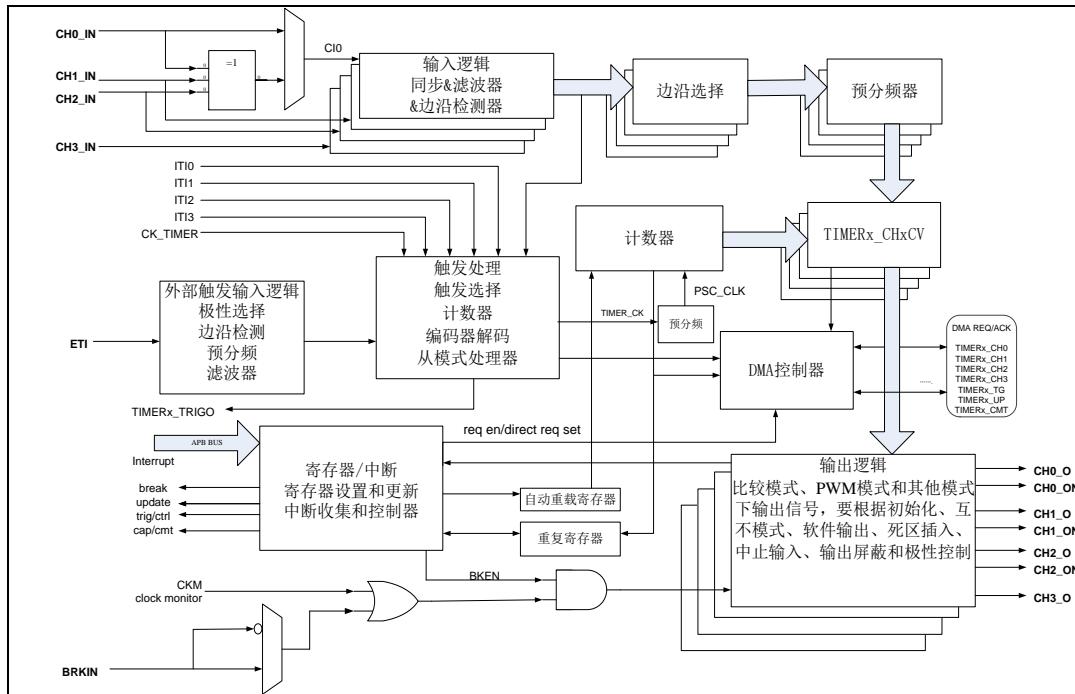
16.1.2. 主要特征

- 总通道数: 4;
- 计数器宽度: 16位;
- 定时器时钟源可选: 内部时钟, 内部触发, 外部输入, 外部触发;
- 多种计数模式: 向上计数, 向下计数和中央计数;
- 正交编码器接口: 用来追踪运动和分辨旋转方向和位置;
- 霍尔传感器接口: 用来做三相电机控制;
- 可编程的预分频器: 16位。运行时可以被改变;
- 每个通道可配置: 输入捕获模式, 输出比较模式, 可编程的PWM模式, 单脉冲模式;
- 可编程的死区时间;
- 自动重装载功能;
- 可编程的计数器重复功能;
- 中止输入功能;
- 中断输出和DMA请求: 更新事件, 触发事件, 比较/捕获事件和中止事件;
- 多个定时器的菊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. 功能说明

时钟源选择

高级定时器的时钟源可以是内部时钟源CK_TIMER，或者是由SMC (TIMERx_SMCFG寄存器位[2:0]) 位确定的时钟源。

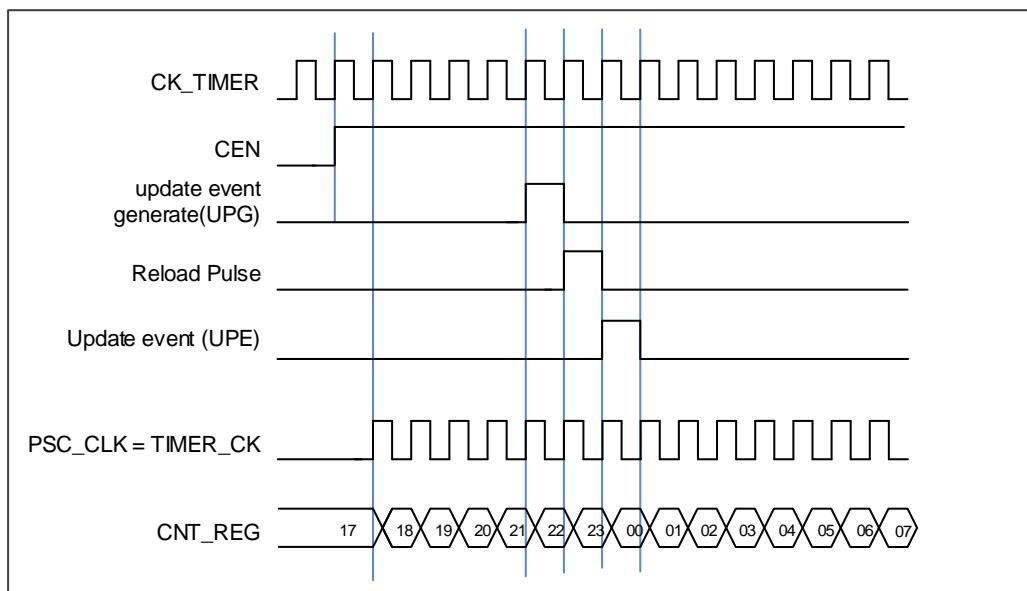
- SMC[2:0]=3'b000, 定时器选择内部时钟源 (连接到RCU模块的CK_TIMER)

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_CH0/TIMERx_CH1引脚的每个上升沿或下降沿计数。这种模式可以通过设置SMC[2:0]为0x7同时设置TRGS[2:0]为0x4, 0x5或0x6来选择。Clx是TIMERx_Clx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ITIO/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。

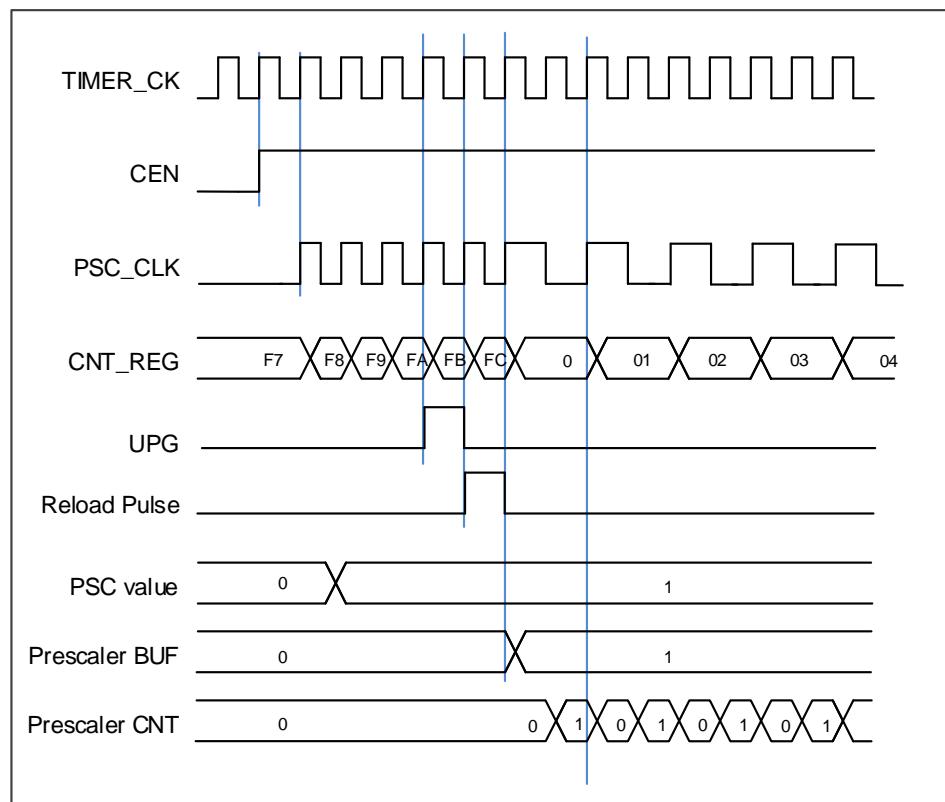
- SMC1=1'b1 (外部时钟模式1), 定时器选择外部输入引脚ETI作为时钟源。

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(重复计数寄存器，自动重载寄存器，预分频寄存器)都将被更新。

[图16-4. 向上计数时序图, PSC=0/1](#)和[图16-5. 向上计数时序图, 在运行时改变TIMERx_CAR寄存器的值](#)给出了一些例子，当`TIMERx_CAR=0x63`时，计数器在不同预分频因子下的行为。

图 16-4. 向上计数时序图, PSC=0/1

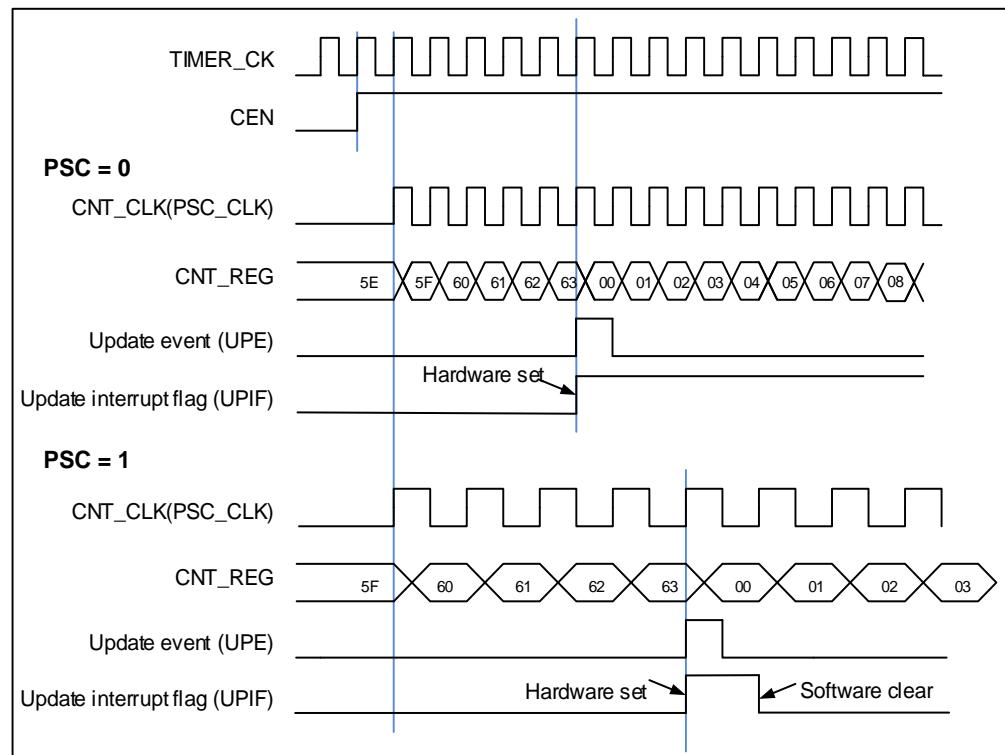
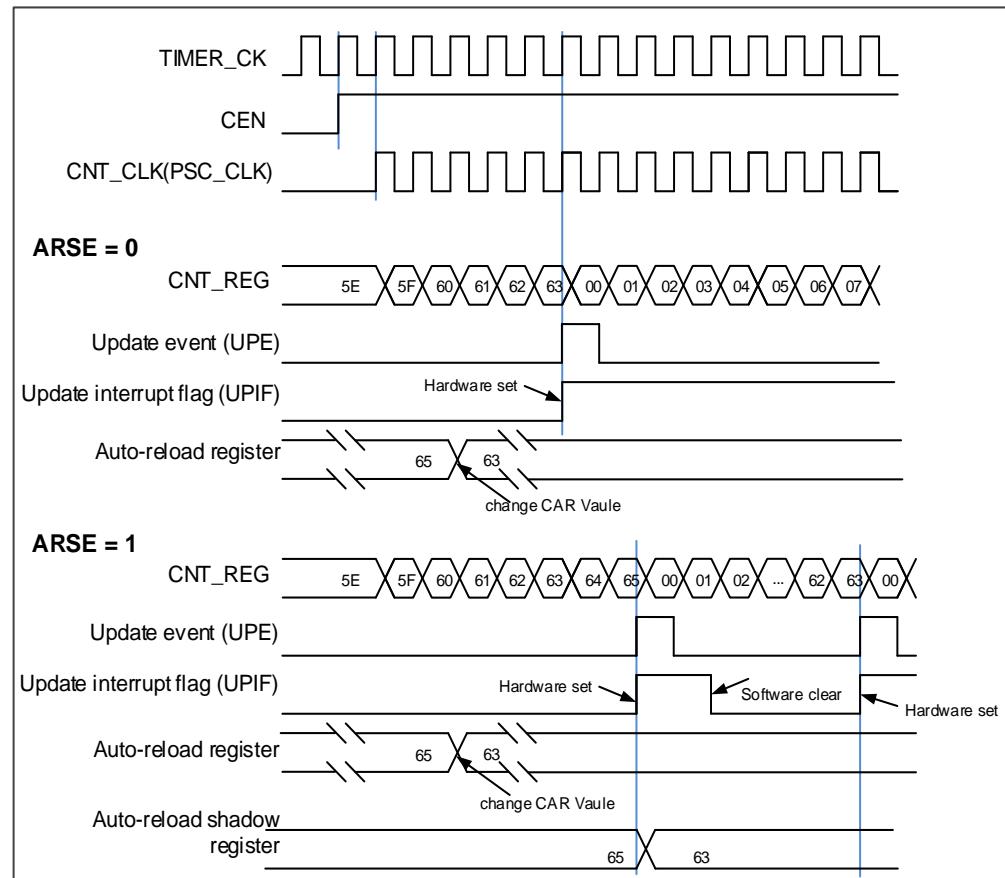


图 16-5. 向上计数时序图, 在运行时改变 TIMERx_CAR 寄存器的值



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。

[图16-6. 向下计数时序图, PSC=0/1](#)和[图16-7. 向下计数时序图, 在运行时改变TIMERx_CAR寄存器值](#)给出了一些例子，当TIMERx_CAR=0x63时，计数器在不同时钟频率下的行为。

图 16-6. 向下计数时序图, PSC=0/1

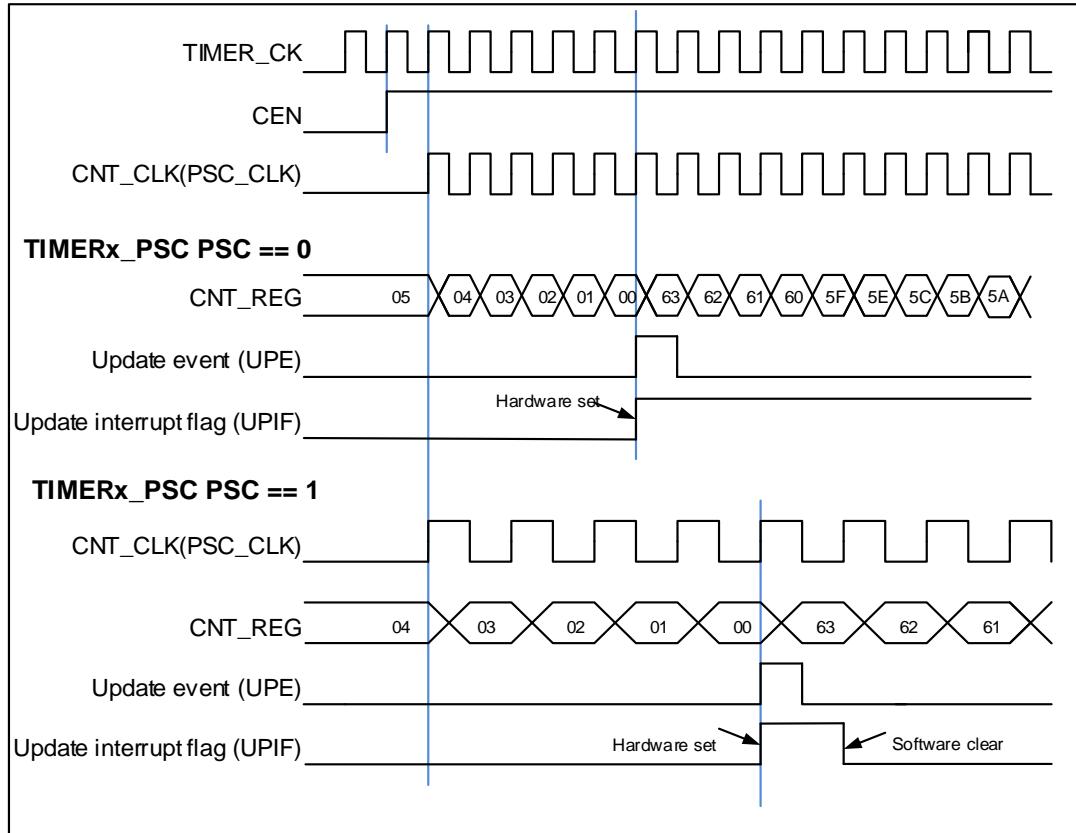
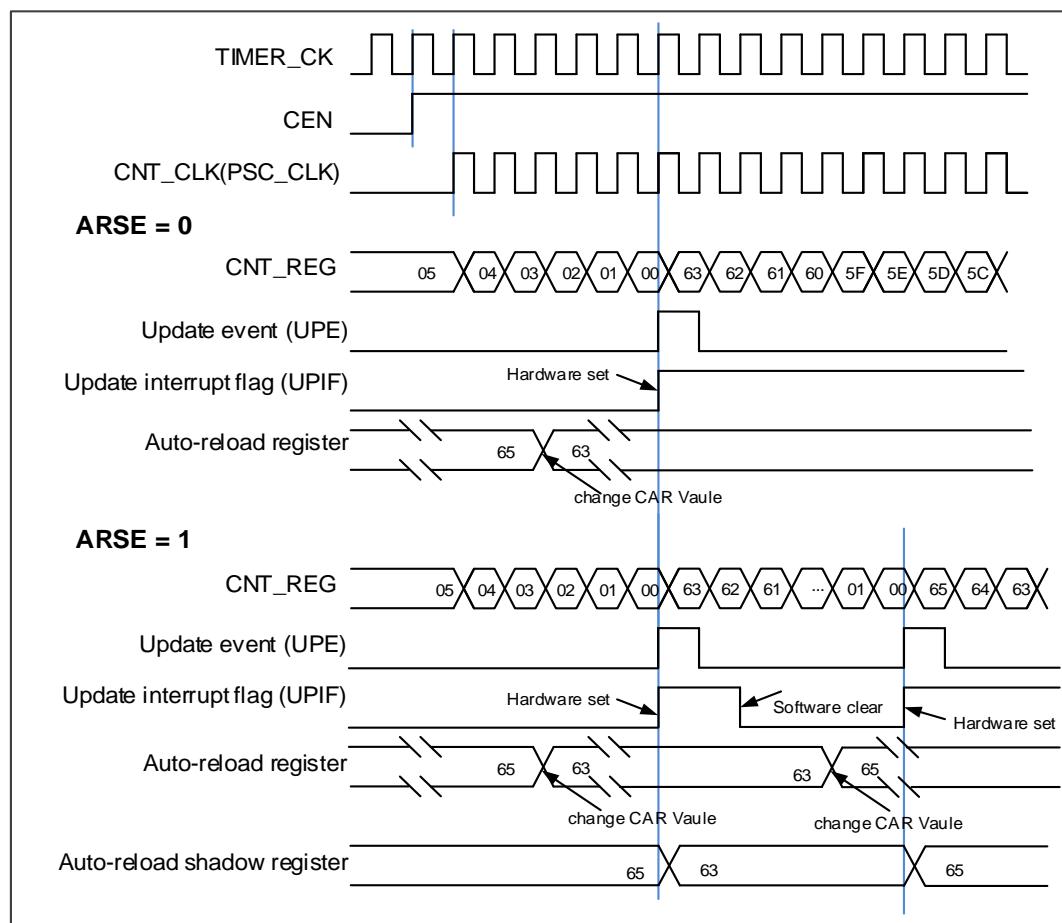


图 16-7. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值



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。向上计数模式中，定时器模块在计数器计数到 (TIMERx_CREP -1) 时产生一个上溢事件；向下计数模式中，定时器模块在计数器计数到1时产生一个下溢事件。在中央计数模式中，TIMERx_CTL0寄存器中的计数方向控制位DIR只读，表明计数方向。计数方向控制位DIR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。

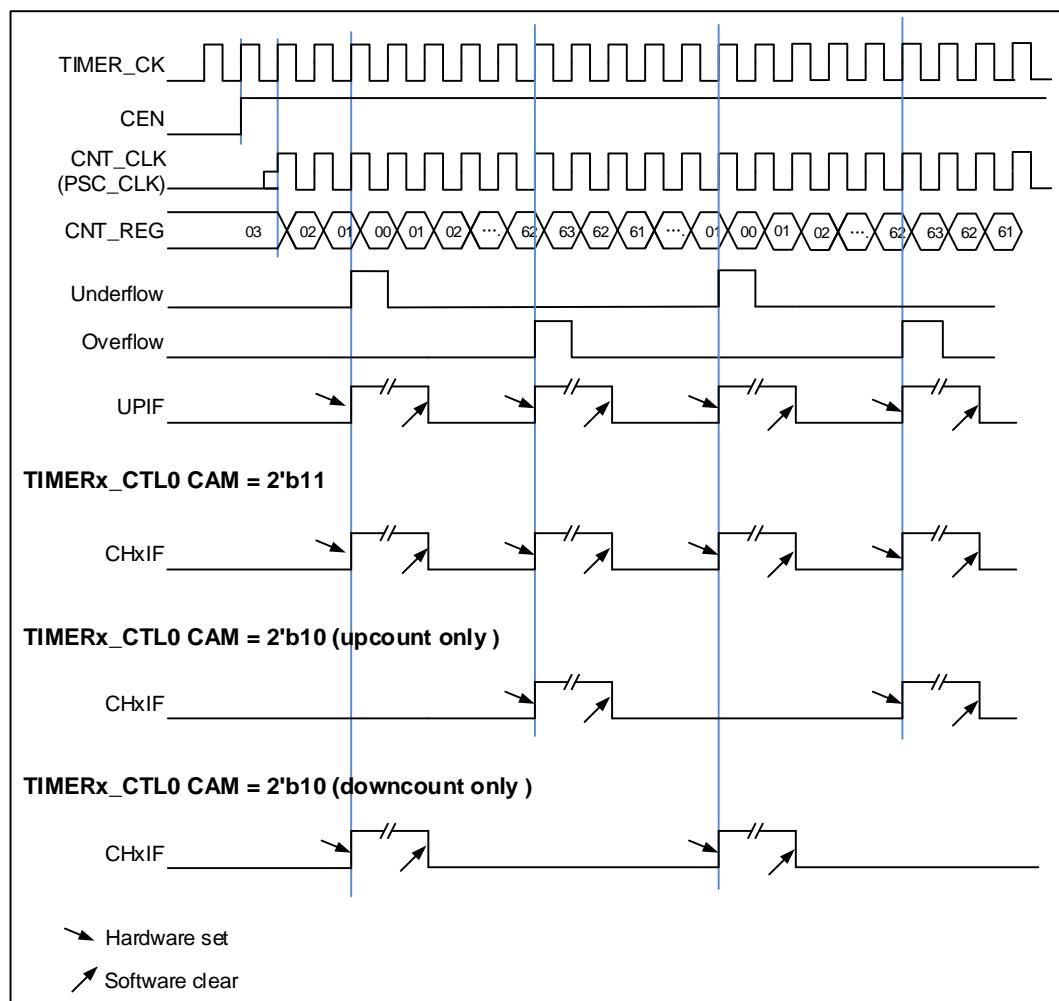
上溢或者下溢时，TIMERx_INTF寄存器中的UPIF位都会被置1。但是CHxIF位是否置1与TIMERx_CTL0寄存器中CAM的值有关。具体细节参考[图16-8. 中央计数模式计数器时序图](#)。

如果TIMERx_CTL0寄存器的UPDIS置1，则禁止更新事件。

当发生更新事件时，所有的寄存器(重复计数器，自动重载寄存器，预分频寄存器)都将被更新。

[图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，计数器的时序图。

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** 位的值。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

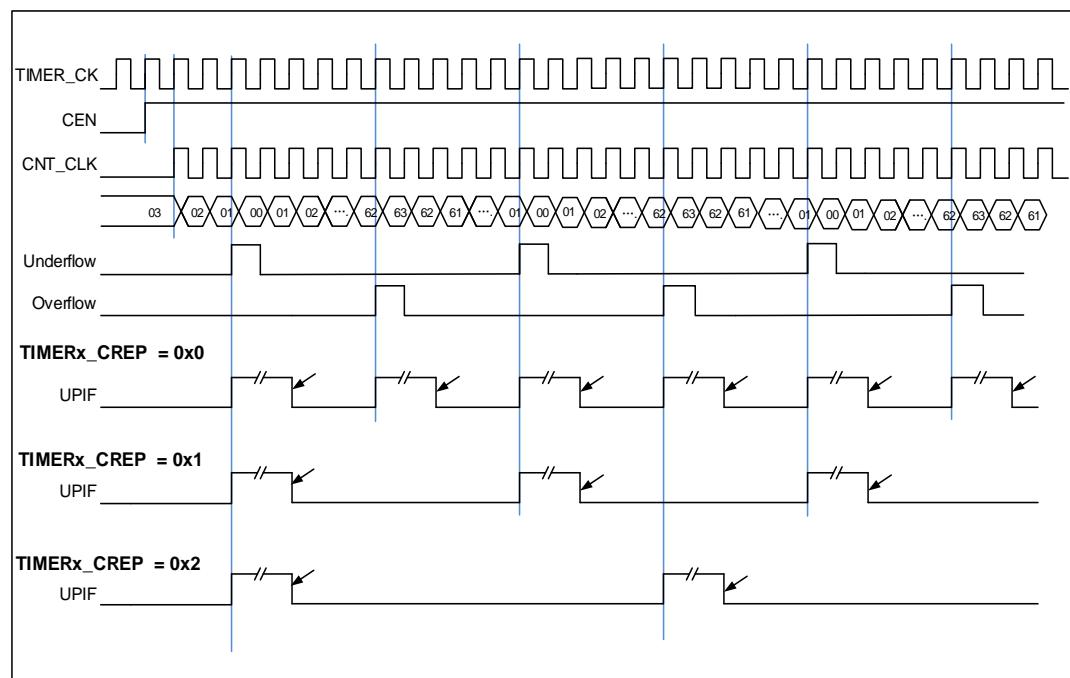


图 16-10. 在向上计数模式下计数器重复时序图

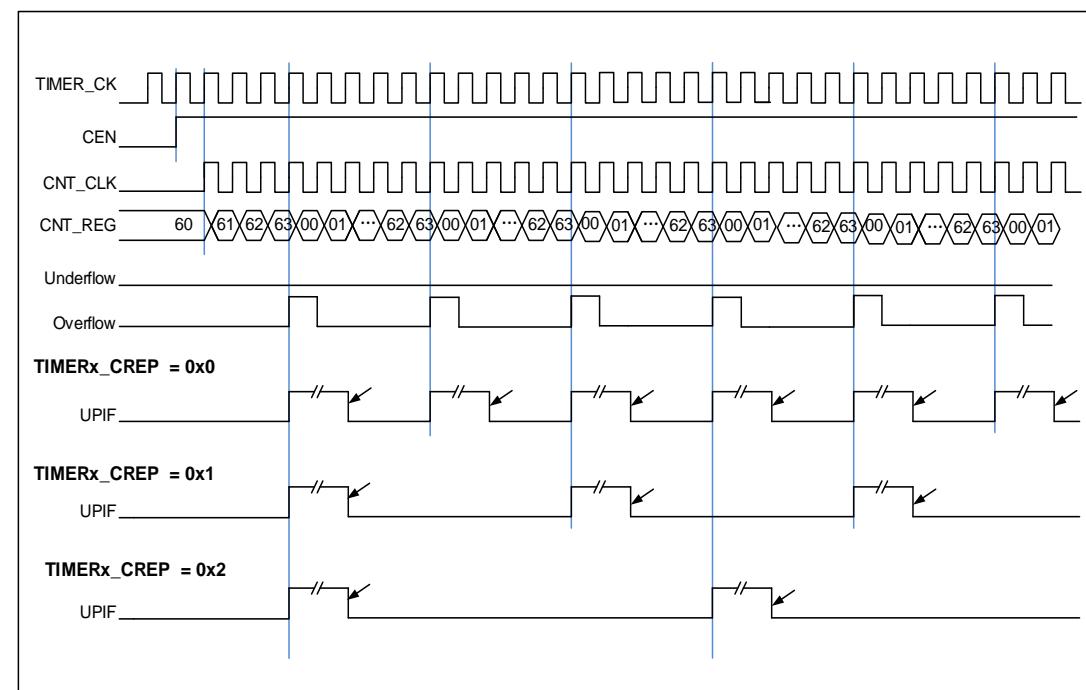
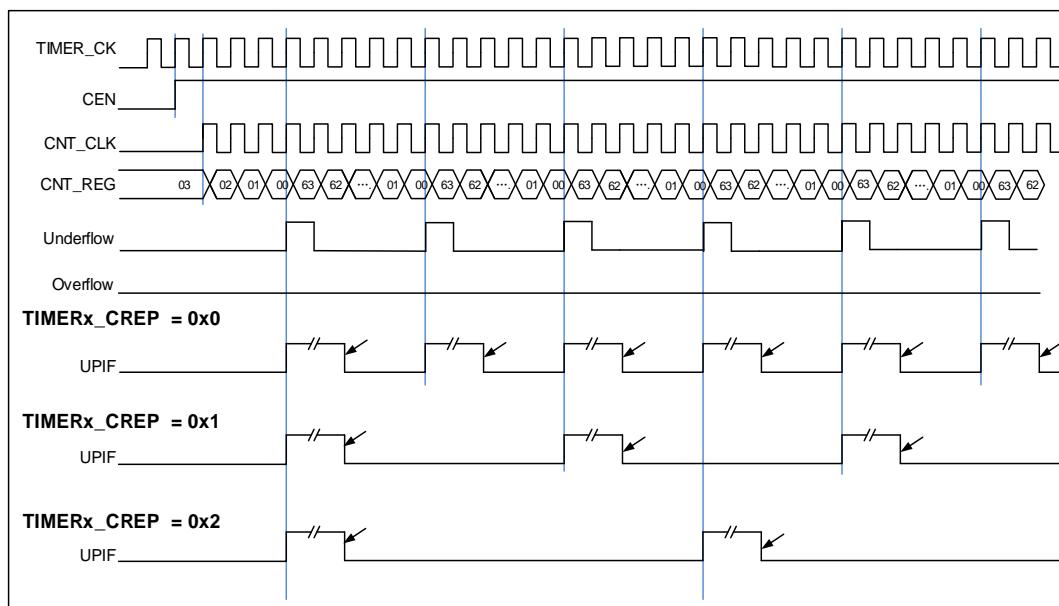


图 16-11. 在向下计数模式下计数器重复时序图



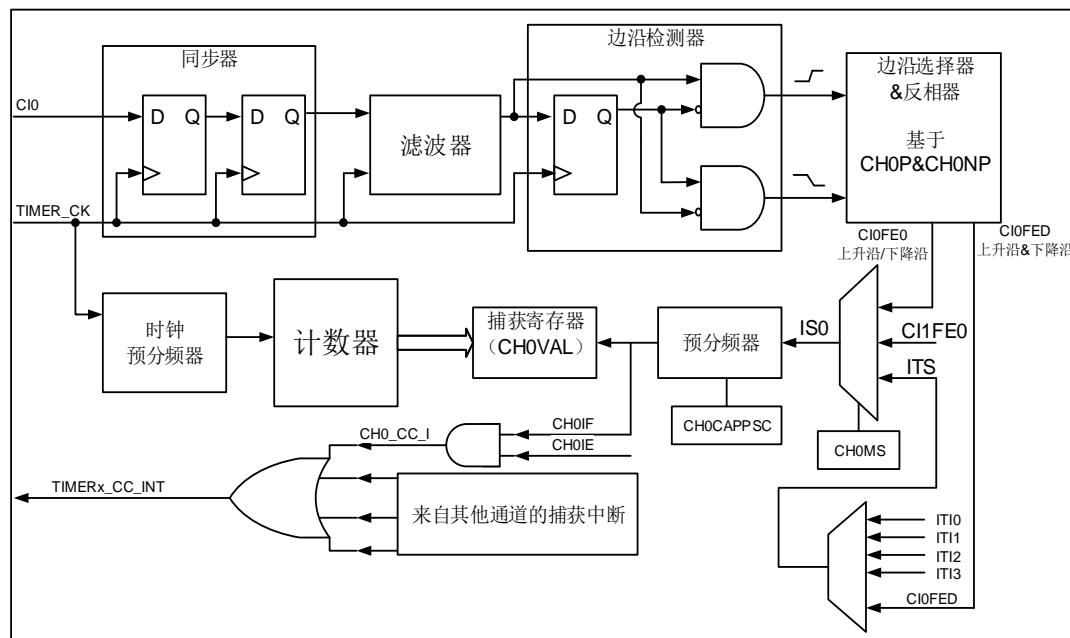
捕获/比较通道

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

输入捕获模式允许通道测量一个波形的时序，频率，周期和占空比等。输入级包括一个数字滤波器，一个通道极性选择，边沿检测和一个通道预分频器。如果在输入引脚上出现被选择的边沿，TIMERx_CHxCV寄存器会捕获计数器当前的值，同时CHxIF位被置1，若CHxIE=1则产生通道中断。

图 16-12. 输入捕获逻辑



通道输入信号 Cl_x 有两种选择，一种是 $\text{TIMER}_{\text{x}}\text{-CH}_{\text{x}}$ 信号，另一种是 $\text{TIMER}_{\text{x}}\text{-CH}_0$ ， $\text{TIMER}_{\text{x}}\text{-CH}_1$ 和 $\text{TIMER}_{\text{x}}\text{-CH}_2$ 异或之后的信号。通道输入信号 Cl_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 $\text{CH}_{\text{x}}\text{P}$ 选择使用上升沿或者下降沿。通过配置 $\text{CH}_{\text{x}}\text{MS}$ ，还可以选择其他通道的输入信号或内部触发信号作为捕获信号。配置 IC 预分频器，使得若干个输入事件后才产生一个有效的捕获事件。捕获事件发生， $\text{TIMER}_{\text{x}}\text{-CH}_{\text{x}}\text{CV}$ 存储计数器的值。

配置步骤如下：

第一步：滤波器配置（ $\text{TIMER}_{\text{x}}\text{-CHCTL0}$ 寄存器中 $\text{CH}_{\text{x}}\text{CAPFLT}$ ）：

根据输入信号和请求信号的质量，配置相应的 $\text{CH}_{\text{x}}\text{CAPFLT}$ 。

第二步：边沿选择（ $\text{TIMER}_{\text{x}}\text{-CHCTL2}$ 寄存器中 $\text{CH}_{\text{x}}\text{P}/\text{CH}_{\text{x}}\text{NP}$ ）：

配置 $\text{CH}_{\text{x}}\text{P}/\text{CH}_{\text{x}}\text{NP}$ 选择上升沿或者下降沿。

第三步：捕获源选择（ $\text{TIMER}_{\text{x}}\text{-CHCTL0}$ 寄存器中 $\text{CH}_{\text{x}}\text{MS}$ ）：

一旦通过配置 $\text{CH}_{\text{x}}\text{MS}$ 选择输入捕获源，必须确保通道配置在输入模式 ($\text{CH}_{\text{x}}\text{MS} \neq 0x0$)，而且 $\text{TIMER}_{\text{x}}\text{-CH}_{\text{x}}\text{CV}$ 寄存器不能再被写。

第四步：中断使能（ $\text{TIMER}_{\text{x}}\text{-DMAINTEN}$ 寄存器中 $\text{CH}_{\text{x}}\text{IE}$ 和 $\text{CH}_{\text{x}}\text{DEN}$ ）：

使能相应中断，可以获得中断和 DMA 请求。

第五步：捕获使能（ $\text{TIMER}_{\text{x}}\text{-CHCTL2}$ 寄存器中 $\text{CH}_{\text{x}}\text{EN}$ ）。

结果：当期望的输入信号发生时， $\text{TIMER}_{\text{x}}\text{-CH}_{\text{x}}\text{CV}$ 被设置成当前计数器的值， $\text{CH}_{\text{x}}\text{IF}$ 为置 1。

如果 $\text{CH}_{\text{x}}\text{IF}$ 位已经为 1，则 $\text{CH}_{\text{x}}\text{OF}$ 位置 1。根据 $\text{TIMER}_{\text{x}}\text{-DMAINTEN}$ 寄存器中 $\text{CH}_{\text{x}}\text{IE}$ 和 $\text{CH}_{\text{x}}\text{DEN}$ 的配置，判断相应的中断和 DMA 请求是否被提出。

直接产生：软件设置 $\text{CH}_{\text{x}}\text{G}$ 位，会直接产生中断和 DMA 请求。

输入捕获模式也可用来测量 $\text{TIMER}_{\text{x}}\text{-CH}_{\text{x}}$ 引脚上信号的脉冲波宽度。例如，一个 PWM 波连接到 Cl_0 。配置 $\text{TIMER}_{\text{x}}\text{-CHCTL0}$ 寄存器中 CH_0MS 为 $2'b01$ ，选择通道 0 的捕获信号为 Cl_0 ，同时设置上升沿捕获。配置 $\text{TIMER}_{\text{x}}\text{-CHCTL0}$ 寄存器中 CH_1MS 为 $2'b10$ ，选择通道 1 捕获信号为 Cl_0 ，同时设置下降沿捕获。计数器配置为复位模式，在通道 0 的上升沿复位。 $\text{TIMER}_{\text{x}}\text{-CH}_0\text{CV}$ 寄存器测量 PWM 的周期值， $\text{TIMER}_{\text{x}}\text{-CH}_1\text{CV}$ 寄存器测量 PWM 占空比值。

■ 输出比较模式

图 16-13. 输出比较逻辑（带有互补输出的通道， $x=0,1,2$ ）

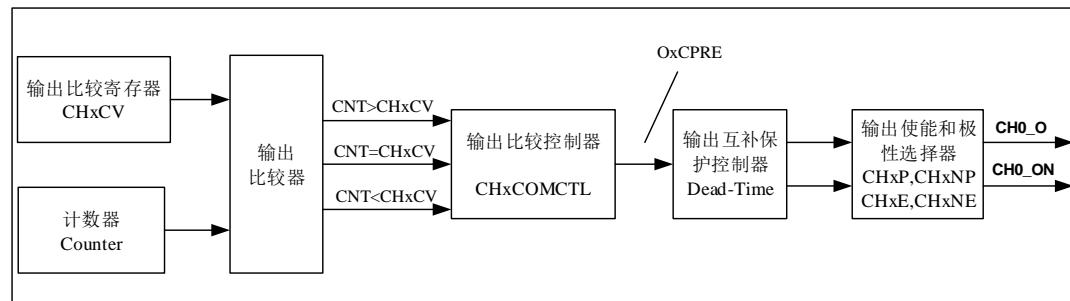
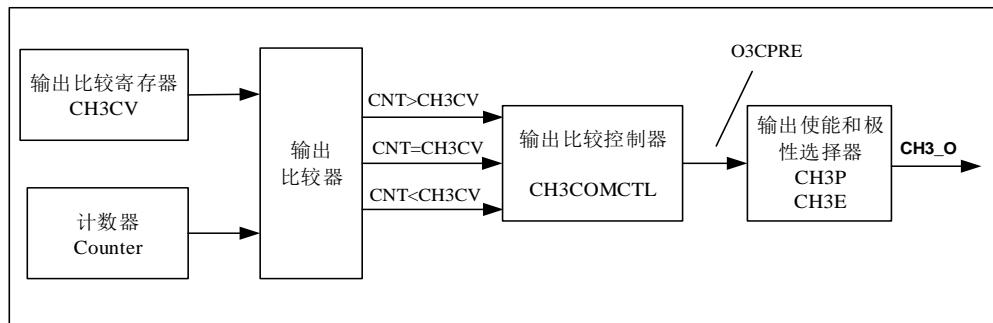


图 16-14. 输出比较逻辑



[图16-13. 输出比较逻辑（带有互补输出的通道， \$x=0,1,2\$ ）](#) 和 [图16-14. 输出比较逻辑](#) 分别给出了输出比较的逻辑电路。通道输出信号 CHx_O/CHx_ON 与 $OxCPRE$ 信号（详情请见[通道输出准备信号](#)）的关系描述如下： $OxCPRE$ 信号高电平有效， CHx_O/CHx_ON 的输出情况与 $OxCPRE$ 信号， $CHxP/CHxNP$ 位和 $CHxE/CHxNE$ 位有关（具体情况请见 $TIMERx_CHCTL2$ 寄存器中的描述）。例如：

1) 当设置 $CHxP=0$ (CHx_O 高电平有效，与 $OxCPRE$ 输出极性相同)、 $CHxE=1$ (CHx_O 输出使能) 时：

若 $OxCPRE$ 输出有效 (高) 电平，则 CHx_O 输出有效 (高) 电平；
若 $OxCPRE$ 输出无效 (低) 电平，则 CHx_O 输出无效 (低) 电平。

2) 当设置 $CHxNP=1$ (CHx_ON 低电平有效，与 $OxCPRE$ 输出极性相反)、 $CHxNE=1$ (CHx_ON 输出使能) 时：

若 $OxCPRE$ 输出有效 (高) 电平，则 CHx_ON 输出有效 (低) 电平；
若 $OxCPRE$ 输出无效 (低) 电平，则 CHx_ON 输出无效 (高) 电平。

当 $CH0_O$ 和 $CH0_ON$ 同时输出时， $CH0_O$ 和 $CH0_ON$ 的具体输出情况还与 $TIMERx_CCHP$ 寄存器中的相关位 (ROS、IOS、POE 和 DTCFG 等位) 有关。详情请见[互补输出](#)。

在输出比较模式， $TIMERx$ 可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的 $TIMERx_CHxCV$ 寄存器与计数器的值匹配时，根据 $CHxCOMCTL$ 的配置，这个通道的输出可以被置高电平，被置低电平或者反转。当计数器的值与 $TIMERx_CHxCV$ 寄存器的值匹配时， $CHxIF$ 位被置 1，如果 $CHxIE = 1$ 则会产生中断，如果 $CxDCE = 1$ 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 $CHxIE/CxDCE$ 位配置中断/DMA 请求使能。

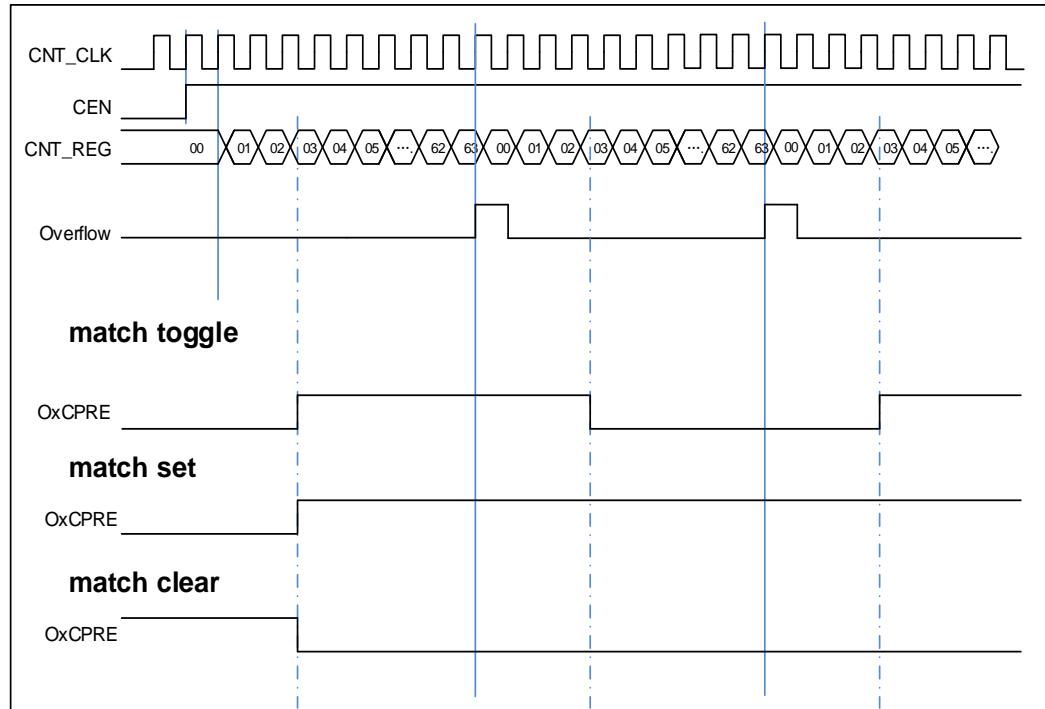
第四步：通过 $TIMERx_CAR$ 寄存器和 $TIMERx_CHxCV$ 寄存器配置输出比较时基：

TIMERx_CHxCV可以在运行时根据你所期望的波形而改变。

第五步：设置CEN位使能定时器。

[图16-15. 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 16-15. 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，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 16-16. EAPWM 时序图](#)显示了 CAPWM 的输出波形和中断。

CAPWM 的周期由 (2 * TIMERx_CAR 寄存器值) 决定，占空比由 (2 * TIMERx_CHxCV 寄存器值) 决定。[图 16-17. CAPWM 时序图](#)显示了 CAPWM 的输出波形和中断。

在向上计数模式中，PWM 模式 0 下 (CHxCOMCTL=3'b110)，如果 TIMERx_CHxCV 寄存器的值大于 TIMERx_CAR 寄存器的值，通道输出一直为无效电平；PWM 模式 1 下 (CHxCOMCTL=3'b111)，如果 TIMERx_CHxCV 寄存器的值大于 TIMERx_CAR 寄存器的值，通道输出一直为有效电平。

图 16-16. EAPWM 时序图

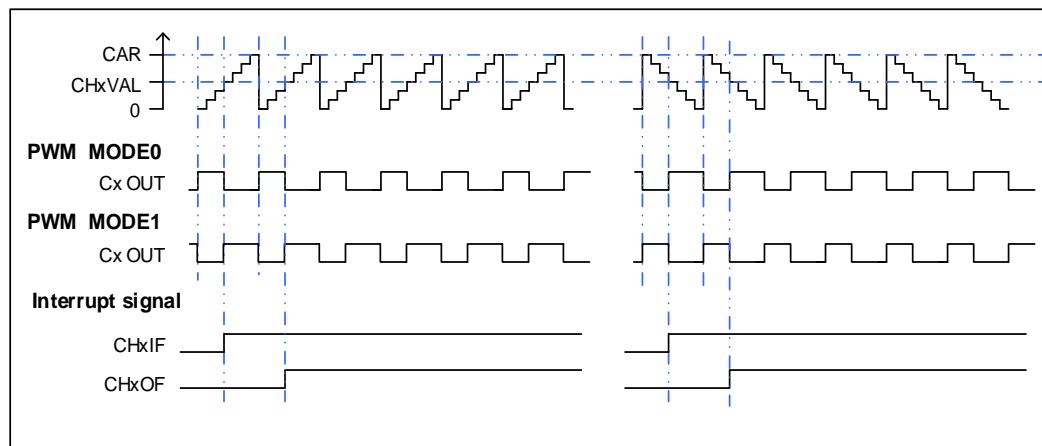
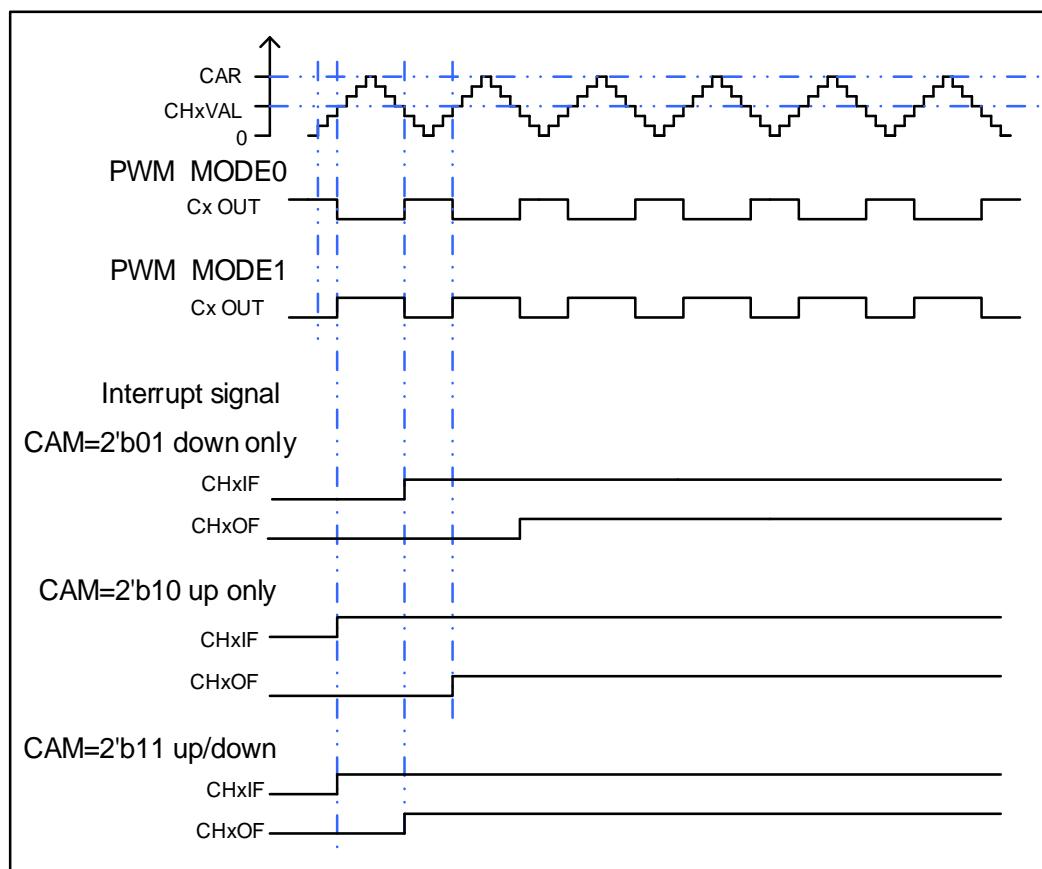


图 16-17. CAPWM 时序图



通道输出准备信号

根据 [图16-13. 输出比较逻辑（带有互补输出的通道, x=0,1,2）](#) 所示, 当TIMERx用于输出匹配比较模式下, 在通道输出信号之前会产生一个中间信号OxCPRE信号(通道x输出准备信号)。设置CHxCOMCTL位可以定义OxCPRE信号类型。OxCPRE信号有若干类型的输出功能, 包括, 设置CHxCOMCTL=0x00可以保持原始电平; 设置CHxCOMCTL=0x01可以将OxCPRE信号设置为高电平; 设置CHxCOMCTL=0x02可以将OxCPRE信号设置为低电平; 设置

CHxCOMCTL=0x03，在计数器值和TIMERx_CHxCV寄存器的值匹配时，可以翻转输出信号。

PWM模式0和PWM模式1是OxCPRE的另一种输出类型，设置CHxCOMCTL位域为0x06或0x07可以配置PWM模式0/PWM模式1。在这些模式中，根据计数器值和TIMERx_CHxCV寄存器值的关系以及计数方向，OxCPRE信号改变其电平。具体细节描述，请参考相应的位。

设置CHxCOMCTL=0x04或0x05可以实现OxCPRE信号的强制输出功能。输出比较信号能够直接由软件强置为有效或无效状态，而不依赖于TIMERx_CHxCV的值和计数器值之间的比较结果。

设置CHxCOMCEN=1，当由外部ETI引脚信号产生的ETIFP信号为高电平时，OxCPRE被强制为低电平。在下一次更新事件到来时，OxCPRE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寄存器中的POEN、ROS和IOS位，TIMERx_CTL1寄存器中的ISOx和ISOxN位。输出极性由TIMERx_CHCTL2寄存器中的CHxP和CHxNP位来决定。

表 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。 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。	
				1	CHx_O = CHxP CHx_ON = CHxNP CHx_O/CHx_ON输出使能。 如果时钟使能：CHx_O = ISOx CHx_ON = ISOxN	
		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	
			1	0	CHx_O=OxCPRE⊕CHxP CHx_O输出使能	CHx_ON = LOW CHx_ON输出禁用。

互补参数					输出状态	
POEN	ROS	IOS	CHxEN	CHxNEN	CHx_O	CHx_ON
1	0	1	CHx_O=OxCPRE⊕CHxP CHx_O输出使能	CHx_ON=(! OxCPRE) ⊕ CHxNP CHx_ON输出使能		
		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	CHx_O=OxCPRE⊕CHxP CHx_O输出使能	CHx_ON = CHxNP CHx_ON输出使能		

死区时间插入

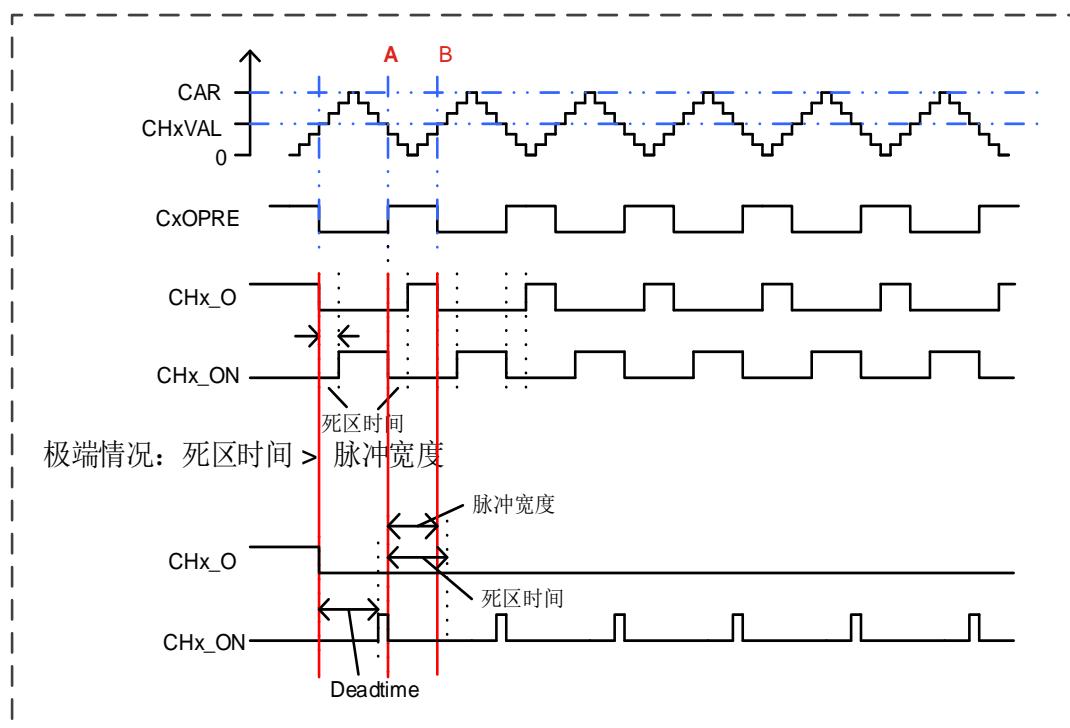
设置CHxEN和CHxNEN为1'b1的同时，设置POEN为1，死区插入就会被使能。DTCFG位域定义了死区时间，死区时间对除了通道3以外的通道有效。死区时间的细节，请参考TIMERx_CCHP寄存器。

死区时间的插入，确保了通道互补的两路信号不会同时有效。

在 PWM 模式 0，当通道 x 匹配事件发生时 (TIMERx 计数器=CHxVAL)，OxCPRE 反转。在 [图 16-18. 带死区时间的互补输出](#) 中的 A 点，CHx_O 信号在死区时间内为低电平，直到死区时间过后才变为高电平，而 CHx_ON 信号立刻变为低电平。同样，在 B 点，通道 x 匹配事件再次发生 (TIMERx 计数器=CHxVAL)，OxCPRE 信号被清 0，CHx_O 信号被立即清零，CHx_ON 信号在死区时间内仍然是低电平，在死区时间过后才变为高电平。

有时会有一些极端事件发生，例如：如果死区延时大于或者等于CHx_ON信号的占空比，CHx_ON信号一直为无效值。

图 16-18. 带死区时间的互补输出



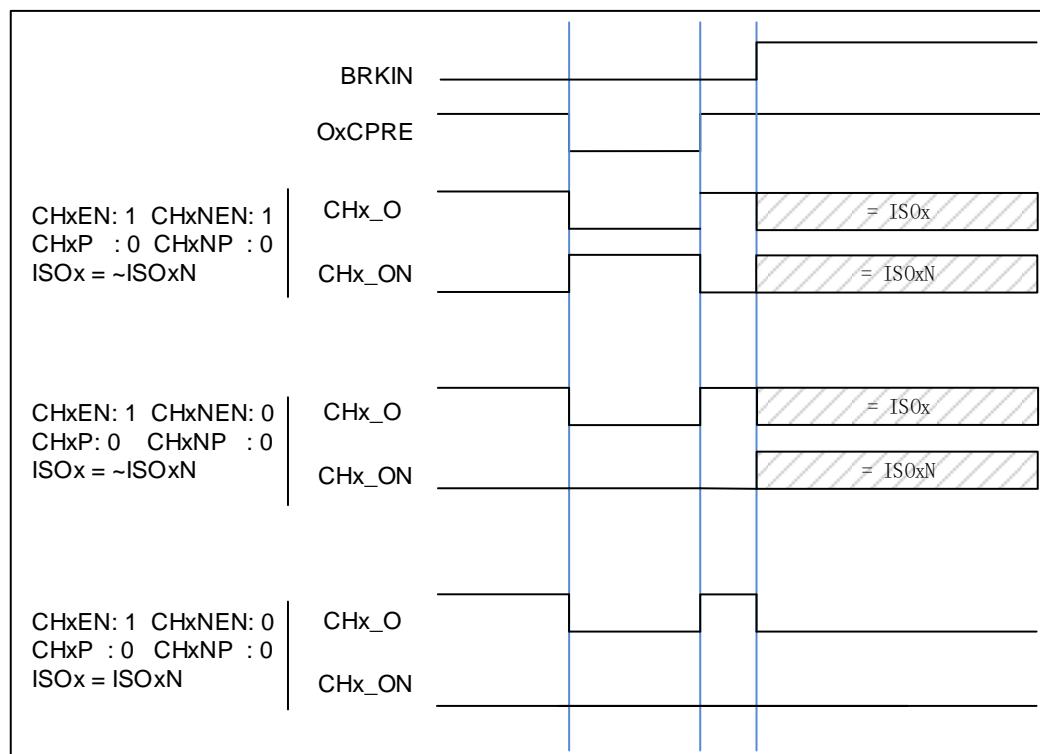
中止功能

使用中止功能时，输出CHx_O和CHx_ON的信号电平被以下位控制，**TIMERx_CCHP**寄存器的POEN，IOS和ROS位，**TIMERx_CTL1**寄存器的ISOx和ISOxN位。任何情况下，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的输出电平由**TIMERx_CTL1**寄存器中的ISOx位和ISOxN位决定。如果IOS=0，定时器释放输出使能，否则输出使能仍然为高。起初互补输出被置于复位状态，然后死区时间产生器重新被激活，以便在一个死区时间后驱动输出，输出电平由ISOx和ISOxN位配置。

发生中止时，**TIMERx_INTF**寄存器的BRKIF位被置1。如果BRKIE=1，中断产生。

图 16-19. 通道响应中止输入（高电平有效）时，输出信号的行为



正交译码器

正交译码器功能使用由TIMERx_CH0和TIMERx_CH1引脚生成的CI0和CI1正交信号各自相互作用产生计数值。通过设置SMC=0x01, 0x02或0x03来选择是仅由CI0, 仅由CI1, 或者由CI0和CI1来决定定时器的计数方向。在每个方向选择源的电平改变期间, DIR位是由硬件自动改变的。计数器计数方向改变的机制如[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	CI0FE0		CI1FE1	
		上升	下降	上升	下降
只有CI0	CI1FE1=1	Down	Up	-	-
	CI1FE1=0	Up	Down	-	-
只有CI1	CI0FE0=1	-	-	Up	Down
	CI0FE0=0	-	-	Down	Up
CI0和CI1	CI1FE1=1	Down	Up	X	X
	CI1FE1=0	Up	Down	X	X
	CI0FE0=1	X	X	Up	Down
	CI0FE0=0	X	X	Down	Up

注意：“-”意思是“无计数”；“X”意思是不可能。

图 16-20. 编码器接口模式下计数器运行例子

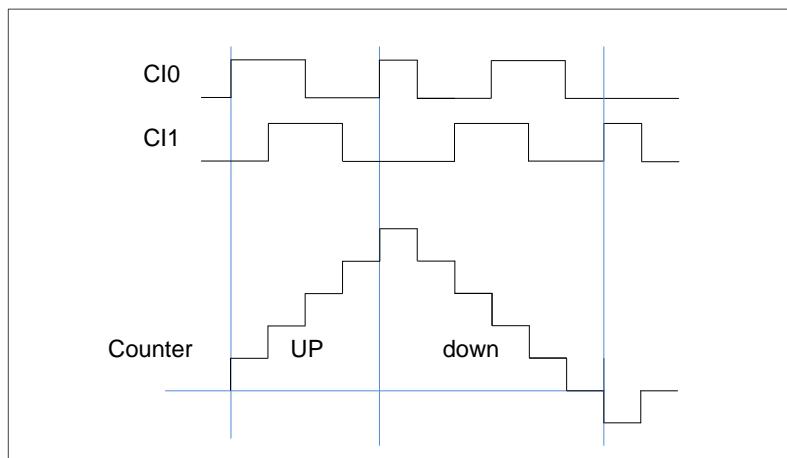
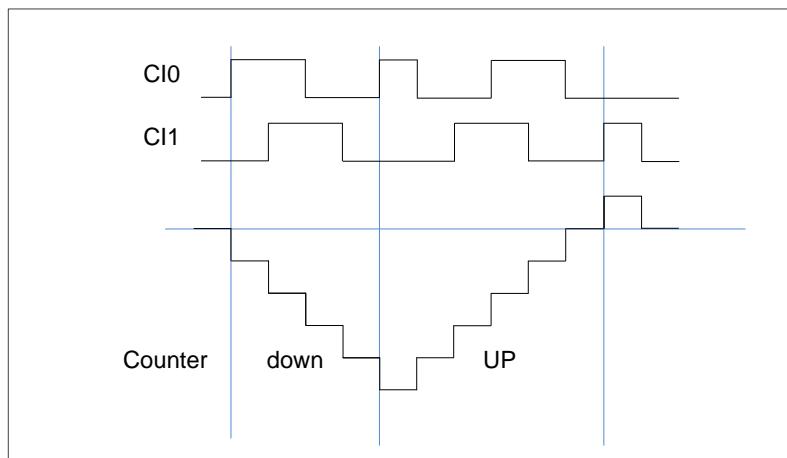


图 16-21. CI0FE0 极性反相的编码器接口模式下的例子



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 16-22. 霍尔传感器用在 BLDC 电机控制中是定时器和电机的连接示意图。众所周知，我们要两个定时器。**TIMER_in** 定时器（可以是高级定时器或者通用 L0 定时器）接收来自电机霍尔传感器的三路信号，这三路信号是电机转子的位置信号。

三个霍尔传感器与**TIMER_in**定时器的三路输入捕获引脚一一对应连接，每个霍尔传感器输入一路波形到输入引脚，分析三路霍尔信号可以计算出转子的位置和速度。

通过定时器内部连接功能（**TRGO-ITIx**），**TIMER_in**定时器和**TIMER_out**定时器可以连接在一起。**TIMER_out**定时器根据ITIx触发信号输出PWM波，驱动BLDC电机，控制BLDC电机的速度。这样，**TIMER_in**定时器和**TIMER_out**定时器的连接形成了一个反馈电路，可以根据需求改变配置。

高级定时器和通用L0定时器具有输入异或功能，可作为**TIMER_in**定时器。同时，高级定时器

具备互补输出和死区插入功能，可作为TIMER_out定时器。

另外，根据定时器的内部互连关系，可以选择成对的互连定时器，例如：

TIMER_in (TIMER0) -> TIMER_out (TIMER7 ITI0)
 TIMER_in (TIMER1) -> TIMER_out (TIMER0 ITI1)

选择好合适的互连定时器，线路也已经连接好，就可以配置定时器。有以下关键配置：

- 通过设置TIOS，来使能异或功能。三路输入信号的任何一路发生变化，CI0都会反转，CH0VAL此时会捕获计数器的当前值。
- 通过设置CCUC和CCSE，来选择ITIx触发换相。
- 根据需求配置PWM参数。

图 16-22. 霍尔传感器用在 BLDC 电机控制中

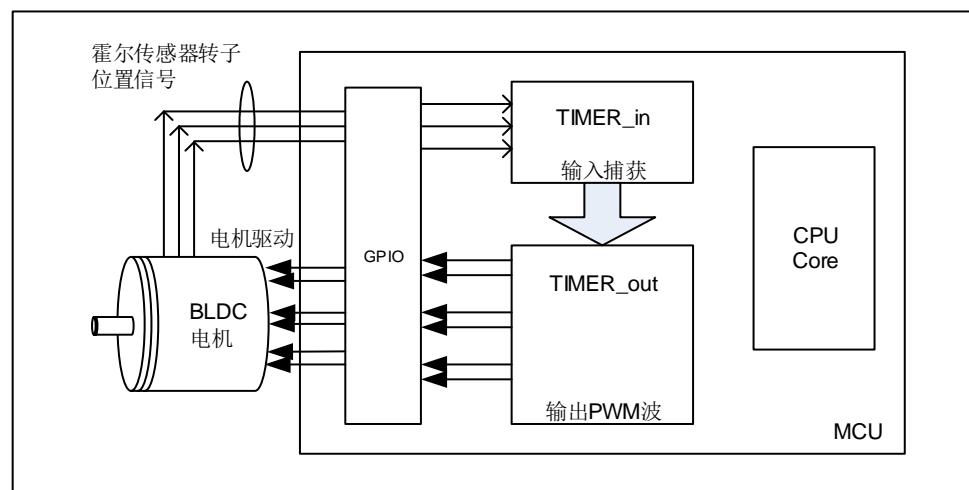
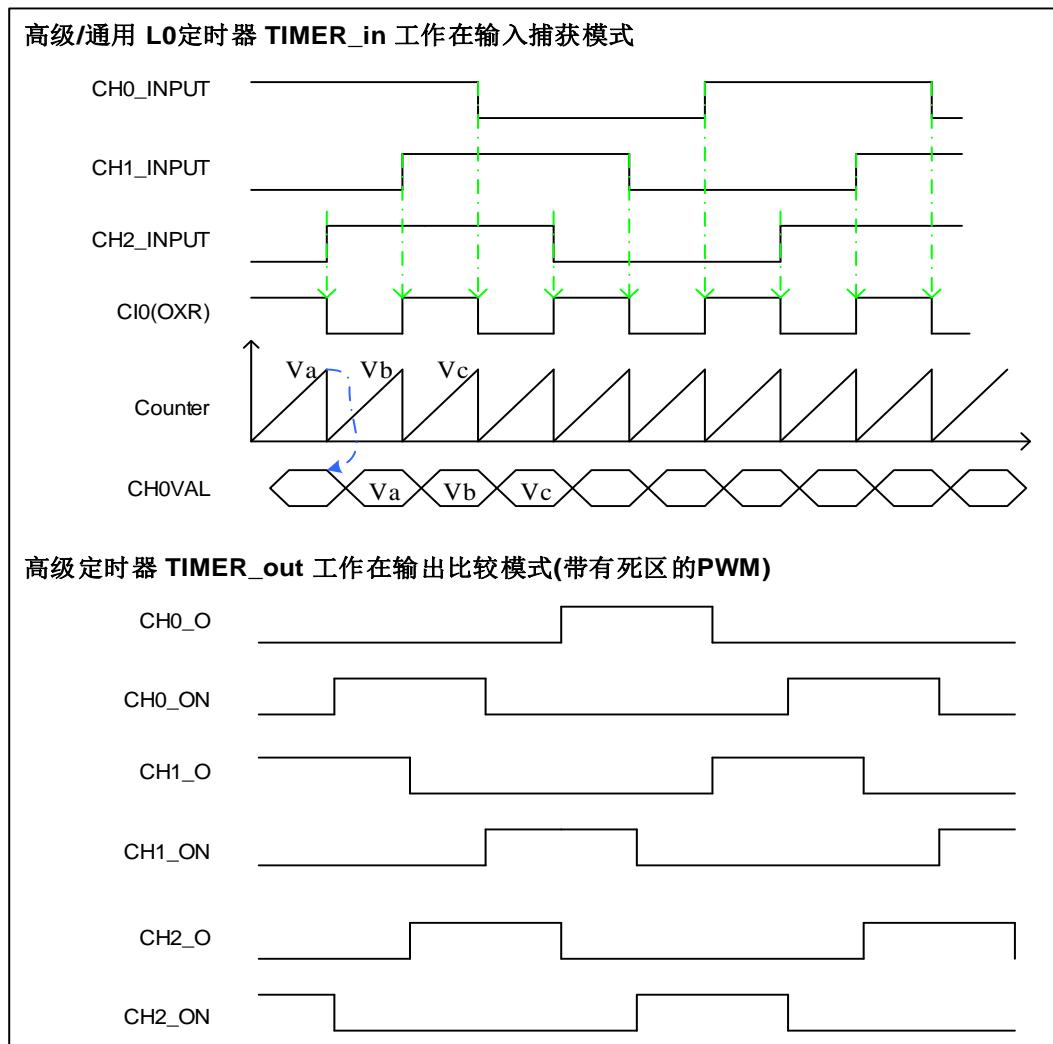


图 16-23. 两个定时器之间的霍尔传感器时序图

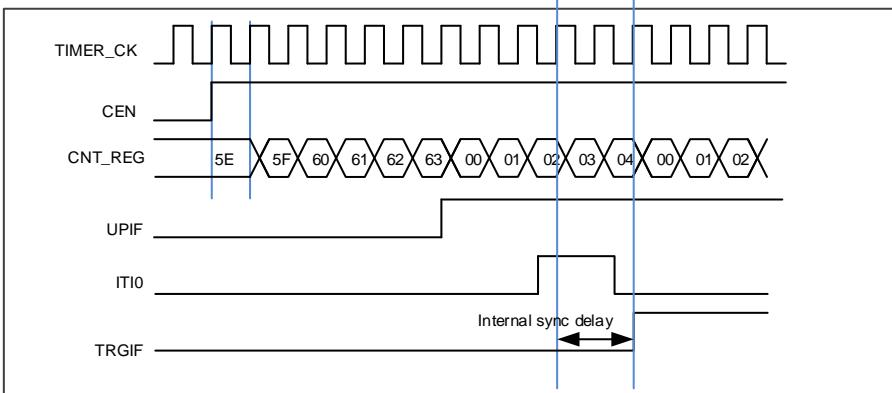
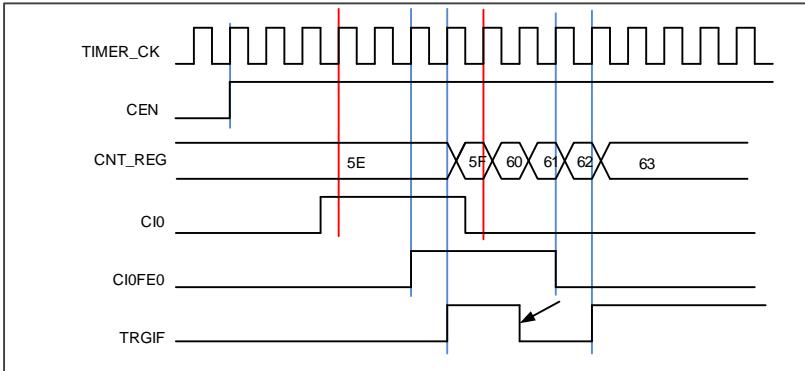


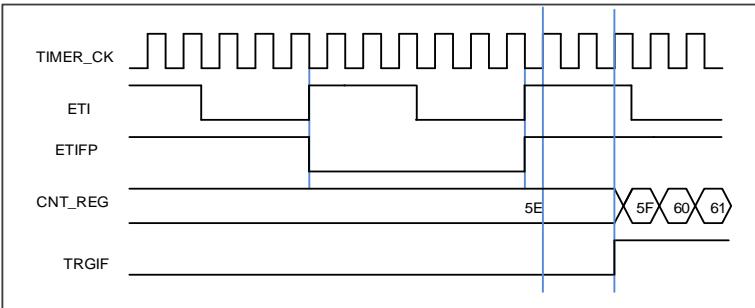
从控制器

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. 从模式示例

	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CI1FE，配置CHxP和CHxNP来选择极性和反相。 如果触发源是ETIFFP，配置ETP选择极性和反相。	若触发源为ITIx，滤波和预分频不可用。 若触发源为CIx，可配置CHxCAPFLT设置滤波，预分频不可用。 若触发源为ETIFFP，滤波和预分频均可用。

	模式选择	触发源选择	极性选择	滤波和预分频
		111: ETIFP		
	复位模式 当触发输入上升沿到来时，计数器清零重启。	TRGS[2:0]=3'b000 选择ITIO为触发源。	若触发源是ITIO，极性选择不可用。	若触发源是ITIO，滤波和预分频不可用。
图 16-24. 复位模式				
例1				
	暂停模式 当触发输入为低的时候，计数器暂停计数，当触发输入为高时，计数器计数。	TRGS[2:0]=3'b101 选择CI0FE0为触发源。	TIOS=0 (非异或) [CH0NP=0, CH0P=0] CI0FE0不反相。捕获发生在上升沿。	在这个例子中滤波被旁路。
例2				
例3	事件模式 触发输入的上升沿计数器开始计数。	TRGS[2:0]=3'b111 选择ETIFP为触发源。	ETP = 0, ETI极性不变。 ETPSC = 1, ETI 2分频。 ETFC = 0, ETI 无滤波。	

	模式选择	触发源选择	极性选择	滤波和预分频
图 16-26. 事件模式				
				 <p>The diagram shows the timing relationships between the timer clock (TIMER_CK), external trigger input (ETI), external trigger filter (ETIFP), counter register (CNT_REG), and trigger flag (TRGIF). The counter increments on each rising edge of the timer clock. When an external trigger (ETI) occurs, it triggers a counter update. The ETIFP signal is used to filter the trigger edge. The counter value is shown in hex: 5E, 5F, 60, 61. The TRGIF flag is set when the counter reaches the value of 61.</p>

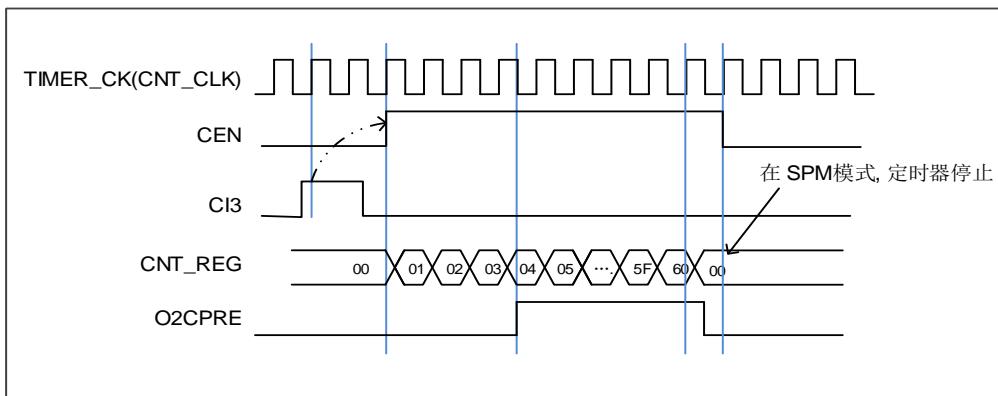
单脉冲模式

设置TIMERx_CTL0寄存器的SPM位置1，使能单脉冲模式。当SPM置1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置CHxCOMCTL配置TIMERx为PWM模式或者比较模式。

一旦设置定时器运行在单脉冲模式下，没有必要设置TIMERx_CTL0寄存器的定时器使能位CEN=1来使能计数器。触发信号沿或者软件写CEN=1都可以产生一个脉冲，此后CEN位一直保持为1直到更新事件发生或者CEN位被软件写0。如果CEN位被软件清0，计数器停止工作，计数值被保持。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PWM模式0或PWM模式1时CHxCOMFEN位才可用，触发源来源于触发信号。

图 16-27. 单脉冲模式，TIMERx_CHxCV = 0x04，TIMERx_CAR=0x60

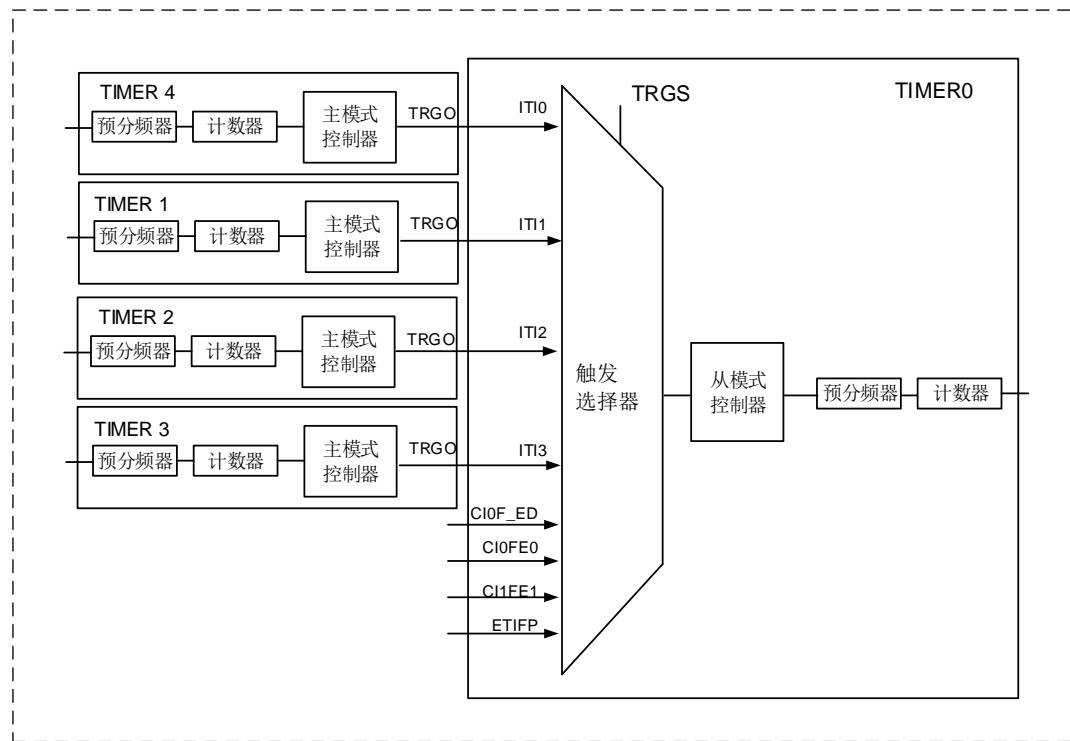


定时器互连

定时器之间的相互连接可以实现定时器的级联或者同步。可以通过配置一个定时器工作在主模式，另一个定时器工作在从模式来实现。下面的几张图显示了一些主从模式触发选择的例子。

图 16-28. 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 16-28. 定时器 0 主/从模式的例子



其他定时器互连的例子：

- 定时器2作为定时器0的预分频器

参考 [图 16-28. 定时器 0 主/从模式的例子](#) 连接配置定时器 2 为定时器 0 的预分频器，步骤如下：

1. 配置定时器2为主模式，选择其更新事件(UPE)为触发输出(配置TIMER2_CTL1寄存器的MMC=3'b010)。定时器2在每次计数器溢出产生更新事件时，输出一个周期信号；
2. 配置定时器2周期(TIMER2_CAR寄存器)；
3. 选择定时器0输入触发源为定时器2 (配置TIMERx_SMCFG寄存器的TRGS=3'b010)；
4. 配置定时器0在外部时钟模式0(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2 (TIMER2_CTL0寄存器)。

- 用定时器2的使能/更新信号来启动定时器0

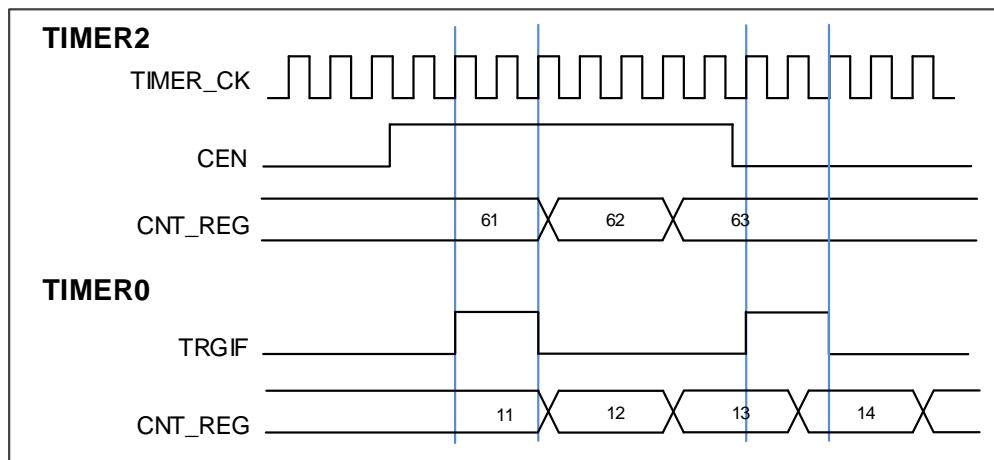
用定时器 2 的使能信号来启动定时器 0, 见[图 16-29. 用定时器 2 的使能信号触发定时器 0](#)。

在定时器 2 使能信号输出后, 定时器 0 按照分频后的内部时钟从当前值开始计数。

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

1. 配置定时器2为主模式, 发送它的使能信号作为触发输出(配置TIMER2_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2 (配置TIMERx_SMCFG寄存器的TRGS=3'b010);
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2 (TIMER2_CTL0寄存器)。

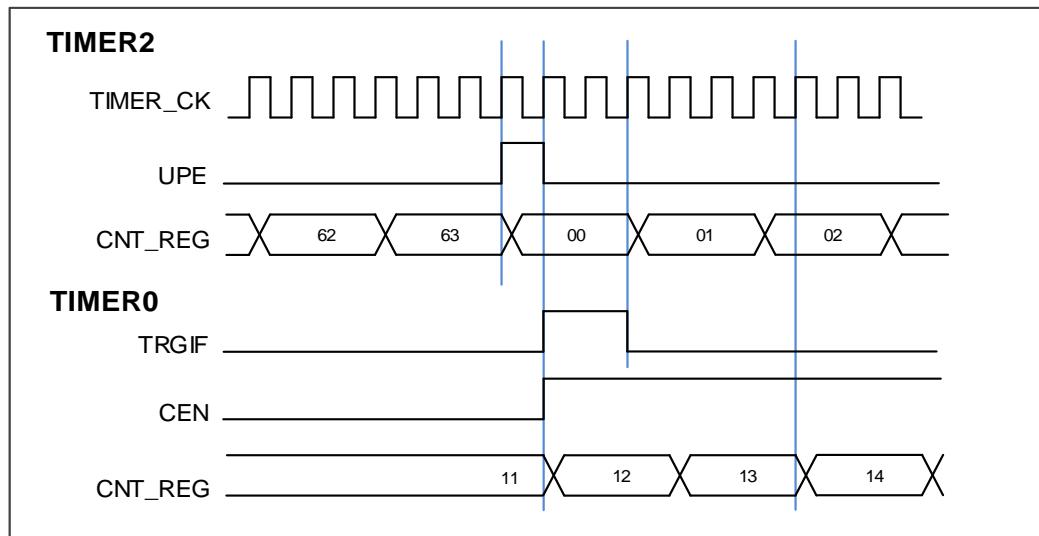
图 16-29. 用定时器 2 的使能信号触发定时器 0



在这个例子中, 也可以使用更新事件代替使能信号作为触发源。见[图 16-30. 用定时器 2 的更新事件来触发定时器 0](#), 按以下步骤进行:

1. 配置定时器2为主模式, 发送它的更新事件 (UPE) 作为触发输出(配置TIMER2_CTL1寄存器的MMC=3'b010);
2. 配置定时器2的周期 (TIMER2_CARL寄存器);
3. 配置定时器0选择输入触发来自定时器2 (配置TIMERx_SMCFG寄存器的TRGS=3'b010);
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2 (TIMER2_CTL0寄存器)。

图 16-30. 用定时器 2 的更新事件来触发定时器 0

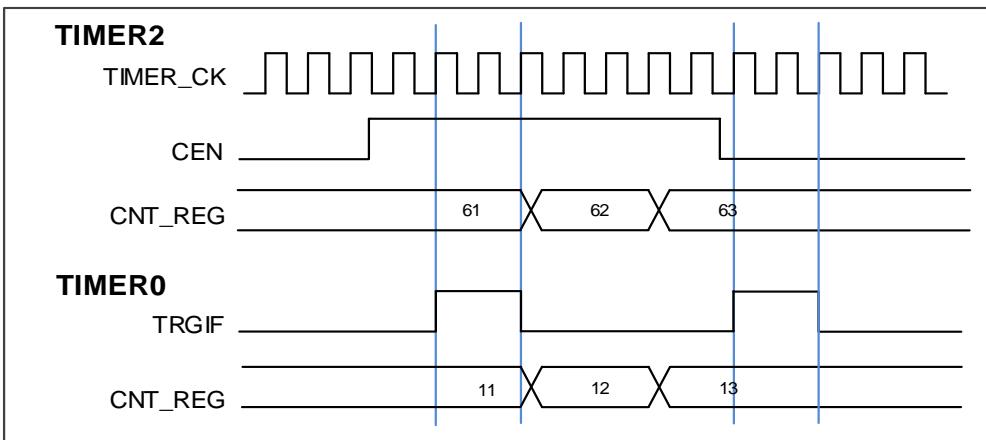


- 使用定时器2的使能/O0CPRE信号来使能定时器0计数。

在这个例子中，使用定时器2的使能信号来使能定时器0。如[图16-31. 用定时器2的使能信号来控制定时器0](#)，在定时器2被使能后，定时器0在内部分频的时钟上开始计数。两个计数器的时钟频率都是由**TIMER_CK**时钟3分频得来($f_{PSC_CLK} = f_{TICK_CK}/3$)，步骤如下：

1. 配置定时器2在主模式，配置其输出使能信号作为触发输出(配置**TIMER2_CTL1**寄存器的MMC=3'b001);
2. 配置定时器0从定时器2获取输入触发(配置**TIMERx_SMCFG**寄存器的TRGS=3'b010);
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2 (**TIMER0_CTL0**寄存器);
6. 写0到CEN位来停止定时器2 (**TIMER0_CTL0**寄存器)。

图 16-31. 用定时器 2 的使能信号来控制定时器 0 的暂停模式

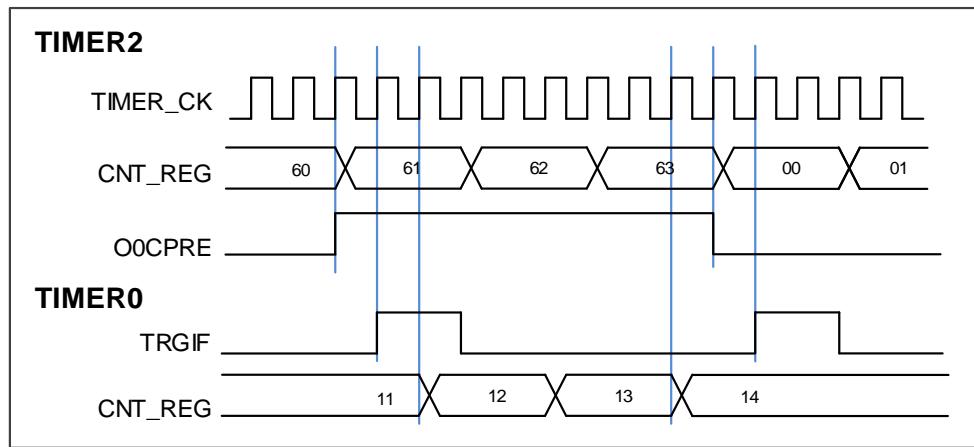


这个例子中，我们也可以使用定时器2的O0CPRE信号代替其使能信号输出作为触发源。步骤

如下：

1. 配置定时器2在主模式下，配置O0CPRE信号为触发输出(配置TIMER2_CTL1寄存器的MMS=3'b100);
2. 配置定时器2的O0CPRE波形(TIMER2_CHCTL0寄存器);
3. 配置定时器0获取来自定时器2的输入触发(配置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2 (TIMER0_CTL0寄存器)。

图 16-32. 用定时器 2 的 O0CPRE 信号控制定时器 0 的暂停模式



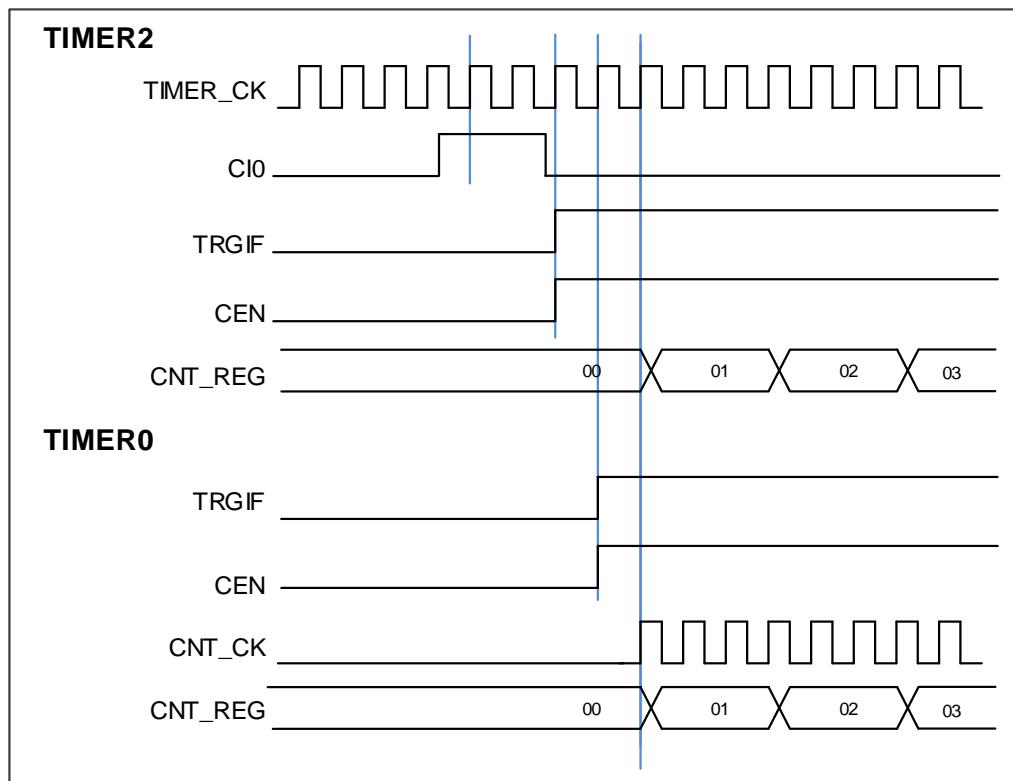
- 使用一个外部触发来同步两个定时器。

配置定时器2的使能信号触发定时器0的开启，配置定时器2的CI0输入信号上升沿来触发定时器2。为了确保两个定时器同步开启，定时器2必须配置在主/从模式。步骤如下：

1. 配置定时器2工作在从模式，并选择CI0_ED作为触发输入(配置TIMER2_SMCFG寄存器的TRGS=3'b100);
2. 配置定时器2工作在事件模式(配置TIMER2_SMCFG寄存器的SMC=3'b110);
3. 写MSM=1(TIMER2_SMCFG寄存器)来配置定时器2工作在主/从模式;
4. 配置定时器0的触发输入为定时器2 (配置TIMERx_SMCFG 寄存器的TRGS=3'b010);
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2的CI0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 16-33. 用定时器 2 的 CI0 信号来触发定时器 0 和定时器 2



定时器 DMA 模式

定时器DMA模式是指通过DMA模块配置定时器的寄存器。有两个跟定时器DMA模式相关的寄存器：**TIMERx_DMACFG**和**TIMERx_DMATB**。必须使能相应的DMA请求位，一些内部中断事件才可以产生DMA请求。当中断事件发生，**TIMERx**会给DMA发送请求。DMA配置成M2P（传输方向为从内存到外设）模式，**PADDR**（外设基地址）为**TIMERx_DMATB**寄存器地址，DMA就会访问**TIMERx_DMATB**寄存器。实际上，**TIMERx_DMATB**寄存器只是一个缓冲，定时器会将**TIMERx_DMATB**映射到一个内部寄存器，这个内部寄存器由**TIMERx_DMACFG**寄存器中的**DMATA**来指定。如果**TIMERx_DMACFG**寄存器的**DMATC**位域值为0，表示1次传输，定时器发送1个DMA请求就可以完成。如果**TIMERx_DMACFG**寄存器的**DMATC**位域值不为1，例如其值为3，表示4次传输，定时器就需要再多发3次DMA请求。在这3次请求下，DMA对**TIMERx_DMATB**寄存器的访问会映射到访问定时器的**DMATA+0x4**，**DMATA+0x8**，**DMATA+0xC**寄存器。总之，发生一次DMA内部中断请求，定时器会连续发送(**DMATC+1**)次请求。

如果再来1次DMA请求事件，**TIMERx**将会重复上面的过程。

定时器调试模式

当Cortex™-M4内核停止，**DBG_CTL**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=0,7)

TIMER0基地址: 0x4001 2C00

TIMER7基地址: 0x4001 3400

控制寄存器 0 (TIMERx_CTL0)

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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				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
31:10	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TIMER_CK(定时器时钟)与DTS(死区时间和采样时钟)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: $f_{DTS}=f_{TIMER_CK}$</p> <p>01: $f_{DTS}=f_{TIMER_CK} / 2$</p> <p>10: $f_{DTS}=f_{TIMER_CK} / 4$</p> <p>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</p> <p>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。</p> <p>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。</p> <p>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。</p> <p>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，该位不能从0x00切换到非0x00.</p>

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: 单脉冲模式禁能，更新事件发生后，计数器继续计数。 1: 单脉冲模式使能，在下一次更新事件发生时， CEN 位硬件清零并且计数器停止计数。
2	UPS	更新请求源 软件配置该位，选择更新事件源。 0: 使能后，下述任一事件产生更新中断或DMA请求： – UPG位被置1； – 计数器溢出/下溢； – 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。事件模式可以自动地通过硬件将 CEN 位置位。

控制寄存器 1 (TIMERx_CTL1)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	ISO3	ISO2N	ISO2	ISO1N	ISO1	ISO0N	ISO0	TIO5	MMC[2:0]	DMAS	CCUC	保留	CCSE		

位/位域	名称	描述
31:15	保留	必须保持复位值
14	ISO3	通道3的空闲状态输出 参考ISO0位
13	ISO2N	通道2的互补通道空闲状态输出 参考ISO0N位
12	ISO2	通道2的空闲状态输出 参考ISO0位
11	ISO1N	通道1的互补通道空闲状态输出 参考ISO0N位
10	ISO1	通道1的空闲状态输出 参考ISO0位
9	ISO0N	通道0的互补通道空闲状态输出 0: 当POEN位复位时, CH0_ON设置低电平 1: 当POEN位复位时, CH0_ON设置高电平 此位只有在TIMERx_CCHP寄存器的PROT[1:0]位为00的时候可以被更改。
8	ISO0	通道0的空闲状态输出 0: 当POEN位复位时, CH0_O设置低电平 1: 当POEN位复位时, CH0_O设置高电平 如果CH0_ON生效, 一个死区时间后CH0_O输出改变。此位只有在TIMERx_CCHP寄存器的PROT[1:0]位为00的时候可以被更改。
7	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, TIMERx_CH1和TIMERx_CH2引脚异或的结果作为通道0的触发输入
6:4	MMC[2:0]	主模式控制 这些位控制TRGO信号的选择, TRGO信号由主定时器发给从定时器用于同步功能。 000: 复位。TIMERx_SWEVG寄存器的UPG位被置1或从模式控制器产生复位时, 触发一次TRGO脉冲, 后一种情况下, TRGO上的信号相对实际的复位会有一个延迟。 001: 使能。此模式可用于同时启动多个定时器或控制在一段时间内使能从定时器。主模式控制器选择计数器使能信号作为触发输出TRGO。当CEN控制位被置1或者暂停模式下触发输入为高电平时, 计数器使能信号被置1。在暂停模式下, 计数器使能信号受控于触发输入, 在触发输入和TRGO上会有一个延迟, 除非选择了主/从模式。 010: 更新。主模式控制器选择更新事件作为TRGO。 011: 捕获/比较脉冲。通道0在发生一次捕获或一次比较成功时, 主模式控制器产生一个TRGO脉冲。 100: 比较。在这种模式下, 主模式控制器选择O0CPRE信号作为触发输出TRGO。 101: 比较。在这种模式下, 主模式控制器选择O1CPRE信号作为触发输出TRGO。

110: 比较。在这种模式下, 主模式控制器选择O2CPRE信号作为触发输出TRGO。
 111: 比较。在这种模式下, 主模式控制器选择O3CPRE信号作为触发输出TRGO。

3	DMAS	DMA请求源选择 0: 当通道捕获/比较事件发生时, 发送通道x的DMA请求 1: 当更新事件发生, 发送通道x的DMA请求
2	CCUC	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 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	SMC1	ETPSC[1:0]		ETFC[3:0]		MSM		TRGS[2:0]		保留		SMC[2:0]			

位/位域	名称	描述
31:16	保留	必须保持复位值
15	ETP	外部触发极性 该位指定ETI信号的极性 0: ETI高电平或上升沿有效 1: ETI低电平或下降沿有效
14	SMC1	SMC的一部分使能外部时钟模式1 在外部时钟模式1, 计数器由ETIFP信号上的任意有效边沿驱动 0: 外部时钟模式1禁能 1: 外部时钟模式1使能

复位模式，暂停模式和事件模式可以与外部时钟模式1同时使用。但是不能将TRGS设为3'b111。

如果同时使能外部时钟模式0和外部时钟模式1，外部时钟的输入是ETIFP。

注意：外部时钟模式0使能在寄存器的SMC位域。

13:12	ETPSC[1:0]	外部触发预分频 外部触发信号ETIFP的频率不能超过TIMER_CK频率的1/4。当输入较快的外部时钟时，可以使用预分频降低ETIFP的频率。 00: 预分频禁能 01: ETIFP频率被2分频 10: ETIFP频率被4分频 11: ETIFP频率被8分频
11:8	ETFC[3:0]	外部触发滤波控制 数字滤波器是一个事件计数器，N个事件后会产生一个输出的跳变。这些位定义了对ETIFP信号采样的频率和对ETIFP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 该位用来同步被选择的定时器同时开始计数。TRGI用做启动事件，通过TRGO，定时器被连接在一起。 0: 主从模式禁能 1: 主从模式使能
6:4	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定时器的触发输入源 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: 内部触发输入3 (ITI3) 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	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: 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。在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 0000

该寄存器只能按字(32位)访问。

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

位/位域	名称	描述
31:15	保留	必须保持复位值
14	TRGDEN	触发DMA请求使能 0: 禁止触发DMA请求 1: 使能触发DMA请求
13	CMTDEN	换相DMA更新请求使能 0: 禁止换相DMA更新请求 1: 使能换相DMA更新请求
12	CH3DEN	通道3比较/捕获DMA请求使能 0: 禁止通道3比较/捕获DMA请求

		1: 使能通道3比较/捕获DMA请求
11	CH2DEN	通道2比较/捕获DMA请求使能 0: 禁止通道2比较/捕获DMA请求 1: 使能通道2比较/捕获DMA请求
10	CH1DEN	通道1比较/捕获DMA请求使能 0: 禁止通道1比较/捕获DMA请求 1: 使能通道1比较/捕获DMA请求
9	CH0DEN	通道0比较/捕获DMA请求使能 0: 禁止通道0比较/捕获DMA请求 1: 使能通道0比较/捕获DMA请求
8	UPDEN	更新DMA请求使能 0: 禁止更新DMA请求 1: 使能更新DMA请求
7	BRKIE	中止中断使能 0: 禁止中止中断 1: 使能中止中断
6	TRGIE	触发中断使能 0: 禁止触发中断 1: 使能触发中断
5	CMTIE	换相更新中断使能 0: 禁止换相更新中断 1: 使能换相更新中断
4	CH3IE	通道3比较/捕获中断使能 0: 禁止通道3中断 1: 使能通道3中断
3	CH2IE	通道2比较/捕获中断使能 0: 禁止通道2中断 1: 使能通道2中断
2	CH1IE	通道1比较/捕获中断使能 0: 禁止通道1中断 1: 使能通道1中断
1	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 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	CH3OF	CH2OF	CH1OF	CH0OF	保留	BRKIF	TRGIF	CMTIF	CH3IF	CH2IF	CH1IF	CHOIF	UPIF	rc_w0	rc_w0

位/位域	名称	描述
31:13	保留	必须保持复位值
12	CH3OF	通道3捕获溢出标志 参见CH0OF描述
11	CH2OF	通道2捕获溢出标志 参见CH0OF描述
10	CH1OF	通道1捕获溢出标志 参见CH0OF描述
9	CH0OF	通道0捕获溢出标志 当通道0被配置为输入模式时，在CH0IF标志位已经被置1后，捕获事件再次发生时，该标志位可以由硬件置1。该标志位由软件清0。 0: 无捕获溢出中断发生 1: 发生了捕获溢出中断
8	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 此位在更新事件发生时由硬件置1, 软件清0。 0: 无更新中断发生 1: 发生更新中断

软件事件产生寄存器 (**TIMERx_SWEVG**)

地址偏移: 0x14

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								BRKG	TRGG	CMTG	CH3G	CH2G	CH1G	CH0G	UPG
w								w	w	w	w	w	w	w	w

位/位域	名称	描述
31:8	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，根据CCSE位（ TIMERx_CTL1 寄存器中）的值，通道捕获/比较控制寄存器（CHxEN, CHxNEN 和CHxCOMCTL）的互补输出被更新。 0：不产生通道换相更新事件 1：产生通道换相更新事件
4	CH3G	通道3捕获或比较事件发生 参见CH0G描述
3	CH2G	通道2捕获或比较事件发生 参见CH0G描述
2	CH1G	通道1捕获或比较事件发生 参见CH0G描述
1	CH0G	通道0捕获或比较事件发生 该位由软件置1，用于在通道0产生一个捕获/比较事件，由硬件自动清0。当此位被置1，CH0IF标志位被置1，若开启对应的中断和DMA，则发出相应的中断和DMA请求。此外，如果通道0配置为输入模式，计数器的当前值被捕获到 TIMERx_CH0CV 寄存器，如果CH0IF标志位已经为1，则CH0OF标志位被置1。 0：不产生通道0捕获或比较事件 1：发生通道0捕获或比较事件
0	UPG	更新事件产生 此位由软件置1，被硬件自动清0。当此位被置1，如果选择了中央对齐或向上计数模式，计数器被清0。否则(向下计数模式)计数器将载入自动重载值，预分频计数器将同时被清除。 0：无更新事件产生 1：产生更新事件

通道控制寄存器 0 (**TIMERx_CHCTL0**)

地址偏移： 0x18

复位值： 0x0000 0000

该寄存器只能按字(32位)访问。

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

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

CH1COM CEN	CH1COMCTL[2:0]	CH1COM SEN	CH1COM FEN	CH1MS[1:0]	CH0COM CEN	CH0COMCTL[2:0]	CH0COM SEN	CH0COM FEN	CH0MS[1:0]
CH1CAPFLT[3:0]	CH1CAPPSC[1:0]				CH0CAPFLT[3:0]	CH0CAPPSC[1:0]			

rw

rw

rw

rw

rw

rw

输出比较模式:

位/位域	名称	描述
31:16	保留	必须保持复位值
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ETIFP输入高电平时, O0CPRE参考信号被清0 0: 禁止通道0输出比较清零 1: 使能通道0输出比较清零
6:4	CH0COMCTL[2:0]	通道0输出比较模式 此位定义了O0CPRE的动作, 而O0CPRE决定了CH0_O、CH0_ON的值。O0CPRE高电平有效, 而CH0_O、CH0_ON的有效电平取决于CH0P、CH0NP位。 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。在向上计数时，一旦计数器值小于TIMERx_CH0CV时，O0CPRE为有效电平，否则为无效电平。在向下计数时，一旦计数器的值大于TIMERx_CH0CV时，O0CPRE为无效电平，否则为有效电平。

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

在PWM模式0或PWM模式1中，只有当比较结果改变了或者输出比较模式中从冻结模式切换到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</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PWM模式0或者PWM模式1，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，CH0_O被设置为比较电平而与比较结果无关。</p> <p>0: 禁止通道0输出比较快速。当触发器的输入有一个有效沿时，激活CH0_O输出的最小延时为5个时钟周期</p> <p>1: 使能通道0输出比较快速。当触发器的输入有一个有效沿时，激活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</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CI0FE0上。</p> <p>10: 通道0配置为输入，ISO映射在CI1FE0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
31:16	保留	必须保持复位值
15:12	CH1CAPFLT[3:0]	通道1输入捕获滤波控制 参见CH0CAPFLT描述
11:10	CH1CAPPSC[1:0]	通道1输入捕获预分频器

参见CH0CAPPSC描述

9:8	CH1MS[1:0]	通道1模式选择 与输出模式相同
7:4	CH0CAPFLT[3:0]	通道0输入捕获滤波控制 数字滤波器由一个事件计数器组成，N个输入事件后会产生一个输出的跳变。这些位定义了CIO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 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
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

CH3COM CEN	CH3COMCTL[2:0]	CH3COM SEN	CH3COM FEN	CH3MS[1:0]	CH2COM CEN	CH2COMCTL[2:0]	CH2COM SEN	CH2COM FEN	CH2MS[1:0]
CH3CAPFLT[3:0]	CH3CAPPSC[1:0]				CH2CAPFLT[3:0]	CH2CAPPSC[1:0]			

rw

rw

rw

rw

rw

rw

输出比较模式:

位/位域	名称	描述
31:16	保留	必须保持复位值
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。在向上计数时，一旦计数器值小于TIMERx_CH2CV时，O2CPRE 为有效电平，否则为无效电平。在向下计数时，一旦计数器的值大于TIMERx_CH2CV 时，O2CPRE 为无效电平，否则为有效电平。

111: PWM 模式1。在向上计数时，一旦计数器值小于TIMERx_CH2CV时，O2CPRE 为无效电平，否则为有效电平。在向下计数时，一旦计数器的值大于TIMERx_CH2CV 时，O2CPRE 为有效电平，否则为无效电平。

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PWM模式0或者PWM模式1，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，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
31:16	保留	必须保持复位值
15:12	CH3CAPFLT[3:0]	通道3输入捕获滤波控制 参见CH0CAPFLT描述
11:10	CH3CAPPSC[1:0]	通道3输入捕获预分频器

参见CH0CAPPSC描述

9:8	CH3MS[1:0]	通道3模式选择 与输出模式相同
7:4	CH2CAPFLT[3:0]	通道2输入捕获滤波控制 数字滤波器由一个事件计数器组成， N个输入事件后会产生一个输出的跳变。这些位定义了CI2输入信号的采样频率和数字滤波器的长度。 0000: 无滤波器, $f_{SAMP} = f_{DTS}$, $N=1$ 。 0001: $f_{SAMP} = f_{PCLK}$, $N=2$ 。 0010: $f_{SAMP} = f_{PCLK}$, $N=4$ 。 0011: $f_{SAMP} = f_{PCLK}$, $N=8$ 。 0100: $f_{SAMP} = f_{DTS}/2$, $N=6$ 。 0101: $f_{SAMP} = f_{DTS}/2$, $N=8$ 。 0110: $f_{SAMP} = f_{DTS}/4$, $N=6$ 。 0111: $f_{SAMP} = f_{DTS}/4$, $N=8$ 。 1000: $f_{SAMP} = f_{DTS}/8$, $N=6$ 。 1001: $f_{SAMP} = f_{DTS}/8$, $N=8$ 。 1010: $f_{SAMP} = f_{DTS}/16$, $N=5$ 。 1011: $f_{SAMP} = f_{DTS}/16$, $N=6$ 。 1100: $f_{SAMP} = f_{DTS}/16$, $N=8$ 。 1101: $f_{SAMP} = f_{DTS}/32$, $N=5$ 。 1110: $f_{SAMP} = f_{DTS}/32$, $N=6$ 。 1111: $f_{SAMP} = f_{DTS}/32$, $N=8$ 。
3:2	CH2CAPPSC[1:0]	通道2输入捕获预分频器 这2位定义了通道2输入的预分频系数。当TIMERx_CHCTL2寄存器中的CH2EN =0时，则预分频器复位。 00: 无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获。 01: 每2个事件触发一次捕获。 10: 每4个事件触发一次捕获。 11: 每8个事件触发一次捕获。
1:0	CH2MS[1:0]	通道2模式选择 与输出比较模式相同

通道控制寄存器 2 (TIMERx_CHCTL2)

地址偏移: 0x20

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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

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
	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	CH3P	通道3极性 参考CH0P描述
12	CH3EN	通道3使能 参考CH0EN描述
11	CH2NP	通道2互补输出极性 参考CH0NP描述
10	CH2NEN	通道2互补输出使能 参考CH0NEN描述
9	CH2P	通道2极性 参考CH0P描述
8	CH2EN	通道2使能 参考CH0EN描述
7	CH1NP	通道1互补输出极性 参考CH0NP描述
6	CH1NEN	通道1互补输出使能 参考CH0NEN描述
5	CH1P	通道1极性 参考CH0P描述
4	CH1EN	通道1使能 参考CH0EN描述
3	CH0NP	通道0互补输出极性 当通道0配置为输出模式，此位定义了互补输出信号的极性。 0：通道0高电平有效 1：通道0低电平有效 当通道0配置为输入模式时，此位和CH0P联合使用，作为输入信号CI0的极性选择控制信号。 当TIMERx_CCHP寄存器的PROT[1:0]=11或10时此位不能被更改。
2	CH0NEN	通道0互补输出使能 当通道0配置为输出模式时，将此位置1使能通道0的互补输出。 0：禁止通道0互补输出 1：使能通道0互补输出
1	CH0P	通道0极性

当通道0配置为输出模式时，此位定义了输出信号极性。

0：通道0高电平有效

1：通道0低电平有效

当通道0配置为输入模式时，此位定义了CI0信号极性。

CH0P将选择CI0FE0或者CI1FE0的有效边沿或者捕获极性。

CH0P=0：把CIxFE0的上升沿作为捕获或者从模式下触发的有效信号，并且CIxFE0不会被翻转。

CH0P=1：把CIxFE0的下降沿作为捕获或者从模式下触发的有效信号，并且CIxFE0会被翻转。

当TIMERx_CCHP寄存器的PROT[1:0]=11或10 时此位不能被更改。

0	CH0EN	通道0捕获/比较使能
		当通道0配置为输出模式时，将此位置1使能CH0_O信号有效。当通道0配置为输入模式时，将此位置1使能通道0上的捕获事件。
	0：禁止通道0	
	1：使能通道0	

计数器寄存器 (TIMERx_CNT)

地址偏移： 0x24

复位值： 0x0000 0000

该寄存器只能按字(32位)访问。

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
CNT[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器 (TIMERx_PSC)

地址偏移： 0x28

复位值： 0x0000 0000

该寄存器只能按字(32位)访问。

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															

PSC[15:0]

rw

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

计数器自动重载寄存器 (TIMERx_CAR)

地址偏移: 0x2C

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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
CARL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

重复计数寄存器 (TIMERx_CREP)

地址偏移: 0x30

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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								CREP[7:0]							

rw

位/位域	名称	描述
31:8	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 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
CH0VAL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH0VAL[15:0]	通道0的捕获或比较值 当通道0配置为输入模式时，这些位决定了上次捕获事件的计数器值，并且本寄存器为只读。 当通道0配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 1 捕获/比较寄存器 (**TIMERx_CH1CV**)

地址偏移: 0x38

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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
CH1VAL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH1VAL[15:0]	通道1的捕获或比较值 当通道1配置为输入模式时，这些位决定了上次捕获事件的计数器值，并且本寄存器为只读。

当通道1配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 2 捕获/比较寄存器 (**TIMERx_CH2CV**)

地址偏移: 0x3C

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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
CH2VAL[15:0]															

rw

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

通道 3 捕获/比较寄存器 (**TIMERx_CH3CV**)

地址偏移: 0x40

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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
CH3VAL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH3VAL[15:0]	通道3的捕获或比较值 当通道3配置为输入模式时，这些位决定了上次捕获事件的计数器值，并且本寄存器为只读。 当通道3配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

存器后，影子寄存器值随每次更新事件更新。

互补通道保护寄存器 (TIMERx_CCHP)

地址偏移: 0x44

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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

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	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	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BRKIN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时，使能中止事件和CKM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，CHx_O/CHx_ON的</p>

输出使能信号和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, CHx_O/CHx_ON的输出使能信号和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]	死区时间配置 这些位定义了插入互补输出之间的死区持续时间。DTCFG值和死区时间的关系如下: DTCFG[7:5] = 3'b0xx: DT value = DTCFG[7:0] * tDT, tDT=tDTS。 DTCFG[7:5] = 3'b10x: DT value = (64+DTCFG[5:0]) * tDT, tDT = tDTS*2。 DTCFG[7:5] = 3'b110: DT value = (32+DTCFG[4:0]) * tDT, tDT = tDTS*8。 DTCFG[7:5] = 3'b111: DT value = (32+DTCFG[4:0]) * tDT, tDT = tDTS*16。 此位只有在TIMERx_CCHP寄存器的PROT[1:0]=00时才可修改。

DMA 配置寄存器 (TIMERx_DMACFG)

地址偏移: 0x48

复位值: 0x0000 0000

该寄存器只能按字(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		DMATC[4:0]				保留		DTCFG[7:0]							
rw															

位/位域	名称	描述
------	----	----

31:13	保留	必须保持复位值
12:8	DMATC[4:0]	DMA传输计数 该位域定义了DMA访问（读/写） TIMERx_DMATB 寄存器的次数。 5'b00000: 1次传输 5'b00001: 2次传输 ... 5'b10001: 18次传输
7:5	保留	必须保持复位值
4:0	DMATA[4:0]	DMA传输起始地址 该位域定义了DMA访问 TIMERx_DMATB 寄存器的第一个地址。当第一次访问 TIMERx_DMATB 寄存器时，实际访问的就是该位域指定的地址。第二次访问 TIMERx_DMATB 时，将访问（起始地址+0x4）。 5'b00000: TIMERx_CTL0 5'b00001: TIMERx_CTL1 ... 5'b10010: TIMERx_DMACFG 总之：起始地址 = TIMERx_CTL0 + DMATA*4

DMA发送缓冲区寄存器 (TIMERx_DMATB**)**

地址偏移: 0x4C

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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
DMATB[15:0]															

rw

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

配置寄存器 (TIMERx_CFG**)**

地址偏移: 0xFC

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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 rw

位/位域	名称	描述
31:2	保留	必须保持复位值
1	CHVSEL	<p>写捕获比较寄存器选择位 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>

16.2. 通用定时器 L0 (TIMERx, x=1,2,3,4)

16.2.1. 简介

通用定时器L0 (TIMER1/2/3/4) 是4通道定时器，支持输入捕获，输出比较，产生PWM信号控制电机和电源管理。通用定时器L0计数器是16位无符号计数器。

通用定时器L0是可编程的，可以被用来计数，其外部事件可以驱动其他定时器。

定时器和定时器之间是相互独立，但是它们的计数器可以被同步在一起形成一个更大的定时器。

16.2.2. 主要特征

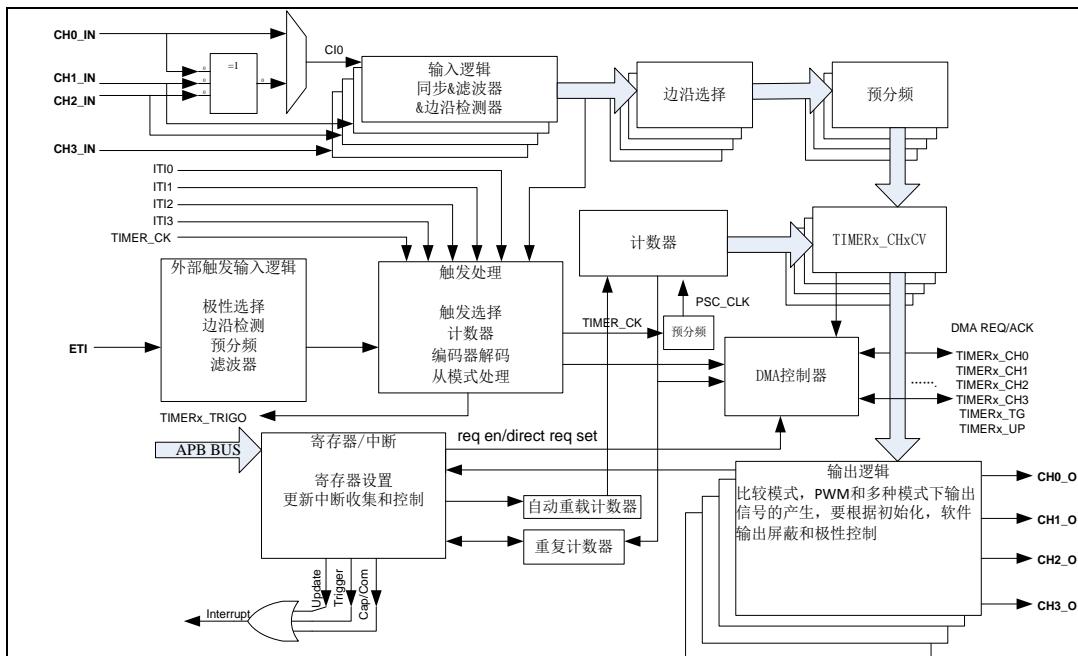
- 总通道数： 4；
- 计数器宽度： 16位；
- 时钟源可选： 内部时钟，内部触发，外部输入，外部触发；
- 多种计数模式： 向上计数，向下计数和中央计数；
- 正交编码器接口： 被用来追踪运动和分辨旋转方向和位置；
- 霍尔传感器接口： 用来做三相电机控制；
- 可编程的预分频器： 16位，运行时可以被改变；
- 每个通道可配置： 输入捕获模式，输出比较模式，可编程的PWM模式，单脉冲模式；
- 自动重装载功能；

- 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 16-34. 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 16-34. 通用定时器 L0 结构框图



16.2.4. 功能说明

时钟源选择

通用定时器L0可以是内部时钟源CK_TIMER, 或者是由SMC(TIMERx_SMCFG寄存器位[2:0])位确定的时钟源。

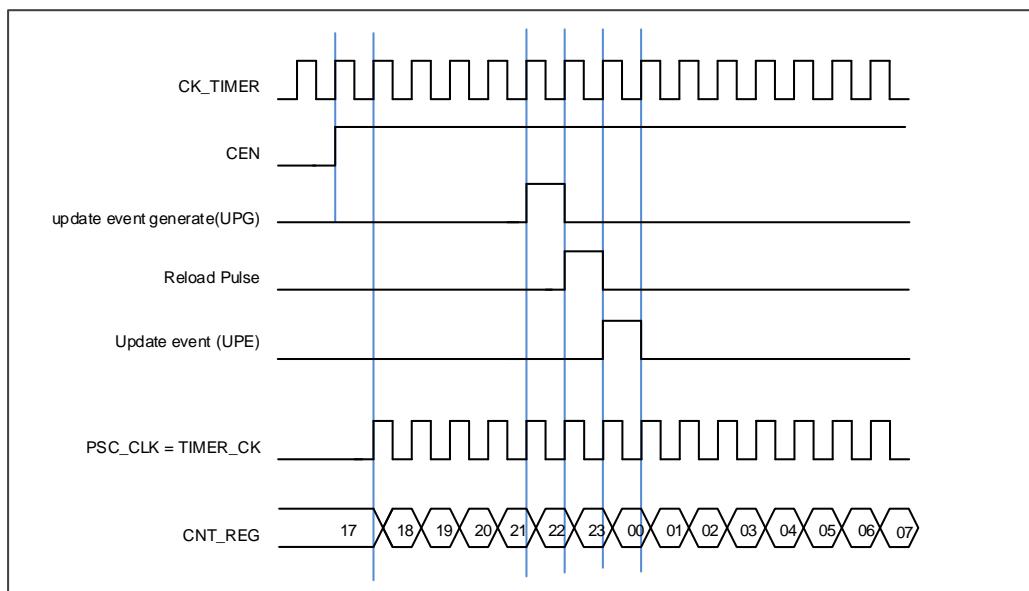
- SMC[2:0]=3'b000, 定时器选择内部时钟源 (连接到RCU模块的CK_TIMER)

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 16-35. 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_CH0/TIMERx_CH1引脚的每个上升沿或下降沿计数。这种模式可以通过设置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ITIO/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。

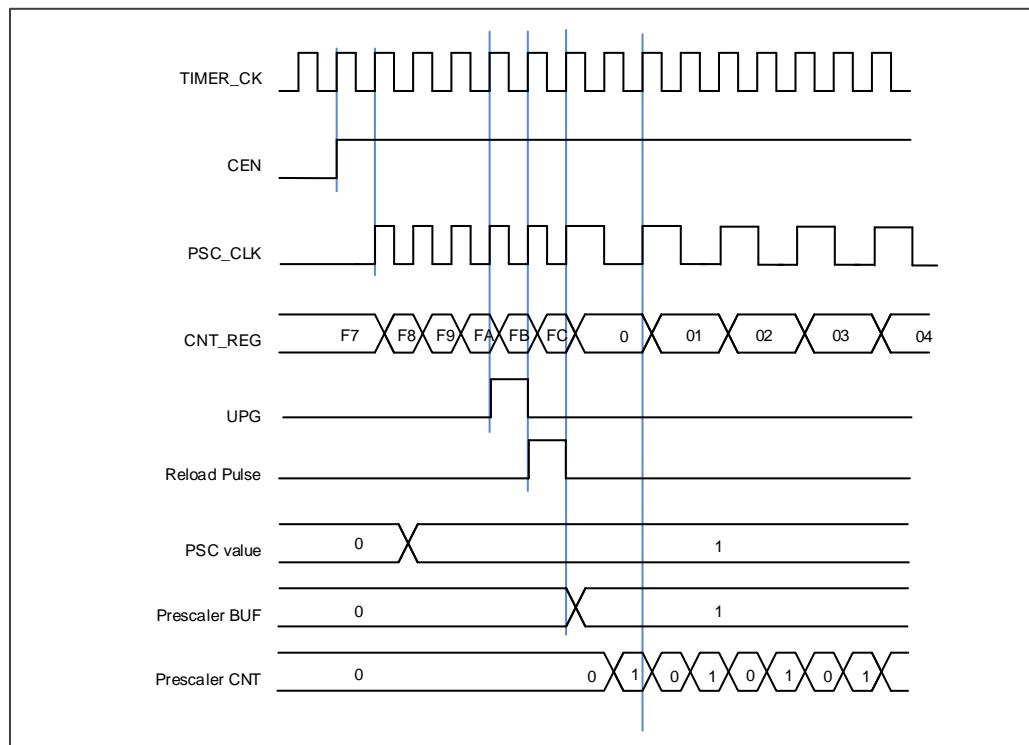
- SMC1=1'b1 (外部时钟模式1)，定时器选择外部输入引脚ETI作为时钟源。

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 16-36. 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(重复计数寄存器，自动重载寄存器，预分频寄存器)都将被更新。

[图 16-37. 向上计数时序图，PSC=0/1](#) 和 [图 16-38. 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 16-37. 向上计数时序图, PSC=0/1

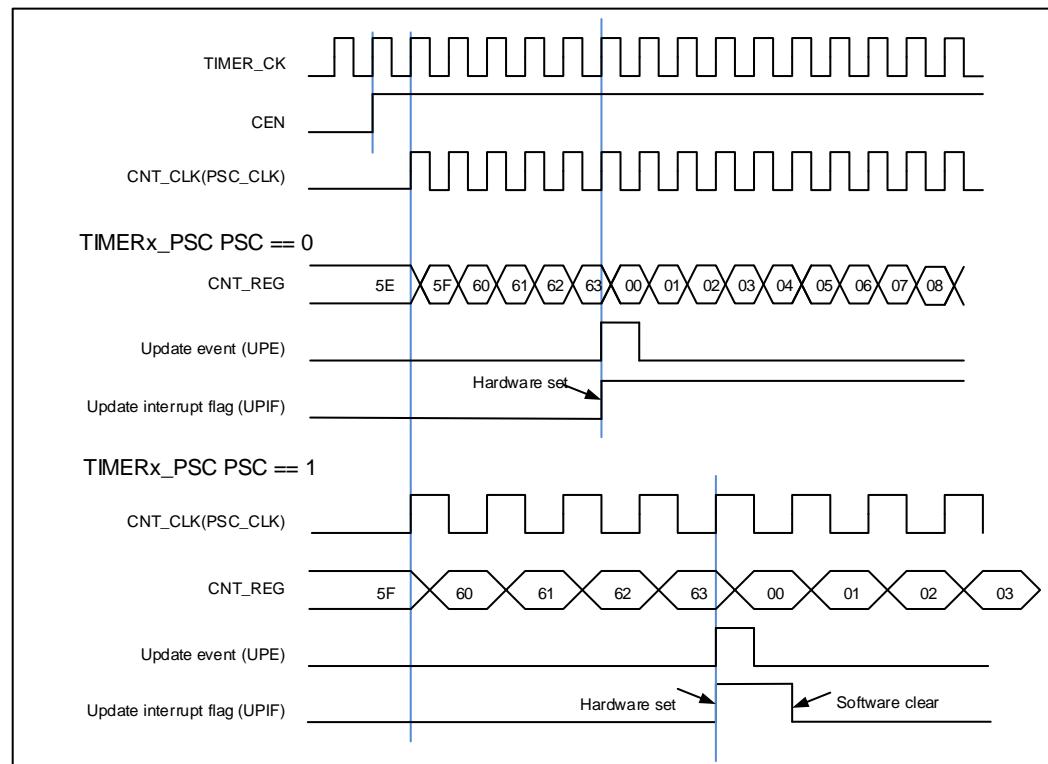
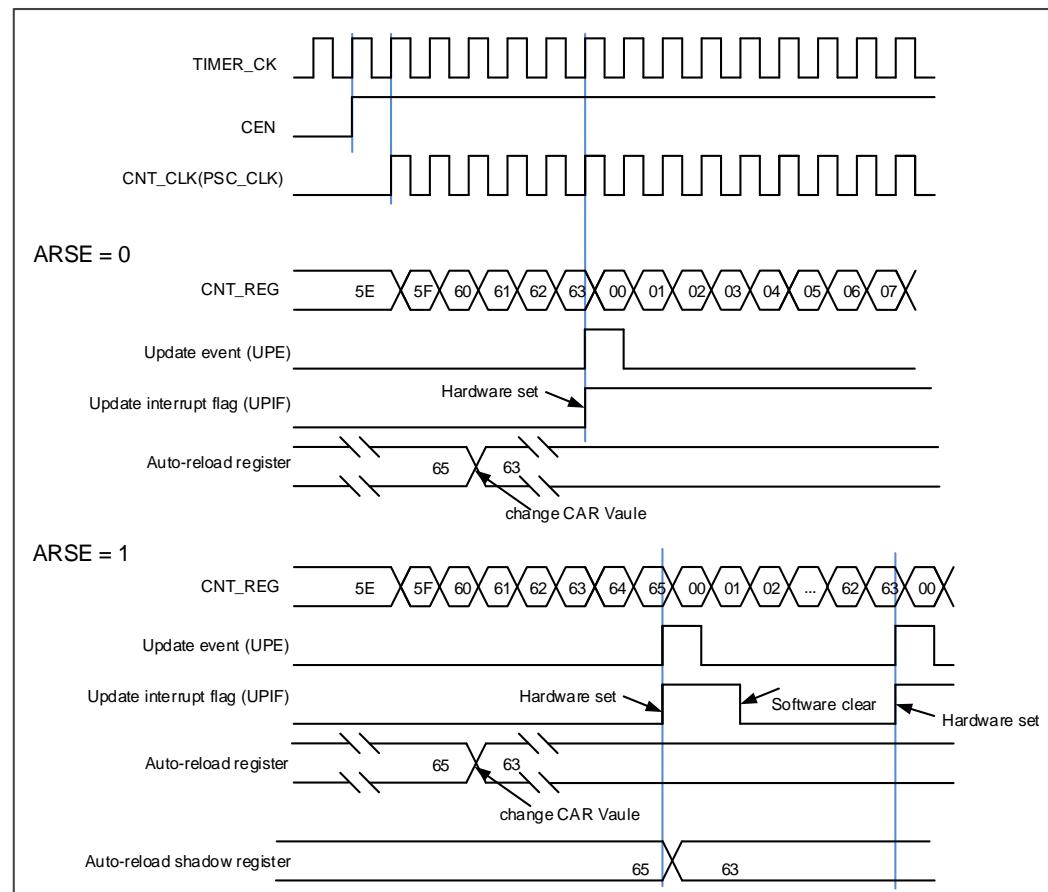


图 16-38. 向上计数时序图, 在运行时改变 TIMERx_CAR 寄存器的值



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。在向下计数模式中，TIMERx_CTL0寄存器中的计数方向控制位DIR应该被设置成1。

当通过TIMERx_SWEVG寄存器的UPG位置1来设置更新事件时，计数值会被初始化为自动加载值，并产生更新事件。

如果TIMERx_CTL0寄存器的UPDIS置1，则禁止更新事件。

当发生更新事件时，所有的寄存器(重复计数器，自动重载寄存器，预分频寄存器)都将被更新。

[图16-39. 向下计数时序图，PSC=0/1](#) 和 [图16-40. 向下计数时序图，在运行时改变TIMERx CAR寄存器值](#)给出了一些例子，当TIMERx_CAR=0x63时，计数器在不同时钟频率下的行为。

图 16-39. 向下计数时序图，PSC=0/1

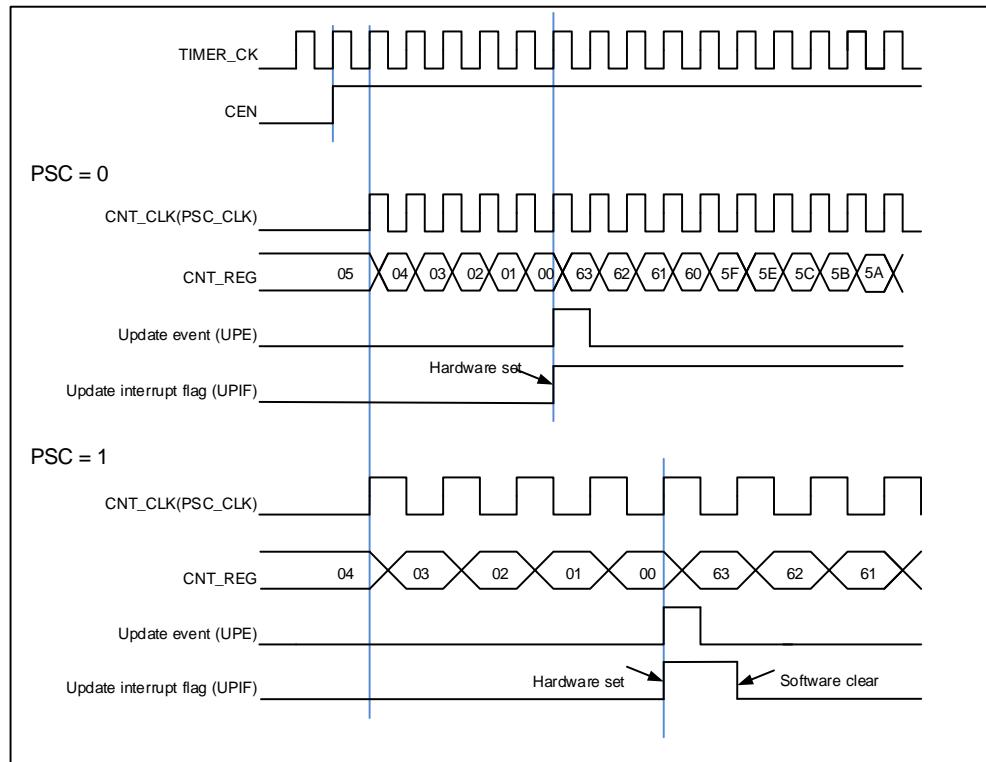
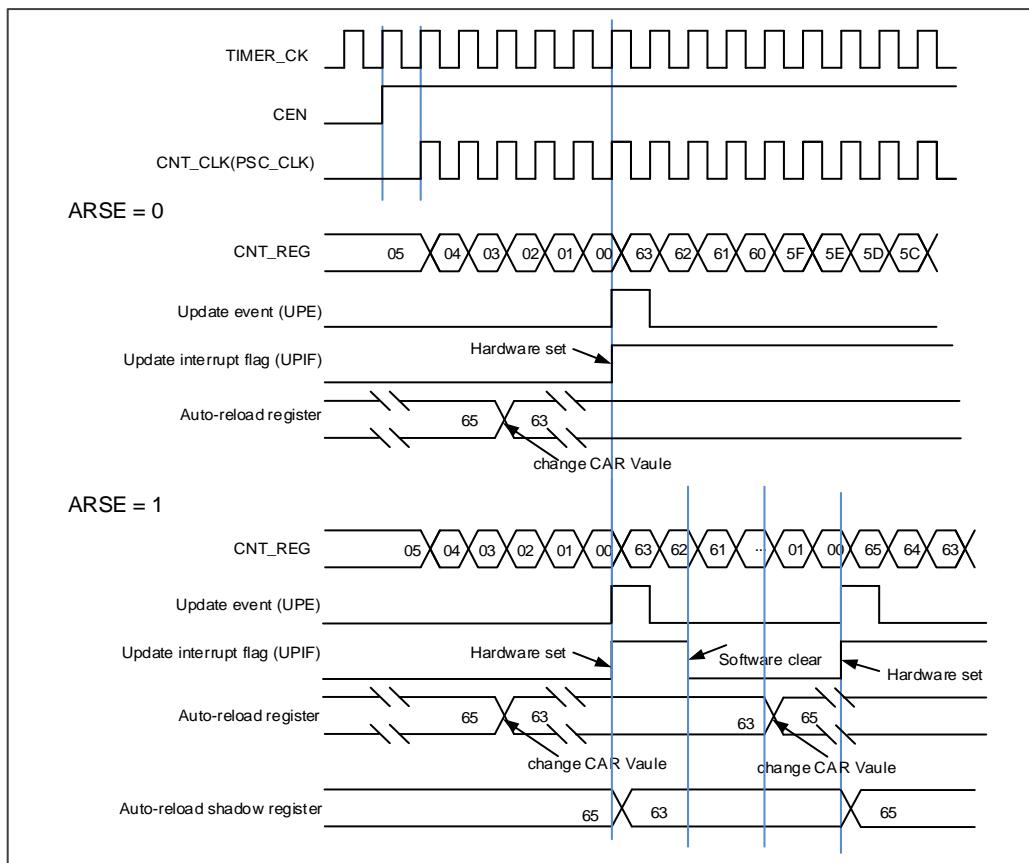


图 16-40. 向下计数时序图，在运行时改变 TIMERx_CAR 寄存器值



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。向上计数模式中，定时器模块在计数器计数到($\text{TIMERx_CREP}-1$)时产生一个上溢事件；向下计数模式中，定时器模块在计数器计数到1时产生一个下溢事件。在中央计数模式中，**TIMERx_CTL0**寄存器中的计数方向控制位DIR只读，表明计数方向。计数方向控制位DIR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。

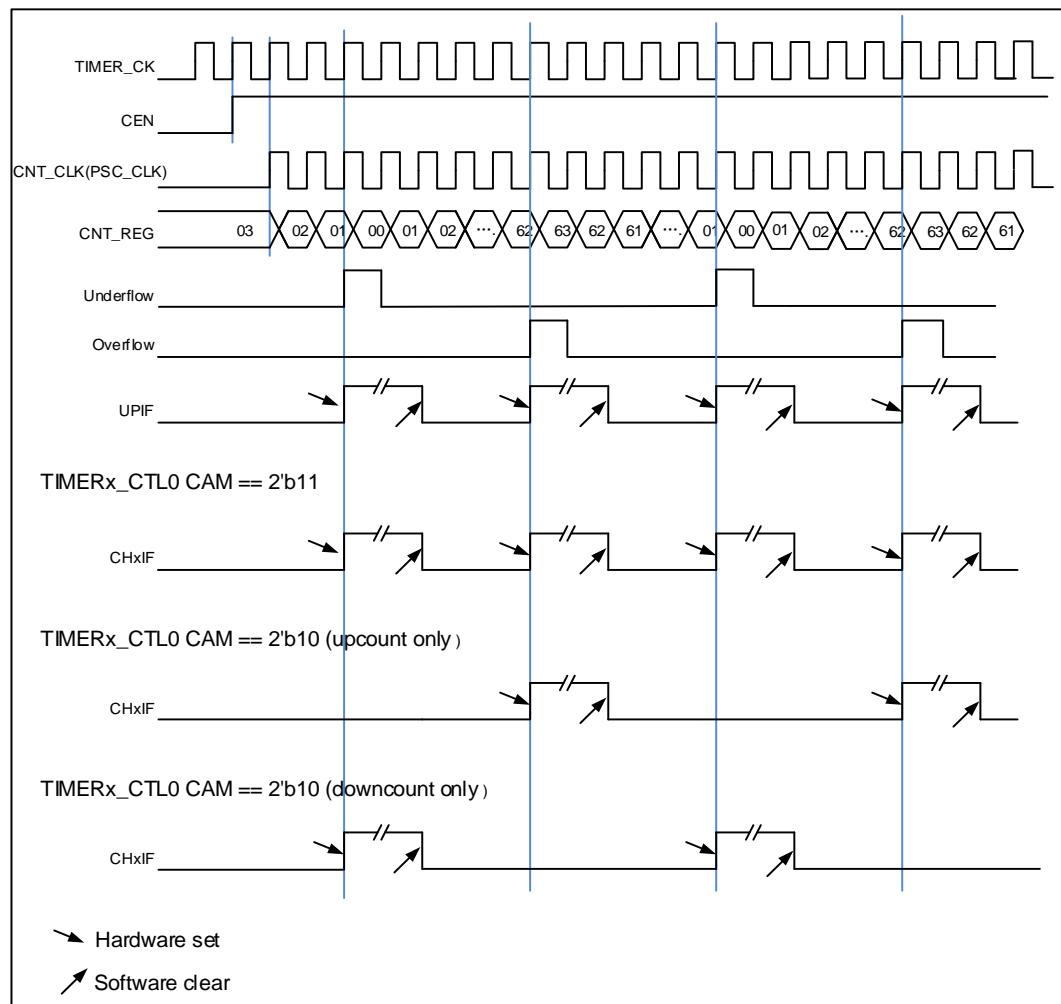
上溢或者下溢时，**TIMERx_INTF**寄存器中的UPIF位都会被置1。但是CHxIF位是否置1与**TIMERx_CTL0**寄存器中CAM的值有关。具体细节参考[图16-41. 中央计数模式计数器时序图](#)。

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。

[图16-41. 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 16-41. 中央计数模式计数器时序图



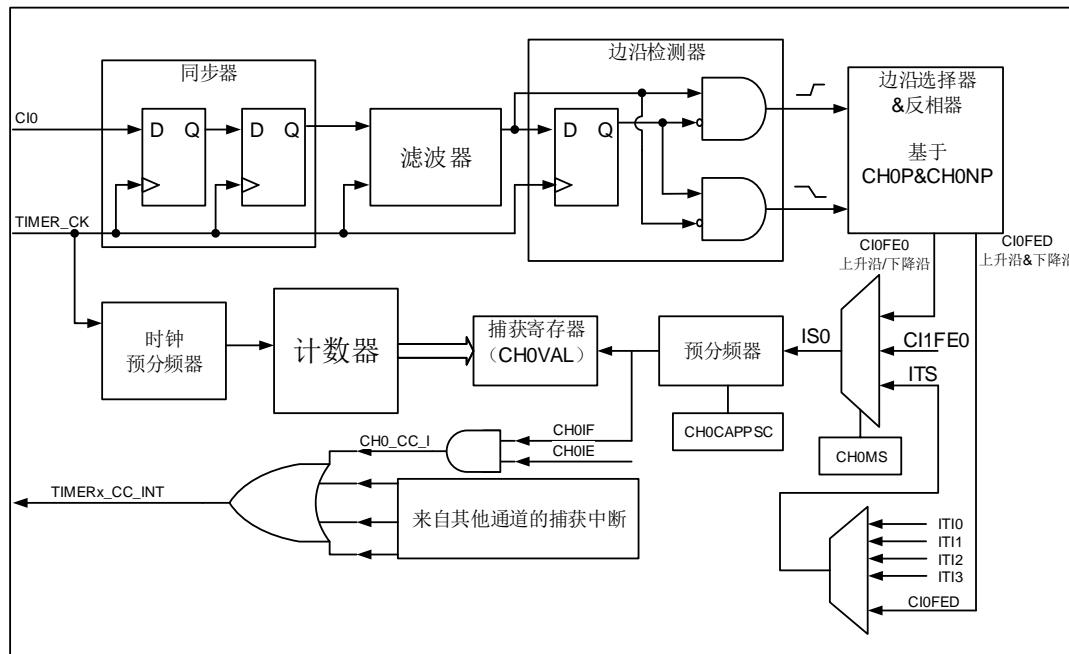
捕获/比较通道

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

输入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 16-42. 输入捕获逻辑



通道输入信号 CI_x 有两种选择，一种是 TIMER_x_CH_x 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 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 (TIMERRx_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 (TIMERRx_CHCTL2 寄存器中 CHxP):

配置 CHxP 选择上升沿或者下降沿。

第三步：捕获源选择 (TIMERRx_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 TIMERRx_CHxCV 寄存器不能再被写。

第四步：中断使能 (TIMERRx_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 (TIMERRx_CHCTL2 寄存器中 CHxEN)。

结果：当期望的输入信号发生时，TIMERRx_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 TIMERRx_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

图 16-43. 输出比较逻辑 (x=0,1,2,3)

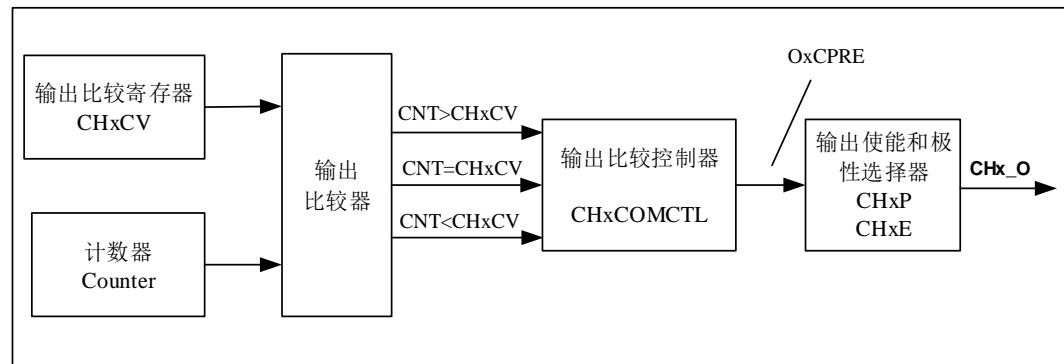


图 16-43. 输出比较逻辑 (x=0,1,2,3) 给出了输出比较的逻辑电路。通道输出信号CHx_O与OxCPRE信号（详情请见[通道输出准备信号](#)）的关系描述如下：OxCPRE信号高电平有效，CHx_O的输出情况与OxCPRE信号，CHxP位和CHxE位有关（具体情况请见TIMERx_CHCTL2寄存器中的描述）。例如，当设置CHxP=0（CHx_O高电平有效，与OxCPRE输出极性相同）、CHxE=1（CHx_O输出使能）时：

- 若OxCPRE输出有效（高）电平，则CHx_O输出有效（高）电平；
- 若OxCPRE输出无效（低）电平，则CHx_O输出无效（低）电平。

在输出比较模式，TIMERx可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的CHxCV寄存器与计数器的值匹配时，根据CHxCOMCTL的配置，这个通道的输出可以被置高电平，被置低电平或者反转。当计数器的值与CHxCV寄存器的值匹配时，CHxIF位被置1，如果CHxIE = 1则会产生中断，如果CxODE=1则会产生DMA请求。

配置步骤如下：

第一步：时钟配置：

配置定时器时钟源，预分频器等。

第二步：比较模式配置：

设置CHxCOMSEN位来配置输出比较影子寄存器；

设置CHxCOMCTL位来配置输出模式（置高电平/置低电平/反转）；

设置CHxP位来选择有效电平的极性；

设置CHxEN使能输出。

第三步：通过CHxIE/CxDDE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：

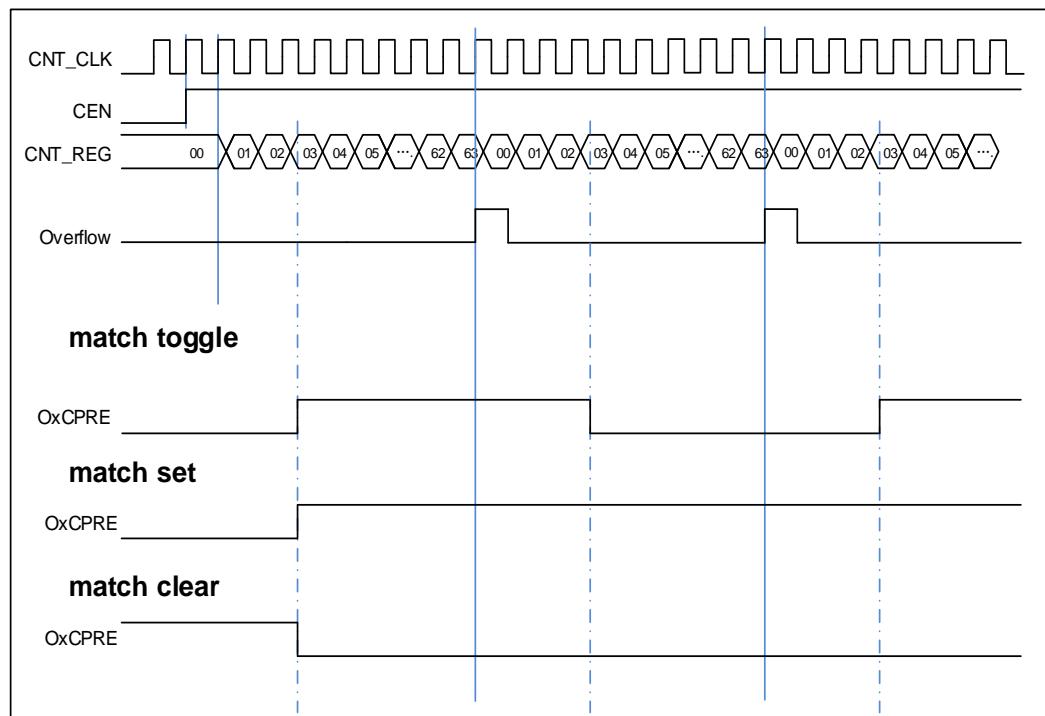
CHxVAL可以在运行时根据你所期望的波形而改变。

第五步：设置CEN位使能定时器。

图 16-44. 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 16-44. 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 [16-45. 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 [16-46. CAPWM 时序图](#) 显示了 CAPWM 的输出波形和中断。

在向上计数模式中, PWM 模式 0 下(CHxCOMCTL=3'b110), 如果 TIMERx_CHxCV 寄存器的值大于 TIMERx_CAR 寄存器的值, 通道输出一直为无效电平; PWM 模式 1 下(CHxCOMCTL=3'b111), 如果 TIMERx_CHxCV 寄存器的值大于 TIMERx_CAR 寄存器的值, 通道输出一直为有效电平。

图 16-45. EAPWM 时序图

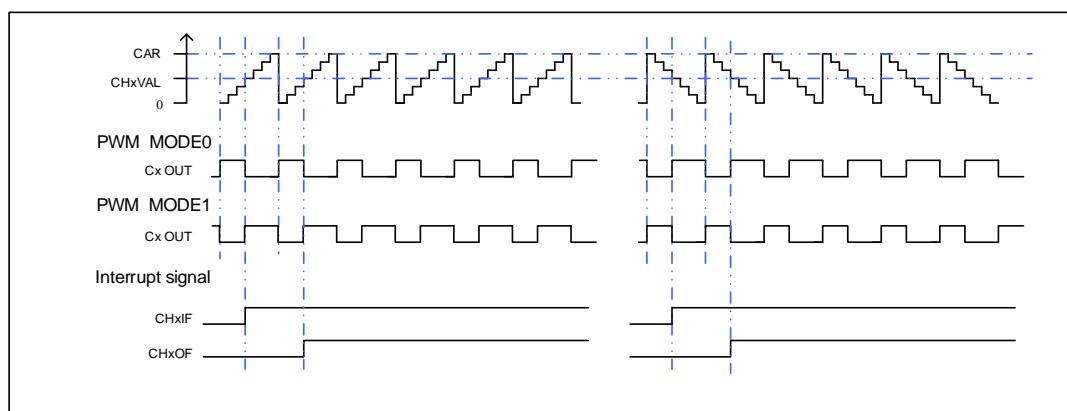
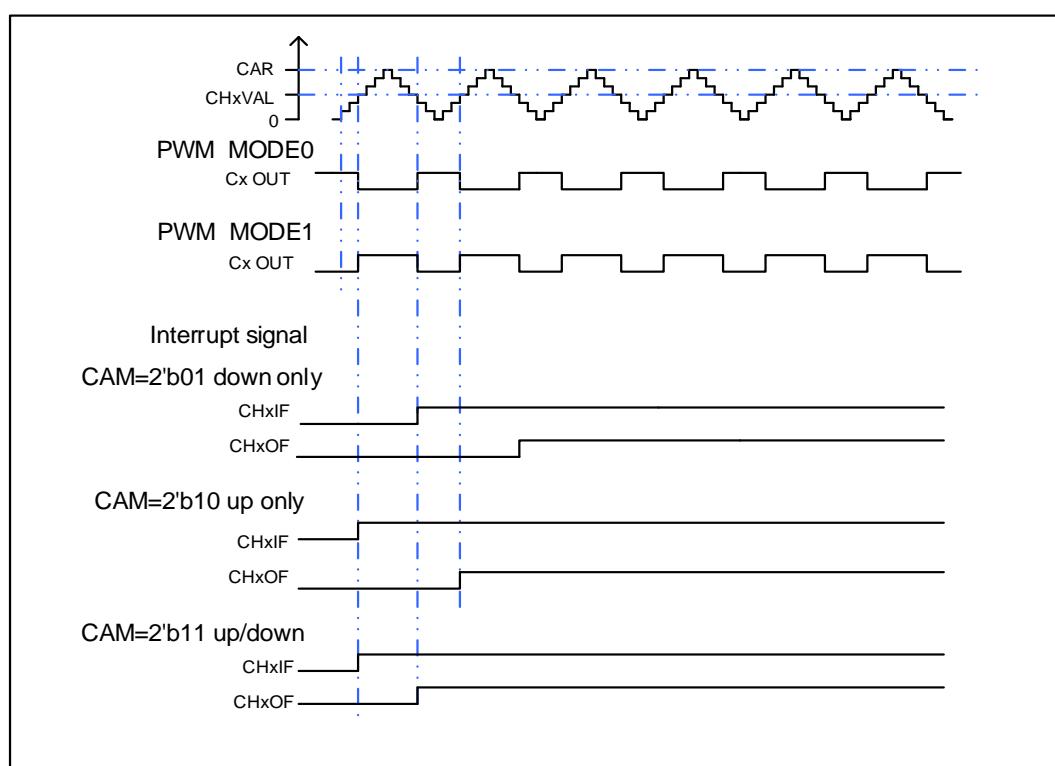


图 16-46. CAPWM 时序图



通道输出准备信号

根据 [图16-43. 输出比较逻辑 \(x=0,1,2,3\)](#) 所示，当 TIMERx 用于输出匹配比较模式下，在通道输出信号之前会产生一个中间信号 OxCPRE 信号（通道 x 输出准备信号）。设置 CHxCOMCTL 位可以定义 OxCPRE 信号类型。当 TIMERx 用于输出匹配比较模式下，设置 CHxCOMCTL 位可以定义 OxCPRE 信号(通道 x 输出准备信号)类型。OxCPRE 信号有若干类型的输出功能，包括，设置 CHxCOMCTL=0x00 可以保持原始电平；设置 CHxCOMCTL=0x01 可以将 OxCPRE 信号设置为高电平；设置 CHxCOMCTL=0x02 可以将 OxCPRE 信号设置为低电平；设置 CHxCOMCTL=0x03，在计数器值和 TIMERx_CHxCV 寄存器的值匹配时，可以翻转输出信号。

PWM 模式 0 和 PWM 模式 1 是 OxCPRE 的另一种输出类型，设置 CHxCOMCTL 位域为 0x06 或 0x07 可以配置 PWM 模式 0/PWM 模式 1。在这些模式中，根据计数器值和 TIMERx_CHxCV 寄存器值的关系以及计数方向，OxCPRE 信号改变其电平。具体细节描述，请参考相应的位。

设置 $\text{CHxCOMCTL}=0x04$ 或 $0x05$ 可以实现 OxCPRE 信号的强制输出功能。输出比较信号能够直接由软件强置为有效或无效状态，而不依赖于 TIMERx_CHxCV 的值和计数器值之间的比较结果。

设置 $\text{CHxCOMCEN}=1$ ，当由外部ETI引脚信号产生的ETIFP信号为高电平时， OxCPRE 被强制为低电平。在下一次更新事件到来时， OxCPRE 信号才会回到有效电平状态。

正交译码器

正交译码器功能使用由 TIMERx_CH0 和 TIMERx_CH1 引脚生成的 CI0 和 CI1 正交信号各自相互作用产生计数值。通过设置 $\text{SMC}=0x01$, $0x02$ 或 $0x03$ 来选择是仅由 CI0 ，仅由 CI1 ，或者由 CI0 和 CI1 来决定定时器的计数方向。在每个方向选择源的电平改变期间， DIR 位是由硬件自动改变的。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	CI0FE0		CI1FE1	
		上升	下降	上升	下降
只有 CI0	$\text{CI1FE1}=1$	向下	向上	-	-
	$\text{CI1FE1}=0$	向上	向下	-	-
只有 CI1	$\text{CI0FE0}=1$	-	-	向上	向下
	$\text{CI0FE0}=0$	-	-	向下	向上
CI0和CI1	$\text{CI1FE1}=1$	向下	向上	X	X
	$\text{CI1FE1}=0$	向上	向下	X	X
	$\text{CI0FE0}=1$	X	X	向上	向下
	$\text{CI0FE0}=0$	X	X	向下	向上

注意：“-”意思是“无计数”；“X”意思是不可能。

图 16-47. 编码器接口模式下计数器运行例子

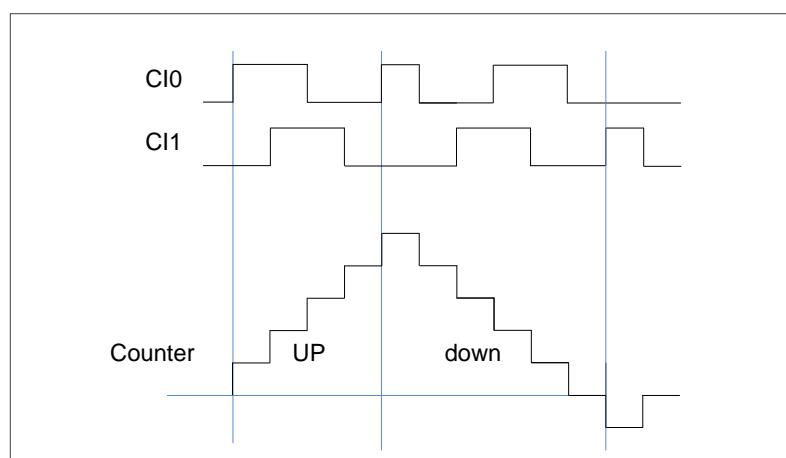
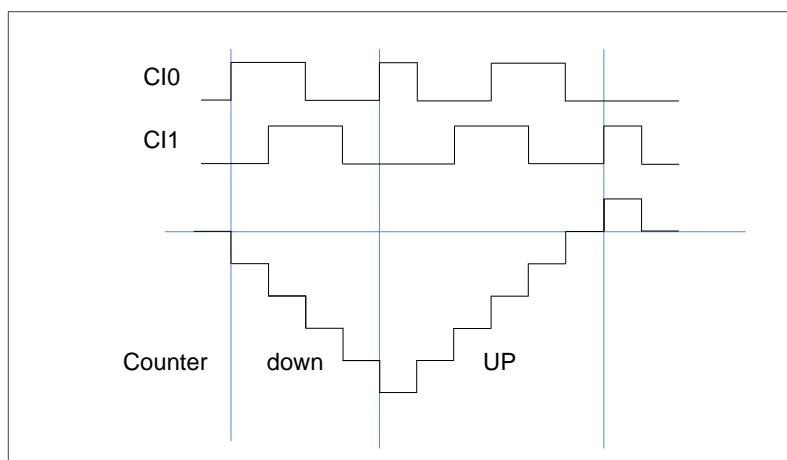


图 16-48. CI0FE0 极性反相的编码器接口模式下的例子



霍尔传感器接口功能

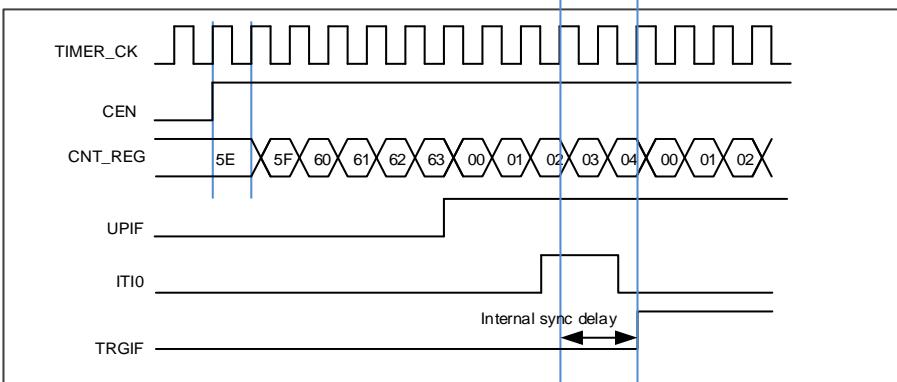
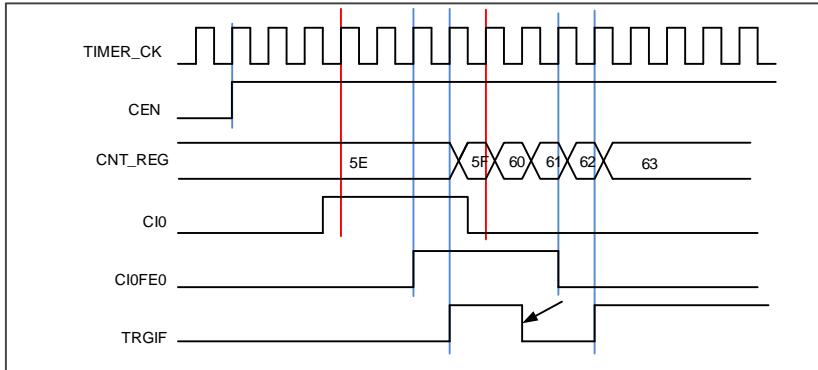
参考[高级定时器\(TIMERx,x=0,7\)霍尔传感器接口功能](#)

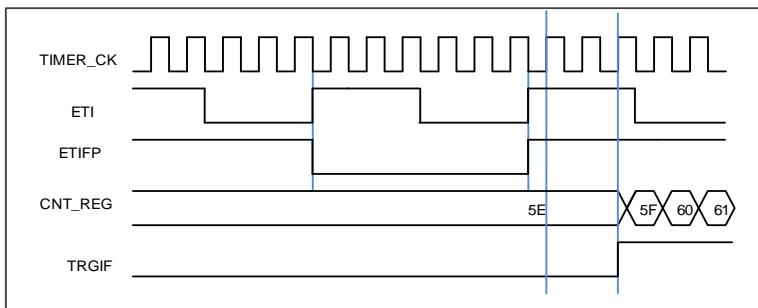
从控制器

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. 从模式示例（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 111: ETIFP	如果触发源是CI0FE0或者CI1FE1，配置CHxP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	若触发源为ITIx，滤波和预分频不可用。 若触发源为CIx，可配置CHxCAPFLT设置滤波，预分频不可用。 若触发源为ETIFP，滤波和预分频均可。
例1	复位模式 当触发输入上升沿到来时，计数器清零重启。	TRGS[2:0]=3'b000 选择ITI0为触发源。	若触发源是ITI0，极性选择不可用。	若触发源是ITI0，滤波和预分频不可用。

	模式选择	触发源选择	极性选择	滤波和预分频
	图 16-49. 复位模式			
				
	暂停模式 当触发输入为低的时候，计数器暂停计数，当触发输入为高时，计数器计数。	TRGS[2:0]=3'b101 选择CI0FE0为触发源。	TI0S=0 (非异或) [CH0P=0] CI0FE0不反相。捕获发生在上升沿。	在这个例子中滤波被旁路。
图 16-50. 暂停模式				
例2				
例3	事件模式 触发输入的上升沿计数器开始计数。	TRGS[2:0]=3'b111 选择ETIFP为触发源。	ETP = 0, ETI极性不变。 ETFC = 0, ETI 无滤波。	ETPSC = 1, ETI 2分频。

	模式选择	触发源选择	极性选择	滤波和预分频
图 16-51. 事件模式				
		 <p>The diagram shows the timing sequence for Event Mode. The TIMER_CK signal is a continuous square wave. The ETI signal has a single rising edge. The ETIFP signal has a single falling edge. The CNT_REG signal shows the count value starting at 5E and jumping to 60, then 61. The TRGIF signal has a single rising edge corresponding to the event.</p>		

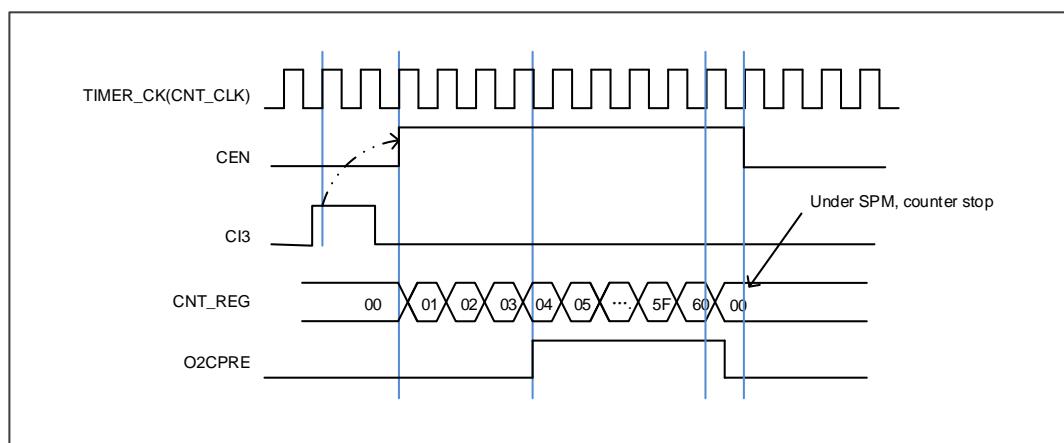
单脉冲模式

设置**TIMERx_CTL0**寄存器的SPM位置1，使能单脉冲模式。当SPM置1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置**CHxCOMCTL**配置**TIMERx**为PWM模式或者比较模式。

一旦设置定时器运行在单脉冲模式下，没有必要设置**TIMERx_CTL0**寄存器的定时器使能位**CEN=1**来使能计数器。触发信号沿或者软件写**CEN=1**都可以产生一个脉冲，此后**CEN**位一直保持为1直到更新事件发生或者**CEN**位被软件写0。如果**CEN**位被软件清0，计数器停止工作，计数值被保持。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PWM模式0或PWM模式1时**CHxCOMFEN**位才可用，触发源来源于触发信号。

图 16-52. 单脉冲模式，**TIMERx_CHxCV = 0x04**, **TIMERx_CAR=0x60**



定时器互连

参考[高级定时器\(TIMERx,x=0,7\)互连](#)。

定时器 DMA 模式

定时器DMA模式是指通过DMA模块配置定时器的寄存器。有两个跟定时器DMA模式相关的寄存器：TIMERx_DMACFG和TIMERx_DMATB。必须使能相应的DMA请求位，一些内部中断事件才可以产生DMA请求。当中断事件发生，TIMERx会给DMA发送请求。DMA配置成M2P（传输方向为从内存到外设）模式，PADDR（外设基地址）为TIMERx_DMATB寄存器地址，DMA就会访问TIMERx_DMATB寄存器。实际上，TIMERx_DMATB寄存器只是一个缓冲，定时器会将TIMERx_DMATB映射到一个内部寄存器，这个内部寄存器由TIMERx_DMACFG寄存器中的DMATA来指定。如果TIMERx_DMACFG寄存器的DMATC位域值为0，表示1次传输，定时器发送1个DMA请求就可以完成。如果TIMERx_DMACFG寄存器的DMATC位域值不为1，例如其值为3，表示4次传输，定时器就需要再多发3次DMA请求。在这3次请求下，DMA对TIMERx_DMATB寄存器的访问会映射到访问定时器的DMATA+0x4，DMATA+0x8，DMATA+0xC寄存器。总之，发生一次DMA内部中断请求，定时器会连续发送(DMATC+1)次请求。

如果再来1次DMA请求事件，TIMERx将会重复上面的过程。

定时器调试模式

当Cortex™-M4内核停止，DBG_CTL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 ($x=1,2,3,4$)

TIMER1基地址: 0x4000 0000

TIMER2基地址: 0x4000 0400

TIMER3基地址: 0x4000 0800

TIMER4基地址: 0x4000 0C00

控制寄存器 0 (TIMERx_CTL0)

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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				CKDIV[1:0]		ARSE		CAM[1:0]		DIR		SPM		UPS		UPDIS		CEN	
				RW		RW		RW		RW		RW		RW		RW			

位/位域	名称	描述
31:10	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TIMER_CK(定时器时钟)与DTS(死区时间和采样时钟)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: $f_{DTS}=f_{TIMER_CK}$</p> <p>01: $f_{DTS}=f_{TIMER_CK}/2$</p> <p>10: $f_{DTS}=f_{TIMER_CK}/4$</p> <p>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</p> <p>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。</p> <p>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。</p> <p>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。</p> <p>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</p>

断标志都会置1

当计数器使能以后，该位不能从0x00切换到非0x00。

4	DIR	方向 0: 向上计数 1: 向下计数 当计数器配置为中央对齐模式或编码器模式时，该位为只读。
3	SPM	单脉冲模式 0: 单脉冲模式禁能，更新事件发生后，计数器继续计数。 1: 单脉冲模式使能，在下一次更新事件发生时，CEN位硬件清零并且计数器停止计数。
2	UPS	更新请求源 软件配置该位，选择更新事件源 0: 使能后，下述任一事件产生更新中断或DMA请求： – UPG位被置1； – 计数器溢出/下溢； – 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。事件模式可以自动地通过硬件将CEN位置位。

控制寄存器 1 (TIMERx_CTL1)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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

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

位/位域	名称	描述
31:8	保留	必须保持复位值
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, TIMERx_CH1和TIMERx_CH2引脚异或的结果作为通道0的触发输入
6:4	MMC[2:0]	主模式控制 这些位控制TRGO信号的选择，TRGO信号由主定时器发给从定时器用于同步功能 000: 复位。TIMERx_SWEVG寄存器的UPG位被置1或从模式控制器产生复位触发一次TRGO脉冲，后一种情况下，TRGO上的信号相对实际的复位会有一个延迟。 001: 使能。此模式可用于同时启动多个定时器或控制在一段时间内使能从定时器。主模式控制器选择计数器使能信号作为触发输出TRGO。当CEN控制位被置1或者暂停模式下触发输入为高电平时，计数器使能信号被置1。在暂停模式下，计数器使能信号受控于触发输入，在触发输入和TRGO 上会有一个延迟，除非选择了主/从模式。 010: 更新。主模式控制器选择更新事件作为TRGO。 011: 捕获/比较脉冲。通道0在发生一次捕获或一次比较成功时，主模式控制器产生一个TRGO脉冲 100: 比较。在这种模式下，主模式控制器选择O0CPRE信号作为触发输出TRGO。 101: 比较。在这种模式下，主模式控制器选择O1CPRE信号作为触发输出TRGO。 110: 比较。在这种模式下，主模式控制器选择O2CPRE信号作为触发输出TRGO。 111: 比较。在这种模式下，主模式控制器选择O3CPRE信号作为触发输出TRGO。
3	DMAS	DMA请求源选择 0: 当通道捕获/比较事件发生时，发送通道x的DMA请求 1: 当更新事件发生，发送通道x的DMA请求
2:0	保留	必须保持复位值

从模式配置寄存器 (TIMERx_SMCFG)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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

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

ETP	SMC1	ETPSC[1:0]	ETFC[3:0]	MSM	TRGS[2:0]	保留	SMC[2:0]
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	ETP	<p>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</p>
14	SMC1	<p>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。但是不能将TRGS设为3'b111。 如果同时使能外部时钟模式0和外部时钟模式1, 外部时钟的输入是ETIFFP。 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 外部触发信号ETIFFP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ETIFFP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</p>
11:8	ETFC[3:0]	<p>外部触发滤波控制 数字滤波器是一个事件计数器, 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: $f_{SAMP} = f_{DTS}/32$, $N=5$。</p>

1110: $f_{SAMP}=f_{DTS}/32$, N=6。

1111: $f_{SAMP}=f_{DTS}/32$, N=8。

7	MSM	主-从模式 该位用来同步被选择的定时器同时开始计数。TRGI用做启动事件，通过TRGO，定时器被连接在一起。 0: 主从模式禁能 1: 主从模式使能
6:4	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定时器的触发输入源 000: 内部触发输入0 (ITI0) 001: I内部触发输入1 (ITI1) 010: 内部触发输入2 (ITI2) 011: 内部触发输入3 (ITI3) 100: CI0的边沿标志位 (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	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: 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。在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 0000

该寄存器只能按字(32位)访问。

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

保留	TRGDEN	保留	CH3DEN	CH2DEN	CH1DEN	CH0DEN	UPDEN	保留	TRGIE	保留	CH3IE	CH2IE	CH1IE	CHOIE	UPIE
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	TRGDEN	触发DMA请求使能 0: 禁止触发DMA请求 1: 使能触发DMA请求
13	保留	必须保持复位值
12	CH3DEN	通道3比较/捕获 DMA请求使能 0: 禁止通道3比较/捕获DMA请求 1: 使能通道3比较/捕获DMA请求
11	CH2DEN	通道2比较/捕获 DMA请求使能 0: 禁止通道2比较/捕获DMA请求 1: 使能通道2比较/捕获DMA请求
10	CH1DEN	通道1比较/捕获 DMA请求使能 0: 禁止通道1比较/捕获DMA请求 1: 使能通道1比较/捕获DMA请求
9	CH0DEN	通道0比较/捕获 DMA请求使能 0: 禁止通道0比较/捕获DMA请求 1: 使能通道0比较/捕获DMA请求
8	UPDEN	更新DMA请求使能 0: 禁止更新DMA请求 1: 使能更新DMA请求
7	保留	必须保持复位值
6	TRGIE	触发中断使能 0: 禁止触发中断 1: 使能触发中断
5	保留	必须保持复位值
4	CH3IE	通道3比较/捕获中断使能 0: 禁止通道3中断 1: 使能通道3中断
3	CH2IE	通道2比较/捕获中断使能 0: 禁止通道2中断 1: 使能通道2中断
2	CH1IE	通道1比较/捕获中断使能 0: 禁止通道1中断

1: 使能通道1中断

1	CH0IE	通道0比较/捕获中断使能 0: 禁止通道0中断 1: 使能通道0中断
0	UPIE	更新中断使能 0: 禁止更新中断 1: 使能更新中断

中断标志寄存器 (TIMERx_INTF)

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:13	保留	必须保持复位值
12	CH3OF	通道3捕获溢出标志 参见CH0OF描述
11	CH2OF	通道2捕获溢出标志 参见CH0OF描述
10	CH1OF	通道1捕获溢出标志 参见CH0OF描述
9	CH0OF	通道0捕获溢出标志 当通道0被配置为输入模式时，在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
8:7	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CH0IF描述
3	CH2IF	通道2比较/捕获中断标志 参见CH0IF描述
2	CH1IF	通道1比较/捕获中断标志 参见CH0IF描述
1	CH0IF	通道0比较/捕获中断标志 此标志由硬件置1软件清0。当通道0在输入模式下时，捕获事件发生时此标志位被置1；当通道0在输出模式下时，此标志位在一个比较事件发生时被置1。 0: 无通道0中断发生 1: 通道0中断发生
0	UPIF	更新中断标志 此位在更新事件发生时由硬件置1，软件清0。 0: 无更新中断发生 1: 发生更新中断

软件事件产生寄存器 (**TIMERx_SWEVG**)

地址偏移: 0x14

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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								TRGG	保留	CH3G	CH2G	CH1G	CH0G	UPG	

位/位域	名称	描述
31:7	保留	必须保持复位值
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	保留	必须保持复位值

4	CH3G	通道3捕获或比较事件发生 参见CH0G描述
3	CH2G	通道2捕获或比较事件发生 参见CH0G描述
2	CH1G	通道1捕获或比较事件发生 参见CH0G描述
1	CH0G	通道0捕获或比较事件发生 该位由软件置1，用于在通道0产生一个捕获/比较事件，由硬件自动清0。当此位被置1，CH0IF标志位被置1，若开启对应的中断和DMA，则发出相应的中断和DMA请求。此外，如果通道0配置为输入模式，计数器的当前值被捕获到TIMERx_CH0CV寄存器，如果CH0IF标志位已经为1，则CH0OF标志位被置1。 0：不产生通道0捕获或比较事件 1：发生通道0捕获或比较事件
0	UPG	更新事件产生 此位由软件置1，被硬件自动清0。当此位被置1，如果选择了中央对齐或向上计数模式，计数器被清0。否则(向下计数模式)计数器将载入自动重载值，预分频计数器将同时被清除。 0：无更新事件产生 1：产生更新事件

通道控制寄存器 0 (TIMERx_CHCTL0)

地址偏移: 0x18

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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

输出比较模式:

位/位域	名称	描述
31:16	保留	必须保持复位值
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。在向上计数时, 一旦计数器值小于TIMERx_CH0CV时, O0CPRE为有效电平, 否则为无效电平。在向下计数时, 一旦计数器的值大于TIMERx_CH0CV时, O0CPRE为无效电平, 否则为有效电平。 111: PWM模式1。在向上计数时, 一旦计数器值小于TIMERx_CH0CV时, O0CPRE为无效电平, 否则为有效电平。在向下计数时, 一旦计数器的值大于TIMERx_CH0CV时, O0CPRE为有效电平, 否则为无效电平。 在PWM模式0或PWM模式1中, 只有当比较结果改变了或者输出比较模式中从冻结模式切换到PWM模式时, CxCOMR电平才改变。
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
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PWM模式0或者PWM模式1, 会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配, 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
31:16	保留	必须保持复位值
15:12	CH1CAPFLT[3:0]	通道1输入捕获滤波控制 参见CH0CAPFLT描述
11:10	CH1CAPPSC[1:0]	通道1输入捕获预分频器 参见CH0CAPPSC描述
9:8	CH1MS[1:0]	通道1模式选择 与输出模式相同
7:4	CH0CAPFLT[3:0]	通道0输入捕获滤波控制 数字滤波器由一个事件计数器组成, N个输入事件后会产生一个输出的跳变。这些位定义了CI0输入信号的采样频率和数字滤波器的长度。 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: $f_{SAMP}=f_{DTS}/4$, 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 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		
CH3COM CEN	CH3COMCTL[2:0]		CH3COM SEN	CH3COM FEN	CH3MS[1:0]	CH2COM CEN	CH2COMCTL[2:0]		CH2COM SEN	CH2COM FEN	CH2MS[1:0]						
CH3CAPFLT[3:0]		CH3CAPPSC[1:0]		CH2CAPFLT[3:0]		CH2CAPPSC[1:0]											
rw		rw		rw		rw		rw		rw		rw					

输出比较模式:

位/位域	名称	描述
31:16	保留	必须保持复位值
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ETIFP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。在向上计数时, 一旦计数器值小于TIMERx_CH2CV时, O2CPRE为有效电平, 否则为无效电平。在向下计数时, 一旦计数器的值大于TIMERx_CH2CV时, O2CPRE 为无效电平, 否则为有效电平。 111: PWM 模式1。在向上计数时, 一旦计数器值小于TIMERx_CH2CV时, O2CPRE为无效电平, 否则为有效电平。在向下计数时, 一旦计数器的值大于TIMERx_CH2CV时, O2CPRE为有效电平, 否则为无效电平。 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。
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

2	CH2COMFEN	通道2输出比较快速使能 当该位为1时, 如果通道配置为PWM模式0或者PWM模式1, 会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效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
31:16	保留	必须保持复位值
15:12	CH3CAPFLT[3:0]	通道3输入捕获滤波控制 参见CH0CAPFLT描述
11:10	CH3CAPPSC[1:0]	通道3输入捕获预分频器 参见CH0CAPPSC描述
9:8	CH3MS[1:0]	通道3模式选择 与输出模式相同
7:4	CH2CAPFLT[3:0]	通道2输入捕获滤波控制 数字滤波器由一个事件计数器组成, N个输入事件后会产生一个输出的跳变。这些位定义了CI2输入信号的采样频率和数字滤波器的长度。 0000: 无滤波器, 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: $f_{SAMP}=f_{DTS}/8$, N=8。
- 1010: $f_{SAMP}=f_{DTS}/16$, N=5。
- 1011: $f_{SAMP}=f_{DTS}/16$, N=6。
- 1100: $f_{SAMP}=f_{DTS}/16$, N=8。
- 1101: $f_{SAMP}=f_{DTS}/32$, N=5。
- 1110: $f_{SAMP}=f_{DTS}/32$, N=6。
- 1111: $f_{SAMP}=f_{DTS}/32$, N=8。

3:2	CH2CAPPSC[1:0]	通道2输入捕获预分频器 这2位定义了通道2输入的预分频系数。当TIMERx_CHCTL2寄存器中的CH2EN =0时，则预分频器复位。 00: 无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获。 01: 每2个事件触发一次捕获。 10: 每4个事件触发一次捕获。 11: 每8个事件触发一次捕获。
1:0	CH2MS[1:0]	通道2模式选择 与输出比较模式相同

通道控制寄存器 2 (TIMERx_CHCTL2)

地址偏移: 0x20

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	CH3P	CH3EN	保留	CH2P	CH2EN	保留	CH1P	CH1EN	保留	CH0P	CH0EN	保留	CH0P	CH0EN	保留

位/位域	名称	描述
31:14	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
11:10	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

7:6	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
3:2	保留	必须保持复位值
1	CH0P	通道0极性 当通道0配置为输出模式时，此位定义了输出信号极性。 0: 通道0高电平有效 1: 通道0低电平有效 当通道0配置为输入模式时，此位定义了CI0信号极性 CH0P将选择CI0FE0或者CI1FE0的有效边沿或者捕获极性 CH0P=0: 把CIxFE0的上升沿作为捕获或者从模式下触发的有效信号，并且CIxFE0不会被翻转。 CH0P=1: 把CIxFE0的下降沿作为捕获或者从模式下触发的有效信号，并且CIxFE0会被翻转。
0	CH0EN	通道0捕获/比较使能 当通道0配置为输出模式时，将此位置1使能CH0_O信号有效。当通道0配置为输入模式时，将此位置1使能通道0上的捕获事件。 0: 禁止通道0 1: 使能通道0

计数器寄存器 (TIMERx_CNT)

地址偏移: 0x24

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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
CNT[15:0]															

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器 (TIMERx_PSC)

地址偏移: 0x28

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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
PSC[15:0]															

rw

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

计数器自动重载寄存器 (TIMERx_CAR)

地址偏移: 0x2C

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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
CARL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

通道 0 捕获/比较值寄存器 (TIMERx_CH0CV)

地址偏移: 0x34

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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
CH0VAL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH0VAL[15:0]	通道0的捕获或比较值 当通道0配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道0配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 1 捕获/比较值寄存器 (**TIMERx_CH1CV**)

地址偏移: 0x38

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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
CH1VAL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH1VAL[15:0]	通道1的捕获或比较值 当通道1配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道1配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 2 捕获/比较值寄存器 (**TIMERx_CH2CV**)

地址偏移: 0x3C

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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
CH2VAL[15:0]															

rw

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

通道 3 捕获/比较值寄存器 (**TIMERx_CH3CV**)

地址偏移: 0x40

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
CH3VAL[15:0]															

rw

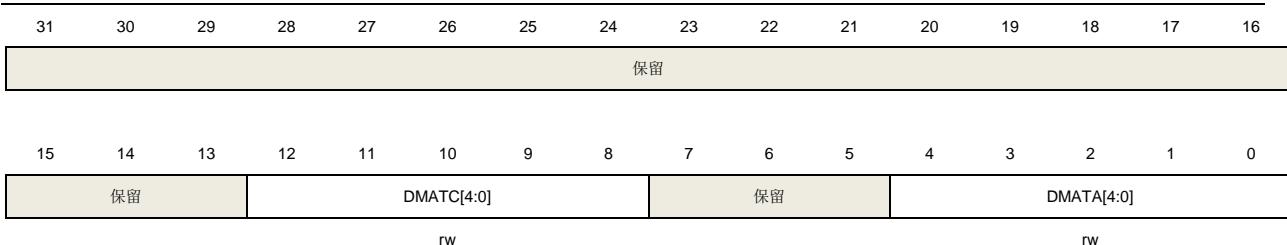
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH3VAL[15:0]	通道3的捕获或比较值 当通道3配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道3配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

DMA 配置寄存器 (**TIMERx_DMACFG**)

地址偏移: 0x48

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。



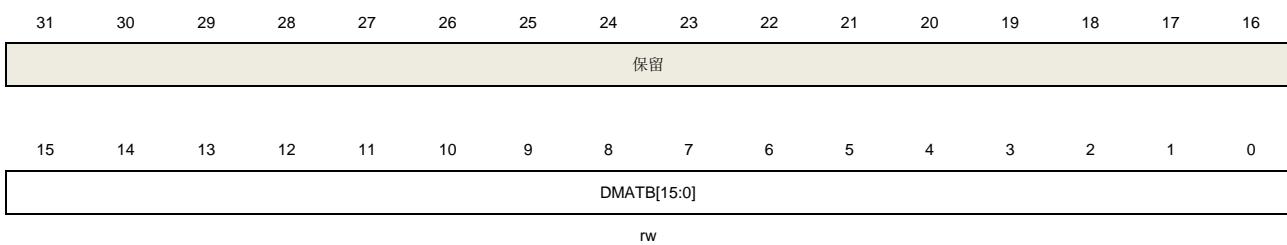
位/位域	名称	描述
31:13	保留	必须保持复位值
12:8	DMATC[4:0]	<p>DMA传输计数</p> <p>该位域定义了DMA访问（读写）TIMERx_DMATB寄存器的数量</p> <p>5'b00000: 1次传输</p> <p>5'b00001: 2次传输</p> <p>...</p> <p>5'b10001: 18次传输</p>
7:5	保留	必须保持复位值
4:0	DMATA[4:0]	<p>DMA传输起始地址</p> <p>该位域定义了DMA访问TIMERx_DMATB寄存器的第一个地址。当通过TIMERx_DMATB第一次访问时，访问的就是该位域指定的地址。第二次访问TIMERx_DMATB时，将访问起始地址+0x4。</p> <p>5'b00000: TIMERx_CTL0</p> <p>5'b00001: TIMERx_CTL1</p> <p>...</p> <p>5'b10010: TIMERx_DMACFG</p> <p>总之：起始地址 = TIMERx_CTL0 + DMATA*4</p>

DMA 发送缓冲区寄存器 (TIMERx_DMATB)

地址偏移: 0x4C

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。



位/位域	名称	描述
------	----	----

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

配置寄存器 (TIMERx_CFG)

地址偏移: 0xFC

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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															CHVSEL

rw

位/位域	名称	描述
31:2	保留	必须保持复位值
1	CHVSEL	写捕获比较寄存器选择位 此位由软件写1或清0。 1: 当写入捕获比较寄存器的值与寄存器当前值相等时，写入操作无效 0: 无影响
0	保留	必须保持复位值

16.3. 通用定时器 L1 (TIMERx, x=8,11)

16.3.1. 简介

通用定时器L1(TIMER8/11)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L1含有一个16位无符号计数器。

通用定时器L1是可编程的，可以用来计数，其外部事件可以驱动其他定时器

定时器和定时器之间是相互独立，但是它们的计数器可以被同步在一起形成一个更大的定时器。

16.3.2. 主要特征

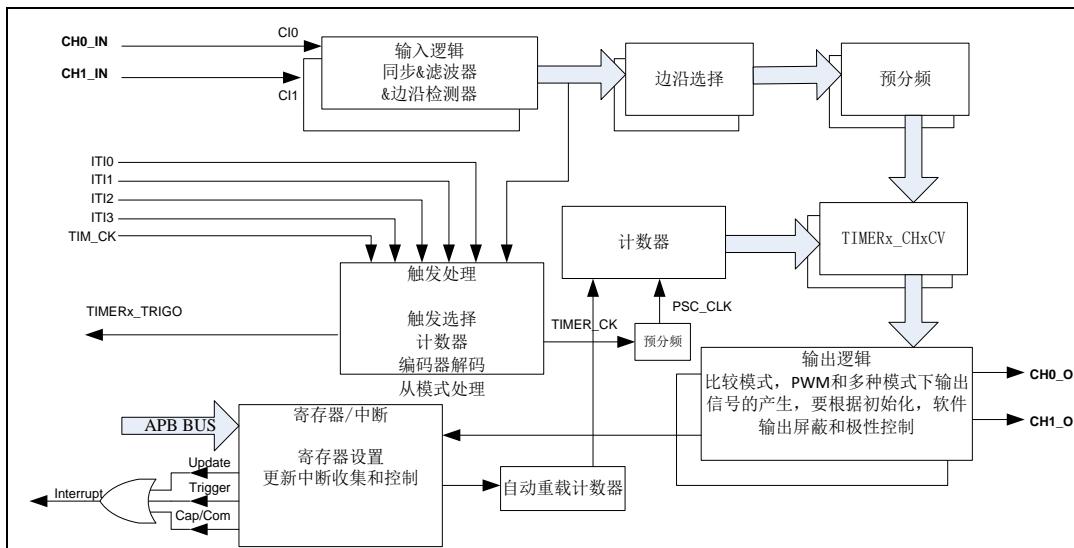
- 总通道数: 2;
- 计数器宽度: 16位;
- 时钟源可选: 内部时钟, 内部触发, 外部输入;

- 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；
- 自动重装载功能；
- 中断输出：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。

16.3.3. 结构框图

[图 16-53. 通用定时器 L1 结构框图](#)提供了通用定时器 L1 的内部配置细节。

图 16-53. 通用定时器 L1 结构框图



16.3.4. 功能说明

时钟源选择

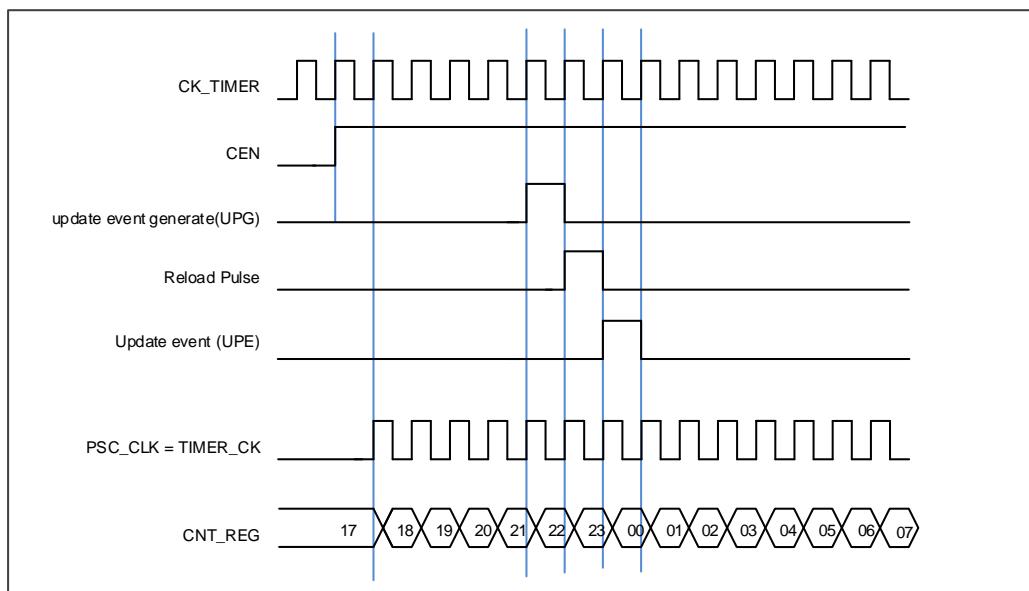
通用定时器L1可以是内部时钟源CK_TIMER, 或者是由SMC(TIMERx_SMCFG寄存器位[2:0])位确定的时钟源。

- 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_CLK驱动。

图 16-54. 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。

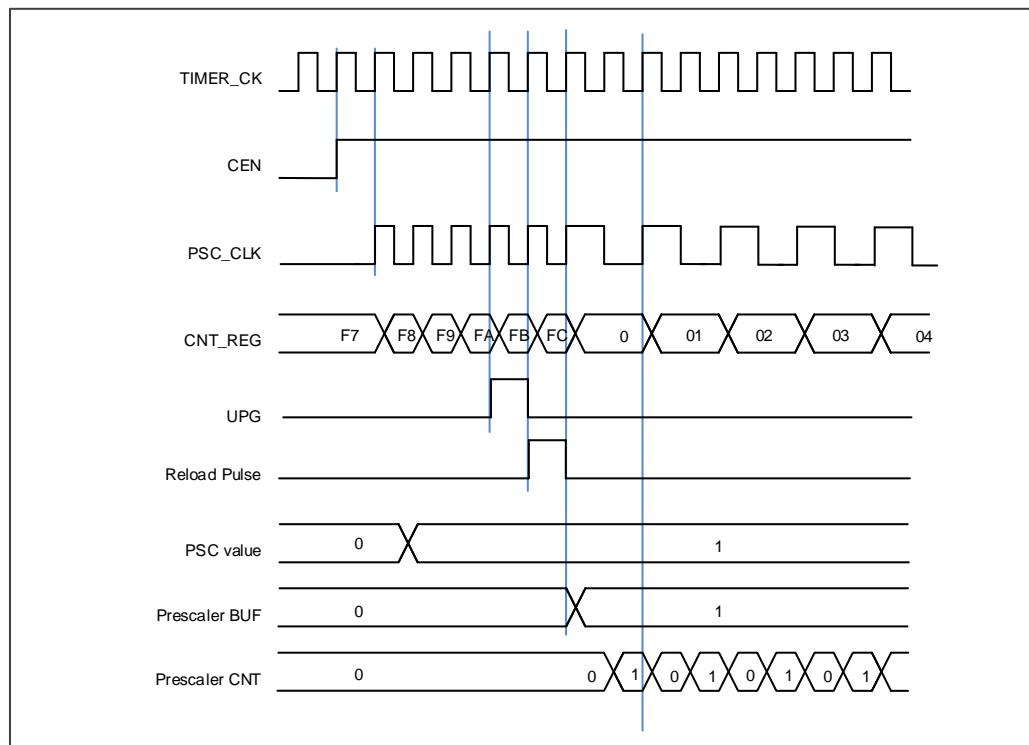
计数器预分频器可以在TIMERx_CH0/TIMERx_CH1引脚的每个上升沿或下降沿计数。这种模式可以通过设置SMC[2:0]为0x7同时设置TRGS[2:0]为0x4, 0x5或0x6来选择。Clx是TIMERx_Cl通过数字滤波器采样后的信号。

计数器预分频器也可以在内部触发信号ITIO/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 16-55. 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(重复计数寄存器，自动重载寄存器，预分频寄存器)都将被更新。

[图 16-56. 向上计数时序图，PSC=0/1](#) 和 [图 16-57. 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 16-56. 向上计数时序图, PSC=0/1

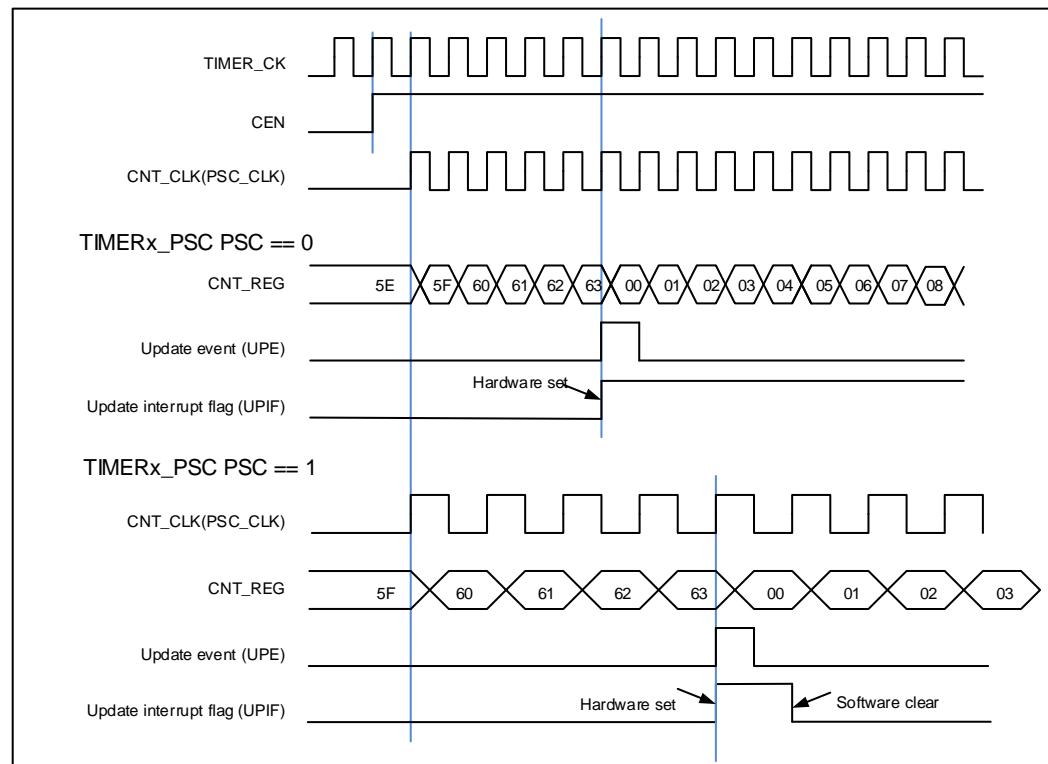
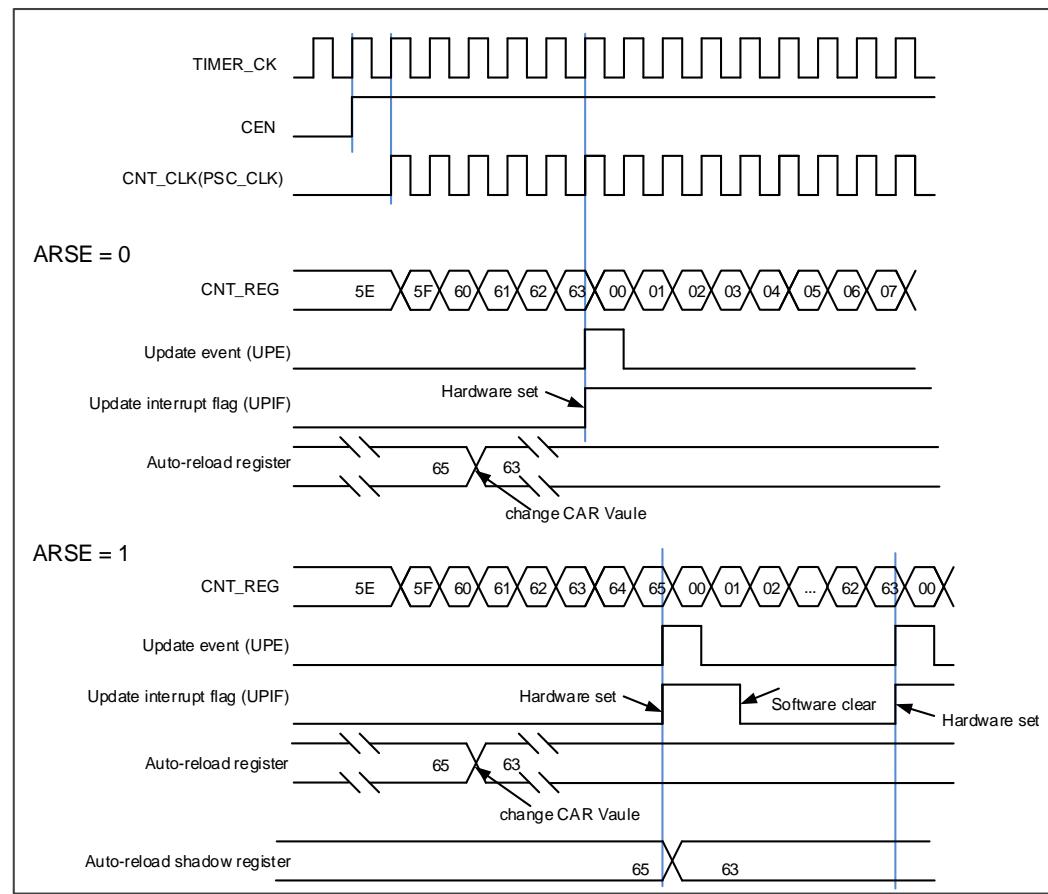


图 16-57. 向上计数时序图, 在运行时改变 TIMERx_CAR 寄存器的值



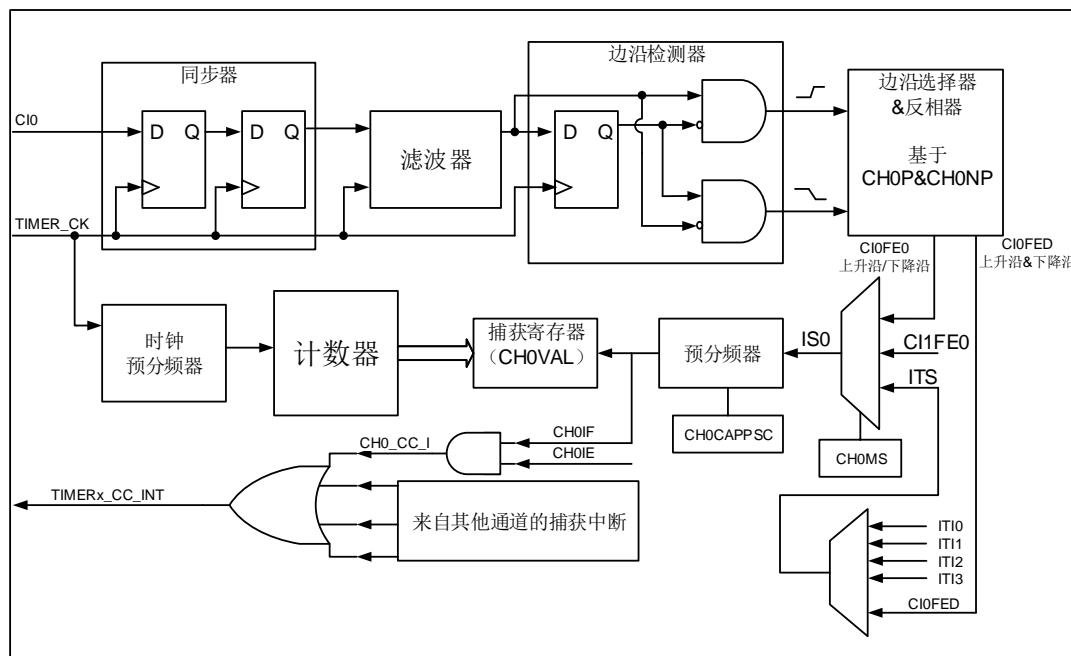
捕获/比较通道

通用定时器L1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

输入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 16-58. 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，**TIMERx_CHxCV**存储计数器的值。

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，下降沿或者双边沿（上升沿和下降沿）。

第三步：捕获源选择 (**TIMERx_CHCTL0**寄存器中**CHxMS**)：

一旦通过配置**CHxMS**选择输入捕获源，必须确保通道配置在输入模式(**CHxMS!=0x0**)，而且**TIMERx_CHxCV**寄存器不能再被写。

第四步：中断使能 (**TIMERx_DMAINTEN**寄存器中**CHxIE** 和 **CHxDEN**)：

使能相应中断，可以获得中断和DMA请求。

第五步：捕获使能（**TIMERx_CHCTL2**寄存器中**CHxEN**）。

结果：当期望的输入信号发生时，**TIMERx_CHxCV**被设置成当前计数器的值，**CHxIF**为置1。

如果**CHxIF**位已经为1，则**CHxOF**位置1。根据**TIMERx_DMINTEN**寄存器中**CHxIE**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C10。配置**TIMERx_CHCTL0**寄存器中**CH0MS**为2'b01，选择通道0的捕获信号为C10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C10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

图 16-59. 输出比较逻辑 (x=0,1)

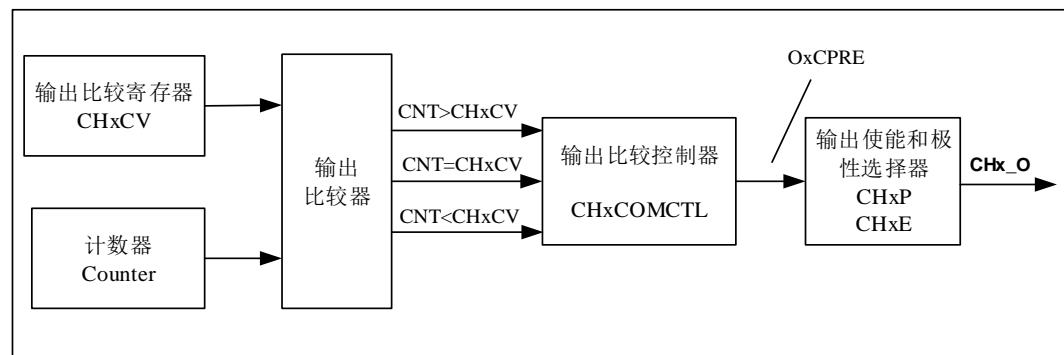


图16-59. 输出比较逻辑(x=0,1)给出了输出比较的逻辑电路。通道输出信号**CHx_O**与**OxCPRE**信号（详情请见[通道输出准备信号](#)）的关系描述如下：**OxCPRE**信号高电平有效，**CHx_O**的输出情况与**OxCPRE**信号，**CHxP**位和**CHxE**位有关（具体情况请见**TIMERx_CHCTL2**寄存器中的描述）。例如，当设置**CHxP=0**（**CHx_O**高电平有效，与**OxCPRE**输出极性相同）、**CHxE=1**（**CHx_O**输出使能）时：

若**OxCPRE**输出有效（高）电平，则**CHx_O**输出有效（高）电平；

若**OxCPRE**输出无效（低）电平，则**CHx_O**输出无效（低）电平。

在输出比较模式，**TIMERx**可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的**TIMERx_CHxCV**寄存器与计数器的值匹配时，根据**CHxCOMCTL**的配置，这个通道的输出可以被置高电平，被置低电平或者反转。当计数器的值与**TIMERx_CHxCV**寄存器的值匹配时，**CHxIF**位被置1，如果**CHxIE = 1**则会产生中断，如果**CxDCE=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/CxDCE位配置中断/DMA请求使能。

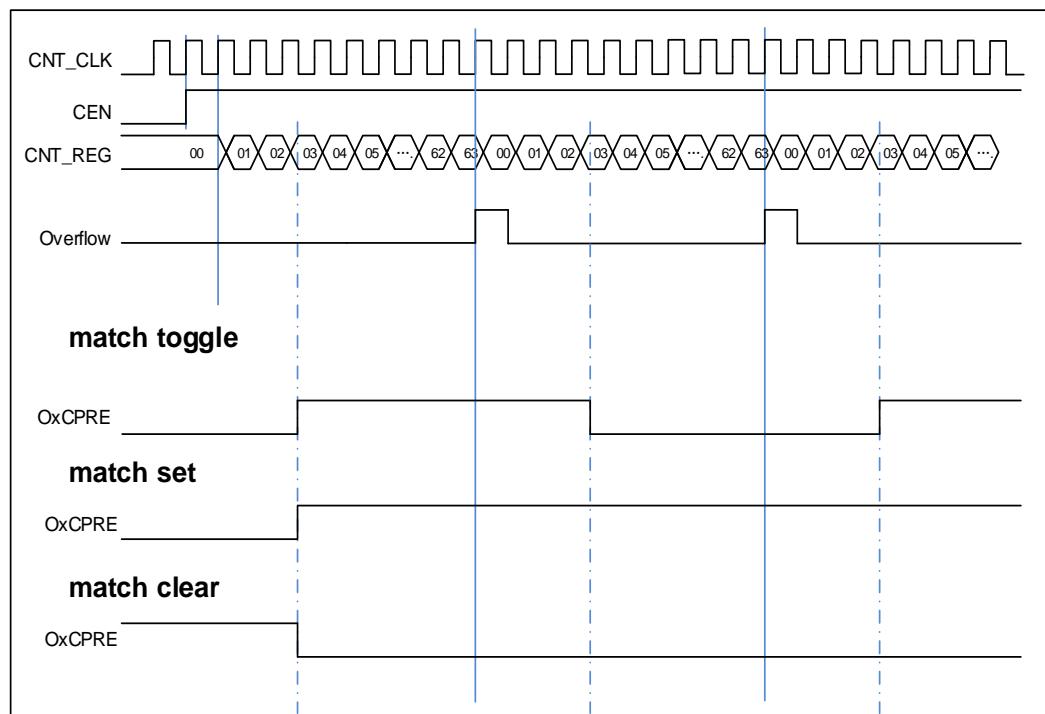
第四步: 通过TIMERx_CAR寄存器和TIMERx_CHxCV寄存器配置输出比较时基:

TIMERx_CHxCV可以在运行时根据你所期望的波形而改变。

第五步: 设置CEN位使能定时器。

图 16-60. 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 16-60. 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 [16-61. 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 [16-62. CAPWM 时序图](#) 显示了 CAPWM 的输出波形和中断。

在向上计数模式中, PWM 模式 0 下 (CHxCOMCTL=3'b110), 如果 TIMERx_CHxCV 寄存器的

值大于TIMERx_CAR寄存器的值，通道输出一直为无效电平；PWM模式1下（CHxCOMCTL=3'b111），如果TIMERx_CHxCV寄存器的值大于TIMERx_CAR寄存器的值，通道输出一直为有效电平。

图 16-61. EAPWM 时序图

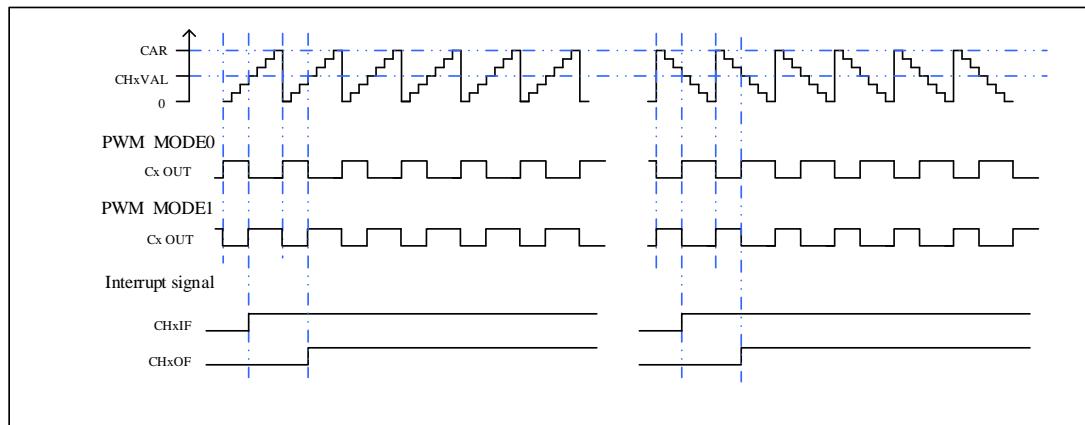
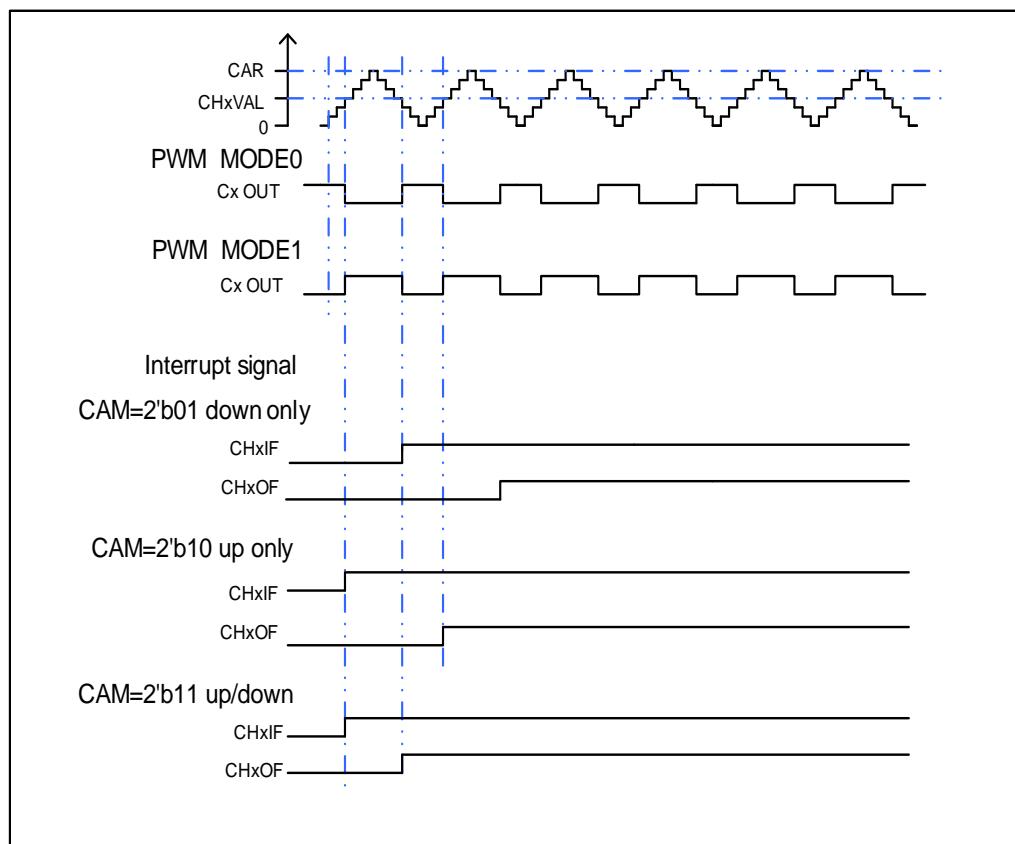


图 16-62. CAPWM 时序图



通道输出准备信号

根据[图16-59. 输出比较逻辑 \(x=0,1\)](#)所示，当TIMERx用于输出匹配比较模式下，在通道输出信号之前会产生一个中间信号OxCPRE信号(通道x输出准备信号)。设置CHxCOMCTL位可以定义OxCPRE信号类型。当TIMERx用于输出匹配比较模式下，设置CHxCOMCTL位可以定义

OxCPRE信号(通道x输出准备信号)类型。OxCPRE信号有若干类型的输出功能，包括，设置CHxCOMCTL=0x00可以保持原始电平；设置CHxCOMCTL=0x01可以将OxCPRE信号设置为高电平；设置CHxCOMCTL=0x02可以将OxCPRE信号设置为低电平；设置CHxCOMCTL=0x03，在计数器值和TIMERx_CHxCV寄存器的值匹配时，可以翻转输出信号。

PWM模式0和PWM模式1是OxCPRE的另一种输出类型，设置CHxCOMCTL位域为0x06或0x07可以配置PWM模式0/PWM模式1。在这些模式中，根据计数器值和TIMERx_CHxCV寄存器值的关系以及计数方向，OxCPRE信号改变其电平。具体细节描述，请参考相应的位。

设置CHxCOMCTL=0x04或0x05可以实现OxCPRE信号的强制输出功能。输出比较信号能够直接由软件强置为有效或无效状态，而不依赖于TIMERx_CHxCV的值和计数器值之间的比较结果。

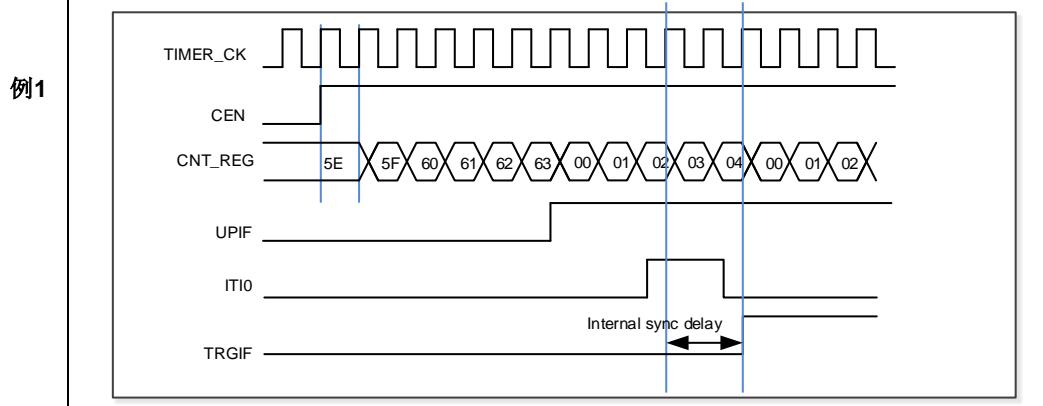
从控制器

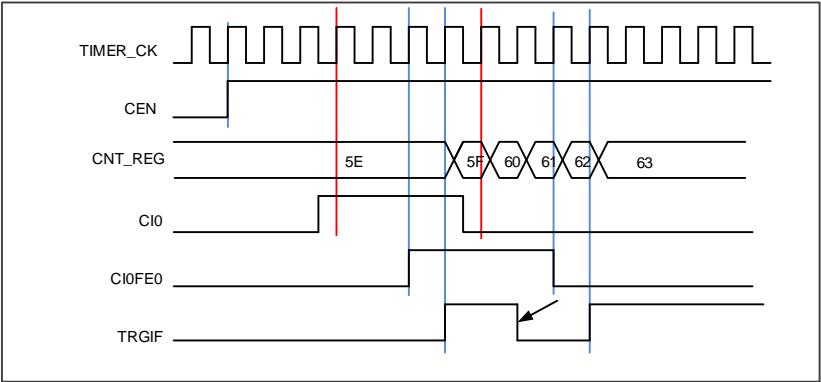
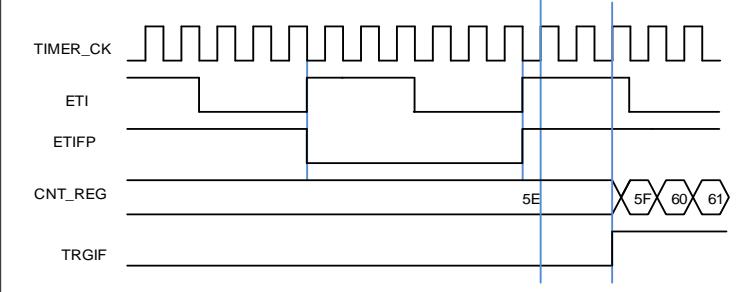
TIMERx能在多种模式下同步外部触发，包括复位模式，暂停模式和事件模式，可以通过设置TIMERx_SMCFG寄存器中的SMC[2:0]配置这些模式。这些模式的输入触发源可以通过设置TIMERx_SMCFG寄存器中的TRGS[2:0]来选择。

表 16-7. 从机模式示例（通用定时器 L1）

	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: ETIFP	如果触发源是CI0FE0或者CI1FE1，配置CHxP和CHxNP来选择极性和反相。 如果触发源是ETIFP，配置ETP选择极性和反相。	若触发源为ITIx，滤波和预分频不可用。 若触发源为CIx，可配置CHxCAPFLT设置滤波，预分频不可用。 若触发源为ETIFP，滤波和预分频均可用。
	复位模式 当触发输入上升沿到来时，计数器清零重启。	TRGS[2:0]=3'b000 选择ITI0为触发源。	若触发源是ITI0，极性选择不可用。	若触发源是ITI0，滤波和预分频不可用。

图 16-63. 复位模式



	模式选择	触发源选择	极性选择	滤波和预分频
	暂停模式 当触发输入为低的时候，计数器暂停计数，当触发输入为高时，计数器计数。	TRGS[2:0]=3'b101 选择CIOFE0为触发源。	TIOSS=0 (非异或) [CH0NP=0, CH0P=0] CIOFE0不反相。捕获发生在上升沿。	在这个例子中滤波被旁路。
图 16-64. 暂停模式				
例2				 <p>The timing diagram illustrates the operation of the timer in pause mode. The TIMER_CK signal is a square wave. The CEN signal is asserted (high) at the start of the first cycle. The CNT_REG signal shows the count value starting from 5E. The CI0 signal is asserted during the first cycle. The CIOFE0 signal is asserted during the second cycle. The TRGIF signal is asserted on the rising edge of CI0 during the first cycle.</p>
	事件模式 触发输入的上升沿计数器开始计数。	TRGS[2:0]=3'b111 选择ETIFP为触发源。	ETP = 0, ETI极性不变。 ETPSC = 1, ETI 2分频。 ETFC = 0, ETI 无滤波。	
图 16-65. 事件模式				
例3				 <p>The timing diagram illustrates the operation of the timer in event mode. The TIMER_CK signal is a square wave. The ETI signal is asserted during the first cycle. The ETIFP signal is asserted during the second cycle. The CNT_REG signal shows the count value starting from 5E. The TRGIF signal is asserted on the rising edge of ETIFP during the second cycle.</p>

单脉冲模式

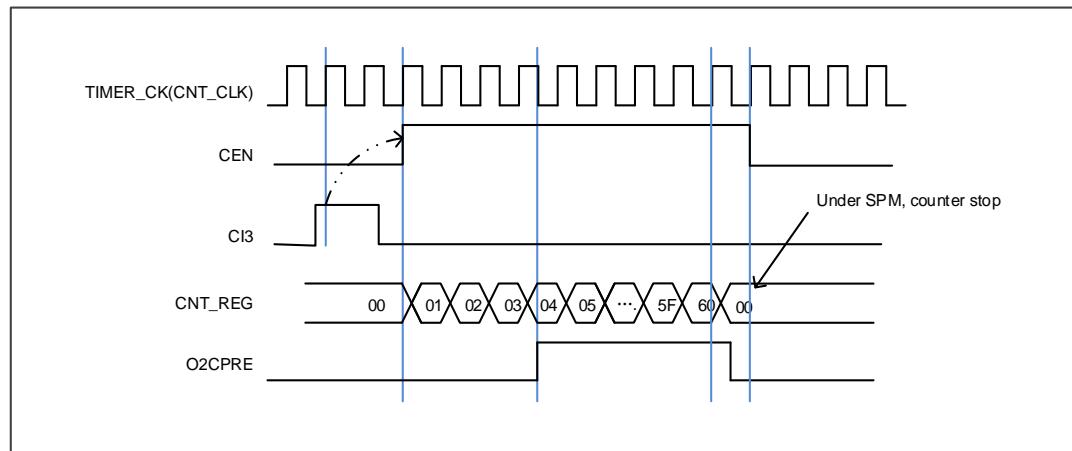
设置TIMERx_CTL0寄存器的SPM位置1，使能单脉冲模式。当SPM置1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置CHxCOMCTL配置TIMERx为PWM模式或者比较模式。

一旦设置定时器运行在单脉冲模式下，没有必要设置TIMERx_CTL0寄存器的定时器使能位CEN=1来使能计数器。触发信号沿或者软件写CEN=1都可以产生一个脉冲，此后CEN位一直保持为1直到更新事件发生或者CEN位被软件写0。如果CEN位被软件清0，计数器停止工作，计数值被保持。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，OxCMPRE信号将被立即强制转换为与发生比较匹配时相同的电平，但是不用考虑比较结果。只有输出通道配置为PWM模式0或PWM模式1时CHxCOMFEN位才可用，触发源来源于触发信号。

图 16-66. 单脉冲模式，TIMERx_CHxCV = 0x04，TIMERx_CAR=0x60



定时器互连

参考[高级定时器\(TIMERx, x=0, 7\)互连](#)

定时器调试模式

当Cortex™-M4内核停止，DBG_CTL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=8,11)

TIMER8基地址: 0x4001 4C00

TIMER11基地址: 0x4000 1800

控制寄存器 0 (TIMERx_CTL0)

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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				CKDIV[1:0]		ARSE		保留				SPM		UPS		UPDIS		CEN	
rw				rw		rw				rw		rw		rw		rw			

位/位域	名称	描述
31:10	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TIMER_CK(定时器时钟)与DTS(死区时间和采样时钟)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: $f_{DTS}=f_{TIMER_CK}$</p> <p>01: $f_{DTS}=f_{TIMER_CK} / 2$</p> <p>10: $f_{DTS}=f_{TIMER_CK} / 4$</p> <p>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</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: 单脉冲模式禁能，更新事件发生后，计数器继续计数。</p> <p>1: 单脉冲模式使能，在下一次更新事件发生时，CEN位硬件清零并且计数器停止计数。</p>
2	UPS	<p>更新请求源</p> <p>软件配置该位，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。

1: 使能后只有计数器溢出/下溢才产生更新中断或DMA请求。

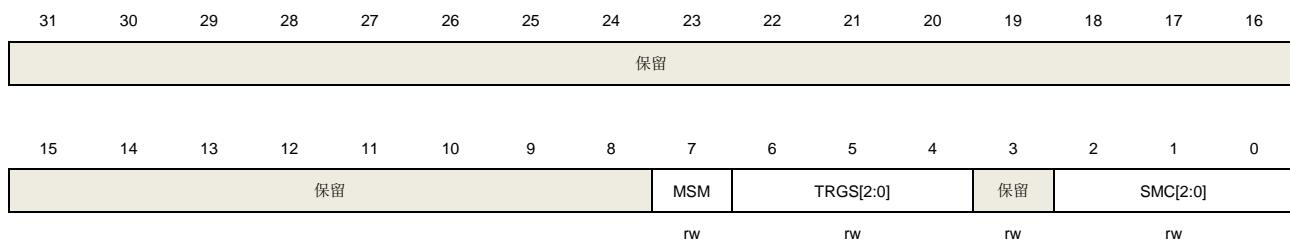
1	UPDIS	禁止更新 该位用来使能或禁能更新事件的产生。 0: 更新事件使能。当以下事件之一发生时，更新事件产生，具有缓存的寄存器被装入它们的预装载值： – UPG位被置1； – 计数器溢出/下溢； – 从模式控制器产生一个更新事件。 1: 更新事件禁能。带有缓存的寄存器保持原有值，如果UPG位被置1或者从模式控制器产生一个硬件复位事件，计数器和预分频器被重新初始化。
0	CEN	计数器使能 0: 计数器禁能 1: 计数器使能 在软件将CEN位置1后，外部时钟、暂停模式和编码器模式才能工作。事件模式可以自动地通过硬件将CEN位置位。

从模式配置寄存器 (**TIMERx_SMCFG**)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:8	保留	必须保持复位值
7	MSM	主-从模式 该位用来同步被选择的定时器同时开始计数。TRGI用做启动事件，通过TRGO，定时器被连接在一起。 0: 主从模式禁能 1: 主从模式使能
6:4	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定时器的触发输入源 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: 内部触发输入3 (ITI3)

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: 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。在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 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								TRGIE	保留				CH1IE	CHOIE	UPIE
rw									rw				rw		

位/位域	名称	描述
31:7	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
5:3	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	CH0IE	通道0比较/捕获中断使能 0: 禁止通道0中断 1: 使能通道0中断
0	UPIE	更新中断使能 0: 禁止更新中断 1: 使能更新中断

中断标志寄存器 (**TIMERx_INTF**)

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:11	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
5:3	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

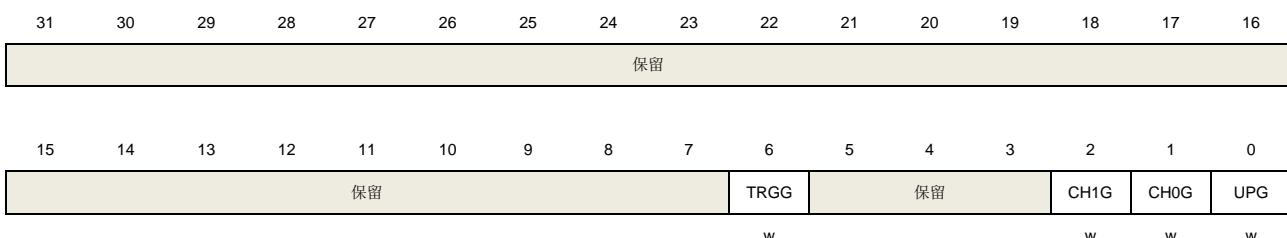
1	CH0IF	通道0比较/捕获中断标志 此标志由硬件置1软件清0。当通道0在输入模式下时，捕获事件发生时此标志位被置1；当通道0在输出模式下时，此标志位在一个比较事件发生时被置1。 0：无通道0中断发生 1：通道0中断发生
0	UPIF	更新中断标志 此位在更新事件发生时由硬件置1，软件清0。 0：无更新中断发生 1：发生更新中断

软件事件产生寄存器 (**TIMERx_SWEVG**)

地址偏移: 0x14

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:7	保留	必须保持复位值
6	TRGG	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
5:3	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，计数器的当前值被捕获到 TIMERx_CH0CV 寄存器，如果CH0IF标志位已经为1，则CH0OF标志位被置1。 0：不产生通道0捕获或比较事件 1：发生通道0捕获或比较事件

0	UPG	更新事件产生
		此位由软件置1，被硬件自动清0。当此位被置1，如果选择了中央对齐或向上计数模式，计数器被清0。否则(向下计数模式)计数器将载入自动重载值，预分频计数器将同时被清除。
	0:	无更新事件产生
	1:	产生更新事件

通道控制寄存器 0 (TIMERx_CHCTL0)

地址偏移: 0x18

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	CH1COMCTL[2:0]	CH1CO MSEN	CH1CO MFEN	CH1MS[1:0]	保留	CH0COMCTL[2:0]	CH0CO MSEN	CH0CO MFEN	CH0MS[1:0]						
CH1CAPFLT[3:0]	CH1CAPPSC[1:0]	CH0CAPFLT[3:0]	CH0CAPPSC[1:0]												
RW		RW		RW		RW		RW		RW		RW		RW	

输出比较模式:

位/位域	名称	描述
31:15	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。在向上计数时，一旦计数器值小于TIMERx_CH0CV时，O0CPRE为有效电平，否则为无效电平。在向下计数时，一旦计数器的值大于TIMERx_CH0CV时，O0CPRE为无效电平，否则为有效电平。		
111: PWM模式1。在向上计数时，一旦计数器值小于TIMERx_CH0CV时，O0CPRE为无效电平，否则为有效电平。在向下计数时，一旦计数器的值大于TIMERx_CH0CV时，O0CPRE为有效电平，否则为无效电平。		
在PWM模式0或PWM模式1中，只有当比较结果改变了或者输出比较模式中从冻结模式切换到PWM模式时，CxCOMR电平才改变。		
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		
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PWM模式0或者PWM模式1，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，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，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)

输入捕获模式:

位/位域	名称	描述
31:16	保留	必须保持复位值
15:12	CH1CAPFLT[3:0]	通道1输入捕获滤波控制 参见CH0CAPFLT描述
11:10	CH1CAPPSC[1:0]	通道1输入捕获预分频器 参见CH0CAPPSC描述
9:8	CH1MS[1:0]	通道1模式选择 与输出模式相同
7:4	CH0CAPFLT[3:0]	通道0输入捕获滤波控制 数字滤波器由一个事件计数器组成，N个输入事件后会产生一个输出的跳变。这些位定义了CI0输入信号的采样频率和数字滤波器的长度。 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 2 (TIMERx_CHCTL2)

地址偏移: 0x20

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								CH1NP	保留	CH1P	CH1EN	CH0NP	保留	CH0P	CH0EN

位/位域	名称	名称
31:8	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。
2	保留	必须保持复位值
1	CH0P	通道0极性 当通道0配置为输出模式时，此位定义了输出信号极性。 0: 通道0高电平有效 1: 通道0低电平有效 当通道0配置为输入模式时，此位定义了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CIxFE0的上升沿和下降沿都作为捕获或者从模式下触发的有效信号，并且CIxFE0不会被翻转。

0	CH0EN	通道0捕获/比较使能 当通道0配置为输出模式时，将此位置1使能CH0_O信号有效。当通道0配置为输入模式时，将此位置1使能通道0上的捕获事件。 0: 禁止通道0 1: 使能通道0
---	-------	--

计数器寄存器 (TIMERx_CNT)

地址偏移: 0x24

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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
CNT[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器 (TIMERx_PSC)

地址偏移: 0x28

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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
PSC[15:0]															

rw

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

入当前预分频寄存器。

计数器自动重载寄存器 (**TIMERx_CAR**)

地址偏移: 0x2C

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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
CARL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

通道 0 捕获/比较值寄存器 (**TIMERx_CH0CV**)

地址偏移: 0x34

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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
CH0VAL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH0VAL[15:0]	通道0的捕获或比较值 当通道0配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道0配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道1捕获/比较值寄存器 (**TIMERx_CH1CV**)

地址偏移: 0x38

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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
CH1VAL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH1VAL[15:0]	通道1的捕获或比较值 当通道1配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道1配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

配置寄存器 (**TIMERx_CFG**)

地址偏移: 0xFC

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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	描述
31:2	保留	必须保持复位值
1	CHVSEL	写捕获比较寄存器选择位 此位由软件写1或清0。 1: 当写入捕获比较寄存器的值与寄存器当前值相等时，写入操作无效 0: 无影响
0	保留	必须保持复位值

16.4. 通用定时器 L2 (TIMERx, x=9,10,12,13)

16.4.1. 简介

通用定时器L2 (TIMERx, x=9, 10, 12, 13)是单通道定时器，支持输入捕获和输出比较。可以产生PWM信号控制电机和电源管理。通用定时器L2含有一个16位无符号计数器。

通用定时器L2是可编程的，可以用来计数，其外部事件可以驱动其他定时器。

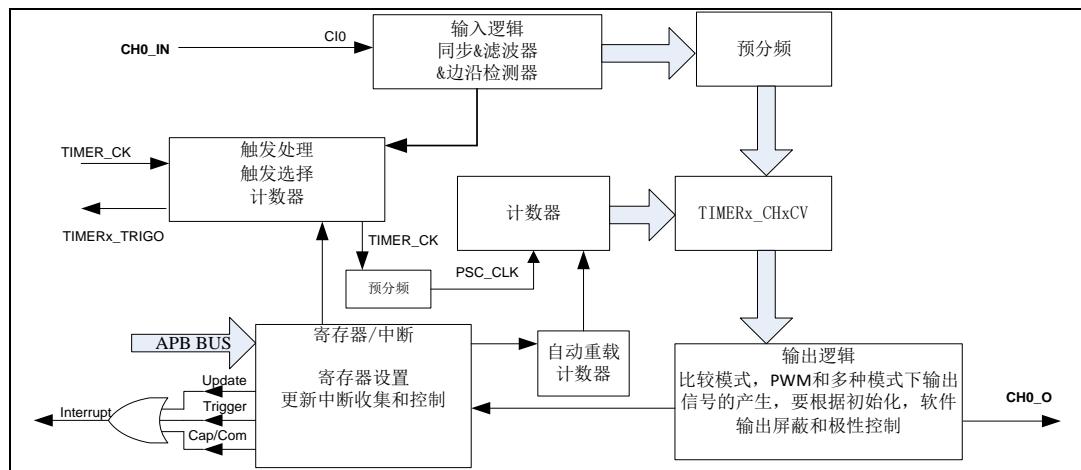
16.4.2. 主要特征

- 总通道数: 1;
- 计数器宽度: 16位;
- 时钟源: 内部时钟;
- 计数模式: 向上计数;
- 可编程的预分频器: 16位, 运行时可以被改变;
- 每个通道可配置: 输入捕获模式, 输出比较模式, 可编程的PWM模式;
- 自动重装载功能;
- 中断输出: 更新事件, 比较/捕获事件。

16.4.3. 结构框图

[图 16-67. 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 16-67. 通用定时器 L2 结构框图



16.4.4. 功能说明

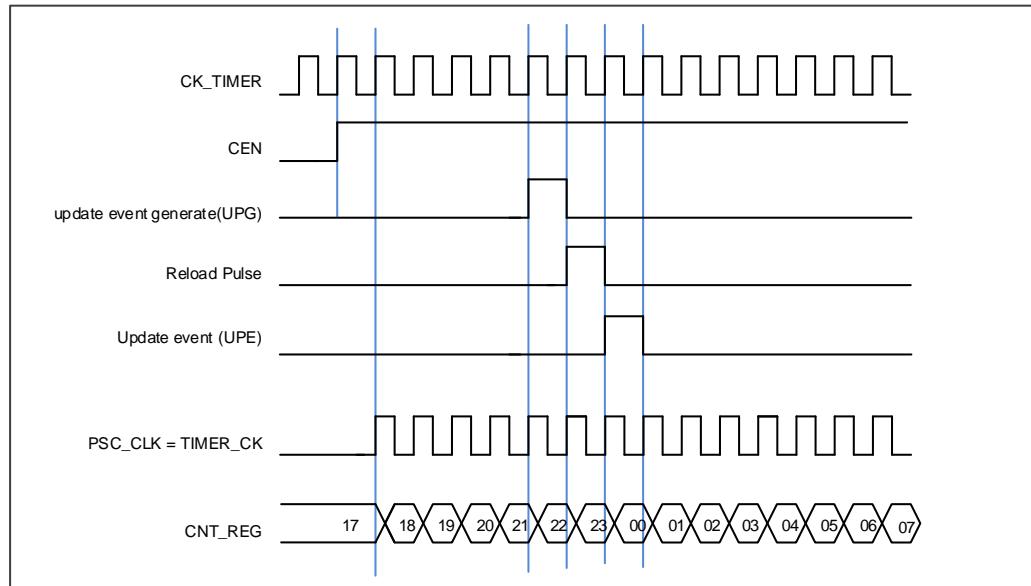
时钟源选择

通用定时器L2可以是内部时钟源TIMER_CK。

- 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。

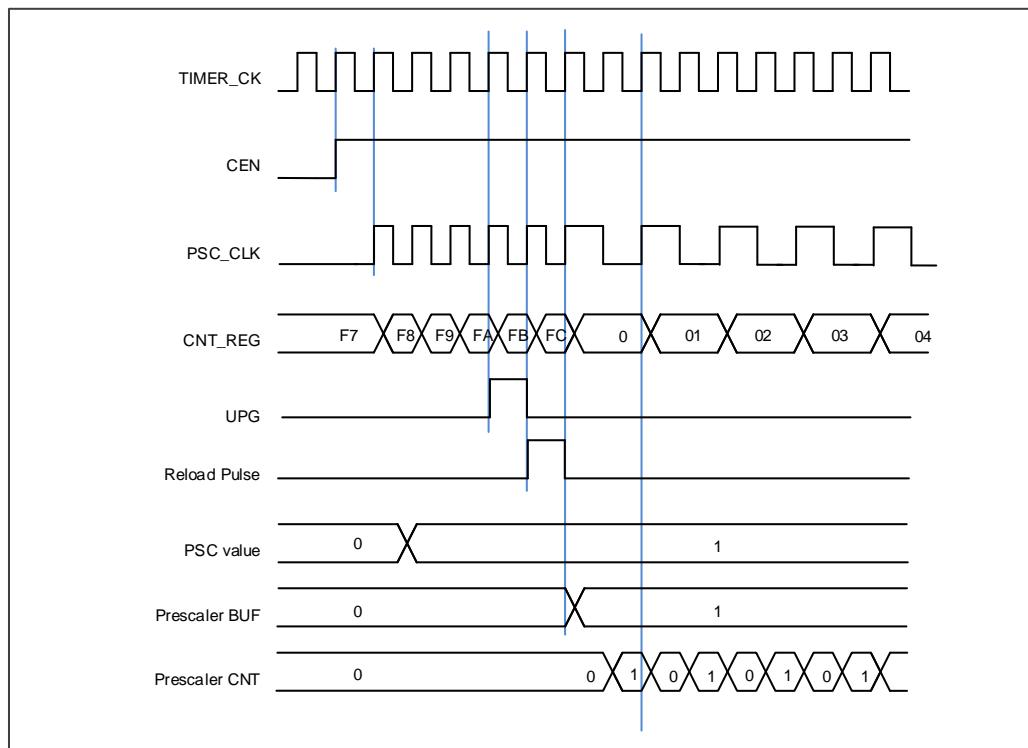
图 16-68. 内部时钟分频为 1 时正常模式下的控制电路



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 16-69. 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(重复计数寄存器，自动重载寄存器，预分频寄存器)都将被更新。

[图 16-70. 向上计数时序图, PSC=0/1](#) 和 [图 16-71. 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 16-70. 向上计数时序图, PSC=0/1

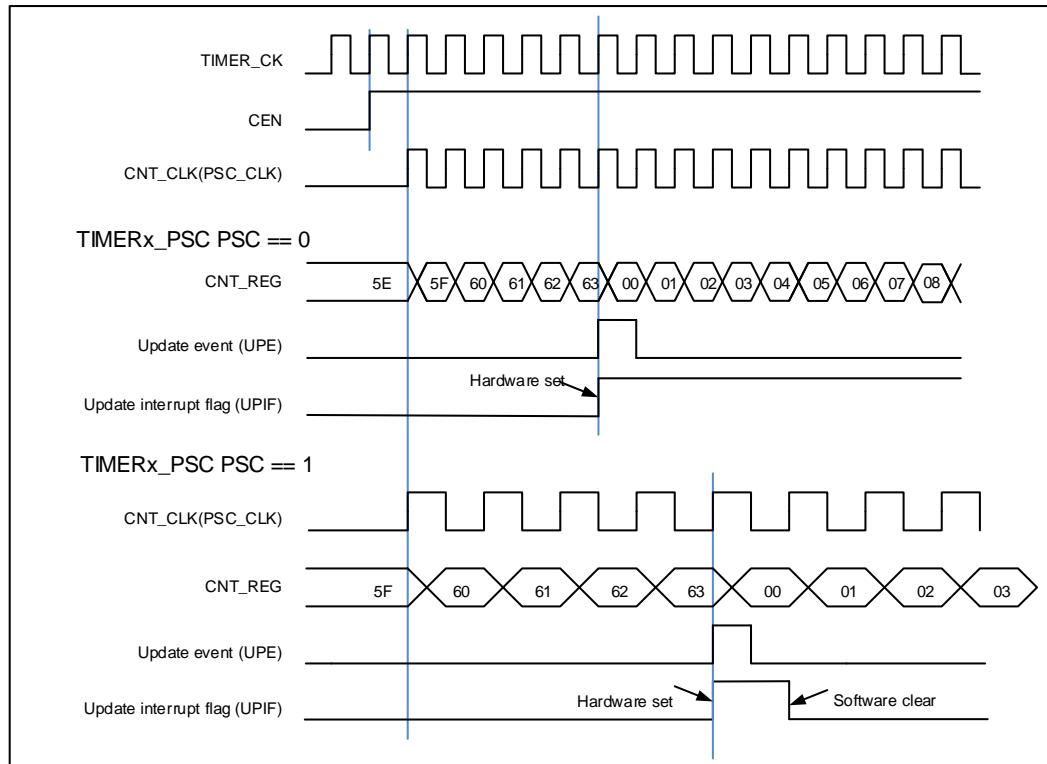
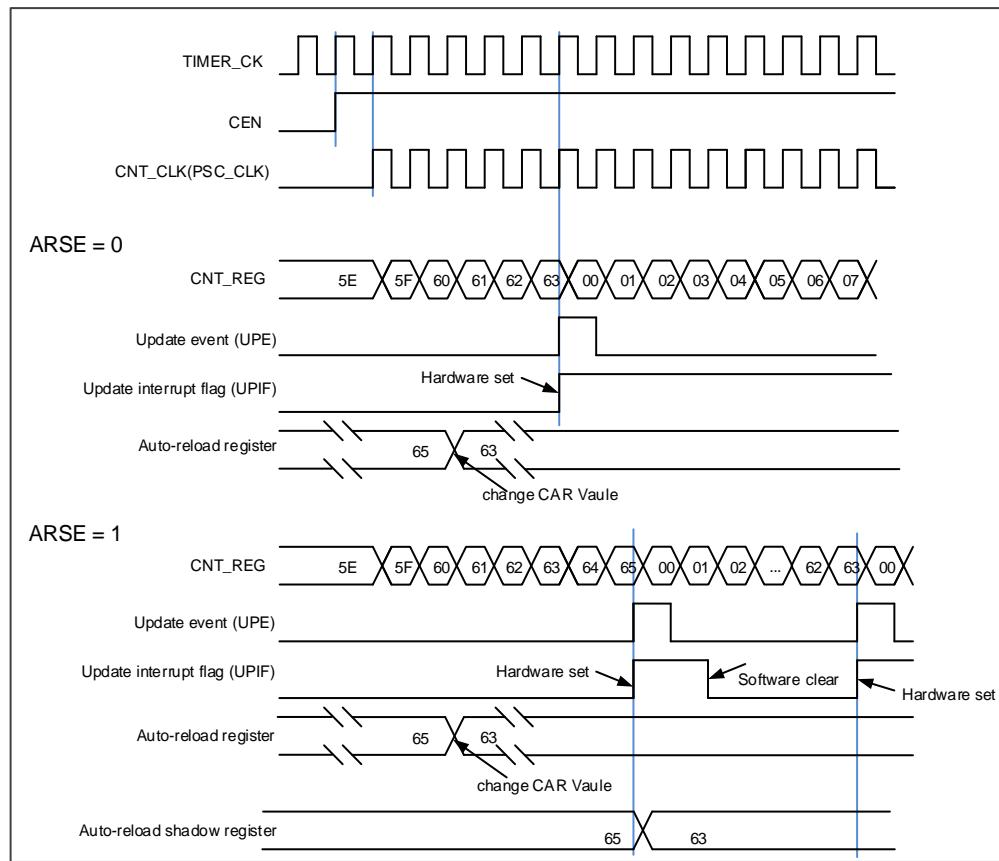


图 16-71. 向上计数时序图, 在运行时改变 TIMERx_CAR 寄存器的值



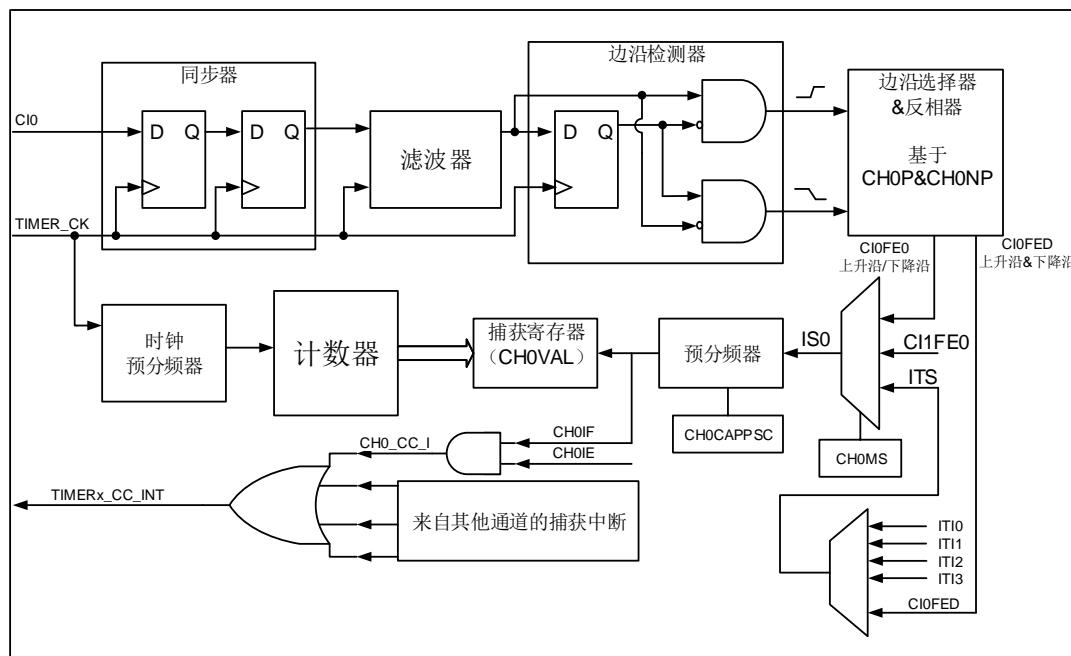
捕获/比较通道

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

输入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 16-72. 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，**TIMERx_CHxCV**存储计数器的值。

配置步骤如下：

第一步：滤波器配置 (TIMERx_CHCTL0寄存器中CHxCAPFLT):

根据输入信号和请求信号的质量，配置相应的**CHxCAPFLT**。

第二步：边沿选择 (TIMERx_CHCTL2寄存器中CHxP/CHxNP):

配置**CHxP/CHxNP**选择上升沿，下降沿或者双边沿（上升沿和下降沿）。

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

图 16-73. 输出比较逻辑

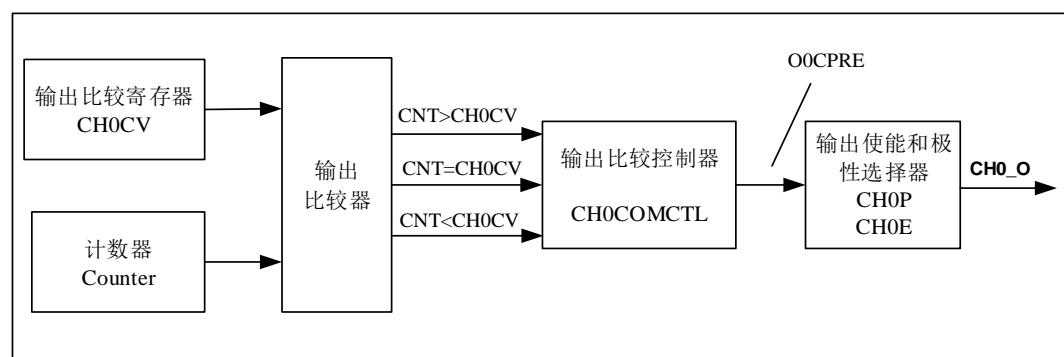


图16-73. 输出比较逻辑给出了输出比较的逻辑电路。通道输出信号CHx_O与OxCPRE信号(详情请见[通道输出准备信号](#))的关系描述：OxCPRE信号高电平有效，CHx_O的输出情况与OxCPRE信号，CHxP位和CHxE位有关(具体情况请见TIMERx_CHCTL2寄存器中的描述)。例如，当设置CHxP=0 (CHx_O高电平有效，与OxCPRE输出极性相同)、CHxE=1 (CHx_O输出使能)时：

若OxCPRE输出有效(高)电平，则CHx_O输出有效(高)电平；

若OxCPRE输出无效(低)电平，则CHx_O输出无效(低)电平。

在输出比较模式，TIMERx可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的TIMERx_CHxCV寄存器与计数器的值匹配时，根据CHxCOMCTL的配置，这个通道的输出可以被置高电平，被置低电平或者反转。当计数器的值与TIMERx_CHxCV寄存器的值匹配时，CHxIF位被置1，如果CHxIE = 1则会产生中断，如果CxODE=1则会产生DMA请求。

配置步骤如下：

第一步：时钟配置：

配置定时器时钟源，预分频器等。

第二步：比较模式配置：

设置CHxCOMSEN位来配置输出比较影子寄存器；

设置CHxCOMCTL位来配置输出模式（置高电平/置低电平/反转）；

设置CHxP/CHxNP位来选择有效电平的极性；

设置CHxEN使能输出。

第三步：通过CHxIE位配置中断使能。

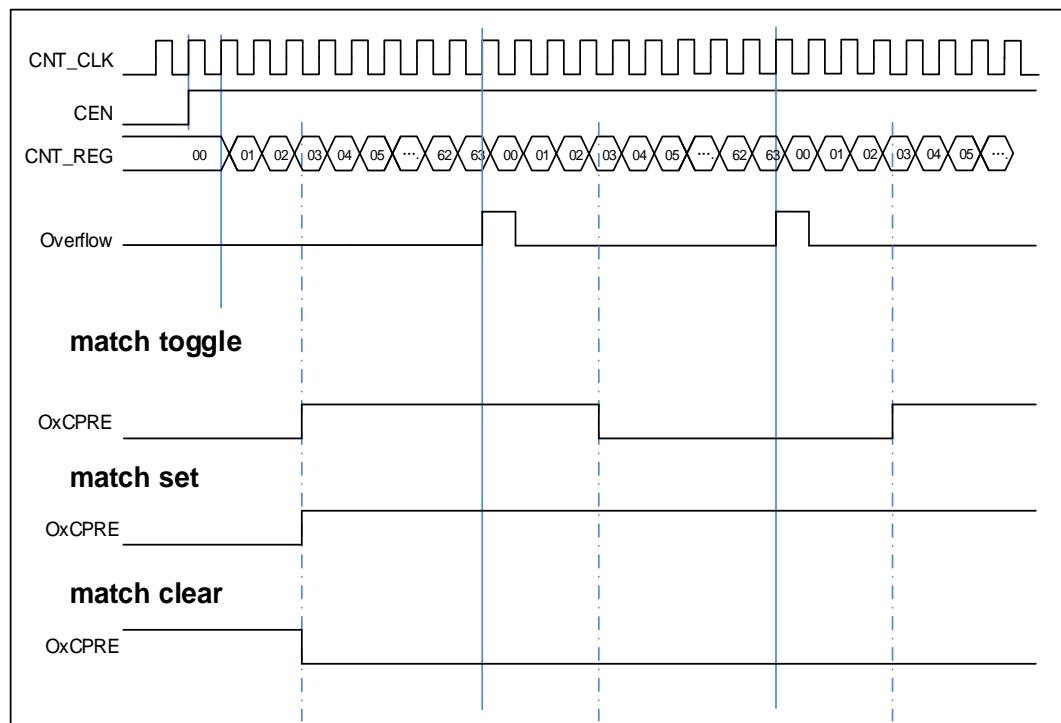
第四步：通过TIMERx_CAR寄存器和TIMERx_CHxCV寄存器配置输出比较时基：

TIMERx_CHxCV可以在运行时根据你所期望的波形而改变。

第五步：设置CEN位使能定时器。

图 16-74. 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 16-74. 三种输出比较模式



通道输出准备信号

根据**图16-73. 输出比较逻辑**所示，当TIMERx用于输出匹配比较模式下，在通道输出信号之前会产生一个中间信号OxCPRE信号(通道x输出准备信号)。设置CHxCOMCTL位可以定义OxCPRE信号类型。当TIMERx用于输出匹配比较模式下，设置CHxCOMCTL位可以定义OxCPRE信号(通道x输出准备信号)类型。OxCPRE信号有若干类型的输出功能，包括，设置CHxCOMCTL=0x00可以保持原始电平；设置CHxCOMCTL=0x01可以将OxCPRE信号设置为高电平；设置CHxCOMCTL=0x02可以将OxCPRE信号设置为低电平；设置CHxCOMCTL=0x03，在计数器值和TIMERx_CHxCV寄存器的值匹配时，可以翻转输出信号。

PWM模式0和PWM模式1是OxCPRE的另一种输出类型，设置CHxCOMCTL位域为0x06或0x07可以配置PWM模式0/PWM模式1。在这些模式中，根据计数器值和TIMERx_CHxCV寄存器值的关系以及计数方向，OxCPRE信号改变其电平。具体细节描述，请参考相应的位。

设置CHxCOMCTL=0x04或0x05可以实现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TIMER_x_CHxCV的值和计数器值之间的比较结果。

定时器调试模式

当Cortex™-M4内核停止，DBG_CTL寄存器中的TIMER_x_HOLD配置位被置1，定时器计数器停止。

16.4.5. TIMERx 寄存器(x=9,10,12,13)

TIMER9地址: 0x4001 5000

TIMER10地址: 0x4001 5400

TIMER12地址: 0x4000 1C00

TIMER13地址: 0x4000 2000

控制寄存器 0 (TIMERx_CTL0)

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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				CKDIV[1:0]		ARSE		保留				UPS		UPDIS	

位/位域	名称	描述
31:10	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TIMER_CK(定时器时钟)与DTS(死区时间和采样时钟)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: $f_{DTS}=f_{TIMER_CK}$</p> <p>01: $f_{DTS}=f_{TIMER_CK}/2$</p> <p>10: $f_{DTS}=f_{TIMER_CK}/4$</p> <p>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</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>软件配置该位，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	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。事件模式可以自动地通过硬件将CEN位置位。

控制寄存器 1 (TIMERx_CTL1)

地址偏移：0x04

复位值：0x0000 0000

该寄存器只能按字(32位)访问。

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

rw

位/位域	名称	描述
31:7	保留	必须保持复位值
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> <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。</p> <p>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。</p> <p>010: 更新。主模式控制器选择更新事件作为TRGO。</p> <p>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。</p> <p>100: 比较。在这种模式下，主模式控制器选择O0CPRE信号作为触发输出TRGO。</p> <p>101: 保留。</p>

110: 保留。

111: 保留。

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 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														CHOIE	UPIE

rW rW

位/位域	名称	描述
31:2	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 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					CHOOFF	保留					CHOIF	UPIF	rc_w0	rc_w0	rc_w0

位/位域	名称	描述
------	----	----

31:10	保留	必须保持复位值
9	CH0OF	<p>通道0捕获溢出标志</p> <p>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。</p> <p>0：无捕获溢出中断发生</p> <p>1：发生了捕获溢出中断</p>
8:2	保留	必须保持复位值
1	CH0IF	<p>通道0比较/捕获中断标志</p> <p>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。</p> <p>0：无通道0中断发生</p> <p>1：通道0中断发生</p>
0	UPIF	<p>更新中断标志</p> <p>此位在更新事件发生时由硬件置1，软件清0。</p> <p>0：无更新中断发生</p> <p>1：发生更新中断</p>

软件事件产生寄存器 (**TIMERx_SWEVG**)

地址偏移：0x14

复位值：0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												CH0G	UPG		

位/位域	名称	描述
31:2	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。</p> <p>此外，如果通道0配置为输入模式，计数器的当前值被捕获到TIMERx_CH0CV寄存器，如果CH0IF标志位已经为1，则CH0OF标志位被置1。</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	更新事件产生

此位由软件置1，被硬件自动清0。当此位被置1并且向上计数模式，计数器被清0，预分频计数器将同时被清除。

- 0: 无更新事件产生
- 1: 产生更新事件

通道控制寄存器 0 (TIMERx_CHCTL0)

地址偏移: 0x18

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
保留								保留	CH0COMCTL[2:0]	CH0COM SEN	CH0COM FEN	CH0MS[1:0]		CH0CAPFLT[3:0]		CH0CAPPSC[1:0]

rw

rw

rw

输出比较模式:

位/位域	名称	描述
31:7	保留	必须保持复位值
6:4	CH0COMCTL[2:0]	通道0输出比较模式 此位定义了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。在向上计数时，一旦计数器值小于TIMERx_CH0CV时，O0CPRE为有效电平，否则为无效电平。在向下计数时，一旦计数器的值大于TIMERx_CH0CV时，O0CPRE为无效电平，否则为有效电平。 111: PWM模式1。在向上计数时，一旦计数器值小于TIMERx_CH0CV时，O0CPRE为无效电平，否则为有效电平。在向下计数时，一旦计数器的值大于TIMERx_CH0CV时，O0CPRE为有效电平，否则为无效电平。 在PWM模式0或PWM模式1中，只有当比较结果改变了或者输出比较模式中从冻结模

式切换到PWM模式时，CxCOMR电平才改变。

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。
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PWM模式0或者PWM模式1，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配， 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 CI0FE1 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
31:8	保留	必须保持复位值
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，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 2 (TIMERx_CHCTL2)

地址偏移: 0x20

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留.										CH0NP	保留	CH0P	CH0EN	rw	

位/位域	名称	描述
31:4	保留	必须保持复位值
3	CH0NP	通道0互补输出极性 当通道0配置为输出模式，此位定义了互补输出信号的极性。 0: 通道0高电平有效 1: 通道0低电平有效 当通道0配置为输入模式时，此位和CH0P联合使用，作为输入信号CI0的极性选择控制信号。
2	保留	必须保持复位值
1	CH0P	通道0极性 当通道0配置为输出模式时，此位定义了输出信号极性。 0: 通道0高电平有效

1: 通道0低电平有效

当通道0配置为输入模式时，此位定义了CI0信号极性

[CH0NP, CH0P] 将选择CI0FE0或者CI1FE0的有效边沿或者捕获极性

[CH0NP=0, CH0P=0]: 把CIxFE0的上升沿作为捕获或者从模式下触发的有效信号，并且CIxFE0不会被翻转。

[CH0NP=0, CH0P=1]: 把CIxFE0的下降沿作为捕获或者从模式下触发的有效信号，并且CIxFE0会被翻转。

[CH0NP=1, CH0P=0]: 保留。

[CH0NP=1, CH0P=1]: 把CIxFE0的上升沿和下降沿都作为捕获或者从模式下触发的有效信号，并且CIxFE0不会被翻转。

0	CH0EN	通道0捕获/比较使能
		当通道0配置为输出模式时，将此位置1使能CH0_O信号有效。当通道0配置为输入模式时，将此位置1使能通道0上的捕获事件。
	0: 禁止通道0	
	1: 使能通道0	

计数器寄存器 (TIMERx_CNT)

地址偏移: 0x24

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15															
CNT[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器 (TIMERx_PSC)

地址偏移: 0x28

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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
PSC[15:0]															

rw

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

计数器自动重载寄存器 (TIMERx_CAR)

地址偏移: 0x2C

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

通道 0 捕获/比较值寄存器 (TIMERx_CH0CV)

地址偏移: 0x34

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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

rw

位/位域	名称	描述
------	----	----

31:16	保留	必须保持复位值
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>

配置寄存器 (**TIMERx_CFG**)

地址偏移: 0xFC

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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	描述
31:2	保留	必须保持复位值
1	CHVSEL	<p>写捕获比较寄存器选择位</p> <p>此位由软件写1或清0。</p> <p>1: 当写入捕获比较寄存器的值与寄存器当前值相等时，写入操作无效</p> <p>0: 无影响</p>
0	保留	必须保持复位值

16.5. 基本定时器 (TIMERx, x=5,6)

16.5.1. 简介

基本定时器(TIMER5/6)包含一个无符号16位计数器。基本定时器可以配置产生DMA请求，产生TRGO触发连接到DAC。

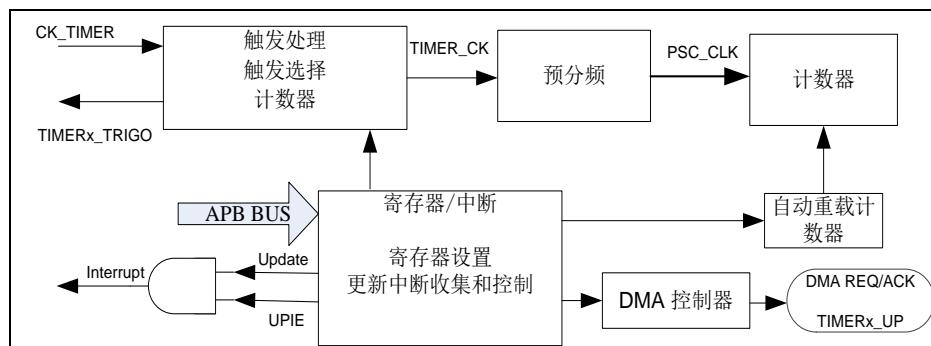
16.5.2. 主要特征

- 计数器宽度：16位；
- 定时器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。

16.5.3. 结构框图

[图 16-75. 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 16-75. 基本定时器结构框图



16.5.4. 功能说明

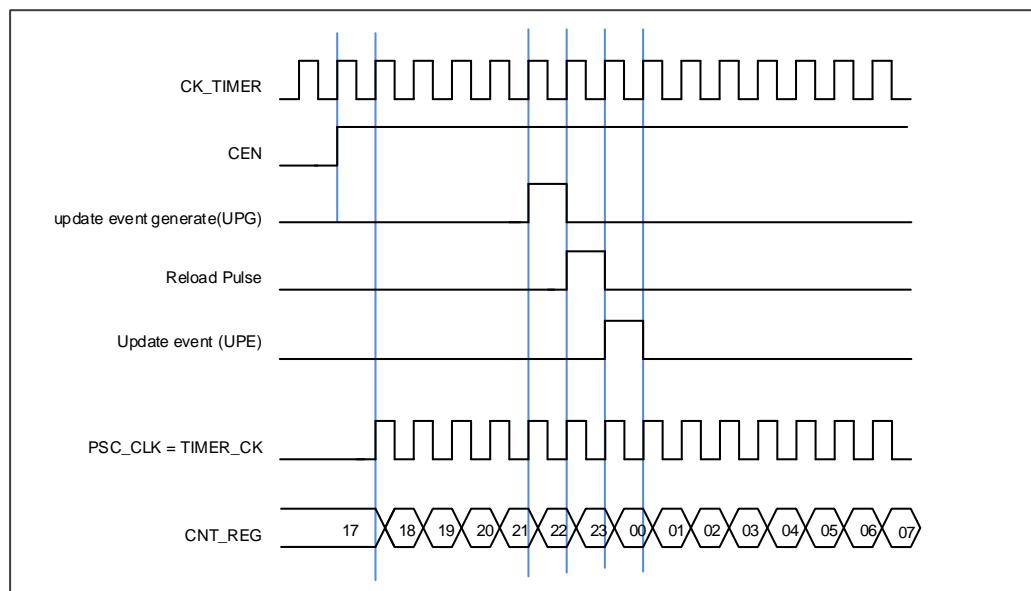
时钟源选择

基本定时器可以是内部时钟源CK_TIMER。

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。

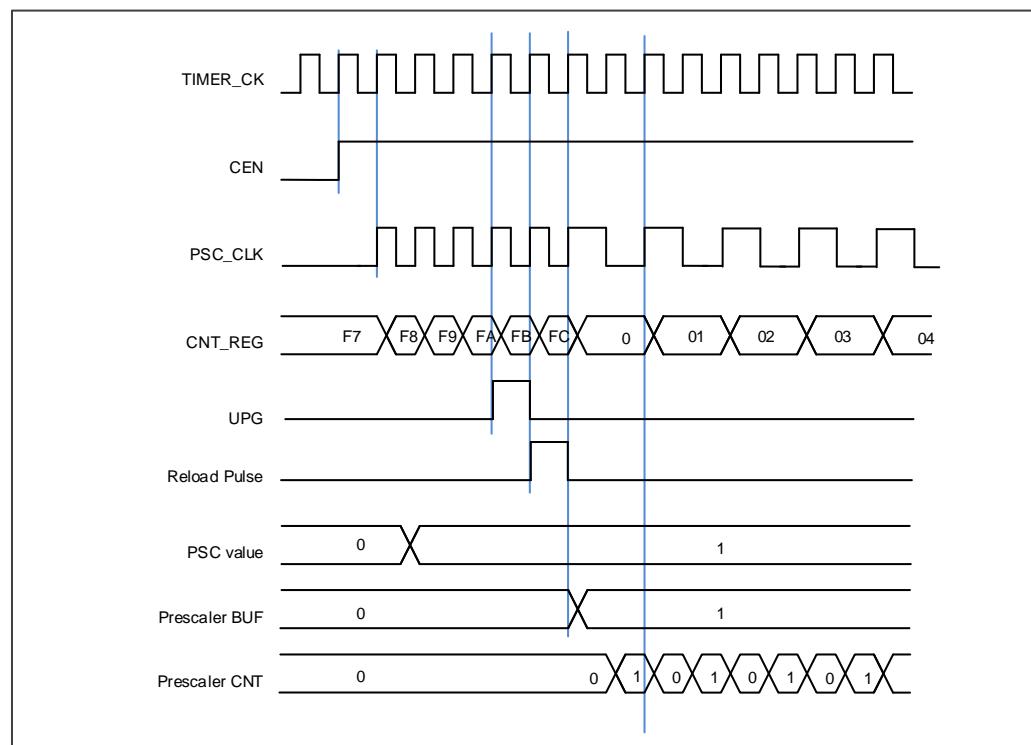
图 16-76. 内部时钟分频为 1 时正常模式下的控制电路



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 16-77. 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(重复计数寄存器，自动重载寄存器，预分频寄存器)都将被更新。

[图16-78. 向上计数时序图，PSC=0/1](#) 和 [图16-79. 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 16-78. 向上计数时序图，PSC=0/1

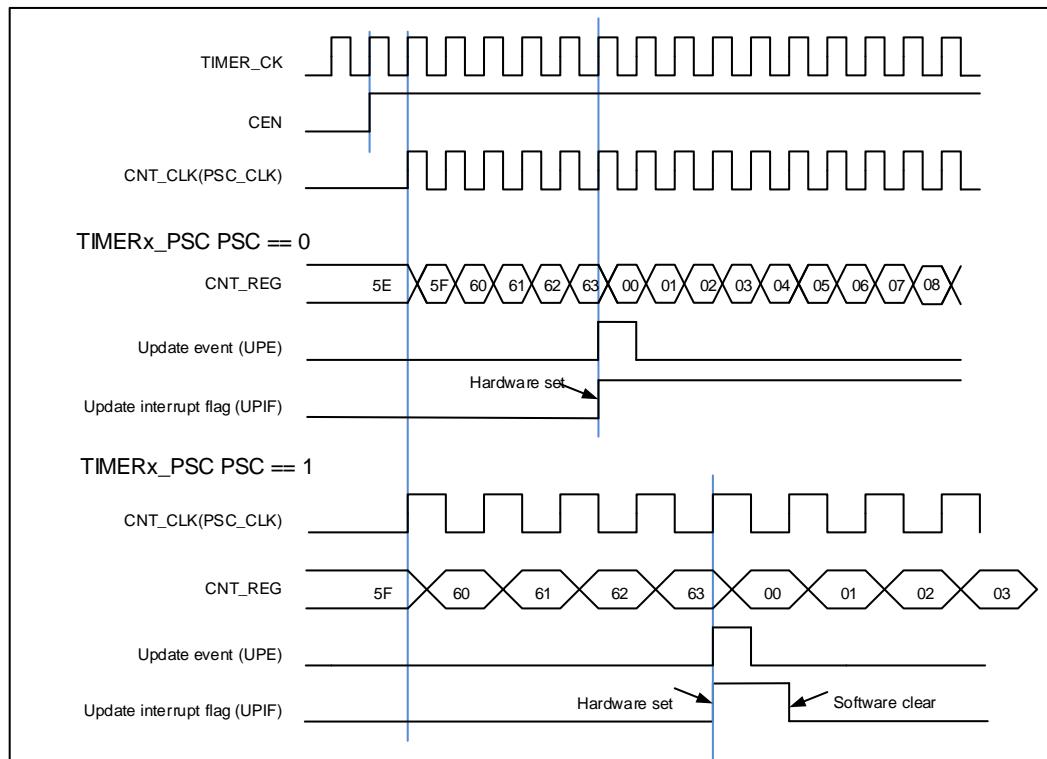
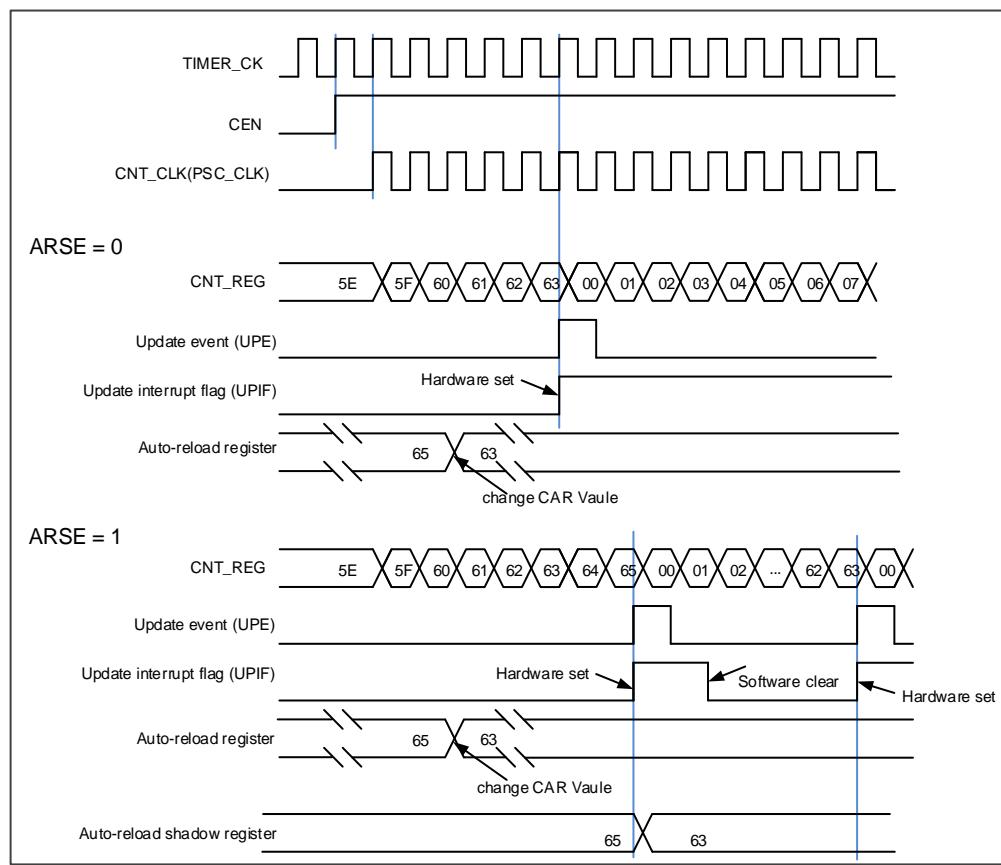


图 16-79. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值



定时器调试模式

当Cortex™-M4内核停止，DBG_CTL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=5,6)

TIMER5基地址: 0x4000 1000

TIMER6基地址: 0x4000 1400

控制寄存器 0 (TIMERx_CTL0)

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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						ARSE	保留			SPM	UPS	UPDIS	CEN		

位/位域	名称	描述
31:8	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	单脉冲模式 0: 单脉冲模式禁能, 更新事件发生后, 计数器继续计数 1: 单脉冲模式使能, 在下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 软件配置该位, 选择更新事件源. 0: 使能后, 下述任一事件产生更新中断或DMA请求: – UPG位被置1; – 计数器溢出/下溢; – 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

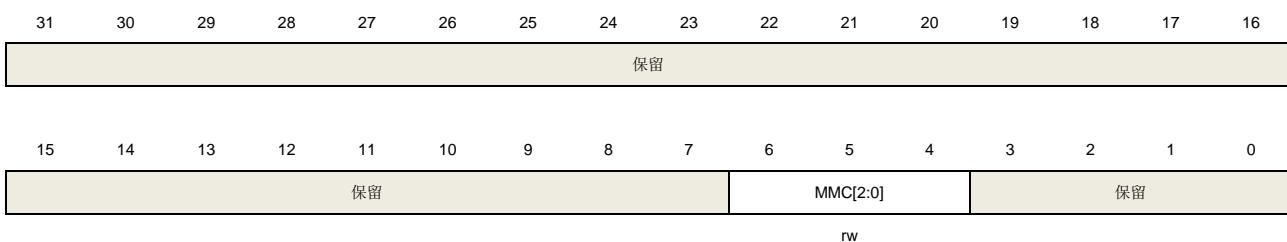
0	CEN	计数器使能
0:	计数器禁能	
1:	计数器使能	在软件将CEN位置1后，外部时钟、暂停模式和编码器模式才能工作。事件模式可以自动地通过硬件将CEN位置位。

控制寄存器 1 (TIMERx_CTL1)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。



位/位域	名称	描述
31:7	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 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				UPDEN		保留				UPIE					
rw								rw							

位/位域	名称	描述
31:9	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 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														UPIF	

rc_w0

位/位域	名称	描述
31:1	保留	必须保持复位值
0	UPIF	更新中断标志 此位在更新事件发生时由硬件置1，软件清0。 0: 无更新中断发生 1: 发生更新中断

软件事件产生寄存器 (TIMERx_SWEVG)

地址偏移: 0x14

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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

保留

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

保留

UPG

w

位/位域	名称	描述
31:1	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</p> <p>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 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

CNT[15:0]

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

预分频寄存器 (TIMERx_PSC)

地址偏移: 0x28

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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

PSC[15:0]

rw

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

计数器自动重载寄存器 (TIMERx_CAR)

地址偏移: 0x2C

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

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
CARL[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

17. 通用同步异步收发器 (USART)

17.1. 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。

USART支持DMA功能，以实现高速率的数据通信，除了UART4。

17.2. 主要特征

- NRZ标准格式。
- 全双工异步通信。
- 可编程的波特率产生器：
 - 由外设时钟分频产生，其中USART0由PCLK2分频得到，USART1/2和UART3/4由PCLK1分频得到；
 - 16倍过采样；
 - 当时钟频率为120M，过采样为16，最高速度可到7.5MBits/s。
- 完全可编程的串口特性：
 - 偶校验位，奇校验位，无校验位的生成/检测；
 - 数据位(8或9位)；
 - 产生0.5，1，1.5或者2个停止位。
- 发送器和接收器可分别使能。
- 支持硬件Modem流控操作(CTS/RTS)。
- DMA访问数据缓冲区。
- 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。
- 多种状态标志：

- 传输检测标志：接收缓冲区不为空(RBNE)，发送缓冲区为空(TBE)，传输完成(TC)，忙(BSY)；
- 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)；
- 智能卡模式标志：块结束(EBF)和接收超时(RTF)；
- 若相应的中断使能，这些事件发生将会触发中断。

USART0/1/2完全实现上述功能，但是UART3/4只实现了上面所介绍的部分功能，下面这些功能在UART3/4中没有实现：

- 智能卡模式；
- 同步模式；
- 硬件流操作(CTS/RTS)；
- 设置数据极性。

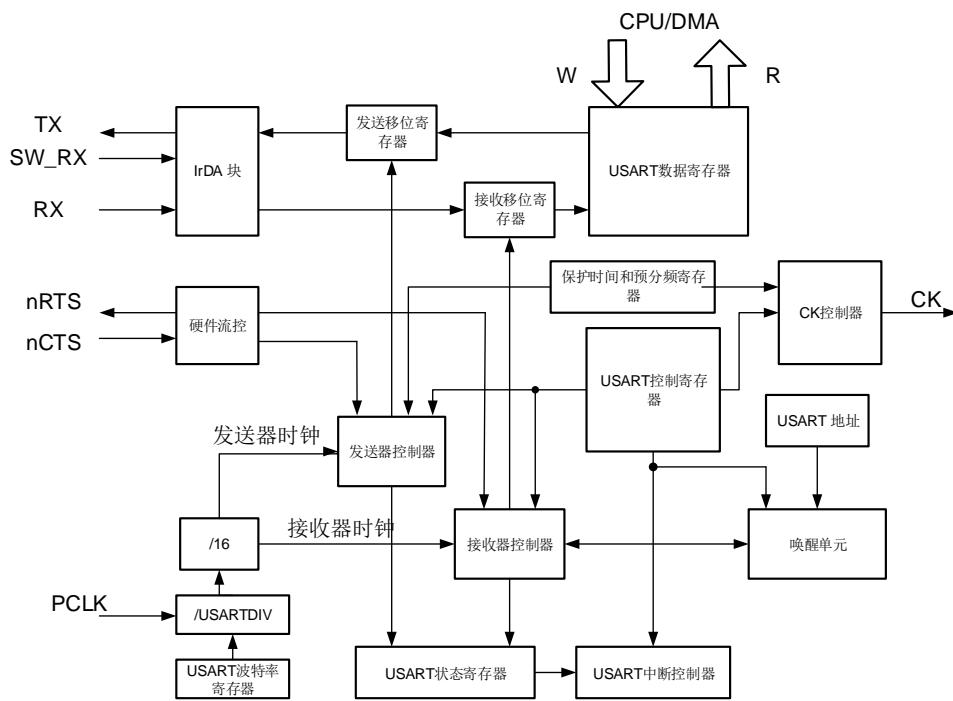
17.3. 功能说明

USART 接口通过[表 17-1. USART 重要引脚描述](#)中主要引脚从外部连接到其他设备。

表 17-1. USART 重要引脚描述

引脚	类型	描述
RX	输入	接收数据
TX	输出 I/O (单线模式/智能卡模式)	发送数据。当 USART 使能后，若无数据发送，默认为高电平
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

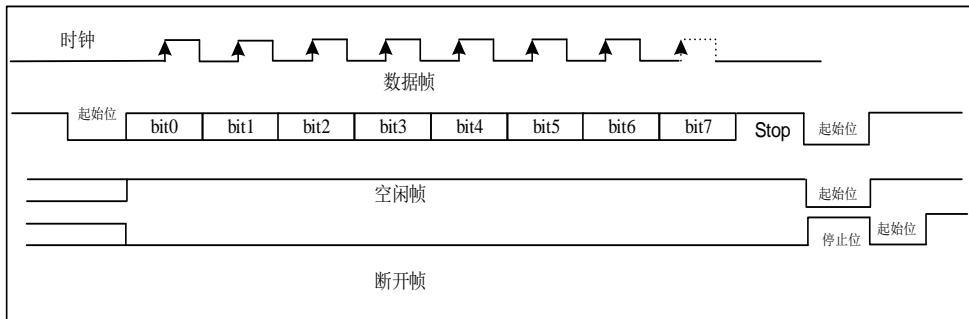
图 17-1. USART 模块内部框图



17.3.1. 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 17-2. USART 字符帧 (8 数据位和 1 停止位)



在发送和接收中，停止位可以由 USART_CTL1 寄存器中 STB[1:0] 位域配置。

表 17-2. 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，单线以及调制解调模式
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。

17.3.2. 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。

如果过采样率是16，波特率分频系数(USARTDIV)与系统时钟具有如下关系：

$$\text{USARTDIV} = \frac{\text{PCLK}}{16 \times \text{Baud Rate}} \quad (17-1)$$

USART0的系统时钟为PCLK2，USART1/2和UART3/4的系统时钟为PCLK1。在使能USART之前，必须在时钟控制单元使能系统时钟。

1. 由USART_BAUD寄存器的值得到USARTDIV:

假设USART_BAUD=0x21D，则INTDIV=33 (0x21)，FRADIV=13 (0xD)。
 $\text{UASRTDIV}=33+13/16=33.81$ 。

2. 由USARTDIV得到USART_BAUD寄存器的值:

假设要求UASRTDIV=30.37，INTDIV=30 (0x1E)
 $16*0.37=5.92$ ，接近整数6，所以FRADIV=6 (0x6)
 $\text{USART_BAUD}=0x1E6$ 。

注意：若取整后FRADIV=16（溢出），则进位必须加到整数部分。

17.3.3. 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。

TEN置位后发送器会发出一个空闲帧。TEN位在数据发送过程中是不可以被复位的。

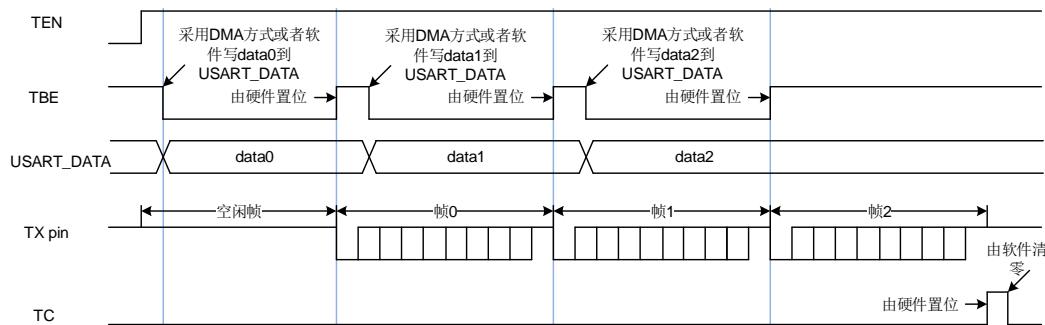
系统上电后，TBE默认为高电平。在USART_STAT0寄存器中TBE置位时，数据可以在不覆盖前一个数据的情况下写入USART_DATA寄存器。当数据写入USART_DATA寄存器，TBE位将被清0。在数据由USART_DATA移入移位寄存器后，该位由硬件置1。如果数据在一个发送过程正在进行时被写入USART_DATA寄存器，它将首先被存入发送缓冲区，在当前发送过程完成时传输到发送移位寄存器中。如果数据在写入USART_DATA寄存器时，没有发送过程正在进行，TBE位将被清零然后迅速置位，原因是数据将立刻传输到发送移位寄存器。

假如一帧数据已经发送出去，并且TBE位已经置位，那么USART_STAT0寄存器中TC位将被置1。如果USART_CTL0寄存器中的中断使能位(TCIE)为1，将会产生中断。

图 17-3. USART 发送步骤给出了USART发送步骤。软件操作按以下流程进行：

1. 在USART_CTL0寄存器中置位UEN位，使能USART；
2. 通过USART_CTL0寄存器的WL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(DENT位)；
5. 在USART_BAUD寄存器中设置波特率；
6. 在USART_CTL0寄存器中设置TEN位；
7. 等待TBE置位；
8. 向USART_DATA寄存器写数据；
9. 若DMA未使能，每发送一个字节都需重复步骤7-8；
10. 等待TC=1，发送完成。

图 17-3. 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_STAT0然后再写USART_DATA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。

17.3.4. USART 接收器

上电后，USART接收器使能按以下步骤进行：

1. 在USART_CTL0寄存器中置位UEN位，使能USART；
2. 写USART_CTL0寄存器的WL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便开始接收码流。在接收一个数据帧的过程中会检测噪声错误，奇偶校验错误，帧错误和过载错误。

当接收到一个数据帧，USART_STAT0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_STAT0寄存器中可以观察接收状态标志。

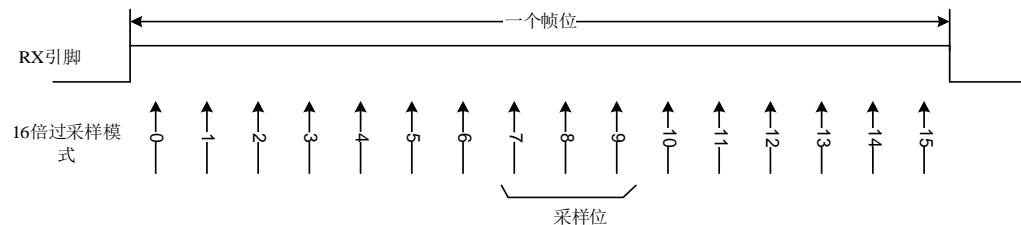
软件可以通过读USART_DATA寄存器或者DMA方式获取接收到的数据。不管是直接读寄存器还是通过DMA，只要是对USART_DATA寄存器的一个读操作都可以清除RBNE位。

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。

图 17-4. 过采样方式接收一个数据位



通过置位USART_CTL0寄存器中的PCEN位使能奇偶校验功能，接收器在接收一个数据帧时计算预期奇偶校验值，并将其与接收到的奇偶校验位进行比较。如果不相等，USART_STAT0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_STAT0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_STAT0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。

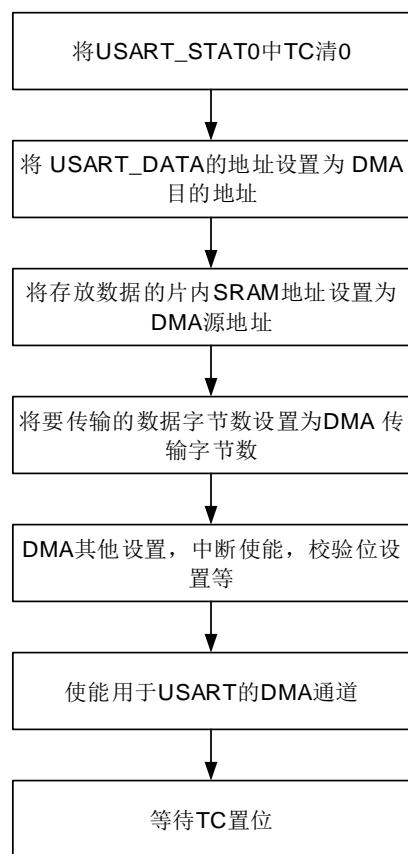
在一个接收过程中，RBNE、NERR、PERR、FERR和ORERR总是同时置位。如果没有使能DMA，软件需检查RBNE中断是否由NERR、PERR、FERR或者ORERR置位产生。

17.3.5. 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17-5. 采用DMA方式实现USART数据发送配置步骤](#)所示。

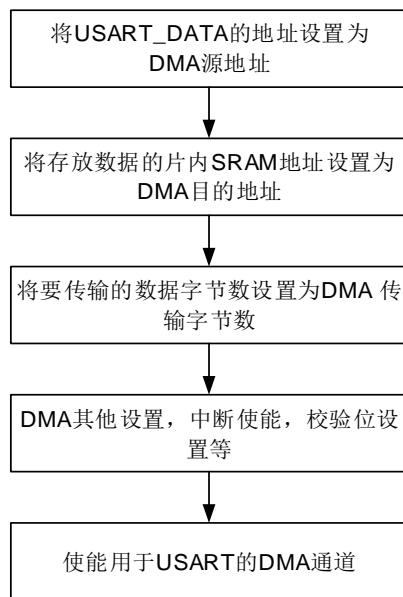
图 17-5. 采用 DMA 方式实现 USART 数据发送配置步骤



所有数据帧都传输完成后，USART_STAT0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17-6. 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_STAT0寄存器中的错误标志位(FERR、ORERR和NERR)被置位时将产生中断。

图 17-6. 采用 DMA 方式实现 USART 数据接收配置步骤

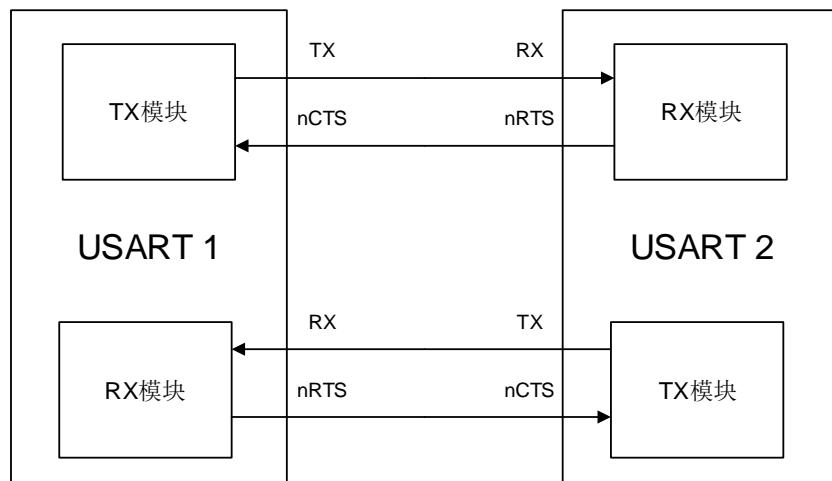


当USART接收到的数据数量达到了DMA传输数据数量，DMA模块将产生传输完成中断。

17.3.6. 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 17-7. 两个 USART 之间的硬件流控制



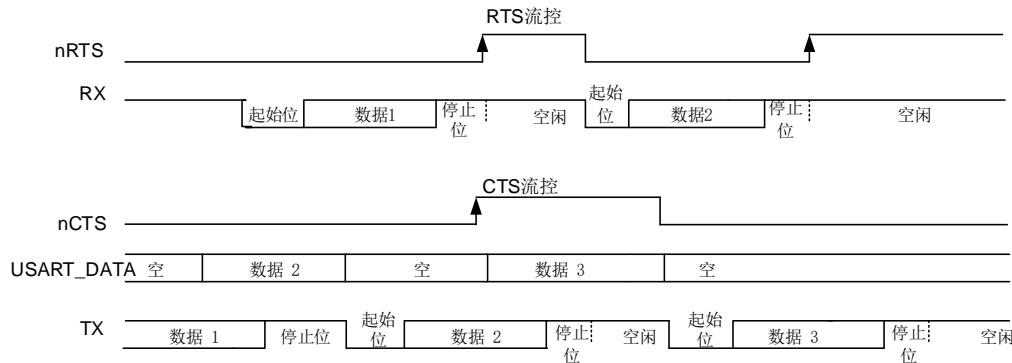
RTS 流控

USART接收器输出nRTS，它用于反映接收缓冲区状态。当一帧数据接收完成，nRTS变成高电平，这样是为了阻止发送器继续发送下一帧数据。当接收缓冲区满时，nRTS保持高电平，可以通过读USART_DATA寄存器来清零。

CTS 流控

USART发送器监视nCTS输入引脚来决定数据帧是否可以发送。如果USART_STAT0寄存器中TBE位是0且nCTS为低电平，发送器发送数据帧。在发送期间，若nCTS信号变为高电平，发送器将会在当前数据帧发送完成后停止发送。

图 17-8. 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_STAT0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。

17.3.7. 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_CTL0寄存器中RWU位置1使一个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。软件可以通过对RWU清零来唤醒USART。

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_STAT0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。如果地址字节的低4位与USART_CTL1寄存器中的ADDR[3:0]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_STAT0寄存器中获取。如果地址字节的低4位与USART_CTL1寄存器中的ADDR[3:0]不相同，硬件会置位RWU并进入静默模式。在这种情况下，RBNE不会被置位。

如果采用地址掩码检测，默认情况下，接收器对地址字节不做奇偶校验。如果USART_CTL0寄存器中PCEN位被置位。地址字节最高位被视为校验位，其余位被视为地址位。

17.3.8. 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，WL，STB[1:0]以及USART_CTL2的SCEN，

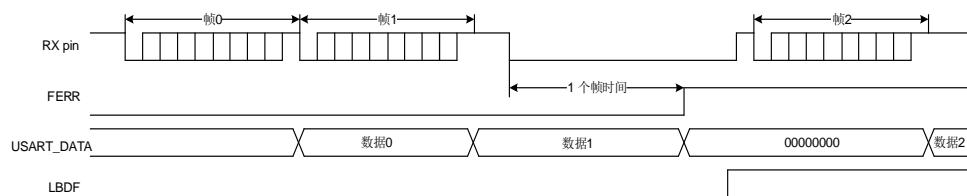
HDEN, IREN位都应该被清0。

在发送一个普通数据帧时, LIN发送过程与普通发送过程相同。当USART_CTL0寄存器中SBKCMD置位时, USART在发送完一个停止位后会连续发送13个0。

断开检测功能完全独立于普通USART接收器。因此, 断开检测可以是在空闲状态下, 也可以在数据传输过程中。USART_CTL1寄存器中LBLEN位可以选择断开帧长度。如果在RX引脚检测到大于或等于与预期断开帧长度相等数量的0(LBLEN=0时, 10个0; LBLEN=1时, 11个0), USART_STAT0寄存器中LBDF置位。如果USART_CTL1寄存器中LBDIE被置位, 将产生中断。

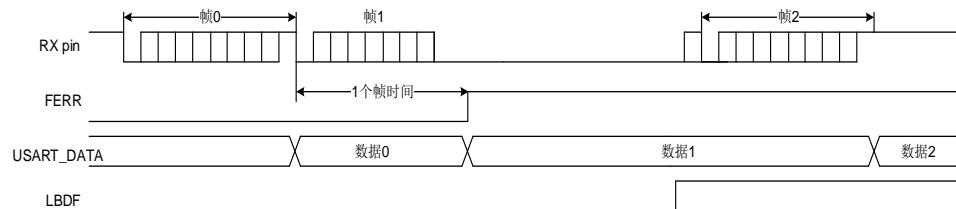
如[图17-9. 空闲状态下检测断开帧](#)所示, 如果断开帧发生在空闲状态下, USART接收器会接收到一个全0数据帧, 同时FERR置位。

图 17-9. 空闲状态下检测断开帧



如[图17-10. 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 17-10. 数据传输过程中检测断开帧



17.3.9. 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位应该被清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。USART_CTL1的CPH位用来决定数据在第一个时钟沿被采样还是在第二个时钟沿被采样。USART_CTL1的CPL位用来决定在USART同步模式空闲状态下, 时钟引脚的电平。

CK引脚输出波形由USART_CTL1寄存器中CPL, CPH, CLEN位决定。软件仅在USART禁用(UEN=0)时才可以改变它们的值。

如果USART_CTL0寄存器中REN置位, 接收器的工作方式与普通模式下接收方式是不同的。接收器在时钟捕获沿采样数据, 并无任何过采样。

图 17-11. 同步模式下的 USART 示例

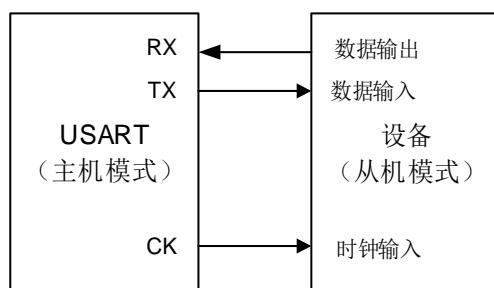
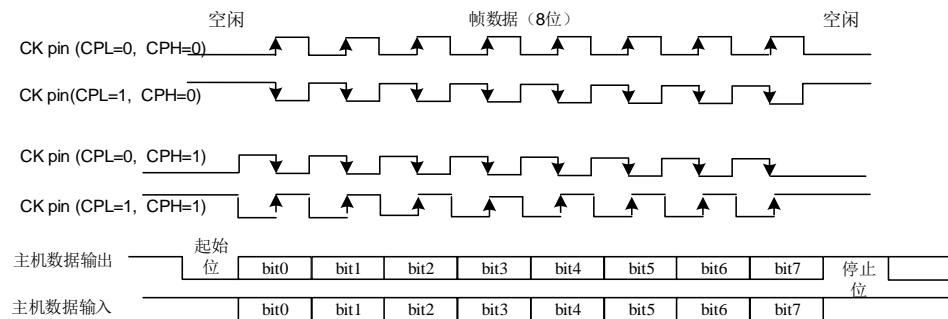


图 17-12. 8-bit 格式的 USART 同步通信波形(CLEN=1)

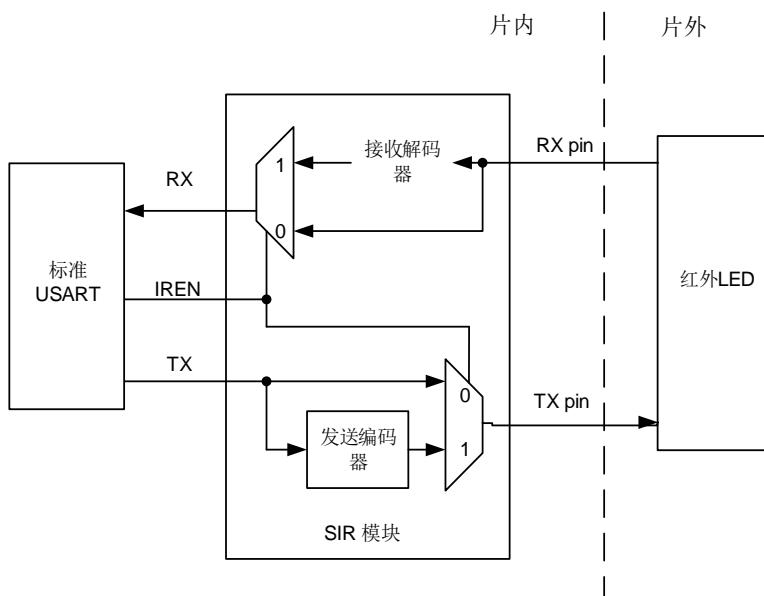


17.3.10. 串行红外(IrDA SIR)编解码功能模块

串行红外编解码功能通过置位 USART_CTL2 寄存器中 IREN 使能。在 IrDA 模式下，USART_CTL1 寄存器的 LMEN，STB[1:0]，CKEN 位和 USART_CTL2 寄存器的 HDEN，SCEN 位将被清 0。

在 IrDA 模式下，USART 数据帧由 SIR 发送编码器进行调制，调制后的信号经由红外 LED 进行发送，经解调后将数据发送至 USART 接收器。对于编码器而言，波特率应小于 115200。

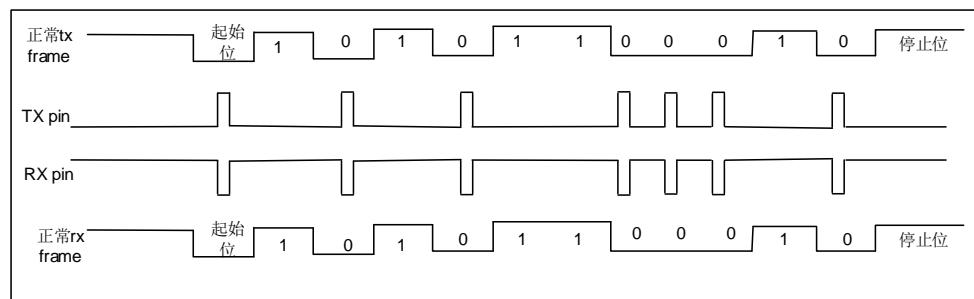
图 17-13. IrDA SIR ENDEC 模块



在IrDA模式下，TX引脚电平与RX引脚不同。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则无法可靠的检测到。

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 17-14. IrDA 数据调制



将USART_CTL2寄存器中IRLP置位可以使SIR子模块工作在低功耗模式下。发送编码器由PCLK分频得到的低速时钟来驱动。分频系数在USART_GP寄存器中PSC[7:0]位配置。TX引脚脉冲宽度可以为低功耗波特率的3倍。接收器解码器工作模式与正常IrDA模式相同。

17.3.11. 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，TX引脚和RX引脚将从内部连接到一起，RX引脚不再使用。TX引脚应该被配置为开漏输出模式。通信冲突由软件处理。

17.3.12. 智能卡(ISO7816-3)模式

智能卡模式是一种异步通信模式，支持ISO7816-3协议。支持字节模式($T=0$)和块模式($T=1$)。将USART_CTL2寄存器的SCEN位置1，即可使能智能卡模式。在智能卡模式下，USART_CTL1寄存器的LMEN位和USART_CTL2的HDEN，IREN位应该清0。

如果CKEN位被置位，USART通过CK引脚向智能卡提供一个由PCLK分频得到的时钟。分频系数可在USART_GP寄存器中PSC[4:0]配置。CK引脚只为智能卡提供时钟源。

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引脚需要被设置成开漏模式，外接上拉电阻，这个引脚将会与智能卡驱动同一条双向连线。智能卡模式下的帧格式为：1起始位+9数据位(包括1奇偶校验位)+1.5停止位。其中0.5个停止位被配置为接收器的停止位。

图 17-15. ISO7816-3 数据帧格式



字节模式(T=0)

相较于正常操作模式下的时序，从发送移位寄存器到TX引脚的传递时间延迟了半个波特率时钟，并且TC标志的置位将根据USART_GP寄存器的GUAT[7:0]设置延迟某一特定时间。在智能卡模式下，在最后一帧数据的停止位之后，内部保护时间计数器将开始计数，GUAT[7:0]的值配置为ISO7816-3协议的CGT减12。在保护时间寄存器向上计数这段时间TC将被强制拉低，当计数达到设定值时，TC被置位。

在USART发送期间，如果检测到有奇偶校验错误，TX引脚在停止位最后一个位时间内被拉低，智能卡发送一个NACK信号。根据协议，USART会自动重发SCRTNUM次。在重发数据帧前面会插入2.5位的帧间隔。最后一次重发字节后，TC会立即被置位。如果在最大重发次数后仍然收到NACK信号，USART将会停止发送，帧错误标志被置位。USART不会将NACK信号作为起始位。

在USART接收期间，如果在当前数据帧检测到校验错误，TX引脚在停止位的最后一个位时间内会被拉低。智能卡会接收到NACK信号。然后在智能卡端会产生一个帧错误。如果接收到的字节是错误的，RBNE中断和接收DMA请求都不会被激活。根据协议，智能卡将要重新发送数据。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信号和标注这个错误为奇偶校验错误。将USART_CTL2寄存器中的NKEN置位可以使能NACK信号。

空闲帧和断开帧在智能卡模式下不适用。

块模式(T=1)

在T=1(块模式)下，USART_CTL2寄存器的NKEN位应该清零来关闭校验错误发送。

当要从智能卡读取数据时，软件必须将USART_RT寄存器设置成BWT(块等待)-11的值并将RBNEIE置位。这个超时时间体现在波特时间单元。如果这个时间到了，还没有从智能卡收到应答，USART_STAT1寄存器中RTF位被置位。如果设置了USART_CTL3寄存器中RTIE位，将产生中断。如果在超时之前收到了第一个字节，则会引起RBNE中断。如果用DMA从智能卡读取数据，也只能在第一个字节接收好后再去使能DMA。

第一个字节接收到后，RT[23:0]的值设置成CWT(字节等待时间)-11来使能两个连续字节间最大帧间隔自动校验。如果在RT[23:0]周期内智能卡停止发送字节，USART_STAT1寄存器中RTF将被置位。

USART用一个块长度计数器统计收到的字节数，这个计数器在USART开始发送的时候自动清0(TBE=0)。这个块长度信息位于智能卡发出数据的第三个字节(序言部分)，这个值必须写入

USART_RT寄存器BL[7:0]。块长度计数器从0开始计数到最大值BL[7:0]+4。在块计数器计数到最大值时，USART_STAT1寄存器中块结束状态标志位EBF置位。如果设置了USART_CTL3寄存器中的EBIE位，将会产生中断。如果块长度发生错误，RTF置位。

当使用DMA模式接收时，在块开始之前，这个寄存器必须被设定为最小值(0x0)。为了得到这个值，在收到第四个字节后，会引起一个中断。软件可以从接收缓冲区读取第三个字节作为块长度。

如果接收时不使用DMA方式，为避免产生EBF状态标志，BL[7:0]需首先配置为最大值0xFF。在收到第三个字节后，真正的块长度值可以重新写入到BL[7:0]。

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，USART_CTL3寄存器中MSBF位和DINV位都为0。

如果选择反向转换，从数据帧的最高位开始传输，TX引脚高电平代表逻辑‘0’，偶校验。在这种情况下，USART_CTL3寄存器中MSBF位和DINV位都为1。

17.3.13. USART 中断

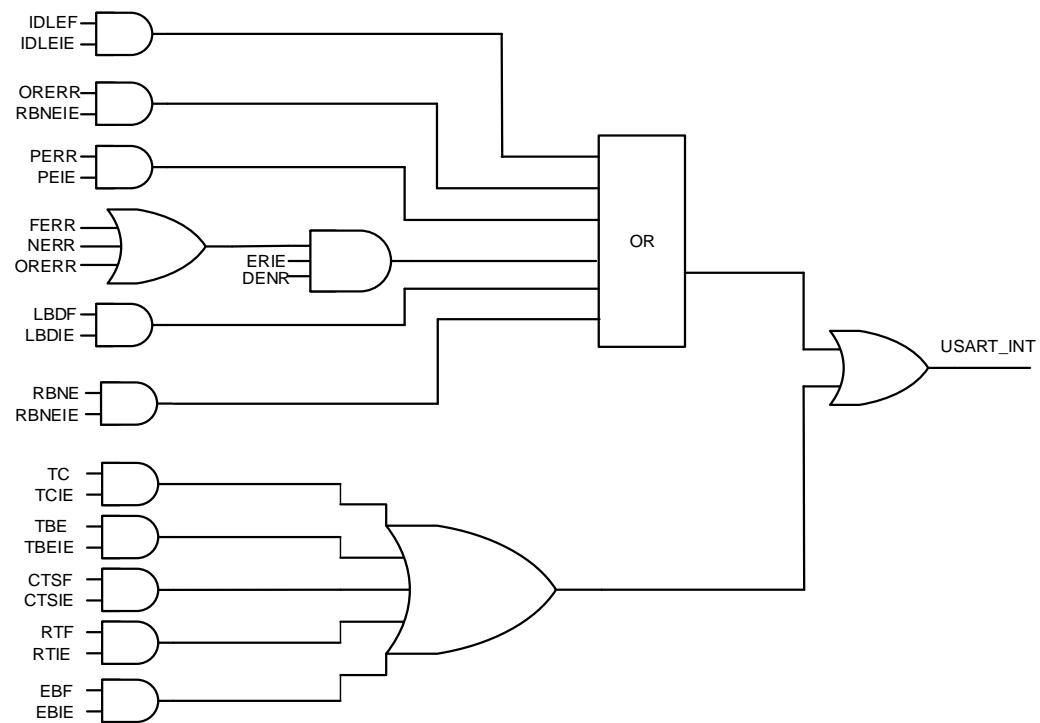
USART 中断事件和标志如表 17-3. USART 中断请求所示：

表 17-3. USART 中断请求

中断事件	事件标志	控制寄存器	使能控制位
发送数据寄存器空	TBE	USART_CTL0	TBEIE
CTS 标志	CTSF	USART_CTL2	CTSIE
发送结束	TC	USART_CTL0	TCIE
接收到的数据可以读取	RBNE	USART_CTL0	RBNEIE
检测到过载错误	ORERR		
检测到线路空闲	IDLEF	USART_CTL0	IDLEIE
奇偶校验错误	PERR	USART_CTL0	PERRIE
LIN模式下，检测到断开标志	LBDF	USART_CTL1	LBDIE
接收超时错误	RTF	USART_CTL3	RTIE
发现块尾	EBF	USART_CTL3	EBIE
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	USART_CTL2	ERRIE

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 17-16. USART 中断映射框图



17.4. USART 寄存器

USART0 基地址: 0x4001 3800

USART1 基地址: 0x4000 4400

USART2 基地址: 0x4000 4800

UART3 基地址: 0x4000 4C00

UART4 基地址: 0x4000 5000

17.4.1. 状态寄存器 0 (USART_STAT0)

地址偏移: 0x00

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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				CTSF	LBDF	TBE	TC	RBNE	IDLEF	ORERR	NERR	FERR	PERR		

位/位域	名称	描述
31:10	保留	必须保持复位值。
9	CTSF	<p>CTS变化标志</p> <p>如果设置了USART_CTL2寄存器中CTSEN位，当nCTS输入变化时，该位由硬件置位。如果设置了USART_CTL2寄存器中CTSIE位，将产生中断。</p> <p>该位由软件清0。</p> <p>0: nCTS状态线没有变化</p> <p>1: nCTS状态线发生变化</p> <p>该位对UART3/4无效。</p>
8	LBDF	<p>LIN断开检测标志</p> <p>寄存器USART_CTL1寄存器中LMEN置位，说明检测到LIN断开。如果USART_CTL1寄存器中LBDIE被置位时，将产生中断。</p> <p>该位由软件清0。</p> <p>0: 没有检测到LIN断开字符</p> <p>1: 检测到LIN断开字符</p>
7	TBE	<p>发送数据缓冲区空</p> <p>上电复位或待发送数据已发送至移位寄存器后，该位置1。USART_CTL0寄存器中TBEIE被置位将产生中断。</p>

		该位在软件将待发送数据写入USART_DATA时被清0。
6	TC	<p>0: 发送数据缓冲区不为空</p> <p>1: 发送数据缓冲区空</p> <p>发送完成</p> <p>上电复位后，该位被置1。如果TBE置位，在当前数据发送完成时该位置1。</p> <p>USART_CTL0寄存器中TCIE被置位将产生中断。</p> <p>该位由软件清0。</p> <p>0: 发送没有完成</p> <p>1: 发送完成</p>
5	RBNE	<p>读数据缓冲区非空</p> <p>当读数据缓冲区接收到来自移位寄存器的数据时，该位置1。当寄存器USART_CTL0的RBNEIE位被置位，将会有中断产生。</p> <p>软件可以通过对该位写0或读USART_DATA寄存器来将该位清0。</p> <p>0: 读数据缓冲区为空</p> <p>1: 读数据缓冲区不为空</p>
4	IDLEF	<p>空闲线检测标志</p> <p>在一个帧时间内，在RX引脚检测到空闲状态，该位置1。当寄存器USART_CTL0的IDLEIE位被置位，将会有中断产生。</p> <p>软件先读USART_STAT0，再读USART_DATA可清除该位。</p> <p>0: 未检测到空闲帧</p> <p>1: 检测到空闲帧</p>
3	ORERR	<p>溢出错误</p> <p>在RBNE置位的情况下，如果USART_DATA寄存器接收到来自移位寄存器的数据，该位置1。当寄存器USART_CTL0的RBNEIE位被置位，将会有中断产生。在多处理器通信或者DMA模式下，当寄存器USART_CTL2的ERRIE位被置位，将会有中断产生。</p> <p>软件先读USART_STAT0，再读USART_DATA可清除该位。</p> <p>0: 没有检测到溢出错误</p> <p>1: 检测到溢出错误</p>
2	NERR	<p>噪声错误标志</p> <p>在接收数据时，如果在RX引脚检测到噪声，该位被置位。当寄存器USART_CTL2的ERRIE位被置位，将会有中断产生。</p> <p>软件先读USART_STAT0，再读USART_DATA可清除该位。</p> <p>0: 没检测到噪声错误</p> <p>1: 检测到噪声错误</p>
1	FERR	<p>帧错误</p> <p>接收数据期间，在停止位传输过程中，RX引脚检测到低电平，该位被置位。当寄存器USART_CTL2的ERRIE位被置位，将会有中断产生。</p> <p>软件先读USART_STAT0，再读USART_DATA可清除该位。</p> <p>0: 未检测到帧错误</p>

1: 检测到帧错误

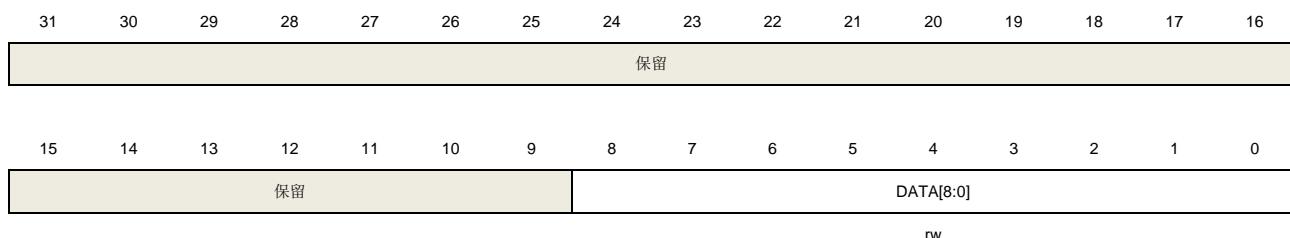
0	PERR	校验错误 当接收到的数据帧校验位与预期校验值不同时，该位置位。 软件先读USART_STAT0，再读USART_DATA可清除该位。 0: 没检测到校验错误 1: 检测到校验错误
---	------	---

17.4.2. 数据寄存器 (USART_DATA)

地址偏移: 0x04

复位值: 未定义

该寄存器只能按字(32位)访问



位/位域	名称	描述
31:9	保留	必须保持复位值。
8:0	DATA[8:0]	发送或接收的数据值 软件可以通过写这些位来改变发送数据，或读这些位的值来获取接收数据。 如果使能了奇偶校验，当发送数据被写入寄存器，数据的最高位(第7位或第8位取决于USART_CTL0寄存器的WL位)将被校验位取代。

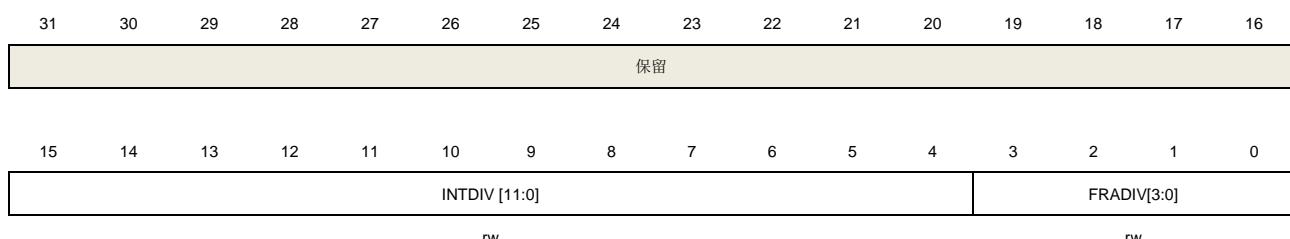
17.4.3. 波特率寄存器 (USART_BAUD)

地址偏移: 0x08

复位值: 0x0000 0000

使能USART(UEN=1)时，不能写该寄存器。

该寄存器只能按字(32位)访问



位/位域	名称	描述
------	----	----

31:16	保留	必须保持复位值。
15:4	INTDIV[11:0]	波特率分频器的整数部分
3:0	FRADIV [3:0]	波特率分频器的小数部分

17.4.4. 控制寄存器 0 (USART_CTL0)

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	UEN	WL	WM	PCEN	PM	PERRIE	TBEIE	TCIE	RBNEIE	IDLEIE	TEN	REN	RWU	SBKCMD	rw

位/位域	名称	描述
31:14	保留	必须保持复位值。
13	UEN	USART使能 0: USART禁用 1: USART使能
12	WL	字长 0: 8数据位 1: 9数据位
11	WM	从静默模式唤醒方法 0: 空闲线 1: 地址掩码
10	PCEN	校验控制使能 0: 校验控制禁用 1: 校验控制被使能
9	PM	校验模式 0: 偶校验 1: 奇校验
8	PERRIE	校验错误中断使能 如果该位置1, USART_STAT0寄存器中PERR被置位时产生中断。 0: 校验错误中断禁用 1: 校验错误中断使能
7	TBEIE	发送缓冲区空中断使能

		如果该位置1, USART_STAT0寄存器中TBE被置位时产生中断。
		0: 发送缓冲区空中断禁止 1: 发送缓冲区空中断使能
6	TCIE	发送完成中断使能 如果该位置1, USART_STAT0寄存器中TC被置位时产生中断。 0: 发送完成中断禁用 1: 发送完成中断使能
5	RBNEIE	读数据缓冲区非空中断和过载错误中断使能 如果该位置1, USART_STAT0寄存器中RBNE或ORERR被置位时产生中断。 0: 读数据缓冲区非空中断和过载错误中断禁用 1: 读数据缓冲区非空中断和过载错误中断使能
4	IDLEIE	IDLE线检测中断使能 如果该位置1, USART_STAT0寄存器中IDLEF被置位时产生中断。 0: IDLE线检测中断禁用 1: IDLE线检测中断禁用使能
3	TEN	发送器使能 0: 发送器禁用 1: 发送器使能
2	REN	接收器使能 0: 接收器禁用 1: 接收器使能
1	RWU	接收器从静默模式中唤醒 软件可以通过将该位置1使得USART进入静默模式, 将该位清0唤醒USART。 空闲帧唤醒模式下(WM=0), 当检测到空闲帧时, 该位由硬件清0。地址掩码模式下(WM=1), 当接收到一个地址匹配帧时, 该位由硬件清0; 或接收到一个地址非匹配帧时, 由硬件置1。 0: 接收器处于正常工作模式 1: 接收器处于静默模式
0	SBKCMD	发送断开帧 软件通过将该位置1发送断开帧。 断开帧传输结束由硬件清0。 0: 没有发送断开帧 1: 发送断开帧

17.4.5. 控制寄存器 1 (USART_CTL1)

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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	LMEN	STB[1:0]	CKEN	CPL	CPH	CLEN	保留	LBDIE	LBLEN	保留		ADDR[3:0]			

位/位域	名称	描述
31:15	保留	必须保持复位值。
14	LMEN	LIN模式使能 0: LIN模式禁用 1: LIN模式使能
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 对于UART3/4，只有1位停止位和2位停止位是有效的。
11	CKEN	CK引脚使能 0: CK引脚禁用 1: CK引脚使能 该位对于UART3/4无效。
10	CPL	时钟极性 该位用来设定在同步模式下CK引脚的极性。 0: CK引脚不对外发送时保持为低电平 1: CK引脚不对外发送时保持为高电平 该位对于UART3/4无效。
9	CPH	时钟相位 该位用来设定在同步模式下CK引脚的相位。 0: 在首个时钟边沿采样第一个数据 1: 在第二个时钟边沿采样第一个数据 该位对于UART3/4无效。
8	CLEN	CK信号长度 该位用来设定在同步模式下CK信号的长度。 0: 8位数据帧中有7个CK脉冲，9位数据帧中有8个CK脉冲 1: 8位数据帧中有8个CK脉冲，9位数据帧中有9个CK脉冲 该位对于UART3/4无效。
7	保留	必须保持复位值。
6	LBDIE	LIN断开信号检测中断使能

如果该位置1，当USART_STAT0寄存器中LBDF被置位时将产生中断。

0: 断开信号检测中断禁用

1: 断开信号检测中断使能

5	LBLEN	LIN断开帧长度 该位用来设定在断开帧长度。 0: 10位 1: 11位
4	保留	必须保持复位值。
3:0	ADDR[3:0]	USART地址 地址掩码唤醒模式下(WM=1)，如果接收到的数据帧低四位与ADDR[3:0]值不相等，USART就会进入静默模式；如果接收到的数据帧低四位与ADDR[3:0]值相等，USART就会被唤醒。

17.4.6. 控制寄存器2 (USART_CTL2)

地址偏移: 0x14

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				CTSIE	CTSEN	RTSEN	DENT	DENR	SCEN	NKEN	HDEN	IRLP	IREN	ERRIE	
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:11	保留	必须保持复位值。
10	CTSIE	CTS 中断使能 如果该位置1，当USART_STAT0寄存器中CTSF被置位时将产生中断。 0: CTS中断禁用 1: CTS中断使能 该位对于UART3/4无效。
9	CTSEN	CTS 使能 该位用于使能 CTS 硬件流控制功能。 0: CTS硬件流控制禁用 1: CTS硬件流控制使能 该位对于UART3/4无效。
8	RTSEN	RTS 使能 该位用于使能 RTS 硬件流控制功能。

		0: RTS硬件流控制禁用 1: RTS硬件流控制使能 该位对于UART3/4无效。
7	DENT	DMA发送使能 0: DMA发送模式禁用 1: DMA发送模式使能
6	DENR	DMA接收使能 0: DMA接收模式禁用 1: DMA接收模式使能
5	SCEN	智能卡模式使能 该位用于使能智能卡模式。 0: 智能卡模式禁用 1: 智能卡模式使能 该位对于UART3/4无效。
4	NKEN	在智能卡模式NACK使能 该位用于智能卡模式在奇偶校验错误发生时使能NACK发送。 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 该位对于UART3/4无效。
3	HDEN	半双工使能 该位用于使能半双工模式。 0: 半双工模式禁用 1: 半双工模式使能
2	IRLP	IrDA低功耗模式 该位用于为IrDA模式选择低功耗模式。 0: 正常模式 1: 低功耗模式
1	IREN	IrDA模式使能 0: IrDA禁用 1: IrDA使能
0	ERRIE	错误中断使能 当DMA接收模式(DENR=1)使能时，如果该位被置1，USART_STAT0寄存器中FERR, ORERR, NERR被置位将产生中断。 0: 错误中断禁用 1: 错误中断使能。

17.4.7. 保护时间和预分频器寄存器 (USART_GP)

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
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]	智能卡模式下的保护时间值。 TC标志置位时间延时GUAT[7:0]个波特时钟周期。 该位对于UART3/4无效。
7:0	PSC[7:0]	使能USART IrDA低功耗模式, 这些位用来设定将外设时钟(PCLK1/PCLK2)分频产生低功耗频率的分频系数。 00000000: 保留 – 不要写入该值 00000001: 对源时钟1分频 ... 11111111: 对源时钟255分频 在IrDA正常模式下, PSC只能设置成00000001。 在智能卡模式下, PSC[4:0]用于设定外设时钟(APB1/APB2)生成智能卡时钟的分频系数。实际的分频系数为PSC[4:0]设定值的两倍。 00000: 保留 – 不要写入该值 00001: 对源时钟2分频 00010: 对源时钟4分频 ... 11111: 对源时钟62分频 在智能卡模式下, PSC[7:5]保留。

17.4.8. 控制寄存器 3 (USART_CTL3)

偏移地址: 0x80

复位值: 0x0000 0000

UART3/4未使用该寄存器

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
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

保留	MSBF	DINV	TINV	RINV	保留	EBIE	RTIE	SCRTNUM[2:0]	RTEN
	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	MSBF	<p>高位在前</p> <p>该位用于设定数据在发送或接收时的顺序。</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> <p>USART被使能(UEN=1)时, 这一位不能被改写。</p>
10	DINV	<p>数据位反转</p> <p>该位用于设定在发送或接收时数据位的极性。</p> <p>0: 数据位信号值没有反转</p> <p>1: 数据位信号值被反转</p> <p>USART被使能(UEN=1)时, 这一位不能被改写。</p>
9	TINV	<p>TX 引脚电平反转</p> <p>该位用于设定TX引脚极性。</p> <p>0: TX引脚信号值没有反转</p> <p>1: TX引脚信号值被反转</p> <p>USART被使能(UEN=1)时, 这一位不能被改写。</p>
8	RINV	<p>RX引脚电平反转</p> <p>该位用于设定RX引脚极性。</p> <p>0: RX引脚信号值没有反转</p> <p>1: RX引脚信号值被反转</p> <p>USART被使能(UEN=1)时, 这一位不能被改写。</p>
7:6	保留	必须保持复位值。
5	EBIE	<p>块结束标志中断使能位</p> <p>如果该位置1, USART_STAT1寄存器中EBF被置位时产生中断。</p> <p>0: 块中断使能</p> <p>1: 块中断禁用</p>
4	RTIE	<p>接收超时标志中断使能位</p> <p>如果该位置1, USART_STAT1寄存器中RTF被置位时产生中断。</p> <p>0: 接收超时中断使能</p> <p>1: 接收超时中断禁用</p>
3:1	SCRTNUM[2:0]	<p>智能卡自动重试次数寄存器</p> <p>在智能卡模式下, 这些位用来设定在发送和接收时重试的次数。</p> <p>在发送模式下, 一帧数据可以重发 SCRTNUM 次。如果一帧数据发送失败 SCRTNUM+1次, FERR被置位。</p> <p>在接收模式下, USART接收一个数据帧可以执行SCRTNUM+1次。如果一个数据帧</p>

校验位不匹配事件产生SCRTNUM+1次，RBNE位和PERR位被置位。

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。

0	RTEN	接收器超时使能 该位用于使能USART接收超时。 0：接收器超时检测功能禁用 1：接收器超时检测功能被使能
---	------	--

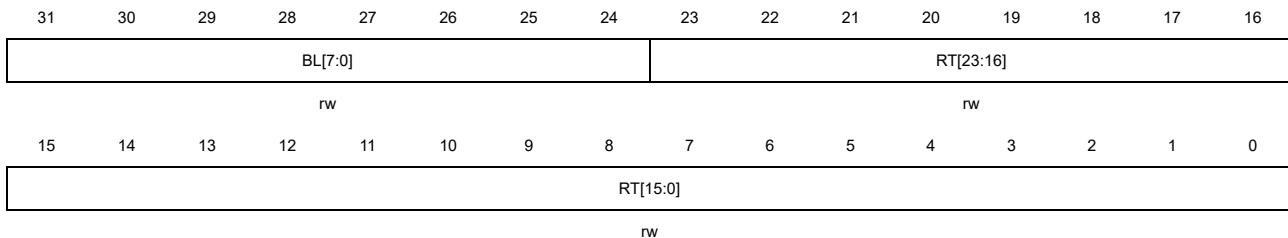
17.4.9. 接收超时寄存器 (USART_RT)

偏移地址: 0x84

复位值: 0x0000 0000

UART3/4未使用该寄存器

该寄存器只能按字(32位)访问



位/位域	名称	描述
31:24	BL[7:0]	块长度 这些位用于设定智能卡T=1的接收时，块的长度。它的值等于信息字节的长度+结束域的长度(1-LEC/2-CRC) – 1。 这个值可以在块接收开始去设置（采用从序言域中提取的LEN字段），这个值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。 在其他模式下，当REN=0 (禁用接收器)或者当USART_STAT1寄存器的EBF位被写0时，块的长度计数器被清0。
23:0	RT[23:0]	接收器超时阈值 该位域用于指定接收超时值，单位是波特时钟的时长。 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，USART_STAT1寄存器中RTF标志被置位。 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算的。 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。

17.4.10. 状态寄存器 1 (USART_STAT1)

偏移地址: 0x88

复位值: 0x0000 0000

UART3/4未使用该寄存器

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															BSY
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															保留
w0								w0							

位/位域	名称	描述
31:17	保留	必须保持复位值。
16	BSY	忙标志 USART接收一帧数据时被置位。 0: USART接收通道空闲 1: USART接收通道忙
15:13	保留	必须保持复位值。
12	EBF	块结束标志 该位在接收字节数(从块起始开始计数, 包含序言)等于或者大于BLEN+4时被置位。 USART_CTL3寄存器中EBIE被置位将产生中断。 软件可以通过写0清除该位。 0: 块结束事件没有发生 1: 块结束事件发生
11	RTF	接收超时标志 该位在RX引脚空闲时间已经超过RT值时被置位。USART_CTL3寄存器中RTIE被置位将产生中断。 软件可以通过写0清除该位。 0: 接收器超时事件没有发生 1: 接收器超时事件发生
10:0	保留	必须保持复位值。

17.4.11. 兼容性控制寄存器 (USART_CHC)

偏移地址: 0xC0

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	EPERR	保留	HCM
w0c		rw	

位/位域	名称	描述
31:9	保留	必须保持复位值。
8	EPERR	<p>校验错误超前检测标志。</p> <p>该标志将在RBNE置位前，校验位被检测到时置位。</p> <p>软件写0可以清除该位。</p> <p>0: 没有检测到校验错误</p> <p>1: 检测到校验错误</p>
7:1	保留	必须保持复位值。
0	HCM	<p>硬件流控制兼容模式</p> <p>0: nRTS信号与USART_STAT0寄存器中RBNE位相同</p> <p>1: nRTS信号在最后一个数据位被采样后被置位</p>

18. 内部集成电路总线接口（I2C）

18.1. 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。

18.2. 主要特征

- 并行总线至 I2C 总线协议的转换及接口；
- 同一接口既可实现主机功能又可实现从机功能；
- 主从机之间的双向数据传输；
- 支持 7 位和 10 位的地址模式和广播寻址；
- 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；
- 支持 SAM_V 模式；
- 数字滤波器。

18.3. 功能说明

I2C 接口的内部结构如 [图 18-1. I2C 模块框图](#) 所示。

图 18-1. I2C 模块框图

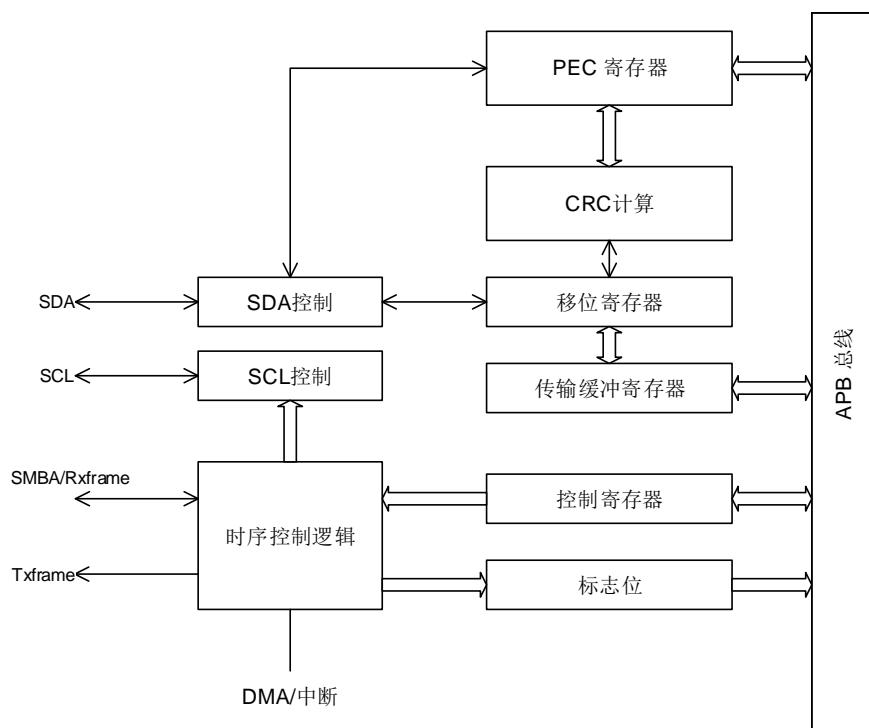


表 18-1. I2C 总线术语说明（参考飞利浦 I2C 规范）

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	多个主机在不破坏信息的前提下同时控制总线
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

18.3.1. SDA 线和 SCL 线

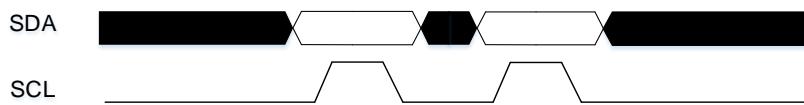
I2C 模块有两条接口线：串行数据 SDA 线和串行时钟 SCL 线。连接到总线上的设备通过这两根线互相传递信息。SDA 和 SCL 都是双向线，通过一个电流源或者上拉电阻接到电源正极。当总线空闲时，两条线都是高电平。连接到总线的设备输出极必须带开漏或者开集，以提供线与功能。I2C 总线上的数据在标准模式下可以达到 100 kbit/s，在快速模式下可以达到 400Kbit/s，当 I2C_FMPCFG 寄存器中 FMPEN 位被置位时，在快速+ 模式下可达 1Mbit/s。由于 I2C 总线上可能会连接不同工艺的设备 (CMOS, NMOS, 双极性器件)，逻辑‘0’和逻辑‘1’的电平并不

是固定的，取决于 V_{DD} 的实际电平。

18.3.2. 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 18-2. 数据有效性)。每个数据比特传输需要一个时钟脉冲。

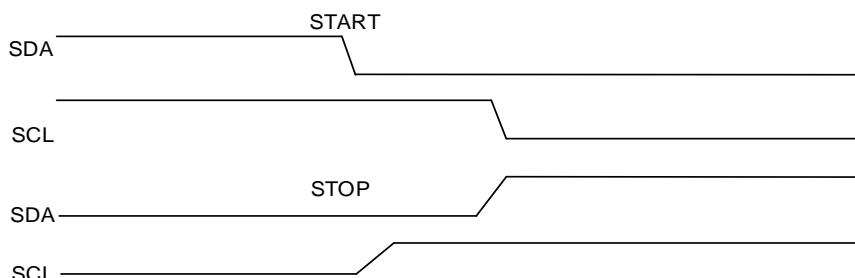
图 18-2. 数据有效性



18.3.3. 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 18-3. 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 18-3. 开始和停止状态

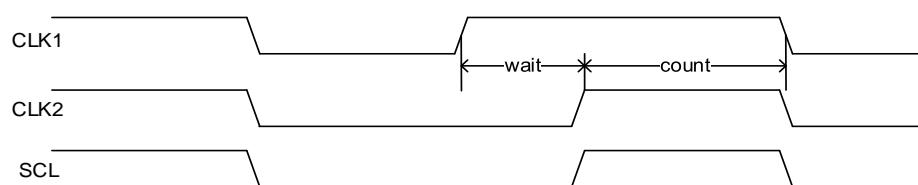


18.3.4. 时钟同步

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计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 18-4. 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 18-4. 时钟同步



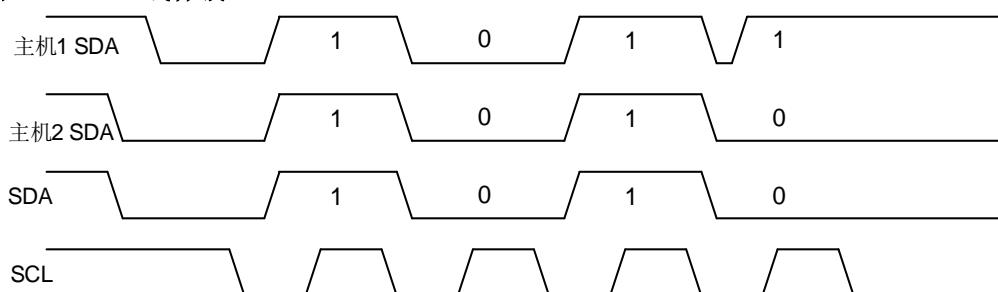
18.3.5. 仲裁

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 18-5. SDA 线仲裁



18.3.6. 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 18-6. 7 位地址的 I2C 通讯流程

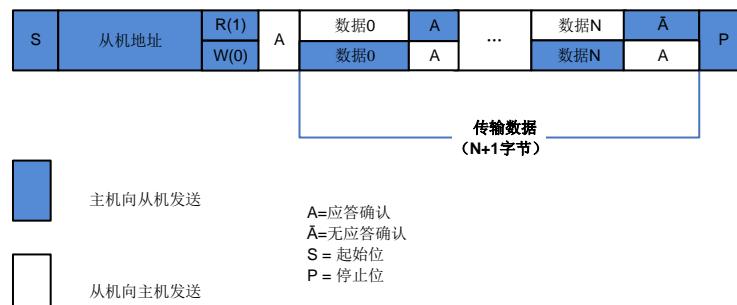
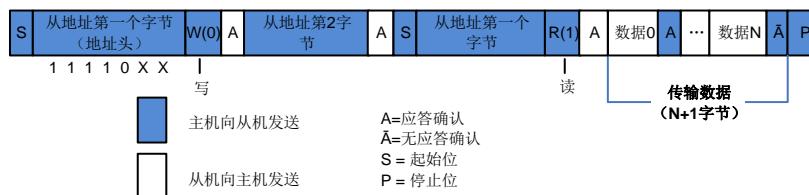


图 18-7. 10 位地址的 I2C 通讯流程（主机发送）



图 18-8. 10 位地址的 I2C 通讯流程（主机接收）



18.3.7. 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

如[图18-9. 从机发送模式](#)所示，在从机模式下要发送数据到I2C总线，软件应该按照以下步骤来运行操作：

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AERR（应答错误）会置起以通知软件发送结束。软件写0到AERR位可以清除此位。

图 18-9. 从机发送模式

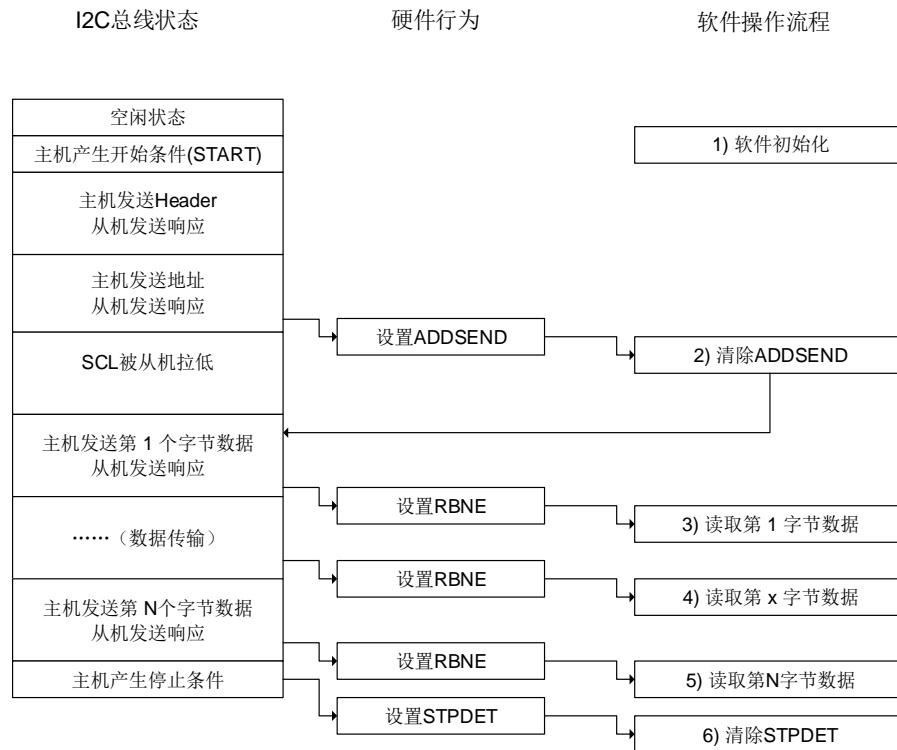


从机接收模式下的软件流程

如[图 18-10. 从机接收模式](#)所示，在从机模式下接收数据时，软件应该遵循以下步骤来操作：

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0的ADDSEND位置1，此位应该通过软件轮询或者中断来监测，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，软件可以从I2C_DATA寄存器读取第一个字节，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，软件可以读取最后一个字节。
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 18-10. 从机接收模式



主机发送模式下的软件流程

如[图18-11. 主机发送模式](#)所示，在主机模式下发送数据到I2C总线时，软件应该遵循以下步骤来运行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10位地址的地址头到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位地址的地址头，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寄存器的字节会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 18-11. 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，需要特别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

- 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。
- 软件将START位置1，从而在I2C在总线上产生一个START起始位。
- 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10位地址的地址头到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位地址的地址头，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。
- 7位或10位的地址位发送出去之后，I2C硬件将ADDSEND位置1，软件应该通过读I2C_STAT0寄存器然后读I2C_STAT1寄存器清除ADDSEND位。如果地址是10位格式，软

件应该再次将START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 18-12. 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START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10位地址的地址头到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位地址的地址头，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START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 18-13. 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ACKEN位清0，在清除ADDSEND位之后将STOP位置1。当N=1时步骤5是最最后一步。

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ACKEN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 18-13. 主机接收使用方案 B 模式



18.3.8. 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 SS 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。

18.3.9. DMA 模式下数据传输

按照前面的软件流程，每当 TBE 位或 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 状态位之后被置位。如果一个从机的 SCL 线延长功能被禁止，DMAON 位应该在 ADDSEND 事件前被置位。

参考 DMA 控制器的关于 DMA 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（EOT）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 EOT 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 ADDSEND 状态后或在 DMA EOT 中断 ISR 中，软件置位 STOP 位，产生一个停止状态。

18.3.10. 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 位和 PECTRANS 位被置 1，I2C 将自动发送或者检查 PEC 值。

18.3.11. 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 是基于 I2C 硬件实现的，它使用了 I2C 的硬件寻址方式，但在 I2C 的基础上增加了二级软件处理，建立自己独特的系统。比较特别的是 SMBus 规范包含一个地址解析协议，可用于实现

动态地址分配。动态识别硬件和软件使得总线设备能够支持热插拔，无需重启系统便能即插即用。总线中的设备将被自动识别并分配唯一地址。这个优点非常有利于实现即插即用的用户界面。在这些协议中，系统host与其它在系统中作为master或者slave的设备之间有一个非常有用的区别。

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还有一个额外可选的共享的中断信号，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（SMBSEL=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SMBALT标志位，并实现相应的功能。

18.3.12. SAM_V 支持

为了支持SAM_V标准，I2C模块增加两个附加的引脚：**txframe** 和 **rxframe**。当这个pin置高时，表示I2C正在主机模式下工作。**rxframe**是一个输入引脚，它应该和**SMBALERT**信号复用。

SAM_V 模式通过置位 I2C_SAMCS 寄存器的 SAMEN 位使能。txframe 和 rxframe 引脚的状态可以通过 I2C_SAMCS 寄存器的 RFR, RFF, TFR, TFF, RXF 和 TXF 标志反映。如果对应的中断使能位置位，将产生 I2C 中断。

18.3.13. 状态、错误和中断

I2C有一些状态、错误标志位，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 18-2. 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
RFR	SAM_V 模式时检测到 rxframe 上升沿
RFF	SAM_V 模式时检测到 rxframe 下降沿
TFR	SAM_V 模式时检测到 txframe 上升沿
TFF	SAM_V 模式时检测到 txframe 下降沿

表 18-3. 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

18.4. I2C 寄存器

I2C0 基地址: 0x4000 5400

I2C1 基地址: 0x4000 5800

18.4.1. 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	PECTRA NS	POAP	ACKEN	STOP	START	SS	GCEN	PECEN	ARPEN	SMBSEL	保留	SMBEN	I2CEN

rw rw

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， 或监测到START/STOP结束位， 或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位决定对当前正在接收的字节是否发送ACK/NACK； 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/NACK， PECTRANS位表明下一个即将被接收的字节是PEC
10	ACKEN	ACK使能 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	SS	SCL拉低 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	广播呼叫使能 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使能 0: PEC计算禁用 1: PEC计算使能
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

18.4.2. 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[5:0]					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: 禁用缓冲区中断 1: 使能缓冲区中断, 如果EVIE=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BUFIE=1时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BERR、LOSTARB、AERR、OUERR、PECERR、SMBTO或SMBALT标志位生效时产生中断。
7:6	保留	必须保持复位值。
5:0	I2CCLK[5:0]	I2C外设时钟频率 I2CCLK[5:0]应该是输入APB1时钟频率, 最低2MHz。 000000 - 000001: 无时钟 000010 - 111100: 2 MHz~60MHz 111101 - 111111: 由于APB1时钟限制, 无时钟 注意: 在标准模式下, APB1时钟频率需大于或者等于2MHz。在快速模式下, APB1时钟频率需大于或者等于8MHz。在快速+模式下, APB1时钟频率需大于或者等于24MHz。

18.4.3. 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
ADDFOR MAT	保留				ADDRESS[9:8]	ADDRESS[7:1]				ADDRESS S0					

位/位域	名称	描述
15	ADDFORMAT	I2C从机地址格式 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

18.4.4. 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			

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	双重地址模式使能 0: 禁用双重地址模式 1: 使能双重地址模式

18.4.5. 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。

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

18.4.6. 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。



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引脚被拉低且接收到警报地址(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但校验错误，此时I2C将无视ACKEN位直接发送NACK
11	OUERR	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。 硬件置1，软件写0清0。 0: 无溢出和欠载错误发生 1: 发生溢出或欠载错误
10	AERR	应答错误 硬件置1，软件写0清0。 0: 未发生应答错误 1: 发生了应答错误

9	LOSTARB	主机模式下仲裁丢失 硬件置1，软件写0清0。 0: 无仲裁丢失 1: 发生仲裁丢失，I2C模块返回从机模式。
8	BERR	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。 硬件置1，软件写0清0。 0: 无总线错误 1: 发生了总线错误
7	TBE	发送期间I2C_DATA为空 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） 0: I2C_DATA非空 1: I2C_DATA空，软件可以写
6	RBNE	接收期间I2C_DATA非空 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I2C_DATA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将被立即移到I2C_DATA。（详见主机/从机接收模式下的软件操作流程） 0: I2C_DATA为空 1: I2C_DATA非空，软件可以读
5	保留	必须保持复位值。
4	STPDET	从机模式下监测到STOP结束位 此位被硬件置1，先读I2C_STAT0然后写I2C_CTL0可以清除此位。 0: 从机模式下未监测到STOP结束位 1: 从机模式下监测到STOP结束位
3	ADD10SEND	主机模式下10位地址地址头被发送 该位由硬件置1，软件读I2C_STAT0和写I2C_DATA清除此位。 0: 主机模式下未发送10位地址的地址头 1: 主机模式下发送10位地址的地址头
2	BTC	字节发送结束 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，硬件就会置起BTC标志位。 此位由硬件置1。 可由以下三种方式清除： 软件清除：读I2C_STAT0，然后读或写I2C_DATA寄存器清除此位 硬件清除：发送一个STOP或START信号 寄存器I2C_CTL0中I2CEN=0 0: 未发生BTC

1: 发生了BTC

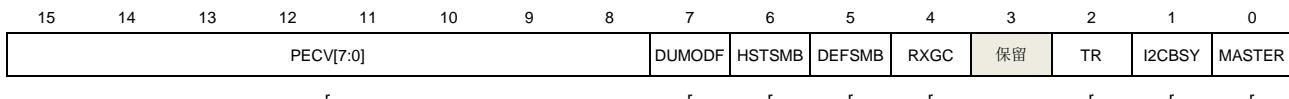
1	ADDSEND	主机模式下：成功发送了地址 从机模式下：接收到了地址并且和自身的地址匹配 此位由硬件置1，软件读I2C_STAT0寄存器和读I2C_STAT1清0。 0: 无地址被发送或接收 1: 地址在主机模式下被发送或从机模式下接收到匹配地址
0	SBSEND	主机模式下发送START起始位 此位由硬件置1，软件读I2C_STAT0和写I2C_DATA清0。 0: 未发送START条件 1: START条件被发送

18.4.7. 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。



位/位域	名称	描述
15:8	PECV[7:0]	当PEC使能后硬件计算出的PEC值。
7	DUMODF	从机模式下双标志位表明哪个地址和双地址模式匹配 STOP或START信号产生后或I2CEN=0时此位由硬件清0。 0: 地址和I2C_SADDR0匹配 1: 地址和I2C_SADDR1匹配
6	HSTSMB	从机模式下监测到SMBus主机地址头 STOP或START信号产生后或I2CEN=0时此位由硬件清0。 0: 未监测到SMBus主机地址头 1: 监测到SMBus主机地址头
5	DEFSMB	SMBus设备缺省地址 STOP或START信号产生后或I2CEN=0时此位由硬件清0。 0: SMBus设备没有缺省地址 1: SMBus设备接收到一个缺省地址
4	RXGC	是否接收到广播地址(0x00) STOP或START信号产生后或I2CEN=0时此位由硬件清0。 0: 未接收到广播呼叫地址(0x00) 1: 接收到广播呼叫地址(0x00)

3	保留	必须保持复位值。
2	TR	<p>发送端或接收端</p> <p>该位表明I2C作发送端还是接收端, STOP或START信号产生后或I2CEN=0或LOSTARB=1时此位由硬件清0。</p> <p>0: 接收端</p> <p>1: 发送端</p>
1	I2CBSY	<p>忙标志</p> <p>STOP结束位后硬件清0。</p> <p>0: 无I2C通讯</p> <p>1: I2C正在通讯</p>
0	MASTER	<p>主机模式</p> <p>表明I2C时钟在主机模式还是从机模式的标志位, START信号产生后此位由硬件置1, STOP信号产生后或I2CEN=0或LOSTARB=1时此位由硬件清0。</p> <p>0: 从机模式</p> <p>1: 主机模式</p>

18.4.8. 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]	<p>主机模式下I2C时钟控制</p> <p>标准速度模式下: $T_{high}=T_{low}=CLKC*T_{PCLK1}$</p> <p>如果 DTCY=0, 快速模式或快速+ 模式下:</p> <p>$T_{high}=CLKC*T_{PCLK1}$, $T_{low}=2*CLKC*T_{PCLK1}$</p> <p>如果 DTCY=1, 快速模式或快速+ 模式下:</p> <p>$T_{high}=9*CLKC*T_{PCLK1}$, $T_{low}=16*CLKC*T_{PCLK1}$</p>

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。

18.4.9. 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[5:0]							
rw															

位/位域	名称	描述
15:6	保留	必须保持复位值。
5:0	RISETIME[5:0]	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

18.4.10. SAM 控制状态寄存器 (I2C_SAMCS)

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。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RFR	RFF	TFR	TFF	保留	RXF	TXF	RFRIE	RFFIE	TFRIE	TFFIE	保留	STOEN	SAMEN		
r_w0	r_w0	r_w0	r_w0		r	r	rw	rw	rw	rw		rw	rw		

位/位域	名称	描述
15	RFR	接收帧上升沿标志, 由软件写0清除
14	RFF	接收帧下降沿标志, 由软件写0清除
13	TFR	发送帧上升沿标志, 由软件写0清除
12	TFF	发送帧下降沿标志, 由软件写0清除
11:10	保留	必须保持复位值。
9	RXF	接收帧信号电平
8	TXF	发送帧信号电平
7	RFRIE	接收帧上升沿中断使能 0: 接收帧上升沿中断失能

		1: 接收帧上升沿中断使能
6	RFFIE	接收帧下降沿中断使能 0: 接收帧下降沿中断失能 1: 接收帧下降沿中断使能
5	TFRIE	发送帧上升沿中断使能 0: 发送帧上升沿中断失能 1: 发送帧上升沿中断使能
4	TFFIE	发送帧下降沿中断使能 0: 发送帧下降沿中断失能 1: 发送帧下降沿中断使能
3:2	保留	必须保持复位值。
1	STOEN	SAM_V接口超时检测使能 0: SAM_V接口超时检测失能 1: SAM_V接口超时检测使能
0	SAMEN	SAM_V 接口使能 0: SAM_V 接口失能 1: SAM_V 接口使能

18.4.11. 快速+ 模式配置寄存器 (I2C_FMPCFG)

地址偏移: 0x90

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														FMPE	
														RW	

位/位域	名称	描述
15:1	保留	必须保持复位值。
0	FMPE	快速+ 模式使能 当该位被置1时， I2C设备支持高达1MHz 0: 失能快速+ 模式 1: 使能快速+ 模式

19. 串行外设接口/I片上音频接口（SPI/I2S）

19.1. 简介

SPI/I2S模块可以通过SPI协议或I2S音频协议与外部设备进行通信。

串行外设接口（Serial Peripheral Interface，缩写为SPI）提供了基于SPI协议的数据发送和接收功能，可以工作于主机或从机模式。SPI接口支持具有硬件CRC计算和校验的全双工和单工模式。只有SPI0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。

19.2. 主要特征

19.2.1. SPI 主要特征

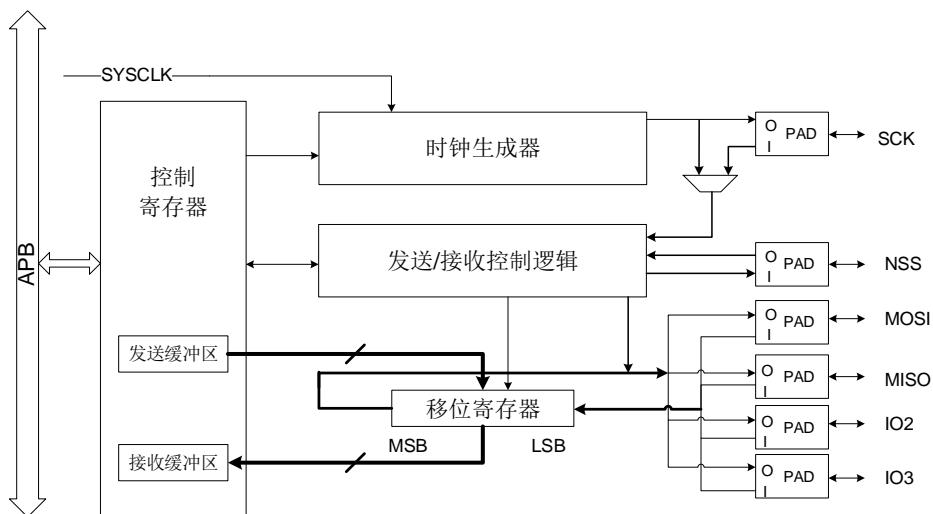
- 具有全双工和单工模式的主从操作；
- 16位宽度，独立的发送和接收缓冲区；
- 8位或16位数据帧格式；
- 低位在前或高位在前的数据位顺序；
- 软件和硬件NSS管理；
- 硬件CRC计算、发送和校验；
- 发送和接收支持DMA模式；
- 支持SPI TI模式；
- 支持SPI NSS脉冲模式；
- 支持SPI四线功能的主机模式（只有SPI0）。

19.2.2. I2S 主要特征

- 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。

19.3. SPI 结构框图

图 19-1. SPI 结构框图



19.4. SPI 信号线描述

19.4.1. 常规配置（非 SPI 四线模式）

表 19-1. 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。

19.4.2. SPI 四线配置

SPI默认配置为单路模式，当SPI_QCTL中的QM0D位置1时，配置为SPI四线模式（只适用于

SPI0)。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 19-2. SPI 四线信号描述

引脚名称	方向	描述
SCK	O	SPI 时钟输出
MOSI	I/O	发送或接收数据 0
MISO	I/O	发送或接收数据 1
IO2	I/O	发送或接收数据 2
IO3	I/O	发送或接收数据 3
NSS	O	NSS 输出

19.5. SPI 功能说明

19.5.1. 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 19-2. 常规模式下的 SPI 时序图

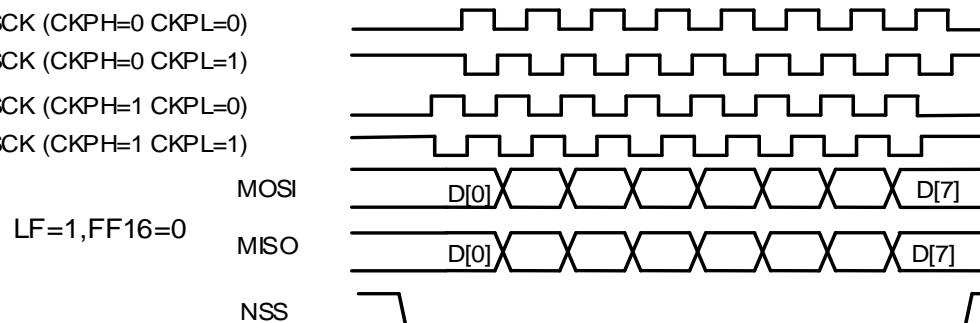
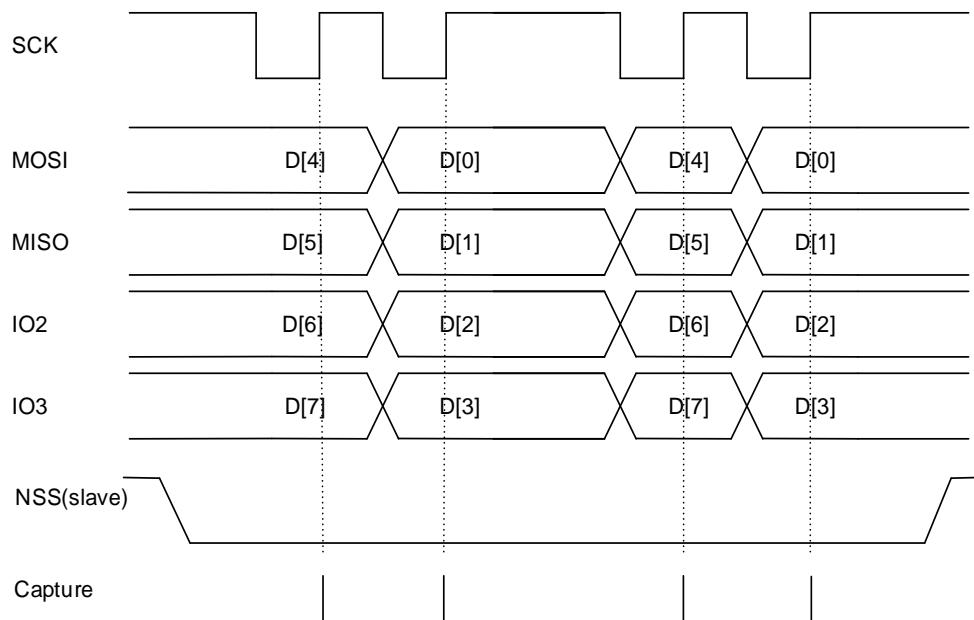


图 19-3. SPI 四线模式下的 SPI 时序图(CKPL=1, CKPH=1, LF=0)



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。

19.5.2. 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。当禁用SPI时，NSS变为高电平。

应用程序可以使用一个通用I/O口作为NSS引脚，以实现更加灵活的NSS应用。

19.5.3. SPI 运行模式

表 19-3. SPI 运行模式

模式	描述	寄存器配置	数据引脚用法
MFD	全双工主机模式	MSTMOD = 1 RO = 0 BDEN = 0 BDOEN: 不要求	MOSI: 发送 MISO: 接收
MTU	单向线连接主机发送模式	MSTMOD = 1 RO = 0 BDEN = 0 BDOEN: 不要求	MOSI: 发送 MISO: 不使用
MRU	单向线连接主机接收模式	MSTMOD = 1 RO = 1 BDEN = 0 BDOEN: 不要求	MOSI: 不使用 MISO: 接收
MTB	双向线连接主机发送模式	MSTMOD = 1 RO = 0 BDEN = 1 BDOEN = 1	MOSI: 发送 MISO: 不使用
MRB	双向线连接主机接收模式	MSTMOD = 1 RO = 0 BDEN = 1 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	数据引脚用法
		BDOEN = 0	

图 19-4. 典型的全双工模式连接

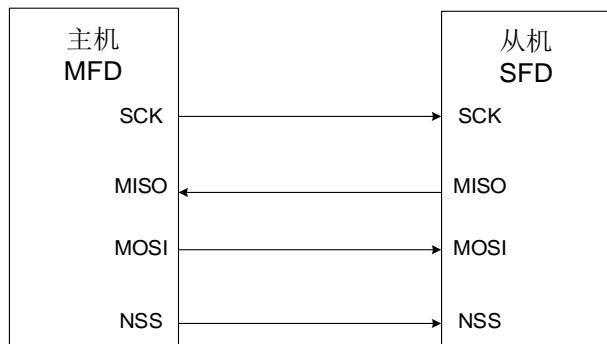


图 19-5. 典型的单工模式连接（主机：接收，从机：发送）

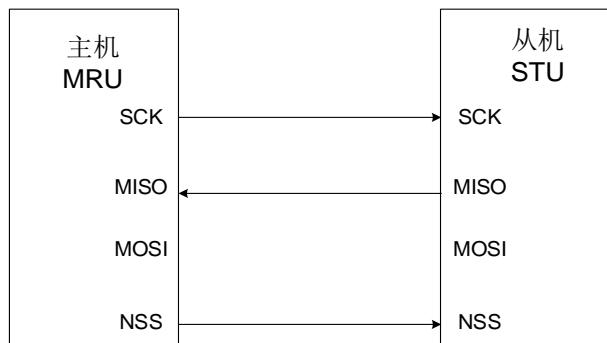


图 19-6. 典型的单工模式连接（主机：只发送，从机：接收）

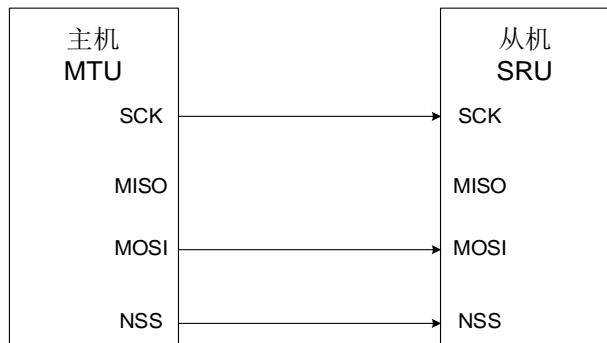
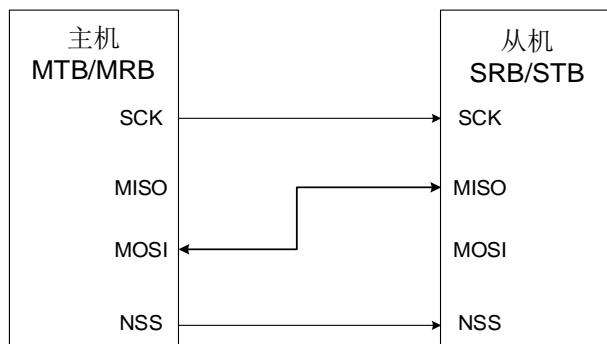


图 19-7. 典型的双向线连接



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[表19-3. SPI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，仅当发送缓冲区非空时，硬件才接收下一个数据帧。

SPI 不同模式下的操作流程（非 SPI 四线模式，TI 模式或 NSSP 模式）

在全双工模式下，无论是MFD模式或者SFD模式，应用程序都应该监视RBNE标志位和TBE标志位，并且遵循上文描述的操作流程。

发送模式（MTU, MTB, STU或STB）与全双工模式中的发送流程类似，不同的是需要忽略RBNE位和OVRE位。

相比于发送模式的情况，主机接收模式（MRU或MRB）与全双工的接收流程大不相同。在MRU模式或MRB模式下，在SPI使能后，SPI产生连续的SCK信号，直到SPI停止。所以，软件应该

忽略TBE标志位，并且在RBNE位置1后，读出接收缓冲区内的数据，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 19-8. 主机 TI 模式在不连续发送时的时序图

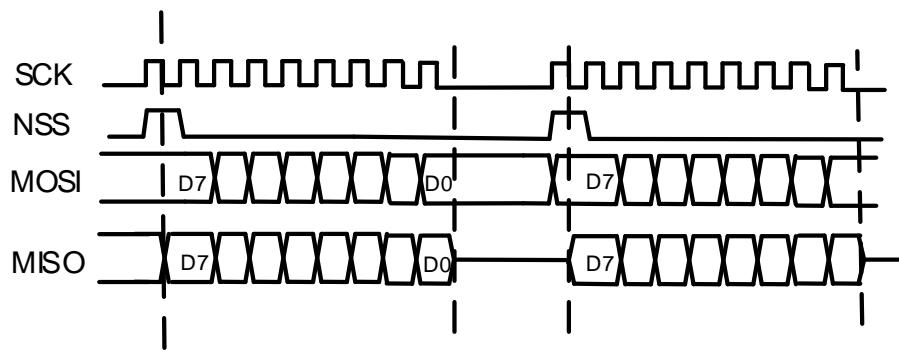
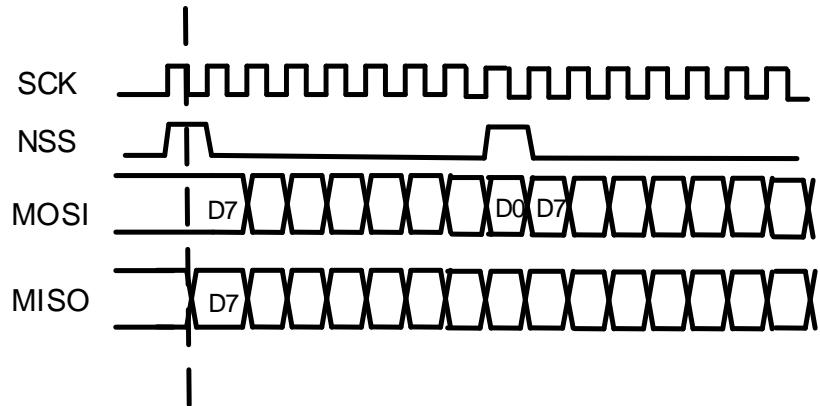
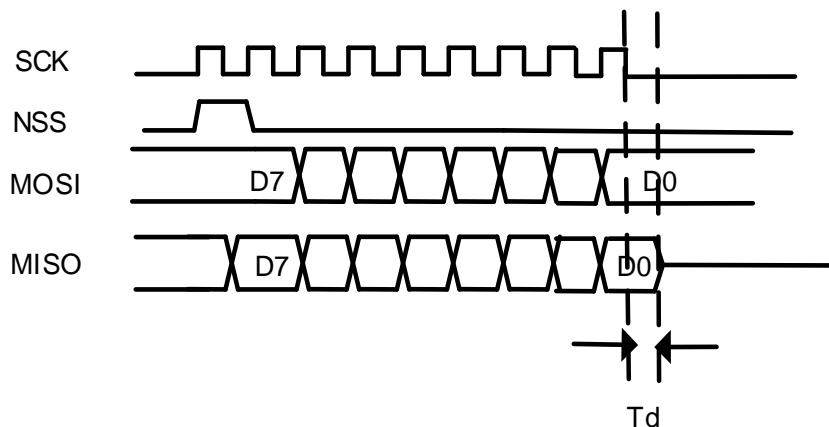


图 19-9. 主机 TI 模式在连续发送时的时序图



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位的时钟周期覆盖。

图 19-10. 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 T_d ， T_d 通过SPI_CTL0寄存器中的PSC[2:0]位来设置。

$$T_d = \frac{T_{bit}}{2} + 5 * T_{pclk} \quad (19-1)$$

例如，如果PSC[2:0] = 010，那么 T_d 数值为 $9 * T_{pclk}$ 。

在从机模式下，从机需要监视NSS信号，如果检测到错误的NSS信号，将会置位FERR标志位。例如，NSS信号在一个字节的中间位发生翻转。

NSS 脉冲模式操作流程

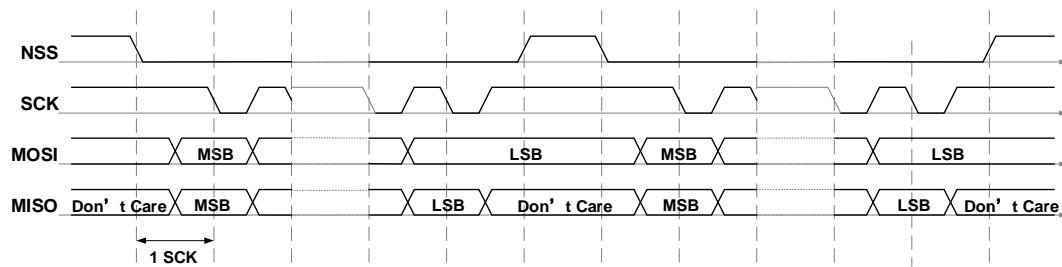
配置SPI_CTL1寄存器中的NSSP位使能该功能，为了确保使用该功能实现，需满足以下几个条件：配置设备为主机模式，使用普通SPI协议的数据帧格式，同时在第一个时钟跳变沿采样数据。

总之：MSTMOD = 1， NSSP = 1， CKPH = 0。

当使用NSS脉冲模式时，根据内部数据发送缓冲区的状态，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 19-11. 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。

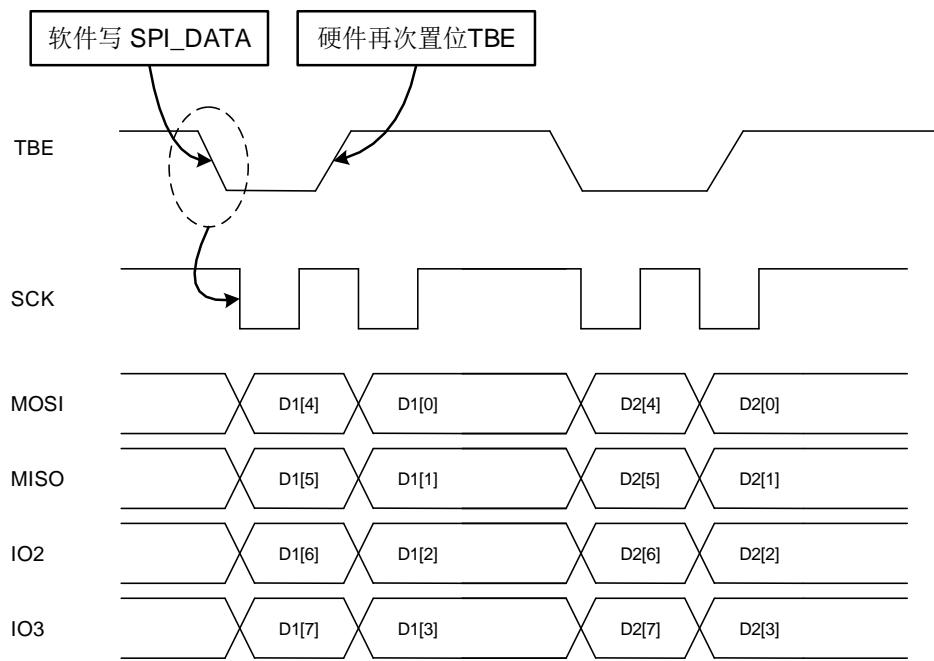
四线写模式

当SPI_QCTL寄存器中的QMOD位置1且QRD位清零时，SPI工作在四路写模式。在四路写模式中，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。

四路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 19-12. SPI 四线模式四线写操作时序图



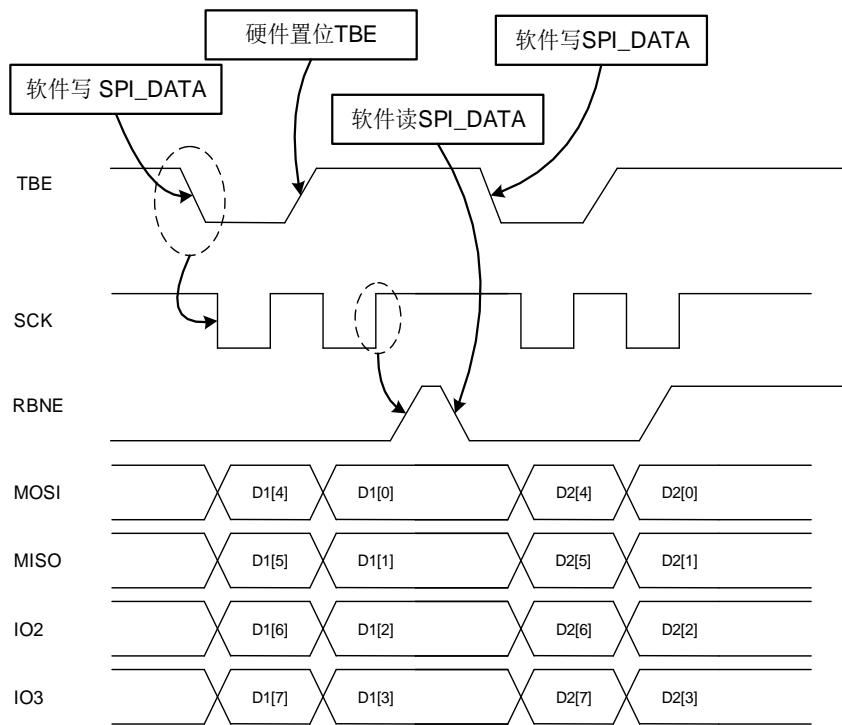
四线读模式

当SPI_QCTL寄存器中的QM0D位都置1时，SPI工作在四路读模式。在四路读模式中，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。

四路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QM0D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 19-13. SPI 四路模式四路读操作时序图



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四路模式

在禁用SPI四路模式和关闭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。

19.5.4. 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。

19.5.5. 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CRCEN位使能CRC功能。对于数据线上每个发送和接收的数据，CRC单元逐位计算CRC值，计算得到的CRC值可以从SPI_TCRC寄存器和SPI_RCRC寄存器中读取。

为了传输计算得到的CRC值，应用程序需要在最后一个数据写入发送缓冲区之后，设置SPI_CTL0中的CRCNT位。在全双工模式（MFD或SFD），当SPI发送一个CRC值并且准备校验接收到的CRC值时，会将最新接收到的数据当作CRC值。在接收模式（MRB，MRU，SRU和SRB）下，在倒数第二个数据帧被接收后，软件将CRCNT位置1。在CRC校验失败时，CRCERR错误标志位将会置1。

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。

19.6. SPI 中断

19.6.1. 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。

19.6.2. 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）

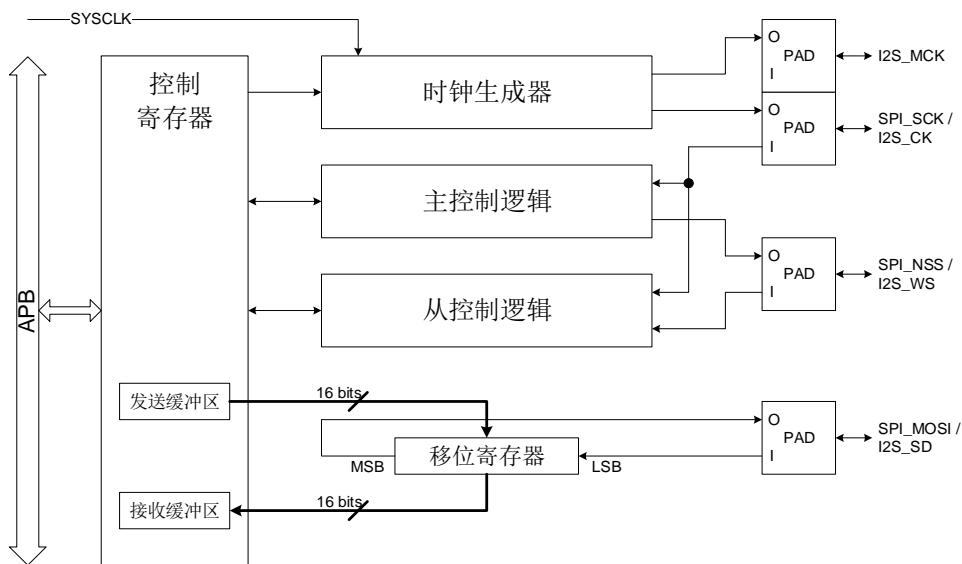
当CRCEN位置1时，SPI_RCRC寄存器中接收到的数据的CRC计算值将会和紧随着最后一帧数据接收到的CRC值进行比较，当两者不同时，CRCERR位将会置1。

表 19-4. 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	

19.7. I2S 结构框图

图 19-14. 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I2S_CK和I2S_WS信号来控制从机模式的通信。移位寄存器控制I2S_SD上的串行数据发送和接收。

19.8. 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。

19.9. I2S 功能说明

19.9.1. 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 19-15. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=0, CKPL=0）

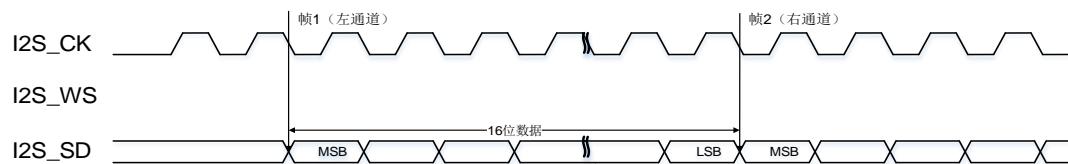
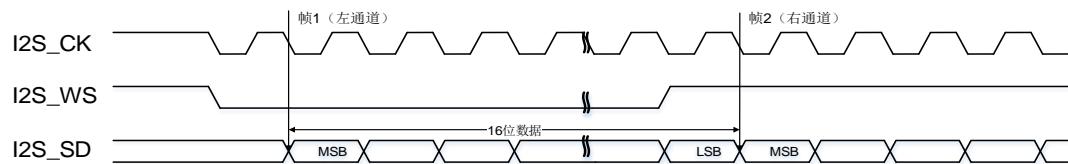


图 19-16. I2S 飞利浦标准时序图（DTLEN=00, CHLEN=0, CKPL=1）



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 19-17. I2S 飞利浦标准时序图（DTLEN=10, CHLEN=1, CKPL=0）

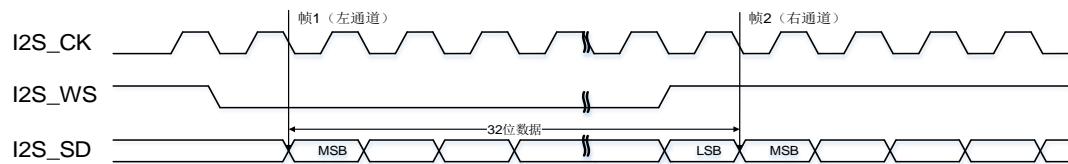
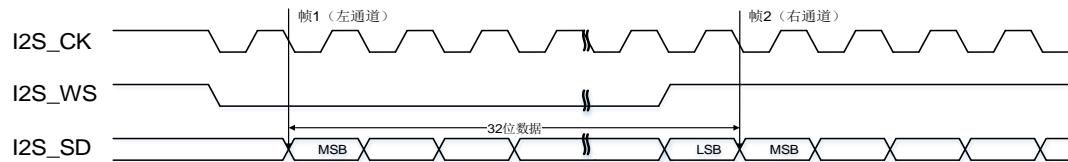


图 19-18. I2S 飞利浦标准时序图（DTLEN=10, CHLEN=1, CKPL=1）



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

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 19-19. I2S 飞利浦标准时序图 (DTLEN=01, CHLEN=1, CKPL=0)

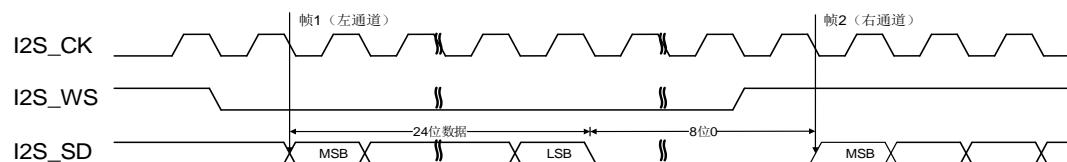
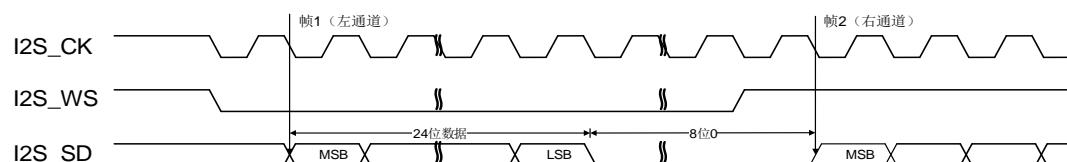


图 19-20. I2S 飞利浦标准时序图 (DTLEN=01, CHLEN=1, CKPL=1)



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位数据全都是0。

图 19-21. I2S 飞利浦标准时序图 (DTLEN=00, CHLEN=1, CKPL=0)

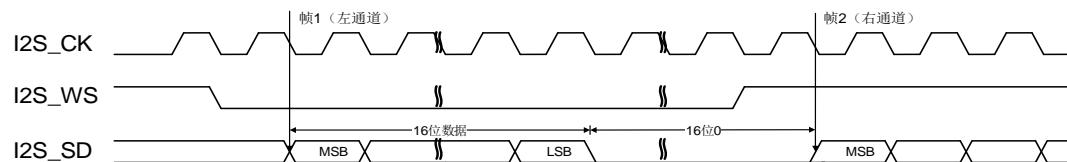
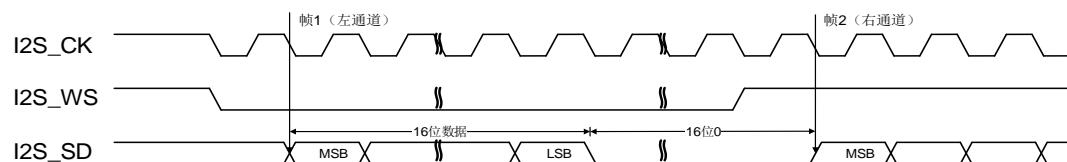


图 19-22. I2S 飞利浦标准时序图 (DTLEN=00, CHLEN=1, CKPL=1)



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 19-23. MSB 对齐标准时序图 (DTLEN=00, CHLEN=0, CKPL=0)

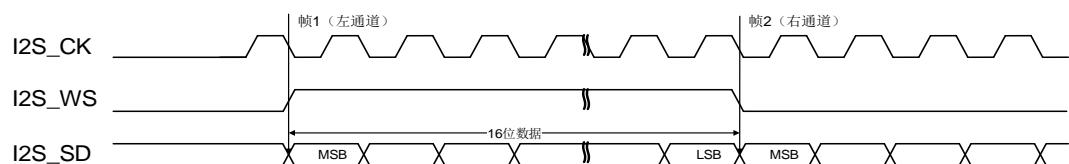


图 19-24. MSB 对齐标准时序图 (DTLEN=00, CHLEN=0, CKPL=1)

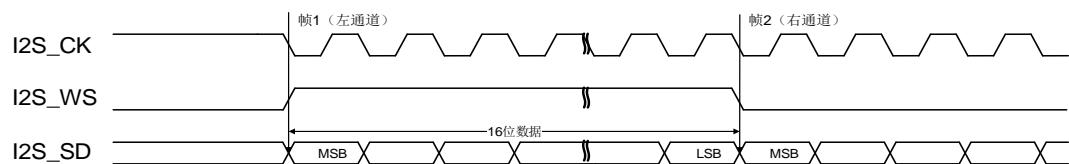


图 19-25. MSB 对齐标准时序图 (DTLEN=10, CHLEN=1, CKPL=0)

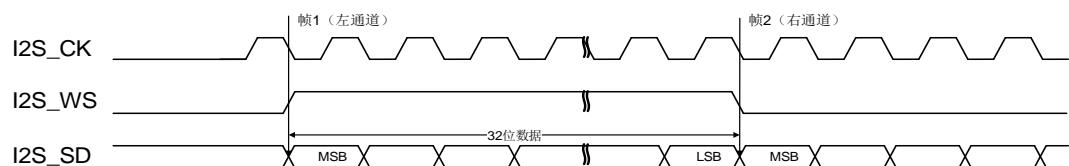


图 19-26. MSB 对齐标准时序图 (DTLEN=10, CHLEN=1, CKPL=1)

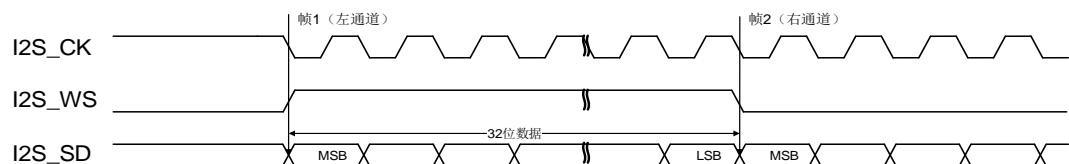


图 19-27. MSB 对齐标准时序图 (DTLEN=01, CHLEN=1, CKPL=0)

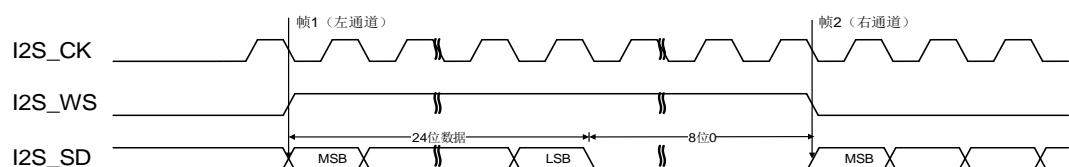


图 19-28. MSB 对齐标准时序图 (DTLEN=01, CHLEN=1, CKPL=1)

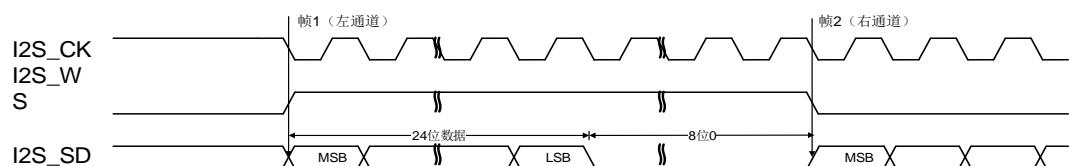


图 19-29. MSB 对齐标准时序图 (DTLEN=00, CHLEN=1, CKPL=0)

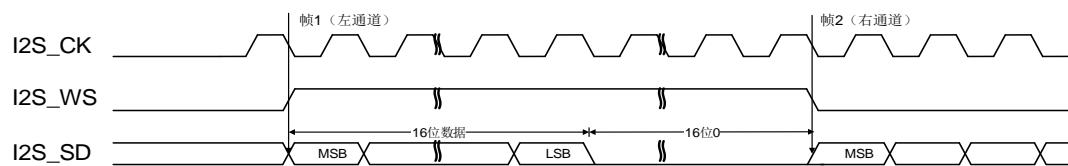
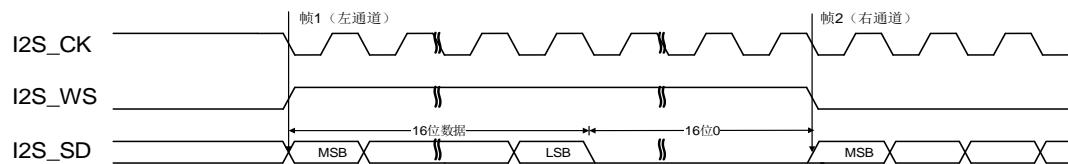


图 19-30. MSB 对齐标准时序图 (DTLEN=00, CHLEN=1, CKPL=1)



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 19-31. LSB 对齐标准时序图 (DTLEN=01, CHLEN=1, CKPL=0)

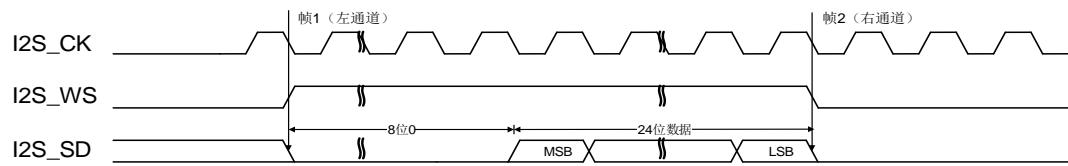
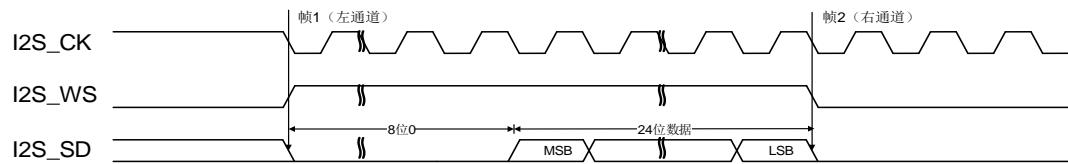


图 19-32. LSB 对齐标准时序图 (DTLEN=01, CHLEN=1, CKPL=1)



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 19-33. LSB 对齐标准时序图 (DTLEN=00, CHLEN=1, CKPL=0)

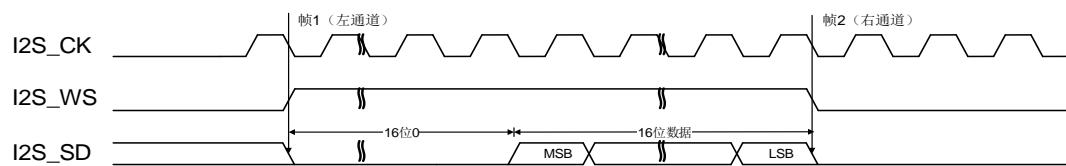
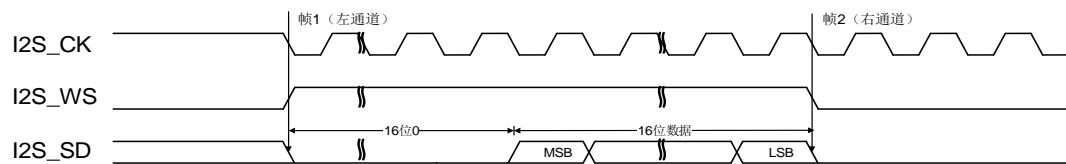


图 19-34. LSB 对齐标准时序图 (DTLEN=00, CHLEN=1, CKPL=1)



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 19-35. PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=0)

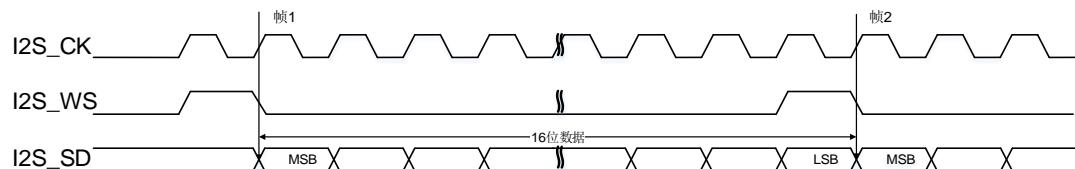


图 19-36. PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=1)

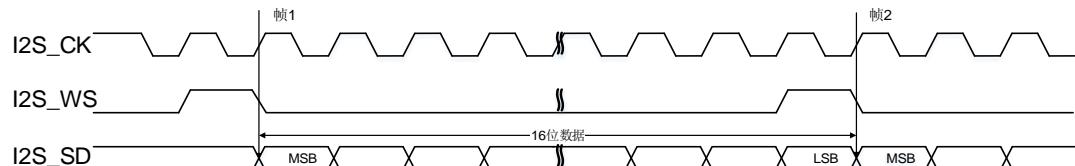


图 19-37. PCM 标准短帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=0)

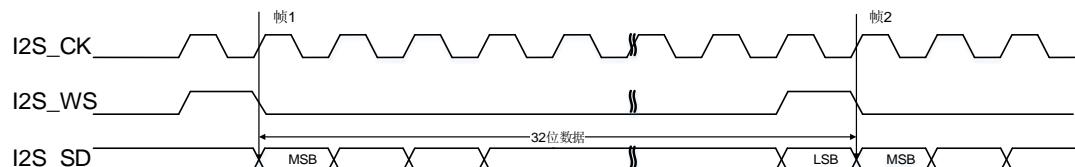


图 19-38. PCM 标准短帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=1)

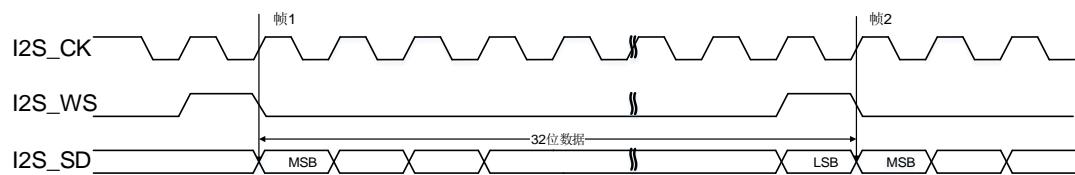


图 19-39. PCM 标准短帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=0)

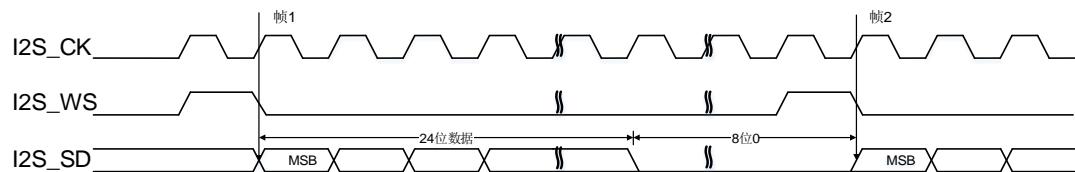


图 19-40. PCM 标准短帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=1)

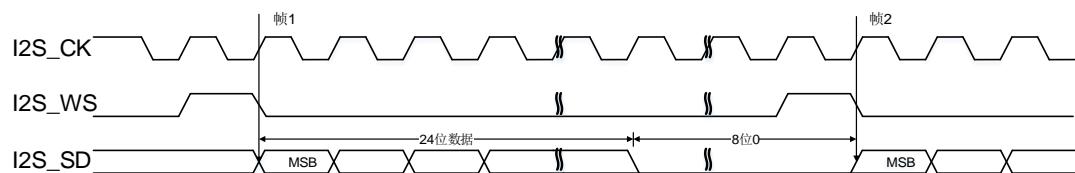


图 19-41. PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=0)

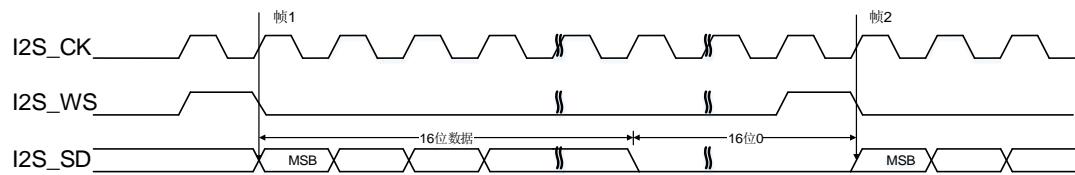
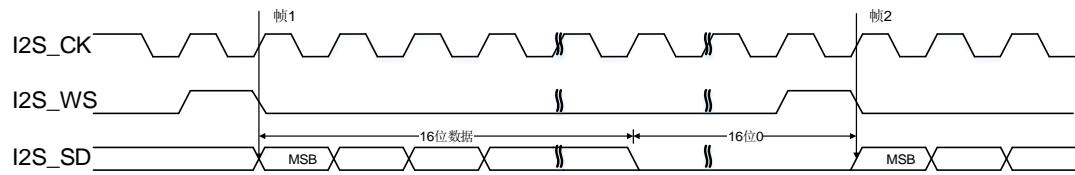


图 19-42. PCM 标准短帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=1)



长帧同步模式的各种配置情况时序图如下所示。

图 19-43. PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=0)

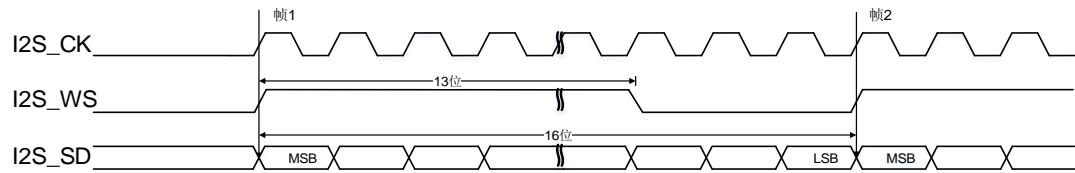


图 19-44. PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=0, CKPL=1)

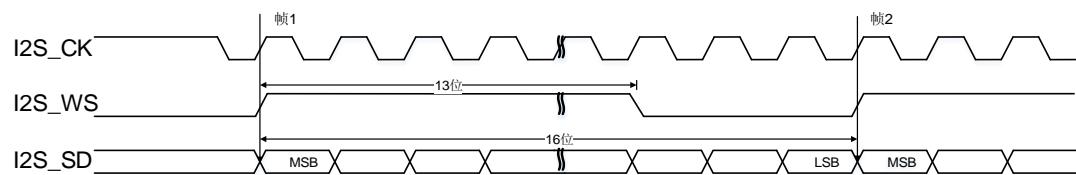


图 19-45. PCM 标准长帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=0)

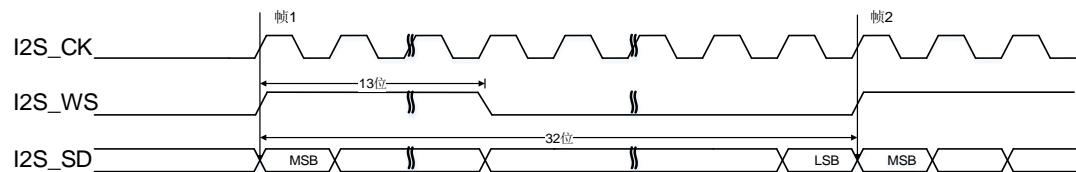


图 19-46. PCM 标准长帧同步模式时序图 (DTLEN=10, CHLEN=1, CKPL=1)

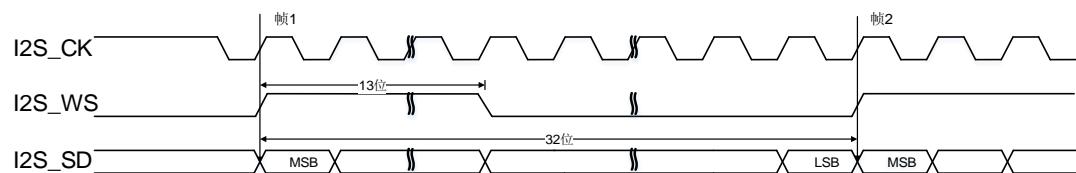


图 19-47. PCM 标准长帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=0)

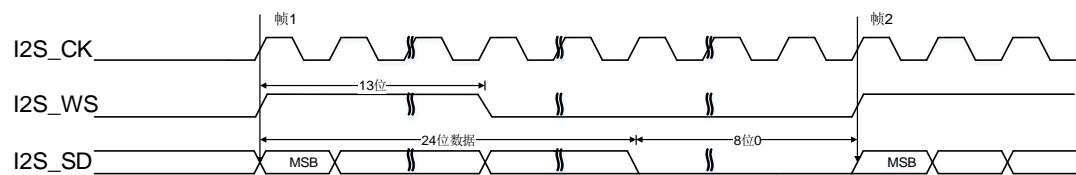


图 19-48. PCM 标准长帧同步模式时序图 (DTLEN=01, CHLEN=1, CKPL=1)

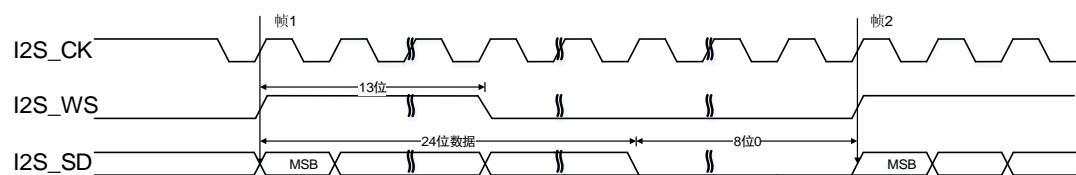


图 19-49. PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=0)

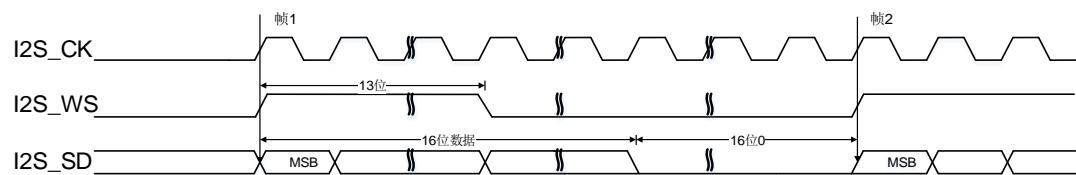
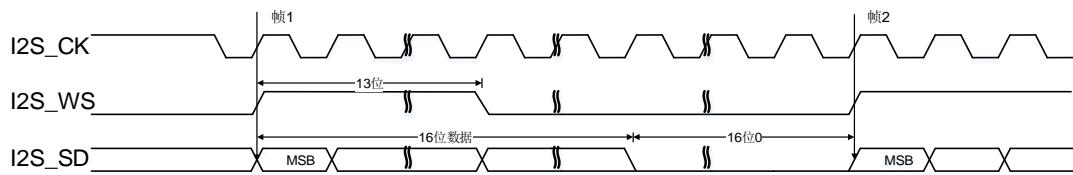
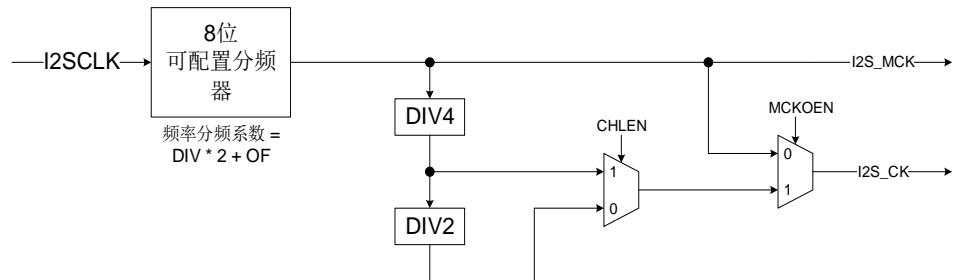


图 19-50. PCM 标准长帧同步模式时序图 (DTLEN=00, CHLEN=1, CKPL=1)



19.9.2. I2S 时钟

图 19-51. I2S 时钟生成结构框图



I2S 时钟生成器框图如 [图 19-51. 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。时钟源是系统时钟 (CK_SYS)。I2S 比特率可以通过 [表 19-5. I2S 比特率计算公式](#) 所示的公式计算。

表 19-5. 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 19-6. 音频采样频率计算公式](#) 所列的公式进行配置。

表 19-6. 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	$I2SCLK / (256 * (DIV * 2 + OF))$

19.9.3. 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 19-7. 各种运行模式下 I2S 接口信号的方向](#) 所示。

表 19-7. 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

1. NU表示该引脚没有被I2S使用，可以用于其他功能。

I2S 初始化流程

I2S初始化过程包括以下五个步骤。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，I2SSSTD[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RXORERR标志位会被置1，如果SPI_CTL1寄存器的ERRIE位为1，将会产生中断。这种情况下，必须先关闭I2S再打开I2S，然后再恢复通讯。

对于除PCM之外的所有标准来说，I2SCH标志用来区分当前传输数据所属的通道。I2SCH标志在每次RBNE标志由0变1时更新。

为了关闭I2S，不同的音频标准，数据长度和通道长度采用不同的操作步骤。每种情况的操作步骤如下所示。

- 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。

19.9.4. 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。

19.10. I2S 中断

19.10.1. 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。

19.10.2. 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。

■ 帧格式错误（FERR）：

在从I2S模式下，I2S模块监视I2S_WS信号，如果I2S_WS信号在一个错误的位置发生翻转，将会置位FERR帧错误标志位。

表 19-8. I2S 中断总结了 I2S 中断事件和相应的使能位。

表 19-8. I2S 中断

中断标志	描述	清除方式	中断使能位
TBE	发送缓冲区空	写 SPI_DATA 寄存器	TBEIE
RBNE	接收缓冲区非空	读 SPI_DATA 寄存器	RBNEIE
TXURERR	发送欠载错误	读 SPI_STAT 寄存器	ERRIE
RXORERR	接收过载错误	读 SPI_DATA 寄存器，然后再读 SPI_STAT 寄存器	
FERR	I2S 帧错误	读 SPI_STAT 寄存器	

19.11. SPI/I2S 寄存器

SPI0 基地址: 0x4001 3000

SPI1/I2S1 基地址: 0x4000 3800

SPI2/I2S2 基地址: 0x4000 3C00

19.11.1. 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	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15	BDEN	双向数据模式使能 0: 2线单向传输模式 1: 1线双向传输模式。数据在主机的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 (TCRC) 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

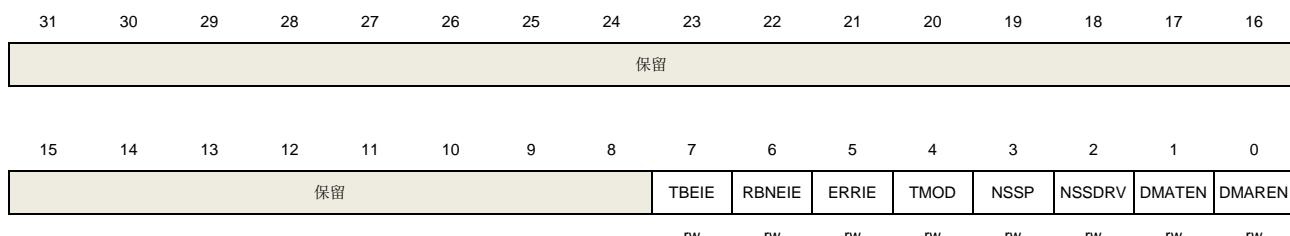
		0: 8位数据帧格式 1: 16位数据帧格式
10	RO	只接收模式 当BDEN清零时，该位决定了数据的传输方向。 0: 全双工模式 1: 只接收模式
9	SWNSSEN	NSS软件模式使能 0: NSS硬件模式， NSS电平取决于NSS引脚 1: NSS软件模式， NSS电平取决于SWNSS位 该位在SPI TI模式下没有意义。
8	SWNSS	NSS软件模式下NSS引脚选择 0: NSS引脚拉低 1: NSS引脚拉高 只有在SWNSSEN置位时，该位有效。 该位在SPI TI模式下没有意义。
7	LF	最低有效位先发模式 0: 先发送最高有效位 1: 先发送最低有效位 该位在SPI TI模式下没有意义。
6	SPIEN	SPI使能 0: SPI设备禁止 1: SPI设备使能
5:3	PSC[2:0]	主时钟预分频选择 000: PCLK/2 100: 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和SPI2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

19.11.2. 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。



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。
---	---------------	--

19.11.3. 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	
rc_w0						r	r	r	r	rc_w0	r	r	r	r	r

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： 0：没有TI模式帧错误发生 1：TI模式帧错误发生</p> <p>I2S模式： 0：没有I2S帧错误发生 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 1：SPI或I2S当前正在发送且/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 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 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</p>

模式时SWNSS位为0，都会产生CONFERR错误）

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。

I2S模式下不使用该位。

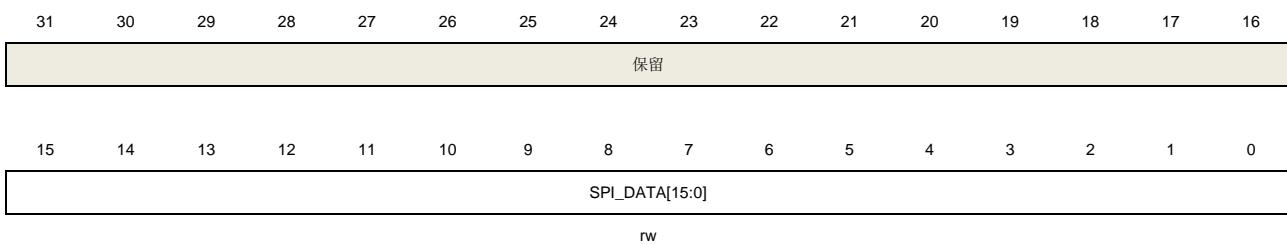
4	CRCERR	SPI CRC错误标志 0: SPI_RCRC值等于最后接收到的CRC值 1: SPI_RCRC值不等于最后接收到的CRC值 该位由硬件置位，可以通过写0清除。 I2S模式下不使用该位。
3	TXURERR	发送欠载错误标志 0: 无发送欠载错误发生 1: 发送欠载错误发生 该位由硬件置位，通过写SPI_STAT寄存器清除。 SPI模式下不使用该位。
2	I2SCH	I2S通道标志 0: 下一个将要发送或接收的数据属于左通道 1: 下一个要发送或接收的数据属于右通道 该位由硬件置位和清除。 SPI模式下该位无用，I2S PCM模式下该位没有意义。
1	TBE	发送缓冲区空 0: 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

19.11.4. 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。



位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	SPI_DATA[15:0]	数据传输寄存器值

硬件有两个缓冲区：发送缓冲区和接收缓冲区。向SPI_DATA写数据将会把数据存入发送缓冲区，从SPI_DATA读数据，将从接收缓冲区获得数据。

当数据帧格式为8位时，SPI_DATA[15:8]强制为0，SPI_DATA[7:0]用来发送和接收数据，发送和接收缓冲区都是8位。如果数据帧格式为16位，SPI_DATA[15:0]用于发送和接收数据，发送和接收缓冲区也是16位。

19.11.5. 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
CRCPOLY[15:0]															

rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CRCPOLY[15:0]	CRC多项式寄存器值 该值包含了CRC多项式，用于CRC计算，默认值为0007h。

19.11.6. 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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCRC[15:0]															

r

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	RCRC[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RCRC寄存器中。

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RCRC[7:0]。

如果是16位数据帧格式，CRC计算基于CRC16标准进行，保存数据到RCRC[15:0]。

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位或RCU复位寄存器中的SPIxRST位置位时，该寄存器复位。

19.11.7. 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
TCRC[15:0]															

r

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	TCRC[15:0]	发送CRC寄存器值 当SPI_CTL0中的CRCEN置位时，硬件计算发送数据的CRC值，并保存到TCRC寄存器中。如果是8位数据帧格式，CRC计算基于CRC8标准进行，保存数据到TCRC[7:0]。如果是16位数据帧格式，CRC计算基于CRC16标准进行，保存数据到TCRC[15:0]。硬件在发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。当SPI_CTL0寄存器中的CRCEN位或RCU复位寄存器中的SPIxRST位置位时，该寄存器复位。

19.11.8. 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
保留 I2SSEL I2SEN I2SOPMOD[1:0] PCMS MOD 保留 I2SSTD[1:0] CKPL DTLEN[1:0] CHLEN															

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	I2S模式选择 0: SPI模式 1: I2S模式 当SPI或I2S关闭时配置该位。
10	I2SEN	I2S使能 0: I2S禁止 1: I2S使能 SPI模式不使用该位。
9:8	I2SOPMOD[1:0]	I2S运行模式 00: 从机发送模式 01: 从机接收模式 10: 主机发送模式 11: 主机接收模式 当I2S关闭时配置该位。SPI模式不使用该位。
7	PCMSMOD	PCM帧同步模式 0: 短帧同步 1: 长帧同步 只有在PCM标准下，该位才有意义。 当I2S关闭时配置该位。SPI模式不使用该位。
6	保留	必须保持复位值
5:4	I2SSTD[1:0]	I2S标准选择 00: I2S飞利浦标准 01: MSB对齐标准 10: LSB对齐标准 11: PCM标准 当I2S关闭时配置该位。SPI模式不使用该位。
3	CKPL	空闲状态时钟极性 0: I2S_CK空闲状态为低电平 1: I2S_CK空闲状态为高电平 当I2S关闭时配置该位。SPI模式不使用该位。
2:1	DTLEN[1:0]	数据长度 00: 16位 01: 24位 10: 32位 11: 保留

当I2S关闭时配置该位。SPI模式不使用该位。

0	CHLEN	通道长度 0: 16位 1: 32位 通道长度必须大于或等于数据长度。 当I2S关闭时配置该位。SPI模式不使用该位。
---	-------	---

19.11.9. 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]								rw rw rw

位/位域	名称	描述
31:10	保留	必须保持复位值
9	MCKOEN	I2S_MCK输出使能 0: I2S_MCK输出禁止 1: I2S_MCK输出使能 当I2S关闭时配置该位。 SPI模式不使用该位。
8	OF	预分频器的奇系数 0: 实际分频系数为DIV * 2 1: 实际分频系数为DIV * 2 + 1 当I2S关闭时配置该位。SPI模式下不使用该位。
7:0	DIV[7:0]	预分频器的分频系数 实际分频系数是DIV * 2 + OF。 DIV不能为0。 当I2S关闭时配置该位。SPI模式下不使用该位。

19.11.10. SPI0 四路 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_DRV | QRD | QMOD

rw rw rw

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: 单路模式下IO2和IO3输出关闭 1: 单路模式下IO2和IO3输出高电平 该位仅适用于SPI0。
1	QRD	四路SPI模式读选择 0: SPI四路模式写操作 1: SPI四路模式读操作 该位仅能在SPI未通信时配置（TRANS位清零）。 该位仅适用于SPI0。
0	QMOD	四路SPI模式使能 0: SPI工作在单路模式 1: SPI工作在四路模式 该位仅能在SPI未通信时配置（TRANS位清零）。 该位仅适用于SPI0。

20. 外部存储器控制器（EXMC）

20.1. 简介

外部存储器控制器EXMC，用来访问各种片外存储器。通过配置寄存器，EXMC可以把AMBA协议转换为专用的片外存储器通信协议，包括SRAM，PSRAM，ROM和NOR Flash。用户还可以调整配置寄存器中的时间参数来提高通信效率。

20.2. 主要特征

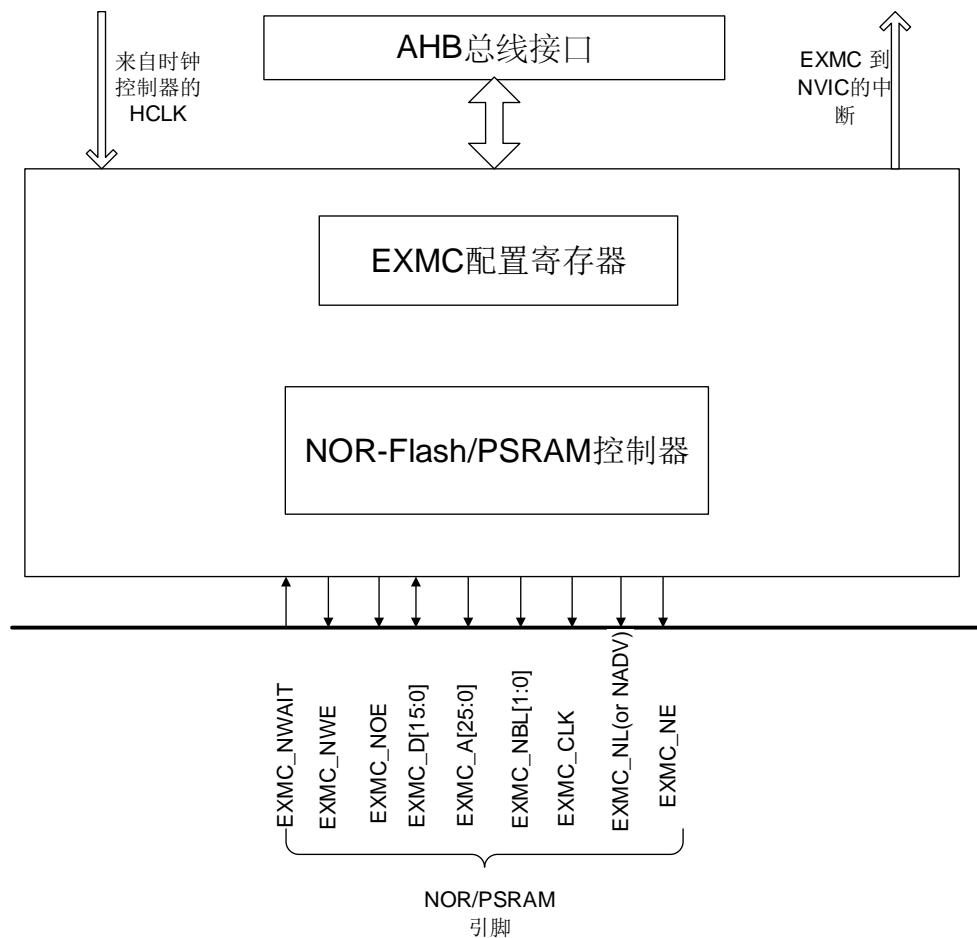
- 支持片外存储器类型：
 - SRAM
 - PSRAM
 - ROM
 - NOR Flash
- AMBA协议与各种片外存储器协议转换；
- 时序参数可编程可以满足用户特定需求；
- 对于部分存储器类型支持独立的读写时序；
- 支持8位，或16位总线带宽；
- NOR Flash和PSRAM支持地址总线和数据总线的复用；
- 提供写使能和字节选择信号；
- 当AMBA总线宽度与外部存储器数据宽度不同时，会自动分割操作。

20.3. 功能说明

20.3.1. 结构框图

EXMC由4个模块组成：AHB总线接口，EXMC配置寄存器，NOR/PSRAM控制器和外部设备接口。AHB时钟（HCLK）是参考时钟。

图 20-1. EXMC 结构框图

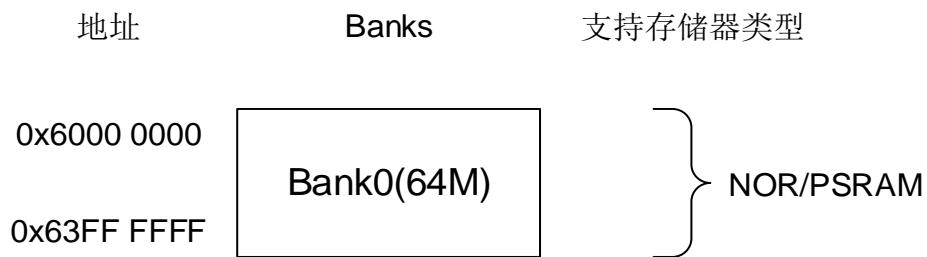


20.3.2. EXMC 访问基本规范

EXMC是AHB总线至外部设备协议的转换接口。32位的AHB读写操作可以转化为几个连续的8位或16位读写操作。在数据传输的过程中，AHB数据宽度和存储器数据宽度可能不相同。为了保证数据传输的一致性，EXMC读写访问需要遵从以下规范。

- AHB访问宽度等于存储器宽度，则没有问题。
- AHB访问宽度大于存储器宽度，则自动将AHB访问分割成几个连续的存储器数据宽度的传输。
- AHB访问宽度小于存储器宽度，如果外部存储设备具有字节选择功能，如SRAM、ROM、PSRAM，则可通过它的字节通道EXMC_NBL[1:0]来访问对应的字节。否则禁止写操作，只允许读操作。

图 20-2. EXMC Bank 划分



EXMC访问区域为Bank0，占64M字节，用于访问NOR、PSRAM设备。

由于HADDR[25:0]是字节地址，而外部存储器访问有可能不是按字节访问的，所以会出现地址不一致的情况，但EXMC能实现对HADDR的调整以适应外部存储器的数据宽度。具体规则如下：

- 如果外部存储器的数据宽度是8位按字节对齐，HADDR[25:0]与EXMC_A[25:0]相连，然后用EXMC_A[25:0]去连接外部存储器的地址线；
- 如果外部存储器的数据宽度是16位按半字对齐，就需要将HADDR的字节地址转化为半字地址之后再连接外存储器，所以需要将HADDR[25:1]与EXMC_A[24:0]相连。然后用EXMC_A[24:0]去连接外部存储器的地址线。

20.3.3. NOR/PSRAM 控制器

控制器控制Bank0，它可以支持NOR Flash、PSRAM、SRAM、ROM和蜂窝RAM外部存储器。

注意：

在异步模式下，所有控制器输出信号在内部AHB总线时钟（HCLK）的上升沿改变。

在同步模式下，所有控制器输出数据在外部存储器时钟（EXMC_CLK）的下降沿改变。

NOR/PSRAM 接口描述

表 20-1. NOR Flash 接口信号描述

EXMC 引脚	传输方向	模式	功能描述
EXMC_CLK	输出	同步	同步时钟信号
非复用 EXMC_A[25:0]	输出	异步/同步	地址总线信号
复用 EXMC_A[25:16]			
EXMC_D[15:0]	输入/输出	异步/同步 (复用)	地址/数据总线
	输入/输出	异步/同步 (非复用)	数据总线
EXMC_NE	输出	异步/同步	片选
EXMC_NOE	输出	异步/同步	输出使能（读使能）
EXMC_NWE	输出	异步/同步	写使能
EXMC_NWAIT	输入	异步/同步	等待输入信号
EXMC_NL (NADV)	输出	异步/同步	地址有效

表 20-2. PSRAM 非复用接口信号描述

EXMC 引脚	传输方向	模式	功能描述
EXMC_CLK	输出	同步	同步时钟信号
EXMC_A[25:0]	输出	异步/同步	地址总线
EXMC_D[15:0]	输入/输出	异步/同步	数据总线
EXMC_NE	输出	异步/同步	片选
EXMC_NOE	输出	异步/同步	输出使能（读使能）
EXMC_NWE	输出	异步/同步	写使能
EXMC_NWAIT	输入	异步/同步	等待输入信号
EXMC_NL (NADV)	输出	异步/同步	锁存使能（地址有效使能, NADV）
EXMC_NBL[1]	输出	异步/同步	高字节使能
EXMC_NBL[0]	输出	异步/同步	低字节使能

支持的存储器访问模式

下表列出了当NOR, PSRAM和SRAM存储器数据总线为16位时，所支持的设备类型、访问模式和传输的示例。

表 20-3. EXMC 的 Bank0 支持的所有传输

存储器类型	访问模式	读/写	AHB 传输宽度	存储器传输宽度	注释
NOR Flash	异步	R	8	16	
	异步	R	16	16	
	异步	W	16	16	
	异步	R	32	16	分成 2 次 EXMC 访问
	异步	W	32	16	分成 2 次 EXMC 访问
	同步	R	16	16	
	同步	R	32	16	
PSRAM	异步	R	8	16	
	异步	W	8	16	使用字节信号 EXMC_NBL[1:0]
	异步	R	16	16	
	异步	W	16	16	
	异步	R	32	16	分成 2 次 EXMC 访问
	异步	W	32	16	分成 2 次 EXMC 访问
	同步	R	16	16	
	同步	R	32	16	
	同步	W	8	16	使用字节信号 EXMC_NBL[1:0]
	同步	W	16	16	
SRAM 和	同步	W	32	16	分成 2 次 EXMC 访问
	异步	R	8	8	

存储器类型	访问模式	读/写	AHB 传输宽度	存储器传输宽度	注释
ROM	异步	R	8	16	
	异步	R	16	8	分成 2 次 EXMC 访问
	异步	R	16	16	
	异步	R	32	8	分成 4 次 EXMC 访问
	异步	R	32	16	分成 2 次 EXMC 访问
	异步	W	8	8	
	异步	W	8	16	使用字节信号 EXMC_NBL[1:0]
	异步	W	16	8	
	异步	W	16	16	
	异步	W	32	8	
	异步	W	32	16	

NOR Flash/PSRAM 控制时序

EXMC为SRAM、ROM、PSRAM、NOR Flash等外部静态存储器提供可编程的时序参数以及多种时序模型以满足不同的需求。

表 20-4. NOR/PSRAM 控制时序参数

参数	功能	访问模式	单位	最小值	最大值
CKDIV	同步时钟分频比	同步	HCLK	2	16
DLAT	数据延迟	异步	EXMC_CLK	2	17
BUSLAT	总线延迟	异步/同步读	HCLK	1	16
DSET	数据建立时间	异步	HCLK	2	256
AHLD	地址保持时间	异步(复用)	HCLK	2	16
ASET	地址建立时间	异步	HCLK	1	16

表 20-5. EXMC 时序模型

时序模型	扩展模式	模式描述		写时序参数	读时序参数
异步	模式 1	0	SRAM/PSRAM/CRAM		DSET ASET
	模式 2	0	NOR Flash		DSET ASET
	模式 A	1	SRAM/PSRAM/CRAM 在数据阶段时 EXMC_NOE 翻转		WDSET WASET
	模式 B	1	NOR Flash		WDSET WASET
	模式 C	1	NOR Flash 在数据阶段时 EXMC_NOE 翻转		WDSET WASET
	模式 D	1	有地址保持功能		WDSET WAHLD WASET

时序模型		扩展模式	模式描述	写时序参数	读时序参数
	模式 AM	0	NOR Flash 数据/地址复用	DSET AHLD ASET BUSLAT	DSET AHLD ASET BUSLAT
同步	模式 E	0	NOR/PSRAM/CRAM 同步读, PSRAM/CRAM 同步写	DLAT CKDIV	DLAT CKDIV
	模式 SM	0	NOR Flash 数据/地址复用	DLAT CKDIV	DLAT CKDIV

如[表20-5. EXMC时序模型](#)所示，EXMC模块NOR Flash/PSRAM控制器可以提供多种时序模型。用户可以通过修改[表20-4. NOR/PSRAM控制时序参数](#)中列出的参数来使之适合不同类型外部存储器的时序以及满足用户的要求。当将寄存器EXMC_SNCTL位EXMODEN置1使能扩展模式后，可以通过寄存器EXMC_SNTCFG和EXMC_SNWTCFG将读写配置成独立的时序。

异步访问时序

模式1 - SRAM/CRAM

图 20-3. 模式 1 读访问

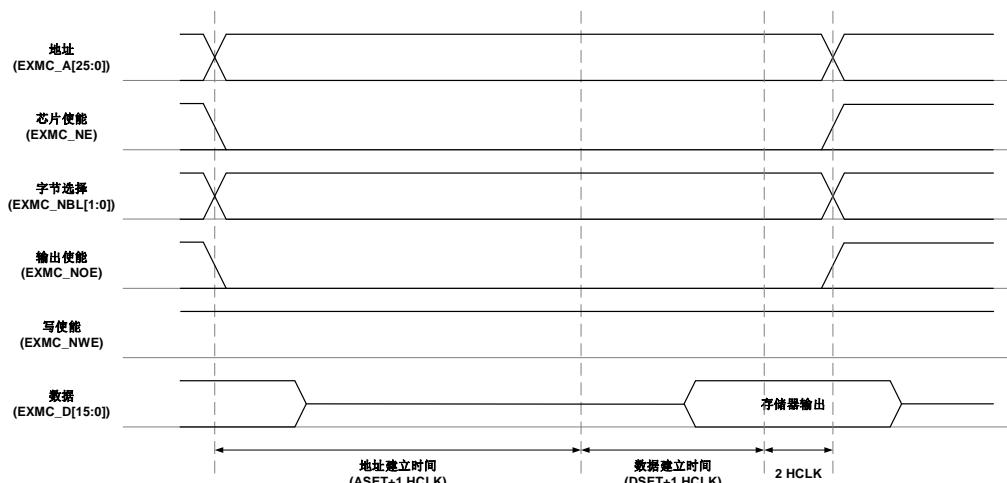


图 20-4. 模式 1 写访问

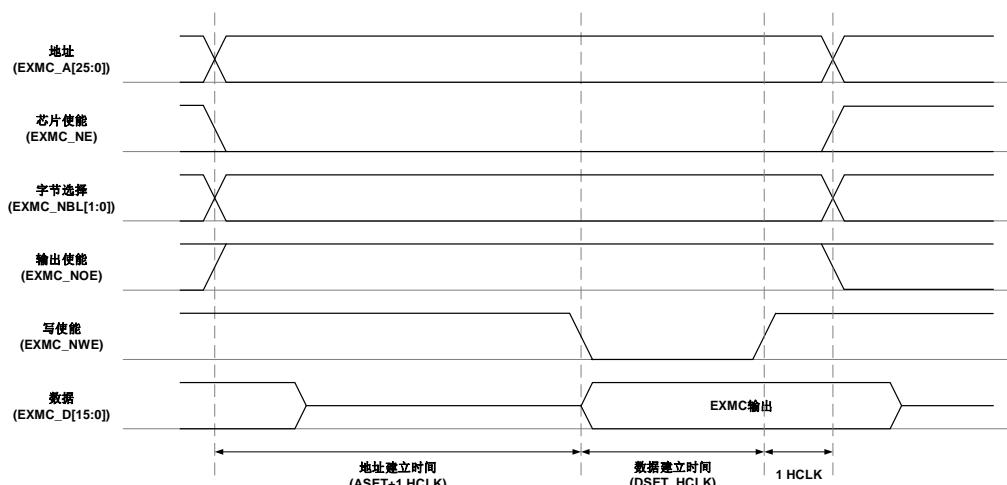


表 20-6. 模式 1 相关寄存器配置

位/位域	位名	参考设定值
EXMC_SNCTL		
31-20	保留	0x000
19	SYNCWR	0x0
18-16	CPS	0x0
15	ASYNCWAIT	取决于存储器
14	EXMODEN	0x0
13	NRWTEN	0x0
12	WREN	取决于用户
11	NRWTCFG	无影响
10	WRAPEN	0x0
9	NRWTPOL	仅当位 15 为 1 时有效
8	SBRSTEN	0x0
7	保留	0x1
6	NREN	无影响
5-4	NRW	取决于存储器
3-2	NRTP	取决于存储器，除了 0x2 (NOR Flash)
1	NRMUX	0x0
0	NRBKEN	0x1
EXMC_SNTCFG		
31-30	保留	0x0
29-28	ASYNCFMOD	无影响
27-24	DLAT	无影响
23-20	CKDIV	无影响
19-16	BUSLAT	从 EXMC_NE 上升沿到下降沿的时间
15-8	DSET	取决于存储器与用户 (写操作为 DSET+1 HCLK 时钟周期, 读操作为 DSET+3 HCLK 时钟周期)
7-4	AHLD	无影响
3-0	ASET	取决于存储器与用户

模式A - SRAM/PSRAM (CRAM) OE翻转

图 20-5. 模式 A 读访问

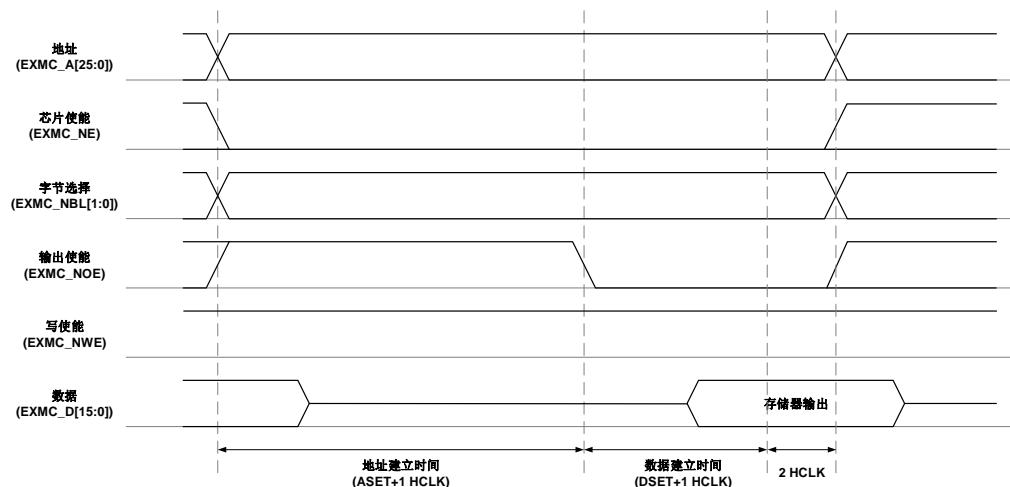
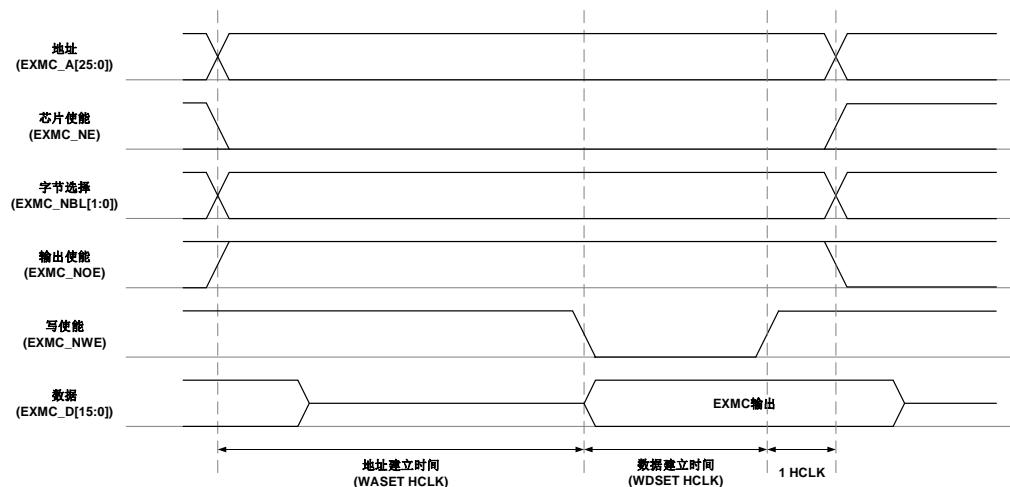


图 20-6. 模式 A 写访问



模式A和模式1写时序的区别在于，当读和写访问具有相同的时序配置时，模式A的写时序是独立于读时序的。

表 20-7. 模式 A 相关寄存器配置

位/位域	位名	参考设定值
EXMC_SNCTL		
31-20	保留	0x000
19	SYNCWR	0x0
18-16	CPS	0x0
15	ASYNCWTEN	取决于存储器
14	EXMODEN	0x1
13	NRWTEN	0x0
12	WREN	取决于用户
11	NRWTCFG	无影响
10	WRAPEN	0x0
9	NRWTPOL	仅当位 15 为 1 时有效
8	SBRSTEN	0x0

位/位域	位名	参考设定值
7	保留	0x1
6	NREN	无影响
5-4	NRW	取决于存储器
3-2	NRTP	取决于存储器, 除了 0x2 (NOR Flash)
1	NRMUX	0x0
0	NRBKEN	0x1
EXMC_SNTCFG (读)		
31-30	保留	0x0
29-28	ASYNCMOD	0x0
27-24	DLAT	无影响
23-20	CKDIV	无影响
19-16	BUSLAT	从 EXMC_NE 上升沿到下降沿的时间
15-8	DSET	取决于存储器与用户 (读操作为 DSET+3 HCLK 时钟周期)
7-4	AHLD	无影响
3-0	ASET	取决于存储器与用户
EXMC_SNWT CFG (写)		
31-30	保留	0x0
29-28	WASYNCMOD	0x0
27-20	保留	0xFF
19-16	WBUSLAT	从 EXMC_NE 上升沿到下降沿的时间
15-8	WDSET	取决于存储器与用户 (写操作为 WDSET+1 HCLK 时钟周期)
7-4	WAHLD	0x0
3-0	WASET	取决于存储器与用户

模式2/B - NOR Flash

图 20-7. 模式 2/B 读访问

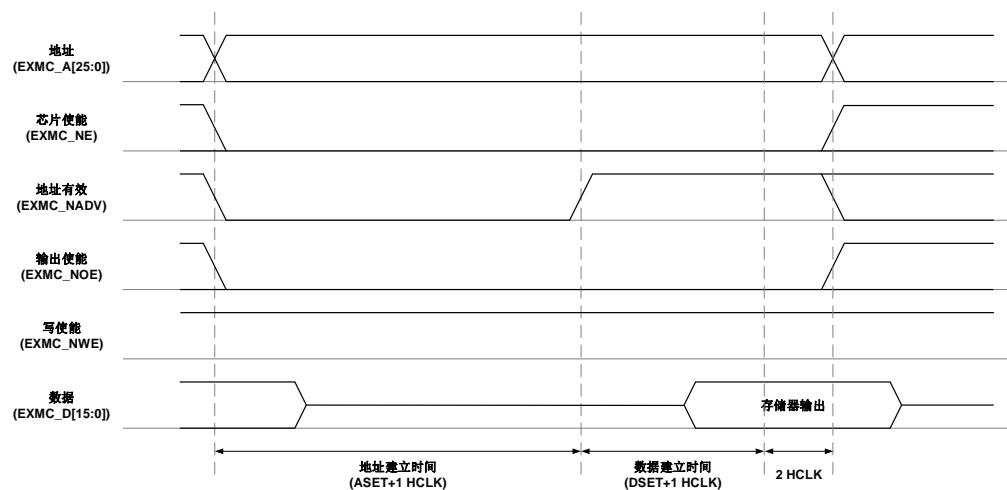


图 20-8. 模式 2 写访问

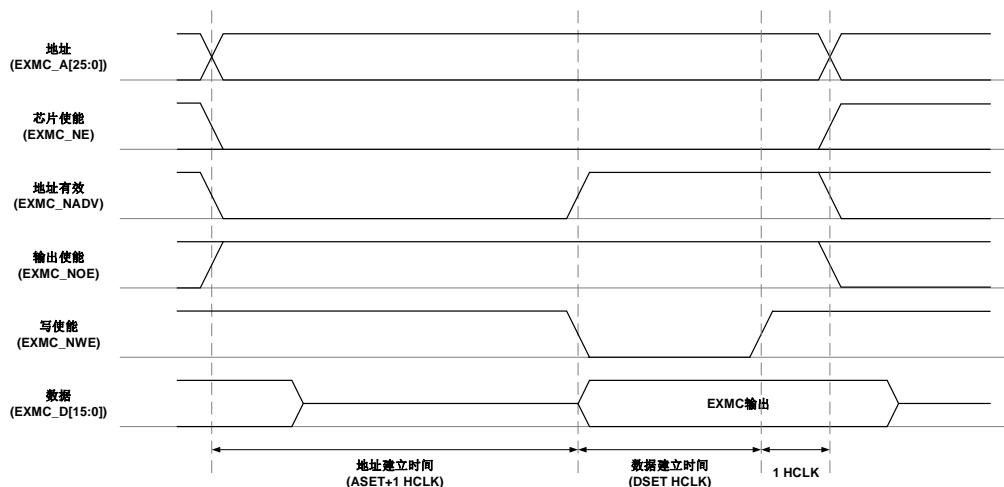


图 20-9. 模式 B 写访问

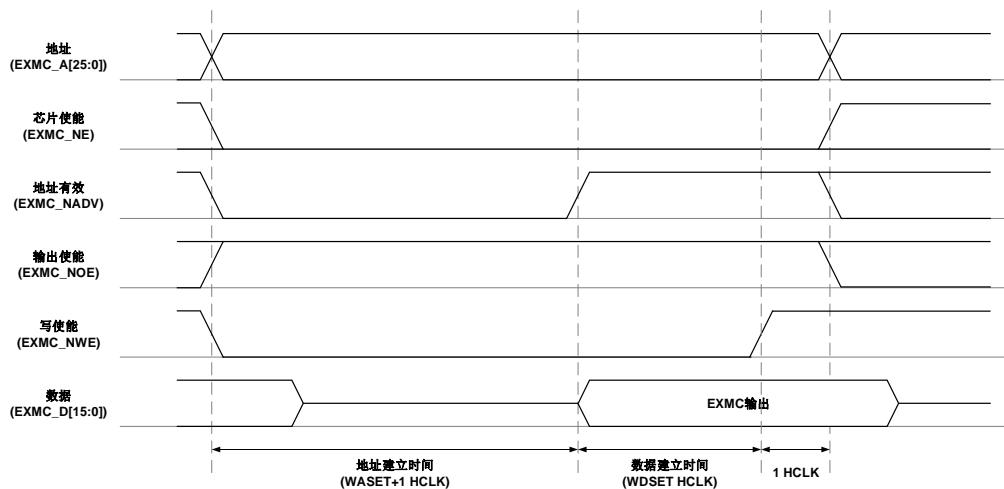


表 20-8. 模式 2/B 相关寄存器配置

位/位域	位名	参考设定值
EXMC_SNCTL (模式 2, 模式 B)		
31-20	保留	0x000
19	SYNCWR	0x0
18-16	CPS	0x0
15	ASYNCWTEN	取决于存储器
14	EXMODEN	模式 2: 0x0, 模式 B: 0x1
13	NRWTEN	0x0
12	WREN	取决于用户
11	NRWTCFG	无影响
10	WRAPEN	0x0
9	NRWTPOL	仅当位 15 为 1 时有效
8	SBRSTEN	0x0
7	保留	0x1
6	NREN	0x1

位/位域	位名	参考设定值
5-4	NRW	取决于存储器
3-2	NRTP	NOR Flash: 0x2
1	NRMUX	0x0
0	NRBKEN	0x1
EXMC_SNTCFG (模式 2 读/写操作, 模式 B 读操作)		
31-30	保留	0x0
29-28	ASYNCFMOD	模式 B: 0x1
27-24	DLAT	无影响
23-20	CKDIV	无影响
19-16	BUSLAT	从 EXMC_NE 上升沿到下降沿的时间
15-8	DSET	取决于存储器与用户 (读操作为 DSET+3 HCLK 时钟周期)
7-4	AHLD	0x0
3-0	ASET	取决于存储器与用户
EXMC_SNWTCFG (模式 B 写操作)		
31-30	保留	0x0
29-28	WASYNCFMOD	模式 B: 0x1
27-20	保留	0xFF
19-16	WBUSLAT	从 EXMC_NE 上升沿到下降沿的时间
15-8	WDSET	取决于存储器与用户 (写操作为 WDSET+1 HCLK 时钟周期)
7-4	WAHLD	0x0
3-0	WASET	取决于存储器与用户

模式C - NOR Flash OE翻转

图 20-10. 模式 C 读访问

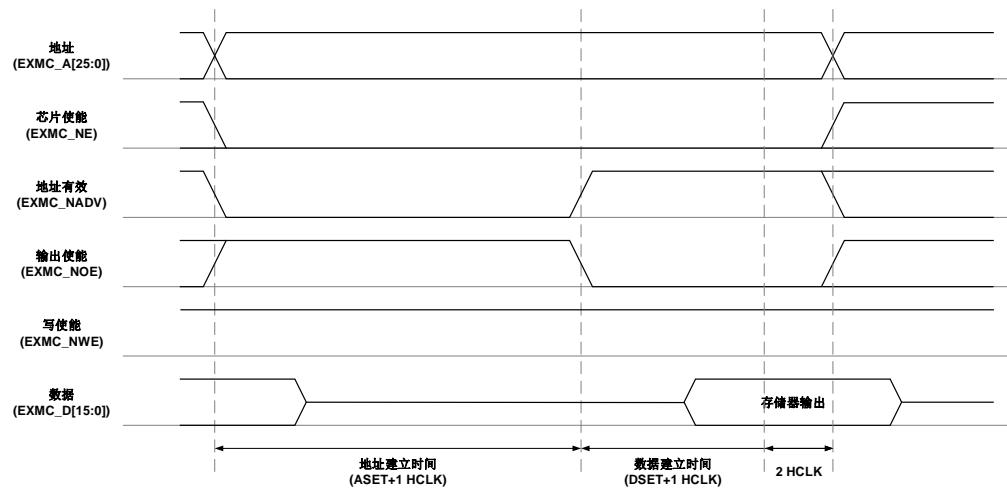
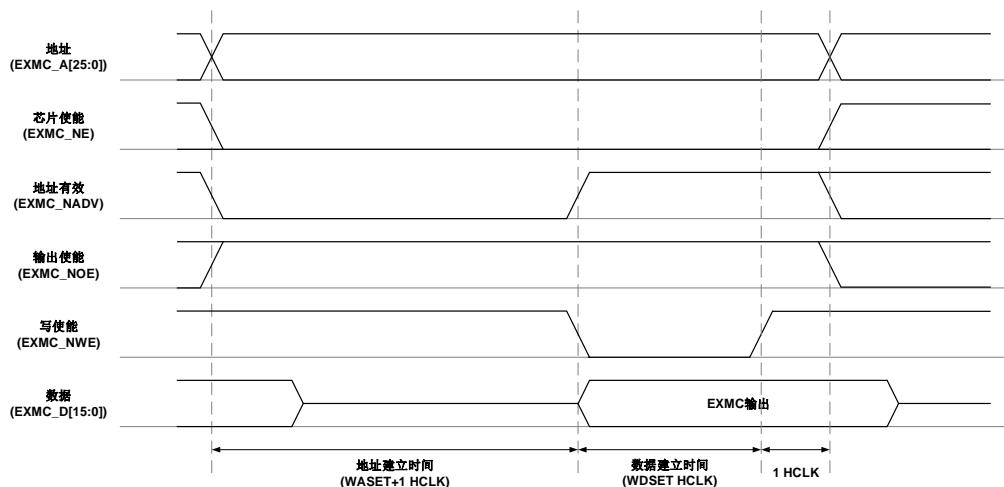


图 20-11. 模式 C 写访问



模式C和模式1写时序的区别在于，当读和写访问具有相同的时序配置时，模式C的写时序是独立于读时序的。

表 20-9. 模式 C 相关寄存器配置

位/位域	位名	参考设定值
EXMC_SNCTL		
31-20	保留	0x000
19	SYNCWR	0x0
18-16	CPS	0x0
15	ASYNCWTEN	取决于存储器
14	EXMODEN	0x1
13	NRWTEN	0x0
12	WREN	取决于用户
11	NRWTCFG	无影响
10	WRAPEN	0x0
9	NRWTPOL	仅当位 15 为 1 时有效
8	SBRSTEN	0x0
7	保留	0x1
6	NREN	0x1
5-4	NRW	取决于存储器
3-2	NRTP	NOR Flash: 0x2
1	NRMUX	0x0
0	NRBKEN	0x1
EXMC_SNTCFG		
31-30	保留	0x0
29-28	ASYNCFMOD	模式 C: 0x2
27-24	DLAT	0x0
23-20	CKDIV	0x0
19-16	BUSLAT	从 EXMC_NE 上升沿到下降沿的时间
15-8	DSET	取决于存储器与用户（读操作为 DSET+3 HCLK 时钟

位/位域	位名	参考设定值
		(周期)
7-4	AHLD	0x0
3-0	ASET	取决于存储器与用户
EXMC_SNWTCFG		
31-30	保留	0x0
29-28	WASYNCFMOD	模式 C: 0x2
27-20	保留	0xFF
19-16	WBUSLAT	从 EXMC_NE 上升沿到下降沿的时间
15-8	WDSET	取决于存储器与用户 (写操作为 WDSET+1 HCLK 时钟周期)
7-4	WAHLD	0x0
3-0	WASET	取决于存储器与用户

模式D -带地址扩展的异步操作

图 20-12. 模式 D 读访问

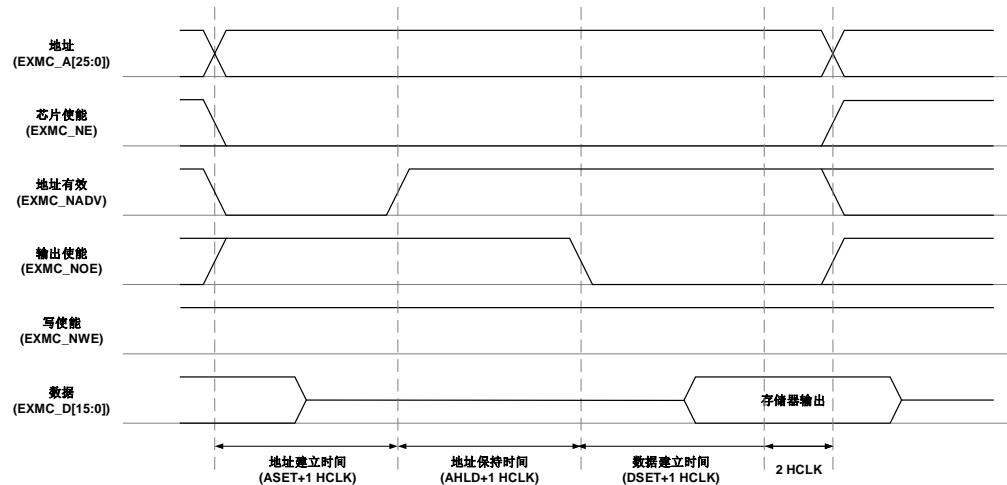


图 20-13. 模式 D 写访问

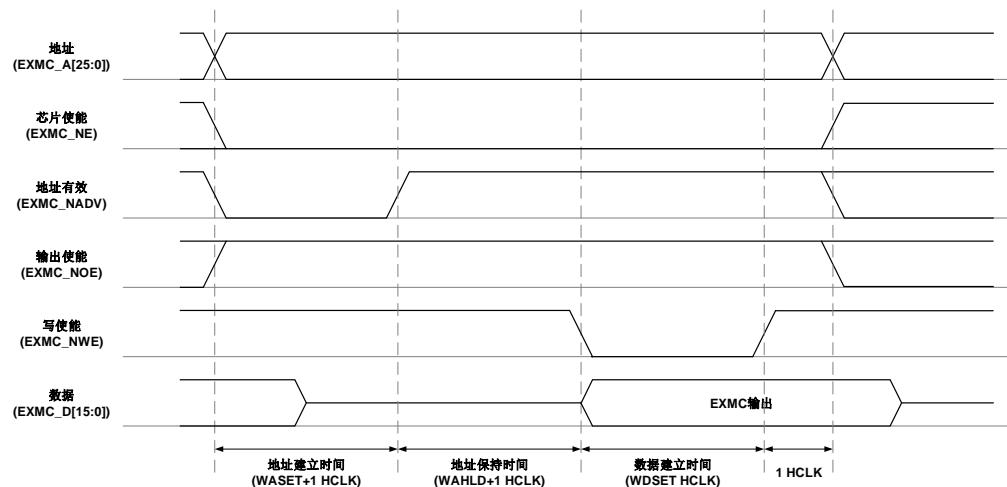


表 20-10. 模式 D 相关寄存器配置

位/位域	位名	参考设定值
EXMC_SNCTL		
31-20	保留	0x000
19	SYNCWR	0x0
18-16	CPS	0x0
15	ASYNCWTEN	取决于存储器
14	EXMODEN	0x1
13	NRWTEN	0x0
12	WREN	取决于用户
11	NRWTCFG	无影响
10	WRAPEN	0x0
9	NRWTPOL	仅当位 15 为 1 时有效
8	SBRSTEN	0x0
7	保留	0x1
6	NREN	取决于存储器
5-4	NRW	取决于存储器
3-2	NRTP	取决于存储器
1	NRMUX	0x0
0	NRBKEN	0x1
EXMC_SNTCFG		
31-30	保留	0x0
29-28	ASYNCMOD	模式 D: 0x3
27-24	DLAT	无关
23-20	CKDIV	无影响
19-16	BUSLAT	从 EXMC_NE 上升沿到下降沿的时间
15-8	DSET	取决于存储器与用户 (读操作为 DSET+3 HCLK 时钟周期)
7-4	AHLD	取决于存储器与用户
3-0	ASET	取决于存储器与用户
EXMC_SNWTCFG		
31-30	保留	0x0
29-28	WASYNCMOD	模式 D: 0x3
27-20	保留	0xFF
19-16	WBUSLAT	从 EXMC_NE 上升沿到下降沿的时间
15-8	WDSET	取决于存储器与用户 (写操作为 WDSET+1 HCLK 时钟周期)
7-4	WAHLD	取决于存储器与用户
3-0	WASET	取决于存储器与用户

模式AM - NOR Flash地址/数据总线复用

图 20-14. 复用模式读访问

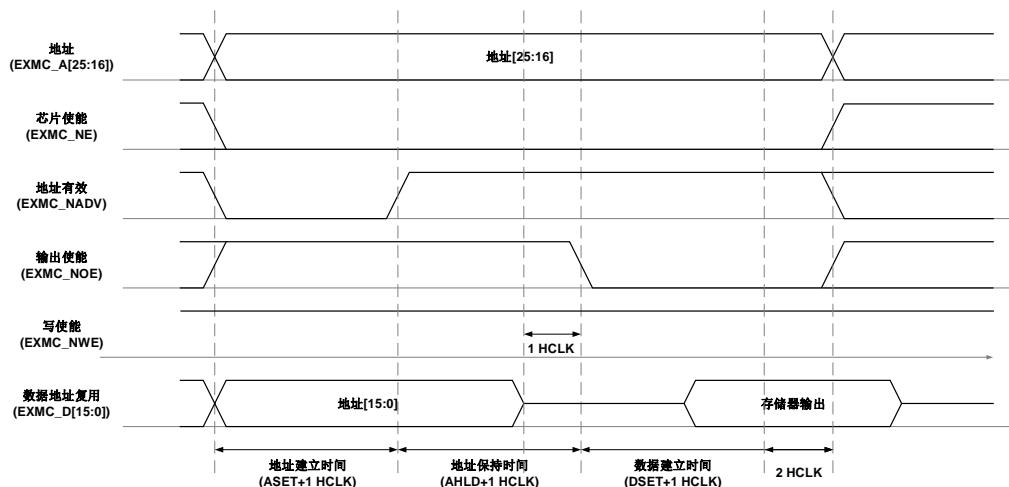


图 20-15. 复用模式写访问

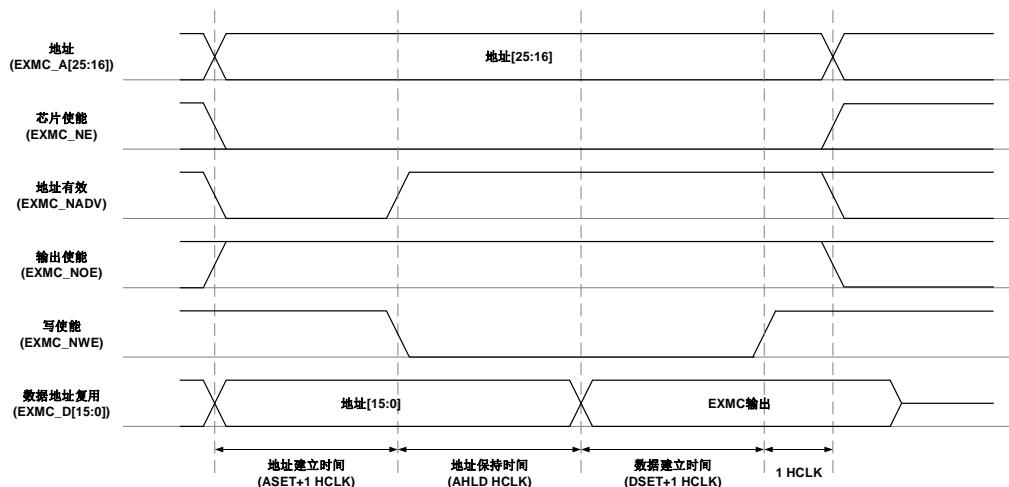


表 20-11. 复用模式相关寄存器配置

位/位域	位名	参考设定值
EXMC_SNCTL		
31-20	保留	0x000
19	SYNCWR	0x0
18-16	CPS	0x0
15	ASYNCWTEN	取决于存储器
14	EXMODEN	0x0
13	NRWTEN	0x0
12	WREN	取决于存储器
11	NRWTCFG	无影响
10	WRAPEN	0x0
9	NRWTPOL	仅当位 15 为 1 时有效
8	SBRSTEN	0x0
7	保留	0x1
6	NREN	0x1

位/位域	位名	参考设定值
5-4	NRW	取决于存储器
3-2	NRTP	0x2: NOR Flash
1	NRMUX	0x1
0	NRBKEN	0x1
EXMC_SNTCFG		
31-30	保留	0x0
29-28	ASYNCMOD	0x0
27-24	DLAT	无影响
23-20	CKDIV	无影响
19-16	BUSLAT	从 EXMC_NE 上升沿到下降沿的时间
15-8	DSET	取决于存储器与用户（写操作为 DSET+2 HCLK 时钟周期，读操作为 DSET+3 HCLK 时钟周期）
7-4	AHLD	取决于存储器与用户
3-0	ASET	取决于存储器与用户

异步通信的等待时序

等待功能由寄存器EXMC_SNCTL位ASYNCWAIT控制。在访问外部存储器期间，若使能异步等待功能(ASYNCWAIT=1)，数据建立时间将会根据EXMC_NWAIT的有效信号而自动延长。延长时间的计算如下：

若存储器等待信号与EXMC_NOE/ EXMC_NWE信号对齐：

$$T_{DATA_SETUP} \geq \max T_{WAIT_ASSERTION} + 4HCLK \quad (20-1)$$

若存储器等待信号与EXMC_NE信号对齐：

如果

$$\max T_{WAIT_ASSERTION} \geq T_{ADDRES_PHASE} + T_{HOLD_PHASE} \quad (20-2)$$

则

$$T_{DATA_SETUP} \geq (\max T_{WAIT_ASSERTION} - T_{ADDRES_PHASE} - T_{HOLD_PHASE}) + 4HCLK \quad (20-3)$$

否则

$$T_{DATA_SETUP} \geq 4HCLK \quad (20-4)$$

图 20-16. 异步等待有效时的读时序

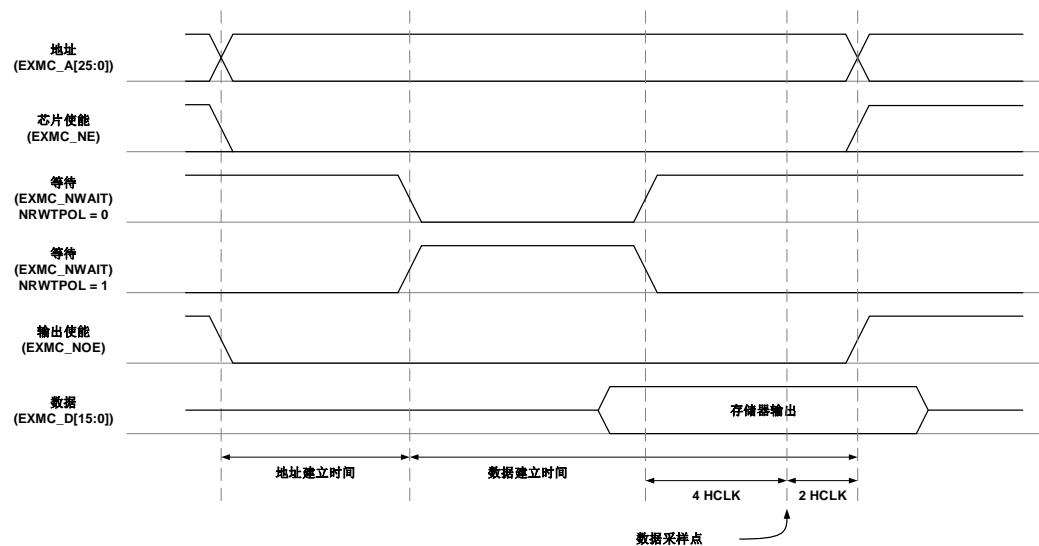
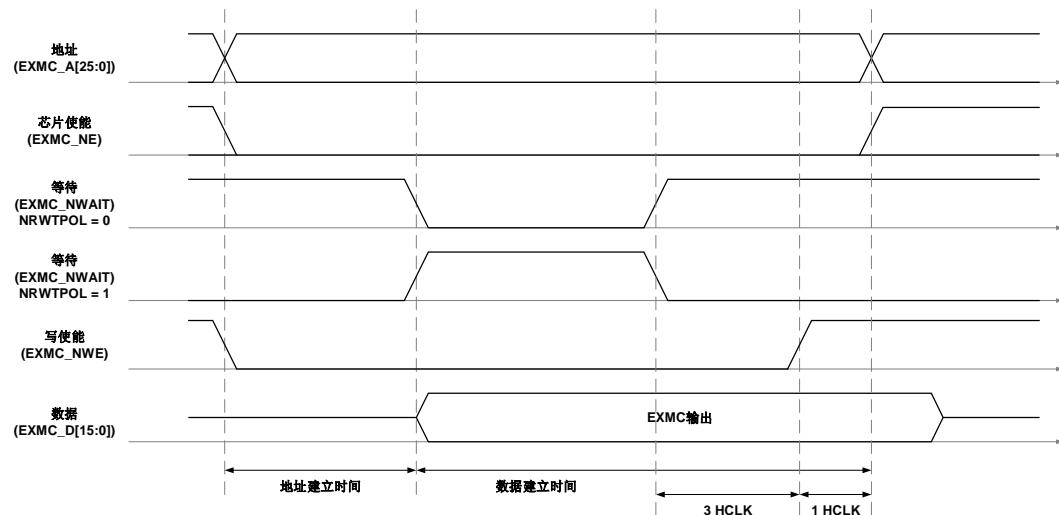


图 20-17. 异步等待有效时的写时序



同步访问时序

存储器时钟 (EXMC_CLK) 与系统时钟 (HCLK) 关系如下:

$$\text{EXMC_CLK} = \frac{\text{HCLK}}{\text{CKDIV}+1} \quad (20-5)$$

其中CKDIV是同步时钟分频比，通过配置寄存器EXMC_SNTCFG中的CKDIV位来设置不同的值。

1. 数据延迟与NOR Flash延迟

数据延迟DLAT是指在采样数据之前需要等待的EXMC_CLK周期数。它和NOR闪存延迟的关系如下所述。

NOR闪存延迟不包含EXMC_NADV，二者之间的关系为:

NOR 闪存延迟包含EXMC_NADV，二者之间的关系为：

$$\text{NOR 闪存延迟} = \text{DLAT} + 3 \quad (20-7)$$

2. 数据等待

用户需要保证EXMC_NWAIT信号与外部设备一致。该信号通过寄存器EXMC_SNCTL来设置，由位NRWTEN来使能，位NRWTCFG决定EXMC_NWAIT信号是等待状态同时有效，或者比等待状态提前一个时钟周期有效，位NRWTPOL设置EXMC_NWAIT信号极性。

在NOR Flash的同步突发模式中，当寄存器EXMC_SNCTL的位NRWTEN置为1，则在数据延迟之后会检测EXMC_NWAIT信号。如果检测到EXMC_NWAIT有效，就会插入等待时钟，直到EXMC_NWAIT变为无效。

■ EXMC_NWAIT有效极性：

NRWTPOL= 1, EXMC_NWAIT高电平有效

NRWTPOL= 0, EXMC_NWAIT低电平有效

■ 在同步突发模式中，EXMC_NWAIT信号有两种配置：

NRWTCFG = 1, EXMC_NWAIT信号有效时，当前时钟周期数据无效

NRWTCFG = 0, EXMC_NWAIT信号有效时，下一个时钟周期数据无效，这是复位后的默认配置。

在EXMC_NWAIT信号有效的等待周期内，EXMC会持续的给存储器发送时钟信号，保持片选和输出使能有效，并且忽视总线上的无效数据。

3. CRAM页边界突发传输的自动分组

CRAM1.5中禁止突发传输跨越页边界，EXMC遇到边界会进行传输的自动分组。为了保证正确的突发分组操作，用户需要在寄存器EXMC_SNCTL位CPS中需要设定CRAM的页大小。

4. 模式SM - 单次突发传输

对于同步突发传输，如果AHB需要的数据为16位，则EXMC会执行一次长度为1的成组传输；如果AHB需要的数据为32位，则EXMC会把这次传输分成2次16位的传输，即执行一次长度为2的突发传输。

对于其他的配置，请参考[表20-3. EXMC的Bank0支持的所有传输](#)。

同步复用突发读时序 – NOR, PSRAM (CRAM)

图 20-18. 同步复用突发传输读时序

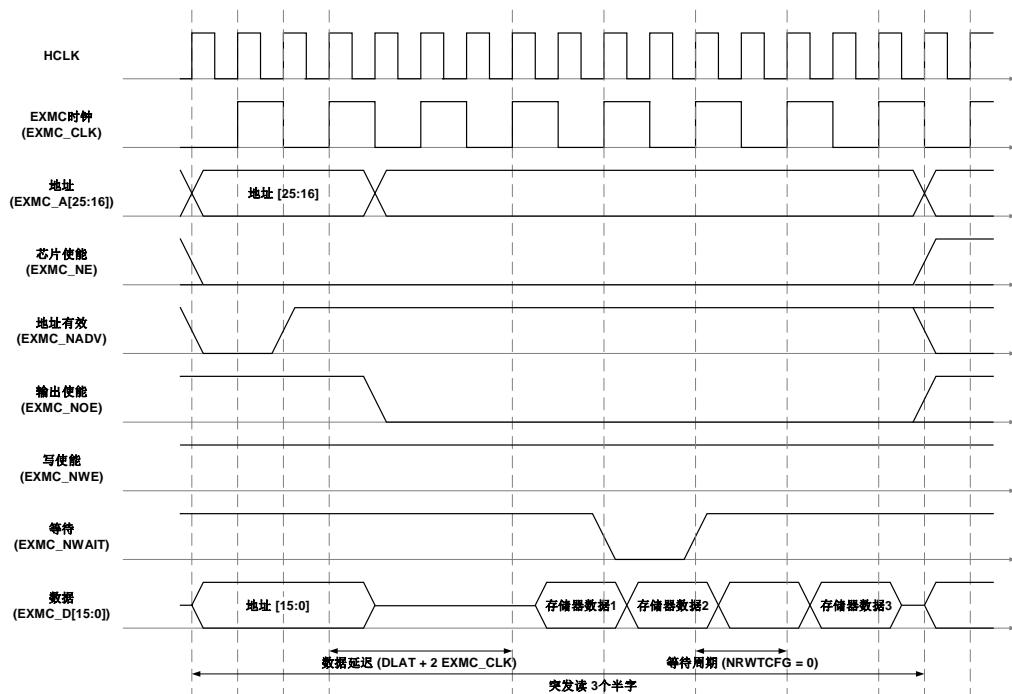


表 20-12. 同步复用模式读时序配置

位/位域	位名	参考设定值
EXMC_SNCTL		
31-20	保留	0x000
19	SYNCWR	无影响
18-16	CPS	0x0
15	ASYNCTEN	0x0
14	EXMODEN	0x0
13	NRWTEN	取决于存储器
12	WREN	无影响
11	NRWTCFG	取决于存储器
10	WRAPEN	0x0
9	NRWTPOL	取决于存储器
8	SBRSTEN	0x1, 突发读使能
7	保留	0x1
6	NREN	取决于存储器
5-4	NRW	0x1
3-2	NRTP	取决于存储器, 0x1/0x2
1	NRMUX	0x1, 取决于存储器与用户
0	NRBKEN	0x1
EXMC_SNTCFG (读)		
31-30	保留	0x0
29-28	ASYNCMOD	0x0
27-24	DLAT	数据延迟

位/位域	位名	参考设定值
23-20	CKDIV	上图设置: 0x1, EXMC_CLK=2HCLK
19-16	BUSLAT	从 EXMC_NE 上升沿到下降沿的时间
15-8	DSET	无影响
7-4	AHLD	无影响
3-0	ASET	无影响

模式SM – 同步复用突发写时序– PSRAM (CRAM)

图 20-19. 同步复用突发传输写时序

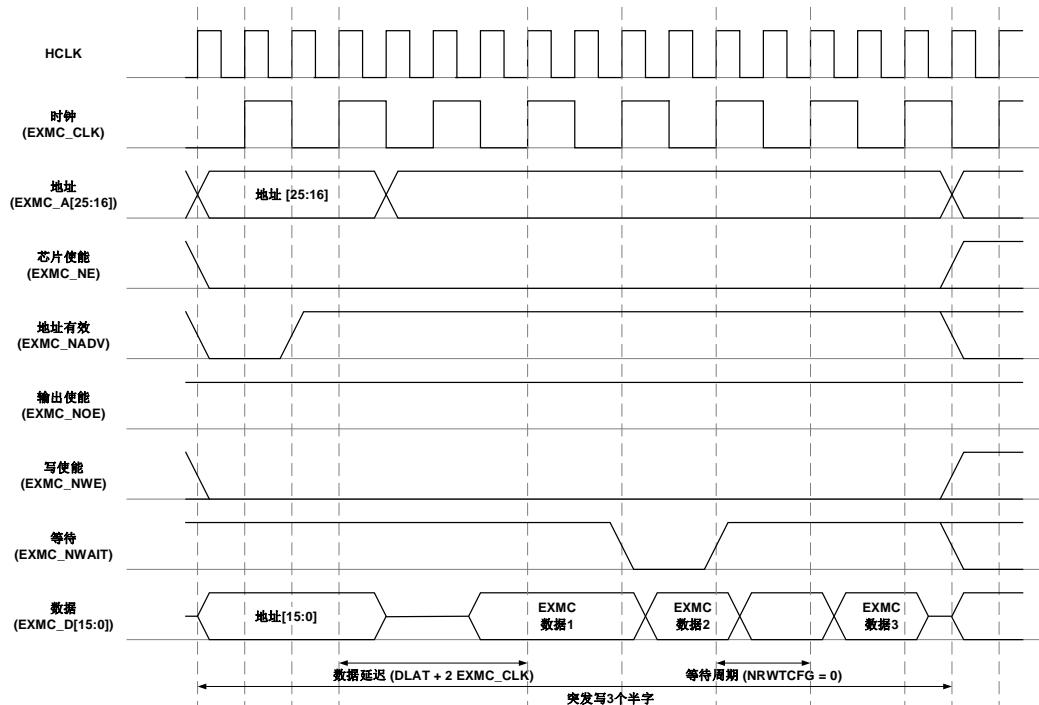


表 20-13. 同步复用模式写时序配置

位/位域	位名	参考设定值
EXMC_SNCTL		
31-20	保留	0x000
19	SYNCWR	0x1, 同步写使能
18-16	CPS	0x0
15	AYSNCWAIT	0x0
14	EXMODEN	0x0
13	NRWTEN	取决于存储器
12	WREN	0x1
11	NRWTCFG	0x0 (这里必须为 0)
10	WRAPEN	0x0
9	NTWTPOL	取决于存储器
8	SBRSTEN	无影响
7	保留	0x1
6	NREN	取决于存储器

位/位域	位名	参考设定值
5-4	NRW	0x1
3-2	NRTP	0x1
1	NRMUX	0x1, 取决于用户
0	NRBKEN	0x1
EXMC_SNTCFG (写)		
31-30	保留	0x0
29-28	ASYNCMOD	0x0
27-24	DLAT	数据延迟
23-20	CKDIV	上图设置: 0x1, EXMC_CLK=2HCLK
19-16	BUSLAT	从 EXMC_NE 上升沿到下降沿的时间
15-8	DSET	无影响
7-4	AHLD	无影响
3-0	ASET	无影响

20.4. EXMC 寄存器

EXMC基地址: 0xA000 0000

20.4.1. SRAM/NOR Flash 控制寄存器 (EXMC_SNCTL)

偏移地址: 0x00

复位值: 0x0000 30DB

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												SYNCWR	CPS[2:0]		
rw														rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASYNCW AIT	EXMODE N	NRWTEN	WREN	NRWTCF G	WRAPEN	NRWTP0 L	SBRSTE N	保留	NREN	NRW[1:0]	NRTP[1:0]	NRMUX	NRBKEN		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:20	保留	必须保持复位值
19	SYNCWR	选择写操作模式 0: 异步写操作 1: 同步写操作
18:16	CPS[2:0]	CRAM页大小 000: 页边界自动突发分割 001: 128字节 010: 256字节 011: 512字节 100: 1024字节 其他: 保留
15	ASYNCPWAIT	异步等待功能使能位 0: 禁用异步等待功能 1: 使能异步等待功能
14	EXMODEN	扩展模式使能 0: 禁用扩展模式 1: 使能扩展模式
13	NRWTEN	NWAIT信号使能 对于存储器的突发模式访问，该位使能/禁用在NWAIT信号中插入等待状态功能。 0: 禁用NWAIT信号 1: 使能NWAIT信号

12	WREN	写操作使能 0: 禁止EXMC对外部存储器的写操作, 否则产生一个AHB错误 1: 允许EXMC对外部存储器的写操作 (复位缺省值)
11	NRWTCFG	NWAIT信号配置, 只在同步模式有效 0: NWAIT信号在等待状态前的一个数据周期有效 1: NWAIT信号在等待状态期间有效
10	WRAPEN	非对齐突发模式使能 0: 禁止非对齐突发操作 1: 允许非对齐突发操作
9	NRWTPOL	NWAIT信号极性 0: NWAIT低电平有效 1: NWAIT高电平有效
8	SBRSTEN	同步突发模式使能 0: 禁止同步突发模式 1: 使能同步突发模式
7	保留	必须保持复位值
6	NREN	NOR闪存访问使能 0: 禁止NOR Flash访问 1: 允许NOR Flash访问
5:4	NRW[1:0]	存储器数据宽度 00: 8位 01: 16位 (复位缺省值) 10/11: 保留
3:2	NRTP[1:0]	存储器类型 00: SRAM 01: PSRAM (CRAM) 10: NOR Flash (复位之后的默认值) 11: 保留
1	NRMUX	数据线/地址线复用 0: 禁用地址/数据复用功能 1: 允许地址/数据复用功能
0	NRBKEN	存储块使能 0: 禁用对应的存储器块 1: 使能对应的存储器块

20.4.2. SRAM/NOR Flash 时序配置寄存器 (EXMC_SNTCFG)

偏移地址: 0x04

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
保留	ASYNCFMOD[1:0]	DLAT[3:0]			CKDIV[3:0]			BUSLAT[3:0]							
		rw			rw			rw			rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		DSET[7:0]			AHLD[3:0]			ASET[3:0]							
		rw			rw			rw			rw				

位/位域	名称	描述
31:30	保留	必须保持复位值
29:28	ASYNCFMOD[1:0]	异步访问模式 该位域仅在当EXMC_SNCTL寄存器的EXMODEN位为1时有效。 00: 模式A 01: 模式B 10: 模式C 11: 模式D
27:24	DLAT[3:0]	NOR Flash数据延时，仅在同步模式有效 0x0: 第一组突发传输时，数据延迟时间为2个EXMC_CLK时钟周期 0x1: 第一组突发传输时，数据延迟时间为3个EXMC_CLK时钟周期 0xF: 第一组突发传输时，数据延迟时间为17个EXMC_CLK时钟周期
23:20	CKDIV[3:0]	同步模式时钟分频比，仅在同步模式有效 0x0: 保留 0x1: EXMC_CLK周期=2个HCLK周期 0xF: EXMC_CLK周期=16个HCLK周期
19:16	BUSLAT[3:0]	总线延迟时间 在复用读模式中使用，避免总线冲突，是总线恢复到高阻态的最长时间。 0x0: 总线延迟=1个HCLK周期 0x1: 总线延迟=2个HCLK周期 0xF: 总线延迟=16个HCLK周期
15:8	DSET[7:0]	异步数据建立时间 该位域仅在异步模式有效。 0x00: 保留 0x01: 数据建立时间=2个HCLK周期 0xFF: 数据建立时间=256个HCLK周期

7:4	AHLD[3:0]	异步地址保持时间 该位域设置地址保持时间，仅在模式D与复用模式有效。 0x0: 保留 0x1: 地址保持时间=2个HCLK 0xF: 地址保持时间=16个HCLK
3:0	ASET[3:0]	异步地址建立时间 该位域设置地址建立时间。 注意： 该位域仅在SRAM, ROM, NOR Flash的异步模式有效。 0x0: 地址建立时间= 1个HCLK 0xF: 地址建立时间= 16个HCLK

20.4.3. SRAM/NOR Flash 写时序寄存器 (EXMC_SNWTCFG)

偏移地址: 0x104

复位值: 0x0FFF FFFF

该寄存器仅在扩展模式使能（寄存器EXMC_SNCTL位EXMODEN置1）后有效。

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		WASYNCMOD[1:0]	保留									WBUSLAT[3:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												WDSET[7:0]	WAHLD[3:0]	WASET[3:0]	

位/位域	名称	描述
31:30	保留	必须保持复位值
29:28	WASYNCMOD[1:0]	异步访问模式 该位只有在扩展模式中使用。 00: 模式A 01: 模式B 10: 模式C 11: 模式D
27:20	保留	必须保持复位值
19:16	WBUSLAT[3:0]	总线延迟时间 在每次写传输结束的时候增加总线延时时间来满足连续传输之间的最长时间。 0x0: 总线延迟=1个HCLK周期 0x1: 总线延迟=2个HCLK周期

	
15:8	WDSET[7:0]	0xF: 总线延迟=16个HCLK周期 异步数据建立时间 该位域仅在异步模式有效。 0x00: 保留 0x01: 数据建立时间=2个HCLK周期
7:4	WAHLD[3:0]	0xFF: 数据建立时间=256个HCLK周期 异步地址保持时间 该位域设置地址保持时间，仅在模式D与复用模式有效。 0x0: 保留 0x1: 地址保持时间=2个HCLK
3:0	WASET[3:0]	0xF: 地址保持时间=16个HCLK 异步地址建立时间 该位域设置地址建立时间 注意： 该位域仅在SRAM, ROM, NOR Flash的异步模式有效。 0x0: 地址建立时间= 1个HCLK 0x1: 地址建立时间= 2个HCLK 0xF: 地址建立时间= 16个HCLK

21. 控制器局域网络（CAN）

21.1. 简介

CAN (Controller Area Network) 总线是一种可以在无主机情况下实现微处理器或者设备之间相互通信的总线标准。

CAN 总线控制器作为 CAN 网络接口，遵循 CAN 总线协议 2.0A、2.0B、ISO11891-1:2015 和 BOSCH CAN-FD 规范。CAN 总线控制器可以处理总线上的数据收发并具有 28 个过滤器，过滤器用于筛选并接收用户需要的消息。用户可以通过 3 个发送邮箱将待发送数据传输至总线，邮箱发送的顺序由发送调度器决定。并通过 2 个深度为 3 的接收 FIFO 获取总线上的数据，接收 FIFO 的管理完全由硬件控制。同时 CAN 总线控制器硬件支持时间触发通信（Time-triggered communication）功能。

21.2. 主要特征

- 支持 CAN 总线协议 2.0A 和 2.0B;
- 支持 CAN-FD 帧（ISO11898-1 和 CAN-FD 规范 V1.0）;
- 常规帧：通信波特率最大为 1Mbit/s;
- CAN-FD 帧：通信波特率最大为 6Mbit/s;
- 支持传输延迟补偿;
- 支持时间触发通信（Time-triggered communication）;
- 中断使能和清除。

发送功能

- 3 个发送邮箱;
- 支持发送优先级;
- 支持发送时间戳。

接收功能

- 2 个深度为 3 的接收 FIFO;
- 具有 28 个标识符过滤器;
- FIFO 锁定功能。

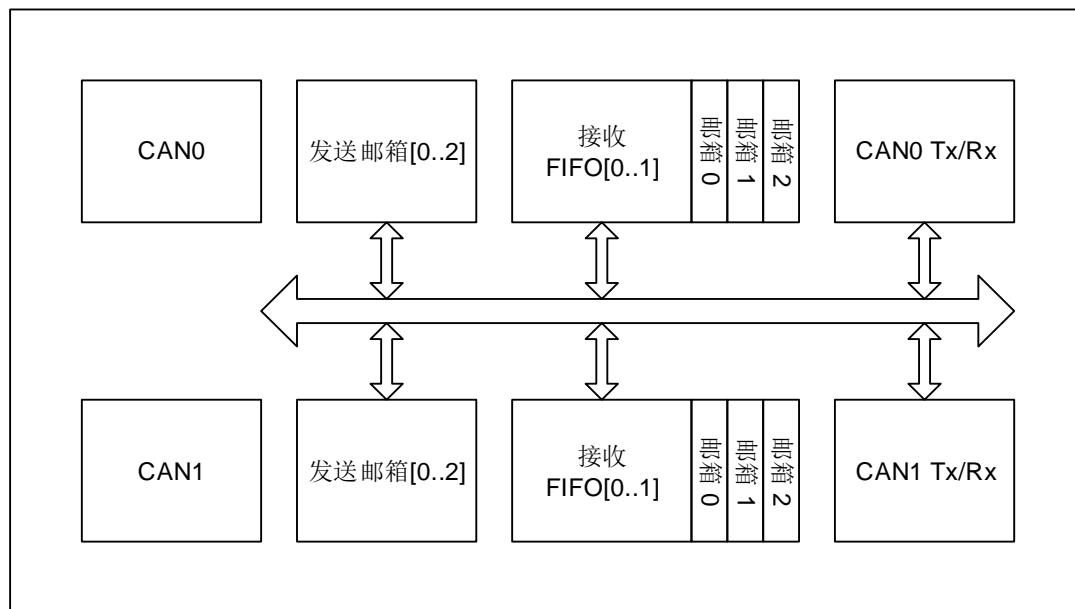
时间触发通信

- 在时间触发通信模式下禁用自动重传;
- 16 位定时器;
- 接收时间戳;
- 发送时间戳。

21.3. 功能说明

CAN 模块结构框图如 [图 21-1. CAN 模块结构框图](#) 所示。

图 21-1. CAN 模块结构框图



21.3.1. 工作模式

CAN 总线控制器有 3 种工作模式：

- 睡眠工作模式；
- 初始化工作模式；
- 正常工作模式。

睡眠工作模式

芯片复位后，CAN总线控制器处于睡眠工作模式。该模式下CAN总线控制器的时钟停止工作并处于一种低功耗状态。

将CAN_CTL寄存器的SLPWMOD位置1，可以使CAN总线控制器进入睡眠工作模式。当进入睡眠工作模式后，CAN_STAT寄存器的SLPWS位将被硬件置1。

将CAN_CTL寄存器的AWU位置1，并当CAN检测到总线活动时，CAN总线控制器将自动退出睡眠工作模式。将CAN_CTL寄存器的SLPWMOD位清0，也可以退出睡眠工作模式。

由睡眠模式进入初始化工作模式：将CAN_CTL寄存器的IWMOD位置1，SLPWMOD位清0。

由睡眠模式进入正常工作模式：将CAN_CTL寄存器的IWMOD位和SLPWMOD位清0。

初始化工作模式

如果需要配置 CAN 总线通信参数，CAN 总线控制器必须进入初始化工作模式。将 CAN_CTL

寄存器的 IWMOD 位置 1，使 CAN 总线控制器进入初始化工作模式，将其清 0 则离开初始化工作模式。在进入初始化工作模式后，CAN_STAT 寄存器的 IWS 位将被硬件置 1。

由初始化模式进入睡眠模式：CAN_CTL 寄存器的 SLPWMOD 位置 1，IWMOD 位清 0。

由初始化模式进入正常工作模式：CAN_CTL 寄存器的 SLPWMOD 位和 IWMOD 位清 0。

正常工作模式

在初始化工作模式中配置完 CAN 总线通信参数后，将 CAN_CTL 寄存器的 IWMOD 位清 0 可以进入正常工作模式并与 CAN 总线网络中的节点进行正常通信。

由正常工作模式进入睡眠工作模式：CAN_CTL 寄存器的 SLPWMOD 位置 1，并等待当前数据收发过程结束。

由正常工作模式初始化工作模式：CAN_CTL 寄存器的 IWMOD 位置 1，并等待当前数据收发过程结束。

21.3.2 通信模式

CAN 总线控制器有 4 种通信模式：

- 静默（Silent）通信模式；
- 回环（Loopback）通信模式；
- 回环静默（Loopback and Silent）通信模式；
- 正常（Normal）通信模式。

静默（Silent）通信模式

在静默通信模式下，可以从 CAN 总线接收数据，但不向总线发送任何数据。将 CAN_BT 寄存器中的 SCMOD 位置 1，使 CAN 总线控制器进入静默通信模式，将其清 0 可以退出静默通信模式。

静默通信模式可以用来监控 CAN 网络上的数据传输。

回环（Loopback）通信模式

在回环通信模式下，由 CAN 总线控制器发送的数据可以被自己接收并存入接收 FIFO，同时这些发送数据也送至 CAN 网络。将 CAN_BT 寄存器中的 LCMOD 位置 1，使 CAN 总线控制器进入回环通信模式，将其清 0 可以退出回环通信模式。

回环通信模式通常用来进行 CAN 通信自测。

回环静默（Loopback and Silent）通信模式

在回环静默通信模式下，CAN 的 RX 和 TX 引脚与 CAN 网络断开。CAN 总线控制器既不从 CAN 网络接收数据，也不向 CAN 网络发送数据，其发送的数据仅可以被自己接收。将 CAN_BT 寄存器中的 LCMOD 位和 SCMOD 位置 1，使 CAN 总线控制器进入回环静默通信模式，将它们清 0 可以退出回环静默通信模式。

回环静默通信模式通常用来进行 CAN 通信自测。对外 TX 引脚保持隐性状态（逻辑 1），RX 引脚保持高阻态。

正常（Normal）通信模式

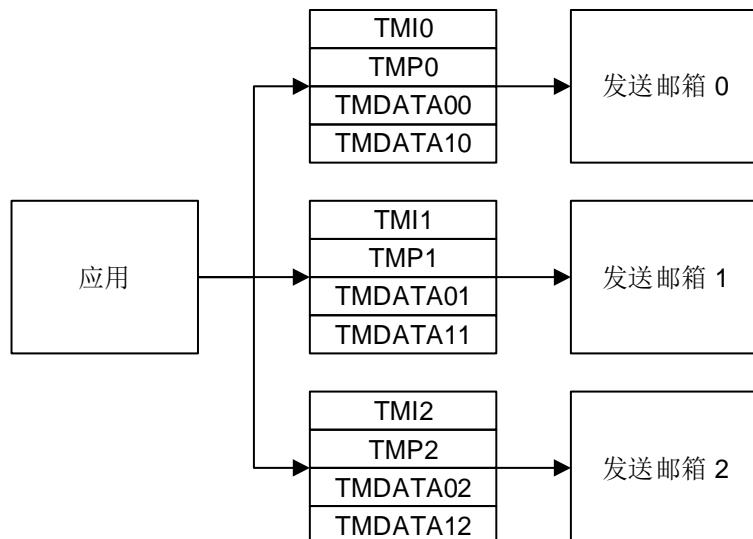
CAN 总线控制器通常工作在正常通信模式下，可以从 CAN 总线接收数据，也可以向 CAN 总线发送数据。这时需要将 CAN_BT 寄存器的 LCMOD 位和 SCMOD 位清 0。

21.3.3. 数据发送

发送寄存器

数据发送通过 3 个发送邮箱进行，可以通过寄存器 CAN_TMIx, CAN_TMPx, CAN_TMDATA0x 和 CAN_TMDATA1x 对发送邮箱进行配置。如 [图 21-2. 发送寄存器](#) 所示。

图 21-2. 发送寄存器



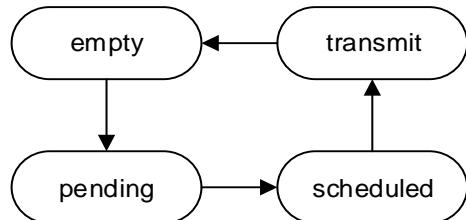
如果想要发送CAN-FD帧，使用发送邮箱x（x等于0到2）时，仅需要写对应的TMDATA0x（x等于0到2）寄存器。例如软件想要使用发送邮箱0发送64字节数据，需要通过写TMDATA00寄存器16次将待发送数据写入内部专用SRAM区。

发送邮箱状态转换

当发送邮箱处于 **empty** 状态时，应用程序才可以对邮箱进行配置。当邮箱被配置完成后，可以将 CAN_TMIx 寄存器的 TEN 位置 1，从而向 CAN 总线控制器提交发送请求，这时发送邮箱处于 **pending** 状态。当超过 1 个邮箱处于 **pending** 状态时，需要对多个邮箱进行调度，这时发送邮箱处于 **scheduled** 状态。当调度完成后，发送邮箱中的数据开始向 CAN 总线上发送，这时发送邮箱处于 **transmit** 状态。当数据发送完成，邮箱变为空闲，可以再次交给应用

程序使用，这时发送邮箱重新变为 **empty** 状态。如[图 21-3. 发送邮箱状态转换](#)所示。

图 21-3. 发送邮箱状态转换



发送状态和错误信息

CAN_TSTAT 寄存器中的 MTF, MTFNERR, MAL 和 MTE 位用来说明发送状态和错误信息。

- **MTF**: 发送完成标志位。当数据发送完成时，MTF 置 1。
- **MTFNERR**: 无错误发送完成标志位。当数据发送完成且没有错误时，MTFNERR 置 1。
- **MAL**: 仲裁失败标志位。当发送数据过程中出现仲裁失败时，MAL 置 1。
- **MTE**: 发送错误标志位。当发送过程中检测到总线错误时，MTE 置 1。

数据发送步骤

数据发送步骤如下：

第1步：选择一个空闲发送邮箱；

第2步：根据应用程序要求，配置4个发送寄存器；

第3步：将CAN_TMIx寄存器的TEN置1；

第4步：检测发送状态和错误信息。典型情况是检测到MTF和MTFNERR置1，说明数据被成功发送。

发送选项

中止数据发送

将CAN_TSTAT寄存器的MST置1，可以中止数据发送。

当发送邮箱处于**pending**和**scheduled**状态，CAN_TSTAT寄存器的MST置1可以立即中止数据发送。

当发送邮箱处于**transmit**状态，则面临两种情况。一种情况是数据发送被成功地完成，MTF和MTFNERR为1，这时发送邮箱将转换为**empty**状态。相对的，如果数据发送过程中出现了问题，这时发送邮箱将转换为**scheduled**状态，这时数据发送被中止。

发送优先级

当有2个及其以上发送邮箱等待发送时，寄存器CAN_CTL的TFO位的值可以决定发送顺序。

当TFO为1，所有等待发送的邮箱按照先来先发送（FIFO）的顺序进行。

当TFO为0，具有最小标识符（Identifier）的邮箱最先发送。如果所有的标识符（Identifier）相等，具有最小邮箱编号的邮箱最先发送。

21.3.4. 数据接收

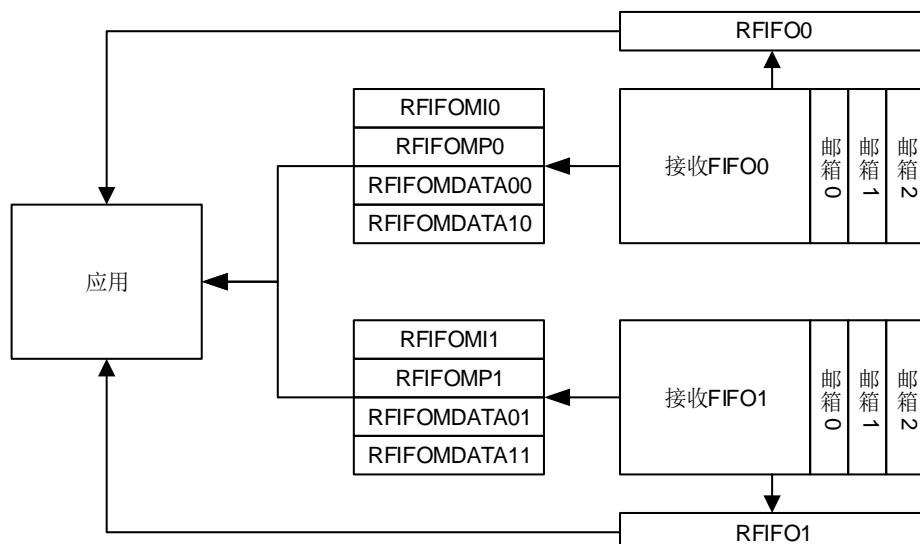
接收寄存器

应用程序通过2个深度为3的FIFO接收来自CAN网络的数据。

寄存器 CAN_RFIFOx 可以操作 FIFO，也包含 FIFO 状态。寄存器 CAN_RFIFOMIx , CAN_RFIFOMPx, CAN_RFIFOMDATA0x和CAN_RFIFOMDATA1x用于接收数据帧。

如[图 21-4. 接收寄存器](#)所示。

图 21-4. 接收寄存器



接收 FIFO

每个接收FIFO包含3个接收邮箱，用来接收存储数据帧。这些邮箱按照先进先出方式进行组织，最早从CAN网络接收的数据，最早被应用程序处理。

寄存器CAN_RFIFOx包含FIFO状态信息和帧的数量。当FIFO中包含数据时，可以通过寄存器 CAN_RFIFOMIx, CAN_RFIFOMPx, CAN_RFIFOMDATA0x和CAN_RFIFOMDATA1x读取数据，之后将寄存器CAN_RFIFOx的RFD位置1释放邮箱，并且等待其由硬件自动清0。

如果接收到 CAN-FD 帧，数据将被存储到内部专用 SRAM 中，并通过多次读 CAN_RFIFOMDATA0x寄存器将数据取出。FIFO0使用CAN_RFIFOMDATA00，FIFO1使用 CAN_RFIFOMDATA01寄存器。例如，如果软件想要从FIFO0中读64字节数据，需要通过读 CAN_RFIFOMDATA00寄存器16次将数据全部读出。

接收 FIFO 状态信息

接收FIFO状态信息包含在寄存器CAN_RFIFOx中。

RFL: FIFO中包含的帧数量。FIFO为空时，RFL为0；FIFO为满时，RFL为3。

RFF: FIFO满状态标志位。这时RFL为3。

RFO: FIFO溢出标志位。当FIFO已经包含了3个数据帧时，新的数据帧到来使FIFO发生溢出。如果CAN_CTL寄存器的RFOD位被置1，新的数据帧将丢弃。如果该位被清0，新的数据帧将覆盖接收FIFO中最后一帧数据。

数据接收步骤

第1步：查看FIFO中帧的数量。

第2步：通过CAN_RFIFOMIx, CAN_RFIFOMPx, CAN_RFIFOMDATA0x和CAN_RFIFOMDATA1x读取数据。

第3步：将寄存器CAN_RFIFOx的RFD置1释放邮箱，并且等待其由硬件自动清0。

21.3.5. 过滤功能

一个待接收的数据帧会根据其标识符（Identifier）进行过滤：硬件会将通过过滤的帧送至接收FIFO，并丢弃没有通过过滤的帧。

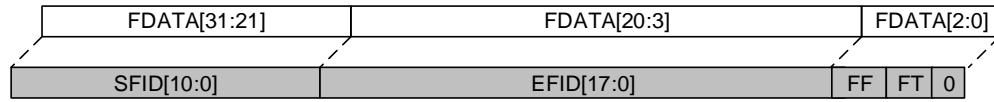
过滤器位宽

过滤器包含28个单元，它们是bank0到bank27。

每一个过滤器单元有2个寄存器CAN_FxDATA0和CAN_FxDATA1，它们可以配置为2种位宽：32-bit位宽和16-bit位宽。

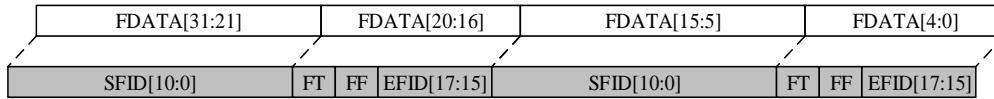
32-bit 位宽：CAN_FDATAx 包含字段 SFID[10:0], EFID[17:0], FF 和 FT。如[图 21-5. 32-bit 位宽过滤器](#)所示。

图 21-5. 32-bit 位宽过滤器



16-bit 位宽：CAN_FDATAx 包含字段：SFID[10:0], FT, FF 和 EFID[17:15]。如[图 21-6. 16-bit 位宽过滤器](#)所示。

图 21-6. 16-bit 位宽过滤器



掩码模式

对于一个待过滤的数据帧的标识符 (Identifier)，掩码模式用来指定哪些位必须与预设的标识符相同，哪些位无需判断。

一个 32-bit 位宽掩码模式过滤器如 [图 21-7. 32-bit 位宽掩码模式过滤器](#) 所示。

图 21-7. 32-bit 位宽掩码模式过滤器

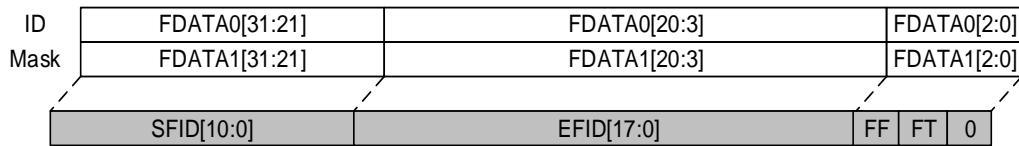
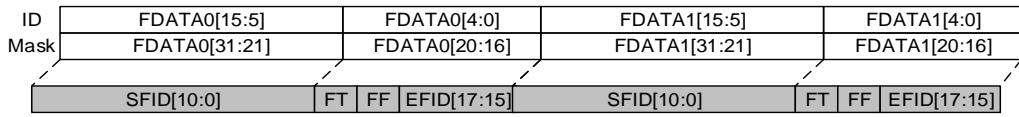


图 21-8. 16-bit 位宽掩码模式过滤器



列表模式

对于一个待过滤的数据帧的标识符 (Identifier)，列表模式用来表示与预设的标识符列表中能够匹配则通过，否则丢弃。

一个 32-bit 位宽列表模式过滤器如 [图 21-9. 32-bit 位宽列表模式过滤器](#) 所示。

图 21-9. 32-bit 位宽列表模式过滤器

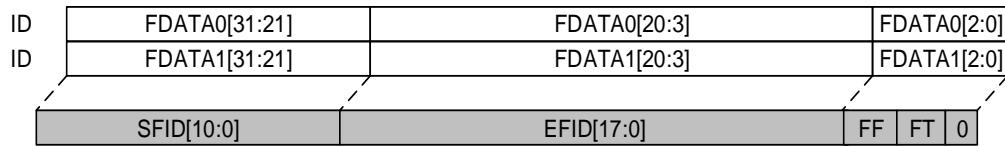
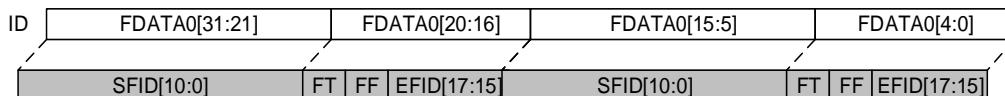


图 21-10. 16-bit 位宽列表模式过滤器



过滤序号

过滤器由若干过滤单元 (Bank) 组成，每个过滤单元因为位宽和模式的选择不同，而具有不同的过滤效果。例如 [表 21-1. 32-bit 过滤序号](#) 所示的 2 个过滤单元，Bank0 是 32-bit 位宽掩码模式，Bank1 是 32-bit 位宽列表模式。

表 21-1. 32-bit 过滤序号

过滤单元	过滤器数据寄存器	过滤序号
0	F0DATA0-32bit-ID	0
	F0DATA1-32bit-Mask	
1	F1DATA0-32bit-ID	1

	F1DATA1-32bit-ID	2
--	------------------	---

过滤器关联的 FIFO

28个过滤单元均可以关联接收FIFO0或接收FIFO1。一旦一个过滤单元关联到接收FIFO，只有通过这个过滤单元的帧才会被传送到接收FIFO中存储。

过滤器激活控制

一个过滤单元如果被应用程序用到，就必须激活。通过CAN_FW寄存器可以进行配置。

过滤索引

一个包含过滤序号（Filter Number）N 的过滤单元通过了某个帧，则该帧数据的过滤索引（Filtering Index）为 N。这时 CAN_RFIFOMPx 中 FI 的值为 N。

在表 21-2. 过滤索引中，如果一个帧通过了 FIFO0 中过滤序号 10（Filter Number=10）的过滤单元，那么该帧的过滤索引为 10。这时 CAN_RFIFOMPx 中 FI 的值为 10。

过滤序号不关心对应的过滤单元（Bank）是否处于工作状态。例如Bank3被关联到FIFO0，且为“不激活”状态，但它仍然包含过滤序号3和4。

表 21-2. 过滤索引

过滤单元	FIFO0	激活	过滤序号	过滤单元	FIFO1	激活	过滤序号
0	F0DATA0-32bits-ID	是	0	2	F2DATA0[15:0]-16bits-ID	是	0
	F0DATA1-32bits-Mask				F2DATA0[31:16]-16bits-Mask		
1	F1DATA0-32bits-ID	是	1	2	F2DATA1[15:0]-16bits-ID	是	1
	F1DATA1-32bits-ID		2		F2DATA1[31:16]-16bits-Mask		
3	F3DATA0[15:0]-16bits-ID	否	3	4	F4DATA0-32bits-ID	否	2
	F3DATA0[31:16]-16bits-Mask				F4DATA1-32bits-Mask		
	F3DATA1[15:0]-16bits-ID		4	5	F5DATA0-32bits-ID	否	3
	F3DATA1[31:16]-16bits-Mask				F5DATA1-32bits-ID		
7	F7DATA0[15:0]-16bits-ID	否	5	6	F6DATA0[15:0]-16bits-ID	是	5
	F7DATA0[31:16]-16bits-ID		6		F6DATA0[31:16]-16bits-ID		6
	F7DATA1[15:0]-		7		F6DATA1[15:0]-16bits-		7

过滤单元	FIFO0	激活	过滤序号	过滤单元	FIFO1	激活	过滤序号
	16bits-ID		8		ID		8
	F7DATA1[31:16]-16bits-ID				F6DATA1[31:16]-16bits-ID		
8	F8DATA0[15:0]-16bits-ID	是	9	10	F10DATA0[15:0]-16bits-ID	9	9
	F8DATA0[31:16]-16bits-ID		10		F10DATA0[31:16]-16bits-Mask		
	F8DATA1[15:0]-16bits-ID		11		F10DATA1[15:0]-16bits-ID	否	10
	F8DATA1[31:16]-16bits-ID		12		F10DATA1[31:16]-16bits-Mask		
9	F9DATA0[15:0]-16bits-ID	是	13	11	F11DATA0[15:0]-16bits-ID	11	11
	F9DATA0[31:16]-16bits-Mask		14		F11DATA0[31:16]-16bits-ID		
	F9DATA1[15:0]-16bits-ID		13		F11DATA1[15:0]-16bits-ID	否	13
	F9DATA1[31:16]-16bits-Mask		14		F11DATA1[31:16]-16bits-ID		
12	F12DATA0-32bits-ID	是	15	13	F13DATA0-32bits-ID	15	15
	F12DATA1-32bits-Mask		16		F13DATA1-32bits-ID		

优先级

过滤器优先级规则如下：

- 1、32-bits位宽模式高于16-bits位宽模式；
- 2、列表模式高于掩码模式；
- 3、较小的过滤序号（Filter Number）具有较高的优先级。

21.3.6. 时间触发通信

时间触发通信是CAN数据链路层应用协议。CAN网络中的所有节点都按照一个预先设定的时间序列进行通信，尤其适合于时间周期性应用和时间确定性应用。

在这种通信模式下，一个内部的16-bit计数器开始工作，在每一个CAN位时间（Bit time）增1。这个内部计数器为数据发送和数据接收提供时间戳，这些时间戳存放在寄存器CAN_RFIFOMPx和CAN_TMPx中。

在这种通信模式下，自动重发功能是禁止的。

21.3.7. 通信参数

自动重发禁止模式

在时间触发通信模式下，要求自动重发必须是禁止的，可以通过将CAN_CTL寄存器的ARD位置1满足要求。

在这种模式下，数据只会被发送一次，如果因为仲裁失败或者总线错误而导致发送失败，CAN总线控制器不会像通常那样进行数据自动重发。

发送结束时，寄存器CAN_TSTAT的MTF位被硬件置1，而发送状态信息可以通过MTFNERR, MAL和MTE获得。

位时序 (Bit time)

CAN协议采用位同步传输方式。这种方式不仅增大了传输容量，而且意味着需要一种复杂的位同步方法。面向字节传输的位同步方式适用于接收在每个字节前都有起始位的情况，而同步传输协议只要求数据帧的最开始有一个起始位。为保证接收器能正确读取信息，需要不断地进行重新同步。因此，在相位缓冲段采样点前面和后面都应该插入一个帧间隔。

可以通过位操作仲裁方式访问CAN总线。信号从发送器到接收器，再回到发送器必须在一个位时间内完成。为了达到同步的目的，除了相位缓冲段外，还需要一个传输延时段。在信号传输过程中，传输延时段被视为发送或接收延时。

CAN总线控制器将位时间分为3个部分。

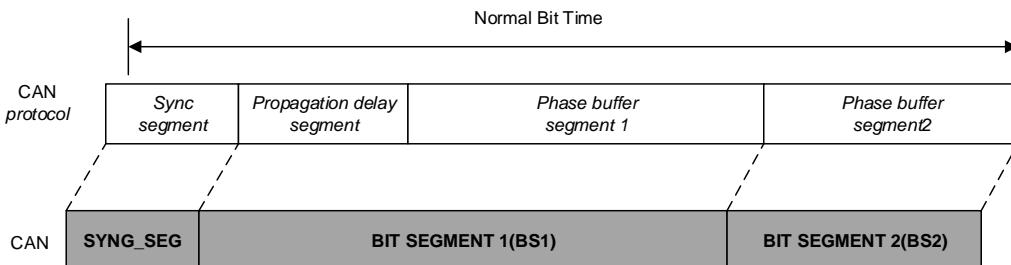
同步段 (Synchronization segment)，记为SYNC_SEG。该段占用1个时间单元 ($1 \times t_{CAN}$)。

位段1 (Bit segment 1)，记为BS1。相对于CAN协议而言，BS1相当于传播时间段 (Propagation delay segment) 和相位缓冲段1 (Phase buffer segment 1)。

位段2 (Bit segment 2)，记为BS2。相对于CAN协议而言，BS2相当于相位缓冲段2 (Phase buffer segment 2)。

对比与 CAN 协议，位时序如 [图 21-11. 位时序](#) 所示。

图 21-11. 位时序



再同步补偿宽度SJW (resynchronization Jump Width) 对CAN网络节点同步误差进行补偿。

有效跳变定义为，在CAN控制器，没有发送隐性位时，一个位时间内显性位到隐性位的第一次转变。

如果有效跳变在BS1期间被检测到，而不是SYNC_SEG期间，BS1将会最多被延长SJW，因此采样点延时。

相反，如果有效跳变在BS2期间被检测到，而不是SYNC_SEG期间，BS2将会最多被缩短SJW，因此采样点提前。

波特率

波特率计算公式如下：

$$\text{BaudRate} = \frac{1}{\text{Normal Bit Time}} \quad (\text{式22-1})$$

$$\text{Normal Bit Time} = t_{\text{SYNC_SEG}} + t_{\text{BS1}} + t_{\text{BS2}} \quad (\text{式22-2})$$

其中：

$$t_{\text{SYNC_SEG}} = 1 \times t_q \quad (\text{式22-3})$$

$$t_{\text{BS1}} = (1 + \text{BT.BS1}) \times t_q \quad (\text{式22-4})$$

$$t_{\text{BS2}} = (1 + \text{BT.BS2}) \times t_q \quad (\text{式22-5})$$

$$t_q = (1 + \text{BT.BAUDPSC}) \times t_{\text{PCLK1}} \quad (\text{式22-6})$$

21.3.8. CAN FD 操作

通过将 CAN_FDCTL 寄存器的 FDEN 位置 1，可以使能 CAN FD (CAN with Flexible Data rate) 功能。如果 FDEN 位被清 0，CAN 总线控制器仅支持常规帧（标准帧和扩展帧）的收发，若 FDEN 位被置 1，则 CAN 总线控制器同时支持常规帧（标准帧和扩展帧）以及 FD 帧的收发。根据协议，当前帧是否为 FD 帧是通过帧的 FDF 位来判断（在常规帧中该位为保留位）。如果 FDF 位为隐性，表示是 CAN FD 帧；如果为显性，表示是常规帧。

通过配置 CAN_FDCTL 寄存器的 NISO 位，可以选择 CAN-FD 功能支持 ISO11898-1 或 BOSCH CAN FD 规范 V1.0。

在 CAN-FD 帧的帧结构中，FDF 位之后是保留位和 BRS 位。BRS 位决定数据位速率，当 BRS 位为显性时，表示不能通过配置 CAN_DBT 寄存器来切换数据位速率。当 BRS 位为隐性时，可以通过配置 CAN_DBT 寄存器使得数据段（从 BRS 位到 ACK 场之前）的数据速率高于仲裁段的数据速率。详情请参考 ISO11898-1 或 BOSCH CAN FD 规范 V1.0。

通过将 CAN_FDCTL 寄存器的 PRED 位清 0，可以使能协议异常处理功能。此时，在接收帧数据过程中检测到隐性的保留位时，该功能将使操作状态转变为 IDLE 并在下一个采样点中止当前帧。反之，将 PRED 位置 1，该功能将被禁止，隐性的保留位将被视为格式错误，并当做错误帧来进行处理，同时 CAN_FDSTAT 寄存器的 PRE 位将被置 1。

ISO11898-1 或 BOSCH CAN FD 规范 V1.0 规定的发送 ESI 位（该位位于 CAN FD 帧的 DLC 位域之前）功能通过 CAN_FDCTL 寄存器的 ESIMOD 位和 CAN_TMPx 寄存器的 ESI 位来实现。如果将 ESIMOD 位清 0，当 CAN 总线控制器处于被动错误状态时，该位为隐性；当处于主动错误状态时，该位为显性。若将 ESIMOD 位置 1，将根据 CAN_TMPx 寄存器的 ESI 位的值来决定该位为显性还是隐性。

发送帧 FDF 位和 BRS 位的总线电平逻辑由 CAN_TMPx 寄存器的 FDF 位和 BRS 位的值决定。

21.3.9. 传输延迟补偿

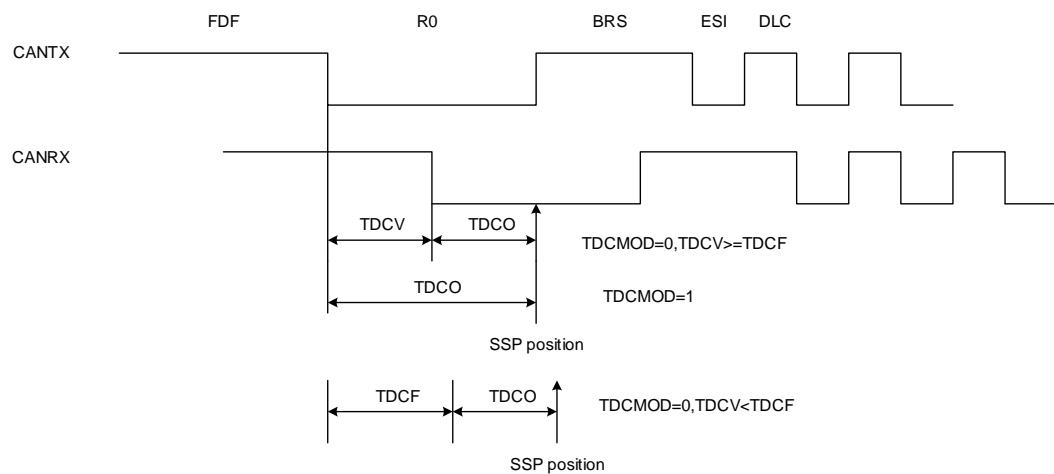
CAN-FD 协议支持传输延迟补偿机制。由于 CAN 收发器存在回路延迟，因此当发送 CAN-FD 帧的高速数据段的位时间长度小于收发器内部回路延迟的限定值时，该机制可以避免当采样点到来时发送节点还没有收到其自己发出的位，从而报错的情况发生。关于传输延迟补偿的具体定义，请参考 ISO11898-1 或 BOSCH CAN FD 规范 V1.0。

将 CAN_FDCTL 寄存器的 TDCEN 位置 1 将使能传输延迟补偿。

传输延迟补偿机制可以调节次级采样点 (SSP) 的位置。SSP_Delay 被定义为 CANTX 上的信号到 SSP 采样点的延迟。如果 CAN_FDCTL 寄存器的 TDCMOD 位被置 1，SSP_Delay 的值由 CAN_FDTDC 寄存器的 TDCO 位域软件配置决定。如果 TDCMOD 位被清 0，硬件将自动计算位速率转换之前的 FDF 位到 RES0 位的下降沿在 CAN_TX 与 CAN_RX 上出现的延迟，并将计算值存入 CAN_FDSTAT 寄存器的 TDCV 位域。由于存在信号毛刺，可能导致硬件自动计算的 SSP 位置比预期的提前，为了避免 TDCV 的值过小，可以使用 CAN_FDTDC 寄存器的 TDVF 位域。如果 TDCV 的值大于 TDVF，SSP_Delay 的值被定义为 TDCO 加上 TDCV，否则 SSP_Delay 的值被定义为 TDCO 加上 TDVF。

SSP_Delay 的值不能大于 3 个数据位时间。

图 21-12 传输延迟测量



21.3.10. 错误标志

CAN总线的状态可以通过CAN_ERR寄存器的发送错误计数值（Transmit Error Counter，记为TECNT）和接收错误计数值（Receive Error Counter，记为RECNT）反映，其值会根据错误的情况由硬件增加或减少，软件可以通过这些值判断CAN网络的稳定性。关于错误计数值的详细信息请参考CAN协议相关章节。

同时寄存器CAN_ERR还可以表明当前错误状态，这些错误状态在寄存器CAN_INTEN控制下产生中断。

离线恢复

当TECNT大于255时，CAN总线控制器进入离线状态，这时寄存器CAN_ERR中的BOERR置1，并且发送和接收失效。

根据寄存器CAN_CTL中的ABOR配置，离线恢复（变为主动错误状态）有2种方式。这两种方式都要求处于离线状态的CAN总线控制器检测到CAN协议所定义的离线恢复序列（在CAN_RX检测到128次连续11个位的隐性位）时，才会自动恢复。

如果ABOR为1，将在检测到离线恢复序列后自动恢复。

如果ABOR为0，则必须先将CAN_CTL中的IWMOD置1进入初始化工作模式，然后进入正常工作模式并在检测到离线恢复序列后恢复。

21.3.11. 中断

CAN总线控制器占用4个中断向量，通过寄存器CAN_INTEN进行控制。这4个中断向量对应4类中断源：

- 发送中断；
- FIFO0 中断；
- FIFO1 中断；
- 错误和状态改变中断。

发送中断

发送中断包括：

- 寄存器CAN_TSTAT中的MTF0置1：发送邮箱0变为空闲。
- 寄存器CAN_TSTAT中的MTF1置1：发送邮箱1变为空闲。
- 寄存器CAN_TSTAT中的MTF2置1：发送邮箱2变为空闲。

FIFO0 中断

FIFO0中断包括：

- FIFO0中包含待接收数据：寄存器CAN_RFIFO0中的RFL0不为0，CAN_INTEN寄存器中RFNEIE0被置位；
- FIFO0满：寄存器CAN_RFIFO0中的RFF0为1，CAN_INTEN寄存器中RFFIE0被置位；
- FIFO溢出：寄存器CAN_RFIFO0中的RFO0为1，CAN_INTEN寄存器中RFOIE0被置位。

FIFO1 中断

FIFO1中断包括：

- FIFO1中包含待接收数据：寄存器CAN_RFIFO1中的RFL1不为0，CAN_INTEN寄存器中RFNEIE1被置位；
- FIFO1满：寄存器CAN_RFIFO1中的RFF1为1，CAN_INTEN寄存器中RFFIE1被置位；
- FIFO溢出：寄存器CAN_RFIFO1中的RFO1为1，CAN_INTEN寄存器中RFOIE1被置位。

错误和工作模式改变中断

错误和工作模式改变中断可由以下条件触发：

- 错误： CAN_STAT 寄存器的ERRIF 和 CAN_INTEN 寄存器的ERRIE 被置位，请参考 CAN_STAT 寄存器中ERRIF位描述；
- 唤醒： CAN_STAT 寄存器中的WUIF 和 CAN_INTEN 寄存器的WIE 被置位；
- 进入睡眠模式： CAN_STAT 寄存器中的SLPIF 和 CAN_INTEN 寄存器的SLPWIE 被置位。

21.4. CAN 寄存器

CAN0基址: 0x4000 6400

CAN1基址: 0x4000 6800

21.4.1. 控制寄存器 (CAN_CTL)

地址偏移: 0x00

复位值: 0x0001 0002

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															DFZ
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWRST	保留							TTC	ABOR	AWU	ARD	RFOD	TFO	SLPWMOD	IWMOD
rs	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	DFZ	<p>调试冻结</p> <p>如果DBG_CTL寄存器中CANx_HOLD被置位，该位用来定义CAN控制器工作在调试冻结或正常工作状态。如果DBG_CTL寄存器中CANx_HOLD被清零，该位无效。</p> <p>0: 处于Debug时，CAN接收和发送正常工作</p> <p>1: 处于Debug时，CAN接收和发送停止</p>
15	SWRST	<p>软件复位</p> <p>0: 正常操作</p> <p>1: 复位CAN并进入睡眠工作模式。该位会自动清0。</p>
14:8	保留	必须保持复位值。
7	TTC	<p>时间触发通信</p> <p>0: 禁用时间触发通信</p> <p>1: 使能时间触发通信</p>
6	ABOR	<p>自动离线恢复</p> <p>0: 通过软件手动地从离线状态恢复</p> <p>1: 通过硬件自动的从离线状态恢复</p>
5	AWU	<p>自动唤醒</p> <p>一旦自动唤醒后，CAN_CTL寄存器的SLPWMOD位将自动被清0。</p> <p>0: 通过软件手动的从睡眠工作模式唤醒</p> <p>1: 通过硬件自动的从睡眠工作模式唤醒</p>

4	ARD	自动重发禁止 0: 使能自动重发 1: 禁用自动重发
3	RFOD	禁用接收FIFO满时覆盖 0: 使能接收FIFO满时覆盖。当接收FIFO满时, FIFO中的数据被新来的数据覆盖 1: 禁用接收FIFO满时覆盖。当接收FIFO满时, 新来的数据被丢弃, FIFO中的数据保持不变, 不会被覆盖
2	TFO	发送FIFO顺序 0: 标识符 (Identifier) 较小的帧先发送 1: 所有等待发送的邮箱按照先进先出 (FIFO) 的顺序发送
1	SLPWMOD	睡眠工作模式 如果软件将该位置1, CAN将会在当前发送或接收完成时进入睡眠工作模式。该位可由软件或者硬件清0。如果CAN_CTL寄存器中AWU被置位, 当检测到CAN总线工作时, 该位被清0。 0: 禁用睡眠工作模式 1: 使能睡眠工作模式
0	IWMOD	初始化工作模式 0: 禁用初始化工作模式 1: 使能初始化工作模式

21.4.2. 状态寄存器 (CAN_STAT)

地址偏移: 0x04

复位值: 0x0000 0C02

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		RXL	LASTRX	RS	TS	保留		保留		SLPIF	WUIF	ERRIF	SLPWS	IWS	
	r	r	r	r	r					rc_w1	rc_w1	rc_w1	r	r	

位/位域	名称	描述
31:12	保留	必须保持复位值。
11	RXL	RX引脚电平
10	LASTRX	RX引脚最近一次的采样值
9	RS	接收状态 0: CAN当前不是接收器

		1: CAN当前是接收器
8	TS	<p>发送状态</p> <p>0: CAN当前不是发送器</p> <p>1: CAN当前是发送器</p>
7:5	保留	必须保持复位值。
4	SLPIF	<p>进入睡眠工作模式的状态改变中断标志</p> <p>该位在进入睡眠工作模式时由硬件置位。当CAN不再处于睡眠工作模式时由硬件清零。该位也可以由软件写1清0。</p> <p>0: CAN没有进入睡眠工作模式</p> <p>1: CAN进入睡眠工作模式。如果相应的中断使能位为1，则发生中断</p>
3	WUIF	<p>从睡眠工作模式唤醒的状态改变中断标志</p> <p>该位在睡眠工作模式时检测到CAN总线上的活动时由硬件置位。该位由软件写1清0。</p> <p>0: 没有检测到唤醒信号</p> <p>1: 发现唤醒信号。如果相应的中断使能位为1，则发生中断。</p>
2	ERRIF	<p>错误中断标志</p> <p>该位由以下事件置位。CAN_ERR寄存器中BOERR位和CAN_INTEN寄存器中BOIE位都置位。或CAN_ERR寄存器中PERR位和CAN_INTEN寄存器中PERRIE位都置位。或CAN_ERR寄存器中WERR位和CAN_INTEN寄存器中WERRIE位都置位。或CAN_ERR寄存器中ERRN位域的值不为0且CAN_INTEN寄存器中ERRNIE位置位。该位由软件写1清零。</p> <p>0: 没有错误</p> <p>1: 发生错误。如果相应的中断使能位为1，则发生中断</p>
1	SLPWS	<p>睡眠工作状态</p> <p>将CAN_CTL寄存器中SLPWMOD位置位进入睡眠工作模式后该位由硬件置位。当CAN由正常通信模式切换到睡眠工作模式，需等待当前发送过程或者接收过程完成。当CAN离开睡眠工作模式（清除CAN_CTL寄存器中SLPWMOD位或是在CAN_CTL寄存器中AWU置位时检测到CAN总线上的活动）时，该位由硬件清零。如果由睡眠工作模式切换到正常工作模式，该位在CAN接收到来自总线的连续11个隐性位后被清0。</p> <p>0: CAN没有处于睡眠工作状态</p> <p>1: CAN处于睡眠工作状态</p>
0	IWS	<p>初始化工作状态</p> <p>将CAN_CTL寄存器中IWMOD位置位进入初始化工作模式后该位由硬件置位。当CAN由正常通信模式切换到初始化工作模式，需等待当前发送过程或者接收过程完成。在清除CAN_CTL寄存器中IWMOD位离开初始化模式后，该位由硬件清0。如果由初始化工作模式切换到正常工作模式，该位在CAN接收到来自总线的连续11个隐性位后被清0。</p> <p>0: CAN没有处于初始化工作状态</p> <p>1: CAN处于初始化工作状态</p>

21.4.3. 发送状态寄存器 (CAN_TSTAT)

地址偏移: 0x08

复位值: 0x1C00 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
TMLS2	TMLS1	TMLS0	TME2	TME1	TME0	NUM[1:0]	MST2	保留	MTE2	MAL2	MTFNERR2	MTF2			
r	r	r	r	r	r	r	rs		rc_w1	rc_w1	rc_w1	rc_w1			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MST1	保留		MTE1	MAL1	MTFNERR1	MTF1	MST0	保留	MTE0	MAL0	MTFNERR0	MTF0			
rs			rc_w1	rc_w1	rc_w1	rc_w1	rs		rc_w1	rc_w1	rc_w1	rc_w1			

位/位域	名称	描述
31	TMLS2	在发送FIFO中邮箱2最后发送 该位为1表明，当有2个及其以上帧等待发送时，发送邮箱2具有最后的发送顺序。
30	TMLS1	在发送FIFO中邮箱1最后发送 该位为1表明，当有2个及其以上帧等待发送时，发送邮箱1具有最后的发送顺序。
29	TMLS0	在发送FIFO中邮箱0最后发送 该位为1表明，当有2个及其以上帧等待发送时，发送邮箱0具有最后的发送顺序。
28	TME2	发送邮箱2空 0: 发送邮箱2不为空 1: 发送邮箱2空
27	TME1	发送邮箱1空 0: 发送邮箱1不为空 1: 发送邮箱1空
26	TME0	发送邮箱0空 0: 发送邮箱0不为空 1: 发送邮箱0空
25:24	NUM[1:0]	当发送FIFO不满时，NUM表示下一个将要发送的邮箱号。 当发送FIFO满时，NUM表示最后一个将要发送的邮箱号。
23	MST2	邮箱2停止发送 将其置1，将停止邮箱2的发送过程。 当邮箱2变为empty状态时，该位被硬件自动清0。
22:20	保留	必须保持复位值。
19	MTE2	邮箱2发送错误 当发生发送错误时，该位由硬件置1。由软件写1清0或对CAN_TSTAT寄存器中MTF2写1清0。也可以在下一次发送开始时由硬件清0。

		当发生错误时该位被置1。
18	MAL2	<p>邮箱2仲裁失败</p> <p>当发生发送错误时，该位由硬件置1。由软件写1清0或对CAN_TSTAT寄存器中MTF2写1清0。也可以在下一次发送开始时由硬件清0。</p>
17	MTFNERR2	<p>邮箱2无错发送完成</p> <p>当发送结束并且没有错误产生时，该位由硬件置1。由软件写1清0或对CAN_TSTAT寄存器中MTF2写1清0。也可以在无错传输结束时由硬件清0。</p> <p>0：传输结束时发生了错误 1：传输结束且没有错误</p>
16	MTF2	<p>邮箱2发送完成</p> <p>当发送完成或被中止时，该位由硬件置1。由软件写1清0，或当CAN_TIM2寄存器的TEN被置位时清0。</p> <p>0：发送邮箱2正在发送 1：发送邮箱2完成发送</p>
15	MST1	<p>邮箱1停止发送</p> <p>将其置1，将停止邮箱1的发送过程。</p> <p>当邮箱1变为empty状态时，该位被硬件自动清0。</p>
14:12	保留	必须保持复位值。
11	MTE1	<p>邮箱1发送错误</p> <p>当发生发送错误时，该位由硬件置1。由软件写1清0或对CAN_TSTAT寄存器中MTF1写1清0。也可以在下一次发送开始时由硬件清0。</p> <p>当发生错误时该位被置1。</p>
10	MAL1	<p>邮箱1仲裁失败</p> <p>当发生发送错误时，该位由硬件置1。由软件写1清0或对CAN_TSTAT寄存器中MTF1写1清0。也可以在下一次发送开始时由硬件清0。</p> <p>当发生仲裁失败时该位被置1。</p>
9	MTFNERR1	<p>邮箱1无错发送完成</p> <p>当发送结束并且没有错误产生时，该位由硬件置1。由软件写1清0或对CAN_TSTAT寄存器中MTF1写1清0。也可以在无错传输结束时由硬件清0。</p> <p>0：传输结束时发生了错误 1：传输结束且没有错误</p>
8	MTF1	<p>邮箱1发送完成</p> <p>当发送完成或被中止时，该位由硬件置1。由软件写1清0，或当CAN_TIM1寄存器的TEN被置位时清0。</p> <p>0：发送邮箱1正在发送 1：发送邮箱1完成发送</p>
7	MST0	<p>邮箱0停止发送</p> <p>将其置1，将停止邮箱0的发送过程。</p> <p>当邮箱0变为empty状态时，该位被硬件自动清0。</p>

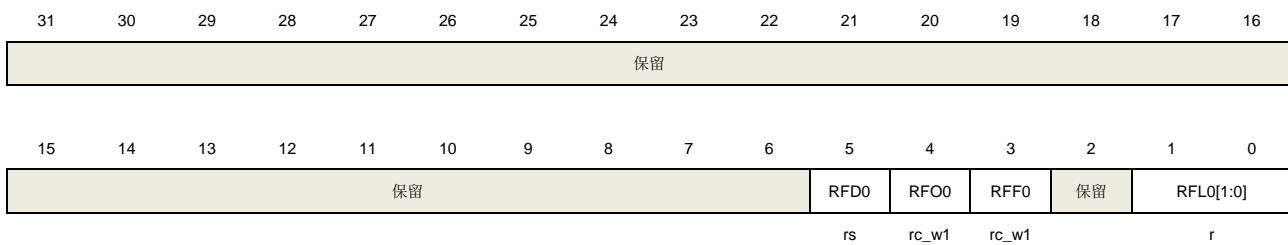
6:4	保留	必须保持复位值。
3	MTE0	<p>邮箱0发送错误</p> <p>当发生发送错误时，该位由硬件置1。由软件写1清0或对CAN_TSTAT寄存器中MTF0写1清0。也可以在下一次发送开始时由硬件清0。</p> <p>当发生错误时该位被置1。</p>
2	MAL0	<p>邮箱0仲裁失败</p> <p>当发生发送错误时，该位由硬件置1。由软件写1清0或对CAN_TSTAT寄存器中MTF0写1清0。也可以在下一次发送开始时由硬件清0。</p> <p>当发生仲裁失败时该位被置1。</p>
1	MTFNERR0	<p>邮箱0无错发送完成</p> <p>当发送结束并且没有错误产生时，该位由硬件置1。由软件写1清0或对CAN_TSTAT寄存器中MTF0写1清0。也可以在无错传输结束时由硬件清0。</p> <p>0：传输结束时发生了错误</p> <p>1：传输结束且没有错误</p>
0	MTF0	<p>邮箱0发送完成</p> <p>当发送完成或被中止时，该位由硬件置1。由软件写1清0，或当CAN_TIM0寄存器的TEN被置位时清0。</p> <p>0：发送邮箱0正在发送</p> <p>1：发送邮箱0完成发送</p>

21.4.4. 接收 FIFO0 寄存器 (CAN_RFIFO0)

地址偏移: 0x0C

复位值: 0x0000 0000

该寄存器只能按字(32位)访问



位/位域	名称	描述
31:6	保留	必须保持复位值。
5	RFDO	<p>释放一次FIFO0中的数据</p> <p>该位被置1， 将释放FIFO0中的一帧数据。</p> <p>FIFO0释放相应的数据空间后，该位被清0。</p>
4	RFO0	<p>接收FIFO0溢出</p> <p>当接收FIFO0溢出时被置位，由软件写1清0。</p>

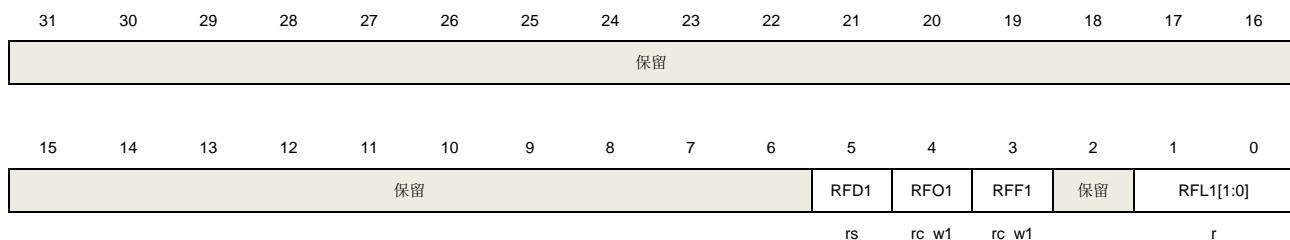
		0: 接收FIFO0没有溢出 1: 接收FIFO0溢出
3	RFF0	接收FIFO0满 当接收FIFO0满时被置位，由软件写1清0。 0: 接收FIFO0不满 1: 接收FIFO0满
2	保留	必须保持复位值。
1:0	RFL0[1:0]	接收FIFO0中帧的数量

21.4.5. 接收 FIFO1 寄存器 (**CAN_RFIFO1**)

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
31:6	保留	必须保持复位值。
5	RFD1	释放一次FIFO1中的数据 该位被置1，将释放FIFO1中的一帧数据。 FIFO1释放相应的数据空间后，该位被清0。
4	RFO1	接收FIFO1溢出 当接收FIFO1溢出时被置位，由软件写1清0。 0: 接收FIFO1没有溢出 1: 接收FIFO1溢出
3	RFF1	接收FIFO1满 当接收FIFO1满时被置位，由软件写1清0。 0: 接收FIFO1不满 1: 接收FIFO1满
2	保留	必须保持复位值。
1:0	RFL1[1:0]	接收FIFO1中帧的数量

21.4.6. 中断使能寄存器 (CAN_INTEN)

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														SLPWIE	WIE
rw														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIE	保留		ERRNIE	BOIE	PERRIE	WERRIE	保留	RFOIE1	RFFIE1	RFNEIE1	RFOIE0	RFFIE0	RFNEIE0	TMEIE	
rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:18	保留	必须保持复位值。
17	SLPWIE	睡眠中断使能 0: 禁用睡眠中断 1: 使能睡眠中断
16	WIE	唤醒中断使能 0: 禁用唤醒中断 1: 使能唤醒中断
15	ERRIE	错误中断使能 0: 禁用错误中断 1: 使能错误中断
14:12	保留	必须保持复位值。
11	ERRNIE	错误种类中断使能 0: 禁用错误种类中断 1: 使能错误种类中断
10	BOIE	离线中断使能 0: 禁用离线中断 1: 使能离线中断
9	PERRIE	被动错误中断使能 0: 禁用被动错误 1: 使能被动错误
8	WERRIE	警告错误中断使能 0: 禁用警告错误中断 1: 使能警告错误中断
7	保留	必须保持复位值。

6	RFOIE1	接收FIFO1溢出中断使能 0: 禁用接收FIFO1溢出中断 1: 使能接收FIFO1溢出中断
5	RFIE1	接收FIFO1满中断使能 0: 禁用接收FIFO1满中断 1: 使能接收FIFO1满中断
4	RFNEIE1	接收FIFO1非空中断使能 0: 禁用接收FIFO1非空中断 1: 使能接收FIFO1非空中断
3	RFOIE0	接收FIFO0溢出中断使能 0: 禁用接收FIFO0溢出中断 1: 使能接收FIFO0溢出中断
2	RFIE0	接收FIFO0满中断使能 0: 禁用接收FIFO0满中断 1: 使能接收FIFO0满中断
1	RFNEIE0	接收FIFO0非空中断使能 0: 禁用接收FIFO0非空中断 1: 使能接收FIFO0非空中断
0	TMEIE	发送邮箱空中断使能 0: 禁用发送邮箱空中断 1: 使能发送邮箱空中断

21.4.7. 错误寄存器 (CAN_ERR)

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
31:24	RECNT[7:0]	接收错误计数值
23:16	TECNT[7:0]	发送错误计数值
15:7	保留	必须保持复位值。

6:4	ERRN[2:0]	错误种类
		ERRN由硬件更新，可以反映位传输过程中的错误情况。当位传输成功没有错误时，ERRN为0。软件可以设置ERRN为0b111。
	000:	无错误
	001:	填充错误
	010:	格式错误
	011:	ACK错误
	100:	位隐性错
	101:	位显性错误
	110:	CRC错误
	111:	软件设置值
3	保留	必须保持复位值。
2	BOERR	离线错误 当TEC 上溢（超过 255）时，CAN总线控制器进入离线状态，该位被置1。
1	PERR	被动错误 当TECNT或者RECNT大于127时，该位由硬件置1。
0	WERR	警告错误 当TECNT或RECNT大于等于96时，该位由硬件置1。

21.4.8. 位时序寄存器 (CAN_BT)

地址偏移: 0x1C

复位值: 0x0123 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
SCMOD	LCMOD	保留	SJW[4:0]				保留	BS2[2:0]			BS1[3:0]				
rw	rw		rw					rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	BS2[4:3]		BS1[6:4]			BAUDPSC[9:0]									
	rw		rw			rw									

位/位域	名称	描述
31	SCMOD	静默通信模式 0: 禁用静默通信模式 1: 使能静默通信模式
30	LCMOD	回环通信模式 0: 禁用回环通信模式 1: 使能回环通信模式
29	保留	必须保持复位值。

28:24	SJW[4:0]	再同步补偿宽度 再同步补偿占用的时间单元数量= SJW[4:0]+1
23	保留	必须保持复位值。
22:20	BS2[2:0]	位段2的2:0位 位段2占用的时间单元数量=BS2[4:0]+1
19:16	BS1[3:0]	位段1的3:0位 位段1占用的时间单元数量=BS1[6:0]+1
15:10	保留	必须保持复位值。
14:13	BS2[4:3]	位段2的4:3位 参考CAN_BT寄存器的22:20位 (FDEN为1时配置有效)
12:10	BS1[6:4]	位段1的6:4位 参考CAN_BT寄存器的19:16位 (FDEN为1时配置有效)
9:0	BAUDPSC[9:0]	波特率分频系数

21.4.9. FD 控制寄存器 (CAN_FDCTL)

地址偏移: 0x20

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

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

位/位域	名称	描述
31:7	保留	必须保持复位值。
6	ESIMOD	错误状态指示模式 0: 发送帧的ESI位始终显示节点错误状态。主动错误状态节点发送显性位, 被动错误状态节点发送隐性位 1: 当节点处于主动错误状态时, 通过CAN_TMPx寄存器的ESI位值决定发送帧的ESI位显示的错误状态。当节点处于被动错误状态时, 发送帧该位发送隐性位
5	TDCMOD	传输延迟补偿模式 0: 测量值加软件偏移 1: 仅使用软件偏移
4	TDCEN	传输延迟补偿使能 0: 传输延迟补偿禁止

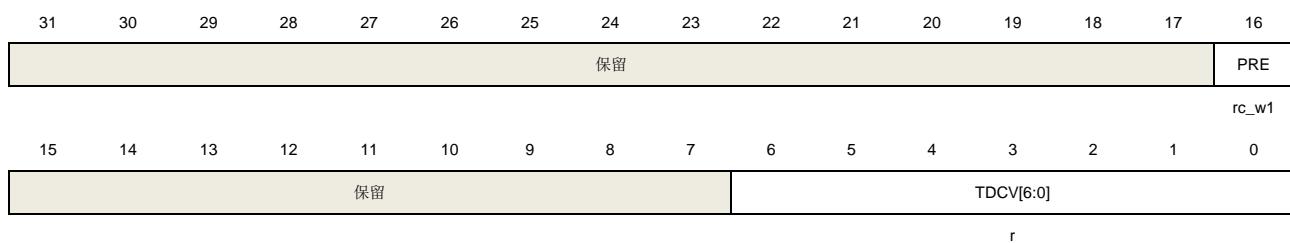
		1: 传输延迟补偿使能
3	NSIO	ISO/BOSCH 0: ISO 1: BOSCH
2	PRED	协议异常事件检测禁用 0: 协议异常事件检测使能（当接收到隐性保留位时转为IDLE状态） 1: 协议异常事件检测禁用（当接收到隐性保留位时，将此视为帧格式错误）
1	保留	必须保持复位值。
0	FDEN	FD功能使能 0: CAN-FD功能禁止 1: CAN-FD功能使能

21.4.10. FD 状态寄存器 (CAN_FDSTAT)

地址偏移: 0x24

复位值: 0x0000 0000

该寄存器只能按字(32位)访问



位/位域	名称	描述
31:17	保留	必须保持复位值。
16	PRE	协议异常事件 当协议异常事件被检测到时该位由硬件置1，软件写1可清0该位。
15:7	保留	必须保持复位值。
6:0	TDCV[6:0]	传输延迟补偿值 这些位由硬件控制，显示硬件自动计算的传输延迟补偿值。

21.4.11. FD 传输延迟补偿寄存器 (CAN_FDTDC)

地址偏移: 0x28

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	TDCO[6:0]								保留	TDCF[6:0]							
rw								rw									

位/位域	名称	描述
31:15	保留	必须保持复位值。
14:8	TDCO[6:0]	传输延迟补偿偏移 这些位被用于设置次级采样点（SSP）基于测量的补偿时间的偏移，测量的补偿时间是由硬件计算信号从CAN_TX发出到从CAN_RX接收到的过程的延迟时间得出的。
7	保留	必须保持复位值。
6:0	TDCF[6:0]	传输延迟补偿过滤器 这些位定义了次级采样点（SSP）的最小值。当配置的TDCF值大于TDCV时，这些位的值用于替代TDCV，以过滤脉冲毛刺引起的测量误差导致的SSP采样点过早的情况发生。

21.4.12. 数据位时序寄存器 (CAN_DBT)

地址偏移: 0x2C

复位值: 0x0123 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				DSJW[2:0]				保留	DBS2[2:0]				DBS1[3:0]		
rw								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								DBAUDPSC[9:0]							
rw															

位/位域	名称	描述
31:27	保留	必须保持复位值。
26:24	DSJW[2:0]	再同步补偿宽度 再同步补偿占用的时间单元数量= SJW[2:0]+1
23	保留	必须保持复位值。
22:20	DBS2[2:0]	位段2的2:0位 位段2占用的时间单元数量=BS2[2:0]+1
19:16	DBS1[3:0]	位段1的3:0位 位段1占用的时间单元数量=BS1[3:0]+1

15:10 保留 必须保持复位值。

9:0 DBAUDPSC[9:0] 波特率分频系数

21.4.13. 发送邮箱标识符寄存器 (CAN_TMI x) ($x=0..2$)

地址偏移: 0x180, 0x190, 0x1A0

复位值: 0xFFFF XXXX (bit0=0)

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
SFID[10:0]/EFID[28:18]								EFID[17:13]							
rw								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EFID[12:0]								rw							

位/位域	名称	描述
31:21	SFID[10:0]/EFID[28:1]	标识符 SFID[10:0]: 标准格式帧标识符 EFID[28:18]: 扩展格式帧标识符
20:16	EFID[17:13]	标识符 EFID[17:13]: 扩展格式帧标识符
15:3	EFID[12:0]	标识符 EFID[12:0]: 扩展格式帧标识符
2	FF	帧格式 0: 标准格式帧 1: 扩展格式帧
1	FT	帧种类 0: 数据帧 1: 遥控帧
0	TEN	发送使能 当应用程序想要发送数据时，该位被置1将启动发送过程。当发送结束，发送邮箱为空时，该位由硬件清0。 0: 禁用发送 1: 使能发送

21.4.14. 发送邮箱属性寄存器 (CAN_TMP x) ($x=0..2$)

地址偏移: 0x184, 0x194, 0x1A4

复位值: 0xFFFF 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
TS[15:0]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					TSEN	FDF	保留	BRS	ESI	DLEN[3:0]					
					rw	rw		rw	rw						

位/位域	名称	描述
31:16	TS[15:0]	时间戳 发送时间戳
15:9	保留	必须保持复位值。
8	TSEN	时间戳使能 0: 禁用时间戳 1: 使能时间戳。时间戳TS[15:0]将放在寄存器CAN_TMDATA1的DATA6和DATA7中只有当寄存器CAN_CTL中的TTC为1时，该位才有效。
7	FDF	CAN-FD帧标志位 0: 常规帧（标准帧或扩展帧） 1: CAN-FD帧
6	保留	必须保持复位值。
5	BRS	位速率转换开关 0: 不转换速率 1: 转换可变速率
4	ESI	错误状态指示 仅当CAN_FDCTL寄存器的ESIMOD位为1时，该位有效 0: 发送帧中的ESI位将发送显性位 1: 发送帧中的ESI位将发送隐性位
3:0	DLEN[3:0]	数据长度，DLEN[3:0]表示帧内数据长度。

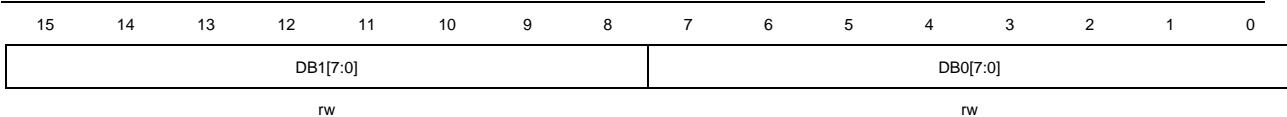
21.4.15. 发送邮箱 data0 寄存器 (CAN_TMDATA0x) (x=0..2)

地址偏移: 0x188, 0x198, 0x1A8

复位值: 0xFFFF 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
DB3[7:0]								DB2[7:0]							
rw								rw							



位/位域	名称	描述
31:24	DB3[7:0]	字节3
23:16	DB2[7:0]	字节2
15:8	DB1[7:0]	字节1
7:0	DB0[7:0]	字节0

21.4.16. 发送邮箱 data1 寄存器 (CAN_TMDATA1x) (x=0..2)

地址偏移: 0x18C, 0x19C, 0x1AC

复位值: 0xFFFF XXXX

该寄存器只能按字(32位)访问



位/位域	名称	描述
31:24	DB7[7:0]	字节7
23:16	DB6[7:0]	字节6
15:8	DB5[7:0]	字节5
7:0	DB4[7:0]	字节4

21.4.17. 接收 FIFO 邮箱标识符寄存器 (CAN_RFIFO1x) (x=0,1)

地址偏移: 0x1B0, 0x1C0

复位值: 0xFFFF XXXX

该寄存器只能按字(32位)访问



位	区域	说明
31:21	SFID[10:0]/EFID[28:1]	标识符 SFID[10:0]: 标准格式帧标识符 EFID[28:18]: 扩展格式帧标识符
20:16	EFID[17:13]	标识符 EFID[17:13]: 扩展格式帧标识符
15:3	EFID[12:0]	标识符 EFID[12:0]: 扩展格式帧标识符
2	FF	帧格式 0: 标准格式帧 1: 扩展格式帧
1	FT	帧种类 0: 数据帧 1: 遥控帧
0	保留	必须保持复位值。

21.4.18. 接收 FIFO 邮箱属性寄存器 (CAN_RFIFOMPx) (x=0,1)

地址偏移: 0x1B4, 0x1C4

复位值: 0xFFFF XXXX

该寄存器只能按字(32位)访问



位/位域	名称	描述
31:16	TS[15:0]	时间戳 接收时间戳
15:8	FI[7:0]	过滤索引 帧通过过滤器时的过滤序号
7	FDF	CAN-FD帧标志位 0: 常规帧 (标准帧或扩展帧) 1: CAN-FD帧

6	保留	必须保持复位值。
5	BRS	位速率转换开关 0: 不转换速率 1: 转换可变速率
4	ESI	错误状态指示 0: 接收帧中的ESI位为显性位 1: 接收帧中的ESI位为隐性位
3:0	DLENC[3:0]	数据长度 DLENC[3:0]表示帧内数据长度。

21.4.19. 接收 FIFO 邮箱 data0 寄存器 (CAN_RFIFOMDATA0x) (x=0,1)

地址偏移: 0x1B8, 0x1C8

复位值: 0xFFFF XXXX

该寄存器只能按字(32位)访问



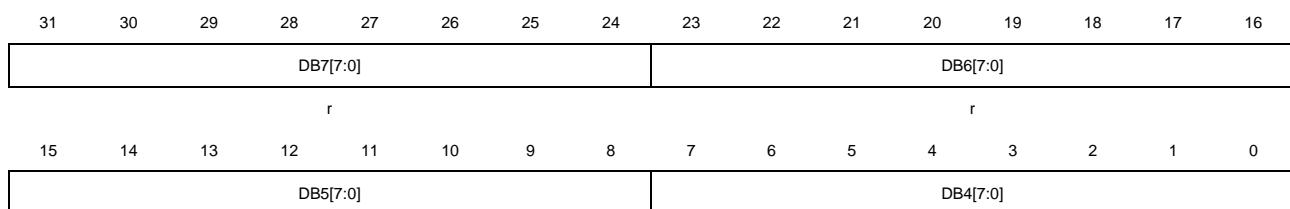
位/位域	名称	描述
31:24	DB3[7:0]	字节3
23:16	DB2[7:0]	字节2
15:8	DB1[7:0]	字节1
7:0	DB0[7:0]	字节0

21.4.20. 接收 FIFO 邮箱 data1 寄存器 (CAN_RFIFOMDATA1x) (x=0,1)

地址偏移: 0x1BC, 0x1CC

复位值: 0xFFFF XXXX

该寄存器只能按字(32位)访问



位/位域	名称	描述
31:24	DB7[7:0]	字节7
23:16	DB6[7:0]	字节6
15:8	DB5[7:0]	字节5
7:0	DB4[7:0]	字节4

21.4.21. 过滤器控制寄存器 (CAN_FCTL)

地址: 0x40006600

复位值: 0x2A1C 0E01

该寄存器只能按字(32位)访问

位/位域	名称	描述
31:14	保留	必须保持复位值。
13:8	HBC1F[5:0]	CAN1过滤器单元起始位置 这些位用来定义CAN1过滤器起始位置。CAN0可以用编号为0~HBC1F-1过滤器，CAN1可以用编号为HBC1F~27过滤器。当这些位的值为0，CAN0将没有过滤器可以使用。当这些位的值为28时，CAN1将没有过滤器可以使用。
7:1	保留	必须保持复位值。
0	FLD	过滤器锁禁用 0: 使能过滤器锁 1: 禁用过滤器锁

21.4.22. 过滤器模式配置寄存器 (CAN_FMCFG)

地址: 0x40006604

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	FMOD27	FMOD26	FMOD25	FMOD24	FMOD23	FMOD22	FMOD21	FMOD20	FMOD19	FMOD18	FMOD17	FMOD16			

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:0	FMODx	过滤器模式 0: 掩码模式 1: 列表模式

21.4.23. 过滤器位宽配置寄存器 (CAN_FSCFG)

地址: 0x4000660C

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:0	FSx	<p>过滤器位宽</p> <p>0: 16-bit位宽</p> <p>1: 32-bit位宽</p>

21.4.24. 过滤器关联 FIFO 寄存器 (CAN_FAFIFO)

地址: 0x40006614

复位值: 0x0000 0000

该寄存器只能按字(32位)访问

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:0	FAFx	过滤器关联FIFO 0: 关联FIFO0 1: 关联FIFO1

21.4.25. 过滤器激活寄存器 (CAN_FW)

地址: 0x4000661C

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				FW27	FW26	FW25	FW24	FW23	FW22	FW21	FW20	FW19	FW18	FW17	FW16
				rw											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FW15	FW14	FW13	FW12	FW11	FW10	FW9	FW8	FW7	FW6	FW5	FW4	FW3	FW2	FW1	FW0
rw															

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:0	FWx	过滤器激活 0: 没有激活 1: 激活工作

21.4.26. 过滤器(x)数据(y)寄存器 (CAN_FxDATAY) (x=0..27, y=0,1)

地址: 0x40006640+8*x+4*y, (x=0..27, y=0,1)

复位值: 0xFFFF 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
FD31	FD30	FD29	FD28	FD27	FD26	FD25	FD24	FD23	FD22	FD21	FD20	FD19	FD18	FD17	FD16
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0
rw															

位/位域	名称	描述
31:0	FDx	过滤器数据 掩码模式下:

0: 标识符的Bit(x)不需参与比较

1: 标识符的Bit(x)需要参与比较

列表模式下:

0: 标识符的Bit(x)必须为0

1: 标识符的Bit(x)必须为1

22. 通用串行总线全速接口（USBFS）

USBFS适用于GD32C10x系列芯片。

22.1. 概述

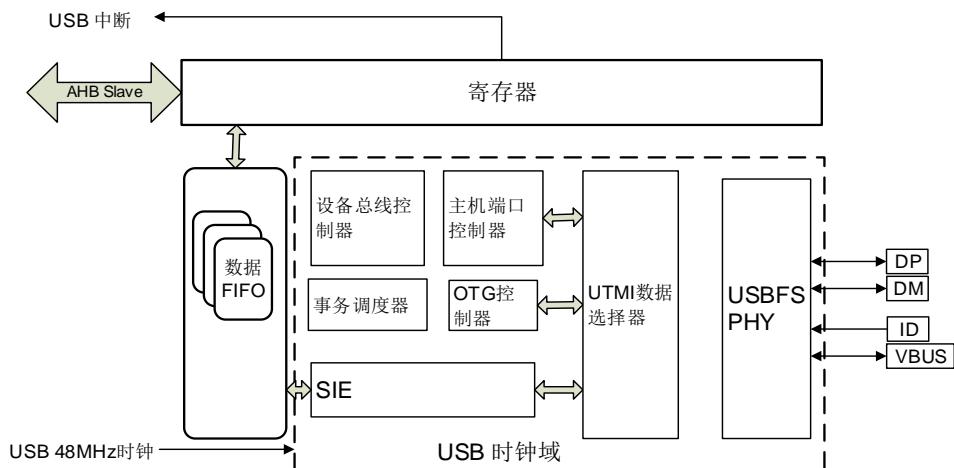
USB全速（USBFS）控制器为便携式设备提供了一套USB互联解决方案。USBFS不仅支持主机模式和设备模式，也支持遵循HNP（主机协商协议）和SRP（会话请求协议）的OTG模式。USBFS包含了一个内部的全速USB PHY，并且不再需要外部PHY芯片。USBFS可以支持USB 2.0协议所定义的所有四种传输方式（控制传输、批量传输、中断传输和同步传输）。

22.2. 主要特征

- 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。

22.3. 结构框图

图 22-1. USBFS 结构框图



22.4. 信号线描述

表 22-1. USBFS 信号线描述

I/O 端口	类型	描述
VBUS	输入/输出	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

22.5. 功能说明

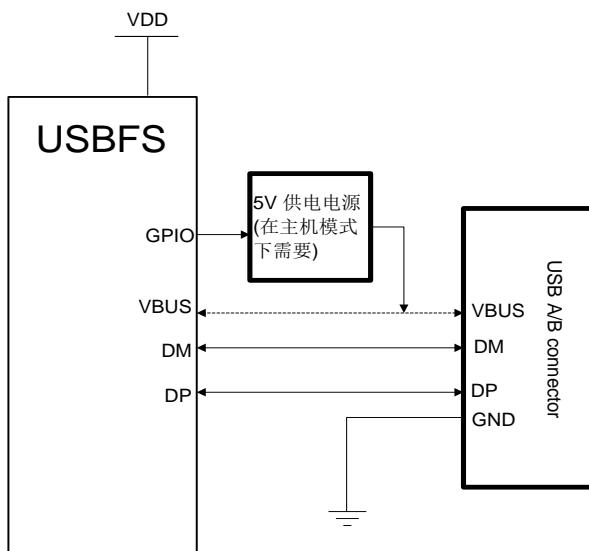
22.5.1. USBFS 时钟及工作模式

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控制。一个利用内部全速PHY的典型连接示意图如[图22-2. 在主机或设备模式下连接示意图](#)所示。

图 22-2. 在主机或设备模式下连接示意图

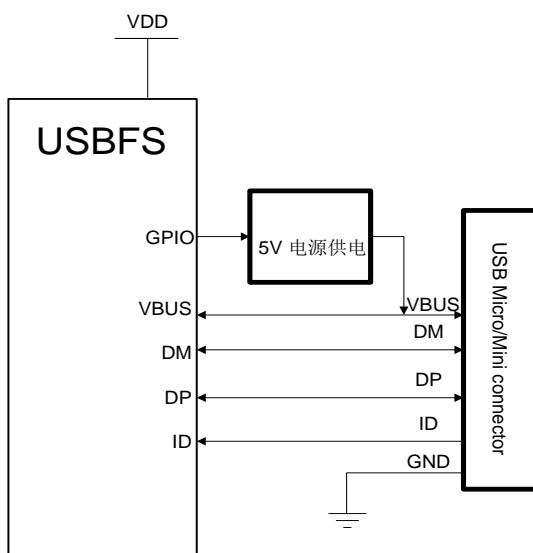


当USBFS工作在主机模式下时（FHM控制位置位、FDM控制位清除），VBUS为USB协议所定义的5V电源检测引脚。内部PHY不能提供5V VBUS电源，仅在VBUS信号线上具有电压比较器和充电、放电电路。所以，如果应用需要提供VBUS电源，那么则需要一个外部的供电电源IC。在主机模式下，USBFS和USB连接器之间的VUBS连接可以被忽略，这是由于USBFS并不检测VBUS引脚的电平状态，并假定5V供电电源一直存在。

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控制位所配置。因此，如果设备不需要检测VBUS引脚电压，可以配置VBUSIG控制位，并可释放VBUS引脚作为其他用途。否则，VBUS引脚的连接不能够被忽略，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，从而产生一个断开状态。

OTG 模式连接示意图如 [图 22-3. 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 USBFS_GCCFG 寄存器的 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 22-3. OTG 模式下连接示意图

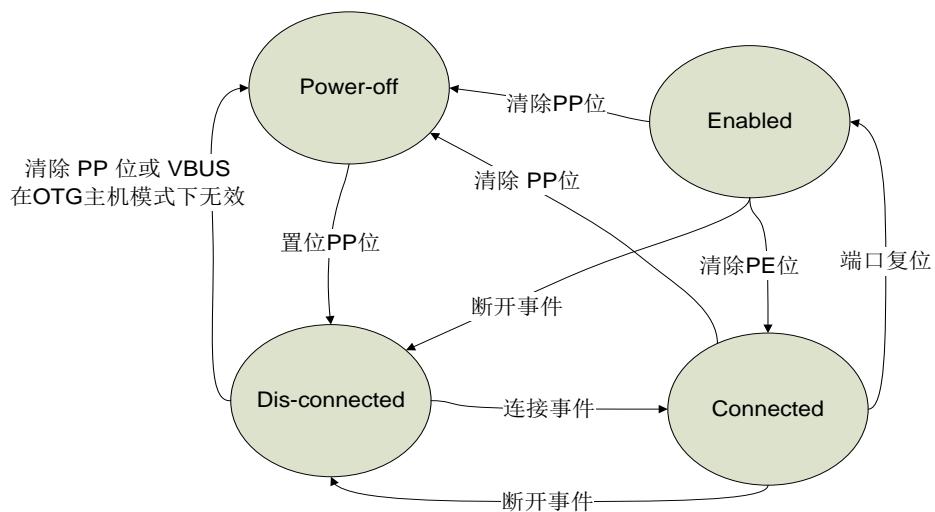


22.5.2. USB 主机功能

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 22-4. 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。在挂起状态下，**USBFS**停止在USB总线上发送SOF，并且这样会让所连接的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，当清除该控制位时，则可以停止恢复序列。如果主机在挂起状态下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所定义的周期发送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位显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。

22.5.3. 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。

22.5.4. 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。

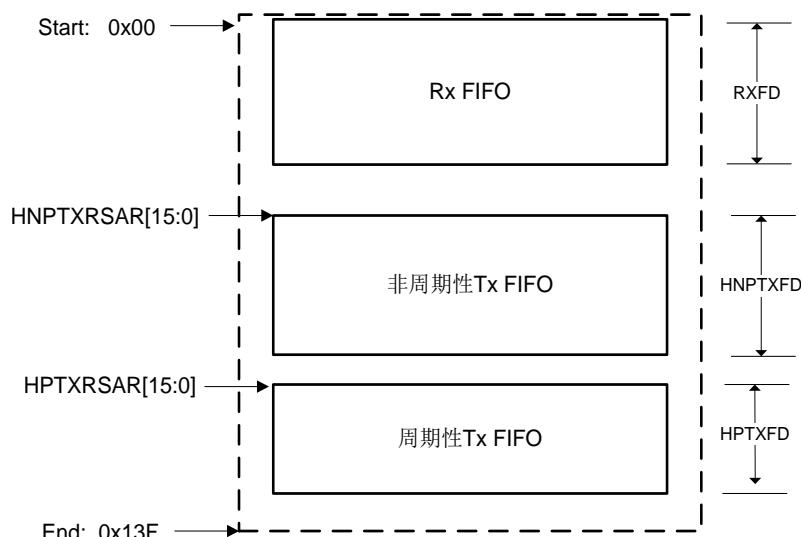
22.5.5. 数据 FIFO

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 22-5. 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 22-5. 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 22-6. 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，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。

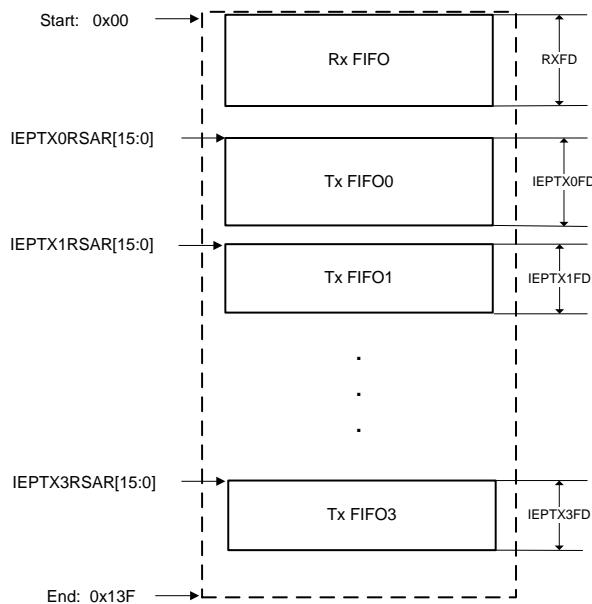
图 22-6. 主机模式 FIFO 访问寄存器映射表



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 22-7. 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 22-7. 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 22-8. 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，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_GRXTATR/USBFS_GRSTATP 来访问 Rx FIFO。

图 22-8. 设备模式 FIFO 访问寄存器映射表

1000h-1FFFFh	IN端点0 FIFO Write
2000h-2FFFFh	IN端点1 FIFO Write
⋮	
4000h-4FFFFh	IN端点3 FIFO Write

22.5.6. 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事务的总线时间足够，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。

22.6. 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 22-2. USBFS 全局中断](#) 中。

表 22-2. 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
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

中断标志	描述	运行模式
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。

22.7. USBFS 寄存器

USBFS 基地址: 0x5000 0000

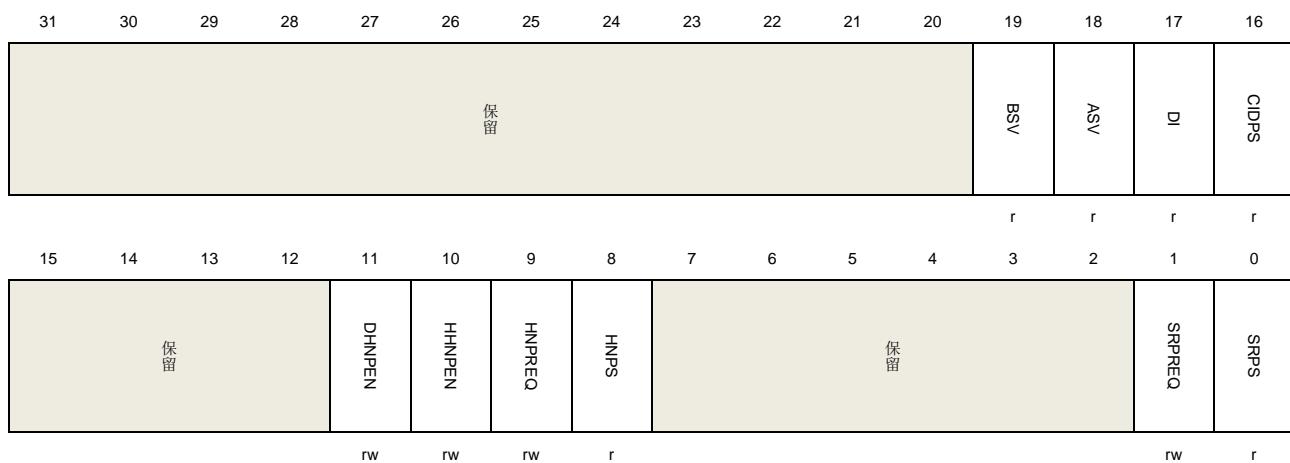
22.7.1. 全局控制与状态寄存器组

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V _{BUS} 电压水平低于V _{BSESSVLD} 1: OTG B设备V _{BUS} 电压水平不低于V _{BSESSVLD} 注意: 仅在OTG 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V _{BUS} 电压水平低于V _{ASESSVLD} 1: OTG A设备V _{BUS} 电压水平不低于V _{ASESSVLD} 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	<p>ID引脚状态</p> <p>连接器ID引脚的电压水平</p> <p>0: USBFS工作在A设备模式</p> <p>1: USBFS工作在B设备模式</p> <p>注意：在设备和主机模式下均可访问</p>
15:12	保留	必须保持复位值。
11	DHNPN	<p>设备HNP使能</p> <p>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。</p> <p>0: HNP功能不使能</p> <p>1: HNP功能使能</p> <p>注意：仅在设备模式下访问</p>
10	HHNPN	<p>主机HNP使能</p> <p>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。</p> <p>0: HNP功能不使能</p> <p>1: HNP功能使能</p> <p>注意：仅在主机模式下访问</p>
9	HNPREQ	<p>HNP请求</p> <p>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。</p> <p>0: 不发送HNP请求</p> <p>1: 发送HNP请求</p> <p>注意：仅在设备模式下访问</p>
8	HNPS	<p>HNP成功标志位</p> <p>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。</p> <p>0: HNP失败</p> <p>1: HNP成功</p> <p>注意：仅在设备模式下访问</p>
7:2	保留	必须保持复位值。
1	SRPREQ	<p>SRP请求</p> <p>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。</p> <p>0: 没有会话请求</p> <p>1: 会话请求</p> <p>注意：仅在设备模式下访问</p>
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SRPREQ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

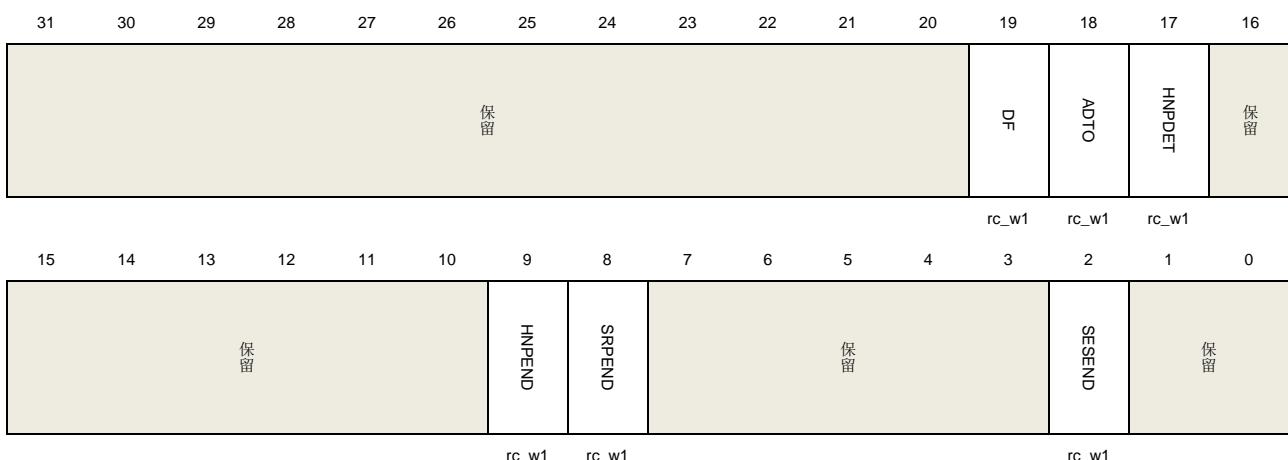
注意：仅在设备模式下访问

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	HNП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。

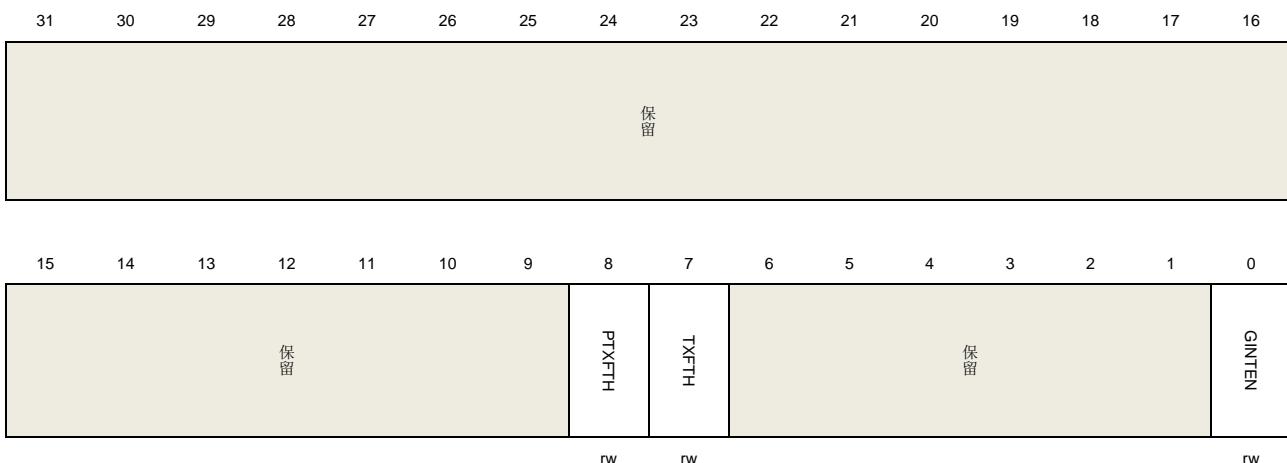
8	SRPEND	SRPEND
		当一个SRP结束时，内核置位该标志位。软件应该读取USBFS_GOTGCS寄存器中SRPS标志位，以获取SRP结果。
		注意： 在设备和主机模式下，均可访问。
7:3	保留	必须保持复位值。
2	SESEND	会话结束 当V _{BUS} 电压低于V _{b_ses_vld} 时，内核置位该标志位。
1:0	保留	必须保留复位值

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

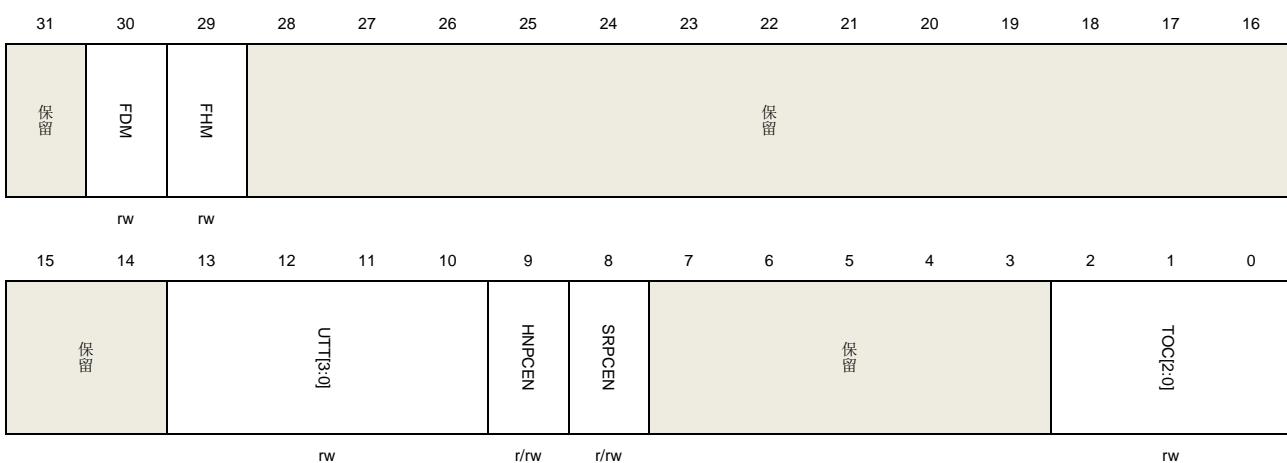
6:1	保留	必须保持复位值。
0	GINTEN	<p>全局中断使能</p> <p>0: 全局中断不使能</p> <p>1: 全局中断使能</p> <p>注意: 在主机和设备模式下, 均可访问</p>

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</p> <p>1: 设备模式</p> <p>设置该控制位后, 应用必须等待至少25ms, 让修改产生作用。</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</p> <p>1: 主机模式</p> <p>设置该控制位后, 应用必须等待至少25ms, 让修改产生作用。</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]	USB运转时间 以物理时钟数来设定运转时间 注意: 仅在设备模式下访问
9	HNP CEN	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	SRP CEN	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

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				TXFNUM[4:0]				RW		RS		RS		RS	
保留				TXFF				RS		RS		RS		RS	
保留				RXFF				RS		RS		RS		RS	
保留				HFORST				RS		RS		RS		RS	
保留				HCSRST				RS		RS		RS		RS	
保留				CSRST				RS		RS		RS		RS	

位/位域	名称	描述
------	----	----

31:11	保留	必须保持复位值。
10:6	TXFNUM[4:0]	<p>Tx FIFO序号</p> <p>当本寄存器中TXFF控制位置位时，该标志位决定那个Tx FIFO会被刷新</p> <p>主机模式：</p> <ul style="list-style-type: none"> 00000: 仅非周期性Tx FIFO被刷新 00001: 仅周期性Tx FIFO被刷新 1xxxx: 周期性和非周期性Tx FIFO均被刷新 其他: 没有数据被刷新 <p>设备模式：</p> <ul style="list-style-type: none"> 00000: 仅Tx FIFO0被刷新 00001: 仅Tx FIFO1被刷新 ... 00011: 仅Tx FIFO3被刷新 1xxxx: 所有的Tx FIFO均被刷新 其他: 没有数据被冲刷
5	TXFF	<p>Tx FIFO刷新控制位</p> <p>应用通过置位该控制位来刷新Tx FIFO数据，并且TXFNUM[4:0]决定刷新的FIFO数目。当刷新完成后，硬件自动清除该控制位。置位该控制位后，应用程序应该等待该控制位清除，并且，在此之前USBFS不应有其他操作。</p> <p>注意：在设备和主机模式下，均可访问</p>
4	RXFF	<p>Rx FIFO刷新控制位</p> <p>应用通过置位该控制位来刷新Rx FIFO数据。当刷新完成后，硬件自动清除该控制位。置位该控制位后，应用程序应该等待该控制位清除，并且，在此之前USBFS不应有其他操作。</p> <p>注意：在设备和主机模式下，均可访问</p>
3	保留	必须保持复位值。
2	HFCRST	<p>主机帧计数器复位</p> <p>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。</p> <p>注意：仅在主机模式下访问</p>
1	HCSRST	<p>HCLK软件复位</p> <p>应用通过置位该控制位来复位AHB时钟域电路</p> <p>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，应用程序应该等待该控制位清除，并且，在此之前USBFS不应有其他操作。</p> <p>注意：在设备和主机模式下，均可访问</p>
0	CSRST	<p>USB内核软件复位</p> <p>复位AHB和USB时钟域电路，以及大多数的寄存器。</p>

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WKUPIF	SESIF	DISCIF	IDPSC	保留	PTXFEIF	HCF	HPIF	保留	保留	PXNCIF/ ISOONCIF	ISONCIF	OEPIF	IEPIF	保留	保留
rc_w1	rc_w1	rc_w1	rc_w1	r	r	r	r	rc_w1	rc_w1	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOPIF	ISOPDIF	ENUMF	RST	SP	ESP	保留	GONAK	GNPINAK	NPTXFEIF	RXNFEIF	SOF	OTGIF	MIF	COPM	rc_w1

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）。
	ISOONCIF	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	IEEIF	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	EOPFIF	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	IDPSIE	保留	PTRXEIE	HCE	HPIE	保留	ISOONCIE	PXNCIE/	ISOONCIE	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	GNPINAKIE	NPTXFEIE	RXFNEIE	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	SESIE	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/弹出寄存器(USBFS_GRSTATR/USBFS_GRSTATP)

读地址偏移: 0x001C

弹出地址偏移: 0x0020

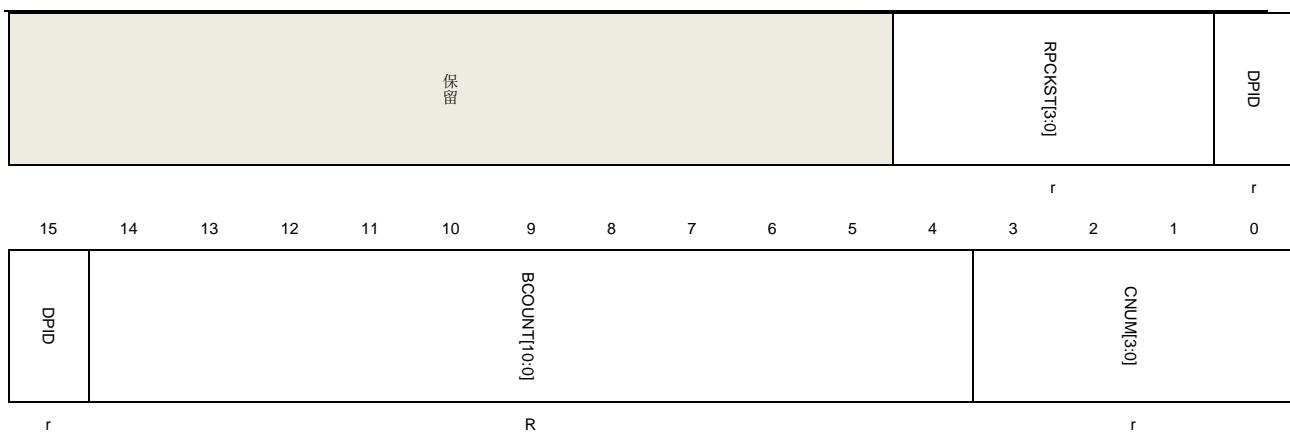
复位值: 0x0000 0000

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。对接收状态弹出寄存器的读操作, 将弹出Rx FIFO的顶部条目。

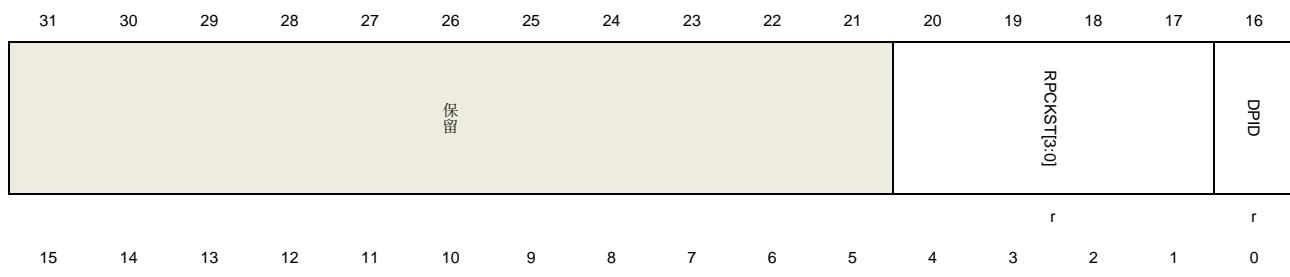
在主机模式和设备模式下, Rx FIFO中的条目具有不同的含义。当全局中断标志寄存器(USBFS_GINTF)中的接收FIFO非空中断标志位(RXFNEIF)置位后, 软件应该读取该寄存器。

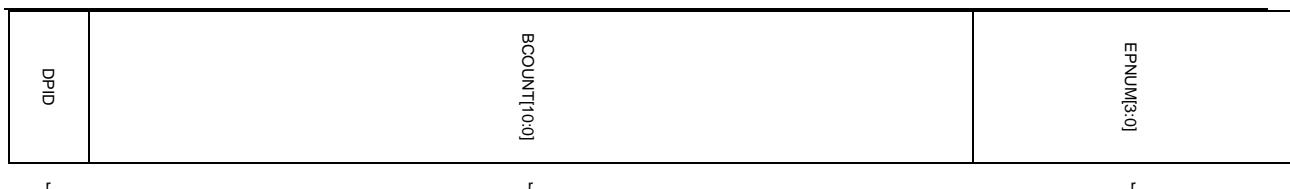
该寄存器只能按字(32位)访问

主机模式:



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	DPID[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:




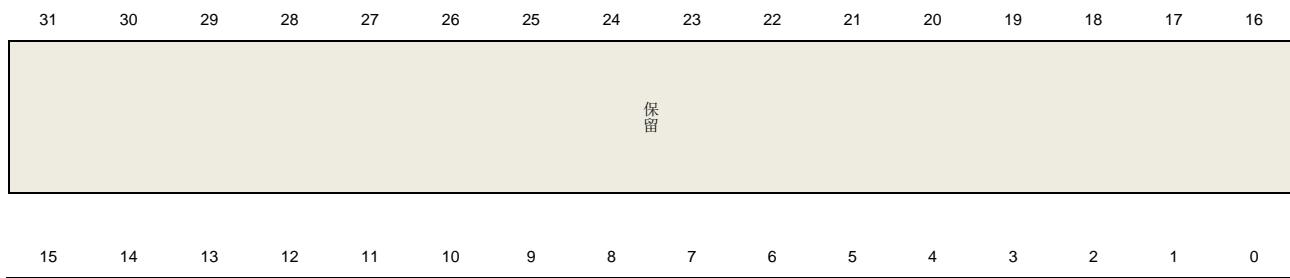
位/位域	名称	描述
31:21	保留	必须保持复位值。
20:17	RPCKST[3:0]	接收包状态 0001: 全局OUT NAK (产生一个中断) 0010: 接收到OUT数据包 0011: OUT传输完成 (产生一个中断) 0100: SETUP传输完成 (产生一个中断) 0110: 接收到SETUP数据包 其他: 保留
16:15	DPID[1:0]	数据PID 接收到OUT数据包的数据PID 00: DATA0 10: DATA1 01: DATA2 11: MDATA
14:4	BCOUNT[10:0]	字节数 接收数据包的字节数
3:0	EPNUM[3:0]	端点号 当前接收包所属端点编号

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





r/rw

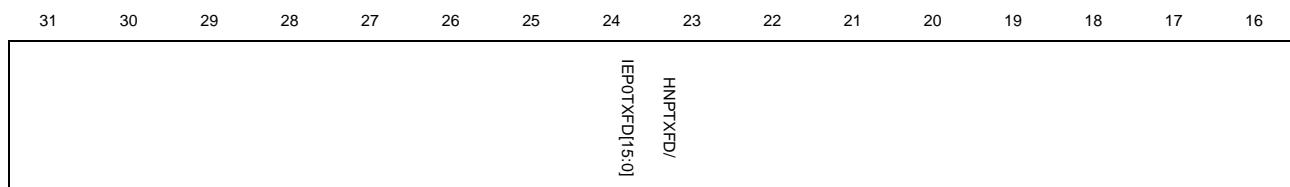
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	RXFD[15:0]	Rx FIFO深度 以32位字计数 $1 \leq RXFD \leq 1024$

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)

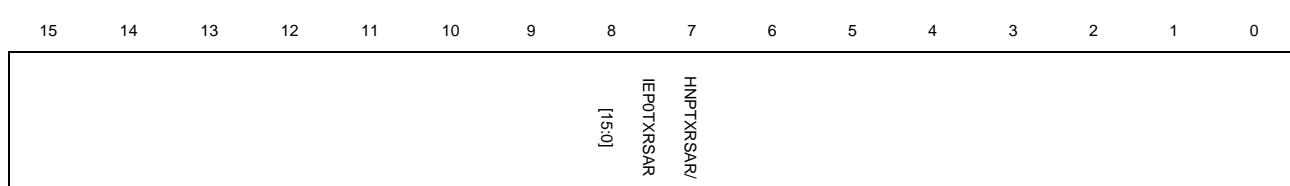
地址偏移: 0x0028

复位值: 0x0200 0200

该寄存器只能按字（32位）访问



r/rw



r/rw

主机模式:

位/位域	名称	描述
31:16	HNPTXFDFD[15:0]	主机非周期性Tx FIFO深度 以32位字计数 $1 \leq HNPTXFDFD \leq 1024$
15:0	HNPTXRSAR[15:0]	主机非周期性Tx FIFO RAM起始地址 非周期性发送FIFO RAM的起始地址以32位字计数

设备模式:

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

$16 \leq IEP0TXFD \leq 140$

15:0 IEP0TXRSAR[15:0] 输入端点0 TX FIFO RAM起始地址
 端点0发送FIFO RAM的起始地址以32位字计数

主机非周期性发送 FIFO/队列状态寄存器 (**USBFS_HNPTFQSTAT**)

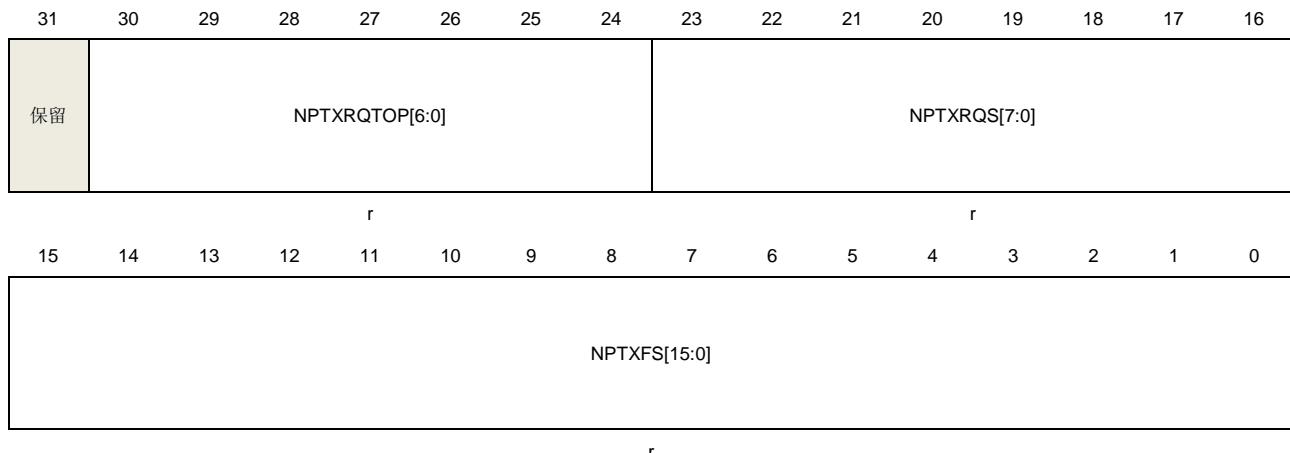
地址偏移: 0x002C

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: - 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: 请求队列满 1: 1个条目

2: 2个条目

...

n: n个条目 (0≤n≤8)

其他: 保留

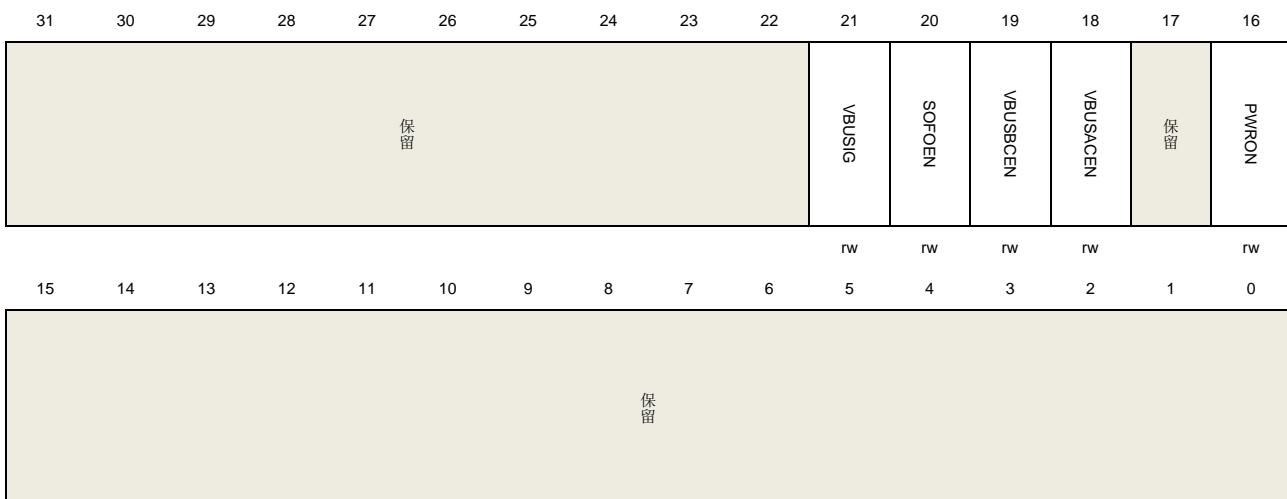
15:0	NPTXFS[15:0]	非周期性Tx FIFO空间 非周期性发送FIFO剩余空间 以32位字计数 0: 非周期性Tx FIFO满 1: 1个字 2: 2个字 ...
		n: n个字(0≤n≤NPTXFD) 其他: 保留

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	VBUS忽略 当该控制位被置位, USBFS并不监测VBUS引脚电压, 并且认为在主机和设备模式下, VBUS电压一直有效, 然后可释放VBUS引脚作为其他用途。 0: VBUS不被忽略 1: VBUS被忽略, 并认为VBUS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。

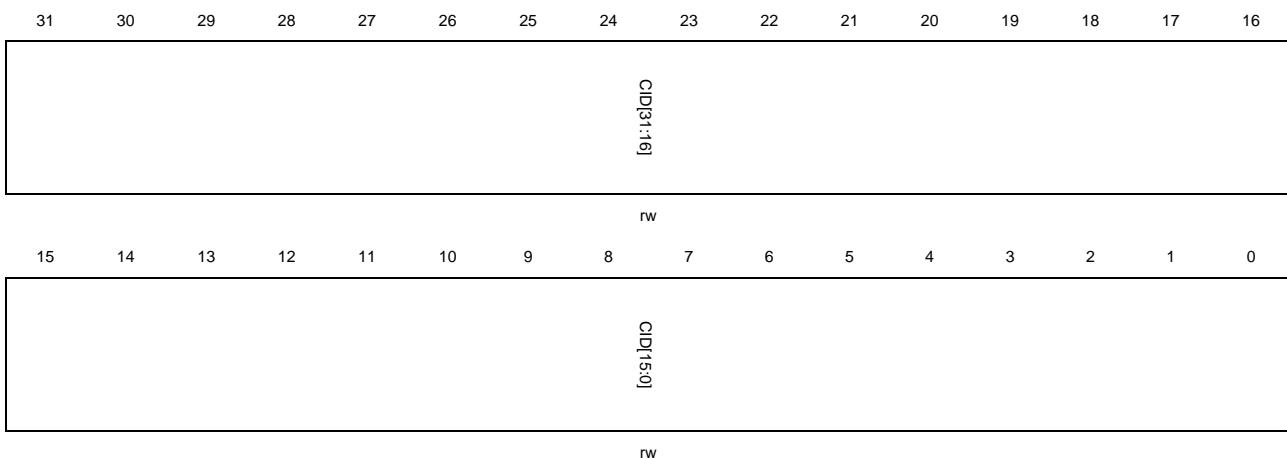
内核 ID 寄存器 (**USBFS_CID**)

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



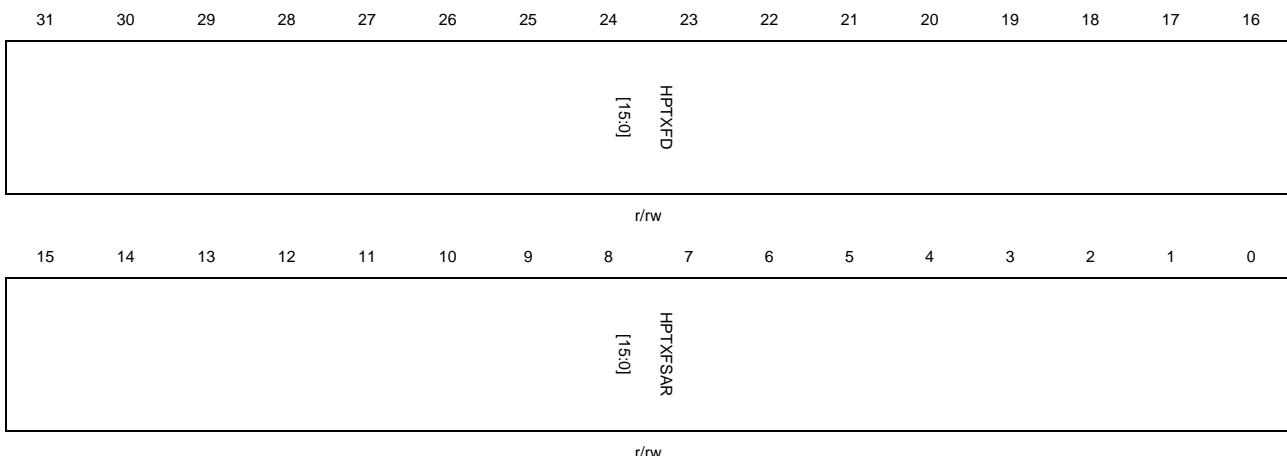
位/位域	名称	描述
31:0	CID[31:0]	内核ID 软件能够写入或读取该域值，并利用该域值为应用产生一个唯一ID。

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



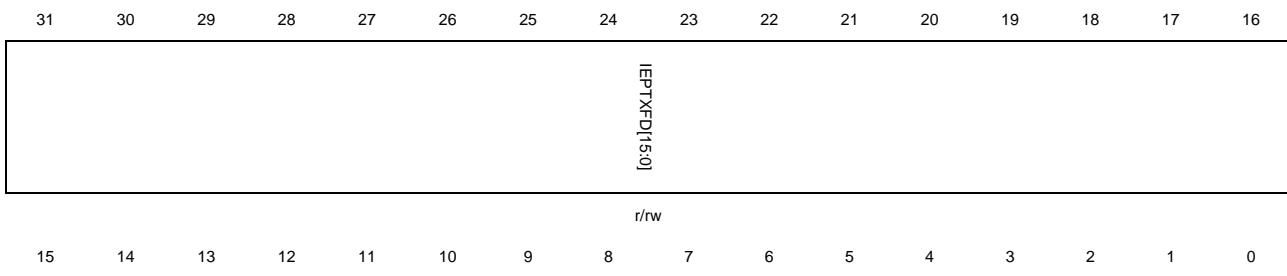
位/位域	名称	描述
31:16	HPTXFD[15:0]	主机周期性Tx FIFO深度 以32位字计数 $1 \leq \text{HPTXFD} \leq 1024$
15:0	HPTXFSAR[15:0]	主机周期性Tx FIFO RAM起始地址 主机周期性发送FIFO RAM起始地址以32位字计数

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 + (FIFO_number - 1) × 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 $1 \leq HPTXFD \leq 1024$
15:0	IEPTXRSAR[15:0]	IN端点Tx FIFO RAM起始地址 以32位字为单位的IN端点发送FIFOx起始地址

22.7.2. 主机控制和状态寄存器

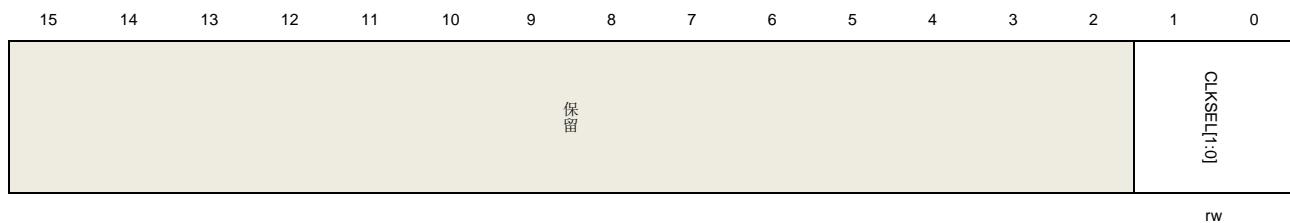
主机控制寄存器 (USBFS_HCTL)

地址偏移: 0x0400

复位值: 0x0000 0000

在主机模式下, 上电后, 该寄存器配置USB内核。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

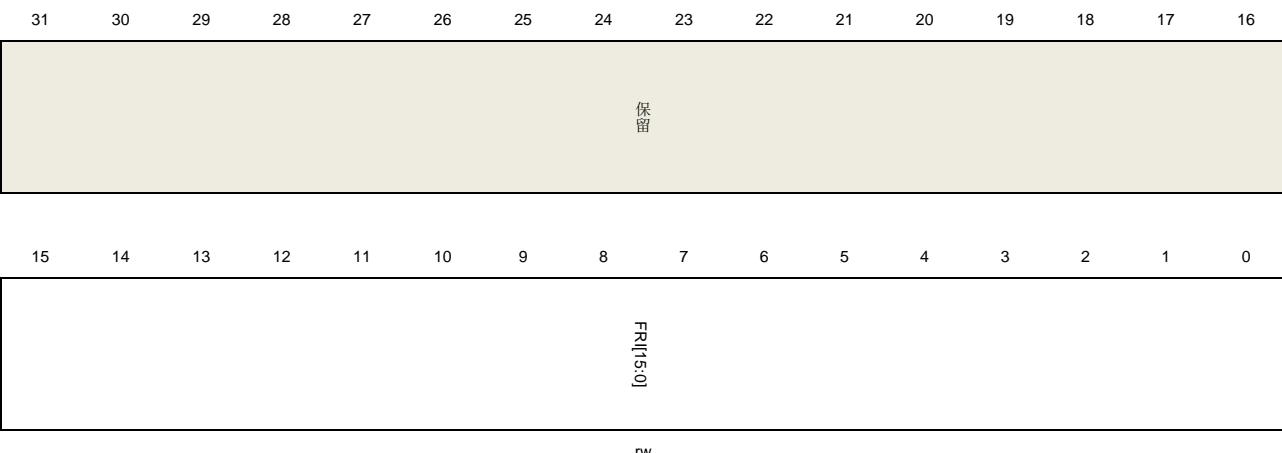
主机帧间隔寄存器 (USBFS_HFT)

地址偏移: 0x0404

复位值: 0x0000 BB80

当USBFS控制器正在枚举USB设备时，该寄存器设置帧间隔。

该寄存器只能按字（32位）访问



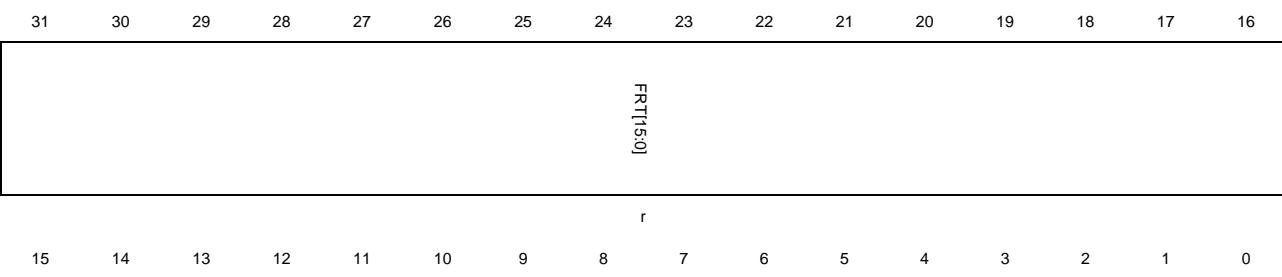
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	FRI[15:0]	<p>帧间隔</p> <p>该值描述了以PHY时钟为单位的帧周期。每次端口复位操作后，端口被使能，USBFS根据当前速度采用一个固有值，并且软件可以向该位域写值以改变该固有值。该值需要采用以下描述的频率来进行计算：</p> <p>全速: 48MHz</p> <p>低速: 6MHz</p>

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。

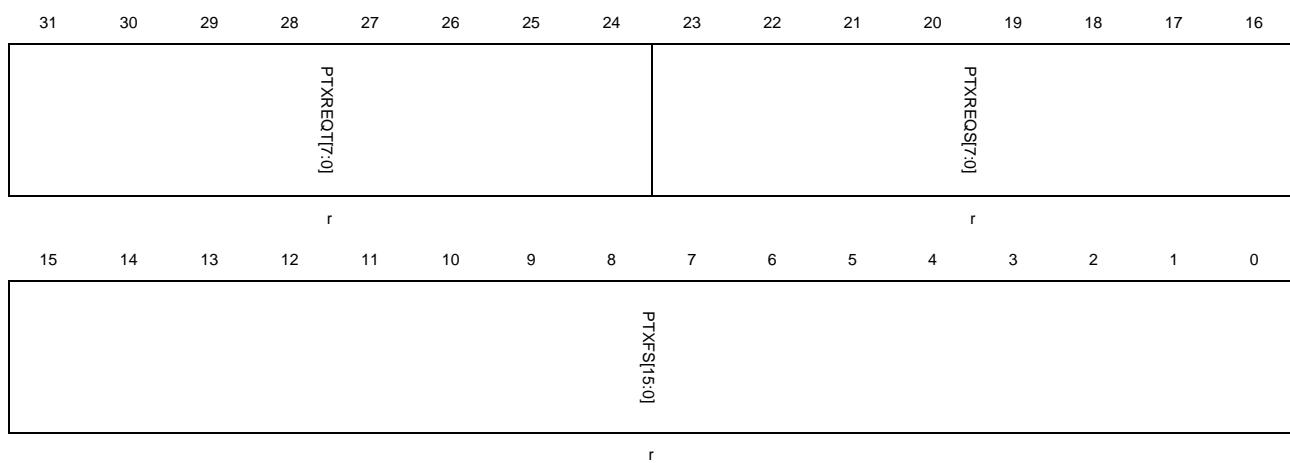
主机周期性发送 FIFO/队列状态寄存器 (USBFS_HPTFQSTAT)

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]	周期性发送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: 请求队列满 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满 1: 1个字 2: 2个字 ... n: n个字 (0≤n≤PTXFD)) 其他: 保留

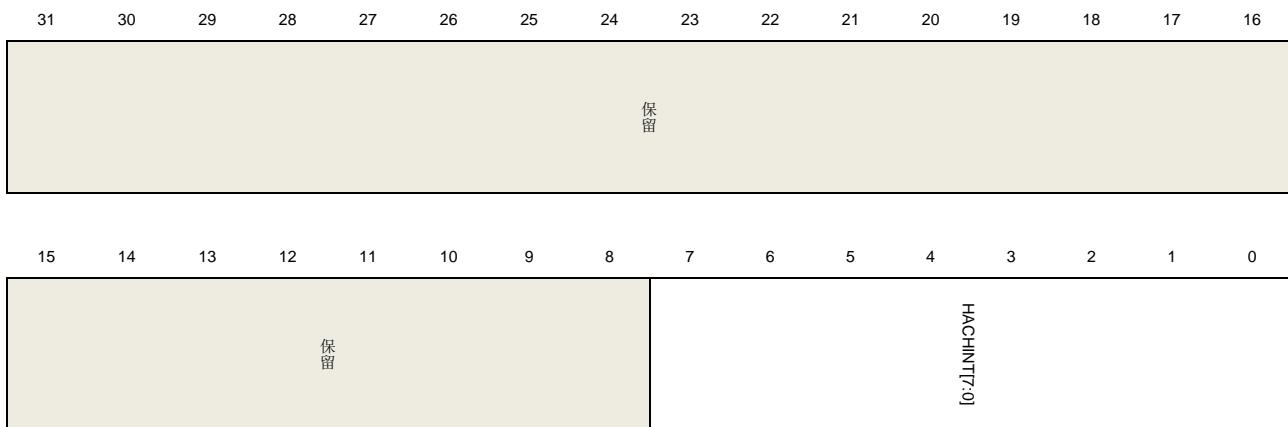
主机所有通道中断寄存器 (USBFS_HACHINT)

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。只有当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HCIF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**)

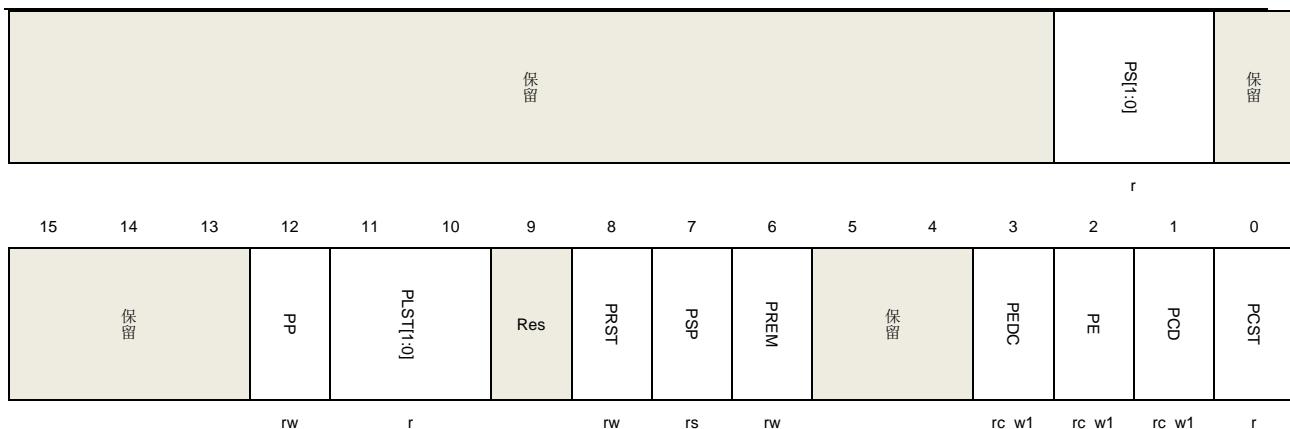
地址偏移: 0x0440

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

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



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, 从而使端口进入挂起状态。当该控制位被置位后, 端口停止发送 SOF 令牌包。该控制位只能够通过以下操作清除。 – 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/禁止变化

当该寄存器中的位2状态变化时，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：

- 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

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

EPDIR	EPNUM[3:0]		MP[10:0]
rw	rw		rw

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），该位控制通道事务是在奇数帧还是偶数帧中处理。 0：偶数帧 1：奇数帧
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
------	-----------	-------------------

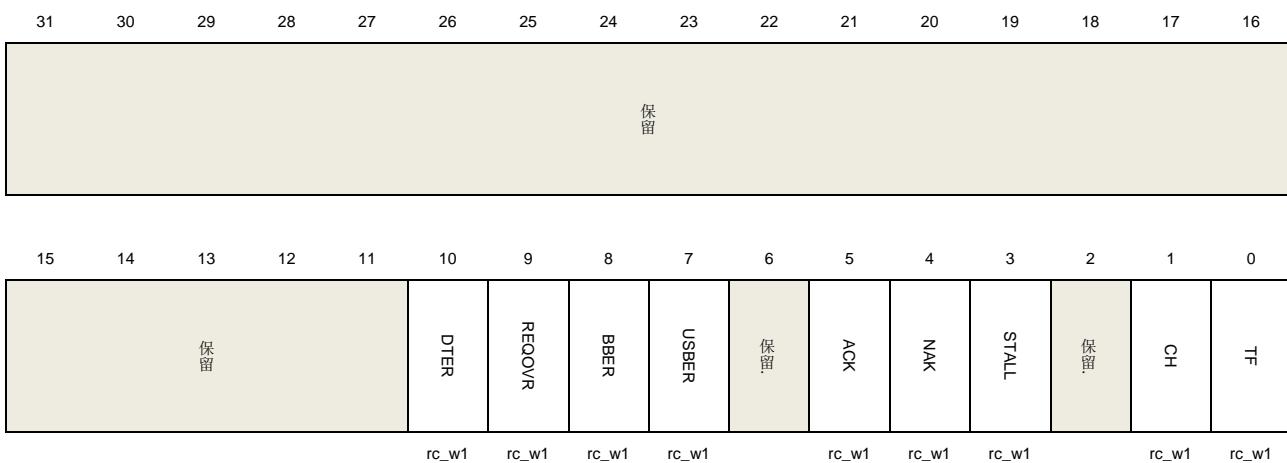
主机通道 x 中断标志寄存器 (**USBFS_HCHxINTF**) (x = 0...7, 其中 x = 通道号)

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，当一个通道中断发生时，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



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 当请求队列已满且出现新的传输时，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 接收到或者发送了一个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。

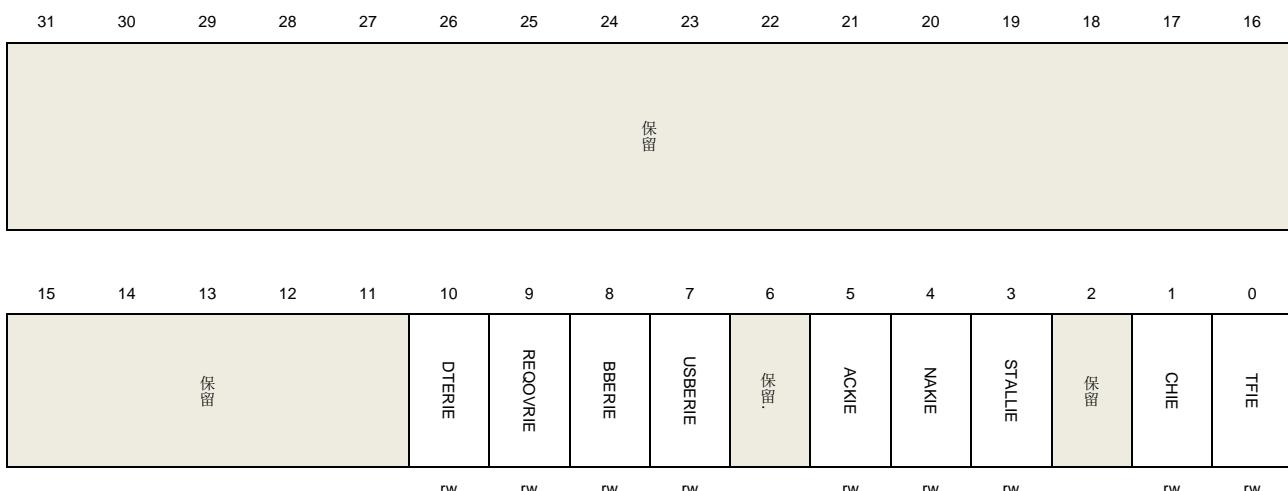
主机通道 x 中断使能寄存器 (**USBFS_HCHxINTEN**) (x = 0...7, 其中 x = 通道号)

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

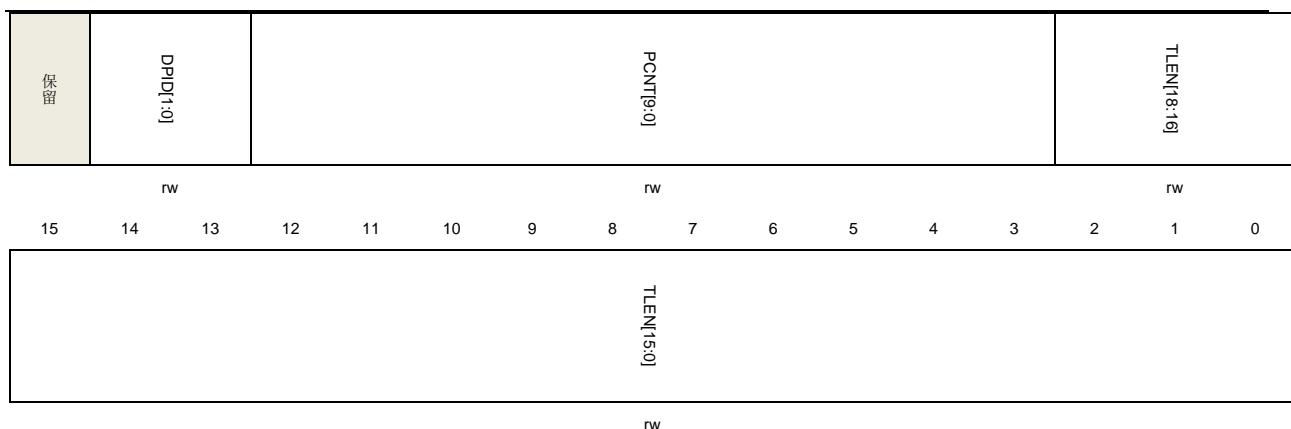
		0: 禁用数据切换错误中断 1: 使能数据切换错误中断
9	REQOVRIE	请求队列上溢中断使能 0: 禁用请求队列上溢中断 1: 使能请求队列上溢中断
8	BBERIE	串扰错误中断使能 0: 禁用串扰错误中断 1: 使能串扰错误中断
7	USBERIE	USB总线错误中断使能 0: 禁用USB总线错误中断 1: 使能USB总线错误中断
6	保留	必须保持复位值。
5	ACKIE	ACK中断使能 0: 禁用ACK中断 1: 使能ACK中断
4	NAKIE	NAK中断使能 0: 禁用NAK中断 1: 使能NAK中断
3	STALLIE	STALL中断使能 0: 禁用STALL中断 1: 使能STALL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]	<p>数据PID</p> <p>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。</p> <p>00: DATA0</p> <p>10: DATA1</p> <p>11: SETUP（仅对于控制传输）</p> <p>01: 保留</p>
28:19	PCNT[9:0]	<p>包计数</p> <p>在一个传输中希望发送（OUT）或接收（IN）的数据包个数。</p> <p>软件应该在通道使能之前写该位域。在传输启动之后，该位域在正确传输每个数据包后，自动减少。</p>
18:0	TLEN[18:0]	<p>传输长度</p> <p>一次传输的总数据字节数。</p> <p>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。当软件正确向通道的数据Tx FIFO中写入一个包时，该位域以包中字节大小进行递减。</p> <p>对于IN传输，每次软件从Rx FIFO中读取一个包后，该位域也以包中字节大小进行减少。</p>

22.7.3. 设备控制和状态寄存器

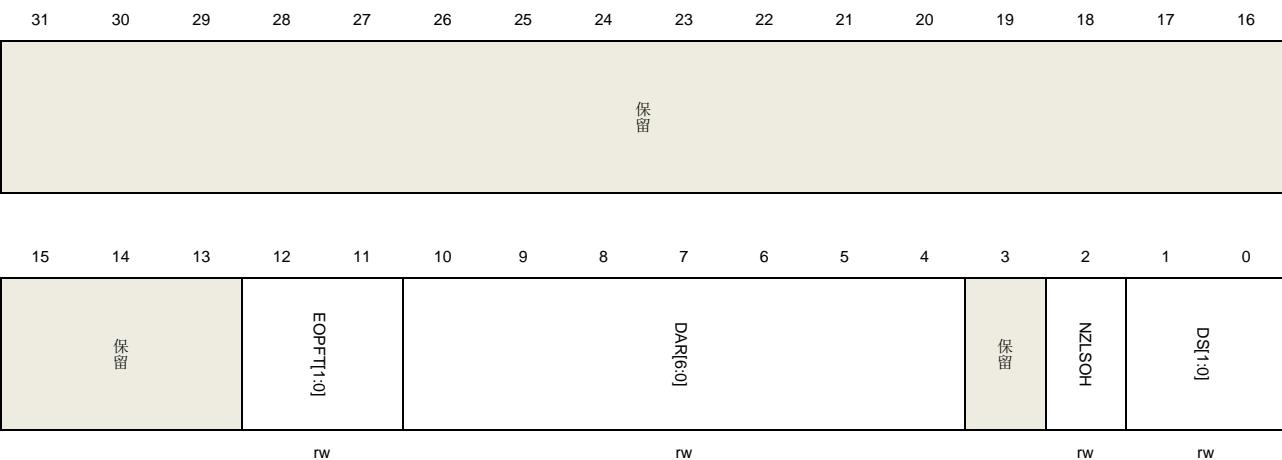
设备配置寄存器 (USBFS_DCFG)

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



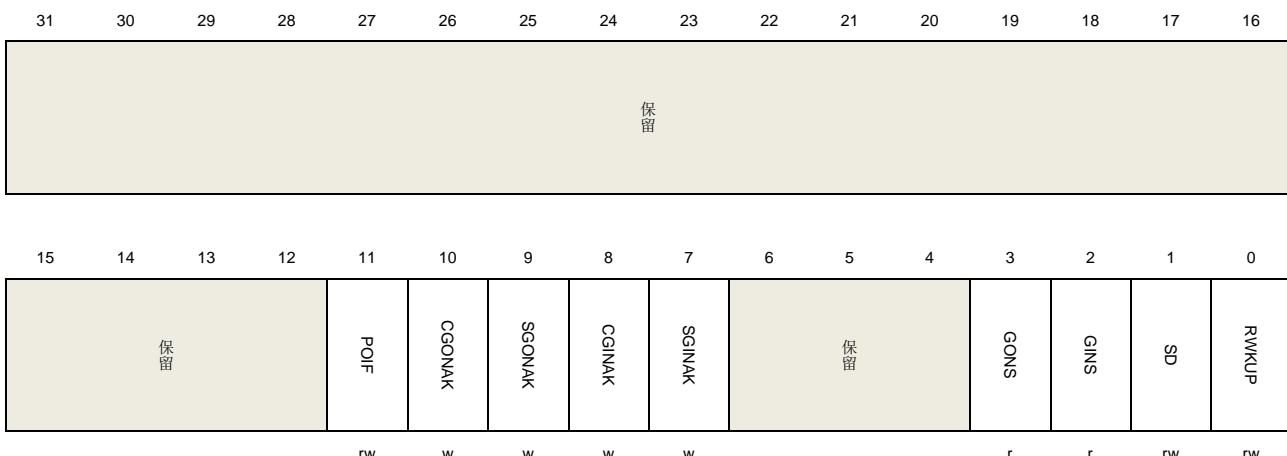
位/位域	名称	描述
31:13	保留	必须保持复位值。
12:11	EOPFT[1:0]	周期性帧尾时间 该域定义周期性帧时间的帧尾标志触发的时间点 00: 80%的帧时间 01: 85%的帧时间 10: 90%的帧时间 11: 95%的帧时间
10:4	DAR[6:0]	设备地址 该位定义USB设备地址，USBFS采用该位匹配接收的设备令牌地址域，在接收到来自主机的设置地址的命令后，软件设置该域。
3	保留	必须保持复位值。
2	NZLSOH	非零长度OUT状态阶段握手 在OUT状态阶段，当USB设备接收到一个非零长度数据包时，该域控制USBFS是接收该包，还是用STALL握手信号拒绝该包。 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	必须保持复位值。
11	POIF	上电初始化完成 软件通过设置该位，通知USBFS寄存器从掉电模式下唤醒后已完成初始化。
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，设置该位一段时间后会引起USBFS_GINTF寄存器的GONAK标志触发，软件应该在再次写该位前清除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，设置该位一段时间会引起USBFS_GINTF寄存器的GINAK标志触发，软件应该在再次写该位前清除GINAK标志。
6:4	保留	必须保持复位值。
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NAK、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NAK、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

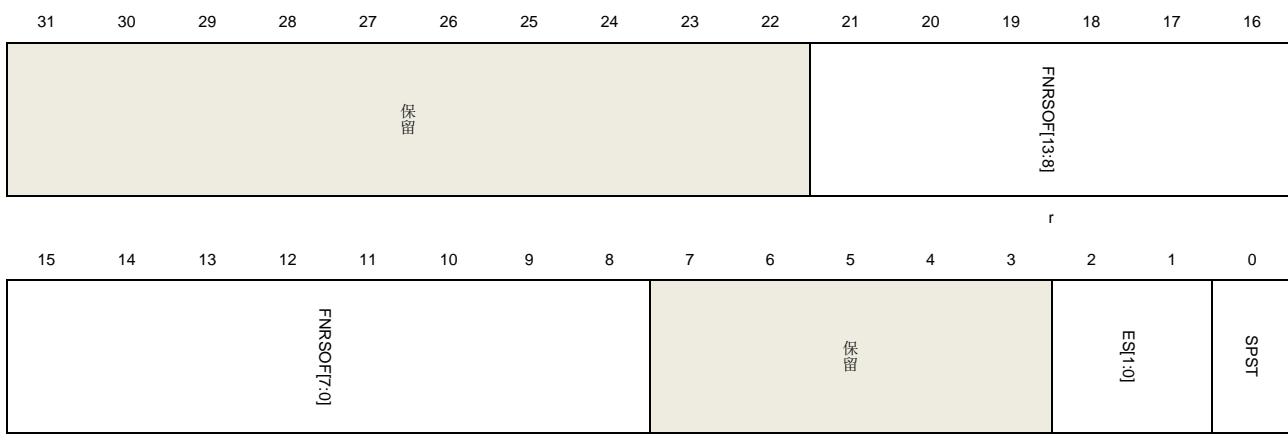
设备状态寄存器 (**USBFS_DSTAT**)

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



位/位域	名称	描述
31:22	保留	必须保持复位值。
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	必须保持复位值。
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。

11: 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: 设备不在挂起状态 1: 设备在挂起状态
---	-------------	--

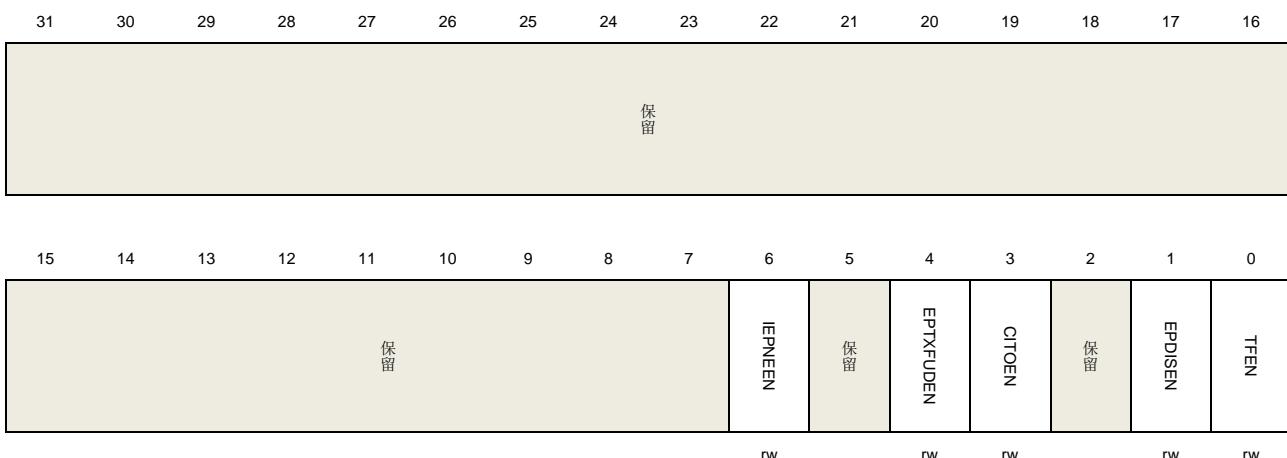
设备 IN 端点通用中断使能寄存器 (**USBFS_DIEPINTEN**)

地址偏移: 0x0810

复位值: 0x0000 0000

该寄存器包含寄存器 **USBFS_DIEPxINTF** 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寄存器 **USBFS_DAEPINT** 中的一个端点中断。该位可以通过软件置位和清零。

该寄存器采用字 (32 位) 访问



位/位域	名称	描述
31:7	保留	必须保持复位值。
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	必须保持复位值。
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	必须保持复位值。
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

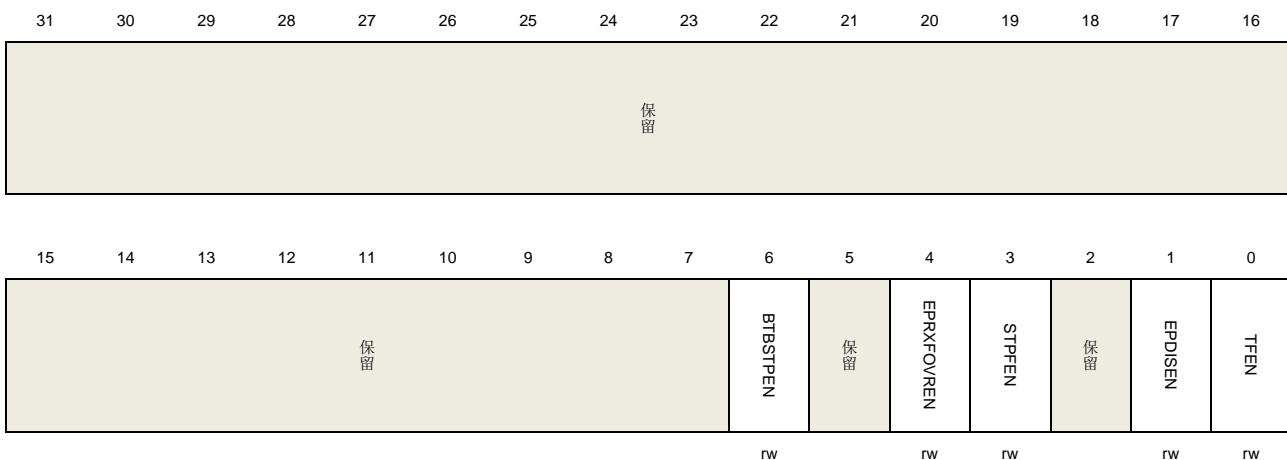
设备 OUT 端点通用中断使能寄存器 (USBFS_DOEPINTEN)

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寄存器 USBFS_DAEPINT 中的一个端点中断。该位可以通过软件置位和清零。

该寄存器采用字 (32 位) 访问



位/位域	名称	描述
31:6	保留	必须保持复位值。
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	必须保持复位值。
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	必须保持复位值。
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

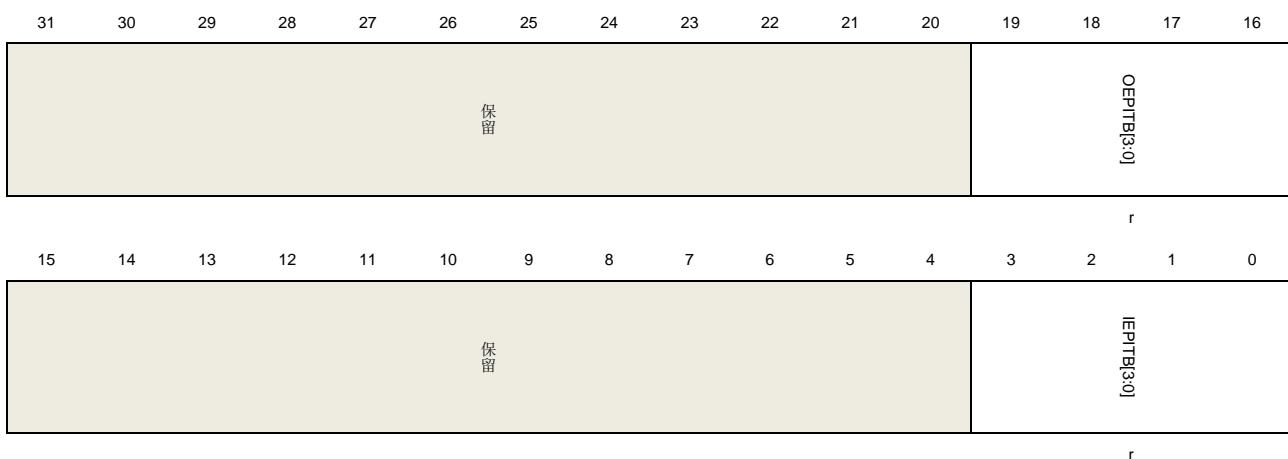
设备端点中断寄存器 (**USBFS_DAEPIINT**)

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获取在本次中断中的端点号。

该寄存器采用字（32 位）访问



位/位域	名称	描述
31:20	保留	必须保持复位值。
19:16	OEPITB[3:0]	设备OUT端点中断位 每个位代表一个OUT端点：位16代表OUT端点0，位19代表OUT端点3。
15:4	保留	必须保持复位值。
3:0	IEPITB[3:0]	设备IN端点中断位 每个位代表一个IN端点：位0代表IN端点0，位3代表IN端点3

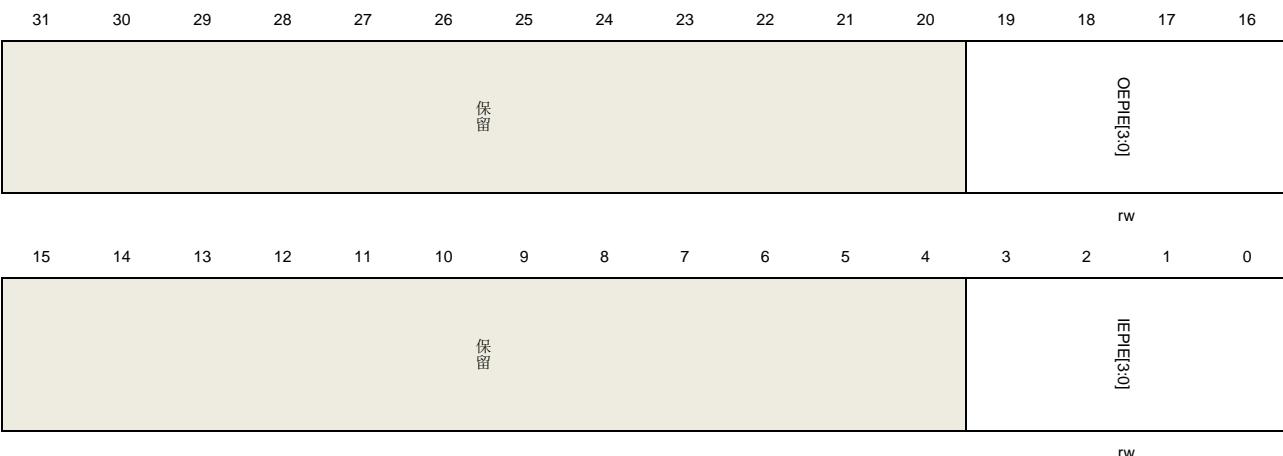
设备端点中断使能寄存器 (USBFS_DAEPINTEN)

地址偏移: 0x081C

复位值: 0x0000 0000

软件可通过该寄存器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



位/位域	名称	描述
31:20	保留	必须保持复位值。
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: 位16对应OUT端点0, 位19对应OUT端点3。
15:4	保留	必须保持复位值。
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: 位0对应IN端点0, 位3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



保留

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	DVBUSDT[15:0]	<p>设备V_{BUS}放电时间</p> <p>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$1024 \times DVBUSDT[15:0] \times T_{USBCLK}$，$T_{USBCLK}$是USB时钟周期时间。</p>

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

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

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

位/位域	名称	描述
31:12	保留	必须保持复位值。
11:0	DVBUSPT[11:0]	<p>设备V_{BUS}脉冲时间</p> <p>该域定义V_{BUS}的脉冲时间，真正的充电时间是$1024 \times DVBUSPT[15:0] \times T_{USBCLK}$，$T_{USBCLK}$是USB时钟周期时间</p>

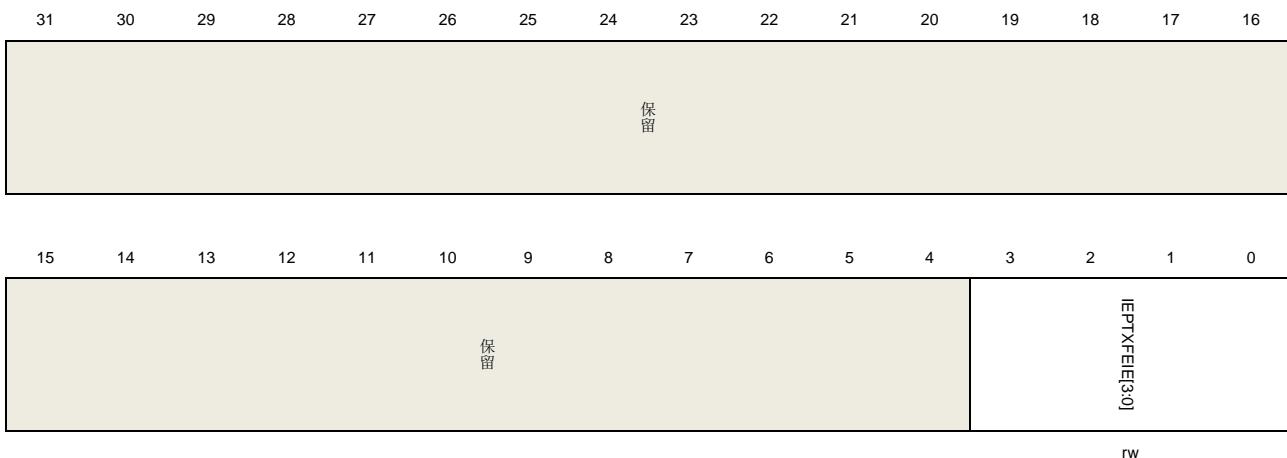
设备 IN 端点 FIFO 空中断使能寄存器 (USBFS_DIEPFEINTEN)

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



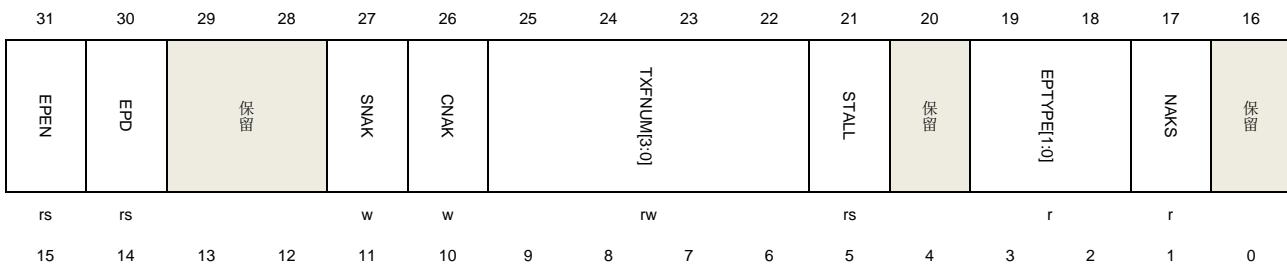
位/位域	名称	描述
31:4	保留	必须保持复位值。
3:0	IEPTXFEIE[3:0]	<p>IN端点Tx FIFO空中断的使能位</p> <p>该域控制着 USBFS_DIEPxINTF 寄存器的 TXFE 位能否置位寄存器 USBFS_DAEPINT 中的端点中断位。</p> <p>位0对应IN端点0，位3对应IN端点3</p> <p>0: 除能FIFO空中断</p> <p>1: 使能FIFO空中断</p>

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



EPACT	保留	MP[1:0]
-------	----	---------

r

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。
29:28	保留	必须保持复位值。
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	必须保持复位值。
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	必须保持复位值。
15	EPACT	端点激活 对于端点0来说，该域固定为'1'

14:2	保留	必须保持复位值。
1:0	MPL[1:0]	<p>最大包长</p> <p>此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：</p> <ul style="list-style-type: none"> 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, x 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/SD1 PID	SDOPID/ SEVENFRM	SNAK	CNAK		TXFNUM[3:0]		STALL	保留	EPTYPEE[1:0]		NAKS		EOFRM/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	<p>端点使能</p> <p>软件置位，USBFS清零</p> <p>0: 端点除能</p> <p>1: 端点使能</p> <p>软件应该按照操作指南使能或除能端点</p>
30	EPD	<p>端点除能</p> <p>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。</p>
29	SODDFRM	<p>设置奇数帧 (适用于同步IN端点)</p> <p>软件通过置1该位置1该寄存器的EOFRM位。</p>
	SD1PID	<p>设置DATA1 PID(适用于中断和大容量IN端点)</p> <p>软件可通过置1该位置1该寄存器的DPID位。</p>
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	必须保持复位值。
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仅使用奇数帧或偶数帧为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

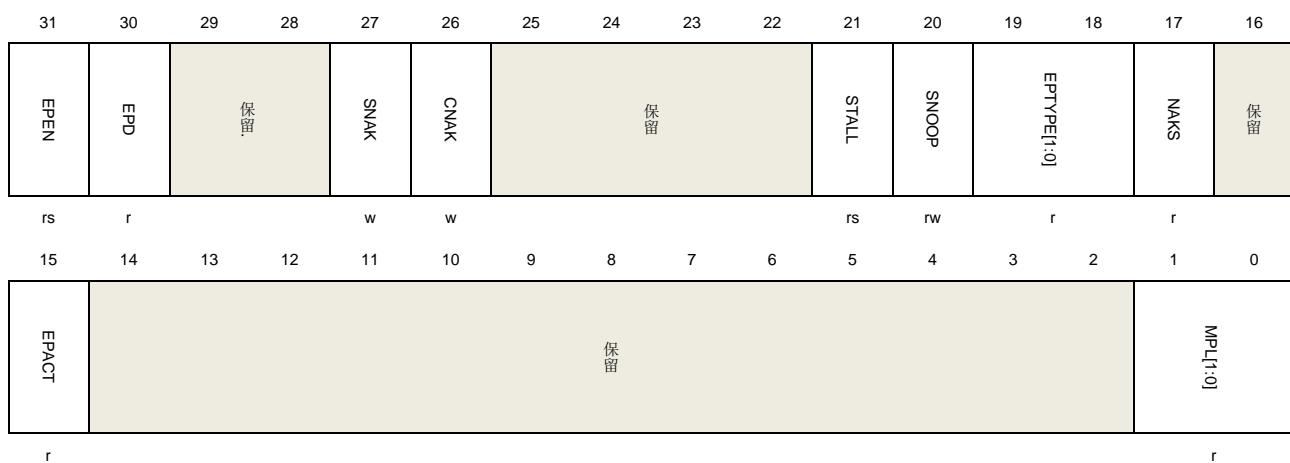
15	EPACT	端点激活
14:11	保留	该位控制端点是否激活，当端点没有激活，忽略任何令牌，不做任何回复。
10:0	MPL[10:0]	必须保持复位值。

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	必须保持复位值。
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	必须保持复位值。
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	必须保持复位值。
15	EPACT	端点激活 对于端点0，该域固定为1。
14:2	保留	必须保持复位值。
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_DIEPOCTL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 + (端点编号 × 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

EPEN	EPD	SODDFRM/SID1	SD0PID / SEVENFRM /	SNAK	CNAK	保留	STALL	SNOOP	EPTYPE[1:0]	NAKS	EOFRM/DPID
rs	rs	w	w	w	w				rw/rs	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4
EPACT	保留								MPL[1:0]		
	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	必须保持复位值。
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丢弃数据包。 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 此位控制端点是否激活，当端点没有激活，忽略任何令牌，不作任何回复。
14:11	保留	必须保持复位值。
10:0	MPL[10:0]	该位定义最大包长

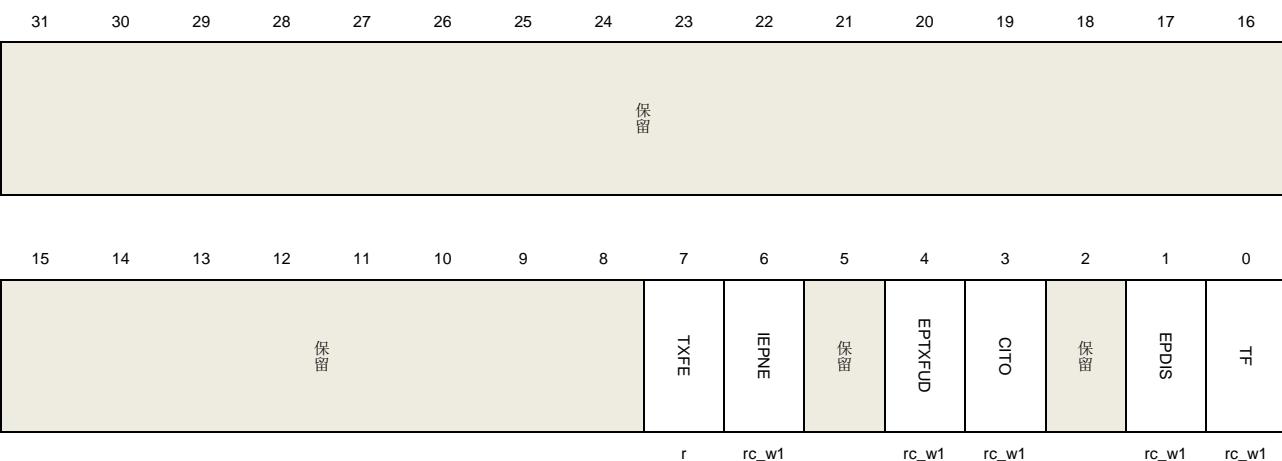
设备 IN 端点 x 中断标志寄存器 (USBFS_DIEPxINTF) (x = 0...3, x 是端点编号)

地址偏移：0x0908 + (端点编号 × 0x20)

复位值：0x0000 0080

该寄存器包含 IN 端点的状态和事件，当一个 IN 端点的中断发生时，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由硬件置位，除了只读的 TXFE 位，其他位写 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	必须保持复位值。
7	TXFE	发送FIFO空 IN端点的Tx FIFO达到寄存器USBFS_GAHBCS的位TXFTH所定义的空阈值。
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或置位CNAK位来清零。
5	保留	必须保持复位值。
4	EPTXFUD	端点Tx FIFO下溢 如果当IN令牌被接收后，Tx FIFO没有数据包可发送，该标志位被触发。
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	必须保持复位值。
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。

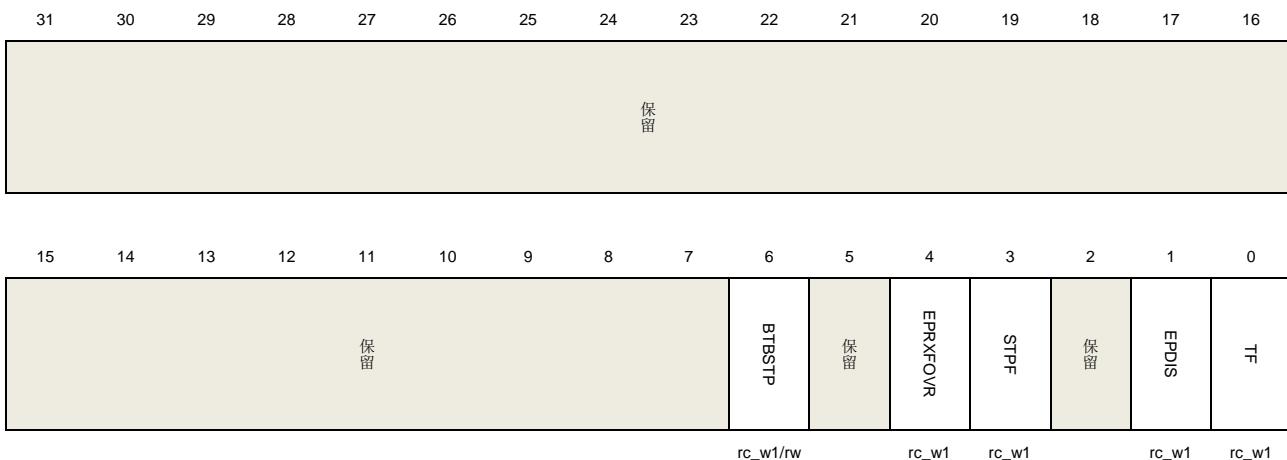
设备 OUT 端点 x 中断标志寄存器 (USBFS_DOEPxINTF) (x = 0..3, x 是端点编号)

地址偏移: 0x0B08 + (端点编号 × 0x20)

复位值: 0x0000 0000

该寄存器包含 OUT 端点的状态和事件，当一个 OUT 端点中断发生时，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



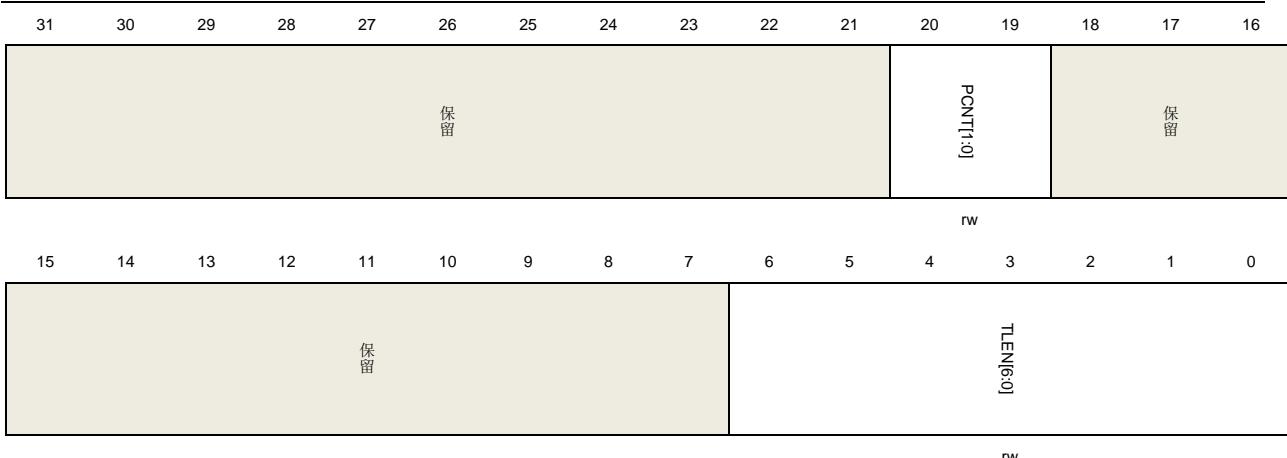
位/位域	名称	描述
31:7	保留	必须保持复位值。
6	BTBSTOP	连续SETUP包（适用于控制OUT端点） 当一个控制OUT端点接收超过连续3个SETUP包时，该标志被触发。
5	保留	必须保持复位值。
4	EPRXFOVR	端点Rx FIFO上溢 当OUT令牌被接收时，如果OUT端点的Rx FIFO没有足够的空间存放数据包，该位被触发。在这种情况下，USBFS不能接收OUT数据包，发送一个NAK握手包。
3	STPF	SETUP阶段完成（适用于控制OUT端点） 当一个SETUP阶段完成，也就是USBFS在一个setup令牌后接收了一个IN或OUT令牌，该位被置位。
2	保留	必须保持复位值。
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	必须保持复位值。
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递减。
18:7	保留	必须保持复位值。
6:0	TLEN[6:0]	传输长度 一次传输的数据总字节数。 该域是IN传输中需要发送的数据包的总字节数，在端点使能之前，软件设置该位，在成功地将数据包写入端点的Tx FIFO中，该域按照数据包大小递减。

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



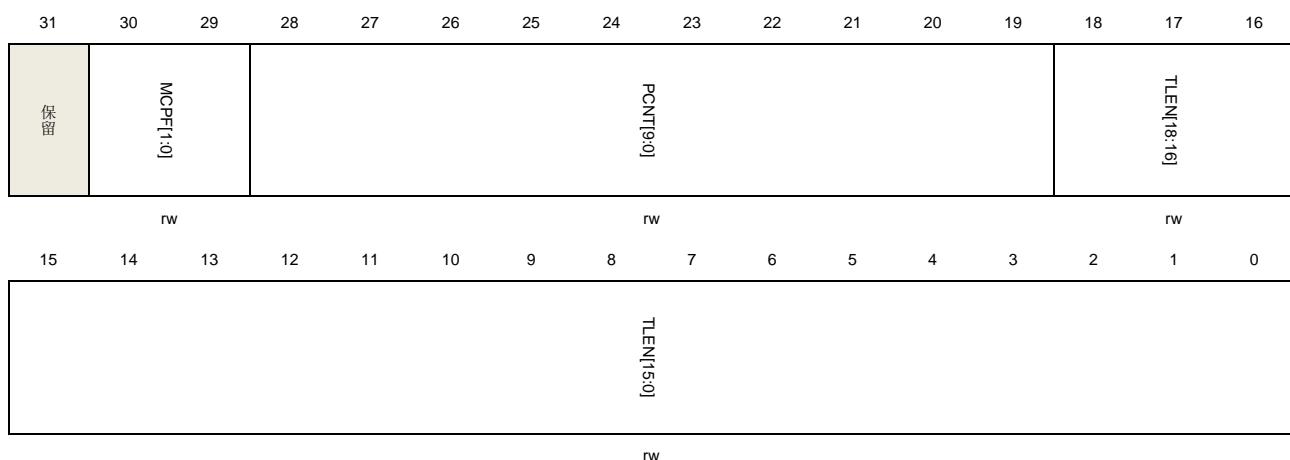
位/位域	名称	描述
31	保留	必须保持复位值。
30:29	STPCNT[1:0]	<p>SETUP包数目</p> <p>该域定义端点可以接受的最大连续SETUP包数量。</p> <p>在SETUP传输之前，设置该域，每当连续SETUP包接收到时，该域值减1，当该域达到0时，寄存器USBFS_DOEP0INTF的BTBSTOP标志被触发。</p> <p>00: 0个包 01: 1个包 10: 2个包 11: 3个包</p>
28:20	保留	必须保持复位值。
19	PCNT	<p>包数目</p> <p>一次传输中应该接收到包数量。</p> <p>在使能端点前，软件设置该位，在传输开始后，每当数据包接收到后，该域数值递减。</p>
18:7	保留	必须保持复位值。
6:0	TLEN[6:0]	<p>传输长度</p> <p>传输中数据总字数。</p> <p>该域是OUT传输中需要接收的数据包的总字节数，在端点使能之前，软件设置该位，在软件成功地将数据包从Rx FIFO中读取后，该域按照数据包大小递减。</p>

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 + (端点编号 × 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	必须保持复位值。

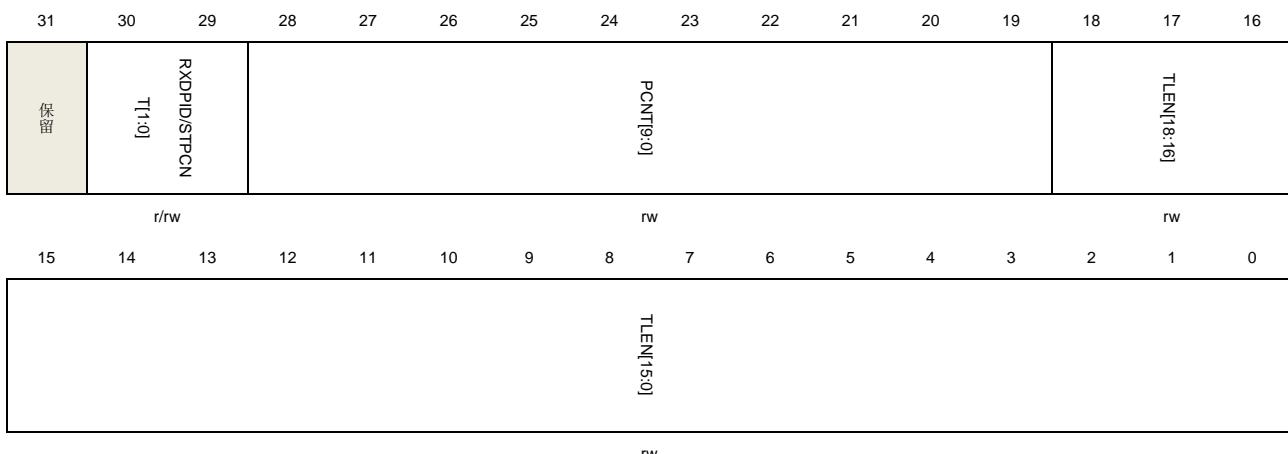
30:29	MCPF[1:0]	每帧多包数目 该域是USB周期性IN端点每帧必须发的包数目。用于计算同步IN端点的数据PID。 01: 1个包 10: 2个包 11: 3个包
28:19	PCNT[9:0]	包数量 传输中被发送的数据包数量。 在端点使能之前，软件设置该位，在传输开始后，该域在每次数据包成功发送后递减。
18:0	TLEN[18:0]	传输长度 传输的数据总字节数。 该域是IN传输中需要发送的数据包的总字节数，在端点使能之前，软件设置该位，在成功地将数据包写入端点的Tx FIFO后，该域按照数据包大小递减。

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 + (端点编号 × 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	必须保持复位值。
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
	STPCNT[1:0]	SETUP包数（适用于控制OUT端点） 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]	包数目 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，该域数值递减。
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，在软件成功地将数据包从Rx FIFO中读取后，该域按照数据包大小递减。

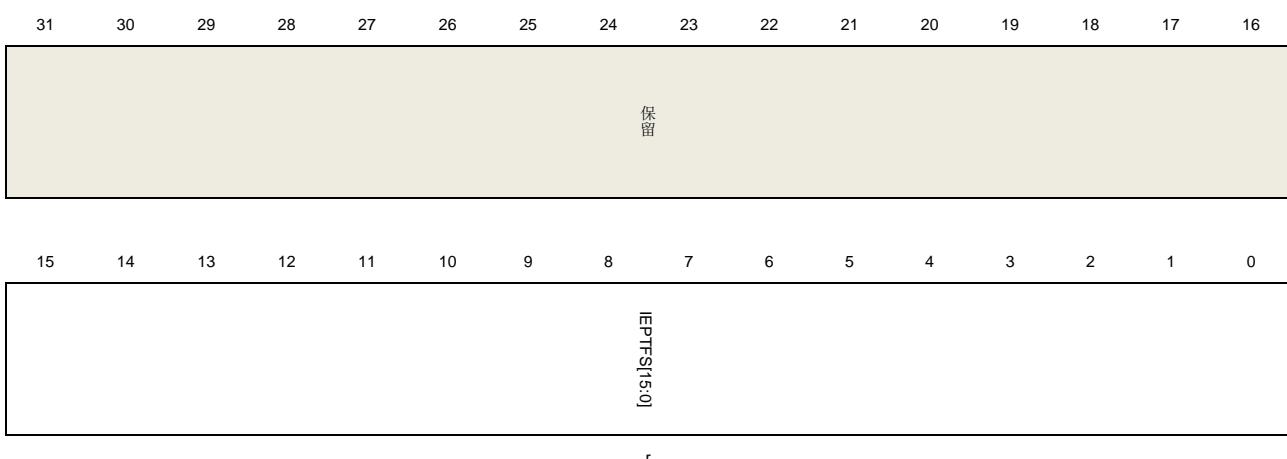
设备 IN 端点 x 发送 FIFO 状态寄存器 (**USBFS_DIEPxTFSTAT**) (x = 0...3, x 是端点编号)

地址偏移: 0x0918 + (端点编号 × 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	必须保持复位值。
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: Tx FIFO满 1: 1字可用 ...

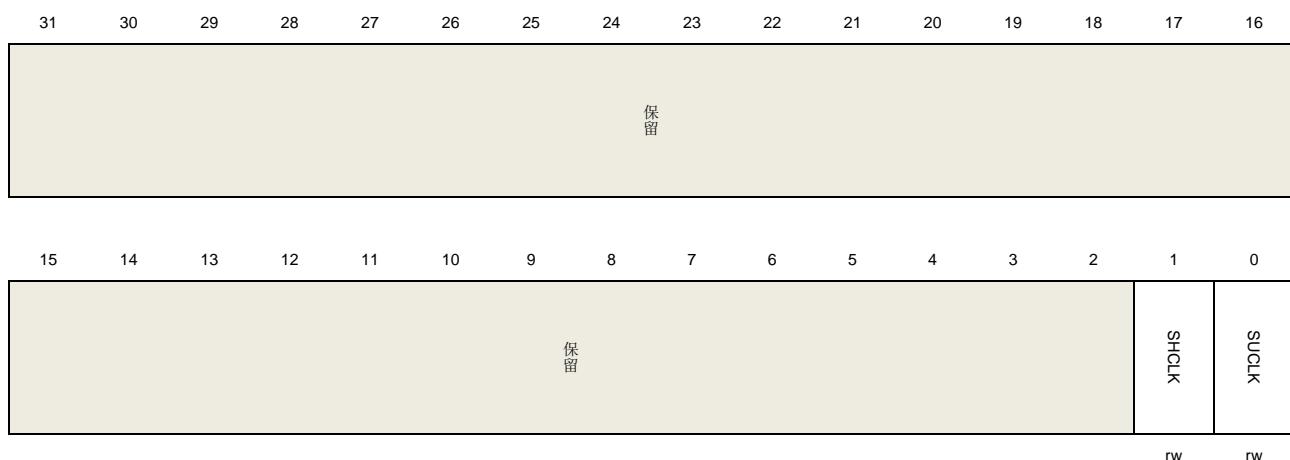
n: n字可用

22.7.4. 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	必须保持复位值。
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

23. 文档附录

23.1. 寄存器表中使用的缩写列表

表 23-1. 寄存器功能位访问属性

功能位访问属性	描述
读/写(rw)	软件可以对这个位进行读写。
只读(r)	软件只能对这个位进行读。
只写(w)	软件只能对这个位进行写。读取该位将返回复位值。
读/写 1 清零(rc_w1)	软件可以读该位，对该位写入 1 可以清除这个位。写入 0 对位值没有影响。
读/写 0 清零(rc_w0)	软件可以读该位，对该位写入 0 可以清除这个位。写入 1 对位值没有影响。
翻转(t)	软件可以通过写 1 来翻转该位。写入 0 对位值没有效果。
只读/写 1 触发 (rt_w1)	软件可以读该位，写入 1 触发事件，但对位值没有影响。

23.2. 术语表

表 23-2. 术语

术语	描述
字	32 位长度数据
半字	16 位长度数据.
字节	8 位长度数据
IAP(应用内编程)	IAP 是在用户程序运行时对微控制器的闪存重新编程的能力。
ICP(在线编程)	ICP 是当设备安装在用户应用板上时，一个使用 JTAG 协议，SWD 协议或引导加载程序的微控制器的闪存编程能力。
选项字节	存储在闪存中的产品配置位
AHB	高级高性能总线
APB	高级外设总线
RAZ	读为 0
WI	写忽略
RAZ/WI	读为 0/写忽略

23.3. 可用外设

对于各个MCU系列的外设及其数量，请参考相应型号的数据手册。

24. 版本历史

表 24-1. 版本历史

版本号.	说明	日期
1.0	初稿发布	2020 年 12 月
1.1	<p>1.修改 CAN 章节 接收 FIFO 邮箱属性寄存器 (CAN_RFIFOMP_x) (x=0,1) 的 ESI 位描述和 过滤器控制寄存器 (CAN_FCTL) 的 HBC1F 位描述。</p> <p>2.修改 WDGT 章节 表 14-1. 独立看门狗定时器在 40kHz (IRC40K)时的最小/最大超时周期 中的参数。</p> <p>3.修改 GPIO 章节 AFIO 端口配置寄存器 0 (AFIO_PCF0) 中关于 bit29 的位域描述。</p> <p>4.修改系统及存储架构章节 表 2-1. GD32C10x 闪存基址址和构成 中的 SRAM 和 FLASH 地址范围。</p> <p>5.修改 PMU 章节 电源控制和状态寄存器 (PMU_CS) 中关于 bit0 的位域描述。</p> <p>6.删除系统及存储架构章节的 ARM Cortex-M4 处理器 部分中涉及 ETM 的内容。</p>	2021 年 12 月 16 日

Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the "Product"), is owned by the Company under the intellectual property laws and treaties of the People's Republic of China and other jurisdictions worldwide. The Company reserves all rights under such laws and treaties and does not grant any license under its patents, copyrights, trademarks, or other intellectual property rights. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

The Company makes no warranty of any kind, express or implied, with regard to this document or any Product, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Company does not assume any liability arising out of the application or use of any Product described in this document. Any information provided in this document is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Except for customized products which has been expressly identified in the applicable agreement, the Products are designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only. The Products are not designed, intended, or authorized for use as components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, atomic energy control instruments, combustion control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or Product could cause personal injury, death, property or environmental damage ("Unintended Uses"). Customers shall take any and all actions to ensure using and selling the Products in accordance with the applicable laws and regulations. The Company is not liable, in whole or in part, and customers shall and hereby do release the Company as well as its suppliers and/or distributors from any claim, damage, or other liability arising from or related to all Unintended Uses of the Products. Customers shall indemnify and hold the Company as well as its suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Products.

Information in this document is provided solely in connection with the Products. The Company reserves the right to make changes, corrections, modifications or improvements to this document and Products and services described herein at any time, without notice.