

GigaDevice Semiconductor Inc.

GD32F50x

Arm® Cortex®-M33 32-bit MCU

适用于 GD32F505xx, GD32F503xx

用户手册

1.0 版本

(2025 年 10 月)

目录

目录.....	2
图索引	19
表索引	27
1. 系统及存储器架构	30
1.1. Arm® Cortex®-M33 处理器	30
1.2. 系统架构	31
1.3. 存储器映射	34
1.3.1. 片上 SRAM 存储器	38
1.3.2. 片上 FLASH 存储器概述	39
1.4. 引导配置	39
1.5. 系统配置寄存器(SYSCFG)	41
1.5.1. 配置寄存器 0 (SYSCFG_CFG0).....	41
1.5.2. 配置寄存器 1(SYSCFG_CFG1).....	41
1.5.3. 锁定控制寄存器 (SYSCFG_LKCTL)	42
1.5.4. 总线超时寄存器(SYSCFG_BUSTO)	43
1.5.5. TIMER 输入选择寄存器 (SYSCFG_TIMERCISEL).....	43
1.5.6. FPU 中断使能寄存器(SYSCFG_FPUINTEN).....	44
1.5.7. SRAM 写保护寄存器(SYSCFG_SRAMWP)	45
1.5.8. SRAM ECC 状态寄存器 (SYSCFG_SRAMECCSTAT)	45
1.5.9. SRAM ECC 控制状态寄存器(SYSCFG_SRAMECCCS).....	46
1.5.10. 总线超时状态寄存器(SYSCFG_BUSTOSTAT).....	47
1.5.11. TIMERx 配置寄存器 0 (SYSCFG_TIMERxCFG0, x=0, 7)	48
1.5.12. TIMERx 配置寄存器 1 (SYSCFG_TIMERxCFG1, x=0, 7)	50
1.5.13. TIMERx 配置寄存器 2 (SYSCFG_TIMERxCFG2, x=0, 7)	50
1.5.14. TIMERx 配置寄存器 0 (SYSCFG_TIMERxCFG0, x=1, 2, 3, 4)	51
1.5.15. TIMERx 配置寄存器 1 (SYSCFG_TIMERxCFG1, x=1, 2, 3, 4)	53
1.5.16. TIMERx 配置寄存器 2 (SYSCFG_TIMERxCFG2, x=1, 2, 3, 4)	54
1.5.17. TIMERx 配置寄存器 0 (SYSCFG_TIMERxCFG0, x=15, 16)	55
1.5.18. TIMERx 配置寄存器 1 (SYSCFG_TIMERxCFG1, x=15, 16)	57
1.5.19. TIMERx 配置寄存器 2 (SYSCFG_TIMERxCFG2, x=15, 16)	58
1.6. 设备电子签名.....	59
1.6.1. 存储容量信息	59
1.6.2. 设备唯一 ID(96 位/位域).....	59
2. 中断/事件控制器 (EXTI)	61

2.1.	简介	61
2.2.	主要特征	61
2.3.	功能说明	61
2.4.	外部中断及事件 (EXTI) 框图	65
2.5.	外部中断及事件功能概述	65
2.6.	EXTI 寄存器	68
2.6.1.	中断使能寄存器 (EXTI_INTEN)	68
2.6.2.	事件使能寄存器 (EXTI_EVEN)	68
2.6.3.	上升沿触发使能寄存器 (EXTI_RTEN)	69
2.6.4.	下降沿触发使能寄存器 (EXTI_FTEN)	69
2.6.5.	软件中断事件寄存器 (EXTI_SWIEV)	70
2.6.6.	挂起寄存器 (EXTI_PD)	70
3.	直接存储器访问控制器 (DMA)	71
3.1.	简介	71
3.2.	主要特性	71
3.3.	结构框图	72
3.4.	功能描述	72
3.4.1.	DMA 操作	72
3.4.2.	外设握手	73
3.4.3.	仲裁	74
3.4.4.	地址生成	74
3.4.5.	循环模式	74
3.4.6.	存储器到存储器模式	75
3.4.7.	通道配置	75
3.4.8.	中断	75
3.4.9.	DMA 请求映射	76
3.5.	DMA 寄存器	77
3.5.1.	中断标志位寄存器 (DMA_INTF)	77
3.5.2.	中断标志位清除寄存器 (DMA_INTC)	78
3.5.3.	通道 x 控制寄存器 (DMA_CHxCTL)	78
3.5.4.	通道 x 计数寄存器 (DMA_CHxCNT)	80
3.5.5.	通道 x 外设基地址寄存器 (DMA_CHxPADDR)	81
3.5.6.	通道 x 存储器基地址寄存器 (DMA_CHxMADDR)	81
4.	DMA 请求多路复用器 (DMAMUX)	83
4.1.	简介	83
4.2.	主要特征	83

4.3. 结构框图	84
4.4. 功能说明	84
4.4.1. DMAMUX 信号	85
4.4.2. DMAMUX 请求路由器	85
4.4.3. DMAMUX 请求生成器	87
4.4.4. 通道配置	88
4.4.5. 中断	88
4.4.6. DMAMUX 映射	89
4.5. DMAMUX 寄存器	94
4.5.1. 请求路由通道 x 配置寄存器 (DMAMUX_RM_CHxCFG)	94
4.5.2. 请求路由通道中断标志位寄存器 (DMAMUX_RM_INTF)	95
4.5.3. 请求路由通道中断标志位清除寄存器 (DMAMUX_RM_INTC)	95
4.5.4. 请求生成通道 x 配置寄存器 (DMAMUX_RG_CHxCFG)	96
4.5.5. 请求生成通道中断标志位寄存器 (DMAMUX_RG_INTF)	97
4.5.6. 请求生成通道中断标志位清除寄存器 (DMAMUX_RG_INTC)	97
5. 闪存控制器 (FMC)	99
5.1. 简介	99
5.2. 主要特征	99
5.3. 功能说明	99
5.3.1. 闪存结构	99
5.3.2. 读操作	100
5.3.3. FMC_CTLx / FMC_OBCTLx 寄存器解锁	101
5.3.4. 页擦除	101
5.3.5. 整片擦除	102
5.3.6. 主存储闪存块编程	104
5.3.7. 选项字节块修改	105
5.3.8. 选项字节块说明	106
5.3.9. 页擦除/编程保护	107
5.3.10. OTP 闪存块编程	107
5.3.11. 安全保护	109
5.3.12. 频率控制	110
5.4. FMC 寄存器	111
5.4.1. 解锁寄存器 (FMC_KEY0)	111
5.4.2. 选项字节操作解锁寄存器 (FMC_OBKEY)	111
5.4.3. 状态寄存器 0 (FMC_STAT0)	112
5.4.4. 控制寄存器 0 (FMC_CTL0)	112
5.4.5. 地址寄存器 0 (FMC_ADDR0)	114
5.4.6. 选项字节控制寄存器 0 (FMC_OBCTL0)	114

5.4.7.	选项字节控制寄存器 1 (FMC_OBCTL1).....	115
5.4.8.	选项字节控制寄存器 2 (FMC_OBCTL2).....	116
5.4.9.	OTP1 配置寄存器 (FMC_OTP1CFG).....	116
5.4.10.	选项字节状态寄存器 (FMC_OBSTAT).....	117
5.4.11.	解锁寄存器 1 (FMC_KEY1).....	117
5.4.12.	状态寄存器 1 (FMC_STAT1).....	118
5.4.13.	控制寄存器 1 (FMC_CTL1).....	118
5.4.14.	地址寄存器 1 (FMC_ADDR1).....	119
5.4.15.	OTP3 状态寄存器 (FMC_OTP3_STAT).....	120
5.4.16.	产品 ID 寄存器 (FMC_PID).....	121
6.	电源管理单元 (PMU)	122
6.1.	简介	122
6.2.	主要特征.....	122
6.3.	功能说明.....	123
6.3.1.	电池备份域.....	123
6.3.2.	VDD / VDDA 电源域.....	124
6.3.3.	VCORE 电源域.....	127
6.3.4.	省电模式.....	129
6.4.	PMU 寄存器.....	132
6.4.1.	控制寄存器 0 (PMU_CTL0).....	132
6.4.2.	电源控制和状态寄存器 (PMU_CS).....	135
6.4.3.	控制寄存器 1 (PMU_CTL1).....	136
7.	备份寄存器 (BKP)	138
7.1.	简介	138
7.2.	主要特点.....	138
7.3.	功能描述.....	138
7.3.1.	RTC 时钟校准.....	138
7.3.2.	侵入检测.....	138
7.4.	BKP 寄存器.....	140
7.4.1.	备份数据寄存器 (BKP_DATAx) (x= 0..41).....	140
7.4.2.	RTC 信号输出控制寄存器 (BKP_OCTL).....	140
7.4.3.	侵入引脚控制寄存器 (BKP_TPCTL).....	141
7.4.4.	侵入控制状态寄存器 (BKP_TPCS).....	142
8.	复位和时钟单元 (RCU)	144
8.1.	复位控制单元 (RCTL)	144
8.1.1.	简介.....	144
8.1.2.	功能说明.....	144

8.2. 时钟控制单元 (CCTL)	145
8.2.1. 简介	145
8.2.2. 主要特征	148
8.2.3. 功能说明	148
8.3. RCU 寄存器	153
8.3.1. 控制寄存器 (RCU_CTL)	153
8.3.2. 时钟配置寄存器 0 (RCU_CFG0)	155
8.3.3. 时钟中断寄存器 (RCU_INT)	158
8.3.4. APB2 复位寄存器 (RCU_APB2RST)	161
8.3.5. APB1 复位寄存器 (RCU_APB1RST)	163
8.3.6. AHB 使能寄存器 (RCU_AHBEN)	166
8.3.7. APB2 使能寄存器 (RCU_APB2EN)	168
8.3.8. APB1 使能寄存器 (RCU_APB1EN)	171
8.3.9. 备份域控制寄存器 (RCU_BDCTL)	173
8.3.10. 复位源/时钟寄存器 (RCU_RSTSCK)	175
8.3.11. AHB 复位寄存器 (RCU_AHBRSRST)	177
8.3.12. 时钟配置寄存器 1 (RCU_CFG1)	178
8.3.13. PLL 带宽配置寄存器 (RCU_PLLBWCFG)	180
8.3.14. 深度睡眠模式电压寄存器 (RCU_DSV)	181
8.3.15. 时钟频率监视器配置寄存器 0 (RCU_CKFMCFG0)	182
8.3.16. 时钟频率监视器配置寄存器 1 (RCU_CKFMCFG1)	183
8.3.17. 时钟频率监视器配置寄存器 2 (RCU_CKFMCFG2)	184
8.3.18. 时钟频率监视器配置寄存器 3 (RCU_CKFMCFG3)	185
8.3.19. 附加时钟控制寄存器 (RCU_ADDCTL)	186
8.3.20. 附加时钟中断寄存器 (RCU_ADDINT)	187
8.3.21. APB1 附加复位寄存器 (RCU_ADDAPB1RST)	188
8.3.22. APB1 附加使能寄存器 (RCU_ADDAPB1EN)	189
8.3.23. 上锁寄存器 (RCU_LOCK)	189
9. 时钟校准控制器 (CTC)	191
9.1. 简介	191
9.2. 主要特征	191
9.3. 功能说明	191
9.3.1. REF 同步脉冲发生器	192
9.3.2. CTC 校准计数器	192
9.3.3. 频率评估和自动校准过程	193
9.3.4. 软件编程指南	194
9.4. CTC 寄存器	196
9.4.1. 控制寄存器 0 (CTC_CTL0)	196
9.4.2. 控制寄存器 1 (CTC_CTL1)	197

9.4.3.	状态寄存器 (CTC_STAT)	198
9.4.4.	中断清除寄存器 (CTC_INTC)	200
10.	通用和备用输入/输出接口 (GPIO 和 AFIO)	202
10.1.	简介	202
10.2.	主要特征	202
10.3.	功能说明	202
10.3.1.	GPIO 引脚配置	203
10.3.2.	外部中断/事件线	204
10.3.3.	备用功能 (AF)	204
10.3.4.	附加功能	204
10.3.5.	输入配置	205
10.3.6.	输出配置	205
10.3.7.	模拟配置	206
10.3.8.	备用功能 (AF) 配置	206
10.3.9.	GPIO 锁定功能	207
10.4.	GPIO 寄存器	208
10.4.1.	端口控制寄存器 0 (GPIOx_CTL, x=A..E)	208
10.4.2.	端口输出模式寄存器 (GPIOx_OMODE, x=A..E)	210
10.4.3.	端口输出速度寄存器 (GPIOx_OSPD, x=A..E)	211
10.4.4.	端口上拉/下拉寄存器 (GPIOx_PUD, x=A..E)	213
10.4.5.	端口输入状态寄存器 (GPIOx_ISTAT, x=A..E)	215
10.4.6.	端口输出控制寄存器 (GPIOx_OCTL, x=A..E)	215
10.4.7.	端口位操作寄存器 (GPIOx_BOP, x=A..E)	216
10.4.8.	端口配置锁定寄存器 (GPIOx_LOCK, x=A..E)	217
10.4.9.	备用功能选择寄存器 0 (GPIOx_AFSEL0, x=A..E)	217
10.4.10.	备用功能选择寄存器 1 (GPIOx_AFSEL1, x=A..E)	219
10.4.11.	位清除寄存器 (GPIOx_BC, x=A..E)	220
10.4.12.	EXTI 源选择寄存器 0 寄存器 (AFIO_EXTISS0)	220
10.4.13.	EXTI 源选择寄存器 1 寄存器 (AFIO_EXTISS1)	221
10.4.14.	EXTI 源选择寄存器 2 寄存器 (AFIO_EXTISS2)	223
10.4.15.	EXTI 源选择寄存器 3 寄存器 (AFIO_EXTISS3)	224
11.	触发选择控制器 (TRIGSEL)	226
11.1.	简介	226
11.2.	主要特征	226
11.3.	功能说明	226
11.4.	内部连接	227
11.5.	TRIGSEL 寄存器	234

11.5.1.	EXTOUT 触发选择寄存器 0 (TRIGSEL_EXTOUT_0)	234
11.5.2.	EXTOUT 触发选择寄存器 1 (TRIGSEL_EXTOUT_1)	234
11.5.3.	EXTOUT 触发选择寄存器 2 (TRIGSEL_EXTOUT_2)	235
11.5.4.	EXTOUT 触发选择寄存器 3 (TRIGSEL_EXTOUT_3)	236
11.5.5.	TIMER0_ITI 触发选择寄存器 (TRIGSEL_TIMER0ITI)	236
11.5.6.	TIMER1_ITI 触发选择寄存器 (TRIGSEL_TIMER1ITI)	237
11.5.7.	TIMER2_ITI 触发选择寄存器 (TRIGSEL_TIMER2ITI)	238
11.5.8.	TIMER3_ITI 触发选择寄存器 (TRIGSEL_TIMER3ITI)	238
11.5.9.	TIMER4_ITI 触发选择寄存器 (TRIGSEL_TIMER4ITI)	239
11.5.10.	TIMER7_ITI 触发选择寄存器 (TRIGSEL_TIMER7ITI)	239
11.5.11.	TIMER15_ITI 触发选择寄存器 (TRIGSEL_TIMER15ITI)	240
11.5.12.	TIMER16_ITI 触发选择寄存器 (TRIGSEL_TIMER16ITI)	241
11.5.13.	DAC 触发选择寄存器 (TRIGSEL_DAC)	241
11.5.14.	ADC0_ROUTRG 触发选择寄存器 (TRIGSEL_ADC0_ROUTRG)	242
11.5.15.	ADC0_INSTRG 触发选择寄存器 (TRIGSEL_ADC0_INSTRG)	242
11.5.16.	ADC1_ROUTRG 触发选择寄存器 (TRIGSEL_ADC1_ROUTRG)	243
11.5.17.	ADC1_INSTRG 触发选择寄存器 (TRIGSEL_ADC1_INSTRG)	244
11.5.18.	ADC2_ROUTRG 触发选择寄存器 (TRIGSEL_ADC2_ROUTRG)	244
11.5.19.	ADC2_INSTRG 触发选择寄存器 (TRIGSEL_ADC2_INSTRG)	245
11.5.20.	TIMER0_BRKIN 触发选择寄存器 (TRIGSEL_TIMER0BRKIN)	246
11.5.21.	TIMER0_CHBRKIN 触发选择寄存器 (TRIGSEL_TIMER0CHBRKIN)	246
11.5.22.	TIMER7_BRKIN 触发选择寄存器 (TRIGSEL_TIMER7BRKIN)	247
11.5.23.	TIMER7_CHBRKIN 触发选择寄存器 (TRIGSEL_TIMER7CHBRKIN)	248
11.5.24.	TIMER15_BRKIN 触发选择寄存器 (TRIGSEL_TIMER15BRKIN)	248
11.5.25.	TIMER16_BRKIN 触发选择寄存器 (TRIGSEL_TIMER16BRKIN)	249
12.	定时器 (TIMER)	250
12.1.	高级定时器 (TIMERx, x=0, 7)	252
12.1.1.	简介	252
12.1.2.	主要特性	252
12.1.3.	结构框图	252
12.1.4.	功能描述	253
12.1.5.	TIMERx 寄存器 (x=0,7)	292
12.2.	通用定时器 L0 (TIMERx, x=1,2,3,4)	336
12.2.1.	简介	336
12.2.2.	主要特性	336
12.2.3.	结构框图	336
12.2.4.	功能描述	337
12.2.5.	TIMERx 寄存器 (x=1/2/3/4)	357
12.3.	通用定时器 L3 (TIMERx, x=15,16)	382
12.3.1.	简介	382

12.3.2.	主要特性	382
12.3.3.	结构框图	382
12.3.4.	功能描述	383
12.3.5.	TIMERx 寄存器(x=15/16).....	410
12.4.	基本定时器 (TIMERx, x=5,6)	443
12.4.1.	简介	443
12.4.2.	主要特性	443
12.4.3.	结构框图	443
12.4.4.	功能描述	443
12.4.5.	TIMERx 寄存器 (x=5/6)	447
13.	实时时钟 (RTC)	452
13.1.	简介	452
13.2.	主要特性	452
13.3.	功能描述	452
13.3.1.	RTC 复位	453
13.3.2.	RTC 读取	453
13.3.3.	RTC 配置	453
13.3.4.	RTC 标志位	454
13.4.	RTC 寄存器	456
13.4.1.	RTC 中断使能寄存器 (RTC_INTEN)	456
13.4.2.	RTC 控制寄存器 (RTC_CTL)	456
13.4.3.	RTC 预分频寄存器高位 (RTC_PSCH)	457
13.4.4.	RTC 预分频寄存器低位 (RTC_PSCL)	458
13.4.5.	RTC 分频器高位 (RTC_DIVH)	458
13.4.6.	RTC 分频器低位 (RTC_DIVL)	458
13.4.7.	RTC 计数寄存器高位 (RTC_CNTH)	459
13.4.8.	RTC 计数寄存器低位 (RTC_CNTL)	459
13.4.9.	RTC 闹钟寄存器高位 (RTC_ALRMH)	460
13.4.10.	RTC 闹钟寄存器低位 (RTC_ALRML)	460
14.	看门狗定时器 (WDGT)	462
14.1.	独立看门狗定时器 (FWDGT)	462
14.1.1.	简介	462
14.1.2.	主要特征	462
14.1.3.	功能说明	462
14.1.4.	FWDGT 寄存器	465
14.2.	窗口看门狗定时器 (WWDGT)	468
14.2.1.	简介	468
14.2.2.	主要特征	468

14.2.3.	功能说明	468
14.2.4.	WWDGT 寄存器	471
15.	循环冗余校验计算单元 (CRC)	473
15.1.	简介	473
15.2.	主要特性	473
15.3.	功能描述	474
15.4.	CRC 寄存器	475
15.4.1.	数据寄存器 (CRC_DATA)	475
15.4.2.	独立数据寄存器 (CRC_FDATA)	475
15.4.3.	控制寄存器 (CRC_CTL)	476
15.4.4.	初值寄存器 (CRC_IDATA)	477
15.4.5.	多项式寄存器 (CRC_POLY)	477
16.	调试 (DBG)	478
16.1.	简介	478
16.2.	JTAG/SW 功能描述	478
16.2.1.	切换 JTAG/ SW 接口	478
16.2.2.	引脚分配	478
16.2.3.	JTAG 链状结构	479
16.2.4.	调试复位	479
16.2.5.	JEDEC-106 ID code	479
16.3.	调试保持功能描述	480
16.3.1.	低功耗模式调试支持	480
16.3.2.	TIMER, I2C, WWDGT, FWDGT 和 CAN 外设调试支持	480
16.4.	DBG 寄存器	481
16.4.1.	ID 寄存器 (DBG_ID)	481
16.4.2.	控制寄存器 (DBG_CTL)	481
17.	真随机数生成器 (TRNG)	485
17.1.	简介	485
17.2.	主要特征	485
17.3.	功能说明	485
17.3.1.	操作流程	486
17.3.2.	错误标志	486
17.4.	TRNG 寄存器	487
17.4.1.	控制寄存器 (TRNG_CTL)	487
17.4.2.	状态寄存器 (TRNG_STAT)	487

17.4.3. 数据寄存器 (TRNG_DATA)	488
18. 加密处理器 (CAU)	490
18.1. 简介	490
18.2. 主要特征	490
18.3. CAU 数据类型和初始化向量	490
18.3.1. 数据类型	490
18.4. 加密处理器流程	492
18.4.1. AES 加密处理流程	492
18.5. 操作模式	494
18.6. CAU DMA 接口	495
18.7. CAU 中断	495
18.8. CAU 寄存器	497
18.8.1. 控制寄存器 (CAU_CTL)	497
18.8.2. 状态寄存器 0 (CAU_STAT0)	498
18.8.3. 数据输入寄存器 (CAU_DI)	499
18.8.4. 数据输出寄存器 (CAU_DO)	499
18.8.5. DMA 使能寄存器 (CAU_DMAEN)	500
18.8.6. 中断使能寄存器 (CAU_INTEN)	500
18.8.7. 状态寄存器 1 (CAU_STAT1)	501
18.8.8. 中断标志寄存器 (CAU_INTF)	502
18.8.9. 密钥寄存器 (CAU_KEY0..3(H/L))	502
19. 哈希处理器 (HAU)	506
19.1. 简介	506
19.2. 主要特征	506
19.3. HAU 数据类型	506
19.4. HAU 内核	508
19.4.1. 自动数据填充	508
19.4.2. 摘要计算	509
19.4.3. 哈希模式	509
19.5. HAU 中断	510
19.5.1. 输入 FIFO 中断	510
19.5.2. 计算完成中断	510
19.6. HAU 寄存器	511
19.6.1. 控制寄存器 (HAU_CTL)	511
19.6.2. 数据输入寄存器 (HAU_DI)	512

19.6.3.	配置寄存器 (HAU_CFG)	512
19.6.4.	数据输出寄存器 (HAU_DO0..7)	513
19.6.5.	中断使能寄存器 (HAU_INTEN)	516
19.6.6.	状态与标志寄存器 (HAU_STAT)	516
20.	比较器 (CMP)	518
20.1.	简介	518
20.2.	主要特征	518
20.3.	功能描述	518
20.3.1.	比较器时钟	519
20.3.2.	比较器的 I/O 配置	519
20.3.3.	比较器迟滞	520
20.3.4.	比较器寄存器写保护	520
20.3.5.	比较器输出消隐	520
20.3.6.	比较器数字滤波	521
20.3.7.	电压定标器功能	522
20.3.8.	比较器中断	522
20.4.	比较器寄存器	523
20.4.1.	状态寄存器 (CMP_STAT)	523
20.4.2.	中断标志位清除寄存器 (CMP_IFC)	523
20.4.3.	CMP0 控制状态寄存器 (CMP0_CS)	524
21.	模数转换器 (ADC)	527
21.1.	简介	527
21.2.	主要特征	527
21.3.	引脚和内部信号	528
21.4.	功能描述	529
21.4.1.	ADC 时钟	529
21.4.2.	ADC 使能	529
21.4.3.	常规序列和注入序列	529
21.4.4.	运行模式	530
21.4.5.	注入序列管理	534
21.4.6.	转换数据锁存	534
21.4.7.	转换结果阈值监测功能	536
21.4.8.	数据存储模式	536
21.4.9.	采样时间配置	537
21.4.10.	外部触发	537
21.4.11.	DMA 请求	537
21.4.12.	ADC 内部通道	538

21.4.13.	可编程分辨率 (DRES)	538
21.4.14.	片上硬件过采样	539
21.5.	ADC 同步模式	540
21.5.1.	独立模式	542
21.5.2.	常规并行模式	542
21.5.3.	注入并行模式	543
21.5.4.	常规快速交叉模式	543
21.5.5.	常规慢速交叉模式	544
21.5.6.	注入交替触发模式	545
21.5.7.	常规并行和注入并行组合模式	546
21.5.8.	常规并行和注入交替触发组合模式	546
21.5.9.	注入并行和交叉组合模式	547
21.6.	中断	548
21.7.	ADC 寄存器	549
21.7.1.	状态寄存器 (ADC_STAT)	549
21.7.2.	控制寄存器 0 (ADC_CTL0)	550
21.7.3.	控制寄存器 1 (ADC_CTL1)	552
21.7.4.	采样时间寄存器 0 (ADC_SAMPT0)	553
21.7.5.	采样时间寄存器 1 (ADC_SAMPT1)	554
21.7.6.	注入通道数据偏移寄存器 x (ADC_IOFFx) (x=0..3)	555
21.7.7.	看门狗高阈值寄存器 (ADC_WD0HT)	556
21.7.8.	看门狗低阈值寄存器 (ADC_WD0LT)	556
21.7.9.	常规序列寄存器 0 (ADC_RSQ0)	557
21.7.10.	常规序列寄存器 1 (ADC_RSQ1)	557
21.7.11.	常规序列寄存器 2 (ADC_RSQ2)	558
21.7.12.	注入序列寄存器 (ADC_ISQ)	558
21.7.13.	锁存数据寄存器 x (ADC_LDATABx) (x= 0..3)	559
21.7.14.	常规数据寄存器 (ADC_RDATA)	560
21.7.15.	注入数据寄存器 (ADC_IDATA)	560
21.7.16.	锁存数据控制寄存器 (ADC_LDCTL)	561
21.7.17.	过采样控制寄存器 (ADC_OVSAMPCTL)	562
22.	数模转换器 (DAC)	564
22.1.	简介	564
22.2.	主要特征	564
22.3.	功能描述	565
22.3.1.	DAC 使能	565
22.3.2.	DAC 输出缓冲	565
22.3.3.	DAC 数据配置	565

22.3.4.	DAC 触发	566
22.3.5.	DAC 转换	566
22.3.6.	DAC 噪声波	566
22.3.7.	DAC 输出电压.....	567
22.3.8.	DMA 请求	567
22.4.	DAC 寄存器.....	569
22.4.1.	DACx 控制寄存器 (DAC_CTL0).....	569
22.4.2.	DACx 软件触发寄存器 (DAC_SWT).....	570
22.4.3.	DACx_OUT0 12 位右对齐数据保持寄存器 (DAC_OUT0_R12DH)	571
22.4.4.	DACx_OUT0 12 位左对齐数据保持寄存器 (DAC_OUT0_L12DH)	571
22.4.5.	DACx_OUT0 8 位右对齐数据保持寄存器 (DAC_OUT0_R8DH).....	572
22.4.6.	DACx_OUT0 数据输出寄存器 (DAC_OUT0_DO).....	572
22.4.7.	DAC 状态寄存器 0 (DAC_STAT0)	573
23.	通用同步异步收发器 (USART)	574
23.1.	简介	574
23.2.	主要特征	574
23.3.	功能说明	575
23.3.1.	USART 帧格式	576
23.3.2.	波特率发生	577
23.3.3.	USART 发送器	577
23.3.4.	USART 接收器	578
23.3.5.	DMA 方式访问数据缓冲区	580
23.3.6.	硬件流控制	581
23.3.7.	多处理器通信	583
23.3.8.	LIN 模式.....	583
23.3.9.	同步通信模式	584
23.3.10.	串行红外(IrDA SIR)编解码功能模块.....	585
23.3.11.	半双工通信模式.....	586
23.3.12.	智能卡(ISO7816-3)模式	586
23.3.13.	USART 中断	588
23.4.	USART 寄存器	590
23.4.1.	状态寄存器 0 (USART_STAT0)	590
23.4.2.	数据寄存器 (USART_DATA)	592
23.4.3.	波特率寄存器 (USART_BAUD)	592
23.4.4.	控制寄存器 0 (USART_CTL0)	593
23.4.5.	控制寄存器 1 (USART_CTL1)	595
23.4.6.	控制寄存器 2 (USART_CTL2)	596
23.4.7.	保护时间和预分频器寄存器 (USART_GP)	598
23.4.8.	控制寄存器 3 (USART_CTL3)	599

23.4.9.	接收超时寄存器 (USART_RT)	601
23.4.10.	状态寄存器 1 (USART_STAT1)	601
23.4.11.	兼容性控制寄存器 (USART_CHC)	602
24.	串行外设接口/片上音频接口 (SPI/I2S)	604
24.1.	简介	604
24.2.	主要特性	604
24.2.1.	SPI 主要特性	604
24.2.2.	I2S 主要特性	604
24.3.	SPI 功能说明	605
24.3.1.	SPI 结构框图	605
24.3.2.	SPI 信号线描述	605
24.3.3.	SPI 时序和数据帧格式	606
24.3.4.	NSS 功能	607
24.3.5.	SPI 运行模式	608
24.3.6.	DMA 功能	617
24.3.7.	CRC 功能	617
24.3.8.	SPI 中断	618
24.4.	I2S 功能说明	619
24.4.1.	I2S 结构框图	619
24.4.2.	I2S 信号线描述	620
24.4.3.	I2S 音频标准	620
24.4.4.	I2S 时钟	627
24.4.5.	运行	628
24.4.6.	DMA 功能	632
24.4.7.	I2S 中断	632
24.5.	SPI/I2S 寄存器	634
24.5.1.	控制寄存器 0 (SPI_CTL0)	634
24.5.2.	控制寄存器 1 (SPI_CTL1)	636
24.5.3.	状态寄存器 (SPI_STAT)	637
24.5.4.	数据寄存器 (SPI_DATA)	639
24.5.5.	CRC 多项式寄存器 (SPI_CRCPOLY)	639
24.5.6.	接收 CRC 寄存器 (SPI_RCRC)	640
24.5.7.	发送 CRC 寄存器 (SPI_TCRC)	640
24.5.8.	I2S 控制寄存器 (SPI_I2SCTL)	641
24.5.9.	I2S 时钟预分频寄存器 (SPI_I2SPSC)	642
24.5.10.	SPI0 四路 SPI 控制寄存器 (SPI_QCTL)	643
25.	内部集成电路总线接口 (I2C)	645
25.1.	简介	645

25.2.	主要特征	645
25.3.	功能说明	645
25.3.1.	时钟要求.....	646
25.3.2.	I2C 通讯流程.....	647
25.3.3.	噪声滤波器	649
25.3.4.	I2C 时序配置.....	650
25.3.5.	I2C 复位.....	651
25.3.6.	数据传输.....	652
25.3.7.	I2C 从机模式.....	654
25.3.8.	I2C 主机模式.....	659
25.3.9.	SMBus 支持	664
25.3.10.	SMBus 模式.....	667
25.3.11.	DMA 模式下数据传输	668
25.3.12.	I2C 错误和中断.....	668
25.3.13.	I2C 调试模式.....	669
25.4.	I2C 寄存器	670
25.4.1.	控制寄存器 0 (I2C_CTL0)	670
25.4.2.	控制寄存器 1 (I2C_CTL1)	672
25.4.3.	从机地址寄存器 0 (I2C_SADDR0)	674
25.4.4.	从机地址寄存器 1 (I2C_SADDR1)	675
25.4.5.	时序寄存器 (I2C_TIMING)	676
25.4.6.	超时寄存器 (I2C_TIMEOUT)	677
25.4.7.	状态寄存器 (I2C_STAT)	678
25.4.8.	状态清除寄存器 (I2C_STATC)	681
25.4.9.	PEC 寄存器 (I2C_PEC)	682
25.4.10.	接收数据寄存器 (I2C_RDATA)	682
25.4.11.	发送数据寄存器 (I2C_TDATA)	682
25.4.12.	控制寄存器 2 (I2C_CTL2)	683
26.	控制器局域网络 (CAN)	684
26.1.	简介	684
26.2.	主要特征	684
26.3.	功能说明	685
26.3.1.	工作模式.....	685
26.3.2.	通信模式.....	686
26.3.3.	数据发送.....	687
26.3.4.	数据接收.....	689
26.3.5.	过滤功能.....	690
26.3.6.	时间触发通信	694
26.3.7.	通信参数.....	694

26.3.8.	CAN FD 操作	696
26.3.9.	传输延迟补偿	696
26.3.10.	错误标志	697
26.3.11.	中断	698
26.4.	CAN 寄存器	700
26.4.1.	控制寄存器 (CAN_CTL)	700
26.4.2.	状态寄存器 (CAN_STAT)	701
26.4.3.	发送状态寄存器 (CAN_TSTAT)	703
26.4.4.	接收 FIFO0 寄存器 (CAN_RFIFO0)	705
26.4.5.	接收 FIFO1 寄存器 (CAN_RFIFO1)	706
26.4.6.	中断使能寄存器 (CAN_INTEN)	707
26.4.7.	错误寄存器 (CAN_ERR)	709
26.4.8.	位时序寄存器 (CAN_BT)	710
26.4.9.	FD 控制寄存器 (CAN_FDCTL)	711
26.4.10.	FD 状态寄存器 (CAN_FDSTAT)	712
26.4.11.	FD 传输延迟补偿寄存器 (CAN_FDTDC)	712
26.4.12.	数据位时序寄存器 (CAN_DBT)	713
26.4.13.	发送邮箱标识符寄存器 (CAN_TMIx) (x = 0..2)	714
26.4.14.	发送邮箱属性寄存器 (CAN_TMPx) (x = 0..2)	715
26.4.15.	发送邮箱 data0 寄存器 (CAN_TMDATA0x) (x=0..2)	715
26.4.16.	发送邮箱 data1 寄存器 (CAN_TMDATA1x) (x=0..2)	716
26.4.17.	接收 FIFO 邮箱标识符寄存器 (CAN_RFIFOMIx) (x=0,1)	716
26.4.18.	接收 FIFO 邮箱属性寄存器 (CAN_RFIFOMPx) (x=0,1)	717
26.4.19.	接收 FIFO 邮箱 data0 寄存器 (CAN_RFIFOMDATA0x) (x=0,1)	718
26.4.20.	接收 FIFO 邮箱 data1 寄存器 (CAN_RFIFOMDATA1x) (x=0,1)	719
26.4.21.	过滤器控制寄存器 (CAN_FCTL)	719
26.4.22.	过滤器模式配置寄存器 (CAN_FMCFG)	720
26.4.23.	过滤器位宽配置寄存器 (CAN_FSCFG)	720
26.4.24.	过滤器关联 FIFO 寄存器 (CAN_FAFIFO)	721
26.4.25.	过滤器激活寄存器 (CAN_FW)	721
26.4.26.	过滤器(x) 数据(y) 寄存器 (CAN_FxDATAy) (x=0..27, y=0,1)	722
27.	外部存储器控制器 (EXMC)	723
27.1.	简介	723
27.2.	主要特性	723
27.3.	功能描述	723
27.3.1.	结构框图	723
27.3.2.	EXMC 访问基本规范	724
27.3.3.	外部设备地址映射	725
27.3.4.	NOR/PSRAM 控制器	727

27.3.5. NAND Flash 控制器	736
27.4. EXMC 寄存器	740
27.4.1. NOR/PSRAM 控制器寄存器	740
27.4.2. NAND Flash 控制器寄存器	743
28. 通用串行总线全速接口 (USBFS)	749
28.1. 概述	749
28.2. 主要特性	749
28.3. 结构框图	750
28.4. 信号线描述	750
28.5. 功能描述	750
28.5.1. USBFS 时钟及工作模式	750
28.5.2. USB 主机功能	751
28.5.3. USB 设备功能	753
28.5.4. 数据 FIFO	754
28.5.5. 操作手册	756
28.6. 中断	760
28.7. USBFS 寄存器	762
28.7.1. 全局控制与状态寄存器组	762
28.7.2. 主机控制和状态寄存器组	779
28.7.3. 设备控制和状态寄存器组	791
28.7.4. 电源和时钟控制寄存器 (USBFS_PWRCLKCTL)	813
29. 附录	815
29.1. 寄存器表中使用的缩写列表	815
29.2. 术语表	815
29.3. 可用外设	816
30. 版本历史	817

图索引

图 1-1. Cortex®-M33 处理器结构框图	31
图 1-2. GD32F50x 系列器件的系统架构示意图	33
图 1-3. ECC 解码器示意图	38
图 2-1. EXTI 框图	65
图 3-1. DMA 结构框图	72
图 3-2. 握手机制	74
图 3-3. DMA 中断逻辑图	76
图 4-1. DMAMUX 结构框图	84
图 4-2. 同步模式	86
图 4-3. 通道事件输出	87
图 5-1. 页擦除操作流程	102
图 5-2. 整片擦除操作流程	103
图 5-3. 字编程操作流程	105
图 6-1. 电源域概览	123
图 6-2. 上电/掉电复位波形图	125
图 6-3. LVD 阈值波形图	126
图 6-4. VAVD 阈值监测波形图	127
图 6-5. VOVD 阈值监测波形图	128
图 6-6. VUVD 阈值监测波形图	129
图 8-1. 系统复位电路	145
图 8-2. 时钟树（适用于 GD32F503xx）	146
图 8-3. 时钟树（适用于 GD32F505xx）	147
图 8-4. HXTAL 时钟源	148
图 8-5. 旁路模式下 HXTAL 时钟源	149
图 9-1. CTC 简介	192
图 9-2. CTC 校准计数器	193
图 10-1. 标准 I/O 端口位的基本结构	203
图 10-2. 输入配置的基本结构	205
图 10-3. 输出配置的基本结构	206
图 10-4. 模拟配置的基本结构	206
图 10-5. 备用功能配置的基本结构	207
图 11-1. TRGSEL 主要组成示例	227
图 12-1. 高级定时器结构框图	253
图 12-2. 内部时钟分频为 1 时正常模式下的控制电路	254
图 12-3. 当预分频器的参数从 1 变到 2 时，计数器的时序图	255
图 12-4. 向上计数时序图，PSC=0/2	256
图 12-5. 向上计数时序图，在运行时改变 TIMERx_CAR 寄存器的值	256
图 12-6. 向下计数时序图，PSC=0/2	257

图 12-7. 向下计数时序图, 在运行时改变 TIMERx_CAR 寄存器值	258
图 12-8. 中央计数模式计数器时序图	259
图 12-9. 中央计数模式下计数器重复时序图	260
图 12-10. 在向上计数模式下计数器重复时序图	260
图 12-11. 在向下计数模式下计数器重复时序图	261
图 12-12. 通道 0 输入捕获逻辑	262
图 12-13. 输出比较逻辑	263
图 12-14. 三种输出比较模式	264
图 12-15. EAPWM 时序图	265
图 12-16. CAPWM 时序图	265
图 12-17. 通道 x 输出 PWM ($CHxVAL < CHxCOMVAL\_ADD$)	267
图 12-18. 通道 x 输出 PWM ($CHxVAL = CHxCOMVAL\_ADD$)	267
图 12-19. 通道 x 输出 PWM ($CHxVAL > CHxCOMVAL\_ADD$)	268
图 12-20. 通道 x 输出 PWM ($CHxVAL$ 或 $CHxCOMVAL\_ADD > CARL$)	268
图 12-21. 通道 x 输出 PWM 占空比随着 $CHxCOMVAL\_ADD$ 值而改变	269
图 12-22. 复合 PWM 模式下四通道输出	269
图 12-23. 通道 x 输出 PWM ($CHxVAL < CHxCOMVAL\_ADD / CHxCOMVAL\_ADD < CHxVAL$)	271
图 12-24. 通道 x 输出 PWM ($CHxVAL = CHxCOMVAL\_ADD$)	272
图 12-25. 通道 x 输出 PWM ($CHxVAL$ 或 $CHxCOMVAL\_ADD > CARL$)	273
图 12-26. 边沿对齐模式下 CHx_O 输出脉冲 ($CHxOMPSEL \neq 2'b00$)	274
图 12-27. 中央对齐模式下 CHx_O 输出脉冲 ($CHxOMPSEL \neq 2'b00$)	274
图 12-28. 带死区时间的互补输出	277
图 12-29. 不同死区时间的互补输出 ($DTDIFEN=1$)	277
图 12-30. BREAK 中止功能逻辑图	278
图 12-31. 通道响应 BREAK 中止输入 (高电平有效) 时, 输出信号的行为 ($IOS=1$)	279
图 12-32. 通道中止功能逻辑图	280
图 12-33. 译码器接口模式下计数器运行例子	281
图 12-34. CI0FE0 极性反相的译码器接口模式下的例子	281
图 12-35. 霍尔传感器用在 BLDC 电机控制中	282
图 12-36. 两个定时器之间的霍尔传感器时序图	283
图 12-37. 复位模式	284
图 12-38. 暂停模式下的控制电路	284
图 12-39. 事件模式	285
图 12-40. 单脉冲模式, $TIMERx\_CHxCV = 0x04$ $TIMERx\_CAR=0x60$	285
图 12-41. 用定时器 2 的使能信号触发定时器 0	286
图 12-42. 用定时器 2 的更新事件来触发定时器 0	287
图 12-43. 用定时器 2 的使能来选通定时器 0	288
图 12-44. 用定时器 2 的 O0CPRE 信号选通定时器 0	288
图 12-45. 用定时器 2 的 CI0 输入来触发定时器 0 和定时器 2	289
图 12-46. 用主计数器的 trigger 信号触发 4 个计数器	290
图 12-47. CINITDIR 为 0 或 1 时的计数方向	290

图 12-48. 通用定时器 L0 结构框图.....	337
图 12-49. 内部时钟分频为 1 时正常模式下的控制电路.....	338
图 12-50. 当预分频器的参数从 1 变到 2 时, 计数器的时序图	339
图 12-51. 向上计数时序图, PSC=0/2.....	340
图 12-52. 向上计数时序图, 在运行时改变 TIMEx_CAR 寄存器的值	341
图 12-53. 向下计数时序图, PSC=0/2.....	342
图 12-54. 向下计数时序图, 在运行时改变 TIMEx_CAR 寄存器值	343
图 12-55. 中央计数模式计数器时序图	344
图 12-56. 输入捕获逻辑.....	345
图 12-57. 输出比较逻辑 (x=0,1,2,3)	346
图 12-58. 三种输出比较模式	347
图 12-59. EAPWM 时序图.....	348
图 12-60. CAPWM 时序图	348
图 12-61. 译码器接口模式下计数器运行例子.....	350
图 12-62. CIOFE0 极性反相的译码器接口模式下的例子	350
图 12-63. 正交译码器信号断线检测框图	351
图 12-64. 译码器模式 0/1 计数器运行实例 (CH1P=0)	352
图 12-65. 译码器模式 2 / 3 计数器运行实例 (CH0P / CH1P=0)	352
图 12-66. 复位模式	354
图 12-67. 暂停模式	354
图 12-68. 事件模式	355
图 12-69. 单脉冲模式, TIMEx_CHxCV = 0x04 TIMEx_CAR=0x60	355
图 12-70. 通用定时器 L3 结构框图.....	383
图 12-71. 内部时钟分频为 1 时正常模式下的控制电路.....	384
图 12-72. 当预分频器的参数从 1 变到 2 时, 计数器的时序图	385
图 12-73. 向上计数时序图, PSC=0/2.....	386
图 12-74. 向上计数时序图, 在运行时改变 TIMEx_CAR 寄存器的值	386
图 12-75. 向下计数时序图, PSC=0/2.....	387
图 12-76. 向下计数时序图, 在运行时改变 TIMEx_CAR 寄存器值	388
图 12-77. 中央计数模式计数器时序图.....	389
图 12-78. 中央计数模式下计数器重复时序图.....	390
图 12-79. 在向上计数模式下计数器重复时序图.....	390
图 12-80. 在向下计数模式下计数器重复时序图.....	391
图 12-81. 通道 0 输入捕获逻辑	392
图 12-82. 多模式通道 0 输入捕获逻辑.....	392
图 12-83. 输出比较逻辑 (当 MCHxMSEL = 2'00 时, x=0)	393
图 12-84. 输出比较逻辑 (当 MCHxMSEL = 2'11 时, x=0)	394
图 12-85. 输出比较逻辑 (x=1)	394
图 12-86. 三种输出比较模式	396
图 12-87. EAPWM 时序图.....	397
图 12-88. CAPWM 时序图	397

图 12-89. 通道 x 输出 PWM (CHxVAL < CHxCOMVAL_ADD)	399
图 12-90. 通道 x 输出 PWM (CHxVAL = CHxCOMVAL_ADD)	399
图 12-91. 通道 x 输出 PWM (CHxVAL > CHxCOMVAL_ADD)	400
图 12-92. 通道 x 输出 PWM (CHxVAL 或 CHxCOMVAL_ADD > CARL)	400
图 12-93. 通道 x 输出 PWM 占空比随着 CHxCOMVAL_ADD 值而改变	401
图 12-94. 边沿对齐模式下 CHx_O 输出脉冲 (CHxOMPSEL≠2'b00)	402
图 12-95. 中央对齐模式下 CHx_O 输出脉冲 (CHxOMPSEL≠2'b00)	402
图 12-96. 带死区时间的互补输出	405
图 12-97. BREAK 中止功能逻辑图	406
图 12-98. 通道响应 BREAK 中止输入 (高电平有效) 时, 输出信号的行为 (IOS=1)	406
图 12-99. 复位模式下的控制电路	407
图 12-100. 暂停模式下的控制电路	408
图 12-101. 事件模式下的控制电路	408
图 12-102. 单脉冲模式, TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60	409
图 12-103. 基本定时器结构框图	443
图 12-104. 内部时钟分频为 1 时正常模式下的控制电路	444
图 12-105. 当预分频器的参数从 1 变到 2 时, 计数器的时序图	444
图 12-106. 向上计数时序图, PSC=0/2 (TIMERx, x=5,6)	445
图 12-107. 向上计数时序图, 在运行时改变 TIMERx_CAR 寄存器的值 (TIMERx, x=5,6)	446
图 13-1. RTC 框图	453
图 13-2. RTC 秒信号及闹钟信号的波形 (RTC_PSC = 3, RTC_ALRM = 2)	454
图 13-3. RTC 秒信号及溢出信号的波形 (RTC_PSC = 3)	455
图 14-1. 独立看门狗定时器框图	463
图 14-2. 窗口看门狗定时器框图	468
图 14-3. 窗口看门狗定时器时序图	469
图 15-1. CRC 计算单元框图	473
图 17-1. TRNG 模块框图	485
图 18-1. DATAM 不交换/半字交换	491
图 18-2. DATATM 字节交换/位交换	491
图 18-3. CAU 框图	492
图 18-4. AES ECB 加密	493
图 18-5. AES ECB 解密	494
图 19-1. DATAM 不交换/半字交换	507
图 19-2. DATAM 字节交换/位交换	507
图 19-3. HAU 结构框图	508
图 20-1. 比较器框图	519
图 20-2. 比较器迟滞	520
图 20-3. 比较器的输出消隐	521
图 20-4. 比较器噪声滤波和中断操作	521
图 21-1. ADC 模块框图	529
图 21-2. 单次运行模式	530

图 21-3. 连续运行模式	531
图 21-4. 扫描运行模式，且连续运行模式禁能	532
图 21-5. 扫描运行模式，连续运行模式使能	532
图 21-6. 间断运行模式	533
图 21-7. 自动注入，CTN=1	534
图 21-8. 触发注入	534
图 21-9. 数据锁存	535
图 21-10. ADC_LDATABx 读取序列数据	535
图 21-11. 12 位数据存储模式	536
图 21-12. 6 位数据存储模式	537
图 21-13. 20 位到 16 位的结果截断	539
图 21-14. 右移 5 位和取整的数例	540
图 21-15. ADC 同步框图	542
图 21-16. 基于 16 个通道的常规并行模式	543
图 21-17. 4 个通道的注入并行模式	543
图 21-18. 常规序列上的快速交叉模式（两个 ADC 的 CTN=1）	544
图 21-19. 常规序列上的慢速交叉模式	545
图 21-20. 注入交替触发模式：DISIC=0, IL=1	545
图 21-21. 注入交替触发模式：DISIC=1, IL=1	546
图 21-22. 常规并行和交替触发组合模式	547
图 21-23. 在注入序列转换过程中注入触发出现	547
图 21-24. 单通道交叉转换被注入序列中断	547
图 22-1. DAC 结构框图	564
图 22-2. DAC LFSR 算法	567
图 22-3. DAC 三角噪声模式生成的波形	567
图 23-1. USART 模块内部框图	576
图 23-2. USART 字符帧 (8 数据位和 1 停止位)	576
图 23-3. USART 发送步骤	578
图 23-4. 过采样方式接收一个数据位 (OSB=0)	579
图 23-5. 采用 DMA 方式实现 USART 数据发送配置步骤	580
图 23-6. 采用 DMA 方式实现 USART 数据接收配置步骤	581
图 23-7. 两个 USART 之间的硬件流控制	582
图 23-8. 硬件流控制	582
图 23-9. 空闲状态下检测断开帧	584
图 23-10. 数据传输过程中检测断开帧	584
图 23-11. 同步模式下的 USART 示例	584
图 23-12. 8-bit 格式的 USART 同步通信波形(CLEN=1)	585
图 23-13. IrDA SIR ENDEC 模块	585
图 23-14. IrDA 数据调制	586
图 23-15. ISO7816-3 数据帧格式	587
图 23-16. USART 中断映射框图	589

图 24-1. SPI 结构框图	605
图 24-2. 常规模式下的 SPI 时序图	606
图 24-3. SPI 四线模式下的 SPI 时序图(CKPL=1, CKPH=1, LF=0)	607
图 24-4. 典型的全双工模式连接	609
图 24-5. 典型的单工模式连接（主机：接收，从机：发送）	610
图 24-6. 典型的单工模式连接（主机：只发送，从机：接收）	610
图 24-7. 典型的双向线连接	610
图 24-8. 主机 TI 模式在不连续发送时的时序图	612
图 24-9. 主机 TI 模式在连续发送时的时序图	612
图 24-10. 从机 TI 模式时序图	613
图 24-11. NSS 脉冲模式时序图（主机连续发送）	614
图 24-12. SPI 四线模式四线写操作时序图	615
图 24-13. SPI 四路模式四路读操作时序图	616
图 24-14. I2S 结构框图	619
图 24-15. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=0, CKPL=0)	620
图 24-16. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=0, CKPL=1)	620
图 24-17. I2S 飞利浦标准时序图(DTLEN=10, CHLEN=1, CKPL=0)	621
图 24-18. I2S 飞利浦标准时序图(DTLEN=10, CHLEN=1, CKPL=1)	621
图 24-19. I2S 飞利浦标准时序图(DTLEN=01, CHLEN=1, CKPL=0)	621
图 24-20. I2S 飞利浦标准时序图(DTLEN=01, CHLEN=1, CKPL=1)	621
图 24-21. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=1, CKPL=0)	622
图 24-22. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=1, CKPL=1)	622
图 24-23. MSB 对齐标准时序图(DTLEN=00, CHLEN=0, CKPL=0)	622
图 24-24. MSB 对齐标准时序图(DTLEN=00, CHLEN=0, CKPL=1)	622
图 24-25. MSB 对齐标准时序图(DTLEN=10, CHLEN=1, CKPL=0)	622
图 24-26. MSB 对齐标准时序图(DTLEN=10, CHLEN=1, CKPL=1)	623
图 24-27. MSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=0)	623
图 24-28. MSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=1)	623
图 24-29. MSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=0)	623
图 24-30. MSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=1)	623
图 24-31. LSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=0)	624
图 24-32. LSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=1)	624
图 24-33. LSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=0)	624
图 24-34. LSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=1)	624
图 24-35. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=0)	625
图 24-36. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=1)	625
图 24-37. PCM 标准短帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=0)	625
图 24-38. PCM 标准短帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=1)	625
图 24-39. PCM 标准短帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=0)	625
图 24-40. PCM 标准短帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=1)	625
图 24-41. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=0)	626

图 24-42. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=1)	626
图 24-43. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=0)	626
图 24-44. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=1)	626
图 24-45. PCM 标准长帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=0)	626
图 24-46. PCM 标准长帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=1)	626
图 24-47. PCM 标准长帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=0)	627
图 24-48. PCM 标准长帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=1)	627
图 24-49. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=0)	627
图 24-50. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=1)	627
图 24-51. I2S 时钟生成结构框图	627
图 24-52. I2S 初始化流程	629
图 24-53. I2S 主机接收禁能流程	631
图 25-1. I2C 模块框图	646
图 25-2. 数据有效性	647
图 25-3. 开始和停止信号	648
图 25-4. 10 位地址的 I2C 通讯流程（主机发送）	648
图 25-5. 7 位地址的 I2C 通讯流程（主机发送）	649
图 25-6. 7 位地址的 I2C 通讯流程（主机接收）	649
图 25-7. 10 位地址的 I2C 通讯流程（主机接收, HEAD10R=0）	649
图 25-8. 10 位地址的 I2C 通讯流程（主机接收, HEAD10R=1）	649
图 25-9. 数据保持时间	650
图 25-10. 数据建立时间	650
图 25-11. 数据发送	652
图 25-12. 数据接收	653
图 25-13. I2C 从机初始化	656
图 25-14. I2C 从机发送编程模型（SS=0）	657
图 25-15. I2C 从机发送编程模型（SS=1）	658
图 25-16. I2C 从机接收编程模型	659
图 25-17. I2C 主机初始化	660
图 25-18. I2C 主机发送编程模型（N<=255）	661
图 25-19. I2C 主机发送编程模型（N>255）	662
图 25-20. I2C 主机接收编程模型（N<=255）	663
图 25-21. I2C 主机接收编程模型（N>255）	664
图 25-22. SMBus 主机发送器和从机接收器通信流程	667
图 25-23. SMBus 主机接收器和从机发送器通信流程	668
图 26-1. CAN 模块结构框图	685
图 26-2. 发送寄存器	687
图 26-3. 发送邮箱状态转换	688
图 26-4. 接收寄存器	689
图 26-5. 32-bit 位宽过滤器	691
图 26-6. 16-bit 位宽过滤器	691

图 26-7. 32-bit 位宽掩码模式过滤器	691
图 26-8. 16-bit 位宽掩码模式过滤器	691
图 26-9. 32-bit 位宽列表模式过滤器	691
图 26-10. 16-bit 位宽列表模式过滤器	692
图 26-11. 位时序	695
图 26-12. 传输延迟测量	697
图 27-1. 系统架构	724
图 27-2. EXMC Bank 划分	725
图 27-3. Bank0 地址映射	726
图 27-4. NAND 地址映射	726
图 27-5. Bank1 通用空间	727
图 27-6. 复用模式读时序	730
图 27-7. 复用模式写时序	730
图 27-8. 异步等待有效时的读时序	732
图 27-9. 异步等待有效时的写时序	732
图 27-10. 同步复用突发传输读时序	734
图 27-11. 同步复用突发传输写时序	735
图 27-12. NAND flash 通用空间操作时序	738
图 27-13. NCE 敏感 NAND Flash 访问时序	739
图 28-1. USBFS 结构框图	750
图 28-2. 在主机或设备模式下连接示意图	751
图 28-3. 主机端口状态转移图	752
图 28-4. 主机模式 FIFO 空间	755
图 28-5. 主机模式 FIFO 访问寄存器映射表	755
图 28-6. 设备模式 FIFO 空间	756
图 28-7. 设备模式 FIFO 访问寄存器映射表	756

表索引

表 1-1. AHB 互联矩阵的互联关系列表.....	31
表 1-2. GD32F50x 系列器件的存储器映射表	34
表 1-3. 引导模式.....	40
表 2-1. Cortex®-M33 中的 NVIC 异常类型	61
表 2-2. 中断向量表.....	62
表 2-3. EXTI 触发源.....	66
表 3-1. DMA 传输操作.....	72
表 3-2. 中断事件.....	75
表 4-1. DMAMUX 信号	85
表 4-2. 中断事件.....	89
表 4-3. DMAMUX 请求路由输入信号映射.....	89
表 4-4. 触发输入信号映射	92
表 4-5. 同步输入信号映射	93
表 5-1. GD32F50x 闪存基地址和构成.....	100
表 5-2. 选项字节.....	106
表 5-3. 页保护 WP 位.....	107
表 5-4. OTP 可用编程位宽	108
表 5-5. OTP0 锁.....	108
表 5-6. OTP1 锁.....	108
表 5-7. OTP2 锁.....	109
表 5-8. OTP3 锁.....	109
表 5-9. 安全保护.....	110
表 6-1. 节电模式总结	131
表 8-1. 时钟输出的时钟源选择.....	151
表 8-2. 深度睡眠模式下 V _{CORE} 域电压选择	152
表 10-1. GPIO 配置表.....	203
表 11-1. 触发输入位域选择	227
表 11-2. TRIGSEL 输入输出映射关系	229
表 12-1. 定时器 (TIMERx) 分为 4 种类型.....	250
表 12-2 复合 PWM 脉冲宽度.....	266
表 12-3 非对称 PWM 脉冲宽度.....	270
表 12-4. 由参数控制的互补输出表.....	275
表 12-5. BREAK 输入信号时, TIMER 互补通道输出情况 (break 输入高电平有效)	279
表 12-6. 不同译码器模式下的计数方向	280
表 12-7. 从模式例子列表.....	283
表 12-8. 不同译码器模式下的计数方向	349
表 12-9. 译码器模式 1 的计数情况.....	351
表 12-10. 译码器模式 2 / 3 下的计数方向	353

表 12-11. 从模式列表和举例（通用定时器 L0）	353
表 12-12. 复合 PWM 脉冲宽度.....	398
表 12-13. 由参数控制的互补输出表（MCHxMSEL=2'b11）	403
表 12-14. 从模式例子列表.....	407
表 14-1. 独立看门狗定时器在 40kHz（IRC40K）时的最小 / 最大超时周期.....	463
表 14-2. 在 140MHz (f _{PCLK1}) 时的最大/最小超时值.....	469
表 16-1. 调试引脚分配.....	479
表 20-1. 输入和输出汇总	519
表 21-1. ADC 内部信号	528
表 21-2. ADC 引脚定义	528
表 21-3. 外部触发模式和触发类型.....	537
表 21-4. 不同分辨率对应的 t _{CONV} 时间.....	538
表 21-5. N 和 M 的最大输出值（灰色部分表示截断）	540
表 21-6. ADC 同步模式表	541
表 22-1. DAC 引脚.....	565
表 22-2. DAC 触发与输出	565
表 22-3. DAC 外部触发	566
表 23-1. USART 重要引脚描述.....	575
表 23-2. 停止位配置.....	576
表 23-3. USART 中断请求	588
表 24-1. SPI 信号描述	605
表 24-2. SPI 四线信号描述.....	606
表 24-3. 从机模式 NSS 功能	607
表 24-4. 主机模式 NSS 功能	608
表 24-5. SPI 运行模式	608
表 24-6. SPI 中断请求	619
表 24-7. I2S 比特率计算公式	628
表 24-8. 音频采样频率计算公式.....	628
表 24-9. 各种运行模式下 I2S 接口信号的方向.....	628
表 24-10. I2S 中断.....	633
表 25-1. I2C 总线术语说明（参考飞利浦 I2C 规范）	646
表 25-2. 数据建立时间和数据保持时间	651
表 25-3. 可关闭通信模式.....	653
表 25-4. I2C 错误标志	668
表 25-5. I2C 中断事件	668
表 26-1. 32-bit 过滤序号	692
表 26-2. 过滤索引	692
表 26-3. CAN 事件/中断标志	699
表 27-1. NOR Flash 接口信号描述	728
表 27-2. PSRAM 复用接口信号描述	728
表 27-3. EXMC 的 Bank0 支持的访问模式.....	728

表 27-4. NOR/PSRAM 控制时序参数.....	729
表 27-5. EXMC 时序模型	729
表 27-6. 复用模式相关寄存器配置.....	730
表 27-7. 同步复用模式读时序配置.....	734
表 27-8. 同步复用模式写时序配置.....	735
表 27-9. 8 位/16 位 NAND 接口信号描述.....	736
表 27-10. Bank1 支持的访问模式	736
表 27-11. NADN Flash 可编程参数.....	737
表 28-1. USBFS 信号线描述.....	750
表 28-2. USBFS 全局中断.....	760
表 29-1. 寄存器功能位访问属性.....	815
表 29-2. 术语	815
表 30-1. 版本历史	817

1. 系统及存储器架构

GD32F50x 系列器件是基于 Arm® Cortex®-M33 处理器的 32 位通用微控制器。Arm® Cortex®-M33 处理器包括数据总线和系统总线这两条 AHB 总线。Cortex®-M33 处理器的所有存储访问，根据不同的目的和目标存储空间，都会在这两条 AHB 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。

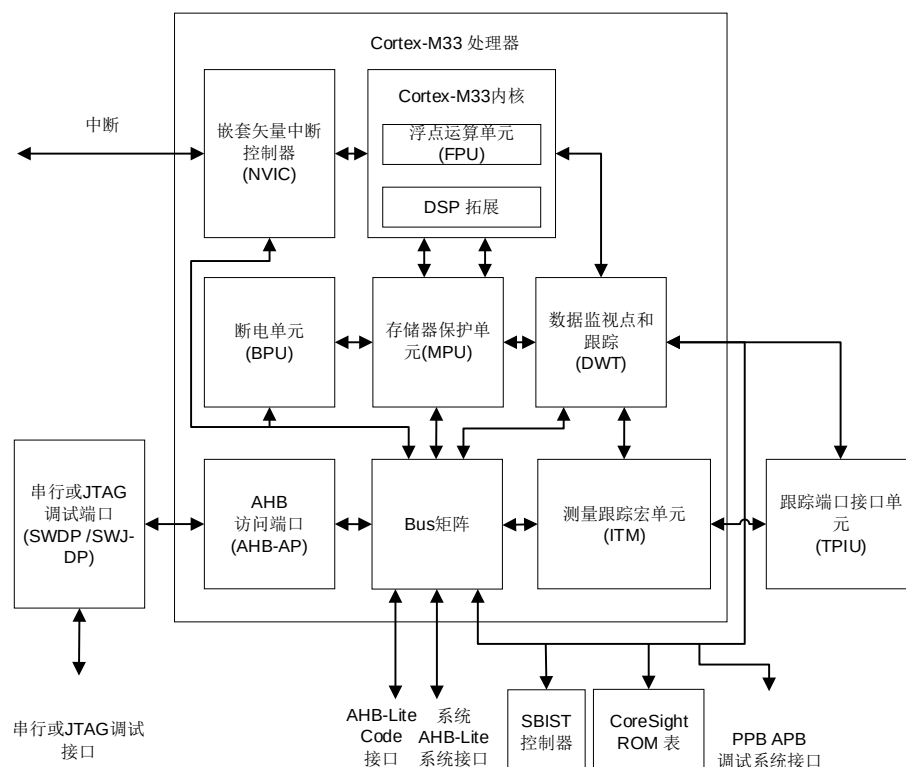
1.1. Arm® Cortex®-M33 处理器

Cortex®-M33 处理器是一个 32 位处理器，具有低中断延迟和低成本调试的特点。集成和先进的特性使 Cortex®-M33 处理器适合于需要高性能和低功耗微控制器的市场产品。Cortex®-M33 处理器基于 Armv8 架构，支持强大的可扩展指令集，包括通用数据处理 I/O 控制任务、增强的数据处理域操作和 DSP。下面列出由 Cortex®-M33 提供的一些系统外设：

- 与代码总线、系统总线和专用外围总线（PPB）相连的内部总线矩阵和调试访问；
- 嵌套矢量中断控制器（NVIC）；
- 断点单元（BPU）；
- 数据监视点和跟踪（DWT）；
- 测量跟踪宏单元（ITM）；
- 串行 JTAG 调试端口（SWJ-DP）；
- 跟踪端口接口单元（TPIU）；
- 存储器保护单元（MPU）；
- 浮点运算单元（FPU）；
- DSP 拓展（DSP）
- 软件内置自检（SBIST）控制器。

[图 1-1. Cortex®-M33 处理器结构框图](#)显示了 Cortex®-M33 处理器结构框图。欲了解更多信息，请参阅 Arm® Cortex®-M33 技术参考手册。

图 1-1. Cortex®-M33 处理器结构框图



1.2. 系统架构

GD32F50x 系列器件采用 32 位多层总线结构，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。多层总线结构包括一个 AHB 互联矩阵、一个 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以通过 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

	CBUS	SBUS	DMA0	DMA1
FMC	1		1	1
SRAM	1	1	1	1
AHB		1	1	1
EXMC	1	1	1	1

如[表 1-1. AHB 互联矩阵的互联关系列表](#)所示，AHB 互联矩阵共连接多个主机，分别为：CBUS、SBUS、DMA0、DMA1。CBUS 是 Cortex®-M33 内核的指令总线，用于从代码区域中取指令和向

量。SBUS 是 Cortex®-M33 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 的总线。

GD32F50x 系列芯片内置了主机端口超时监测机制，用于检测系统总线传输超时情况。

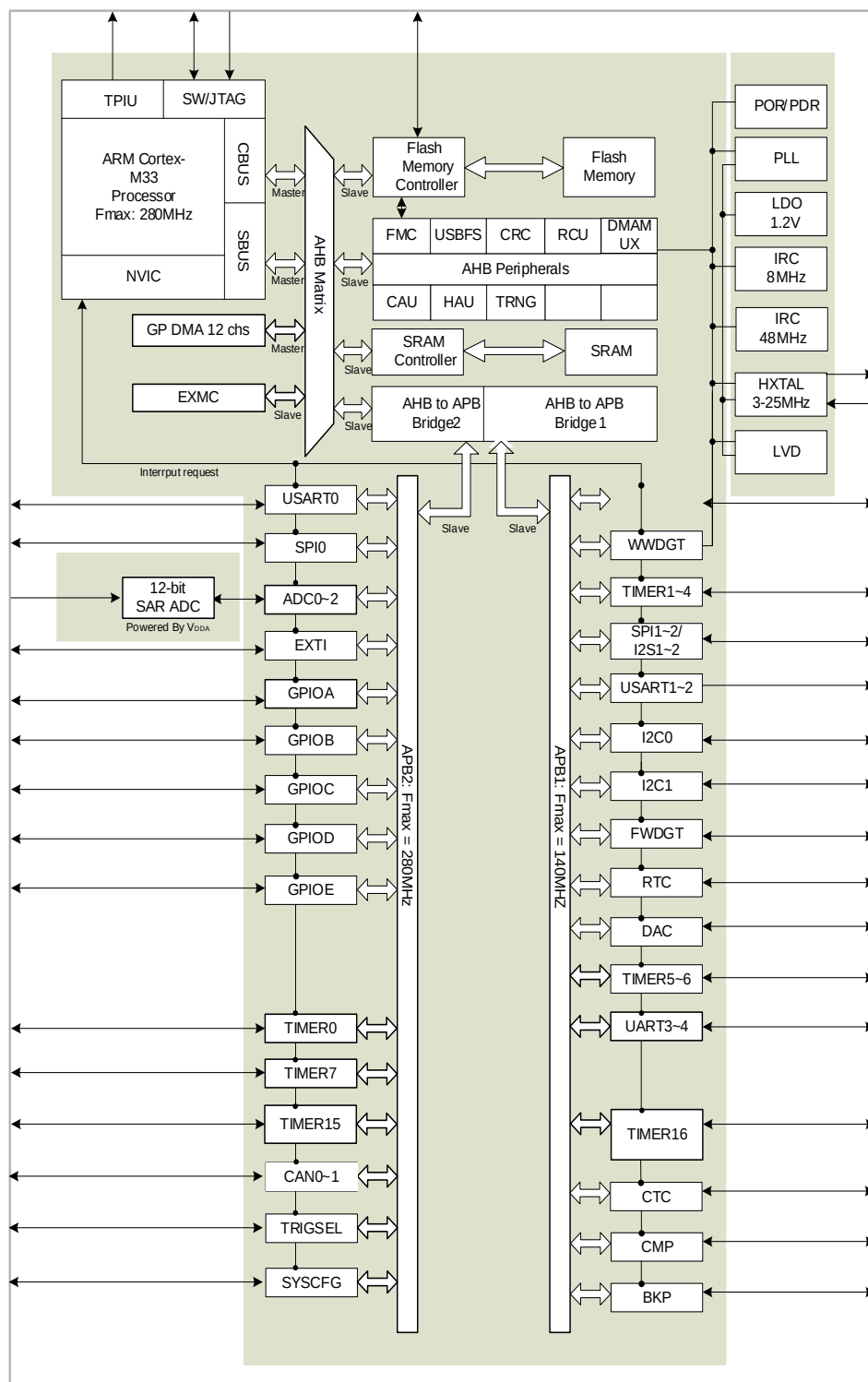
系统对 4 个 AHB 主机端口（CPU CBUS、CPU SBUS、DMA0、DMA1）进行超时监测。当访问异常导致长时间卡住总线访问时，硬件自动产生总线错误反馈给相应的 master，同时 SYSCFG 模块设置相应的超时状态标志位并触发 NMI 中断进行异常处理。

通过总线超时寄存器（SYSCFG_BUSTO）使能对应的主机端口超时监测功能，通过总线超时状态寄存器（SYSCFG_BUSTOSTAT）查看各主机端口的超时状态标志位。

AHB 互联矩阵也连接了多个从机，分别为：FMC、SRAM、EXMC、AHB、APB1 和 APB2。FMC 是闪存存储器控制器。SRAM 是片上静态随机存取存储器。EXMC 是外部存储器控制器。AHB 连接所有 AHB 从机的 AHB 总线，而 APB1 和 APB2 是连接所有 APB 从机的两条 APB 总线。APB1 最高可达 140MHz，APB2 可以全速运行（最高可到 280MHz）。

GD32F50x 系列器件的系统架构如 [图 1-2. GD32F50x 系列器件的系统架构示意图](#) 所示。

图 1-2. GD32F50x 系列器件的系统架构示意图



注意：GD32F503xx 系统时钟的最大运行时钟频率可以达到 252MHz，GD32F505xx 系统时钟的最大运行时钟频率可以达到 280MHz。

1.3. 存储器映射

Arm® Cortex®-M33 处理器采用哈佛架构，可以使用单独的总线来提取指令和加载/存储数据。程序存储器，数据存储器，寄存器和 I/O 端口组织在同一线性 4GB 地址空间内，这是 Cortex®-M33 的最大地址范围，因为总线地址宽度为 32 位。此外，Cortex®-M33 处理器提供了预定义的内存映射，以减少不同客户在相同应用时的软件复杂度。在内存映射表中，Arm® Cortex®-M33 系统外围设备使用的某些区域无法修改。但是，其他区域可供供应商使用。[表 1-2. GD32F50x 系列器件的存储器映射表](#)显示了 GD32F50x 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. GD32F50x 系列器件的存储器映射表

预先定义的地 址空间	总线	地址	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	EXMC - NAND
		0x6000 0000 - 0x6FFF FFFF	EXMC - NOR/PSRAM
外设	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	TRNG
		0x4002 3800 - 0x4002 3BFF	HAU
		0x4002 3400 - 0x4002 37FF	CAU
		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

预先定义的地 址空间	总线	地址	外设
		0x4002 1400 - 0x4002 17FF	保留
		0x4002 1000 - 0x4002 13FF	RCU
		0x4002 0C00 - 0x4002 0FFF	保留
		0x4002 0800 - 0x4002 0BFF	DMAMUX
		0x4002 0400 - 0x4002 07FF	DMA1
		0x4002 0000 - 0x4002 03FF	DMA0
		0x4001 8400 - 0x4001 FFFF	保留
		0x4001 8000 - 0x4001 83FF	保留
	APB2	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 6500 - 0x4001 67FF	保留
		0x4001 6400 - 0x4001 64FF	Shared CAN/APB SRAM1
		0x4001 6000 - 0x4001 63FF	Shared CAN/APB SRAM0
		0x4001 5C00 - 0x4001 5FFF	CAN1
		0x4001 5800 - 0x4001 5BFF	CAN0
		0x4001 5400 - 0x4001 57FF	保留
		0x4001 5000 - 0x4001 53FF	TIMER15
		0x4001 4C00 - 0x4001 4FFF	保留
		0x4001 4800 - 0x4001 4BFF	保留
		0x4001 4400 - 0x4001 47FF	TRIGSEL
		0x4001 4000 - 0x4001 43FF	SYSCFG
		0x4001 3C00 - 0x4001 3FFF	ADC2
		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
		0x4001 1800 - 0x4001 1BFF	GPIOE
		0x4001 1400 - 0x4001 17FF	GPIOD

预先定义的地 址空间	总线	地址	外设
		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
	APB1	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	CMP
		0x4000 7400 - 0x4000 77FF	DAC
		0x4000 7000 - 0x4000 73FF	PMU
		0x4000 6C00 - 0x4000 6FFF	BKP
		0x4000 6800 - 0x4000 6BFF	保留
		0x4000 6400 - 0x4000 67FF	保留
		0x4000 6000 - 0x4000 63FF	保留
		0x4000 5C00 - 0x4000 5FFF	保留
		0x4000 5800 - 0x4000 5BFF	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	保留
		0x4000 1C00 - 0x4000 1FFF	保留
		0x4000 1800 - 0x4000 1BFF	TIMER16
		0x4000 1400 - 0x4000 17FF	TIMER6
		0x4000 1000 - 0x4000 13FF	TIMER5

预先定义的地 址空间	总线	地址	外设
		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
		0x2000 0000 - 0x2002 FFFF	SRAM
代码	AHB	0x1FFF F810 - 0x1FFF FFFF	保留
		0x1FFF F800 - 0x1FFF F80F	Option Bytes
		0x1FFF B000 - 0x1FFF F7FF	Boot loader
		0x1FFF 7A10 - 0x1FFF AFFF	保留
		0x1FFF 7940 - 0x1FFF 7A0F	保留
		0x1FFF 7930 - 0x1FFF 793F	OTP3(lock area)
		0x1FFF 7900 - 0x1FFF 792F	OTP3(data area)
		0x1FFF 7880 - 0x1FFF 788F	保留
		0x1FFF 7840 - 0x1FFF 787F	OTP0(lock area)
		0x1FFF 7800 - 0x1FFF 783F	OTP0(data area)
		0x1FFF 0000 - 0x1FFF 77FF	保留
		0x1FFE C010 - 0x1FFE FFFF	保留
		0x1FFE C000 - 0x1FFE C00F	保留
		0x1FF2 0190 - 0x1FFE BFFF	保留
		0x1FF2 0110 - 0x1FF2 018F	OTP2 (lock area)
		0x1FF2 0100 - 0x1FF2 010F	OTP1 (lock area)
		0x1FF2 0000 - 0x1FF2 00FF	OTP2 (data area)
		0x1FF0 0000 - 0x1FF1 FFFF	OTP1 (data area)
		0x1001 0000 - 0x1FEF FFFF	保留
		0x1000 0000 - 0x1000 FFFF	保留
		0x083C 0000 - 0x0FFF FFFF	保留
		0x0830 0000 - 0x083B FFFF	保留
		0x0810 0000 - 0x082F FFFF	保留
		0x0800 0000 - 0x080F FFFF	Main Flash
		0x0030 0000 - 0x07FF FFFF	保留
		0x0010 0000 - 0x002F FFFF	Aliased to Main
		0x0002 0000 - 0x000F FFFF	Flash or Boot loader

预先定义的地 址空间	总线	地址	外设
		0x0000 0000 - 0x0001 FFFF	

1.3.1. 片上 SRAM 存储器

GD32F50x 系列微控制器含有高达 192KB 的片上 SRAM（起始地址为 0x2000 0000）。支持字节、半字（16 比特）和整字（32 比特）访问。

ECC

当读写前 32KB SRAM 时，SRAM 支持 7 比特的 ECC 功能。可纠错 1 比特，发现多比特（两比特）错误。

读之前必须先写入，否则很可能会导致 ECC 错误。非对齐的读操作会按照 32 比特的读操作来执行。非对齐的写操作会产生一个读改写的流程。例如，16 比特写，首先会先读 16 比特，再和需要写入的 16 比特一起写入。所以初始化 SRAM 时，只能按照 32 位的来写入。

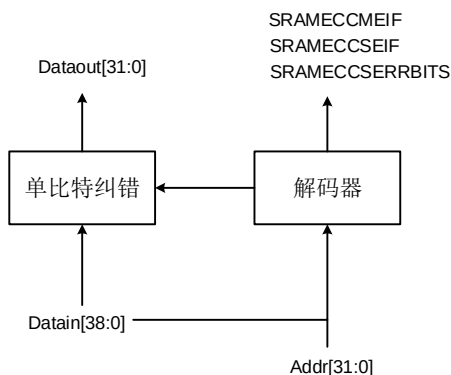
ECC 模块由编码器和解码器两部分构成：

编码器：在进行 SRAM 写操作时，会产生一个 7 比特的 ECC 码，和数据一起写入 SRAM。

解码器：在进行 SRAM 读操作时，使用与编码器相同的算法，解码生成一个 7 比特的 ECC 码。ECC 码包括 ECC 错误状态和 32 位数据中哪位存在单比特位错误的信息。

解码器如下 [图 1-3. ECC 解码器示意图](#) 所示：

图 1-3. ECC 解码器示意图



EEIC

EEIC（ECC Error Interrupt Control）模块提供了 ECC 错误状态管理和 ECC 中断配置的功能。

单比特可纠错事件

当检测到发生了单比特可纠错的事件时，EEIC 模块有如下配置：

- (1) 在 SYSCFG_SRAM ECCSTAT 寄存器中的 SRAMECCSEIF 位将被置位。软件可以通过写入 1 来清除它。
- (2) SYSCFG_SRAM ECCCS 记录发生单比特校正错误事件的地址。

多比特（两比特）不可纠错事件

当检测到发生了多比特（两比特）不可纠错的事件时，EEIC 模块有如下配置：

- (1) 在 SYSCFG_SRAM ECCSTAT 寄存器中的 SRAMECCMEIF 位将被置位。软件可以通过写入 1 来清除它。
- (2) SYSCFG_SRAM ECCCS 记录发生多比特非校正错误事件的地址。

单比特可纠错中断

在 SYSCFG_SRAM ECCCS 寄存器中设置 SRAMECCSEIE 位，当检测到一个单比特可纠正错误事件时，将产生 NMI 和 SRAM ECC 中断。

多比特（两比特）不可纠错中断

在 SYSCFG_SRAM ECCCS 寄存器中设置 SRAMECCMEIE 位。当检测到一个多比特（两比特）不可纠错错误事件时，将产生 NMI 和 SRAM ECC 中断。

1.3.2. 片上 FLASH 存储器概述

GD32F50x 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1024KB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

GD32F50x 设备提供四种启动源。启动模式受到安全保护、OTP3 中的 NBTSB 和 BTFOSEL 位以及启动引脚的影响，详细说明见[表 1-3. 引导模式](#)。

在复位后，BOOT0 和 BOOT1 引脚上的值在 CK_SYS 的第 4 个上升沿被锁存。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

安全保护	OTP3		启动模式选择引脚		BOOT_MOD E[2:0]	引导源选择
	NBTSB	BTFOSEL	BOOT0	BOOT1		
无保护/保护等级低	0	x	1	1	011	片上 SRAM
无保护/保护等级低	0	x	1	0	001	引导装载程序
无保护/保护等级低	0	0	0	x	000	主 Flash 存储器
无保护/保护等级低	0	1	0	x	101	OTP1
x	1	0	x	x	000	主 Flash 存储器
x	1	1	x	x	101	OTP1
保护等级高	x	0	x	x	000	主 Flash 存储器
保护等级高	x	1	x	x	101	OTP1

上电序列或系统复位后，Arm® Cortex®-M33 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地址，然后从引导代码的基地址开始执行程序。所选引导源对应的存储空间会被映射到引导存储空间，即从 0x0000 0000 开始的地址空间。如果片上 SRAM（开始于 0x2000 0000 的存储空间）被选为引导源，用户必须在应用程序初始化代码中通过修改 NVIC 异常向量表和偏移地址将向量表重置到 SRAM 中。

当主 FLASH 存储器被选择作为引导源，从 0x0800 0000 开始的存储空间会被映射到引导存储空间。当选择 OTP1 作为引导源时，从地址 0x1FF0 0000 开始的内存空间会被映射到引导存储空间。

嵌入式启动加载程序位于系统内存中，用于重新编程闪存。GD32F50x MCU 嵌入式启动加载程序支持多种接口来更新闪存。在 GD32F50x 系列产品上可以使 USART 端口，以及标准 USB 端口。

1.5. 系统配置寄存器(SYSCFG)

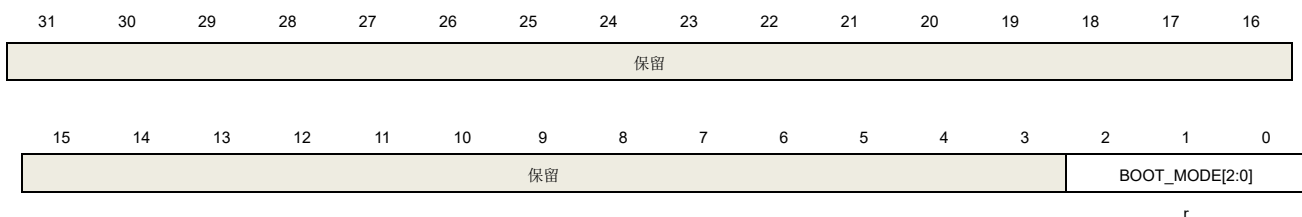
SYSCFG 基地址: 0x4001 4000

1.5.1. 配置寄存器 0 (SYSCFG_CFG0)

地址偏移: 0x00

复位值: 0x0000 000X (根据 BOOT0 和 BOOT1 引脚的状态, X 表示 BOOT_MODE[2:0], 可能为任意值)

该寄存器只能按字 (32 位) 访问。



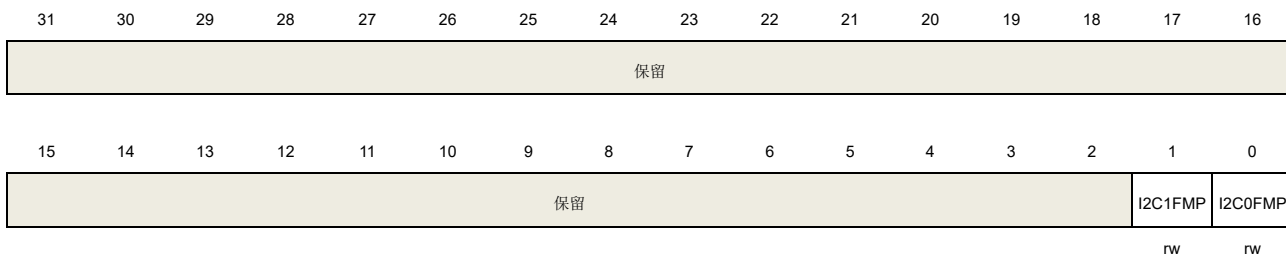
位/位域	名称	描述
31:3	保留	必须保持复位值。
2:0	BOOT_MODE [2:0]	引导模式(详细请参考 表1-3. 引导模式)。

1.5.2. 配置寄存器 1(SYSCFG_CFG1)

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
31:2	保留	必须保持复位值。
1	I2C1FMP	在 I2C1 上使能 Fast mode+ 0: 失能

		1: 使能
0	I2C0FMP	在 I2C0 上使能 Fast mode+
		0: 失能
		1: 使能

1.5.3. 锁定控制寄存器 (SYSCFG_LKCTL)

地址偏移: 0x018

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	SRAM_LOCK	SRAM ECC 双差错锁定位 该位由软件置位，由系统复位清零。 0: SRAM ECC 双差错信号从 TIMER0/7/15/16 中止输入端断开 1: SRAM ECC 双差错信号与 TIMER0/7/15/16 中止输入端连接
11:7	保留	必须保持复位值。
6	CPU_LOCK	CPU 锁定位 该位由软件置位，由系统复位清零。 0: CPU 锁定信号从 TIMER0/7/15/16 中止输入端断开 1: CPU 锁定信号与 TIMER0/7/15/16 中止输入端连接
5:3	保留	必须保持复位值。
2	LVD_LOCK	LVD 锁定位 该位由软件置位，由系统复位清零。 0: LVD 信号从 TIMER0/7/15/16 中止输入端断开 1: LVD 信号与 TIMER0/7/15/16 中止输入端连接
1:0	保留	必须保持复位值。

1.5.4. 总线超时寄存器(SYSCFG_BUSTO)

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												DMA1BU STO	DMA0BU STO	CPUSBU STO	CPUCBU STO
												rw	rw	rw	rw

位/位域	名称	描述
31:4	保留	必须保持复位值。
3	DMA1BUSTO	DMA1总线超时使能位 0: 禁用 1: 启用
2	DMA0BUSTO	DMA0总线超时使能位 0: 禁用 1: 启用
1	CPUSBUSTO	CPU Sbus 超时使能位 0: 禁用 1: 启用
0	CPUCBUSTO	CPU Cbus 超时使能位 0: 禁用 1: 启用

1.5.5. TIMER 输入选择寄存器 (SYSCFG_TIMERCISEL)

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											TIMER16_CIO_SEL[1:0]	保留	TIMER15_CIO_SEL[1 :0]		

r/w

r/w

位/位域	名称	描述
31:6	保留	必须保持复位值。
5:4	TIMER16_CIO_SEL[1:0]	TIMER16_CIO输入选择 该位选择TIMER输入源 00: TIMER16_CH0输入 01: 保留 10: 保留 11: CKOUT
3:2	保留	必须保持复位值。
1:0	TIMER15_CIO_SEL[1:0]	TIMER15_CIO输入选择 该位选择TIMER输入源 00: TIMER15_CH0输入 01: IRC40K 10: LXTAL 11: 保留

1.5.6. FPU 中断使能寄存器(SYSCFG_FPUINTEN)

地址偏移: 0x24

复位值: 0x0000 001F

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										IXIE	IAIE	OVFIE	UFIE	DZIE	IOPIE
										r/w	r/w	r/w	r/w	r/w	r/w

位/位域	名称	描述
31:6	保留	必须保持复位值。
5	IXIE	不精确中断使能位 0: 不精确中断禁用 1: 不精确中断启用
4	IAIE	输入异常中断使能位 0: 输入异常中断禁用

		1: 输入异常中断启用
3	OVFIE	溢出中断使能位 0: 溢出中断禁用 1: 溢出中断启用
2	UFIE	下溢中断使能位 0: 下溢中断禁用 1: 下溢中断启用
1	DZIE	DZIE 除以0中断使能位 0: 除以0中断禁用 1: 除以0中断启用
0	IOPIE	IOPIE 无效操作中断使能位 0: 无效操作中断禁用 1: 无效操作中断启用

1.5.7. SRAM 写保护寄存器(SYSCFG_SRAMWP)

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
SRAM_P31WP	SRAM_P30WP	SRAM_P29WP	SRAM_P28WP	SRAM_P27WP	SRAM_P26WP	SRAM_P25WP	SRAM_P24WP	SRAM_P23WP	SRAM_P22WP	SRAM_P21WP	SRAM_P20WP	SRAM_P19WP	SRAM_P18WP	SRAM_P17WP	SRAM_P16WP
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRAM_P15WP	SRAM_P14WP	SRAM_P13WP	SRAM_P12WP	SRAM_P11WP	SRAM_P10WP	SRAM_P9WP	SRAM_P8WP	SRAM_P7WP	SRAM_P6WP	SRAM_P5WP	SRAM_P4WP	SRAM_P3WP	SRAM_P2WP	SRAM_P1WP	SRAM_P0WP
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

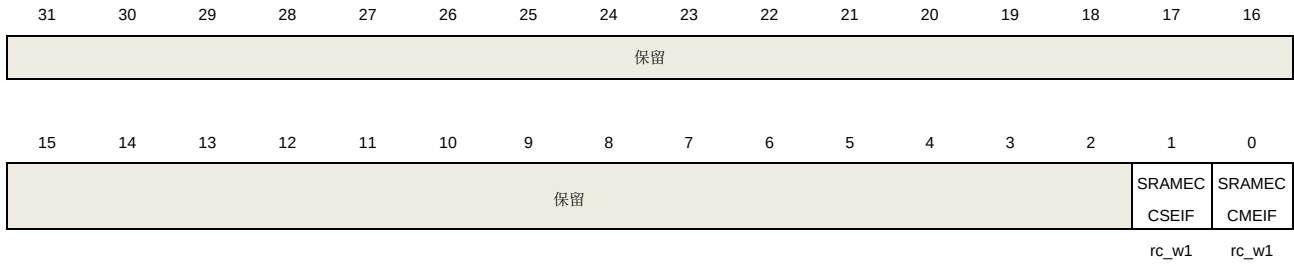
位/位域	名称	描述
31:0	SRAM_PxWP (x=0 到 31)	SRAM页x写保护 这些位由软件设置，仅通过系统复位清除。 0: SRAM页x的写保护已禁用 1: SRAM页x的写保护已启用

1.5.8. SRAM ECC 状态寄存器 (SYSCFG_SRAM_ECC_STAT)

地址偏移: 0x2C

复位值: 0x0000 0000

该寄存器只能按字（32 位）访问。



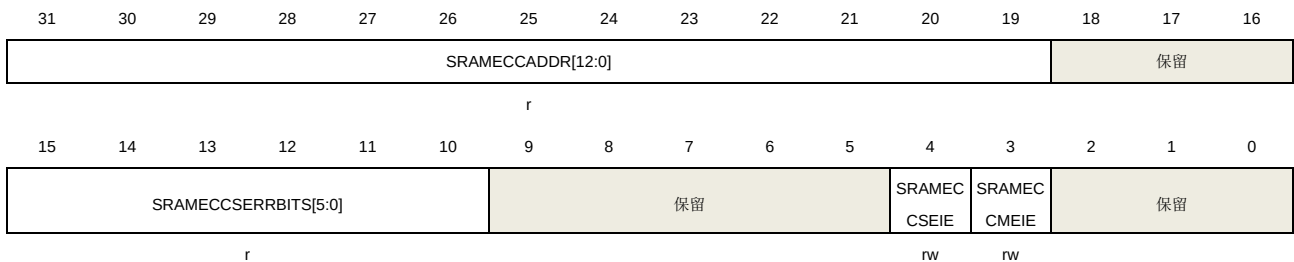
位/位域	名称	描述
31:2	保留	必须保持复位值。
1	SRAMECCSEIF	SRAM 单比特可纠正错误事件中断标志 软件写 1 清零。 0: 没有检测到 SRAM 单比特可纠正错误事件。 1: 检测到 SRAM 单比特可纠正错误事件。
0	SRAMECCMEIF	SRAM 两比特不可纠正错误事件中断标志 软件写 1 清零。 0: 没有检测到 SRAM 两比特不可纠正错误事件。 1: 检测到 SRAM 两比特不可纠正错误事件。

1.5.9. SRAM ECC 控制状态寄存器(SYSCFG_SRAM ECCCS)

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
31:19	SRAMECCADDR[12:0]	记录上一次发生 ECC 事件的 SRAM 故障系统地址
18:16	保留	必须保持复位值。
15:10	SRAMECCSERRBIT S[5:0]	指示错误位 哪个位有 SRAM ECC 单比特可校正错误

		000000: 没有错误
		000001: 位 0
		...
		111001: 位 38
9:5	保留	必须保持复位值。
4	SRAMECCSEIE	SRAM 单比特校正中断使能 0: 禁用 1: 启用
3	SRAMECCMEIE	SRAM 多比特非校正中断使能 0: 禁用 1: 启用
2:0	保留	必须保持复位值。

1.5.10. 总线超时状态寄存器(SYSCFG_BUSTOSTAT)

地址偏移: 0x34

复位值: 0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:4	保留	必须保持复位值。
3	DMA1BUSTOF	DMA1 总线超时标志 0: DMA1 总线未超时 1: DMA1 总线超时
2	DMA0BUSTOF	DMA0 总线超时标志 0: DMA0 总线未超时 1: DMA0 总线超时
1	CPUSBUSTOF	CPU Sbus 超时标志 0: CPU Sbus 未超时 1: CPU Sbus 超时

0	CPUCBUSTOF	CPU Cbus 超时标志
		0: CPU Cbus 未超时
		1: CPU Cbus 超时

1.5.11. TIMERx 配置寄存器 0 (SYSCFG_TIMERxCFG0, x=0, 7)

地址偏移: 0x100 for TIMER0

地址偏移: 0x13C for TIMER7

复位值: 0x0000 0000

TSCFG0[4:0]..TSCFG6[4:0]之间相互互斥, 不能同时配置。

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	TSCFG5[4:0]					TSCFG4[4:0]					TSCFG3[4:0]				
	rw					rw					rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TSCFG2[4:0]					TSCFG1[4:0]					TSCFG0[4:0]				
	rw					rw					rw				

位/位域	名称	描述
31	保留	必须保持复位值。
30:26	TSCFG5[4:0]	事件模式配置 计数器在触发输入的上升沿启动。 00000: 事件模式禁能 00001: 内部触发输入 0 (ITI0) 00010: 内部触发输入 1 (ITI1) 00011: 内部触发输入 2 (ITI2) 00100: 内部触发输入 3 (ITI3) 00101: CI0 的边沿标志位 (CI0F_ED) 00110: 滤波后的通道 0 输入 (CI0FE0) 00111: 滤波后的通道 1 输入 (CI1FE1) 01000: 滤波后的外部触发输入 (ETIFP) 01001: 滤波后的通道 2 输入 (CI2FE2) 01010: 滤波后的通道 3 输入 (CI3FE3) 其他值: 保留
25:21	TSCFG4[4:0]	暂停模式配置 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。 00000: 暂停模式禁能 00001: 内部触发输入 0 (ITI0)

		00010: 内部触发输入 1 (ITI1) 00011: 内部触发输入 2 (ITI2) 00100: 内部触发输入 3 (ITI3) 00101: 保留 00110: 滤波后的通道 0 输入 (CI0FE0) 00111: 滤波后的通道 1 输入 (CI1FE1) 01000: 滤波后的外部触发输入 (ETIFP) 01001: 滤波后的通道 2 输入 (CI2FE2) 01010: 滤波后的通道 3 输入 (CI3FE3) 其他: 保留
20:16	TSCFG3[4:0]	复位模式配置 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。 00000: 复位模式禁能 00001: 内部触发输入 0 (ITI0) 00010: 内部触发输入 1 (ITI1) 00011: 内部触发输入 2 (ITI2) 00100: 内部触发输入 3 (ITI3) 00101: CI0 的边沿标志位 (CI0F_ED) 00110: 滤波后的通道 0 输入 (CI0FE0) 00111: 滤波后的通道 1 输入 (CI1FE1) 01000: 滤波后的外部触发输入 (ETIFP) 01001: 滤波后的通道 2 输入 (CI2FE2) 01010: 滤波后的通道 3 输入 (CI3FE3) 其他: 保留
15	保留	必须保持复位值。
14:10	TSCFG2[4:0]	正交译码器模式 2 配置 00000: 编码器模式 2 禁能 其他: 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。
9:5	TSCFG1[4:0]	正交译码器模式 1 配置 00000: 编码器模式 1 禁能 其他: 根据 CI0FE0 的电平, 计数器在 CI1FE1 的边沿向上/下计数。
4:0	TSCFG0[4:0]	正交译码器模式 0 配置 00000: 编码器模式 0 禁能 其他: 根据 CI1FE1 的电平, 计数器在 CI0FE0 的边沿向上/下计数。

1.5.12. TIMERx 配置寄存器 1 (SYSCFG_TIMERxCFG1, x=0, 7)

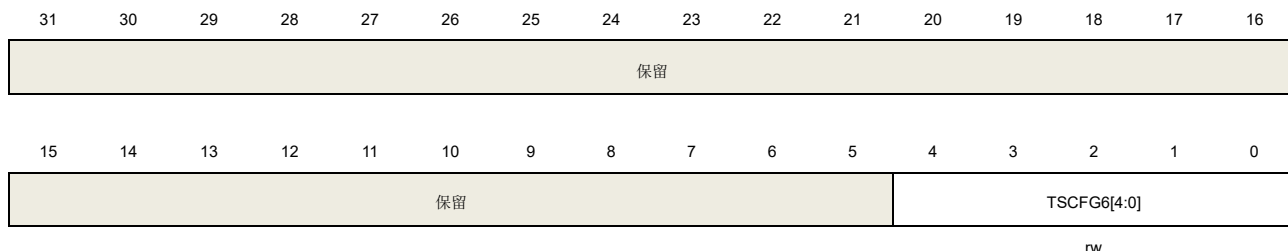
地址偏移: 0x104 for TIMER0

地址偏移: 0x140 for TIMER7

复位值: 0x0000 0000

TSCFG0[4:0]..TSCFG6[4:0]之间相互互斥, 不能同时配置。

该寄存器只能按字 (32位) 访问。



位/位域	名称	描述
31:5	保留	必须保持复位值。
4:0	TSCFG6[4:0]	外部时钟模式 0 配置 选中的触发输入的上升沿驱动计数器。 00000: 外部时钟模式 0 禁能 00001: 内部触发输入 0 (ITI0) 00010: 内部触发输入 1 (ITI1) 00011: 内部触发输入 2 (ITI2) 00100: 内部触发输入 3 (ITI3) 00101: CI0 的边沿标志位 (CI0F_ED) 00110: 滤波后的通道 0 输入 (CI0FE0) 00111: 滤波后的通道 1 输入 (CI1FE1) 01000: 滤波后的外部触发输入 (ETIFP) 01001: 滤波后的通道 2 输入 (CI2FE2) 01010: 滤波后的通道 3 输入 (CI3FE3) 其他: 保留

1.5.13. TIMERx 配置寄存器 2 (SYSCFG_TIMERxCFG2, x=0, 7)

地址偏移: 0x108 for TIMER0

地址偏移: 0x144 for TIMER7

复位值: 0x0000 0000

TSCFG0[4:0]..TSCFG6[4:0]之间相互互斥, 不能同时配置。

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											TSCFG15[4:0]				
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
31:21	保留	必须保持复位值。
20:16	TSCFG15[4:0]	内部触发输入源配置 00000: 保留 00001: 内部触发输入 0 (ITI0) 00010: 内部触发输入 1 (ITI1) 00011: 内部触发输入 2 (ITI2) 00100: 内部触发输入 3 (ITI3) 00101: CI0 的边沿标志位 (CI0F_ED) 其他: 保留 注意: 使用 TSCFG15[4:0]时, 需保证 TSCFGy[4:0] (y=0..6) 为零, 否则 ITS 触发源需与 TSCFGy[4:0]选择的通道输入源保持一致。
15:0	保留	必须保持复位值。

1.5.14. TIMERx 配置寄存器 0 (SYSCFG_TIMERxCFG0, x=1, 2, 3, 4)

地址偏移: 0x10C for TIMER1

地址偏移: 0x118 for TIMER2

地址偏移: 0x124 for TIMER3

地址偏移: 0x130 for TIMER4

复位值: 0x0000 0000

TSCFG0[4:0]..TSCFG6[4:0], TSCFG9[4:0]..TSCFG15[4:0]之间相互互斥, 不能同时配置。

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	TSCFG5[4:0]					TSCFG4[4:0]					TSCFG3[4:0]				
rw					rw					rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TSCFG2[4:0]					TSCFG1[4:0]					TSCFG0[4:0]				
rw					rw					rw					

位/位域	名称	描述

31	保留	必须保持复位值。
30:26	TSCFG5[4:0]	事件模式配置 计数器在触发输入的上升沿启动。 00000: 事件模式禁能 00001: 内部触发输入 0 (ITI0) 00010: 内部触发输入 1 (ITI1) 00011: 内部触发输入 2 (ITI2) 00100: 内部触发输入 3 (ITI3) 00101: CI0 的边沿标志位 (CI0F_ED) 00110: 滤波后的通道 0 输入 (CI0FE0) 00111: 滤波后的通道 1 输入 (CI1FE1) 01000: 滤波后的外部触发输入 (ETIFP) 01001: 滤波后的通道 2 输入 (CI2FE2) 01010: 滤波后的通道 3 输入 (CI3FE3) 其他: 保留
25:21	TSCFG4[4:0]	暂停模式配置 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。 00000: 暂停模式禁能 00001: 内部触发输入 0 (ITI0) 00010: 内部触发输入 1 (ITI1) 00011: 内部触发输入 2 (ITI2) 00100: 内部触发输入 3 (ITI3) 00101: 保留 00110: 滤波后的通道 0 输入 (CI0FE0) 00111: 滤波后的通道 1 输入 (CI1FE1) 01000: 滤波后的外部触发输入 (ETIFP) 01001: 滤波后的通道 2 输入 (CI2FE2) 01010: 滤波后的通道 3 输入 (CI3FE3) 其他: 保留
20:16	TSCFG3[4:0]	复位模式配置 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。 00000: 复位模式禁能 00001: 内部触发输入 0 (ITI0) 00010: 内部触发输入 1 (ITI1) 00011: 内部触发输入 2 (ITI2) 00100: 内部触发输入 3 (ITI3) 00101: CI0 的边沿标志位 (CI0F_ED) 00110: 滤波后的通道 0 输入 (CI0FE0) 00111: 滤波后的通道 1 输入 (CI1FE1)

		01000: 滤波后的外部触发输入 (ETIFP)
		01001: 滤波后的通道 2 输入 (CI2FE2)
		01010: 滤波后的通道 3 输入 (CI3FE3)
		其他: 保留
15	保留	必须保持复位值。
14:10	TSCFG2[4:0]	正交译码器模式 2 配置 00000: 编码器模式 2 禁能 其他: 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。
9:5	TSCFG1[4:0]	正交译码器模式 1 配置 00000: 编码器模式 1 禁能 其他: 根据 CI0FE0 的电平, 计数器在 CI1FE1 的边沿向上/下计数。
4:0	TSCFG0[4:0]	正交译码器模式 0 配置 00000: 编码器模式 0 禁能 其他: 根据 CI1FE1 的电平, 计数器在 CI0FE0 的边沿向上/下计数。

1.5.15. TIMERx 配置寄存器 1 (SYSCFG_TIMERxCFG1, x=1, 2, 3, 4)

地址偏移: 0x110 for TIMER1

地址偏移: 0x11C for TIMER2

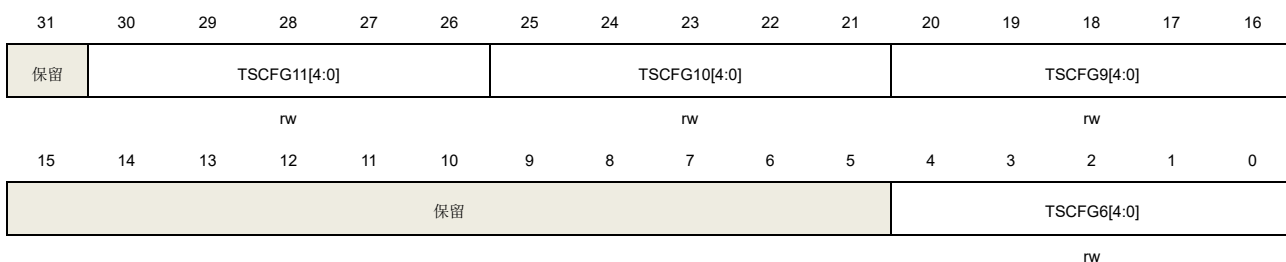
地址偏移: 0x128 for TIMER3

地址偏移: 0x134 for TIMER4

复位值: 0x0000 0000

TSCFG0[4:0]..TSCFG6[4:0], TSCFG9[4:0]..TSCFG15[4:0]之间相互互斥, 不能同时配置。

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。
30:26	TSCFG11[4:0]	译码器模式 2 配置 00000: 禁能 其他值: 计数器将在 CI0FE0 和 CI1FE1 信号的上升沿和下降沿计数, 计数方向由

		CH0P 位和 CH1P 位决定。
25:21	TSCFG10[4:0]	译码器模式 1 配置 00000: 禁能 其他值: CI0FE0 作为计数方向信号, CI1FE1 信号作为计数脉冲。计数器在 CI1FE1 的边沿开始计数, 该边沿由 CH1P 设置。
20:16	TSCFG9[4:0]	译码器模式 0 配置 00000: 禁能 其他值: CI0FE0 作为计数方向信号, CI1FE1 信号作为计数脉冲。计数器在 CI1FE1 的上升沿和下降沿开始计数。
15:5	保留	必须保持复位值。
4:0	TSCFG6[4:0]	外部时钟模式 0 配置 选中的触发输入的上升沿驱动计数器。 00000: 外部时钟模式 0 禁能 00001: 内部触发输入 0 (ITI0) 00010: 内部触发输入 1 (ITI1) 00011: 内部触发输入 2 (ITI2) 00100: 内部触发输入 3 (ITI3) 00101: CI0 的边沿标志位 (CI0F_ED) 00110: 滤波后的通道 0 输入 (CI0FE0) 00111: 滤波后的通道 1 输入 (CI1FE1) 01000: 滤波后的外部触发输入 (ETIFP) 01001: 滤波后的通道 2 输入 (CI2FE2) 01010: 滤波后的通道 3 输入 (CI3FE3) 其他: 保留

1.5.16. TIMERx 配置寄存器 2 (SYSCFG_TIMERxCFG2, x=1, 2, 3, 4)

地址偏移: 0x114 for TIMER1

地址偏移: 0x120 for TIMER2

地址偏移: 0x12C for TIMER3

地址偏移: 0x138 for TIMER4

复位值: 0x0000 0000

TSCFG0[4:0]..TSCFG6[4:0], TSCFG9[4:0]..TSCFG15[4:0]之间相互互斥, 不能同时配置。

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											TSCFG15[4:0]				

rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TSCFG14[4:0]					TSCFG13[4:0]					TSCFG12[4:0]				
	rw					rw					rw				

位/位域	名称	描述
31:21	保留	必须保持复位值。
20:16	TSCFG15[4:0]	内部触发输入源配置 00000: 保留 00001: 内部触发输入 0 (ITI0) 00010: 内部触发输入 1 (ITI1) 00011: 内部触发输入 2 (ITI2) 00100: 内部触发输入 3 (ITI3) 00101: CI0 的边沿标志位 (CI0F_ED) 其他: 保留 注意: 使用 TSCFG15[4:0]时, 需保证 TSCFGy[4:0] (y=0..6, 9..15) 为零, 否则 ITS 触发源需与 TSCFGy[4:0]选择的通道输入源保持一致。
15	保留	必须保持复位值。
14:10	TSCFG14[4:0]	正交译码器模式 4 配置 00000: 禁能 其他: 根据 CI0FE1 的电平, 计数器仅在 CI1FE1 的边沿开始计数。
9:5	TSCFG13[4:0]	正交译码器模式 3 配置 00000: 禁能 其他值: 根据 CI0FE0 的电平, 计数器仅在 CI1FE0 的边沿开始计数。
4:0	TSCFG12[4:0]	译码器模式3配置 00000: 禁用 其他: 计数器计数 CI0FE0 和 CI1FE1 信号的上升沿或下降沿。当 CHxP=0 时, 计数器将在高电平或下降沿或 C1xFEx 信号上开始计数; 当 CHxP=1 时, 计数器将在低电平或上升沿或 C1xFEx 信号上开始计数。

1.5.17. TIMERx 配置寄存器 0 (SYSCFG_TIMERxCFG0, x=15, 16)

地址偏移: 0x148 for TIMER15

地址偏移: 0x154 for TIMER16

复位值: 0x0000 0000

TSCFG3[4:0]..TSCFG6[4:0]之间相互互斥, 不能同时配置。

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	TSCFG5[4:0]					TSCFG4[4:0]					TSCFG3[4:0]				
	rw					rw					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	保留	必须保持复位值。
30:26	TSCFG5[4:0]	事件模式配置 计数器在触发输入的上升沿启动。 00000: 事件模式禁能 00001: 内部触发输入 0 (ITI0) 00010: 内部触发输入 1 (ITI1) 00011: 内部触发输入 2 (ITI2) 00100: 内部触发输入 3 (ITI3) 00101: CI0 的边沿标志位 (CI0F_ED) 00110: 滤波后的通道 0 输入 (CI0FE0) 00111: 滤波后的通道 1 输入 (CI1FE1) 01011: 滤波后的多模式通道 0 输入 (MCIOFEM0) 10011: 内部触发输入 14 (ITI14) 其他: 保留
25:21	TSCFG4[4:0]	暂停模式配置 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。 00000: 暂停模式禁能 00001: 内部触发输入 0 (ITI0) 00010: 内部触发输入 1 (ITI1) 00011: 内部触发输入 2 (ITI2) 00100: 内部触发输入 3 (ITI3) 00101: 保留 00110: 滤波后的通道 0 输入 (CI0FE0) 00111: 滤波后的通道 1 输入 (CI1FE1) 01000: 保留 01001: 保留 01010: 保留 01011: 滤波后的多模式通道 0 输入 (MCIOFEM0) 10011: 内部触发输入 14 (ITI14) 其他: 保留
20:16	TSCFG3[4:0]	复位模式配置 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。

		00000: 复位模式禁能
		00001: 内部触发输入 0 (ITI0)
		00010: 内部触发输入 1 (ITI1)
		00011: 内部触发输入 2 (ITI2)
		00100: 内部触发输入 3 (ITI3)
		00101: CI0 的边沿标志位 (CI0F_ED)
		00110: 滤波后的通道 0 输入 (CI0FE0)
		00111: 滤波后的通道 1 输入 (CI1FE1)
		01000: 保留
		01001: 保留
		01010: 保留
		01011: 滤波后的多模式通道 0 输入 (MCI0FEM0)
		10011: 内部触发输入 14 (ITI14)
		其他: 保留
15:0	保留	必须保持复位值。

1.5.18. TIMERx 配置寄存器 1 (SYSCFG_TIMERxCFG1, x=15, 16)

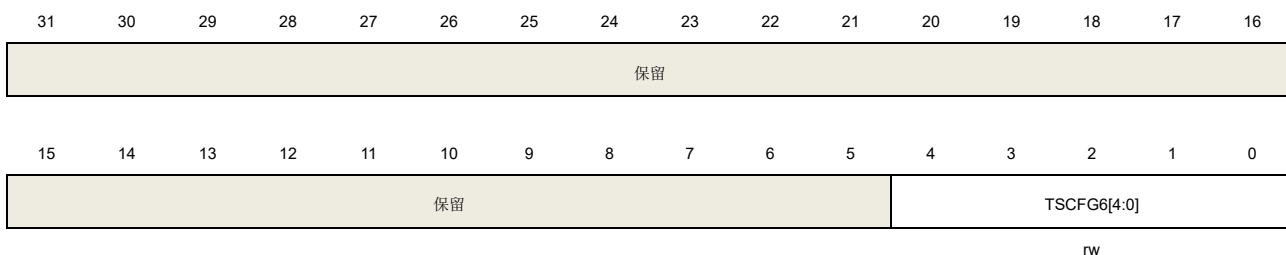
地址偏移: 0x14C for TIMER15

地址偏移: 0x158 for TIMER16

复位值: 0x0000 0000

TSCFG3[4:0]..TSCFG6[4:0]之间相互互斥, 不能同时配置。

该寄存器只能按字 (32位) 访问。



位/位域	名称	描述
31:5	保留	必须保持复位值。
4:0	TSCFG6[4:0]	外部时钟模式 0 配置 选中的触发输入的上升沿驱动计数器。 00000: 外部时钟模式 0 禁能 00001: 内部触发输入 0 (ITI0) 00010: 内部触发输入 1 (ITI1) 00011: 内部触发输入 2 (ITI2)

00100: 内部触发输入 3 (ITI3)
 00101: CIO 的边沿标志位 (CIOF_ED)
 00110: 滤波后的通道 0 输入 (CIOFE0)
 00111: 滤波后的通道 1 输入 (CI1FE1)
 01000: 保留
 01001: 保留
 01010: 保留
 01011: 滤波后的多模式通道 0 输入 (MCIOFEM0)
 10011: 内部触发输入 14 (ITI14)
 其他: 保留

1.5.19. TIMERx 配置寄存器 2 (SYSCFG_TIMERxCFG2, x=15, 16)

地址偏移: 0x150 for TIMER15

地址偏移: 0x15C for TIMER16

复位值: 0x0000 0000

TSCFG3[4:0]..TSCFG6[4:0]之间相互互斥, 不能同时配置。

该寄存器只能按字 (32位) 访问。



位/位域	名称	描述
31:21	保留	必须保持复位值。
20:16	TSCFG15[4:0]	内部触发输入源配置 00000: 保留 00001: 内部触发输入 0 (ITI0) 00010: 内部触发输入 1 (ITI1) 00011: 内部触发输入 2 (ITI2) 00100: 内部触发输入 3 (ITI3) 00101: CIO 的边沿标志位 (CIOF_ED) 00110: 保留 00111: 保留 01000: 保留 01001: 保留 01010: 保留

		01011: 保留
		10011: 内部触发输入 14 (ITI14)
		其他: 保留
		注意: 使用 TSCFG15[4:0]时, 需保证 TSCFGy[4:0] (y=3..6) 为零, 否则 ITS 触发源需与 TSCFGy[4:0]选择的通道输入源保持一致。
15:0	保留	必须保持复位值。

1.6. 设备电子签名

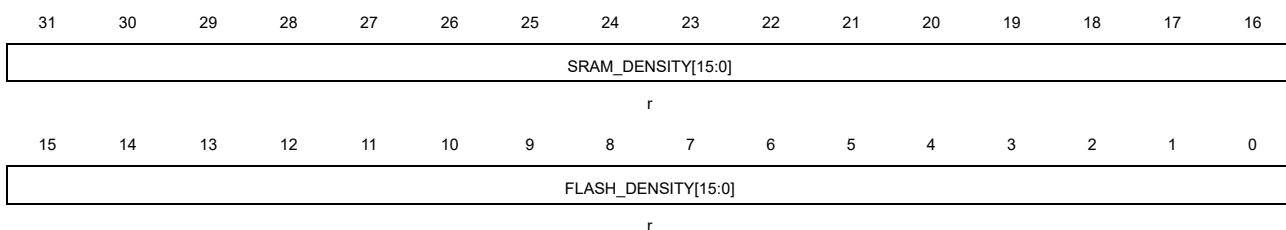
设备的电子签名中包含的存储容量信息和 96 位的唯一设备 ID。96 位唯一设备 ID 对于每颗芯片而言都是唯一的。它可以用作序列号, 或安全密钥的一部分等等。

1.6.1. 存储容量信息

基地址: 0x1FFF F7E0

该值是原厂设定的, 不能由用户修改。

该寄存器只能按字 (32 位) 访问



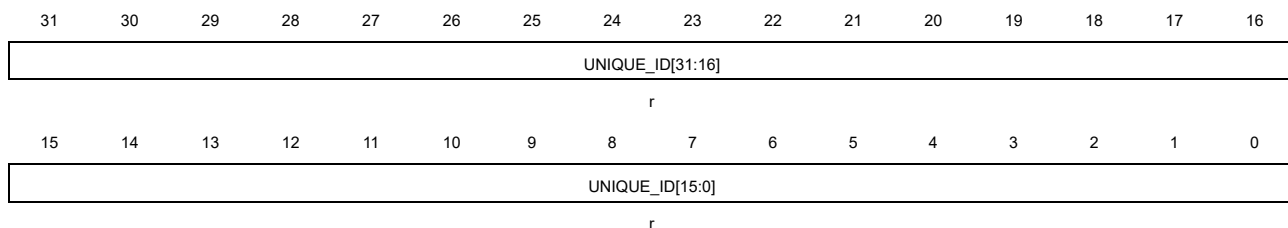
位/位域	名称	描述
31:16	SRAM_DENSITY [15:0]	SRAM存储器容量 该值表明芯片的片上SRAM存储器容量, 以Kbytes为单位 例如: 0x0008表示8Kbytes。
15:0	FLASH_DENSITY [15:0]	Flash存储器容量 该值表明芯片的片上FLASH容量, 以Kbytes为单位 例如: 0x0020表示32Kbytes。

1.6.2. 设备唯一 ID(96 位/位域)

基地址: 0x1FFF F7E8

该值是原厂设定的, 不能由用户修改。

该寄存器只能按字（32 位）访问

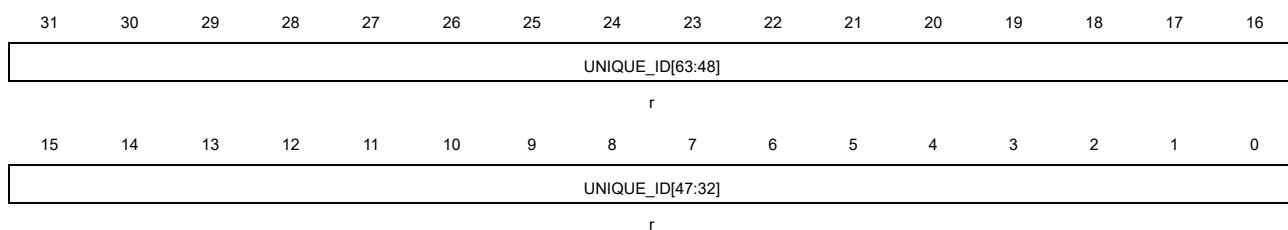


位/位域	名称	描述
31:0	UNIQUE_ID[31:0]	设备唯一 ID

基地址：0x1FFF F7EB

该值是原厂设定的，不能由用户修改。

该寄存器只能按字（32 位）访问

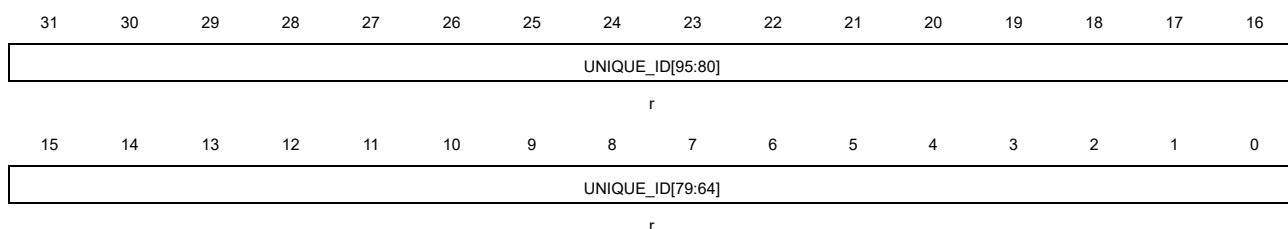


位/位域	名称	描述
31:0	UNIQUE_ID[63:32]	设备唯一 ID

基地址：0x1FFF F7F0

该值是原厂设定的，不能由用户修改。

该寄存器只能按字（32 位）访问



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

2. 中断/事件控制器（EXTI）

2.1. 简介

Cortex®-M33 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®-M33 技术参考手册》。

EXTI（中断/事件控制器）包括 20 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。

2.2. 主要特征

- Cortex®-M33 系统异常；
- 多达 79 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 20 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。

2.3. 功能说明

Arm Cortex®-M33 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状态压栈并行进行的，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状态所带来的开销。[表 2-1. Cortex®-M33 中的 NVIC 异常类型](#)和[表 2-2. 中断向量表](#)列出了 Cortex®-M33 中的 NVIC 异常类型。

表 2-1. Cortex®-M33 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

异常类型	向量编号	优先级 (a)	向量地址	描述
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
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

表 2-2. 中断向量表

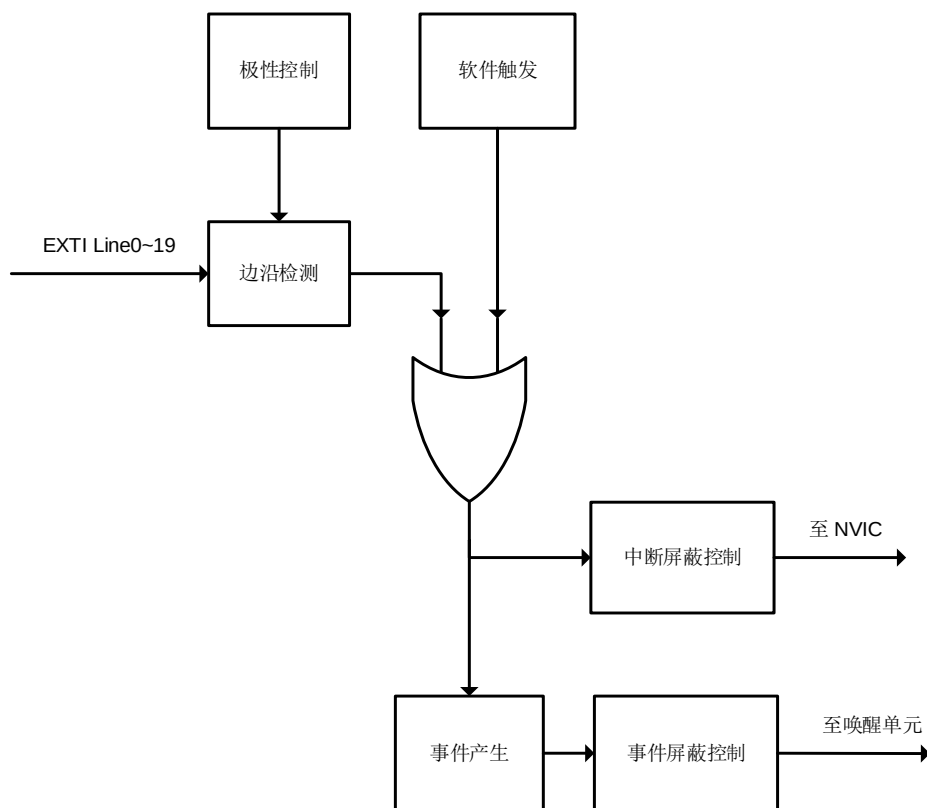
中断编号	向量编号	外设中断描述	向量地址
IRQ 0	16	窗口看门狗中断	0x0000_0040
IRQ 1	17	连接到 EXTI 线的 LVD / VAVD 中断	0x0000_0044
IRQ 2	18	侵入检测中断	0x0000_0048
IRQ 3	19	RTC 全局中断	0x0000_004C
IRQ 4	20	FMC 全局中断	0x0000_0050
IRQ 5	21	RCU 和 CTC 中断	0x0000_0054
IRQ 6	22	EXTI 线 0 中断	0x0000_0058
IRQ 7	23	EXTI 线 1 中断	0x0000_005C
IRQ 8	24	EXTI 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

中断编号	向量编号	外设中断描述	向量地址
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
IRQ 24	40	TIMER0 中止中断	0x0000_00A0
IRQ 25	41	TIMER0 更新中断	0x0000_00A4
IRQ 26	42	TIMER0 触发与通道换相中断	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	0x0000_00EC
IRQ 44	60	TIMER7 更新中断	0x0000_00F0
IRQ 45	61	TIMER7 触发与通道换相中断	0x0000_00F4
IRQ 46	62	TIMER7 通道捕获比较中断	0x0000_00F8
IRQ 47	63	ADC2 全局中断	0x0000_00FC
IRQ 48	64	保留	0x0000_0100
IRQ 49	65	RCU 时钟频率监控中断	0x0000_0104
IRQ50	66	连接 EXTI 线的 CMP 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

中断编号	向量编号	外设中断描述	向量地址
IRQ56	72	DMA1 通道 0 全局中断	0x0000_0120
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	DAC 全局中断	0x0000_0134
IRQ62	78	VUVD 或 VOVD 中断	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	CAN1EWMC 中断	0x0000_0148
IRQ67	83	SRAM ECC 中断	0x0000_014C
IRQ68	84	FPU 中断	0x0000_0150
IRQ69	85	CMP 中断	0x0000_0154
IRQ70	86	DMAMUX 中断	0x0000_0158
IRQ71	87	CAU 中断	0x0000_015C
IRQ72	88	HAU 中断	0x0000_0160
IRQ73	89	TRNG 中断	0x0000_0164
IRQ74	90	USBFS 中断	0x0000_0168
IRQ75	91	TIMER4 中断	0x0000_016C
IRQ76	92	TIMER15 中断	0x0000_0170
IRQ77	93	TIMER16 中断	0x0000_0174
IRQ78	94	TIMER0 通道中止中断	0x0000_0178
IRQ79	95	TIMER7 通道中止中断	0x0000_017C

2.4. 外部中断及事件（EXTI）框图

图 2-1. EXTI 框图



2.5. 外部中断及事件功能概述

EXTI 包含多达 20 个相互独立的边沿检测电路并且可以向处理器产生中断请求或事件唤醒。EXTI 提供 3 种触发类型：上升沿触发，下降沿触发和任意沿触发。EXTI 中每个边沿检测电路都可以分别予以配置或屏蔽。

EXTI 触发源包括来自 I/O 管脚的 16 根线以及来自内部模块的 4 根线具体细节参考[表 2-3. EXTI 触发源](#)。通过配置 GPIO 模块的 AFIO_EXTISSx 寄存器，所有的 GPIO 管脚都可以被选作 EXTI 的触发源，具体细节请参考[通用和备用输入输出接口（GPIO 和 AFIO）](#)章节。

除了中断，EXTI 还可以向处理器提供事件信号。The Cortex®-M33 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，例如一个特定的 I/O 管脚电平翻转或者 RTC 闹钟动作，EXTI 能唤醒处理器及整个系统。

硬件触发

硬件触发被用来检测外部或内部信号的电压变化。软件需要按如下步骤配置来使用这项功能：

1. 根据应用需要配置 AFIO 模块中的 EXTI 触发源；
2. 配置 EXTI_RTEN 寄存器和 EXTI_FTEN 寄存器以使能相应引脚的上升沿或下降沿检测（软件应当同时配置引脚对应的 RTENx 和 FTENx 位以检测该引脚上升沿和下降沿的变化）；
3. 通过配置引脚对应的 EXTI_INTEN 或 EXTI_EVENTEN 位，使能中断或事件；
4. EXTI 开始检测被配置的引脚上的电平变化，当这些引脚上期望的变化被检测到时，使能的中断或事件将被触发。如果为中断触发，则对应的 PD 位将立刻被置 1；如果为事件触发，则对应的 PD 位不被置 1。软件需要响应该中断或事件并清除相应 PDx 位。

软件触发

按照如下步骤软件也可以触发 EXTI 中断或事件：

1. 配置对应的 EXTI_INTEN 或 EXTI_EVENTEN 位使能中断或事件；
2. 配置 EXTI_SWIEV 寄存器的对应 SWIEVx 位，使能的中断或事件将被立即触发。如果为中断触发，则对应的 PD 位将立刻被置 1；如果为事件触发，则对应的 PD 位不被置 1。软件需要响应该中断或事件并清除相应 PDx 位。

表 2-3. 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 / VAVD
17	RTC 闹钟
18	CMP 唤醒

EXTI 线编号	触发源
19	USBFS 唤醒

2.6. EXTI 寄存器

EXTI 基地址: 0x4001 0400

2.6.1. 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												INTEN19	INTEN18	INTEN17	INTEN16
												rw	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
31:20	保留	必须保持复位值
19:0	INTENx	中断使能位 0: 第x线中断被禁用 1: 第x线中断被使能

2.6.2. 事件使能寄存器 (EXTI_EVEN)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留												EVEN19	EVEN18	EVEN17	EVEN16
												rw	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
31:20	保留	必须保持复位值。
19:0	EVENx	事件使能位x (x = 0...19) 0: 第x线事件被禁用。

1: 第x线事件被使能

2.6.3. 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												RTEN19	RTEN18	RTEN17	RTEN16
												r/w	r/w	r/w	r/w
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
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

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

2.6.4. 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												FTEN19	FTEN18	FTEN17	FTEN16
												r/w	r/w	r/w	r/w
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
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

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

2.6.5. 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												SWIEV19	SWIEV18	SWIEV17	SWIEV16
												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
31:20	保留	必须保持复位值。
19:0	SWIEVx	中断/事件软件触发x (x = 0...19) 0: 禁用EXTI线x软件中断/事件请求 1: 激活EXTI线x软件中断/事件请求

2.6.6. 挂起寄存器 (EXTI_PD)

地址偏移: 0x14

复位值: 0xFFFF XXXX, X 表示未定义

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												PD19	PD18	PD17	PD16
												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
31:20	保留	必须保持复位值。
19:0	PDx	中断挂起状态x (x = 0...19) 0: EXTI线x没有被触发 1: EXTI线x被触发 对这些位写1, 可将其清0。

3. 直接存储器访问控制器（DMA）

3.1. 简介

DMA 控制器提供了一种硬件的方式在外设和存储器之间或者存储器和存储器之间传输数据，而无需 CPU 的介入，从而使 CPU 可以专注在处理其他系统功能上。DMA 控制器有 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。

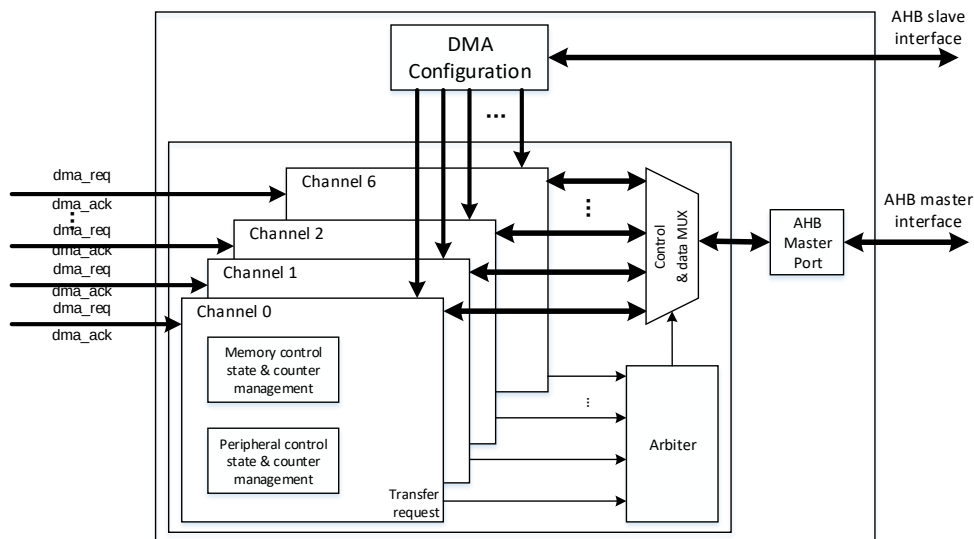
DMA 控制器和 Cortex®-M33 内核共享系统总线。当 DMA 和 CPU 访问同样的地址空间时，DMA 访问可能会阻挡 CPU 访问系统总线几个总线周期。总线矩阵中实现了循环仲裁算法来分配 DMA 与 CPU 的访问权，它可以确保 CPU 得到至少一半的系统总线带宽。

3.2. 主要特性

- 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。

3.3. 结构框图

图 3-1. DMA 结构框图



由图 3-1. DMA 结构框图所示，DMA 控制器由 4 部分组成：

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

3.4. 功能描述

3.4.1. 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 PWIDTH 和 MWIDTH 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 PWIDTH 和 MWIDTH 的各种配置，DMA 传输的操作详见表 3-1. DMA 传输操作。

表 3-1. DMA 传输操作

传输宽度		传输操作	
源	目标	源	目标
32 bits	32 bits	1: Read B3B2B1B0[31:0] @0x0 2: Read B7B6B5B4[31:0] @0x4	1: Write B3B2B1B0[31:0] @0x0 2: Write B7B6B5B4[31:0] @0x4

传输宽度		传输操作	
源	目标	源	目标
		3: Read BBBAB9B8[31:0] @0x8 4: Read BFBEBDBC[31:0] @0xC	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

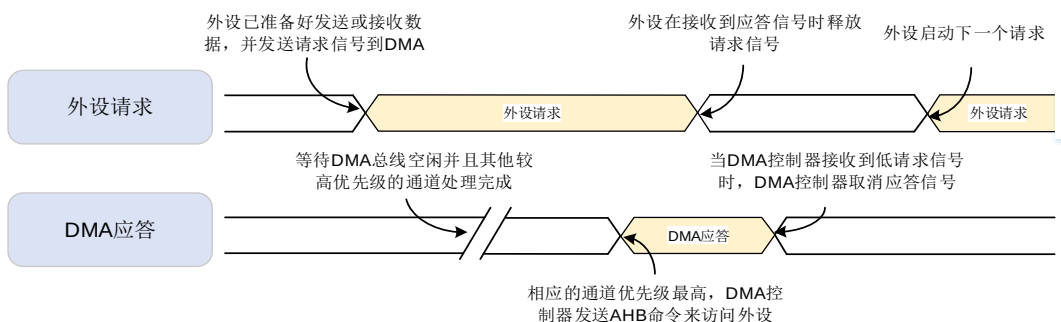
- 若 CHEN 位被清零时 DMA 传输还未完成，重新使能 CHEN 位将分两种情况：
 - 在重新使能DMA通道前，未对该通道的相关寄存器进行操作，则DMA将继续完成上次的传输。
 - 在重新使能DMA通道前，对任意相关寄存器进行了操作，则DMA将开始一次新的传输。
- 若清零 CHEN 位时，DMA 传输已经完成，之后未对任意寄存器进行操作前便使能 DMA 通道，则不会触发任何 DMA 传输。

3.4.2. 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：

- 图 3-2. 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 3-2. 握手机制



3.4.3. 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。

3.4.4. 地址生成

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

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

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

3.4.5. 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。

3.4.6. 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。

3.4.7. 通道配置

要启动一次新的 DMA 数据传输，建议遵循以下步骤进行操作：

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

3.4.8. 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。

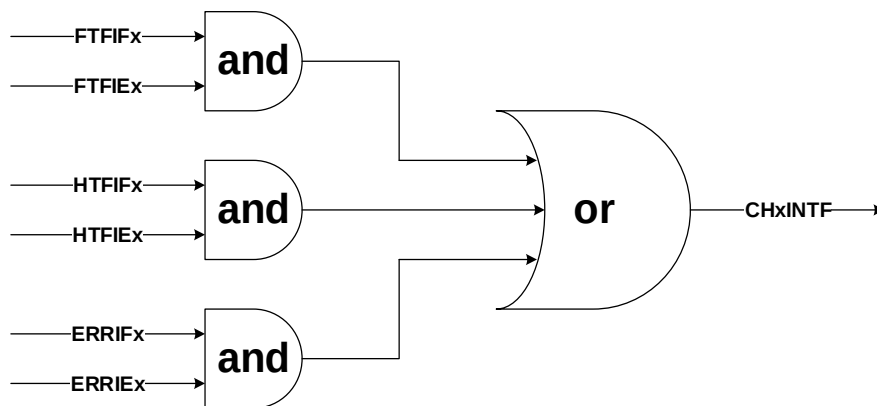
每一个中断事件在DMA_INTF寄存器中有专用的标志位，在DMA_INTC寄存器中有专用的清除位，在DMA_CHxCTL寄存器中有专用的使能位。[表3-2. 中断事件](#)描述了其对应关系。

表 3-2. 中断事件

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

DMA中断逻辑如[图3-3. 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 3-3. 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）

3.4.9. DMA 请求映射

每个 DMA 通道的请求都连接至由 DMAMUX 请求复用器的对应通道输出来转发的 AHB/APB 外设请求，参考 [DMA 请求多路复用器 \(DMAMUX\)](#)。

3.5. 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。

3.5.1. 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
				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIF3	HTFIF3	FTFIF3	GIF3	ERRIF2	HTFIF2	FTFIF2	GIF2	ERRIF1	HTFIF1	FTFIF1	GIF1	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/ 15/11/7/3	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 0: 通道x未发生传输错误 1: 通道x发生传输错误
26/22/18/ 14/10/6/2	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 0: 通道x半传输未完成 1: 通道x半传输完成
25/21/17/ 13/9/5/1	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 0: 通道x传输未完成 1: 通道x传输完成
24/20/16/ 12/8/4/0	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 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

3.5.2. 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/ 15/11/7/3	ERRIFCx	清除通道x(x=0...6)的错误标志位 0: 无影响 1: 清零DMA_INTF寄存器的ERRIFx位
26/22/18/ 14/10/6/2	HTFIFCx	清除通道x(x=0...6)的半传输完成标志位 0: 无影响 1: 清零DMA_INTF寄存器的HTFIFx位
25/21/17/ 13/9/5/1	FTFIFCx	清除通道x(x=0...6)的传输完成标志位 0: 无影响 1: 清零DMA_INTF寄存器的FTFIFx位
24/20/16/ 12/8/4/0	GIFCx	清除通道x(x=0...6)的全局中断标志位 0: 无影响 1: 清零DMA_INTF寄存器的GIFx, ERRIFx, HTFIFx和FTFIFx位

3.5.3. 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



位/位域	名称	描述
31:15	保留	必须保持复位值。
14	M2M	存储器到存储器模式 软件置位和清零 0: 禁止存储器到存储器模式 1: 使能存储器到存储器模式 CHEN位为1时, 该位不能被配置
13:12	PRI0[1:0]	软件优先级 软件置位和清零 00: 低 01: 中 10: 高 11: 极高 CHEN位为1时, 该位域不能被配置
11:10	MWIDTH[1:0]	存储器的传输数据宽度 软件置位和清零 00: 8-bit 01: 16-bit 10: 32-bit 11: 保留 CHEN位为1时, 该位域不能被配置
9:8	PWIDTH[1:0]	外设的传输数据宽度 软件置位和清零 00: 8-bit 01: 16-bit 10: 32-bit 11: 保留 CHEN位为1时, 该位域不能被配置
7	MNAGA	存储器的地址生成算法 软件置位和清零 0: 固定地址模式 1: 增量地址模式 CHEN位为1时, 该位不能被配置
6	PNAGA	外设的地址生成算法 软件置位和清零 0: 固定地址模式 1: 增量地址模式

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

3.5.4. 通道 x 计数寄存器 (DMA_CHxCNT)

$x = 0 \dots 6$, x 为通道序号

地址偏移: $0x0C + 0x14 \times x$

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

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]	传输计数 CHEN位为1时，该位域不能被配置 该寄存器表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载为初始设置值。

3.5.5. 通道 x 外设基地址寄存器 (DMA_CHxPADDR)

$x = 0 \dots 6$, x 为通道序号

地址偏移: $0x10 + 0x14 \times x$

复位值: 0x0000 0000

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

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

3.5.6. 通道 x 存储器基地址寄存器 (DMA_CHxMADDR)

$x = 0 \dots 6$, x 为通道序号

地址偏移: $0x14 + 0x14 \times x$

复位值: 0x0000 0000

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

MADDR[15:0]

rw

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

4. DMA 请求多路复用器 (DMAMUX)

4.1. 简介

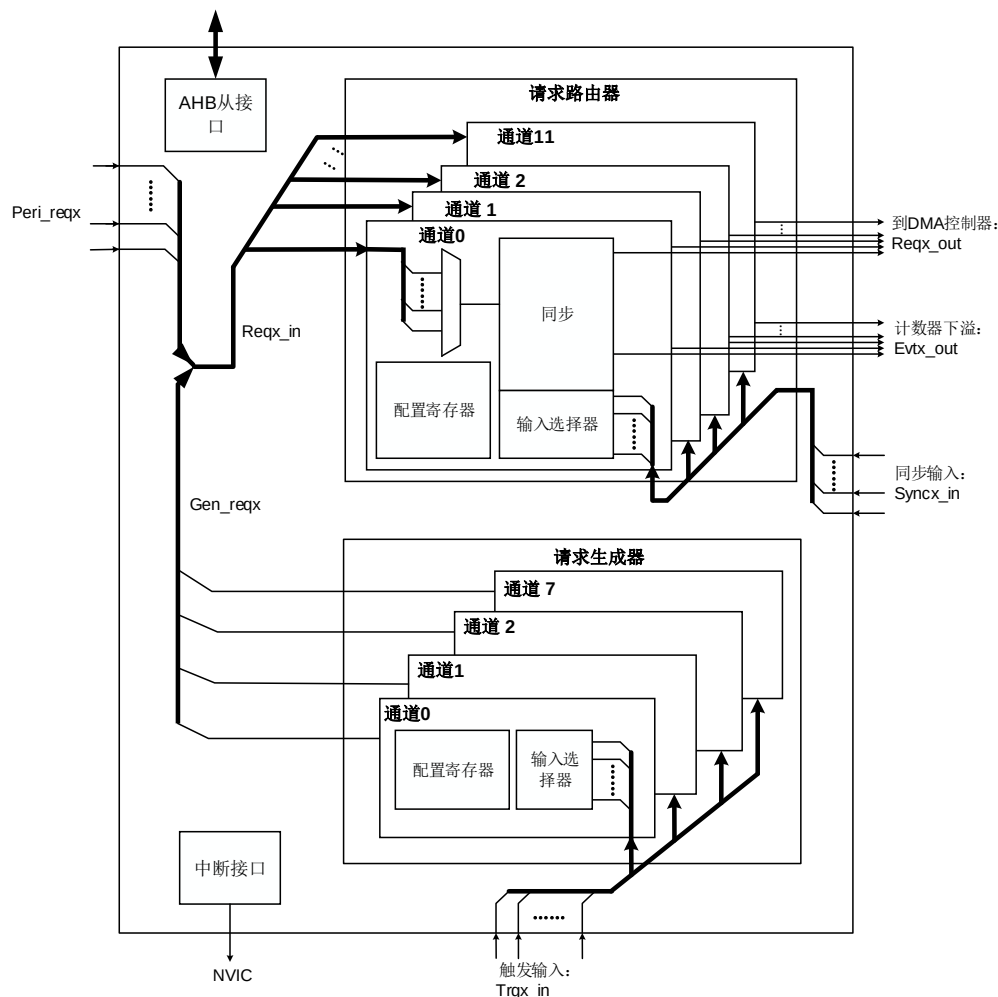
DMAMUX 是 DMA 请求的传输调度器。可编程的 DMA 请求多路复用器 DMAMUX，可在外设和 DMA 控制器之间路由 DMA 请求线路，或者 DMAMUX 也可以将可编程事件连入到输入触发信号上，作为一个 DMAMUX 请求发生器，再由 DMAMUX 请求路由器在 DMAMUX 请求生成器产生的 DMA 请求和 DMA 控制器之间路由 DMA 请求线路。每个 DMAMUX 请求路由通道选择一条唯一的 DMA 请求线路，无条件地或同步地从它的 DMAMUX 同步输入事件。DMA 请求信号会一直挂起，直到 DMA 控制器响应它，并且产生一个 DMA 确认信号，此时相应的 DMA 请求信号被释放。

4.2. 主要特征

- 12 个可配置的 DMAMUX 请求路由输出通道；
- 4 个 DMAMUX 请求生成通道；
- 20 路触发输入信号到 DMAMUX 请求生成器；
- 20 路同步输入信号；
- 每个 DMAMUX 请求生成通道包含一个 DMAMUX 请求触发输入选择器，一个 DMAMUX 请求生成计数器，和一个指示被选中的 DMAMUX 请求触发输入信号的事件溢出标志；
- 每个 DMAMUX 请求路由输出通道包含 189 路外设 DMAMUX 请求输入信号，一个同步输入信号选择器，一条 DMA 请求路由输出线路，一个路由事件输出信号用于 DMA 请求级联，一个 DMAMUX 请求路由计数器，和一个指示被选中的同步输入信号的事件溢出标志。

4.3. 结构框图

图 4-1. DMAMUX 结构框图



4.4. 功能说明

如 [图 4-1. DMAMUX 结构框图](#) 所示，DMAMUX 包含两个子模块：

■ DMAMUX 请求路由器

DMAMUX 请求路由器输入（Reqx_in）来自两部分：

- 一部分来自外设请求（Peri_reqx）；
- 另一部分来自 DMAMUX 请求生成器（Gen_reqx）。

DMAMUX 请求路由输出到 DMA 控制器对应的通道（Reqx_out）。

同步输入（Syncx_in）来自内部或外部信号。

■ DMAMUX 请求生成器

DMAMUX 请求触发输入 (Trgx_in) 来自内部或外部信号。

4.4.1. DMAMUX 信号

表 4-1. DMAMUX 信号

信号名称	描述
Reqx_in	DMAMUX 请求路由输入信号 (来自外设的请求或者 DMAMUX 请求生成器生成的请求)
Peri_reqx	从外设输入到 DMAMUX 的 DMA 请求线路
Gen_reqx	DMAMUX 请求生成器生成输出的 DMA 请求信号
Reqx_out	DMAMUX 请求输出信号 (到 DMA 控制器)
Trgx_in	DMAMUX 请求触发输入信号 (到 DMAMUX 请求生成器)
Syncx_in	DMAMUX 同步输入信号 (到 DMAMUX 请求路由器)
Evtx_out	DMAMUX 请求路由计数器下溢事件输出信号

4.4.2. DMAMUX 请求路由器

DMAMUX 请求路由器可在外设/DMAMUX 请求生成器,与 DMA 控制器之间路由 DMA 请求线路。DMAMUX 请求路由器由 DMAMUX 请求路由通道组成。DMA 请求输入信号并联至所有的 DMAMUX 请求路由通道。每个 DMAMUX 请求路由通道都有一个同步单元。同步输入信号并联至所有 DMAMUX 请求路由通道的同步单元。每个 DMAMUX 请求路由通道都有一个内部的 DMAMUX 请求路由计数器。

DMAMUX 请求路由通道

DMAMUX 请求路由通道 x 的请求路由输入由 DMAMUX_RM_CHxCFG 寄存器的 MUXID[6:0]位域来配置,请求路由输入可选为外设 DMA 请求,或者 DMAMUX 请求生成器产生的 DMA 请求,参考[表 4-3. DMAMUX 请求路由输入信号映射](#)。一个 DMAMUX 请求路由通道只能与一个 DMA 控制器通道相连接。

注意:当 MUXID[6:0]值为 0 时,没有 DMA 请求线路被映射到 DMAMUX 请求路由通道上。DMAMUX 不允许将同一个 DMA 请求线路 (相同 MUXID[6:0]且非空) 映射到两个不同的 DMAMUX 请求路由通道上。

当同步模式禁能时

每当连到 DMAMUX 的 DMA 请求被 DMA 控制器服务,这个 DMA 请求将取消挂起,内部的 DMAMUX 请求路由计数器将减 1。当 DMAMUX 请求路由计数器发生下溢时,DMAMUX_RM_CHxCFG 寄存器的 NBR[4:0]值将自动重装载到计数器中。如果将 EVGEN 位置位,使能通道事件输出,则通道事件输出前,DMA 请求数量为 NBR[4:0] + 1。

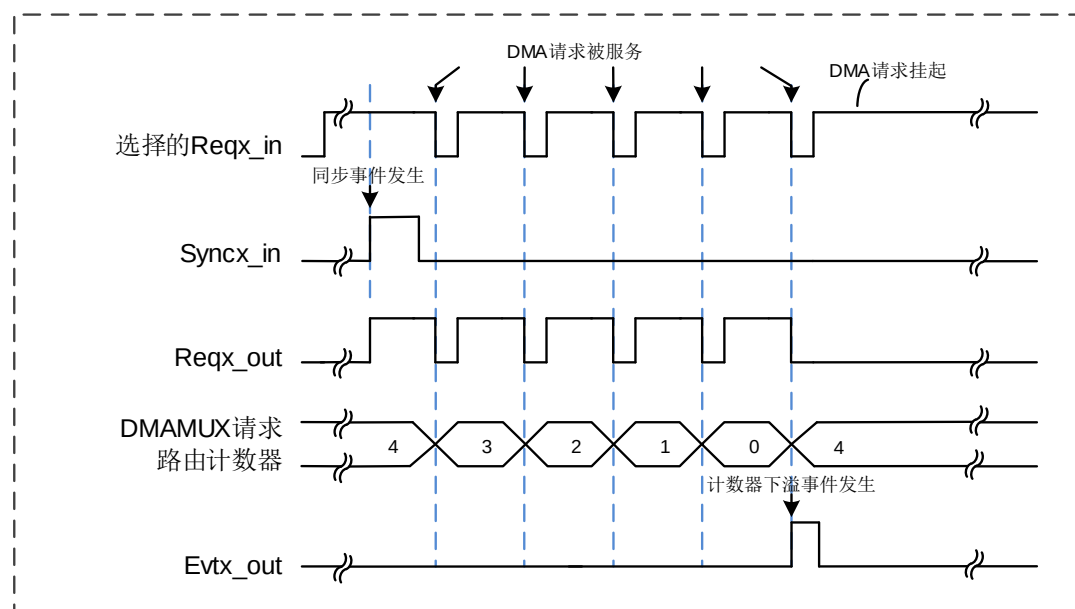
注意：只有当 DMAMUX 请求路由通道 x 的同步使能位 SYNCEN 位和通道事件输出使能位 EVGEN 位都为 0 时，才能配置其 NBR[4:0]位域。

当同步模式使能时

如果 DMAMUX 请求路由通道 x 工作在同步模式下，当检测到选择的同步输入信号的上升沿或者下降沿时，挂起的 DMA 请求将被连到 DMAMUX 请求路由通道 x 的输出。每当连到 DMAMUX 的 DMA 请求被 DMA 控制器服务，这个 DMA 请求将取消挂起，内部的 DMAMUX 请求路由计数器将减 1。当 DMAMUX 请求路由计数器发生下溢时，DMA 请求线路将断开与 DMAMUX 请求路由通道 x 的输出的连接，并且 DMAMUX_RM_CHxCFG 寄存器的 NBR[4:0]值将自动重载到计数器中。一个同步事件可传输 NBR[4:0] + 1 个 DMA 请求到 DMAMUX 请求路由通道 x 的输出上。

图 4-2. 同步模式为当 NBR[4:0]=4, SYNCEN=1, EVGEN=1, SYNCPC[1:0]=0b01 时的举例。

图 4-2. 同步模式



置位 DMAMUX_RM_CHxCFG 寄存器的 SYNCEN 位可使能 DMAMUX 请求路由通道 x 的同步模式。同步输入信号可由 DMAMUX_RM_CHxCFG 寄存器的 SYNCID[4:0]位域来配置，参考[表 4-5. 同步输入信号映射](#)。同步输入信号的有效边沿由 DMAMUX_RM_CHxCFG 寄存器的 SYNCPC[1:0]位域来配置。

注意：如果同步输入事件发生时，DMAMUX 输入上没有挂起的 DMA 请求，则这个同步输入事件将被忽略，之后如有 DMA 请求被挂起，它将不会被连接到 DMAMUX 请求路由通道 x 的输出，直到发生下一个同步输入事件。

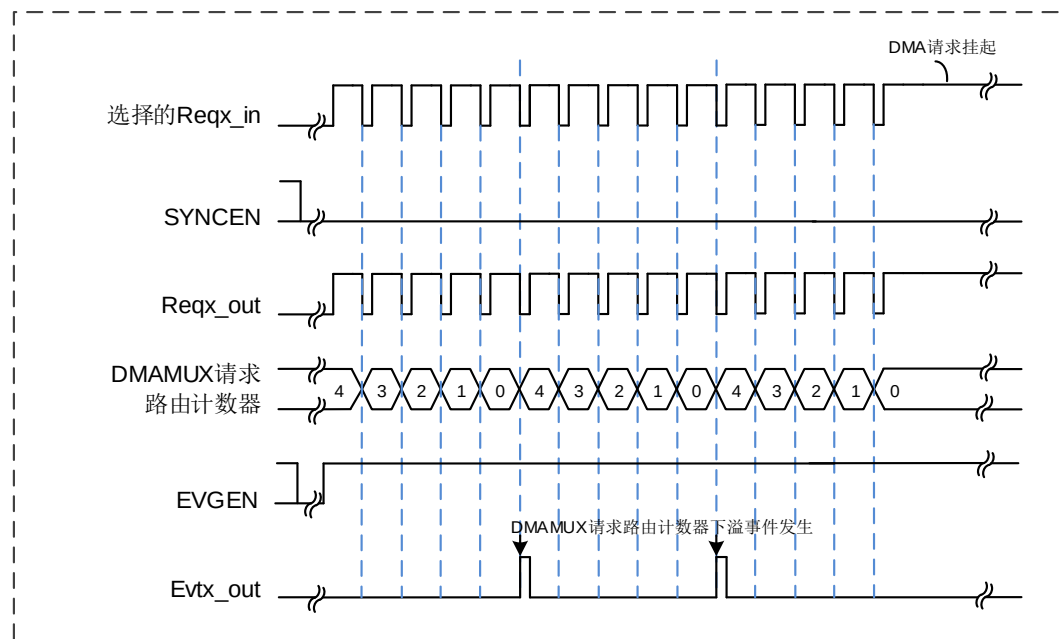
通道事件输出

每个 DMAMUX 请求路由通道都有一个通道事件输出信号 Evtx_out，用于 DMAMUX 请求路由计数器的下溢事件输出。Evt0_out ~ Evt3_out 信号可用于 DMA 请求级联。如果通过置位

DMAMUX_RM_CHxCFG 寄存器的 EVGEN 位来使能 DMAMUX 请求路由通道 x 的通道事件输出，当 DMAMUX 请求路由计数器自动重载为 NBR[4:0]值时，发生一个通道事件，输出为一个 AHB 时钟周期脉冲。

[图 4-3. 通道事件输出](#)为当 NBR[4:0]=4，SYNCEN=0，EVGEN=1 时的举例。

图 4-3. 通道事件输出



注意：如果 EVGEN = 1 且 NBR[4:0] = 0，则每次 DMA 请求被服务时都会输出一个通道事件。

同步溢出

如果在 DMAMUX 请求路由计数器下溢之前又发生了新的同步事件，则 DMAMUX_RM_INTF 寄存器的同步溢出标志位 SOIFx 位将置位。

注意：建议在 DMA 控制器对应通道请求被取消时，配置 DMAMUX_RM_CHxCFG 寄存器的 SYNCEN 位为 0 来禁用 DMAMUX 请求路由通道 x 的同步模式。否则，当又发生一个新的同步事件时，由于接收不到 DMA 的响应信号将会发生同步溢出事件。

4.4.3. DMAMUX 请求生成器

DMAMUX 请求生成器在触发输入事件发生时会产生 DMA 请求。DMAMUX 请求生成器由 DMAMUX 请求生成通道组成。DMA 请求触发输入信号并联至所有 DMAMUX 请求生成通道。每个 DMAMUX 请求生成通道都有一个内部的 DMAMUX 请求生成计数器。

触发输入信号的有效边沿由 DMAMUX_RG_CHxCFG 寄存器的 RGTP[1:0]位域来配置。DMAMUX 请求生成通道 x 的触发输入信号由 DMAMUX_RG_CHxCFG 寄存器的 TID[4:0]位域来配置，参考[表 4-4. 触发输入信号映射](#)。置位 DMAMUX_RG_CHxCFG 寄存器的 RGEN 位来使能 DMAMUX

请求生成通道 x 。

DMAMUX 请求生成通道

当发生触发输入事件时，对应的 DMAMUX 请求生成通道 x 开始产生 DMA 请求到通道的输出上，通道输出连到 DMAMUX 请求路由器的输入上。每当 DMAMUX 生成的 DMA 请求被 DMA 控制器服务，这个 DMA 请求将取消挂起，内部的 DMAMUX 请求生成计数器将减 1。当 DMAMUX 请求生成计数器发生下溢时，DMAMUX 请求生成通道将停止产生 DMA 请求，在下一个触发输入事件发生时，DMAMUX 请求生成计数器将自动重载为 DMAMUX_RG_CHxCFG 寄存器的 NBRG[4:0] 位域值。

注意：触发输入事件后产生的 DMA 请求数量为 NBRG[4:0] + 1。只有当 DMAMUX 请求生成通道 x 的 RGEN 位为 0 时才可以配置 NBRG[4:0] 位域。

触发溢出

如果 RGEN 位为 1，DMAMUX 请求生成通道 x 被使能，当一个新的触发输入信号发生了，而此时 DMAMUX 请求生成计数器还未发生下溢，则 DMAMUX_RG_INTFx 寄存器的 TOIFx 位将硬件置位以指示发生了触发溢出事件。

注意：建议在 DMA 控制器对应通道请求被取消时，配置 DMAMUX_RG_CHxCFG 寄存器的 RGEN 位为 0 来禁能 DMAMUX 请求生成通道 x 。否则，当又发生一个新的触发输入事件时，由于接收不到 DMA 的响应信号将会发生触发溢出事件。

4.4.4. 通道配置

根据以下步骤来配置 DMAMUX 的通道 y 和对应的 DMA 通道 x ：

1. 完整配置 DMA 通道 x 相关参数，除了 DMA 通道 x 的使能；
2. 完整配置 DMAMUX 通道 y 相关参数；
3. 设置 DMA_CHxCTL 寄存器的 CHEN 位 1 来使能 DMA 通道 x 。

4.4.5. 中断

DMAMUX 模块有两种类型的中断事件，包括 DMAMUX 请求路由通道的同步溢出事件，和 DMAMUX 请求生成通道的触发溢出事件。

每个中断事件都有一个专用的标志位，专用的清除位和专用的使能位。[表 4-2. 中断事件](#) 描述了其对应关系。

表 4-2. 中断事件

中断事件	标志位	清除位	使能位
DMAMUX 请求路由通道 x 上的同步溢出事件	DMAMUX_RM_INTF 寄存器的 SOIFx 位	DMAMUX_RM_INTC 寄存器的 SOIFCx 位	DMAMUX_RM_CHxCFG 寄存器的 SOIE 位
DMAMUX 请求生成通道 y 上的触发溢出事件	DMAMUX_RG_INTF 寄存器的 TOIFy 位	DMAMUX_RG_INTC 寄存器的 TOIFCy 位	DMAMUX_RG_CHxCFG 寄存器的 TOIE 位

触发溢出中断

当 DMAMUX 请求生成触发溢出标志位 TOIFx 置位，并且触发溢出中断使能位 TOIE 位置位，则会产生一个触发溢出中断。写 1 到 DMAMUX_RG_INTC 寄存器的对应触发溢出清除位 TOIFCx 将会清除触发溢出标志位 TOIFx。

同步溢出中断

当 DMAMUX 请求路由同步溢出标志位 SOIFx 置位，并且触发同步溢出中断使能位 SOIE 位置位，则会产生一个同步溢出中断。写 1 到 DMAMUX_RM_INTC 寄存器的对应同步溢出清除位 SOIFCx 将会清除同步溢出标志位 SOIFx。

4.4.6. DMAMUX 映射

DMAMUX 与 DMA0 和 DMA1 配合使用。DMAMUX 的通道 0 到 7 与 DMA0 的通道 0 到 7 相连，DMAMUX 的通道 8 到 15 与 DMA1 的通道 0 到 7 相连。

DMAMUX 请求路由输入映射

DMAMUX 请求路由输入可来自于外设或者 DMAMUX 请求生成器，参考[表 4-3. DMAMUX 请求路由输入信号映射](#)，由 DMAMUX_RM_CHxCFG 寄存器的 MUXID[6:0]位域配置 DMAMUX 请求路由通道 x 的输入。

表 4-3. DMAMUX 请求路由输入信号映射

请求路由通道输入标识MUXID[6:0]	来源
1	Gen_reqx0
2	Gen_reqx1
3	Gen_reqx2
4	Gen_reqx3
5	ADC0_ROUTINE
6	ADC0_INSERTED
7	DAC0_CH0
8	TIMER5_UP

请求路由通道输入标识MUXID[6:0]	来源
9	TIMER6_UP
10	SPI0_RX
11	SPI0_TX
12	SPI1_RX
13	SPI1_TX
14	SPI2_RX
15	SPI2_TX
16	I2C0_RX
17	I2C0_TX
18	I2C1_RX
19	I2C1_TX
20	USART0_RX
21	USART0_TX
22	USART1_RX
23	USART1_TX
24	USART2_RX
25	USART2_TX
26	UART3_RX
27	UART3_TX
28	TIMER0_CH0
29	TIMER0_CH1
30	TIMER0_CH2
31	TIMER0_CH3
32	TIMER0_UP
33	TIMER0_TI
34	TIMER0_COM
35	TIMER7_CH0
36	TIMER7_CH1
37	TIMER7_CH2
38	TIMER7_CH3
39	TIMER7_UP
40	TIMER7_TI
41	TIMER7_COM
42	TIMER1_CH0
43	TIMER1_CH1
44	TIMER1_CH2
45	TIMER1_CH3
46	TIMER1_UP
47	TIMER1_TI

请求路由通道输入标识MUXID[6:0]	来源
48	TIMER2_CH0
49	TIMER2_CH1
50	TIMER2_CH2
51	TIMER2_CH3
52	TIMER2_UP
53	TIMER2_TI
54	TIMER3_CH0
55	TIMER3_CH1
56	TIMER3_CH2
57	TIMER3_CH3
58	TIMER3_UP
59	TIMER3_TI
60	TIMER4_CH0
61	TIMER4_CH1
62	TIMER4_CH2
63	TIMER4_CH3
64	TIMER4_UP
65	TIMER4_TI
66	ADC1_ROUTINE
67	ADC1_INSERTED
68	ADC2_ROUTINE
69	ADC2_INSERTED
70	Reserved
71	Reserved
72	TIMER15_CH0
73	TIMER15_CH1
74	TIMER15_CHLN
75	TIMER15_UP
76	TIMER15_TI
77	TIMER15_COM
78	TIMER16_CH0
79	TIMER16_CH1
80	TIMER16_CHLN
81	TIMER16_UP
82	TIMER16_TI
83	TIMER16_COM
84	CAU_IN
85	CAU_OUT
86	HAU_IN

触发输入映射

DMAMUX 请求生成通道 x 的触发输入可由 DMAMUX_RG_CHxCFG 寄存器的 TID[4:0]位域来配置，参考[表 4-4. 触发输入信号映射](#)。

表 4-4. 触发输入信号映射

触发输入标识TID[4:0]	来源
0	Evtx_out0
1	Evtx_out1
2	Evtx_out2
3	Evtx_out3
4	EXTI_0_INT
5	EXTI_1_INT
6	EXTI_2_INT
7	EXTI_3_INT
8	EXTI_4_INT
9	EXTI_5_INT
10	EXTI_6_INT
11	EXTI_7_INT
12	EXTI_0_EVT
13	EXTI_1_EVT
14	EXTI_2_EVT
15	EXTI_3_EVT
16	EXTI_4_EVT
17	EXTI_5_EVT
18	EXTI_6_EVT
19	EXTI_7_EVT
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved

注意：EXTI_x(x=0...15)仅 EXTI 中断事件发生时会产生 DMA 请求。

同步输入映射

同步输入由 DMAMUX_RM_CHxCFG 寄存器的 SYNCID[4:0]位域来配置，参考[表 4-5. 同步输入信号映射](#)。

表 4-5. 同步输入信号映射

同步输入标识SYNCID[4:0]	来源
0	Evtx_out0
1	Evtx_out1
2	Evtx_out2
3	Evtx_out3
4	EXTI_0_INT
5	EXTI_1_INT
6	EXTI_2_INT
7	EXTI_3_INT
8	EXTI_4_INT
9	EXTI_5_INT
10	EXTI_6_INT
11	EXTI_7_INT
12	EXTI_0_EVT
13	EXTI_1_EVT
14	EXTI_2_EVT
15	EXTI_3_EVT
16	EXTI_4_EVT
17	EXTI_5_EVT
18	EXTI_6_EVT
19	EXTI_7_EVT
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved

4.5. DMAMUX 寄存器

DMAMUX 基地址：0x4002 0800

4.5.1. 请求路由通道 x 配置寄存器（DMAMUX_RM_CHxCFG）

$x = 0 \dots 11$ ，其中 x 为通道序号

地址偏移：0x00 + 0x04 * x

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:29	保留	必须保持复位值。
28:24	SYNCID[4:0]	同步输入标识 选择同步输入源。
23:19	NBR[4:0]	传递的DMA请求数量 在同步输入事件之后，或者通道事件输出之前，将传递到DMA控制器的DMA请求数量为NBR[4:0] + 1。 该位域只能在SYNCEN位和EVGEN位都禁能时才能配置。
18:17	SYNCP[1:0]	同步输入极性 00：不检测事件 01：上升沿 10：下降沿 11：上升和下降沿
16	SYNCEN	同步模式使能 0：禁能同步模式 1：使能同步模式
15:10	保留	必须保持复位值。
9	EVGEN	事件输出使能 0：禁能事件输出

		1: 使能事件输出
8	SOIE	同步溢出中断使能 0: 禁能中断 1: 使能中断
7	保留	必须保持复位值。
6:0	MUXID[6:0]	请求路由标识 选择DMAMUX请求路由通道的DMA请求输入源。

4.5.2. 请求路由通道中断标志位寄存器 (DMAMUX_RM_INTF)

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				SOIF11	SOIF10	SOIF9	SOIF8	SOIF7	SOIF6	SOIF5	SOIF4	SOIF3	SOIF2	SOIF1	SOIF0
				r	r	r	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
31:12	保留	必须保持复位值。
11:0	SOIFx	请求路由通道x的同步溢出事件标志位(x=0..11) 如果同步输入事件发生时, DMAMUX请求路由计数器值小于NBR[4:0], 则该位置位。 通过对DMAMUX_RM_INTC寄存器的SOIFCx位写1来清除相应通道的同步溢出标志。

4.5.3. 请求路由通道中断标志位清除寄存器 (DMAMUX_RM_INTC)

地址偏移: 0x84

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				SOIFC11	SOIFC10	SOIFC9	SOIFC8	SOIFC7	SOIFC6	SOIFC5	SOIFC4	SOIFC3	SOIFC2	SOIFC1	SOIFC0

W W W W W W W W W W W

位/位域	名称	描述
31:12	保留	必须保持复位值。
11:0	SOIFCx	请求路由通道x的同步溢出事件标志清除位(x=0..11) 写1可清除相应通道在DMAMUX_RM_INTF寄存器的同步溢出标志SOIFx。

4.5.4. 请求生成通道 x 配置寄存器 (DMAMUX_RG_CHxCFG)

x = 0...3, 其中 x 为通道序号

地址偏移: $0x100 + 0x04 * x$

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。

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

位/位域	名称	描述
31:24	保留	必须保持复位值。
23:19	NBRG[4:0]	待产生的DMA请求数量 在触发输入事件之后, 待产生的DMA请求数量为NBRG[4:0] + 1。 注意: 只有当RGEN位为0时才能写该位域。
18:17	RGTP[1:0]	DMAMUX请求生成触发输入极性 00: 不检测事件 01: 上升沿 10: 下降沿 11: 上升沿和下降沿
16	RGEN	DMAMUX请求生成通道x使能 0: 禁能DMAMUX请求生成通道x 1: 使能DMAMUX请求生成通道x
15:9	保留	必须保持复位值。
8	TOIE	触发溢出中断使能 0: 禁能中断

1: 使能中断

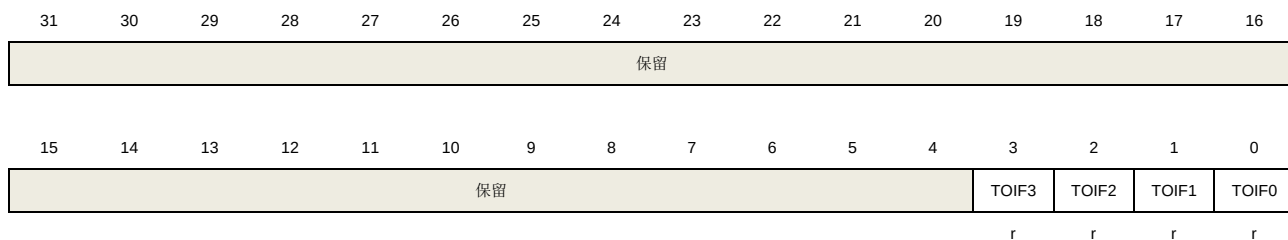
7:5	保留	必须保持复位值。
4:0	TID[4:0]	触发输入标识 选择DMAMUX请求生成通道的触发输入源。

4.5.5. 请求生成通道中断标志位寄存器 (DMAMUX_RG_INTF)

地址偏移: 0x140

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



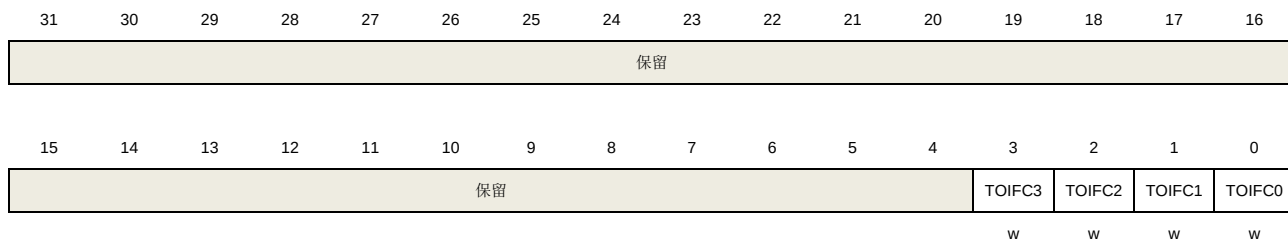
位/位域	名称	描述
31:4	保留	必须保持复位值。
3:0	TOIFx	DMAMUX请求生成通道x的触发溢出标志位(x=0..3) 如果触发输入事件在DMAMUX请求生成计数器下溢之前发生, 则该位置位。 通过对DMAMUX_RG_INTC寄存器的TOIFCx位写1来清除相应通道的触发溢出标志。

4.5.6. 请求生成通道中断标志位清除寄存器 (DMAMUX_RG_INTC)

地址偏移: 0x144

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



位/位域	名称	描述

31:4	保留	必须保持复位值。
3:0	TOIFCx	DMAMUX请求生成通道x的触发溢出标志清除位(x=0..3) 写1可清除相应通道在DMAMUX_RG_INTF寄存器的触发溢出标志TOIFx。

5. 闪存控制器（FMC）

5.1. 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 256K 字节空间内，CPU 执行指令零等待。FMC 也提供了页擦除，整片擦除，以及 32 位整字/16 位半字编程等闪存操作。

5.2. 主要特征

- 高达1024KB字节的片上闪存可用于存储指令或数据；
- 在闪存的前256K字节空间内，CPU执行指令零等待，在此范围外，CPU读取指令存在较长延时；
- 使用了两片闪存，前512KB容量在第一片闪存（bank0）中，后续的容量在第二片闪存（bank1）中；
- Bank0的闪存页大小为2KB，bank1的闪存页大小为4KB；
- 支持32位整字/16位半字编程，页擦除和整片擦除操作；
- 大小为16字节的选项字节块可根据用户需求配置；
- 每次系统复位后选项字节中的内容将重新加载到选项字节控制寄存器中；
- 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。
- 18K字节信息块，用于引导装载程序；
- 64字节OTP0（一次性编程）块用于用户数据存储。
- 128K字节OTP1用于引导入口或用户数据存储。
- 256字节OTP2具有写锁和读锁，用于用户数据存储。
- 48字节OTP3用于关键安全配置。

5.3. 功能说明

5.3.1. 闪存结构

对于 GD32F50x，闪存最多有 256 页的 2K 字节和 128 页的 4K 字节。每一页都可以单独擦除。

闪存结构见[表 5-1. GD32F50x 闪存基地址和构成](#)。

表 5-1. GD32F50x 闪存基地址和构成

闪存块		名称	地址范围	大小
主闪存块	Bank0 512KB	第0页	0x0800 0000 - 0x0800 07FF	2KB
		第1页	0x0800 0800 - 0x0800 0FFF	2KB
		...	...	...
		第255页	0x0807 F800 - 0x0807 FFFF	2KB
	Bank1 512KB	第256页	0x0808 0000 – 0x0808 0FFF	4KB
		第257页	0x0808 1000 – 0x0808 1FFF	4KB
		...		...
		第383页	0x080F F000 – 0x080F FFFF	4KB
信息块		Boot loader	0x1FFF B000- 0x1FFF F7FF	18KB
选项字节块		选项字节	0x1FFF F800 - 0x1FFF F80F	16B
OTP0 Block		数据块	0x1FFF 7800 - 0x1FFF 783F	64B
		锁定块	0x1FFF 7840 - 0x1FFF 787F	64B
OTP1 Block		数据块	0x1FF0 0000 - 0x1FF1 FFFF	128KB
		锁定块	0x1FF2 0100 - 0x1FF2 010F	16B
OTP2 Block		数据块	0x1FF2 0000 - 0x1FF2 00FF	256B
		锁定块	0x1FF2 0110 - 0x1FF2 018F	128B
OTP3 Block		数据块	0x1FFF 7900 - 0x1FFF 792F	48B
		锁定块	0x1FFF 7930 - 0x1FFF 793F	16B

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

5.3.2. 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 CBUS 总

线。

对于代码区为 128KB 和 192KB 的器件，闪存在执行擦除和编程操作时会占用 CBUS，此时中断响应会被延迟。对于代码区为 256KB 的器件不受影响。不同器件的代码区大小可参考数据手册。可将中断向量表和中断中断回调函数放在 SRAM 中规避该问题。

5.3.3. FMC_CTLx / FMC_OBCTLx 寄存器解锁

复位后，FMC_CTLx 寄存器无法以写模式访问，且 FMC_CTLx 寄存器中的 LK 位为 1。解锁序列包括两次写操作，向 FMC_KEY0 寄存器写入数据以打开对 FMC_CTL0 寄存器的访问权限。这两次写操作分别是向 FMC_KEY0 寄存器写入 0x45670123 和 0xCDEF89AB。在完成这两次写操作后，FMC_CTL0 寄存器中的 LK 位会被硬件复位为 0。软件可以通过将 FMC_CTL0 寄存器中的 LK 位设置为 1 来重新锁定 FMC_CTL。任何对 FMC_KEY0 的错误操作都会将 LK 位设置为 1，从而锁定 FMC_CTL0 寄存器，并导致总线错误。

即使 FMC_CTL 已解锁，FMC_OBCTLx (x = 0,1,2) 寄存器仍然受到保护。解锁序列包括两次写操作，分别向 FMC_OBKEY 寄存器写入 0x45670123 和 0xCDEF89AB。在完成这两次写操作后，FMC_OBCTL0 寄存器中的 OB_LK 位会被硬件复位为 0。软件可以通过先将 FMC_CTL0 寄存器中的 LK 位设置为 1，再将 FMC_OBCTL0 寄存器中的 OB_LK 位设置为 1 来重新锁定 FMC_OBCTLx。

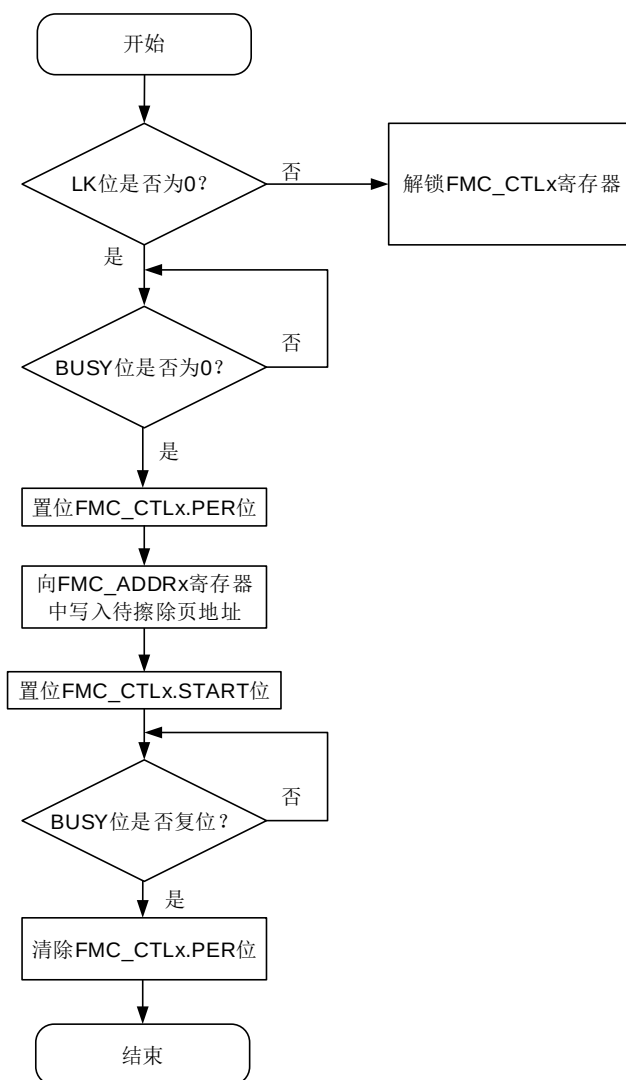
5.3.4. 页擦除

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。FMC 擦除页步骤如下：

1. 确保 FMC_CTLx 寄存器不处于锁定状态；
2. 检查 FMC_STATx 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；
3. 置位 FMC_CTLx 寄存器的 PER 位；
4. 将待擦除页的绝对地址 (0x08XX XXXX) 写到 FMC_ADDRx 寄存器；
5. 通过将 FMC_CTLx 寄存器的 START 位置 1 来发送页擦除命令到 FMC；
6. 等待擦除指令执行完毕，FMC_STATx 寄存器的 BUSY 位清 0；
7. 如果需要，使用 CBUS 读并验证该页是否擦除成功。

当页擦除成功执行，FMC_STATx 寄存器的 ENDF 位将置位。若 FMC_CTLx 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_CTLx 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_STATx 寄存器的 WPERR 位来判断该中断是否发生。[图 5-1. 页擦除操作流程](#)显示了页擦除操作流程。

图 5-1. 页擦除操作流程



FMC_STAT0 寄存器反应对 bank0 和选项字节块的操作状态，FMC_STAT1 反应对 bank1 的操作状态。对 bank1 的页擦除操作与对 bank0 的页擦除操作类似。需要注意的是，在安全保护状态下，对 bank1 的页擦除，需将地址同时写至 FMC_ADDR1 和 FMC_ADDR0 寄存器。

5.3.5. 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_CTL0 寄存器中 MER 为 1 时，擦除过程仅作用于 Bank0，当设置 FMC_CTL1 寄存器中 MER 为 1 时，擦除过程仅作用于 Bank1，当设置 FMC_CTL0 和 FMC_CTL1 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：

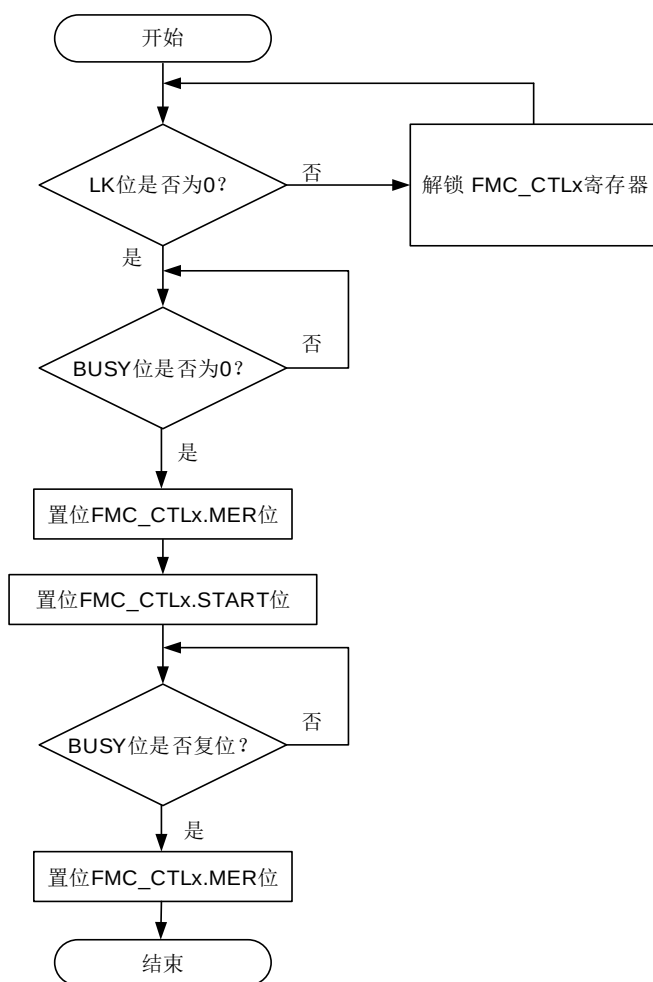
1. 确保FMC_CTLx寄存器不处于锁定状态；

2. 等待FMC_STATx寄存器的BUSY位变为0;
3. 如果单独擦除Bank0, 置位FMC_CTL0寄存器的MER位。如果单独擦除Bank1, 置位FMC_CTL1寄存器的MER位。如果整片擦除闪存, 同时置位FMC_CTL0和FMC_CTL1寄存器的MER位;
4. 通过将FMC_CTLx寄存器的START位置1来发送整片擦除命令到FMC;
5. 等待擦除指令执行完毕, FMC_STATx寄存器的BUSY位清0;
6. 如果需要, 使用CBUS读并验证是否擦除成功。

当整片擦除成功执行, FMC_STATx 寄存器的 ENDF 位置位。若 FMC_CTLx 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。

[图 5-2. 整片擦除操作流程](#)显示了整片擦除操作流程。

图 5-2. 整片擦除操作流程



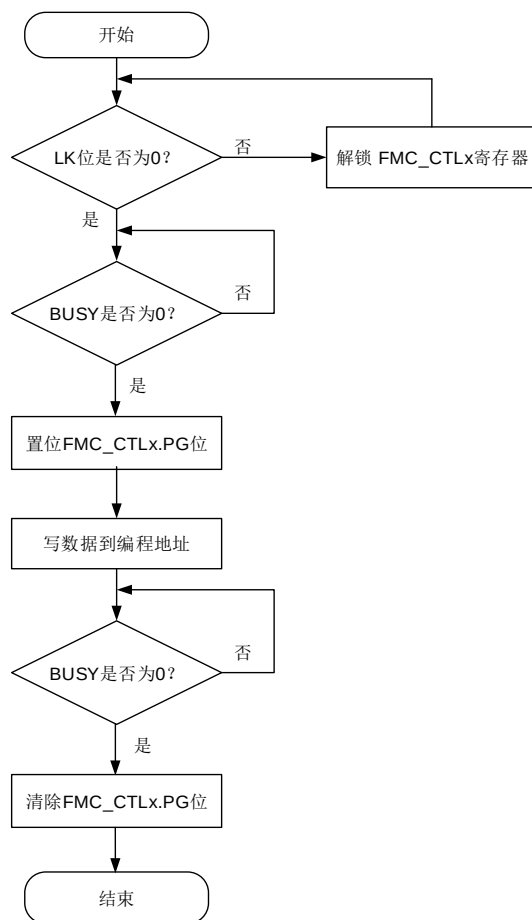
5.3.6. 主存储闪存块编程

FMC 提供了一个 32 位整字/16 位半字编程功能，用来修改主存储闪存块内容。编程操作使用各寄存器流程如下：

1. 确保FMC_CTLx寄存器不处于锁定状态；
2. 等待FMC_STATx寄存器的BUSY位变为0；
3. 置位FMC_CTLx寄存器的PG位；
4. CBUS写一个32位整字/16位半字到目的绝对地址（0x08XX XXXX）；
5. 等待编程指令执行完毕，FMC_STATx寄存器的BUSY位清0；
6. 如果需要，使用CBUS读并验证是否编程成功。

当主存储块编程成功执行，FMC_STATx 寄存器的 ENDF 位置位。若 FMC_CTLx 寄存器的 ENDIE 位被置 1，FMC 将触发一个中断。需要注意的是，执行整字/半字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，对该地址写一个非 0x0 值，FMC_STATx 寄存器的 PGERR 位将被置 1，对该地址的编程操作无效（当写内容为 0x0 时，即使目的地址没有被正常擦除，也可以正确编程）。另一方面，如果目的地址在一个处于擦除和编程保护的页中，编程不会成功且 FMC_STATx 寄存器的 WPERR 位将会置位。在这两种情形下，如果 FMC_CTLx 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，可以检查 FMC_STATx 寄存器的 PGERR 位和 WPERR 位来判断哪一种错误发生了。[图 5-3. 字编程操作流程](#)显示了主存储块编程操作流程。

图 5-3. 字编程操作流程



注意：当 CPU 进入省电模式时，对闪存的操作将失败。

5.3.7. 选项字节块修改

FMC 提供擦写编程功能用来修改选项字节块内容。选项字节块共有 8 对选项字节。每对选项字节的高字节是低字节的补。当低字节被修改时，FMC 自动生成该选项字节的高字节。字节块编程操作过程如下。

1. 确保FMC_CTL0寄存器不处于锁定状态；
2. 等待FMC_STAT0寄存器的BUSY位变为0；
3. 解锁FMC_CTL0寄存器的选项字节操作位；
4. 通过编程FMC_OBCTLx寄存器写入新的选项字节内容；
5. 通过在FMC_OBCTL0寄存器中设置OB_START位，向FMC发送选项字节修改命令；
6. 通过检查FMC_STAT0寄存器中BUSY位的值，等待所有操作完成；
7. 如果需要，使用CBUS访问读取并验证闪存。

当选项字节块编程成功执行，FMC_STAT0寄存器的ENDF位置位。若FMC_CTL0寄存器的ENDIE

位被置 1，FMC 将触发一个中断。

注意：修改后的选项字节仅在产生系统复位（应置位 FMC_CTL0 的 NWLDE 位）后生效。

5.3.8. 选项字节块说明

每次系统复位（应置位 FMC_CTL0 的 NWLDE 位）后，闪存选项字节块被重加载到 FMC_OBSTAT 和 FMC_WP 寄存器，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[表 5-2. 选项字节](#)。

表 5-2. 选项字节

地址	名称	出厂值	说明
0x1fff f800	SPC	0xAA	选项字节安全保护值 0xAA：无安全保护 0xCC：安全保护级别高 除0xAA或0xCC以外的任何值：安全保护级别低。
0x1fff f801	SPC_N	0x55	SPC补字节
0x1fff f802	USER	0x9F	[7]：保留 [6:5]：NWLD_CLK 选择无等待时间区域加载的时钟。 00：200M PLL CLK 01：160M PLL CLK 10：120M PLL CLK 11：8M IRC8M CLK [4]：ECC_EN 0：禁用SRAM ECC 1：启用SRAM ECC [3]：SRAM_RST 0：电源复位后初始化SRAM。 1：电源复位后不初始化SRAM。 [2]：nRST_STDBY 0：设置待机模式时产生复位而不是进入待机模式； 1：设置待机模式时进入待机模式而不产生复位。 [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

地址	名称	出厂值	说明
			1: 软件使能独立看门狗功能
0x1fff f803	USER_N	0x60	USER补字节值
0x1fff f804	DATA[7:0]	0xFF	用户定义数据7到0位
0x1fff f805	DATA_N[7:0]	0xFF	DATA补字节值的7到0位
0x1fff f806	DATA[15:8]	0xFF	用户定义数据15到8位
0x1fff f807	DATA_N[15:8]	0xFF	DATA补字节值的15到8位
0x1fff f808	WP[7:0]	0xFF	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]	0xFF	WP补字节值的7到0位
0x1fff f80a	WP[15:8]	0xFF	页擦除/编程的保护值的15到8位
0x1fff f80b	WP_N[15:8]	0xFF	WP补字节值的15到8位
0x1fff f80c	WP[23:16]	0xFF	页擦除/编程的保护值的23到16位
0x1fff f80d	WP_N[23:16]	0xFF	WP补字节值的23到16位
0x1fff f80e	WP[31:24]	0xFF	页擦除/编程的保护值的31到24位
0x1fff f80f	WP_N[31:24]	0xFF	WP补字节值的31到24位

5.3.9. 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_STATx 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。配置选项字节块的 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。

表 5-3. 页保护 WP 位

WP 位	所保护的页
WP[0]	Page0, Page1
WP[1]	Page2, Page3
...	...
WP[29]	Page58, Page59
WP[30]	Page60, Page61
WP[31]	Page62 - Page383

5.3.10. OTP 闪存块编程

OTP 编程方法与 Bank0 相同，使用 FMC_CTL0 和 FMC_STAT0 寄存器。所有 OTP 只能编程一次，且无法擦除。锁定块的每个字节只能从 0xFF 编程为 0x00，不能设置为其他值。

FMC 提供了一个 32 位整字 / 16 位半字 / 8 位字节的编程功能，用于修改 OTP0/OTP1/OTP2/OTP3 的内容。可用的编程宽度如[表 5-4. OTP 可用编程位宽](#)。

表 5-4. OTP 可用编程位宽

块	名称	大小	可用编程位宽		
			32位	16位	8位
OTP0	数据区	64*1字节	-	-	1
	锁定区	1*64字节	-	-	1
OTP1	数据区	16*8K字节	1	1	1
	锁定区	1*16字节	-	-	1
OTP2	数据区	64*4字节	1	1	1
	锁定区	1*128字节	-	-	1
OTP3	数据区	3*4字节	1	-	-
	锁定区	3*4字节	1	-	-

OTP0 块可以划分为 64 个数据块（每块包含 1 字节）和 1 个锁定块（包含 64 字节）。锁定块的地址范围为 0x1FFF 7840 到 0x1FFF 787F。数据块的地址范围为 0x1FFF 7800 到 0x1FFF 783F。每个锁定字节（0x00：锁定，0xFF：不锁定）可以锁定对应的数据块，从而防止对该数据块进行编程。位于地址 0x1FFF 7840 的锁定字节 0 锁定地址 0x1FFF 7800 的数据块 0，以此类推。

表 5-5. OTP0 锁

锁字节	锁字节地址	被锁数据块	被锁数据地址
0	0x1FFF 7840	0	0x1FFF 7800
1	0x1FFF 7841	1	0x1FFF 7801
...	...	...	...
62	0x1FFF 787E	62	0x1FFF 783E
63	0x1FFF 787F	63	0x1FFF 783F

OTP1 块可以划分为 16 个数据块（每块包含 8K 字节）和 1 个锁定块（包含 16 字节）。数据块的地址范围为 0x1FF0 0000 到 0x1FF1 FFFF。锁定块的地址范围为 0x1FF2 0100 到 0x1FF2 010F。每个锁定字节（0x00：锁定，0xFF：不锁定）可以锁定对应的数据块，从而防止对该数据块进行编程。OTP1 数据块是否可读由 FMC_OTP1CFG 寄存器中的 OTP1REN[15:0]决定。读取被锁定的数据块将会导致总线错误。

表 5-6. OTP1 锁

锁字节	锁字节地址	被锁数据块	被锁数据地址
0	0x1FF2 0100	0	0x1FF0_0000 - 0x1FF0 1FFF
1	0x1FF2 0101	1	0x1FF0 2000 - 0x1FF0 3FFF
...	...	...	...
14	0x1FF2 010E	14	0x1FF1 C000 - 0x1FF1 DFFF
15	0x1FF2 010F	15	0x1FF1 E000 - 0x1FF1 FFFF

OTP2 块可以划分为 64 个数据块（每块包含 4 字节）和 1 个锁定块（包含 128 字节）。数据块的

地址范围为 0x1FF2 0000 到 0x1FF2 00FF。写锁定块的地址范围为 0x1FF2 0110 到 0x1FF2 018F。每个锁定字节（0x00：锁定，0xFF：不锁定）可以锁定对应的数据块，从而防止对该数据块进行编程。位于地址 0x1FF2 0110 的锁定字节 0 锁定数据块 0，以此类推。

读锁定块的地址范围为 0x1FF2 0150 到 0x1FF2 018F。每个锁定字节（0x00：锁定，0xFF：不锁定）可以锁定对应的数据块，从而防止读取。位于地址 0x1FF2 0170 的锁定字节 32 锁定数据块 0，以此类推。

当 FMC_CTL0 寄存器中的 RLBE 位被置位时，OTP2 数据块中对应于读锁定块的数据将无法读取。例如，安全验证数据存储在 OTP2 中，且从 OTP1 开始的安全启动程序可以读取 OTP2 信息进行验证。在完成验证后，RLBE 被设置，然后跳转到其他程序。在下次复位之前，与读锁定块对应的 OTP2 数据块都无法读取。

表 5-7. OTP2 锁

写锁字节	写锁字节地址	读锁字节	读锁字节地址	被锁数据块	被锁数据地址
0	0x1FF2 0110	64	0x1FF2 0150	0	0x1FF2 0000 - 0x1FF2 0003
1	0x1FF2 0111	65	0x1FF2 0151	1	0x1FF2 0004 - 0x1FF2 0007
...	...	...	...	...	...
63	0x1FF2 014F	127	0x1FF2 018F	63	0x1FF2 00FC - 0x1FF2 00FF

OTP3 块可以划分为 3 个数据块（每块包含 4 字节）和 3 个锁定块（每块包含 4 字节）。锁定块的地址分别为 0x1FFF 7930、0x1FFF 7934 和 0x1FFF 7938。数据块的地址分别为 0x1FFF 7900、0x1FFF 7910 和 0x1FFF 7920。每个锁定字（全 0：锁定，全 1：不锁定）可以锁定对应的数据块，从而防止对该数据块进行编程。地址范围 0x1FFF 7930 - 0x1FFF 7933 的锁定块 0 锁定地址范围 0x1FFF 7900 - 0x1FFF 7903 的数据块 0，以此类推。OTP3 在成功编程后将立即生效。

表 5-8. OTP3 锁

锁块	锁块地址 只可写	数据块	被锁数据地址 只可写	数据名称 全0：有效('1')
0	0x1FFF 7930 - 0x1FFF 7933	0	0x1FFF 7900 - 0x1FFF 7903	NDBG
1	0x1FFF 7934 - 0x1FFF 7937	1	0x1FFF 7910 - 0x1FFF 7913	NBTSB
2	0x1FFF 7938 - 0x1FFF 793B	2	0x1FFF 7920 - 0x1FFF 7923	BTFOSSEL

5.3.11. 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。[表 5-9. 安全保护](#)表示不同配置的安全保护等级，安全保护等级划分三等。

无保护状态：当 SPC 字节设置为 0xAA，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。

保护等级低：当设置 SPC 字节为除 0xAA 或 0xCC 外的任何值，激活低安全保护等级。主存储闪存块仅能被用户代码访问。在调试模式或者从 SRAM 中启动或者从 bootloader 模式启动时，这些模式下对主存储块的操作都被禁止。无论是在调试模式或者从 SRAM 中启动，还是从 bootloader

模式启动，如果对主存储块执行一次读操作，将会产生一个总线错误。在调试模式或者从 SRAM 中启动时或者从 bootloader 模式启动，如果对主存储块执行一次编程/擦除操作，FMC_STATx 寄存器中的 WPERR 位会被置位。在低安全保护等级下，对于选项字节的所有操作都被允许。如果通过设置 SPC 字节为 0xAA 进入无保护状态，主存储闪存块将执行一次整片擦除操作。

保护等级高：当设置 SPC 字节为 0xCC，激活高安全保护等级。当编程选择该保护等级时，调试模式，从 SRAM 中启动，或者从 bootloader 启动都被禁止。主存储闪存块可由用户代码的所有操作进行访问。SPC 字节禁止再次编程。所以，如果高保护等级被激活，将不能再降回到低保护等级或无保护等级。

表 5-9. 安全保护

SPC[7:0]	安全保护
0xAA	无保护
除0xAA或0xCC之外	保护等级低
0xCC	保护等级高

5.3.12. 频率控制

FMC 模块使用时钟为 CK_FMC 访问 sip flash，与系统时钟 CK_SYS 需要满足一定的频率关系，CK_FMC 需要不慢于系统时钟 CK_SYS，但不高于系统时钟的 7 倍：

$$CK\_FMC \geq CK\_SYS \geq 1/7 CK\_FMC$$

推荐的切频配置方法如下：

如果升频：

1. 需要保持CK_FMC选择时钟源为系统时钟CK_SYS，同时将CK_FMC与CK_SYS的频率提高；
2. 再根据需要单独提高CK_FMC频率；

如果降频：

1. 先将CK_FMC选择时钟源为系统时钟CK_SYS；
2. CK_FMC与CK_SYS同步降频。

5.4. FMC 寄存器

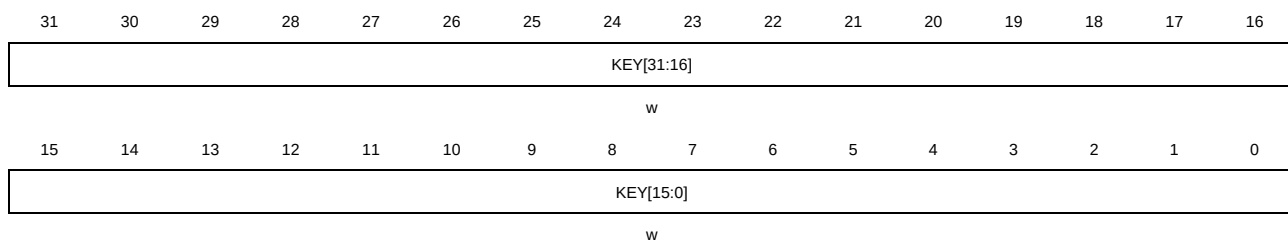
FMC 基地址：0x4002 2000

5.4.1. 解锁寄存器 (FMC_KEY0)

地址偏移：0x04

复位值：0x0000 0000

该寄存器只能按字(32 位)访问



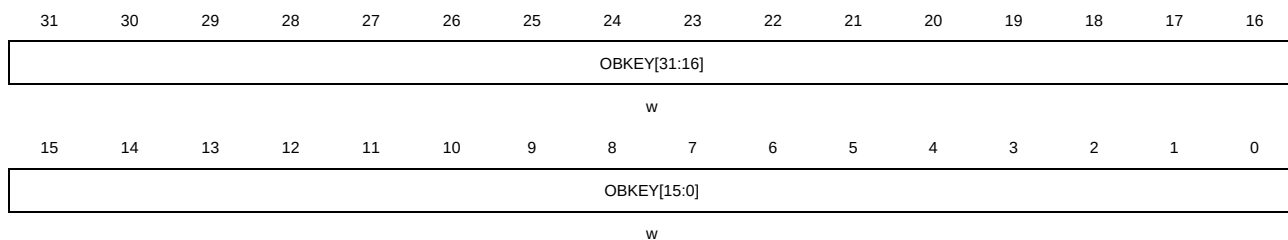
位/位域	名称	描述
31:0	KEY[31:0]	FMC_CTL0解锁寄存器 这些位仅能被软件写。 写解锁值到KEY[31:0]可以解锁FMC_CTL0寄存器。

5.4.2. 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



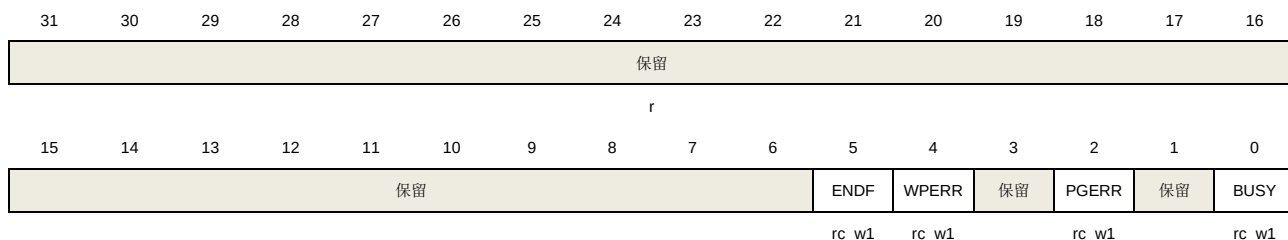
位/位域	名称	描述
31:0	OBKEY[31:0]	选项字节操作解锁寄存器 这些位仅软件可写。 写解锁值到OBKEY[31:0]解锁FMC_OBCTLx (x = 0,1,2)寄存器的选项字节命令。

5.4.3. 状态寄存器 0 (FMC_STAT0)

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	保留	必须保持复位值。
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。

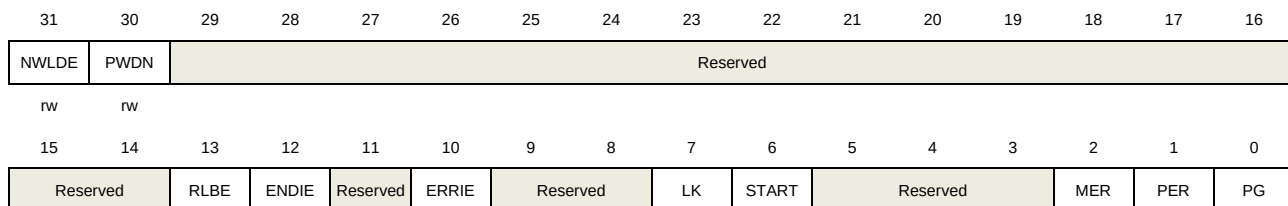
5.4.4. 控制寄存器 0 (FMC_CTL0)

地址偏移: 0x10

复位值: 0x0000 0080

该寄存器只能按字(32 位)访问

注意: 当相应闪存操作完成后，该寄存器需处于复位状态。



		W	RW	RW	RS	RS		RW	RW	RW
位/位域	名称	描述								
31	NWLDE	在系统复位时使能零等待时间区域加载。零等待时间区域加载包括加载选项字节、一次性可编程（OTP）和主闪存代码区。 该位由软件设置或清除，系统复位后不复位，上电复位后复位。 0：系统复位后不拷贝闪存内容到缓冲区。 1：系统复位后拷贝闪存内容到缓冲区。								
30	PWDN	当没有操作时，闪存进入深度掉电模式。 该位由软件设置或清除，系统复位后不复位，上电复位后复位。 1：进入深度掉电模式 0：不进入深度掉电模式 注意： 1、在省电模式下，闪存仅在PWDN位置为1时才会进入深度掉电模式。 2、在使能CPU Cbus 超时（CPUCBUSTO=1）且PWDN=1的情况下，闪存进入深度掉电模式后，访问闪存非零等待区（data区）会导致CBUS超时，引发Hardfault 错误。 3、在未使能 CPU Cbus 超时（CPUCBUSTO=0）且 PWDN=1 的情况下，闪存进入深度掉电模式后，访问闪存非零等待区（data 区）会唤醒闪存，需要等待闪存唤醒时间，访问闪存零等待区（code 区）不会唤醒闪存且无需等待。								
29:14	保留	必须保持复位值。								
13	RLBE	使能OTP2读锁定块。 只能软件置1使能读锁定。软件一旦置1，读锁定块对应的数据块无法被读。 复位后恢复复位值。								
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								
9:8	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。								

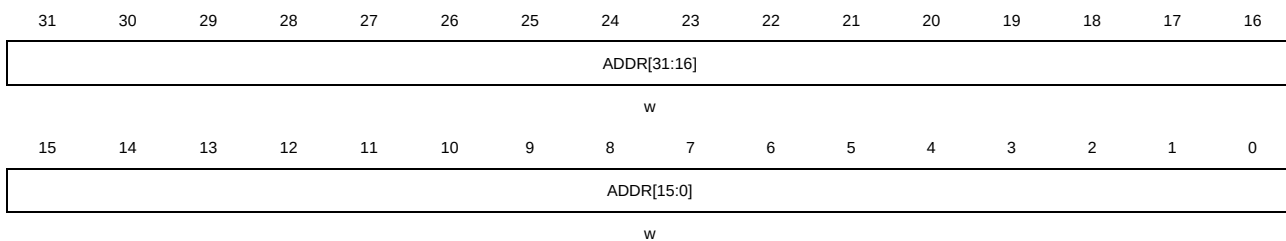
6	START	向FMC发送擦除命令位 软件置1可以发送擦除命令到FMC。当BUSY位被清0时，此位由硬件清0。
5:3	保留	必须保持复位值。
2	MER	主存储块整片擦除命令位 软件置1和清0。 0: 无作用 1: 主存储块整片擦除命令
1	PER	主存储块页擦除命令位 软件置1和清0。 0: 无作用 1: 主存储块页擦除命令
0	PG	主存储块编程命令位 软件置1和清0。 0: 无作用 1: 主存储块编程命令

5.4.5. 地址寄存器 0 (FMC_ADDR0)

地址偏移: 0x14

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问



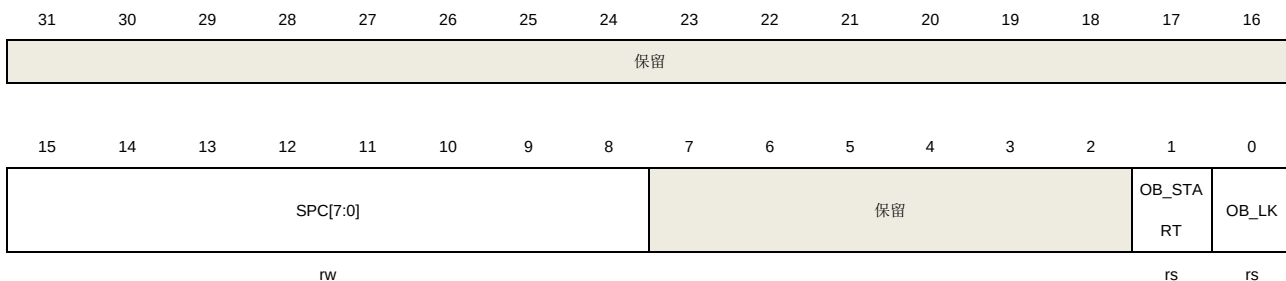
位/位域	名称	描述
31:0	ADDR[31:0]	闪存擦除地址 该位通过软件设置。 ADDR位是闪存擦除命令的地址。

5.4.6. 选项字节控制寄存器 0 (FMC_OBCTL0)

地址偏移: 0x18

复位值: 0x0XXX XXXX

该寄存器只能按字(32 位)访问



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	SPC[7:0]	选项字节安全保护代码 0xAA: 无安全保护 0xCC: 安全保护高 除0xAA或0xCC之外任何值: 安全保护低
7:2	保留	必须保持复位值。
1	OB_START	发送选项字节命令到FMC 该位由软件设置。 当BUSY位清0时由硬件清除该位。
0	OB_LK	FMC_OBCTL0/1/2锁定位 当往FMC_OBKEY寄存器写值顺序正确时, 该位由硬件清0。 软件置位。

5.4.7. 选项字节控制寄存器 1 (FMC_OBCTL1)

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块的DATA[15:0]部分

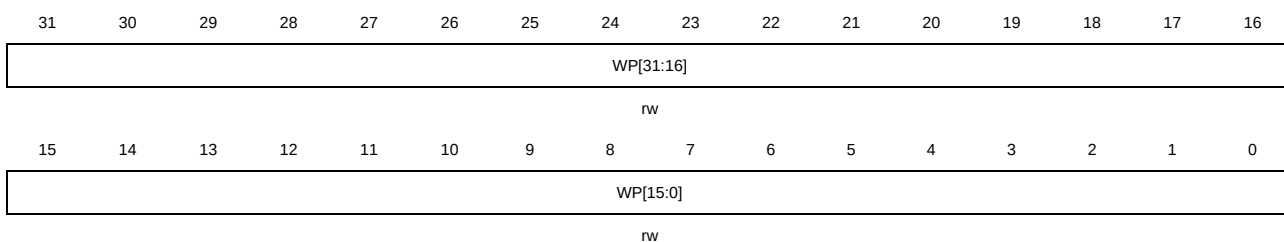
9:2	USER[7:0]	系统复位后保存选项字节块的USER字节
1	保留	必须保持复位值。
0	OBERR	选项字节读错误位 当选项字节和它的补字节不匹配时此位由硬件置1，选项字节被强制设置为0xFF。

5.4.8. 选项字节控制寄存器 2 (FMC_OBCTL2)

地址偏移：0x20

复位值：0xFFFF XXXX

该寄存器只能按字(32 位)访问。



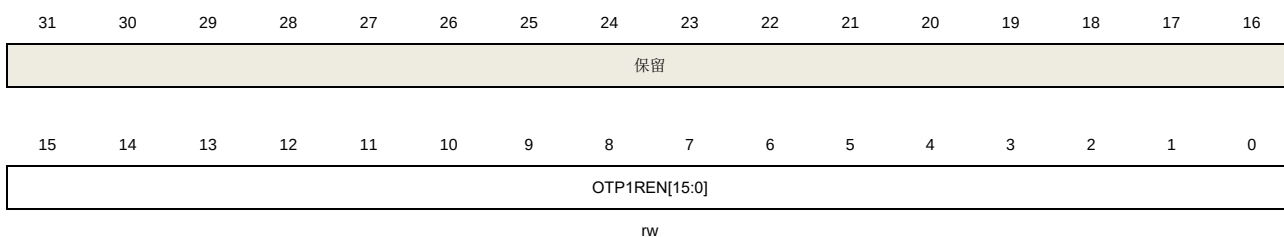
位/位域	名称	描述
31:0	WP[31:0]	系统复位后保存选项字节块的WP[31:0]部分。

5.4.9. OTP1 配置寄存器 (FMC_OTP1CFG)

地址偏移：0x24

复位值：0x0000 FFFF.

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	OTP1REN[15:0]	使能OTP1读 OTP1REN[x]决定OTP1数据块x是否可读，x=0..15。 0：数据块不可读 1：数据块可读

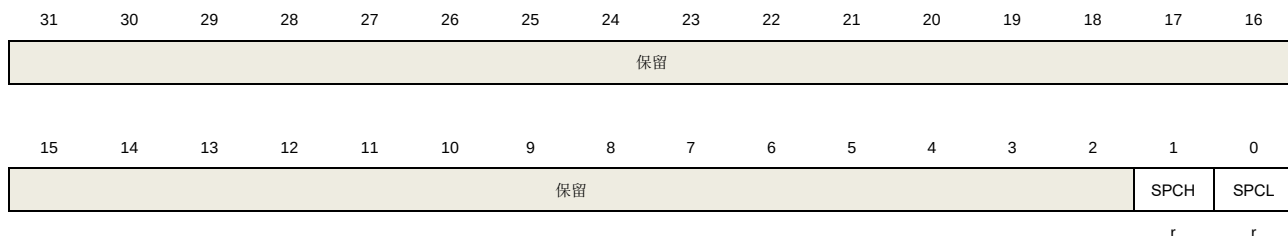
软件可写0，但仅复位后设为1。

5.4.10. 选项字节状态寄存器 (FMC_OBSTAT)

地址偏移：0x40

复位值：0x0000 0000.

该寄存器只能按字（32 位）访问。



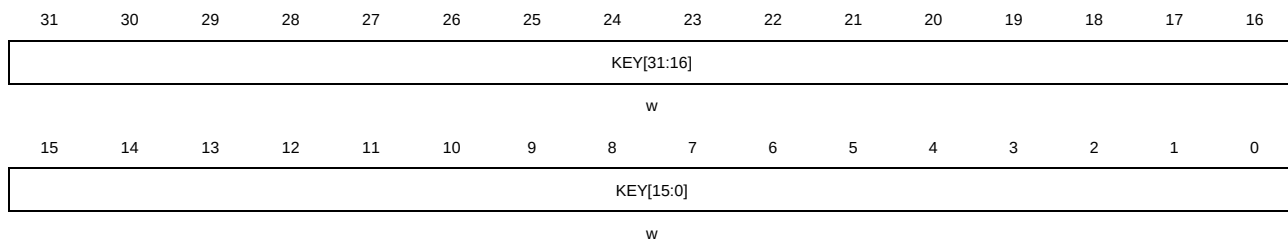
位/位域	名称	描述
31:2	保留	必须保持复位值。
1	SPCH	芯片当前安全保护为高等级时为1。
0	SPCL	芯片当前安全保护为低等级时为1。

5.4.11. 解锁寄存器 1 (FMC_KEY1)

地址偏移：0x44

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_CTL1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_CTL1寄存器。

5.4.12. 状态寄存器 1 (FMC_STAT1)

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
保留										ENDF	WPERR	保留	PGERR	保留	BUSY
										rc_w1	rc_w1		rc_w1		rc_w1

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	保留	必须保持复位值。
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。

5.4.13. 控制寄存器 1 (FMC_CTL1)

地址偏移: 0x50

复位值: 0x0000 0080

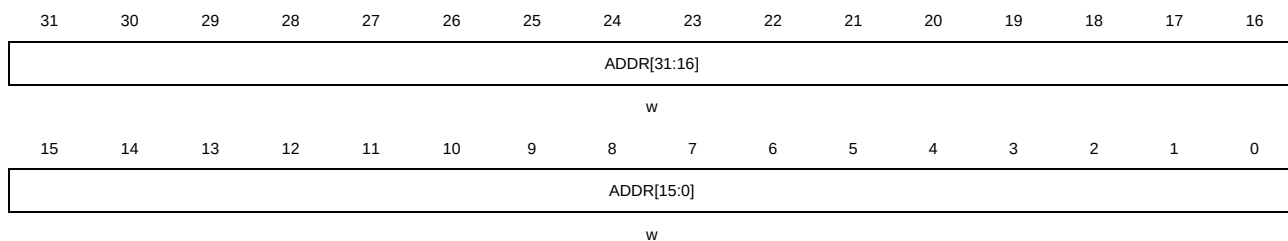
该寄存器只能按字(32 位)访问。

注意: 当相应闪存操作完成后，该寄存器需处于复位状态。

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			ENDIE	保留	ERRIE	保留			LK	START	保留		MER	PER	PG

地址偏移: 0x54
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]	闪存擦除地址 该位通过软件设置。 ADDR位是闪存擦除命令的地址

5.4.15. OTP3 状态寄存器 (FMC_OTP3_STAT)

地址偏移: 0x60

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	BTFOSEL_LK	BTFOSEL 锁状态 0: 未锁定 1: 锁定
5	NBTSB_LK	NBTSB 锁状态 0: 未锁定 1: 锁定
4	NDBG_LK	NDBG 锁状态 0: 未锁定 1: 锁定

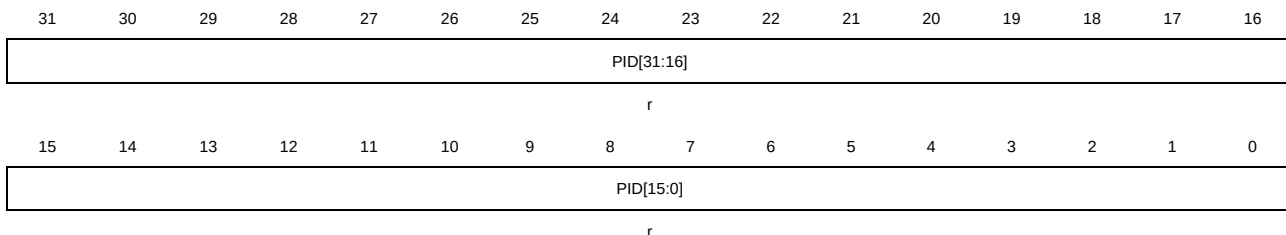
3	保留	必须保持复位值。
2	BTFOSEL	选择从主闪存或 OTP1 启动。NBTSB=1 或 BOOT0=0 时该位有效。 0: 从主闪存启动 1: 从 OTP1 启动
1	NBTSB	禁止从 SRAM 或 bootloader 中启动 0: 能够从 SRAM 或 bootloader 中启动 1: 禁止从 SRAM 或 bootloader 中启动
0	NDBG	设置调试权限 0: 无影响 1: 不可调试

5.4.16. 产品 ID 寄存器 (FMC_PID)

地址偏移: 0x100

复位值: 0xFFFF XXXX

该寄存器只能按字(32 位)访问



位/位域	名称	描述
31:0	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。

6. 电源管理单元（PMU）

6.1. 简介

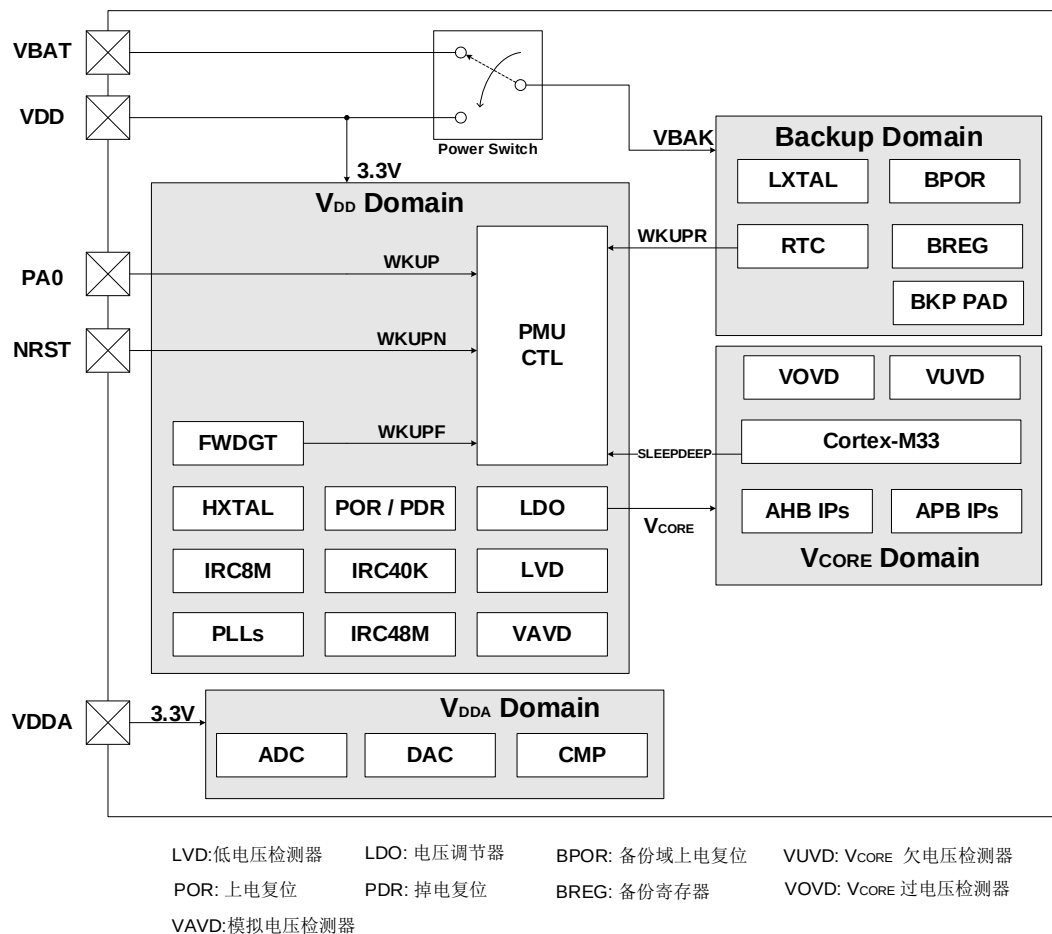
功耗设计是本系列产品比较注重的问题之一。电源管理单元提供了三种省电模式，包括睡眠模式，深度睡眠模式和待机模式。这些模式能减少电源能耗，且使得应用程序可以在CPU运行时间要求、速度和功耗的相互冲突中获得最佳折衷。如[图6-1. 电源域概览](#)所示，本系列设备有三个电源域，包括V_{DD}/V_{DDA}域，VCORE域和备份域。V_{DD}/V_{DDA}域由电源直接供电。在V_{DD}/V_{DDA}域中嵌入了一个LDO，用来为VCORE域供电。在备份域中有一个电源切换器，当V_{DD}电源关闭时，电源切换器可以将备份域的电源切换到VBAT引脚，此时备份域由VBAT引脚（电池）供电。

6.2. 主要特征

- 三个电源域：备份域、V_{DD}/V_{DDA}域和VCORE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VCORE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VBAT（电池）为备份域供电；
- LDO输出电压用于节约能耗；
- 供电监控：POR/PDR监控、LVD监控、VAVD监控、VOVD监控、VUVD监控；
- 低驱动模式用于在深入睡眠模式下超低功耗。

6.3. 功能说明

图6-1. 电源域概览



6.3.1. 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BKP 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V_{DD}/V_{DDA}域掉电复位电路控制的。对于没有外部电池的应用，建议将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128分频，或AHB时钟10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®-M33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VDD供电（VBAK连接至VDD）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VBAT电源供电时（VBAK连接至VBAT），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）。

6.3.2. VDD / VDDA 电源域

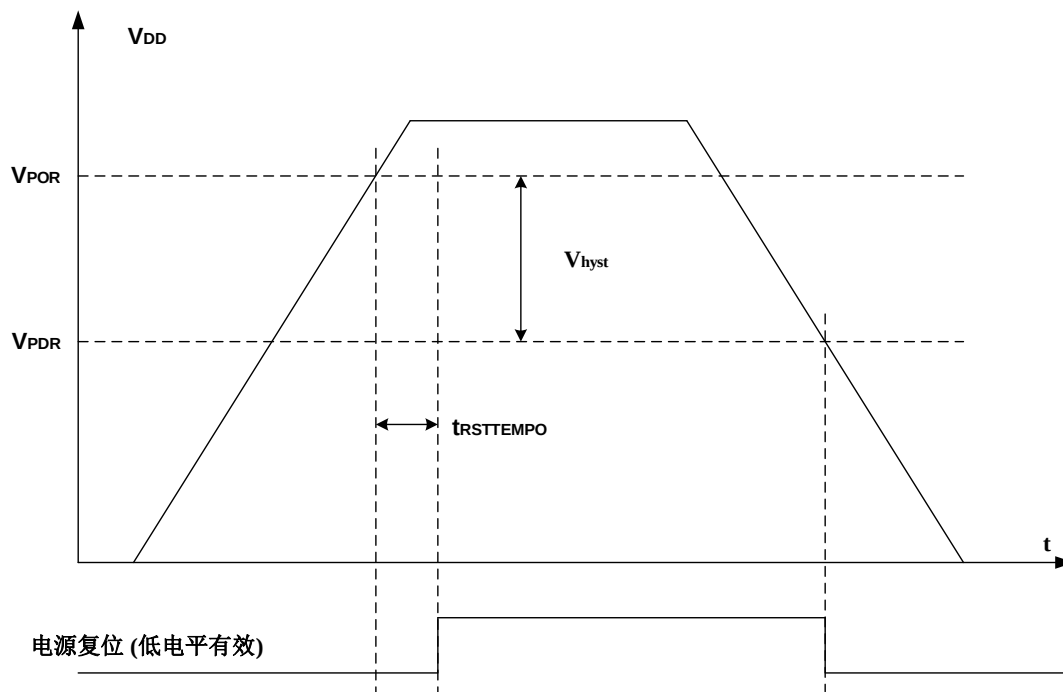
VDD / VDDA 域包括 VDD 域和 VDDA 域两部分。VDD 域包括 HXTAL（高速外部晶体振荡器）、LDO（电压调节器）、LVD（低电压检测器）、POR / PDR（上电/掉电复位）、FWDGT（独立看门狗定时器）、IRC8M（内部 8M RC 振荡器）、IRC48M（内部 48M RC 振荡器）、VAVD(模拟电压检测)、IRC40K（内部 40KHz RC 振荡器）和 PLLs（锁相环）和除 PC13、PC14 和 PC15 之外的所有 PAD 等。VDDA 域包括 ADC / DAC（AD / DA 转换器）、CMP(比较器)等。

VDD 域

为 VCORE 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）。

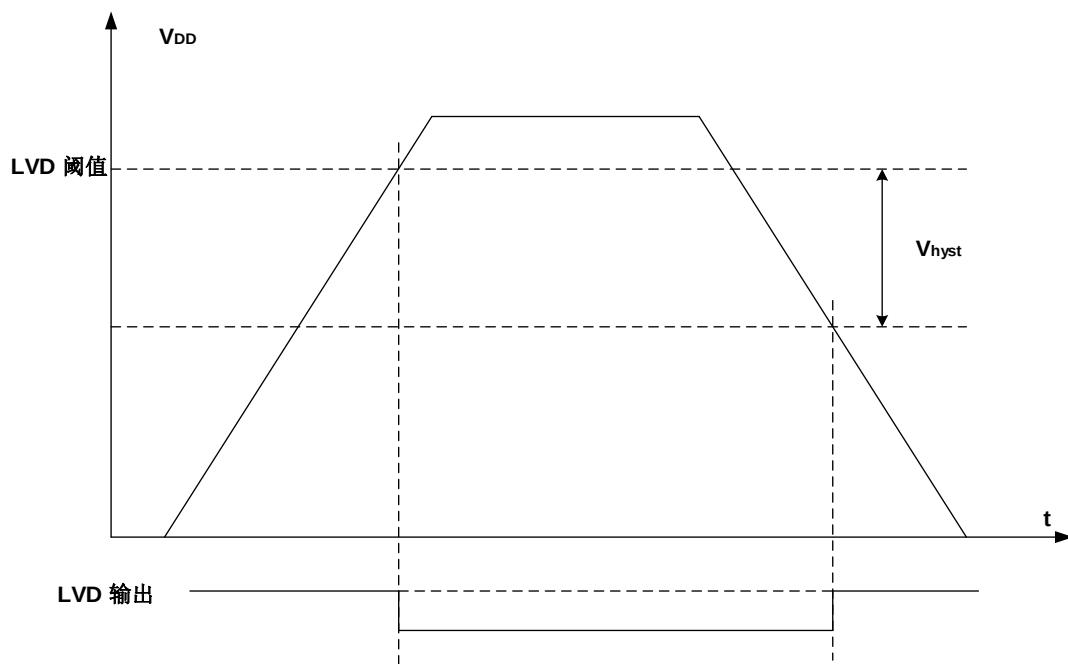
POR / PDR（上电/掉电复位）电路检测 VDD 并在电压低于特定阈值时产生电源复位信号复位除备份域之外的整个芯片。[图6-2. 上电/掉电复位波形图](#)显示了供电电压和电源复位信号之间的关系。V_{POR} 表示上电复位的阈值电压，V_{PDR} 表示掉电复位的阈值电压，V_{hyst} 表示迟滞电压，具体数值可参考值 datasheet。

图6-2. 上电/掉电复位波形图



LVD 的功能是检测 V_{DD} 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_CTL0)中的 LVDT[2:0]位进行配置。LVD 通过 LVDEN 置位使能，位于电源状态寄存器(PMU_CS)中的 LVDF 位表示低电压事件是否出现，该事件连接至 EXTI 的第 16 线，用户可以通过配置 EXTI 的第 16 线产生相应的中断。[图 6-3. LVD 阈值波形图](#)显示了 V_{DD} 供电电压和 LVD 输出信号的关系。迟滞电压 V_{hyst} 值参考 datasheet。

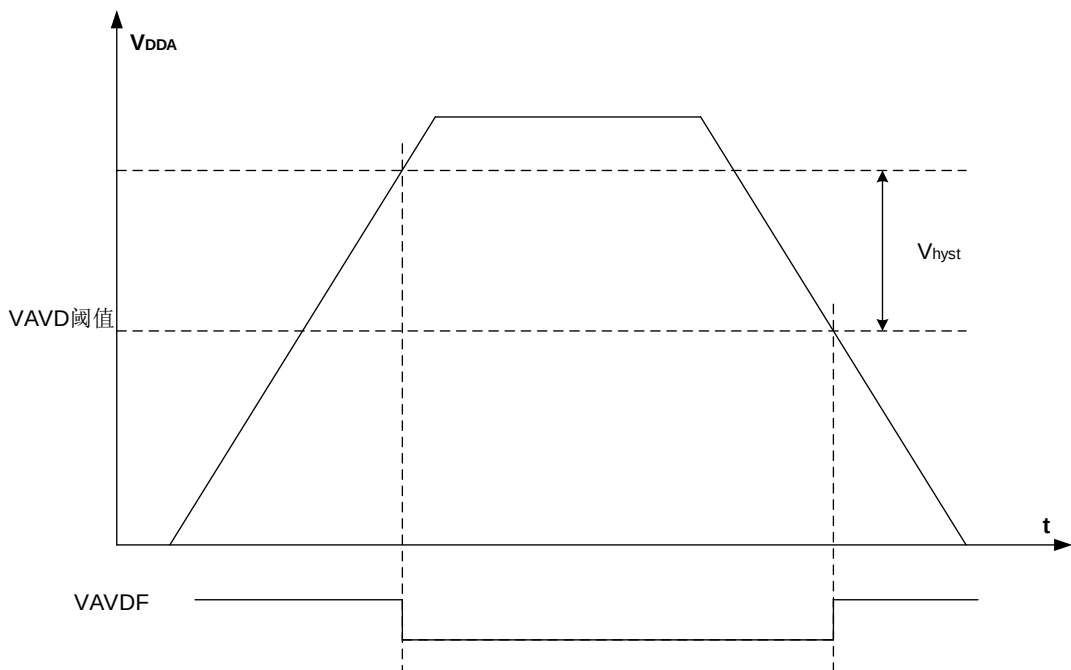
图6-3. LVD阈值波形图



VDDA 域

V_{DD} 模拟电压检测器（VAVD）用于检测 V_{D_{DA}} 电源电压是否低于电源控制寄存器（PMU_CTL0）中 VAVDVC[1:0]位域选择的编程阈值。通过置位 VAVDEN 位能够使能 VAVD，PMU_CS 寄存器中的 VAVDF 位指示 V_{D_{DA}} 高于或低于指定的 VAVD 阈值。[图 6-4. VAVD 阈值监测波形图](#)显示了 VAVD 门限与 VAVDF 之间的关系。迟滞电压 V_{hyst} 值参考 datasheet。

图 6-4. VAVD 阈值监测波形图



一般来说，数字电路由 VDD 供电，而大多数的模拟电路由 VDDA 供电。为了提高 ADC 和 DAC 的转换精度，为 VDDA 独立供电可使模拟电路达到更好的特性。为避免噪声，VDDA 通过外部滤波电路连接至 VDD，相应的 VSSA 通过特定电路连接至 VSS。否则，当 VDD/VDDA 不是同一个电源提供时，在上电和运行过程中 VDD 与 VDDA 差值不超过 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 VREFP/VREFN。根据不同的封装，VREFP 可被连接至 VDDA 引脚，或者外部参考电压，外部参考电压的范围请参考 [表 21-2. ADC 引脚定义](#) 和 [表 22-1. DAC 引脚](#)。VREFN 须被连接至 VSSA 引脚。VREFN 仅存在于不小于 48-pin 封装上，在其他封装里，其内部连接至 VSSA。

6.3.3. VCORE 电源域

VCORE 电源域为 Cortex®-M33 内核逻辑、AHB/APB 外设、备份域和 VDD/VDDA 域的 APB 接口等供电。包括 VUVD（VCORE 欠压检测器）和 VOVD（VCORE 过压检测器）。当 VCORE 电压上电后，POR 将在 VCORE 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。该电源域的电压可以通过 PMU_CTL0 寄存器中的 LDOVS[2:0] 配置。VCORE 电压值请参考数据手册。

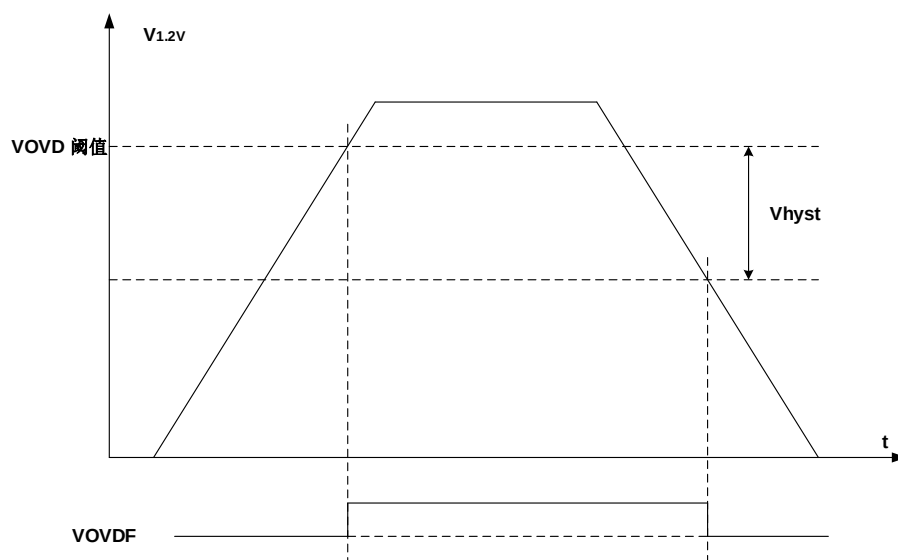
VCORE 电源域电源监测

芯片内部有一个 VCORE 电源域电压监测器，当 VOVDEN 为 0b1，将使能 VCORE 电源域电压检测器。一旦 VCORE 电源域超过电源控制寄存器（PMU_CTL0）中 VOVDVC[1:0] 位选择的编程阈值，在模拟两个触发器同步后，VOVDF 将立即被置位。通过配置 PMU_CTL1 寄存器中的

VOVDO_DNF[7:0]位, 可以使用数字滤波后的 VOVDF。允许抑制峰值的可编程长度为 $1024 * T_{PCLK}$ (PCLK1 的周期) 的 1 到 255。图 6-5. VOVDF 阈值监测波形图显示了 VOVD 门限与 VOVDF 之间的关系, 迟滞电压 V_{hyst} 值参考 datasheet。

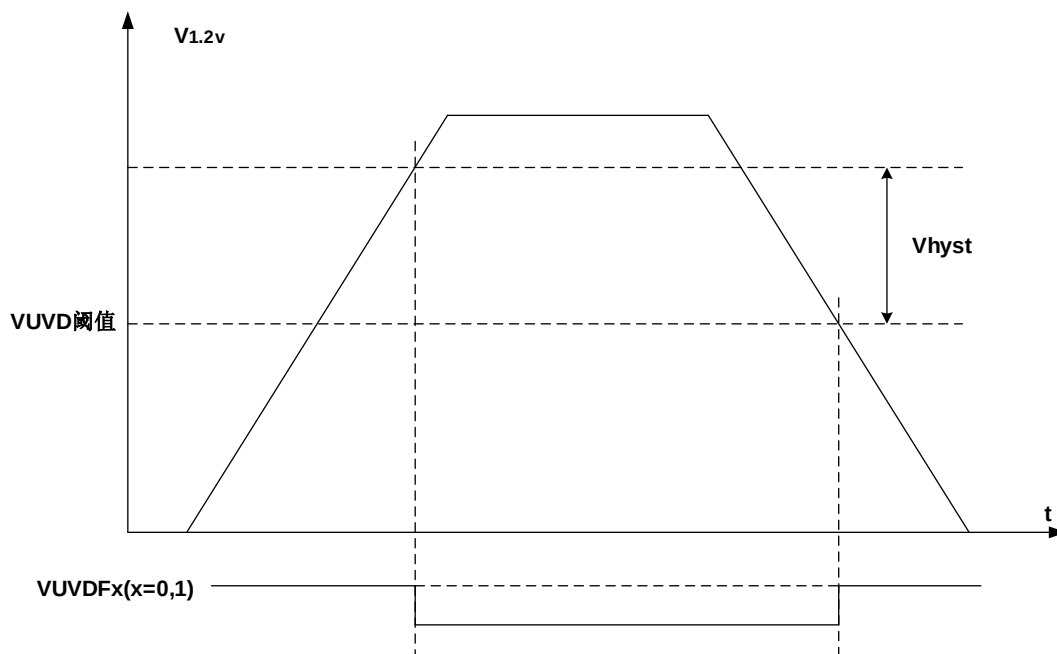
注意: 使能 VOVDF 前, 需要先使能 LVD, 延迟 50us 之后, 再使能 VOVDF。否则 VOVDF 会有误触发信号产生。后面 LVD 可以维持使能状态也可以关闭。

图6-5. VOVDF阈值监测波形图



芯片内部有一个 VCORE 电源域电压监测器, 当 VUVDEN 为 0b1, 将使能 VCORE 电源域电压监测器。一旦 VCORE 电源域低于电源控制寄存器 (PMU_CTL0) 中 VUVDVC[1:0]位选择的编程阈值, 在模拟两个触发器同步后, VUVDFO 将立即被置位。通过配置 PMU_CTL1 寄存器中的 VUVDON_DNF[7:0]位, 可以使用数字滤波后的 VUVDFO。允许抑制峰值的可编程长度为 $1024 * T_{pclk}$ (PCLK1 的周期) 的 1 到 255。图 6-6. VUVD 阈值监测波形图显示了 VOVDF 门限与 VOVDF 之间的关系, 迟滞电压 V_{hyst} 值参考 datasheet。

图6-6. VUVD阈值监测波形图



6.3.4. 省电模式

系统复位或电源复位后，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_CTL0 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 VCORE 电源域。当 PLL 关闭时，LDO 输出低电压模式驱动 VCORE 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®-M33 的 SLEEPING 模式相对应。在睡眠模式下，仅关闭 Cortex®-M33 的时钟。如需进入睡眠模式，只要清除 Cortex®-M33 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®-M33 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®-M33 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®-M33 的 SLEEPDEEP 模式相对应。在深度睡眠模式下，VCORE 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_CTL0 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®-M33 系统控制寄存器的 SLEEPDEEP 位置 1，再清除 PMU_CTL0 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®-M33 技术手册）。刚退出深度睡眠模式时，IRC8M 被选中作为系统时钟。请注意，如果 LDO 工作在低功耗模式，那么唤醒时需额外的延时时间。

在深度睡眠模式下，通过配置 PMU_CTL0 寄存器的 LDEN，LDNP，LDLP，LDOLP 位可以进入低驱动模式。低驱动模式具有低驱动能力，低能耗。

正常驱动/正常功耗：将 PMU_CTL0 寄存器的 LDEN 位配置为 0b00，深度睡眠模式就工作在正常驱动模式下。将 PMU_CTL0 寄存器的 LDOLP 清 0 可以退出低功耗模式。

正常驱动/低功耗：将 PMU_CTL0 寄存器的 LDEN 位配置为 0b00，深度睡眠模式就工作在正常驱动模式下。将 PMU_CTL0 寄存器的 LDOLP 置 1 可以进入低功耗模式。

低驱动/正常功耗：将 PMU_CTL0 寄存器的 LDEN 设置为 0b11，LDNP 置 1 可以进入深度睡眠模式的低驱动模式。将 PMU_CTL0 寄存器的 LDOLP 清 0 可以使 LDO 处于正常功耗模式。

低驱动/低功耗：将 PMU_CTL0 寄存器的 LDEN 设置为 0b11，LDLP 置 1 可以进入深度睡眠模式的低驱动模式。将 PMU_CTL0 寄存器的 LDOLP 清 1 可以使 LDO 处于正常功耗模式。

非低驱动：将 PMU_CTL0 寄存器的 LDEN 设置为 0b00，深度睡眠模式将不会处在低驱动模式。

注意：为了顺利进入深度睡眠模式，所有 EXTI 线上的挂起状态（在 EXTI_PD 寄存器中）和相关外设标志位必须被复位，参考[表 2-3. 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

待机模式是基于 Cortex®-M33 的 SLEEPDEEP 模式实现的。在待机模式下，整个 VCORE 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®-M33 系统控制寄存器的 SLEEPDEEP 位置 1，再将 PMU_CTL0 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 报警，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 VCORE 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®-M33 将从 0x00000000 地

址开始执行指令代码。

表6-1. 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 VCORE 电源域的所有时钟 2、关闭 IRC8M、IRC48M、HXTAL 和 PLL	1、关闭 VCORE 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、正常驱动模式）	开启（正常功耗模式或低功耗模式、正常驱动模式或低驱动模式）	关闭
配置	SLEEPDEEP = 0	SLEEPDEEP = 1 STBMOD = 0	SLEEPDEEP = 1 STBMOD = 1, WURST = 1
进入指令	WFI 或 WFE	WFI 或 WFE	WFI 或 WFE
唤醒	若通过 WFI 进入，则任何中断均可唤醒； 若通过 WFE 进入，则任何事件（或 SEVONPEND = 1 时的中断）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
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 NRST 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。

6.4. PMU 寄存器

PMU 基地址：0x4000 7000

6.4.1. 控制寄存器 0 (PMU_CTL0)

地址偏移：0x00

复位值：0x1600 7000（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
LOCK	保留		VUVDVC[1:0]		VOVDVC[1:0]		VUVDEN	VOVDEN	VAVDVC[1:0]		VAVDEN	LDEN[1:0]		保留	
rw			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	LDOVS[2:0]			LDNP	LDLP	PDRVS	BKPWEN	LVDT[2:0]			LV DEN	STBRST	WURST	STBMOD	LDOLP
	rw			rw	rw	rw	rw	rw			rw	rc_w1	rc_w1	rw	rw

位/位域	名称	描述
31	LOCK	寄存器 lock 0: 其他寄存器和位都不可写 1: 其他寄存器和位都可写
30:29	保留	必须保持复位值。
28:27	VUVDVC[1:0]	V _{CORE} 低压检测器电压等级配置位 这些位由软件置位和清除。 00: 配置V _{CORE} 低压检测器电压等级为1.05V 01: 配置V _{CORE} 低压检测器电压等级为0.95V 1x: 配置V _{CORE} 低压检测器电压等级为0.85V
26:25	VOVDVC[1:0]	V _{CORE} 过压检测器电压等级配置位 这些位由软件置位和清除。 00: 配置V _{CORE} 过压检测器电压等级为1.25V 01: 配置V _{CORE} 过压检测器电压等级为1.30V 10: 配置V _{CORE} 过压检测器电压等级为1.35V 11: 配置V _{CORE} 过压检测器电压等级为1.40V
24	VUVDEN	V _{CORE} 低压检测器使能位 该位由软件置位和清除。 0: 失能V _{CORE} 低压检测器 1: 使能 V _{CORE} 低压检测器
23	VOVDEN	V _{CORE} 过压检测器使能位

		该位由软件置位和清除。 0: 失能V _{CORE} 过压检测器 1: 使能V _{CORE} 过压检测器
22:21	VAVDVC[1:0]	V _{DDA} 模拟电压检测器电压等级配置位 这些位由软件置位和清除。 00: 配置V _{DDA} 模拟电压检测器电压等级为2.3V 01: 配置V _{DDA} 模拟电压检测器电压等级为2.5V 10: 配置V _{DDA} 模拟电压检测器电压等级为2.7V 11: 配置V _{DDA} 模拟电压检测器电压等级为2.9V
20	VAVDEN	V _{DDA} 模拟电压检测器使能位 该位由软件置位和清除。 0: 失能V _{DDA} 模拟电压检测器 1: 使能V _{DDA} 模拟电压检测器
19:18	LDEN[1:0]	深度睡眠模式下，低驱动模式使能 00: 在深度睡眠模式下，禁用低驱动模式 01: 保留 10: 保留 11: 在深度睡眠模式下，使能低驱动模式
17:15	保留	必须保持复位值。
14:12	LDOVS[2:0]	LDO 输出电压选择 在 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 输出默认值。 000: 保留 001: 0.9v（不建议客户使用） 010: 0.95v（不建议客户使用） 011: 1.0v（不建议客户使用） 100: 1.05v（不建议客户使用） 101: 1.1v 110: 1.15v 111: 1.2v 注意： 该位域不支持 0b000 配置，配置为 0b000 将会导致不可预知的错误。
11	LDNP	使用正常功耗 LDO 时，工作在低驱动模式 0: 使用正常功耗 LDO 时，工作在正常驱动模式 1: 使用正常功耗 LDO 且 LDEN 为 11 时，低驱动模式被使能
10	LDLP	使用低功耗 LDO 时，工作在低驱动模式 0: 使用低功耗 LDO 时，工作在正常驱动模式 1: 使用低功耗 LDO 且 LDEN 为 11 时，低驱动模式被使能

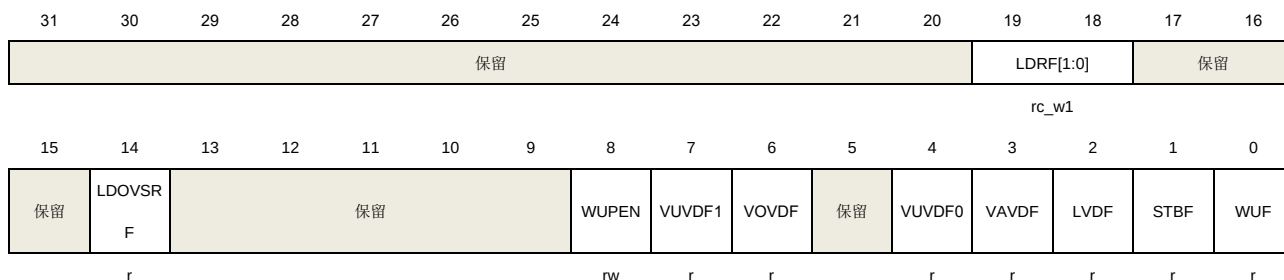
9	PDRVS	PDR 阈值电压选择 0: 2.35V 1: 1.8V
8	BKPWEN	备份域写使能 0: 禁止对备份域寄存器的写访问 1: 允许对备份域寄存器的写访问 复位之后, 任何对备份域寄存器的写访问都将被禁止。如需对备份域寄存器做写访问, 需先将该位置 1。
7:5	LVDT[2:0]	低电压检测器阈值 000: 2.1V 001: 2.3V 010: 2.4V 011: 2.6V 100: 2.7V 101: 2.8V 110: 3.0V 111: 3.1V
4	LVDEN	低电压检测器使能 0: 关闭低电压检测器 1: 开启低电压检测器 注意: 当 SYSCFG_LKCTL 寄存器里的 LVD_LOCK 位被置 1 时, LV DEN 和 LVDT[2:0] 仅可读。
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®-M33 进入 SLEEPDEEP 模式时, 系统进入深度睡眠模式 1: 当 Cortex®-M33 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

6.4.2. 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



位/位域	名称	描述
31:20	保留	必须保持复位值。
19:18	LDRF[1:0]	低驱动模式就绪标志 在深度睡眠模式下，且 LDO 处于低驱动模式，这些位由硬件设置。软件对这些位写 11 可以清 0。 00: 深度睡眠模式下，普通驱动模式 01: 保留 10: 保留 11: 深度睡眠模式下，低驱动模式
17:15	保留	必须保持复位值。
14	LDOVSRF	LDO 电压选择就绪标志 0: LDO 电压选择未就绪 1: LDO 电压选择就绪
13:9	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 在进入待机模式之前置 1，WKUP 引脚的上升沿会将系统从待机模式唤醒。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	VUVDF1	数字滤波后V_{CORE}低电压检测器标志 位由硬件设置和清除，仅当VUVDEN使能时有效 0: V_{CORE}大于VUVD阈值。 1: V_{CORE}小于等于 VUVD 阈值。

6	VOVDF	数字滤波后V_{CORE}过电压检测器标志 位由硬件设置和清除，仅当VOVDEN使能时有效 0: V_{CORE}小于VOVDVC[1:0]阈值。 1: V_{CORE} 大于等于 VOVDVC[1:0]阈值。
5	保留	必须保持复位值。
4	VUVDF0	数字滤波后V_{CORE}低电压检测器标志 位由硬件设置和清除，仅当VUVDEN使能时有效 0: V_{CORE}大于VUVD阈值。 1: V_{CORE} 小于等于 VUVD 阈值。
3	VAVDF	V_{DDA}模拟电压检测器输出标志 位由硬件设置和清除，仅当VAVDEN使能时有效 0: V_{DDA}大于等于由VAVDVC位配置的VAVD阈值。 1: V_{DDA} 小于由 VAVDVC 位配置的 VAVD 阈值。
2	LVDF	低电压状态标志 0: 低电压事件没出现（V_{DD} 高于设定的 LVD 阈值） 1: 低电压事件出现（V_{DD} 等于或低于 LVD 阈值） 注意：LVD 功能在待机模式被禁用。
1	STBF	待机标志 0: 设备没进入过待机模式 1: 设备曾进入过待机模式 该位只能由 POR / PDR 或通过设置 PMU_CTL0 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_CTL0 寄存器的 WURST 位来清零。

6.4.3. 控制寄存器 1（PMU_CTL1）

地址偏移：0x08

复位值：0x0000 0000

该寄存器可以按字（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

VUVDO_DNF[7:0]	VOVDO_DNF[7:0]
rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	VUVDO_DNF[7:0]	VUVD 模拟输出数字噪声滤波器 这些位用于配置 VUVD 模拟输出上的数字噪声滤波器，数字滤波器将滤波峰值的长度高达 $VUVDO\_DNF[7:0] * 1024 * T_{PCLK1}$ 0: 关闭数字滤波器 1: 开启数字滤波器，滤波峰值长度高达 $1024 * T_{PCLK1}$... 255: 开启数字滤波器，滤波峰值长度高达 $255 * 1024 * T_{PCLK1}$
7:0	VOVDO_DNF[7:0]	VOVD 模拟输出数字噪声滤波器 这些位用于配置 VOVd 模拟输出上的数字噪声滤波器，数字滤波器将滤波峰值的长度高达 $VOVDO\_DNF[7:0] * 1024 * T_{PCLK1}$ 0: 关闭数字滤波器 1: 开启数字滤波器，滤波峰值长度高达 $1024 * T_{PCLK1}$... 255: 开启数字滤波器，滤波峰值长度高达 $255 * 1024 * T_{PCLK1}$

7. 备份寄存器（BKP）

7.1. 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。

7.2. 主要特点

- 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）可实现侵入检测中断或事件的控制。

7.3. 功能描述

7.3.1. 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 $1000000/2^{20}$ ppm 的比例减慢 RTC 时钟。

7.3.2. 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。

7.4. BKP 寄存器

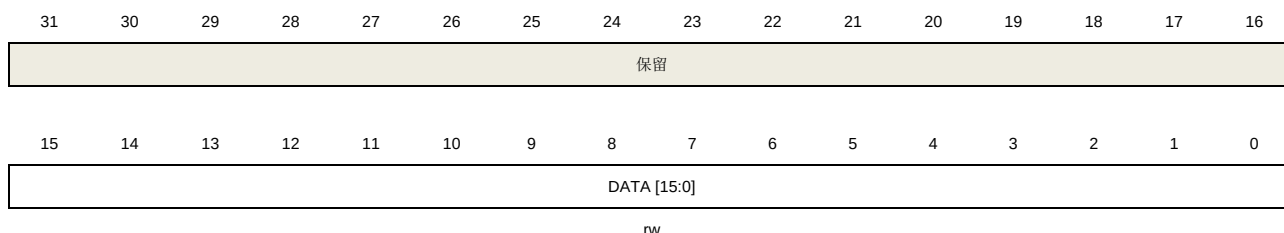
BKP 基地址：0x4000 6C00

7.4.1. 备份数据寄存器 (BKP_DATAx) (x= 0..41)

地址偏移：0x04 + 0x04*x(x=0..9), 0x40 + 0x04*(x-10)(x=10..41)

复位值：0x0000 0000

该寄存器可以按半字（16 位）或字（32 位）访问。



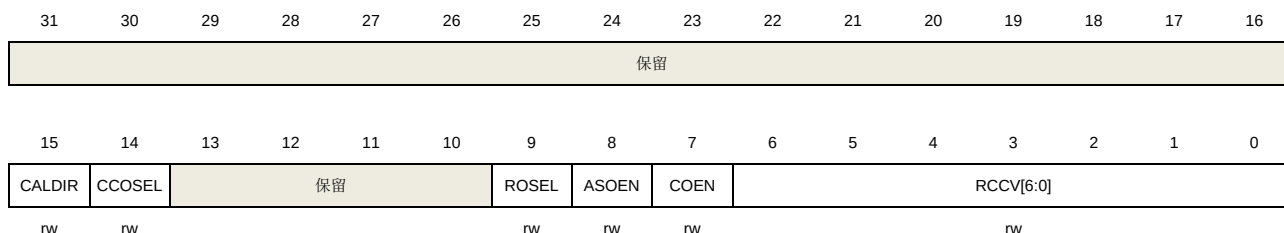
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	DATA[15:0]	备份数据 这些位用来存储一般用户数据。即使从待机模式唤醒或系统复位或电源复位后，BKP_DATAx 寄存器的内容仍旧不会丢失。

7.4.2. RTC 信号输出控制寄存器 (BKP_OCTL)

地址偏移：0x2C

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
31:16	保留	必须保持复位值。
15	CALDIR	RTC 时钟校方向 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。

7.4.3. 侵入引脚控制寄存器 (BKP_TPCTL)

地址偏移: 0x30

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													TPAL	TPEN	

rw rw

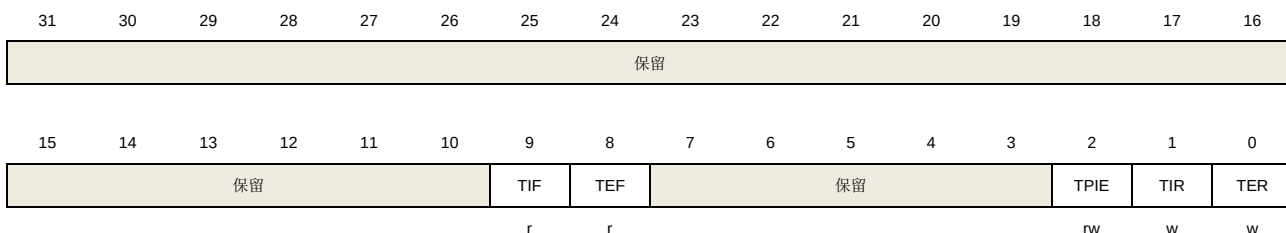
位/位域	名称	描述
31:2	保留	必须保持复位值。
1	TPAL	TAMPER 引脚有效电平 0: TAMPER 引脚高电平有效 1: TAMPER 引脚低电平有效
0	TPEN	TAMPER 引脚使能 0: TAMPER 引脚作为 GPIO 口使用 1: TAMPER 引脚可实现备份复位功能。TAMPER 引脚上的有效电平将复位 BKP_DATAx 寄存器中所有数据。

7.4.4. 侵入控制状态寄存器 (BKP_TPCS)

地址偏移: 0x34

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
31:10	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。

8. 复位和时钟单元（RCU）

8.1. 复位控制单元（RCTL）

8.1.1. 简介

GD32F50x 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，其复位除了备份域的所有系统。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。

8.1.2. 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 V_{CORE} 域电压时，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

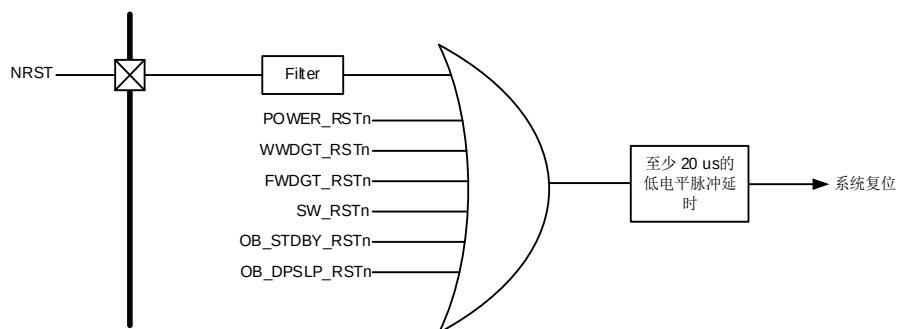
当发生以下任一事件时，产生一个系统复位：

- 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®-M33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将产生复位（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 8-1. 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）。

8.2. 时钟控制单元（CCTL）

8.2.1. 简介

时钟控制单元提供了一系列频率的时钟功能，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）、两个锁相环（PLL）、一个 HXTAL 时钟监视器、一个 LXTAL 时钟监视器、时钟频率监视器、时钟预分频器、时钟多路复用器和时钟门控电路。

AHB、APB 和 Cortex®-M33 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 252MHz（适用于 GD32F503xx）或 280MHz（适用于 GD32F505xx）。

图 8-2. 时钟树 (适用于 GD32F503xx)

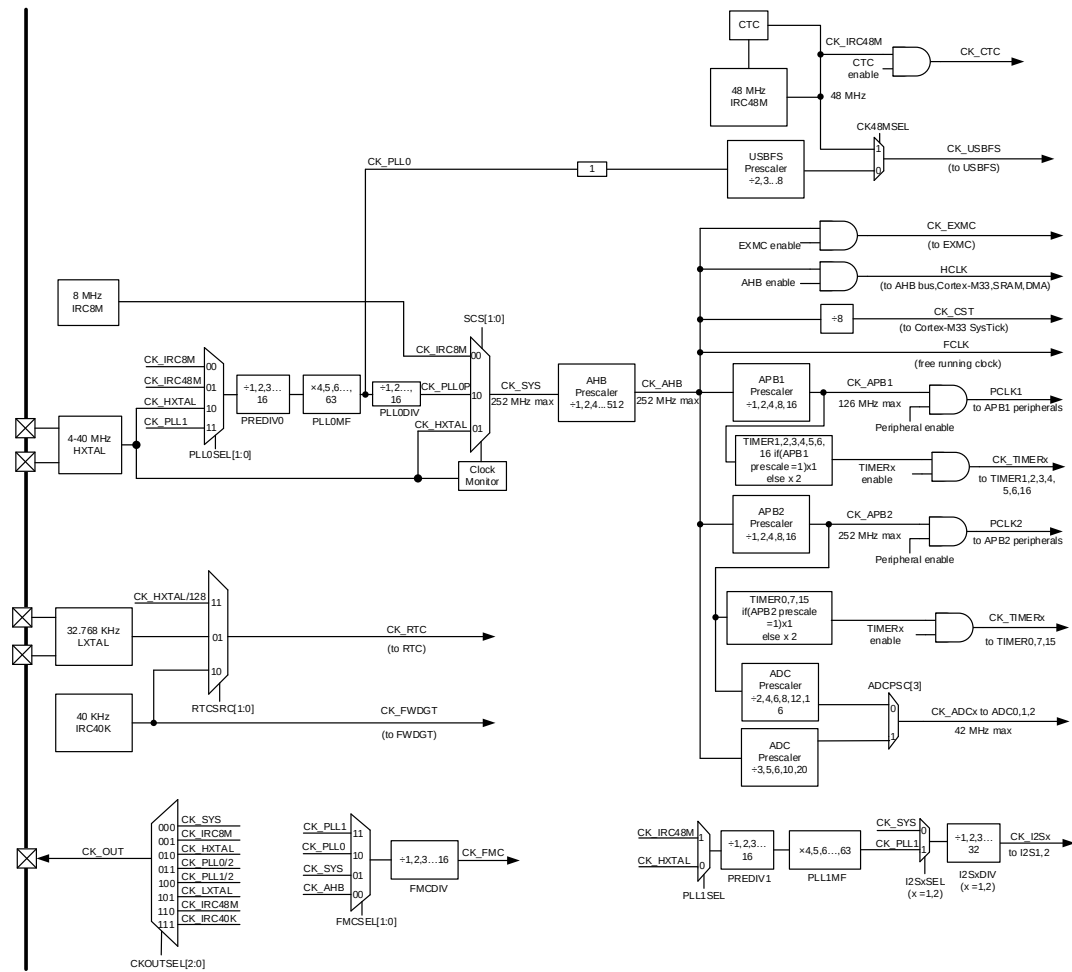
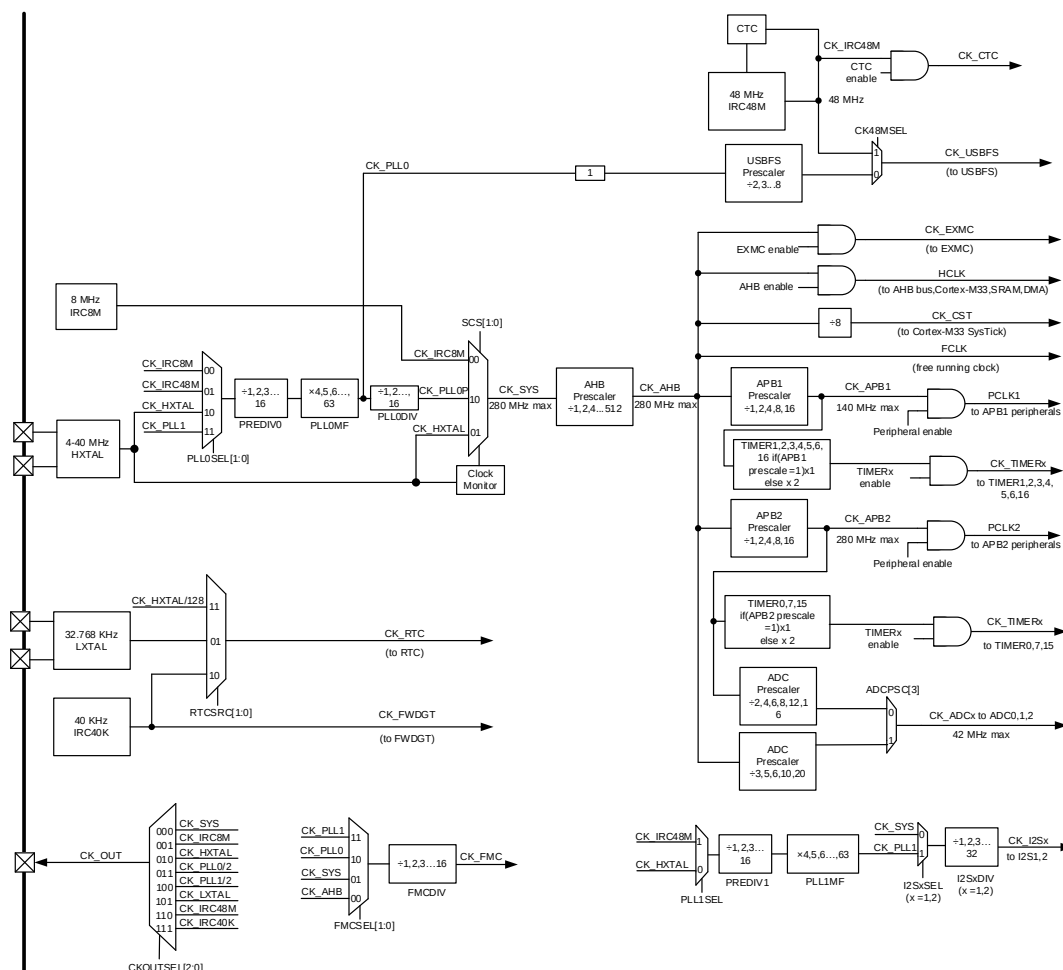


图 8-3. 时钟树（适用于 GD32F505xx）



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 252MHz、252MHz、126MHz（适用于 GD32F503xx）或 280MHz、280MHz、140MHz（适用于 GD32F505xx）。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、6、10、20 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 的分频系数为 1，则 TIMER 时钟为 CK_APBx；如果 APBx 的分频系数不为 1，则 TIMER 时钟为 CK_APBx 的两倍。

USBFS 的时钟由 CK48M 时钟提供。通过配置 RCU_ADDCTL 寄存器的 CK48MSEL 及 PLL48MSEL 位可以选择 CK_PLL0 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 CK_PLL1 提供，通过配置 RCU_CFG1 寄存器的 I2SxSEL（x = 1, 2）

来选择。

FMC 接口时钟可通过 RCU_ADDCTL 寄存器中的 FMCSEL 位域选择为 CK_AHB、CK_SYS、CK_PLL0 和 CK_PLL1 中的一个，并通过 FMCDIV 分频后提供。

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。

8.2.2. 主要特征

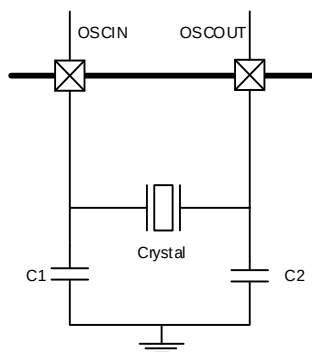
- 4到40MHz外部高速晶体振荡器（HXTAL）；
- 内部8MHz RC振荡器（IRC8M）；
- 内部48MHz RC振荡器（IRC48M）；
- 32,768 Hz外部低速晶体振荡器（LXTAL）；
- 内部40KHz RC振荡器（IRC40K）；
- PLL0时钟源可选HXTAL、IRC8M或IRC48M；
- HXTAL时钟监视器；
- LXTAL时钟监视器；
- 时钟频率监视器。

8.2.3. 功能说明

外部高速晶体振荡时钟（HXTAL）

4 到 40M 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 8-4. HXTAL 时钟源

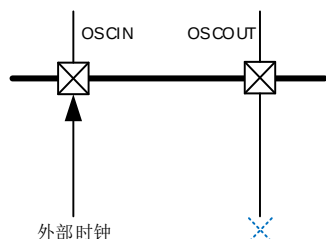


HXTAL 晶体振荡器可以通过设置控制寄存器 RCU_CTL 的 HXTALEN 位来启动或关闭，在控制寄存器 RCU_CTL 中的 HXTALSTB 位用来指示外部高速振荡器是否已稳定。在启动时，直到这一位

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。旁路输入时，信号接至 OSCIN，OSCOUT 保持悬空状态，如 [图 8-5. 旁路模式下 HXTAL 时钟源](#) 所示。此时，CK_HXTAL 等于驱动 OSCIN 管脚的外部时钟。

图 8-5. 旁路模式下 HXTAL 时钟源



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 PLL0 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 PLL0 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。

锁相环（PLL）

时钟部分有两个锁相环，PLL0，PLL1。

PLL0 可以通过设置 RCU_CTL 寄存器中的 PLL0EN 位被启动和关闭。RCU_CTL 寄存器中的 PLL0STB 位用来指示 PLL0 时钟是否稳定。如果 RCU_INT 寄存器中的相应中断使能位 PLL0STBIE 被置‘1’，在 PLL0 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。

当进入 Deepsleep/Standby 模式或者 HXTAL 监视器检测到时钟阻塞时（HXTAL 作为锁相环的输入时钟），两个 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度的时钟源。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。

工厂会校准 IRC40K 时钟频率的精度。

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_PLL0P。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 PLL0 间接作为系统时钟时，它将不能被停止。

HXTAL 时钟监视器 (HCKM)

设置控制寄存器 RCU_CTL 中的 HXTAL 时钟监视使能位 HCKMEN, 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 HCKMIF 将被置‘1’, 产生 HXTAL 故障事件。这个故障引发的中断和 Cortex®-M33 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。

LXTAL 时钟监视器 (LCKM)

设置时钟控制寄存器 RCU_BDCTL 中的 LXTAL 时钟监视使能位 LCKMEN, LXTAL 可以使能时钟监视功能。该功能必须在 LXTAL 启动延迟完毕后使能。

LXTAL 上的时钟监视器在除 VBAT 以外的所有模式下工作。如果在外部 32 kHz 振荡器上检测到故障, 可以向 CPU 发送中断。

然后, 软件必须禁用 LCKMEN 位, 停止有缺陷的 32 kHz 振荡器, 并更改 RTC 时钟源, 或采取任何必要的措施来保护应用程序。

当 LCKMEN 启用时, 一个 4 位加一个计数器将在 IRC40K 域工作。如果 LXTAL 时钟卡在 0/1 错误或减慢约 20KHz, 计数器将溢出。将发现 LXTAL 时钟故障。一旦监测到 LXTAL 故障, 中断寄存器 RCU_INT 中的 LXTAL 时钟阻塞中断标志位 LCKMIF 将被置‘1’, 产生 LXTAL 故障事件。

时钟频率监视器 (CKFM)

该时钟频率监测器可以使用 IRC48M 对 IRC8M、HXTAL、PLL0P 和 PLL1 的时钟频率范围进行监控。IRC8M 和 HXTAL 采用 1000 个 IRC48M 时钟周期作为监控窗口。对于 PLL0P 和 PLL1, 采用 100 IRC48M 时钟周期作为监控窗口。用户可以通过配置 RCU_CKFMCFGx (x = 0, 1, 2, 3) 寄存器来配置时钟频率的监控范围。如果使能了相应的中断, 并设置了时钟频率失效标志, 则将发生中断。

当 IRC48M 时钟被禁用或丢失时, 时钟频率监控将失效。

时钟输出功能

通过设置时钟配置寄存器 0 (RCU_CFG0) 中的 CK_OUT 时钟源选择位域 CKOUTSEL 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 8-1. 时钟输出的时钟源选择

时钟输出的时钟源选择位域	时钟源
000	CK_SYS
001	CK_IRC8M
010	CK_HXTAL

时钟输出的时钟源选择位域	时钟源
011	CK_PLL0/2
100	CK_PLL1/2
101	CK_LXTAL
110	CK_IRC48M
111	CK_IRC40K

电压控制

深度睡眠模式电压寄存器（RCU_DSV）中的 DSLPVS[2:0]位域可以控制 V_{CORE} 域在深度睡眠模式下的电压。

表 8-2. 深度睡眠模式下 V_{CORE} 域电压选择

DSLPVS[2:0]	深度睡眠模式电压（V）
000	默认值
001	默认值-0.05
010	默认值-0.1
011	默认值-0.15
100	默认值-0.2
101	默认值-0.25
110	默认值-0.3
111	默认值-0.35

8.3. RCU 寄存器

RCU 基地址：0x4002 1000

8.3.1. 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				PLL1STB	PLL1EN	PLL0STB	PLL0EN	保留				HCKMEN	HXTALB PS	HXTALST B	HXTALE N
				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
31:28	保留	必须保持复位值。
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	PLL0STB	PLL0 时钟稳定标志位 硬件置 1 来表示 PLL0 输出时钟是否稳定待用 0: PLL0 未稳定 1: PLL0 已稳定
24	PLL0EN	PLL0 使能 软件置位或复位，当 PLL0 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: PLL0 被关闭

		1: PLL0 被打开
23:20	保留	必须保持复位值。
19	HCKMEN	HXTAL 时钟监视器使能 0: 禁止高速 4 ~ 40 MHz 晶体振荡器 (HXTAL) 时钟监视器 1: 使能高速 4 ~ 40 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 HCKMIF 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 ~ 40 MHz 晶体振荡器被关闭 1: 高速 4 ~ 40 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

8.3.2. 时钟配置寄存器 0 (RCU_CFG0)

地址偏移: 0x04

复位值: 0x0010 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
USBFSP SC[2]	PLL0MF[5:4]		ADCPSC[2]	保留	CKOUTSEL[2:0]			USBFSPSC[1:0]		PLL0MF[3:0]			PREDIV0 _LSB	保留	
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]			AHBPSC[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
30:29	PLL0MF[5:4]	PLL0MF 的第 5 位和第 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	保留	必须保持复位值。
26:24	CKOUTSEL[2:0]	CKOUT 时钟源选择 由软件置位或清零 000: 选择系统时钟 CK_SYS 001: 选择内部 8MHz RC 振荡器时钟 010: 选择高速晶体振荡器时钟 (HXTAL) 011: 选择 (CK_PLL0 / 2) 时钟 100: 选择 (CK_PLL1 / 2) 时钟 101: 选择低速晶体振荡器时钟 (LXTAL) 110: 选择内部 48MHz RC 振荡器时钟 111: 选择内部 40KHz RC 振荡器时钟
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_PLL0 / 3 001: CK_USBFS = CK_PLL0 / 2 010: CK_USBFS = CK_PLL0 / 5 011: CK_USBFS = CK_PLL0 / 4 100: CK_USBFS = CK_PLL0 / 6 101: CK_USBFS = CK_PLL0 / 7 11x: CK_USBFS = CK_PLL0 / 8
21:18	PLL0MF[3:0]	PLL0 时钟倍频因子 与寄存器 RCU_CFG0 的 29,30 位共同构成倍频因子，由软件置位或清零 注意：PLL0 输出时钟频率不能超过 252MHz（适用于 GD32F503xx）；PLL0 输出时钟频率不能超过 280MHz（适用于 GD32F505xx） 0000xx: 保留 000100:（PLL 时钟源 x 4） 000101:（PLL 时钟源 x 5） 000110:（PLL 时钟源 x 6） ... 111111:（PLL 时钟源 x 63）
17	PREDIV0_LSB	PREDIV0 分频因子的最低位 与寄存器 RCU_CFG1 位 PREDIV0 第 0 位相同，通过寄存器 RCU_CFG1 来改变 PREDIV0 的值，此位也会一同改。当 PREDIV0 的第 1 到 3 位未修改时，此位决定 PREDIV0 的输入时钟是否二分频。
16	保留	必须保持复位值。
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 1000: CK_ADC = CK_AHB / 3 1100: CK_ADC = CK_AHB / 5 1x01: CK_ADC = CK_AHB / 6 1x10: CK_ADC = CK_AHB / 10 1x11: CK_ADC = CK_AHB / 20

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_PLL0P 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_PLL0P 时钟作为 CK_SYS 时钟源

11: 保留

8.3.3. 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						LCKMIF	LCKMIC	HCKMIC	保留	PLL1 STBIC	PLL0 STBIC	HXTAL STBIC	IRC8M STBIC	LXTAL STBIC	IRC40K STBIC
						r	w	w		w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCKMIE	保留	PLL1 STBIE	PLL0 STBIE	HXTAL STBIE	IRC8M STBIE	LXTAL STBIE	IRC40K STBIE	HCKMIF	保留	PLL1 STBIF	PLL0 STBIF	HXTAL STBIF	IRC8M STBIF	LXTAL STBIF	IRC40K STBIF
rw		rw	rw	rw	rw	rw	rw	r		r	r	r	r	r	r

位/位域	名称	描述
31:26	保留	必须保持复位值。
25	LCKMIF	LXTAL 时钟阻塞中断标志位 当 LXTAL 时钟被阻塞时由硬件置位 软件置位 LCKMIC 位时清除该位 0: 时钟正常运行 1: LXTAL 时钟阻塞
24	LCKMIC	LXTAL 时钟阻塞中断清零 软件写 1 复位 LCKMIF 标志位 0: 不复位 LCKMIF 标志位 1: 复位 LCKMIF 标志位
23	HCKMIC	HXTAL 时钟阻塞中断清零 软件写 1 复位 HCKMIF 标志位 0: 不复位 HCKMIF 标志位 1: 复位 HCKMIF 标志位
22	保留	必须保持复位值。
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	PLL0STBIC	PLL0 时钟稳定中断清零

		软件写 1 复位 PLL0STBIF 标志位 0: 不复位 PLL0STBIF 标志位 1: 复位 PLL0STBIF 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	LCKMIE	LXTAL 时钟阻塞中断使能 软件置位和复位来使能/禁止 LXTAL 时钟阻塞中断 0: 禁止 LXTAL 时钟阻塞中断 1: 使能 LXTAL 时钟阻塞中断
14	保留	必须保持复位值。
13	PLL1STBIE	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 0: 禁止 PLL1 时钟稳定中断 1: 使能 PLL1 时钟稳定中断
12	PLL0STBIE	PLL0 时钟稳定中断使能 软件置位和复位来使能/禁止 PLL0 时钟稳定中断 0: 禁止 PLL0 时钟稳定中断 1: 使能 PLL0 时钟稳定中断
11	HXTALSTBIE	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 0: 禁止 HXTAL 时钟稳定中断 1: 使能 HXTAL 时钟稳定中断

10	IRC8MSTBIE	IRC8M 时钟稳定中断使能 软件置位和复位来使能/禁止 IRC8M 时钟稳定中断 0: 禁止 IRC8M 时钟稳定中断 1: 使能 IRC8M 时钟稳定中断
9	LXTALSTBIE	LXTAL 时钟稳定中断使能 软件置位和复位来使能/禁止 LXTAL 时钟稳定中断 0: 禁止 LXTAL 时钟稳定中断 1: 使能 LXTAL 时钟稳定中断
8	IRC40KSTBIE	IRC40K 时钟稳定中断使能 软件置位和复位来使能/禁止 IRC40K 时钟稳定中断 0: 禁止 IRC40K 时钟稳定中断 1: 使能 IRC40K 时钟稳定中断
7	HCKMIF	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 HCKMIC 位时清除该位 0: 时钟正常运行 1: HXTAL 时钟阻塞
6	保留	必须保持复位值。
5	PLL1STBIF	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 0: 无 PLL1 时钟稳定中断产生 1: 产生 PLL1 时钟稳定中断
4	PLL0STBIF	PLL0 时钟稳定中断标志位 当 PLL0 时钟稳定且 PLL0STBIE 位被置 1 时由硬件置 1 软件置位 PLL0STBIC 位时清除该位 0: 无 PLL0 时钟稳定中断产生 1: 产生 PLL0 时钟稳定中断
3	HXTALSTBIF	HXTAL 时钟稳定中断标志位 当高速 4~40 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

8.3.4. 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				CAN1RS	CAN0RS	SYSCFG	TRIGSEL	保留				TIMER15	保留		
				T	T	RST	RST					RST			
				RW	RW	RW	RW					RW			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC2RS	USART0	TIMER7R	SPI0RST	TIMER0R	ADC1RS	ADC0RS	保留		PERST	PDRST	PCRST	PBRST	PARST	保留	AFRST
T	RST	ST		ST	T	T									
RW	RW	RW	RW	RW	RW	RW			RW	RW	RW	RW	RW		RW

位/位域	名称	描述
31:28	保留	必须保持复位值。
27	CAN1RST	CAN1 复位 由软件置位或复位 0: 无作用 1: 复位 CAN1
26	CAN0RST	CAN0 复位 由软件置位或复位 0: 无作用 1: 复位 CAN0
25	SYSCFGRST	SYSCFG 复位 由软件置位或复位

		0: 无作用 1: 复位 SYSCFG
24	TRIGSELRST	TRIGSEL 复位 由软件置位或复位 0: 无作用 1: 复位 TRIGSEL
23:21	保留	必须保持复位值。
20	TIMER15RST	TIMER15 复位 由软件置位或复位 0: 无作用 1: 复位 TIMER15
19:16	保留	必须保持复位值。
15	ADC2RST	ADC2 复位 由软件置位或复位 0: 无作用 1: 复位 ADC2
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

8.3.5. 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	保留	保留	保留	保留	CMRST	I2C1RST	I2C0RST	UART4RST	UART3RST	USART2RST	USART1RST	保留
	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
SPI2RST	SPI1RST	保留	保留	WWDGT	保留	保留	保留	保留	TIMER16RST	TIMER6RST	TIMER5RST	TIMER4RST	TIMER3RST	TIMER2RST	TIMER1RST
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	DACRST	DAC 复位 由软件置位或复位 0: 无作用 1: 复位 DAC
28	PMURST	PMU 复位 由软件置位或复位 0: 无作用 1: 复位 PMU
27	BKPIRST	BKPI 复位 由软件置位或复位 0: 无作用 1: 复位 BKP
26:24	保留	必须保持复位值。
23	CMRST	CMP 复位 由软件置位或复位 0: 无作用 1: 复位 CMP
22	I2C1RST	I2C1 复位 由软件置位或复位 0: 无作用 1: 复位 I2C1
21	I2C0RST	I2C0 复位 由软件置位或复位 0: 无作用 1: 复位 I2C0
20	UART4RST	UART4 复位

		由软件置位或复位 0: 无作用 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
10:7	保留	必须保持复位值。
6	TIMER16RST	TIMER16 复位 由软件置位或复位 0: 无作用 1: 复位 TIMER16
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

8.3.6. 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												DMAMUX EN	TRNGEN	HAUEN	CAUEN
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			USBFS EN	保留			EXMCEN	保留	CRCEN	保留	FMCSPE EN	保留	SRAMSP EN	DMA1EN	DMA0EN
			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	DMAMUXEN	DMAMUX 时钟使能 由软件置位或复位 0: 关闭 DMAMUX 时钟 1: 开启 DMAMUX 时钟
18	TRNGEN	TRNG 时钟使能 由软件置位或复位 0: 关闭 TRNG 时钟 1: 开启 TRNG 时钟
17	HAUEN	HAU 时钟使能 由软件置位或复位 0: 关闭 HAU 时钟 1: 开启 HAU 时钟
16	CAUEN	CAU 时钟使能 由软件置位或复位 0: 关闭 CAU 时钟 1: 开启 CAU 时钟
15:13	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	FMCSPEEN	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

8.3.7. 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				CAN1EN	CAN0EN	SYSCFG EN	TRIGSEL EN	保留				TIMER15 EN	保留		
				rW	rW	rW	rW					rW			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC2EN	USART0 EN	TIMER7E N	SPI0EN	TIMER0E N	ADC1EN	ADC0EN	保留	PEEN	PDEN	PCEN	PBEN	PAEN	保留	AFEN	
rW	rW	rW	rW	rW	rW	rW		rW	rW	rW	rW	rW	rW	rW	

位/位域	名称	描述
31:28	保留	必须保持复位值。
27	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
26	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
25	SYSCFGEN	SYSCFG 时钟使能 由软件置位或复位 0: 关闭 SYSCFG 时钟 1: 开启 SYSCFG 时钟
24	TRIGSELEN	TRIGSEL 时钟使能 由软件置位或复位 0: 关闭 TRIGSEL 时钟 1: 开启 TRIGSEL 时钟
23:21	保留	必须保持复位值。
20	TIMER15EN	TIMER15 时钟使能 由软件置位或复位 0: 关闭 TIMER15 时钟 1: 开启 TIMER15 时钟
19:16	保留	必须保持复位值
15	ADC2EN	ADC2 时钟使能 由软件置位或复位 0: 关闭 ADC2 时钟 1: 开启 ADC2 时钟
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 TIMERO
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

8.3.8. APB1 使能寄存器 (RCU_APB1EN)

地址偏移: 0x1C

复位值: 0x1000 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	保留	保留	保留	保留	CMPEN	I2C1EN	I2C0EN	UART4E	UART3E	USART2	USART1	保留
	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	保留	WWDGT	保留	保留	保留	保留	TIMER16	TIMER6E	TIMER5E	TIMER4E	TIMER3E	TIMER2E	TIMER1E
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
26:24	保留	必须保持复位值。
23	CMPEN	CMP 时钟使能 由软件置位或复位 0: 关闭 CMP 时钟 1: 开启 CMP 时钟
22	I2C1EN	I2C1 时钟使能 由软件置位或复位 0: 关闭 I2C1 时钟 1: 开启 I2C1 时钟

21	I2C0EN	I2C0 时钟使能 由软件置位或复位 0: 关闭 I2C0 时钟 1: 开启 I2C0 时钟
20	UART4EN	UART4 时钟使能 由软件置位或复位 0: 关闭 UART4 时钟 1: 开启 UART4 时钟
19	UART3EN	UART3 时钟使能 由软件置位或复位 0: 关闭 UART3 时钟 1: 开启 UART3 时钟
18	USART2EN	USART2 时钟使能 由软件置位或复位 0: 关闭 USART2 时钟 1: 开启 USART2 时钟
17	USART1EN	USART1 时钟使能 由软件置位或复位 0: 关闭 USART1 时钟 1: 开启 USART1 时钟
16	保留	必须保持复位值。
15	SPI2EN	SPI2 时钟使能 由软件置位或复位 0: 关闭 SPI2 时钟 1: 开启 SPI2 时钟
14	SPI1EN	SPI1 时钟使能 由软件置位或复位 0: 关闭 SPI1 时钟 1: 开启 SPI1 时钟
13:12	保留	必须保持复位值。
11	WWDGTEN	WWDGT 时钟使能 由软件置位或复位 0: 关闭 WWDGT 时钟 1: 开启 WWDGT 时钟
10:7	保留	必须保持复位值。
6	TIMER16EN	TIMER16 时钟使能

		由软件置位或复位 0: 关闭 TIMER16 时钟 1: 开启 TIMER16 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

8.3.9. 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**、**RTC SRC** 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。

保留															BKPRST
															r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCEN	保留					RTCSRC[1:0]		LXTALRD YRST	LCKMD	LCKMEN	LXTALDRI[1:0]		LXTALBP S	LXTALST B	LXTALEN
r/w						r/w		w	r	r/w	r/w		r/w	r	r/w

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	LXTALRDYRST	LXTAL 就绪复位 0: LXTAL 就绪时不复位 1: LXTAL 就绪时复位
6	LCKMD	LXTAL 时钟故障检测 由硬件置位，当外部 32 KHz 振荡器（LXTAL）上的时钟安全系统检测到故障。当 LCKMEN 或 LXTALEN 关闭时，该位清零。 0: LXTAL（32 KHz 振荡器）上未检测到故障 1: 在 LXTAL（32 KHz 振荡器）上检测到故障
5	LCKMEN	LXTAL 时钟监视器使能 0: 禁止 LXTAL 时钟监视器 1: 使能 LXTAL 时钟监视器 通过软件设置，启用 LXTAL（32 KHz 振荡器）上的时钟安全系统。LCKMEN 必须

在 LXTAL 已启用（LXTALEN 位已启用）和就绪（LXTALSTB 标志由硬件设置）。

注意：一旦该位被置位，该位可以通过备份域复位清除或者在检测到 LXTAL 时钟故障后（LCKMD = 1）通过复位 LCKMEN 清除。

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

8.3.10. 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	WWDGT RSTF	FWDGT RSTF	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	r	

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	窗口看门狗定时器复位标志位 窗口看门狗定时器复位发生时由硬件置 1 向 RSTFC 位写 1 来清除该位 0: 无窗口看门狗复位发生 1: 发生窗口看门狗复位
29	FWDGTRSTF	独立看门狗定时器复位标志位 独立看门狗复位发生时由硬件置 1 向 RSTFC 位写 1 来清除该位 0: 无独立看门狗定时器复位发生 1: 发生独立看门狗定时器复位
28	SWRSTF	软件复位标志位 软件复位发生时由硬件置 1 向 RSTFC 位写 1 来清除该位 0: 无软件复位发生 1: 发生软件复位
27	PORRSTF	电源复位标志位 电源复位发生时由硬件置 1 向 RSTFC 位写 1 来清除该位 0: 无电源复位发生 1: 发生电源复位
26	EPRSTF	外部引脚复位标志位 外部引脚复位发生时由硬件置 1 向 RSTFC 位写 1 来清除该位 0: 无外部引脚复位发生 1: 发生外部引脚复位
25	保留	必须保持复位值。
24	RSTFC	清除复位标志位 由软件置 1 来清除所有复位标志位 0: 无作用 1: 清除所有复位标志位

23:2	保留	必须保持复位值。
1	IRC40KSTB	IRC40K 时钟稳定标志位 该位由硬件置 1 指示 IRC40K 输出时钟是否稳定待用 0: IRC40K 时钟未稳定 1: IRC40K 已稳定
0	IRC40KEN	IRC40K 使能 由软件置位和复位 0: 关闭 IRC40K 时钟 1: 开启 IRC40K 时钟

8.3.11. 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										DMA1RS	DMA0RS	DMAMUX	TRNGRS	HAURST	CAURST
										T	T	RST	T		
										rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留			USBFSR	保留											
			ST												
rw															

位/位域	名称	描述
31:22	保留	必须保持复位值。
21	DMA1RST	DMA1 复位 由软件置位或复位 0: 无作用 1: 复位 DMA1
20	DMA0RST	DMA0 复位 由软件置位或复位 0: 无作用 1: 复位 DMA0
19	DMAMUXRST	DMAMUX 复位 由软件置位或复位 0: 无作用

		1: 复位 DMAMUX
18	TRNGRST	TRNG 复位 由软件置位或复位 0: 无作用 1: 复位 TRNG
17	HAURST	HAU 复位 由软件置位或复位 0: 无作用 1: 复位 HAU
16	CAURST	CAU 复位 由软件置位或复位 0: 无作用 1: 复位 CAU
15:13	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。

8.3.12. 时钟配置寄存器 1 (RCU_CFG1)

地址偏移: 0x2C

复位值: 0x0000 0400

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	ADCPSC[3]	IRC40KCALIB[4:0]						保留	PLL0SEL[1:0]		HXTALR DYRST	I2S2SEL	I2S1SEL	PLL1SEL	
rw		r						rw		rw		rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	PLL1MF[5:0]						PREDIV1[3:0]				PREDIV0[3:0]				
rw						rw				rw					

位/位域	名称	描述
31:30	保留	必须保持复位值。
29	ADCPSC[3]	ADCPSC 的第 3 位

参考寄存器 RCU_CFG0 的 14 到 15 位		
28:24	IRC40KCALIB[4:0]	内部 40KHz RC 振荡器校准值寄存器 上电时自动加载这些位
23:22	保留	必须保持复位值。
21:20	PLL0SEL[1:0]	PLL0 时钟源选择 通过软件置位和复位控制 PLL0 时钟源。 00: 选择 IRC8M 时钟作为 PLL0 的时钟源 01: 选择 IRC48M 时钟作为 PLL0 的时钟源 10: 选择 HXTAL 时钟作为 PLL0 的时钟源 11: 选择 CK_PLL1 时钟作为 PLL0 的时钟源
19	HXTALRDYRST	HXTAL 就绪复位 0: HXTAL 就绪不复位 1: HXTAL 就绪复位
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_PLL1 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_PLL1 被选择为 I2S1 时钟的时钟源
16	PLL1SEL	PLL1 时钟源选择 通过软件置位和复位控制 PLL1 时钟源。 0: 选择 HXTAL 时钟作为 PLL1 的时钟源 1: 选择 IRC48M 时钟作为 PLL1 的时钟源
15:14	保留	必须保持复位值。
13:8	PLL1MF[3:0]	PLL1 时钟倍频因子 由软件置位或清零 0000xx: 保留 000100: (PLL1 源时钟 x 4) 000101: (PLL1 源时钟 x 5) ... 111111: (PLL1 源时钟 x 63)
7:4	PREDIV1[3:0]	PREDIV1 分频因子 由软件置位或清零，PLL1 未使能时，可以修改这些位。

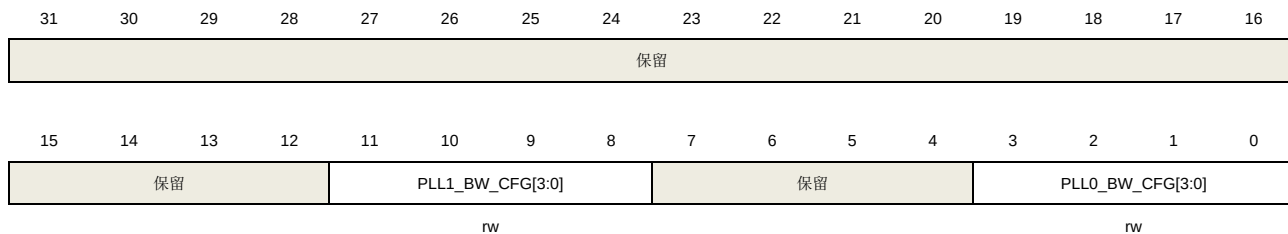
		0000: PREDIV1 输入源时钟未分频
		0001: PREDIV1 输入源时钟 2 分频
		0010: PREDIV1 输入源时钟 3 分频
		0011: PREDIV1 输入源时钟 4 分频
		0100: PREDIV1 输入源时钟 5 分频
		0101: PREDIV1 输入源时钟 6 分频
		0110: PREDIV1 输入源时钟 7 分频
		0111: PREDIV1 输入源时钟 8 分频
		1000: PREDIV1 输入源时钟 9 分频
		1001: PREDIV1 输入源时钟 10 分频
		1010: PREDIV1 输入源时钟 11 分频
		1011: PREDIV1 输入源时钟 12 分频
		1100: PREDIV1 输入源时钟 13 分频
		1101: PREDIV1 输入源时钟 14 分频
		1110: PREDIV1 输入源时钟 15 分频
		1111: PREDIV1 输入源时钟 16 分频
3:0	PREDIV0[3:0]	PREDIV0 分频因子 由软件置位或清零，PLL0 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: PREDIV0 输入源时钟未分频 0001: PREDIV0 输入源时钟 2 分频 0010: PREDIV0 输入源时钟 3 分频 0011: PREDIV0 输入源时钟 4 分频 0100: PREDIV0 输入源时钟 5 分频 0101: PREDIV0 输入源时钟 6 分频 0110: PREDIV0 输入源时钟 7 分频 0111: PREDIV0 输入源时钟 8 分频 1000: PREDIV0 输入源时钟 9 分频 1001: PREDIV0 输入源时钟 10 分频 1010: PREDIV0 输入源时钟 11 分频 1011: PREDIV0 输入源时钟 12 分频 1100: PREDIV0 输入源时钟 13 分频 1101: PREDIV0 输入源时钟 14 分频 1110: PREDIV0 输入源时钟 15 分频 1111: PREDIV0 输入源时钟 16 分频

8.3.13. PLL 带宽配置寄存器（RCU_PLLBWCFG）

地址偏移：0x30

复位值：0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。



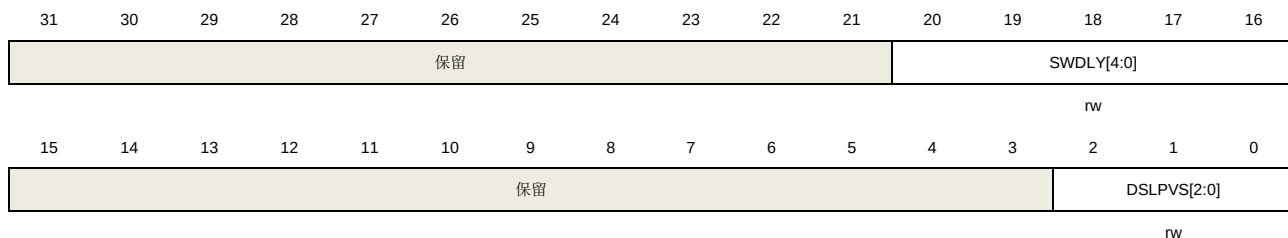
位/位域	名称	描述
31:12	保留	必须保持复位值。
11:8	PLL1_BW_CFG[3:0]	PLL1 带宽配置 Bit[2]和 Bit[0]在进入 PLL1 模拟接口之前取反
7:4	保留	必须保持复位值。
3:0	PLL0_BW_CFG[3:0]	PLL0 带宽配置 Bit[2]和 Bit[0]在进入 PLL0 模拟接口之前取反

8.3.14. 深度睡眠模式电压寄存器（RCU_DSV）

地址偏移：0x34

复位值：0x000A 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
31:21	保留	必须保持复位值。
20:16	SWDLY[4:0]	切换延迟时间 当进入深度睡眠模式时，切换到 IRC8M 时钟，需等待 SWDLY 个 IRC8M 时钟周期，然后进入深度睡眠模式。默认为 10 个 IRC8M 时钟。 注意： 不要配置为 0，因为它是一个序列。
15:3	保留	必须保持复位值。
2:0	DSL PVS[2:0]	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缺省值

001：在深度睡眠模式下内核电压为（缺省值-0.05）V

010：在深度睡眠模式下内核电压为（缺省值-0.1）V

011：在深度睡眠模式下内核电压为（缺省值-0.15）V

100：在深度睡眠模式下内核电压为（缺省值-0.2）V

101：在深度睡眠模式下内核电压为（缺省值-0.25）V

110：在深度睡眠模式下内核电压为（缺省值-0.3）V（不建议客户使用）

111：在深度睡眠模式下内核电压为（缺省值-0.35）V（不建议客户使用）

8.3.15. 时钟频率监视器配置寄存器 0（RCU_CKFMCFG0）

地址偏移：0x40

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										IRC8MCK	IRC8MCK	IRC8MCK	IRC8MCKFMC[1:0]		IRC8MCK
										FFIE	FFC	FFF			FMEN
										rw	w	r	rw		rw

位/位域	名称	描述
31:6	保留	必须保持复位值。
5	IRC8MCKFFIE	IRC8M时钟频率故障中断使能 该位由软件置位和复位。 0：关闭IRC8M时钟频率故障中断 1：使能IRC8M时钟频率故障中断
4	IRC8MCKFFC	IRC8M时钟频率故障标志清除 用软件编写1复位IRC8MCKFFF标志，并在IRC8MCKFFF被清除后由硬件自动复位。 0：不复位IRC8MCKFFF标志 1：复位IRC8MCKFFF标志位
3	IRC8MCKFFF	IRC8M时钟频率故障标志 该位表示IRC8M时钟频率是否超出监控范围。 0：IRC8M时钟频率未超出监控范围 1：IRC8M时钟频率超出监控范围
2:1	IRC8MCKFMC[1:0]	IRC8M时钟频率监视器配置

该位由软件设置和复位。

00: 时钟频率监控范围为8M Hz±5%

01: 时钟频率监控范围为8M Hz±10%

10: 时钟频率监控范围为8M Hz±15%

11: 时钟频率监控范围为8M Hz±20%

0	IRC8MCKFMEN	IRC8M时钟频率监控使能 该位由软件设置和复位。 0: 关闭内部8M RC振荡器（IRC8M）时钟频率监视器 1: 使能内部8M RC振荡器（IRC8M）时钟频率监控器

8.3.16. 时钟频率监视器配置寄存器 1（RCU_CKFMCFG1）

地址偏移: 0x44

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
HXTALC KFFC	HXTALC KFFF	保留				HXTALC KFFIE	HXTALCKFMIN[11:0]								
w	r					rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HXTALCKFMIN[11:0]						HXTALCKFMAX[11:0]									HXTALC KFMEN
rw						rw									rw

位/位域	名称	描述
31	HXTALCKFFC	HXTAL时钟频率故障标志清除 用软件写1复位HXTALCKKFFF标志，并在HXTALCKKFFF被清除后由硬件自动复位。 0: 不复位HXTALCKKFFF标志 1: 复位HXTALCKKFFF标志位
30	HXTALCKFFF	HXTAL时钟频率故障标志 该位表示HXTAL时钟频率是否超出监控范围。 0: HXTAL时钟频率未超出监控范围 1: HXTAL时钟频率超出监控范围
29:26	保留	必须保持复位值。
25	HXTALCKFFIE	HXTAL时钟频率故障中断使能 该位由软件置位和复位。 0: 关闭HXTAL时钟频率故障中断 1: 使能HXTAL时钟频率故障中断

24:13	HXTALCKFMIN[11:0]	HXTAL时钟频率监视器配置最小值 该位由软件设置和复位。 参考时钟为IRC48M，监控窗口为1000个IRC48M时钟周期。
12:1	HXTALCKFMAX[11:0]	HXTAL时钟频率监视器配置最大值 该位由软件设置和复位。 参考时钟为IRC48M，监控窗口为1000个IRC48M时钟周期。
0	HXTALCKFMEN	HXTAL时钟频率监控使能 该位由软件设置和复位。 0：关闭HXTAL时钟频率监视器 1：使能HXTAL时钟频率监视器

8.3.17. 时钟频率监视器配置寄存器 2 (RCU_CKFMCFG2)

地址偏移：0x48

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
PLL0PCK FFC	PLL0PCK FFF	保留						PLL0PCK FFIE	保留	PLL0PCKFMIN[9:0]					
w	r							rw		rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLL0PCKFMIN[9:0]				保留	PLL0PCKFMAX[9:0]										
rw					rw										
															PLL0PCK FMEN
															rw

位/位域	名称	描述
31	PLL0PCKFFC	PLL0P时钟频率故障标志清除 用软件写1复位PLL0PCKFFF标志，并在PLL0PCKFFF被清除后由硬件自动复位。 0：不复位PLL0PCKFFF标志 1：复位PLL0PCKFFF标志位
30	PLL0PCKFFF	PLL0P时钟频率故障标志 该位表示PLL0P时钟频率是否超出监控范围。 0：PLL0P时钟频率未超出监控范围 1：PLL0P时钟频率超出监控范围
29:24	保留	必须保持复位值。
23	PLL0PCKFFIE	PLL0P时钟频率故障中断使能 该位由软件置位和复位。 0：关闭PLL0P时钟频率故障中断

		1: 使能PLL0P时钟频率故障中断
22	保留	必须保持复位值。
21:12	PLL0PCKFMIN[9:0]	PLL0P时钟频率监视器配置最小值 该位由软件设置和复位。 参考时钟为IRC48M，监控窗口为100个IRC48M时钟周期。
11	保留	必须保持复位值。
10:1	PLL0PCKFMAX[9:0]	PLL0P时钟频率监视器配置最大值 该位由软件设置和复位。 参考时钟为IRC48M，监控窗口为100个IRC48M时钟周期。
0	PLL0PCKFMEN	PLL0P时钟频率监控使能 该位由软件设置和复位。 0: 关闭PLL0P时钟频率监视器 1: 使能PLL0P时钟频率监视器

8.3.18. 时钟频率监视器配置寄存器 3 (RCU_CKFMCFG3)

地址偏移: 0x4C

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
PLL1CKF FC	PLL1CKF FF	保留						PLL1CKF FIE	保留	PLL1CKFMIN[9:0]					
w	r							rw		rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLL1CKFMIN[9:0]				保留	PLL1CKFMAX[9:0]										
rw					rw										
														PLL1CKF MEN	
														rw	

位/位域	名称	描述
31	PLL1CKFFC	PLL1时钟频率故障标志清除 用软件写1复位PLL1CKFFFF标志，并在PLL1CKFFFF被清除后由硬件自动复位。 0: 不复位PLL1CKFFFF标志 1: 复位PLL1CKFFFF标志位
30	PLL1CKFFF	PLL1时钟频率故障标志 该位表示PLL1时钟频率是否超出监控范围。 0: PLL1时钟频率未超出监控范围 1: PLL1时钟频率超出监控范围

29:24	保留	必须保持复位值。
23	PLL1CKFFIE	PLL1时钟频率故障中断使能 该位由软件置位和复位。 0: 关闭PLL1时钟频率故障中断 1: 使能PLL1时钟频率故障中断
22	保留	必须保持复位值。
21:12	PLL1CKFMIN[9:0]	PLL1时钟频率监视器配置最小值 该位由软件设置和复位。 参考时钟为IRC48M，监控窗口为100个IRC48M时钟周期。
11	保留	必须保持复位值。
10:1	PLL1CKFMAX[9:0]	PLL1时钟频率监视器配置最大值 该位由软件设置和复位。 参考时钟为IRC48M，监控窗口为100个IRC48M时钟周期。
0	PLL1CKFMEN	PLL1时钟频率监控使能 该位由软件设置和复位。 0: 关闭PLL1时钟频率监视器 1: 使能PLL1时钟频率监视器

8.3.19. 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	I2S2DIV[4:0]				IRC48MS TB	IRC48ME N	
r									rw					r	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FMCDIV[3:0]				PLL0DIV[3:0]				I2S1DIV[4:0]				FMCSEL[1:0]		CK48MS EL	
rw				rw				rw				rw		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	保留	必须保持复位值。

22:18	I2S2DIV[4:0]	I2S2分频因子 分频因子为I2S2DIV +1。
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
15:12	FMCDIV[3:0]	FMC分频因子 分频因子为FMCDIV +1。 注意: 在整个芯片的生命周期中,需确保FMC接口时钟大于等于CK_AHB时钟,但不超过CK_AHB时钟的7倍,否则可能发生不可预知的问题。
11:8	PLL0DIV[3:0]	PLL0分频因子 分频因子为PLL0DIV +1。
7:3	I2S1DIV[4:0]	I2S1分频因子 分频因子为I2S1DIV +1。
2:1	FMCSEL[1:0]	FMC时钟源选择 通过软件置位和复位控制FMC时钟源 00: 选择CK_AHB作为FMC时钟源 01: 选择CK_SYS作为FMC时钟源 10: 选择CK_PLL0作为FMC时钟源 11: 选择CK_PLL1作为FMC时钟源 注意: 在整个芯片的生命周期中,需确保FMC接口时钟大于等于CK_AHB时钟,但不超过CK_AHB时钟的7倍,否则可能发生不可预知的问题。
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

8.3.20. 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 TBIC	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。

8.3.21. 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	保留										
rw															

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

位/位域	名称	描述
31:28	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。

8.3.22. 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。

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

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

8.3.23. 上锁寄存器 (RCU_LOCK)

地址偏移: 0x100

复位值: 0x0000 0000

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

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

位/位域	名称	描述
31	LOCK	RCU 寄存器上锁位 该位由软件置位和复位。 0: RCU 寄存器可配置 1: RCU 寄存器不可配置，只能读 注意： 当 LOCK = 1，写 RCU 寄存器时，会产生 HardFault 异常。
30:0	保留	必须保持复位值。

9. 时钟校准控制器（CTC）

9.1. 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）。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。

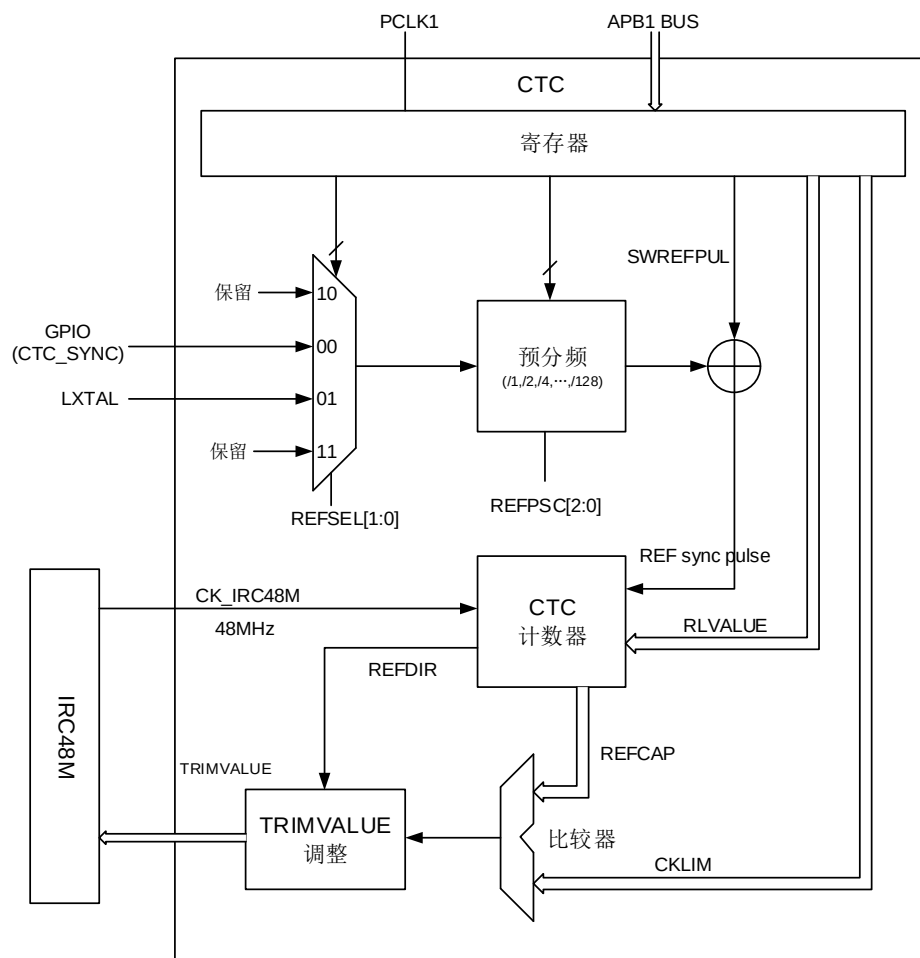
9.2. 主要特征

- 两个外部参考信号源：GPIO（CTC_SYNC），LXTAL时钟；
- 提供软件参考同步脉冲；
- 硬件自动校准，无需软件操作；
- 具有参考信号源捕获和重载功能的16 bits校准计数器；
- 用于频率评估和自动校准的8 bits时钟校准基值；
- 标志位和中断，用于指示时钟校准的状态：校准成功状态（CKOKIF），警告状态（CKWARNIF）和错误状态（ERRIF）。

9.3. 功能说明

CTC 模块的内部结构图如[图9-1. CTC 简介](#)。

图 9-1. CTC 简介



9.3.1. REF 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 (CTC_SYNC)，LXTAL 时钟输出。

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

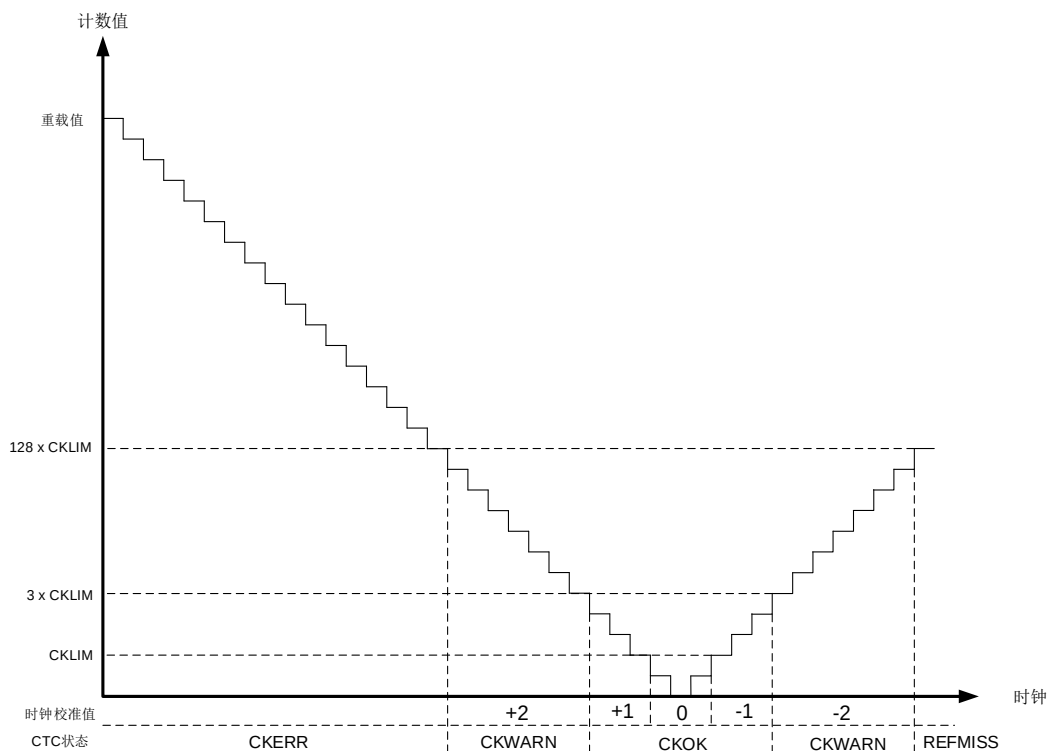
如果需要使用软件参考脉冲信号，则需要设置 CTC_CTL0 寄存器中的 SWREFPUL 位为 1。软件参考脉冲信号与外部参考脉冲信号最后进行逻辑或操作。

9.3.2. CTC 校准计数器

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

新开始向下计数。如果始终检测不到 REF 同步脉冲信号，计数器会向下计数到零，然后再向上计数到 $128 \times \text{CKLIM}$ （CKLIM 在 CTC_CTL1 中定义），最后停止，直到检测到下一个 REF 同步脉冲信号。一旦检测到 REF 同步脉冲信号，当前 CTC 校准计数器的计数值被捕获存入 CTC_STAT 寄存器中的 REFCAP 位，同时，当前计数器的计数方向被存入 CTC_STAT 寄存器中的 REFDIR 位。详细内容如 [图9-2. CTC 校准计数器](#)所示。

图 9-2. CTC 校准计数器



9.3.3. 频率评估和自动校准过程

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

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

- Counter < CKLIM时，检测到REF同步脉冲信号；

CTC_STAT 中的 CKOKIF 位(时钟校准成功标志位)被置位,同时,如果 CTC_CTL0 中的 CKOKIE 位(时钟校准完成中断使能位)置 1,将会产生一个中断。

如果 CTC_CTL0 中的 AUTOTRIM 置 1, CTC_CTL0 中的 TRIMVALUE 值不变。

■ $CKLIM \leq Counter < 3 \times CKLIM$ 时,检测到REF同步脉冲信号;

CTC_STAT 中的 CKOKIF 位被置位,同时,如果 CTC_CTL0 中的 CKOKIE 位置 1,将会产生一个中断。

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

■ $3 \times CKLIM \leq Counter < 128 \times CKLIM$ 时,检测到REF同步脉冲信号;

CTC_STAT 中的 CKWARNIF 位(时钟校准警告中断位)被置位,同时,如果 CTC_CTL0 中的 CKWARNIE 位(时钟校准警告中断使能位)置 1,将会产生一个中断。

如果 CTC_CTL0 中的 AUTOTRIM 位置 1,在计数器向下计数过程中,CTC_CTL0 中的 TRIMVALUE 值将加 2,而在向上计数过程中将减 2。

■ $Counter \geq 128 \times CKLIM$,计数器在向下计数过程中,检测到REF同步脉冲信号;

CTC_STAT 中的 CKERR 位(时钟校准错误位)被置位,同时,如果 CTC_CTL0 中的 ERRIE 位(错误中断使能位)置 1,将会产生一个中断。

CTC_CTL0 中的 TRIMVALUE 值不变。

■ $Counter = 128 \times CKLIM$,计数器在向上计数过程中;

CTC_STAT 中的 REFMISS 位(REF 同步脉冲丢失位)被置位,同时,如果 CTC_CTL0 中的 ERRIE 位置 1,将会产生一个中断。

CTC_CTL0 中的 TRIMVALUE 值不变。

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

9.3.4. 软件编程指南

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

$$RLVALUE = (F_{clock} \div F_{REF}) - 1 \quad (9-1)$$

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

$$CKLIM = (F_{clock} \div F_{REF}) \times 0.12\% \div 2 \quad (9-2)$$

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

9.4. CTC 寄存器

CTC 基地址：0x4000 C800

9.4.1. 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
rw		rw						w	rw	rw		rw	rw	rw	rw

位/位域	名称	描述
31:14	保留	必须保持复位值。
13:8	TRIMVALUE[5:0]	IRC48M 校准值 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。 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。 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。
7	SWREFPUL	软件生成同步参考信号脉冲 该位由软件置位，并为 CTC 计数器提供一个同步参考脉冲信号。该位由硬件自动清除，读操作时返回 0。 0: 没有影响 1: 软件产生一个同步参考脉冲信号
6	AUTOTRIM	硬件自动校准模式 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。 0: 禁止硬件自动校准模式 1: 使能硬件自动校准模式
5	CNTEN	CTC 计数器使能

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。

0：禁止 CTC 计数器

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

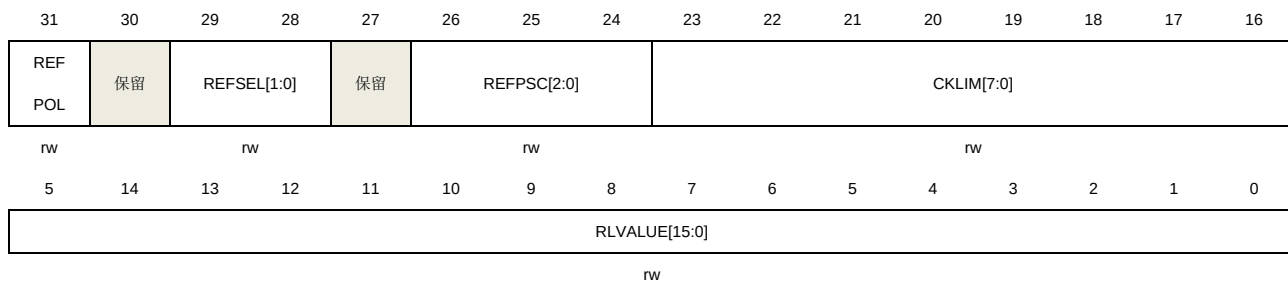
9.4.2. 控制寄存器 1（CTC_CTL1）

地址偏移：0x04

复位值：0x2022 BB7F

该寄存器只能按字（32 位）访问。

注意：当 CNTEN 为 1 时，不能修改该寄存器的值。



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

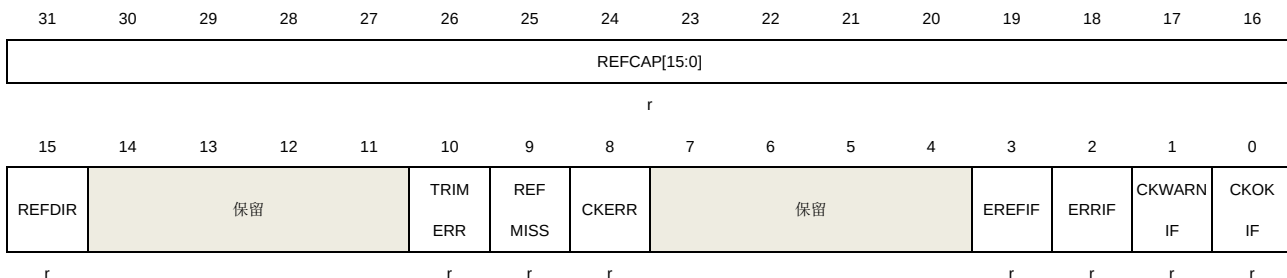
30	保留	必须保持复位值。
29:28	REFSEL[1:0]	参考信号源选择 该位由软件置位或清除，用于选择参考信号源 00: 选择 GPIO (CTC_SYNC) 输入信号 01: 选择 LXTAL 时钟 10: 保留 11: 保留
27	保留	必须保持复位值。
26:24	REFPSC[2:0]	参考信号源预分频 该位由软件置位或清除 000: 参考信号不分频 001: 参考信号 2 分频 010: 参考信号 4 分频 011: 参考信号 8 分频 100: 参考信号 16 分频 101: 参考信号 32 分频 110: 参考信号 64 分频 111: 参考信号 128 分频
23:16	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，当检测到一个同步参考脉冲时，该值将重载到 CTC 校准计数器中。

9.4.3. 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。



位/位域	名称	描述
31:16	REFCAP[15:0]	CTC 计数器捕获值 当检测到一个同步参考脉冲信号时，CTC 校准计数器中的计数值被存入到 REFCAP 位中。
15	REFDIR	CTC 校准时钟计数方向 当检测到一个同步参考脉冲信号时，CTC 校准计数器的计数方向被存入 REFDIR 位中。 0: 向上计数 1: 向下计数
14:11	保留	必须保持复位值。
10	TRIMERR	校准值错误位 当 CTC_CTL0 中的 TRIMVALUE 值发生上溢或下溢时，该位由硬件置位。若 CTC_CTL0 中的 ERRIE 位置 1，则会产生一个中断。通过写 1 到 CTC_INTC 中的 ERRIC 位，可以将 TRIMERR 位清零。 0: 无校准值错误发生 1: 发生校准值错误
9	REFMISS	同步参考脉冲信号丢失 当同步参考脉冲信号丢失时，该位由硬件置位。当 CTC 校准计数器在增计数的过程中计数到 128 x 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。 0: 无同步参考脉冲信号丢失 1: 同步参考脉冲信号丢失
8	CKERR	时钟校准错误位 当时钟校准错误产生时，该位由硬件置位。当 CTC 校准计数器计数值在减计数的过程中大于或等于 128 x 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。 0: 无时钟校准错误发生 1: 发生时钟校准错误
7:4	保留	必须保持复位值。
3	EREFIF	期望参考中断标志位 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。 0: 无期望参考信号产生 1: 期望参考信号产生

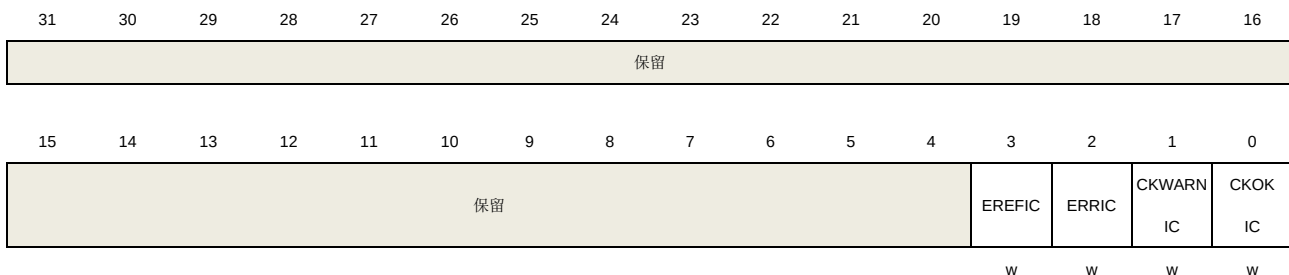
2	ERRIF	错误中断标志位 当发生一个错误时，该位由硬件置位。只要有 TRIMERR，REFMISS 或者 CKERR 错误发生时，该位置位。当 CTC_CTL0 中的 ERRIE 置位时，产生一个中断。通过写 1 到 CTC_INTC 中的 ERRIC 位，可以将 ERRIF 位清零。 0：无错误发生 1：发生错误
1	CKWARNIF	时钟校准警告中断标志位 当时钟校准警告产生时，该位由硬件置位。当 CTC 校准计数器计数值大于或等于 3 x CKLIM 且小于 128 x CKLIM，并检测到同步参考脉冲信号时，CKWARNIF 置位。这说明当前时钟频率太慢或者太快，但可以通过校准达到期望频率值。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。 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 x 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

9.4.4. 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。



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。

10. 通用和备用输入/输出接口（GPIO 和 AFIO）

10.1. 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 和 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。外设 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。

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、下拉或浮空。除模拟模式外，所有的 GPIO 引脚都具备大电流驱动能力。

10.2. 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；
- 端口锁定配置；

10.3. 功能说明

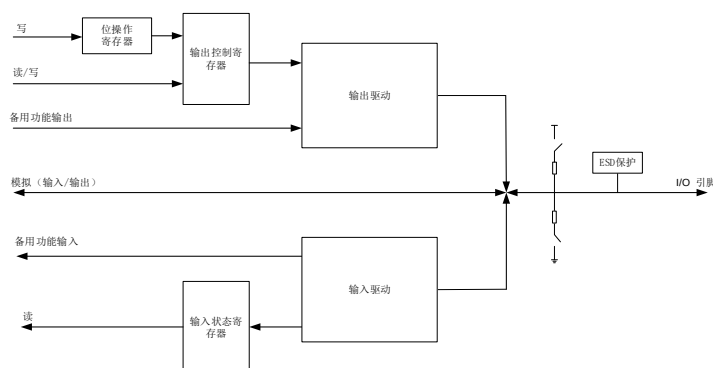
每个通用 I/O 端口都可以通过 32 位控制寄存器（GPIOx_CTL）配置为 GPIO 输入，GPIO 输出，AF 功能或模拟模式。当选择 AF 功能时，引脚 AF 输入/输出是通过 AF 功能输出使能来选择。当端口配置为输出（GPIO 输出或 AFIO 输出）时，可以通过 GPIO 输出模式寄存器（GPIOx_OMODE）配置为推挽或开漏模式。输出端口的最大速度可以通过 GPIO 输出速度寄存器（GPIOx_OSPD）配置。每个端口可以通过 GPIO 上/下拉寄存器（GPIOx_PUD）配置为浮空（无上拉或下拉），上拉或下拉功能。详情请见[表 10-1. GPIO 配置表](#)。

表 10-1. GPIO 配置表

PAD TYPE			CTLy	OMy	PUDy
GPIO 输入	X	浮空	00	X	00
		上拉			01
		下拉			10
GPIO 输出	推挽	浮空	01	0	00
		上拉			01
		下拉			10
	开漏	浮空		1	00
		上拉			01
		下拉			10
AFIO 输入	X	浮空	10	X	00
		上拉			01
		下拉			10
AFIO 输出	推挽	浮空	10	0	00
		上拉			01
		下拉			10
	开漏	浮空		1	00
		上拉			01
		下拉			10
模拟	X	X	11	X	XX

图 10-1. 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 10-1. 标准 I/O 端口位的基本结构



10.3.1. 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。

注意:

当 HXTAL 功能禁用时, 对 PD0/PD1 进行配置时, 需要注意以下事项:

1. 配置为普通 GPIO 输出时, PD0 的输出同时会在 PD0/OSCIN 上体现, PD1 的输出同时会在 PD1/OSCCOUT 上体现。
2. 配置为普通 GPIO 输入时, 对应 ISTAT 中的状态只显示 OSCIN / OSCOUT 上的引脚状态, 而不体现 PD0/PD1 的引脚状态; 若选择 PD0/PD1 作为 EXTI 触发源时, 只能通过 OSCIN/OSCCOUT 引脚进行触发。
3. 配置为 AF 功能时, HXTAL 功能禁用时, AF8 功能只作用在 OSCIN / OSCOUT 上; HXTAL 功能打开时, AF8 功能只作用在 PD0 / PD1 上; 另外不论 HXTAL 功能禁用还是打开, AF0~AF7 只作用在 PD0/PD1 上。

10.3.2. 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。

10.3.3. 备用功能 (AF)

当端口配置为 AFIO (设置 GPIOx_CTL 寄存器中的 CTLy 值为“0b10”) 时, 该端口用作外设备用功能。通过配置 GPIO 备用功能选择寄存器 (GPIOx_AFSELz (z=0,1)), 每个端口可以配置 16 个备用功能。端口备用功能分配的详细介绍见芯片数据手册。

10.3.4. 附加功能

有些引脚具有附加功能, 它们优先于标准 GPIO 寄存器中的配置。当用作 ADC 或 DAC 附加功能

时，引脚必须配置成模拟模式。当引脚用作 RTC、WKUPx 和振荡器附加功能时，端口类型通过相关的 RTC、PMU 和 RCU 寄存器自动设置。当附加功能禁用时，这些端口可用作普通 GPIO。

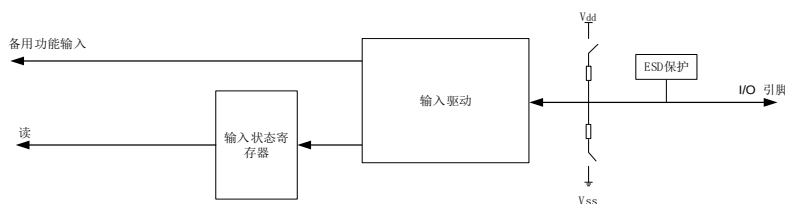
10.3.5. 输入配置

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 10-2. 输入配置的基本结构](#)显示 I/O 引脚的输入配置。

图 10-2. 输入配置的基本结构



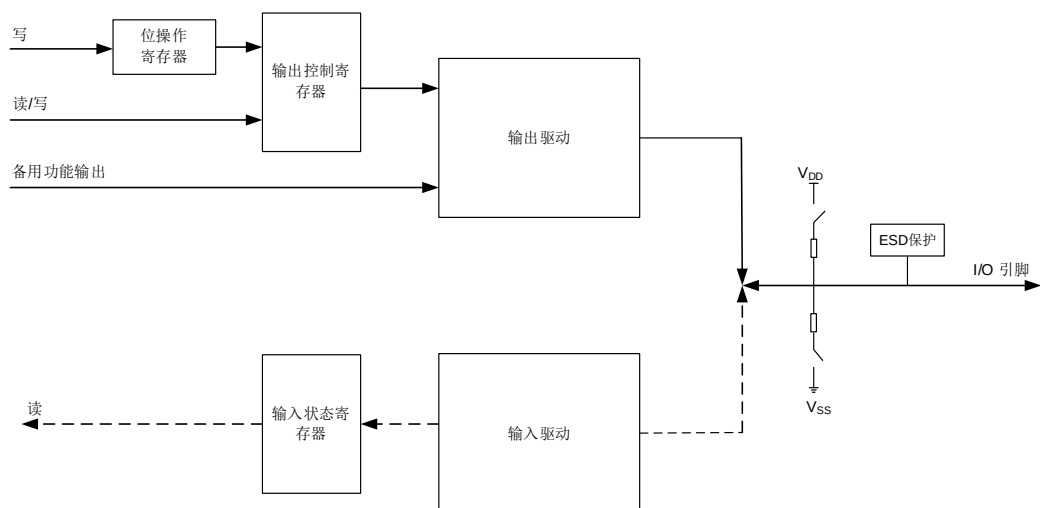
10.3.6. 输出配置

当 GPIO 配置为输出时：

- 施密特触发输入使能；
- 可选的弱上拉和下拉电阻；
- 输出缓冲器使能；
- 开漏模式：输出控制寄存器设置为“0”时，相应引脚输出低电平；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 10-3. 输出配置的基本结构](#)是 I/O 端口的输出配置。

图 10-3. 输出配置的基本结构



10.3.7. 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为“0”。

[图 10-4. 模拟配置的基本结构](#)是 I/O 端口的模拟模式配置。

图 10-4. 模拟配置的基本结构



10.3.8. 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。

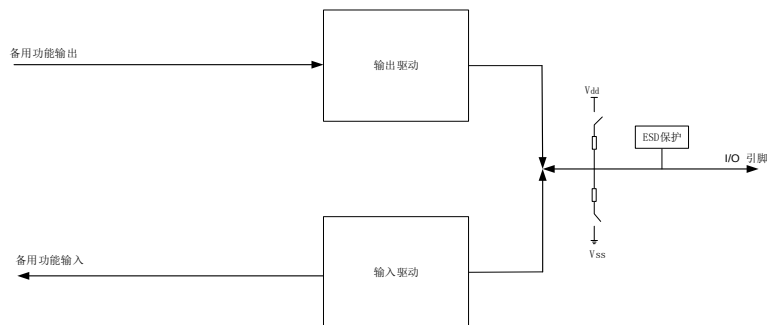
当引脚配置为备用功能时：

- 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 10-5. 备用功能配置的基本结构](#)是 I/O 端口备用功能配置图。

图 10-5. 备用功能配置的基本结构



10.3.9. GPIO 锁定功能

GPIO 的锁定机制可以保护 I/O 端口的配置。

被保护的寄存器有：GPIOx_CTL，GPIOx_OMODE，GPIOx_OSPD，GPIOx_PUD 和 GPIOx_AFSELz (z=0,1)。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。

10.4. 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

10.4.1. 端口控制寄存器 0 (GPIOx_CTL, x=A...E)

地址偏移: 0x00

复位值: 端口 A 0xABFF FFFF; 端口 B 0xFFFF FE8F; 其他端口 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
CTL15[1:0]		CTL14[1:0]		CTL13[1:0]		CTL12[1:0]		CTL11[1:0]		CTL10[1:0]		CTL9[1:0]		CTL8[1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTL7[1:0]		CTL6[1:0]		CTL5[1:0]		CTL4[1:0]		CTL3[1:0]		CTL2[1:0]		CTL1[1:0]		CTL0[1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	

位/位域	名称	描述
31:30	CTL15[1:0]	Pin 15 配置位 该位由软件置位和清除。 参考 CTL0[1:0]的描述
29:28	CTL14[1:0]	Pin 14 配置位 该位由软件置位和清除。 参考 CTL0[1:0]的描述
27:26	CTL13[1:0]	Pin 13 配置位 该位由软件置位和清除。 参考 CTL0[1:0]的描述
25:24	CTL12[1:0]	Pin 12 配置位 该位由软件置位和清除。 参考 CTL0[1:0]的描述
23:22	CTL11[1:0]	Pin 11 配置位

		该位由软件置位和清除。 参考 CTL0[1:0]的描述
21:20	CTL10[1:0]	Pin 10 配置位 该位由软件置位和清除。 参考 CTL0[1:0]的描述
19:18	CTL9[1:0]	Pin 9 配置位 该位由软件置位和清除。 参考 CTL0[1:0]的描述
17:16	CTL8[1:0]	Pin 8 配置位 该位由软件置位和清除。 参考 CTL0[1:0]的描述
15:14	CTL7[1:0]	Pin 7 配置位 该位由软件置位和清除。 参考 CTL0[1:0]的描述
13:12	CTL6[1:0]	Pin 6 配置位 该位由软件置位和清除。 参考 CTL0[1:0]的描述
11:10	CTL5[1:0]	Pin 5 配置位 该位由软件置位和清除。 参考 CTL0[1:0]的描述
9:8	CTL4[1:0]	Pin 4 配置位 该位由软件置位和清除。 参考 CTL0[1:0]的描述
7:6	CTL3[1:0]	Pin 3 配置位 该位由软件置位和清除。 参考 CTL0[1:0]的描述
5:4	CTL2[1:0]	Pin 2 配置位 该位由软件置位和清除。 参考 CTL0[1:0]的描述
3:2	CTL1[1:0]	Pin 1 配置位 该位由软件置位和清除。 参考 CTL0[1:0]的描述
1:0	CTL0[1:0]	Pin 0 配置位 该位由软件置位和清除。 00: GPIO 输入模式（复位值） 01: GPIO 输出模式

- 10: 备用功能描述
11: 模拟模式（输入和输出）

10.4.2. 端口输出模式寄存器（GPIOx_OMODE, x=A...E）

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
OM15	OM14	OM13	OM12	OM11	OM10	OM9	OM8	OM7	OM6	OM5	OM4	OM3	OM2	OM1	OM0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	OM15	Pin 15 输出模式位 该位由软件置位和清除。 参考 OM0 的描述
14	OM14	Pin 14 输出模式位 该位由软件置位和清除。 参考 OM0 的描述
13	OM13	Pin 13 输出模式位 该位由软件置位和清除。 参考 OM0 的描述
12	OM12	Pin 12 输出模式位 该位由软件置位和清除。 参考 OM0 的描述
11	OM11	Pin 11 输出模式位 该位由软件置位和清除。 参考 OM0 的描述
10	OM10	Pin 10 输出模式位 该位由软件置位和清除。 参考 OM0 的描述
9	OM9	Pin 9 输出模式位 该位由软件置位和清除。

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

10.4.3. 端口输出速度寄存器（GPIOx_OSPD, x=A...E）

地址偏移：0x08

复位值：端口 A 0x0C00 0000；端口 B 0x0000 00C0；其他端口 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

OSPD15[1:0]	OSPD14[1:0]	OSPD13[1:0]	OSPD12[1:0]	OSPD11[1:0]	OSPD10[1:0]	OSPD9[1:0]	OSPD8[1:0]
rw	rw	rw	rw	rw	rw	rw	rw
15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
OSPD7[1:0]	OSPD6[1:0]	OSPD5[1:0]	OSPD4[1:0]	OSPD3[1:0]	OSPD2[1:0]	OSPD1[1:0]	OSPD0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

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

参考 OSPD0[1:0]的描述

11:10	OSPD5[1:0]	Pin 5 输出最大速度位 该位由软件置位和清除。 参考 OSPD0[1:0]的描述
9:8	OSPD4[1:0]	Pin 4 输出最大速度位 该位由软件置位和清除。 参考 OSPD0[1:0]的描述
7:6	OSPD3[1:0]	Pin 3 输出最大速度位 该位由软件置位和清除。 参考 OSPD0[1:0]的描述
5:4	OSPD2[1:0]	Pin 2 输出最大速度位 该位由软件置位和清除。 参考 OSPD0[1:0]的描述
3:2	OSPD1[1:0]	Pin 1 输出最大速度位 该位由软件置位和清除。 参考 OSPD0[1:0]的描述
1:0	OSPD0[1:0]	Pin 0 输出最大速度位 该位由软件置位和清除。 00: 输出速度等级 0（复位值） 01: 输出速度等级 1 10: 输出速度等级 2 11: 输出速度等级 3

10.4.4. 端口上拉/下拉寄存器（GPIOx_PUD, x=A...E）

地址偏移：0x0C

复位值：端口 A 0x6400 0000；端口 B 0x0000 0100；其他端口 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
PUD15[1:0]		PUD14[1:0]		PUD13[1:0]		PUD12[1:0]		PUD11[1:0]		PUD10[1:0]		PUD9[1:0]		PUD8[1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUD7[1:0]		PUD6[1:0]		PUD5[1:0]		PUD4[1:0]		PUD3[1:0]		PUD2[1:0]		PUD1[1:0]		PUD0[1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	

位/位域	名称	描述
31:30	PUD15[1:0]	Pin 15 上拉/下拉位

		该位由软件置位和清除。 参考 PUD0[1:0]的描述
29:28	PUD14[1:0]	Pin 14 上拉/下拉位 该位由软件置位和清除。 参考 PUD0[1:0]的描述
27:26	PUD13[1:0]	Pin 13 上拉/下拉位 该位由软件置位和清除。 参考 PUD0[1:0]的描述
25:24	PUD12[1:0]	Pin 12 上拉/下拉位 该位由软件置位和清除。 参考 PUD0[1:0]的描述
23:22	PUD11[1:0]	Pin 11 上拉/下拉位 该位由软件置位和清除。 参考 PUD0[1:0]的描述
21:20	PUD10[1:0]	Pin 10 上拉/下拉位 该位由软件置位和清除。 参考 PUD0[1:0]的描述
19:18	PUD9[1:0]	Pin 9 上拉/下拉位 该位由软件置位和清除。 参考 PUD0[1:0]的描述
17:16	PUD8[1:0]	Pin 8 上拉/下拉位 该位由软件置位和清除。 参考 PUD0[1:0]的描述
15:14	PUD7[1:0]	Pin 7 上拉/下拉位 该位由软件置位和清除。 参考 PUD0[1:0]的描述
13:12	PUD6[1:0]	Pin 6 上拉/下拉位 该位由软件置位和清除。 参考 PUD0[1:0]的描述
11:10	PUD5[1:0]	Pin 5 上拉/下拉位 该位由软件置位和清除。 参考 PUD0[1:0]的描述
9:8	PUD4[1:0]	Pin 4 上拉/下拉位 该位由软件置位和清除。 参考 PUD0[1:0]的描述

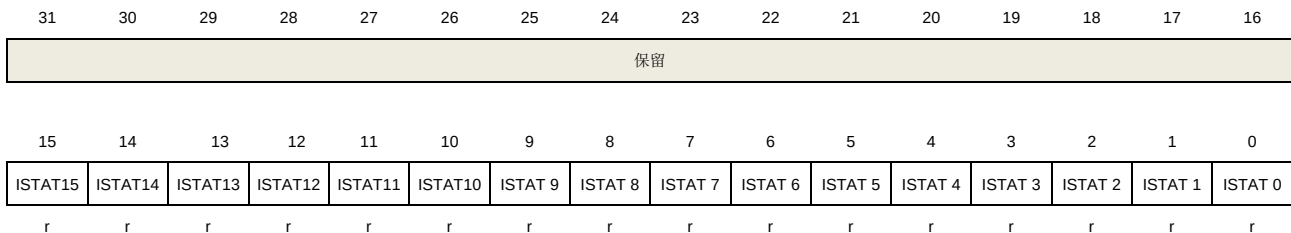
7:6	PUD3[1:0]	Pin 3 上拉/下拉位 该位由软件置位和清除。 参考 PUD0[1:0]的描述
5:4	PUD2[1:0]	Pin 2 上拉/下拉位 该位由软件置位和清除。 参考 PUD0[1:0]的描述
3:2	PUD1[1:0]	Pin 1 上拉/下拉位 该位由软件置位和清除。 参考 PUD0[1:0]的描述
1:0	PUD0[1:0]	Pin 0 上拉/下拉位 该位由软件置位和清除。 00: 浮空模式，无上拉/下拉（复位值） 01: 端口上拉模式 10: 端口下拉模式 11: 保留

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

地址偏移: 0x10

复位值: 0x0000 XXXX

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	ISTATy	端口输入状态位（y=0..15） 这些位由硬件置位和清除。 0: 引脚输入信号为低电平 1: 引脚输入信号为高电平

10.4.6. 端口输出控制寄存器（GPIOx_OCTL, 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
OCTL15	OCTL14	OCTL13	OCTL12	OCTL11	OCTL10	OCTL9	OCTL8	OCTL7	OCTL6	OCTL5	OCTL4	OCTL3	OCTL2	OCTL1	OCTL0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

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

10.4.7. 端口位操作寄存器（GPIOx_BOP, 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
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

10.4.8. 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															LKK
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LK15	LK14	LK13	LK12	LK11	LK10	LK9	LK8	LK7	LK6	LK5	LK4	LK3	LK2	LK1	LK0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:17	保留	必须保持复位值。
16	LKK	锁定序列键 该位只能通过使用 Lock Key 写序列设置，始终可读。 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。
15:0	LKy	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

10.4.9. 备用功能选择寄存器 0 (GPIOx_AFSEL0, x=A...E)

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
SEL7[3:0]				SEL6[3:0]				SEL5[3:0]				SEL4[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEL3[3:0]				SEL2[3:0]				SEL1[3:0]				SEL0[3:0]			
rw				rw				rw				rw			

位/位域	名称	描述

31:28	SEL7[3:0]	Pin 7 备用功能选择 该位由软件置位和清除。 参考 SEL0 [3:0]的描述
27:24	SEL6[3:0]	Pin 6 备用功能选择 该位由软件置位和清除。 参考 SEL0 [3:0]的描述
23:20	SEL5[3:0]	Pin 5 备用功能选择 该位由软件置位和清除。 参考 SEL0 [3:0]的描述
19:16	SEL4[3:0]	Pin 4 备用功能选择 该位由软件置位和清除。 参考 SEL0 [3:0]的描述
15:12	SEL3[3:0]	Pin 3 备用功能选择 该位由软件置位和清除。 参考 SEL0 [3:0]的描述
11:8	SEL2[3:0]	Pin 2 备用功能选择 该位由软件置位和清除。 参考 SEL0 [3:0]的描述
7:4	SEL1[3:0]	Pin 1 备用功能选择 该位由软件置位和清除。 参考 SEL0 [3:0]的描述
3:0	SEL0[3:0]	Pin 0 备用功能选择 该位由软件置位和清除。 0000: 选择 AF0 功能（复位值） 0001: 选择 AF1 功能 0010: 选择 AF2 功能 0011: 选择 AF3 功能 0100: 选择 AF4 功能 0101: 选择 AF5 功能 0110: 选择 AF6 功能 0111: 选择 AF7 功能 1000: 选择 AF8 功能 ... 1001~1111: 保留

10.4.10. 备用功能选择寄存器 1 (GPIOx_AFSEL1, x=A...E)

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
SEL15[3:0]				SEL14[3:0]				SEL13[3:0]				SEL12[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEL11[3:0]				SEL10[3:0]				SEL9[3:0]				SEL8[3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:28	SEL15[3:0]	Pin 15 备用功能选择 该位由软件置位和清除。 参考 SEL8[3:0]的描述
27:24	SEL14[3:0]	Pin 14 备用功能选择 该位由软件置位和清除。 参考 SEL8[3:0]的描述
23:20	SEL13[3:0]	Pin 13 备用功能选择 该位由软件置位和清除。 参考 SEL8[3:0]的描述
19:16	SEL12[3:0]	Pin 12 备用功能选择 该位由软件置位和清除。 参考 SEL8[3:0]的描述
15:12	SEL11[3:0]	Pin 11 备用功能选择 该位由软件置位和清除。 参考 SEL8[3:0]的描述
11:8	SEL10[3:0]	Pin 10 备用功能选择 该位由软件置位和清除。 参考 SEL8[3:0]的描述
7:4	SEL9[3:0]	Pin 9 备用功能选择 该位由软件置位和清除。 参考 SEL8[3:0]的描述
3:0	SEL8[3:0]	Pin 8 备用功能选择 该位由软件置位和清除。 0000: 选择 AF0 功能 (复位值)

0001: 选择 AF1 功能
0010: 选择 AF2 功能
0011: 选择 AF3 功能
0100: 选择 AF4 功能
0101: 选择 AF5 功能
0110: 选择 AF6 功能
0111: 选择 AF7 功能
1000: 选择 AF8 功能
...
1001~1111: 保留

10.4.11. 位清除寄存器（GPIOx_BC, x=A...E）

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
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

10.4.12. EXTI 源选择寄存器 0 寄存器（AFIO_EXTISS0）

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器只能按字（32 位）访问。

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



EXTI3_SS [3:0]	EXTI2_SS [3:0]	EXTI1_SS [3:0]	EXTI0_SS [3:0]
rw	rw	rw	rw

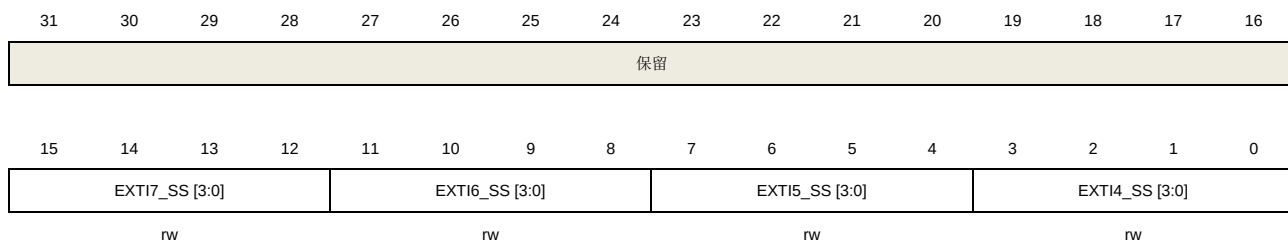
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI3_SS[3:0]	EXTI 3 源选择 0000: PA3 引脚 0001: PB3 引脚 0010: PC3 引脚 0011: PD3 引脚 0100: PE3 引脚 其他配置保留。
11:8	EXTI2_SS[3:0]	EXTI 2 源选择 0000: PA2 引脚 0001: PB2 引脚 0010: PC2 引脚 0011: PD2 引脚 0100: PE2 引脚 其他配置保留。
7:4	EXTI1_SS[3:0]	EXTI 1 源选择 0000: PA1 引脚 0001: PB1 引脚 0010: PC1 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。

10.4.13. EXTI 源选择寄存器 1 寄存器 (AFIO_EXTISS1)

地址偏移: 0x0C

复位值: 0x0000 0000

该寄存器只能按字（32 位）访问。



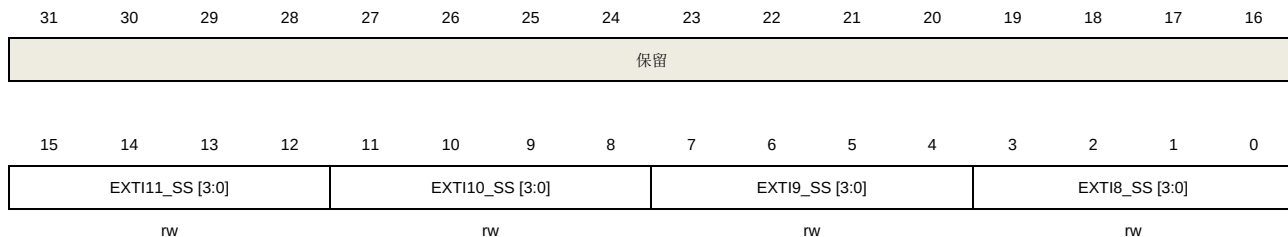
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI7_SS[3:0]	EXTI 7 源选择 0000: PA7 引脚 0001: PB7 引脚 0010: PC7 引脚 0011: PD7 引脚 0100: PE7 引脚 其他配置保留。
11:8	EXTI6_SS[3:0]	EXTI 6 源选择 0000: PA6 引脚 0001: PB6 引脚 0010: PC6 引脚 0011: PD6 引脚 0100: PE6 引脚 其他配置保留。
7:4	EXTI5_SS[3:0]	EXTI 5 源选择 0000: PA5 引脚 0001: PB5 引脚 0010: PC5 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。

10.4.14. EXTI 源选择寄存器 2 寄存器 (AFIO_EXTISS2)

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI11_SS[3:0]	EXTI 11 源选择 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

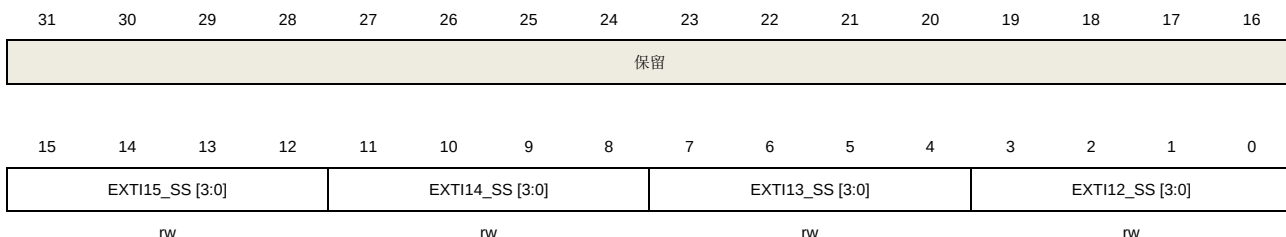
0010: PC8 引脚
 0011: PD8 引脚
 0100: PE8 引脚
 其他配置保留。

10.4.15. EXTI 源选择寄存器 3 寄存器 (AFIO_EXTISS3)

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
31:16	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。

11. 触发选择控制器（TRIGSEL）

11.1. 简介

触发选择控制器（TRIGSEL）可通过软件配置的方式，为各种外设选择触发输入信号。TRIGSEL 提供了灵活的机制，可以为外设选择不同的触发输入。

使用 TRIGSEL，每个外设最多可以配置 3 路 TRIGSEL 输出作为该外设的触发输入信号。配置相应的触发选择寄存器，可以为外设的指定触发输入选择不同的触发信号。

11.2. 主要特征

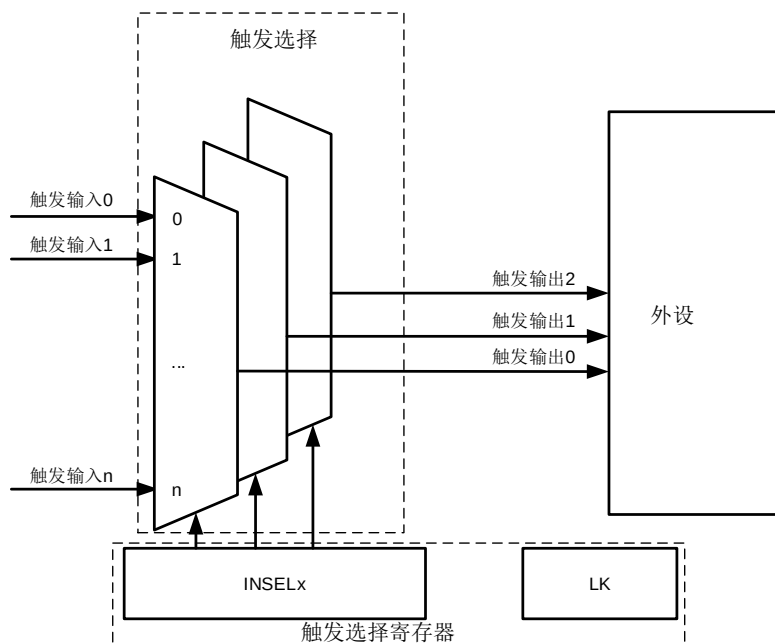
- 支持多达74个触发输入信号；
- 每个外设都有专用的触发信号选择寄存器；
- TRIGSEL寄存器最多可以配置3个输出到外围设备；
- 触发选择控制器的输入信号可来源于外部输入或外设输出；
- 触发选择控制器的输出信号可输出到外部输出或者到外设输入。

11.3. 功能说明

支持触发源选择的外设均具有专用 TRIGSEL 寄存器，用来为该外设选择不同的触发输入源。每个 TRIGSEL 寄存器可以配置 3 路输出，这些输出连接到外设的触发输入。每路输出均可从不同的触发输入源中选择。

[图 11-1. TRGSEL 主要组成示例](#)显示了 TRIGSEL 的主要组成结构。

图 11-1. TRIGSEL 主要组成示例



11.4. 内部连接

TRIGSEL 允许软件方式为外设选择触发输入。[表 11-1. 触发输入位域选择](#)给出了触发输入寄存器的位域值对应的触发输入选择。

表 11-1. 触发输入位域选择

位域名称	位域值	触发输入选择
INSELx	0x00	V _{SS}
	0x01	V _{DD}
	0x02	TRIGSEL_IN0
	0x03	TRIGSEL_IN1
	0x04	TRIGSEL_IN2
	0x05	TRIGSEL_IN3
	0x06	TRIGSEL_IN4
	0x07	TRIGSEL_IN5
	0x08	TRIGSEL_IN6
	0x09	TRIGSEL_IN7
	0x0a	TIMER0_TRGO
	0x0b	TIMER0_CH0
	0x0c	TIMER0_CH1
	0x0d	TIMER0_CH2

位域名称	位域值	触发输入选择
	0x0e	TIMER0_CH3
	0x0f	TIMER7_TRGO
	0x10	TIMER7_CH0
	0x11	TIMER7_CH1
	0x12	TIMER7_CH2
	0x13	TIMER7_CH3
	0x14	TIMER5_TRGO
	0x15	TIMER6_TRGO
	0x16	TIMER1_TRGO
	0x17	TIMER1_CH0
	0x18	TIMER1_CH1
	0x19	TIMER1_CH2
	0x1a	TIMER1_CH3
	0x1b	TIMER2_TRGO
	0x1c	TIMER2_CH0
	0x1d	TIMER2_CH1
	0x1e	TIMER2_CH2
	0x1f	TIMER2_CH3
	0x20	TIMER3_TRGO
	0x21	TIMER3_CH0
	0x22	TIMER3_CH1
	0x23	TIMER3_CH2
	0x24	TIMER3_CH3
	0x25	TIMER4_TRGO
	0x26	TIMER4_CH0
	0x27	TIMER4_CH1
	0x28	TIMER4_CH2
	0x29	TIMER4_CH3
	0x2f	TIMER15_TRGO
	0x30	TIMER15_CH0
	0x31	TIMER15_CH1
	0x32	TIMER15_MCH0
	0x33	TIMER16_TRGO
	0x34	TIMER16_CH0
	0x35	TIMER16_CH1
	0x36	TIMER16_MCH0
	0x37	ADC0_WD0_OUT

位域名称	位域值	触发输入选择
	0x38	ADC0_WD1_OUT
	0x39	ADC0_WD2_OUT
	0x3a	ADC1_WD0_OUT
	0x3b	ADC1_WD1_OUT
	0x3c	ADC1_WD2_OUT
	0x3d	ADC2_WD0_OUT
	0x3e	ADC2_WD1_OUT
	0x3f	ADC2_WD2_OUT
	0x40	CMP0_OUT
	0x41	CK_OUT
	0x42	TIMER0_BKIN
	0x43	TIMER0_CH0BKIN
	0x44	TIMER0_CH1BKIN
	0x45	TIMER0_CH2BKIN
	0x46	TIMER7_BKIN
	0x47	TIMER7_CH0BKIN
	0x48	TIMER7_CH1BKIN
	0x49	TIMER7_CH2BKIN
	0x4a	TIMER15_BKIN
	0x4b	TIMER16_BKIN
	0x4c	EXTI9
	0x4d	EXTI11
	0x4e	EXTI15

如[表 11-2. TRIGSEL 输入输出映射关系](#)所示，表明了 TRIGSEL 输入输出之间的连接关系。通过 TRIGSEL 寄存器的 INSELx[7:0]位域，可以给 TRIGSEL 的输出选择一个输入触发源。每个 TRIGSEL 寄存器可以配置 3 路输出，这些输出连接到对应的外设。

表 11-2. TRIGSEL 输入输出映射关系

触发源	触发选择	TRIGSEL 寄存器	TRIGSEL 输出	外设
0	INSELx[7:0]	TRIGSEL_EXTOUT0	输出 0	TRIGSEL_OUT0
1			输出 1	TRIGSEL_OUT1
TRIGSEL_IN0		TRIGSEL_EXTOUT1	输出 0 输出 1	TRIGSEL_OUT2 TRIGSEL_OUT3
TRIGSEL_IN1				
TRIGSEL_IN2				
TRIGSEL_IN3				
TRIGSEL_IN4				
TRIGSEL_IN5				

触发源	触发选择	TRIGSEL 寄存器	TRIGSEL 输出	外设
TRIGSEL_IN6 TRIGSEL_IN7 TIMER0_TRGO TIMER0_CH0 TIMER0_CH1 TIMER0_CH2 TIMER0_CH3 TIMER7_TRGO TIMER7_CH0 TIMER7_CH1 TIMER7_CH2 TIMER7_CH3 TIMER5_TRGO TIMER6_TRGO TIMER1_TRGO TIMER1_CH0 TIMER1_CH1 TIMER1_CH2 TIMER1_CH3 TIMER2_TRGO TIMER2_CH0 TIMER2_CH1 TIMER2_CH2 TIMER2_CH3 TIMER3_TRGO TIMER3_CH0 TIMER3_CH1 TIMER3_CH2 TIMER3_CH3 TIMER4_TRGO TIMER4_CH0 TIMER4_CH1 TIMER4_CH2 TIMER4_CH3 TIMER15_TRGO TIMER15_CH0		TRIGSEL_EXTOUT2	输出 0 输出 1	TRIGSEL_OUT4 TRIGSEL_OUT5
		TRIGSEL_EXTOUT3	输出 0 输出 1	TRIGSEL_OUT6 TRIGSEL_OUT7
		TRIGSEL_TIMER0ITI	输出 0	TIMER0_ITI
		TRIGSEL_TIMER1ITI	输出 0	TIMER1_ITI
		TRIGSEL_TIMER2ITI	输出 0	TIMER2_ITI
		TRIGSEL_TIMER3ITI	输出 0	TIMER3_ITI
		TRIGSEL_TIMER4ITI	输出 0	TIMER4_ITI
		TRIGSEL_TIMER7ITI	输出 0	TIMER7_ITI
		TRIGSEL_TIMER15ITI	输出 0	TIMER15_ITI

触发源	触发选择	TRIGSEL 寄存器	TRIGSEL 输出	外设
TIMER15_CH1 TIMER15_MCH0 TIMER16_TRGO TIMER16_CH0 TIMER16_CH1 TIMER16_MCH0 ADC0_WD0_OUT ADC0_WD1_OUT ADC0_WD2_OUT ADC1_WD0_OUT ADC1_WD1_OUT ADC1_WD2_OUT ADC2_WD0_OUT ADC2_WD1_OUT ADC2_WD2_OUT CMP0_OUT CK_OUT TIMER0_BKIN TIMER0_CH0BKIN TIMER0_CH1BKIN TIMER0_CH2BKIN TIMER7_BKIN TIMER7_CH0BKIN TIMER7_CH1BKIN TIMER7_CH2BKIN TIMER15_BKIN TIMER16_BKIN EXTI9 EXTI11 EXTI15		TRIGSEL_TIMER16I TI	输出 0	TIMER16_ITI
		TRIGSEL_DAC	输出 0	DAC0_OUT_EXTRG
		TRIGSEL_ADC0_RO UTRG	输出 0	ADC0_ROUTRG
		TRIGSEL_ADC0_INS TRG	输出 0	ADC0_INSTRG
		TRIGSEL_ADC1_RO UTRG	输出 0	ADC1_ROUTRG
		TRIGSEL_ADC1_INS TRG	输出 0	ADC1_INSTRG
		TRIGSEL_ADC2_RO UTRG	输出 0	ADC2_ROUTRG
		TRIGSEL_ADC2_INS TRG	输出 0	ADC2_INSTRG
		TRIGSEL_TIMER0BR KIN	输出 0	TIMER0_BRKIN

触发源	触发选择	TRIGSEL 寄存器	TRIGSEL 输出	外设
		TRIGSEL_TIMER0C HBRKIN	输出 0 输出 1 输出 2	TIMER0_CH0BRKIN TIMER0_CH1BRKIN TIMER0_CH2BRKIN
		TRIGSEL_TIMER7BR KIN	输出 0	TIMER7_BRKIN
		TRIGSEL_TIMER7C HBRKIN	输出 0 输出 1 输出 2	TIMER7_CH0BRKIN TIMER7_CH1BRKIN TIMER7_CH2BRKIN
		TRIGSEL_TIMER15B RKIN	输出 0	TIMER15_BRKIN
		TRIGSEL_TIMER16B RKIN	输出 0	TIMER16_BRKIN

注意：

1. TRIGSEL_OUTx (x=0,...,7) 不能选择任何 TRIGSEL_INy (y=0,...,7) 作为触发源。
2. 所有 TIMERx_ITI (x=0,...,4,7, 15,16) 只能选择来自TIMER和CK_OUT的触发输入源.其中, TIMERx_ITI (x= 0,...,4,7) 不能选择来自其自身模块的信号 (TIMERx_TRGO \ TIMERx_CH0 \ TIMERx_CH1 \ TIMERx_CH2 \ TIMERx_CH3) 作为触发源, TIMERx_ITI (x= 15,16) 不能选择来自其自身模块的信号 (TIMERx_TRGO \ TIMERx_CH0 \ TIMERx_CH1 \ TIMERx_MCH0) 作为触发源。
3. 除1、2所述之外, 所有输出都可以选择所有输入作为触发源。

触发输入选择 INSELx[7:0]位域值配置为 0x00 时, TRIGSEL 触发输入选择为低电平; 配置为 0x01

时，TRIGSEL 触发输入选择为高电平。当选择了非法触发输入时，其输出将被强制选择为 0。

11.5. TRIGSEL 寄存器

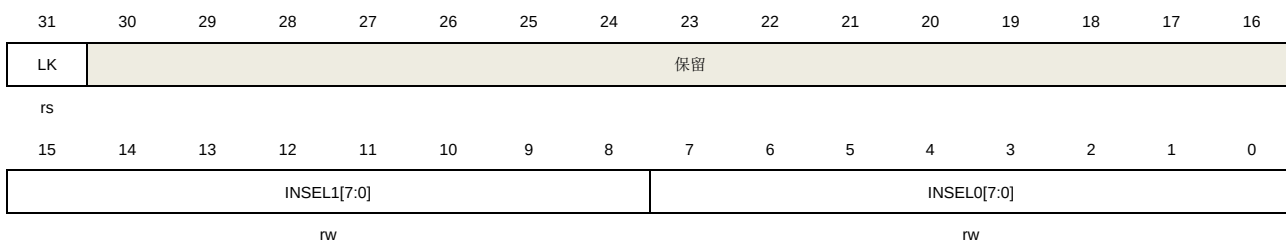
TRIGSEL 基地址：0x4001 4400

11.5.1. EXTOUT 触发选择寄存器 0 (TRIGSEL_EXTOUT_0)

地址偏移：0x00

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



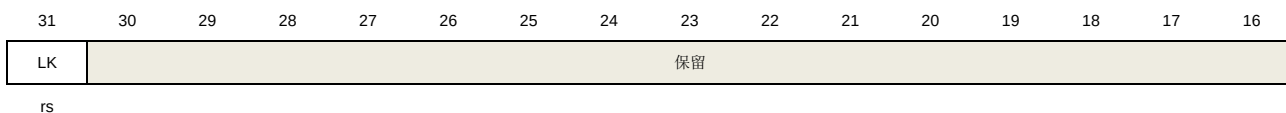
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_EXTOUT_0将不能被修改。 0: TRIGSEL_EXTOUT_0 寄存器可写可读 1: TRIGSEL_EXTOUT_0 寄存器只读
30:16	保留	必须保持复位值。
15:8	INSEL1[7:0]	触发输出 1 的输入源选择 这些位用来选择连接到输出 1 的触发输入信号，输出 1 作为 TRIGSEL_OUT1（外部输出 1）的信号源。关于具体配置请参考 表 11-1. 触发输入位域选择 。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TRIGSEL_OUT0（外部输出 0）的信号源。关于具体配置请参考 表 11-1. 触发输入位域选择 。

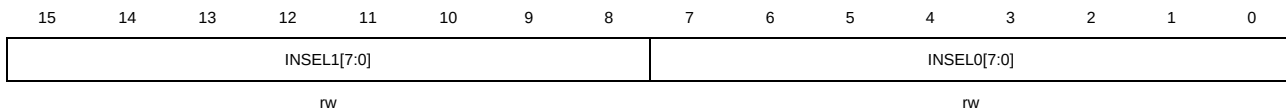
11.5.2. EXTOUT 触发选择寄存器 1 (TRIGSEL_EXTOUT_1)

地址偏移：0x04

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。





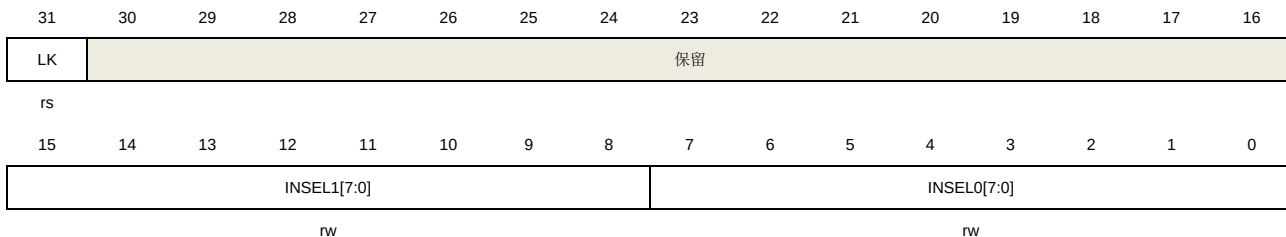
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_EXTOUT_1将不能被修改。 0: TRIGSEL_EXTOUT_1 寄存器可写可读 1: TRIGSEL_EXTOUT_1 寄存器只读
30:16	保留	必须保持复位值。
15:8	INSEL1[7:0]	触发输出 1 的输入源选择 这些位用来选择连接到输出 1 的触发输入信号，输出 1 作为 TRIGSEL_OUT3（外部输出 3）的信号源。关于具体配置请参考表 11-1. 触发输入位域选择。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TRIGSEL_OUT2（外部输出 2）的信号源。关于具体配置请参考表 11-1. 触发输入位域选择。

11.5.3. EXTOUT 触发选择寄存器 2（TRIGSEL_EXTOUT_2）

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	LK	TRIGSEL 锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_EXTOUT_2将不能被修改。 0: TRIGSEL_EXTOUT_2 寄存器可读可写 1: TRIGSEL_EXTOUT_2 寄存器只读
30:16	保留	必须保持复位值。
15:8	INSEL1[7:0]	触发输出 1 的输入源选择

这些位用来选择连接到输出 1 的触发输入信号，输出 1 作为 TRIGSEL_OUT5（外部输出 5）的信号源。关于具体配置请参考[表 11-1. 触发输入位域选择](#)。

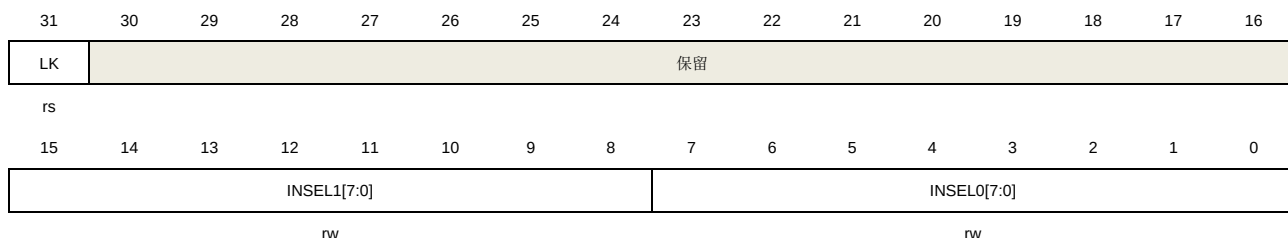
7:0 INSEL0[7:0] 触发输出 0 的输入源选择
这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TRIGSEL_OUT4（外部输出 4）的信号源。关于具体配置请参考[表 11-1. 触发输入位域选择](#)。

11.5.4. EXTOUT 触发选择寄存器 3（TRIGSEL_EXTOUT_3）

地址偏移：0x0C

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



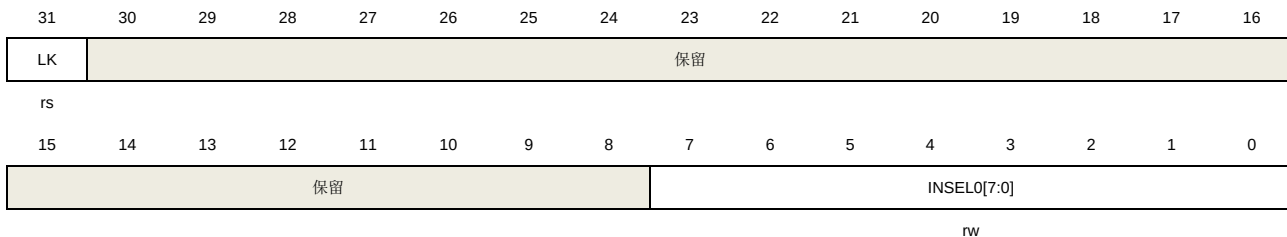
位/位域	名称	描述
31	LK	TRIGSEL 锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_EXTOUT_3将不能被修改。 0: TRIGSEL_EXTOUT_3 寄存器可读可写 1: TRIGSEL_EXTOUT_3 寄存器只读
30:16	保留	必须保持复位值。
15:8	INSEL1[7:0]	触发输出 1 的输入源选择 这些位用来选择连接到输出 1 的触发输入信号，输出 1 作为 TRIGSEL_OUT7（外部输出 7）的信号源。关于具体配置请参考 表 11-1. 触发输入位域选择 。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TRIGSEL_OUT6（外部输出 6）的信号源。关于具体配置请参考 表 11-1. 触发输入位域选择 。

11.5.5. TIMER0_ITI 触发选择寄存器（TRIGSEL_TIMER0ITI）

地址偏移：0x10

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



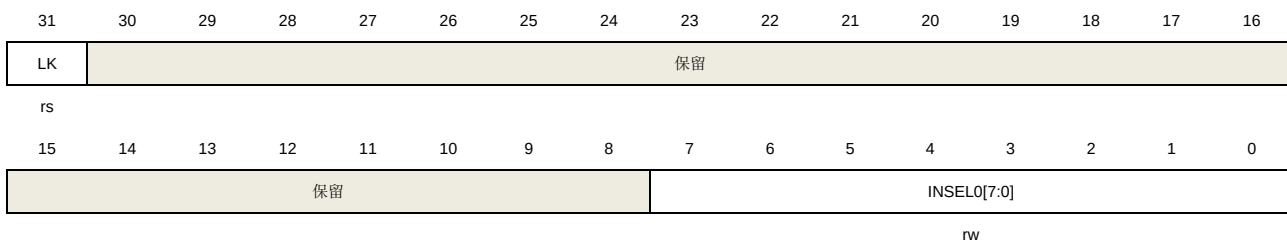
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER0ITI将不能被修改。 0: TRIGSEL_TIMER0ITI 寄存器可读可写 1: TRIGSEL_TIMER0ETI 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER0_ETI 的触发源。 关于具体配置请参考表 11-1. 触发输入位域选择。

11.5.6. TIMER1_ITI 触发选择寄存器（TRIGSEL_TIMER1ITI）

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	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER1ITI将不能被修改。 0: TRIGSEL_TIMER1ITI 寄存器可读可写 1: TRIGSEL_TIMER1ITI 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择

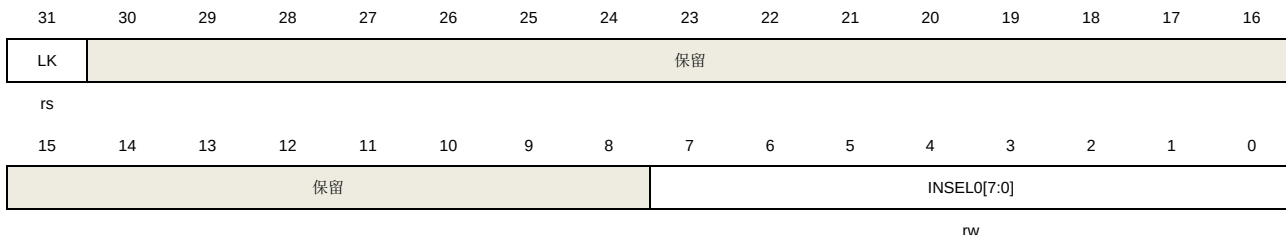
这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER1_ITI 的触发源。
关于具体配置请参考[表 11-1. 触发输入位域选择](#)。

11.5.7. TIMER2_ITI 触发选择寄存器 (TRIGSEL_TIMER2ITI)

地址偏移：0x18

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



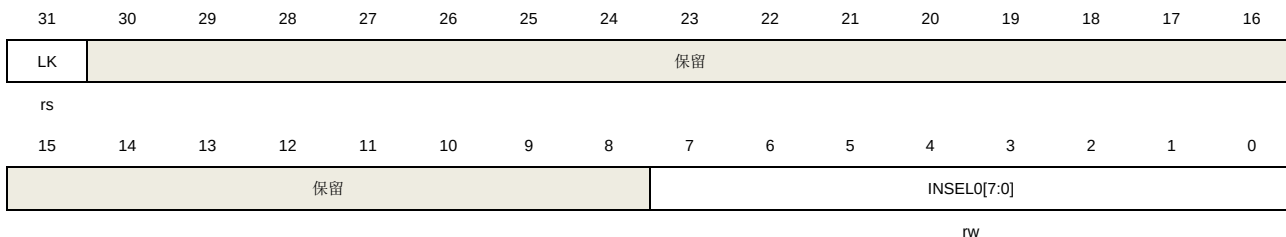
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER2ITI 将不能被修改。 0: TRIGSEL_TIMER2ITI 寄存器可读可写 1: TRIGSEL_TIMER2ITI 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER2_ITI 的触发源。 关于具体配置请参考 表 11-1. 触发输入位域选择 。

11.5.8. TIMER3_ITI 触发选择寄存器 (TRIGSEL_TIMER3ITI)

地址偏移：0x1C

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。





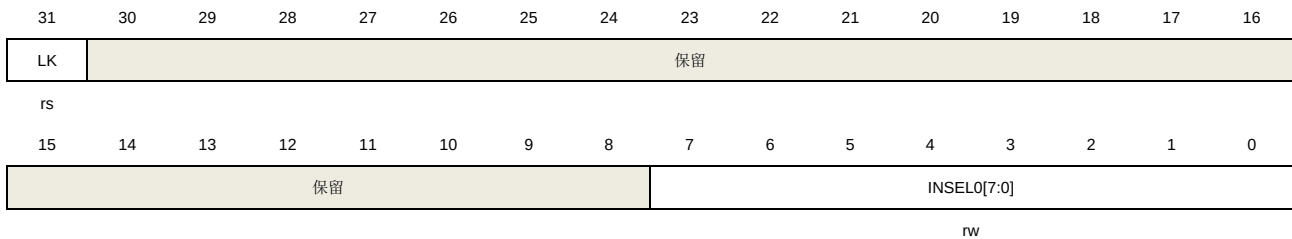
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER2ITI将不能被修改。 0: TRIGSEL_TIMER3ITI 寄存器可读可写 1: TRIGSEL_TIMER3ITI 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER3_ITI 的触发源。 关于具体配置请参考 表 11-1. 触发输入位域选择 。

11.5.9. TIMER4_ITI 触发选择寄存器 (TRIGSEL_TIMER4ITI)

地址偏移: 0x20

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



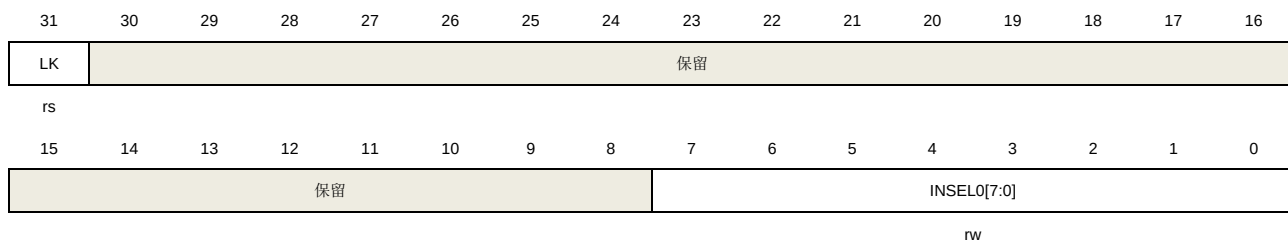
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER4ITI将不能被修改。 0: TRIGSEL_TIMER4ITI 寄存器可读可写 1: TRIGSEL_TIMER4ITI 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER4_ITI 的触发源。 关于具体配置请参考 表 11-1. 触发输入位域选择 。

11.5.10. TIMER7_ITI 触发选择寄存器 (TRIGSEL_TIMER7ITI)

地址偏移: 0x24

复位值: 0x0000 0000

该寄存器只能按字（32 位）访问。



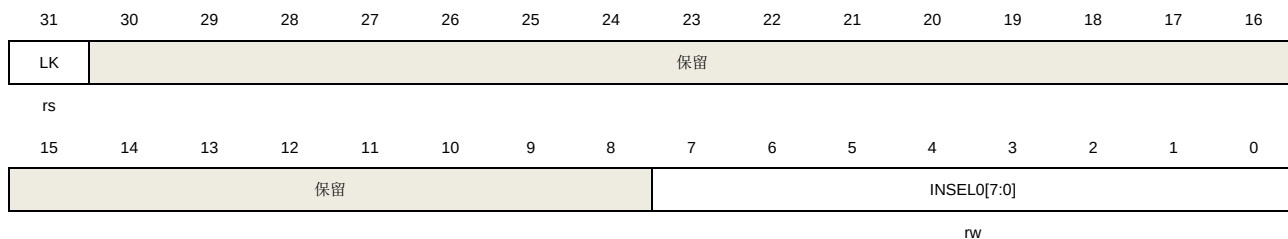
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER7ITI将不能被修改。 0: TRIGSEL_TIMER7ITI 寄存器可读可写 1: TRIGSEL_TIMER7ITI 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER7_ITI 的触发源。 关于具体配置请参考表 11-1. 触发输入位域选择。

11.5.11. TIMER15_ITI 触发选择寄存器（TRIGSEL_TIMER15ITI）

地址偏移：0x2C

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER15ITI将不能被修改。 0: TRIGSEL_TIMER15ITI 寄存器可读可写 1: TRIGSEL_TIMER15ITI 寄存器只读
30:8	保留	必须保持复位值。

7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER15_ITI 的触发源。关于具体配置请参考 表 11-1. 触发输入位域选择 。

11.5.12. TIMER16_ITI 触发选择寄存器（TRIGSEL_TIMER16ITI）

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
LK	保留														
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								INSEL0[7:0]							
rw															

位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER16ITI 将不能被修改。 0: TRIGSEL_TIMER16ITI 寄存器可读可写 1: TRIGSEL_TIMER16ITI 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER16_ITI 的触发源。关于具体配置请参考 表 11-1. 触发输入位域选择 。

11.5.13. DAC 触发选择寄存器（TRIGSEL_DAC）

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
LK	保留														
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								INSEL0[7:0]							
rw															

位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_DAC将不能被修改。 0: TRIGSEL_DAC 寄存器可读可写 1: TRIGSEL_DAC 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 DAC_OUT_EXTRIG (DAC_OUT 外部触发)的触发源。关于具体配置请参考 表 11-1. 触发输入位域选择 。

11.5.14. ADC0_ROUTRG 触发选择寄存器（TRIGSEL_ADC0_ROUTRG）

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
LK	保留														
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								INSEL0[7:0]							
rw															

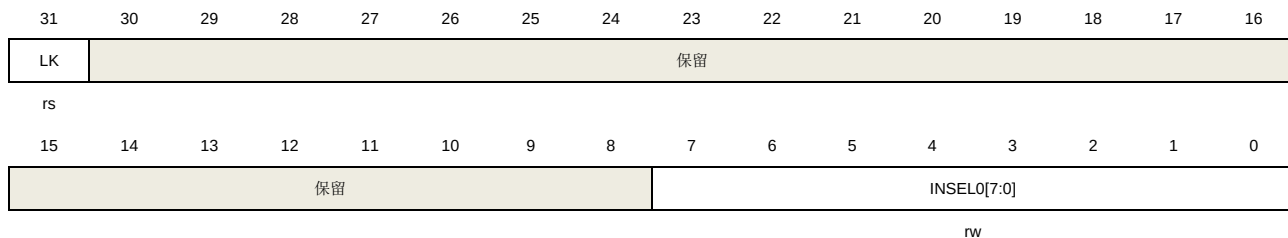
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_ADC0_ROUTRG 将不能被修改。 0: TRIGSEL_ADC0_ROUTRG 寄存器可读可写 1: TRIGSEL_ADC0_ROUTRG 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TRIGSEL_ADC0_ROUTRG 的触发源。关于具体配置请参考 表 11-1. 触发输入位域选择 。

11.5.15. ADC0_INSTRG 触发选择寄存器（TRIGSEL_ADC0_INSTRG）

地址偏移：0x3C

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



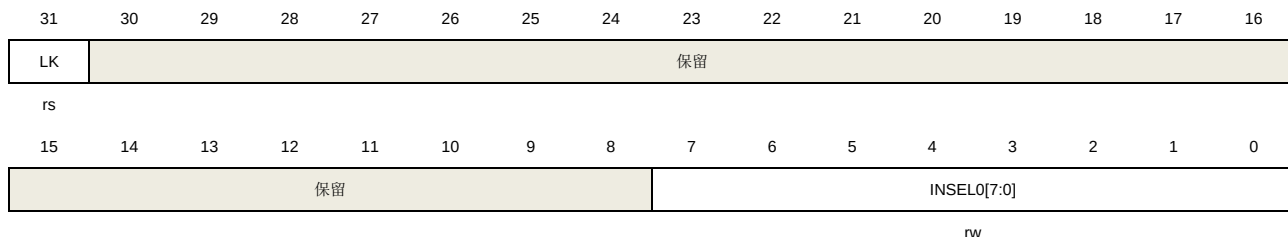
位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_ADC0_INSTRG将不能被修改。 0: TRIGSEL_ADC0_INSTRG 寄存器可读可写 1: TRIGSEL_ADC0_INSTRG 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 ADC0_INSTRG 的触发源。关于具体配置请参考 表 11-1. 触发输入位域选择 。

11.5.16. ADC1_ROUTRG 触发选择寄存器（TRIGSEL_ADC1_ROUTRG）

地址偏移：0x40

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	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_ADC1_ROUTRG将不能被修改。 0: TRIGSEL_ADC1_ROUTRG 寄存器可读可写 1: TRIGSEL_ADC1_ROUTRG 寄存器只读

30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 ADC1_ROUTRG 的触发源。关于具体配置请参考 表 11-1. 触发输入位域选择 。

11.5.17. ADC1_INSTRG 触发选择寄存器（TRIGSEL_ADC1_INSTRG）

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
LK	保留														
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								INSEL0[7:0]							
rw															

位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_ADC1_INSTRG 将不能被修改。 0: TRIGSEL_ADC1_INSTRG 寄存器可读可写 1: TRIGSEL_ADC1_INSTRG 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 ADC1_INSTRG 的触发源。关于具体配置请参考 表 11-1. 触发输入位域选择 。

11.5.18. ADC2_ROUTRG 触发选择寄存器（TRIGSEL_ADC2_ROUTRG）

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
LK	保留														
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								INSEL0[7:0]							

rw

位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_ADC2_ROUTRG 将不能被修改。 0: TRIGSEL_ADC2_ROUTRG 寄存器可读可写 1: TRIGSEL_ADC2_ROUTRG 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 ADC2_ROUTRG 的触发源。关于具体配置请参考 表 11-1. 触发输入位域选择 。

11.5.19. ADC2_INSTRG 触发选择寄存器（TRIGSEL_ADC2_INSTRG）

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
LK	保留														
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								INSEL0[7:0]							
rw															

位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_ADC2_INSTRG 将不能被修改。 0: TRIGSEL_ADC2_INSTRG 寄存器可读可写 1: TRIGSEL_ADC2_INSTRG 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 ADC2_INSTRG 的触发源。关于具体配置请参考 表 11-1. 触发输入位域选择 。

11.5.20. TIMER0_BRKIN 触发选择寄存器 (TRIGSEL_TIMER0BRKIN)

地址偏移: 0x50

复位值: 0x0000 0042

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
LK	保留														
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								INSEL0[7:0]							
rw															

位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER0BRKIN将不能被修改。 0: TRIGSEL_TIMER0BRKIN 寄存器可读可写 1: TRIGSEL_TIMER0BRKIN 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER0_BRINK 的触发源。关于具体配置请参考 表 11-1. 触发输入位域选择 。

11.5.21. TIMER0_CHBRKIN 触发选择寄存器 (TRIGSEL_TIMER0CHBRKIN)

地址偏移: 0x54

复位值: 0x0045 4443

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
LK	保留							INSEL2[7:0]							
rs								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INSEL1[7:0]								INSEL0[7:0]							
rw								rw							

位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER0CHBRKIN将不能被修改。

		0: TRIGSEL_TIMER0BRKIN 寄存器可读可写 1: TRIGSEL_TIMER0BRKIN 寄存器只读
30:24	保留	必须保持复位值。
23:16	INSEL2[7:0]	触发输出 2 输入源选择 这些位用来选择连接到输出 2 的触发输入信号, 输出 2 作为 TIMER0_CH2BRINK 的触发源。关于具体配置请参考 表 11-1. 触发输入位域选择 。
15:8	INSEL1[7:0]	触发输出 1 输入源选择 这些位用来选择连接到输出 1 的触发输入信号, 输出 1 作为 TIMER0_CH1BRINK 的触发源。关于具体配置请参考 表 11-1. 触发输入位域选择 。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号, 输出 0 作为 TIMER0_CH0BRINK 的触发源。关于具体配置请参考 表 11-1. 触发输入位域选择 。

11.5.22. TIMER7_BRKIN 触发选择寄存器 (TRIGSEL_TIMER7BRKIN)

地址偏移: 0x58

复位值: 0x0000 0046

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
LK	保留														
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								INSEL0[7:0]							
rw															

位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位, 通过系统复位清除。该位置位后, TRIGSEL_TIMER7BRKIN 将不能被修改。 0: TRIGSEL_TIMER7BRKIN 寄存器可读可写 1: TRIGSEL_TIMER7BRKIN 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号, 输出 0 作为 TIMER7_BRINK 的触发源。关于具体配置请参考 表 11-1. 触发输入位域选择 。

11.5.23. TIMER7_CHBRKIN 触发选择寄存器 (TRIGSEL_TIMER7CHBRKIN)

地址偏移: 0x5C

复位值: 0x0049 4847

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
LK	保留							INSEL2[7:0]							
rs								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INSEL1[7:0]								INSEL0[7:0]							
rw								rw							

位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位, 通过系统复位清除。该位置位后, TRIGSEL_TIMER7CHBRKIN 将不能被修改。 0: TRIGSEL_TIMER7BRKIN 寄存器可读可写 1: TRIGSEL_TIMER7BRKIN 寄存器只读
30:24	保留	必须保持复位值。
23:16	INSEL2[7:0]	触发输出 2 输入源选择 这些位用来选择连接到输出 2 的触发输入信号, 输出 2 作为 TIMER7_CH2BRINK 的触发源。关于具体配置请参考 表 11-1. 触发输入位域选择 。
15:8	INSEL1[7:0]	触发输出 1 输入源选择 这些位用来选择连接到输出 1 的触发输入信号, 输出 1 作为 TIMER7_CH1BRINK 的触发源。关于具体配置请参考 表 11-1. 触发输入位域选择 。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号, 输出 0 作为 TIMER7_CH0BRINK 的触发源。关于具体配置请参考 表 11-1. 触发输入位域选择 。

11.5.24. TIMER15_BRKIN 触发选择寄存器 (TRIGSEL_TIMER15BRKIN)

地址偏移: 0x60

复位值: 0x0000 004A

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

保留	INSEL0[7:0]
rw	

位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER15BRKIN将不能被修改。 0: TRIGSEL_TIMER15BRKIN 寄存器可读可写 1: TRIGSEL_TIMER15BRKIN 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER15_BRINK 的触发源。关于具体配置请参考 表 11-1. 触发输入位域选择 。

11.5.25. TIMER16_BRKIN 触发选择寄存器（TRIGSEL_TIMER16BRKIN）

地址偏移：0x64

复位值：0x0000 004B

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
LK	保留														
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								INSEL0[7:0]							
rw															

位/位域	名称	描述
31	LK	TRIGSEL 寄存器锁定标志位 该位通过软件置位，通过系统复位清除。该位置位后，TRIGSEL_TIMER16BRKIN将不能被修改。 0: TRIGSEL_TIMER16BRKIN 寄存器可读可写 1: TRIGSEL_TIMER16BRKIN 寄存器只读
30:8	保留	必须保持复位值。
7:0	INSEL0[7:0]	触发输出 0 的输入源选择 这些位用来选择连接到输出 0 的触发输入信号，输出 0 作为 TIMER16_BRINK 的触发源。关于具体配置请参考 表 11-1. 触发输入位域选择 。

12. 定时器（TIMER）

表 12-1. 定时器（TIMERx）分为 4 种类型

定时器	定时器 0/7	定时器 1/2/3/4	定时器 15/16	定时器 5/6
类型	高级	通用 L0	通用 L3	基本
预分频器	16 位	16 位	16 位	16 位
计数器	16 位	16 位（TIMER2/3/4） 32 位（TIMER1）	16 位	16 位
计数模式	向上，向下，中央对齐	向上，向下，中央对齐	向上，向下，中央对齐	只有向上
可重复性	●	×	●	×
捕获/比较通道数	4	4	3	0
复合 PWM 模式	●	×	●	×
非对称 PWM 模式	●	×	×	×
输出匹配脉冲选择	●	×	●	×
互补和死区时间	●	×	●	×
中止功能	BREAK	●	×	×
	通道中止	●	×	×
单脉冲	●	●	●	●
正交译码器	●	●	×	×
译码器	×	●	×	×
主-从管理	●	●	●	×
内部连接	● ⁽¹⁾	● ⁽²⁾	● ⁽³⁾	TRGO TO DAC
同步和初始值、计数方向刷新	●	×	●	×
DMA	●	●	●	● ⁽⁴⁾
Debug 模式	●	●	●	●

	TIMERx	ITI0	ITI1	ITI2	ITI3	ITI14
(1)	TIMER0	TIMER4_TRGO	TIMER1_TRGO	TIMER2_TRGO	TRIGSEL	-
	TIMER7	TIMER0_TRGO	TIMER1_TRGO	TIMER3_TRGO		

	TIMERx	ITI0	ITI1	ITI2	ITI3	ITI14
(2)	TIMER1	TIMER0_TRGO	TIMER7_TRGO	TIMER2_TRGO		
	TIMER2	TIMER0_TRGO	TIMER1_TRGO	TIMER4_TRGO		
	TIMER3	TIMER0_TRGO	TIMER1_TRGO	TIMER2_TRGO		
	TIMER4	TIMER1_TRGO	TIMER2_TRGO	TIMER3_TRGO		
(3)	TIMER15	TIMER0_TRGO	TIMER2_TRGO	TIMER3_TRGO	TIMER16_CH0	TRIGS
	TIMER16	TIMER0_TRGO	TIMER2_TRGO	TIMER15_CH0	TIMER7_TRGO	EL
(4)	只有更新事件可以产生DMA请求。但是定时器5/6中没有DMA配置DMAS（DMA请求源选择）。					

12.1. 高级定时器 (TIMERx, x=0, 7)

12.1.1. 简介

高级定时器 (TIMER0/7) 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，可以用于计数，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。

12.1.2. 主要特性

- 总通道数：4；
- 计数器宽度：16位；
- 时钟源可选：内部时钟，内部触发，外部输入，外部触发；
- 多种计数模式：向上计数，向下计数和中央计数；
- 正交译码器接口：用来追踪运动和分辨旋转方向和位置；
- 霍尔传感器接口：用来进行三相电机控制；
- 可编程的预分频器：16位，运行时可以被改变；
- 每个通道相互独立且可配置：输入捕获模式，输出比较模式，可编程的PWM模式，单脉冲模式和触发输出；
- 可编程的死区时间和独立的死区时间配置；
- 自动重装载功能；
- 可编程的计数器重复功能；
- 中止输入功能：BREAK和通道0/1/2中止；
- 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-从管理。

12.1.3. 结构框图

[图 12-1. 高级定时器结构框图](#)提供了高级定时器的内部配置细节。

[illegible]

时钟源选择

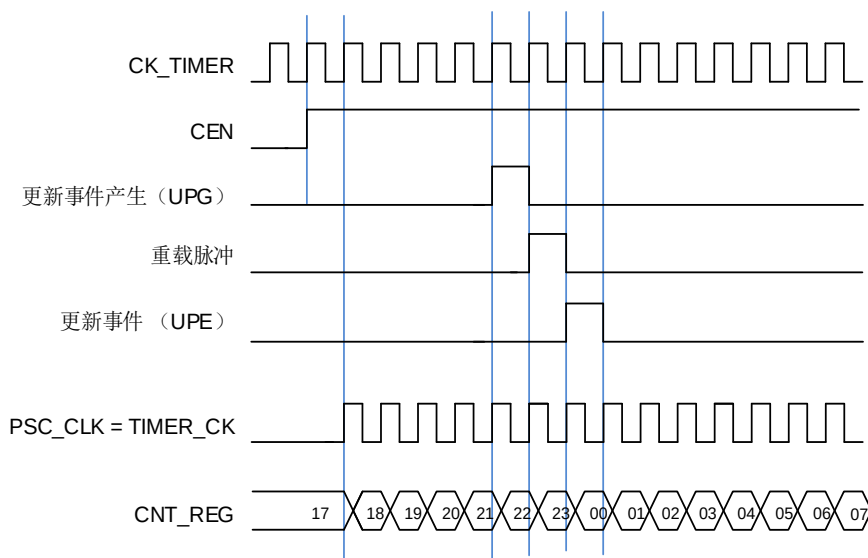
- 当SYSCFG_TIMERxCFG(x=0,7)寄存器中的TSCFGy[4:0]=5'b00000(y=0...6)时,定时器选择内部时钟源(连接到RCU模块的CK_TIMER)

如果 SYSCFG_TIMERxCFG(x=0,7)寄存器中的 TSCFGy[4:0] =5'b00000(y=0...6)，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 SYSCFG_TIMERxCFG(x=0,7)寄存器中的 TSCFGy[4:0] (y=0..2,6)位域设置为非零值, 预分频器被其他时钟源驱动, 具体在下文说明。当 TSCFGy[4:0] (y=3,4,5)被设置为非零值时, 计数器预分频器时钟源由内部时钟 **TIMER CK** 驱动。

图 12-2. 内部时钟分频为 1 时正常模式下的控制电路



■ TSCFG6[4:0] != 5'b00000 (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_CHn` ($n=0..3$) 引脚的每个上升沿或下降沿计数。这种模式可以通过设置 `TSCFG6[4:0]` 为 `0x5~0x7` 和 `0x9/0xA` 来选择。

计数器预分频器也可以在内部触发信号 `ITI0/1/2/3` 的上升沿计数。这种模式可以通过设置 `TSCFG6[4:0]` 为 `0x1~0x4` 来选择。

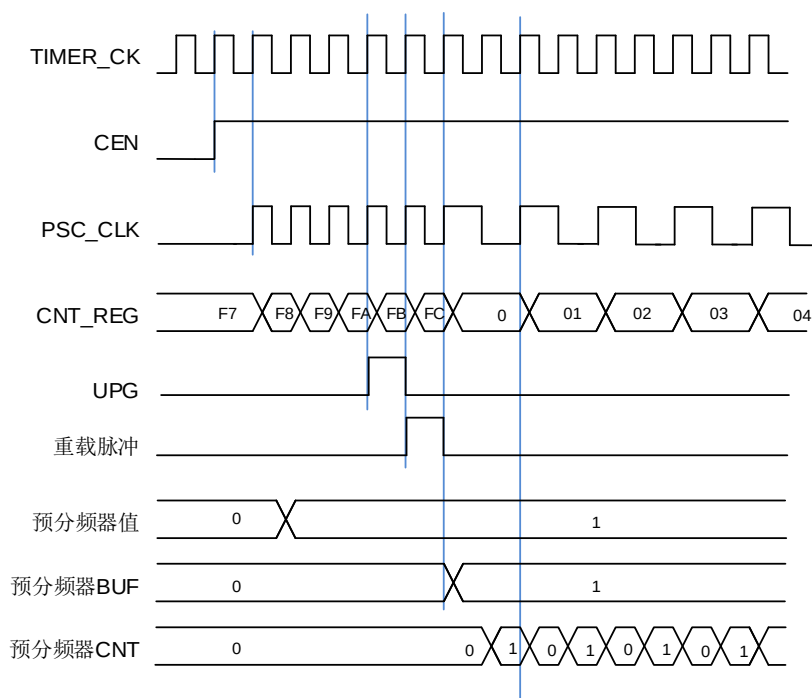
■ SMC1=1'b1 (外部时钟模式1)，定时器选择外部输入引脚ETI作为时钟源

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 `TSCFG6[4:0]` 为 `0x8`。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。

时钟预分频器

预分频器可以将定时器的时钟 (`TIMER_CLK`) 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 12-3. 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_CREP0+1)` 次上溢后产生更新事件，否则在每次上溢时都会产生更新事件。在向上计数模式中，`TIMERx_CTL0` 寄存器中的计数方向控制位 `DIR` 应该被设置成 0。

当通过 `TIMERx_SWEVG` 寄存器的 `UPG` 位置 1 来设置更新事件时，计数值会被清 0，并产生更新事件。

如果 `TIMERx_CTL0` 寄存器的 `UPDIS` 置 1，则禁止更新事件。

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

[图 12-4. 向上计数时序图, PSC=0/2](#) 和 [图 12-5. 向上计数时序图, 在运行时改变 `TIMERx\_CAR` 寄存器的值](#)给出一些例子，当 `TIMERx_CAR=0x99` 时，计数器在不同预分频因子下的行为。

图 12-4. 向上计数时序图，PSC=0/2

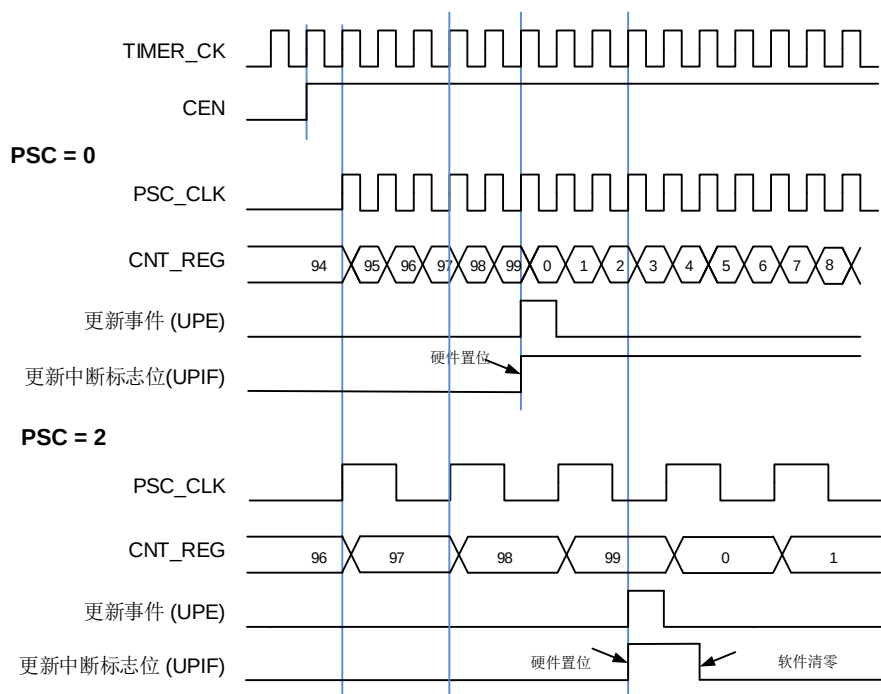
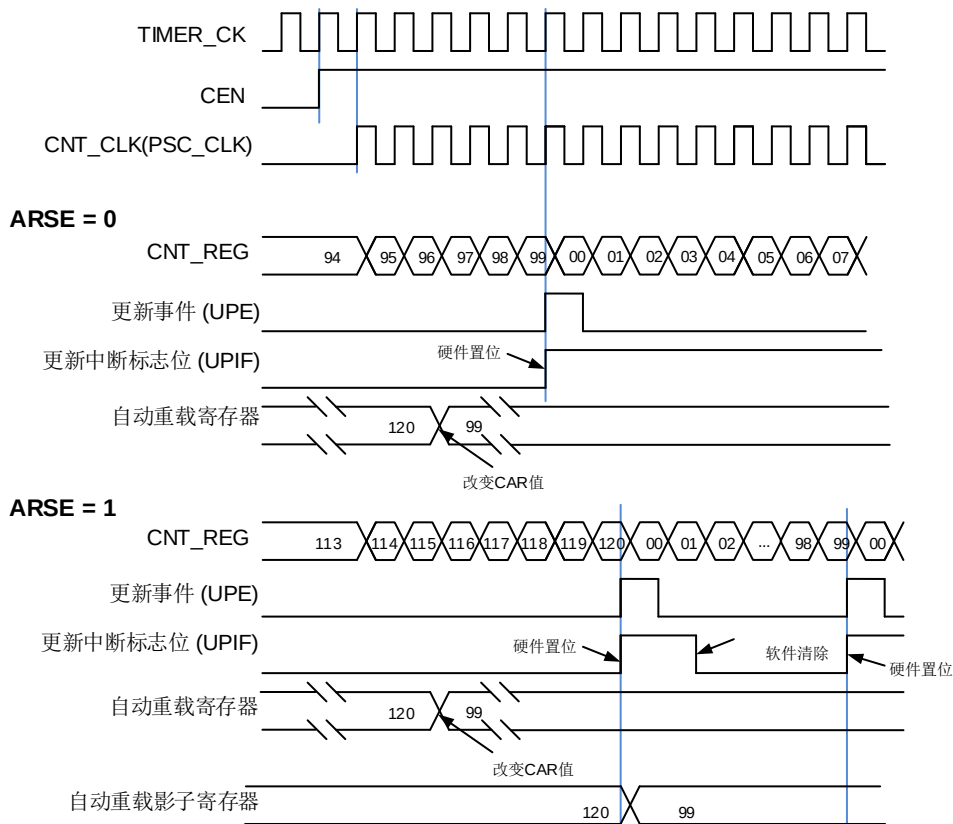


图 12-5. 向上计数时序图，在运行时改变 **TIMERx_CAR** 寄存器的值



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_CREP0+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 12-6. 向下计数时序图, `PSC=0/2`](#) 和 [图 12-7. 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=0x99` 时，计数器在不同时钟频率下的行为。

图 12-6. 向下计数时序图，`PSC=0/2`

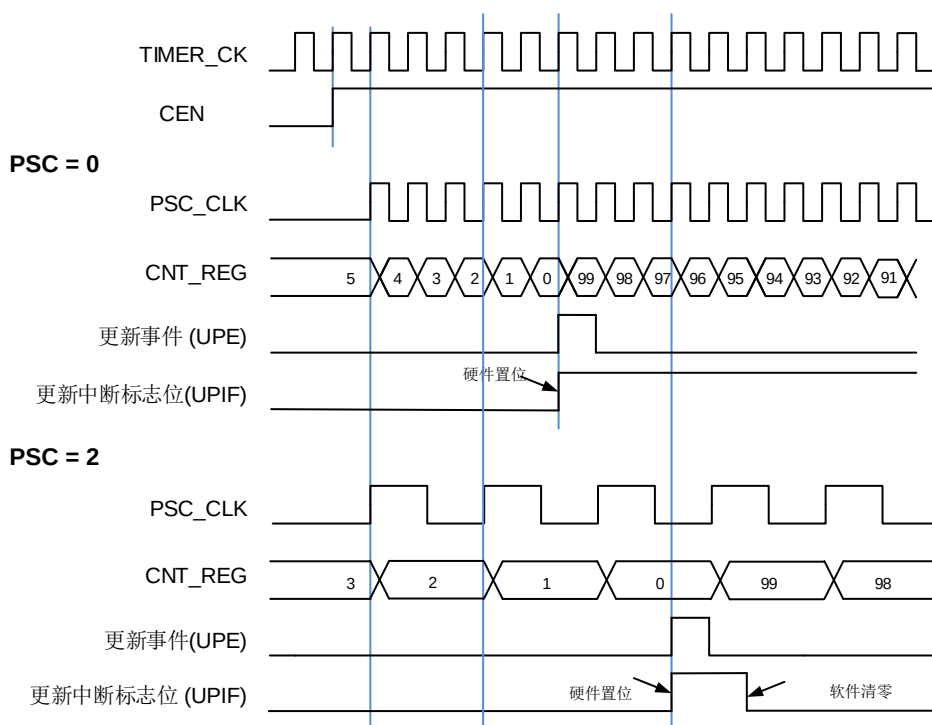
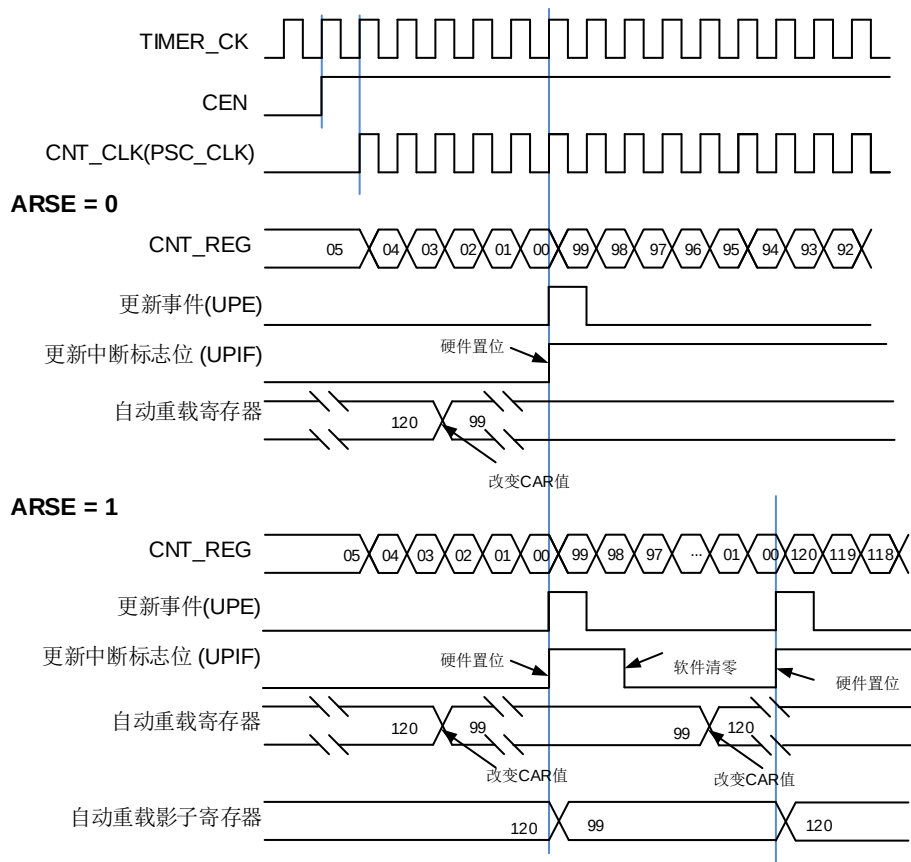


图 12-7. 向下计数时序图，在运行时改变 `TIMERx_CAR` 寄存器值

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_CAR-1`) 产生一个上溢事件；向下计数模式中，定时器模块在计数器计数到 1 时产生一个下溢事件。在中央计数模式中，`TIMERx_CTL0` 寄存器中的计数方向控制位 `DIR` 只读，表明了计数方向。计数方向被硬件自动更新。

将 `TIMERx_SWEVG` 寄存器的 `UPG` 位置 1 可以初始化计数值为 0，并产生一个更新事件，而无需考虑计数器在中央模式下是向上计数还是向下计数。

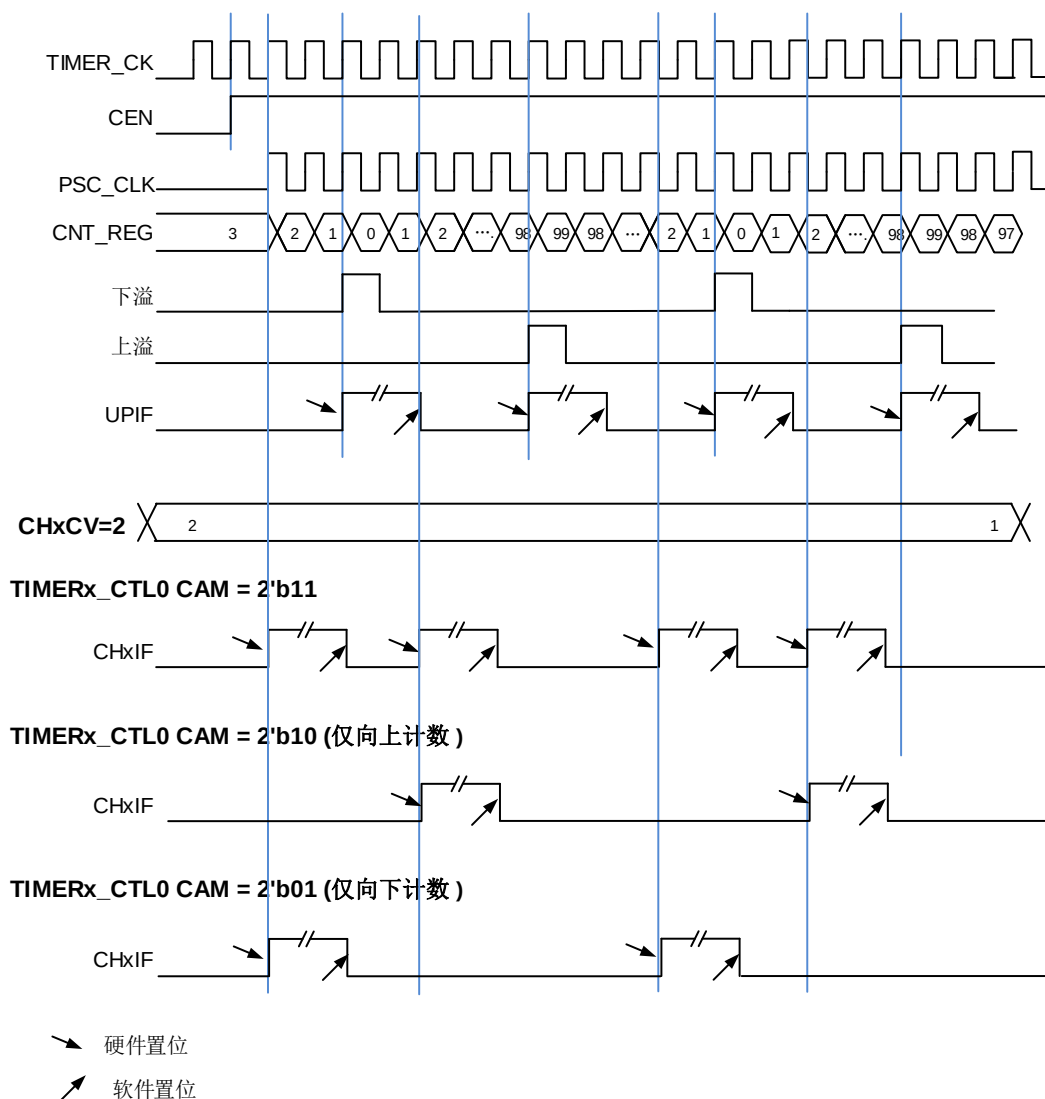
上溢或者下溢时，`TIMERx_INTF` 寄存器中的 `UPIF` 位都会被置 1，然而 `CHxIF` 位置 1 与 `TIMERx_CTL0` 寄存器中 `CAM` 的值有关。具体细节参考 [图 12-8. 中央计数模式计数器时序图](#)。

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 12-8. 中央计数模式计数器时序图](#) 给出了一些例子，当 `TIMERx_CAR=0x99`，`TIMERx_PSC=0x0` 时，计数器的行为。

图 12-8. 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_CREP0** 寄存器的 **CREP0** 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_CREP0** 寄存器中 **CREP0** 的值并产生一个更新事件。

新写入的 **CREP0** 值将在下一次更新事件到来时生效。当 **CREP0** 的值为奇数，并且计数器在中央对齐模式下计数时，更新事件发生在上溢或下溢取决于写入的 **CREP0** 值何时生效。如果在写入奇数到 **CREP0** 寄存器后由软件生成更新事件 (**UPG** 位置 1)，则在下溢时产生更新事件。如果在写

入奇数到 CREP0 寄存器后下一个更新事件发生在上溢，此后将在上溢时产生更新事件。

图 12-9. 中央计数模式下计数器重复时序图

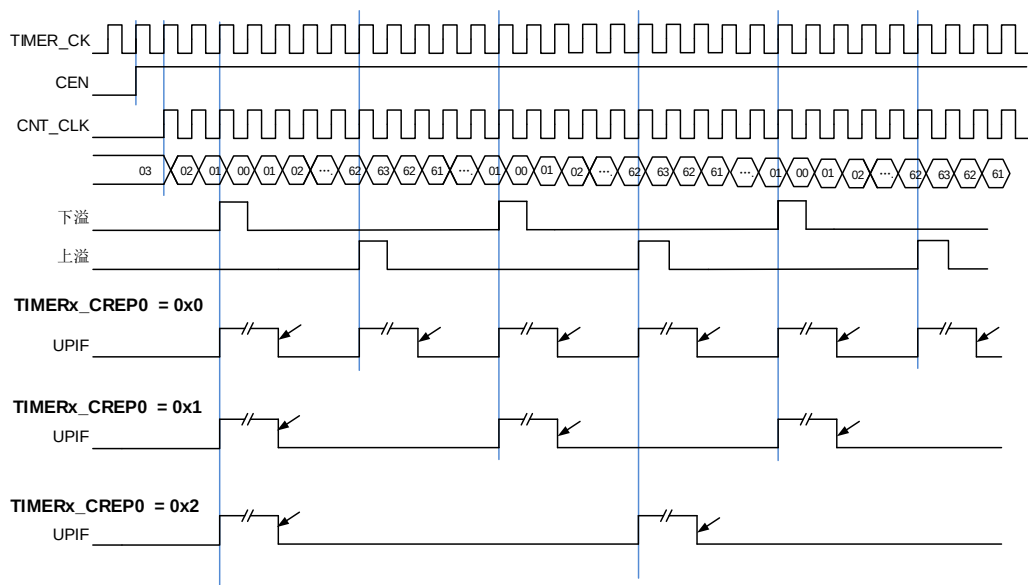


图 12-10. 在向上计数模式下计数器重复时序图

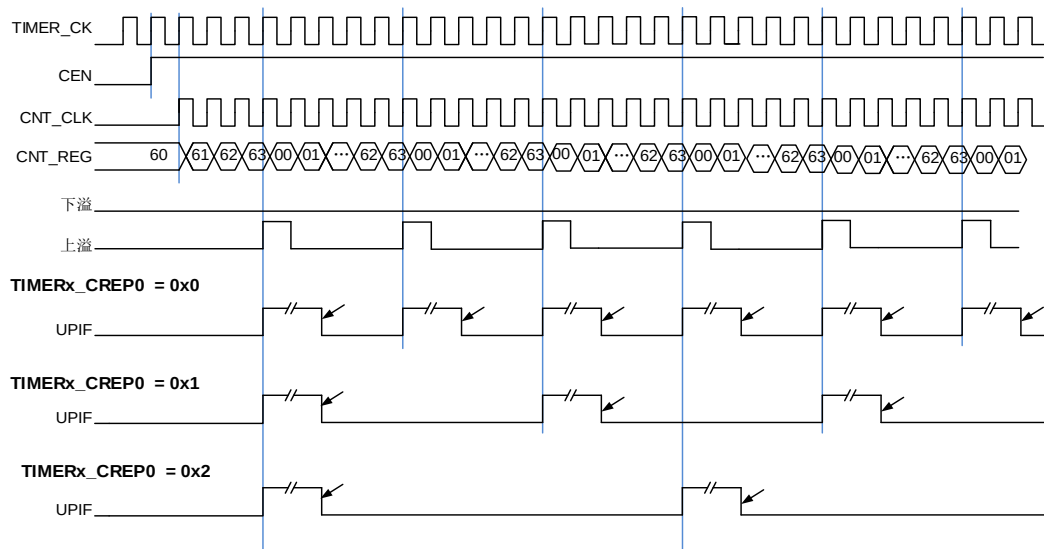
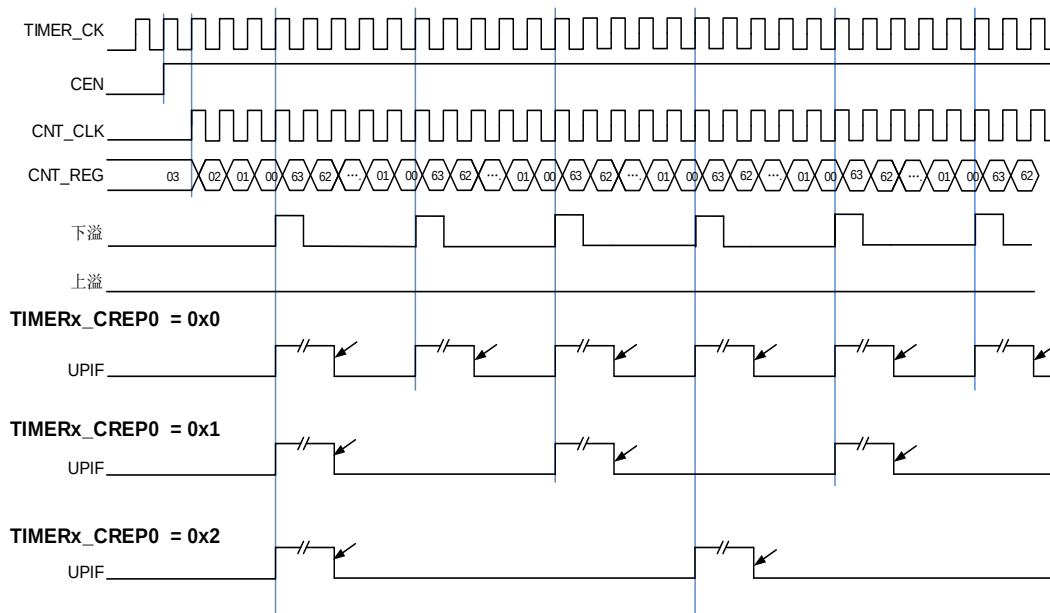


图 12-11. 在向下计数模式下计数器重复时序图



捕获/比较通道

高级定时器拥有 4 个独立的通道用于捕获输入或比较输出是否匹配。每个通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级。

■ 输入捕获模式

捕获模式允许通道测量一个波形时序，频率，周期，占空比等。输入级包括一个数字滤波器，一个通道极性选择，边沿检测和一个通道预分频器。如果在输入引脚上出现被选择的边沿，TIMERx_CHxCV (x=0..3) 寄存器会捕获计数器当前的值，同时 CHxIF (x=0..3) 位置 1，如果 CHxIE = 1 (x=0..3)，则产生相应的通道中断。

[illegible]

通道输入信号 **Clx** 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/CHxNP** 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或内部触发信号。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 位）：

第二步：边沿选择（TIMERx CHCTL2 寄存器中 CHxP 和 CHxNP 位）：

第三步：捕获源选择（TIMERx CHCTL0 寄存器中 CHxMS）：

第四步：中断使能（TIMERx DMAINTEN 寄存器中 CHxIE、CHxDEN 位）：

第五步：捕获使能（TIMERx CHCTL2 寄存器中 CHxEN）。

262

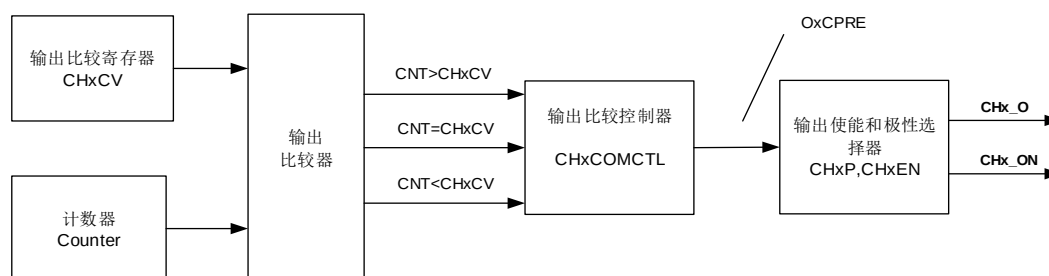
直接产生：软件设置 CHxG 位，会直接产生中断和 DMA 请求。

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 `3'b001`，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 `3'b010`，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 12-13. 输出比较逻辑](#)给出了通道的输出比较逻辑。

图 12-13. 输出比较逻辑



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，被置低电平或者翻转。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 `CHxDEN` = 1 则会产生 DMA 请求。

配置步骤如下：

第一步：时钟配置：

配置定时器时钟源，预分频器等。

第二步：比较模式配置：

- 设置 `CHxCOMSEN` 位来配置输出比较影子寄存器；
- 设置 `CHxCOMCTL` 位来配置输出模式（置高电平/置低电平/翻转）；
- 设置 `CHxP/CHxNP` 位来选择有效电平的极性；
- 设置 `CHxEN/CHxNEN` 使能输出。

第三步：通过 `CHxIE/CHxDEN` 位配置中断/DMA 请求使能。

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

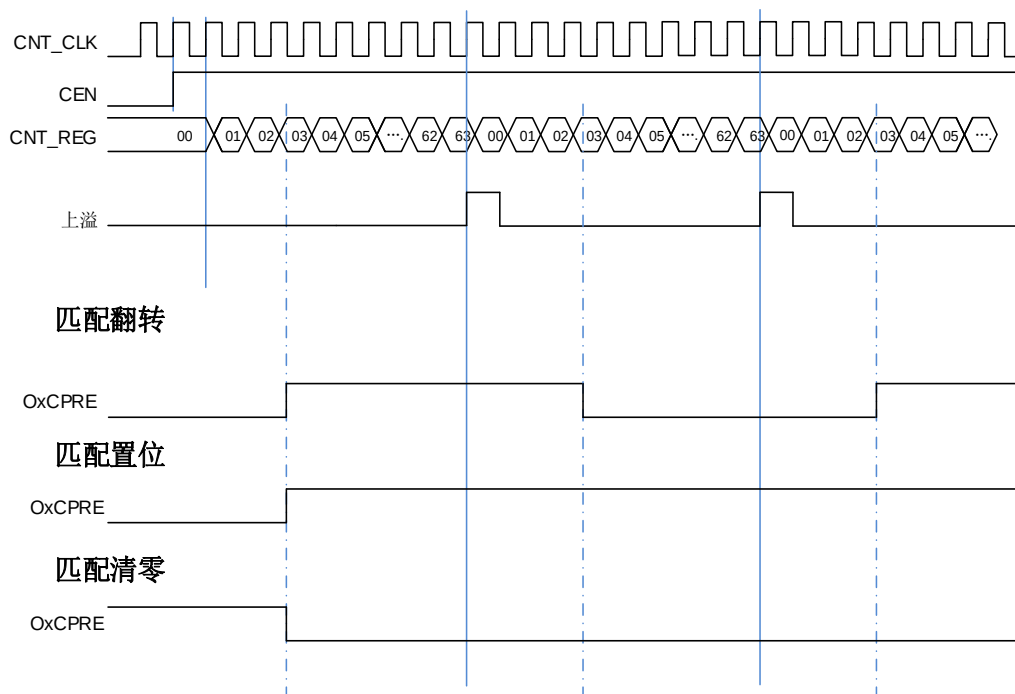
`TIMERx_CHxCV` 可以在运行时根据你所期望的波形而改变。

第五步：设置 `CEN` 位使能定时器。

[图 12-14. 三种输出比较模式](#)显示了三种比较输出模式：翻转/置高电平/置低电平，`CAR=0x63`,

CHxVAL=0x3。

图 12-14. 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 4'b0110，PWM 模式 1 是配置 CHxCOMCTL 为 4'b0111），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 12-15. 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 12-16. CAPWM 时序图](#)显示了 CAPWM 的输出波形和中断。

当计数器向上计数时，在 PWM0 模式下（CHxCOMCTL=4'b0110），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；PWM1 模式下（CHxCOMCTL=4'b0111），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。

图 12-15. EAPWM 时序图

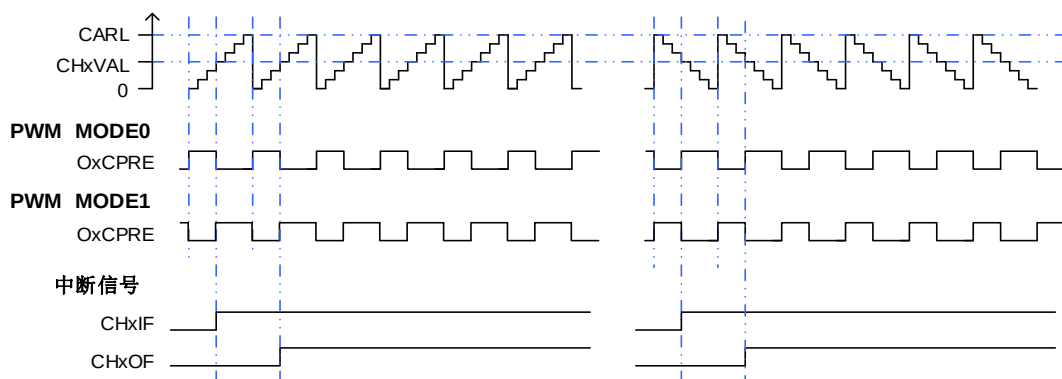
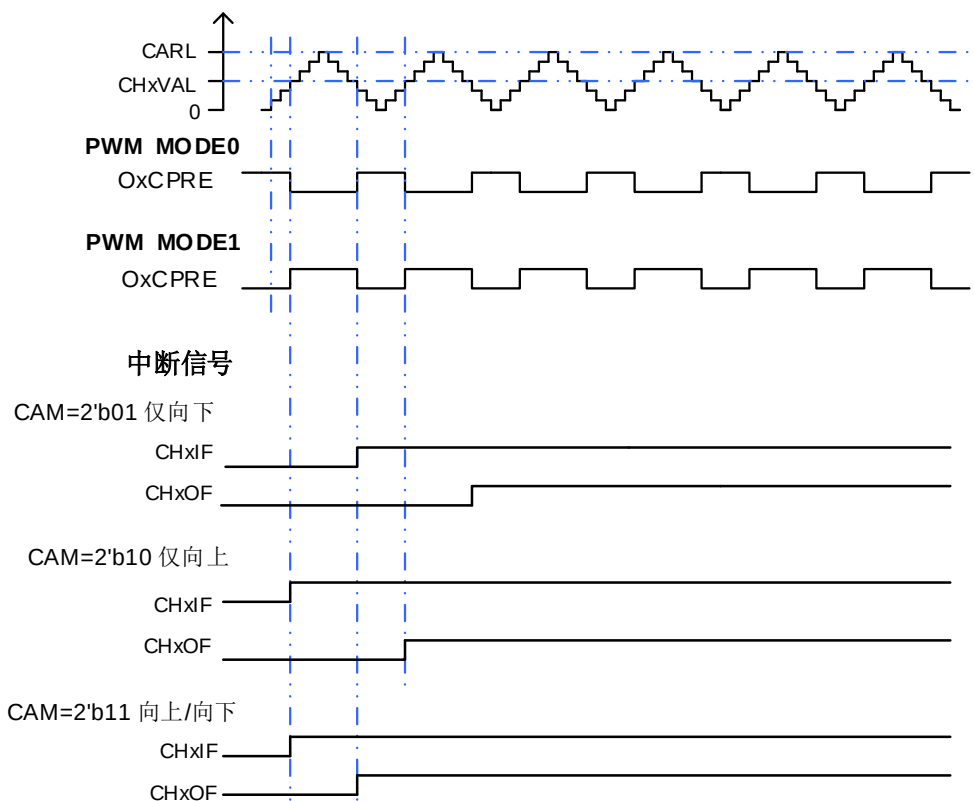


图 12-16. CAPWM 时序图



复合 PWM 模式

在复合 PWM 模式中 ($CHxCPWMEN = 1'b1$, $CHxMS[2:0] = 3'b000$ 和 $CHxCOMCTL = 4'b0110$ 、 $4'b0111$)，通道 x ($x=0..3$) 上的 PWM 输出信号由 $CHxVAL$ 和 $CHxCOMVAL\_ADD$ 位确定。

如果 $CHxCOMCTL = 4'b0110$ (PWM 模式 0) 且 $DIR = 1'b0$ (向上计数模式)，或者 $CHxCOMCTL = 4'b0111$ (PWM 模式 1) 且 $DIR = 1'b1$ (向下计数模式)，当计数器和 $CHxVAL$ 的值相匹配时通道 x 输出强制为低。当计数器与 $CHxCOMVAL\_ADD$ 的值相匹配时，通道 x 输出强制为高。

如果 $CHxCOMCTL = 4'b0111$ (PWM 模式 1) 且 $DIR = 1'b0$ (向上计数模式), 或者 $CHxCOMCTL = 4'b0110$ (PWM 模式 0) 且 $DIR = 1'b1$ (向下计数模式), 当计数器和 $CHxVAL$ 的值相匹配时通道 x 输出强制为高。当计数器与 $CHxCOMVAL\_ADD$ 的值相匹配时, 通道 x 输出强制为低。

当 $CHxVAL$ 或 $CHxCOMVAL\_ADD = 0 / CARL$, 通道输出被进行了特殊处理, 通过 $TIMERx\_CHCTL2$ 寄存器的 $CHxPERFOREN$ 位置位, 强制 $OxCPRE$ 输出高电平或低电平 (根据所选复合 PWM 模式确定)。

PWM 的周期取决于 $(CARL + 0x0001)$, PWM 脉冲宽度可以下 [表 12-2 复合 PWM 脉冲宽度](#) 计算。

表 12-2 复合 PWM 脉冲宽度

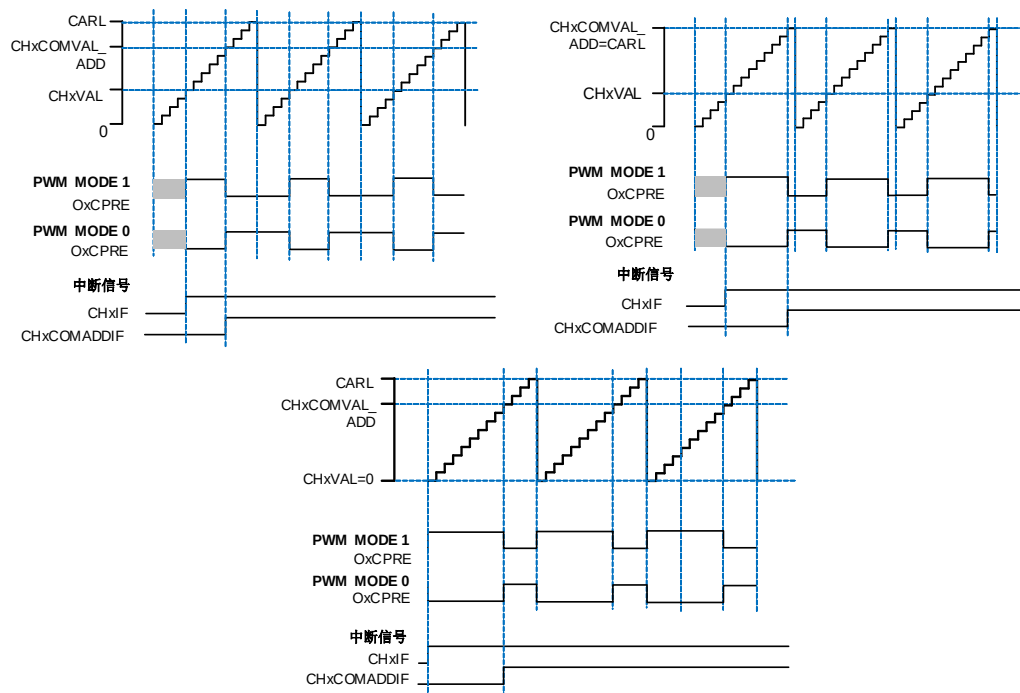
条件	模式	PWM 脉冲宽度
$CHxVAL < CHxCOMVAL\_ADD \leq CARL$	PWM 模式 0	$(CARL + 0x0001) + (CHxVAL - CHxCOMVAL\_ADD)$
	PWM 模式 1	$(CHxCOMVAL\_ADD - CHxVAL)$
$CHxCOMVAL\_ADD < CHxVAL \leq CARL$	PWM 模式 0	$(CHxVAL - CHxCOMVAL\_ADD)$
	PWM 模式 1	$(CARL + 0x0001) + (CHxCOMVAL\_ADD - CHxVAL)$
$(CHxVAL = CHxCOMVAL\_ADD \leq CARL)$ 或 $(CHxVAL > CARL > CHxCOMVAL\_ADD)$	PWM 模式 0 (向上计数) 或 PWM 模式 1 (向下计数)	100%
	PWM 模式 0 (向下计数) 或 PWM 模式 1 (向上计数)	0%
	PWM 模式 0 (向上计数) 或 PWM 模式 1 (向下计数)	0%
	PWM 模式 0 (向下计数) 或 PWM 模式 1 (向上计数)	100%
$CHxCOMVAL\_ADD > CARL > CHxVAL$	PWM 模式 0 (向上计数) 或 PWM 模式 1 (向下计数)	0%
	PWM 模式 0 (向下计数) 或 PWM 模式 1 (向上计数)	100%
$(CHxVAL > CARL)$ 且 $(CHxCOMVAL\_ADD > CARL)$	-	CHx_O 输出保持

当计数器计数到 $CHxVAL$, $CHxIF$ 位置 1 且如果 $CHxIE=1$ 通道 x 产生中断, 如果 $CHxDEN=1$, 则产生 DMA 请求。当计数器计数到 $CHxCOMVAL\_ADD$ 时, $CHxCOMADDIF$ 位置 1 (该中断标志位只在复合 PWM 模式有效, $CHxCPWMEN=1$), 如果 $CHxCOMADDIE = 1$ 通道 x 附加比较中断产生 (只有中断产生, 没有 DMA 请求响应)。

根据 $CHxVAL$, $CHxCOMVAL\_ADD$ 和 $CARL$ 之间的关系, 可以分为四种情况:

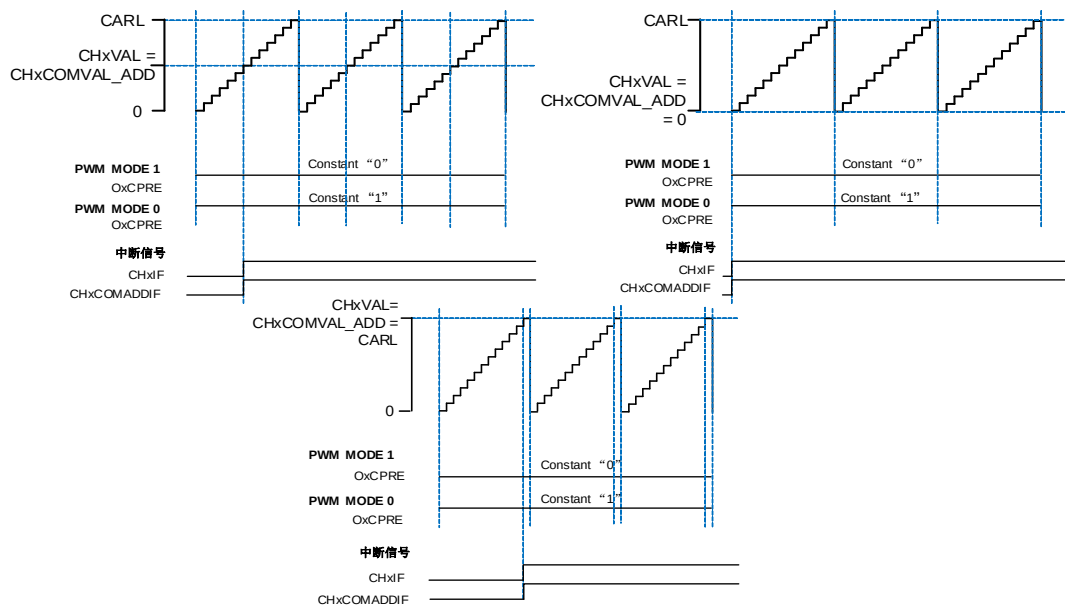
- 1) $CHxVAL < CHxCOMVAL\_ADD$, $CHxVAL$ 和 $CHxCOMVAL\_ADD$ 值介于 0 和 $CARL$ 之间。

图 12-17. 通道 x 输出 PWM (CHxVAL < CHxCOMVAL_ADD)



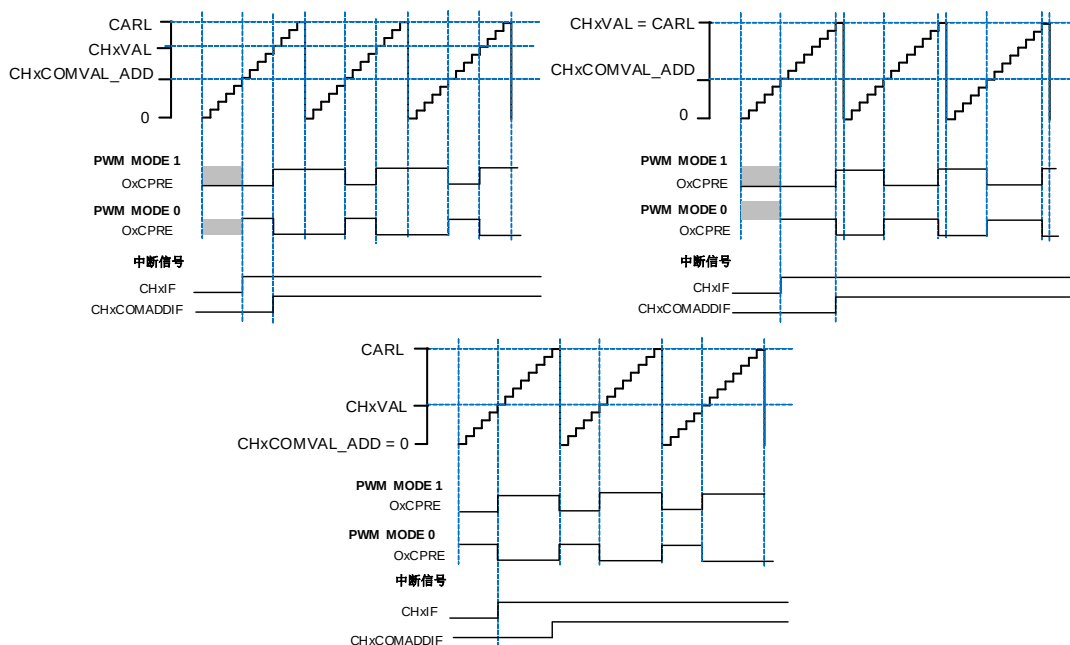
2) $CHxVAL = CHxCOMVAL\_ADD$, $CHxVAL$ 和 $CHxCOMVAL\_ADD$ 值介于0和CARL之间。

图 12-18. 通道 x 输出 PWM (CHxVAL = CHxCOMVAL_ADD)



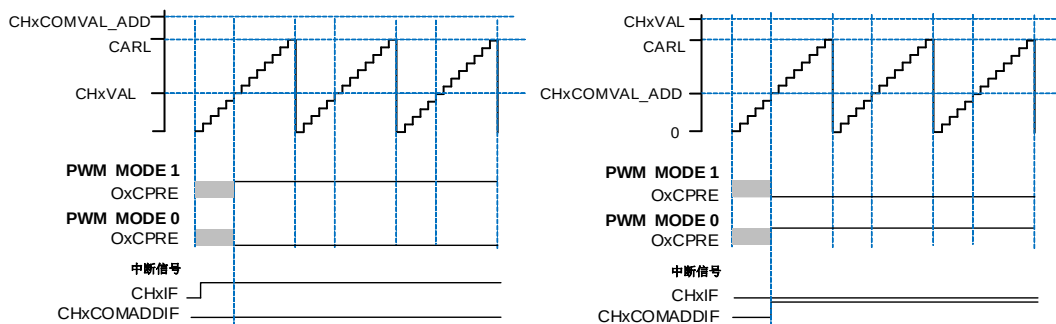
3) $CHxVAL > CHxCOMVAL\_ADD$, $CHxVAL$ 和 $CHxCOMVAL\_ADD$ 值介于0和CARL之间。

图 12-19. 通道 x 输出 PWM (CHxVAL > CHxCOMVAL_ADD)



4) $CHxVAL$ 或 $CHxCOMVAL\_ADD$ 值大于 $CARL$ 。

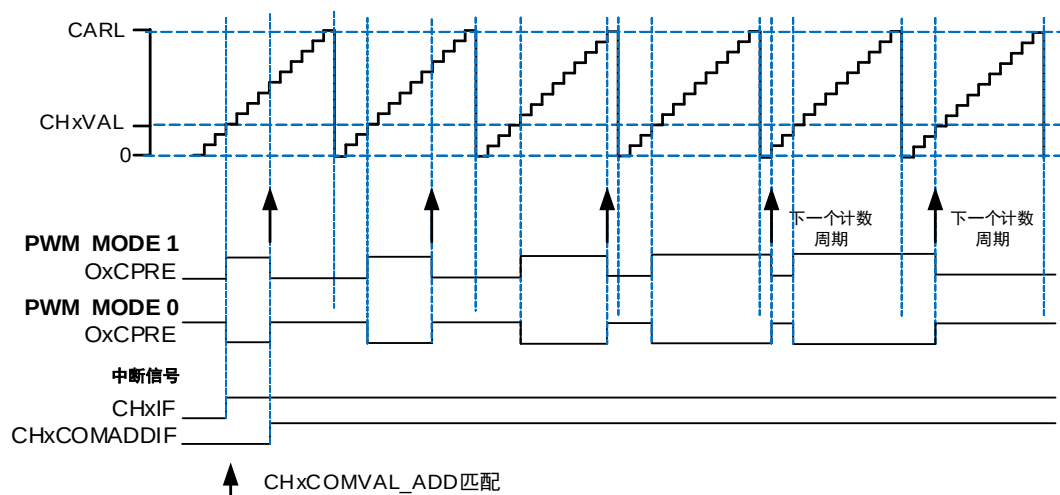
图 12-20. 通道 x 输出 PWM ($CHxVAL$ 或 $CHxCOMVAL\_ADD > CARL$)



复合 PWM 模式支持不修改周期只修改占空比的 PWM 信号的生成。[图 12-21. 通道 x 输出 PWM 占空比随着 CHxCOMVAL_ADD 值而改变](#)显示 PWM 输出和中断波形。

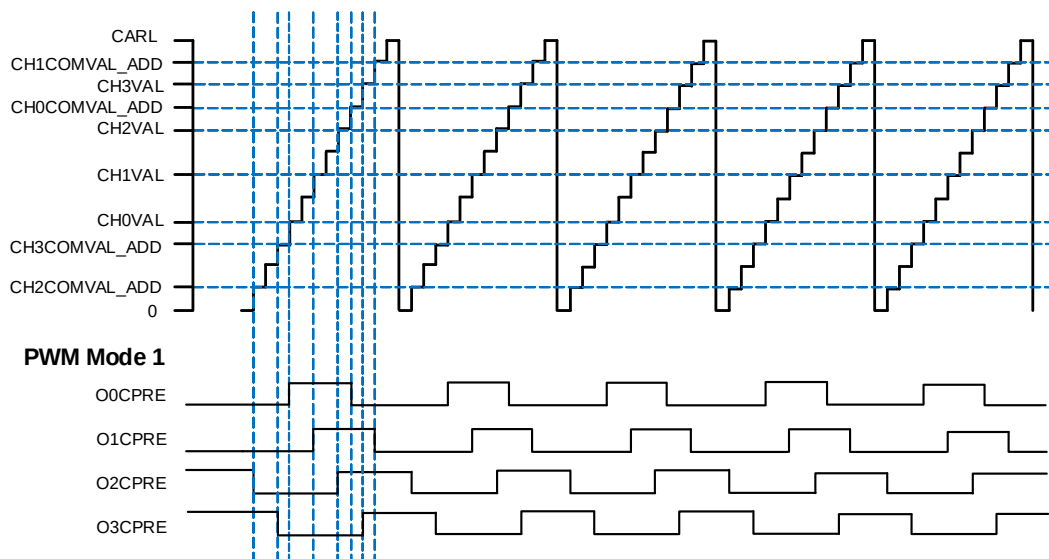
在某些情况下， $CHxCOMVAL\_ADD$ 的匹配事件可以发生在下一个计数周期（ $CHxCOMVAL\_ADD$ 值在计数器到达 $CHxVAL$ 值之后被写入，且 $CHxCOMVAL\_ADD$ 值小于或者等于 $CHxVAL$ 值）。

图 12-21. 通道 x 输出 PWM 占空比随着 CHxCOMVAL_ADD 值而改变



如果多个通道配置为复合 PWM 模式，可以为每对通道 x 的匹配边沿设定一个偏移量（相对于其它通道）。这种特性在产生照明 PWM 控制信号时非常有用，因为在这种情况下，希望彼此边缘不重合，以消除噪声的产生。CHxVAL 寄存器值是 PWM 脉冲相对于计数器周期开始的偏移。

图 12-22. 复合 PWM 模式下四通道输出



非对称 PWM 模式

非对称 PWM 波允许两个中央对齐的 PWM 波之间产生一个可编程的相位偏移。在非对称 PWM 波模式下（CHxCOMCTL[3:0]=4' b1010/4' b1011 and CHxMS[2:0]=3' b000），在通道上输出的 PWM 波信号是由 CHxVAL 和 CHxCOMVAL_ADD 位共同控制的，非对称 PWM 波只在中央对齐模式下有效。

如果 CHxCOMCTL = 4' b1010（非对称 PWM 波模式 0），向上计数模式（DIR = 1' b0），当计

数值达到 CHxVAL 的值时，输出强制为低电平。

如果 CHxCOMCTL = 4' b1010（非对称 PWM 波模式 0），向下计数模式（DIR = 1' b1），当计数值达到 CHxCOMVAL_ADD 的值时，输出强制为高电平。

如果 CHxCOMCTL = 4' b1011（非对称 PWM 波模式 1），向上计数模式（DIR = 1' b0），当计数值达到 CHxVAL 的值时，输出强制为高电平。

如果 CHxCOMCTL = 4' b1011（非对称 PWM 波模式 1），向下计数模式（DIR = 1' b1），当计数值达到 CHxCOMVAL_ADD 的值时，输出强制为低电平。

与复合 PWM 模式相比，非对称 PWM 模式只在中央对齐模式下有效，同时 CHxVAL 只在向上计数模式时有效，CHxCOMVAL_ADD 只在向下计数模式时有效。当 CHxVAL 或 CHxCOMVAL_ADD = 0 / CARL，通道输出被进行了特殊处理，通过 TIMEx_CHCTL2 寄存器的 CHxPERFOREN 位置位，强制 OxCPre 输出高电平或低电平（根据所选非对称 PWM 模式确定）。

当 CHxVAL = 0，在非对称 PWM0 模式下，通道输出被强制为低电平，在非对称 PWM1 模式下，通道输出被强制为高电平。

当 CHxCOMVAL_ADD = 0，在非对称 PWM0 模式下，通道输出被强制为高电平，在非对称 PWM1 模式下，通道输出被强制为低电平。

当 CHxVAL = CARL，在非对称 PWM0 模式下，通道输出被强制为低电平，在非对称 PWM1 模式下，通道输出被强制为高电平。

当 CHxCOMVAL_ADD = CARL，在非对称 PWM0 模式下，通道输出被强制为高电平，在非对称 PWM1 模式下，通道输出被强制为低电平。

PWM 的周期取决于 2*CARL，PWM 脉冲宽度可以下 [表 12-3 非对称 PWM 脉冲宽度](#) 计算。

表 12-3 非对称 PWM 脉冲宽度

条件	模式	PWM 脉冲宽度
CHxVAL < CHxCOMVAL_ADD ≤ CARL 或者 CHxCOMVAL_ADD < CHxVAL ≤ CARL 或者 CHxVAL = CHxCOMVAL_ADD	非对称 PWM 模式 0	CHxCOMVAL_ADD + CHxVAL
	非对称 PWM 模式 1	2*CARL – CHxCOMVAL_ADD – CHxVAL
CHxVAL > CARL > CHxCOMVAL_ADD	非对称 PWM 模式 0	100%
	非对称 PWM 模式 1	0%
CHxCOMVAL_ADD > CARL > CHxVAL	非对称 PWM 模式 0	0%
	非对称 PWM 模式 1	100%
(CHxVAL > CARL) 且 (CHxCOMVAL_ADD > CARL)	-	CHx_O 输出保持

当计数器计数到 CHxVAL，CHxIF 位置 1 且如果 CHxIE=1 通道 x 产生中断，如果 CHxDEN=1，则产生 DMA 请求。当计数器计数到 CHxCOMVAL_ADD 时，CHxCOMADDIF 位置 1（该中断标

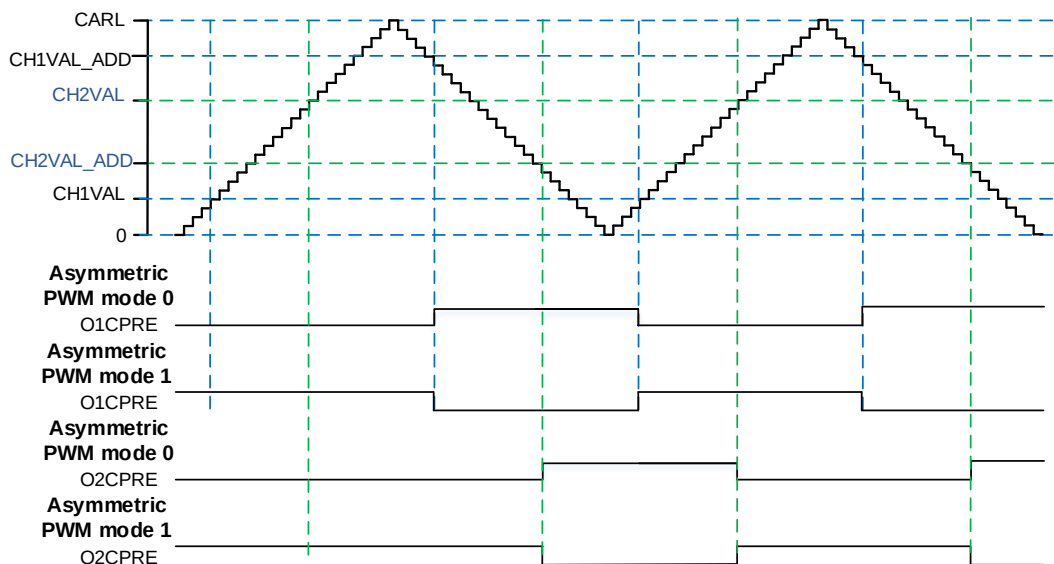
志位只在复合 PWM 模式和非对称 PWM 模式有效，当 $CHxCOMCTL[3:2] = 2'b10$ ，如果 $CHxCOMADDIE = 1$ 通道 x 附加比较中断产生（只有中断产生，没有 DMA 请求响应）。

根据 $CHxVAL$ ， $CHxCOMVAL\_ADD$ 和 $CARL$ 之间的关系，可以分为四种情况：

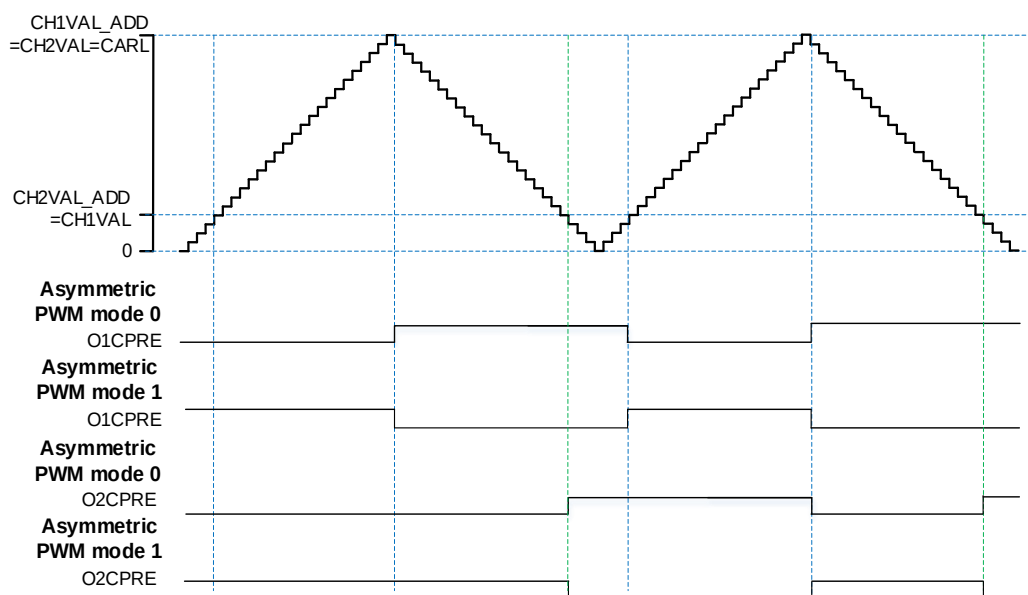
- 1) $CHxVAL < CHxCOMVAL\_ADD$ ， $CHxVAL$ 和 $CHxCOMVAL\_ADD$ 值介于 0 和 $CARL$ 之间。

图 12.23 通道 x 输出 PWM ($CHxVAL < CHxCOMVAL\_ADD / CHxCOMVAL\_ADD < CHxVAL$)

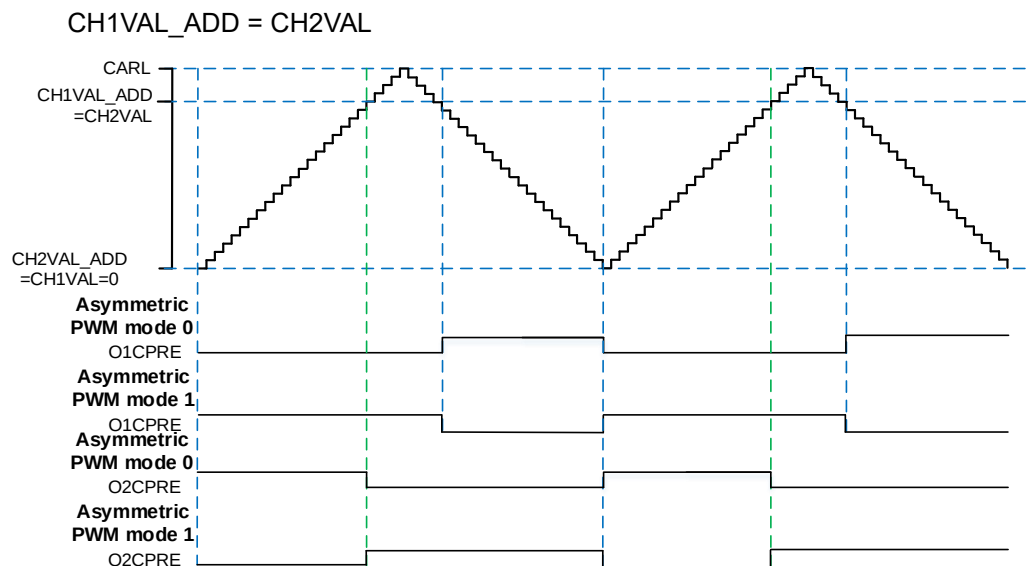
- a. $0 < CH1VAL < CH1VAL\_ADD < CARL / 0 < CH2VAL\_ADD < CH2VAL < CARL$



- b. $0 < CH1VAL < CH1VAL\_ADD = CARL / 0 < CH2VAL\_ADD < CH2VAL = CARL$ ，同时 $CH1VAL = CH2VAL\_ADD$



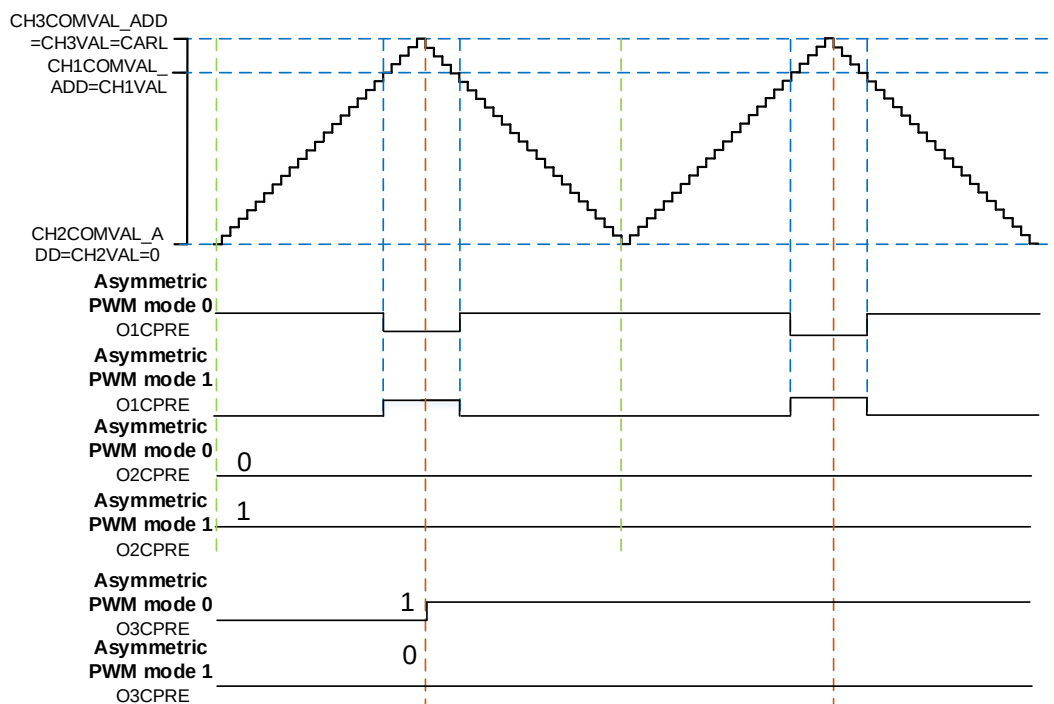
- c. $0 = CH1VAL < CH1VAL\_ADD < CARL / 0 = CH2VAL\_ADD < CH2VAL < CARL$ ，同时



2) $CHxVAL = CHxCOMVAL\_ADD$, $CHxVAL$ 和 $CHxCOMVAL\_ADD$ 值介于0和CARL之间。

图 12.24. 通道 x 输出 PWM ($CHxVAL = CHxCOMVAL\_ADD$)

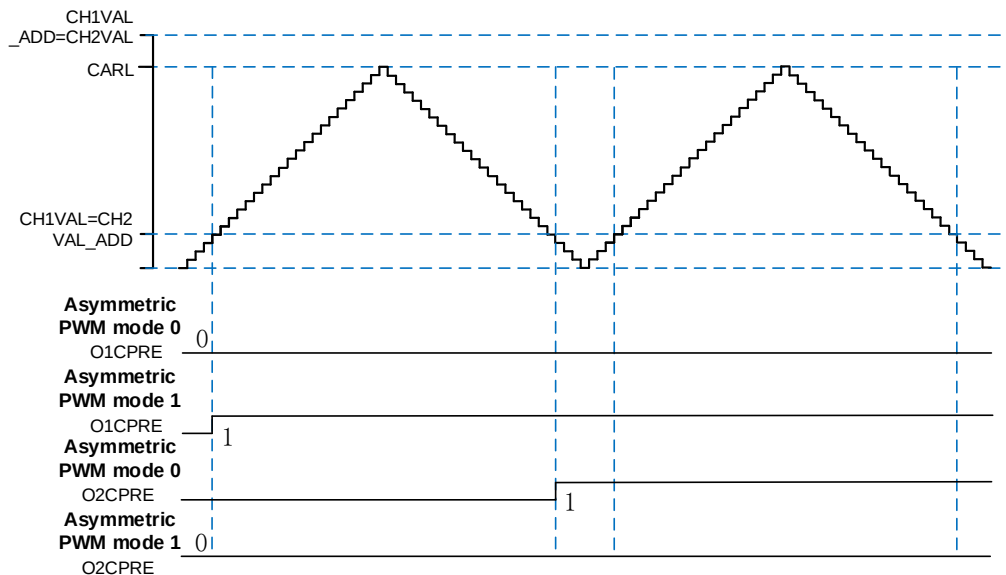
$0 < CH1VAL = CH1VAL\_ADD < CARL / CH2VAL = CH2VAL\_ADD = 0 / CH3VAL = CH3VAL\_ADD = CARL$



3) $CHxVAL$ 或 $CHxCOMVAL\_ADD$ 值大于CARL。

图 12.25. 通道 x 输出 PWM (CHxVAL 或 CHxCOMVAL_ADD > CARL)

$0 < CH1VAL < CARL < CH1VAL\_ADD / 0 < CH2VAL\_ADD < CARL < CH2VAL$



输出匹配脉冲选择

当发生匹配事件时，CHx_O (x=0..3) 的输出由 CHxCOMCTL[3:0] (x=0..3) 位设置，通过配置 CHxOMPSEL[1:0] (x=0..3) 位，可选择 CHx_O (x=0..3) 的输出信号正常或者脉冲。

当匹配事件发生时，CHxOMPSEL[1:0] (x=0..3) 用于选择 OxCPRE 信号输出（驱动 CHx_O）：

- CHxOMPSEL = 2'b00, OxCPRE信号根据CHxCOMCTL[3:0]位的配置正常输出；
- CHxOMPSEL = 2'b01, 只有在计数器向上计数，发生匹配事件时，OxCPRE信号输出一个脉冲，且脉冲宽度为一个CK_TIMER时钟周期；
- CHxOMPSEL = 2'b10, 只有在计数器向下计数，发生匹配事件时，OxCPRE信号输出一个脉冲，且脉冲宽度为一个CK_TIMER时钟周期；
- CHxOMPSEL = 2'b11, 无论计数器向上计数还是向下计数，发生匹配事件时，OxCPRE信号输出一个脉冲，且脉冲宽度为一个CK_TIMER时钟周期；

图 12-26. 边沿对齐模式下 CHx_O 输出脉冲 (CHxOMPSEL≠2'b00)

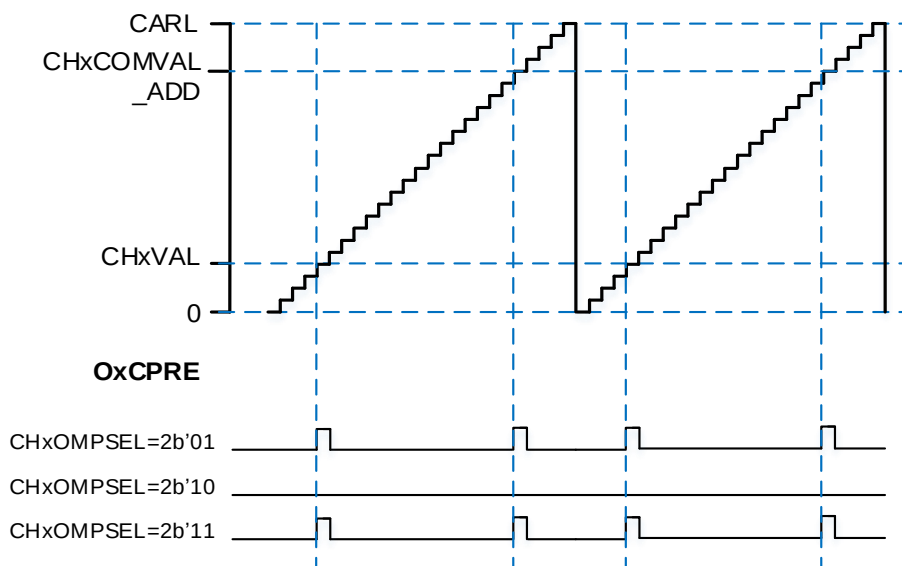
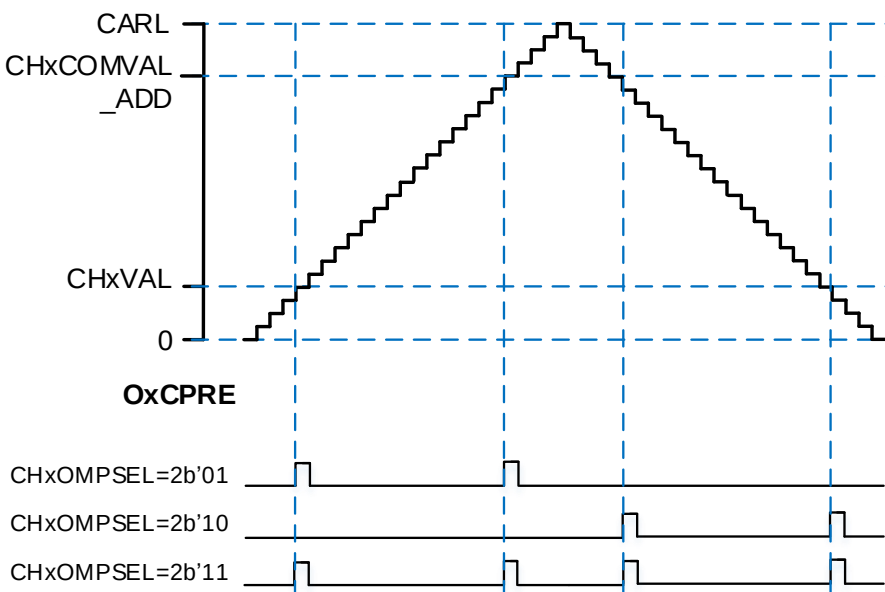


图 12-27. 中央对齐模式下 CHx_O 输出脉冲 (CHxOMPSEL≠2'b00)



通道输出参考信号

如[图 12-13. 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将产生一个中间信号，即 OxCPRE 信号（通道 x 参考信号）。

OxCPRE 信号有若干类型的输出功能，通过配置 CHxCOMCTL 位定义 OxCPRE 信号类型。

下面以 OxCPRE 为例进行说明，设置 CHxCOMCTL=0x00 可以保持原始电平；设置 CHxCOMCTL=0x01 可以将 OxCPRE 信号设置为高电平；设置 CHxCOMCTL=0x02 可以将 OxCPRE 信号设置为低电平；设置 CHxCOMCTL=0x03，在计数器值和 TIMERx_CHxCV 寄存器

的值匹配时，可以翻转输出信号。

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

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

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。在下次更新事件（OCPRECIWM = 1）或者上溢/下溢事件（OCPRECIWM = 0）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 4 对通道，所有 4 对通道都具有此功能。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_CCHP0 寄存器中和 TIMERx_CTL1 寄存器中的 POEN&CHPOENx(x=0...2)、ROS、IOS、ISOx 和 ISOxN 位。输出极性由 TIMERx_CHCTL2 寄存器中的 CHxP 和 CHxNP 位来决定。

当 CHx_O 和 CHx_ON 的输出互补时，有三种输出情况：输出使能、输出关闭状态和输出禁能，具体情况可参考 [表 12-4. 由参数控制的互补输出表](#)。

表 12-4. 由参数控制的互补输出表

互补参数					输出状态	
POEN &CHP OENx	ROS	IOS	CHxEN	CHxNEN	CHx_O	CHx_ON
0	0/1	0	0	0	CHx_O / CHx_ON = LOW CHx_O / CHx_ON 输出禁能 ⁽¹⁾	
				1	CHx_O/CHx_ON输出关闭状态 ⁽²⁾ ：	
			1	0	通道先输出无效电平：CHx_O = CHxP，CHx_ON = CHxNP；如果死区产生时钟未失效，在死区时间之后：CHx_O = ISOx，CHx_ON = ISOxN ⁽³⁾	
				1		
1	0	0/1	0	0	CHx_O/CHx_ON = LOW CHx_O/CHx_ON输出禁能	
				1	CHx_O = LOW CHx_O输出禁能	CHx_ON=OxCPRE ⊕ ⁽⁴⁾ CHxNP
		1	x	x	CHx_O/CHx_ON输出关闭状态：	
					通道先输出无效电平：CHx_O = CHxP，CHx_ON = CHxNP；如果死区产生时钟未失效，在死区时间之后：CHx_O = ISOx，CHx_ON = ISOxN	

互补参数					输出状态	
POEN &CHP OENx	ROS	IOS	CHxEN	CHxNEN	CHx_O	CHx_ON
						CHx_ON输出使能
			1	0	$CHx\_O = OxCPre \oplus CHxP$ CHx_O输出使能	CHx_ON = LOW CHx_ON输出禁能
				1	$CHx\_O = OxCPre \oplus CHxP$ CHx_O输出使能	$CHx\_ON = (!OxCPre)^{(5)} \oplus CHxNP$ CHx_ON输出使能
			0	0	CHx_O = CHxP CHx_O输出关闭状态	CHx_ON = CHxNP CHx_ON输出关闭状态
				1	CHx_O = CHxP CHx_O输出关闭状态	$CHx\_ON = OxCPre \oplus CHxNP$ CHx_ON输出使能
	1		1	0	$CHx\_O = OxCPre \oplus CHxP$ CHx_O输出使能	CHx_ON = CHxNP CHx_ON输出关闭状态
				1	$CHx\_O = OxCPre \oplus CHxP$ CHx_O输出使能	$CHx\_ON = (!OxCPre) \oplus CHxNP$ CHx_ON输出使能

注意：

- (1) 输出禁能：CHx_O / CHx_ON 输出与对应引脚断开，对应引脚电平受 GPIO 上下拉配置控制，无上下拉时为悬空高阻态；
- (2) 输出关闭状态：CHx_O / CHx_ON 输出无效电平（ $CHx\_O = 0 \oplus CHxP = CHxP$ ）；
- (3) 详情见中止模式章节。
- (4) $\oplus$ ：异或操作；
- (5) $(!OxCPre)$ ：OxCPre 信号的互补信号。

死区时间插入

设置 CHxEN 和 CHxNEN 为 1'b1 同时设置 POEN&CHPOENx(x=0...2)，死区插入就会被使能。DTCFG 位域定义了死区时间，死区时间对所有通道有效。死区时间设置的细节请参考[互补通道保护寄存器0 \(TIMERx_CCHP0\)](#)。

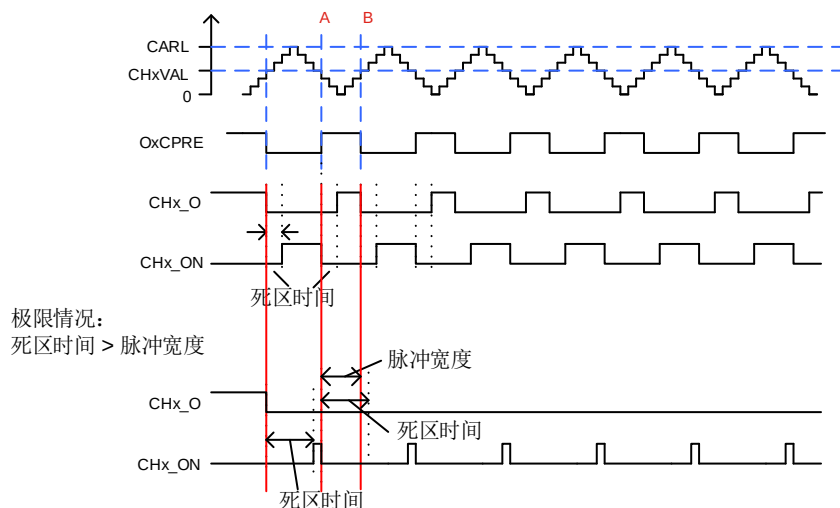
死区时间的插入，确保了通道互补的两路信号不会同时有效。

在 PWM0 模式，当通道 x 匹配发生时（TIMERx 计数器=TIMERx_CHxCV），OxCPre 翻转。在[图 12-28. 带死区时间的互补输出](#)的 A 点，CHx_O 信号在死区时间内为低电平，直到死区时间过后才变为高电平，而 CHx_ON 信号立刻变为低电平。同样，在 B 点，计数器再次匹配（TIMERx 计数器=TIMERx_CHxCV），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。

有时会有一些死角事件发生，例如：如果死区延时大于或者等于 CHx_O 信号的占空比，CHx_O

信号一直为无效值。如 [图 12-28. 带死区时间的互补输出](#) 所示。

图 12-28. 带死区时间的互补输出



通过配置 `TIMERx_CTL2` 寄存器中的 `DTIENCHx` ($x = 0..3$) 位, 可实现对每对通道的死区插入功能的独立控制。当 `DTIENCHx` ($x = 0..3$) 位为“0”时, 相应的通道 `CHx_O` 和 `CHx_ON` 将不会插入死区。

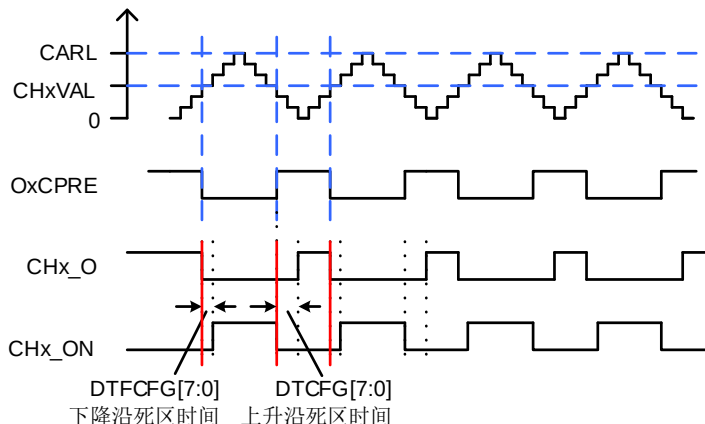
不同的死区时间插入

当 `DTDIFEN` 位 (在 `TIMERx_CCHP1` 寄存器中) 设置为 1 时, `CHx_O` 和 `CHx_ON` 信号可以输出不同的死区时间, 具体如 [图 12-29. 不同死区时间的互补输出 \(DTDIFEN=1\)](#) 所示。

通道输出准备信号 `OxCPre` 上升沿的死区时间由 `TIMERx_CCHP0` 寄存器中的 `DTCFG[7:0]` 位域配置。`OxCPre` 信号的下降沿的死区时间由 `TIMERx_CCHP1` 寄存器中的 `DTFCFG[7:0]` 位域配置。

`DTDIFEN` 位必须在使能计数器之前写入, 且当 `CEN=1` 时不能修改。

图 12-29. 不同死区时间的互补输出 (DTDIFEN=1)



中止功能

使用中止功能时，CHx_O 和 CHx_ON 信号的输出电平由以下位控制：TIMERx_CCHP0 寄存器的 POEN、IOS 和 ROS 位，TIMERx_CTL1 寄存器的 ISOx 和 ISOxN 位。在这种情况下，CHx_O 和 CHx_ON 信号不能同时设置为有效电平。

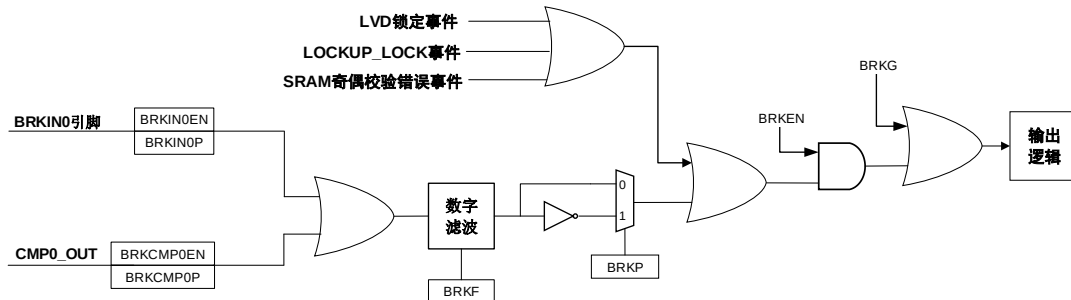
中止事件是所有源逻辑或运算的结果。中止功能可以处理三种类型的事件源：

- 外部信号源：来自BRKIN0输入；
- 系统源：LVD锁定事件，Cortex®-M33锁定事件或SRAM奇偶校验错误事件；
- 片上外设源：比较器输出。

中止事件也可以由软件置位 TIMERx_SWEVG 寄存器中的 BRKG 位产生。

BREAK 中止功能逻辑如[图 12-30. BREAK 中止功能逻辑图](#)所示，其中 BRKIN0 可以从 TRIGSEL 模块选择 GPIO 引脚，具体可参考[TIMER0 BRKIN 触发选择寄存器 \(TRIGSEL_TIMER0BRKIN\)](#)和[TIMER7 BRKIN 触发选择寄存器 \(TRIGSEL_TIMER7BRKIN\)](#)。

图 12-30. BREAK 中止功能逻辑图



BREAK 可用于处理系统源、片上外设和外部输入信号源的故障，当发生 BREAK 中止事件时，输出强制为无效电平，或在死区持续时间之后，输出将以预定的电平（有效或无效）强制输出；

当发生 BREAK 中止事件时，POEN 位被异步清除，一旦 POEN 位为 0，CHx_O 和 CHx_ON 的输出由 TIMERx_CTL1 寄存器中的 ISOx 位和 ISOxN 位确定。如果 IOS=0，定时器释放输出使能，否则输出使能仍然为高。当 IOS=1 时，通道输出情况如[图 12-31. 通道响应 BREAK 中止输入（高电平有效）时，输出信号的行为 \(IOS=1\)](#)所示，首先通道互补输出为复位状态，然后死区时间发生器重新被激活，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。

图 12-31. 通道响应 **BREAK** 中止输入（高电平有效）时，输出信号的行为（**IOS=1**）

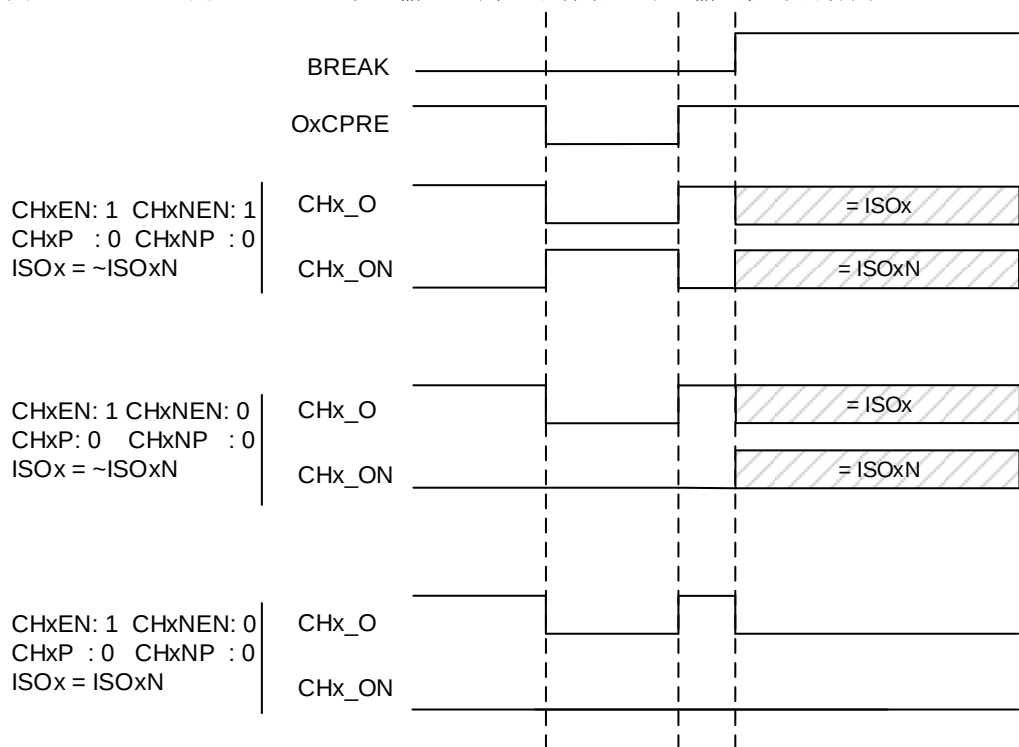


表 12-5. **BREAK** 输入信号时，**TIMER** 互补通道输出情况（**break** 输入高电平有效）

BREAK 输入	Output Status	
	CHx_O	CHx_ON
高电平	IOS=1: CHx_O 输出无效，然后在一个死区时间之后输出空闲电平（由 IOSx 位确定）。	IOS=1: CHx_ON 输出无效，然后在一个死区时间之后输出空闲电平（由 IOSxN 位确定）。
	IOS=0: CHx_O 输出禁能（无效）	IOS=0: CHx_ON 输出禁能（无效）
低电平	CHx_O 输出禁能（无效）	CHx_ON 输出禁能（无效）

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

通过配置 **TIMERx_CTL2** 寄存器中的 **BRKENCHx** ($x = 0..3$) 位，可实现对每对通道的中止功能进行独立控制。当 **BRKENCHx** ($x = 0..3$) 位为“0”且发生中止事件时，相应的通道 **CHx_O** 和 **CHx_ON** 输出保持不变。

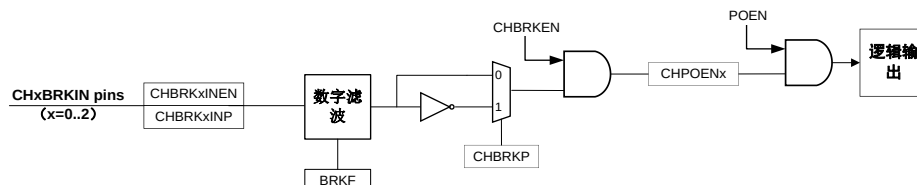
为了增强中止功能的灵活性，对于高级定时器，每对通道都有单独的中止输入 **CHxBRKIN** ($x=0..2$)，当相应的 **CHxBRKIN** ($x=0..2$) 输入有效电平时，可以异步对相应的 **CHPOENx** ($x=0..2$) 清 0，可实现单独的中止输入控制对应的每对通道中止后输出。通过在 **TIMERx_CCHP0** 寄存器中设置 **CHBRKEN** 位，可以开启通道中止。通过在 **TIMERx_CCHP0** 寄存器中设置 **CHBRKP** 位，可以配置通道中止的极性。

此外，如果启用了通道的中止功能，**CHx_O** 和 **CHx_ON** ($x=0..2$) 不仅受 **POEN** 控制，还受 **CHPOENx** ($x=0..2$) 控制。**CHx** 和 **CHx_ON** ($x=0..2$) 在被中止后的输出与 **BREAK** 功能相同。

每个通道中止输入都有自己的使能位 CHBRKxINEN(x=0...2)和极性控制位 CHBRKxINP(x=0...2), 并与 BREAK 有相同的滤波配置位域 BRKF[3:0]。请参考[图 12-32. 通道中止功能逻辑图](#)。

当通道 x(x=0...2)的 BREAK 输入有效电平时, 相应的通道中止标志位 CHxBRKIF(x=0...2)会被设置为 1。如果通道 x(x=0...2)的 BREAK 输入在多个周期内保持有效状态, 标志位 CHxBRKMIF(x=0...2)将被设置为 1。周期计数可以通过配置 TIMERx_CHBRKPER 寄存器来确定。

图 12-32. 通道中止功能逻辑图



正交译码器

正交译码器功能使用由 TIMERx_CH0 和 TIMERx_CH1 引脚生成的 CI0 和 CI1 正交信号各自相互作用产生计数值。在每个输入源改变期间, DIR 位被硬件自动改变。

输入源可以是只有 CI0, 可以只有 CI1, 或着可以同时有 CI0 和 CI1, 通过设置 TSCFGy[4:0](y=0, 1, 2) != 5'b00000 来选择使用哪种模式。计数器计数方向改变的机制如[表 12-6. 不同译码器模式下的计数方向](#)所示。其中, CI0FE0、CI1FE1 是经过滤波和极性选择后的 CI0、CI1 信号。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 12-6. 不同译码器模式下的计数方向

计数模式	电平	CI0FE0		CI1FE1	
		上升	下降	上升	下降
正交译码器模式0 TSCFG0[4:0] != 5'b00000	CI1FE1=1	向下	向上	-	-
	CI1FE1=0	向上	向下	-	-
正交译码器模式1 TSCFG1[4:0] != 5'b00000	CI0FE0=1	-	-	向上	向下
	CI0FE0=0	-	-	向下	向上
正交译码器模式2 TSCFG2[4:0] != 5'b00000	CI1FE1=1	向下	向上	X	X
	CI1FE1=0	向上	向下	X	X
	CI0FE0=1	X	X	向上	向下
	CI0FE0=0	X	X	向下	向上

注意: “-”意思是“无计数”; “X”意思是不可能。“0”意思是低电平, “1”意思是高电平。

图 12-33. 译码器接口模式下计数器运行例子

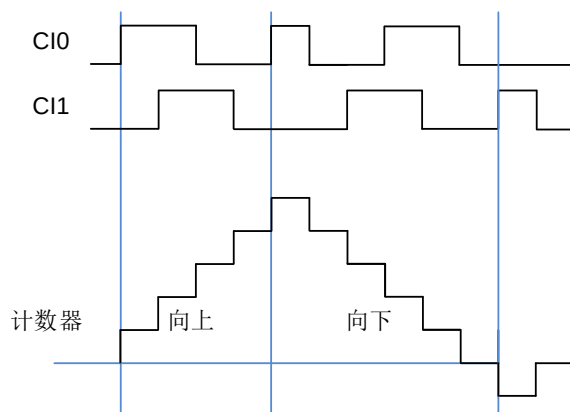
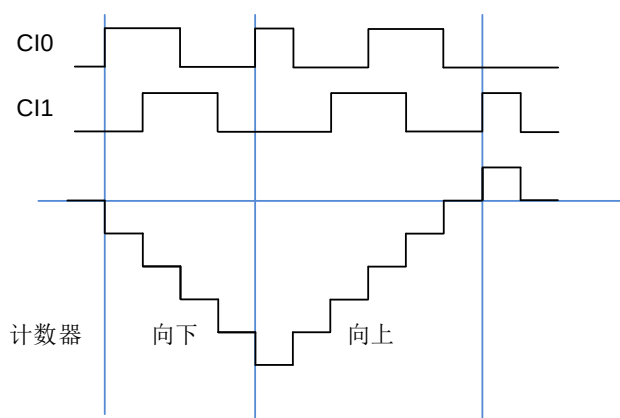


图 12-34. CI0FE0 极性反相的译码器接口模式下的例子



正交译码器和译码器的时钟输出

定时器可以通过 TRGO 输出译码器时钟输出信号。该功能仅用于正交译码器模式 0~2，通过将 `TIMERx_CTL1` 寄存器中 `MMC[3:0]` 位域设置为 `4'b1000` 来使能。

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 12-35. 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,例如 TRGO-ITIx, TIMER_in 定时器和 TIMER_out 定时器可以连接在一起。TIMER_out 定时器根据 ITIx 触发信号输出 PWM 波,驱动 BLDC 电机,控制 BLDC 电机的速度。这样, TIMER_in 定时器和 TIMER_out 定时器的连接形成了一个反馈电路,可以根据需求改变配置。

TIMER_in 定时器需要具备输入异或功能,所以可以选择高级定时器和通用 L0 定时器。

TIMER_out 定时器需要具备互补输出和死区插入功能,所以可以选择高级定时器。

另外,可以通过 TRIGSEL 模块,选择互连的定时器,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,定时器和 BLDC 的线路也已经连接好,我们就可以配置定时器了。有以下关键配置:

- 设置 TI0S, 使能异或功能。三路输入信号的任何一路发生变化, CIO 都会翻转, CH0VAL 此时会捕获计数器的当前值。
- 设置 CCUC 和 CCSE, 使能 ITIx 直接连接到换相功能。
- 根据需求配置 PWM 参数。

图 12-35. 霍尔传感器用在 BLDC 电机控制中

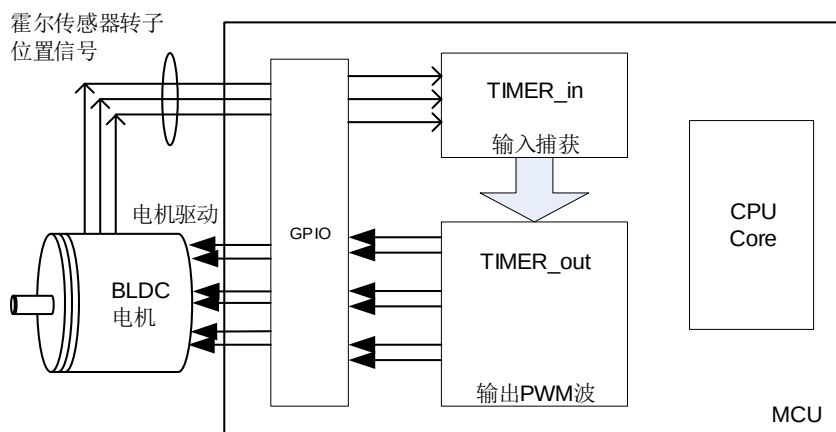
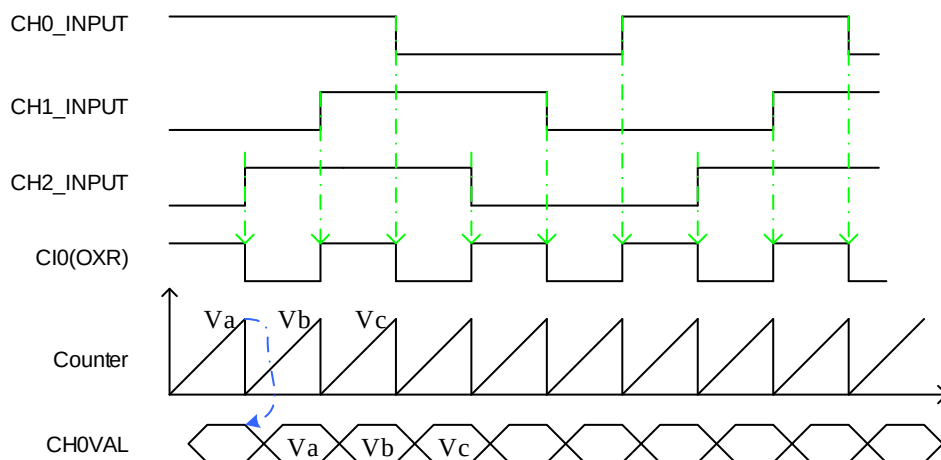
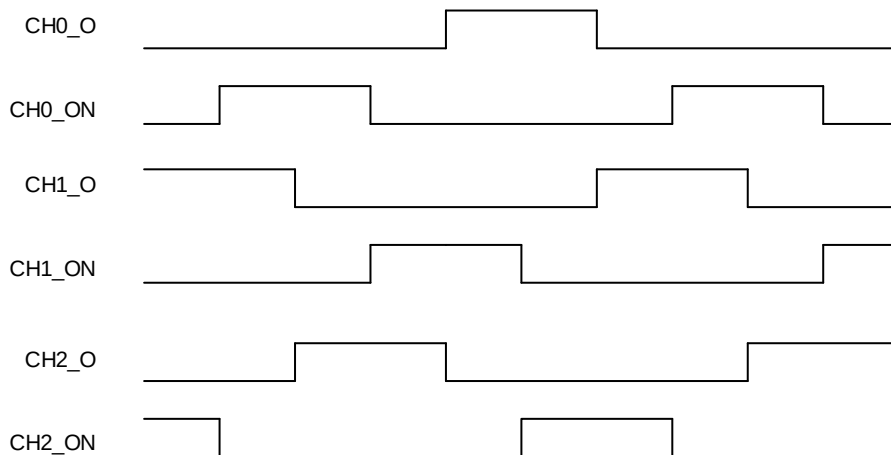


图 12-36. 两个定时器之间的霍尔传感器时序图

高级/通用L0定时器TIMER_in工作在输入捕获模式



高级定时器TIMER_out工作在输出比较模式（带有死区的PWM）



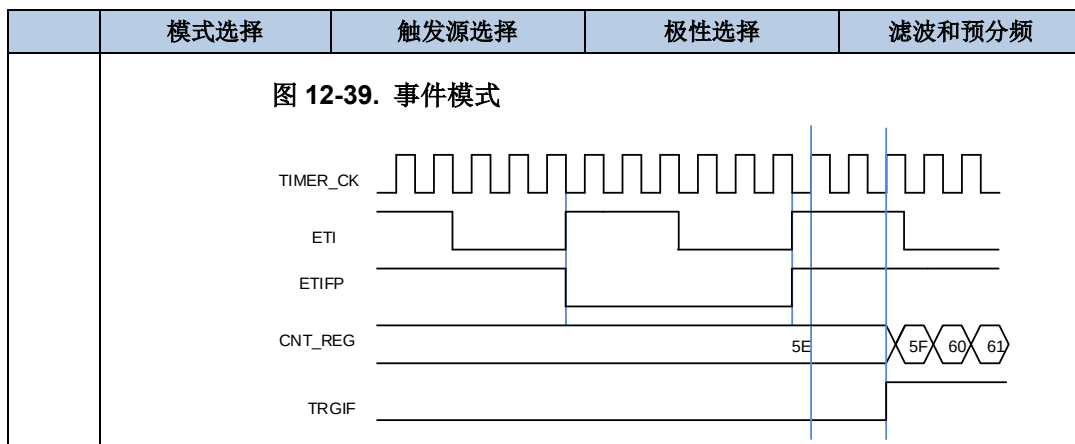
主-从管理

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等，可以通过设置 SYSCFG_TIMERxCFG(x=0,7) 寄存器中的 TSCFGy[4:0] (y=3..6) 位域来确定，具体的输入触发源可以通过 TSCFGy[4:0] (y=3..6) 位域值来选择。

表 12-7. 从模式例子列表

	模式选择	触发源选择	极性选择	滤波和预分频
列举	TSCFGy[4:0] y=3: 复位模式	TSCFGy[4:0] 00000: Mode disable	如果触发源是 ClxFEx (x=0..3)，配置 CHxP、CHxNP 来选择极性和反相。	触发源 ITIx，滤波和预分频不可用
	y=4: 暂停模式	00001: ITI0		触发源 Clx，配置 CHxCAPFLT 设置滤波，分频不可用
	y=5: 事件模式	00010: ITI1		
	y=6: 外部时钟模式0	00011: ITI2	如果触发源是 ETIFP	

	模式选择	触发源选择	极性选择	滤波和预分频
		00100: ITI3 00101: CI0F_ED 00110: CI0FE0 00111: CI1FE1 01000: ETIFP ⁽¹⁾ 01001: CI2FE2 01010: CI3FE3	(滤波后的ETI外部触发输入), 配置ETP选择极性和反相	触发源是ETIFP, 配置ETFC设置滤波, 配置ETPSC设置预分频
例1	复位模式 当触发输入上升沿, 计数器清零重启	TSCFG3[4:0] 5'b00001, 选择ITI0为触发源	触发源是ITI0, 极性选择不可用	触发源是ITI0, 滤波和预分频不可用
	图 12-37. 复位模式			
例2	暂停模式 当触发输入为低的时候, 计数器暂停计数	TSCFG4[4:0] =5'b00110, 选择CI0FE0为触发源	TI0S=0 (非异或) [CH0NP=0, CH0P=0] CI0FE0不反相, 在上升沿捕获。	在这个例子中滤波被旁路
	图 12-38. 暂停模式下的控制电路			
例3	事件模式 触发输入的上升沿计数器开始计数	TSCFG5[4:0] =5'b01000, 选择ETIFP为触发源	ETP = 0没有极性改变	ETPSC = 1, 2分频 ETFC = 0, 无滤波



单脉冲模式

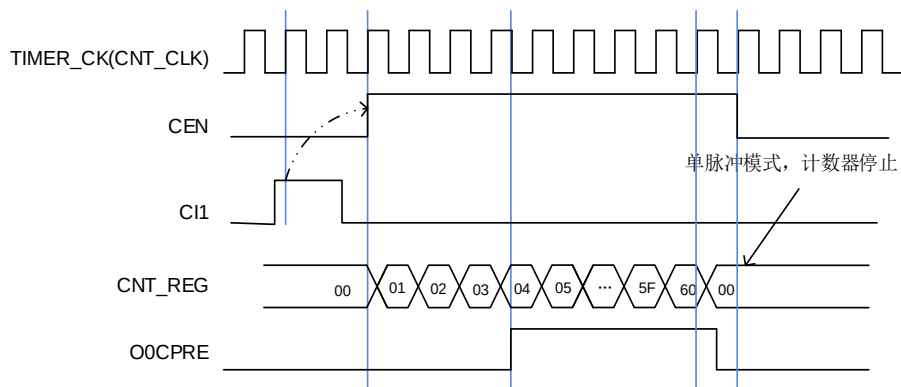
单脉冲模式与重复模式是相反的,设置 `TIMERx_CTL0` 寄存器的 `SPM` 位置 1,可启用单脉冲模式。当 `SPM` 置 1,计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波,可以通过设置 `CHxCOMCTL` 配置 `TIMERx` 为 PWM 模式或者比较模式。

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。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。

单脉冲模式也同样适用于复合 PWM 模式 (`CHxCPWMEN = 1'b1` 和 `CHxMS[2:0] = 3'b000`)。

图 12-40. 单脉冲模式, `TIMERx_CHxCV = 0x04` `TIMERx_CAR=0x60`



定时器互连

定时器之间可以内部级联或者同步，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。

互连的例子：

■ 定时器2作为定时器0的预分频器

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=4'b0010）。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. 配置定时器0工作在外部时钟模式0，定时器0输入触发源为定时器2（SYSCFG_TIMER0CFG1寄存器中的TSCFG6[4:0] = 5b'00011）；
4. 写1到CEN位启动定时器0（TIMER0_CTL0寄存器）；
5. 写1到CEN位启动定时器2（TIMER2_CTL0寄存器）。

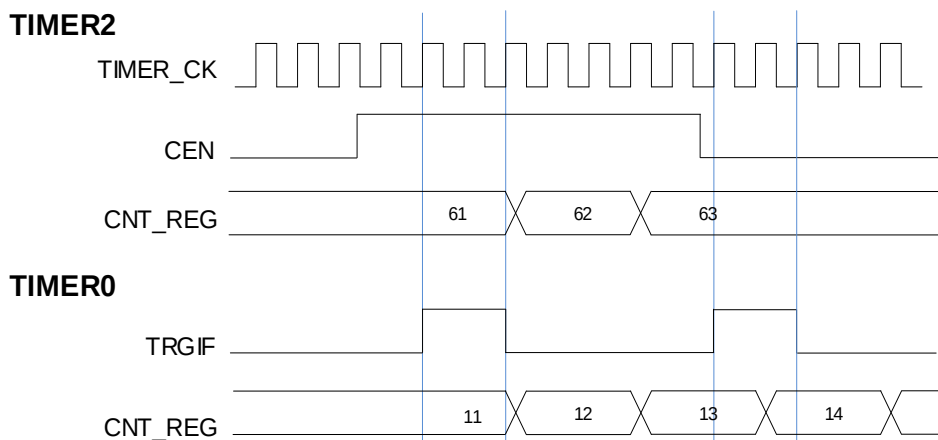
■ 用定时器2的使能/更新信号来启动定时器0

用定时器 2 的使能信号来启动定时器 0，见[图12-41. 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_CLK 经过预分频器 3 分频后频率（ $f_{CNT\_CLK} = f_{TIMER\_CK}/3$ ）。步骤如下：

1. 配置定时器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=4'b0001）；
2. 配置定时器0工作在事件模式，定时器0输入触发源为定时器2（SYSCFG_TIMER0CFG0寄存器中的TSCFG5[4:0] = 5b'00011）；
3. 写1到CEN来开启定时器2（TIMER2_CTL0寄存器）。

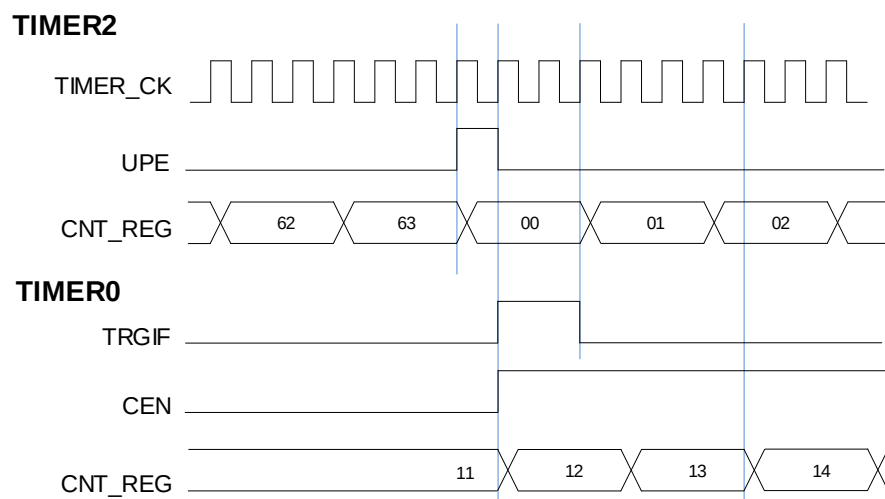
图 12-41. 用定时器 2 的使能信号触发定时器 0



在这个例子中，我们也可以使用更新事件代替使能信号作为触发源。见[图12-42. 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=4'b0010）；
2. 配置定时器2的周期（TIMER2_CARL寄存器）；
3. 配置定时器0工作在事件模式，定时器0输入触发源为定时器2（SYSCFG_TIMER0CFG0寄存器中的TSCFG5[4:0] = 5b'00011）
4. 写1到CEN来开启定时器2（TIMER2_CTL0寄存器）。

图 12-42. 用定时器 2 的更新事件来触发定时器 0

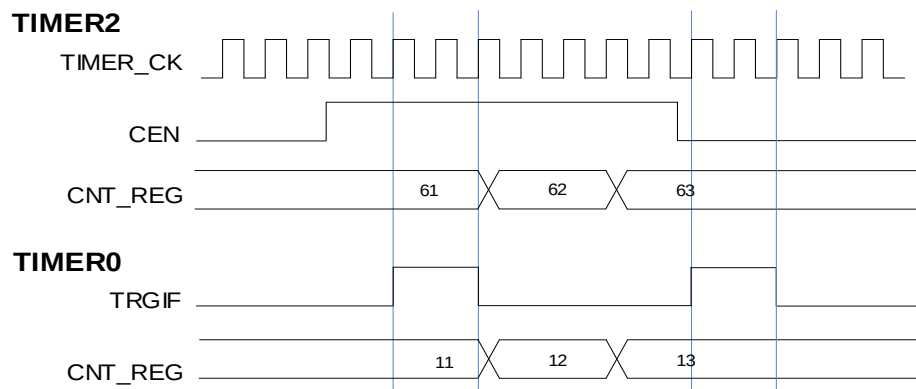


- 使用定时器2的使能/O0CPRE参考信号来使能定时器0计数。

在这个例子中，我们使用定时器2的使能输出来控制定时器0的使能。如[图12-43. 用定时器2的使能来选通定时器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（ $f_{CNT\_CLK} = f_{TIMER\_CK}/3$ ），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=4'b0001）；
2. 配置定时器0工作在暂停模式，定时器0输入触发源为定时器2（SYSCFG_TIMER0CFG0寄存器中的TSCFG4[4:0] = 5b'00011）；
3. 写1到CEN位来使能定时器0（TIMER0_CTL0寄存器）；
4. 写1到CEN位来启动定时器2（TIMER0_CTL0寄存器）；
5. 写0到CEN位来停止定时器2（TIMER0_CTL0寄存器）。

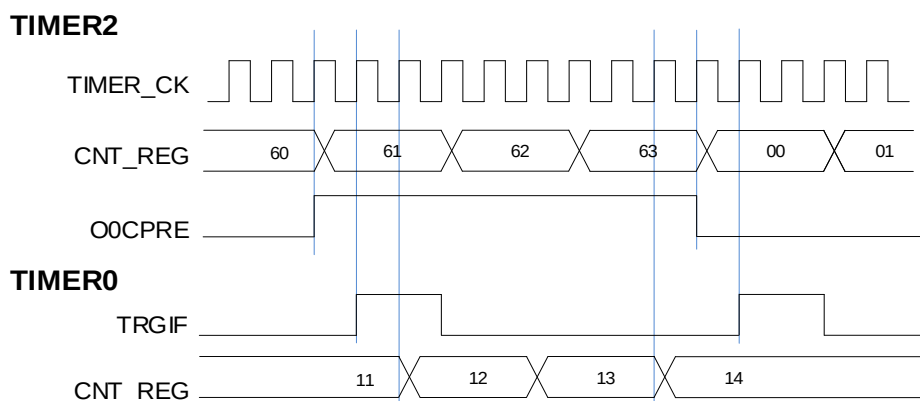
图 12-43. 用定时器 2 的使能来选通定时器 0



这个例子中，我们也可以使用定时器 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_CH0CTL寄存器）；
3. 配置定时器0工作在暂停模式，定时器0输入触发源为定时器2（SYSCFG_TIMER0CFG0寄存器中的TSCFG4[4:0] = 5b'00011）；
4. 写1到CEN位来使能定时器0（TIMER0_CTL0寄存器）；
5. 写1到CEN位来开启定时器2（TIMER0_CTL0寄存器）。

图 12-44. 用定时器 2 的 O0CPRE 信号选通定时器 0



■ 使用一个外部触发来同步两个定时器

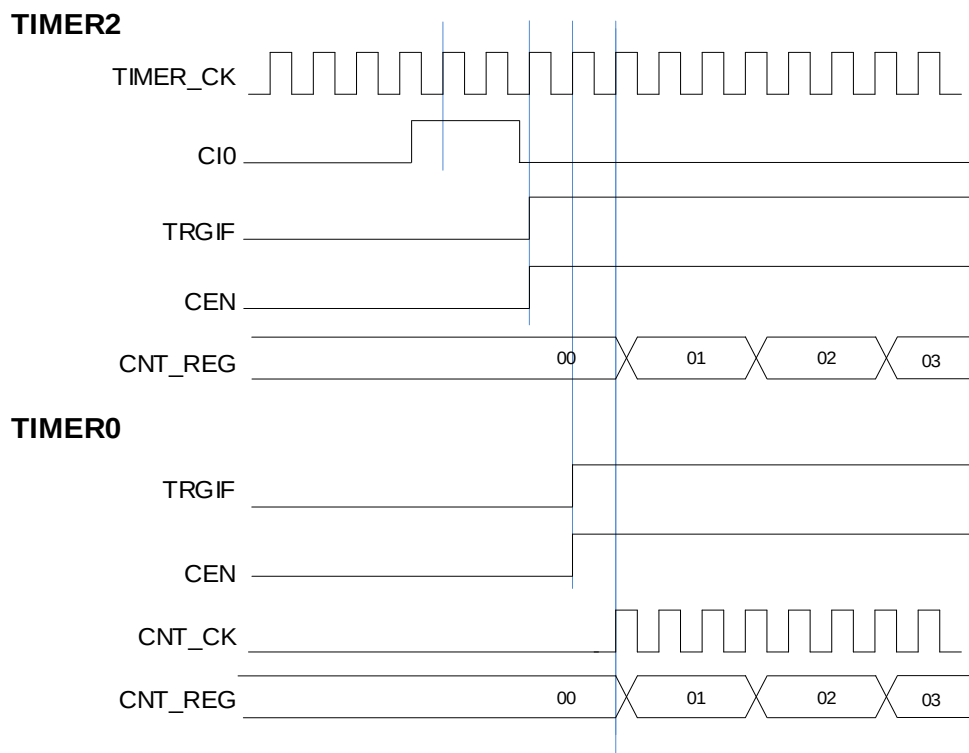
配置定时器 2 的使能信号触发定时器 0 的开启，配置定时器 2 的 CI0 输入信号上升沿来触发定时器 2。为了确保两个定时器同步开启，定时器 2 必须配置在主/从模式。步骤如下：

1. 配置定时器2工作在事件模式，定时器2输入触发源为CI0的触发输入CI0F_ED（SYSCFG_TIMER0CFG0寄存器中的TSCFG5[4:0] = 5b'00101）；

2. 写MSM=1 (TIMER2_SMCFG寄存器) 来配置定时器2工作在主/从模式;
3. 配置定时器0工作在事件模式, 定时器0输入触发源为定时器2 (SYSCFG_TIMER0CFG0寄存器中的TSCFG5[4:0] = 5b'00011)。

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 12-45. 用定时器 2 的 CI0 输入来触发定时器 0 和定时器 2



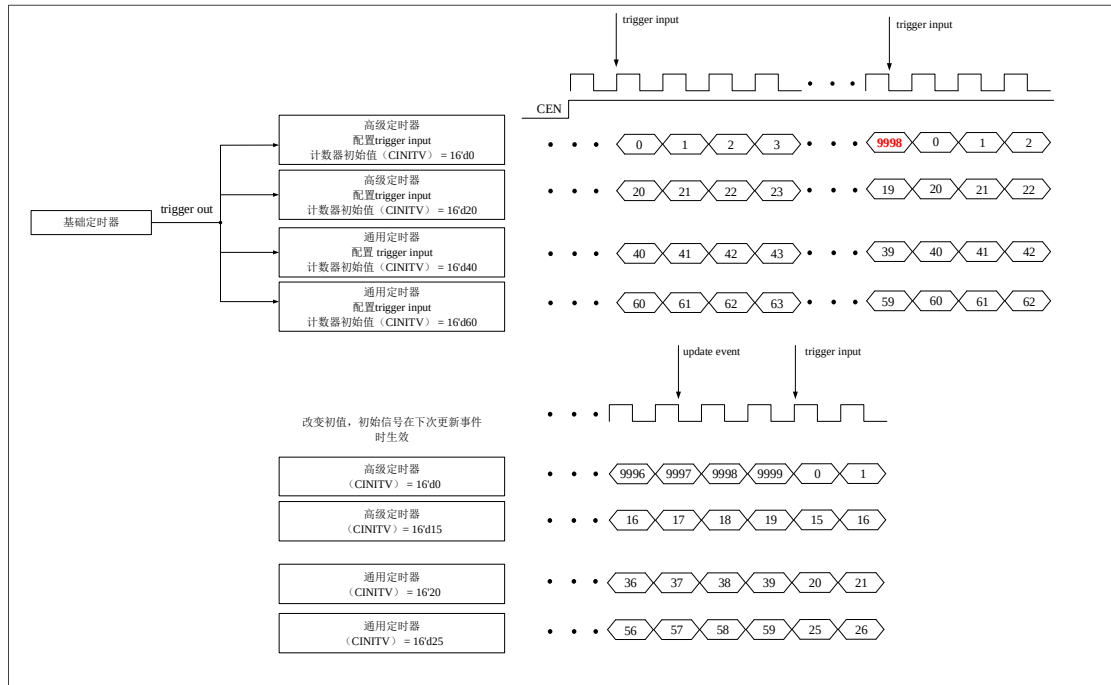
计数器同步、初始方向和值刷新

在一些菊链的配置下, 多个计数器被触发并且被同步在同一个时刻开始计数。由于不同的计数器之间在长时间计数之后可能会产生相位偏移, 因此通过外部触发定时地刷新同步计数器来消除硬件上产生的多个计数器之间产生的相位偏移。此外, 通过软件可配的计数器初始值配置 (TIMERx_CINITV), 可以控制多个被触发的计数器之间的相位偏移关系。

4 个从计数器被配置存在同一触发输入源, 通过使能 TIMERx_CINITCTL 寄存器的 CINITVEN 位域, 可以使能计数器的初值载入功能, 4 个从计数器的载入初值分别位 0, 20, 40, 60, 因此之间产生了 20 个 CK_TIMER 的等差相移。

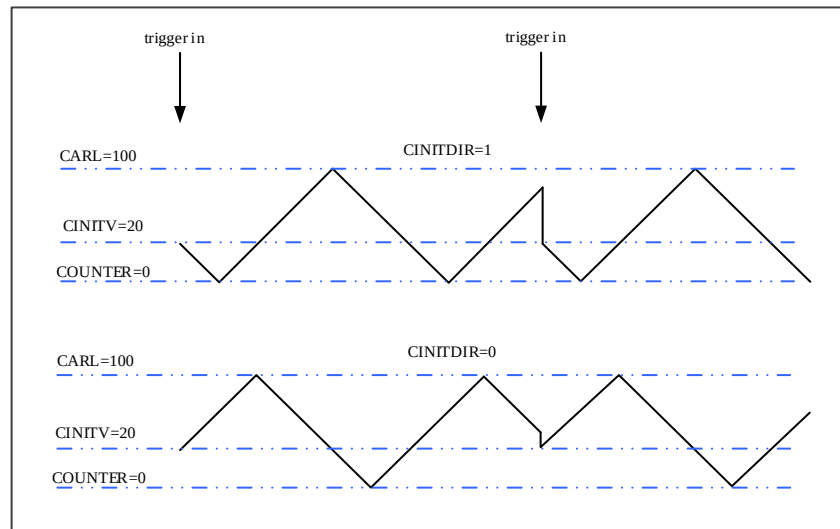
下图中的 9998 计数值, 表明了长时间计数之后, 不同计数器的之间产生了相位偏移, 通过基础定时器的触发输出, 刷新同步从定时器的计数值, 避免了相位偏移的累积。

图 12.46. 用主计数器的 trigger 信号触发 4 个计数器



当计数器处于中央对齐的计数方式时，可以通过配置 `TIMERx_CINITCTL` 寄存器的 `CINITDIR` 位域来配置计数器复位之后的计数方向。`CINITDIR` 只有当使能了 `CINITVEN` 时才会生效，另外当计数器处于向上/向下计数模式下时，`CINITDIR` 的配置也不会生效。

当 `CINITDIR` 为 0 时，复位计数器之后的计数方向为向上；当 `CINITDIR` 为 1 时，复位计数器之后的计数方向为向下。如下图所示：

图 12.47. `CINITDIR` 为 0 或 1 时的计数方向

计数器的初始计数方向和初始值也可以通过软件同步事件刷新。将 `TIMERx_CINITCTL` 寄存器中的 `SWSYNCG` 位置 1，可以产生一个软件同步事件，用于刷新 `TIMERx` 计数器的初始计数方向和

初始值。

当高级定时器使用主模式下来同步其他高级定时器时，在 `TIMERx_CTL1` 寄存器中 `MMC[3:0]` 位域需要设置为 `4'b1001`。此时，`TIMERx` 的软件同步事件（设置 `SWSYNCG` 位为 1 产生）可以作为 `TRGO` 信号输出，用于同步其他高级定时器。

定时器 DMA 模式

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

如果再来 1 次 DMA 请求事件，`TIMERx` 将会重复上面的过程。

定时器调试模式

当 Cortex®-M33 内核停止，`DBG_CTL` 寄存器中的 `TIMERx_HOLD` 位置 1 时，定时器的计数器停止计数。

12.1.5. TIMERx 寄存器 (x=0,7)

TIMER0 基地址: 0x4001 2C00

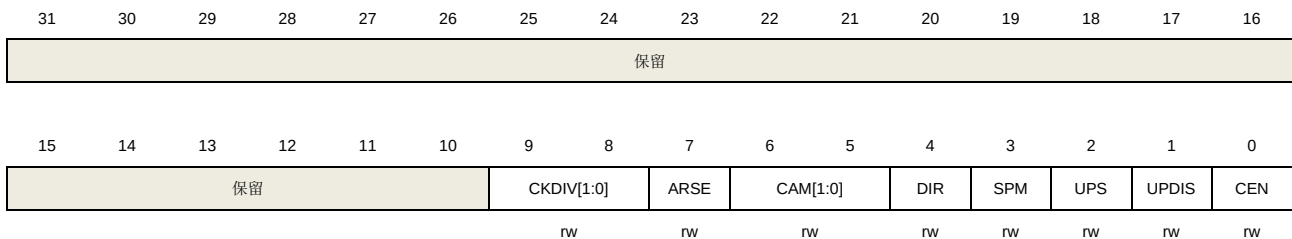
TIMER7 基地址: 0x4001 3400

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。



位/位域	名称	描述
31:10	保留	必须保持复位值
9:8	CKDIV[1:0]	时钟分频 通过软件配置 CKDIV, 规定定时器时钟 (CK_TIMER) 与死区时间和采样时钟 (DTS) 之间的分频系数, 死区发生器和数字滤波器会用到 DTS 时间。 00: $f_{DTS} = f_{CK\_TIMER}$ 01: $f_{DTS} = f_{CK\_TIMER} / 2$ 10: $f_{DTS} = f_{CK\_TIMER} / 4$ 11: 保留
7	ARSE	自动重载影子寄存器使能 0: 禁能 TIMERx_CAR 寄存器的影子寄存器 1: 使能 TIMERx_CAR 寄存器的影子寄存器
6:5	CAM[1:0]	计数器对齐模式选择 00: 无中央对齐模式 (边沿对齐模式)。DIR 位指定了计数方向 01: 中央对齐向下计数置 1 模式。计数器在中央计数模式计数, 通道被配置在输出模式 (TIMERx_CHCTL0 寄存器中 CHxMS=3'b000), 只有在向下计数时, 通道的比较中断标志置 1 10: 中央对齐向上计数置 1 模式。计数器在中央计数模式计数, 通道被配置在输出模式 (TIMERx_CHCTL0 寄存器中 CHxMS=3'b000), 只有在向上计数时, 通道的比较中断标志置 1 11: 中央对齐上下计数置 1 模式。计数器在中央计数模式计数, 通道被配置在输出模式 (TIMERx_CHCTL0 寄存器中 CHxMS=3'b000), 在向上和向下计数时, 通

道的比较中断标志都会置 1

当计数器使能以后，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译码器模式时，该位为只读。
3	SPM	单脉冲模式 0: 更新事件发生后，计数器继续计数 1: 在下次更新事件发生时，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 – 计数器上溢/下溢 – 从模式控制器产生的更新 1: 使能后只有计数器上溢/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 – 计数器上溢/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、暂停模式和译码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。

控制寄存器 1 (TIMERx_CTL1)

地址偏移: 0x04

复位值: 0x0000 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
CCUC[2:1]		保留				MMC[3]		保留							

rw					rw										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISO3N	ISO3	ISO2N	ISO2	ISO1N	ISO1	ISO0N	ISO0	TI0S	MMC[2:0]			DMAS	CCUC[0]	保留	CCSE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	

位/位域	名称	描述
31:30	CCUC[2:1]	换相控制影子寄存器更新控制 请参考 CCUC [0]的描述。
29:26	保留	必须保持复位值。
25	MMC[3]	主模式控制 请参考 MMC[2:0]的描述。
24:16	保留	必须保持复位值
15	ISO3N	通道 3 的互补通道空闲状态输出 参考 ISO0N 位。
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&CHPOENx(x=0...2)复位, CH0_ON 输出低电平 1: 当 POEN&CHPOENx(x=0...2)复位, CH0_ON 输出高电平 此位只有在 TIMERx_CCHP0 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&CHPOENx(x=0...2)复位, CH0_O 输出低电平 1: 当 POEN&CHPOENx(x=0...2)复位, CH0_O 输出高电平 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_CCHP0 寄存器的 PROT [1:0]位为 00 的时候可以被更改。
7	TI0S	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

6:4	MMC[2:0]	1: 选择 TIMERx_CH0, CH1 和 CH2 引脚异或的结果作为通道 0 的触发输入 主模式控制 该位域控制 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。 0000: 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。 0001: 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。 0010: 更新。主模式控制器选择更新事件作为 TRGO。 0011: 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。 0100: 比较。在这种模式下主模式控制器选择 O0CPRE 信号被用于作为触发输出 TRGO。 0101: 比较。在这种模式下主模式控制器选择 O1CPRE 信号被用于作为触发输出 TRGO。 0110: 比较。在这种模式下主模式控制器选择 O2CPRE 信号被用于作为触发输出 TRGO。 0111: 比较。在这种模式下主模式控制器选择 O3CPRE 信号被用于作为触发输出 TRGO。 1000: 译码器时钟输出。在这种模式下主模式控制器选择译码器时钟信号作为触发输出 TRGO。此值仅用于正交译码器模式 0~2。 1001: 在这种模式下主模式控制器选择软同步事件信号(设置 SWSYNCG 位为 1 产生)作为触发输出 TRGO。 1010~1111: 保留
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 CHx 的 DMA 请求 1: 当更新事件发生, 发送通道 CHx 的 DMA 请求
2	CCUC[0]	换相控制影子寄存器更新控制 CCUC[2:1]和 CCUC[0]位域用于控制换相控制影子寄存器的更新。当换相控制影子寄存器使能(CCSE=1), 这些影子寄存器更新根据 CCUC[2:0]位域的控制如下: 000: CMTG 位被置 1 时, 更新影子寄存器 001: 当 CMTG 位被置 1 或检测到 TRGI 上升沿时, 影子寄存器更新 100: 当计数器上溢事件发生时, 影子寄存器更新 101: 当计数器下溢事件发生时, 影子寄存器更新 110: 当计数器上溢/ 下溢事件发生时, 影子寄存器更新 其他值: 保留

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	换相控制影子寄存器使能 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	保留						
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	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	从模式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 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 TSCFGy[4:0](y=3,4,5)位域的值不能为 5b'01000。 如果外部时钟模式 0 和外部时钟模式 1 同时被使能，外部时钟的输入是 ETIFP。 注意: 外部时钟模式 0 使能在 SYSCFG_TIMERxCFG1 寄存器中的 TSCFG6[4:0] 位域。
13:12	ETPSC[1:0]	外部触发预分频 外部触发信号 ETI 的频率不能超过 TIMER_CK 频率的 1/4。当输入较快的外部时钟时，可以使用预分频降低 ETIFP 的频率。

		00: 预分频禁能
		01: ETI 频率被 2 分频
		10: ETI 频率被 4 分频
		11: ETI 频率被 8 分频
11:8	ETFC[3:0]	外部触发滤波控制 数字滤波器是一个事件计数器，它记录到 N 个事件后会产生一个输出的跳变。这些位定义了对 ETI 信号采样的频率和对 ETI 数字滤波的带宽。 0000: 滤波器禁能 $f_{SAMP} = f_{DTS}$, N=1 0001: $f_{SAMP} = f_{CK\_TIMER}$, N=2 0010: $f_{SAMP} = f_{CK\_TIMER}$, N=4 0011: $f_{SAMP} = f_{CK\_TIMER}$, N=8 0100: $f_{SAMP} = f_{DTS}/2$, N=6 0101: $f_{SAMP} = f_{DTS}/2$, N=8 0110: $f_{SAMP} = f_{DTS}/4$, N=6 0111: $f_{SAMP} = f_{DTS}/4$, N=8 1000: $f_{SAMP} = f_{DTS}/8$, N=6 1001: $f_{SAMP} = f_{DTS}/8$, N=8 1010: $f_{SAMP} = f_{DTS}/16$, N=5 1011: $f_{SAMP} = f_{DTS}/16$, N=6 1100: $f_{SAMP} = f_{DTS}/16$, N=8 1101: $f_{SAMP} = f_{DTS}/32$, N=5 1110: $f_{SAMP} = f_{DTS}/32$, N=6 1111: $f_{SAMP} = f_{DTS}/32$, N=8
7	MSM	主-从模式 该位被用来同步被选择的定时器同时开始计数。通过 TRIGI 和 TRGO，定时器被连接在一起，TRGO 用做启动事件。 0: 主从模式禁能 1: 主从模式使能
6:0	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
CH3COM	CH2COM	CH1COM	CH0COM	保留											
ADDIE	ADDIE	ADDIE	ADDIE												
IW	IW	IW	IW												

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	CH0IE	UPIE
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位/位域	名称	描述
31	CH3COMADDIE	通道 3 附加比较中断使能 0: 禁止通道 3 附加比较中断 1: 使能通道 3 附加比较中断 注意: 此中断使能位仅用于复合 PWM 模式。
30	CH2COMADDIE	通道 2 附加比较中断使能 0: 禁止通道 2 附加比较中断 1: 使能通道 2 附加比较中断 注意: 此中断使能位仅用于复合 PWM 模式。
29	CH1COMADDIE	通道 1 附加比较中断使能 0: 禁止通道 1 附加比较中断 1: 使能通道 1 附加比较中断 注意: 此中断使能位仅用于复合 PWM 模式。
28	CH0COMADDIE	通道 0 附加比较中断使能 0: 禁止通道 0 附加比较中断 1: 使能通道 0 附加比较中断 注意: 此中断使能位仅用于复合 PWM 模式。
27:15	保留	必须保持复位值。
14	TRGDEN	触发 DMA 请求使能 0: 禁止触发 DMA 请求 1: 使能触发 DMA 请求
13	CMTDEN	换相 DMA 更新请求使能 0: 禁止换相 DMA 更新请求 1: 使能换相 DMA 更新请求
12	CH3DEN	通道 3 比较/捕获 DMA 请求使能 0: 禁止通道 3 比较/捕获 DMA 请求 1: 使能通道 3 比较/捕获 DMA 请求
11	CH2DEN	通道 2 比较/捕获 DMA 请求使能 0: 禁止通道 2 比较/捕获 DMA 请求 1: 使能通道 2 比较/捕获 DMA 请求
10	CH1DEN	通道 1 比较/捕获 DMA 请求使能 0: 禁止通道 1 比较/捕获 DMA 请求

		1: 使能通道 1 比较/捕获 DMA 请求
9	CH0DEN	通道 0 比较/捕获 DMA 请求使能 0: 禁止通道 0 比较/捕获 DMA 请求 1: 使能通道 0 比较/捕获 DMA 请求
8	UPDEN	更新 DMA 请求使能 0: 禁止更新 DMA 请求 1: 使能更新 DMA 请求
7	BRKIE	中止中断使能 0: 禁止中止中断 1: 使能中止中断
6	TRGIE	触发中断使能 0: 禁止触发中断 1: 使能触发中断
5	CMTIE	换相更新中断使能 0: 禁止换相更新中断 1: 使能换相更新中断
4	CH3IE	通道 3 比较/捕获中断使能 0: 禁止通道 3 中断 1: 使能通道 3 中断
3	CH2IE	通道 2 比较/捕获中断使能 0: 禁止通道 2 中断 1: 使能通道 2 中断
2	CH1IE	通道 1 比较/捕获中断使能 0: 禁止通道 1 中断 1: 使能通道 1 中断
1	CH0IE	通道 0 比较/捕获中断使能 0: 禁止通道 0 中断 1: 使能通道 0 中断
0	UPIE	更新中断使能 0: 禁止更新中断 1: 使能更新中断

中断标志寄存器 (TIMERx_INTF)

地址偏移: 0x10

复位值: 0x0000 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
CH3COM ADDIF	CH2COM ADDIF	CH1COM ADDIF	CH0COM ADDIF	保留											
rc_w0	rc_w0	rc_w0	rc_w0												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	SYSBIF	CH3OF	CH2OF	CH1OF	CH0OF	保留	BRKIF	TRGIF	CMTIF	CH3IF	CH2IF	CH1IF	CH0IF	UPIF	
	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位/位域	名称	描述
31	CH3COMADDIF	通道 3 附加比较中断标志 参见 CH0COMADDIF 描述。
30	CH2COMADDIF	通道 2 附加比较中断标志 参见 CH0COMADDIF 描述。
29	CH1COMADDIF	通道 1 附加比较中断标志 参见 CH0COMADDIF 描述。
28	CH0COMADDIF	通道 0 附加比较中断标志 此标志由硬件置 1 软件清 0。当通道 0 用于输出模式时，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 注意： 此标志仅用于复合 PWM 模式。
27:14	保留	必须保持复位值
13	SYSBIF	系统源中止事件中断标志位 当系统中止源有效时，该位由硬件置1，当系统源无效时，该位由软件清零。 0：无系统中止事件中断发生 1：系统中止事件中断发生 注意： 当该位置1时，在通道输出恢复前，该位必须由软件清零。
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：捕获溢出中断发生
8	保留	必须保持复位值
7	BRKIF	BREAK 中断标志位 一旦 BREAK 输入有效，由硬件对该位置‘1’。如果 BREAK 输入无效，则该位可由软件清‘0’。 0：无 BREAK 事件产生 1： BREAK 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。 当通道 0 在输入模式下时，通过读 <code>TIMERx_CH0CV</code> 寄存器可以清零该位。 0：无通道 0 中断发生 1：通道 0 中断发生
0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置 1，软件清 0。 0：无更新中断发生 1：发生更新中断

软件事件产生寄存器 (TIMERx_SWEVG)

地址偏移: 0x14

复位值: 0x0000 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
CH3COM ADDG	CH2COM ADDG	CH1COM ADDG	CH0COM ADDG	保留											
W	W	W	W												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								BRKG	TRGG	CMTG	CH3G	CH2G	CH1G	CH0G	UPG
								W	W	W	W	W	W	W	W

位/位域	名称	描述
31	CH3COMADDG	通道 3 附加比较事件发生 参见 CH0COMADDG 描述。
30	CH2COMADDG	通道 2 附加比较事件发生 参见 CH0COMADDG 描述。
29	CH1COMADDG	通道 1 附加比较事件发生 参见 CH0COMADDG 描述。
28	CH0COMADDG	通道 0 附加比较事件发生 该位由软件置 1，用于在通道 0 产生一个比较事件，由硬件自动清 0。当此位被置 1，CH0COMADDIF 标志位被置 1，若开启对应的中断和 DMA，则发出相应的中断请求。 0: 不产生通道 0 附加比较事件 1: 发生通道 0 附加比较事件 注意: 此位仅用于复合 PWM 模式。
27:8	保留	必须保持复位值
7	BRKG	产生 BREAK 事件 该位由软件置 1，用于产生一个 BREAK 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 BREAK 事件 1: 产生 BREAK 事件
6	TRGG	触发事件产生 此位由软件置 1，由硬件自动清 0。当此位被置 1，TIMERx_INTF 寄存器的 TRGIF 标志位被置 1，若开启对应的中断和 DMA，则产生相应的中断和 DMA 传输。 0: 无触发事件产生

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

通道控制寄存器 0 (TIMERx_CHCTL0)

地址偏移: 0x18

复位值: 0x0000 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
CH1MS	CH0MS	CH1COM	CH0COM	CH1ADDU	CH0ADDU		CH1COM								CH0COM
[2]	[2]	ADDSEN	ADDSEN	PS	PS	保留	CTL[3]								CTL[3]
							保留								保留
r/w	r/w	r/w	r/w	r/w	r/w		r/w								r/w
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	CH1MS[2]	通道 1 I/O 模式选择 参考 CH1MS[1:0]描述。
30	CH0MS[2]	通道 0 I/O 模式选择 参考 CH0MS[1:0]描述。
29	CH1COMADDSSEN	通道 1 附加输出比较影子寄存器使能 参考 CH0COMADDSSEN 描述。
28	CH0COMADDSSEN	通道 0 附加输出比较影子寄存器使能 当此位被置 1, <code>TIMERx_CH0COMV_ADD</code> 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 附加比较输出影子寄存器 1: 使能通道 0 附加比较输出影子寄存器 仅在单脉冲模式下 (<code>TIMERx_CTL0</code> 寄存器的 <code>SPM = 1</code>), 可以在未确认预装载寄存器情况下使用 PWM 模式。 当 <code>TIMERx_CCHP</code> 寄存器的 <code>PROT [1:0] = 11</code> 且 <code>CH0MS = 000</code> 时此位不能被改变。
27	CH1ADDUPS	通道 1 附加寄存器更新源 0: 在发生更新事件时, 更新 <code>TIMERx_CH1COMV_ADD</code> 寄存器 1: 在发生计数器计数值匹配 <code>CH1VAL</code> 值时, 更新 <code>TIMERx_CH1COMV_ADD</code> 寄存器
26	CH0ADDUPS	通道 0 附加寄存器更新源 0: 在发生更新事件时, 更新 <code>TIMERx_CH0COMV_ADD</code> 寄存器 1: 在发生计数器计数值匹配 <code>CH0VAL</code> 值时, 更新 <code>TIMERx_CH0COMV_ADD</code> 寄存器
25	保留	必须保持复位值
24	CH1COMCTL[3]	通道 1 输出比较控制 参见 CH0COMCTL[2:0]描述
23:17	保留	必须保持复位值
16	CH0COMCTL[3]	通道 0 输出比较控制 参见 CH0COMCTL[2:0]描述
15	CH1COMCEN	通道 1 输出比较清 0 使能 参见 CH0COMCEN 描述

14:12	CH1COMCTL[2:0]	通道 1 输出比较控制 参见 CH0COMCTL[2:0]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 CH1MS[2:0]位域定义了通道的方向和输入信号的选择。只有当通道关闭（当 TIMERx_CHCTL2 寄存器的 CH1EN 位清 0）时，这些位才可以写。 000：通道 1 配置为输出 001：通道 1 配置为输入，IS1 映射在 CI1FE1 上 010：通道 1 配置为输入，IS1 映射在 CI0FE1 上 011：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 SYSCFG_TIMERxCFG2(x=0,7)寄存器中的 TSCFG15[4:0]位域选择）。 100~111：保留
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 输出比较控制 CH0COMCTL[3]和 CH0COMCTL[2:0]位域定义了输出准备信号 O0CPRE 的动作，而 O0CPRE 决定了 CH0_O 和 CH0_ON 的值。O0CPRE 高电平有效，而 CH0_O 和 CH0_ON 的有效电平取决于 CH0P 和 CH0NP 位。 0000：时基。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 0001：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。 0010：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。 0011：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。 0100：强制为低。强制 O0CPRE 为低电平。 0101：强制为高。强制 O0CPRE 为高电平。 0110：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。 0111：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。

		1000: 保留。
		1001: 保留。
		1010: 非对称 PWM 模式 0, 仅对中央对齐模式有效, 在向上计数时, 一旦计数器值等于 <code>TIMERx_CH0CV</code> 时, 输出强制为无效电平。在向下计数时, 一旦计数器值等于 <code>TIMERx_CH0CV_ADD</code> 时, 输出强制为有效电平。
		1011: 非对称 PWM 模式 1, 仅对中央对齐模式有效, 在向上计数时, 一旦计数器值等于 <code>TIMERx_CH0CV</code> 时, 输出强制为有效电平。在向下计数时, 一旦计数器值等于 <code>TIMERx_CH0CV_ADD</code> 时, 输出强制为无效电平。
		1100~1111: 保留
		注意: 在复合 PWM 模式下 (<code>CH0CPWMEN = 1'b1</code> 和 <code>CH0MS = 3'b000</code>), 通道 0 的 PWM 输出信号由 <code>TIMERx_CH0CV</code> 和 <code>TIMERx_CH0COMV_ADD</code> 寄存器共同确定。详细信息请参考 复合 PWM 模式 。
		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时基模式切换到 PWM 模式时, <code>O0CPRE</code> 电平才改变。
		当 <code>CH0_O</code> 和 <code>CH0_ON</code> 输出互补时, 该位域预装载。若 <code>CCSE = 1</code> , 则该位域只在通道换相事件发生时更新。
		当 <code>TIMERx_CCHP0</code> 寄存器的 <code>PROT [1:0]=11</code> 且 <code>CH0MS = 000</code> (比较模式) 时, 此位不能被改变。
3	CH0COMSEN	通道 0 输出比较影子寄存器使能 当此位被置 1, <code>TIMERx_CH0CV</code> 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 (<code>TIMERx_CTL0</code> 寄存器的 <code>SPM = 1</code>), 可以在未确认预装载寄存器情况下使用 PWM 模式。 当 <code>TIMERx_CCHP0</code> 寄存器的 <code>PROT [1:0]=11</code> 且 <code>CH0MS = 000</code> 时此位不能被改变。
2	CH0COMFEN	通道 0 输出比较快速使能 当该位为 1 时, 如果通道配置为 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, <code>CH0_O</code> 被设置为比较电平而与比较结果无关。 0: 通道 0 输出比较快速禁能。当触发器的输入有一个有效沿时, 激活 <code>CH0_O</code> 输出的最小延时为 5 个时钟周期 1: 通道 0 输出比较快速使能。当触发器的输入有一个有效沿时, 激活 <code>CH0_O</code> 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 (当 <code>TIMERx_CHCTL2</code> 寄存器的 <code>CH0EN</code> 位清 0) 时, <code>CH0MS[2:0]</code> 才可写。 000: 通道 0 配置为输出 001: 通道 0 配置为输入, <code>IS0</code> 映射在 <code>CI0FE0</code> 上

010: 通道 0 配置为输入, IS0 映射在 CI1FE0 上

011: 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 (由 SYSCFG_TIMERxCFG2(x=0,7)寄存器中的 TSCFG15[4:0]位域选择)。

100~111: 保留

输入捕获模式:

位/位域	名称	描述
31	CH1MS[2]	通道 1 模式选择 与输出模式相同。
30	CH0MS[2]	通道 0 模式选择 与输出模式相同。
29:16	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, 它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 CI0 输入信号的采样频率和数字滤波器的长度。 0000: 无滤波器, $f_{SAMP} = f_{DTS}$, $N=1$ 0001: $f_{SAMP} = f_{CK\_TIMER}$, $N=2$ 0010: $f_{SAMP} = f_{CK\_TIMER}$, $N=4$ 0011: $f_{SAMP} = f_{CK\_TIMER}$, $N=8$ 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 <code>TIMERx_CHCTL2</code> 寄存器中的 <code>CH0EN = 0</code> 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	
CH3MS [2]	CH2MS [2]	CH3COM	CH2COM	CH3ADDU	CH2ADDU	保留	CH3COM	保留							CH2COM	
		ADDSEN	ADDSEN	PS	PS		CTL[3]								CTL[3]	
		保留	保留	保留	保留		保留								保留	
rw	rw	rw	rw	rw	rw	rw										rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
CH3CO MCEN	CH3COMCTL[2:0]			CH3CO MSEN	CH3CO MFEN	CH3MS[1:0]			CH2CO MCEN	CH2COMCTL[2:0]		CH2CO MSEN	CH2CO MFEN	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	CH3MS[2]	通道 3 I/O 模式选择 参考 CH3MS[1:0]描述。
30	CH2MS[2]	通道 2 I/O 模式选择 参考 CH2MS[1:0]描述。
29	CH3COMADDSEN	通道 3 附加输出比较影子寄存器使能 参考 CH2COMADDSEN 描述。
28	CH2COMADDSEN	通道 2 附加输出比较影子寄存器使能 当此位被置 1, <code>TIMERx_CH2COMV_ADD</code> 寄存器的影子寄存器使能，影子寄存器在每次更新事件时都会被更新。 0: 禁止通道 2 附加输出/比较影子寄存器 1: 使能通道 2 附加输出/比较影子寄存器 仅在单脉冲模式下 (<code>TIMERx_CTL0</code> 寄存器的 <code>SPM = 1</code>)，可以在未确认预装载寄

		存器情况下使用 PWM 模式。 当 <code>TIMERx_CCHP</code> 寄存器的 <code>PROT [1:0]=11</code> 且 <code>CH2MS =000</code> 时此位不能被改变。
27	CH3ADDUPS	通道3附加寄存器更新源 0: 在发生更新事件时, 更新 <code>TIMERx_CH3COMV_ADD</code> 寄存器 1: 在发生计数器计数值匹配 <code>CH3VAL</code> 值时, 更新 <code>TIMERx_CH3COMV_ADD</code> 寄存器
26	CH2ADDUPS	通道2附加寄存器更新源 0: 在发生更新事件时, 更新 <code>TIMERx_CH2COMV_ADD</code> 寄存器 1: 在发生计数器计数值匹配 <code>CH2VAL</code> 值时, 更新 <code>TIMERx_CH2COMV_ADD</code> 寄存器
25	保留	必须保持复位值.
24	CH3COMCTL[3]	通道 3 输出比较控制 请参考 <code>CH2COMCTL[2:0]</code> 描述
23:17	保留	必须保持复位值.
16	CH2COMCTL[3]	通道 2 输出比较控制 请参考 <code>CH2COMCTL[2:0]</code> 描述
15	CH3COMCEN	通道 3 输出比较清 0 使能 参见 <code>CH0COMCEN</code> 描述。
14:12	CH3COMCTL[2:0]	通道 3 输出比较控制 参见 <code>CH0COMCTL</code> 描述。
11	CH3COMSEN	通道 3 输出比较影子寄存器使能 参见 <code>CH0COMSEN</code> 描述。
10	CH3COMFEN	通道3输出比较快速使能 参见 <code>CH2COMFEN</code> 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（当 <code>TIMERx_CHCTL2</code> 寄存器的 <code>CH3EN</code> 位清 0）时，这些位才可以写。 000: 通道 3 配置为输出 001: 通道 3 配置为输入, <code>IS3</code> 映射在 <code>CI3FE3</code> 上 010: 通道 3 配置为输入, <code>IS3</code> 映射在 <code>CI2FE3</code> 上 011: 通道 3 配置为输入, <code>IS3</code> 映射在 <code>ITS</code> 上, 此模式仅工作在内部触发器输入被选中时（由 <code>SYSCFG_TIMERxCFG2(x=0,7)</code> 寄存器中的 <code>TSCFG15[4:0]</code> 位域选择）。 100~111: 保留
7	CH2COMCEN	通道 2 输出比较清 0 使能 当此位被置 1, 当检测到 <code>ETIFP</code> 输入高电平时, <code>O2CPRE</code> 参考信号被清 0

6:4

CH2COMCTL[2:0]

0: 使能通道 2 输出比较清零

1: 禁止通道 2 输出比较清零

通道 2 输出比较控制

此位定义了输出准备信号 O2CPRE 的动作，而 O2CPRE 决定了 CH2_O 和 CH2_ON 的值。O2CPRE 高电平有效，而 CH2_O 和 CH2_ON 的有效电平取决于 CH2P 和 CH2NP 位。

0000: 时基。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

0001: 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。

0010: 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。

0011: 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。

00100: 强制为低。强制 O2CPRE 为低电平

101: 强制为高。强制 O2CPRE 为高电平

0110: 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。

0111: 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。

1000: 保留。

1001: 保留。

1010: 非对称 PWM 模式 0，仅对中央对齐模式有效，在向上计数时，一旦计数器值等于 TIMERx_CH0CV 时，输出强制为无效电平。在向下计数时，一旦计数器值等于 TIMERx_CH0CV_ADD 时，输出强制为有效电平。

1011: 非对称 PWM 模式 1，仅对中央对齐模式有效，在向上计数时，一旦计数器值等于 TIMERx_CH0CV 时，输出强制为有效电平。在向下计数时，一旦计数器值等于 TIMERx_CH0CV_ADD 时，输出强制为无效电平。

1100~1111: 保留

注意：在复合 PWM 模式下（CH2CPWMEN = 1'b1 和 CH2MS = 3'b000），通道 2 的 PWM 输出信号由 TIMERx_CH2CV 和 TIMERx_CH2COMV_ADD 寄存器共同确定。详细信息请参考[复合 PWM 模式](#)。

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时基模式切换到 PWM 模式时，O2CPRE 电平才改变。

当 CH2_O 和 CH2_ON 输出互补时，该位域预装载。若 CCSE = 1，则该位域只在通道换相事件发生时更新。

当 TIMERx_CCHP0 寄存器的 PROT [1:0]=11 且 CH2MS = 000（比较模式）时此位不能被改变。

3	CH2COMSEN	通道 0 输出比较影子寄存器使能 当此位被置 1，TIMERx_CH2CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 2 输出/比较影子寄存器 1：使能通道 2 输出/比较影子寄存器 仅在单脉冲模式下（TIMERx_CTL0 寄存器的 SPM =1），可以在未确认预装载寄存器情况下使用 PWM 模式。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH2MS =000 时此位不能被改变。
2	CH2COMFEN	通道2输出比较快速使能 当该位为1时，如果通道配置为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：通道2输出比较快速禁能。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：通道0输出比较快速使能。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（当 TIMERx_CHCTL2 寄存器的 CH2EN 位清 0）时，这些位才可写。 000：通道 2 配置为输出 001：通道 2 配置为输入，IS2 映射在 CI2FE2 上 010：通道 2 配置为输入，IS2 映射在 CI3FE2 上 011：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（由 SYSCFG_TIMERxCFG2(x=0,7)寄存器中的 TSCFG15[4:0]位域选择）。 100~111：保留

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，它记录 N 个输入事件后会产生一个输出的跳

变。这些位定义了 CI2 输入信号的采样频率和数字滤波器的长度。

0000: 无滤波器, $f_{SAMP} = f_{DTS}$, $N=1$

0001: $f_{SAMP} = f_{CK\_TIMER}$, $N=2$

0010: $f_{SAMP} = f_{CK\_TIMER}$, $N=4$

0011: $f_{SAMP} = f_{CK\_TIMER}$, $N=8$

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 <code>TIMERx_CHCTL2</code> 寄存器中的 <code>CH2EN = 0</code> 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
CH3PER	CH2PER	CH1PER	CH0PER	保留											
FOREN	FOREN	FOREN	FOREN												
RW	RW	RW	RW												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3NP	CH3NEN	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	RW	RW

位/位域	名称	描述
31	CH3PERFOREN	Channel 3在复合PWM或者非对称PWM模式下，周期点匹配时刻O3CPRE电平值设置。 参考CH0PERFOREN描述。
30	CH2PERFOREN	Channel 2在复合PWM或者非对称PWM模式下，周期点匹配时刻O2CPRE电平值设置。 参考CH0PERFOREN描述。
29	CH1PERFOREN	Channel 1在复合PWM或者非对称PWM模式下，周期点匹配时刻O1CPRE电平值设置。 参考CH0PERFOREN描述。
28	CH0PERFOREN	Channel 0在复合PWM或者非对称PWM模式下，周期点匹配时刻O0CPRE电平值设置。 0：禁能 1：复合PWM0或者非对称PWM0模式下，周期点匹配时刻电平值强制拉高，复合PWM1或者非对称PWM1模式下，周期点匹配时刻电平值强制拉低。 注意：中央对齐模式下，周期点为计数器下溢时刻；边沿对齐模式下，周期点为计数器溢出时刻。
27:16	保留	必须保持复位值
15	CH3NP	通道 3 互补捕获/比较极性 参考 CH0NP 描述。
14	CH3NEN	通道 3 互补捕获/比较使能 参考 CH0NEN 描述。
13	CH3P	通道 3 捕获/比较极性 参考 CH0P 描述。
12	CH3EN	通道 3 捕获/比较使能 参考 CH0EN 描述。
11	CH2NP	通道 2 互补捕获/比较极性 参考 CH0NP 描述。
10	CH2NEN	通道 2 互补捕获/比较使能 参考 CH0NEN 描述。
9	CH2P	通道 2 捕获/比较极性 参考 CH0P 描述。
8	CH2EN	通道 2 捕获/比较使能 参考 CH0EN 描述。

7	CH1NP	通道 1 互补捕获/比较极性 参考 CH0NP 描述。
6	CH1NEN	通道 1 互补捕获/比较使能 参考 CH0NEN 描述。
5	CH1P	通道 1 捕获/比较极性 参考 CH0P 描述。
4	CH1EN	通道 1 捕获/比较使能 参考 CH0EN 描述。
3	CH0NP	通道 0 互补捕获/比较极性 当通道 0 配置为输出模式，此位定义了通道 0 输出信号 CH0_ON 的极性。 0: 通道 0 互补输出高电平有效 1: 通道 0 互补输出低电平有效 当通道 0 配置为输入模式时，此位和 CH0P 联合使用，作为通道 0 的极性选择控制信号。 当 TIMEx_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_ON 信号有效。 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 捕获/比较极性 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，此位定义了通道 0 输入信号的极性。[CH0NP, CH0P] 用于选择通道 0 输入信号信号有效边沿或者捕获极性。 00: 把通道 0 输入信号的上升沿作为捕获或者从模式下触发的有效信号，且通道 0 输入信号不会被翻转。 01: 把通道 0 输入信号的下降沿作为捕获或者从模式下触发的有效信号，且通道 0 输入信号会被翻转。 10: 保留。 11: 把通道 0 输入信号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，且通道 0 输入信号不翻转。 当 TIMEx_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

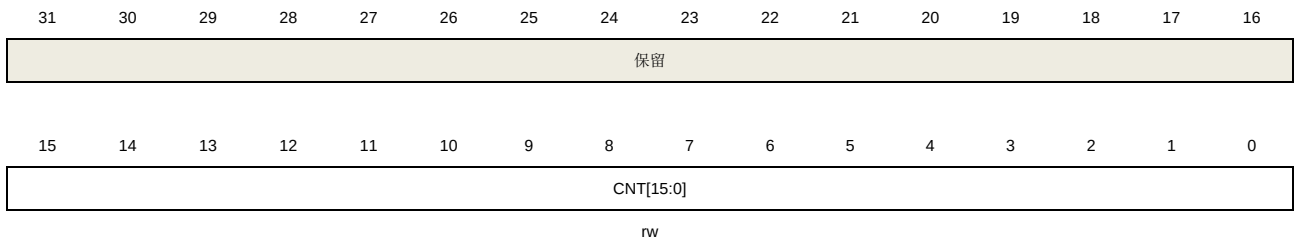
1: 使能通道 0

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。



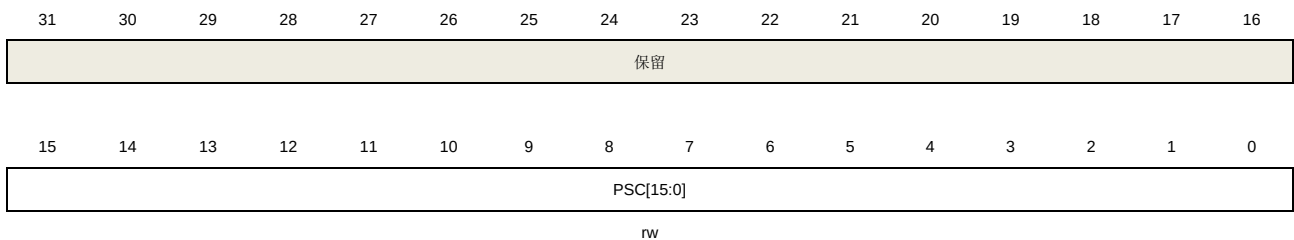
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

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。



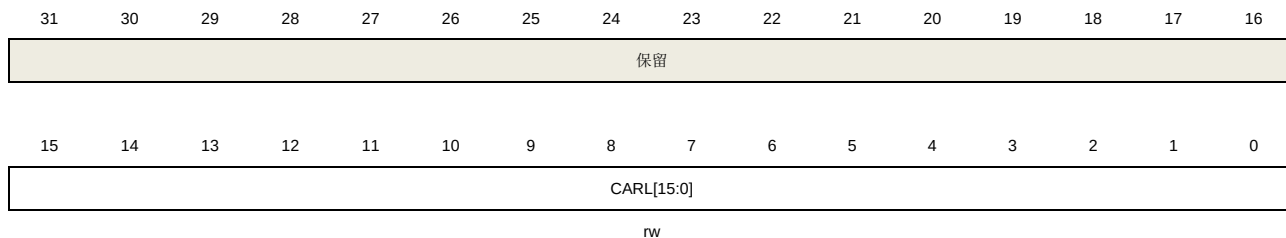
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 PSC 时钟除以 (PSC+1)，每次当更新事件产生时，PSC 的值被装入当前预分频寄存器。

计数器自动重载寄存器 (TIMERx_CAR)

地址偏移: 0x2C

复位值：0x0000 FFFF

该寄存器只能按字（32 位）访问。



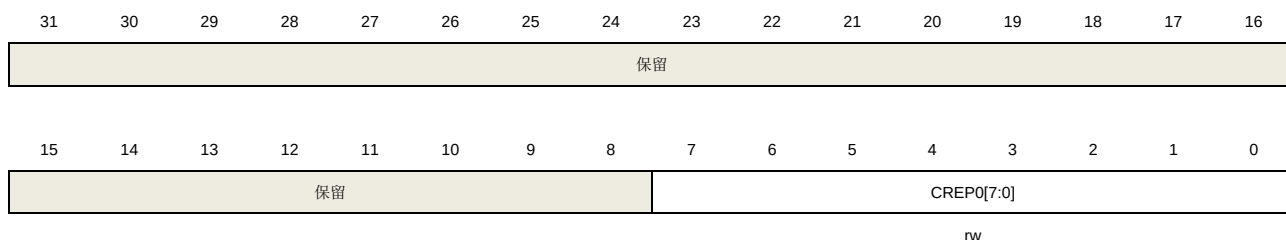
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

重复计数寄存器 0（TIMERx_CREP0）

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
31:8	保留	必须保持复位值。
7:0	CREP0[7:0]	重复计数器的值 0 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）。 注意： 当 TIMERx_CFG 寄存器中的 CREPSEL =0 时，使用该位。

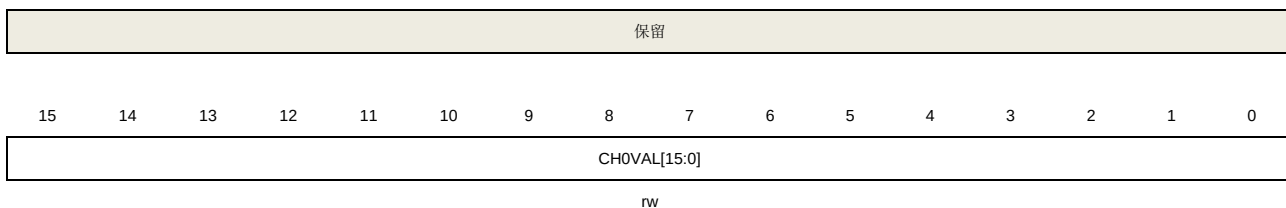
通道 0 捕获/比较寄存器（TIMERx_CH0CV）

地址偏移：0x34

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。





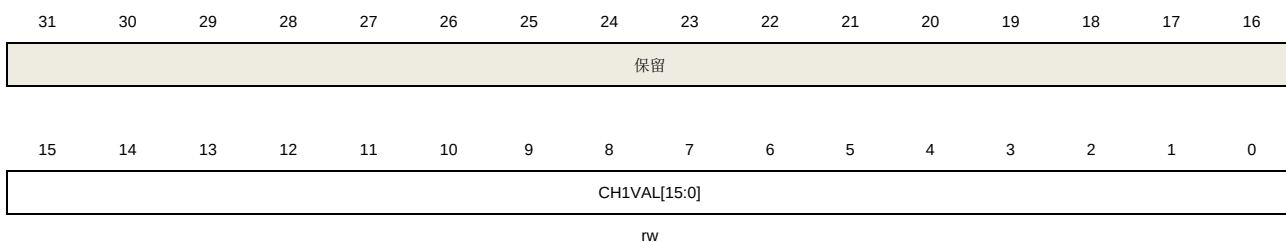
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CHOVAL[15:0]	通道 0 的捕获或比较值 当通道 0 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

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。



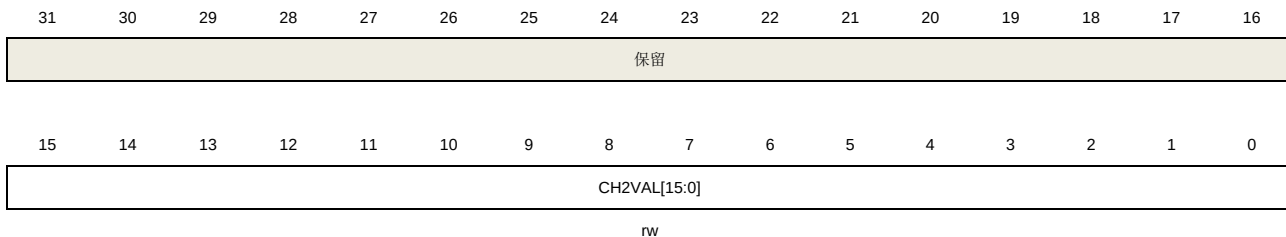
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH1VAL[15:0]	通道 1 的捕获或比较值 当通道 1 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 1 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

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。



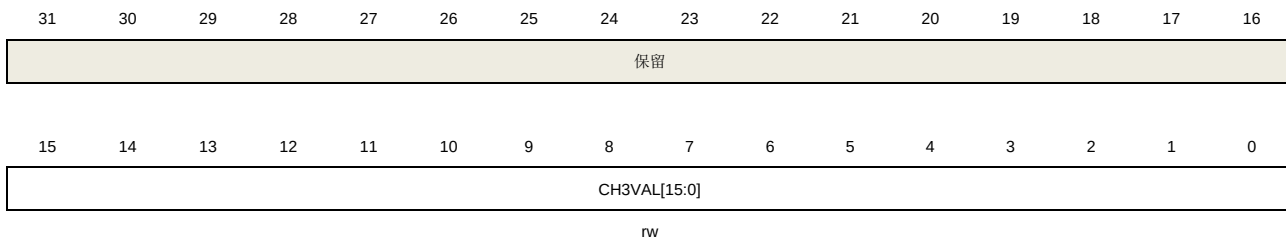
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH2VAL[15:0]	通道 2 的捕获或比较值 当通道 2 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 2 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

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。



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。

互补通道保护寄存器 0 (TIMERx_CCHP0)

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
CHBRKP	CHBRKE N	保留										BRKF[3:0]			
rw	rw											rw			
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							

位/位域	名称	描述
31	CHBRKP	通道中止输入信号极性 该位用于配置通道中止输入信号的极性。 0: 通道中止输入信号低电平有效 1: 通道中止输入信号高电平有效 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] =00时才可修改。
30	CHBRKEN	通道中止输入信号使能 该位置1时，使能通道中止输入信号。 0: 通道中止输入禁能 1: 通道中止输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] =00时才可修改。
29:20	保留	必须保持复位值
19:16	BRKF[3:0]	BREAK和通道中止输入信号滤波 数字滤波器由一个事件计数器组成，它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 BREAK 和通道中止输入信号的采样频率和数字滤波器的长度。 0000: 无滤波器，BREAK 和通道中止异步有效，N=1 0001: $f_{SAMP} = f_{CK\_TIMER}$, N=2 0010: $f_{SAMP} = f_{CK\_TIMER}$, N=4 0011: $f_{SAMP} = f_{CK\_TIMER}$, 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: $f_{SAMP} = f_{DTS}/32, N=8$ 此位只有在 <code>TIMERx_CCHP0</code> 寄存器的 <code>PROT [1:0] =00</code> 时才可修改。
15	POEN	所有的通道输出使能 根据 <code>OAEN</code> 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 (<code>TIMERx_CHCTL2</code> 寄存器的 <code>CHxEN</code> 位, <code>CHxNEN</code> 位), 则使能 <code>CHx_O</code> 和 <code>CHx_ON</code> 得输出。 0: 禁止通道输出或强制为空闲状态 1: 通道输出使能
14	OAEN	自动输出使能 此位定义了 <code>POEN</code>、<code>CHPOENx</code> ($x=0...2$) 位是否可以被硬件自动置 1。 0: <code>POEN</code>、<code>CHPOENx</code> ($x=0...2$) 位不能被硬件置 1 1: 如果中止输入无效, 下一次更新事件发生时, <code>POEN</code>、<code>CHPOENx</code> ($x=0...2$) 位能被硬件自动置 1 此位只有在 <code>TIMERx_CCHP0</code> 寄存器的 <code>PROT [1:0] =00</code> 时才可修改。
13	BRKP	<code>BREAK</code> 输入信号极性 此位定义了 <code>BREAK</code> 输入的极性。 0: <code>BREAK</code> 输入低电平有效 1: <code>BREAK</code> 输入高电平有效 此位只有在 <code>TIMERx_CCHP0</code> 寄存器的 <code>PROT [1:0] =00</code> 时才可修改。
12	BRKEN	<code>BREAK</code> 输入信号使能 此位置 1 使能 <code>BREAK</code> 输入信号。 0: <code>BREAK</code> 输入禁能 1: <code>BREAK</code> 输入使能 此位只有在 <code>TIMERx_CCHP0</code> 寄存器的 <code>PROT [1:0] =00</code> 时才可修改。
11	ROS	运行模式下“关闭状态”使能 当 <code>POEN&CHPOENx(x=0...2)</code> 位被置 1 (运行模式), 此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出“关闭状态”。参见表 12-4. 由参数控制的互补输出表。 0: 输出“关闭状态”禁能。当 <code>CHxEN</code> 或者 <code>CHxNEN</code> 位被清零, 对应通道为输出“禁能状态”。 1: 输出“关闭状态”使能。当 <code>CHxEN</code> 或者 <code>CHxNEN</code> 位被清零, 对应通道为输出“关闭状态”。 此位在 <code>TIMERx_CCHP0</code> 寄存器的 <code>PROT [1:0]=10</code> 或 <code>11</code> 时不能被更改。
10	IOS	空闲模式下“关闭状态”使能 当 <code>POEN&CHPOENx(x=0...2)</code> 位被清 0 (空闲模式), 此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出“关闭状态”。参见表 12-4. 由参数控制的互补输出表。

0: 输出“关闭状态”禁能。当 CHxEN 和 CHxNEN 位均被清零，对应通道为输出“禁能状态”。

1: 输出“关闭状态”使能。不论 CHxEN 和 CHxNEN 位的值，对应通道为输出“关闭状态”。

此位在 TIMERx_CCHP0 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_CCHP0 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_CCHP0 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_CHCTL0/ 1 寄存器中 CHxCOMCTL/ CHxCOMSEN/ CHxCOMADDSSEN 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_CCHP0 寄存器被写入，这两位被写保护。

7:0 DTCFG[7:0]

死区时间控制

这些位定义了插入互补输出之间的死区持续时间。DTCFG 值和死区时间的关系如下：

DTCFG [7:5] =3'b0xx: DTvalue =DTCFG [7:0]x t_{DT}, t_{DT}=t_{DTs}.

DTCFG [7:5] =3'b 10x: DTvalue = (64+DTCFG [5:0]) x t_{DT}, t_{DT} =t_{DTs}*2.

DTCFG [7:5] =3'b 110: DTvalue = (32+DTCFG [4:0]) x t_{DT}, t_{DT}=t_{DTs}*8.

DTCFG [7:5] =3'b 111: DTvalue = (32+DTCFG [4:0]) x t_{DT}, t_{DT} =t_{DTs}*16.

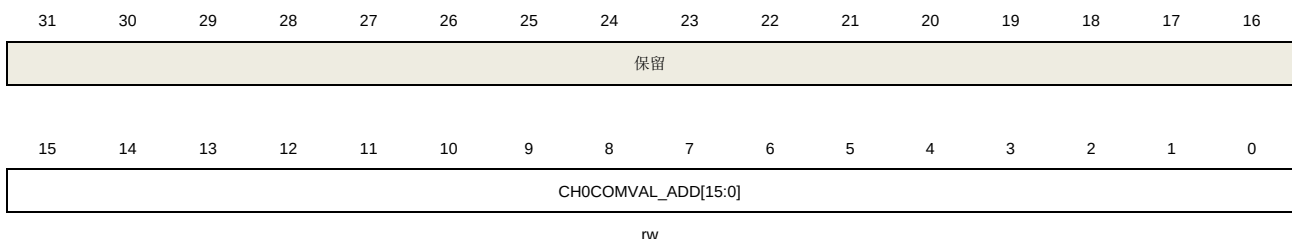
此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0]=00 时才可修改。

通道 0 附加比较寄存器 (TIMERx_CH0COMV_ADD)

地址偏移: 0x64

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



位/位域	名称	描述

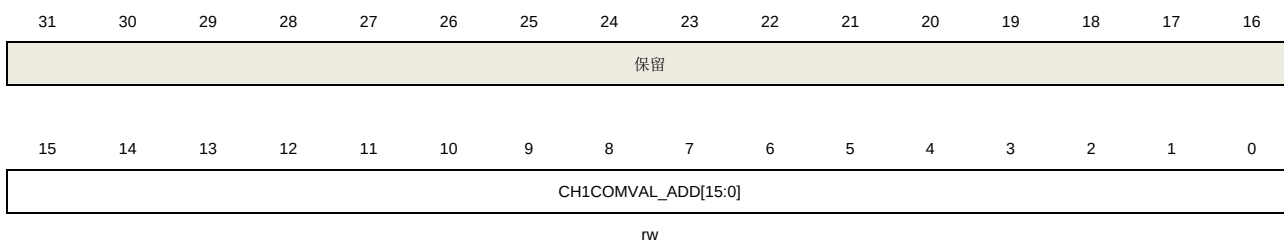
31:16	保留	必须保持复位值
15:0	CH0COMVAL_ADD [15:0]	通道0附加比较值 当通道0配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。 注意： 该寄存器仅用于复合PWM模式（当CH0CPWMEN=1时）。

通道 1 附加比较寄存器（TIMERx_CH1COMV_ADD）

地址偏移：0x68

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



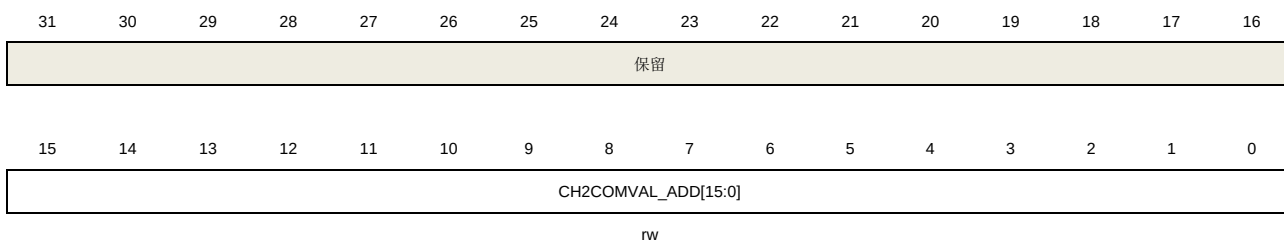
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH1COMVAL_ADD [15:0]	通道1附加比较值 当通道1附加配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。 注意： 该寄存器仅用于复合PWM模式（当CH1CPWMEN=1时）。

通道 2 附加比较寄存器（TIMERx_CH2COMV_ADD）

地址偏移：0x6C

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

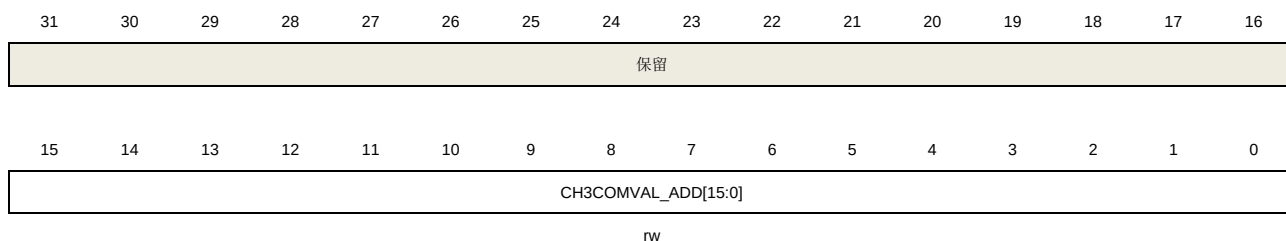
15:0 **CH2COMVAL_ADD** 通道2附加比较值
[15:0] 当通道2附加配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。
注意：该寄存器仅用于复合PWM模式（当CH2CPWMEN=1时）。

通道 3 附加比较寄存器（TIMERx_CH3COMV_ADD）

地址偏移：0x70

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



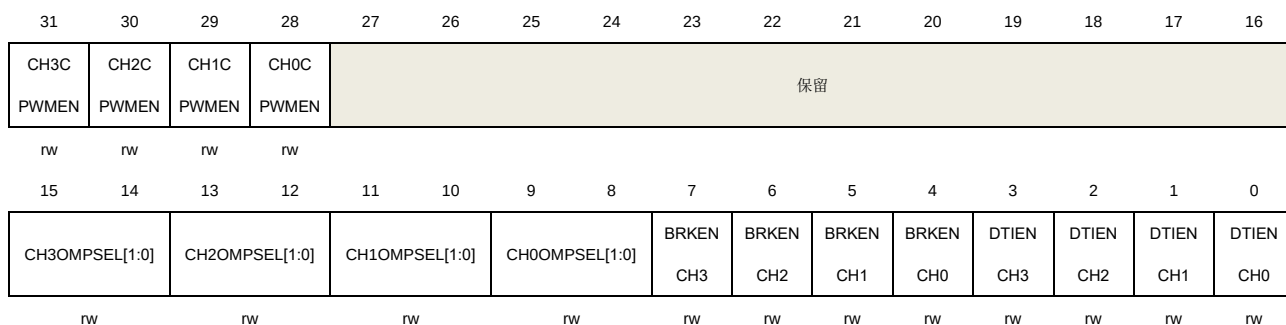
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH3COMVAL_ADD [15:0]	通道3附加比较值 当通道3附加配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。 注意： 该寄存器仅用于复合PWM模式（当CH3CPWMEN=1时）。

控制寄存器 2（TIMERx_CTL2）

地址偏移：0x74

复位值：0x0000 00FF

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31	CH3CPWMEN	通道 3 复合 PWM 模式使能

		0: 通道 3 复合 PWM 模式禁能 1: 通道 3 复合 PWM 模式使能
30	CH2CPWMEN	通道 2 复合 PWM 模式使能 0: 通道 2 复合 PWM 模式禁能 1: 通道 2 复合 PWM 模式使能
29	CH1CPWMEN	通道 1 复合 PWM 模式使能 0: 通道 1 复合 PWM 模式禁能 1: 通道 1 复合 PWM 模式使能
28	CH0CPWMEN	通道 0 复合 PWM 模式使能 0: 通道 0 复合 PWM 模式禁能 1: 通道 0 复合 PWM 模式使能
27:16	保留	必须保持复位值
15:14	CH3OMPSEL[1:0]	通道 3 输出匹配脉冲选择 当匹配事件发生时, 该位用于选择准备输出信号 O3CPRE (用来驱动 CH3_O 信号)。 00: O3CPRE 信号根据 CH3COMCTL[2:0]位的配置输出。 01: 只有在计数器向上计数, 匹配事件发生时, O3CPRE 信号输出一个脉冲, 且脉冲宽度是一个 CK_TIMER 时钟周期。 10: 只有在计数器向下计数, 匹配事件发生时, O3CPRE 信号输出一个脉冲, 且脉冲宽度是一个 CK_TIMER 时钟周期。 11: 在计数器向上计数或向下计数, 匹配事件发生时, O3CPRE 信号输出一个脉冲, 且脉冲宽度是一个 CK_TIMER 时钟周期。
13:12	CH2OMPSEL[1:0]	通道 2 输出匹配脉冲选择 当匹配事件发生时, 该位用于选择准备输出信号 O2CPRE (用来驱动 CH2_O 信号)。 00: O2CPRE 信号根据 CH2COMCTL[2:0]位的配置正常输出。 01: 只有在计数器向上计数, 匹配事件发生时, O2CPRE 信号输出一个脉冲, 且脉冲宽度是一个 CK_TIMER 时钟周期。 10: 只有在计数器向下计数, 匹配事件发生时, O2CPRE 信号输出一个脉冲, 且脉冲宽度是一个 CK_TIMER 时钟周期。 11: 在计数器向上计数或者向下计数, 匹配事件发生时, O2CPRE 信号输出一个脉冲, 且脉冲宽度是一个 CK_TIMER 时钟周期。
11:10	CH1OMPSEL[1:0]	通道 1 输出匹配脉冲选择 当匹配事件发生时, 该位用于选择准备输出信号 O1CPRE (用来驱动 CH1_O 信号)。 00: O1CPRE 信号根据 CH1COMCTL[2:0]位的配置正常输出。 01: 只有在计数器向上计数, 匹配事件发生时, O1CPRE 信号输出一个脉冲, 且脉冲宽度是一个 CK_TIMER 时钟周期。 10: 只有在计数器向下计数, 匹配事件发生时, O1CPRE 信号输出一个脉冲, 且脉冲宽度是一个 CK_TIMER 时钟周期。 11: 在计数器向上计数或者向下计数, 匹配事件发生时, O1CPRE 信号输出一个脉

冲，且脉冲宽度是一个 CK_TIMER 时钟周期。

9:8	CH0OMPSEL[1:0]	通道 0 输出匹配脉冲选择 当匹配事件发生时，该位用于选择准备输出信号 O0CPRE (用来驱动 CH0_O 信号)。 00: O0CPRE 信号根据 CH0COMCTL[2:0]位的配置正常输出。 01: 只有在计数器向上计数，匹配事件发生时，O0CPRE 信号输出一个脉冲，并且脉冲宽度是一个 CK_TIMER 时钟周期。 10: 只有在计数器向下计数，匹配事件发生时，O0CPRE 信号输出一个脉冲，脉冲宽度是一个 CK_TIMER 时钟周期。 11: 在计数器向上计数或者向下计数，匹配事件发生时，O0CPRE 信号输出一个脉冲，脉冲宽度是一个 CK_TIMER 时钟周期。
7	BRKENCH3	通道 3 中止控制使能 0: 通道 3 中止控制禁能 1: 通道 3 中止控制使能
6	BRKENCH2	通道 2 中止控制使能 0: 通道 2 中止控制禁能 1: 通道 2 中止控制使能
5	BRKENCH1	通道 1 中止控制使能 0: 通道 1 中止控制禁能 1: 通道 1 中止控制使能
4	BRKENCH0	通道 0 中止控制使能 0: 通道 0 中止控制禁能 1: 通道 0 中止控制使能
3	DTIENCH3	通道 3 死区时间插入使能 在 CH3_ON 和 CH3_O 输出中使能死区时间插入。 0: 通道 3 死区时间插入禁能 1: 通道 3 死区时间插入使能
2	DTIENCH2	通道 2 死区时间插入使能 在 CH2_ON 和 CH2_O 输出中使能死区时间插入。 0: 通道 2 死区时间插入禁能 1: 通道 2 死区时间插入使能
1	DTIENCH1	通道 1 死区时间插入使能 在 CH1_ON 和 CH1_O 输出中使能死区时间插入。 0: 通道 1 死区时间插入禁能 1: 通道 1 死区时间插入使能

0	DTIENCH0	通道 0 死区时间插入使能 在 CH0_ON 和 CH0_O 输出中使能死区时间插入。 0: 通道 0 死区时间插入禁能 1: 通道 0 死区时间插入使能

备用功能控制寄存器 0 (TIMERx_AFCTL0)

地址偏移: 0x8C

复位值: 0x0000 0001

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						BRKCOMP 0P	保留								BRKIN0P
						rw								rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						BRKCOMP 0EN	保留								BRKIN0E N
						rw								rw	

位/位域	名称	描述
31:26	保留	必须保持复位值。
25	BRKCOMP0P	BREAK CMP0输入极性 0: CMP0输入信号不反相 (BRKP =0, 输入信号低有效; BRKP =1, 输入信号高有效) 1: CMP输入信号反相 (BRKP =0, 输入信号高有效; BRKP =1, 输入信号低有效) 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] =00时才可修改。
24:17	保留	必须保持复位值。
16	BRKIN0P	BREAK BRKIN0备用输入极性 该位用于配置BRKIN0输入极性, 具体极性是由该位和BRKP位共同确定。 0: BRKIN0输入信号不反相 (BRKP =0, 输入信号低有效; BRKP =1, 输入信号高有效) 1: BRKIN0输入信号反相 (BRKP =0, 输入信号高有效; BRKP =1, 输入信号低有效) 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] =00时才可修改。
15:10	保留	必须保持复位值。
9	BRKCOMP0EN	BREAK CMP0输入使能 0: CMP0输入禁能 1: CMP0输入使能

此位只有在TIMERx_CCHP0寄存器的PROT [1:0] =00时才可修改。

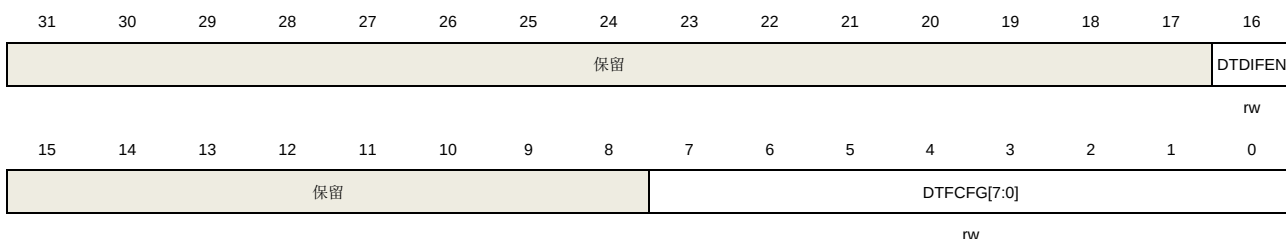
8:1	保留	必须保持复位值。
0	BRKIN0EN	BREAK BRKIN0备用输入使能 0: BRKIN0输入禁能 1: BRKIN0输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] =00时才可修改。

互补通道保护寄存器 1 (TIMERx_CCHP1)

地址偏移: 0x9C

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	DTDIFEN	死区时间不同配置使能 0: 上升沿和下降沿的死区时间 (在 TIMERx_CCHP0 寄存器的 DTFCFG[7:0]位域配置) 相同。 1: 上升沿的死区时间由 TIMERx_CCHP0 寄存器的 DTFCFG[7:0]位域配置; 下降沿的死区时间由 TIMERx_CCHP1 寄存器的 DTFCFG[7:0]位域配置。 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] = 00时才可修改。
15:8	Reserved	Must be kept at reset value.
7:0	DTFCFG[7:0]	死区时间控制 这些位定义了插入互补输出之间的死区持续时间。DTFCFG 值和死区时间的关系如下: $DTFCFG[7:5] = 3'b0xx: DTvalue = DTFCFG[7:0] \times t_{DT}, t_{DT} = t_{DTS}$ $DTFCFG[7:5] = 3'b10x: DTvalue = (64 + DTFCFG[5:0]) \times t_{DT}, t_{DT} = t_{DTS} \times 2$ $DTFCFG[7:5] = 3'b110: DTvalue = (32 + DTFCFG[4:0]) \times t_{DT}, t_{DT} = t_{DTS} \times 8$ $DTFCFG[7:5] = 3'b111: DTvalue = (32 + DTFCFG[4:0]) \times t_{DT}, t_{DT} = t_{DTS} \times 16$ 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0] = 00 时才可修改。

计数器初值控制寄存器(TIMERx_CINITCTL)

地址偏移: 0xA4

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													SWSYNC	CINITDIR	CINITVE
													G	N	N
													rw	rw	rw

位/位域	名称	描述
31:3	保留	必须保持复位值。
2	SWSYNCG	软件同步事件产生 该位由软件置位，硬件自动清零，对该位进行读操作，读取值均为0。 0: 无效 1: 产生软件更新事件
1	CINITDIR	计数器初始计数方向 0: 同步事件发生时，计数器向上计数 1: 同步事件发生时，计数器向下计数 当同步事件发生，且计数器从TIMERx_CINITV寄存器加载计数器初始值后，该位表明了此时计数器的计数初始方向。 注意: 该位仅用于CAM[1:0] ≠ 00的情况。
0	CINITVEN	计数器初值寄存器使能 0: 计数器初值寄存器禁能。计数器寄存器不从计数器初值寄存器加载初值。 1: 计数器初值寄存器使能。当同步事件发生（通过软件将SWSYNCG位置1产生软件同步事件，输入引脚输入硬件同步事件）时，计数器寄存器可以从计数器初值寄存器加载初值。

计数器初值寄存器(TIMERx_CINITV)

地址偏移: 0xA8

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															

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

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CINITVAL[15:0]	计数器初始值 该位域用于表示计数器的初始值。 CINITVEN位为0时，计数器的寄存器不能加载在CINITVAL[15:0]位域中设置的初始值。 CINITVEN位为1时，当同步事件发生时，计数器的寄存器可以加载在CINITVAL[15:0]位域中设置的初始值。

通道中止控制寄存器（TIMERx_CHBRKCTL）

地址偏移：0xAC

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									CHPOEN	CHPOEN	CHPOEN	保留			
									2	1	0				
									rw	rw	rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	CH2BRKI	CH1BRKI	CH0BRKI	保留	CH2BRKI	CH1BRKI	CH0BRKI	保留	CH2BRKI	CH1BRKI	CH0BRKI	保留	CH2BRK	CH1BRK	CH0BRK
	NEN	NEN	NEN		E	E	E		NP	NP	NP		MIE	MIE	MIE
	rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw

位/位域	名称	描述
31:23	保留	必须保持复位值。
22	CHPOEN2	通道输出使能 2 根据 OAEN 位，该位可以软件设置或者硬件自动设置。一旦通道 2 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（CH2EN 位，CH2NEN 位），则使能 CH2_O 和 CH2_ON 输出。 0：禁止 CH2_O/CH2_ON 输出或强制为空闲状态 1：CH2_O/CH2_ON 输出使能
21	CHPOEN1	通道输出使能 1 根据 OAEN 位，该位可以软件设置或者硬件自动设置。一旦通道 1 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（CH1EN 位，CH1NEN 位），则使能 CH1_O 和 CH1_ON 输出。

		0: 禁止 CH1_O/CH1_ON 输出或强制为空闲状态 1: CH1_O/CH1_ON 输出使能
20	CHPOEN0	通道输出使能 0 根据 OAEN 位，该位可以软件设置或者硬件自动设置。一旦通道 0 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（CH0EN 位，CH0NEN 位），则使能 CH0_O 和 CH0_ON 输出。 0: 禁止 CH0_O/CH0_ON 输出或强制为空闲状态 1: CH0_O/CH0_ON 输出使能
19:15	保留	必须保持复位值。
14	CH2BRKINEN	通道2中止输入信号使能 该位置1时，使能通道2中止输入信号。 0: 通道2中止输入禁能 1: 通道 2 中止输入使能
13	CH1BRKINEN	通道1中止输入信号使能 该位置1时，使能通道1中止输入信号。 0: 通道1中止输入禁能 1: 通道 1 中止输入使能
12	CH0BRKINEN	通道0中止输入信号使能 该位置1时，使能通道0中止输入信号。 0: 通道0中止输入禁能 1: 通道0中止输入使能
11	保留	必须保持复位值。
10	CH2BRKIE	通道2中止中断使能 0: 禁能 1: 使能
9	CH1BRKIE	通道1中止中断使能 0: 禁能 1: 使能
8	CH0BRKIE	通道0中止中断使能 0: 禁能 1: 使能
7	保留	必须保持复位值。
6	CH2BRKINP	通道2中止输入信号极性 该位用于配置通道2中止输入信号的极性 0: 通道2中止输入信号不反相

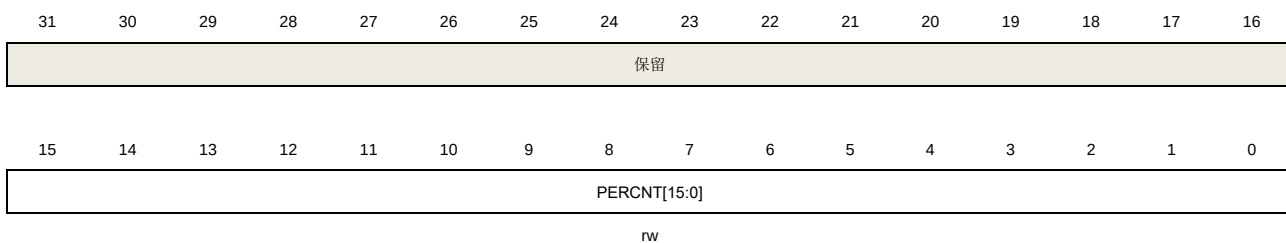
		1: 通道 2 中止输入信号反相
5	CH1BRKINP	通道1中止输入信号极性 该位用于配置通道1中止输入信号的极性 0: 通道1中止输入信号不反相 1: 通道 1 中止输入信号反相
4	CH0BRKINP	通道0中止输入信号极性 该位用于配置通道0中止输入信号的极性 0: 通道0中止输入信号不反相 1: 通道 0 中止输入信号反相
3	保留	必须保持复位值。
2	CH2BRKMIE	通道2中止多周期中断使能 0: 禁能 1: 使能
1	CH1BRKMIE	通道1中止多周期中断使能 0: 禁能 1: 使能
0	CH0BRKMIE	通道0中止多周期中断使能 0: 禁能 1: 使能

通道中止周期寄存器（TIMERx_CHBRKPER）

地址偏移：0xB0

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	PERCNT[15:0]	通道中止周期数 在边沿对齐模式下，该位域代表 PERCNT 个 PWM 周期。在中央对齐模式下，该位域代表 PERCNT/2 个 PWM 周期。

通道中止中断标志寄存器 (TIMERx_CHBRKINTF)

地址偏移: 0xB4

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									CH2BRKI	CH1BRKI	CH0BRKI	保留	CH2BRK	CH1BRK	CH0BRK
									F	F	F		MIF	MIF	MIF
									rc_w0	rc_w0	rc_w0		rc_w0	rc_w0	rc_w0

位/位域	名称	描述
31:7	保留	必须保持复位值
6	CH2BRKIF	通道 2 中止中断标志位 一旦通道 2 中止输入有效, 由硬件对该位置‘1’。如果通道 2 中止输入无效, 则该位可由软件清‘0’。 0: 通道 2 中止输入未检测到有效电平 1: 通道 2 中止输入检测到有效电平 注意: 当 CH2BRKIE =1 和 BRKIE =1 时, 如果该位被置 1, 则会产生中断。
5	CH1BRKIF	通道 1 中止中断标志位 一旦通道 1 中止输入有效, 由硬件对该位置‘1’。如果通道 1 中止输入无效, 则该位可由软件清‘0’。 0: 通道 1 中止输入未检测到有效电平 1: 通道 1 中止输入检测到有效电平 注意: 当 CH1BRKIE =1 和 BRKIE =1 时, 如果该位被置 1, 则会产生中断。
4	CH0BRKIF	通道 0 中止中断标志位 一旦通道 0 中止输入有效, 由硬件对该位置‘1’。如果通道 0 中止输入无效, 则该位可由软件清‘0’。 0: 通道 0 中止输入未检测到有效电平 1: 通道 0 中止输入检测到有效电平 注意: 当 CH0BRKIE =1 和 BRKIE =1 时, 如果该位被置 1, 则会产生中断。
3	保留	必须保持复位值
2	CH2BRKMIF	通道 2 中止多周期中断标志位 当通道2中止输入在多个连续周期内 (由TIMERx_CHBRKPER寄存器决定) 处于有效状态, 硬件会该位置‘1’, 如果通道2中止输入不在有效状态, 则软件可清除此标志。 0: 通道2中止输入未检测到多个连续周期有效电平

1: 通道2中止输入检测到多个连续周期有效电平
 注意: 当 CH2BRKMIE =1 和 BRKIE =1 时, 如果该位被置 1, 则会产生中断。

- 1

CH1BRKMIF

通道 1 中止多周期中断标志位

当通道1中止输入在多个连续周期内（由TIMERx_CHBRKPER寄存器决定）处于有效状态, 硬件会该位置‘1’, 如果通道1中止输入不在有效状态, 则软件可清除此标志。

0: 通道1中止输入未检测到多个连续周期有效电平

1: 通道1中止输入检测到多个连续周期有效电平

注意: 当 CH1BRKMIE =1 和 BRKIE =1 时, 如果该位被置 1, 则会产生中断。
- 0

CH0BRKMIF

通道 0 中止多周期中断标志位

当通道0中止输入在多个连续周期内（由TIMERx_CHBRKPER寄存器决定）处于有效状态, 硬件会该位置‘1’, 如果通道0中止输入不在有效状态, 则软件可清除此标志。

0: 通道0中止输入未检测到多个连续周期有效电平

1: 通道0中止输入检测到多个连续周期有效电平

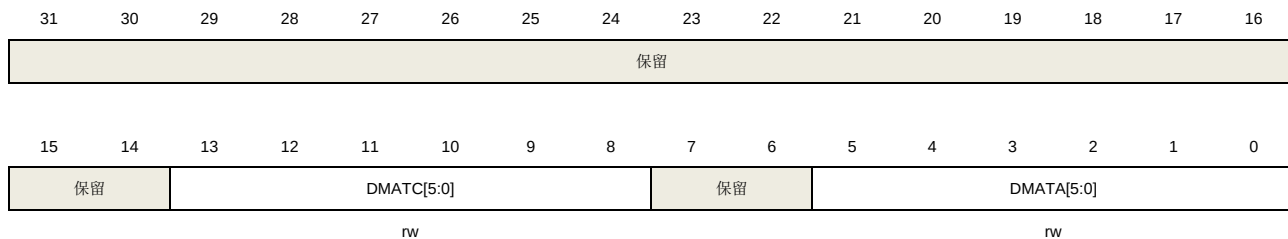
注意: 当 CH0BRKMIE =1 和 BRKIE =1 时, 如果该位被置 1, 则会产生中断。

DMA 配置寄存器 (TIMERx_DMACFG)

地址偏移: 0xE0

复位值: 0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:14	保留	必须保持复位值
13:8	DMATC[5:0]	DMA 传输计数 该位域定义了 DMA 访问（读/写）TIMERx_DMATB 寄存器的次数。 6'b000000: 传输1次 6'b000001: 传输2次 ... 6'b111111: 传输 63 次
7:6	保留	必须保持复位值
5:0	DMATA[5:0]	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。当通过

TIMERx_DMA 第一次访问时，访问的就是该位域指定的地址。第二次访问
TIMERx_DMATB 时，将访问起始地址+0x4。

6'b0_0000: TIMERx_CTL0

6'b0_0001: TIMERx_CTL1

...

6'b111111: TIMERx_DMACFG

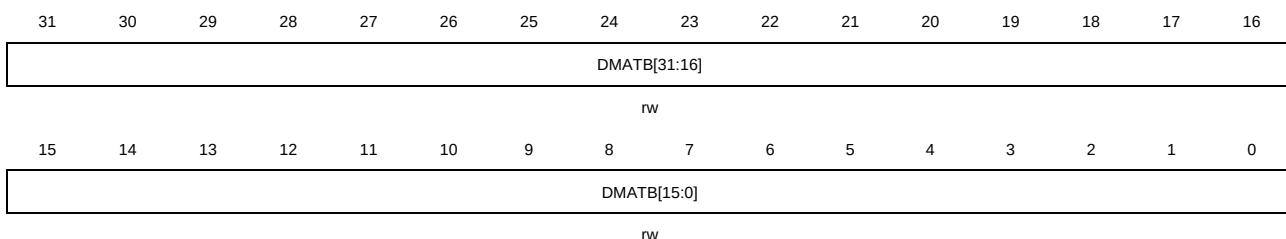
总之： 起始地址 = TIMERx_CTL0 + DMATA*4

DMA 发送缓冲区寄存器 (TIMERx_DMATB)

地址偏移：0xE4

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



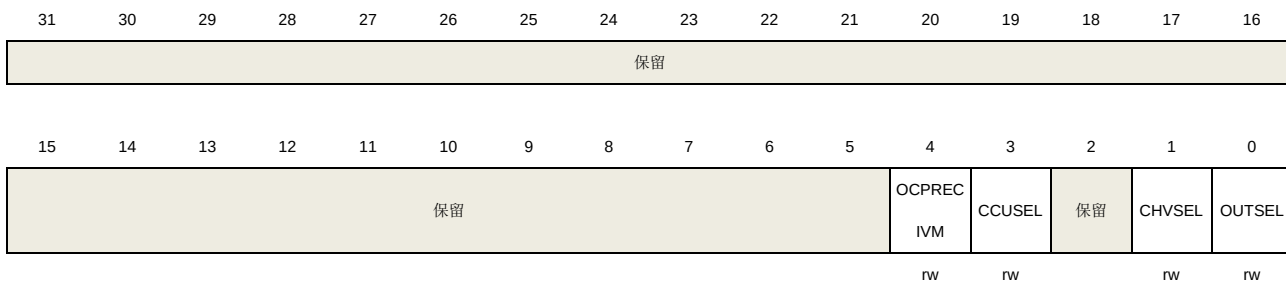
位/位域	名称	描述
31:0	DMATB [31:0]	DMA 发送缓冲 对这个寄存器的读或写，（起始地址+传输次数*4）地址范围内的寄存器会被访问 传输次数由硬件计算，范围为 0 到 DMATC。

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。



位/位域	名称	描述
31:5	保留	必须保持复位值

4	OCPRECIVM	输出准备信号清除无效模式选择 这个位决定了硬件何时将输出准备信号清除设置为无效。 0: 下个计数器上溢或者下溢事件 1: 下个计数器更新事件
3	CCUSEL	换相控制影子寄存器更新选择 只有当CCUC[2:0]位域配置为100,101和110时, 该位才有效。 0: 当计数器产生一个上溢/下溢事件时, 影子寄存器才更新 1: 当重复计数器值为0, 且计数器产生一个上溢/下溢事件时, 影子寄存器才更新
2	保留	必须保持复位值
1	CHVSEL	写捕获比较寄存器选择位 此位由软件写 1 或清 0。 1: 当写入捕获比较寄存器的值与寄存器当前值相等时, 写入操作无效 0: 无影响
0	OUTSEL	输出值选择位 此位由软件写 1 或清 0。 1: 如果 POEN 位与 IOS 位均为 0, 则输出无效 0: 无影响

12.2. 通用定时器 L0 (TIMERx, x=1,2,3,4)

12.2.1. 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 的计数器是 16 位或 32 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。

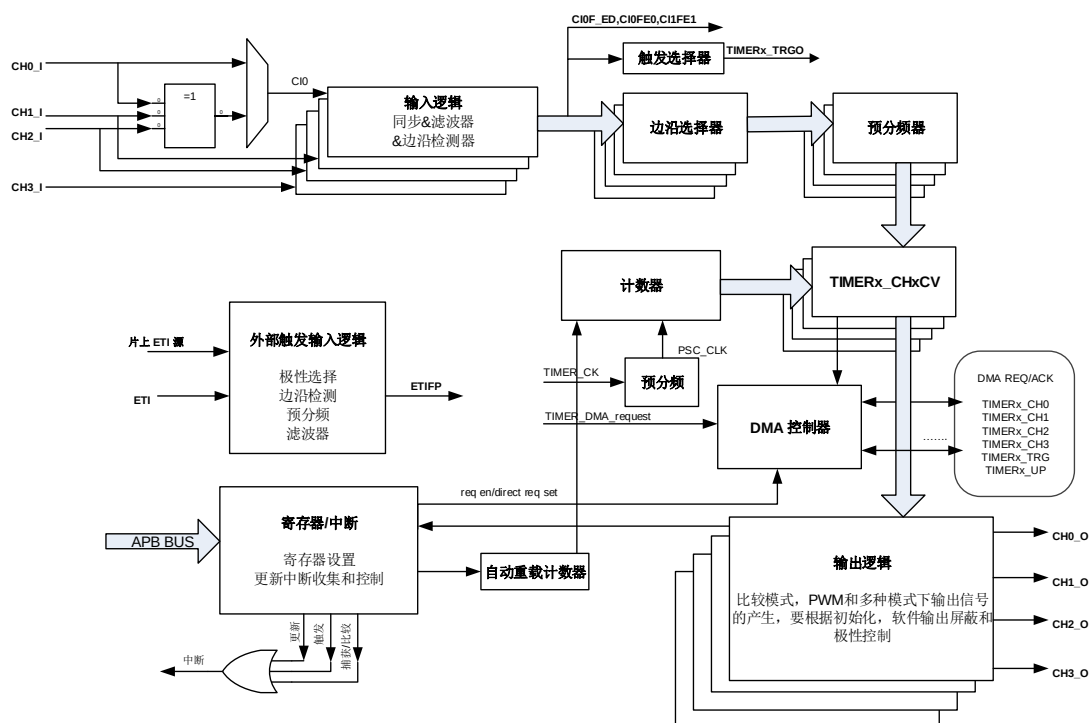
12.2.2. 主要特性

- 总通道数：4；
- 计数器宽度：16位 (TIMER2/3/4) 和32位 (TIMER1)；
- 时钟源可选：内部时钟，内部触发，外部输入，外部触发；
- 多种计数模式：向上计数，向下计数和中央计数；
- 正交译码器接口：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-从管理。

12.2.3. 结构框图

[图 12-48. 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 12-48. 通用定时器 L0 结构框图



12.2.4. 功能描述

时钟源选择

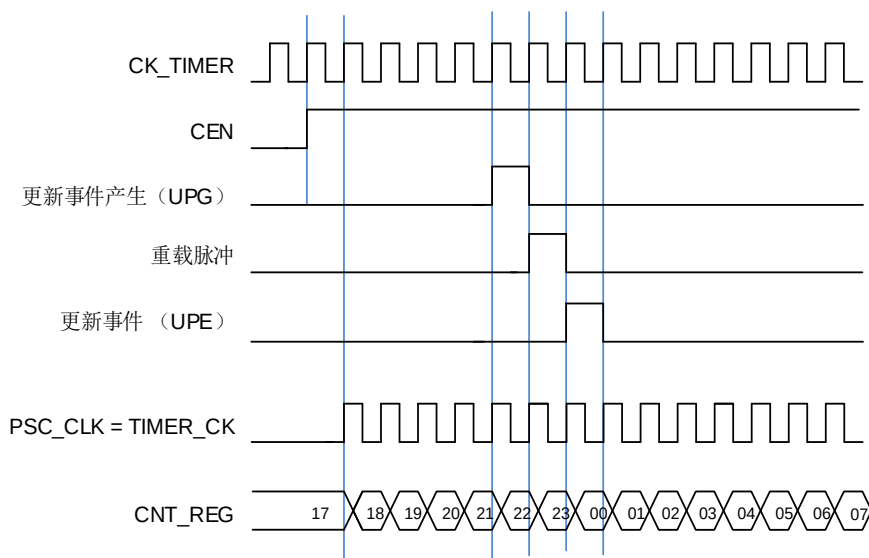
通用定时器 L0 可以由内部时钟源 CK_TIMER 或者由 SYSCFG_TIMERxCFG(x=1..4)寄存器中的 TSCFGy[4:0] (y=0..6,9..15)位域控制的复用时钟源驱动。

- 当SYSCFG_TIMERxCFG(x=1..4)寄存器中的TSCFGy[4:0]=5'b00000(y= y=0..6,9..15)时，定时器选择内部时钟源（连接到RCU模块的CK_TIMER）

如果 SYSCFG_TIMERxCFG(x=1..4)寄存器中的 TSCFGy[4:0] =5'b00000 (y=0..6,9..15)，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 SYSCFG_TIMERxCFG(x=1..4)寄存器中的 TSCFGy[4:0] (y=0..2,6,9..14)位域设置为非零值，预分频器被其他时钟源驱动，具体在下文说明。当 TSCFGy[4:0] (y=3,4,5)被设置为非零值时，计数器预分频器时钟源由内部时钟 TIMER_CK 驱动。

图 12-49. 内部时钟分频为 1 时正常模式下的控制电路



- **TSCFG6[4:0] != 5'b00000** (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_CIO/ TIMERx_CII** 引脚的每个上升沿或下降沿计数。这种模式可以通过设置 **TSCFG6[4:0]** 为 **0x5**, **0x6** 或 **0x7** 来选择。**CIX** 是 **TIMERx_CIX** 通过数字滤波器采样后的信号。

计数器预分频器也可以在内部触发信号 **ITI0~ITI3** 的上升沿计数。这种模式可以通过设置 **TSCFG6[4:0]** 为 **0x1~0x4** 来选择。

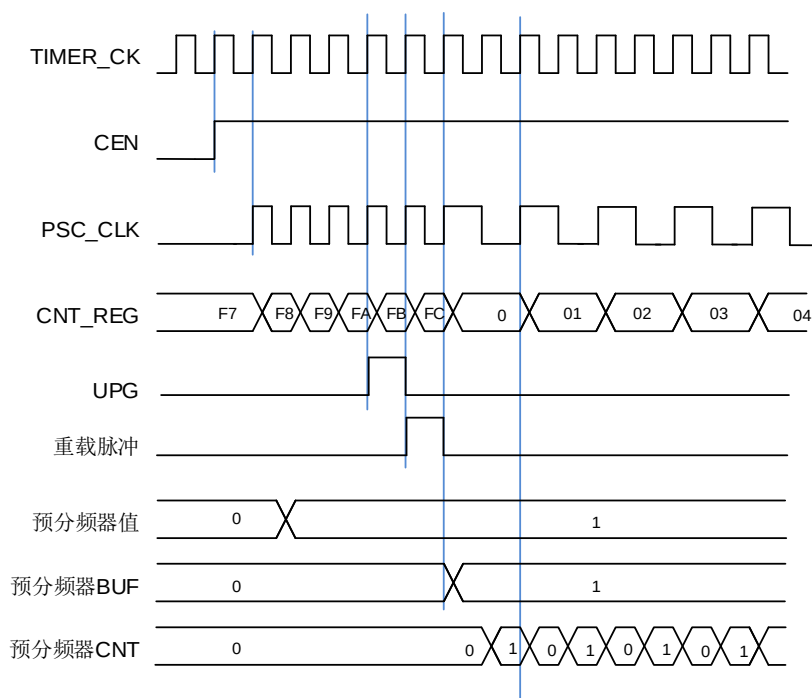
- **SMC1= 1'b1** (外部时钟模式1)，定时器选择外部输入引脚**ETI**作为时钟源

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 **TSCFG6[4:0]** 为 **0x8**。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。

时钟预分频器

预分频器可以将定时器的时钟 (**TIMER_CLK**) 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 12-50. 当预分频器的参数从 1 变到 2 时，计数器的时序图



向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从 0 开始向上连续计数到自动加载值（定义在 `TIMERx_CAR` 寄存器中），一旦计数器计数到自动加载值，会重新从 0 开始向上计数并产生上溢事件。在向上计数模式中，`TIMERx_CTL0` 寄存器中的计数方向控制位 `DIR` 应该被设置成 0。

当通过 `TIMERx_SWEVG` 寄存器的 `UPG` 位置 1 来设置更新事件时，计数值会被清 0，并产生更新事件。

如果 `TIMERx_CTL0` 寄存器的 `UPDIS` 置 1，则禁止更新事件。

当发生更新事件时，所有的寄存器（自动重载寄存器，预分频寄存器）都将被更新。

[图 12-51. 向上计数时序图, PSC=0/2](#) 和 [图 12-52. 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=0x99` 时，计数器在不同预分频因子下的行为。

图 12-51. 向上计数时序图, PSC=0/2

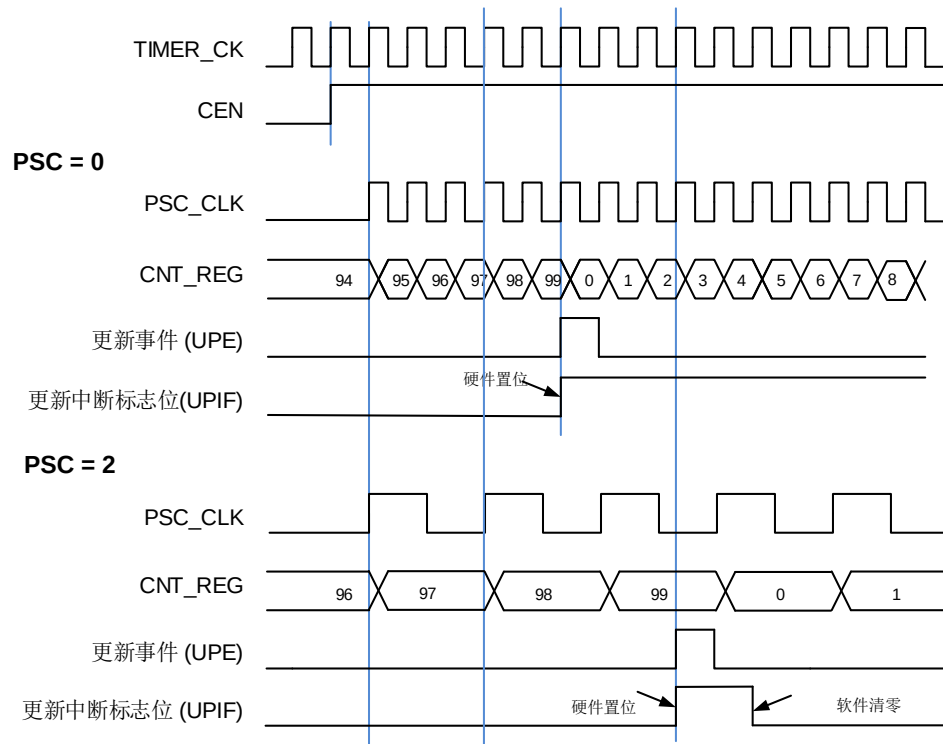
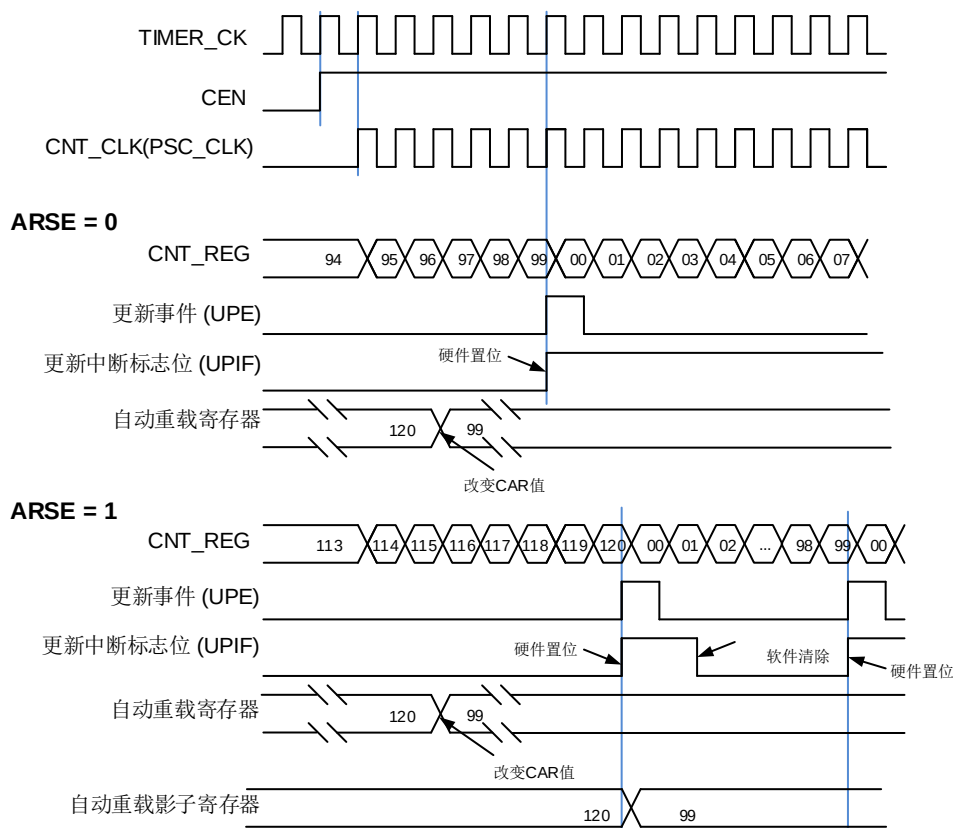


图 12-52. 向上计数时序图，在运行时改变 **TIMERx_CAR** 寄存器的值

向下计数模式

在这种模式，计数器的计数方向是向下计数。计数器从自动加载值（定义在 **TIMERx_CAR** 寄存器中）向下连续计数到 0。一旦计数器计数到 0，计数器会重新从自动加载值开始计数并产生下溢事件。在向下计数模式中，**TIMERx_CTL0** 寄存器中的计数方向控制位 **DIR** 应该被设置成 1。

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

如果 **TIMERx_CTL0** 寄存器的 **UPDIS** 置 1，则禁止更新事件。

当发生更新事件时，所有的寄存器（自动重载寄存器，预分频寄存器）都将被更新。

[图 12-53. 向下计数时序图, PSC=0/2](#) 和 [图 12-54. 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=0x99** 时，计数器在不同时钟频率下的行为。

图 12-53. 向下计数时序图, PSC=0/2

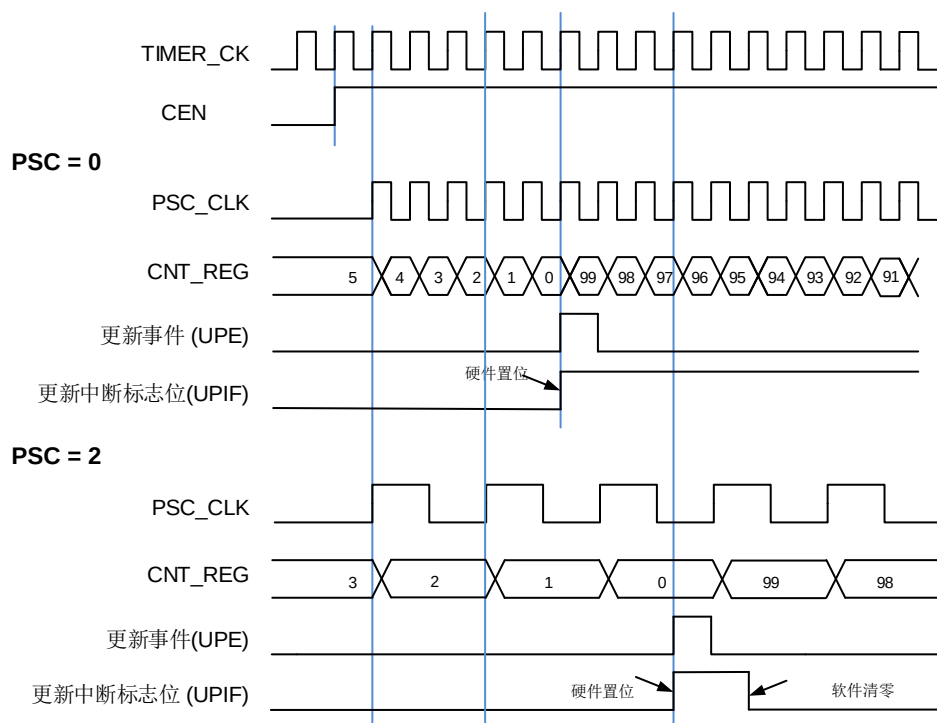
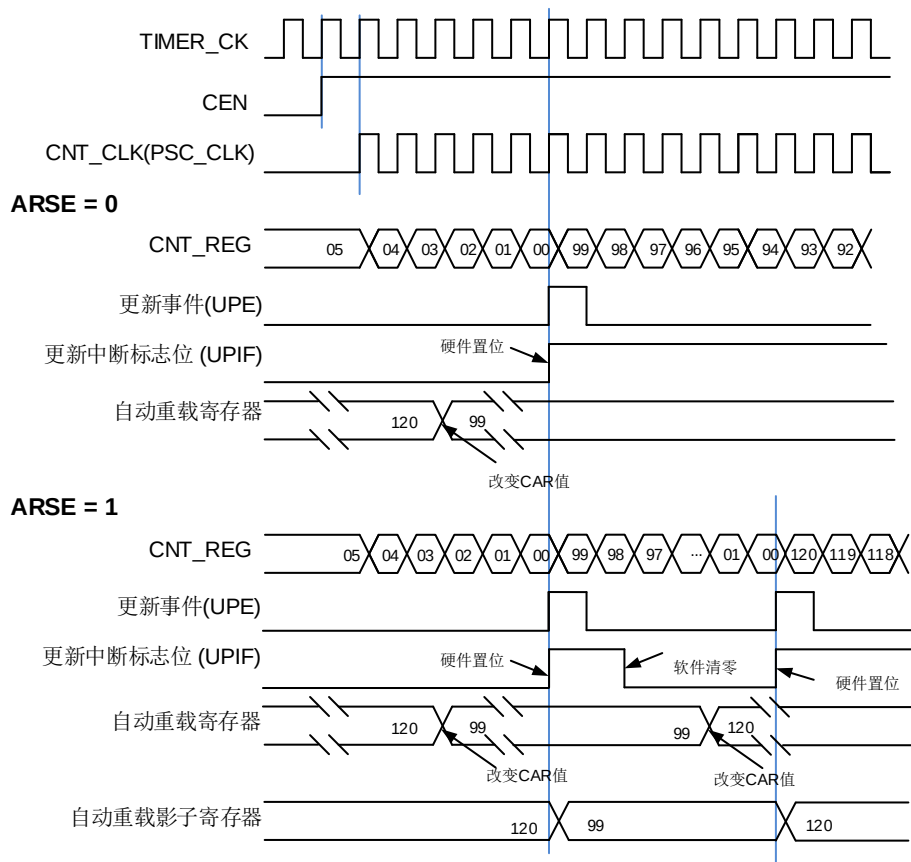


图 12-54. 向下计数时序图，在运行时改变 **TIMERx_CAR** 寄存器值

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_CAR-1**) 产生一个上溢事件；向下计数模式中，定时器模块在计数器计数到 1 时产生一个下溢事件。在中央计数模式中，**TIMERx_CTL0** 寄存器中的计数方向控制位 **DIR** 只读，表明了计数方向。计数方向被硬件自动更新。

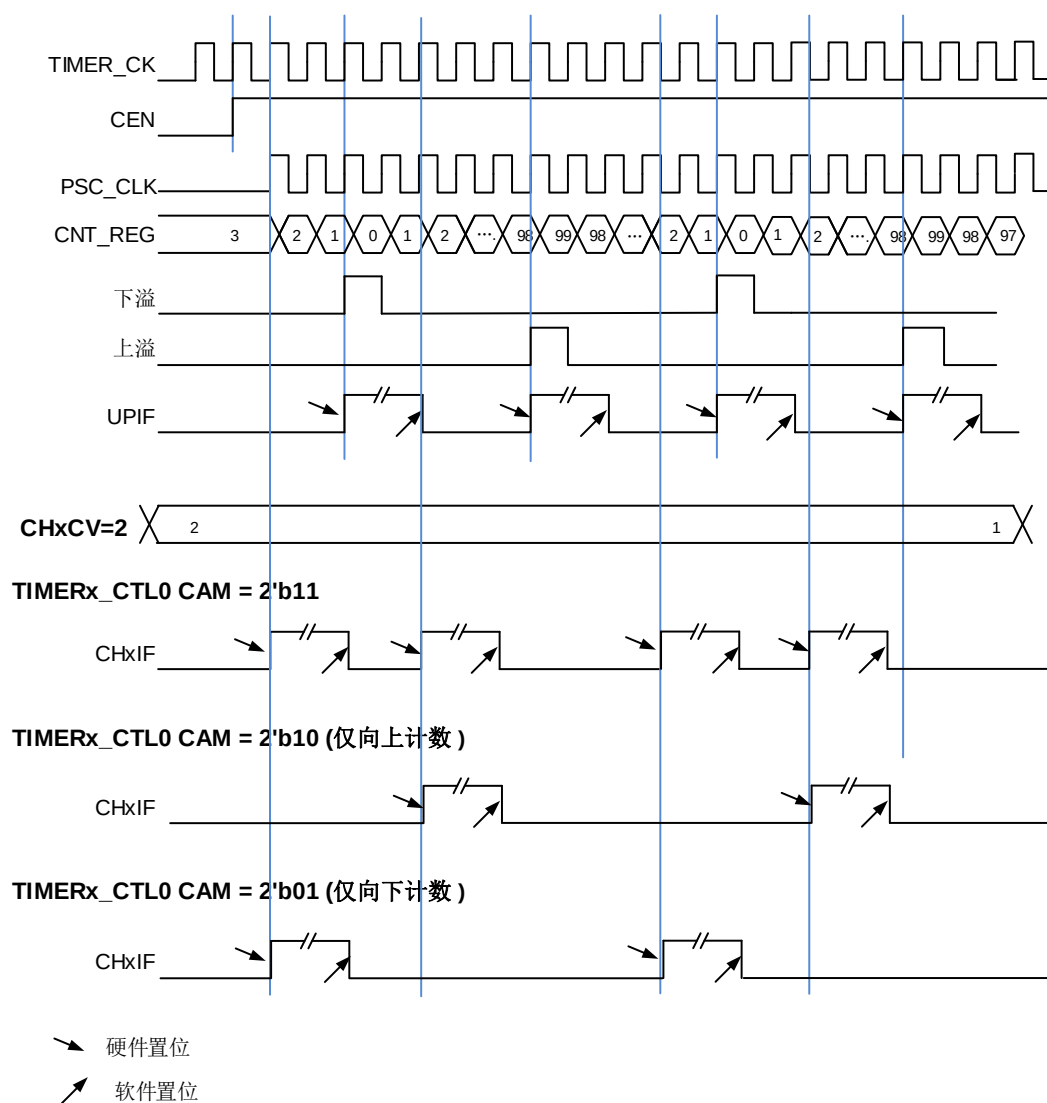
将 **TIMERx_SWEVG** 寄存器的 **UPG** 位置 1 可以初始化计数值为 0，并产生一个更新事件，而无需考虑计数器在中央模式下是向上计数还是向下计数。

上溢或者下溢时，**TIMERx_INTF** 寄存器中的 **UPIF** 位都会被置 1，然而 **CHxIF** 位置 1 与 **TIMERx_CTL0** 寄存器中 **CAM** 的值有关。具体细节参考 [图 12-55. 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 12-55. 中央计数模式计数器时序图](#) 给出了一些例子，当 **TIMERx_CAR=0x99**，**TIMERx_PSC=0x0** 时，计数器的行为

图 12-55. 中央计数模式计数器时序图



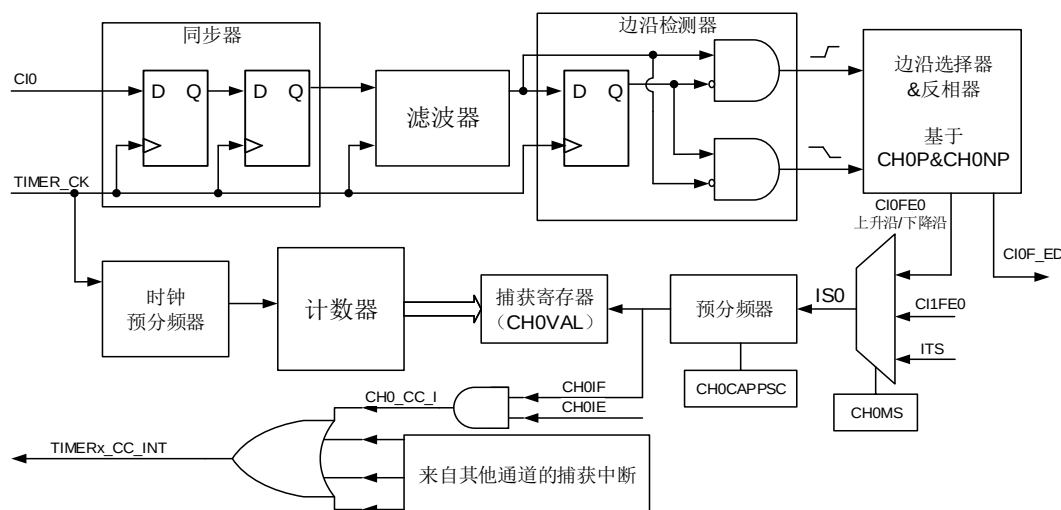
捕获/比较通道

通用定时器 L0 拥有四个独立的通道用于捕获输入或比较输出是否匹配。每个通道都围绕一个通道捕获比较寄存器建立，包括一个输入级，通道控制器和输出级。

■ 输入捕获模式

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

图 12-56. 输入捕获逻辑



通道输入信号 Clx 有两种选择，一种是 $TIMERx\_CHx$ 信号，另一种是 $TIMERx\_CH0, TIMERx\_CH1$ 和 $TIMERx\_CH2$ 异或之后的信号（仅限于 $Cl0$ ）。通道输入信号 Clx 先被 $TIMER\_CK$ 信号同步，然后经过数字滤波器采样，产生一个被滤波后的信号。通过边沿检测器，可以选择检测上升沿或者下降沿。通过配置 $CHxP$ 选择使用上升沿或者下降沿。配置 $CHxMS$ ，可以选择其他通道的输入信号，内部触发信号。配置 IC 预分频器，使得若干个输入事件后才产生一个有效的捕获事件。捕获事件发生， $CHxVAL$ 存储计数器的值。

配置步骤如下：

第一步：滤波器配置（ $TIMERx\_CHCTL0$ 寄存器中 $CHxCAPFLT$ ）：

根据输入信号和请求信号的质量，配置相应的 $CHxCAPFLT$ 。

第二步：边沿选择（ $TIMERx\_CHCTL2$ 寄存器中 $CHxP$ ）：

配置 $CHxP$ 选择上升沿或者下降沿。

第三步：捕获源选择（ $TIMERx\_CHCTL0$ 寄存器中 $CHxMS$ ）：

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

第四步：中断使能（ $TIMERx\_DMAINTEN$ 寄存器中 $CHxIE$ 和 $CHxDEN$ ）：

使能相应中断，可以获得中断和 DMA 请求。

第五步：捕获使能（ $TIMERx\_CHCTL2$ 寄存器中 $CHxEN$ ）。

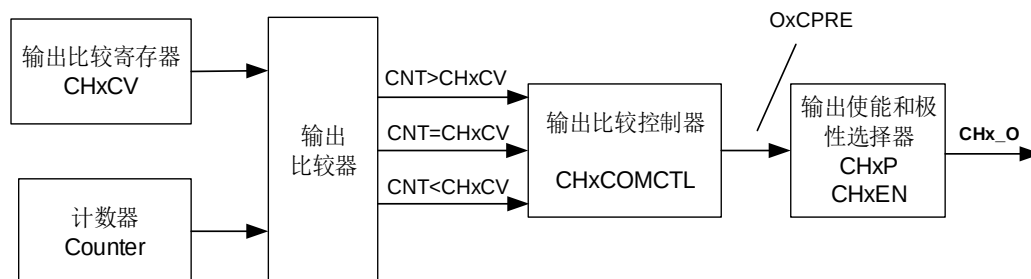
结果：当期望的输入信号发生时， $TIMERx\_CHxCV$ 被设置成当前计数器的值， $CHxIF$ 为置 1。如果 $CHxIF$ 位已经为 1，则 $CHxOF$ 位置 1。根据 $TIMERx\_DMAINTEN$ 寄存器中 $CHxIE$ 和 $CHxDEN$ 的配置，相应的中断和 DMA 请求会被提出。

直接产生：软件设置 $CHxG$ 位，会直接产生中断和 DMA 请求。

输入捕获模式也可用来测量 $TIMERx\_CHx$ 引脚上信号的脉冲波宽度。例如，一个 PWM 波连接到 $CI0$ 。配置 $TIMERx\_CHCTL0$ 寄存器中 $CH0MS$ 为 $2'b01$ ，选择通道 0 的捕获信号为 $CI0$ 并设置上升沿捕获。配置 $TIMERx\_CHCTL0$ 寄存器中 $CH1MS$ 为 $2'b10$ ，选择通道 1 捕获信号为 $CI0$ 并设置下降沿捕获。计数器配置为复位模式，在通道 0 的上升沿复位。 $TIMERx\_CH0CV$ 寄存器测量 PWM 的周期值， $TIMERx\_CH1CV$ 寄存器测量 PWM 占空比值。

■ 输出比较模式

图 12-57. 输出比较逻辑 ($x=0,1,2,3$)



[图 12-57. 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[通道输出参考信号](#)）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 $CHxEN$ 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）、 $CHxEN=1$ （ 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，被置低电平或者翻转。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 $CxCDE=1$ 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/翻转）；

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/CxCDE 位配置中断/DMA 请求使能。

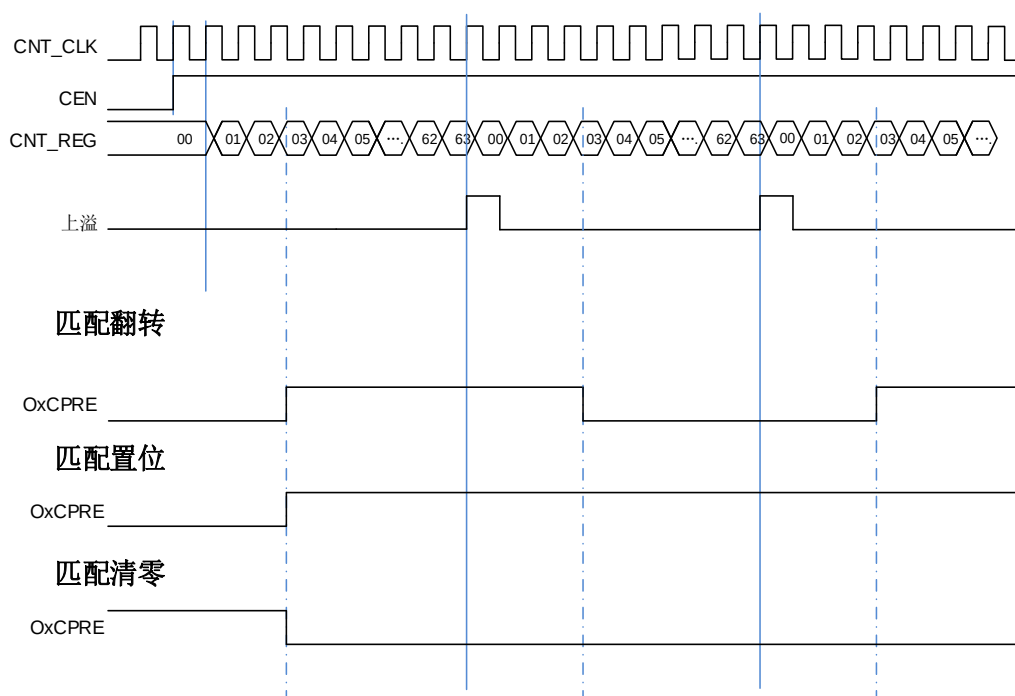
第四步：通过 TIMERx_CAR 寄存器和 TIMERx_CHxCV 寄存器配置输出比较时基：

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 12-58. 三种输出比较模式](#)显示了三种比较输出模式：翻转/置高电平/置低电平，CAR=0x63, CHxVAL=0x3。

图 12-58. 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 4'b0110，PWM 模式 1 是配置 CHxCOMCTL 为 4'b0111），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 12-59. 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 12-60. CAPWM 时序图](#)显示了 CAPWM 的输出波形和中断。

当计数器向上计数时，在 PWM0 模式下（CHxCOMCTL=4'b0110），如果 TIMEx_CHxCV 寄存器的值大于 TIMEx_CAR 寄存器的值，通道输出一直为有效电平；PWM1 模式下（CHxCOMCTL=4'b0111），如果 TIMEx_CHxCV 寄存器的值大于 TIMEx_CAR 寄存器的值，通道输出一直为无效电平。

图 12-59. EAPWM 时序图

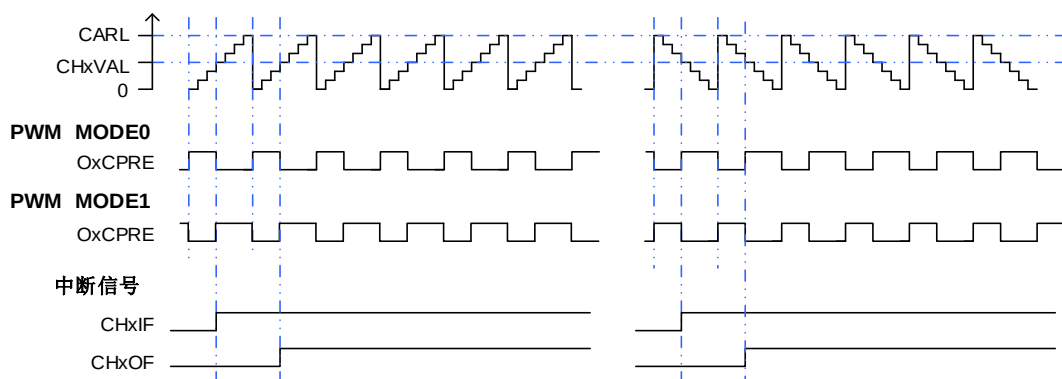
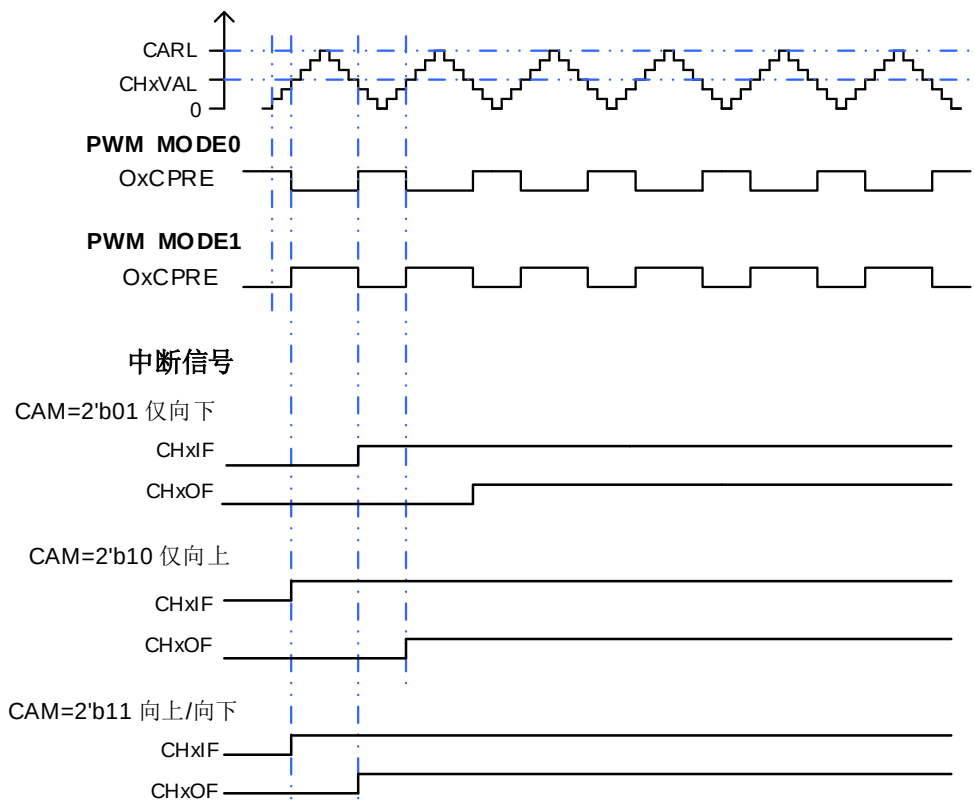


图 12-60. CAPWM 时序图



通道输出参考信号

根据 [图 12-57. 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, 设置 `CHxCOMCTL` 位可以定义 `OxCPRE` 信号 (通道 `x` 准备信号) 类型。`OxCPRE` 信号有若干类型的输出功能, 包括, 设置 `CHxCOMCTL=0x00` 可以保持原始电平; 设置 `CHxCOMCTL=0x01` 可以将 `OxCPRE` 信号设置为高电平; 设置 `CHxCOMCTL=0x02` 可以将 `OxCPRE` 信号设置为低电平; 设置 `CHxCOMCTL=0x03`, 在计数器值和 `TIMERx_CHxCV` 寄存器的值匹配时, 可以翻转输出信号。

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

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

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。

正交译码器

正交译码器功能使用由 `TIMERx_CH0` 和 `TIMERx_CH1` 引脚生成的 `CI0` 和 `CI1` 正交信号各自相互作用产生计数值。在每个输入源改变期间, `DIR` 位被硬件自动改变。

输入源可以是只有 `CI0`, 可以只有 `CI1`, 或者可以同时有 `CI0` 和 `CI1`, 通过设置 `TSCFGy[4:0] (y = 0, 1, 2, 13, 14) != 5'b00000` 来选择使用哪种模式。计数器计数方向改变的机制如 [表 12-6. 不同译码器模式下的计数方向](#) 所示。其中, `CI0FE0`、`CI1FE1` 是经过滤波和极性选择后的 `CI0`、`CI1` 信号。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 12-8. 不同译码器模式下的计数方向

计数模式	电平	CI0FE0		CI1FE1	
		上升	下降	上升	下降
正交译码器模式0 TSCFG0[4:0] != 5'b00000	CI1FE1=1	向下	向上	-	-
	CI1FE1=0	向上	向下	-	-
正交译码器模式1 TSCFG1[4:0] != 5'b00000	CI0FE0=1	-	-	向上	向下
	CI0FE0=0	-	-	向下	向上
正交译码器模式2 TSCFG2[4:0] != 5'b00000	CI1FE1=1	向下	向上	X	X
	CI1FE1=0	向上	向下	X	X
	CI0FE0=1	X	X	向上	向下
	CI0FE0=0	X	X	向下	向上
正交译码器模式3 TSCFG13[4:0] != 5'b00000	CI0FE1=1	向下	向上	-	-
	CI0FE1=0	-	-	-	-
正交译码器模式4	CI1FE0=1	-	-	向上	向下

计数模式	电平	CI0FE0		CI1FE1	
		上升	下降	上升	下降
TSCFG14[4:0] != 5'b00000	CI1FE0=0	-	-	-	-

注意：“-”意思是“无计数”；“X”意思是不可能。“0”意思是低电平，“1”意思是高电平。

图 12-61. 译码器接口模式下计数器运行例子

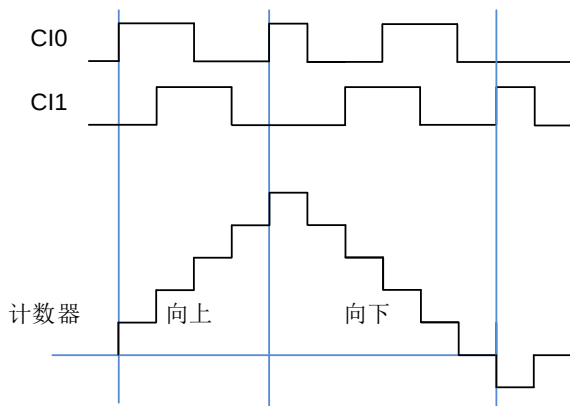
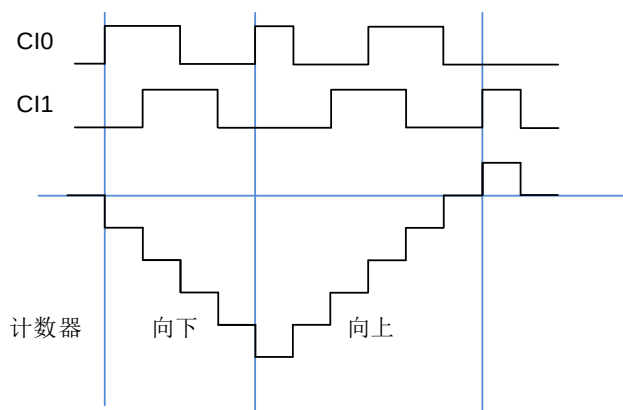


图 12-62. CI0FE0 极性反相的译码器接口模式下的例子



当正交译码器模式下计数器计数方向发生变化时，TIMERx_CTL0 寄存器中的 DIR 位发生改变。

正交译码器信号检测

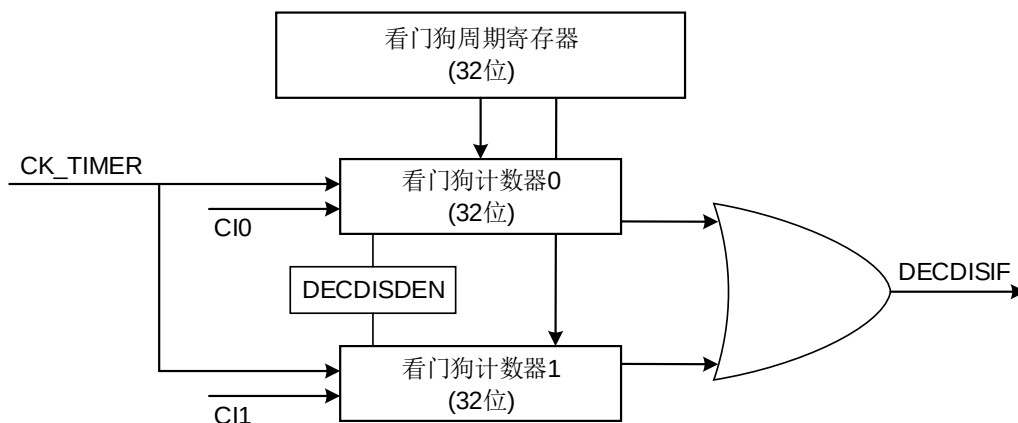
支持两种正交译码器信号检测：信号跳变检测和断线检测。

正交译码器信号跳变检测功能可用于检测两个正交译码器输入信号 CI0、CI1 的电平跳变沿（上升沿或下降沿）是否同时发生，可通过将 TIMERx_CTL2 寄存器中的 DECJDEN 位置 1 来使能。当 DECJDEN=1 时，若两个正交信号 CI0 和 CI1 的电平跳变同时发生，则中断标志位 DECJIF 置位。若 DECJIE=1，则相应的中断产生。

正交译码器信号断线检测功能可用于检测正交译码器输入信号 $CI0$ 、 $CI1$ 是否正常，可通过将 $TIMERx\_CTL2$ 寄存器中的 $DECDISDEN$ 位置 1 来使能。正交译码器信号检测模块包括 2 个 32 位的看门狗计数器和 1 个周期寄存器，具体如 [图 12-63. 正交译码器信号断线检测框图](#) 所示， $CI0FE0$ 、 $CI1FE1$ 信号分别用于复位 2 个看门狗计数器。

当 $DECDISDEN=1$ 时，2 个看门狗计数器同时开始向上计数，若看门狗计数器计数到看门狗周期值（该值由 $TIMERx\_WDGPER$ 寄存器中的 $WDGPER[31:0]$ 位域确定），则看门狗计数器计数超时，中断标志位 $DECDISIF$ 置位。若 $DECDISIE=1$ ，则相应的中断产生。

图 12-63. 正交译码器信号断线检测框图



译码器

译码器功能有 4 种模式：译码器模式 0~3，通过设置 $TSCFGy[4:0](y = 9, 10, 11, 12) \neq 5'b00000$ 来选择。这 4 种计数模式下的输入源有两个： $CI0$ 和 $CI1$ ，其中， $CI0FE0$ 、 $CI1FE1$ 是经过滤波和极性选择后的 $CI0$ 、 $CI1$ 信号。

使用译码器模式 0/1 时， $CI0FE0$ 作为计数方向信号， $CI1FE1$ 作为计数脉冲。

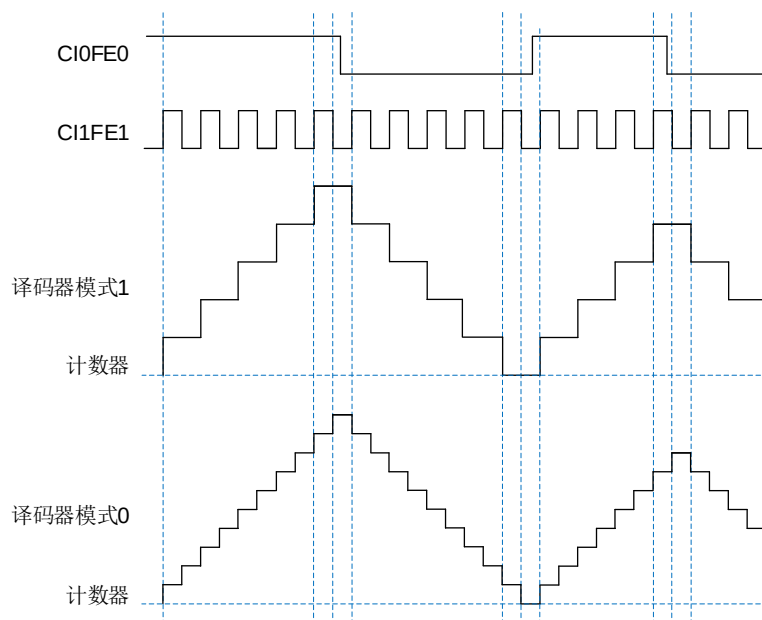
其中， $CH0P$ 用于计数方向选择：当 $CH0P=0$ 时， $CI0FE0$ 为高电平时向上计数， $CI0FE0$ 为低电平时向下计数；当 $CH0P=1$ 时， $CI0FE0$ 为高电平时向下计数， $CI0FE0$ 为低电平时向上计数。

$CH1P$ 用于选择 $CI1FE1$ 信号的计数边沿：译码器模式 0 时，计数器在 $CI1FE1$ 信号的上升沿和下降沿进行计数；译码器模式 1 时，当 $CH1P=0$ 时，在 $CI1FE1$ 信号的上升沿计数，当 $CH1P=1$ 时，在 $CI1FE1$ 信号的下降沿计数。更多译码器模式 1 的细节如 [表 12-9. 译码器模式 1 的计数情况](#) 和 [图 12-64. 译码器模式 0/1 计数器运行实例 \(\$CH1P=0\$ \)](#) 所示。

表 12-9. 译码器模式 1 的计数情况

CH1P	电平	计数器计数情况
0	$CI0FE0$ 为高电平	计数器在 $CI1FE1$ 信号的上升沿向上计数
	$CI0FE0$ 为低电平	计数器在 $CI1FE1$ 信号的上升沿向下计数
1	$CI0FE0$ 为高电平	计数器在 $CI1FE1$ 信号的下降沿向上计数
	$CI0FE0$ 为低电平	计数器在 $CI1FE1$ 信号的下降沿向下计数

图 12-64. 译码器模式 0/1 计数器运行实例 (CH1P=0)



译码器模式 2/3 由 CI0FE0、CI1FE1 信号各自相互作用产生计数值，DIR 位被硬件自动改变。

译码器模式 2 时，计数器在 CI0FE0、CI1FE1 信号的上升沿和下降沿进行计数，计数方向由 CH0P 和 CH1P 确定；译码器模式 3，计数器在 CI0FE0、CI1FE1 信号的上升沿或下降沿进行计数，当 CHxP=0 时，信号为高电平时计数，或在信号的下降沿计数；当 CHxP=1 时，信号为低电平时计数，或在信号上升沿计数。具体情况请见 [图 12-65. 译码器模式 2 / 3 计数器运行实例 \(CH0P / CH1P=0\)](#) 和 [表 12-10. 译码器模式 2 / 3 下的计数方向](#)。

图 12-65. 译码器模式 2 / 3 计数器运行实例 (CH0P / CH1P=0)

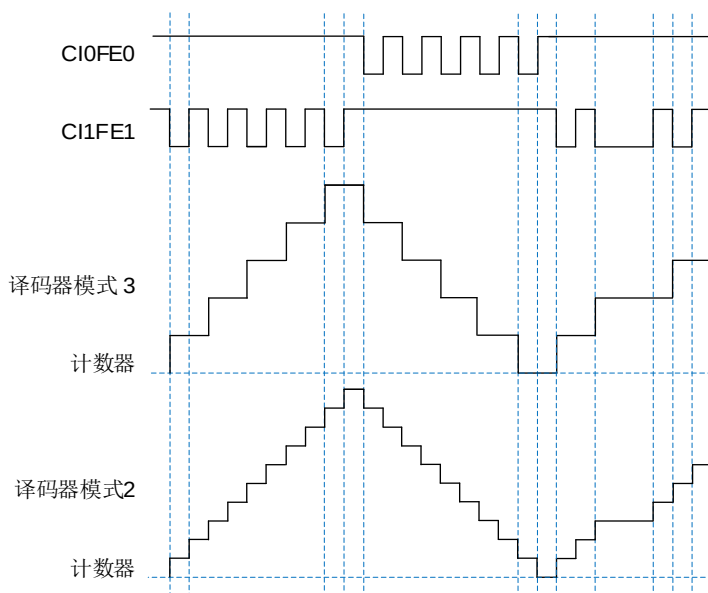


表 12-10. 译码器模式 2 / 3 下的计数方向

计数模式	极性	电平	CI0FE0		CI1FE1	
			上升	下降	上升	下降
译码器模式2 TSCFG11[4:0]! = 5'b00000	CHxP=0 (x = 0, 1)	CI1FE1=1	向下	向下	x	x
		CI1FE1=0	-	-	x	x
		CI0FE0=1	x	x	向上	向上
		CI0FE0=0	x	x	-	-
	CHxP=1 (x = 0, 1)	CI1FE1=1	-	-	x	x
		CI1FE1=0	向下	向下	x	x
		CI0FE0=1	x	x	-	-
		CI0FE0=0	x	x	向上	向上
译码器模式3 TSCFG12[4:0]! = 5'b00000	CHxP=0 (x = 0, 1)	CI1FE1=1	-	向下	x	x
		CI1FE1=0	-	-	x	x
		CI0FE0=1	x	x	-	向上
		CI0FE0=0	x	x	-	-
	CHxP=1 (x = 0, 1)	CI1FE1=1	-	-	x	x
		CI1FE1=0	向下	-	x	x
		CI0FE0=1	x	x	-	-
		CI0FE0=0	x	x	向上	-

当译码器模式下计数器计数方向发生变化时，TIMERx_CTL0 寄存器中的 DIR 位改变。

正交译码器和译码器的时钟输出

定时器可以通过 TRGO 输出译码器时钟输出信号。该功能仅用于正交译码器模式 0~4 和译码器模式 0~3，通过将 TIMERx_CTL1 寄存器中 MMC[3:0]位域设置为 4'b1000 来使能。

霍尔传感器接口功能

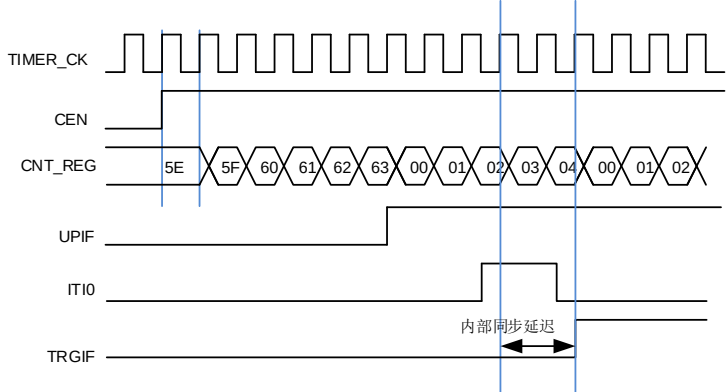
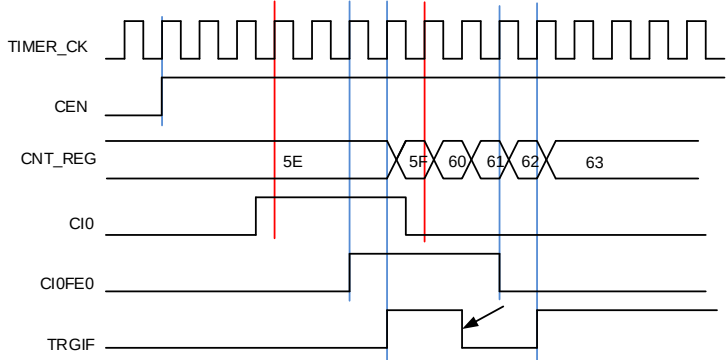
参考 [高级定时器 \(TIMERx, x=0, 7\) 霍尔传感器接口功能](#)。

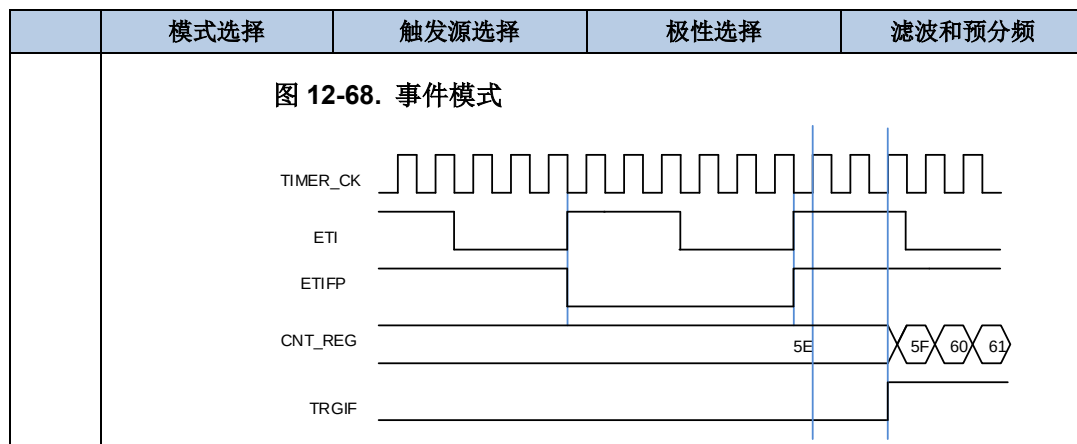
主-从管理

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等，可以通过设置 SYSCFG_TIMERxCFG(x=1..4)寄存器中的 TSCFGy[4:0] (y=3..6)位域来确定，具体的输入触发源可以通过 TSCFGy[4:0] (y=3..6)位域值来选择。

表 12-11. 从模式列表和举例（通用定时器 L0）

	模式选择	触发源选择	极性选择	滤波和预分频
列举	TSCFGy[4:0] y=3: 复位模式 y=4: 暂停模式 y=5: 事件模式	TSCFGy[4:0] 00000: Mode disable 00001: ITIO 00010: ITI1	如果触发源是 CI0FE0 或者 CI1FE1，配置 CHxP 和 CHxNP 来选择 极性和反相。	触发源 ITIx，滤波和预 分频不可用 触发源 CIX，配置 CHxCAPFLT 设置滤

	模式选择	触发源选择	极性选择	滤波和预分频
	y=6: 外部时钟模式0	00011: ITI2 00100: ITI3 00101: CI0F_ED 00110: CI0FE0 00111: CI1FE1 01000: ETIFP ⁽¹⁾ 01001: CI2FE2 01010: CI3FE3	如果触发源是 ETIFP (滤波后的 ETI 外部触发输入), 配置 ETP 选择极性和反相	波, 分频不可用 触发源是 ETIFP, 配置 ETFC 设置滤波, 配置 ETPSC 设置预分频
例1	复位模式 当触发输入上升沿, 计数器清零重启	TSCFG3[4:0] 5'b00001, 选择 ITI0 为触发源	触发源是 ITI0, 极性选择不可用	触发源是 ITI0, 滤波和预分频不可用
	图 12-66. 复位模式 			
例2	暂停模式 当触发输入为低的时候, 计数器暂停计数	TSCFG4[4:0] =5'b00110, 选择 CI0FE0 为触发源	TI0S=0 (非异或) [CH0NP=0, CH0P=0] 不反相, 在上升沿捕获	在这个例子中滤波被旁路
	图 12-67. 暂停模式 			
例3	事件模式 触发输入的上升沿计数器开始计数	TSCFG5[4:0] =5'b01000, 选择 ETIFP 为触发源	ETP = 0 没有极性改变	ETPSC = 1, 2分频 ETFC = 0, 无滤波



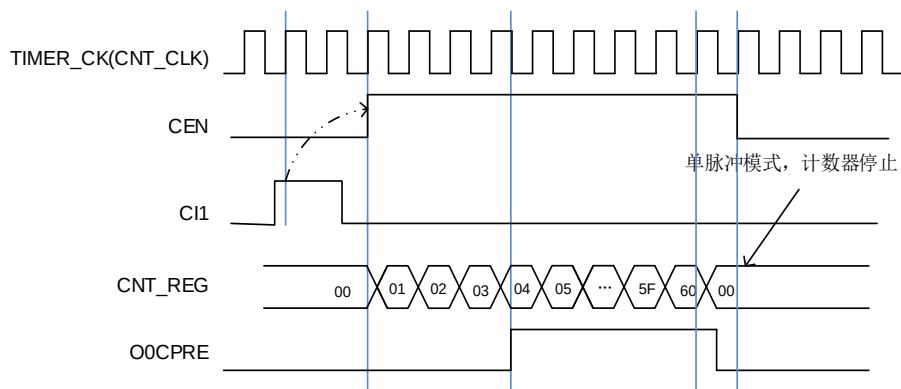
单脉冲模式

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

一旦设置定时器运行在单脉冲模式下,没有必要设置 `TIMERx_CTL0` 寄存器的定时器使能位 `CEN=1` 来使能计数器。触发信号沿或者软件写 `CEN=1` 都可以产生一个脉冲,此后 `CEN` 位一直保持为 1 直到更新事件发生或者 `CEN` 位被软件写 0。如果 `CEN` 位被软件清 0,计数器停止工作,计数值被保持。如果 `CEN` 值被硬件更新事件自动清 0,计数器将被再次初始化。

在单脉冲模式下,有效的外部触发边沿会将 `CEN` 位置 1,使能计数器。然而,执行计数值和 `TIMERx_CHxCV` 寄存器值的比较结果依然存在一些时钟延迟。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。

图 12-69. 单脉冲模式, `TIMERx_CHxCV = 0x04` `TIMERx_CAR=0x60`



定时器互连

参考 [高级定时器 \(TIMERx, x=0, 7\) 定时器互连](#)。

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。当然，必须要使能 DMA 请求，一些内部中断事件可以产生 DMA 请求。当中断事件发生，TIMERx 会给 DMA 发送请求。DMA 配置成 M2P 模式，PADDR 是 TIMERx_DMATB 寄存器地址，DMA 就会访问 TIMERx_DMATB 寄存器。实际上，TIMERx_DMATB 寄存器只是一个缓冲，定时器会将 TIMERx_DMATB 映射到一个内部寄存器，这个内部寄存器由 TIMERx_DMACFG 寄存器中的 DMATA 来指定。如果 TIMERx_DMACFG 寄存器的 DMATC 位域值为 0，表示 1 次传输，定时器的发送 1 个 DMA 请求就可以完成。如果 TIMERx_DMACFG 寄存器的 DMATC 位域值不为 1，例如其值为 3，表示 4 次传输，定时器就需要再多发 3 次 DMA 请求。在这 3 次请求下，DMA 对 TIMERx_DMATB 寄存器的访问会映射到访问定时器的 DMATA+0x4, DMATA+0x8, DMATA+0xc 寄存器。总之，发生一次 DMA 内部中断请求，定时器会连续发送 (DMATC+1) 次请求。

如果再来 1 次 DMA 请求事件，TIMERx 将会重复上面的过程。

定时器调试模式

当 Cortex®-M33 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。

12.2.5. 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

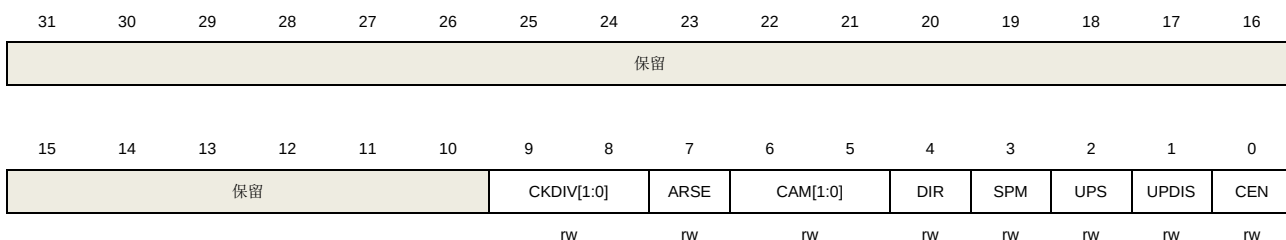
TIMER4 基地址: 0x4000 0C00

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。



位/位域	名称	描述
31:10	保留	必须保持复位值
9:8	CKDIV[1:0]	时钟分频 通过软件配置 CKDIV, 规定定时器时钟 (CK_TIMER) 与死区时间和采样时钟 (DTS) 之间的分频系数, 死区发生器和数字滤波器会用到 DTS 时间。 00: $f_{DTS} = f_{CK\_TIMER}$ 01: $f_{DTS} = f_{CK\_TIMER} / 2$ 10: $f_{DTS} = f_{CK\_TIMER} / 4$ 11: 保留
7	ARSE	自动重载影子使能 0: 禁能 TIMERx_CAR 寄存器的影子寄存器 1: 使能 TIMERx_CAR 寄存器的影子寄存器
6:5	CAM[1:0]	计数器对齐模式选择 00: 无中央对齐模式 (边沿对齐模式)。DIR 位指定了计数方向。 01: 中央对齐向下计数置 1 模式。计数器在中央计数模式计数, 通道被配置在输出模式 (TIMERx_CHCTL0 寄存器中 CHxMS=3'b000), 只有在向下计数时, 通道的比较中断标志置 1 10: 中央对齐向上计数置 1 模式。计数器在中央计数模式计数, 通道被配置在输出模式 (TIMERx_CHCTL0 寄存器中 CHxMS=3'b000), 只有在向上计数时, 通道

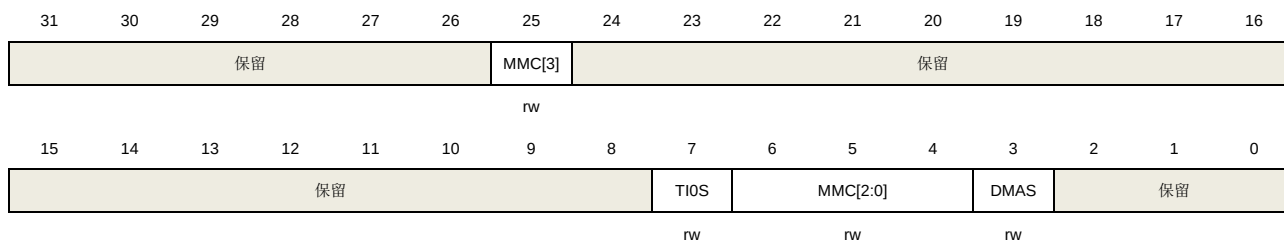
		的比较中断标志置 1
		11: 中央对齐上下计数置 1 模式。计数器在中央计数模式计数，通道被配置在输出模式（TIMERx_CHCTL0 寄存器中 CHxMS=3'b000），在向上和向下计数时，通道的比较中断标志都会置 1
		当计数器使能以后，该位不能从 0x00 切换到非 0x00 状态。
4	DIR	方向 0: 向上计数 1: 向下计数 当计数器配置为中央对齐模式或译码器模式时，该位为只读。
3	SPM	单脉冲模式 0: 更新事件发生后，计数器继续计数 1: 在下一更新事件发生时，CEN 硬件清零并且计数器停止计数
2	UPS	更新请求源 软件配置该位，选择更新事件源。 0:使能后，下述任一事件产生更新中断或 DMA 请求： – UPG 位被置 1 – 计数器上溢/下溢 – 从模式控制器产生的更新 1:使能后只有计数器上溢/ 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 – 计数器上溢/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、暂停模式和译码器模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。

控制寄存器 1（TIMERx_CTL1）

地址偏移：0x04

复位值：0x0000 0000

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	MMC[3]	主模式控制 请参考 MMC[2:0] 的描述。
24:8	保留	必须保持复位值
7	TI0S	通道 0 触发输入选择 0: 选择 TIMEx_CH0 引脚作为通道 0 的触发输入 1: 选择 TIMEx_CH0, CH1 和 CH2 引脚异或的结果作为通道 0 的触发输入
6:4	MMC[2:0]	主模式控制 这些位控制 TRGO 信号的选择，TRGO 信号由主定时器发给从定时器用于同步功能 0000: 复位。TIMEx_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。 0001: 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。 0010: 更新。主模式控制器选择更新事件作为 TRGO。 0011: 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 0100: 比较。在这种模式下主模式控制器选择 O0CPRE 信号被用于作为触发输出 TRGO 0101: 比较。在这种模式下主模式控制器选择 O1CPRE 信号被用于作为触发输出 TRGO 0110: 比较。在这种模式下主模式控制器选择 O2CPRE 信号被用于作为触发输出 TRGO 0111: 比较。在这种模式下主模式控制器选择 O3CPRE 信号被用于作为触发输出 TRGO 1000: 译码器时钟输出。在这种模式下主模式控制器选择译码器时钟信号作为触发输出 TRGO。此值仅用于正交译码器模式 0~4 和译码器模式 0~3。

		1001~1111: 保留
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 CHx 的 DMA 请求 1: 当更新事件发生, 发送通道 CHx 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	保留						
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	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	从模式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 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 TSCFGy[4:0](y=3,4,5)位域的值不能为 5b'01000。 如果外部时钟模式 0 和外部时钟模式 1 同时被使能, 外部时钟的输入是 ETIFP。 注意: 外部时钟模式 0 使能在 SYSCFG_TIMERxCFG1 寄存器中的 TSCFG6[4:0] 位域。
13:12	ETPSC[1:0]	外部触发预分频 外部触发信号 ETI 的频率不能超过 TIMER_CK 频率的 1/4。当输入较快的外部时钟时, 可以使用预分频降低 ETIFP 的频率。 00: 预分频禁能 01: ETI 频率被 2 分频

		10: ETI 频率被 4 分频
		11: ETI 频率被 8 分频
11:8	ETFC[3:0]	外部触发滤波控制 数字滤波器是一个事件计数器，它记录到 N 个事件后会产生一个输出的跳变。这些位定义了对 ETI 信号采样的频率和对 ETI 数字滤波的带宽。 0000: 滤波器禁能 $f_{SAMP} = f_{DTS}$, $N=1$ 0001: $f_{SAMP} = f_{CK\_TIMER}$, $N=2$ 0010: $f_{SAMP} = f_{CK\_TIMER}$, $N=4$ 0011: $f_{SAMP} = f_{CK\_TIMER}$, $N=8$ 0100: $f_{SAMP} = f_{DTS}/2$, $N=6$ 0101: $f_{SAMP} = f_{DTS}/2$, $N=8$ 0110: $f_{SAMP} = f_{DTS}/4$, $N=6$ 0111: $f_{SAMP} = f_{DTS}/4$, $N=8$ 1000: $f_{SAMP} = f_{DTS}/8$, $N=6$ 1001: $f_{SAMP} = f_{DTS}/8$, $N=8$ 1010: $f_{SAMP} = f_{DTS}/16$, $N=5$ 1011: $f_{SAMP} = f_{DTS}/16$, $N=6$ 1100: $f_{SAMP} = f_{DTS}/16$, $N=8$ 1101: $f_{SAMP} = f_{DTS}/32$, $N=5$ 1110: $f_{SAMP} = f_{DTS}/32$, $N=6$ 1111: $f_{SAMP} = f_{DTS}/32$, $N=8$
7	MSM	主-从模式 该位被用来同步被选择的定时器同时开始计数。通过 TRGI 和 TRGO，定时器被连接在一起，TRGO 用做启动事件。 0: 主从模式禁能 1: 主从模式使能
6:0	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														DECDISE	DECJIE
														rw	rw
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	CH0IE	UPIE

362

		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
31:18	保留	必须保持复位值
17	DECDISIF	正交译码器信号断线中断标志位 0: 无正交译码器信号断线中断发生 1: 正交译码器信号断线中断发生 注意: 该位仅用于正交译码器信号断线检测使能 (DECDISDEN =1) 时。

16	DECJIF	正交译码器信号跳变（两个信号同时发生跳变）中断标志位 0：无正交译码器信号跳变中断发生 1：正交译码器信号跳变中断发生 注意： 该位仅用于正交译码器信号同时跳变检测使能（DECJDEN =1）时。
15:13	保留	必须保持复位值
12	CH3OF	通道 3 捕获溢出标志 参见 CH0OF 描述。
11	CH2OF	通道 2 捕获溢出标志 参见 CH0OF 描述。
10	CH1OF	通道 1 捕获溢出标志 参见 CH0OF 描述。
9	CH0OF	通道 1 捕获溢出标志 当通道 0 被配置为输入模式时，在 CH0IF 标志位已经被置 1 后，捕获事件再次发生时，该标志位可以由硬件置 1。该标志位由软件清 0。 0：无捕获溢出中断发生 1：发生了捕获溢出中断
8:7	保留	必须保持复位值
6	TRGIF	触发中断标志 当发生触发事件时，此标志由硬件置 1。此位由软件清 0。当从模式控制器处于除暂停模式外的其它模式时，在 TRGI 输入端检测到有效边沿，产生触发事件。当从模式控制器处于暂停模式时，TRGI 的任意边沿都可以产生触发事件。 0：无触发事件产生 1：触发中断产生
5	保留	必须保持复位值
4	CH3IF	通道 3 比较/捕获中断标志 参见 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。 当通道 0 在输入模式下时，通过读 TIMERx_CH0CV 寄存器可以清零该位。 0：无通道 0 中断发生

		1: 通道 0 中断发生
0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置 1，软件清 0。 0: 无更新中断发生 1: 发生更新中断

软件事件产生寄存器 (TIMERx_SWEVG)

地址偏移: 0x14

复位值: 0x0000 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
									W		W	W	W	W	W

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 TIMERx_CH0CV 寄存器捕获，如果 CH0IF 标志位已经为 1，则 CH0OF 标志位被

置 1。

0：不产生通道 0 捕获或比较事件

1：发生通道 0 捕获或比较事件

0 UPG

更新事件产生

此位由软件置 1，被硬件自动清 0。当此位被置 1，如果选择了中央对齐或向上计数模式，计数器被清 0。否则（向下计数模式）计数器将载入自动重载值，预分频计数器将同时被清除。

0：无更新事件产生

1：产生更新事件

通道控制寄存器 0 (TIMERx_CHCTL0)

地址偏移：0x18

复位值：0x0000 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
CH1MS [2]	CH0MS [2]	保留					CH1COM	保留							CH0COM
							CTL[3]								CTL[3]
							保留								保留
rw		rw		rw										rw	
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	CH1MS[2]	通道 1 I/O 模式选择 参考 CH1MS[1:0]描述。
30	CH0MS[2]	通道 0 I/O 模式选择 参考 CH0MS[1:0]描述。
29:25	保留	必须保持复位值
24	CH1COMCTL[3]	通道 1 输出比较控制 参见 CH0COMCTL[2:0]描述
23:17	保留	必须保持复位值
16	CH0COMCTL[3]	通道 0 输出比较控制 参见 CH0COMCTL[2:0]描述

15	CH1COMCEN	通道 1 输出比较清 0 使能 参见 CH0COMCEN 描述。
14:12	CH1COMCTL[2:0]	通道 1 输出比较模式 参见 CH0COMCTL 描述。
11	CH1COMSEN	通道 1 输出比较影子寄存器使能 参见 CH0COMSEN 描述。
10	CH1COMFEN	通道 1 输出比较快速使能 参见 CH0COMFEN 描述
9:8	CH1MS[1:0]	通道 1 模式选择 CH1MS[2:0]位域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。 000: 通道 1 配置为输出 001: 通道 1 配置为输入，IS1 映射在 CI1FE1 上 010: 通道 1 配置为输入，IS1 映射在 CI0FE1 上 011: 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 SYSCFG_TIMERxCFG2(x=1..4)寄存器中的 TSCFG15[4:0]位域选择）。 100~111: 保留
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 CH0COMCTL[3]和 CH0COMCTL[2:0]位域定义了输出准备信号 O0CPRE 的动作，而 O0CPRE 决定了 CH0_O 的值。O0CPRE 高电平有效，而 CH0_O 的有效电平取决于 CH0P 位。 0000: 时基。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 0001: 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。 0010: 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。 0011: 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。 0100: 强制为低。强制 O0CPRE 为低电平 0101: 强制为高。强制 O0CPRE 为高电平 0110: 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。

0111: 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。

1000~1111: 保留

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时基模式切换到 PWM 模式时，`O0CPRE` 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，<code>TIMERx_CH0CV</code> 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（<code>TIMERx_CTL0</code> 寄存器的 <code>SPM = 1</code>），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，<code>CH0_O</code> 被设置为比较电平而与比较结果无关。 0: 通道 0 输出比较快速禁能。当触发器的输入有一个有效沿时，激活 <code>CH0_O</code> 输出的最小延时为 5 个时钟周期 1: 通道 0 输出比较快速使能。当触发器的输入有一个有效沿时，激活 <code>CH0_O</code> 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（<code>TIMERx_CHCTL2</code> 寄存器的 <code>CH0EN</code> 位被清 0）时，<code>CH0MS[2:0]</code> 位才可以写。 000: 通道 0 配置为输出 001: 通道 0 配置为输入，IS0 映射在 <code>CI0FE0</code> 上 010: 通道 0 配置为输入，IS0 映射在 <code>CI1FE0</code> 上 011: 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（由 <code>SYSCFG_TIMERx_CFG2(x=1..4)</code> 寄存器中的 <code>TSCFG15[4:0]</code> 位域选择）。 100~111: 保留

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 <code>CH0CAPFLT</code> 描述。
11:10	CH1CAPPSC[1:0]	通道 1 输入捕获预分频器 参见 <code>CH0CAPPSC</code> 描述。

9:8	CH1MS[1:0]	通道 1 模式选择 与输出模式相同。
7:4	CH0CAPFLT[3:0]	通道 0 输入捕获滤波控制 数字滤波器由一个事件计数器组成，它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 CH0 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: $f_{SAMP} = f_{CK\_TIMER}$, $N=2$ 0010: $f_{SAMP} = f_{CK\_TIMER}$, $N=4$ 0011: $f_{SAMP} = f_{CK\_TIMER}$, $N=8$ 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 <code>TIMERx_CHCTL2</code> 寄存器中的 <code>CH0EN = 0</code> 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
CH3MS	CH2MS	保留						CH3COM	保留						CH2COM
[2]	[2]							CTL[3]							CTL[3]

							保留									保留
rw		rw		rw										rw		
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		rw		

输出比较模式:

位/位域	名称	描述
31	CH3MS[2]	通道 3 I/O 模式选择 参考 CH3MS[1:0]描述。
30	CH2MS[2]	通道 2 I/O 模式选择 参考 CH2MS[1:0]描述。
29:25	保留	必须保持复位值。
24	CH3COMCTL[3]	通道 3 输出比较控制 请参考 CH2COMCTL[2:0]描述
23:17	保留	必须保持复位值
16	CH2COMCTL[3]	通道 2 输出比较控制 请参考 CH2COMCTL[2:0]描述
15	CH3COMCEN	通道 3 输出比较清 0 使能 参见 CH0COMCEN 描述。
14:12	CH3COMCTL[2:0]	通道 3 输出比较模式 参见 CH0COMCTL 描述。
11	CH3COMSEN	通道 3 输出比较影子寄存器使能 参见 CH0COMSEN 描述。
10	CH3COMFEN	通道3输出比较快速使能 参见 CH2COMFEN 描述
9:8	CH3MS[1:0]	通道 3 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 CH3EN 位被清 0) 时这些位才可以写。 000: 通道 3 配置为输出 001: 通道 3 配置为输入, IS3 映射在 CI3FE3 上 010: 通道 3 配置为输入, IS3 映射在 CI2FE3 上 011: 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 SYSCFG_TIMERxCFG2(x=1..4)寄存器中的 TSCFG15[4:0]位域选择)。 100~111: 保留

7	CH2COMCEN	通道 2 输出比较清 0 使能 当此位被置 1，当检测到 ETIF 输入高电平时，O2CPRE 参考信号被清 0 0：使能通道 2 输出比较清零 1：禁止通道 2 输出比较清零
6:4	CH2COMCTL[2:0]	通道 2 输出比较模式 此位定义了输出准备信号 O2CPRE 的动作，而 O2CPRE 决定了 CH2_O 的值。 O2CPRE 高电平有效，而 CH2_O 的有效电平取决于 CH2P 位。 0000：时基。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 0001：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。 0010：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。 0011：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。 0100：强制为低。强制 O2CPRE 为低电平 0101：强制为高。强制 O2CPRE 为高电平 0110：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。 0111：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 1 或 PWM 模式 2 中，只有当比较结果改变了或者输出比较模式中从时基模式切换到 PWM 模式时，O2CPRE 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：通道2输出比较快速禁能。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：通道0输出比较快速使能。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。 000: 通道 2 配置为输出 001: 通道 2 配置为输入，IS2 映射在 CI2FE2 上 010: 通道 2 配置为输入，IS2 映射在 CI3FE2 上 011: 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（由 SYSCFG_TIMERxCFG2(x=1..4)寄存器中的 TSCFG15[4:0]位域选择）。 100~111: 保留

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，它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 CI2 输入信号的采样频率和数字滤波器的长度。 0000: 无滤波器, $f_{SAMP} = f_{DTS}$, $N=1$ 0001: $f_{SAMP} = f_{CK\_TIMER}$, $N=2$ 0010: $f_{SAMP} = f_{CK\_TIMER}$, $N=4$ 0011: $f_{SAMP} = f_{CK\_TIMER}$, $N=8$ 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: $f_{SAMP}=f_{DTS}/32$, $N=8$

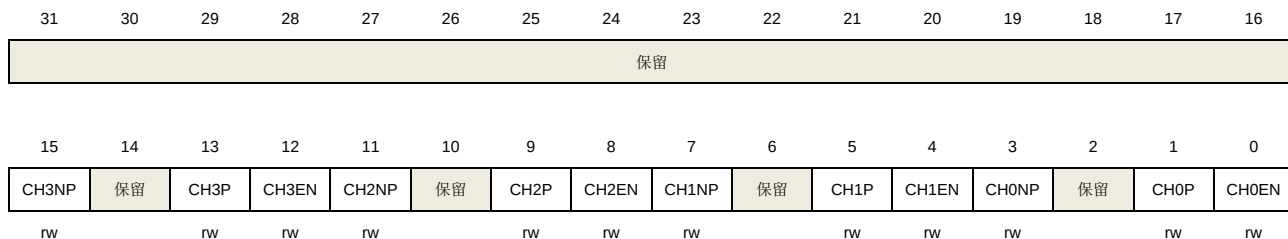
3:2	CH2CAPPSC[1:0]	通道 2 输入捕获预分频器 这 2 位定义了通道 2 输入的预分频系数。当 <code>TIMERx_CHCTL2</code> 寄存器中的 <code>CH2EN = 0</code> 时，则预分频器复位。 00: 无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获 01: 每 2 个事件触发一次捕获 10: 每 4 个事件触发一次捕获 11: 每 8 个事件触发一次捕获
1:0	CH2MS[1:0]	通道 2 模式选择 与输出比较模式相同。

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	CH3NP	通道 3 互补输出极性 参考 CH0NP 描述。
14	保留	必须保持复位值。
13	CH3P	通道 3 极性 参考 CH0P 描述。
12	CH3EN	通道 3 使能 参考 CH0EN 描述。
11	CH2NP	通道 2 互补输出极性 参考 CH0NP 描述。
10	保留	必须保持复位值
9	CH2P	通道 2 极性

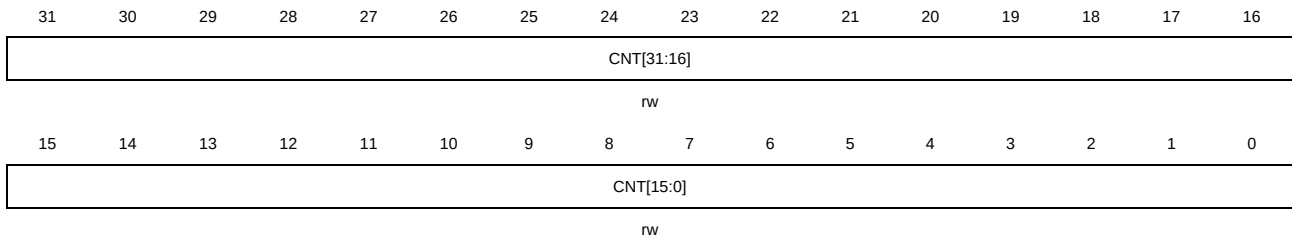
		参考 CH0P 描述。
8	CH2EN	通道 2 使能 参考 CH0EN 描述。
7	CH1NP	通道 1 互补输出极性 参考 CH0NP 描述。
6	保留	必须保持复位值
5	CH1P	通道 1 极性 参考 CH0P 描述。
4	CH1EN	通道 1 使能 参考 CH0EN 描述。
3	CH0NP	通道 0 互补输出极性 当通道 0 配置为输出模式，该位必须保持复位值。 当通道 0 配置为输入模式时，此位和 CH0P 联合使用，作为输入信号 CIO 的极性选择控制信号。
2	保留	必须保持复位值
1	CH0P	通道 0 极性 当通道 0 配置为输出模式时，此位定义了输出信号极性。 0: 通道 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，此位定义了通道 0 输入信号极性。[CH0NP, CH0P] 将选择 CIOFE0 或者 CI1FE0 的有效边沿或者捕获极性。 00: 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。 01: 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。 10: 保留。 11: 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。



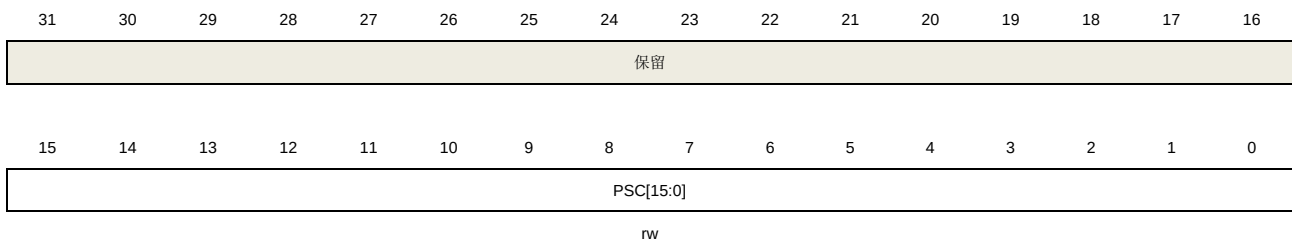
位/位域	名称	描述
31:16	CNT[31:16]	这些位是当前的计数值。写操作能改变计数器值。（bit 16 到 bit 31） 该位域仅用于 TIMER1 。
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

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。



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。

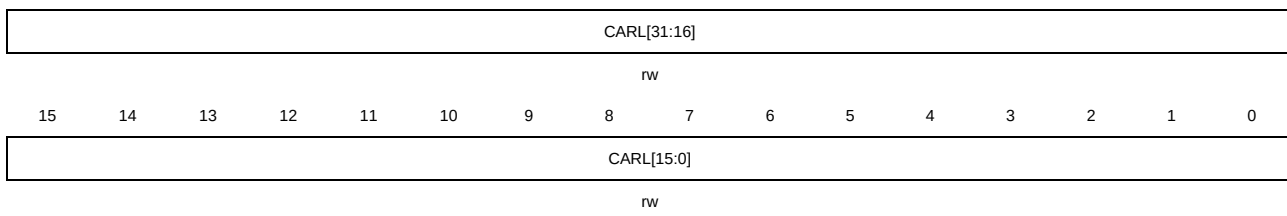
计数器自动重载寄存器（TIMERx_CAR）

地址偏移：0x2C

复位值：0xFFFF FFFF

该寄存器只能按字（32 位）访问。





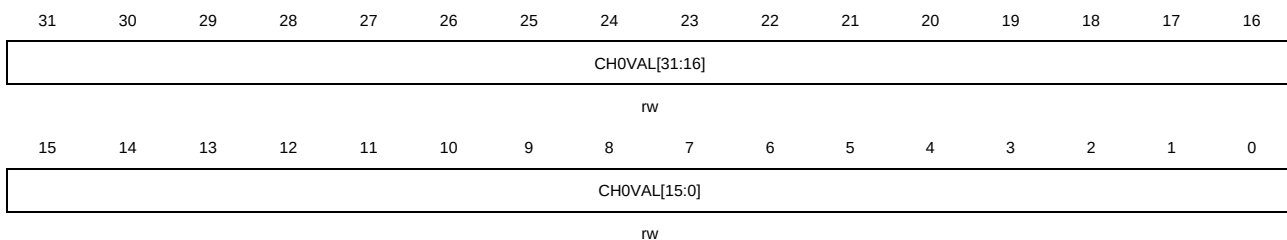
位/位域	名称	描述
31:16	CARL[31:16]	计数器自动重载值（bit 16 到 bit 31） 该位域仅用于 TIMER1 。
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

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。



位/位域	名称	描述
31:16	CH0VAL[31:16]	通道 0 的捕获或比较值（bit 16 到 bit 31） 该位域仅用于 TIMER1 。
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。

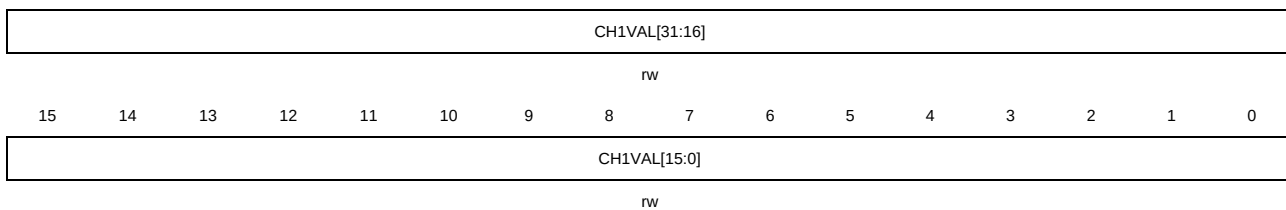
通道 1 捕获/比较值寄存器（TIMERx_CH1CV）

地址偏移：0x38

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。





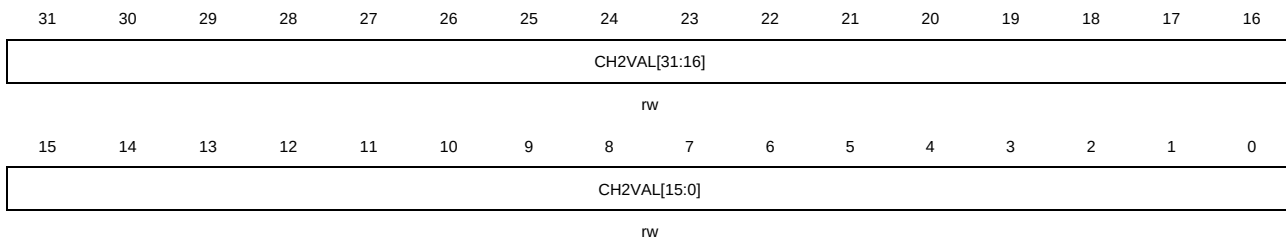
位/位域	名称	描述
31:16	CH1VAL[31:16]	通道 1 的捕获或比较值（bit 16 到 bit 31） 该位域仅用于 TIMER1。
15:0	CH1VAL[15:0]	通道 1 的捕获或比较值 当通道 1 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 1 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

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。



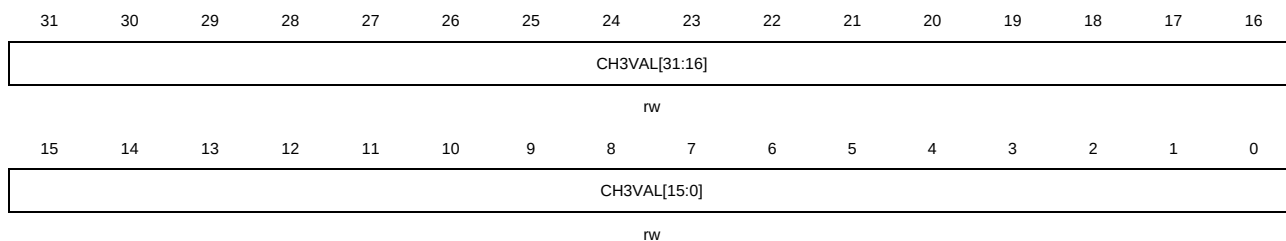
位/位域	名称	描述
31:16	CH2VAL[31:16]	通道 2 的捕获或比较值（bit 16 到 bit 31） 该位域仅用于 TIMER1。
15:0	CH2VAL[15:0]	通道 2 的捕获或比较值 当通道 2 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 2 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

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。



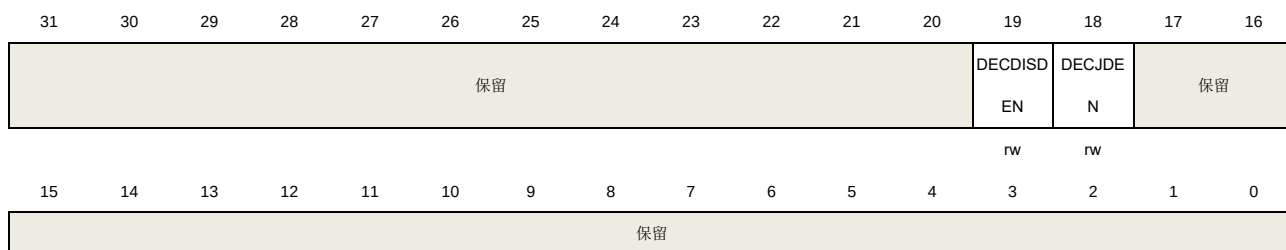
位/位域	名称	描述
31:16	CH3VAL[31:16]	通道 3 的捕获或比较值（bit 16 到 bit 31） 该位域仅用于 TIMER1。
15:0	CH3VAL[15:0]	通道 3 的捕获或比较值 当通道 3 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 3 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

控制寄存器 2（TIMERx_CTL2）

地址偏移：0x74

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	DECDISDEN	正交译码器信号断线检测使能 0：正交译码器信号断线检测禁能 1：正交译码器信号断线检测使能
18	DECJDEN	正交译码器信号跳变（两个信号同时发生跳变沿）检测使能 0：正交译码器信号跳变（两个信号同时发生跳变沿）检测禁能 1：正交译码器信号跳变（两个信号同时发生跳变沿）检测使能

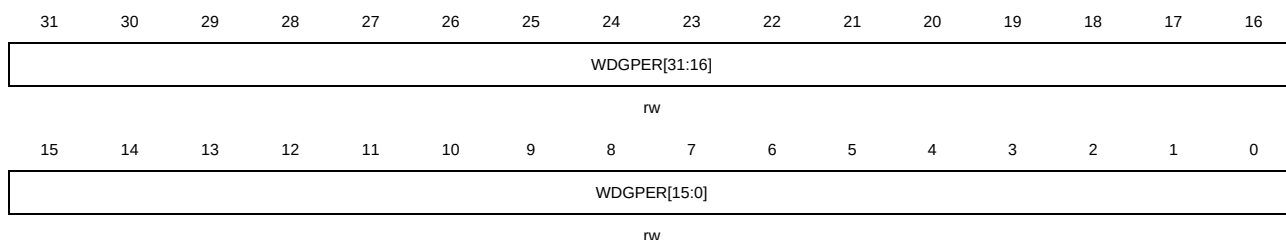
17:0 保留 必须保持复位值

看门狗计数器周期寄存器 (TIMERx_WDGPERR)

地址偏移: 0x94

复位值: 0xFFFF FFFF

该寄存器只能按字 (32 位) 访问。



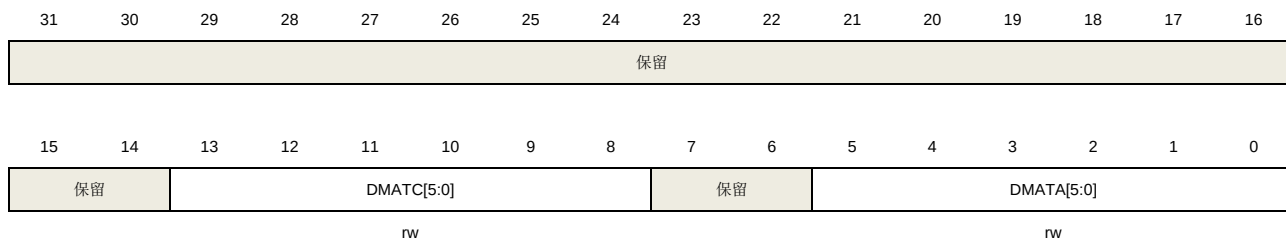
位/位域	名称	描述
31:0	WDGPERR[31:0]	看门狗计数器周期值 这些位用于配置两个看门狗的计数器周期。当看门狗计数器连续计数到该值时，计数器计数超时且中断标志位DECDSIF位置位。若DECDSIE=1，则相应的中断产生。 注意： 该寄存器位仅用于正交译码器信号断线检测功能 (DECDSDEN =1) 使能。

DMA 配置寄存器 (TIMERx_DMACFG)

地址偏移: 0xE0

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



位/位域	名称	描述
31:14	保留	必须保持复位值
13:8	DMATC [5:0]	DMA 传输计数 该位域定义了 DMA 访问 (读写) TIMERx_DMATB 寄存器的数量。 6'b000000: 传输1次 6'b000001: 传输2次 ...

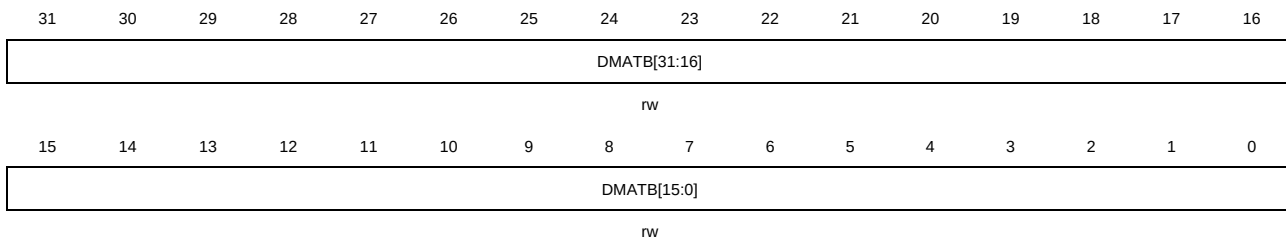
		6'b111111: 传输 63 次
7:6	保留	必须保持复位值
5:0	DMATA [5:0]	DMA 传输起始地址 该位域定义了 DMA 访问 TIMERx_DMATB 寄存器的第一个地址。当通过 TIMERx_DMA 第一次访问时，访问的就是该位域指定的地址。第二次访问 TIMERx_DMATB 时，将访问起始地址+0x4。 6'b000000: TIMERx_CTL0 6'b000001: TIMERx_CTL1 ... 6'b111111: TIMERx_DMACFG 总之：起始地址 = TIMERx_CTL0 + DMATA*4。

DMA 发送缓冲区寄存器（TIMERx_DMATB）

地址偏移：0xE4

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



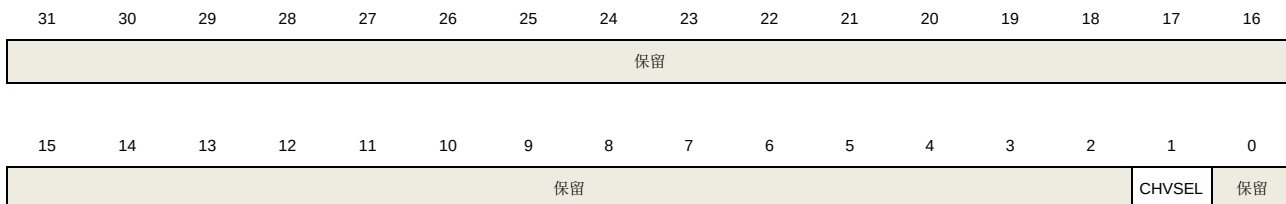
位/位域	名称	描述
31:0	DMATB [31:0]	DMA 发送缓冲 对这个寄存器的读或写，（起始地址+传输次数*4）地址范围内的寄存器会被访问 传输次数由硬件计算，范围为 0 到 DMATC。

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。



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

12.3. 通用定时器 L3 (TIMERx, x=15,16)

12.3.1. 简介

通用定时器 L3 (TIMER15/16) 是 3 通道定时器，支持输入捕获和输出比较。可以产生 PWM 信号控制电机和电源管理。通用定时器 L3 含有一个 16 位无符号计数器。

通用定时器 L3 是可编程的，可以被用来计数，其外部事件可以驱动其他定时器

通用定时器 L3 包含了一个死区时间插入模块，非常适合电机控制。

定时器和定时器之间是相互独立，但是他们可以被同步在一起形成一个更大的定时器，这些定时器的计数器一致地增加。

12.3.2. 主要特性

- 总通道数：3；
- 计数器宽度：16位；
- 时钟源可选：内部时钟，内部触发，外部输入；
- 多种计数模式：向上计数，向下计数和中央计数；
- 可编程的预分频器：16位，运行时可以被改变；
- 每个通道可配置：输入捕获模式，输出比较模式，可编程的PWM模式，单脉冲模式；
- 可编程的死区时间；
- 自动重装载功能；
- 可编程的计数器重复功能；
- 中止输入功能：BREAK；
- 中断输出和DMA请求：更新事件，比较/捕获事件和中止事件；
- 多个定时器的菊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-从管理。

12.3.3. 结构框图

[图 12-70. 通用定时器 L3 结构框图](#)提供了通用定时器 L3 的内部配置细节

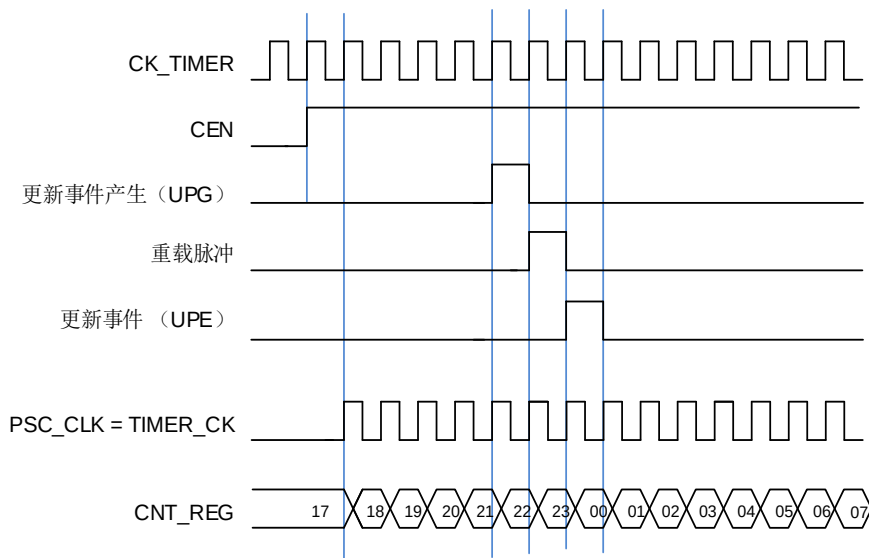
时钟源选择

- 当SYSCFG_TIMERxCFG(x=15,16)寄存器中的TSCFGy[4:0]=5'b00000(y=3..6,15), 定时器选择内部时钟源（连接到RCU模块的CK_TIMER）

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 SYSCFG_TIMERxCFG(x=15,16)寄存器中的 TSCFG6[4:0]位域设置为非零值，预分频器被其他时钟源驱动，具体在下文说明。当 TSCFGy[4:0] (y=3,4,5)被设置为非零值时，计数器预分频器时钟源由内部时钟 TIMER CK 驱动。

图 12-71. 内部时钟分频为 1 时正常模式下的控制电路



- TSCFG6[4:0] 设置为非零值（外部时钟模式0），定时器选择外部输入引脚作为时钟源

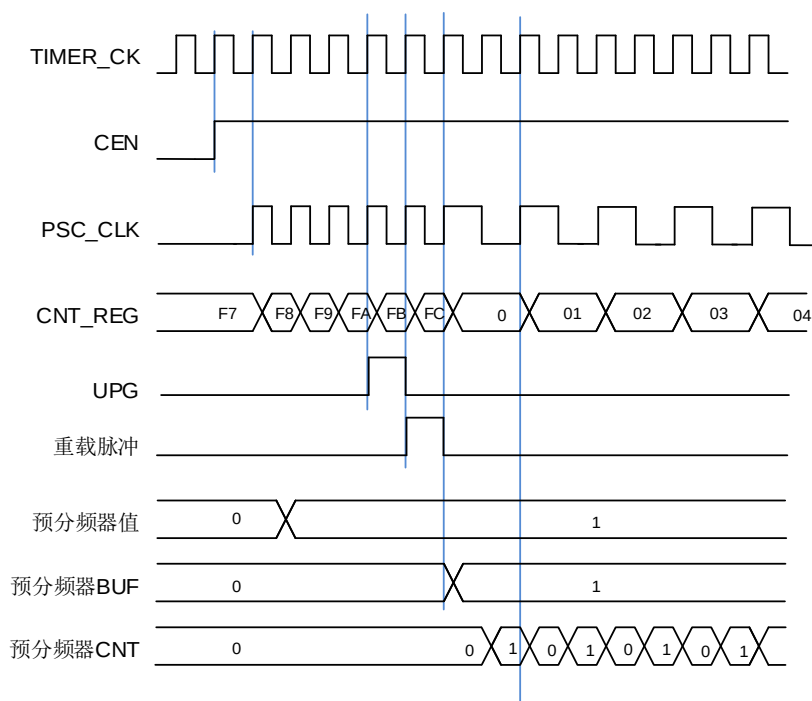
计数器预分频器可以在 **TIMERx_CIO/ TIMERx_CII** 引脚的每个上升沿或下降沿计数。这种模式可以通过设置 **TSCFG6[4:0]** 为 **0x5~0x7** 和 **0xB**。

计数器预分频器也可以在内部触发信号 **ITI0/1/2/3** 的上升沿计数。这种模式可以通过设置 **TSCFG6[4:0]** 为 **0x1~0x4**。

时钟预分频器

预分频器可以将定时器的时钟（**TIMER_CLK**）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 12-72. 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_CREP0+1)` 次上溢后产生更新事件，否则在每次上溢时都会产生更新事件。在向上计数模式中，`TIMERx_CTL0` 寄存器中的计数方向控制位 `DIR` 应该被设置成 0。

当通过 `TIMERx_SWEVG` 寄存器的 `UPG` 位置 1 来设置更新事件时，计数值会被清 0，并产生更新事件。

如果 `TIMERx_CTL0` 寄存器的 `UPDIS` 置 1，则禁止更新事件。

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

[图 12-73. 向上计数时序图, PSC=0/2](#) 和 [图 12-74. 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=0x99` 时，计数器在不同预分频因子下的行为。

图 12-73. 向上计数时序图, PSC=0/2

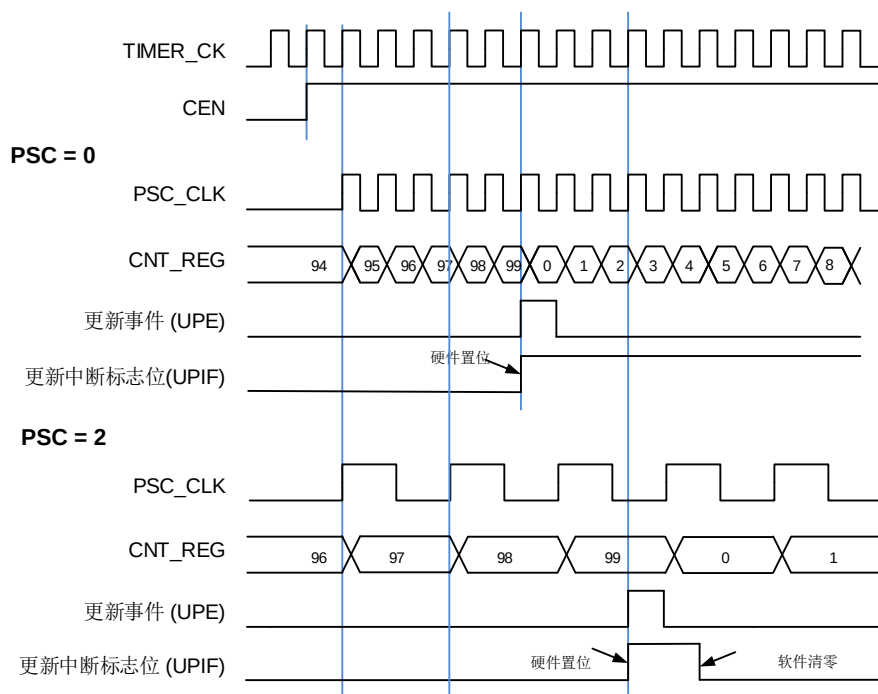
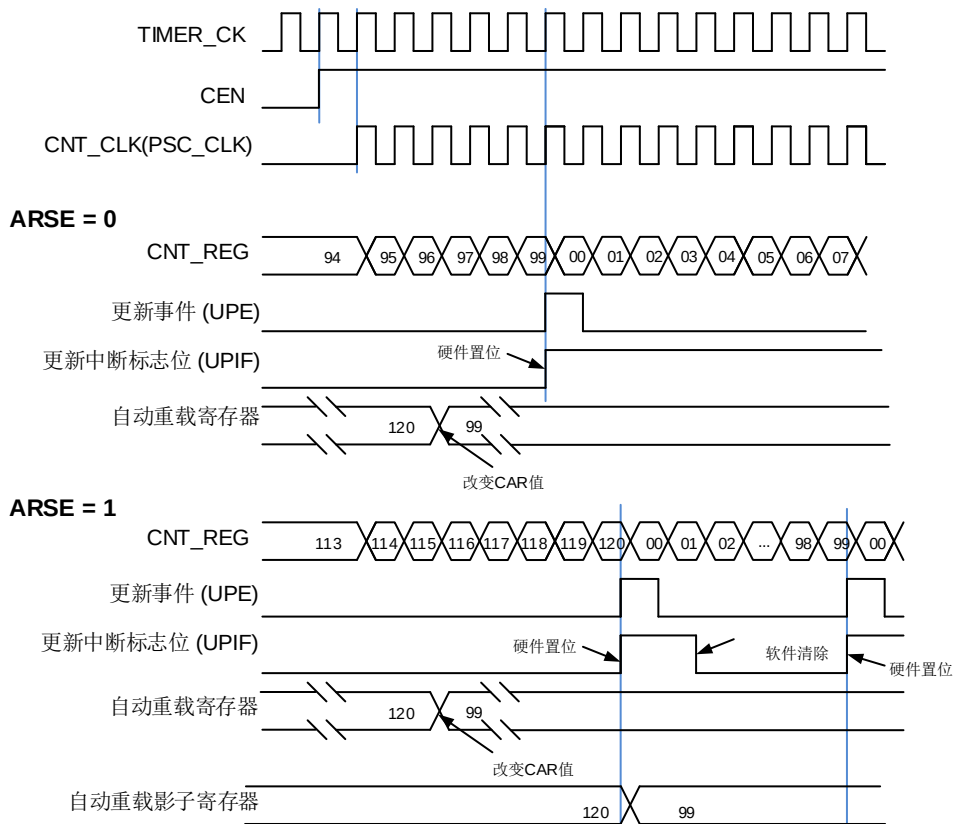


图 12-74. 向上计数时序图, 在运行时改变 **TIMERx_CAR** 寄存器的值



向下计数模式

在这种模式，计数器的计数方向是向下计数。计数器从自动加载值（定义在 `TIMERx_CAR` 寄存器中）向下连续计数到 0。一旦计数器计数到 0，计数器会重新从自动加载值开始计数并产生下溢事件。在向下计数模式中，`TIMERx_CTL0` 寄存器中的计数方向控制位 `DIR` 应该被设置成 1。

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

如果 `TIMERx_CTL0` 寄存器的 `UPDIS` 置 1，则禁止更新事件。

当发生更新事件时，所有的寄存器（自动重载寄存器，预分频寄存器）都将被更新。

[图 12-53. 向下计数时序图, PSC=0/2](#) 和 [图 12-54. 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=0x99` 时，计数器在不同时钟频率下的行为。

图 12-75. 向下计数时序图，PSC=0/2

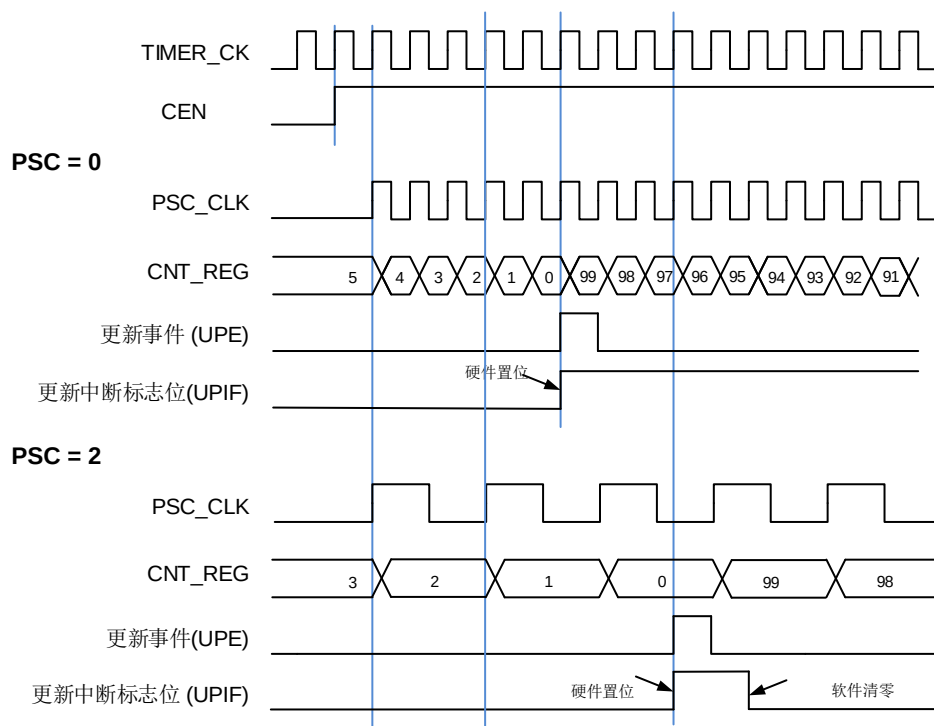
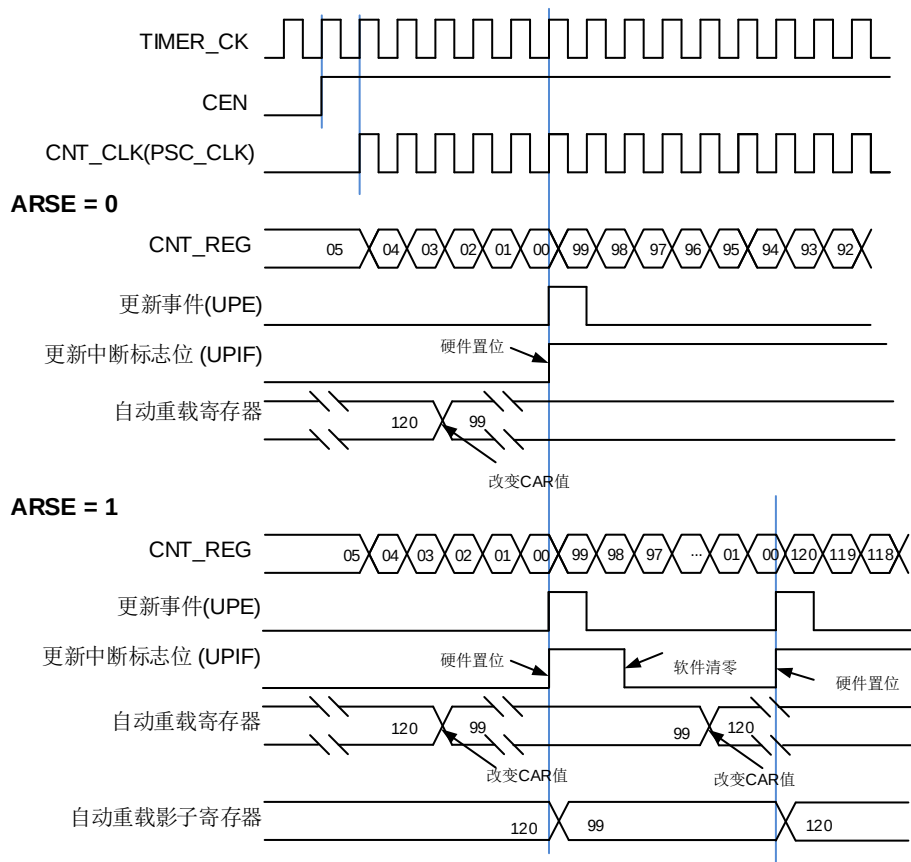


图 12-76. 向下计数时序图，在运行时改变 **TIMERx_CAR** 寄存器值

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_CAR-1**) 产生一个上溢事件；向下计数模式中，定时器模块在计数器计数到 1 时产生一个下溢事件。在中央计数模式中，**TIMERx_CTL0** 寄存器中的计数方向控制位 **DIR** 只读，表明了计数方向。计数方向被硬件自动更新。

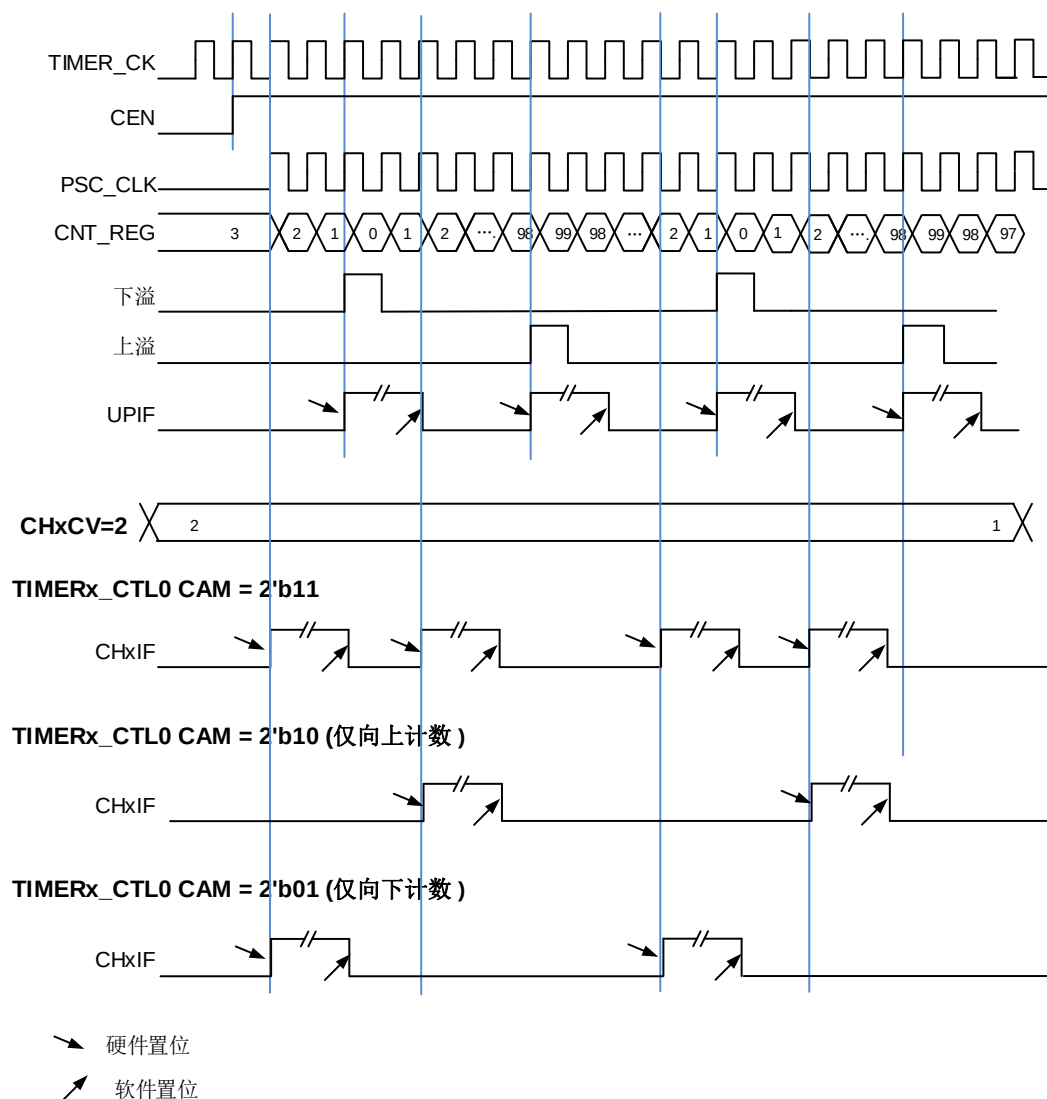
将 **TIMERx_SWEVG** 寄存器的 **UPG** 位置 1 可以初始化计数值为 0，并产生一个更新事件，而无需考虑计数器在中央模式下是向上计数还是向下计数。

上溢或者下溢时，**TIMERx_INTF** 寄存器中的 **UPIF** 位都会被置 1，然而 **CHxIF** 位置 1 与 **TIMERx_CTL0** 寄存器中 **CAM** 的值有关。具体细节参考 [图 12-55. 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 12-55. 中央计数模式计数器时序图](#) 给出了一些例子，当 **TIMERx_CAR=0x99**，**TIMERx_PSC=0x0** 时，计数器的行为

图 12-77. 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_CREP0** 寄存器的 **CREP0** 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_CREP0** 寄存器中 **CREP0** 的值并产生一个更新事件。

新写入的 **CREP0** 值将在下一次更新事件到来时生效。当 **CREP0** 的值为奇数，并且计数器在中央对齐模式下计数时，更新事件发生在上溢或下溢取决于写入的 **CREP0** 值何时生效。如果在写入奇数到 **CREP0** 寄存器后由软件生成更新事件 (**UPG** 位置 1)，则在下溢时产生更新事件。如果在写

入奇数到 CREP0 寄存器后下一个更新事件发生在上溢，此后将在上溢时产生更新事件。

图 12-78. 中央计数模式下计数器重复时序图

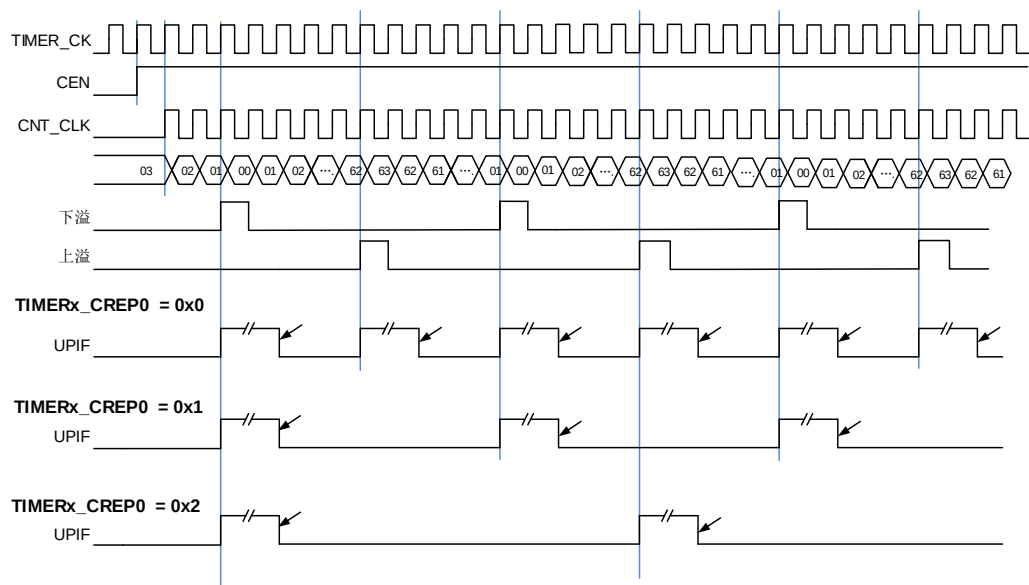


图 12-79. 在向上计数模式下计数器重复时序图

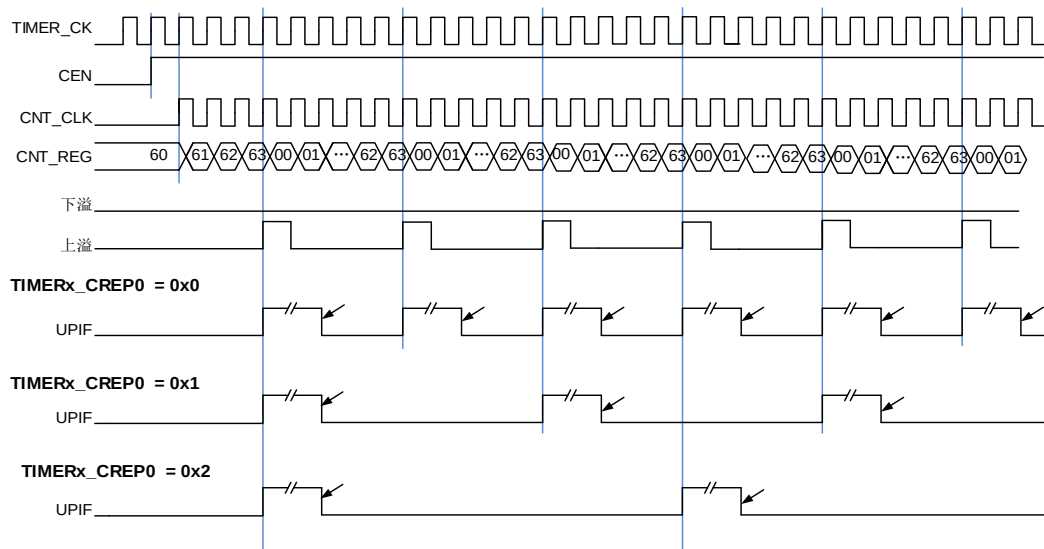
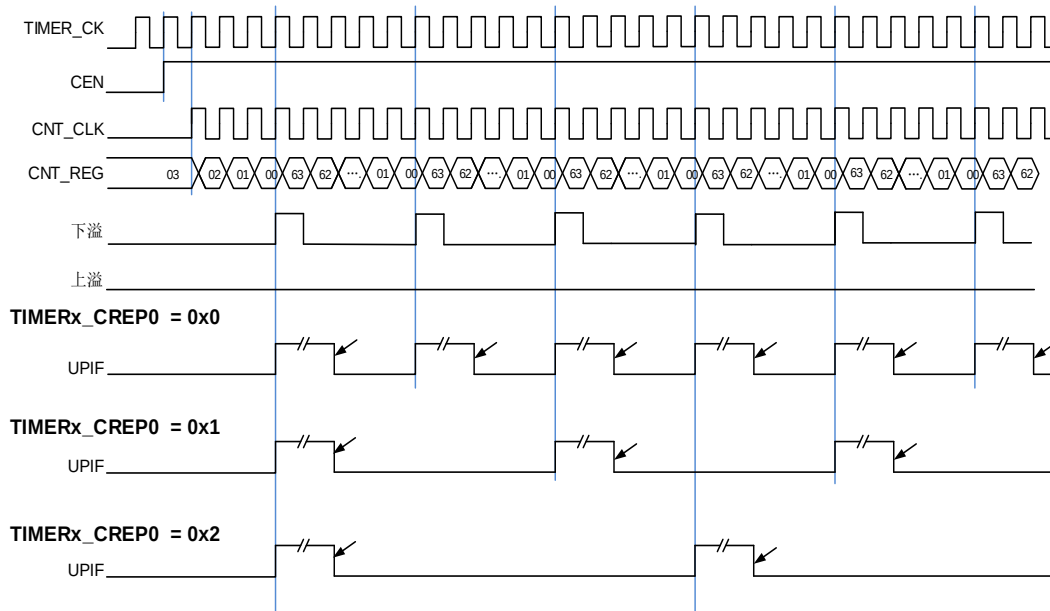


图 12-80. 在向下计数模式下计数器重复时序图



捕获/比较通道

通用定时器 L3 拥有 3 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。

当通道用于输入时，通道 x 和多模式通道 x 可独立进行输入捕获；当通道用于比较输出时，通道 x 和多模式通道 x 可输出独立和互补。

■ 输入捕获模式

当 $MCHxMSEL=2'b00$ （独立模式）时，通道 x 和多模式通道 x 才可以独立进行输入捕获。

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， $TIMERx\_CHxCV/$ $TIMERx\_MCH0CV$ ($x=0,1$) 寄存器会捕获计数器当前的值，同时 $CHxIF/$ $MCH0IF$ ($x=0,1$) 位置 1，如果 $CHxIE/$ $MCH0IE=1$ ($x=0,1$)，则产生相应的通道中断。

The diagram illustrates the internal architecture of the CH0 channel capture and compare module. Key components and their interconnections include:

- Inputs:** CIO (Channel Input), TIMERx_CC_INT (Timer Channel Input), CIOFE0 (Channel Input Edge 0), CIOF_ED (Channel Input Edge 0), C11FE0 (Channel Input Edge 1), MCIOFE0 (Master Channel Input Edge 0), ITS (Interrupt Status), CH0APPSC (Channel Auto-Prescaler Select), and CH0MS (Channel Mode Select).
- Internal Blocks:**
 - 同步器 (Synchronizer):** Receives CIO and TIMERx_CC_INT. It contains two D flip-flops and two comparators.
 - 滤波器 (Filter):** Receives the output of the synchronizer and filters the signal.
 - 边沿检测器 (Edge Detector):** Detects rising and falling edges of the filtered signal using D flip-flops and AND/OR gates.
 - 边沿选择器 & 反相器 (Edge Selector & Inverter):** Selects between rising and falling edges and inverts the signal. It outputs CIOFE0 and CIOF_ED.
 - 基于 CH0P & MCH0P (Based on CH0P & MCH0P):** A block that receives CIOFE0 and CIOF_ED and outputs C11FE0 and MCIOFE0.
 - 时钟预分频器 (Clock Prescaler):** Receives TIMERx_CC_INT and outputs to the counter.
 - 计数器 (Counter):** Counts the clock signal and outputs to the capture register.
 - 捕获寄存器 (CH0VAL):** Captures the value of the counter when a capture event occurs.
 - 预分频器 (Prescaler):** Receives the output of the counter and outputs to the CH0APPSC register.
 - CH0APPSC (Channel Auto-Prescaler Select):** A register that selects the auto-prescaler value.
 - CH0MS (Channel Mode Select):** A register that selects the channel mode.
 - CH0IF (Channel Input Flag) and CH0IE (Channel Input Enable):** Registers that control the capture and compare functionality.
- Outputs:** CH0_CC_I (Channel Capture/Compare Input), CH0IF (Channel Input Flag), CH0IE (Channel Input Enable), and the final output signal.

[illegible]

通道输入信号 **CIX/ MCIX** 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/ MCHxP**、**MCHxFP** 选择使用上升沿或者下降沿。配置 **CHxMS/ MCHxMS**，可以选择其他通道的输入信号或内部触发信号。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/ TIMERx MCHxCV** 存储计数器的值。

第一步：滤波器配置（TIMERx_CHCTL0 寄存器中 CHxCAPFLT 位和 TIMERx_MCHCTL0 寄存器中 CHxMCAPFLT）：

根据输入信号和请求信号的质量，配置相应的 CHxCAPFLT/ CHxMCAPFLT 位。

第二步：边沿选择（TIMERx_CHCTL2 寄存器中 CHxP 和 MCHxP 位，TIMERx_MCHCTL2 寄存器中 MCHxP[1:0]位域）：

配置 CHxP 和 MCHxP 位或 MCHxP 位域选择上升沿或者下降沿。

第三步：捕获源选择（TIMERx_CHCTL0 寄存器中 CHxMS、TIMERx_MCHCTL0 寄存器中 MCHxMS）：

一旦通过配置 CHxMS/ MCHxMS 选择输入捕获源，必须确保通道配置在输入模式（CHxMS!=0x000 或 MCHxMS!=0x000），而且 TIMERx_CHxCV/TIMERx_MCHxCV 寄存器不能再被写。

第四步：中断使能（TIMERx_DMAINTEN 寄存器中 CHxIE、CHxDEN 位和 MCHxIE、MCHxDEN 位）：

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/ MCHxEN）。

结果：当期望的输入信号发生时，TIMERx_CHxCV/ TIMERx_MCHxCV 被设置成当前计数器的值，CHxIF/ MCHxIF 位置 1。如果 CHxIF/ MCHxIF 位已经为 1，则 CHxOF/ MCHxOF 位置 1。根据 TIMERx_DMAINTEN 寄存器中 CHxIE、CHxDEN 位和 MCHxIE、MCHxDEN 位的配置，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 和 TIMERx_MCHx 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 3'b001，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 3'b010，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 12-83. 输出比较逻辑（当 MCHxMSEL = 2'00 时，x=0）](#)，[图 12-84. 输出比较逻辑（当 MCHxMSEL = 2'11 时，x=0）](#)和[图 12-85. 输出比较逻辑（x=1）](#)给出了通道的输出比较逻辑。

图 12-83. 输出比较逻辑（当 MCHxMSEL = 2'00 时，x=0）

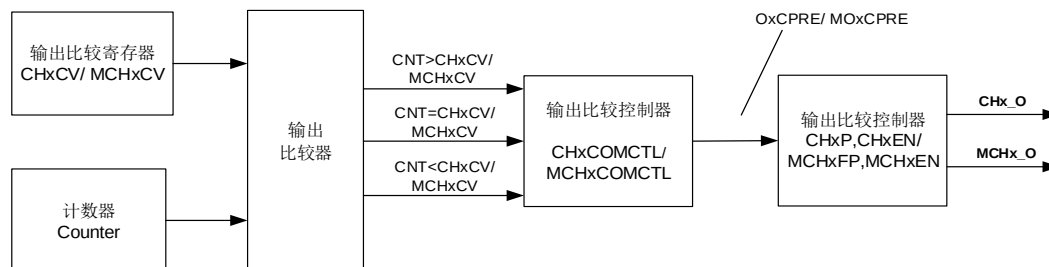
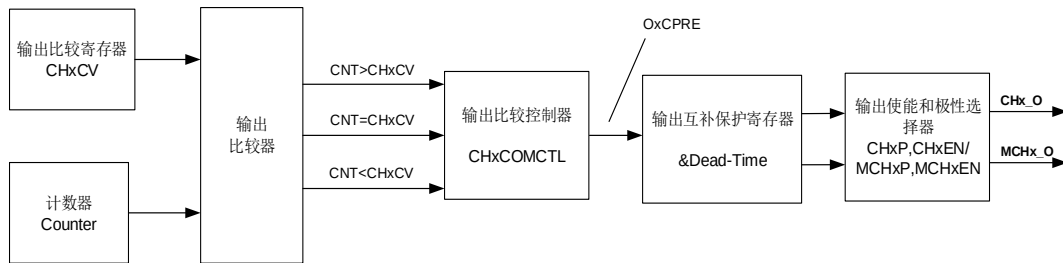
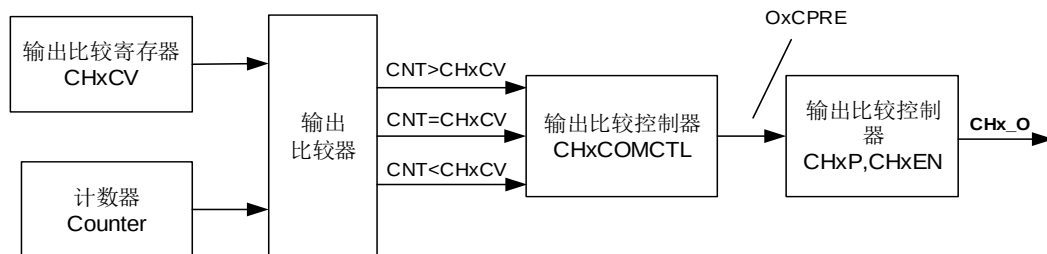


图 12-84. 输出比较逻辑（当 $MCHxMSEL = 2'11$ 时， $x=0$ ）图 12-85. 输出比较逻辑（ $x=1$ ）

通道输出信号 $CHx\_O/MCHx\_O$ 与 $OxCPRE/ MOxCPRE$ 信号（详情请见[通道输出参考信号](#)）的关系描述如下（ $OxCPRE/ MOxCPRE$ 信号高电平有效）：

- 当 $MCHxMSEL = 2'b00$ （ $TIMERx\_CTL2$ 寄存器中）， $MCHx\_O$ 输出与 CHx_O 输出相互独立。 CHx_O 输出电平取决于 $OxCPRE$ 信号、 $CHxP$ 位和 $CHxEN$ 位（详细内容参考 $TIMERx\_CHCTL2$ 寄存器）。 $MCHx\_O$ 输出电平取决于 $MOxCPRE$ 信号、 $MCHxP[1:0]$ 位和 $MCHxEN$ 位（详细内容参考 $TIMERx\_CHCTL2$ 和 $TIMERx\_MCHCTL2$ 寄存器）。请参考[图12-83. 输出比较逻辑（当 \$MCHxMSEL = 2'00\$ 时， \$x=0\$ ）](#)。
- 当 $MCHxMSEL = 2'b11$ ， $MCHx\_O$ 输出和 CHx_O 输出互补。 $CHx\_O/MCHx\_O$ 输出电平取决于 $OxCPRE$ 信号、 $CHxP/ MCHxP$ 位和 $CHxEN/ MCHxEN$ 位。请参考[图12-84. 输出比较逻辑（当 \$MCHxMSEL = 2'11\$ 时， \$x=0\$ ）](#)。

例如（ $MCHx\_O$ 输出与 CHx_O 输出相互独立）：

1) 当设置 $CHxP=0$ （ CHx_O 高电平有效，与 $OxCPRE$ 输出极性相同）、 $CHxEN=1$ （ 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 $MCHxP=1$ （ $MCHx\_O$ 低电平有效，与 $MOxCPRE$ 输出极性相反）、 $MCHxEN=1$ （ $MCHx\_O$ 输出使能）时：

若 $MOxCPRE$ 输出有效（高）电平，则 $MCHx\_O$ 输出有效（低）电平；

若 $MOxCPRE$ 输出无效（低）电平，则 $MCHx\_O$ 输出无效（高）电平。

当 $MCHxMSEL = 2'b11$ ， CHx_O 和 $MCHx\_O$ 同时输出时， CHx_O 和 $MCHx\_O$ 的具体输出情况

还与 `TIMERx_CCHP0` 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`/ `TIMERx_MCHxCV` 寄存器与计数器的值匹配时，根据 `CHxCOMCTL`/ `MCHxCOMCTL` 的配置，这个通道的输出可以被置高电平，被置低电平或者翻转。当计数器的值与 `TIMERx_CHxCV`/ `TIMERx_MCHxCV` 寄存器的值匹配时，`CHxIF`/ `MCHxIF` 位被置 1，如果 `CHxIE`/ `MCHxIE` = 1 则会产生中断，如果 `CHxDEN`/ `MCHxDEN` = 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`/ `MCHxCOMSEN` 位来配置输出比较影子寄存器；
- 设置 `CHxCOMCTL`/ `MCHxCOMCTL` 位来配置输出模式（置高电平/置低电平/翻转）；
- 设置 `CHxP`/ `MCHxP`/ `MCHxFP` 位来选择有效电平的极性；
- 设置 `CHxEN`/ `MCHxEN` 使能输出。

第三步：通过 `CHxIE`/ `MCHxIE`/ `CHxDEN`/ `MCHxDEN` 位配置中断/DMA 请求使能。

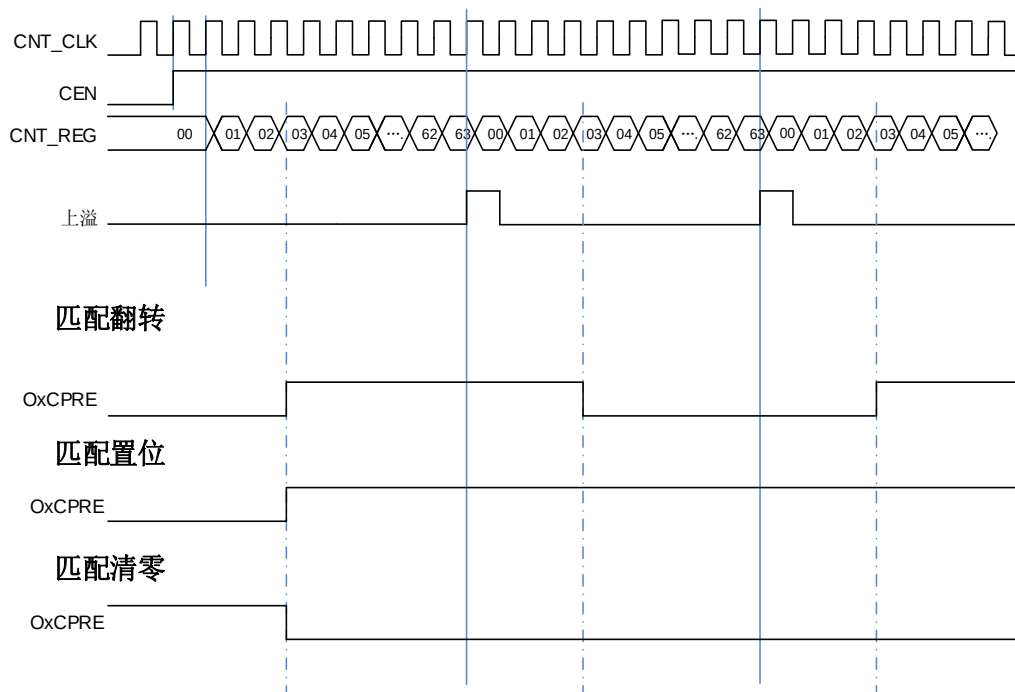
第四步：通过 `TIMERx_CAR` 寄存器和 `TIMERx_CHxCV` 寄存器配置输出比较时基：

`TIMERx_CHxCV`/ `TIMERx_MCHxCV` 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 12-86. 三种输出比较模式](#)显示了三种比较输出模式：翻转/置高电平/置低电平，`CAR=0x63`，`CHxVAL=0x3`。

图 12-86. 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/ MCHxCOMCTL 为 4'b0110，PWM 模式 1 是配置 CHxCOMCTL/ MCHxCOMCTL 为 4'b0111），通道根据 TIMERx_CAR 寄存器和 TIMERx_CHxCV/ TIMERx_MCHxCV 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/ TIMERx_MCHxCV 寄存器值决定。[图 12-15. 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/ TIMERx_MCHxCV 寄存器值）决定。[图 12-16. CAPWM 时序图](#)显示了 CAPWM 的输出波形和中断。

当计数器向上计数时，在 PWM0 模式下（CHxCOMCTL/ MCHxCOMCTL =4'b0110），如果 TIMERx_CHxCV/ TIMERx_MCHxCV 寄存器的值大于 TIMERx_CAR 寄存器的值，通道输出一一直为有效电平；PWM1 模式下（CHxCOMCTL/ MCHxCOMCTL=4'b0111），如果 TIMERx_CHxCV/ TIMERx_MCHxCV 寄存器的值大于 TIMERx_CAR 寄存器的值，通道输出一一直为无效电平。

图 12-87. EAPWM 时序图

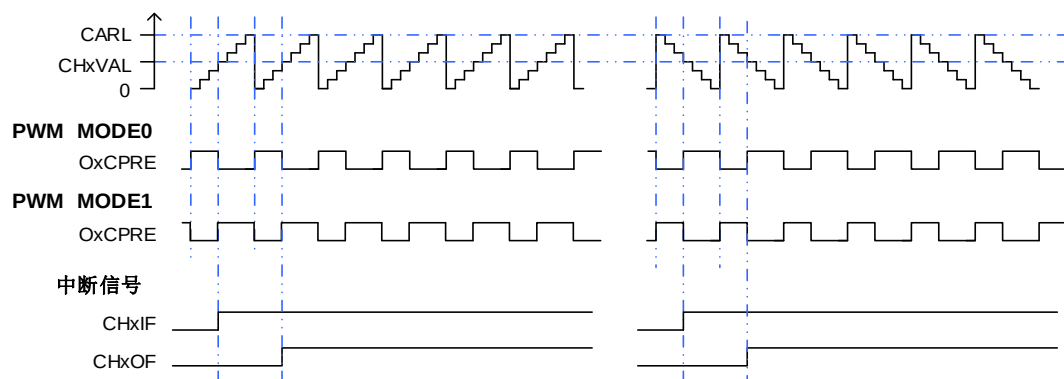
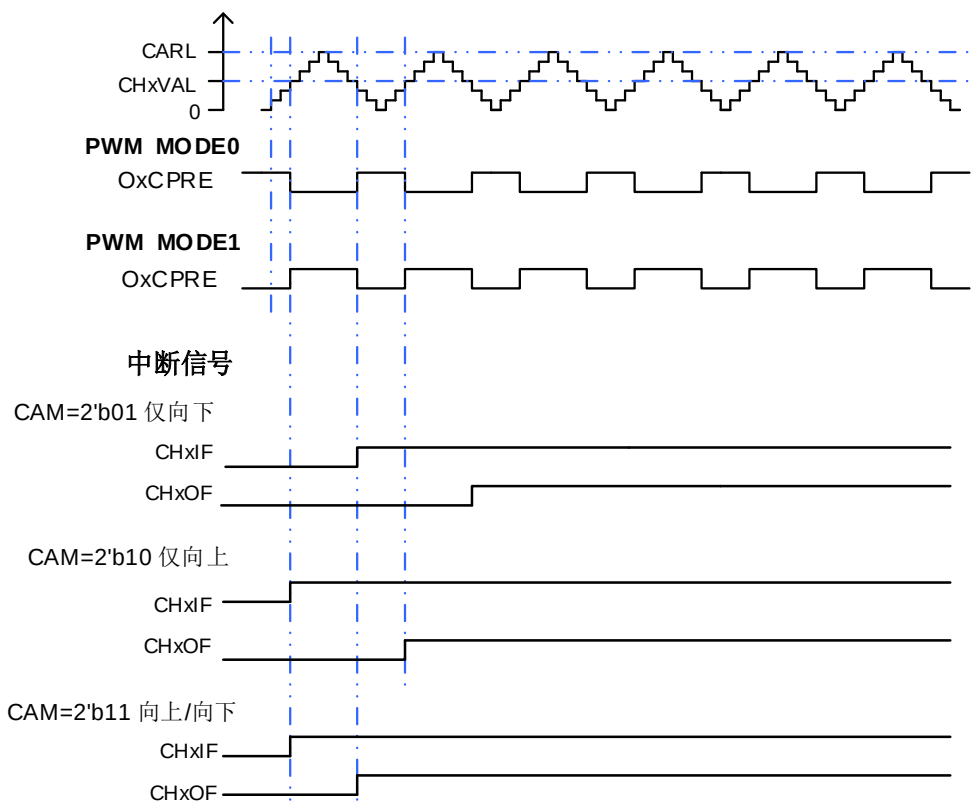


图 12-88. CAPWM 时序图



复合 PWM 模式

在复合 PWM 模式中 ($CHxCPWMEN = 1'b1$, $CHxMS[2:0] = 3'b000$ 和 $CHxCOMCTL = 4'b0110$ 、 $4'b0111$)，通道 x ($x=0..1$) 上的 PWM 输出信号由 $CHxVAL$ 和 $CHxCOMVAL\_ADD$ 位确定。

如果 $CHxCOMCTL = 4'b0110$ (PWM 模式 0) 且 $DIR = 1'b0$ (向上计数模式)，或者 $CHxCOMCTL = 4'b0111$ (PWM 模式 1) 且 $DIR = 1'b1$ (向下计数模式)，当计数器和 $CHxVAL$ 的值相匹配时通道 x 输出强制为低。当计数器与 $CHxCOMVAL\_ADD$ 的值相匹配时，通道 x 输出强制为高。

如果 $CHxCOMCTL = 4'b0111$ (PWM 模式 1) 且 $DIR = 1'b0$ (向上计数模式), 或者 $CHxCOMCTL = 4'b0110$ (PWM 模式 0) 且 $DIR = 1'b1$ (向下计数模式), 当计数器和 $CHxVAL$ 的值相匹配时通道 x 输出强制为高。当计数器与 $CHxCOMVAL\_ADD$ 的值相匹配时, 通道 x 输出强制为低。

当 $CHxVAL$ 或 $CHxCOMVAL\_ADD = 0 / CARL$, 通道输出被进行了特殊处理, 通过 $TIMERx\_CHCTL2$ 寄存器的 $CHxPERFOREN$ 位置位, 强制 $OxCPRE$ 输出高电平或低电平 (根据所选复合 PWM 模式确定)。

PWM 的周期取决于 $(CARL + 0x0001)$, PWM 脉冲宽度可以下 [表 12-12. 复合 PWM 脉冲宽度](#) 计算。

表 12-12. 复合 PWM 脉冲宽度

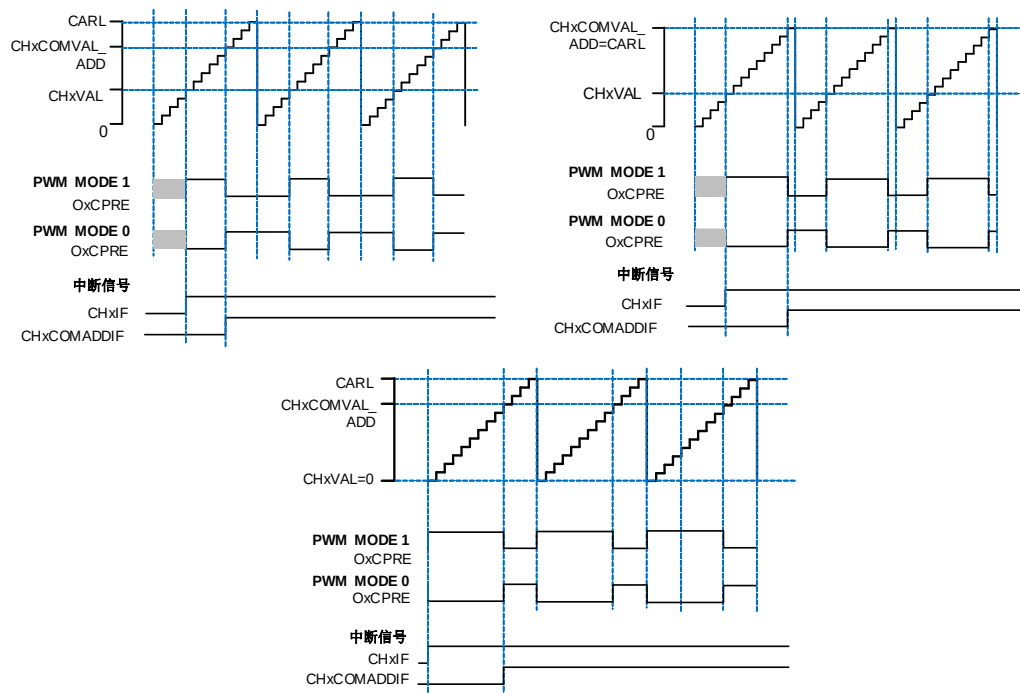
条件	模式	PWM 脉冲宽度
$CHxVAL < CHxCOMVAL\_ADD \leq CARL$	PWM 模式 0	$(CARL + 0x0001) + (CHxVAL - CHxCOMVAL\_ADD)$
	PWM 模式 1	$(CHxCOMVAL\_ADD - CHxVAL)$
$CHxCOMVAL\_ADD < CHxVAL \leq CARL$	PWM 模式 0	$(CHxVAL - CHxCOMVAL\_ADD)$
	PWM 模式 1	$(CARL + 0x0001) + (CHxCOMVAL\_ADD - CHxVAL)$
$(CHxVAL = CHxCOMVAL\_ADD \leq CARL)$ 或 $(CHxVAL > CARL > CHxCOMVAL\_ADD)$	PWM 模式 0 (向上计数) 或 PWM 模式 1 (向下计数)	100%
	PWM 模式 0 (向下计数) 或 PWM 模式 1 (向上计数)	0%
	PWM 模式 0 (向上计数) 或 PWM 模式 1 (向下计数)	0%
	PWM 模式 0 (向下计数) 或 PWM 模式 1 (向上计数)	100%
$CHxCOMVAL\_ADD > CARL > CHxVAL$	PWM 模式 0 (向上计数) 或 PWM 模式 1 (向下计数)	0%
	PWM 模式 0 (向下计数) 或 PWM 模式 1 (向上计数)	100%
$(CHxVAL > CARL)$ 且 $(CHxCOMVAL\_ADD > CARL)$	-	CHx_O 输出保持

当计数器计数到 $CHxVAL$, $CHxIF$ 位置 1 且如果 $CHxIE=1$ 通道 x 产生中断, 如果 $CHxDEN=1$, 则产生 DMA 请求。当计数器计数到 $CHxCOMVAL\_ADD$ 时, $CHxCOMADDIF$ 位置 1 (该中断标志位只在复合 PWM 模式有效, $CHxCPWMEN=1$), 如果 $CHxCOMADDIE = 1$ 通道 x 附加比较中断产生 (只有中断产生, 没有 DMA 请求响应)。

根据 $CHxVAL$, $CHxCOMVAL\_ADD$ 和 $CARL$ 之间的关系, 可以分为四种情况:

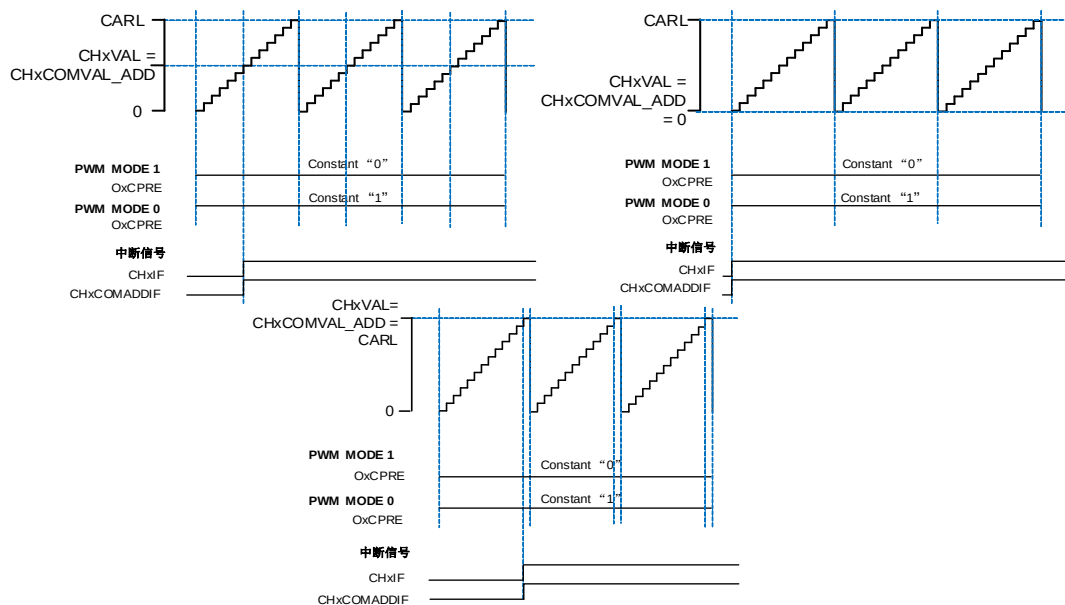
- 1) $CHxVAL < CHxCOMVAL\_ADD$, $CHxVAL$ 和 $CHxCOMVAL\_ADD$ 值介于 0 和 $CARL$ 之间。

图 12-89. 通道 x 输出 PWM (CHxVAL < CHxCOMVAL_ADD)



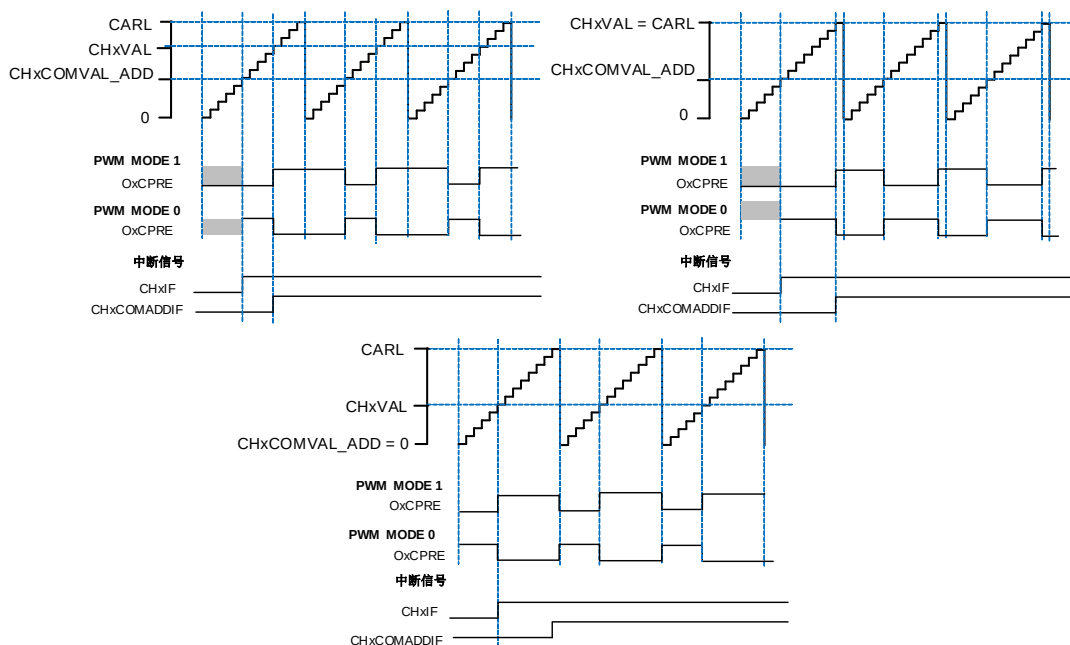
2) $CHxVAL = CHxCOMVAL\_ADD$, $CHxVAL$ 和 $CHxCOMVAL\_ADD$ 值介于0和 $CARL$ 之间。

图 12-90. 通道 x 输出 PWM (CHxVAL = CHxCOMVAL_ADD)



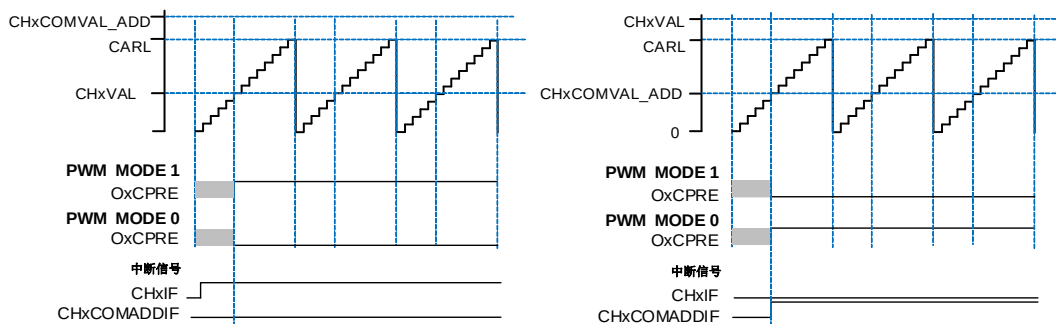
3) $CHxVAL > CHxCOMVAL\_ADD$, $CHxVAL$ 和 $CHxCOMVAL\_ADD$ 值介于0和 $CARL$ 之间。

图 12-91. 通道 x 输出 PWM (CHxVAL > CHxCOMVAL_ADD)



4) $CHxVAL$ 或 $CHxCOMVAL\_ADD$ 值大于 $CARL$ 。

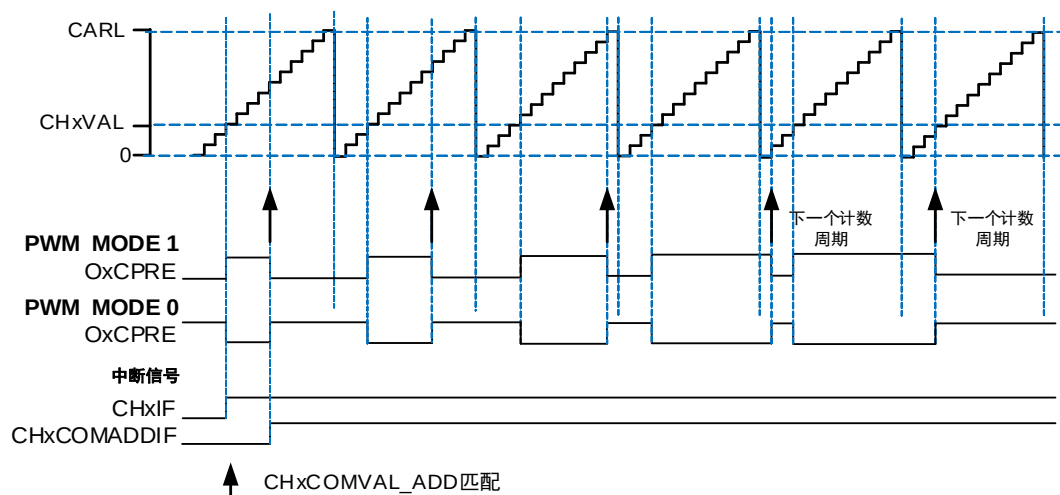
图 12-92. 通道 x 输出 PWM ($CHxVAL$ 或 $CHxCOMVAL\_ADD > CARL$)



复合 PWM 模式支持不修改周期只修改占空比的 PWM 信号的生成。[图 12-93. 通道 x 输出 PWM 占空比随着 CHxCOMVAL_ADD 值而改变](#)显示 PWM 输出和中断波形。

在某些情况下， $CHxCOMVAL\_ADD$ 的匹配事件可以发生在下一个计数周期（ $CHxCOMVAL\_ADD$ 值在计数器到达 $CHxVAL$ 值之后被写入，且 $CHxCOMVAL\_ADD$ 值小于或者等于 $CHxVAL$ 值）。

图 12-93. 通道 x 输出 PWM 占空比随着 CHxCOMVAL_ADD 值而改变



如果多个通道配置为复合 PWM 模式，可以为每对通道 x 的匹配边沿设定一个偏移量（相对于其它通道）。这种特性在产生照明 PWM 控制信号时非常有用，因为在这种情况下，希望彼此边缘不重合，以消除噪声的产生。CHxVAL 寄存器值是 PWM 脉冲相对于计数器周期开始的偏移。

输出匹配脉冲选择

当发生匹配事件时，CHx_O (x=0, 1) 的输出由 CHxCOMCTL[3:0] (x=0, 1) 位设置，通过配置 CHxOMPSEL[1:0] (x=0, 1) 位，可选择 CHx_O (x=0, 1) 的输出信号正常或者脉冲。

当匹配事件发生时，CHxOMPSEL[1:0] (x=0, 1) 用于选择 OxCPRE 信号输出（驱动 CHx_O）：

- CHxOMPSEL = 2'b00，OxCPRE信号根据CHxCOMCTL[3:0]位的配置正常输出；
- CHxOMPSEL = 2'b01，只有在计数器向上计数，发生匹配事件时，OxCPRE信号输出一个脉冲，且脉冲宽度为一个CK_TIMER时钟周期；
- CHxOMPSEL = 2'b10，只有在计数器向下计数，发生匹配事件时，OxCPRE信号输出一个脉冲，且脉冲宽度为一个CK_TIMER时钟周期；
- CHxOMPSEL = 2'b11，无论计数器向上计数还是向下计数，发生匹配事件时，OxCPRE信号输出一个脉冲，且脉冲宽度为一个CK_TIMER时钟周期；

图 12-94. 边沿对齐模式下 CHx_O 输出脉冲 (CHxOMPSEL≠2'b00)

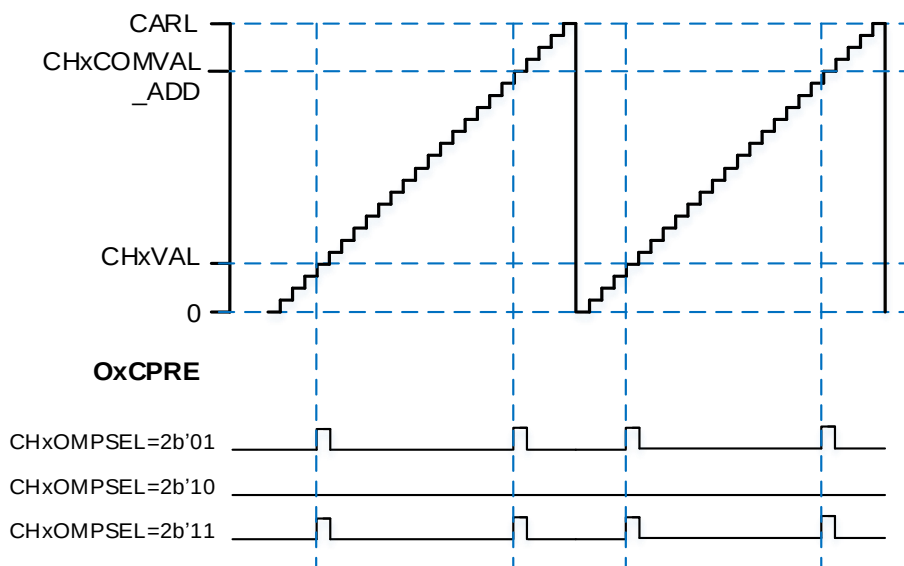
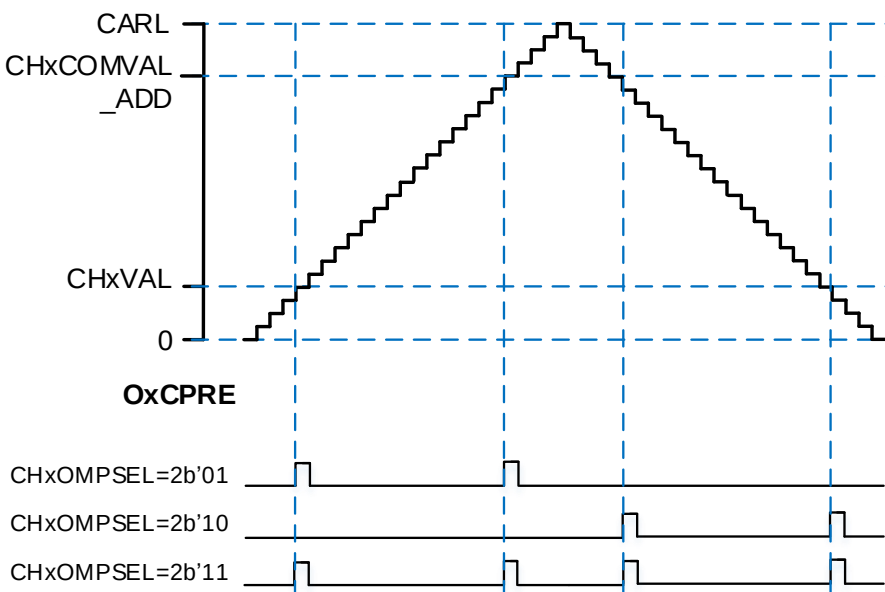


图 12-95. 中央对齐模式下 CHx_O 输出脉冲 (CHxOMPSEL≠2'b00)



通道输出参考信号

如[图 12-83. 输出比较逻辑 \(当 MCHxMSEL = 2'00 时, x=0\)](#), [图 12-84. 输出比较逻辑 \(当 MCHxMSEL = 2'11 时, x=0\)](#)和[图 12-85. 输出比较逻辑 \(x=1\)](#)所示, 当 TIMERx 用于输出匹配比较模式下, 在通道输出信号之前将产生一个中间信号, 即 OxCPRE 或 MOxCPRE 信号 (通道 x 或多模式通道 x 参考信号)。

OxCPRE 和 MOxCPRE 信号有若干类型的输出功能, 通过配置 CHxCOMCTL 位定义 OxCPRE 信号类型, 通过配置 MCHxCOMCTL 位定义 MOxCPRE 信号类型。

下面以 OxCPRE 为例进行说明，设置 CHxCOMCTL=0x00 可以保持原始电平；设置 CHxCOMCTL=0x01 可以将 OxCPRE 信号设置为高电平；设置 CHxCOMCTL=0x02 可以将 OxCPRE 信号设置为低电平；设置 CHxCOMCTL=0x03，在计数器值和 TIMERx_CHxCV 寄存器的值匹配时，可以翻转输出信号。

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

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

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 MCHx_O 的输出具有两种情况：

- MCHxMSEL=2'b00: MCHx_O 输出独立于 CHx_O 输出。
- MCHxMSEL=2'b11: MCHx_O 输出与 CHx_O 输出互补，且 MCHx_O 的输出不由 CHxMOMCTL 位配置。

当 CHx_O 和 MCHx_O 输出互补时，这两个信号不能同时有效。TIMERx 的 1 对通道具有此功能。互补信号 CHx_O 和 MCHx_O 是由一组参数来决定：TIMERx_CHCTL2 寄存器中的 CHxEN 和 MCHxEN 位，TIMERx_CCHP0 寄存器中和 TIMERx_CTL1 寄存器中的 POEN、ROS、IOS、ISOx 和 ISOxN 位。输出极性由 TIMERx_CHCTL2 寄存器中的 CHxP 和 MCHxP 位来决定。

当 CHx_O 和 MCHx_O 的输出互补时，有三种输出情况：输出使能、输出关闭状态和输出禁能，具体情况可参考 [表 12-13. 由参数控制的互补输出表 \(MCHxMSEL=2'b11\)](#)。

表 12-13. 由参数控制的互补输出表 (MCHxMSEL=2'b11)

互补参数					输出状态	
POEN	POEN	POEN	POEN	POEN	CHx_O	MCHx_O
0	0/1	0	0	0	CHx_O / MCHx_O = LOW CHx_O / MCHx_O 输出禁能 ⁽¹⁾	
				1	CHx_O / MCHx_O 输出关闭状态 ⁽²⁾ ： 通道先输出无效电平：CHx_O = CHxP，MCHx_O = MCHxP；如果死区产生时钟未失效，在死区时间之后： CHx_O = ISOx，MCHx_O = ISOxN ⁽³⁾	
			1	0		
				1		
		1	x	x	CHx_O / MCHx_O 输出关闭状态： 通道先输出无效电平：CHx_O = CHxP，MCHx_O = MCHxP；如果死区产生时钟未失效，在死区时间之后： CHx_O = ISOx，MCHx_O = ISOxN	

互补参数					输出状态	
POEN	POEN	POEN	POEN	POEN	CHx_O	MCHx_O
1	0	0/1	0	0	CHx_O/MCHx_O = LOW CHx_O/MCHx_O输出禁能	
				1	CHx_O = LOW CHx_O输出禁能	MCHx_O=OxCPRE ⊕ (4)MCHxP MCHx_O输出使能
			1	0	CHx_O=OxCPRE ⊕ CHxP CHx_O输出使能	MCHx_O = LOW MCHx_O输出禁能
				1	CHx_O=OxCPRE ⊕ CHxP CHx_O输出使能	MCHx_O=(!OxCPRE)(5) ⊕ MCHxP MCHx_O输出使能
	1		0	0	CHx_O = CHxP CHx_O输出关闭状态	MCHx_O = MCHxP MCHx_O输出关闭状态
				1	CHx_O = CHxP CHx_O输出关闭状态	MCHx_O=OxCPRE ⊕ MCHxP MCHx_O输出使能
			1	0	CHx_O=OxCPRE ⊕ CHxP CHx_O输出使能	MCHx_O = MCHxP MCHx_O输出关闭状态
				1	CHx_O=OxCPRE ⊕ CHxP CHx_O输出使能	MCHx_O=(!OxCPRE) ⊕ MCHxP MCHx_O输出使能

注意：

- (1) 输出禁能：CHx_O / CHx_ON 输出与对应引脚断开，对应引脚电平受 GPIO 上下拉配置控制，无上下拉时为悬空高阻态；
- (2) 输出关闭状态：CHx_O / CHx_ON 输出无效电平（CHx_O = 0 $\oplus$ CHxP = CHxP）；
- (3) 详情见中止模式章节。
- (4) $\oplus$ ：异或操作；
- (5) (!OxCPRE)：OxCPRE 信号的互补信号。

死区时间插入

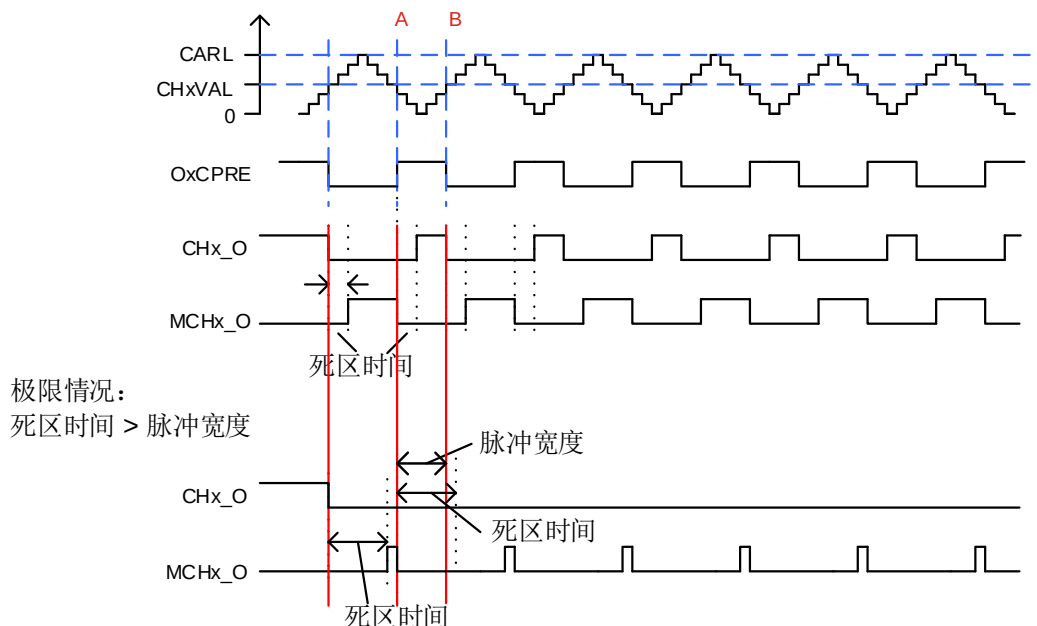
设置 MCHxMSEL=2'b11，CHxEN 和 MCHxEN 为 1'b1，同时设置 POEN=1，就可以使能死区插入功能。DTCFG 位域定义了死区时间，死区时间对所有通道有效。死区时间设置的细节请参考 [互补通道保护寄存器 0 \(TIMERx CCHP0\)](#)。

死区时间的插入，确保了通道互补的两路信号不会同时有效。

在 PWM0 模式，当通道 x 匹配发生时（TIMERx 计数器=TIMERx_CHxCV），OxCPRE 翻转。在 [图 12-96. 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 MCHx_O 信号立刻变为低电平。同样，在 B 点，计数器再次匹配（TIMERx 计数器=TIMERx_CHxCV），OxCPRE 信号被清 0，CHx_O 信号被立即清零，MCHx_O 信号在死区时间内仍然是低电平，在死区时间过后才变为高电平。有时会有一些死角事件发生，例如：如果

死区延时大于或者等于 CHx_O 信号的占空比, CHx_O 信号一直为无效值, 如 [图 12-96. 带死区时间的互补输出](#) 所示。

图 12-96. 带死区时间的互补输出



中止功能

当 MCHxMSEL = 2'b11 (MCHx_O 的输出不使用 CHxMOMCTL 位配置) 时, MCHx_O 输出与 CHx_O 输出互补。在这种情况下, CHx_O 和 MCHx_O 信号不能同时设置为有效电平。

通用 L3 定时器具有 BREAK 中止功能。可以通过将 TIMERx_CCHP0 寄存器中的 BRKEN 位置 1 来使能。中止输入极性由 TIMERx_CCHP0 寄存器中的 BRKP 位配置, 电平有效。

使用 BREAK 功能时, CHx_O 和 MCHx_O 信号的输出电平由以下位控制: TIMERx_CCHP0 寄存器的 POEN、IOS 和 ROS 位, TIMERx_CTL1 寄存器的 ISOx 和 ISOxN 位。

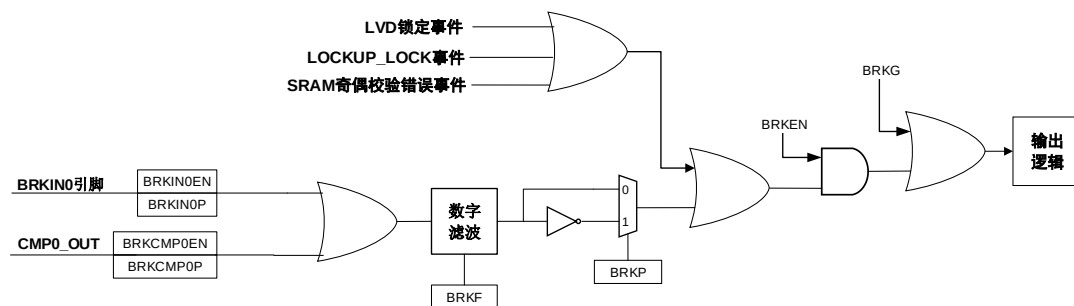
中止事件是所有源逻辑或运算的结果。中止功能可以处理三种类型的事件源:

- 外部信号源: 来自 BRKIN0 输入;
- 系统源: LVD 锁定事件, Cortex®-M33 锁定事件或 SRAM 奇偶校验错误事件;
- 片上外设源: 比较器输出。

BREAK 中止事件也可以由软件置位 TIMERx_SWEVG 寄存器中的 BRKG 位产生。

如 [图 12-97. BREAK 中止功能逻辑图](#) 所示, BRKIN0 可以从 TRIGSEL 模块选择 GPIO 引脚, 具体可参考 [TIMER15 BRKIN 触发选择寄存器 \(TRIGSEL_TIMER15BRKIN\)](#) 和 [TIMER16 BRKIN 触发选择寄存器 \(TRIGSEL_TIMER16BRKIN\)](#)。

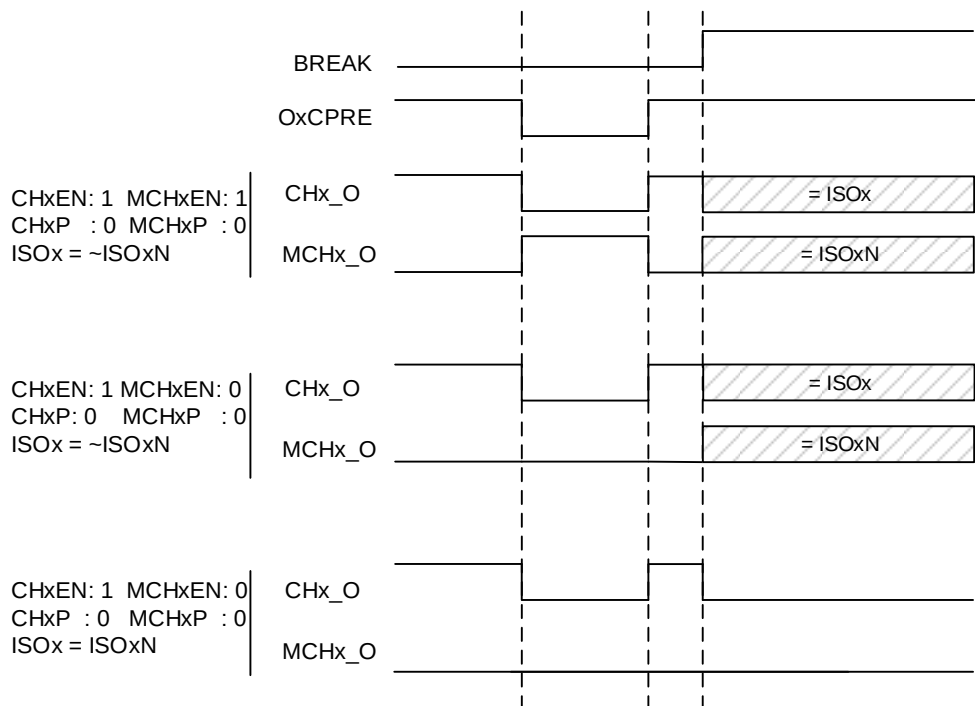
图 12-97. BREAK 中止功能逻辑图



BREAK 可用于处理系统源、片上外设和外部输入信号源的故障，当发生 BREAK 中止事件时，输出强制为无效电平，或在死区持续时间之后，输出将以预定的电平（有效或无效）强制输出。

当 $MCHxMSEL = 2'b11$ 且发生 BREAK 中止事件时，POEN 位被异步清除，一旦 POEN 位为 0，CHx_O 和 MCHx_O 的输出由 $TIMERx\_CTL1$ 寄存器中的 ISOx 位和 ISOxN 位确定。如果 $IOS=0$ ，定时器释放输出使能，否则输出使能仍然为高。当 $IOS=1$ 时，通道输出情况如 [图 12-98. 通道响应 BREAK 中止输入（高电平有效）时，输出信号的行为（IOS=1）](#) 所示，首先通道互补输出为复位状态，然后死区时间发生器重新被激活，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。

图 12-98. 通道响应 BREAK 中止输入（高电平有效）时，输出信号的行为（IOS=1）

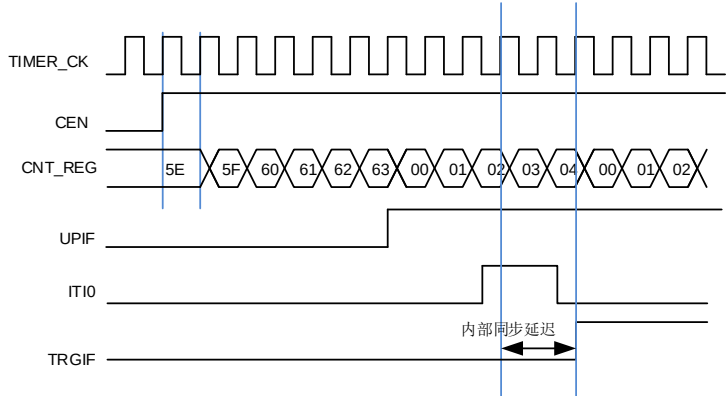


发生中止事件时， $TIMERx\_INTF$ 寄存器的 BRKIF 位被置 1。如果 $BRKIE=1$ ，中断产生。

主-从管理

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等，可以通过设置 SYSCFG_TIMERxCFG(x=15,16)寄存器中的 TSCFGy[4:0] (y=3..6)位域来确定，具体的输入触发源可以通过 TSCFGy[4:0] (y=3..6)位域的值来选择。

表 12-14. 从模式例子列表

	模式选择	触发源选择	极性选择	滤波和预分频
列举	TSCFGy[4:0] y=3: 复位模式 y=4: 暂停模式 y=5: 事件模式 y=6: 外部时钟模式0	TSCFGy[4:0] 00000: Mode disable 00001: ITI0 00010: ITI1 00011: ITI2 00100: ITI3 00101: CI0F_ED 00110: CI0FE0 00111: CI1FE1 10010: MCI0FEM0	如果触发源是CIxFEx (x=0, 1) 或者 MCixFEMx (x=0)，配置CHxP、MCHxP和 MCHxFP来选择极性和反相。	触发源ITIx，滤波和预分频不可用 触发源CIx/MCIx，配置CHxCAPFLT/MCHxCAPFLT设置滤波，分频不可用
例1	复位模式 当触发输入上升沿，计数器清零重启	TSCFG3[4:0] 5'b00001，选择ITI0为触发源	触发源是ITI0，极性选择不可用	触发源是ITI0，滤波和预分频不可用
	图 12-99. 复位模式下的控制电路 			
例2	暂停模式 当触发输入为低的时候，计数器暂停计数	TSCFG4[4:0] =5'b00110，选择CI0FE0为触发源	TI0S=0（非异或） [MCH0P=0,CH0P=0] CI0FE0不反相，在上升沿捕获	在这个例子中滤波被旁路

	模式选择	触发源选择	极性选择	滤波和预分频
	图 12-100. 暂停模式下的控制电路			
例3	事件模式 触发输入的上升沿计数器开始计数	TSCFG5[4:0] =5'b01000, 选择 ETIFP为触发源	ETP = 0没有极性改变	ETPSC = 1, 2分频 ETFC = 0, 无滤波
	图 12-101. 事件模式下的控制电路			

单脉冲模式

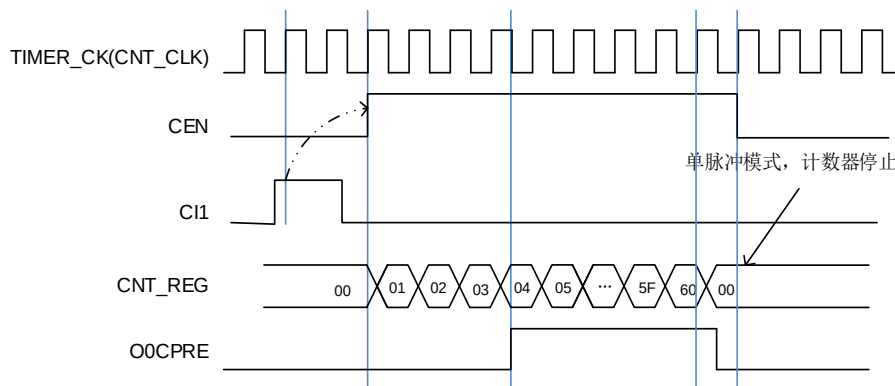
单脉冲模式与重复模式是相反的,设置 `TIMERx_CTL0` 寄存器的 `SPM` 位置 1,可启用单脉冲模式。当 `SPM` 置 1,计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波,可以通过设置 `CHxCOMCTL/ MCHxCOMCTL` 配置 `TIMERx` 为 PWM 模式或者比较模式。

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。单脉冲模式下,触发上升沿产生之后, `OxCPRE/ MOxCPRE` 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。

单脉冲模式也同样适用于复合 PWM 模式（CHxCPWMEN = 1'b1 和 CHxMS[2:0] = 3'b000）。

图 12-102. 单脉冲模式，TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60



定时器互连

参考 [高级定时器 \(TIMERx, x=0, 7\) 定时器互连](#)。

计数器同步、初始方向和值刷新

参考 [高级定时器 \(TIMERx, x=0, 7\) 计数器同步、初始方向和值刷新](#)。

定时器 DMA 模式

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

如果再来 1 次 DMA 请求事件，TIMERx 将会重复上面的过程。

定时器调试模式

当 Cortex®-M33 内核停止，DBG_CTL1 寄存器中的 TIMERx_HOLD 配置位被置 1，定时器计数器停止。

12.3.5. TIMERx 寄存器(x=15/16)

TIMER15 基地址: 0x4001 5000

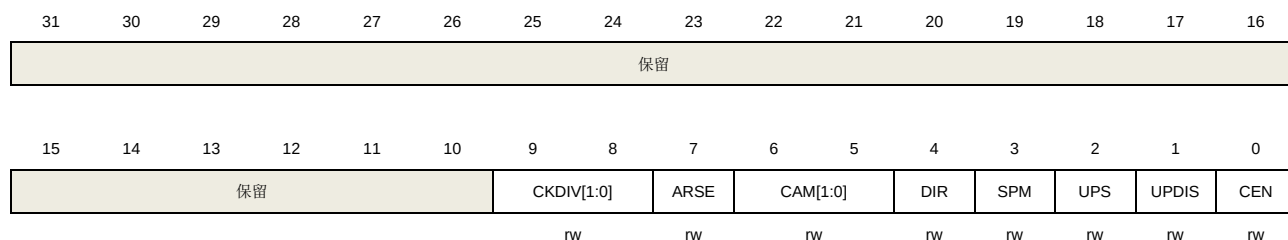
TIMER16 基地址: 0x4000 1800

控制寄存器 0 (TIMERx_CTL0)

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器通过字访问 (32 位)。



位/位域	名称	描述
31:10	保留	必须保持复位值
9:8	CKDIV[1:0]	时钟分频 通过软件配置 CKDIV, 规定定时器时钟 (TIMER_CK) 与死区时间和采样时钟 (DTS) 之间的分频系数, 死区发生器和数字滤波器会用到 DTS 时间。 00: $f_{DTS}=f_{TIMER\_CK}$ 01: $f_{DTS}=f_{TIMER\_CK}/2$ 10: $f_{DTS}=f_{TIMER\_CK}/4$ 11: 保留
7	ARSE	自动重载影子使能 0: 禁能 TIMERx_CAR 寄存器的影子寄存器 1: 使能 TIMERx_CAR 寄存器的影子寄存器
6:5	CAM[1:0]	计数器对齐模式选择 00: 无中央对齐模式 (边沿对齐模式)。DIR 位指定了计数方向。 01: 中央对齐向下计数置 1 模式。计数器在中央计数模式计数, 通道被配置在输出模式 (TIMERx_CHCTL0 寄存器中 CHxMS=3'b000), 只有在向下计数时, 通道的比较中断标志置 1 10: 中央对齐向上计数置 1 模式。计数器在中央计数模式计数, 通道被配置在输出模式 (TIMERx_CHCTL0 寄存器中 CHxMS=3'b000), 只有在向上计数时, 通道的比较中断标志置 1 11: 中央对齐上下计数置 1 模式。计数器在中央计数模式计数, 通道被配置在输出模式 (TIMERx_CHCTL0 寄存器中 CHxMS=00), 在向上和向下计数时, 通道的

比较中断标志都会置 1

当计数器使能以后，该位不能从 0x00 切换到非 0x00 状态。

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时，该位为只读。
3	SPM	单脉冲模式 0: 更新事件发生后，计数器继续计数 1: 在下次更新事件发生时，CEN 硬件清零并且计数器停止计数
2	UPS	更新请求源 软件配置该为，选择更新事件源。 0: 使能后，下述任一事件产生更新中断或 DMA 请求： – UPG 位被置 1 – 计数器上溢或者下溢 – 从模式控制器产生的更新 1: 使能后只有计数器上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 – 计数器上溢或者下溢 – 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，外部时钟模式才能工作。触发模式可以自动地通过硬件设置 CEN 位。

控制寄存器 1 (TIMERx_CTL1)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器通过字访问 (32 位)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCUC[2:1]			保留			MMC[3]			保留						

rw					rw											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
保留					ISO1	ISO0N	ISO0	TI0S	MMC[2:0]			DMAS	CCUC[0]	保留	CCSE	
					rw	rw	rw	rw	rw			rw	rw			rw

位/位域	名称	描述
31:30	CCUC[2:1]	换相控制影子寄存器更新控制 请参考 CCUC [0] 的描述。
29:26	保留	必须保持复位值
25	MMC[3]	主模式控制 请参考 MMC[2:0] 的描述。
24:11	保留	必须保持复位值
10	ISO1	通道 1 的空闲状态输出 参考 ISO0 位。
9	ISO0N	通道 0 的互补通道空闲状态输出 0: 当 POEN 复位, CH0_ON 输出低电平 1: 当 POEN 复位, CH0_ON 输出高电平 此位只有在 TIMERx_CCHP0 寄存器的 PROT[1:0] 位为 00 的时候可以被更改。
8	ISO0	通道 0 的空闲状态输出 0: 当 POEN 复位, CH0_O 输出低电平 1: 当 POEN 复位, CH0_O 输出高电平 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_CCHP0 寄存器的 PROT[1:0] 位为 00 的时候可以被更改。
7	TI0S	通道 0 触发输入选择 0: 选择 TIMERx_CH0 引脚作为通道 0 的触发输入 1: 选择 TIMERx_CH0, CH1 和 CH2 引脚异或的结果作为通道 0 的触发输入
6:4	MMC[2:0]	主模式控制 这些位控制 TRGO 信号的选择, TRGO 信号由主定时器发给从定时器用于同步功能 0000: 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。 0001: 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。 0010: 更新。主模式控制器选择更新事件作为 TRGO。

		0011: 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
		0100: 比较。在这种模式下主模式控制器选择 O0CPRE 信号被用于作为触发输出 TRGO
		0101: 比较。在这种模式下主模式控制器选择 O1CPRE 信号被用于作为触发输出 TRGO
		1000: 保留
		1001: 在这种模式下主模式控制器选择软同步事件信号(设置SWSYNCG位为1产生)作为触发输出TRGO。
		1010~1111: 保留
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 CHx/MCHx 的 DMA 请求 . 1: 当更新事件发生, 发送通道 CHx/MCHx 的 DMA 请求
2	CCUC[0]	换相控制影子寄存器更新控制 当换相控制影子寄存器 (CHxEN, MCHxEN 和 CHxCOMCTL 位) 使能 (CCSE=1), 这些影子寄存器更新控制如下: 000: CMTG 位被置 1 时, 更新影子寄存器 001: 当 CMTG 位被置 1 或检测到 TRGI 上升沿时, 影子寄存器更新 100: 当计数器上溢事件发生时, 影子寄存器更新 101: 当计数器下溢事件发生时, 影子寄存器更新 110: 当计数器上溢/ 下溢事件发生时, 影子寄存器更新 其他值: 保留 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、MCHxEN 和 CHxCOMCTL 位禁能. 1: 影子寄存器 CHxEN、MCHxEN 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

该寄存器通过字访问 (32 位)。

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	MSM	保留
rw		

位/位域	名称	描述
31:8	保留	必须保持复位值
7	MSM	主-从模式 该位被用来同步被选择的定时器同时开始计数。通过 TRIGI 和 TRGO，定时器被连接在一起，TRGO 用做启动事件。 0：主从模式禁能 1：主从模式使能
6:0	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

该寄存器通过字访问（32 位）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	CH1COM ADDIE	CH0COM ADDIE	保留	保留	保留	保留	MCH0DE N	保留	保留	保留	MCH0IE	保留	保留	保留	保留
rw		rw	rw				rw				rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TRGDEN	CMTDEN	保留	保留	CH1DEN	CH0DEN	UPDEN	BRKIE	TRGIE	CMTIE	保留	保留	CH1IE	CH0IE	UPIE
	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	CH1COMADDIE	通道 1 附加比较中断使能 0：禁止通道 1 附加比较中断 1：使能通道 1 附加比较中断 注意： 此中断使能位仅用于复合 PWM 模式。
28	CH0COMADDIE	通道 0 附加比较中断使能 0：禁止通道 0 附加比较中断 1：使能通道 0 附加比较中断 注意： 此中断使能位仅用于复合 PWM 模式。
27:25	保留	必须保持复位值。
24	MCH0DEN	多模式通道 0 比较/捕获 DMA 请求使能

		0: 禁止多模式通道 0 比较/捕获 DMA 请求 1: 使能多模式通道 0 比较/捕获 DMA 请求 注意: 此 DMA 使能位仅用于多模式通道输入和输出独立模式 (当 MCH0MSEL[1:0] = 2b'00 时)。
23:21	保留	必须保持复位值.
20	MCH0IE	多模式通道 0 比较/捕获中断使能 0: 禁止多模式通道 0 中断 1: 使能多模式通道 0 中断 注意: 此中断使能位仅用于多模式通道输入和输出独立模式 (当 MCH0MSEL[1:0] = 2b'00 时)。
19:15	保留	必须保持复位值.
14	TRGDEN	触发 DMA 请求使能 0: 禁止触发 DMA 请求 1: 使能触发 DMA 请求
13	CMTDEN	换相 DMA 更新请求使能 0: 禁止换相 DMA 更新请求 1: 使能换相 DMA 更新请求
12:11	保留	必须保持复位值.
10	CH1DEN	通道 1 比较/捕获 DMA 请求使能 0: 禁止通道 1 比较/捕获 DMA 请求 1: 使能通道 1 比较/捕获 DMA 请求
9	CH0DEN	通道 0 比较/捕获 DMA 请求使能 0: 禁止通道 0 比较/捕获 DMA 请求 1: 使能通道 0 比较/捕获 DMA 请求
8	UPDEN	更新 DMA 请求使能 0: 禁止更新 DMA 请求 1: 使能更新 DMA 请求
7	BRKIE	中止中断使能 0: 禁止中止中断 1: 使能中止中断
6	TRGIE	触发中断使能 0: 禁止触发中断 1: 使能触发中断
5	CMTIE	换相更新中断使能 0: 禁止换相更新中断

		1: 使能换相更新中断
4:3	保留	必须保持复位值.
2	CH1IE	通道 1 比较/捕获中断使能 0: 禁止通道 1 中断 1: 使能通道 1 中断
1	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

该寄存器通过字访问 (32 位)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	CH1COM ADDIF	CH0COM ADDIF	保留	保留	保留	保留	MCH0OF	保留	保留	保留	MCH0IF	保留	保留	保留	保留
	rc_w0	rc_w0					rc_w0				rc_w0				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	保留	保留	保留	保留	保留	保留	保留	BRKIF	TRGIF	CMTIF	保留	CH1IF	CH0IF	UPIF	
							rc_w0	rc_w0		rc_w0	rc_w0	rc_w0		rc_w0	rc_w0

位/位域	名称	描述
31:30	保留	必须保持复位值
29	CH1COMADDIF	通道 1 附加比较中断标志 参见 CH0COMADDIF 描述。
28	CH0COMADDIF	通道 0 附加比较中断标志 此标志由硬件置 1 软件清 0。当通道 0 用于输出模式时，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 注意: 此标志仅用于复合 PWM 模式。
27:25	保留	必须保持复位值
24	MCH0OF	多模式通道 0 捕获溢出标志

		当通道 0 被配置为输入模式时，在 MCH0IF 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：捕获溢出中断发生
23:21	保留	必须保持复位值
20	MCH0IF	多模式通道 0 比较/捕获中断标志 此标志由硬件置 1 软件清 0。当多模式通道 0 用于输入模式时，捕获事件发生时此标志位置 1；当多模式通道 0 用于输出模式时，此标志位在一个比较事件发生时置 1。 当多模式通道 0 在输入模式下时，通过读 TIMERx_MCH0CV 寄存器可以清零该位。 0：无多模式通道 0 中断发生 1：多模式通道 0 中断发生
19:11	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	保留	必须保持复位值。
7	BRKIF	BREAK 中断标志位 一旦 BREAK 输入有效，由硬件对该位置‘1’。如果 BREAK 输入无效，则该位可由软件清‘0’。 0：无 BREAK 事件产生 1：BREAK 输入上检测到有效电平
6	TRGIF	触发中断标志 当发生触发事件时，此标志由硬件置 1。此位由软件清 0。当从模式控制器处于除暂停模式外的其它模式时，在触发输入端检测到有效边沿，产生触发事件。当从模式控制器处于暂停模式时，触发输入的任意边沿都可以产生触发事件。 0：无触发事件产生 1：触发中断产生
5	CMTIF	通道换相更新中断标志 当通道换相更新事件发生时此标志位被硬件置 1，此位由软件清 0。 0：无通道换相更新中断发生

		1: 通道换相更新中断发生
4:3	保留	必须保持复位值.
2	CH1IF	通道 1 比较/捕获中断标志 参见 CH0IF 描述
1	CH0IF	通道 0 比较/捕获中断标志 此标志由硬件置 1 软件清 0。当通道 0 在输入模式下时，捕获事件发生时此标志位被置 1；当通道 0 在输出模式下时，此标志位在一个比较事件发生时被置 1。 当通道 0 在输入模式下时，通过读 <code>TIMERx_CH0CV</code> 寄存器可以清零该位。 0: 无通道 0 中断发生 1: 通道 0 中断发生
0	UPIF	更新中断标志 此位在任何更新事件发生时由硬件置 1，软件清 0。 0: 无更新中断发生 1: 发生更新中断

软件事件产生寄存器 (TIMERx_SWEVG)

地址偏移: 0x14

复位值: 0x0000 0000

该寄存器通过字访问 (32 位)。



位/位域	名称	描述
31:30	保留	必须保持复位值
29	CH1COMADDG	通道 1 附加比较事件发生 参见 CH0COMADDG 描述。
28	CH0COMADDG	通道 0 附加比较事件发生 该位由软件置 1，用于在通道 0 产生一个比较事件，由硬件自动清 0。当此位被置 1，CH0COMADDIF 标志位被置 1，若开启对应的中断和 DMA，则发出相应的中断请求。 0: 不产生通道 0 附加比较事件 1: 发生通道 0 附加比较事件

注意：此位仅用于复合 PWM 模式。

27:21	保留	必须保持复位值
20	MCH0G	多模式通道 0 互补捕获或比较事件发生 该位由软件置 1，用于在多模式通道 0 产生一个捕获/比较事件，由硬件自动清 0。当此位被置 1，MCH0IF 标志位被置 1，若开启相应的中断和 DMA，则发出相应的中断和 DMA 请求。此外，如果多模式通道 0 配置为输入模式，计数器的当前值被 TIMERx_MCH0CV 寄存器捕获，如果 MCH0IF 标志位已经为 1，则 MCH0OF 标志位被置 1。 0：不产生多模式通道 0 捕获或比较事件 1：发生多模式通道 0 捕获或比较事件
19:8	保留	必须保持复位值
7	BRKG	产生 BREAK 事件 该位由软件置 1，用于产生一个 BREAK 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 BREAK 事件 1：产生 BREAK 事件
6	TRGG	触发事件产生 此位由软件置 1，由硬件自动清 0。当此位被置 1，TIMERx_INTF 寄存器的 TRGIF 标志位被置 1，若开启对应的中断和 DMA，则产生相应的中断和 DMA 传输。 0：无触发事件产生 1：产生触发事件
5	CMTG	通道换相更新事件发生 此位由软件置 1，由硬件自动清 0。当此位被置 1，通道捕获/比较控制寄存器（CHxEN、MCHxEN 和 CHxCOMCTL）的互补输出被更新（根据 TIMERx_CTL1 中 CCSE 值）。 0：不产生通道控制更新事件 1：产生通道控制更新事件
4:3	保留	必须保持复位值。
2	CH1G	通道 1 捕获或比较事件发生 参见 CH0G 描述。
1	CH0G	通道 0 捕获或比较事件发生 该位由软件置 1，用于在通道 0 产生一个捕获/比较事件，由硬件自动清 0。当此位被置 1，CH0IF 标志位被置 1，若开启对应的中断和 DMA，则发出相应的中断和 DMA 请求。此外，如果通道 0 配置为输入模式，计数器的当前值被 TIMERx_CH0CV 寄存器捕获，如果 CH0IF 标志位已经为 1，则 CH0OF 标志位被置 1。

0: 不产生通道 0 捕获或比较事件

1: 发生通道 0 捕获或比较事件

0

UPG

更新事件产生

此位由软件置 1，被硬件自动清 0。当此位被置 1，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

该寄存器通过字访问 (32 位)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH1MS [2]	CH0MS [2]	CH1COM	CH0COM	CH1ADDU	CH0ADDU	保留	CH1COM	保留							CH0COM
		ADDSEN	ADDSEN	PS	PS		CTL[3]								CTL[3]
		保留					保留								保留
rw	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	CH1COMCTL[2:0]			CH1COM SEN	CH1COM FEN	CH1MS[1:0]		保留	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	CH1MS[2]	通道 1 I/O 模式选择 参考 CH1MS[1:0]描述。
30	CH0MS[2]	通道 0 I/O 模式选择 参考 CH0MS[1:0]描述。
29	CH1COMADDSEN	通道 1 附加输出比较影子寄存器使能 参考 CH0COMADDSEN 描述。
28	CH0COMADDSEN	通道 0 附加输出比较影子寄存器使能 当此位被置 1，TIMERx_CH0COMV_ADD 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: 通道 0 附加比较输出影子寄存器禁能 1: 通道 0 附加比较输出影子寄存器使能 仅在单脉冲模式下 (TIMERx_CTL0 寄存器的 SPM =1)，可以在未确认预装载寄存器情况下使用 PWM 模式。 当 TIMERx_CCHP0 寄存器的 PROT [1:0]=11 且 CH0MS =000 时此位不能被改

		变。
27	CH1ADDUPS	通道1附加寄存器更新源 0: 在发生更新事件时, 更新TIMERx_CH1COMV_ADD寄存器 1: 在发生计数器计数值匹配 CH1VAL 值时, 更新 TIMERx_CH1COMV_ADD 寄存器
26	CH0ADDUPS	通道0附加寄存器更新源 0: 在发生更新事件时, 更新TIMERx_CH0COMV_ADD寄存器 1: 在发生计数器计数值匹配 CH0VAL 值时, 更新 TIMERx_CH0COMV_ADD 寄存器
25	保留	必须保持复位值
24	CH1COMCTL[3]	通道 1 输出比较控制 参见 CH0COMCTL[2:0]描述
23:17	保留	必须保持复位值
16	CH0COMCTL[3]	通道 0 输出比较控制 参见 CH0COMCTL[2:0]描述
15	保留	必须保持复位值
14:12	CH1COMCTL[2:0]	通道 1 输出比较模式 参见 CH0COMCTL 描述
11	CH1COMSEN	通道 1 输出比较影子寄存器使能 参见 CH0COMSEN 描述
10	CH1COMFEN	通道1输出比较快速使能 参见 CH0COMFEN 描述
9:8	CH1MS[1:0]	通道 1 模式选择 CH1MS[2:0]位域定义了通道的方向和输入信号的选择。只有当通道关闭（当 MCH1MSEL[1:0] = 2b'00 时, TIMERx_CHCTL2 寄存器的 CH1EN 位清 0; 当 MCH1MSEL[1:0] = 2b'01 或 2b'11 时, TIMERx_CHCTL2 寄存器的 CH1EN、MCH1EN 位清 0）时, 这些位才可以写。 000: 通道 1 配置为输出 001: 通道 1 配置为输入, IS1 映射在 CI1FE1 上 010: 通道 1 配置为输入, IS1 映射在 CI0FE1 上 011: 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 SYSCFG_TIMERxCFG2(x=15,16)寄存器中的 TSCFG15[4:0]位域选择）。 100~111: 保留
7	保留	必须保持复位值

6:4	CH0COMCTL[2:0]	通道 0 输出比较模式 CH0COMCTL[3]和 CH0COMCTL[2:0]位域定义了输出准备信号 O0CPRE 的动作，而 O0CPRE 决定了 CH0_O 和 MCH0_O 的值。O0CPRE 高电平有效，而 CH0_O 和 MCH0_O 的有效电平取决于 CH0P 和 MCH0P 位。 注意：当多模式通道 0 配置为输出模式，且 MCH0MSEL[1:0] = 2b'11 时，CH0COMCTL[3]和 CH0COMCTL[2:0]位域定义了输出准备信号 O0CPRE 的动作，而 O0CPRE 决定了 CH0_O、MCH0_O 的值。O0CPRE 高电平有效，CH0_O、MCH0_O 的有效电平取决于 CH0P、MCH0P 位。 0000：时基。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 0001：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。 0010：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。 0011：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。 0100：强制为低。强制 O0CPRE 为低电平 0101：强制为高。强制 O0CPRE 为高电平 0110：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。 0111：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。 1000~1111：保留 注意：在复合 PWM 模式下（CH0CPWMEN = 1'b1 和 CH0MS = 3'b000），通道 0 的 PWM 输出信号由 TIMERx_CH0CV 和 TIMERx_CH0COMV_ADD 寄存器共同确定。详细信息请参考复合 PWM 模式。 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时基模式切换到 PWM 模式时，O0CPRE 电平才改变。 当 CH0 和 MCH0 输出互补时，该位域预装载。若 CCSE =1，则该位域只在通道换相事件发生时更新。 当 TIMERx_CCHP0 寄存器的 PROT [1:0]=11 且 CH0MS =000（比较模式）时此位不能被改变。
3	CH0COMSEN	通道 0 输出比较影子寄存器使能 当此位被置 1，TIMERx_CH0CV 寄存器的影子寄存器被使能，影子寄存器在每次更新事件时都会被更新。 0：禁止通道 0 输出/比较影子寄存器 1：使能通道 0 输出/比较影子寄存器 仅在单脉冲模式下（TIMERx_CTL0 寄存器的 SPM =1），可以在未确认预装载寄存器情况下使用 PWM 模式

当 `TIMERx_CCHP0` 寄存器的 `PROT [1:0]=11` 且 `CH0MS =000` 时此位不能被改变。

2	CH0COMFEN	通道0输出比较快速使能 当该位为1时，如果通道配置为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，<code>CH0_O</code>被设置为比较电平而与比较结果无关。 0：通道0输出比较快速禁能。当触发器的输入有一个有效沿时，激活<code>CH0_O</code>输出的最小延时为5个时钟周期 1：通道0输出比较快速使能。当触发器的输入有一个有效沿时，激活<code>CH0_O</code>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（当 <code>MCH0MSEL[1:0] = 2b'00</code> 时，<code>TIMERx_CHCTL2</code> 寄存器的 <code>CH1EN</code> 位清 0；当 <code>MCH0MSEL[1:0] = 2b'01</code> 或 <code>2b'11</code> 时，<code>TIMERx_CHCTL2</code> 寄存器的 <code>CH0EN</code>、<code>MCH0EN</code> 位清 0）时，<code>CH0MS[2:0]</code>才可写。 000：通道 0 配置为输出 001：通道 0 配置为输入，IS0 映射在 <code>CI0FE0</code> 上 010：通道 0 配置为输入，IS0 映射在 <code>CI1FE0</code> 上 011：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（由 <code>SYSCFG_TIMERxCFG2(x=15,16)</code>寄存器中的 <code>TSCFG15[4:0]</code>位域选择）。 100：通道 0 配置为输入，IS0 映射在 <code>MCI0FE0</code> 上 101~111：保留

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 <code>CH0CAPFLT</code> 描述
11:10	CH1CAPPSC[1:0]	通道 1 输入捕获预分频器 参见 <code>CH0CAPPSC</code> 描述
9:8	CH1MS[1:0]	通道 1 模式选择 与输出模式相同
7:4	CH0CAPFLT[3:0]	通道 0 输入捕获滤波控制 数字滤波器由一个事件计数器组成，它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 <code>CI0</code> 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：$f_{SAMP} = f_{CK\_TIMER}$，$N=2$ 0010：$f_{SAMP} = f_{CK\_TIMER}$，$N=4$

		0011: $f_{SAMP} = f_{CK\_TIMER}$, $N=8$
		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 <code>TIMERx_CHCTL2</code> 寄存器中的 <code>CH0EN</code> =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.		CH1PER FOREN	CH0PER FOREN	保留.											
		r/w	r/w												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留.								MCH1P	保留	CH1P	CH1EN	MCH0P	MCH0EN	CH0P	CH0EN
								r/w		r/w	r/w	r/w	r/w	r/w	r/w

位/位域	名称	描述
31:30	保留	必须保持复位值
29	CH1PERFOREN	Channel 1 在复合 PWM 模式下，周期点匹配时刻 O1CPRE 电平值设置。

		参考 CH0PERFOREN 描述。
28	CH0PERFOREN	Channel 0在复合PWM模式下，周期点匹配时刻O0CPRE电平值设置。 0：禁能 1：复合PWM0模式下，周期点匹配时刻电平值强制拉高，复合PWM1模式下，周期点匹配时刻电平值强制拉低。 注意：中央对齐模式下，周期点为计数器下溢时刻；边沿对齐模式下，周期点为计数器溢出时刻。
27:8	保留	必须保持复位值
7	MCH1P	多模式通道 1 捕获/比较极性 参考 MCH0P 描述。
6	保留	必须保持复位值
5	CH1P	通道 1 捕获/比较极性 参考 CH0P 描述。
4	CH1EN	通道 1 捕获/比较使能 参考 CH0EN 描述。
3	MCH0P	多模式通道 0 捕获/比较极性 当通道 0 配置为输出模式，且 MCH0MSEL[1:0] = 2b'11 时，此位定义了多模式通道 0 输出信号 MCH0_O 的极性。 0：多模式通道 0 高电平有效 1：多模式通道 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，作为通道 0 的极性选择控制信号。 当 TIMERx_CCHP0 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。
2	MCH0EN	多模式通道 0 捕获/比较使能 当多模式通道 0 配置为输出模式时，将此位置 1 使能 MCH0_O 信号有效。当多模式通道 0 配置为输入模式时，将此位置 1 使能多模式通道 0 上的捕获事件。 0：禁止多模式通道 0 1：使能多模式通道 0
1	CH0P	通道 0 捕获/比较极性 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，此位定义了通道 0 输入信号的极性。[MCH0P, CH0P] 用于选择通道 0 输入信号信号有效边沿或者捕获极性。 00：把通道 0 输入信号的上升沿作为捕获或者从模式下触发的有效信号，且通道 0 输入信号不会被翻转。 01：把通道 0 输入信号的下降沿作为捕获或者从模式下触发的有效信号，且通道 0

输入信号会被翻转。

10: 保留。

11: 把通道 0 输入信号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，且通道 0 输入信号不翻转。

当 `TIMERx_CCHP0` 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`

该寄存器通过字访问 (32 位)。



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`

该寄存器通过字访问 (32 位)。



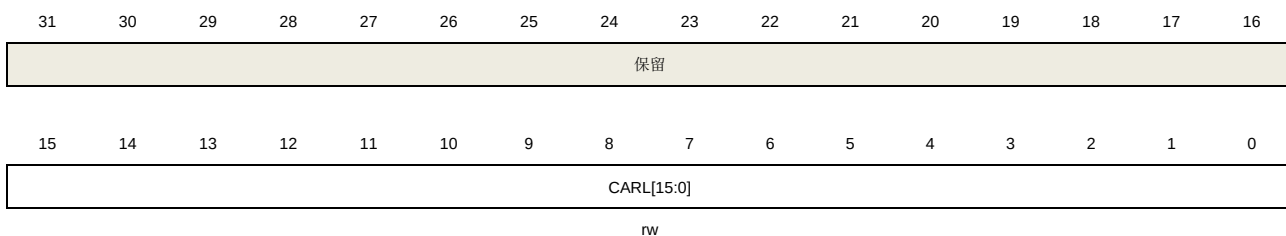
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 PSC 时钟除以 (PSC+1)，每次当更新事件产生时，PSC 的值被装入当前预分频寄存器。

计数器自动重载寄存器 (TIMERx_CAR)

地址偏移：0x2C

复位值：0x0000 FFFF

该寄存器通过字访问 (32 位)。



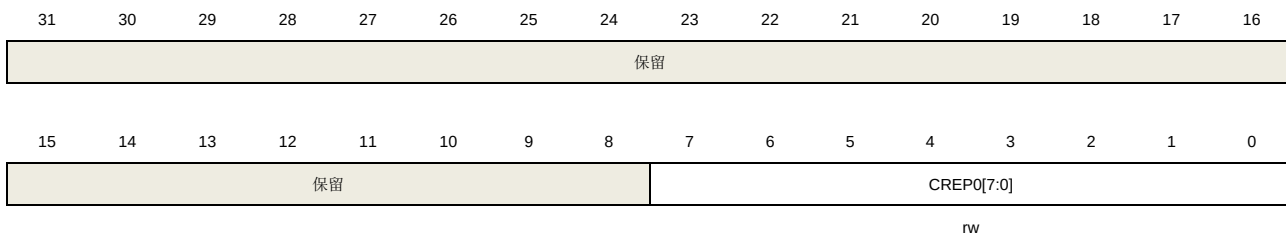
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

重复计数寄存器 0 (TIMERx_CREP0)

地址偏移：0x30

复位值：0x0000 0000

该寄存器通过字访问 (32 位)。



位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	CREP0[7:0]	重复计数器的值 0 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）。

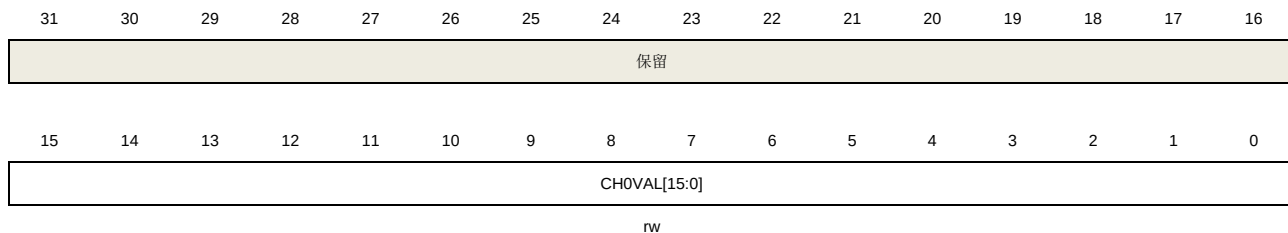
注意：当 `TIMERx_CFG` 寄存器中的 `CREPSEL = 0` 时，使用该位。

通道 0 捕获/比较寄存器（`TIMERx_CH0CV`）

地址偏移：0x34

复位值：0x0000 0000

该寄存器通过字访问（32 位）。



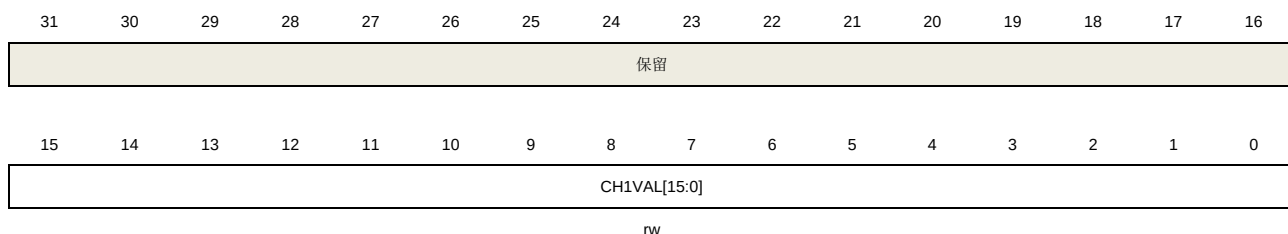
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CHOVAL[15:0]	通道 0 的捕获或比较值 当通道 0 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 1 捕获/比较寄存器（`TIMERx_CH1CV`）

地址偏移：0x38

复位值：0x0000 0000

该寄存器通过字访问（32 位）。



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。

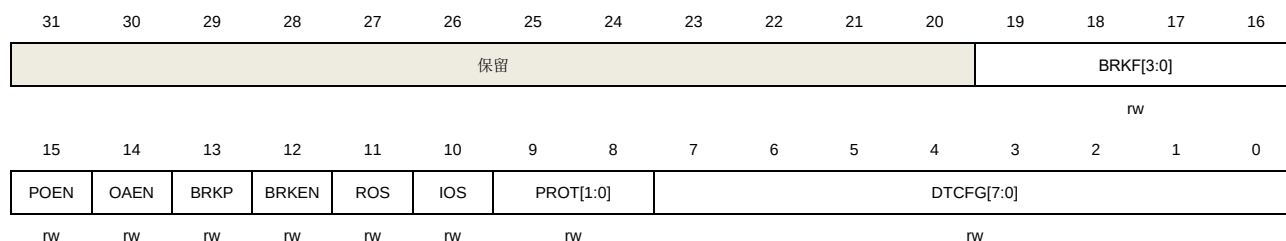
当通道 1 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

互补通道保护寄存器 0 (TIMERx_CCHP0)

地址偏移：0x44

复位值：0x0000 0000

该寄存器通过字访问（32 位）。



位/位域	名称	描述
31:20	保留	必须保持复位值
19:16	BRKF[3:0]	BREAK输入信号滤波 数字滤波器由一个事件计数器组成，它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 BREAK 输入信号的采样频率和数字滤波器的长度。 0000: 无滤波器，BREAK 异步有效，N=1 0001: $f_{SAMP} = f_{CK\_TIMER}$, N=2 0010: $f_{SAMP} = f_{CK\_TIMER}$, N=4 0011: $f_{SAMP} = f_{CK\_TIMER}$, 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 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0] =00 时才可修改。
15	POEN	所有的通道输出使能 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, MCHxEN 位），则开启 CHx_O 和 MCHx_O 输出。 0: 禁止通道输出或强制为空闲状态 1: 使能通道输出
14	OAEN	自动输出使能 此位定义了 POEN 位是否可以被硬件自动置 1。 0: POEN 位不能被硬件置 1 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 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0] =00 时才可修改。
13	BRKP	BREAK 输入信号极性 此位定义了 BREAK 输入的极性。 0: BREAK 输入低电平有效 1: BREAK 输入高电平有效 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0] =00 时才可修改。
12	BRKEN	BREAK 输入信号使能 此位置 1 使能 BREAK 输入信号。 0: BREAK 输入禁能 1: BREAK 输入使能 此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0] =00 时才可修改。
11	ROS	运行模式下“关闭状态”使能 当 POEN 位被置 1（运行模式），此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出“关闭状态”。参见 表 12-13. 由参数控制的互补输出表 (MCHxMSEL=2'b11)。 0: 输出“关闭状态”禁能。当 CHxEN 或者 MCHxEN 位被清零，对应通道为输出“禁能状态”。 1: 输出“关闭状态”使能。当 CHxEN 或者 MCHxEN 位被清零，对应通道为输出“关闭状态”。 此位在 TIMERx_CCHP0 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”使能 当 POEN 位被清 0（空闲模式），此位可以被置 1 来使能通道(带有互补输出且配置为输出模式)的输出“关闭状态”。参见 表 12-13. 由参数控制的互补输出表 (MCHxMSEL=2'b11)。 0: 输出“关闭状态”禁能。当 CHxEN 和 MCHxEN 位均被清零，对应通道为输出“禁能状态”。 1: 输出“关闭状态”使能。不论 CHxEN 和 MCHxEN 位的值，对应通道为输出“关闭状态”。 此位在 TIMERx_CCHP0 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_CCHP0 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/MCHxP 位 (如果相应通道配置为输出模式), TIMERx_CCHP0 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_CHCTLR0/1 中 CHxCOMCTL/ CHxCOMSEN 位 (如果相关通道配置为输出模式) 写保护。

11: PROT 模式 2。除了 PROT 模式 1 下的寄存器写保护外, 还有 TIMERx_CHCTLR0 及 TIMERx_MCHCTL0 寄存器中 CHxCOMCTL/ CHxCOMSEN/ CHxCOMADDSEN/ MCHxCOMCTL/ MCHxCOMSEN 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_CCHP0 寄存器被写入, 这两位被写保护

7:0 DTCFG[7:0] 死区时间控制

这些位定义了插入互补输出之间的死区持续时间。DTCFG 值和死区时间的关系如下:

DTCFG [7:5] =3'b0xx: DTvalue =DTCFG [7:0]x tDT, tDT=tDTS.

DTCFG [7:5] =3'b 10x: DTvalue = (64+DTCFG [5:0]) xtDT, tDT =tDTS*2.

DTCFG [7:5] =3'b 110: DTvalue = (32+DTCFG [4:0]) xtDT, tDT=tDTS*8.

DTCFG [7:5] =3'b 111: DTvalue = (32+DTCFG [4:0]) xtDT, tDT =tDTS*16.

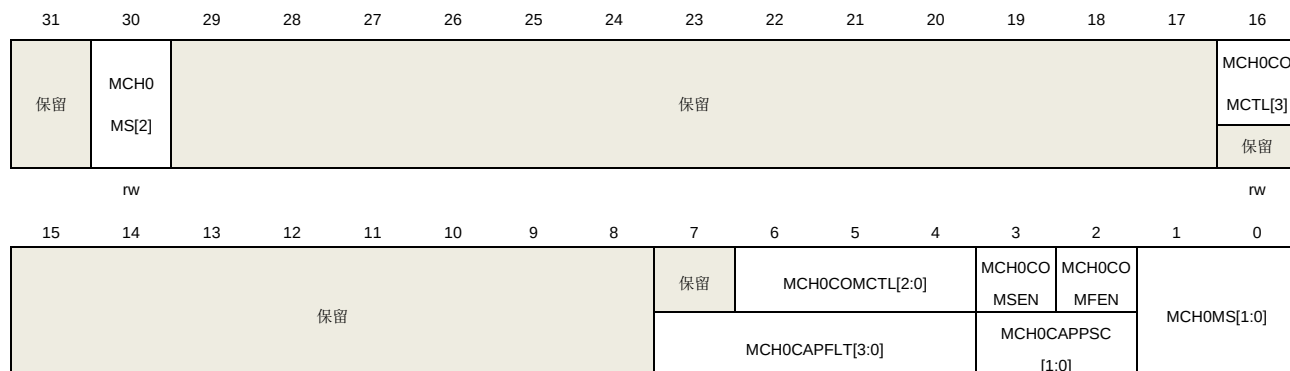
此位只有在 TIMERx_CCHP0 寄存器的 PROT [1:0]=00 时才可修改。

多模式通道控制寄存器 0 (TIMERx_MCHCTL0)

地址偏移: 0x48

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



输出比较模式:

位/位域	名称	描述
31	保留	必须保持复位值。
30	MCH0MS[2]	多模式通道 0 I/O 模式选择 参考MCH0MS[1:0]描述。
29:17	保留	必须保持复位值。
16	MCH0COMCTL [3]	多模式通道 0 输出比较控制 请参考MCH0COMCTL[2:0]描述。
15:7	保留	必须保持复位值。
6:4	MCH0COMCTL [2:0]	多模式通道 0 输出比较控制 当多模式通道 0 配置为输出模式, 并且 MCH0MSEL[1:0] = 2b'00, MCH0COMCTL[3] 和 MCH0COMCTL[2:0]位域定义了输出准备信号 MO0CPRE 的动作, 而 MO0CPRE 决定了 MCH0_O 的值。MO0CPRE 高电平有效, 而 MCH0_O 的有效电平取决于 MCH0FP[1:0]位。 注意: 当多模式通道 0 配置为输出模式, 且 MCH0MSEL[1:0] = 2b'11 时, CH0COMCTL[2:0]位定义了输出准备信号 O0CPRE 的动作, 而 O0CPRE 决定了 CH0_O、MCH0_O 的值。O0CPRE 高电平有效, CH0_O、MCH0_O 的有效电平取决于 CH0P、MCH0P 位。 0000: 时基。输出比较寄存器 TIMERx_MCH0CV 与计数器 TIMERx_CNT 间的比较对 MO0CPRE 不起作用 0001: 匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_MCH0CV 相同时, 强制 MO0CPRE 为高。 0010: 匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_MCH0CV 相同时, 强制 MO0CPRE 为低。 0011: 匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_MCH0CV 相同时, 强制 MO0CPRE 翻转。 0100: 强制为低。强制 MO0CPRE 为低电平 0101: 强制为高。强制 MO0CPRE 为高电平 0110: PWM 模式 0。在向上计数时, 一旦计数器值小于 TIMERx_MCH0CV 时, MO0CPRE 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_MCH0CV 时, MO0CPRE 为无效电平, 否则为有效电平。 0111: PWM 模式 1。在向上计数时, 一旦计数器值小于 TIMERx_MCH0CV 时, MO0CPRE 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_MCH0CV 时, MO0CPRE 为有效电平, 否则为无效电平。 1000~1111: 保留 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时基模式切换到 PWM 模式时, MO0CPRE 电平才改变。

当 CH0 和 MCH0 输出互补时，该位域预装载。若 CCSE =1，则该位域只在通道换相事件发生时更新。

当TIMERx_CCHP0寄存器的PROT[1:0]=11且MCH0MS=000（比较模式）时此位不能被改变。

3	MCH0COMSEN	多模式通道 0 输出比较影子寄存器使能 当此位被置 1，TIMERx_MCH0CV 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：禁止多模式通道 0 输出/比较影子寄存器 1：使能多模式通道 0 输出/比较影子寄存器 仅在单脉冲模式下（TIMERx_CTL0 寄存器的 SPM =1），可以在未确认预装载寄存器情况下使用 PWM 模式。 当TIMERx_CCHP0寄存器的PROT [1:0]=11且CH0MS=00时此位不能被改变。
2	MCH0COMFEN	多模式通道0输出比较快速使能 当该位为1时，如果通道配置为PWM0模式或者PWM1模式，会加快捕获/比较输出对触发输入事件的响应。输出通道将触发输入信号的有效边沿作为一个比较匹配，MCH0_O被设置为比较电平而与比较结果无关。 0：禁止多模式通道0输出比较快速。当触发器的输入有一个有效沿时，激活MCH0_O输出的最小延时为5个时钟周期 1：使能多模式通道0输出比较快速。当触发器的输入有一个有效沿时，激活MCH0_O输出的最小延时为3个时钟周期
1:0	MCH0MS[1:0]	多模式通道 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 MCH0EN 位清 0）时，MCH0MS[2:0]才可写。 000：多模式通道 0 配置为输出 001：多模式通道 0 配置为输入，MIS0 映射在 MCIOFEM0 上 010：保留 011：多模式通道 0 配置为输入，MIS0 映射在 ITS 上。此模式仅工作在内部触发输入被选中时（由 SYSCFG_TIMERxCFG2(x=14,40,41,42,43,44)寄存器中的 TSCFG15[4:0]位域选择）。 100：多模式通道0配置为输入，MIS0映射在CIOFEM0上。 101~111：保留

输入捕获模式：

位/位域	名称	描述
31	保留	必须保持复位值
30	MCH0MS[2]	多模式通道 0 I/O 模式选择 参考MCH0MS[1:0]描述。
29:8	保留	必须保持复位值

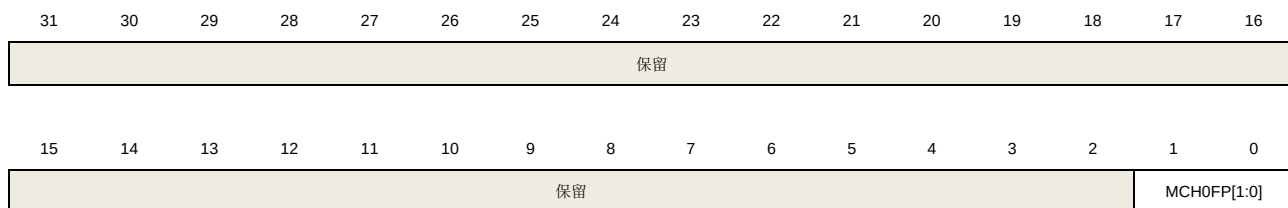
7:4	MCH0CAPFLT[3:0]	通道 0 输入捕获滤波控制 数字滤波器由一个事件计数器组成，它记录 N 个输入事件后会产生一个输出的跳变。这些位定义了 MCH0 输入信号的采样频率和数字滤波器的长度。 0000: 无滤波器, $f_{SAMP} = f_{DTS}$, $N=1$ 0001: $f_{SAMP} = f_{CK\_TIMER}$, $N=2$ 0010: $f_{SAMP} = f_{CK\_TIMER}$, $N=4$ 0011: $f_{SAMP} = f_{CK\_TIMER}$, $N=8$ 0100: $f_{SAMP} = f_{DTS}/2$, $N=6$ 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	MCH0APPSC[1:0]	多模式通道 0 输入捕获预分频器 这 2 位定义了多模式通道 0 输入的预分频系数。当 <code>TIMERx_CHCTL2</code> 寄存器中的 <code>MCH0EN = 0</code> 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	MCH0MS[1:0]	多模式通道 0 模式选择 与输出比较模式相同

多模式通道控制寄存器 2 (TIMERx_MCHCTL2)

地址偏移: 0x50

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



rw

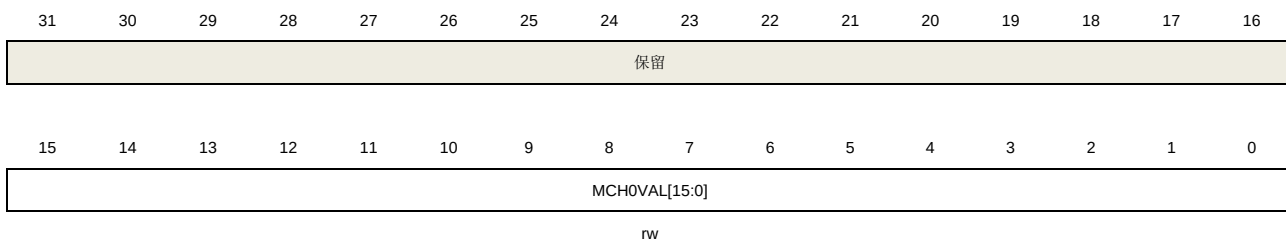
位/位域	名称	描述
31:2	保留	必须保持复位值
1:0	MCH0FP[1:0]	多模式通道 0 捕获/比较独立极性控制 当多模式通道 0 配置为输出模式时，且 MCH0MSEL[1:0] = 2b'00，此位定义了输出信号极性。 00: 多模式通道 0 高电平有效 01: 多模式通道 0 低电平有效 10: 保留 11: 保留 当通道 0 配置为输入模式时，此位定义了多模式通道 0 输入信号的极性。 MCH0FP[1:0]将选择多模式通道 0 输入信号的有效边沿或者捕获极性。 00: 把多模式通道 0 输入信号的上升沿作为捕获或者从模式下触发的有效信号，且多模式通道 0 输入信号不会被翻转。 01: 把多模式通道 0 输入信号的下降沿作为捕获或者从模式下触发的有效信号，且多模式通道 0 输入信号会被翻转。 10: 保留。 11: 把多模式通道 0 输入信号的上升沿或下降沿作为捕获或者从模式下触发的有效信号，并且多模式通道 0 输入信号不会被翻转。 当 TIMERx_CCHP0 寄存器的 PROT [1:0]=11 或 10 时此位不能被更改。

多模式通道 0 捕获/比较寄存器 (TIMERx_MCH0CV)

地址偏移: 0x54

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	MCH0VAL[15:0]	多模式通道 0 的捕获或比较值 当多模式通道 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，且本寄存器为只读。 当多模式通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相

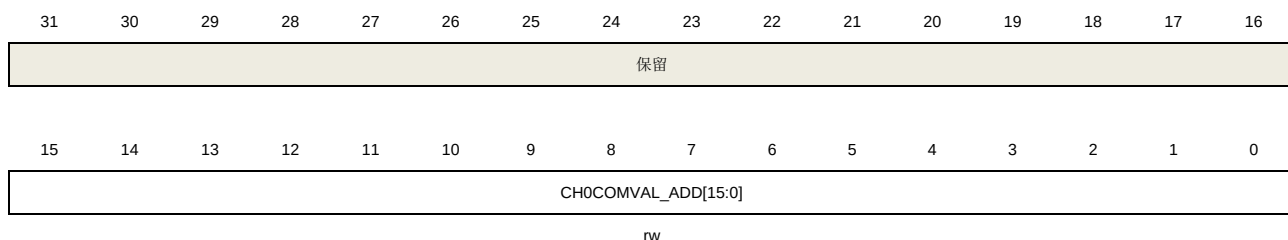
应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 0 附加比较寄存器（TIMERx_CH0COMV_ADD）

地址偏移：0x64

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



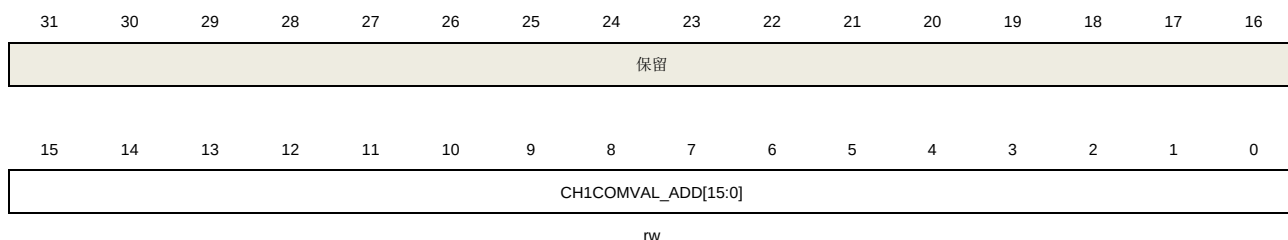
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CH0COMVAL_ADD [15:0]	通道0附加比较值 当通道0配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。 注意： 该寄存器仅用于复合PWM模式（当CH0CPWMEN=1时）。

通道 1 附加比较寄存器（TIMERx_CH1COMV_ADD）

地址偏移：0x68

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	CH1COMVAL_ADD [15:0]	通道1附加比较值 当通道1附加配置为输出模式时，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。 注意： 该寄存器仅用于复合PWM模式（当CH0CPWMEN=1时）。

控制寄存器 2 (TIMERx_CTL2)

地址偏移: 0x74

复位值: 0x0030 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		CH1C PWMEN	CH0C PWMEN	保留						MCH0MSEL[1:0]		保留			
		rw	rw							rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				CH1OMPSEL[1:0]		CH0OMPSEL[1:0]		保留							
				rw		rw									

位/位域	名称	描述
31:30	保留	必须保持复位值
29	CH1CPWMEN	通道 1 复合 PWM 模式使能 0: 通道 1 复合 PWM 模式禁能 1: 通道 1 复合 PWM 模式使能
28	CH0CPWMEN	通道 0 复合 PWM 模式使能 0: 通道 0 复合 PWM 模式禁能 1: 通道 0 复合 PWM 模式使能
27:22	保留	必须保持复位值
21:20	MCH0MSEL[1:0]	多模式通道 0 模式选择 00: 独立模式, MCH0 独立于 CH0 01: 保留 10: 保留 11: 互补模式, 只有 CH0 可用于输入, MCH0 输出与 CH0 输出互补
19:12	保留	必须保持复位值
11:10	CH1OMPSEL[1:0]	通道 1 输出匹配脉冲选择 当匹配事件发生时, 该位用于选择准备输出信号 O1CPRE (用来驱动 CH1_O 信号)。 00: O1CPRE 信号根据 CH1COMCTL[2:0]位的配置正常输出。 01: 只有在计数器向上计数, 匹配事件发生时, O1CPRE 信号输出一个脉冲, 且脉冲宽度是一个 CK_TIMER 时钟周期。 10: 只有在计数器向下计数, 匹配事件发生时, O1CPRE 信号输出一个脉冲, 脉冲宽度是一个 CK_TIMER 时钟周期。 11: 在计数器向上计数或者向下计数, 匹配事件发生时, O1CPRE 信号输出一个脉冲, 脉冲宽度是一个 CK_TIMER 时钟周期。

9:8	CH0OMPSEL[1:0]	通道 0 输出匹配脉冲选择 当匹配事件发生时, 该位用于选择准备输出信号 O0CPRE (用来驱动 CH0_O 信号)。 00: O0CPRE 信号根据 CH0COMCTL[2:0]位的配置正常输出。 01: 只有在计数器向上计数, 匹配事件发生时, O0CPRE 信号输出一个脉冲, 并且脉冲宽度是一个 CK_TIMER 时钟周期。 10: 只有在计数器向下计数, 匹配事件发生时, O0CPRE 信号输出一个脉冲, 脉冲宽度是一个 CK_TIMER 时钟周期。 11: 在计数器向上计数或者向下计数, 匹配事件发生时, O0CPRE 信号输出一个脉冲, 脉冲宽度是一个 CK_TIMER 时钟周期。
7:0	保留	必须保持复位值

TIMERx 备用功能控制寄存器 0 (TIMERx_AFCTL0)

地址偏移: 0x8C

复位值: 0x0000 0001

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						BRKCOMP 0P	保留								BRKIN0P
rw							rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						BRKCOMP 0EN	保留								BRKIN0E N
rw							rw								

位/位域	名称	描述
31:26	保留	必须保持复位值。
25	BRKCOMP0P	BREAK CMP0输入极性 0: CMP0输入信号不反相 (BRKP =0, 输入信号低有效; BRKP =1, 输入信号高有效) 1: CMP输入信号反相 (BRKP =0, 输入信号高有效; BRKP =1, 输入信号低有效) 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] =00时才可修改。
24:17	保留	必须保持复位值。
16	BRKIN0P	BREAK BRKIN0备用输入极性 该位用于配置BRKIN0输入极性, 具体极性是由该位和BRKP位共同确定。 0: BRKIN0输入信号不反相 (BRKP =0, 输入信号低有效; BRKP =1, 输入信号高有效) 1: BRKIN0输入信号反相 (BRKP =0, 输入信号高有效; BRKP =1, 输入信号低有效)

		此位只有在TIMERx_CCHP0寄存器的PROT [1:0] =00时才可修改。
15:10	保留	必须保持复位值。
9	BRKCMPOEN	BREAK CMP0输入使能 0: CMP0输入禁能 1: CMP0输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] =00时才可修改。
8:1	保留	必须保持复位值。
0	BRKINOEN	BREAK BRKINO备用输入使能 0: BRKINO输入禁能 1: BRKINO输入使能 此位只有在TIMERx_CCHP0寄存器的PROT [1:0] =00时才可修改。

计数器初值控制寄存器(TIMERx_CINITCTL)

地址偏移: 0xA4

复位值: 0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:3	保留	必须保持复位值。
2	SWSYNCG	软件同步事件产生 该位由软件置位，硬件自动清零，对该位进行读操作，读取值均为0。 0: 无效 1: 产生软件更新事件
1	CINITDIR	计数器初始计数方向 0: 同步事件发生时，计数器向下计数 1: 同步事件发生时，计数器向上计数 当同步事件发生，且计数器从TIMERx_CINITV寄存器加载计数器初始值后，该位表明了此时计数器的计数初始方向。 注意： 该位仅用于CAM[1:0] ≠ 00的情况。

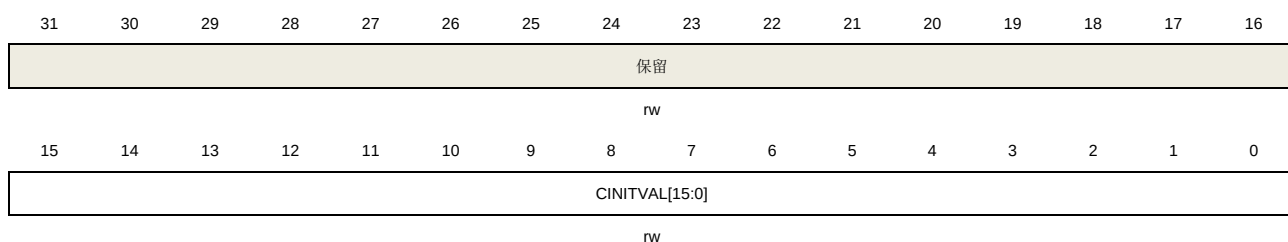
0	CINITVEN	计数器初值寄存器使能
		0: 计数器初值寄存器禁能。计数器寄存器不从计数器初值寄存器加载初值。
		1: 计数器初值寄存器使能。当同步事件发生（通过软件将SWSYNCG位置1产生软件同步事件，输入引脚输入硬件同步事件）时，计数器寄存器可以从计数器初值寄存器加载初值。

计数器初值寄存器(TIMERx_CINITV)

地址偏移: 0xA8

复位值: 0x0000 0000

该寄存器只能按字（32 位）访问。



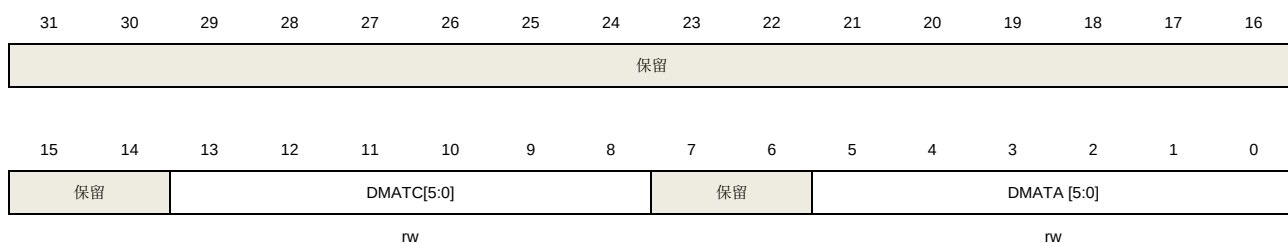
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CINITVAL[15:0]	计数器初始值 该位域用于表示计数器的初始值。 CINITVEN位为0时，计数器的寄存器不能加载在CINITVAL[15:0]位域中设置的初始值。 CINITVEN位为1时，当同步事件发生时，计数器的寄存器可以加载在CINITVAL[15:0]位域中设置的初始值。

DMA 配置寄存器 (TIMERx_DMACFG)

地址偏移: 0xE0

复位值: 0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述

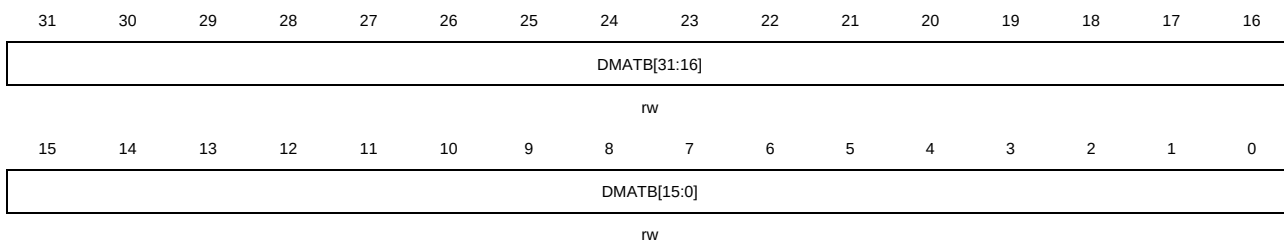
31:14	保留	必须保持复位值
13:8	DMATC [5:0]	DMA 传输计数 该位域定义了 DMA 访问（读/写）TIMERx_DMATB 寄存器的次数。 6'b000000: 传输1次 6'b000001: 传输2次 ... 6'b111111: 传输 63 次
7:6	保留	必须保持复位值
5:0	DMATA [5:0]	DMA 传输起始地址 该位域定义了 DMA 访问 TIMERx_DMAVB 寄存器的第一个地址。当通过 TIMERx_DMA 第一次访问时，访问的就是该位域指定的地址。第二次访问 TIMERx_DMATB 时，将访问起始地址+0x4。 6'b0_0000: TIMERx_CTL0 6'b0_0001: TIMERx_CTL1 ... 6'b111111: 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：0xE4

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



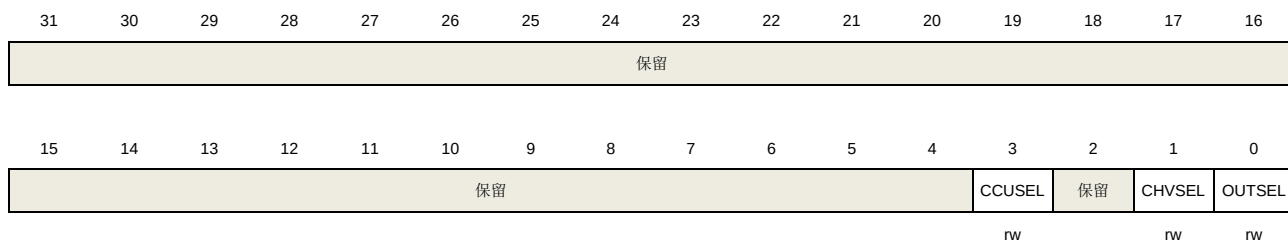
位/位域	名称	描述
31:0	DMATB [31:0]	DMA 发送缓冲 对这个寄存器的读或写，（起始地址+传输次数*4）地址范围内的寄存器会被访问 传输次数由硬件计算，范围为 0 到 DMATC。

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。



位/位域	名称	描述
31:4	保留	必须保持复位值
3	CCUSEL	换相控制影子寄存器更新选择 只有当CCUC[2:0]位域配置为100,101和110时，该位才有效。 0: 当计数器产生一个上溢/下溢事件时，影子寄存器才更新 1: 当重复计数器值为0，且计数器产生一个上溢/下溢事件时，影子寄存器才更新
2	保留	必须保持复位值
1	CHVSEL	写捕获比较寄存器选择位 此位由软件写 1 或清 0。 1: 当写入捕获比较寄存器的值与寄存器当前值相等时，写入操作无效 0: 无影响
0	OUTSEL	输出值选择位 此位由软件写 1 或清 0。 1: 如果 POEN 位与 IOS 位均为 0，则输出无效 0: 无影响

12.4. 基本定时器 (TIMERx, x=5,6)

12.4.1. 简介

基本定时器 (TIMER5/6) 包含一个无符号 16 位计数器。可以被用作通用定时器和为 DAC (数字到模拟转换器) 提供时钟。基本定时器可以配置产生 DMA 请求, TRGO 触发连接到 DAC。

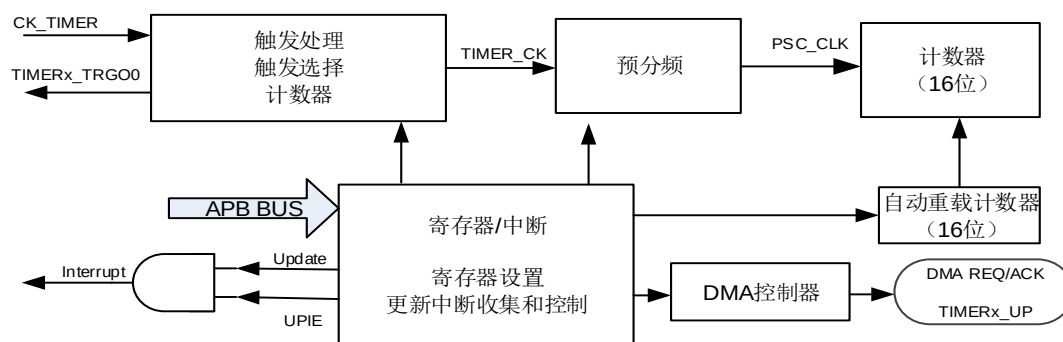
12.4.2. 主要特性

- 计数器宽度: 16位 (TIMER5/6)
- 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
- 自动重载功能.
- 中断输出和DMA请求: 更新事件

12.4.3. 结构框图

[图 12-103. 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 12-103. 基本定时器结构框图



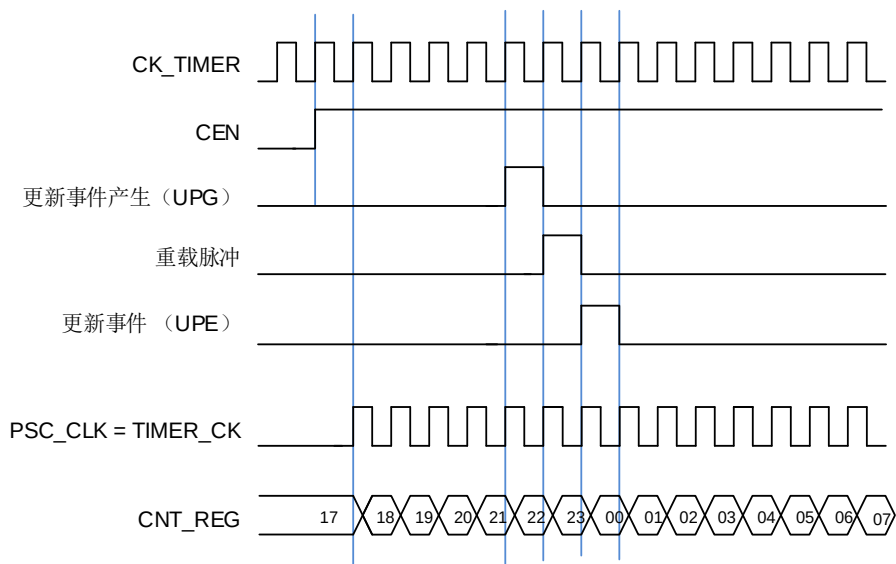
12.4.4. 功能描述

时钟源选择

基本定时器只能由内部时钟源 CK_TIMER 驱动 (来自 RCU 模块)。

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。

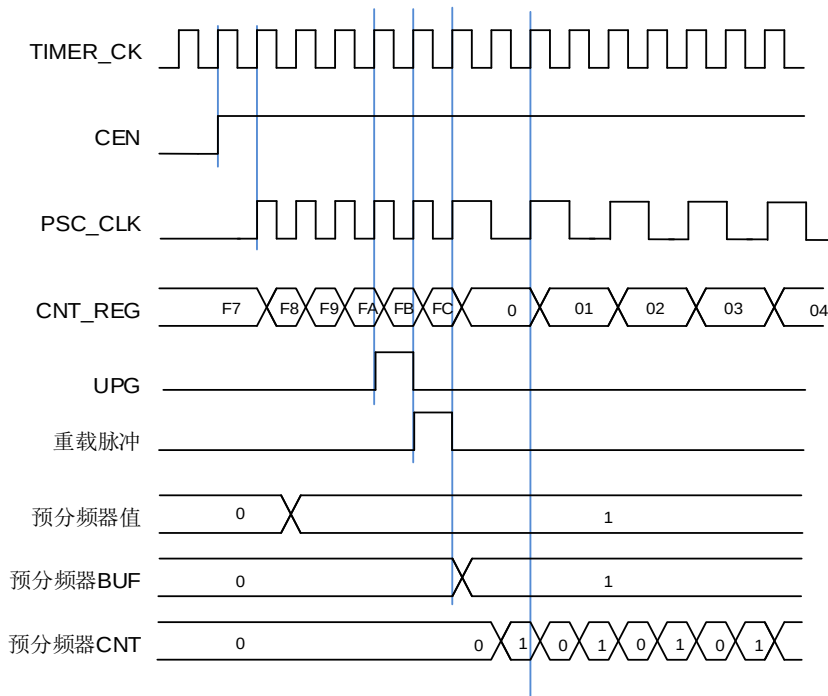
图 12-104. 内部时钟分频为 1 时正常模式下的控制电路



预分频

预分频器可以将定时器的时钟（TIMER_CLK）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 12-105. 当预分频器的参数从 1 变到 2 时，计数器的时序图



向上计数模式

在这种模式，计数器的计数方向是向上计数。计数器从 0 开始向上连续计数到自动加载值（定义在 `TIMERx_CAR` 寄存器中），一旦计数器计数到自动加载值，会重新从 0 开始向上计数并产生上溢事件。在向上计数模式中，`TIMERx_CTL0` 寄存器中的计数方向控制位 `DIR` 应该被设置成 0。

当通过 `TIMERx_SWEVG` 寄存器的 `UPG` 位置 1 来设置更新事件时，计数值会被清 0，并产生更新事件。

如果 `TIMERx_CTL0` 寄存器的 `UPDIS` 置 1，则禁止更新事件。

当发生更新事件时，所有的寄存器（自动重载寄存器，预分频寄存器）都将被更新。

下面这些图给出了一些例子，当 `TIMERx_CAR=0x99`（`TIMERx`, $x=5,6$ ）时，计数器在不同预分频因子下的行为。

图 12-106. 向上计数时序图，`PSC=0/2`（`TIMERx`, $x=5,6$ ）

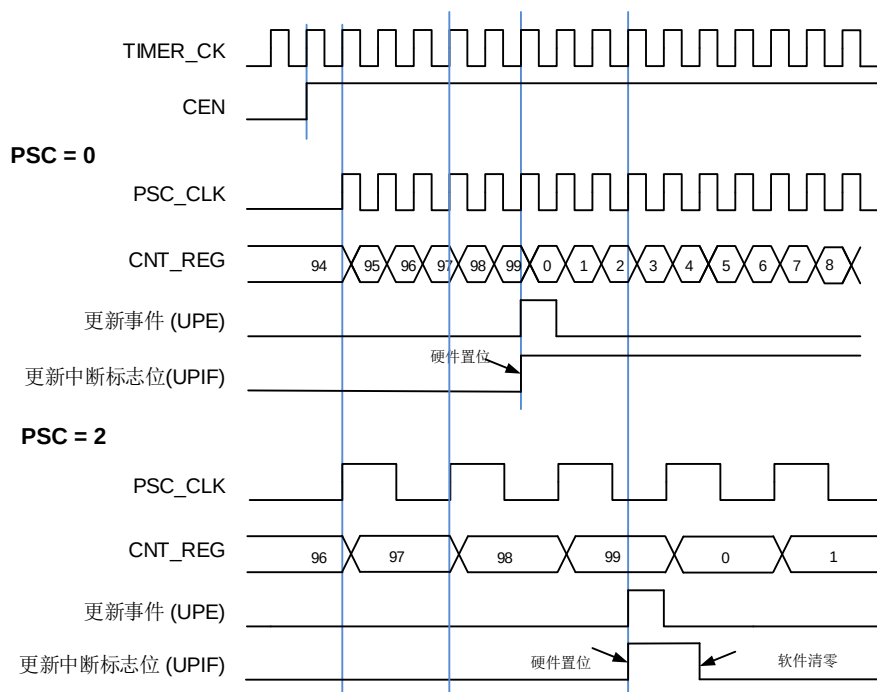
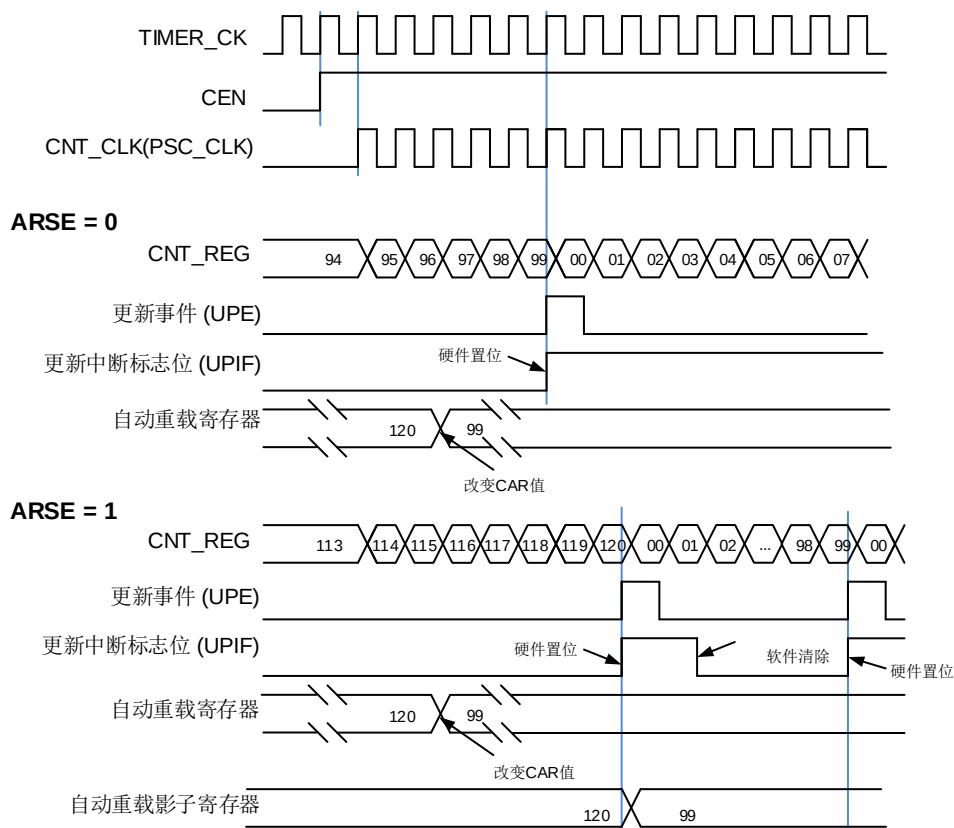


图 12-107. 向上计数时序图，在运行时改变 **TIMERx_CAR** 寄存器的值 (**TIMERx, x=5,6**)

定时器调试模式

当 Cortex®-M33 内核停止，DBG_CTL0 寄存器中的 **TIMERx_HOLD** 配置位被置 1，定时器计数器停止。

12.4.5. TIMERx 寄存器 (x=5/6)

TIMER5 基地址: 0x4000 1000

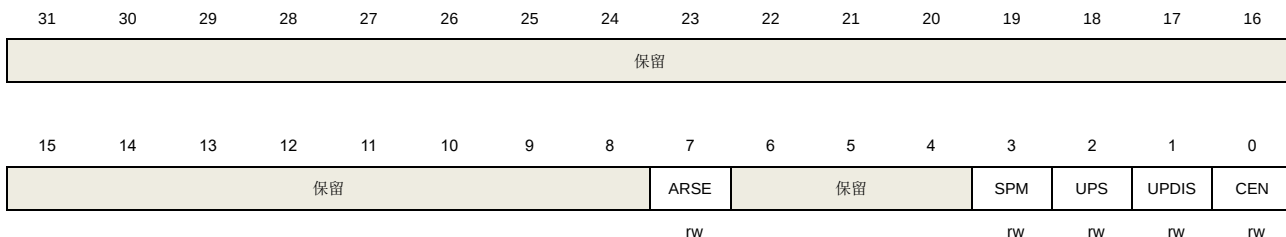
TIMER6 基地址: 0x4000 1400

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。



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: 更新事件发生后, 计数器继续计数 1: 在下次更新事件发生时, 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 – 计数器上溢 – 从模式控制器产生的更新 1: 使能后只有计数器上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

- 计数器上溢
 - 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

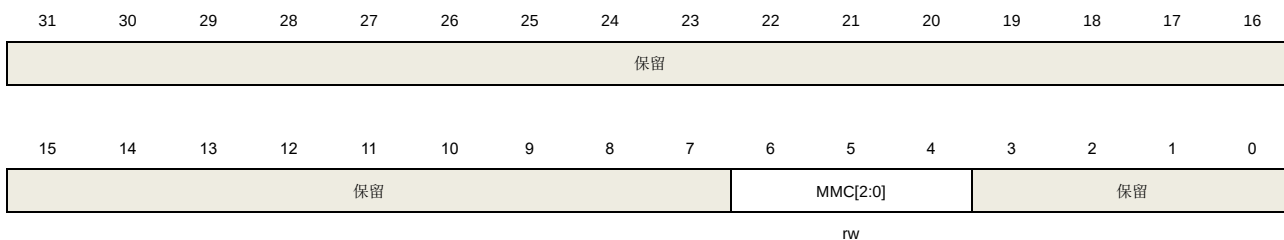
1: 计数器使能

在软件将 **CEN** 位置 1 后，外部时钟、暂停模式和译码器模式才能工作。触发模式可以自动地通过硬件设置 **CEN** 位。

控制寄存器 1 (TIMERx_CTL1)

地址偏移: 0x04
复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



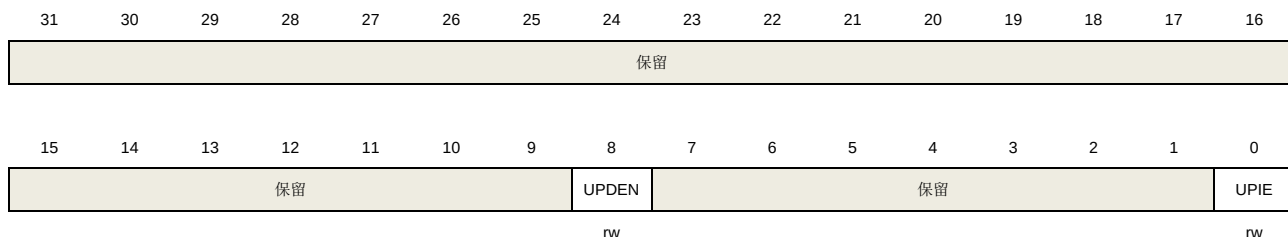
位/位域	名称	描述
31:7	保留	必须保持复位值
6:4	MMC[2:0]	这些位控制 TRGO 信号的选择，TRGO 信号由主定时器发给从定时器用于同步功能 000: 复位。TIMERx_SWEVG 寄存器的 UPG 位被置 1 或从模式控制器产生复位触发一次 TRGO 脉冲，后一种情况下，TRGO 上的信号相对实际的复位会有一个延迟。 001: 使能。此模式可用于同时启动多个定时器或控制在一段时间内使能从定时器。主模式控制器选择计数器使能信号作为触发输出 TRGO。当 CEN 控制位被置 1 或者暂停模式下触发输入为高电平时，计数器使能信号被置 1。在暂停模式下，计数器使能信号受控于触发输入，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。 100~111: 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。



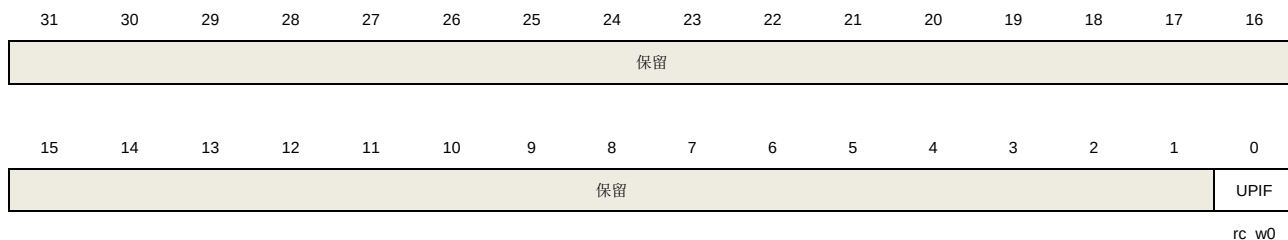
位/位域	名称	描述
31:9	保留	必须保持复位值
8	UPDEN	更新 DMA 请求使能 0: 禁止更新 DMA 请求 1: 使能更新 DMA 请求
7:1	保留	必须保持复位值
0	UPIE	更新中断使能 0: 禁止更新中断 1: 使能更新中断

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。



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 此位在任何更新事件发生时由硬件置 1，软件清 0。

0: 无更新中断发生

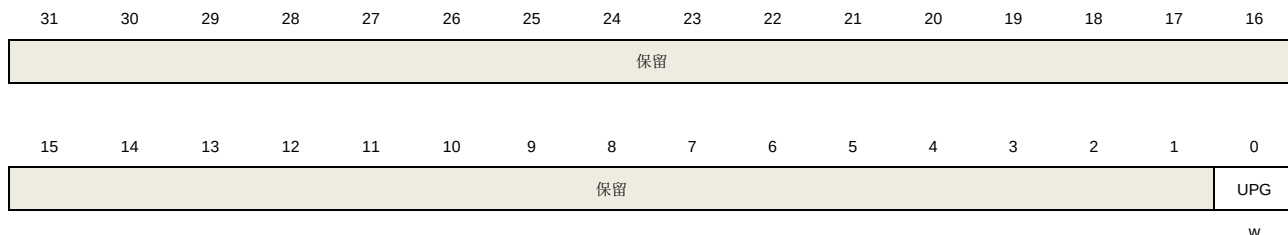
1: 发生更新中断

软件事件产生寄存器 (TIMERx_SWEVG)

地址偏移: 0x14

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



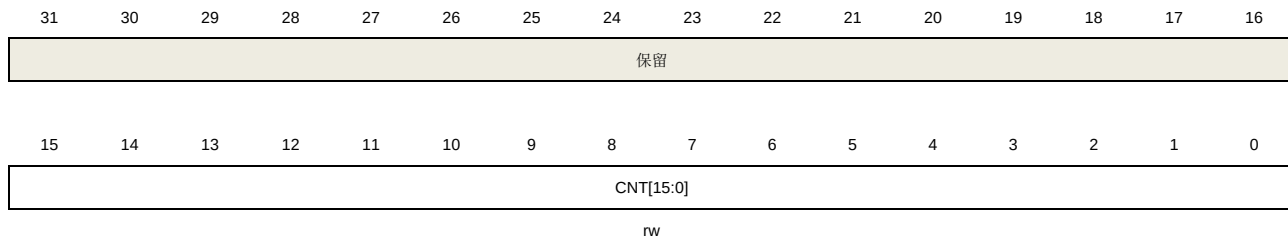
位/位域	名称	描述
31:1	保留	必须保持复位值
0	UPG	更新事件产生 此位由软件置 1, 被硬件自动清 0。当此位被置 1 并且向上计数模式, 计数器被清 0, 预分频计数器将同时被清除。 0: 无更新事件产生 1: 产生更新事件

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。



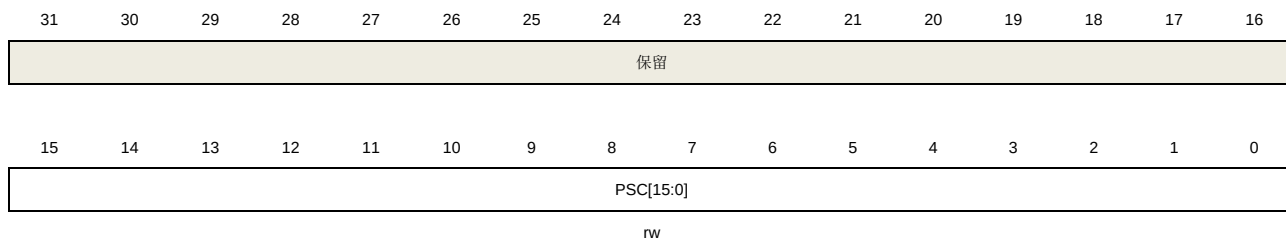
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

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。



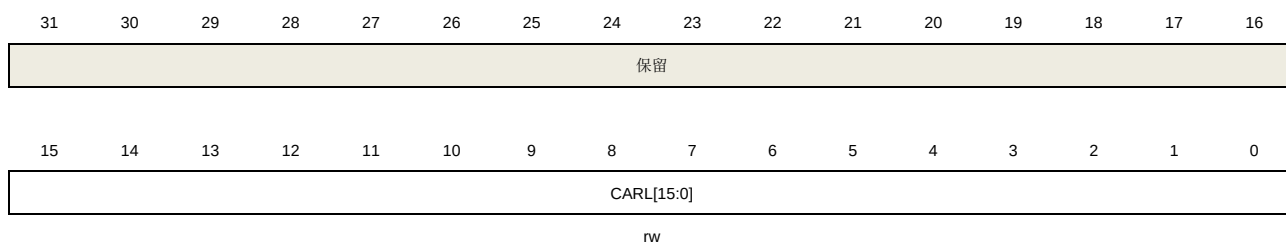
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 PSC 时钟除以 (PSC+1)，每次当更新事件产生时，PSC 的值被装入当前预分频寄存器。

计数器自动重载寄存器 (TIMERx_CAR)

地址偏移: 0x2C

复位值: 0x0000 FFFF

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	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。

13. 实时时钟（RTC）

13.1. 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 VDD 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。

13.2. 主要特性

- 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 2^{20}
- 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
 - AHB 时钟除以 10
- 可屏蔽的中断源：
 - 闹钟中断
 - 秒中断
 - 溢出中断

13.3. 功能描述

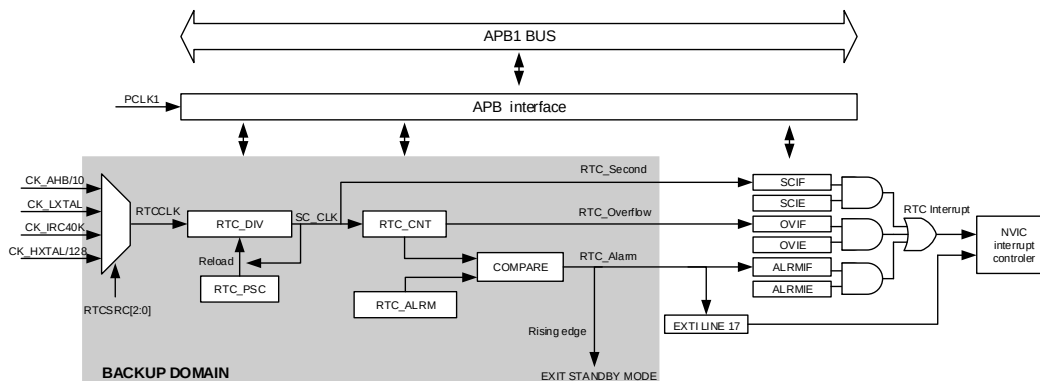
RTC 电路包含两个主要部分，位于 PCLK1 时钟域的 APB 接口和位于 RTC 时钟域的 RTC 内核。

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位当前系统时间。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 13-1. RTC 框图



13.3.1. 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 BKPEN 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。

13.3.2. 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。

13.3.3. 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 RTCCLK 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. 通过将 RTC_CTL 寄存器中的 CMF 置位来进入配置模式；
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。

13.3.4. RTC 标志位

RTC 秒中断标志（SCIF）在 RTC 计数器更新之前的每个 RTCCLK 周期置位。

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任一序列进行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 13-2. RTC 秒信号及闹钟信号的波形 (RTC_PSC = 3,RTC_ ALRM = 2)

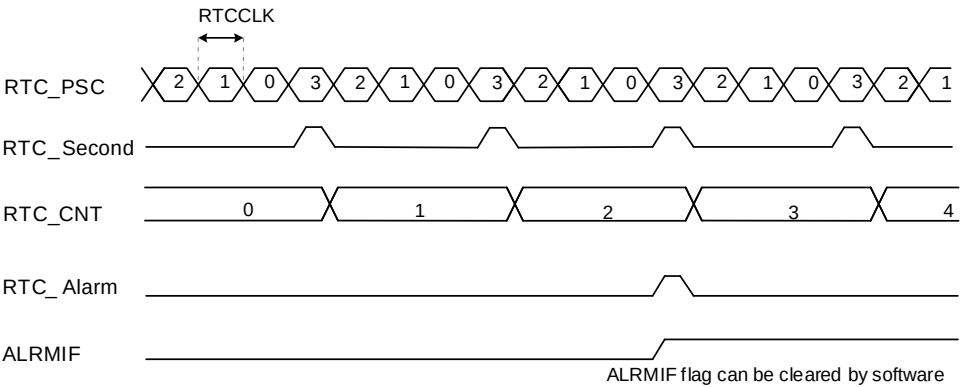
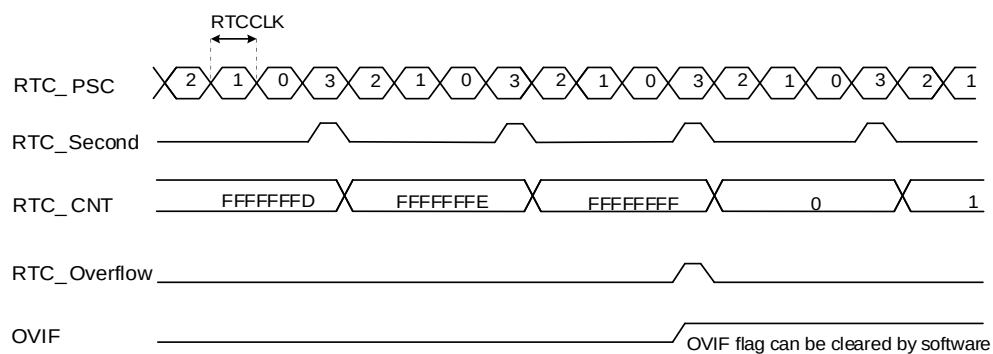


图 13-3. RTC 秒信号及溢出信号的波形(RTC_PSC = 3)



13.4. RTC 寄存器

RTC 基地址：0x4000 2800

13.4.1. RTC 中断使能寄存器（RTC_INTEN）

偏移地址：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
保留													OVIE	ALRMIE	SCIE
													rw	rw	rw

位/位域	名称	描述
31:3	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

13.4.2. RTC 控制寄存器（RTC_CTL）

偏移地址：0x04

复位值：0x0000 0020

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										LWOF	CMF	RSYNF	OVIF	ALRMIF	SCIF

r rw rc_w0 rc_w0 rc_w0 rc_w0

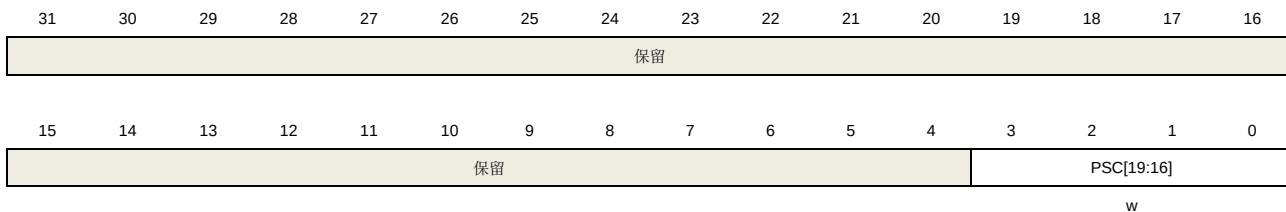
位/位域	名称	描述
31:6	保留	必须保持复位值
5	LWOFF	上次对 RTC 寄存器写操作标志 0: 上次对 RTC 寄存器写操作没有完成 1: 上次对 RTC 寄存器写操作已经完成
4	CMF	配置模式标志 0: 退出配置模式 1: 进入配置模式
3	RSYNF	寄存器同步标志 0: 寄存器没有与 APB1 时钟同步 1: 寄存器已经与 APB1 时钟同步
2	OVIF	溢出中断标志 0: 没有检测到溢出事件 1: 检测到溢出事件。当 RTC_INTEN 寄存器的 OVIE 位被置 1，中断发生。
1	ALRMIF	闹钟中断标志 0: 没有检测到闹钟事件 1: 检测到闹钟事件。当 RTC_INTEN 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。

13.4.3. RTC 预分频寄存器高位 (RTC_PSCH)

偏移地址: 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

31:4	保留	必须保持复位值。
3:0	PSC[19:16]	RTC 预分频器高位值

13.4.4. RTC 预分频寄存器低位 (RTC_PSCL)

偏移地址：0x0C

复位值：0x0000 8000

该寄存器可以按半字（16 位）或字（32 位）访问。



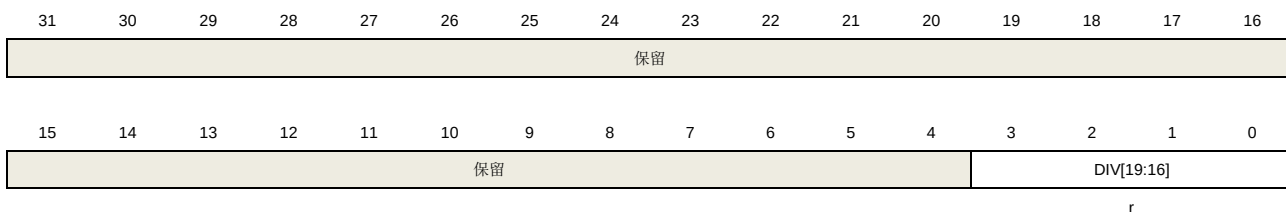
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	PSC[15:0]	RTC 预分频器低位值 SC_CLK 的频率是 RTCCLK 的频率除以(PSC[19:0]+1).

13.4.5. RTC 分频器高位 (RTC_DIVH)

偏移地址：0x10

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
31:4	保留	必须保持复位值。
3:0	DIV[19:16]	RTC 分频器高位

13.4.6. RTC 分频器低位 (RTC_DIVL)

偏移地址：0x14

复位值：0x0000 8000

该寄存器可以按半字（16 位）或字（32 位）访问。



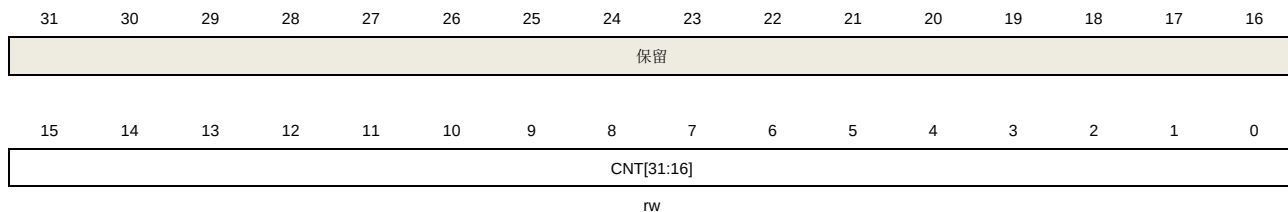
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	DIV[15:0]	RTC 分频器低位 当 RTC 预分频寄存器或者 RTC 计数寄存器更新时，RTC 分频器寄存器会由硬件自动加载。

13.4.7. RTC 计数寄存器高位（RTC_CNTH）

偏移地址：0x18

复位值：0x0000 0000

该寄存器可以按半字（16 位）或字（32 位）访问。



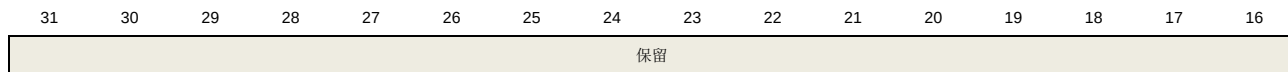
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CNT[31:16]	RTC 计数寄存器高位。

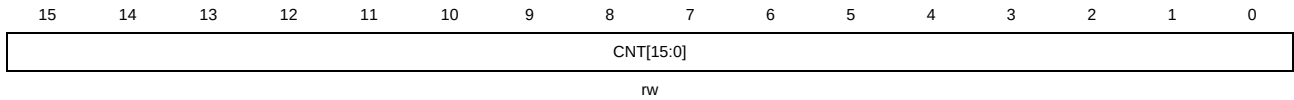
13.4.8. RTC 计数寄存器低位（RTC_CNTL）

偏移地址：0x1C

复位值：0x0000 0000

该寄存器可以按半字（16 位）或字（32 位）访问。





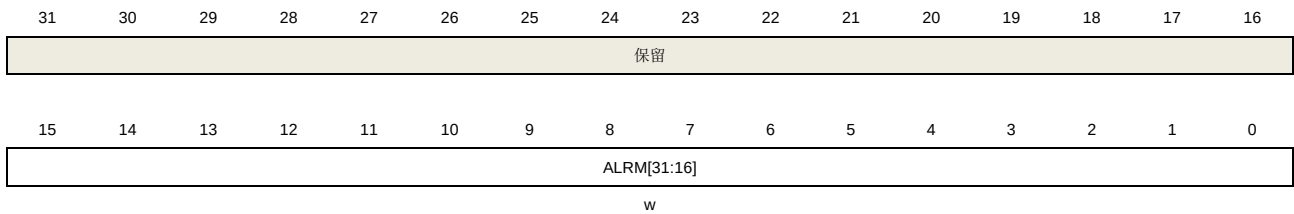
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CNT[15:0]	RTC 计数寄存器低位。

13.4.9. RTC 闹钟寄存器高位 (RTC_ALRMH)

偏移地址：0x20

复位值：0x0000 FFFF

该寄存器可以按半字（16 位）或字（32 位）访问。



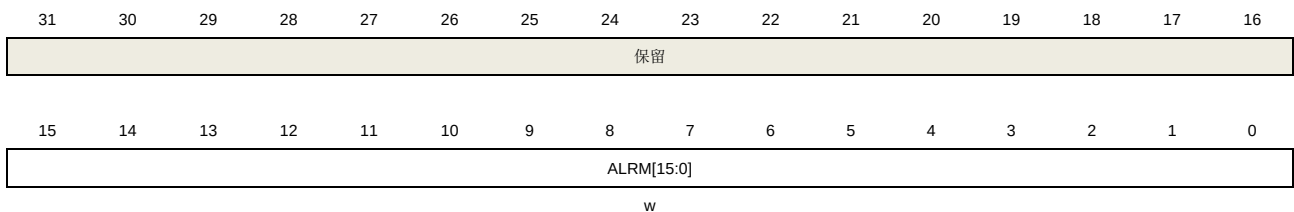
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	ALRM[31:16]	RTC 闹钟值高位。

13.4.10. RTC 闹钟寄存器低位 (RTC_ALRML)

偏移地址：0x24

复位值：0x0000 FFFF

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 ALRM[15:0] RTC 闹钟值低位。

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预设门限的时候，会触发一个复位。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）有独立时钟源（IRC40K）。即使主时钟失效，FWDGT 依然能保持正常工作状态，适用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系统复位。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。

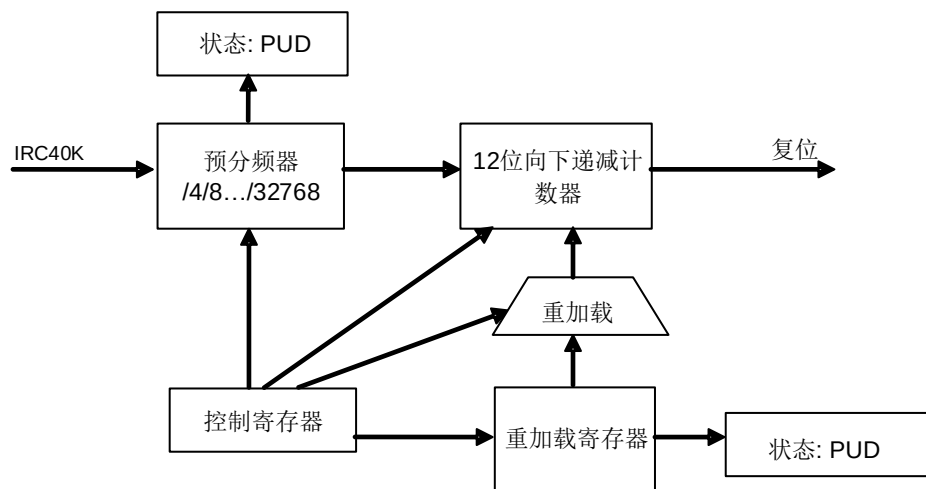
14.1.2. 主要特征

- 独立运行的12位向下计数器；
- 使能看门狗定时器，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，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。参考[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，产生一次系统复位。

在任何时候向 FWDGT_CTL 中写 0xAAAA 都可以重装载计数器，重装载值来源于重装载寄存器（FWDGT_RLD）。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产生系统复位。

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。为了避免系统复位，软件应该在计数器达到 0x000 之前重装载计数器。

预分频寄存器（FWDGT_PSC）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 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 FWDGT_PSC 或者 FWDGT_RLD 更新时，FWDGT_STAT 寄存器的状态位被置 1。

如果 DBG 中控制寄存器 0（DBG_CTL0）中 FWDGT_HOLD 位被清 0，即使 Cortex®-M33 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 位置 1，独立看门狗定时器将在 Cortex®-M33 内核停止时（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[3:0] 位	最小超时(ms) RLD[11:0]=0x000	最大超时(ms) RLD[11:0]=0xFFFF
1 / 4	0000	0.1	409.6
1 / 8	0001	0.2	819.2
1 / 16	0010	0.4	1638.4
1 / 32	0011	0.8	3276.8
1 / 64	0100	1.6	6553.6
1 / 128	0101	3.2	13107.2
1 / 256	0110	6.4	26214.4
1 / 512	0111	12.8	52428.8

预分频系数	PSC[3:0] 位	最小超时(ms) RLD[11:0]=0x000	最大超时(ms) RLD[11:0]=0xFFFF
1 / 1024	1000	25.6	104857.6
1 / 2048	1001	51.2	209715.2
1 / 4096	1010	102.4	419430.4
1 / 8192	1011	204.8	838860.8
1 / 16384	1100	409.6	1677721.6
1 / 32768	1101~1111	819.2	3355443.2

通过校准 IRC40K 可以使自由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

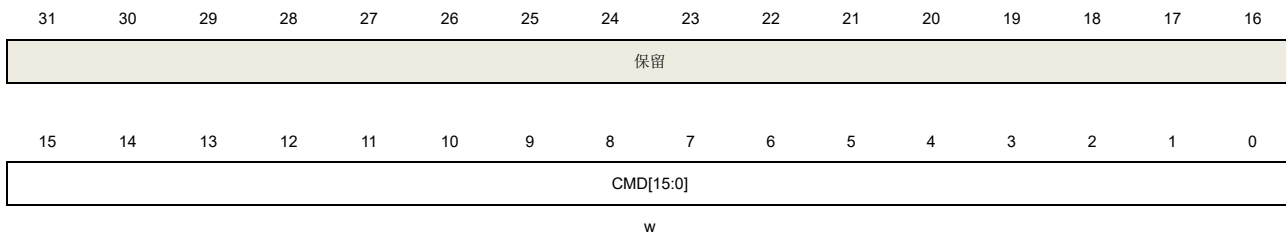
FWDGT 基地址：0x4000 3000

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。



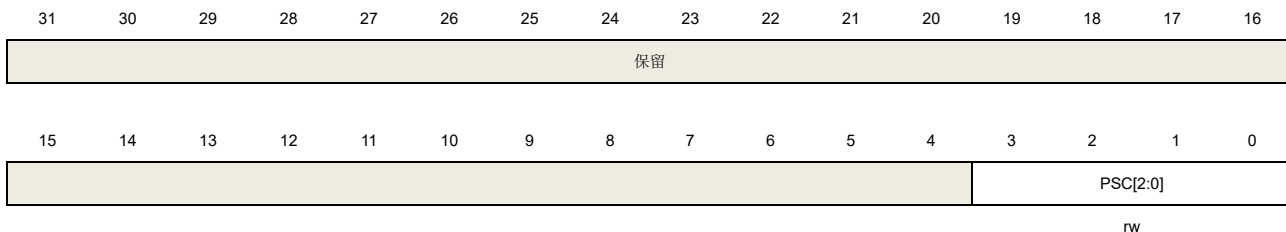
位/位域	名称	说明
31:16	保留	必须保持复位值。
15:0	CMD[15:0]	只可写，写入不同的值来产生不同的功能 0x5555: 关闭FWDGT_PSC和FWDGT_RLD的写保护 0xCCCC: 开启独立看门狗定时器定时计数器。计数减到0时产生复位 0xAAAA: 重装载计数器

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。



位/位域	名称	说明
31:4	保留	必须保持复位值。
3:0	PSC[3:0]	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。

0000: 1 / 4
0001: 1 / 8
0010: 1 / 16
0011: 1 / 32
0100: 1 / 64
.....
1100: 1 / 16384
1101~1111: 1 / 32768

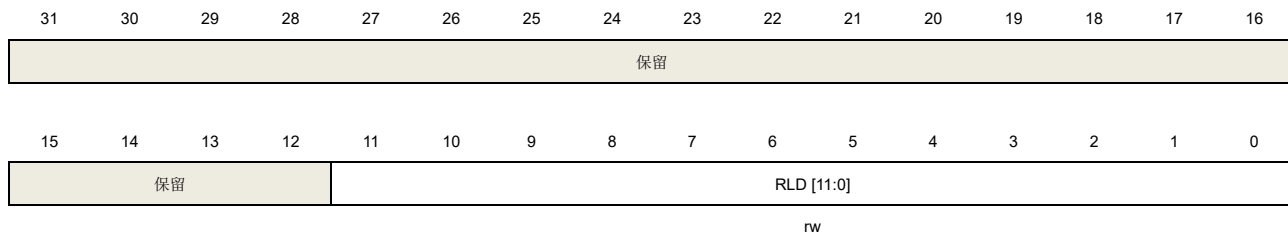
如果应用需要使用不同的预分频系数，改变预分频值之前必须等到PUD位被清0。更新了预分频寄存器中的值后，在代码持续执行之前不必等待PUD值被清零（在进入省电模式前需要等待PUD值清零）。

重载寄存器（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。



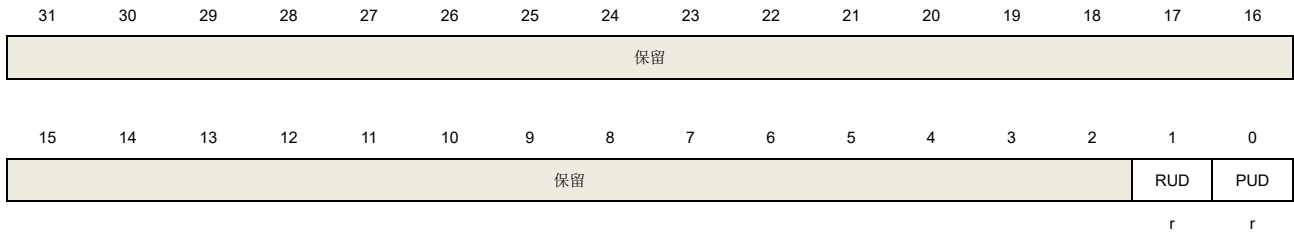
位/位域	名称	说明
31:12	保留	必须保持复位值。
11:0	RLD[11:0]	独立看门狗定时器定时计数器重载值，向FWDGT_CTL寄存器写入0xAAAA的时候，这个值会被更新到看门狗定时器计数器中。 这些位有写保护功能。在写这些位之前需向FWDGT_CTL寄存器中写0x5555。在改写这个寄存器的过程中，FWDGT_STAT寄存器的RUD位被置1，从此寄存器中读取的任何值都是无效的。 如果应用需要使用不同的重载值，改变重加载值之前必须等到RUD位被清0。更新了重加载寄存器的值后，在代码持续执行之前不必等待RUD值被清零（在进入省电模式前需要等待RUD值清零）。

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。



位/位域	名称	说明
31:2	保留	必须保持复位值。
1	RUD	独立看门狗定时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 位向下递减计数器值逐渐减小。计数值达到 0x3F 时会产生系统复位（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 会产生一个提前唤醒标志，如果使能中断将会产生中断。

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。

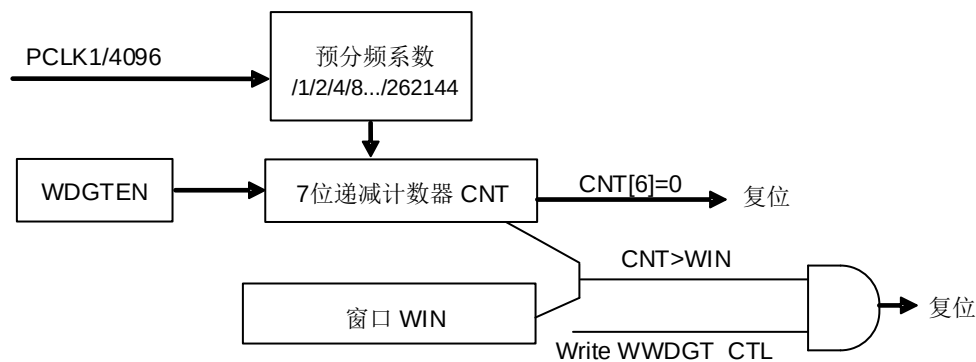
14.2.2. 主要特征

- 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）：看门狗定时器打开，中断使能，当计数值达到0x40时将会产生中断。
- 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 WDG TEN 位置 1），计数值达到 0x3F 的时候产生系统复位（CNT[6]位被清 0）。或是在计数值达到窗口寄存器值之前，更新计数器也会产生系统复位。

图 14-2. 窗口看门狗定时器框图



上电复位之后窗口看门狗定时器总是关闭的。软件可以向 WWDGT_CTL 的 WDG TEN 写 1 开启窗口看门狗定时器。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

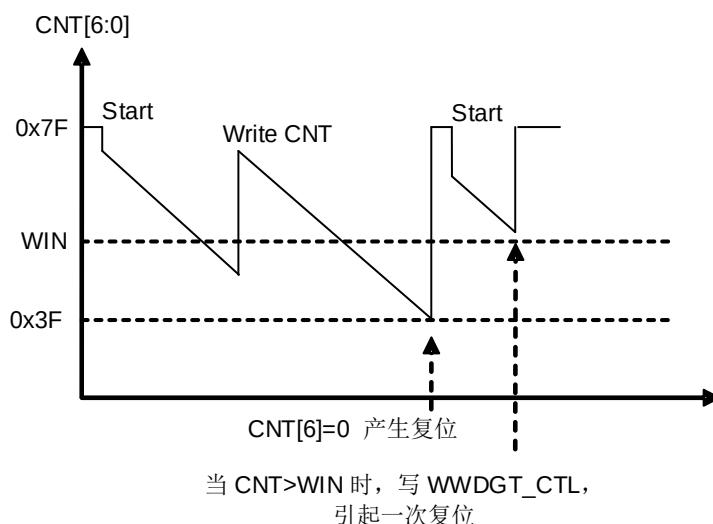
0x3F，也就是说 CNT[6]位应该被置 1。CNT[5:0]决定了两次重载之间的最大间隔时间。计数器的递减速度取决于 APB1 时钟和预分频器（WWDGT_CFG 寄存器的 PSC[1:0]位）。

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，重载向下计数器可以避免复位，否则在其他时候进行重载就会引起复位。

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 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：

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

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 140MHz \(f_{PCLK1}\) 时的最大/最小超时值](#)。

表 14-2. 在 140MHz (f_{PCLK1}) 时的最大/最小超时值

预分频系数	PSC[4:0]	最小超时 CNT[6:0] = 0x40	最大超时 CNT[6:0] = 0x7F
1 / 1	00000	0.02926ms	1.872ms

预分频系数	PSC[4:0]	最小超时 CNT[6:0] = 0x40	最大超时 CNT[6:0] = 0x7F
1 / 2	00001	0.05851ms	3.745ms
1 / 4	00010	0.11703ms	7.49ms
1 / 8	00011	0.23406ms	14.98ms
1 / 16	00100	0.46811ms	29.96ms
1 / 32	00101	0.93623ms	59.92ms
1 / 64	00110	1.87ms	119.84ms
1 / 128	00111	3.75ms	239.68ms
1 / 256	01000	7.49ms	479.36ms
1 / 512	01001	14.98ms	958.72ms
1 / 1024	01010	29.96ms	1917.44ms
1 / 2048	01011	59.92ms	3834.88ms
1 / 4096	01100	119.84ms	7669.76ms
1 / 8192	01101	239.67ms	15339.52ms
1 / 16384	01110	479.35ms	30679.04ms
1 / 32768	01111	958.70ms	61358.08ms
1 / 65536	10000	1917.40ms	122716.16ms
1 / 131072	10001	3834.79ms	245432.32ms
1 / 262144	10010~11111	7669.58ms	490864.64ms

如果 MCU 调试模块中的 WWDGT_HOLD 位被清 0，即使 Cortex®-M33 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

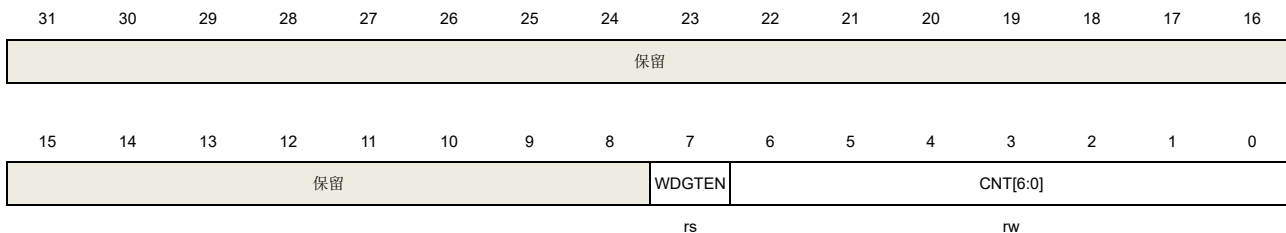
WWDGT 基地址：0x4000 2C00

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。



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

配置寄存器（WWDGT_CFG）

地址偏移：0x04

复位值：0x0000 007F

该寄存器可以按半字（16 位）或字（32 位）访问。



位/位域	名称	说明
31:13	保留	必须保持复位值。
12:10	PSC[4:2]	预分频器位值高。PSC[4:0]由高位值PSC[4:2]和低位值PSC[1:0]组成。PSC[4:0]为看

门狗计数器的时基:

00000: (PCLK1 / 4096) / 1

00001: (PCLK1 / 4096) / 2

00010: (PCLK1 / 4096) / 4

00011: (PCLK1 / 4096) / 8

.....

10001: (PCLK1 / 4096) / 131072

10010~11111: (PCLK1 / 4096) / 262144

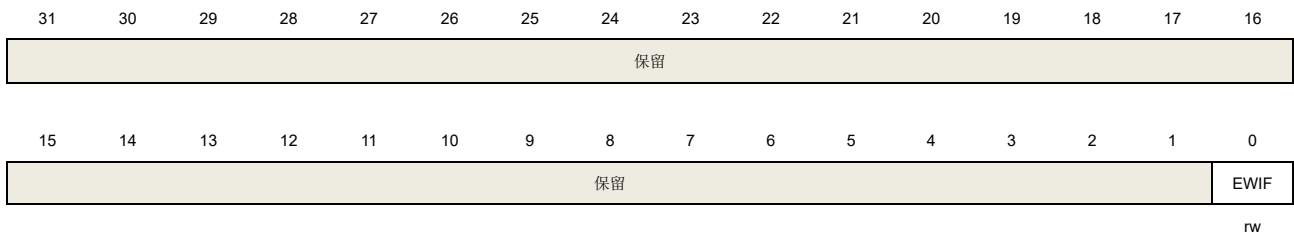
9	EWIE	提前唤醒中断使能。如果该位被置1，计数值达到0x40时触发中断。该位由硬件复位清0，或通过RCU模块的WWDGT软件复位来清0。写0没有任何作用。
8:7	PSC[1:0]	预分频器低位，看门狗定时器计数器的时间基准。PSC[4:0]由高位值PSC[4:2]和低位值PSC[1:0]组成。
6:0	WIN[6:0]	窗口值，当看门狗定时器计数器的值大于窗口值时，写看门狗定时器计数器（WWDGT_CTL的CNT位）会产生复位。

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	说明
31:1	保留	必须保持复位值。
0	EWIF	提前唤醒中断标志位。当计数值达到0x40，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. 循环冗余校验计算单元（CRC）

15.1. 简介

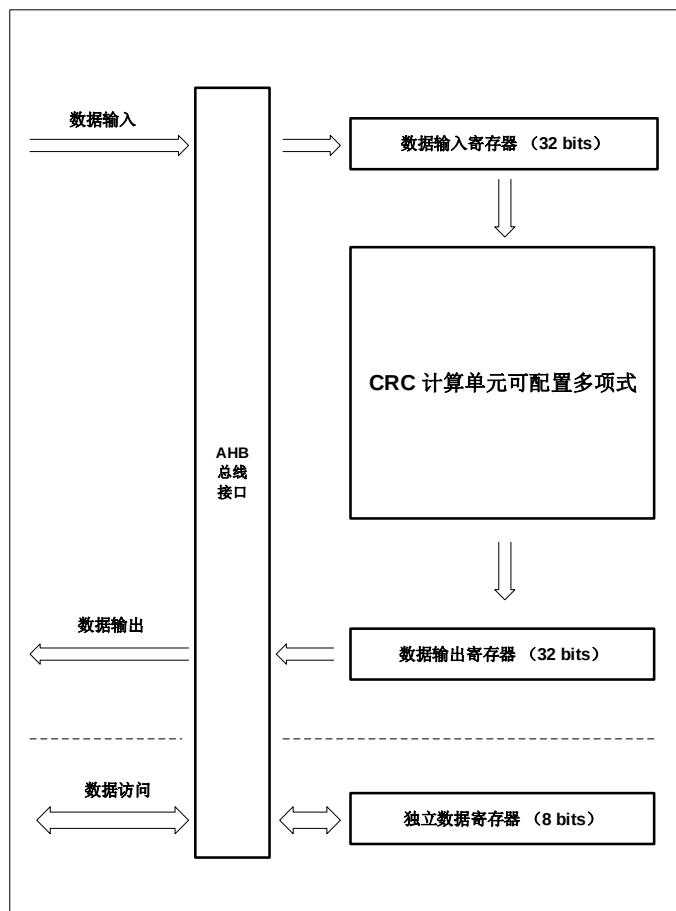
循环冗余校验码是一种用在数字网络和存储设备上的差错校验码，可以校验原始数据的偶然误差。

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

15.2. 主要特性

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

图 15-1. CRC 计算单元框图



15.3. 功能描述

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

如果不通过软件设置 CRC_CTL 寄存器的方式来清除 CRC_DATA 寄存器，新输入的原始数据将会基于前一次 CRC_DATA 寄存器中的结果进行计算。

对于 32 / 16 / 8 位的数据长度，CRC 的计算分别要花费 4 / 2 / 1 个 AHB 的时钟周期。在此期间，因为 32 位输入缓存的原因，AHB 总线将不会被挂起。

- 此模块提供了一个8位的独立寄存器CRC_FDATA，CRC_FDATA与CRC计算无关，任何时候都可以进行独立的读写操作。
- 逆序功能可以交换输入输出数据的位序。

输入数据可选择三种逆序形式。

以原始数据 0x3456CDEF 为例：

1) 按字节逆序：

32 位数据被分成四组，组内完成颠倒。逆序后的数据为：0x2C6AB3F7

2) 按半字逆序：

32 位数据被分成两组，组内完成颠倒。逆序后的数据为：0x6A2CF7B3

3) 按字逆序：

32 位数据被分成一组，组内完成颠倒，逆序后的数据为：0xF7B36A2C

对于输出数据来说，逆序形式为按字逆序。

例如：当 REV_O=1，计算结果 0x3344CCDD 将被逆序成 0xBB3322CC。

- 用户可配置的初始计算数据。

当 RST 位置位或对 CRC_IDATA 寄存器进行写操作时，CRC_DATA 寄存器将自动初始化为 CRC_IDATA 寄存器中的值。

- 用户配置多项式。

通过配置 PS[1:0]，用户可以选择有效多项式和输出数据位宽。当 PS[1:0]或多项式改变后，需要复位 CRC。

15.4. CRC 寄存器

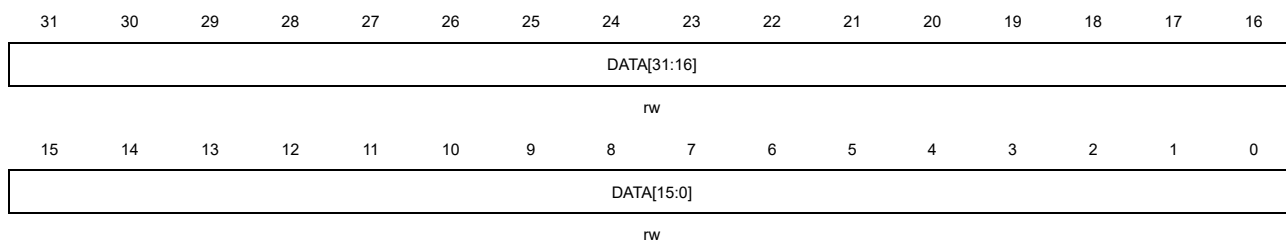
CRC 基地址：0x4002 3000

15.4.1. 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。



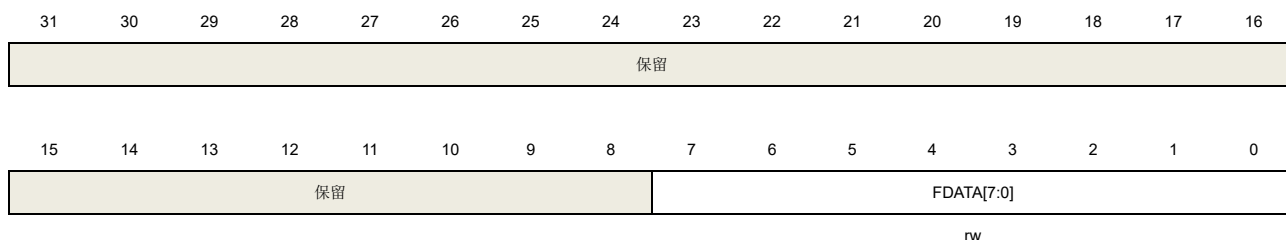
位/位域	名称	描述
31:0	DATA[31:0]	CRC 计算结果位 软件可读可写。 该寄存器用于接收待计算的新数据，直接将其写入即可。刚写入的数据不能被读出来因为读取该寄存器得到的是上次 CRC 计算的结果。

15.4.2. 独立数据寄存器（CRC_FDATA）

地址偏移：0x04

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	FDATA[7:0]	独立数据寄存器位 软件可读可写。

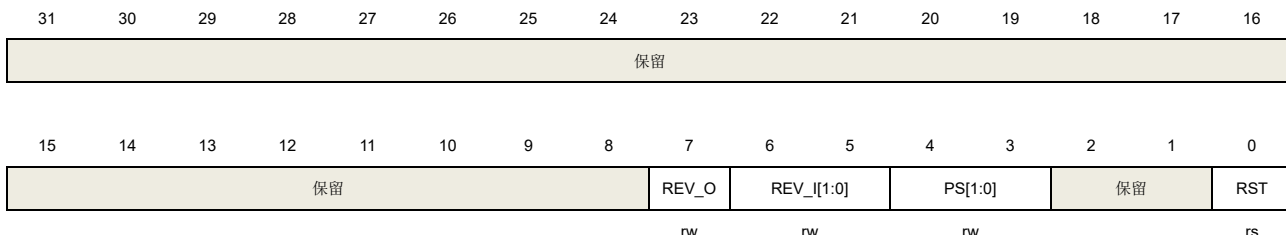
这些位与 CRC 计算无关。该字节能被任何其他外设用于其他任何目的。该字节不受 CRC_CTL 寄存器的影响。

15.4.3. 控制寄存器（CRC_CTL）

地址偏移：0x08

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



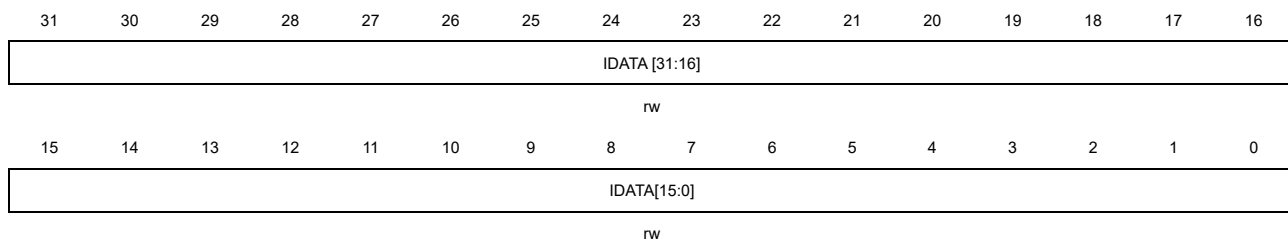
位/位域	名称	描述
31:8	保留	必须保持复位值。
7	REV_O	按位顺序翻转输出数据功能 0: 输出数据不翻转 1: 输出数据按位顺序翻转
6:5	REV_I[1:0]	翻转输入数据功能 0: 输入数据不翻转 1: 输入数据按字节翻转 2: 输入数据按半字翻转 3: 输入数据按字翻转
4:3	PS[1:0]	多项式长度 0: 32 位 1: 16（POLY[15:0]用于计数）位 2: 8（POLY[7:0]用于计数）位 3: 7（POLY[6:0]用于计数）位
2:1	保留	必须保持复位值。
0	RST	软件可读写 该位用来复位 CRC_DATA 寄存器。 置位时，CRC_DATA 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

15.4.4. 初值寄存器（CRC_IDATA）

地址偏移：0x10

复位值：0xFFFF FFFF

该寄存器只能按字（32 位）访问。



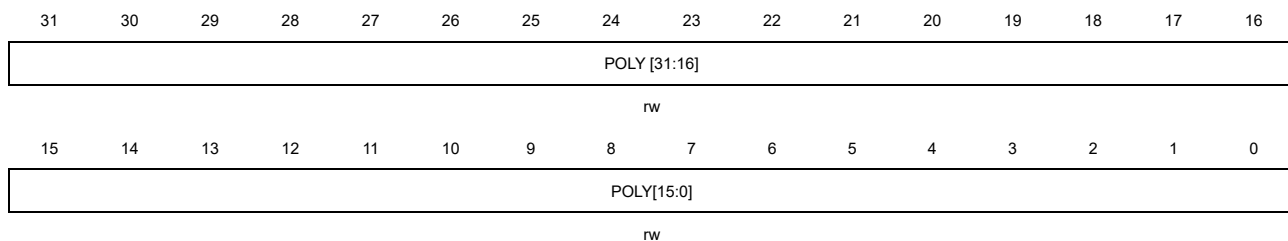
位/位域	名称	描述
31:0	IDATA[31:0]	配置 CRC 初值 CRC_CTL 寄存器的 RST 位置位后，CRC_DATA 寄存器的值将被更新为此寄存器的值。

15.4.5. 多项式寄存器（CRC_POLY）

地址偏移：0x14

复位值：0x04C1 1DB7

该寄存器只能按字（32 位）访问。



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

16. 调试 (DBG)

16.1. 简介

GD32F50x 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®-M33 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®-M33技术参考手册;
- Arm®调试接口v5结构规范。

调试系统帮助调试者在低功耗模式下调试或者一些外设调试。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、WWDGT、FWDGT、I2C 和 CAN。

16.2. JTAG/SW 功能描述

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。

16.2.1. 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 = 1110011110011110 (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 = 1110011110011110 (0xE73C LSB)信号;
- 发送50个以上TCK周期的TMS=1信号。

16.2.2. 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）。

表 16-1. 调试引脚分配

引脚	调试接口
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[通用和备用输入/输出接口\(GPIO 和 AFIO\)](#)。

16.2.3. JTAG 链状结构

Cortex®-M33 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®-M33 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®-M33 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。

16.2.4. 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®-M33 的绝大部分组件，除了 NVIC，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。

16.2.5. JEDEC-106 ID code

Cortex®-M33 集成了 JEDEC-106 ID 代码。位于 ROM 表中，映射地址为 0xE00FF000_0xE00FFFFF。

16.3. 调试保持功能描述

16.3.1. 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。

16.3.2. 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:

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。

16.4. DBG 寄存器

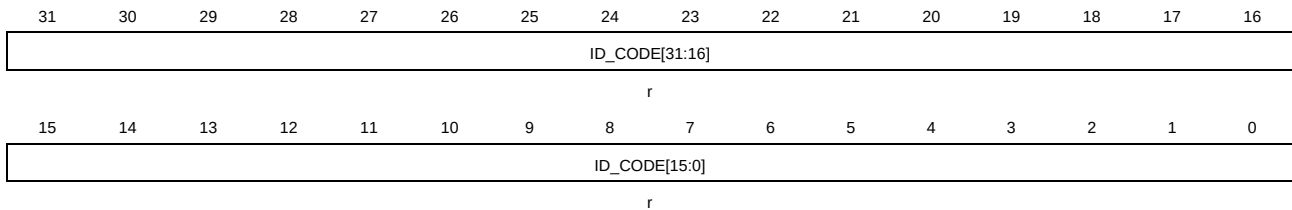
DEBUG 基地址: 0xE004 5000

16.4.1. ID 寄存器 (DBG_ID)

地址偏移: 0x00

只读寄存器

该寄存器只能按字(32 位)访问



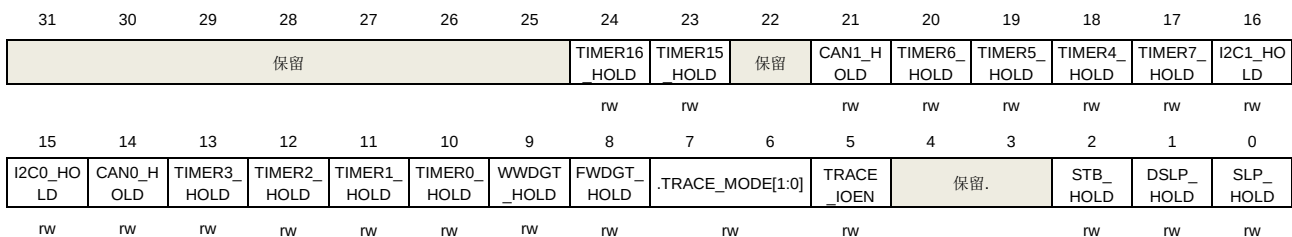
位/位域	名称	描述
31:0	ID_CODE[31:0]	DBG ID 寄存器 这些位由软件读取，这些位是不变的常数

16.4.2. 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



位/位域	名称	描述
31:25	保留	必须保持复位值。
24	TIMER16_HOLD	TIMER16 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 16 计数器不变，用于调试

23	TIMER15_HOLD	TIMER15 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 15 计数器不变，用于调试
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	TIMER6_HOLD	TIMER6 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
19	TIMER5_HOLD	TIMER5 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
18	TIMER4_HOLD	TIMER4 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
17	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
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 SMBUS 状态不变，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 SMBUS 状态不变，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 该位域由软件置位和复位 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

17. 真随机数生成器（TRNG）

17.1. 简介

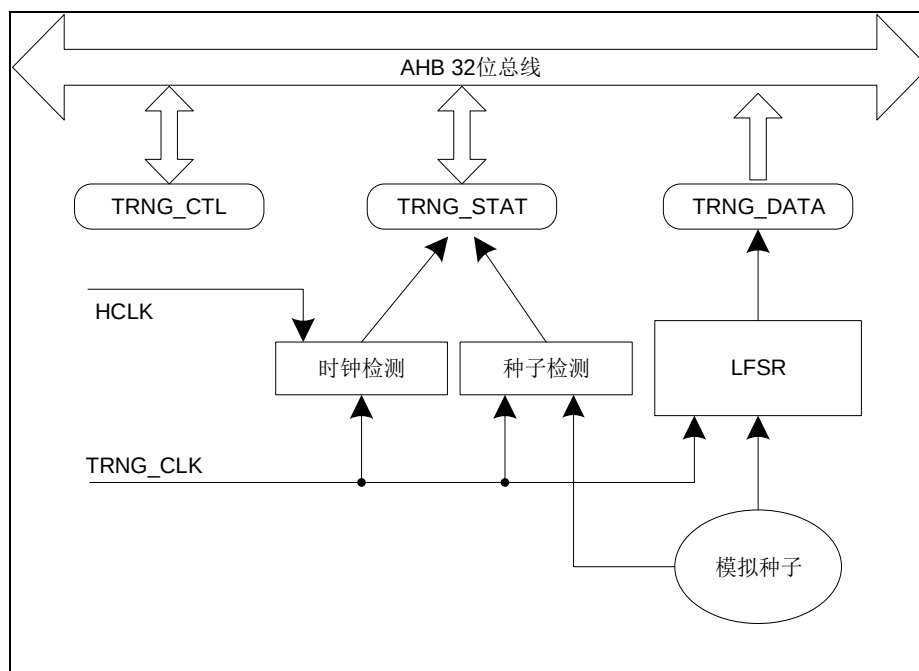
真随机数发生器模块（TRNG）能够通过连续模拟噪声生成一个 32 位的随机数值。

17.2. 主要特征

- 两个连续随机数的间隔大约为40个TRNG_CLK时钟周期；
- 32位随机数的种子是由模拟噪声产生的，因此该随机数是一个真随机数值。

17.3. 功能说明

图 17-1. TRNG 模块框图



随机数种子由模拟电路实现。模拟种子信号输出到一个线性反馈移位寄存器（LFSR）之后在该寄存器中转化成 32 位宽度的随机数。

该模拟种子由几个环形振荡器的输出生成。LFSR 由可配置的 TRNG_CLK 时钟（参考 [RCU](#) 相关章节）驱动，因此随机数质量仅与 TRNG_CLK 时钟有关，与 HCLK 频率无关。

当有足够数量的种子被输入 LFSR 之后，LFSR 会输出 32 位数据到 TRNG_DATA 寄存器。

同时，系统会监视模拟种子和 TRNG_CLK 时钟。一旦模拟种子发生错误或者时钟产生错误，TRNG_STAT 寄存器的相关状态位将被置 1，如果 TRNG_CTL 寄存器的 TRNGIE 位同时被置 1 还将产生中断。

17.3.1. 操作流程

以下步骤为 TRNG 模块的推荐操作流程：

- 1) 根据需要使能中断，这样当随机数或错误产生时，将会触发一个中断；
- 2) 使能 IRC48M 时钟，或者选择 USBFSPSC[1:0]为 CK_PLL0，来使能 CK_TRNG 时钟；
- 3) 使能 TRNGEN 位；
- 4) 等待中断发生，检测 TRNG_STAT 寄存器，如果 SEIF = 0，CEIF = 0 并且 DRDY = 1 那么数据寄存器中的随机值可以被读取。

按照 FIPS PUB 140-2 的要求，数据寄存器中的第一个随机数需要保留而不是被使用。每一个新生成的随机数应当与之前的随机数相比较。只有当该随机数与前一个随机数不相等时，该数据才可被使用。

17.3.2. 错误标志

■ 时钟错误

当 TRNG_CLK 时钟频率低于 HCLK 频率的 1/16 时，CECS 和 CEIF 位将被置 1。此时，软件应当检查 TRNG_CLK 和 HCLK 时钟频率配置并清除 CEIF 位。时钟错误对上一个产生的随机数没有影响。

■ 种子错误

当模拟种子的值在 64 个 TRNG_CLK 时钟周期内不发生变化或连续不断的翻转，SECS 和 SEIF 位将被置位。此时，数据寄存器中的随机数值不应当被使用，并且软件需要清除 SEIF 位。之后将 TRNGEN 位清零并置 1 以便重新启动 TRNG 模块。

17.4. TRNG 寄存器

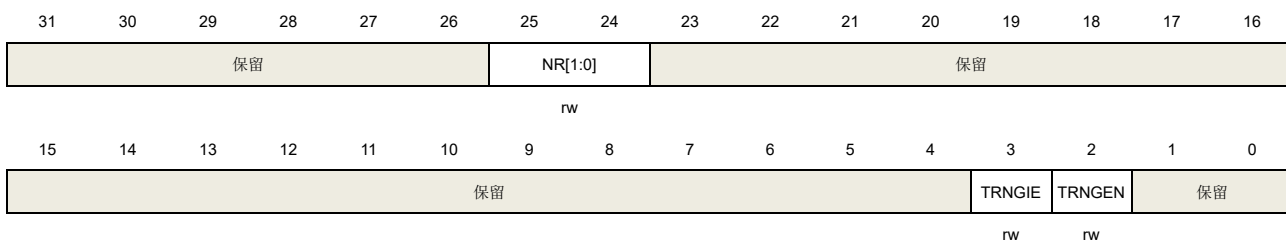
TRNG 基地址：0x4002 3C00

17.4.1. 控制寄存器（TRNG_CTL）

地址偏移：0x00

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。



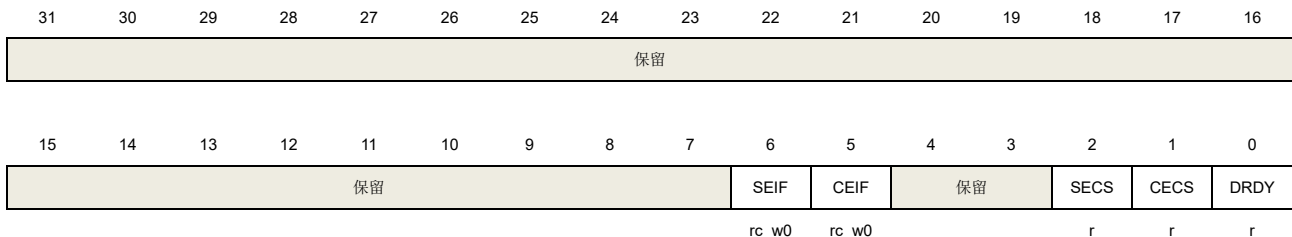
位/位域	名称	描述
31:26	保留	必须保持复位值。
25:24	NR[1:0]	模拟 rng 电源模式。 00: 超低；可能导致弱随机性 01: 低 10: 中等 11: 高
23:4	保留	必须保持复位值。
3	TRNGIE	中断使能位，当 DRDY，SEIF 或 CEIF 位被置位时该位控制生成一个中断。 0: 禁止 TRNG 中断 1: 使能 TRNG 中断
2	TRNGEN	TRNG 使能位 0: 禁止 TRNG 模块 1: 使能 TRNG 模块
1:0	保留	必须保持复位值。

17.4.2. 状态寄存器（TRNG_STAT）

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。



位/位域	名称	描述
31:7	保留	必须保持复位值。
6	SEIF	种子错误中断标志位 如果超过 64 个连续位具有相同值或超过 32 组连续交替的 0 和 1 被检测到则此位将置 1。 0: 未检测到错误 1: 检测到种子错误。写 0 将清除该位
5	CEIF	时钟错误中断标志位 如果 TRNG_CLK 时钟频率低于 HCLK 频率的 1 / 16 时该位被置位。 0: 未检测到错误 1: 检测到时钟错误。写 0 将清除该位
4:3	保留	必须保持复位值。
2	SECS	种子错误当前状态 0: 当前未检测到种子错误。如果 SEIF = 1 和 SECS = 0, 说明之前已经检测到种子错误但现在已恢复正常。 1: 当前检测到种子错误。如果超过 64 个连续位具有相同值或超过 32 组连续交替的 0 和 1 被检测到时, 该位置 1。
1	CECS	时钟错误当前状态 0: 当前未检测到时钟错误。如果 CEIF = 1 和 CECS = 0, 则意味着之前已检测到时钟错误但现在已恢复正常。 1: 当前检测到时钟错误。此时 TRNG_CLK 时钟频率低于 1 / 16 HCLK 频率。
0	DRDY	随机数准备状态位 读 TRNG_DATA 寄存器会清零该位, 当一个新的随机数产生时被置位。 0: TRNG 数据寄存器的内容无效 1: TRNG 数据寄存器的内容有效

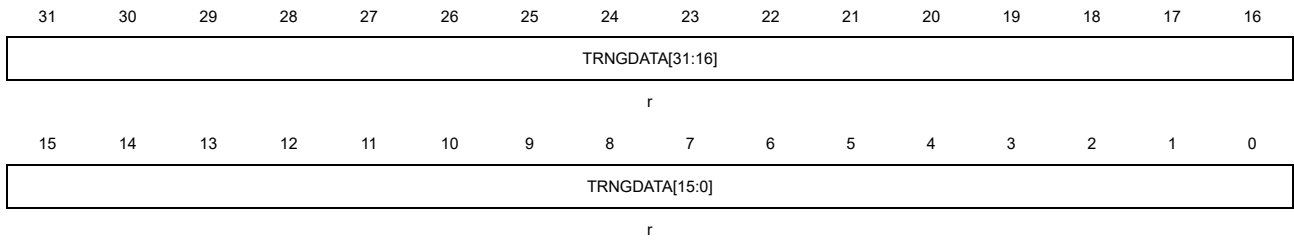
17.4.3. 数据寄存器 (TRNG_DATA)

地址偏移: 0x08

复位值: 0x0000 0000

在读此寄存器之前，软件必须确保 **DRDY** 位已置 1。

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:0	TRNGDATA[31:0]	32 位随机数据

18. 加密处理器（CAU）

18.1. 简介

加密处理单元支持处理 AES（128，192 或 256）算法，对数据进行加密或解密。加密处理器完全兼容下列标准：

- 联邦信息处理标准出版物（FIPS PUB 197，2001年11月26日）规定的高级加密标准（AES）。

CAU 处理器可在多种模式下使用多种长度密钥的 AES 算法执行数据加密和解密。

CAU 外设为 32 位 AHB 外设，它支持对输入 FIFO 和输出 FIFO 的 DMA 传输。

18.2. 主要特征

- 支持AES加密解密算法；
- 输入与输出FIFO支持DMA传输；
- 支持电子密码本（ECB）；
- 支持128位、192位或256位密钥；
- 输入和输出FIFO各8字深；
- 对于输入/输出FIFO的数据支持半字、字节、位交换或不交换；
- 数据可通过DMA或CPU中断进行传输，也可以不通过两者进行传输。

18.3. CAU 数据类型和初始化向量

18.3.1. 数据类型

CAU 处理器一次输入 32 位（字）数据，DES 每 64 位对数据流进行处理，AES 每 128 位对数据流进行处理。对于每个数据块，在其进入 CAU 处理器之前，可对这些数据执行位、字节、半字交换或不交换操作（取决于要加密的数据类型）。在 CAU 数据写入 OUT FIFO 之前，需要对其执行同样的交换操作。注意由于系统存储器结构采用小端模式，无论使用何种数据类型，最低有效数据均占用最低地址位置。

[图 18-1. DATAM 不交换/半字交换](#)和[图 18-2. DATATM 字节交换/位交换](#)介绍了 128 位 AES 块在不同数据类型下的数据交换。（对于 DES，数据块大小为 2 个 32 位字，请参考图中前两个字的数据交换）

图 18-1. DATAM 不交换/半字交换

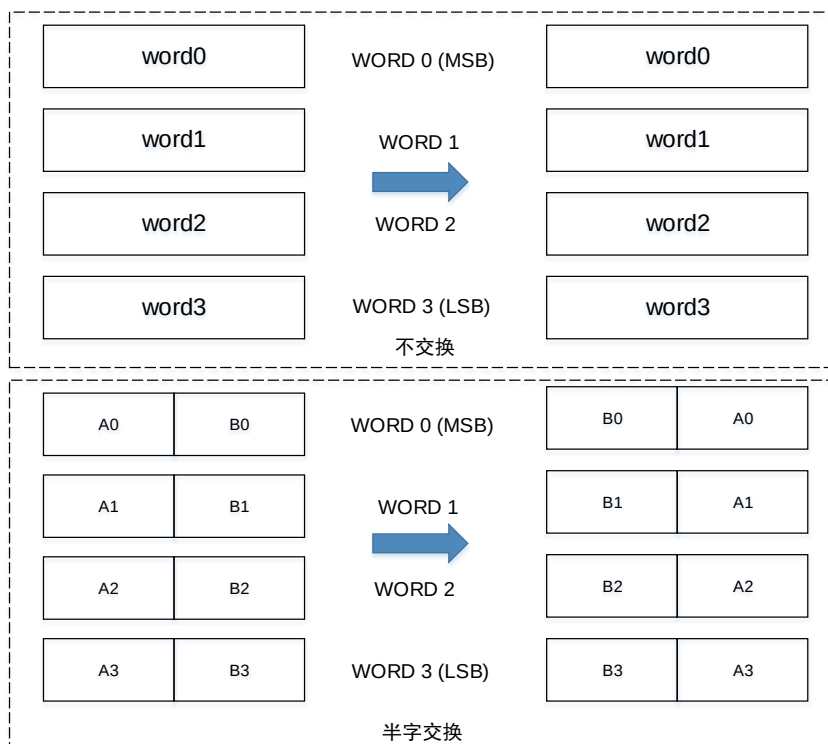
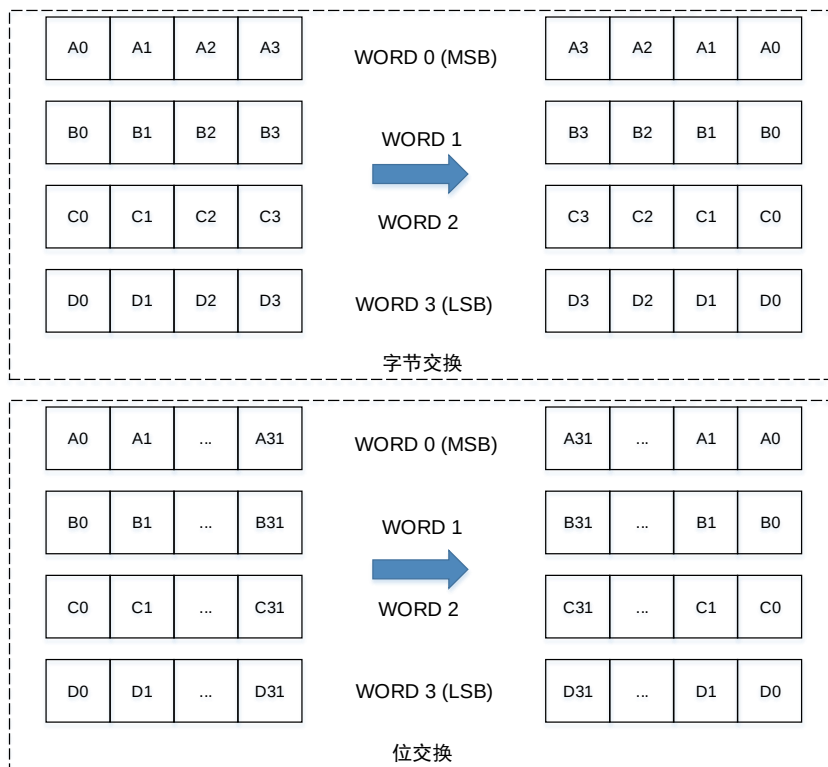


图 18-2. DATATM 字节交换/位交换

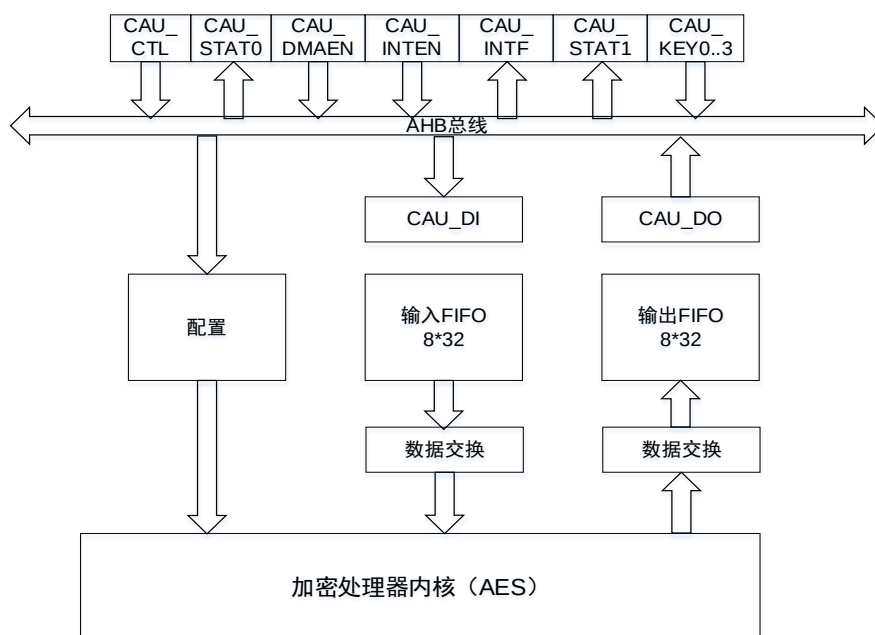


18.4. 加密处理器流程

加密处理器关于 DES 和 AES 加密处理的实现具体请参考章节 [AES 加密处理流程](#)。

[图 18-3. CAU 框图](#)为加密处理器的模块框图。

图 18-3. CAU 框图



18.4.1. AES 加密处理流程

AES 加密处理器由 AES 算法（AEA），多个密钥，以及初始化向量或随机数三部分组成。

AES 支持三种长度的密钥：128、192 和 256 位密钥，根据操作模式的不同使用不同数目的初始化向量或随机数。

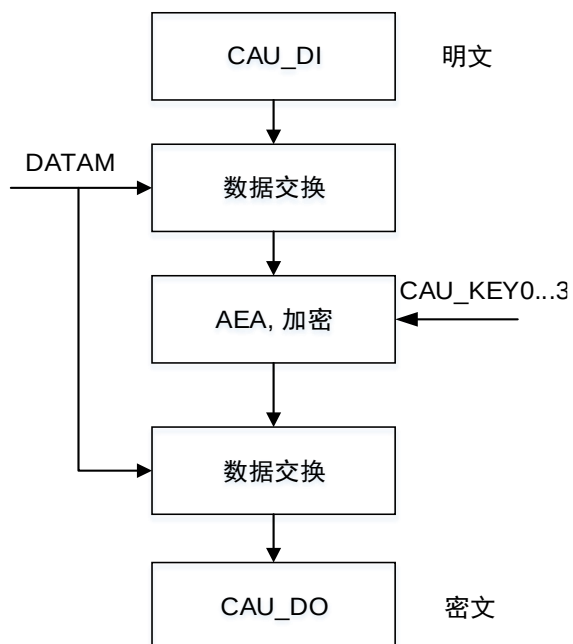
使用 128 位密钥时 AES 密钥为[KEY3 KEY2]，使用 192 位密钥时 AES 密钥为[KEY3 KEY2 KEY1]，使用 256 位密钥时 AES 密钥为[KEY3 KEY2 KEY1 KEY0]。

FIPS PUB 197（2001 年 11 月 26 日）中对 AES 中使用的密钥进行了详细的解释，本手册不再赘述。

AES 电子密码本（ECB）加密

根据数据类型进行数据交换后，首先得到 128 位输入明文数据块。输入数据块通过 AEA 使用 128 位，或 192 位，或 256 位密钥进行加密处理。处理结果再根据数据类型进行数据交换，生成一个 128 位密文输出数据块，并存储在输出 FIFO 中。AES 电子密码本加密流程图见 [图 18-4. AES ECB 加密](#)。

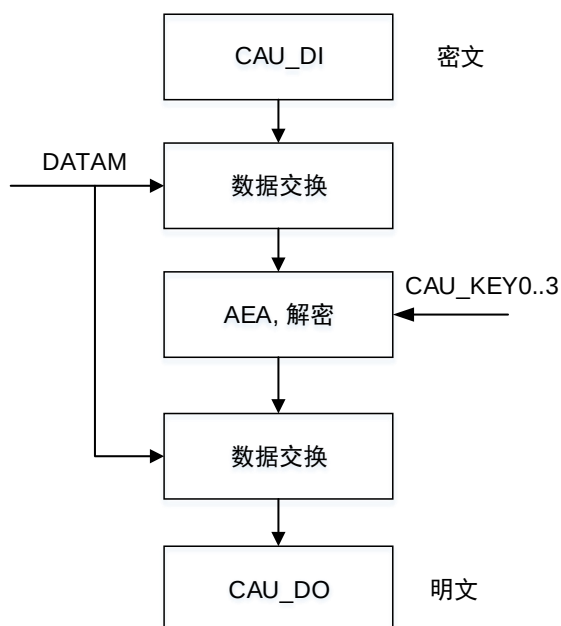
图 18-4. AES ECB 加密



AES 电子密码本（ECB）解密

首先需要准备密钥，以用于解密，密钥准备过程的输入密钥与加密处理中的密钥相同。从上述操作中获得的最后一个密钥将作为解密处理用的第一个密钥。密钥准备完成后，首先根据数据类型进行数据交换得到 128 位输入密文数据块。输入数据块在 AEA 中读取并使用上面准备的密钥进行解密处理。处理结果输出再根据数据类型值进行数据交换，生成一个 128 位明文输出数据块。AES 电子密码本解密流程图见[图 18-5. AES ECB 解密](#)。

图 18-5. AES ECB 解密



18.5. 操作模式

加密

1. 将CAU_CTL寄存器的CAUEN位清零，以禁用CAU；
2. 若选择了AES算法，则对CAU_CTL寄存器的KEYM位进行选择设置，配置密钥的长度；
3. 将CAU_CTL寄存器中的KEY_SEL位清零，然后根据算法配置CAU_KEY0..3(H/L)寄存器；
4. 设置CAU_CTL寄存器的DATAM位，配置数据交换类型；
5. 设置CAU_CTL寄存器的ALGM[2:0]位，配置算法（AES）和模式（ECB）；
6. 设置CAU_CTL寄存器的CAUDIR位为0，配置为加密操作；
7. 在CAUEN位为0时，设置CAU_CTL寄存器的FFLUSH位，配置刷新输入FIFO和输出FIFO；
8. 设置CAU_CTL寄存器的CAUEN位为1，使能CAU；
9. 当CAU_STAT0寄存器的INF位为1时，向CAU_DI寄存器写数据块。数据可以通过DMA传输或者CPU中断传输，也可不通过两者进行传输；
10. 等待CAU_STAT0寄存器的ONE位为1时，读CAU_DO寄存器。输出数据可以通过DMA传输或者CPU中断传输，也可不通过两者进行传输；
11. 重复步骤9和步骤10，直到所有的数据块都完成加密。

解密

1. 将CAU_CTL寄存器的CAUEN位清零，以禁用CAU；
2. 若选择了AES算法，则对CAU_CTL寄存器的KEYM位进行选择设置，配置密钥的长度；
3. 将CAU_CTL寄存器中的KEY_SEL位清零，然后根据算法配置CAU_KEY0..3(H / L)寄存器；

4. 设置CAU_CTL寄存器的DATAM位，配置数据交换类型；
5. 设置CAU_CTL寄存器的ALGM[2:0]位为“111”，配置准备密钥用于解密；
6. 设置CAU_CTL寄存器的CAUEN位为1，使能CAU；
7. 等待BUSY位和CAUEN位为0，确保解密用的密钥已准备好；
8. 设置CAU_CTL寄存器的ALGM[3:0]位，配置算法（AES）和模式（ECB）；
9. 设置CAU_CTL寄存器的CAUDIR位为1，配置为解密操作；
10. 在CAUEN位为0时，设置CAU_CTL寄存器的FFLUSH位，配置刷新输入FIFO和输出FIFO；
11. 设置CAU_CTL寄存器的CAUEN位为1，使能CAU；
12. 当CAU_STAT0寄存器的INF位为1时，向CAU_DI寄存器写数据块。数据可以通过DMA传输或者CPU中断传输，也可不通过两者进行传输；
13. 等待CAU_STAT0寄存器的ONE位为1时，读CAU_DO寄存器。输出数据可以通过DMA传输或者CPU中断传输，也可不通过两者进行传输；
14. 重复步骤12和步骤13，直到所有的数据块都完成解密。

18.6. CAU DMA 接口

DMA 可用于 CAU 模块的数据块传输。DMA 的传输操作由 CAU_DMAEN 寄存器来控制。DMAIEN 位用于输入数据的 DMA 请求传输使能，由 DMA 将一个字节数据写入 CAU_DI 寄存器。DMAOEN 位用于输出数据的 DMA 请求传输使能，获得 CAU 输出的一个字。

CAU DMA 接口支持单数据传输和突发传输以确保即使被传输数据的长度不是突发传输宽度的整数倍也能被正确传输。需要注意的是配置 DMA 控制器突发传输的长度应该不超过四个字以确保数据不会丢失。DMA 输出数据的传输请求优先级高于输入数据的传输请求，因此输出 FIFO 空的事件可能会早于输入 FIFO 满的事件。

18.7. CAU 中断

CAU 有两个中断状态寄存器，CAU_STAT1 和 CAU_INTF 寄存器。CAU 中的中断用于指示输入和输出 FIFO 的状态。

可以通过配置 CAU_INTEN 寄存器来使能或禁用输入或输出 FIFO 中断。将寄存器中相应位置 1 可以使能相应中断。

输入 FIFO 中断

当输入 FIFO 中的数据少于 4 个字时产生输入 FIFO 中断，ISTA 位置位。此时如果 IINTEN 位为 1，使能了输入 FIFO 中断，则 IINTF 位将置位。注意当 CAUEN 位为 0 时，ISTA 位和 IINTF 位将保持为 0。

输出 FIFO 中断

当输出 FIFO 中存在一个或多个字节数据时产生输出 FIFO 中断，OSTA 位置位。此时如果 OINTEN

位为 1 从而使能了输出 FIFO 中断，则 OINTF 位将置位。注意与输入 FIFO 中断不同的是，当 CAUEN 位为 0 时，不会影响到 OSTA 位与 OINTF 位的状态。

18.8. CAU 寄存器

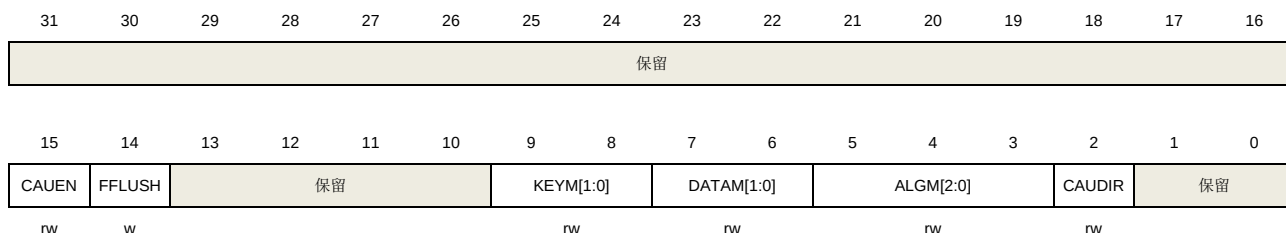
CAU 基地址：0x4002 3400

18.8.1. 控制寄存器（CAU_CTL）

偏移地址：0x00

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	CAUEN	加密处理器使能 0: 加密处理器禁用 1: 加密处理器使能 注意： 当准备密钥（ALGM=111b）完成后，CAUEN 位将硬件自动清零。
14	FFLUSH	FIFO 刷新 0: 不产生影响 1: 当 CAUEN=1 时，刷新输入和输出 FIFO 读取该位时，始终返回 0
13:10	保留	必须保持复位值。
9:8	KEYM[1:0]	AES 密钥长度配置，必须在 BUSY=0 时才可配置 00: 128 位密钥长度 01: 192 位密钥长度 10: 256 位密钥长度 11: 保留

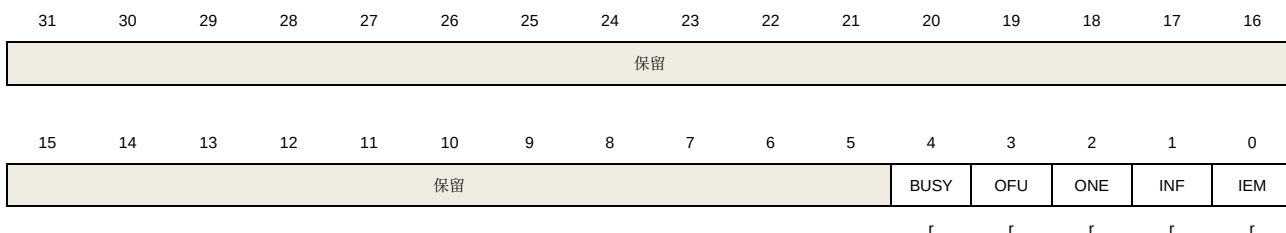
7:6	DATAM[1:0]	数据交换模式配置，必须在 BUSY=0 时才配置 00: 不交换 01: 半字交换 10: 字节交换 11: 位交换
5:3	ALGM[2:0]	加密/解密算法模式位 0 到 2 100: AES-ECB（AES 电子密码本），使用 CAU_KEY0, 1, 2, 3. 111: AES 解密密钥准备模式。输入密钥必须与加密处理中用的密钥相同。BUSY 位将保持置位直到完成密钥的准备，随后 CAUEN 位会清零。
2	CAUDIR	CAU 算法方向，必须在 BUSY=0 时才配置 0: 加密 1: 解密
1:0	保留	必须保持复位值。

18.8.2. 状态寄存器 0 (CAU_STAT0)

偏移地址：0x04

复位值：0x0000 0003

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:5	保留	必须保持复位值。
4	BUSY	忙碌标志位 0: CAU 内核空闲，这是由于 - CAUEN=0 从而 CAU 内核被禁用，或这处理已完成 - 正在等待输入数据或输出 FIFO 有足够的自由空间来处理数据块 1: CAU 内核忙碌，正在处理数据块或准备密钥
3	OFU	输出 FIFO 满 0: 输出 FIFO 未滿 1: 输出 FIFO 满

2	ONE	输出 FIFO 非空 0: 输出 FIFO 为空 1: 输出 FIFO 非空
1	INF	输入 FIFO 未滿 0: 输入 FIFO 滿 1: 输入 FIFO 未滿
0	IEM	输入 FIFO 空 0: 输入 FIFO 非空 1: 输入 FIFO 空

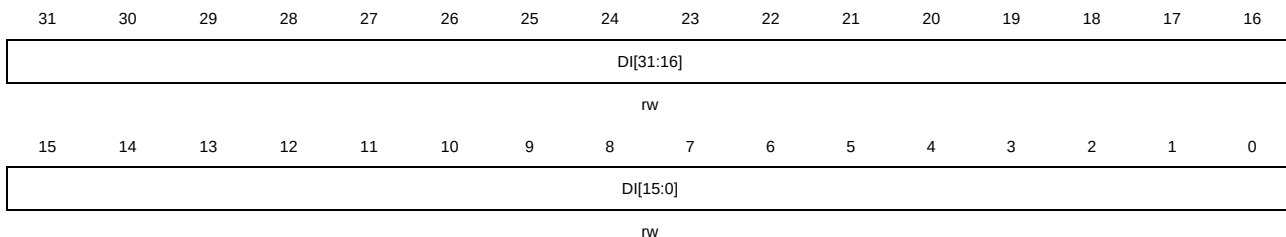
18.8.3. 数据输入寄存器 (CAU_DI)

偏移地址: 0x08

复位值: 0x0000 0000

数据输入寄存器用于传输明文或密文数据块到输入 FIFO 中进行处理。首先写入 FIFO 的是数据块的 MSB, 最后才是 LSB。当 CAUEN 位为 0, 并且输入 FIFO 非空时, 读取该寄存器时返回 FIFO 中的首个字。当 CAUEN 位为 1 时, 读取该寄存器返回一个不确定的值。一旦执行了读操作, 则必须要刷新 FIFO 以处理新数据块。

该寄存器只能按字 (32 位) 访问。



位/位域	名称	描述
31:0	DI[31:0]	数据输入 写这些位, 数据会写入输入 FIFO。当 CAUEN 位为 0 时, 读这些位将返回输入 FIFO 中的值, 否则将返回不确定的值。

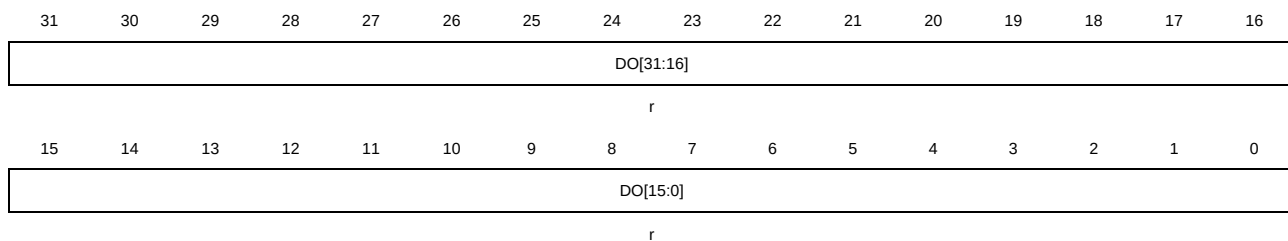
18.8.4. 数据输出寄存器 (CAU_DO)

偏移地址: 0x0C

复位值: 0x0000 0000

数据输出寄存器是只读寄存器, 用于接收来自输出 FIFO 的明文或密文处理结果。与 CAU_DI 类似, 读取时首先读取的是数据块的 MSB, 最后才是 LSB。

该寄存器只能按字（32 位）访问。



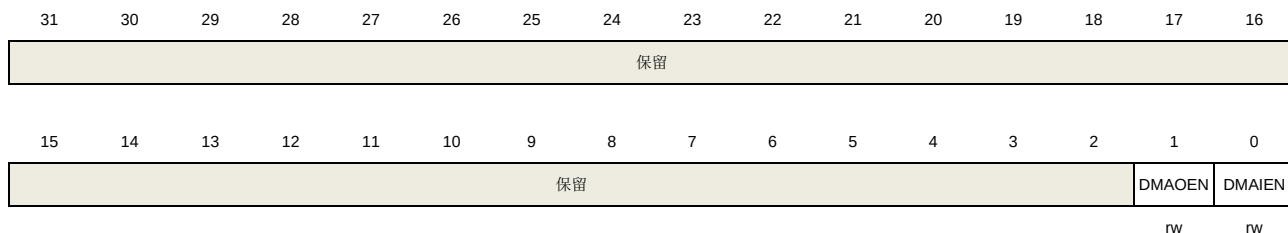
位/位域	名称	描述
31:0	DO[31:0]	数据输出 这些位为只读，读这些位将返回输出 FIFO 中的值

18.8.5. DMA 使能寄存器 (CAU_DMAEN)

偏移地址：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
31:2	保留	必须保持复位值。
1	DMAOEN	DMA 输出使能 0：禁用用于输出 FIFO 数据传输的 DMA 1：使能用于输出 FIFO 数据传输的 DMA
0	DMAIEN	DMA 输入使能 0：禁用用于输入 FIFO 数据传输的 DMA 1：使能用于输入 FIFO 数据传输的 DMA

18.8.6. 中断使能寄存器 (CAU_INTEN)

偏移地址：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														OINTEN	IINTEN
														rw	rw

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	OINTEN	输出 FIFO 中断使能 0: 禁用输出 FIFO 中断 1: 使能输出 FIFO 中断
0	IINTEN	输入 FIFO 中断使能 0: 禁用输入 FIFO 中断 1: 使能输入 FIFO 中断

18.8.7. 状态寄存器 1 (CAU_STAT1)

偏移地址: 0x18

复位值: 0x0000 0001

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														OSTA	ISTA
														r	r

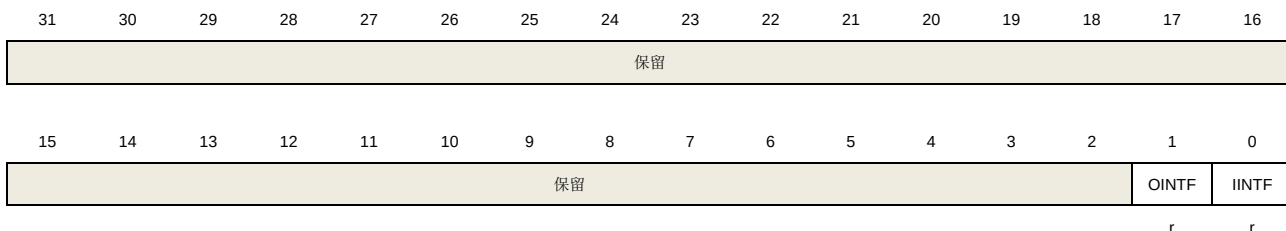
位/位域	名称	描述
31:2	保留	必须保持复位值。
1	OSTA	输出 FIFO 状态 0: 输出 FIFO 状态未挂起 1: 输出 FIFO 状态挂起
0	ISTA	输入 FIFO 状态 0: 输入 FIFO 状态未挂起 1: 输入 FIFO 状态挂起

18.8.8. 中断标志寄存器 (CAU_INTF)

偏移地址：0x1C

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:2	保留	必须保持复位值。
1	OINTF	输出 FIFO 中断标志 0: 输出 FIFO 中断状态未挂起 1: 输出 FIFO 中断状态挂起
0	IINTF	输入 FIFO 中断标志 0: 输入 FIFO 中断状态未挂起 1: 当 CAUEN 位为 1 时输入 FIFO 中断状态挂起

18.8.9. 密钥寄存器 (CAU_KEY0..3(H/L))

偏移地址：0x20~0x3C

复位值：0x0000 0000

该寄存器只能按字（32 位）访问，必须在 BUSY 位为 0 时写这些寄存器。

在 AES-128 模式下，KEY2H[31:0]和 KEY2L[31:0]分别对应于 AES_KEY[0:63]的高 32 位与低 32 位，而 KEY3H[31:0]和 KEY3L[31:0]分别对应于 AES_KEY[64:127]的高 32 位与低 32 位。

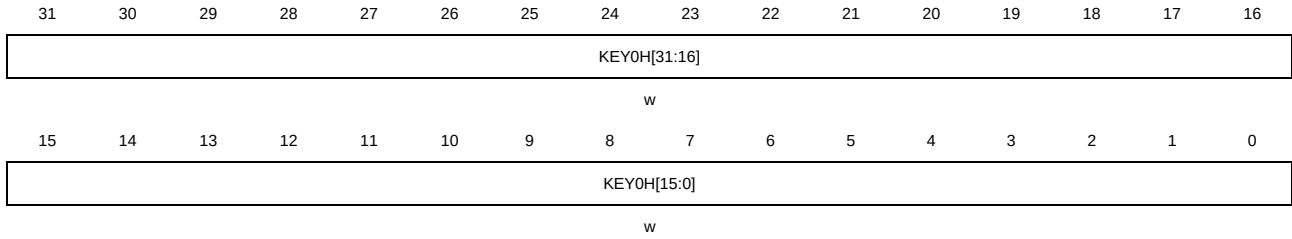
在 AES-192 模式下，KEY1H[31:0]和 KEY1L[31:0]分别对应于 AES_KEY[0:63]的高 32 位与低 32 位，KEY2H[31:0]和 KEY2L[31:0]分别对应于 AES_KEY[64:127]的高 32 位与低 32 位，KEY3H[31:0]和 KEY3L[31:0]分别对应于 AES_KEY[128:191]的高 32 位与低 32 位。

在 AES-256 模式下，KEY0H[31:0]和 KEY0L[31:0]分别对应于 AES_KEY[0:63]的高 32 位与低 32 位，KEY1H[31:0]和 KEY1L[31:0]分别对应于 AES_KEY[64:127]的高 32 位与低 32 位，KEY2H[31:0]和 KEY2L[31:0]分别对应于 AES_KEY[128:191]的高 32 位与低 32 位，KEY3H[31:0]和 KEY3L[31:0]分别对应于 AES_KEY[192:255]的高 32 位与低 32 位。

CAU_KEY0H

偏移地址: 0x20

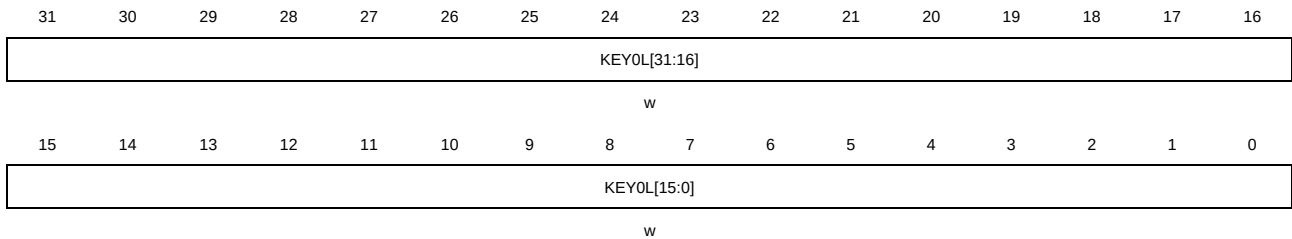
复位值: 0x0000 0000



CAU_KEY0L

偏移地址: 0x24

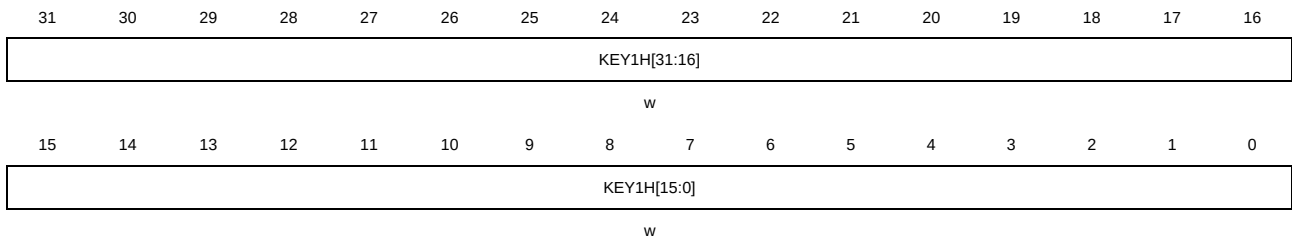
复位值: 0x0000 0000



CAU_KEY1H

偏移地址: 0x28

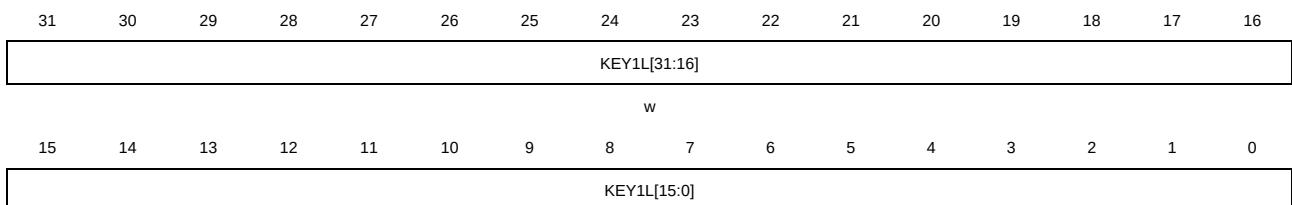
复位值: 0x0000 0000



CAU_KEY1L

偏移地址: 0x2C

复位值: 0x0000 0000

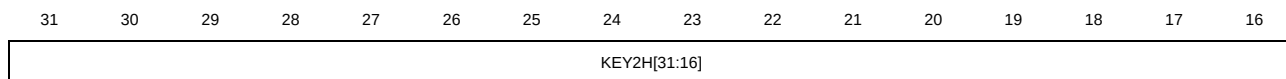


w

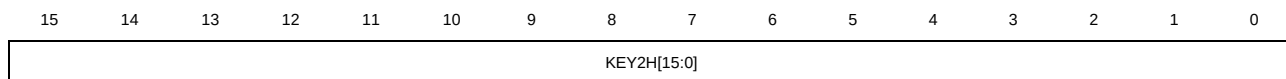
CAU_KEY2H

偏移地址: 0x30

复位值: 0x0000 0000



w

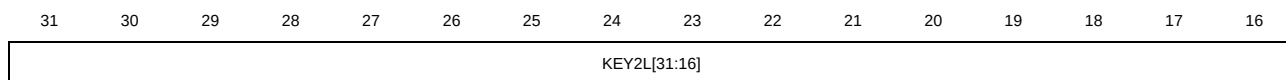


w

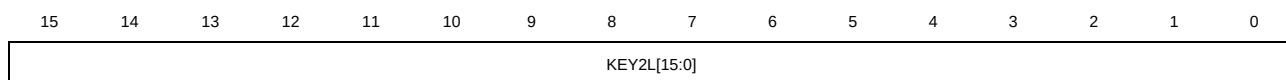
CAU_KEY2L

偏移地址: 0x34

复位值: 0x0000 0000



w

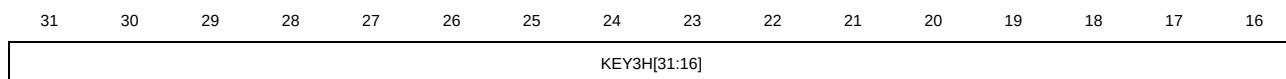


w

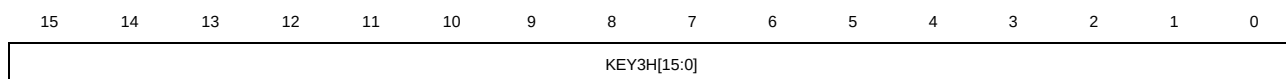
CAU_KEY3H

偏移地址: 0x38

复位值: 0x0000 0000



w

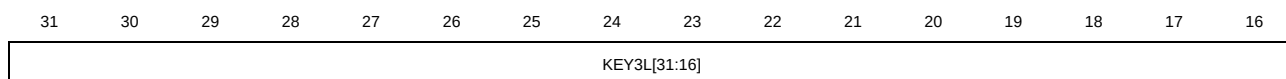


w

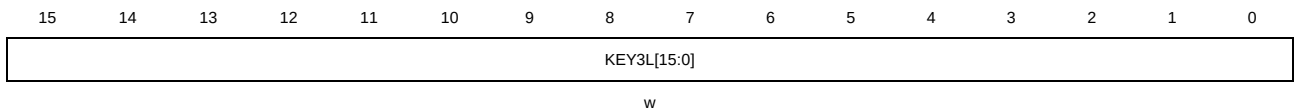
CAU_KEY3L

偏移地址: 0x3C

复位值: 0x0000 0000



w



位/位域	名称	描述
31:0	KEY0...3(H/L)	用于 DES 或 TDES 或 AES 的密钥

19. 哈希处理器（HAU）

19.1. 简介

哈希处理器应用于信息安全。支持应用于多种场合的安全哈希算法（SHA-256）。对长达（ $2^{64}-1$ ）位的消息，哈希处理器计算消息摘要长度对应于 SHA-256 为 256 位。

哈希处理器完全兼容下列标准：

- 联邦信息处理标准出版物180-4（FIPS PUB 180-4）；
- 安全散列标准规范（SHA256）；

19.2. 主要特征

- 32位AHB从外设；
- 高性能的哈希算法运算；
- 小端数据表示；
- 支持多种数据交换类型，包括32位字不交换，半字交换，字节交换和位交换；
- 可自动填充来适应模数为512位（ 16×32 位）消息摘要的计算；
- 支持DMA模式的数据流传输；

19.3. HAU 数据类型

哈希处理器每次接收 32 位字，但每次计算处理一个 512 位块。对每个输入字，在送入哈希内核之前都会根据数据类型进行位/字节/半字/不交换。同样在数据输出之前也要进行相同的数据交换。注意由于系统存储器结构采用小端模式，无论使用何种数据类型，最低有效数据均占用最低地址位置。SHA-256 的计算均为大端模式。

[图 19-1. DATAM 不交换/半字交换](#)和[图 19-2. DATAM 字节交换/位交换](#)介绍了在不同数据类型下的数据交换。

图 19-1. DATAM 不交换/半字交换

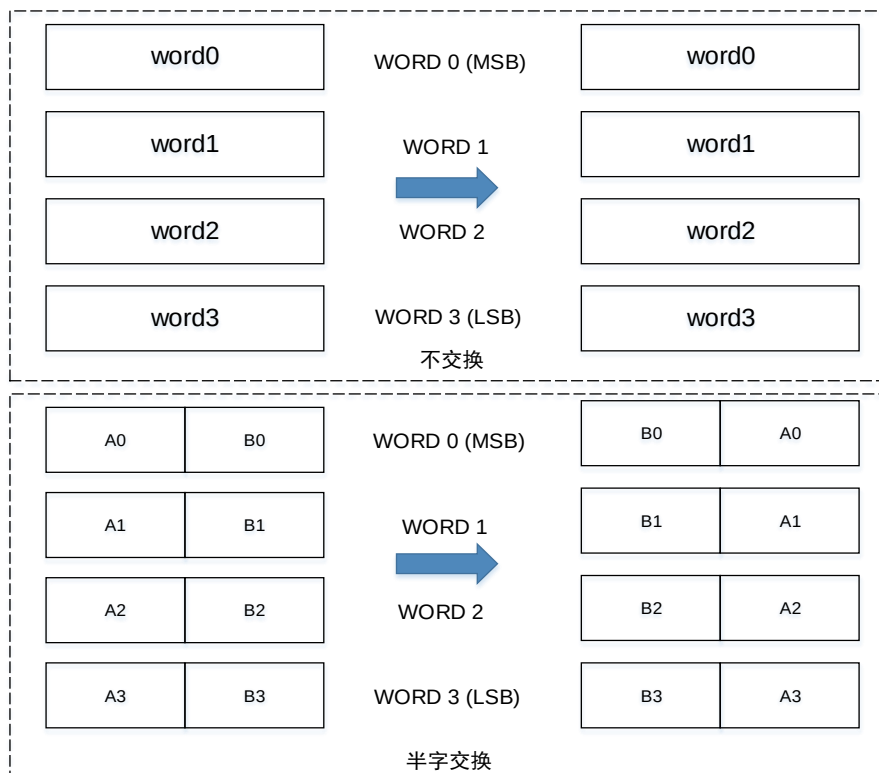
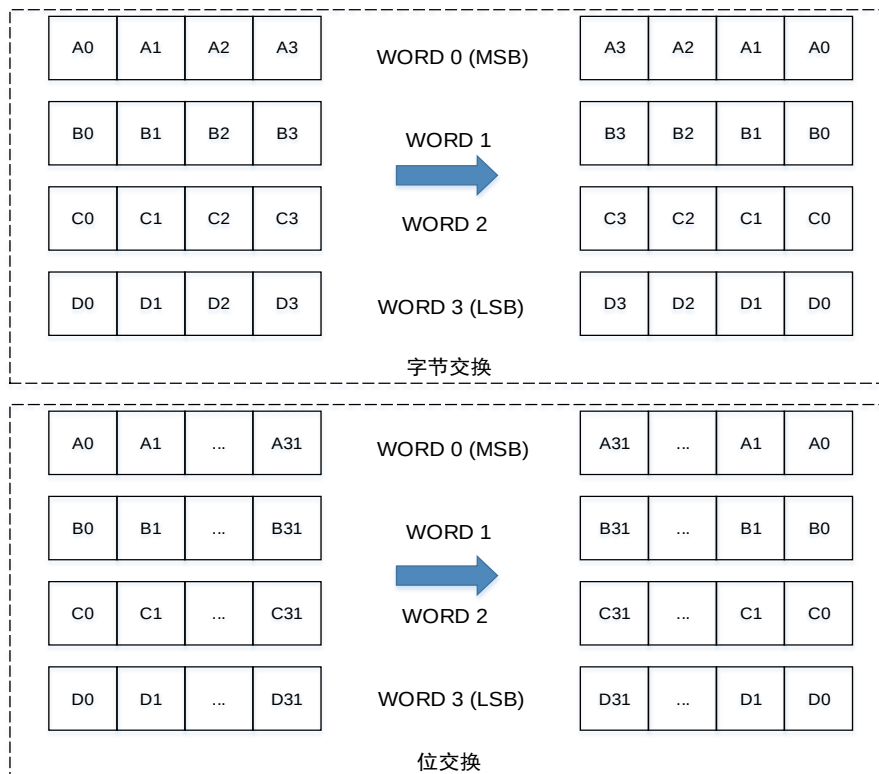


图 19-2. DATAM 字节交换/位交换

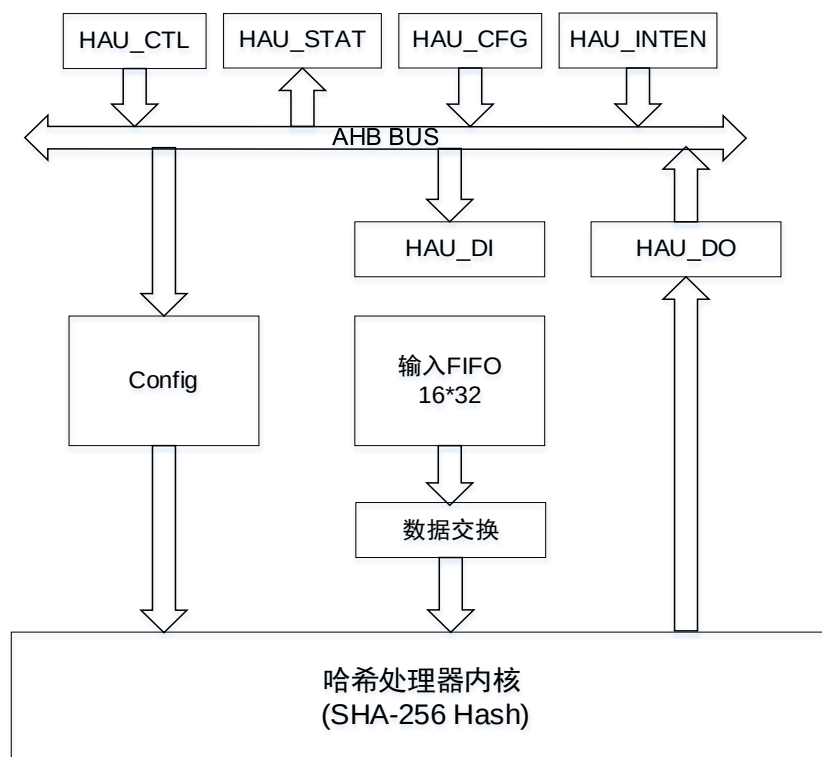


19.4. HAU 内核

哈希处理器使用安全哈希算法对输入消息进行信息压缩计算。对长度为 $(2^{64}-1)$ 位的消息摘要计算结果的长度对应于 SHA-256 为 256 位。哈希处理器可用于生成和验证消息签名，并由于摘要远远小于消息的大小而具有更高的效率。

要由哈希处理器处理的消息应视为位串。消息长度为消息的位数。哈希处理器可以确保信息的安全，因为根据某个给定消息摘要来寻找原对应的消息在计算层面是无法实现的，而在原输入消息上任何的改动都将导致生成完全不同的消息摘要。

图 19-3. HAU 结构框图



19.4.1. 自动数据填充

为确保输入 HAU 内核的数据为 512 位的整数倍，需要对输入消息进行填充。消息填充操作由在原始消息的结尾添加一个 1，后跟几个 0 和一个 64 位整数，填充物（0）将消息填充到整个 512 位的前 448 位，实现生成一个长度为 512 的填充消息块。

消息填充完成后，通过配置 HAU_CFG 寄存器的 VBL 位域来设置上面所述的 64 位整数。设置 HAU_CFG 寄存器的 CALEN 位为 1，开始计算上个数据块的摘要。

数据填充示例：输入消息为“HAU”，对应的 ASCII 码 16 进制表示为

484155

接着根据消息的有效位长度，设置 HAU_CFG 寄存器的 VBL 位域为 24。接着在位串的第 24 位处添加一个“1”，随后填充数个“0”使位串模数为 448，十六进制结果如下所示：

```
48415580 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000
```

之后，添加 64 位整数到已填充的输入消息后，该 64 位整数十六进制值为 18，则最后的结果应该是：

```
48415580 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000018
```

19.4.2. 摘要计算

数据填充完成之后，通过 DMA 或 CPU 每次将 512 位的数据块送入 HAU 内核，HAU 对每个数据块进行计算。在 HAU 内核开始计算之前，外设需要知道 HAU_DI 寄存器是否包含消息的最后一位。这可以从输入 FIFO 的状态和 HAU_DI 寄存器来确认。

通过 DMA 传输数据：

数据块传输的状态将自动通过 DMA 控制器发送的信息来解释。当 HAU_CFG 寄存器的 CALEN 位置 1 时，将自动开始进行数据填充和摘要计算。

注意：如果消息是个大文件并需要多个 DMA 传输，则应将 MDS 位置 1。另外在传输之前需要设置 VBL 位域。在 DMA 的传输完成之后硬件不会自动将 CALEN 位置 1，以便可以接收新的 DMA 传输。在最后的 DMA 传输期间，需要将 MDS 位清零，从而在最后一个块传输结束时硬件自动将 CALEN 位置 1。

若消息不需要多个 DMA 传输，则将 MDS 置 0 即可，这样在一个 DMA 传输完成之后就会硬件自动置位 CALEN 位。同样的，在 DMA 传输之前也需要先设置 VBL 位域。

通过 CPU 传输数据：

- 当 HAU_DI 寄存器中写入下一个数据块的第一个字时，将开始计算当前数据块的摘要；
- 将 HAU_CFG 寄存器中 CALEN 位置 1，将开始最后一个数据块的摘要计算。

19.4.3. 哈希模式

当从 HAU_DI 寄存器和接收 FIFO 中接收到 512 位的消息块时，将根据 DMA 和 CALEN 位状态开

始摘要的计算。

最终的计算结果可以从 HAU_DO0...7 寄存器中读取。

19.5. HAU 中断

HAU 具有两个独立的中断源,并在 HAU_STAT 有相应的状态位。这两个状态位用于指示输入 FIFO 的状态,以及摘要的计算是否完成。

HAU_INTEN 寄存器为中断使能寄存器。将相应位置 1 可以使能中断。

19.5.1. 输入 FIFO 中断

当输入 FIFO 中的数据处理已完成时,输入 FIFO 标志位 DIF 置位。如果置位 DIIE 位使能了输入 FIFO 中断,则当输入 FIFO 标志位 DIF 置位时会发生输入 FIFO 中断。

19.5.2. 计算完成中断

当摘要计算完成时,计算完成标志位 CCF 将置位。如果置位 CCIE 位使能了计算完成中断,则当计算完成标志位 CCF 置位时会发生计算完成中断。

19.6. HAU 寄存器

HAU 基地址：0x4002 3800

19.6.1. 控制寄存器（HAU_CTL）

地址偏移：0x00

复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

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

位/位域	名称	描述
31:14	保留	必须保持复位值。
13	MDS	多DMA选择 如果哈希消息为大型文件需要多个DMA传输时，将此位置1 0：仅需要单次DMA传输，在DMA传输结束时硬件自动将CALEN位置1 1：需要多次DMA传输，在DMA传输结束时硬件不自动将CALEN位置1
12	DINE	DI寄存器非空 0：DI寄存器空 1：DI寄存器非空 注意： 当START位或CALEN位为1时此位会清零
11:8	NWIF[3:0]	输入FIFO中的字数 注意： 当START位置位时，或开始进行摘要计算时（CALEN位置位，或者DMA传输结束），该位域清零
7:6	保留	必须保持复位值。
5:4	DATAM[1:0]	数据交换类型 定义输入到HAU_DI寄存器中的数据格式 00：不交换，写入到HAU_DI寄存器的数据将直接送入FIFO，不进行交换 01：半字交换。写入到HAU_DI寄存器的数据在送入FIFO前，需要进行半字交换 10：字节交换。写入到HAU_DI寄存器的数据在送入FIFO前，需要进行字节交换 11：位交换。写入到HAU_DI寄存器的数据在送入FIFO前，需要进行位交换
3	DMAE	DMA使能

0: 禁止DMA传输

1: 使能DMA传输

注意: 1.当DMA传输消息的最后一个数据时,将由硬件清零该位。当START置位时,不会清零该位。

2.如果DMA正在传输数据,将该位写入0不会中止当前的传输,而直到当前传输结束或START位置为1之后,才会禁止传输。

2	START	开始摘要计算 0: 没有影响 1: 开始新消息的摘要计算 注意: 读取该位将始终返回0
1:0	保留	必须保持复位值。

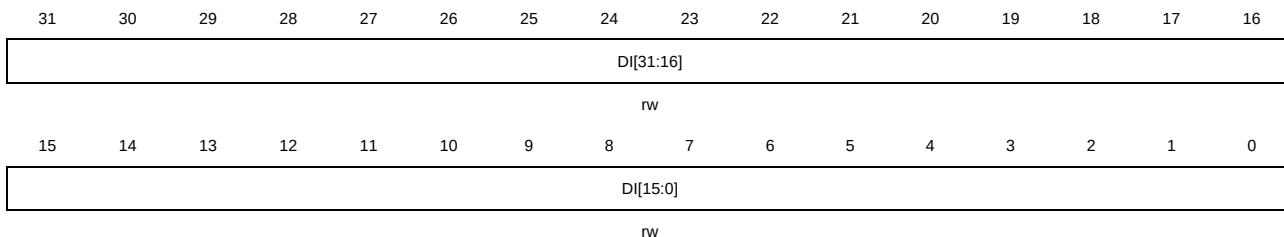
19.6.2. 数据输入寄存器 (HAU_DI)

地址偏移: 0x04

复位值: 0x0000 0000

该数据输入寄存器用于将 512 位的数据块送入输入 FIFO 进行处理。当正在进行摘要计算时,所有对该寄存器的新的写访问将被延迟,直到计算完成。

该寄存器只能按字(32 位)访问。



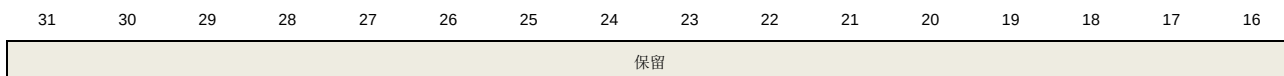
位/位域	名称	描述
31:0	DI[31:0]	消息数据输入 当数据写入这些寄存器时,寄存器中当前的内容被推入输入FIFO中同时更新为新的值。当读寄存器时,返回寄存器的当前内容。

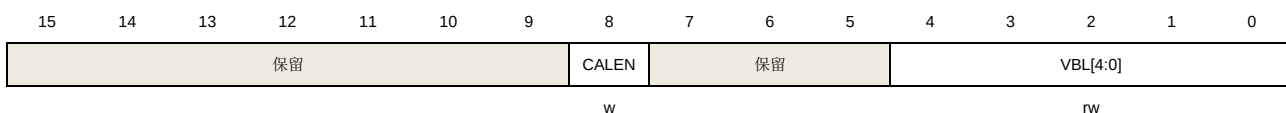
19.6.3. 配置寄存器 (HAU_CFG)

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
31:9	保留	必须保持复位值。
8	CALEN	使能摘要计算 0: 不计算 1: 先使用VBL位域对数据进行数据填充，然后开始计算最终消息摘要 注意： 读该位将返回0
7:5	保留	必须保持复位值。
4:0	VBL[4:0]	消息的最后一个字中的有效位数 0x00: 对于写入HAU_DI寄存器的最后一个数据，所有32位（在数据交换后）均有效。 0x01: 对于写入HAU_DI寄存器的最后一个数据，仅位[31]（在数据交换后）有效。 0x02: 对于写入HAU_DI寄存器的最后一个数据，仅位[31:30]（在数据交换后）有效。 0x03: 对于写入HAU_DI寄存器的最后一个数据，仅位[31:29]（在数据交换后）有效。 ... 0x1F: 对于写入HAU_DI寄存器的最后一个数据，仅位[31:1]（在数据交换后）有效。 注意： 必须在置位CALEN位之前配置该位。

19.6.4. 数据输出寄存器 (HAU_DO0..7)

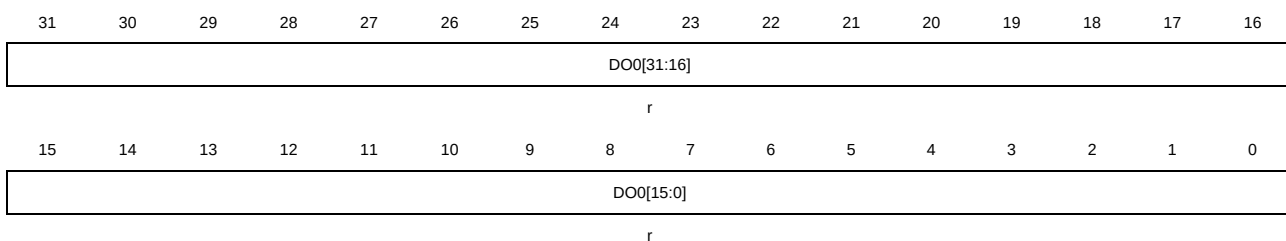
数据输出寄存器为只读寄存器，用于从输出 FIFO 中接收计算结果。置位 **START** 位将复位该寄存器。当正在进行摘要计算时，所有对该寄存器的新的读访问将被延迟，直到计算完成。

HAU_DO0

地址偏移：0x0C 和 0x310

复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

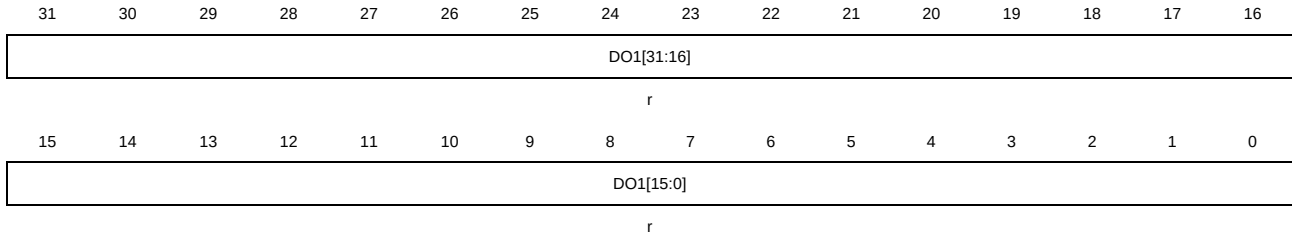


HAU_DO1

地址偏移：0x10 和 0x314

复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

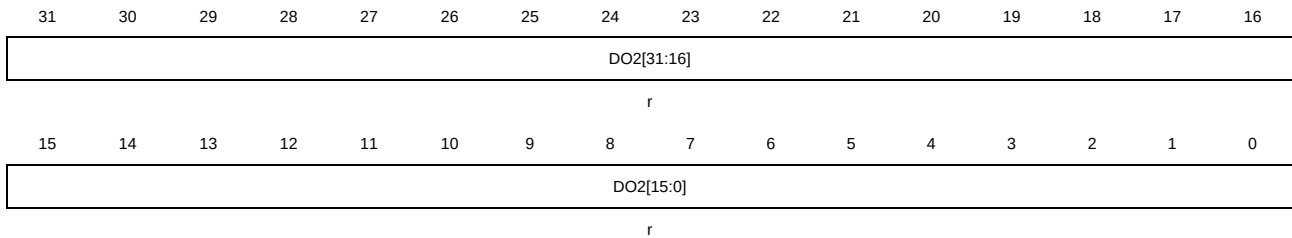


HAU_DO2

地址偏移：0x14 和 0x318

复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

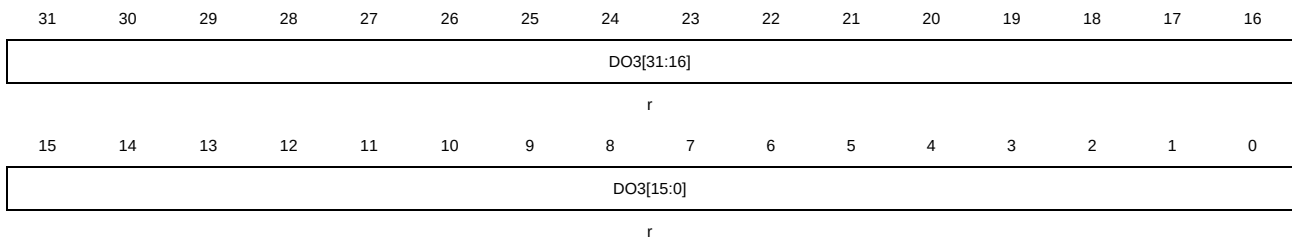


HAU_DO3

地址偏移：0x18 和 0x31C

复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

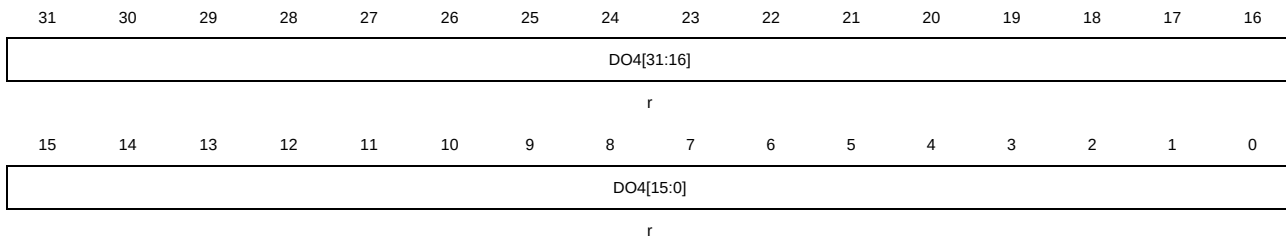


HAU_DO4

地址偏移：0x1C 和 0x320

复位值：0x0000 0000

该寄存器只能按字(32 位)访问。

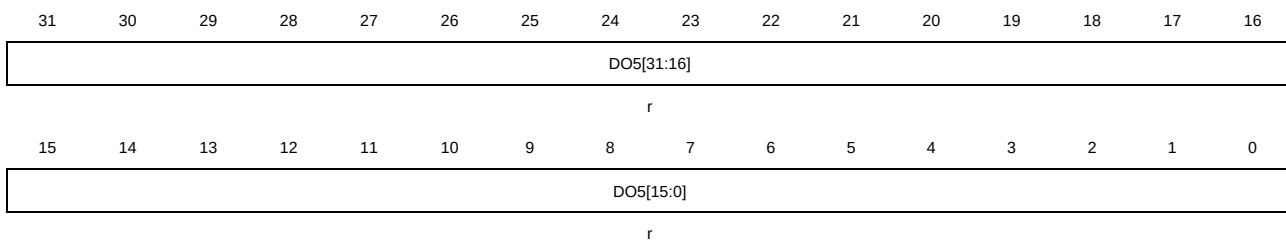


HAU_DO5

地址偏移: 0x324

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问。

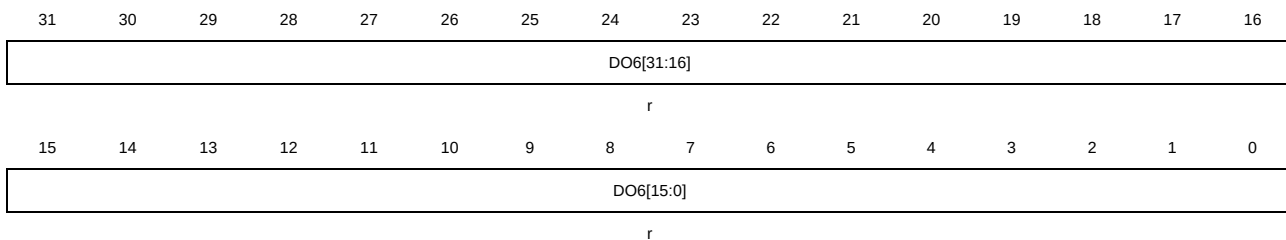


HAU_DO6

地址偏移: 0x328

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问。

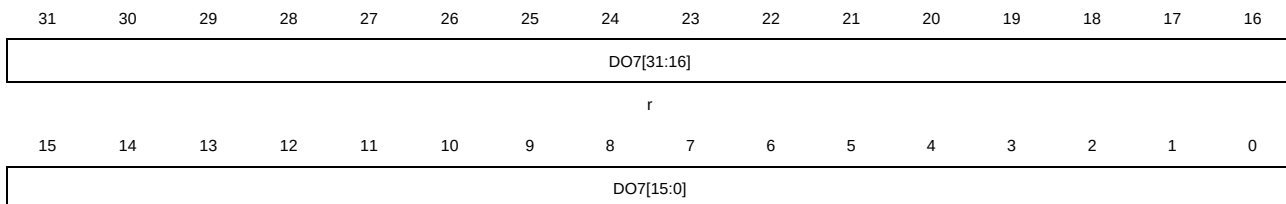


HAU_DO7

地址偏移: 0x32C

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问。



r

位/位域	名称	描述
31:0	DO0..7[31:0]	消息摘要结果

19.6.5. 中断使能寄存器 (HAU_INTEN)

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														CCIE	DIIE
														rw	rw

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	CCIE	计算完成中断使能 0: 禁止计算完成中断 1: 使能计算完成中断
0	DIIE	数据输入中断使能 0: 禁止数据输入中断 1: 使能数据输入中断

19.6.6. 状态与标志寄存器 (HAU_STAT)

地址偏移: 0x24

复位值: 0x0000 0001

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												BUSY	DMAS	CCF	DIF
												r	r	rc_w0	rc_w0



位/位域	名称	描述
31:4	保留	必须保持复位值。
3	BUSY	忙标志位 0: 未处理任何块 1: 正在处理某个数据块
2	DMAS	DMA状态标志 0: DMA接口被禁用(DMAE=0)并且未在进行任何传输 1: DMA接口被使能(DMAE=1)并且未在进行任何传输
1	CCF	计算完成状态标志 0: 计算未完成 1: 所有消息摘要计算完成
0	DIF	数据输入状态标志 0: 有一个字数据写入数据输入寄存器 1: 完成一个字数据的初步处理（只有在输入FIFO中的数据才会被处理）

20. 比较器（CMP）

20.1. 简介

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

比较器可通过模拟信号将 MCU 从低功耗模式中唤醒，在一定的条件下，可将模拟信号作为触发源，结合 DAC 和定时器的 PWM 输出，可以实现电流控制。比较器的输出消隐功能可以用于防止电机控制中的伪电流检测。

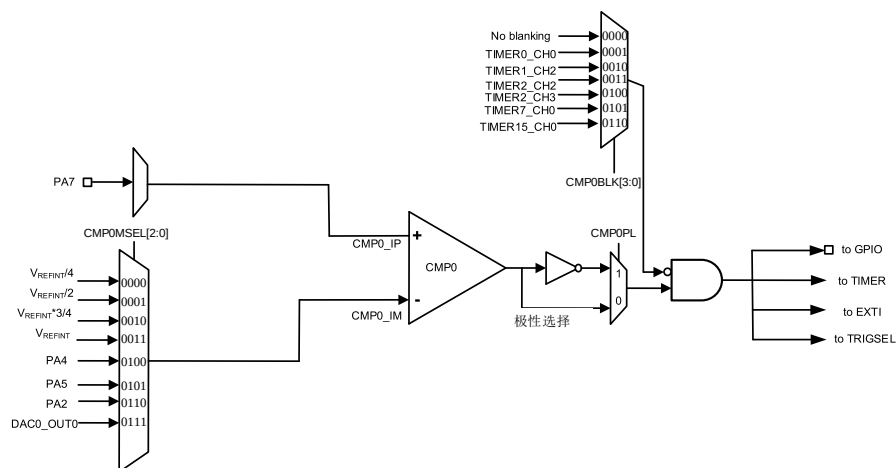
20.2. 主要特征

- 轨对轨比较器；
- 迟滞可配置；
- 每个比较器可配置以下模拟信号作为输入源：
 - DAC输出；
 - 多路复用I/O引脚；
 - 内部参考电压的整体或分数倍值；
- 比较器输出消隐；
- 输出到I/O口；
- 作为触发源输出到定时器；
- 输出到EXTI；
- 输出到NVIC；
- 输出到TRIGSEL。

20.3. 功能描述

[图 20-1. 比较器框图](#)展示如下：

图 20-1.比较器框图



注意：VREFINT 是 1.2V。

20.3.1. 比较器时钟

比较器的时钟与 PCLK 同步，它有专用的复位和时钟使能位。

20.3.2. 比较器的 I/O 配置

在被选为比较器输入端之前，相应管脚必须配置为模拟模式。

比较器的输出可同时实现内部和外部输出。

参考 Datasheet 的引脚定义，比较器输出可以通过 GPIO 的备用功能连接到对应的 I/O 口。

比较器输出内部连接到定时器，他们的连接关系如下：

- CMP 输出连接到定时器中止功能。

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

[表 20-1. 输入和输出汇总](#)详细描述了 CMP 的输入和输出。

表 20-1. 输入和输出汇总

	CMP
CMP 同相输入连接到 I/O	PA7
CMP 反相输入连接到 I/O	PA4 PA5 PA2
CMP 反相输入连接到内部信号	VREFINT/4, VREFINT/2, VREFINT*3/4,

	CMP
	V _{REFINT} , DAC0_OUT0
CMP 输出连接到 I/O	PA0 PA1 PA6 PB0 PB1
CMP 输出连接到 EXTI	•
CMP 输出连接到 TRIGSEL	•
CMP 输出连接到 NVIC	•
CMP 输出连接到 TIMER break	•

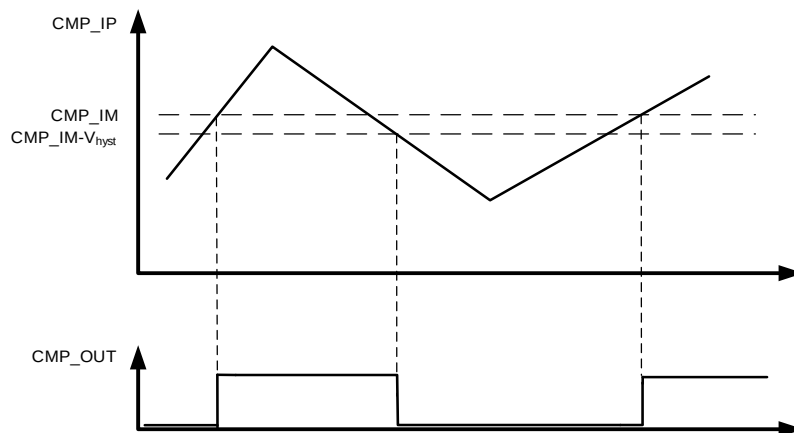
注意：

1. 有关CMP输出连接到TIMER break的详细信息，请参阅TIMER模块；

20.3.3. 比较器迟滞

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

图 20-2. 比较器迟滞



20.3.4. 比较器寄存器写保护

比较器的控制状态寄存器（CMP0_CS）可通过设置 CMP0LK 位为 1 来进行保护，CMP0_CS 寄存器，包含 CMP0LK 位，就会变为只读位，只有在 MCU 复位时才可以复位。

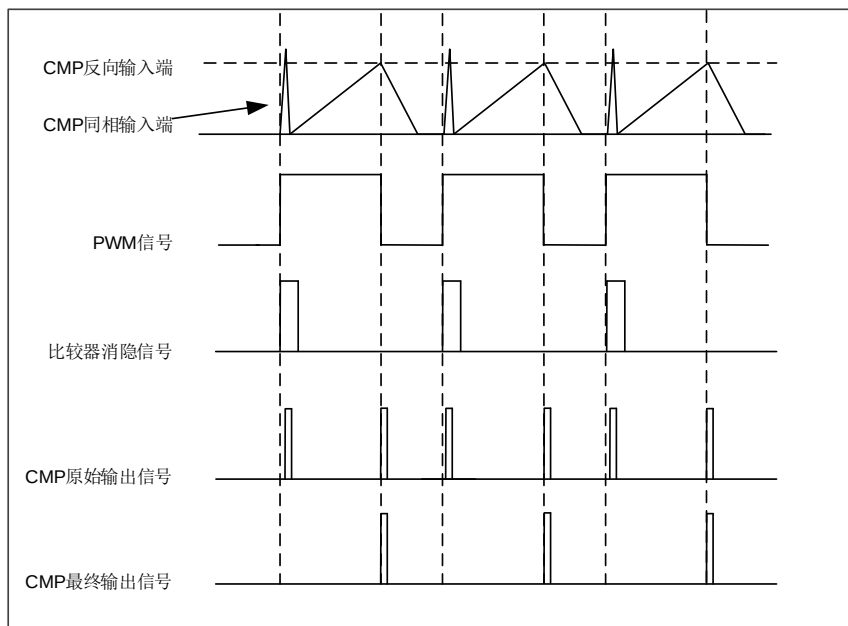
20.3.5. 比较器输出消隐

比较器输出消隐功能可以避免比较器输入信号中的短脉冲对输出信号的干扰。如果 CMP0_CS 寄

寄存器中的 CMP0BLK[2:0]位域设置为有效值，则比较器最终输出的信号由所选消隐信号的互补信号和比较器的原始输出进行“与”运算获得。

图 20-3. 比较器的输出消隐显示了比较器的输出消隐功能。

图 20-3. 比较器的输出消隐



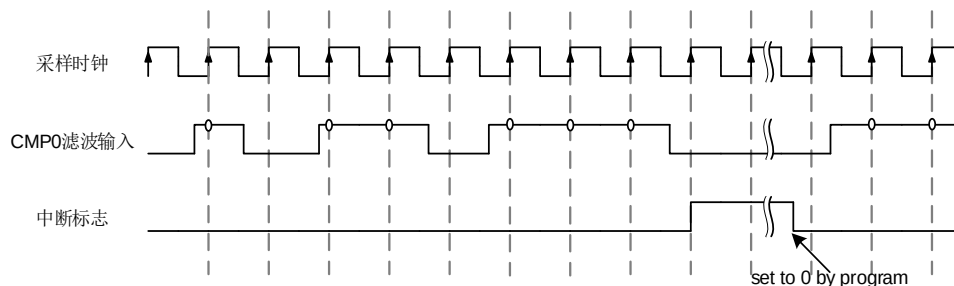
20.3.6. 比较器数字滤波

数字滤波器的采样时钟可由 CMP0DFSCDIV[1:0]位选择，匹配次数可由 CMP0_CS 寄存器的 CMP0DFSNUM 位选择。数字滤波器函数的作用如下：

- 可选择三种采样周期之一。
- 也可以关闭过滤功能。
- 经过噪声滤波的信号可用于产生中断请求输出、TIMER 内部触发源输出和比较结果输出。

每个采样时钟对比较器输出进行采样，如果对相同的值进行三次或四次采样，则将该值确定为下一个采样时钟的噪声滤波器输出。数字噪声滤波器的细节如图 20-4. 比较器噪声滤波和中断操作 所示。

图 20-4. 比较器噪声滤波和中断操作



上面的操作示例是使能数字噪声滤波，适用于 CMP0DFSCDIV[1:0]位为 2'b01, 2'b10 和 2'b11, CMP0DFSNUM 位为 0 的情况。通过设置 CMP0DFSCDIV[1:0]位的值来选择采样时间，可以改变消噪宽度。

20.3.7. 电压定标器功能

电压定标器功能可为 CMP 输入提供 1 / 4、1 / 2、3 / 4 参考电压供选择。它由 CMP0SEN 位和 CMP0BEN 位控制，分别用来启用或禁用模拟放大器和电阻桥。

20.3.8. 比较器中断

CMP 输出连接到 EXTI，EXTI 线对每个 CMP 都是独占的。通过这个功能，可以产生中断或者事件，用于退出睡眠模式或者深度睡眠模式。

CMP 还可以产生对 NVIC 的中断。它是一个序列逻辑信号，因此需要 PCLK。

20.4. 比较器寄存器

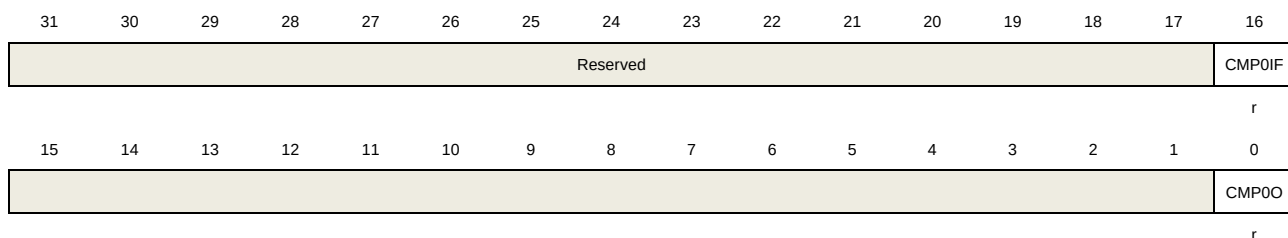
CMP 基地址：0x4000 7800

20.4.1. 状态寄存器（CMP_STAT）

地址偏移：0x00

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



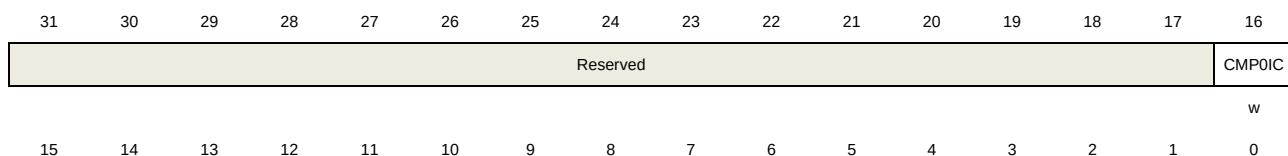
位/位域	名称	描述
31:17	保留	必须保持复位值。
16	CMP0IF	CMP0中断标志位 0：无CMP0中断发生 1：CMP0中断发生 当CMP0输出置位时，该位由硬件置1。 软件写1至CMP_IFC寄存器CMP0IC位清0。
15:1	保留	必须保持复位值。
0	CMP0O	CMP0 输出 该位反映 CMP0 输出状态，是只读位 0：正相输入端低于反相输入端，输出为低电平 1：正相输入端高于反相输入端，输出为高电平

20.4.2. 中断标志位清除寄存器（CMP_IFC）

地址偏移：0x04

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



Reserved

位/位域	名称	描述
31:17	保留	必须保持复位值。
16	CMP0IC	CMP0中断标志位清除 0: 无清除中断标志位发生 1: 清除中断标志位
15:0	保留	必须保持复位值。

20.4.3. CMP0 控制状态寄存器 (CMP0_CS)

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
CMP0LK	CMP0DFSNUM	CMP0DFSCDIV[1:0]	CMP0BLK[3:0]				保留				CMP0MSEL[3:0]				
rwo	rw	rw	rw								rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						CMP0HST[1:0]		保留	CMP0INTEN	保留	CMP0PL	CMP0SEN	CMP0BEN	CMP0EN	
						rw			rw		rw	rw	rw	rw	

位/位域	名称	描述
31	CMP0LK	CMP0 写保护 该位可将 CMP0 的各控制位设为只读。该位只可通过软件置位一次，通过系统复位清除。 0: CMP0_CS 是可读可写位 1: CMP0_CS 是只读位
30	CMP0DFSNUM	CMP0 数字滤波采样次数 0: 同一值采样 3 次，输出发生变化。 1: 同一值采样 4 次，输出发生变化
29:28	CMP0DFSCDIV[1:0]	CMP0 数字滤波采样时钟分频 该位被用于选择数字滤波的采样频率。 00: 未使用数字滤波器 01: 采样频率为 $f_{PCLK} / 8$ 。 10: 采样频率为 $f_{PCLK} / 16$ 。

		11: 采样频率为 $f_{PCLK} / 32$ 。
27:24	CMP0BLK[3:0]	CMP0 输出消隐源 该位域用于选择哪个定时器输出控制比较器 0 的输出消隐。 0000: 无消隐 0001: 选择 TIMER0_CH0 输出比较信号为消隐源 0010: 选择 TIMER1_CH2 输出比较信号为消隐源 0011: 选择 TIMER2_CH2 输出比较信号为消隐源 0100: 选择 TIMER2_CH3 输出比较信号为消隐源 0101: 选择 TIMER7_CH0 输出比较信号为消隐源 0110: 选择 TIMER15_CH0 输出比较信号为消隐源 其它: 保留。
23:20	保留	必须保持复位值。
19:16	CMP0MSEL[3:0]	CMP0_IM 输入选择 该位域用来选择 CMP0 的 CMP0_IM 输入源选择。 0000: $V_{REFINT} / 4$ 0001: $V_{REFINT} / 2$ 0010: $V_{REFINT} * 3 / 4$ 0011: V_{REFINT} 0100: PA4 0101: PA5 0110: PA2 0111: DAC_OUT0 其它: 保留。
15:10	保留	必须保持复位值。
9:8	CMP0HST[1:0]	CMP0 迟滞 该位域用于控制迟滞水平 00: 无迟滞 01: 低迟滞 10: 中迟滞 11: 高迟滞
7	保留	必须保持复位值。
6	CMP0INTEN	CMP0 中断使能 0: 禁能 1: 使能
5:4	保留	必须保持复位值。
3	CMP0PL	CMP0 输出极性 该位是用来控制 CMP0 输出极性

		0: 输出是正相的 1: 输出是反相的
2	CMP0SEN	电压定标器使能位 当 CMP0 使能时, 该位必须置位 0: 如果 CMP0_CS 的 CMP0SEN 位也复位, 则禁用 V _{REFINT} 电压定标器 1: 启用电压定标器
1	CMP0BEN	定标器使能位 0: 如果 CMP0_CS 的 CMP0BEN 位也复位, 则禁用定标器电阻桥 1: 启用定标器电阻桥
0	CMP0EN	CMP0 使能 0: CMP0 禁能 1: CMP0 使能

21. 模数转换器（ADC）

21.1. 简介

MCU 片上集成了 12 位逐次逼近式模数转换器模块（ADC），可以采样来自于外部输入通道和内部通道的模拟信号。ADC0 有 16 个外部通道和 2 个内部通道（温度传感器 V_{SENSE} 和参考电压 V_{REFINT} ），ADC1 有 18 个外部通道，ADC2 有 17 个外部通道。

模拟看门狗允许应用程序来检测输入电压是否超出用户设定的高低阈值。

所有的 ADC 采样通道都支持多种运行模式，采样转换后，转换结果可以按照最低有效位对齐或最高有效位对齐的方式保存在相应的数据寄存器中。

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。

21.2. 主要特征

- 高性能：
 - ADC 采样分辨率：12 位、10 位、8 位、或者 6 位分辨率；
 - 可编程采样时间；
 - 数据存储模式：最高有效位对齐和最低有效位对齐；
 - 常规和注入序列都支持 DMA 请求。
- 模拟输入通道：
 - ADC0 有 16 个外部模拟输入通道，ADC1 有 18 个外部模拟输入通道，ADC2 有 17 个外部模拟输入通道；
 - 1 个内部温度传感器通道（ 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；
 - TRIGSEL 触发。
- 运行模式：
 - 转换单个通道，或者扫描一序列的通道；
 - 单次运行模式，每次触发转换一次选择的输入通道；
 - 连续运行模式，连续转换所选择的输入通道；
 - 间断运行模式；
 - 同步模式（适用于具有两个或多个 ADC 的设备）。
- 转换结果阈值监测功能：模拟看门狗。
- 中断产生：
 - 序列转换结束；
 - 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2x到256x；
 - 高达8位的可编程数据移位。
- ADC输入范围： $V_{REFN} \leq V_{IN} \leq V_{REFP}$ 。

21.3. 引脚和内部信号

[图 21-1. ADC 模块框图](#) 给出了 ADC 模块框图。[表 21-1. ADC 内部信号](#)给出了 ADC 内部信号。[表 21-2. ADC 引脚定义](#)给出了 ADC 引脚说明。

表 21-1. ADC 内部信号

内部信号名称	说明
V_{SENSE}	内部温度传感器输出电压
V_{REFINT}	内部参考输出电压

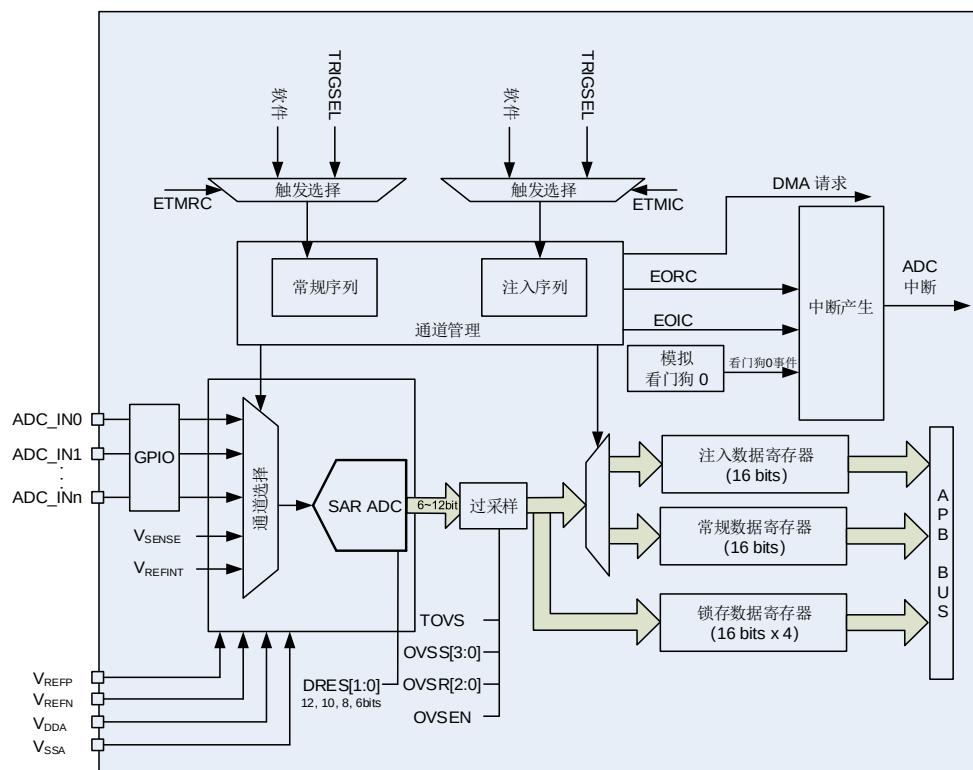
表 21-2. ADC 引脚定义

名称	注释
V_{DDA}	模拟电源输入,等于 V_{DD}
V_{SSA}	模拟地，等于 V_{SS}
V_{REFP}	ADC正参考电压
V_{REFN}	ADC负参考电压
ADCx_IN[17:0]	多达18路外部通道

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

21.4. 功能描述

图 21-1. ADC 模块框图



21.4.1. ADC 时钟

ADC 最大工作时钟频率为 42MHz。CK_ADC 时钟是由时钟控制器提供的，它和 AHB、APB2 时钟保持同步。ADC 时钟可以在 RCU 时钟控制器中进行分频和配置。

想要更多 ADC 时钟产生的信息，可以参考 RCU 章节内容。

21.4.2. ADC 使能

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 t_{su} 时间后才能采样， t_{su} 数值详见芯片相关型号 Datasheet。

21.4.3. 常规序列和注入序列

通道管理电路把采样通道组织成两个序列：一个常规序列和一个注入序列。

常规序列支持高达 16 个通道，每个通道称为常规通道。ADC_RSQ0 寄存器的 RL[3:0]位规定了整

个常规序列的长度。ADC_RSQ0~ADC_RSQ2 寄存器规定了常规序列的通道选择。

注入序列支持高达 4 个通道，每个通道称为注入通道。ADC_ISQ 寄存器的 IL[1:0]位规定了整个注入序列的长度。ADC_ISQ 寄存器规定了注入序列的通道选择。

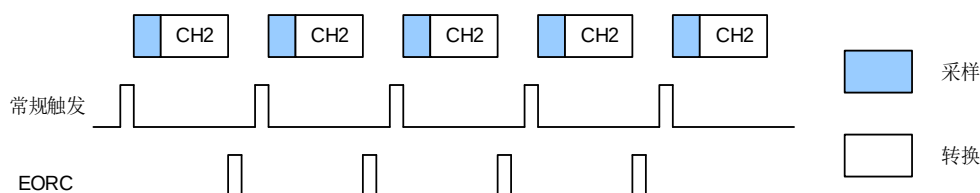
注意：尽管 ADC 支持 18 个通道，但常规序列最大长度为 16 个通道，注入序列最大长度为 4 个通道。

21.4.4. 运行模式

单次运行模式

该模式能够运行在常规序列和注入序列。单次运行模式下，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 TRIGSEL 触发发生，ADC 就会采样和转换一个通道。

图 21-2. 单次运行模式



常规序列的通道单次转换结束后，转换数据将被存放于 ADC_RDATA 寄存器中，EORC 将会置 1。如果 EORCIE 位被置 1，将产生一个中断。

注入序列的通道单次转换结束后，转换数据将被存放于 ADC_IDATA 寄存器中，EOIC 位将会置 1。如果 EOICIE 位被置 1，将产生一个中断。

常规序列单次运行模式的软件流程：

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[4:0]位域；
3. 配置ADC_SAMPTx寄存器；
4. 如果有需要，可以配置ADC_CTL1寄存器的ETMRC[1:0]位域；
5. 设置SWRCST位，或者为常规序列产生一个TRIGSEL触发信号；
6. 等到EORC置1；
7. 从ADC_RDATA寄存器中读ADC转换结果；
8. 写0清除EORC标志位。

注入序列单次运行模式的软件流程：

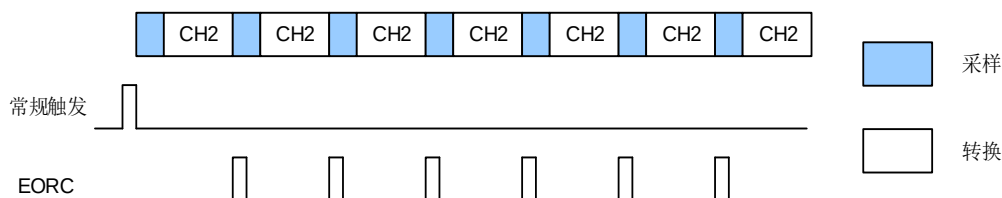
1. 确保ADC_CTL0寄存器的DISIC和SM位为0；
2. 用模拟通道编号来配置ISQ3[4:0]位域；
3. 配置ADC_SAMPTx寄存器；
4. 如果有需要，可以配置ADC_CTL1寄存器的ETMIC[1:0]位域；

5. 设置SWICST位，或者为注入序列产生一个TRIGSEL触发信号；
6. 等到EOIC置1；
7. 从ADC_IDATA寄存器中读ADC转换结果；
8. 写0清除EOIC标志位。

连续运行模式

该模式能够运行在常规序列。将 ADC_CTL1 寄存器的 CTN 位置 1 可以使能连续运行模式。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 TRIGSEL 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 21-3. 连续运行模式



常规序列连续运行模式的软件流程：

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[4:0]；
3. 配置ADC_SAMPTx寄存器；
4. 如果有需要，配置ADC_CTL1寄存器的ETMRC[1:0]位域；
5. 设置SWRCST位，或者给常规序列产生一个TRIGSEL触发信号；
6. 等待EORC标志位置1；
7. 从ADC_RDATA寄存器中读ADC转换结果；
8. 写0清除EORC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，不需循环查询 EORC 标志位，软件流程如下：

1. 设置ADC_CTL1寄存器的CTN和RDMA位为1；
2. 根据模拟通道编号配置RSQ0[4:0]；
3. 配置ADC_SAMPTx寄存器；
4. 如果有需要，配置ADC_CTL1寄存器的ETMRC[1:0]位域；
5. 准备DMA模块，用于传输来自ADC_RDATA的数据；
6. 设置SWRCST位，或者给常规序列产生一个TRIGSEL触发。

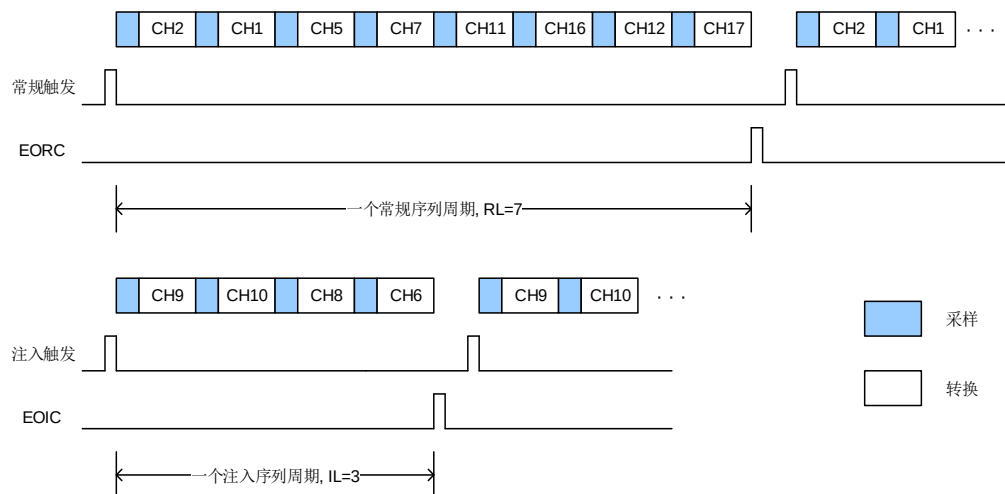
扫描运行模式

扫描运行模式可以通过将 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 TRIGSEL 触发产生，ADC 就会一个接一个的采样和转换常规序列

或注入序列通道。转换数据存储在 ADC_RDATA 或 ADC_IDATA 寄存器中。常规序列或注入序列转换结束后，EORC 或者 EOIC 位将被置 1。如果 EORCIE 或 EOICIE 位被置 1，将产生中断。当常规或者注入序列工作在扫描模式下时，可以通过 ADC_CTL1 寄存器的 RDMA 或者 IDMA 位置 1 使能相应的 DMA 功能。

如果 ADC_CTL1 寄存器的 CTN 位也被置 1，则在常规序列转换完之后，这个转换自动重新开始。

图 21-4. 扫描运行模式，且连续运行模式禁能



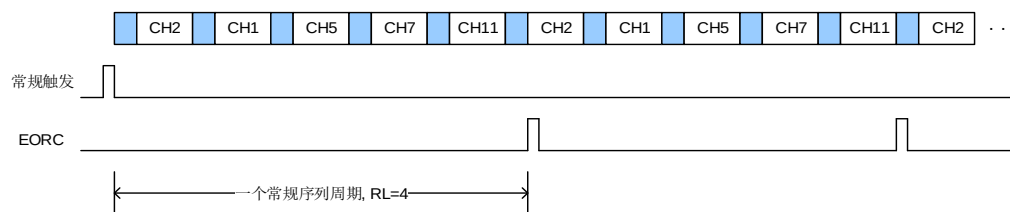
常规序列扫描运行模式的软件流程：

1. 设置 ADC_CTL0 寄存器的 SM 位和 ADC_CTL1 寄存器的 RDMA 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 ETMRC[1:0]位；
4. 准备 DMA 模块，用于传输来自 ADC_RDATA 的数据；
5. 设置 SWRCST 位，或者给常规序列产生一个 TRIGSEL 触发；

注入序列扫描运行模式的软件流程：

1. 设置 ADC_CTL0 寄存器的 SM 位和 ADC_CTL1 寄存器的 IDMA 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 ETMIC[1:0]位；
4. 设置 SWICST 位，或者给注入序列产生一个 TRIGSEL 触发；
5. 准备 DMA 模块，用于传输来自 ADC_IDATA 的数据；

图 21-5. 扫描运行模式，连续运行模式使能



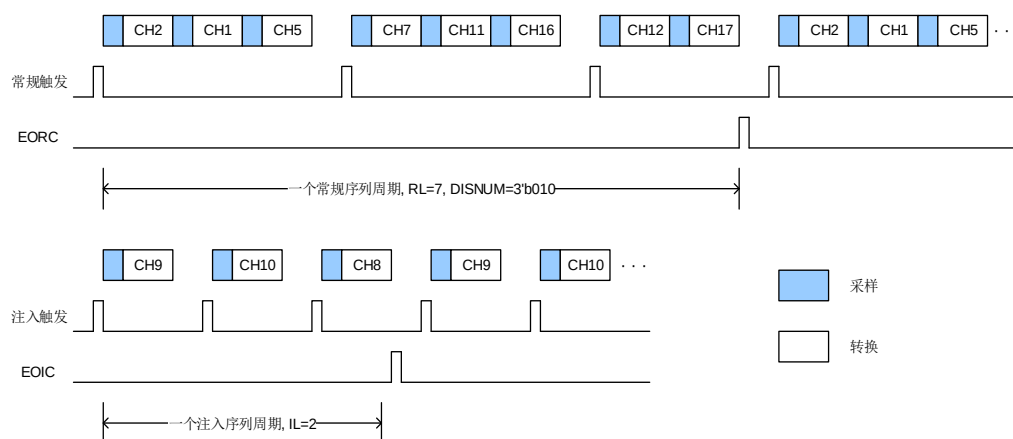
中断运行模式

对于常规序列，当 ADC_CTL0 寄存器的 DISRC 位置 1 时，常规序列中断运行模式使能。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 DISNUM[2:0] 位给出。当相应的软件触发或 TRIGSEL 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，直到常规序列中所有的通道转换完成。每个常规序列转换周期结束后，EORC 位将置 1。如果 EORCIE 位置 1 将产生一个中断。

对于注入序列，当 ADC_CTL0 寄存器的 DISIC 位置 1 时，注入序列中断运行模式使能。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 TRIGSEL 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，直到注入序列中所有通道转换完成。每个注入序列转换周期结束后，EOIC 位将被置 1。如果 EOICIE 位被置 1 将产生一个中断。

常规序列和注入序列不能同时工作在间断模式，同一时刻只能有一个序列被设置成间断模式。

图 21-6. 间断运行模式



常规序列中断模式的软件流程：

1. 设置 ADC_CTL0 寄存器的 DISRC 位和 ADC_CTL1 寄存器的 RDMA 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 ETMRC[1:0] 位；
5. 准备 DMA 模块，用于传输来自 ADC_RDATA 的数据；
6. 设置 SWRCST 位，或者给常规序列产生一个 TRIGSEL 触发；
7. 如果需要，重复步骤 6；

注入序列中断模式的软件流程：

1. 设置 ADC_CTL0 寄存器的 DISIC 位和 ADC_CTL1 寄存器的 IDMA 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 ETMIC[1:0] 位；

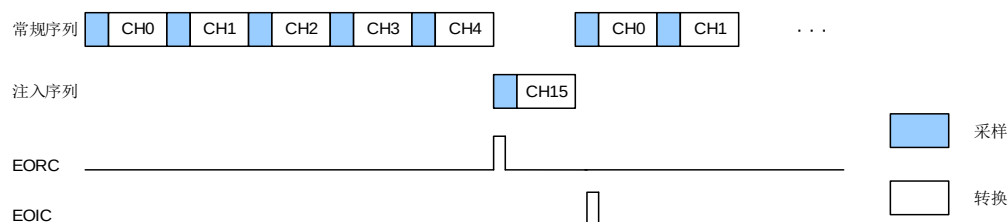
4. 准备 DMA 模块，用于传输来自 ADC_IDATA 的数据；
5. 设置 SWICST 位，或者给注入序列产生一个 TRIGSEL 触发；
6. 如果需要，重复步骤 5；

21.4.5. 注入序列管理

自动注入

如果将 ADC_CTL0 寄存器的 ICA 位置 1，在常规序列之后，注入序列被自动转换。该模式下注入序列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 个通道。除了 ICA 位之外，如果 CTN 位也被置 1，常规序列将在注入序列之后被自动转换。

图 21-7. 自动注入，CTN=1

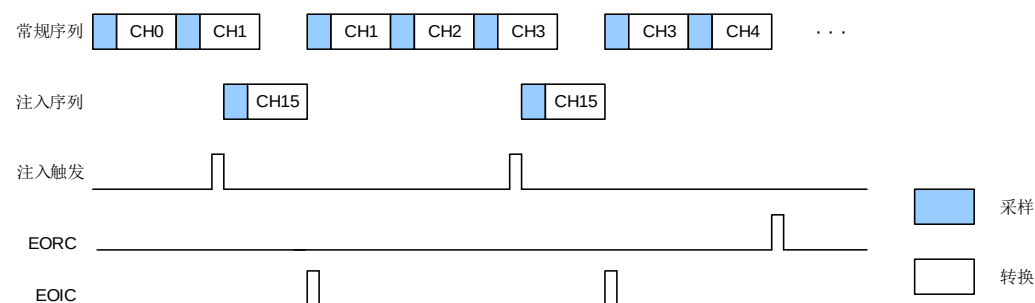


不能同时使用自动注入和间断模式。

触发注入

清除 ICA 位，在常规序列转换期间如果注入序列的软件触发或者 TRIGSEL 触发发生，则启动注入序列转换。这种情况下，ADC 取消常规序列中当前正在转换的通道，注入序列进行转换。注入序列转换结束后，ADC 从上次被取消的常规序列通道处重新开始转换。

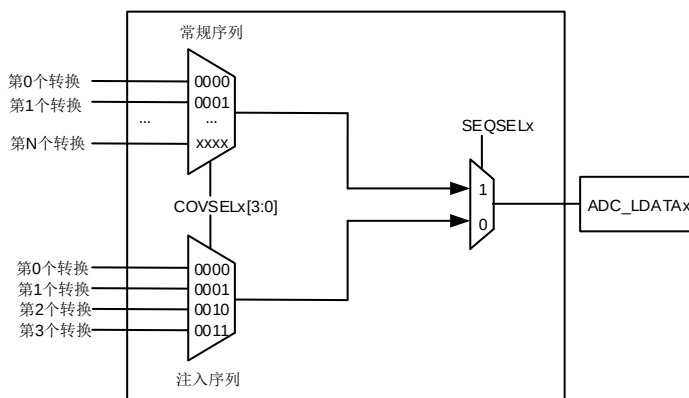
图 21-8. 触发注入



21.4.6. 转换数据锁存

ADC 拥有 4 个锁存数据寄存器，ADC_LDATAB_x (x=0...3)，这些寄存器可以在常规或注入序列通道转换完成后锁存数据。锁存数据寄存器的控制逻辑如 [图 21-9. 数据锁存](#)。

图 21-9. 数据锁存

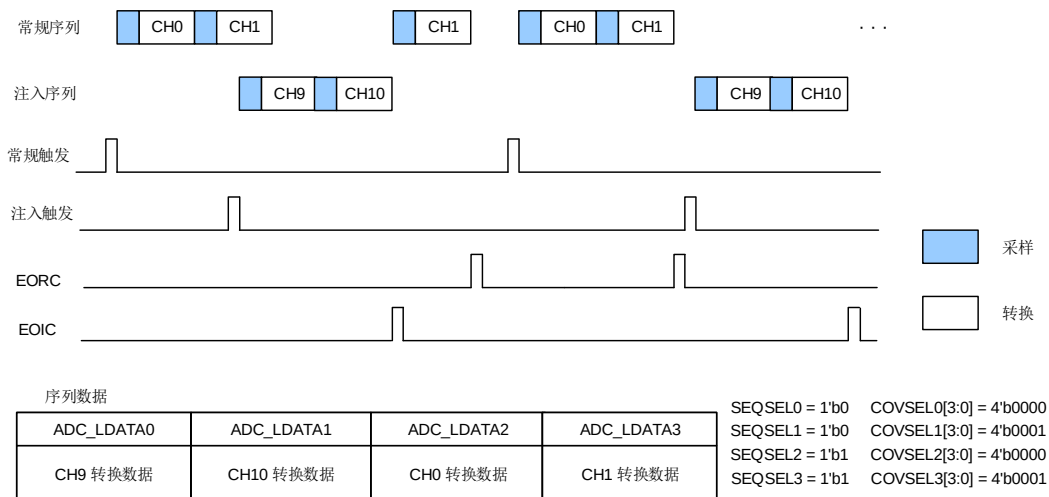


是从常规序列还是从注入序列中选择要锁存的数据,可以通过 `ADC_LDCTL` 寄存器中的 `SEQSELx` ($x=0\ldots3$) 位来决定,而 `COVSELx` 位则选择要锁存序列中的哪个转换结果。

设置 `ADC_LDATABx` ($x=0\ldots3$) 寄存器来存储注入序列或常规序列的转换结果,默认存储注入序列的第 x 次转换结果。

因此,除了使用 DMA 从 `ADC_RDATA` 或 `ADC_IDATA` 寄存器传输转换数据外,还可以直接从锁存数据寄存器读取数据,这为序列的连续转换提供了更大的灵活性。

图 21-10. `ADC_LDATABx` 读取序列数据



触发注入模式下从 `ADC_LDATABx` 读取序列数据的软件流程:

1. 设置 `ADC_CTL0` 寄存器的 `SM` 位;
2. 配置 `ADC_RSQx`, `ADC_ISQ` 和 `ADC_SAMPTx` 寄存器;
3. 配置 `ADC_LDCTL` 寄存器的 `SEQSEL2`, `COVSEL2[3:0]`, `SEQSEL3` 和 `COVSEL3[3:0]` 位域用于常规序列;
4. 配置 `ADC_LDCTL` 寄存器的 `SEQSEL0`, `COVSEL0[3:0]`, `SEQSEL1` 和 `COVSEL1[3:0]` 位域用于注入序列;
5. 如果有需要,配置 `ADC_CTL1` 寄存器中的 `ETMRC[1:0]` 和 `ETMIC[1:0]` 位域;

6. 设置 SWRCST 位，或者给常规序列产生一个 TRIGSEL 触发；
7. 设置 SWICST 位，或者给注入序列产生一个 TRIGSEL 触发；
8. 等待 EORC 或 EOIC 标志位置 1；
9. 如果 EORC 标志位置 1，从 ADC_LDATAL2 和 ADC_LDATAL3 寄存器中读常规序列 ADC 转换结果；
10. 写 0 清除 EORC 标志位。
11. 如果 EOIC 标志位置 1，从 ADC_LDATAL0 和 ADC_LDATAL1 寄存器中读注入序列 ADC 转换结果；
12. 写 0 清除 EOIC 标志位。

21.4.7. 转换结果阈值监测功能

模拟看门狗 0

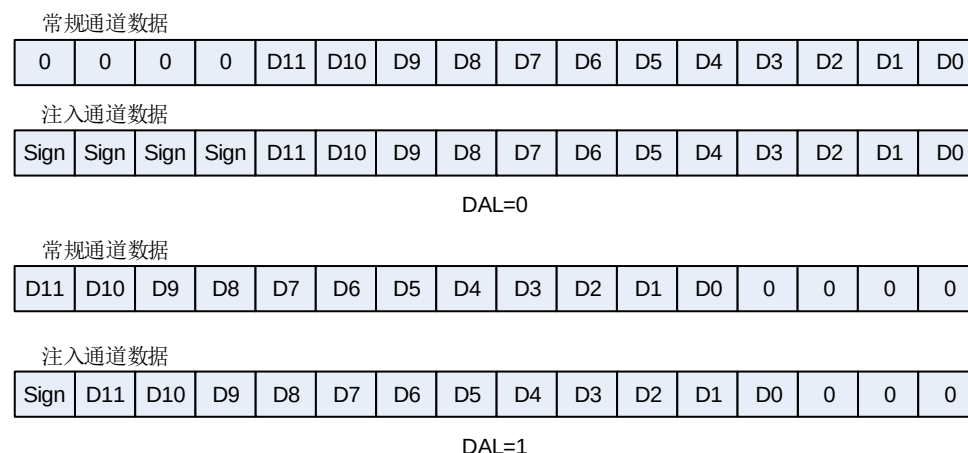
ADC_CTL0 寄存器的 RWD0EN 和 IWD0EN 位置 1 将分别使能常规序列和注入序列的模拟看门狗 0 功能。该功能用于监测转换结果是否超过设定的阈值。如果 ADC 的模拟转换电压低于低阈值或高于高阈值，ADC_STAT 状态寄存器的 WD0E 位将被置 1。如果 WD0EIE 位被置 1，将产生中断。ADC_WD0HT 和 ADC_WD0LT 寄存器用来设定高低阈值。内部数据的比较在对齐之前完成，因此阈值与 ADC_CTL1 寄存器的 DAL 位确定的对齐方式无关。ADC_CTL0 寄存器的 RWD0EN，IWD0EN，WD0SC 和 WD0CHSEL[4:0]位可以用来选择模拟看门狗 0 监控单一通道或者多通道。

21.4.8. 数据存储模式

ADC_CTL1 寄存器的 DAL 位确定转换后数据存储的对齐方式。

注入序列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。

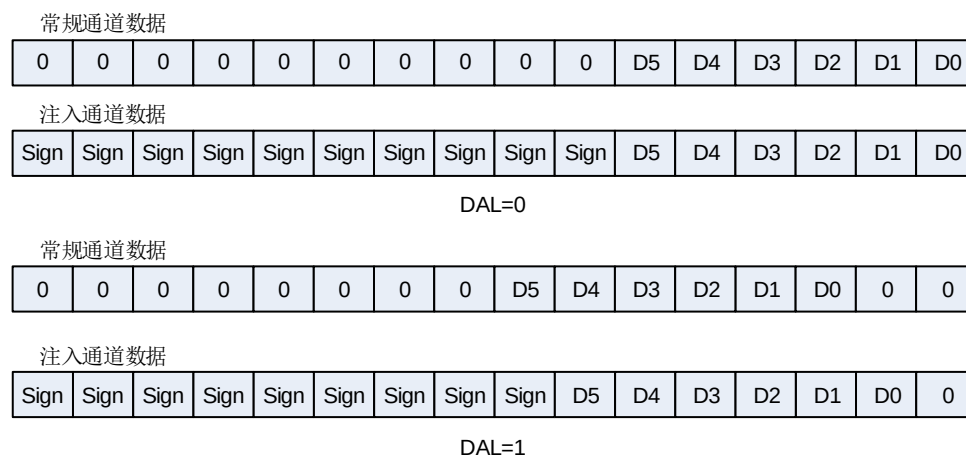
图 21-11. 12 位数据存储模式



6 位分辨率的数据存储模式不同于 12 位/10 位/8 位分辨率数据存储模式，如 [图 21-12. 6 位数据存](#)

储模式。

图 21-12. 6 位数据存储模式



21.4.9. 采样时间配置

ADC 使用若干个 CK_ADC 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，总采样转换时间=采样时间+12.5 个 CK_ADC 周期。

例如：

CK_ADC = 42MHz，采样时间为 1.5 个周期，那么总的转换时间为：“1.5+12.5”个 CK_ADC 周期，即 0.333us。

21.4.10. 外部触发

TRIGSEL 触发输入的上升沿或者软件触发可以触发常规序列或注入序列的转换。ADC_CTL1 寄存器的 ETMRC[1:0]和 ETMIC[1:0]分别控制常规序列或注入序列的触发模式。

表 21-3.外部触发模式和触发类型

ETMRC[1:0]/ETMIC[1:0]	触发模式	触发类型
00, 01, 10	外部触发使能	硬件触发： 来自TRIGSEL的信号
11	外部触发禁能	软件触发： SWRCST/SWICST

21.4.11. DMA 请求

DMA 请求，可以通过设置 ADC_CTL1 寄存器的 RDMA 或者 IDMA 位来使能，它用于传输常规或者注入序列多个通道的转换结果。ADC 在常规或者注入序列的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 或者 ADC_IDATA 寄存器传输到用户指定的目的地址。

21.4.12. ADC 内部通道

将 ADC_CTL1 寄存器的 TSVEN 位置 1 可以使能温度传感器通道 (ADC0_IN16)，将 ADC_CTL1 寄存器的 INREFEN 位置 1 可以使能 VREFINT 通道 (ADC0_IN17)。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 t_{s_temp} μs （具体数值请参考 Datasheet）。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会发生线性变化，由于芯片生产过程的多样化，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。

使用温度传感器：

1. 配置温度传感器通道 (ADC0_IN16) 的转换序列和采样时间大于 t_{s_temp} μs ；
2. 置位 ADC_CTL1 寄存器的 TSVEN 位，使能温度传感器；
3. 置位 ADC_CTL1 寄存器的 ADCON 位，或者由 TRIGSEL 触发启动 ADC 转换；
4. 读取内部温度传感器输出电压 $V_{temperature}$ ，并由下面公式计算出实际温度：

$$\text{温度 (}^{\circ}\text{C)} = \frac{V_{25} - V_{temperature}}{\text{Avg\_Slope}} + 25 \quad (21-1)$$

$V_{temperature}$ ：温度传感器的输出电压。

V_{25} ：内部温度传感器在 25°C 时的输出电压，典型值请参考相关型号 Datasheet。

Avg_Slope：温度与内部温度传感器输出电压曲线的均值斜率，典型值请参考相关型号 datasheet。

内部电压参考 (VREFINT) 提供了一个稳定的（带隙基准）电压输出给 ADC 和比较器。VREFINT 内部连接到 ADC0_IN17 输入通道。

21.4.13. 可编程分辨率 (DRES)

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 位为 0 时，才能修改 DRES[1:0] 的值。ADC 转换的结果只有 12 位，其余没有被用到的低位读出来都是为 0。较低的分辨率能够减少逐次逼近步骤所需的转换时间，如 [表 21-4. 不同分辨率对应的 \$t_{CONV}\$ 时间](#) 所示。

表 21-4. 不同分辨率对应的 t_{CONV} 时间

DRES[1:0] bits	t_{CONV} (ADC clock)	$t_{CONV}(ns)$ at $f_{ADC}=42MHz$	$t_s(min)$ (ADC clock)	t_{ADC} (ADC clock)	$t_{ADC}(ns)$ at $f_{ADC}=42MHz$

	cycles)		cycles)	cycles)	
12	12.5	298 ns	1.5	14	333 ns
10	10.5	250 ns	1.5	12	286 ns
8	8.5	202 ns	1.5	10	238 ns
6	6.5	155 ns	1.5	8	190 ns

21.4.14. 片上硬件过采样

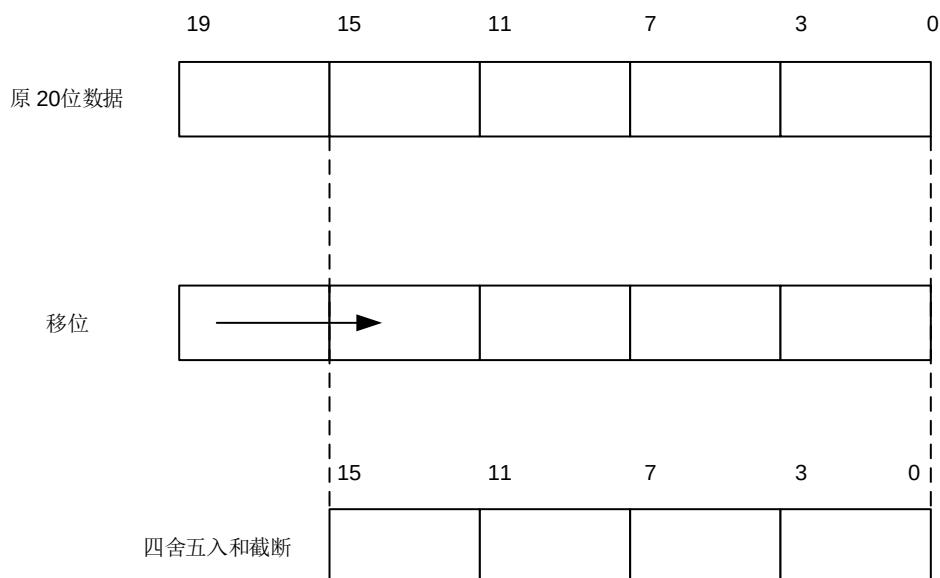
片上硬件过采样单元执行数据预处理以减轻 CPU 负担。它能够处理多个转换，并将多个转换的结果取平均，得出一个 16 位宽的数据。其结果根据如下公式计算得出，其中 N 和 M 的值可以被调整。D_{out}(n) 是指 ADC 输出的第 n 个数字信号：

$$\text{Result} = \frac{1}{M} * \sum_{n=0}^{N-1} D_{\text{out}}(n) \quad (21-2)$$

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 OVSRR[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。

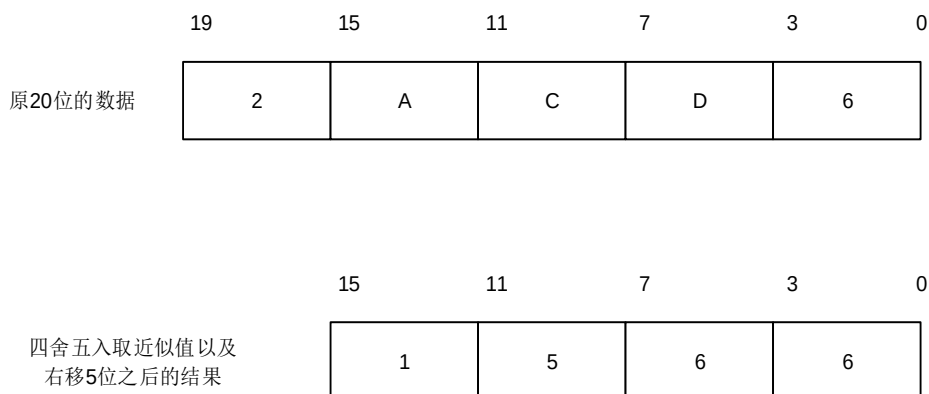
图 21-13. 20 位到 16 位的结果截断



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

[图 21-14. 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 21-14. 右移 5 位和取整的数例



[表 21-5. N 和 M 的最大输出值（灰色部分表示截断）](#)给出了 N 和 M 各种组合的数据格式，初始转换值为 0xFFFF。

表 21-5. N 和 M 的最大输出值（灰色部分表示截断）

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	0xFFF0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF
32x	0x1FFE0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF
64x	0x3FFC0	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF
128x	0x7FF80	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF
256x	0xFFF00	0xFF00	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF

和标准的运行模式相比，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：

$$N \times t_{\text{ADC}} = N \times (t_{\text{SMPL}} + t_{\text{CONV}}) \quad (21-3)$$

21.5. ADC 同步模式

在具有两个或三个 ADC 的设备上，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，转换的启动可以是

ADC0 主和 ADC1 从的交替触发或同步触发。在同步模式下，ADC0 配置为 TRIGSEL 触发转换时，ADC1 必须配置为软件触发。

ADC 同步模式如[表 21-6. ADC 同步模式表](#)所示。

在 ADC 同步模式下，即使不使用 DMA 功能，也要将 RDMA 置位。ADC1 的转换数据可以通过 ADC0 的常规数据寄存器（ADC0_RDATA）读取。

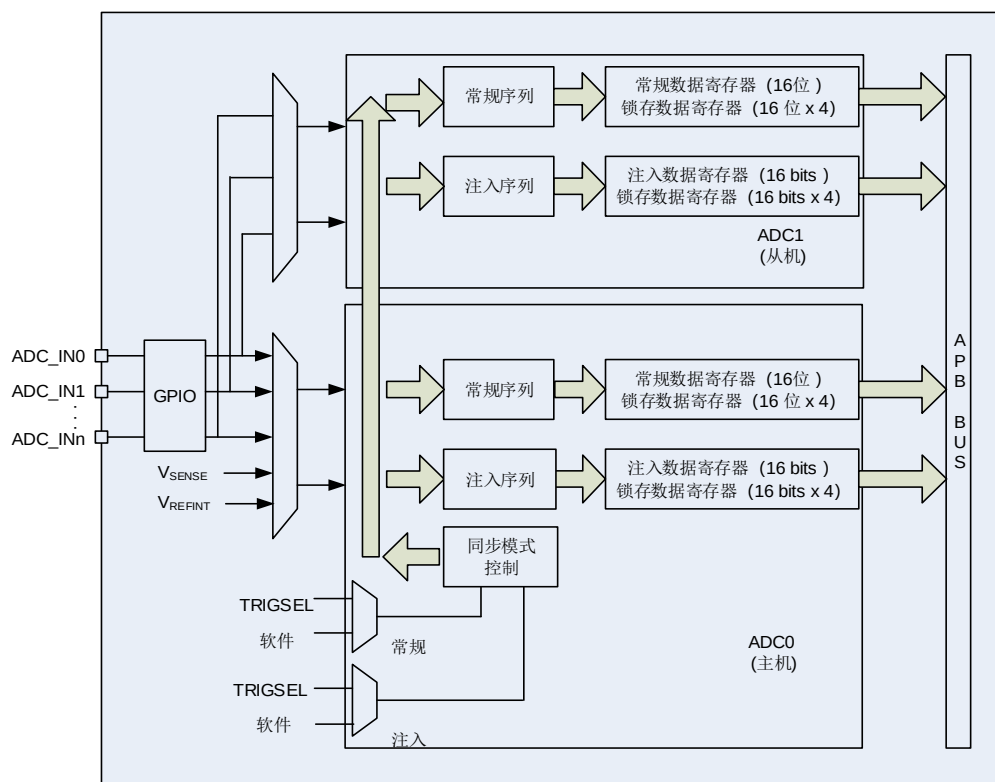
表 21-6. ADC 同步模式表

SYNCR[3: 0]	模式
0000	独立模式。所有的ADC都独立工作。
0001	ADC0和ADC1工作在常规并行和注入并行组合模式。
0010	ADC0和ADC1工作在常规并行和注入交替触发组合模式。
0011	ADC0和ADC1工作在注入并行和常规快速交叉组合模式。
0100	ADC0和ADC1工作在注入并行和常规慢速交叉组合模式。
0101	ADC0和ADC1工作在注入并行模式。
0110	ADC0和ADC1工作在常规并行模式。
0111	ADC0和ADC1工作在常规快速交叉模式。
1000	ADC0和ADC1工作在常规慢速交叉模式。
1001	ADC0和ADC1工作在注入交替触发模式。

当 ADC 工作在同步模式，而非独立模式时，如果需要再将 ADC 配置成其他同步模式，则需要先在配置成其他同步模式前，首先将 ADC 配置成独立模式。注入数据寄存器（ADC_IDATA）不能用于 DMA 传输。

ADC 同步框图如[图 21-15. ADC 同步框图](#)所示。

图 21-15. ADC 同步框图



21.5.1. 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。

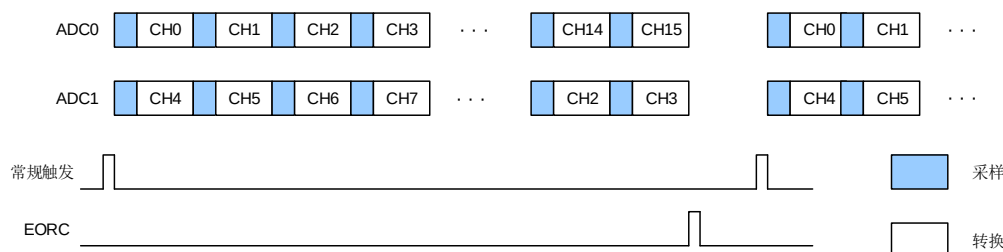
21.5.2. 常规并行模式

设置 ADC0_CTL0 寄存器中 SYNCNM[3:0] 位为 0110，使能常规并行模式。在常规并行模式中，根据 ADC0 中选择的外部触发，ADC0 和 ADC1 并行的转换常规通道。触发选择由 ADC0 的 ADC_CTL1 寄存器 ETMRC[1:0] 位进行配置。

在转换结束时产生 EORC 中断（如果 ADC 使能了该中断）。常规并行模式的行为如 [图 21-16. 基于 16 个通道的常规并行模式](#) 所示。

32 位 ADC0_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 ADC0_RDATA 中的数据传送到 SRAM。

图 21-16. 基于 16 个通道的常规并行模式



注意：

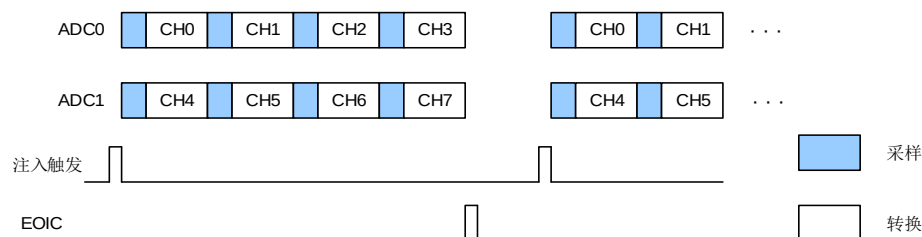
1. 两个 ADC 不能同时转换同一个通道。（当两个 ADC 转换同一通道时，采样时间不可重叠）
2. 在并行模式下，ADC0 和 ADC1 并行采样的两个通道需要设置为相同的采样时间。
3. 确保设置 ADCON 位为 1 前进行 ADC 模块复位。

21.5.3. 注入并行模式

设置 ADC_CTL0 寄存器中 SYNCNM[3:0] 位为 0101，使能注入并行模式。在注入并行模式中，根据 ADC0 中选择的外部触发，ADC0 和 ADC1 并行的转换注入通道。触发选择由 ADC0 的 ADC_CTL1 寄存器 ETMIC[1:0] 位进行配置。

在注入序列转换结束时产生 EOIC 中断（如果 ADC 使能了该中断）。转换的数据都存储在每个 ADC 的 ADC_IDATA 寄存器中。注入并行模式的行为如 [图 21-17. 4 个通道的注入并行模式](#) 所示。

图 21-17. 4 个通道的注入并行模式



注意：

1. 两个 ADC 不能同时转换同一个通道。（当两个 ADC 转换同一通道时，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需要设置为相同的采样时间。
3. 注入数据寄存器（ADC_IDATA）不能用于 DMA 传输。

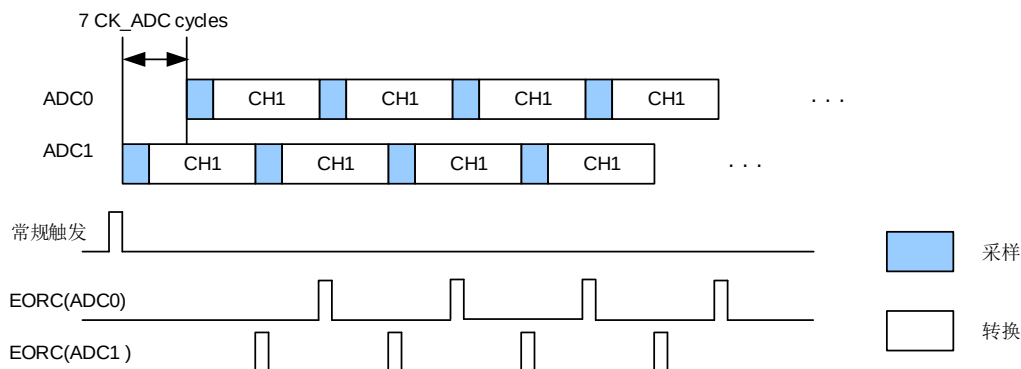
21.5.4. 常规快速交叉模式

此模式应用于 ADC 的常规序列（通常采样同一个通道），设置 ADC_CTL0 寄存器中 SYNCNM[3:0] 位为 0111，使能快速交叉模式。ADC0 中选择的外部触发产生时，ADC1 立刻启动，而 ADC0 在 7 个 ADC 时钟周期后启动。触发选择由 ADC0 的 ADC_CTL1 寄存器 ETMRC[1:0] 位进行配置。

如果 ADC0 和 ADC1 的 CTN 位被置位，所选的常规序列通道将在两个 ADC 中被不停的转换。

32 位 ADC0_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 EORC 中断后（如果 ADC 使能了该中断），可通过 32 位 DMA 将 ADC0_RDATA 中数据传送到 SRAM。

图 21-18. 常规序列上的快速交叉模式（两个 ADC 的 CTN=1）



注意：

1. 可允许的最大采样时间必须小于 7 个 CK_ADC 采样时钟，从而避免 ADC0 和 ADC1 在采样相同通道时出现采样时间重叠。
2. 确保设置 ADCON 位为 1 前先进进行 ADC 模块复位。

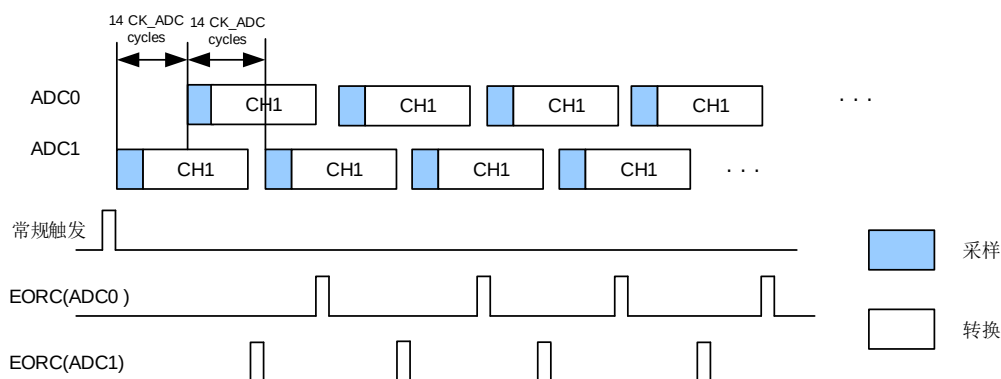
21.5.5. 常规慢速交叉模式

此模式应用于 ADC 的常规序列（通常采样同一个通道），设置 ADC_CTL0 寄存器中 SYNCM[3:0] 位为 1000，使能慢速交叉模式。ADC0 中选择的外部触发产生时，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。触发选择由 ADC0 的 ADC_CTL1 寄存器 ETMRC[1:0] 位进行配置。

在这种模式下，不能使用连续运行模式，因为在这种模式下所选的常规序列通道在两个 ADC 中被不停的转换，如 [图 21-19. 常规序列上的慢速交叉模式](#) 所示。

32 位 ADC0_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 EORC 中断后（如果 ADC 使能了该中断），可通过 32 位 DMA 将 ADC0_RDATA 中数据传送到 SRAM。

图 21-19. 常规序列上的慢速交叉模式



注意：

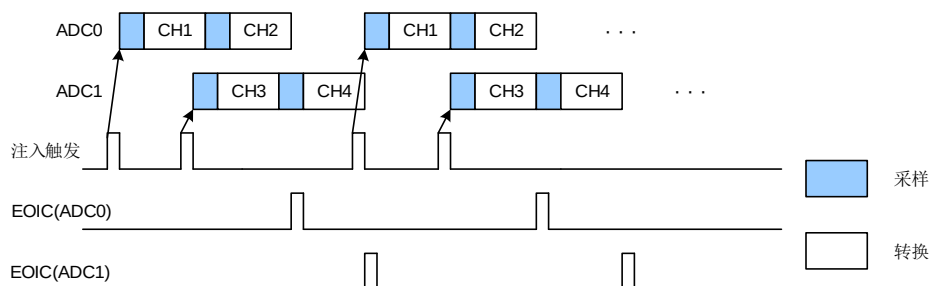
1. 可允许的最大采样时间必须小于 14 个 CK_ADC 采样时钟，从而避免 ADC0 和 ADC1 在转换相同通道时出现采样时间重叠。
2. 确保设置 ADCON 位为 1 前先进进行 ADC 模块复位。

21.5.6. 注入交替触发模式

此模式应用于注入序列，设置 ADC_CTL0 寄存器中 SYNCM[3:0] 位为 1001，使能注入交替触发模式。如果 ADC0 和 ADC1 注入序列间断运行模式禁能，ADC0 中选择的外部触发第一次产生时，ADC0 注入序列的所有通道被转换，当第二次触发产生，ADC1 注入序列的所有通道被转换。触发选择由 ADC0 的 ADC_CTL1 寄存器 ETMIC[1:0] 位进行配置。

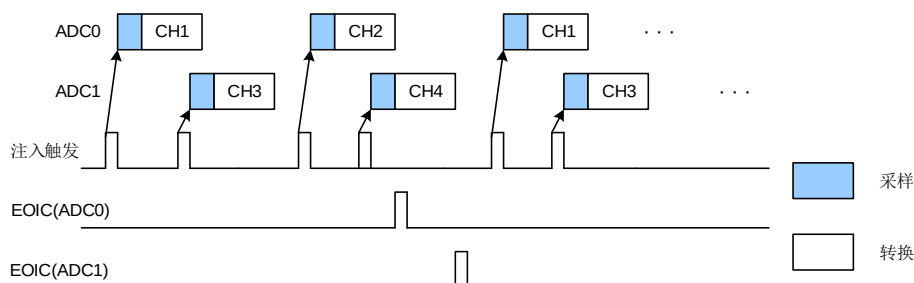
在注入序列转换结束时产生 EOIC 中断（如果 ADC 使能了该中断）。注入交替触发模式在 DISIC=0 时的行为如 [图 21-20. 注入交替触发模式：DISIC=0, IL=1](#) 所示。

图 21-20. 注入交替触发模式：DISIC=0, IL=1



如果 ADC0 和 ADC1 注入序列间断运行模式使能，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。注入交替触发模式在 DISIC=1 时的行为如 [图 21-21. 注入交替触发模式：DISIC=1, IL=1](#) 所示。

图 21-21. 注入交替触发模式：DISIC=1, IL=1

**注意：**

1. 两个 ADC 不能同时转换同一个通道。（当两个 ADC 转换同一通道时，采样时间不可重叠）
2. 确保设置 ADCON 位为 1 前先进进行 ADC 模块复位。
3. 注入数据寄存器（ADC_IDATA）不能用于 DMA 传输。

21.5.7. 常规并行和注入并行组合模式

设置 ADC_CTL0 寄存器中 SYNCNM[3:0]位为 0001，使能常规并行和注入并行组合模式。在常规并行和注入并行组合同步模式中，注入序列的并行转换可以中断常规并行转换。

在常规序列转换结束时产生 EORC 中断（如果 ADC 使能了该中断）。

在注入序列转换结束时产生 EOIC 中断（如果 ADC 使能了该中断）。

注意：

1. 在常规并行和注入并行组合模式下，ADC0 和 ADC1 并行采样的两个通道需要设置为相同的采样时间。
2. 确保设置 ADCON 位为 1 前先进进行 ADC 模块复位。
3. 注入数据寄存器（ADC_IDATA）不能用于 DMA 传输。

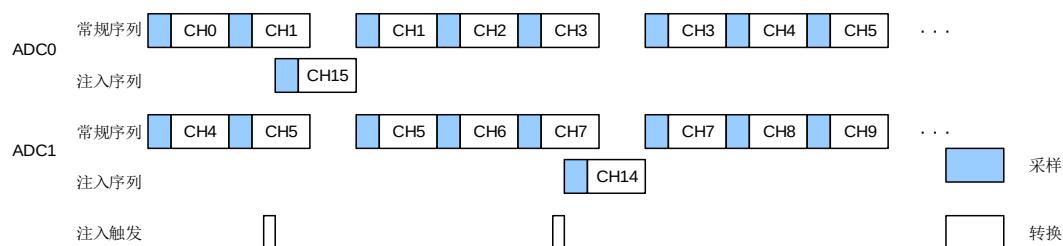
21.5.8. 常规并行和注入交替触发组合模式

设置 ADC_CTL0 寄存器中 SYNCNM[3:0]位为 0010，使能常规并行和交替触发组合模式。在常规并行和注入交替触发组合模式中，注入序列的交替触发转换可以中断常规并行转换。当注入序列触发出现时，ADC 的注入序列转换立即开始。当常规序列转换被中断，ADC 的常规序列转换会停止在注入序列触发的时刻，并且在注入序列转换结束时，从被中断的位置恢复到并行模式。常规序列转换被注入交替触发中断的行为如 [图 21-22. 常规并行和交替触发组合模式](#) 所示。

在常规序列转换结束时产生 EORC 中断（如果 ADC 使能了该中断）。

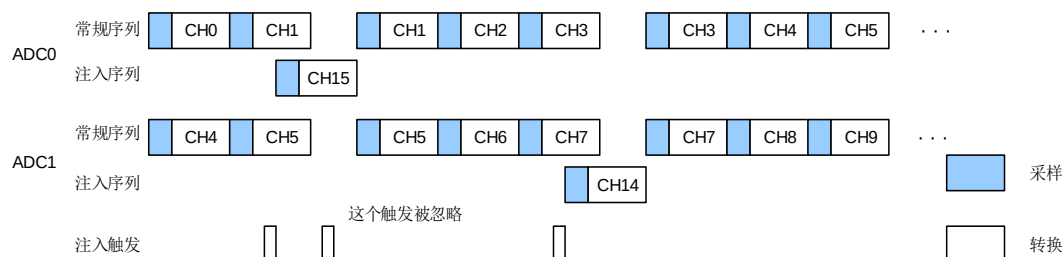
在注入序列转换结束时产生 EOIC 中断（如果 ADC 使能了该中断）。

图 21-22. 常规并行和交替触发组合模式



如果在一个注入序列转换期间, 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 21-23. 在注入序列转换过程中注入触发出现](#) 所示。

图 21-23. 在注入序列转换过程中注入触发出现



注意:

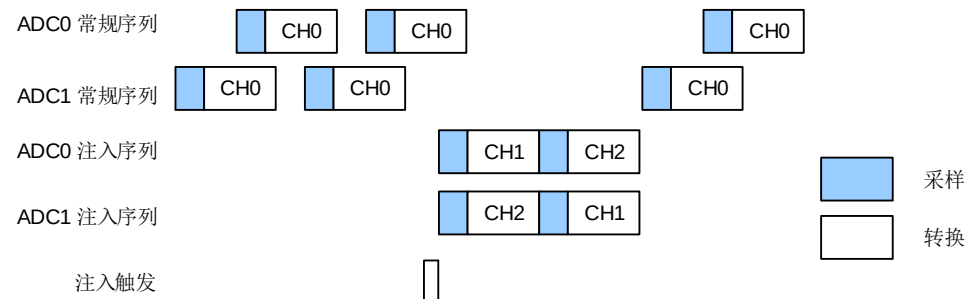
1. 确保设置 `ADCON` 位为 1 前先进进行 ADC 模块复位。
2. 注入数据寄存器 (`ADC_IDATA`) 不能用于 DMA 传输。

21.5.9. 注入并行和交叉组合模式

设置 `ADC_CTL0` 寄存器中 `SYNCR[3:0]` 位为 0011 或 0100, 使能注入并行和快速 (慢速) 交叉组合模式。

注入并行转换可以中断 (快速和慢速) 交叉转换, 当注入触发发生时, 交叉转换被中断, 注入序列转换启动。在注入序列转换完成后, 交叉转换恢复, 如 [图 21-24. 单通道交叉转换被注入序列中断](#) 所示。

图 21-24. 单通道交叉转换被注入序列中断



注意:

1. 确保设置 **ADCON** 位为 1 前先进进行 **ADC** 模块复位。
2. 注入数据寄存器 (**ADC_IDATA**) 不能用于 **DMA** 传输。

21.6. 中断

以下任一个事件发生都可以产生中断：

- 常规序列和注入序列转换结束；
- 模拟看门狗事件；

单独的中断使能位可使得使用更灵活。**ADC0**、**ADC1** 被映射到同一个中断向量 **IRQ18**。**ADC2** 都被映射到中断向量 **IRQ47**。

21.7. ADC 寄存器

ADC0 基地址：0x4001 2400

ADC1 基地址：0x4001 2800

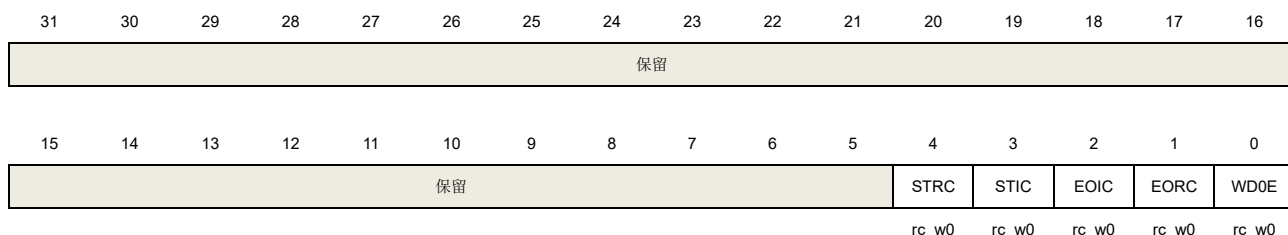
ADC2 基地址：0x4001 3C00

21.7.1. 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



位/位域	名称	说明
31:5	保留	必须保持复位值。
4	STRC	常规序列转换开始标志 0：常规序列转换没有开始 1：常规序列转换开始 常规序列转换开始时硬件置位，软件写0清除。
3	STIC	注入序列转换开始标志 0：注入序列转换没有开始 1：注入序列转换开始 注入序列转换开始时硬件置位，软件写 0 清除。
2	EOIC	注入序列转换结束标志 0：注入序列转换没有结束 1：注入序列转换结束 注入序列所有的通道转换结束时硬件置位，软件写0清除。
1	EORC	常规序列转换结束标志 0：常规序列转换没有结束 1：常规序列转换结束 常规序列转换结束时硬件置位，软件写0或读ADC_RDATA寄存器清除。

0	WD0E	模拟看门狗0事件标志 0: 没有模拟看门狗0事件 1: 产生模拟看门狗0事件 转换电压超过ADC_WD0LT和ADC_WD0HT寄存器设置的阈值时由硬件置1。 软件写 0 清除。

21.7.2. 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								RWD0EN	IWD0EN	保留		SYNCM[3:0]			
								rw	rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISNUM[2:0]			DISIC	DISRC	ICA	WD0SC	SM	EOICIE	WD0EIE	EORCIE	WD0CHSEL[4:0]				
rw			rw	rw	rw	rw	rw	rw	rw	rw	rw				

位/位域	名称	说明
31:24	保留	必须保持复位值。
23	RWD0EN	常规序列模拟看门狗 0 使能 0: 常规序列模拟看门狗 0 禁能 1: 常规序列模拟看门狗 0 使能
22	IWD0EN	注入序列模拟看门狗 0 使能 0: 注入序列模拟看门狗 0 禁能 1: 注入序列模拟看门狗 0 使能
21:20	保留	必须保持复位值
19:16	SYNCM[3: 0]	同步模式选择 这些位用于运行模式选择 0000: 独立模式 0001: 常规并行和注入并行组合模式 0010: 常规并行和注入交替触发组合模式 0011: 注入并行和常规快速交叉组合模式 0100: 注入并行和常规慢速交叉组合模式 0101: 注入并行模式 0110: 常规并行模式 0111: 常规快速交叉模式 1000: 常规慢速交叉模式

		1001: 注入交替触发模式
		注意: 1) 这些位只用于 ADC0; 2) 建议用户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	注入序列间断模式 0: 注入序列间断运行模式禁能 1: 注入序列间断运行模式使能
11	DISRC	常规序列间断模式 0: 常规序列间断运行模式禁能 1: 常规序列间断运行模式使能
10	ICA	注入序列自动转换 0: 注入序列自动转换禁能 1: 注入序列自动转换使能
9	WD0SC	扫描模式下, 模拟看门狗 0 在通道配置 0: 模拟看门狗 0 在所有通道有效 1: 模拟看门狗 0 在单通道有效
8	SM	扫描模式 0: 扫描运行模式禁能 1: 扫描运行模式使能
7	EOICIE	EOIC 中断使能 0: EOIC 中断禁能 1: EOIC 中断使能
6	WD0EIE	WD0E 中断使能 0: WD0E 中断禁能 1: WD0E 中断使能
5	EORCIE	EORC 中断使能 0: EORC 中断禁能 1: EORC 中断使能
4:0	WD0CHSEL[4:0]	模拟看门狗 0 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} 。

21.7.3. 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							TSVEN	INREFEN	SWRCST	SWICST	Reserved	ETMRC[1:0]		保留	
							r/w	r/w	r/w	r/w		r/w			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	ETMIC[1:0]		保留	DAL	保留		RDMA	IDMA	保留					CTN	ADCON
	r/w			r/w			r/w	r/w						r/w	r/w

位/位域	名称	说明
31:24	保留	必须保持复位值。
24	TSVEN	ADC0 的通道 16（温度传感器）使能 0: ADC0 的通道 16 禁能 1: ADC0 的通道 16 使能
23	INREFEN	ADC0 的通道 17（内部参考电压 V_{REFINT} ）使能 0: ADC0 的通道 17 禁能 1: ADC0 的通道 17 使能
22	SWRCST	常规序列软件启动转换 如果 ETSRC 是 111，该位置‘1’开启常规序列转换。软件置位，软件清零，或转换开始后，由硬件清零。
21	SWICST	注入序列软件启动转换

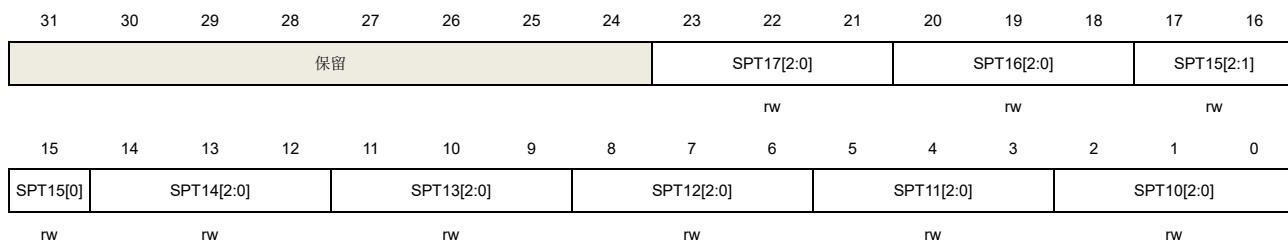
		如果 ETSIC 是 111，该位置‘1’开启注入序列转换。软件置位，软件清零，或转换开始后，由硬件清零。
20	保留	必须保持复位值。
19: 18	ETMRC[1:0]	常规序列外部触发模式 00, 01, 10: 常规序列外部触发上升沿使能 11: 常规序列外部触发禁能
17: 15	保留	必须保持复位值。
14: 13	ETMIC[1:0]	注入序列外部触发模式 00, 01, 10: 注入序列外部触发上升沿使能 11: 注入序列外部触发禁能
12	保留	必须保持复位值。
11	DAL	数据对齐 0: 最低有效位对齐 1: 最高有效位对齐
10: 9	保留	必须保持复位值。
8	RDMA	常规序列的DMA请求使能 0: DMA请求禁止 1: DMA 请求使能
7	IDMA	注入序列的DMA请求使能 0: DMA请求禁止 1: DMA 请求使能
6: 2	保留	必须保持复位值。
1	CTN	连续模式 0: 禁能连续运行模式 1: 使能连续运行模式
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 关闭电源 1: 使能 ADC

21.7.4. 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



位/位域	名称	说明
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: 通道采样时间为1.5周期 001: 通道采样时间为7.5周期 010: 通道采样时间为13.5周期 011: 通道采样时间为28.5周期 100: 通道采样时间为41.5周期 101: 通道采样时间为55.5周期 110: 通道采样时间为71.5周期 111: 通道采样时间为 239.5 周期

21.7.5. 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



SPT5[0]	SPT4[2:0]	SPT3[2:0]	SPT2[2:0]	SPT1[2:0]	SPT0[2:0]
rw	rw	rw	rw	rw	rw

位/位域	名称	说明
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: 通道采样时间为1.5周期 001: 通道采样时间为7.5周期 010: 通道采样时间为13.5周期 011: 通道采样时间为28.5周期 100: 通道采样时间为41.5周期 101: 通道采样时间为55.5周期 110: 通道采样时间为71.5周期 111: 通道采样时间为 239.5 周期

21.7.6. 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															

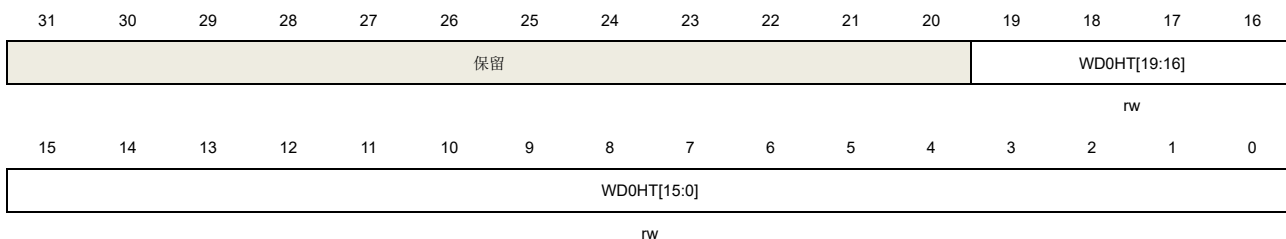
位/位域	名称	说明
31:12	保留	必须保持复位值。
11:0	IOFF[11:0]	注入通道 x 的数据偏移 当转换注入通道时，这些位定义了用于从原始转换数据中减去的数值。转换的结果可以在 ADC_IDATA 寄存器中读出。

21.7.7. 看门狗高阈值寄存器（ADC_WD0HT）

地址偏移：0x24

复位值：0x000F FFFF

该寄存器只能按字（32 位）访问



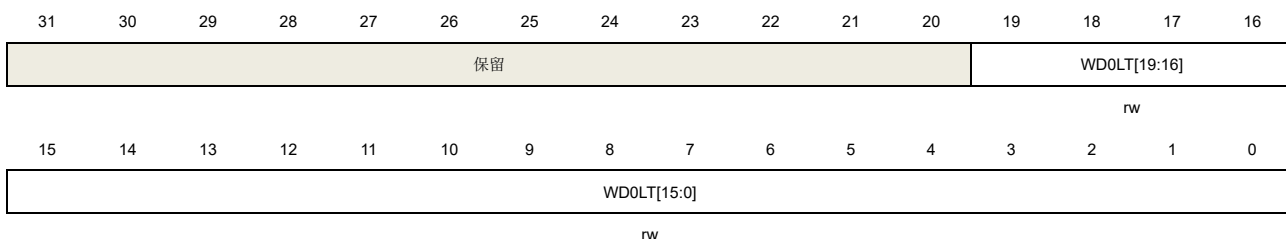
位/位域	名称	说明
31:20	保留	必须保持复位值。
19:0	WD0HT[19:0]	模拟看门狗 0 高侧阈值 这些位定义了模拟看门狗 0 的高阈值。

21.7.8. 看门狗低阈值寄存器（ADC_WD0LT）

地址偏移：0x28

复位值：0x000F FFFF

该寄存器只能按字（32 位）访问



位/位域	名称	说明
31:20	保留	必须保持复位值。

19:0 WD0LT[19:0] 模拟看门狗 0 低侧阈值
这些位定义了模拟看门狗 0 的低阈值。

21.7.9. 常规序列寄存器 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



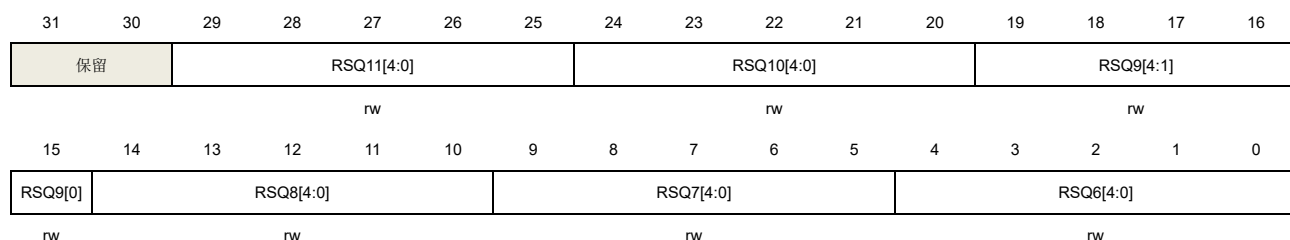
位/位域	名称	说明
31:24	保留	必须保持复位值。
23:20	RL[3:0]	常规序列长度 常规转换序列中的总的通道数目为 RL[3:0]+1。
19:15	RSQ15[4:0]	参考 RSQ0[4:0]的描述
14:10	RSQ14[4:0]	参考 RSQ0[4:0]的描述
9:5	RSQ13[4:0]	参考 RSQ0[4:0]的描述
4:0	RSQ12[4:0]	参考 RSQ0[4:0]的描述

21.7.10. 常规序列寄存器 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



位/位域	名称	说明

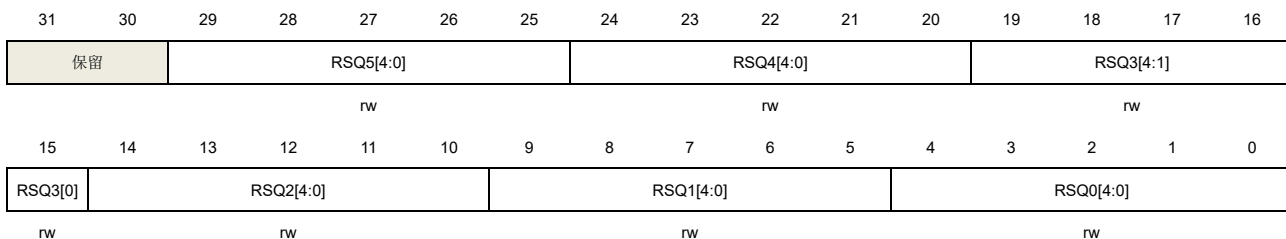
31:30	保留	必须保持复位值。
29:25	RSQ11[4:0]	参考 RSQ0[4:0]的描述
24:20	RSQ10[4:0]	参考 RSQ0[4:0]的描述
19:15	RSQ9[4:0]	参考 RSQ0[4:0]的描述
14:10	RSQ8[4:0]	参考 RSQ0[4:0]的描述
9:5	RSQ7[4:0]	参考 RSQ0[4:0]的描述
4:0	RSQ6[4:0]	参考 RSQ0[4:0]的描述

21.7.11. 常规序列寄存器 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



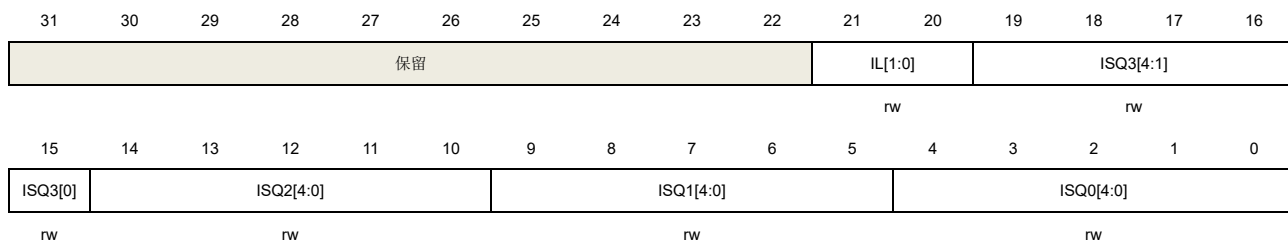
位/位域	名称	说明
31:30	保留	必须保持复位值。
29:25	RSQ5[4:0]	参考 RSQ0[4:0]的描述
24:20	RSQ4[4:0]	参考 RSQ0[4:0]的描述
19:15	RSQ3[4:0]	参考 RSQ0[4:0]的描述
14:10	RSQ2[4:0]	参考 RSQ0[4:0]的描述
9:5	RSQ1[4:0]	参考 RSQ0[4:0]的描述
4:0	RSQ0[4:0]	通道编号 (0..17) 写入这些位来选择常规序列的第 n 个转换的通道

21.7.12. 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



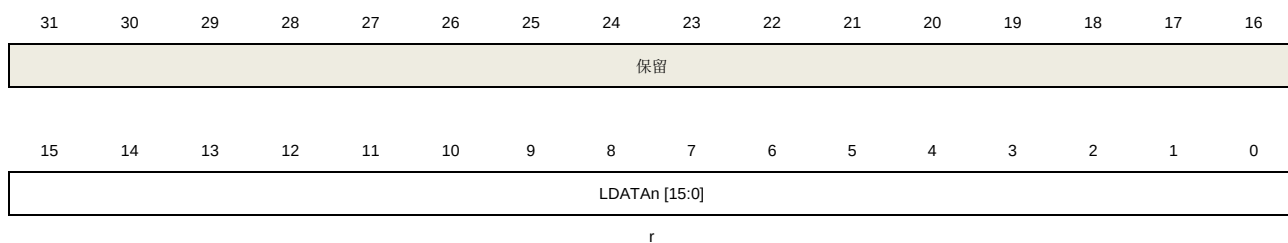
位/位域	名称	说明
31:22	保留	必须保持复位值。
21:20	IL[1:0]	注入序列长度 注入序列总的通道数目为 IL[1:0]+1。
19:15	ISQ3[4:0]	参考 ISQ0[4:0]的描述
14:10	ISQ2[4:0]	参考 ISQ0[4:0]的描述
9:5	ISQ1[4:0]	参考 ISQ0[4:0]的描述
4:0	ISQ0[4:0]	通道编号（0..17）写入这些位来选择注入序列的第 n 个转换的通道 和常规序列不同的是，如果 IL[1:0]长度不足 4，注入通道转换从（4-IL[1:0]-1）开始。 IL 注入通道转换顺序 3 ISQ0 >> ISQ1 >> ISQ2 >> ISQ3 2 ISQ1 >> ISQ2 >> ISQ3 1 ISQ2 >> ISQ3 0 ISQ3

21.7.13. 锁存数据寄存器 x（ADC_LDATABx）（x= 0..3）

地址偏移：0x3C + 0x04 * x (x=0..3)

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述

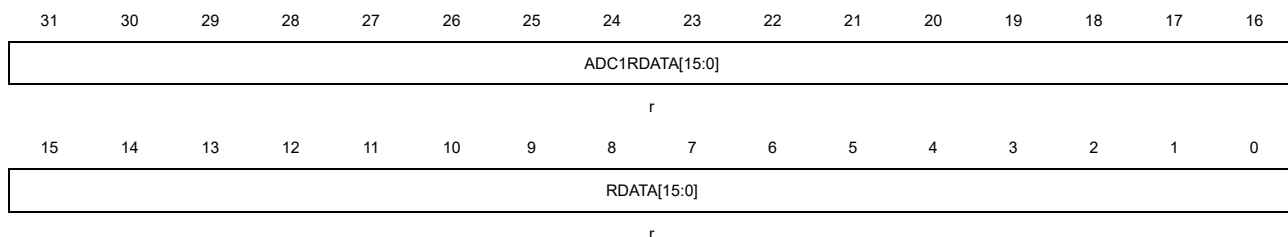
31:16	保留	必须保持复位值。
15:0	LDATA _n [15:0]	注入或者常规序列第 <i>n</i> 个转换数据 这些位包含了注入或者常规序列第 <i>n</i> 个转换数据，只读。

21.7.14. 常规数据寄存器（ADC_RDATA）

地址偏移：0x4C

复位值：0x0000 0000

该寄存器只能按字（32 位）访问



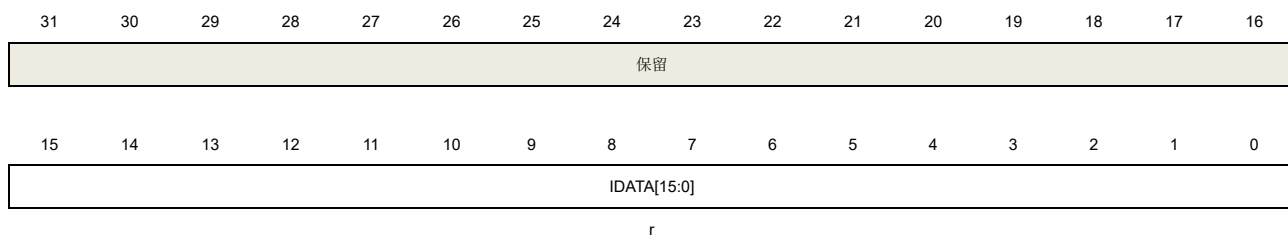
位/位域	名称	说明
31:16	ADC1RDTR[15:0]	ADC1常规通道数据 在ADC0中：在同步模式下，这些位包含着ADC1的常规通道数据。 在 ADC1 中：这些位保留。
15:0	RDATA[15:0]	常规通道数据 这些位包含了常规通道的转换结果，只读。

21.7.15. 注入数据寄存器（ADC_IDATA）

地址偏移：0x50

复位值：0x0000 0000

该寄存器只能按字（32 位）访问



位/位域	名称	说明
31:16	保留	必须保持复位值。
15:0	IDATA[15:0]	注入通道转换的数据

这些位包含了注入通道的转换结果，只读。

21.7.16. 锁存数据控制寄存器（ADC_LDCTL）

地址偏移：0x54

复位值：0x0001 0203

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
SEQSEL 0	保留			COVSEL0[3:0]				SEQSEL 1	保留			COVSEL1[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEQSEL 2	保留			COVSEL2[3:0]				SEQSEL 3	保留			COVSEL3[3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31	SEQSEL0	ADC_LDATAL0寄存器序列源选择 0：选择注入序列 1：选择常规序列 注意： 只有在ADC禁能（ADCON=0）时，才能软件写这些位。
30:28	保留	必须保持复位值。
27:24	COVSEL0[3:0]	ADC_LDATAL0寄存器转换源选择 当ADC_LDATAL0寄存器用于序列转换时，可以选择将注入序列（SEQSEL0 = 0）或者常规序列（SEQSEL0 = 1）的第n个转换的数据存储在ADC_LDATAL0寄存器中。 0000：序列第0个转换 0001：序列第1个转换 0010：序列第2个转换 ... 1111：序列第15个转换 其他值保留。 注意： 在注入序列（SEQSEL0 = 0）中，只能为4'b0000、4'b0001、4'b0010和4'b0011，默认值为4'b0000。只有在ADC禁能（ADCON=0）时，才能软件写这些位。
23	SEQSEL1	参考SEQSEL0描述。
22:20	保留	必须保持复位值。
19:16	COVSEL1[3:0]	参考COVSEL0[3:0]描述。
15	SEQSEL2	参考SEQSEL0描述。

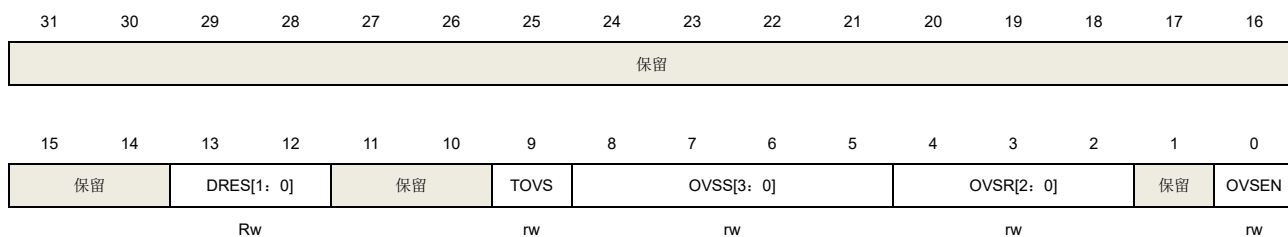
14:12	保留	必须保持复位值。
11:8	COVSEL2[3:0]	参考COVSEL0[3:0]描述。
7	SEQSEL3	参考SEQSEL0描述。
6:4	保留	必须保持复位值。
3:0	COVSEL3[3:0]	参考COVSEL0[3:0]描述。

21.7.17. 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



位/位域	名称	说明
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	过采样触发 该位通过软件置位和清除。 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： 只有在 ADCON=0 的时候才允许通过软件对该位进行写操作（确保没有转换正在执行）。
8:5	OVSS[3:0]	过采样移位 该位通过软件置位和清除。 0000: 不移位

		0001: 移 1 位
		0010: 移 2 位
		0011: 移 3 位
		0100: 移 4 位
		0101: 移 5 位
		0110: 移 6 位
		0111: 移 7 位
		1000: 移 8 位
		其余值都保留
		注意: 只有在 ADCON=0 的时候才允许通过软件对该位进行写操作（确保没有转换正在执行）。
4:2	OVS[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: 只有在 ADCON=0 的时候才允许通过软件对该位进行写操作（确保没有转换正在执行）。
1	保留	必须保持复位值。
0	OVSEN	过采样使能
		该位通过软件置位和清除。
		0: 过采样禁能
		1: 过采样使能
		注意: 只有在 ADCON=0 的时候才允许通过软件对该位进行写（确保没有转换正在执行）。

22.1. 简介

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。

22.2. 主要特征

- 8 位或 12 位分辨率；
- 数据左对齐或右对齐；
- DMA 功能与欠载检测；
- 同步更新转换；
- 外部事件触发转换；
- 可配置的内部缓冲区；
- 输入参考电压 V_{REFP} ；
- 噪声波生成（LSFR 噪声模式和三角噪声模式）；

图 22-1. DAC 结构框图

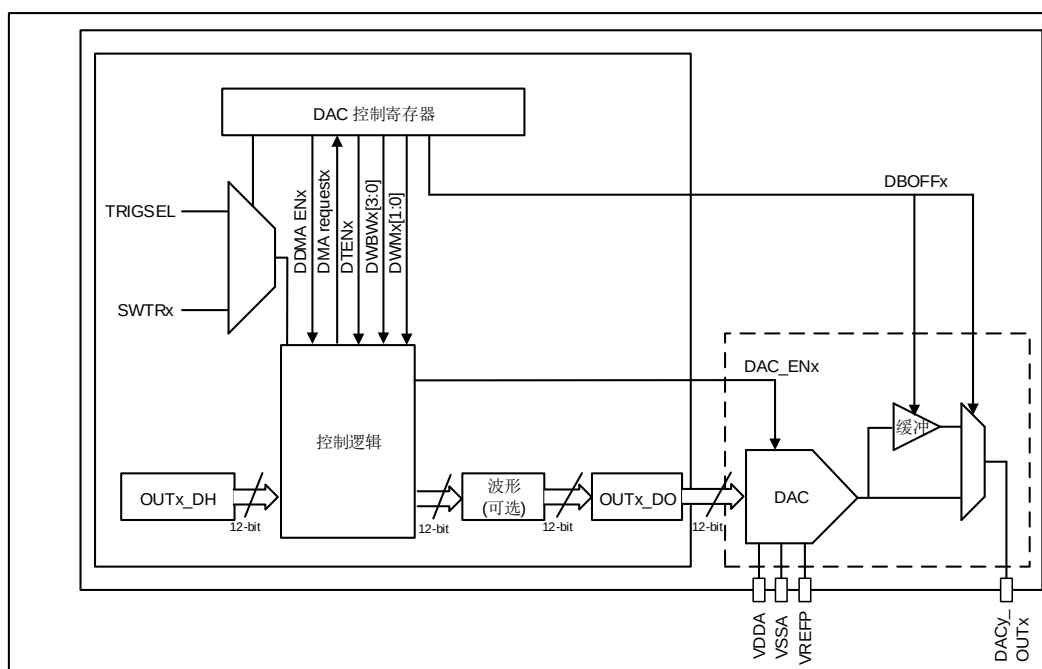


表 22-1. DAC 引脚

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 _{REFP}	DAC 正参考电压	输入，模拟正参考电压
DACy_OUTx	DAC 模拟输出	模拟输出信号

下表详细列出了 DAC 的触发与输出信号。

表 22-2. DAC 触发与输出

	DAC0
通道	通道 0
DAC 输出 I/O	PA4 / PA5
DAC 输出 BUFFER 功能	•
软件触发功能	•
TRIGSEL 触发信号	•

注意：在使能 DAC 模块前，GPIO 口（DAC 输出 I/O）应配置为模拟模式。

22.3. 功能描述

22.3.1. DAC 使能

将 DAC_CTL0 寄存器中的 DENx 位置 1，可以给 DAC 模块上电，DAC 子模块完全启动需要等待 t_{WAKEUP} 时间。

22.3.2. 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 模块内部各集成了一个输出缓冲区。

默认情况下，输出缓冲区是开启的，可以通过设置 DAC_CTL0 寄存器的 DBOFFx 位来开启或关闭缓冲区。

22.3.3. DAC 数据配置

对于 12 位的 DAC 保持数据 (OUTx_DH)，可以通过对 DAC_OUTx_R12DH、DAC_OUTx_L12DH 和 DAC_OUTx_R8DH 中的任意一个寄存器写入数据来配置。当数据被加载到 DAC_OUTx_R8DH 寄存器时，只有 8 位最高有效位是可配置，4 位最低有效位被强制置为 4'b0000。

22.3.4. DAC 触发

DAC 可以通过软件或者外部信号的上升沿触发。外部触发可以通过设置 DAC_CTL0 寄存器中 DTENx 位来使能。触发源可以通过 DAC_CTL0 寄存器中 DTSELx 位来进行选择，如[表 22-3. DAC 外部触发](#)所示。

表 22-3. DAC 外部触发

DTSELx[1:0]	触发源	触发类型
2b'00	TRIGSEL	硬件触发
2b'01 / 2b'10	保留	保留
2b'11	SWTR	软件触发

外部触发器由 TRIGSEL 生成，软件触发是通过设置 DAC_SWT 寄存器的 SWTRx 位生成的。

注意：只能在 dac 模块使能之后（DEN0=1），再去触发 DAC 模块。

22.3.5. DAC 转换

如果使能了外部触发（通过设置 DAC_CTL0 寄存器的 DTENx 位），当已经选择的触发事件发生，DAC 保持数据（OUTx_DH）会被转移到 DAC 数据输出寄存器（DAC_OUTx_DO）。而在外部触发未使能的情况下，DAC 保持数据（OUTx_DH）会被自动转移到 DAC 数据输出寄存器（DAC_OUTx_DO）。

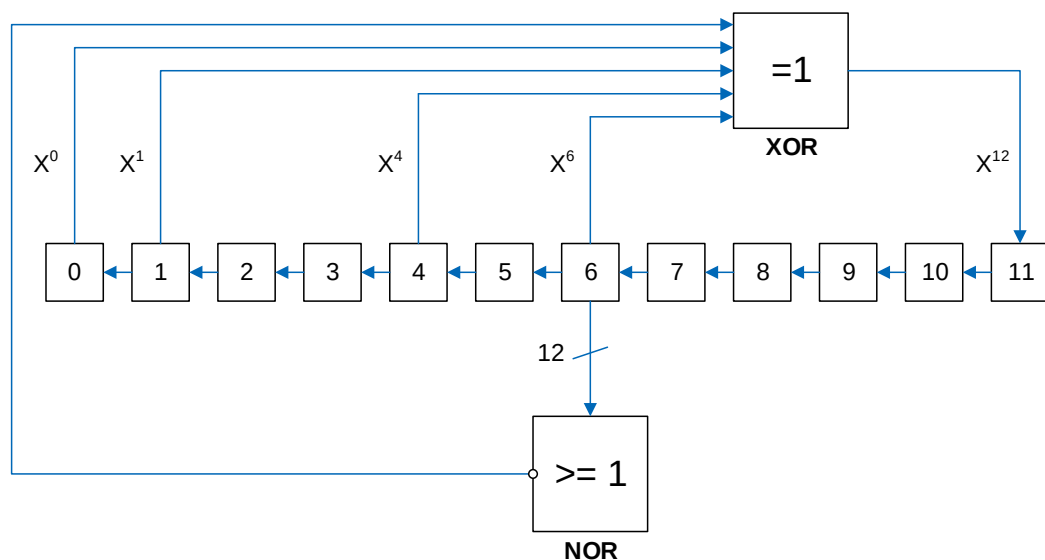
当 DAC 保持数据（OUTx_DH）加载到 DAC_OUTx_DO 寄存器时，经过 tSETTLING 时间之后，模拟输出变得有效，tSETTLING 的值与电源电压和模拟输出负载有关。

22.3.6. 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_CTL0 寄存器的 DWMx 位来进行选择。噪声的幅值可以通过配置 DAC_CTL0 寄存器的 DAC 噪声波位宽（DWBWx）位来进行设置。

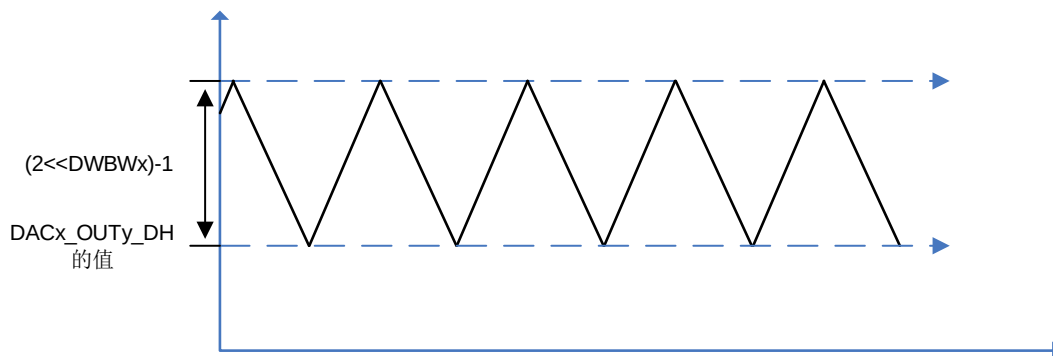
LFSR 噪声模式：在 DAC 控制逻辑中有一个线性反馈移位寄存器（LFSR）。在此模式下，LFSR 的值与 OUTx_DH 值相加后，被写入到 DAC 数据输出寄存器（DAC_OUTx_DO）。当配置的 DAC 噪声波位宽小于 12 时，LFSR 的值等于 LFSR 寄存器最低的 DWBWx 位，高位被屏蔽。

图 22-2. DAC LFSR 算法



三角噪声模式：三角波幅值与 $OUTx\_DH$ 值相加后，被写入到 DAC 数据输出寄存器 (DAC_OUTx_DO)。三角波幅值的最小值为 0，最大值为 $(2^{DWBWx} - 1)$ 。

图 22-3. DAC 三角噪声模式生成的波形



22.3.7. DAC 输出电压

DAC 引脚上的模拟输出电压取决于下面的等式：

$$V_{DAC\_OUT} = V_{REFP} * OUTx\_DO / 4096 \quad (22-1)$$

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_{REFP} 。

22.3.8. DMA 请求

在外部触发使能的情况下，通过设置 DAC_CTL0 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），则产生一个 DMA 请求。

22.4. DAC 寄存器

DAC0 基地址：0x4000 7400

22.4.1. DACx 控制寄存器 (DAC_CTL0)

地址偏移：0x00

复位值：0x0000 8000

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
OUTSEL	DCSEL0	DDUDR	DDMA	DWBW0[3:0]				DWM0[1:0]		保留	DTSEL0[1:0]		DTEN0	DBOFF0	DEN0
0		IE0	EN0												
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	OUTSEL0	DACx_OUT0 DAC 输出引脚选择 0: DACx_OUT0 连接到 PA5 1: DACx_OUT0 连接到 PA4（默认值是 1）
14	DCSEL0	DACx_OUT0 与 CMP 相连时，DACx_OUT0 与 GPIO 的连接选择 0: DACx_OUT0与GPIO连接 1: DACx_OUT0不与GPIO连接
13	DDUDRIE0	DACx_OUT0 DMA欠载中断使能 0: DACx_OUT0 DMA欠载中断禁能 1: DACx_OUT0 DMA欠载中断使能
12	DDMAEN0	DACx_OUT0 DMA 使能 0: DACx_OUT0 DMA 模式禁能 1: DACx_OUT0 DMA模式使能
11:8	DWBW0[3:0]	DACx_OUT0 噪声波位宽 这些位指定了 DACx_OUT0 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

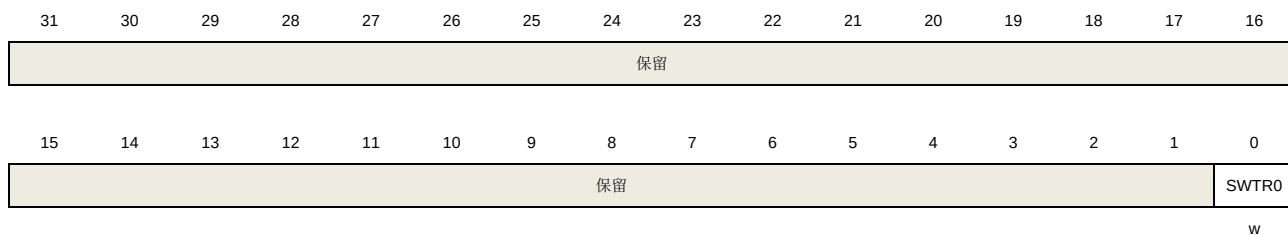
		0010: 波形信号的位宽为 3
		0011: 波形信号的位宽为 4
		0100: 波形信号的位宽为 5
		0101: 波形信号的位宽为 6
		0110: 波形信号的位宽为 7
		0111: 波形信号的位宽为 8
		1000: 波形信号的位宽为 9
		1001: 波形信号的位宽为 10
		1010: 波形信号的位宽为 11
		≥1011: 波形信号的位宽为12
		注意: 当DAC使能 (DEN0=1) 时, 该位域不能写入。
7:6	DWM0[1:0]	DACx_OUT0 噪声波模式 这些位指定了在 DACx_OUT0 外部触发使能(DTEN0=1)的情况下, DACx_OUT0 的噪声波模式的选择。 00: 波形生成禁能 01: LFSR 噪声模式 1x: 三角噪声模式
5	保留	必须保持复位值
4:3	DTSEL0[1:0]	DACx_OUT0 触发选择 这些位仅在 DTEN0=1 并选择用于触发 DAC 的外部事件时使用。 00: TRIGSEL 11: 软件触发 其他所有值: 保留
2	DTEN0	DACx_OUT0 触发使能 0: DACx_OUT0 触发禁能 1: DACx_OUT0触发使能
1	DBOFF0	DACx_OUT0输出缓冲区关闭 0: DACx_OUT0输出缓冲区打开, 以降低输出阻抗, 提高驱动能力 1: DACx_OUT0输出缓冲区关闭
0	DEN0	DACx_OUT0 使能 0: DACx_OUT0 禁能 1: DACx_OUT0使能

22.4.2. DACx 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。



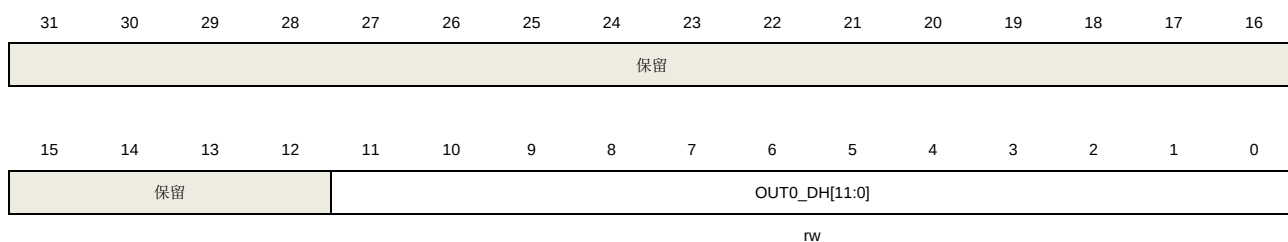
位/位域	名称	描述
31:1	保留	必须保持复位值
0	SWTR0	DACx_OUT0 软件触发，由硬件清除。 0：软件触发禁能 1：软件触发使能

22.4.3. DACx_OUT0 12 位右对齐数据保持寄存器 (DAC_OUT0_R12DH)

地址偏移：0x08

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



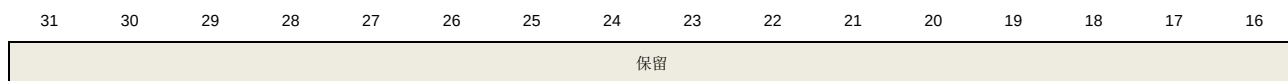
位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	OUT0_DH[11:0]	DACx_OUT0 12 位右对齐数据 这些位指定了将由 DACx_OUT0 转换的数据。

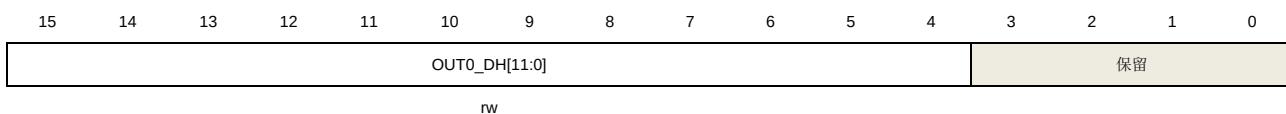
22.4.4. DACx_OUT0 12 位左对齐数据保持寄存器 (DAC_OUT0_L12DH)

地址偏移：0x0C

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。





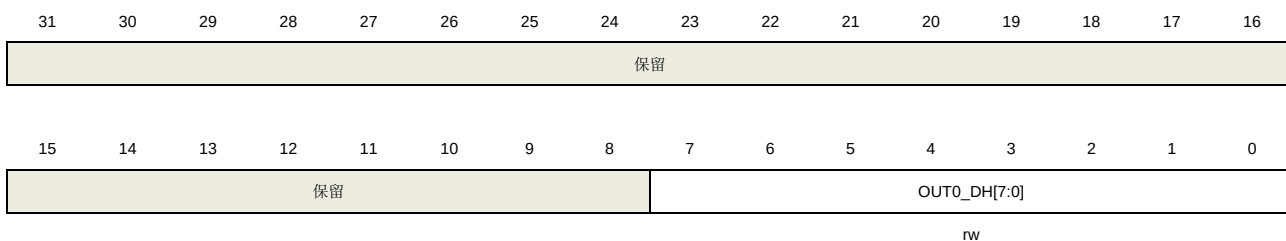
位/位域	名称	描述
31:16	保留	必须保持复位值
15:4	OUT0_DH[11:0]	DACx_OUT0 12 位左对齐数据 这些位指定了将由 DACx_OUT0 转换的数据。
3:0	保留	必须保持复位值

22.4.5. DACx_OUT0 8 位右对齐数据保持寄存器 (DAC_OUT0_R8DH)

地址偏移：0x10

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



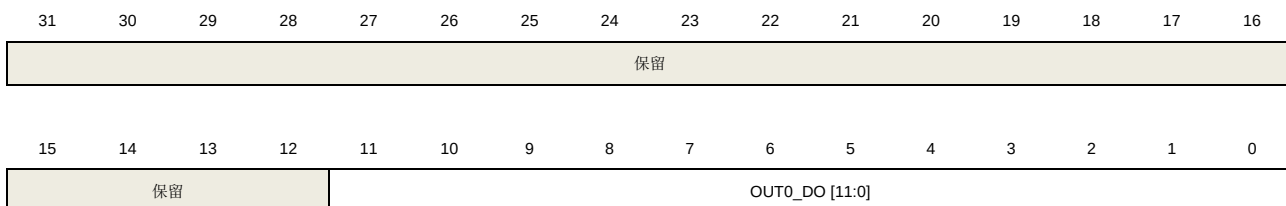
位/位域	名称	描述
31:8	保留	必须保持复位值
7:0	OUT0_DH[7:0]	DACx_OUT0 8 位右对齐数据 这些位指定了将由 DACx_OUT0 转换的数据的最高 8 位有效位。

22.4.6. DACx_OUT0 数据输出寄存器 (DAC_OUT0_DO)

地址偏移：0x2C

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



r

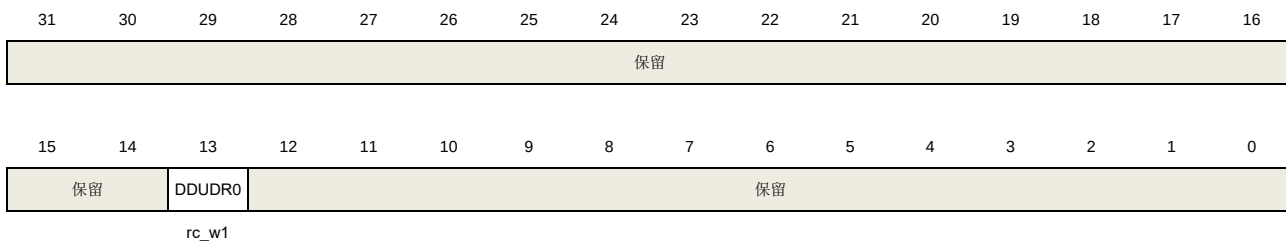
位/位域	名称	描述
31:12	保留	必须保持复位值
11:0	OUT0_DO [11:0]	DACx_OUT0 数据输出。 这些位为只读类型，存储由 DACx_OUT0 转换的数据。

22.4.7. DAC 状态寄存器 0 (DAC_STAT0)

地址偏移：0x34

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:14	保留	必须保持复位值
13	DDUDR0	DACx_OUT0 DMA 欠载标志位，硬件置位，软件清零。 0：没有欠载发生 1：发生欠载（DAC 触发产生速度快于 DMA 传输速度）
12:0	保留	必须保持复位值

23. 通用同步异步收发器（USART）

23.1. 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 UCLK（PCLK1 或 PCLK2）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，半双工以及同步模式。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。

23.2. 主要特征

- NRZ标准格式（Mark/Space）。
- 全双工异步通信。
- 半双工单线通信。
- 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。
 - 8倍或16倍过采样。
 - 当时钟频率为280M，过采样为8，最高速度可到35Mbits/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匹配从静默模式唤醒。
- 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

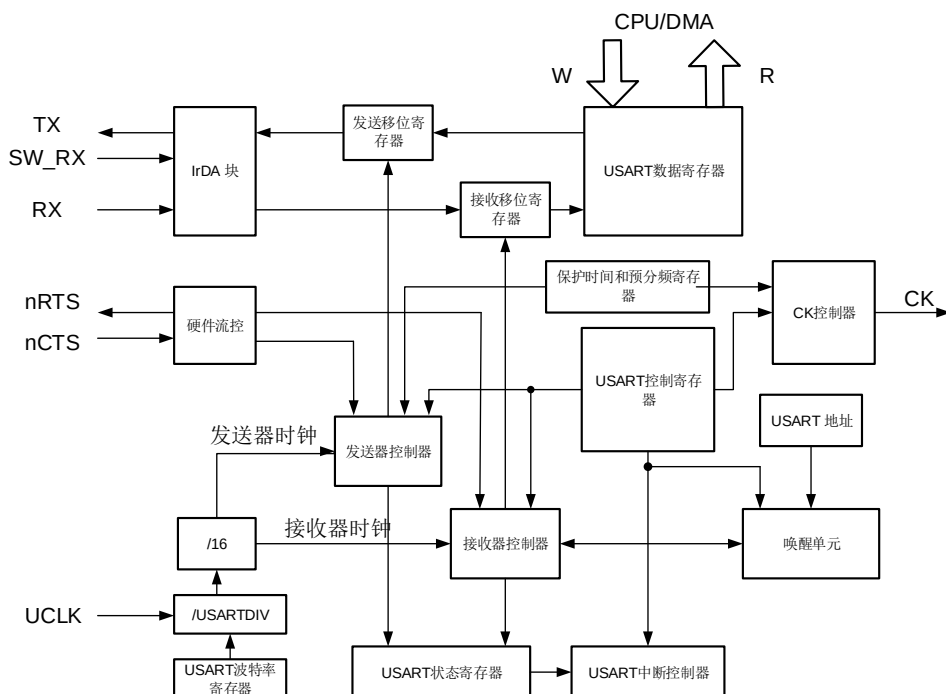
23.3. 功能说明

USART接口通过[表23-1. USART重要引脚描述](#)中主要引脚从外部连接到其他设备。

表 23-1. 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

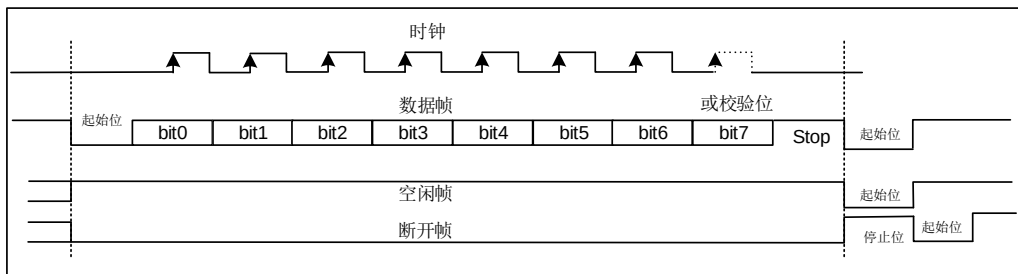
图 23-1. USART 模块内部框图



23.3.1. 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 23-2. 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 23-2. 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

STB[1:0]	停止位长度（位）	功能描述
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 UCLK 时钟频率，波特率发生器的配置共同决定。

23.3.2. 波特率发生

波特率分频系数是一个 16 位的数字，包含 12 位整数部分和 4 位小数部分。波特率发生器使用这两部分组合所得的数值来确定波特率。由于具有小数部分的波特率分频系数，将使 USART 能够产生所有标准波特率。

波特率分频系数（USARTDIV）与系统时钟具有如下关系：

如果过采样率是 16，公式为：

$$\text{USARTDIV} = \frac{\text{UCLK}}{16 \times \text{Baud Rate}} \quad (23-1)$$

如果过采样率是 8，公式为：

$$\text{USARTDIV} = \frac{\text{UCLK}}{8 \times \text{Baud Rate}} \quad (23-2)$$

例如，当过采样是 16：

1. 由 USART_BAUD 寄存器的值得到 USARTDIV：

假设 USART_BAUD=0x21D，则 INTDIV=33（0x21），FRADIV=13（0xD）。

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）。

USART_BAUD=0x1E6。

注意：若取整后 FRADIV=16（溢出），则进位必须加到整数部分。

23.3.3. 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。

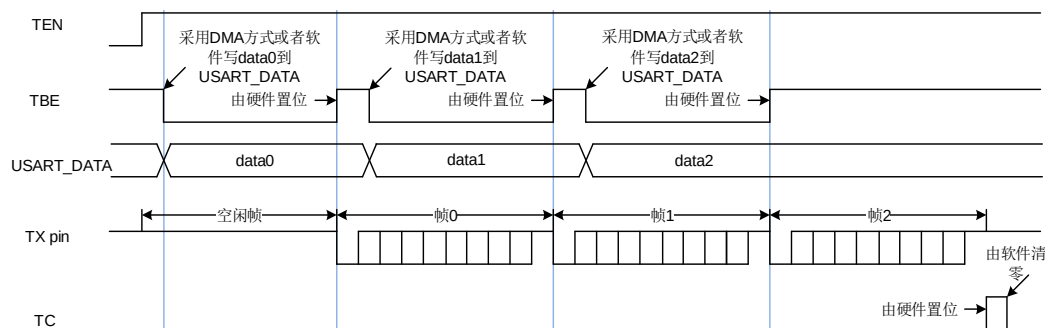
系统上电后，TBE 默认为高电平。在 USART_STAT0 寄存器中 TBE 置位时，数据可以在不覆盖前一个数据的情况下写入 USART_DATA 寄存器。当数据写入 USART_DATA 寄存器，TBE 位将被清 0。在数据由 USART_DATA 移入移位寄存器后，该位由硬件置 1。如果数据在一个发送过程正在进行时被写入 USART_DATA 寄存器，它将首先被存入发送缓冲区，在当前发送过程完成时传输到发送移位寄存器中。如果数据在写入 USART_DATA 寄存器时，没有发送过程正在进行，TBE 位将被清零然后迅速置位，原因是数据将立刻传输到发送移位寄存器。

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 23-3. 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 23-3. 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。

23.3.4. 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。

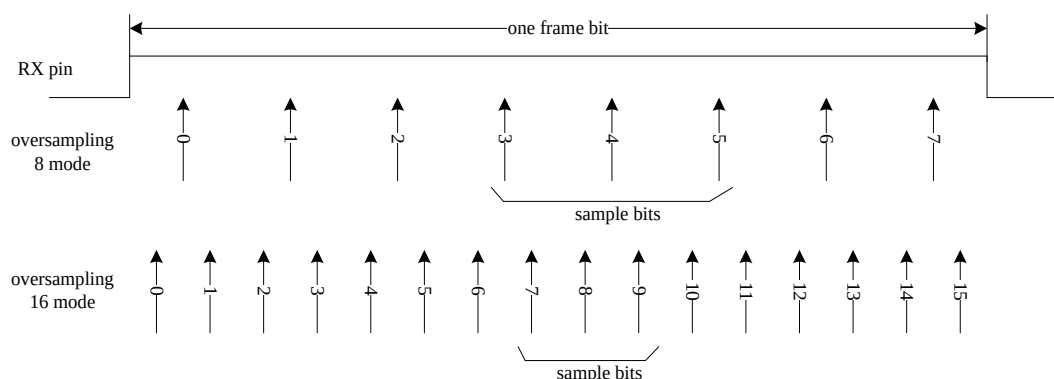
当接收到一个数据帧，USART_STAT0寄存器中的RBNE置位，如果设置了USART_CTL0寄存器中相应的中断使能位RBNEIE，将会产生中断。在USART_STAT0寄存器中可以观察接收状态标志。

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。如果是8倍过采样模式，选择第3、4、5个采样点；如果是16倍过采样模式，选择第7、8、9个采样点。如果在3个采样点中有2个或3个为0，该数据位被视为0，否则为1。如果3个采样点中有一个采样点的值与其他两个不同，不管是起始位，数据位，奇偶校验位或者停止位，都将产生噪声错误（NERR）。如果使能DMA，并置位USART_CTL2寄存器中ERRIE，将会产生中断。如果在USART_CTL2中置位OSB，接收器将仅获取一个采样点来估计一个数据位的值。在这种情况下将不会检测到噪声错误。

图 23-4. 过采样方式接收一个数据位（OSB=0）



通过置位 USART_CTL0 寄存器中的 PCEN 位使能奇偶校验功能，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。

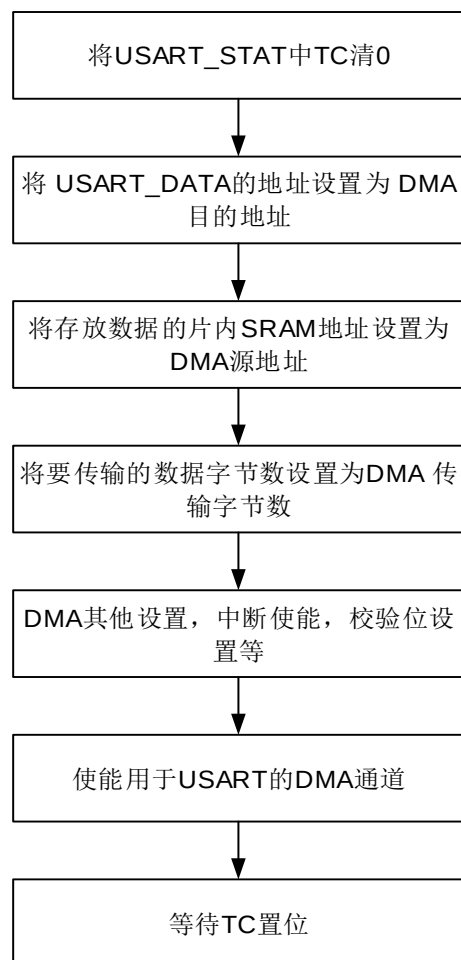
若接收过程中，产生了噪声错误（NERR）、校验错误（PERR）、帧错误（FERR）或溢出错误（ORERR），则 NERR、PERR、FERR 或 ORERR 将和 RBNE 同时置位。如果没有使能 DMA，RBNE 中断发生时，软件需检查是否有噪声错误、校验错误、帧错误或溢出错误产生。

23.3.5. 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 23-5. 采用 DMA 方式实现 USART 数据发送配置步骤](#) 所示。

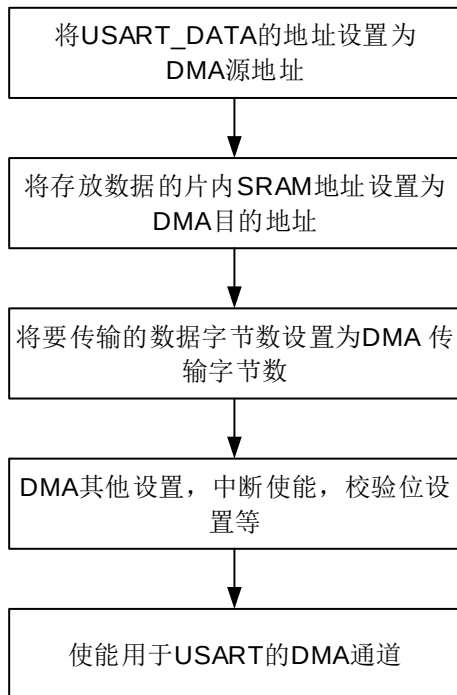
图 23-5. 采用 DMA 方式实现 USART 数据发送配置步骤



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 23-6. 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 23-6. 采用 DMA 方式实现 USART 数据接收配置步骤

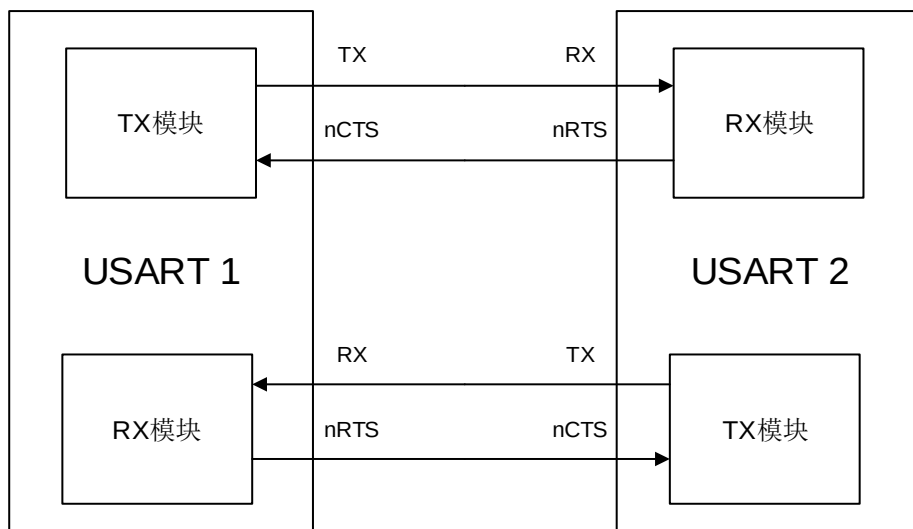


当 USART 接收到的数据数量达到了 DMA 传输数据数量，DMA 模块将产生传输完成中断。

23.3.6. 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 23-7. 两个 USART 之间的硬件流控制



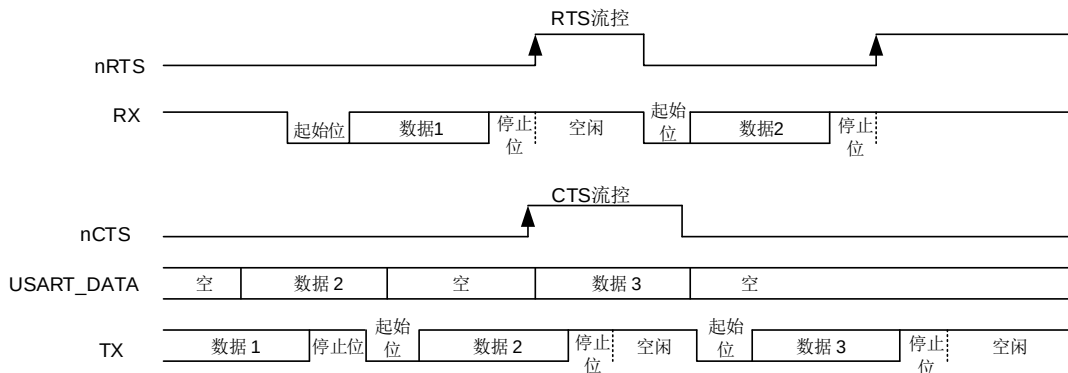
RTS 流控

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 23-8. 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。

23.3.7. 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匹配检测。

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。

如果采用地址匹配检测，默认情况下，IDLEF 不会置位并且接收器对地址字节不做奇偶校验。如果 USART_CHC 寄存器中 PCM 位以及 USART_CTL0 寄存器中 PCEN 位被置位，则 MSB 位将被检查为奇偶校验位，并且 MSB 位之前的位被检测为地址标志。

当 RBNE 为 0 时，RWU 可以被写为 0 或 1。

23.3.8. LIN 模式

将 USART_CTL1 寄存器的 LMEN 置位即可使能本地互连网络模式。

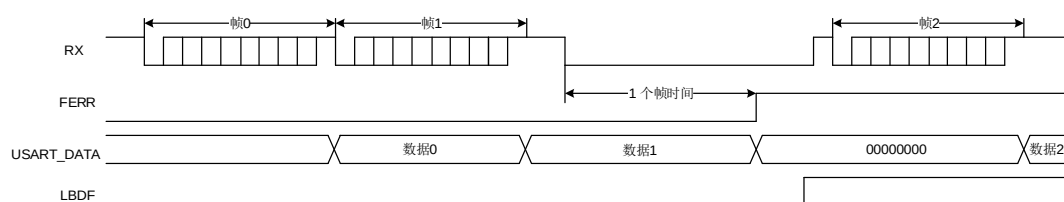
在 LIN 模式下，USART_CTL1 寄存器中 CKEN, WL, STB[1:0]以及 USART_CTL2 的 SCEN, HDEN, IREN 位都应该被清 0。

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。

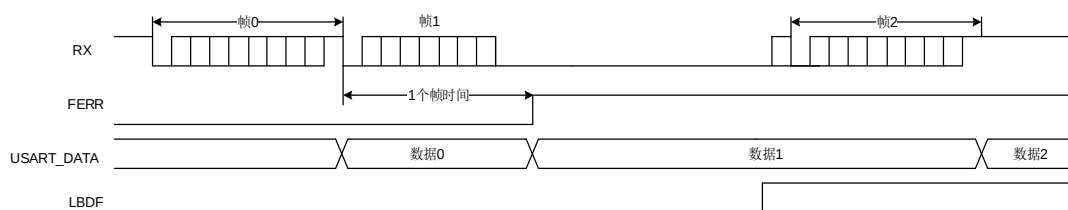
如 [图 23-9. 空闲状态下检测断开帧](#) 所示，如果断开帧发生在空闲状态下，USART 接收器会接收到一个全 0 数据帧，同时 FERR 置位。

图 23-9. 空闲状态下检测断开帧



如 [图 23-10. 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 23-10. 数据传输过程中检测断开帧



23.3.9. 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 23-11. 同步模式下的 USART 示例

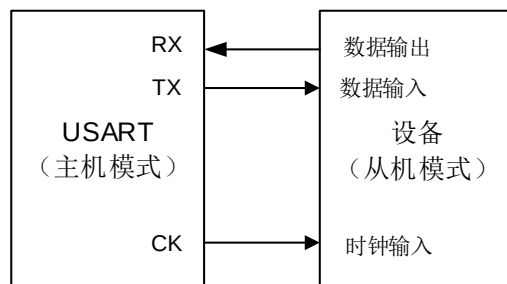
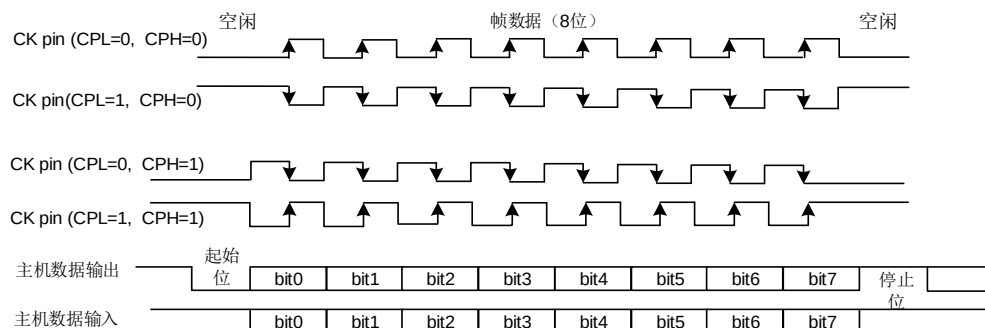


图 23-12. 8-bit 格式的 USART 同步通信波形(CLEN=1)

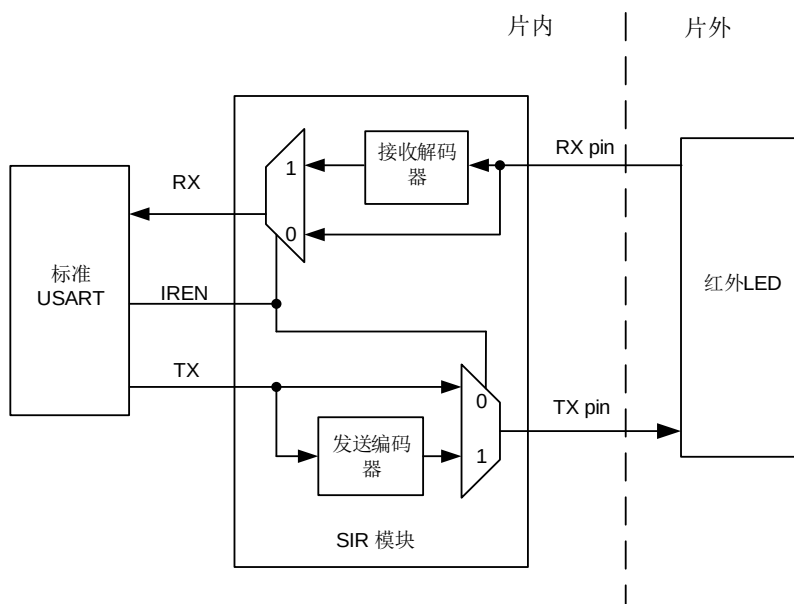


23.3.10. 串行红外(IrDA SIR)编解码功能模块

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。

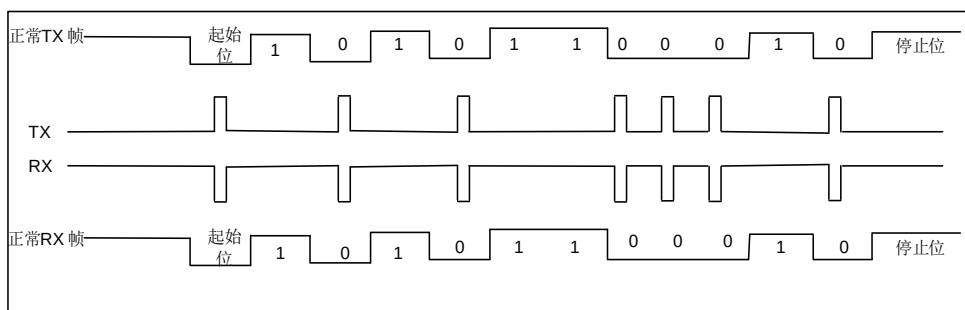
图 23-13. IrDA SIR ENDEC 模块



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 1 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 23-14. 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。

23.3.11. 半双工通信模式

通过设置 USART_CTL2 寄存器的 HDEN 位，可以使能半双工模式。

在半双工通信模式下，USART_CTL1 寄存器的 LMEN, CKEN 位和 USART_CTL2 寄存器的 SCEN, IREN 位清零。

半双工模式下仅用单线通信，TX 引脚和 RX 引脚将从内部连接到一起，RX 引脚不再使用。TX 引脚应被配置为开漏模式，通信冲突由软件处理。

23.3.12. 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 23-15. 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，将会产生中断。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‘1’，偶校验。在这种情况下，USART_CTL3 寄存器中 MSBF 位和 DINV 位都为 1。

23.3.13. USART 中断

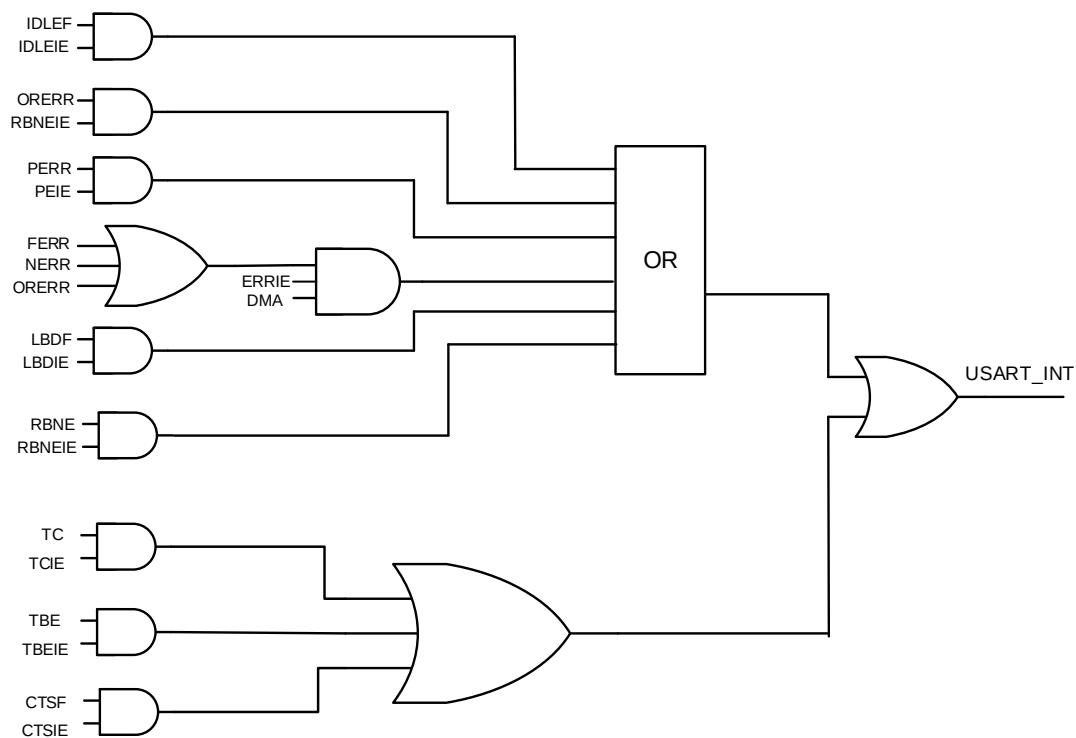
USART 中断事件和标志如 [表 23-3. USART 中断请求](#) 所示：

表 23-3. 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或ORERR或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 23-16. USART 中断映射框图



23.4. 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

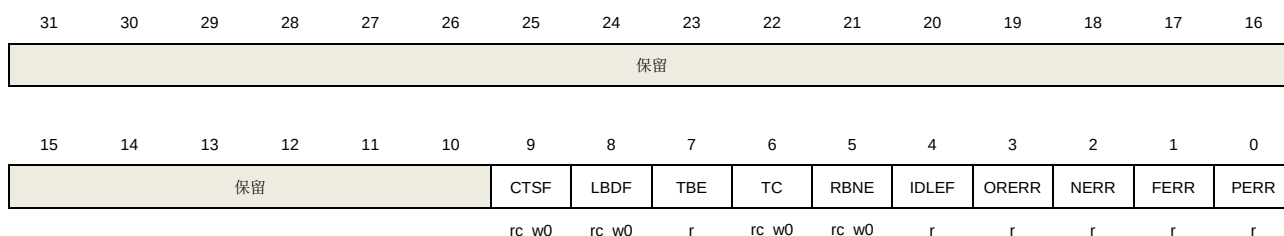
UART4 基地址：0x4000 5000

23.4.1. 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。



位/位域	名称	描述
31:10	保留	必须保持复位值。
9	CTSF	CTS变化标志 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。 该位由软件清0。 0：nCTS状态线没有变化。 1：nCTS状态线发生变化。 该位对UART3/4无效。
8	LBDF	LIN断开检测标志。 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。 该位由软件清0。 0：没有检测到LIN断开字符。 1：检测到LIN断开字符。
7	TBE	发送数据缓冲区空。

		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。 该位在软件将待发送数据写入USART_DATA时被清0。 0：发送数据缓冲区不为空。 1：发送数据缓冲区空。
6	TC	发送完成 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。USART_CTL0寄存器中TCIE被置位将产生中断。 该位由软件清0。 0：发送没有完成 1：发送完成
5	RBNE	读数据缓冲区非空。 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。 软件可以通过对该位写0或读USART_DATA寄存器来将该位清0。 0：读数据缓冲区为空。 1：读数据缓冲区不为空。
4	IDLEF	空闲线检测标志。 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。 软件先读USART_STAT0，再读USART_DATA可清除该位。 0：未检测到空闲帧。 1：检测到空闲帧。
3	ORERR	溢出错误 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_CTL2的ERRIE位被置位，将会有中断产生。 软件先读USART_STAT0，再读USART_DATA可清除该位。 0：没有检测到溢出错误。 1：检测到溢出错误。
2	NERR	噪声错误标志 将USART_CTL2寄存器中OSB清0，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。 软件先读USART_STAT0，再读USART_DATA可清除该位。 0：没检测到噪声错误。 1：检测到噪声错误。
1	FERR	帧错误 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。

软件先读USART_STAT0，再读USART_DATA可清除该位。

0：未检测到帧错误。

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。

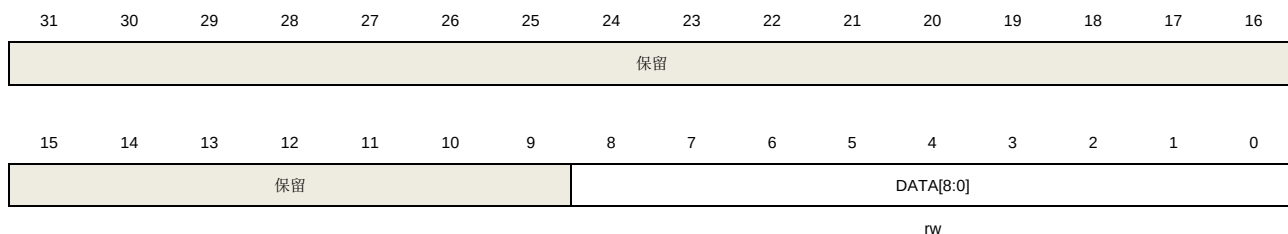
1：检测到校验错误。

23.4.2. 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。

23.4.3. 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。



INTDIV [11:0]	FRADIV[3:0]
rw	rw

位/位域	名称	描述
31:18	保留	必须保持复位值。
17:4	INTDIV[13:0]	波特率分频器的整数部分。 注意： 在IrDA低功耗模式下，INTDIV需要配置为PSC的16倍。
3:0	FRADIV [3:0]	波特率分频器的小数部分。 如果使能了8倍过采样，软件必须保证FRADIV[3]为0。

23.4.4. 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
OVSMOD	保留	UEN	WL	WM	PCEN	PM	PERRIE	TBEIE	TCIE	RBNEIE	IDLEIE	TEN	REN	RWU	SBKCMD
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	OVSMOD	采样模式 0: 16倍采样 1: 8倍采样 如果SCEN=1，IREN=1或者LMEN=1，OVSMOD由硬件强制为0。
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: 地址匹配
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。地址匹配模式下

(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 软件通过发送断开帧将该位置1。 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

23.4.5. 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]			
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	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 地址匹配唤醒模式下(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。

23.4.6. 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				OSB	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
31:12	保留	必须保持复位值。
11	OSB	选择采样方法选择位 该位用于选择采样方法。当该位被置位时，USART对一个数据位取一个采样点，而不是一个数据位取三个采样点。当选择一个采样位方法时，噪声错误标志NERR被禁用。 0：三个采样位法。 1：一个采样位法。
10	CTSIE	CTS中断使能 如果该位置1，当USART_STAT0寄存器中CTSIF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。

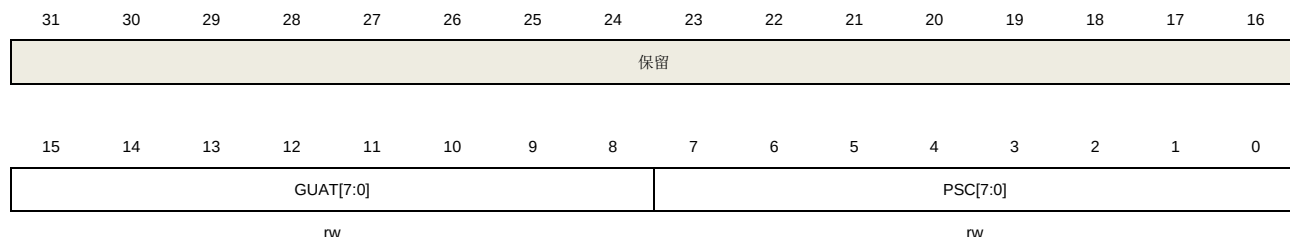
		该位对于UART3/4无效。
4	NKEN	在智能卡模式NACK使能。 该位用于智能卡模式在奇偶校验错误发生时使能NACK发送。 0: 当出现校验错误时不发送NACK。 1: 当出现校验错误时发送NACK。 该位对于UART3/4无效。
3	HDEN	半双工使能 该位用于使能半双工模式。 0: 半双工模式禁用。 1: 半双工模式使能。
2	IRLP	IrDA低功耗模式 该位用于为IrDA模式选择低功耗模式。 0: 正常模式 1: 低功耗模式
1	IREN	IrDA模式使能 0: IrDA禁用 1: IrDA使能
0	ERRIE	错误中断使能 当DMA接收模式（DENR=1）使能时，如果该位被置1，USART_STAT0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。

23.4.7. 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。



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。

23.4.8. 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	高位在前 该位用于设定数据在发送或接收时的顺序。 0：数据发送/接收，采用低位在前。 1：数据发送/接收，采用高位在前。

		USART被使能（UEN=1）时，这一位不能被改写。
10	DINV	数据位反转 该位用于设定在发送或接收时数据位的极性。 0：数据位信号值没有反转。 1：数据位信号值被反转。 USART被使能（UEN=1）时，这一位不能被改写。
9	TINV	TX 引脚电平反转 该位用于设定TX引脚极性。 0：TX引脚信号值没有反转。 1：TX引脚信号值被反转。 USART被使能（UEN=1）时，这一位不能被改写。
8	RINV	RX引脚电平反转。 该位用于设定RX引脚极性。 0：RX引脚信号值没有反转。 1：RX引脚信号值被反转。 USART被使能（UEN=1）时，这一位不能被改写。
7:6	保留	必须保持复位值。
5	EBIE	块结束标志中断使能位。 如果该位置1，USART_STAT1寄存器中EBF被置位时产生中断。 0：块中断禁用。 1：块中断使能。
4	RTIE	接收超时标志中断使能位。 如果该位置1，USART_STAT1寄存器中RTF被置位时产生中断。 0：接收超时中断禁用。 1：接收超时中断使能。
3:1	SCRTNUM[2:0]	智能卡自动重试次数寄存器。 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。 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。 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。

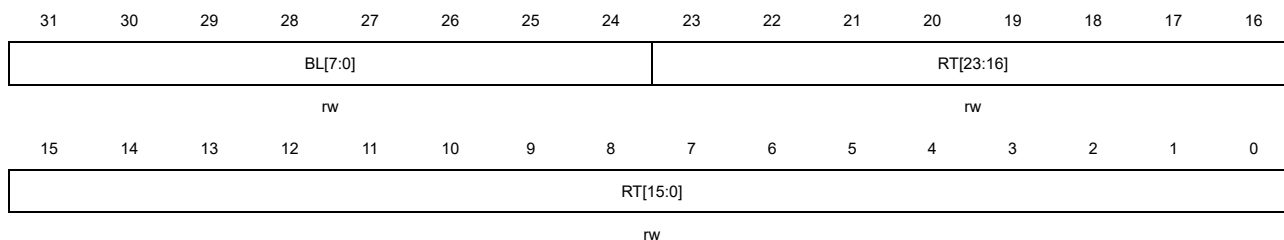
23.4.9. 接收超时寄存器（USART_RT）

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+结束部分的长度（1-LEC/2-CRC）-1。 这个值可以在块接收开始去设置（用于需要从块的序言提取块的长度的情形），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。 注意：在16倍过采样时，接收器超时阈值可配置的最大值为0xFFFFFE。

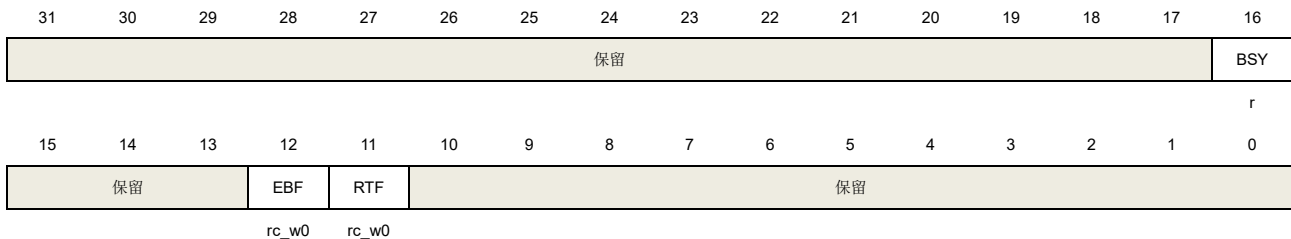
23.4.10. 状态寄存器 1（USART_STAT1）

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。



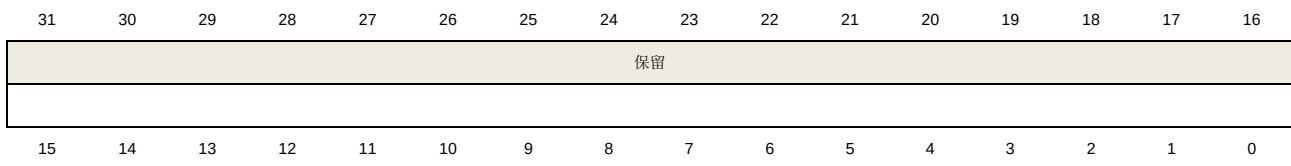
位/位域	名称	描述
31:17	保留	必须保持复位值。
16	BSY	忙标志 USART接收一帧数据时被置位。 0: USART接收通道空闲。 1: USART接收通道忙。
15:13	保留	必须保持复位值。
12	EBF	块结束标志 该位在接收字节数（从块起始开始计数，包含序言）等于或者大于BLEN+4时被置位。 USART_CTL3寄存器中EBIE被置位将产生中断。 软件可以通过写0清除该位。 0: 块结束事件没有发生。 1: 块结束事件发生。
11	RTF	接收超时标志 该位在RX引脚空闲时间已经超过RT值时被置位。USART_CTL3寄存器中RTIE被置位将产生中断。 软件可以通过写0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。

23.4.11. 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。



保留	EPERR	保留	BCM	PCM	HCM
	rc_w0		rw	rw	rw

位/位域	名称	描述
31:9	保留	必须保持复位值。
8	EPERR	校验错误超前检测标志。 该标志将在RBNE置位前，校验位被检测到时置位。 软件写0可以清除该位。 0：没有检测到校验错误 1：检测到校验错误
7:3	保留	必须保持复位值。
2	BCM	断开帧兼容模式 0：当CTS硬件流控制使能时，发送器在发送断开帧之前不会检测nCTS的输入状态 1：当CTS硬件流控制使能时，发送器在发送断开帧之前检测nCTS的输入状态 该位对于UART3/4无效。
1	PCM	校验兼容模式 0：在地址匹配模式唤醒情况下，数据的最高位被用来识别地址，而不是用作校验位。 接收器不会对地址数据进行校验（PERR将不会被置位除非发生校验错误） 1：在地址匹配模式唤醒情况下，接收器对地址进行校验，当检测到校验错误时PERR被置位
0	HCM	硬件流控制兼容模式 0：nRTS信号与USART_STAT0寄存器中RBNE位相同 1：nRTS信号在最后一个数据位被采样后被置位 该位对于UART3/4无效。

24. 串行外设接口/片上音频接口（SPI/I2S）

24.1. 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。SPI0 还支持 SPI 四线主机模式。

片上音频接口（Inter-IC Sound，缩写为 I2S）支持四种音频标准，分别是 I2S 飞利浦标准，MSB 对齐标准，LSB 对齐标准和 PCM 标准。它可以在四种模式下运行，包括主机发送模式，主机接收模式，从机发送模式和从机接收模式。

24.2. 主要特性

24.2.1. SPI 主要特性

- 具有全双工和单工模式的主从操作；
- 16位宽度，独立的发送和接收缓冲区；
- 8位或16位数据帧格式；
- 低位在前或高位在前的数据位顺序；
- 软件和硬件NSS管理；
- 硬件CRC计算、发送和校验；
- 发送和接收支持DMA模式；
- 支持SPI TI模式；
- 支持SPI NSS脉冲模式
- 支持SPI四线功能的主机模式（仅在SPI0中）。

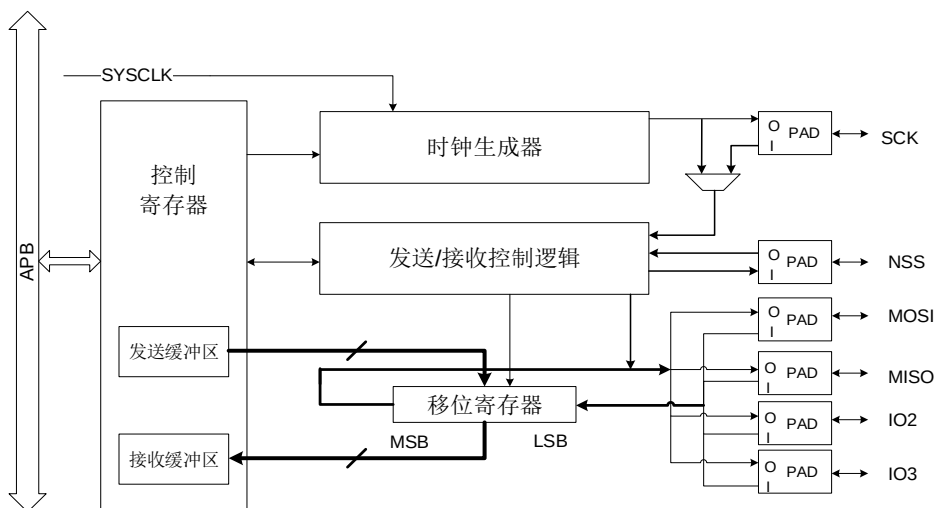
24.2.2. I2S 主要特性

- 具有发送和接收功能的主从操作；
- 支持四种I2S音频标准：飞利浦标准，MSB对齐标准，LSB对齐标准和PCM标准；
- 数据长度可以为16位，24位和32位；
- 通道长度为16位或32位；
- 16位缓冲区用于发送和接收；
- 通过I2S时钟分频器，可以得到8 kHz到192 kHz的音频采样频率；
- 可编程空闲状态时钟极性；
- 可以输出主时钟（MCK）；
- 发送和接收支持DMA功能。

24.3. SPI 功能说明

24.3.1. SPI 结构框图

图 24-1. SPI 结构框图



24.3.2. SPI 信号线描述

常规配置（非 SPI 四线模式）

表 24-1. 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 从机双向线模式：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：NSSDRV=1 时，为 NSS 输出，适用于单主机模式；NSSDRV=0 时，为 NSS 输入，适用于多主机模式。

引脚名称	方向	描述
		从机硬件 NSS 模式：为 NSS 输入，作为从机的片选信号。

SPI 四线配置

SPI 默认配置为单路模式，当 SPI_QCTL 中的 QMOD 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。

在 SPI 四线模式下，SPI 通过以下 6 个引脚与外部设备连接：

表 24-2. SPI 四线信号描述

引脚名称	方向	描述
SCK	O	SPI 时钟输出
MOSI	I / O	发送或接收数据 0 线
MISO	I / O	发送或接收数据 1 线
IO2	I / O	发送或接收数据 2 线
IO3	I / O	发送或接收数据 3 线
NSS	O	NSS 输出

24.3.3. 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 24-2. 常规模式下的 SPI 时序图

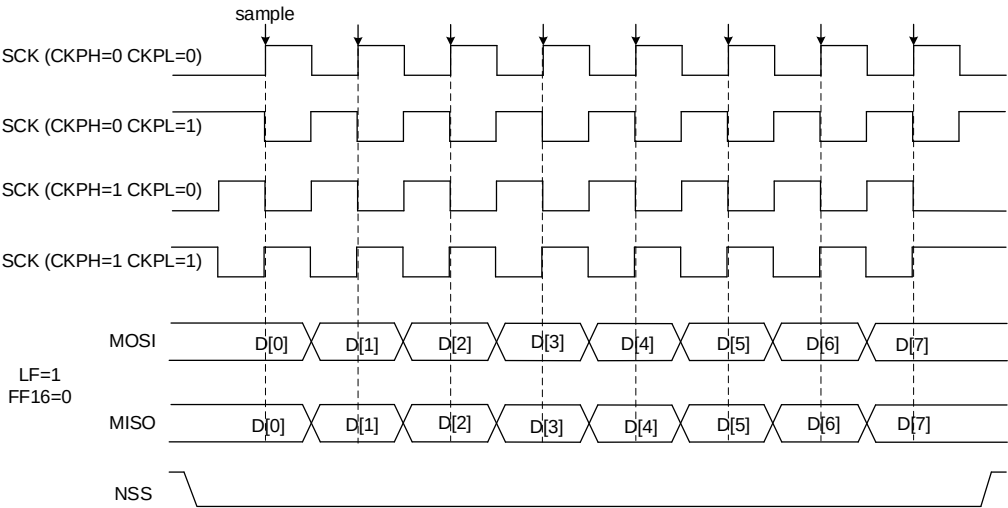
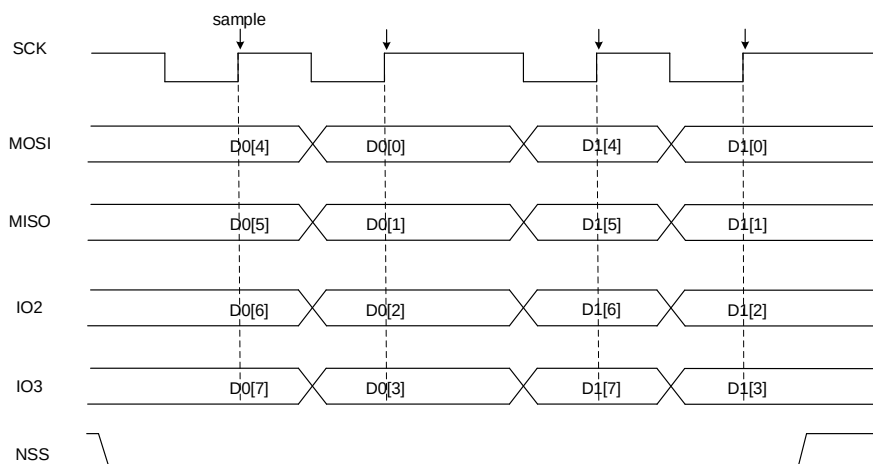


图 24-3. SPI 四线模式下的 SPI 时序图(CKPL=1, CKPH=1, LF=0)



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。

24.3.4. 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 模式 (SWNSSEN = 1) 下，SPI 根据 SWNSS 位得到 NSS 电平。只有当 NSS 为低电平时，才能发送或接收数据。在软件 NSS 模式下，不使用 NSS 引脚。

表 24-3. 从机模式 NSS 功能

模式	寄存器配置	描述
从机硬件 NSS 模式	MSTMOD = 0 SWNSSEN = 0	SPI 从机 NSS 电平从 NSS 引脚获取。
从机软件 NSS 模式	MSTMOD = 0 SWNSSEN = 1	SPI 从机 NSS 电平由 SWNSS 位决定。 SWNSS = 0: NSS 电平为低 SWNSS = 1: NSS 电平为高

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 **SWNSS** 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 (**SWNSSSEN**=0, **NSSDRV**=1)。使能 **SPI** 后, **NSS** 变为低电平。

应用程序可以使用一个通用 **I/O** 口作为 **NSS** 引脚, 以实现更加灵活的 **NSS** 应用。

表 24-4. 主机模式 NSS 功能

模式	寄存器配置	描述
主机硬件 NSS 输出模式	MSTMOD = 1 SWNSSSEN = 0 NSSDRV =1	适用于单主机模式, 主机使用 NSS 引脚控制 SPI 从设备, 此时 NSS 配置为硬件输出模式。使能 SPI 后 NSS 为低电平。
主机硬件 NSS 输入模式	MSTMOD = 1 SWNSSSEN = 0 NSSDRV =0	适用于多主机模式, 此时 NSS 配置为硬件输入模式, 一旦 NSS 引脚被拉低, SPI 将自动进入从机模式, 并且产生主机配置错误, CONFERR 位置 1。
主机软件 NSS 模式	MSTMOD = 1 SWNSSSEN = 1 SWNSS = 0 NSSDRV : 不要求	适用于多主机模式, 一旦 SWNSS = 0, SPI 将自动进入从机模式, 并且产生主机配置错误, CONFERR 位置 1。
	MSTMOD = 1 SWNSSSEN = 1 SWNSS = 1 NSSDRV : 不要求	从机可以使用硬件或软件 NSS 模式

24.3.5. SPI 运行模式

表 24-5. SPI 运行模式

模式	描述	寄存器配置	使用的数据引脚
MFD	全双工主机模式	MSTMOD = 1 RO = 0 BDEN = 0 BDOEN : 不要求	MOSI : 发送 MISO : 接收
MTU	单向线连接主机发送模式	MSTMOD = 1 RO = 0 BDEN = 0 BDOEN : 不要求	MOSI : 发送 MISO : 不使用
MRU	单向线连接主机接收模式	MSTMOD = 1 RO = 1 BDEN = 0	MOSI : 不使用 MISO : 接收

模式	描述	寄存器配置	使用的数据引脚
		BDOEN: 不要求	
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 BDOEN = 0	MOSI: 不使用 MISO: 接收

图 24-4. 典型的全双工模式连接

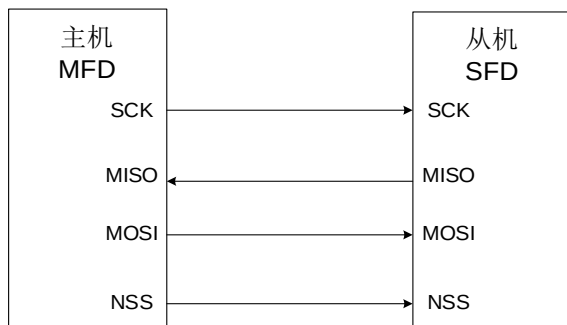


图 24-5. 典型的单工模式连接（主机：接收，从机：发送）

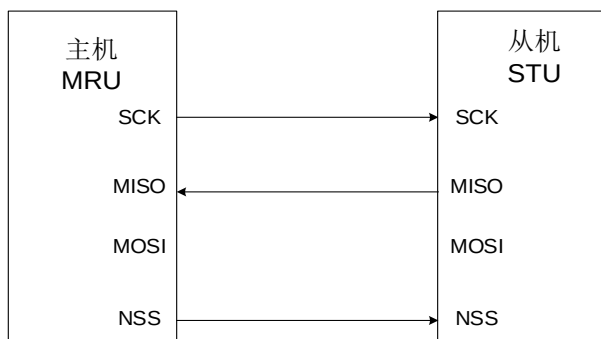


图 24-6. 典型的单工模式连接（主机：只发送，从机：接收）

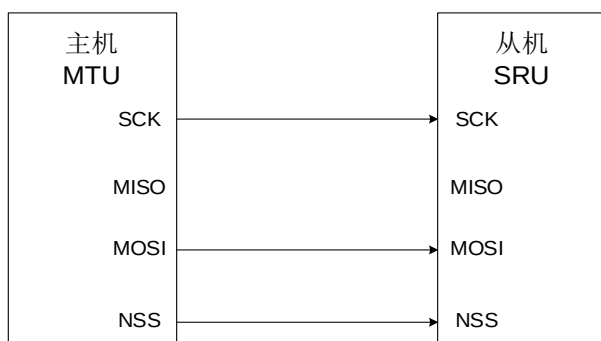
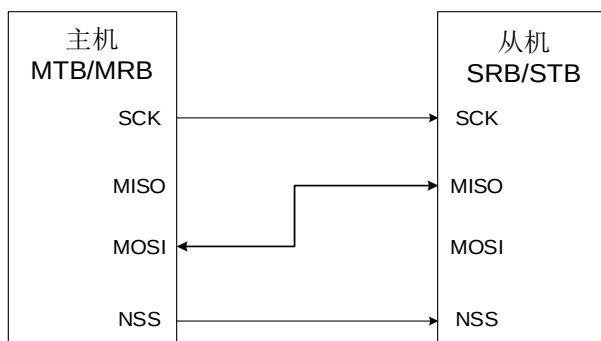


图 24-7. 典型的双向线连接



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. 如果工作在NSSP模式, 需要将SPI_CTL1中的NSSP位置1, 否则, 忽略此步骤。
8. 根据上面描述的运行模式, 配置MSTMOD位、RO位、BDEN位和BDOEN位。
9. 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。
10. 使能SPI (将SPIEN位置1)。

SPI 基本发送和接收流程

发送流程

在完成初始化过程之后, SPI 模块使能并保持在空闲状态。在主机模式下, 当软件写一个数据到发送缓冲区时, 发送过程开始。在从机模式下, 当 SCK 引脚上的 SCK 信号开始翻转, 且 NSS 引脚电平为低, 发送过程开始。所以, 在从机模式下, 应用程序必须确保在数据发送开始前, 数据已经写入发送缓冲区中。

当 SPI 开始发送一个数据帧时, 首先将这个数据帧从数据缓冲区加载到移位寄存器中, 然后开始发送加载的数据。在数据帧的第一位发送之后, TBE (发送缓冲区空) 位置 1。TBE 标志位置 1, 说明发送缓冲区为空, 此时如果需要发送更多数据, 软件应该继续写 SPI_DATA 寄存器。

在主机模式下, 若想要实现连续发送功能, 那么在当前数据帧发送完成前, 软件应该将下一个数据写入 SPI_DATA 寄存器中。

接收流程

在最后一个采样时钟边沿之后, 接收到的数据将从移位寄存器存入到接收缓冲区, 且 RBNE (接收缓冲区非空) 位置 1。软件通过读 SPI_DATA 寄存器获得接收的数据, 此操作会自动清除 RBNE 标志位。在 MRU 和 MRB 模式中, 为了接收下一个数据帧, 硬件需要连续发送时钟信号, 而在全双工主机模式 (MFD) 中, 当发送缓冲区非空时, 硬件只接收下一个数据帧。

SPI 不同模式下的操作流程 (非 SPI 四线模式, TI 模式或 NSSP 模式)

在全双工模式下, 无论是 MFD 模式或者 SFD 模式, 应用程序都应该监视 RBNE 标志位和 TBE 标志位, 并且遵循上文描述的操作流程。

除了忽略 RBNE 位和 RXORERR 位, 且只执行上述的发送流程之外, 发送模式 (MTU, MTB, STU 和 STB) 与全双工模式类似。

在主机接收模式 (MRU 或 MRB) 下, 全双工模式和发送模式是不同的。在 MRU 模式或 MRB 模式下, 在 SPI 使能后, SPI 产生连续的 SCK 信号, 直到 SPI 停止。所以, 软件应该忽略 TBE 标志位, 并且在 RBNE 位置 1 后及时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 24-8. 主机 TI 模式在不连续发送时的时序图

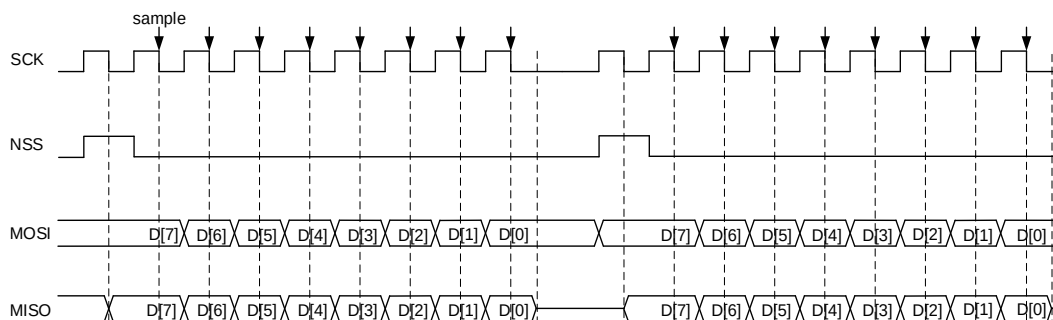
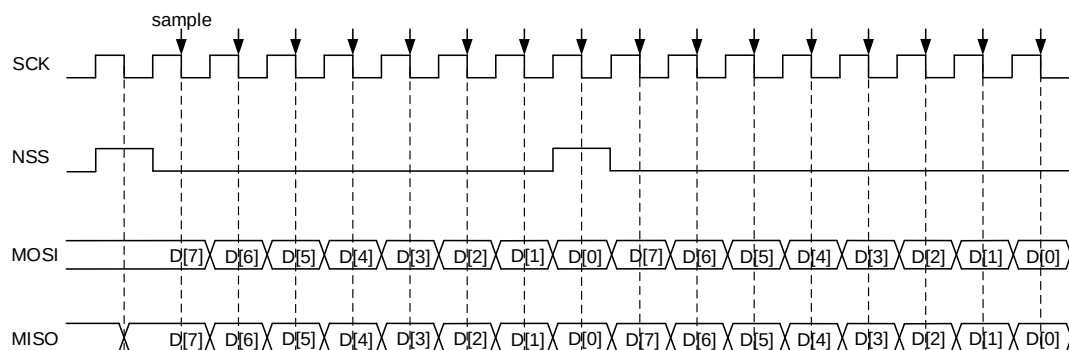
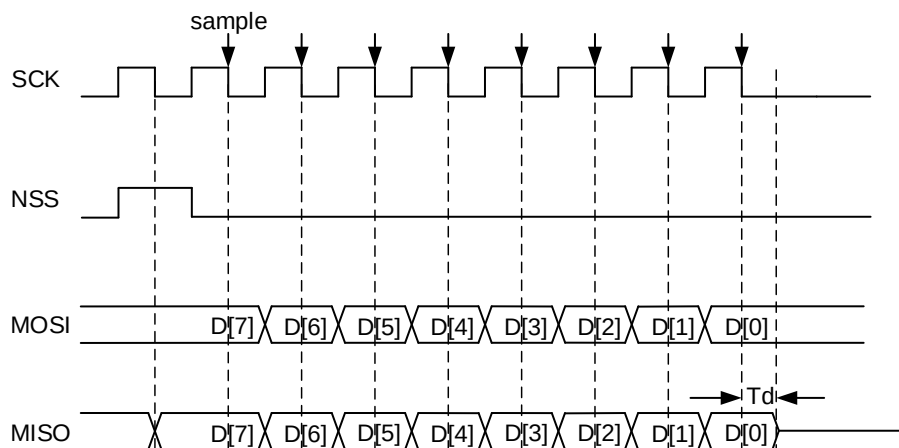


图 24-9. 主机 TI 模式在连续发送时的时序图



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。但是在连续传输中，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 24-10. 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。

$$T_d = \frac{T_{\text{bit}}}{2} + 5 * T_{\text{pclk}} \quad (24-1)$$

例如，如果 PSC[2:0] = 010，那么 T_d 数值为 $9 * T_{\text{pclk}}$ 。

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

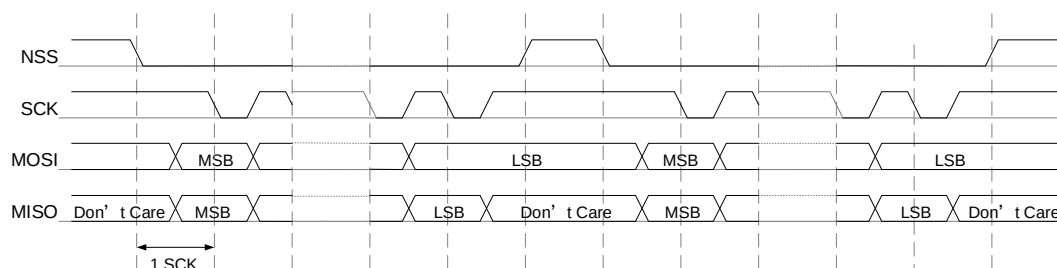
配置 SPI_CTL1 寄存器中的 NSSP 位使能该功能，为了确保使用该功能实现，需满足以下几个条件：配置设备为主机模式，使用普通 SPI 协议的数据帧格式，同时在第一个时钟跳变沿采样数据。

总之：NSSP = 1，MSTMOD = 1，CKPH = 0。

当使用 NSS 脉冲模式时，根据内部数据发送缓冲区的状态，NSS 脉冲会在两个连续的数据帧之间产生，且持续时间至少为 1 个 SCK 时钟周期。如果数据发送缓冲区保持为空，可能会持续多个 SCK 时钟周期。NSS 脉冲功能专为单一的主从应用设计，支持从机锁存数据。

下图描述了 NSS 脉冲模式在主机连续发送时的时序图。

图 24-11. 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。

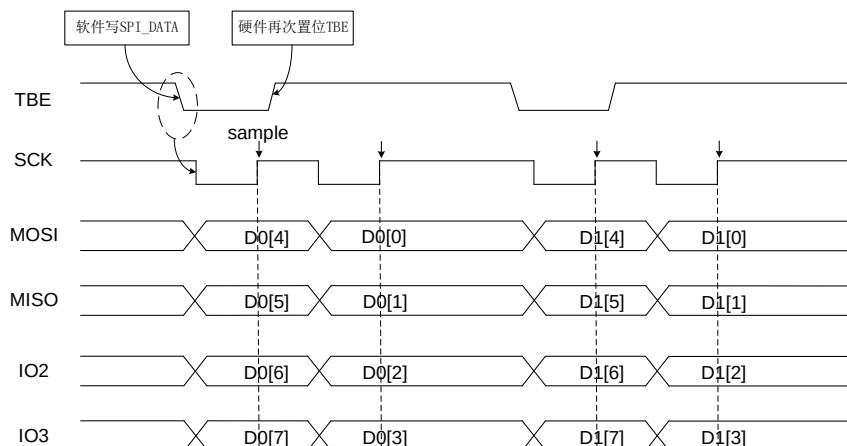
四线写模式

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，SPI 将会通过这四个引脚发送写入的数据。一旦 SPI 开始数据传输，它总是在数据帧结束时检查 TBE 标志位的状态，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 24-12. SPI 四线模式四线写操作时序图



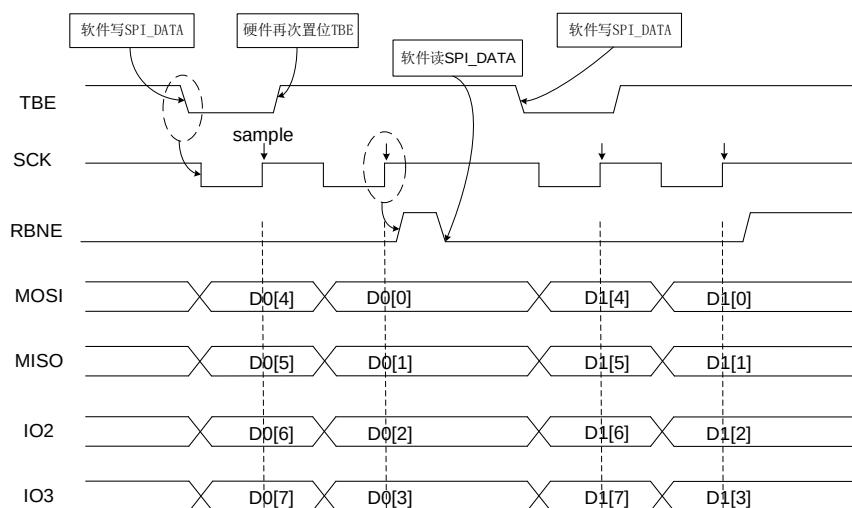
四线读模式

当 `SPI_QCTL` 寄存器中的 `QMOD` 位和 `QRD` 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。当数据写入 `SPI_DATA` 寄存器(此时 `TBE` 位被清零)且 `SPIEN` 位置 1 时, SPI 开始在 `SCK` 信号线上产生时钟信号。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 `QMOD` 位和 `QRD` 位置 1, 然后将 `SPI_CTL0` 中的 `SPIEN` 位置 1 来使能 SPI 功能;
3. 写任意数据(例如 `0xFF`)到 `SPI_DATA` 寄存器;
4. 等待 `RBNE` 位置 1, 然后读 `SPI_DATA` 寄存器来获取接收的数据;
5. 写任意数据(例如 `0xFF`)到 `SPI_DATA` 寄存器, 以接收下一个字节数据。

图 24-13. 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。然后等待 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

当应用程序不想接收数据时，可以禁用 SPI，然后等待 TRANS=0 以确保正在进行的传输完成。

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 四路模式或者关闭 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。

24.3.6. 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，SPI 将会发出一个 DMA 请求，然后 DMA 应答该请求，并自动从 SPI_DATA 寄存器读取数据。

24.3.7. 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，软件应该把 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。

注意：当 SPI 处于从机模式且 CRC 功能使能时，无论 SPI 是否使能，CRC 计算器都对输入 SCK 时钟敏感。只有当时钟稳定时，软件才能启用 CRC，以避免错误的 CRC 计算。当 SPI 作为从机工作时，在数据阶段和 CRC 阶段之间，内部 NSS 信号需要保持低电平。

当配置 SPI 为从模式并且使用 CRC 功能时，即使 NSS 引脚为高时仍然会执行 CRC 的计算（当 NSS 信号为高时，只要 SCK 引脚上有时钟脉冲，则 CRC 计算会继续执行）。当主设备交替地与多个从设备进行通信时，将会出现这种情况，此时建议在 NSS 信号为低时重启 CRC 功能。当从设备未选中（NSS 信号为高）转换到被选中为一个新的从设备（NSS 信号为低）的时候，为了保持主从设备端下次 CRC 计算结果的同步，应该清除主从两端的 CRC 数值。建议按照下述步骤清除 CRC 数值：

1. 关闭 SPI 模块（SPIEN=0）；

2. 清除 CRCEN 位 (CRCEN=0) ;
3. 设置 CRCEN 位 (CRCEN=1) ;
4. 使能 SPI 模块 (SPIEN=1) 。

24.3.8. SPI 中断

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。

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 CRC 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。

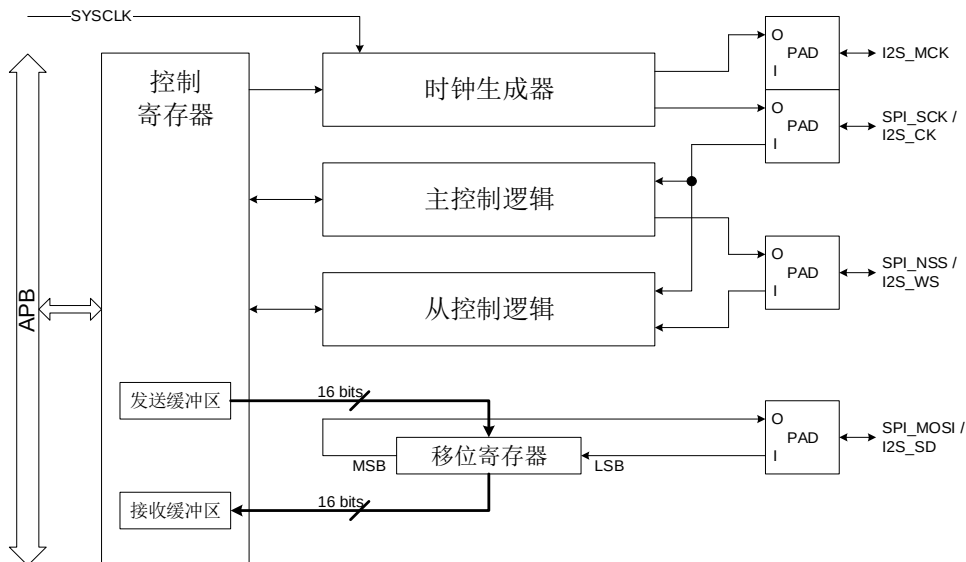
表 24-6. SPI 中断请求

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	

24.4. I2S 功能说明

24.4.1. I2S 结构框图

图 24-14. I2S 结构框图



I2S 功能有 5 个子模块, 分别是控制寄存器、时钟生成器、主机控制逻辑、从机控制逻辑和移位寄存器。所有的用户可配置寄存器都在控制寄存器模块实现, 其中包括发送缓冲区和接收缓冲区。时钟生成器用来在主机模式下生成 I2S 通信时钟。主机控制逻辑用来在主机模式下生成 I2S_WS 信号并控制通信。从机控制逻辑根据接收到的 I2SCK 和 I2S_WS 信号来控制从机模式的通信。移位寄存器控制 I2S_SD 上的串行数据发送和接收。

24.4.2. 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 F_s 的时钟频率，其中 F_s 是音频采样率。

24.4.3. I2S 音频标准

I2S 音频标准是通过设置 SPI_I2SCTL 寄存器中的 I2SSTD 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，I2S_WS 在数据的前一个时钟开始有效。各种配置情况的时序图如下所示。

图 24-15. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=0, CKPL=0)

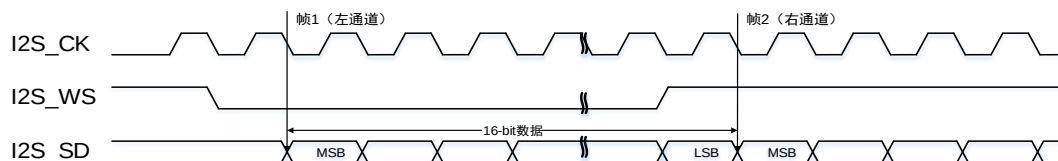
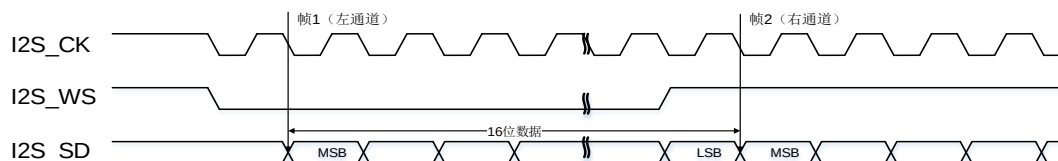


图 24-16. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=0, CKPL=1)



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 24-17. I2S 飞利浦标准时序图(DTLEN=10, CHLEN=1, CKPL=0)

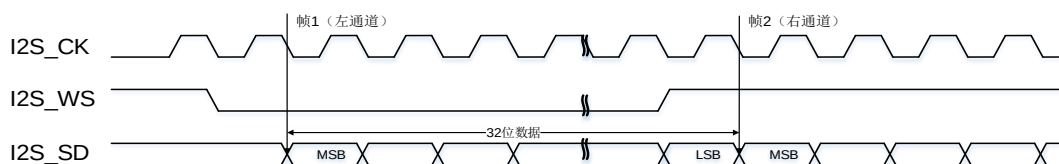
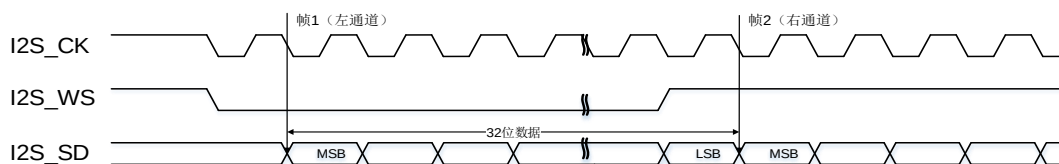


图 24-18. I2S 飞利浦标准时序图(DTLEN=10, CHLEN=1, CKPL=1)



当 32 位数据打包成 32 位数据帧的帧格式时，每完成 1 帧数据的传输需要访问 SPI_DATA 寄存器 2 次。在发送模式下，如果要发送一个 32 位数据，第一个写入 SPI_DATA 寄存器的数据应该是高 16 位数据，第二个数据应该是低 16 位数据。在接收模式下，如果要接收一个 32 位数据，第一个从 SPI_DATA 寄存器读到的数据应该是高 16 位数据，第二个数据应该是低 16 位数据。

图 24-19. I2S 飞利浦标准时序图(DTLEN=01, CHLEN=1, CKPL=0)

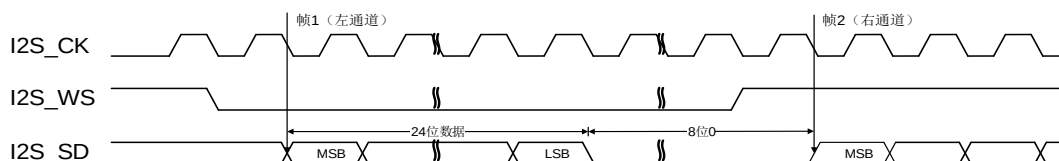
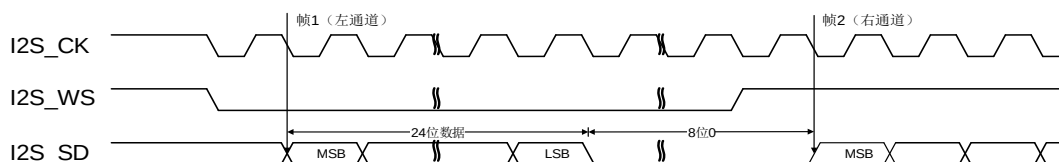


图 24-20. I2S 飞利浦标准时序图(DTLEN=01, CHLEN=1, CKPL=1)



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 位数据全是 0。

图 24-21. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=1, CKPL=0)

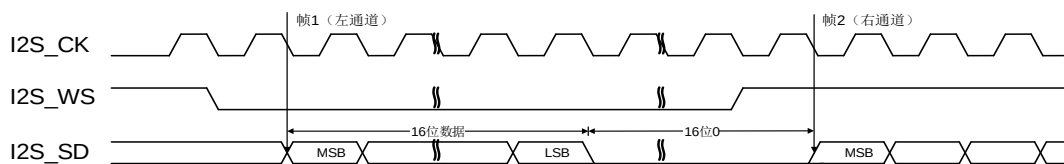
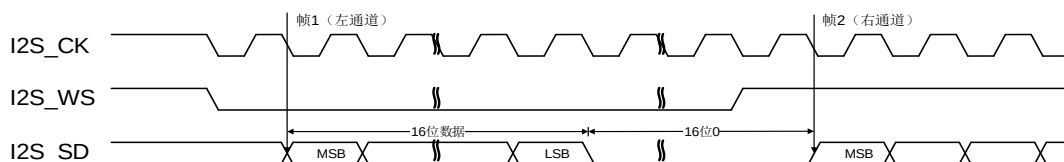


图 24-22. I2S 飞利浦标准时序图(DTLEN=00, CHLEN=1, CKPL=1)



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 24-23. MSB 对齐标准时序图(DTLEN=00, CHLEN=0, CKPL=0)

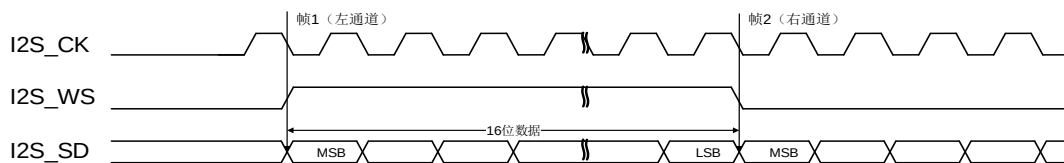


图 24-24. MSB 对齐标准时序图(DTLEN=00, CHLEN=0, CKPL=1)

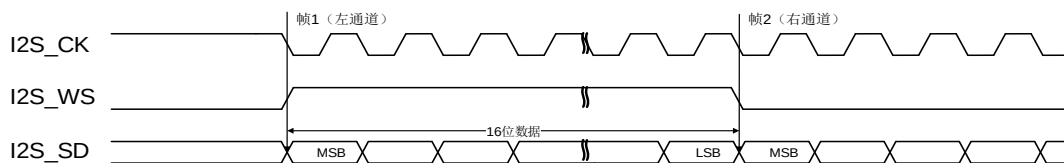


图 24-25. MSB 对齐标准时序图(DTLEN=10, CHLEN=1, CKPL=0)

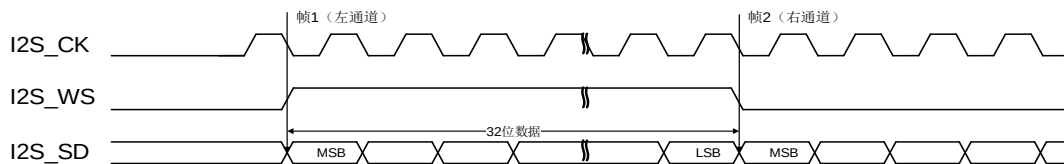


图 24-26. MSB 对齐标准时序图(DTLEN=10, CHLEN=1, CKPL=1)

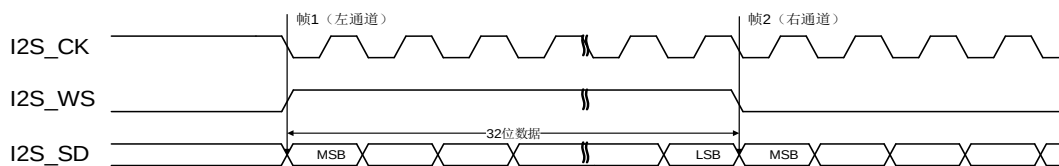


图 24-27. MSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=0)

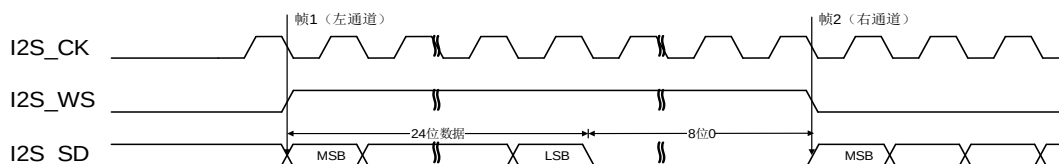


图 24-28. MSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=1)

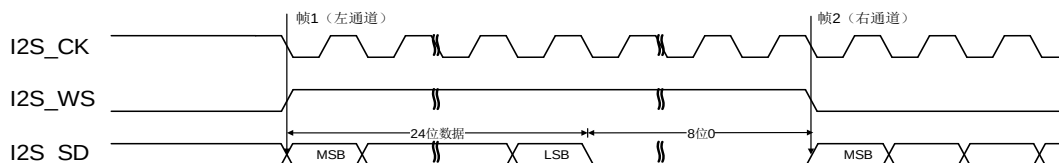


图 24-29. MSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=0)

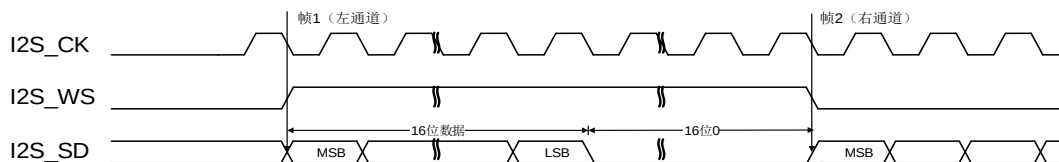
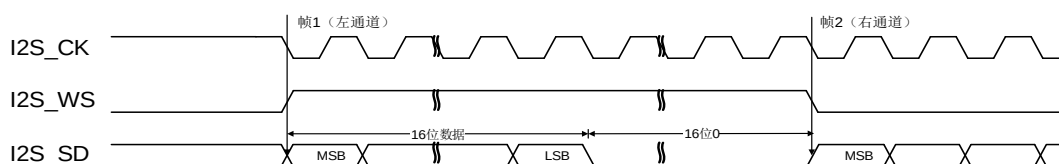


图 24-30. MSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=1)



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 24-31. LSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=0)

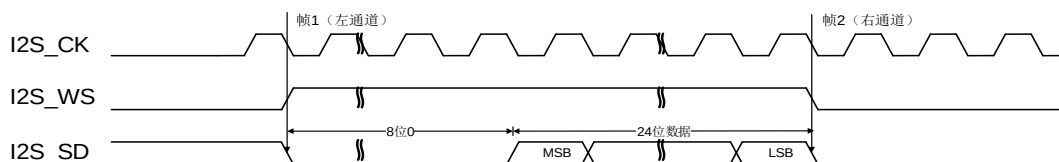
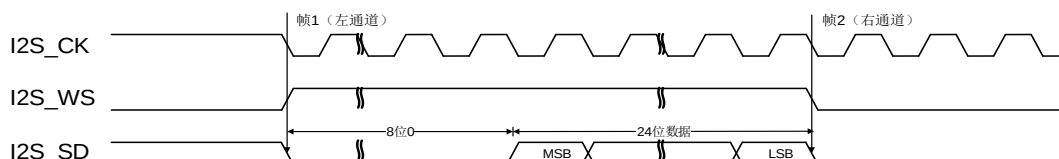


图 24-32. LSB 对齐标准时序图(DTLEN=01, CHLEN=1, CKPL=1)



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 24-33. LSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=0)

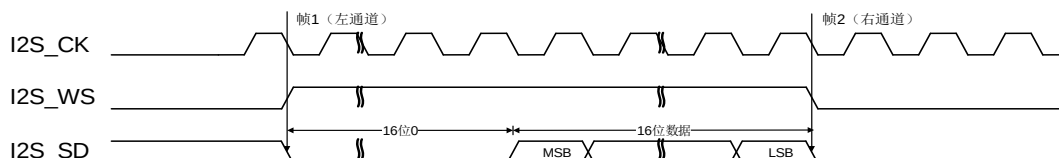
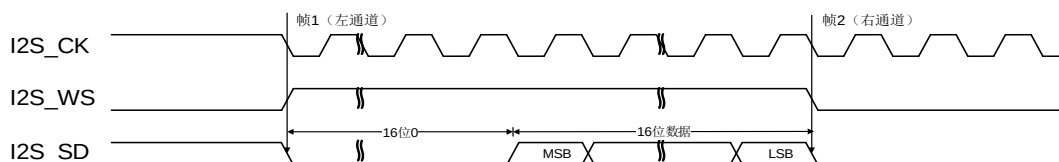


图 24-34. LSB 对齐标准时序图(DTLEN=00, CHLEN=1, CKPL=1)



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 24-35. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=0)

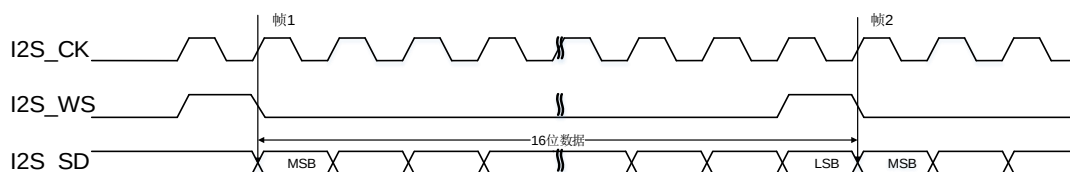


图 24-36. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=1)

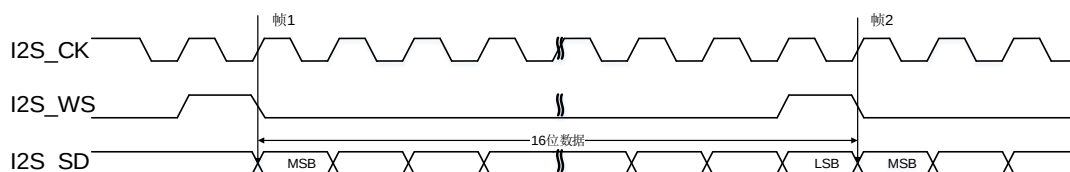


图 24-37. PCM 标准短帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=0)

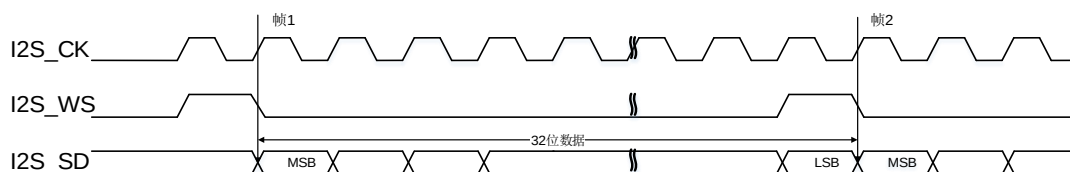


图 24-38. PCM 标准短帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=1)

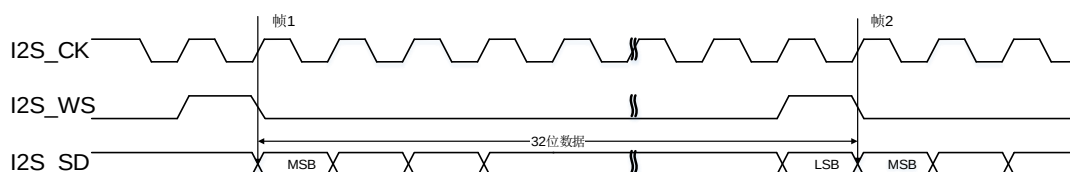


图 24-39. PCM 标准短帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=0)

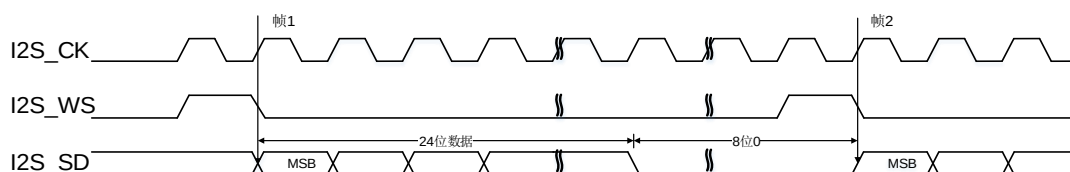


图 24-40. PCM 标准短帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=1)

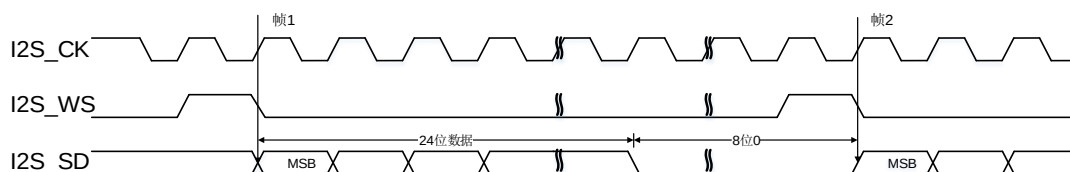


图 24-41. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=0)

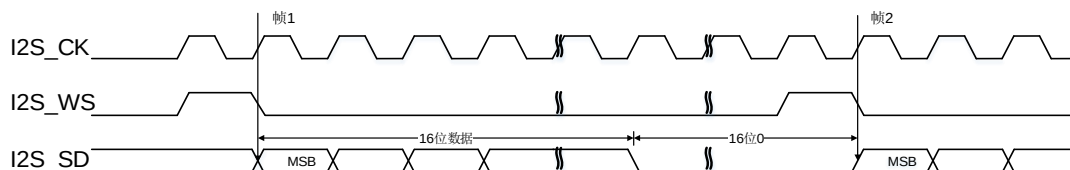
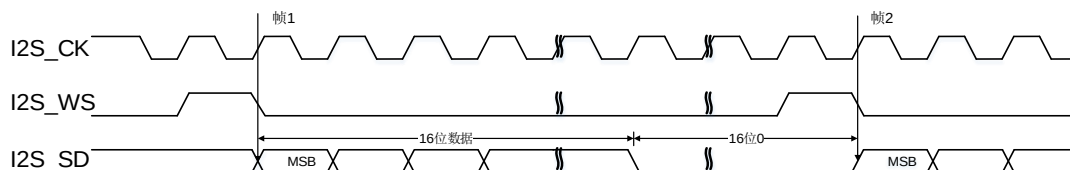


图 24-42. PCM 标准短帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=1)



长帧同步模式的各种配置情况时序图如下所示。

图 24-43. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=0)

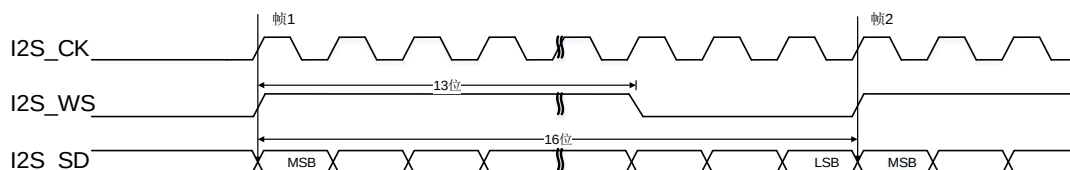


图 24-44. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=0, CKPL=1)

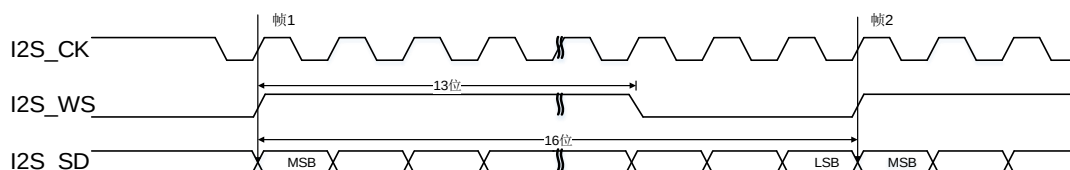


图 24-45. PCM 标准长帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=0)

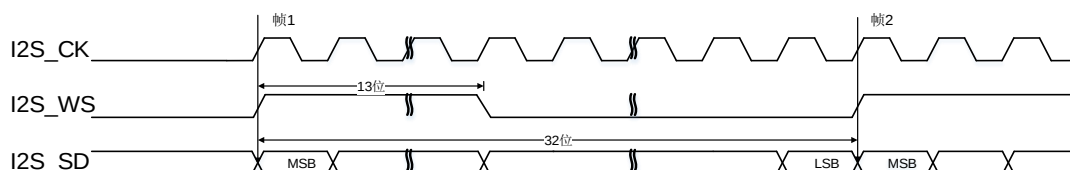


图 24-46. PCM 标准长帧同步模式时序图(DTLEN=10, CHLEN=1, CKPL=1)

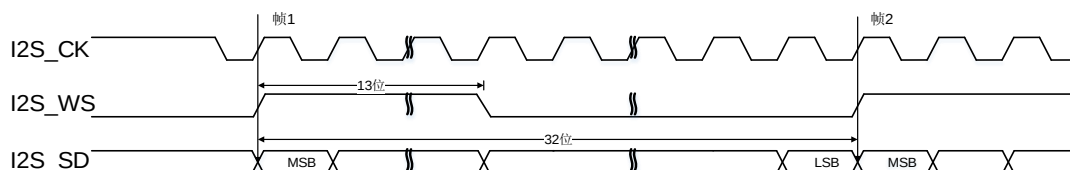


图 24-47. PCM 标准长帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=0)

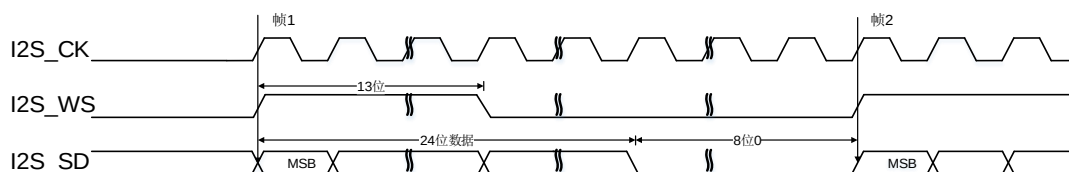


图 24-48. PCM 标准长帧同步模式时序图(DTLEN=01, CHLEN=1, CKPL=1)

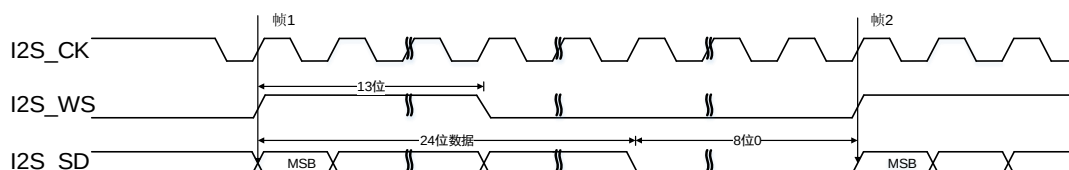


图 24-49. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=0)

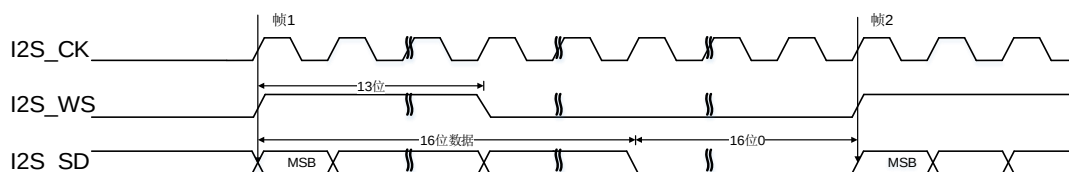
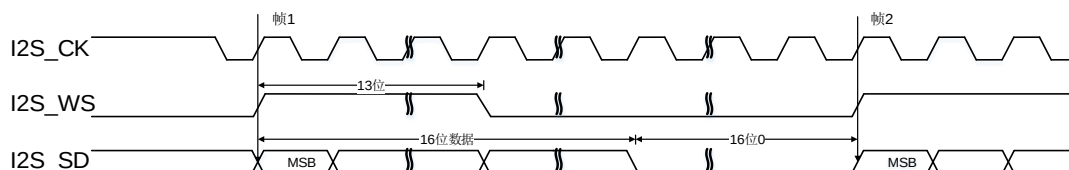
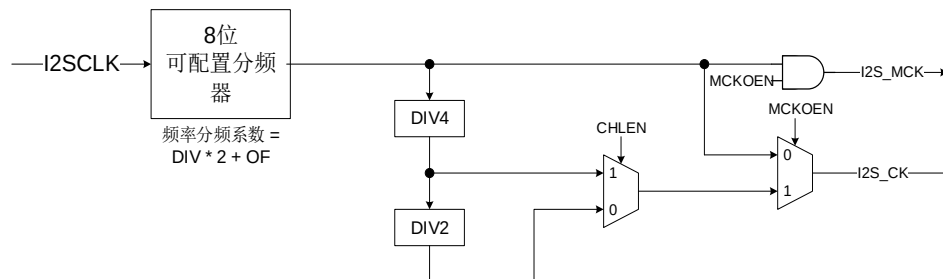


图 24-50. PCM 标准长帧同步模式时序图(DTLEN=00, CHLEN=1, CKPL=1)



24.4.4. I2S 时钟

图 24-51. I2S 时钟生成结构框图



I2S 时钟生成器框图如 [图 24-51. 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。时钟源是系统时钟或 CK_PLL1 (CK_SYS/CK_PLL1)。比特率可以通过 [表 24-7. I2S 比特率计算公式](#) 所示的公式计算。

注意：I2S 串行时钟的配置值需设置为低于相应 PCLK 时钟的 1/6 倍以下(不包含 1/6)。

表 24-7. 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 24-8. 音频采样频率计算公式](#)所列的公式进行配置。

表 24-8. 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	$I2SCLK / (256 * (DIV * 2 + OF))$

24.4.5. 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 24-9. 各种运行模式下 I2S 接口信号的方向](#)所示。

表 24-9. 各种运行模式下 I2S 接口信号的方向

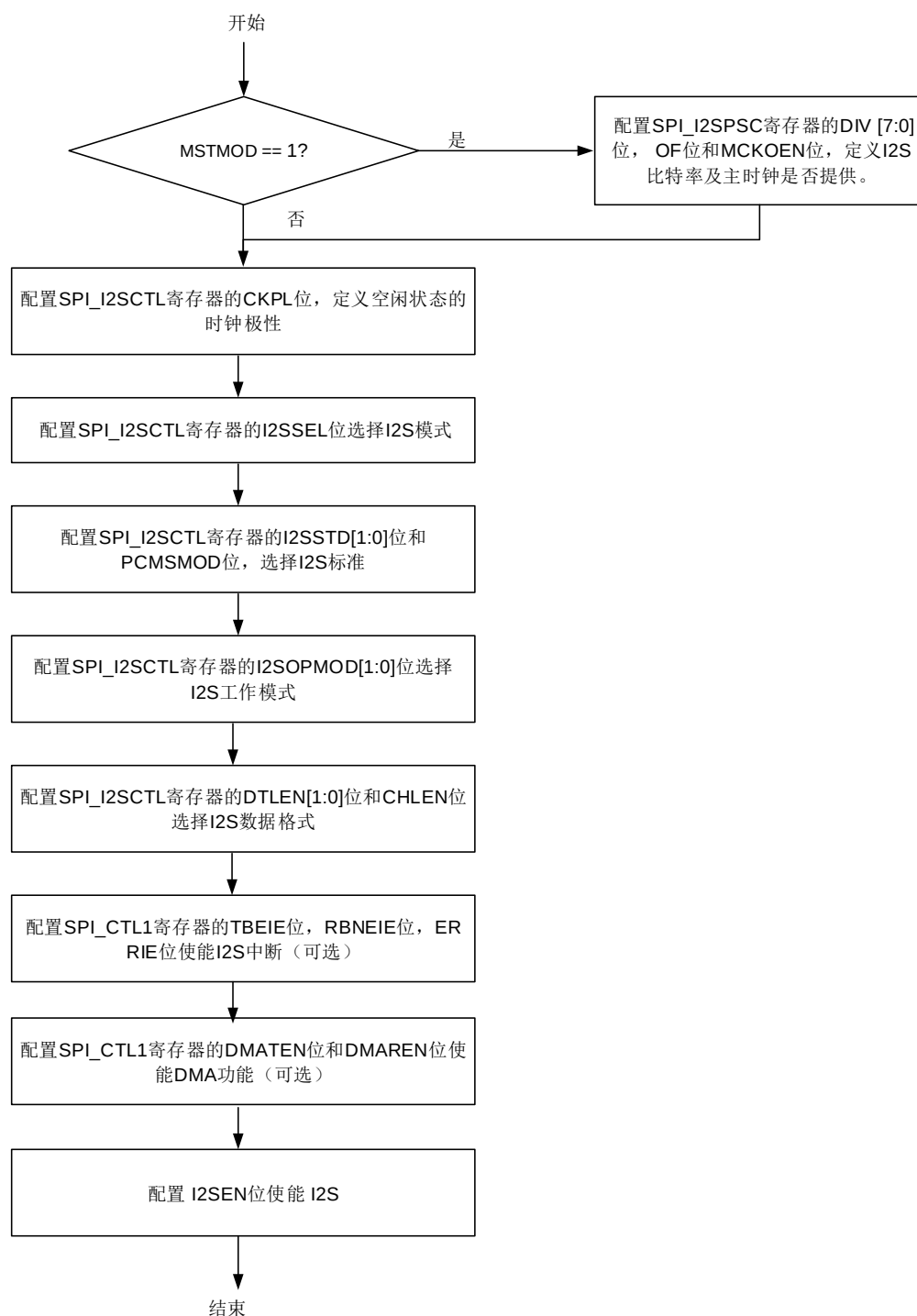
运行模式	I2S_MCK	I2S_CK	I2S_WS	I2S_SD
主机发送	输出或 NU(1)	输出	输出	输出
主机接收	输出或 NU(1)	输出	输出	输入
从机发送	输入或 NU(1)	输入	输入	输出
从机接收	输入或 NU(1)	输入	输入	输入

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 初始化过程如 [图 24-52. I2S 初始化流程](#) 所示。

图 24-52. I2S 初始化流程



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。

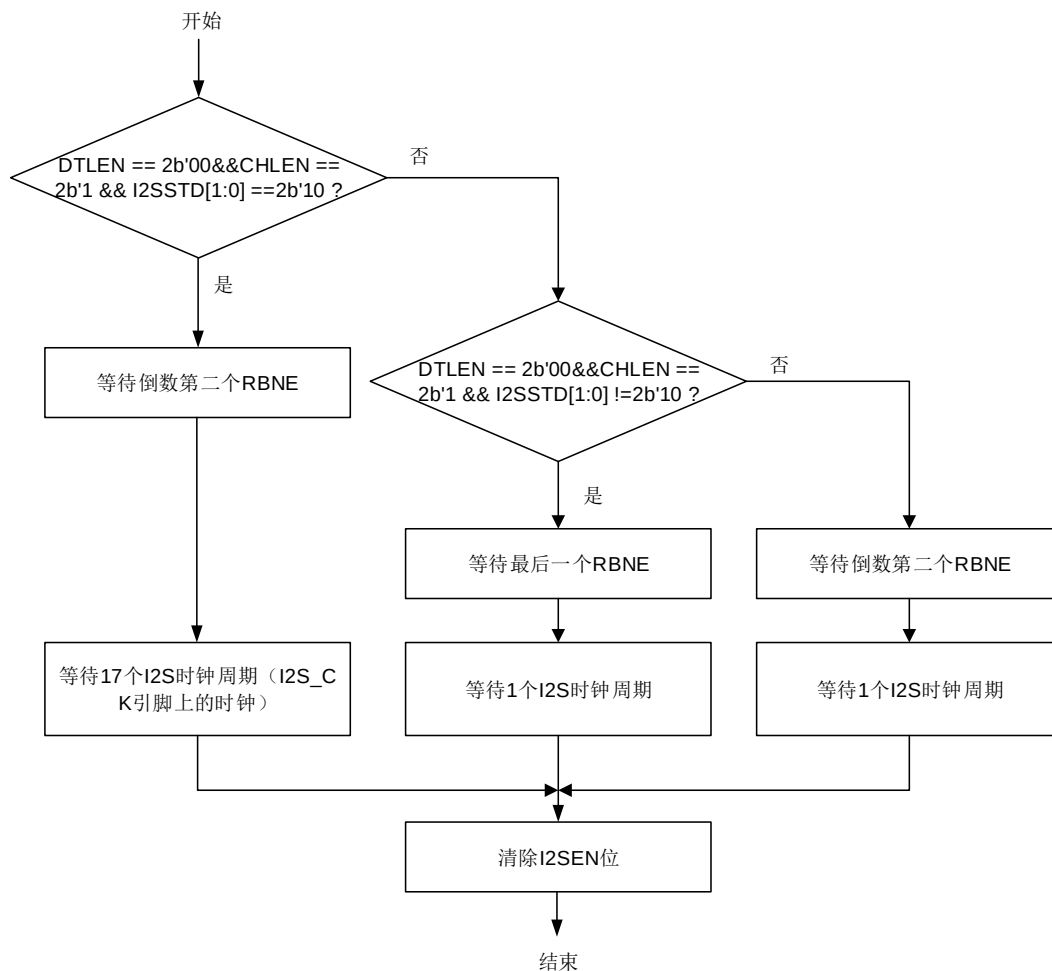
I2S 主机接收流程

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 [图 24-53. I2S 主机接收禁能流程](#) 所示。

图 24-53. I2S 主机接收禁能流程



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。

为了关闭 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。

24.4.6. 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。

24.4.7. I2S 中断

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，无法通过软件进行操作。该标志位不会产生如何中断。

■ 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。

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，当有效的 SCK 信号开始发送时，如果发送缓冲区为空时，发送欠载错误标志 TXURERR 将会置位。

■ 接收过载错误标志(RXORERR):

当接收缓冲区已满且又接收到一个新的数据时，接收过载错误标志 RXORERR 置位。当接收过载发生时，接收缓冲区中的数据没有更新，新接收的数据丢失。

■ 帧错误(FERR):

在从 I2S 模式下，I2S 模块监视 I2S_WS 信号，如果 I2S_WS 信号在一个错误的位置发生翻转，将会置位 FERR 帧错误标志位。

[表 24-10. I2S 中断](#)总结了 I2S 中断事件和相应的使能位。

表 24-10. I2S 中断

中断事件	描述	清除方式	中断使能位
TBE	发送缓冲区空	写 SPI_DATA 寄存器	TBEIE
RBNE	接收缓冲区非空	读 SPI_DATA 寄存器	RBNEIE
TXURERR	发送欠载错误	读 SPI_STAT 寄存器	ERRIE
RXORERR	接收过载错误	读 SPI_DATA 寄存器，然后再读 SPI_STAT 寄存器	
FERR	I2S 帧错误	读 SPI_STAT 寄存器	

24.5. 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

24.5.1. 控制寄存器 0 (SPI_CTL0)

地址偏移：0x00

复位值：0x0000 0000

该寄存器可以按字（32 位）访问。

该寄存器在 I2S 模式下没有意义。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BDEN	BDOEN	CRCEN	CRCNT	FF16	RO	SWNSS EN	SWNSS	LF	SPIEN	PSC [2:0]		MSTMOD	CKPL	CKPH	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		RW	RW	RW	

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	BDEN	双向数据模式 0: 2 线单向传输模式 1: 1 线双向传输模式。数据在主机 MOSI 引脚和从机 MISO 引脚之间传输。
14	BDOEN	双向传输输出使能 当 BDEN 置位时，该位决定了数据的传输方向。 0: 工作在只接收模式 1: 工作在只发送模式
13	CRCEN	CRC 计算使能 0: CRC 计算禁止 1: CRC 计算使能
12	CRCNT	下一次传输 CRC 0: 下一次传输值为数据 1: 下一次传输值为 CRC 值（TCR） 当数据传输由 DMA 管理时，CRC 值由硬件传输，该位应该被清零。

在全双工和只发送模式下，当最后一个数据写入 SPI_DATA 寄存器后应将该位置 1。在只接收模式下，在接收完倒数第二个数据后应将该位置 1。

11	FF16	数据帧格式 0: 8 位数据帧格式 1: 16 位数据帧格式
10	RO	只接收模式 当 BDEN 清零时，该位决定了数据的传输方向。 0: 全双工模式 1: 只接收模式
9	SWNSSEN	NSS 软件模式选择 0: NSS 硬件模式，NSS 电平取决于 NSS 引脚 1: NSS 软件模式，NSS 电平取决于 SWNSS 位 该位在 SPI TI 模式下没有意义。
8	SWNSS	NSS 软件模式下 NSS 引脚选择 0: NSS 引脚拉低 1: NSS 引脚拉高 只有在 SWNSSEN 置位时，该位有效。 该位在 SPI TI 模式下没有意义。
7	LF	最低有效位先发模式 0: 先发送最高有效位 1: 先发送最低有效位 该位在 SPI TI 模式下没有意义。
6	SPIEN	SPI 使能 0: SPI 设备禁止 1: SPI 设备使能
5:3	PSC[2:0]	主时钟预分频选择 000: PCLK/2 100: PCLK/32 001: PCLK/4 101: PCLK/64 010: PCLK/8 110: PCLK/128 011: PCLK/16 111: PCLK/256 当使用 SPI0 时，PCLK=PCLK2，当使用 SPI1 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

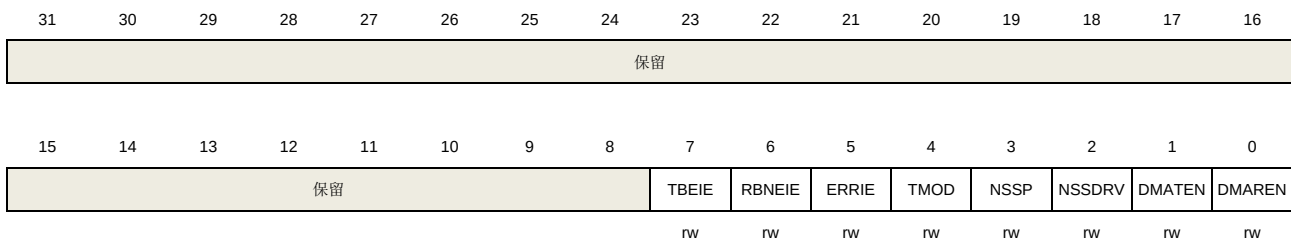
1: 在第二个时钟跳变沿时钟跳变沿采集第一个数据

24.5.2. 控制寄存器 1 (SPI_CTL1)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器可以按字 (32 位) 访问。



位/位域	名称	描述
31:8	保留	必须保持复位值。
7	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。

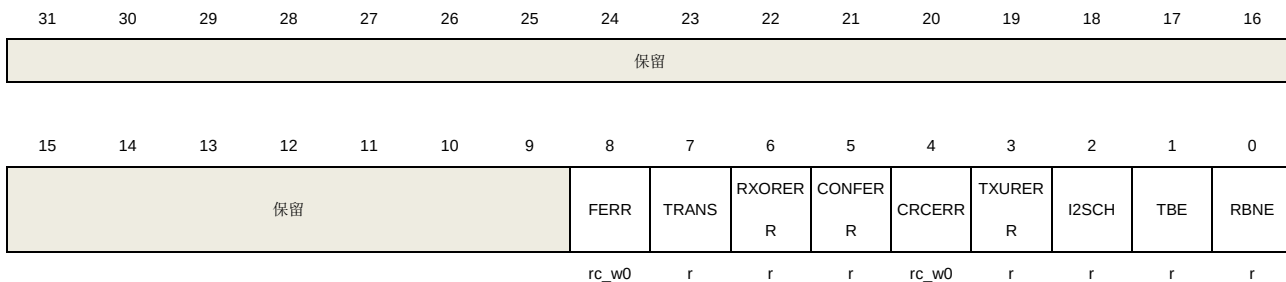
1	DMATEN	发送缓冲区 DMA 使能 0: 发送缓冲区 DMA 禁止 1: 发送缓冲区 DMA 使能。当 SPI_STAT 中的 TBE 置位时，将会在相应的 DMA 通道上产生一个 DMA 请求。
0	DMAREN	接收缓冲区 DMA 使能 0: 接收缓冲区 DMA 禁止 1: 接收缓冲区 DMA 使能。当 SPI_STAT 中的 RBNE 置位时，将会在相应的 DMA 通道上产生一个 DMA 请求。

24.5.3. 状态寄存器 (SPI_STAT)

地址偏移: 0x08

复位值: 0x0002

该寄存器可以按字（32 位）访问。



位/位域	名称	描述
31:9	保留	必须保持复位值。
8	FERR	帧错误 SPI TI 模式: 0: 没有 TI 模式帧错误发生 1: TI 模式帧错误发生 I2S 模式: 0: 没有 I2S 帧错误发生 1: I2S 帧错误发生 该位由硬件置位，可以通过写 0 清除。
7	TRANS	通信进行中标志 0: SPI 或 I2S 空闲 1: SPI 或 I2S 当前正在发送或接收数据

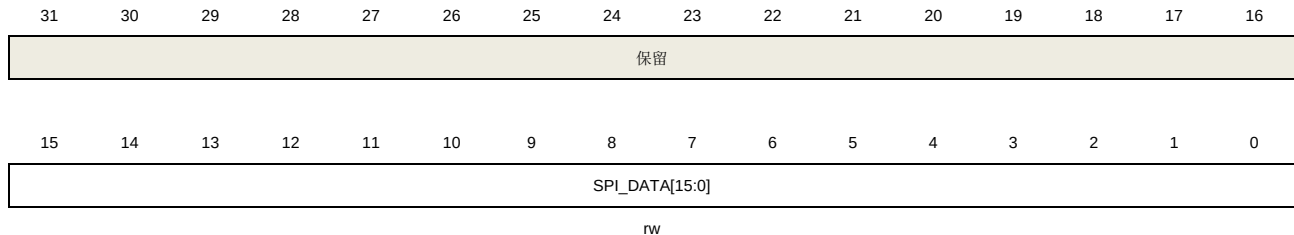
		该位由硬件置位和清除。
6	RXORERR	接收过载错误标志 0: 没有接收过载错误发生 1: 接收过载错误发生 该位由硬件置位，软件序列清零。软件序列为：先读 SPI_DATA 寄存器，然后读 SPI_STAT 寄存器。
5	CONFERR	SPI 配置错误 0: 无配置错误发生 1: 配置错误发生（主机模式下，在硬件 NSS 模式时 NSS 引脚被拉低，或者软件 NSS 模式时 SWNSS 位为 0，都会产生 CONFERR 错误） 该位由硬件置位，软件序列清零。软件序列为：先读或写 SPI_STAT 寄存器，然后写 SPI_CTL0 寄存器。 I2S 模式下不使用该位。
4	CRCERR	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，通过读 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

24.5.4. 数据寄存器 (SPI_DATA)

地址偏移：0x0C

复位值：0x0000 0000

该寄存器可以按字（32 位）访问。



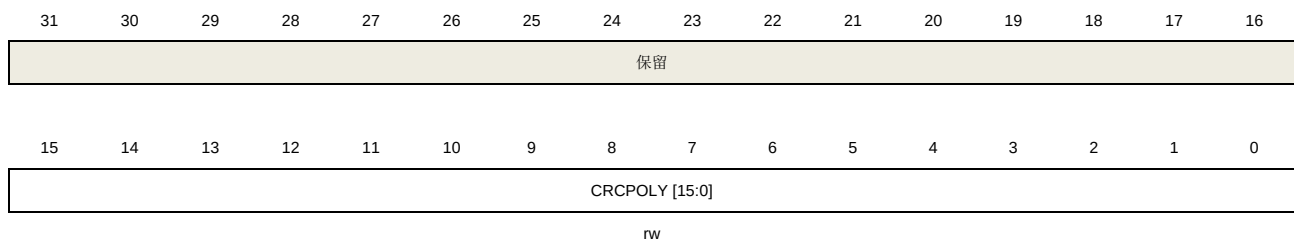
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	SPI_DATA[15:0]	数据传输寄存器值 硬件有两个缓冲区：发送缓冲区和接收缓冲区。向 SPI_DATA 写数据将会把数据存入发送缓冲区，从 SPI_DATA 读数据，将从接收缓冲区获得数据。 当数据帧格式为 8 位时，SPI_DATA[15:8]强制为 0，SPI_DATA[7:0]用来发送和接收数据，发送和接收缓冲区都是 8 位。如果数据帧格式为 16 位，SPI_DATA[15:0]用于发送和接收数据，发送和接收缓冲区也是 16 位。

24.5.5. CRC 多项式寄存器 (SPI_CRCPOLY)

地址偏移：0x10

复位值：0x0000 0007

该寄存器可以按字（32 位）访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	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。

24.5.6. 接收 CRC 寄存器 (SPI_RCR)

地址偏移：0x14

复位值：0x0000 0000

该寄存器可以按字（32 位）访问。



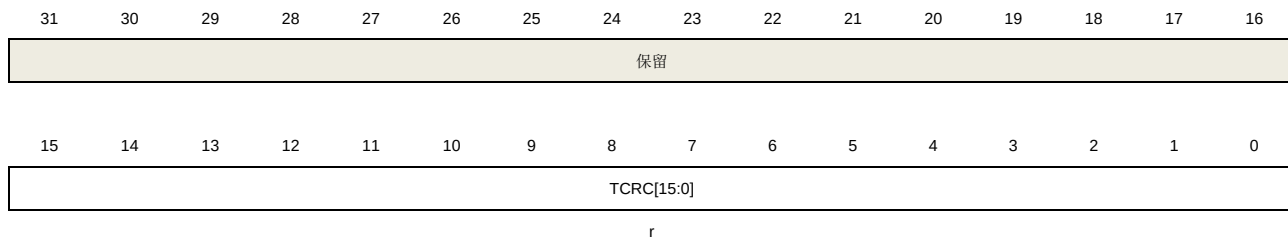
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	RCRC[15:0]	接收 CRC 寄存器值 当 SPI_CTL0 中的 CRCEN 置位时，硬件计算接收数据的 CRC 值，并保存到 RCR 寄存器中。如果是 8 位数据帧格式，CRC 计算基于 CRC8 标准进行，保存数据到 RCRC [7:0]。如果是 16 位数据帧格式，CRC 计算基于 CRC16 标准进行，保存数据到 RCRC [15:0]。 硬件在接收到每个数据位后都会计算 CRC 值，当 TRANS 置位时，读该寄存器将返回一个中间值。 当 SPI_CTL0 寄存器中的 CRCEN 位和 SPIEN 位清零时，该寄存器复位。

24.5.7. 发送 CRC 寄存器 (SPI_TCR)

地址偏移：0x18

复位值：0x0000 0000

该寄存器可以按字（32 位）访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	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 TCR

寄存器中。如果是 8 位数据帧格式，CRC 计算基于 CRC8 标准进行，保存数据到 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]。

硬件在发送出每个数据位后都会计算 CRC 值，当 TRANS 置位时，读该寄存器将返回一个中间值。不同的数据帧格式（SPI_CTL0 中的 LF 位决定）将会得到不同的 CRC 值。

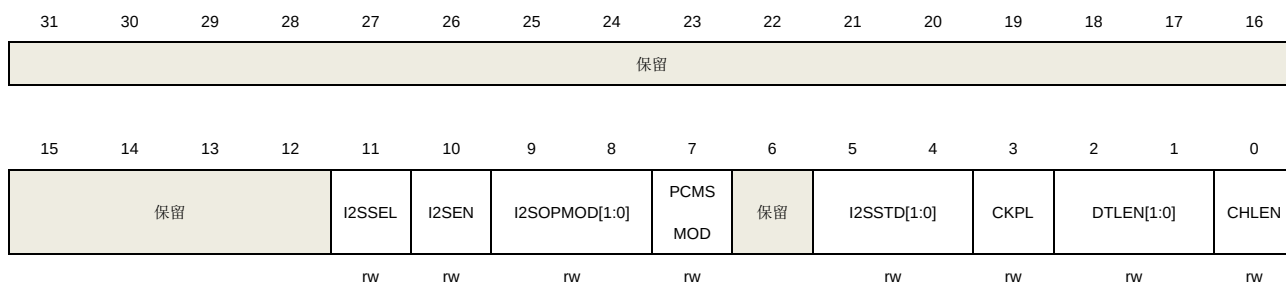
当 SPI_CTL0 寄存器中的 CRCEN 位和 SPIEN 位清零时，该寄存器复位。

24.5.8. I2S 控制寄存器 (SPI_I2SCTL)

地址偏移：0x1C

复位值：0x0000 0000

该寄存器可以按字（32 位）访问。



位/位域	名称	描述
31:12	保留	必须保持复位值。
11	I2SSEL	I2S 模式选择 0: SPI 模式 1: I2S 模式 当 SPI 模式或 I2S 模式关闭时配置该位。
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 模式关闭时配置该位。SPI 模式不使用该位。
7	PCMSMOD	PCM 帧同步模式

		0: 短帧同步 1: 长帧同步 只有在 PCM 标准下，该位才有意义。 当 I2S 模式关闭时配置该位。 SPI 模式不使用该位。
6	保留	必须保持复位值。
5:4	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 模式关闭时配置该位。 SPI 模式不使用该位。
3	CKPL	空闲状态时钟极性 0: I2S_CK 空闲状态为低电平 1: I2S_CK 空闲状态为高电平 当 I2S 模式关闭时配置该位。 SPI 模式不使用该位。
2:1	DTLEN[1:0]	数据长度 00: 16 位 01: 24 位 10: 32 位 11: 保留 当 I2S 模式关闭时配置该位。 SPI 模式不使用该位。
0	CHLEN	通道长度 0: 16 位 1: 32 位 通道长度必须大于或等于数据长度。 当 I2S 模式关闭时配置该位。 SPI 模式不使用该位。

24.5.9. I2S 时钟预分频寄存器 (SPI_I2SPSC)

地址偏移: 0x20

复位值: 0x0000 0002

该寄存器可以按字（32 位）访问。



位/位域	名称	描述
31:10	保留	必须保持复位值。
9	MCKOEN	I2S_MCK 输出使能 0: I2S_MCK 输出禁止 1: I2S_MCK 输出使能 当 I2S 模式关闭时配置该位。 SPI 模式不使用该位。
8	OF	预分频器的奇系数 0: 实际分频系数为 $DIV * 2$ 1: 实际分频系数为 $DIV * 2 + 1$ 当 I2S 模式关闭时配置该位。SPI 模式下不使用该位。
7:0	DIV[7:0]	预分频器的分频系数 实际分频系数是 $DIV * 2 + OF$ 。 DIV 不能为 0。 当 I2S 模式关闭时配置该位。SPI 模式下不使用该位。

24.5.10. SPI0 四路 SPI 控制寄存器 (SPI_QCTL)

地址偏移: 0x80

复位值: 0x0000 0000

该寄存器可以按字（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														QRD	QMOD
														rw	rw

位/位域	名称	描述
31:2	保留	必须保持复位值。
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。

25. 内部集成电路总线接口（I2C）

25.1. 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（Fast mode plus, Fm+），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，I2C 接口模块还支持多主机 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。

25.2. 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。
- 多个 7 位从机地址（两个地址，其中一个可配置地址位屏蔽）。
- 可编程的建立时间和保持时间。
- 支持 I2C 多主机模式。
- 支持标速（最高 100 kHz），快速（最高 400 kHz）和 Fm+模式（最高 1MHz，必须在 SYSCFG_CFG1 中使能 Fm+）。
- 从机模式下可配置的 SCL 主动拉低。
- 支持 DMA 模式。
- 兼容 SMBus 3.0 和 PMBus 1.3。
- 可选择的 PEC（报文错误校验）生成和校验。
- 可编程模拟过滤器和数字过滤器。

25.3. 功能说明

I2C 接口的内部结构如[图 25-1. I2C 模块框图](#)所示。

图 25-1. I2C 模块框图

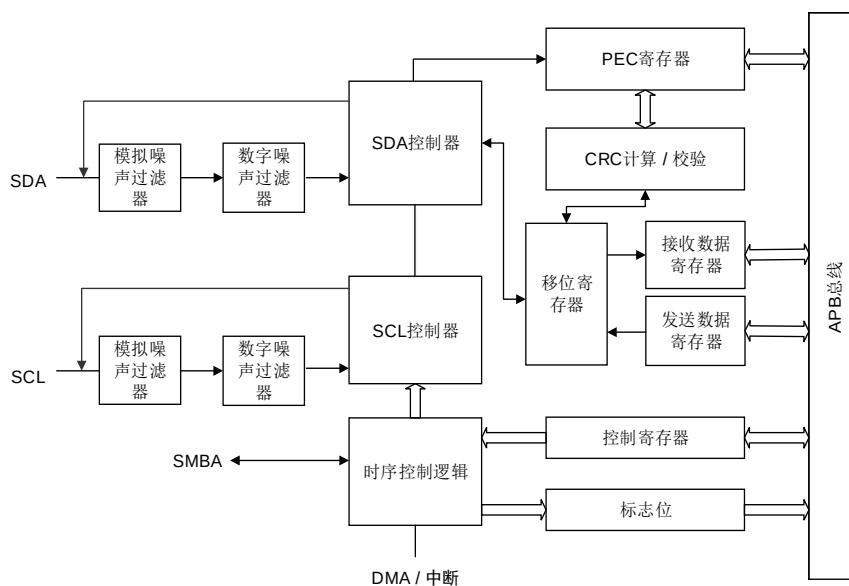


表 25-1. I2C 总线术语说明（参考飞利浦 I2C 规范）

术语	说明
发送器	发送数据到总线的设备
接收器	从总线接收数据的设备
主机	初始化数据传输，产生时钟信号和结束数据传输的设备
从机	由主机寻址的设备
多主	不破坏信息的前提下同时控制总线的多个主机
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

25.3.1. 时钟要求

I2C 时钟周期 t_{I2CCLK} 必须满足以下条件：

- $t_{I2CCLK} < (t_{LOW} - t_{filters}) / 4$
- $t_{I2CCLK} < t_{HIGH}$

其中：

t_{LOW} ：SCL 低电平时间

t_{HIGH} ：SCL 高电平时间

$t_{filters}$ ：在使能滤波器时，表示模拟滤波器和数字滤波器产生的延时总和。模拟滤波器产生的延时最大值为 126ns，数字滤波器产生的延时为 $DNF[3:0] \times t_{I2CCLK}$ 。

PCLK 时钟周期 t_{PCLK} 必须满足以下条件:

- $t_{PCLK} < 4/3 * t_{SCL}$

其中:

t_{SCL} : SCL 周期

注意: 当 I2C 内核时钟由 PCLK 提供时, PCLK 必须符合 t_{I2CCLK} 的条件。

25.3.2. I2C 通讯流程

主机和从机都能实现数据收发, 因此, I2C 可以实现四种工作模式:

- 从机发送
- 从机接收
- 主机发送
- 主机接收

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 25-2. 数据有效性](#))。每个数据比特传输需要一个时钟脉冲。

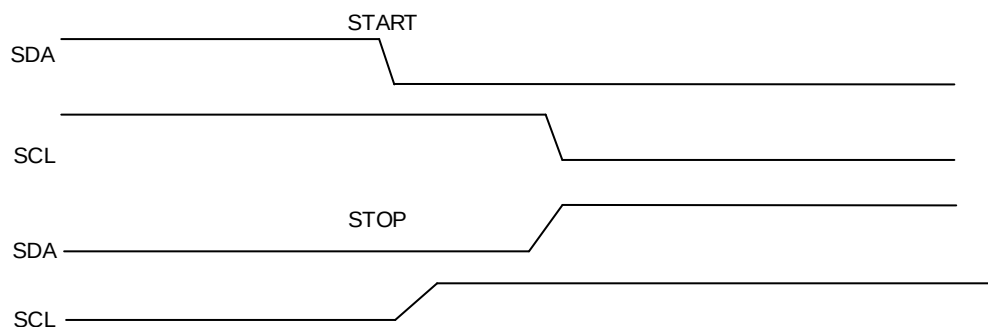
图 25-2. 数据有效性



开始和停止信号

所有的数据传输起始于一个 START 结束于一个 STOP (参见 [图 25-3. 开始和停止信号](#))。START 信号定义为, 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 25-3. 开始和停止信号



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。在默认情况下，I2C 设备工作在从机模式下。当 START 信号产生时，I2C 设备由从机模式切换成主机模式。如果仲裁丢失或者 STOP 信号产生时，I2C 由主机模式切换成从机模式。支持 I2C 多主机模式。

I2C 从机检测到 I2C 总线上的 START 信号之后，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，当两个地址相同时，I2C 从机将发送一个确认应答（ACK），并响应总线的后续命令：发送或接收所需数据。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 模块支持 7 位和 10 位的地址模式。

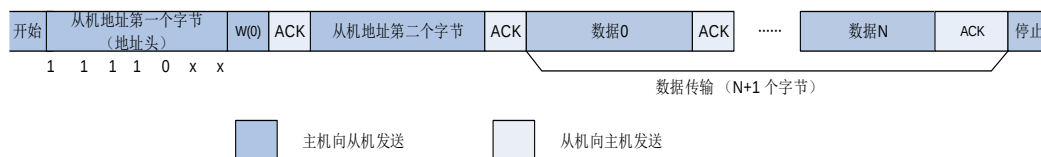
数据和地址都是 8 位传输，高位在前。START 信号之后的字节（在 7 位地址模式下是一个字节，10 位地址模式下是两个字节）是主机发送的从机地址。

8 个时钟周期字节发送后，第 9 个时钟脉冲期间接收器会发送应答信号至发送器。是否产生 ACK 信号可以软件配置。

I2C 主机负责产生 START 信号和 STOP 信号来开始和结束一次传输，并且负责产生 SCL 时钟。

在主机模式下，如果 AUTOEND=1，STOP 信号由硬件产生。如果 AUTOEND=0，STOP 信号由软件产生，或者主机可以产生 RESTART 信号来启动新的数据传输。

图 25-4. 10 位地址的 I2C 通讯流程（主机发送）



注意：在 10 位地址主机发送场景下，一个数据包发送完毕后，主机必须先发送一个停止信号，然后再发送一个开始信号来发起一个新的数据包的传输。

图 25-5. 7 位地址的 I2C 通讯流程（主机发送）

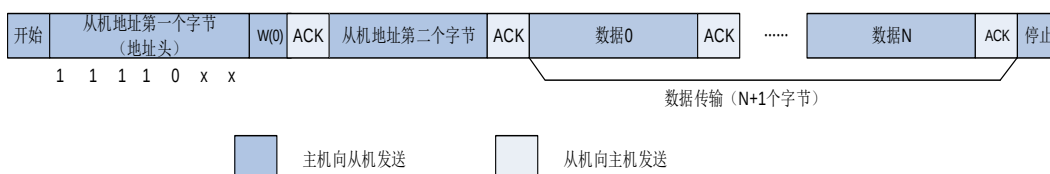
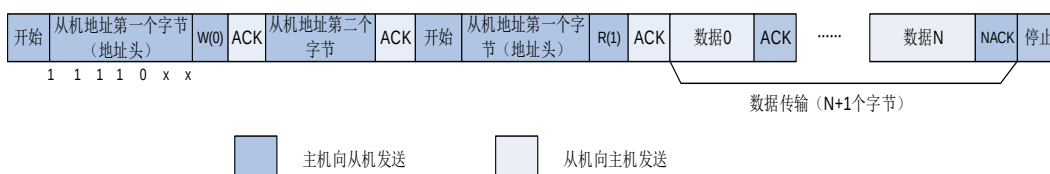


图 25-6. 7 位地址的 I2C 通讯流程（主机接收）



在 10 位寻址模式中，配置 HEAD10R 位可以选择执行完整的寻址序列或只发送地址头。当 HEAD10R=0，执行完整的 10 位地址寻址读序列 START+10 位地址头（写）+第二个地址字节+RESTART+10 位地址头（读），如[图 25-7. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R=0）](#)所示。

在 10 位寻址模式中，当 HEAD10R=1 时，I2C 传输序列限制为主机发送数据完毕后紧接着主机接收数据。该传输序列如[图 25-8. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R=1）](#)所示。

图 25-7. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R=0）

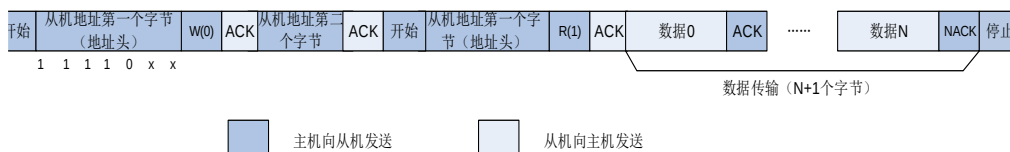
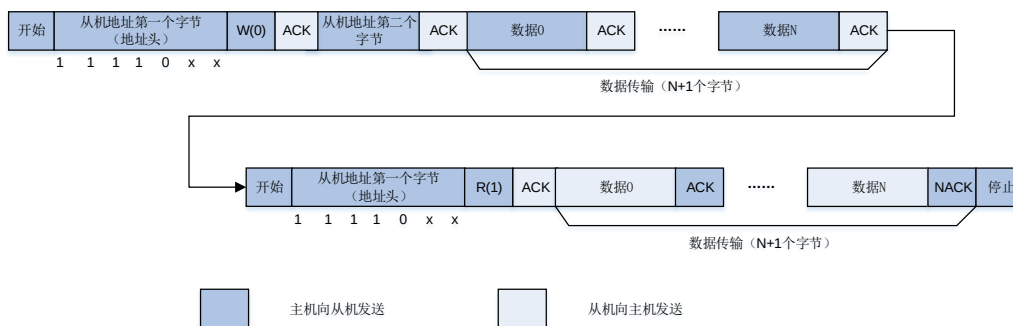


图 25-8. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R=1）



25.3.3. 噪声滤波器

I2C 外设集成了模拟噪声滤波器和数字噪声滤波器，噪声滤波器可根据实际需要在 I2C 外设启用前进行配置。

将 I2C_CTL0 寄存器中 ANOFF 位置 1 可以禁用模拟噪声滤波器，将 ANOFF 位清 0 时使能模拟

噪声滤波器。在快速模式和快速+ 模式下，模拟滤波器可以抑制脉冲宽度小于 50ns 的噪声。

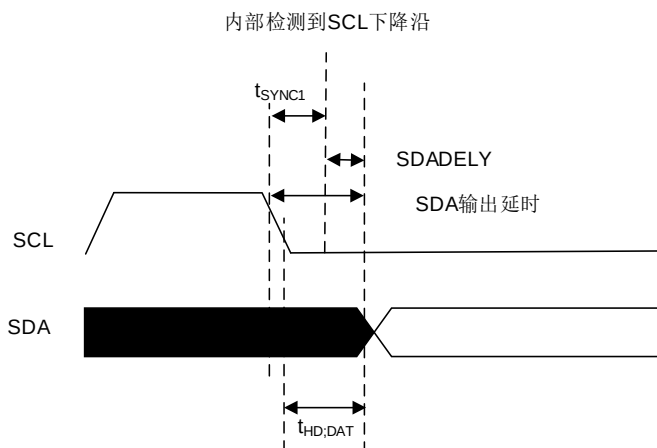
数字滤波器由 I2C_CTL0 寄存器中 DNF[3:0]位来配置。当数字滤波器使能时，SCL 和 SDA 电平保持稳定的时间大于 $DNF[3:0] \times t_{I2CCLK}$ 才会发生内部变化。抑制峰值宽度可由 DNF[3:0]配置。

25.3.4. I2C 时序配置

在 I2C 通信中，I2C_TIMING 寄存器中 PSC[3:0]，SCLDELY[3:0]和 SDADELY[3:0]用于保证正确的数据保持时间和数据建立时间。

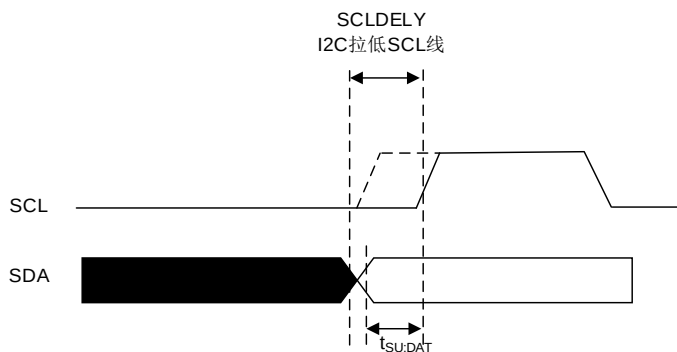
如果数据已经在 I2C_TDATA 寄存器中，在经历 SDADELY 延时后，数据由 SDA 发送，如 [图 25-9. 数据保持时间](#)所示。

图 25-9. 数据保持时间



当数据经过 SDA 发送时，SCLDELY 计数器开启。如图 [图 25-10. 数据建立时间](#)所示。

图 25-10. 数据建立时间



当内部检测到 SCL 下降沿时，在 SDA 发送之前会插入一个延时。该延时为 $t_{SDADELY} = SDADELY \times t_{PSC} + t_{I2CCLK}$ ，其中 $t_{PSC} = (PSC + 1) \times t_{I2CCLK}$ 。 $t_{SDADELY}$ 会影响 $t_{HD,DAT}$ 。 SDA 输出总延时为 $t_{SYNC1} + \{[SDADELY \times (PSC + 1) + 1] \times t_{I2CCLK}\}$ 。 t_{SYNC1} 由 SCL 下降斜率，模拟滤波器延时，

数字滤波器延时和 SCL 与 I2CCLK 时钟的同步延时共同决定。SCL 与 I2CCLK 时钟的同步延时为 2 至 3 个 t_{I2CCLK} 。

注意：当 SDADELY=0 时， $t_{SDADELY}=t_{PSC}$ 。

SDADELY 必须符合以下条件：

- $SDADELY \geq \{t_r(\max) + t_{HD;DAT}(\min) - t_{AF}(\min) - [(DNF+3) \cdot t_{I2CCLK}]\} / [(PSC+1) \cdot t_{I2CCLK}]$
- $SDADELY \leq \{t_{HD;DAT}(\max) - t_{AF}(\max) - [(DNF+4) \cdot t_{I2CCLK}]\} / [(PSC+1) \cdot t_{I2CCLK}]$

注意： t_{AF} 为模拟滤波器延时， $t_{HD;DAT}$ 必须小于 $t_{VD;DAT}$ 的最大值。

当 SS=0 时，经过延时 $t_{SDADELY}$ ，在数据写入 I2C_TDATA 寄存器之前，从机会拉低时钟线。在数据建立时间期间 SCL 保持低电平。数据建立时间 $t_{SCLDELY} = (SCLDELY+1) \cdot t_{PSC}$ 。 $t_{SCLDELY}$ 影响 $t_{SU;DAT}$ 。

SCLDELY 必须符合以下条件：

- $SCLDELY \geq \{[t_r(\max) + t_{SU;DAT}(\min)] / [(PSC+1) \cdot t_{I2CCLK}]\} - 1$

在主机模式下，SCL 时钟高低电平由 I2C_TIMING 寄存器中 PSC[3:0]，SCLH[7:0]和 SCLL[7:0]控制。

当内部检测到 SCL 下降沿，在释放 SCL 输出之前会插入一个延时，该延时为 $t_{SCLL} = (SCLL+1) \cdot t_{PSC}$ ，其中 $t_{PSC} = (PSC+1) \cdot t_{I2CCLK}$ 。 t_{SCLL} 影响 SCL 低电平持续时间 t_{LOW} 。

当内部检测到 SCL 上升沿，在将 SCL 拉低之前会插入一个延时，该延时为 $t_{SCLH} = (SCLH+1) \cdot t_{PSC}$ ，其中 $t_{PSC} = (PSC+1) \cdot t_{I2CCLK}$ 。 t_{SCLH} 影响 SCL 高电平持续时间 t_{HIGH} 。

注意：时序配置和 SS 位在 I2C 外设使能时是不能改变的。

表 25-2. 数据建立时间和数据保持时间

符号	参数	标准模式		快速模式		快速 + 模式		SMBus		单位
		最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值	
$t_{HD;DAT}$	数据保持时间	0	-	0	-	0	-	0.3	-	us
$t_{VD;DAT}$	数据有效时间	-	3.45	-	0.9	-	0.45	-	-	
$t_{SU;DAT}$	数据建立时间	250	-	100	-	50	-	250	-	ns
t_r	SCL 和 SDA 信号上升时间	-	1000	-	300	-	120	-	1000	
t_f	SCL 和 SDA 信号下降时间	-	300	-	300	-	120	-	300	

25.3.5. I2C 复位

清除 I2C_CTL0 寄存器中 I2CEN 位可以实现软件复位。当软件复位产生时，SCL 和 SDA 均被释放。通信控制位和状态位也还原成复位值。软件复位对配置寄存器无影响。受到影响的位为 I2C_CTL1 寄存器中 START, STOP 和 NACKEN, I2C_STAT 寄存器中 I2CBSY, TBE, TI, RBNE, ADDSEND, NACK, TCR, TC, STPDET, BERR, LOSTARB 和 OUERR。另外，如果支持 SMBus

模式, I2C_CTL1 寄存器中 PECTRANS 位, I2C_STAT 寄存器中 PECERR, TIMEOUT 和 SMBALT 位也会受到影响。

为了实现软件复位, I2CEN 必须在至少 3 个 APB 时钟周期内保持低电平。可以通过以下写软件序列来保证软件复位:

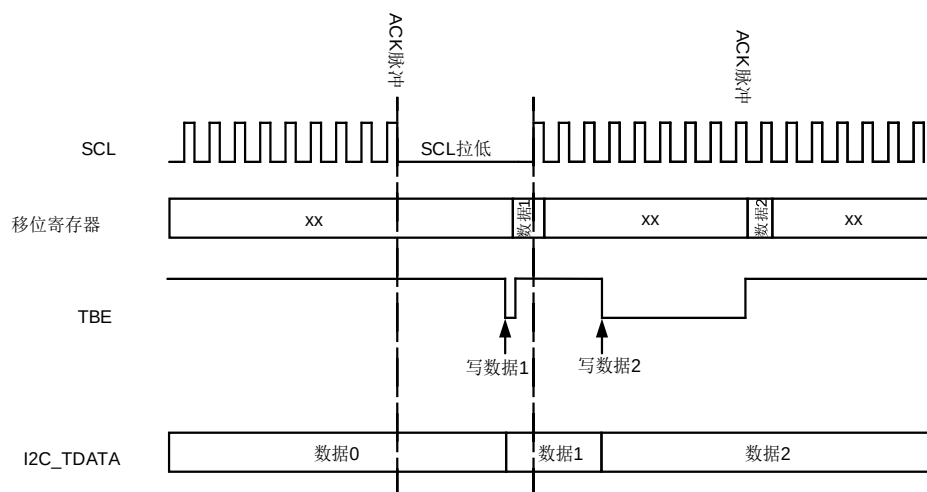
- I2CEN 写 0
- 检查 I2CEN 是否为 0
- I2CEN 写 1

25.3.6. 数据传输

数据发送

在发送数据时, 如果 TBE 为 0, 表明 I2C_TDATA 寄存器非空, 在第九个 SCL 脉冲(应答脉冲)后, I2C_TDATA 寄存器中的数据移入到移位寄存器。移位寄存器中的数据通过 SDA 线移出。如果 TBE 为 1, 则表明 I2C_TDATA 寄存器为空, 在 I2C_TDATA 不为空之前 SCL 将被拉低。SCL 拉低是在第九个 SCL 脉冲之后。

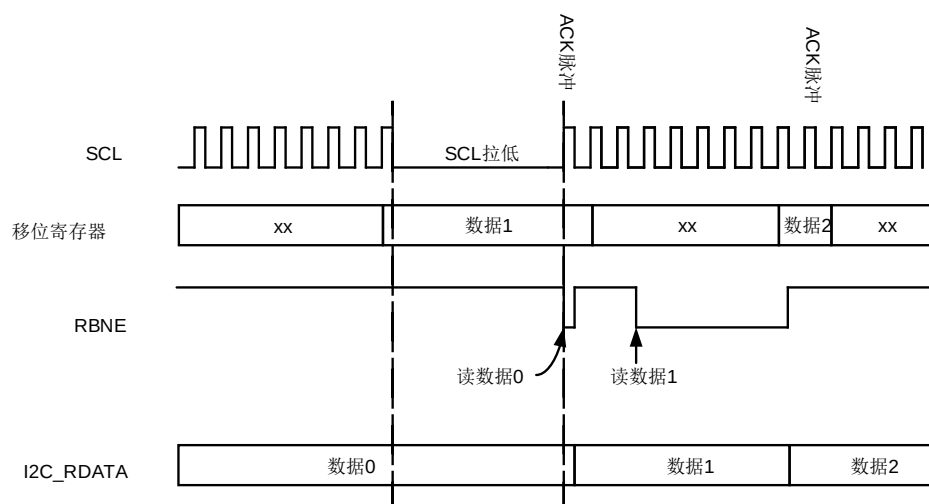
图 25-11. 数据发送



数据接收

在接收数据时, 数据首先被接收到移位寄存器。如果 RBNE 为 0, 移位寄存器中的数据将被移入 I2C_RDATA 寄存器。如果 RBNE 为 1, SCL 时钟将被拉低, 直到之前接收到的数据字节被读取。这个时钟拉低被插入应答脉冲之前。

图 25-12. 数据接收



重载和自动结束模式

为了管理字节传输和中断如[表 25-3. 可关闭通信模式](#)所示几种通信模式，I2C 硬件嵌入了字节计数器。

表 25-3. 可关闭通信模式

工作模式	行为
主机模式	产生 NACK, STOP 和 RESTART
从机接收模式	ACK 控制
SMBus 模式	PEC 生成/校验

传输的字节数由 BYTENUM[7:0]在 I2C_CTL1 寄存器中配置。如果 BYTENUM 大于 255，或者处于从机字节控制模式，则必须通过将 I2C_CTL1 寄存器中 RELOAD 位置 1 来使能重载模式。在重载模式下，当 BYTENUM 计数到 0 时，TCR 位将置 1，如果 TCIE 位置 1 将产生中断。当 TCR 位置 1 时，SCL 将被拉低。在 BYTENUM 写一个非零值将清除 TCR 位。

注意：重载模式必须在 BYTENUM[7:0]最后一次重载后禁用。

当使能自动结束模式时，必须禁用重载模式。在自动结束模式下，当 BYTENUM[7:0]计数到 0 时，主机将自动发送一个 STOP 信号。

当重载模式和自动结束模式都被禁用时，I2C 通信进程需要由软件终止。如果 BYTENUM[7:0]中的字节数已经传输完成，软件应将 STOP 位置 1 来产生一个 STOP 信号，然后清除 TC。

25.3.7. I2C 从机模式

初始化

从机模式下，至少使能一个从机地址。第一个从机地址写在 I2C_SADDR0 寄存器中，第二个从机地址写在 I2C_SADDR1 寄存器中。在使用从机地址时，必须相应地将 I2C_SADDR0 寄存器中 ADDRESSEN 位和 I2C_SADDR1 寄存器中 ADDRESS2EN 置 1。通过设置 I2C_SADDR0 寄存器中 ADDFORMAT 位可以选择 7 位地址或 10 位地址，该地址被写在 ADDRESS[9:0]。

I2C_CTL2 寄存器中 ADDM[6:0]定义 ADDRESS[7:1]的哪些位和接收到的地址进行比较，哪些位不比较。

ADDMSK2[2:0]用于屏蔽 I2C_SADDR1 寄存器中 ADDRESS2[7:1]，相关详细信息参考 I2C_SADDR1 寄存器 ADDMSK2[2:0]位域描述。

当 I2C 接收到的地址与使能的地址其中一个匹配成功时，ADDSEND 将被置 1，如果 ADDMIE 置位，将产生中断。I2C_STAT 寄存器 READDR[6:0]将会存储接收到的地址。在 ADDSEND 置位时，I2C_STAT 寄存器中 TR 位状态更新。TR 的状态指示从机是作为发送器还是接收器。

SCL 线控制

当 SS=0 时，时钟拉低功能默认用在从机模式下，在需要的时候 SCL 会被拉低。在下列情况下，SCL 会被拉低。

- 当 ADDSEND 置位时 SCL 线拉低，并在 ADDSEND 位清零之后释放。
- 在从机发送模式下，ADDSEND 清零之后，SCL 在第一个字节写入 I2C_TDATA 寄存器之前都是被拉低的。在前一个字节发送完成之后，新的字节写入 I2C_TDATA 寄存器之前，SCL 也是被拉低的。
- 在从机接收模式下，接收过程已完成但是 I2C_RDATA 寄存器中的数据还未被读取，SCL 将被拉低。
- 当 SBCTL=1 且 RELOAD=1 时，在最后一个字节传输结束后，TCR 置位。在 TCR 清除之前 SCL 将被拉低。
- SCL 下降沿被检测到之后，在 $[(SDELY+SCLDELY+1)*(PSC+1)+1]*t_{I2CCLK}$ 期间 SCL 被拉低。

SCL 线控制可以通过将 I2C_CTL0 寄存器中 SS 位置 1 来禁能。在下列情况下，SCL 不会被拉低。

- 在 ADDSEND 置位时 SCL 将不会被拉低。
- 在从机发送模式下，数据必须在它传输过程产生的第一个 SCL 脉冲之前写入 I2C_TDATA 寄存器。否则 I2C_STAT 寄存器中 OUERR 位将会置 1，如果 ERRIE 位也被置 1，将产生一个中断。当 STPDET 位置 1 并且第一个数据开始发送，I2C_STAT 寄存器中 OUERR 位也将置 1。
- 在从机接收模式下，数据必须在下一个字节接收产生的第九个 SCL 脉冲（ACK 脉冲）之前读取。否则 I2C_STAT 寄存器中 OUERR 位也将置 1。如果 ERRIE 位也被置 1，将产生一个中

断。

从机字节控制模式

在从机接收模式下要实现字节 ACK 控制，可以通过将 I2C_CTL0 寄存器中 SBCTL 位置 1 来使能从机字节控制模式。当 SS=1 时，从机字节控制模式无效。

在使用从机字节控制模式时，必须通过置位 I2C_CTL1 寄存器中 RELOAD 位来使能重载模式。从机字节控制模式中，在 ADDSEND 中断服务程序中 I2C_CTL1 寄存器中 BYTENUM[7:0] 必须配置为 1，并且在每个字节接收完成时重载为 1。当接收到一个字节时，I2C_STAT 寄存器中 TCR 位置 1，在第八个和第九个 SCL 时钟脉冲之间从机将 SCL 时钟拉低。然后数据可以从 I2C_RDATA 寄存器中读取出来，通过配置 I2C_CTL1 寄存器中 NACKEN 位，从机可以决定发送 ACK 或者是 NACK。当在 BYTENUM[7:0] 写入非零值时，从机释放 SCL 时钟线。

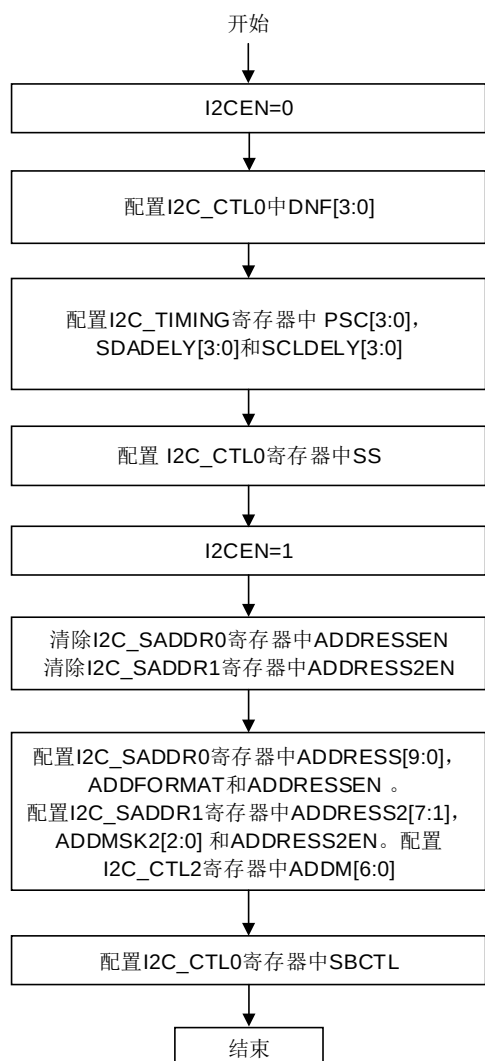
当 BYTENUM[7:0] 大于 0x1 时，在 BYTENUM[7:0] 数据接收期间，数据流是连续的。

注意：在下列情况下，可以配置 SBCTL 位：

- I2CEN=0。
- 从机还未被寻址。
- ADDSEND=1。

当 ADDSEND=1，或者 TCR=1 时，RELOAD 才可以被修改。

图 25-13. I2C 从机初始化



从机发送模式下的软件流程

当 I2C_TDATA 寄存器为空, I2C_STAT 寄存器中 TI 位将会置位。如果 I2C_CTL0 寄存器中 TIE 位置 1, 将产生中断。当接收到 NACK 时, I2C_STAT 寄存器中 NACK 位会置位。如果 I2C_CTL0 寄存器中 NACKIE 位置 1, 将产生中断。当接收到 NACK 信号时, I2C_STAT 寄存器中 TI 位将不会置位。

当接收到 STOP 信号时, I2C_STAT 寄存器中 STPDET 位将置 1。如果 I2C_CTL0 寄存器中 STPDETIE 位置 1, 将产生中断。

当 SBCTL=0 时, 如果 ADDSEND=1, 且 I2C_STAT 寄存器中 TBE 位为 0, 可以选择发送 I2C_TDATA 寄存器中的数据或者是将 TBE 置 1 来清空 I2C_TDATA 寄存器。

当 SBCTL=1 时, 从机工作在字节控制模式, BYTENUM[7:0]必须在 ADDSEND 中断服务程序中配置。TI 事件的数量与 BYTENUM[7:0]的值相等。

当 SS=1 时，I2C_STAT 寄存器中 ADDSEND 位置位时 SCL 时钟线不会被拉低。在这种情况下，I2C_TDATA 寄存器中数据不能在 ADDSEND 中断服务程序中清空。因此待发送的第一个字节应该在 ADDSEND 置位之前就被编程到 I2C_TDATA 寄存器。

- 该数据可以是上一次数据传输最后一次 TI 事件写入的数据。
- 如果该数据不是待发送数据，可通过将 TBE 位置 1 来刷新 I2C_TDATA 寄存器，从而编程新的数据。在数据发送开始时 STPDET 位必须为 0。否则 I2C_STAT 寄存器中 OUERR 位将置 1 并产生下溢错误。
- 从机发送模式下使用中断或者 DMA 时，如果需要一个 TI 事件，TI 位和 TBE 位都必须置 1。

图 25-14. I2C 从机发送编程模型（SS=0）

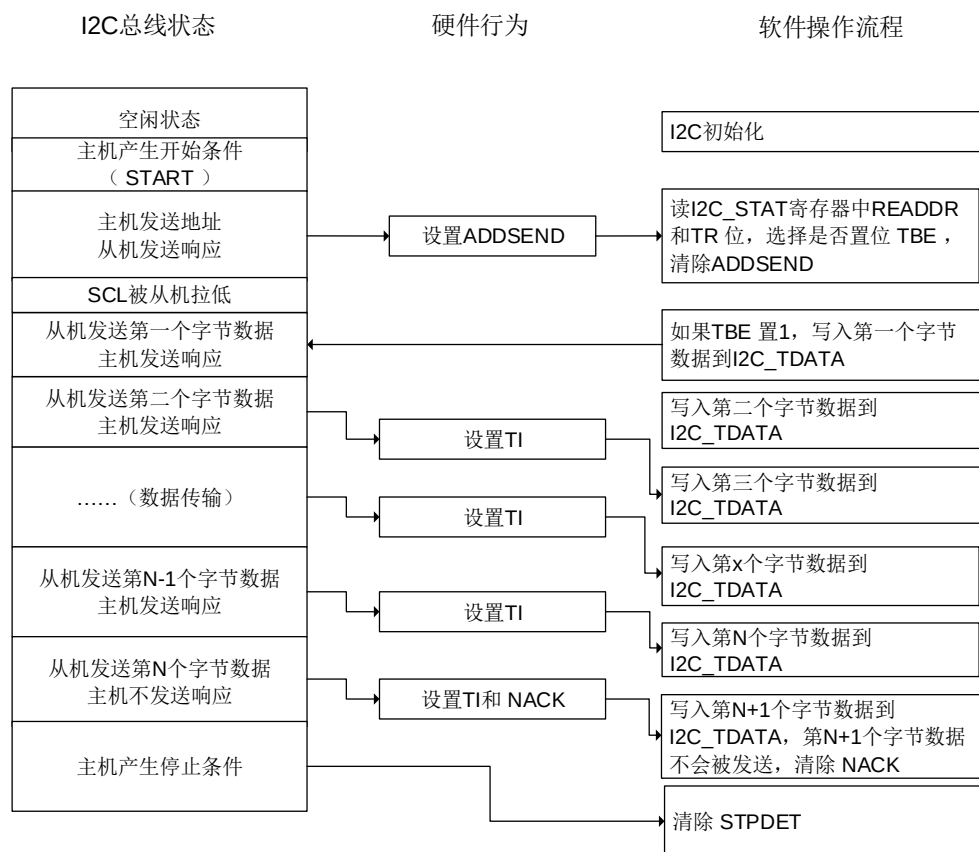
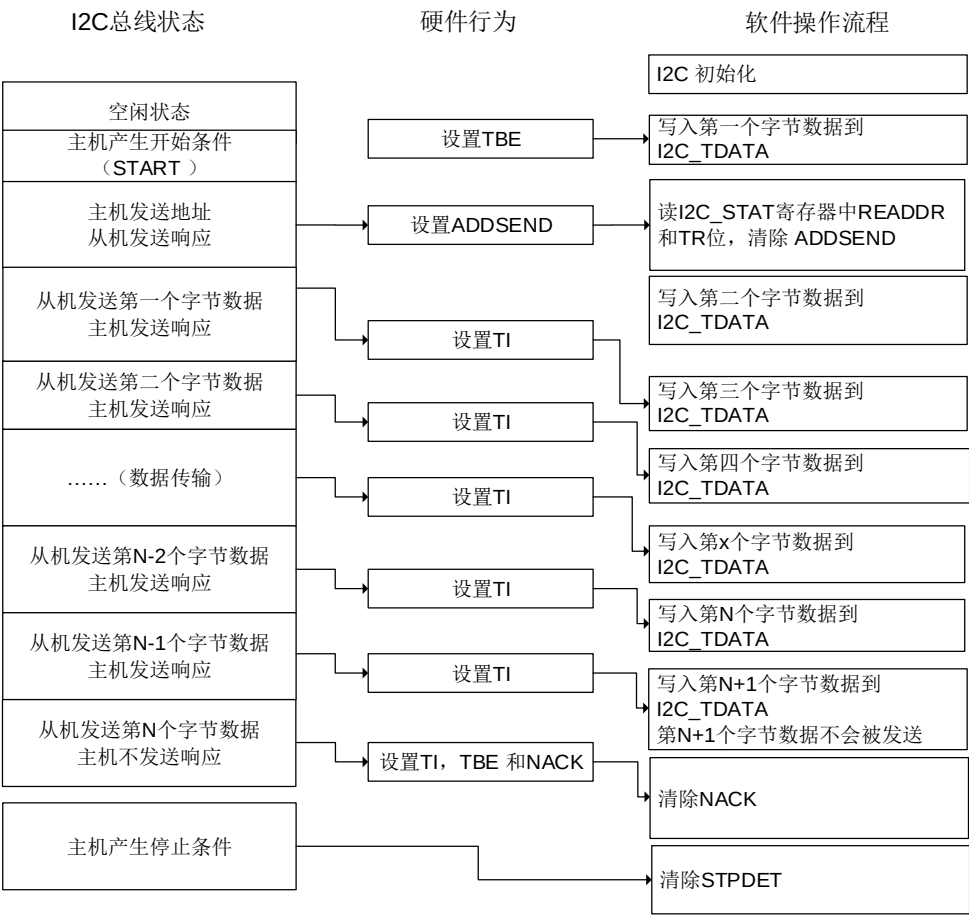


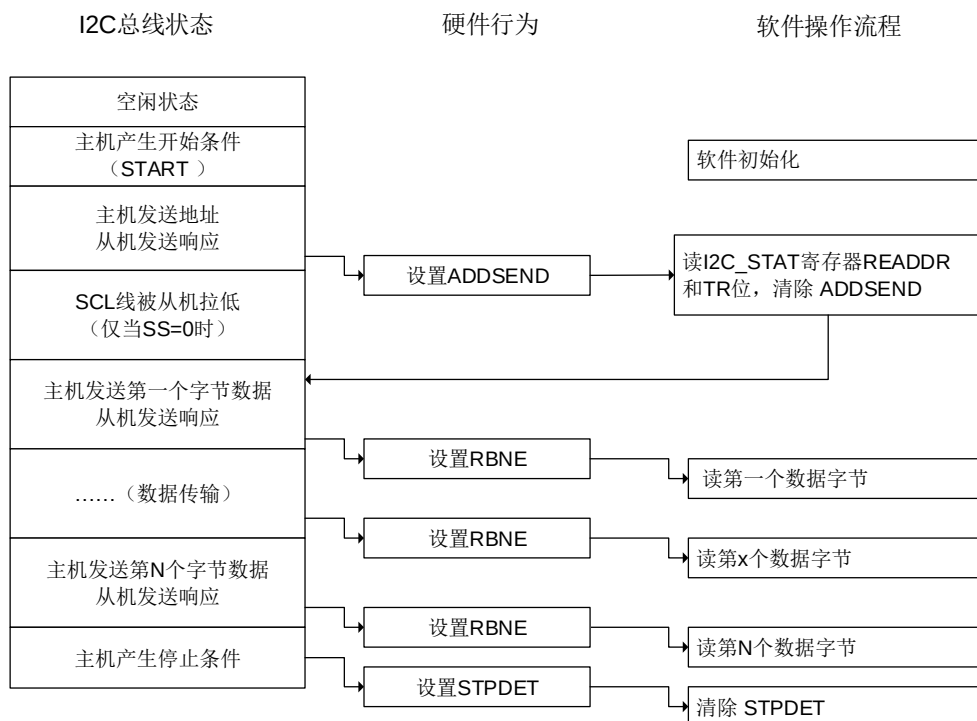
图 25-15. I2C 从机发送编程模型（SS=1）



从机接收模式下的软件流程

当 I2C_RDATA 寄存器非空, I2C_STAT 寄存器中 RBNE 位置 1, 如果 I2C_CTL0 寄存器中 RBNEIE 位置 1, 将产生中断。当接收到 STOP 信号时, I2C_STAT 寄存器中 STPDET 位将置 1。如果 I2C_CTL0 寄存器中 STPDETIE 置 1, 将产生中断。

图 25-16. I2C 从机接收编程模型



25.3.8. I2C 主机模式

初始化

I2C_TIMING 寄存器中 SCLH[7:0]和 SCLL[7:0]必须在 I2CEN=0 时配置。为了支持多主机通信和从机时钟拉低, I2C 实现了时钟同步机制。

SCLL[7:0]和 SCLH[7:0]分别用于低电平计数和高电平计数。经过 t_{SYNC1} 延时后, 当检测到 SCL 低电平时, SCLL[7:0]开始计数, 如果 SCLL[7:0]计数器的值达到 I2C_TIMING 寄存器中 SCLL[7:0]时, I2C 将释放 SCL 时钟。经过 t_{SYNC2} 延时后, 当检测到 SCL 高电平时, SCLH[7:0]开始计数, 如果 SCLH[7:0]计数器的值达到 I2C_TIMING 寄存器中 SCLH[7:0]时, I2C 将拉低 SCL 时钟。

因此主机时钟周期为: $t_{\text{SCL}} = t_{\text{SYNC1}} + t_{\text{SYNC2}} + \{[(\text{SCLH}[7:0] + 1) + (\text{SCLL}[7:0] + 1)] * (\text{PSC} + 1) * t_{\text{I2CCLK}}\}$ 。

t_{SYNC1} 取决于 SCL 下降沿斜率, SCL 输入模拟和数字噪声滤波器延时以及 SCL 与 I2CCLK 时钟的同步产生的延时, 一般为 2 到 3 个 I2CCLK 时钟周期。 t_{SYNC2} 取决于 SCL 上升沿斜率, SCL 输入模拟和数字噪声滤波器延时以及 SCL 与 I2CCLK 时钟的同步产生的延时, 一般为 2 到 3 个 I2CCLK 时钟周期。数字噪声滤波器产生的延时为 $\text{DNF}[3:0] * t_{\text{I2CCLK}}$ 。

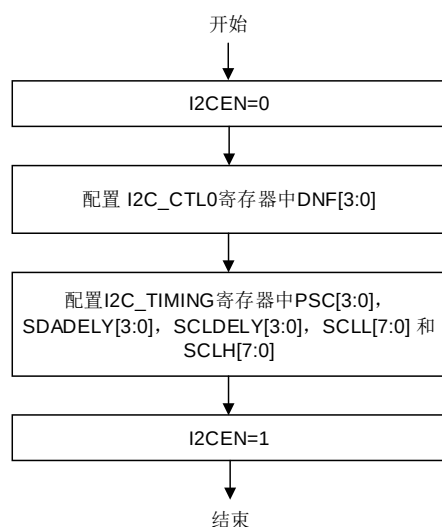
在主机模式下, 必须配置 I2C_CTL1 寄存器中 ADD10EN, SADDRESS[9:0]以及 TRDIR 位。当在主机接收模式下使用 10 位寻址时, 必须配置 HEAD10R 来选择是执行完整的地址寻址序列, 还是只发送地址头。待传输的字节数在 I2C_CTL1 寄存器 BYTENUM[7:0]配置。如果待传输的字节数大于或者等于 255, 必须将 BYTENUM[7:0]配置为 0xFF。然后主机发送 START 信号。以上提到

的所有位必须在 **START** 位置 1 之前配置。**START** 信号发送完成之后,待 **I2C_STAT** 寄存器 **I2CBSY** 位为 0 时,发送从机地址。当仲裁丢失时,主机切换成从机模式,**START** 位由硬件清零。当从机地址发送完成时,**START** 位由硬件清零。

在 10 位寻址模式下,在发送 10 位地址头之后,如果主机接收到 **NACK**,主机将重发 10 位地址头直到收到 **ACK**。将 **ADDSEND** 置 1 可以停止重发从机地址。

如果 **START** 位置 1 时, **I2C** 作为从机被寻址成功, **ADDSEND** 置 1, 主机将切换为从机模式。**START** 位将在 **ADDSEND** 置 1 时清零。

图 25-17. I2C 主机初始化



主机发送模式下的软件流程

在主机发送模式下,每一个字节发送完成并接收到 **ACK** 信号之后, **TI** 位将置 1。如果 **I2C_CTL0** 寄存器中 **TIE** 位置 1, 将产生中断。待发送的字节数编程在 **I2C_CTL1** 寄存器 **BYTENUM[7:0]**。如果发送字节数大于 255, 必须通过将 **I2C_CTL1** 寄存器 **RELOAD** 位置 1 来使能重载模式。在重载模式下, 当 **BYTENUM[7:0]** 个字节传输完成, **I2C_STAT** 寄存器 **TCR** 位将置 1, 并且在 **BYTENUM[7:0]** 更新一个非零值之前, **SCL** 被拉低。

如果接收到 **NACK**, **TI** 位将不会置 1。

- 如果 **BYTENUM[7:0]** 个字节传输完成且 **RELOAD=0**, 将 **I2C_CTL1** 寄存器中 **AUTOEND** 置 1 可以自动产生 **STOP** 信号。当 **AUTOEND=0** 时, **I2C_STAT** 寄存器 **TC** 位将置 1 且 **SCL** 被拉低。在这种情况下, 主机可以通过将 **I2C_CTL1** 寄存器中 **STOP** 位置 1 来产生 **STOP** 信号。或者产生 **RESTART** 信号来开始一个新的数据传输过程。将 **START / STOP** 置 1 可以清除 **TC** 位。
- 如果接收到 **NACK** 信号, **I2C** 将自动产生 **STOP** 信号。 **I2C_CTL0** 寄存器中 **NACK** 将置 1, 如果 **NACKIE** 位置 1, 将产生中断。

注意: 当 **RELOAD=1** 时, **AUTOEND** 位无效。

图 25-18. I2C 主机发送编程模型 (N<=255)

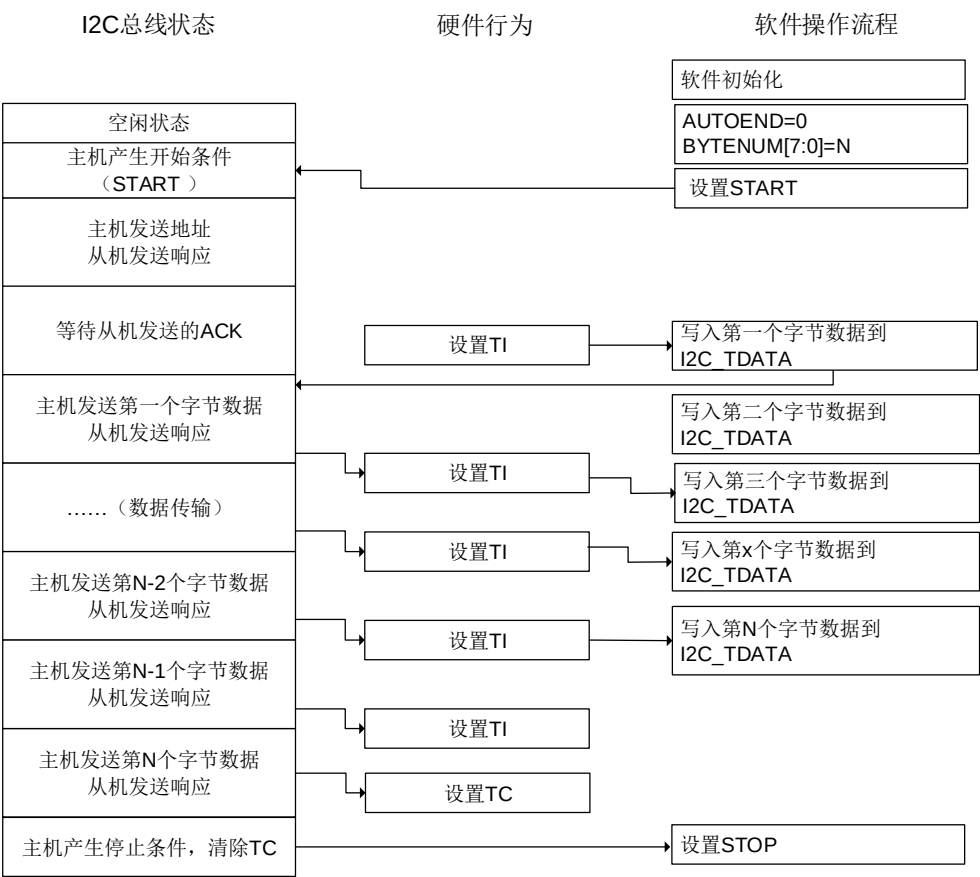


图 25-19. I2C 主机发送编程模型 (N>255)



主机接收模式下的软件流程

在主机接收模式下, 当接收到一个字节时, I2C_STAT 寄存器中 RBNE 位置 1。如果 I2C_CTL0 寄存器中 RBNEIE 置 1, 将产生一个中断。如果待接收字节数大于 255, 必须将 I2C_CTL1 寄存器中 RELOAD 位置 1 来使能重载模式。在重载模式下, 当 BYTENUM[7:0]个字节传输完成, I2C_STAT 寄存器中 TCR 位将置 1, 在 BYTENUM[7:0]中写入一个非零值之前, SCL 被拉低。

如果 BYTENUM[7:0]个字节传输完成且 RELOAD=0, 将 I2C_CTL1 寄存器中 AUTOEND 置 1 可以自动产生 STOP 信号。当 AUTOEND=0 时, I2C_STAT 寄存器 TC 位将置 1 且 SCL 被拉低。在这种情况下, 主机可以通过将 I2C_CTL1 寄存器中 STOP 位置 1 来产生 STOP 信号。或者产生 RESTART 信号来开始一个新的数据传输过程。将 START / STOP 置 1 可以清除 TC 位。

图 25-20. I2C 主机接收编程模型 (N<=255)

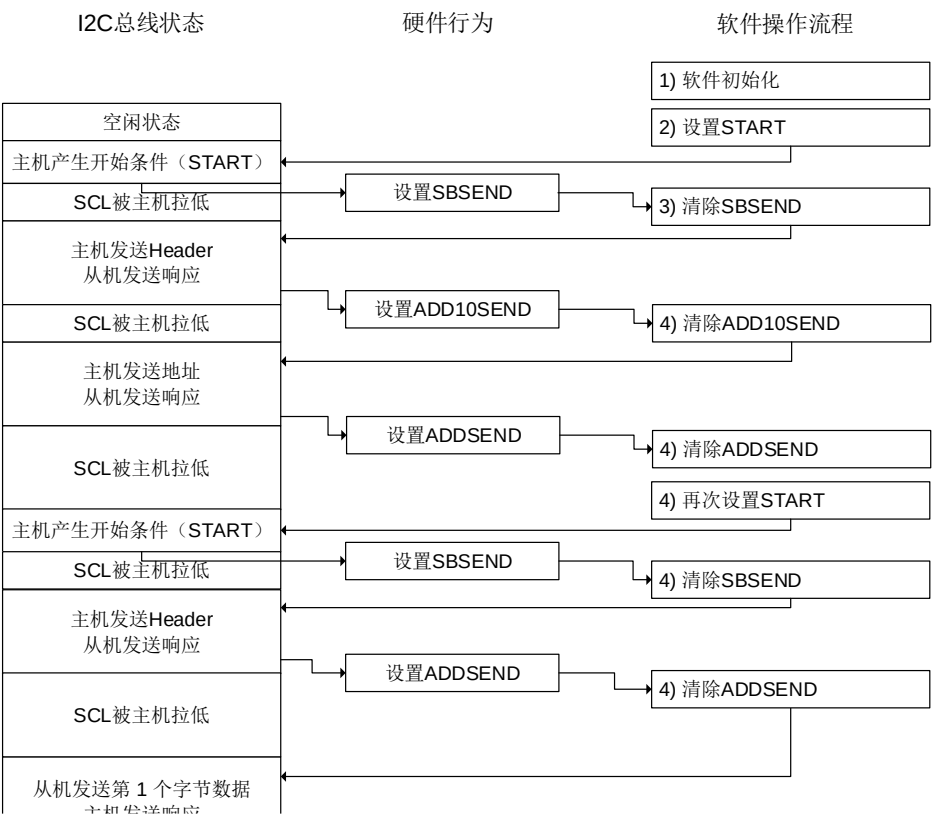


图 25-21. I2C 主机接收编程模型（N>255）



25.3.9. SMBus 支持

系统管理总线（System Management Bus，简称为 SMBus 或 SMB）是一种结构简单的单端双线制总线，可实现轻量级的通信需求。一般来说，SMBus 最常见于计算机主板，主要用于电源传输 ON/OFF 指令的通信。SMBus 是 I2C 的一种衍生总线形式，主要用于计算机主板上的低带宽设备间通信，尤其是与电源相关的芯片，例如笔记本电脑的可充电电池子系统（参见 Smart Battery Data）。

SMBus 协议

SMBus 上每个报文交互都遵从 SMBus 协议中预定义的格式。SMBus 是 I2C 规范中数据传输格式

的子集。只要 I2C 设备可通过 SMBus 协议之一进行访问，便视为兼容 SMBus 规范。不符合这些协议的 I2C 设备，将无法被 SMBus 和 ACPI 规范所定义的标准方法访问。

地址解析协议

SMBus 采用了 I2C 硬件以及 I2C 的硬件寻址方式，但在 I2C 的基础上增加了二级软件处理，建立自己独特的系统。比较特别的是 SMBus 规范包含一个地址解析协议，可用于实现动态地址分配。动态识别硬件和软件使得总线设备能够支持热插拔，无需重启系统便能即插即用。总线中的设备将被自动识别并分配唯一地址。这个优点非常有利于实现即插即用的用户界面。在此协议中，系统中的 host 与设备之间有一个重要的区别，即 host 具有分配地址的功能。

SMBus 从机字节控制

SMBus 接收器从机字节控制与 I2C 一样。它允许 ACK 控制每个字节。必须能对接收到的命令或者数据进行 NACK 应答。通过将 I2C_CTL0 寄存器中 SBCTL 位置 1 来使能从机字节控制模式。

主机通知协议

通过将 I2C_CTL0 寄存器 SMBHAEN 位置 1，SMBus 可以支持主机通知协议。在该协议中，从设备作为主机，主设备作为从机，主机将应答 SMBus 主机地址。

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之后，将继续 I2C 通信。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 25~35ms。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。

将 I2C_TIMEOUT 寄存器中 TOEN 位和 EXTTOEN 位置 1 可以使能超时检测。配置定时器必须保证在 SMBus 规范规定的时间最大值之前检测出超时情况。

在 BUSTOA[11:0]中编程的值被用来检查 t_{TIMEOUT} 参数。必须将 TOIDLE 位配置为 0，以检测 SCL 低电平超时。将 I2C_TIMEOUT 寄存器中 TOEN 位置 1 来使能定时器，在 TOEN 置 1 之后，BUSTOA[11:0]和 TOIDLE 位不能被修改。如果 SCL 低电平时间大于 $(\text{BUSTOA}+1)*2048*t_{\text{I2CCCLK}}$ ，I2C_STAT 寄存器中 TIMEOUT 位将置 1。

BUSTOA[11:0]为从机校验 $t_{\text{LOW:SEXT}}$ ，为主机校验 $t_{\text{LOW:MEXT}}$ 。通过将 I2C_TIMEOUT 寄存器中 EXTTOEN 位置 1 来使能定时器。在 EXTTOEN 置 1 之后，BUSTOB[11:0]不能被修改。如果 SMBus 外设 SCL 拉低时间大于 $(\text{BUSTOB}+1)*2048*t_{\text{I2CCCLK}}$ ，并且达到了总线空闲检测章节中描述的超时时间间隔，I2C_STAT 寄存器中 TIMEOUT 位将置 1。

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。一个 PEC 字节（PEC 错误码）附加在每次传输结束。PEC 的计算方式是对所有消息字节（包含地址和读/写位）使用 CRC-8 计算校验和。CRC-8 多项式位 x^8+x^2+x+1 （CRC-8-ATM HEC 算法，初始化为 0）。

当 I2C 被禁用时，通过 I2C_CTL0 寄存器中的 PECEN 位置 1 可以使能 PEC。由于 PEC 传输是由 I2C_CTL1 寄存器中 BYTENUM[7:0]管理的，因此在从机模式下必须将 SBCTL 位置 1。当 PECTRANS 置 1，RELOAD 为 0 时，在 BYTENUM[7:0]-1 数据字节后发送 PEC。PEC 在 BYTENUM[7:0]-1 传输完成后发送。当 RELOAD 置 1 时 PECTRANS 无效。

SMBus 警报

SMBus 还有一个额外的共享的中断信号，称为 SMBALERT#。从机上发生事件后，可通过这个信号通知主机来访问从机。主机会处理该中断，并通过报警响应地址，同时访问所有 SMBALERT# 设备。如果 SMBALERT# 电平被设备拉低，这些设备会应答报警响应地址。当配置为从设备（SMBHAEN=0）时，通过将 I2C_CTL0 寄存器中 SMBALTEN 置 1 可以将 SMBA 引脚电平拉低。同时也使能了报警响应地址。当配置为主设备（SMBHAEN=1），且 SMBALTEN 置 1 时，当在 SMBA 引脚检测到下降沿时，I2C_STAT 寄存器中 SMBALT 位将置 1。如果 I2C_CTL0 寄存器中 ERRIE 位置 1，将产生中断。当 SMBALTEN=0 时，即使外部 SMBA 引脚为低电平，ALERT 线也将被视为高电平。当 SMBALTEN=0 时，SMBA 引脚可用作标准 GPIO。

总线空闲检测

如果主机检测到时钟信号和数据信号的高电平持续时间大于 $t_{HIGH,MAX}$ ，总线被视为空闲。

该时序参数已考虑到主机已动态添加至总线，但可能还未检测到 SMBCLK 或 SMBDAT 线上的状态转换的情况。在这种情况下，为了保证当前没有数据传输正在进行，主机必须等待足够长的时间。

要使能 t_{IDLE} 检查，必须将 BUSTOA[11:0]编程为定时器重载值，以获取 t_{IDLE} 参数。必须将 TIDLE 位置 1，以检测 SCL 和 SDA 高电平超时。然后通过将 I2C_TIMEOUT 寄存器中的 TOEN 位置 1 来使能定时器。TOEN 置 1 后，BUSTOA[11:0]和 TIDLE 不能被修改。如果 SCL 和 SDA 的高电平持续时间都大于 $(BUSTOA+1)*4*t_{I2CCLK}$ ，I2C_STAT 寄存器中 TIMEOUT 位将置位。

SMBus 从机模式

SMBus 接收器必须能够对接收到的命令和数据进行 NACK 应答。对于从机模式下的 ACK 控制，通过将 I2C_CTL0 寄存器中 SBCTL 位置 1 可以使能从机字节控制模式。

必要时应使能特定的 SMBus 地址。通过将 I2C_CTL0 寄存器中 SMBDAEN 置 1 可以使能 SMBus 设备默认地址（0b1100 001）。通过将 I2C_CTL0 寄存器中 SMBHAEN 置 1 可以使能 SMBus 主机地址（0b0001 000）。通过将 I2C_CTL0 寄存器中 SMBALTEN 置 1 可以使能报警响应地址（0b0001 100）。

25.3.10. SMBus 模式

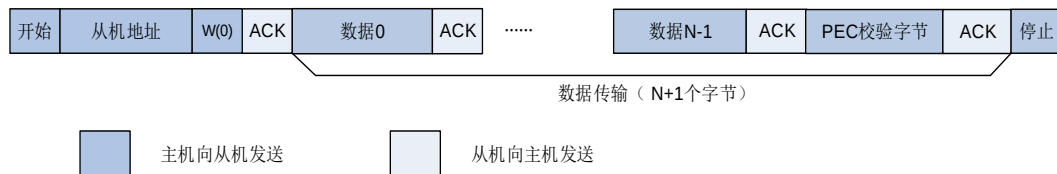
SMBus 主机发送器和从机接收器

当 SMBus 主机发送 PEC 时, 必须在 START 位置 1 前, 将 PECTRANS 位置 1 并在 BYTENUM[7:0] 位域中配置字节数。在这种情况下, 总 TI 中断数为 BYTENUM-1。因此, 如果 BYTENUM=0x1 且 PECTRANS 位置 1, 则 I2C_PEC 寄存器的数据将自动发送。如果 AUTOEND 为 1, SMBus 主机在 PEC 字节发送完成之后将自动发送 STOP 信号。如果 AUTOEND 为 0, SMBus 主机可以在 PEC 字节发送完成之后发送 RESTART 信号。I2C_PEC 寄存器中的数据将在 BYTENUM -1 个字节发送完成后发送, PEC 字节发送完成后 TC 位将置 1。SCL 线被拉低。RESTART 位必须在 TC 中断服务程序中置 1。

SMBus 作为从机接收器时, 为了在数据发送完成时进行 PEC 校验, SBCTL 位必须置 1。要对每个字节进行 ACK 控制, 必须通过将 RELOAD 位置 1 来使能 RELOAD 模式。如果要校验 PEC 字节, 必须将 RELOAD 位清零同时将 PECTRANS 置 1。在 BYTENUM-1 个字节接收完成后, 接收的下一个字节将与 I2C_PEC 寄存器中的数据进行比较。如果校验值不匹配, 将自动产生 NACK 信号; 如果校验值匹配将自动产生 ACK 信号, 将忽略 NACKEN 位的值。当接收到 PEC 字节时, PEC 字节会存到 I2C_RDATA 寄存器中, RBNE 位将置 1。如果 I2C_CTL0 寄存器中 ERRIE 位置 1, 且 PEC 值不匹配, PECERR 将会置 1 并产生中断。如果无须使用 ACK 控制, PECTRANS 可以设置为 1, BYTENUM 可以根据待接收字节数来配置。

注意: 在 RELOAD 位置 1 之后, PECTRANS 不可以被修改。

图 25-22. SMBus 主机发送器和从机接收器通信流程



SMBus 主机接收器和从机发送器

如果 SMBus 主机需要在数据传输完成后接收 PEC 字节, 可以使能自动结束模式。在 START 信号发送之前, 必须将 PECTRANS 位置 1, 且配置好从机地址。在接收 BYTENUM-1 数据之后, 接收的下一个字节将自动与 I2C_PEC 寄存器中的数据进行比较。在停止信号发送之前, 接收 PEC 字节之后会给出 NACK 响应。

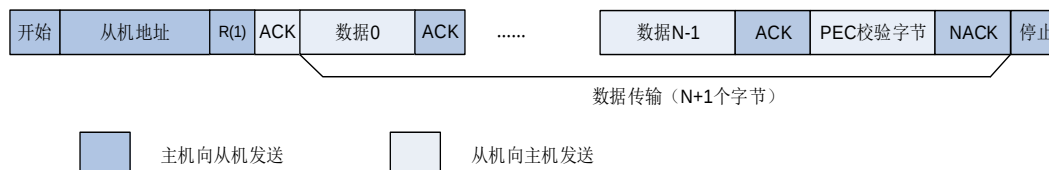
如果 SMBus 主机需要在接收到 PEC 字节之后产生 RESTART 信号, 需要禁能自动结束模式。在 START 信号发送之前, PECTRANS 位必须置 1, 且配置好从机地址。在接收 BYTENUM-1 数据之后, 接收的下一个字节将自动与 I2C_PEC 寄存器中的数据进行比较。在 PEC 字节发送完成之后 TC 位将置 1, SCL 线被拉低。在 TC 中断服务程序中可将 RESTART 位置 1。

当 SMBus 作为从机发送器时, 为了在 BYTENUM[7:0]个字节发送完成之后发送 PEC 字节, SBCTL 位必须置 1。如果 PECTRANS 置 1, 字节数 BYTENUM[7:0]包含 PEC 字节。在这种情况下, 如

果主机请求接收的字节数大于 BYTENUM-1，总 TI 中断数为 BYTENUM-1，I2C_PEC 寄存器中的数据将自动发送。

注意：PECTRANS 位在 RELOAD 置 1 之后不能被修改。

图 25-23. SMBus 主机接收器和从机发送器通信流程



25.3.11. DMA 模式下数据传输

如 I2C 从机模式和主机模式中描述，每当 TI 位和 RBNE 位被置 1 之后，软件都应该写或读一个字节，这样将导致 CPU 的负荷较重。I2C 的 DMA 功能可以在 TI 或 RBNE 位置 1 时，自动进行一次写或读操作。

将 I2C_CTL0 寄存器中 DENT 置 1 可以使能 DMA 发送请求。将 I2C_CTL0 寄存器中 DENR 置 1 可以使能 DMA 接收请求。在主机模式下，由软件写入从机地址，传输方向，待发送字节数和 START 位。DMA 必须在 START 位置 1 之前初始化。在 I2C_CTL1 寄存器 BYTENUM[7:0]位配置待传输字节数。在从机模式下，DMA 必须在地址匹配事件发生之前或 ADDSEND 中断服务程序中清除 ADDSEND 标志之前完成初始化。

25.3.12. I2C 错误和中断

I2C 错误标志如[表 25-4. I2C 错误标志](#)所示。

表 25-4. I2C 错误标志

I2C 错误名称	描述
BERR	总线错误
LOSTARB	仲裁丢失
OUERR	上溢 / 下溢标志
PECERR	CRC 值不匹配
TIMEOUT	SMBus 模式下总线超时
SMBALT	SMBus 报警

I2C 中断和事件标志如[表 25-5. I2C 中断事件](#)所示。

表 25-5. I2C 中断事件

中断事件	事件标志	使能控制位
在接收期间 I2C_RDATA 非空	RBNE	RBNEIE
发送中断	TI	TIE
从机模式下检测到 STOP 信号	STPDET	STPDETIE

中断事件	事件标志	使能控制位
传输完成重载	TCR	TCIE
传输完成	TC	
地址匹配	ADDSEND	ADDMIE
接收到 NACK	NACK	NACKIE
总线错误	BERR	ERRIE
仲裁丢失	LOSTARB	
上溢/下溢错误	OUERR	
PEC 错误	PECERR	
超时错误	TIMEOUT	
SMBus 报警	SMBALT	

25.3.13. I2C 调试模式

当控制器进入调试模式（Cortex®-M33 内核停止），SMBus 超时定时器会根据 DBG 模块中的 I2Cx_HOLD 配置位选择继续正常工作还是停止工作。

25.4. I2C 寄存器

I2C0 基地址：0x4000 5400

I2C1 基地址：0x4000 5800

25.4.1. 控制寄存器 0 (I2C_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								PECEN	SMBALT EN	SMBDAE N	SMBHAE N	GCEN	保留	SS	SBCTL
								rw	rw	rw	rw	rw		rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DENR	DENT	保留	ANOFF	DNF[3:0]				ERRIE	TCIE	STPDETI E	NACKIE	ADDmie	RBNEIE	TIE	I2CEN
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	PECEN	PEC 计算开关。 0: PEC 计算关闭。 1: PEC 计算打开。
22	SMBALTEN	SMBus 报警使能。 0: 从机模式下 SMBA 引脚高电平或主机模式下 SMBus 报警引脚 SMBA 禁能。 1: 从机模式下 SMBA 引脚低电平或主机模式下 SMBus 报警引脚 SMBA 使能。
21	SMBDAEN	SMBus 设备默认地址使能。 0: 设备默认地址禁能，对默认地址 0b1100001x 进行 NACK 应答。 1: 设备默认地址使能，对默认地址 0b1100001x 进行 ACK 应答。
20	SMBHAEN	SMBus 主机地址使能。 0: 主机地址禁能，对地址 0b0001000x 进行 NACK 应答。 1: 主机地址使能，对地址 0b0001000x 进行 ACK 应答。
19	GCEN	是否响应对地址（0x00）的广播呼叫。 0: 从机不响应广播呼叫。 1: 从机将响应广播呼叫。

18	保留	必须保持复位值。
17	SS	在从机模式下数据未就绪时是否将 SCL 拉低。 软件置 1 和清 0。 0: 拉低 SCL 1: 不拉低 SCL 注意: 在主机模式下, 该位必须为 0。该位只能在 I2CEN=0 时被修改。
16	SBCTL	从机模式下字节控制。 该位用于在从机模式下使能硬件字节控制。 0: 从机模式下字节控制禁能。 1: 从机模式下字节控制使能。
15	DENR	DMA 接收使能 0: DMA 接收禁能 1: DMA 接收使能
14	DENT	DMA 发送使能 0: DMA 发送禁能 1: DMA 发送使能
13	保留	必须保持复位值。
12	ANOFF	模拟噪声滤波器禁能 0: 模拟噪声滤波器使能。 1: 模拟噪声滤波器禁能。 注意: 该位只有在 I2C 禁能 (I2CEN=0) 时被编程。
11:8	DNF[3:0]	数字噪声滤波器 0000: 数字噪声滤波器禁能。 0001: 数字噪声滤波使能并且可以滤除脉宽宽度不大于 1 t_{I2CCLK} 的尖峰。 ... 1111: 数字噪声滤波使能并且可以滤除脉宽宽度不大于 15 t_{I2CCLK} 的尖峰。 这些位只能在 I2C 禁能 (I2CEN = 0) 时修改。
7	ERRIE	错误中断使能 0: 错误中断禁能 1: 错误中断使能, 当 BERR, LOSTARB, OUERR, PECERR, TIMEOUT 或 SMBALT 位置 1 时, 将产生中断。
6	TCIE	传输完成中断使能 0: 传输完成中断禁能。 1: 传输完成中断使能。
5	STPDETIE	停止信号检测中断使能 0: 停止信号 (STPDET) 检测中断禁能。

		1: 停止信号（STPDET）检测中断使能。
4	NACKIE	接收到 NACK 应答中断使能 0: 接收到 NACK 应答中断禁能。 1: 接收到 NACK 应答中断使能。
3	ADDMIE	从机模式下地址匹配中断使能 0: 地址匹配中断禁能。 1: 地址匹配中断使能。
2	RBNEIE	接收中断使能 0: 接收（RBNE）中断禁能。 1: 接收（RBNE）中断使能。
1	TIE	发送中断使能 0: 发送中断（TI）禁能。 1: 发送中断（TI）使能。
0	I2CEN	I2C 外设使能 0: I2C 禁能。 1: I2C 使能。

25.4.2. 控制寄存器 1（I2C_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					PECTRA NS	AUTOEN D	RELOAD	BYTENUM[7:0]							
					rw	rw	rw	rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NACKEN	STOP	START	HEAD10 R	ADD10E N	TRDIR	SADDRESS[9:0]									
rw	rw	rw	rw	rw	rw	rw									

位/位域	名称	描述
31:27	保留	必须保持复位值。
26	PECTRANS	PEC 传输 软件置 1 和清 0，硬件在以下条件下清除此位：PEC 传输完成或者 ADDSEND 置 1 或者检测到 STOP 信号或者 I2CEN=0。 0: 不传输 PEC 值。

		1: 传输的 PEC 值。 注意: 当 RELOAD=1 或者从机模式下 SBCTL=0 时, 该位无效。
25	AUTOEND	主机模式下自动结束模式 0: 当 BYTENUM[7:0]个字节传输完成后时, TC 位置 1。 1: 当 BYTENUM[7:0]个字节传输完成后时, 自动发送 STOP 信号。 注意: 该位仅在 RELOAD=0 时有效。该位由软件置 1 和清 0。
24	RELOAD	重载模式使能 0: 当 BYTENUM[7:0]个字节传输完成后时, 传输结束。 1: 当 BYTENUM[7:0]个字节传输完成后时, 传输未结束, 重载新的 BYTENUM[7:0]。 每次 BYTENUM[7:0]个字节传输完成, I2C_STAT 寄存器中 TCR 位将置 1。 该位由软件置 1 和清 0。
23:16	BYTENUM[7:0]	待传输的字节数 这些用来编程待传输的字节数。当 SBCTL=0 时, 这些位无效。 注意: 当 START 位置 1 时, 这些位不能被修改。
15	NACKEN	从机模式下产生 NACK 0: 在接收到新的字节时, 发送 ACK。 1: 在接收到新的字节时, 发送 NACK。 注意: 该位可由软件置 1, 并在以下情况下由硬件清零: NACK 发送完成或检测到 STOP 信号或 ADDSEND 置 1, 或 I2CEN=0。当 PEC 使能时, 发送 ACK 还是 NACK 与 NACKEN 值无关。当 SS=1 时, 且 OUERR 位置 1, NACKEN 的值会被忽略, 并且发送 NACK。
14	STOP	I2C 总线上产生一个 STOP 结束信号。 该位由软件置 1, 并在 I2CEN=0 或检测到 STOP 信号时由硬件清零。 0: 不发送 STOP。 1: 发送 STOP。
13	START	I2C 总线上产生一个 START 信号 该位由软件置 1, 并在从机地址发送后由硬件清零。当仲裁丢失时, 或发生超时错误, 或 I2CEN=0 时, 该位也可以由硬件清零。将 I2C_STATC 寄存器中 ADDSEND 位置 1 可以软件清除该位。 0: 不发送 START。 1: 发送 START。
12	HEAD10R	在主机接收模式下仅执行 10 位地址头读操作。 0: 主机发送 10 位从机地址读序列为 START + 10 位地址头 (写) + 第二个地址字节 + RESTART + 10 位地址头 (读)。 1: 主机寻址读序列为 RESTART + 10 位地址头 (读)。 注意: 当 START 位置 1 时, 该位不能被修改。
11	ADD10EN	主机模式下使能 10 位寻址模式

		0: 主机工作在 7 位寻址模式下。 1: 主机工作在 10 位寻址模式下。 注意: 当 START 位置 1 时, 该位不能被修改。
10	TRDIR	主机模式下传输方向 0: 主机发送 1: 主机接收 注意: 当 START 位置 1 时, 该位不能被修改。
9:0	SADDRESS[9:0]	待发送的从机地址 SADDRESS[9:8]: 从机地址 9:8 位。 如果 ADD10EN = 0, 该位域无效。 如果 ADD10EN = 1, 将该位域写入待发送从机地址的 9:8 位。 SADDRESS[7:1]: 从机地址 7:1 位。 如果 ADD10EN = 0, 在这些位写入待发送 7 位从机地址。 如果 ADD10EN = 1, 在这些位写入待发送从机地址的 7:1 位。 SADDRESS0: 从机地址 0 位。 如果 ADD10EN = 0, 这些位无效。 如果 ADD10EN = 1, 在这些位写入待发送从机地址的 0 位。 注意: 当 START 位置 1 时, 该位不能被修改。

25.4.3. 从机地址寄存器 0 (I2C_SADDR0)

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
31:16	保留	必须保持复位值。
15	ADDRESSEN	I2C 地址使能 0: I2C 地址禁能。 1: I2C 地址使能。
14:11	保留	必须保持复位值。

10	ADDFORMAT	I2C 从机地址模式 0: 7 位地址。 1: 10 位地址。 注意: 当 ADDRESSEN =1 时, 该位不能被改写。
9:8	ADDRESS[9:8]	10 位地址的最高两位 注意: 当 ADDRESSEN =1 时, 该位不能被改写。
7:1	ADDRESS[7:1]	7 位地址或者 10 位地址的第 7-1 位 注意: 1. 当 ADDRESSEN =1 时, 该位不能被改写。 2. 当从机地址设置为保留地址 0b'1111000 时, 如果地址匹配, 从机不响应。
0	ADDRESS0	10 位地址的第 0 位 注意: 当 ADDRESSEN =1 时, 该位不能被改写。

25.4.4. 从机地址寄存器 1 (I2C_SADDR1)

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
31:16	保留	必须保持复位值。
15	ADDRESS2EN	I2C 第二个地址使能 0: I2C 第二个地址禁能 1: I2C 第二个地址使能
14:11	保留	必须保持复位值。
10:8	ADDMSK2[2:0]	ADDRESS2[7:1]掩码 定义接收到的地址哪些位需要与 ADDRESS2[7:1]进行比较, 哪些位屏蔽 (不比较)。 000: 不屏蔽, 所有的位都进行比较。 n (001~110): ADDRESS2[n:1]屏蔽。ADDRESS2[7:n+1]需要进行比较。 111: ADDRESS2[7:1]屏蔽。对于接收到的所有 7 位地址都会进行 ACK 应答, 保留

地址（0b0000xxx 和 0b1111xxx）除外。

注意：当 ADDRESS2EN=1 时，该位不能被改写。如果 ADDMSK2 不等于 0，即使所有位都匹配，I2C 保留地址（0b0000xxx 和 0b1111xxx）也不会进行 ACK 应答。

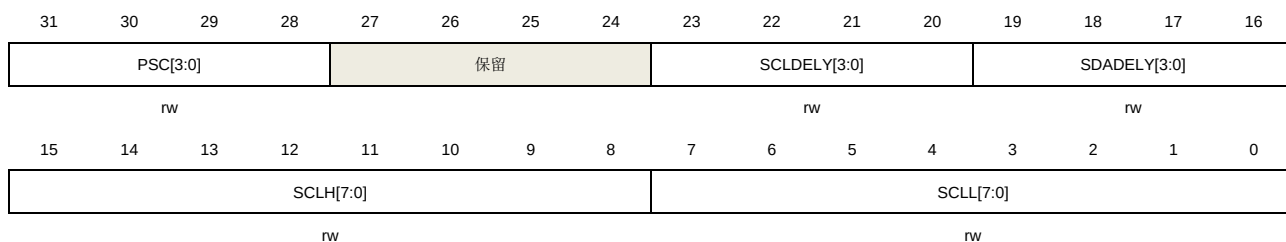
7:1	ADDRESS2[7:1]	I2C 从机的第二个地址 注意： 1. 当 ADDRESS2EN=1 时，该位不能被改写。 2. 当从机地址设置为保留地址 0b'1111000 时，如果地址匹配，从机不响应。
0	保留	必须保持复位值。

25.4.5. 时序寄存器（I2C_TIMING）

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
31:28	PSC[3:0]	时序预分频 为了生成用于数据建立和数据保持的计数器的时钟周期 t_{PSC} ，这些位用于配置 I2CCLK 时钟预分频。 t_{PSC} 也用于 SCL 高电平和低电平计数器。 $t_{PSC}=(PSC+1)*t_{I2CCLK}$ 。
27:24	保留	必须保持复位值。
23:20	SCLDEL[3:0]	数据建立时间 这些位用于在 SDA 边沿和 SCL 上升沿之间生成延时 t_{SCLDEL} 。在主机模式下和在从机模式下 SS=0 时，在 t_{SCLDEL} 期间 SCL 线被拉低。 $t_{SCLDEL}=(SCLDEL+1)*t_{PSC}$ 。
19:16	SDADEL[3:0]	数据保持时间 这些位用于在 SCL 下降沿和 SDA 边沿之间生成延时 t_{SDADEL} 。在主机模式下和在从机模式下 SS=0 时，在 t_{SDADEL} 期间 SCL 线被拉低。 0x0: $t_{SDADEL} = 1*t_{PSC}$ 0x1-0xF: $t_{SDADEL}=SDADEL*t_{PSC}$
15:8	SCLH[7:0]	SCL 高电平周期

SCL 高电平周期可以通过配置这些位来产生。

$$t_{SCLH} = (SCLH + 1) * t_{PSC}$$

注意：这些位只能用于主机模式。

7:0 SCLL[7:0]

SCL 低电平周期

SCL 低电平周期可以通过配置这些位来产生。

$$t_{SCLL} = (SCLL + 1) * t_{PSC}$$

注意：这些位只能用于主机模式。

25.4.6. 超时寄存器 (I2C_TIMEOUT)

地址偏移: 0x14

复位值: 0x0000 0000

当 I2C (I2CEN=1) 被使能时, 该寄存器不能被改写。

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
EXTOEN	保留			BUSTOB[11:0]											
rw				rw											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOEN	保留		TOIDLE	BUSTOA[11:0]											
rw			rw			rw									

位/位域	名称	描述
31	EXTOEN	时钟信号延展超时使能 当 SCL 累计拉低时间大于 $t_{LOW:EXT}$ 时, 将会产生超时错误, $t_{LOW:EXT} = (BUSTOB + 1) * 2048 * t_{I2CCLK}$ 0: 时钟信号延展超时检测禁能。 1: 时钟信号延展超时检测使能。
30:28	保留	必须保持复位值。
27:16	BUSTOB[11:0]	总线超时 B 配置累积时钟延展超时。在主机模式下, 检测主机累计时钟低电平延展时间 $t_{LOW:MEXT}$ 。从机模式下, 检测从机累计时钟低电平延展时间 $t_{LOW:SEXT}$ 。 $t_{LOW:EXT} = (BUSTOB + 1) * 2048 * t_{I2CCLK}$ 注意： 该位域仅在 EXTOEN = 0 时可以被修改。
15	TOEN	时钟超时使能 当 TOIDLE = 0, SCL 拉低时间大于 $t_{TIMEOUT}$ 或当 TOIDLE = 1, SCL 拉低时间大于 t_{IDLE} , 将检测到超时错误。 0: SCL 超时检测禁能

		1: SCL 超时检测使能
14:13	保留	必须保持复位值。
12	TOIDLE	空闲时钟超时检测 0: BUSTOA 用于检测 SCL 低电平超时。 1: BUSTOA 用于检测 SCL 和 SDA 高电平超时（总线空闲条件）。 注意： 该位域仅在 TOEN = 0 时可以被改写。
11:0	BUSTOA[11:0]	总线超时 A 当 TOIDLE=0 时, $t_{\text{TIMEOUT}}=(\text{BUSTOA}+1)*2048*t_{\text{I2CCCLK}}$ 当 TOIDLE=1 时, $t_{\text{IDLE}}=(\text{BUSTOA}+1)*4*t_{\text{I2CCCLK}}$ 注意： 该位域仅在 TOEN = 0 时可以被改写。

25.4.7. 状态寄存器（I2C_STAT）

地址偏移：0x18

复位值：0x0000 0001

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								READDR[6:0]							TR
								r							r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2CBSY	保留	SMBALT	TIMEOUT	PECERR	OUERR	LOSTAR B	BERR	TCR	TC	STPDET	NACK	ADDSEN D	RBNE	TI	TBE
r		r	r	r	r	r	r	r	r	r	r	r	r	rW	rW

位/位域	名称	描述
31:24	保留	必须保持复位值。
23:17	READDR[6:0]	从机模式下接收到的匹配地址 当 ADDSEND 置 1 时，这些位用于存储接收到的地址。在 10 位地址情况下，READDR[6:0]存储 10 位地址头和地址的最高两位。
16	TR	I2C 在从机模式下作为发送端还是接收端 该位在 ADDSEND 位置 1 时更新。 0: 接收端 1: 发送端
15	I2CBSY	忙标志 该位在硬件检测到 START 信号时置 1。在 STOP 信号后硬件清 0。当 I2CEN=0 时，由硬件清零。 0: 无 I2C 通讯

		1: I2C 正在通讯
14	保留	必须保持复位值。
13	SMBALT	SMBus 报警 当 SMBHAEN=1, SMBALTEN=1 且在 SMBA 引脚检测到 SMBALERT 事件（下降沿）时，该位由硬件置 1。SMBALTC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。 0: 在 SMBA 引脚上检测到 SMBALERT 事件。 1: 在 SMBA 引脚上未检测到 SMBALERT 事件。
12	TIMEOUT	超时标志 当发生超时或延展时钟超时，该位将置 1。TIMEOUTC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。 0: 无超时或延展时钟超时发生。 1: 发生超时或延展时钟超时。
11	PECERR	PEC 错误 当接收到的 PEC 字节与 I2C_PEC 寄存器中的内容不匹配时，该位置 1。然后将自动发生 NACK。PECERRC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。 0: 接收到 PEC 与 I2C_PEC 的内容匹配。 1: 接收到 PEC 与 I2C_PEC 的内容不匹配，此时 I2C 将忽略 NACKEN 位的值，并直接发送 NACK。
10	OUERR	从模式下上溢 / 下溢错误 在从机模式下且 SS=1，当发生上溢 / 下溢错误时，该位置 1。OUERRC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。 0: 未发生上溢 / 下溢错误。 1: 发生上溢 / 下溢错误。
9	LOSTARB	仲裁丢失 LOSTARBC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。 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 当 I2C 总线上发生了预料之外的 START 信号或 STOP 信号时，将产生总线错误，该位将置 1。BERRC 置 1 可以将该位软件清零。当 I2CEN=0 时，该位由硬件清零。 0: 无总线错误。 1: 发生了总线错误。
7	TCR	传输完成重载 当 RELOAD=1 且 BYTENUM[7:0] 个字节传输完成时，该位置 1。在 BYTENUM[7:0] 写入一个非零值可以软件清零该位。

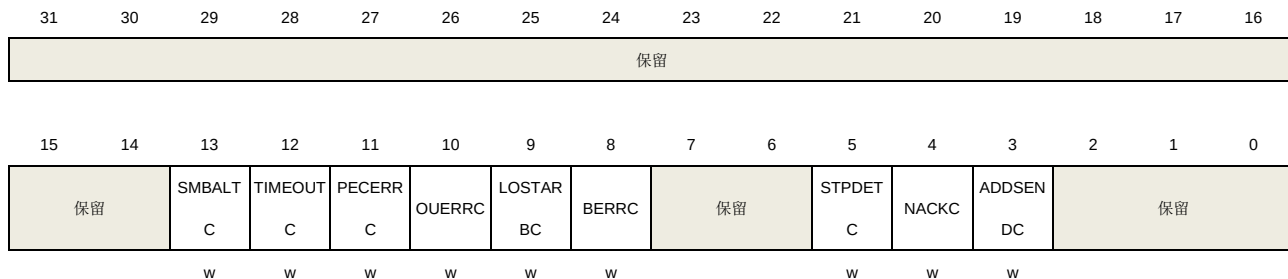
		0: 当 RELOAD=1 时, BYTENUM[7:0]个字节传输未完成。 1: 当 RELOAD=1 时, BYTENUM[7:0]个字节传输完成。
6	TC	主机模式下传输完成 当 RELOAD=0, AUTOEND=0 且 BYTENUM[7:0]个字节传输完成时, 该位置 1。当 START 位或 STOP 位置 1 时该位清零。 0: BYTENUM[7:0]个字节传输未完成。 1: BYTENUM[7:0]个字节传输完成。
5	STPDET	总线上检测到 STOP 信号 当在总线上检测到 STOP 信号时, 主机和从机的该位由硬件置 1。STPDETC 置 1 可以将该位软件清零。当 I2CEN=0 时, 该位由硬件清零。 0: 未监测到 STOP 结束位。 1: 监测到 STOP 结束位。
4	NACK	接收到 NACK 应答 当接收到 NACK 时, 该位置 1。NACKC 置 1 可以将该位软件清零。当 I2CEN=0 时, 该位由硬件清零。 0: 接收到 ACK。 1: 接收到 NACK。
3	ADDSEND	从机模式下接收到的地址与自身地址匹配 当接收到的地址与使能的从机地址之一匹配时, 该位由硬件置 1。ADDSEND 置 1 可以将该位软件清零。当 I2CEN=0 时, 该位由硬件清零。 0: 接收到的地址不匹配。 1: 接收到的地址匹配。
2	RBNE	接收期间 I2C_RDATA 非空 当接收到的数据移入 I2C_RDATA 寄存器时, 该位置 1。读 I2C_RDATA 可清除该位。 0: I2C_RDATA 空。 1: I2C_RDATA 非空, 软件可以读。
1	TI	发送中断 当 I2C_TDATA 为空且 I2C 已经做好发送数据准备时, 该位置 1。在下一个待发送字节写入 I2C_TDATA 寄存器时该位清零。当 SS=1 时, 可由软件将该位置 1 来产生 TI 事件 (TIE=1 时为中断, DENT=1 时为 DMA 请求)。 0: I2C_TDATA 非空或者 I2C 还未做好发送数据准备。 1: I2C_TDATA 空且 I2C 已经做好发送数据准备。
0	TBE	发送期间 I2C_TDATA 空 当 I2C_TDATA 寄存器为空, 该位置 1。当下一个待发送数据写入 I2C_TDATA 寄存器时, 该位清零。可以软件将该位置 1 来清空 I2C_TDATA 寄存器。 0: I2C_TDATA 非空。 1: I2C_TDATA 空。

25.4.8. 状态清除寄存器 (I2C_STATC)

地址偏移: 0x1C

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



位/位域	名称	描述
31:14	保留	必须保持复位值。
13	SMBALTC	SMBus 报警标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 SMBALT 位。
12	TIMEOUTC	TIMEOUT 标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 TIMEOUT 位。
11	PECERRC	PEC 错误标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 PECERR 位。
10	OUERRC	上溢 / 下溢标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 OUERR 位。
9	LOSTARBC	仲裁丢失标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 LOSTARB 位。
8	BERRC	总线错误标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 BERR 位。
7:6	保留	必须保持复位值。
5	STPDETC	停止位检测标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 STPDET 位。
4	NACKC	NACK 标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 NACK 位。
3	ADDSENDC	地址匹配标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 ADDSEND 位。

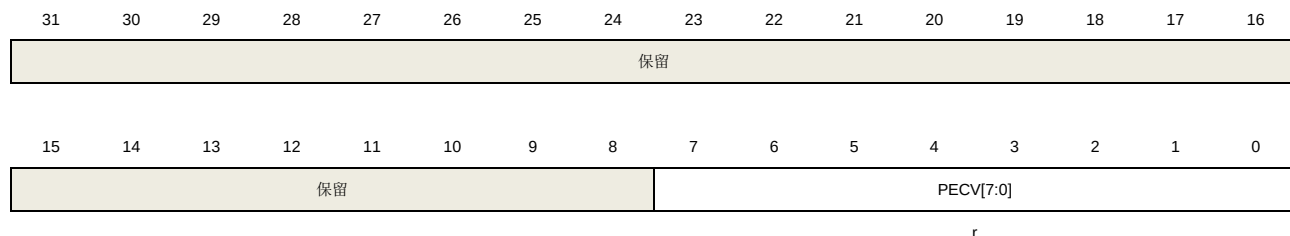
2:0 保留 必须保持复位值。

25.4.9. PEC 寄存器（I2C_PEC）

地址偏移：0x20

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



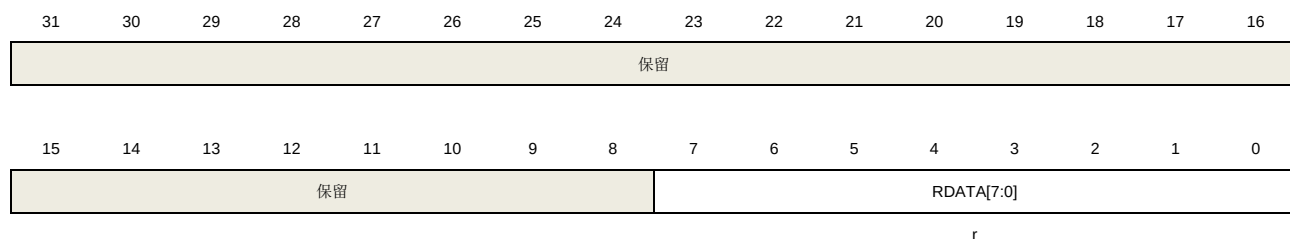
位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	PECV[7:0]	在 PEC 使能时，由硬件计算出来的 PEC 值。 当 I2CEN = 0 时，PECV 由硬件清零。

25.4.10. 接收数据寄存器（I2C_RDATA）

地址偏移：0x24

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



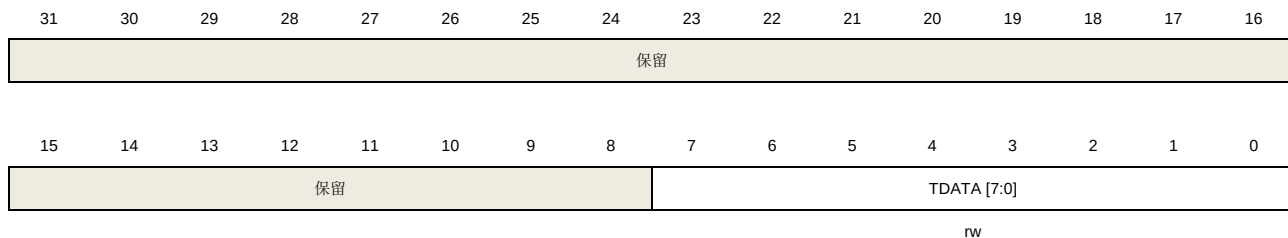
位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	RDATA[7:0]	接收到的数据

25.4.11. 发送数据寄存器（I2C_TDATA）

地址偏移：0x28

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



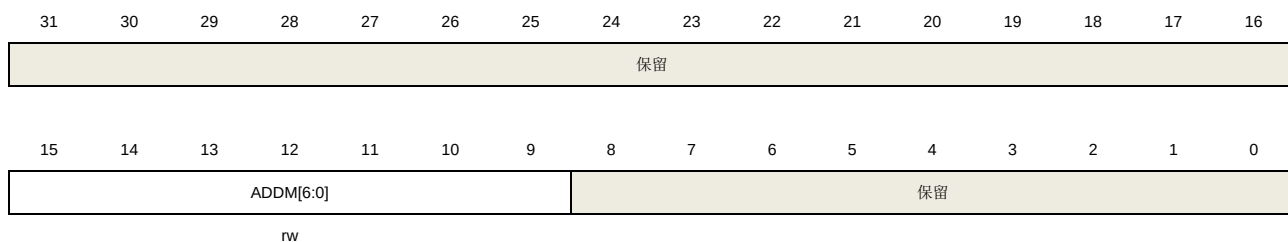
位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	TDATA[7:0]	发送的数据

25.4.12. 控制寄存器 2 (I2C_CTL2)

地址偏移：0x90

复位值：0x0000 FE00

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。
15:9	ADDM[6:0]	定义 ADDRESS[7:1]的哪些位和接收到的地址进行比较，哪些位不比较。ADDM[6:0]中设置为 1 的位使能 ADDRESS[7:1]中的相应位与接收到的地址进行比较，设置为 0 的位则忽略（此时接收到的地址在该位可以为 0 或 1）。
8:0	保留	必须保持复位值。

26. 控制器局域网（CAN）

26.1. 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、ISO11898-1:2015 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-trigger communication）功能。

26.2. 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 8Mbit/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。

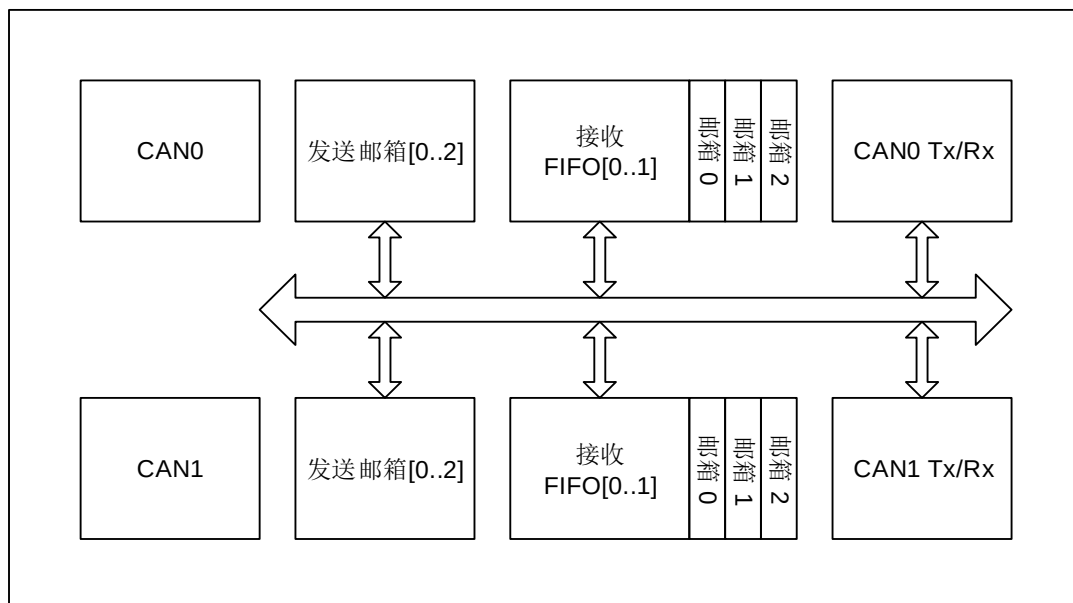
时间触发通信

- 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。

26.3. 功能说明

CAN 模块结构框图如 [图 26-1. CAN 模块结构框图](#) 所示。

图 26-1. CAN 模块结构框图



26.3.1. 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。

26.3.2. 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。

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。

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。

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 26-2. 发送寄存器所示。

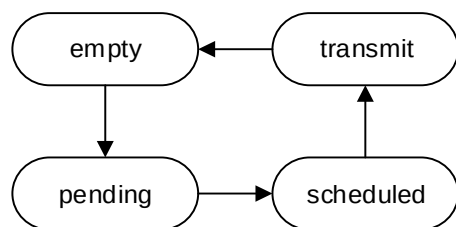
The diagram illustrates a mailbox-based communication system. On the left, a box labeled "应用" (Application) is connected to three parallel processing channels. Each channel consists of a vertical stack of four boxes: "TMI" (Mailbox Index), "TMP" (Mailbox Pointer), "TMDATA0", and "TMDATA1". The channels are labeled "TMI0/TMP0/TMDATA00/TMDATA10", "TMI1/TMP1/TMDATA01/TMDATA11", and "TMI2/TMP2/TMDATA02/TMDATA12". Arrows point from the "应用" box to the "TMP" box of each channel. From the "TMDATA0" box of each channel, an arrow points to a corresponding "发送邮箱" (Transmit Mailbox) box on the right, labeled "发送邮箱 0", "发送邮箱 1", and "发送邮箱 2" respectively.

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 26-3. 发送邮箱状态转换](#) 所示。

图 26-3. 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。

26.3.4. 数据接收

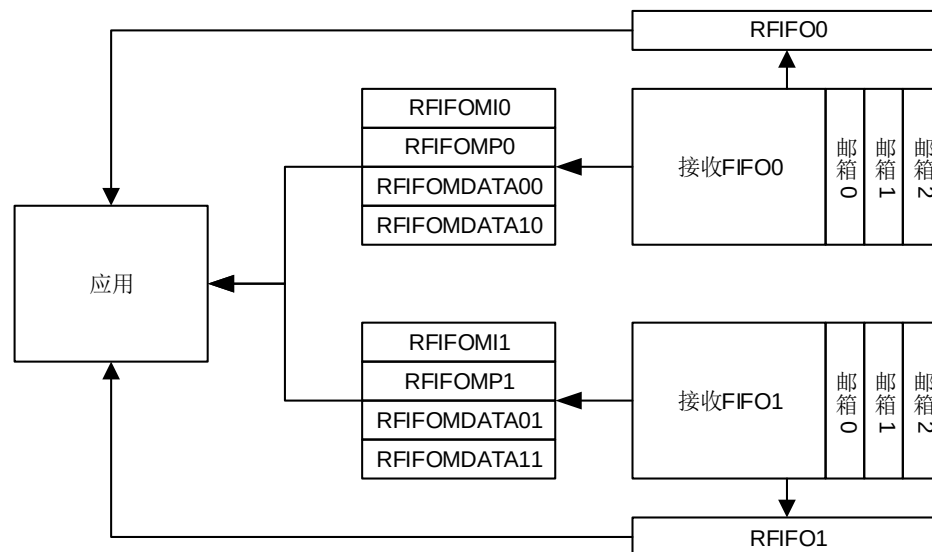
接收寄存器

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_RFIFOMx 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 26-4. 接收寄存器](#) 所示。

图 26-4. 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。

26.3.5. 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。

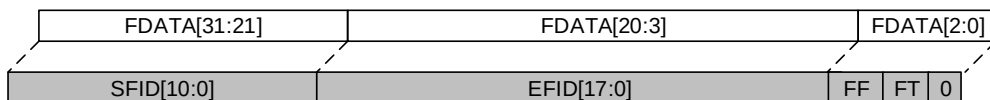
过滤器位宽

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。

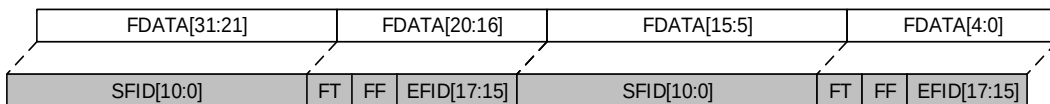
32-bit 位宽：CAN_FDATAx 包含字段 SFID[10:0]，EFID[17:0]，FF 和 FT。如[图 26-5. 32-bit 位宽过滤器](#)所示。

图 26-5. 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 26-6. 16-bit 位宽过滤器](#) 所示。

图 26-6. 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 26-7. 32-bit 位宽掩码模式过滤器](#) 所示。

图 26-7. 32-bit 位宽掩码模式过滤器

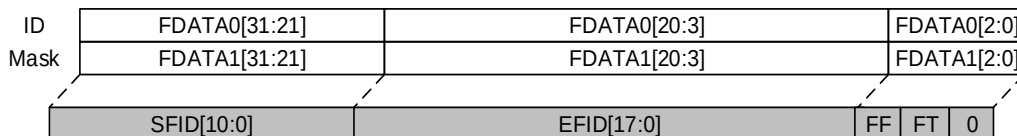
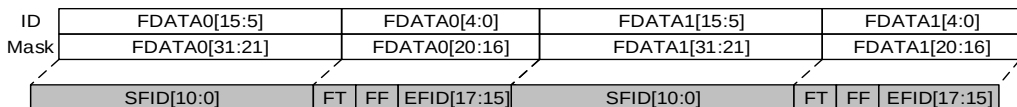


图 26-8. 16-bit 位宽掩码模式过滤器



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 26-9. 32-bit 位宽列表模式过滤器](#) 所示。

图 26-9. 32-bit 位宽列表模式过滤器

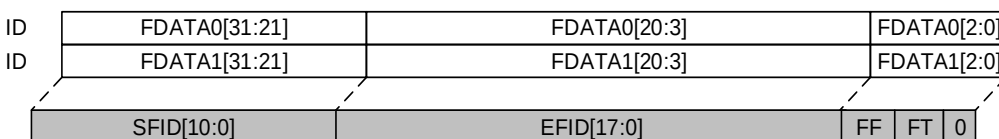
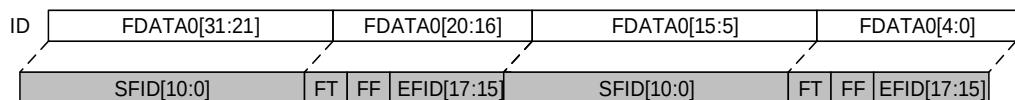


图 26-10. 16-bit 位宽列表模式过滤器



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 26-1. 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 26-1. 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 26-2. 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 26-2. 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-		

过滤单元	FIFO0	激活	过滤序号	过滤单元	FIFO1	激活	过滤序号
					Mask		
1	F1DATA0-32bits-ID	是	1		F2DATA1[15:0]-16bits-ID		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		4
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]-16bits-ID		7		F6DATA1[15:0]-16bits-ID		7
	F7DATA1[31:16]-16bits-ID		8		F6DATA1[31:16]-16bits-ID		8
8	F8DATA0[15:0]-16bits-ID	是	9	10	F10DATA0[15:0]-16bits-ID	否	9
	F8DATA0[31:16]-16bits-ID		10		F10DATA0[31:16]-16bits-Mask		
	F8DATA1[15:0]-16bits-ID		11		F10DATA1[15:0]-16bits-ID		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	否	11
	F9DATA0[31:16]-16bits-Mask				F11DATA0[31:16]-16bits-ID		12
	F9DATA1[15:0]-16bits-ID		14		F11DATA1[15:0]-16bits-ID		13
	F9DATA1[31:16]-16bits-Mask				F11DATA1[31:16]-16bits-ID		14
12	F12DATA0-32bits-ID	是	15	13	F13DATA0-32bits-ID	是	15
	F12DATA1-32bits-ID				F13DATA1-32bits-ID		16

过滤单元	FIFO0	激活	过滤序号	过滤单元	FIFO1	激活	过滤序号
	Mask						

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。

26.3.6. 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。

26.3.7. 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。

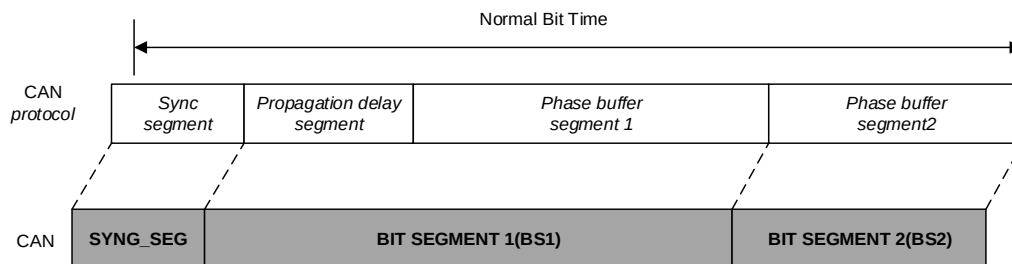
同步段（Synchronization segment），记为 SYNC_SEG。该段占用 1 个时间单元（ $1 \times t_{CAN}$ ）。

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 26-11. 位时序](#) 所示。

图 26-11. 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：

$$\text{BaudRate} = \frac{1}{\text{Normal Bit Time}} \quad (26-1)$$

$$\text{Normal Bit Time} = t_{\text{SYNC\_SEG}} + t_{\text{BS1}} + t_{\text{BS2}} \quad (26-2)$$

其中：

$$t_{\text{SYNC\_SEG}} = 1 \times t_q \quad (26-3)$$

$$t_{\text{BS1}} = (1 + \text{BT.BS1}) \times t_q \quad (26-4)$$

$$t_{\text{BS2}} = (1 + \text{BT.BS2}) \times t_q \quad (26-5)$$

$$t_q = (1 + \text{BT.BAUDPSC}) \times t_{\text{PCLK1}} \quad (26-6)$$

26.3.8. 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）的位速率高于仲裁段的位速率。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。

26.3.9. 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。

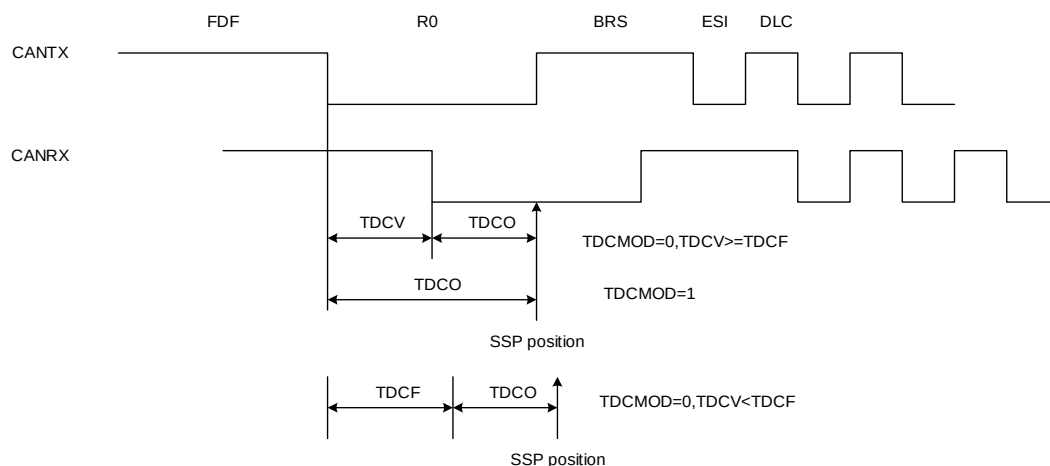
将 CAN_FDCTL 寄存器的 TDCEN 位置 1 将使能传输延迟补偿。

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 TDCF 位域。如果

TDCV 的值大于 TDCF，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 TDCF。

SSP_Delay 的值不能大于 3 个数据位时间。

图 26-12 传输延迟测量



26.3.10. 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。

26.3.11. 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；
- FIFO0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；
- FIFO1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。

CAN 总线控制器的中断产生条件可参考[表 26-3. CAN 事件/中断标志](#)。

表 26-3. CAN 事件/中断标志

中断事件	事件/中断标志		使能控制位	
发送中断	发送邮箱 0 空闲标志 MTF0		TMEIE	
	发送邮箱 1 空闲标志 MTF1			
	发送邮箱 2 空闲标志 MTF2			
FIFO0 中断	接收 FIFO0 中帧的数量 RFL0[1:0]		RFNEIE0	
	接收 FIFO0 满 RFF0		RFFIE0	
	接收 FIFO0 溢出 RFO0		RFOIE0	
FIFO1 中断	接收 FIFO1 中帧的数量 RFL1[1:0]		RFNEIE1	
	接收 FIFO1 满 RFF1		RFFIE1	
	接收 FIFO1 溢出 RFO1		RFOIE1	
EWMC 中断	警告错误 WERR	错误中断标志 ERRIF	WERRIE	ERRIE
	被动错误 PERR		PERRIE	
	离线错误 BOERR		BOIE	
	错误种类 1<= ERRN[2:0] <= 6		ERRNIE	
	从睡眠工作模式唤醒的状态改变中断标志WUIF		WIE	
进入睡眠工作模式的状态改变中断标志SLPIF		SLPWIE		

26.4. CAN 寄存器

CAN0 基地址：0x4001 5800

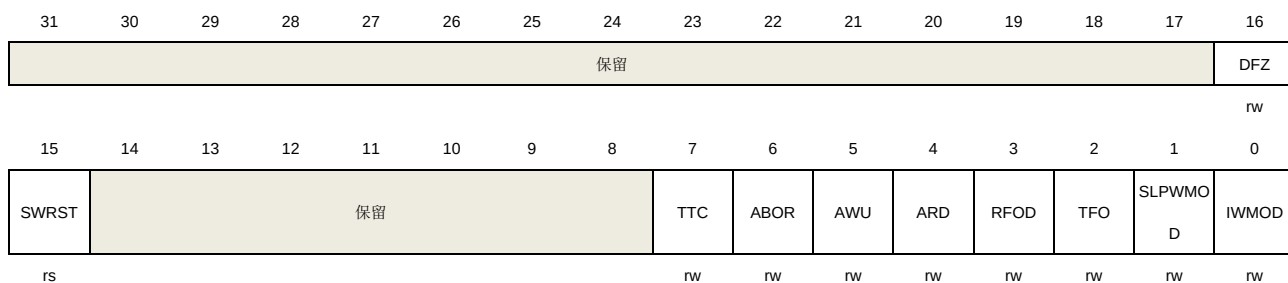
CAN1 基地址：0x4001 5C00

26.4.1. 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。



位/位域	名称	描述
31:17	保留	必须保持复位值。
16	DFZ	调试冻结 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。 0：处于Debug时，CAN接收和发送正常工作 1：处于Debug时，CAN接收和发送停止
15	SWRST	软件复位 0：正常操作 1：复位CAN并进入睡眠工作模式。该位会自动清0。
14:8	保留	必须保持复位值。
7	TTC	时间触发通信 0：禁用时间触发通信 1：使能时间触发通信
6	ABOR	自动离线恢复 0：通过软件手动地从离线状态恢复 1：通过硬件自动的从离线状态恢复
5	AWU	自动唤醒

一旦自动唤醒后，CAN_CTL寄存器的SLPWMOD位将自动被清0。

0：通过软件手动的从睡眠工作模式唤醒

1：通过硬件自动的从睡眠工作模式唤醒

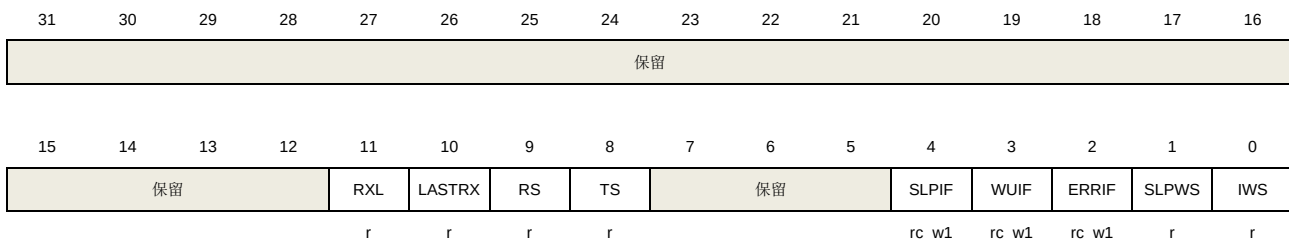
4	ARD	自动重发禁止 0：使能自动重发 1：禁用自动重发
3	RFOD	禁用接收FIFO满时覆盖 0：使能接收FIFO满时覆盖。当接收FIFO满时，FIFO中的数据被新来的数据覆盖 1：禁用接收FIFO满时覆盖。当接收FIFO满时，新来的数据被丢弃，FIFO中的数据保持不变，不会被覆盖
2	TFO	发送FIFO顺序 0：标识符（Identifier）较小的帧先发送 1：所有等待发送的邮箱按照先进先出（FIFO）的顺序发送
1	SLPWMOD	睡眠工作模式 如果软件将该位置1，CAN将会在当前发送或接收完成时进入睡眠工作模式。该位可由软件或者硬件清0。如果CAN_CTL寄存器中AWU被置位，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

26.4.2. 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。



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	发送状态 0: CAN当前不是发送器 1: CAN当前是发送器
7:5	保留	必须保持复位值。
4	SLPIF	进入睡眠工作模式的状态改变中断标志 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。 0: CAN没有进入睡眠工作模式 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
3	WUIF	从睡眠工作模式唤醒的状态改变中断标志 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。 0: 没有检测到唤醒信号 1: 发现唤醒信号。如果相应的中断使能位为1，则发生中断。
2	ERRIF	错误中断标志 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。 0: 没有错误 1: 发生错误。如果相应的中断使能位为1，则发生中断
1	SLPWS	睡眠工作状态 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。 0: CAN没有处于睡眠工作状态 1: CAN处于睡眠工作状态
0	IWS	初始化工作状态 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。

0: CAN没有处于初始化工作状态

1: CAN处于初始化工作状态

26.4.3. 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	MTFNER R2	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	MTFNER R1	MTF1	MST0	保留			MTE0	MAL0	MTFNER R0	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	邮箱2仲裁失败 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。
17	MTFNERR2	邮箱2无错发送完成 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。 0：传输结束时发生了错误 1：传输结束且没有错误
16	MTF2	邮箱2发送完成 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。 0：发送邮箱2正在发送 1：发送邮箱2完成发送
15	MST1	邮箱1停止发送 将其置1，将停止邮箱1的发送过程。 当邮箱1变为 empty 状态时，该位被硬件自动清0。
14:12	保留	必须保持复位值。
11	MTE1	邮箱1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MTF1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。
10	MAL1	邮箱1仲裁失败 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。 当发生仲裁失败时该位被置1。
9	MTFNERR1	邮箱1无错发送完成 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。

0: 传输结束时发生了错误

1: 传输结束且没有错误

8	MTF1	邮箱1发送完成 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。 0: 发送邮箱1正在发送 1: 发送邮箱1完成发送
7	MST0	邮箱0停止发送 将其置1，将停止邮箱0的发送过程。 当邮箱0变为empty状态时，该位被硬件自动清0。
6:4	保留	必须保持复位值。
3	MTE0	邮箱0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MTF0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。
2	MAL0	邮箱0仲裁失败 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。 当发生仲裁失败时该位被置1。
1	MTFNERR0	邮箱0无错发送完成 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。 0: 传输结束时发生了错误 1: 传输结束且没有错误
0	MTF0	邮箱0发送完成 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。 0: 发送邮箱0正在发送 1: 发送邮箱0完成发送

26.4.4. 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。

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							RFOVRN0	保留		RFD0	RFO0	RFF0	保留	RFL0[1:0]	
							r			rs	rc w1	rc w1	r		

位/位域	名称	描述
31:9	保留	必须保持复位值。
8	RFOVRN0	Rx FIFO0溢出 当启用了Rx FIFO覆盖写入功能且Rx FIFO0发生溢出时，该位会被硬件置1。 0: Rx FIFO0未发生溢出 1: Rx FIFO0发生溢出 注意：当该位被置位时，不要从FIFO中读取数据。
7:6	保留	必须保持复位值。
5	RFD0	释放一次FIFO0中的数据 该位被置1，将释放FIFO0中的一帧数据。 FIFO0释放相应的数据空间后，该位被清0。
4	RFO0	接收FIFO0溢出 当接收FIFO0溢出时被置位，由软件写1清0。 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

26.4.5. 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。

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	RFOVRN1	保留	RFD1	RFO1	RFF1	保留	RFL1[1:0]
	r		rs	rc_w1	rc_w1		r

位/位域	名称	描述
31:9	保留	必须保持复位值。
8	RFOVRN1	Rx FIFO1溢出 当启用了Rx FIFO覆盖写入功能且Rx FIFO1发生溢出时，该位会被硬件置1。 0: Rx FIFO1未发生溢出 1: Rx FIFO1发生溢出 注意：当该位被置位时，不要从FIFO中读取数据。
7:6	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

26.4.6. 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
														r/w	r/w
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
r/w				r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w	r/w	r/w

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	RFFIE1	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

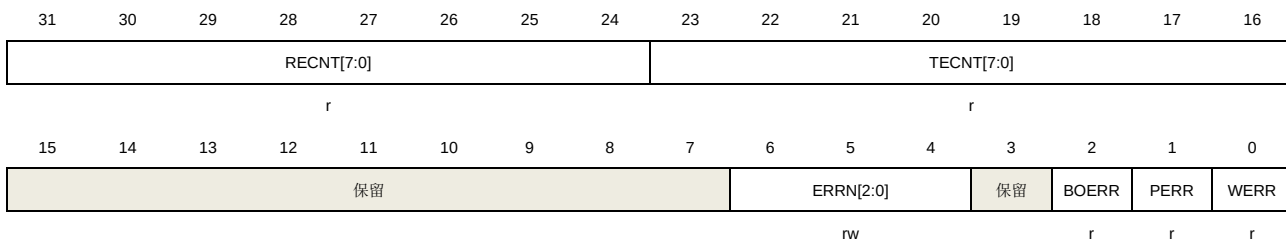
3	RFOIE0	接收FIFO0溢出中断使能 0: 禁用接收FIFO0溢出中断 1: 使能接收FIFO0溢出中断
2	RFFIE0	接收FIFO0满中断使能 0: 禁用接收FIFO0满中断 1: 使能接收FIFO0满中断
1	RFNEIE0	接收FIFO0非空中断使能 0: 禁用接收FIFO0非空中断 1: 使能接收FIFO0非空中断
0	TMEIE	发送邮箱空中断使能 0: 禁用发送邮箱空中断 1: 使能发送邮箱空中断

26.4.7. 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RECNr大于127时，该位由硬件置1。
0	WERR	警告错误 当TECNT或RECNr大于等于96时，该位由硬件置1。

26.4.8. 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

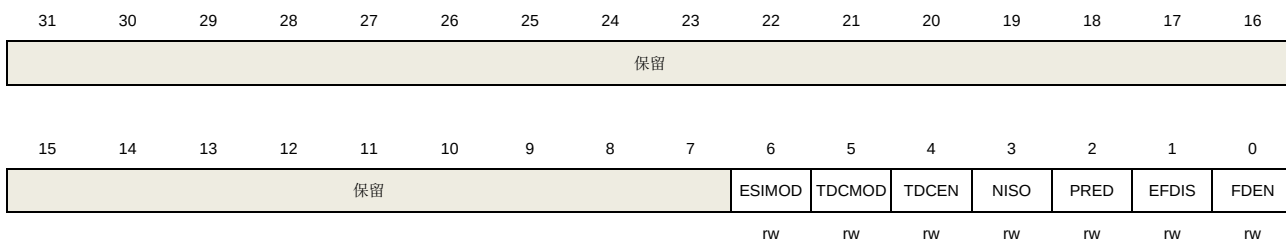
		位段2占用的时间单元数量=BS2[4:0]+1
19:16	BS1[3:0]	位段1的3:0位 位段1占用的时间单元数量=BS1[6:0]+1
15	保留	必须保持复位值。
14:13	BS2[4:3]	位段2的4:3位 参考CAN_BT寄存器的22:20位（FDEN为1时配置有效）
12:10	BS1[6:4]	位段1的6:4位 参考CAN_BT寄存器的19:16位（FDEN为1时配置有效）
9:0	BAUDPSC[9:0]	波特率分频系数

26.4.9. FD 控制寄存器（CAN_FDCTL）

地址偏移：0x20

复位值：0x0000 0002

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	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	NISO	ISO/BOSCH

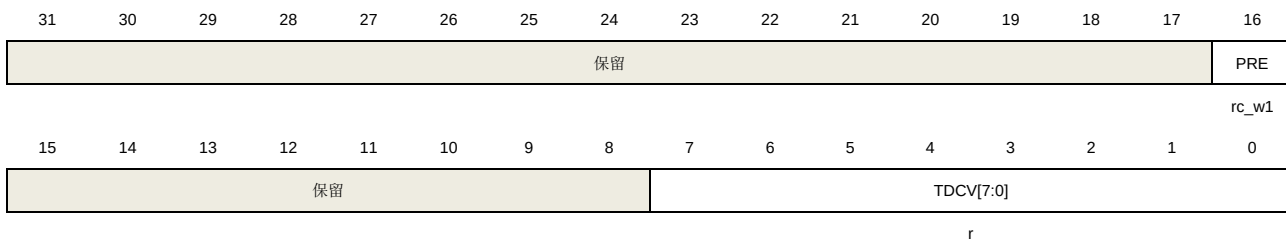
		0: ISO
		1: BOSCH
2	PRED	协议异常事件检测禁用 0: 协议异常事件检测使能（当接收到隐性保留位时转为IDLE状态） 1: 协议异常事件检测禁用（当接收到隐性保留位时，将此视为帧格式错误）
1	EFDIS	边沿过滤禁能 0: 使能边沿过滤 1: 禁能边沿过滤
0	FDEN	FD功能使能 0: CAN-FD功能禁止 1: CAN-FD功能使能

26.4.10. 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。
15:8	保留	必须保持复位值。
7:0	TDCV[7:0]	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。

26.4.11. 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。

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
TDCO[7:0]								TDCF[7:0]							
rw								rw							

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	TDCO[7:0]	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。
7:0	TDCF[7:0]	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。

26.4.12. 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					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

26.4.13. 发送邮箱标识符寄存器（CAN_TMIx）（x = 0..2）

地址偏移：0x180 + 0x10 * x

复位值：0xFFFF XXXX (bit0=0)

该寄存器只能按字（32 位）访问。



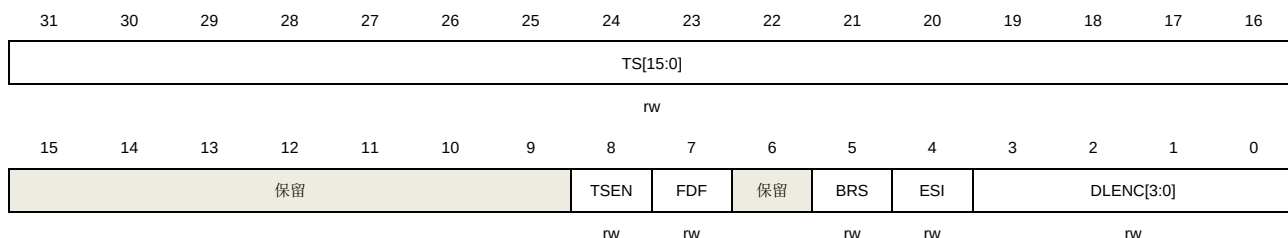
位/位域	名称	描述
31:21	SFID[10:0]/EFID[28:18]	标识符
	8]	SFID[10:0]: 标准格式帧标识符 EFID[28:18]: 扩展格式帧标识符
20:16	EFID[17:13]	标识符 EFID[17:13]: 扩展格式帧标识符
15:3	EFID[12:0]	标识符 EFID[12:0]: 扩展格式帧标识符
2	FF	帧格式 0: 标准格式帧 1: 扩展格式帧
1	FT	帧种类 0: 数据帧 1: 遥控帧
0	TEN	发送使能 当应用程序想要发送数据时，该位被置1将启动发送过程。当发送结束，发送邮箱为空时，该位由硬件清0。 0: 禁用发送 1: 使能发送

26.4.14. 发送邮箱属性寄存器（CAN_TMPx）（x = 0..2）

地址偏移：0x184 + 0x10 * x

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 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	DLENC[3:0]	数据长度，DLENC[3:0]表示帧内数据长度。

26.4.15. 发送邮箱 data0 寄存器（CAN_TMDATA0x）（x=0..2）

地址偏移：0x188 + 0x10 * x

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

26.4.16. 发送邮箱 data1 寄存器（CAN_TMDATA1x）（x=0..2）

地址偏移：0x18C + 0x10 * x

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

26.4.17. 接收 FIFO 邮箱标识符寄存器（CAN_RFIFOMIx）（x=0,1）

地址偏移：0x1B0 + 0x10 * x

复位值：0xFFFF XXXX

该寄存器只能按字（32 位）访问。



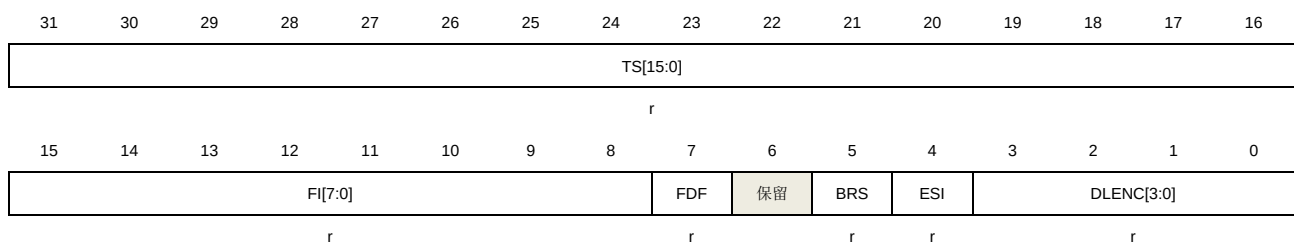
位	区域	说明
31:21	SFID[10:0]/EFID[28:18]	标识符 SFID[10:0]: 标准格式帧标识符 EFID[28:18]: 扩展格式帧标识符
20:16	EFID[17:13]	标识符 EFID[17:13]: 扩展格式帧标识符
15:3	EFID[12:0]	标识符 EFID[12:0]: 扩展格式帧标识符
2	FF	帧格式 0: 标准格式帧 1: 扩展格式帧
1	FT	帧种类 0: 数据帧 1: 遥控帧
0	保留	必须保持复位值。

26.4.18. 接收 FIFO 邮箱属性寄存器（CAN_RFIFOMPx）（x=0,1）

地址偏移: $0x1B4 + 0x10 * x$

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。

26.4.19. 接收 FIFO 邮箱 data0 寄存器（CAN_RFIFOMDATA0x）（x=0,1）

地址偏移：0x1B8 + 0x10 * x

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

26.4.20. 接收 FIFO 邮箱 data1 寄存器 (CAN_RFIFOMDATA1x) (x=0,1)

地址偏移: $0x1BC + 0x10 * x$

复位值: 0xFFFF XXXX

该寄存器只能按字 (32 位) 访问。



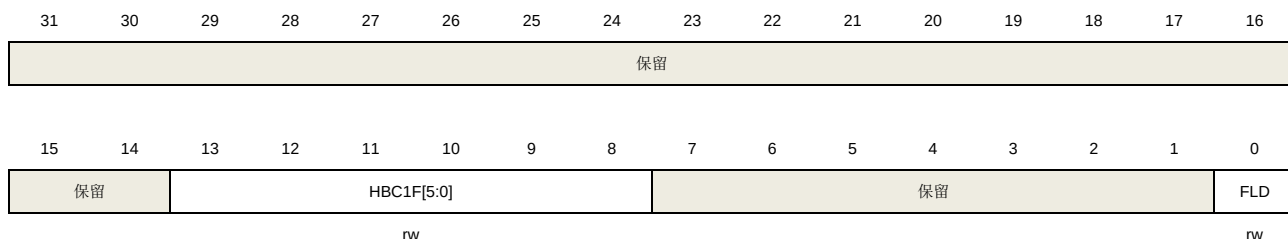
位/位域	名称	描述
31:24	DB7[7:0]	字节7
23:16	DB6[7:0]	字节6
15:8	DB5[7:0]	字节5
7:0	DB4[7:0]	字节4

26.4.21. 过滤器控制寄存器 (CAN_FCTL)

地址: 0x4001 5A00

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

26.4.22. 过滤器模式配置寄存器 (CAN_FMCFG)

地址: 0x4001 5A04

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
				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
FMOD15	FMOD14	FMOD13	FMOD12	FMOD11	FMOD10	FMOD9	FMOD8	FMOD7	FMOD6	FMOD5	FMOD4	FMOD3	FMOD2	FMOD1	FMOD0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:0	FMODx	过滤器模式 0: 掩码模式 1: 列表模式

26.4.23. 过滤器位宽配置寄存器 (CAN_FSCFG)

地址: 0x4001 5A0C

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				FS27	FS26	FS25	FS24	FS23	FS22	FS21	FS20	FS19	FS18	FS17	FS16
				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
FS15	FS14	FS13	FS12	FS11	FS10	FS9	FS8	FS7	FS6	FS5	FS4	FS3	FS2	FS1	FS0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

位/位域	名称	描述
31:28	保留	必须保持复位值。

27:0 FSx 过滤器位宽
0: 16-bit位宽
1: 32-bit位宽

26.4.24. 过滤器关联 FIFO 寄存器 (CAN_FAFIFO)

地址: 0x4001 5A14
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				FAF27	FAF26	FAF25	FAF24	FAF23	FAF22	FAF21	FAF20	FAF19	FAF18	FAF17	FAF16
				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
FAF15	FAF14	FAF13	FAF12	FAF11	FAF10	FAF9	FAF8	FAF7	FAF6	FAF5	FAF4	FAF3	FAF2	FAF1	FAF0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:0	FAFx	过滤器关联FIFO 0: 关联FIFO0 1: 关联FIFO1

26.4.25. 过滤器激活寄存器 (CAN_FW)

地址: 0x4001 5A1C
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	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
FW15	FW14	FW13	FW12	FW11	FW10	FW9	FW8	FW7	FW6	FW5	FW4	FW3	FW2	FW1	FW0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:0	FWx	过滤器激活 0: 没有激活

1: 激活工作

26.4.26. 过滤器(x) 数据(y) 寄存器 (CAN_FxDATAy) (x=0..27, y=0,1)

地址: $0x4001\ 5A40 + 8 * x + 4 * y$

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
FD31	FD30	FD29	FD28	FD27	FD26	FD25	FD24	FD23	FD22	FD21	FD20	FD19	FD18	FD17	FD16
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
FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	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

27. 外部存储器控制器（EXMC）

27.1. 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 PSRAM，NOR Flash，NAND 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。EXMC 的访问空间被划分为许多个块（Bank），每个块支持特定的存储器类型，用户可以通过对 Bank 的控制寄存器配置来控制外部存储器。

27.2. 主要特性

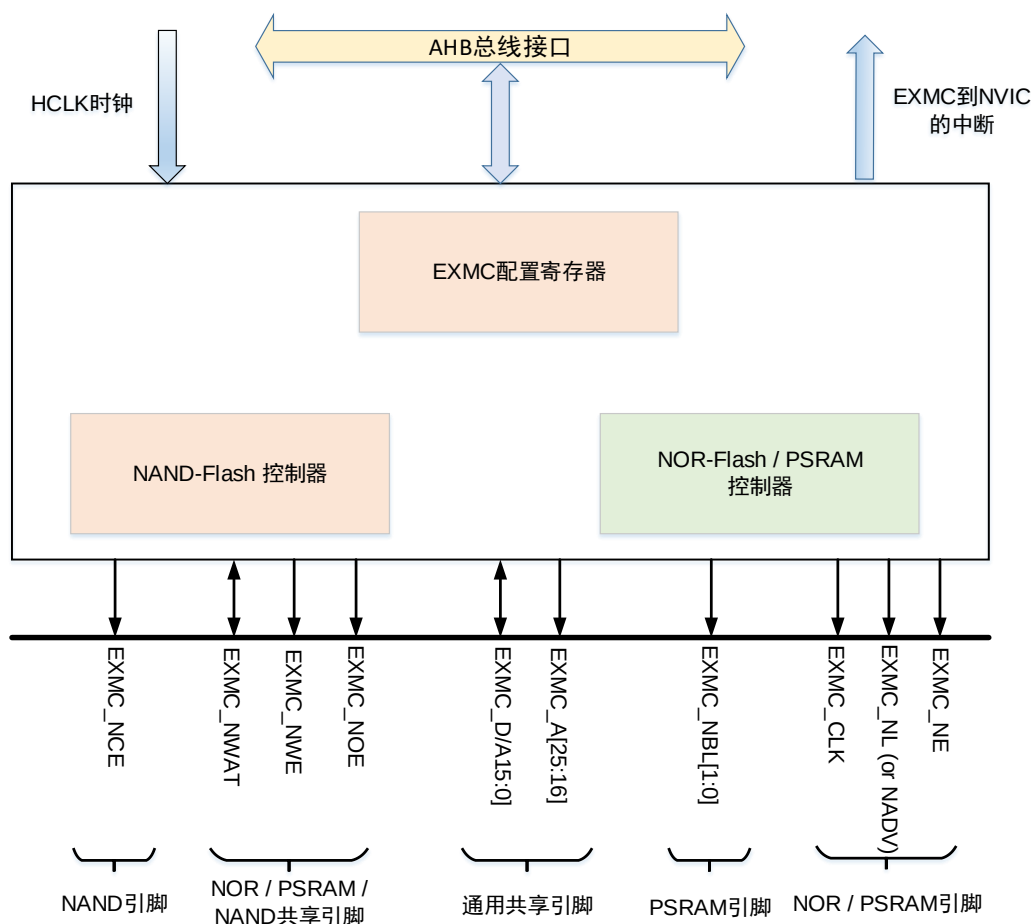
- 支持片外存储器类型：
 - PSRAM;
 - NOR Flash;
 - 8 位或 16 位 NAND Flash;
- AMBA 协议与各种片外存储器协议转换;
- 时序参数可编程可以满足用户特定需求;
- 每个 Bank 有独立的片选信号;
- 对于部分存储器类型支持独立的读写时序;
- 对于 NAND Flash 内置硬件 ECC;
- 支持 8 位，16 位或 32 位总线带宽;
- NOR Flash 和 PSRAM 支持地址总线和数据总线的复用;
- 提供写使能和字节选择信号;
- 当 AMBA 总线宽度与外部存储器数据宽度不同时，会自动分割操作。

27.3. 功能描述

27.3.1. 结构框图

EXMC 由 5 个模块组成：AHB 总线接口，EXMC 配置寄存器，NOR/PSRAM 控制器，NAND 控制器和外部设备接口。AHB 时钟（HCLK）是参考时钟。

图 27-1. 系统架构



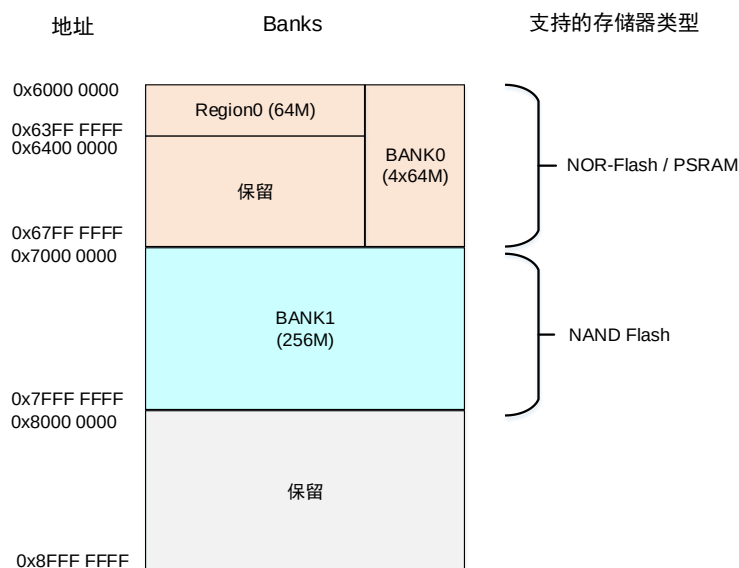
27.3.2. EXMC 访问基本规范

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 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。

27.3.3. 外部设备地址映射

图 27-2. EXMC Bank 划分



EXMC 将外部存储器分成多个 Bank，每个 Bank 占 256M 字节，其中 Bank0 又分为 4 个 Region，每个 Region 占 64M 字节。Bank1 被分成 2 个 Section，分别是属性存储空间和通用存储空间。

每个 Bank 或 Region 都有独立的片选控制信号，也都能进行独立的配置。

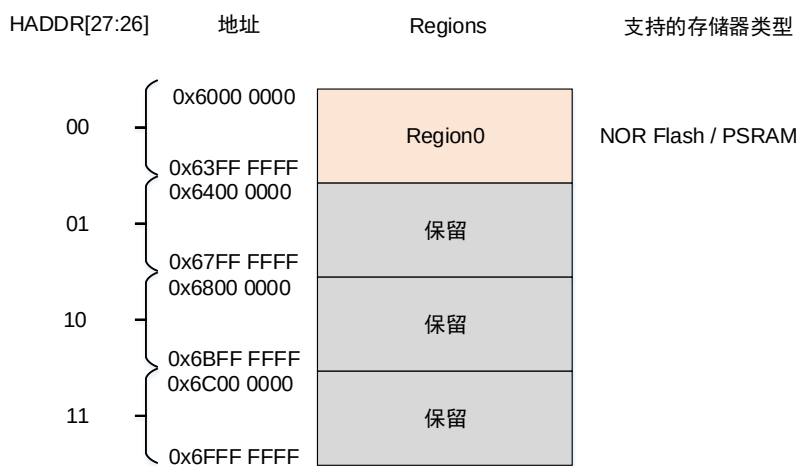
Bank0 用于访问 NOR、PSRAM 设备。

Bank1 用于连接 NAND Flash。

NOR 和 PSRAM 的地址映射

[图 27-3. Bank0 地址映射](#)是 Bank0 四个 Region 的地址映射。AHB 地址线 HADDR[27:26]作为四个 Region 的片选信号。

图 27-3. Bank0 地址映射



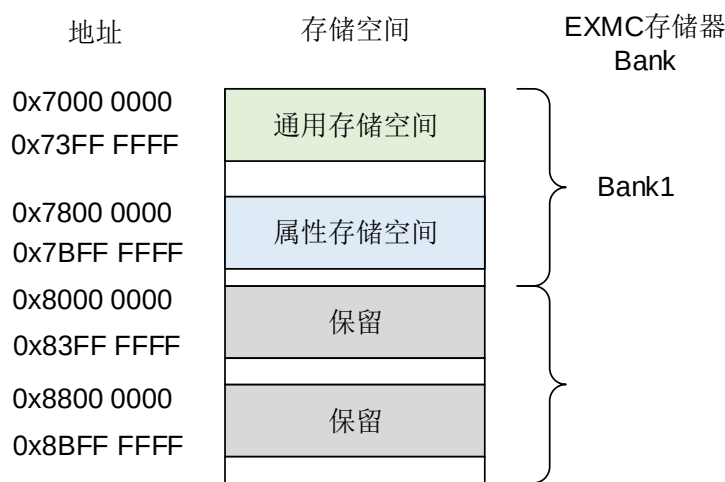
由于 HADDR[25:0]是字节地址，而外部存储器访问有可能不是按字节访问的，所以会出现地址不一致的情况，但 EXMC 能实现对 HADDR 的调整以适应外部存储器的数据宽度。具体规则如下：

- 如果外部存储器的数据宽度是 8 位按字节对齐,EXMC 内部将 HADDR[25:0]与 EXMC_A[25:0]相连，然后 EXMC_A[25:0]与外部存储器的地址线相连；
- 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，EXMC 内部将 HADDR[25:1]与 EXMC_A[24:0]相连，然后 EXMC_A[24:0]与外部存储器的地址线相连。

NAND 地址映射

Bank1 用来访问 NAND Flash，Bank1 如[图 27-4. NAND 地址映射](#)被分为多个存储空间。

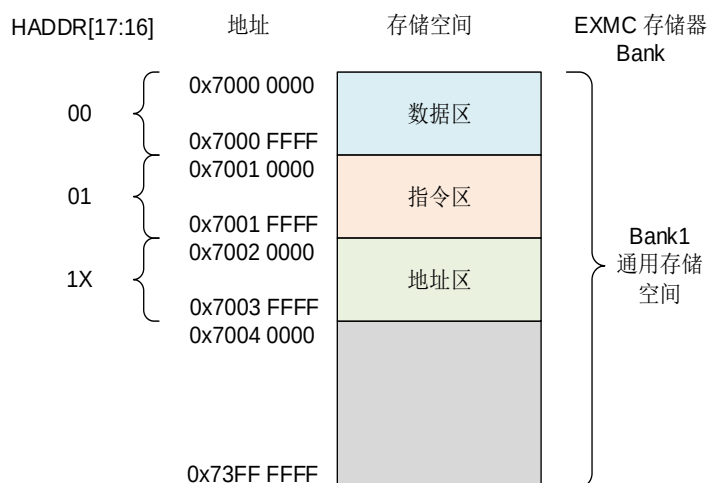
图 27-4. NAND 地址映射



NAND 地址映射

对于 NAND Flash，通用和属性空间又可以细划分为 3 个区域。[图 27-5. Bank1 通用空间](#)为 Bank1 通用存储空间的数据区域，指令区域和地址区域的划分。

图 27-5. Bank1 通用空间



AHB 利用 HADDR[17:16]来实现对以上三个区的选择：

- HADDR[17:16]=00,即选择数据区；
- HADDR[17:16]=01 即选择命令区；
- HADDR[17:16]=1X 即选择地址区。

应用软件使用这 3 个区访问 NAND Flash。操作规则如下：

指令区：指定 NAND Flash 将要执行的指令，软件在命令区写入指令。在指令传输过程中，EXMC 会使能命令锁存信号（CLE），CLE 映射到 EXMC_A[16]。

地址区：指定操作 NAND Flash 的地址，软件在地址区写入地址。在地址传输过程中，EXMC 会使能地址锁存信号（ALE），ALE 映射到 EXMC_A[17]。

数据区：NAND Flash 读写数据，软件在数据区读出或写入数据。当 EXMC 在数据发送模式，软件需要在数据区写入数据，当 EXMC 在数据接收模式，软件需要在数据区读取数据。由于 NAND Flash 会自动累加其内部操作地址，故在读写时不需要软件修改操作地址。

27.3.4. NOR/PSRAM 控制器

EXMC 模块的 NOR/PSRAM 控制器控制 Bank0，它可以支持 NOR Flash、PSRAM。EXMC 对 Bank0 每个 Region 输出一个唯一的片选信号，EXMC_NE，用于 Region 的片选，所有其他的信号都是共享的。Region 有专门的寄存器控制。

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 27-1. 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
复用 EXMC_A[25:16]	输出	异步/同步	地址总线
EXMC_D[15:0]	输入/输出	异步/同步 (复用)	地址/数据总线
EXMC_NE	输出	异步/同步	片选
EXMC_NOE	输出	异步/同步	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 27-2. PSRAM 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
复用 EXMC_A[25:16]	输出	异步/同步	地址总线
EXMC_D[15:0]	输入/输出	异步/同步	地址/数据总线
EXMC_NE	输出	异步/同步	片选
EXMC_NOE	输出	异步/同步	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	地址锁存信号（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

[表 27-3. EXMC 的 Bank0 支持的访问模式](#)列出了 EXMC 对 NOR 和 PSRAM 支持的访问模式。

表 27-3. EXMC 的 Bank0 支持的访问模式

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	

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
	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 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 NBL[1:0]
	同步	W	16	16	
	同步	W	32	16	

NOR Flash/PSRAM 控制时序

EXMC 为 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 27-4. 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	异步读写	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 27-5. 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 AM	0	NOR Flash 数据/地址复用	DSET AHLD ASET BUSLAT	DSET AHLD ASET BUSLAT
同步	模式 SM	0	NOR Flash 数据/地址复用	DLAT CKDIV	DLAT CKDIV

如表 27-5. 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 27-4. 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。

异步访问时序

模式 AM – NOR Flash 地址/数据总线复用

图 27-6. 复用模式读时序

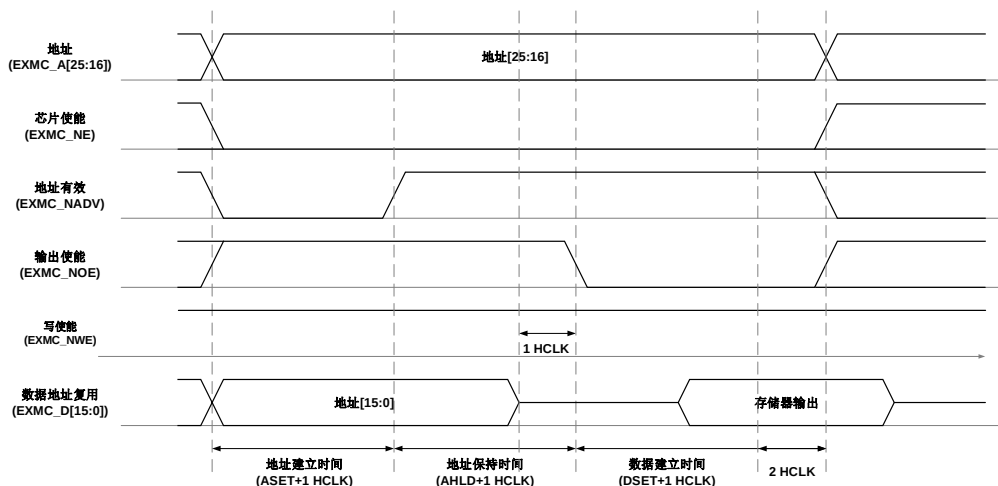


图 27-7. 复用模式写时序

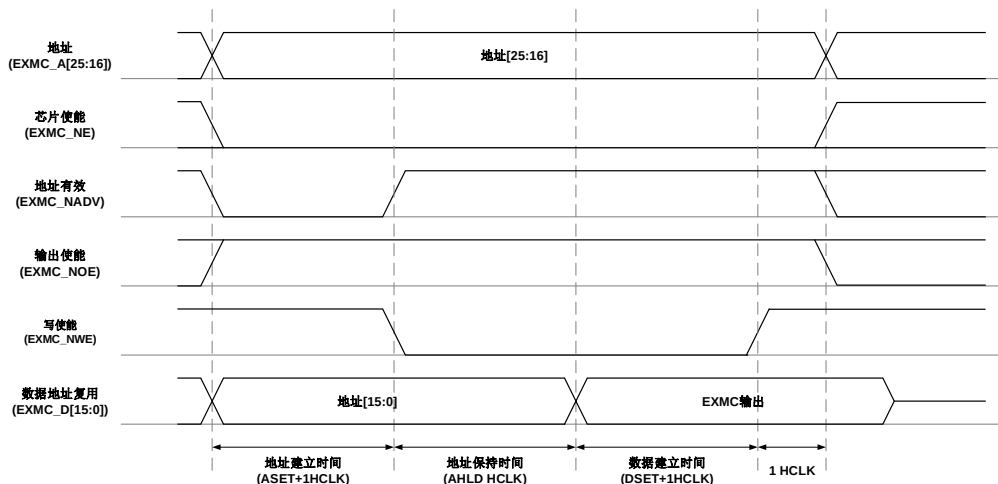


表 27-6. 复用模式相关寄存器配置

位域/位	名称	参考设定值
EXMC_PNCTL		
31-20	保留	0x000
19	SYNCWR	0x0
18-16	CPS	0x0
15	ASYNCWTE	取决于存储器

位域/位	名称	参考设定值
EXMC_PNCTL		
14	保留	0x0
13	NRWTEN	0x0
12	WEN	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
5-4	NRW	取决于存储器
3-2	NRTP	0x2: NOR Flash
1	NRMUX	0x1
0	NRBKEN	0x1
EXMC_PNTCFG		
31-28	保留	0x0
27-24	DLAT	无影响
23-20	CKDIV	无影响
19-16	BUSLAT	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+2HCLK 时钟周期，读操作为 DSET+3HCLK 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

异步通信的等待时间：

等待功能由寄存器 EXMC_PNCTLx 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：

$$T_{DATA\_SETUP} \geq \max T_{WAIT\_ASSERTION} + 4HCLK \quad (27-1)$$

若存储器等待信号与 EXMC_NE 信号对齐：

如果

$$\max T_{WAIT\_ASSERTION} \geq T_{ADDRESS\_PHASE} + T_{HOLD\_PHASE} \quad (27-2)$$

则

$$T_{DATA\_SETUP} \geq (\max T_{WAIT\_ASSERTION} - T_{ADDRESS\_PHASE} - T_{HOLD\_PHASE}) + 4HCLK \quad (27-3)$$

否则

$$T_{DATA\_SETUP} \geq 4HCLK$$

(27-4)

图 27-8. 异步等待有效时的读时序

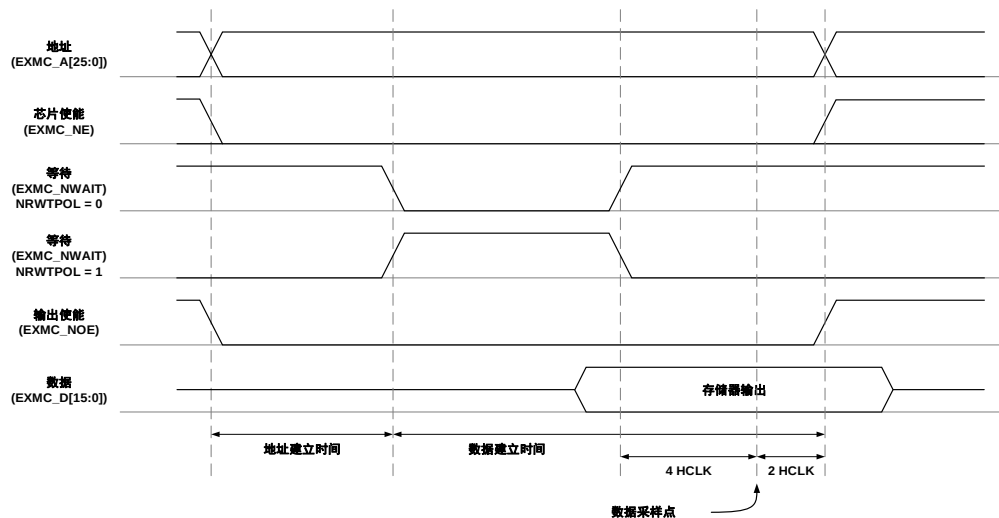
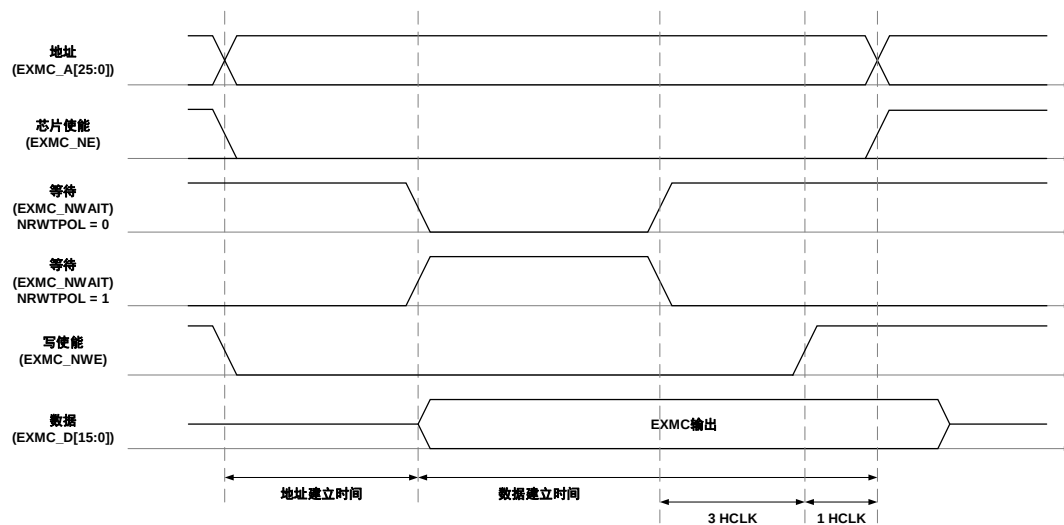


图 27-9. 异步等待有效时的写时序



同步访问时序

同步访问模式中，存储器时钟（EXMC_CLK）与系统时钟（HCLK）关系如下：

$$EXMC\_CLK = \frac{HCLK}{CKDIV+1} \quad (27-5)$$

其中 CKDIV 是同步时钟分频比，通过配置寄存器 EXMC_PNTCFG 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:

NOR 闪存延迟不包含 NADV, 二者之间的关系为:

$$\text{NOR 闪存延迟} = \text{DLAT} + 2 \quad (27-6)$$

NOR 闪存延迟包含 NADV, 二者之间的关系为:

$$\text{NOR 闪存延迟} = \text{DLAT} + 3 \quad (27-7)$$

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_PNCTL 来设置, 位 NRWTEN 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_PNCTL 位 NRWTEN 置 1, 在数据延迟之后检测到 EXMC_NWAIT 信号。如果 EXMC_NWAIT 有效, 在 EXMC_NWAIT 无效之前会一直插入等待时钟。

■ 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_PNCTL 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 27-3. EXMC 的 Bank0 支持的访问模式](#)。

同步复用突发读时序 – NOR,PSRAM(CRAM)

图 27-10. 同步复用突发传输读时序

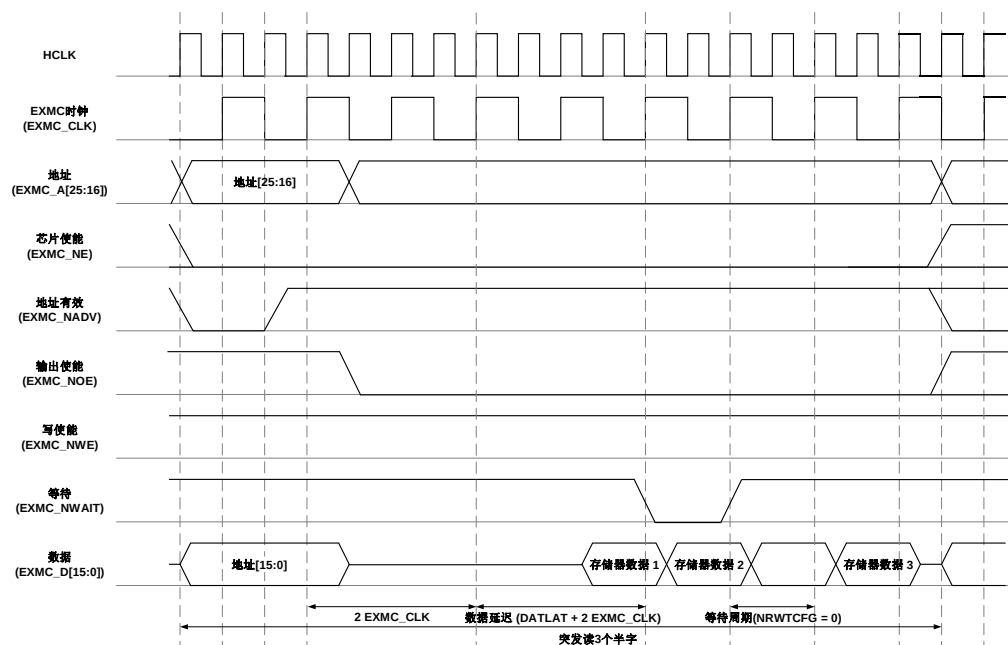


表 27-7. 同步复用模式读时序配置

位域/位	名称	参考设定值
EXMC_PNCTL		
31-20	保留	0x000
19	SYNCWR	无影响
18-16	CPS	0x0
15	ASYNCWTEEN	0x0
14	保留	0x0
13	NRWTEN	取决于存储器
12	WEN	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_PNTCFG(Read)		
31-28	保留	0x0
27-24	DLAT	数据延迟

位域/位	名称	参考设定值
23-20	CKDIV	上图设置: 0x1, EXMC_CLK=2HCLK
19-16	BUSLAT	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

同步复用突发写时序 – NOR,PSRAM(CRAM)

图 27-11. 同步复用突发传输写时序

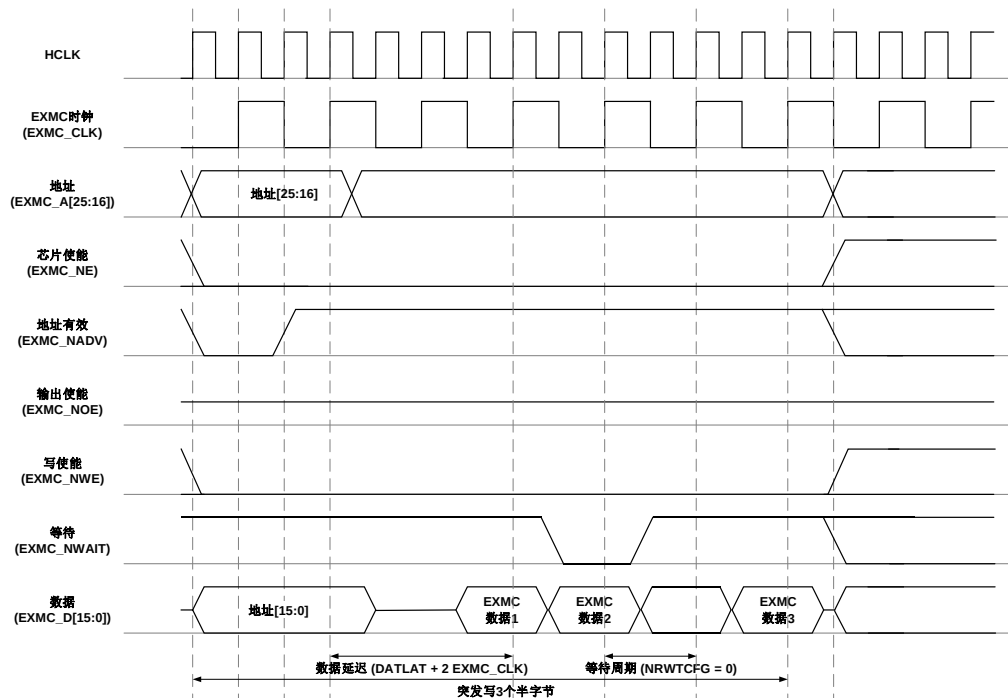


表 27-8. 同步复用模式写时序配置

位域/位	名称	参考设定值
EXMC_PNCTL		
31-20	保留	0x000
19	SYNCWR	0x1, 同步写使能
18-16	CPS	0x0
15	AYSNCWAIT	0x0
14	保留	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

位域/位	名称	参考设定值
7	保留	0x1
6	NREN	取决于存储器
5-4	NRW	0x1
3-2	NRTP	0x1
1	NRMUX	0x1, 取决于用户
0	NRBKEN	0x1
EXMC_PNTCFG(Write)		
31-28	保留	0x0
27-24	DLAT	数据延迟
23-20	CKDIV	上图设置: 0x1, EXMC_CLK=2HCLK
19-16	BUSLAT	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

27.3.5. NAND Flash 控制器

EXMC 模块 Bank1 支持 NAND Flash。对于 Bank1, EXMC 提供独立的寄存器来配置访问时序, 支持 8 位、16 位的 NAND Flash。对于 NAND Flash, EXMC 还提供 ECC 计算模块, 保证数据传输和保存的鲁棒性。

NAND Flash 接口功能

表 27-9. 8 位/16 位 NAND 接口信号描述

EXMC 引脚	传输方向	功能描述
EXMC_A[17]	输出	NAND Flash 地址锁存 (ALE)
EXMC_A[16]	输出	NAND Flash 命令锁存 (CLE)
EXMC_D[7:0]/ EXMC_D[15:0]	输入 /输出	8 位复用, 双向地址/数据总线
		16 位复用, 双向地址/数据总线
EXMC_NCE	输出	片选
EXMC_NOE(NRE)	输出	输出使能
EXMC_NWE	输出	写使能
EXMC_NWAIT	输入	NAND Flash 就绪/忙输入信号 EXMC

支持的存储器访问模式

表 27-10. Bank1 支持的访问模式

存储器	模式	读/写	AHB 传输宽度	注释

8 位 NAND	异步	R	8	分成 2 次 EXMC 访问
	异步	W	8	
	异步	R	16	
	异步	W	16	
	异步	R	32	分成 4 次 EXMC 访问
	异步	W	32	
16 位 NAND	异步	R	8	不支持此操作
	异步	W	8	
	异步	R	16	分成 2 次 EXMC 访问
	异步	W	16	
	异步	R	32	
	异步	W	32	

NAND Flash 的控制时序

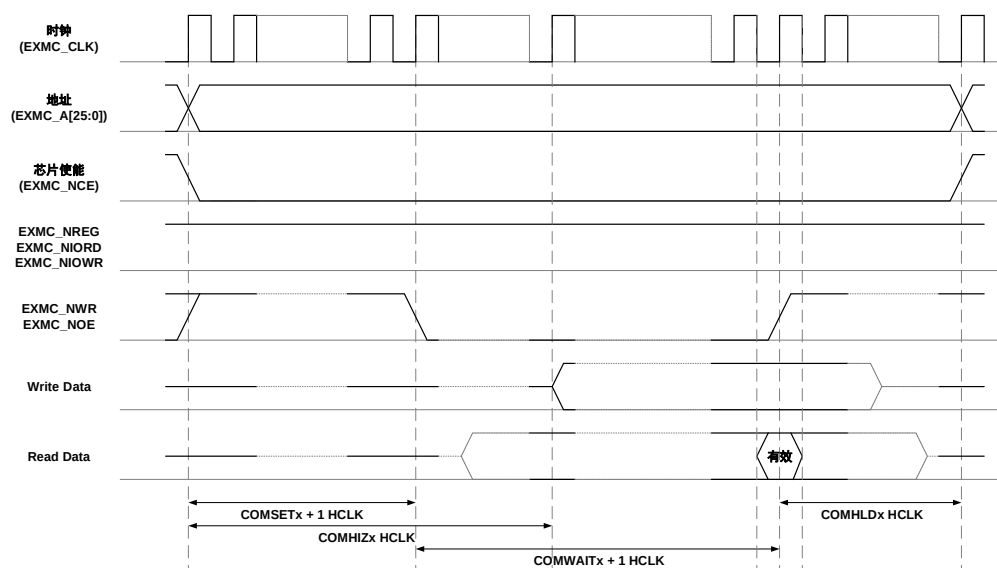
EXMC 能够为 NAND Flash 设备产生合适的时序信号。每个 Bank 都有相应的寄存器来对外部存储器进行管理和控制，EXMC_NCTL、EXMC_NSTAT、EXMC_NCTCFG、EXMC_NATCFG、EXMC_NECC，其中寄存器 EXMC_NCTCFG、EXMC_NATCFG 都可以配置 4 个时序参数，可以根据用户需求和外部存储器的特性来进行相应的配置。

表 27-11. NADN Flash 可编程参数

参数	读/写	单位	功能描述	NAND Flash	
				最小值	最大值
存储器数据总线高阻时间 (HIZ)	W/R	HCLK	启动写操作之后保持数据总线为高阻态的时间	0	255
存储器保持时间 (HLD)	W/R	HCLK	在发送命令结束后保持地址的 (HCLK) 时钟周期数目，写操作时也是数据的保持时间	1	254
存储器等待时间 (WAIT)	W/R	HCLK	发出命令的最短持续时间 (HCLK) 时钟周期数目	2	256
存储器建立时间 (SET)	W/R	HCLK	发出命令之前建立地址的 (HCLK) 时钟周期数目	1	255

[图 27-12. NAND flash 通用空间操作时序](#)显示了在公共存储空间操作中定义的可编程参数。属性存储空间的可编程参数也已定义。

图 27-12. NAND flash 通用空间操作时序



NAND Flash 操作

EXMC 在对 NAND Flash 发送命令或地址时，需要利用其命令锁存信号（A[16]）或地址锁存信号（A[17]）这两条地址线，即 CPU 需要在特定的地址进行写操作。

示例：NAND Flash 读操作步骤：

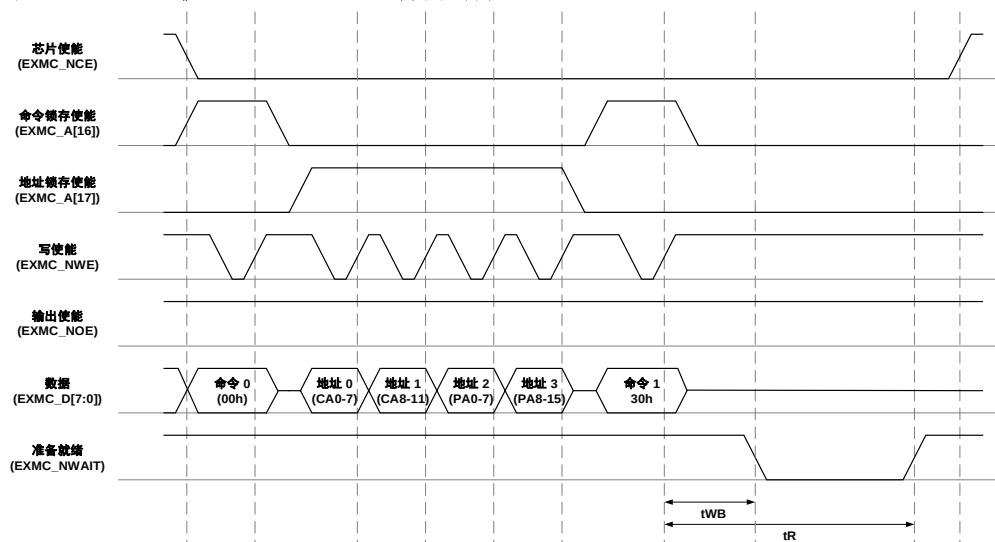
1. 配置 EXMC_NCTL、EXMC_NCTCFG，若需要预等待功能，还需配置 EXMC_NATCFG；
2. 往通用空间写入 NAND Flash 读数据命令，即在 EXMC_NCE 和 EXMC_NWE 有效期间，EXMC_CLE（A[16]）变为有效电平（高），则被 NAND 认为写入命令；
3. 往通用空间写入读操作的起始地址，即在 EXMC_NCE 和 EXMC_NWE 有效期间，EXMC_ALE（A[17]）变为有效电平（高），则被 NAND 认为写入地址；
4. 等待 NAND 就绪信号，NAND 控制器会在这期间将和 EXMC_NCE 一直保持有效；
5. 从通用空间的数据区逐字节的读出数据；
6. 在不写入新的命令和地址，可以自动读出 NAND 下一页数据；或转到 3）写入新的地址进行下一页的读取；或转到 2）写入新的命令和地址。

NAND Flash 预等待功能

某些 NAND Flash 要求在输入最后一个地址字节后，控制器等待 NAND Flash 就绪，并且还有一些对 EXMC_NCE 敏感型的 NAND Flash 还要求在其就绪前 NCE 必须保持有效。

下面以 TOSHIBA128M*8bit NADN Flash 为例：

图 27-13. NCE 敏感 NAND Flash 访问时序



1. 往 NAND 的通用空间命令区写入命令 CMD0
2. 往 NAND 的通用空间地址区写入操作地址 ADD0
3. 往 NAND 的通用空间地址区写入操作地址 ADD1
4. 往 NAND 的通用空间地址区写入操作地址 ADD2
5. 往 NAND 的通用空间地址区写入操作地址 ADD3
6. 往 NAND 的属性空间命令区写入命令 CMD1

在 6) 中写命令操作, EXMC 使用的是寄存器 EXMC_NATCFG 定义的时序。经过 ATTHLD 时间后, NAND Flash 等待 EXMC_NWAIT 信号, ATTHLD 要大于 t_{WB} (EXMC_NWE 高到 EXMC_NWAIT 低)。对于那些对 EXMC_NCE 敏感的 NAND Flash,

对于那些对片选信号敏感的 NAND Flash, 在地址字节之后的第一个地址字节输入后, 一直到 B/NB 就绪状态到来的这段时间中, 要求片选信号 NCE 一直保持低电平。这里可以通过配置属性存储空间的 ATHT 的值来满足 t_{WB} 的时序, 这样 CPU 只有在地址字节之后写入第一个命令字节时才使用属性存储空间的时序, 而在其他时候都使用通用存储空间的时序。

NAND Flash 的 ECC 计数模块

EXMC 模块中的 Bank1 有一个 ECC 计算的硬件模块, 用户可以根据 EXMC_NCTL 中的 ECCSZ 来选择 ECC 计算的页面大小, 通过 ECC 计算可以矫正 1 个 bit 的错误并且能检测 2 个 bit 的错误。

当 NAND 存储器块使能, ECC 模块就会检测 D[15:0] 以及 EXMC_NCE、EXMC_NWE 信号。当已经完成 ECCSZ 大小字节的读写操作时, 软件必须读出 EXMC_NECC 中的结果值。如果需要再次开始 ECC 计算, 软件需要先将 EXMC_NECC 中 ECCEN 清 0 来清除 EXMC_NCTL 中的值, 再将 ECCEN 置 1 来重新启动 ECC 计算。

27.4. EXMC 寄存器

27.4.1. NOR/PSRAM 控制器寄存器

PSRAM/NOR Flash 控制寄存器 (EXMC_PNCTL)

偏移地址：0x00

复位值：0x0000 30DB（对于 region0）

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												SYNC WR	CPS[2:0]		
												rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASYNC WAIT	保留	NRWT EN	WREN	NRWT CFG	WRAPEN	NRWT POL	SBR STEN	保留	NR EN	NRW[1:0]		NRTP[1:0]		NR MUX	NRBK EN
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	ASYNCWAIT	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	保留	必须保持复位值。
13	NRWTEN	NWAIT信号使能 对于存储器的突发模式访问，该位使能/禁用等待状态插入NWAIT信号功能 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	非对齐成组模式使能 0: 禁止非对齐成组操作 1: 允许非对齐成组操作
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: 保留 01: PSRAM（CRAM） 10: NOR Flash(region0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

PSRAM/NOR Flash 时序寄存器 (EXMC_PNTCFG)

偏移地址：0x04

复位值：0x0FFF FFFF

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。
27:24	DLAT[3:0]	NOR Flash数据延时，仅在同步模式有效 0x0：第一数据的保持时间为2个EXMC_CLK时钟周期 0x1：第一数据的保持时间为3个EXMC_CLK时钟周期 0xF：第一数据的保持时间为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小时间 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	AHL D[3:0]	异步地址保持时间 该位域设置地址保持时间，仅在复用模式有效

		0x0: 保留
		0x1: 地址建立时间=2个HCLK
		
		0xF: 地址建立时间=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

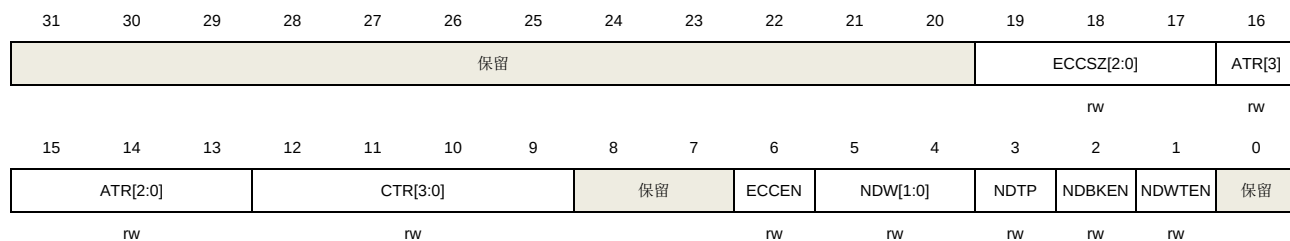
27.4.2. NAND Flash 控制器寄存器

NAND Flash 控制寄存器 (EXMC_NCTL)

偏移地址: 0x60

复位值: 0x0000 0018

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。
19:17	ECCSZ[2:0]	ECC块大小 000: 256字节 001: 512字节 010: 1024字节 011: 2048字节 100: 4096字节 101: 8192字节
16:13	ATR[3:0]	ALE至RE的延迟 0x0: 1个HCLK 0xF: 16个HCLK

12:9	CTR[3:0]	CLE至RE的延迟 0x0: 1个HCLK 0x1: 2个HCLK 0xF: 16个HCLK
8:7	保留	必须保持复位值。
6	ECCEN	ECC使能 0: 关闭ECC，并复位EXMC_NECCx 1: 打开ECC
5:4	NDW[1:0]	外部存储器数据宽度 00: 8位 01: 16位 其他: 保留
3	NDTP	外部存储器的类型 0: 保留 1: NAND Flash
2	NDBKEN	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
1	NDWTEN	NWAIT信号使能位 0: 关闭等待功能 1: 使能等待功能
0	保留	必须保持复位值。

NAND Flash 状态寄存器 (EXMC_NSTAT)

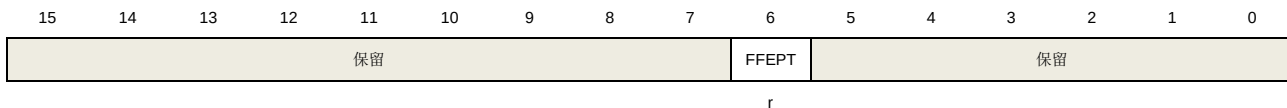
偏移地址: 0x64

复位值: 0x0000 0042 (对于 bank1)

该寄存器只能按字 (32 位) 访问。

该寄存器包含一个 FIFO 空状态位，该位主要用于 ECC。当写外部存储器时，FIFO 可以容纳来自 AHB 访问的 2 个字，使得 AHB 总线可以暂时被释放而用于其他外设。ECC 计算是基于从 FIFO 传递的数据。为了得到正确的 ECC 值，用户应该在 FIFO 空状态标志位为 1 时才去读 ECC 寄存器。





位/位域	名称	描述
31:7	保留	必须保持复位值。
6	FFEPT	FIFO空标志位 0: FIFO非空 1: FIFO空
5:0	保留	必须保持复位值。

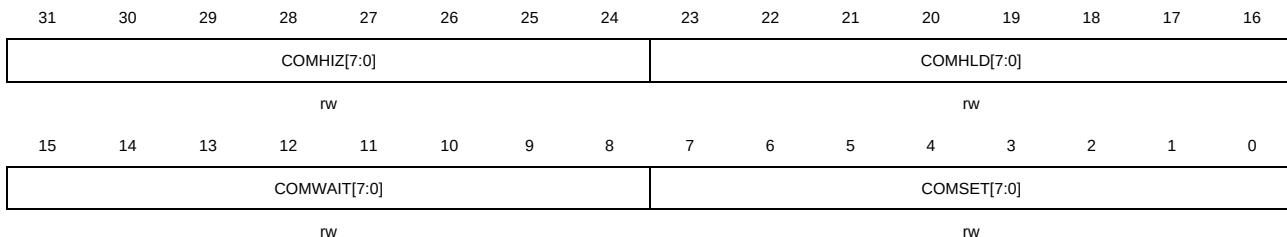
NAND Flash 通用空间时序寄存器 (EXMC_NCTCFG)

偏移地址: 0x68

复位值: 0xFCFC FCFC

该寄存器只能按字（32 位）访问。

这些操作适用于以下类型的外部存储器的通用存储空间 16 位的 NAND Flash。



位/位域	名称	描述
31:24	COMHIZ[7:0]	通用空间数据总线的高阻时间 定义在通用空间进行写操作后数据总线保持高阻态时间 0x00: 1个HCLK 0xFE: 255个HCLK 0xFF: 保留
23:16	COMHLD[7:0]	通用空间的保持时间 在发送地址后的地址保持时间，在写操作时，也作为数据信号保持的时间 0x00: 保留 0x01: 1个HCLK 0xFE: 254个HCLK 0xFF: 保留

15:8	COMWAIT[7:0]	通用空间的等待时间 定义了保持命令的最小时间 0x00: 保留 0x01: 2个HCLK (加上NWAIT时钟周期) 0xFE: 255个HCLK (加上NWAIT时钟周期) 0xFF: 保留
7:0	COMSET[7:0]	通用空间的建立时间 定义地址信号的建立时间 0x00: 1个HCLK 0xFE: 255个HCLK 0xFF: 保留

NAND Flash 属性空间时序寄存器 (EXMC_NATCFG)

偏移地址: 0x6C

复位值: 0xFCFC FCFC

该寄存器只能按字（32 位）访问。

用于对 NAND Flash 属性存储空间的 8 位访问，适用于最后一个地址或命令写入访问，如果需要应用其他时序。



位/位域	名称	描述
31:24	ATTHIZ[7:0]	属性空间数据总线的高阻时间 定义在属性空间进行写操作后数据总线保持高阻态时间 0x00: 1个HCLK 0xFE: 255个HCLK 0xFF: 保留
23:16	ATTHLD[7:0]	属性空间的保持时间 在发送地址后的地址保持时间，在写操作时，也作为数据信号保持的时间 0x00: 保留 0x01: 1个HCLK

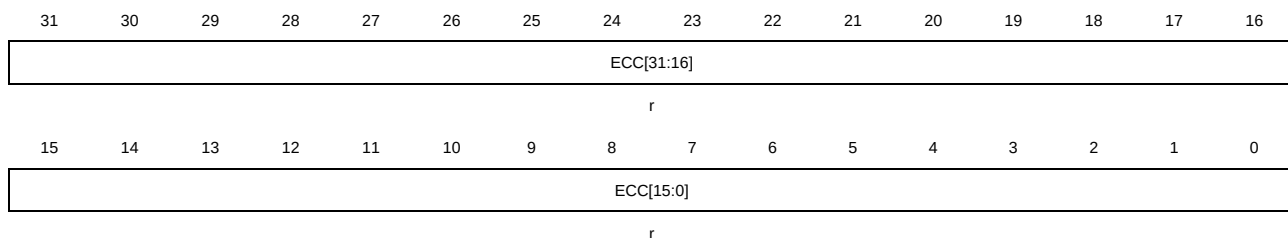
		
		0xFE: 254个HCLK
		0xFF: 保留
15:8	ATTWAIT[7:0]	属性空间的等待时间 定义了保持命令的最小时间 0x00: 保留 0x01: 2个HCLK (加上NWAIT时钟周期) 0xFE: 255个HCLK (加上NWAIT时钟周期) 0xFF: 保留
7:0	ATTSET[7:0]	属性空间的建立时间 定义地址信号的建立时间 0x00: 1个HCLK 0xFE: 255个HCLK 0xFF: 保留

NAND Flash ECC 结果寄存器 (EXMC_NECC)

偏移地址: 0x74

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	ECC[31:0]	ECC计算结果 <table border="1"> <thead> <tr> <th>ECCSZ[2:0]</th><th>NAND Flash 页大小</th><th>ECC 位</th></tr> </thead> <tbody> <tr> <td>0b000</td><td>256</td><td>ECC[21:0]</td></tr> <tr> <td>0b001</td><td>512</td><td>ECC[23:0]</td></tr> <tr> <td>0b010</td><td>1024</td><td>ECC[25:0]</td></tr> <tr> <td>0b011</td><td>2048</td><td>ECC[27:0]</td></tr> <tr> <td>0b100</td><td>4096</td><td>ECC[29:0]</td></tr> <tr> <td>0b101</td><td>8192</td><td>ECC[31:0]</td></tr> </tbody> </table>	ECCSZ[2:0]	NAND Flash 页大小	ECC 位	0b000	256	ECC[21:0]	0b001	512	ECC[23:0]	0b010	1024	ECC[25:0]	0b011	2048	ECC[27:0]	0b100	4096	ECC[29:0]	0b101	8192	ECC[31:0]
ECCSZ[2:0]	NAND Flash 页大小	ECC 位																					
0b000	256	ECC[21:0]																					
0b001	512	ECC[23:0]																					
0b010	1024	ECC[25:0]																					
0b011	2048	ECC[27:0]																					
0b100	4096	ECC[29:0]																					
0b101	8192	ECC[31:0]																					

28. 通用串行总线全速接口（USBFS）

28.1. 概述

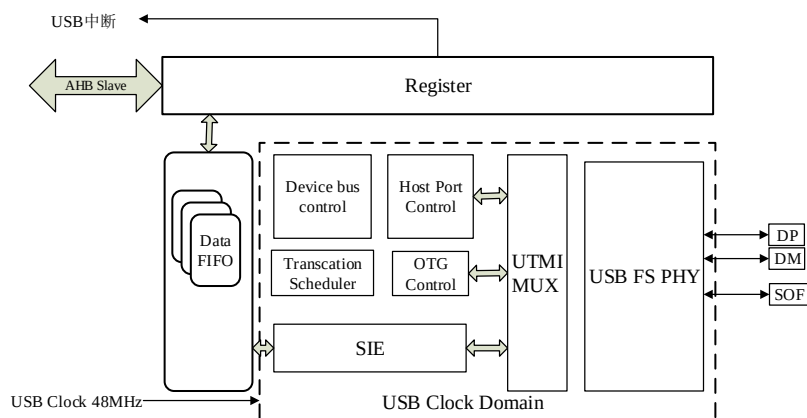
USB全速（USBFS）控制器为便携式设备提供了一套USB互联解决方案。USBFS支持主机模式和设备模式。USBFS包含了一个内部的全速USB PHY，并且不再需要外部PHY芯片。USBFS可以支持USB 2.0协议所定义的所有四种传输方式（控制传输、批量传输、中断传输和同步传输）。

28.2. 主要特性

- 支持USB 2.0全速（12Mb/s）/低速（1.5Mb/s）主机模式；
- 支持USB 2.0全速（12Mb/s）设备模式；
- 支持所有的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；
- 在主机模式下，需要外部部件为连接的USB设备提供电源。

28.3. 结构框图

图 28-1. USBFS 结构框图



28.4. 信号线描述

表 28-1. USBFS 信号线描述

I/O 端口	类型	描述
DM	输入/输出	差分信号 D-端口
DP	输入/输出	差分信号 D+端口
SOF	输出	USB SOF 信号输出

28.5. 功能描述

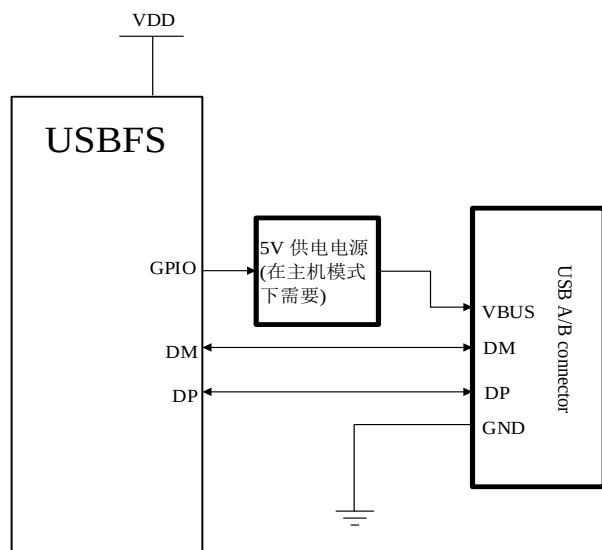
28.5.1. USBFS 时钟及工作模式

USBFS 可以作为一个主机、一个设备，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。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）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28-2. 在主机或设备模式下连接示意图](#)所示。

图 28-2. 在主机或设备模式下连接示意图



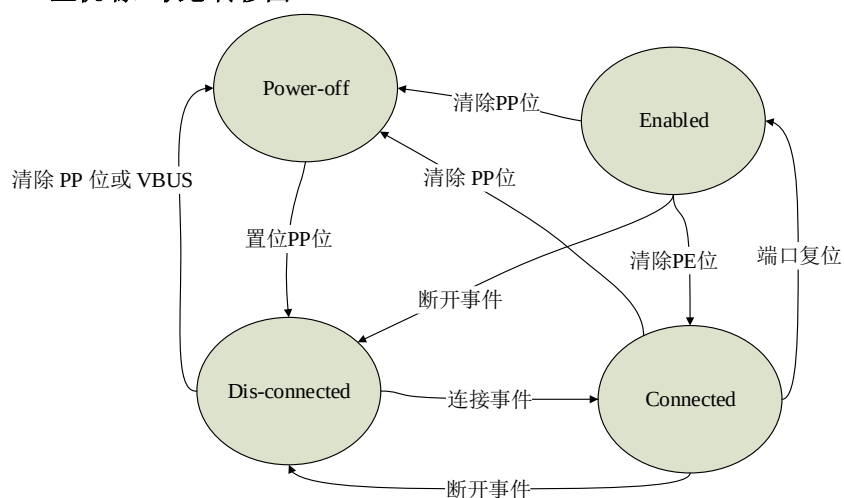
当 USBFS 工作在主机模式下时（FHM 控制位置位、FDM 控制位清除），VBUS 为 USB 协议所定义的 5V 电源检测引脚。内部 PHY 不能提供 5V VBUS 电源，仅在 VBUS 信号线上具有电压比较器和充电、放电电路。所以，如果应用需要提供 VBUS 电源，那么则需要一个外部的供电电源 IC。在主机模式下，USBFS 和 USB 接头之间的 VUBS 连接可以被忽略，这是由于 USBFS 并不检测 VBUS 引脚的电平状态，并假定 5V 供电电源一直存在。

28.5.2. 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 28-3. 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 12 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。

28.5.3. 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，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 0b '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 3 毫秒无变化，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。

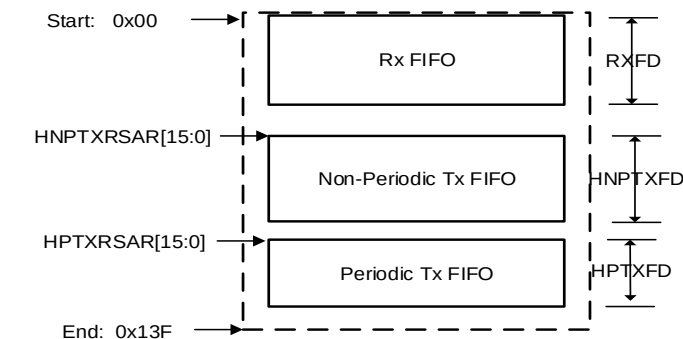
28.5.4. 数据 FIFO

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 28-4. 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 28-4. 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 28-5. 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。

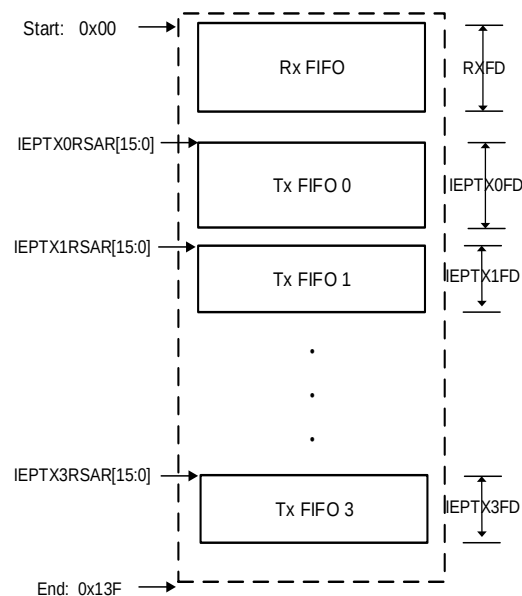
图 28-5. 主机模式 FIFO 访问寄存器映射表

1000h-1FFFh	CH0 FIFO Write/Read
2000h-2FFFh	CH1 FIFO Write/Read
⋮	
8000h-8FFFh	CH7 FIFO Write/Read

设备模式

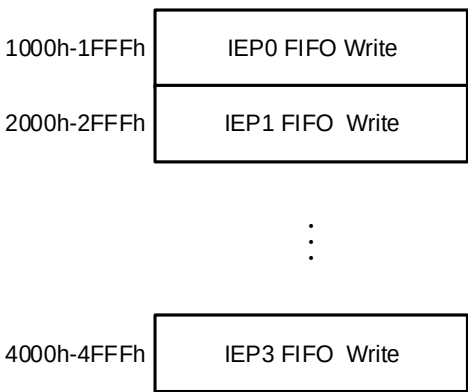
在设备模式下，数据 FIFO 分为多个部分，其中包含 1 个 Rx FIFO 和 4 个 Tx FIFO，每个 Tx FIFO 对应着一个 IN 端点，所有的 OUT 端点通过共享 Rx FIFO 接收数据包。通过寄存器 USBFS_GRFLEN 和 USBFS_DIEPxTFLEN (x=0...3)，程序可配置数据 FIFO 的大小和起始偏移地址。[图 28-6. 设备模式 FIFO 空间](#)所描述的是 SRAM 中各 FIFO 的结构，图中的数值是以按照 32 位写的。

图 28-6. 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 28-7. 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 28-7. 设备模式 FIFO 访问寄存器映射表



28.5.5. 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），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；

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；

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；

4. 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的最后数据包顶部，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），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_DIEPCTL 或 USBFS_DOEPCTL；
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_DIEPLEN 或 USBFS_DOEPLEN，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；

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_DIEPCTL 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；

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；

4. 置位 USBFS_DIEPCTL 或 USBFS_DOEPCTL 寄存器 EPEN 位使能端点。

端点除能顺序：

当 USBFS_DIEPCTL 或 USBFS_DOEPCTL 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_DIEPLEN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_DIEPLEN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的最后的數據包的顶部，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。

28.6. 中断

USBFS 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 28-2. USBFS 全局中断](#)。

表 28-2. USBFS 全局中断

中断标志	描述	运行模式
SESIF	会话中断	主机或设备模式
DISCIF	断开连接中断标志	主机模式
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
PXNCIF/ ISOONCIF	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
OEIF	OUT 端点中断标志	设备模式
IEPIF	IN 端点中断标志	设备模式
EOPFIF	周期性帧尾中断标志	设备模式
ISOOPDIF	同步 OUT 丢包中断标志	设备模式
ENUMF	枚举完成	设备模式
RST	USB 复位	设备模式
SP	USB挂起	设备模式
ESP	早挂起	设备模式
GONAK	全局OUT NAK有效	设备模式
GNPINAK	全局非周期IN NAK有效	设备模式
NPTXFEIF	非周期Tx FIFO空中断标志	主机模式
RXFNEIF	Rx FIFO非空中断标志	主机或设备模式
SOF	帧首	主机或设备模式
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。

28.7. USBFS 寄存器

USBFS 基地址：0x5000 0000

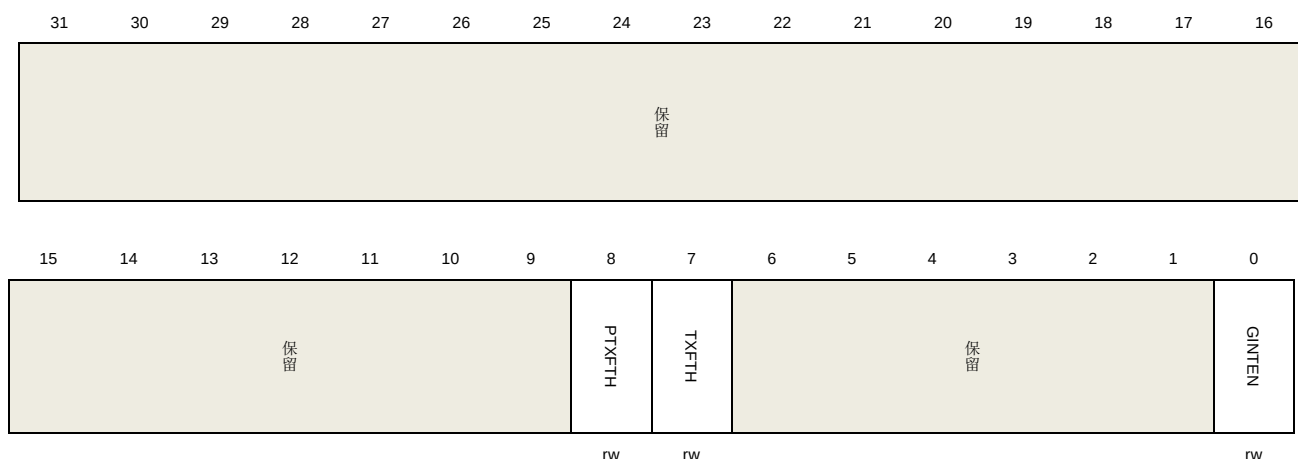
28.7.1. 全局控制与状态寄存器组

全局 AHB 控制和状态寄存器（USBFS_GAHBCS）

地址偏移：0x0008

复位值：0x0000 0000

该寄存器只能按字（32 位）访问



位/位域	名称	描述
31:9	保留	必须保持复位值。
8	PTXFTH	周期性Tx FIFO阈值 0: 当周期性发送FIFO半空时，将触发PTXFEIF标志位 1: 当周期性发送FIFO全空时，将触发PTXFEIF标志位 注意：只在主机模式下访问
7	TXFTH	Tx FIFO 阈值 设备模式： 0: 当IN端点发送FIFO半空时，将触发TXFEIF标志位 1: 当IN端点发送FIFO全空时，将触发TXFEIF标志位 主机模式： 0: 当非周期性发送FIFO半空时，将触发NPTXFEIF标志位 1: 当非周期性发送FIFO全空时，将触发NPTXFEIF标志位
6:1	保留	必须保持复位值。

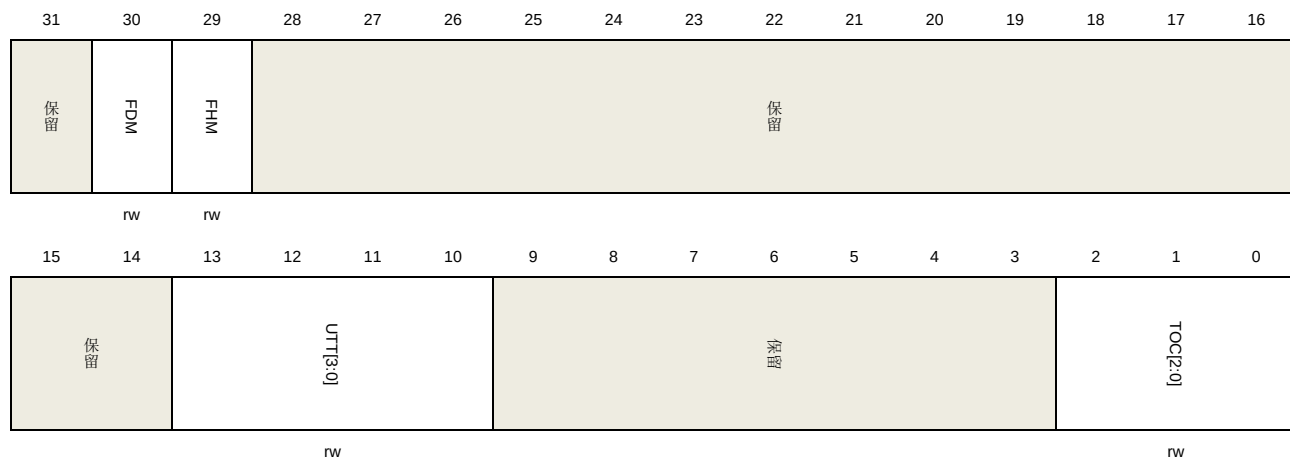
0	GINTEN	全局中断使能 0: 全局中断不使能 1: 全局中断使能 注意: 在主机和设备模式下, 均可访问

全局 USB 控制和状态寄存器 (USBFS_GUSBCS)

地址偏移: 0x000C

复位值: 0x0000 0880

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	强制设备模式 通过置位该控制位, 可强制USB内核为设备模式 0: 正常模式 1: 设备模式 设置该控制位后, 应用必须等待至少25ms, 让变化产生作用。 注意: 在设备和主机模式下, 均可访问。
29	FHM	强制主机模式 通过置位该控制位, 可强制USB内核为主机模式 0: 正常模式 1: 主机模式 设置该控制位后, 应用必须等待至少25ms, 让变化产生作用。 注意: 在设备和主机模式下, 均可访问
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

9:3	保留	必须保持复位值。
2:0	TOC[2:0]	超时校准 当等待一个包时，USBFS需要使用USB2.0协议中需要的超时数值。应用可以使用TOC[2:0]增加该数值（以PHY时钟为单位）。PHY时钟频率为48MHz。

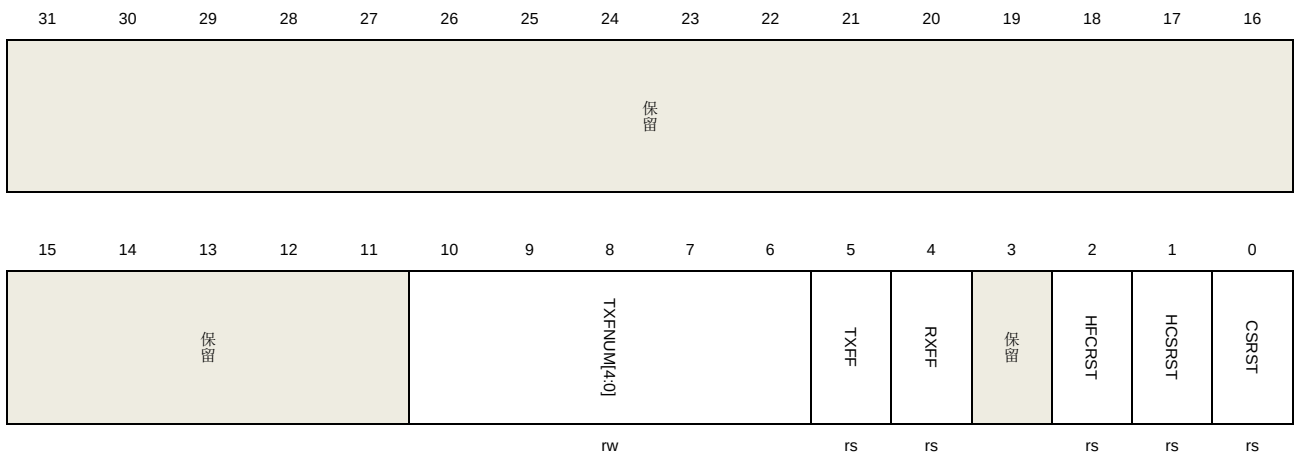
全局复位控制寄存器（USBFS_GRSTCTL）

地址偏移：0x0010

复位值：0x8000 0000

应用通过该寄存器来复位内核的不同硬件特性。

该寄存器只能按字（32 位）访问



位/位域	名称	描述
31:11	保留	必须保持复位值。
10:6	TXFNUM[4:0]	Tx FIFO数目 当本寄存器中TXFF控制位置位时，该标志位决定那个Tx FIFO会被冲刷 主机模式： 00000： 仅非周期性Tx FIFO被冲刷 00001： 仅周期性Tx FIFO被冲刷 1xxxx： 周期性和非周期性Tx FIFO均被冲刷 其他： 没有数据被冲刷 设备模式： 00000： 仅Tx FIFO0被冲刷 00001： 仅Tx FIFO1被冲刷 ...

		00011: 仅Tx FIFO3被冲刷 1XXXX: 所有的Tx FIFO均被冲刷 其他: 没有数据被冲刷
5	TXFF	Tx FIFO冲刷控制位 应用通过置位该控制位来冲刷Tx FIFO数据, 并且TXFNUM[4:0]决定冲刷的FIFO数目。当冲刷完成后, 硬件自动清除该控制位。置位该控制位后, 应用应该等待该控制位清除, 并且, 在此之前USBFS不应有其他操作。 注意: 在设备和主机模式下, 均可访问
4	RXFF	Rx FIFO冲刷控制位 应用通过置位该控制位来冲刷Rx FIFO数据。当冲刷完成后, 硬件自动清除该控制位。置位该控制位后, 应用应该等待该控制位清除, 并且, 在此之前USBFS不应有其他操作。 注意: 在设备和主机模式下, 均可访问
3	保留	必须保持复位值。
2	HFCRST	主机帧计数器复位 应用通过置位该控制位来复位USBFS内的帧计数器。该控制位置位后, 接下来SOF的帧计数器将变为0。当复位操作完成后, 硬件自动清除该控制位。置位该控制位后, 应用应该等待该控制位清除, 并且, 在此之前USBFS不应有其他操作。 注意: 仅在主机模式下访问
1	HCSRST	HCLK软件复位 应用通过置位该控制位来复位ABH时钟域电路 在复位操作完成后, 硬件自动清除该控制位。置位该控制位后, 应用应该等待该控制位清除, 并且, 在此之前USBFS不应有其他操作。 注意: 在设备和主机模式下, 均可访问
0	CSRST	USB内核软件复位 复位AHB和USB时钟域电路, 以及大多数的寄存器。

全局中断标志寄存器 (USBFS_GINTF)

地址偏移: 0x0014

复位值: 0x0400 0021

该寄存器只能按字(32位)访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WkupIF	保留	DISCIF	保留	PTXFEIF	HCFE	HPIF	保留	PXNCF/ ISONCF	ISONCF	OEPIF	IEPIF	保留			
rc_w1		rc_w1		r	r	r		rc_w1	rc_w1	r	r				

15	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOPHF	ISOOPDIF	ENUMF	RST	SD	ESP	保留	GONAK	GAPINAK	NPTXFEIF	RXENEIF	SOF	保留	MFIF	COPM
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1		r	r	r	r	rc_w1		rc_w1	r

位/位域	名称	描述
31	WKUPIF	唤醒中断标志位 当在USB总线上检测到一个恢复信号（在设备模式下）或者一个远程唤醒信号（在主机模式下），硬件将置位该中断标志位。 注意： 在设备和主机模式下，均可访问
30	保留	必须保持复位值。
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
28:27	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_HPCS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	OEPIF	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	IEPIF	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	保留	必须保持复位值。
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	保留	DISCIE	保留		PTXFEIE	HCIE	HPIE	保留		ISOONCIE	PXNCIE/ ISOONCIE	ISOINCIE	OEP1E	IEP1E	保留
r/w		r/w			r/w	r/w	r			r/w	r/w	r/w	r/w		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOP1E	ISOOP1E	ENDMP1E	RST1E	SPIE	ESPIE	保留		GONAKIE	GNPINAKIE	NPTXFEIE	RXFNEIE	SOF1E	保留	MF1E	保留
r/w	r/w	r/w	r/w	r/w	r/w			r/w	r/w	r/w	r/w	r/w		r/w	

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	保留	必须保持复位值。
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
28:27	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	保留	必须保持复位值。
1	MFIE	模式错误中断使能 0: 禁止模式错误中断 1: 使能模式错误中断 注意: 在设备模式下与主机模式下, 均可访问
0	保留	必须保持复位值。

全局接收状态读取/接收状态读取和弹出寄存器 (USBFS_GRSTATR/USBFS_GRSTATP)

读地址偏移: 0x001C

弹出地址偏移: 0x0020

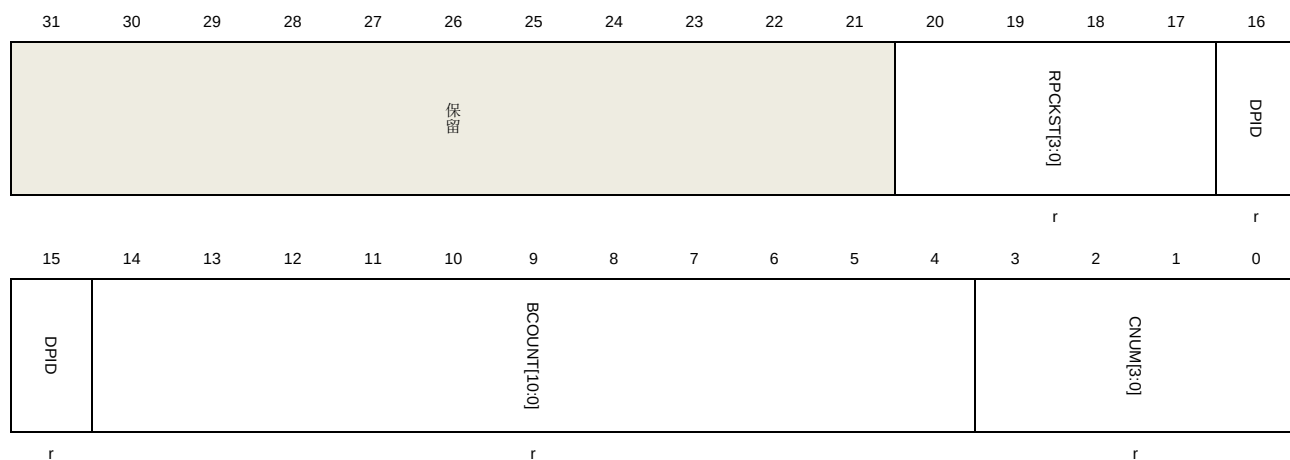
复位值: 0x0000 0000

对接收状态寄存器的读操作, 将返回接收 FIFO 中顶部的条目。对接收状态读取和弹出寄存器的读操作, 将额外的弹出 Rx FIFO 的顶部条目。

在主机模式和设备模式下, Rx FIFO 中的条目具有不同的含义。当全局中断标志寄存器 (USBFS_GINTF) 中的接收 FIFO 非空中断标志位 (RXFNEIF) 置位后, 软件应该读取该寄存器。

该寄存器只能按字(32 位)访问

主机模式:



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

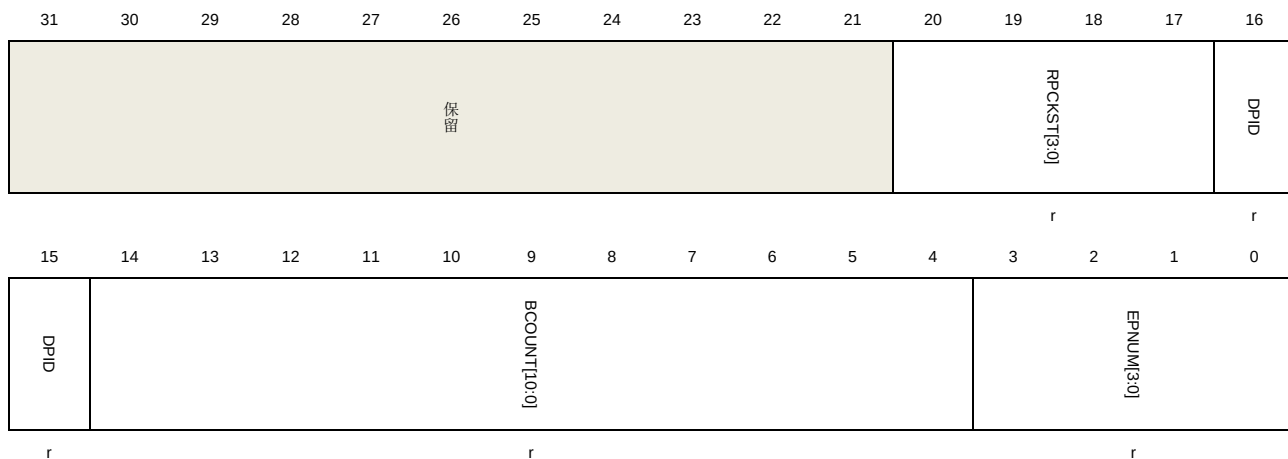
0011: IN传输完成（如果取出，触发一个中断）
 0101: 数据翻转错误（如果取出，触发一个中断）
 0111: 通道中止（如果取出，触发一个中断）
 其他: 保留

16:15 DPID[1:0] 数据PID
 接收包的数据PID
 00: DATA0
 10: DATA1
 其他: 保留

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

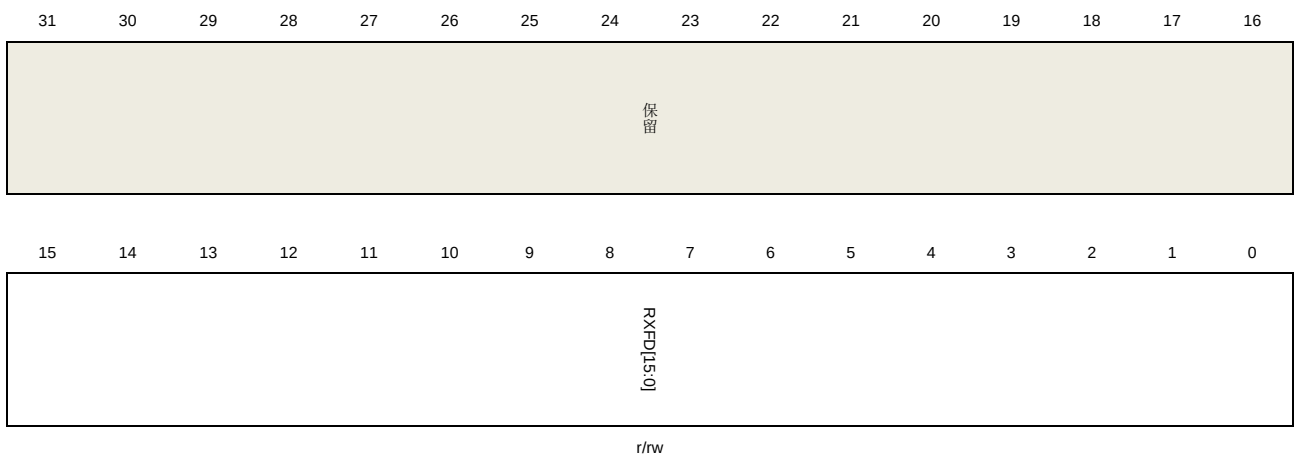
		00: DATA0
		10: DATA1
		其他: 保留
14:4	BCOUNT[10:0]	字节数 接收数据包的字节数
3:0	EPNUM[3:0]	端点号 当前接收包所属端点编号

全局接收 FIFO 长度寄存器（USBFS_GRFLEN）

地址偏移：0x0024

复位值：0x0000 0200

该寄存器只能按字（32 位）访问



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	RXFD[15:0]	Rx FIFO 深度 以32位字计数 $1 \leq RXFD \leq 1024$

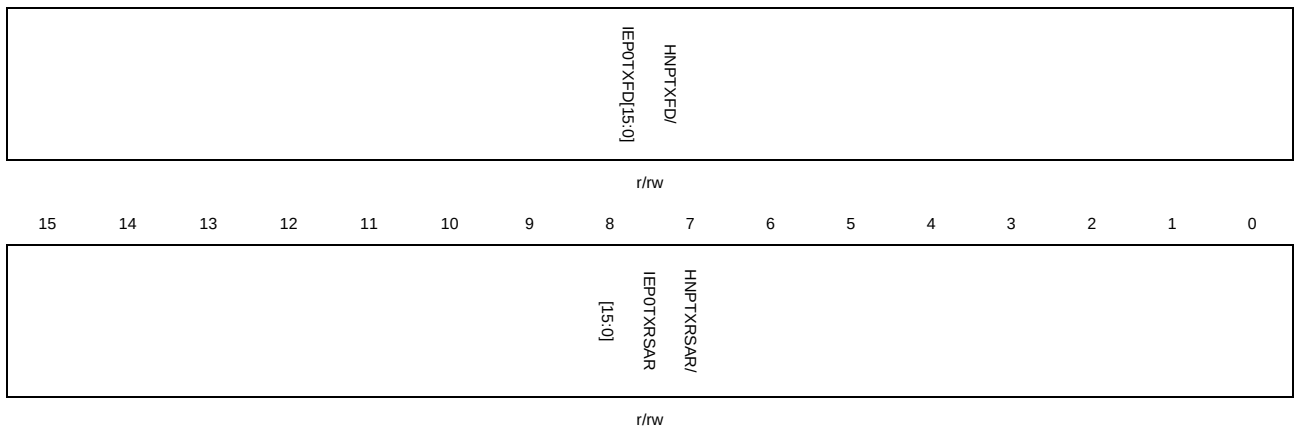
主机非周期性发送 FIFO 长度寄存器/设备 IN 端点 0 发送 FIFO 长度寄存器（USBFS_HNPTFLEN_DIEP0TFLEN）

地址偏移：0x0028

复位值：0x0200 0200

该寄存器只能按字（32 位）访问





主机模式下：

位/位域	名称	描述
31:16	HNPTXFD[15:0]	主机非周期性Tx FIFO深度 以32位字计数 $1 \leq \text{HNPTXFD} \leq 1024$
15:0	HNPTXRSAR[15:0]	主机非周期性Tx RAM起始地址 非周期性发送FIFO RAM的起始地址

设备模式下：

位/位域	名称	描述
31:16	IEP0TXFD[15:0]	输入端点0 Tx FIFO深度 以32位字计数 $16 \leq \text{IEP0TXFD} \leq 140$
15:0	IEP0TXRSAR[15:0]	输入端点0 TX RAM起始地址 端点0发送FIFO RAM的起始地址

主机非周期性发送 FIFO/队列状态寄存器（USBFS_HNPTFQSTAT）

地址偏移：0x002C

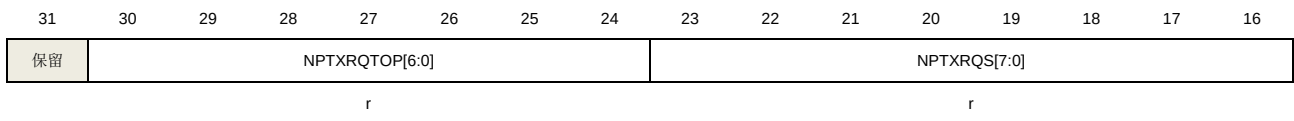
复位值：0x0008 0200

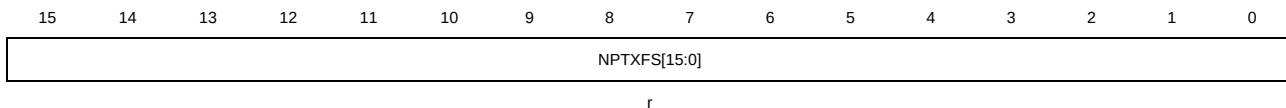
该寄存器反映了非周期性 Tx FIFO 和请求队列的当前状态。

请求队列包括在主机模式下的 IN、OUT 或其他请求条目。

注意：在设备模式下，该寄存器不可用。

该寄存器只能按字（32 位）访问





位/位域	名称	描述
31	保留	必须保持复位值。
30:24	NPTXRQTOP[6:0]	非周期性发送请求队列的顶部条目 在非周期性传输请求队列中的条目。 位30:27: 通道号 位26:25: – 00: IN/OUT 令牌 – 01: 0长度OUT包 – 11: 通道中止请求 位24: 结束标志位, 表明所选通道的最后一个条目
23:16	NPTXRQS[7:0]	非周期性发送请求队列空间 非周期性请求队列的剩余空间 0: 请求队列空 1: 1个条目 2: 2个条目 ... n: n个条目 (0≤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为空 1: 1个字 2: 2个字 ... n: n个字(0≤n≤NPTXFD) 其他: 保留

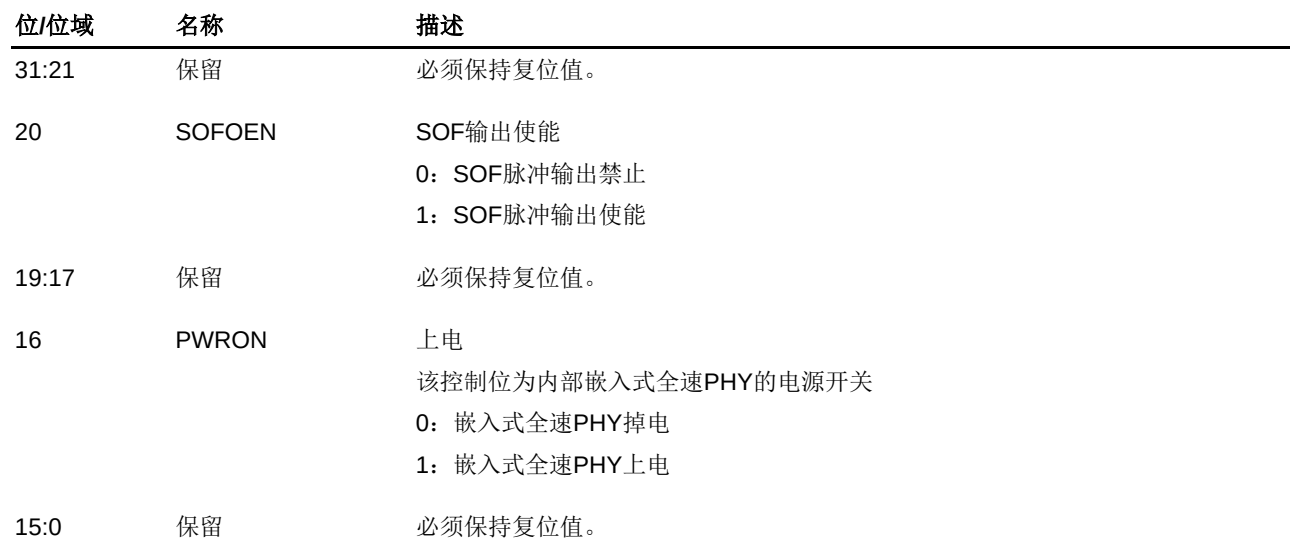
全局内核配置寄存器（USBFS_GCCFG）

地址偏移: 0x0038

复位值: 0x0000 0000

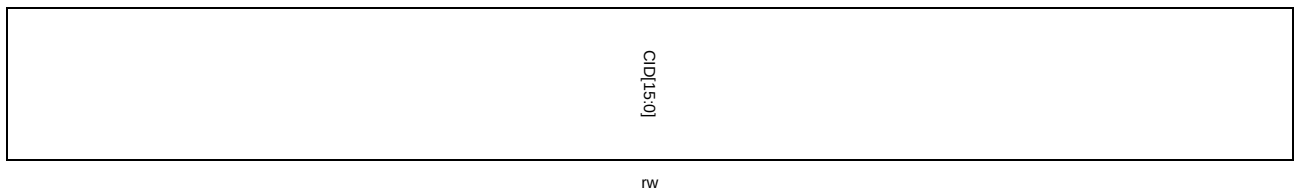
该寄存器只能按字（32 位）访问





该寄存器只能按字（32 位）访问





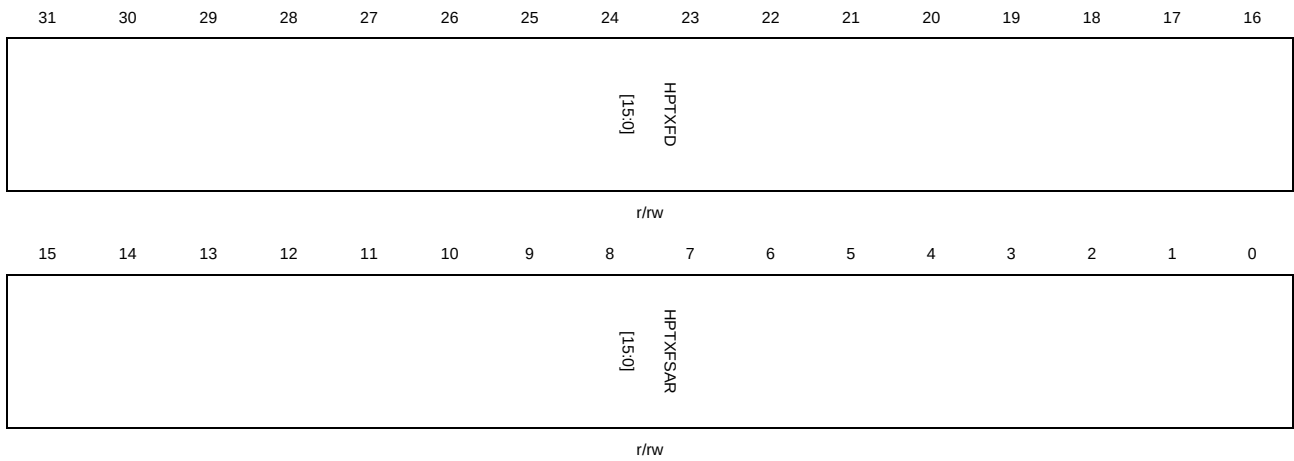
位/位域	名称	描述
31:0	CID[31:0]	内核ID 软件能够写入或读取该域值，并利用该域值为应用产生一个唯一ID。

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



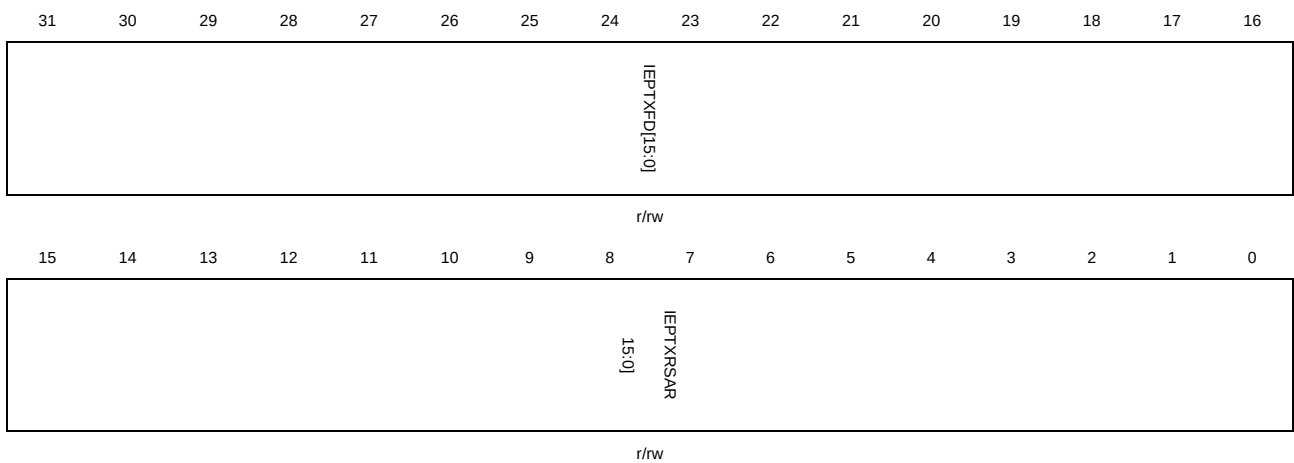
位/位域	名称	描述
31:16	HPTXFD[15:0]	主机周期性Tx FIFO深度 以32位字计数 $1 \leq \text{HPTXFD} \leq 1024$
15:0	HPTXFSAR[15:0]	主机周期性Tx RAM起始地址 主机周期性发送FIFO RAM起始地址

设备 IN 端点发送 FIFO 长度寄存器（USBFS_DIEPxTFLEN）（x = 1..3，其中 x 是 FIFO 编号）

地址偏移：0x0104 + （FIFO 编号 - 1） × 0x04

复位值：0x0200 0400 + （FIFO 编号 - 1） × 0x200

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 $1 \leq \text{HPTXFD} \leq 1024$
15:0	IEPTXRSAR[15:0]	IN 端点FIFOx Tx RAM起始地址 以32位字为单位的IN端点发送FIFOx起始地址

28.7.2. 主机控制和状态寄存器组

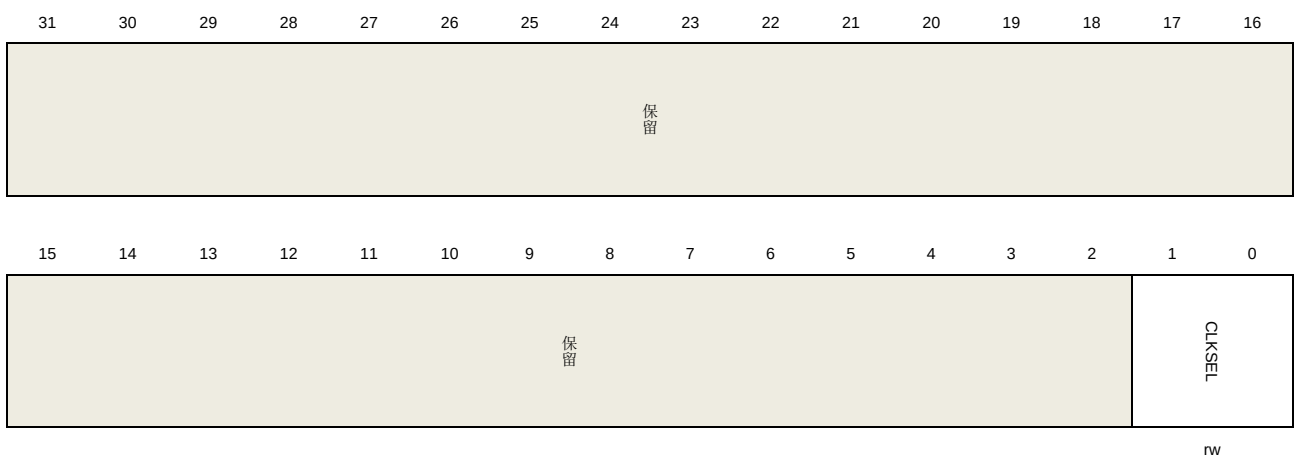
主机控制寄存器（USBFS_HCTL）

地址偏移：0x0400

复位值：0x0000 0000

在主机模式下，上电后，该寄存器有 USB 内核配置。主机初始化后，无需修改。

该寄存器只能按字（32 位）访问



位/位域	名称	描述
31:2	保留	必须保持复位值。
1:0	CLKSEL[1:0]	USB时钟选择 01: 48MHz时钟 其他: 保留

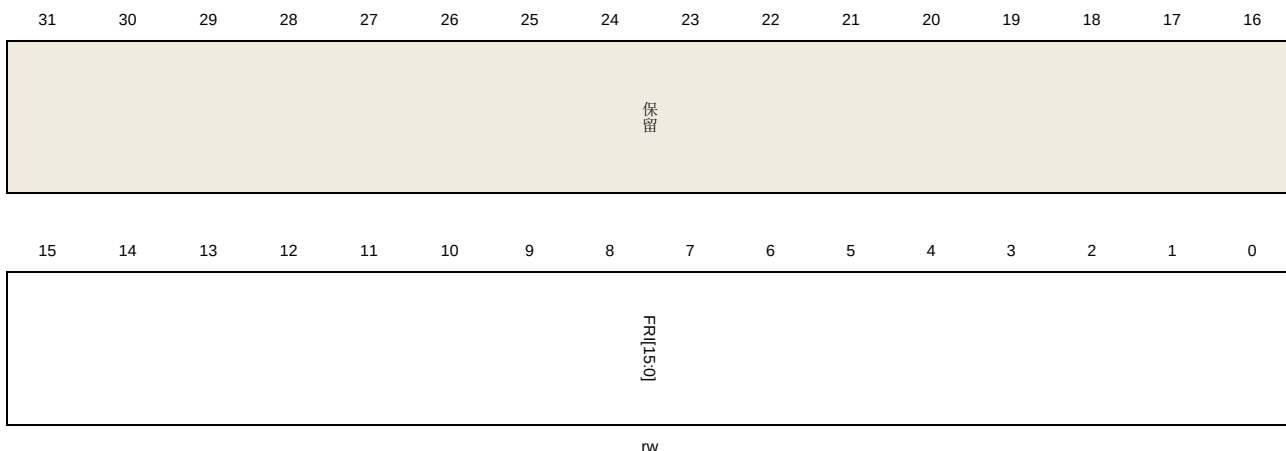
主机帧间隔寄存器（USBFS_HFT）

地址偏移: 0x0404

复位值: 0x0000 BB80

当 USBFS 控制器正在枚举中时，该寄存器为当前枚举速度设置帧间隔。帧间隔的修改将在下一帧生效。

该寄存器只能按字（32 位）访问



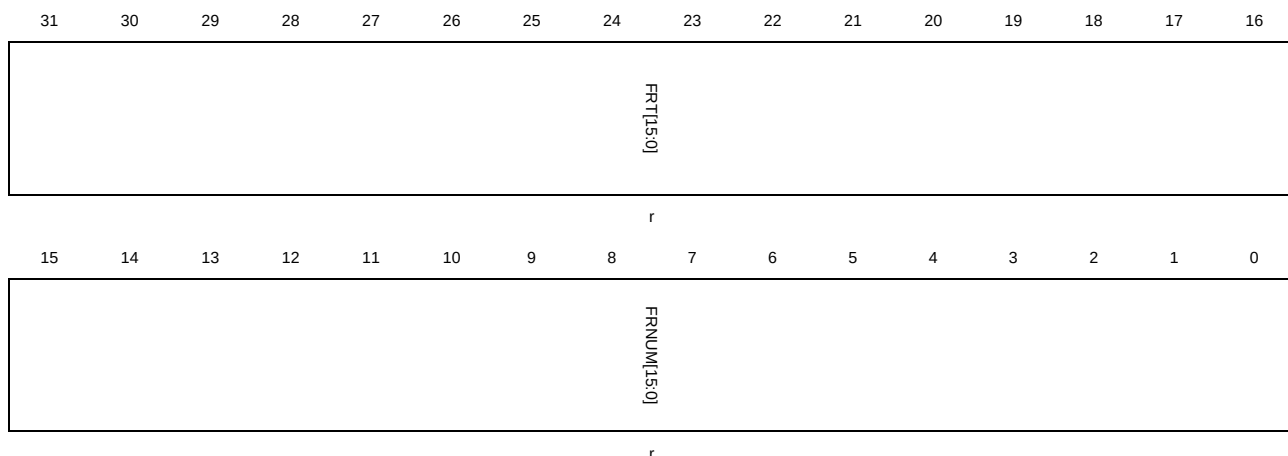
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	FRI[15:0]	帧间隔 该值描述了以PHY时钟为单位的帧周期。每次端口复位操作后，端口被使能，USBFS根据当前速度，采用一个固有值，并且软件可以向该位域写值以改变该固有值。该值需要采用以下描述的频率来进行计算： 全速: 48MHz 低速: 6MHz

主机帧信息保持寄存器（USBFS_HFINFR）

地址偏移: 0x0408

复位值: 0xBB80 0000

该寄存器只能按字（32 位）访问



位/位域	名称	描述
31:16	FRT[15:0]	帧剩余时间 该位域以PHY时钟为单位反映了当前帧剩余时间。
15:0	FRNUM[15:0]	帧号 该位域反映了当前帧的帧号，当其增加到0x3FFF后，其值变为0。

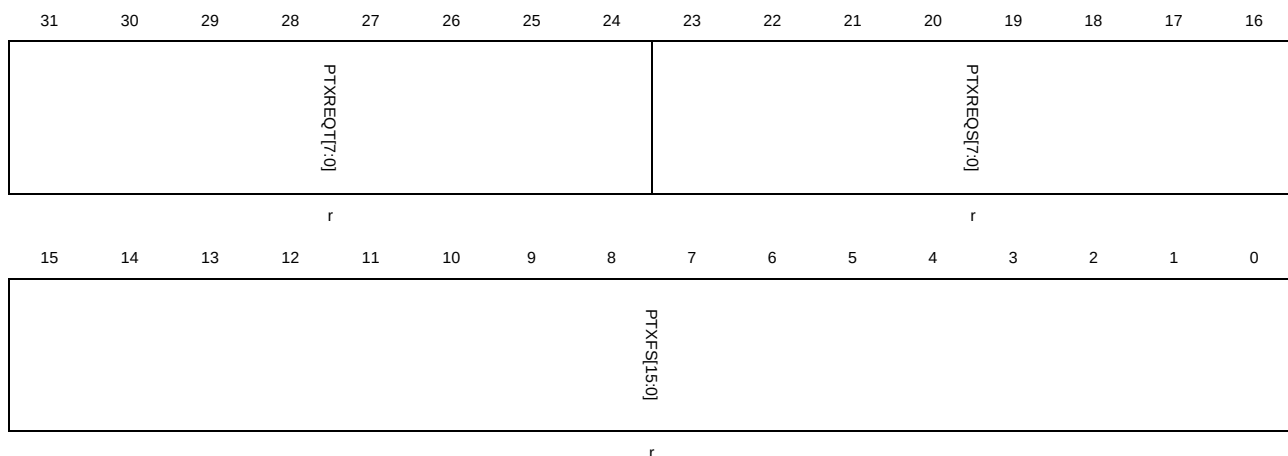
主机周期性发送 FIFO/队列状态寄存器（USBFS_HPTFQSTAT）

地址偏移：0x0410

复位值：0x0008 0200

该寄存器反映了主机周期性 Tx FIFO 和请求队列的当前状态。请求队列包括在主机模式下的 IN、OUT 或其他请求条目。

该寄存器只能按字（32 位）访问



位/位域	名称	描述

31:24	PTXREQT[7:0]	周期性Tx 请求队列的顶部条目 在周期性发送请求队列中的条目 位30:27: 通道号 位26:25: – 00: IN/OUT 令牌 – 01: 0长度OUT包 – 11: 通道中止请求 位24: 中止标志, 指示所选通道的最后一个条目
23:16	PTXREQS[7:0]	周期性发送请求队列空间 周期性发送请求队列剩余空间 0: 请求队列为空 1: 1个条目 2: 2个条目 ... n: n个条目 ($0 \leq n \leq 8$) 其他: 保留
15:0	PTXFS[15:0]	周期性发送FIFO空间 周期性发送FIFO剩余空间 以32位字计数 0: 周期性发送FIFO为空 1: 1个字 2: 2个字 ... n: n个字 ($0 \leq n \leq PTXFD$) 其他: 保留

主机所有通道中断寄存器（USBFS_HACHINT）

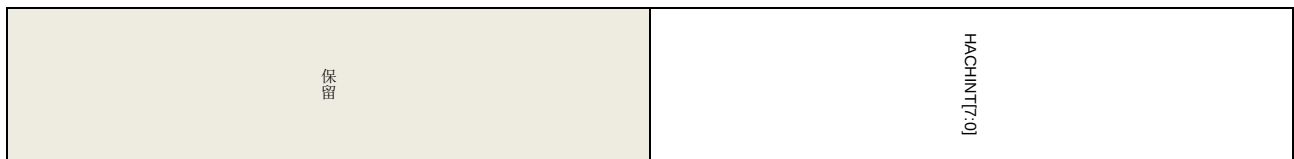
地址偏移: 0x0414

复位值: 0x0000 0000

当触发一个通道中断时，USBFS 在该寄存器中置位相应的位，并且软件可以读取该寄存器以获取产生中断的通道。

该寄存器只能按字（32 位）访问





位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	HACHINT[7:0]	主机所有通道中断 每一位表示一个通道：位0代表通道0，位7表示通道7

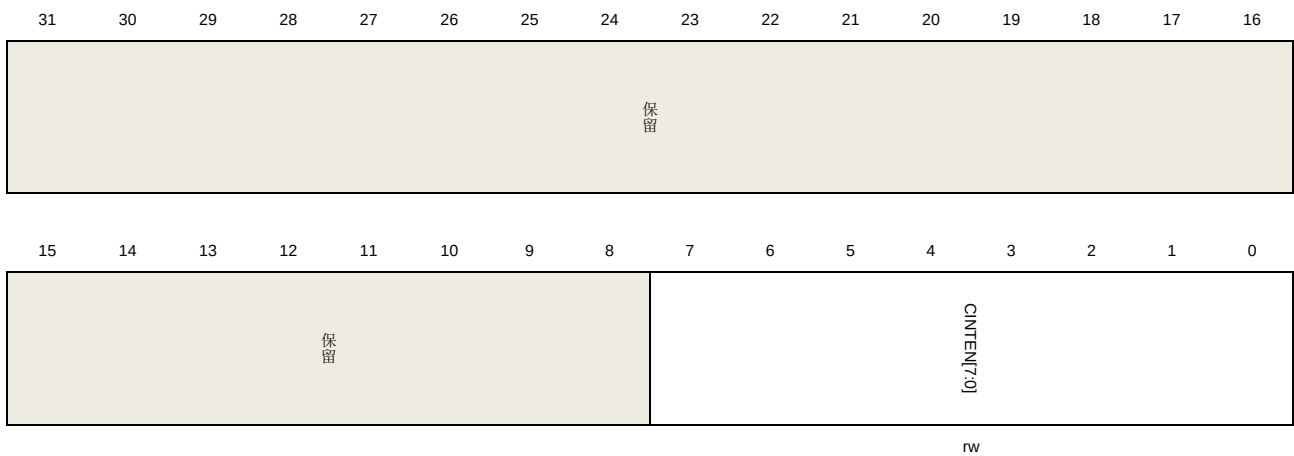
主机所有通道中断使能寄存器（USBFS_HACHINTEN）

地址偏移：0x0418

复位值：0x0000 0000

软件可以使用该寄存器使能或禁用一个通道的中断。只有该寄存器中相应通道的中断使能控制位被置位，USBFS_GINTF 寄存器中的通道中断标志位 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	通道中断使能 0：禁用通道n中断 1：使能通道n中断 每一位表示一个通道：位0代表通道0，位7代表通道7

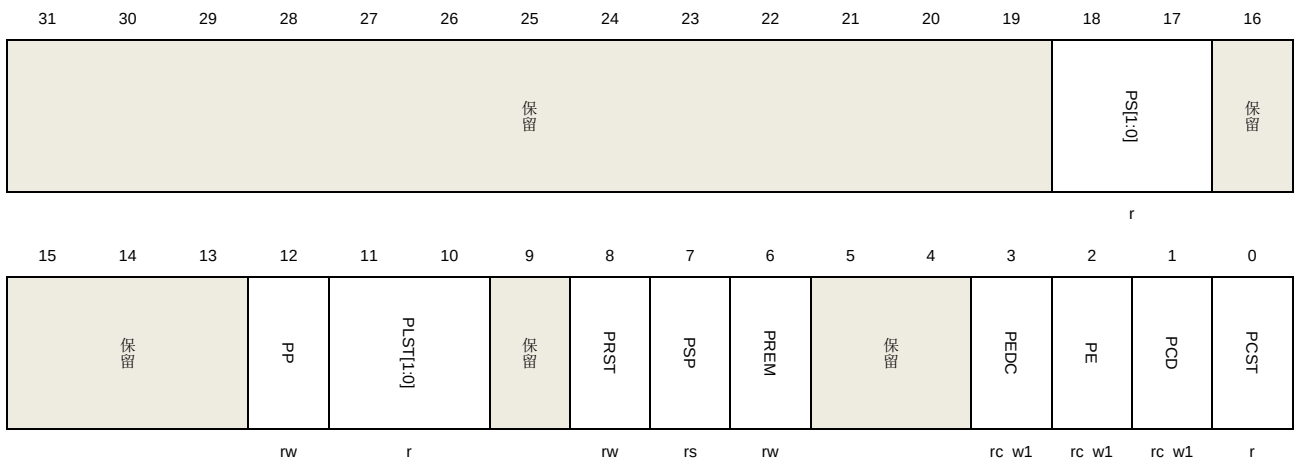
主机端口控制和状态寄存器（USBFS_HPCS）

地址偏移：0x0440

复位值：0x0000 0000

该寄存器控制端口行为，并且也包含一些反映端口状态的标志位。如果本寄存器中的 PRST、PEDC 和 PCD 标志位被 USBFS 置位的话，USBFS_GINTF 寄存器中的 HPIF 标志位会被置位。

该寄存器只能按字（32 位）访问



位/位域	名称	描述
31:19	保留	必须保持复位值。
18:17	PS[1:0]	端口速度 反映连接到该端口的设备的枚举速度。 01: 全速 10: 低速 其他: 保留
16:13	保留	必须保持复位值。
12	PP	端口供电 在端口被使用后，该控制位应该被置位。由于USBFS不具有电源供应能力，它只能使用该控制位以获取该端口是否在供电状态。软件应该在设置该控制位之前，保证在VBUS引脚上具有电源供应。 0: 端口掉电 1: 端口供电
11:10	PLST[1:0]	端口线状态 反映USB数据线当前状态 位10: DP线状态 位11: DM线状态

9	保留	必须保持复位值。
8	PRST	端口复位 应用通过设置该控制位以在USB端口上启动一个复位信号。当应用希望停止复位信号时，应用应该清除该控制位。 0：端口不在复位状态 1：端口处于复位状态
7	PSP	端口挂起 应用设置该控制位来将端口进入挂起状态。当该控制位被置位后，端口停止发送SOF令牌包。该控制位只能够通过以下操作清除。 - 应用置位该寄存器中的PRST控制位 - 置位该寄存器中的PREM控制位 - 检测到一个远程唤醒信号 - 检测到一个设备断开 0：端口不在挂起状态 1：端口处于挂起状态
6	PREM	端口恢复 应用通过置位该控制位以在USB端口上启动一个恢复信号。当应用希望停止恢复信号时，应用可以清除该控制位。 0：无恢复驱动 1：恢复驱动
5:4	保留	必须保持复位值。
3	PEDC	端口使能/禁止更改 当该寄存器中的位2端口使能控制位更改时，USB内核置位该标志位。
2	PE	端口使能 当USB复位信号完成后，USBFS自动置位该位，并且该位不可由软件置位。 该位可通过以下事件清除： - 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 是通道编号）

地址偏移：0x0500 +（通道编号 × 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	CDIS	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]					MPU[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），该位控制将要处理的通道事务为奇数帧还是偶数帧。
28:22	DAR[1:0]	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	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

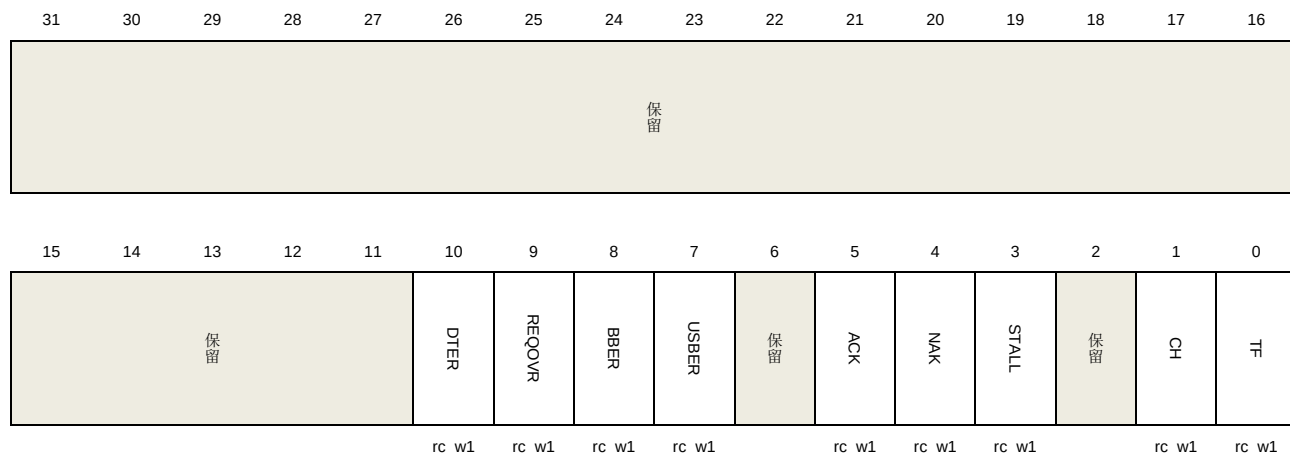
主机通道 x 中断标志寄存器（USBFS_HCHxINTF）（x = 0...7，其中 x 是通道编号）

地址偏移：0x0508 + (通道编号 × 0x20)

复位值：0x0000 0000

该寄存器包含一个通道的状态和事件，当软件获取一个通道中断时，软件需要为相应通道读取该寄存器以获取产生中断的中断源。该寄存器中的标志位均由硬件置位，并且写 1 清除。

该寄存器只能按字（32 位）访问



位/位域	名称	描述
31:11	保留	必须保持复位值。
10	DTER	数据切换错误 IN事务获取一个数据包，但是该包的PID和USBFS_HCHxLEN寄存器中的DPID[1:0]控制位不匹配。
9	REQOVR	请求队列上溢

		当软件启动新的传输时，请求队列上溢。
8	BBER	串扰错误 USB总线上发生一个串扰事件。产生串扰事件的典型原因是端点发送了一个数据包，但是数据包长度超过了端点的最大包长。
7	USBER	USB总线错误 当在接收一个数据包的过程中，发生以下事件时，将置位USB总线错误标志位： 接收包有一个错误的CRC域 在USB总线上检测到填充错误 当等待一个响应包时，超时
6	保留	必须保持复位值。
5	ACK	ACK 接收或者发送一个ACK响应包
4	NAK	NAK 接收到一个NAK响应包
3	STALL	STALL 接收到一个STALL响应包
2	保留	必须保持复位值。
1	CH	通道中止 通道被当前请求所禁用，在当前请求处理的过程中，并不响应其他请求处理。
0	TF	发送完成 该通道所有的事务成功完成并且无错误发生。 对于IN通道，在USBFS_HCHxLEN寄存器的PCNT位减到0后，该标志位被置位。 对于OUT通道，当软件从RxFIFO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 是通道编号）

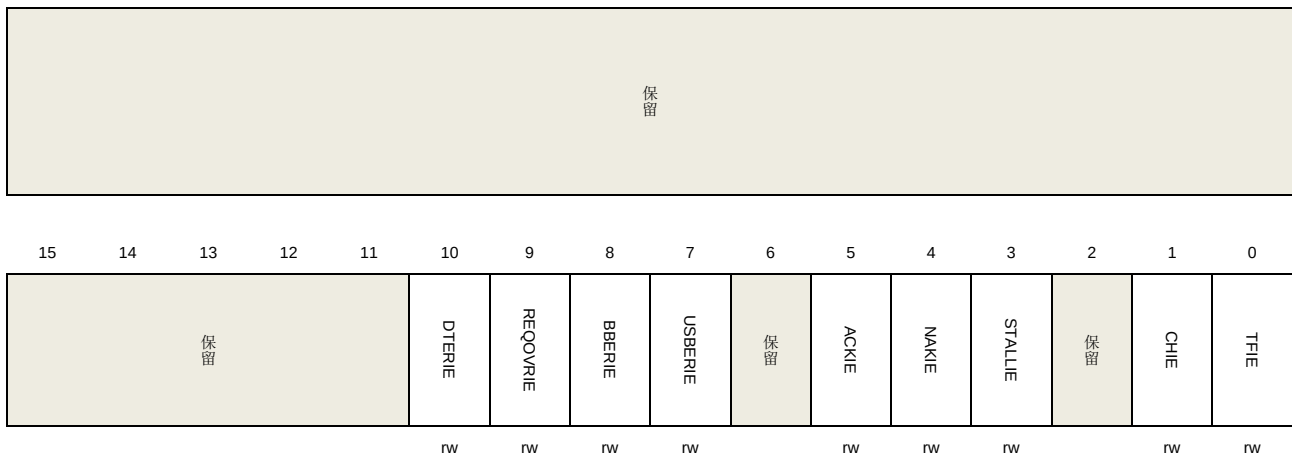
地址偏移：0x050C +（通道编号 × 0x20）

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

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



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

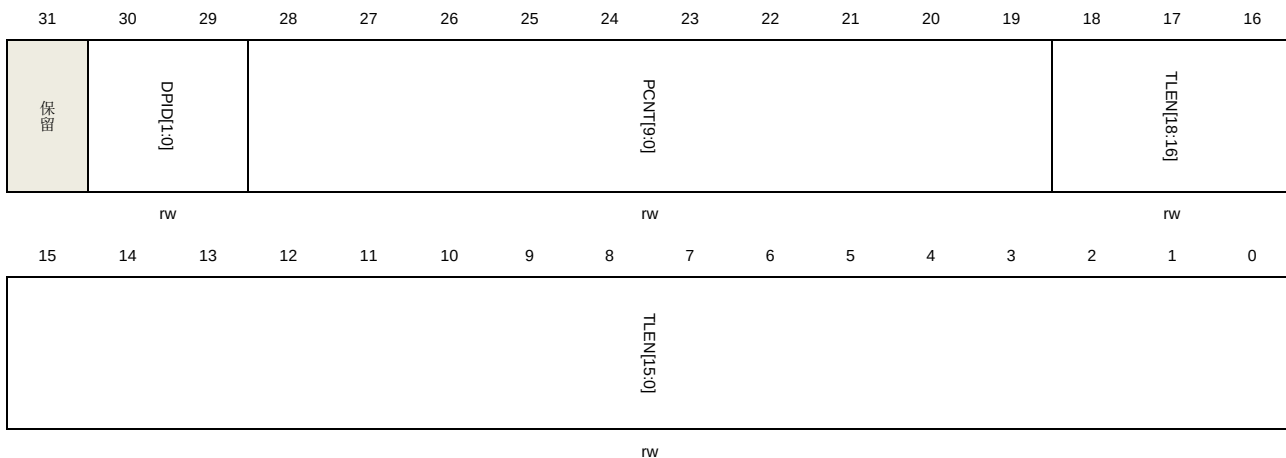
2	保留	必须保持复位值。
1	CHIE	通道中止中断使能 0: 禁用通道中止中断 1: 使能通道中止中断
0	TFIE	传输完成中断使能 0: 禁用传输完成中断 1: 使能传输完成中断

主机通道 x 长度寄存器（USBFS_HCHxLEN）（x = 0...7，其中 x 是通道编号）

地址偏移：0x0510 +（通道编号 × 0x20）

复位值：0x0000 0000

该寄存器只能按字（32 位）访问



位/位域	名称	描述
31	保留	必须保持复位值。
30:29	DPID[1:0]	数据PID 软件应该在传输起始之前写该段位域。对于OUT传输，该位域包含第一个传输包的数据PID。对于IN传输，该位域包含第一个接收包的数据PID，并且如果数据PID不匹配的话，将会触发DTER标志位。在传输开始之后，USBFS遵循USB协议自动改变和切换该位域。 00: DATA0 10: DATA1 11: SETUP（仅对于控制传输） 01: 保留
28:19	PCNT[9:0]	包计数 在一个传输中希望发送（OUT）或接收（IN）的数据包个数。

软件应该在通道使能之前写该位域。在传输启动之后，该位域在USBFS正确传输每个数据包后，自动减少。

18:0	TLEN[18:0]	传输长度 一次传输的总数据字节数。 对于OUT传输，该位域为OUT传输中期望发送的所有数据包总数据字节数。软件应该在通道使能之前写该位域。当软件或DMA正确向通道的数据FIFO中写入一个包时，该位域以包中字节大小进行减少。 对于IN传输，每次软件或DMA从RxFIFO中读取一个包后，该位域也以包中字节大小进行减少。

28.7.3. 设备控制和状态寄存器组

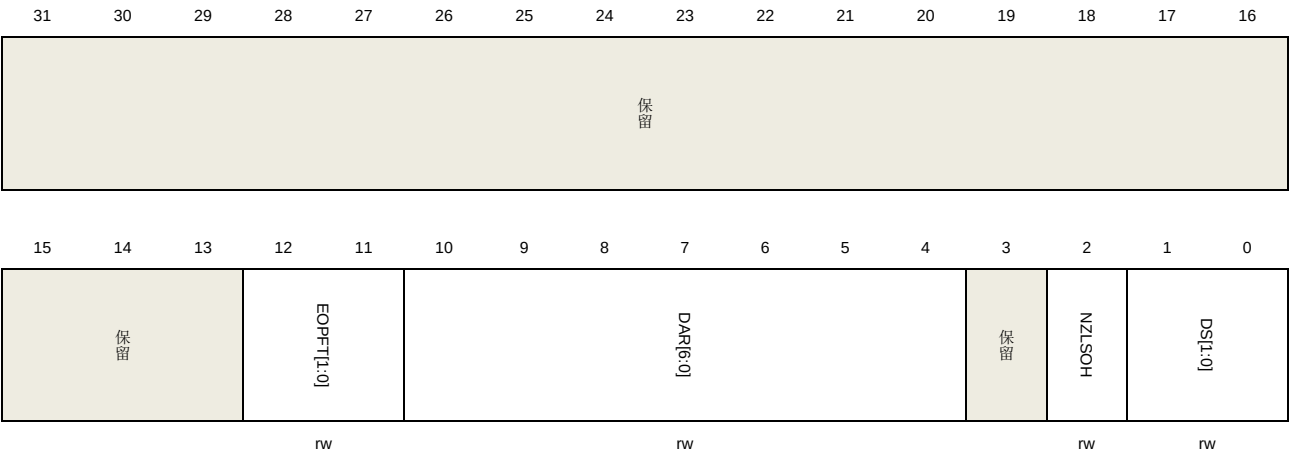
设备配置寄存器（USBFS_DCFG）

地址偏移：0x0800

复位值：0x0000 0003

在上电、枚举或执行某些控制命令后，该寄存器配置内核为设备模式。在设备初始化后，不可以改变该寄存器值。

该寄存器采用字（32 位）访问



位/位域	名称	描述
31:13	保留	必须保留为复位值。
12:11	EOPFT[1:0]	周期性帧尾时间 该域定义周期性帧时间的帧尾标志触发的时间点 00: 80%的帧时间 01: 85%的帧时间 10: 90%的帧时间

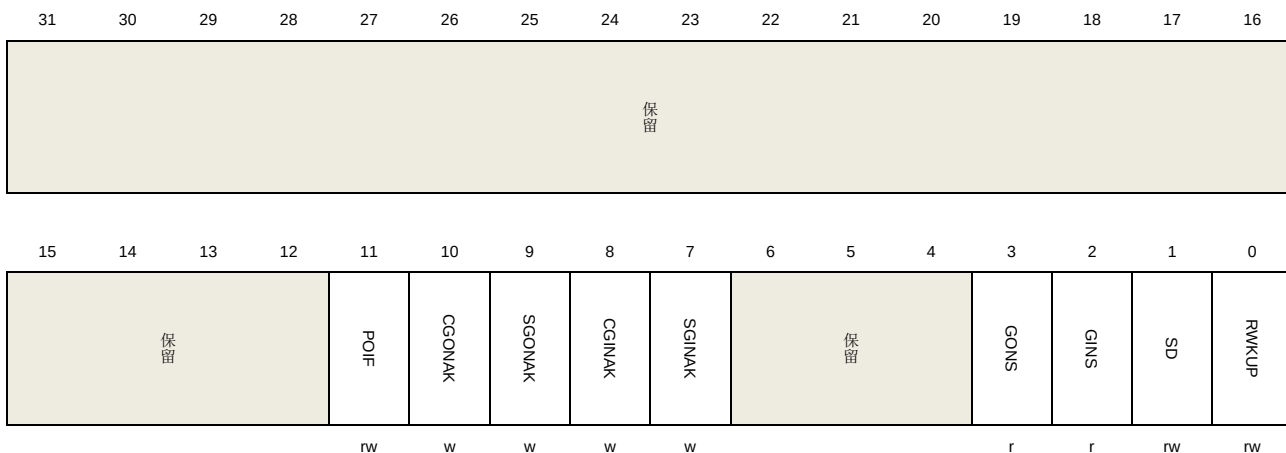
		11: 95%的帧时间
10:4	DAR[6:0]	设备地址 该位定义USB设备地址，USBFS采用该位匹配接收的设备令牌地址域，在接收到来自主机的设置地址的命令后，软件设置该域
3	保留	必须保留为复位值。
2	NZLSOH	非零长度OUT状态阶段握手 在控制传输的OUT状态阶段，当USB设备接收到一个非零长度数据包时，该域控制控制USBFS是接收该包，还是用STALL握手信号拒绝该包。 0: 把该包视为正常包，根据设备OUT端点控制寄存器的NAKS和STALL位，回复握手相应握手包 1: 发送STALL握手，不保存接收到的OUT数据包
1:0	DS[1:0]	设备速度 该域控制设备连入主机后的设备速度 11: 全速 其他: 保留

设备控制寄存器 (USBFS_DCTL)

地址偏移: 0x0804

复位值: 0x0000 0000

该寄存器采用字（32 位）访问



位/位域	名称	描述
31:12	保留	必须保留为复位值。
11	POIF	上电初始化完成 软件通过设置该位，通知USBFS寄存器在从掉电模式下唤醒，然后完成初始化。

10	CGONAK	清零全局OUT NAK 软件设置该位从而清零该寄存器的GONS位
9	SGONAK	设置全局OUT NAK 软件设置该位从而实现该寄存器的位GONS置位。 当GONS位为零，设置该位会引起USBFS_GINTF寄存器的GONAK标志触发，软件应该在再写该位前清除GONAK标志。
8	CGINAK	清零全局IN NAK 软件设置该位从而清零该寄存器的GINS位
7	SGINAK	设置全局IN NAK 软件设置该位从而实现该寄存器的位GINS置位 当GINS位为零，设置该位会引起USBFS_GINTF寄存器的GINAK标志触发，软件应该在再写该位前清除GINAK标志。
6:4	保留	必须保留为复位值。
3	GONS	全局OUT NAK状态 0: USBFS回复OUT事务的握手信号以及是否保存OUT数据包由Rx FIFO状态、端点的NAKS、STALL位确定。 1: USBFS回复OUT事务NAK握手信号，不保存接收的OUT数据包。
2	GINS	全局IN NAK状态 0: USBFS回复IN事务的握手信号由Tx FIFO状态、端点的NAKS、STALL位确定。 1: USBFS通常回复IN事务NAK握手信号
1	SD	软断开 软件可实现USB总线上的软断开，在置1该位后，关掉DP线上的上拉电阻，从而引起主机检测设备的断开。 0: 没有软断开生成 1: 生成软断开
0	RWKUP	远程唤醒 在挂起状态，软件可通过该位来生成一个远程唤醒信号来通知主机恢复USB总线 0: 没有远程唤醒信号生成 1: 生成远程唤醒信号

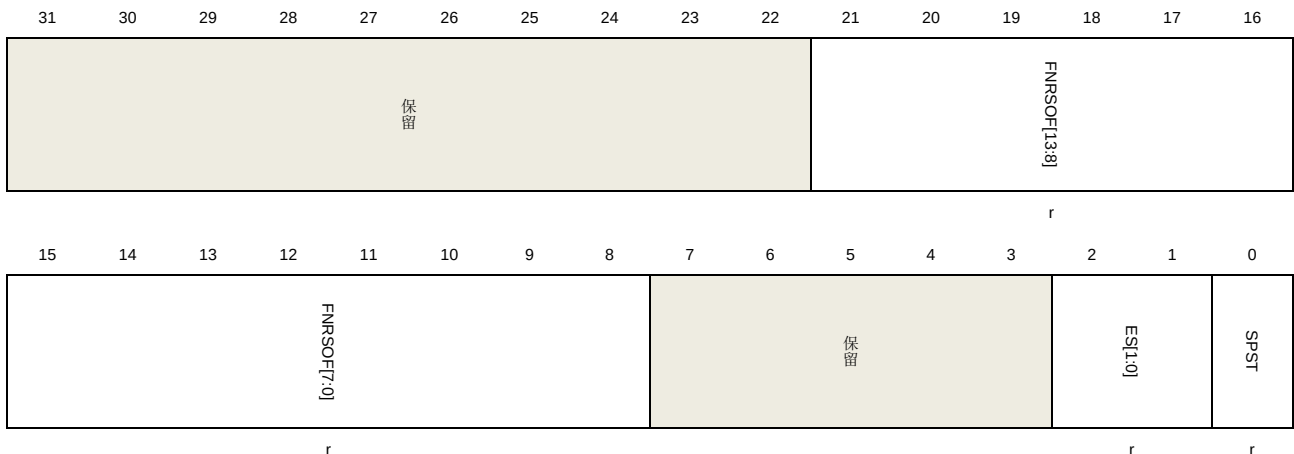
设备状态寄存器（USBFS_DSTAT）

地址偏移：0x0808

复位值：0x0000 0000

该寄存器包含设备模式下的 USBFS 的状态和信息。

该寄存器采用字（32 位）访问



位/位域	名称	描述
31:22	保留	必须保留为复位值。
21:8	FNRSOFF[13:0]	所接收的SOF帧编号 USBFS会在接收到一个SOF令牌后更新该域。
7:3	保留	必须保留为复位值。
2:1	ES[1:0]	枚举速度 该域指示所枚举的设备速度，在寄存器USBFS_GINTF的ENUMF标志触发后，软件可以读取该域。 01 : 全速 其他: 保留
0	SPST	挂起状态 该位指示设备是否处于挂起状态。 0 : 设备不在挂起状态 1 : 设备在挂起状态

设备 IN 端点通用中断使能寄存器（USBFS_DIEPINTEN）

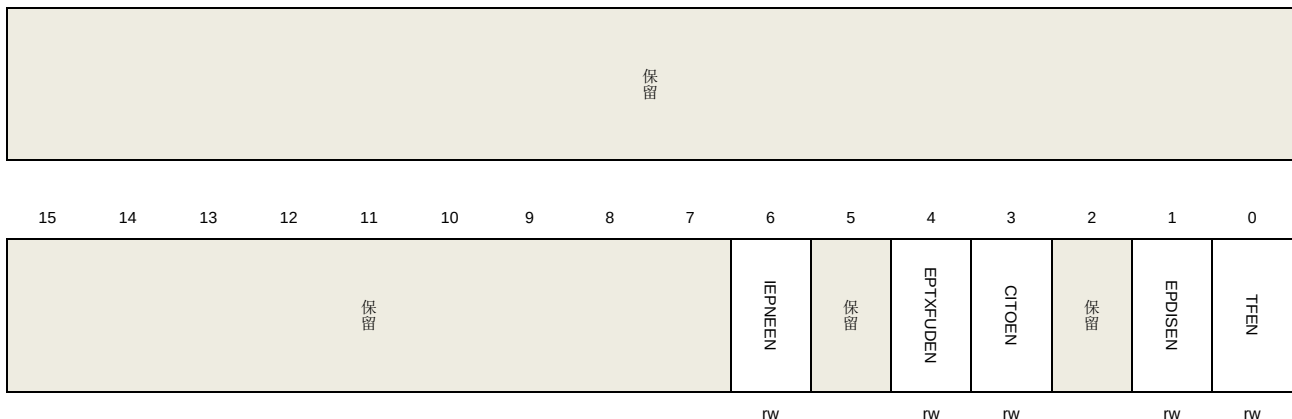
地址偏移: 0x810

复位值: 0x0000 0000

该寄存器包含寄存器 USBFS_DIEPxINTF 中的标志的中断使能位，如果软件置 1 某位，其在寄存器 USBFS_DIEPxINTF 中对应的位可以触发一个寄存器 USBFS_DAEPINT 端点中断。该位可以通过软件置位和清零。

该寄存器采用字（32 位）访问





位/位域	名称	描述
31:7	保留	必须保留为复位值。
6	IEPNEEN	IN端点NAK有效中断使能位 0: 除能中断 1: 使能中断
5	保留	必须保留为复位值。
4	EPTXFUDEN	端点Tx FIFO下溢中断使能位 0: 除能中断 1: 使能中断
3	CITOEN	控制IN事务超时中断使能位 0: 除能中断 1: 使能中断
2	保留	必须保留为复位值。
1	EPDISEN	端点除能中断使能位 0: 除能中断 1: 使能中断
0	TFEN	传输完成中断使能位 0: 除能中断 1: 使能中断

设备 OUT 端点通用中断使能寄存器（USBFS_DOEPINTEN）

地址偏移：0x0814

复位值：0x0000 0000

该寄存器包含寄存器 USBFS_DOEPxINTF 中的标志的中断使能位，如果软件置 1 某位，其在寄存器 USBFS_DOEPxINTF 中对应的位可以触发一个寄存器 USBFS_DAEPINT 端点中断。该位可

以通过软件置位和清零。

该寄存器采用字（32 位）访问



位/位域	名称	描述
31:7	保留	必须保留为复位值。
6	BTBSTPEN	连续SETUP包中断使能位（仅适用于控制OUT端点） 0：除能中断 1：使能中断
5	保留	必须保留为复位值。
4	EPRXFOVREN	端点Rx FIFO上溢中断使能位 0：除能中断 1：使能中断
3	STPFEN	SETUP阶段完成中断使能位（仅适用于控制OUT端点） 0：除能中断 1：使能中断
2	保留	必须保留为复位值。
1	EPDISEN	端点除能中断使能位 0：除能中断 1：使能中断
0	TFEN	传输完成中断使能位 0：除能中断 1：使能中断

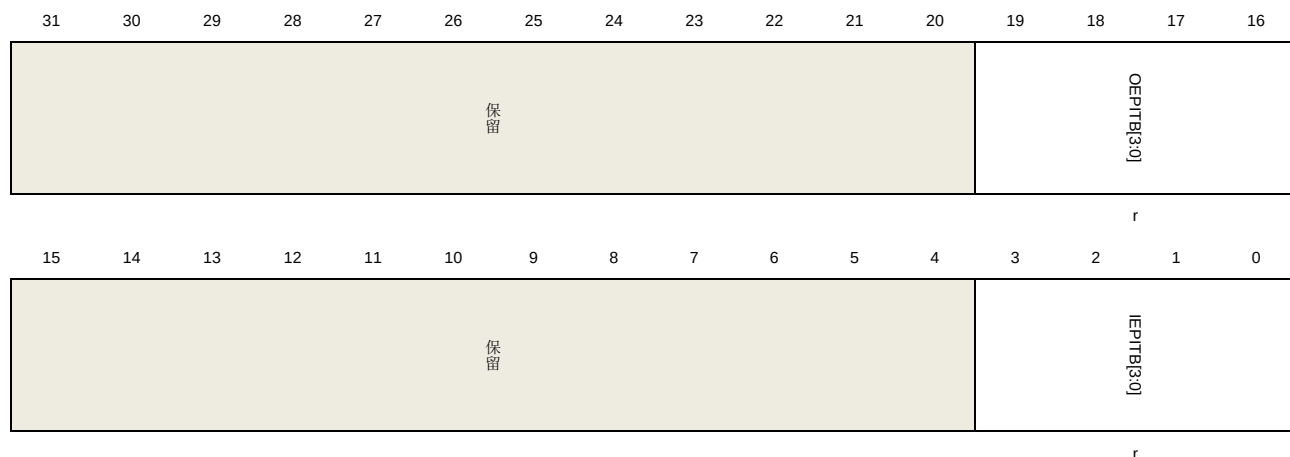
设备端点中断寄存器（USBFS_DAEPINT）

地址偏移：0x0818

复位值：0x0000 0000

当一个端点的中断被触发，USBFS 置 1 该寄存器的相应位，软件可通过该寄存器知道在本次中断中的端点号。

该寄存器采用字（32 位）访问



位/位域	名称	描述
31:20	保留	必须保留为复位值。
19:16	OEPITB[3:0]	设备OUT端点中断位 每个位代表一个OUT端点：Bit16代表OUT端点0，Bit19代表OUT端点3
15:4	保留	必须保留为复位值。
3:0	IEPITB[3:0]	设备IN端点中断位 每个位代表一个IN端点：Bit0代表IN端点0，Bit3代表IN端点3

设备端点中断使能寄存器（USBFS_DAEPINTEN）

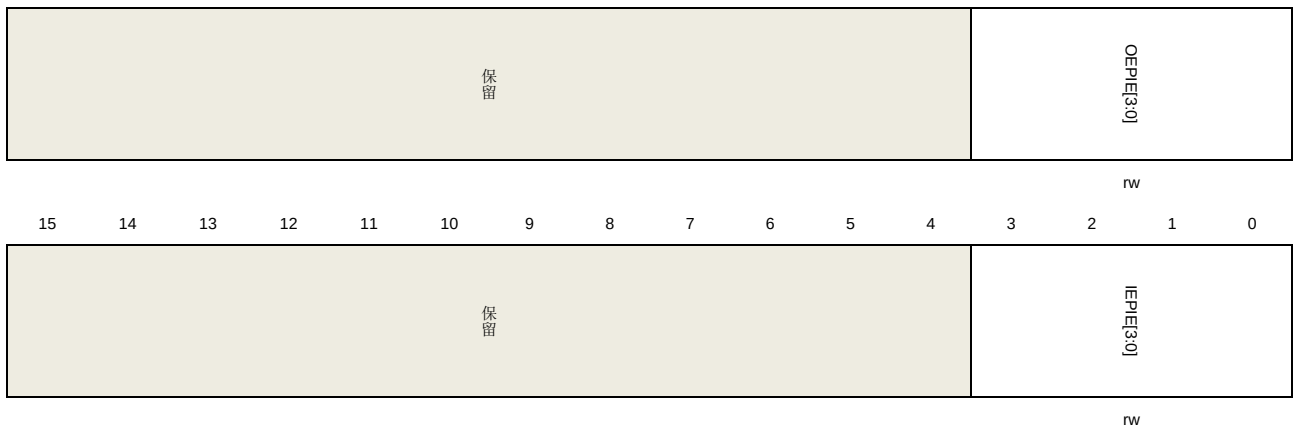
地址偏移：0x081C

复位值：0x0000 0000

该寄存器可通过软件使能或除能端点的中断，只有当端点在该寄存器中相应位被置 1 才能触发寄存器 USBFS_GINTF 的端点中断标志 OEPIF 或 IEPF。

该寄存器采用字（32 位）访问





位/位域	名称	描述
31:20	保留	必须保留为复位值。
19:16	OEPIE[3:0]	OUT 端点中断使能位 0: 除能OUT端点n中断 1: 使能OUT端点n中断 每个位代表一个OUT端点: Bit16对应OUT端点0, Bit19对应OUT端点3
15:4	保留	必须保留为复位值。
3:0	IEPIE[3:0]	IN 端点中断使能位 0: 除能IN端点n中断 1: 使能IN端点n中断 每个位代表一个IN端点: Bit0对应IN端点0, Bit3对应IN端点3

设备 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



保留		IEPTXFEIE[3:0]
		rw

位/位域	名称	描述
31:4	保留	必须保留为复位值。
3:0	IEPTXFEIE[3:0]	IN端点Tx FIFO空中断的使能位 该域控制着USBFS_DIEPxINTF寄存器的TXFE位能否生成一个寄存器 USBFS_DAEPIINT的端点中断位 Bit0对应IN端点0，Bit5对应IN端点5 0：除能FIFO空中断 1：使能FIFO空中断

设备 IN 端点 0 控制寄存器（USBFS_DIEP0CTL）

地址偏移：0x0900

复位值：0x0000 8000

该寄存器采用字（32 位）访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPEN	EPD	保留	保留	SNAK	CNAK	TXFNUM[3:0]				STALL	保留	EPTYPE[1:0]		NAKS	保留
rs	rs			w	w	rw				rs		r		r	
												r			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPACT	保留														MP_L[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	必须保留为复位值。
27	SNAK	置位NAK 软件置位该位来设置该寄存器的NAKS位
26	CNAK	清零NAK 软件置位该位来清零该寄存器的NAKS位
25:22	TXFNUM[3:0]	Tx FIFO编号 定义IN端点0的Tx FIFO编号
21	STALL	STALL握手 当接收IN令牌时，软件可以通过置1该位发送STALL握手包，对于相应的OUT端点0，在接收SETUP令牌后，USBFS清除此位。该位比该寄存器的NAKS位和寄存器USBFS_DCTL的GINS位优先级要高，如果STALL和NAKS位都被置位，STALL位生效。
20	保留	必须保留为复位值。
19:18	EPTYPE[1:0]	端点类型 该域固定为'00',控制端点。
17	NAKS	NAK状态 当该寄存器的STALL位和寄存器USBFS_DCTL的位GINS被清零，该位控制USBFS的NAK状态。 0: 根据端点Tx FIFO的状态，USBFS发送数据或握手包 1: USBFS总为IN令牌发送NAK握手包 该位是只读位，可以通过该寄存器的位CNAK和位SNAK控制该位
16	保留	必须保留为复位值。
15	EPACT	端点激活 对于端点0来说，该域固定为'1'
14:2	保留	必须保留为复位值。
1:0	MPL[1:0]	最大包长 域定义了控制数据包的最大包长，如USB 2.0协议所描述，对控制传输而言，有四种包长度： 00: 64字节 01: 32字节 10: 16字节 11: 8字节

设备 IN 端点 x 控制寄存器 (USBFS_DIEPxCTL) (x = 1..3, x 是端点编号)

地址偏移: 0x0900 + (端点编号 × 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	SD0PID/ SEVENFRM	SNAK	CNAK	TXFNUM[3:0]				STALL	保留	EP[TYPE[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				MP[10:0]									
rw						rw									

位/位域	名称	描述
31	EPEN	端点使能 软件置位, USBFS清零 0: 端点除能 1: 端点使能 软件应该按照操作指南使能或除能端点
30	EPD	端点除能 软件可通过置位该位从而除能端点, 软件应该按照操作指南使能或除能端点。
29	SODDFRM	设置奇数帧 (适用于同步IN端点) 软件通过置1该位置1该寄存器的EOFRM位
	SD1PID	设置DATA1 PID(适用于中断和大容量IN端点) 软件可通过置1该位置1该寄存器的DPID位
28	SEVENFRM	设置偶数帧(适用于同步IN端点) 软件通过置1该位清零该寄存器的EOFRM位
	SD0PID	设置DATA1(适用于中断和大容量IN端点) 软件可通过置1该位清零该寄存器的DPID位
27	SNAK	设置NAK 软件置1该位置1该寄存器的NAKS位
26	CNAK	清零NAK

		软件置1该位清零该寄存器的NAKS位
25:22	TXFNUM[3:0]	Tx FIFO编号 该位定义了IN端点的Tx FIFO编号
21	STALL	STALL握手 当接收IN令牌时，软件可以通过置1该位发送STALL握手包。该位比该寄存器的NAKS位和寄存器USBFS_DCTL的GINS位优先级要高，如果STALL和NAKS位都被置位，STALL位生效。 对于控制IN端点： 当对应的OUT端点接收到SETUP令牌时，只有USBFS可以清零此位，软件不可清除此位。 对于中断或大容量IN端点： 只有软件可以清零此位。
20	保留	必须保留为复位值。
19:18	EPTYPE[1:0]	端点类型 该域定义端点的传输类型： 00：控制 01：同步 10：大容量 11：中断
17	NAKS	NAK状态 当该寄存器的STALL位和寄存器USBFS_DCTL的位GINS被清零，该位控制USBFS的NAK状态： 0：根据端点Tx FIFO的状态，USBFS发送数据或握手包 1：USBFS总为IN令牌发送NAK握手包 该位是只读位，可以通过该寄存器的位CNAK和位SNAK控制该位
16	EOFRM	奇偶帧（适用于同步IN端点） 对于同步传输，软件通过使用该位控制USBFS只在奇数帧或偶数帧为IN事务发送数据包，如果当前帧号的奇偶性不匹配该位，USBFS回复一个零长度的包： 0：只在偶数帧发送数据 1：只在奇数帧发送数据
	DPID	端点数据PID（适用于中断或大容量IN端点） 在端点或大容量传输中，有数据PID翻转机制，在传输开始之前，软件通过设定SD0PID来设置此位，按照USB协议中描述的数据PID翻转机制，USBFS在传输过程中保持该位。 0：数据包的PID是DATA0 1：数据包的PID是DATA1
15	EPACT	端点激活

该位控制端点是否激活，当端点没有激活，忽略任何令牌，不做任何回复。

14:11	保留	必须保留为复位值。
10:0	MPL[10:0]	该域定义最大包长

设备 OUT 端点 0 控制寄存器（USBFS_DOEP0CTL）

地址偏移: 0x0B00

复位值: 0x0000 8000

该寄存器采用字（32 位）访问

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

位/位域	名称	描述
31	EPEN	端点使能 软件置位，USBFS清零 0：端点除能 1：端点使能 软件应该按照操作指南使能或除能端点。
30	EPD	端点除能 对于OUT端点0，该位固定为0
29:28	保留	必须保留为复位值。
27	STALL	设置NAK 软件置1该位置1该寄存器的NAKS位
26	CNAK	清零NAK 软件置1该位清零该寄存器的NAKS位
25:22	保留	必须保留为复位值。

21	STALL	STALL握手 在OUT事务中，软件可以通过置1该位发送STALL握手包，对于OUT端点0，在接收SETUP令牌后，USBFS清除此位。该位比该寄存器的NAKS位和寄存器USBFS_DCTL的GINS位优先级要高，即如果STALL和NAKS位都被置位，STALL位生效。
20	SNOOP	调查模式 该位控制OUT端点的调查模式，在调查模式中，USBFS不再检查接收数据包的CRC值 0：调查模式除能 1：调查模式使能
19:18	EPTYPE[1:0]	端点类型 对于控制端点，该位固定为“00”
17	NAKS	NAK状态 当该寄存器的STALL位和寄存器USBFS_DCTL的位GINS被清零，该位控制USBFS的NAK状态： 0：根据端点Rx FIFO的状态，USBFS发送数据或握手包 1：USBFS为OUT事务发NAK握手包 该位是只读位，通过该寄存器的CNAK和SNAK位控制该位
16	保留	必须保留为复位值。
15	EPACT	端点激活 对于端点0，该域固定为1
14:2	保留	必须保留为复位值。
1:0	MPL[1:0]	最大包长 该位是只读位，其数值来自于寄存器USBFS_DIEP0CTL的位MPL： 00：64字节 01：32字节 10：16字节 11：8字节

设备 OUT 端点 x 控制寄存器（USBFS_DOEPxCTL）（x= 1..3，x 是端点编号）

地址偏移：0x0B00 + （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

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

EOFRM/DPID	NAKS	EPTYPE[1:0]		SNOOP	STALL	保留				CNAK	SNAK	SEVENFRM / SD0PID	SODDFRM/SD1 PID	EPD	EPEN		
r	r	rw		rw	rw/rs					w	w	w	w	rs	rs		
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
MPL[100]																保留	EPACT
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	必须保留为复位值。
21	STALL	STALL握手 在OUT事务中，软件可以通过置1该位发送STALL握手包。该位比该寄存器的NAKS位和寄存器USBFS_DCTL的GINS位优先级要高，如果STALL和NAKS位都被置

		位，STALL位生效。 对于控制OUT端点： 当OUT端点接收SETUP令牌时，只有USBFS可以清零该位，软件不可清零此位。 对于中断或大容量OUT端点 只有软件可以清零该位
20	SNOOP	调查模式 该位控制OUT端点的调查模式，在调查模式中，USBFS不再检查接收数据包的CRC值 0：调查模式除能 1：调查模式使能
19:18	EPTYPE[1:0]	端点类型 该域定义端点的传输类型 00：控制 01：同步 10：大容量 11：中断
17	NAKS	NAK状态 当该寄存器的STALL位和寄存器USBFS_DCTL的位GONS被清零，该位控制USBFS的NAK状态： 0：根据端点的Rx FIFO的状态，发送握手包 1：USBFS为OUT事务发送NAK握手 该位是只读位，通过该寄存器的CNAK和SNAK位控制该位
16	EOFRM	奇偶帧（适用于同步OUT端点） 对于同步传输，软件通过使用该位控制USBFS只在奇数帧或偶数帧发送数据包给OUT事务，如果当前帧号的奇偶性不匹配该位，USBFS不保存数据包 0：只在偶数帧发送数据 1：只在奇数帧发送数据
	DPID	端点数据PID（适用于中断或大容量端点） 在端点或大容量传输中，有数据PID翻转机制，在传输开始之前，软件通过设定SD0PID来设置此位，按照USB协议中描述的数据PID翻转机制，USBFS在传输过程中保持该位。 0：数据包PID是DATA0 1：数据包PID是DATA1
15	EPACT	端点激活 位控制端点是否激活，当端点没有激活，忽略任何令牌，不做任何回复
14:11	保留	必须保留为复位值。
10:0	MPL[10:0]	该位定义最大包长

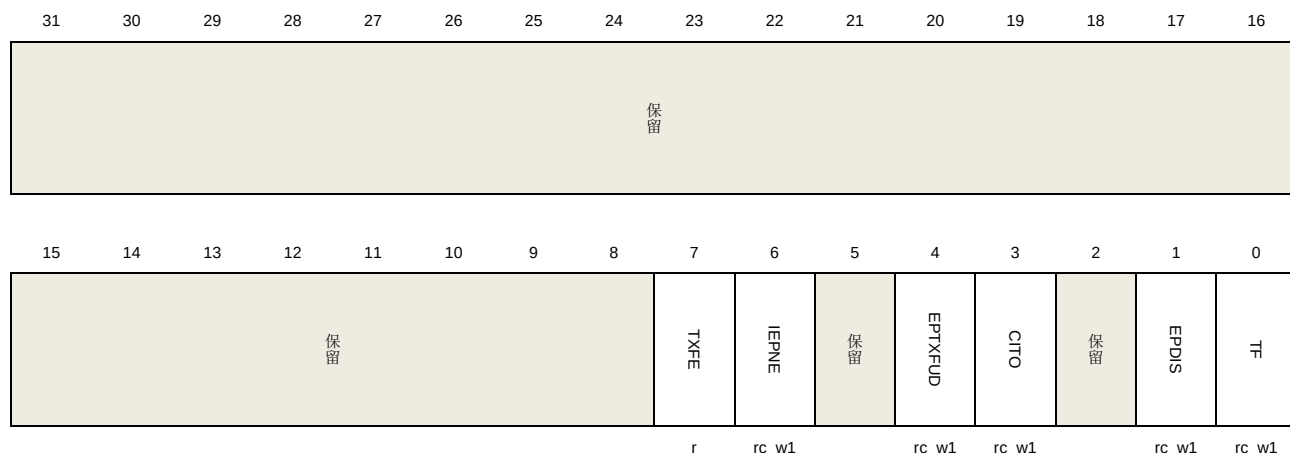
设备 IN 端点 x 中断标志寄存器（USBFS_DIEPxINTF）（x = 0..3，x 是端点编号）

地址偏移：0x0908 + （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，当获得一个 IN 端点的中断时，应该读取该端点的中断标志寄存器，从而获知中断源。该寄存器的标志位通常硬件置位，除了 TXFE 位，各位写 1 清零。

该寄存器采用字（32 位）访问



位/位域	名称	描述
31:8	保留	必须保留为复位值。
7	TXFE	发送FIFO空 端点的Tx FIFO达到寄存器USBFS_GAHBCS的位TXFTH定义的空阈值。
6	IEPNE	IN端点NAK有效 寄存器USBFS_DIEPxCTL的位SNAK的设置生效，该位可以通过写1清零或设置CNAK位
5	保留	必须保留为复位值。
4	EPTXFUD	端点Tx FIFO下溢 如果当IN令牌被接收后，Tx FIFO没有包数据，该标志被触发。
3	CITO	控制IN事务超时中断 在控制IN事务中，如果设备等待的握手包超时，该标志位被触发
2	保留	必须保留为复位值。
1	EPDIS	端点除能 端点除能时，该标志位被触发
0	TF	传输完成 当该端点的所有IN事务完成，该标志位被触发。

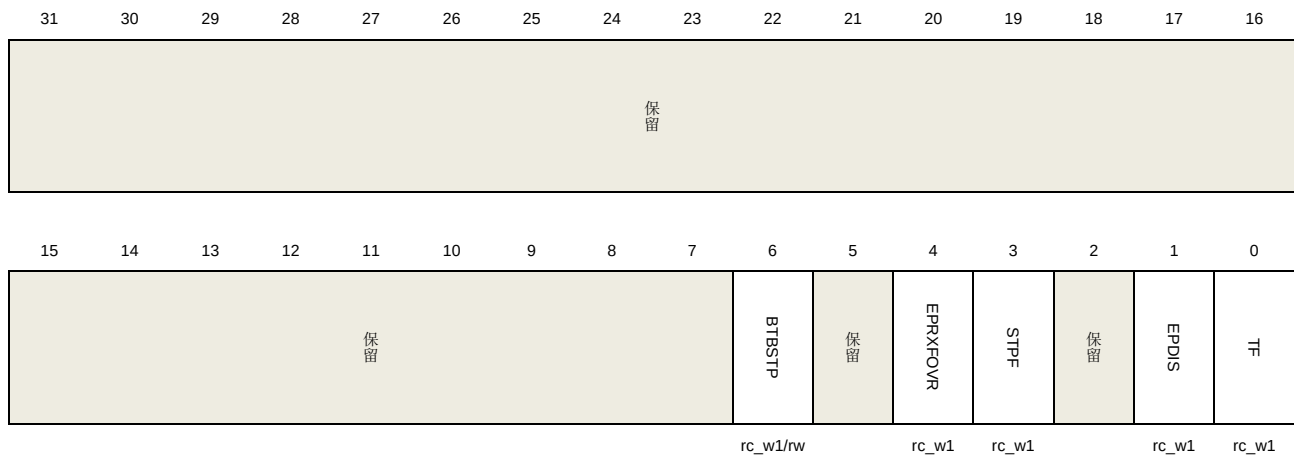
设备 OUT 端点 x 中断标志寄存器（USBFS_DOEPxINTF）（x = 0..3, x 是端点编号）

地址偏移：0x0B08 + （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，当获得一个 OUT 端点的中断时，应该读取该端点的中断标志寄存器，从而获知中断源。该寄存器的标志位通常硬件置位，各位写 1 清零。

该寄存器采用字（32 位）访问



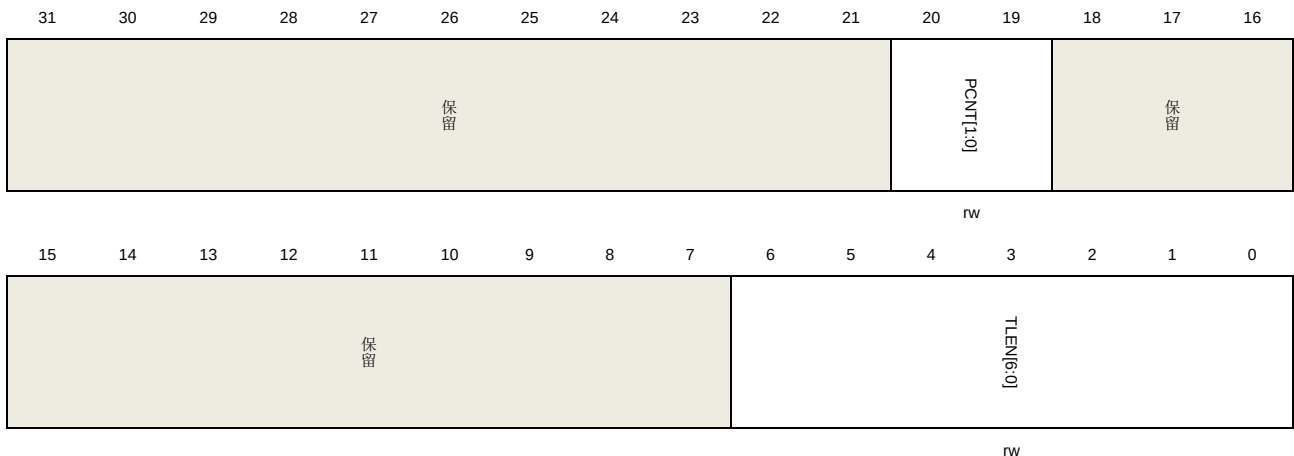
位/位域	名称	描述
31:7	保留	必须保留为复位值。
6	BTBSTP	连续SETUP包（适用于控制OUT端点） 当一个控制OUT端点接收超过连续3个SETUP包时，该标志被触发。
5	保留	必须保留为复位值。
4	EPRXFOVR	端点Rx FIFO上溢 当OUT令牌被接收时，如果OUT端点的Rx FIFO没有足够的空间存放数据包，该位被触发。在这种情况下，USBFS不能接收OUT数据包，发送一个NAK握手包。
3	STPF	SETUP阶段完成（适用于控制OUT端点） 当一个SETUP阶段完成，也就是USBFS在一个setup令牌后接收了一个IN或OUT令牌，该位被置位。
2	保留	必须保留为复位值。
1	EPDIS	端点除能 端点除能时，该标志位被触发
0	TF	传输完成 当该端点的所有OUT事务完成，该标志位被触发

设备 IN 端点 0 传输长度寄存器（USBFS_DIEP0LEN）

地址偏移：0x0910

复位值：0x0000 0000

该寄存器采用字（32 位）访问



位/位域	名称	描述
31:21	保留	必须保留为复位值。
20:19	PCNT[1:0]	包数 传输中被发送的数据包数量 在端点使能之前，软件设置该位，在传输开始后，该域在每次数据包成功发送后自动减少。
18:7	保留	必须保留为复位值。
6:0	TLEN[6:0]	传输长度 一次传输的数据总字节数 该域是IN传输中需要发送的包数据的总字节数，在端点使能之前，软件设置该位，在软件或DMA成功地将包数据写入端点的Tx FIFO中，该域减少与包数据大小相同的数值。

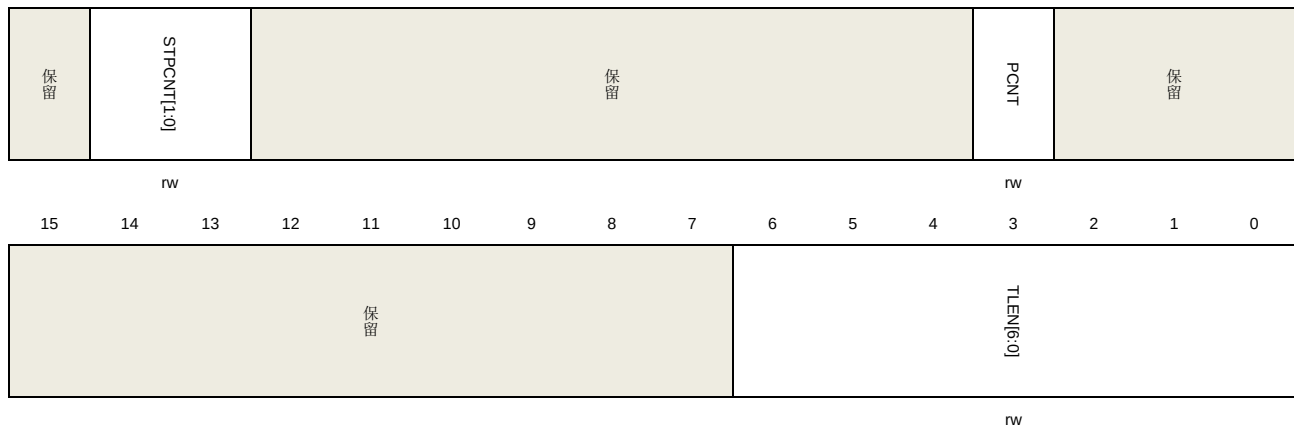
设备 OUT 端点 0 传输长度寄存器（USBFS_DOEP0LEN）

地址偏移：0x0B10

复位值：0x0000 0000

该寄存器采用字（32 位）访问





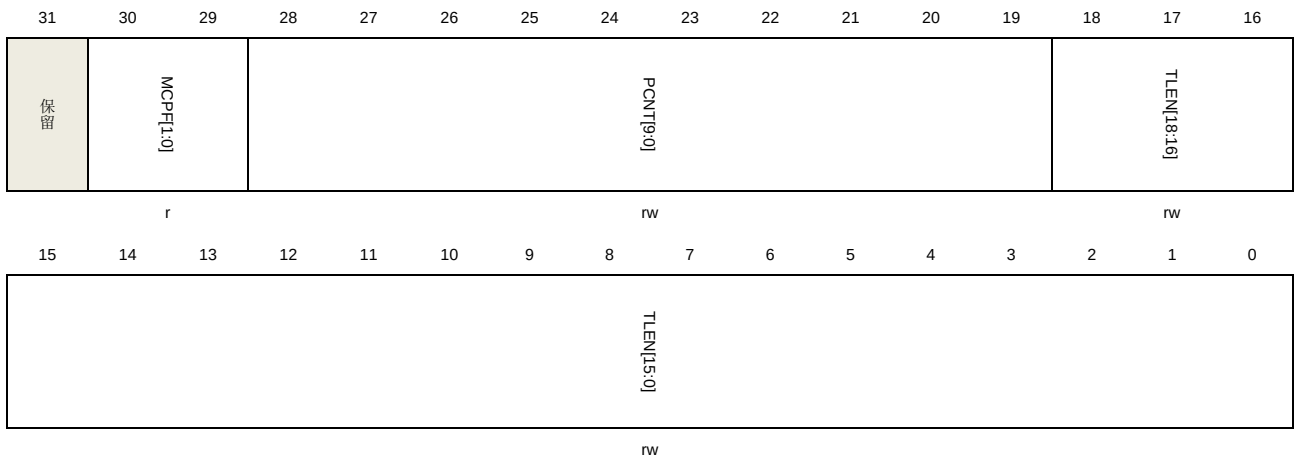
位/位域	名称	描述
31	保留	必须保留为复位值。
30:29	STPCNT[1:0]	SETUP包计数 该域定义端点可以接受的最大连续SETUP包数量 在SETUP传输之前，设置该域，每当连续SETUP包接收到时，该域值减1，当该域达到0时，寄存器USBFS_DOEP0INTF的BTBSTP标志被触发。 00: 0个包 01: 1个包 10: 2个包 11: 3个包
28:20	保留	必须保留为复位值。
19	PCNT	包计数 一次传输中应该接收到包数量。 在端点使能前，软件设置该位，在传输开始后，每当数据包接收到后，该域数值自动减少。
18:7	保留	必须保留为复位值。
6:0	TLEN[6:0]	传输长度 传输中数据总字数。 该域是OUT传输中需要接收的包数据的总字节数，在端点使能之前，软件设置该位，在软件或DMA成功地将包数据读取端点的Rx FIFO中，该域减少与包数据大小相同的数值。

设备 IN 端点 x 传输长度寄存器（USBFS_DIEPxLEN）（x = 1..3, x 是端点编号）

地址偏移：0x910 + （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	必须保留为复位值。
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自动减少。
18:0	TLEN[18:0]	传输长度 传输的数据总字节数 该域是IN传输中需要发送的包数据的总字节数，在端点使能之前，软件设置该位，在软件或DMA成功地将包数据写入端点的Tx FIFO中，该域减少与包数据大小相同的数值。

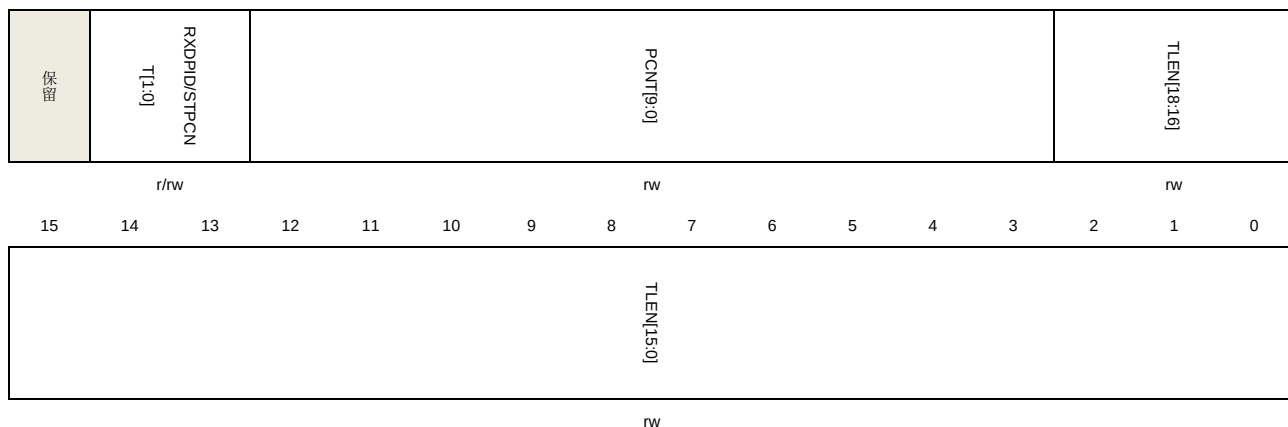
设备 OUT 端点 x 传输长度寄存器（USBFS_DOEPxLEN）（x = 1..3, x 是端点编号）

地址偏移：0x0B10 + （x × 0x20）

复位值：0x0000 0000

该寄存器采用字（32 位）访问





位/位域	名称	描述
31	保留	必须保留为复位值。
30:29	RXDPID[1:0]	接收数据PID（适用于同步OUT端点） 该域保存该端点该数据包所接受的最后一个数据包的PID 00: DATA0 10: DATA1 其他: 保留
	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BTBSTP标志被触发。 00: 0个包 01: 1个包 10: 2个包 11: 3个包
28:19	PCNT[9:0]	包数 传输中应该接收到包数量 在端点使能前，软件设置该位，在传输开始后，每当数据包接收到后，该域数值自动减少。
18:0	TLEN[18:0]	传输长度 传输中数据总字数 该域是IN传输中需要接收的包数据的总字节数，在端点使能之前，软件设置该位，在软件或DMA成功地将包数据读取端点的Rx FIFO中，该域减少与包数据大小相同的数值。

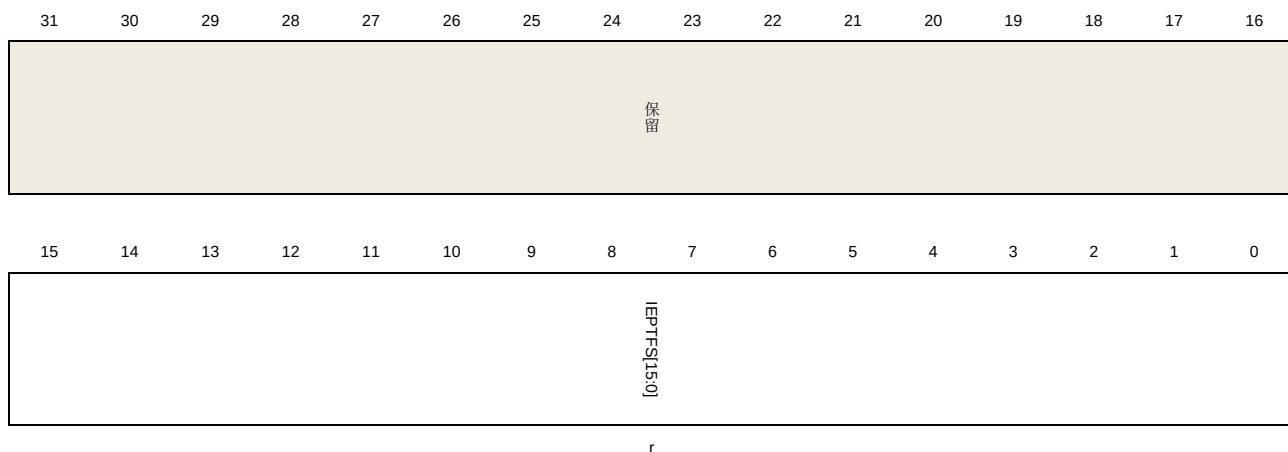
设备 IN 端点 x 发送 FIFO 状态寄存器（USBFS_DIEPxTFSTAT）（ $x = 0..3$, x 是端点编号）

地址偏移：0x0918 + （端点编号 $\times$ 0x20）

复位值：0x0000 0200

该寄存器包含每个端点的 Tx FIFO 的信息。

该寄存器采用字（32 位）访问



位/位域	名称	描述
31:16	保留	必须保留为复位值。
15:0	IEPTFS[15:0]	IN端点的Tx FIFO可用空间 IN端点的Tx FIFO可用空间用32位字为单位 0: FIFO是满的 1: 1字可用 ... n: n字可用

28.7.4. 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



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

位/位域	名称	描述
31:2	保留	必须保留为复位值。
1	SHCLK	停止HCLK 停止HCLK，节省电量 0: HCLK未停止 1: HCLK停止
0	SUCLK	停止USB时钟 停止USB时钟，节省电量 0: USB时钟未停止 1: USB时钟停止

29. 附录

29.1. 寄存器表中使用的缩写列表

表 29-1. 寄存器功能位访问属性

功能位访问属性	描述
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/写(rw)	软件可以对这个位进行读写。
读/写一次(rwo)	软件只可写该位一次，可以读任意次。只有复位可将该位恢复为默认值。
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。
读/写清零(rc_w)	软件可以读该位，对该位写可以清除这个位。写 0 和写 1 效果相同。
写清零(c_w)	软件只可对该位写清除。写 0 和写 1 效果相同。
写 1 清除(c_w1)	软件只可对该位写入 1 清除这个位。写入 0 对位值没有影响。
可读/读清零(rc_r)	软件可以读该位，读该位会自动清零。写入 0 对位值没有影响。
可读/可置位(rs)	软件可以读该位，也可以将这个位设置为 1。写入 0 对位值没有影响。
可读/读置位(rs_r)	软件可以读该位，读该位会置位该位。写对位值没有影响。
可写/读置位(ws_r)	软件可以写该位，读该位会置位该位。
可读/写置位(rs_w)	软件可以读该位，写该位会置位该位。写 0 和写 1 效果相同。
写置位(s_w)	软件只可写置位该位。写 0 和写 1 效果相同。
可读/写 0 置位 (rs_w0)	软件可以读该位，写 0 会置位该位。写入 1 对位值没有影响。
只读/写触发(rt_w)	可读，可写 0/1 触发，但对该位值无影响。
只读/写 1 触发 (rt_w1)	软件只能读该位，写入 1 触发事件，但对位值没有影响。
只读/写 0 触发 (rt_w0)	软件只能读该位，写入 0 触发事件，但对位值没有影响。
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。

29.2. 术语表

表 29-2. 术语

术语	描述
字	32 位长度数据
半字	16 位长度数据。

术语	描述
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

29.3. 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。

30. 版本历史

表 30-1. 版本历史

版本号	描述	日期
1.0	首次发布	2025 年 10 月 31 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 according to the laws of the People's Republic of China and other applicable laws. The Company reserves all rights under such laws and no Intellectual Property Rights are transferred (either wholly or partially) or licensed by the Company (either expressly or impliedly) herein. 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.

To the maximum extent permitted by applicable law, the Company makes no representations or warranties of any kind, express or implied, with regard to the merchantability and the fitness for a particular purpose of the Product, nor does the Company assume any liability arising out of the application or use of any Product. Any information provided in this document is provided only for reference purposes. It is the sole responsibility of the user of this document to determine whether the Product is suitable and fit for its applications and products planned, and properly design, program, and test the functionality and safety of its applications and products planned using the Product. The Product is designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only, and the Product is not designed or intended for use in (i) safety critical applications such as weapons systems, nuclear facilities, atomic energy controller, combustion controller, aeronautic or aerospace applications, traffic signal instruments, pollution control or hazardous substance management; (ii) life-support systems, other medical equipment or systems (including life support equipment and surgical implants); (iii) automotive applications or environments, including but not limited to applications for active and passive safety of automobiles (regardless of front market or aftermarket), for example, EPS, braking, ADAS (camera/fusion), EMS, TCU, BMS, BSG, TPMS, Airbag, Suspension, DMS, ICMS, Domain, ESC, DCDC, e-clutch, advanced-lighting, etc.. Automobile herein means a vehicle propelled by a self-contained motor, engine or the like, such as, without limitation, cars, trucks, motorcycles, electric cars, and other transportation devices; and/or (iv) other uses where the failure of the device or the Product can reasonably be expected to result in personal injury, death, or severe property or environmental damage (collectively "Unintended Uses"). Customers shall take any and all actions to ensure the Product meets the applicable laws and regulations. The Company is not liable for, in whole or in part, and customers shall hereby 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 Product. Customers shall indemnify and hold the Company, and its officers, employees, subsidiaries, affiliates 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 Product.

Information in this document is provided solely in connection with the Product. The Company reserves the right to make changes, corrections, modifications or improvements to this document and the Product described herein at any time without notice. The Company shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. Information in this document supersedes and replaces information previously supplied in any prior versions of this document.