

GigaDevice Semiconductor Inc.

GD32M531

ARM® Cortex®-M33 32-bit MCU

适用于 GD32M531

用户手册

1.0 版本

(2026 年 3 月)

目 录

目 录	2
图索引	20
表索引	27
1. 系统和内存架构.....	30
1.1. ARM Cortex-M33 处理器	30
1.2. 系统架构.....	31
1.3. 内存映射.....	33
1.3.1. 片上 SRAM 存储器.....	36
1.3.2. 片上 FLASH 存储器概述	38
1.4. 启动配置.....	38
1.5. 系统配置控制器	38
1.6. 系统配置寄存器	40
1.6.1. 配置寄存器 0 (SYSCFG_CFG0)	40
1.6.2. 配置寄存器 1 (SYSCFG_CFG1)	40
1.6.3. EXTI 源选择寄存器 0 (SYSCFG_EXTISS0)	41
1.6.4. EXTI 源选择寄存器 1 (SYSCFG_EXTISS1)	42
1.6.5. EXTI 源选择寄存器 2 (SYSCFG_EXTISS2)	42
1.6.6. EXTI 源选择寄存器 3 (SYSCFG_EXTISS3)	43
1.6.7. 状态寄存器 (SYSCFG_STAT)	44
1.6.8. 配置寄存器 2 (SYSCFG_CFG2)	46
1.6.9. 配置寄存器 3 (SYSCFG_CFG3)	47
1.6.10. 配置寄存器 4 (SYSCFG_CFG4)	48
1.6.11. TIMER 输入源选择寄存器 0 (SYSCFG_TIMERCISEL0)	49
1.6.12. TIMER 输入源选择寄存器 1 (SYSCFG_TIMERCISEL1)	50
1.6.13. 配置寄存器 3 (SYSCFG_CFG5)	51
1.6.14. SRAM 配置寄存器 (SYSCFG_SRAMCFG)	52
1.6.15. 保护配置寄存器 (SYSCFG_PRCFG)	53
1.6.16. TIMERx ADC 启动请求配置寄存器 (TADSRCFG)	54
1.6.17. TIMERx 配置寄存器 0 (SYSCFG_TIMERxCFG0, x=0,7)	55
1.6.18. TIMERx 配置寄存器 1 (SYSCFG_TIMERxCFG1, x=0,7)	57
1.6.19. TIMERx 配置寄存器 2 (SYSCFG_TIMERxCFG2, x=0,7)	59
1.6.20. TIMERx 配置寄存器 0 (SYSCFG_TIMERxCFG0, x=1,2)	60
1.6.21. TIMERx 配置寄存器 1 (SYSCFG_TIMERxCFG1, x=1,2)	61
1.6.22. TIMERx 配置寄存器 2 (SYSCFG_TIMERxCFG2, x=1,2)	63
1.6.23. 配置寄存器 6 (SYSCFG_CFG6)	63
1.1. 电子设备签名.....	64
1.1.1. 存储密度信息	64

1.1.2.	设备唯一 ID (96 位)	64
2.	闪存控制器 (FMC)	66
2.1.	简介	66
2.2.	主要特征	66
2.3.	功能说明	66
2.3.1.	闪存结构	66
2.3.2.	错误检查与纠正 (ECC)	67
2.3.3.	读操作	68
2.3.4.	FMC_CTL 寄存器解锁	69
2.3.5.	页擦除	70
2.3.6.	整片擦除	71
2.3.7.	闪存块编程	72
1.1.3.	闪存块快速编程	74
2.3.8.	数据闪存操作	77
2.3.9.	查空命令	77
2.3.10.	OTP 编程	77
2.3.11.	选项字节擦除	78
2.3.12.	选项字节编程	78
2.3.13.	选项字节说明	78
2.3.14.	页擦除/编程保护	81
2.3.15.	安全保护	82
2.4.	FMC 寄存器	84
2.4.1.	等待状态寄存器 (FMC_WS)	84
2.4.2.	解锁寄存器 (FMC_KEY)	85
2.4.3.	选项字节操作解锁寄存器 (FMC_OBKEY)	85
2.4.4.	状态寄存器 (FMC_STAT)	86
2.4.5.	控制寄存器 (FMC_CTL)	86
2.4.6.	地址寄存器 (FMC_ADDR)	88
2.4.7.	编程数据寄存器 (FMC_DATA)	89
2.4.8.	ECC 控制及状态寄存器 (FMC_ECCCS)	89
2.4.9.	选项字节状态寄存器 (FMC_OBSTAT)	91
2.4.10.	擦除/编程保护寄存器 (FMC_WP)	91
2.4.11.	产品 ID 寄存器 0 (FMC_PID0)	92
2.4.12.	产品 ID 寄存器 1 (FMC_PID1)	92
3.	电源管理单元 (PMU)	93
3.1.	简介	93
3.2.	主要特征	93
3.3.	功能说明	94
3.3.1.	V _{DD} / V _{DDA} 电源域	94
3.3.2.	V _{CORE} 电源域	95

3.3.3.	省电模式	95
3.3.4.	低压差电压检测器	97
3.4.	PMU 寄存器	100
3.4.1.	控制寄存器 (PMU_CTL)	100
3.4.2.	电源控制和状态寄存器 (PMU_CS)	101
3.4.3.	低电压检测器 1 控制寄存器 (PMU_LVD1CTL)	102
3.4.4.	低电压检测器 2 控制寄存器 (PMU_LVD2CTL)	103
4.	复位和时钟单元 (RCU)	105
4.1.	复位控制单元 (RCTL)	105
4.1.1.	简介	105
4.1.2.	功能描述	105
4.2.	时钟控制单元 (CCTL)	106
4.2.1.	简介	106
4.2.2.	主要特性	107
4.2.3.	功能描述	108
4.2.4.	事件互连的操作	110
4.3.	RCU 寄存器	112
4.3.1.	控制寄存器 (RCU_CTL)	112
4.3.2.	配置寄存器 0 (RCU_CFG0)	114
4.3.3.	中断寄存器 (RCU_INT)	116
4.3.4.	APB2 复位寄存器 (RCU_APB2RST)	119
4.3.5.	APB1 复位寄存器 (RCU_APB1RST)	121
4.3.6.	AHB 使能寄存器 (RCU_AHBEN)	122
4.3.7.	APB2 使能寄存器 (RCU_APB2EN)	125
4.3.8.	APB1 使能寄存器 (RCU_APB1EN)	127
4.3.9.	复位源/时钟寄存器 (RCU_RSTSCK)	129
4.3.10.	AHB 复位寄存器 (RCU_AHBRSRST)	132
4.3.11.	配置寄存器 1 (RCU_CFG1)	134
4.3.12.	配置寄存器 2 (RCU_CFG2)	134
5.	时钟频率测量单元 (CFMU)	136
5.1.	概述	136
5.2.	特性	136
5.3.	功能概述	136
5.3.1.	CFMU 框图	136
5.3.2.	时钟频率测量	136
5.3.3.	CFMUREF 引脚上的数字滤波器功能	137
5.3.4.	CFMU 中断	137
5.4.	寄存器定义	138
5.4.1.	控制寄存器 (CFMU_CTL)	138
5.4.2.	状态寄存器 (CFMU_STAT)	140

5.4.3.	电流限制值配置寄存器 (CFMU_LVCFG)	140
5.4.4.	计数器值寄存器 (CFMU_CNT)	141
6.	中断/事件控制器 (EXTI)	142
6.1.	简介	142
6.2.	主要特性	142
6.3.	中断功能描述	142
6.4.	外部中断及事件(EXTI)框图	146
6.5.	外部中断及事件功能概述	146
6.6.	EXTI 寄存器	149
6.6.1.	中断使能寄存器(EXTI_INTEN)	149
6.6.2.	事件使能寄存器(EXTI_EVEN)	149
6.6.3.	上升沿触发使能寄存器(EXTI_RTEN)	150
6.6.4.	下降沿触发使能寄存器(EXTI_FTEN)	150
6.6.5.	软件中断事件寄存器(EXTI_SWIEV)	150
6.6.6.	挂起寄存器(EXTI_PD)	151
6.6.7.	数字滤波使能寄存器(EXTI_DFEN)	151
6.6.8.	采样时钟选择寄存器(EXTI_SCS)	152
7.	通用和备用输入/输出接口 (GPIO 和 AFIO)	153
7.1.	简介	153
7.2.	主要特征	153
7.3.	功能说明	153
7.3.1.	GPIO 引脚配置	155
7.3.2.	外部中断/事件线	155
7.3.3.	备用功能(AF)	155
7.3.4.	附加功能	155
7.3.5.	输入配置	155
7.3.6.	输出配置	156
7.3.7.	模拟配置	157
7.3.8.	备用功能(AF)配置	157
7.3.9.	GPIO 锁定功能	158
7.3.10.	GPIO 单周期输出翻转功能	158
7.3.11.	GPIO 引脚电平保持功能	158
7.4.	GPIO 寄存器	159
7.4.1.	端口控制寄存器 (GPIOx_CTL, x=A..G, N)	159
7.4.2.	端口输出模式寄存器 (GPIOx_OMODE, x=A..G, N)	161
7.4.3.	端口输出速度寄存器 (GPIOx_OSPD, x=A..G, N)	162
7.4.4.	端口上拉/下拉寄存器 (GPIOx_PUD, x=A..G, N)	164
7.4.5.	端口输入状态寄存器 (GPIOx_ISTAT, x=A..G, N)	166
7.4.6.	端口输出控制寄存器 (GPIOx_OCTL, x=A..G, N)	166

7.4.7.	端口位操作寄存器 (GPIOx_BOP, x=A..G, N)	167
7.4.8.	端口配置锁定寄存器 (GPIOx_LOCK, x=A..G, N)	167
7.4.9.	备用功能选择寄存器 0 (GPIOx_AFSEL0, x=A..G, N)	168
7.4.10.	备用功能选择寄存器 1 (GPIOx_AFSEL1, x=A..G, N)	169
7.4.11.	位清除寄存器 (GPIOx_BC, x=A..G, N)	170
7.4.12.	端口位翻转寄存器 (GPIOx_TG, x=A..G, N)	171
7.4.13.	引脚电平保持配置寄存器 (GPIOx_HLDCFG, x=A..G, N)	171
7.4.14.	引脚功能选择寄存器 (GPIOx_FUNCSEL, x=N)	172
7.4.15.	引脚电平保持控制寄存器 (GPIOx_HLDCTL, x=N)	172
8.	端口输出控制器 (POC)	174
8.1.	简介	174
8.2.	主要特征	174
8.3.	功能描述	174
8.3.1.	POC 时钟	176
8.3.2.	输出禁能功能	176
8.3.3.	目标定时器选择	179
8.3.4.	输出切换	180
8.3.5.	检测屏蔽功能	180
8.3.6.	POC 中断	181
8.4.	POC 寄存器	182
8.4.1.	状态寄存器 0 (POC_STAT0)	182
8.4.2.	输入 n 检测配置寄存器 (POC_INnDCFG) (n=0..5)	184
8.4.3.	控制寄存器 0 (POC_CTL0)	186
8.4.4.	软件禁能请求生成寄存器 (POC_SWDRG)	186
8.4.5.	互补检测配置寄存器 0 (POC_CDCFG0)	187
8.4.6.	互补检测配置寄存器 1 (POC_CDCFG1)	189
8.4.7.	输出禁能模式寄存器 0 (POC_ODMODE0)	191
8.4.8.	输出禁能模式寄存器 1 (POC_ODMODE1)	193
8.4.9.	请求选择寄存器 0 (POC_REQSEL0)	195
8.4.10.	请求选择寄存器 1 (POC_REQSEL1)	198
8.4.11.	状态寄存器 1 (POC_STAT1)	204
8.4.12.	控制寄存器 1 (POC_CTL1)	205
8.4.13.	扩展控制寄存器 0 (POC_EXTCTL0)	207
8.4.14.	扩展控制寄存器 1 (POC_EXTCTL1)	209
8.4.15.	输入 n 检测屏蔽寄存器 (POC_INnDMK) (n=0..5)	212
8.4.16.	比较器 n 检测屏蔽寄存器 (POC_CMPnDMK) (n=0..3)	213
9.	GPTIMER 输出控制器 (GTOC)	214
9.1.	简介	214
9.2.	主要特征	214
9.3.	功能描述	214

9.3.1.	GTOC 时钟.....	215
9.3.2.	输出关闭功能	215
9.3.3.	GTOC 中断.....	218
9.4.	GTOC 寄存器	219
9.4.1.	GTOCx 配置寄存器 (GTOCx_CFG) (x=0...3)	219
9.4.2.	GTOCx 输出关闭请求控制寄存器 (GTOCx_OCRCTL) (x=0...3)	221
9.4.3.	GTOCx 写保护寄存器 (GTOCx_WP) (x=0...3)	222
9.4.4.	GTOCx 扩展关闭请求控制寄存器 (GTOCx_ECRCTL) (x=0...3)	222
10.	循环冗余校验管理单元 (CRC)	224
10.1.	简介.....	224
10.2.	主要特性.....	224
10.3.	功能描述.....	225
10.4.	CRC 寄存器.....	226
10.4.1.	数据寄存器 (CRC_DATA)	226
10.4.2.	独立数据寄存器 (CRC_FDATA)	226
10.4.3.	控制寄存器 (CRC_CTL)	227
10.4.4.	初值寄存器 (CRC_IDATA)	228
10.4.5.	多项式寄存器 (CRC_POLY)	228
11.	三角函数加速器 (TMU)	229
11.1.	简介.....	229
11.2.	主要特征.....	229
11.3.	结构框图.....	229
11.4.	功能描述.....	230
11.4.1.	数据格式和配置.....	230
11.4.2.	浮点数据格式	231
11.4.3.	模式配置.....	231
11.4.4.	TMU 运算挂起.....	238
11.4.5.	零开销	238
11.4.6.	TMU 指令集	238
11.4.7.	中断和 DMA	238
11.5.	TMU 寄存器.....	239
11.5.1.	控制和状态寄存器 (TMU_CS)	239
11.5.2.	输入数据寄存器 (TMU_IDATA)	241
11.5.3.	输出数据寄存器 (TMU_ODATA)	242
12.	直接存储器访问控制器 (DMA)	243
12.1.	简介.....	243
12.2.	主要特征.....	243

12.3.	结构框图.....	244
12.4.	功能说明.....	244
12.4.1.	DMA 操作.....	244
12.4.2.	外设握手.....	245
12.4.3.	仲裁.....	246
12.4.4.	地址生成.....	246
12.4.5.	循环模式.....	246
12.4.6.	存储器到存储器模式	247
12.4.7.	通道配置.....	247
12.4.8.	中断.....	247
12.4.9.	DMA 请求映射	248
12.5.	DMA 寄存器	249
12.5.1.	中断标志位寄存器 (DMA_INTF)	249
12.5.2.	中断标志位清除寄存器 (DMA_INTC)	249
12.5.3.	通道 x 控制寄存器 (DMA_CHxCTL)	250
12.5.4.	通道 x 计数寄存器 (DMA_CHxCNT)	252
12.5.5.	通道 x 外设基地址寄存器 (DMA_CHxPADDR)	253
12.5.6.	通道 x 存储器基地址寄存器 (DMA_CHxMADDR)	253
13.	DMA 请求多路复用器 (DMAMUX)	255
13.1.	简介.....	255
13.2.	主要特征.....	255
13.3.	结构框图.....	256
13.4.	功能说明.....	256
13.4.1.	DMAMUX 信号	257
13.4.2.	DMAMUX 请求路由器	257
13.4.3.	DMAMUX 请求生成器	259
13.4.4.	通道配置.....	260
13.4.5.	中断.....	260
13.4.6.	DMAMUX 映射	261
13.5.	DMAMUX 寄存器	266
13.5.1.	请求路由通道 x 配置寄存器 (DMAMUX_RM_CHxCFG)	266
13.5.2.	请求路由通道中断标志位寄存器 (DMAMUX_RM_INTF)	267
13.5.3.	请求路由通道中断标志位清除寄存器 (DMAMUX_RM_INTC)	267
13.5.4.	请求生成通道 x 配置寄存器 (DMAMUX_RG_CHxCFG)	268
13.5.5.	请求生成通道中断标志位寄存器 (DMAMUX_RG_INTF)	269
13.5.6.	请求生成通道中断标志位清除寄存器 (DMAMUX_RG_INTC)	269
14.	调试 (DBG)	270
14.1.	简介.....	270
14.2.	JTAG/SW 功能说明.....	270

14.2.1.	切换 JTAG/ SW 接口	270
14.2.2.	引脚分配.....	270
14.2.3.	JTAG 链状结构.....	271
14.2.4.	调试复位.....	271
14.2.5.	JEDEC-106 ID 代码	271
14.3.	调试保持功能说明	271
14.3.1.	低功耗模式调试支持	271
14.3.2.	TIMER, I2C, WWDGT, FWDGT 和 CAN 外设调试支持	272
14.4.	DBG 寄存器	273
14.4.1.	ID 寄存器 (DBG_ID).....	273
14.4.2.	控制寄存器 (DBG_CTL).....	273
15.	模数转换器 (ADC)	276
15.1.	简介.....	276
15.2.	主要特征.....	276
15.3.	引脚和内部信号	277
15.4.	结构框图.....	278
15.5.	功能说明.....	279
15.5.1.	ADC 时钟	279
15.5.2.	ADCON 使能	279
15.5.3.	扫描运行模式	280
15.5.4.	Group_pri1 组单次扫描模式.....	280
15.5.5.	Group_pri1 组连续扫描模式.....	287
15.5.6.	多组扫描模式	294
15.5.7.	通道优先级配置功能	315
15.5.8.	转换结果阈值监测功能	316
15.5.9.	数据存储模式	317
15.5.10.	采样时间配置.....	322
15.5.11.	触发配置.....	322
15.5.12.	DMA 功能	323
15.5.13.	DMA 溢出检测	324
15.5.14.	ADC 内部通道	325
15.5.15.	可编程分辨率 (DRES)	326
15.5.16.	断开辅助检测功能 (预充放电)	327
15.5.17.	自诊断	327
15.5.18.	采样保持功能 (仅 ADC0)	328
15.5.19.	通道专用 AD 转换加法功能 (数据累加求和/平均模式)	329
15.5.20.	数据寄存器自动清零/置位	329
15.5.21.	ADC 中断	330
15.6.	ADC 寄存器.....	330
15.6.1.	软件 EOC 控制寄存器 (ADC_EOCCTL)	330

15.6.2.	控制寄存器 0 (ADC_CTL0)	333
15.6.3.	组配置寄存器 0 (ADC_GPCGF0)	336
15.6.4.	组配置寄存器 1 (ADC_GPCGF1)	337
15.6.5.	通道选择寄存器 0 (ADC_CHSEL0)	340
15.6.6.	通道选择寄存器 1 (ADC_CHSEL1)	342
15.6.7.	采样时间寄存器 0 (ADC_SAMPR0)	344
15.6.8.	采样时间寄存器 1 (ADC_SAMPR1)	345
15.6.9.	采样时间寄存器 2 (ADC_SAMPR2)	345
15.6.10.	通道优先级寄存器 0 (ADC_CHPRI0)	346
15.6.11.	通道优先级寄存器 1 (ADC_CHPRI1)	347
15.6.12.	通道加法次数寄存器 0 (ADC_ADDT0)	348
15.6.13.	通道加法次数寄存器 1 (ADC_ADDT1)	349
15.6.14.	自诊断数据寄存器 (ADC_SDDATA)	350
15.6.15.	Group_pri x 组分叉数据寄存器 (ADC_GPxBIDATA, x=1,2,3,4)	351
15.6.16.	Group_pri x 组分叉数据寄存器 1 (ADC_GPxBIDATA1, x=1,2,3,4)	351
15.6.17.	Group_pri x 组分叉数据寄存器 2 (ADC_GPxBIDATA2, x=1,2,3,4)	352
15.6.18.	Group_pri x 组 DMA 数据寄存器 (ADC_GPxDMA, x=1,2,3,4)	352
15.6.19.	通道 x 数据寄存器 (ADC_CHxDATA)	353
15.6.20.	温度传感器通道数据寄存器 (ADC_TEMPDATA)	353
15.6.21.	内部参考电压通道数据寄存器 (ADC_VINTDATA)	354
15.6.22.	看门狗控制寄存器 (ADC_WDCTL)	354
15.6.23.	模拟看门狗 A 阈值寄存器 (ADC_WDATHOLD)	356
15.6.24.	模拟看门狗 B 阈值寄存器 (ADC_WDATHOLD)	356
15.6.25.	模拟看门狗 A 通道配置寄存器 (ADC_WDACHCFG)	357
15.6.26.	模拟看门狗 A 通道状态寄存器 (ADC_WDACHSTAT)	359
15.6.27.	控制寄存器 1 (ADC_CTL1)	361
15.6.28.	采样保持控制寄存器 (ADC_SHCTL)	361
15.6.29.	DMA 控制寄存器 (ADC_DMACTL)	362
15.6.30.	分叉触发控制寄存器 (ADC_BITRGCTL)	365
16.	数模转换器 (DAC)	368
16.1.	简介.....	368
16.2.	主要特性.....	368
16.3.	功能描述.....	369
16.3.1.	DAC 使能	369
16.3.2.	DAC 数据配置	370
16.3.3.	DAC 触发	370
16.3.4.	DAC 转换	370
16.3.5.	DAC 噪声波.....	371
16.3.6.	DAC 输出电压	372
16.3.7.	D/A 和 A/D 防干扰.....	372
16.3.8.	DAC 并发转换	372
16.3.9.	DAC 低功耗模式	373

16.4. DAC 寄存器.....	374
16.4.1. DACx 控制寄存器 0 (DAC_CTL0)	374
16.4.2. DACx 软件触发寄存器 (DAC_SWT)	376
16.4.3. DACx_OUT0 12 位右对齐数据保持寄存器 (DAC_OUT0_R12DH)	377
16.4.4. DACx_OUT0 12 位左对齐数据保持寄存器 (DAC_OUT0_L12DH)	377
16.4.5. DACx_OUT1 12 位右对齐数据保持寄存器 (DAC_OUT1_R12DH)	378
16.4.6. DACx_OUT1 12 位左对齐数据保持寄存器 (DAC_OUT1_L12DH)	378
16.4.7. DACx 并发模式 12 位右对齐数据保持寄存器 (DACC_R12DH)	379
16.4.8. DACx 并发模式 12 位左对齐数据保持寄存器 (DACC_L12DH)	379
16.4.9. DACx_OUT0 数据输出寄存器 (DAC_OUT0_DO)	380
16.4.10. DACx_OUT1 数据输出寄存器 (DAC_OUT1_DO)	380
17. 比较器 (CMP).....	382
17.1. 简介.....	382
17.2. 主要特征.....	382
17.3. 功能描述.....	382
17.3.1. 比较器时钟	384
17.3.2. 比较器的 I/O 配置.....	384
17.3.3. 比较器迟滞	385
17.3.4. 比较器寄存器写保护	385
17.3.5. 比较器中断.....	385
17.3.6. 比较器数字滤波.....	385
17.3.7. 比较器省电模式.....	386
17.4. 比较器寄存器	387
17.4.1. 状态寄存器 (CMP_STAT)	387
17.4.2. 中断标志位清除寄存器 (CMP_IFC)	388
17.4.3. CMP0 控制状态寄存器 (CMP0_CS)	389
17.4.4. CMP1 控制状态寄存器 (CMP1_CS)	391
17.4.5. CMP2 控制状态寄存器 (CMP2_CS)	393
17.4.6. CMP3 控制状态寄存器 (CMP3_CS)	395
18. 事件互连单元 (EVIC).....	398
18.1. 简介.....	398
18.2. 主要特征.....	398
18.3. 功能描述.....	398
18.3.1. 事件源选择.....	398
18.3.2. I/O 引脚的事件互连.....	401
18.4. EVIC 寄存器.....	406
18.4.1. 软件事件寄存器 (EVIC_SWEV)	406
18.4.2. 单个 I/O 事件互连寄存器 x (EVIC_SGIOx) (x=0...3)	406
18.4.3. 输入组 E 事件互连寄存器 (EVIC_INGRPE)	407
18.4.4. 输入组 F 事件互连寄存器 (EVIC_INGRPF)	408

18.4.5.	输出组 E 事件互连寄存器 (EVIC_OUTGRPE)	408
18.4.6.	输出组 F 事件互连寄存器 (EVIC_OUTGRPF)	409
18.4.7.	DAC0 转换事件互连寄存器 (EVIC_DAC0COV)	409
18.4.8.	ADC0 转换事件互连寄存器 (EVIC_ADC0COV)	410
18.4.9.	ADC2 转换事件互连寄存器 (EVIC_ADC2COV)	411
18.4.10.	RCU 事件互连寄存器 (EVIC_RCU)	411
18.4.11.	TIMER0 事件互连寄存器 (EVIC_TIMER0)	412
18.4.12.	TIMER7 事件互连寄存器 (EVIC_TIMER7)	412
18.4.13.	CPTIMER0 事件互连寄存器 (EVIC_CPTIMER0)	413
18.4.14.	CPTIMERW 事件互连寄存器 (EVIC_CPTIMERW)	414
18.4.15.	TIMER1 事件互连寄存器 (EVIC_TIMER1)	414
18.4.16.	TIMER2 事件互连寄存器 (EVIC_TIMER2)	415
18.4.17.	GPTIMER 事件互连寄存器 0 (EVIC_GPTIMER_0)	415
18.4.18.	GPTIMER 事件互连寄存器 1 (EVIC_GPTIMER_1)	416
18.4.19.	GPTIMER 事件互连寄存器 2 (EVIC_GPTIMER_2)	417
18.4.20.	GPTIMER 事件互连寄存器 3 (EVIC_GPTIMER_3)	417
18.4.21.	DAC0 使能事件互连寄存器 (EVIC_DAC0EN)	418
18.4.22.	DMAMUX 事件互连寄存器 0 (EVIC_DMAMUX_0)	419
18.4.23.	DMAMUX 事件互连寄存器 1 (EVIC_DMAMUX_1)	419
18.4.24.	DMAMUX 事件互连寄存器 2 (EVIC_DMAMUX_2)	420
18.4.25.	DMAMUX 事件互连寄存器 3 (EVIC_DMAMUX_3)	421
18.4.26.	DMAMUX 事件互连寄存器 4 (EVIC_DMAMUX_4)	421
18.4.27.	DMAMUX 事件互连寄存器 5 (EVIC_DMAMUX_5)	422
18.4.28.	从模式配置寄存器 0 (EVIC_SMCFG0)	423
18.4.29.	从模式配置寄存器 1 (EVIC_SMCFG1)	423
18.4.30.	单个 I/O 配置寄存器 x (EVIC_SGIOCFGx) (x=0..3)	424
18.4.31.	组 E 配置寄存器 (EVIC_GRPECFG)	425
18.4.32.	组 E 数据保持寄存器 (EVIC_GRPEDH)	426
18.4.33.	组 F 配置寄存器 (EVIC_GPRFCFG)	427
18.4.34.	组 F 数据保持寄存器 (EVIC_GRPFDH)	428
19.	空间矢量脉宽调制 (SVPWM)	430
19.1.	简介	430
19.2.	主要特征	430
19.3.	结构框图	430
19.4.	功能说明	430
19.4.1.	数据格式配置	430
19.4.2.	空间矢量脉宽调制原理	431
19.4.3.	空间矢量脉宽调制实现	432
19.4.4.	五段式 SVPWM	432
19.4.5.	七段式 SVPWM	434
19.4.6.	过调制处理	436

19.5. SVPWM 寄存器	437
19.5.1. 控制寄存器 0 (SVPWM_CTL0)	437
19.5.2. Ualpha 寄存器 (SVPWM_UALPHA)	437
19.5.3. Ubeta 寄存器 (SVPWM_UBETA)	438
19.5.4. 自动重载寄存器 (SVPWM_CAR)	438
19.5.5. Ta 寄存器 (SVPWM_TA)	439
19.5.6. Tb 寄存器 (SVPWM_TB)	439
19.5.7. Tc 寄存器 (SVPWM_TC)	439
19.5.8. 扇区寄存器 (SVPWM_SEC)	440
19.5.9. 状态寄存器 (SVPWM_STAT)	440
20. 看门狗定时器 (WDGT)	442
20.1. 独立看门狗定时器 (FWDGT)	442
20.1.1. 简介	442
20.1.2. 特征	442
20.1.3. 功能说明	442
20.1.4. FWDGT 寄存器	445
20.2. 窗口看门狗定时器 (WWDGT)	449
20.2.1. 简介	449
20.2.2. 主要特征	449
20.2.3. 功能说明	449
20.2.4. WWDGT 寄存器	453
21. 定时器 (TIMER)	456
21.1. 高级定时器 (TIMERx, x = 0, 7)	457
21.1.1. 简介	457
21.1.2. 主要特征	457
21.1.3. 结构框图	457
21.1.4. 功能描述	458
21.1.5. TIMERx 寄存器 (x = 0, 7)	504
21.2. 通用定时器 L0 (TIMERx, x = 1, 2)	559
21.2.1. 简介	559
21.2.2. 主要特征	559
21.2.3. 结构框图	559
21.2.4. 功能描述	560
21.2.5. TIMERx 寄存器 (x = 1, 2)	575
22. 通用定时器 (GPTIMER)	596
22.1. 简介	596
22.2. 主要特征	596
22.3. 结构框图	597
22.4. 功能描述	597

22.4.1.	时钟源配置	597
22.4.2.	时钟预分频器	598
22.4.3.	定时器使能/禁能/复位	599
22.4.4.	周期计数模式	600
22.4.5.	事件计数模式	603
22.4.6.	计数方向	604
22.4.7.	重复计数器	604
22.4.8.	影子寄存器加载功能	606
22.4.9.	捕获/比较通道	608
22.4.10.	PWM 模式	611
22.4.11.	互补输出和死区时间插入	612
22.4.12.	占空比 0%或 100%输出	613
22.4.13.	GPTIMERx 之间的同步	614
22.4.14.	单脉冲模式	617
22.4.15.	中断跳过功能	618
22.4.16.	ADC 触发请求	620
22.4.17.	与 EVIC 的交互	621
22.4.18.	保护功能	622
22.4.19.	输出引脚初始化	622
22.4.20.	定时器 DMA 模式	623
22.4.21.	定时器调试模式	623
22.5.	GPTIMERx 寄存器(x = 0, 1)	624
22.5.1.	寄存器写保护(GPTIMERx_WP)	624
22.5.2.	软件使能寄存器(GPTIMERx_SWEN)	625
22.5.3.	软件禁能寄存器(GPTIMERx_SWDIS)	625
22.5.4.	软件复位寄存器(GPTIMERx_SWRST)	626
22.5.5.	计数器使能源选择寄存器(GPTIMERx_ESSEL)	626
22.5.6.	计数器禁能源选择寄存器(GPTIMERx_DSSEL)	628
22.5.7.	计数器复位源选择寄存器(GPTIMERx_RSSEL)	629
22.5.8.	通道 0 捕获源选择寄存器(GPTIMERx_CH0CSSEL)	631
22.5.9.	通道 1 捕获源选择寄存器(GPTIMERx_CH1CSSEL)	632
22.5.10.	控制寄存器 0(GPTIMERx_CTL0)	634
22.5.11.	向上计数事件源选择寄存器(GPTIMERx_CUPEVSSEL)	636
22.5.12.	向下计数事件源选择寄存器(GPTIMERx_CDNEVSSEL)	637
22.5.13.	通道 0 控制寄存器(GPTIMERx_CH0CTL)	639
22.5.14.	通道 1 控制寄存器(GPTIMERx_CH1CTL)	641
22.5.15.	通道控制寄存器(GPTIMERx_CHCTL)	644
22.5.16.	DMA 和中断使能寄存器(GPTIMERx_DMAINTEN)	646
22.5.17.	中断标志寄存器(GPTIMERx_INTF)	648
22.5.18.	更新源选择寄存器(GPTIMERx_UPSSEL)	650
22.5.19.	计数器寄存器(GPTIMERx_CNT)	651
22.5.20.	预分频寄存器(GPTIMERx_PSC)	652
22.5.21.	计数器自动重载寄存器(GPTIMERx_CAR)	652

22.5.22.	通道 0 捕获/比较寄存器(GPTIMERx_CH0CV).....	652
22.5.23.	通道 1 捕获/比较寄存器(GPTIMERx_CH1CV).....	653
22.5.24.	通道 0 附加比较寄存器(GPTIMERx_CH0COMV_ADD)	653
22.5.25.	通道 1 附加比较寄存器(GPTIMERx_CH1COMV_ADD)	654
22.5.26.	死区控制寄存器(GPTIMERx_DTCTL)	654
22.5.27.	ADC 触发控制寄存器(GPTIMERx_ADCTL).....	655
22.5.28.	ADC 触发比较值 1 寄存器(GPTIMERx_ADCCR1)	657
22.5.29.	ADC 触发比较值 2 寄存器(GPTIMERx_ADCCR2)	658
22.5.30.	ADC 触发跳过寄存器(GPTIMERx_ADCTRGs).....	658
22.5.31.	附加中断跳过控制寄存器 0(GPTIMERx_ADDINTSCTL0).....	659
22.5.32.	附加中断跳过控制寄存器 1(GPTIMERx_ADDINTSCTL1).....	661
22.5.33.	中断和 ADC 触发信号跳过寄存器(GPTIMERx_IADCTSS).....	662
22.5.34.	重复计数器寄存器(GPTIMERx_CREP)	663
22.5.35.	同步复位控制寄存器(GPTIMERx_SYN_RSTCTL)	664
22.5.36.	DMA 配置寄存器(GPTIMERx_DMACFG).....	665
22.5.37.	DMA 发送缓冲区寄存器(GPTIMERx_DMATB)	666
22.5.38.	配置寄存器(GPTIMERx_CFG).....	666
23.	比较匹配定时器 W (CPTIMERW)	668
23.1.	简介.....	668
23.2.	主要特征.....	668
23.3.	功能说明.....	668
23.3.1.	时钟源	669
23.3.2.	时钟预分频器	669
23.3.3.	计数器向上计数.....	670
23.3.4.	计数器位宽配置.....	671
23.3.5.	输入捕获和输出比较	671
23.3.6.	EVIC 互连	673
23.4.	CPTIMERW 寄存器.....	675
23.4.1.	控制寄存器 0 (CPTIMERW_CTL0)	675
23.4.2.	控制寄存器 1 (CPTIMERW_CTL1)	676
23.4.3.	预分频器寄存器 (CPTIMERW_PSC)	677
23.4.4.	计数器寄存器 (CPTIMERW_CNT)	678
23.4.5.	中断使能寄存器 (CPTIMERW_INTEN)	678
23.4.6.	中断标志寄存器 (CPTIMERW_INTF)	679
23.4.7.	软件事件生成寄存器 (CPTIMERW_SWEVG)	680
23.4.8.	输入捕获通道 0 捕获值寄存器 (CPTIMERW_ICH0CV)	681
23.4.9.	输入捕获通道 1 捕获值寄存器 (CPTIMERW_ICH1CV)	681
23.4.10.	输出比较通道 0 比较值寄存器 (CPTIMERW_OCH0CV)	681
23.4.11.	输出比较通道 1 比较值寄存器 (CPTIMERW_OCH1CV)	682
23.4.12.	输出比较通道 2 比较值寄存器 (CPTIMERW_OCH2CV)	682
23.4.13.	输出比较通道 3 比较值寄存器 (CPTIMERW_OCH3CV)	683
23.4.14.	计数器自动重载寄存器 (CPTIMERW_CAR)	683

24. 比较匹配定时器 (CPTIMER)	684
24.1. 简介	684
24.2. 主要特征	684
24.3. 功能说明	684
24.3.1. 时钟源	684
24.3.2. 时钟预分频器	685
24.3.3. 计数器向上计数模式	685
24.3.4. EVIC 互连	686
24.4. CPTIMER 寄存器	688
24.4.1. 控制寄存器 0 (CPTIMER_CTL0)	688
24.4.2. 计数器 0 寄存器 (CPTIMER_CNT0)	688
24.4.3. 计数器 1 寄存器 (CPTIMER_CNT1)	689
24.4.4. 计数器 0 预分频器寄存器 (CPTIMER_CNT0PSC)	689
24.4.5. 计数器 1 预分频器寄存器 (CPTIMER_CNT1PSC)	690
24.4.6. 中断使能寄存器 (CPTIMER_INTEN)	690
24.4.7. 中断标志寄存器 (CPTIMER_INTF)	691
24.4.8. 软件事件生成寄存器 (CPTIMER_SWEVG)	691
24.4.9. 计数器 0 自动重载寄存器 (CPTIMER_CNT0CAR)	692
24.4.10. 计数器 1 自动重载寄存器 (CPTIMER_CNT1CAR)	692
25. 通用异步收发器 (UART)	694
25.1. 简介	694
25.2. 主要特征	694
25.3. 功能说明	695
25.3.1. UART 帧格式	695
25.3.2. 波特率发生	696
25.3.3. UART 发送器	696
25.3.4. UART 接收器	697
25.3.5. DMA 方式访问数据缓冲区	699
25.3.6. 多处理器通信	700
25.3.7. LIN 模式	701
25.3.8. 串行红外 (IrDA SIR) 编解码功能模块	701
25.3.9. 半双工通信模式	702
25.3.10. UART 中断	703
25.4. UART 寄存器	704
25.4.1. 状态寄存器 0 (UART_STAT0)	704
25.4.2. 数据寄存器 (UART_DATA)	706
25.4.3. 波特率寄存器 (UART_BAUD)	706
25.4.4. 控制寄存器 0 (UART_CTL0)	707
25.4.5. 控制寄存器 1 (UART_CTL1)	709
25.4.6. 控制寄存器 2 (UART_CTL2)	710

25.4.7.	保护时间和预分频器寄存器 (UART_GP)	712
25.4.8.	兼容性控制寄存器 (UART_CHC)	712
26.	内部集成电路总线接口 (I2C)	714
26.1.	简介.....	714
26.2.	主要特征.....	714
26.3.	功能说明.....	714
26.3.1.	时钟要求.....	715
26.3.2.	I2C 通讯流程	716
26.3.3.	数字噪声滤波器.....	718
26.3.4.	I2C 时序配置	718
26.3.5.	I2C 复位	720
26.3.6.	数据传输.....	720
26.3.7.	I2C 从机模式	722
26.3.8.	I2C 主机模式	727
26.3.9.	SMBus 支持	732
26.3.10.	SMBus 模式	734
26.3.11.	从省电模式唤醒.....	736
26.3.12.	DMA 模式下数据传输.....	736
26.3.13.	I2C 错误和中断.....	736
26.3.14.	I2C 调试模式	737
26.3.15.	事件链接功能 (输出)	737
26.4.	I2C 寄存器.....	738
26.4.1.	控制寄存器 0 (I2C_CTL0)	738
26.4.2.	控制寄存器 1 (I2C_CTL1)	740
26.4.3.	从机地址寄存器 0 (I2C_SADDR0)	742
26.4.4.	从机地址寄存器 1 (I2C_SADDR1)	743
26.4.5.	时序寄存器 (I2C_TIMING)	744
26.4.6.	超时寄存器 (I2C_TIMEOUT)	744
26.4.7.	状态寄存器 (I2C_STAT)	745
26.4.8.	状态清除寄存器 (I2C_STATC)	748
26.4.9.	PEC 寄存器 (I2C_PEC)	749
26.4.10.	接收数据寄存器 (I2C_RDATA)	749
26.4.11.	发送数据寄存器 (I2C_TDATA)	750
26.4.12.	控制寄存器 2 (I2C_CTL2)	750
27.	串行外设接口 (SPI)	752
27.1.	简介.....	752
27.2.	主要特性.....	752
27.2.1.	SPI 主要特性.....	752
27.3.	SPI 功能说明.....	753
27.3.1.	SPI 结构框图.....	753

27.3.2.	SPI 信号线描述	753
27.3.3.	SPI 时序和数据帧格式	754
27.3.4.	NSS 功能	755
27.3.5.	SPI 运行模式	756
27.3.6.	DMA 功能	764
27.3.7.	CRC 功能	764
27.3.8.	SPI 中断	765
27.4.	SPI 寄存器	767
27.4.1.	控制寄存器 0 (SPI_CTL0)	767
27.4.2.	控制寄存器 1 (SPI_CTL1)	769
27.4.3.	状态寄存器 (SPI_STAT)	770
27.4.4.	数据寄存器 (SPI_DATA)	771
27.4.5.	CRC 多项式寄存器 (SPI_CRCPOLY)	771
27.4.6.	接收 CRC 寄存器 (SPI_RCRC)	772
27.4.7.	发送 CRC 寄存器 (SPI_TCRC)	772
27.4.8.	SPI 四线 SPI 控制寄存器 (SPI_QCTL)	773
28.	控制器局域网络 (CAN)	775
28.1.	简介	775
28.2.	主要特征	775
28.3.	功能说明	775
28.3.1.	工作模式	776
28.3.2.	通信模式	777
28.3.3.	数据发送	778
28.3.4.	数据接收	779
28.3.5.	过滤功能	780
28.3.6.	时间触发通信	784
28.3.7.	通信参数	784
28.3.8.	错误标志	785
28.3.9.	中断	786
28.4.	CAN 寄存器	788
28.4.1.	控制寄存器 (CAN_CTL)	788
28.4.2.	状态寄存器 (CAN_STAT)	789
28.4.3.	发送状态寄存器 (CAN_TSTAT)	791
28.4.4.	接收 FIFO0 寄存器 (CAN_RFIFO0)	794
28.4.5.	接收 FIFO1 寄存器 (CAN_RFIFO1)	794
28.4.6.	中断使能寄存器 (CAN_INTEN)	795
28.4.7.	错误寄存器 (CAN_ERR)	797
28.4.8.	位时序寄存器 (CAN_BT)	797
28.4.9.	发送邮箱标识符寄存器 (CAN_TMIx) (x = 0..2)	798
28.4.10.	发送邮箱属性寄存器 (CAN_TMPx) (x = 0..2)	799
28.4.11.	发送邮箱 data0 寄存器 (CAN_TMDATA0x) (x = 0..2)	800

28.4.12.	发送邮箱 data1 寄存器 (CAN_TMDATA1x) (x = 0...2)	800
28.4.13.	接收 FIFO 邮箱标识符寄存器 (CAN_RFIFOMIx) (x = 0, 1)	801
28.4.14.	接收 FIFO 邮箱属性寄存器 (CAN_RFIFOMPx) (x = 0, 1)	801
28.4.15.	接收 FIFO 邮箱 data0 寄存器 (CAN_RFIFOMDATA0x) (x = 0, 1)	802
28.4.16.	接收 FIFO 邮箱 data1 寄存器 (CAN_RFIFOMDATA1x) (x = 0, 1)	802
28.4.17.	过滤器控制寄存器 (CAN_FCTL)	803
28.4.18.	过滤器模式配置寄存器 (CAN_FMCFG)	803
28.4.19.	过滤器位宽配置寄存器 (CAN_FSCFG)	804
28.4.20.	过滤器关联 FIFO 寄存器 (CAN_FAFIFO)	804
28.4.21.	过滤器激活寄存器 (CAN_FW)	805
28.4.22.	过滤器 (x) 数据 (y) 寄存器 (CAN_FxDATAy) (x = 0...27, y = 0, 1)	805
29.	附录	806
29.1.	寄存器表中使用的缩写列表	806
29.2.	术语表	806
29.3.	可用外设	806
30.	版本历史	807

图索引

图 1-1. Cortex®-M33 处理器的结构.....	31
图 1-2. GD32M531 系统架构示意图	33
图 1-3. ECC 解码器.....	37
图 2-1. 页擦除操作流程.....	71
图 2-2. 整片擦除操作流程	72
图 2-3. 字编程操作流程.....	74
图 2-4. 快速编程操作流程	76
图 3-1. 电源域概览	94
图 3-2. 上电 / 掉电复位波形图	95
图 3-3. 低电压检测器 0 框图	97
图 3-4. 低电压检测器 1 框图	98
图 3-5. 低电压检测器 2 框图	99
图 4-1. 系统复位电路	106
图 4-2. 时钟树	107
图 4-3. HXTAL 时钟源	108
图 4-4. 旁路模式下 HXTAL 时钟源	108
图 5-1. CFMU 框图	136
图 5-2. 时钟频率精度测量	137
图 6-1. EXTI 框图.....	146
图 7-1. GPIO 端口位的基本结构	154
图 7-2. 输入配置的基本结构.....	156
图 7-3. 输出配置的基本结构.....	157
图 7-4. 模拟配置的基本结构.....	157
图 7-5. 备用功能配置的基本结构	158
图 8-1. POC 模块框图.....	175
图 8-2. TIMER 和 GPTIMER 通道引脚输出禁能.....	175
图 8-3. POC 引脚下降沿检测.....	177
图 8-4. POC 引脚低电平检测 (INDSNUM[3:0]=4'b0000)	177
图 8-5. 互补 PWM 通道输出检测.....	178
图 8-6. POC 引脚下降沿输入检测屏蔽	181
图 9-1. GTOCx (x=0...3) 模块框图	215
图 9-2. 具有数字滤波的 GTOCx_IN 引脚输入检测 (DFSNUM[1:0] = 2'b00, INPL=1, INDM=0)	216
图 10-1. CRC 计算单元框图.....	224
图 11-1. TMU 模块结构框图.....	229
图 12-1. DMA 结构框图	244
图 12-2. 握手机制.....	246
图 12-3. DMA 中断逻辑图.....	248
图 13-1. DMAMUX 结构框图	256
图 13-2. 同步模式.....	258
图 13-3. 通道事件输出	259
图 15-1. ADC0 模块框图	278

图 15-2. ADC2 模块框图	279
图 15-3. ADC0 的单次扫描基本运行（无通道专用采样保持电路）	281
图 15-4. ADC0 的单次扫描基本运行（使用通道专用采样保持电路，禁用恒采样模式）	282
图 15-5. ADC0 的单次扫描基本运行（使用通道专用采样保持电路，使能恒采样模式）	283
图 15-6. ADC2 单次扫描模式中的通道选择和自诊断（无专用的采样保持电路）	284
图 15-7. ADC0 单次扫描模式中的通道选择和自诊断（使用通道专用采样保持电路，禁用恒采样模式）	285
图 15-8. ADC0 单次扫描模式中的通道选择和自诊断（使用通道专用采样保持电路，使能恒采样模式）	286
图 15-9. ADC0 Group_pri1 在分叉触发模式下的 A/D 转换	287
图 15-10. ADC0 的连续扫描基本运行（无通道专用采样保持电路）	288
图 15-11. ADC0 的连续扫描基本运行（使用通道专用采样保持电路，禁用恒采样模式）	289
图 15-12. ADC0 的连续扫描基本运行（使用通道专用采样保持电路，使能恒采样模式）	290
图 15-13. ADC2 单次扫描模式中的通道选择和自诊断（无专用的采样保持电路）	291
图 15-14. ADC0 单次扫描模式中的通道选择和自诊断（使用通道专用采样保持电路，禁用恒采样模式）	292
图 15-15. ADC0 连续扫描模式中的通道选择和自诊断（使用通道专用采样保持电路，使能恒采样模式）	293
图 15-16. 多组扫描模式基本运行	295
图 15-17. 在分叉触发模式下的基本 A/D 转换（无专用采保电路）	296
图 15-18. 在分叉触发模式下的基本 A/D 转换（使用专用采保电路）	296
图 15-19. 在分叉触发模式下的基本 A/D 转换和组重启操作（无专用采保电路）	297
图 15-20. 在分叉触发模式下的基本 A/D 转换和组重启操作（无专用采保电路）	297
图 15-21. 在分叉触发模式下的组重启操作和组优先级控制（无专用采保电路）	298
图 15-22. 在分叉触发模式下的组重启操作和组优先级控制（使用专用采保电路）	298
图 15-23. 在 Group_pri2 扫描期间，Group_pri1 触发输入（重启扫描设置为 GP2RTCH = 1）	304
图 15-24. 在 Group_pri2 扫描期间，Group_pri1 触发输入（重启扫描设置为 GP2RTCH = 0）	305
图 15-25. 在 Group_pri2 重启扫描期间，Group_pri1 触发输入（重启扫描设置为 GP2RTCH = 1）	305
图 15-26. 在 Group_pri2 重启扫描期间，Group_pri1 触发输入（重启扫描设置为 GP2RTCH = 0）	306
图 15-27. 在 Group_pri1 扫描期间，Group_pri2 触发输入（重启扫描设置为 GP2RTCH = 1）	307
图 15-28. 在 Group_pri2 扫描期间，Group_pri1 触发输入（禁能重启扫描 GP2RTEN = 1）	308
图 15-29. Group_pri2 连续的单扫描运行(LPGCNT = 1, GP2RTCH = 1)	309
图 15-30. 交替自诊断（SDSFIX = 0），重启扫描设置为 GP2RTCH = 0	309
图 15-31. 固定自诊断（SDSFIX = 0），重启扫描设置为 GP2RTCH = 0	309
图 15-32. 在低优先级组扫描期间，高优先级组触发输入（GP2RTCH = 1, GP3RTCH = 1）	311
图 15-33. 在低优先级组扫描期间，高优先级组触发输入（GP2RTCH = 0, GP3RTCH = 0）	311
图 15-34. 在低优先级组扫描期间，高优先级组触发输入（GP2RTCH = 1, GP3RTCH = 1）	312
图 15-35. 在低优先级组扫描期间，高优先级组触发输入（GPxRTEN = 1）	313
图 15-36. 在低优先级组扫描期间，高优先级组触发输入（GPxRTEN = 0）	314
图 15-37. 在 Group_pri3 上进行连续的单次扫描运行(GP2RTCH = 1, GP3RTCH = 1)	315
图 15-38. ADC2 的 Group_pri1 组实际 A/D 转换顺序	316
图 15-39. ADC2 的 Group_pri2 组实际 A/D 转换顺序	316
图 15-40. 12 位数据存储模式	317
图 15-41. 10 位数据存储模式	317
图 15-42. 8 位数据存储模式	318

图 15-43. 6 位数据存储模式.....	318
图 15-44. 自诊断 12 位数据存储模式.....	318
图 15-45. 自诊断 10 位数据存储模式.....	318
图 15-46. 自诊断 8 位数据存储模式.....	319
图 15-47. 自诊断 6 位数据存储模式.....	319
图 15-48. ADC2 Group_pri1 DMA 传输例子.....	324
图 15-49. ADC2 Group_pri1 DMA 传输例子（使能分叉触发功能）.....	324
图 15-50. ADC2 Group_pri2 溢出事件举例.....	325
图 15-51. 当辅助断开检测功能使用时，AD 转换的运行.....	327
图 15-52. 低优先级组重新扫描期间高优先级组触发输入.....	329
图 16-1. DAC 结构框图.....	369
图 16-2. DAC LFSR 算法.....	371
图 16-3. DAC 三角噪声模式生成的波形.....	372
图 16-4. D/A 和 A/D 防干扰.....	372
图 17-1. 比较器框图.....	383
图 17-2. 比较器迟滞.....	385
图 17-3. 比较器滤波输出.....	386
图 18-1. EVIC 框图.....	398
图 18-2. 单个 I/O 引脚输出举例.....	402
图 18-3. 组 E 的 I/O 引脚输入加载举例.....	403
图 18-4. 组 I/O 引脚输出举例.....	404
图 18-5. 组 I/O 引脚输出循环模式.....	404
图 19-1. SVPWM 模块框图.....	430
图 19-2. SVPWM 输出基本电压矢量.....	431
图 19-3. 平均矢量调制.....	431
图 19-4. 五段式 SVPWM 输出波形图.....	433
图 19-5. 五段式 SVPWM 算法.....	434
图 19-6. 七段式 SVPWM 输出波形图.....	435
图 19-7. 七段式 SVPWM 算法.....	436
图 20-1. 独立看门狗定时器框图.....	443
图 20-2. 窗口看门狗定时器框图.....	450
图 20-3. 窗口看门狗定时器时序图.....	451
图 21-1. 高级定时器结构框图.....	458
图 21-2. 内部时钟分频为 1 时正常模式下的控制电路.....	459
图 21-3. 当预分频器的参数从 1 变到 2 时，计数器的时序图.....	460
图 21-4. 向上计数时序图，PSC = 0/2.....	461
图 21-5. 向上计数时序图，在运行时改变 TIMEx_CAR 寄存器的值（ARSE=0，CAR 值立即更新， ARSE=1，CARUPSEL =2'b00，CAR 值会在下一个更新事件来临时更新）.....	461
图 21-6. 向下计数时序图，PSC = 0/2.....	462
图 21-7. 向下计数时序图，在运行时改变 TIMEx_CAR 寄存器值（ARSE=0，CAR 值立即更新， ARSE=1，CARUPSEL =2'b00，CAR 值会在下一个更新事件来临时更新）.....	462
图 21-8. 中央计数模式计数器时序图.....	464
图 21-9. 中央计数模式下计数器重复时序图.....	465

图 21-10. 在向上计数模式下计数器重复时序图	466
图 21-11. 在向下计数模式下计数器重复时序图	466
图 21-12. 中央计数模式下中断重复时序图	467
图 21-13. 在向上计数模式下中断重复时序图.....	467
图 21-14. 在向下计数模式下中断重复时序图.....	468
图 21-15. 更新事件时全局更新模式下的更新时序图	468
图 21-16. 通道 0 输入捕获逻辑	469
图 21-17. 多模式通道 0 输入捕获逻辑	469
图 21-18. 输出比较逻辑 (当 MCHxMSEL = 2'b00 时, x = 0,1,2,3)	471
图 21-19. 输出比较逻辑 (当 MCHxMSEL = 2'b11 时, x = 0,1,2,3)	471
图 21-20. 三种输出比较模式	473
图 21-21. EAPWM 时序图	474
图 21-22. CAPWM 时序图	474
图 21-23 通道 x 输出 PWM (CHxVAL < CHxCOMVAL_ADD)	476
图 21-24 通道 x 输出 PWM (CHxVAL = CHxCOMVAL_ADD)	476
图 21-25. 通道 x 输出 PWM (CHxVAL > CHxCOMVAL_ADD)	477
图 21-26. 通道 x 输出 PWM (CHxVAL 或 CHxCOMVAL_ADD > CARL)	477
图 21-27. 通道 x 输出 PWM 占空比随着 CHxCOMVAL_ADD 值而改变	478
图 21-28. 复合 PWM 模式下四通道输出	478
图 21-29 通道 x 输出 PWM (CHxVAL < CHxCOMVAL_ADD / CHxCOMVAL_ADD < CHxVAL)	480
图 21-30. 通道 x 输出 PWM (CHxVAL = CHxCOMVAL_ADD)	481
图 21-31. 通道 x 输出 PWM (CHxVAL 或 CHxCOMVAL_ADD > CARL)	481
图 21-32. 边沿对齐模式下 CHx_O 输出脉冲 (CHxOMPSEL#2'b00)	482
图 21-33. 中央对齐模式下 CHx_O 输出脉冲 (CHxOMPSEL#2'b00)	483
图 21-34. 带死区时间的互补输出	486
图 21-35. BREAK0 中止功能逻辑图.....	487
图 21-36. 通道响应 BREAK0 中止输入 (高电平有效) 时, 输出信号的行为 (IOS=1)	487
图 21-37. 译码器接口模式下计数器运行例子.....	488
图 21-38. CIOFE0 极性反相的译码器接口模式下的例子.....	489
图 21-39. 非正交译码器模式 0/1 计数器运行实例 (CH1P=0)	490
图 21-40. 非正交译码器模式 2 / 3 计数器运行实例 (CH0P / CH1P=0)	490
图 21-41. 霍尔传感器用在 BLDC 电机控制中.....	492
图 21-42. 两个定时器之间的霍尔传感器时序图	493
图 21-43. 复位模式.....	494
图 21-44. 暂停模式下的控制电路	494
图 21-45. 事件模式.....	495
图 21-46. 单脉冲模式, TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60	495
图 21-47. 用定时器 2 的使能信号触发定时器 0.....	496
图 21-48. 用定时器 2 的更新事件来触发定时器 0	497
图 21-49. 用定时器 2 的使能来选通定时器 0	497
图 21-50. 用定时器 2 的 O0CPRE 信号选通定时器 1	498
图 21-51. 用定时器 2 的 CIO 输入来触发定时器 0 和定时器 2	499
图 21-52. ADC 触发信号产生在 ADC 不同比较匹配触发模式 (ADT1UPGTEN=1, ADT2UPGTEN=1, ADT1DWGTEN=1, ADT2DWGTEN=0)	499

图 21-53. ADC 触发模式下的 ADC 重复计数功能(ADT1UPGTEN =1, ADT2UPGTEN =1, ADT1DWGTEN =1, ADT2DWGTEN =1)	501
图 21-54. ADC 触发器同步监控器信号时序图(ADCSM1SEL = 3'b010)	502
图 21-55. 通用定时器 L0 结构框图	560
图 21-56. 内部时钟分频为 1 时正常模式下的控制电路	561
图 21-57. 当预分频器的参数从 1 变到 2 时, 计数器的时序图	562
图 21-58. 向上计数时序图, PSC=0 / 2	563
图 21-59. 向上计数时序图, 在运行时改变 TIMERx_CAR 寄存器的值	563
图 21-60. 向下计数时序图, PSC=0 / 2	564
图 21-61. 向下计数时序图, 在运行时改变 TIMERx_CAR 寄存器值	565
图 21-62. 中央计数模式计数器时序图	566
图 21-63. 输入捕获逻辑	567
图 21-64. 输出比较逻辑 (x = 0, 1, 2, 3)	568
图 21-65. 三种输出比较模式	569
图 21-66. EAPWM 时序图	570
图 21-67. CAPWM 时序图	570
图 21-68. 复位模式	572
图 21-69. 暂停模式	572
图 21-70. 事件模式	572
图 21-71. 单脉冲模式, TIMERx_CHxCV = 0x04, TIMERx_CAR = 0x60	573
图 21-72. GPTIMER 结构框图	597
图 21-73. 内部时钟分频为 1 时正常模式下的控制电路	598
图 21-74. 当预分频器的参数从 1 变到 2 时, 计数器的时序图	598
图 21-75. 向上计数时序图, PSC = 0/2	600
图 21-76. 向上计数时序图, 在运行时改变 GPTIMERx_CAR 寄存器的值	601
图 21-77. 向下计数时序图, PSC = 0/2	602
图 21-78. 向下计数时序图, 在运行时改变 GPTIMERx_CAR 寄存器值	602
图 21-79. 中央计数模式计数器时序图	603
图 21-80. 中央计数模式下计数器重复时序图	605
图 21-81. 在向上计数模式下计数器重复时序图	605
图 21-82. 在向下计数模式下计数器重复时序图	606
图 21-83. 非全局一次性重载模式下更新事件的更新时序图	607
图 21-84. 全局一次性重载模式下更新事件的更新时序图 (GLBUPSEL[1:0] ≠ 2'b11)	607
图 21-85. 全局一次性重载模式下更新事件的更新时序图 (GLBUPSEL[1:0] = 2'b11)	608
图 21-86. 通道 0 输入捕获逻辑	609
图 21-87. 输出比较逻辑	610
图 21-88. 三种输出比较模式	611
图 21-89. 复合 PWM 时序图	612
图 21-90. 非对称 PWM 时序图	612
图 21-91. 非对称死区时间	613
图 21-92. 软件同步操作无初始值	614
图 21-93. 软件同步操作有初始值	615
图 21-94. 硬件同步操作源	616

图 21-95. 通过通道同步复位源.....	617
图 21-96. 单脉冲模式, GPTIMERx_CHxCV=0x04, GPTIMERx_CAR=0x60, CH0COMUP[1:0] = 2'b10	617
图 21-97. 中心对齐模式下中断跳过的示例 (跳过值为 2)	619
图 21-98. 边沿对齐模式下中断跳过的示例 (改变方向和跳过值为 2)	619
图 21-99. ADC 触发信号产生在 ADC 不同比较匹配触发模式 (ADT1UPGTEN=1, ADT2UPGTEN=1, ADT1DWGTEN=1, ADT2DWGTEN=0)	620
图 21-100. ADC 触发器同步监控器信号时序图(ADCSM3SEL = 0)	621
图 21-101. 复位后的通道输出	623
图 23-1. 比较匹配定时器 W 框图	669
图 23-2. PSC 值从 0 变化到 2 的时序图, CAR=99.....	670
图 23-3. 向上计数时序图, PSC=0/2.....	671
图 23-4. 通道输入捕获原理	672
图 23-5. 两种模式下的输出比较.....	673
图 24-1. 比较匹配定时器框图.....	684
图 24-2. CNTxPSC 值从 0 变化到 2 的时序图, CNTxCAR=99.....	685
图 24-3. 向上计数时序图, CNTxPSC=0/2	686
图 25-1. UART 模块内部框图	695
图 25-2. UART 字符帧 (8 数据位和 1 停止位)	695
图 25-3. UART 发送步骤.....	697
图 25-4. 过采样方式接收一个数据位 (OSB = 0)	698
图 25-5. 采用 DMA 方式实现 UART 数据发送配置步骤	699
图 25-6. 采用 DMA 方式实现 UART 数据接收配置步骤	700
图 25-7. 空闲状态下检测断开帧.....	701
图 25-8. 数据传输过程中检测断开帧.....	701
图 25-9. IrDA SIR ENDEC 模块.....	702
图 25-10. IrDA 数据调制	702
图 25-11. UART 中断映射框图.....	703
图 26-1. I2C 模块框图	715
图 26-2. 数据有效性	716
图 26-3. 开始和停止信号	717
图 26-4. 10 位地址的 I2C 通讯流程 (主机发送)	717
图 26-5. 7 位地址的 I2C 通讯流程 (主机发送)	717
图 26-6. 7 位地址的 I2C 通讯流程 (主机接收)	718
图 26-7. 10 位地址的 I2C 通讯流程 (主机接收, HEAD10R=0)	718
图 26-8. 10 位地址的 I2C 通讯流程 (主机接收, HEAD10R=1)	718
图 26-9. 数据保持时间	719
图 26-10. 数据建立时间	719
图 26-11. 数据发送.....	721
图 26-12. 数据接收.....	721
图 26-13. I2C 从机初始化	724
图 26-14. I2C 从机发送编程模型 (SS=0)	725
图 26-15. I2C 从机发送编程模型 (SS=1)	726

图 26-16. I2C 从机接收编程模型	727
图 26-17. I2C 主机初始化	728
图 26-18. I2C 主机发送编程模型 ($N \leq 255$)	729
图 26-19. I2C 主机发送编程模型 ($N > 255$)	730
图 26-20. I2C 主机接收编程模型 ($N \leq 255$)	731
图 26-21. I2C 主机接收编程模型 ($N > 255$)	732
图 26-22. SMBus 主机发送器和从机接收器通信流程	735
图 26-23. SMBus 主机接收器和从机发送器通信流程	736
图 26-24. SPI 结构框图	753
图 26-25. 常规模式下的 SPI 时序图	754
图 26-26. SPI 四线模式下的 SPI 时序图 ($CKPL=1, CKPH=1, LF=0$)	755
图 26-27. 典型的全双工模式连接	757
图 26-28. 典型的单工模式连接 (主机: 接收, 从机: 发送)	758
图 26-29. 典型的单工模式连接 (主机: 只发送, 从机: 接收)	758
图 26-30. 典型的双向线连接	758
图 26-31. 主机 TI 模式在不连续发送时的时序图	760
图 26-32. 主机 TI 模式在连续发送时的时序图	760
图 26-33. 从机 TI 模式时序图	760
图 26-34. NSS 脉冲模式时序图 (主机连续发送)	761
图 26-35. SPI 四线模式四线写操作时序图	762
图 26-36. SPI 四线模式四线读操作时序图	763
图 28-1. CAN 模块结构框图	776
图 28-2. 发送寄存器	778
图 28-3. 发送邮箱状态转换	778
图 28-4. 接收寄存器	780
图 28-5. 32-bit 位宽过滤器	781
图 28-6. 16-bit 位宽过滤器	781
图 28-7. 32-bit 位宽掩码模式过滤器	781
图 28-8. 16-bit 位宽掩码模式过滤器	781
图 28-9. 32-bit 位宽列表模式过滤器	782
图 28-10. 16-bit 位宽列表模式过滤器	782
图 28-11. 位时序	785

表索引

表 1-1. 总线互连矩阵	31
表 1-2. GD32M531 系列器件的存储器映射表.....	34
表 1-3. ECC 中断使能和标志	37
表 1-4. 启动模式.....	38
表 2-1. GD32M531 闪存基地址和构成.....	66
表 2-2. WSCNT 与 AHB 时钟频率对应关系	68
表 2-3. 选项字节.....	79
表 2-4. OB_WP 位对应页保护	82
表 3-1. 节电模式总结	96
表 4-1. 时钟源的选择	110
表 5-1. CFMU 中断请求	137
表 6-1. Cortex®-M33 中的 NVIC 异常类型	142
表 6-2. 中断向量表	143
表 6-3. EXTI 触发源	147
表 7-1. GPIO 配置表	154
表 8-1. 输出禁能的目标定时器和通道	176
表 8-2. 输出检测的互补 PWM 通道.....	178
表 8-3. 禁能请求的目标定时器选择.....	179
表 9-1. 输出关闭控制的 GPTIMER 通道	215
表 11-1. 输入数据配置.....	230
表 11-2. 输出数据配置.....	231
表 11-3. IEEE754 32 位单精度浮点格式	231
表 11-4. TMU 模式配置	232
表 11-5. 模式 0 描述, IFLTEN 位和 OFLTEN 位均置位时	232
表 11-6. 模式 0 描述, IFLTEN 位或 OFLTEN 位复位时	232
表 11-7. 模式 1 描述, IFLTEN 位和 OFLTEN 位均置位时	233
表 11-8. 模式 1 描述, IFLTEN 位或 OFLTEN 位复位时	233
表 11-9. 模式 2 描述, IFLTEN 位和 OFLTEN 位均置位时	234
表 11-10. 模式 2 描述, IFLTEN 位或 OFLTEN 位复位时	235
表 11-11. 模式 3 描述, IFLTEN 位和 OFLTEN 位均置位时	236
表 11-12. 模式 3 描述, IFLTEN 位或 OFLTEN 位复位时	236
表 11-13. 模式 4 描述, IFLTEN 位和 OFLTEN 位均置位时	237
表 11-14. 模式 4 描述, IFLTEN 位或 OFLTEN 位复位时	237
表 11-15. 模式 4 推荐的缩放因子	237
表 11-16. TMU 指令集.....	238
表 12-1. DMA 传输操作	244
表 12-2. 中断事件.....	247
表 13-1. DMAMUX 信号.....	257
表 13-2. 中断事件.....	260
表 13-3. DMAMUX 请求路由输入信号映射	261
表 13-4. 触发输入信号映射	263

表 13-5. 同步输入信号映射	264
表 14-1. 引脚分配.....	271
表 15-1. ADC 内部输入信号.....	277
表 15-2. ADC 输入引脚定义.....	277
表 15-3. 根据 GPxRTEN 位的设置对组的扫描操作进行控制 (GPRIEN = 1, GP3EN = 1, GP4EN = 1)	299
表 15-4. Group_pri1/pri2 组优先级运行设置和运行模式 (GPRIEN = 1, GP3EN = 0, GP4EN = 0).....	301
表 15-5. Group_pri1/pri2/pri3 组优先级运行设置和运行模式 (GPRIEN = 1, GP3EN = 1, GP4EN = 0). 301	
表 15-6. Group_pri1/pri2/pri3/pri4 组优先级运行设置和运行模式 (GPRIEN = 1, GP3EN = 1, GP4EN = 1)	302
表 15-7. ADC2 通道优先级设置举例.....	316
表 15-8. ADC 同步/异步触发使能	322
表 15-9. ADC 触发边沿选择.....	322
表 15-10. ADC 同步触发源	323
表 15-11. ADC 异步触发源.....	323
表 15-12. 不同分辨率对应的 t _{CONV} 时间	326
表 16-1. DAC 引脚.....	369
表 16-2. DAC 触发与输出	369
表 16-3. DAC 外部触发.....	370
表 16-4. DAC 转换模式.....	370
表 17-1. CMP 的输入和输出总结	384
表 18-1. 事件源位域选择.....	399
表 19-1. 不同扇区与作用时间对应关系	432
表 20-1. 独立看门狗定时器在 32 kHz (IRC32K) 时的最小/最大超时周期.....	444
表 20-2. 在 90MHz (f _{PCLK1})时的最大/最小超时值	452
表 21-1. 定时器 (TIMERx) 分为 2 种类型.....	456
表 21-2. 高级定时器通道介绍.....	458
表 21-3. 复合 PWM 脉冲宽度.....	475
表 21-4. 非对称 PWM 脉冲宽度	479
表 21-5. 由参数控制的互补输出表 (MCHxMSEL = 2'b11)	484
表 21-6. 不同译码器模式下的计数方向	488
表 21-7. 非正交译码器模式 1 的计数情况	489
表 21-8. 非正交译码器模式 2 / 3 下的计数方向	491
表 21-9. 从模式例子列表.....	493
表 21-10. TIMER0/TIMER7 ADC 触发器信号及其操作	500
表 21-11. 从模式列表和举例 (通用定时器 L0)	571
表 21-12. 波形输出示意表.....	614
表 21-13. GPTIMER0/1 ADC 触发器信号及其操作	621
表 25-1. UART 重要引脚描述	695
表 25-2. 停止位配置	696
表 25-3. UART 中断请求.....	703
表 26-1. I2C 总线术语说明 (参考飞利浦 I2C 规范)	715
表 26-2. 数据建立时间和数据保持时间	720

表 26-3. 可关闭通信模式.....	722
表 26-4. I2C 错误标志	736
表 26-5. I2C 中断事件	737
表 26-6. SPI 信号描述.....	753
表 26-7. SPI 四线信号描述.....	754
表 26-8. 从机模式 NSS 功能	755
表 26-9. 主机模式 NSS 功能	756
表 26-10. SPI 运行模式.....	756
表 26-11. SPI 中断请求.....	766
表 28-1. 32-bit 过滤序号.....	782
表 28-2. 过滤索引.....	783
表 28-3. CAN 事件/中断标志	787
表 29-1. 寄存器功能位访问属性.....	806
表 29-2. 术语	806
表 30-1. 版本历史.....	807

1. 系统和内存架构

GD32M531 系列的器件是基于 Arm® Cortex®-M33 处理器的 32 位通用微控制器。Arm® Cortex®-M33 处理器包括两条 AHB 总线，称为代码总线和系统总线。Arm® Cortex®-M33 处理器的所有内存访问都是根据不同的目的和目标内存空间在这两条总线上执行的。内存组织采用哈佛架构，预定义的内存映射和高达 4GB 的内存空间，使系统灵活且可扩展。

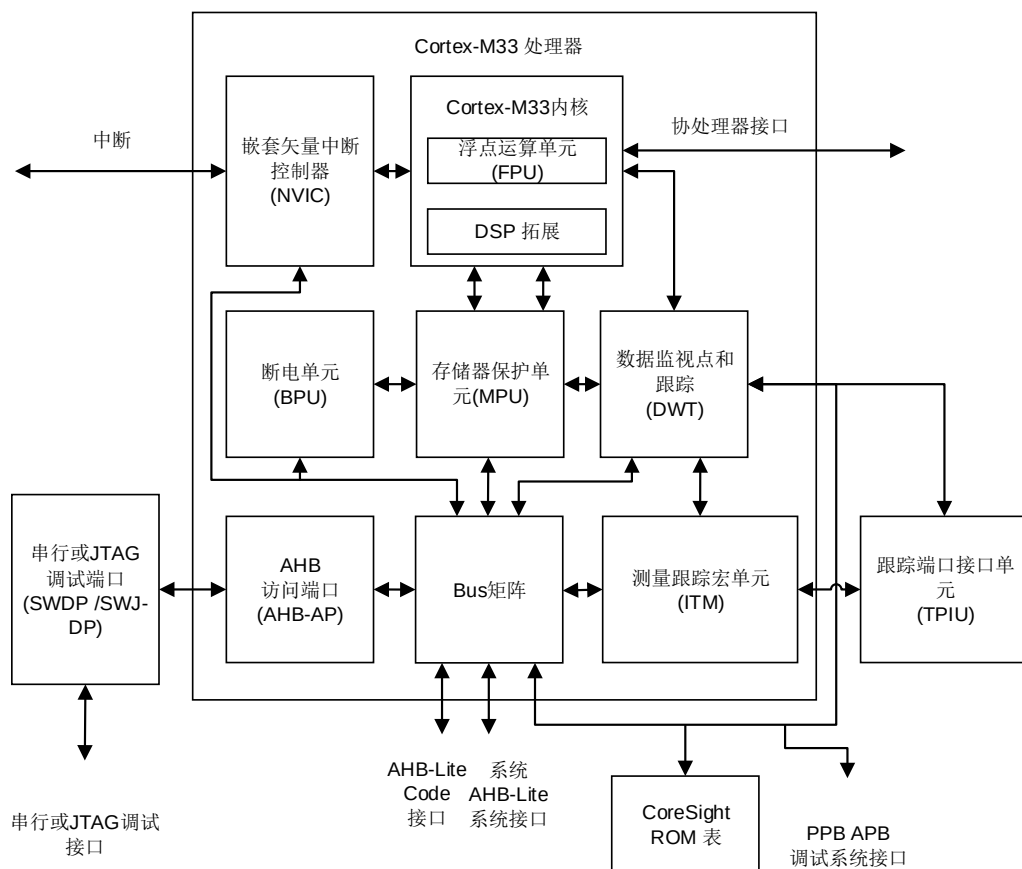
1.1. ARM Cortex-M33 处理器

Cortex®-M33 处理器是一款 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 处理器非常适合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，支持强大且可扩展的指令集，包括通用数据处理、I/O 控制任务、高级数据处理位字段操作和 DSP。Cortex®-M33 还提供了一些系统外设：

- 内部总线矩阵连接了代码总线、系统总线和专用外设总线（PPB）以及调试访问。
- 嵌套矢量中断控制器。
- 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）。
- 浮点单元（FPU）。
- 数字信号处理扩展（DSP）。
- 针对 TMU 的协处理器接口。

[图 1-1. Cortex®-M33 处理器的结构](#)展示了 Cortex®-M33 处理器的块图。更多信息，请参考 Arm® Cortex®-M33 技术参考手册。

图 1-1. Cortex®-M33 处理器的结构



1.2. 系统架构

在 GD32M531 设备中实现了一个 32 位的多层总线，它允许系统内多个主设备和从设备之间的并行访问路径。多层总线由一个 AHB 互连矩阵、两个 AHB 总线和两个 APB 总线组成。AHB 互连矩阵的互连关系如下表所示。在下面的表中，“1”表示相应的主设备能够通过 AHB 互连矩阵访问相应的从设备，而空白则表示相应的主设备不能通过 AHB 互连矩阵访问相应的从设备。

这种架构在[表 1-1. 总线互连矩阵](#)。

表 1-1. 总线互连矩阵

	CBUS	SBUS	DMA0	DMA1
FMC	1	0	1	1
SRAM	1	1	1	1
AHB1	0	1	1	1
AHB2	0	1	1	1
APB1	0	1	1	1
APB2	0	1	1	1

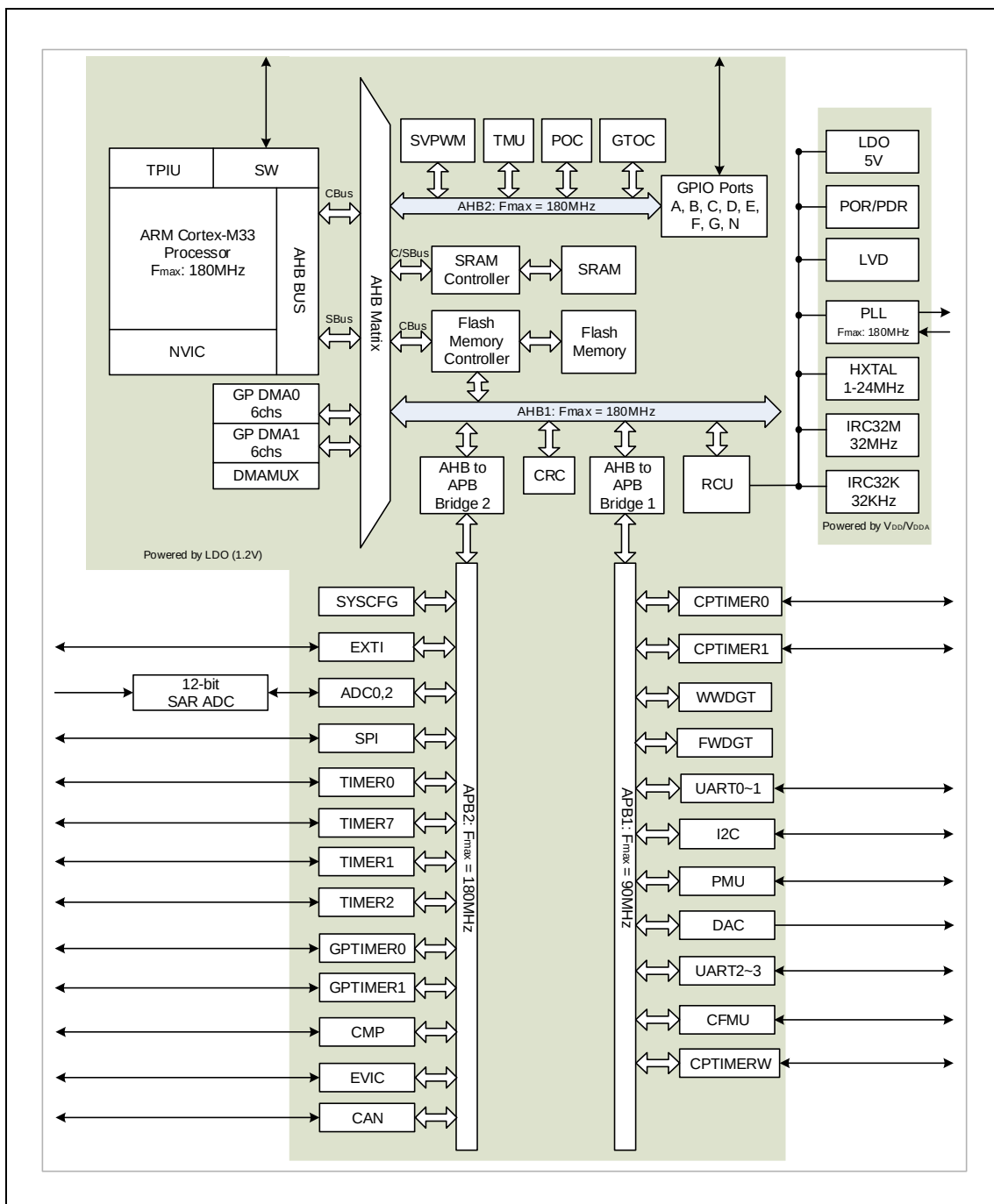
如上所示，有几个主设备连接到 AHB 互连矩阵，包括 CBUS、SBUS、DMA0 和 DMA1。CBUS 是 Cortex®-M33 核心的代码总线，用于任何指令获取和对代码区域的数据访问。类似地，SBUS 是 Cortex®-M33 核心的系统总线，用于系统区域的指令/向量获取、数据加载/存储和调试访问。系统区域包括内部 SRAM 区域和外设区域。DMA0 和 DMA1 分别是 DMA0 和 DMA1 的总线。

AHB 互连矩阵连接了多个从设备，包括 FMC、SRAM、AHB1、AHB2、APB1 和 APB2。FMC 是闪存控制器的总线接口。SRAM 是片上静态随机存取存储器。AHB1 是与除 GPIO 之外的所有 AHB 从设备相连的 AHB 总线。AHB2 是与 GPIO 相连的 AHB 总线。而 APB1 和 APB2 是与所有 APB 从设备相连的两条 APB 总线。

两个 APB 总线连接所有 APB 外设。APB1 最高可达 90MHz，APB2 以全速运行（根据设备不同，最高可达 180MHz）。

这些通过 [图 1-2. GD32M531 系统架构](#) 中所所示的多层 AHB 总线架构相互连接。

图 1-2. GD32M531 系统架构示意图



1.3. 内存映射

Arm® Cortex®-M33 处理器采用哈佛架构，可以使用不同的总线来获取指令和加载/存储数据。程序存储器、数据存储器、寄存器和 I/O 端口都组织在同一个线性的 4G 字节地址空间内，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 处理器提供了一个预定义的内存映射，以减少不同器件供应商重复实现的软件复杂性。在映射中，一些区域由 Arm® Cortex®-M33 系统外设使用，不能修改。然而，其他区域可供供应商使用。[表 1-2. GD32M531 系列器件的存储器映射表](#)显示了 GD32M531 器件的内存映射，包括代码、SRAM、

外设和其他预定义区域。几乎每个外设都分配了 1KB 的空间。这允许简化每个外设的地址解码。

表 1-2. GD32M531 系列器件的存储器映射表

预先定义的地址空间	总线	地址范围	外设
外部 RAM		0x6000 0000 - 0xDFFF FFFF	保留
外设		0x5000 0000 - 0x5FFF FFFF	保留
	AHB2	0x4802 4800 - 0x4FFF FFFF	保留
		0x4802 4400 - 0x4802 47FF	TMU
		0x4802 4000 - 0x4802 43FF	SVPWM
		0x4800 4C00 - 0x4802 3FFF	保留
		0x4800 4800 - 0x4800 4BFF	GTOC
		0x4800 4400 - 0x4800 47FF	POC
		0x4800 4000 - 0x4801 43FF	GPION
		0x4800 1C00 - 0x4800 3FFF	保留
		0x4800 1800 - 0x4800 1BFF	GPIOG
		0x4800 1400 - 0x4800 17FF	GPIOF
		0x4800 1000 - 0x4800 13FF	GPIOE
		0x4800 0C00 - 0x4800 0FFF	GPIOD
		0x4800 0800 - 0x4800 0BFF	GPIOC
		0x4800 0400 - 0x4800 07FF	GPIOB
		0x4800 0000 - 0x4800 03FF	GPIOA
	AHB1	0x4002 3400 - 0x47FF FFFF	保留
		0x4002 3000 - 0x4002 33FF	CRC
		0x4002 2400 - 0x4002 2FFF	保留
		0x4002 2000 - 0x4002 23FF	FMC
		0x4002 1400 - 0x4002 1FFF	保留
		0x4002 1000 - 0x4002 13FF	RCU
		0x4002 0C00 - 0x4002 0FFF	保留
		0x4002 0800 - 0x4002 0BFF	DMAMUX
		0x4002 0400 - 0x4002 07FF	DMA1
		0x4002 0000 - 0x4002 03FF	DMA0
	APB2	0x4001 B000 - 0x4001 FFFF	保留
		0x4001 A000 - 0x4001 AFFF	CAN
		0x4001 8800 - 0x4001 9FFF	保留
		0x4001 8400 - 0x4001 87FF	EVIC
		0x4001 8000 - 0x4001 83FF	保留
		0x4001 7C00 - 0x4001 7FFF	CMP
		0x4001 6800 - 0x4001 7BFF	保留
		0x4001 6100 - 0x4001 67FF	GPTIMER1
		0x4001 6000 - 0x4001 60FF	GPTIMER0
		0x4001 4800 - 0x4001 5FFF	保留

预先定义地址空间	总线	地址范围	外设
		0x4001 4400 - 0x4001 47FF	TIMER2
		0x4001 4000 - 0x4001 43FF	TIMER1
		0x4001 3800 - 0x4001 7BFF	保留
		0x4001 3400 - 0x4001 37FF	TIMER7
		0x4001 3000 - 0x4001 33FF	SPI
		0x4001 2C00 - 0x4001 2FFF	TIMER0
		0x4001 2800 - 0x4001 2BFF	保留
		0x4001 2400 - 0x4001 27FF	ADC2
		0x4001 2000 - 0x4001 23FF	ADC0
		0x4001 0800 - 0x4001 1FFF	保留
		0x4001 0400 - 0x4001 07FF	EXTI
		0x4001 0000 - 0x4001 03FF	SYSCFG
	APB1	0x4000 E400 - 0x4000 FFFF	保留
		0x4000 E000 - 0x4000 E3FF	CPTIMERW
		0x4000 CC00 - 0x4000 DFFF	保留
		0x4000 C800 - 0x4000 CBFF	CFMU
		0x4000 8000 - 0x4000 C7FF	保留
		0x4000 7C00 - 0x4000 7FFF	UART3
		0x4000 7800 - 0x4000 7BFF	UART2
		0x4000 7400 - 0x4000 77FF	DAC
		0x4000 7000 - 0x4000 73FF	PMU
		0x4000 5800 - 0x4000 6FFF	保留
		0x4000 5400 - 0x4000 57FF	I2C
		0x4000 5000 - 0x4000 53FF	UART1
		0x4000 4C00 - 0x4000 4FFF	UART0
		0x4000 3400 - 0x4000 4BFF	保留
		0x4000 3000 - 0x4000 33FF	FWDGT
		0x4000 2C00 - 0x4000 2FFF	WWDGT
		0x4000 0800 - 0x4000 2BFF	保留
		0x4000 0400 - 0x4000 07FF	CPTIMER1
		0x4000 0000 - 0x4000 03FF	CPTIMER0
SRAM		0x2000 8000 - 0x3FFF FFFF	保留
		0x2000 0000 - 0x2000 7FFF	SRAM (32KB)
Code		0x1FFF F820 - 0x1FFF FFFF	保留
		0x1FFF F800 - 0x1FFF F81F	Option Bytes (32B)
		0x1FFF E000 - 0x1FFF F7FF	System Memory (6KB)
		0x1FFF 7200 - 0x1FFF DFFF	保留
		0x1FFF 7000~0x1FFF 71FF	OTP (512B)
		0x1000 8000 - 0x1FFF 6FFF	保留
		0x1000 0000 - 0x1000 7FFF	SRAM Aliased

预先定义的地址空间	总线	地址范围	外设
		0x0881 0000 - 0x0FFF FFFF	保留
		0x0880 0000 - 0x0880 FFFF	Data Flash (64KB)
		0x0804 0000 - 0x0807 FFFF	保留
		0x0800 0000 - 0x0803 FFFF	Main Flash (256KB)
		0x0002 6000 - 0x07FF FFFF	保留
		0x0000 0000 - 0x0002 5FFF	Aliased to Flash or system memory

1.3.1. 片上 SRAM 存储器

GD32M531 系列设备包含高达 32KB 的片上 SRAM，起始地址为 0x2000 0000。它支持字节（8 位）、半字（16 位）和字（32 位）访问。

在 SYSCFG_SRAMCFG 寄存器中的 SRAM_REMAP_SEL[1:0]可以用来控制映射到 0x1000 0000 的 SRAM 的大小（即 0x2000 0000 的 SRAM 重映射）。CPU 可以通过 CBUS 而不是 SBUS 访问映射到 0x1000 0000 的 SRAM。

ECC

在读写 SRAM 时，它支持 7 位 ECC 功能。能够纠正 1 比特，并检测多比特（两比特）错误。

在读取 SRAM 之前必须先写入，否则可能会引起 ECC 错误。未对齐的读操作将按照 32 位读操作执行。未对齐的写操作将产生读-改-写的过程。例如，当 16 位数据被写入 SRAM 时，首先从 SRAM 中读出另外 16 位数据，需要写入的 16 位数据被组合成 32 位数据，最后将 32 位数据一起写入 SRAM。因此，在初始化 SRAM 时，只能以 32 位宽度进行写入。

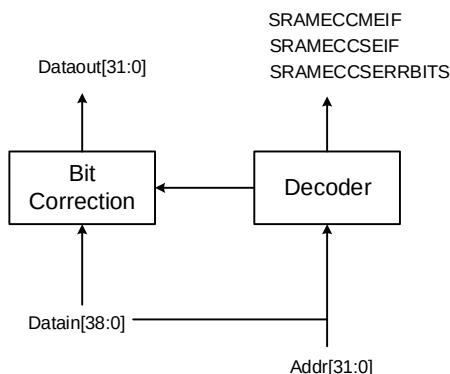
ECC 模块由编码器和解码器组成。

编码器：执行 SRAM 写操作时，将生成一个 7 位的 ECC 码，并与数据一起写入 SRAM。

解码器：在执行 SRAM 读取操作时，它使用与编码器相同的算法来解码并生成一个 7 位的 ECC 码。ECC 码包括 ECC 错误状态和信息，指示 32 位数据中哪个位有单比特错误。

解码器如 [图 1-3. ECC 解码器](#) 如下：

图 1-3. ECC 解码器



EEIC

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

表 1-3. ECC 中断使能和标志

ECC	中断使能	事件标志	锁定连接使能
SRAM	ECCSEIE0	ECCSEIF0	-
	ECCMEIE0	ECCMEIF0	SRAM_LOCK
CAN receive FIFO	ECCSEIE1	ECCSEIF1	-
	ECCMEIE1	ECCMEIF1	CANRX_LOCK
CAN filter	ECCSEIE2	ECCSEIF2	-
	ECCMEIE2	ECCMEIF2	CANFILTER_LOCK
Flash SRAM	ECCSEIE3	ECCSEIF3	-
	ECCMEIE3	ECCMEIF3	FLASHSRAM_LOCK
Flash	ECCMEIE4	ECCMEIF4	FLASH_LOCK

单比特可纠错事件

当在 SRAM / CAN 接收 FIFO / CAN 过滤器 / Flash SRAM 中检测到单比特可纠错事件时，EEIC:

- (1) 在 SYSCFG_STAT 寄存器中，ECCSEIFx (x = 0..3) 位将被置位。软件可写 1 来清除。
- (2) ECCSEBIT[5:0]和 ECCADDR[12:0]位在 SYSCFG_CFG2 中记录了单比特可纠错事件的地址和位。

多比特不可纠错事件

当在 SRAM / CAN 接收 FIFO / CAN 滤波器 / Flash SRAM / Flash 中检测到多比特不可纠错事件时，EEIC:

- (1) 在 SYSCFG_STAT 寄存器中，ECCMEIFx (x = 0..4) 位将被置位。软件可写入 1 来清除。

- (2) ECCEADDR[12:0]位在 SYSCFG_CFG2 中记录了发生多比特不可纠错事件的 SRAM 地址。

单比特可纠错中断

设置 ECCSEIE_x($x = 0..3$) 位。当检测到单比特可纠错事件时，将生成相应的中断。

多比特不可纠错中断

设置 ECCMEIE_x($x = 0..4$) 位。当检测到多比特不可纠错事件时，将生成一个 NMI 中断。

1.3.2. 片上 FLASH 存储器概述

器件提供了高密度的片上闪存，其组织方式如下：

- 高达256KB的主Flash存储器。
- 高达 64KB 的数据 Flash 存储器。
- 高达512字节的一次性可编程（OTP）存储器，用于用户数据存储。
- 最多6KB的信息块用于引导程序。
- 32字节选项字节用于配置设备。

请参阅[闪存控制器（FMC）](#)章节以获取更多详细信息。

1.4. 启动配置

GD32M531 器件提供了两种启动源，可以通过 BOOT-PN6 选择。具体细节如下表所示。引脚上的值在复位后的第 4 个 CK_SYS 上升沿上被锁定。用户需要在上电复位或系统复位后设置 BOOT-PN6 引脚以选择所需的启动源。一旦引脚被采样，将会被释放并可以用于其他用途。

表 1-4. 启动模式

引导源选择	引导模式选择引脚
	BOOT-PN6
System Memory	0
Main Flash Memory	1

上电序列或系统复位后，Arm® Cortex®-M33 处理器依次从地址 0x0000 0000 获取栈顶值，从 0x0000 0004 获取启动代码的基地址。然后，它开始从启动代码的基地址执行代码。

根据选定的启动源，主 Flash（从 0x0800 0000 开始的存储空间）或系统存储器（从 0x1FFF E000 开始的存储空间）被映射到起始地址为 0x0000 0000 的启动内存空间。

嵌入式引导加载程序位于系统内存中，用于重新编程 Flash 存储器。

1.5. 系统配置控制器

系统配置控制器（SYSCFG）的主要目的是以下内容：

- I2C Fm+配置。
- 管理外部中断线连接到GPIOs。
- 重新映射存储器区域。
- 模拟开关配置管理。
- 管理定时器中断或POC输入锁定。

1.6. 系统配置寄存器

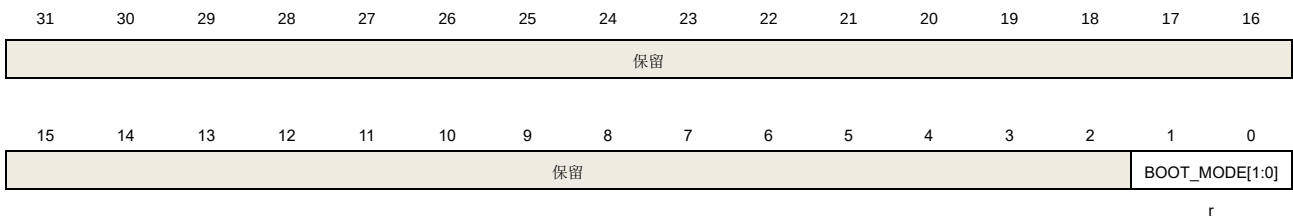
SYSTEM 基地址: 0x4001 0000

1.6.1. 配置寄存器 0 (SYSCFG_CFG0)

地址偏移: 0x00

复位值: 0x0000 000X (X 表示 BOOT_MODE[1:0] 可以是任何值, 根据 BOOT-PN6 引脚确定)

这个寄存器必须以字 (32 位) 访问。



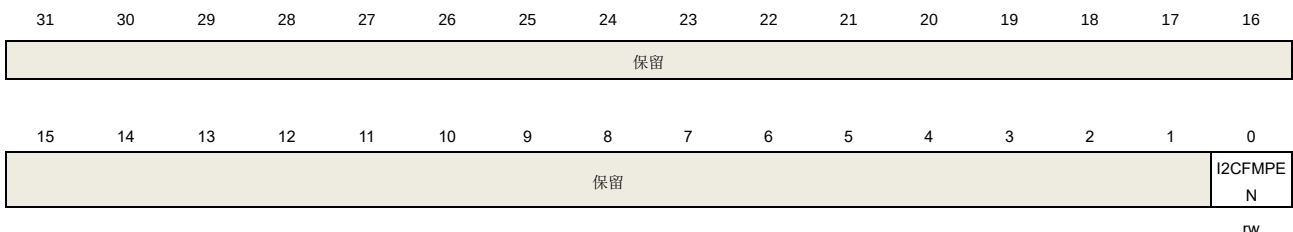
位/位域	名称	描述
31:2	保留	必须保持复位值。
1:0	BOOT_MODE[1:0]	引导模式 (详情可参考 表1-4. 启动模式) Bit0映射到BOOT-PN6引脚, BIT1的值为0。 x0: 从系统存储器启动 01: 从主Flash启动 11: 保留

1.6.2. 配置寄存器 1 (SYSCFG_CFG1)

地址偏移: 0x04

复位值: 0x0000 0000

这个寄存器必须通过字 (32 位) 来访问。



位/位域	名称	描述
31:1	保留	必须保持复位值。
0	I2CFMPEN	使能I2C的Fm+模式 0: 禁能Fm+模式

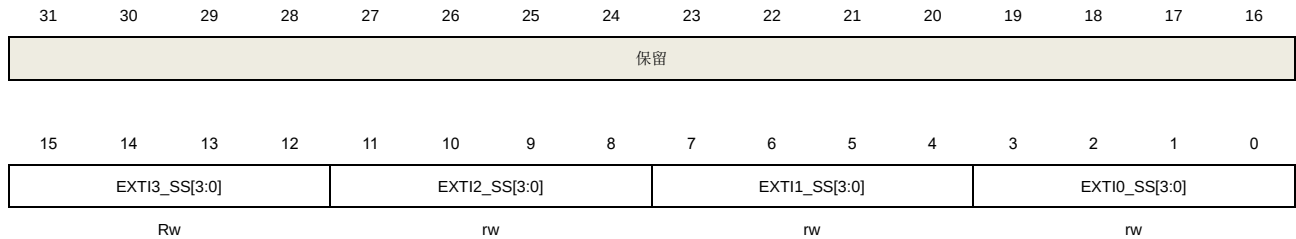
1: 使能Fm+模式

1.6.3. EXTI 源选择寄存器 0 (SYSCFG_EXTISS0)

地址偏移: 0x08

复位值: 0x0000 0000

这个寄存器必须通过字 (32 位) 来访问。



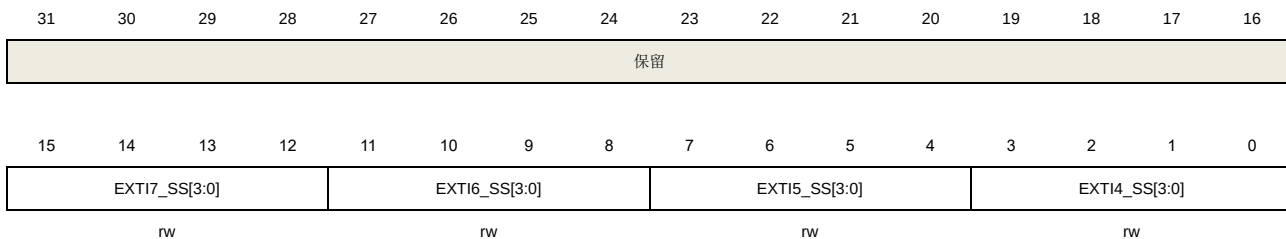
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI3_SS[3:0]	EXTI3源选择 0000: PF12引脚 其他配置保留
11:8	EXTI2_SS[3:0]	EXTI2源选择 0000: PA0引脚 0001: PC12引脚 0010: PF14引脚 0011: PG12引脚 其他配置保留
7:4	EXTI1_SS[3:0]	EXTI1源选择 0000: PA9引脚 0001: PC11引脚 0010: PE13引脚 0011: PC9引脚 其他配置保留
3:0	EXTI0_SS[3:0]	EXTI0源选择 0000: PA8引脚 0001: PC10引脚 0010: PN2引脚 0011: PC8引脚 其他配置保留

1.6.4. EXTI 源选择寄存器 1 (SYSCFG_EXTISS1)

地址偏移: 0x0C

复位值: 0x0000 0000

这个寄存器需要通过字（32 位）来访问。



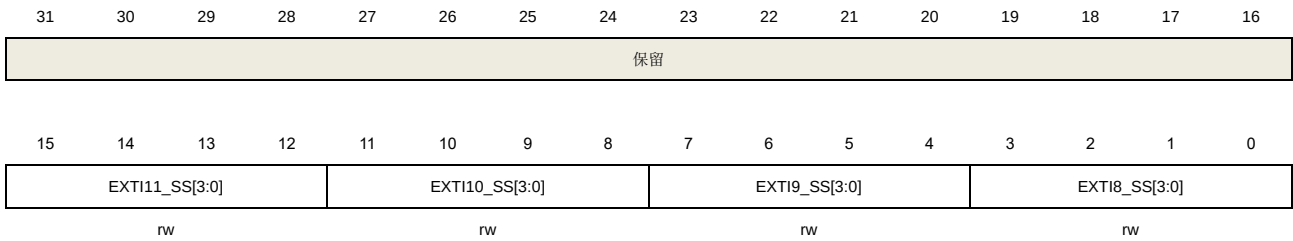
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI7_SS[3:0]	EXTI7源选择 0000: PB0引脚 0001: PC7引脚 0010: PN5引脚 其他配置保留
11:8	EXTI6_SS[3:0]	EXTI6源选择 0000: PB1引脚 0001: PC6引脚 0010: PD2引脚 0011: PG13引脚 其他配置保留
7:4	EXTI5_SS[3:0]	EXTI5源选择 0000: PD8引脚 0001: PG14引脚 0010: PN7引脚 其他配置保留
3:0	EXTI4_SS[3:0]	EXTI4源选择 0000: PA1引脚 0001: PE14引脚 0010: PF9引脚 0011: PF13引脚 其他配置保留

1.6.5. EXTI 源选择寄存器 2 (SYSCFG_EXTISS2)

地址偏移: 0x10

复位值: 0x0000 0000

这个寄存器需要通过字（32 位）来访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI11_SS[3:0]	EXTI11源选择 0000: PC1引脚 0001: PD11引脚 0010: PG11引脚 其他配置保留
11:8	EXTI10_SS[3:0]	EXTI10源选择 0000: PB2引脚 0001: PC0引脚 0010: PD10引脚 0011: PE10引脚 0100: PF10引脚 其他配置保留
7:4	EXTI9_SS[3:0]	EXTI9源选择 0000: PD5引脚 0001: PD9引脚 0010: PE9引脚 0011: PF11引脚 其他配置保留
3:0	EXTI8_SS[3:0]	EXTI8源选择 0000: PD4引脚 0001: PE8引脚 0010: PF8引脚 0011: PG15引脚 其他配置保留

1.6.6. EXTI 源选择寄存器 3（SYSCFG_EXTISS3）

地址偏移：0x14

复位值：0x0000 0000

这个寄存器需要通过字（32 位）来访问。

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

保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI15_SS[3:0]				EXTI14_SS[3:0]				EXTI13_SS[3:0]				EXTI12_SS[3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	EXTI15_SS[3:0]	EXTI15源选择 0000: PB15引脚 0001: PC5引脚 其他配置保留
11:8	EXTI14_SS[3:0]	EXTI14源选择 0000: PB14引脚 0001: PC4引脚 0010: PD14引脚 0011: PE11引脚 其他配置保留
7:4	EXTI13_SS[3:0]	EXTI13源选择 0000: PC3引脚 0001: PD13引脚 其他配置保留
3:0	EXTI12_SS[3:0]	EXTI12源选择 0000: PC2引脚 0001: PD12引脚 0010: PE12引脚 其他配置保留

1.6.7. 状态寄存器（SYSCFG_STAT）

地址偏移量：0x1C

复位值：0x0000 0000

这个寄存器可以通过字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		FWDGT_NMIIF	WWDGT_NMIIF	PIN_NMIIF	HXTAL_NMIIF	保留									
rc_w1		r	r	r	r										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留							ECCMEIF4	ECCSEIF3	ECCMEIF3	ECCSEIF2	ECCMEIF2	ECCSEIF1	ECCMEIF1	ECCSEIF0	ECCMEIF0
							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
31:30	保留	必须保持复位值。
29	FWDGT_NMIIF	FWDGT NMI中断标志 软件写1清零。 0: 未检测到FWDGT NMI事件 1: 检测到FWDGT NMI事件
28	WWDGT_NMIIF	WWDGT NMI中断标志 0: 未检测到WWDGT NMI事件 1: 检测到 WWDGT NMI 事件
27	PIN_NMIIF	NMI引脚中断标志 0: 无输入 1: 来自NMI引脚的中断输入
26	HXTAL_NMIIF	HXTAL时钟监测故障NMI中断状态 0: 未检测到HXTAL时钟监测故障 1: 检测到HXTAL时钟监测故障
25:9	保留	必须保持复位值。
8	ECCMEIF4	Flash 多比特错误标志 软件写 1 清零。. 0: 未检测到 Flash 多比特错误 1: 检测到Flash多比特错误
7	ECCSEIF3	Flash SRAM 单比特错误标志 Flash SRAM 包括 I-cache、D-cache 和快速编程 RAM。 软件写 1 清零。. 0: 未检测到 Flash SRAM 单比特错误 1: 检测到 Flash SRAM 单比特错误
6	ECCMEIF3	Flash SRAM 多比特错误标志 Flash SRAM 包括 I-cache、D-cache 和快速编程 RAM。 软件写 1 清零。 0: 未检测到 Flash SRAM 多比特错误 1: 检测到 Flash SRAM 多比特错误
5	ECCSEIF2	CAN 滤波器单比特错误 软件写 1 清零。. 0: 未检测到 CAN 滤波器单比特错误 1: 检测到 CAN 滤波器单比特错误
4	ECCMEIF2	CAN 滤波器多比特错误 软件写 1 清零。. 0: 未检测到 CAN 滤波器多比特错误

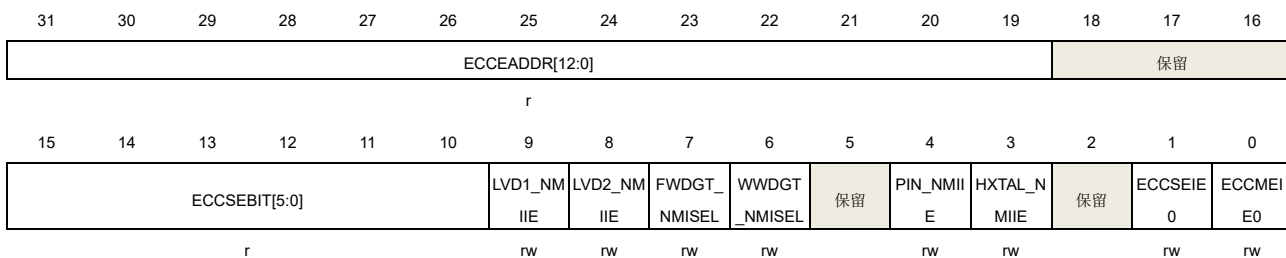
		1: 检测到 CAN 滤波器多比特错误
3	ECCSEIF1	CAN 接收 FIFO 单比特错误标志 软件写 1 清零。 0: CAN 接收 FIFO 单比特错误 1: CAN 接收 FIFO 单比特错误
2	ECCMEIF1	CAN 接收 FIFO 多比特错误标志 软件写 1 清零。 0: CAN 接收 FIFO 多比特错误 1: CAN 接收 FIFO 多比特错误
1	ECCSEIF0	SRAM 单比特错误标志 软件写 1 清零。 0: 未检测到 SRAM 单比特错误 1: 检测到 SRAM 单比特错误
0	ECCMEIF0	SRAM 多比特错误标志 软件写 1 清零。 0: 未检测到 SRAM 多比特错误 1: 检测到 SRAM 多比特错误

1.6.8. 配置寄存器 2 (SYSCFG_CFG2)

地址偏移: 0x20

复位值: 0x0000 0008

这个寄存器可以通过字 (32 位) 访问。



位/位域	名称	描述
31:19	ECCEADDR[12:0]	记录上一次发生 SRAM ECC 事件的 SRAM 故障系统地址 Note: SRAM 物理字地址 = SRAM 基地址/4 + ECCEADDR
18:16	保留	必须保持复位值。
15:10	ECCSEBIT[5:0]	发生 ECC 单比特可纠错事件的比特 0: 无错误 1: 比特 0 ... 32: 比特 31

9	LVD1_NMIIE	LVD1 NMI中断使能 0: 禁能 1: 使能
8	LVD2_NMIIE	LVD2 NMI中断使能 0: 禁能 1: 使能
7	FWDGT_NMISEL	FWDGT NMI 中断类型选择 0: 可屏蔽中断 1: 不可屏蔽中断
6	WWDGT_NMISEL	WWDGT NMI 中断类型选择 0: 可屏蔽中断 1: 不可屏蔽中断
5	保留	必须保持复位值。
4	PIN_NMIIE	NMI引脚中断使能 0: 禁能 1: 使能
3	HXTAL_NMIIE	HXTAL时钟监控NMI中断使能 0: 禁能 1: 使能
2	保留	必须保持复位值。
1	ECCSEIE0	SRAM单比特中断使能 0: 禁能 1: 使能
0	ECCMEIE0	SRAM多比特中断使能 0: 禁能 1: 使能

1.6.9. 配置寄存器 3 (SYSCFG_CFG3)

地址偏移: 0x24

复位值: 0x0000 0000

这个寄存器可以通过字 (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												ECCSEIE	ECCMEI	ECCSEIE	ECCMEIE
												2	E2	1	1
												RW	RW	RW	RW

位/位域	名称	描述
31:4	保留	必须保持复位值。
3	ECCSEIE2	CAN滤波器单比特中断使能 0: 禁能 1: 使能
2	ECCMEIE2	CAN滤波器多比特中断使能 0: 禁能 1: 使能
1	ECCSEIE1	CAN接收FIFO单比特中断使能 0: 禁能 1: 使能
0	ECCMEIE1	CAN接收FIFO多比特中断使能 0: 禁能 1: 使能

1.6.10. 配置寄存器 4 (SYSCFG_CFG4)

地址偏移: 0x28

复位值: 0x0000 0000

这个寄存器可以通过字 (32 位) 来访问。

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													ECCMEI E4	ECCSEIE 3	ECCMEIE 3
													rw	rw	rw

位/位域	名称	描述
31:3	保留	必须保持复位值。
2	ECCMEIE4	FLASH多比特中断使能 0: 禁能 1: 使能
1	ECCSEIE3	FLASH SRAM单比特中断使能 0: 禁能 1: 使能
0	ECCMEIE3	FLASH SRAM多比特中断使能 0: 禁能

1.6.11. TIMER 输入源选择寄存器 0 (SYSCFG_TIMERCISEL0)

地址偏移: 0x34

复位值: 0x0000 0000

这个寄存器必须通过字 (32 位) 来访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIMER0_CI3_SEL[3:0]				TIMER0_CI2_SEL[3:0]				TIMER0_CI1_SEL[3:0]				TIMER0_CI0_SEL[3:0]			
Rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER7_CI3_SEL[3:0]				TIMER7_CI2_SEL[3:0]				TIMER7_CI1_SEL[3:0]				TIMER7_CI0_SEL[3:0]			
Rw				rw				rw				rw			

位/位域	名称	描述
31:28	TIMER0_CI3_SEL[3:0]	选择 TIMER0_CI3 的输入源 0000: TIMER0_CH3 输入 其他: 保留
27:24	TIMER0_CI2_SEL[3:0]	选择 TIMER0_CI2 的输入源 0000: TIMER0_CH2 输入 其他: 保留
23:20	TIMER0_CI1_SEL[3:0]	选择 TIMER0_CI1 的输入源 0000: TIMER0_CH1 输入 其他: 保留
19:16	TIMER0_CI0_SEL[3:0]	选择 TIMER0_CI0 的输入源 0000: TIMER0_CH0 输入 0001: CMP0 输出 其他: 保留
15:12	TIMER7_CI3_SEL[3:0]	选择 TIMER7_CI3 的输入源 0000: TIMER7_CH3 输入 其他: 保留
11:8	TIMER7_CI2_SEL[3:0]	选择 TIMER7_CI2 的输入源 0000: TIMER7_CH2 输入 其他: 保留
7:4	TIMER7_CI1_SEL[3:0]	选择 TIMER7_CI1 的输入源 0000: TIMER7_CH1 输入 其他: 保留
3:0	TIMER7_CI0_SEL[3:0]	选择 TIMER7_CI0 的输入源 0000: TIMER7_CH0 输入

0001: CMP1 输出

其他: 保留

1.6.12. TIMER 输入源选择寄存器 1 (SYSCFG_TIMERCISEL1)

地址偏移: 0x38

复位值: 0x0000 0000

这个寄存器必须通过字 (32 位) 来访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIMER1_CI3_SEL[3:0]				TIMER1_CI2_SEL[3:0]				TIMER1_CI1_SEL[3:0]				TIMER1_CIO_SEL[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER2_CI3_SEL[3:0]				TIMER2_CI2_SEL[3:0]				TIMER2_CI1_SEL[3:0]				TIMER2_CIO_SEL[3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:28	TIMER1_CI3_SEL[3:0]	TIMER1_CI3 的输入源 0000: TIMER1_CH3 输入 0001: CMP0 输出 0010: CMP1 输出 0011: CMP0 输出或 CMP1 输出 其他: 保留
27:24	TIMER1_CI2_SEL[3:0]	TIMER1_CI2 的输入源 0000: TIMER1_CH2 输入 其他: 保留
23:20	TIMER1_CI1_SEL[3:0]	TIMER1_CI1 的输入源 0000: TIMER1_CH1 输入 1000: TIMER 外部触发0 1001: TIMER 外部触发1 1010: TIMER 外部触发2 1011: TIMER 外部触发3 其他: 保留
19:16	TIMER1_CIO_SEL[3:0]	TIMER1_CIO 的输入源 0000: TIMER1_CH0 输入 1000: TIMER 外部触发0 1001: TIMER 外部触发1 1010: TIMER 外部触发2 1011: TIMER 外部触发3 其他: 保留
15:12	TIMER2_CI3_SEL[3:0]	TIMER2_CI3 的输入源 0000: TIMER2_CH3 输入

		1110: IRC32K
		1111: CK_OUT
		其他: 保留
11:8	TIMER2_CI2_SEL[3:0]	TIMER2_CI2的输入源 0000: TIMER2_CH2输入 其他: 保留
7:4	TIMER2_CI1_SEL[3:0]	TIMER2_CI1的输入源 0000: TIMER2_CH1输入 1000: TIMER外部触发0 1001: TIMER外部触发1 1010: TIMER外部触发2 1011: TIMER外部触发3 其他: 保留
3:0	TIMER2_CIO_SEL[3:0]	TIMER2_CIO的输入源 0000: TIMER2_CH0输入 0001: CMP0输出 0010: CMP1输出 0011: CMP0输出或CMP1输出 1000: TIMER外部触发0 1001: TIMER外部触发1 1010: TIMER外部触发2 1011: TIMER外部触发3 其他: 保留

1.6.13. 配置寄存器 3 (SYSCFG_CFG5)

地址偏移: 0x54

复位值: 0x0000 0000 (系统复位值)

这个寄存器必须通过字 (32 位) 来访问。

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									FLASH_L OCK	FLASHS RAM_LO CK	CANFILT ER_LOC K	CANRX_ LOCK	SRAM_L OCK	LVD_LOC K	LOCKUP _LOCK
									rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:7	保留	必须保持复位值。
6	FLASH_LOCK	Flash ECC 多比特错误锁定使能

		该位由软件设置，仅在系统复位被清除。
		该位用于使能和锁定 Flash ECC 错误连接到 TIMER0/7 的 break 输入。
		0: 断开
		1: 连接
5	FLASHSRAM_LOCK	Flash SRAM ECC 多比特错误锁定使能
		该位由软件设置，仅在系统复位被清除。
		该位用于使能和锁定 Flash SRAM ECC 错误连接到 TIMER0/7 的 break 输入。
		0: 断开
		1: 连接
4	CANFILTER_LOCK	CAN filter ECC 多比特错误锁定使能
		该位由软件设置，仅在系统复位被清除。
		该位用于使能和锁定 CAN 滤波器 ECC 错误连接到 TIMER0/7 的 break 输入。
		0: 断开
		1: 连接
3	CANRX_LOCK	CAN 接收 FIFO ECC 多比特错误锁定使能
		该位由软件设置，仅在系统复位被清除。
		该位用于使能和锁定 CAN 接收 FIFO ECC 错误连接到 TIMER0/7 的 break 输入。
		0: 断开
		1: 连接
2	SRAM_LOCK	SRAM ECC 多比特错误锁定使能
		该位由软件设置，仅在系统复位被清除。
		该位用于使能和锁定 SRAM ECC 错误连接到 TIMER0/7 的 break 输入。
		0: 断开
		1: 连接
1	LVD_LOCK	LVD 锁定使能
		该位由软件设置，仅在系统复位被清除。
		该位用于使能和锁定 LVD 连接到 TIMER0/7 的 break 输入。
		0: 断开
		1: 连接
0	LOCKUP_LOCK	Cortex-M33 lockup (Hardfault) 输出锁定使能
		该位由软件设置，仅在系统复位被清除。
		POC 该位用于使能和锁定 LVD 连接到 TIMER0/7 的 break 输入和 POC 输入。
		0: 断开
		1: 连接

1.6.14. SRAM 配置寄存器 (SYSCFG_SRAMCFG)

地址偏移: 0x68

复位值: 0x0000 0003

这个寄存器必须通过字 (32 位) 来访问。

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

位/位域	名称	描述
31:2	保留	必须保持复位值。
1:0	SRAM_REMAP_SEL [1:0]	这个位用于控制在0x1000 0000中别名SRAM的大小(0x2000 0000的SRAM重新映射)。CPU可以通过CBUS而不是SBUS访问0x1000 0000的别名SRAM。 00: 0KB 01: 8KB 10: 16KB 11: 32KB

1.6.15. 保护配置寄存器（SYSCFG_PRCFG）

地址偏移：0x6C

复位值：0x0000 0000（电源复位）

这个寄存器需要按字（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
PRKEY[7:0]								保留				PRC2	保留	PRC1	PRC0
rw												rw		rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	PRKEY[7:0]	保护区域写密钥 仅在软件写入正确的密钥（0xA5）到这些位，保护选择位PRCFG2 / PRCFG1 / PRCFG0可用。
7:4	保留	必须保持复位值。
3	PRCFG2	保护配置选择2 使能PMU寄存器写使能：PMU_LVD1CTL、PMU_LVD2CTL。 0: 写禁能 1: 写使能
2	保留	必须保持复位值。

1	PRCFG1	保护配置选择1 使能RCU寄存器写使能：RCU_APB1RST、RCU_APB2RST、RCU_AHBEN、RCU_APB1EN、RCU_APB2EN、RCU_RSTSCK、RCU_AHBRSR。 0：写禁能 1：写使能
0	PRCFG0	保护配置选择0 使能RCU寄存器写使能：RCU_CTL、RCU_CFG0、RCU_INT、RCU_CFG1、RCU_CFG2。 0：写禁能 1：写使能

1.6.16. TIMERx ADC 启动请求配置寄存器（TADSRCFG）

地址偏移：0x70

复位值：0x0000 0000

这个寄存器需要通过字（32 位）来访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15	ADCSM2EN	ADCSM2引脚输出使能 0：禁能 1：使能
14:12	保留	必须保持复位值。
11:8	ADCSM2SEL[3:0]	选择连接到ADCSM2引脚的ADC触发源。 0000: TIMER0_TRGOF 0001: TIMER0_TRGUF 0010: TIMER0_TRGA 0011: TIMER0_TRGB 0100: TIMER0_TRGAB 0101: TIMER7_TRGOF 0110: TIMER7_TRGUF 0111: TIMER7_TRGA 1000: TIMER7_TRGB 1001: TIMER7_TRGAB

		1010~1111: 保留
7	ADCSM1EN	ADCSM1引脚输出使能 0: 禁能 1: 使能
6:4	保留	必须保持复位值。
3:0	ADCSM1SEL[3:0]	选择连接到ADCSM1引脚的ADC触发源。 0000: TIMER0_TRGOF 0001: TIMER0_TRGUF 0010: TIMER0_TRGA 0011: TIMER0_TRGB 0100: TIMER0_TRGAB 0101: TIMER7_TRGOF 0110: TIMER7_TRGUF 0111: TIMER7_TRGA 1000: TIMER7_TRGB 1001: TIMER7_TRGAB 1010~1111: 保留

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

TIMER0 的地址偏移: 0x100

TIMER7 的地址偏移: 0x124

复位值: 0x0000 0000

这个寄存器必须通过字 (32 位) 来访问。

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 (IT10) 00010: 内部触发输入 1 (IT11) 00011: 内部触发输入 2 (IT12) 00100: 内部触发输入 3 (IT13)

		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) 01011: 滤波后的多模式通道 0 输入 (MCI0FEM0) 01100: 滤波后的多模式通道 1 输入 (MCI1FEM1) 01101: 滤波后的多模式通道 2 输入 (MCI2FEM2) 01110: 滤波后的多模式通道 3 输入 (MCI3FEM3) 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: 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) 01011: 滤波后的多模式通道 0 输入 (MCI0FEM0) 01100: 滤波后的多模式通道 1 输入 (MCI1FEM1) 01101: 滤波后的多模式通道 2 输入 (MCI2FEM2) 01110: 滤波后的多模式通道 3 输入 (MCI3FEM3) 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)

		01011: 滤波后的多模式通道 0 输入 (MCIOFEM0)
		01100: 滤波后的多模式通道 1 输入 (MCII1FEM1)
		01101: 滤波后的多模式通道 2 输入 (MCII2FEM2)
		01110: 滤波后的多模式通道 3 输入 (MCII3FEM3)
		其他: 保留
15	保留	必须保持复位值。
14:10	TSCFG2[4:0]	编码器模式 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]	编码器模式 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]	编码器模式 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。

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

TIMER0 的地址偏移: 0x104

TIMER7 的地址偏移: 0x128

复位值: 0x0000 0000

这个寄存器必须通过字 (32 位) 来访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	TSCFG11[4:0]					TSCFG10[4:0]					TSCFG9[4:0]				
rw					rw					rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						TSCFG7[4:0]					TSCFG6[4:0]				
						rw					rw				

位/位域	名称	描述
31	保留	必须保持复位值。
30:26	TSCFG11[4:0]	非正交译码器模式 2 配置 00000: 禁能 其他: CI0FE0 和 CI1FE1 信号的下降/上升双沿计数。CH0P 和 CH1P 信号决定计数方向。
25:21	TSCFG10[4:0]	非正交译码器模式 1 配置 00000: 禁能 其他: CI1FE1 作为计数脉冲, CI0FE0 作为计数选择信号。CH1P 决定计数器在

		CI1FE1 的上升或下降边沿计数。
20:16	TSCFG9[4:0]	非正交译码器模式 0 配置 00000: 禁能 其他: CI1FE1 作为计数脉冲, CI0FE0 作为计数选择信号。计数器在 CI1FE1 的上升/下降边沿均计数。
15:10	保留	必须保持复位值。
9:5	TSCFG7[4:0]	复位+事件模式配置 当 TSCFG7[4:0]不为零, 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+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) 01011: 滤波后的多模式通道 0 输入 (MCI0FEM0) 01100: 滤波后的多模式通道 1 输入 (MCI1FEM1) 01101: 滤波后的多模式通道 2 输入 (MCI2FEM2) 01110: 滤波后的多模式通道 3 输入 (MCI3FEM3) 其他: 保留
4:0	TSCFG6[4:0]	外部时钟模式 0 配置 在 TSCFG4[4:0]非零时, 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) 01011: 滤波后的多模式通道 0 输入 (MCI0FEM0) 01100: 滤波后的多模式通道 1 输入 (MCI1FEM1) 01101: 滤波后的多模式通道 2 输入 (MCI2FEM2)

01110: 滤波后的多模式通道 3 输入 (MCI3FEM3)

其他: 保留

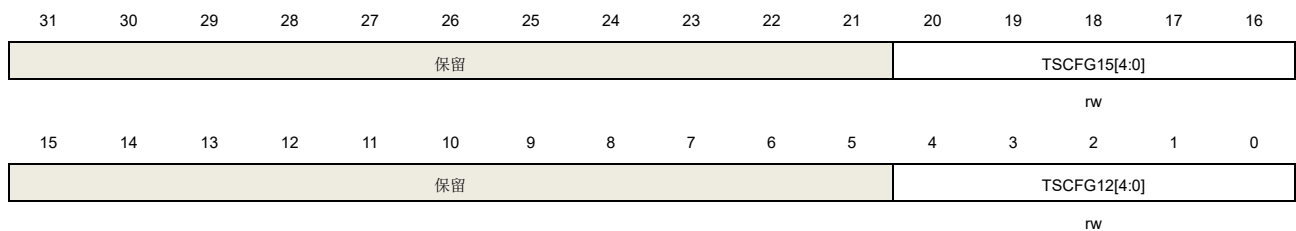
1.6.19. TIMERx 配置寄存器 2 (SYSCFG_TIMERxCFG2, x=0, 7)

TIMER0 的地址偏移: 0x108

TIMER7 的地址偏移: 0x12C

复位值: 0x0000 0000

这个寄存器需要通过字 (32 位) 来访问。



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: 内部触发输入 0 (ITI0) 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) 01011: 滤波后的多模式通道 0 输入 (MCI0FEM0) 01100: 滤波后的多模式通道 1 输入 (MCI1FEM1) 01101: 滤波后的多模式通道 2 输入 (MCI2FEM2) 01110: 滤波后的多模式通道 3 输入 (MCI3FEM3) 其他: 保留 注意: 使用 TSCFG15[4:0]时, 需保证 TSCFGy[4:0] (y=0..9) 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。
15:5	保留	必须保持复位值。
4:0	TSCFG12[4:0]	非正交译码器模式 3 配置 00000: 禁能 其他: 计数器会在 CI0FE0 和 CI1FE1 信号的上升/下降沿计数。CHxP =0 时, 只有

当 ClxFEx 输入信号为高电平或下降沿时，计数器才会计数；CHxP =1 时，只有当 ClxFEx 输入信号为低电平或上升沿时，计数器才会计数。

1.6.20. TIMERx 配置寄存器 0 (SYSCFG_TIMERxCFG0, x=1, 2)

TIMER1 的地址偏移：0x10C

TIMER2 的地址偏移：0x118

复位值：0x0000 0000

这个寄存器必须通过字（32 位）来访问。

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 (ITIO) 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 (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: 滤波后的外部触发输入 (ETIFP) 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 (ITIO) 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 (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: 滤波后的外部触发输入 (ETIFP)

		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） 其他：保留
15	保留	必须保持复位值。
14:10	TSCFG2[4:0]	编码器模式 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]	编码器模式 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]	编码器模式 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。

1.6.21. TIMERx 配置寄存器 1（SYSCFG_TIMERxCFG1, x=1, 2）

TIMER1 的地址偏移：0x110

TIMER2 的地址偏移：0x11C

复位值：0x0000 0000

这个寄存器需要通过字（32 位）来访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	TSCFG11[4:0]					TSCFG10[4:0]					TSCFG9[4:0]				
rw					rw					rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						TSCFG7[4:0]					TSCFG6[4:0]				
						rw					rw				

位/位域	名称	描述

31	保留	必须保持复位值。
30:26	TSCFG11[4:0]	非正交译码器模式 2 配置 00000: 禁能 其他: CI0FE0 和 CI1FE1 信号的下降/上升双沿计数。CH0P 和 CH1P 信号决定计数方向。
25:21	TSCFG10[4:0]	非正交译码器模式 1 配置 00000: 禁能 其他: CI1FE1 作为计数脉冲, CI0FE0 作为计数选择信号。CH1P 决定计数器在 CI1FE1 的上升或下降边沿计数。
20:16	TSCFG9[4:0]	非正交译码器模式 0 配置 00000: 禁能 其他: CI1FE1 作为计数脉冲, CI0FE0 作为计数选择信号。计数器在 CI1FE1 的上升/下降边沿均计数。
15:10	保留	必须保持复位值。
9:5	TSCFG7[4:0]	复位+事件模式配置 当 TSCFG7[4:0]不为零, 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+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) 其他: 保留
4:0	TSCFG6[4:0]	外部时钟模式 0 配置 在 TSCFG4[4:0]非零时, 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) 其他: 保留

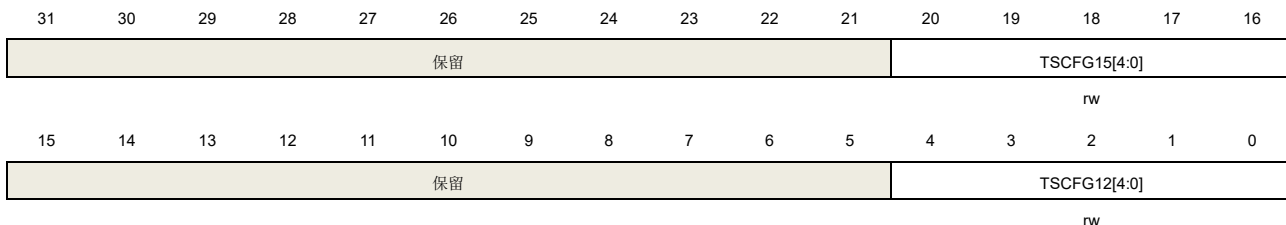
1.6.22. TIMERx 配置寄存器 2 (SYSCFG_TIMERxCFG2, x=1, 2)

TIMER1 的地址偏移量: 0x114

TIMER2 的地址偏移: 0x120

重置值: 0x0000 0000

这个寄存器必须通过字 (32 位) 来访问。



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: 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) 其他: 保留
15:5	保留	必须保持复位值。
4:0	TSCFG12[4:0]	非正交译码器模式 3 配置 00000: 禁能 其他: 计数器会在 CI0FE0 和 CI1FE1 信号的上升/下降沿计数。CHxP =0 时, 只有当 ClxFEx 输入信号为高电平或下降沿时, 计数器才会计数; CHxP =1 时, 只有当 ClxFEx 输入信号为低电平或上升沿时, 计数器才会计数。

1.6.23. 配置寄存器 6 (SYSCFG_CFG6)

地址偏移: 0x204

重置值: 0x0000 0000

这个寄存器必须通过字 (32 位) 来访问。



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

位/位域	名称	描述
31:1	保留	必须保持复位值。
0	CK_CMP_SEL	CK_CMP 选择 0: CK_CMP=fPCLK/2 1: CK_CMP=fPCLK/4

1.1. 电子设备签名

该设备的电子签名包含存储在 Flash 存储器信息块中的内存大小信息和 96 位唯一设备 ID。96 位唯一设备 ID 对于任何设备都是唯一的。它可以用作序列号，或作为安全密钥的一部分等。

1.1.1. 存储密度信息

基址：0x1FFF F7E0

该值是原厂预设的，用户永远无法更改。

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

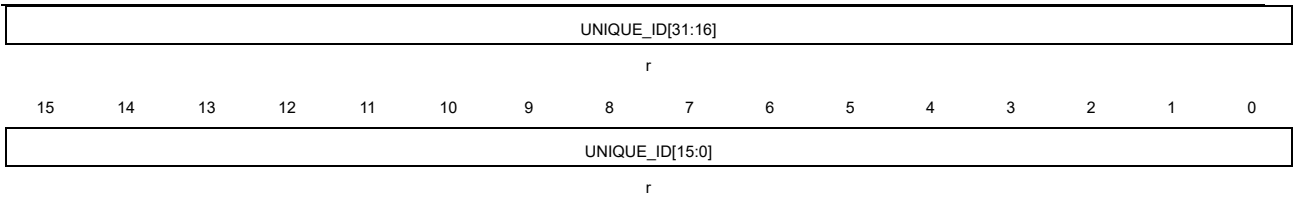
位/位域	名称	描述
31:16	SRAM_DENSITY [15:0]	SRAM 存储器容量 该值表明芯片的片上 SRAM 存储器容量，以 Kbytes 为单位 例如：0x0020 表示 32Kbytes。
15:0	FLASH_DENSITY [15:0]	Flash 存储器容量 该值表明芯片的片上FLASH容量，以Kbytes为单位 例如：0x0100 表示 256Kbytes。

1.1.2. 设备唯一 ID（96 位）

基址：0x1FFF F7E8

该值是原厂预设的，用户无法更改。

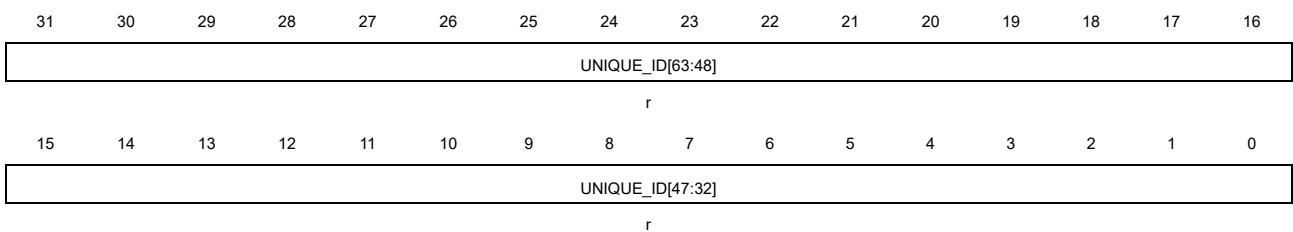
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16



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

基址：0x1FFF F7EC

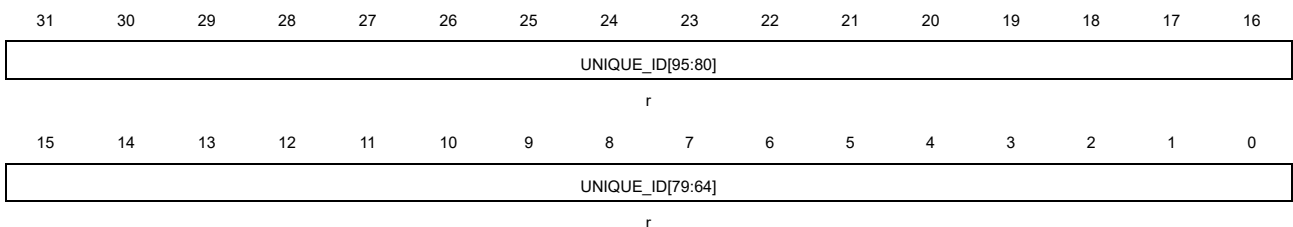
这个值是原厂预设的，用户永远无法更改。



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

基址：0x1FFF F7F0

该值是原厂预设的，用户永远无法更改。



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

2. 闪存控制器（FMC）

2.1. 简介

闪存控制器（FMC），提供了片上闪存需要的所有功能。在主闪存和数据闪存内，CPU 执行指令需要少量等待时间。FMC 也提供了页擦除，整片擦除，以及编程操作。

2.2. 主要特征

- 高达256KB的片上主闪存可用于存储指令或数据，最大512B OTP，最大64KB数据闪存，6KB的信息块；
 - 主闪存：最大256KB
 - 数据闪存：64KB
 - OTP：512B，可用来存储用户数据
 - 信息块：6KB
 - 选项字节：32B，用于用户应用程序需求
- 支持128位四字编程，页擦除和整片擦除操作；
- 闪存页大小为1KB；
- 支持单比特可纠错和多比特检测ECC功能；
- 支持快速编程；
- 支持查空；
- 在主闪存的256K字节空间及64K字节数据闪存内，CPU执行指令需要0~6个等待时间；
- 闪存支持128位宽数据读取；
- 预取缓冲区以加速读操作；
- 支持8K字节I-cache高速缓存区，256字节D-cache高速缓存区；
- 当系统复位时，选项字节被上载到选项字节控制寄存器；
- 具有安全保护状态，可阻止对代码或数据的非法读访问；
- 具有擦除和编程保护状态，可阻止意外写操作。

2.3. 功能说明

2.3.1. 闪存结构

闪存页大小为 1KB。主存储闪存的每页都可以单独擦除。闪存结构细节见[表 2-1. GD32M531 闪存基地址和构成](#)。

表 2-1. GD32M531 闪存基地址和构成

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

闪存块	名称	地址范围	大小（字节）
数据闪存块	第255页	0x0803 FC00 - 0x0803 FFFF	1KB
	第0页	0x0880 0000 - 0x0880 03FF	1KB
	第1页	0x0880 0400 - 0x0880 07FF	1KB
	第2页	0x0880 0800 - 0x0880 0BFF	1KB
	第63页	0x0880 FC00 - 0x0880 FFFF	1KB
信息块	引导装载程序	0x1FFF E000- 0x1FFF F7FF	6KB
选项字节块	选项字节	0x1FFF F800 - 0x1FFF F81F	32B
一次性编程块	选项字节	0x1FFF 7000~0x1FFF 71FF	512B

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

2.3.2. 错误检查与纠正（ECC）

ECC机制支持：

- 单个位错误检测和纠错
- 双位错误检测

当单个位错误被检测与纠正时：

- 当ECC错误发生时，FMC_ECCCS寄存器中的ECCCOR位将置位。如果设置了FMC_ECCCS寄存器中的ECCCORIE位，则会产生对应中断。OB_ECC / OTP_ECC / MF_ECC / DF_ECC / SYS_ECC / SYSECCBK位表示出现错误的存储空间。ECCADDR表示各个存储空间出现错误的偏移地址。向FMC_ECCCS寄存器ECCCOR位写1可以清除ECCCOR位。

当检测到双位错误时：

- 当ECC错误发生时，FMC_ECCCS寄存器中的ECCDET位将置位。如果设置了FMC_ECCCS寄存器中的ECCDETIE位，则会产生对应中断。OB_ECC / OTP_ECC / MF_ECC / DF_ECC / SYS_ECC / SYSECCBK表示出现错误的存储空间。ECCADDR位表示各个存储空间出现错误的偏移地址。数据返回全F。向FMC_ECCCS寄存器ECCDET位写1可以清除ECCDET位。

注意：

1. 主闪存中的数据是按137位存储的，每个四字（128位）后加9位纠错码。9位纠错码由硬件自动计算，用户不可访问。数据闪存中的数据是按72位存储的，每个双字（64位）后加8位纠错码。8位纠错码由硬件自动计算，用户不可访问。

- 对于主闪存原始数据0x1FF FFFF FFFF FFFF FFFF FFFF FFFF FFFF，不支持ECC。对于数据闪存原始数据0xFF FFFF FFFF FFFF FFFF，不支持ECC。
- OB_ECC / OTP_ECC / SYS_ECC / MF_ECC / ECCADDR / SYSECCBK表示在单个位或者双位错误中首次出现错误的位置。如果ECCCOR或ECCDET置1，即使有新的错误产生，这些位的值将不会改变。
- 当上报ECC错误时，如果数据仍然存在于当前的缓冲区或预取缓冲区中，即使清除ECCCOR和ECCDET，重新读取该错误地址也不会产生ECC错误。
- 当编程原始数据为全F时，ECC计算会bypass本次操作。不需要擦除，再次编程其他值可正常写入。

2.3.3. 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使用 CPU 的 AHB BUS 总线。

增加等待状态：

根据 AHB 时钟频率，读闪存时需正确配置 FMC_WS 寄存器中的 WSCNT 位。WSCNT 位和 AHB 时钟频率的对应关系见[表 2-2. WSCNT 与 AHB 时钟频率对应关系](#)。

表 2-2. WSCNT 与 AHB 时钟频率对应关系

AHB时钟频率	WSCNT配置
<= 25MHz	0（0等待状态增加）
<= 50MHz	1（1等待状态增加）
<= 75MHz	2（2等待状态增加）
<= 100MHz	3（3等待状态增加）
<= 125MHz	4（4等待状态增加）
<= 150MHz	5（5等待状态增加）
<= 180MHz	6（6等待状态增加）

如果发生系统复位，AHB 时钟频率为 32MHz，此时 WSCNT 置为 1。

注意：

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

由于添加了等待状态，读效率非常低（例如，120MHz 时需添加 3 个等待状态）。为了加速读操作，需要用到一些功能。

当前缓存区:

当前缓存区总是被使能的。每次从闪存中读取数据时，将数据读取并存储在当前缓冲区中。（64 位当前缓存区用于数据，128 位当前缓存区用于指令）。因为 CPU 每次读操作只需要 32 位或 16 位数据。因此在顺序代码下，CPU 所需数据可以从当前缓存区获取而不必重复从闪存中获取。

预取缓存区:

置位 FMC_WS 寄存器中 PFEN 位来使能预取缓存区。预取缓存区仅在取指时执行。在顺序代码下，当 CPU 执行来自当前缓存区的数据时（128 位），按 32 位执行时需要至少 4 个时钟周期，按 16 位执行时需要至少 8 个时钟周期。在这种情况下，从闪存中预取下一个双字地址的数据并存储在预取缓存区。当 CPU 执行完当前缓存区的数据时，预取缓存区提供下次需要执行的数据。

I-cache 高速缓存区:

置位 FMC_WS 寄存器中 ICEN 位来使能 I-cache 高速缓存区。I-cache 高速缓存区仅在取指令时使用。I-cache 高速缓存区为 8K 字节，由 256 条缓存线组成，每条缓存线为 2*128 位。

当指令数据存在于 I-cache 高速缓存区（I-cache 命中）时，CPU 从 I-cache 高速缓存区读取指令数据时无延迟。当指令数据不存在于 I-cache 高速缓存区（I-cache 失效）并且也不存在于当前缓存区或预取缓存区时，高速缓存线从闪存中读取数据并复制到 I-cache 高速缓存区。当所有高速缓存线被填充，LRU（最近最少使用）策略被用于转移高速缓存线中数据。

D-cache 高速缓存区:

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

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

2.3.4. FMC_CTL 寄存器解锁

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

FMC_CTL 寄存器的 OBPB 位和 OBER 位在 FMC_CTL 寄存器解锁后，仍然被保护。解锁序列是向 FMC_OBKEY 寄存器先后写入 0x45670123 和 0xCDEF89AB，然后硬件会将 FMC_CTL

寄存器的 OBWEN 位置 1。软件可以将 FMC_CTL 的 OBWEN 位清 0 来锁定 FMC_CTL 的 OBPG 位和 OBER 位。

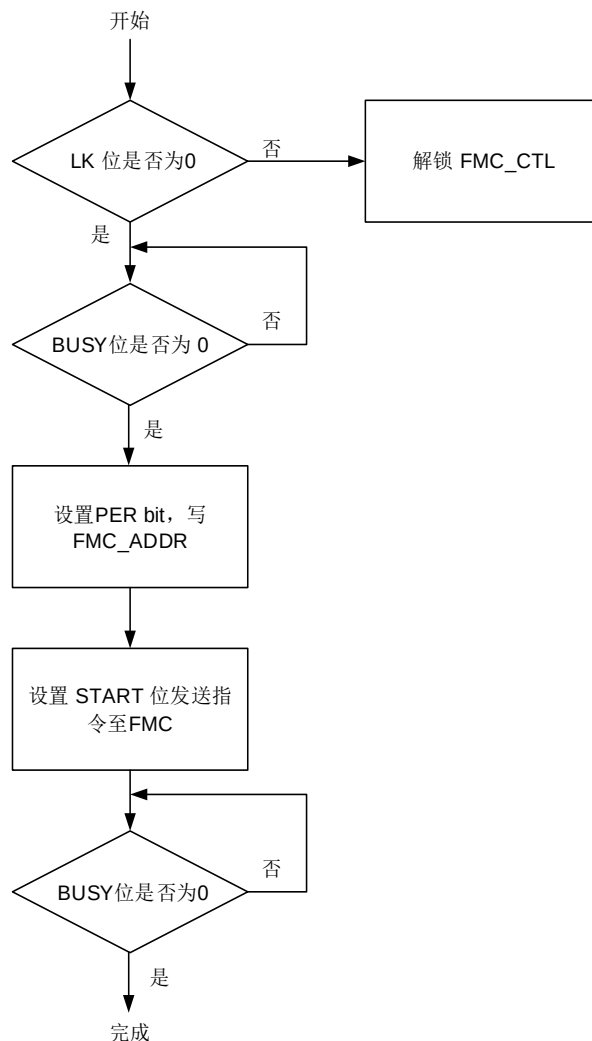
2.3.5. 页擦除

FMC 的页擦除功能使得主存储闪存（BKSEL = 0）和数据闪存（BKSEL = 1）的页内容初始化为高电平。每一页都可以被独立擦除，而不影响其他页内容。页擦除页操作，寄存器设置具体步骤如下：

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

当页擦除成功执行，FMC_STAT 寄存器的 ENDF 位将置位。若 FMC_CTL 寄存器的 ENDIE 位被置 1，则 FMC 将触发一个中断。需要注意的是，用户需确保写入的是正确的擦除地址。否则当待擦除页的地址被用来取指令或访问数据时，软件将会“跑飞”。该情况下，FMC 不会提供任何出错通知。另一方面，对擦写保护的页进行擦除操作将无效。如果 FMC_CTL 寄存器的 ERRIE 位被置位，该操作将触发操作出错中断。中断服务程序可通过检测 FMC_STAT 寄存器的 WPERR 位来判断该中断是否发生。[图 2-1. 页擦除操作流程](#)显示了页擦除操作流程。

图 2-1. 页擦除操作流程



2.3.6. 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（BKSEL = 0）和数据闪存（BKSEL = 1）的内容。当设置 FMC_CTL 寄存器中 MER 为 1 时，擦除过程作用于整片闪存。整片擦除操作，寄存器设置具体步骤如下：

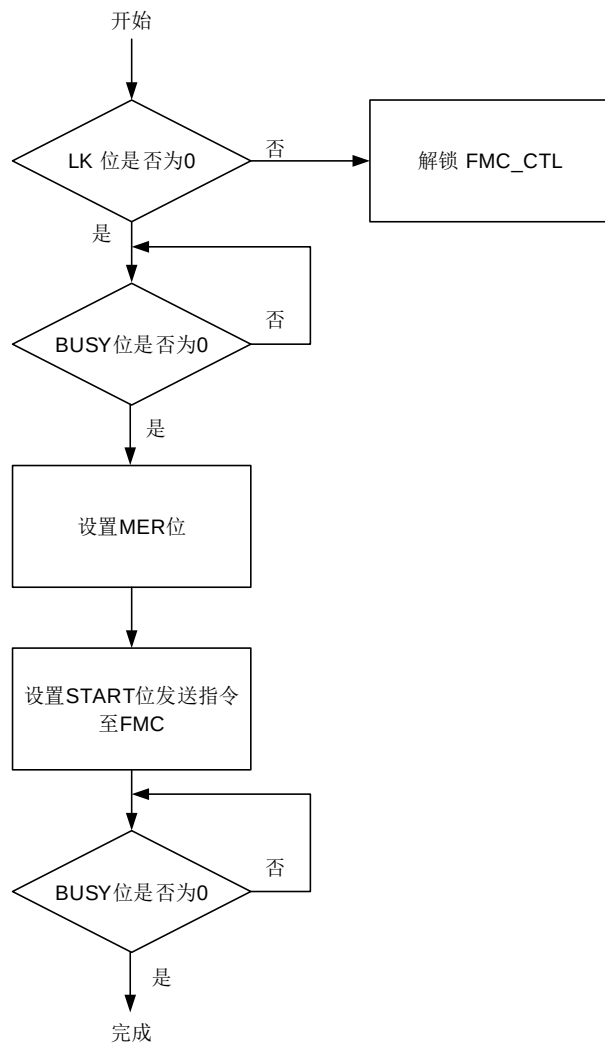
- 确保FMC_CTL寄存器不处于锁定状态；
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0；
- 如果整片擦除闪存，置位FMC_CTL寄存器的MER位；
- 通过将FMC_CTL寄存器的START位置1来发送整片擦除命令到FMC；
- 等待擦除指令执行完毕，FMC_STAT寄存器的BUSY位清0；
- 如果需要，读并验证是否擦除成功。

当整片擦除成功执行，FMC_STAT 寄存器的 ENDF 位置位。若 FMC_CTL 寄存器的 ENDIE 位被置 1，FMC 将触发一个中断。由于所有的闪存数据都将被复位为 0xFFFF_FFFF，可以通

过运行在 SRAM 中的程序或使用调试工具直接访问 FMC 寄存器来实现整片擦除操作。此外，如果任何闪存页处于擦除/编程保护下，整片擦除操作会被忽略。在这种情况下，如果 FMC_CTL 寄存器的 ERRIE 位被置位，该操作将触发操作出错中断。在中断服务程序中，软件可以通过检查 FMC_STAT 寄存器中的 WPERR 位来检测这种情况。

[图 2-2. 整片擦除操作流程](#)显示了整片擦除操作流程。

图 2-2. 整片擦除操作流程



2.3.7. 闪存块编程

FMC 提供了一个修改主存储闪存（BKSEL = 0）和数据闪存（BKSEL = 1）内容的 128 位四字编程功能。

编程操作，寄存器设置具体步骤如下：

- 确保FMC_CTL寄存器不处于锁定状态；
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0；

- 置位FMC_CTL寄存器的PG位;
- 写一个128位四字到目的绝对地址 (0x08XX XXXX);
- 等待编程指令执行完毕, FMC_STAT寄存器的BUSY位清0;
- 如果需要, 读并验证是否编程成功。

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

编程操作时需要检查目的地址是否已经被擦除。如果该地址没有被擦除, FMC_STAT 寄存器的 PGERR 位也将被置 1。每个四字在擦除后和下次擦除前只能编程一次。注意 PG 位必须在编程操作前被置位。

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

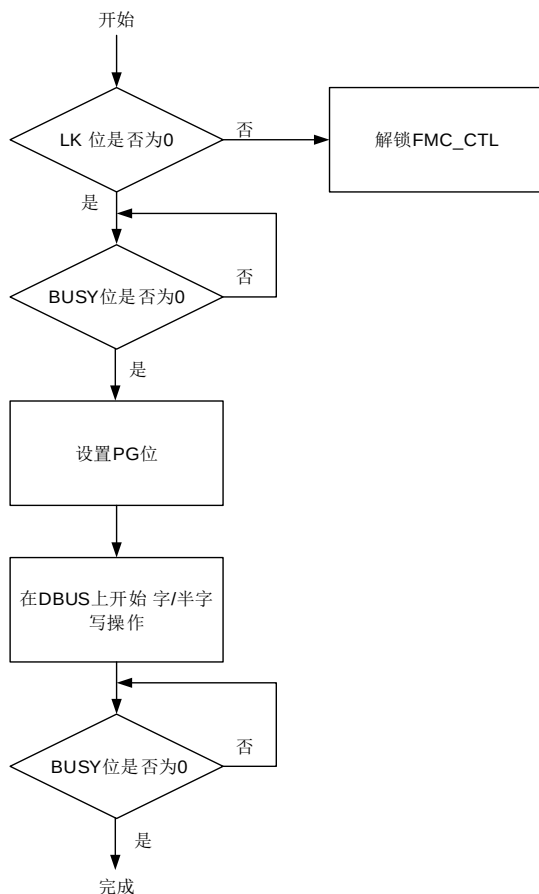
在下列情况下, FMC_STAT 寄存器中的 PGAERR 位会被置位。

- 按字节编程 (非32位或16位)
- 混合编程。不允许混合32位和16位编程
- 编程未对齐。

注意: 如果编程数据未能写满 128 位, 这些数据不会被编程入闪存, 并且不会有任何提示。

在这些情况下, 如果 FMC_CTL 寄存器的 ERRIE 位被置 1, FMC 将触发一次闪存操作错误中断。在中断服务程序中, 软件可以通过检查 FMC_STAT 寄存器中的 PGSERR / PGAERR / WPERR / PGERR 位来检测发生了哪种错误。下图显示了主存储块字编程操作流程。

图 2-3. 字编程操作流程



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

1.1.3. 闪存块快速编程

FMC提供了一个修改主存储闪存（BKSEL = 0）和数据闪存（BKSEL = 1）内容的快速编程功能。该模式可编程一行数据（16四字主闪存或4四字数据闪存），通过消除编程前对闪存单元的验证，从而缩短了页编程时间。同时针对每个字避免了高电压的上升和下降时间。

快速编程操作，寄存器设置具体步骤如下：

- 检查闪存的一行，确保为全FF；
- 确保FMC_CTL寄存器不处于锁定状态；
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0；
- 置位FMC_CTL寄存器的FSTPG位；
- 将待编程的绝对地址（0x08XX XXXX）写到FMC_ADDR寄存器；
- BUS将一行数据写入FMC_DATA。一行数据应该连续写入，否则，在设置FMC_CTL寄存器的START位之后，FMC_CTL寄存器中的PGERR位将被置位；
- 设置FMC_CTL寄存器的START位，启动快速编程操作；

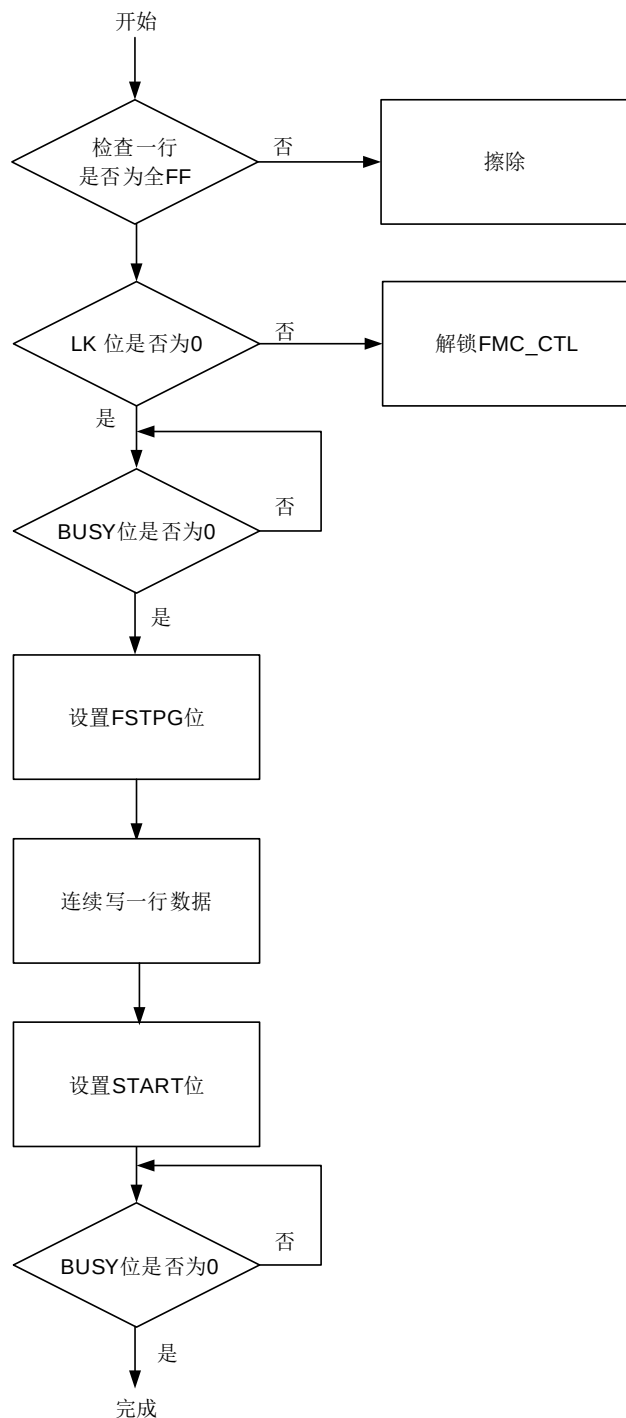
- 等待编程指令执行完毕，FMC_STAT寄存器的BUSY位清0；
- 如果需要，使用BUS读并验证是否编程成功。

当主存储块快速编程成功执行，FMC_STAT寄存器的ENDF位置位。若FMC_CTL寄存器的ENDIE位被置1，FMC将触发一个中断。在顺序快速编程中，如果确认该行已被擦除，则重复步骤6~8来快速编程下一行。

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

在这些情况下，如果FMC_CTL寄存器的ERRIE位被置1，FMC将触发一次闪存操作错误中断。在中断服务程序中，软件可以通过检查FMC_STAT寄存器中的PGSERR位、PGERR位，PGAERR位或者WPERR位来检测发生了哪种错误。[图2-4. 快速编程操作流程](#)显示了主存储块快速编程操作流程。

图 2-4. 快速编程操作流程

**注意:**

1. 一行数据必须连续写入。
2. 一行数据必须对齐。
3. 因为快速程序不会硬件检查闪存是否为0xFF，所以软件必须首先检查待编程区域是否为0xFF。在两次擦除操作之间，软件不能编程一行两次或两次以上。如果对一行进行两次或两次以上编程，可能会发生不可预测的结果。

4. 在设置FSTPG位和START位之间允许读操作，但读地址不能是要编程的行地址，否则可能读出的数据是旧数据。
5. 在快速编程之前必须刷新缓存。

2.3.8. 数据闪存操作

读操作

读数据闪存与主闪存方法一样。参考[读操作](#)。

编程操作

数据闪存编程与主闪存编程方法一样。参考[闪存块编程](#)和[闪存块快速编程](#)。

页擦除操作

数据闪存擦除与主闪存擦除方法一样。参考[页擦除](#)。

全片擦除操作

数据闪存整片擦除与主闪存整片擦除方法一样。参考[整片擦除](#)。

注意：

1. 通过配置 FMC_CTL 寄存器的 BKSEL 位，选择主闪存还是数据闪存进行页擦除操作、全片擦除操作和快速编程操作。
2. 编程数据闪存时，不得阻塞主闪存中正在运行的程序。不支持同时编程主闪存和数据闪存。

2.3.9. 查空命令

查空命令用于检查 FMC_ADDR 和 FMC_CTL 寄存器中 CBCMDLEN 指定的闪存区域是否都是 0xFF。通过配置 FMC_CTLx 寄存器的 CBCMD 位来配置查空命令，将 FMC_CTLx 寄存器中 START 位置 1 将发送查空命令，BUSY 位将置 1，硬件将检查闪存区域是否都是 0xFF。如果闪存区域全部为 0xFF，则设置 ENDF，并清除 BUSY 位，否则 CBCMDERR 位将置 1。仅主闪存及数据闪存支持查空命令。

注意：

1. 被检查的 flash 区域必须在一个页中，不超过 1KB 的边界。
2. 起始地址必须是四字对齐的。

2.3.10. OTP 编程

OTP 编程方法与主储存闪存编程相同，但 BKSEL 位需置位。OTP 块只能被编程一次并且不能被擦除。

注意：必须确保在 OTP 编程操作时不会发生任何意外中断，例如系统复位或掉电。如果发生

意外中断，闪存中的数据有很小可能性会出错。

2.3.11. 选项字节擦除

FMC 提供了一个擦除功能用来初始化闪存中的选项字节。选项字节擦除过程如下所示。

- 确保FMC_CTL寄存器不处于锁定状态；
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0；
- 解锁FMC_CTL寄存器的选项字节操作位；
- 等待FMC_CTL寄存器的OBWEN位置1；
- 置位FMC_CTL寄存器的BKSEL位；
- 置位FMC_CTL寄存器的OBER位；
- 通过将FMC_CTL寄存器的START位置1来发送选项字节擦除命令到FMC；
- 等待擦除指令执行完毕，FMC_STAT寄存器的BUSY位清0；
- 如果需要，读并验证是否擦除成功。

当选项字节擦除成功执行，FMC_STAT寄存器的ENDF位置位。若FMC_CTL寄存器的ENDIE位被置1，FMC将触发一个中断。

2.3.12. 选项字节编程

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

- 确保FMC_CTL寄存器不处于锁定状态；
- 检查FMC_STAT寄存器的BUSY位来判定闪存是否正处于擦写访问状态，若BUSY位为1，则需等待该操作结束，BUSY位变为0；
- 解锁FMC_CTL寄存器的选项字节操作位；
- 等待FMC_CTL寄存器的OBWEN位置1；
- 置位FMC_CTL寄存器的OBPG位；
- 写一个128位四字到目的地址；
- 等待编程指令执行完毕，FMC_STAT寄存器的BUSY位清0；
- 如果需要，读并验证是否编程成功。

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

选项字节修改后只有在系统复位或置位OBRD位后生效。

2.3.13. 选项字节说明

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

若选项字节和其补字节同为 0xFF，则 OBERR 位不置位。选项字节详情见下表。

表 2-3. 选项字节

地址	名称	出厂值	说明
0x1fff f800	SPC	0xA5	选项字节安全保护值 0xA5: 未保护状态 除0xA5和0xCC外的任何值: 保护级别低 0xCC: 保护级别高
0x1fff f801	SPC_N	0x5A	SPC补字节
0x1fff f802	USER	0xFF	[7:6]: LVD0T[1:0] (LVD0电压电平选择) 00: 保留 01: 保留 10: 2.90V/2.82V(上升/下降) 11: 4.31V/4.23V(上升/下降) 使用电压检测0电路时，禁止进行以上以外的设置。 [5]: FWDGSPD_STDBY 0: 待机模式时挂起独立看门狗 1: 待机模式时运行独立看门狗 [4]: FWDGSPD_DPSLP 0: 深度睡眠模式时挂起独立看门狗 1: 深度睡眠模式时运行独立看门狗 [3]: LVD0EN 0: 电压监控0复位后使能 1: 电压监控0复位复位后禁能 [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]: nFWDG_HW 0: 硬件使能独立看门狗功能 1: 软件使能独立看门狗功能
0x1fff f803	USER_N	0xFF	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

地址	名称	出厂值	说明
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 WP[22:0]: 每个bit可设置8KB闪存的保护状态。这些位总计可设置前184KB的闪存保护状态。 WP[23]: 第23位可设置闪存剩下部分的保护状态。
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 WP[31:24]: 每个bit可设置8KB闪存的保护状态。这些位总计可设置64KB的数据闪存保护状态。
0x1fff f80f	WP_N[31:24]	0xFF	WP补字节值的31到24位
0x1fff f810	USER_1	0xFF	[7:6]: nRST_MODE 00: GPIO模式, 仅内部RESET可能 01: NRST引脚配置为复位输入/输出模式 (传统内部RESET不传播到NSRT引脚) 10: NRST引脚配置为复位输入/输出模式 (传统内部RESET不传播到NSRT引脚) 11: 仅复位输入: NRST引脚上的低电平产生系统复位模式 [5:3]: 保留 [2]: SWDMOD 0: SWD关闭 1: SWD正常工作模式 [1]: SRAMECCEN 0: 关闭SRAM ECC 1: 使能SRAM ECC [0]: nWWDG_HW 0: 硬件使能窗口看门狗功能 1: 软件使能窗口看门狗功能
0x1fff f811	USER_1_N	0xFF	USER_1补字节
0x1fff f812	WWDGT_0	0xFF	[7]: WWDGT_WDGTEN 0: 禁能WWDGT产生复位的功能 1: 使能WWDGT产生复位的功能 [6]: WWDGT_EWIE 0: 关闭早唤醒中断 1: 开启早唤醒中断 [5:4]: WWDGT_WEPS 硬件启动模式下WWDGT窗口结束位置。 00: 75% 01: 50% 10: 25% 11: 0% (无窗口结束位置设置) [3:2]: WWDGT_WSPS

地址	名称	出厂值	说明
			在硬件启动模式WWDGT窗口启动位置。 00: 25% 01: 50% 10: 75% 11: 100% (无窗口起始位置设置) [1:0]: WWDGT_PSC 硬件启动模式下的WWDGT分频器。
0x1fff f813	WWDGT_0_N	0xFF	WWDGT_0补字节
0x1fff f814	WWDGT_1	0xFF	[7:0]: WWDGT_CNT [13:6] 硬件启动模式下窗口看门狗定时器计数器13 ~ 6位的值。
0x1fff f815	WWDGT_1_N	0xFF	WWDGT_1补字节
0x1fff f816	WWDGT_2	0xFF	[7:6]: 保留 [5:0]: WWDGT_CNT [5:0] 硬件启动模式下窗口看门狗定时器计数器5 ~ 0位的值。
0x1fff f817	WWDGT_2_N	0xFF	WWDGT_2补字节
0x1fff f818	FWDGT_0	0xFF	[7:0]: FWDGT_RLD [7:0] 在硬件启动模式下，独立看门狗定时器计数器的重新加载值的7到0位。
0x1fff f819	FWDGT_0_N	0xFF	FWDGT_0补字节
0x1fff f81a	FWDGT_1	0xFF	[7]: FWDGT_WDGTEN 0: 禁能FWDGT产生复位的功能 1: 使能FWDGT产生复位的功能 [6:4]: FWDGT_PSC [2:0] 独立看门狗定时器分频系数选择。 [3:0]: FWDGT_WND [3:0] 硬件启动模式下独立看门狗计数器窗口值的3到0位。
0x1fff f81b	FWDGT_1_N	0xFF	FWDGT_1补字节
0x1fff f81c	FWDGT_2	0xFF	[7:0]: FWDGT_WND [11:4] 硬件启动模式下独立看门狗计数器窗口值的11到4位。
0x1fff f81d	FWDGT_2_N	0xFF	FWDGT_2补字节
0x1fff f81e	FWDGT_3	0xFF	[3:0]: FWDGT_RLD [11:8] 在硬件启动模式下，独立看门狗定时器计数器的重新加载值的11到8位。
0x1fff f81f	FWDGT_3_N	0xFF	FWDGT_3补字节

注意：FWDGT 和 WWDGT 相关位域说明，详细可参考[看门狗定时器 \(WDGT\)](#)。

2.3.14. 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操作。当 FMC 对被保护页进行页擦除或编程操作时，操作本身无效且 FMC_STAT 寄存器的 WPERR 位将被置 1。如果 WPERR 位被置 1 且 FMC_CTL 寄存器的 ERRIE 位也被置 1 来使能相应的中断，FMC 将触发闪存操作出

错中断，等待 CPU 处理。配置选项字节的 WP[31:0]某位为 0 可以单独使能某几页的保护功能。如果在选项字节块执行了擦除操作，所有的闪存页擦除和编程保护功能都将失效。当选项字节的 WP 被改变时，需要系统复位使之生效。

表 2-4. OB_WP 位对应页保护

OB_WP位	页保护
OB_WP[0]	主闪存页0 ~ 7
OB_WP[1]	主闪存页8 ~ 15
OB_WP[2]	主闪存页16 ~ 23
.	.
.	.
.	.
OB_WP[22]	主闪存页176 ~ 183
OB_WP[23]	主闪存页184 ~ 256
OB_WP[24]	数据闪存页0 ~ 7
OB_WP[25]	数据闪存页8 ~ 15
OB_WP[26]	数据主闪存页16 ~ 23
.	.
.	.
.	.
OB_WP[30]	数据闪存页48 ~ 55
OB_WP[31]	数据闪存页56 ~ 63

2.3.15. 安全保护

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

未保护状态：当将 SPC 字节和它的补字节被设置为 0x5AA5，系统复位以后，闪存将处于非安全保护状态。主存储块和选项字节可以被所有操作模式访问。

保护等级低：当设置 SPC 字节值为任何除 0xA5 或 0xCC 外的值，系统复位以后，低安全保护状态生效。需要注意的是，若该修改过程中，MCU 的调试模块依然和外部 JTAG/SWD 设备相连，需要用上电复位代替系统复位以使得修改后的保护状态生效。在低安全保护状态下，主存储闪存块仅能被用户代码访问且前 8KB 的闪存自动处于页擦除/编程保护状态下。在调试模式下，或从 SRAM 中启动时，以及从 boot loader 区启动时，这些模式下对主存储块的操作都被禁止。如果在这些模式下读主存储块，将产生总线错误。如果在这些模式下，对主存储块进行编程或擦除操作，FMC_STAT 寄存器的 WPERR 位将被置 1。但这些模式下都可以对选项字节进行操作，从而可以通过该方式失能安全保护功能。如果将 SPC 字节和它的补字节设置为 0x5AA5，安全保护功能将失效，并自动触发一次整片擦除操作。

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 boot loader 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。

2.4. FMC 寄存器

FMC 基地址：0x4002 2000

2.4.1. 等待状态寄存器（FMC_WS）

地址偏移：0x00

复位值：0x0000 0611

该寄存器只能按字（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			DCRST	ICRST	DCEN	ICEN	保留				PFEN	保留	WSCNT[2:0]		
			rw	rw	rw	rw					rw		rw		

位/位域	名称	描述
31:13	保留	必须保持复位值。
12	DCRST	复位 D-cache 高速缓存区。该位仅可在 DCEN 位置 0 时可写。 0: 无效果 1: 复位 D-cache 高速缓存区
11	ICRST	复位 I-cache 该位仅可在 ICEN 位置 0 时可写。 0: 无效果 1: 复位 I-cache 高速缓存区
10	DCEN	D-cache 高速缓存区使能位 0: 失能 D-cache 高速缓存区 1: 使能 D-cache 高速缓存区
9	ICEN	I-cache 高速缓存区使能位 0: 失能 I-cache 高速缓存区 1: 使能 I-cache 高速缓存区
8:5	保留	必须保持复位值。
4	PFEN	预取功能使能位 0: 失能预取功能 1: 使能预取功能
3	保留	必须保持复位值。
2:0	WSCNT[2:0]	等待状态计数寄存器 软件置 1 和清 0。 000: 不增加等待状态

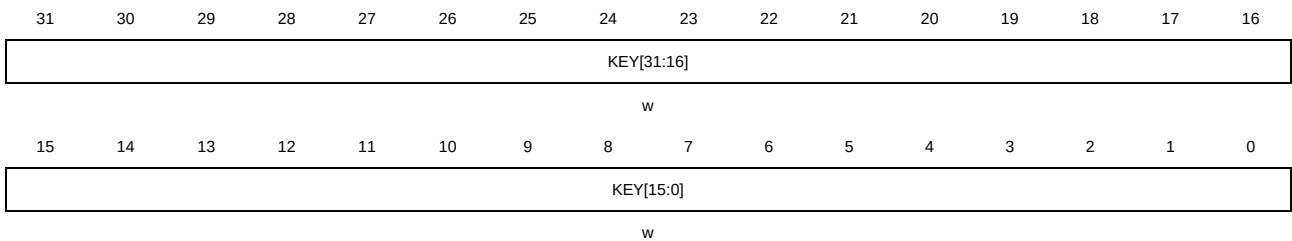
001: 增加 1 个等待状态
010: 增加 2 个等待状态
011: 增加 3 个等待状态
100: 增加 4 个等待状态
101: 增加 5 个等待状态
110: 增加 6 个等待状态
111: 保留

2.4.2. 解锁寄存器 (FMC_KEY)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



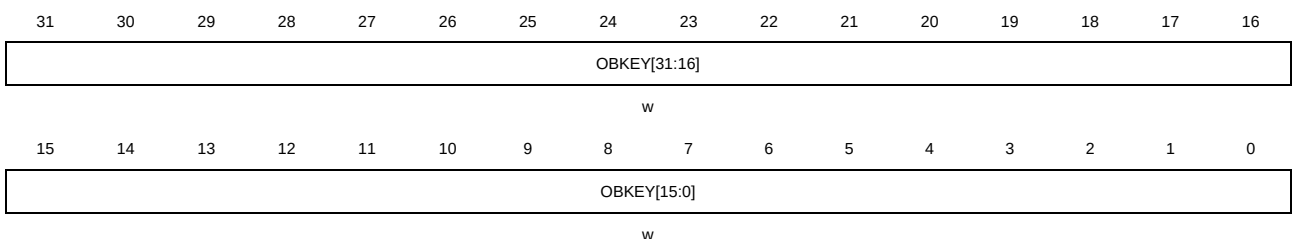
位/位域	名称	描述
31:0	KEY[31:0]	FMC_CTL 解锁寄存器 这些位仅能被软件写。 写解锁值到KEY[31:0]可以解锁 FMC_CTL寄存器。

2.4.3. 选项字节操作解锁寄存器 (FMC_OBKEY)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



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

2.4.4. 状态寄存器（FMC_STAT）

地址偏移：0x0C

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



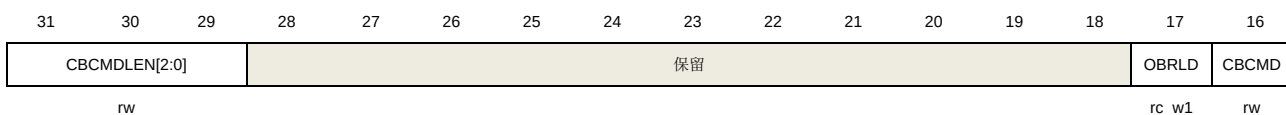
位/位域	名称	描述
31:7	保留	必须保持复位值。
6	CBCMDERR	查空命令检查的区域是否全为0xFF。 0: 检查的区域全为0xFF。 1: 检查的区域不是全为0xFF。
5	ENDF	操作结束标志位 操作成功执行后，此位被硬件置1。软件写1清0。
4	WPERR	擦除/编程保护错误标志位 在受保护的页上擦除/编程操作时，此位被硬件置1。软件写1清0。
3	PGAERR	编程对齐错误标志位 当写数据不对齐时，此位被硬件置1。软件写1清0。
2	PGERR	编程错误标志位 当被编程区域状态不为0xFFFF时，对闪存编程，此位被硬件置1。软件写1清0。
1	PGSERR	编程序列错误标志位 当编程顺序错误时，此位被硬件置1。软件写1清0。
0	BUSY	闪存忙标志 当闪存操作正在进行时，此位被置1。当操作结束或者出错，此位被清0。

2.4.5. 控制寄存器（FMC_CTL）

地址偏移：0x10

复位值：0x0000 0080

该寄存器只能按字（32 位）访问。



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

位/位域	名称	描述
31:29	CBCMDLEN[2:0]	CBCMD读数据长度2 ⁿ （CBCMDLEN）。 查空命令读数据长度。 读数据长度为 2 ⁿ CBCMDLEN 个四字。
28:18	保留	必须保持复位值。
17	OBRDL	选项字节重加载位 软件置位。 0: 无作用 1: 选项字节强制重加载并产生一次系统复位 注意: 如果 OBLK 置位, 该位则无法写入。
16	CBCMD	查询选中区域是否为空命令。
15	BKSEL	选择操作的闪存块 当闪存繁忙时, 不能修改此位。 0: MER/PER/FSTPG/PG 用于主闪存 1: MER/PER/FSTPG/PG 用于数据闪存
14:13	保留	必须保持复位值。
12	ENDIE	操作结束中断使能位 软件置 1 和清 0 0: 无硬件中断产生 1: 使能操作结束中断
11	保留	必须保持复位值。
10	ERRIE	出错中断使能位 软件置 1 和清 0 0: 无硬件中断产生. 1: 使能出错中断
9	OBWEN	选项字节擦除/编程使能位 当正确的序列写入 FMC_OBKEY 寄存器, 此位由硬件置 1。此位可以被软件清 0。
8	保留	必须保持复位值。
7	LK	FMC_CTL0 寄存器锁定标志位 当正确的序列写入 FMC_KEY0 寄存器, 此位由硬件清 0。此位可以由软件置 1。
6	START	向 FMC 发送擦除命令位 软件置 1 可以发送擦除命令到 FMC。当 BUSY 位被清 0 时, 此位由硬件清 0。
5	OBER	选项字节擦除命令位

		软件置 1 和清 0
		0: 无作用
		1: 选项字节擦除命令
4	OBPG	选项字节编程命令位
		软件置 1 和清 0
		0: 无作用
		1: 选项字节编程命令
3	保留	必须保持复位值
2	MER	主存储块整片擦除命令位
		软件置 1 和清 0
		0: 无作用
		1: 主存储块整片擦除命令
1	PER	主存储块页擦除命令位
		软件置 1 和清 0
		0: 无作用
		1: 主存储块页擦除命令
0	PG	主存储块编程命令位
		软件置 1 和清 0
		0: 无作用
		1: 主存储块编程命令

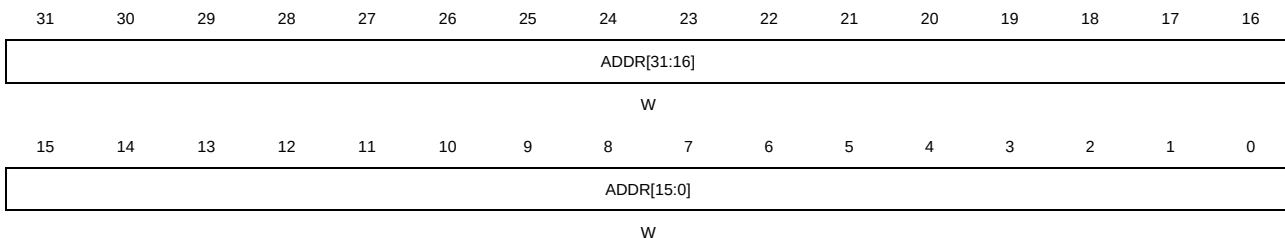
注意：当相应闪存操作完成后，该寄存器需处于复位状态。

2.4.6. 地址寄存器（FMC_ADDR）

地址偏移：0x14

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



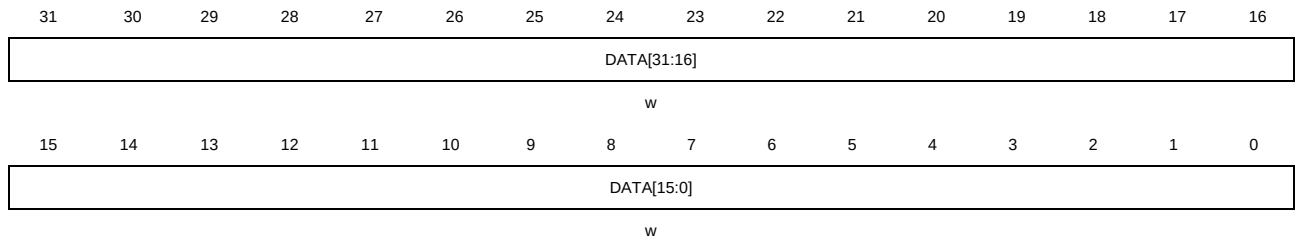
位/位域	名称	描述
31:0	ADDR[31:0]	闪存擦除或编程地址 该位通过软件设置。 ADDR 位是闪存擦除/编程命令的地址

2.4.7. 编程数据寄存器 (FMC_DATA)

地址偏移: 0x18

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



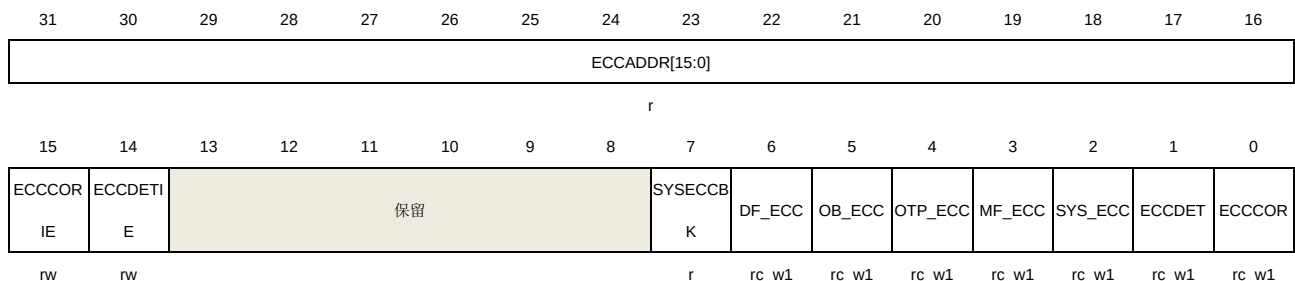
位/位域	名称	描述
31:0	DATA[31:0]	闪存擦除/编程命令数据位 这些位是由软件配置的。这是用在主闪存快速编程。

2.4.8. ECC 控制及状态寄存器 (FMC_ECCCS)

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	ECCADDR[15:0]	检测到 ECC 错误的四字的偏移地址。 错误地址 = 基地址 + ECCADDR[15:0] * 8, 基地址可以是主闪存, 数据闪存, system 区, 选项字节以及 OTP 的起始地址。 主闪存: 0 ~ 0x7FFF (256K/8 - 1) 数据闪存: 0 ~ 0x1FFF (64K/8 - 1) 选项字节: 0 ~ 0x01 (16/8 - 1) OTP: 0 ~ 0x3F (512/8 - 1) 系统存储区域: 0 ~ 0x6FF (14K/8 - 1)
15	ECCCORIE	纠正单个位中断使能 0: 纠正单个位中断失能。 1: 纠正单个位中断使能。

14	ECCDETIE	检测到双位错误中断使能。 0: 检测到双位错误中断失能 1: 检测到双位错误中断使能
13:8	保留	必须保持复位值。
7	SYSECCBK	该位只用于指示 ECC 错误是系统存储区的前 3k 或后 3k 区域。 0: 系统存储区的前 3K 区域发生 ECC 错误。 1: 系统存储区的后 3K 区域发生 ECC 错误。
6	DF_ECC	如果在数据闪存中检测到 ECC 错误, 该位置 1。ECCADDR 会记录出错的数据闪存偏移地址。ECCDET 位和 ECCCOR 位表示是可纠正的单个位错误还是不可纠正的双位错误。该位写 1 清零。 0: 在数据闪存中未检测到 ECC 错误。 1: 在数据闪存中检测到 ECC 错误。
5	OB_ECC	如果在读取选项字节时检测到 ECC 错误, 该位置 1。ECCADDR 会记录出错的选项字节偏移地址。ECCDET 位和 ECCCOR 位表示是可纠正的单个位错误还是不可纠正的双位错误。该位写 1 清零。 0: 在读取选项字节时未检测到 ECC 错误 1: 在读取选项字节时检测到 ECC 错误
4	OTP_ECC	如果在 OTP 中检测到 ECC 错误, 该位置 1。ECCADDR 会记录出错 OTP 偏移地址。ECCDET 位和 ECCCOR 位表示是可纠正的单个位错误还是不可纠正的双位错误。该位写 1 清零。 0: 在 OTP 中未检测到 ECC 错误。 1: 在 OTP 中检测到 ECC 错误。
3	MF_ECC	如果在主闪存中检测到 ECC 错误, 该位置 1。ECCADDR 会记录出错的主闪存偏移地址。ECCDET 位和 ECCCOR 位表示是可纠正的单个位错误还是不可纠正的双位错误。该位写 1 清零。 0: 在主闪存中未检测到 ECC 错误。 1: 在主闪存中检测到 ECC 错误。
2	SYS_ECC	如果在系统存储中检测到 ECC 错误, 该位置 1。ECCADDR 会记录出错的系统存储偏移地址。ECCDET 位和 ECCCOR 位表示是可纠正的单个位错误还是不可纠正的双位错误。该位写 1 清零。 0: 在系统存储中未检测到 ECC 错误。 1: 在系统存储中检测到 ECC 错误。
1	ECCDET	检测到双位错误标志。当检测到双位错误时, 该位置 1。 该位写 1 清零。 0: 未检测到 ECC 双位错误。 1: 检测到 ECC 双位错误。
0	ECCCOR	检测并纠正单个位错误标志 该位写 1 清零。 0: 未检测并纠正 ECC 单个位错误。

1: 检测并纠正 ECC 单个位错误。

2.4.9. 选项字节状态寄存器 (FMC_OBSTAT)

地址偏移: 0x38

复位值: 0x0XXX XXXX

该寄存器只能按字 (32 位) 访问。



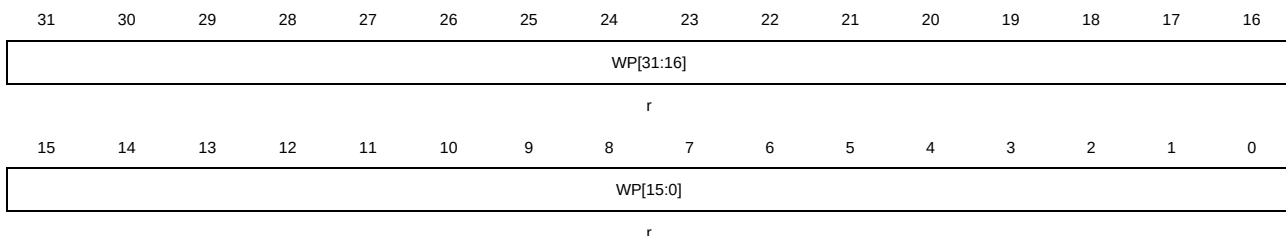
位/位域	名称	描述
31:26	保留	必须保持复位值。
25:10	DATA[15:0]	系统复位后保存选项字节的 DATA[15:0]部分
9:2	USER[7:0]	系统复位后保存选项字节块的 USER 字节
1	SPC	安全保护状态 0: 未保护 1: 已保护
0	OBERR	选项字节读错误位 当选项字节和它的补字节不匹配时此位由硬件置 1, 选项字节被强制设置为 0xFF。

2.4.10. 擦除/编程保护寄存器 (FMC_WP)

地址偏移: 0x3C

复位值: 0xFFFF XXXX

该寄存器只能按字 (32 位) 访问。



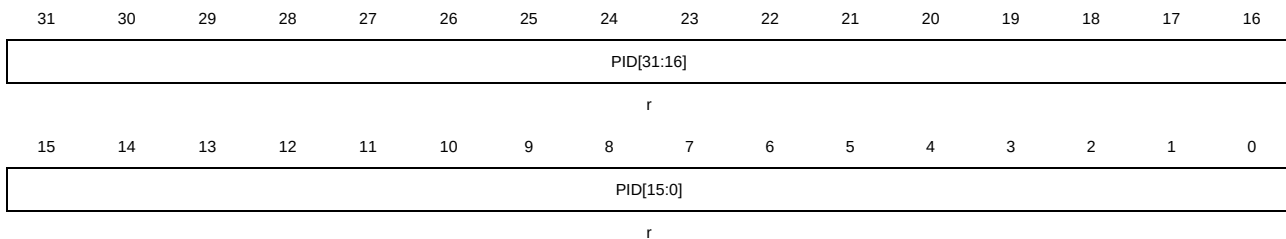
位/位域	名称	描述
31:0	WP[31:0]	系统复位后保存选项字节块的 WP[31:0]部分

2.4.11. 产品 ID 寄存器 0 (FMC_PID0)

地址偏移: 0x100

复位值: 0xFFFF XXXX

该寄存器只能按字 (32 位) 访问。



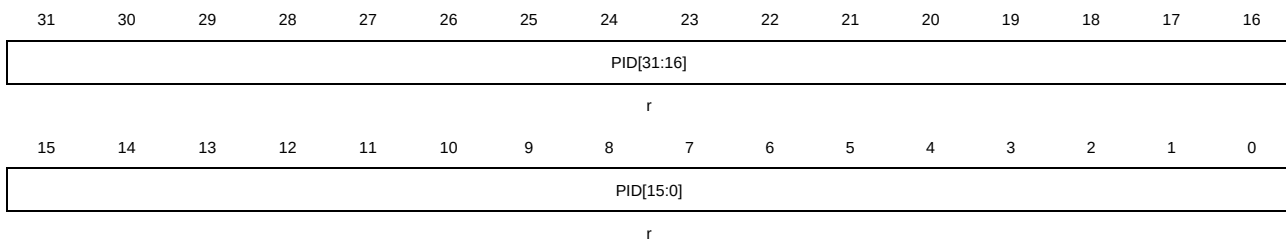
位/位域	名称	描述
31:0	PID[31:0]	产品保留 ID 该寄存器为只读 上电后这些位始终不会改变，该寄存器在生产过程中被一次性编程。

2.4.12. 产品 ID 寄存器 1 (FMC_PID1)

地址偏移: 0x104

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 该寄存器为只读 上电后这些位始终不会改变，该寄存器在生产过程中被一次性编程。

3. 电源管理单元（PMU）

3.1. 简介

功耗设计是GD32M531系列产品最注重的问题之一。电源管理单元提供了三种省电模式，包括睡眠模式、深度睡眠模式和待机模式。这些模式能减少电源能耗，且使得应用程序可以在CPU运行时间要求、速度和功耗的相互冲突中获得最佳折衷。如 [图3-1. 电源域概览](#)所示，GD32M531系列设备有两个电源域，包括V_{DD} / V_{DDA}域和V_{CORE}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V_{CORE}域供电。

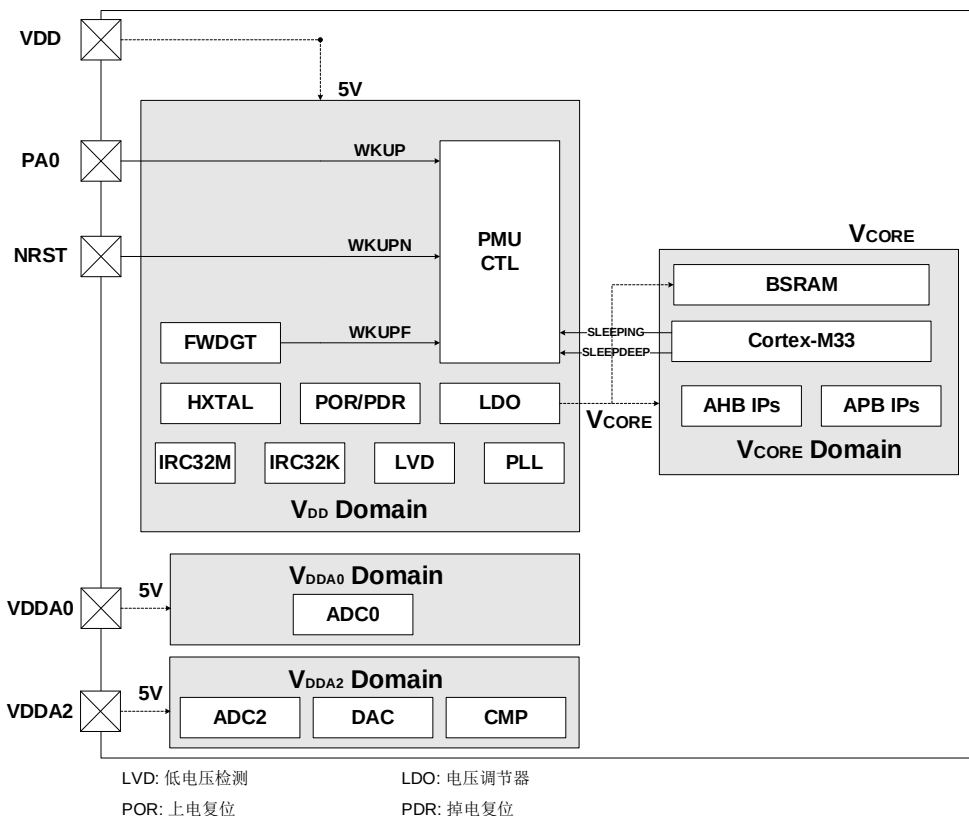
3.2. 主要特征

- 两个电源域：V_{DD} / V_{DDA}域和V_{CORE}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V_{CORE}电源；
- 提供三个低电压检测器（LVD）检测低电压事件，当电压低于或高于所设定的阈值时LVD1和LVD2能产生中断，LVD0、LVD1和LVD2能产生系统复位。

功能说明

图3-1. 电源域概览提供了PMU及相关电源域的内部结构框图。

图3-1. 电源域概览



3.3.1. V_{DD} / V_{DDA} 电源域

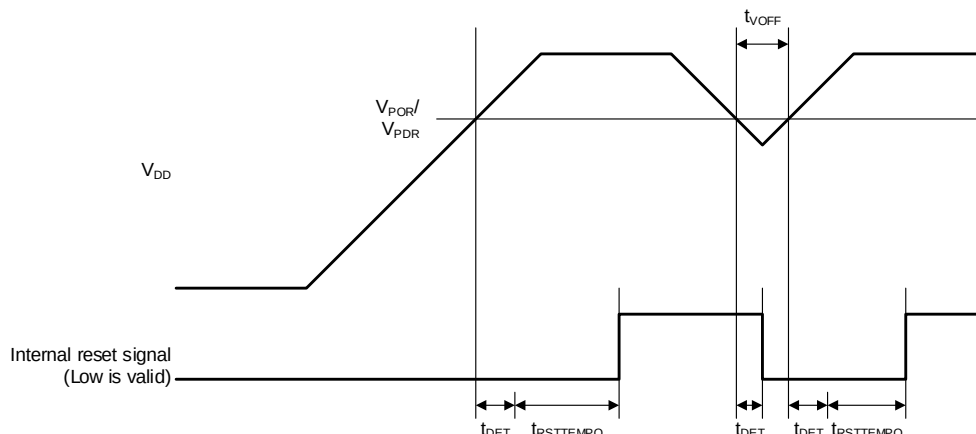
V_{DD} / V_{DDA} 域包括 V_{DD} 域和 V_{DDA} 域两部分。V_{DD} 域包括 HXTAL（高速外部晶体振荡器）、LDO（电压调节器）、POR / PDR（上电 / 掉电复位）、FWDGT（独立看门狗定时器）和除 PD4 / D5、PC0-PC12 之外的所有 PAD 以及 IRC32M（内部 32M RC 振荡器）、IRC32K（内部 32KHz RC 振荡器）、PLLs（锁相环）和 LVD（低电压检测器）等等。V_{DDA} 域包括 V_{DDA0} 和 V_{DDA2} 两部分，其中 V_{DDA0} 包括 ADC0，V_{DDA2} 包括 ADC2，DAC（AD / DA 转换器）、CMP 等等。

V_{DD} 域

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

POR / PDR（上电 / 掉电复位）电路检测 V_{DD} / V_{DD} 并在电压低于特定阈值时产生电源复位信号复位整个芯片。[图3-2. 上电 / 掉电复位波形图](#)显示了供电电压和电源复位信号之间的关系。 V_{POR} 表示上电复位的阈值电压， V_{PDR} 表示掉电复位的阈值电压， $V_{PDRhyst}$ 表示迟滞电压值，以上参数值请参见数据手册。

图3-2. 上电 / 掉电复位波形图



(1) 可以通过监测 GPIO 标志位的数值来近似判断内部复位信号的状态。通过测量 V_{POR} 与 GPIO 标志位由低跳变为高之间的时间差，可以对 t_{DET} 与 $t_{RSTTEMPO}$ 的总和进行大致验证。

(2) NRST 引脚上的电压并不严格等同于内部复位信号的状态。

V_{DDA} 域

一般来说，数字电路由 V_{DD} 供电，而大多数的模拟电路由 V_{DDA} 供电。为了提高 ADC 和 DAC 的转换精度，为 V_{DDA} 独立供电可使模拟电路达到更好的特性。为避免噪声， V_{DDA} 通过外部滤波电路连接至 V_{DD} ，相应的 V_{SSA} 通过特定电路连接至 V_{SS} 。否则，当 V_{DD} 和 V_{DDA} 不是同一个电源提供时，在上电和运行过程中 V_{DD} 与 V_{DDA} 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 V_{DDA} 引脚，或者外部参考电压，外部参考电压的范围请参考 [表 15-2. ADC 输入引脚定义](#) 和 [表 16-1. DAC 引脚](#)，VREFN 须被连接至 V_{SSA} 引脚。

3.3.2. V_{CORE} 电源域

V_{CORE} 电源域为 Cortex®-M33 内核逻辑、AHB / APB 外设和 V_{DD} / V_{DDA} 域的 APB 接口等供电。当 V_{CORE} 电压上电后，POR 将在 V_{CORE} 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。 V_{CORE} 电压值请参考数据手册。

3.3.3. 省电模式

系统复位或电源复位后，GD32M531 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两种：减慢系统时钟（HCLK, PCLK1, PCLK2），关闭未使用的外设的时钟。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，V_{CORE} 域中的所有时钟全部关闭，IRC8M、IRC48M、HXTAL 及 PLLs 也全部被禁用。SRAM 和寄存器中的内容被保留。根据 PMU_CTL 寄存器的 LDOLP 位的配置，可控制 LDO 工作在正常模式或低功耗模式。进入深度睡眠模式之前，先将 Cortex®-M33 系统控制寄存器的 SLEEPDEEP 位置 1，再清除 PMU_CTL 寄存器的 STBMOD 位，然后执行 WFI 或 WFE 指令即可进入深度睡眠模式。如果睡眠模式是通过执行 WFI 指令进入的，任何来自 EXTI 的中断可以将系统从深度睡眠模式中唤醒。如果睡眠模式是通过执行 WFE 指令进入的，任何来自 EXTI 的事件可以将系统从深度睡眠模式中唤醒（如果 SEVONPEND 为 1，任何来自 EXTI 的中断都可以唤醒系统，请参考 Cortex®-M33 技术手册）。刚退出深度睡眠模式时，IRC32M 被选中作为系统时钟。请注意，如果 LDO 工作在低功耗模式，那么唤醒时需额外的延时时间。

注意：为了顺利进入深度睡眠模式，所有 EXTI 线上的挂起状态（在 EXTI_PD 寄存器中）和相关外设标志位必须被复位，参考[表 6-3. EXTI 触发源](#)。否则，程序将直接跳过深度睡眠模式进入过程而继续执行下面的程序。

待机模式

待机模式也是基于 Cortex®-M33 的 SLEEPDEEP 模式实现的。在待机模式下，整个 V_{CORE} 域全部停止供电，同时 LDO 和包括 IRC32M、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_CTL 寄存器的 STBMOD 位置 1，再清除 PMU_CS 寄存器的 WUF 位，然后执行 WFI 或 WFE 指令，系统进入待机模式，PMU_CS 寄存器的 STBF 位状态表示 MCU 是否已进入待机模式。待机模式有三个唤醒源，包括来自 NRST 引脚的外部复位，FWDGT 复位和 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 V_{CORE} 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 0x0000 0000 地址开始执行指令代码。

表3-1. 节电模式总结

模式	睡眠	深度睡眠	待机
描述	仅关闭 CPU 时钟	1. 关闭 V _{CORE} 电源域的所有时钟	1. 关闭 V _{CORE} 电源域的供电

模式	睡眠	深度睡眠	待机
		2. 关闭 IRC32M、HXTAL 和 PLL	2. 关闭 IRC32M、HXTAL 和 PLL
LDO 状态	开启（正常功耗模式）	开启（正常功耗模式或者低功耗模式）	关闭
配置	SLEEPDEEP = 0	SLEEPDEEP = 1 STBMOD = 0	SLEEPDEEP = 1 STBMOD = 1, WURST=1
进入指令	WFI 或 WFE	WFI 或 WFE	WFI 或 WFE
唤醒	若通过 WFI 进入，则任何中断均可唤醒； 若通过 WFE 进入，则任何事件（或 SEVONPEND=1 时的中断）均可唤醒	若通过 WFI 进入，来自 EXTI 的任何中断可唤醒； 若通过 WFE 进入，来自 EXTI 的任何事件（或 SEVONPEND=1 时的中断）可唤醒	1. NRST 引脚 2. WKUP 引脚 3. FWDGT 复位
唤醒延迟	无	IRC32M 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 引脚和使能的 WKUP 引脚，其他所有 I/O 都处于高阻态。

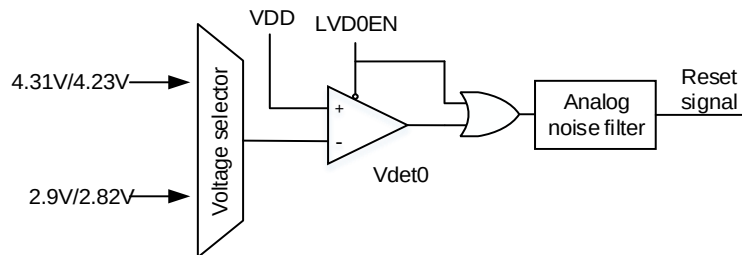
3.3.4. 低压差电压检测器

三个低压检测器（LVD）用于检测低电压事件。

LVD0

LVD0 用于检测 VDD 供电电压是否低于选项字节中的 LVD0T[7:6]位所选择的设定阈值。如果选项字节中的 LVD0EN 设置为 0，当 LVD0 检测到电压低于设定阈值时，将产生系统复位。

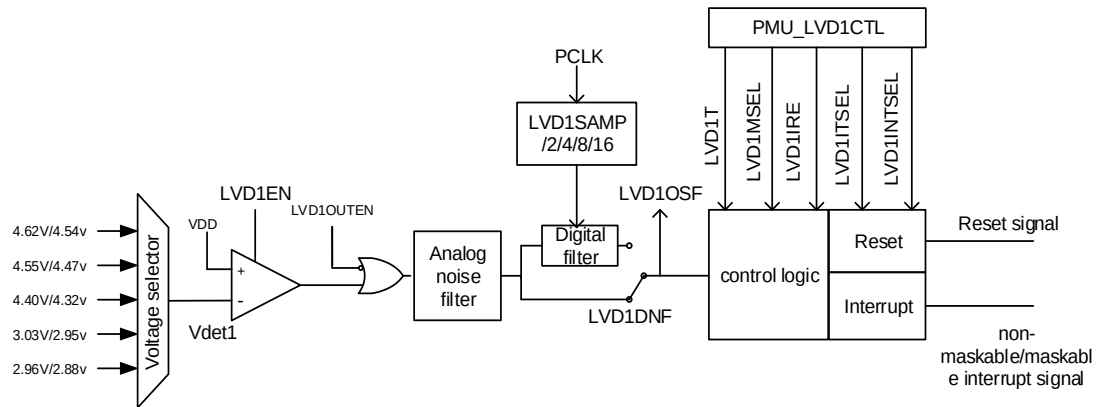
图3-3. 低电压检测器0框图



LVD1

低电压检测器 1（LVD1）用于检测 VDD / VDDA 供电电压是否低于或高于由 PMU_LVD1CTL 中 LVD1T[3:0]位配置的检测阈值。当 LVD1 被启用时，如果检测到 $VDD / VDDA > Vdet1$ 或 $VDD / VDDA \leq Vdet1$ ，LVD1 可以产生系统复位或中断。

图3-4. 低电压检测器1框图



通过以下寄存器设置序列，LVD1 可以产生中断。

1. 通过 SYSCFG_PRCFG 中的 PRCFG2 位解锁 PMU_LVD1CTL。
2. 设置 LVD1T[3:0]位选择 LVD1 的检测电压阈值。
3. 设置 LVD1MSEL 位选择 LVD1 的中断模式。
4. 配置 LVD1INTSEL[1:0]位设置生成 LVD1 中断的条件。
5. 配置 LVD1ITSEL 位设置中断类型。
6. 设置 LVD1EN 位启用 LVD1。
7. 设置 LVD1OUTEN 位来启用 LVD1 比较结果输出。
8. 将 LVD1DETF 位设置为 0。
9. 设置 LVD1IRE 位来启用 LVD1 中断。

注意：在以下两种情况下，不需要执行步骤 1.2.5：

- a. 在停止监控后，监控会在不更改监控设置的情况下重新启动。
- b. 停止监控后，更改 LVD1ITSEL 和 LVD1INTSEL 监控将重新启动。

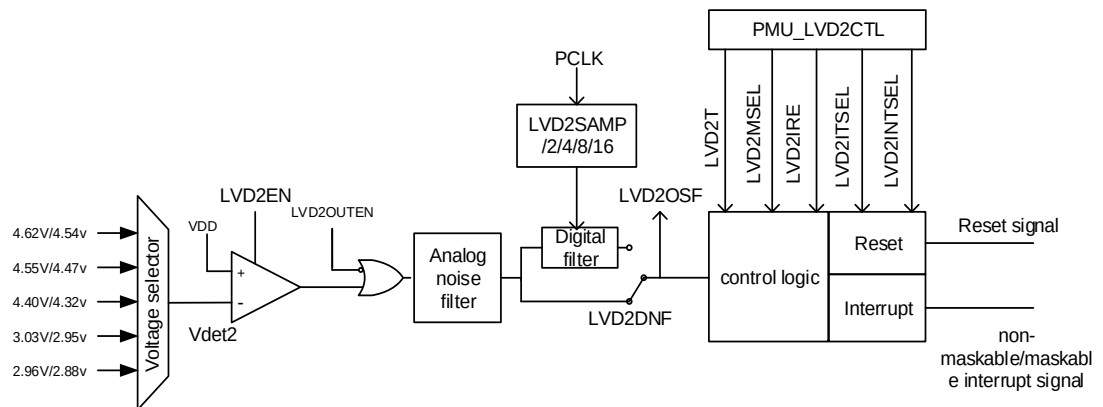
通过以下步骤的寄存器设置序列，LVD1 可以产生系统复位。

1. 设置 LVD1T[3:0]位来选择 LVD1 的检测电压阈值。
2. 设置 LVD1MSEL 位选择 LVD1 的复位模式。
3. 设置 LVD1RSTSEL 位来设置 LVD1 复位信号产生的时长。
4. 设置 LVD1IRE 位来启用 LVD1 复位模式。
5. 设置 LVD1EN 位来启用 LVD1。
6. 设置 LVD1OUTEN 位来启用 LVD1 比较结果输出。

LVD2

低压检测器 2 (LVD2) 用于检测 VDD/VDDA 供电电压是否低于或高于由 PMU_LVD2CTL 中的 LVD2T[3:0]位选择的检测阈值。当 LVD2 启用时，如果检测到 $VDD/VDDA > Vdet2$ 或 $VDD/VDDA \leq Vdet2$ ，LVD2 可以产生系统复位或中断。

图3-5. 低电压检测器2框图



通过以下寄存器设置序列，LVD2 可以产生中断。

1. 通过 SYSCFG_PRCFG 中 PRCFG2 位来解锁 PMU_LVD2CTL。
2. 设置 LVD2T[1:0] 位来选择 LVD2 的检测电压阈值。
3. 设置 LVD2MSEL 位来选择 LVD2 的中断模式。
4. 配置 LVD2INTSEL[1:0] 位来设置生成 LVD2 中断的条件。
5. 配置 LVD2ITSEL 位来设置中断类型。
6. 设置 LVD2EN 位来启用 LVD2。
7. 设置 LVD2OUTEN 位来启用 LVD2 比较结果输出。
8. 将 LVD2DETF 位设置为 0。
9. 设置 LVD2IRE 位来启用 LVD2 中断。

注意：以下两种情况下不需要执行步骤 1.2.5:

- a. 在停止监控后，不更改监控设置就重新启动监控。
- b. 监控停止后，更改 LVD2ITSEL 和 LVD2INTSEL 监控将重新启动。

通过以下步骤寄存器设置序列，LVD2 可以生成系统重置。

1. 设置 LVD2T[1:0] 位来选择 LVD2 的检测电压阈值。
2. 设置 LVD2MSEL 位来选择 LVD2 的复位模式。
3. 设置 LVD2RSTSEL 位来设置 LVD2 复位信号产生的时长。
4. 设置 LVD2IRE 位来启用 LVD2 复位模式。
5. 设置 LVD2EN 位来启用 LVD2。
6. 设置 LVD2OUTEN 位来启用 LVD2 比较结果输出。

3.4. PMU 寄存器

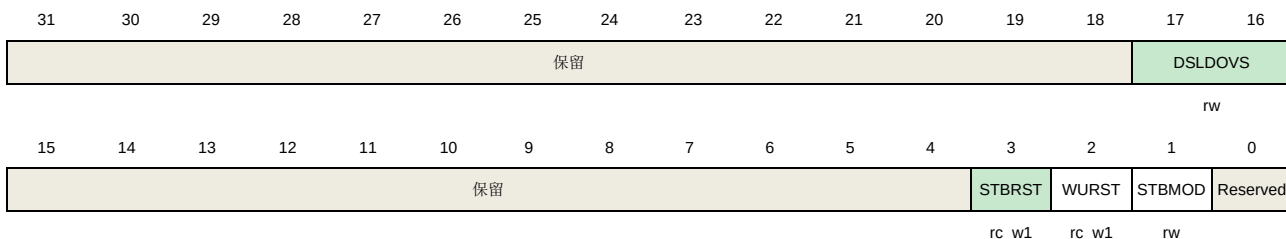
PMU基地址：0x4000 7000

3.4.1. 控制寄存器（PMU_CTL）

地址偏移：0x00

复位值：0x0001 0000 （从待机模式唤醒后复位）

该寄存器可以按字（32位）访问。



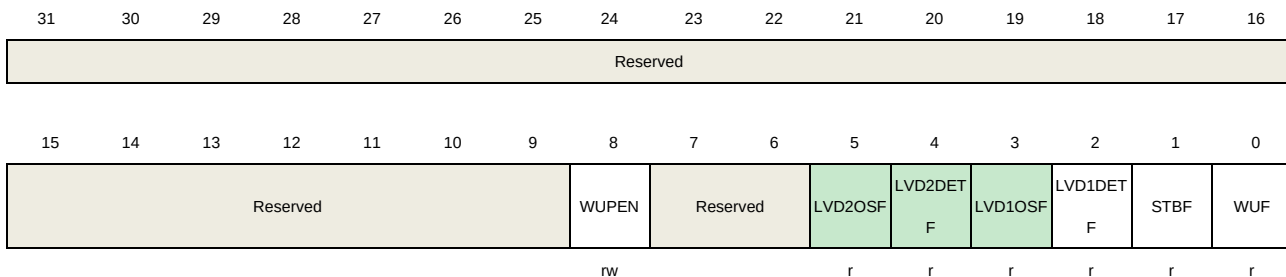
位/位域	名称	描述
31:16	保留	必须保持复位值。
17:16	DSLDOVS[1:0]	深度睡眠模式下LDO输出 00: LDO输出0.9V 01: LDO输出1.0V 10: LDO输出1.1V 11: LDO输出1.2V
15:4	保留	必须保持复位值。
3	STBRST	待机标志复位 0: 无影响 1: 复位待机标志 读该位，始终返回0
2	WURST	唤醒标志复位 0: 无影响 1: 复位唤醒标志 读该位，始终返回0
1	STBMOD	待机模式 0: 当Cortex®-M33进入SLEEPDEEP模式时，系统进入深度睡眠模式 1: 当Cortex®-M33进入SLEEPDEEP模式时，系统进入待机模式
0	保留	必须保持复位值。

3.4.2. 电源控制和状态寄存器 (PMU_CS)

地址偏移: 0x04

复位值: 0x0000 0028 (从待机模式唤醒后不复位)

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	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。
7:6	保留	必须保持复位值
5	LVD2OSF	低电压检测器 2 输出信号标志 0: $VCC < V_{det2}$ 1: $VCC \geq V_{det2}$ 或 $LVD2EN = 0$
4	LVD2DETF	低电压检测器2状态标志 0: 未发生检测事件 1: 发生检测事件
3	LVD1OSF	低电压检测器 1 输出信号标志 0: $VCC < V_{det1}$ 1: $VCC \geq V_{det1}$ 或 $LVD1EN = 0$
2	LVD1DETF	低电压检测器1状态标志 0: 未发生检测事件 1: 发生检测事件
1	STBF	待机标志 0: 设备没进入过待机模式 1: 设备曾进入过待机模式 该位只能由POR/PDR或通过设置PMU_CTL寄存器的STBRST位来清零。
0	WUF	唤醒标志

0: 没有收到唤醒事件

1: 收到来自WKUP引脚

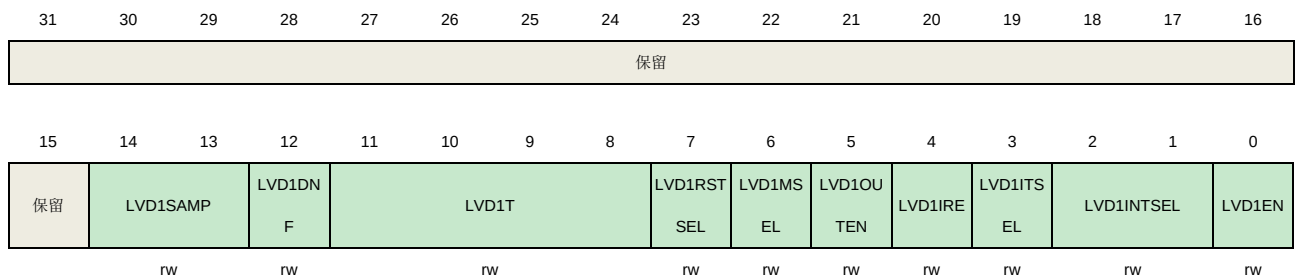
该位只能由POR / PDR或通过设置PMU_CTL寄存器的WURST位来清零。

3.4.3. 低电压检测器 1 控制寄存器 (PMU_LVD1CTL)

地址偏移: 0x08

复位值: 0x0000 1682 (从待机模式唤醒后不复位)

该寄存器可以按字 (32位) 访问。



位/位域	名称	描述
31:15	保留	必须保持复位值。
14:13	LVD1SAMP	采样时钟选择 00: PCLK1 频率的 1 / 2 01: PCLK1 频率的 1 / 4 10: PCLK1 频率的 1 / 8 11: PCLK1 频率的 1 / 16
12	LVD1DNF	LVD1 数字噪声滤波器使能 0: 数字噪声滤波器使能 1: 数字噪声滤波器失能
11:8	LVD1T[3:0]	LVD1 检测阈值 0100: 4.62V / 4.54V (上升 / 下降) 0101: 4.55V / 4.47V (上升 / 下降) 0110: 4.40V / 4.32V (上升 / 下降) 1010: 3.03V / 2.95V (上升 / 下降) 1011: 2.96V / 2.88V (上升 / 下降) 1100-1111: 保留
7	LVD1RSTSEL	LVD1 复位信号解除选择: 0: 在检测到 VCC > Vdet1 后, 复位信号才会在稳定时间 (t _{LVD1}) 后解除 (在 VCC <= Vdet1 时 MCU 一直处于复位状态)。 1: LVD1 复位信号产生后, 在经过一个稳定时间 (t _{LVD1}) 后, 复位信号将解除 (MCU 产生复位一次)。
6	LVD1MSEL	LVD1 模式选择:

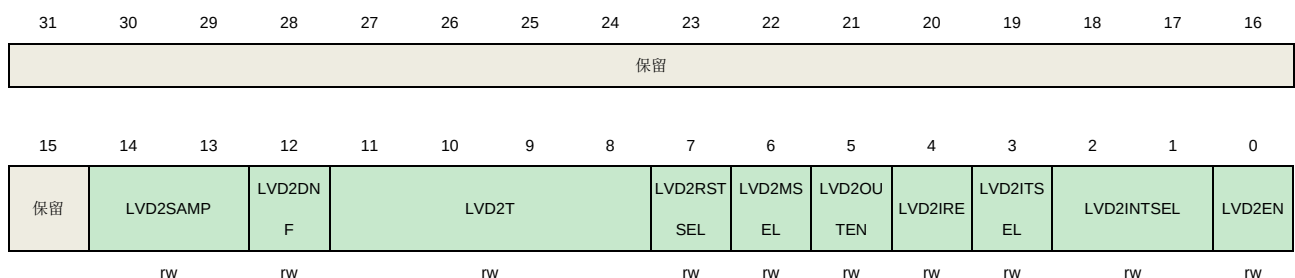
		0: 中断模式 1: 复位模式
5	LVD1OUTEN	LVD1 比较结果输出使能: 0: LVD1 结果输出失能 1: LVD1 结果输出使能
4	LVD1IRE	低电压检测器 1 中断/复位使能 0: LVD1 中断 / 复位模式失能 1: LVD1 中断 / 复位模式使能
3	LVD1ITSEL	LVD1 中断类型选择 0: 不可屏蔽中断 1: 可屏蔽中断
2:1	LVD1INTSEL[1:0]	LVD1 中断触发模式 00: VCC >= Vdet1 (上升) 01: VCC < Vdet1 (下降) 10: VCC >= Vdet1 或 VCC < Vdet1 (上升或下降) 11: 保留
0	LVD1EN	LVD1 使能 0: LVD1 失能 1: LVD1 使能

3.4.4. 低电压检测器 2 控制寄存器 (PMU_LVD2CTL)

地址偏移: 0x0C

复位值: 0x0000 1682 (从待机模式唤醒后不复位)

该寄存器可以按字 (32位) 访问。



位/位域	名称	描述
31:15	保留	必须保持复位值。
14:13	LVD2SAMP	采样时钟选择 00: PCLK1 频率的 1 / 2 01: PCLK1 频率的 1 / 4 10: PCLK1 频率的 1 / 8

		11: PCLK1 频率的 1 / 16
12	LVD2DNF	LVD2 数字噪声滤波器使能 0: 数字噪声滤波器使能 1: 数字噪声滤波器失能
11:8	LVD2T[3:0]	LVD2 检测阈值 0100: 4.62V / 4.54V (上升 / 下降) 0101: 4.55V / 4.47V (上升 / 下降) 0110: 4.40V / 4.32V (上升 / 下降) 1010: 3.03V / 2.95V (上升 / 下降) 1011: 2.96V / 2.88V (上升 / 下降) 1100-1111: 保留
7	LVD2RSTSEL	LVD2 复位信号解除选择: 0: 在检测到 $VCC > V_{det2}$ 后, 复位信号才会在稳定时间 (t_{LVD2}) 后解除 (在 $VCC \leq V_{det2}$ 时 MCU 一直处于复位状态)。 1: LVD2 复位信号产生后, 在经过一个稳定时间 (t_{LVD2}) 后, 复位信号将解除 (MCU 产生复位一次)。
6	LVD2MSEL	LVD2 模式选择: 0: 中断模式 1: 复位模式
5	LVD2OUTEN	LVD2 比较结果输出使能: 0: LVD2 结果输出失能 1: LVD2 结果输出使能
4	LVD2IRE	LVD2 中断 / 复位模式使能 0: LVD2 中断 / 复位模式失能 1: LVD2 中断 / 复位模式使能
3	LVD2ITSEL	LVD2 中断类型选择 0: 不可屏蔽中断 1: 可屏蔽中断
2:1	LVD2INTSEL[1:0]	LVD2 中断触发模式 00: $VCC \geq V_{det2}$ (上升) 01: $VCC < V_{det2}$ (下降) 10: $VCC \geq V_{det2}$ 或 $VCC < V_{det2}$ (上升或下降) 11: 保留
0	LVD2EN	LVD2 使能 0: LVD2 失能 1: LVD2 使能

4. 复位和时钟单元（RCU）

4.1. 复位控制单元（RCTL）

4.1.1. 简介

GD32M531复位控制包括2种复位控制：电源复位、系统复位。电源复位又称为冷复位，电源启动时复位所有系统。除了SW-DP控制器，系统复位将复位处理器内核和外设IP部分。复位被外部信号、内部事件和复位发生器触发。接下章节将详细介绍这些复位。

4.1.2. 功能描述

电源复位

当以下事件中之一发生时，产生电源复位：1、上电/掉电复位（POR/PDR复位）；2、从待机模式中返回后由内部复位发生器产生。电源复位复位所有的寄存器。电源复位为低电平有效，当内部LDO电源基准准备好提供1.2V电压给GD32M531产品时，电源复位电平将变为无效。复位入口向量被固定在存储器映射地址0x0000_0004。

系统复位

当发生以下任一事件时，产生一个系统复位：

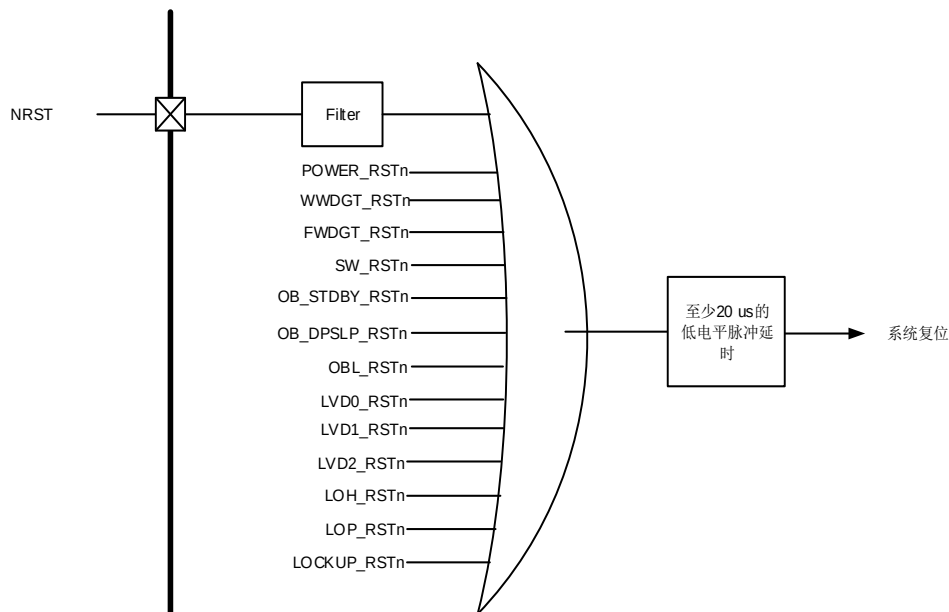
1. 电源复位（POWER_RSTn）；
2. 外部引脚复位（NRST）；
3. 窗口看门狗定时器计数终止（WWDGT_RSTn）；
4. 独立看门狗定时器计数终止（FWDGT_RSTn）；
5. Cortex®-M33的中断应用和复位控制寄存器中的SYSRESETREQ位置‘1’（SW_RSTn）；
6. 选项字节装载器复位（OBL_RSTn）；
7. 选项字节寄存器nRST_STDBY设置为0，并且进入待机模式时将产生复位（OB_STDBY_RSTn）；
8. 选项字节寄存器nRST_DPSLP设置为0，并且进入深度睡眠模式时将产生复位（OB_DPSLP_RSTn）；
9. 低电压0检测复位（LVD0_RSTn）；
10. 低电压1检测复位（LVD1_RSTn）；
11. 低电压2检测复位（LVD2_RSTn）；
12. HXTAL丢失复位（LOH_RSTn）；
13. PLL丢失复位（LOP_RSTn）；
14. CPU锁死复位（LOCKUP_RSTn）。

LVD_RSTn / LOH_RSTn / LOP_RSTn / LOCKUP_RSTn应通过软件启用。

除了SW-DP控制器，系统复位将复位处理器内核和外设IP部分。

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

图4-1. 系统复位电路



注意：LVD_RSTn/LOH_RSTn/LOP_RSTn/LOCKUP_RSTn产生复位是有条件的。必须在PMU部分设置LVD0RSTEN / LVD1RSTEN / LVD2RSTEN，以及在复位源/时钟寄存器（RCU_RSTSCK）中设置 LOHRSTEN / LOPRSTEN / LOCKUPRSTEN位。

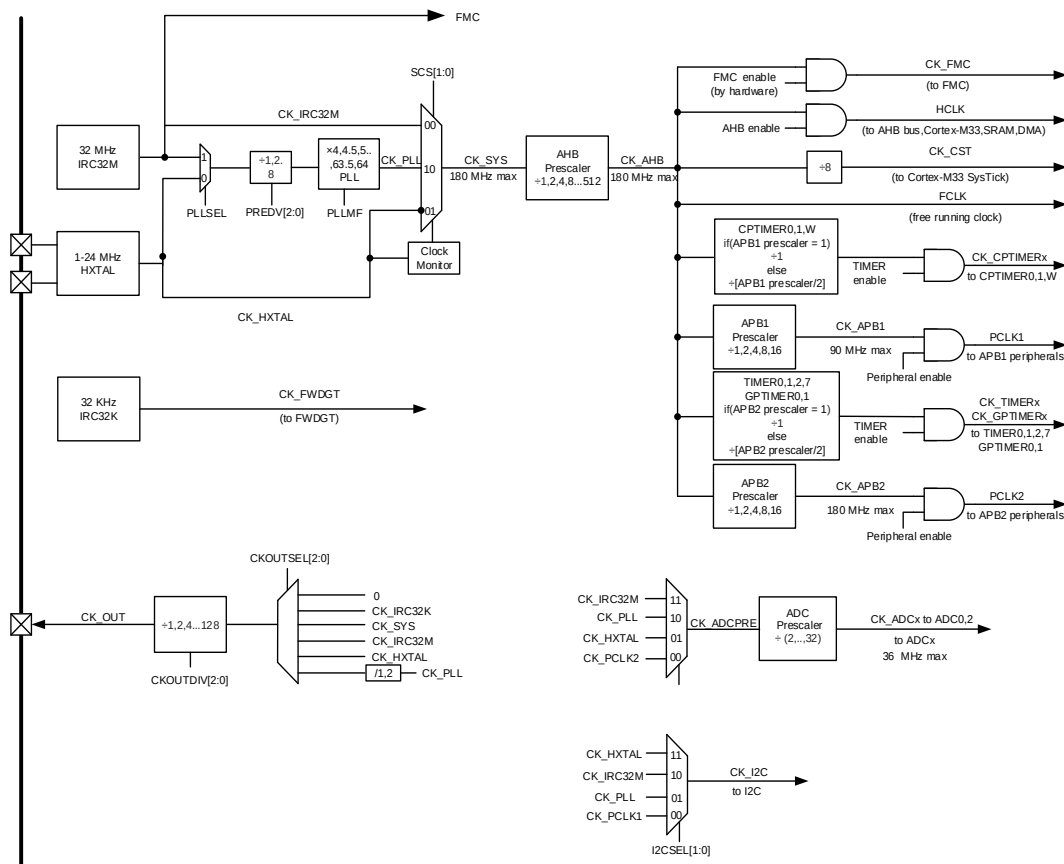
4.2. 时钟控制单元（CCTL）

4.2.1. 简介

时钟控制单元提供了一系列频率和时钟功能，包括一个内部32M RC振荡器时钟(IRC32M)、一个外部高速晶体振荡器时钟（HXTAL）、一个内部低速RC振荡器时钟（IRC32K）、一个锁相环（PLL）、PLL时钟监控器（PLLM）、一个HXTAL时钟监视器、时钟预分频器、时钟多路复用器和时钟选通电路。

AHB、APB和Cortex®-M33时钟都源自系统时钟（CK_SYS），系统时钟的时钟源为IRC32M、HXTAL或PLL。系统时钟的最大运行时钟频率可以达到180MHz。

图4-2. 时钟树



预分频器可以配置 AHB、APB2 和 APB1 域的时钟频率。AHB、APB2 和 APB1 域的最高时钟频率分别为 180MHz、180MHz 和 90MHz。RCU 通过 AHB 时钟 (HCLK) 8 分频后作为 Cortex 系统定时器 (SysTick) 的外部时钟。通过对 SysTick 控制与状态寄存器的设置, 可选择上述时钟或 Cortex (HCLK) 时钟作为 SysTick 时钟。

模数转换器 (ADCs) 由 AHB 时钟驱动, 时钟公式为: $CK\_ADC = CK\_AHB / (ADCPSC + 2)$ 。

注意: ADC 的最高时钟需要参考数据手册。

I2C 由 CK_APB1、CK_PLL、CK_IRC32M 或 CK_HXTAL 时钟驱动, 这些时钟由 RCU_CFG2 中的 I2CSEL 位定义。I2C 支持动态时钟切换。

FWDGT 时钟可以选择 IRC32K 时钟, 当 FWDGT 启动时强制选择。

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

4.2.2. 主要特性

- 1 到 24 MHz 外部高速晶体振荡器 (HXTAL)
- 32 MHz 内部高速 RC 振荡器 (IRC32M)
- 32 KHz 内部低速 RC 振荡器 (IRC32K)
- PLL 时钟源可以是 HXTAL 或 IRC32M

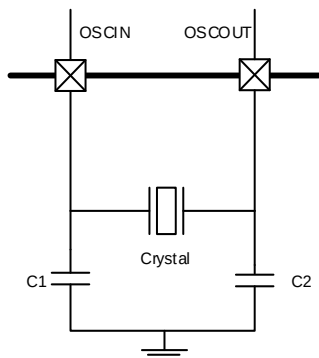
■ HXTAL时钟监视

4.2.3. 功能描述

高速外部晶体振荡器时钟（HXTAL）

1到24MHz的外部振荡器可为系统提供更为精确的主时钟。带有特定频率的晶体必须靠近两个HXTAL的引脚。和晶体连接的外部电阻和电容必须根据所选择的振荡器来调整。

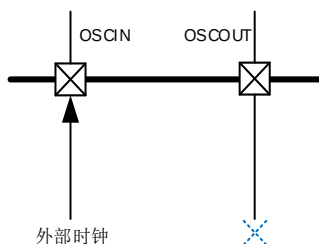
图4-3. HXTAL时钟源



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

将控制寄存器RCU_CTL的HXTALBPS和HXTALEN位置‘1’来使能外部旁路模式。旁路输入时，信号接至OSCIN，OSCOUT保持悬空状态，如[图4-4. 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4-4. 旁路模式下HXTAL时钟源



通过使用控制寄存器（RCU_CTL）中的HXTALSCAL位选择HXTAL频率范围。如果HXTAL频率高于10MHz，则必须将HXTALSCAL位置1。

高速内部32MHz RC振荡器时钟（IRC32M）

高速内部32MHz RC振荡器时钟，简称IRC32M时钟，拥有32MHz的固定频率，设备上电后CPU默认选择的时钟源就是IRC32M时钟。IRC32M RC振荡器能够在不需要任何外部器件的条件下提供更低成本类型的时钟源。IRC32M晶体可以通过设置时钟控制寄存器（RCU_CTL）中的

IRC32MEN位被启动和关闭。时钟控制寄存器RCU_CTL中的IRC32MSTB位用来指示IRC32M内部RC振荡器是否稳定。IRC32M振荡器的启动时间比HXTAL晶体振荡器要更短。如果时钟中断寄存器RCU_INT中的相应中断使能位IRC32MSTBIE被置‘1’，在IRC32M稳定以后，将产生一个中断。IRC32M时钟也可用作PLL输入时钟。

工厂会校准IRC32M时钟频率的精度，但是它的精度仍然比HXTAL时钟要差。用户需求、环境条件和成本将决定选择哪个时钟作为系统时钟源。

如果HXTAL或者PLL是系统时钟源，为了最大程度减小系统从深度睡眠模式启动的时间，系统从深度睡眠模式初始唤醒的时候硬件强制IRC32M时钟作为系统时钟。

锁相环（PLL）

内部锁相环PLL通过对输入参考频率为4~16MHz的时钟基准4~64倍频，可以提供40~180 MHz的时钟输出。

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

低速内部RC振荡器时钟（IRC32K）

IRC32K RC振荡器时钟担当一个低功耗时钟源的角色，它的时钟频率大约32KHz，为独立看门狗定时器和实时时钟电路提供时钟。IRC32K提供低成本的时钟源，因为不需要外部器件。IRC32K RC振荡器可以通过设置控制/状态寄存器RCU_RSTSCK中的IRC32KEN位被启动和关闭。控制/状态寄存器RCU_RSTSCK中的IRC32KSTB位用来指示IRC32K时钟是否已稳定。如果时钟中断寄存器RCU_INT中的相应中断使能位IRC32KSTBIE被置‘1’，在IRC32K稳定以后，将产生一个中断。

系统时钟（CK_SYS）选择

系统复位后，IRC32M时钟被选为系统时钟，改变时钟配置寄存器RCU_CFG0中的系统时钟变换位SCS可以切换系统时钟源为HXTAL或PLL。当SCS的值改变，系统时钟将使用原来的时钟源继续运行直到转换的目标时钟源稳定。当一个时钟源被直接或通过PLL间接作为系统时钟时，它将不能被停止。

PLL时钟监控器（PLLM）

PLL时钟监控器功能通过控制寄存器（RCU_CTL）中的PLL时钟监控器使能位PLLMEN来使能。该功能应在PLL启动延迟后启用，并在PLL停止或频率降低时禁用。一旦检测到PLL故障，就会产生PLL丢失复位或中断。复位功能由寄存器RCU_RSTSCK中的LOPRSTEN位决定。中断功能由寄存器RCU_INT中的PLLMIE位决定。

PLL时钟由IRC32M监控。

如果LOPRSTEN位为0，将置位中断寄存器RCU_INT中的PLL丢失标志PLLMIF，并生成PLL故障中断。PLL将被自动禁用。如果选择PLL作为CK_SYS的时钟源，则PLL故障将迫使CK_SYS源为IRC32M，并且PLL将被自动禁用。

如果LOPRSTEN为1，则会产生PLL丢失复位。

HXTAL时钟监视器（CKM）

设置时钟控制寄存器RCU_CTL中的HXTAL时钟监视使能位CKMEN，HXTAL可以使能时钟监视功能。该功能必须在HXTAL启动延迟完毕后使能，在HXTAL停止后禁止。一旦检测到HXTAL故障，HXTAL丢失复位由寄存器RCU_RSTSCK中的LOHRSTEN位决定是否产生。NMI中断是由SYSCFG_CFG2寄存器的HXTAL_NMIIE位决定。

如果LOHRSTEN为0，则将时钟中断寄存器RCU_INT中的HXTAL时钟阻塞标志位CKMIF将被置‘1’，产生HXTAL故障事件。HXTAL将被自动禁用。此故障中断连接到Cortex®-M33的不可屏蔽中断NMI。如果选择HXTAL作为CK_SYS或PLL的时钟源，则HXTAL故障将迫使CK_SYS源为IRC32M，并且PLL将被自动禁用。

如果LOHRSTEN为1，则会生成HXTAL丢失复位。

时钟输出功能

时钟输出功能输出从32kHz到180MHz的时钟。通过设置时钟配置寄存器RCU_CFG0中的CK_OUT时钟源选择位CKOUTSEL能够选择不同的时钟信号。相应的GPIO引脚应该被配置成复用功能I/O（AFIO）模式来输出选择的时钟信号

表 4-1. 时钟源的选择

时钟源选择位	时钟源
000	无时钟
001	保留
010	CK_IRC32K
011	保留
100	CK_SYS
101	CK_IRC32M
110	CK_HXTAL
111	CK_PLL 或 CK_PLL/8

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

4.2.4. 事件互连的操作**事件信号输出到EVIC**

当事件互连单元（EVIC）使用其中断请求信号作为事件信号，并检测到HXTAL时钟振荡停止时，时钟生成电路能够与预先设置的另一个模块进行操作互连。

时钟生成电路输出事件信号，不论相应的HXTAL停止检测中断使能位（RCU_INT.CKMIF）的设置如何,详细信息请参考事件互连单元（EVIC）。

从EVIC输入事件信号

当EVIC中的EVIC_RCU寄存器指定的事件发生时，时钟生成电路能够响应预先设置的事件，将时钟源切换到IRC32M。在使用此功能时，从停止模式返回时不能使用时钟源切换。

4.3. RCU 寄存器

RCU基地址：0x4002 1000

4.3.1. 控制寄存器（RCU_CTL）

地址偏移：0x00

复位值：0x0000 XX83 X表示未定义

注意：在重写此寄存器之前，需将 SYSCFG.PRCFG0 位设置为 1（写使能）。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留						PLLSTB	PLLEN	保留	HXTALS CAL	PLLMEN	CKMEN	HXTALBP S	HXTALST B	HXTALR ST	HXTALE N
						r	rw		rw	rw	rw	rw	r	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRC32MCALIB[7:0]								IRC32MADJ[4:0]					保留	IRC32MS TB	IRC32ME N
														r	rw

位/位域	名称	描述
31:26	保留	必须保持复位值。
25	PLLSTB	PLL时钟稳定标志位 硬件置‘1’来指示PLL输出时钟是否稳定待用。 0: PLL没稳定 1: PLL 稳定
24	PLLEN	PLL使能 软件置位或复位。如果PLL时钟作为系统时钟的时候该位不能被复位。进入深度睡眠或待机模式时硬件自动复位。 0: PLL被关闭 1: PLL 被打开
23	保留	必须保持复位值。
22	HXTALSCAL	HXTAL 频率范围选择 仅当 HXTALEN 为 0 时，才能写入 HXTALSCAL 位。当 HXTAL 频率为 10MHz 时，HXTALSCAL 应被设为 1。仅在使用无源晶振（HXTALBPS 为 0）时需要配置该位。 0: HXTAL 频率范围为 1 ~ 10MHz 1: HXTAL 频率范围为 10 ~ 24MHz
21	PLLMEN	PLL 时钟监控器使能 0: 禁能 PLL 时钟监控器 1: 使能 PLL 时钟监控器

		PLL MEN 使硬件能够检测到 PLL 时钟卡在低/高状态。 如果禁用了 PLL，则该位不能设置为 1。当进入深度睡眠或待机模式或检测到 PLL 解锁时，它会由硬件复位。
20	CKMEN	HXTAL 时钟监视使能 0: 禁止外部 1 ~ 24 MHz 晶体振荡器 (HXTAL) 时钟监视器 1: 使能外部 1 ~ 24 MHz 晶体振荡器 (HXTAL) 时钟监视器 当硬件监测到 HXTAL 时钟一直停留在低或者高的状态，内部硬件将切换系统时钟到 IRC32M RC 时钟。恢复原来系统时钟的方式有以下几种：外部复位，上电复位，软件清除 CKMIF 位。 注意：使能 HXTAL 时钟监视器以后，硬件无视控制位 IRC32MEN 的状态，自动使能 IRC32M 时钟。
19	HXTALBPS	外部晶体振荡器 (HXTAL) 时钟旁路模式使能 只有在 HXTALEN 位为 0 时，HXTALBPS 位才可写。 0: 禁止 HXTAL 旁路模式 1: 使能 HXTAL 旁路模式，HXTAL 输出时钟等于输入时钟
18	HXTALSTB	外部晶体振荡器 (HXTAL) 时钟稳定状态标志位 硬件置 '1' 来指示 HXTAL 振荡器时钟是否稳定待用。 0: HXTAL 振荡器未稳定 1: HXTAL 振荡器已稳定
17	HXTALRST	外部高速振荡器稳定复位 通过软件设置和复位。如果 HXTAL 时钟被用作系统时钟或 PLL 输入时钟，则此位不能被复位。在进入深度睡眠或待机模式时，硬件将复位此位。 0: 不复位 1: 复位 HXTAL 注意：当 HXTAL 不被用作系统时钟时，可以设置此位。
16	HXTALEN	外部高速振荡器时钟使能 软件置 '1' 或清 '0'。如果 HXTAL 时钟或者 PLL 输入时钟作为系统时钟，该位不能被复位。进入深度睡眠或待机模式时硬件自动复位。 0: 禁止外部 1 ~ 24 MHz 晶体振荡器 1: 使能外部 1 ~ 24 MHz 晶体振荡器
15:8	IRC32MCALIB[7:0]	高速内部振荡器校准值寄存器 上电时自动加载这些位
7:3	IRC32MADJ[4:0]	高速内部振荡器时钟调整值 这些位由软件置位，最终调整值为 IRC32MADJ 当前值加上 IRC32MCALIB[7:0] 位的值。最终调整值应该调整 IRC32M 到 32 MHz $\pm$ 1%。
2	保留	必须保持复位值。
1	IRC32MSTB	高速内部 (IRC32M) 时钟稳定状态标志位 硬件置 '1' 来指示 IRC32M 振荡器时钟是否稳定待用。 0: IRC32M 振荡器未稳定

1: IRC32M 振荡器已稳定

0 IRC32MEN

高速内部振荡器使能

软件复位置位。如果IRC32M时钟用作系统时钟时该位不能被复位。当从待机和深度睡眠模式返回或用作系统时钟的HXTAL振荡器发生故障时，该位由硬件置1来启动IRC32M振荡器。

0: 内部32MHz RC振荡器关闭

1: 内部 32MHz RC 振荡器开启

4.3.2. 配置寄存器 0 (RCU_CFG0)

地址偏移: 0x04

复位值: 0x001E 0400

注意: 在重写此寄存器之前，需将 SYSCFG.PRCFG0 位设置为 1（写使能）。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

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

位/位域	名称	描述
31	PLLDV	CK_PLL 1或8分频来用作CK_OUT 0: CK_PLL 8分频用作CK_OUT 1: CK_PLL 用作 CK_OUT
30:28	CKOUTDIV[2:0]	CK_OUT分频器，来降低CK_OUT频率 CK_OUT的选择参考RCU_CFG0的26:24位。 000: CK_OUT不分频 001: CK_OUT 2分频 010: CK_OUT 4分频 011: CK_OUT 8分频 100: CK_OUT 16分频 101: CK_OUT 32分频 110: CK_OUT 64分频 111: CK_OUT 128 分频
27	保留	必须保持复位值。
26:24	CKOUTSEL[2:0]	CK_OUT时钟源选择 软件置位或清零。 000: 没有时钟被选择 001: 保留

		010: 选择内部32K RC振荡器时钟
		011: 保留
		100: 选择系统时钟
		101: 选择内部32M RC振荡器时钟
		110: 选择外部高速振荡器时钟
		111: 依赖于 PLLDV 选择 (CK_PLL / 8) 或 CK_PLL
23:17	PLLMF[6:0]	PLL倍频因子
		0000111: (PLL 时钟源 x 4)
		0001000: (PLL 时钟源 x 4.5)
		0001001: (PLL 时钟源 x 5)
		0001010: (PLL 时钟源 x 5.5)
		
		1111110: (PLL 时钟源 x 63.5)
		1111111: (PLL时钟源 x 64)
		注意: PLL 输出频率不能超过 180MHz。
16	PLLSEL	PLL时钟源选择
		软件置1或清0来控制PLL时钟源
		0: 选择HXTAL为PLL时钟源
		1: 选择 IRC32M 为 PLL 时钟源
15:14	保留	必须保持复位值。
13:11	APB2PSC[2:0]	APB2预分频选择
		软件置1和清0来控制APB2时钟分频因子。
		0xx: 选择AHB时钟不分频
		100: 选择AHB时钟2分频
		101: 选择AHB时钟4分频
		110: 选择AHB时钟8分频
		111: 选择 AHB 时钟 16 分频
10:8	APB1PSC[2:0]	APB1预分频选择
		软件设置和清除来控制APB1时钟分频因子。
		0xx: 选择AHB时钟不分频
		100: 选择AHB时钟2分频
		101: 选择AHB时钟4分频
		110: 选择AHB时钟8分频
		111: 选择 AHB 时钟 16 分频
7:4	AHBPSC[3:0]	AHB预分频选择
		软件设置和清除来控制AHB时钟分频因子。
		0xxx: 选择CK_SYS系统时钟不分频
		1000: 选择CK_SYS系统时钟2分频
		1001: 选择CK_SYS系统时钟4分频
		1010: 选择CK_SYS系统时钟8分频

		1011: 选择CK_SYS系统时钟16分频 1100: 选择CK_SYS系统时钟64分频 1101: 选择CK_SYS系统时钟128分频 1110: 选择CK_SYS系统时钟256分频 1111: 选择 CK_SYS 系统时钟 512 分频
3:2	SCSS[1:0]	系统时钟转换状态 硬件设置和清除指示系统当前时钟源 00: 选择CK_IRC32M作为CK_SYS系统时钟源 01: 选择CK_HXTAL作为CK_SYS系统时钟源 10: 选择CK_PLL作为CK_SYS系统时钟源 11: 保留
1:0	SCS[1:0]	系统时钟转换 软件设置选择系统时钟源。由于CK_SYS的改变有固有的延迟, 需要软件读SCSS位来确保转换是否结束。在从深度睡眠或待机模式中返回时, 或作为系统时钟或PLL时钟源的HXTAL出现故障时, 强制选择IRC32M作为系统时钟或PLL时钟。 00: 选择IRC32M时钟作为CK_SYS系统时钟源 01: 选择HXTAL时钟作为CK_SYS系统时钟源 10: 选择PLL作为CK_SYS系统时钟源 11: 保留

4.3.3. 中断寄存器（RCU_INT）

地址偏移: 0x08

复位值: 0x0000 0000

注意: 在重写此寄存器之前, 需将 SYSCFG.PRCFG0 位设置为 1 (写使能)。

该寄存器可以按字节 (8位)、半字 (16位) 或字 (32位) 访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								CKMIC	PLLMIC	保留	PLL STBIC	HXTAL STBIC	IRC32M STBIC	保留	IRC32K STBIC
								w	w		w	w	w		w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	PLLMIE	保留	PLL STBIE	HXTAL STBIE	IRC32M STBIE	保留	IRC32K STBIE	CKMIF	PLLMIF	保留	PLL STBIF	HXTAL STBIF	IRC32M STBIF	保留	IRC32K STBIF
rw		rw		rw	rw	rw		r	r		r	r	r		r

位/位域	名称	描述
31:24	保留	必须保持复位值。
23	CKMIC	HXTAL时钟阻塞中断清除 软件写1复位CKMIF标志位。 0: 不复位CKMIF标志位

		1: 复位 CKMIF 标志位
22	PLLMIF	PLL 时钟监控器中断清除 软件写 1 复位 PLLMIF 标志位。 0: 不复位 PLLMIF 标志 1: 重置 PLLMIF 标志
21	保留	必须保持复位值。
20	PLLSTBIC	PLL 稳定中断清除 软件写 1 复位 PLLSTBIF 标志位。 0: 不复位 PLLSTBIF 标志位 1: 复位 PLLSTBIF 标志位
19	HXTALSTBIC	HXTAL 时钟稳定中断清除 软件写 1 复位 HXTALSTBIF 标志位。 0: 不复位 HXTALSTBIF 标志位 1: 复位 HXTALSTBIF 标志位
18	IRC32MSTBIC	IRC32M 时钟稳定中断清除 软件写 1 复位 IRC32MSTBIF 标志位。 0: 不复位 IRC32MSTBIF 标志位 1: 复位 IRC32MSTBIF 标志位
17	保留	必须保持复位值。
16	IRC32KSTBIC	IRC32K 时钟稳定中断清除 软件写 1 复位 IRC32KSTBIF 标志位。 0: 不复位 IRC32KSTBIF 标志位 1: 复位 IRC32KSTBIF 标志位
15	保留	必须保持复位值。
14	PLLMIE	PLL 时钟监控器中断使能 软件置 1 和清 0 来使能/禁止 PLL 时钟监控器中断。 0: 禁止 PLL 时钟监视器中断 1: 使能 PLL 时钟监控器中断
13	保留	必须保持复位值。
12	PLLSTBIE	PLL 时钟稳定中断使能 软件置 1 和清 0 来使能/禁止 PLL 时钟稳定中断。 0: 禁止 PLL 时钟稳定中断 1: 使能 PLL 时钟稳定中断
11	HXTALSTBIE	HXTAL 时钟稳定中断使能 软件置 1 和清 0 来使能/禁止 HXTAL 时钟稳定中断。 0: 禁止 HXTAL 时钟稳定中断 1: 使能 HXTAL 时钟稳定中断

10	IRC32MSTBIE	IRC32M时钟稳定中断使能 软件置1和清0来使能/禁止IRC32M时钟稳定中断。 0: 禁止IRC32M时钟稳定中断 1: 使能 IRC32M 时钟稳定中断
9	保留	必须保持复位值。
8	IRC32KSTBIE	IRC32K时钟稳定中断使能 IRC32K时钟稳定中断使能/禁止控制。 0: 禁止IRC32K时钟稳定中断 1: 使能 IRC32K 时钟稳定中断
7	CKMIF	HXTAL时钟阻塞中断标志位 当HXTAL时钟阻塞时硬件置1。 软件置 CKMIC=1 时清除该位。 软件置 CKMNMIIE=1 时清除该位。 0: 时钟运行正常 1: HXTAL 时钟阻塞
6	PLLMIF	PLL 时钟监视器中断标志 当 PLL 时钟卡住时，由硬件置 1。 软件置 PLLMIC=1 时清除该位。 0: PLL 时钟运行正常 1: PLL 时钟卡住
5	保留	必须保持复位值。
4	PLLSTBIF	PLL稳定中断清除 软件写1复位PLLSTBIF标志位。 0: 不复位PLLSTBIF标志位 1: 复位 PLLSTBIF 标志位
3	HXTALSTBIF	HXTAL时钟稳定中断标志位 当外部1 ~ 24 MHz晶体振荡器时钟稳定且HXTALSTBIE位被置1时由硬件置1。 软件置HXTALSTBIC=1时清除该位。 0: 无HXTAL时钟稳定中断发生 1: 发生 HXTAL 时钟稳定中断
2	IRC32MSTBIF	IRC32M时钟稳定中断标志位 当内部32 MHz RC振荡器时钟稳定且IRC32MSTBIE位被置1时由硬件置1。 软件置IRC32MSTBIC=1时清除该位。 0: 无IRC32M时钟稳定中断产生 1: 产生 IRC32M 时钟稳定中断
1	保留	必须保持复位值。
0	IRC32KSTBIF	IRC32K时钟稳定中断标志位 当内部32KHz RC振荡器时钟稳定且IRC32KSTBIE位被置1时由硬件置1。

软件置IRC32KSTBIC =1时清除该位。

0: 无IRC32K时钟稳定中断产生

1: 产生 IRC32K 时钟稳定中断

4.3.4. APB2 复位寄存器 (RCU_APB2RST)

地址偏移: 0x0C

复位值: 0x0000 0000

注意: 在重写此寄存器之前, 需将 SYSCFG.PRCFG1 位设置为 1 (写使能)。

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	CANRST	保留							GPTIMER1RST	GPTIME R0RST	保留			TIMER2R ST	TIMER1R ST
rw									rw	rw	rw				rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		TIMER7R ST.	SPIRST	TIMER0R ST	保留	ADC2RS T	ADC0RS T	保留						CMRPRST	CFGRST
rw			rw	rw	rw		rw	rw							rw

位/位域	名称	描述
31	保留	必须保持复位值。
30	CANRST	CAN 复位 由软件置 1 或清 0。 0: 无复位 1: 复位 CAN
29:23	保留	必须保持复位值。
22	GPTIMER1RST	GPTIMER1 复位 由软件置 1 或清 0。 0: 无复位 1: 复位 GPTIMER1
21	GPTIMER0RST	GPTIMER0 复位 由软件置 1 或清 0。 0: 无复位 1: 复位 GPTIMER0
20:18	保留	必须保持复位值。
17	TIMER2RST	TIMER2 复位 由软件置 1 或清 0。 0: 无复位

		1: 复位 TIMER2
16	TIMER1RST	TIMER1 复位 由软件置 1 或清 0。 0: 无复位 1: 复位 TIMER1
15:14	保留	必须保持复位值。
13	TIMER7RST	TIMER7 复位 由软件置 1 或清 0。 0: 无复位 1: 复位 TIMER7
12	SPIRST	SPI 复位 由软件置 1 或清 0。 0: 无复位 1: 复位 SPI
11	TIMER0RST	TIMER0 复位 由软件置 1 或清 0。 0: 无复位 1: 复位 TIMER0
10	保留	必须保持复位值。
9	ADC2RST	ADC2 复位 由软件置 1 或清 0。 0: 无复位 1: 复位 ADC2
8	ADC0RST	ADC0 复位 由软件置 1 或清 0。 0: 无复位 1: 复位 ADC0
7:2	保留	必须保持复位值。
1	CMPRST	比较器复位 由软件置 1 或清 0。 0: 无复位 1: 复位比较器
0	CFGRST	系统配置复位 由软件置 1 或清 0。 0: 无复位 1: 复位系统配置

地址偏移: 0x10
复位值: 0x0000 0000

注意：在重写此寄存器之前，需将 SYSCFG.PRCFG1 位设置为 1（写使能）。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	保留			CFMURST	保留		I2CRST	保留	UART3RST	UART2RST	UART1RST	UART0RST
		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			WWDGTRST	保留							CPTIMERWRST	保留		CPTIMER1RST	CPTIMER0RST
			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 1 或清 0。 0: 无复位 1: 复位 DAC
28	PMURST	电源控制复位 由软件置1或清0。 0: 无复位 1: 复位电源控制单元
27:25	保留	必须保持复位值。
24	CFMURST	CFMU 复位 由软件置 1 或清 0。 0: 无复位 1: 复位 CFMU
23:22	保留	必须保持复位值。
21	I2CRST	I2C 复位 由软件置 1 或清 0。 0: 无复位 1: 复位 I2C
20	保留	必须保持复位值。
19	UART3RST	UART3 复位 由软件置 1 或清 0。 0: 无复位

		1: 复位 UART3
18	UART2RST	UART2 复位 由软件置 1 或清 0。 0: 无复位 1: 复位 UART2
17	UART1RST	UART1 复位 由软件置 1 或清 0。 0: 无复位 1: 复位 UART1
16	UART0RST	UART0 复位 由软件置 1 或清 0。 0: 无复位 1: 复位 UART0
15:13	保留	必须保持复位值。
12	WWDGTRST	WWDGT 复位 由软件置 1 或清 0。 0: 无复位 1: 复位 WWDGT
11:5	保留	必须保持复位值。
4	CPTIMERWRST	CPTIMERW 定时器复位 由软件置 1 或清 0。 0: 无复位 1: 复位 CPTIMERW 定时器
3:2	保留	必须保持复位值。
1	CPTIMER1RST	CPTIMER1 定时器复位 由软件置 1 或清 0。 0: 无复位 1: 复位 CPTIMER1 定时器
0	CPTIMER0RST	CPTIMER0 定时器复位 由软件置 1 或清 0。 0: 无复位 1: 复位 CPTIMER0 定时器

4.3.6. AHB 使能寄存器 (RCU_AHBEN)

地址偏移: 0x14

复位值: 0x0000 0014

注意: 在重写此寄存器之前, 需将 SYSCFG.PRCFG1 位设置为 1 (写使能)。Bit 4 不能被清

为 0。

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	TMUEN	SVPWME N	GTOCEN	POCEN	保留	PNEN	PGEN	PFEN	PEEN	PDEN	PCEN	PBEN	PAEN		
	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
保留								CRCCEN	保留	DMAMUX EN	SRAMSP EN	DMA1EN	DMA0EN		
								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	TMUEN	TMU时钟使能 由软件置1或清0。 0: TMU时钟关闭 1: TMU 时钟开启
28	SVPWMEN	SVPWM时钟使能 由软件置1或清0。 0: SVPWM时钟关闭 1: SVPWM 时钟开启
27	GTOCEN	GTOC时钟使能 由软件置1或清0。 0: GTOC时钟关闭 1: GTOC 时钟开启
26	POCEN	POC时钟使能 由软件置1或清0。 0: POC时钟关闭 1: POC 时钟开启
25:24	保留	必须保持复位值。
23	PNEN	GPION时钟使能 由软件置1或清0。 0: GPION时钟关闭 1: GPION 时钟开启
22	PGEN	GPIOG时钟使能 由软件置1或清0。 0: GPIOG时钟关闭 1: GPIOG 时钟开启
21	PFEN	GPIOF时钟使能

		由软件置1或清0。 0: GPIOF时钟关闭 1: GPIOF 时钟开启
20	PEEN	GPIOE时钟使能 由软件置1或清0。 0: GPIOE时钟关闭 1: GPIOE 时钟开启
19	PDEN	GPIOD时钟使能 由软件置1或清0。 0: GPIOD时钟关闭 1: GPIOD 时钟开启
18	PCEN	GPIOC时钟使能 由软件置1或清0。 0: GPIOC时钟关闭 1: GPIOC 时钟开启
17	PBEN	GPIOB时钟使能 由软件置1或清0。 0: GPIOB时钟关闭 1: GPIOB 时钟开启
16	PAEN	GPIOA时钟使能 由软件置1或清0。 0: GPIOA时钟关闭 1: GPIOA 时钟开启
15:7	保留	必须保持复位值。
6	CRCEN	CRC 时钟使能 由软件置1或清0。 0: CRC时钟关闭 1: CRC 时钟开启
5:4	保留	必须保持复位值。
3	DMAMUXEN	DMAMUX 时钟使能 由软件置 1 或清 0。 0: DMAMUX时钟关闭 1: DMAMUX 时钟开启
2	SRAMSPEN	SRAM接口时钟使能 由软件置1或清0来开启/关闭在睡眠模式下的SRAM时钟。 0: 关闭睡眠模式下的SRAM接口时钟 1: 开启睡眠模式下的 SRAM 接口时钟
1	DMA1EN	DMA1时钟使能

由软件置1或清0。

0: 关闭DMA1时钟

1: 开启 DMA1 时钟

0	DMA0EN	DMA0时钟使能 由软件置1或清0。 0: 关闭DMA0时钟 1: 开启 DMA0 时钟

4.3.7. APB2 使能寄存器 (RCU_APB2EN)

地址偏移: 0x18

复位值: 0x0000 0001

注意: 在重写此寄存器之前, 需将 SYSCFG.PRCFG1 位设置为 1 (写使能)。

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	CANEN	保留					EVICEN	保留	GPTIMER1EN	GPTIME R0EN	保留			TIMER2E N	TIMER1E N	
rw							rw		rw		rw		rw			rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
保留		TIMER7E N	SPIEN	TIMER0E N	保留	ADC2EN	ADC0EN	保留							CMPEN	CFGEN
		rw	rw	rw	rw		rw								rw	rw

位/位域	名称	描述
31	保留	必须保持复位值。
30	CANEN	CAN 时钟使能 由软件置 1 或清 0。 0: 关闭CAN时钟 1: 开启 CAN 时钟
29:25	保留	必须保持复位值。
24	EVICEN	EVIC定时器时钟使能 由软件置1或清0。 0: 关闭EVIC定时器时钟 1: 开启 EVIC 定时器时钟
23	保留	必须保持复位值。
22	GPTIMER1EN	GPTIMER1定时器时钟使能 由软件置1或清0。 0: 关闭GPTIMER1定时器时钟 1: 开启 GPTIMER1 定时器时钟

21	GPTIMER0EN	GPTIMER0定时器时钟使能 由软件置1或清0。 0: 关闭GPTIMER0定时器时钟 1: 开启 GPTIMER0 定时器时钟
20:18	保留	必须保持复位值。
17	TIMER2EN	TIMER2定时器时钟使能 由软件置1或清0。 0: 关闭TIMER2定时器时钟 1: 开启 TIMER2 定时器时钟
16	TIMER1EN	TIMER1定时器时钟使能 由软件置1或清0。 0: 关闭TIMER1定时器时钟 1: 开启 TIMER1 定时器时钟
15:14	保留	必须保持复位值。
13	TIMER7EN	TIMER7定时器时钟使能 由软件置1或清0。 0: 关闭TIMER7定时器时钟 1: 开启 TIMER7 定时器时钟
12	SPIEN	SPI时钟使能 由软件置1或清0。 0: 关闭SPI时钟 1: 开启 SPI 时钟
11	TIMER0EN	TIMER0定时器时钟使能 由软件置1或清0。 0: 关闭TIMER0定时器时钟 1: 开启 TIMER0 定时器时钟
10	保留	必须保持复位值。
9	ADC2EN	ADC2接口时钟使能 由软件置1或清0。 0: 关闭ADC2接口时钟 1: 开启 ADC2 接口时钟
8	ADC0EN	ADC0接口时钟使能 由软件置1或清0。 0: 关闭ADC0接口时钟 1: 开启 ADC0 接口时钟
7:2	保留	必须保持复位值。
1	CMPEN	比较器时钟使能

由软件置 1 或清 0。

0: 关闭比较器时钟

1: 开启比较器时钟

0	CFGEN	系统配置时钟使能 由软件置 1 或清 0。 0: 关闭系统配置时钟 1: 开启系统配置时钟

4.3.8. APB1 使能寄存器 (RCU_APB1EN)

地址偏移: 0x1C

复位值: 0x0000 0000

注意: 在重写此寄存器之前, 需将 SYSCFG.PRCFG1 位设置为 1 (写使能)。

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	保留			CFMUEN	保留		I2CEN	保留	UART3E N	UART2E N	UART1E N	UART0E N	
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			WWDGT EN	保留							CPTIMER WEN	保留		CPTIMER 1EN	CPTIMER 0EN	
			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 1 或清 0。 0: 关闭 DAC 时钟 1: 开启 DAC 时钟
28	PMUEN	电源接口时钟使能 由软件置1或清0。 0: 关闭电源接口时钟 1: 开启电源接口时钟
27:25	保留	必须保持复位值。
24	CFMUEN	CFMU时钟使能 由软件置1或清0。 0: 关闭CFMU时钟 1: 开启 CFMU 时钟
23:22	保留	必须保持复位值。

21	I2CEN	I2C时钟使能 由软件置1或清0。 0: 关闭I2C时钟 1: 开启 I2C 时钟
20	保留	必须保持复位值。
19	UART3EN	UART3时钟使能 由软件置1或清0。 0:关闭UART3时钟 1:开启 UART3 时钟
18	UART2EN	UART2时钟使能 由软件置1或清0。 0:关闭UART2时钟 1:开启 UART2 时钟
17	UART1EN	UART1时钟使能 由软件置1或清0。 0:关闭UART1时钟 1:开启 UART1 时钟
16	UART0EN	UART0时钟使能 由软件置1或清0。 0:关闭UART0时钟 1:开启 UART0 时钟
15:13	保留	必须保持复位值。
12	WWDGTEN	窗口看门狗定时器时钟使能 由软件置1或清0。 0: 关闭窗口看门狗定时器时钟 1: 开启窗口看门狗定时器时钟
11:5	保留	必须保持复位值。
4	CPTIMERWEN	CPTIMERW定时器时钟使能 由软件置1或清0。 0: 关闭CPTIMERW定时器时钟 1: 开启 CPTIMERW 定时器时钟
3:2	保留	必须保持复位值。
1	CPTIMER1EN	CPTIMER1定时器时钟使能 由软件置1或清0。 0: 关闭CPTIMER1定时器时钟 1: 开启 CPTIMER1 定时器时钟
0	CPTIMER0EN	CPTIMER0定时器时钟使能

由软件置1或清0。

0: 关闭CPTIMER0定时器时钟

1: 开启 CPTIMER0 定时器时钟

4.3.9. 复位源/时钟寄存器（RCU_RSTSCK）

地址偏移：0x24

复位值：0x0C80 0000，除复位标志外由系统复位清除，复位标志只能由电源复位清除

注意：在重写此寄存器之前，需将 SYSCFG.PRCFG1 位设置为 1（写使能）。

该寄存器可以按字节（8位）、半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LP RSTF	WWDGT RSTF	FWDGT RSTF	SW RSTF	POR RSTF	EP RSTF	OBL RSTF	RSTFC	V11 RSTF	LOP RSTF	LOH RSTF	LVD2RST F	LVD1RST F	LOCKUP RSTF	保留	LVD0RST F
r	r	r	r	r	r	r	rw	r	r	r	r	r	r		r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	LPRST EN	LOHRST EN	保留	LOCKUP RSTEN	保留								IRC32K STB	IRC32K EN	
	rw	rw		rw									r	rw	

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

27	PORRSTF	电源复位标志位 电源复位发生时由硬件置1。 由软件通过写1到RSTFC位来清除该位。 0: 无电源复位发生 1: 发生电源复位
26	EPRSTF	外部引脚复位标志位 当有外部引脚复位发生时由硬件置1。 由软件通过写1到RSTFC位来清除该位。 0: 无外部引脚复位发生 1: 发生外部引脚复位
25	OBLRSTF	选项字节装载器复位标志位 选项字节装载器装载字节时由硬件置1。 由软件通过写1到RSTFC位来清除该位。 0: 无选项字节装载器复位发生 1: 发生选项字节装载器复位
24	RSTFC	清除复位标志位 由软件置1来清除所有复位标志位。 0: 无作用 1: 清除复位标志位
23	V11RSTF	1.1V域电源复位标志位 当有1.1V域电源复位发生时由硬件置1。 由软件通过写1到RSTFC位来清除该位。 0: 无1.1V域电源复位发生 1: 发生 1.1V 域电源复位
22	LOPRSTF	PLL 丢失错误复位标志 当产生 PLL 丢失错误复位时，由硬件置位。 由软件通过写1到RSTFC位来清除该位。 0: 没有 PLL 丢失错误复位 1: PLL 丢失错误复位
21	LOHRSTF	HXTAL 丢失错误复位标志 当产生 HXTAL 丢失错误复位时，由硬件置位。 由软件通过写1到RSTFC位来清除该位。 0: 没有 HXTAL 丢失错误复位 1: HXTAL 丢失错误复位
20	LVD2RSTF	低电压检测 2 复位标志 当产生低电压检测 2 复位时，由硬件设置。 通过向 RSTFC 位写入 1 来复位。 0: 没有产生低电压检测复位 1: 产生了低电压检测复位

19	LVD1RSTF	低电压检测 1 复位标志 当产生低电压检测 1 复位时，由硬件设置。 通过向 RSTFC 位写入 1 来复位。 0: 没有产生低电压检测复位 1: 产生了低电压检测复位
18	LOCKUPRSTF	CPU 锁死错误复位标志 发生 CPU 锁死错误复位时，由硬件置位。 由软件通过写 1 到 RSTFC 位来清除该位。 0: 未产生 CPU 锁死错误复位 1: 产生 CPU 锁死错误复位
17	保留	必须保持复位值。
16	LVD0RSTF	低电压检测 0 复位标志 当产生低电压检测 0 复位时，由硬件设置。 通过向 RSTFC 位写入 1 来复位。 0: 没有产生低电压检测复位 1: 产生了低电压检测复位
15	保留	必须保持复位值。
14	LOPRSTEN	PLL 丢失复位使能 0: 如果 PLLMIE 为 1，则当检测到错误时，PLL 检测器会生成中断 1: 当检测到错误时，PLL 检测器产生复位
13	LOHRSTEN	HXTAL 丢失复位使能 0: 无复位产生 1: 当检测到错误时，HXTAL 检测器产生复位
12:11	保留	必须保持复位值。
10	LOCKUPRSTEN	CPU 锁死复位使能 0: 无复位产生 1: CPU 锁死产生复位
9:2	保留	必须保持复位值。
1	IRC32KSTB	IRC32K 时钟稳定状态位 该位由硬件置 1 指示 IRC32K 输出时钟是否稳定待用。 0: IRC32K 时钟未稳定 1: IRC32K 时钟已稳定
0	IRC32KEN	IRC32K 时钟使能 软件置 1 和清 0。 0: 关闭 IRC32K 时钟 1: 开启 IRC32K 时钟

地址偏移: 0x28
复位值: 0x0000 0000

注意：在重写此寄存器之前，需将 SYSCFG.PRCFG1 位设置为 1（写使能）。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		TMURST	SVPWMR ST	GTOCRS T	POCRST	保留		PNRST	PGRST	PFRST	PERST	PDRST	PCRST	PBRST	PARST
		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
保留									CRCRST	保留		DMAMUX RST	保留	DMA1RS T	DMA0RS T
									rw			rw			rw

位/位域	名称	描述
31:30	保留	必须保持复位值。
29	TMURST	TMU复位 由软件置1或清0。 0: 无作用 1: 复位 TMU
28	SVPWMRST	SVPWM复位 由软件置1或清0。 0: 无作用 1: 复位 SVPWM
27	GTOCRST	GTOC复位 由软件置1或清0。 0: 无作用 1: 复位 GTOC
26	POCRST	POC复位 由软件置1或清0。 0: 无作用 1: 复位 POC
25:24	保留	必须保持复位值。
23	PNRST	GPION复位 由软件置1或清0。 0: 无作用 1: 复位 GPION
22	PGRST	GPIOG复位

		由软件置1或清0。 0: 无作用 1: 复位 GPIOG
21	PFRST	GPIOF复位 由软件置1或清0。 0: 无作用 1: 复位 GPIOF
20	PERST	GPIOE复位 由软件置1或清0。 0: 无作用 1: 复位 GPIOE
19	PDRST	GPIOD复位 由软件置1或清0。 0: 无作用 1: 复位 GPIOD
18	PCRST	GPIOC复位 由软件置1或清0。 0: 无作用 1: 复位 GPIOC
17	PBRST	GPIOB复位 由软件置1或清0。 0: 无作用 1: 复位 GPIOB
16	PARST	GPIOA复位 由软件置1或清0。 0: 无作用 1: 复位 GPIOA
15:7	保留	必须保持复位值。
6	CRCRST	CRC复位 由软件置1或清0。 0: 无作用 1: 复位 CRC
5:4	保留	必须保持复位值。
3	DMAMUXRST	DMAMUX 复位 由软件置 1 或清 0。 0: 无作用 1: 复位 DMAMUX

2	保留	必须保持复位值。
1	DMA1RST	DMA1 复位 由软件置 1 或清 0。 0: 无作用 1: 复位 DMA1
0	DMA0RST	DMA0 复位 由软件置 1 或清 0。 0: 无作用 1: 复位 DMA0

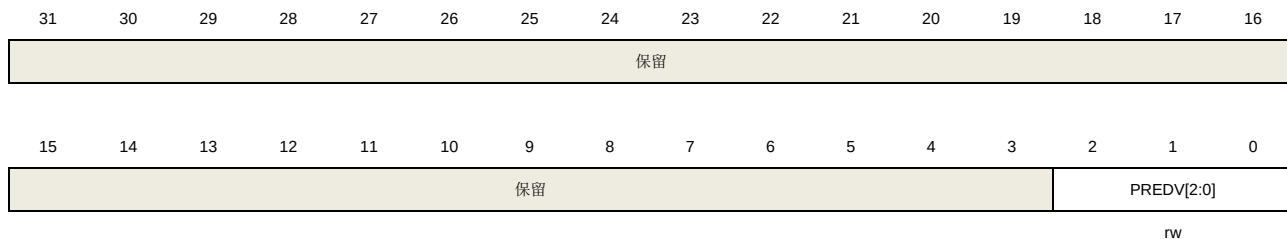
4.3.11. 配置寄存器 1 (RCU_CFG1)

地址偏移: 0x2C

复位值: 0x0000 0000

注意: 在重写此寄存器之前, 需将 SYSCFG.PRCFG0 位设置为 1 (写使能)。

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
31:3	保留	必须保持复位值。
2:0	PREDV[2:0]	CK_HXTAL 作为 PLL 输入源分频因子 由软件置 1 或清 0。这些位仅能在 PLL 关闭时改写。 CK_HXTAL 时钟分频因子为 (PREDV + 1)。 000: PLL的输入, 不分频 001: PLL的输入2分频 010: PLL的输入3分频 011: PLL的输入4分频 100: PLL的输入5分频 101: PLL的输入6分频 110: PLL的输入7分频 111: PLL的输入8分频

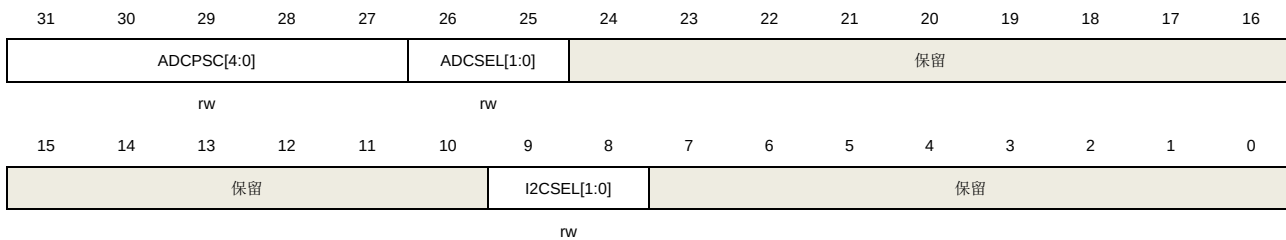
4.3.12. 配置寄存器 2 (RCU_CFG2)

地址偏移: 0x30

复位值: 0x1800 0000

注意：在重写此寄存器之前，需将 SYSCFG.PRCFG0 位设置为 1（写使能）。

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
31:27	ADCPSC[4:0]	ADC 时钟预分频器选择 由软件置1或清0。 注意： 这些位不能设置为“11111”。 $CK\_ADC = CK\_ADCPRE / (ADCPSC + 2)$
29:25	ADCSEL[1:0]	CK_ADCPRE 时钟源选择 由软件置 1 或清 0。 00: CK_ADCPRE 选择 CK_PCLK2 01: CK_ADCPRE 选择 CK_HXTAL 10: CK_ADCPRE 选择 CK_PLL 11: CK_ADCPRE 选择 CK_IRC32M
24:10	保留	必须保持复位值。
9:8	I2CSEL[1:0]	CK_I2C 时钟源选择 由软件置 1 或清 0。 00: CK_I2C 选择 CK_PCLK1 01: CK_I2C 选择 CK_PLL 10: CK_I2C 选择 CK_IRC32M 11: CK_I2C 选择 CK_HXTAL
7:0	保留	必须保持复位值。

5. 时钟频率测量单元（CFMU）

5.1. 概述

时钟频率测量单元提供时钟频率精度测量功能。使用参考时钟作为时间基准，对被测时钟的脉冲进行计数，然后通过计数结果判断被测时钟的精度是否在允许范围内。当时钟频率精度测量完成或脉冲计数不在允许范围内时，将产生中断请求。

5.2. 特性

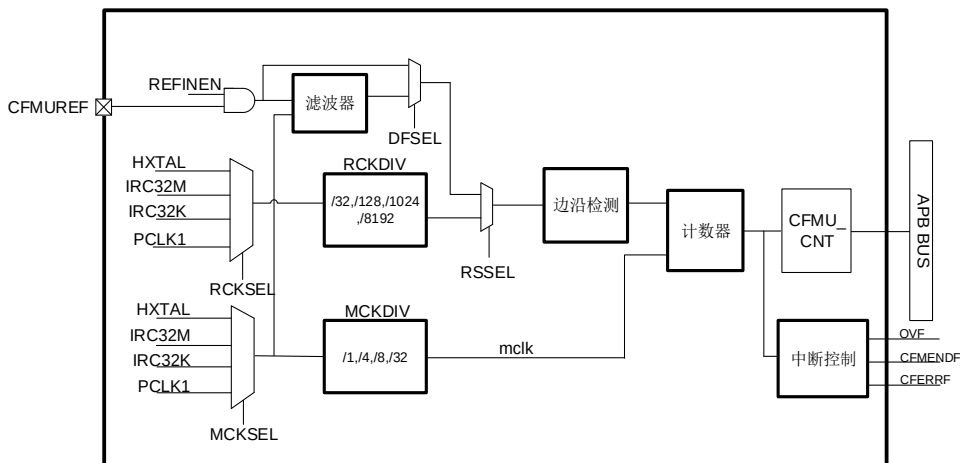
- 不同目标时钟的时钟频率测量：HXTAL、IRC32M、IRC32K或PCLK1；
- 可选的参考时钟：外部时钟输入至CFMUREF引脚，HXTAL，IRC32M，IRC32K或PCLK1；
- CFMUREF引脚上的数字滤波器功能；
- 中断和标志位用于指示时钟频率测量状态：时钟测量结束、时钟频率错误或溢出。

5.3. 功能概述

5.3.1. CFMU 框图

CFMU 的块图如[图5-1. CFMU 框图](#)所示。两个时钟选择器分别用于测量目标和参考时钟，并且有两个相应的时钟分频器。边缘检测功能可以通过 CFMU_CTL 寄存器中的 EDGSEL 位配置。CFMU 可以在中断控制单元中生成不同的中断请求。

图5-1. CFMU框图

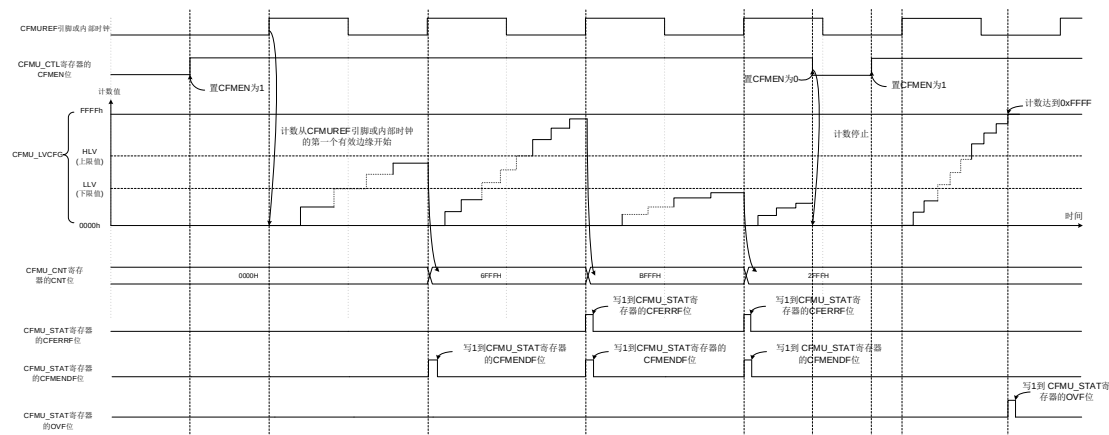


5.3.2. 时钟频率测量

通过 CFMUREF 引脚输入的时钟频率或由 CFMU_CTL 寄存器中的 RCKSEL 位和 RSSEL 位配置的内部时钟，可以通过时钟频率精度测量单元进行测量。[图 5-2. 时钟频率精度测量](#)显示

了时钟频率精度测量单元的操作。

图5-2. 时钟频率精度测量



注意：由于第一次使能 CFMU 时，有可能采样的不是一个完整的参考时钟周期，所以需要忽略第一次 CFMU_CNT 值和时钟频率错误标志。

5.3.3. CFMUREF 引脚上的数字滤波器功能

在 CFMUREF 引脚上有一个数字滤波器，这个功能可以通过 CFMU_CTL 寄存器中的 DFSEL 位来配置。

5.3.4. CFMU 中断

CFMU 生成三种类型的中断请求。

- 频率误差中断
- 测量结束中断
- 溢出中断

当产生中断时，相应的状态标志变为 1，详细信息请参考[表 5-1. CFMU 中断请求](#)。

表 5-1. CFMU 中断请求

标志	描述	清除方法	中断使能位
CFERRF	时钟频率错误标志	设置 CFMU_CTL 寄存器的 CFERRCLR 位	CFERRIE
CFMENDF	时钟频率测量结束标志	设置 CFMU_CTL 寄存器的 CFMENDFCLR 位	CFMENDIE
OVF	计数器溢出标志	设置 CFMU_CTL 寄存器的 OVFCLR 位	OVIE

5.4. 寄存器定义

CFMU 基地址: 0x4000 C800

5.4.1. 控制寄存器 (CFMU_CTL)

地址偏移: 0x00

重置值: 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	OVFCLR	CFMEND FCLR	CFERRC LR	保留	OVIE	CFMENDI E	CFERRIE	DFSEL[1:0]	RCKDIV[1:0]	MCKSEL[2:0]	RSSEL				
	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
EDGSEL[1:0]	MCKDIV[1:0]	RCKSEL[2:0]	REFINEN	保留							CFMEN				
rw	rw	rw	rw								rw				

位/位域	名称	描述
31	保留	必须保持复位值。
30	OVFCLR	溢出标志清除 当向该位写入 1 时, OVF 标志被清除。该位读出为 0。
29	CFMENDFCLR	时钟频率测量结束标志清除 当向该位写入 1 时, CFMENDF 标志被清除。该位读出为 0。
28	CFERRCLR	时钟频率错误标志清除 当向该位写入 1 时, CFERR 标志被清除。该位读出为 0。
27	保留	必须保持复位值。
26	OVIE	溢出中断使能 0: 中断禁能 1: 中断使能
25	CFMENDIE	时钟频率测量结束中断使能 0: 中断禁能 1: 中断使能
24	CFERRIE	时钟频率错误中断使能 0: 中断禁能 1: 中断使能
23:22	DFSEL[1:0]	数字滤波器选择 这些位配置数字滤波器的采样时钟。 00: 数字滤波器禁用

		01: 选择频率测量时钟除以 1 10: 选择频率测量时钟除以 4 11: 选择频率测量时钟除以 16
21:20	RCKDIV[1:0]	参考时钟频率分频器 这些位配置参考时钟频率的分频比率。 00: 时钟频率除以 32 01: 时钟频率除以 128 10: 时钟频率除以 1024 11: 时钟频率除以 8192
19:17	MCKSEL[2:0]	测量时钟选择 000: 选择外部 HXTAL 振荡器时钟 010: 选择内部 32MHz RC 振荡器时钟 011: 选择内部 32K RC 振荡器时钟 100: 选择外设模块时钟 1 (PCLK1) 其他: 保留
16	RSSEL	参考信号选择 0: 选择 CFMUREF 引脚输入 1: 选择内部时钟
15:14	EDGSEL[1:0]	边沿选择 00: 选择上升沿 01: 选择下降沿 10: 上升沿和下降沿均被选择 11: 保留
13:12	MCKDIV[1:0]	测量时钟频率分频器 这些位配置测量时钟频率的分频比率。 00: 时钟频率除以 1 01: 时钟频率除以 4 10: 时钟频率除以 8 11: 时钟频率除以 32
11:9	RCKSEL[2:0]	参考时钟选择 000: 选择外部 HXTAL 振荡器时钟 010: 选择内部 32MHz RC 振荡器时钟 011: 选择内部 32K RC 振荡器时钟 100: 选择外设模块时钟 1 (PCLK1) 其他: 保留
8	REFINEN	CFMUREF 引脚输入使能 0: CFMUREF 引脚输入禁能 1: CFMUREF 引脚输入使能
7:1	保留	必须保持复位值。

0	CFMEN	时钟频率测量使能 0: 时钟频率测量禁能 1: 时钟频率测量使能

5.4.2. 状态寄存器（CFMU_STAT）

地址偏移：0x04

重置值：0x0002 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													CFERRF	CFMEND F	OVF
													r	r	r

位/位域	名称	描述
31:3	保留	必须保持复位值。
2	CFERRF	时钟频率错误标志 0: 时钟频率在设定范围内 1: 时钟频率不在设定范围内
1	CFMENDF	时钟频率测量结束标志 0: 时钟频率测量正在进行中 1: 时钟频率测量已经结束
0	OVF	计数器溢出标志 0: 计数器没有溢出 1: 计数器已经溢出

5.4.3. 电流限制值配置寄存器（CFMU_LVCFG）

地址偏移：0x08

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

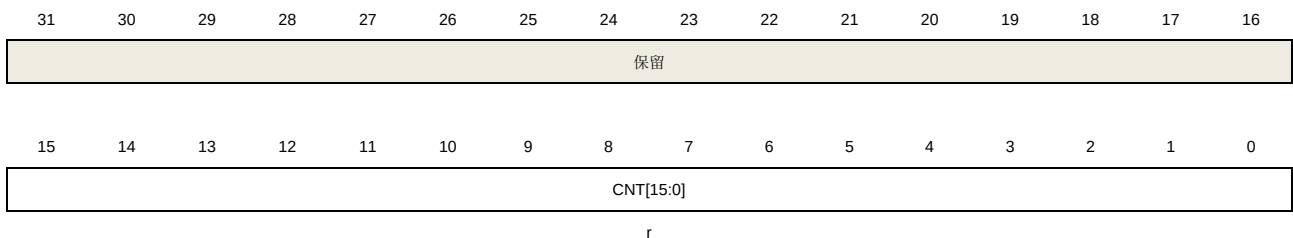
位/位域	名称	描述
31:16	HLV[15:0]	上限值 当时钟频率大于这个值时，会检测到频率误差。 当 CFMEN 位为 0 时，这些位可以被写入
15:0	LLV[15:0]	下限值 当时钟频率低于这个值时，会检测到频率误差。 当 CFMEN 位为 0 时，这些位可以被写入。

5.4.4. 计数器值寄存器（CFMU_CNT）

地址偏移：0x0C

重置值：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	CNT[15:0]	计数器值 这个位字段指示当前的计数器值。它在 CFMENDF 位被设置后更新。

6. 中断/事件控制器（EXTI）

6.1. 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 25 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、下降沿触发和双边沿触发。EXTI 中的每一个边沿检测电路都可以独立配置和屏蔽。

6.2. 主要特性

- Cortex®-M33 系统异常；
- 多达 93 种可屏蔽的外设中断；
- 4 位中断优先级配置位，可配置 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 25 个相互独立的边沿检测电路；
- 3 种触发类型：上升沿触发，下降沿触发和任意沿触发；
- 软件中断或事件触发；
- 可配置的触发源。

6.3. 中断功能描述

Arm Cortex®-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 6-1. Cortex®-M33 中的 NVIC 异常类型](#)和[表 6-2. 中断向量表](#)列出了 Cortex®-M33 中的 NVIC 异常类型。

表 6-1. 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
复位	1	-3	0x0000_0004	复位
NMI	2	-2	0x0000_0008	不可屏蔽中断
硬件故障	3	-1	0x0000_000C	各种硬件级别的故障

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

表 6-2. 中断向量表

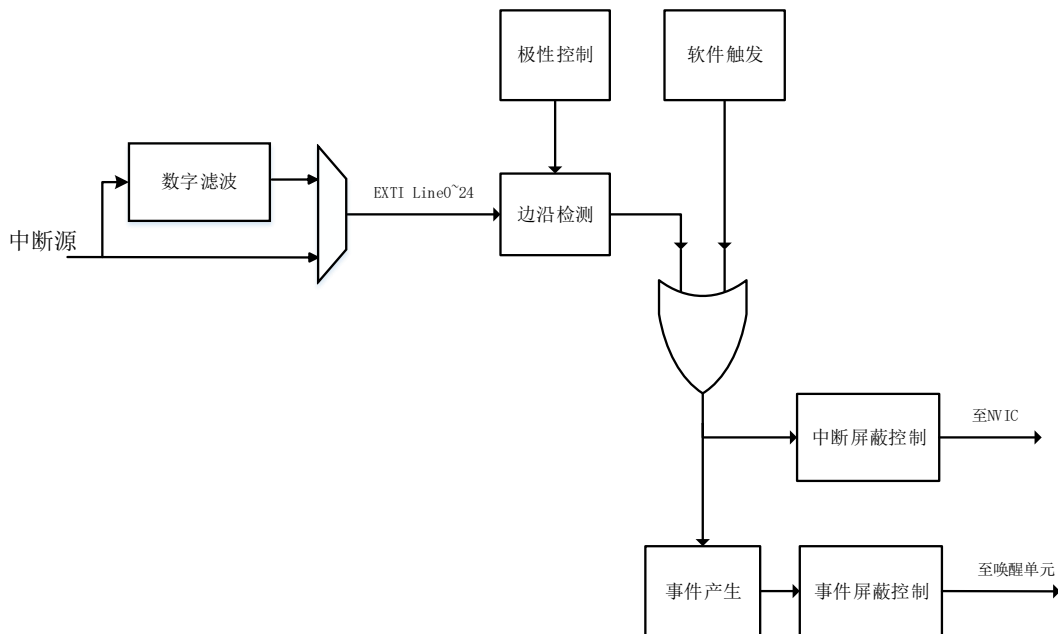
中断编号	向量编号	外设中断描述	向量地址
IRQ 0	16	窗口看门狗中断	0x0000_0040
IRQ 1	17	连接到 EXTI 线的 LVD1 中断	0x0000_0044
IRQ 2	18	连接到 EXTI 线的 LVD2 中断	0x0000_0048
IRQ 3	19	保留	0x0000_004C
IRQ 4	20	FMC 全局中断	0x0000_0050
IRQ 5	21	RCU 中断	0x0000_0054
IRQ 6	22	EXTI 线 0 中断	0x0000_0058
IRQ 7	23	EXTI 线 1 中断	0x0000_005C
IRQ 8	24	EXTI 线 2 中断	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	保留	0x0000_0084
IRQ 18	34	ADC0 全局中断	0x0000_0088
IRQ 19	35	CAN 发送中断	0x0000_008C
IRQ 20	36	CAN 接收 0 中断	0x0000_0090
IRQ 21	37	CAN 接收 1 中断	0x0000_0094
IRQ 22	38	CAN 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

中断编号	向量编号	外设中断描述	向量地址
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	GPTIMER0 全局中断	0x0000_00B8
IRQ 31	47	I2C 事件中断	0x0000_00BC
IRQ 32	48	I2C 错误中断	0x0000_00C0
IRQ 33-34	49-50	保留	0x0000_00C4- 0x0000_00C8
IRQ 35	51	SPI 全局中断	0x0000_00CC
IRQ 36	52	保留	0x0000_00D0
IRQ 37	53	UART0 全局中断	0x0000_00D4
IRQ 38	54	UART1 全局中断	0x0000_00D8
IRQ 39	55	保留	0x0000_00DC
IRQ 40	56	EXTI 线[15:10]中断	0x0000_00E0
IRQ 41-42	57-58	保留	0x0000_00E4- 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	TMU 全局中断	0x0000_00FC
IRQ 48-49	64-65	保留	0x0000_0100- 0x0000_0104
IRQ50	66	GPTIMER1 全局中断	0x0000_0108
IRQ51	67	保留	0x0000_010C
IRQ52	68	UART2 全局中断	0x0000_0110
IRQ53	69	UART3 全局中断	0x0000_0114
IRQ54	70	CPTIMER0 全局中断	0x0000_0118
IRQ55	71	CPTIMER1 全局中断	0x0000_011C
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	DMA1 通道 5 全局中断	0x0000_0134
IRQ62	78	DMAMUX 溢出中断	0x0000_0138
IRQ63	79	CPTIMERW 全局中断	0x0000_013C
IRQ64	80	保留	0x0000_0140
IRQ65	81	CFMU 中断	0x0000_0144

中断编号	向量编号	外设中断描述	向量地址
IRQ66	82	连接到 EXTI 线的 I2C 唤醒中断	0x0000_0148
IRQ67	83	连接到 EXTI 线的 FWDGT 中断	0x0000_014C
IRQ68-69	84-85	保留	0x0000_0150- 0x0000_0154
IRQ70	86	CMP0 中断	0x0000_0158
IRQ71	87	CMP1 中断	0x0000_015C
IRQ72	88	CMP2 中断	0x0000_0160
IRQ73	89	CMP3 中断	0x0000_0164
IRQ74	90	保留	0x0000_0168
IRQ75	91	ADC2 全局中断	0x0000_016C
IRQ76	92	保留	0x0000_0170
IRQ77	93	POC 全局中断	0x0000_0174
IRQ79	95	GTOC0 全局中断	0x0000_017C
IRQ80	96	GTOC1 全局中断	0x0000_0180
IRQ81	97	GTOC2 全局中断	0x0000_0184
IRQ82	98	GTOC3 全局中断	0x0000_0188
IRQ83-84	99-100	保留	0x0000_018C- 0x0000_0190
IRQ85	101	连接 EXTI 线的 CMP0 中断	0x0000_0194
IRQ86	102	连接 EXTI 线的 CMP1 中断	0x0000_0198
IRQ87	103	连接 EXTI 线的 CMP2 中断	0x0000_019C
IRQ88	104	连接 EXTI 线的 CMP3 中断	0x0000_01A0
IRQ89-91	105-107	reserved	0x0000_01A4- 0x0000_01AC
IRQ92	108	SRAM ECC 全局中断	0x0000_01B0

6.4. 外部中断及事件(EXTI)框图

图 6-1. EXTI 框图



GD32M531 EXTI 拥有一个数字滤波器，可以通过过滤 EXTI 外部源（来自 GPIO 引脚和 NMI 引脚的信号，即 EXTI 线 0 到 15 和线 24）来降低噪声。配置 EXTI_DFEN 寄存器以使能或禁用数字滤波器。在使能数字滤波器后，只有当数字滤波器连续三次从 EXTI 外部源采样到相同电平时，EXTI 触发信号才被视为有效。数字滤波器的采样时钟可以通过 EXTI_SCS 寄存器进行配置。

注意：当系统处于深度睡眠模式或待机模式时，数字滤波器不能使用。

6.5. 外部中断及事件功能概述

EXTI 包含多达 25 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 9 根线，具体细节参考[表 6-3. EXTI 触发源](#)。通过配置 GPIO 模块的 SYSCFG_EXTISSx 寄存器，所有的 GPIO 管脚都可以被选作 EXTI 的触发源，具体细节请参考[系统配置寄存器](#)。

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，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_FTEV 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 FTEVx 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_EVEN 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_EVEN 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 6-3. EXTI 触发源

EXTI 线编号	触发源
0	PA8 / PC10 / PN2 / PC8
1	PA9 / PC11 / PE13 / PC9
2	PA0 / PC12 / PF14 / PG12
3	PF12
4	PA1 / PE14 / PF9 / PF13
5	PD8 / PG14 / PN7
6	PB1 / PC6 / PD2 / PG13
7	PB0 / PC7 / PN5
8	PD4 / PE8 / PF8 / PG15
9	PD5 / PD9 / PE9 / PF11
10	PB2 / PC0 / PD10 / PE10 / PF10
11	PC1 / PD11 / PG11
12	PC2 / PD12 / PE12
13	PC3 / PD13
14	PB14 / PC4 / PD14 / PE11
15	PB15 / PC5
16	LVD0
17	LVD1
18	CMP0
19	CMP1

EXTI 线编号	触发源
20	CMP2
21	CMP3
22	FWDGT
23	I2C 唤醒
24	NMI (PN2)

6.6. EXTI 寄存器

EXTI 基地址: 0x4001 0400

6.6.1. 中断使能寄存器(EXTI_INTEN)

地址偏移: 0x00

复位值: 0x0100 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							INTEN24	INTEN23	INTEN22	INTEN21	INTEN20	INTEN19	INTEN18	INTEN17	INTEN16
							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
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
31:25	保留	必须保持复位值。
24:0	INTENx	中断使能位 x (x = 0...24) 0: 第 x 线中断被禁用 1: 第 x 线中断被使能

6.6.2. 事件使能寄存器(EXTI_EVEN)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								EVEN23	EVEN22	EVEN21	EVEN20	EVEN19	EVEN18	EVEN17	EVEN16
								rW	rW	rW	rW	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
31:24	保留	必须保持复位值。
23:0	EVENx	事件使能位 x (x = 0...23) 0: 第 x 线事件被禁用 1: 第 x 线事件被使能

6.6.3. 上升沿触发使能寄存器(EXTI_RTEN)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留							RTEN24	RTEN23	RTEN22	RTEN21	RTEN20	RTEN19	RTEN18	RTEN17	RTEN16
							rW	rW	rW	rW	W	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTEN15	RTEN14	RTEN13	RTEN12	RTEN11	RTEN10	RTEN9	RTEN8	RTEN7	RTEN6	RTEN5	RTEN4	RTEN3	RTEN2	RTEN1	RTEN0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

位/位域	名称	描述
31:25	保留	必须保持复位值。
24:0	RTENx	上升沿触发使能x (x = 0...24) 0: 第x线上升沿触发无效 1: 第x线上升沿触发有效 (中断/事件请求)

6.6.4. 下降沿触发使能寄存器(EXTI_FTEN)

地址偏移: 0x0C

复位值: 0x0100 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							FTEN24	FTEN23	FTEN22	FTEN21	FTEN20	FTEN19	FTEN18	FTEN17	FTEN16
							rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FTEN15	FTEN14	FTEN13	FTEN12	FTEN11	FTEN10	FTEN9	FTEN8	FTEN7	FTEN6	FTEN5	FTEN4	FTEN3	FTEN2	FTEN1	FTEN0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

位/位域	名称	描述
31:25	保留	必须保持复位值。
24:0	FTENx	下降沿触发使能 x (x = 0...24) 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)

6.6.5. 软件中断事件寄存器(EXTI_SWIEV)

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								SWIEV23	SWIEV22	SWIEV21	SWIEV20	SWIEV19	SWIEV18	SWIEV17	SWIEV16
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIEV15	SWIEV14	SWIEV13	SWIEV12	SWIEV11	SWIEV10	SWIEV9	SWIEV8	SWIEV7	SWIEV6	SWIEV5	SWIEV4	SWIEV3	SWIEV2	SWIEV1	SWIEV0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:25	保留	必须保持复位值。
24:0	SWIEVx	中断/事件软件触发 x (x = 0...23) 0: 禁用 EXTI 线 x 软件中断/事件请求 1: 激活 EXTI 线 x 软件中断/事件请求

6.6.6. 挂起寄存器(EXTI_PD)

地址偏移: 0x14

复位值: 0xFFFF FFFF

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								PD24	PD23	PD22	PD21	PD20	PD19	PD18	PD17
								rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位/位域	名称	描述
31:25	保留	必须保持复位值。
24:0	PDx	中断挂起状态 x (x = 0...24) 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。

6.6.7. 数字滤波使能寄存器(EXTI_DFEN)

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							DFEN24	保留							
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DFEN15	DFEN14	DFEN13	DFEN12	DFEN11	DFEN10	DFEN9	DFEN8	DFEN7	DFEN6	DFEN5	DFEN4	DFEN3	DFEN2	DFEN1	DFEN0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:25	保留	必须保持复位值。
24	DFENx	数字滤波使能 x (x = 24) 0: EXTI线x数字滤波禁能 1: EXTI 线 x 数字滤波使能
23:16	保留	必须保持复位值。
15:0	DFENx	数字滤波使能 x (x = 0...15) 0: EXTI线x数字滤波禁能 1: EXTI 线 x 数字滤波使能

6.6.8. 采样时钟选择寄存器(EXTI_SCS)

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														SCS[1:0]	
rw															

位/位域	名称	描述
31:2	保留	必须保持复位值。
1:0	SCS	选择采样时钟 00: 采样时钟是 PCLK 01: 采样时钟是 PCLK / 8 10: 采样时钟是 PCLK / 32 11: 采样时钟是 PCLK / 64。

7. 通用和备用输入/输出接口（GPIO 和 AFIO）

7.1. 简介

最多可支持53个通用I/O引脚（GPIO），分别为PA0~PA1，PA9，PB0~PB2，PB14~PB15，PC0~PC12，PD4~PD14，PE8~PE14，PF8~PF14，PG11，PG13，PG15，PN2，PN5，PN6，PN7。各片上设备用其来实现逻辑输入/输出功能。每个GPIO端口有相关的控制和配置寄存器以满足特定应用的需求。GPIO引脚上的外部中断在中断 / 事件控制器（EXTI）中有相关的控制和配置寄存器。

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

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

7.2. 主要特征

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

7.3. 功能说明

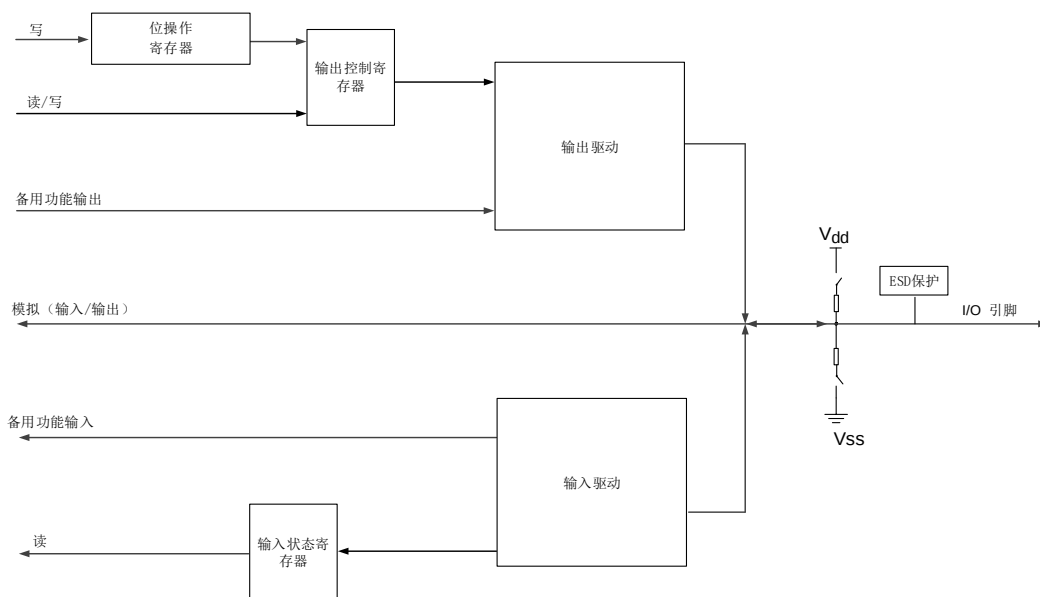
每个通用I/O端口都可以通过32位控制寄存器（GPIOx_CTL）配置为GPIO输入，GPIO输出，AF功能或模拟模式。引脚AFIO输入/输出是通过AFIO功能使能来选择。当端口配置为输出（GPIO输出或AFIO输出）时，可以通过GPIO输出模式寄存器（GPIOx_OMODE）配置为推挽或开漏模式。输出端口的最大速度可以通过GPIO输出速度寄存器（GPIOx_OSPD）配置。每个端口可以通过GPIO上/下拉寄存器（GPIOx_PUD）配置为浮空（无上拉或下拉），上拉或下拉功能。详情请见[表7-1. GPIO配置表](#)。

表 7-1. GPIO 配置表

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

图 7-1. GPIO 端口位的基本结构为标准 I/O 端口位的基本结构图。

图 7-1. GPIO 端口位的基本结构



7.3.1. GPIO 引脚配置

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

注意：在LQFP64中，将GPIOx_FUNCSEL寄存器中的PN7FUNC位设置为0以启用TSEL功能，并拉高PN7时，引脚PG11-PG15将用于实现JTAG/SWD功能。并且JTAG/串行线调试引脚处于上拉下拉模式时：

PG13: JTDI处于上拉模式

PG12: JTCK/SWCLK处于下拉模式

PG14: JTMS/SWDIO处于上拉模式

PG15: NJRST处于上拉模式

PG11: JTDO处于输出模式

调试引脚详细功能参考[14.2 JTAG/SW 功能说明](#)。

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AHB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，或用于翻转操作的GPIOx_TG）修改一位或几位，该过程仅需要一个最小的AHB写访问周期，而其他位不受影响。

7.3.2. 外部中断/事件线

只有在输入模式下配置，端口才能使用外部中断/事件线。

7.3.3. 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_AFSELY(y=0..1)），每个端口可以配置16个备用功能。端口备用功能分配的详细介绍见芯片数据手册。

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

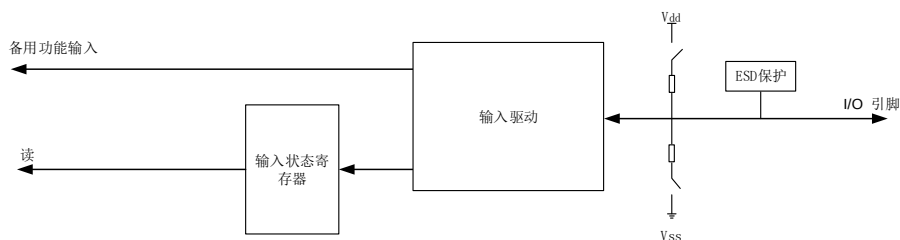
7.3.5. 输入配置

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AHB时钟周期都会被采样并存入端口输入状态寄存器；
- 输出缓冲器禁用。

[图 7-2. 输入配置的基本结构](#)显示 I/O 引脚的输入配置。

图 7-2. 输入配置的基本结构



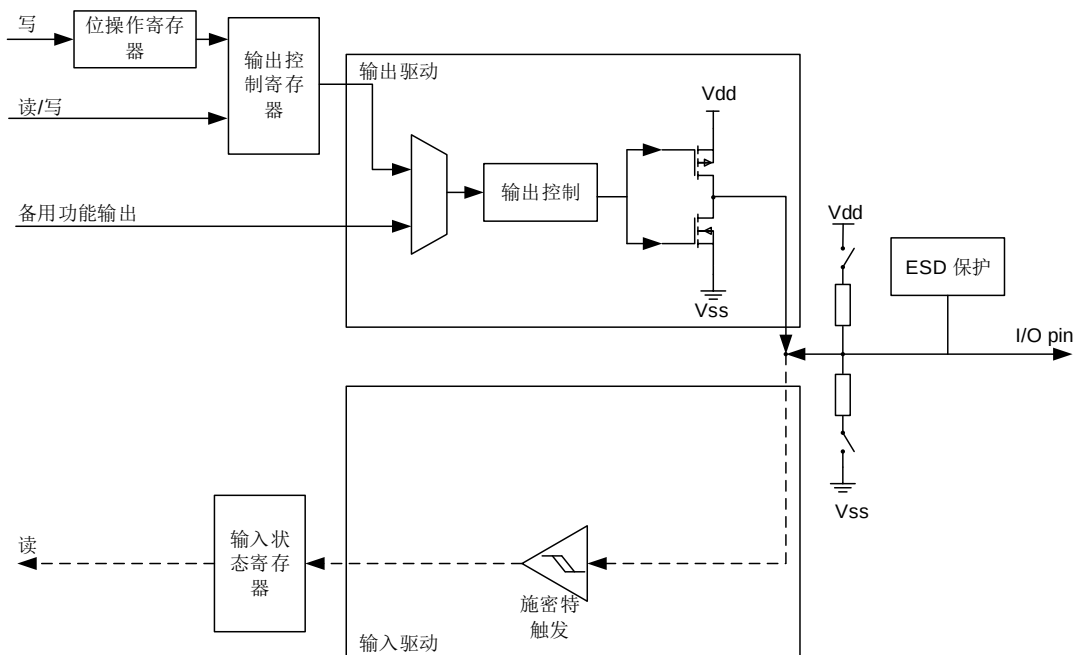
7.3.6. 输出配置

当GPIO配置为输出时：

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

图 7-3. 输出配置的基本结构是 I/O 端口的输出配置。

图 7-3. 输出配置的基本结构



7.3.7. 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读取为“0”。

图 7-4. 模拟配置的基本结构是 I/O 端口的模拟模式配置。

图 7-4. 模拟配置的基本结构



7.3.8. 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。

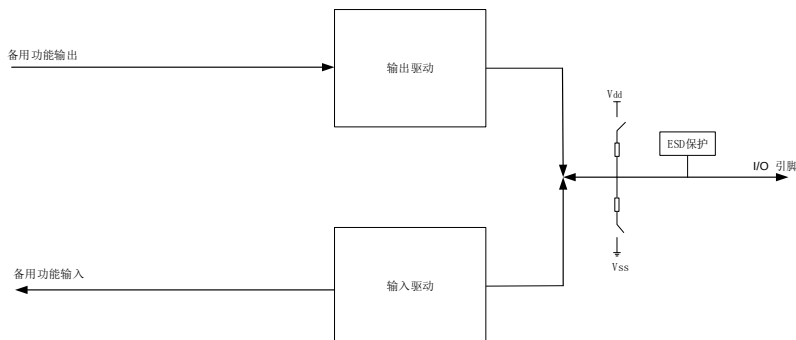
当引脚配置为备用功能时：

- 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 7-5. 备用功能配置的基本结构](#)是 I/O 端口备用功能配置图。

图 7-5. 备用功能配置的基本结构



7.3.9. 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、GPIOx_AFSELz (z=0, 1) 和GPIOx_HLD。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。

7.3.10. GPIO 单周期输出翻转功能

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

7.3.11. GPIO 引脚电平保持功能

通过将GPIOx_HLDCFG寄存器中对应的位写1，GPIO可以保持住I/O的输出电平。一旦开启该功能，即使GPIOx_OCTL寄存器中的值发生变化，GPIO的输出电平仍保持不变。仅当GPIOx_OCTL寄存器中的HLDUPD位被置位时，引脚输出电平将被更新为GPIOx_OCTL寄存器中的值。

7.4. GPIO 寄存器

GPIOA基地址: 0x4800 0000
 GPIOB基地址: 0x4800 0400
 GPIOC基地址: 0x4800 0800
 GPIOD基地址: 0x4800 0C00
 GPIOE基地址: 0x4800 1000
 GPIOF基地址: 0x4800 1400
 GPIOG基地址: 0x4800 1800
 GPION 基地址: 0x4800 4000 - 0x480143FF

7.4.1. 端口控制寄存器（GPIOx_CTL, x=A..G, N）

地址偏移: 0x00
 复位值: 0x0800 0000（端口 G）
 复位值: 0x0000 2020（端口 N）
 复位值: 0x0000 0000（其他端口）

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTL15[1:0]		CTL14[1:0]		CTL13[1:0]		CTL12[1:0]		CTL11[1:0]		CTL10[1:0]		CTL9[1:0]		CTL8[1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	
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。 参照CTL0[1:0]的描述
29:28	CTL14[1:0]	Pin 14配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
27:26	CTL13[1:0]	Pin 13配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
25:24	CTL12[1:0]	Pin 12配置位 该位由软件置位和清除。 参照CTL0[1:0]的描述
23:22	CTL11[1:0]	Pin 11配置位 该位由软件置位和清除。

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

7.4.2. 端口输出模式寄存器（GPIOx_OMODE, x=A..G, N）

地址偏移：0x04

复位值：0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OM15	OM14	OM13	OM12	OM11	OM10	OM9	OM8	OM7	OM6	OM5	OM4	OM3	OM2	OM1	OM0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	OM15	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: 输出推挽模式（复位值） 1: 输出开漏模式

7.4.3. 端口输出速度寄存器（GPIOx_OSPD, x=A..G, N）

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
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。 输出驱动能力控制 GPIO_N5T_HIA 类型为: PD9-PD14 / PE8-13 / PF13 / PG11 00: 低驱动 (IOL=5mA, 25Mhz) 01: 高驱动 (IOL=10mA, 50Mhz) 1x: 大电流驱动 (IOL=15mA) GPIO_5T类型为: PF8 / PF9 / PF10 00: 低驱动 (IOL=3mA, 25Mhz) 01: 高驱动 (IOL=6mA, 50Mhz) 1x: 保留 GPIO_N5T / GPIO_N5T_WKUP / GPIO_N5T_SE 类型为: 其他 00: 低驱动 (IOL=1mA, 25Mhz) 01: 高驱动 (IOL=2mA, 50Mhz) 1x: 保留

7.4.4. 端口上拉/下拉寄存器 (GPIOx_PUD, x=A..G, N)

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
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上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述

29:28	PUD14[1:0]	Pin 14上拉或下拉位

		该位由软件置位和清除。 参照PUD0[1:0]的描述
27:26	PUD13[1:0]	Pin 13上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
25:24	PUD12[1:0]	Pin 12上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
23:22	PUD11[1:0]	Pin 11上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
21:20	PUD10[1:0]	Pin 10上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
19:18	PUD9[1:0]	Pin 9上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
17:16	PUD8[1:0]	Pin 8上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
15:14	PUD7[1:0]	Pin 7上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
13:12	PUD6[1:0]	Pin 6上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
11:10	PUD5[1:0]	Pin 5上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
9:8	PUD4[1:0]	Pin 4上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
7:6	PUD3[1:0]	Pin 3上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
5:4	PUD2[1:0]	Pin 2上拉或下拉位 该位由软件置位和清除。

参照PUD0[1:0]的描述

3:2	PUD1[1:0]	Pin 1上拉或下拉位 该位由软件置位和清除。 参照PUD0[1:0]的描述
1:0	PUD0[1:0]	Pin 0上拉或下拉位 该位由软件置位和清除。 00: 悬空模式，无上拉和下拉（复位值） 01: 端口上拉模式 10: 端口下拉模式 11: 保留

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

地址偏移: 0x10

复位值: 0x0000 XXXX

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISTAT15	ISTAT14	ISTAT13	ISTAT12	ISTAT11	ISTAT10	ISTAT9	ISTAT8	ISTAT7	ISTAT6	ISTAT5	ISTAT4	ISTAT3	ISTAT2	ISTAT1	ISTAT0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

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

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

地址偏移: 0x14

复位值: 0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OCTL15	OCTL14	OCTL13	OCTL12	OCTL11	OCTL10	OCTL9	OCTL8	OCTL7	OCTL6	OCTL5	OCTL4	OCTL3	OCTL2	OCTL1	OCTL0

rW rW rW rW rW rW rW rW rW rW rW rW rW rW rW

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) 该位由软件置位和清除。 0: 引脚输出低电平 1: 引脚输出高电平

7.4.7. 端口位操作寄存器（GPIOx_BOP, x=A..G, N）

地址偏移: 0x18

复位值: 0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

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

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

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

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															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	锁定键 该位只能通过Lock Key写序列置位，始终可读。 0: GPIOx_LOCK寄存器和端口配置没有锁定 1: 直到下一次MCU复位前，GPIOx_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 y(y=0..15)的值必须保持。
15:0	LKy	端口锁定位y(y=0..15) 该位由软件置位和清除。 0: 端口配置没有锁定 1: 端口配置锁定

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

地址偏移: 0x20

复位值: 0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

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

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

		该位由软件置位和清除。 参照SEL0 [3:0]的描述
15:12	SEL3[3:0]	Pin 3选择备用功能 该位由软件置位和清除。 参照SEL0 [3:0]的描述
11:8	SEL2[3:0]	Pin 2选择备用功能 该位由软件置位和清除。 参照SEL0 [3:0]的描述
7:4	SEL1[3:0]	Pin 1选择备用功能 该位由软件置位和清除。 参照SEL0 [3:0]的描述
3:0	SEL0[3:0]	Pin 0选择备用功能 该位由软件置位和清除。 0000: 选择AF0功能（复位值） 0001: 选择AF1功能 0010: 选择AF2功能 0011: 选择AF3功能 ... 1111: 选择AF15功能

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

地址偏移：0x24

复位值：0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL15[3:0]				SEL14[3:0]				SEL13[3:0]				SEL12[3:0]			
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选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
27:24	SEL14[3:0]	Pin 14选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
23:20	SEL13[3:0]	Pin 13选择备用功能

		该位由软件置位和清除。 参照SEL8[3:0]的描述
19:16	SEL12[3:0]	Pin 12选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
15:12	SEL11[3:0]	Pin 1选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
11:8	SEL10[3:0]	Pin 10选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
7:4	SEL9[3:0]	Pin 9选择备用功能 该位由软件置位和清除。 参照SEL8[3:0]的描述
3:0	SEL8[3:0]	Pin 8选择备用功能 该位由软件置位和清除。 0000: 选择AF0功能（复位值） 0001: 选择AF1功能 0010: 选择AF2功能 0011: 选择AF3功能 ... 1111: 选择AF15功能

7.4.11. 位清除寄存器（GPIOx_BC，x=A..G，N）

地址偏移：0x28

复位值：0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

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

1: 清除相应的OCTLy位

7.4.12. 端口位翻转寄存器 (GPIOx_TG, x=A..G, N)

地址偏移: 0x2C

复位值: 0x0000 0000

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TG15	TG14	TG13	TG12	TG11	TG10	TG9	TG8	TG7	TG6	TG5	TG4	TG3	TG2	TG1	TG0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

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

7.4.13. 引脚电平保持配置寄存器 (GPIOx_HLDCFG, x=A..G, N)

地址偏移: 0x50

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
HEN15	HEN14	HEN13	HEN12	HEN11	HEN10	HEN9	HEN8	HEN7	HEN6	HEN5	HEN4	HEN3	HEN2	HEN1	HEN0
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	HENy	引脚电平保持使能位y (y=0..15) 该位由软件置位和清除。 0: 失能引脚电平保持功能 1: 使能引脚电平保持功能

7.4.14. 引脚功能选择寄存器（GPIOx_FUNCSEL, x=N）

地址偏移：0x60

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														PN7FUNC	PN6FUNC
														C	C
														rw	rw

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	PN7FUNC	PN7功能选择位 0: PN7选择为TSEL功能 1: PN7选择为通用输入输出引脚
0	PN6FUNC	PN6功能选择位 0: PN6选择为BOOT/One-line功能 1: PN6选择为通用输入输出引脚

7.4.15. 引脚电平保持控制寄存器（GPIOx_HLDCTL, x=N）

地址偏移：0x64

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															HLDUPD
															w

位/位域	名称	描述
31:1	保留	必须保持复位值。
0	HLDUPD	引脚电平保持更新位 0: 无影响 1: 引脚电平将更新为GPIOx_OCTL寄存器中的值。 注意： 该位仅能被设置为1，并会自动被硬件清零。

8. 端口输出控制器 (POC)

8.1. 简介

端口输出控制器 (POC) 可以禁能 TIMER 和 GPTIMER 通道引脚输出，并适用于各种场合。被禁能的引脚可以选择高阻 (Hi-Z) 输出或者 GPIO 模块控制输出。只有当这些引脚被复用于 TIMER / GPTIMER 通道输出功能时，POC 才能控制这些引脚的禁能。

8.2. 主要特征

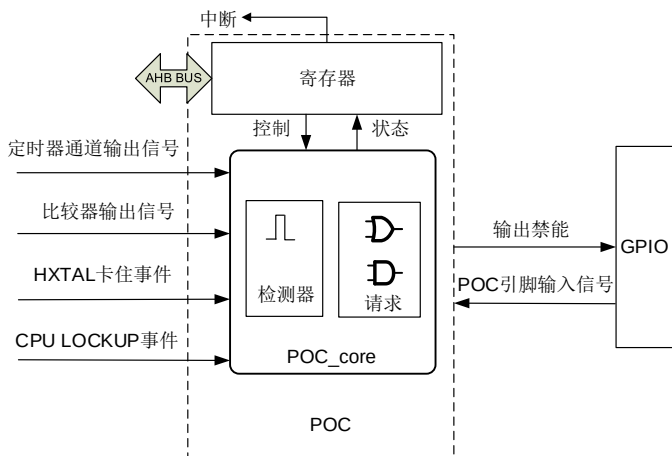
POC 的主要特征如下：

- POC_INn (n=0,1...5) 引脚输入检测：
 - 边沿检测；
 - 电平检测；
 - 可配置的采样时钟频率；
 - 可选的有效采样次数；
 - 输入反相控制；
- 引脚禁能请求源：
 - POC_INn (n=0,1...5) 引脚输入检测；
 - TIMER0/7 互补 PWM 通道输出检测；
 - 软件生成；
 - HXTAL 卡住事件；
 - CPU 内核锁定事件；
 - 比较器输出检测；
- 引脚禁能的目标定时器是可选的；
- 检测屏蔽功能：
 - POC_INn (n=0,1...5) 引脚输入检测屏蔽；
 - 比较器输出检测屏蔽；
- 中断生成：
 - POC_INn (n=0,1...5) 引脚输入有效信号；
 - TIMER0/7 互补 PWM 通道同时输出有效电平；

8.3. 功能描述

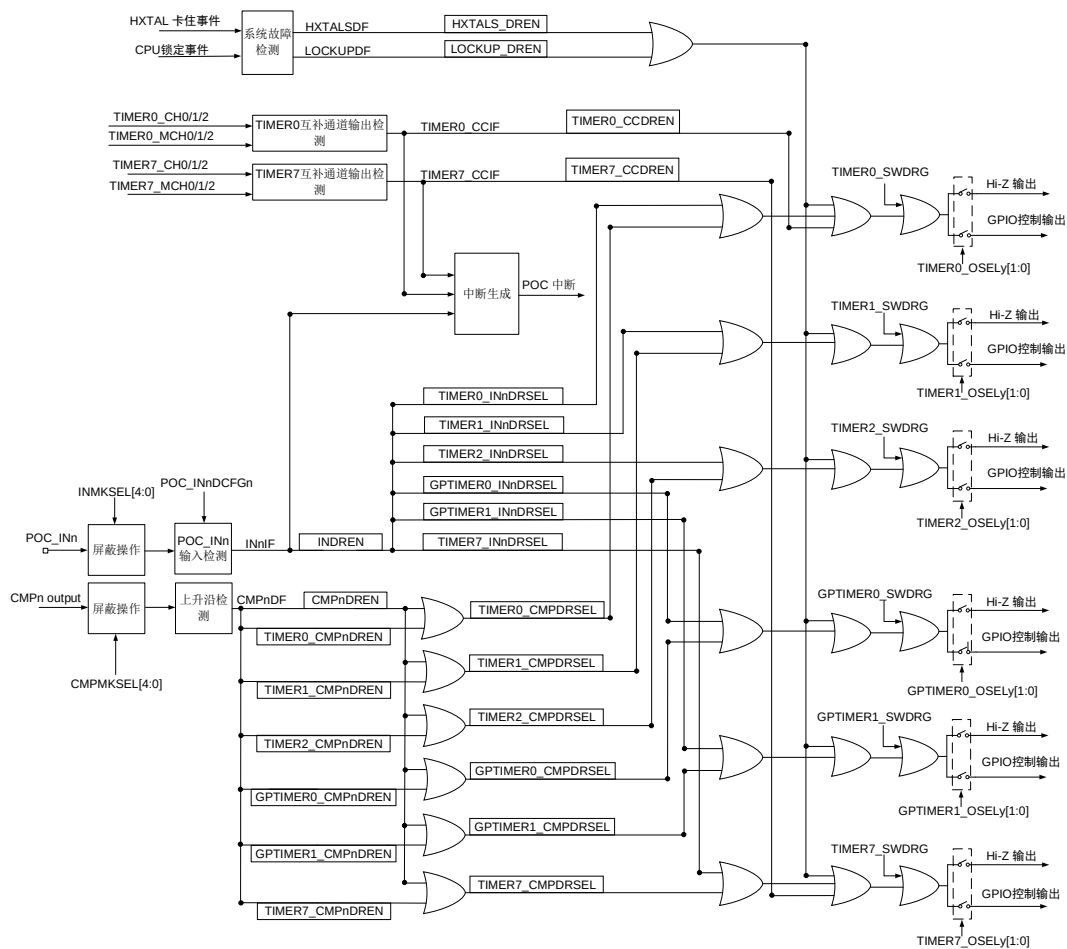
POC 模块框图展示如下：

图 8-1. POC 模块框图



引脚输出禁能框图展示如下：

图 8-2. TIMER 和 GPTIMER 通道引脚输出禁能



输出禁能的目标定时器和通道展示如下：

表 8-1. 输出禁能的目标定时器和通道

	TIMER				GPTIMER	
目标定时器	TIMER0	TIMER7	TIMER1	TIMER2	GPTIMER0	GPTIMER1
目标通道	TIMER0_CH0	TIMER7_CH0	TIMER1_	TIMER2_		
	TIMER0_CH1	TIMER7_CH1	CH0	CH0		
	TIMER0_CH2	TIMER7_CH2	TIMER1_	TIMER2_	GPTIMER0	GPTIMER1
	TIMER0_CH3	TIMER7_CH3	CH1	CH1	_CH0	_CH0
	TIMER0_MCH0	TIMER7_MCH0	TIMER1_	TIMER2_	GPTIMER0	GPTIMER1
	TIMER0_MCH1	TIMER7_MCH1	CH2	CH2	_CH1	_CH1
	TIMER0_MCH2	TIMER7_MCH2	TIMER1_	TIMER2_		
	TIMER0_MCH3	TIMER7_MCH3	CH3	CH3		

8.3.1. POC 时钟

POC 模块与 AHB 总线连接，时钟与 HCLK 同步。

8.3.2. 输出禁能功能

POC 输入检测

POC 模块有六个外部输入引脚。每个 POC_INn (n=0,1...5) 引脚的输入信号可以被用于边沿和电平检测。当 POC_INnDCFG (n=0,1...5) 寄存器的 INPL 位被设置为 0 时，则选择下降沿或者低电平作为有效检测信号。当 INPL 位被设置为 1 时，则进行上升沿或者高电平检测。

注意：在进行 POC 引脚输入检测的软件配置时，先配置 POC 检测相关功能，再配置 POC 引脚为对应的 GPIO 复用功能。

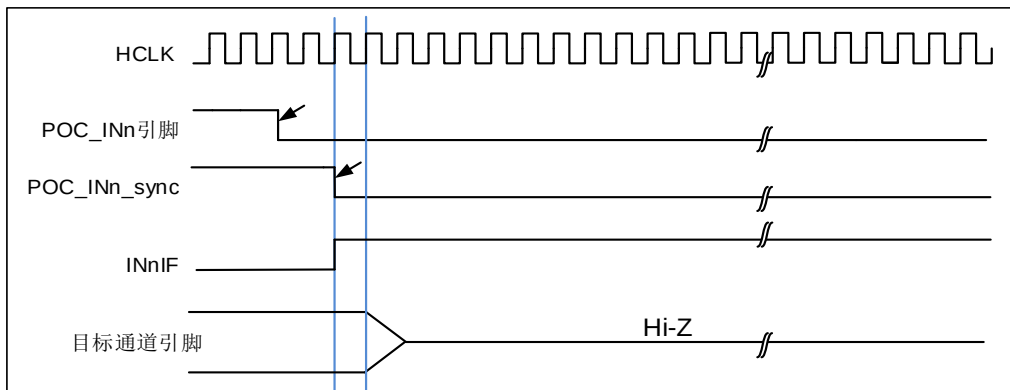
边沿检测模式

采用 HCLK 作为 POC_INn (n=0,1...5) 引脚输入信号的采样时钟。当 POC_INnDCFG (n=0,1...5) 寄存器的 INDMSEL[3:0] 位域被设置成 4'b0000 时，如果 POC_INn (n=0,1...5) 引脚输入了下降沿 (INPL=0) 或者上升沿 (POCPL=1) 信号，在信号被同步之后，则 POC_STAT0 寄存器的标志位 INnIF (n=0,1...5) 将会被硬件置起，且会生成一个禁能请求。

如果该禁能请求已经通过设置 POC_INnDCFG (n=0,1...5) 寄存器的 INDREN 位进行了使能，且利用 POC_REQSEL0/1 寄存器为禁能请求选择了目标定时器，那么，在一个 HCLK 时钟后，与目标定时器通道相连的引脚将会被禁能。被禁能的引脚可以通过设置 POC_ODMODE0/1 寄存器来输出高阻态。

通过清除中断标志位 INnIF (n=0,1...5)，被失能的引脚可以从禁能输出中恢复。

图 8-3. POC 引脚下降沿检测

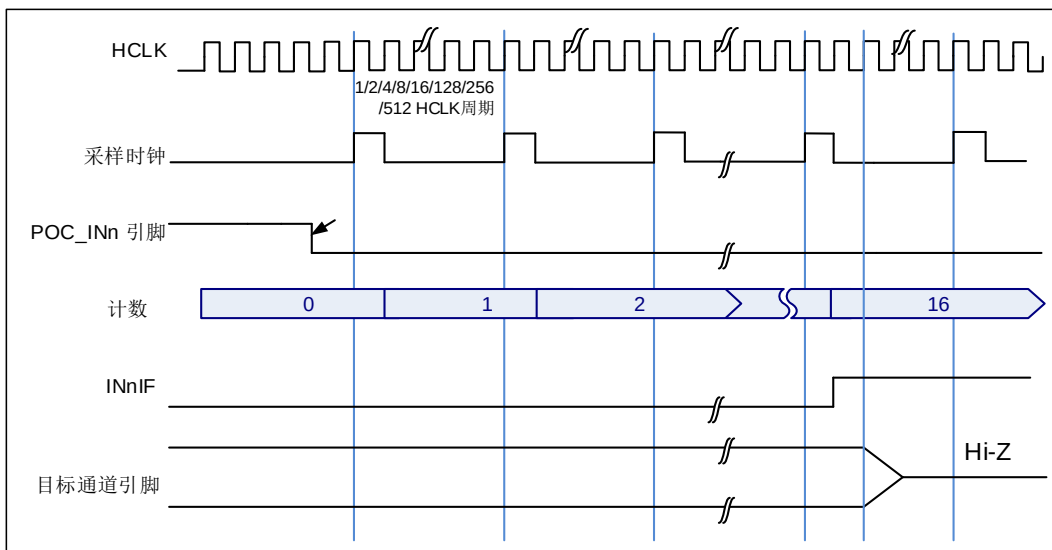


电平检测模式

可以通过设置 `POC_INnDCFG` ($n=0,1...5$) 寄存器的 `INDMSEL[3:0]` 位域选择一个合适的采样时钟频率来对 `POC_INn` ($n=0,1...5$) 引脚输入信号进行电平检测。如果在所选的时钟频率下，已经连续采样低电平($INPL=0$)或者高电平($INPL=1$)的次数达到 `POC_INnDCFG` ($n=0,1...5$) 寄存器的 `INDSNM[3:0]` 位域所设定的有效次数，那么，标志位 `INnIF` ($n=0,1...5$) 将会在计数到所设定的有效次数时被硬件置起。同时，一个禁能请求将产生。

相似地，如果该禁能请求已被使能，且目标定时器已被选择，在一个 `HCLK` 后，与目标定时器通道相连的输出引脚将被禁能且可以选择高阻输出。

只有当一个高电平信号 ($INPL=0$) 或者低电平信号 ($INPL=1$) 输入到 `POC_INn` ($n=0,1...5$) 引脚时，再向标志位 `POCnIF` ($n=0,1...5$) 写 0，才可以清除该标志位，否则，写 0 是无效的。当该标志位被清除后，被禁能的引脚可以从禁能输出中恢复。

图 8-4. POC 引脚低电平检测 (`INDSNM[3:0]=4'b0000`)

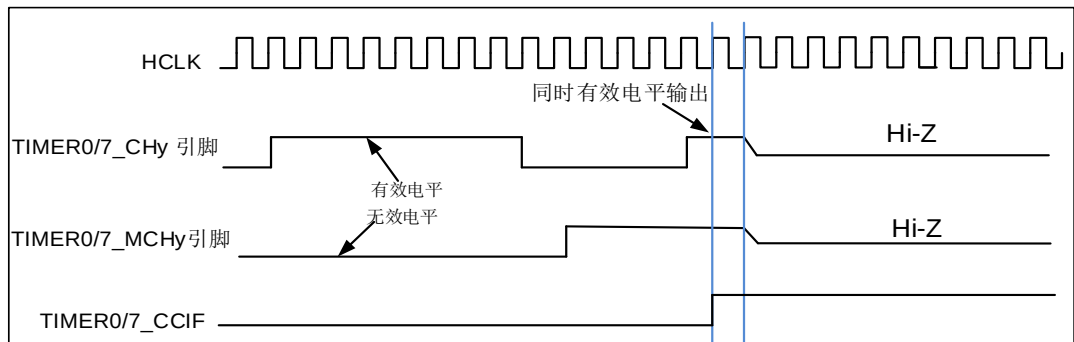
互补通道检测

当 `TIMER0/7` 互补 PWM 通道 (`CH0/1/2` 和 `MCH0/1/2`) 同时输出有效电平时 (至少有一对互

补通道同时输出有效电平), POC_STAT0 寄存器的标志位 TIMER0/7_CCIF 会被硬件置 1, 且一个禁能请求将会生成。如果该禁能请求已经通过设置 TIMER0/7_CCDREN 位使能, 那么, 与 TIMER0/7 所有互补 PWM 通道 (CH0/1/2/3 和 MCH0/1/2/3) 相连的输出引脚将会被禁能, 且被禁能的引脚可以通过设置 POC_ODMODE0 寄存器的 TIMER0/7_OSELy [1:0] (y=0,1,2,3) 位域选择高阻输出。

只有在一个无效电平从 TIMER0/7 互补通道输出后, 通过写 0 到 TIMER0/7_CCIF 标志位才能清除该标志位, 否则, 写 0 清除该标志位无效。在这个标志位被清除后, 被禁能的引脚将从失能输出中恢复。

图 8-5. 互补 PWM 通道输出检测



注意: TIMERx_CHy 和 TIMERx_MCHy 通道的有效电平为高。

此外, TIMER0/7_PSELEN 位可以决定 TIMER0/7 通道 (CH0/1/2 和 MCH0/1/2) 的有效极性配置是由 POC 相关寄存器来配置还是由 TIMER0/7 相关寄存器来配置。关于有效极性设置的细节请参考 TIMER0/7_PSELEN 位描述。

表 8-2. 输出检测的互补 PWM 通道

CH	MCH	中断标志位
TIMER0_CH0	TIMER0_MCH0	TIMER0_CCIF
TIMER0_CH1	TIMER0_MCH1	
TIMER0_CH2	TIMER0_MCH2	
TIMER7_CH0	TIMER7_MCH0	TIMER7_CCIF
TIMER7_CH1	TIMER7_MCH1	
TIMER7_CH2	TIMER7_MCH2	

软件禁能输出

通过设置 POC_SWDRG 寄存器的 GPTIMERx_SWDRG (x=0,1) 或者 TIMERx_SWDRG (x=0,1,2,7) 位为 1, 可以利用软件生成一个禁能请求。软件生成的禁能请求无需进行请求使能, 一旦禁能请求生成, 与目标定时器通道相连的引脚输出将被禁能。被禁能的引脚可以变成高阻态。

例如, 与 TIMER1_CH0/1/2/3 相连的输出引脚可以直接通过设置 TIMER1_SWDR 位为 1 来禁能。

系统故障检测

HXTAL 卡住事件

当 HXTAL 卡住事件发生时，POC_STAT0 寄存器的 HXTALSDF 标志位会被硬件置 1，并且，一个禁能请求将产生。如果已经设置 POC_CTL0 寄存器的 HXTALS_DREN 位使能了禁能请求，那么，[表 8-1. 输出禁能的目标定时器和通道](#)所示的所有通道引脚都将被禁能。

CPU 锁定事件

当 CPU 内核锁定事件发生时，POC_STAT0 寄存器的 LOCKUPDF 标志位会被硬件置 1，并且，一个禁能请求将产生。如果已经设置 POC_CTL0 寄存器的 LOCKUP_DREN 位使能了禁能请求，那么，[表 8-1. 输出禁能的目标定时器和通道](#)所示的所有通道引脚都将被禁能。

比较器输出检测

当 CMPn (n=0,1,...,3) 输出上升沿时，检测标志位 (n=0,1,...,3) 会被硬件置 1，且一个禁能请求将会生成。

有两种方式去使能由 CMPn (n=0,1,...,3) 输出检测生成的禁能请求。一种方式是设置 POC_CTL1 寄存器的 CMPnDREN 位为 1 来对禁能请求进行使能。通过这种方式可以给 TIMERx (x=0,1,2,7) 和 GPTIMERx (x=0,1) 所有通道引脚打开禁能请求开关。另一种方式是设置 POC_EXTCTL0/1 寄存器的 TIMERx_CMPnDREN (x=0,1,2,7 n=0,1,...,3) 位或者 GPTIMERx_CMPnDREN (x=0,1 n=0,1,...,3) 位为 1 来对禁能请求进行使能。这种方式可以给相应的定时器通道引脚打开禁能请求开关。

如果由 CMPn (n=0,1,...,3) 输出检测生成的禁能请求已经被使能，且已选择了目标定时器，那么，与目标定时器通道相连的引脚将会被禁能，且被禁能的引脚可以变成高阻态。

通过清除标志位 CMPnDF (n=0,1,...,3)，引脚可以从禁能输出中恢复。

8.3.3. 目标定时器选择

由 POC_INn (n=0,1,...,5) 引脚输入检测和 CMPn (n=0,1,...,3) 输出检测产生的禁能请求可以通过配置 POC_REQSEL0/1 寄存器选择合适的目标 TIMER。当禁能请求到达时，与目标定时器通道相连的输出引脚将会被禁能。

需要注意的是有一些禁能请求在复位后已经默认选择了目标定时器。

此外，一个禁能请求能够选择不同的定时器作为目标定时器，且不同的禁能请求也可以选择同一个定时器作为目标定时器。

表 8-3. 禁能请求的目标定时器选择

禁能请求源	默认选择	非默认选择
POC_IN0	TIMER0	TIMER1, TIMER2, GPTIMER0, GPTIMER1, TIMER7
POC_IN1	TIMER7	TIMER0, TIMER1, TIMER2, GPTIMER0,

禁能请求源	默认选择	非默认选择
		GPTIMER1
POC_IN2	TIMER1, TIMER2, GPTIMER0, GPTIMER1	TIMER0, TIMER7
POC_IN3	No timer	TIMER0, TIMER1, TIMER2, GPTIMER0, GPTIMER1, TIMER7
POC_IN4	TIMER1	TIMER0, TIMER2, GPTIMER0, GPTIMER1 TIMER7
POC_IN5	TIMER2	TIMER0, TIMER1, GPTIMER0, GPTIMER1, TIMER7
CMPn (n=0,1,...,3)	No timer	TIMER0, TIMER1, TIMER2, GPTIMER0, GPTIMER1, TIMER7

8.3.4. 输出切换

当一个输出引脚被禁能时,如果 POC_ODMODE0/1 寄存器的 $TIMERx\_OSELy[1:0]$ ($x=0,1,2,7$ $y=0,1,2,3$) 或者 $GPTIMERx\_OSELy[1:0]$ ($x=0,1$ $y=0,1,2,3$) 位域被设置成 2'b01, 则该引脚可以切换成高阻态输出。

如果 $TIMERx\_OSELy[1:0]$ ($x=0,1,2,7$ $y=0,1,2,3$) 或者 $GPTIMERx\_OSELy[1:0]$ ($x=0,1$ $y=0,1,2,3$) 位域被设置成 2'b00 时, 引脚将输出定时器通道的信号。

如果 $TIMERx\_OSELy[1:0]$ ($x=0,1,2,7$ $y=0,1,2,3$) 或者 $GPTIMERx\_OSELy[1:0]$ ($x=0,1$ $y=0,1,2,3$) 位域被设置成 2'b10 时, 引脚输出由 GPIO 控制, 且引脚的输出状态由 $GPIOx\_OCTL$ ($x=A,B,...,H$) 寄存器决定。

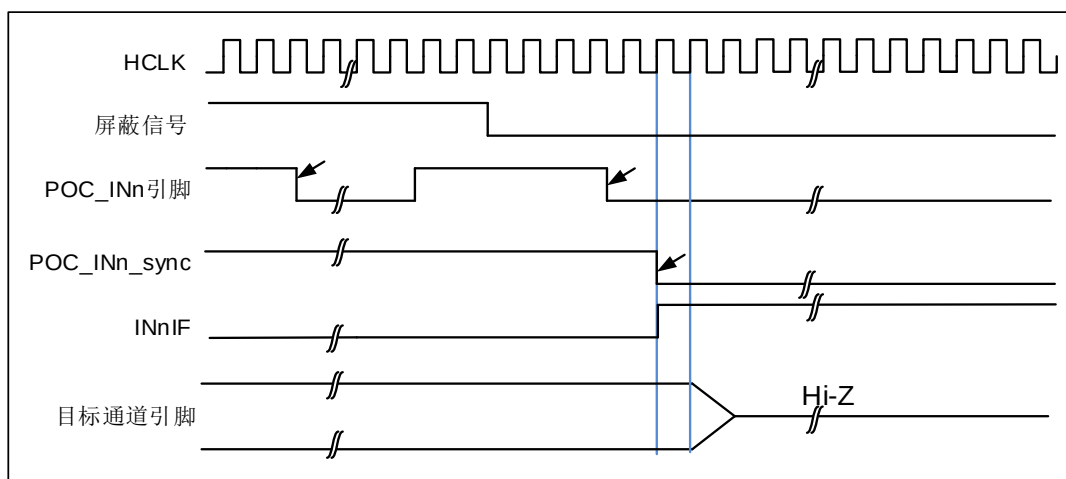
8.3.5. 检测屏蔽功能

检测屏蔽功能可以用来屏蔽 POC_INn ($n=0,1,...,5$) 引脚的输入信号和 CMPn ($n=0,1,...,3$) 输出检测信号。

如果 POC_INnDMK ($n=0,1,...,5$) 寄存器的 INMKSEL[4:0] 位域没有被设置成 4'b0000, 那么, 当被选择的定时器通道输出高电平时, POC_INn ($n=0,1,...,5$) 引脚的输入信号将被屏蔽。

相似地, 当由 POC_CMPnDMK ($n=0,1,...,3$) 寄存器的 CMPMKSEL[4:0] 位域确定的定时器通道输出高电平时, CMPn ($n=0,1,...,4$) 输出检测信号将会被屏蔽。

图 8-6. POC 引脚下降沿输入检测屏蔽



8.3.6. POC 中断

下面的检测可以产生中断：

- 与中断标志位INnIF (n=0,1...5)有关的POC_INn (n=0,1...5)引脚输入检测。
- 与中断标志位TIMER0/7_CCIF有关的TIMER0/7互补通道输出检测。

以上所有中断都映射的同一的中断向量。

8.4. POC 寄存器

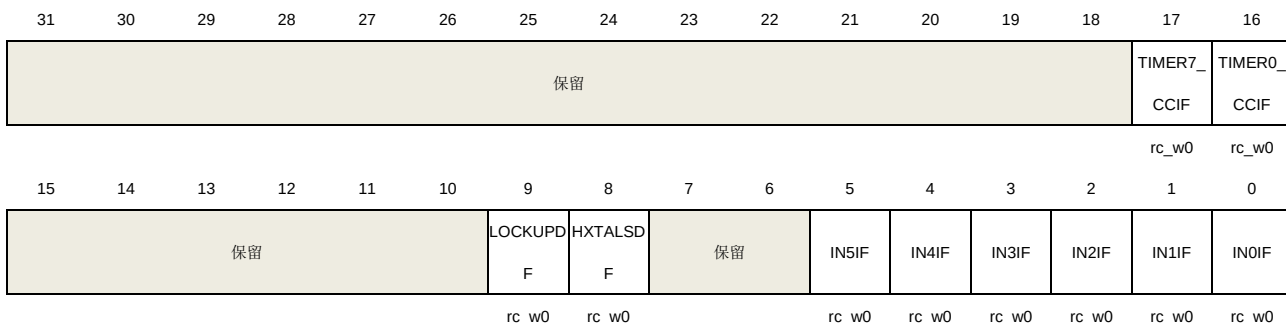
POC基地址：0x4800 4400

8.4.1. 状态寄存器 0（POC_STAT0）

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
31:18	保留	必须保持复位值。
17	TIMER7_CCIF	TIMER7同时导通中断标志 该位表明TIMER7互补通道的同时导通检测是否产生禁能请求。 0：未产生禁能请求 1：产生禁能请求 当TIMER7至少有一对互补通道（TIMER7_CH0/1/2和TIMER7_MCH0/1/2）同时输出有效电平时该位被硬件置1。 软件写0清除。
16	TIMER0_CCIF	TIMER0同时导通中断标志 该位表明TIMER0互补通道的同时导通检测是否产生禁能请求。 0：未产生禁能请求 1：产生禁能请求 当TIMER0至少有一对互补通道（TIMER0_CH0/1/2和TIMER0_MCH0/1/2）同时输出有效电平时该位被硬件置1。 软件写0清除。
15:10	保留	必须保持复位值。
9	LOCKUPDF	CPU锁定检测标志 该位表明CPU锁定事件检测是否产生禁能请求。 0：未产生禁能请求 1：产生禁能请求 当CPU锁定事件发生时该位被硬件置1。

		软件写0清除。
8	HXTALSDF	HXTAL卡住检测标志 该位表明HXTAL卡住检测是否产生禁能请求。 0: 未产生禁能请求 1: 产生禁能请求 当HXTAL卡住事件发生时该位被硬件置1。 软件写0清除。
7:6	保留	必须保持复位值。
5	IN5IF	POC_IN5输入中断标志 该位表明POC_IN5输入检测是否产生禁能请求。 0: 未产生禁能请求 1: 产生禁能请求 当POC_IN5引脚输入POC_IN5DCFG寄存器的INDMSEL[3:0]位域所选择的有效信号时该位被硬件置1。 软件写0清除。
4	IN4IF	POC_IN4输入中断标志 该位表明POC_IN4输入检测是否产生禁能请求。 0: 未产生禁能请求 1: 产生禁能请求 当POC_IN4引脚输入POC_IN4DCFG寄存器的INDMSEL[3:0]位域所选择的有效信号时该位被硬件置1。 软件写0清除。
3	IN3IF	POC_IN3输入中断标志 该位表明POC_IN3输入检测是否产生禁能请求。 0: 未产生禁能请求 1: 产生禁能请求 当POC_IN3引脚输入POC_IN3DCFG寄存器的INDMSEL[3:0]位域所选择的有效信号时该位被硬件置1。 软件写0清除。
2	IN2IF	POC_IN2输入中断标志 该位表明POC_IN2输入检测是否产生禁能请求。 0: 未产生禁能请求 1: 产生禁能请求 当POC_IN2引脚输入POC_IN2DCFG寄存器的INDMSEL[3:0]位域所选择的有效信号时该位被硬件置1。 软件写0清除。
1	IN1IF	POC_IN1输入中断标志 该位表明POC_IN1输入检测是否产生禁能请求。 0: 未产生禁能请求 1: 产生禁能请求

当POC_IN1引脚输入POC_IN1DCFG寄存器的INDMSEL[3:0]位域所选择的有效信号时该位被硬件置1。

软件写0清除。

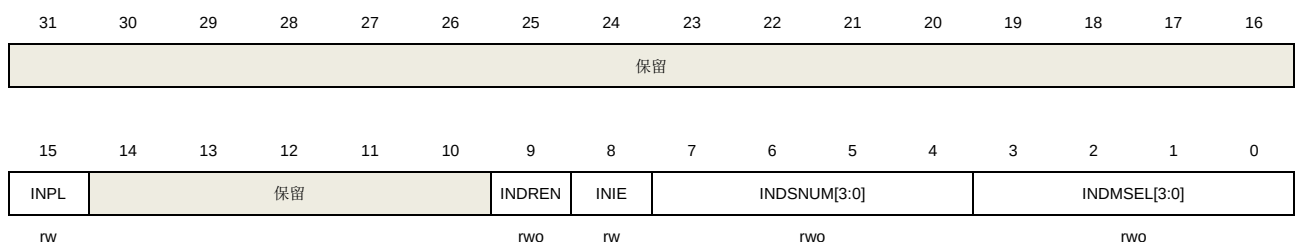
0	IN0IF	POC_IN0输入中断标志 该位表明POC_IN0输入检测是否产生禁能请求。 0: 未产生禁能请求 1: 产生禁能请求 当POC_IN0引脚输入POC_IN0DCFG寄存器的INDMSEL[3:0]位域所选择的有效信号时该位被硬件置1。 软件写0清除。

8.4.2. 输入 n 检测配置寄存器 (POC_INnDCFG) (n=0...5)

地址偏移: $0x04 + 0x04 * n$

复位值: 0x04/0x08: 0x0000 0200; 其他: 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
31:16	保留	必须保持复位值。
15	INPL	POC_INn引脚输入极性 该位用于选择POC_INn引脚的输入极性。 0: 输入不反相 1: 输入反相
14:10	保留	必须保持复位值。
9	INDREN	POC_INn禁能请求使能 该位用于确定POC_INn引脚输入检测生成的禁能请求是否对TIMERx (x=0,1,2,7) 和 GPTIMERx (x=0,1) 使能。 0: 禁能请求未使能 1: 禁能请求使能 如果该位使能, 当 INnIF = 1 和 TIMERx_INnDRSEL = 1 时, TIMERx (x=0,1,2,7) 通道引脚输出被禁能。 如果该位使能, 当 INnIF = 1 和 GPTIMERx_INnDRSEL = 1 时, GPTIMERx (x=0,1) 通道引脚输出被禁能。 注意: 该位在复位后只能写入一次。

8	INIE	POC_INn输入中断使能 该位用于当INnIF = 1时使能或者禁止中断。 0: 中断禁止 1: 中断使能
7:4	INDSNUM[3:0]	POC_INn输入检测采样数 该位域用于选择POC_INn输入检测的有效采样数。 0000: 16次 0001: 4次 0010: 8次 0011: 9次 0100: 10次 0101: 11次 0110: 12次 0111: 13次 1000: 14次 1001: 15次 其他: 保留。 注意: 该位域在复位后只能写入一次。
3:0	INDMSEL[3:0]	POC_INn输入检测模式选择 这些位用于选择POC_INn输入检测的有效信号和采样频率，以生成禁能请求。 0000: 采样频率为f_{HCLK}，在下降沿（INPL=0）或上升沿（INPL=1）生成请求。 0001: 采样频率为$f_{HCLK}/8$，当低电平（INPL=0）或高电平（INPL=1）被连续采样一定次数后产生一个请求。 0010: 采样频率为$f_{HCLK}/16$，当低电平（INPL=0）或高电平（INPL=1）被连续采样一定次数后产生一个请求。 0011: 采样频率为$f_{HCLK}/128$，当低电平（INPL=0）或高电平（INPL=1）被连续采样一定次数后产生一个请求。 0100: 采样频率为f_{HCLK}，当低电平（INPL=0）或高电平（INPL=1）被连续采样一定次数后产生一个请求。 0101: 采样频率为$f_{HCLK}/2$，当低电平（INPL=0）或高电平（INPL=1）被连续采样一定次数后产生一个请求。 0110: 采样频率为$f_{HCLK}/4$，当低电平（INPL=0）或高电平（INPL=1）被连续采样一定次数后产生一个请求。 0111: 采样频率为$f_{HCLK}/256$，当低电平（INPL=0）或高电平（INPL=1）被连续采样一定次数后产生一个请求。 1000: 采样频率为$f_{HCLK}/512$，当低电平（INPL=0）或高电平（INPL=1）被连续采样一定次数后产生一个请求。 1001: 采样频率为$f_{HCLK}/32$，当低电平（INPL=0）或高电平（INPL=1）被连续采样一定次数后产生一个请求。 1010: 采样频率为$f_{HCLK}/64$，当低电平（INPL=0）或高电平（INPL=1）被连续采样一定次数后产生一个请求。 1011: 采样频率为$f_{HCLK}/2$，在下降沿（INPL=0）或上升沿（INPL=1）生成请求。

1100: 采样频率为 $f_{HCLK}/4$ ，在下降沿（INPL=0）或上升沿（INPL=1）生成请求。

其他：保留。

注意：该位域在复位后只能写入一次。

8.4.3. 控制寄存器 0（POC_CTL0）

地址偏移：0x20

复位值：0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

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

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	LOCKUP_DREN	CPU锁定禁能请求使能 该位用于确定CPU锁定事件产生的禁能请求是否使能。 0：禁能请求不使能 1：禁能请求使能 如果该位使能，当 LOCKUPDF = 1 时，所有与 POC 相关的定时器通道引脚输出都被禁能。 注意：该位在复位后只能写入一次。
0	HXTALS_DREN	HXTAL卡住禁能请求使能 该位用于确定HXTAL卡住事件产生的禁能请求是否使能。 0：禁能请求不使能 1：禁能请求使能 如果该位使能，当 HXTALSDF = 1 时，所有与 POC 相关的定时器通道引脚输出都被禁能。 注意：该位在复位后只能写入一次。

8.4.4. 软件禁能请求生成寄存器（POC_SWDRG）

地址偏移：0x24

复位值：0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留										GPTIMER1_SWDRG	GPTIMER0_SWDRG	TIMER2_SWDRG	TIMER1_SWDRG	TIMER7_SWDRG	TIMER0_SWDRG
										rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:6	保留	必须保持复位值。
5	GPTIMER1_SWDRG	GPTIMER1的软件禁能请求生成 该位用于软件生成禁能请求来禁能GPTIMER1通道引脚输出。 0: 软件不生成禁能请求 1: 软件生成禁能请求
4	GPTIMER0_SWDRG	GPTIMER0的软件禁能请求生成 该位用于软件生成禁能请求来禁能GPTIMER0通道引脚输出。 0: 软件不生成禁能请求 1: 软件生成禁能请求
3	TIMER2_SWDRG	TIMER2的软件禁能请求生成 该位用于软件生成禁能请求来禁能TIMER2通道引脚输出。 0: 软件不生成禁能请求 1: 软件生成禁能请求
2	TIMER1_SWDRG	TIMER1的软件禁能请求生成 该位用于软件生成禁能请求来禁能TIMER1通道引脚输出。 0: 软件不生成禁能请求 1: 软件生成禁能请求
1	TIMER7_SWDRG	TIMER7的软件禁能请求生成 该位用于软件生成禁能请求来禁能TIMER7通道引脚输出。 0: 软件不生成禁能请求 1: 软件生成禁能请求
0	TIMER0_SWDRG	TIMER0的软件禁能请求生成 该位用于软件生成禁能请求来禁能TIMER0通道引脚输出。 0: 软件不生成禁能请求 1: 软件生成禁能请求

8.4.5. 互补检测配置寄存器 0 (POC_CDCFG0)

地址偏移: 0x28

复位值: 0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						TIMER0_ CCDREN	TIMER0_ CCIE	TIMER0_P SELEN	保留	TIMER0_ MCH2_APS SEL	TIMER0_ CH2_APS EL	TIMER0_ MCH1_APS SEL	TIMER0_ CH1_APS EL	TIMER0_ MCH0_APS SEL	TIMER0_ CH0_APS EL
						rwo	rw	rwo		rwo	rwo	rwo	rwo	rwo	rwo

位/位域	名称	描述
31:10	保留	必须保持复位值。
9	TIMER0_CCDREN	TIMER0同时导通禁能请求使能 该位用于确定TIMER0同时导通检测产生的禁能请求是否使能。 0: 禁能请求不使能 1: 禁能请求使能 如果该位使能，当TIMER0_CCIF = 1时，TIMER0互补PWM通道引脚输出被禁能。 注意：该位在复位后只能写入一次。
8	TIMER0_CCIE	TIMER0同时导通中断使能 该位用于当TIMER0_CCIF = 1时使能或者禁止中断。 0: 中断禁止 1: 中断使能
7	TIMER0_PSELEN	TIMER0极性选择使能 该位用于使能POC_CDCFG0寄存器来控制TIMER0通道的有效极性选择。 0: 禁止 1: 使能 如果该位禁止，TIMER0_CHCTL2寄存器的CHxP (x=0,1,2) 和MCHxP (x=0,1,2) 位决定TIMER0_CH0/1/2和TIMER0_MCH0/1/2的有效极性。 如果该位使能，TIMER0_CHx_APSEL (x=0,1,2) 和TIMER0_MCHx_APSEL (x=0,1,2) 位决定TIMER0_CH0/1/2和TIMER0_MCH0/1/2的有效极性。 注意：该位在复位后只能写入一次。
6	保留	必须保持复位值。
5	TIMER0_MCH2_APS EL	TIMER0_MCH2有效极性选择 该位用于选择TIMER0_MCH2的有效极性。 0: 高电平有效 1: 低电平有效 注意：该位在复位后只能写入一次。
4	TIMER0_CH2_APSE L	TIMER0_CH2有效极性选择 该位用于选择TIMER0_CH2的有效极性。 0: 高电平有效 1: 低电平有效 注意：该位在复位后只能写入一次。
3	TIMER0_MCH1_APS	TIMER0_MCH1有效极性选择

EL	该位用于选择TIMER0_MCH1的有效极性。 0: 高电平有效 1: 低电平有效 注意: 该位在复位后只能写入一次。
2	TIMER0_CH1_APSE L TIMER0_CH1有效极性选择 该位用于选择TIMER0_CH1的有效极性。 0: 高电平有效 1: 低电平有效 注意: 该位在复位后只能写入一次。
1	TIMER0_MCH0_APS EL TIMER0_MCH0有效极性选择 该位用于选择TIMER0_MCH0的有效极性。 0: 高电平有效 1: 低电平有效 注意: 该位域在复位后只能写入一次。
0	TIMER0_CH0_APSE L TIMER0_CH0有效极性选择 该位用于选择TIMER0_CH0的有效极性。 0: 高电平有效 1: 低电平有效 注意: 该位域在复位后只能写入一次。

8.4.6. 互补检测配置寄存器 1 (POC_CDCFG1)

地址偏移: 0x2C

复位值: 0x0000 0000

该寄存器可以按字节（8 位）、半字（16 位）或字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						TIMER7_ CCDREN	TIMER7_ CCIE	TIMER7_ SELEN	保留	TIMER7_ MCH2_AP SEL	TIMER7_ CH2_APS EL	TIMER7_ MCH1_AP SEL	TIMER7_ CH1_APS EL	TIMER7_ MCH0_AP SEL	TIMER7_ CH0_APS EL
						rwo	rw	rwo		rwo	rwo	rwo	rwo	rwo	rwo

Bits	Fields	Descriptions
31:10	保留	必须保持复位值。
9	TIMER7_CCDREN	TIMER7同时导通禁能请求使能 该位用于确定TIMER7同时导通检测产生的禁能请求是否使能。 0: 禁能请求不使能 1: 禁能请求使能

		如果该位使能，当TIMER7_CCIF = 1时，TIMER7互补PWM通道引脚输出被禁能。 注意： 该位在复位后只能写入一次。
8	TIMER7_CCIE	TIMER7同时导通中断使能 该位用于当TIMER7_CCIF = 1时使能或者禁止中断。 0：中断禁止 1：中断使能
7	TIMER7_PSELEN	TIMER7极性选择使能 该位用于使能POC_CDCFG1寄存器来控制TIMER7通道的有效极性选择。 0：禁止 1：使能 如果该位禁止，TIMER7_CHCTL2寄存器的CHxP（x=0,1,2）和MCHxP（x=0,1,2）位决定TIMER7_CH0/1/2和TIMER7_MCH0/1/2的有效极性。 如果该位使能，TIMER7_CHx_APSEL（x=0,1,2）和TIMER7_MCHx_APSEL（x=0,1,2）位决定TIMER7_CH0/1/2和TIMER7_MCH0/1/2的有效极性。 注意： 该位在复位后只能写入一次。
6	保留	必须保持复位值。
5	TIMER7_MCH2_APS EL	TIMER7_MCH2有效极性选择 该位用于选择TIMER7_MCH2的有效极性。 0：高电平有效 1：低电平有效 注意： 该位在复位后只能写入一次。
4	TIMER7_CH2_APSE L	TIMER7_CH2有效极性选择 该位用于选择TIMER7_CH2的有效极性。 0：高电平有效 1：低电平有效 注意： 该位在复位后只能写入一次。
3	TIMER7_MCH1_APS EL	TIMER7_MCH1有效极性选择 该位用于选择TIMER7_MCH1的有效极性。 0：高电平有效 1：低电平有效 注意： 该位在复位后只能写入一次。
2	TIMER7_CH1_APSE L	TIMER7_CH1有效极性选择 该位用于选择TIMER7_CH1的有效极性。 0：高电平有效 1：低电平有效 注意： 该位在复位后只能写入一次。
1	TIMER7_MCH0_APS EL	TIMER7_MCH0有效极性选择 该位用于选择TIMER7_MCH0的有效极性。 0：高电平有效

1: 低电平有效

注意: 该位在复位后只能写入一次。

- 0 TIMER7_CH0_APSE TIMER7_CH0有效极性选择
- L 该位用于选择TIMER7_CH0的有效极性。
- 0: 高电平有效
- 1: 低电平有效
- 注意:** 该位在复位后只能写入一次。

8.4.7. 输出禁能模式寄存器 0 (POC_ODMODE0)

地址偏移: 0x40

复位值: 0x0000 1515

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
TIMER7_OSEL3[1:0]		TIMER7_OSEL2[1:0]		TIMER7_OSEL1[1:0]		TIMER7_OSEL0[1:0]		TIMER0_OSEL3[1:0]		TIMER0_OSEL2[1:0]		TIMER0_OSEL1[1:0]		TIMER0_OSEL0[1:0]	
rwo		rwo		rwo		rwo		rwo		rwo		rwo		rwo	

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:14	TIMER7_OSEL3[1:0]	TIMER7_CH3/ TIMER7_MCH3通道引脚输出选择 该位域确定当TIMER7_CH3/ TIMER7_MCH3引脚被禁能时，是选择Hi-Z输出还是GPIO输出。 00: TIMER7_CH3/TIMER7_MCH3输出 01: Hi-Z输出 10: GPIO输出 11: 保留 注意: 该位在复位后只能写入一次。
13:12	TIMER7_OSEL2[1:0]	TIMER7_CH2/ TIMER7_MCH2通道引脚输出选择 该位域确定当TIMER7_CH2/ TIMER7_MCH2引脚被禁能时，是选择Hi-Z输出还是GPIO输出。 00: TIMER7_CH2/TIMER7_MCH2输出 01: Hi-Z输出 10: GPIO输出 11: 保留 注意: 该位在复位后只能写入一次。复位后的默认值是 2'b01。
11:10	TIMER7_OSEL1[1:0]	TIMER7_CH1/ TIMER7_MCH1通道引脚输出选择 该位域确定当TIMER7_CH1/ TIMER7_MCH1引脚被禁能时，是选择Hi-Z输出还是

		GPIO输出。 00: TIMER7_CH1/TIMER7_MCH1输出 01: Hi-Z输出 10: GPIO输出 11: 保留 注意: 该位在复位后只能写入一次。复位后的默认值是 2'b01。
9:8	TIMER7_OSEL0[1:0]	TIMER7_CH0/ TIMER7_MCH0通道引脚输出选择 该位域确定当TIMER7_CH0/ TIMER7_MCH0引脚被禁能时，是选择Hi-Z输出还是GPIO输出。 00: TIMER7_CH1/TIMER7_MCH1输出 01: Hi-Z输出 10: GPIO输出 11: 保留 注意: 该位在复位后只能写入一次。复位后的默认值是 2'b01。
7:6	TIMER0_OSEL3[1:0]	TIMER0_CH3/ TIMER0_MCH3通道引脚输出选择 该位域确定当TIMER0_CH3/ TIMER0_MCH3引脚被禁能时，是选择Hi-Z输出还是GPIO输出。 00: TIMER0_CH3/TIMER0_MCH3输出 01: Hi-Z输出 10: GPIO输出 11: 保留 注意: 该位在复位后只能写入一次。
5:4	TIMER0_OSEL2[1:0]	TIMER0_CH2/ TIMER0_MCH2通道引脚输出选择 该位域确定当TIMER0_CH2/ TIMER0_MCH2引脚被禁能时，是选择Hi-Z输出还是GPIO输出。 00: TIMER0_CH2/TIMER0_MCH2输出 01: Hi-Z输出 10: GPIO输出 11: 保留 注意: 该位在复位后只能写入一次。复位后的默认值是 2'b01。
3:2	TIMER0_OSEL1[1:0]	TIMER0_CH1/ TIMER0_MCH1通道引脚输出选择 该位域确定当TIMER0_CH1/ TIMER0_MCH1引脚被禁能时，是选择Hi-Z输出还是GPIO输出。 00: TIMER0_CH1/TIMER0_MCH1输出 01: Hi-Z输出 10: GPIO输出 11: 保留 注意: 该位在复位后只能写入一次。复位后的默认值是 2'b01。
1:0	TIMER0_OSEL0[1:0]	TIMER0_CH0/ TIMER0_MCH0通道引脚输出选择 该位域确定当TIMER0_CH0/ TIMER0_MCH0引脚被禁能时，是选择Hi-Z输出还是GPIO输出。

00: TIMER0_CH0/TIMER0_MCH0输出

01: Hi-Z输出

10: GPIO输出

11: 保留

注意：该位在复位后只能写入一次。复位后的默认值是 2'b01。

8.4.8. 输出禁能模式寄存器 1 (POC_ODMODE1)

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
保留				GPTIMER1_OSEL1[1:0]		GPTIMER1_OSEL0[1:0]		保留				GPTIMER0_OSEL1[1:0]		GPTIMER0_OSEL0[1:0]	
rwo				rwo		rwo				rwo		rwo		rwo	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER2_OSEL3[1:0]		TIMER2_OSEL2[1:0]		TIMER2_OSEL1[1:0]		TIMER2_OSEL0[1:0]		TIMER1_OSEL3[1:0]		TIMER1_OSEL2[1:0]		TIMER1_OSEL1[1:0]		TIMER1_OSEL0[1:0]	
rwo		rwo		rwo		rwo		rwo		rwo		rwo		rwo	

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:26	GPTIMER1_OSEL1[1:0]	GPTIMER1_CH1通道引脚输出选择 该位域确定当GPTIMER1_CH1引脚被禁能时，是选择Hi-Z输出还是GPIO输出。 00: GPTIMER1_CH1输出 01: Hi-Z输出 10: GPIO输出 11: 保留 注意：该位在复位后只能写入一次。
25:24	GPTIMER1_OSEL0[1:0]	GPTIMER1_CH0通道引脚输出选择 该位域确定当GPTIMER1_CH0引脚被禁能时，是选择Hi-Z输出还是GPIO输出。 00: GPTIMER1_CH0输出 01: Hi-Z输出 10: GPIO输出 11: 保留 注意：该位在复位后只能写入一次。
23:20	保留	必须保持复位值。
19:18	GPTIMER0_OSEL1[1:0]	GPTIMER0_CH1通道引脚输出选择 该位域确定当GPTIMER0_CH1引脚被禁能时，是选择Hi-Z输出还是GPIO输出。 00: GPTIMER0_CH1输出 01: Hi-Z输出

		10: GPIO输出
		11: 保留
		注意: 该位在复位后只能写入一次。
17:16	GPTIMER0_OSEL0[1:0]	GPTIMER0_CH0通道引脚输出选择 该位域确定当GPTIMER0_CH0引脚被禁能时, 是选择Hi-Z输出还是GPIO输出。 00: GPTIMER0_CH0输出 01: Hi-Z输出 10: GPIO输出 11: 保留 注意: 该位在复位后只能写入一次。
15:14	TIMER2_OSEL3[1:0]	TIMER2_CH3通道引脚输出选择 该位域确定当TIMER2_CH3引脚被禁能时, 是选择Hi-Z输出还是GPIO输出。 00: TIMER2_CH3输出 01: Hi-Z输出 10: GPIO输出 11: 保留 注意: 该位在复位后只能写入一次。
13:12	TIMER2_OSEL2[1:0]	TIMER2_CH2通道引脚输出选择 该位域确定当TIMER2_CH2引脚被禁能时, 是选择Hi-Z输出还是GPIO输出。 00: TIMER2_CH2输出 01: Hi-Z输出 10: GPIO输出 11: 保留 注意: 该位在复位后只能写入一次。
11:10	TIMER2_OSEL1[1:0]	TIMER2_CH1通道引脚输出选择 该位域确定当TIMER2_CH1引脚被禁能时, 是选择Hi-Z输出还是GPIO输出。 00: TIMER2_CH1输出 01: Hi-Z输出 10: GPIO输出 11: 保留 注意: 该位在复位后只能写入一次。
9:8	TIMER2_OSEL0[1:0]	TIMER2_CH0通道引脚输出选择 该位域确定当TIMER2_CH0引脚被禁能时, 是选择Hi-Z输出还是GPIO输出。 00: TIMER2_CH0输出 01: Hi-Z输出 10: GPIO输出 11: 保留 注意: 该位在复位后只能写入一次。
7:6	TIMER1_OSEL3[1:0]	TIMER1_CH3通道引脚输出选择 该位域确定当TIMER1_CH3引脚被禁能时, 是选择Hi-Z输出还是GPIO输出。

00: TIMER1_CH3输出

01: Hi-Z输出

10: GPIO输出

11: 保留

注意：该位在复位后只能写入一次。

5:4 TIMER1_OSEL2[1:0] TIMER1_CH2通道引脚输出选择

该位域确定当TIMER1_CH2引脚被禁能时，是选择Hi-Z输出还是GPIO输出。

00: TIMER1_CH2输出

01: Hi-Z输出

10: GPIO输出

11: 保留

注意：该位在复位后只能写入一次。

3:2 TIMER1_OSEL1[1:0] TIMER1_CH1通道引脚输出选择

该位域确定当TIMER1_CH1引脚被禁能时，是选择Hi-Z输出还是GPIO输出。

00: TIMER1_CH1输出

01: Hi-Z输出

10: GPIO输出

11: 保留

注意：该位在复位后只能写入一次。

1:0 TIMER1_OSEL0[1:0] TIMER1_CH0通道引脚输出选择

该位域确定当TIMER1_CH0引脚被禁能时，是选择Hi-Z输出还是GPIO输出。

00: TIMER1_CH0输出

01: Hi-Z输出

10: GPIO输出

11: 保留

注意：该位在复位后只能写入一次。

8.4.9. 请求选择寄存器 0 (POC_REQSEL0)

地址偏移: 0x48

复位值: 0x0000 0201

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	TIMER7_	TIMER7_I	TIMER7_I	TIMER7_I	TIMER7_I	TIMER7_I	TIMER7_I	保留	TIMER0_	TIMER0_I	TIMER0_I	TIMER0_I	TIMER0_I	TIMER0_I	TIMER0_I
	CMPDRS	N5DRSEL	N4DRSEL	N3DRSEL	N2DRSEL	N1DRSEL	N0DRSEL		CMPDRS	N5DRSEL	N4DRSEL	N3DRSEL	N2DRSEL	N1DRSEL	N0DRSEL
	EL								EL						
	rwo	rwo	rwo	rwo	rwo	rwo	rwo		rwo	rwo	rwo	rwo	rwo	rwo	rwo

位/位域	名称	描述
31:15	保留	必须保持复位值。
14	TIMER7_CMPDRSE L	CMPn (n=0,1,2,3) 禁能请求选择TIMER7 该位用于确定CMPn (n=0,1,2,3) 输出检测生成的禁能请求是否选择TIMER7作为目标定时器。 0: TIMER7未被选择 1: TIMER7被选择 当CMPn生成禁能请求时 (CMPnDF被硬件设置为1), 如果该位被设置为1, 并且CMPnDREN或TIMER7_CMPnDREN也被设置为1, 则TIMER7通道引脚输出被禁能。 注意: 该位在复位后只能写入一次。
13	TIMER7_IN5DRSEL	POC_IN5禁能请求选择TIMER7 该位用于确定POC_IN5输入检测生成的禁能请求是否选择TIMER7作为目标定时器。 0: TIMER7未被选择 1: TIMER7被选择 当POC_IN5生成禁能请求时 (IN5IF被硬件设置为1), 如果该位被设置为1, 并且POC_IN5DCFG寄存器的INDREN位也被设置为1, 则TIMER7通道引脚输出被禁能。 注意: 该位在复位后只能写入一次。
12	TIMER7_IN4DRSEL	POC_IN4禁能请求选择TIMER7 该位用于确定POC_IN4输入检测生成的禁能请求是否选择TIMER7作为目标定时器。 0: TIMER7未被选择 1: TIMER7被选择 当POC_IN4生成禁能请求时 (IN4IF被硬件设置为1), 如果该位被设置为1, 并且POC_IN4DCFG寄存器的INDREN位也被设置为1, 则TIMER7通道引脚输出被禁能。 注意: 该位在复位后只能写入一次。
11	TIMER7_IN3DRSEL	POC_IN3禁能请求选择TIMER7 该位用于确定POC_IN3输入检测生成的禁能请求是否选择TIMER7作为目标定时器。 0: TIMER7未被选择 1: TIMER7被选择 当POC_IN3生成禁能请求时 (IN3IF被硬件设置为1), 如果该位被设置为1, 并且POC_IN3DCFG寄存器的INDREN位也被设置为1, 则TIMER7通道引脚输出被禁能。 注意: 该位在复位后只能写入一次。
10	TIMER7_IN2DRSEL	POC_IN2禁能请求选择TIMER7 该位用于确定POC_IN2输入检测生成的禁能请求是否选择TIMER7作为目标定时器。 0: TIMER7未被选择 1: TIMER7被选择 当POC_IN2生成禁能请求时 (IN2IF被硬件设置为1), 如果该位被设置为1, 并且POC_IN2DCFG寄存器的INDREN位也被设置为1, 则TIMER7通道引脚输出被禁能。 注意: 该位在复位后只能写入一次。
9	TIMER7_IN1DRSEL	POC_IN1禁能请求选择TIMER7

		该位用于确定POC_IN1输入检测生成的禁能请求是否选择TIMER7作为目标定时器。 0: TIMER7未被选择 1: TIMER7被选择 当POC_IN1生成禁能请求时（IN1IF被硬件设置为1），如果该位被设置为1，并且POC_IN1DCFG寄存器的INDREN位也被设置为1，则TIMER7通道引脚输出被禁能。 注意： 该位在复位后只能写入一次。复位后默认值为1。
8	TIMER7_IN0DRSEL	POC_IN0禁能请求选择TIMER7 该位用于确定POC_IN0输入检测生成的禁能请求是否选择TIMER7作为目标定时器。 0: TIMER7未被选择 1: TIMER7被选择 当POC_IN0生成禁能请求时（IN0IF被硬件设置为1），如果该位被设置为1，并且POC_IN0DCFG寄存器的INDREN位也被设置为1，则TIMER7通道引脚输出被禁能。 注意： 该位在复位后只能写入一次。
7	保留	必须保持复位值。
6	TIMER0_CMPDRSEL	CMPn（n=0,1,2,3）禁能请求选择TIMER0 该位用于确定CMPn（n=0,1,2,3）输出检测生成的禁能请求是否选择TIMER0作为目标定时器。 0: TIMER0未被选择 1: TIMER0被选择 当CMPn生成禁能请求时（CMPnDF被硬件设置为1），如果该位被设置为1，并且CMPnDREN或TIMER0_CMPnDREN也被设置为1，则TIMER0通道引脚输出被禁能。 注意： 该位在复位后只能写入一次。
5	TIMER0_IN5DRSEL	POC_IN5禁能请求选择TIMER0 该位用于确定POC_IN5输入检测生成的禁能请求是否选择TIMER0作为目标定时器。 0: TIMER0未被选择 1: TIMER0被选择 当POC_IN5生成禁能请求时（IN5IF被硬件设置为1），如果该位被设置为1，并且POC_IN5DCFG寄存器的INDREN位也被设置为1，则TIMER0通道引脚输出被禁能。 注意： 该位在复位后只能写入一次。
4	TIMER0_IN4DRSEL	POC_IN4禁能请求选择TIMER0 该位用于确定POC_IN4输入检测生成的禁能请求是否选择TIMER0作为目标定时器。 0: TIMER0未被选择 1: TIMER0被选择 当POC_IN4生成禁能请求时（IN4IF被硬件设置为1），如果该位被设置为1，并且POC_IN4DCFG寄存器的INDREN位也被设置为1，则TIMER0通道引脚输出被禁能。 注意： 该位在复位后只能写入一次。
3	TIMER0_IN3DRSEL	POC_IN3禁能请求选择TIMER0 该位用于确定POC_IN3输入检测生成的禁能请求是否选择TIMER0作为目标定时器。 0: TIMER0未被选择

1: TIMER0被选择

当POC_IN3生成禁能请求时（IN3IF被硬件设置为1），如果该位被设置为1，并且POC_IN3DCFG寄存器的INDREN位也被设置为1，则TIMER0通道引脚输出被禁能。

注意：该位在复位后只能写入一次。

2 TIMER0_IN2DRSEL POC_IN2禁能请求选择TIMER0

该位用于确定POC_IN2输入检测生成的禁能请求是否选择TIMER0作为目标定时器。

0: TIMER0未被选择

1: TIMER0被选择

当POC_IN2生成禁能请求时（IN2IF被硬件设置为1），如果该位被设置为1，并且POC_IN2DCFG寄存器的INDREN位也被设置为1，则TIMER0通道引脚输出被禁能。

注意：该位在复位后只能写入一次。

1 TIMER0_IN1DRSEL POC_IN1禁能请求选择TIMER0

该位用于确定POC_IN1输入检测生成的禁能请求是否选择TIMER0作为目标定时器。

0: TIMER0未被选择

1: TIMER0被选择

当POC_IN1生成禁能请求时（IN1IF被硬件设置为1），如果该位被设置为1，并且POC_IN1DCFG寄存器的INDREN位也被设置为1，则TIMER0通道引脚输出被禁能。

注意：该位在复位后只能写入一次。

0 TIMER0_IN0DRSEL POC_IN0禁能请求选择TIMER0

该位用于确定POC_IN0输入检测生成的禁能请求是否选择TIMER0作为目标定时器。

0: TIMER0未被选择

1: TIMER0被选择

当POC_IN0生成禁能请求时（IN0IF被硬件设置为1），如果该位被设置为1，并且POC_IN0DCFG寄存器的INDREN位也被设置为1，则TIMER0通道引脚输出被禁能。

注意：该位在复位后只能写入一次。复位后默认值为1。

8.4.10. 请求选择寄存器 1（POC_REQSEL1）

地址偏移：0x4C

复位值：0x0404 2414

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	GPTIMER1_CMPDRSEL	GPTIMER1_IN5DRSEL	GPTIMER1_IN4DRSEL	GPTIMER1_IN3DRSEL	GPTIMER1_IN2DRSEL	GPTIMER1_IN1DRSEL	GPTIMER1_IN0DRSEL	保留	GPTIMER0_CMPDRSEL	GPTIMER0_IN5DRSEL	GPTIMER0_IN4DRSEL	GPTIMER0_IN3DRSEL	GPTIMER0_IN2DRSEL	GPTIMER0_IN1DRSEL	GPTIMER0_IN0DRSEL
	SEL	EL	EL	EL	EL	EL	EL		SEL	EL	EL	EL	EL	EL	EL
	rwo	rwo	rwo	rwo	rwo	rwo	rwo		rwo	rwo	rwo	rwo	rwo	rwo	rwo
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	TIMER2_CMPDRSEL	TIMER2_IN5DRSEL	TIMER2_IN4DRSEL	TIMER2_IN3DRSEL	TIMER2_IN2DRSEL	TIMER2_IN1DRSEL	TIMER2_IN0DRSEL	保留	TIMER1_CMPDRSEL	TIMER1_IN5DRSEL	TIMER1_IN4DRSEL	TIMER1_IN3DRSEL	TIMER1_IN2DRSEL	TIMER1_IN1DRSEL	TIMER1_IN0DRSEL
	EL								EL						
	rwo	rwo	rwo	rwo	rwo	rwo	rwo		rwo	rwo	rwo	rwo	rwo	rwo	rwo

位/位域	名称	描述
31	保留	必须保持复位值。
30	GPTIMER1_CMPDR SEL	CMPn (n=0,1,2,3) 禁能请求选择GPTIMER1 该位用于确定CMPn (n=0,1,2,3) 输出检测生成的禁能请求是否选择GPTIMER1作为目标定时器。 0: GPTIMER1未被选择 1: GPTIMER1被选择 当CMPn生成禁能请求时 (CMPnDF被硬件设置为1), 如果该位被设置为1, 并且CMPnDREN或GPTIMER1_CMPnDREN也被设置为1, 则GPTIMER1通道引脚输出被禁能。 注意: 该位在复位后只能写入一次。
29	GPTIMER1_IN5DRS EL	POC_IN5禁能请求选择GPTIMER1 该位用于确定POC_IN5输入检测生成的禁能请求是否选择GPTIMER1作为目标定时器。 0: GPTIMER1未被选择 1: GPTIMER1被选择 当POC_IN5生成禁能请求时 (IN5IF被硬件设置为1), 如果该位被设置为1, 并且POC_IN5DCFG寄存器的INDREN位也被设置为1, 则GPTIMER1通道引脚输出被禁能。 注意: 该位在复位后只能写入一次。
28	GPTIMER1_IN4DRS EL	POC_IN4禁能请求选择GPTIMER1 该位用于确定POC_IN4输入检测生成的禁能请求是否选择GPTIMER1作为目标定时器。 0: GPTIMER1未被选择 1: GPTIMER1被选择 当POC_IN4生成禁能请求时 (IN4IF被硬件设置为1), 如果该位被设置为1, 并且POC_IN4DCFG寄存器的INDREN位也被设置为1, 则GPTIMER1通道引脚输出被禁能。 注意: 该位在复位后只能写入一次。
27	GPTIMER1_IN3DRS EL	POC_IN3禁能请求选择GPTIMER1 该位用于确定POC_IN3输入检测生成的禁能请求是否选择GPTIMER1作为目标定时器。 0: GPTIMER1未被选择 1: GPTIMER1被选择 当POC_IN3生成禁能请求时 (IN3IF被硬件设置为1), 如果该位被设置为1, 并且POC_IN3DCFG寄存器的INDREN位也被设置为1, 则GPTIMER1通道引脚输出被禁能。 注意: 该位在复位后只能写入一次。
26	GPTIMER1_IN2DRS EL	POC_IN2禁能请求选择GPTIMER1 该位用于确定POC_IN2输入检测生成的禁能请求是否选择GPTIMER1作为目标定时器。

		器。
		0: GPTIMER1未被选择
		1: GPTIMER1被选择
		当POC_IN2生成禁能请求时（IN2IF被硬件设置为1），如果该位被设置为1，并且POC_IN2DCFG寄存器的INDREN位也被设置为1，则GPTIMER1通道引脚输出被禁能。
		注意： 该位在复位后只能写入一次。复位后默认值为1。
25	GPTIMER1_IN1DRS EL	POC_IN1禁能请求选择GPTIMER1 该位用于确定POC_IN1输入检测生成的禁能请求是否选择GPTIMER1作为目标定时器。 0: GPTIMER1未被选择 1: GPTIMER1被选择 当POC_IN1生成禁能请求时（IN1IF被硬件设置为1），如果该位被设置为1，并且POC_IN1DCFG寄存器的INDREN位也被设置为1，则GPTIMER1通道引脚输出被禁能。 注意： 该位在复位后只能写入一次。
24	GPTIMER1_IN0DRS EL	POC_IN0禁能请求选择GPTIMER1 该位用于确定POC_IN0输入检测生成的禁能请求是否选择GPTIMER1作为目标定时器。 0: GPTIMER1未被选择 1: GPTIMER1被选择 当POC_IN0生成禁能请求时（IN0IF被硬件设置为1），如果该位被设置为1，并且POC_IN0DCFG寄存器的INDREN位也被设置为1，则GPTIMER1通道引脚输出被禁能。 注意： 该位在复位后只能写入一次。
23	保留	必须保持复位值。
22	GPTIMER0_CMPDR SEL	CMPn（n=0,1,2,3）禁能请求选择GPTIMER0 该位用于确定CMPn（n=0,1,2,3）输出检测生成的禁能请求是否选择GPTIMER0作为目标定时器。 0: GPTIMER0未被选择 1: GPTIMER0被选择 当CMPn生成禁能请求时（CMPnDF被硬件设置为1），如果该位被设置为1，并且CMPnDREN或GPTIMER0_CMPnDREN也被设置为1，则GPTIMER0通道引脚输出被禁能。 注意： 该位在复位后只能写入一次。
21	GPTIMER0_IN5DRS EL	POC_IN5禁能请求选择GPTIMER0 该位用于确定POC_IN5输入检测生成的禁能请求是否选择GPTIMER0作为目标定时器。 0: GPTIMER0未被选择 1: GPTIMER0被选择 当POC_IN5生成禁能请求时（IN5IF被硬件设置为1），如果该位被设置为1，并且

POC_IN5DCFG寄存器的INDREN位也被设置为1，则GPTIMER0通道引脚输出被禁能。

注意：该位在复位后只能写入一次。

20	GPTIMER0_IN4DRS EL	POC_IN4禁能请求选择GPTIMER0 该位用于确定POC_IN4输入检测生成的禁能请求是否选择GPTIMER0作为目标定时器。 0: GPTIMER0未被选择 1: GPTIMER0被选择 当POC_IN4生成禁能请求时（IN4IF被硬件设置为1），如果该位被设置为1，并且POC_IN4DCFG寄存器的INDREN位也被设置为1，则GPTIMER0通道引脚输出被禁能。 注意：该位在复位后只能写入一次。
19	GPTIMER0_IN3DRS EL	POC_IN3禁能请求选择GPTIMER0 该位用于确定POC_IN3输入检测生成的禁能请求是否选择GPTIMER0作为目标定时器。 0: GPTIMER0未被选择 1: GPTIMER0被选择 当POC_IN3生成禁能请求时（IN3IF被硬件设置为1），如果该位被设置为1，并且POC_IN3DCFG寄存器的INDREN位也被设置为1，则GPTIMER0通道引脚输出被禁能。 注意：该位在复位后只能写入一次。
18	GPTIMER0_IN2DRS EL	POC_IN2禁能请求选择GPTIMER0 该位用于确定POC_IN2输入检测生成的禁能请求是否选择GPTIMER0作为目标定时器。 0: GPTIMER0未被选择 1: GPTIMER0被选择 当POC_IN2生成禁能请求时（IN2IF被硬件设置为1），如果该位被设置为1，并且POC_IN2DCFG寄存器的INDREN位也被设置为1，则GPTIMER0通道引脚输出被禁能。 注意：该位在复位后只能写入一次。复位后默认值为1。
17	GPTIMER0_IN1DRS EL	POC_IN1禁能请求选择GPTIMER0 该位用于确定POC_IN1输入检测生成的禁能请求是否选择GPTIMER0作为目标定时器。 0: GPTIMER0未被选择 1: GPTIMER0被选择 当POC_IN1生成禁能请求时（IN1IF被硬件设置为1），如果该位被设置为1，并且POC_IN1DCFG寄存器的INDREN位也被设置为1，则GPTIMER0通道引脚输出被禁能。 注意：该位在复位后只能写入一次。
16	GPTIMER0_IN0DRS EL	POC_IN0禁能请求选择GPTIMER0 该位用于确定POC_IN0输入检测生成的禁能请求是否选择GPTIMER0作为目标定时器。

		器。
		0: GPTIMER0未被选择
		1: GPTIMER0被选择
		当POC_IN0生成禁能请求时（IN0IF被硬件设置为1），如果该位被设置为1，并且POC_IN0DCFG寄存器的INDREN位也被设置为1，则GPTIMER0通道引脚输出被禁能。
		注意： 该位在复位后只能写入一次。
15	保留	必须保持复位值。
14	TIMER2_CMPDRSEL	CMPn（n=0,1,2,3）禁能请求选择TIMER2
	L	该位用于确定CMPn（n=0,1,2,3）输出检测生成的禁能请求是否选择TIMER2作为目标定时器。
		0: TIMER2未被选择
		1: TIMER2被选择
		当CMPn生成禁能请求时（CMPnDF被硬件设置为1），如果该位被设置为1，并且CMPnDREN或TIMER2_CMPnDREN也被设置为1，则TIMER2通道引脚输出被禁能。
		注意： 该位在复位后只能写入一次。
13	TIMER2_IN5DRSEL	POC_IN5禁能请求选择TIMER2
		该位用于确定POC_IN5输入检测生成的禁能请求是否选择TIMER2作为目标定时器。
		0: TIMER2未被选择
		1: TIMER2被选择
		当POC_IN5生成禁能请求时（IN5IF被硬件设置为1），如果该位被设置为1，并且POC_IN5DCFG寄存器的INDREN位也被设置为1，则TIMER2通道引脚输出被禁能。
		注意： 该位在复位后只能写入一次。复位后默认值为1。
12	TIMER2_IN4DRSEL	POC_IN4禁能请求选择TIMER2
		该位用于确定POC_IN4输入检测生成的禁能请求是否选择TIMER2作为目标定时器。
		0: TIMER2未被选择
		1: TIMER2被选择
		当POC_IN4生成禁能请求时（IN4IF被硬件设置为1），如果该位被设置为1，并且POC_IN4DCFG寄存器的INDREN位也被设置为1，则TIMER2通道引脚输出被禁能。
		注意： 该位在复位后只能写入一次。
11	TIMER2_IN3DRSEL	POC_IN3禁能请求选择TIMER2
		该位用于确定POC_IN3输入检测生成的禁能请求是否选择TIMER2作为目标定时器。
		0: TIMER2未被选择
		1: TIMER2被选择
		当POC_IN3生成禁能请求时（IN3IF被硬件设置为1），如果该位被设置为1，并且POC_IN3DCFG寄存器的INDREN位也被设置为1，则TIMER2通道引脚输出被禁能。
		注意： 该位在复位后只能写入一次。
10	TIMER2_IN2DRSEL	POC_IN2禁能请求选择TIMER2
		该位用于确定POC_IN2输入检测生成的禁能请求是否选择TIMER2作为目标定时器。

		0: TIMER2未被选择 1: TIMER2被选择 当POC_IN2生成禁能请求时（IN2IF被硬件设置为1），如果该位被设置为1，并且POC_IN2DCFG寄存器的INDREN位也被设置为1，则TIMER2通道引脚输出被禁能。 注意： 该位在复位后只能写入一次。复位后默认值为1。
9	TIMER2_IN1DRSEL	POC_IN1禁能请求选择TIMER2 该位用于确定POC_IN1输入检测生成的禁能请求是否选择TIMER2作为目标定时器。 0: TIMER2未被选择 1: TIMER2被选择 当POC_IN1生成禁能请求时（IN1IF被硬件设置为1），如果该位被设置为1，并且POC_IN1DCFG寄存器的INDREN位也被设置为1，则TIMER2通道引脚输出被禁能。 注意： 该位在复位后只能写入一次。
8	TIMER2_IN0DRSEL	POC_IN0禁能请求选择TIMER2 该位用于确定POC_IN0输入检测生成的禁能请求是否选择TIMER2作为目标定时器。 0: TIMER2未被选择 1: TIMER2被选择 当POC_IN0生成禁能请求时（IN0IF被硬件设置为1），如果该位被设置为1，并且POC_IN0DCFG寄存器的INDREN位也被设置为1，则TIMER2通道引脚输出被禁能。 注意： 该位在复位后只能写入一次。
7	保留	必须保持复位值。
6	TIMER1_CMPDRSEL	CMPn（n=0,1,2,3）禁能请求选择TIMER1 该位用于确定CMPn（n=0,1,2,3）输出检测生成的禁能请求是否选择TIMER1作为目标定时器。 0: TIMER1未被选择 1: TIMER1被选择 当CMPn生成禁能请求时（CMPnDF被硬件设置为1），如果该位被设置为1，并且CMPnDREN或TIMER1_CMPnDREN也被设置为1，则TIMER1通道引脚输出被禁能。 注意： 该位在复位后只能写入一次。
5	TIMER1_IN5DRSEL	POC_IN5禁能请求选择TIMER1 该位用于确定POC_IN5输入检测生成的禁能请求是否选择TIMER1作为目标定时器。 0: TIMER1未被选择 1: TIMER1被选择 当POC_IN5生成禁能请求时（IN5IF被硬件设置为1），如果该位被设置为1，并且POC_IN5DCFG寄存器的INDREN位也被设置为1，则TIMER1通道引脚输出被禁能。 注意： 该位在复位后只能写入一次。
4	TIMER1_IN4DRSEL	POC_IN4禁能请求选择TIMER1 该位用于确定POC_IN4输入检测生成的禁能请求是否选择TIMER1作为目标定时器。 0: TIMER1未被选择 1: TIMER1被选择

当POC_IN4生成禁能请求时（IN4IF被硬件设置为1），如果该位被设置为1，并且POC_IN4DCFG寄存器的INDREN位也被设置为1，则TIMER1通道引脚输出被禁能。
注意：该位在复位后只能写入一次。复位后默认值为1。

3	TIMER1_IN3DRSEL	POC_IN3禁能请求选择TIMER1 该位用于确定POC_IN3输入检测生成的禁能请求是否选择TIMER1作为目标定时器。 0: TIMER1未被选择 1: TIMER1被选择 当POC_IN3生成禁能请求时（IN3IF被硬件设置为1），如果该位被设置为1，并且POC_IN3DCFG寄存器的INDREN位也被设置为1，则TIMER1通道引脚输出被禁能。 注意：该位在复位后只能写入一次。
2	TIMER1_IN2DRSEL	POC_IN2禁能请求选择TIMER1 该位用于确定POC_IN2输入检测生成的禁能请求是否选择TIMER1作为目标定时器。 0: TIMER1未被选择 1: TIMER1被选择 当POC_IN2生成禁能请求时（IN2IF被硬件设置为1），如果该位被设置为1，并且POC_IN2DCFG寄存器的INDREN位也被设置为1，则TIMER1通道引脚输出被禁能。 注意：该位在复位后只能写入一次。复位后默认值为1。
1	TIMER1_IN1DRSEL	POC_IN1禁能请求选择TIMER1 该位用于确定POC_IN1输入检测生成的禁能请求是否选择TIMER1作为目标定时器。 0: TIMER1未被选择 1: TIMER1被选择 当POC_IN1生成禁能请求时（IN1IF被硬件设置为1），如果该位被设置为1，并且POC_IN1DCFG寄存器的INDREN位也被设置为1，则TIMER1通道引脚输出被禁能。 注意：该位在复位后只能写入一次。
0	TIMER1_IN0DRSEL	POC_IN0禁能请求选择TIMER1 该位用于确定POC_IN0输入检测生成的禁能请求是否选择TIMER1作为目标定时器。 0: TIMER1未被选择 1: TIMER1被选择 当POC_IN0生成禁能请求时（IN0IF被硬件设置为1），如果该位被设置为1，并且POC_IN0DCFG寄存器的INDREN位也被设置为1，则TIMER1通道引脚输出被禁能。 注意：该位在复位后只能写入一次。

8.4.11. 状态寄存器 1（POC_STAT1）

地址偏移：0x60

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										CMP3DF	CMP2DF	CMP1DF	CMP0DF
										rc_w0	rc_w0	rc_w0	rc_w0

位/位域	名称	描述
31:4	保留	必须保持复位值。
3	CMP3DF	比较器3检测标志 该位表明比较器3输出检测是否产生禁能请求。 0: 未产生禁能请求 1: 产生禁能请求 当比较器3输出信号从低电平变为高电平时该位被硬件置1。 软件写 0 清除。
2	CMP2DF	比较器2检测标志 该位表明比较器2输出检测是否产生禁能请求。 0: 未产生禁能请求 1: 产生禁能请求 当比较器2输出信号从低电平变为高电平时该位被硬件置1。 软件写 0 清除。
1	CMP1DF	比较器1检测标志 该位表明比较器1输出检测是否产生禁能请求。 0: 未产生禁能请求 1: 产生禁能请求 当比较器1输出信号从低电平变为高电平时该位被硬件置1。 软件写 0 清除。
0	CMP0DF	比较器0检测标志 该位表明比较器0输出检测是否产生禁能请求。 0: 未产生禁能请求 1: 产生禁能请求 当比较器0输出信号从低电平变为高电平时该位被硬件置1。 软件写 0 清除。

8.4.12. 控制寄存器 1 (POC_CTL1)

地址偏移: 0x64

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				CMP3DR	CMP2DRE	CMP1DRE	CMP0DR
				EN	N	N	EN
				rwo	rwo	rwo	rwo

位/位域	名称	描述
31:4	保留	必须保持复位值。
3	CMP3DREN	比较器3禁能请求使能 该位用于确定比较器3输出检测生成的禁能请求是否对TIMERx (x=0,1,2,7) 和 GPTIMERx (x=0,1) 使能。 0: 禁能请求不使能 1: 禁能请求使能 如果该位使能,当 CMP3DF = 1 和 TIMERx_CMPDRSEL = 1 时,TIMERx(x=0,1,2,7) 通道引脚输出被禁能。 如果该位使能,当 CMP3DF = 1 和 GPTIMERx_CMPDRSEL = 1 时, GPTIMERx (x=0,1) 通道引脚输出被禁能。 注意: 该位在复位后只能写入一次。
2	CMP2DREN	比较器2禁能请求使能 该位用于确定比较器2输出检测生成的禁能请求是否对TIMERx (x=0,1,2,7) 和 GPTIMERx (x=0,1) 使能。 0: 禁能请求不使能 1: 禁能请求使能 如果该位使能,当 CMP2DF = 1 和 TIMERx_CMPDRSEL = 1 时,TIMERx(x=0,1,2,7) 通道引脚输出被禁能。 如果该位使能,当 CMP2DF = 1 和 GPTIMERx_CMPDRSEL = 1 时, GPTIMERx (x=0,1) 通道引脚输出被禁能。 注意: 该位在复位后只能写入一次。
1	CMP1DREN	比较器1禁能请求使能 该位用于确定比较器1输出检测生成的禁能请求是否对TIMERx (x=0,1,2,7) 和 GPTIMERx (x=0,1) 使能。 0: 禁能请求不使能 1: 禁能请求使能 如果该位使能,当 CMP1DF = 1 和 TIMERx_CMPDRSEL = 1 时,TIMERx(x=0,1,2,7) 通道引脚输出被禁能。 如果该位使能,当 CMP1DF = 1 和 GPTIMERx_CMPDRSEL = 1 时, GPTIMERx (x=0,1) 通道引脚输出被禁能。 注意: 该位在复位后只能写入一次。
0	CMP0DREN	比较器0禁能请求使能 该位用于确定比较器0输出检测生成的禁能请求是否对TIMERx (x=0,1,2,7) 和 GPTIMERx (x=0,1) 使能。 0: 禁能请求不使能 1: 禁能请求使能

如果该位使能,当 $CMP0DF = 1$ 和 $TIMERx\_CMPDRSEL = 1$ 时, $TIMERx(x=0,1,2,7)$ 通道引脚输出被禁能。

如果该位使能,当 $CMP0DF = 1$ 和 $GPTIMERx\_CMPDRSEL = 1$ 时, $GPTIMERx(x=0,1)$ 通道引脚输出被禁能。

注意: 该位在复位后只能写入一次。

8.4.13. 扩展控制寄存器 0 (POC_EXTCTL0)

地址偏移: 0x68

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				TIMER7_	TIMER7_	TIMER7_	TIMER7_	保留				TIMER0_C	TIMER0_C	TIMER0_C	TIMER0_C
				CMP3DRE	CMP2DRE	CMP1DRE	CMP0DRE					MP3DREN	MP2DREN	MP1DREN	MP0DREN
				N	N	N	N								
				rwo	rwo	rwo	rwo					rwo	rwo	rwo	rwo

位/位域	名称	描述
31:12	保留	必须保持复位值。
11	TIMER7_CMP3DRE N	比较器3对TIMER7禁能请求使能 该位用于确定比较器3输出检测生成的禁能请求是否对TIMER7使能。 0: 禁能请求不使能 1: 禁能请求使能 如果该位使能,当 $CMP3DF = 1$ 和 $TIMER7\_CMPDRSEL = 1$ 时, $TIMER7$ 通道引脚输出被禁能。 注意: 该位在复位后只能写入一次。
10	TIMER7_CMP2DRE N	比较器2对TIMER7禁能请求使能 该位用于确定比较器2输出检测生成的禁能请求是否对TIMER7使能。 0: 禁能请求不使能 1: 禁能请求使能 如果该位使能,当 $CMP2DF = 1$ 和 $TIMER7\_CMPDRSEL = 1$ 时, $TIMER7$ 通道引脚输出被禁能。 注意: 该位在复位后只能写入一次。
9	TIMER7_CMP1DRE N	比较器1对TIMER7禁能请求使能 该位用于确定比较器1输出检测生成的禁能请求是否对TIMER7使能。 0: 禁能请求不使能 1: 禁能请求使能

		如果该位使能，当 $CMP1DF = 1$ 和 $TIMER7\_CMPDRSEL = 1$ 时， $TIMER7$ 通道引脚输出被禁能。
		注意： 该位在复位后只能写入一次。
8	$TIMER7\_CMP0DRE$ N	比较器0对 $TIMER7$ 禁能请求使能 该位用于确定比较器0输出检测生成的禁能请求是否对 $TIMER7$ 使能。 0：禁能请求不使能 1：禁能请求使能 如果该位使能，当 $CMP0DF = 1$ 和 $TIMER7\_CMPDRSEL = 1$ 时， $TIMER7$ 通道引脚输出被禁能。 注意： 该位在复位后只能写入一次。
7:4	保留	必须保持复位值。
3	$TIMER0\_CMP3DRE$ N	比较器3对 $TIMER0$ 禁能请求使能 该位用于确定比较器3输出检测生成的禁能请求是否对 $TIMER0$ 使能。 0：禁能请求不使能 1：禁能请求使能 如果该位使能，当 $CMP3DF = 1$ 和 $TIMER0\_CMPDRSEL = 1$ 时， $TIMER7$ 通道引脚输出被禁能。 注意： 该位在复位后只能写入一次。
2	$TIMER0\_CMP2DRE$ N	比较器2对 $TIMER0$ 禁能请求使能 该位用于确定比较器2输出检测生成的禁能请求是否对 $TIMER0$ 使能。 0：禁能请求不使能 1：禁能请求使能 如果该位使能，当 $CMP3DF = 1$ 和 $TIMER0\_CMPDRSEL = 1$ 时， $TIMER7$ 通道引脚输出被禁能。 注意： 该位在复位后只能写入一次。
1	$TIMER0\_CMP1DRE$ N	比较器1对 $TIMER0$ 禁能请求使能 该位用于确定比较器1输出检测生成的禁能请求是否对 $TIMER0$ 使能。 0：禁能请求不使能 1：禁能请求使能 如果该位使能，当 $CMP1DF = 1$ 和 $TIMER0\_CMPDRSEL = 1$ 时， $TIMER7$ 通道引脚输出被禁能。 注意： 该位在复位后只能写入一次。
0	$TIMER0\_CMP0DRE$ N	比较器0对 $TIMER0$ 禁能请求使能 该位用于确定比较器0输出检测生成的禁能请求是否对 $TIMER0$ 使能。 0：禁能请求不使能 1：禁能请求使能 如果该位使能，当 $CMP0DF = 1$ 和 $TIMER0\_CMPDRSEL = 1$ 时， $TIMER7$ 通道引脚输出被禁能。 注意： 该位在复位后只能写入一次。

8.4.14. 扩展控制寄存器 1 (POC_EXTCTL1)

地址偏移: 0x6C

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				GPTIMER1_CMP3D	GPTIMER1_CMP2D	GPTIMER1_CMP1D	GPTIMER1_CMP0D	保留				GPTIMER0_CMP3D	GPTIMER0_CMP2D	GPTIMER0_CMP1D	GPTIMER0_CMP0D
				REN	REN	REN	REN					REN	REN	REN	REN
				rwo	rwo	rwo	rwo					rwo	rwo	rwo	rwo
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				TIMER2_CMP3DRE	TIMER2_CMP2DRE	TIMER2_CMP1DRE	TIMER2_CMP0DRE	保留				TIMER1_CMP3DRE	TIMER1_CMP2DRE	TIMER1_CMP1DRE	TIMER1_CMP0DRE
				N	N	N	N					N	N	N	N
				rwo	rwo	rwo	rwo					rwo	rwo	rwo	rwo

位/位域	名称	描述
31:28	保留	必须保持复位值。
27	GPTIMER1_CMP3D REN	比较器3对GPTIMER1禁能请求使能 该位用于确定比较器3输出检测生成的禁能请求是否对GPTIMER1使能。 0: 禁能请求不使能 1: 禁能请求使能 如果该位使能, 当CMP3DF = 1和GPTIMER1_CMPDRSEL = 1时, GPTIMER1通道引脚输出被禁能。 注意: 该位在复位后只能写入一次。
26	GPTIMER1_CMP2D REN	比较器2对GPTIMER1禁能请求使能 该位用于确定比较器2输出检测生成的禁能请求是否对GPTIMER1使能。 0: 禁能请求不使能 1: 禁能请求使能 如果该位使能, 当CMP2DF = 1和GPTIMER1_CMPDRSEL = 1时, GPTIMER1通道引脚输出被禁能。 注意: 该位在复位后只能写入一次。
25	GPTIMER1_CMP1D REN	比较器1对GPTIMER1禁能请求使能 该位用于确定比较器1输出检测生成的禁能请求是否对GPTIMER1使能。 0: 禁能请求不使能 1: 禁能请求使能 如果该位使能, 当CMP1DF = 1和GPTIMER1_CMPDRSEL = 1时, GPTIMER1通道引脚输出被禁能。 注意: 该位在复位后只能写入一次。
24	GPTIMER1_CMP0D REN	比较器0对GPTIMER1禁能请求使能 该位用于确定比较器0输出检测生成的禁能请求是否对GPTIMER1使能。

		0: 禁能请求不使能 1: 禁能请求使能 如果该位使能, 当CMP0DF = 1和GPTIMER1_CMPDRSEL = 1时, GPTIMER1通道引脚输出被禁能。 注意: 该位在复位后只能写入一次。
23:20	保留	必须保持复位值。
19	GPTIMER0_CMP3DREN	比较器3对GPTIMER0禁能请求使能 该位用于确定比较器3输出检测生成的禁能请求是否对GPTIMER0使能。 0: 禁能请求不使能 1: 禁能请求使能 如果该位使能, 当CMP3DF = 1和GPTIMER0_CMPDRSEL = 1时, GPTIMER0通道引脚输出被禁能。 注意: 该位在复位后只能写入一次。
18	GPTIMER0_CMP2DREN	比较器2对GPTIMER0禁能请求使能 该位用于确定比较器2输出检测生成的禁能请求是否对GPTIMER0使能。 0: 禁能请求不使能 1: 禁能请求使能 如果该位使能, 当CMP2DF = 1和GPTIMER0_CMPDRSEL = 1时, GPTIMER0通道引脚输出被禁能。 注意: 该位在复位后只能写入一次。
17	GPTIMER0_CMP1DREN	比较器1对GPTIMER0禁能请求使能 该位用于确定比较器1输出检测生成的禁能请求是否对GPTIMER0使能。 0: 禁能请求不使能 1: 禁能请求使能 如果该位使能, 当CMP1DF = 1和GPTIMER0_CMPDRSEL = 1时, GPTIMER0通道引脚输出被禁能。 注意: 该位在复位后只能写入一次。
16	GPTIMER0_CMP0DREN	比较器0对GPTIMER0禁能请求使能 该位用于确定比较器0输出检测生成的禁能请求是否对GPTIMER0使能。 0: 禁能请求不使能 1: 禁能请求使能 如果该位使能, 当CMP0DF = 1和GPTIMER0_CMPDRSEL = 1时, GPTIMER0通道引脚输出被禁能。 注意: 该位在复位后只能写入一次。
15:12	保留	必须保持复位值。
11	TIMER2_CMP3DREN	比较器3对TIMER2禁能请求使能 该位用于确定比较器3输出检测生成的禁能请求是否对TIMER2使能。 0: 禁能请求不使能 1: 禁能请求使能 如果该位使能, 当CMP3DF = 1和TIMER2_CMPDRSEL = 1时, TIMER2通道引脚输

		出被禁能。
		注意： 该位在复位后只能写入一次。
10	TIMER2_CMP2DRE N	比较器2对TIMER2禁能请求使能 该位用于确定比较器2输出检测生成的禁能请求是否对TIMER2使能。 0：禁能请求不使能 1：禁能请求使能 如果该位使能，当CMP2DF = 1和TIMER2_CMPDRSEL = 1时，TIMER2通道引脚输出被禁能。 注意：该位在复位后只能写入一次。
9	TIMER2_CMP1DRE N	比较器1对TIMER2禁能请求使能 该位用于确定比较器1输出检测生成的禁能请求是否对TIMER2使能。 0：禁能请求不使能 1：禁能请求使能 如果该位使能，当CMP1DF = 1和TIMER2_CMPDRSEL = 1时，TIMER2通道引脚输出被禁能。 注意：该位在复位后只能写入一次。
8	TIMER2_CMP0DRE N	比较器0对TIMER2禁能请求使能 该位用于确定比较器0输出检测生成的禁能请求是否对TIMER2使能。 0：禁能请求不使能 1：禁能请求使能 如果该位使能，当CMP0DF = 1和TIMER2_CMPDRSEL = 1时，TIMER2通道引脚输出被禁能。 注意：该位在复位后只能写入一次。
7:4	保留	必须保持复位值。
3	TIMER1_CMP3DRE N	比较器3对TIMER1禁能请求使能 该位用于确定比较器3输出检测生成的禁能请求是否对TIMER1使能。 0：禁能请求不使能 1：禁能请求使能 如果该位使能，当CMP3DF = 1和TIMER1_CMPDRSEL = 1时，TIMER1通道引脚输出被禁能。 注意：该位在复位后只能写入一次。
2	TIMER1_CMP2DRE N	比较器2对TIMER1禁能请求使能 该位用于确定比较器2输出检测生成的禁能请求是否对TIMER1使能。 0：禁能请求不使能 1：禁能请求使能 如果该位使能，当CMP2DF = 1和TIMER1_CMPDRSEL = 1时，TIMER1通道引脚输出被禁能。 注意：该位在复位后只能写入一次。
1	TIMER1_CMP1DRE N	比较器1对TIMER1禁能请求使能 该位用于确定比较器1输出检测生成的禁能请求是否对TIMER1使能。

0: 禁能请求不使能

1: 禁能请求使能

如果该位使能，当CMP1DF = 1和TIMER1_CMPDRSEL = 1时，TIMER1通道引脚输出被禁能。

注意：该位在复位后只能写入一次。

0 TIMER1_CMP0DRE 比较器0对TIMER1禁能请求使能
N

该位用于确定比较器0输出检测生成的禁能请求是否对TIMER1使能。

0: 禁能请求不使能

1: 禁能请求使能

如果该位使能，当CMP0DF = 1和TIMER1_CMPDRSEL = 1时，TIMER1通道引脚输出被禁能。

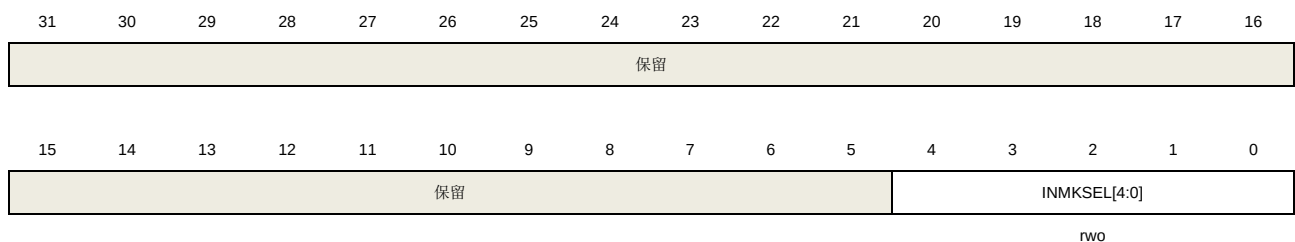
注意：该位在复位后只能写入一次。

8.4.15. 输入 n 检测屏蔽寄存器（POC_INnDMK）（n=0...5）

地址偏移：0x70 + 0x04 * n

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
31:5	保留	必须保持复位值。
4:0	INMKSEL[4:0]	POC_INn输入检测屏蔽源选择 这些位用于选择一个定时器通道信号来屏蔽POC_INn引脚输入。当所选通道信号为高时，来自POC_INn引脚的输入信号被屏蔽。 00000: 不屏蔽 00001: TIMER0_CH0 00010: TIMER0_CH1 00011: TIMER0_CH2 00100: TIMER0_CH3 00101: TIMER0_MCH0 00110: TIMER0_MCH1 00111: TIMER0_MCH2 01000: TIMER0_MCH3 01001: TIMER7_CH0 01010: TIMER7_CH1 01011: TIMER7_CH2

01100: TIMER7_CH3
01101: TIMER7_MCH0
01110: TIMER7_MCH1
01111: TIMER7_MCH2
10000: TIMER7_MCH3
10001: TIMER1_CH0
10010: TIMER1_CH1
10011: TIMER1_CH2
10100: TIMER1_CH3
10101: TIMER2_CH0
10110: TIMER2_CH1
10111: TIMER2_CH2
11000: TIMER2_CH3
11001: GPTIMER0_CH0
11010: GPTIMER0_CH1
11011: GPTIMER1_CH0
11100: GPTIMER1_CH1

其他：保留。

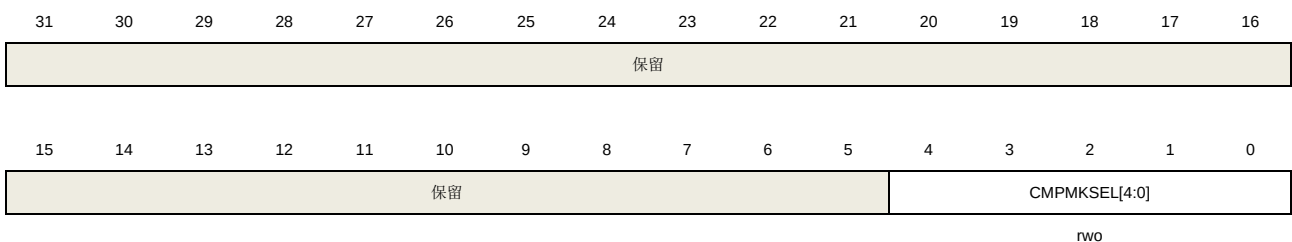
注意：该位在复位后只能写入一次。

8.4.16. 比较器 n 检测屏蔽寄存器 (POC_CMPnDMK) (n=0...3)

地址偏移：0x90 + 0x04 * n

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
31:5	保留	必须保持复位值。
4:0	CMPMKSEL[4:0]	比较器n输出检测屏蔽源选择 这些位用于选择一个定时器通道信号来屏蔽比较器n输出信号。当所选通道信号为高时，来自比较器n的输出信号被屏蔽。 请参考INMKSEL[4:0]描述。 注意：该位在复位后只能写入一次。

9. GPTIMER 输出控制器（GTOC）

9.1. 简介

GTOC 可以生成请求来关闭 GPTIMER 输出引脚。被关闭的引脚可输出高阻（Hi-Z）态。输出关闭请求源可以来自芯片上的不同模块，如 CMP 和 GPTIMER 等。

9.2. 主要特征

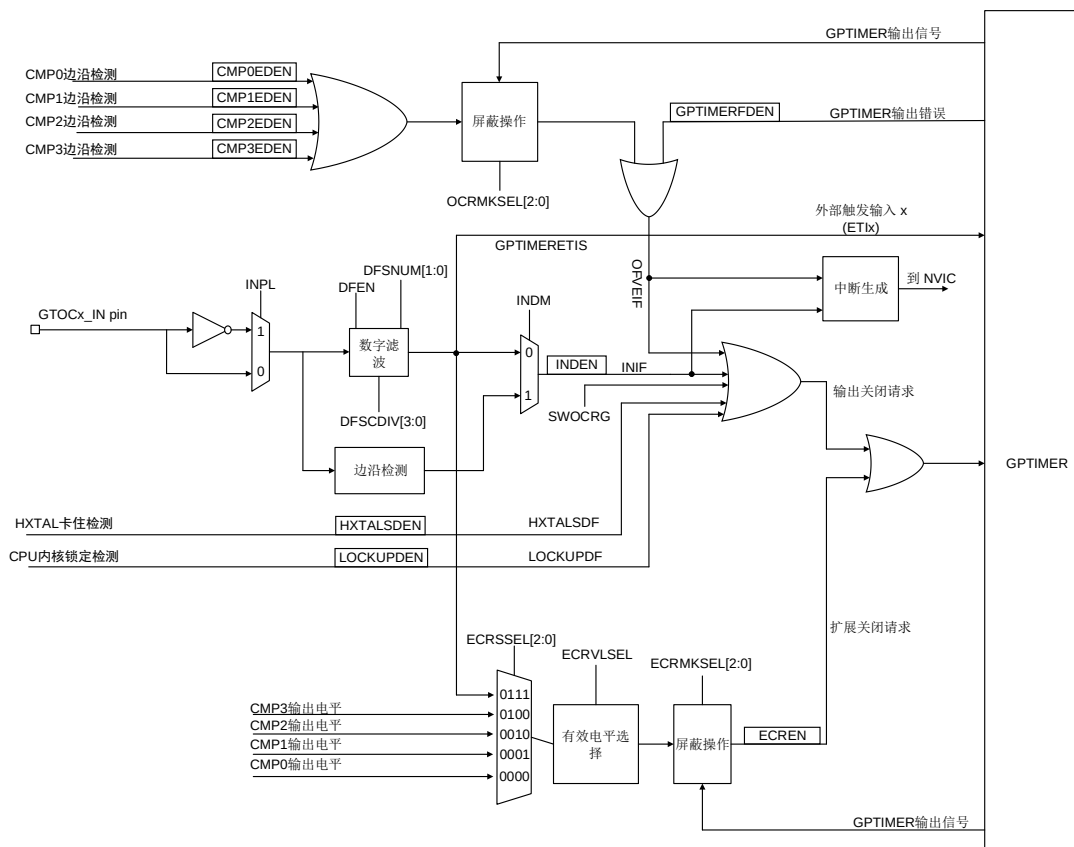
GTOC 的主要特征如下：

- GTOCx_IN (x=0...3) 引脚输入检测：
 - 边沿检测；
 - 电平检测；
 - 输入反相控制；
 - 数字滤波；
- GPTIMER 输出关闭请求源：
 - GTOCx_IN (x=0...3) 引脚输入检测；
 - GPTIMER 输出错误检测（在通道 0 和通道 1 上同时低电平或者高电平输出）；
 - 软件生成；
 - HXTAL 卡住事件；
 - CPU 内核锁定事件；
 - 比较器输出检测；
- 请求屏蔽功能：
 - 比较器边沿检测生成的输出关闭请求可以被 GPTIMER 输出信号屏蔽；
 - 扩展关闭请求可以被 GPTIMER 输出信号屏蔽；
- 中断生成：
 - GTOCx_IN (x=0...3) 有效信号输入；
 - GPTIMER 输出错误或者比较器有效边沿输出；

9.3. 功能描述

GTOCx (x=0...3) 模块的框图如下：

图 9-1. GTOCx (x=0...3) 模块框图



GTOCx (x=0...3) 的目标通道如下所示:

表 9-1. 输出关闭控制的 GPTIMER 通道

	GPTIMER	
目标 GPTIMER	GPTIMER0	GPTIMER1
目标通道	GPTIMER0_CH0 GPTIMER0_CH1	GPTIMER1_CH0 GPTIMER1_CH1

9.3.1. GTOC 时钟

GTOC 模块与 AHB 总线连接, 时钟与 HCLK 同步。

9.3.2. 输出关闭功能

GTOCx (x=0...3) 可以生成两种关闭请求, 一种是基于标志位和软件设置的输出关闭请求, 另一种是基于电平检测的扩展关闭请求。

GTOC 输入检测

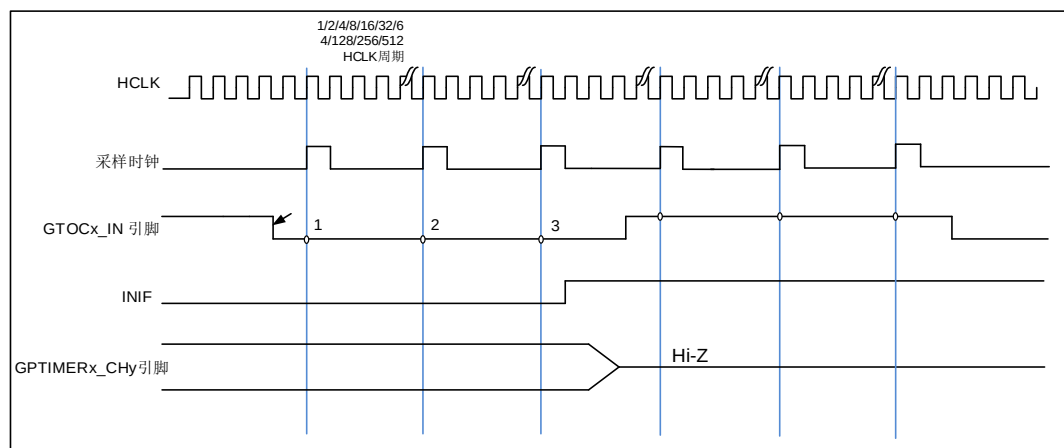
每个 GTOC 模块都有一个外部输入引脚。GTOCx (x=0...3) 可以对 GTOCx_IN (x=0...3) 引

脚输入信号进行电平检测或边沿检测，在 GTOCx_CFG ($x=0\dots3$) 寄存器中设置 INDM 位可以选择电平检测或边沿检测。如果使能 GTOCx_IN ($x=0\dots3$) 输入检测 ($\text{INDEN} = 1$)，一旦有效边沿或电平输入 GTOCx_IN ($x=0\dots3$) 引脚， GTOCx_CFG ($x=0\dots3$) 寄存器中的中断标志位 INIF 将被硬件设置为 1，并生成输出关闭请求。

通过设置 GTOCx_CFG ($x=0\dots3$) 寄存器中的 INPL 位，可以使 GTOCx_IN ($x=0\dots3$) 引脚输入信号反相。如果 INPL 位设置为 0，则进行上升沿和高电平检测。否则，进行下降沿和低电平检测。

通过设置 GTOCx_CFG ($x=0\dots3$) 寄存器中的 DFEN 位来启用数字滤波功能。 GTOCx_CFG ($x=0\dots3$) 寄存器中的 $\text{DFSNUM}[1:0]$ 位可以决定同一电平连续采样的有效次数。如果同一电平被连续采样特定次数，则该电平将被视为滤波后的输出。滤波器的采样频率可以通过在 GTOCx_CFG ($x=0\dots3$) 寄存器中设置 $\text{DFSCDIV}[3:0]$ 位来确定。

图 9-2. 具有数字滤波的 GTOCx_IN 引脚输入检测($\text{DFSNUM}[1:0] = 2'b00, \text{INPL}=1, \text{INDM}=0$)



GTOC 引脚输入检测的软件配置流程如下：

1. 配置GTOC引脚为对应的GPIO复用功能。
2. 配置 GTOCx_CFG ($x=0\dots3$) 寄存器中的输入极性位 INPL 。
3. 配置 GTOCx_CFG ($x=0\dots3$) 寄存器中的检测模式位 INDM 。
4. 如果用到数字滤波器功能，配置 GTOCx_CFG ($x=0\dots3$) 寄存器中的 $\text{DFSNUM}[1:0]$ 位， $\text{DFSCDIV}[3:0]$ 位和使能位 DFEN 。
5. 配置 GTOCx_CFG ($x=0\dots3$) 寄存器中的检测使能位 INDEN 。

注意：如果使用 GTOC 低电平检测时，必须配置对应 GPIO 的 AF 功能，且必须在 GTOC 检测使能之前配置对应的 AF 功能。

GTOCx_IN ($x=0\dots3$) 引脚输入信号可用于生成扩展关闭请求，也可作为 GPTIMER 的外部触发输入。

清除 GTOCx_CFG ($x=0\dots3$) 寄存器中的中断标志位 INIF 可以停止输出关闭请求。

GPTIMER 输出检测

当 GPTIMER 通道 0 和通道 1 同时输出高电平或低电平时，如果打开相应的检测开关（GPTIMER 模块中的 CHHOUTDET 位或 CHLOUTDET 位），并使能 GPTIMER 输出错误检测功能（GPTIMERFDEN = 1），则 GTOCx_CFG（x=0...3）寄存器中的中断标志位 OFVEIF 会被硬件置 1，且产生输出关闭请求。请参考 GPTIMER 章节了解更多细节。

清除 GTOCx_CFG（x=0...3）寄存器中的中断标志位 OFVEIF 可以停止输出关闭请求。

系统故障检测

HXTAL 卡住事件

当 HXTAL 卡住事件发生时，如果 HXTAL 卡住检测使能位为 1（HXTALSDEN = 1），则 GTOCx_CFG（x=0...3）寄存器中的检测标志位 HXTALSDF 会被硬件置 1，并生成输出关闭请求。

清除 GTOCx_CFGx（x=0...3）寄存器中的检测标志位 HXTALSDF 可以停止输出关闭请求。

CPU 锁定事件

当检测到 CPU 锁定事件时，只有当 GTOCx_CFG（x=0...3）寄存器中的 LOCKUPDEN 位为 1 时，GTOCx_CFG（x=0...3）寄存器中的 LOCKUPDF 标志才会变为 1，并生成输出关闭请求。

清除 GTOCx_CFG（x=0...3）寄存器中的检测标志位 LOCKUPDF 可以停止输出关闭请求。

软件关闭输出

通过将 GTOCx_CFG（x=0...3）寄存器中的 SWOCRG 位设置为 1，可以生成输出关闭请求。

将 0 写入 GTOCx_CFG（x=0...3）寄存器中的 SWOCRG 位可以停止输出关闭请求。

比较器输出检测

对于比较器边沿检测，如果 CMPn（n=0...3）输出有效边沿，并且相应的比较器边沿检测使能位设置为 1（CMPnEDEN = 1），则中断标志位 OFVEIF 将被硬件置 1，且生成输出关闭请求。

比较器输出电平检测可以生成扩展关闭请求。ECRSSEL[2:0]位域可以用来选择一个 CMPn（n=0...3）输出信号或一个 GTOCx_IN（x=0...3）引脚输入信号来生成一个扩展关闭请求。如果选择的信号与 GTOCx_ECRCTL（x=0...3）寄存器中 ECRVLSSEL 位设置的有效电平相匹配，则将生成扩展关闭请求。只有在相应的扩展关闭使能位设置为 1（ECREN = 1）时，扩展关闭请求才能输出到 GPTIMER 模块。

由比较器边沿检测产生的输出关闭请求可以通过清除中断标志位 OFVEIF 来停止。

当比较器输出电平或 GTOCx_IN（x=0...3）引脚输入电平与 GTOCx_ECRCTL（x=0...3）寄存器中的 ECRVLSSEL 位设置的有效电平不匹配时，扩展的关闭请求可以被停止。

输出关闭屏蔽功能

由 CMPn (n=0...3) 边沿检测产生的输出关闭请求和扩展关闭请求可以被 GPTIMER 通道输出信号屏蔽。当所选通道信号为高电平时，关闭请求会被屏蔽。

在 GTOCx_OCRCTL (x=0...3) 寄存器中设置 OCRMKSEL[2:0]位可以选择 GPTIMER 通道信号来屏蔽由 CMPn (n=0...3) 边沿检测产生的输出关闭请求。

在 GTOCx_EXTCTL (x=0...3) 寄存器中设置 ECRMKSEL[2:0]位可以选择 GPTIMER 通道信号来屏蔽扩展关闭请求。

9.3.3. GTOC 中断

下面的检测可以产生中断：

- 与中断标志 INIF 相关的 GTOCx_IN (x=0...3) 引脚输入检测。
- 与中断标志 OFVEIF 相关的 GPTIMER 输出故障检测。
- 与中断标志 OFVEIF 相关的 CMPn (n=0...3) 边沿检测。

每个 GTOC 都有一个中断。GTOC0、GTOC1、GTOC2 和 GTOC3 的中断被映射成四个不同的中断向量 ISR。

9.4. GTOC 寄存器

GTOC基地址：0x4800 4800

9.4.1. GTOCx 配置寄存器（GTOCx_CFG）（x=0...3）

地址偏移：0x00 + 0x10 * x

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
DFSCDIV[1:0]	DFEN	INPL	DFSNUM[1:0]	INDM	DFSCDIV[3:2]	保留							SWOCRG	GPTIMER	ETIS
rw	rw	rw	rw	rw	rw								rw	r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				CMP3EDE	CMP2EDE	CMP1EDE	CMP0EDE	LOCKUPD	HXTALSD	GPTIMER	INDEN	LOCKUPD	HXTALSD	OFVEIF	INIF
				N	N	N	N	EN	EN	FDEN		F	F		
				rwo	rwo	rwo	rwo	rwo	rwo	rwo	rwo	rc_w0	rc_w0	rc_w0	rc_w0

位/位域	名称	描述
31:30	DFSCDIV[1:0]	GTOCx数字滤波器采样时钟分频 DFSCDIV[3:2]和DFSCDIV[1:0]位用于选择数字滤波器的采样频率。 0000: 采样频率为 f_{HCLK} 0001: 采样频率为 $f_{HCLK}/8$ 0010: 采样频率为 $f_{HCLK}/32$ 0011: 采样频率为 $f_{HCLK}/128$ 0100: 采样频率为 $f_{HCLK}/2$ 0101: 采样频率为 $f_{HCLK}/4$ 0110: 采样频率为 $f_{HCLK}/16$ 0111: 采样频率为 $f_{HCLK}/64$ 1000: 采样频率为 $f_{HCLK}/256$ 1001: 采样频率为 $f_{HCLK}/512$ 其他: 保留。
29	DFEN	GTOCx数字滤波器使能 0: 数字滤波器禁止 1: 数字滤波器使能
28	INPL	GTOCx_IN 输入极性 0: 输入不反相 1: 输入反相
27:26	DFSNUM[1:0]	GTOCx数字滤波器采样数 该位域被用于选择数字滤波器的有效采样数 00: 同一值被连续采样 3 次。

		01: 同一值被连续采样 4 次。 10: 同一值被连续采样 5 次。 11: 同一值被连续采样6次。
25	INDM	GTOCx_IN 输入检测模式 0: 电平检测 1: 边沿检测 注意: 当 INPL 为 0 时, 高电平 (INDM=0) 或上升沿 (INDM=1) 为有效检测信号。 当 INPL 为 1 时, 低电平 (INDM=0) 或下降沿 (INDM=1) 是有效的检测信号。
24:23	DFSCDIV[3:2]	参考 DFSCDIV[1:0]位域。
22:18	保留	必须保持复位值。
17	SWOCRG	软件生成输出关闭请求 0: 软件不生成输出关闭请求 1: 软件生成输出关闭请求
16	GPTIMERETIS	GPTIMER外部触发输入x状态 GPTIMER外部触发输入x是经过反相器和滤波器的GTOCx_IN输入信号。 0: 低电平 1: 高电平 注意: 该位是只读的。
15:12	保留	必须保持复位值。
11	CMP3EDEN	CMP3有效边沿检测使能 0: 禁止 1: 使能 注意: 该位在复位后只能写入一次。
10	CMP2EDEN	CMP2有效边沿检测使能 0: 禁止 1: 使能 注意: 该位在复位后只能写入一次。
9	CMP1EDEN	CMP1有效边沿检测使能 0: 禁止 1: 使能 注意: 该位在复位后只能写入一次。
8	CMP0EDEN	CMP0有效边沿检测使能 0: 禁止 1: 使能 注意: 该位在复位后只能写入一次。
7	LOCKUPDEN	CPU锁定检测使能 0: 禁止 1: 使能

注意： 该位在复位后只能写入一次。		
6	HXTALSDEN	HXTAL卡住检测使能 0：禁止 1：使能 注意： 该位在复位后只能写入一次。
5	GPTIMERFDEN	GPTIMER输出错误检测使能 0：禁止 1：使能 注意： 该位在复位后只能写入一次。
4	INDEN	GTOCx_IN输入检测使能 0：禁止 1：使能 注意： 该位在复位后只能写入一次。
3	LOCKUPDF	CPU锁定检测标志 0：CPU锁定未被检测到 1：CPU锁定被检测到
2	HXTALSDF	HXTAL卡住检测标志 0：HXTAL卡住未被检测到 1：HXTAL卡住被检测到
1	OFVEIF	GPTIMER输出错误或CMP有效边沿中断标志 当检测到GPTIMER输出错误或检测到CMP有效边沿时，该位由硬件置1。 0：没有检测到 GPTIMER 输出错误和 CMP 有效边沿。 1：检测到 GPTIMER 输出错误或 CMP 有效边沿。
0	INIF	GTOCx_IN输入中断标志 当 GTOCx_IN 引脚输入信号为有效电平（INDM=0）或有效边沿（INDM=1）时，该位由硬件置1。 0：没有检测到有效输入 1：检测到有效输入

9.4.2. GTOCx 输出关闭请求控制寄存器（GTOCx_OCRCTL）（x=0...3）

地址偏移：0x04 + 0x10 * x

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													OCRMKSEL[2:0]		

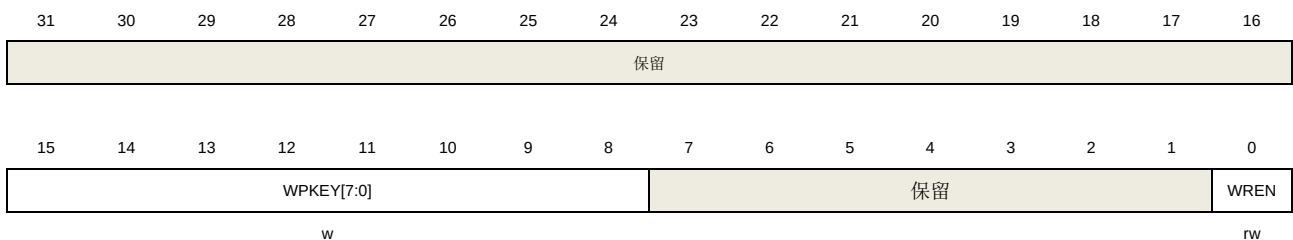
位/位域	名称	描述
31:3	保留	必须保持复位值。
2:0	OCRMKSEL[2:0]	输出关闭请求屏蔽源选择 当所选输出通道为高电平时，可以屏蔽CMP有效边沿检测生成的输出关闭请求。 000: 不屏蔽 001: GPTIMER0_CH0. 010: GPTIMER0_CH1. 011: GPTIMER1_CH0. 100: GPTIMER1_CH1. 其他: 保留。 注意: 该位在复位后只能写入一次。

9.4.3. GTOCx 写保护寄存器 (GTOCx_WP) (x=0...3)

地址偏移: $0x08 + 0x10 * x$

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
31:16	保留	必须保持复位值。
15:8	WPKEY[7:0]	写保护密钥 这些位决定是否可以将新值写入 GTOCx_WP 寄存器。0xA5 必须在写入其他位的同时写入 WPKEY[7:0]，否则写入新值无效。这些位读取为 0x00。
7:1	保留	必须保持复位值。
0	WREN	GTOCx_ECRCTL 寄存器写使能 该位由软件置 1 或清 0。当该位设置为 0 时，可以写 GTOCx_ECRCTL 寄存器。 0: GTOCx_ECRCTL 寄存器写使能。 1: GTOCx_ECRCTL 寄存器写禁能

9.4.4. GTOCx 扩展关闭请求控制寄存器 (GTOCx_ECRCTL) (x=0...3)

地址偏移: $0x0C + 0x10 * x$

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		ECRMKSEL[2:0]				保留		ECRVLSEL	保留		ECRSSEL[2:0]		保留		ECREN
rw		rw				rw		rw	rw		rw		rw		rw

位/位域	名称	描述
31:14	保留	必须保持复位值。
13:11	ECRMKSEL[2:0]	扩展关闭请求屏蔽源选择 当选择的输出通道是高电平时，扩展关闭请求可以被屏蔽。 000：不屏蔽 001：GPTIMER0_CH0 010：GPTIMER0_CH1 011：GPTIMER1_CH0 100：GPTIMER1_CH1 其他：保留。
10:9	保留	必须保持复位值。
8	ECRVLSEL	扩展关闭请求有效电平选择 当ECRSSEL[2:0]位域选择的信号达到有效电平时，产生扩展关闭请求。 0：低电平有效 1：高电平有效
7	保留	必须保持复位值。
6:4	ECRSSEL[2:0]	扩展关闭请求源选择 000：CMP0 输出 001：CMP1 输出 010：CMP2 输出 100：CMP3 输出 111：GTOCx_IN 输入 其他：保留。
3:1	保留	必须保持复位值。
0	ECREN	扩展关闭请求使能 0：禁能 1：使能

10. 循环冗余校验管理单元（CRC）

10.1. 简介

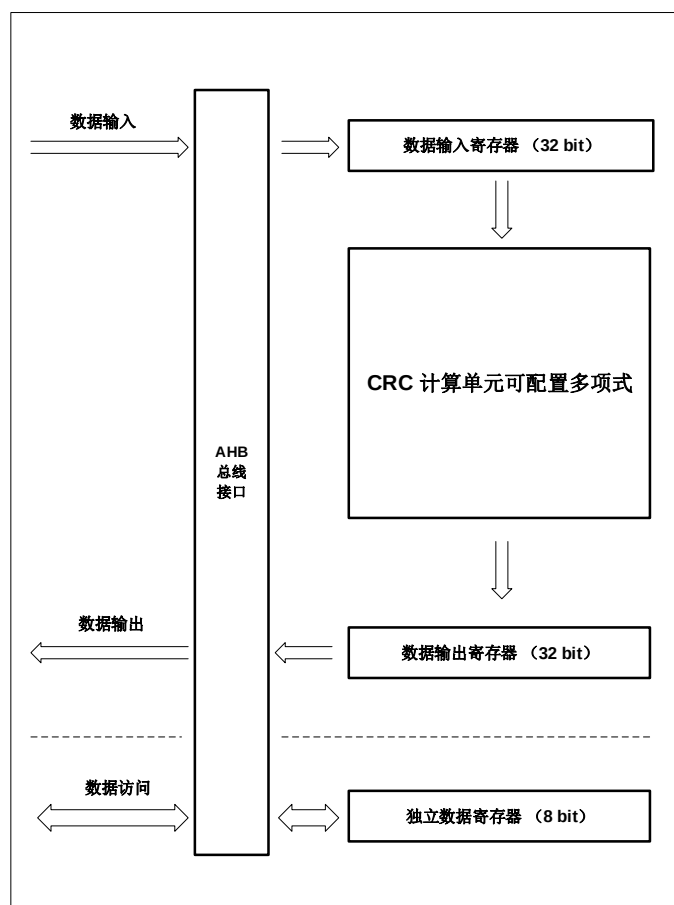
循环冗余校验码是一种用在数字网络和存储设备上的差错校验码，可以校验原始数据的偶然误差。

CRC 计算管理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。

10.2. 主要特性

- 支持7/8/16/32位数据输入；
- 对于7（8）/16/32位的输入数据长度，计算周期分别为1/2/4个AHB时钟周期；
- 对于32位输入数据长度，计算周期可为1个AHB时钟周期（32位并行CRC计算带有两个固定多项式）；
- 用户可以配置多项式及多项式长度；
- CRC复位后，用户可以配置计算初值；
- 配有与计算无关的独立8位寄存器，可以供其他任何外设使用。

图 10-1. CRC 计算单元框图



10.3. 功能描述

- CRC计算单元可以用来计算32位的原始数据，CRC_DATA寄存器接收原始数据并存储计算结果。
如果不通过软件设置CRC_CTL寄存器的方式来清除CRC_DATA寄存器，新输入的原始数据将会基于前一次CRC_DATA寄存器中的结果进行计算。
对于32/16/8（7）位的数据长度，CRC的计算分别要花费4/2/1个AHB的时钟周期。在此期间，因为32位输入缓存的原因，AHB总线将不会被挂起。对于32位输入数据长度，通过在CRC_CTL中配置PS[1:0]=00和PAR_32=1，计算周期可以是1个AHB时钟周期。
- 此模块提供了一个8位的独立寄存器CRC_FDATA，CRC_FDATA与CRC计算无关，任何时候都可以进行独立的读写操作。
- 逆序功能可以交换输入输出数据的位序。
输入数据可选择三种逆序形式。
以原始数据0x1A2B3C4D为例：
1) 按字节逆序：
32位数据被分成四组，组内完成颠倒。逆序后的数据为：0x58D43CB2
2) 按半字逆序：
32位数据被分成两组，组内完成颠倒。逆序后的数据为：0xD458B23C
3) 按字逆序：
32位数据被分成一组，组内完成颠倒，逆序后的数据为：0xB23CD458
对于输出数据来说，逆序形式为按字逆序。
例如：当REV_O=1，计算结果0x22CC4488将被逆序成0x11223344。
- 用户可配置的初始计算数据。
当RST位置位或对CRC_IDATA寄存器进行写操作时，CRC_DATA寄存器将自动初始化为CRC_IDATA寄存器中的值。
- 用户配置多项式，当选择32位并行计算(PS= 2'b00, PAR_32=1)时，只有两个固定多项式。通过配置PS[1:0]，用户可以选择有效多项式和输出数据位宽。如果多项式少于32位，那么输入和输出数据的高位无效。在更改PS[1:0]位、PAR_32位(在CRC_CTL寄存器中)或多项式之前，用户必须复位CRC单元。

10.4. CRC 寄存器

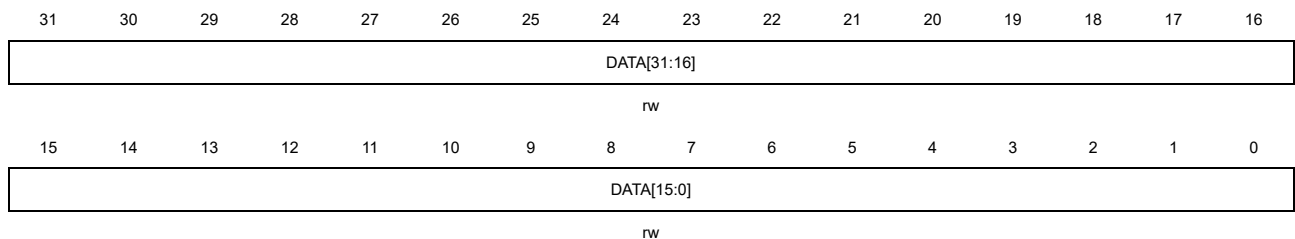
CRC基地址：0x4002 3000

10.4.1. 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。



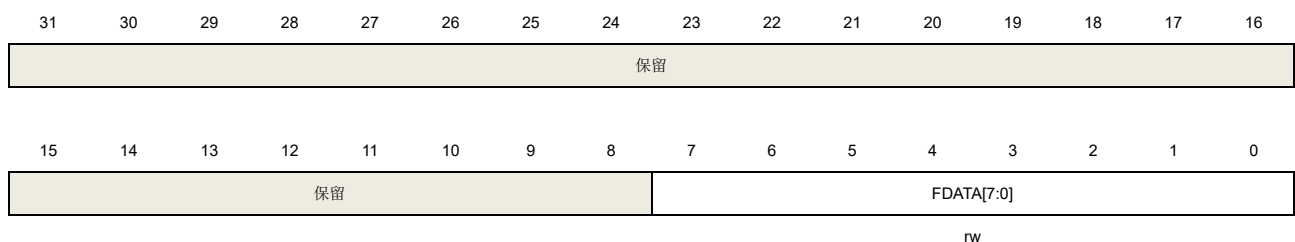
位/位域	名称	描述
31:0	DATA[31:0]	CRC 计算结果位 软件可读可写。 该寄存器用于接收待计算的新数据，直接将其写入即可。刚写入的数据不能被读出来因为读取该寄存器得到的是上次 CRC 计算的结果。

10.4.2. 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。



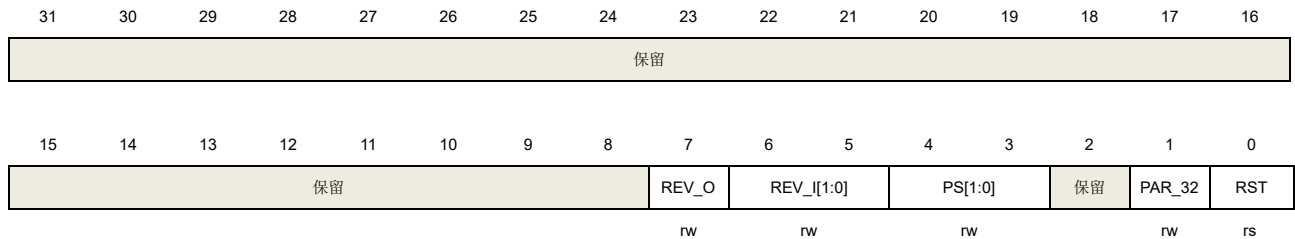
位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	FDATA[7:0]	独立数据寄存器位 软件可读可写。 这些位与 CRC 计算无关。该字节能被任何其他外设用于其他任何目的。该字节不受 CRC_CTL 寄存器的影响。

10.4.3. 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 当 PAR_32 = 0 时，采用 8 位 CRC 并行计算单元。 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 当 PAR_32 = 1 时，采用 32 位 CRC 并行计算单元，具有两个固定多项式。 00: 32 位 CRC 并行计算，POLY=0x4C11DB7 01: 32 位 CRC 并行计算，POLY=0x1EDC6F41 10:保留 11:保留
2	保留	必须保持复位值。
1	PAR_32	并行计算方式 1: CRC 工作在 32 位并行计算模式。 0: CRC 工作在 8 位并行计算模式。
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

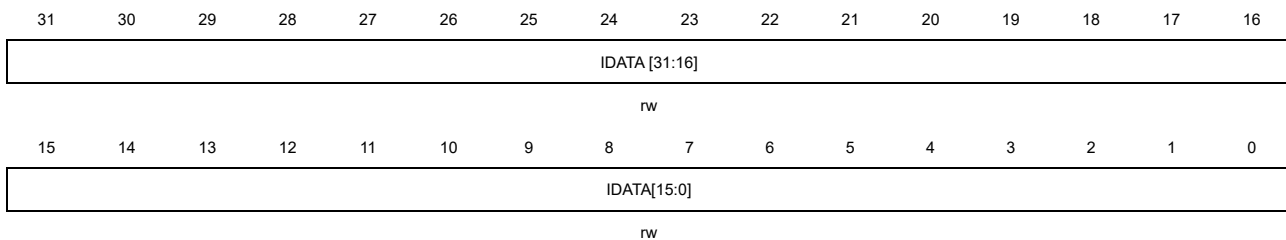
自动清零。该位对 CRC_FDATA 寄存器没有影响。

10.4.4. 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



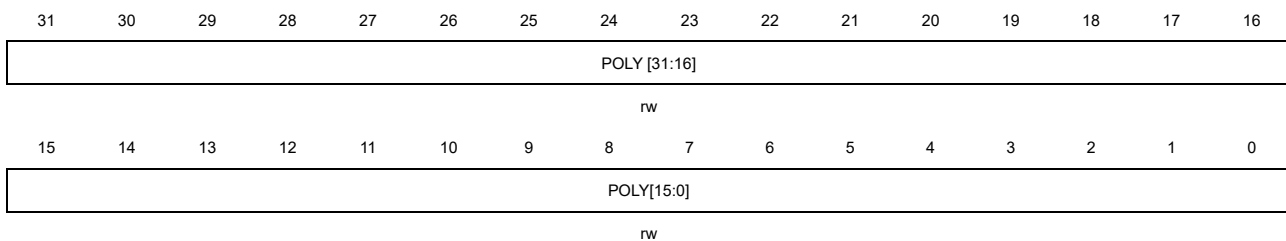
位/位域	名称	描述
31:0	IDATA[31:0]	配置 CRC 初值 CRC_CTL 寄存器的 RST 位置位后，CRC_DATA 寄存器的值将被更新为此寄存器的值。

10.4.5. 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。

11. 三角函数加速器（TMU）

11.1. 简介

三角函数加速器（TMU）是一个完全可配置的单元，可执行常见的三角运算和算术运算操作。TMU 可以减轻 CPU 的负担，通常应用于电机控制，信号处理和很多其他应用场景。

TMU 可以计算 5 种函数，输入和输出数据支持 q1.31、q1.15 定点格式和 IEEE754 32 位单精度浮点格式。

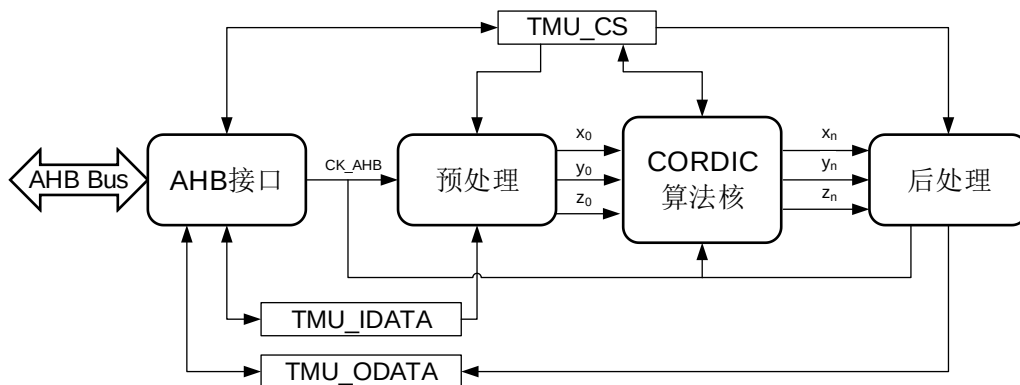
11.2. 主要特征

- 5 种函数；
- 中断和 DMA 请求；
- 可配置的数据格式：q1.31、q1.15 定点格式和 IEEE754 32 位单精度浮点格式；
- 可编程的计算精度；
- CORDIC 算法核：支持两种圆周系统和双曲线系统，支持旋转模式和向量模式。
- 作为 CPU 的一个外部协处理器

11.3. 结构框图

图 11-1. TMU 模块结构框图 TMU 模块内部结构细节。

图 11-1. TMU 模块结构框图



预处理模块将输入数据寄存器（TMU_IDATA）中的数据进行转换，得到 CORDIC 算法核需要的初始数据 (x_0, y_0, z_0) 。输入数据寄存器中的数据格式为 q1.31、q1.15 定点格式或 IEEE754 32 位单精度浮点格式。

CORDIC 算法核心模块根据初始数据 (x_0, y_0, z_0) ，经过迭代和运算，得到 (x_n, y_n, z_n) 。TMU 算法核心模块支持圆周系统和双曲线系统，每种系统支持旋转模式和向量模式。

后处理模块对 (x_n, y_n, z_n) 进行数据转换和缩放等处理，并将处理后的数据写入输出数据寄存器（TMU_ODATA）。输出数据寄存器中的数据格式为 q1.31、q1.15 定点格式或 IEEE754 32 位单精度浮点格式。

11.4. 功能描述

11.4.1. 数据格式和配置

当 IFLTEN / OFLTEN 位复位时，TMU 模块的输入/输出数据是定点有符号整型格式（q1.31、q1.15 格式）。当 IFLTEN / OFLTEN 位置位时，TMU 模块的输入/输出数据是 IEEE754 32 位单精度浮点格式。

q1.31 格式中，第 31 位是符号位，0~30 位是小数位，表达数值范围是 $[-1, 1-2^{-31}]$ ，对应 [0x80000000, 0x7FFFFFFF]。

q1.15 格式中，第 15 位是符号位，0~14 位是小数位，表达数值范围是 $[-1, 1-2^{-15}]$ ，对应 [0x8000, 0x7FFF]。

IEEE754 32 位单精度浮点格式的数据范围如[表 11-3. IEEE754 32 位单精度浮点格式](#)所示。

当输入数据格式为定点时（IFLTEN=0），可以通过 TMU_CS 寄存器中的 IWIDTH 位配置输入数据的定点格式。当输入数据格式为浮点时（IFLTEN=1），IWIDTH 位的配置无效。每个模式需要的输入数据的个数有所不同（例如，模式 0 需要两个输入数据，模式 5 只需要一个），可以通过 TMU_CS 寄存器的 INUM 位配置输入数据的数量。详细配置参考[表 11-1. 输入数据配置](#)。

注意：当输入数据配置为 q1.15 格式，如果所配模式需要两个输入数据，只需要写一次 TMU_IDATA 寄存器，第一个输入数据在低半字，第二个输入数据在高半字。如果所配模式只需要一个输入数据，则只使用低半字，高半字的不使用。

表 11-1. 输入数据配置

IWIDTH 位	INUM 位	IFLTEN 位	数据格式	写 TMU_IDATA 寄存器
0	0	0	q1.31 定点	写一次
0	1	0	q1.31 定点	连续写两次
1	0	0	q1.15 定点	写一次
1	1	0	q1.15 定点	不可用
X	0	1	IEEE754 32 位单精度浮点	写一次
X	1	1		连续写两次

当输出数据格式为定点时（OFLTEN=0），可以通过 TMU_CS 寄存器的 OWIDTH 位配置输出数据的定点格式。当输出数据格式为浮点时（OFLTEN=1），OWIDTH 位的配置无效。每个模式的输出数据的个数有所不同（例如，模式 0 有两个输出数据，模式 8 只有一个），可以通过 TMU_CS 寄存器的 ONUM 位配置输出数据的数量。详细配置参考[表 11-2. 输出数据配置](#)。

注意：当输出数据配置为 q1.15 格式，如果所配模式有两个输出数据，只需要读一次 TMU_ODATA 寄存器。第一个输出数据在低半字，第二个输出数据在高半字。如果所配模式只

有一个输出，则只使用低半字，不使用高半字。

表 11-2. 输出数据配置

OWIDTH 位	ONUM 位	OFLTEN 位	数据格式	读 TMU_ODATA 寄存器
0	0	0	q1.31 定点	读一次
0	1	0	q1.31 定点	连续读两次
1	0	0	q1.15 定点	读一次
1	1	0	q1.15 定点	不可用
X	0	1	IEEE754 32 位	读一次
X	1	1	单精度浮点	连续读两次

11.4.2. 浮点数据格式

表 11-3. IEEE754 32 位单精度浮点格式

S [31]	E [30:23]	M [22:0]	值 (V)
0	0	0	零 (V = 0)
1	0	0	负零 (V = -0)
0 + ve 1 - ve	0	非零	非规格化 ($V = (-1)^S \cdot 2^{(-126)} \cdot (0.M)$)
0 + ve 1 - ve	1 to 254	0 to 0x7FFFFFFF	正常范围 ($V = (-1)^S \cdot 2^{(E-127)} \cdot (1.M)$)
0	254	0x7FFFFFFF	正最大值 (V = +Max)
1	254	0x7FFFFFFF	负最大值 (V = -Max)
0	最大值 255	0	正无穷 (V = +Infinity)
1	最大值 255	0	负无穷 (V = -Infinity)
x	最大值 255	非零	非数 (V = NaN)

TMU 对于 IEEE 浮点数据格式中不同的数据处理如下：

负零：如果运算结果是 0，TMU 将会输出正零 (S=0, E=0, M=0)，不会产生负零。所有的 TMU 运算都将负零当做零处理。

非规格化数：输入的非规格化操作数 (E=0, M!=0) 被视为 0 (E=0, M=0)。

非数 (NaN)：输入的 NaN 操作数 (E=max, M!=0) 被视为无穷 (E=max, M=0)。

溢出：当 IFLTEN 和 OFLTEN 位同时被置位时，如果输入数据或者运算中的数据按照既定浮点格式显得过大会产生上溢，在这种情景下，TMU_CS 寄存器的 OVRF 标志位置位。

11.4.3. 模式配置

TMU_CS 寄存器的 MODE[3:0]位域用来配置 CORDIC 算法核模块的运行模式。不同的模式使用不同的系统（圆周系统或者双曲线系统）和不同的模式（旋转模式或者向量模式）。详细信息参考 [表 11-4. TMU 模式配置](#)。

表 11-4. TMU 模式配置

模式	第一个输入数据	第二个输入数据	第一个输出数据	第二个输出数据	使用的系统和模式
模式 0	θ	m	$m \cdot \cos(\theta)$	$m \cdot \sin(\theta)$	圆周系统, 旋转模式
模式 1	θ	m	$m \cdot \sin(\theta)$	$m \cdot \cos(\theta)$	圆周系统, 旋转模式
模式 2	x	y	$\text{atan2}(y, x)$	$\sqrt{x^2 + y^2}$	圆周系统, 向量模式
模式 3	x	y	$\sqrt{x^2 + y^2}$	$\text{atan2}(y, x)$	圆周系统, 向量模式
模式 4	x	无	$\sqrt{x}$	无	双曲线系统, 向量模式

模式 0: $m \cdot \cos(\theta)$

该模式用来计算余弦函数, 有两个输入数据和两个输出数据。

当 IFLTEN 位和 OFLTEN 位均置位时, 详细信息参考[表 11-5. 模式 0 描述, IFLTEN 位和 OFLTEN 位均置位时](#)。

表 11-5. 模式 0 描述, IFLTEN 位和 OFLTEN 位均置位时

参数	范围	描述
第一个输入数据	$\frac{\theta}{\pi} \in (-2^{24}, 2^{24})$	角度值 θ 单位是弧度 (rad), 超出范围时溢出。
第二个输入数据	$m \in (-\infty, +\infty)$	超出浮点表示范围时溢出
第一个输出数据	$m \cdot \cos(\theta) \in (-\infty, +\infty)$	--
第二个输出数据	$m \cdot \sin(\theta) \in (-\infty, +\infty)$	--
缩放因子 FACTOR[2:0]	不可用	保持复位值 3'b000

当 IFLTEN 位或 OFLTEN 位复位时, 详细信息参考[表 11-6. 模式 0 描述, IFLTEN 位或 OFLTEN 位复位时](#)。

表 11-6. 模式 0 描述, IFLTEN 位或 OFLTEN 位复位时

参数	范围	描述
第一个输入数据	$\frac{\theta}{\pi} \in [-1, 1)$	角度值 θ 单位是弧度 (rad), 范围 $\theta \in [-\pi, \pi)$ 。软件用 θ 除以 π 后, 转换为 $[-1, 1)$ 范围内, 再按照 q1.31、q1.15 或者 float 格式写入 TMU_IDATA 寄存器
第二个输入数据	$m \in [0, 1)$	当 $0 \leq m < 1$ 时, 直接按照 q1.31、q1.15 或者 float 格式写入 TMU_IDATA 寄存器。当 $m \geq 1$ 时, 需要先通过软件缩小 m 到 $[0, 1)$ 范围内, 再写入到寄存器中。
第一个输出数据	$m \cdot \cos(\theta)$, 范围 $[-1, 1)$	如果之前软件缩小过 m , 需要对该输出数据进行相应比例的放大, 以获得真实结果。
第二个输出数据	$m \cdot \sin(\theta)$, 范围 $[-1, 1)$	
缩放因子 FACTOR[2:0]	不可用	保持复位值 3'b000

注意: 当模长 $m > 1$ 时, 软件缩放的比例是自行选择的。

例如，计算 $100 * \cos\left(\frac{\pi}{2}\right)$ ，当 IFLTEN 位和 OFLTEN 位均复位，IWIDTH 位和 OWIDTH 位均置位时，输入输出数据均为 q1.15 格式。可以按照以下步骤进行：

1. 软件处理角度 $\frac{\pi}{2}$ 。 $\frac{\frac{\pi}{2}}{\pi} = \frac{1}{2}$ ，q1.15 格式下数据为 0x4000。
2. 软件处理模长 m ，将其缩小 128 倍。 $\frac{100}{128} = 0.78125$ ，q1.15 格式下数据为 0x6400。
3. 往寄存器 TMU_IDATA 写第一个输入数据：角度值 0x4000。
4. 往寄存器 TMU_IDATA 写第二个输入数据：模长 0x6400。
5. 等待 ENDF 标志置 1，读 TMU_ODATA 获取第一个输出数据 $y_1 = \frac{100}{128} * \cos\left(\frac{\pi}{2}\right)$ ，再读一次 TMU_ODATA 获取第二个输出数据 $y_2 = \frac{100}{128} * \sin\left(\frac{\pi}{2}\right)$ 。
6. 结果处理。由于之前对模长 m 缩小了 128 倍，结果需要再乘以 128，则 $100 * \cos\left(\frac{\pi}{2}\right) = 128 * y_1$ 。

本例（计算 $100 * \cos\left(\frac{\pi}{2}\right)$ ）中对模长 m 和结果处理使用了 128 倍缩放，当然也可以使用其他缩放倍数，比如 101。

模式 1: $m * \sin(\theta)$

该模式用来计算正弦函数，有两个输入数据和两个输出数据。

当 IFLTEN 位和 OFLTEN 位均置位时，详细信息参考 [表](#)

11-7. 模式 1 描述, IFLTEN 位和 OFLTEN 位均置位时。

表

11-7. 模式 1 描述, IFLTEN 位和 OFLTEN 位均置位时

参数	范围	描述
第一个输入数据	$\frac{\theta}{\pi} \in (-2^{24}, 2^{24})$	角度值 θ 单位是弧度 (rad)，超出范围时溢出。
第二个输入数据	$m \in (-\infty, +\infty)$	超出浮点表示范围时溢出
第一个输出数据	$m * \sin(\theta) \in (-\infty, +\infty)$	--
第二个输出数据	$m * \cos(\theta) \in (-\infty, +\infty)$	--
缩放因子 FACTOR[2:0]	不可用	保持复位值 3'b000

当 IFLTEN 位或 OFLTEN 位复位时，详细信息参考 [表 11-8. 模式 1 描述, IFLTEN 位或 OFLTEN 位复位时。](#)

表 11-8. 模式 1 描述, IFLTEN 位或 OFLTEN 位复位时

参数	范围	描述
第一个输入数据	$\frac{\theta}{\pi} \in [-1, 1)$	角度值 θ 单位是弧度 (rad)，范围 $\theta \in [-\pi, \pi)$ 。软件用 θ 除以 π 后，转换为 $[-1, 1)$ 范围内，再按照 q1.31、q1.15 或者 float 格式写入 TMU_IDATA 寄存器

参数	范围	描述
第二个输入数据	$m \in [0,1)$	当 $0 \leq m < 1$ 时，直接按照 q1.31、q1.15 或者 float 格式写入 TMU_IDATA 寄存器。当 $m \geq 1$ 时，需要先通过软件缩小 m 到 $[0,1)$ 范围内，再写入到寄存器中。
第一个输出数据	$m * \sin(\theta)$, 范围 $[-1,1)$	如果之前软件缩小过 m ，需要对该输出数据进行相应比例的放大，以获得真实结果。
第二个输出数据	$m * \cos(\theta)$, 范围 $[-1,1)$	
缩放因子 FACTOR[2:0]	不可用	保持复位值 3'b000

注意：当模长 $m > 1$ 时，软件缩放的比例是自行选择的。

例如，计算 $100 * \sin\left(\frac{\pi}{2}\right)$ ，当 IFLTEN 位和 OFLTEN 位均复位，IWIDTH 位和 OWIDTH 位均置位时，输入输出数据均为 q1.15 格式。可以按照以下步骤进行：

1. 软件处理角度 $\frac{\pi}{2}$ 。 $\frac{\pi}{2} = \frac{1}{2}$ ，q1.15 格式下数据为 0x4000。
2. 软件处理模长 m ，将其缩小 128 倍。 $\frac{100}{128} = 0.78125$ ，q1.15 格式下数据为 0x6400。
3. 往寄存器 TMU_IDATA 写第一个输入数据：角度值 0x4000。
4. 往寄存器 TMU_IDATA 写第二个输入数据：模长 0x6400。
5. 等待 ENDF 标志置 1，读 TMU_ODATA 获取第一个输出数据 $y_1 = \frac{100}{128} * \sin\left(\frac{\pi}{2}\right)$ ，再读一次 TMU_ODATA 获取第二个输出数据 $y_2 = \frac{100}{128} * \cos\left(\frac{\pi}{2}\right)$ 。
6. 结果处理。由于之前对模长 m 缩小了 128 倍，结果需要再乘以 128，则 $100 * \sin\left(\frac{\pi}{2}\right) = 128 * y_1$ 。

本例（计算 $100 * \sin\left(\frac{\pi}{2}\right)$ ）中对模长 m 和结果处理使用了 128 倍缩放，当然也可以使用其他缩放倍数，比如 101。

模式 2: phase= atan2 (y,x)

该模式用来计算 atan2(y,x) 函数，有两个输入数据和两个输出数据。

当 IFLTEN 位和 OFLTEN 位均置位时，详细信息参考 [表 11-9. 模式 2 描述, IFLTEN 位和 OFLTEN 位均置位时](#)。

表 11-9. 模式 2 描述, IFLTEN 位和 OFLTEN 位均置位时

参数	范围	描述
第一个输入数据	$2^{-24} < y/x < 2^{24}$	超出范围时溢出。
第二个输入数据		
第一个输出数据	角度 $\theta \in [-1,1)$	坐标位置对应的角度， $[-1,1)$ 对应 $[-\pi, \pi)$ 。该输出数据乘以 π 得到真实角度值。
第二个输出数据	模长 $m \in [0, +\infty)$	$m = \sqrt{x^2 + y^2}$ ，超出浮点表示范围时溢出

参数	范围	描述
缩放因子 FACTOR[2:0]	不可用	保持复位值 3'b000

当 IFLTEN 位或 OFLTEN 位复位时, 详细信息参考 [表 11-10. 模式 2 描述, IFLTEN 位或 OFLTEN 位复位时](#)。

表 11-10. 模式 2 描述, IFLTEN 位或 OFLTEN 位复位时

参数	范围	描述
第一个输入数据	$x \in [-1, 1)$	笛卡尔坐标系中横坐标值。如果 $x \geq 1$ 或者 $x < -1$, 则需要进行软件缩放。
第二个输入数据	$y \in [-1, 1)$	笛卡尔坐标系中纵坐标。如果 $y \geq 1$ 或者 $y < -1$, 则需要进行软件缩放。
第一个输出数据	角度 $\theta \in [-1, 1)$	坐标位置对应的角度, $[-1, 1)$ 对应 $[-\pi, \pi)$ 。该输出数据乘以 π 得到真实角度值。
第二个输出数据	模长 $m \in [0, 1)$	$m = \sqrt{x^2 + y^2}$ 。如果之前对 x 和 y 进行了缩放, 该模长需要进行等比例放大。
缩放因子 FACTOR[2:0]	不可用	保持复位值 3'b000

注意: 当 IFLTEN 位或 OFLTEN 位复位时, 如果 x 或 y 超出 $[-1, 1)$ 的范围, 或者 $\sqrt{x^2 + y^2} \geq 1$, 需要同时对 x 和 y 进行同比例缩放, 避免出现超出数据范围的情况。不能只缩放一个, 这样可以保证缩放前后坐标对应的角度不变。

例如, 计算 $\theta = \text{atan2}(5, 80)$, 当 IFLTEN 位和 OFLTEN 位均复位, IWIDTH 位和 OWIDTH 位均置位时, 输入输出数据均为 q1.15 格式。可以按照以下步骤进行:

1. 软件缩放。输入数据 (5, 80) 除以 128, 结果为 (0.0390625, 0.625), q1.15 格式下数据为 (0x0500, 0x5000)。
2. 往寄存器 TMU_IDATA 写入第一个输入数据 0x0500。
3. 往寄存器 TMU_IDATA 写入第二个输入数据 0x5000。
4. 等待 ENDF 标志置 1, 读 TMU_ODATA 获取第一个输出数据 θ , 再读一次 TMU_ODATA 获取第二个输出数据 m 。
5. 结果处理。第一个输出数据角度 θ 乘以 π , 得到真实弧度。由于之前输入数据缩小了 128 倍, 读出的第二个输出数据模长 m 需要再乘以 128 才是真实模长。

本例 (计算 $\theta = \text{atan2}(5, 80)$) 中对输入和模长使用了 128 倍缩放, 当然也可以使用其他缩放倍数, 比如 81。

模式 3: $\text{modulus} = \sqrt{x^2 + y^2}$

该模式用来计算 $\sqrt{x^2 + y^2}$ 函数, 有两个输入数据和两个输出数据。

当 IFLTEN 位和 OFLTEN 位均置位时, 详细信息参考 [表 11-11. 模式 3 描述, IFLTEN 位和 OFLTEN 位均置位时](#)。

表 11-11. 模式 3 描述, IFLTEN 位和 OFLTEN 位均置位时

参数	范围	描述
第一个输入数据	$2^{-24} < y/x < 2^{24}$	超出范围时溢出
第二个输入数据		
第一个输出数据	模长 $m \in [0, +\infty)$	角度值 θ 单位是弧度 (rad)
第二个输出数据	角度 $\theta \in [-1, 1)$	坐标位置对应的角度, $[-1, 1)$ 对应 $[-\pi, \pi)$ 。该输出数据乘以 π 得到真实角度值。
缩放因子 FACTOR[2:0]	不可用	保持复位值 3'b000

当 IFLTEN 位或 OFLTEN 位复位时, 详细信息参考 [表 11-12. 模式 3 描述, IFLTEN 位或 OFLTEN 位复位时](#)。

表 11-12. 模式 3 描述, IFLTEN 位或 OFLTEN 位复位时

参数	范围	描述
第一个输入数据	$x \in [-1, 1)$	笛卡尔坐标系中横坐标值。如果 $x \geq 1$ 或者 $x \leq -1$, 则需要进行软件缩放。
第二个输入数据	$y \in [-1, 1)$	笛卡尔坐标系中纵坐标。如果 $y \geq 1$ 或者 $y \leq -1$, 则需要进行软件缩放。
第一个输出数据	模长 $m \in [0, 1)$	模长, $m = \sqrt{x^2 + y^2}$ 。如果之前对 x 和 y 进行了缩放, 该模长需要进行等比例放大。
第二个输出数据	角度 $\theta \in [-1, 1)$	坐标位置对应的角度, $[-1, 1)$ 对应 $[-\pi, \pi)$ 。该输出数据乘以 π 得到真实角度值。
缩放因子 FACTOR[2:0]	不可用	保持复位值 3'b000

注意: 当 IFLTEN 位或 OFLTEN 位复位时, 如果 x 或 y 超出 $[-1, 1)$ 的范围, 或者 $\sqrt{x^2 + y^2} \geq 1$, 需要同时对 x 和 y 进行同比例缩放, 避免出现超出数据范围的情况。不能只缩放一个, 这样可以保证缩放前后坐标对应的角度不变。

例如, 计算 $\sqrt{5^2 + 80^2}$, 当 IFLTEN 位和 OFLTEN 位均复位, IWIDTH 位和 OWIDTH 位均置位时, 输入输出数据均为 q1.15 格式。可以按照以下步骤进行:

1. 软件缩放。输入数据 (5, 80) 除以 128, 结果为 (0.0390625, 0.625), q1.15 格式下数据为 (0x0500, 0x5000)。
2. 往寄存器 TMU_IDATA 写入第一个输入数据 0x0500。
3. 往寄存器 TMU_IDATA 写入第二个输入数据 0x5000, TMU 启动计算。
4. 等待 ENDF 标志置 1, 读 TMU_ODATA 获取第一个输出数据 m , 再读一次 TMU_ODATA 获取第二个输出数据 θ 。
5. 软件结果处理。由于之前输入数据缩小了 128 倍, 读出的第一个输出数据模长 m 需要再乘以 128 才是真实模长。第二个输出数据角度 θ 乘以 π , 得到真实弧度。

本例 (计算 $\sqrt{5^2 + 80^2}$) 中对输入和模长使用了 128 倍缩放, 当然也可以使用其他缩放倍数, 比如 81。

模式 4: $\sqrt{x}$

该模式用来计算 $\sqrt{x}$ 函数，有一个输入数据和一个输出数据。

当 IFLTEN 位和 OFLTEN 位均置位时，详细信息参考[表 11-13. 模式 4 描述, IFLTEN 位和 OFLTEN 位均置位时](#)。

表 11-13. 模式 4 描述, IFLTEN 位和 OFLTEN 位均置位时

参数	范围	描述
输入数据	$x \in [0, +\infty)$	超出范围时溢出
输出数据	$\sqrt{x} \in [0, +\infty)$	--
缩放因子 FACTOR[2:0]	不可用	保持复位值 3'b000

当 IFLTEN 位或 OFLTEN 位复位时，详细信息参考[表 11-14. 模式 4 描述, IFLTEN 位或 OFLTEN 位复位时](#)。

表 11-14. 模式 4 描述, IFLTEN 位或 OFLTEN 位复位时

参数	范围	描述
输入数据	$\frac{x}{2^f} \in [0.027, 0.875]$	$x \in [0.027, 2.34]$ 。软件进行缩放处理，保证 $\frac{x}{2^f} < (1 - \frac{1}{2^{f+2}})$ ，其中 f 为缩放因子。然后将 $\frac{x}{2^f}$ 以 q1.15 或者 q1.31 写入 TMU_IDATA 寄存器。
输出数据	$\frac{\sqrt{x}}{2^f} \in [0.164, 0.866]$	输出数据乘以 2^f 得到 $\sqrt{x}$ 。
缩放因子 FACTOR[2:0]	$f \in [0, 2]$	FACTOR[2:0]配置为 f

注意：当 IFLTEN 位或 OFLTEN 位复位时，缩放因子 FACTOR[2:0]需要配置。

例如，计算 $\sqrt{2}$ ，当 IFLTEN 位和 OFLTEN 位均复位，IWIDTH 位和 OWIDTH 位均置位时，输入输出数据均为 q1.15 格式。可以按照以下步骤进行：

1. 软件缩放。输入数据 2 除以 4(缩放因子 $f=3'b010$)，结果为 0.5。q1.15 格式下数据为 0x4000。
2. 将缩放因子 $f=3'b010$ 写入 TMU_CS 寄存器的 FACTOR[2:0]位域。
3. 往寄存器 TMU_IDATA 写入输入数据 0x4000，TMU 开始计算。
4. 等待 ENDF 标志置 1，读 TMU_ODATA 获取输出数据 $y_1 = \frac{\sqrt{2}}{2^2}$ 。
5. 结果处理。输出数据乘以 2^2 ，得到 $\sqrt{2}$ 。

为保证计算精度，对于不同输入推荐使用[表 11-15. 模式 4 推荐的缩放因子](#)中的缩放因子。

表 11-15. 模式 4 推荐的缩放因子

输入x范围	缩放因子 FACTOR[2:0]	输入数据范围
$0.027 < x < 0.75$	3'b000	[0.027, 0.75)
$0.75 \leq x < 1.75$	3'b001	[0.375, 0.875)
$1.75 \leq x < 2.341$	3'b010	[0.4375, 0.585)

11.4.4. TMU 运算挂起

如果当前正在执行 TMU 运算，可以写 TMU_CS 和 TMU_IDATA 寄存器，写入的内容将被挂起。当 TMU 运算完成（结果被读取，ENDF 标志清零）时，如果挂起的输入数据数量符合配置（定义在被挂起的 TMU_CS 中），TMU 模块将按照挂起的配置和数据开始新一次 TMU 运算。

例如，如果配置的 TMU 模式需要两个 32-bit 输入数据(IWIDTH=0,INUM=1)，当往 TMU_IDATA 寄存器写入两个 32-bit 的数据后，TMU 启动一次运算。如果第二个输入参数在下次 TMU 运算中不改变，此时可以修改 INUM=0。当前一次 TMU 运算结束后，往 TMU_IDATA 寄存器写入一个输入数据，TMU 启动运算，只要 TMU 模式没有改变，第二个参数仍使用之前的数值。

注意：复位后，第二个数值为+1（0x7FFFFFFF）。

如果当前已经存在挂起的 TMU 数据，再往 TMU_CS 和 TMU_IDATA 寄存器写入新的数据，则新数据覆盖原来数据，新的 TMU 数据被挂起，原数据的挂起失效。

11.4.5. 零开销

当一个 TMU 运算开始后，可以直接读取输出数据寄存器，在结果返回之前总线会自动插入等待周期。可以按照以下步骤进行：

1. 根据需要 TMU_CS 寄存器。
2. 往 TMU_IDATA 写入需要的参数，启动一个 TMU 运算。
3. 根据需要配置下次 TMU 模式，并往 TMU_IDATA 写入下次需要的数据。
4. 读取 TMU_ODATA。总线自动插入等待周期。当读取 TMU_ODATA 操作完成后，在第 3 步骤中配置得 TMU 操作会自动启动。
5. 返回第 3 步。

11.4.6. TMU 指令集

TMU 模块作为 M33 内核的协处理器 0，其支持的指令集如。

表 11-16. TMU 指令集

TMU 寄存器	TMU 编号(CRm)	可用的指令	OPCODE	寄存器访问属性
TMU_CS	0	MCR/MRC	0	rw
TMU_IDATA	1	MCR/MCRR/ MRC/MRRC	0	rw
TMU_ODATA	2	MRC/MRRC	0	r
TMU_IDATA	9	MCR/MCRR	Mode0~4	rw

11.4.7. 中断和 DMA

当 ENDF 标志位置 1 时，如果 TMU_CS 寄存器中的 RIE 为 1，则产生中断请求。ENDF 标志清 0 后，中断请求也清除。

当 OVRF 标志设置为 1 时，如果 TMU_CS 寄存器中的 OVRIE 位为 1，则产生中断请求。当 OVRF 标志被清 0 后，中断请求也被清除。

如果 TMU_CS 寄存器中的 WDEN 为 1 并且此时没有 TMU 挂起，则产生 DMA 请求，DMA 请求的数量却决于 TMU_CS 寄存器中的 INUM 位。如果 TMU_CS 寄存器中的 INUM=0，产生一次 DMA 传输请求。如果 TMU_CS 寄存器中的 INUM=1，产生两次 DMA 传输请求。

当 ENDF 标志位置 1 时，如果 TMU_CS 寄存器中的 RDEN 为 1，则产生 DMA 请求，DMA 请求的数量 TMU_CS 寄存器中的 ONUM 位。如果 TMU_CS 寄存器中的 ONUM=0，产生一次 DMA 传输请求。如果 TMU_CS 寄存器中的 ONUM=1，产生两次 DMA 传输请求。

11.5. TMU 寄存器

TMU 基地址：0x4802 4400

11.5.1. 控制和状态寄存器（TMU_CS）

地址偏移：0x00

复位值：0x0000 0050

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ENDF	OVRF	OVRIE	保留						IWIDTH	OWIDTH	INUM	ONUM	WDEN	RDEN	RIE
r	rc_w1	rw							rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IFLTEN	OFLTEN	保留				FACTOR[2:0]				保留			MODE[3:0]		
rw	rw					rw							rw		

位/位域	名称	描述
31	ENDF	TMU运算结束标志 0: TMU当前无运算或者正在进行运算 1: TMU运算结束，结果已经写入TMU_ODATA寄存器 当TMU运算结束并且结果已经写入TMU_ODATA寄存器时，该位硬件置1。 读TMU_ODATA寄存器（ONUM+1）次，该位硬件清0。 注意： 当该位为1时，新的TMU运算不会启动。
30	OVRF	溢出标志 该位由硬件置 1，软件写 1 清 0。 0: 浮点数据无溢出 1: 浮点数据溢出
29	OVRIE	溢出中断使能 0: 溢出中断禁能 1: 溢出中断使能
28:23	保留	必须保持复位值。

22	IWIDTH	输入数据位宽 0: 32-bit 1: 16-bit 该位决定了输入数据的定点格式。 如果配置为32-bit, 则写入TMU_IDATA寄存器的数据为q1.31定点格式。 如果配置为16-bit, 则写入TMU_IDATA寄存器的数据为q1.15定点格式。第一个数据写入TMU_IDATA的低半字, 第二个数据写入TMU_IDATA的高半字。只需要一个输入数据 (INUM=0) 时, TMU_IDATA的高半字不使用。 注意: 该位域仅在输入数据为定点格式 (IFLTEN=0) 时生效。
21	OWIDTH	输出数据位宽 0: 32-bit 1: 16-bit 该位决定了输出数据的定点格式。 如果配置为32-bit, 则TMU_ODATA寄存器包含的输出数据为q1.31定点格式。 如果配置为16-bit, 则TMU_ODATA寄存器包含的输出数据为q1.15定点格式。第一个输出数据在TMU_ODATA的低半字, 第二个输出数据在TMU_IDATA的高半字。 注意: 该位域仅在输出数据为定点格式 (OFLTEN=0) 时生效。
20	INUM	写TMU_IDATA寄存器的次数 0: 一次 32-bit 写操作。写入 TMU_IDATA 寄存器一次, 启动一次 TMU 运算。 1: 两次 32-bit 写操作。连续写入 TMU_IDATA 寄存器两次, 启动一次 TMU 运算。 注意: 当输入数据格式为q1.15时 (IWIDTH=1, IFLTEN=0), 只需要一个32-bit写操作。
19	ONUM	读TMU_ODATA寄存器的次数 0: 一次 32-bit 读操作。当 TMU 运算结束, 只有一个 32-bit 运算结果传输进 TMU_ODATA 寄存器。读一次 TMU_ODATA 寄存器将清除 ENDF 标志。 1: 两次 32-bit 读操作。当 TMU 运算结束, 有两个 32-bit 运算结果传输进 TMU_ODATA 寄存器。读两次 TMU_ODATA 寄存器将清除 ENDF 标志。 注意: 当输出数据格式为q1.15时 (OWIDTH=1, OFLTEN=0), 只需要一个32-bit读操作。
18	WDEN	DMA 写请求使能 0: 禁能 1: 使能。当无 TMU 运算挂起时, 产生 DMA 写请求。
17	RDEN	DMA 读请求使能 0: 禁能 1: 使能。当ENDF置1时, 产生DMA读请求。
16	RIE	读中断使能 0: 禁能 1: 使能。当ENDF置1时, 产生读中断请求。
15	IFLTEN	输入数据浮点格式使能 0: 输入数据为定点有符号整形格式 q1.15 或 q1.31

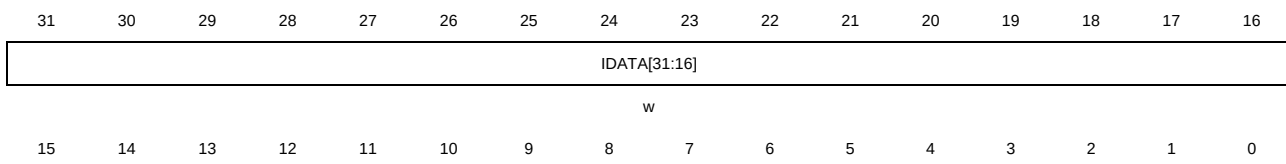
		1: 输入数据为 32 位单精度浮点格式
14	OFLTEN	输出数据浮点格式使能 0: 输出数据为定点有符号整形格式 q1.15 或 q1.31 1: 输出数据为 32 位单精度浮点格式
13:11	保留	必须保持复位值。
10:8	FACTOR[2:0]	缩放因子 该位域定义了缩放因子: $2^{\text{FACTOR}[2:0]}$ 。 000: 2^0 001: 2^1 010: 2^2 ... 110: 2^6 111: 2^7 当实际输入参数超过规定的输入数据范围[-1,1), 实际输入参数需要除以 $2^{\text{FACTOR}[2:0]}$, 并且输出数据需要乘以 $2^{\text{FACTOR}[2:0]}$ 以得到实际输出结果, 细节如下: $\text{TMU\_IDATA} = \text{实际输入参数} / 2^{\text{FACTOR}[2:0]}$ 实际输出结果 = $\text{TMU\_ODATA} * 2^{\text{FACTOR}[2:0]}$ 注意: 对于模式 0、模式 1、模式 2 和模式 3, 建议该位域配置为 3'b000。对模式 4, 该位域针对不同输入参数推荐了一些配置。
7:4	保留	必须保持复位值。
3:0	MODE[3:0]	TMU模式 0000: 模式0, $m * \cos(\theta)$ 0001: 模式1, $m * \sin(\theta)$ 0010: 模式2, $\text{phase} = \text{atan2}(y, x)$ 0011: 模式3, $\text{modulus} = \sqrt{x^2 + y^2}$ 0100: 模式9, $\sqrt{x}$ 0101~1111: 保留 注意: x、 θ : 第一个输入数据 y、m: 第二个输入数据

11.5.2. 输入数据寄存器 (TMU_IDATA)

地址偏移: 0x04

复位值: 0xFFFF XXXX

该寄存器只能按字 (32 位) 访问。



IDATA[15:0]

w

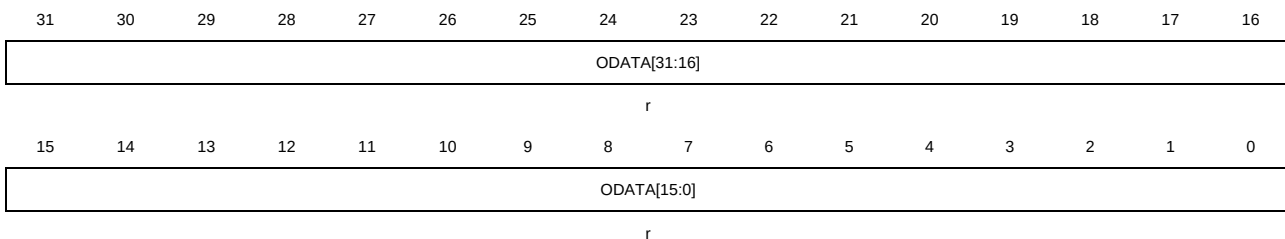
位/位域	名称	描述
31:0	IDATA[31:0]	输入数据 输入数据写入该寄存器。细节参考表 11-1. 输入数据配置。 注意： 1. 当无TMU运算正在进行并且需要的输入参数已经写入该寄存器，将启动一次新的TMU运算。 2. 当TMU正在进行一次运算时，再写入的数据将会被挂起，直到当前的TMU运算结束并且输出数据被读取。在数据挂起期间，如果写入新的数据，则新数据覆盖之前被挂起的数据。

11.5.3. 输出数据寄存器（TMU_ODATA）

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	ODATA[31:0]	输出数据 当TMU运算结束，结果传输进该寄存器。细节参考表11-2. 输出数据配置。 注意： 1. 当ENDF位置1时，读取该寄存器可以获得TMU运算结果。 2. 当符合配置的读操作完成，ENDF位被硬件自动清0。

12. 直接存储器访问控制器（DMA）

12.1. 简介

DMA 控制器提供了一种硬件的方式在外设和存储器之间或者存储器和存储器之间传输数据，而无需 CPU 的介入，从而释放带宽，使 CPU 可以专注在处理其他系统功能上。数据可以通过 DMA 在外设和内存以及内存与内存之间快速移动，而无需任何 CPU 操作。DMA 控制器有 12 个通道（DMA0 有 6 个通道，DMA1 有 6 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。

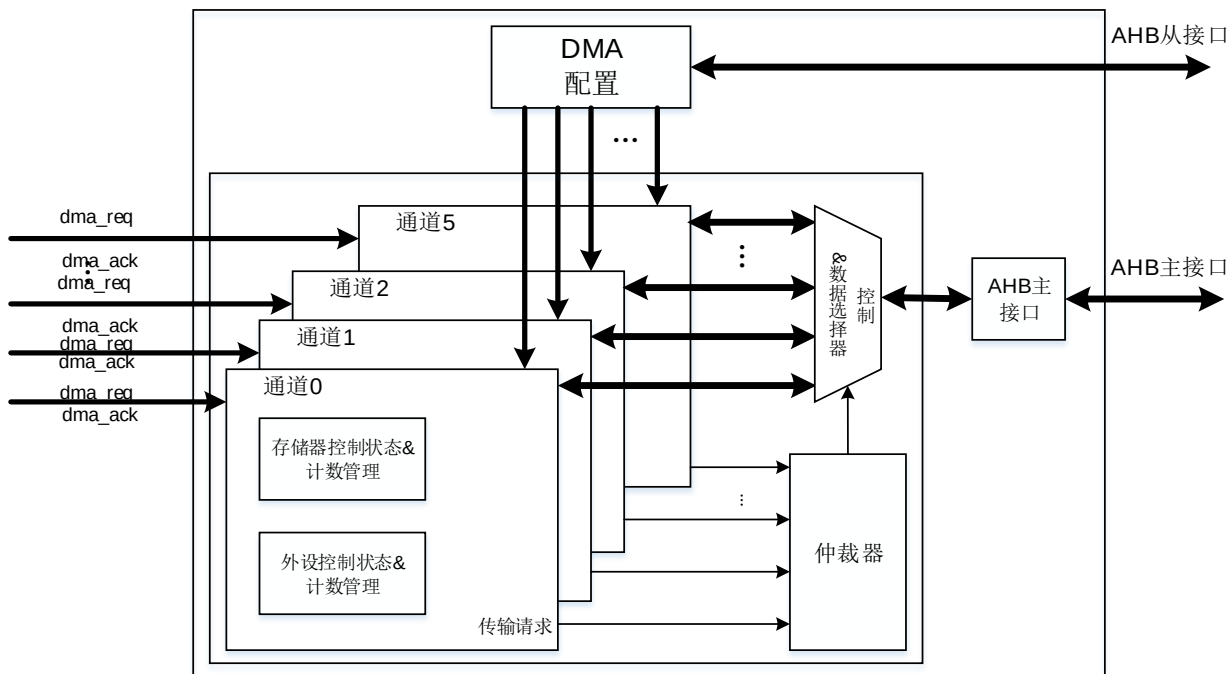
DMA 控制器和 Cortex®-M33 内核共享系统总线。当 DMA 和 CPU 访问同样的地址空间时，DMA 访问可能会阻挡 CPU 访问系统总线几个总线周期。总线矩阵中实现了循环仲裁算法来分配 DMA 与 CPU 的访问权，它可以确保 CPU 得到至少一半的系统总线带宽。

12.2. 主要特征

- 传输数据长度可编程配置，最大到 65535；
- 12 个通道，并且每个通道都可配置（DMA0 有 6 个通道，DMA1 有 6 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；
- 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。

12.3. 结构框图

图 12-1. DMA 结构框图



由图 12-1. DMA 结构框图所示，DMA 控制器由 4 部分组成：

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

12.4. 功能说明

12.4.1. 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 12-1. DMA 传输操作。

表 12-1. DMA 传输操作

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

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

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

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

- 若 CHEN 位被清零时 DMA 传输还未完成，重新使能 CHEN 位将分两种情况：
 - 在重新使能 DMA 通道前，未对该通道的相关寄存器进行操作，则 DMA 将继续完成上次的传输。
 - 在重新使能 DMA 通道前，对任意配置相关寄存器（对应通道的 DMA_CHxCNT / DMA_CHxPADDR / DMA_CHxMADDR 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（对应通道的 DMA_CHxCNT / DMA_CHxPADDR / DMA_CHxMADDR 寄存器）进行操作前便使能 DMA 通道，则不会触发任何 DMA 传输。

12.4.2. 外设握手

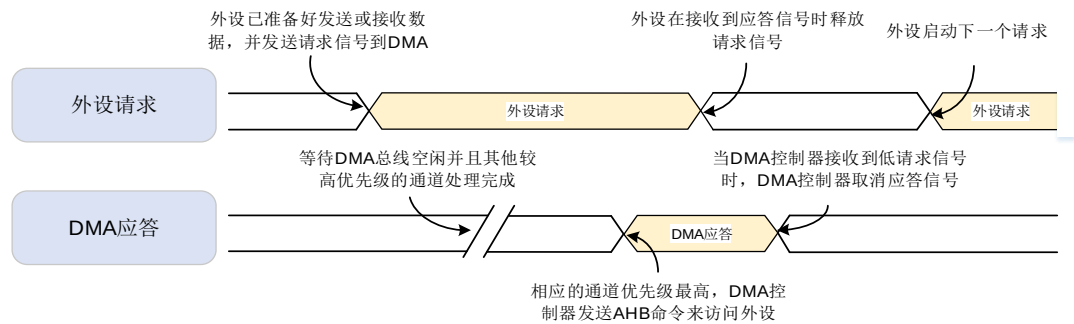
为了保证数据的有效传输，DMA控制器中引入了外设和存储器的握手机制，包括请求信号和应

答信号：

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

[图12-2. 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 12-2. 握手机制



12.4.3. 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。

12.4.4. 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。

12.4.5. 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。

12.4.6. 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。

12.4.7. 通道配置

要启动一次新的 DMA 数据传输，建议遵循以下步骤进行操作：

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

12.4.8. 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。

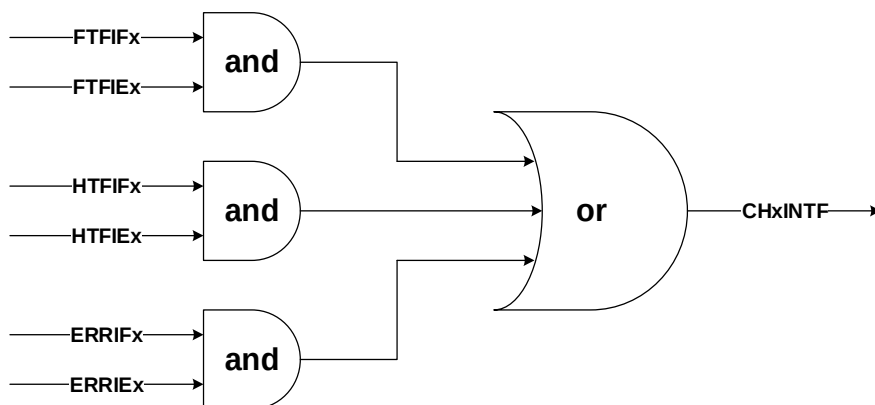
每一个中断事件在DMA_INTF寄存器中有专用的标志位，在DMA_INTC寄存器中有专用的清除位，在DMA_CHxCTL寄存器中有专用的使能位。[表12-2. 中断事件](#)描述了其对应关系。

表 12-2. 中断事件

中断事件	标志位	清除位	使能位
	DMA_INTF	DMA_INTC	DMA_CHxCTL
传输完成（Full transfer finish）	FTFIF	FTFIFC	FTFIE
半传输完成（Half transfer finish）	HTFIF	HTFIFC	HTFIE
传输错误（Transfer error）	ERRIF	ERRIFC	ERRIE

DMA中断逻辑如[图12-3. 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 12-3. DMA 中断逻辑图



注意：“x”表示通道数（DMA0对应x=0...5；DMA1对应x=0...5）

12.4.9. 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[表 13-3. DMAMUX 请求路由输入信号映射](#)。

12.5. DMA 寄存器

DMA0 基地址: 0x4002 0000

DMA1 基地址: 0x4002 0400

12.5.1. 中断标志位寄存器 (DMA_INTF)

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								ERRIF5	HTFIF5	FTFIF5	GIF5	ERRIF4	HTFIF4	FTFIF4	GIF4
								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
31:20	保留	必须保持复位值
23/19/15/11/7 /3	ERRIFx	通道x错误标志位(x=0...5) 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
22/18/14/10/6 /2	HTFIFx	通道x半传输完成标志位(x=0...5) 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
21/17/13/9/5/ 1	FTFIFx	通道x传输完成标志位(x=0...5) 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
20/16/12/8/4/ 0	GIFx	通道x全局中断标志位(x=0...5) 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

12.5.2. 中断标志位清除寄存器 (DMA_INTC)

地址偏移: 0x04

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								ERRIFC5	HTFIFC5	FTFIFC5	GIFC5	ERRIFC4	HTFIFC4	FTFIFC4	GIFC4
								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
31:20	保留	必须保持复位值
23/19/15/11/7 /3	ERRIFCx	清除通道x(x=0...5)的错误标志位 0: 无影响 1: 清零DMA_INTF寄存器的ERRIFx位
22/18/14/10/6 /2	HTFIFCx	清除通道x(x=0...5)的半传输完成标志位 0: 无影响 1: 清零DMA_INTF寄存器的HTFIFx位
21/17/13/9/5/ 1	FTFIFCx	清除通道x(x=0...5)的传输完成标志位 0: 无影响 1: 清零DMA_INTF寄存器的FTFIFx位
20/16/12/8/4/ 0	GIFCx	清除通道x(x=0...5)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

12.5.3. 通道 x 控制寄存器（DMA_CHxCTL）

x = 0...5, x 为通道序号

地址偏移：0x08 + 0x14 × x

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

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

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	PRIQ[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

12.5.4. 通道 x 计数寄存器 (DMA_CHxCNT)

$x = 0 \dots 5$, x 为通道序号

地址偏移: $0x0C + 0x14 \times x$

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



位/位域	名称	描述
31:16	保留	必须保持复位值

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

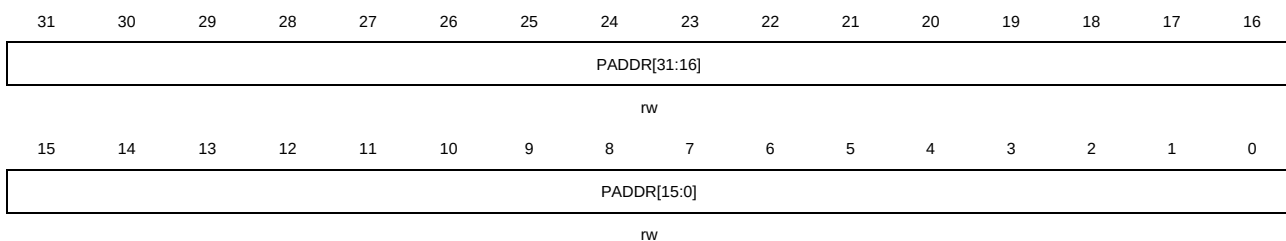
12.5.5. 通道 x 外设基地址寄存器 (DMA_CHxPADDR)

$x = 0 \dots 5$, x 为通道序号

地址偏移: $0x10 + 0x14 \times x$

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



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。

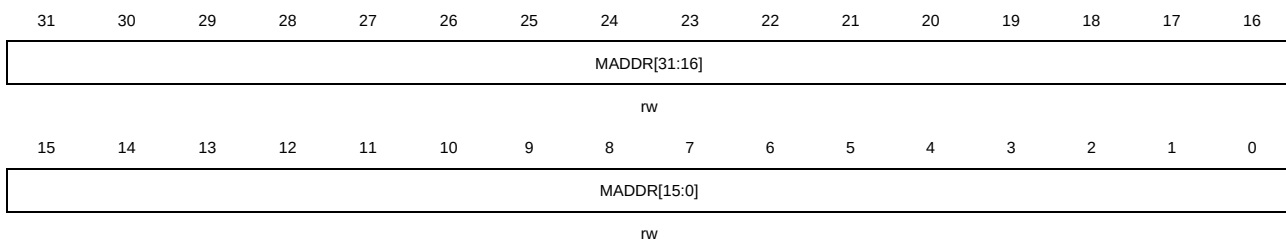
12.5.6. 通道 x 存储器基地址寄存器 (DMA_CHxMADDR)

$x = 0 \dots 5$, x 为通道序号

地址偏移: $0x14 + 0x14 \times x$

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



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。

13. DMA 请求多路复用器 (DMAMUX)

13.1. 简介

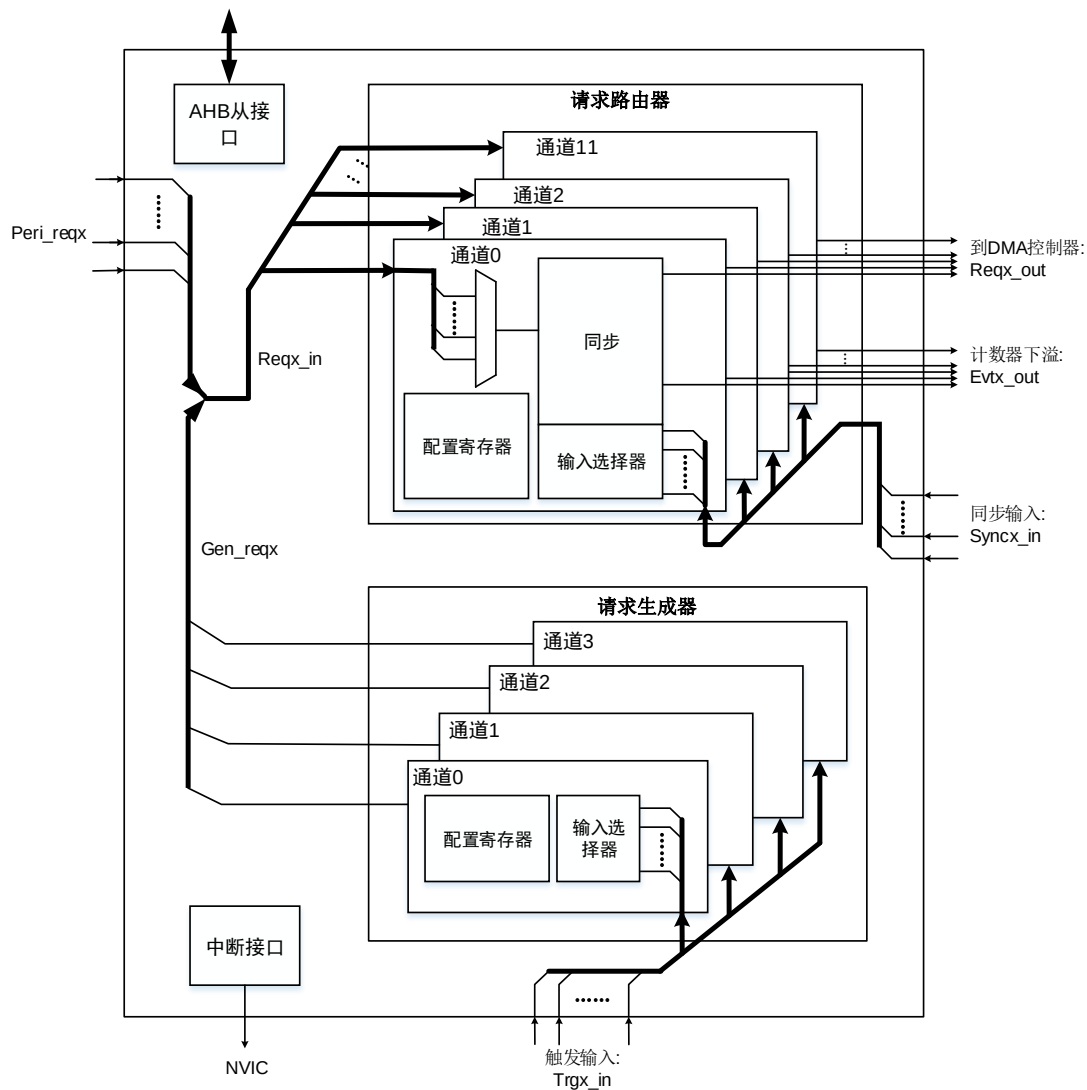
DMAMUX 是 DMA 请求的传输调度器。可编程的 DMA 请求多路复用器 DMAMUX，可在外设和 DMA 控制器之间路由 DMA 请求线路，或者 DMAMUX 也可以将可编程事件连入到输入触发信号上，作为一个 DMAMUX 请求发生器，再由 DMAMUX 请求路由器在 DMAMUX 请求生成器产生的 DMA 请求和 DMA 控制器之间路由 DMA 请求线路。每个 DMAMUX 请求路由通道选择一条唯一的 DMA 请求线路，无条件地或同步地从它的 DMAMUX 同步输入事件。DMA 请求信号会一直挂起，直到 DMA 控制器响应它，并且产生一个 DMA 确认信号，此时相应的 DMA 请求信号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

13.2. 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；
- 27 路触发输入信号到 DMAMUX 请求生成器；
- 27 路同步输入信号；
- 每个 DMAMUX 请求生成通道：
 - DMAMUX 请求触发输入选择器
 - DMAMUX 请求生成计数器
 - 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：
 - 84 路外设 DMAMUX 请求输入信号
 - 同步输入信号选择器
 - 对应一条 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
 - DMAMUX 请求路由计数器
 - 指示被选中的同步输入信号的事件溢出标志

13.3. 结构框图

图 13-1. DMAMUX 结构框图



13.4. 功能说明

如 [图 13-1. 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 2 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。

13.4.1. DMAMUX 信号

表 13-1. 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

13.4.2. DMAMUX 请求路由器

DMAMUX 请求路由器可在外设或者 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 13-3. DMAMUX 请求路由输入信号映射](#)。一个 DMAMUX 请求路由通道与对应的 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。

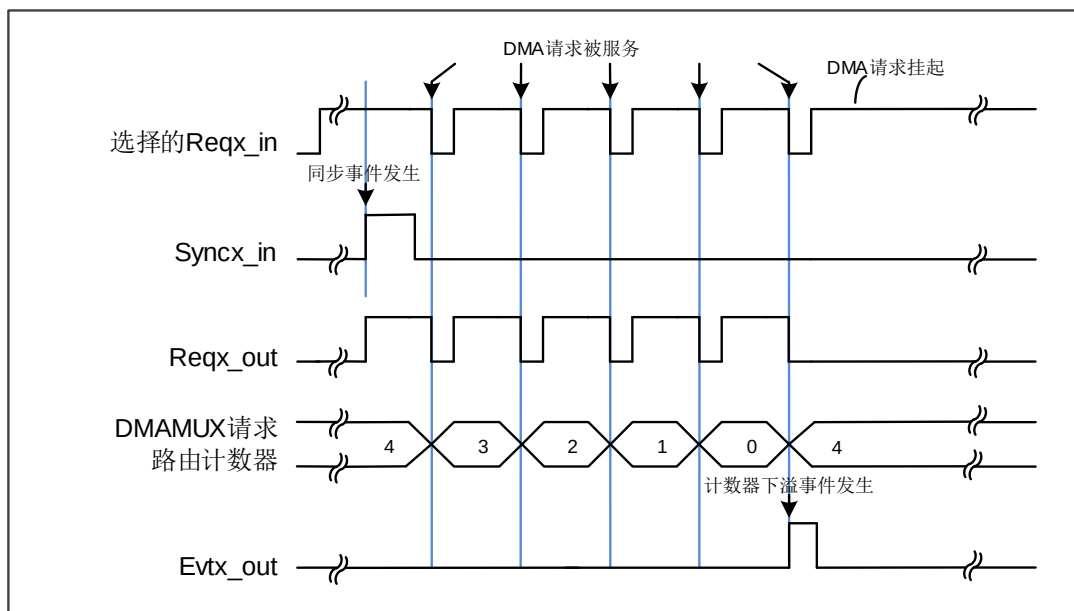
注意：只有当 DMAMUX 请求路由通道 x 的同步使能位 SYNCEN 位和通道事件输出使能位 EVGEN 位都为 0 时，才能配置其 NBR[4:0]位域。

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 13-2. 同步模式为当 $NBR[4:0]=4$, $SYNCEN=1$, $EVGEN=1$, $SYNCP[1:0]=0b01$ 时的举例。

图 13-2. 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 13-5. 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 SYNCP[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。

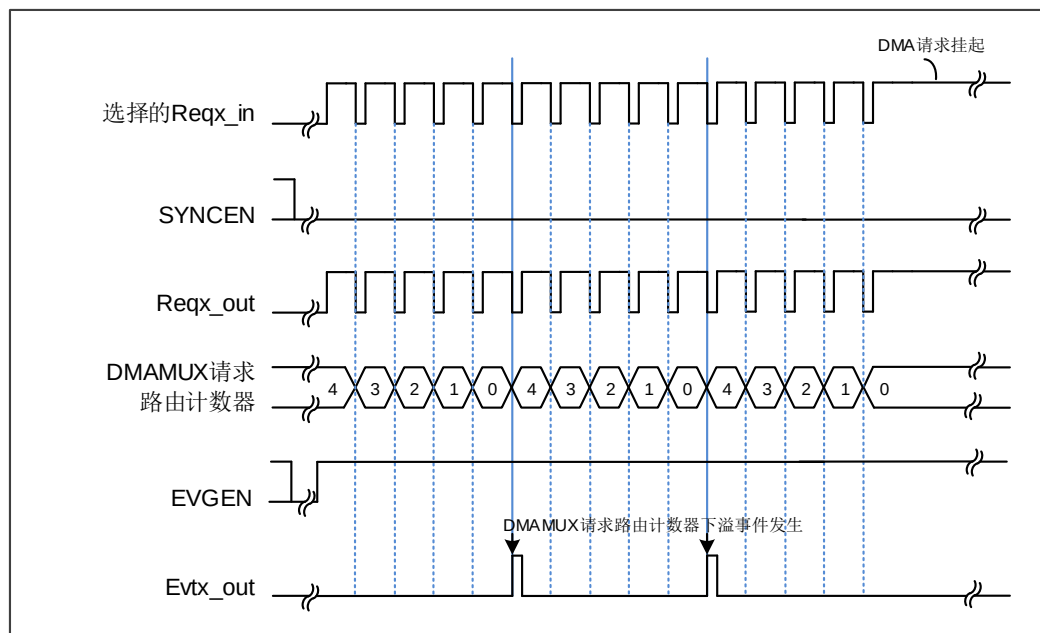
通道事件输出

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 13-3. 通道事件输出为当 NBR[4:0]=4，SYNCEN=0，EVGEN=1 时的举例。

图 13-3. 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 来禁能 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。

13.4.3. 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合至所有 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 13-4. 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_INTF 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。

13.4.4. 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。

13.4.5. 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13-2. 中断事件](#)描述了其对应关系。

表 13-2. 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	SOIFx	SOIFCx	SOIE
DMAMUX 请求生成通道 y 上的触发溢出事件	TOIFy	TOIFCy	TOIE

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`。

13.4.6. DMAMUX 映射

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 13-3. 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 13-3. 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_Group_pri1
6	ADC2_Group_pri1
7	ADC0_Group_pri2
8	ADC2_Group_pri2
9	ADC0_Group_pri3
10	ADC2_Group_pri3
11	I2C_RX
12	I2C_TX
13	UART0_RX
14	UART0_TX
15	UART1_RX
16	UART1_TX
17	UART2_RX
18	UART2_TX
19	UART3_RX
20	UART3_TX
21	SPI0_RX
22	SPI0_TX

请求路由通道输入标识 MUXID[6:0]	来源
23	TIMER0_UP
24	TIMER0_CO
25	TIMER0_TI
26	TIMER0_CH0
27	TIMER0_CH1
28	TIMER0_CH2
29	TIMER0_CH3
30	TIMER0_MCH0
31	TIMER0_MCH1
32	TIMER0_MCH2
33	TIMER0_MCH3
34	TIMER7_UP
35	TIMER7_CO
36	TIMER7_TI
37	TIMER7_CH0
38	TIMER7_CH1
39	TIMER7_CH2
40	TIMER7_CH3
41	TIMER7_MCH0
42	TIMER7_MCH1
43	TIMER7_MCH2
44	TIMER7_MCH3
45	TIMER1_UP
46	TIMER1_TI
47	TIMER1_CH0
48	TIMER1_CH1
49	TIMER1_CH2
50	TIMER1_CH3
51	TIMER2_UP
52	TIMER2_TI
53	TIMER2_CH0
54	TIMER2_CH1
55	TIMER2_CH2
56	TIMER2_CH3
57	TMU_TX
58	TMU_RX
59	GPTIMER0_UP
60	GPTIMER0_TI
61	GPTIMER0_CH0
62	GPTIMER0_CH1

请求路由通道输入标识 MUXID[6:0]	来源
63	GPTIMER1_UP
64	GPTIMER1_TI
65	GPTIMER1_CH0
66	GPTIMER1_CH1
67	EVIC0
68	EVIC1
69	EVIC2
70	EVIC3
71	EVIC4
72	EVIC5
73	EVIC6
74	EVIC7
75	EVIC8
76	EVIC9
77	ADC0_Group_pri4
78	ADC0_Group_pri4
79	GPTIMER0_CH0COMV_ADD
80	GPTIMER0_CH1COMV_ADD
81	GPTIMER1_CH0COMV_ADD
82	GPTIMER1_CH1COMV_ADD
83	EVIC10
84	EVIC11

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 13-4. 触发输入信号映射](#)。

表 13-4. 触发输入信号映射

触发输入标识TID[4:0]	来源
0	EXTI_0
1	EXTI_1
2	EXTI_2
3	EXTI_3
4	EXTI_4
5	EXTI_5
6	EXTI_6
7	EXTI_7
8	EXTI_8
9	EXTI_9
10	EXTI_10

触发输入标识TID[4:0]	来源
11	EXTI_11
12	EXTI_12
13	EXTI_13
14	EXTI_14
15	EXTI_15
16	Evtx_out0
17	Evtx_out1
18	Evtx_out2
19	Evtx_out3
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved

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13-5. 同步输入信号映射](#)。

表 13-5. 同步输入信号映射

同步输入标识SYNCID[4:0]	来源
0	EXTI_0
1	EXTI_1
2	EXTI_2
3	EXTI_3
4	EXTI_4
5	EXTI_5
6	EXTI_6
7	EXTI_7
8	EXTI_8
9	EXTI_9
10	EXTI_10
11	EXTI_11
12	EXTI_12
13	EXTI_13
14	EXTI_14
15	EXTI_15
16	Evtx_out0
17	Evtx_out1

同步输入标识SYNCID[4:0]	来源
18	Evtx_out2
19	Evtx_out3
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved

13.5. DMAMUX 寄存器

DMAMUX基地址：0x4002 0800

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

地址偏移：0x00 + 0x04 * x（x = 0...11，其中 x 为通道序号）

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				SYNCID[4:0]				NBR[4:0]				SYNCP[1:0]		SYNCEN	
				rw				rw				rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						EVGEN		SOIE	保留	MUXID[7: 0]					
						rw		rw							

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。

13.5.2. 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 (x=0..11) 的同步溢出事件标志位 当DMAMUX请求路由通道x发生了同步输入事件, 而此时DMAMUX请求路由计数器值小于NBR[4:0], 则该通道的同步溢出标志位置位。 通过对DMAMUX_RM_INTC寄存器的SOIFC _x 位写1来清除相应通道的同步溢出标志。

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

地址偏移: 0x084

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	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 (x=0..11) 的同步溢出事件标志清除位 写1可清除相应通道在DMAMUX_RM_INTF寄存器的同步溢出标志SOIFx。

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

地址偏移: $0x100 + 0x04 * x$ (x = 0...3, 其中 x 为通道序号)

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。

13.5.5. 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。

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												TOIF3	TOIF2	TOIF1	TOIF0
												r	r	r	r

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 (x=0..3) 的触发溢出标志位 如果在DMAMUX请求生成计数器 (通过DMAMUX_RG_CHxCFG寄存器的NBRG[4:0]位域配置) 发生下溢之前, DMAMUX请求生成通道x发生了一个新的触发输入事件, 则该标志位置位。 通过对DMAMUX_RG_INTFC寄存器的TOIFCx位写1来清除相应通道的触发溢出标志。

13.5.6. 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。

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												TOIFC3	TOIFC2	TOIFC1	TOIFC0
												w	w	w	w

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

14. 调试 (DBG)

14.1. 简介

GD32M531 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 (SWD) 调试和跟踪功能, 也支持 JTAG 调试。调试系统也包括一线调试接口, 通过 GD-Link 支持。跟踪功能请参考下列文档:

- Cortex®-M33 技术参考手册;

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

14.2. JTAG/SW 功能说明

调试工具可以通过串行 (SW) 调试接口或者 JTAG 调试接口来访问调试功能。

14.2.1. 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。

14.2.2. 引脚分配

JTAG 调试提供五个引脚的接口: JTAG 时钟引脚 (JTCK), JTAG 模式选择引脚 (JTMS), JTAG 数据输入引脚 (JTDI), JTAG 数据输出引脚 (JTDO), JTAG 复位引脚 (NJTRST, 低电平有效)。串行调试 (SWD) 提供两个引脚的接口: 数据输入输出引脚 (SWDIO) 和时钟引脚 (SWCLK)。SW 调试接口的两个引脚与 JTAG 调试接口的两个引脚复用, SWDIO 和 JTMS 复用, SWCLK 和 JTCK 复用。

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

调试引脚分配如下:

表 14-1. 引脚分配

引脚 (LFQFP64 且 PN7 引脚为高)	调试接口	引脚 (LFQFP48)	调试接口
PG13	JTDI	-	-
PG12	JTCK/SWCLK	PN2	SWCLK
PG14	JTMS/SWDIO	PG13	SWDIO
PG15	NJTRST	-	-
PG11	JTDO	-	-
BOOT_PN6	一线	BOOT_PN6	一线

注意：当芯片为LFQFP48封装时，只有选项字节中SWDMOD被置1时这些引脚被用于调试接口。

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（NJTRST硬件拉高）。如果切换到SW调试模式，JTDI / NJTRST / JTDO释放作为普通GPIO功能。如果JTAG和SW调试功能都没有使用，这五个引脚都释放作为普通GPIO功能。五个引脚具体配置请参考[GPIO 引脚配置](#)。

14.2.3. JTAG 链状结构

Cortex®-M33内核的JTAG TAP和MCU JTAG TAP串行连接。MCU 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MCU 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MCU JTAG已处在BYPASS模式。

MCU JTAG ID代码是0x790007A3。

14.2.4. 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和调试逻辑（FPB，DWT，ITM）。NJTRST能复位JTAG TAP控制器。所以，可以在系统复位下实现调试功能。例如：复位后停止，用户在系统复位后配置相应停止位，系统复位释放后处理器会立即停止。

14.2.5. JEDEC-106 ID 代码

Cortex®-M33 集成了 JEDEC-106 ID 代码。位于 ROM 表中，映射地址为 0xE00FF000_0xE00FFFFF。

14.3. 调试保持功能说明

14.3.1. 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_IRC32M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_IRC32M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。

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

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

对于TIMER外设，TIMER计数器停止并进行调试；

对于I2C外设，SMBUS保持状态并进行调试；

对于WWDGT或者FWDGT外设，计数器时钟停止并进行调试；

对于CAN外设，接收寄存器停止计数并进行调试。

14.4. DBG 寄存器

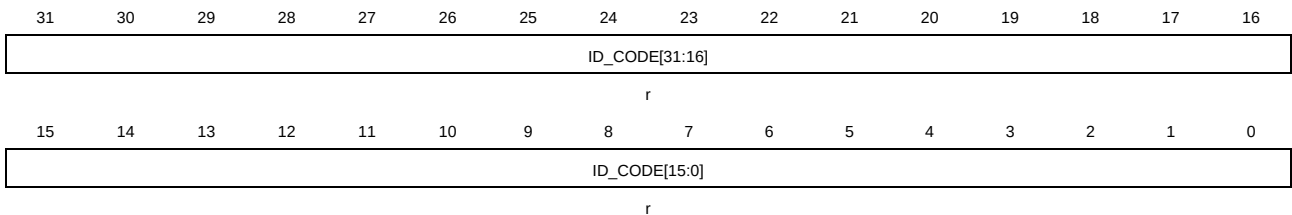
DBG基地址：0xE004 4000

14.4.1. 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



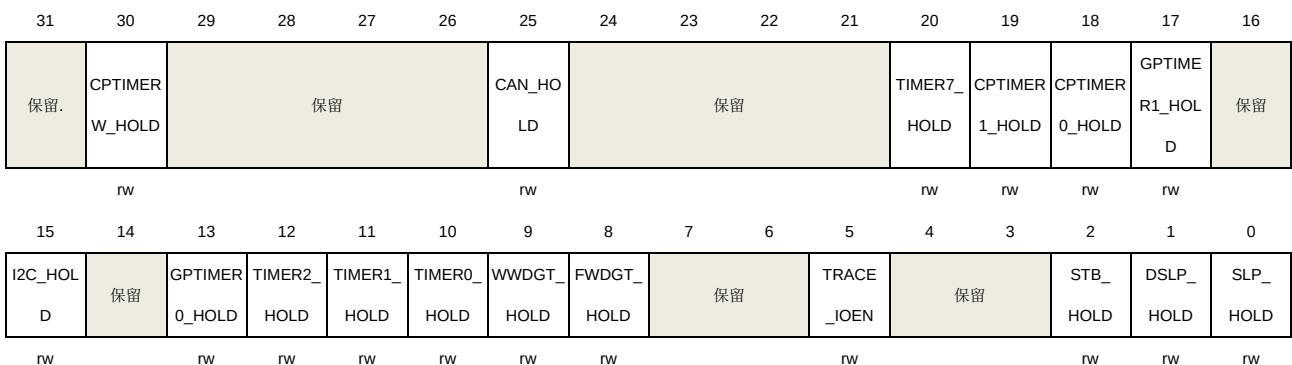
位/位域	名称	描述
31:0	ID_CODE[31:0]	DBG ID 寄存器 这些位由软件读取，这些位是不变的常数

14.4.2. 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	保留	必须保持复位值
30	CPTIMERW_HOLD	CPTIMERW 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 CPTIMERW 计数器不变, 用于调试
29:26	保留	必须保持复位值
25	CAN_HOLD	CAN 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 CAN 接收寄存器停止接收数据
24:21	保留	必须保持复位值
20	TIMER7_HOLD	TIMER7 保持寄存器 该位由软件置位和复位 0: 无影响 1: 当内核停止时保持定时器 7 计数器不变, 用于调试
19	CPTIMER1_HOLD	CPTIMER1 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 CPTIMER1 计数器不变, 用于调试
18	CPTIMER0_HOLD	CPTIMER0 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 CPTIMER0 计数器不变, 用于调试
17	GPTIMER1_HOLD	GPTIMER1 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 GPTIMER1 计数器不变, 用于调试
16	保留	必须保持复位值
15	I2C_HOLD	I2C 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 I2C 的状态不变以避免 SMBUS 超时, 用于调试
14	保留	必须保持复位值
13	GPTIMER0_HOLD	GPTIMER0 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 GPTIMER0 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	保留	必须保持复位值
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_IRC32M 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_IRC32M 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

15. 模数转换器（ADC）

15.1. 简介

MCU片上集成了两个12位逐次逼近式模数转换器模块（ADC0/2），并且独立运行。

ADC0可以采样来自于7个外部通道的模拟信号。ADC2可以采样来自于10个外部通道和2个内部通道(V_{REFINT} , V_{SENSE})的模拟信号。在每个ADC模块中，这些被选择的采样通道都支持各种扫描模式：Group_pri1组单次扫描模式、Group_pri1组连续扫描模式，或者多组扫描模式。在每个通道采样和转换之后，转换结果可以根据最低有效位（LSB）对齐或最高有效位（MSB）对齐存储在相应的数据寄存器中。

在Group_pri1单次扫描模式下，所选的外部通道按默认通道优先级以升序顺序进行转换。在Group_pri1组连续扫描模式下，所选的外部通道按默认通道优先级以升序顺序连续进行转换。在多组扫描模式下，所选外部通道可以任意分为两个组（Group_pri1 & Group_pri2）、三个组（Group_pri1 & Group_pri2 & Group_pri3 或者 Group_pri1 & Group_pri2 & Group_pri4）或四个组（Group_pri1 & Group_pri2 & Group_pri3 & Group_pri4）。当相应组的触发被响应时，每个组中的通道按默认通道优先级以升序顺序进行一次转换。

通道优先级是可配置的，而组优先级是固定的（Group_pri1 > Group_pri2 > Group_pri3 > Group_pri4）。在低优先级组的通道转换过程中，如果接收到高优先级组的触发信号，ADC将中止当前转换并开始高优先级组的通道转换。高优先级组转换完成后，低优先级组的转换可以从上次中止的通道或扫描起始通道继续进行，也可以不转换低优先级。

在分叉触发模式下，每个单独扫描的组中只能选择一个通道，第一次触发信号的转换结果和第二次触发信号的转换结果会存储到不同的数据寄存器。

当使能自诊断功能时，在每次扫描开始时执行一次自诊断，会转换三个内部自诊断电压中的一个。

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。

15.2. 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；
 - 采保电路可编程采样和保持时间；
 - 可编程的断开检测（预充放电时间）；
 - 数据存储模式：最高有效位（MSB）对齐和最低有效（LSB）位对齐；
 - 支持DMA请求。
- 模拟功能：
 - ADC0有7个外部模拟输入通道；
 - ADC2有10个外部模拟输入通道，1个内部参考电压通道，1个内部温度传感器通道；

- 通道专用的采保电路（仅ADC0的三个通道）；
- 模拟输入断开检测辅助功能（预放电/预充电功能）；
- 转换开始：
 - 软件；
 - 同步触发（定时器/事件内部互联模块）；
 - 异步触发。A/D转换可以通过外部触发ADC0_TRG（ADC0），ADC2_TRG（ADC2）引脚触发（两个ADC独立）。
- 运行模式：
 - Group_pri1组单次扫描模式：对Group_pri1组中所选的通道执行一次转换；
 - Group_pri1组连续扫描模式：对Group_pri1组中所选的通道重复执行转换；
 - 多组扫描模式：两个组（(Group_pri1 & Group_pri2)），三个组（Group_pri1 & Group_pri2 & Group_pri3 或者 Group_pri1 & Group_pri2 & Group_pri4）或四个组（Group_pri1 & Group_pri2 & Group_pri3 & Group_pri4）。在每个组中对所选通道执行一次A/D转换。
- 转换结果阈值监测功能：模拟看门狗A和模拟看门狗B。
- 分叉触发模式：同一通道转换结果存储到不同数据寄存器
- 中断产生：
 - Group_pri1/Group_pri2/Group_pri3/Group_pri4的转换轮次结束中断；
 - 模拟看门狗事件（看门狗A通道x比较状态或窗口看门狗B比较标志）；
 - Group_pri1/Group_pri2/Group_pri3/Group_pri4的DMA溢出事件。
- 过采样：可选择A/D转换值加法模式或平均模式
 - 32位的数据寄存器；
 - 加法次数从1、2、3、4、16、32、64次中选择。
- 通道专用数据寄存器
- ADC输入范围： $V_{REFN} \leq V_{IN} \leq V_{REFP}$ 。

15.3. 引脚和内部信号

[表 15-1. ADC内部输入信号](#)和[表 15-2. ADC输入引脚定义](#)给出了ADC内部信号和引脚说明。

表 15-1. ADC 内部输入信号

内部信号名称	信号类型	说明
V _{REFINT}	输入	内部参考输出电压（仅ADC2）
V _{SENSE}	输入	内部温度传感器输出电压（仅ADC2）

表 15-2. ADC 输入引脚定义

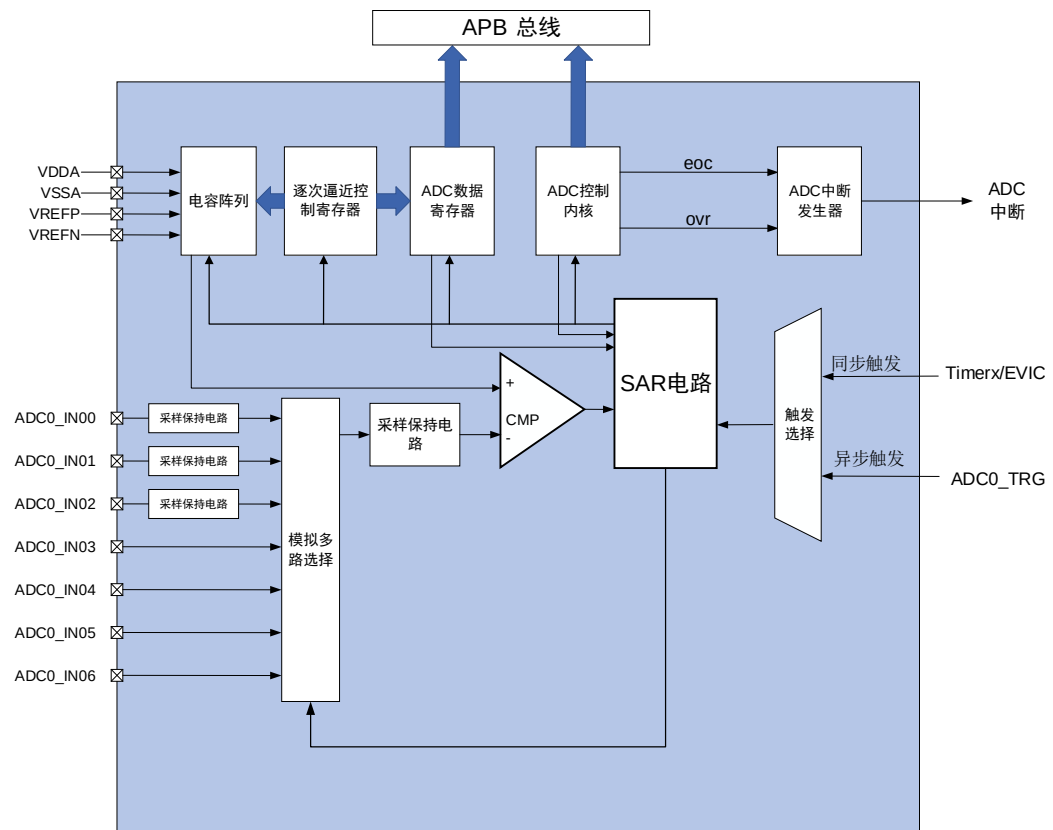
名称	信号类型	注释
V _{DDA}	输入	模拟电源，模拟电源输入等于V _{DD}
V _{SSA}	输入	模拟电源地，等于V _{SS}
V _{REFP}	输入	ADC正参考电压
V _{REFN}	输入	ADC负参考电压，V _{REFN} = V _{SSA}
ADCx_INy	输入	ADC0多达7路外部通道 ADC2多达10路外部通道

注意：在有些封装中，VREFP和VDDA是连在一起的，VREFN和VSSA是连在一起的。

15.4. 结构框图

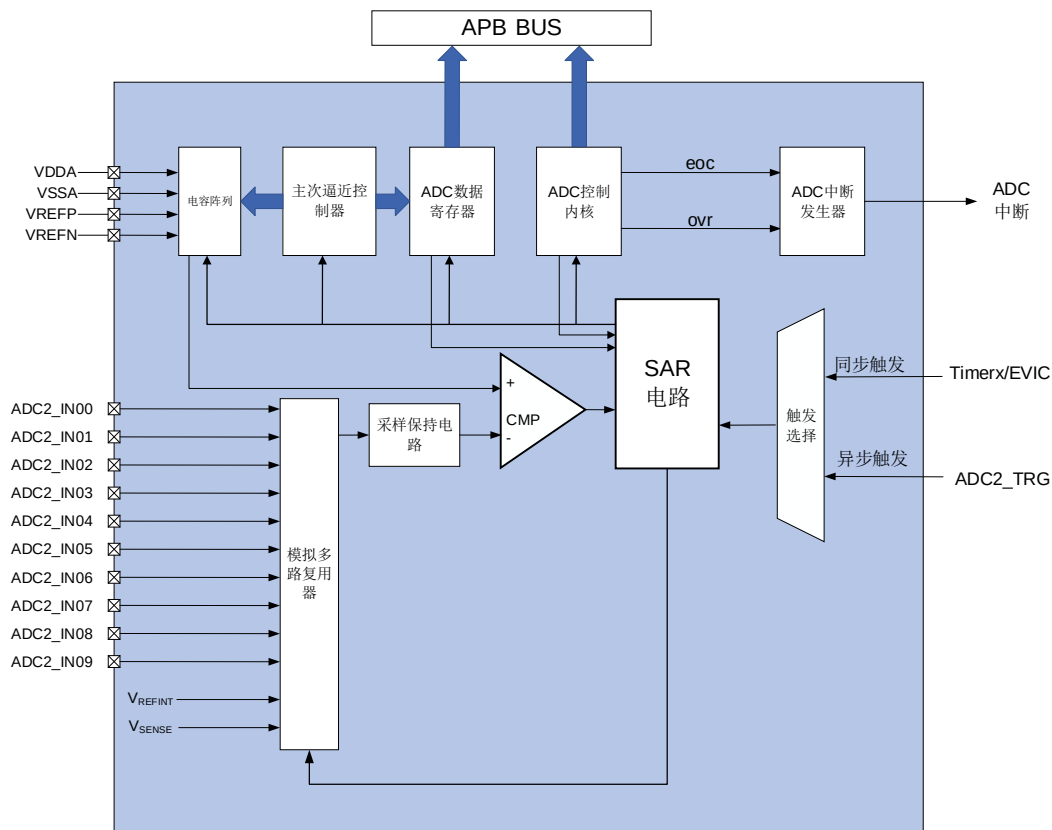
[图15-1. ADC0模块框图](#)给出了ADC0模块框图。

图 15-1. ADC0 模块框图



[图15-2. ADC2模块框图](#)给出了ADC2模块框图。

图 15-2. ADC2 模块框图



15.5. 功能说明

15.5.1. ADC 时钟

CK_ADC时钟是由时钟控制器提供的，有四个时钟源（由RCU_CFG2寄存器中的ADCSEL[1:0]定义）：

- CK_APB2（跟AHB时钟同步）
- CK_PLL（跟AHB时钟同步）
- CK_HXTAL（跟AHB时钟异步）
- CK_IRC32M（跟AHB时钟异步）

ADC时钟由RCU_CFG2寄存器中ADCPSC[4:0]位进行分频和配置。CK_ADC最大频率是36MHz。

注意：不同电压等级，ADC最高工作频率不同，具体请参考数据手册。

15.5.2. ADCON 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} 数值详见芯片数据手册。

15.5.3. 扫描运行模式

在扫描运行模式中，ADC按照指定的通道优先级对所选择的通道进行顺序转换。通道优先级定义在ADC_CHPRix (x=0,1) 寄存器中。

ADC支持三种扫描运行模式（由ADC_CTL0寄存器的GSCAN[1:0]位域配置）：

- Group_pri1组单次扫描模式
- 多组扫描模式
- Group_pri1组连续扫描模式

在Group_pri1组单次扫描模式中，ADC转换一次所有被选择的通道。在Group_pri1组连续扫描模式中，ADC连续转换所选择的通道，通过ADC_CTL0寄存器的SWEND位写1可以结束所有组的转换。在多组扫描模式中，每个组在各自的同步触发后转换一次所选择的通道。

当使能自诊断时，在每次扫描开始时，会执行一次自诊断，转换一个自诊断的电压。

在Group_pri1组单次模式下，将ADC_BITRGCTL寄存器的GP1BIMEN位置1可以使能分叉触发模式。在多组扫描模式中，将ADC_BITRGCTL寄存器的GPxBIMEN位置1可以使能Group_prix组的分叉触发模式，分叉触发功能可用于所有组。当启用了分叉触发模式时，如果GPxTRGSRC[5:0]位选择的同步触发源触发ADC转换，ADC_BITRGCTL寄存器的GPxBICHSEL[4:0]位选择的通道A/D转换数据将被复制。

ADCx_PROCF引脚用于输出ADC_CTL0的PROCF位（只读）的状态。

扩展分叉触发模式指的是在分叉模式下，通过ADC_GPCGF0/1寄存器中的GPxTRGSRC[5:0]位启用两个同步触发源时的一种状态（GPxTRGSRC[5:0]配置为 6'b000110, 6'b001110, 6'b100101, 6'b101101 or 6'b110011）。在扩展分叉触发模式下，除了在分叉触发模式中的正常操作外，A/D转换数据还会根据触发类型存储在ADC_GPxBIDATA1或ADC_GPxBIDATA2中。将TRGA触发的转换结果绑定到ADC_GPxBIDATA1寄存器，将TRGB触发的转换结果绑定到ADC_GPxBIDATA2数据寄存器。EOCx仅在三组触发的转换结束后置位。

当通过ADC_SHCTL寄存器中的SH00EN/SH01EN/SH02EN位设置ADC0_IN0/ADC0_IN1/ADC0_IN2中的任意一个通道为通道专用采样保持电路时，目标模拟输入将在每次扫描的第一次A/D转换之前被采样和保持。

注意：如果在由另一个触发引起的A/D转换过程中输入了新的触发，新触发将被忽略。

15.5.4. Group_pri1 组单次扫描模式

在下面的例子中，通道优先级寄存器ADC_CHPRix设置为默认值。

基本运行（无通道专用采样保持电路）

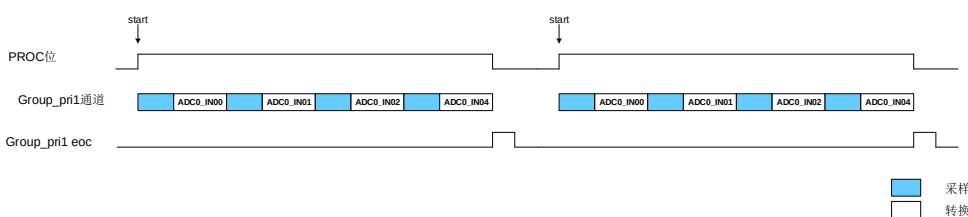
在Group_pri1连续扫描模式的基本运行中，对指定通道的模拟输入进行一次A/D转换。例如，选择ADC0的ADC0_IN0、ADC0_IN1、ADC0_IN2、ADC0_IN4，如[图15-3. ADC0的单次扫描基本运行（无通道专用采样保持电路）](#)所示。

- (1) 当通过软件设置ADC_EOCCTL寄存器中的GP1SWST位为1，或者输入同步或异步触发

信号时，将ADC_CHSELx寄存器所选择的这些通道执行A/D转换，从最小编号的通道开始（此例中为ADC0_IN0）。

- (2) 每次单个通道的A/D转换完成时，A/D转换结果存储到相应的A/D数据寄存器中。
- (3) 当所有选定通道的A/D转换完成时，ADC_EOCCTL寄存器中的EOC1F位被置为1。当（EOC1RCNT[2:0] + 1）轮Group_pri1转换结束时，EOC1RF标志将被置为1，如果ADC_EOCCTL中的EOC1RIE位为1，则生成中断请求。
- (4) 在A/D转换期间，ADC_CTL0中的PROC位保持为1（A/D转换启动），当所有选定通道的A/D转换完成后，PROC位自动清零为0，然后A/D转换器进入等待状态。

图 15-3. ADC0 的单次扫描基本运行（无通道专用采样保持电路）



基本运行（使用通道专用采样保持电路，禁用恒采样模式）

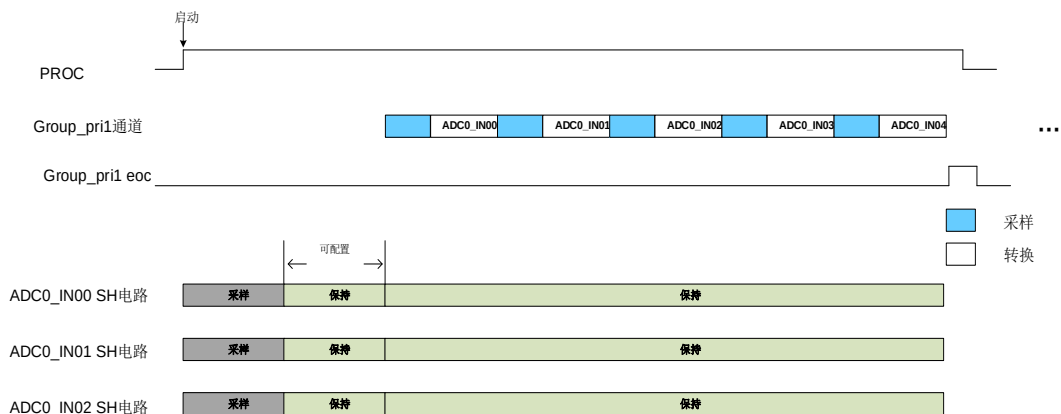
当使用通道专用采样保持电路时，首先执行采样保持操作，然后对所有选定通道的模拟输入进行一次A/D转换。例如，选择ADC0的ADC0_IN0、ADC0_IN1、ADC0_IN2、ADC0_IN3、ADC0_IN4，如[图15-4. ADC0的单次扫描基本运行（使用通道专用采样保持电路，禁用恒采样模式）](#)所示。

ADC_SHCTL寄存器中的SH00EN、SH01EN、SH02EN位用来选择要使用专用采样保持电路的通道。

将ADC_SHCTL寄存器中的SHMD位清零以禁用恒采样模式。

- (1) 当软件设置ADC_EOCCTL寄存器中的GP1SWST位为1，或者输入同步或异步触发信号时，开始对所有选定的使用专用采样保持电路的通道进行模拟输入采样。
- (2) 在采样保持操作之后，对选定的通道进行A/D转换，从编号最小的通道开始（此例中为ADC0_IN0）。
- (3) 每次单个通道的A/D转换完成时，A/D转换结果存储到相应的A/D数据寄存器中。
- (4) 当所有选定通道的A/D转换完成时，ADC_EOCCTL寄存器中的EOC1F位被置为1。当（EOC1RCNT[2:0] + 1）轮Group_pri1转换结束时，EOC1RF标志将被置为1，如果ADC_EOCCTL中的EOC1RIE位为1，则生成中断请求。
- (5) 在A/D转换期间，ADC_CTL0中的PROC位保持为1（A/D转换启动），当所有选定通道的A/D转换完成后，PROC位自动清零为0，然后A/D转换器进入等待状态。

图 15-4. ADC0 的单次扫描基本运行（使用通道专用采样保持电路，禁用恒采样模式）



基本运行（使用通道专用采样保持电路，使能恒采样模式）

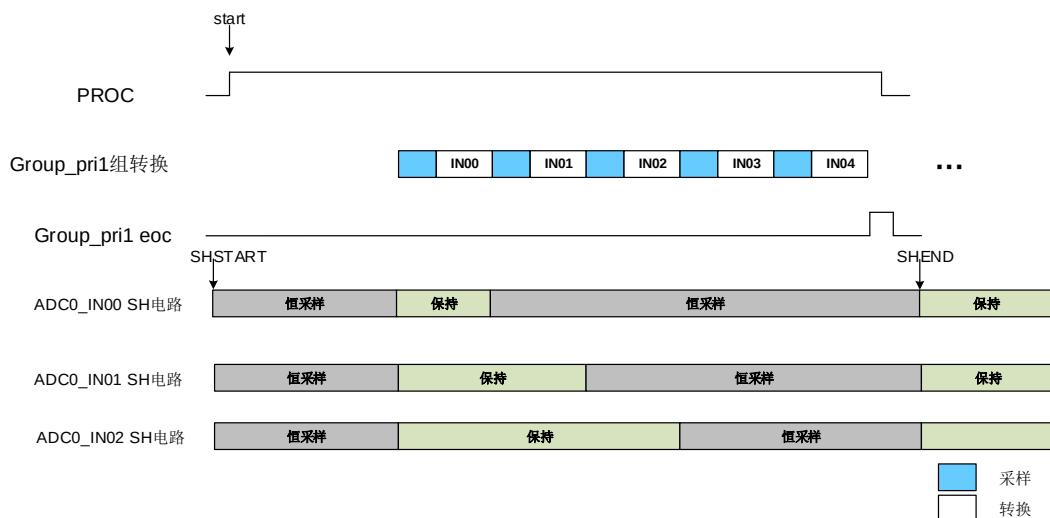
当使用通道专用采样保持电路时，首先执行采样保持操作，然后对所有选定通道的模拟输入进行一次A/D转换。例如，选择ADC0的ADC0_IN0、ADC0_IN1、ADC0_IN2、ADC0_IN3、ADC0_IN4，如[图15-5. ADC0的单次扫描基本运行（使用通道专用采样保持电路，使能恒采样模式）](#)所示。

ADC_SHCTL寄存器中的SH00EN、SH01EN、SH02EN位用来选择要使用专用采样保持电路的通道。

将ADC_SHCTL寄存器中的SHMD位置1以使能恒采样模式。

- (1) 将ADC_SHCTL寄存器中的SHSTART位设置为1，选定的通道专用采样保持电路开始工作。
- (2) 当软件设置ADC_EOCCTL寄存器中的GP1SWST位为1，或者输入同步或异步触发信号时，开始对所有选定的使用专用采样保持电路的通道进行模拟输入采样。
- (3) 在采样保持操作之后，对选定的通道进行A/D转换，从编号最小的通道开始（此例中为ADC0_IN0）。
- (4) 每次单个通道的A/D转换完成时，A/D转换结果存储到相应的A/D数据寄存器中。
- (5) 当所有选定通道的A/D转换完成时，ADC_EOCCTL寄存器中的EOC1F位被置为1。当（EOC1RCNT[2:0] + 1）轮Group_pri1转换结束时，EOC1RF标志将被置为1，如果ADC_EOCCTL中的EOC1RIE位为1，则生成中断请求。
- (6) 在A/D转换期间，ADC_CTL0中的PROC位保持为1（A/D转换启动），当所有选定通道的A/D转换完成后，PROC位自动清零为0，然后A/D转换器进入等待状态。
- (7) 将ADC_SHCTL寄存器中的SHEND位设置为1，以终止选定的通道专用采样保持电路的工作。

图 15-5. ADC0 的单次扫描基本运行（使用通道专用采样保持电路，使能恒采样模式）



通道选择和自诊断（无通道专用采样保持电路）

将ADC_CTL0寄存器中的SDSEN位设置为1以使能自诊断。

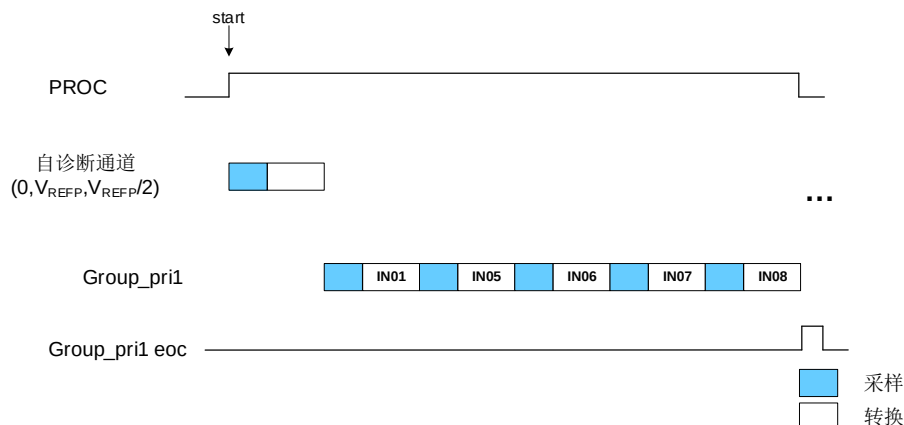
当通道和自诊断都选择时，12位A/D转换器先对参考电压执行一次A/D转换，如[图15-6. ADC2 单次扫描模式中的通道选择和自诊断（无专用的采样保持电路）](#)所示。之后，再对所选通道的模拟输入执行一次A/D转换。在这个例子中，所选通道是ADC2_IN1、ADC2_IN5、ADC2_IN6、ADC2_IN7和ADC2_IN8。

当ADC_CTL0寄存器中的SDSFIX位为1时，将转换一个固定的自诊断电压（由ADC_CTL0寄存器中的SDVOG[1:0]配置）。

当ADC_CTL0寄存器中的SDSFIX位为0时，每次扫描会轮流转换一个自诊断电压：0、 $V_{REFP}/2$ 、 V_{REFP} 。

- (1) 当软件设置 ADC_EOCCTL 寄存器中的 GP1SWST 位为 1，或者输入同步或异步触发信号时，开始转换自诊断电压。
- (2) 当自诊断转换完成后，转换结果将存储到自诊断数据寄存器（ADC_SDDATA）中，并且从编号最小的通道开始执行所选通道的模数转换（此例中为ADC2_IN1）
- (3) 每次单个通道的A/D转换完成时，A/D转换结果存储到相应的A/D数据寄存器中。
- (4) 当所有选定通道的A/D转换完成时，ADC_EOCCTL 寄存器中的EOC1F位被置为1。当（EOC1RCNT[2:0] + 1）轮Group_pri1转换结束时，EOC1RF标志将被置为1，如果ADC_EOCCTL中的EOC1RIE位为1，则生成中断请求。
- (5) 在A/D转换期间，ADC_CTL0中的PROC位保持为1（A/D转换启动），当所有选定通道的A/D转换完成后，PROC位自动清零为0，然后A/D转换器进入等待状态。

图 15-6. ADC2 单次扫描模式中的通道选择和自诊断（无专用的采样保持电路）



通道选择和自诊断（使用通道专用采样保持电路，禁用恒采样模式）

将ADC_CTL0寄存器中的SDSEN位设置为1以使能自诊断。

当使用通道，自诊断和通道专用采样保持电路都选择时，首先执行采样保持操作，然后12位A/D转换器对自诊断电压执行一次A/D转换，如[图15-7. ADC0单次扫描模式中的通道选择和自诊断（使用通道专用采样保持电路，禁用恒采样模式）](#)所示。之后，再对所选通道的模拟输入执行一次A/D转换。在这个例子中，所选通道是ADC0_IN0、ADC0_IN1、ADC0_IN2、ADC0_IN3和ADC0_IN4。

当ADC_CTL0寄存器中的SDSFIX位为1时，将转换一个固定的自诊断电压（由ADC_CTL0寄存器中的SDVOG[1:0]配置）。

当ADC_CTL0寄存器中的SDSFIX位为0时，每次扫描会轮流转换一个自诊断电压：0、VREFP/2、VREFP。

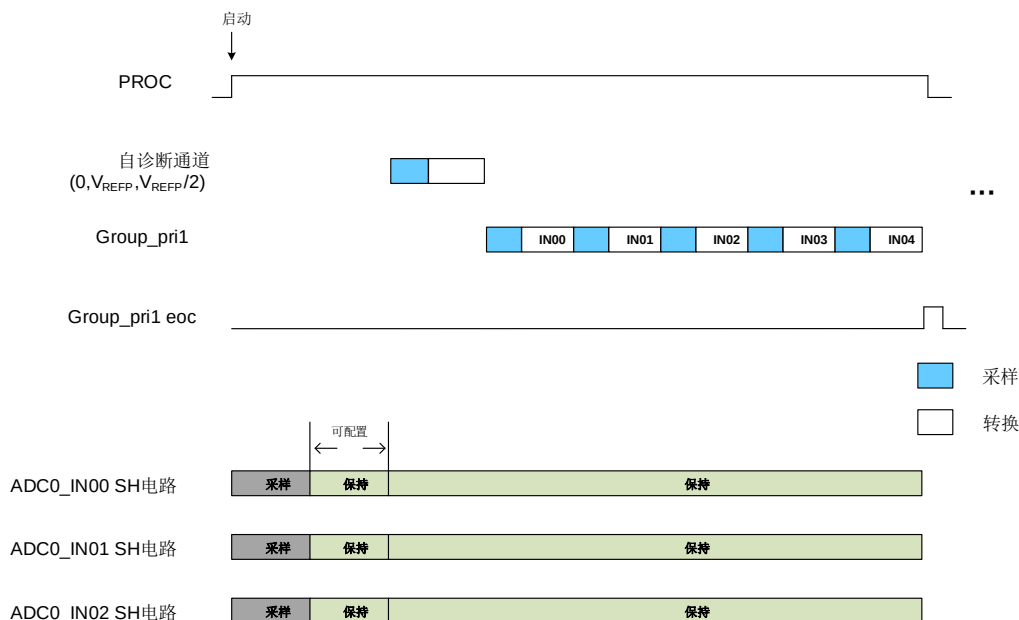
ADC_SHCTL寄存器中的SH00EN、SH01EN、SH02EN位用来选择要使用专用采样保持电路的通道。

将ADC_SHCTL寄存器中的SHMD位清零以禁用恒采样模式。

- (1) 当软件设置 ADC_EOCCTL 寄存器中的 GP1SWST 位为 1，或者输入同步或异步触发信号时，开始对所有选定的使用专用采样保持电路的通道进行模拟输入采样。
- (2) 当采样保持完成后，A/D 开始转换自诊断电压。
- (3) 当自诊断转换完成后，转换结果将存储到自诊断数据寄存器（ADC_SDDATA）中，并且从编号最小的通道开始执行所选通道的模数转换（此例中为ADC0_IN0）
- (4) 每次单个通道的A/D转换完成时，A/D转换结果存储到相应的A/D数据寄存器中。
- (5) 当所有选定通道的A/D转换完成时，ADC_EOCCTL寄存器中的EOC1F位被置为1。当（EOC1RCNT[2:0] + 1）轮Group_pri1转换结束时，EOC1RF标志将被置为1，如果ADC_EOCCTL中的EOC1RIE位为1，则生成中断请求。

- (6) 在A/D转换期间，ADC_CTL0中的PROC位保持为1（A/D转换启动），当所有选定通道的A/D转换完成后，PROC位自动清零为0，然后A/D转换器进入等待状态。

图 15-7. ADC0 单次扫描模式中的通道选择和自诊断（使用通道专用采样保持电路，禁用恒采样模式）



通道选择和自诊断（使用通道专用采样保持电路，使能恒采样模式）

将ADC_CTL0寄存器中的SDSEN位设置为1以使能自诊断。

当使用通道，自诊断和通道专用采样保持电路都选择时，首先执行采样保持操作，然后12位A/D转换器对自诊断电压执行一次A/D转换，如[图15-8. ADC0单次扫描模式中的通道选择和自诊断（使用通道专用采样保持电路，使能恒采样模式）](#)所示。之后，再对所选通道的模拟输入执行一次A/D转换。在这个例子中，所选通道是ADC0_IN0、ADC0_IN1、ADC0_IN2、ADC0_IN3和ADC0_IN4。

当ADC_CTL0寄存器中的SDSFIX位为1时，将转换一个固定的自诊断电压（由ADC_CTL0寄存器中的SDVOG[1:0]配置）。

当ADC_CTL0寄存器中的SDSFIX位为0时，每次扫描会轮流转换一个自诊断电压：0、 $V_{REFP}/2$ 、 V_{REFP} 。

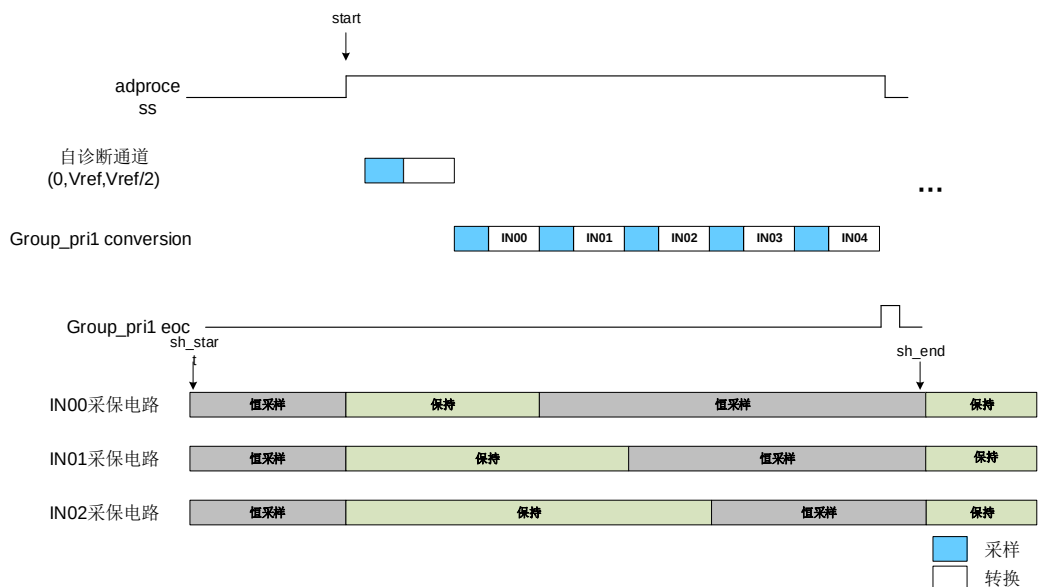
ADC_SHCTL寄存器中的SH00EN、SH01EN、SH02EN位用来选择要使用专用采样保持电路的通道。

将ADC_SHCTL寄存器中的SHMD位置1以使能恒采样模式。

- (1) 将 ADC_SHCTL 寄存器中的 SHSTART 位设置为 1，选定的通道专用采样保持电路开始工作。
- (2) 当软件设置 ADC_EOCCTL 寄存器中的 GP1SWST 位为 1，或者输入同步或异步触发信号时，开始对所有选定的使用专用采样保持电路的通道进行模拟输入采样。

- (3) 当自诊断转换完成后，转换结果将存储到自诊断数据寄存器（ADC_SDDATA）中，并且从编号最小的通道开始执行所选通道的模数转换（此例中为ADC0_IN0）
- (4) 每次单个通道的A/D转换完成时，A/D转换结果存储到相应的A/D数据寄存器中。
- (5) 当所有选定通道的A/D转换完成时，ADC_EOCCTL寄存器中的EOC1F位被置为1。当（EOC1RCNT[2:0] + 1）轮Group_pri1转换结束时，EOC1RF标志将被置为1，如果ADC_EOCCTL中的EOC1RIE位为1，则生成中断请求。
- (6) 在A/D转换期间，ADC_CTL0中的PROC位保持为1（A/D转换启动），当所有选定通道的A/D转换完成后，PROC位自动清零为0，然后A/D转换器进入等待状态。
- (7) 将ADC_SHCTL寄存器中的SHEND位设置为1，以终止选定的通道专用采样保持电路的工作。

图 15-8. ADC0 单次扫描模式中的通道选择和自诊断（使用通道专用采样保持电路，使能恒采样模式）



分叉触发模式下的 ADC 转换

将ADC_BITRGCTL寄存器中的GPxBIMEN位设置为1则使能Group_prix的分叉触发模式。

将需要复制的通道号写入ADC_BITRGCTL寄存器的GPxBICHSEL[4:0]位域，并把ADC_BITRGCTL寄存器中的GPxBIMEN位设置为1，则启用Group_prix的A/D转换数据的复制。当ADC_BITRGCTL寄存器中的GPxBIMEN位被设置为1时，ADC_CHSEL0/1寄存器中选择的通道是无效的。

在Group_prix单次扫描模式下，使用分叉触发模式，由异步/同步触发位或软件启动的单次扫描操作执行两次，如[图15-9. ADC0 Group_pri1在分叉触发模式下的A/D转换](#)所示。

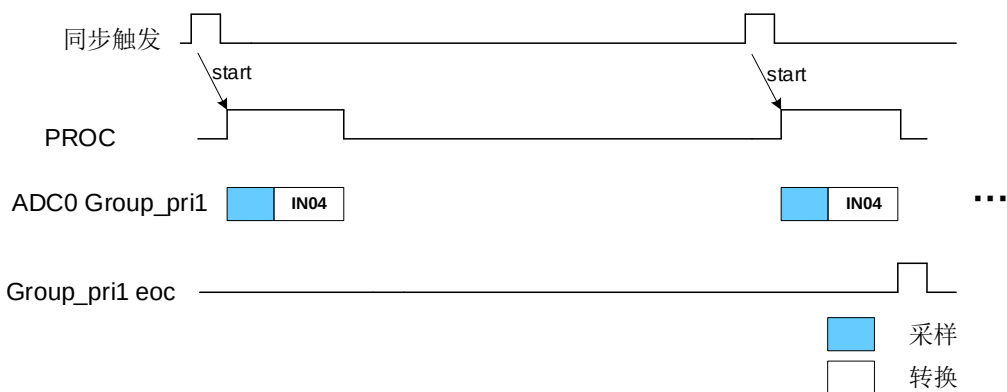
如果使能同步触发，并且GPxTRGSRC[5:0]被配置为6'b000110, 6'b001110, 6'b100101,

6'b101101 or 6'b110011, 那么扩展分叉模式将自动启用。

如果使能了Group_prix的触发模式, 则该组的自诊断功能需要禁能, 该组也不可以选择内部参考电压和温度传感器通道。

- (1) 当同步触发输入时, ADC 将开始转换 ADC_BITRGCTL 寄存器中 GP1BICHSEL[4:0]位域所选择的通道 (本例为 ADC0_IN4), 同时 ADC_CTL0 寄存器中的 PROC 位被设置为 1 (A/D 开始转换)。
- (2) 当 A/D 转换完成时, 转换结果会被存储到相应的通道数据寄存器中。如果运行在扩展分叉模式下, 结果将同时存储到 ADC_GPxBIDATA1 中。ADC_CTL0 寄存器中的 PROC 位自动清除为 0, A/D 转换器进入等待状态。不论 ADC_EOCCTL 寄存器中的 EOC1RIE 位设置 (使能扫描完成中断), 都不会生成扫描结束中断请求。
- (3) 当第二次同步触发输入时, ADC 将开始转换 ADC2_IN4, 同时 ADC_CTL0 寄存器中的 PROC 位被设置为 1 (A/D 开始转换)。
- (4) 当 A/D 转换完成时, A/D 转换结果被存储到 Group_pri x 的 A/D 数据复制寄存器 (ADC_GP1BIDATA) 中, 这个寄存器是分叉触发模式专用。如果运行在扩展分叉模式下, 结果将同时存储在 ADC_GPxBIDATA2 中。
- (5) 如果 ADC_EOCCTL 寄存器中的 EOC1RIE 位为 1 (使能转换结束中断), 则会生成转换结束中断请求 (EOC1RCNT[2:0]=3'b000)。
- (6) ADC_CTL0 寄存器中的 PROC 位在 A/D 转换期间保持为 1, 并在模数转换完成后自动清除为 0。然后 ADC 进入等待状态。

图 15-9. ADC0 Group_pri1 在分叉触发模式下的 A/D 转换



15.5.5. Group_pri1 组连续扫描模式

在下面的例子中, 通道优先级寄存器 ADC_CHPR1x 设置为默认值。

基本运行 (无通道专用采样保持电路)

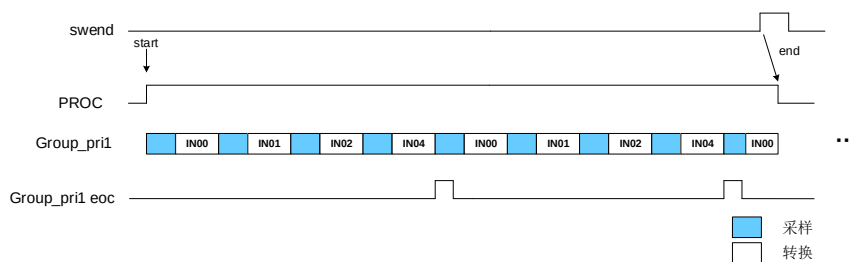
在 Group_pri1 连续扫描模式的基本运行中, 对指定通道的模拟输入进行连续 A/D 转换。例如,

选择ADC0的ADC0_IN0、ADC0_IN1、ADC0_IN2、ADC0_IN4，如[图15-10. ADC0的连续扫描基本运行（无通道专用采样保持电路）](#)所示。

在连续扫描模式下，不要选择内部参考电压和温度传感器通道。

- (1) 当通过软件设置ADC_EOCCTL寄存器中的GP1SWST位为1，或者输入同步或异步触发信号时，将ADC_CHSELx寄存器所选择的这些通道执行A/D转换，从最小编号的通道开始（此例中为ADC0_IN0）。
- (2) 每次单个通道的A/D转换完成时，A/D转换结果存储到相应的A/D数据寄存器中。
- (3) 当所有选定通道的A/D转换完成时，ADC_EOCCTL寄存器中的EOC1F位被置为1。然后，ADC将从编号最小的通道（本例中为ADC0_IN0）开始，连续转换所选择的通道。当（EOC1RCNT[2:0] + 1）轮Group_pri1转换结束时，EOC1RF标志将被置为1，如果ADC_EOCCTL中的EOC1RIE位为1，则生成中断请求。
- (4) PORC位不会自动清0，保持为1，重复步骤2和3。
- (5) 当ADC_CTL0寄存器中的SWEND位被设置为1时，A/D转换停止，A/D转换器进入等待状态。

图 15-10. ADC0 的连续扫描基本运行（无通道专用采样保持电路）



基本运行（使用通道专用采样保持电路，禁用恒采样模式）

当使用通道专用采样保持电路时，首先执行采样保持操作，然后对所有选定通道的模拟输入进行连续A/D转换。采保->模拟输入->...，如此往复。例如，选择ADC0的ADC0_IN0、ADC0_IN1、ADC0_IN2、ADC0_IN4，如[图15-11. ADC0的连续扫描基本运行（使用通道专用采样保持电路，禁用恒采样模式）](#)所示。

ADC_SHCTL寄存器中的SH00EN、SH01EN、SH02EN位用来选择要使用专用采样保持电路的通道。

将ADC_SHCTL寄存器中的SHMD位清零以禁用恒采样模式。

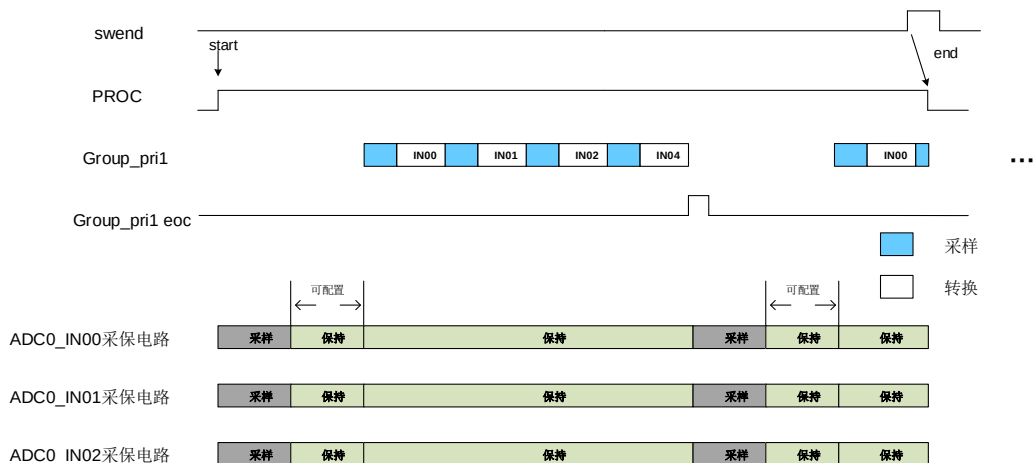
在连续扫描模式下，不要选择内部参考电压和温度传感器通道。

- (1) 当软件设置ADC_EOCCTL寄存器中的GP1SWST位为1，或者输入同步或异步触发信号时，开始对所有选定的使用专用采样保持电路的通道进行模拟输入采样。
- (2) 在采样保持操作之后，对选定的通道进行A/D转换，从编号最小的通道开始（此例中为

ADC0_IN0)。

- (3) 每次单个通道的A/D转换完成时，A/D转换结果存储到相应的A/D数据寄存器中。
- (4) 当所有选定通道的A/D转换完成时，ADC_EOCCTL寄存器中的EOC1F位被置为1。同时，将开始对所有使用通用采样保持电路的通道进行模拟输入采样。当 $(EOC1RCNT[2:0] + 1)$ 轮Group_pri1转换结束时，EOC1RF标志将被置为1，如果ADC_EOCCTL中的EOC1RIE位为1，则生成中断请求。
- (5) PORC位不会自动清0，保持为1，重复步骤2和3。
- (6) 当ADC_CTL0寄存器中的SWEND位被设置为1时，A/D转换停止，A/D转换器进入等待状态。

图 15-11. ADC0 的连续扫描基本运行（使用通道专用采样保持电路，禁用恒采样模式）



基本运行（使用通道专用采样保持电路，使能恒采样模式）

当使用通道专用采样保持电路时，首先执行采样保持操作，然后对所有选定通道的模拟输入进行一次A/D转换。采保->模拟输入->...，如此往复。例如，选择ADC0的ADC0_IN0、ADC0_IN2、ADC0_IN3、ADC0_IN4，如[图15-12. ADC0的连续扫描基本运行（使用通道专用采样保持电路，使能恒采样模式）](#)所示。

ADC_SHCTL寄存器中的SH00EN、SH01EN、SH02EN位用来选择要使用专用采样保持电路的通道。

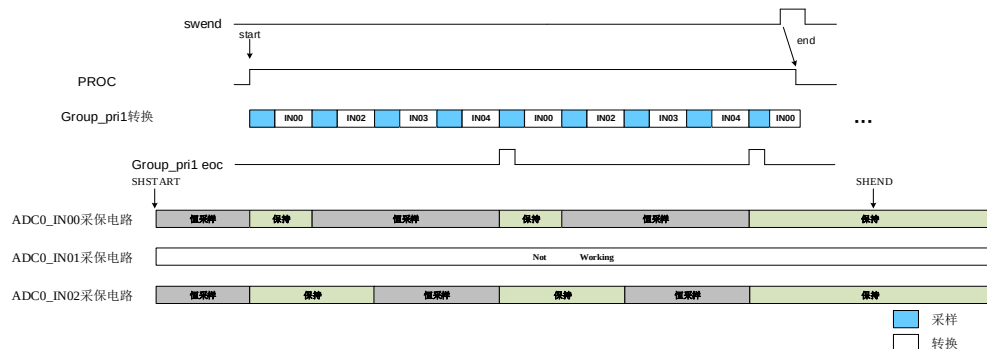
将ADC_SHCTL寄存器中的SHMD位置1以使能恒采样模式。

在连续扫描模式下，不要选择内部参考电压和温度传感器通道。

- (1) 将ADC_SHCTL寄存器中的SHSTART位设置为1，选定的通道专用采样保持电路开始工作。
- (2) 当软件设置ADC_EOCCTL寄存器中的GP1SWST位为1，或者输入同步或异步触发信号时，开始对所有选定的使用专用采样保持电路的通道进行模拟输入采样。
- (3) 对选定的通道进行A/D转换，从编号最小的通道开始（本例中为ADC0_IN0）。每次完成单个通道的A/D转换后，转换的结果将存储在相应的模数数据寄存器（ADC_CHxDATA）中。

- (4) 当所有选定通道的A/D转换完成时，ADC_EOCCTL寄存器中的EOC1F位被置为1。同时，将开始对所有使用专用采样保持电路的通道进行模拟输入采样。当 $(EOC1RCNT[2:0] + 1)$ 轮Group_pri1转换结束时，EOC1RF标志将被置为1，如果ADC_EOCCTL中的EOC1RIE位为1，则生成中断请求。
- (5) PORC位不会自动清0，保持为1，重复步骤2和4。
- (6) 当ADC_CTL0寄存器中的SWEND位被设置为1时，A/D转换停止，A/D转换器进入等待状态。
- (7) 将ADC_SHCTL寄存器中的SHEND位设置为1，以终止选定的通道专用采样保持电路的工作。

图 15-12. ADC0 的连续扫描基本运行（使用通道专用采样保持电路，使能恒采样模式）



通道选择和自诊断（无通道专用采样保持电路）

将ADC_CTL0寄存器中的SDSEN位设置为1以使能自诊断。

当通道和自诊断都选择时，12位A/D转换器先对参考电压执行一次A/D转换，如[图15-13. ADC2 单次扫描模式中的通道选择和自诊断（无专用的采样保持电路）](#)所示。之后，再对所选通道的模拟输入执行一次A/D转换。自诊断->模拟输入->...，如此往复。在这个例子中，所选通道是ADC2_IN0、ADC2_IN1、ADC2_IN3、ADC_IN4和ADC2_IN9。

当ADC_CTL0寄存器中的SDSFIX位为1时，将转换一个固定的自诊断电压（由ADC_CTL0寄存器中的SDVOG[1:0]配置）。

当ADC_CTL0寄存器中的SDSFIX位为0时，每次扫描会轮流转换一个自诊断电压：0、 $V_{REFP}/2$ 、 V_{REFP} 。

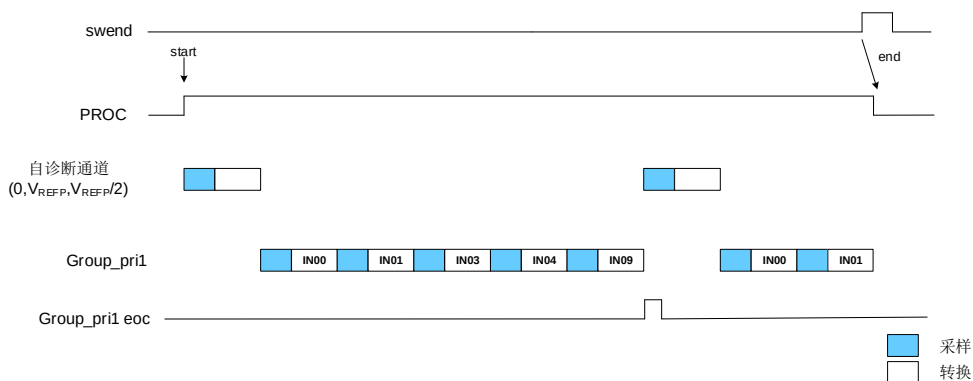
在连续扫描模式下，不要选择内部参考电压和温度传感器通道。

- (1) 当软件设置 ADC_EOCCTL 寄存器中的 GP1SWST 位为 1，或者输入同步或异步触发信号时，开始转换自诊断电压。
- (2) 当自诊断转换完成后，转换结果将存储到自诊断数据寄存器（ADC_SDDATA）中，并且从编号最小的通道开始执行所选通道的模数转换（此例中为ADC2_IN0）
- (3) 每次单个通道的A/D转换完成时，A/D转换结果存储到相应的A/D数据寄存器中。
- (4) 当所有选定通道的A/D转换完成时，ADC_EOCCTL寄存器中的EOC1F位被置为1。同时，

ADC开始进行自诊断A/D转换，然后开始对ADC_CHSEL0/1寄存器选定的通道进行A/D转换，从编号最小的通道开始（本例中为ADC2_IN0）。当 $(EOC1RCNT[2:0] + 1)$ 轮Group_pri1转换结束时，EOC1RF标志将被置为1，如果ADC_EOCCTL中的EOC1RIE位为1，则生成中断请求。

- (5) PORC位不会自动清0，保持为1，重复步骤2和4。
- (6) 当ADC_CTL0寄存器中的SWEND位被设置为1时，A/D转换停止，A/D转换器进入等待状态。

图 15-13. ADC2 单次扫描模式中的通道选择和自诊断（无专用的采样保持电路）



通道选择和自诊断（使用通道专用采样保持电路，禁用恒采样模式）

将ADC_CTL0寄存器中的SDSEN位设置为1以使能自诊断。

当使用通道，自诊断和通道专用采样保持电路都选择时，首先执行采样保持操作，然后12位A/D转换器对自诊断电压执行一次A/D转换，如[图15-14. ADC0单次扫描模式中的通道选择和自诊断（使用通道专用采样保持电路，禁用恒采样模式）](#)所示。之后，再对所选通道的模拟输入执行一次A/D转换。采保->自诊断->模拟输入->...，如此往复。在这个例子中，所选通道是ADC0_IN0、ADC0_IN1、ADC0_IN2、ADC0_IN3和ADC0_IN4。

当ADC_CTL0寄存器中的SDSFIX位为1时，将转换一个固定的自诊断电压（由ADC_CTL0寄存器中的SDVOG[1:0]配置）。

当ADC_CTL0寄存器中的SDSFIX位为0时，每次扫描会轮流转换一个自诊断电压：0、VREFP/2、VREFP。

ADC_SHCTL寄存器中的SH00EN、SH01EN、SH02EN位用来选择要使用专用采样保持电路的通道。

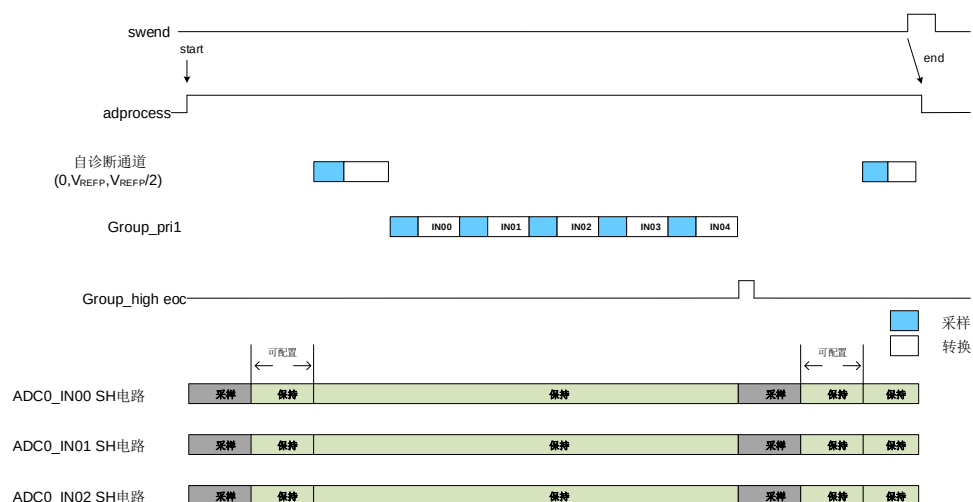
将ADC_SHCTL寄存器中的SHMD位清零以禁用恒采样模式。

在连续扫描模式下，不要选择内部参考电压和温度传感器通道。

- (1) 当软件设置 ADC_EOCCTL 寄存器中的 GP1SWST 位为 1，或者输入同步或异步触发信号时，开始对所有选定的使用专用采样保持电路的通道进行模拟输入采样。

- (2) 当采样保持完成后，A/D 开始转换自诊断电压。
- (3) 当自诊断转换完成后，转换结果将存储到自诊断数据寄存器（ADC_SDDATA）中，并且从编号最小的通道开始执行所选通道的模数转换（此例中为 ADC0_IN0）
- (4) 每次单个通道的A/D转换完成时，A/D转换结果存储到相应的A/D数据寄存器中。
- (5) 当所有选定通道的A/D转换完成时，ADC_EOCCTL寄存器中的EOC1F位被置为1。同时，将开始对所有使用通用采样保持电路的通道进行模拟输入采样。当（EOC1RCNT[2:0] + 1）轮Group_pri1转换结束时，EOC1RF标志将被置为1，如果ADC_EOCCTL中的EOC1RIE位为1，则生成中断请求。
- (6) PORC位不会自动清0，保持为1，重复步骤2和4。
- (7) 当ADC_CTL0寄存器中的SWEND位被设置为1时，A/D转换停止，A/D转换器进入等待状态。

图 15-14. ADC0 单次扫描模式中的通道选择和自诊断（使用通道专用采样保持电路，禁用恒采样模式）



通道选择和自诊断（使用通道专用采样保持电路，禁用恒采样模式）

将ADC_CTL0寄存器中的SDSEN位设置为1以使能自诊断。

当使用通道，自诊断和通道专用采样保持电路都选择时，首先执行采样保持操作，然后12位A/D转换器对自诊断电压执行一次A/D转换，如[图15-15. ADC0连续扫描模式中的通道选择和自诊断（使用通道专用采样保持电路，使能恒采样模式）](#)所示。之后，再对所选通道的模拟输入执行一次A/D转换。采保->自诊断->模拟输入->...，如此往复。在这个例子中，所选通道是ADC0_IN0、ADC0_IN1、ADC0_IN2、ADC0_IN3和ADC0_IN4。

当ADC_CTL0寄存器中的SDSFIX位为1时，将转换一个固定的自诊断电压（由ADC_CTL0寄存器中的SDVOG[1:0]配置）。

当ADC_CTL0寄存器中的SDSFIX位为0时，每次扫描会轮流转换一个自诊断电压：0、 $V_{REFP}/2$ 、 V_{REFP} 。

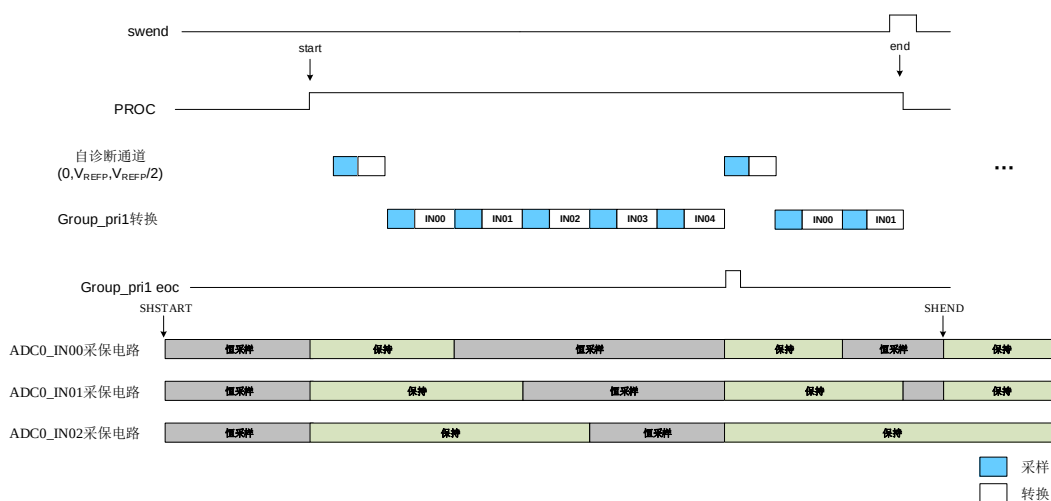
ADC_SHCTL寄存器中的SH00EN、SH01EN、SH02EN位用来选择要使用专用采样保持电路的通道。

将ADC_SHCTL寄存器中的SHMD位置1以使能恒采样模式。

在连续扫描模式下，不要选择内部参考电压和温度传感器通道。

- (1) 将 ADC_SHCTL 寄存器中的 SHSTART 位设置为 1，选定的通道专用采样保持电路开始工作。
- (2) 当软件设置 ADC_EOCCTL 寄存器中的 GP1SWST 位为 1，或者输入同步或异步触发信号时，开始对所有选定的使用专用采样保持电路的通道进行模拟输入采样。
- (3) 当采样保持完成后，A/D 开始转换自诊断电压。当自诊断转换完成后，转换结果将存储到自诊断数据寄存器（ADC_SDDATA）中，并且从编号最小的通道开始执行所选通道的模数转换（此例中为 ADC0_IN0）
- (4) 每次单个通道的A/D转换完成时，A/D转换结果存储到相应的A/D数据寄存器中。
- (5) 当所有选定通道的A/D转换完成时，ADC_EOCCTL寄存器中的EOC1F位被置为1。同时，将开始对所有使用通用采样保持电路的通道进行模拟输入采样。当（EOC1RCNT[2:0] + 1）轮Group_pri1转换结束时，EOC1RF标志将被置为1，如果ADC_EOCCTL中的EOC1RIE位为1，则生成中断请求。
- (6) PORC位不会自动清0，保持为1，重复步骤2和5。
- (7) 当ADC_CTL0寄存器中的SWEND位被设置为1时，A/D转换停止，A/D转换器进入等待状态。
- (8) 将ADC_SHCTL寄存器中的SHEND位设置为1，以终止选定的通道专用采样保持电路的工作。

图 15-15. ADC0 连续扫描模式中的通道选择和自诊断（使用通道专用采样保持电路，使能恒采样模式）



15.5.6. 多组扫描模式

基本运行

在多组扫描模式中，可以选择两组（Group_pri1/pri2）、三组（Group_pri1/pri2/pri3 或 Group_pri1/pri2/pri4）或四组（Group_pri1/pri2/pri3/pri4）作为要使用的组数。

在多组扫描模式基本运行中，一旦通过同步触发启动扫描，就会对Group_pri x中所有指定通道的模拟输入执行一次A/D转换。每个组的扫描运行与单次扫描模式下的扫描运行相似。

注意：不建议将软件触发和异步触发用于多组扫描模式。

ADC_GPCGF0/1中的GPXTRGSRC[5:0]位域用来选择Group_pri x的同步触发源。每个组应该使用不同的同步触发源，以避免同时扫描Group_pri1/pri2/pri3/pri4。

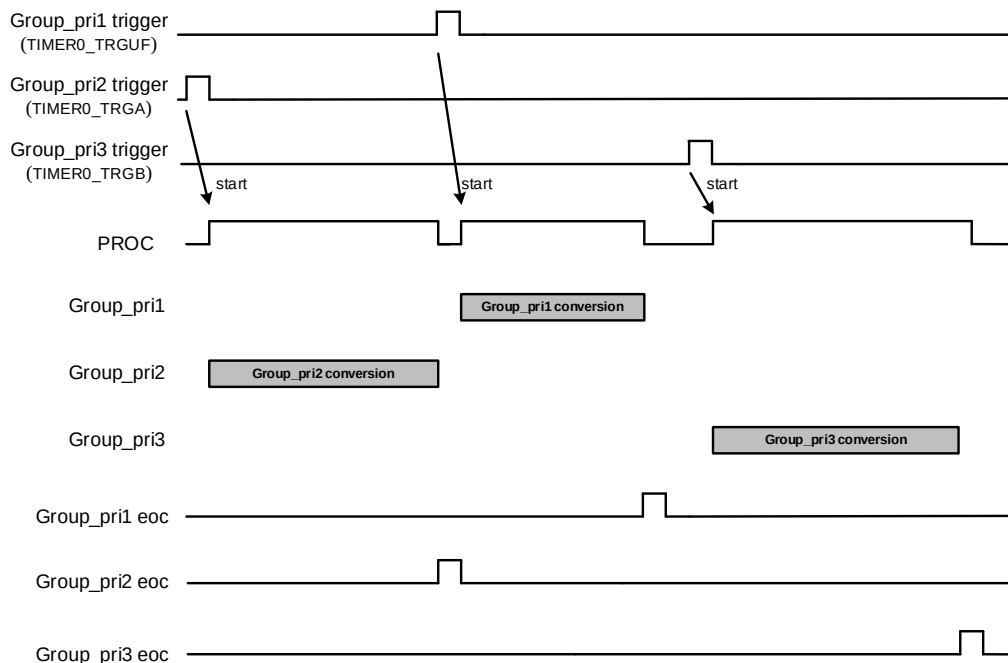
使用ADC_CHSEL0/1寄存器中的位来选择要扫描的通道。

当多组扫描模式中选择自诊断时，将分别对Group_pri x执行自诊断。

[图15-16. 多组扫描模式基本运行](#)展示了多组扫描模式的基本运行，触发源来源于TIMER0。TIMER0_TRGUF、TIMER0_TRGA和TIMER0_TRGB分别用来触发Group_pri1/pri2/pri3组启动转换。

- (1) TIMER0的TIMER0_TRGUF触发启动Group_pri1的扫描。
- (2) 当Group_pri1扫描完成后，ADC_EOCCTL寄存器中的EOC1F位被置为1。当（EOC1RCNT[2:0] + 1）轮Group_pri1转换结束时，EOC1RF标志将被置为1，如果ADC_EOCCTL中的EOC1RIE位为1，则生成中断请求。
- (3) TIMER0的TIMER0_TRGUA触发启动Group_pri2的扫描。
- (4) 当Group_pri2扫描完成后，ADC_EOCCTL寄存器中的EOC2F位被置为1。当（EOC2RCNT[2:0] + 1）轮Group_pri2转换结束时，EOC2RF标志将被置为1，如果ADC_EOCCTL中的EOC2RIE位为1，则生成中断请求。
- (5) TIMER0的TIMER0_TRGUB触发启动Group_pri3的扫描。
- (6) 当Group_pri3扫描完成后，ADC_EOCCTL寄存器中的EOC3F位被置为1。当（EOC3RCNT[2:0] + 1）轮Group_pri3转换结束时，EOC3RF标志将被置为1，如果ADC_EOCCTL中的EOC3RIE位为1，则生成中断请求。

图 15-16. 多组扫描模式基本运行



分叉触发模式下 A/D 转换

将ADC_BITRGCTL寄存器中的GPxBIMEN位设置为1，则使能Group_pri x的分叉触发模式。ADC_BITRGCTL寄存器中的GPxBICHSEL[4:0]位设置Group_pri x的分叉触发通道

当多组扫描模式中选择分叉触发模式时，Group_pri x的执行两轮由同步触发启动的单次扫描。

ADC_GPCGF0/1中的GPXTRGSRC[5:0]位域用来选择Group_pri x的同步触发源。每个组应该使用不同的同步触发源，以避免同时扫描Group_pri1/pri2/pri3/pri4。

注意：不建议将软件触发和异步触发用于多组扫描模式。

如果同步触发被启用，并且GPXTRGSRC[5:0]被配置为6'b000110, 6'b001110, 6'b100101, 6'b101101 or 6'b110011，那么扩展分叉模式将自动启用。

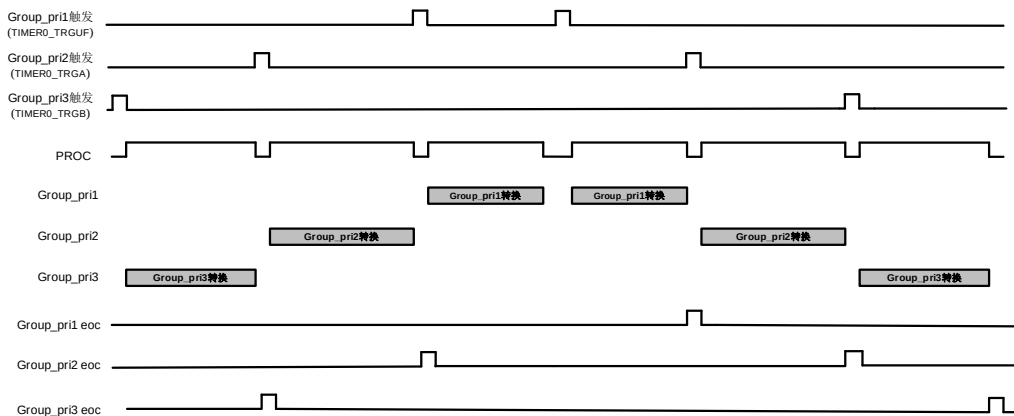
使用ADC_BITRGCTL中的GP4BICHSEL[4:0]位选择Group_pri x组要扫描的通道。

在多组扫描模式中选择分叉触发模式时，不能选择自诊断。

在分叉触发模式下的基本 A/D 转换（无专用采样保持电路）

[图15-17. 在分叉触发模式下的基本A/D转换（无专用采保电路）](#)展示了在Group_pri1中启用分叉触发模式时的运行，使用来自TIMER0的同步触发。在Group_pri2/pri3中没有启用分叉触发模式。假设TIMER0的TIMER0_TRGUF、TIMER0_TRGA和TIMER0_TRGB触发分别用于启动Group_pri1/pri2/pri3的转换。

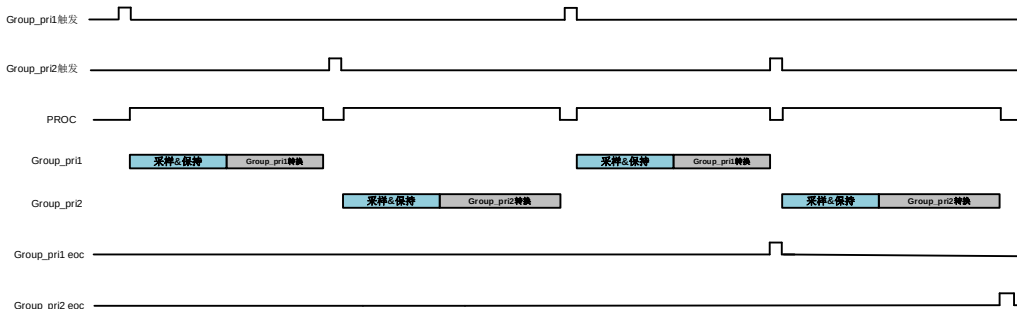
图 15-17. 在分叉触发模式下的基本 A/D 转换（无专用采保电路）



在分叉触发模式下的基本 A/D 转换（使用专用采样保持电路）

[图15-18. 在分叉触发模式下的基本A/D转换（使用专用采保电路）](#)展示了在Group_pri1中启用分叉触发模式时的运行，使用来自TIMER0的同步触发。在Group_pri2/pri3中没有启用分叉触发模式。假设TIMER0的TIMER0_TRGUF、TIMER0_TRGA和TIMER0_TRGB触发分别用于启动Group_pri1/pri2/pri3的转换。

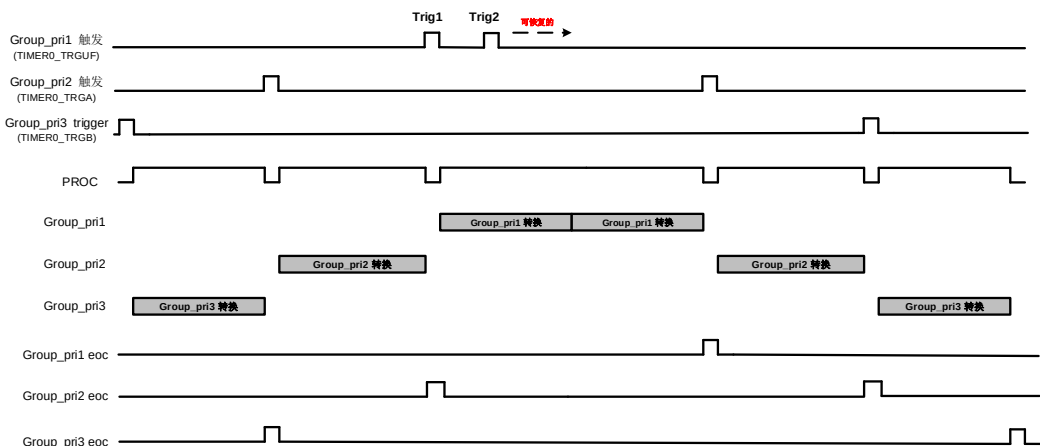
图 15-18. 在分叉触发模式下的基本 A/D 转换（使用专用采保电路）



在分叉触发模式下的基本 A/D 转换和组重启操作（无专用采样保持电路）

[图15-19. 在分叉触发模式下的基本A/D转换和组重启操作（无专用采保电路）](#)展示了在Group_pri1中启用分叉触发模式时的运行，使用来自TIMER0的同步触发。在Group_pri2/pri3中没有启用分叉触发模式。假设TIMER0的TIMER0_TRGUF、TIMER0_TRGA和TIMER0_TRGB触发分别用于启动Group_pri1/pri2/pri3的转换。对于Group_pri1，第二个触发在由第一个触发启动的第一轮A/D转换期间发生。如果ADC_BITRGCTL寄存器中的GP1BITGRS位设置为1，则第二个触发将被恢复，并且在第一轮A/D转换结束时将重新启动第二轮A/D转换（否则将忽略第二个触发）。

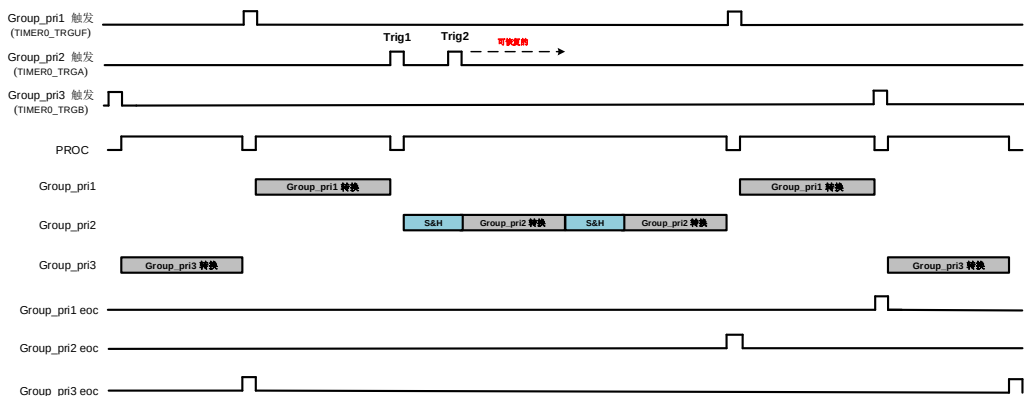
图 15-19. 在分叉触发模式下的基本 A/D 转换和组重启操作（无专用采保电路）



在分叉触发模式下的基本 A/D 转换和组重启操作（使用专用采样保持电路）

[图15-20. 在分叉触发模式下的基本A/D转换和组重启操作（无专用采保电路）](#)展示了在Group_pri1中启用分叉触发模式时的运行，使用来自TIMER0的同步触发。在Group_pri2/pri3中没有启用分叉触发模式。假设TIMER0的TIMER0_TRGUF、TIMER0_TRGA和TIMER0_TRGB触发分别用于启动Group_pri1/pri2/pri3的转换。Group_pri2使用通道专用采样保持电路。对于Group_pri2，第二个触发在由第一个触发启动的第一轮采样和保持操作期间到来。如果ADC_BITRGCTL寄存器中的GP2BITGRS位设置为1，则第二个触发将被恢复，第二轮A/D转换将在第一轮A/D转换结束时重新启动（否则将忽略第二个触发）。

图 15-20. 在分叉触发模式下的基本 A/D 转换和组重启操作（无专用采保电路）

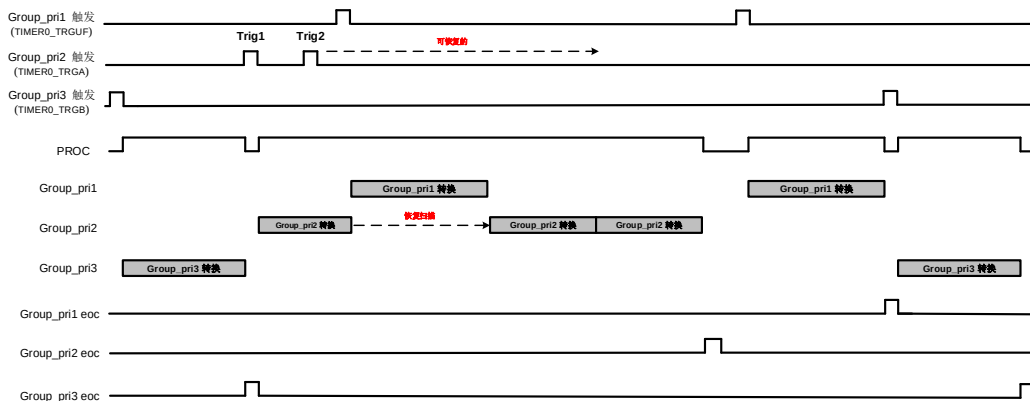


在分叉触发模式下的组重启操作和组优先级控制（无专用采样保持电路）

[图15-21. 在分叉触发模式下的组重启操作和组优先级控制（无专用采保电路）](#)展示了在Group_pri1/pri2中启用分叉触发模式时的运行，使用来自TIMER0的同步触发。在Group_pri3中没有启用分叉触发模式。假设TIMER0的TIMER0_TRGUF、TIMER0_TRGA和TIMER0_TRGB触发分别用于启动Group_pri1/pri2/pri3的转换。在Group_pri2的第一轮A/D转换完成之前，Group_pri1的触发到来并中断了Group_pri2的第一轮A/D转换。在Group_pri1的A/D转换完成后，Group_pri2的第一轮A/D转换开始恢复。如果ADC_BITRGCTL寄存器中的GP2BITGRS位设置为1，则第二个触发将被恢复，并且第二轮A/D转换将在Group_pri2的第一轮A/D转换结束时重新启动（否则将忽略Group_pri2的第二个触发）。

注意：如果禁用了组重启功能，当Group_pri1的A/D转换完成后，Group_pri2的第一轮A/D转换将不会恢复。Group_pri2的第二次触发将等同于第一轮A/D转换的触发。

图 15-21. 在分叉触发模式下的组重启操作和组优先级控制（无专用采保电路）

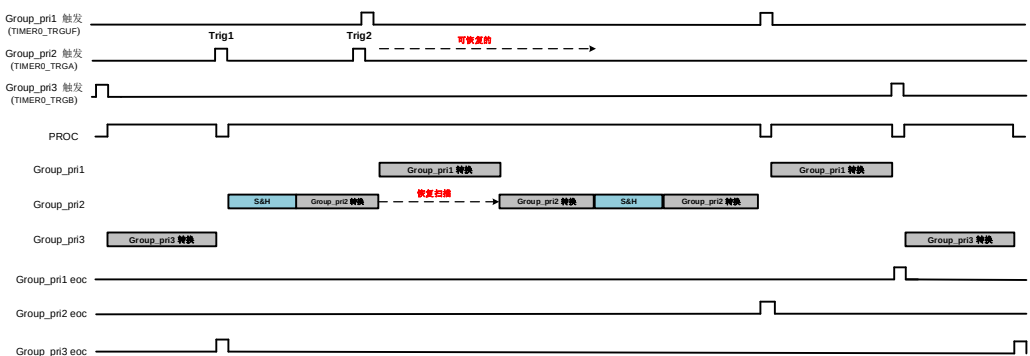


在分叉触发模式下的组重启操作和组优先级控制（无专用采样保持电路）

图15-22. 在分叉触发模式下的组重启操作和组优先级控制（使用专用采保电路）展示了在Group_pri1/pri2中启用分叉触发模式时的运行，使用来自TIMER0的同步触发。在Group_pri3中没有启用分叉触发模式。假设TIMER0的TIMER0_TRGUF、TIMER0_TRGA和TIMER0_TRGB触发分别用于启动Group_pri1/pri2/pri3的转换。Group_pri2使用通道专用采保电路。对于Group_pri2，第二个触发生在由第一个触发启动的第一轮A/D转换期间。在Group_pri2的第一轮A/D转换完成之前，Group_pri1的触发到来并中断了Group_pri2的第一轮A/D转换。在Group_pri1的A/D转换完成后，Group_pri2的第一轮A/D转换开始恢复。如果ADC_BITRGCTL寄存器中的GP2BITGRS位设置为1，则第二个触发将被恢复，并且第二轮A/D转换将在Group_pri2的第一轮A/D转换结束时重新启动（否则将忽略Group_pri2的第二个触发）。

注意：如果禁用了组重启功能，当Group_pri1的A/D转换完成后，Group_pri2的第一轮A/D转换将不会恢复。Group_pri2的第二次触发将等同于第一轮A/D转换的触发。

图 15-22. 在分叉触发模式下的组重启操作和组优先级控制（使用专用采保电路）



在组优先级控制下的运行

将ADC_CTL0寄存器中的GPRIEN位设置为1，使能组优先级控制。

组优先级顺序为Group_pri1 > Group_pri2 > Group_pri3 > Group_pri4。可以选择两组（Group_pri1/pri2）、三组（Group_pri1/pri2/pri3 或 Group_pri1/pri2/pri4）或四组

(Group_pri1/pri2/pri3/pri4) 作为多组扫描模式中使用的组数。如果不遵循该规定，则无法保证正确的扫描操作和数据存储。

在基本的组扫描模式 (GPRIEN = 0) 下，如果Group_pri1/pri2/pri3/pri4正在扫描，所有其他触发输入都被忽略。如果在同一个ADC时钟周期内出现2个或更多的触发输入，只有最高优先级的触发会启动相应的组扫描，其他触发输入被忽略。

在组优先级控制 (GPRIEN = 1) 下，如果在低优先级组的A/D转换期间输入了高优先级组触发，低优先级组的A/D转换将被中断，优先级组的A/D转换将继续进行。

对于任何低优先级组，如果其专用重启扫描功能被禁用 (ADC_GPCGF0/1中的GPxRTEN位为0)，则转换器在完成高优先级组的A/D转换后进入低优先级组的等待状态。在扫描期间，低优先级组的触发输入被忽略。

如果启用了低优先级组专用重扫功能 (在ADC_GPCGF0/1寄存器中的GPxRTEN位设置为1)，转换器会在高优先级组的A/D转换后自动重新扫描低优先级组。同时，在高优先级组扫描期间，低优先级组触发输入是有效的，转换器在扫描完高优先级组后会自动重新扫描低优先级组。

当ADC_GPCGF0/1寄存器中的GPxRTEN位为1而GPxRTCH位为0时，转换器会从低优先级组的头部重新开始扫描。当GPxRTCH位为1时，转换器会从扫描中断的通道处重新开始扫描低优先级组。

当使用自诊断时，转换器在自诊断完成后会根据GPxRTCH的设置，从组的开头或被中断的通道处重新开始扫描。

[表 15-3](#)、[表 15-4](#)、[表 15-5](#)和[表 15-6](#)汇总了在低优先级组重启扫描设置下，在扫描期间输入触发器的响应操作。

当ADC_CTL0中的LPGCNT位设置为1时，最低优先级组的扫描运行在第一次触发输入后将自动连续执行。同时，其组重启扫描操作也会自动执行，忽略ADC_GPCGF0/1寄存器中的GPxRTEN位 (最低优先级) 配置。

在多组扫描模式下的触发设置中，使用GP1TRGSRC[5:0]位选择Group_pri1的同步触发，使用GP2TRGSRC[5:0]位选择与Group_pri1不同的Group_pri2的同步触发，使用GP3TRGSRC[5:0]位选择与Group_pri1/2不同的Group_pri3的同步触发。

注意：当将ADC_CTL0寄存器中的LPGCNT位设置为1时，建议禁用最低优先级组的同步触发。

此外，选择扫描目标时，使用ADC_CHSEL0/1寄存器中的GP1xxx位选择Group_pri1的通道，GP2xxx位选择Group_pri2的通道，GP3xxx位选择Group_pri3的通道，GP4xxx位选择Group_pri4的通道。

表 15-3. 根据 GPxRTEN 位的设置对组的扫描操作进行控制 (GPRIEN = 1, GP3EN = 1, GP4EN = 1)

正在扫描的组	触发输入	每个组根据 GPxRTEN 的扫描运行	
		GPxRTEN = 0	GPxRTEN = 1
当Group_pri1正在进行A/D转换时	Group_pri1 的触发输入	触发输入无效。	触发输入无效。

正在扫描的组	触发输入	每个组根据 GPxRTEN 的扫描运行	
		GPxRTEN = 0	GPxRTEN = 1
	Group_pri2 的触发输入	触 发 输 入 无 效 。 (GP2RTEN = 0)	在完成Group_pri1转换后执行Group_pri2的A/D转换。(GP2RTEN = 1)
	Group_pri3 的触发输入	触 发 输 入 无 效 。 (GP3RTEN = 0)	在完成Group_pri1转换后执行Group_pri3的A/D转换。(GP3RTEN = 1)
	Group_pri4 的触发输入	触 发 输 入 无 效 。 (GP4RTEN = 0)	在完成Group_pri1转换后执行Group_pri4的A/D转换。(GP4RTEN = 1)
当Group_pri2正在进行A/D转换时	Group_pri1 的触发输入	终止 Group_pri2 的 A/D 转换，开始Group_pri1 的转换。(GP2RTEN = 0)	终止Group_pri2的A/D转换，开始Group_pri1 的转换。 在完成Group_pri1转换后开始Group_pri2的A/D转换。(GP2RTEN = 1)
	Group_pri2 的触发输入	触 发 输 入 无 效 。 (GP2RTEN = 0)	触发输入无效。(GP2RTEN = 1)
	Group_pri3 的触发输入	触 发 输 入 无 效 。 (GP3RTEN = 0)	在完成Group_pri2转换后执行Group_pri3的A/D转换。(GP3RTEN = 1)
	Group_pri4 的触发输入	触 发 输 入 无 效 。 (GP4RTEN = 0)	在完成Group_pri2转换后执行Group_pri4的A/D转换。(GP4RTEN = 1)
当Group_pri3正在进行A/D转换时	Group_pri1 的触发输入	终止 Group_pri3 的 A/D 转换，开始Group_pri1 的转换。(GP3RTEN = 0)	终止Group_pri3的A/D转换，开始Group_pri1 的转换。 在完成Group_pri1转换后开始Group_pri3的A/D转换。(GP3RTEN = 1)
	Group_pri2 的触发输入	终止 Group_pri3 的 A/D 转换，开始Group_pri2 的转换。(GP3RTEN = 0)	终止Group_pri3的A/D转换，开始Group_pri2 的转换。 在完成Group_pri2转换后开始Group_pri3的A/D转换。(GP3RTEN = 1)
	Group_pri3 的触发输入	触 发 输 入 无 效 。 (GP3RTEN = 0)	触发输入无效。(GP3RTEN = 1)
	Group_pri4 的触发输入	触 发 输 入 无 效 。 (GP4RTEN = 0)	在完成Group_pri3转换后执行Group_pri4的A/D转换。(GP4RTEN = 1)
当Group_pri4正在进行A/D转换时	Group_pri1 的触发输入	终止 Group_pri4 的 A/D 转换，开始Group_pri1 的转换。(GP4RTEN = 0)	终止Group_pri4的A/D转换，开始Group_pri1 的转换。 在完成Group_pri1转换后开始Group_pri4的A/D转换。(GP4RTEN = 1)
	Group_pri2 的触发输入	终止 Group_pri4 的 A/D 转换，开始 Group_pri2 的转换。(GP4RTEN = 0)	终止Group_pri4的A/D转换，开始Group_pri2 的转换。 在完成Group_pri2转换后开始Group_pri4的A/D转换。(GP4RTEN = 1)
	Group_pri3 的触发输入	终止 Group_pri4 的 A/D 转换，开始 Group_pri3 的转换。	终止Group_pri4的A/D转换，开始Group_pri3 的转换。 在完成Group_pri3转换后开始Group_pri4的A/D转换。

正在扫描的组	触发输入	每个组根据 GPxRTEN 的扫描运行	
		GPxRTEN = 0	GPxRTEN = 1
		的转换。(GP4RTEN = 0)	(GP4RTEN = 1)
	Group_pri4 的触发输入	触 发 输 入 无 效 。 (GP4RTEN = 0)	触发输入无效。(GP4RTEN = 1)

表 15-4. Group_pri1/pri2 组优先级运行设置和运行模式 (GPRIEN = 1, GP3EN = 0, GP4EN = 0)

位值	操作
LPGCNT = 0 GP2RTCH = x GP2RTEN = 0	两组的组优先级运行 当输入 Group_pri1 触发时，Group_pri2 扫描完成（不重新启动）。
LPGCNT = 0 GP2RTCH = 0 GP2RTEN = 1	两组的组优先级运行 Group_pri2扫描被中断后，在Group_pri1扫描完成后，Group_pri2扫描从ADC_CHSEL0寄存器中使用GP2xxx位选择的通道头开始重新启动。
LPGCNT = 0 GP2RTCH = 1 GP2RTEN = 1	两组的组优先级操作 Group_pri2扫描被中断后，在Group_pri1扫描完成后，Group_pri2扫描从ADC_CHSEL0寄存器中使用GP2xxx位选择的通道上次中断的位置重新启动。
LPGCNT = 1 GP2RTCH = 0 GP2RTEN = x	两组的组优先级操作 Group_pri2的单次扫描连续开始，无需启动触发输入。Group_pri2扫描被中断后，在Group_pri1扫描完成后，单次扫描从ADC_CHSEL0寄存器中使用GP2xxx位选择的通道头重新启动。
LPGCNT = 1 GP2RTCH = 1 GP2RTEN = x	两组的组优先级操作 Group_pri2的单次扫描连续开始，无需启动触发输入。Group_pri2扫描被中断后，在Group_pri1扫描完成后，单次扫描从ADC_CHSEL0寄存器中使用GP2xxx位选择的通道上次中断的位置重新启动。

表 15-5. Group_pri1/pri2/pri3 组优先级运行设置和运行模式 (GPRIEN = 1, GP3EN = 1, GP4EN = 0)

位值	操作
LPGCNT = x GP2RTCH = x GP2RTEN = 0	三组的组优先级操作 当输入 Group_pri1 触发时，Group_pri2 扫描完成（不重新启动）。

位值	操作
LPGCNT = x GP2RTCH = 0 GP2RTEN = 1	三组的组优先级操作 Group_pri2扫描被中断后，在Group_pri1扫描完成后，Group_pri2扫描从ADC_CHSEL0寄存器中使用GP2xxx位选择的通道头开始重新启动。
LPGCNT = x GP2RTCH = 1 GP2RTEN = 1	三组的组优先级操作 Group_pri2扫描被中断后，在Group_pri1扫描完成后，Group_pri2扫描从ADC_CHSEL0寄存器中使用GP2xxx位选择的通道上次中断的位置重新启动。
LPGCNT = 0 GP3RTCH = x GP3RTEN = 0	三组的组优先级操作 当输入 Group_pri1/pri2 触发时，Group_pri3 扫描完成（不重新启动）。
LPGCNT = 0 GP3RTCH = 0 GP3RTEN = 1	三组的组优先级操作 Group_pri3扫描被中断后，在Group_pri1/pri2扫描完成后，Group_pri3扫描从ADC_CHSEL1寄存器中使用GP3xxx位选择的通道头开始重新启动。
LPGCNT = 0 GP3RTCH = 1 GP3RTEN = 1	三组的组优先级操作 Group_pri3扫描被中断后，在Group_pri1/pri2扫描完成后，Group_pri3扫描从ADC_CHSEL1寄存器中使用GP3xxx位选择的通道上次中断的位置重新启动。
LPGCNT = 1 GP3RTCH = 0 GP3RTEN = x	三组的组优先级操作 Group_pri3的单次扫描连续开始，无需启动触发输入。Group_pri3扫描被中断后，在Group_pri1/pri2扫描完成后，单次扫描从ADC_CHSEL1寄存器中使用GP3xxx位选择的通道头重新启动。
LPGCNT = 1 GP3RTCH = 1 GP3RTEN = x	三组的组优先级操作 Group_pri3的单次扫描连续开始，无需启动触发输入。Group_pri3扫描被中断后，在Group_pri1/pri2扫描完成后，单次扫描从ADC_CHSEL1寄存器中使用GP3xxx位选择的通道上次中断的位置重新启动。

表 15-6. Group_pri1/pri2/pri3/pri4 组优先级运行设置和运行模式 (GPRIEN = 1, GP3EN = 1, GP4EN =1)

位值	操作
LPGCNT = x GP2RTCH = x GP2RTEN = 0	四组的组优先级操作 当输入 Group_pri1 触发时，Group_pri2 扫描完成（不重新启动）。

位值	操作
LPGCNT = x GP2RTCH = 0 GP2RTEN = 1	四组的组优先级操作 Group_pri2扫描被中断后，在Group_pri1扫描完成后，Group_pri2扫描从ADC_CHSEL0寄存器中使用GP2xxx位选择的通道头开始重新启动。
LPGCNT = x GP2RTCH = 1 GP2RTEN = 1	四组的组优先级操作 Group_pri2扫描被中断后，在Group_pri1扫描完成后，Group_pri2扫描从ADC_CHSEL0寄存器中使用GP2xxx位选择的通道上次中断的位置重新启动。
LPGCNT = x GP3RTCH = x GP3RTEN = 0	四组的组优先级操作 当输入 Group_pri1/pri2 触发时，Group_pri3 扫描完成（不重新启动）。
LPGCNT = x GP3RTCH = 0 GP3RTEN = 1	四组的组优先级操作 Group_pri3扫描被中断后，在Group_pri1/pri2扫描完成后，Group_pri3扫描从ADC_CHSEL1寄存器中使用GP3xxx位选择的通道头开始重新启动。
LPGCNT = x GP3RTCH = 1 GP3RTEN = 1	四组的组优先级操作 Group_pri3扫描被中断后，在Group_pri1/pri2扫描完成后，Group_pri3扫描从ADC_CHSEL1寄存器中使用GP3xxx位选择的通道上次中断的位置重新启动。
LPGCNT = 0 GP3RTCH = x GP4RTEN = 0	四组的组优先级操作 当输入Group_pri1/pri2/pri3触发时，Group_pri4扫描完成（不重新启动）。
LPGCNT = 0 GP3RTCH = 0 GP4RTEN = 1	四组的组优先级操作 Group_pri4扫描被中断后，在Group_pri1/pri2/pri3扫描完成后，Group_pri4扫描从ADC_CHSEL1寄存器中使用GP4xxx位选择的通道头开始重新启动。
LPGCNT = 0 GP3RTCH = 1 GP4RTEN = 1	四组的组优先级操作 Group_pri4扫描被中断后，在Group_pri1/pri2/pri3扫描完成后，Group_pri4扫描从ADC_CHSEL1寄存器中使用GP4xxx位选择的通道上次中断的位置重新启动。
LPGCNT = 1 GP3RTCH = 0 GP4RTEN = x	四组的组优先级操作 Group_pri4的单个扫描连续开始，无需启动触发输入。Group_pri4扫描被中断后，在Group_pri1/pri2/pri3扫描完成后，单个扫描从ADC_CHSEL1寄存器中使用GP4xxx位选择的通道头重新启动。

位值	操作
LPGCNT = 1 GP3RTCH = 1 GP4RTEN = x	四组的组优先级操作 Group_pri4的单个扫描连续开始，无需启动触发输入。Group_pri4扫描被中断后，在Group_pri1/pri2/pri3扫描完成后，单次扫描从ADC_CHSEL1寄存器中使用GP4xxx位选择的通道上次中断的位置重新启动。

Group_pri1 和 Group_pri2 的组优先级操作（GPRIEN = 1，GP3EN = 0，GP4EN = 0）

以下示例展示了在多组扫描模式下（GP2RTEN = 1，LPGCNT = 0）的组优先级操作，Group_pri1选择ADC0_IN0和ADC0_IN4通道，Group_pri2选择ADC0_IN1、ADC0_IN2和ADC0_IN3通道，如[图15-23. 在Group_pri2扫描期间，Group_pri1触发输入（重启扫描设置为GP2RTCH = 1](#)所示

■ 运行示例1：在Group_pri2扫描期间Group_pri1触发输入，重启扫描设置为GP2RTCH = 1

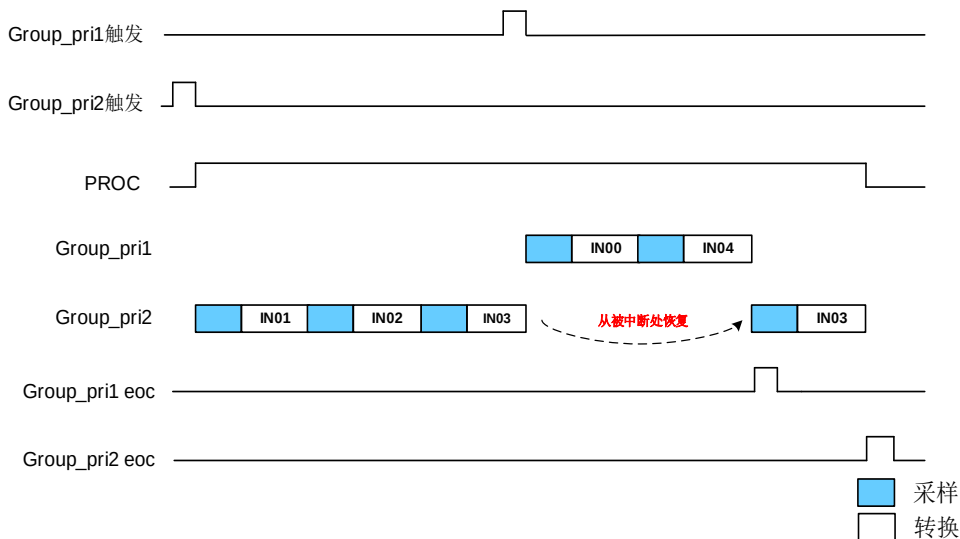
- (1) 当外部触发或软件启动时，PROC位硬件置1（开始A/D转换），扫描ADC_CHSEL0/1寄存器选择的ADC0_INxx通道，从编号最小的通道开始。
- (2) 当一个通道完成A/D转换后，结果存储在相应的A/D数据寄存器（ADC_CHxDATA）中。
- (3) 如果在Group_pri2的扫描过程中输入了Group_pri1触发，PROC位保持为1，Group_pri2的扫描将被中止，并且开始从编号最小的通道扫描Group_pri1中在ADC_CHSEL0/1寄存器中选定的ADC0_INxx通道。

如果扫描中断时A/D转换尚未完成，结果就不会存储在相应的模数数据寄存器中。

- (4) 在一个通道完成模数转换后，结果将存储在相应的模数数据寄存器（ADC_CHxDATA）中。
- (5) 当Group_pri1中所有选定通道的A/D转换完成后，ADC_EOCCTL寄存器中的EOC1F位会被设置为1。每当Group_pri1转换的结束轮数（EOC1RCNT[2:0] + 1）达到时，EOC1RF标志会被设置为1，如果ADC_EOCCTL中的EOC1RIE位为1，则会生成一个中断请求。
- (6) 如果GP2RTCH位是1（由于组优先级操作而中断后，组的扫描重新启动），PROC位保持1，Group_pri2的通道扫描从被中断的通道重新开始。
- (7) 在一个通道A/D转换完成后，结果被存储在相应的A/D数据寄存器（ADC_CHxDATA）中。
- (8) 当Group_pri2中所有选定通道的A/D转换完成后，ADC_EOCCTL寄存器中的EOC2F位将被设置为1。每当Group_pri2转换的结束轮数（EOC2RCNT[2:0] + 1）时，EOC2RF标志将被设置为1，如果ADC_EOCCTL中的EOC2RIE位为1，则会生成一个中断请求。
- (9) 当完成所有选定通道的扫描时，PROC位会自动清零，并且12位A/D转换器进入等待状态。

图 15-23. 在 Group_pri2 扫描期间，Group_pri1 触发输入（重启扫描设置为 GP2RTCH =

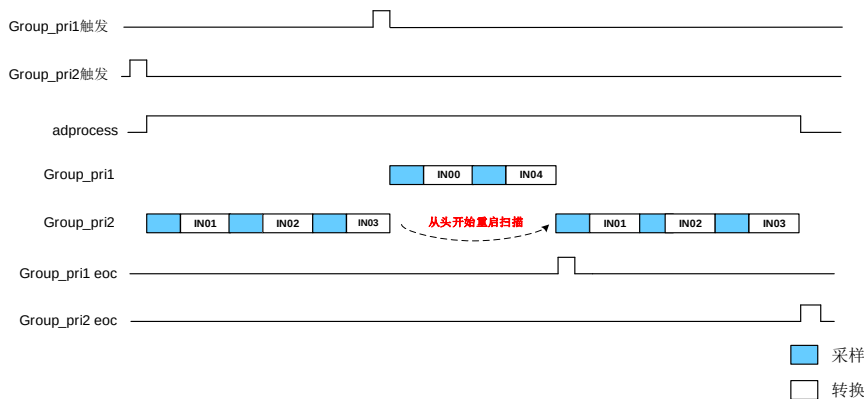
1)



■ **运行示例2: 在Group_pri2扫描期间Group_pri1触发输入, 重启扫描设置为GP2RTCH = 0**

与例1不同, 从Group_pri2的起始通道重新开始扫描, 同时PROC位保持为1, 如 [图15-24. 在 Group_pri2扫描期间, Group_pri1触发输入 \(重启扫描设置为GP2RTCH = 0\)](#) 所示。

图 15-24. 在 Group_pri2 扫描期间, Group_pri1 触发输入 (重启扫描设置为 GP2RTCH = 0)

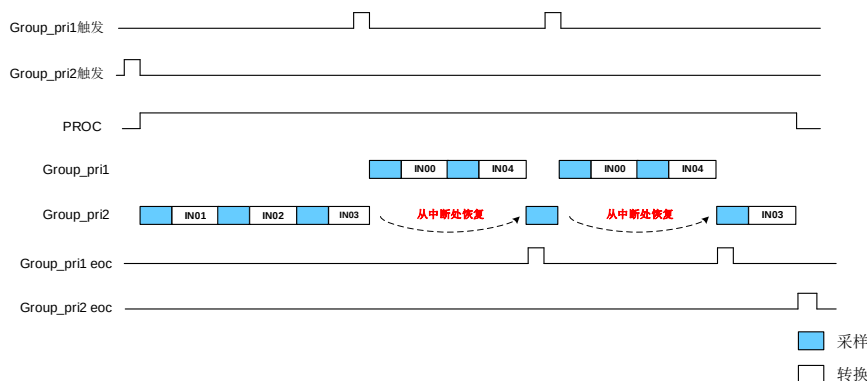


■ **运行示例3: 在Group_pri2重启扫描期间, Group_pri1触发输入, 重启扫描设置为GP2RTCH = 1**

如果接收到Group_pri1触发输入, 即使在重扫操作过程中, 也会开始对Group_pri1进行扫描。Group_pri1扫描完成后, Group_pri2的扫描将从中断的通道开始, 如 [图15-25. 在Group_pri2 重启扫描期间, Group_pri1触发输入 \(重启扫描设置为GP2RTCH= 1\)](#) 所示。

图 15-25. 在 Group_pri2 重启扫描期间, Group_pri1 触发输入 (重启扫描设置为 GP2RTCH=

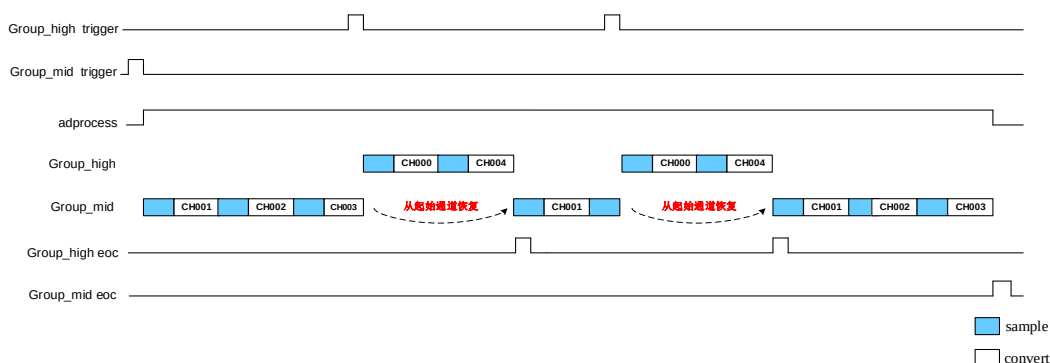
1)



■ 运行示例4：在Group_pri2重启扫描期间，Group_pri1触发输入，重启扫描设置为GP2RTCH = 0

如果收到Group_pri1触发输入，即使在重新扫描操作过程中，也会开始对Group_pri1进行扫描。Group_pri1扫描完成后，Group_pri2的扫描从起始通道开始，如[图15-26. 在Group_pri2重启扫描期间，Group_pri1触发输入（重启扫描设置为GP2RTCH = 0）](#)所示。

图 15-26. 在 Group_pri2 重启扫描期间，Group_pri1 触发输入（重启扫描设置为 GP2RTCH = 0）



■ 运行示例5：在Group_pri1扫描期间Group_pri2触发输入，重启扫描设置为GP2RTEN = 1

以下描述了一个示例，当在Group_pri1的扫描操作期间，Group_pri2触发输入到来时，如果GP2RTEN位为1（由于组优先级操作而中断后，该组的扫描将重新启动）。

如果GP2RTEN位是0，那么在Group_pri1的扫描操作期间输入的所有Group_pri2触发都被禁用。

- (1) 当Group_pri1触发输入时，PROC位设置为1（开始A/D转换），扫描在ADC_CHSEL0/1寄存器中选择的Group_pri1的ADC0_INxx通道，从编号最小的通道开始。
- (2) 如果在扫描Group_pri1期间输入了Group_pri2触发器，可以在Group_pri1扫描完成后开始扫描Group_pri2。
- (3) 在一个通道完成模数转换后，结果存储在相应的数据寄存器（ADC_CHxDATA）中。

(4) 当Group_pri1中所有选定通道的A/D转换完成后，ADC_EOCCTL寄存器中的EOC1F位会被设置为1。每当Group_pri1转换的结束轮数（EOC1RCNT[2:0] + 1）达到时，EOC1RF标志会被设置为1，如果ADC_EOCCTL中的EOC1RIE位为1，则会生成一个中断请求。

(5) 在完成对Group_pri1的扫描后，当PROC位保持为1时，Group_pri2从ADC_CHSEL0/1寄存器选择的ADC2_INxx通道开始扫描，扫描从编号最小的通道开始。

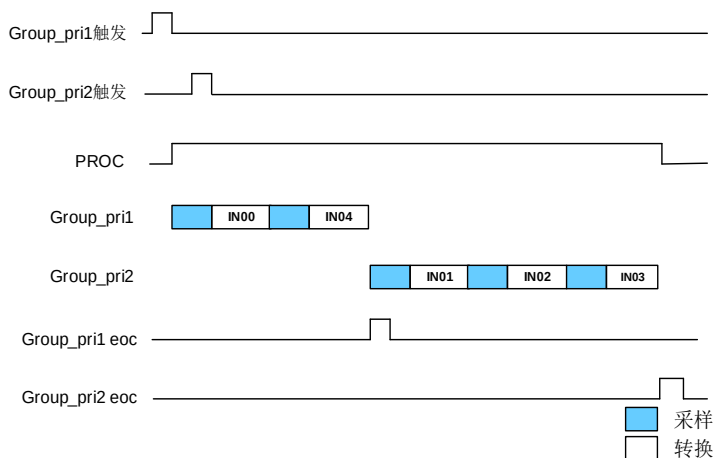
在对Group_pri2进行扫描时，如果输入了Group_pri1触发信号，就像图15-23. 在Group_pri2扫描期间，Group_pri1触发输入（重启扫描设置为GP2RTCH = 1所示，Group_pri1扫描就会开始，Group_pri2扫描会在Group_pri1扫描完成后开始。

(6) 每个通道A/D模数转换后，结果存储在相应的数据寄存器（ADC_CHxDATA）中。

(7) 当Group_pri2中所有选定通道的A/D转换完成后，ADC_EOCCTL寄存器中的EOC2F位将被设置为1。每当Group_pri2转换的结束轮数达到(EOC2RCNT[2:0] + 1)时，EOC2RF标志将被设置为1，如果ADC_EOCCTL中的EOC2RIE位为1，则会生成一个中断请求。

(8) 当所有选定通道的扫描完成后，PROC位会自动清除为0，并且12位A/D转换器进入等待状态。

图 15-27. 在 Group_pri1 扫描期间，Group_pri2 触发输入（重启扫描设置为 GP2RTCH = 1）



■ 运行示例6：在Group_pri2扫描期间Group_pri1触发输入，禁能重启扫描GP2RTEN = 0

(1) 当Group_pri1触发输入时，PROC位设置为1（开始A/D转换），扫描在ADC_CHSEL0/1寄存器中选择的Group_pri1的ADC0_INxx通道，从编号最小的通道开始。

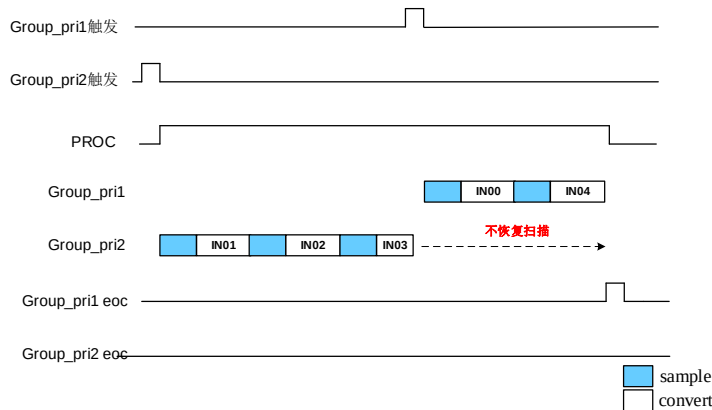
(2) 每个通道完成A/D转换后，结果会存储在相应的数据寄存器（ADC_CHxDATA）中。

(3) 如果在Group_pri2的扫描过程中输入了Group_pri1触发信号，那么Group_pri2的扫描将被中断，同时PROC位保持为1，并且扫描从ADC_CHSEL0/1寄存器中选择的Group_pri1的ADC2_INxx通道，从编号最小的通道开始。

如果扫描中断时A/D转换尚未完成，则结果不会存储在相应的A/D数据寄存器（ADC_CHxDATA）中。

- (4) 在每个通道A/D转换完成后，结果将存储在相应的A/D数据寄存器（ADC_CHxDATA）中。
- (5) 当Group_pri1中所有选定通道的A/D转换完成后，ADC_EOCCTL寄存器中的EOC1F位将被设置为1。每当Group_pri1转换的结束轮数达到(EOC1RCNT[2:0] + 1)时，EOC1RF标志将被设置为1，如果ADC_EOCCTL中的EOC1RIE位为1，则会生成一个中断请求。
- (6) 当扫描Group_pri1完成时，PROC位会自动清零，并进入等待状态。直到下一个Group_pri2触发信号输入，Group_pri2的扫描才会开始。

图 15-28. 在 Group_pri2 扫描期间，Group_pri1 触发输入（禁能重启扫描 GP2RTEN = 1）

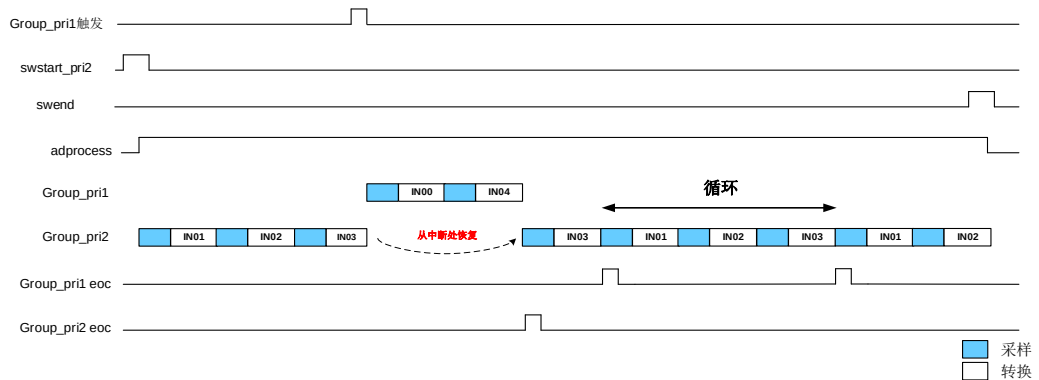


■ 运行示例7：连续单次扫描操作在Group_pri2上(LPGCNT = 1, GP2RTCH = 1)

- (1) 当设置 LPGCNT 位为 1，将 GP2SWST 位置 1（通过软件启动 A/D 转换），将扫描 ADC_CHSEL0/1 寄存器中选定的 Group_pri2 的 ADC0_INxx 通道，从编号最小的通道开始。
- (2) 在单个通道完成模数转换后，结果会存储在相应的数据寄存器（ADC_CHxDATA）中。
- (3) 如果在 Group_pri2 的扫描过程中输入了 Group_pri1 触发信号，那么 Group_pri2 的扫描将被中断，同时 PROC 位保持为 1，并且开始扫描在 ADC_CHSEL0/1 寄存器中选定的 Group_pri1 的 ADC0_INxx 通道，从编号最小的通道开始。如果在扫描中断时 A/D 转换尚未完成，结果将不会被存储在相应的 A/D 数据寄存器中。
- (4) 在单个通道完成模数转换后，结果会存储在对应的数据寄存器（ADC_CHxDATA）中。
- (5) 当 Group_pri1 中所有选定通道的 A/D 转换完成后，ADC_EOCCTL 寄存器中的 EOC1F 位将被设置为 1。每当 Group_pri1 转换的结束轮数达到 (EOC1RCNT[2:0] + 1) 时，EOC1RF 标志将被设置为 1，如果 ADC_EOCCTL 中的 EOC1RIE 位为 1，则会生成一个中断请求。
- (6) 如果 LPGCNT 位设置为 1（最低优先级组连续扫描），则 Group_pri2 的 ADC2_INxx 通道被选择在 ADC_CHSEL0/1 寄存器中时，PROC 位保持为 1，从最小编号的通道重新开始扫描。
- (7) 在单通道完成模数转换后，结果将存储在相应的数据寄存器（ADC_CHxDATA）中。
- (8) 当 Group_pri2 中所有选定通道的 A/D 转换完成后，ADC_EOCCTL 寄存器中的 EOC2F 位将被设置为 1。每当 Group_pri2 转换的结束轮数达到 (EOC2RCNT[2:0] + 1) 结束时，EOC2RF 标志将被设置为 1，如果 ADC_EOCCTL 中的 EOC2RIE 位为 1，则会生成中断请求。
- (9) 如果 LPGCNT 位设置为 1（最低优先级组连续扫描），则对 ADC_CHSEL0/1 寄存器中选定的

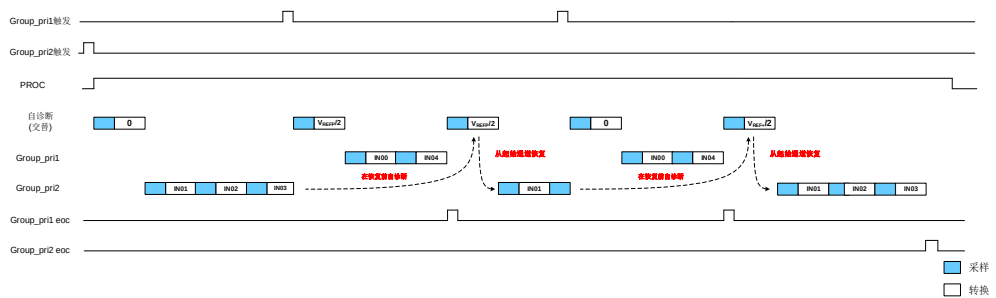
Group_pri2的ADC0_INxx通道进行扫描，从被中断的通道开始，PROC位保持为1。

图 15-29. Group_pri2 连续的单扫描运行(LPGCNT = 1, GP2RTCH = 1)



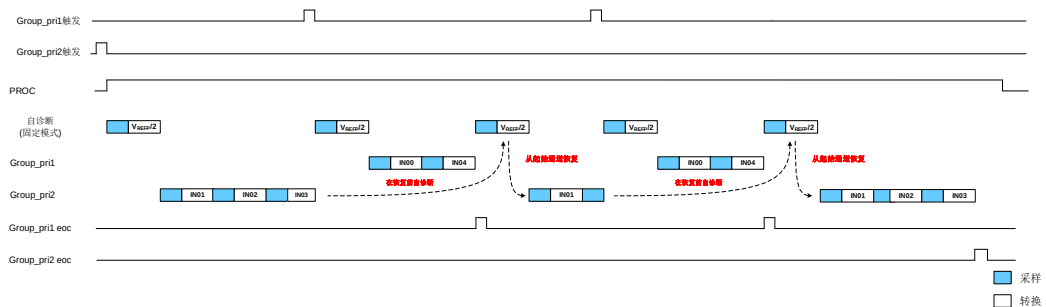
■ 运行示例8：通道选择和交替自诊断（SDSFIX = 0），重启扫描设置为GP2RTCH = 0

图 15-30. 交替自诊断（SDSFIX = 0），重启扫描设置为 GP2RTCH = 0



■ 运行示例9：通道选择和固定V_{REFP/2}自诊断（SDSFIX = 1, SDVOG[1:0] = 2'b10），重启扫描设置为GP2RTCH = 0

图 15-31. 固定自诊断（SDSFIX = 0），重启扫描设置为 GP2RTCH = 0



Group_pri1、Group_pri2 和 Group_pri3 的组优先级运行（GPRIEN = 1, GP3EN = 1, GP4EN = 0）

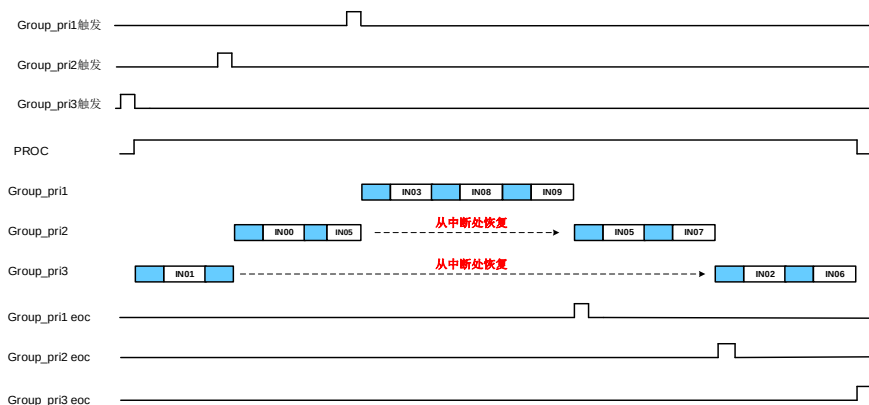
以下示例展示了在多组扫描模式下启用组优先级操作时（GP2RTEN = 1, GP2RTCH = 1, GP3RTEN = 1, GP3RTCH = 1）的情况，当选择ADC2_IN3、ADC2_IN8、ADC2_IN9作为Group_pri1组，选择ADC2_IN0、ADC2_IN5、ADC2_IN7作为Group_pri2组，选择ADC2_IN1、

ADC2_IN2、ADC2_IN6作为Group_pri3组。

■ **运行示例1：在低优先级组扫描期间，高优先级组触发输入，重启扫描设置为GP2RTCH = 1，GP3RTCH = 1**

- (1) 当Group_pri3触发输入，PROC位设置为1（开始A/D转换）时，扫描ADC_CHSEL0/1寄存器中选定的ADC2_INxx通道，从编号最小的通道开始。
- (2) 在单个通道完成A/D转换后，结果将存储在相应的数据寄存器（ADC_CHxDATA）中。
- (3) 如果在Group_pri3的扫描过程中，输入了Group_pri2的触发信号，PROC位保持为1，Group_pri3的扫描会中断，并且会开始扫描在ADC_CHSEL0/1寄存器中选定的Group_pri2的ADC2_INxx通道，从编号最小的通道开始。在完成单个通道的A/D转换后，结果会被存储在对应的数据寄存器（ADC_CHxDATA）中。
- (4) 在单个通道完成A/D转换后，结果将存储在相应的数据寄存器（ADC_CHxDATA）中。
- (5) 如果在Group_pri2的扫描过程中，输入了Group_pri1触发，那么Group_pri2的扫描将被中止，而PROC位保持为1。同时，开始扫描在ADC_CHSEL0/1寄存器中选择的Group_pri1的ADC2_INxx通道，从编号最小的通道开始。如果扫描被中止时A/D转换尚未完成，结果则不会被存储在相应的A/D数据寄存器（ADC_CHxDATA）中。
- (6) 在完成单个通道的A/D转换后，结果会存储在相应的数据寄存器（ADC_CHxDATA）中。
- (7) 当Group_pri1中所有选定通道的A/D转换完成后，ADC_EOCCTL寄存器中的EOC1F位将被设置为1。每当Group_pri1转换的结束轮数达到 $(EOC1RCNT[2:0] + 1)$ 结束时，EOC1RF标志将被设置为1，如果ADC_EOCCTL中的EOC1RIE位为1，则会生成一个中断请求。
- (8) 如果GP2RTEN位是1（由于组优先级操作而中断后，组的扫描重新启动），则恢复ADC_CHSEL0/1寄存器中选择的Group_pri2的ADC2_INxx通道的扫描，从中断A/D转换的通道开始(GP2RTCH=1),PROC位保持为1。
- (9) 在单个通道完成模数转换后，结果将存储在相应的模数数据寄存器(ADC_CHxDATA)中。
- (10) 当Group_pri2中所有选定通道的A/D转换完成后，ADC_EOCCTL寄存器中的EOC2F位将被设置为1。每当Group_pri2转换的结束轮数达到 $(EOC2RCNT[2:0] + 1)$ 时，EOC2RF标志将被设置为1，如果ADC_EOCCTL中的EOC2RIE位为1，则会生成一个中断请求。
- (11) 如果GP3RTEN位是1，则恢复ADC_CHSEL0/1寄存器中选择的Group_pri3的ADC2_INxx通道的扫描，从中断A/D转换的通道开始(GP3RTCH=1)，PROC位保持为1。
- (12) 完成单个通道的模数转换后，结果会存储在相应的数据寄存器（ADC_CHxDATA）中。
- (13) 当所有选定的Group_pri3通道的A/D转换完成后，ADC_EOCCTL寄存器中的EOC3F位将被设置为1。每当Group_pri2转换的结束轮数达到 $(EOC3RCNT[2:0] + 1)$ 时，EOC3RF标志将被设置为1，如果ADC_EOCCTL中的EOC3RIE位为1，则会生成一个中断请求。
- (14) 当对Group_pri1的扫描完成后，ADST位自动清零，并进入等待状态，12位A/D转换器随之进入等待状态。

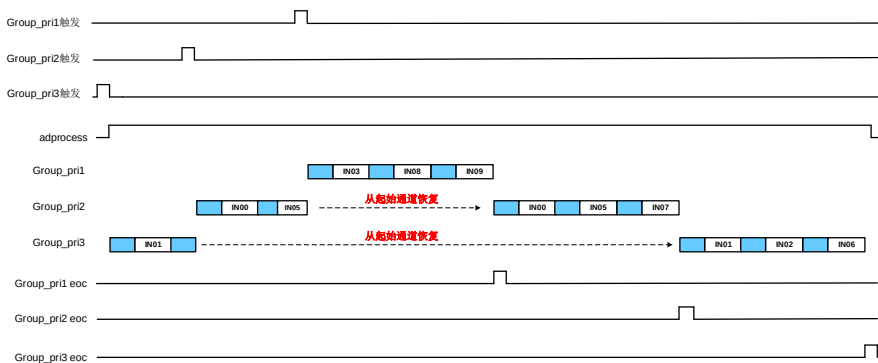
图 15-32. 在低优先级组扫描期间，高优先级组触发输入（GP2RTCH = 1, GP3RTCH = 1）



■ **运行示例2：在低优先级组扫描期间，高优先级组触发输入，重启扫描设置为GP2RTCH = 0, GP3RTCH = 0**

与示例1不同，扫描Group_pri2/3的通道时，就会从起始通道重新开始扫描，PROC位保持为1，如[图15-33. 在低优先级组扫描期间，高优先级组触发输入（GP2RTCH = 0, GP3RTCH = 0）](#)所示。

图 15-33. 在低优先级组扫描期间，高优先级组触发输入（GP2RTCH = 0, GP3RTCH = 0）

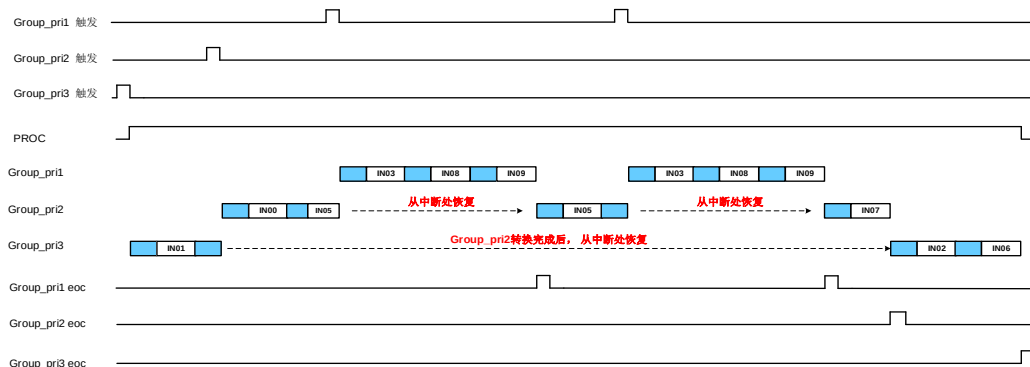


■ **运行示例3：在低优先级组重启扫描期间，高优先级组触发输入，重启扫描设置为GP2RTCH = 1, GP3RTCH = 1**

如果输入了高优先级组的触发（Group_pri1/2优先级高于Group_pri3，Group_pri1优先级高于Group_pri2），即使低优先级组的重扫操作正在进行中，也会开始扫描高优先级组。完成高优先级组的扫描后，被中断的低优先级组扫描将重新开始。

PROC位的操作，A/D转换结果存储到通道数据寄存器（ADC_CHxDATA）的操作，以及中断请求与[图15-32. 在低优先级组扫描期间，高优先级组触发输入（GP2RTCH = 1, GP3RTCH = 1）](#)相同。

图 15-34. 在低优先级组扫描期间，高优先级组触发输入（GP2RTCH = 1, GP3RTCH = 1）



■ 运行示例4：在高优先级组扫描期间，低优先级组触发输入，重启扫描设置为(GPxRTEN = 1)

以下是一些示例：当GPxRTEN位为1时（由于组优先级操作而中断后，组扫描将重新启动），在高优先级组扫描操作期间，低优先级组触发输入。如果GPxRTEN位为0，则在高优先级组扫描操作期间，所有低优先级组输入触发都被禁用。

- (1) 当一个Group_pri1触发输入到来时，开始A/D转换（将PORC位设置为1）。扫描ADC_CHSEL0/1寄存器中选择的Group_pri1的ADC2_INxx通道，从编号最小的通道开始。
- (2) 如果在对Group_pri1进行扫描期间，Group_pri2触发输入，则Group_pri2的触发被挂起。
- (3) 在单个通道完成模数转换后，结果存储在相应的模数数据寄存器（ADC_CHxDATA）中。
- (4) 当Group_pri1中所有选定通道的A/D转换完成后，ADC_EOCCTL寄存器中的EOC1F位将被设置为1。每当Group_pri1转换的结束轮数达到（EOC1RCNT[2:0] + 1）时，EOC1RF标志将被设置为1，如果ADC_EOCCTL中的EOC1RIE位为1，则会生成一个中断请求。
- (5) 在完成对Group_pri1的扫描后，PROC位保持为1，从ADC_CHSEL0/1寄存器中选择的Group_pri2的ADC2_INxx通道的扫描开始，从最小的通道号开始。如果此时GP2RTCH位设置为1，则Group_pri2的扫描将从被中断的A/D转换的通道开始。

当在Group_pri2扫描期间输入Group_pri1触发时，Group_pri1扫描开始，如[图15-32. 在低优先级组扫描期间，高优先级组触发输入（GP2RTCH = 1, GP3RTCH = 1）](#)所示，Group_pri1扫描完成后，Group_pri2扫描开始。

- (6) 如果在扫描Group_pri2期间输入了Group_pri3触发器，则Group_pri3的触发被挂起。
- (7) 在单个通道完成模数转换后，结果将存储在相应的数据寄存器（ADC_CHxDATA）中。
- (8) 当Group_pri2中所有选定通道的A/D转换完成后，ADC_EOCCTL寄存器中的EOC2F位将被设置为1。每当Group_pri2转换的结束轮数达到（EOC2RCNT[2:0] + 1）时，EOC2RF标志将被设置为1，如果ADC_EOCCTL中的EOC2RIE位为1，则会生成一个中断请求。
- (9) 在完成对Group_pri2的扫描后，PROC位保持为1，Group_pri3的ADC2_INxx通道的扫描将从ADC_CHSEL0/1寄存器中选择的具有最小编号的通道开始。如果此时GP3RTCH位被设置为1，那么对Group_pri3的扫描将从被中断的A/D转换的通道开始。

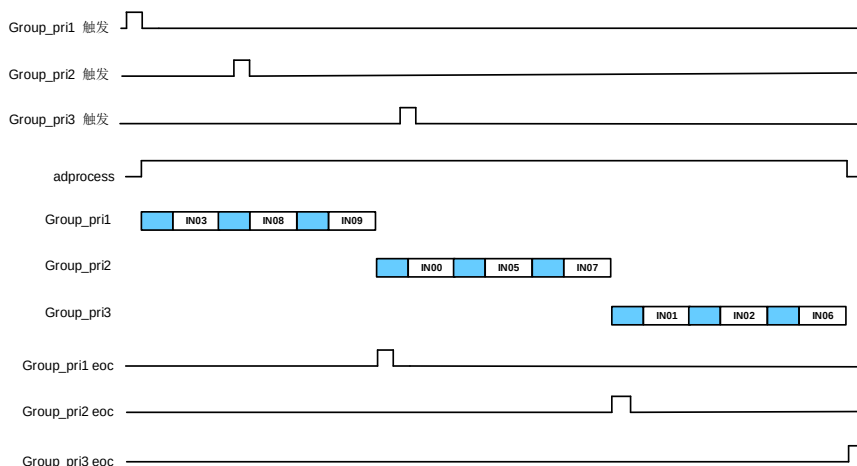
当在对Group_pri3进行扫描时，如果输入了Group_pri1或Group_pri2触发信号，就会像图 15-32. 在低优先级组扫描期间，高优先级组触发输入（GP2RTCH = 1, GP3RTCH = 1）那样开始对Group_pri1或Group_pri2进行扫描，并且在完成Group_pri1或Group_pri2的扫描后，才会开始对Group_pri3的扫描。

(10) 在单个通道完成模数转换后，结果存储在相应的数据寄存器（ADC_CHxDATA）中。

(11) 当Group_pri3中所有选定通道的A/D转换完成后，ADC_EOCCTL寄存器中的EOC3F位会被设置为1。每当Group_pri3转换的结束轮数达到(EOC3RCNT[2:0] + 1)时，EOC3RF标志会被设置为1，如果ADC_EOCCTL中的EOC3RIE位为1，则会生成一个中断请求。

(12) 当所有选定的通道扫描完成后，PROC位会自动清除为0，12位A/D转换器进入等待状态。

图 15-35. 在低优先级组扫描期间，高优先级组触发输入（GPxRTEN = 1）

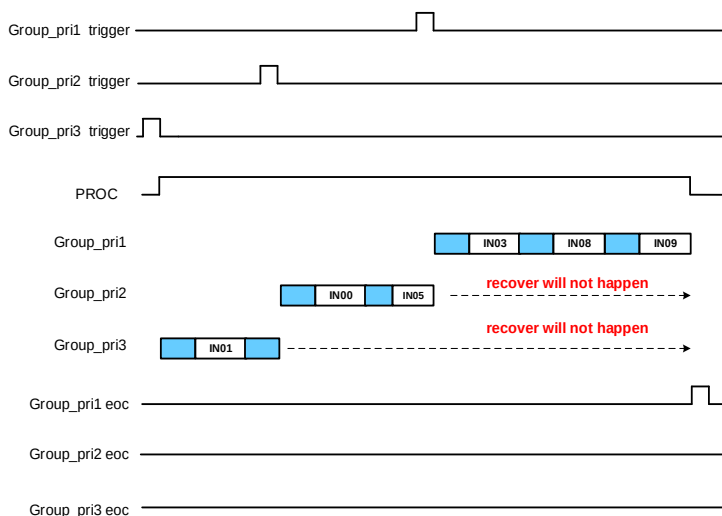


■ 运行示例5：在低优先级组扫描期间，高优先级组触发输入，禁能重启扫描设置为（GPxRTEN = 0）

- (1) 当Group_pri3触发输入来时，开始A/D转换（将PORC位设置为1）。扫描ADC_CHSEL1寄存器中选择的ADC2_INxx通道，从编号最小的通道开始。
- (2) 在单个通道完成A/D转换后，结果会存储在相应的数据寄存器（ADC_CHxDATA）中。
- (3) 如果在Group_pri3的扫描过程中,Group_pri2触发输入，那么Group_pri3的扫描将被中断，同时PROC位保持为1，并且开始扫描在ADC_CHSEL0寄存器中选定的Group_pri2的ADC2_INxx通道，从编号最小的通道开始。如果扫描中断时A/D转换尚未完成，则结果不会存储在相应的A/D数据寄存器（ADC_CHXDATA）中。
- (4) 在完成单个通道的A/D转换后，结果将存储在相应的A/D数据寄存器（ADC_CHXDATA）中。
- (5) 如果在Group_pri2的扫描过程中输入了Group_pri1触发，那么Group_pri2的扫描将被中断，同时PROC位保持为1，并开始从编号最小的通道扫描Group_pri1中在ADC_CHSEL0寄存器中选定的ADC2_INxx通道。如果在扫描中断时A/D转换尚未完成，则结果不会存储在相应的A/D数据寄存器（ADC_CHXDATA）中。
- (6) 当对Group_pri1的扫描完成后，PROC位会自动清零，并且12位A/D转换器进入等待状态。

直到下一个Group_pri2/ Group_pri2对应的触发信号输入，才会开始对其的扫描。

图 15-36. 在低优先级组扫描期间，高优先级组触发输入（GPxRTEN = 0）



■ 运行示例6: Group_pri3连续单扫描运行(GP2RTCH = 1, GP3RTCH = 1)

- (1) 当设置LPGCNT位为1，将GP2SWST位置1（通过软件启动A/D转换），扫描ADC_CHSEL1寄存器中选择的Group_pri3的ADC2_INxx通道，从编号最小的通道开始。
- (2) 在单个通道完成模数转换后，结果被存储在相应的数据寄存器（ADC_CHXDATA）中。
- (3) 如果在Group_pri3的扫描过程中,Group_pri2触发输入，那么Group_pri3的扫描将被中断，同时PROC位保持为1，并且开始扫描在ADC_CHSEL0寄存器中选定的Group_pri2的ADC2_INxx通道，从编号最小的通道开始。如果扫描中断时A/D转换尚未完成，则结果不会存储在相应的A/D数据寄存器（ADC_CHXDATA）中。
- (4) 如果在Group_pri2的扫描过程中输入了Group_pri1触发，那么Group_pri2的扫描将被中断，同时PROC位保持为1，并开始从编号最小的通道扫描Group_pri1中在ADC_CHSEL0寄存器中选定的ADC2_INxx通道。如果扫描中断时A/D转换尚未完成，则结果不会存储在相应的A/D数据寄存器（ADC_CHXDATA）中。
- (5) 单通道完成模数转换后，结果存储在相应的模数数据寄存器（ADC_CHXDATA）中。
- (6) 当Group_pri1中所有选定通道的A/D转换完成后，ADC_EOCCTL寄存器中的EOC1F位将被设置为1。每当Group_pri1转换结束的轮数达到(EOC1RCNT[2:0] + 1)时，EOC1RF标志将被设置为1，如果ADC_EOCCTL中的EOC1RIE位为1，则会生成中断请求。
- (7) 如果GP2RTEN位是1（由于组优先级操作而中断后，组的扫描重新启动），则在ADC_CHSEL0寄存器中选择的Group_pri2的ADC_INxx通道的扫描重新启动，PROC位保持为1，扫描从编号最小的通道开始。如果此时GP2RTCH位设置为1，则Group_pri2的扫描从上次中断的通道开始。
- (8) 在完成单个通道的模数转换后，结果将存储在对应的模数数据寄存器（ADC_CHXDATA）中。
- (9) 当Group_pri2中所有选定通道的A/D转换完成后，ADC_EOCCTL寄存器中的EOC2F位将

被设置为1。每当Group_pri2转换结束的轮数达到（EOC2RCNT[2:0] + 1）时，EOC2RF标志将被设置为1，如果ADC_EOCCTL中的EOC2RIE位为1，则会生成中断请求。

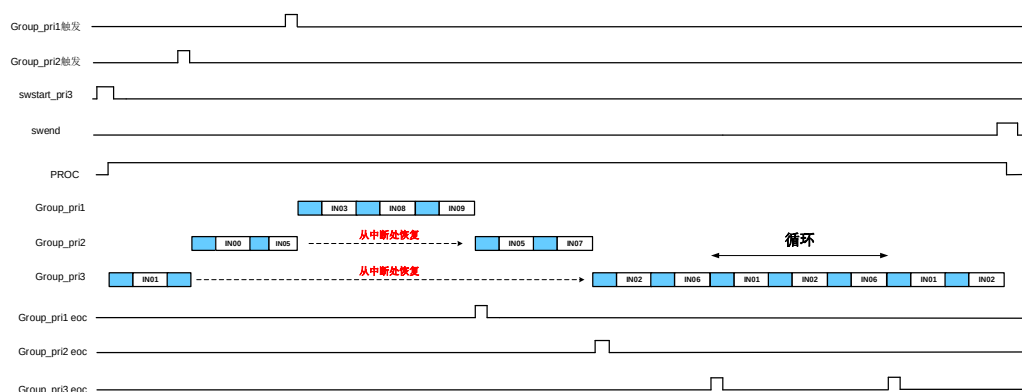
(10)如果LPGCNT位设置为1（最低优先级组连续扫描），PROC位保持1，ADC_INxx通道的Group_pri3选择在ADC_CHSEL1寄存器中，扫描从最小编号的通道重新开始。如果此时GP3RTCH位设置为1，则Group_pri3的扫描从中断A/D转换的通道开始。

(11)在单个通道A/D转换完成后，结果会存储在相应的A/D数据寄存器（ADC_CHXDATA）中。

(12)当Group_pri3中所有选定通道的A/D转换完成后，ADC_EOCCTL寄存器中的EOC3F位将被设置为1。每当Group_pri3转换的结束轮数达到(EOC3RCNT[2:0] + 1)时，这个EOC3RF标志将被设置为1，如果ADC_EOCCTL中的EOC3RIE位为1，将生成一个中断请求。

(13)如果LPGCNT位设置为1（最低优先级组连续扫描），PROC位保持1，从ADC_CHSEL1寄存器中选出的Group_pri3的ADC2_INxx通道进行扫描，并在扫描完成后从编号最小的通道重新开始扫描。

图 15-37. 在 Group_pri3 上进行连续的单次扫描运行(GP2RTCH = 1, GP3RTCH = 1)



15.5.7. 通道优先级配置功能

所有通道的优先级在ADC_CHPRIRx寄存器中配置。每个Group_prix组中，按照所配置的优先级各自转换组内通道。每组的通道由寄存器ADC_CHSELx (x = 0,1)配置。

以ADC2为例：

- ADC_CHPRI0 寄存器配置：PRI0[3:0]=5，PRI1[3:0]=2，PRI2[3:0]=4，PRI3[3:0]=6，PRI4[3:0]=3，PRI4[3:0]=0，PRI4[3:0]=1。优先级ADC2_IN5 > ADC2_IN2 > ADC2_IN4 > ADC2_IN6 > ADC2_IN3 > ADC2_IN0 > ADC2_IN1)，如[表 15-7. ADC2通道优先级设置举例](#)所示。
- ADC_CHSEL0 寄存器中GP1CH0~GP1CH6位配置为1,0,1,0,1,0,0，即选中的通道为ADC2_IN0/2/4。ADC_CHSEL0寄存器中GP2CH0~GP2CH6位配置为0,1,0,1,0,1,1，即选中的通道为ADC2_IN1/3/5/6。

Group_pri1的通道转换顺序是ADC2_IN2, ADC2_IN4, ADC2_IN0.....，而Group_pri2的通道转换顺序是ADC2_IN5, ADC2_IN6, ADC2_IN3, ADC2_IN1.....。

注意：禁止在ADC_CHPRI0/1寄存器中设置相同的通道。

这个功能只能设置模拟通道、内部参考电压通道和温度传感器通道的优先级。自诊断的转换顺序不包括在这个功能中，不能更改。自诊断具有最高优先级。

表 15-7. ADC2 通道优先级设置举例

ADC_CHPRI0设置		通道和优先级		ADC_CHSEL0设置	
位域	值	通道	优先级	Group_pri1	Group_pri2
PRI0[3:0]	5	ADC2_IN5	高 ↓ 低	GP1CH00 = 1	GP2CH00 = 0
PRI1[3:0]	2	ADC2_IN2		GP1CH01 = 0	GP2CH01 = 1
PRI2[3:0]	4	ADC2_IN4		GP1CH02 = 1	GP2CH02 = 0
PRI3[3:0]	6	ADC2_IN6		GP1CH03 = 0	GP2CH03 = 1
PRI4[3:0]	3	ADC2_IN3		GP1CH04 = 1	GP2CH04 = 0
PRI5[3:0]	0	ADC2_IN0		GP1CH05 = 0	GP2CH05 = 1
PRI6[3:0]	1	ADC2_IN1		GP1CH06 = 0	GP2CH06 = 1

图 15-38. ADC2 的 Group_pri1 组实际 A/D 转换顺序

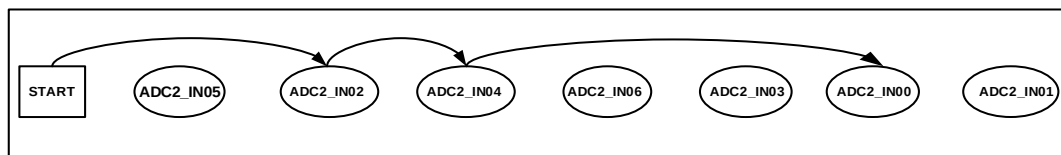
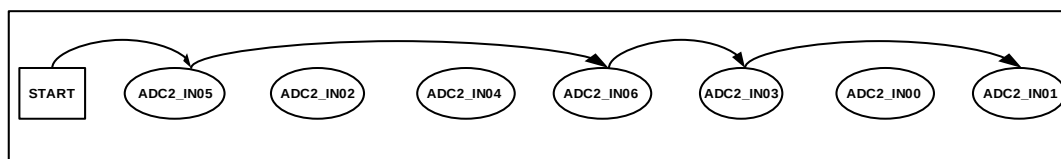


图 15-39. ADC2的Group_pri2组实际A/D转换顺序



15.5.8. 转换结果阈值监测功能

ADCx具有两个模拟看门狗（模拟看门狗A和模拟看门狗B）。这两个看门狗允许在A/D转换过程中监视输入电压，并在输入电压超出预设范围时生成中断或标志。

注意：使用该功能时，不能使用自诊断或分叉触发模式。

模拟看门狗 A

- **启使能看门狗A：**在ADC_WDCTL寄存器中将WDAEN位设置为1可启用看门狗A。
- **操作：**比较功能将ADC_WDATHOLD寄存器中设置的参考值与A/D转换结果进行比较。
- **通道监控：**

对于看门狗A，由ADC_WDACHCFG寄存器中的WDACHx位选择的多个通道被监控。

通道专用匹配条件在ADC_WDACHCFG寄存器中的WDACHxM位中配置。

- **标志和中断：**

如果一个或多个通道的转换数据与通道专用条件设置匹配，ADC_WDACHSTAT寄存器中的通道专用比较状态信号CHxCMPF位将由硬件设置为1。

如果ADC_WDCTL中的WDAIE位为1，则会生成中断请求。

模拟看门狗 B

- **启用看门狗B：** 在ADC_WDCTL寄存器中将WDBEN位设置为1可启用看门狗B。
- **操作：** 比较功能将ADC_WDBTHOLD寄存器中设置的参考值与A/D转换结果进行比较。
- **通道监控：**

对于看门狗B，由ADC_WDCTL寄存器中的WDBCHSEL[3:0]位选择的通道被监控。

看门狗B匹配条件在ADC_WDCTL寄存器中的WDBCM位中配置。

- **标志和中断：**

如果该通道的转换数据与条件设置匹配，ADC_WDCTL寄存器中的WDBEF位将由硬件设置为1。

如果ADC_WDCTL中的WDBIE位为1，则会生成中断请求。

窗口比较功能

还可以使用窗口比较功能（当ADC_WDCTL寄存器中的WINEN为1时），可以比较两个值。

15.5.9. 数据存储模式

当转换加法功能禁能或者选择数据平均模式时

ADC_CTL0寄存器中DAL位指定了转换或者平均后的数据存储方式。数据可以按照MSB或者LSB存储到ADC_SDDATA, ADC_GPxBIDATA, ADC_GPxBIDATA1, ADC_GPxBIDATA2, ADC_GPxDMAR, ADC_CHxDATA, ADC_TEMPDATA和ADC_VINTDATA寄存器中。当选择数据平均模式时，可以选择2,3,4,8,16,32,64次加法。

图 15-40. 12 位数据存储模式

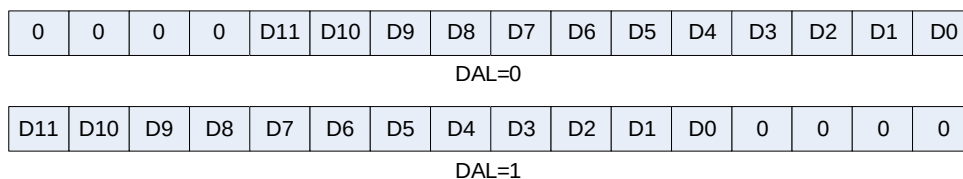


图 15-41. 10 位数据存储模式

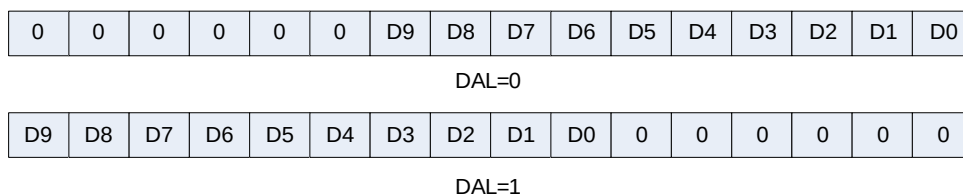


图 15-42. 8 位数据存储模式

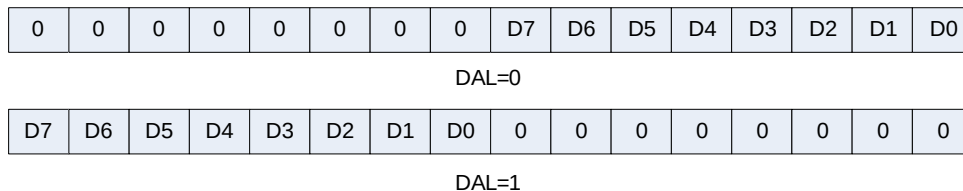
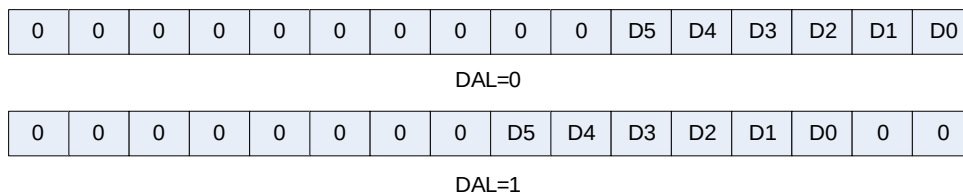


图 15-43. 6 位数据存储模式



自诊断数据寄存器（ADC_SDDATA）分为两部分：自诊断通道转换结果（SDDATA）和自诊断转换的电压状态（SDSTA[1:0]）。

SDSTA[1:0]表示的电压状态：

- 00：从上电开始自诊断未执行过
- 01：自诊断转换0V
- 10：自诊断转换 $V_{REFP}/2$
- 11：自诊断转换 V_{REFP} 对于12/10/8位分辨率：

对于12/10/8位分辨率：

如果DAL=0，SDSTA[1:0]是ADC_SDDATA寄存器的[15:14]位。

如果DAL=1，SDSTA[1:0]是ADC_SDDATA寄存器的[1:0]位。

图 15-44. 自诊断 12 位数据存储模式

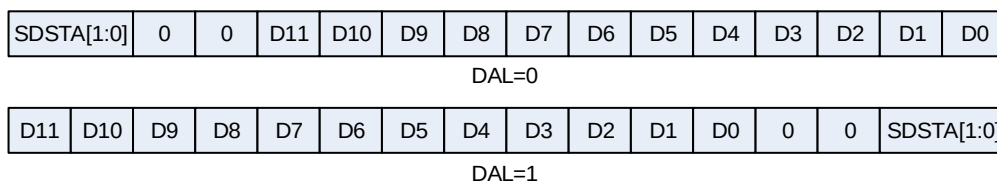


图 15-45. 自诊断 10 位数据存储模式

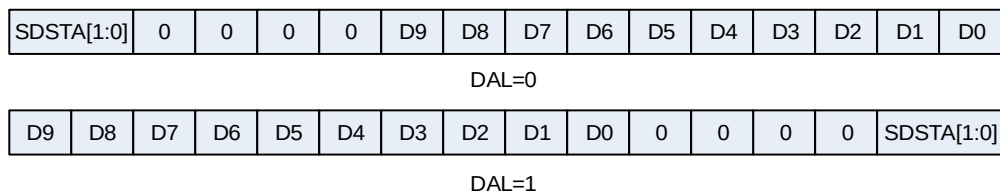
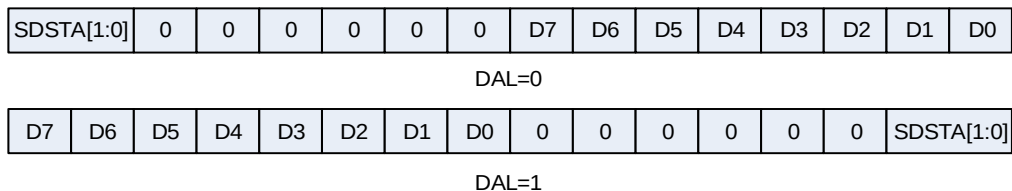


图 15-46. 自诊断 8 位数据存储模式

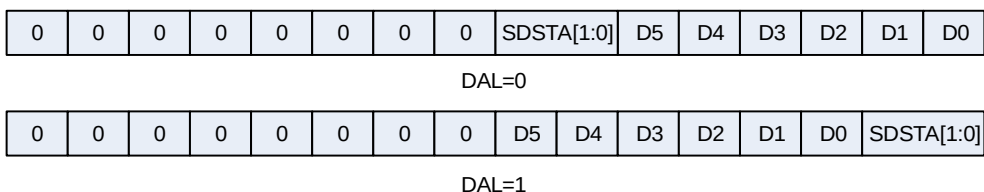


对于6位分辨率:

如果DAL=0, SDSTA[1:0]是ADC_SDDATA寄存器的[7:6]位。

如果DAL=1, SDSTA[1:0]是ADC_SDDATA寄存器的[1:0]位。

图 15-47. 自诊断 6 位数据存储模式



当选择数据加法模式时

ADC_CTL0寄存器中DAL位指定了累加多次后的数据存储方式。数据可以按照MSB或者LSB存储到ADC_SDDATA, ADC_GPxBIDATA, ADC_GPxBIDATA1, ADC_GPxBIDATA2, ADC_GPxDMAR, ADC_CHxDATA, ADC_TEMPDATA和ADC_VINTDATA寄存器中。

12 位数据存储模式

当选择数据加法模式时，可以选择累加2,3,4,8,16次，禁止选择32,64次累加。

(1) 当选择2次时

LSB (DAL=0): 相同通道的AD转换值累加后存储到[12:0]位。

MSB (DAL=1): 相同通道的AD转换值累加后存储到[15:3]位。

(2) 当选择3或者4次时

LSB (DAL=0): 相同通道的AD转换值累加后存储到[13:0]位。

MSB (DAL=1): 相同通道的AD转换值累加后存储到[15:2]位。

(3) 当选择8次时

LSB (DAL=0): 相同通道的AD转换值累加后存储到[14:0]位。

MSB (DAL=1): 相同通道的AD转换值累加后存储到[15:1]位。

(4) 当选择16次时

LSB (DAL=0): 相同通道的AD转换值累加后存储到[15:0]位。

MSB (DAL=1): 相同通道的AD转换值累加后存储到[15:0]位。

10 位数据存储模式

当选择数据加法模式时，可以选择累加2,3,4,8,16,32,64次。

(1) 当选择2次时

LSB (DAL=0): 相同通道的AD转换值累加后存储到[10:0]位。

MSB (DAL=1): 相同通道的AD转换值累加后存储到[15:5]位。

(2) 当选择3或者4次时

LSB (DAL=0): 相同通道的AD转换值累加后存储到[11:0]位。

MSB (DAL=1): 相同通道的AD转换值累加后存储到[15:4]位。

(3) 当选择8次时

LSB (DAL=0): 相同通道的AD转换值累加后存储到[12:0]位。

MSB (DAL=1): 相同通道的AD转换值累加后存储到[15:3]位。

(4) 当选择16次时

LSB (DAL=0): 相同通道的AD转换值累加后存储到[13:0]位。

MSB (DAL=1): 相同通道的AD转换值累加后存储到[15:2]位。

(5) 当选择32次时

LSB (DAL=0): 相同通道的AD转换值累加后存储到[14:0]位。

MSB (DAL=1): 相同通道的AD转换值累加后存储到[15:1]位。

(6) 当选择64次时

LSB (DAL=0): 相同通道的AD转换值累加后存储到[15:0]位。

MSB (DAL=1): 相同通道的AD转换值累加后存储到[15:0]位。

8 位存储模式

当选择数据加法模式时，可以选择累加2,3,4,8,16,32,64次。

(1) 当选择2次时

LSB (DAL=0): 相同通道的AD转换值累加后存储到[8:0]位。

MSB (DAL=1): 相同通道的AD转换值累加后存储到[15:7]位。

(2) 当选择3或者4次时

LSB (DAL=0): 相同通道的AD转换值累加后存储到[9:0]位。

MSB (DAL=1): 相同通道的AD转换值累加后存储到[15:6]位。

(3) 当选择8次时

LSB (DAL=0): 相同通道的AD转换值累加后存储到[10:0]位。

MSB (DAL=1): 相同通道的AD转换值累加后存储到[15:5]位。

(4) 当选择16次时

LSB (DAL=0): 相同通道的AD转换值累加后存储到[11:0]位。

MSB (DAL=1): 相同通道的AD转换值累加后存储到[15:4]位。

(5) 当选择32次时

LSB (DAL=0): 相同通道的AD转换值累加后存储到[12:0]位。

MSB (DAL=1): 相同通道的AD转换值累加后存储到[15:3]位。

(6) 当选择64次时

LSB (DAL=0): 相同通道的AD转换值累加后存储到[13:0]位。

MSB (DAL=1): 相同通道的AD转换值累加后存储到[15:2]位。

6 位存储模式

当选择数据加法模式时，可以选择累加2,3,4,8,16,32,64次。

(1) 当选择2次时

LSB (DAL=0): 相同通道的AD转换值累加后存储到[6:0]位。

MSB (DAL=1): 相同通道的AD转换值累加后存储到[15:9]位。

(2) 当选择3或者4次时

LSB (DAL=0): 相同通道的AD转换值累加后存储到[7:0]位。

MSB (DAL=1): 相同通道的AD转换值累加后存储到[15:8]位。

(3) 当选择8次时

LSB (DAL=0): 相同通道的AD转换值累加后存储到[8:0]位。

MSB (DAL=1): 相同通道的AD转换值累加后存储到[15:7]位。

(4) 当选择16次时

LSB (DAL=0): 相同通道的AD转换值累加后存储到[9:0]位。

MSB (DAL=1): 相同通道的AD转换值累加后存储到[15:6]位。

(5) 当选择32次时

LSB (DAL=0): 相同通道的AD转换值累加后存储到[10:0]位。

MSB (DAL=1): 相同通道的AD转换值累加后存储到[15:5]位。

(6) 当选择64次时

LSB (DAL=0): 相同通道的AD转换值累加后存储到[11:0]位。

MSB (DAL=1): 相同通道的AD转换值累加后存储到[15:4]位。

15.5.10. 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_SAMPTx(x=0..2)寄存器的SPTn[7:0]位更改。每个通道可以用不同的时间采样。在12位分辨率的情况下,总转换时间=采样时间+12.5个CK_ADC周期。

例如:

CK_ADC = 36MHz, 采样时间为2.5个周期, 那么总的转换时间为: “2.5+12.5”个CK_ADC周期, 即0.4167us。

15.5.11. 触发配置

Group_prix组的转换触发请求可以由软件、同步信号和异步信号产生。

软件触发

将ADC_EOCCTL寄存器的GPxSWST位置1, 会产生一个Group_prix组转换触发请求。当任意通道开始转换时, GPxSWST硬件清零。

同步/异步触发

将ADC_GPCGF0/1寄存器的GxTRGEN位置1, 使能同步和异步触发。将ADC_GPCGF0/1寄存器的GPxTRGTY位置1, 则选择异步触发, 反之, 则选择同步触发。如[表 15-8. ADC同步/异步触发](#)所示。

表 15-8. ADC 同步/异步触发使能

GxTRGEN	GPxTRGTY	触发类型
0	-	禁能外部触发
1	0	同步触发
1	1	异步触发

同步或异步触发输入的上升沿、下降沿可以触发Group_prix的转换, 由ADC_GPCGF0/1寄存器的GxTRGEN[1:0]选择触发边沿, 如[表 15-9. ADC触发边沿选择](#)所示。

表 15-9. ADC 触发边沿选择

GxTRGEN[1:0]	触发边沿
00	保留
01	上升沿触发
10	下降沿触发

GxTRGEN[1:0]	触发边沿
11	双边沿触发

ADC_GPCGF0/1寄存器的GPxTRGSRC[5:0]位选择Group_prix组的同步触发源，具体参考[表 15-10. ADC同步触发源](#)。

ADC0/2的异步触发源是固定的（来源于引脚），具体参考[表 15-11. ADC异步触发源](#)。

表 15-10. ADC 同步触发源

GPxTRGSRC[5:0]	触发源	说明
6'b000001	TIMER0_TRGOF	来源于TIMER
6'b000010	TIMER0_TRGUF	
6'b000011	TIMER0_TRGA	
6'b000100	TIMER0_TRGB	
6'b000101	TIMER0_TRGAB	
6'b000110	TIMER0_TRGAORB	
6'b000111	TIMER0_TRGO	
6'b001001	TIMER7_TRGOF	
6'b001010	TIMER7_TRGUF	
6'b001011	TIMER7_TRGA	
6'b001100	TIMER7_TRGB	
6'b001101	TIMER7_TRGAB	
6'b001110	TIMER7_TRGAORB	
6'b001111	TIMER7_TRGO	
6'b010111	TIMER1_TRGO	
6'b011111	TIMER2_TRGO	
6'b100011	GPTIMER0_TRGA	来源于GPTIMER
6'b100100	GPTIMER0_TRGB	
6'b100101	GPTIMER0_TRGAB	
6'b101011	GPTIMER1_TRGA	
6'b101100	GPTIMER1_TRGB	
6'b101101	GPTIMER1_TRGAB	
6'b110001	EVSEL0	来源于EVIC
6'b110010	EVSEL1	
6'b110011	EVSEL0 或 EVSEL1	

表 15-11. ADC 异步触发源

ADCx	触发源	说明
ADC0	ADC0_TRG	来源于 MCU 引脚
ADC2	ADC2_TRG	来源于 MCU 引脚

15.5.12. DMA 功能

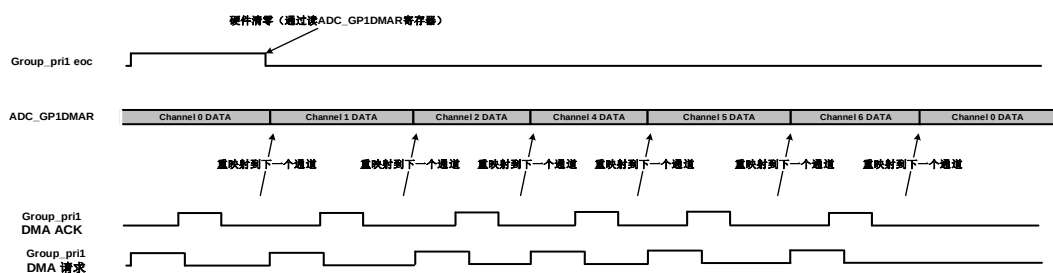
Group_prix的DMA 请求由 ADC_DMACTL 寄存器中的 DMAENx位使能，用于传输多个通道

的转换数据。在Group_pri x转换结束时拉高DMA请求。当Group_pri x DMA请求为高电平时，属于Group_pri x的所有更新的ADC_CHnDATA数据将被传输到用户指定的目标位置。当Group_pri x的DMA确认信号到来时，ADC_GPxDMAR中的数据将重新映射到属于Group_pri x的下一个通道的数据。当Group_pri x的所有更新的ADC_CHnDATA数据被传输到用户指定的目标位置时，ADC则拉低DMA请求。

当在ADC_DMACTL寄存器中设置GPxDDM时，相应的组将在每次DMA传输结束后继续生成DMA请求。建议仅在DMA启用循环模式时将GPxDDM设置为1。

[图 15-48. ADC2 Group_pri1 DMA传输](#)给出了一个ADC2 Group_pri1 DMA传输行为例子，Group_pri2选择通道ADC2_IN0, ADC2_IN1, ADC2_IN2, ADC2_IN4, ADC2_IN5, ADC2_IN6。

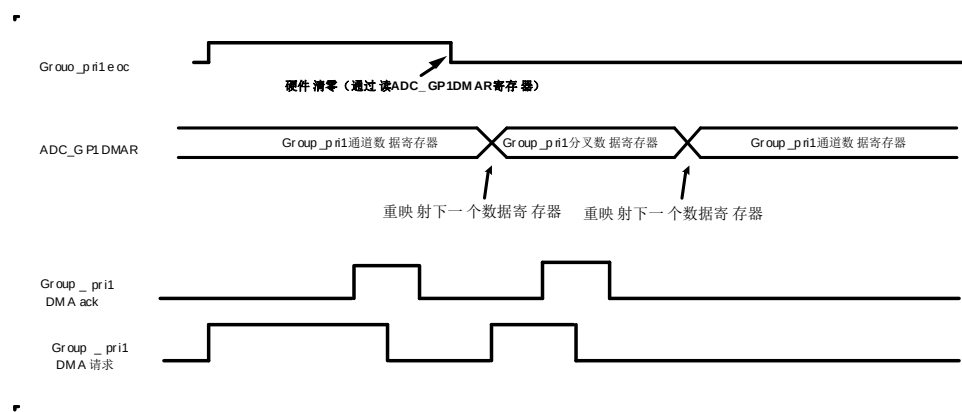
图 15-48. ADC2 Group_pri1 DMA 传输例子



当分叉触发功能用于组（Group_pri x）时，在eoc（转换结束）后，DMA数据寄存器将首先映射到ADC_CHnDATA寄存器（通道由ADC_BITRGCTL寄存器中的GPxBICHSEL[4:0]选择），然后重新映射到ADC_GPxBIDATA，如[图 15-49. ADC2 Group_pri1 DMA传输例子（使能分叉触发功能）](#)所示。

注意：当使用ADC数据寄存器自动清除功能时，在重新映射操作后，ADC_CHnDATA和ADC_GPxBIDATA将相应清除。

图 15-49. ADC2 Group_pri1 DMA 传输例子（使能分叉触发功能）



15.5.13. DMA 溢出检测

溢出检测

当ADC_DMACTL 寄存器中的 DMAENx和 OVRENx 都设置为1时，溢出检测被启用。如果

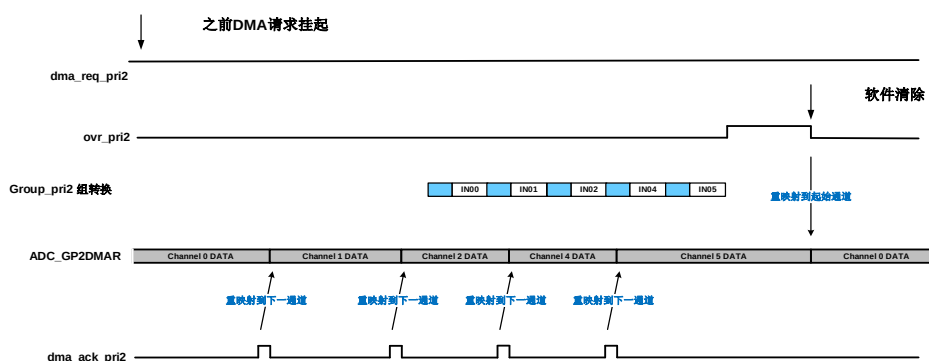
在单个通道的转换完成之前，先前的 ADC_GPxDMAR 中的数据还未传输到用户指定的目标位置，就会发生溢出事件，此时硬件会将 GPxOVRF位设置为1。如果 GPxOVRIE位被设置，将会产生溢出中断请求。

溢出处理

为了使得ADC从ROVF溢出状态中恢复过来，建议对DMA模块重新进行初始化，内部状态机复位，以保证高/中/低组转换数据正确的传输。当检测到溢出，Group_pri x ADC转换将会停止，直到ADC_DMACTL寄存器的GPxOVRF位被软件清零。同时，ADC_GPxDMAR寄存器也会重映射到Group_pri x的起始通道。

[图 15-50. ADC2 Group_pri2溢出事件举例](#)给出了一个ADC2 Group_pri2溢出事件例子，Group_pri2选择通道ADC2_IN0, ADC2_IN1, ADC2_IN2, ADC2_IN4, ADC2_IN5。

图 15-50. ADC2 Group_pri2 溢出事件举例



15.5.14. ADC 内部通道

温度传感器通道

只有ADC2有内部温度传感器通道。

GP1TEMP位在Group_pri1单次扫描模式中选择内部温度传感器通道的A/D转换。

GPxTEMP位在多组扫描模式中选择相应组（Group_pri1/pri2/pri3/pri4）的内部温度传感器通道的A/D转换，优先级可以在ADC_CHPRI0/1中设置。

使用ADC_SAMPR1寄存器中的SPTTEMP[7:0]位设置采样时间。如果使用了辅助断开检测功能（CHRSSEL[4:0]≠5'b00000），在温度传感器通道的A/D转换期间，采样在放电完成后开始，放电周期CHRSSEL[4:0]自动固定为5'b11111。在转换完成以后，CHRSSEL[4:0]自动恢复到之前设置的值。

内部温度传感器可以用来测量设备环境温度。传感器输出电压可通过ADC转换为数字值。建议将温度传感器的采样时间设置为至少 ts_temp μs（具体值请参考数据手册）。

使用步骤：

- (1) 在ADC_CHSEL0/1寄存器中配置GPxTEMP位，并通过ADC_SAMPR1寄存器中的SPTTEMP[7:0]位设置通道的采样时间（ts_temp μs）。

(2) 通过触发器启动ADC转换。

(3) 从ADC_TEMPDATA寄存器中读取温度数据($V_{\text{temperature}}$), 并使用以下公式计算温度:

$$\text{温度 (}^{\circ}\text{C)} = \frac{(V_{\text{temperature}} - V_{-40}) * (-40 - 105)}{V_{-40} - V_{105}} - 40 \quad (15-1)$$

V_{-40} : 内部温度传感器在 -40°C 时的输出电压, 典型值和工厂校准值地址请参考数据手册(TS_CAL1);

V_{105} : 内部温度传感器在 105°C 时的输出电压, 典型值和工厂校准值地址请参考数据手册(TS_CAL2);

内部参考电压通道

只有ADC2有内部参考电压通道。

GP1VINT位在Group_pri1单次扫描模式中选择内部参考电压通道的A/D转换。

GPxVINT位在多组扫描模式中选择相应组(Group_pri1/pri2/pri3/pri4)的内部参考电压通道的A/D转换, 优先级可以在ADC_CHPRI0/1中设置。

使用ADC_SAMPR1寄存器中的SPTVINT[7:0]位设置采样时间。如果使用了辅助断开检测功能(CHRSEL[4:0]≠5'b00000), 在内部参考电压通道的A/D转换期间, 采样在放电完成后开始, 放电周期CHRSEL[4:0]自动固定为5'b11111。在转换完成以后, CHRSEL[4:0]自动恢复到之前设置的值。

采样时间应为5μs或更长。

使用步骤:

- (1) 将采样时间设置为5μs或更长。
- (2) 切换到内部参考电压通道的A/D转换后, 启动该通道的A/D转换。
- (3) A/D转换完成后, 转换结果存储在A/D内部参考电压通道数据寄存器(ADC_VINTDATA)中。
- (4) 在A/D转换期间, PROC位保持为1, 转换完成后自动清零为0。随后, 12位A/D转换器进入等待状态。

15.5.15. 可编程分辨率(DRES)

对寄存器ADC_CTL1中的DRES[1:0]位进行编程即可配置分辨率为6、8、10、12位。对于那些不需要高精度数据的应用, 可以使用较低的分辨率来实现更快速地转换。只有在ADCON比特为0时, 才能修改DRES[1:0]的值。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主次逼近转换时间, 如[表 15-12. 不同分辨率对应的t_{CONV}时间](#)。

表 15-12. 不同分辨率对应的 t_{CONV} 时间

DRES[1:0]	t _{CONV}	t _{CONV} (ns) at	t _{SMPL} (min)	t _{ADC}	t _{ADC} (ns) at

bits	(ADC clock cycles)	$f_{ADC}=36\text{MHz}$	(ADC clock cycles)	(ADC clock cycles)	$f_{ADC}=36\text{MHz}$
12	12.5	347.22 ns	2.5	15	416.67 ns
10	10.5	291.67 ns	2.5	13	361.11 ns
8	8.5	236.11 ns	2.5	11	305.56 ns
6	6.5	180.56 ns	2.5	9	250.00 ns

15.5.16. 断开辅助检测功能（预充放电）

该转换器集成了在A/D转换开始前将采样电容的电荷固定到指定状态的功能。该功能使能了模拟输入、温度传感器通道和内部参考电压通道接线中的断开检测。如果使用断开检测辅助功能，则对于不使用采样保持电路的每个单通道转换，都会进行断开检测操作。当ADC_SHCTL寄存器中的SH00EN、SH01EN或者SH02EN为0（旁路采样保持电路）时，对应通道可以使用断开检测辅助功能。

ADC_CTL0寄存器中的CHRSEL[4:0]位选择预充电或放电，以及用于A/D断开检测辅助功能的预充电/放电周期。CHRSEL[4]位设置为1表示选择预充电，设置为0表示选择放电。使用CHRSEL[3:0]位设置预充电/放电的周期。当CHRSEL[3:0] = 4'b0000或4'b0001时，断开检测辅助功能无效（断开检测被禁用）。除CHRSEL[3:0] = 4'b 0000或4'b 0001的情况外，指定值表示预充电/放电周期的状态数量。当PROC位为0时，应设置CHRSEL[4:0]位。

对于内部参考电压通道和温度传感器通道，在采样通道之前，会插入一个固定的断开检测周期（CHRSEL[3:0] = 0xF）。

只有ADC2有内部参考电压通道和温度传感器通道。

图 15-51. 当辅助断开检测功能使用时，AD 转换的运行



15.5.17. 自诊断

ADC_CTL0寄存器中的SDSEN位用于使能或禁用自诊断功能。

自诊断用于检测12位A/D转换器的故障。具体而言，将转换内部生成的三个电压值之一：0、 $V_{REFP}/2$ 和 V_{REFP} 。当转换完成时，转换的电压信息和转换结果将存储到自诊断数据寄存器(ADC_SDDATA)中。软件可以读取ADC_SDDATA 以确定转换结果是否在正常范围内（正常）或不在正常范围内（异常）。自诊断在每次扫描开始时执行一次，并转换三个电压中的一个。当在多组扫描模式下选择自诊断时，自诊断将在Group_pri1/Group_pri2/Group_pri3/Group_pri4中分别执行。

ADC_CTL0寄存器中的SDSFIX位用于选择是轮流使用三个电压值还是在自诊断中使用固定电压。将SDSFIX位设置为0允许转换模式使用轮流模式，即按顺序转换0、 $V_{REFP}/2$ 和 V_{REFP} 。当自诊断在复位后选择轮流模式时，自诊断从0V开始执行。将SDSFIX位设置为0以启用自诊断电压固定模式，固定电压由SDVOG[1:0]位指定。在自诊断电压轮换模式中，当扫描转换完成时，

自诊断电压值不会返回到0V。因此，当扫描转换重新启动时，轮换从前一个值的下一个电压值开始。如果从固定模式切换到轮换模式，轮换将从固定电压值开始。

SDSEN、SDSFIX、SDVOG[1:0] 位应在PROC位为0时设置。

15.5.18. 采样保持功能（仅 ADC0）

ADC_SHCTL寄存器中的SH00EN/SH01EN/SH02EN位用于选择是否使用或不使用（旁路）ADC0_IN0到ADC0_IN2通道专用采样保持电路。

ADC_SHCTL寄存器中的SHSTIME[7:0]位设置通道专用采样保持电路的采样时间。如果ADCLK时钟为 32MHz，一个状态为31.25 ns（一个状态为一个ADCLK周期）。

在Group_pri1单次扫描模式和连续扫描模式中，如果不使用恒定采样功能，采样初始值为12个状态，保持初始值为8个状态。

如果模拟输入信号源的阻抗过高或者ADCLK时钟较慢，可以调整采样时间，以确保足够的采样时间。采样时间必须设置为4个状态或更多，并且不超过255。此外，采样状态设定值应至少符合规定的测试条件。

ADC_SHCTL寄存器中的SHSTIME[7:0]/SHHTIME[3:0] 位应在PROC位为1之前设置。

在恒定采样模式下，电路工作运行显示在[Group_pri1组单次扫描模式](#)和[Group_pri1组连续扫描模式](#)。在多组扫描模式下，所有使能采样保持的通道可以作为一个整体为Group_pri x选择。通道专用采样保持电路独立工作。如果选择了最后一个具有通道专用采样保持电路的通道并执行连续扫描，则在连续扫描的第二轮及以后将无法确保正常采样的时间。在这种情况下，建议在该通道的末尾至少添加一个其他通道。

在多组扫描模式中，采样初始值为12个状态，保持初始值为8个状态。如果模拟输入信号源的阻抗过高或者ADCLK时钟较慢，可以调整采样时间，以确保足够的采样时间。采样时间必须设置为4个状态或更多，并且不超过255。此外，采样状态设定值应至少符合规定的测试条件。

当包含使能了S&H（采样保持）的通道的组触发到来时，无论当前是否有更高优先级组正在工作，相应的S&H启用通道将立即开始采样保持操作。当所有使能的S&H通道的采样保持操作完成（采样保持操作的持续时间由ADC_SHCTL寄存器中的SHSTIME和SHHTIME设置），对应的Group_pri x组开始进行A/D转换（如果当前有更高优先级的组正在进行A/D转换，该组的通道将进入扩展保持状态，等待高优先级完成A/D转换）。采样保持操作不能被更高优先级组的触发打断或覆盖。这样可以确保采样保持阶段在进行任何后续A/D转换之前正确完成。

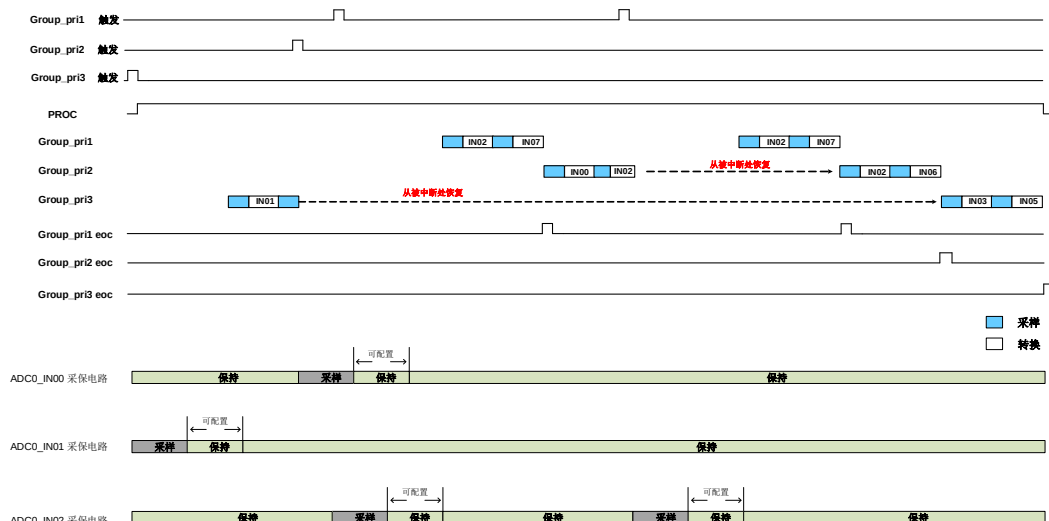
操作示例：低优先级组重新扫描期间高优先级组触发输入（GP2RTCH/GP3RTCH = 1，Group_pri1/Group_pri2/Group_pri3 分别选择 S&H 通道）

使能S&H（采样保持）功能的通道是 ADC0_IN0、ADC0_IN1、ADC0_IN2。Group_pri1选择了 ADC0_IN2/IN7，Group_pri2 选择了 ADC0_IN0/IN2/IN6，Group_pri3 选择了 ADC0_IN1/IN3/IN5。当Group_pri3触发信号到来时，ADC0_IN1的SH电路首先开始采样保持操作，采样保持操作完成后，Group_pri3开始A/D转换。当Group_pri2触发信号到来时，Group_pri3的A/D转换操作被中断，ADC0_IN0的SH电路开始采样保持操作。当Group_pri1触发信号在ADC0_IN0的SH电路采样保持操作期间到来时，ADC0_IN2的SH电路开始采样保持

操作，而ADC0_IN0的SH电路继续未完成的采样保持操作。当ADC0_IN0的SH电路采样保持操作完成后，ADC0_IN0的SH电路进入扩展保持期。当ADC0_IN2的SH电路完成采样保持操作后，Group_pri1开始A/D转换。

注意：对于包含SH功能通道的低优先级组，如果任何更高优先级组中断当前组的操作，当前组的保持和扩展保持期不应超过20us，否则由于电泄漏，SH电路无法保留有效的保持电压。如果Group_pri x组使能了分叉触发模式，并且该组包含使能了SH的通道，也应注意这一点。

图 15-52. 低优先级组重新扫描期间高优先级组触发输入



15.5.19. 通道专用 AD 转换加法功能（数据累加求和/平均模式）

当ADC_CTL0寄存器的ADDEN位置1，AD转换加法功能对所有通道使能。ADC_CTL0的ADDSEL位选择转换数据加法模式：0选择数据累加求和模式，1选择数据平均模式。

在数据累加求和模式中，每个单独通道被连续转换2, 3, 4, 8, 16, 32或者64次，转换结果累加和被存储到对应数据寄存器。ADC_ADDDT0/1寄存器的CHxADDT[3:0]可以设置通道专用的加法次数。

注意：在12位分辨率时使用数据累加求和模式，不可以选择连续转换32和64次，否则不保证结果正确。

在数据平均模式下，每个单独通道被连续转换2, 3, 4, 8, 16, 32或者64次，转换结果累加后的平均值存储到对应数据寄存器。根据存在的噪声成分类型，使用这些结果的平均值可以提高A/D转换的准确性。ADC_ADDDT0/1寄存器的CHxADDT[3:0]可以设置通道专用的加法次数。

AD转换累加求和模式/平均模式可以应用于温度传感器通道和内部参考电压通道（仅ADC2），加法次数由ADC_ADDDT1寄存器的TEMPADDT[3:0] / VINTADDT[3:0]位进行设置。

当ADC_ADDDT0/1寄存器中的专用通道加法次数设置为4'b0000时，则禁能该通道的转换加法。

15.5.20. 数据寄存器自动清零/置位

将ADC_CTL0寄存器中的ACDATA位设置为1，将使能自动清除数据寄存器功能。将

ADC_CTL0寄存器中的ASDATA位设置为1，将使能自动设置数据寄存器功能。

当ADC_CTL0寄存器的ACDATA位置1，包括ADC_CHxDATA, ADC_SDDATA, ADC_BIDATA, ADC_BIDATA1, ADC_BIDATA2, ADC_TEMPDATA 和ADC_VINTDATA的数据寄存器在被CUP或者DMA读取后，其所有位将会被自动清零。

当ADC_CTL0寄存器的ASDATA位置1，包括ADC_CHxDATA, ADC_SDDATA, ADC_BIDATA, ADC_BIDATA1, ADC_BIDATA2 and ADC_VINTDATA的数据寄存器在被CUP或者DMA读取后，其所有位将会被自动置1。

注意：ACDATA和ASDATA只有一个可以被置1，以使能数据自动清零或者置1功能。如果两个位都置1，则两者都无效。

这个功能可以用于数据寄存器更新失败的检测。当数据寄存器被软件读取，或者Group_prix中被更新的通道数据被DMA传输到用户指定的目的地址，数据寄存器的值被清除为0000H或者设置为FFFFH。通过核对读出的数据值是否为0000H或者FFFFH来判定数据寄存器更新失败。

15.5.21. ADC 中断

以下任一个事件发生都可以产生中断：

- Group_pri1/Group_pri2/Group_pri3/Group_pri4的转换轮结束标志；
- 模拟看门狗事件（模拟看门狗A通道x比较状态或模拟看门狗B比较标志）；
- Group_pri1/Group_pri2/Group_pri3/Group_pri4 的DMA溢出事件。

ADC0的中断映射到中断向量 IRQ[18]。

ADC2的中断映射到中断向量 IRQ[75]。

15.6. ADC 寄存器

ADC0 基地址：0x4001 2000

ADC2 基地址：0x4001 2400

15.6.1. 软件 EOC 控制寄存器（ADC_EOCCTL）

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
EOC4RCNT[2:0]			EOC4RF	EOC4F	EOC4RIE	GP4SWST	EVICG4SEL	EOC3RCNT[2:0]			EOC3RF	EOC3F	EOC3RIE	GP3SWST	EVICG3SEL
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
EOC2RCNT[2:0]			EOC2RF	EOC2F	EOC2RIE	GP2SWST	EVICG2SEL	EOC1RCNT[2:0]			EOC1RF	EOC1F	EOC1RIE	GP1SWST	EVICG1SEL
rw			rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw

位/位域	名称	描述
31:29	EOC4RCNT[2:0]	Group_pri4 转换结束轮次 每当 Group_pri4 转换结束的轮次数达到 (EOC4RCNT[2:0] + 1)，EOC4RF 将会置位。
28	EOC4RF	Group_pri4 转换轮次结束标志 当每次 Group_pri4 转换结束的轮次数达到 (EOC4RCNT[2:0] + 1) 时，硬件会将此标志设置为 1。 通过向其写入 0 或通过读取 Group_pri4 的 ADC_GP4DMAR 寄存器来清除。
27	EOC4F	Group_pri4 转换结束标志 0: Group_pri4 转换未结束 1: Group_pri4 转换结束 在 Group_pri4 转换的最后一个通道结束时，由硬件设置。 通过向其写入 0 或通过读取 Group_pri4 的数据寄存器来清除。
26	EOC4RIE	EOC4RF 中断使能 0: EOC4RF 中断禁能 1: EOC4RF 中断使能
25	GP4SWST	软件启动 Group_pri4 组转换 该位置 1 启动 Group_pri4 组转换。 0: Group_pri4 未由软件启动 1: Group_pri4 由软件启动 该位可由软件置位和清零，或者当任意通道开始转换时硬件清零。
24	EVICG4SEL	从 Group_pri4 组选择 EVIC 触发信号 0: EVIC 触发信号选择 EOC4RF 1: EVIC 触发信号选择 EOC4F
23:21	EOC3RCNT[2:0]	Group_pri3 转换结束轮次 每当 Group_pri3 转换结束的轮次数达到 (EOC3RCNT[2:0] + 1)，EOC3RF 将会置位。
20	EOC3RF	Group_pri3 转换轮次结束标志 当每次 Group_pri3 转换结束的轮次数达到 (EOC3RCNT[2:0] + 1) 时，硬件会将此标志设置为 1。 通过向其写入 0 或通过读取 Group_pri3 的 ADC_GP3DMAR 寄存器来清除。
19	EOC3F	Group_pri3 转换结束标志 0: Group_pri3 转换未结束 1: Group_pri3 转换结束 在 Group_pri3 转换的最后一个通道结束时，由硬件设置。 通过向其写入 0 或通过读取 Group_pri3 的数据寄存器来清除。
18	EOC3RIE	EOC3RF 中断使能 0: EOC3RF 中断禁能

		1: EOC3RF中断使能
17	GP3SWST	软件启动 Group_pri3 组转换 该位置 1 启动 Group_pri3 组转换。 0: Group_pri3 未由软件启动 1: Group_pri3 由软件启动 该位可由软件置位和清零，或者当任意通道开始转换时硬件清零。
16	EVICG3SEL	从 Group_pri3 组选择 EVIC 触发信号 0: EVIC触发信号选择EOC3RF 1: EVIC触发信号选择EOC3F
15:13	EOC2RCNT[2:0]	Group_pri2 转换结束轮次 每当Group_pri2转换结束的轮次数达到 (EOC2RCNT[2:0] + 1)，EOC2RF将会置位。
12	EOC2RF	Group_pri2 转换轮次结束标志 当每次 Group_pri2 转换结束的轮次数达到 (EOC2RCNT[2:0] + 1) 时，硬件会将此标志设置为 1。 通过向其写入0或通过读取Group_pri2的ADC_GP2DMAR寄存器来清除。
11	EOC2F	Group_pri2 转换结束标志 0: Group_pri2 转换未结束 1: Group_pri2 转换结束 在 Group_pri2 转换的最后一个通道结束时，由硬件设置。 通过向其写入0或通过读取Group_pri2的数据寄存器来清除。
10	EOC2RIE	EOC2RF 中断使能 0: EOC2RF 中断禁能 1: EOC2RF中断使能
9	GP2SWST	软件启动 Group_pri2 组转换 该位置 1 启动 Group_pri2 组转换。 0: Group_pri2 未由软件启动 1: Group_pri2 由软件启动 该位可由软件置位和清零，或者当任意通道开始转换时硬件清零。
8	EVICG2SEL	从 Group_pri2 组选择 EVIC 触发信号 0: EVIC触发信号选择EOC2RF 1: EVIC触发信号选择EOC2F
7:5	EOC1RCNT[2:0]	Group_pri1 转换结束轮次 每当Group_pri1转换结束的轮次数达到 (EOC1RCNT[2:0] + 1)，EOC1RF将会置位。
4	EOC1RF	Group_pri1 转换轮次结束标志 当每次 Group_pri1 转换结束的轮次数达到 (EOC1RCNT[2:0] + 1) 时，硬件会将此标志设置为 1。

通过向其写入0或通过读取Group_pri1的ADC_GP1DMAR寄存器来清除。

3	EOC1F	Group_pri1 转换结束标志 0: Group_pri1 转换未结束 1: Group_pri1 转换结束 在 Group_pri1 转换的最后一个通道结束时，由硬件设置。 通过向其写入0或通过读取Group_pri1的数据寄存器来清除。
2	EOC1RIE	EOC1RF 中断使能 0: EOC1RF 中断禁能 1: EOC1RF中断使能
1	GP1SWST	软件启动 Group_pri1 组转换 该位置 1 启动 Group_pri1 组转换。 0: Group_pri1 未由软件启动 1: Group_pri1 由软件启动 该位可由软件置位和清零，或者当任意通道开始转换时硬件清零。
0	EVICG1SEL	从 Group_pri1 组选择 EVIC 触发信号 0: EVIC触发信号选择EOC1RF 1: EVIC触发信号选择EOC1F

15.6.2. 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
SWEND	EVICCTL[2:0]			保留			GP4EN	GP3EN	ADDSEL	GSCAN[1:0]		PROCF	保留		ADDEN
rw	rw						rw	rw	rw	rw		r			rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASDATA	ACDATA	SDVOG[1:0]		SDSFIX	SDSEN	DAL	CHRSEL[4:0]					GPRIEN	保留		LPGCNT
rw	rw	rw		rw	rw	rw	rw					rw			rw

位/位域	名称	描述
31	SWEND	软件结束Group_pri1/pri2/pri3/pri4转换 通过软件设置来结束所有组的转换。 通过硬件清除。
30:28	EVICCTL[2:0]	EVIC连接信号控制 000: 在扫描Group_pri1完成时生成事件信号。 001: 在扫描Group_pri2完成时生成事件信号。 010: 在扫描Group_pri3完成时生成事件信号。 011: 在扫描Group_pri4完成时生成事件信号。 100: 在扫描Group_pri1、Group_pri2、Group_pri3或Group_pri4完成时生成事件信号。

		其他数值：保留。 禁止使用除上述设置之外的其他设置。
27:25	保留	必须保持复位值。
24	GP4EN	启用 Group_pri4 组 AD 转换操作 0: Group_pri4 不使用 1: 使用 Group_pri4 使用
23	GP3EN	启用 Group_pri3 组 AD 转换操作 0: Group_pri3 不使用 1: 使用 Group_pri3 使用
22	ADDSEL	A/D 转换数据加法模式选择 0: 数据求和模式 1: 数据平均模式
21:20	GSCAN[1:0]	组扫描模式 当 PROCF 位为 0 时可以配置该位。 00: 优先级 1 组单次扫描模式 01: 多组扫描模式 10: 优先级 1 组连续扫描模式 11: 保留 注意： 当 GSCAN[1:0]为 2'b01，如果应用需要使用两个组，只能使用 Group_pri1 和 Group_pri2。如果应用需要使用三个组，只能使用 Group_pri1，Group_pri2 和 Group_pri3。如果应用需要使用一个组，则只能使用 Group_pri1。
19	PROCF	A/D 转换正在进行标识 该位软件只读，表明 A/D 转换正在进行 0: A/D 转换未进行 1: A/D 转换正在进行
18:17	保留	必须保持复位值。
16	ADDEN	使能加法功能 0: 禁能 AD 转换加法功能 1: 使能 AD 转换加法功能
15	ASDATA	使能数据寄存器自动置位 当该位置 1，数据寄存器被 CPU 或者 DMA 读之后，该数据寄存器所有位置位，包括 ADC_CHxDATA, ADC_SDDATA, ADC_GPxBIDATA, ADC_GPxBIDATA1, ADC_GPxBIDATA2, ADC_TEMPDATA 和 ADC_VINTDATA。 0: 禁能数据寄存器自动置位 1: 使能数据寄存器自动置位
14	ACDATA	使能数据寄存器自动清零 当该位置 1，数据寄存器被 CPU 或者 DMA 读之后，该数据寄存器所有位清零，包括 ADC_CHxDATA, ADC_SDDATA, ADC_GPxBIDATA, ADC_GPxBIDATA1,

		ADC_GPxBIDATA2, ADC_TEMPDATA 和 ADC_VINTDATA。 0: 禁能数据寄存器自动清零 1: 使能数据寄存器自动清零
13:12	SDVOG[1:0]	自诊断固定电压配置 00: 保留 01: 0V 10: $V_{REFP}/2$ 11: V_{REFP}
11	SDSFIX	使能自诊断固定电压 当该为置位, 自诊断转换固定电压 (SDVOG[1:0]配置的电压)。 当该为清零, 自诊断轮流转换 0, $V_{REFP}/2$, V_{REFP} 。 转换结果存储到 ADC_SDDATA 寄存器。 当 PROCF 位为 0 时可以配置该位。 0: 禁能自诊断固定电压 1: 使能自诊断固定电压
10	SDSEN	使能自诊断 当该为置位, 在开始一次扫描之前, 先固定或者轮流转换一个电压 (0, $V_{REFP}/2$, V_{REFP})。 当 PROCF 位为 0 时可以配置该位。 0: 禁能自诊断 1: 使能自诊断
9	DAL	数据对齐 0: (LSB) 最低有效位对齐 1: (MSB) 最高有效位对齐
8:4	CHRSEL[4:0]	预充电或者放电选择 CHRSEL[4]位选择预充电或者放电, CHRSEL[3:0]配置预充电或者放电的时长。 00000~00001: 禁能预充电和放电 00000~01111: 选择放电, 放电时长位 CHRSEL[3:0]个 CLK_ADC 10000~10001: 禁能预充电和放电 10000~11111: 选择预充电, 预充电时长位 CHRSEL[3:0]个 CLK_ADC 注意: 当 CH17EN=1 (使能内部参考电压通道), CHRSEL[4:0]硬件自动固定为 5'b01111。
3	GPRIEN	使能组优先级 当该位置位, 高优先级组可以打断低优先级组的扫描。组优先级顺序为: Group_pri1 > Group_pri2 > Group_pri3 > Group_pri4。 0: 禁能组优先级 1: 使能组优先级 注意: 当 GPRIEN 置位时, GSCAN[1:0]必须设置为 2'b01 (多组扫描模式)。
2:1	保留	必须保持复位值。

0	LPGCNT	使能最低优先级组连续扫描 该位仅当 GPRIEN 置位时有效。 0: 最低优先级组扫描一次 1: 最低优先级组连续扫描 注意: 对于两个组 (SCAN[1:0]=2'b01, GP3EN=0, GP4EN=0), Group_pri1 组是最低优先级组; 对于三个组 (SCAN[1:0]=2'b01, GP3EN=1, GP4EN=0), Group_pri3 组是最低优先级组; 对于三个组 (SCAN[1:0]=2'b01, GP3EN=0, GP4EN=1), Group_pri4 组是最低优先级组。

15.6.3. 组配置寄存器 0 (ADC_GPCGF0)

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		GP2RTEN	GP2RTCH	保留		GP2TRGED[1:0]		GP2TRGSRC[5:0]						GP2TRGTY	G2TRGEN
rw		rw		rw				rw						rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						GP1TRGED[1:0]		GP1TRGSRC[5:0]						GP1TRGTY	G1TRGEN
rw						rw						rw		rw	

位/位域	名称	描述
31:30	保留	必须保持复位值。
29	GP2RTEN	使能 Group_pri2 组重启扫描 该位仅当 GPRIEN 置位时有效。 当该位为 0, 如果低优先级组被高优先级组中止, 当高优先级组扫描完成后, 低优先级组不重启。在高优先级扫描期间产生了低优先级组触发, 该触发被忽略。 当该位为 1, 如果低优先级组被高优先级组中止, 当高优先级组扫描完成后, 低优先级重启扫描。在高优先级扫描期间产生了低优先级组触发, 等高优先级组扫描完成后, 启动触发对应的低优先级组扫描。 0: 禁能组重启 1: 使能组重启。
28	GP2RTCH	选择重启扫描 Group_pri2 组的通道 该位当 GPRIEN=1, GP2RTEN=1。 当 PROCF 位为 0 时可以配置该位。 0: 从初始通道开始重新扫描低优先级组 1: 从上次被中止的通道开始扫描低优先级组
25:24	保留	必须保持复位值。
25:24	GP2TRGED[1:0]	Group_pri2组触发边沿选择 00: 保留 01: 上升沿触发

		01: 下降沿触发 11: 双边沿触发
23:18	GP2TRGSRC[5:0]	Group_pri2组同步触发源 000000: 保留 000001~101000: 表 15-10. ADC同步触发源 。 101001~111111: 保留
17	GP2TRGTY	Group_pri2 组触发类型选择 0: Group_pri2 组选择同步触发 1: Group_pri2组选择异步触发 注意: 在Group_pri1/pri2/pri3/pri4中, 仅一个组可以使能异步触发, 不能同时使能两个或者更多个组的异步触发。
16	G2TRGEN	使能Group_pri2组触发 0: 禁能Group_pri2组同步和异步触发 1: 使能Group_pri2组同步和异步触发 注意: 当该位为1, 软件触发 (GP2SWST) 无效。
15:10	保留	必须保持复位值。
9:8	GP1TRGED[1:0]	Group_pri1组触发边沿选择 00: 保留 01: 上升沿触发 01: 下降沿触发 11: 双边沿触发
7:2	GP1TRGSRC[5:0]	Group_pri1组同步触发源 000000: 保留 000001~101000: 表 15-10. ADC同步触发源 。 101001~111111: 保留
1	GP1TRGTY	Group_pri1 组触发类型选择 0: Group_pri1 组选择同步触发 1: Group_pri1组选择异步触发 注意: 在Group_pri1/pri2/pri3/pri4中, 仅一个组可以使能异步触发, 不能同时使能两个或者更多个组的异步触发。
0	G1TRGEN	使能Group_pri1组触发 0: 禁能Group_pri1组同步和异步触发 1: 使能Group_pri1组同步和异步触发 注意: 当该位为1, 软件触发 (GP1SWST) 无效。

15.6.4. 组配置寄存器 1 (ADC_GPCGF1)

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	GP4RTEN	GP4RTCH	保留	GP4TRGED[1:0]	GP4TRGSRC[5:0]					GP4TRGTY	G4TRGEN				
	rw	rw		rw	rw					rw	rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	GP3RTEN	GP3RTCH	保留	GP3TRGED[1:0]	GP3TRGSRC[5:0]					GP3TRGTY	G3TRGEN				
	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	GP4RTEN	使能 Group_pri4 组重启扫描 该位仅当 GPRIEN 置位时有效。 当该位为 0，如果低优先级组被高优先级组中止，当高优先级组扫描完成后，低优先级组不重启。在高优先级扫描期间产生了低优先级组触发，该触发被忽略。 当该位为 1，如果低优先级组被高优先级组中止，当高优先级组扫描完成后，低优先级组重启扫描。在高优先级扫描期间产生了低优先级组触发，等高优先级组扫描完成后，启动触发对应的低优先级组扫描。 0：禁能组重启 1：使能组重启。
28	GP4RTCH	选择重启扫描 Group_pri4 组的通道 该位当 GPRIEN=1，GP4RTEN=1。 当 PROCF 位为 0 时可以配置该位。 0：从初始通道开始重新扫描低优先级组 1：从上次被中止的通道开始扫描低优先级组
25:24	保留	必须保持复位值。
25:24	GP4TRGED[1:0]	Group_pri4组触发边沿选择 00：保留 01：上升沿触发 01：下降沿触发 11：双边沿触发
23:18	GP4TRGSRC[5:0]	Group_pri4组同步触发源 000000：保留 000001~101000：表 15-10. ADC同步触发源。 101001~111111：保留
17	GP4TRGTY	Group_pri4 组触发类型选择 0：Group_pri4 组选择同步触发 1：Group_pri4组选择异步触发 注意：在Group_pri1/pri2/pri3/pri4中，仅一个组可以使能异步触发，不能同时使能两个或者更多个组的异步触发。
16	G4TRGEN	使能Group_pri4组触发

		0: 禁能Group_pri4组同步和异步触发 1: 使能Group_pri4组同步和异步触发 注意: 当该位为1, 软件触发(GP4SWST)无效。
15:14	保留	必须保持复位值。
13	GP3RTEN	使能 Group_pri3 组重启扫描 该位仅当 GPRIEN 置位时有效。 当该位为 0, 如果低优先级组被高优先级组中止, 当高优先级组扫描完成后, 低优先级组不重启。在高优先级扫描期间产生了低优先级组触发, 该触发被忽略。 当该位为 1, 如果低优先级组被高优先级组中止, 当高优先级组扫描完成后, 低优先级组重启扫描。在高优先级扫描期间产生了低优先级组触发, 等高优先级组扫描完成后, 启动触发对应的低优先级组扫描。 0: 禁能组重启 1: 使能组重启。
12	GP3RTCH	选择重启扫描 Group_pri3 组的通道 该位当 GPRIEN=1, GP4RTEN=1。 当 PROCF 位为 0 时可以配置该位。 0: 从初始通道开始重新扫描低优先级组 1: 从上次被中止的通道开始扫描低优先级组
11:10	保留	必须保持复位值。
9:8	GP3TRGED[1:0]	Group_pri3组触发边沿选择 00: 保留 01: 上升沿触发 01: 下降沿触发 11: 双边沿触发
7:2	GP3TRGSRC[5:0]	Group_pri3组同步触发源 000000: 保留 000001~101000: 表 15-10. ADC同步触发源。 101001~111111: 保留
1	GP3TRGTY	Group_pri3 组触发类型选择 0: Group_pri3 组选择同步触发 1: Group_pri3组选择异步触发 注意: 在Group_pri1/pri2/pri3/pri4中, 仅一个组可以使能异步触发, 不能同时使能两个或者更多个组的异步触发。
0	G3TRGEN	使能Group_pri3组触发 0: 禁能Group_pri3组同步和异步触发 1: 使能Group_pri3组同步和异步触发 注意: 当该位为1, 软件触发(GP3SWST)无效。

15.6.5. 通道选择寄存器 0 (ADC_CHSEL0)

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				GP2VINT	GP2TEMP	GP2CH9	GP2CH8	GP2CH7	GP2CH6	GP2CH5	GP2CH4	GP2CH3	GP2CH2	GP2CH1	GP2CH0
				r/w	r/w	r/w	r/w	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
保留				GP1VINT	GP1TEMP	GP1CH9	GP1CH8	GP1CH7	GP1CH6	GP1CH5	GP1CH4	GP1CH3	GP1CH2	GP1CH1	GP1CH0
				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
31:28	保留	必须保持复位值
27	GP2VINT	Group_pri2组选择转换内部参考电压 参考 GP2CH00。 注意: 该位仅用在 ADC2 中。
26	GP2TEMP	Group_pri2组选择转换温度传感器 参考GP2CH00。 注意: 该位仅用在ADC2中。
25	GP2CH09	Group_pri2组选择ADCx_IN9通道 参考GP2CH00。 注意: 该位仅用在ADC2中。
24	GP2CH08	Group_pri2组选择ADCx_IN8通道 参考GP2CH00。 注意: 该位仅用在ADC2中。
23	GP2CH07	Group_pri2组选择ADCx_IN7通道 参考GP2CH00。 注意: 该位仅用在ADC2中。
22	GP2CH06	Group_pri2组选择ADCx_IN6通道 参考GP2CH00。
21	GP2CH05	Group_pri2组选择ADCx_IN5通道 参考GP2CH00。 注意: 该位仅用在ADC2中。
20	GP2CH04	Group_pri2组选择ADCx_IN4通道 参考GP2CH00。
19	GP2CH03	Group_pri2组选择ADCx_IN3通道 参考GP2CH00。

18	GP2CH02	Group_pri2组选择ADCx_IN2通道 参考GP2CH00。
17	GP2CH01	Group_pri2组选择ADCx_IN1通道 参考GP2CH00。
16	GP2CH00	Group_pri2组选择ADCx_IN0通道 0: 不选择该通道 1: 选择该通道 注意: 1) 当 BIMEN 为 1（使能分叉触发模式），该位无效。 2) 一个通道只能被一个组选中。
15:12	保留	必须保持复位值
11	GP1VINT	Group_pri1组选择转换内部参考电压 参考 GP1CH00。 注意: 该位仅用在ADC2中。
10	GP1TEMP	Group_pri1组选择转换温度传感器 参考GP1CH00。 注意: 该位仅用在ADC2中。
9	GP1CH09	Group_pri1组选择ADCx_IN9通道 参考GP1CH00。 注意: 该位仅用在ADC2中。
8	GP1CH08	Group_pri1组选择ADCx_IN8通道 参考GP1CH00。 注意: 该位仅用在ADC2中。
7	GP1CH07	Group_pri1组选择ADCx_IN7通道 参考GP1CH00。 注意: 该位仅用在ADC2中。
6	GP1CH06	Group_pri1组选择ADCx_IN6通道 参考GP1CH00。
5	GP1CH05	Group_pri1组选择ADCx_IN5通道 参考GP1CH00。 注意: 该位仅用在ADC2中。
4	GP1CH04	Group_pri1组选择ADCx_IN4通道 参考GP1CH00。
3	GP1CH03	Group_pri1组选择ADCx_IN3通道 参考GP1CH00。
2	GP1CH02	Group_pri1组选择ADCx_IN2通道

参考GP1CH00。

- 1 GP1CH01 Group_pri1组选择ADCx_IN1通道
参考GP1CH00。
- 0 GP1CH00 Group_pri1组选择ADCx_IN0通道
0: 不选择该通道
1: 选择该通道
注意:
1) 当BIMEN为1(使能分叉触发模式), 该位无效。
2) 一个通道只能被一个组选中。

15.6.6. 通道选择寄存器 1 (ADC_CHSEL1)

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				GP4VINT	GP4TEMP	GP4CH9	GP4CH8	GP4CH7	GP4CH6	GP4CH5	GP4CH4	GP4CH3	GP4CH2	GP4CH1	GP4CH0
				r/w	r/w	r/w	r/w	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
保留				GP3VINT	GP3TEMP	GP3CH9	GP3CH8	GP3CH7	GP3CH6	GP3CH5	GP3CH4	GP3CH3	GP3CH2	GP3CH1	GP3CH0
				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
31:28	保留	必须保持复位值
27	GP4VINT	Group_pri4组选择转换内部参考电压 参考 GP4CH00。 注意: 该位仅用在ADC2中。
26	GP4TEMP	Group_pri4组选择转换温度传感器 参考GP4CH00。 注意: 该位仅用在ADC2中。
25	GP4CH09	Group_pri4组选择ADCx_IN9通道 参考GP4CH00。 注意: 该位仅用在ADC2中。
24	GP4CH08	Group_pri4组选择ADCx_IN8通道 参考GP4CH00。 注意: 该位仅用在ADC2中。
23	GP4CH07	Group_pri4组选择ADCx_IN7通道 参考GP4CH00。 注意: 该位仅用在ADC2中。

22	GP4CH06	Group_pri4组选择ADCx_IN6通道 参考GP4CH00。
21	GP4CH05	Group_pri4组选择ADCx_IN5通道 参考GP4CH00。 注意： 该位仅用在ADC2中。
20	GP4CH04	Group_pri4组选择ADCx_IN4通道 参考GP4CH00。
19	GP4CH03	Group_pri4组选择ADCx_IN3通道 参考GP4CH00。
18	GP4CH02	Group_pri4组选择ADCx_IN2通道 参考GP4CH00。
17	GP4CH01	Group_pri4组选择ADCx_IN1通道 参考GP4CH00。
16	GP4CH00	Group_pri4组选择ADCx_IN0通道 0：不选择该通道 1：选择该通道 注意： 1) 当 BIMEN 为 1（使能分叉触发模式），该位无效。 2) 一个通道只能被一个组选中。
15:12	保留	必须保持复位值
11	GP3VINT	Group_pri3组选择转换内部参考电压 参考 GP3CH00。 注意： 该位仅用在ADC2中。
10	GP3TEMP	Group_pri3组选择转换温度传感器 参考GP3CH00。 注意： 该位仅用在ADC2中。
9	GP3CH09	Group_pri3组选择ADCx_IN9通道 参考GP3CH00。 注意： 该位仅用在ADC2中。
8	GP3CH08	Group_pri3组选择ADCx_IN8通道 参考GP3CH00。 注意： 该位仅用在ADC2中。
7	GP3CH07	Group_pri3组选择ADCx_IN7通道 参考GP3CH00。 注意： 该位仅用在ADC2中。
6	GP3CH06	Group_pri3组选择ADCx_IN6通道

参考GP3CH00。

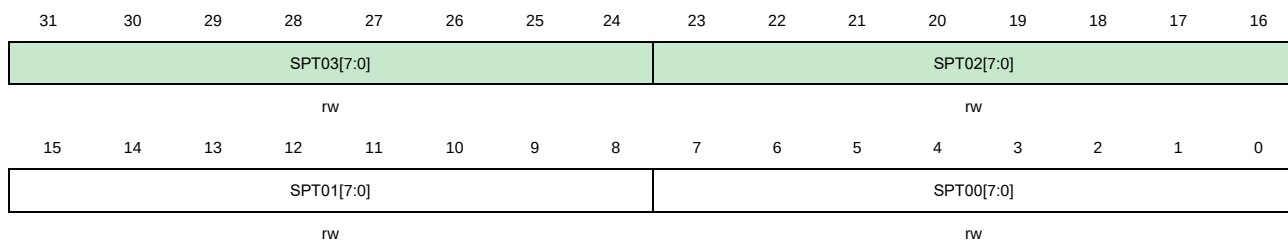
5	GP3CH05	Group_pri3组选择ADCx_IN5通道 参考GP3CH00。 注意： 该位仅用在ADC2中。
4	GP3CH04	Group_pri3组选择ADCx_IN4通道 参考GP3CH00。
3	GP3CH03	Group_pri3组选择ADCx_IN3通道 参考GP3CH00。
2	GP3CH02	Group_pri3组选择ADCx_IN2通道 参考GP3CH00。
1	GP3CH01	Group_pri3组选择ADCx_IN1通道 参考GP3CH00。
0	GP3CH00	Group_pri3组选择ADCx_IN0通道 0：不选择该通道 1：选择该通道 注意： 3) 当BIMEN为1（使能分叉触发模式），该位无效。 4) 一个通道只能被一个组选中。

15.6.7. 采样时间寄存器 0（ADC_SAMPR0）

地址偏移：0x20

复位值：0x0202 0202

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:24	SPT03[7:0]	通道ADCx_IN3采样时间 参考SPT00[7:0]描述。
23:16	SPT02[7:0]	通道ADCx_IN2采样时间 参考SPT00[7:0]描述。
15:8	SPT01[7:0]	通道ADCx_IN1采样时间 参考SPT00[7:0]描述。

7:0	SPT00[7:0]	通道ADCx_IN0采样时间 通道ADCx_IN0采样时间(SPT00[7:0]+0.5)个CLK_ADC。 注意： 最小值为0x02。

15.6.8. 采样时间寄存器 1 (ADC_SAMPR1)

地址偏移：0x24

复位值：0x0202 0202

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPT07[7:0]								SPT06[7:0]							
rw								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPT05[7:0]								SPT04[7:0]							
rw								rw							

位/位域	名称	描述
31:24	SPT07[7:0]	通道ADCx_IN7采样时间 参考SPT04[7:0]描述。 注意： 该位仅用在ADC2中。
23:16	SPT06[7:0]	通道ADCx_IN6采样时间 参考SPT04[7:0]描述。
15:8	SPT05[7:0]	通道ADCx_IN5采样时间 参考SPT04[7:0]描述。
7:0	SPT04[7:0]	通道ADCx_IN4采样时间 通道ADCx_IN4采样时间(SPT04[7:0] +0.5)个CLK_ADC。 注意： 最小值为0x02。

15.6.9. 采样时间寄存器 2 (ADC_SAMPR2)

地址偏移：0x28

复位值：0x0202 0202

该寄存器仅用在 ADC2 中。

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPT11[7:0]								SPT10[7:0]							
rw								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPT09[7:0]								SPT08[7:0]							

rw

rw

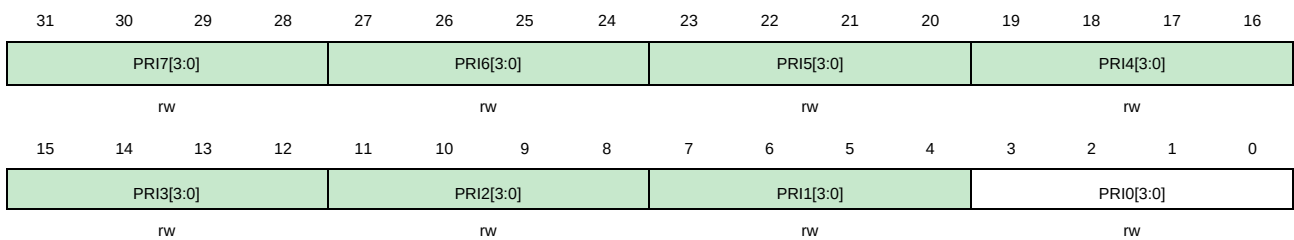
位/位域	名称	描述
31:24	SPTVINT[7:0]	内部参考电压通道采样时间 参考SPT08[7:0]描述。
23:16	SPTTEMP[7:0]	温度传感器通道采样时间 参考SPT08[7:0]描述。
15:8	SPT09[7:0]	通道ADCx_IN9采样时间 参考SPT09[7:0]描述。
7:0	SPT08[7:0]	通道ADCx_IN8采样时间 通道ADCx_IN8采样时间(SPT08[7:0] + 0.5)个CLK_ADC。 注意： 最小值为0x02。

15.6.10. 通道优先级寄存器 0（ADC_CHPRI0）

地址偏移：0x30

复位值：0x7654 3210

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:28	PRI7[3:0]	第7优先级通道 参考PRI0[3:0]。 注意： 该位仅用在ADC2中。
27:24	PRI6[3:0]	第6优先级通道 参考PRI0[3:0]。
23:20	PRI5[3:0]	第5优先级通道 参考PRI0[3:0]。
19:16	PRI4[3:0]	第4优先级通道 参考PRI0[3:0]。
15:12	PRI3[3:0]	第3优先级通道 参考PRI0[3:0]。
11:8	PRI2[3:0]	第2优先级通道

参考PRI0[3:0]。

7:4	PRI1[3:0]	第1优先级通道 参考PRI0[3:0]。
3:0	PRI0[3:0]	第0优先级通道 设置优先级为0的通道编号。 0000: ADCx_IN0 0001: ADCx_IN1 0010: ADCx_IN2 0011: ADCx_IN3 0100: ADCx_IN4 0101: ADCx_IN5 0110: ADCx_IN6 0111: ADCx_IN7, 仅用在ADC2中。 1000: ADCx_IN8, 仅用在ADC2中。 1001: ADCx_IN9, 仅用在ADC2中。 1101: 温度传感器通道, 仅用在ADC2中。 1110: 内部参考电压通道, 仅用在ADC2中。 比如, PRI0[3:0]=4'b0010, 表示通道ADCx_IN2优先级为0。 注意: 相同通道只能配置一个优先级。

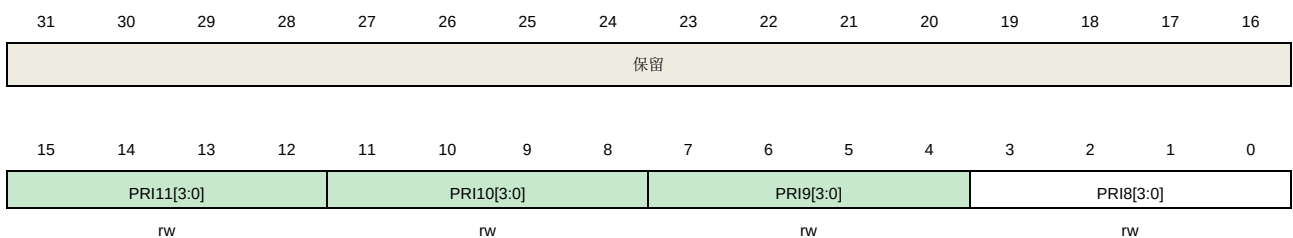
15.6.11. 通道优先级寄存器 1 (ADC_CHPRI1)

地址偏移: 0x34

复位值: 0x0000 BA98

注意: 该寄存器仅用在ADC2中。

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	PRI11[3:0]	第11优先级通道 参考PRI8[3:0]。
11:8	PRI10[3:0]	第10优先级通道 参考PRI8[3:0]。

7:4	PRI9[3:0]	第9优先级通道 参考PRI8[3:0]。
3:0	PRI8[3:0]	第8优先级通道 设置优先级为8的通道编号。 0000: ADCx_IN0 0001: ADCx_IN1 0010: ADCx_IN2 0011: ADCx_IN3 0100: ADCx_IN4 0101: ADCx_IN5 0110: ADCx_IN6 0111: ADCx_IN7, 仅用在ADC2中。 1000: ADCx_IN8, 仅用在ADC2中。 1001: ADCx_IN9, 仅用在ADC2中。 1101: 温度传感器通道, 仅用在ADC2中。 1110: 内部参考电压通道, 仅用在ADC2中。 比如, PRI8[3:0]=4'b0010, 表示通道ADCx_IN2优先级为8。 注意: 相同通道只能配置一个优先级。

15.6.12. 通道加法次数寄存器 0 (ADC_ADDT0)

地址偏移: 0x40

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH7ADDT[3:0]				CH6ADDT[3:0]				CH5ADDT[3:0]				CH4ADDT[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3ADDT[3:0]				CH2ADDT[3:0]				CH1ADDT[3:0]				CH0ADDT[3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:28	CH7ADDT[3:0]	ADCx_IN7总的加法转换次数 参考CH0ADDT[3:0]。 注意: 该位仅用在ADC2中。
27:24	CH6ADDT[3:0]	ADCx_IN6总的加法转换次数 参考CH0ADDT[3:0]。
23:20	CH5ADDT[3:0]	ADCx_IN5总的加法转换次数 参考CH0ADDT[3:0]。
19:16	CH4ADDT[3:0]	ADCx_IN4总的加法转换次数

参考CH0ADDT[3:0]。

15:12	CH3ADDT[3:0]	ADCx_IN3总的加法转换次数 参考CH0ADDT[3:0]。
11:8	CH2ADDT[3:0]	ADCx_IN2总的加法转换次数 参考CH0ADDT[3:0]。
7:4	CH1ADDT[3:0]	ADCx_IN1总的加法转换次数 参考CH0ADDT[3:0]。
3:0	CH0ADDT[3:0]	ADCx_IN0总的加法转换次数 4'b0000: 1次转换（禁能加法功能，正常转换） 4'b0001: 2次转换（1次加法） 4'b0010: 3次转换（2次加法） 4'b0011: 4次转换（3次加法） 4'b0100: 8次转换（7次加法） 4'b0101: 16次转换（15次加法） 4'b0110: 32次转换（31次加法） 4'b0111: 64次转换（63次加法） 其他值：保留

15.6.13. 通道加法次数寄存器 1（ADC_ADDT1）

地址偏移：0x44

复位值：0x0000 0000

该位仅用在ADC2中。

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CHVADDT[3:0]				保留				CH13ADDT[3:0]				CH12ADDT[3:0]			
rw								rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH11ADDT[3:0]				CH10ADDT[3:0]				CH9ADDT[3:0]				CH8ADDT[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	VINTADDT[3:0]	ADCx_IN11总的加法转换次数 参考CH8ADDT[3:0]。
11:8	TEMPADDT[3:0]	ADCx_IN10总的加法转换次数 参考CH8ADDT[3:0]。
7:4	CH9ADDT[3:0]	ADCx_IN9总的加法转换次数

参考CH8ADDT[3:0]。

3:0	CH8ADDT[3:0]	ADCx_IN8总的加法转换次数
		4'b0000: 1次转换（禁能加法功能，正常转换）
		4'b0001: 2次转换（1次加法）
		4'b0010: 3次转换（2次加法）
		4'b0011: 4次转换（3次加法）
		4'b0100: 8次转换（7次加法）
		4'b0101: 16次转换（15次加法）
		4'b0110: 32次转换（31次加法）
		4'b0111: 64次转换（63次加法）
		其他值：保留

15.6.14. 自诊断数据寄存器（ADC_SDDATA）

地址偏移：0x5C

复位值：0x0000 0000

该寄存器只能按字(32 位)访问。



LSB对齐(DAL=0):

位/位域	名称	描述
15:14	SDSTA[1:0]	自诊断转换电压状态 00: 从上电开始自诊断未执行过 01: 自诊断转换0V 10: 自诊断转换 $V_{REFP}/2$ 11: 自诊断转换 V_{REFP}
13:12	保留	必须保持复位值
11:0	SDDATA[11:0]	自诊断数据 这些为包含了自诊断转换结果，只读。

MSB对齐(DAL=1):

位/位域	名称	描述
15:4	SDDATA[11:0]	自诊断数据 这些为包含了自诊断转换结果，只读。

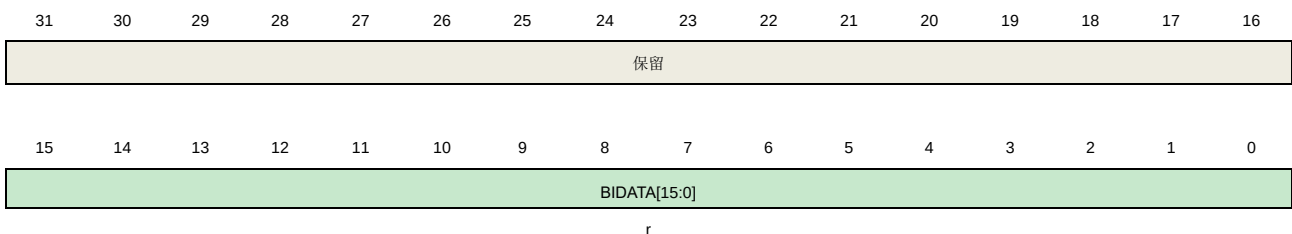
3:2	保留	必须保持复位值
1:0	SDSTA[1:0]	自诊断转换电压状态 00: 从上电开始自诊断未执行过 01: 自诊断转换0V 10: 自诊断转换 $V_{REFP}/2$ 11: 自诊断转换 V_{REFP}

15.6.15. Group_pri x 组分叉数据寄存器 (ADC_GPxBIDATA, x=1,2,3,4)

地址偏移: $0x60+(x-1)*0xC$

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问。



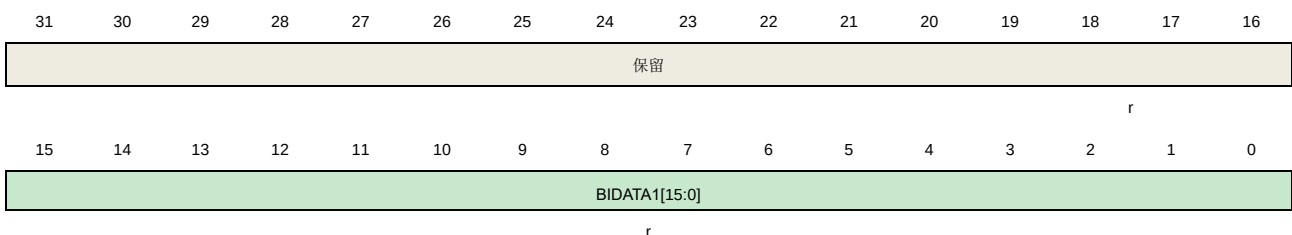
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	BIDATA[15:0]	分叉触发模式中第二次触发转换的数据 如果分叉触发模式使能 (BIMEN=1)，这些位包含了第二次触发通道 (BICHSEL[4:0]) 转换的数据，这些位只读。

15.6.16. Group_pri x 组分叉数据寄存器 1 (ADC_GPxBIDATA1, x=1,2,3,4)

地址偏移: $0x64+(x-1)*0xC$

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问。



位/位域	名称	描述

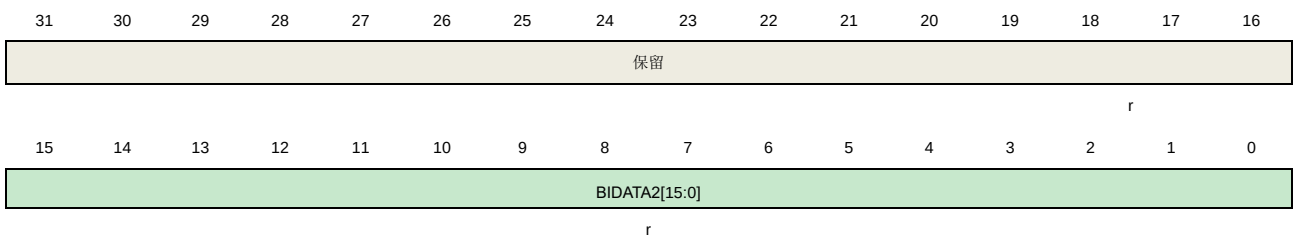
31:16	保留	必须保持复位值
15:0	BIDATA1[15:0]	扩展分叉触发模式中TRGA触发转换的数据 如果分叉触发模式使能（BIMEN=1），这些位包含了第一次触发通道（BICHSEL[4:0]）转换的数据，这些位只读。

15.6.17. Group_pri x 组分叉数据寄存器 2（ADC_GPxBIDATA2, x=1,2,3,4）

地址偏移：0x68+(x-1)*0xC

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	BIDATA2[15:0]	扩展分叉触发模式中TRGB触发转换的数据 如果分叉触发模式使能（BIMEN=1），这些位包含了第二次触发通道（BICHSEL[4:0]）转换的数据，这些位只读。

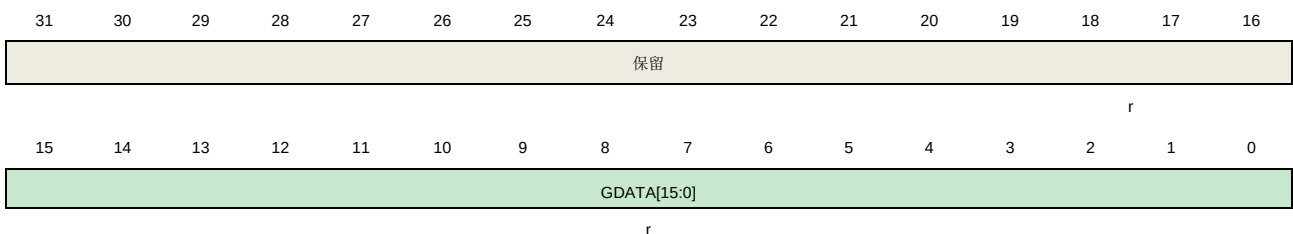
15.6.18. Group_pri x 组 DMA 数据寄存器（ADC_GPxDMAR, x=1,2,3,4）

地址偏移：0x90 + (x-1)*0x4

复位值：0x0000 0000

该寄存器仅推荐使用 DMA 访问（不推荐用 CPU 访问）。

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	GDATA[15:0]	DMA访问的Group_pri x组数据

这些位只读

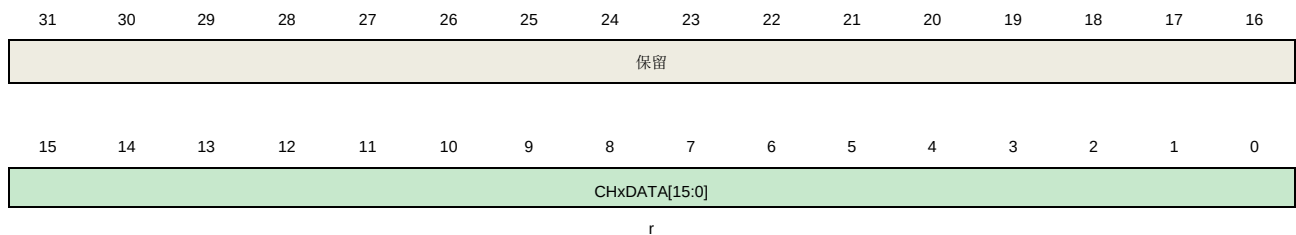
15.6.19. 通道 x 数据寄存器 (ADC_CHxDATA)

地址偏移: $0xA0 + 0x4 \times x$

复位值: 0x0000 0000

对于ADC0, $x = 0..6$ 。对于ADC2, $x = 0..9$ 。

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
15:14	CHxDATA[15:0]	通道x数据 这些位包含了内部参考电压通道的转换结果，这些位只读。

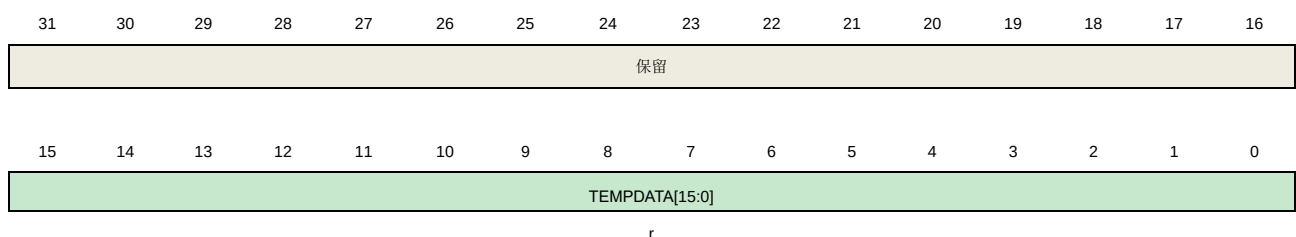
15.6.20. 温度传感器通道数据寄存器 (ADC_TEMPDATA)

地址偏移: 0xC8

复位值: 0x0000 0000

该寄存器仅用在ADC2中。

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	TEMPDATA[15:0]	内部温度传感器通道数据 这些位包含了内部温度传感器通道的转换结果，这些位只读。

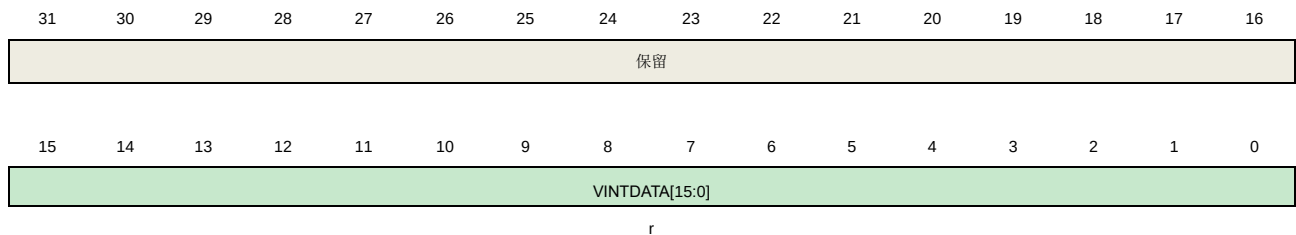
15.6.21. 内部参考电压通道数据寄存器 (ADC_VINTDATA)

地址偏移: 0xCC

复位值: 0x0000 0000

该寄存器仅用在ADC2中。

该寄存器只能按字(32 位)访问。



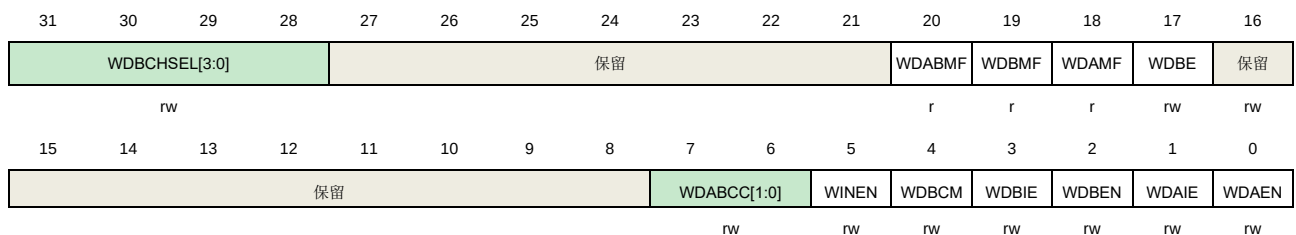
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	VINTDATA[15:0]	内部参考电压通道数据 这些位包含了内部参考电压通道的转换结果，这些位只读。

15.6.22. 看门狗控制寄存器 (ADC_WDCTL)

地址偏移: 0x100

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	WDBCHSEL[3:0]	模拟看门狗B通道选择 0000: ADCx_IN0 0001: ADCx_IN1 0010: ADCx_IN2 0011: ADCx_IN3 0100: ADCx_IN4 0101: ADCx_IN5 0110: ADCx_IN6 0111: ADCx_IN7, 仅在ADC2中可用。 1000: ADCx_IN8, 仅在ADC2中可用。

		1001: ADCx_IN9, 仅在ADC2中可用。
		1010: 温度传感器通道
		1011: 内部参考电压通道
		其他值: 保留
27:21	保留	必须保持复位值
20	WDABMF	模拟看门狗 A/B 复合比较监视标志 此位指示组合结果。仅在同时启用看门狗 A 和看门狗 B 同时运行时, 此位有效 0: 不满足模拟看门狗 A/B 的复合条件 1: 满足模拟看门狗 A/B 的复合条件 由硬件设置和清除。只能由软件读取。
19	WDBMF	模拟看门狗 B 比较监视标志 0: 不满足模拟看门狗 B 的条件 1: 满足模拟看门狗 B 的条件 由硬件设置和清除。只能由软件读取。
18	WDAMF	模拟看门狗 A 比较监视标志 0: 不满足模拟看门狗 A 的条件 1: 满足模拟看门狗 A 的条件 由硬件设置和清除。只能由软件读取。
17	WDBEF	模拟看门狗 B 事件标志 0: 无模拟看门狗 B 事件 1: 发生模拟看门狗 B 事件 当模拟看门狗 B 的比较条件满足时, 由硬件置位。 软件写0可清除该位。
16:8	保留	必须保持复位值
7:6	WDABCC[1:0]	模拟看门狗 A/B 复合条件配置 00: (模拟看门狗 A 条件匹配) OR (模拟看门狗 B 条件匹配) 01: (模拟看门狗 A 条件匹配) XOR (模拟看门狗 B 条件匹配) 10: (模拟看门狗 A 条件匹配) AND (模拟看门狗 B 条件匹配) 11: 保留
5	WINEN	使能窗口模式 0: 禁能窗口模式。看门狗 A 或 B 将低阈值与 A/D 转换值进行比较, 类似于一个比较器。 1: 使能窗口模式。看门狗 A 或 B 将高/低阈值与 A/D 转换值进行比较。
4	WDBCM	模拟看门狗B比较条件模式 当WINEN位为0, 比较比较条件为: 0: WDBLT[15:0] > A/D转换值 1: WDBLT[15:0] < A/D转换值 当WINEN位为1, 比较比较条件为: 0: (A/D转换值 < WDBLT[15:0]) 或者 (WDBHT[15:0] < A/D转换值)

1: WDBLT[15:0] < A/D转换值 < WDBHT[15:0]

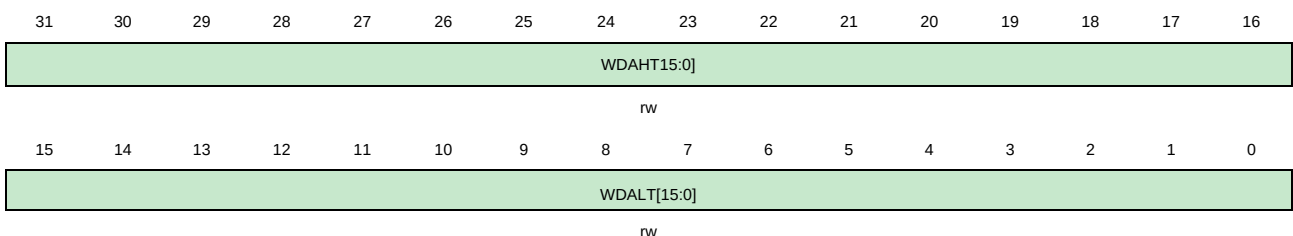
3	WDBIE	WDBEF 中断使能 0: 模拟看门狗 B 中断禁能 1: 模拟看门狗B中断使能
2	WDBEN	模拟看门狗B使能 0: 模拟看门狗 B 禁能 1: 模拟看门狗B使能
1	WDAIE	模拟看门狗 A 中断使能 当该位置 1 时，如果 ADC_WDACHSTAT 寄存器中的任意位为 1，则会产生中断请求。 0: 模拟看门狗 A 中断禁能 1: 模拟看门狗A中断使能
0	WDAEN	模拟看门狗A使能 0: 模拟看门狗 A 禁能 1: 模拟看门狗A使能

15.6.23. 模拟看门狗 A 阈值寄存器（ADC_WDATHOLD）

地址偏移：0x104

复位值：0xFFFF 0000

该寄存器只能按字(32 位)访问。



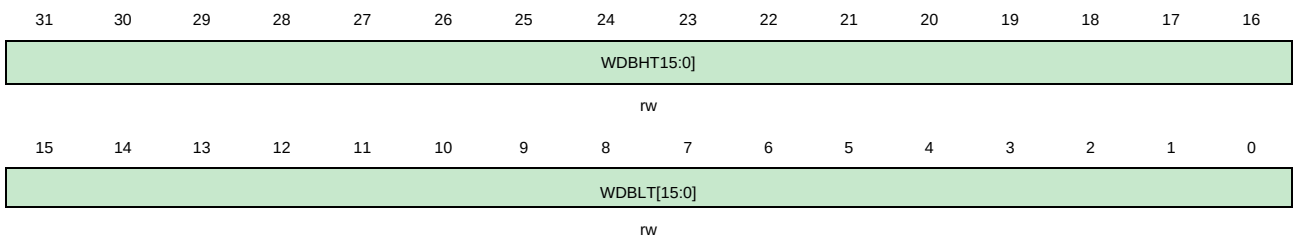
位/位域	名称	描述
31:16	WDAHT[15:0]	模拟看门狗 A 高阈值 这些位定义了模拟看门狗A的高阈值。
15:0	WDALT[15:0]	模拟看门狗 A 低阈值 这些位定义了模拟看门狗A的低阈值。

15.6.24. 模拟看门狗 B 阈值寄存器（ADC_WDATHOLD）

地址偏移：0x108

复位值：0xFFFF 0000

该寄存器只能按字(32 位)访问。



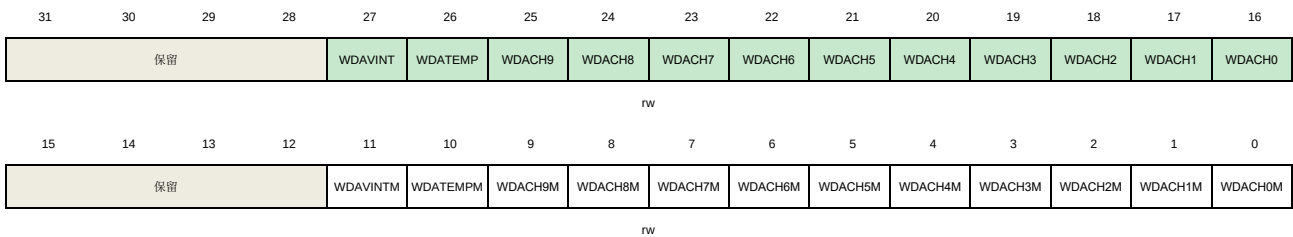
位/位域	名称	描述
31:16	WDBHT[15:0]	模拟看门狗 B 高阈值 这些位定义了模拟看门狗B的高阈值。
15:0	WDBLT[15:0]	模拟看门狗 B 低阈值 这些位定义了模拟看门狗B的低阈值。

15.6.25. 模拟看门狗 A 通道配置寄存器 (ADC_WDACHCFG)

地址偏移: 0x10C

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:28	保留	必须保持复位值
27	WDAVINT	模拟看门狗A选择内部参考电压通道 0: 不选择这个通道 1: 选择则这个通道 注意: 该位仅用在ADC2中。
26	WDATMP	模拟看门狗A选择温度传感器通道 0: 不选择这个通道 1: 选择则这个通道 注意: 该位仅用在ADC2中。
25	WDACH9	模拟看门狗A选择ADCx_IN9 0: 不选择这个通道 1: 选择则这个通道 注意: 该位仅用在ADC2中。
24	WDACH8	模拟看门狗A选择ADCx_IN8

		0: 不选择这个通道 1: 选择则这个通道 注意: 该位仅用在ADC2中。
23	WDACH7	模拟看门狗A选择ADCx_IN7 0: 不选择这个通道 1: 选择则这个通道 注意: 该位仅用在ADC2中。
22	WDACH6	模拟看门狗A选择ADCx_IN6 0: 不选择这个通道 1: 选择则这个通道
21	WDACH5	模拟看门狗A选择ADCx_IN5 0: 不选择这个通道 1: 选择则这个通道
20	WDACH4	模拟看门狗A选择ADCx_IN4 0: 不选择这个通道 1: 选择则这个通道
19	WDACH3	模拟看门狗A选择ADCx_IN3 0: 不选择这个通道 1: 选择则这个通道
18	WDACH2	模拟看门狗A选择ADCx_IN2 0: 不选择这个通道 1: 选择则这个通道
17	WDACH1	模拟看门狗A选择ADCx_IN1 0: 不选择这个通道 1: 选择则这个通道
16	WDACH0	模拟看门狗A选择ADCx_IN0 0: 不选择这个通道 1: 选择则这个通道
15:12	保留	必须保持复位值
11	WDAVINTM	模拟看门狗A中内部参考电压通道比较条件模式 参考WDACH0M。 注意: 该位仅用在ADC2中。
10	WDATMPM	模拟看门狗A中温度传感器通道比较条件模式 参考WDACH0M。 注意: 该位仅用在ADC2中。
9	WDACH9M	模拟看门狗A中通道ADCx_IN9比较条件模式 参考WDACH0M。

注意：该位仅用在ADC2中。

8	WDACH8M	模拟看门狗A中通道ADCx_IN8比较条件模式 参考WDACH0M。 注意： 该位仅用在ADC2中。
7	WDACH7M	模拟看门狗A中通道ADCx_IN7比较条件模式 参考WDACH0M。 注意： 该位仅用在ADC2中。
6	WDACH6M	模拟看门狗A中通道ADCx_IN6比较条件模式 参考WDACH0M。
5	WDACH5M	模拟看门狗A中通道ADCx_IN5比较条件模式 参考WDACH0M。
4	WDACH4M	模拟看门狗A中通道ADCx_IN4比较条件模式 参考WDACH0M。
3	WDACH3M	模拟看门狗A中通道ADCx_IN3比较条件模式 参考WDACH0M。
2	WDACH2M	模拟看门狗A中通道ADCx_IN2比较条件模式 参考WDACH0M。
1	WDACH1M	模拟看门狗A中通道ADCx_IN1比较条件模式 参考WDACH0M。
0	WDACH0M	模拟看门狗A中通道ADCx_IN0比较条件模式 当WINEN位为0，比较比较条件为： 0: WDALT[15:0] > A/D转换值 1: WDALT[15:0] < A/D转换值 当WINEN位为1，比较比较条件为： 0: (A/D转换值 < WDALT[15:0]) 或者 (WDAHT[15:0] < A/D转换值) 1: WDALT[15:0] < A/D转换值 < WDAHT[15:0]

15.6.26. 模拟看门狗 A 通道状态寄存器（ADC_WDACHSTAT）

地址偏移：0x110

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				VINTCMPF	TEMPCMPF	CH9CMPF	CH8CMPF	CH7CMPF	CH6CMPF	CH5CMPF	CH4CMPF	CH3CMPF	CH2CMPF	CH1CMPF	CH0CMPF
				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

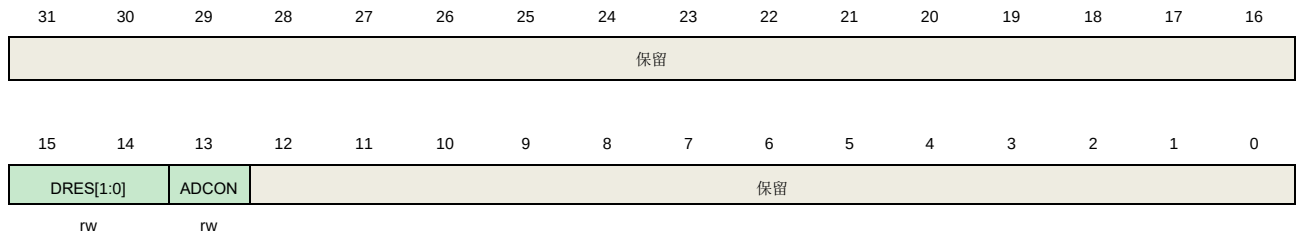
位/位域	名称	描述
31:12	保留	必须保持复位值
11	VINTCMPF	模拟看门狗A中的内部参考电压通道比较状态 0: 不满足比较条件 1: 满足比较条件 注意: 该位仅用在ADC2中。
10	TEMPCMPF	模拟看门狗A中的温度传感器通道比较状态 0: 不满足比较条件 1: 满足比较条件 注意: 该位仅用在ADC2中。
9	CH9CMPF	模拟看门狗A中的ADCx_IN9通道比较状态 参考CH0CMPF的描述。 注意: 该位仅用在ADC2中。
8	CH8CMPF	模拟看门狗A中的ADCx_IN8通道比较状态 参考CH0CMPF的描述。 注意: 该位仅用在ADC2中。
7	CH7CMPF	模拟看门狗A中的ADCx_IN7通道比较状态 参考CH0CMPF的描述。 注意: 该位仅用在ADC2中。
6	CH6CMPF	模拟看门狗A中的ADCx_IN6通道比较状态 参考CH0CMPF的描述。
5	CH5CMPF	模拟看门狗A中的ADCx_IN5通道比较状态 参考CH0CMPF的描述。
4	CH4CMPF	模拟看门狗A中的ADCx_IN4通道比较状态 参考CH0CMPF的描述。
3	CH3CMPF	模拟看门狗A中的ADCx_IN3通道比较状态 参考CH0CMPF的描述。
2	CH2CMPF	模拟看门狗A中的ADCx_IN2通道比较状态 参考CH0CMPF的描述。
1	CH1CMPF	模拟看门狗A中的ADCx_IN1通道比较状态 参考CH0CMPF的描述。
0	CH0CMPF	模拟看门狗A中的ADCx_IN0通道比较状态 0: 不满足比较条件 1: 满足比较条件

15.6.27. 控制寄存器 1 (ADC_CTL1)

地址偏移: 0x150

复位值: 0x0000 1000

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
15:14	DRES[1:0]	ADC 数据分辨率 00: 12 位 01: 10 位 10: 8 位 11: 6 位
13	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对该位写'1'，将开启转换。 0: 禁能ADC并掉电 1: 使能 ADC
12:0	保留	必须保持复位值

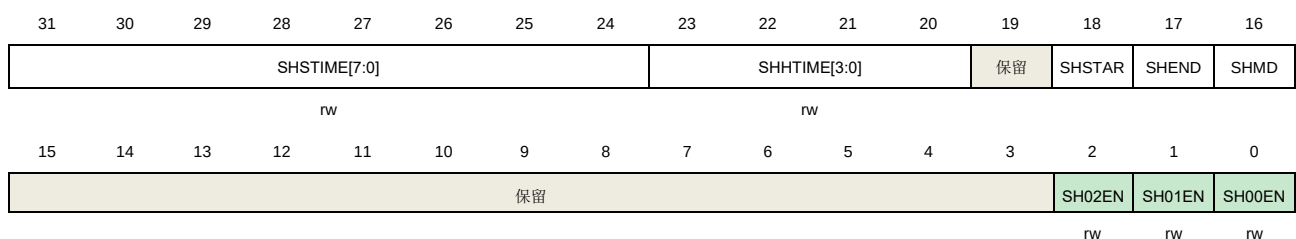
15.6.28. 采样保持控制寄存器 (ADC_SHCTL)

地址偏移: 0x160

复位值: 0x0880 0000

该寄存器仅用在 ADC0 中。

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:24	SHTIME[7:0]	采保电路的采样时间 通道采样保持电路的采样时间为 SHSTIME[7:0]个 CLK_ADC。 注意： SHSTIME[7:0]推荐配置值不小于为8。
23:20	SHHTIME[3:0]	采保电路的保持时间 通道采样保持电路的保持时间为 SHHTIME[3:0]个 CLK_ADC。 注意： SHHTIME[3:0]推荐配置值不小于为 8。
19	保留	必须保持复位值
18	SHSTART	在恒采样模式中软件开启采保 该位仅在SHMD为1时有效。 通过软件将此位设置为1以开启采样保持电路工作。 在采样保持电路开始工作后由硬件清除。
17	SHEND	在恒采样模式中软件结束采保 该位仅在SHMD为1时有效。 通过软件将此位设置为1以结束采样保持电路工作。 在采样保持电路停止工作后由硬件清除。
16	SHMD	通道专用采样保持电路运行模式 0：禁用通道专用采样保持电路的恒采样模式 1：启用通道专用采样保持电路的恒采样模式
15:3	保留	必须保持复位值
2	SH02EN	使能 ADCx_IN2 通道采样保持电路 该位使能 ADCx_IN2 通道采样保持电路。 0：禁能通道采样保持电路 1：使能通道采样保持电路
1	SH01EN	使能 ADCx_IN1 通道采样保持电路 该位使能 ADCx_IN1 通道采样保持电路。 0：禁能通道采样保持电路 1：使能通道采样保持电路
0	SH00EN	使能 ADCx_IN0 通道采样保持电路 该位使能 ADCx_IN0 通道采样保持电路。 0：禁能通道采样保持电路 1：使能通道采样保持电路

15.6.29. DMA 控制寄存器（ADC_DMACTL）

地址偏移：0x170

复位值：0x0000 0000

该寄存器只能按字(32位)访问。

DMAEN4	GP4DDM	GP4OVRF	OVREN4	GP4OVRIE	保留			DMAEN3	GP3DDM	GP3OVRF	OVREN3	GP3OVRIE	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
DMAEN2	GP2DDM	GP2OVRF	OVREN2	GP2OVRIE	保留			DMAEN1	GP1DDM	GP1OVRF	OVREN1	GP1OVRIE	保留		
rw	rw	rw	rw	rw				rw	rw	rw	rw	rw			

位/位域	名称	描述
31	DMAEN4	使能Group_pri4组DMA请求 0: 禁能DMA请求 1: 使能DMA请求
30	GP4DDM	Group_pri4组DMA失能模式 该位配置单ADC的DMA失能模式。 0: 在检测到DMA控制器的传输结束信号之后，禁能DMA传输。 1: 当 DMAEN4=1, 在每次 Group_pri4 组转换结束后产生 DMA 请求。
29	GP4OVRF	Group_pri4 组 DMA 溢出标志 0: 无溢出 1: DMA 溢出 当 DMA 溢出时，硬件置位该位。 软件写 0 清除该位。
28	OVREN4	使能 Group_pri4 组溢出检测 0: 禁能溢出检测 1: 使能溢出检测
27	GP4OVRIE	使能 Group_pri4 组 DMA 溢出检测中断 0: 禁能溢出中断 1: 使能溢出中断
26:24	保留	必须保持复位值
23	DMAEN3	使能Group_pri3组DMA请求 0: 禁能DMA请求 1: 使能DMA请求
22	GP3DDM	Group_pri3组DMA失能模式 该位配置单ADC的DMA失能模式。 0: 在检测到DMA控制器的传输结束信号之后，禁能DMA传输。 1: 当DMAEN3=1, 在每次Group_pri3组转换结束后产生DMA请求。
21	GP3OVRF	Group_pri3 组 DMA 溢出标志 0: 无溢出 1: DMA 溢出 当 DMA 溢出时，硬件置位该位。 软件写 0 清除该位。

20	OVREN3	使能 Group_pri3 组溢出检测 0: 禁能溢出检测 1: 使能溢出检测
19	GP3OVRIE	使能 Group_pri3 组 DMA 溢出检测中断 0: 禁能溢出中断 1: 使能溢出中断
18:16	保留	必须保持复位值
15	DMAEN2	使能Group_pri2组DMA请求 0: 禁能DMA请求 1: 使能 DMA 请求
14	GP2DDM	Group_pri2组DMA失能模式 该位配置单ADC的DMA失能模式。 0: 在检测到DMA控制器的传输结束信号之后，禁能DMA传输。 1: 当 DMAEN2=1, 在每次 Group_pri2 组转换结束后产生 DMA 请求。
13	GP2OVRF	Group_pri2 组 DMA 溢出标志 0: 无溢出 1: DMA 溢出 当 DMA 溢出时，硬件置位该位。 软件写 0 清除该位。
12	OVREN2	使能 Group_pri2 组溢出检测 0: 禁能溢出检测 1: 使能溢出检测
11	GP2OVRIE	使能 Group_pri2 组 DMA 溢出检测中断 0: 禁能溢出中断 1: 使能溢出中断
10:8	保留	必须保持复位值
7	DMAEN1	使能Group_pri1组DMA请求 0: 禁能DMA请求 1: 使能 DMA 请求
6	GP1DDM	Group_pri1组DMA失能模式 该位配置单ADC的DMA失能模式。 0: 在检测到DMA控制器的传输结束信号之后，禁能DMA传输。 1: 当 DMAEN1=1, 在每次 Group_pri1 组转换结束后产生 DMA 请求。
5	GP1OVRF	Group_pri1 组 DMA 溢出标志 0: 无溢出 1: DMA 溢出 当 DMA 溢出时，硬件置位该位。

软件写 0 清除该位。

4	OVREN1	使能 Group_pri1 组溢出检测 0: 禁能溢出检测 1: 使能溢出检测
3	GP1OVRIE	使能 Group_pri1 组 DMA 溢出检测中断 0: 禁能溢出中断 1: 使能溢出中断
2:0	保留	必须保持复位值

15.6.30. 分叉触发控制寄存器 (ADC_BITRGCTL)

地址偏移: 0x180

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
GP4BICHSEL[4:0]					GP4BIMEN	GP4BITGRS	保留	GP3BICHSEL[4:0]					GP3BIMEN	GP3BITGRS	保留
rw					rw	rw		rw					rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP2BICHSEL[4:0]					GP2BIMEN	GP2BITGRS	保留	GP1BICHSEL[4:0]					GP1BIMEN	GP1BITGRS	保留
rw					rw	rw		rw					rw	rw	

位/位域	名称	描述
31:27	GP4BICHSEL[4:0]	优先级 4 组分叉触发通道选择 参考 GP1BICHSEL[4:0]。 当 PROCF 位为 0 时可以配置这些位。
26	GP4BIMEN	优先级 4 组分叉触发模式使能 当 PROCF 位为 0 时可以配置该位。 0: 禁能分叉触发模式 1: 使能分叉触发模式 注意: 1) 当 GSCAN[1:0]为 2'b10, 要禁能分叉触发模式。 2) 分叉触发模式不适用自校准。
25	GP4BITGRS	使能优先级 4 组下次触发恢复功能 该为仅在优先级 4 组分叉触发模式使能时有效 (GP4BIMEN = 1)。 当该位置 1, 对于在当前 A/D 转换轮次期间发生的下一次触发, 将会在当前 A/D 转换结束后恢复转换。否则, 该触发将会被忽略。 0: 忽略下一次触发 1: 恢复下一次触发
24	保留	必须保持复位值。

23:19	GP3BICHSEL[4:0]	优先级 3 组分叉触发通道选择 参考 GP1BICHSEL[4:0]。 当 PROCF 位为 0 时可以配置这些位。
18	GP3BIMEN	优先级 3 组分叉触发模式使能 当 PROCF 位为 0 时可以配置该位。 0: 禁能分叉触发模式 1: 使能分叉触发模式 注意: 1) 当 GSCAN[1:0]为 2'b10, 要禁能分叉触发模式。 2) 分叉触发模式不适用自校准。
17	GP3BITGRS	使能优先级 3 组下次触发恢复功能 该为仅在优先级 3 组分叉触发模式使能时有效 (GP4BIMEN =1)。 当该位置 1, 对于在当前 A/D 转换轮次期间发生的下一次触发, 将会在当前 A/D 转换结束后恢复转换。否则, 该触发将会被忽略。 0: 忽略下一次触发 1: 恢复下一次触发
16	保留	必须保持复位值。
15:11	GP2BICHSEL[4:0]	优先级 2 组分叉触发通道选择 参考 GP1BICHSEL[4:0]。 当 PROCF 位为 0 时可以配置这些位。
10	GP2BIMEN	优先级 2 组分叉触发模式使能 当 PROCF 位为 0 时可以配置该位。 0: 禁能分叉触发模式 1: 使能分叉触发模式 注意: 1) 当 GSCAN[1:0]为 2'b10, 要禁能分叉触发模式。 2) 分叉触发模式不适用自校准。
9	GP2BITGRS	使能优先级 2 组下次触发恢复功能 该为仅在优先级 2 组分叉触发模式使能时有效 (GP4BIMEN =1)。 当该位置 1, 对于在当前 A/D 转换轮次期间发生的下一次触发, 将会在当前 A/D 转换结束后恢复转换。否则, 该触发将会被忽略。 0: 忽略下一次触发 1: 恢复下一次触发
8	保留	必须保持复位值。
7:3	GP1BICHSEL[4:0]	优先级1组分叉触发通道选择 当 PROCF 位为 0 时可以配置该位。 00000: ADCx_IN0 00001: ADCx_IN1 00010: ADCx_IN2

		00011: ADCx_IN3
		00100: ADCx_IN4
		00101: ADCx_IN5
		00110: ADCx_IN6
		00111: ADCx_IN7
		01000: ADCx_IN8
		01001: ADCx_IN9
		01101: 温度传感器通道
		01110: 内部参考电压通道
		Others: 保留
		注意:
		1) 这些位仅在分叉触发模式使能 (BIMEN=1) 时有效。
		2) 当分叉触发模式使能 (BIMEN=1), ADC_CHSEL0 和 ADC_CHSEL1 所选择的通道无效。
2	GP1BIMEN	优先级 1 组分叉触发模式使能 当 PROCF 位为 0 时可以配置该位。 0: 禁能分叉触发模式 1: 使能分叉触发模式 注意: 1) 当 GSCAN[1:0]为 2'b10, 要禁能分叉触发模式。 2) 分叉触发模式不适用自校准。
1	GP1BITGRS	使能优先级 1 组下次触发恢复功能 该为仅在优先级 1 组分叉触发模式使能时有效 (GP4BIMEN =1)。 当该位置 1, 对于在当前 A/D 转换轮次期间发生的下一次触发, 将会在当前 A/D 转换结束后恢复转换。否则, 该触发将会被忽略。 0: 忽略下一次触发 1: 恢复下一次触发
0	保留	必须保持复位值。

16. 数模转换器（DAC）

16.1. 简介

数字/模拟转换器可以将 12 位的数字数据转换为外部引脚上的电压输出。数据可以采用左对齐或右对齐模式。该电压可用于比较器参考输入和模拟输出。

每个 DAC 的两个通道可以独立或并发工作。

16.2. 主要特性

DAC 的主要特征如下：

- 12 位分辨率；
- 数据左对齐或右对齐；
- 同步更新转换；
- 外部事件触发转换；
- 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）；
- DACx 双通道并发模式；
- 比较器参考电压生成；
- 输出可以连接到模拟引脚；
- 通过 EVIC 信号使能 DAC 转换；

[图 16-1. DAC 结构框图](#)为 DAC 的结构框图，[表 16-1. DAC 引脚](#)给出了引脚描述。

图 16-1. DAC 结构框图

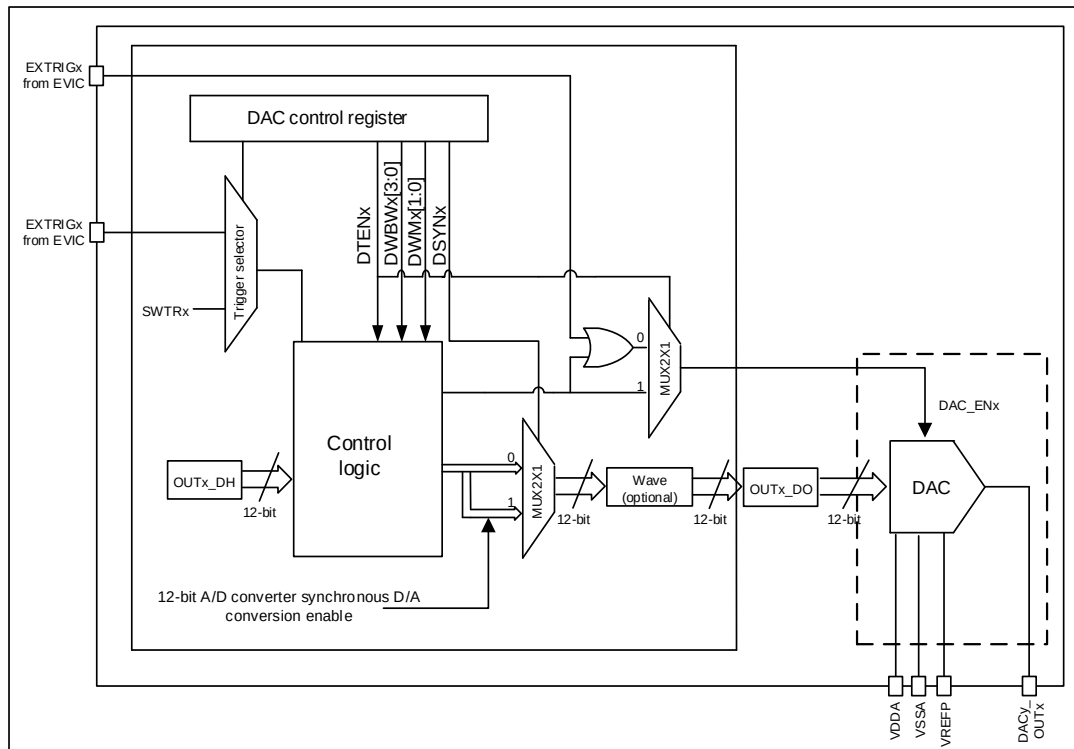


表 16-1. DAC 引脚

名称	描述	信号类型
VDDA	模拟电源	输入，模拟电源
VSSA	模拟电源地	输入，模拟电源地
VREFP	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 16-2. DAC 触发与输出

	DAC0	
通道	通道 0	通道 1
DAC 输出 I/O	PD4	PD5
软件触发功能	•	
EVIC 触发功能	•	

注意：在使能 DAC 模块前，GPIO 口（DAC 输出 I/O）应配置为模拟模式。

16.3. 功能描述

16.3.1. DAC 使能

将 DAC_CTL0 寄存器中的 DENx 位置 1 或者 EVIC 触发事件发生，可以给 DAC 模块上电，

DAC 子模块完全启动需要等待 t_{WAKEUP} 时间。

注意：EVIC 触发事件使能 DAC 仅在 DTENx 位为 1'b0 时可用。

16.3.2. 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 和 DAC_OUTx_L12DH 中的任意一个寄存器写入数据来配置。

16.3.3. 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 16-3. DAC 外部触发](#) 所示。

表 16-3. DAC 外部触发

DTSELx[1:0]	触发源	触发类型
2b'00	EXTRIG	硬件触发
2b'01	SWTR	软件触发
2b'1x	保留	保留

外部触发信号 (EXTRIG) 由事件互连单元 (EVIC) 产生，而软件触发是通过设置 DAC_SWT 寄存器的 SWTRx 位生成的。

16.3.4. DAC 转换

DAC 有两种转换模式，触发转换发生在 DTENx= 1'b1 (在 DAC_CTL 寄存器中)，自动转换发生在 DTENx= 1'b0。如 [表 16-4. DAC 转换模式](#) 所示。

表 16-4. DAC 转换模式

DTENx	DENx	DTSELx[2:0]	描述
1	0	-	无转换
	1	00	外部触发转换
		01	软件触发转换
		1x	无转换
0	0	-	当 EVIC 触发事件发生时自动转换
	1	-	自动转换

触发转换

如果使能了外部触发 (通过设置 DAC_CTL0 寄存器的 DTENx 位为 1)，且使能了 DAC (通过设置 DAC_CTL0 寄存器的 DENx 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)。

自动转换

当外部触发未使能时（通过设置 DAC_CTL0 寄存器的 DTENx 位为 0），且使能了 DAC（通过设置 DAC_CTL0 寄存器的 DENx 位），转移将被自动执行。

当外部触发未使能时（通过设置 DAC_CTL0 寄存器的 DTENx 位为 0），且 EVIC 触发事件发生使能了 DAC，转移将被自动执行。

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

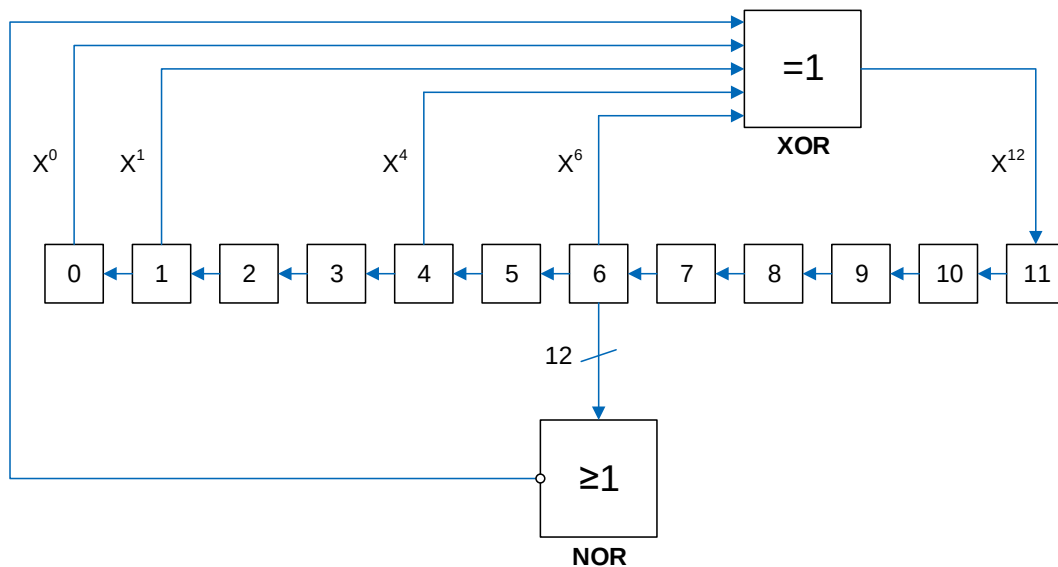
16.3.5. DAC 噪声波

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。

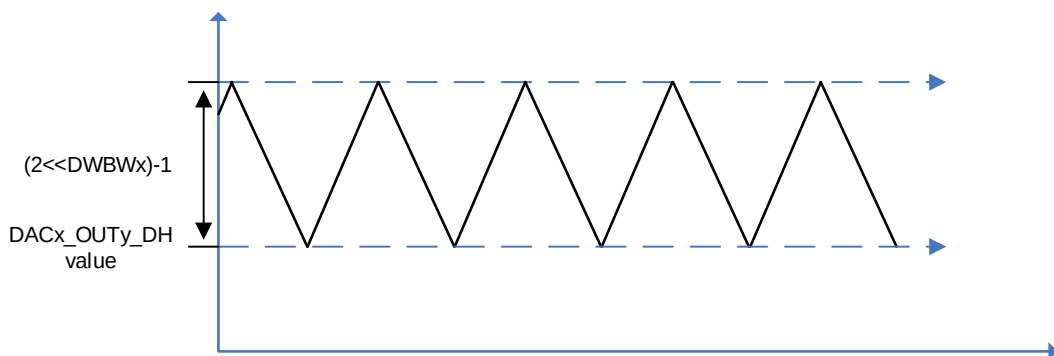
注意： X^n 的初始值是 0xAAA（ $n=0$ 到 11）。

图 16-2. 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 16-3. DAC 三角噪声模式生成的波形



16.3.6. DAC 输出电压

DAC 引脚上的模拟输出电压取决于下面的等式：

$$V_{DAC\_OUT} = V_{REFP} * OUTx\_DO / 4096 \quad (16-1)$$

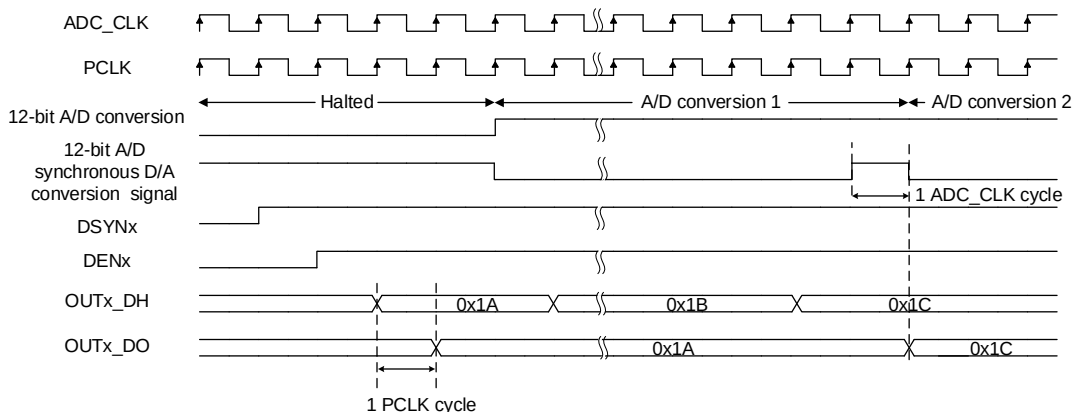
数字输入被线性地转换成模拟输出电压，输出范围为 0 到 V_{REFP} 。

16.3.7. D/A 和 A/D 防干扰

当 D/A 转换开始时，DAC 产生浪涌电流。由于 12 位 D/A 转换器和 12 位 A/D 转换器 (ADC2) 共用同一个模拟电源，因此浪涌电流可能会干扰 ADC2 的正常工作。

当 DAC_CTL0 寄存器的 DSYNx 位为 1 时，即使在 ADC2 转换期间修改了数据保存寄存器 (DAC_OUT0_R12DH, DAC_OUT0_L12DH, DAC_OUT1_R12DH, DAC_OUT1_L12DH, DACC_L12DH 和 DACC_R12DH)，D/A 转换不会立即开始，而是随着 A/D 转换完成同步开始。

图 16-4. D/A 和 A/D 防干扰



16.3.8. DAC 并发转换

当 DAC 的两个通道同时工作时，为了在特定应用中最大限度利用总线带宽，DAC 的两个通道可以被配置为并发模式。在并发模式中，DAC 的 DAC_OUTx_DH 和 DAC_OUTx_DO 值将同

时被更新。

有 2 个并发寄存器可用于加载 `OUTx_DH` 的值，分别是：`DACC_R12DH` 和 `DACC_L12DH` 寄存器，配置其中的任意一个寄存器都可实现同时驱动两个 DAC。

当使能了外部触发时，DAC 两个通道的 `DTENx` 位都需要置 1，需要配置 `DTSEL0/1` 相同来保证同时触发。

噪声模式和噪声位宽可以根据使用情况配置为相同或不同。

16.3.9. DAC 低功耗模式

睡眠模式

在睡眠模式中，DAC 可正常工作。

待机模式

在待机模式中，DAC 停止工作，退出待机模式并重新初始化 DAC，DAC 可再次工作。

16.4. DAC 寄存器

DAC0 基地址: 0x4000 7400

16.4.1. DACx 控制寄存器 0 (DAC_CTL0)

偏移地址: 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
DOCMPR EF1	DCSEL1	保留	DSYN1	DWBW1[3:0]				DWM1[1:0]		保留	DTSEL1[1:0]		DTEN1	保留	DEN1
rw	rw		rw	rw				rw			rw		rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DOCMPR EF0	DCSEL0	保留	DSYN0	DWBW0[3:0]				DWM0[1:0]		保留	DTSEL0[1:0]		DTEN0	保留	DEN0
rw	rw		rw	rw				rw			rw		rw		rw

位/位域	名称	描述
31	DOCMPREF1	DACx_OUT1输出作为CMP参考电压 0: DACx_OUT1不用作CMP的参考电压 1: DACx_OUT1 用作 CMP 的参考电压
30	DCSEL1	DACx_OUT1 连接选择 0: DACx_OUT1 不连接到外部引脚 1: DACx_OUT1 连接到外部引脚
29	保留	必须保持复位值。
28	DSYN1	DACx_OUT1 与 ADC2 同步转换。防止 DAC 和 ADC2 转换之间的干扰。 0: DACx_OUT1 与 ADC2 不同步转换 1: DACx_OUT1 与 ADC2 同步转换
27:24	DWBW1[3:0]	DACx_OUT1 噪声波位宽 这些位指定了 DACx_OUT1 的噪声波信号的位宽。LFSR 噪声模式下, 这些位表示不屏蔽 LFSR 的位[n-1, 0]; 三角噪声模式下, 这些位表示三角波幅值为 $(2^{n-1})-1$ 。其中, n 为噪声波位宽。 0000: 波形信号的位宽为 1 0001: 波形信号的位宽为 2 0010: 波形信号的位宽为 3 0011: 波形信号的位宽为 4 0100: 波形信号的位宽为 5 0101: 波形信号的位宽为 6 0110: 波形信号的位宽为 7

		0111: 波形信号的位宽为 8 1000: 波形信号的位宽为 9 1001: 波形信号的位宽为 10 1010: 波形信号的位宽为 11 ≥1011: 波形信号的位宽为 12
23:22	DWM1[1:0]	DACx_OUT1 噪声波模式 这些位指定了在 DACx_OUT1 外部触发使能 (DTEN1=1) 的情况下, DACx_OUT1 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
21	保留	必须保持复位值。
20:19	DTSEL1[1:0]	DACx_OUT1 触发选择 这些位仅在 DTEN=1 并选择用于触发 DAC 的外部事件时使用。 00: EXTRIG (外部触发来自 EVIC) 01: 软件触发 其他值: 保留
18	DTEN1	DACx_OUT1 触发使能 0: DACx_OUT1 触发禁能 1: DACx_OUT1 触发使能
17	保留	必须保持复位值。
16	DEN1	DACx_OUT1 使能 该位可以被软件或者 EVIC 触发事件置 1 0: DACx_OUT1 禁能 1: DACx_OUT1 使能
15	DOCMPREF0	DACx_OUT0 输出作为 CMP 参考电压 0: DACx_OUT0 不用作 CMP 的参考电压 1: DACx_OUT0 用作 CMP 的参考电压
14	DCSEL0	DACx_OUT0 连接选择 0: DACx_OUT0 不连接到外部引脚 1: DACx_OUT0 连接到外部引脚
13	保留	必须保持复位值。
12	DSYN0	DACx_OUT0 与 ADC2 同步转换。防止 DAC 和 ADC2 转换之间的干扰。 0: DACx_OUT0 与 ADC2 不同步转换 1: DACx_OUT0 与 ADC2 同步转换
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
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 DTEN=1 并选择用于触发 DAC 的外部事件时使用。 00：EXTRIG（外部触发来自 EVIC） 01：软件触发 其他值：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	保留	必须保持复位值。
0	DEN0	DACx_OUT0 使能 该位可以被软件或者 EVIC 触发事件置 1 0：DACx_OUT0 禁能 1：DACx_OUT0 使能

16.4.2. 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。

保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留													SWTR1	SWTR0	
													W	W	

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	SWTR1	DACx_OUT1 软件触发，在下一个时钟由硬件清除。 0: 软件触发禁能 1: 软件触发使能
0	SWTR0	DACx_OUT0 软件触发，在下一个时钟由硬件清除。 0: 软件触发禁能 1: 软件触发使能

16.4.3. 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。

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				OUT0_DH[11:0]											
rw															

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。

16.4.4. 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。

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
OUT0_DH[11:0]												保留			
rw															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:4	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。

16.4.5. DACx_OUT1 12 位右对齐数据保持寄存器 (DAC_OUT1_R12DH)

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				OUT1_DH[11:0]											
rw															

位/位域	名称	描述
31:12	保留	必须保持复位值。
11:0	OUT1_DH[11:0]	DACx_OUT1 12 位右对齐数据 这些位指定了将由 DACx_OUT1 转换的数据。

16.4.6. DACx_OUT1 12 位左对齐数据保持寄存器 (DAC_OUT1_L12DH)

地址偏移: 0x18

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OUT1_DH[11:0]												保留			
rw															

位/位域	名称	描述

31:16	保留	必须保持复位值。
15:4	OUT1_DH[11:0]	DACx_OUT1 12 位左对齐数据 这些位指定了将由 DACx_OUT1 转换的数据。
3:0	保留	必须保持复位值。

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

地址偏移: 0x20

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				OUT1_DH[11:0]											
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				OUT0_DH[11:0]											
rw															

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:16	OUT1_DH[11:0]	DACx_OUT1 12 位右对齐数据 这些位指定了将由 DACx_OUT1 转换的数据。
15:12	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。

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

地址偏移: 0x24

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OUT1_DH[11:0]												Reserved			
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OUT0_DH[11:0]												Reserved			
rw															

位/位域	名称	描述
31:20	OUT1_DH[11:0]	DACx_OUT1 12 位左对齐数据

这些位指定了将由 DACx_OUT1 转换的数据。

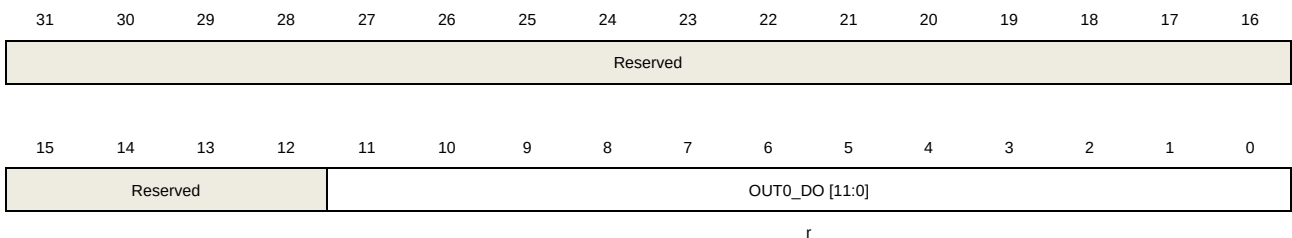
19:16	保留	必须保持复位值。
15:4	OUT0_DH[11:0]	DACx_OUT0 12 位左对齐数据 这些位指定了将由 DACx_OUT0 转换的数据。
3:0	保留	必须保持复位值。

16.4.9. 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。



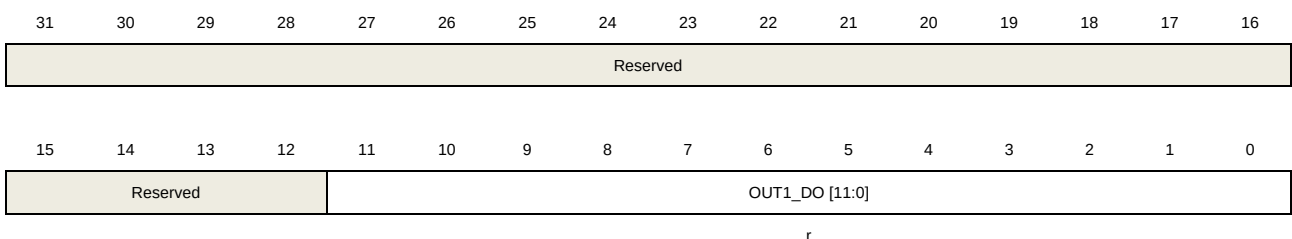
位/位域	名称	描述
31:12	保留	必须保持复位值。
11:0	OUT0_DO [11:0]	DACx_OUT0 数据输出。 这些位为只读类型，存储由 DACx_OUT0 转换的数据。

16.4.10. DACx_OUT1 数据输出寄存器 (DAC_OUT1_DO)

地址偏移: 0x30

复位值: 0x0000 0000

该寄存器只能按字(32 位)访问。



位/位域	名称	描述
31:12	保留	必须保持复位值。
11:0	OUT1_DO [11:0]	DACx_OUT1 数据输出。 这些位为只读类型，存储由 DACx_OUT1 转换的数据。

17. 比较器 (CMP)

17.1. 简介

通用比较器可独立工作，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 TIMER 的触发源，结合 DAC 和 TIMER 的 PWM 输出，可以实现电流控制。

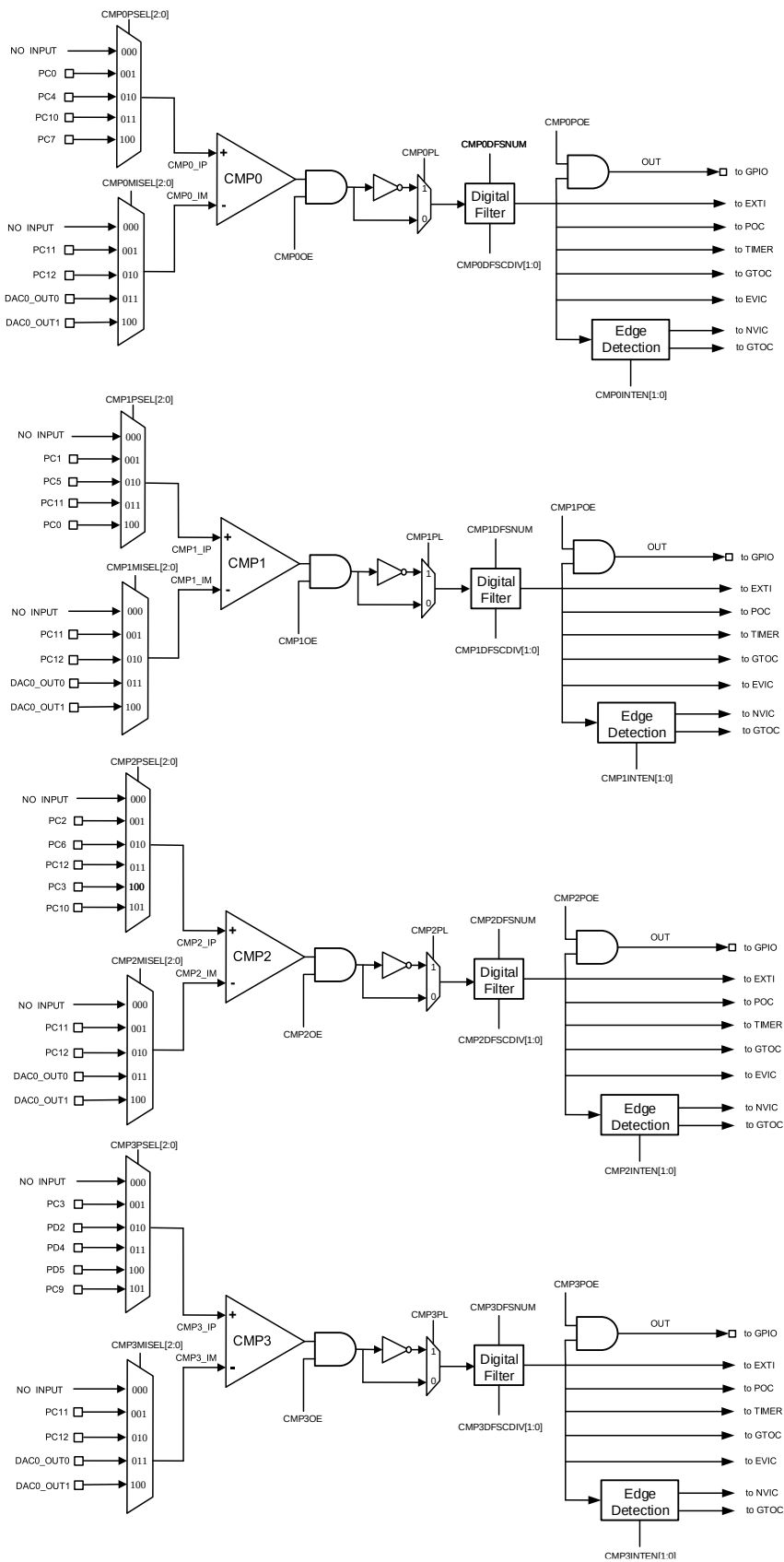
17.2. 主要特征

- 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；
- 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 POC；
- 输出到 GTOC；
- 输出到 EVIC；
- 数字滤波器

17.3. 功能描述

比较器的框图展示如下：

图 17-1. 比较器框图



17.3.1. 比较器时钟

比较器与 APB 总线连接，时钟与 PCLK 同步。

17.3.2. 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。

参考 Datasheet 的引脚定义，比较器输出可以连接到对应的复用 I/O 口。

比较器的输出可同时实现内部和外部输出。连接到极性选择的 CMP 输出可以通过在 CMPx_CS 寄存器中设置 CMPxOE 位来禁用或启用。此外，通过在 CMPx_CS 寄存器中设置 CMPxPOE 位，可以禁用或启用连接到外部引脚的 CMP 输出。

比较器输出内部连接到定时器，他们的连接关系如下：

- CMP 输出连接到定时器输入通道；
- 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 17-1. CMP 的输入和输出总结](#)详细描述了 CMP 的输入和输出。

表 17-1. CMP 的输入和输出总结

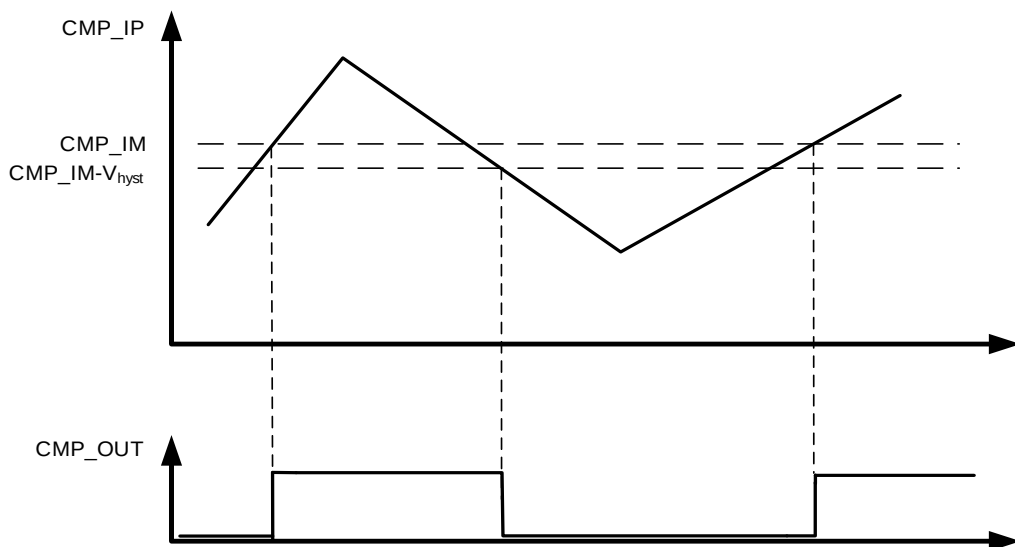
	CMP0	CMP1	CMP2	CMP3
CMP 同相输入连接到 I/O	PC0 PC4 PC10 PC7	PC1 PC5 PC11 PC0	PC2 PC6 PC12 PC3 PC10	PC3 PD2 PD4 PD5 PC9
CMP 反相输入连接到 I/O	PC11 PC12			
CMP 反相输入连接到内部信号	DAC0_OUT0 DAC0_OUT1			
CMP 输出连接到 I/O	PA0	PA1	PB2	PB1
CMP 输出连接到 EXTI	•			
CMP 输出连接到 NVIC	•			
CMP 输出连接到 POC	•			
CMP 输出连接到 GTOC	•			
CMP 输出连接到 EVIC	•			
CMP 输出连接到内部信号	TIMER0、TIMER1、TIMER2、TIMER7			
CMP 输出（电机控制保护）	TIMER0 BREAK TIMER7 BREAK			

注意：有关 CMP 输出连接到内部信号的详细信息，请参阅系统配置寄存器。

17.3.3. 比较器迟滞

为了避免噪声信号所引起的假输出，电路设计了可编程的迟滞功能，通过配置控制状态寄存器来控制迟滞电压值。该功能可以在无需要时关闭。

图 17-2. 比较器迟滞



17.3.4. 比较器寄存器写保护

比较器的控制状态寄存器 (CMPx_CS) 可通过设置 CMPxLK 位为 1 来进行写保护, CMPx_CS 寄存器, 包含 CMPxLK 位, 就会变为只读位, 只有在 MCU 复位时才可以复位。

17.3.5. 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, 用于退出省电模式。

CMP 还可以输出到 NVIC 产生中断。当 CMP 输出有效边沿时, 将生成中断亲请求, 可以通过设置 CMPx_CS 寄存器中的 CMPxINTEN [1:0] 位来选择有效边沿。

17.3.6. 比较器数字滤波

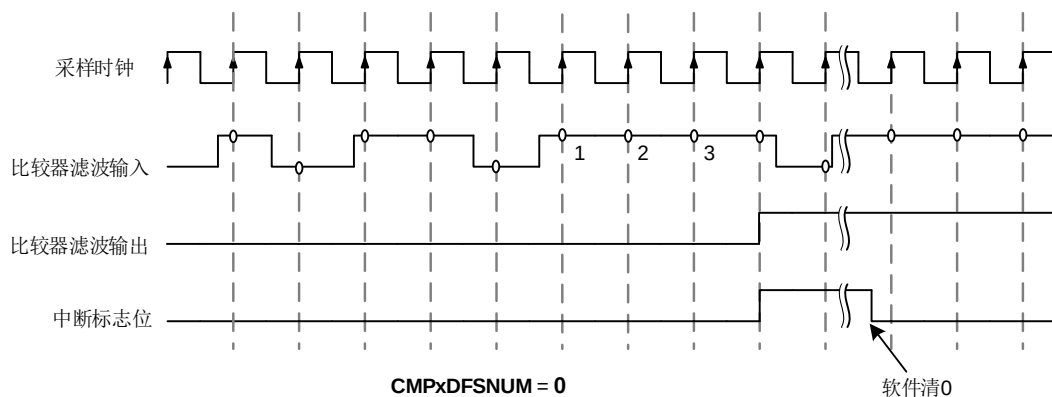
CMPx_CS 寄存器的 CMPxDFSCDIV [1:0] 位决定了数字滤波器的采样频率。并且, 同一电平连续采样的有效次数可以通过 CMPx_CS 寄存器中的 CMPxDFSNUM 位来选择。数字滤波器的特点如下:

- 可选择 CK_CMP/8、CK_CMP/16 或 CK_CMP/32 作为采样时钟。
- 滤波功能是可禁用的。
- 可选择三次或者四次作为连续有效采样数。

CMP 输出信号在每个采样时钟被采样, 当同一电平被连续采样三次或四次时, 该电平在下一

个采样时刻被作为 CMP 滤波输出。图 17-3. 比较器滤波输出展示滤波器输出功能的细节。

图 17-3. 比较器滤波输出



注意：可以通过在 `CMPx_CS` 寄存器中设置 `CMPxDFSCDIV[1:0]` 位和 `CMPxDFSNUM` 位来选择消除噪声的宽度。

17.3.7. 比较器省电模式

睡眠模式

在睡眠模式下，CMP 可以正常工作，可以通过比较器中断/事件退出睡眠模式。

深度睡眠模式

在深度睡眠模式下，CMP 可以正常工作，可以通过 EXTI 中断/事件退出深度睡眠模式。

待机模式

在待机模式下，CMP 寄存器掉电，退出待机模式后必须重新初始化。

17.4. 比较器寄存器

CMP 基地址: 0x4001 7C00

17.4.1. 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。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留												CMP3IF	CMP2IF	CMP1IF	CMP0IF
												r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												CMP3O	CMP2O	CMP1O	CMP0O
												r	r	r	r

位/位域	名称	描述
31:20	保留	必须保持复位值。
19	CMP3IF	CMP3中断标志位 0: 无CMP3中断发生 1: CMP3中断发生 当CMP3输出有效边沿时, 该位由硬件置1。 软件写1至CMP_IFC寄存器CMP3IC位清0。
18	CMP2IF	CMP2中断标志位 0: 无CMP2中断发生 1: CMP2中断发生 当CMP2输出有效边沿时, 该位由硬件置1。 软件写1至CMP_IFC寄存器CMP2IC位清0。
17	CMP1IF	CMP1中断标志位 0: 无CMP1中断发生 1: CMP1中断发生 当CMP1输出有效边沿时, 该位由硬件置1。 软件写1至CMP_IFC寄存器CMP1IC位清0。
16	CMP0IF	CMP0中断标志位 0: 无CMP0中断发生 1: CMP0中断发生 当CMP0输出有效边沿时, 该位由硬件置1。 软件写1至CMP_IFC寄存器CMP0IC位清0。
15:4	保留	必须保持复位值。

3	CMP3O	CMP3 输出 该位反映 CMP3 输出状态，是只读位 0: 同相输入端低于反相输入端，输出为低电平 1: 同相输入端高于反相输入端，输出为高电平
2	CMP2O	CMP2 输出 该位反映 CMP2 输出状态，是只读位 0: 同相输入端低于反相输入端，输出为低电平 1: 同相输入端高于反相输入端，输出为高电平
1	CMP1O	CMP1 输出 该位反映 CMP1 输出状态，是只读位 0: 同相输入端低于反相输入端，输出为低电平 1: 同相输入端高于反相输入端，输出为高电平
0	CMP0O	CMP0 输出 该位反映 CMP0 输出状态，是只读位 0: 同相输入端低于反相输入端，输出为低电平 1: 同相输入端高于反相输入端，输出为高电平

17.4.2. 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。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留												CMP3IC	CMP2IC	CMP1IC	CMP0IC
												W	W	W	W
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
31:20	保留	必须保持复位值。
19	CMP3IC	CMP3 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
18	CMP2IC	CMP2 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
17	CMP1IC	CMP1 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

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。

17.4.3. CMP0 控制状态寄存器 (CMP0_CS)

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
CMP0LK	保留								CMP0PSEL[2:0]			保留	CMP0MISEL[2:0]		
rw								rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	CMP0DFS NUM	保留		CMP0DFSCDIV[1:0]		CMP0HST[1:0]		CMP0INTEN[1:0]		CMP0OE	CMP0POE	CMP0PL	保留		CMP0EN
rw		rw				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
30:23	保留	必须保持复位值。
22:20	CMP0PSEL[2:0]	CMP0_IP 输入选择 该位域用于选择 CMP0 的 CMP0_IP 输入源。 000: 无输入 001: PC0 010: PC4 011: PC10 100: PC7 其他: 保留。
19	保留	必须保持复位值。
18:16	CMP0MISEL[2:0]	CMP0_IM 输入选择 该位域用于选择 CMP0 的 CMP0_IM 输入源。 000: 无输入 001: PC11 010: PC12 011: DAC0_OUT0

		100: DAC0_OUT1 其他: 保留。
15	保留	必须保持复位值。
14	CMP0DFSNUM	CMP0 数字滤波器采样次数 该位用于选择 CMP0 数字滤波器有效采样次数。 0: 连续采样三次 1: 连续采样四次
13:12	保留	必须保持复位值。
11:10	CMP0DFSCDIV[1:0]	CMP0 数字滤波器采样时钟分频 该位域用于选择 CMP0 数字滤波器采样频率。 00: 数字滤波器禁能 01: 采样频率为 $f_{CK\_CMP}/8$ 10: 采样频率为 $f_{CK\_CMP}/16$ 11: 采样频率为 $f_{CK\_CMP}/32$
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
7:6	CMP0INTEN[1:0]	CMP0 中断使能 该位域用于选择 CMP0 输出的有效边沿来产生中断请求。 00: 中断禁能 01: 上升沿 10: 下降沿 11: 双边沿 注意: 该位域也用于为 CMP0IF 选择有效边沿。
5	CMP0OE	CMP0 输出使能 0: 禁能 1: 使能
4	CMP0POE	CMP0 引脚输出使能 0: 禁能 1: 使能
3	CMP0PL	CMP0 输出极性 该位用于控制 CMP0 输出极性。 0: 输出是正相的 1: 输出是反相的
2:1	保留	必须保持复位值。

0	CMP0EN	CMP0 使能 0: CMP0 禁能 1: CMP0 使能

17.4.4. CMP1 控制状态寄存器 (CMP1_CS)

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
CMP1LK	保留								CMP1PSEL[2:0]			保留	CMP1MISEL[2:0]		
rw									rw				rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	CMP1DFS NUM	保留		CMP1DFSCDIV[1:0]		CMP1HST[1:0]		CMP1INTEN[1:0]		CMP1OE	CMP1POE	CMP1PL	保留		CMP1EN
rw		rw				rw		rw		rw	rw	rw	rw		

位/位域	名称	描述
31	CMP1LK	CMP1 写保护 该位可将 CMP1 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: CMP1_CS 是可读可写位 1: CMP1_CS 是只读位
30:23	保留	必须保持复位值。
22:20	CMP1PSEL[2:0]	CMP1_IP 输入选择 该位域用于选择 CMP1 的 CMP1_IP 输入源。 000: 无输入 001: PC1 010: PC5 011: PC11 100: PC0 其他: 保留。
19	保留	必须保持复位值。
18:16	CMP1MISEL[2:0]	CMP1_IM 输入选择 该位域用于选择 CMP1 的 CMP1_IM 输入源。 000: 无输入 001: PC11 010: PC12 011: DAC0_OUT0 100: DAC0_OUT1

		其他：保留。
15	保留	必须保持复位值。
14	CMP1DFSNUM	CMP1 数字滤波器采样次数 该位用于选择 CMP1 数字滤波器有效采样次数。 0：连续采样三次 1：连续采样四次
13:12	保留	必须保持复位值。
11:10	CMP1DFSCDIV[1:0]	CMP1 数字滤波器采样时钟分频 该位域用于选择 CMP1 数字滤波器采样频率。 00：数字滤波器禁能 01：采样频率为 $f_{CK\_CMP}/8$ 10：采样频率为 $f_{CK\_CMP}/16$ 11：采样频率为 $f_{CK\_CMP}/32$
9:8	CMP1HST[1:0]	CMP1 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
7:6	CMP1INTEN[1:0]	CMP1 中断使能 该位域用于选择 CMP1 输出的有效边沿来产生中断请求。 00：中断禁能 01：上升沿 10：下降沿 11：双边沿 注意：该位域也用于为 CMP1IF 选择有效边沿。
5	CMP1OE	CMP1 输出使能 0：禁能 1：使能
4	CMP1POE	CMP1 引脚输出使能 0：禁能 1：使能
3	CMP1PL	CMP1 输出极性 该位用于控制 CMP1 输出极性。 0：输出是正相的 1：输出是反相的
2:1	保留	必须保持复位值。
0	CMP1EN	CMP1 使能

0: CMP1 禁能

1: CMP1 使能

17.4.5. CMP2 控制状态寄存器 (CMP2_CS)

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
CMP2LK	保留								CMP2PSEL[2:0]			保留	CMP2MISEL[2:0]		
rw									rw			rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	CMP2DFS NUM	保留		CMP2DFSCDIV[1:0]		CMP2HST[1:0]		CMP2INTEN[1:0]		CMP2OE	CMP2POE	CMP2PL	保留		CMP2EN
rw		rw				rw		rw		rw	rw	rw	rw		

位/位域	名称	描述
31	CMP2LK	CMP2 写保护 该位可将 CMP2 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: CMP2_CS 是可读可写位 1: CMP2_CS 是只读位
30:23	保留	必须保持复位值。
22:20	CMP2PSEL[2:0]	CMP2_IP 输入选择 该位域用于选择 CMP2 的 CMP2_IP 输入源。 000: 无输入 001: PC2 010: PC6 011: PC12 100: PC3 101: PC10 其他: 保留。
19	保留	必须保持复位值。
18:16	CMP2MISEL[2:0]	CMP2_IM 输入选择 该位域用于选择 CMP2 的 CMP2_IM 输入源。 000: 无输入 001: PC11 010: PC12 011: DAC0_OUT0 100: DAC0_OUT1

		其他：保留。
15	保留	必须保持复位值。
14	CMP2DFSNUM	CMP2 数字滤波器采样次数 该位用于选择 CMP2 数字滤波器有效采样次数。 0：连续采样三次 1：连续采样四次
13:12	保留	必须保持复位值。
11:10	CMP2DFSCDIV[1:0]	CMP2 数字滤波器采样时钟分频 该位域用于选择 CMP2 数字滤波器采样频率。 00：数字滤波器禁能 01：采样频率为 $f_{CK\_CMP}/8$ 10：采样频率为 $f_{CK\_CMP}/16$ 11：采样频率为 $f_{CK\_CMP}/32$
9:8	CMP2HST[1:0]	CMP2 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
7:6	CMP2INTEN[1:0]	CMP2 中断使能 该位域用于选择 CMP2 输出的有效边沿来产生中断请求。 00：中断禁能 01：上升沿 10：下降沿 11：双边沿 注意：该位域也用于为 CMP2IF 选择有效边沿。
5	CMP2OE	CMP2 输出使能 0：禁能 1：使能
4	CMP2POE	CMP2 引脚输出使能 0：禁能 1：使能
3	CMP2PL	CMP2 输出极性 该位用于控制 CMP2 输出极性。 0：输出是正相的 1：输出是反相的
2:1	保留	必须保持复位值。
0	CMP2EN	CMP2 使能

0: CMP2 禁能

1: CMP2 使能

17.4.6. CMP3 控制状态寄存器 (CMP3_CS)

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
CMP3LK	保留								CMP3PSEL[2:0]			保留	CMP3MISEL[2:0]		
rw									rw			rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	CMP3DFS NUM	保留		CMP3DFSCDIV[1:0]		CMP3HST[1:0]		CMP3INTEN[1:0]		CMP3OE	CMP3POE	CMP3PL	保留		CMP3EN
rw		rw				rw		rw		rw	rw	rw	rw		

位/位域	名称	描述
31	CMP3LK	CMP3 写保护 该位可将 CMP3 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: CMP3_CS 是可读可写位 1: CMP3_CS 是只读位
30:23	保留	必须保持复位值。
22:20	CMP3PSEL[2:0]	CMP3_IP 输入选择 该位域用于选择 CMP3 的 CMP3_IP 输入源。 000: 无输入 001: PC3 010: PD2 011: PD4 100: PD5 101: PC9 其他: 保留。
19	保留	必须保持复位值。
18:16	CMP3MISEL[2:0]	CMP2_IM 输入选择 该位域用于选择 CMP2 的 CMP2_IM 输入源。 000: 无输入 001: PC11 010: PC12 011: DAC0_OUT0 100: DAC0_OUT1

		其他：保留。
15	保留	必须保持复位值。
14	CMP3DFSNUM	CMP3 数字滤波器采样次数 该位用于选择 CMP3 数字滤波器有效采样次数。 0：连续采样三次 1：连续采样四次
13:12	保留	必须保持复位值。
11:10	CMP3DFSCDIV[1:0]	CMP3 数字滤波器采样时钟分频 该位域用于选择 CMP3 数字滤波器采样频率。 00：数字滤波器禁能 01：采样频率为 $f_{CK\_CMP}/8$ 10：采样频率为 $f_{CK\_CMP}/16$ 11：采样频率为 $f_{CK\_CMP}/32$
9:8	CMP3HST[1:0]	CMP3 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
7:6	CMP3INTEN[1:0]	CMP3 中断使能 该位域用于选择 CMP3 输出的有效边沿来产生中断请求。 00：中断禁能 01：上升沿 10：下降沿 11：双边沿 注意：该位域也用于为 CMP3IF 选择有效边沿。
5	CMP3OE	CMP3 输出使能 0：禁能 1：使能
4	CMP3POE	CMP3 引脚输出使能 0：禁能 1：使能
3	CMP3PL	CMP3 输出极性 该位用于控制 CMP3 输出极性。 0：输出是正相的 1：输出是反相的
2:1	保留	必须保持复位值。
0	CMP3EN	CMP3 使能

0: CMP3 禁能

1: CMP3 使能

18. 事件互连单元 (EVIC)

18.1. 简介

事件互连单元 (EVIC) 允许软件选择由各种外设产生的事件信号，用于各种应用。EVIC 为外设提供了灵活的机制来选择不同的事件源。因此，不同的外设模块通过 EVIC 可以进行彼此互连。

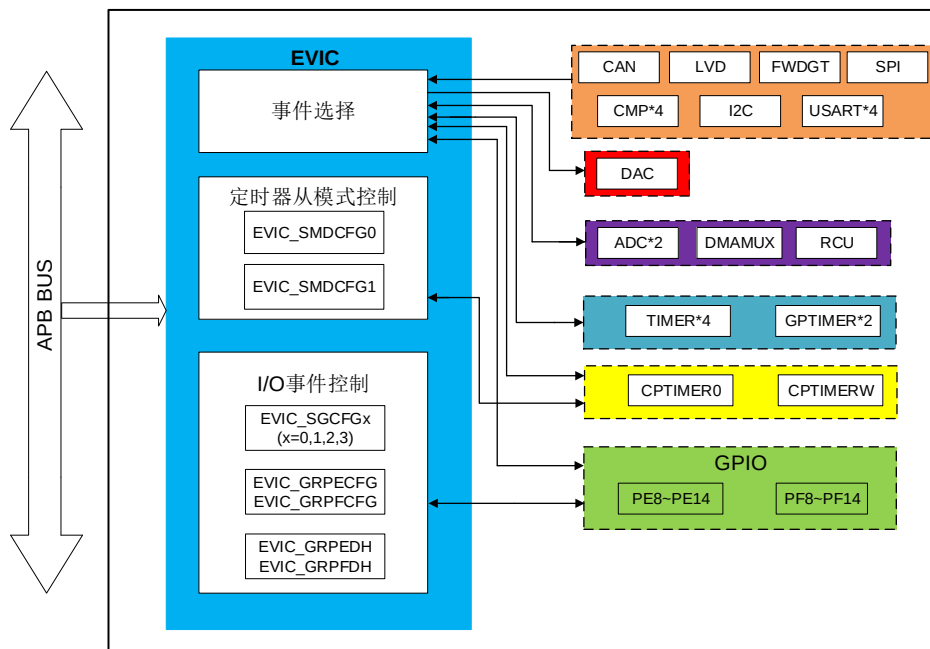
18.2. 主要特征

- 支持不同的事件信号；
- 每个外设有自己的寄存器选择事件源；
- 事件信号能够通过外部引脚输入检测产生或者其他外设输出；
- 事件触发控制 I/O 引脚状态。

18.3. 功能描述

EVIC 模块的框图如下所示：

图 18-1. EVIC 框图



18.3.1. 事件源选择

[表 18-1. 事件源位域选择](#) 给出了来自不同模块的事件信号。

表 18-1. 事件源位域选择

位域名称	值	事件源
EVSELx	0x00	无事件信号被选择
	0x01	软件生成的事件信号
	0x02	单个 I/O 输入检测 0
	0x03	单个 I/O 输入检测 1
	0x04	单个 I/O 输入检测 2
	0x05	单个 I/O 输入检测 3
	0x06	E 组 I/O 输入检测
	0x07	F 组 I/O 输入检测
	0x08	TIMER0 上溢事件
	0x09	TIMER0 下溢事件
	0x0a	TIMER0_CH0 比较事件
	0x0b	TIMER0_CH1 比较事件
	0x0c	TIMER0_CH2 比较事件
	0x0d	TIMER0_CH3 比较事件
	0x0e	TIMER0_MCH0 比较事件
	0x0f	TIMER0_MCH1 比较事件
	0x10	TIMER0_MCH2 比较事件
	0x11	TIMER0_MCH3 比较事件
	0x12	TIMER0 TRGO 信号
	0x13	TIMER7 上溢事件
	0x14	TIMER7 下溢事件
	0x15	TIMER7_CH0 比较事件
	0x16	TIMER7_CH1 比较事件
	0x17	TIMER7_CH2 比较事件
	0x18	TIMER7_CH3 比较事件
	0x19	TIMER7_MCH0 比较事件
	0x1a	TIMER7_MCH1 比较事件
	0x1b	TIMER7_MCH2 比较事件
	0x1c	TIMER7_MCH3 比较事件
	0x1d	TIMER7 TRGO 信号
	0x1e	TIMER1_CH0 比较事件
	0x1f	TIMER1_CH1 比较事件
	0x20	TIMER1_CH2 比较事件
	0x21	TIMER1_CH3 比较事件
	0x22	TIMER1 上溢事件
	0x23	TIMER1 下溢事件
	0x24	TIMER1 TRGO 信号
	0x25	TIMER2_CH0 比较事件
	0x26	TIMER2_CH1 比较事件
	0x27	TIMER2_CH2 比较事件

0x28	TIMER2_CH3 比较事件
0x29	TIMER2 上溢事件
0x2a	TIMER2 下溢事件
0x2b	TIMER2 TRGO 信号
0x2c	CPTIMER0 计数器 0 上溢事件
0x2d	保留
0x2e	CPTIMERW_OCH2 比较匹配事件
0x2f	CPTIMERW_OCH3 比较匹配事件
0x30	CPTIMERW_OCH0 比较匹配事件
0x31	CPTIMERW_OCH1 比较匹配事件
0x32	保留
0x33	保留
0x34	保留
0x35	GPTIMER0_CH0 比较匹配事件
0x36	GPTIMER0_CH1 比较匹配事件
0x37	GPTIMER0_CH0 附加比较匹配事件
0x38	GPTIMER0_CH1 附加比较匹配事件
0x39	GPTIMER0 上溢事件
0x3a	GPTIMER0 下溢事件
0x3b	GPTIMER0 ADTCV1 匹配产生的 ADC 触发事件
0x3c	GPTIMER0 ADTCV2 匹配产生的 ADC 触发事件
0x3d	GPTIMER0 重复计数完成事件
0x3e	保留
0x3f	保留
0x40	CMP0 输出
0x41	CMP1 输出
0x44	CMP2 输出
0x43	CMP3 输出
0x44	ADC0 组扫描完成事件
0x45	ADC2 组扫描完成事件
0x46	LVD1 电压检测
0x47	LVD2 电压检测
0x48	FWDGT 上溢或者刷新
0x49	HXTAL 卡住事件
0x4a	DMA0 通道 0 传输完成
0x4b	DMA0 通道 1 传输完成
0x4c	DMA0 通道 2 传输完成
0x4d	DMA0 通道 3 传输完成
0x4e	DMA0 通道 4 传输完成
0x4f	DMA0 通道 5 传输完成
0x50	DMA1 通道 0 传输完成
0x51	DMA1 通道 1 传输完成

0x52	DMA1 通道 2 传输完成
0x53	DMA1 通道 3 传输完成
0x54	DMA1 通道 4 传输完成
0x55	DMA1 通道 5 传输完成
0x56	CAN 发送中断事件
0x57	CAN FIFO0 中断事件
0x58	CAN FIFO1 中断事件
0x59	CAN EWMC 中断事件
0x5a	保留
0x5b	保留
0x5c	UART0 错误(PERR /FERR /NERR /ORERR)
0x5d	UART0 发送结束
0x5e	UART1 错误(PERR /FERR /NERR /ORERR)
0x5f	UART1 发送结束
0x60	UART2 错误(PERR /FERR /NERR /ORERR)
0x61	UART2 发送结束
0x65	UART3 错误(PERR /FERR /NERR /ORERR)
0x63	UART3 发送结束
0x65	保留
0x65	保留
0x66	保留
0x67	SPI 错误
0x68	SPI 传输完成
0x69	I2C 通信错误/通信事件
0x6a	在接收期间 I2C_RDATA 非空
0x6b	I2C 发送中断
0x6c	I2C 传输完成
0x6d	GPTIMER1_CH0 比较匹配事件
0x6e	GPTIMER1_CH1 比较匹配事件
0x6f	GPTIMER1_CH0 附加比较匹配事件
0x70	GPTIMER1_CH1 附加比较匹配事件
0x71	GPTIMER1 上溢事件
0x72	GPTIMER1 下溢事件
0x73	GPTIMER1 ADTCV1 匹配产生的 ADC 触发事件
0x74	GPTIMER1 ADTCV2 匹配产生的 ADC 触发事件
0x75	GPTIMER1 重复计数完成事件

18.3.2. I/O 引脚的事件互连

用于事件触发和事件生成功能的 I/O 引脚可以从端口 E 和端口 F 中选择。

单个 I/O 和组 I/O

用于事件互连的 I/O 引脚能够分成两种类型：单 I/O 引脚和组 I/O 引脚。

对于前者，可以通过配置 `EVIC_SGIOCFGx` ($x=0\ldots3$) 寄存器的 `SGIOPORT[1:0]` 位域和 `SGIOPIN[2:0]` 位域来确定具体的 I/O 引脚。被选择的单个 I/O 引脚可以进行输入检测生成事件，和选择事件信号触发该引脚相应电平输出。

对于后者，来自同一个端口的两个或者更多的引脚能够形成一个组。组成员被叫做组 I/O 引脚。同一个组的成员只能来自同一个 GPIO 端口（端口 E 或者端口 F），因此，可以分成组 E 和组 F。此外，根据组成员是输入引脚还是输出引脚，组 E/F 可以进一步分成输入组 E/F 和输出组 E/F。可以通过配置 `EVIC_GRPFCFG` 和 `EVIC_GRPFCFG` 寄存器来确定每个组的成员（输入和输出由 GPIO 模块寄存器决定）。每个组有单独的寄存器配置输入检测来生成事件，和选择事件信号来触发引脚输出和引脚电平加载。

如果一个引脚既被选为单个 I/O 引脚，同时也是组 I/O 引脚，单个 I/O 和组 I/O 寄存器都能被用于控制事件生成。但是，事件触发功能仅由组 I/O 寄存器来控制。

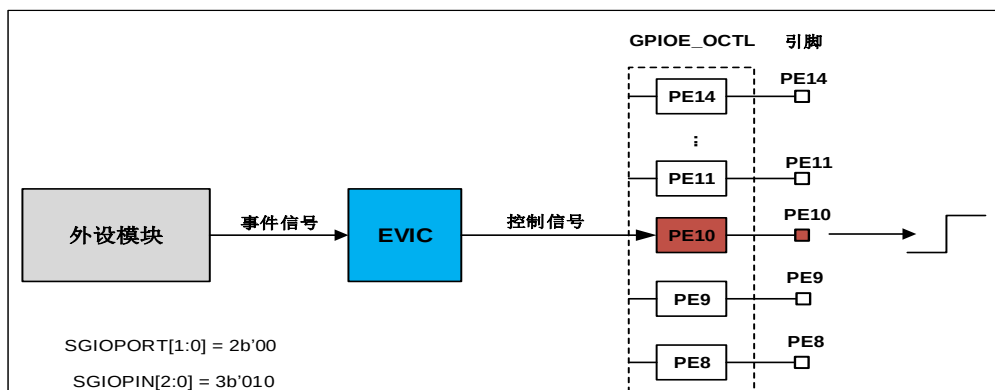
单个 I/O 引脚的事件生成

EVIC 提供了 4 个单个 I/O 输入检测器，可以同时检测多达 4 个单个 I/O 输入。当由 `EVIC_SGIOCFGx` ($x=0\ldots3$) 寄存器选择的单 I/O 引脚被设置成输入时，如果在引脚的输入信号中检测到有效边沿，那么将产生一个事件信号。有效边沿可以通过 `EVIC_SGIOCFGx` ($x=0\ldots3$) 寄存器的 `SGIOM[1:0]` 位来设置。

事件触发单个 I/O 引脚输出

EVIC 提供了 4 个单个 I/O 输出控制器。当由 `EVIC_SGIOCFGx` ($x=0\ldots3$) 寄存器选择的单个 I/O 引脚被设置成输出时，如果通过 `EVIC_SGIOCFGx` ($x=0\ldots3$) 寄存器选择的事件发生，那么，单个 I/O 引脚将输出相应的电平。输出电平可以通过 `EVIC_SGIOCFGx` ($x=0\ldots3$) 寄存器进行设置。

图 18-2. 单个 I/O 引脚输出举例



输入组 I/O 引脚的事件生成

EVIC 提供了组 E 和组 F 的 I/O 输入检测器。当输入组中任意一个引脚输入的有效边沿被检测

到，那么，输入组将产生一个事件信号。输入组 E 或者输入组 F 的有效边沿可以分别通过 EVIC_GRPECFG 寄存器的 GRPEIND[1:0] 位域或者 EVIC_GRPFCFG 寄存器的 GRPFIND[1:0] 位域确定。

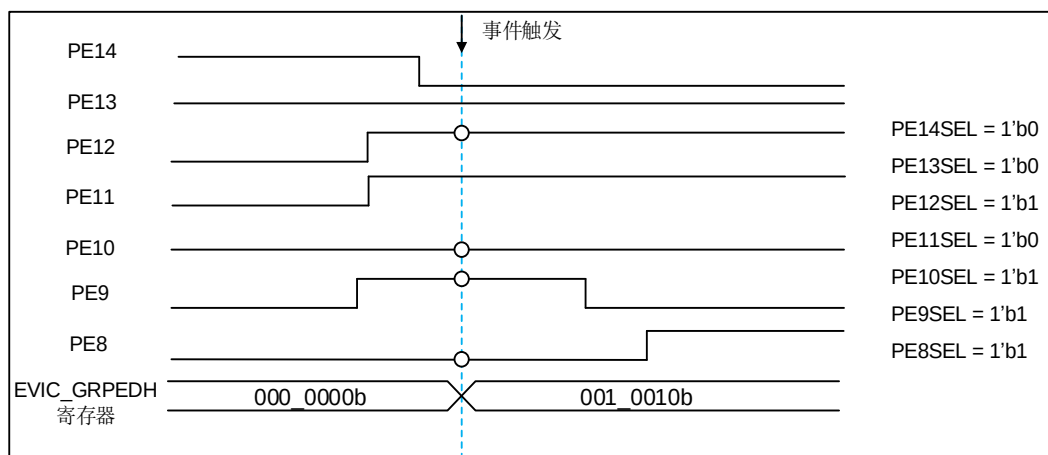
事件触发组 I/O 引脚的输入加载

EVIC 提供了组 E 和组 F 的 I/O 输入控制器。当由 EVIC_INGRPE 或者 EVIC_INGRPF 寄存器选择的事件信号生成时，输入组 I/O 引脚的电平会被加载到 EVIC_GRPEDH 或 EVIC_GRPFDH 寄存器相应的位中。不与输入组引脚相对应的寄存器位值不会被改变。

如果覆写功能使能（EVIC_GRPECFG / EVIC_GRPFCFG 寄存器的 OVWEN = 1'b1），当又一事件信号到来时，当前输入组 I/O 引脚的电平会被加载到数据保存寄存器（EVIC_GRPEDH / EVIC_GRPFDH），以覆盖之前的电平值。

如果覆写功能禁能（OVWEN = 1'b0），又一事件信号的到来不会使得数据保持寄存器（EVIC_GRPEDH 或者 EVIC_GRPFDH）的值被覆写，除非上次加载的值已经被读取。如果上次的加载值没有被读取，那么，事件信号的到来会被忽略。

图 18-3. 组 E 的 I/O 引脚输入加载举例

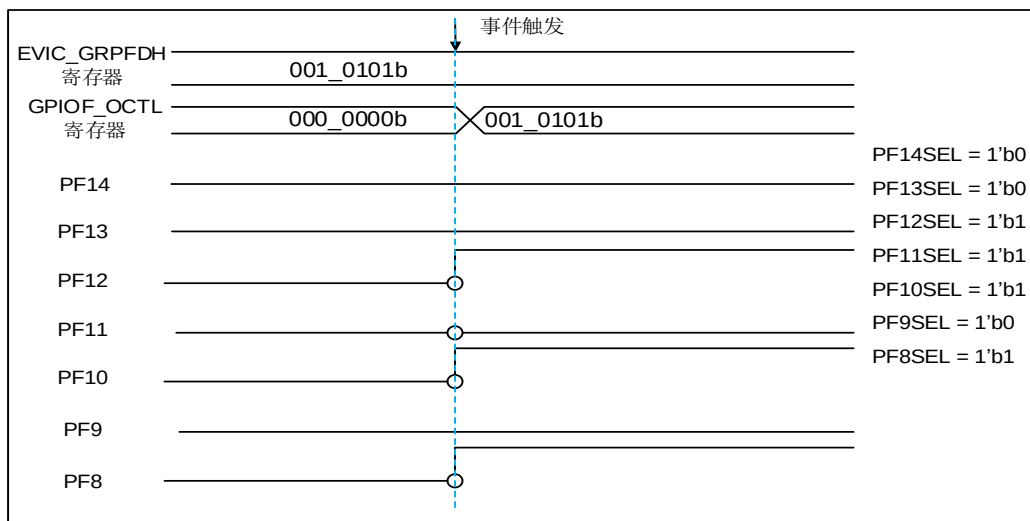


事件触发组 I/O 引脚输出

EVIC 提供了组 E 和组 F 的 I/O 输出控制器。当由 EVIC_OUTEGRP 或者 EVIC_OUTFGRP 寄存器选择的事件源产生事件信号时，输出组 I/O 引脚将输出相应电平。EVIC_GRPECFG 寄存器的 GRPEOSEL[2:0] 位域和 EVIC_GRPFCFG 寄存器的 GRPFOSEL[2:0] 位域分别决定了输出组 E 的 I/O 引脚电平和输出组 F 的 I/O 引脚电平。

如果 GRPEOSEL[2:0] 或者 GRPFOSEL[2:0] 位域的值 3'b011，当所选的事件信号发生时，只有与输出组引脚相对应的 EVIC_GRPEDH 或者 EVIC_GRPFDH 寄存器的位值会被传递到相对应的 GPIOE_OCTL 或者 GPIOF_OCTL 寄存器位中，因此，输出组 E 或者输出组 F 的 I/O 引脚将在事件信号到来时输出 EVIC_GRPEDH 或者 EVIC_GRPFDH 寄存器相对应的电平值。

图 18-4. 组 I/O 引脚输出举例

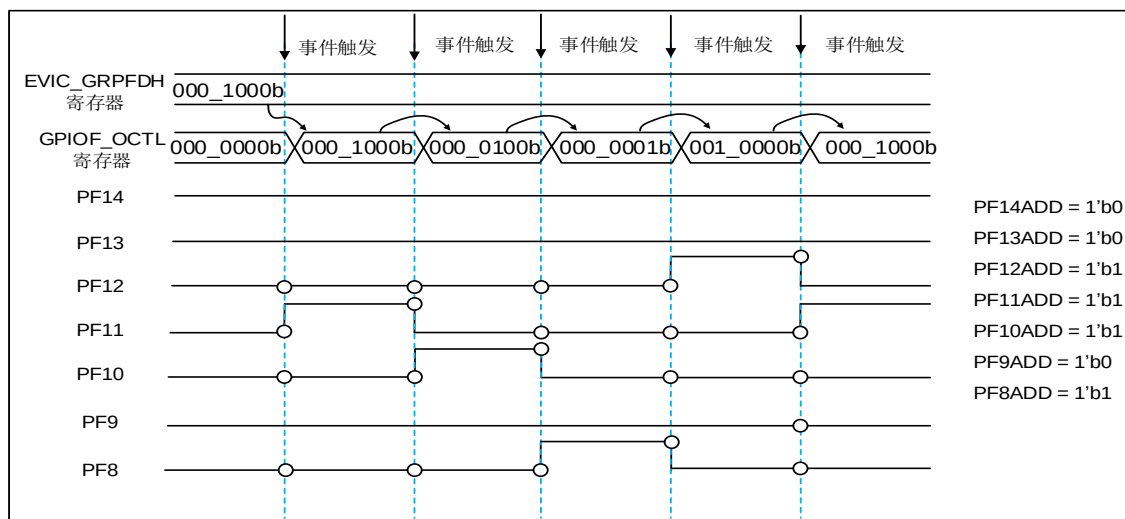


如果 $GRPEOSEL[2:0]$ 或者 $GRPFSEL[2:0]$ 位域的值 3'b1xx，当事件信号第一次到来时，数据保持寄存器（EVIC_GRPEDH 或者 EVIC_GRPFD）的值将传递到 GPIOE_OCTL 或者 GPIOF_OCTL 寄存器中。

此后，每当有事件触发到来，数据保持寄存器（EVIC_GRPEDH 或者 EVIC_GRPFD）的值将从最高有效位向最低有效位进行循环右移一位（只有与输出组 I/O 引脚相对应的寄存器位进行移位）。

如果在移位期间，数据保持寄存器（EVIC_GRPEDH 或者 EVIC_GRPFD）被写入了新值，那么，下一次事件触发不会对 GPIOE_OCTL 或者 GPIOF_OCTL 寄存器进行移位而是将新值传递到 GPIOE_OCTL 或者 GPIOF_OCTL 寄存器中。

图 18-5. 组 I/O 引脚输出循环模式



注意:

1. 向输入组 I/O 引脚对应的数据保持寄存器（EVIC_GRPEDH 或者 EVIC_GRPFDH 寄存器）位写值是禁能的。

2. 如果选择某个 I/O 引脚作为输出组 I/O 引脚或单个 I/O 输出引脚，则禁止写入该 I/O 引脚对应的 GPIOx_OCTL 寄存器位。

18.4. EVIC 寄存器

EVIC 基地址: 0x4001 8400

18.4.1. 软件事件寄存器 (EVIC_SWEV)

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

位/位域	名称	描述
31	RWEN	EVIC_SWEV 寄存器写使能 该位被软件置位或者清 0。该位置位后, EVIC_SWEV 寄存器可以写。 0: EVIC_SWEV 寄存器写禁能 1: EVIC_SWEV 寄存器写使能
30:2	保留	必须保持复位值。
1	SWEVG	软件事件生成 该位由软件置位, 用于产生事件信号, 软件事件产生后由硬件自动清0。 0: 未产生事件信号 1: 产生事件信号 注意: 当BWEN位和RWEN位都为1时, 该位可以被写。
0	BWEN	SWEVG位写使能 该位用于使能对SWEVG位的写访问。该位为1时, SWEVG位可以被写。 0: SWEVG 位写禁能 1: SWEVG 位写使能 注意: 当 RWEN 位都为 1 时, 该位可以被写。

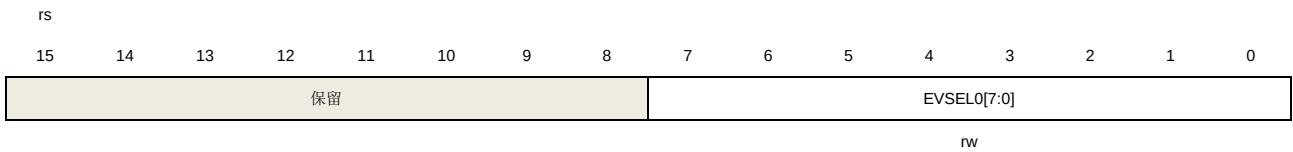
18.4.2. 单个 I/O 事件互连寄存器 x (EVIC_SGIOx) (x=0...3)

地址偏移: 0x04 + 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
LK	保留														



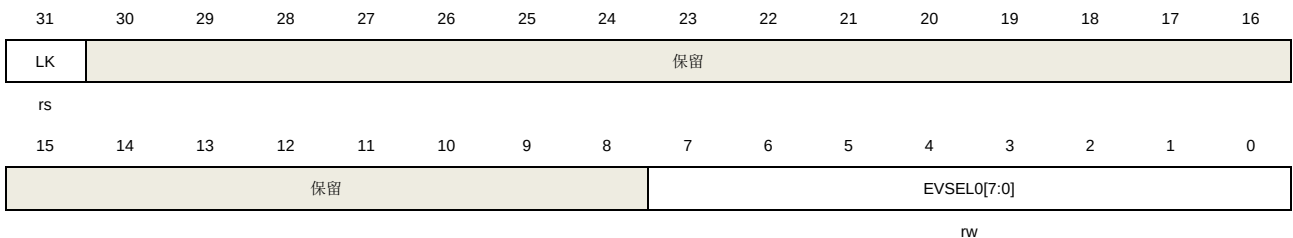
位/位域	名称	描述
31	LK	EVIC_SGIOx 寄存器写保护 该位通过软件置位, 通过系统复位清除。该位置位后, EVIC_INGRPE 寄存器为只读。 0: EVIC_SGIOx 寄存器可读可写 1: EVIC_SGIOx 寄存器只读
30:8	保留	必须保持复位值。
7:0	EVSEL0[7:0]	单个 I/O 的事件源选择 这些位用于选择连接到单个 I/O 输出控制器 x 的事件源。当所选事件发生时, 由 EVIC_SGCFGx 寄存器选择的单个 I/O 引脚将输出特定电平。关于具体配置请参考 表 18-1. 事件源位域选择 。

18.4.3. 输入组 E 事件互连寄存器 (EVIC_INGRPE)

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	EVIC_INGRPE 寄存器写保护 该位通过软件置位, 通过系统复位清除。该位置位后, EVIC_INGRPE 寄存器为只读。 0: EVIC_INGRPE 寄存器可读可写 1: EVIC_INGRPE 寄存器只读
30:8	保留	必须保持复位值。
7:0	EVSEL0[7:0]	输入组 E 的事件源选择 这些位用于选择连接到组 E 的 I/O 输入控制器的事件源。当选择的事件发生时, 组 E 的 I/O 引脚的输入电平将被加载到 EVIC_GRPEDH 寄存器的相应位。关于具体配置请参考 表 18-1. 事件源位域选择 。

18.4.4. 输入组 F 事件互连寄存器 (EVIC_INGRPF)

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
LK	保留														
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								EVSEL0[7:0]							
rw															

位/位域	名称	描述
31	LK	EVIC_INGRPF 寄存器写保护 该位通过软件置位, 通过系统复位清除。该位置位后, EVIC_INGRPF 寄存器为只读。 0: EVIC_INGRPF 寄存器可读可写 1: EVIC_INGRPF 寄存器只读
30:8	保留	必须保持复位值。
7:0	EVSEL0[7:0]	输入组F的事件源选择 这些位用于选择连接到组 F 的 I/O 输入控制器的事件源。当选择的事件发生时, 组 F 的 I/O 引脚的输入电平将被加载到 EVIC_GRPFDH 寄存器的相应位。关于具体配置请参考 表 18-1. 事件源位域选择 。

18.4.5. 输出组 E 事件互连寄存器 (EVIC_OUTGRPE)

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
LK	保留														
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								EVSEL0[7:0]							
rw															

位/位域	名称	描述
31	LK	EVIC_OUTGRPE 寄存器写保护 该位通过软件置位, 通过系统复位清除。该位置位后, EVIC_OUTGRPE 寄存器为只读。 0: EVIC_OUTGRPE 寄存器可读可写

1: EVIC_OUTGRPE 寄存器只读

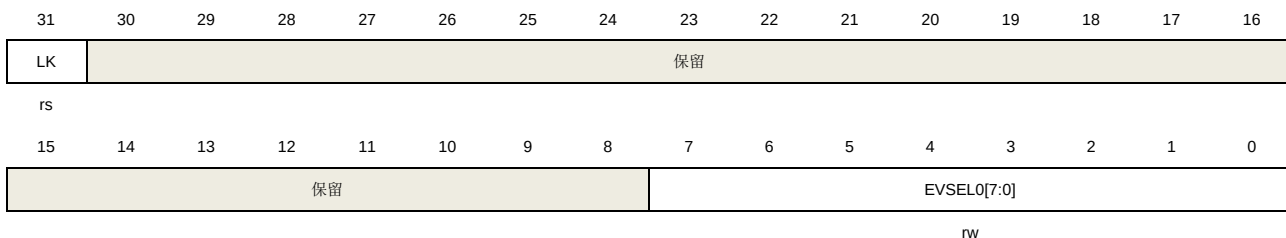
30:8	保留	必须保持复位值。
7:0	EVSEL0[7:0]	输出组E的事件源选择 这些位用于选择连接到组 E 的 I/O 输出控制器的事件源。当选择的事件发生时，组 E 的 I/O 引脚将输出指定的电平。关于具体配置请参考 表 18-1. 事件源位域选择 。

18.4.6. 输出组 F 事件互连寄存器 (EVIC_OUTGRPF)

地址偏移: 0x20

复位值: 0x0000 0000

该寄存器只能按字（32位）访问。



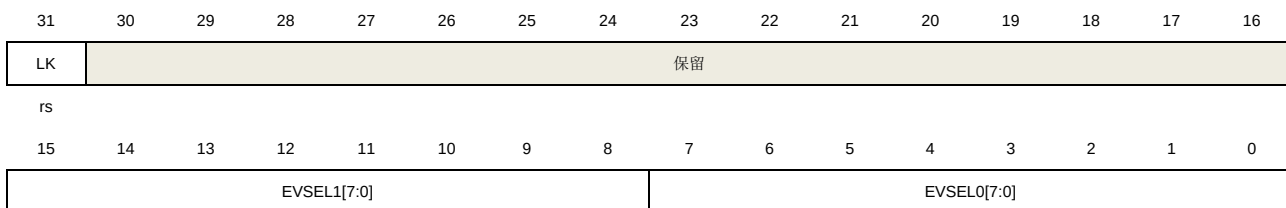
位/位域	名称	描述
31	LK	EVIC_OUTGRPF 寄存器写保护 该位通过软件置位，通过系统复位清除。该位置位后，EVIC_OUTGRPF 寄存器为只读。 0: EVIC_OUTGRPF 寄存器可读可写 1: EVIC_OUTGRPF 寄存器只读
30:8	保留	必须保持复位值。
7:0	EVSEL0[7:0]	输出组F的事件源选择 这些位用于选择连接到组 F 的 I/O 输出控制器的事件源。当选择的事件发生时，组 F 的 I/O 引脚将输出指定的电平。关于具体配置请参考 表 18-1. 事件源位域选择 。

18.4.7. DAC0 转换事件互连寄存器 (EVIC_DAC0COV)

地址偏移: 0x24

复位值: 0x0000 0000

该寄存器只能按字（32位）访问。



rw

rw

位/位域	名称	描述
31	LK	EVIC_DAC0COV 寄存器写保护 该位通过软件置位，通过系统复位清除。该位置位后，EVIC_DAC0COV 寄存器为只读。 0: EVIC_DAC0COV 寄存器可读可写 1: EVIC_DAC0COV 寄存器只读
30:16	保留	必须保持复位值。
15:8	EVSEL1[7:0]	DAC0_OUT1转换的事件源选择 这些位用于选择连接到 DAC0_OUT1 转换外部触发器的事件源。关于具体配置请参考 表 18-1. 事件源位域选择 。
7:0	EVSEL0[7:0]	DAC0_OUT0转换的事件源选择 这些位用于选择连接到 DAC0_OUT0 转换外部触发器的事件源。关于具体配置请参考 表 18-1. 事件源位域选择 。

18.4.8. ADC0 转换事件互连寄存器（EVIC_ADC0COV）

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
LK	保留														
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVSEL1[7:0]								EVSEL0[7:0]							
rw								rw							

位/位域	名称	描述
31	LK	EVIC_ADC0COV 寄存器写保护 该位通过软件置位，通过系统复位清除。该位置位后，EVIC_ADC0COV 寄存器为只读。 0: EVIC_ADC0COV 寄存器可读可写 1: EVIC_ADC0COV 寄存器只读
30:16	保留	必须保持复位值。
15:8	EVSEL1[7:0]	ADC0组转换的事件源选择1 这些位用于选择连接到 ADC0 组转换外部触发 1 的事件源。关于具体配置请参考 表 18-1. 事件源位域选择 。
7:0	EVSEL0[7:0]	ADC0组转换的事件源选择0

这些位用于选择连接到 ADC0 组转换外部触发 0 的事件源。关于具体配置请参考[表 18-1. 事件源位域选择](#)。

18.4.9. ADC2 转换事件互连寄存器 (EVIC_ADC2COV)

地址偏移: 0x2C

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	保留														
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVSEL1[7:0]								EVSEL0[7:0]							
rw								rw							

位/位域	名称	描述
31	LK	EVIC_ADC2COV 寄存器写保护 该位通过软件置位，通过系统复位清除。该位置位后，EVIC_ADC2COV 寄存器为只读。 0: EVIC_ADC2COV 寄存器可读可写 1: EVIC_ADC2COV 寄存器只读
30:16	保留	必须保持复位值。
15:8	EVSEL1[7:0]	ADC2组转换的事件源选择1 这些位用于选择连接到 ADC2 组转换外部触发 1 的事件源。关于具体配置请参考 表 18-1. 事件源位域选择 。
7:0	EVSEL0[7:0]	ADC2组转换的事件源选择0 这些位用于选择连接到 ADC2 组转换外部触发 0 的事件源。关于具体配置请参考 表 18-1. 事件源位域选择 。

18.4.10. RCU 事件互连寄存器 (EVIC_RCU)

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								EVSEL0[7:0]							

rw

rw

位/位域	名称	描述
31	LK	EVIC_RCU 寄存器写保护 该位通过软件置位，通过系统复位清除。该位置位后，EVIC_RCU 寄存器为只读。 0: EVIC_RCU 寄存器可读可写 1: EVIC_RCU 寄存器只读
30:8	保留	必须保持复位值。
7:0	EVSEL0[7:0]	IRC32M切换的事件源选择 这些位用于选择连接到时钟切换触发输入的事件源。当选择的事件发生时，CK_SYS 时钟切换到 IRC32M 时钟。关于具体配置请参考 表 18-1. 事件源位域选择 。

18.4.11. TIMER0 事件互连寄存器 (EVIC_TIMER0)

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								EVSEL0[7:0]							
rw															

位/位域	名称	描述
31	LK	EVIC_TIMER0 寄存器写保护 该位通过软件置位，通过系统复位清除。该位置位后，EVIC_TIMER0 寄存器为只读。 0: EVIC_TIMER0 寄存器可读可写 1: EVIC_TIMER0 寄存器只读
30:8	保留	必须保持复位值。
7:0	EVSEL0[7:0]	TIMER0从模式事件源选择 这些位用于选择连接到TIMER0从模式触发输入的事件源。关于具体配置请参考 表 18-1. 事件源位域选择 。

18.4.12. TIMER7 事件互连寄存器 (EVIC_TIMER7)

地址偏移: 0x3C

复位值: 0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	保留														
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								EVSEL0[7:0]							
rw															

位/位域	名称	描述
31	LK	EVIC_TIMER7 寄存器写保护 该位通过软件置位，通过系统复位清除。该位置位后，EVIC_TIMER7 寄存器为只读。 0: EVIC_TIMER7 寄存器可读可写 1: EVIC_TIMER7 寄存器只读
30:8	保留	必须保持复位值。
7:0	EVSEL0[7:0]	TIMER7从模式事件源选择 这些位用于选择连接到TIMER7从模式触发输入的事件源。关于具体配置请参考 表 18-1. 事件源位域选择 。

18.4.13. CPTIMER0 事件互连寄存器（EVIC_CPTIMER0）

地址偏移：0x40

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	保留														
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								EVSEL0[7:0]							
rw															

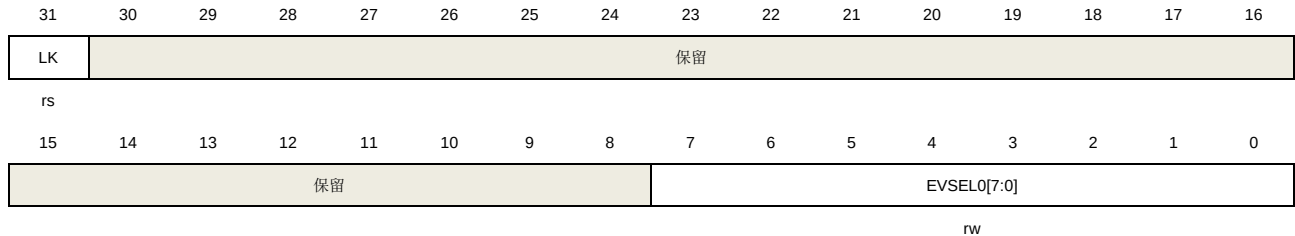
位/位域	名称	描述
31	LK	EVIC_CPTIMER0 寄存器写保护 该位通过软件置位，通过系统复位清除。该位置位后，EVIC_CPTIMER0 寄存器为只读。 0: EVIC_CPTIMER0 寄存器可读可写 1: EVIC_CPTIMER0 寄存器只读
30:8	保留	必须保持复位值。
7:0	EVSEL0[7:0]	CPTIMER0从模式事件源选择 这些位用于选择连接到 CPTIMER0 从模式触发输入的事件源。关于具体配置请参考 表 18-1. 事件源位域选择 。

18.4.14. CPTIMERW 事件互连寄存器 (EVIC_CPTIMERW)

地址偏移：0x44

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



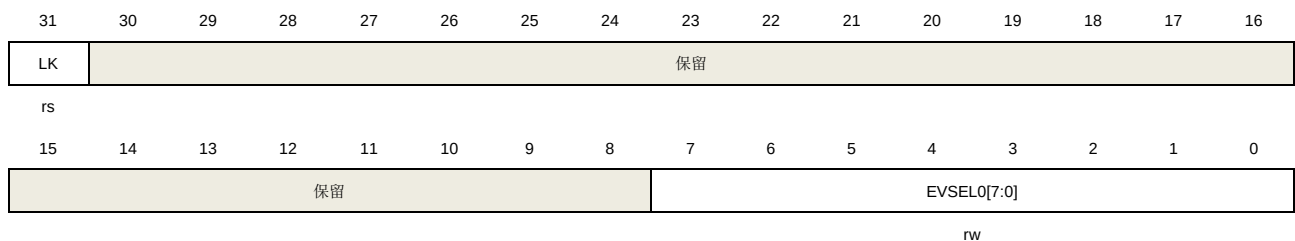
位/位域	名称	描述
31	LK	EVIC_CPTIMERW 寄存器写保护 该位通过软件置位，通过系统复位清除。该位置位后，EVIC_CPTIMERW 寄存器为只读。 0: EVIC_CPTIMERW 寄存器可读可写 1: EVIC_CPTIMERW 寄存器只读
30:8	保留	必须保持复位值。
7:0	EVSEL0[7:0]	CPTIMERW从模式事件源选择 这些位用于选择连接到 CPTIMERW 从模式触发输入的事件源。关于具体配置请参考 表 18-1. 事件源位域选择 。

18.4.15. TIMER1 事件互连寄存器 (EVIC_TIMER1)

地址偏移：0x48

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31	LK	EVIC_TIMER1 寄存器写保护 该位通过软件置位，通过系统复位清除。该位置位后，EVIC_TIMER1 寄存器为只读。 0: EVIC_TIMER1 寄存器可读可写

1: EVIC_TIMER1 寄存器只读

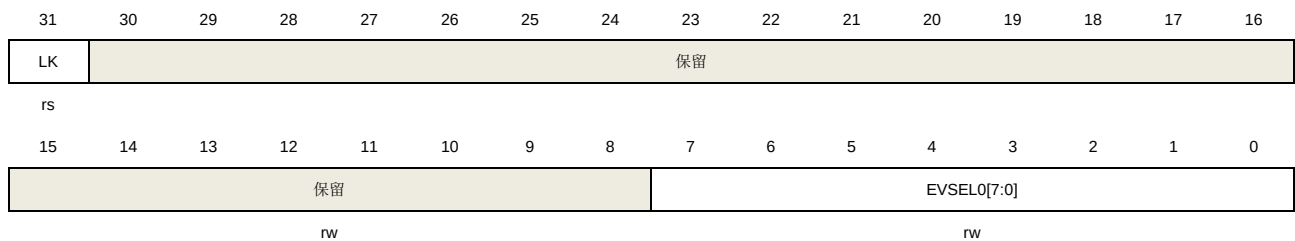
30:8	保留	必须保持复位值。
7:0	EVSEL0[7:0]	TIMER1从模式事件源选择 这些位用于选择连接到 TIMER1 从模式触发输入的事件源。关于具体配置请参考 表 18-1. 事件源位域选择 。

18.4.16. TIMER2 事件互连寄存器 (EVIC_TIMER2)

地址偏移: 0x4C

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问。



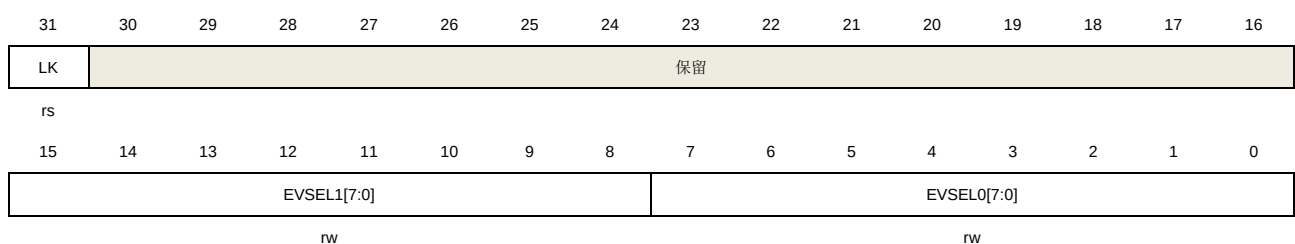
位/位域	名称	描述
31	LK	EVIC_TIMER2 寄存器写保护 该位通过软件置位，通过系统复位清除。该位置位后，EVIC_TIMER2 寄存器为只读。 0: EVIC_TIMER2 寄存器可读可写 1: EVIC_TIMER2 寄存器只读
30:8	保留	必须保持复位值。
7:0	EVSEL0[7:0]	TIMER2从模式事件源选择 这些位用于选择连接到 TIMER2 从模式触发输入的事件源。关于具体配置请参考 表 18-1. 事件源位域选择 。

18.4.17. GPTIMER 事件互连寄存器 0 (EVIC_GPTIMER_0)

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	描述
31	LK	EVIC_GPTIMER_0 寄存器写保护 该位通过软件置位，通过系统复位清除。该位置位后，EVIC_GPTIMER_0 寄存器为只读。 0: EVIC_GPTIMER_0 寄存器可读可写 1: EVIC_GPTIMER_0 寄存器只读
30:16	保留	必须保持复位值。
15:8	EVSEL1[7:0]	GPTIMER从模式事件源选择1 这些位用于选择连接到 GPTIMER 从模式触发输入 1 的事件源。关于具体配置请参考 表 18-1. 事件源位域选择 。
7:0	EVSEL0[7:0]	GPTIMER从模式事件源选择0 这些位用于选择连接到 GPTIMER 从模式触发输入 0 的事件源。关于具体配置请参考 表 18-1. 事件源位域选择 。

18.4.18. GPTIMER 事件互连寄存器 1 (EVIC_GPTIMER_1)

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	LK	EVIC_GPTIMER_1 寄存器写保护 该位通过软件置位，通过系统复位清除。该位置位后，EVIC_GPTIMER_1 寄存器为只读。 0: EVIC_GPTIMER_1 寄存器可读可写 1: EVIC_GPTIMER_1 寄存器只读
30:16	保留	必须保持复位值。
15:8	EVSEL3[7:0]	GPTIMER从模式事件源选择3 这些位用于选择连接到 GPTIMER 从模式触发输入 3 的事件源。关于具体配置请参考 表 18-1. 事件源位域选择 。
7:0	EVSEL2[7:0]	GPTIMER从模式事件源选择2 这些位用于选择连接到 GPTIMER 从模式触发输入 2 的事件源。关于具体配置请参考

考表 18-1. 事件源位域选择。

18.4.19. GPTIMER 事件互连寄存器 2 (EVIC_GPTIMER_2)

地址偏移: 0x58

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问。



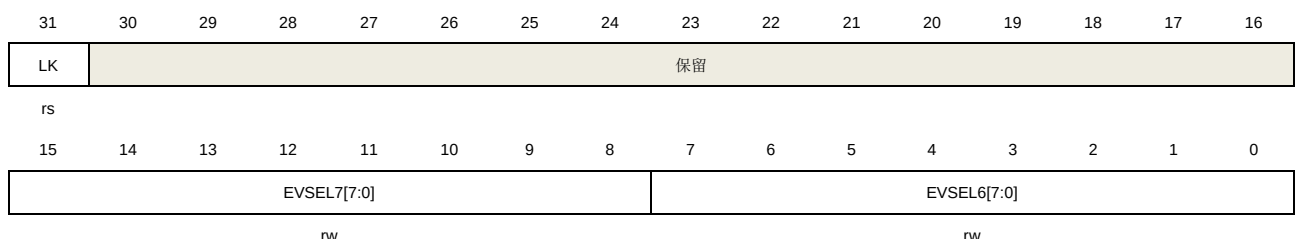
位/位域	名称	描述
31	LK	EVIC_GPTIMER_2 寄存器写保护 该位通过软件置位, 通过系统复位清除。该位置位后, EVIC_GPTIMER_2 寄存器为只读。 0: EVIC_GPTIMER_2 寄存器可读可写 1: EVIC_GPTIMER_2 寄存器只读
30:16	保留	必须保持复位值。
15:8	EVSEL5[7:0]	GPTIMER从模式事件源选择5 这些位用于选择连接到 GPTIMER 从模式触发输入 5 的事件源。关于具体配置请参考表 18-1. 事件源位域选择。
7:0	EVSEL4[7:0]	GPTIMER从模式事件源选择4 这些位用于选择连接到 GPTIMER 从模式触发输入 4 的事件源。关于具体配置请参考表 18-1. 事件源位域选择。

18.4.20. GPTIMER 事件互连寄存器 3 (EVIC_GPTIMER_3)

地址偏移: 0x5C

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	EVIC_GPTIMER_3 寄存器写保护 该位通过软件置位，通过系统复位清除。该位置位后，EVIC_GPTIMER_3 寄存器为只读。 0: EVIC_GPTIMER_3 寄存器可读可写 1: EVIC_GPTIMER_3 寄存器只读
30:16	保留	必须保持复位值。
15:8	EVSEL7[7:0]	GPTIMER从模式事件源选择7 这些位用于选择连接到 GPTIMER 从模式触发输入 7 的事件源。关于具体配置请参考 表 18-1. 事件源位域选择 。
7:0	EVSEL6[7:0]	GPTIMER从模式事件源选择6 这些位用于选择连接到 GPTIMER 从模式触发输入 6 的事件源。关于具体配置请参考 表 18-1. 事件源位域选择 。

18.4.21. DAC0 使能事件互连寄存器（EVIC_DAC0EN）

地址偏移：0x60

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
EVSEL1[7:0]								EVSEL0[7:0]							
rw								rw							

位/位域	名称	描述
31	LK	EVIC_DAC0EN 寄存器写保护 该位通过软件置位，通过系统复位清除。该位置位后，EVIC_DAC0EN 寄存器为只读。 0: EVIC_DAC0EN 寄存器可读可写 1: EVIC_DAC0EN 寄存器只读
30:16	保留	必须保持复位值。
15:8	EVSEL1[7:0]	使能DAC0_OUT1的事件源选择 这些位用于选择触发 DAC0_OUT1 使能的事件源。关于具体配置请参考 表 18-1. 事件源位域选择 。
7:0	EVSEL0[7:0]	使能DAC0_OUT0的事件源选择 这些位用于选择触发 DAC0_OUT0 使能的事件源。关于具体配置请参考 表 18-1. 事件源位域选择 。

[件源位域选择。](#)

18.4.22. DMAMUX 事件互连寄存器 0 (EVIC_DMAMUX_0)

地址偏移：0x64

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



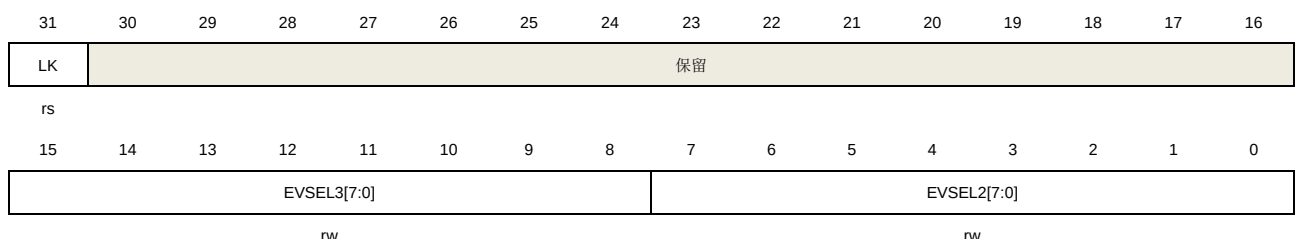
位/位域	名称	描述
31	LK	EVIC_DMAMUX_0 寄存器写保护 该位通过软件置位，通过系统复位清除。该位置位后，EVIC_DMAMUX_0 寄存器为只读。 0: EVIC_DMAMUX_0 寄存器可读可写 1: EVIC_DMAMUX_0 寄存器只读
30:16	保留	必须保持复位值。
15:8	EVSEL1[7:0]	DMAMUX的事件源选择1 这些位用于选择事件源作为 DMAMUX 请求多路复用器输入 1。关于具体配置请参考 表 18-1. 事件源位域选择。
7:0	EVSEL0[7:0]	DMAMUX的事件源选择0 这些位用于选择事件源作为 DMAMUX 请求多路复用器输入 0。关于具体配置请参考 表 18-1. 事件源位域选择。

18.4.23. DMAMUX 事件互连寄存器 1 (EVIC_DMAMUX_1)

地址偏移：0x68

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



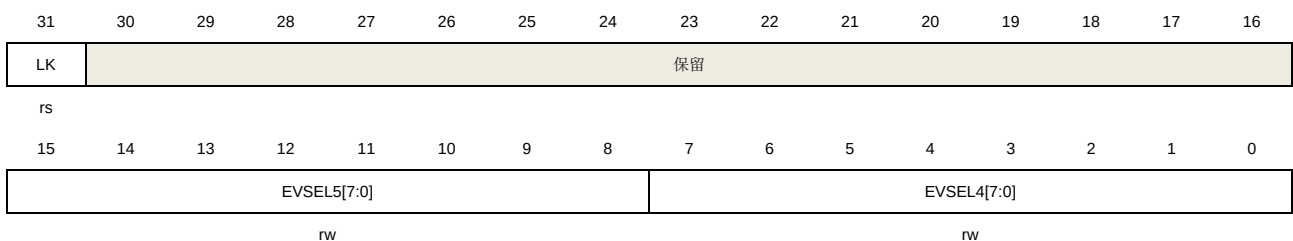
位/位域	名称	描述
31	LK	EVIC_DMAMUX_1 寄存器写保护 该位通过软件置位，通过系统复位清除。该位置位后，EVIC_DMAMUX_1 寄存器为只读。 0: EVIC_DMAMUX_1 寄存器可读可写 1: EVIC_DMAMUX_1 寄存器只读
30:16	保留	必须保持复位值。
15:8	EVSEL3[7:0]	DMAMUX的事件源选择3 这些位用于选择事件源作为 DMAMUX 请求多路复用器输入 3。关于具体配置请参考 表 18-1. 事件源位域选择 。
7:0	EVSEL2[7:0]	DMAMUX的事件源选择2 这些位用于选择事件源作为 DMAMUX 请求多路复用器输入 2。关于具体配置请参考 表 18-1. 事件源位域选择 。

18.4.24. DMAMUX 事件互连寄存器 2 (EVIC_DMAMUX_2)

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	LK	EVIC_DMAMUX_2 寄存器写保护 该位通过软件置位，通过系统复位清除。该位置位后，EVIC_DMAMUX_2 寄存器为只读。 0: EVIC_DMAMUX_2 寄存器可读可写 1: EVIC_DMAMUX_2 寄存器只读
30:16	保留	必须保持复位值。
15:8	EVSEL5[7:0]	DMAMUX的事件源选择5 这些位用于选择事件源作为 DMAMUX 请求多路复用器输入 5。关于具体配置请参考 表 18-1. 事件源位域选择 。
7:0	EVSEL4[7:0]	DMAMUX的事件源选择4 这些位用于选择事件源作为 DMAMUX 请求多路复用器输入 4。关于具体配置请参考

[表 18-1. 事件源位域选择。](#)**18.4.25. DMAMUX 事件互连寄存器 3 (EVIC_DMAMUX_3)**

地址偏移: 0x70

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问。



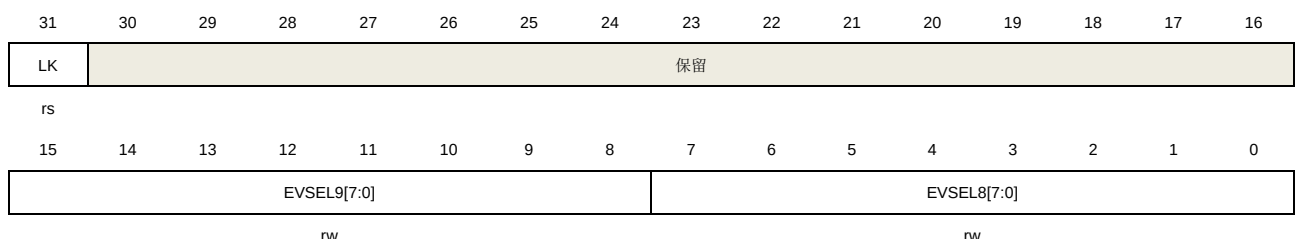
位/位域	名称	描述
31	LK	EVIC_DMAMUX_3 寄存器写保护 该位通过软件置位, 通过系统复位清除。该位置位后, EVIC_DMAMUX_3 寄存器为只读。 0: EVIC_DMAMUX_3 寄存器可读可写 1: EVIC_DMAMUX_3 寄存器只读
30:16	保留	必须保持复位值。
15:8	EVSEL7[7:0]	DMAMUX的事件源选择7 这些位用于选择事件源作为 DMAMUX 请求多路复用器输入 7。关于具体配置请参考 表 18-1. 事件源位域选择。
7:0	EVSEL6[7:0]	DMAMUX的事件源选择6 这些位用于选择事件源作为 DMAMUX 请求多路复用器输入 6。关于具体配置请参考 表 18-1. 事件源位域选择。

18.4.26. DMAMUX 事件互连寄存器 4 (EVIC_DMAMUX_4)

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	LK	EVIC_DMAMUX_4 寄存器写保护 该位通过软件置位，通过系统复位清除。该位置位后，EVIC_DMAMUX_4 寄存器为只读。 0: EVIC_DMAMUX_4 寄存器可读可写 1: EVIC_DMAMUX_4 寄存器只读
30:16	保留	必须保持复位值。
15:8	EVSEL9[7:0]	DMAMUX的事件源选择9 这些位用于选择事件源作为 DMAMUX 请求多路复用器输入 9。关于具体配置请参考 表 18-1. 事件源位域选择 。
7:0	EVSEL8[7:0]	DMAMUX的事件源选择8 这些位用于选择事件源作为 DMAMUX 请求多路复用器输入 8。关于具体配置请参考 表 18-1. 事件源位域选择 。

18.4.27. DMAMUX 事件互连寄存器 5 (EVIC_DMAMUX_5)

地址偏移: 0x78

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	EVIC_DMAMUX_5 寄存器写保护 该位通过软件置位，通过系统复位清除。该位置位后，EVIC_DMAMUX_5 寄存器为只读。 0: EVIC_DMAMUX_5 寄存器可读可写 1: EVIC_DMAMUX_5 寄存器只读
30:16	保留	必须保持复位值。
15:8	EVSEL11[7:0]	DMAMUX的事件源选择11 这些位用于选择事件源作为 DMAMUX 请求多路复用器输入 11。关于具体配置请参考 表 18-1. 事件源位域选择 。
7:0	EVSEL10[7:0]	DMAMUX的事件源选择10 这些位用于选择事件源作为 DMAMUX 请求多路复用器输入 10。关于具体配置请参考

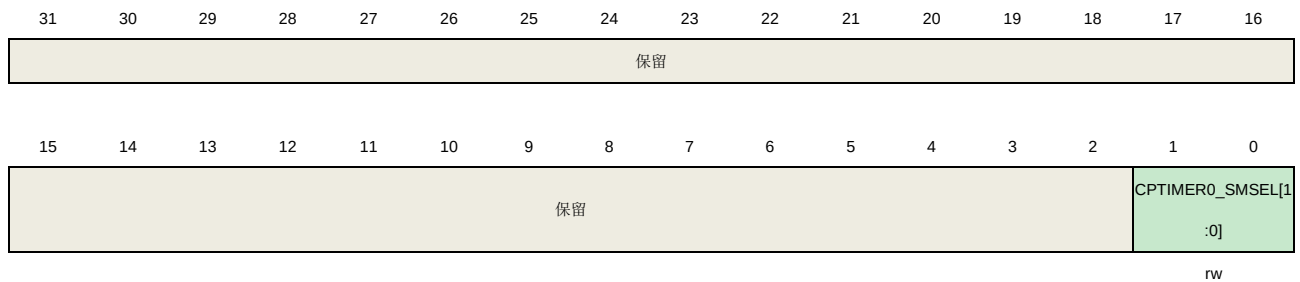
考表 18-1. 事件源位域选择。

18.4.28. 从模式配置寄存器 0 (EVIC_SMCFG0)

地址偏移: 0x7C

复位值: 0x0000 0003

该寄存器只能按字 (32位) 访问。



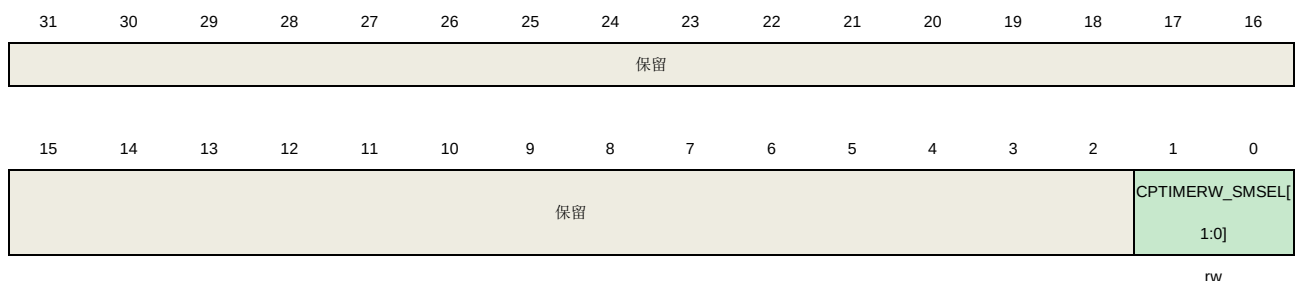
位/位域	名称	描述
31:2	保留	必须保持复位值。
1:0	CPTIMER0_SMSEL[1:0]	CPTIMER0 从模式选择 00: 计数器使能 01: 计数器重新开始 10: 事件计数 11: 事件触发禁能

18.4.29. 从模式配置寄存器 1 (EVIC_SMCFG1)

地址偏移: 0x80

复位值: 0x0000 0003

该寄存器只能按字 (32位) 访问。



位/位域	名称	描述
31:2	保留	必须保持复位值。
1:0	CPTIMERW_SMSEL[1:0]	CPTIMERW 从模式选择 00: 计数器使能

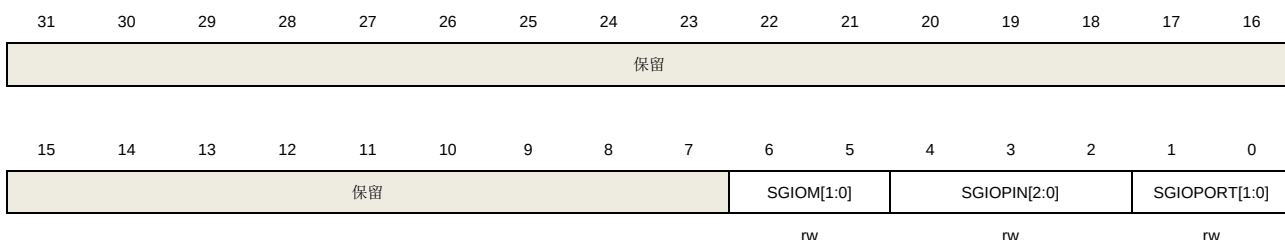
- 01: 计数器重新开始
- 10: 事件计数
- 11: 事件触发禁能

18.4.30. 单个 I/O 配置寄存器 x (EVIC_SGIOCFGx) (x=0...3)

地址偏移: $0x84 + 0x04 * x$ ($x = 0...3$)

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问。



位/位域	名称	描述
31:7	保留	必须保持复位值。
6:5	SGIOM[1:0]	单个 I/O 模式 作为输出引脚: 当事件信号输入到单个 I/O 的输出控制器 x 时, 这些位用于确定所选单个 I/O 引脚输出的电平。 00: 低电平输出 01: 高电平输出 1x: 电平翻转输出 作为输入引脚: 这些位用于指定所选单个 I/O 引脚输入哪个边沿可以生成事件信号。 00: 上升沿 01: 下降沿 1x: 双边沿
4:2	SGIOPIN[2:0]	单个 I/O 引脚 当 SGIOPORT[1:0] 位域设置为 2'b00 时, 这些位用于指定 PE8 到 PE14 中的哪个引脚将被选为单个 I/O 引脚。当 SGIOPORT[1:0] 位域设置为 2'b01 时, 这些位用于指定 PF8 到 PF14 中的哪个引脚将被选为单个 I/O 引脚。 000: PE8 或者 PF8 001: PE9 或者 PF9 010: PE10 或者 PF10 011: PE11 或者 PF11 100: PE12 或者 PF12 101: PE13 或者 PF13

11x: PE14 或者 PF14

1:0 SGIOPORT[1:0] 单个 I/O 端口
 00: 端口E
 01: 端口F
 1x: 禁止设置

18.4.31. 组 E 配置寄存器 (EVIC_GRPECFG)

地址偏移: 0x94

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	PE14SEL	PE13SEL	PE12SEL	PE11SEL	PE10SEL	PE9SEL	PE8SEL	保留		OVWEN	GRPEOSEL[2:0]		GRPEIND[1:0]		
	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	PE14SEL	PE14引脚选择 0: 未被选为组E成员. 1: 被选为组E成员
13	PE13SEL	PE13引脚选择 0: 未被选为组E成员. 1: 被选为组E成员
12	PE12SEL	PE12引脚选择 0: 未被选为组E成员. 1: 被选为组E成员
11	PE11SEL	PE11引脚选择 0: 未被选为组E成员. 1: 被选为组E成员
10	PE10SEL	PE10引脚选择 0: 未被选为组E成员. 1: 被选为组E成员
9	PE9SEL	PE9引脚选择 0: 未被选为组E成员. 1: 被选为组E成员
8	PE8SEL	PE8引脚选择

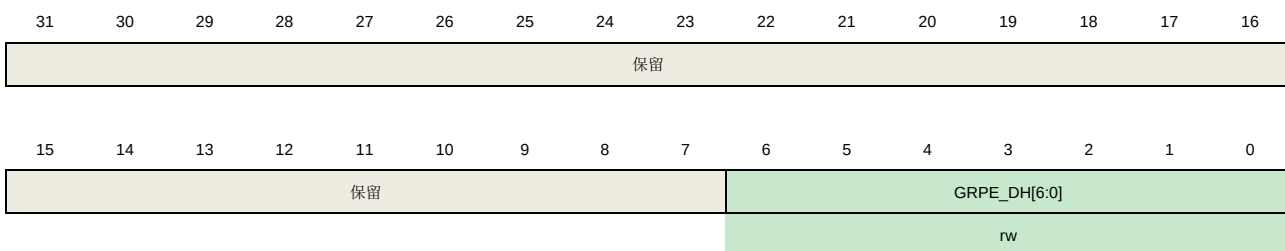
		0: 未被选为组E成员. 1: 被选为组E成员
7:6	保留	必须保持复位值。
5	OVWEN	EVIC_GRPEDH寄存器覆写使能 该位决定在当前数据未被读取时，EVIC_GRPEDH寄存器中相应的位是否可以加载新的数据。 0: 覆写禁能 1: 覆写使能
4:2	GRPEOSEL[2:0]	组E输出选择 这些位用于确定当事件信号输入到组E的I/O输出控制器时，输出组E的I/O引脚输出的电平。 000: 低电平输出 001: 高电平输出 010: 翻转电平输出 011: EVIC_GRPEDH寄存器数据作为输出 1xx: EVIC_GRPEDH寄存器数据循环移位输出（从MSB到LSB）
1:0	GRPEIND[1:0]	组E输入检测 这些位用于确定输入组E的I/O引脚输入哪个边沿可以产生事件信号。 00: 上升沿 01: 下降沿 1x: 双边沿

18.4.32. 组 E 数据保持寄存器（EVIC_GRPEDH）

地址偏移：0x98

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。
6:0	GRPE_DH[6:0]	组E保持数据 输出组E的I/O引脚：如果EVIC_GPRECFG寄存器中的GRPEOSEL[2:0]位域是3'b011或3'b1xx，这些位确定当事件触发时要传输到GPIOE_OCTL寄存器的数据。 输入组E的I/O引脚：这些位确定当事件触发时加载相应引脚的输入电平值的数据。

GRPE_DH[0]: PE8
 GRPE_DH[1]: PE9
 GRPE_DH[2]: PE10
 GRPE_DH[3]: PE11
 GRPE_DH[4]: PE12
 GRPE_DH[5]: PE13
 GRPE_DH[6]: PE14

注意：向输入组E的I/O引脚对应的EVIC_GRPEDH寄存器位写值是禁能的。

18.4.33. 组 F 配置寄存器（EVIC_GPRFCFG）

地址偏移：0x9C

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	PF14SEL	PF13SEL	PF12SEL	PF11SEL	PF10SEL	PF9SEL	PF8SEL	保留		OVWEN	GRPFOSEL[2:0]		GRPFIND[1:0]		
	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	PF14SEL	PF14引脚选择 0: 未被选为组F成员. 1: 被选为组F成员
13	PF13SEL	PF13引脚选择 0: 未被选为组F成员. 1: 被选为组F成员
12	PF12SEL	PF12引脚选择 0: 未被选为组F成员. 1: 被选为组F成员
11	PF11SEL	PF11引脚选择 0: 未被选为组F成员. 1: 被选为组F成员
10	PF10SEL	PF10引脚选择 0: 未被选为组F成员. 1: 被选为组F成员
9	PF9SEL	PF9引脚选择 0: 未被选为组F成员.

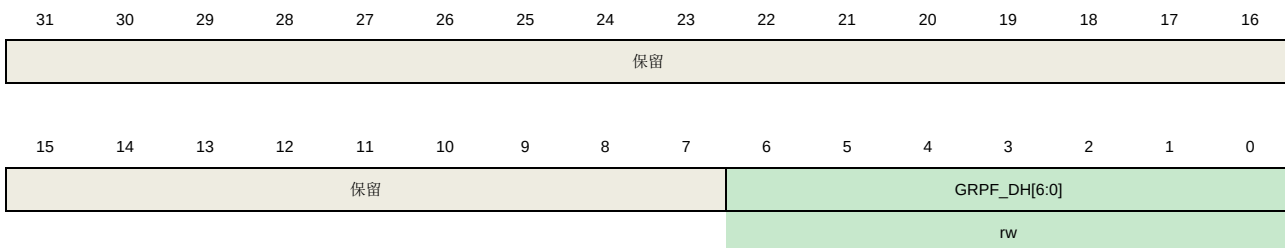
		1: 被选为组F成员
8	PF8SEL	PF8引脚选择 0: 未被选为组F成员. 1: 被选为组F成员
7:6	保留	必须保持复位值。
5	OVWEN	EVIC_GRPFDH寄存器覆写使能 该位决定在当前数据未被读取时，EVIC_GRPFDH寄存器中相应的位是否可以加载新的数据。 0: 覆写禁能 1: 覆写使能
4:2	GRPFOSEL[2:0]	组F输出选择 这些位用于确定当事件信号输入到组F的I/O输出控制器时，输出组F的I/O引脚输出的电平。 000: 低电平输出 001: 高电平输出 010: 翻转电平输出 011: EVIC_GRPEDH寄存器数据作为输出 1xx: EVIC_GRPEDH寄存器数据循环移位输出（从MSB到LSB）
1:0	GRPFIND[1:0]	组F输入检测 这些位用于确定输入组F的I/O引脚输入哪个边沿可以产生事件信号。 00: 上升沿 01: 下降沿 1x: 双边沿

18.4.34. 组 F 数据保持寄存器（EVIC_GRPFDH）

地址偏移：0x100

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。
6:0	GRPF_DH[6:0]	组F保持数据 输出组F的I/O引脚：如果EVIC_GPRFCFG寄存器中的GRPFOSEL[2:0]位域是

3'b011或3'b1xx，这些位确定当事件触发时要传输到GPIOF_OCTL寄存器的数据。

输入组F的I/O引脚：这些位确定当事件触发时加载相应引脚的输入电平值的数据。

GRPF_DH[0]: PF8

GRPF_DH[1]: PF9

GRPF_DH[2]: PF10

GRPF_DH[3]: PF11

GRPF_DH[4]: PF12

GRPF_DH[5]: PF13

GRPF_DH[6]: PF14

注意：向输入组F的I/O引脚对应的EVIC_GRPFDH寄存器位写值是禁能的。

19. 空间矢量脉宽调制（SVPWM）

19.1. 简介

SVPWM（空间矢量脉宽调制）算法单元用于控制特殊开关触发序列和脉冲宽度组合下的三相电压源逆变器功率装置中的交流感应电机或永磁同步电机。这种开关触发序列和组合将在定子线圈中产生三个具有电气角度差 120° 的正弦电流。

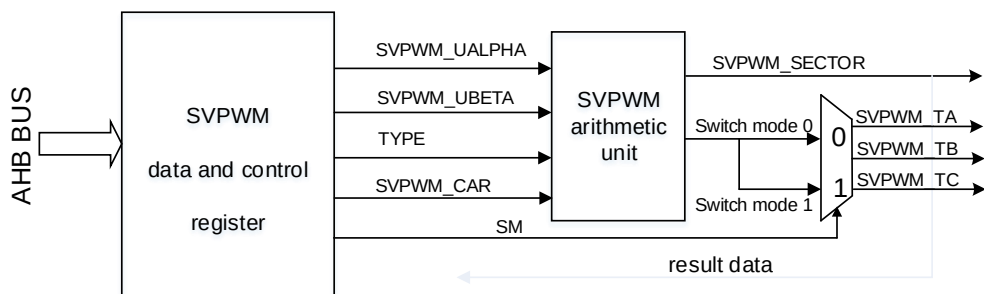
19.2. 主要特征

- 该算法包含七段式PWM和五段式PWM。
- 三相PWM波直接由空间电压矢量合成
- 浮点运算
- 支持合成电压矢量过调制处理。

19.3. 结构框图

[图 19-1. SVPWM 模块框图](#)提供了 SVPWM 模块的内部配置详细信息。

图 19-1. SVPWM 模块框图



19.4. 功能说明

时钟控制器提供的 SVPWM 时钟与 AHB 时钟同步。

19.4.1. 数据格式配置

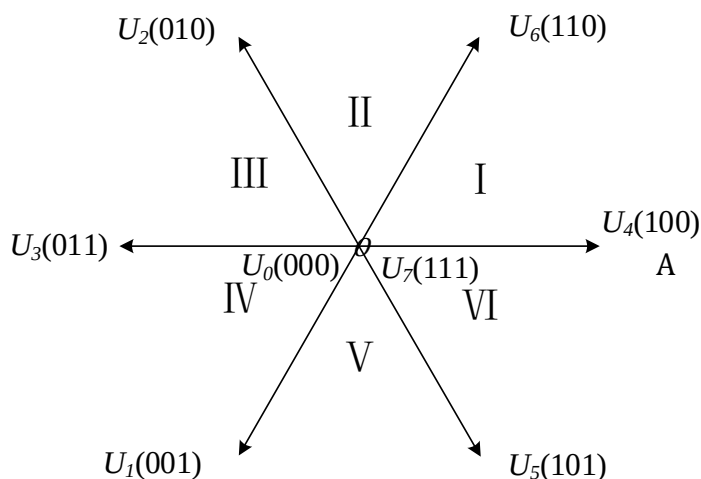
svpwm 模块的输入数据是 IEEE 32 位单精度浮点格式。SVPWM_UALPHA、SVPWM_UBETA 寄存器用于配置输入数据。

可以通过设置 SVPWM_CAR 寄存器来配置 PWM 周期的数量。svpwm 模块的输出数据是 16 位无符号数。

19.4.2. 空间矢量脉宽调制原理

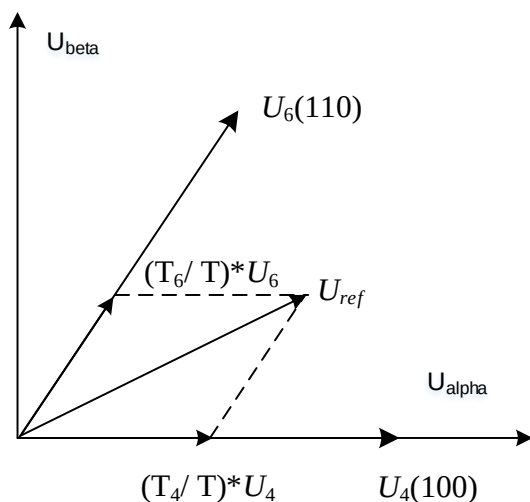
三相电机电压信号的脉冲宽度调制信号是由 SVPWM 算法生成的。三相逆变器的每个相位输出可以是两种状态之一，即逆变器的输出可以连接到母线电压或接地，当连接母线电压时，用 1 表示，当接地时，用 0 表示，因此有八种可能的状态 (000,001,010,011,100,101,110,111)。两种状态 000 和 111 被视为无效状态，因为任何两个相位之间没有线电压。其余的六个状态被表示为具有 60 度旋转间隔的电压矢量。

图 19-2. SVPWM 输出基本电压矢量



空间矢量调制过程允许任何空间电压矢量由两个相邻矢量的分量之和来表示。以第一象限为例，期望的电压矢量 U_{ref} 由有效矢量 $U_4(100)$ 、 $U_6(110)$ 和无效矢量 $U_0(000)$ 、 $U_7(111)$ 组成。

图 19-3. 平均矢量调制



根据伏秒平衡原理。

$$\begin{cases} U_{ref} * T = U_4 * T_4 + U_6 * T_6 \\ T = T_4 + T_6 + T_0^* \end{cases} \quad (1-1)$$

其中， T 是 SVPWM 的开关周期， T_4 和 T_6 是有效向量 $U_4(100)$ 、 $U_6(110)$ 的工作时间， T_0^* 是无效

向量 $U_0(000)$, $U_7(111)$ 的工作时间。

19.4.3. 空间矢量脉宽调制实现

首先，静态二相电压 U_{alpha} 和 U_{beta} 被转换为中间电压变量 U_{s1} U_{s2} U_{s3} 。

$$\begin{cases} U_{s1} = U_{beta} \\ U_{s2} = (-U_{beta} + \sqrt{3}U_{alpha})/2 \\ U_{s3} = (-U_{beta} - \sqrt{3}U_{alpha})/2 \end{cases} \quad (1-2)$$

定义三个变量 a 、 b 、 c ，如果 $U_{s1} > 0$ ，则 $a=1$ ，否则 $a=0$ ；如果 $U_{s2} > 0$ ，则 $b=1$ ，否则 $b=0$ ；如果 $U_{s3} > 0$ ，则 $c=1$ ，否则 $c=0$ ；定义另一个变量 $N=a+b*2+c*4$ ，根据不同的 sec 将空间划分为 6 个区域。

附加定义：

$$\begin{cases} X = \sqrt{3}U_{beta} \\ Y = \sqrt{3}(U_{beta} + \sqrt{3}U_{alpha})/2 \\ Z = \sqrt{3}(U_{beta} - \sqrt{3}U_{alpha})/2 \end{cases} \quad (1-3)$$

在第 I 扇区， T_1 等于矢量 $U_4(100)$ 的作用时间 T_4 ， T_2 等于矢量 $U_6(110)$ 的作用时间 T_6 。在第 II 扇区， T_1 等于矢量 $U_2(010)$ 的作用时间 T_2 ， T_2 等于矢量 $U_6(110)$ 的作用时间 T_6 。在第 III 扇区， T_1 等于矢量 $U_2(010)$ 的作用时间 T_2 ， T_2 等于矢量 $U_3(011)$ 的作用时间 T_3 。在第 IV 扇区， T_1 等于矢量 $U_1(001)$ 的作用时间 T_1 ， T_2 等于矢量 $U_3(011)$ 的作用时间 T_3 。在第 V 扇区， T_1 等于矢量 $U_1(001)$ 的作用时间 T_1 ， T_2 等于矢量 $U_5(101)$ 的作用时间 T_5 。在第 VI 扇区， T_1 等于矢量 $U_4(100)$ 的作用时间 T_4 ， T_2 等于矢量 $U_5(101)$ 的作用时间 T_5 。

不同扇区下 T_1 和 T_2 的对应关系为：

表 19-1. 不同扇区与作用时间对应关系

N	1	2	3	4	5	6
扇区	II	VI	I	IV	III	V
T_1	Z	Y	-Z	-X	X	-Y
T_2	Y	-X	X	Z	-Y	-Z
T_0^*	$T - T_1 - T_2$	$T - T_1 - T_2$	$T - T_1 - T_2$	$T - T_1 - T_2$	$T - T_1 - T_2$	$T - T_1 - T_2$

在 PWM 周期 T 中，向量 T_1 的输出时间为 T_1/T ，向量 T_2 的输出时间为 T_2/T ，其余时间输出无效矢量。无效矢量可以由 $U_0(000)$ 组成，这是一种五段式 SVPWM。也可以由 $U_7(111)$ 在中间和 $U_0(000)$ 在两侧组成，这是一种七段式 SVPWM。

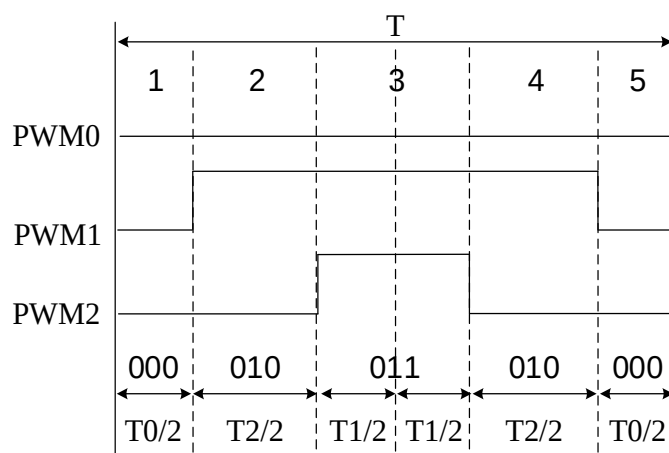
19.4.4. 五段式 SVPWM

在一个 PWM 周期内，五段式 SVPWM 会在中间插入零矢量，导致转矩波动较大，低频时运行不稳定。在每个 PWM 周期中，总有一个相桥臂的开关阶段不需要改变，但在七段式 SVPWM 中，每个相桥臂都需要切换一次。五段桥臂的开关次数比七段桥臂少 $1/3$ ，因此五段的开关功耗最低。

通过设置 SVPWM_CTL0 寄存器中的 TYPE 位，可以选择五段式 SVPWM 方法。下图显示了

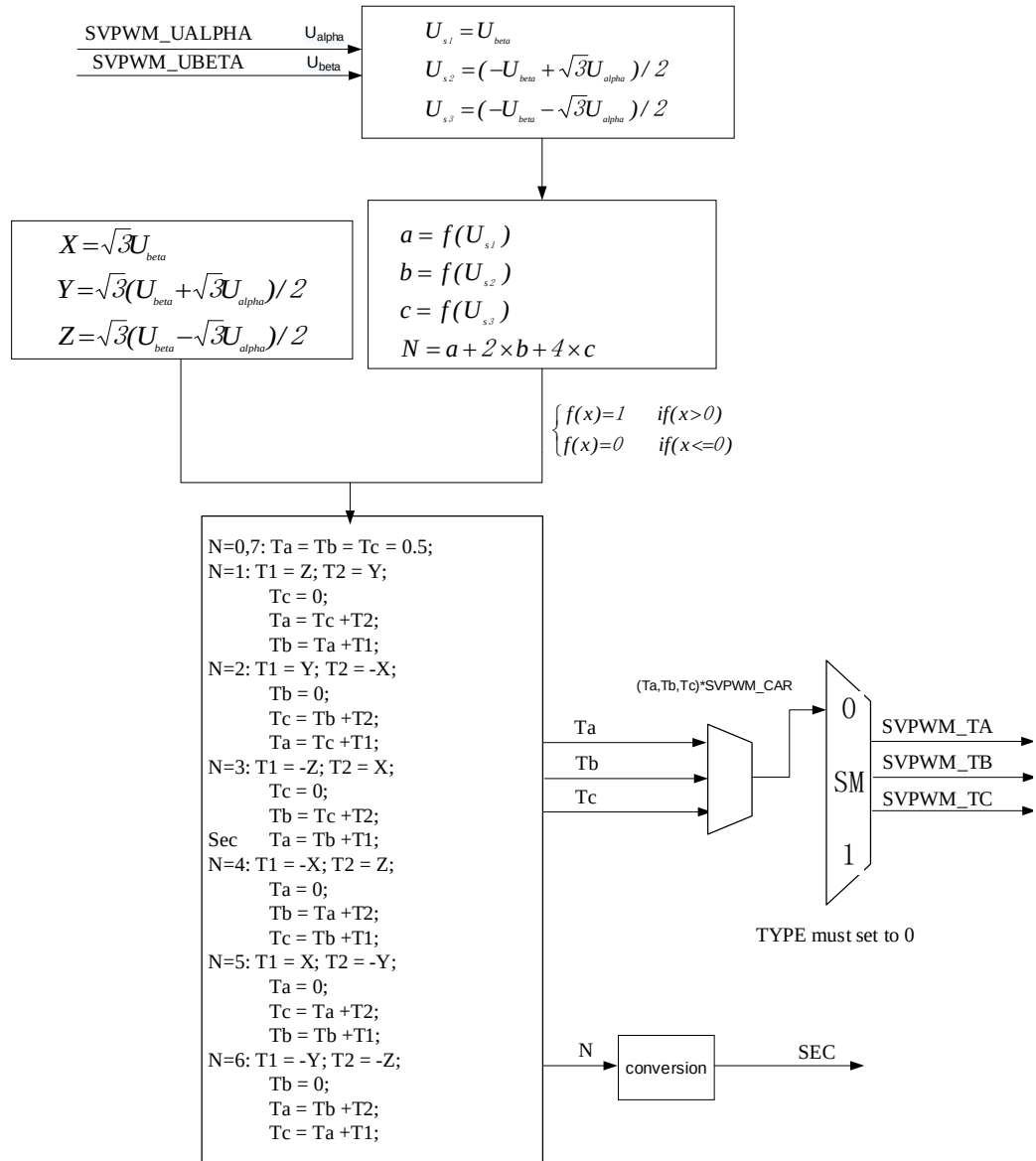
五段式 SVPWM 波形，其中 PWM 信号配置为使输出中央对齐。

图 19-4. 五段式 SVPWM 输出波形图



五段式 SVPWM 算法如图示。Ta、Tb 和 Tc 可以根据 U_{alpha} 和 U_{beta} 计算得出，输入和输出的数据格式可以设置。

图 19-5. 五段式 SVPWM 算法



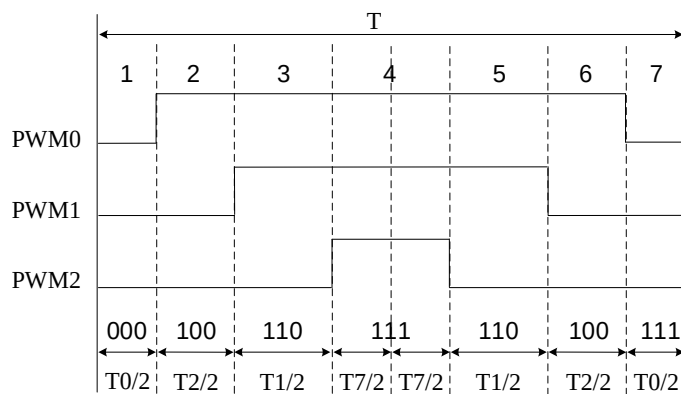
U_{α} , U_{β} , T1, T2, Ta, Tb, Tc 是浮点数据, SECTOR 是无符号的 8 位数字。“conversion”是一个数组, $\text{conversion}[] = \{0, 1, 5, 0, 3, 2, 4, 0\}$ 。并且, 当 SEC 是 0 时 SECTOR 是 I, 当 SEC 是 1 时 SECTOR 是 II, 当 SEC 是 2 时 SECTOR 是 III, 当 SEC 是 3 时 SECTOR 是 IV, 当 SEC 是 4 时 SECTOR 是 V, 当 SEC 是 5 时 SECTOR 是 VI。

19.4.5. 七段式 SVPWM

在七段式 SVPWM 中, 一半的零向量被插入到 PWM 周期的中间, 另一半被插入到 PWM 周期的两侧, 这使得磁通操作更加稳定, 减少了电机转矩波动, 从而使电机在低频时的特性明显优于五段法, 而高频时的特性则没有不同。

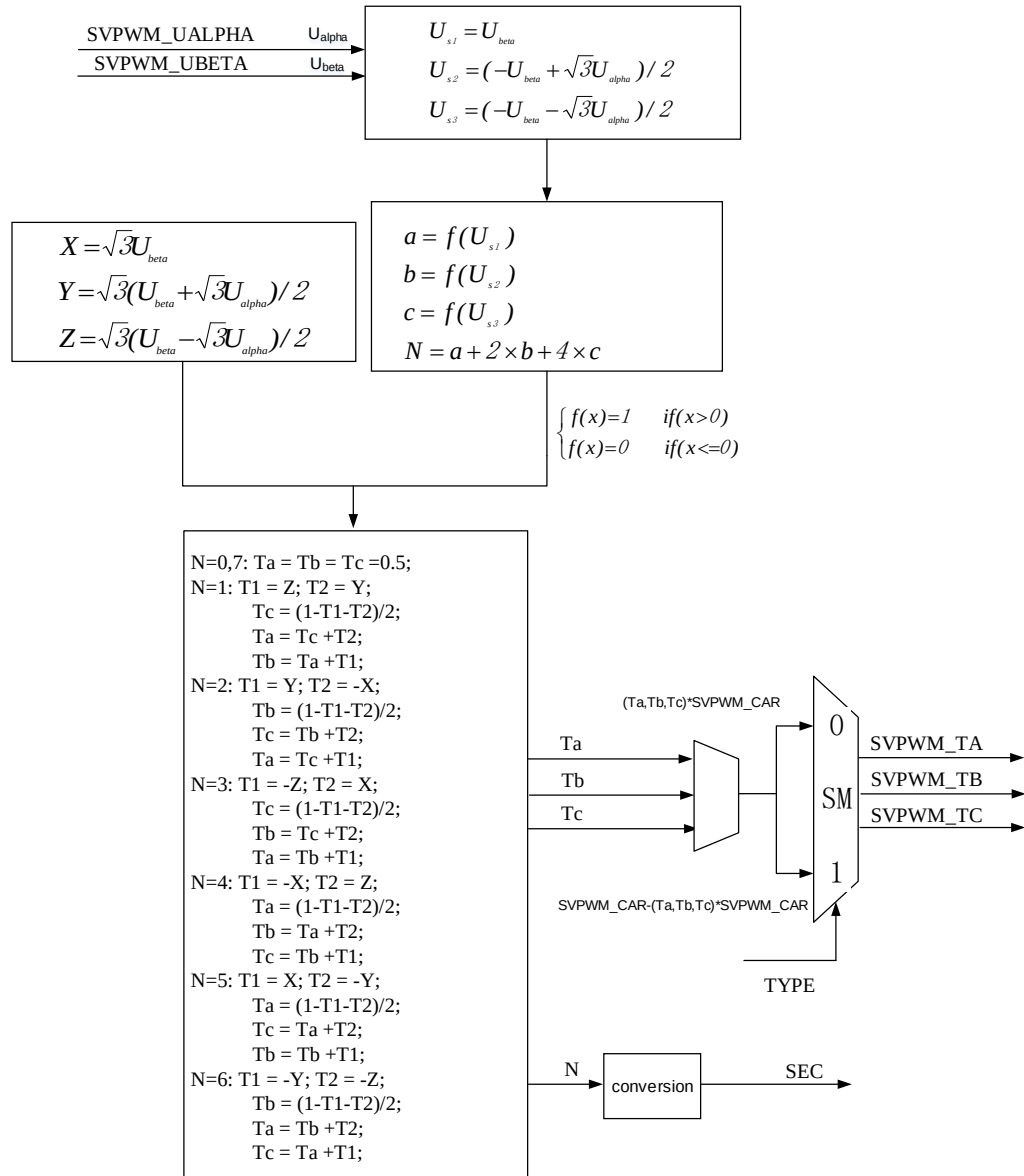
通过设置 SVPWM_CTL0 寄存器中的 TYPE 位, 可以选择七段 SVPWM 方法。下图显示了七段 SVPWM 波形, 其中 PWM 信号配置为使输出中央对齐。

图 19-6. 七段式 SVPWM 输出波形图



七段式 SVPWM 算法如图示。Ta、Tb 和 Tc 可以根据 U_{alpha} 和 U_{beta} 计算得出，输入和输出的数据格式可以设置。

图 19-7. 七段式 SVPWM 算法



U_{α} , U_{β} , T1, T2, Ta, Tb, Tc 是浮点数据, SECTOR 是无符号的 8 位数字。“conversion”是一个数组, $\text{conversion}[] = \{0, 1, 5, 0, 3, 2, 4, 0\}$ 。并且, 当 SEC 是 0 时 SECTOR 是 I, 当 SEC 是 1 时 SECTOR 是 II, 当 SEC 是 2 时 SECTOR 是 III, 当 SEC 是 3 时 SECTOR 是 IV, 当 SEC 是 4 时 SECTOR 是 V, 当 SEC 是 5 时 SECTOR 是 VI。

19.4.6. 过调制处理

在五段式 SVPWM 或七段式 SVPWM 中, 可能存在计算溢出。当 SVPEM_TA、SVPEM_TB、SVPEM_TC 的任何计算结果超出范围时, SVPWM 状态寄存器中的溢出标志(OCH、OBH、OAH、OCL、OBL、OAL) 将被置 1。

19.5. SVPWM 寄存器

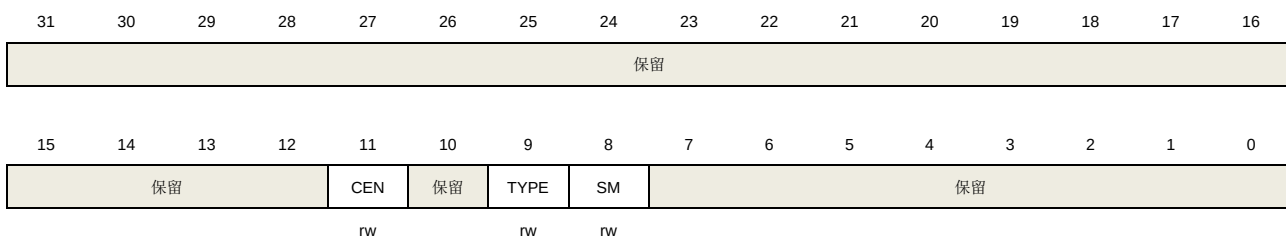
SVPWM 基地址：0x4802 4000

19.5.1. 控制寄存器 0（SVPWM_CTL0）

偏移地址：0x00

系统复位值：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	CEN	SVPWM 外设使能 0: SVPWM 转换已完成或尚未开始 1: 启动 SVPWM 转换。在转换期间，标志为 1。操作完成后，硬件将标志复位。 注意：此位由软件设置，由硬件复位。
10	保留	必须保持复位值。
9	TYPE	五段式 SVPWM 和七段式 SVPWM 选择标志位 0: 五段式 SVPWM 1: 七段式 SVPWM
8	SM	开关模式 0: 开关模式 0。 1: 开关模式 1。 注意：当 SVPWM 外设被使能（CEN=1）时，该位不能被写入，并且只能在七段式 SVPWM 中使用，在五段式 SVPWM 中必须设置为 0。
7:0	保留	必须保持复位值。

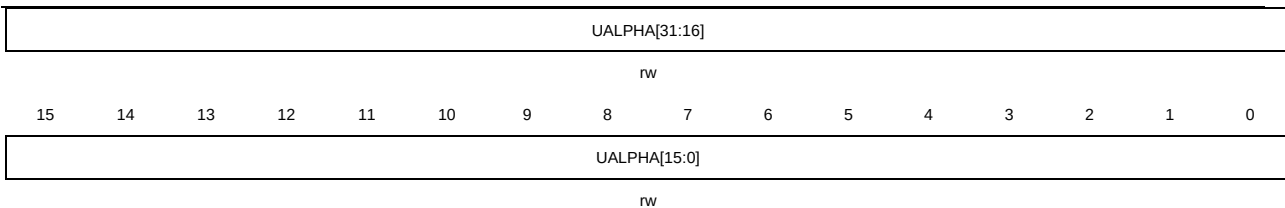
19.5.2. Ualpha 寄存器（SVPWM_UALPHA）

偏移地址：0x04

系统复位值：0x3F80 0000

该寄存器只能按字（32位）访问。





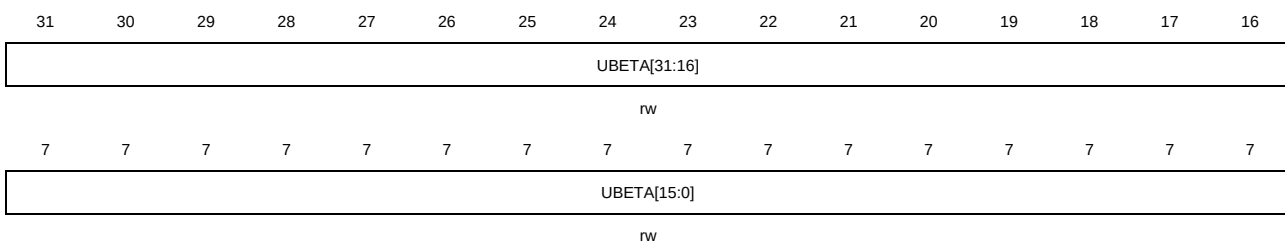
位/位域	名称	描述
31:0	UALPHA[31:0]	这些数据表示 SVPWM 模块的 UALPHA 输入 注意：当 SVPWM 外设被使能（CEN=1）时，这个位字段不能被写入，并且 UALPHA[31:0]必须符合 IEEE 32 位单精度浮点格式。

19.5.3. Ubeta 寄存器（SVPWM_UBETA）

偏移地址：0x08

系统复位值：0x3F80 0000

该寄存器只能按字（32位）访问。



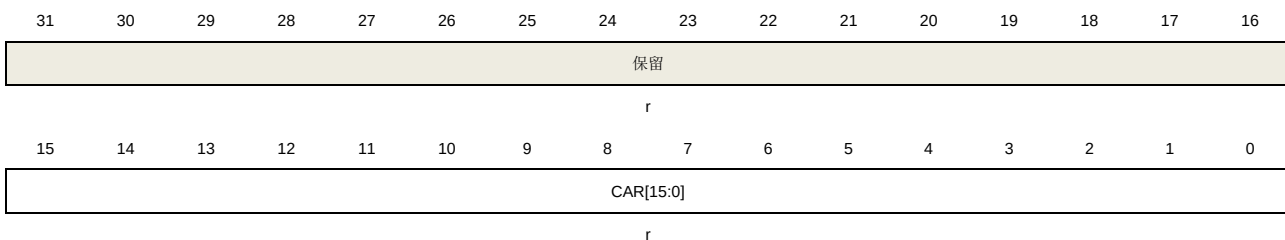
位/位域	名称	描述
31:0	UBETA[31:0]	这些数据表示 SVPWM 模块的 UBETA 输入 注意：当 SVPWM 外设被使能（CEN=1）时，这个位字段不能被写入，并且 UBETA [31:0]必须符合 IEEE 32 位单精度浮点格式。

19.5.4. 自动重载寄存器（SVPWM_CAR）

偏移地址：0x0C

系统复位值：0x0000 0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述

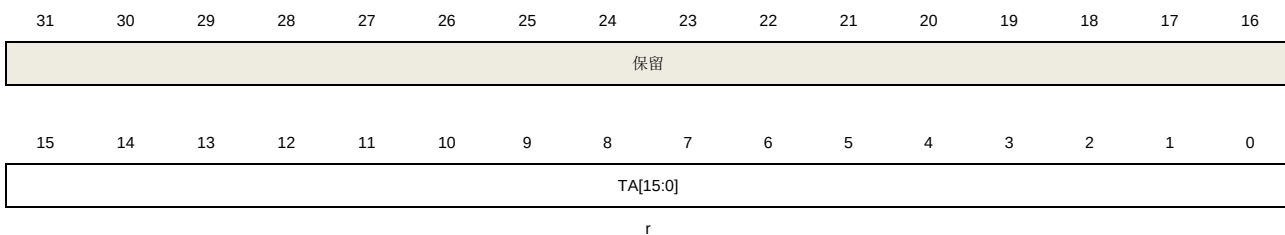
31:16	保留	必须保持复位值。
15:0	CAR[15:0]	这个位字段表示定时器 TIMER 的周期。 注意：当 SVPWM 外设被使能（CEN=1）时，这个位字段不能被写入，它是一个 16 位无符号整数。

19.5.5. Ta 寄存器（SVPWM_TA）

偏移地址：0x10

系统复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



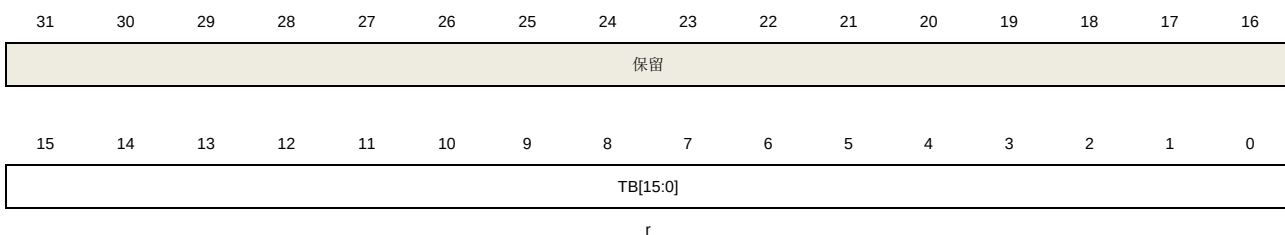
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	TA	这些数据表示 SVPWM 模块的输出。

19.5.6. Tb 寄存器（SVPWM_TB）

偏移地址：0x14

系统复位值：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	TB	这些数据表示 SVPWM 模块的输出。

19.5.7. Tc 寄存器（SVPWM_TC）

偏移地址：0x18

系统复位值：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
TC[15:0]															
r															

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	TC	这些数据表示 SVPWM 模块的输出。

19.5.8. 扇区寄存器（SVPWM_SEC）

偏移地址：0x1C

系统复位值：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													SEC [2:0]		
rw															

位/位域	名称	描述
31:3	保留	必须保持复位值。
2:0	SEC[2:0]	SVPWM 模块的扇区值 当 SEC 为 0 时，SECTOR 是 I；当 SEC 为 1 时，SECTOR 是 II；当 SEC 为 2 时，SECTOR 是 III；当 SEC 为 3 时，SECTOR 是 IV；当 SEC 为 4 时，SECTOR 是 V；当 SEC 为 5 时，SECTOR 是 VI。

19.5.9. 状态寄存器（SVPWM_STAT）

偏移地址：0x20

系统复位值：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													OCH	OBH	OAH
													r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									OSF	保留			OCL	OBL	OAL
													r	r	r

位/位域	名称	描述
31:19	保留	必须保持复位值。
18	OCH	C 相的占空比大于自动重载值（SVPWM_CAR）溢出标志。 0: 结果没有溢出 1: 结果溢出 注意：当发生溢出事件时，这个位由硬件设置，每次计算完成后更新一次。
17	OBH	B 相的占空比大于自动重载值（SVPWM_CAR）溢出标志。 0: 结果没有溢出 1: 结果溢出 注意：当发生溢出事件时，这个位由硬件设置，每次计算完成后更新一次。
16	OAH	A 相的占空比大于自动重载值（SVPWM_CAR）溢出标志。 0: 结果没有溢出 1: 结果溢出 注意：当发生溢出事件时，这个位由硬件设置，每次计算完成后更新一次。
15:7	保留	必须保持复位值。
6	OSF	SVPWM 运行状态标志位。 当 SVPWM 算法运行完成时，这个位由硬件设置。更新 SVPWM_UALPHA 或 SVPWM_UBETA 后，硬件会自动清零。 0: SVPWM 运行未完成 1: SVPWM 运行已完成
5:3	保留	必须保持复位值。
2	OCL	C 相的占空比小于 0 溢出标志。 0: 结果没有溢出 1: 结果溢出 注意：当发生溢出事件时，这个位由硬件设置，每次计算完成后更新一次。
1	OBL	B 相的占空比小于 0 溢出标志。 0: 结果没有溢出 1: 结果溢出 注意：当发生溢出事件时，这个位由硬件设置，每次计算完成后更新一次。
0	OAL	A 相的占空比小于 0 溢出标志。 0: 结果没有溢出 1: 结果溢出 注意：当发生溢出事件时，这个位由硬件设置，每次计算完成后更新一次。

20. 看门狗定时器（WDGT）

看门狗定时器（WDGT）是一种硬件定时电路，可用于检测由于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，会触发一个复位或中断。当处理器工作在调试模式的时候看门狗定时器定时计数器可以停止计数。

20.1. 独立看门狗定时器（FWDGT）

20.1.1. 简介

独立看门狗定时器（FWDGT）有独立的时钟源（IRC32K）。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（或中断）。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。

20.1.2. 特征

- 自由运行的12位向下计数器；
- 如果看门狗定时器被使能，在以下两种情况会产生复位或中断：
 - 当向下计数器的值达到0时产生复位或中断；
 - 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或中断。
- 独立时钟源，独立看门狗定时器在主时钟停止（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；
- 通过配置FWDGSPD_STDBY或FWDGSPD_DPSLP，在待机模式或深度睡眠模式中，FWDGT可以停止工作或唤醒控制器继续工作；
- 可配置的中断或复位请求

20.1.3. 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 20.1. 独立看门狗定时器框图](#)的独立看门狗定时器的功能模块。

[illegible]

在任何时候向 FWDGT_CTL 中写 0xAAAA 都可以重装计数器，重装值来源于重装寄存器（FWDGT_RLD）。软件可以在计数器计数值达到 0x000 之前可以通过重装计数器来阻止看门狗定时器产生系统复位或中断。

独立看门狗定时器也能够工作在窗口看门狗定时器模式下，只要在 FWDGT_WND 寄存器中设置适当的值即可。如果重加载操作执行的同时，看门狗定时器计数器的值大于窗口寄存器（FWDGT_WND）中存储的值，也会引起系统复位或中断。FWDGT_WND 的默认值是 0x0000 0FFF，所以如果没有改写它，那么窗口选项默认是关闭的。窗口值一旦改变，立即就会引起看门狗定时器计数器的一次重加载动作，将向下递减计数器置为 FWDGT_RLD 中的值，并复位预分频计数器。

如果在选项字节中打开了“硬件使能独立看门狗功能硬件”功能，那么在上电的时候看门狗定时器就被自动打开。在复位信号状态下，选项字节中的配置值会重新加载到独立看门狗定时器的寄存器中（包括 RLD / PSC / WND）。为了避免系统复位或中断，软件应该在计数器达到 0x000 之前重装计数器。

FWDGT_PSC 寄存器、FWDGT_RLD 寄存器和 FWDGT_RLD 寄存器都有写保护功能。在写数据到这些寄存器之前，需要写 0x5555 到 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，FWDGT_WND 或者 FWDGT_RLD 更新时，FWDGT_STAT 寄存器对应的状态位会被置 1。

如果 DBG 中控制寄存器 (DBG_CTL) 中 FWDGTx_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 FWDGTx_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 20-1. 独立看门狗定时器在 32 kHz（IRC32K）时的最小/最大超时周期

预分频系数	PSC[2:0] 位	最小超时 (ms) RLD[11:0]=0x000	最大超时 (ms) RLD[11:0]=0xFFFF
1 / 4	000	0.125	512
1 / 8	001	0.25	1024
1 / 16	010	0.5	2048
1 / 32	011	1.0	4096
1 / 64	100	2.0	8192
1 / 128	101	4.0	16384
1 / 256	110 or 111	8.0	32768

通过校准 IRC32K 可以使独立看门狗定时器超时更加准确。

注意：

1. 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 IRC32K 时钟周期；
2. 两次连续喂狗之间需插入 7 个及以上 IRC32K 时钟周期。

20.1.4. FWDGT 寄存器

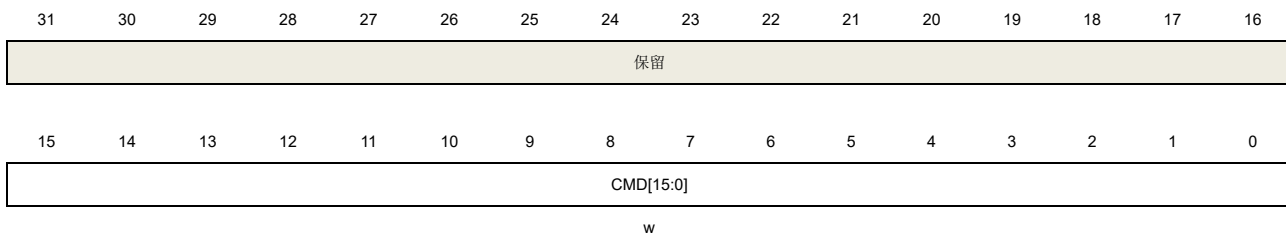
FWDGT 基地址：0x4000 3000

控制寄存器（FWDGT_CTL）

地址偏移：0x00

复位值：0x0000 0000

这个寄存器可以通过半字（16 位）或字（32 位）访问。



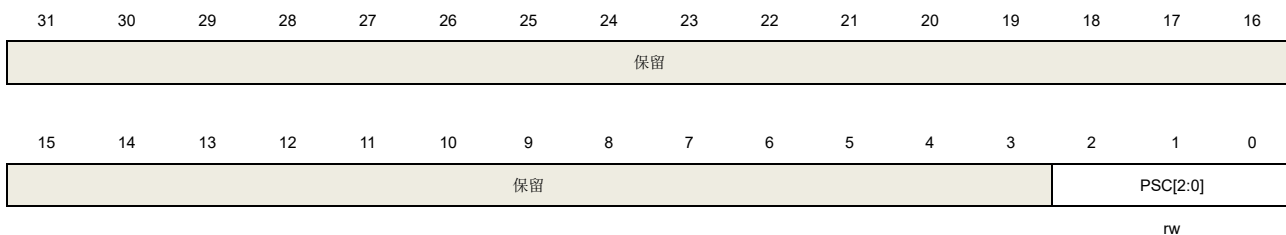
位/位域	名称	说明
31:16	保留	必须保持复位值。
15:0	CMD[15:0]	只可写，写入不同的值来产生不同的功能 0x5555: 关闭FWDGT_PSC、FWDGT_RLD和FWDGT_WND的写保护 0xCCCC: 开启独立看门狗定时器定时计数器。计数减到0时产生复位或中断 0xAAAA: 重装载计数器

预分频寄存器（FWDGT_PSC）

地址偏移：0x04

复位值：0x0000 0000

这个寄存器可以通过半字（16 位）或字（32 位）访问。



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

010: 1 / 16

011: 1 / 32

100: 1 / 64

101: 1 / 128

110: 1 / 256

111: 1 / 256

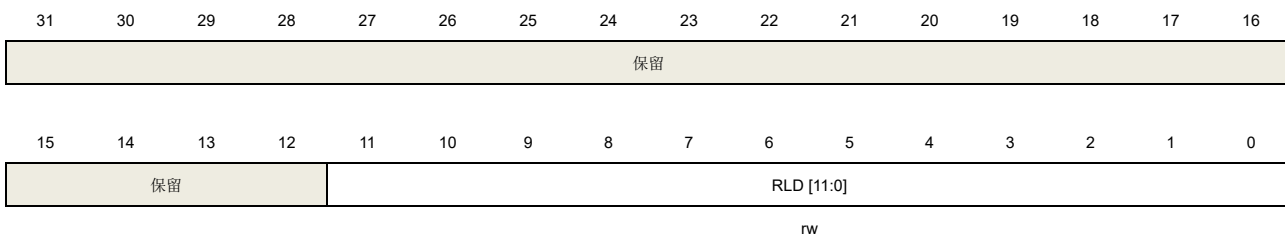
如果应用需要使用不同的预分频系数，改变预分频值之前必须等到PUD位被清0。更新了预分频寄存器中的值后，在代码持续执行之前不必等待PUD值被清零（在进入省电模式前需要等待PUD值清零）。

重加载寄存器（FWDGT_RLD）

地址偏移：0x08

复位值：0x0000 0FFF

这个寄存器可以通过半字（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装载值，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装载值，改变重加载值之前必须等到RUD位被清0。更新了重加载寄存器的值后，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

这个寄存器可以通过半字（16 位）或字（32 位）访问。



保留	WUD	RUD	PUD
	r	r	r

位/位域	名称	说明
31:3	保留	必须保持复位值。
2	WUD	独立看门狗定时器计数器窗口值更新 FWDGT_WND 寄存器写操作时, 该位被置 1, 此时读取 FWDGT_WND 寄存器的任何值都是无效的。
1	RUD	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。

窗口寄存器 (FWDGT_WND)

地址偏移: 0x10

复位值: 0x0000 0FFF

这个寄存器可以通过半字 (16 位) 或字 (32 位) 来访问。

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

位/位域	名称	说明
31:12	保留	必须保持复位值。
11:0	WND	独立看门狗定时器计数器窗口值。这些位将用来将窗口值的上限值与向下递减计数器进行比较。当计数值大于WND[19:0]中值, 重装载操作会引起复位或中断, 若要改变重装载值, FWDGT_STAT 寄存器中的WUD位必须保持复位状态。 这些位有写保护功能。在写这些位之前需向FWDGT_CTL寄存器中写0x5555。 如果应用需要使用几个窗口值, 改变窗口值之前必须等到 WUD 位被清 0。除了在进入低功耗模式下, 更新了窗口值后, 在代码持续执行之前不必等待 WUD 值被清零。

复位控制寄存器 (FWDGT_RSTCTL)

地址偏移: 0x14

复位值: 0x0000 0001

这个寄存器可以通过半字 (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														RSTREQ EN	
rw															

位/位域	名称	说明
31:1	保留	必须保持复位值。
0	RSTREQEN	独立看门狗复位请求使能 0: 使能中断或不可屏蔽中断输出请求 1: 使能复位输出请求 注意: 当选择使能中断或不可屏蔽中断输出请求情况下, 独立看门狗计数器产生中断后, 需重新对控制寄存器 (FWDGT_CTL) 进行重载操作后方可启动独立看门狗计数器。

20.2. 窗口看门狗定时器（WWDGT）

20.2.1. 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，计数器的值会逐渐减少。当计数器达到0（CNT[13:0] = 0）时会产生复位。在计数器计数值达到窗口寄存器开始值之前或在达到窗口寄存器结束值之后，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。当计数器达到0x01时或在允许刷新周期之外写计数器时，窗口看门狗定时器会生成一个提前唤醒状态标志，如果使能中断也将产生中断。

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。

20.2.2. 主要特征

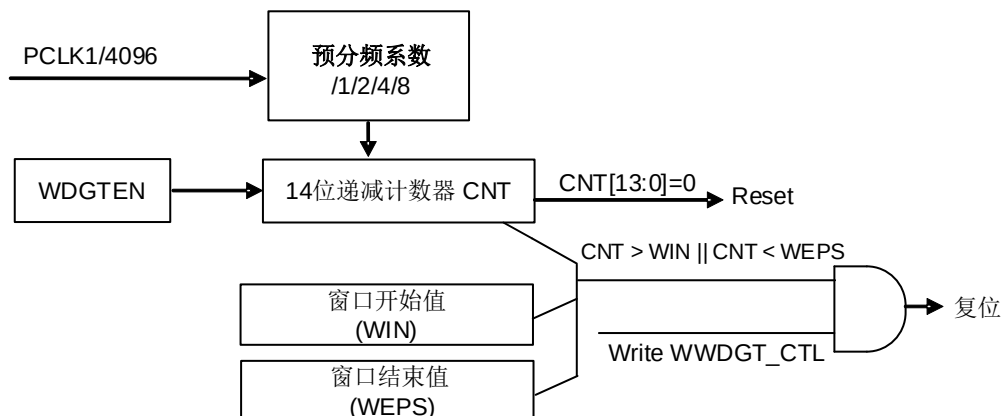
- 可编程的14位自由运行向下递减计数器。
- 两种启动模式：硬件启动模式和软件启动模式
 - 硬件启动：系统重启后自动开始计数；
 - 软件启动：刷新操作发生后开始计数。
- 当窗口看门狗使能后，有以下两种情况会产生复位：
 - 当计数器达到0时产生复位；
 - 当计数器的值大于窗口寄存器开始值或小于窗口寄存器结束值时，更新计数器会产生复位。
- 提前唤醒中断（EWI）：如果看门狗定时器打开，支持中断，当计数值达到0x01时或在允许刷新周期之外写计数器时，会产生中断。
- 可以配置窗口看门狗定时器在调试模式下选择停止还是继续工作。

20.2.3. 功能说明

框图

窗口看门狗定时器如[图 20-2. 窗口看门狗定时器框图](#)所示。

图 20-2. 窗口看门狗定时器框图



软件启动模式

当选项字节中的 `nWWDG_HW` 位为 1 时，选择软件启动模式并启用 `WWDGT_CFG` 寄存器。

复位信号发生后，配置 `WWDGT_CFG` 寄存器，然后在 `WWDGT_CTL` 寄存器中刷新 `CNT[13:0]` 以启动计数器。计数器的速度取决于 `APB1` 时钟和 `WWDGT_CFG` 寄存器中 `PSC[1:0]` 位设置的预分频值。

如果启用了窗口看门狗定时器复位（在 `WWDGT_CTL` 中设置 `WDGTEN` 位），当计数器达到 0（`CNT[13:0] = 0`）时，看门狗定时器将产生复位，或者当计数器大于窗口起始值（由 `WWDGT_CFG` 中的 `WIN` 设置）时刷新计数器，或者当计数器小于窗口结束值（由 `WWDGT_CFG` 中的 `WEPS` 设置）时刷新计数器，看门狗定时器也将引起复位。

如果通过在 `WWDGT_CFG` 寄存器中设置 `EWIE` 位来启用早期唤醒中断（EWI），则当计数器达到 `0x01`（`CNT[13:0] = 0x01`）时，或者在计数器大于窗口开始值（由 `WWDGT_CFG` 中的 `WIN` 设置）时刷新计数器，或者在计数器小于窗口结束值（由 `WWDGT_CFG` 中的 `WEPS` 设置）时刷新计数器，将生成中断。软件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。在这种情况下，`WWDGT` 永远不会生成 `WWDGT` 复位，但可以用于其他事情。通过将 `WWDGT_STAT` 寄存器的 `EWIF` 位写 0 可以清除 EWI 中断。

注意：在软件启动模式下，当中断发生后，`WWDGT_CTL` 中的 `CNT` 会被重置为 0，软件必须重新加载 `CNT` 值以重新启动 `WWDGT` 计数器。

硬件启动模式

当选项字节中的 `nWWDG_HW` 位为 0 时，选择硬件启动模式，选项字节中的配置被启用，而 `WWDGT_CFG` 寄存器被禁用。

在复位信号状态下，选项字节中的配置值会重新加载到 `WWDGT_CFG` 寄存器中（包括 `WIN` / `WEPS` / `EWIE` / `PSC` / `WDGTEN`）。在硬件启动模式下，`WWDGT` 窗口启动值（`WIN`）只能设置为四个固定值。当选项字节中的 `WWDGT_WSPS` 位设置为 `0b00` 时，窗口启动值是 `0x0FFF`（`0x3FFF` 的 25%）。当 `WWDGT_WSPS` 位设置为 `0b01` 时，窗口启动值是 `0x1FFF`（`0x3FFF` 的

50%)。当WWDGT_WSPS位设置为0b10时，窗口启动值是0x2FFF（0x3FFF的75%）。当WWDGT_WSPS位设置为0b11时，窗口启动值是0x3FFF（0x3FFF的100%）。

复位信号发生后，计数器会自动从选项字节中的WWDG_CNT[13:0]值开始工作。

如果启用了窗口看门狗定时器复位（选项字节中的WWDGT_WDGTEN位已置位），当计数器达到0（CNT[13:0] = 0）时，看门狗定时器将产生复位，或者当计数器大于窗口起始值（由选项字节中的WWDGT_WSPS设置）时刷新计数器，或者当计数器小于窗口结束值（由选项字节中的WWDGT_WEPS设置）时刷新计数器，看门狗定时器也将引起复位。

如果启用了早期唤醒中断（EWI）（在选项字节中的WWDGT_EWIE位被置位），当计数器达到0x01（CNT[13:0]=0x01）时，或者在计数器大于窗口起始值（由选项字节中的WWDGT_WIN_OB设置）时进行刷新，或者在计数器小于窗口结束值（由选项字节中的WWDGT_WEPS设置）时进行刷新，将生成中断。软件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。在这种情况下，WWDGT永远不会生成WWDGT复位，但可以用于其他事情。通过将WWDGT_STAT寄存器的EWIF位写0可以清除EWI中断。

复位或中断发生后，向下计数器会自动从选项字节中WWDG_CNT[13:0]位设置的值开始向下计数。

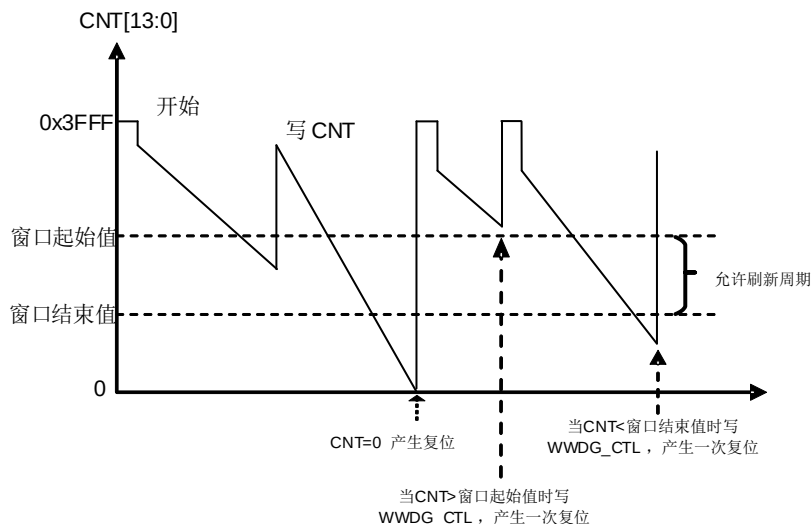
WWDGT_CFG 寄存器写操作

在复位信号和第一次刷新操作之间，只能对WWDGT_CFG寄存器进行一次写操作。在刷新操作或写入WWDGT_CFG寄存器之后，WWDGT_CFG寄存器将受到保护，此保护在WWDGT复位后解除。

定时与超时

窗口看门狗定时器时序图参考 [图20-3. 窗口看门狗定时器时序图](#)。

图 20-3. 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：

$$t_{\text{WWDGT}} = t_{\text{PCLK1}} \times 4096 \times 2^{\text{PSC}} \times (\text{CNT}[13:0]) \quad (\text{ms}) \quad (20-1)$$

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20-2. 在90MHz \(f_{PCLK1}\)时的最大/最小超时值](#)。

表 20-2. 在 90MHz (f_{PCLK1})时的最大/最小超时值

预分频系数	PSC[1:0]	最小超时 CNT[13:0] = 0x01	最大超时 CNT[13:0] = 0x3FFF
1/1	00	45.5 μs	0.746 s
1/2	01	91.0 μs	1.491 s
1/4	10	182.0 μs	2.982 s
1/8	11	364.1 μs	5.965 s

如果DBG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，窗口看门狗定时器会随着内核停止工作而停止计数。

20.2.4. WWDGT 寄存器

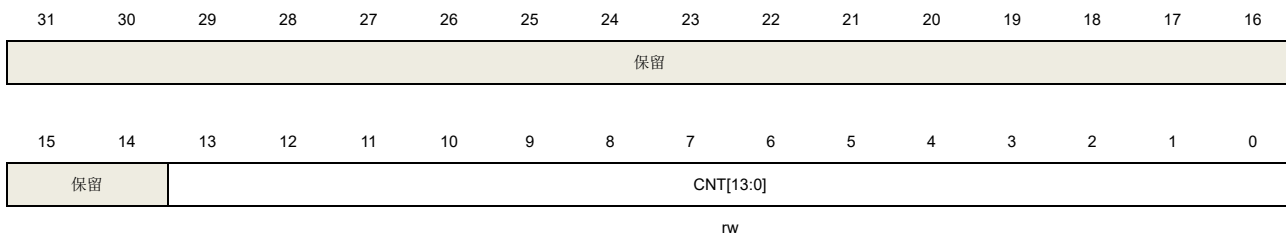
WWDGT 基地址: 0x4000 2C00

控制寄存器 (WWDGT_CTL)

地址偏移: 0x00

重置值: 0x0000 3FFF

这个寄存器可以通过半字 (16 位) 或字 (32 位) 访问。



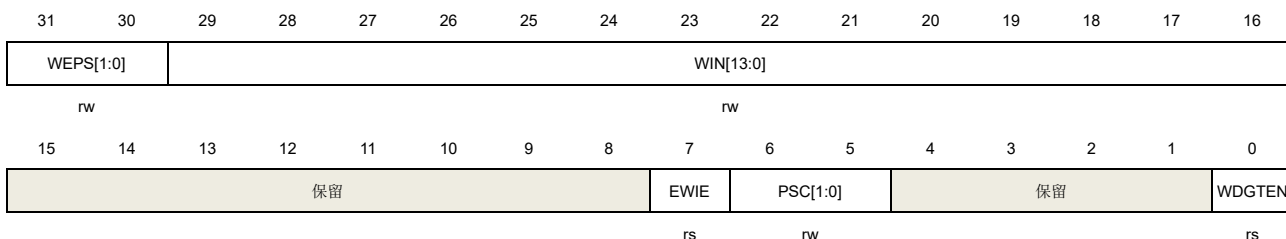
位/位域	名称	说明
31:14	保留	必须保持复位值。
13:0	CNT[13:0]	看门狗定时器计数器的值。当计数器的值从0x01减少到0x0时, 会产生复位。当计数器的值大于窗口起始值 (WIN) 或小于窗口结束值 (WEPS) 时, 写计数器也会产生复位。(在软件启动模式下, 通过刷新CNT值开始倒计时) 在硬件启动模式下, CNT的复位值由选项字节设置。 在软件启动模式下, CNT的复位值是0x3FFF。

配置寄存器 (WWDGT_CFG)

地址偏移: 0x04

复位值: 0xFFFF 0000

这个寄存器只能通过字 (32 位) 来访问。



位/位域	名称	说明
31:30	WEPS	窗口结束位置选择。如果WDGTEN=1, 当看门狗计数器的值小于窗口结束位置值时写入看门狗计数器 (WWDGT_CTL中的CNT位), 将发生复位。 00: 窗口结束位置值为0x2FFF (75% * 0x3FFF) 01: 窗口结束位置值为0x1FFF (50% * 0x3FFF) 10: 窗口结束位置值为0x0FFF (25% * 0x3FFF)

11: 窗口结束位置值为0x0 (0% * 0x3FFF)		
29:16	WIN[13:0]	窗口起始值。如果WDGTEN=1, 当看门狗定时器计数器的值大于窗口起始值时, 写入看门狗定时器计数器 (WWDGT_CTL中的CNT位) 会产生复位。
15:8	保留	必须保持复位值。
7	EWIE	提前唤醒中断使能。如果该位被置1, 则在计数器达到0x01或在允许刷新周期之外写入看门狗定时器计数器 (WWDGT_CTL中的CNT位) 时, 将发生中断。该位由硬件复位清0或通过设置RCU模块的WWDGTRST位进行软件复位来清0。写0操作无效。EWIE位独立于WDGTEN位。当EWIE设置为1时, WDGTEN必须设置为0。
6:5	PSC[1:0]	预分频器, 看门狗定时器的时间基准。 00: (PCLK1 / 4096) / 1 01: (PCLK1 / 4096) / 2 10: (PCLK1 / 4096) / 4 11: (PCLK1 / 4096) / 8
4:1	保留	必须保持复位值。
0	WDGTEN	当WDGTEN=1时, 窗口看门狗定时器计时器可以在计数器达到0x00或在允许刷新周期之外写入看门狗定时器计数器 (WWDGT_CTL中的CNT位) 时产生复位。这个位由软件设置, 并通过硬件复位清0。写0操作无效。当WDGTEN设置为1时, EWIE必须设置为0。 0: 窗口看门狗计时器不生成复位 1: 窗口看门狗计时器可以生成复位

状态寄存器 (WWDGT_STAT)

地址偏移: 0x08

重置值: 0x0000 0000

这个寄存器可以通过半字 (16 位) 或字 (32 位) 来访问。

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													UFEF	REFEF	EWIF
													rc_w0	rc_w0	rc_w0

位/位域	名称	说明
31:3	保留	必须保持复位值。
2	UFEF	下溢错误标志。当计数器到达0x01, 并且WWDGT配置为生成中断 (选项字节WWDGT_0中的WWDGT_EWIE或WWDGT_CFG中的EWIE已设置), UFEF将由硬件置位。通过向WWDG写0或对WWDG进行复位来清除该位。写1操作无效。
1	REFEF	刷新错误标志。当计数器 (WWDGT_CTL中的CNT位) 在允许刷新周期之外被写入,

并且 WWDGT 配置为生成中断（选项字节 WWDGT_0 中的 WWDGT_EWIE 或 WWDGT_CFG 中的 EWIE 已设置），REFEF 将由硬件置位。通过向 WWDGT 写 0 或对 WWDGT 进行复位来清除该位。写 1 操作无效。

0 EWIF

提前唤醒中断标志位。当计数值达到 0x01 或计数器（WWDGT_CTL 中的 CNT 位）在允许刷新周期之外被写入时，即使中断未被使能（选项字节 WWDGT_0 中的 WWDGT_EWIE 或 WWDGT_CFG 中的 EWIE 被清除），该位也会被硬件置位。通过向 WWDGT 写入 0 或对 WWDGT 进行复位来清除该位。写 1 操作无效。

21. 定时器（TIMER）

表 21-1. 定时器（TIMERx）分为 2 种类型

定时器	定时器 0/7	定时器 1/2
类型	高级	通用 L0
预分频器	16 位	16 位
计数器	16 位	16 位
计数模式	向上，向下，中央对齐	向上，向下，中央对齐
可重复性	●	×
捕获/比较 通道数	8	4
非对称 PWM 模式	●	×
复合 PWM 模式	●	×
输出匹配脉冲选择	●	●
互补和 死区时间	●	×
Break	●	×
单脉冲	●	●
可延时的单脉冲	●	●
正交译码器	●	●
译码器	●	●
主-从管理	●	●
内部连接	● ⁽¹⁾	● ⁽²⁾
专用 ADC 触发器	●	×
DMA	●	●
Debug 模式	●	●

	TIMERx	ITI0	ITI1	ITI2	ITI3
(1)	TIMER0	TIMER7_TRGO	TIMER1_TRGO	TIMER2_TRGO	EVIC
	TIMER7	TIMER0_TRGO	TIMER1_TRGO	TIMER2_TRGO	EVIC
(2)	TIMER1	TIMER0_TRGO	TIMER7_TRGO	TIMER2_TRGO	EVIC
	TIMER2	TIMER0_TRGO	TIMER7_TRGO	TIMER1_TRGO	EVIC

21.1. 高级定时器 (TIMERx, x = 0, 7)

21.1.1. 简介

高级定时器 (TIMER0/7) 是八通道定时器，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。

21.1.2. 主要特征

- 总通道数：8；
- 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；
- 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。

21.1.3. 结构框图

[图 21-1. 高级定时器结构框图](#)提供了高级定时器的内部配置细节，[表 21-2. 高级定时器通道介绍](#)介绍了通道输入和输出情况。

图 21-1. 高级定时器结构框图

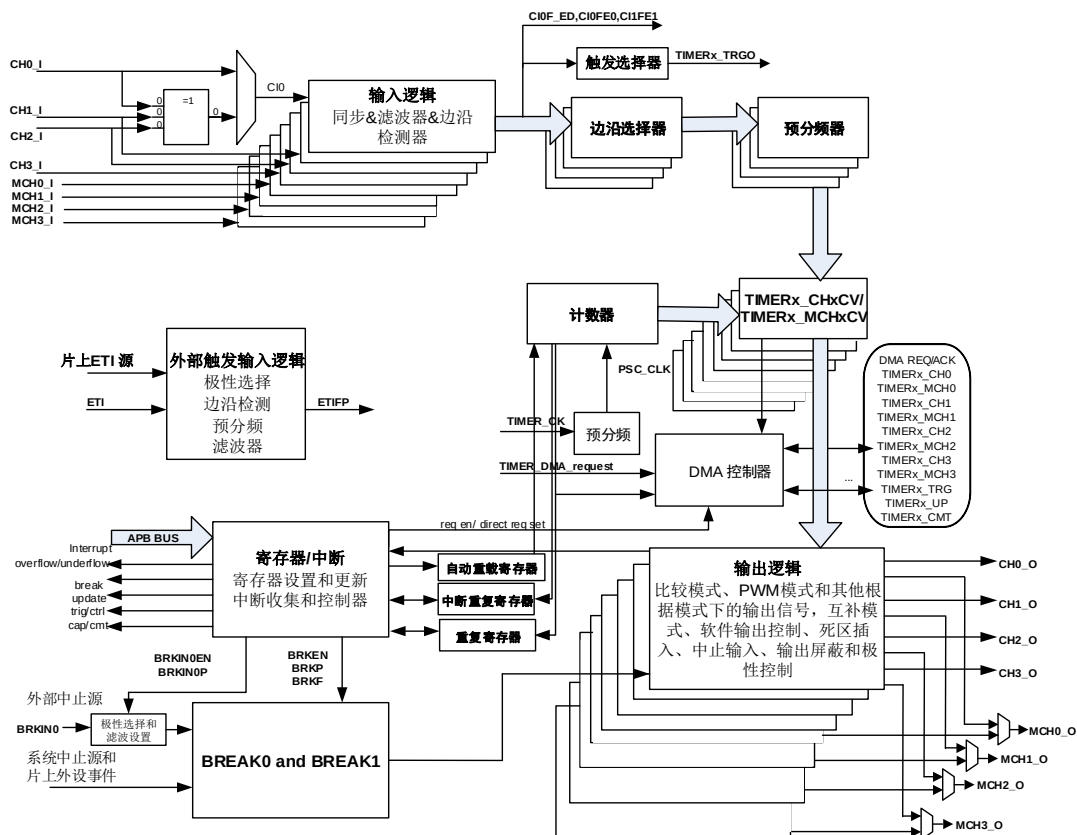


表 21-2. 高级定时器通道介绍

通道名称 ($x = 0 \dots 3$)	MCHxMSEL[1:0]=00 独立模式	MCHxMSEL[1:0]=11 互补模式
CHx (通道 x)	CHx 和 MCHx 可独立输入捕获、独立比较输出	只有 CHx 可用于输入，CHx 和 MCHx 输出互补
MCHx (多模式通道 x)		

21.1.4. 功能描述

时钟源选择

高级定时器可以由内部时钟源 CK_TIMER 或者由 SYSCFG_TIMERxCFG($x = 0, 7$)寄存器中的 TSCFGy[4:0] ($y=0..7, 9..12, 15$)位域控制的复用时钟源驱动。

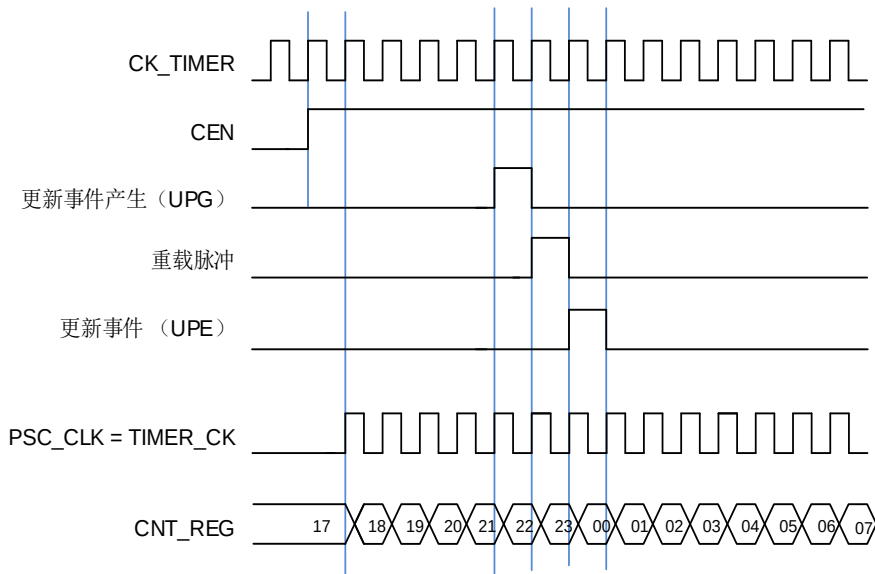
- 当 SYSCFG_TIMERxCFG($x = 0, 7$)寄存器中的 TSCFGy[4:0] = 5'b00000 ($y=0..7, 9..12, 15$) 时，定时器选择内部时钟源（连接到 RCU 模块的 CK_TIMER）

如果 SYSCFG_TIMERxCFG($x = 0, 7$)寄存器中的 TSCFGy[4:0] = 5'b00000 ($y=0..7, 9..12, 15$)，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。

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]` ($y=0..2, 6, 9..12$) 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]` ($y = 3, 4, 5, 7$) 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 21-2. 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/ TIMERx_MCHn` ($n=0..3$) 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 `0x9~0xE` 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

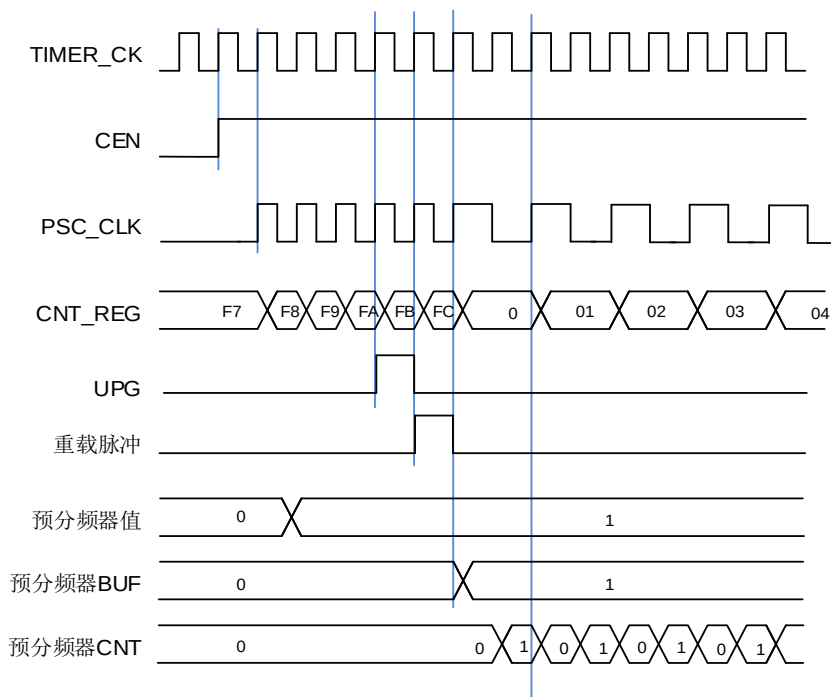
计数器预分频器可以在外部引脚 `ETI` 的每个上升沿或下降沿计数。这种模式可以通过设置 `TIMERx_SMCFG` 寄存器中的 `SMC1` 位为 1 来选择。另一种选择 `ETI` 信号作为时钟源方式是，设置 `TSCFG6[4:0]` 为 `0x8`。注意 `ETI` 信号是通过数字滤波器采样 `ETI` 引脚得到的。如果选择 `ETI` 信号为时钟源，触发控制器包括边沿监测电路将在每个 `ETI` 信号上升沿产生一个时钟脉冲来为计数器预分频器提供时钟。

注意：`ETI` 信号可以从外部 `ETI` 引脚输入，也可由片上外设提供，可以参考 `TIMERx_CTL0` 寄存器的 `ETISEL[1:0]` 位域。

时钟预分频器

预分频器可以将定时器的时钟 (`TIMER_CK`) 频率按 1 到 65536 之间的任意值分频，分频后的时钟 `PSC_CLK` 驱动计数器计数。分频系数受预分频寄存器 `TIMERx_PSC` 控制，这个控制寄存器带有缓冲器，它能够在运行时被改变。新的预分频器的参数在下次更新事件到来时被采用。

图 21-3. 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 $(\text{TIMERx\_CREP} + 1)$ 次上溢后产生更新事件，否则在每次上溢时都会产生更新事件。在向上计数模式中，`TIMERx_CTL0` 寄存器中的计数方向控制位 `DIR` 应该被设置成 0。

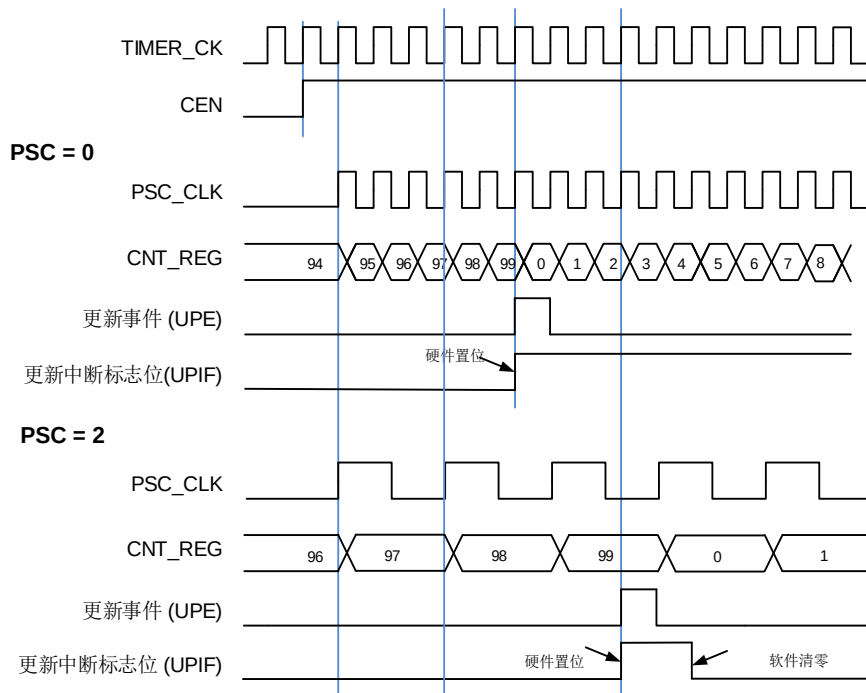
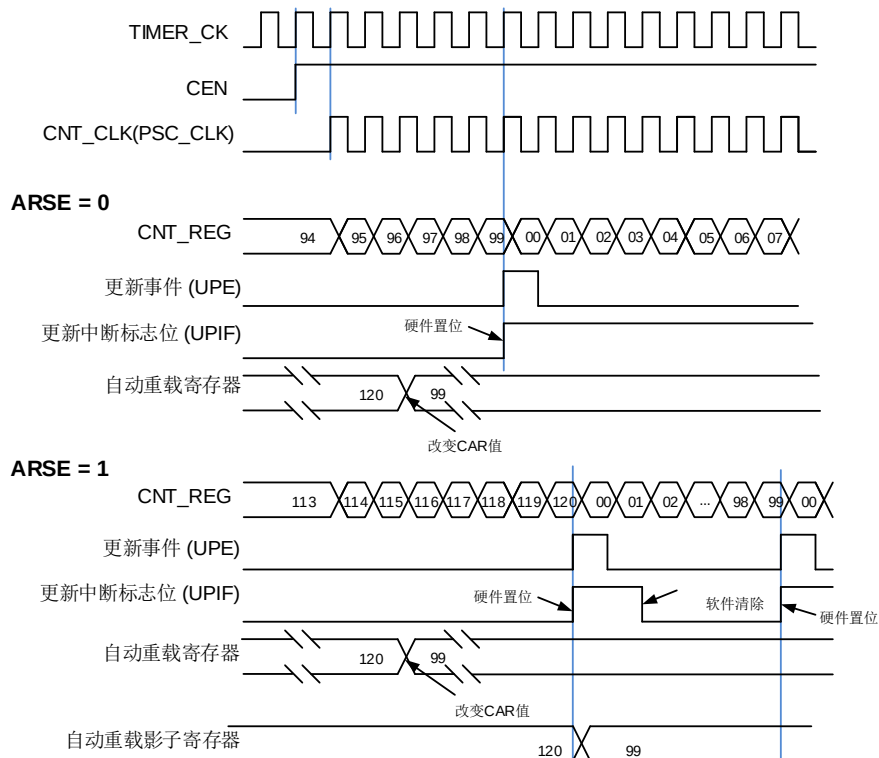
当通过 `TIMERx_SWEVG` 寄存器的 `UPG` 位置 1 来设置更新事件时，计数值会被清 0，并产生更新事件。

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

当发生更新事件时，所有的寄存器（重复计数器，预分频寄存器）都将被更新。自动重载寄存器由影子寄存器控制，详见[影子寄存器加载功能](#)。

[图 21-4. 向上计数时序图，PSC = 0/2](#) 和 [图 21-5. 向上计数时序图，在运行时改变 `TIMERx\_CAR` 寄存器的值 \(`ARSE=0`, `CAR` 值立即更新, `ARSE=1`, `CARUPSEL = 2'b00`, `CAR` 值会在下一个更新事件来临时更新\)](#) 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 21-4. 向上计数时序图, PSC = 0/2

图 21-5. 向上计数时序图, 在运行时改变 **TIMERx_CAR** 寄存器的值 (**ARSE=0**, **CAR** 值立即更新, **ARSE=1**, **CARUPSEL=2'b00**, **CAR** 值会在下一个更新事件来临时更新)

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 $(\text{TIMERx\_CREP} + 1)$ 次下溢后产生更新事件，否则在每次下溢时都会产生更新事件。在向下计数模式中，`TIMERx_CTL0` 寄存器中的计数方向控制位 `DIR` 应该被设置成 1。

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

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

当发生更新事件时，所有的寄存器（重复计数器，预分频寄存器）都将被更新。自动重载寄存器由影子寄存器控制，详见[影子寄存器加载功能](#)。

[图 21-6. 向下计数时序图，PSC = 0/2](#) 和 [图 21-7. 向下计数时序图，在运行时改变 `TIMERx\_CAR` 寄存器值 \(`ARSE=0`, `CAR` 值立即更新, `ARSE=1`, `CARUPSEL = 2'b00`, `CAR` 值会在下一个更新事件来临时更新\)](#) 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 21-6. 向下计数时序图，PSC = 0/2

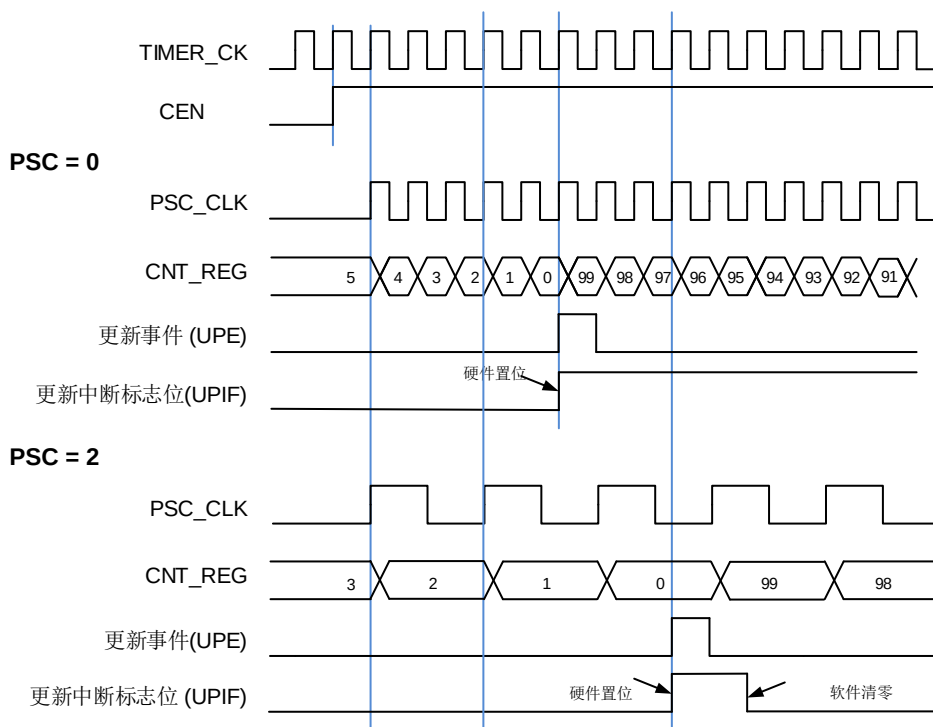
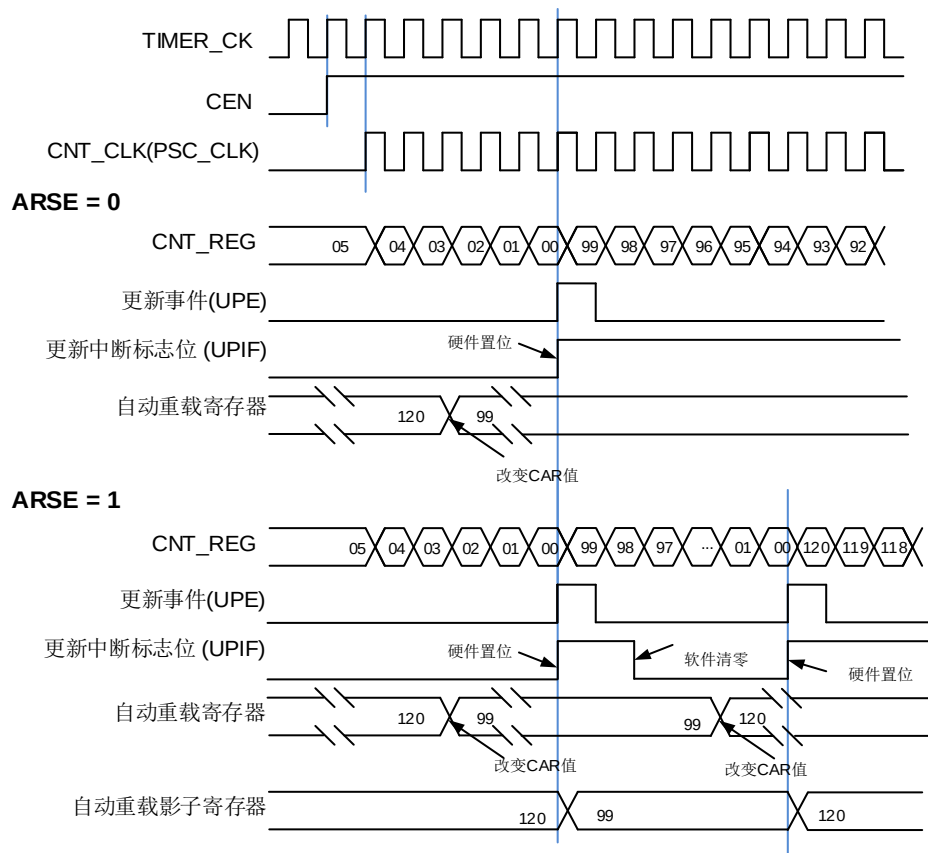


图 21-7. 向下计数时序图，在运行时改变 `TIMERx_CAR` 寄存器值 (`ARSE=0`, `CAR` 值立即更新)

新, $ARSE=1$, $CARUPSEL=2'b00$, CAR 值会在下一个更新事件来临时更新)



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。在中央对齐模式下, 可以产生专用的下溢或上溢中断, $TIMERx\_INTF$ 寄存器中的 $OVERFLOWIF$ 或 $UNDERFLOWIF$ 位将设置为 1。具体细节参考 [图21-8. 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。

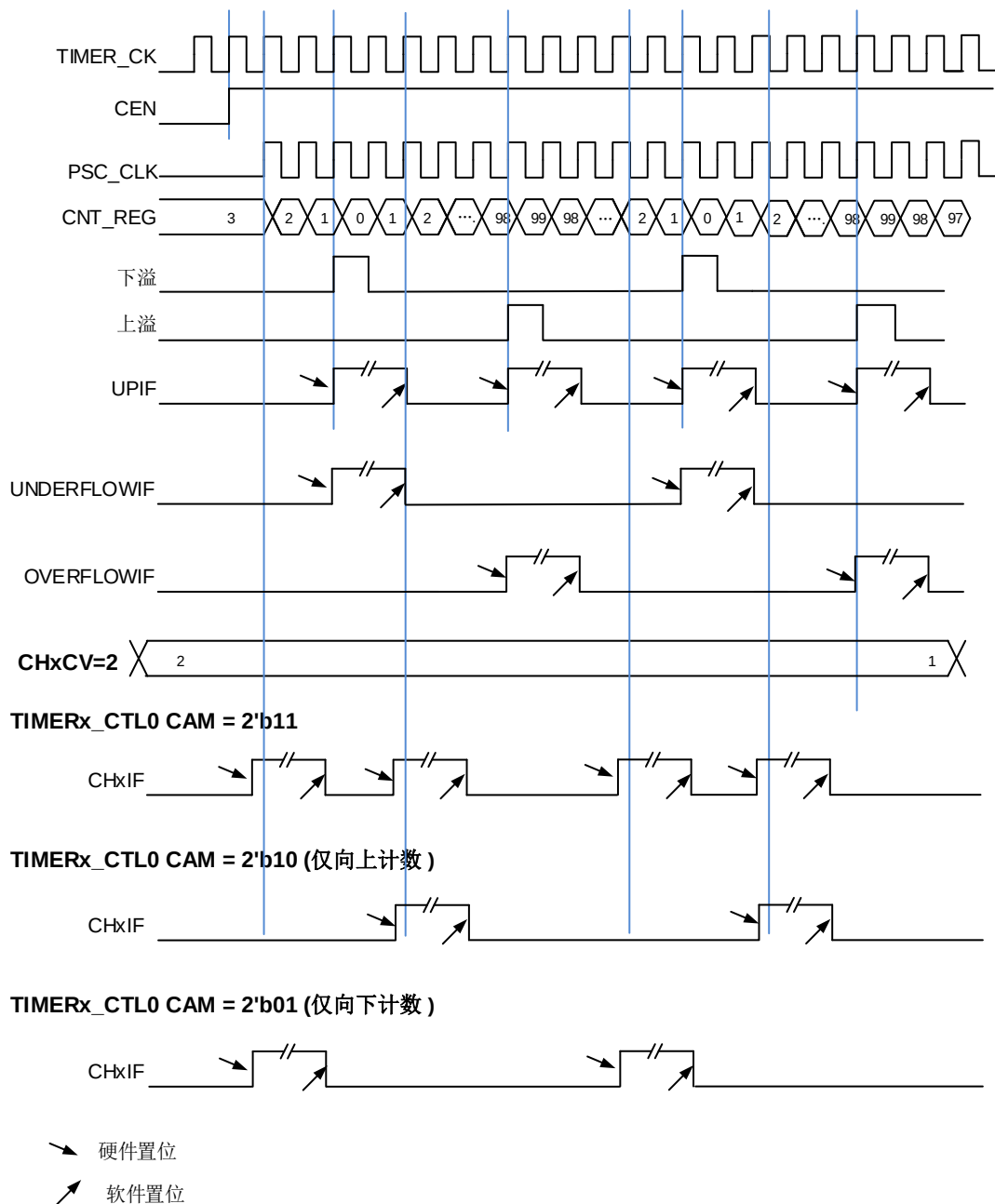
当发生更新事件时, 所有的寄存器 (重复计数器, 预分频寄存器) 都将被更新。自动重载寄存器由影子寄存器控制, 详见 [影子寄存器加载功能](#)。

在输出比较模式或 ADC 触发模式中, 用于比较匹配的比较寄存器和附加比较寄存器 ($CHxCV /$

MCHxCV / CHxCOMVAL_ADD ($x=0..3$)), 以及用于生成 ADC 触发的两个比较寄存器 (TIMERx_ADCCR1 / TIMERx_ADCCR2), 也由影子寄存器控制。在中央对齐模式下, 这些带有影子寄存器的值将根据影子寄存器的配置选择计数器更新事件、上溢、下溢、计数器上溢和下溢、更新事件或立即更新预加载值。详见[影子寄存器加载功能](#)。

[图21-8. 中央计数模式计数器时序图](#)给出了一些例子, 当 $\text{TIMERx\_CAR} = 0x99$, $\text{TIMERx\_PSC} = 0x0$, $\text{TIMERx\_CREP} = 0x0$ 时, 计数器的行为。

图 21-8. 中央计数模式计数器时序图



重复计数器

高级定时器的重复寄存器 TIMERx_CREP , 可用于生成更新事件和 TIMERx_IREP , 用于生成

上溢或下溢中断。

重复计数器用于产生更新事件

重复计数器（TIMERx_CREP）是用来在（N + 1）个计数周期之后产生更新事件，更新定时器的寄存器，N 为 TIMERx_CREP 寄存器的 CREP 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、下溢时递减或者在上溢和下溢时都递减，根据 UPSSEL[1:0]寄存器进行选择。

将 TIMERx_SWEVG 寄存器的 UPG 位置 1 可以重载 TIMERx_CREP 寄存器中 CREP 的值并产生一个更新事件。

在中央对齐模式下，如果同时选择计数器上溢和下溢来减少重复计数器，对于奇数的 CREP，更新事件发生在上溢或下溢取决于 CREP 寄存器的写入时间和计数器的启动时间。当在启动计数器之前写入 CREP 时，更新事件在上溢时生成；当在启动计数器之后写入 CREP 时，在下溢出时生成。

图 21-9. 中央计数模式下计数器重复时序图

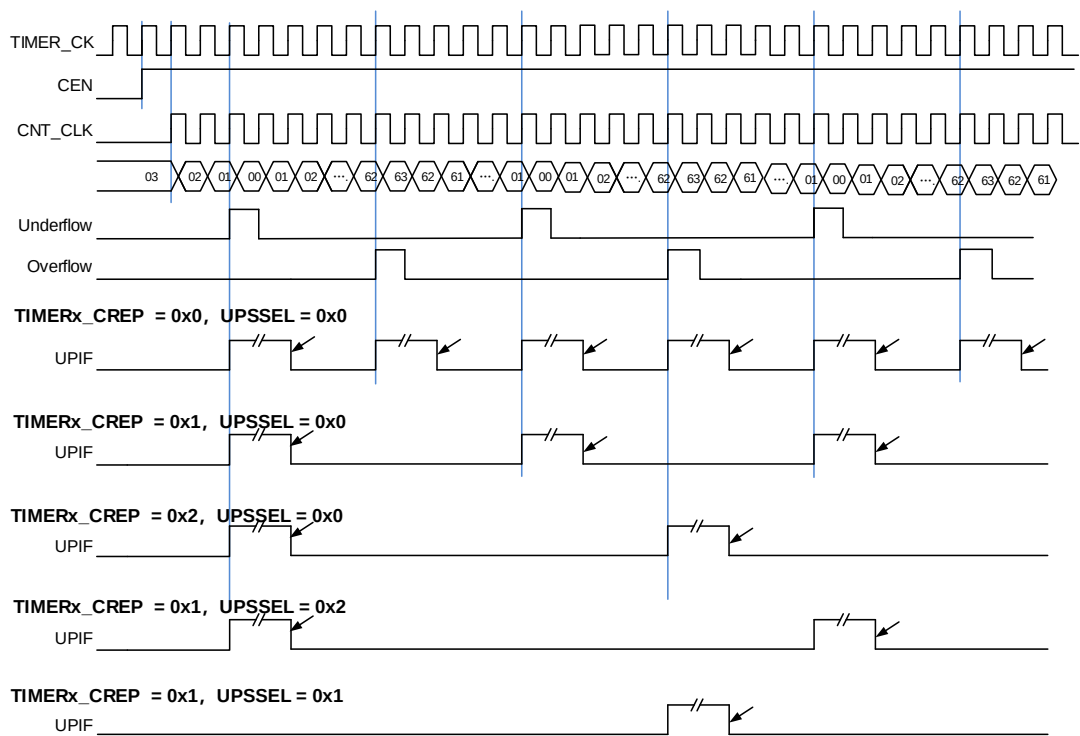


图 21-10. 在向上计数模式下计数器重复时序图

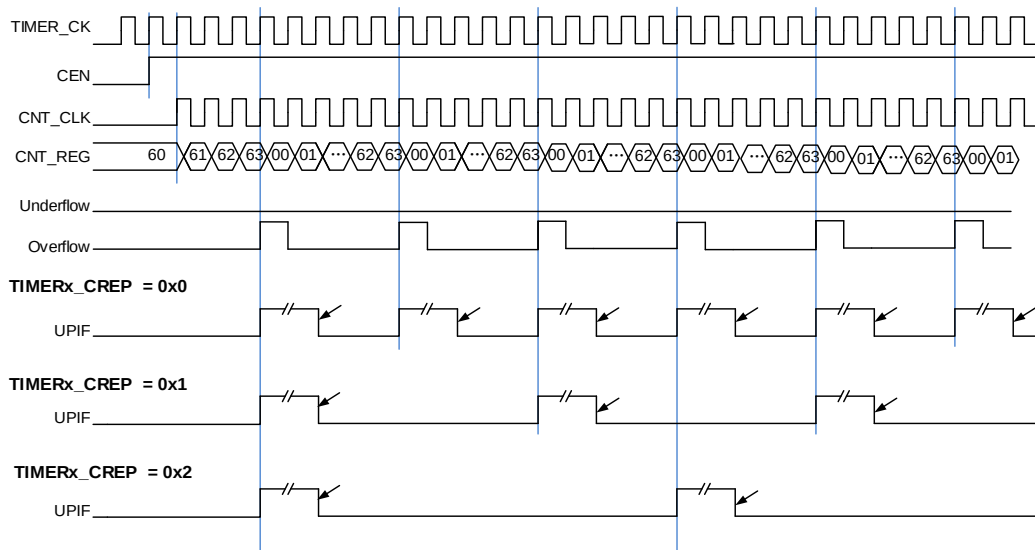
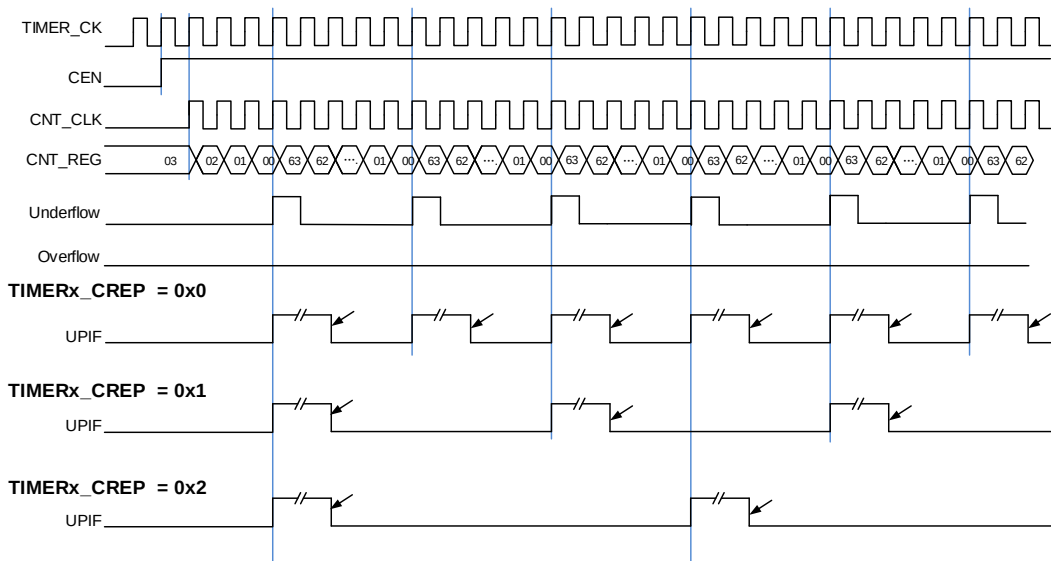


图 21-11. 在向下计数模式下计数器重复时序图



重复计数器用于产生上溢或下溢中断

中断重复计数器(TIMEx_IREF)用于仅在计数器的给定周期数(N+1)之后才产生下溢或上溢中断, 其中 N 是 TIMEx_IREF 寄存器中 IREF 位的值。在向上计数模式下, 每次计数器溢出时中断重复计数器减 1; 在向下计数模式下, 每次计数器下溢时中断重复计数器减 1。

在中央对齐模式下, 中断重复计数器在每次计数器上溢或下溢时递减, 或在每次计数器上溢和下溢时递减, 根据 INTCOSEL[1:0]寄存器进行选择。

在中央对齐模式下, 将根据上溢标志(OVERFLOWIF)或下溢标志(UNDERFLOWIF)分别确定上溢或下溢中断。

图 21-12. 中央计数模式下中断重复时序图

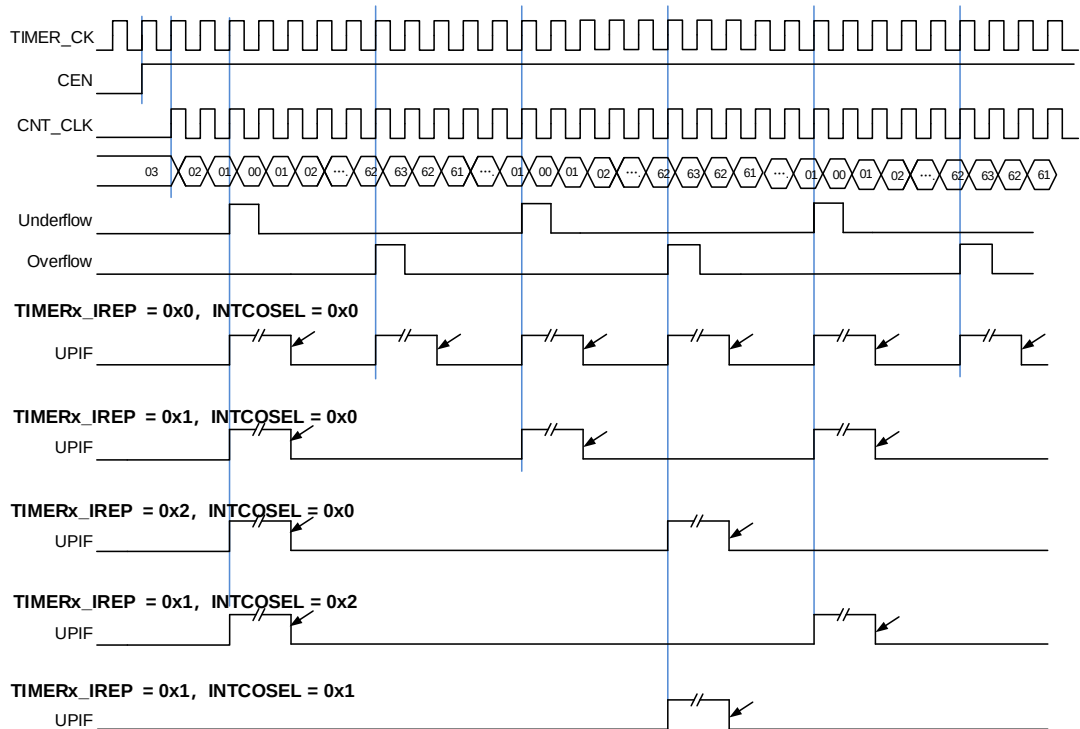


图 21-13. 在向上计数模式下中断重复时序图

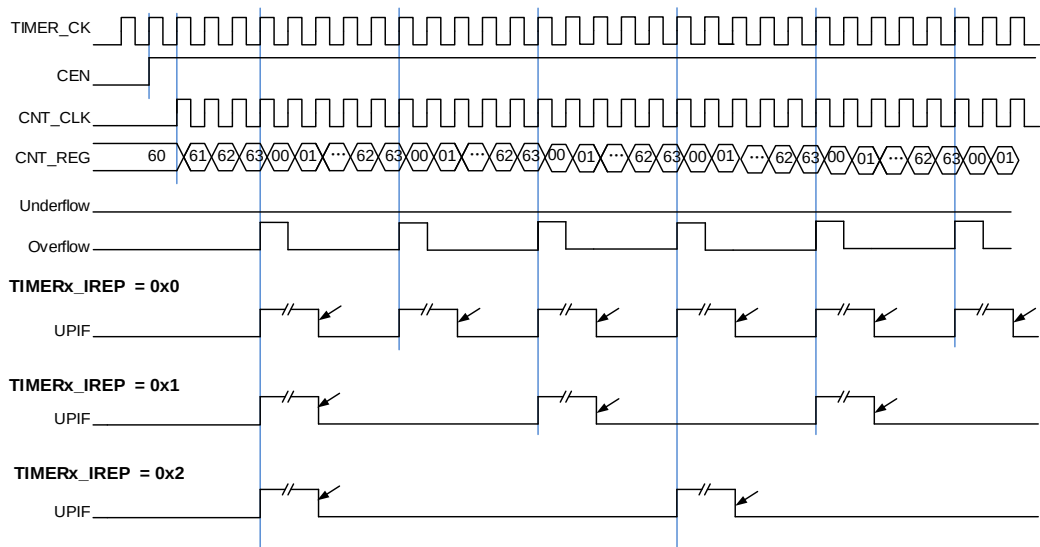
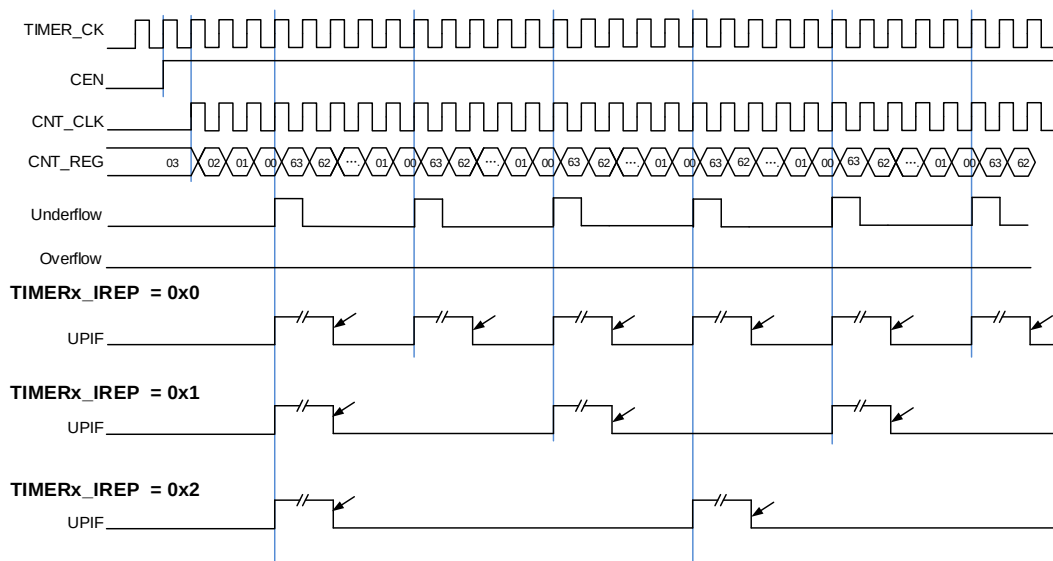


图 21-14. 在向下计数模式下中断重复时序图

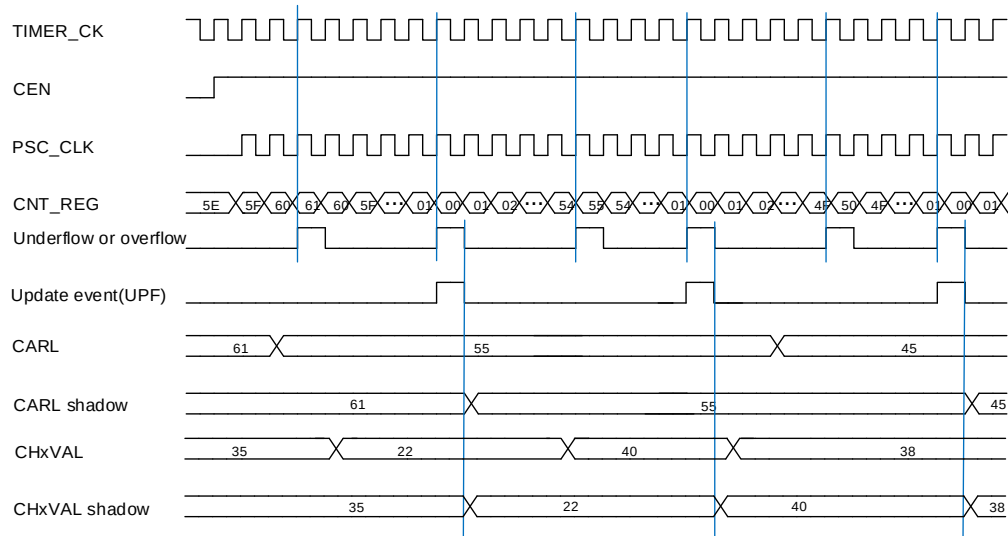


影子寄存器加载功能

TIMERx_CAR, TIMERx_CHxCV, TIMERx_CHxCOMV_ADD, TIMERx_ADCCR1 和 TIMERx_ADCCR2 ($x = 0,7$) 都有单独的影子寄存器使能位。当影子寄存器功能关闭时, 对应的寄存器将立即以新配置的值更新, 定时器将以新配置的值运行。具有影子寄存器功能的寄存器在 TIMERx_SHUPM0 和 TIMERx_SHUPM1 寄存器中分别有单独的 2 位更新模式选择。当影子寄存器功能启用时, 可以选择每个影子寄存器来启用全局更新功能, 或下一次计数器上溢更新, 下溢更新, 或者上、下溢更新。全局更新功能可以实现选定的影子寄存器在更新事件统一更新。

[图 21-15. 更新事件时全局更新模式下的更新时序图](#)显示了 CARL 和 CHxCV ($x = 0..3$) 在全局更新模式下的更新时序图。(ARSE = 1, CH0COMSEN = 1)。

图 21-15. 更新事件时全局更新模式下的更新时序图



捕获 / 比较通道

高级定时器拥有 8 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， $TIMERx\_CHxCV / TIMERx\_MCHxCV$ ($x = 0...3$) 寄存器会捕获计数器当前的值，同时 $CHxIF / MCHxIF$ ($x = 0...3$) 位置 1，如果 $CHxIE / MCHxIE = 1$ ($x = 0...3$)，则产生相应的通道中断。

图 21-16. 通道 0 输入捕获逻辑

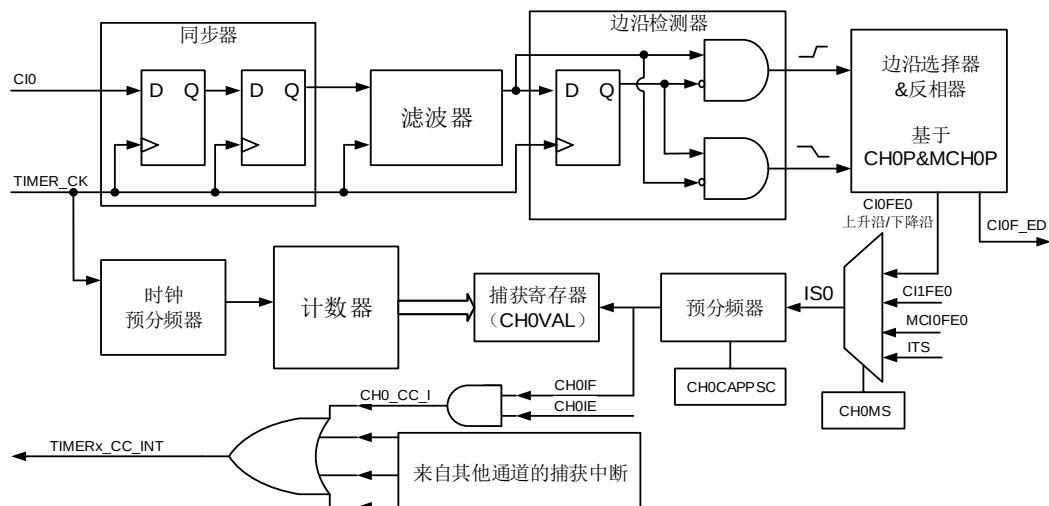
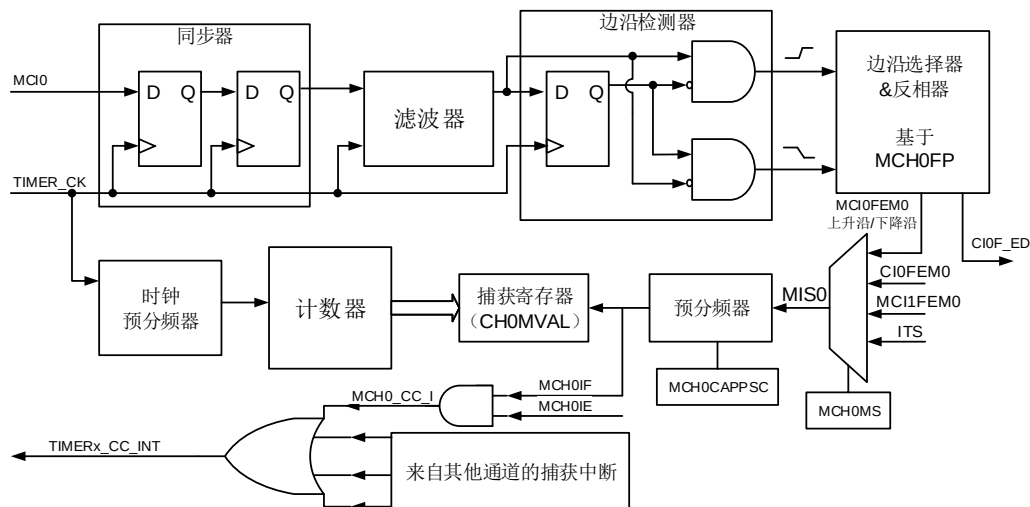


图 21-17. 多模式通道 0 输入捕获逻辑



通道输入信号 $Clx / MClx$ 有两种选择，一种是 $TIMERx\_CHx / TIMERx\_MCHxCV$ 信号，另一种是 $TIMERx\_CH0$, $TIMERx\_CH1$ 和 $TIMERx\_CH2$ 异或之后的信号（仅限于 $CI0$ ）。

通道输入信号 $Clx / MClx$ 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。

配置步骤如下：

第一步：滤波器配置（ $TIMERx\_CHCTL0$ 寄存器中 $CHxCAPFLT$ 位和 $TIMERx\_MCHCTL0$ 寄存器中 $MCHxCAPFLT$ ）：

根据输入信号和请求信号的质量，配置相应的 $CHxCAPFLT / MCHxCAPFLT$ 位。

第二步：边沿选择（ $TIMERx\_CHCTL2$ 寄存器中 $CHxP$ 和 $MCHxP$ 位， $TIMERx\_MCHCTL2$ 寄存器中 $MCHxFP[1:0]$ 位域）：

配置 $CHxP$ 和 $MCHxP$ 位或 $MCHxFP$ 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 21-18. 输出比较逻辑（当 \$MCHxMSEL = 2'b00\$ 时, \$x = 0,1,2,3\$ ）](#)和 [图 21-19. 输出比较逻辑（当 \$MCHxMSEL = 2'b11\$ 时, \$x = 0,1,2,3\$ ）](#) 给出了通道的输出比较逻辑。

图 21-18. 输出比较逻辑（当 $MCHxMSEL = 2'b00$ 时, $x = 0,1,2,3$ ）

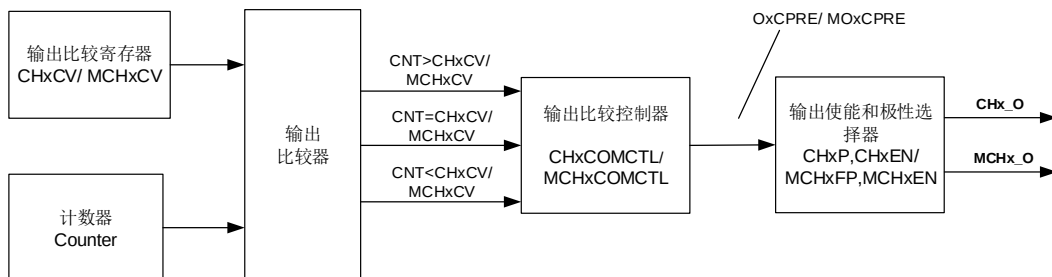
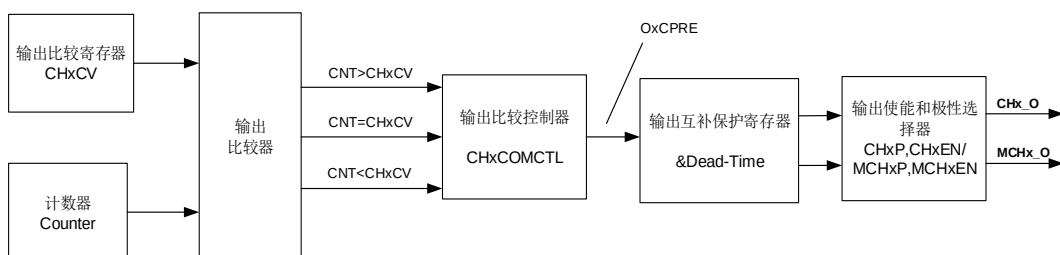


图 21-19. 输出比较逻辑（当 $MCHxMSEL = 2'b11$ 时, $x = 0,1,2,3$ ）



通道输出信号 $CHx\_O/MCHx\_O$ 与 $OxCPRE / MOxCPRE$ 信号（详情请见[清除通道输出准备信号](#)）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21-18. 输出比较逻辑（当 \$MCHxMSEL = 2'b00\$ 时, \$x = 0,1,2,3\$ ）](#)。
- 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21-19. 输出比较逻辑（当 \$MCHxMSEL = 2'b11\$ 时, \$x = 0,1,2,3\$ ）](#)。

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 $TIMERx\_CCHP$ 寄存器中的相关位（ ROS 、 IOS 、 POE 和 $DTCFG$ 等位）有关。详情请见[互补输出](#)。

在输出比较模式， $TIMERx$ 可以产生时控脉冲，其位置，极性，持续时间和频率都是可编程的。当一个输出通道的 $TIMERx\_CHxCV$ / $TIMERx\_MCHxCV$ 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。

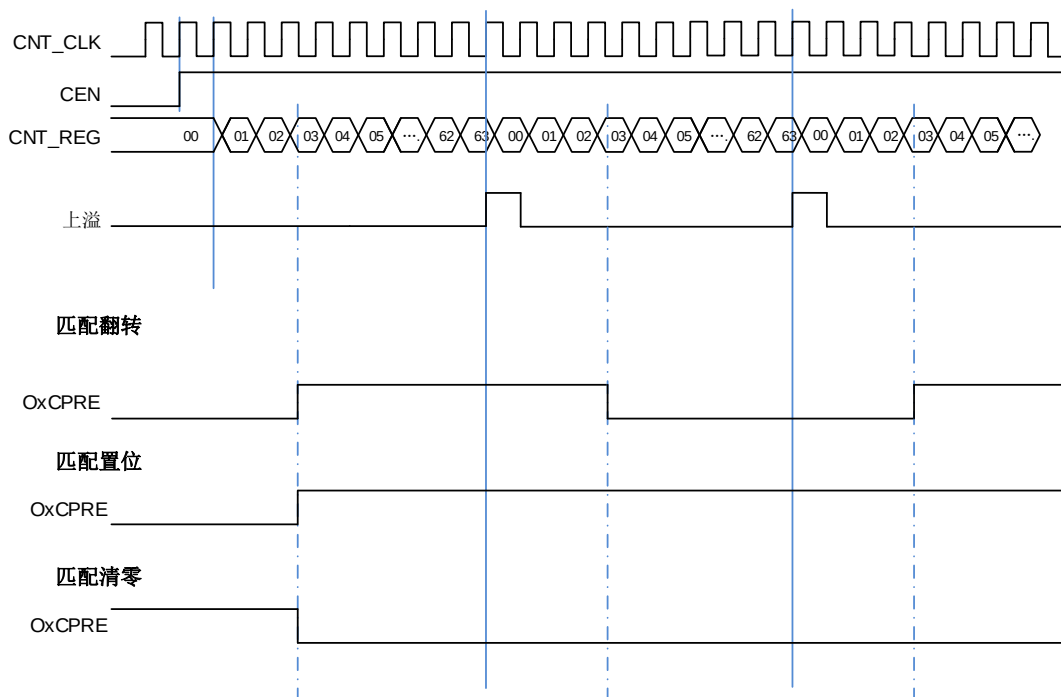
第四步：通过 $TIMERx\_CAR$ 寄存器和 $TIMERx\_CHxCV$ 寄存器配置输出比较时基：

$TIMERx\_CHxCV$ / $TIMERx\_MCHxCV$ 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 21-20. 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 21-20. 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 21-21. 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 21-22. 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，通道输出一直为无效电平。

图 21-21. EAPWM 时序图

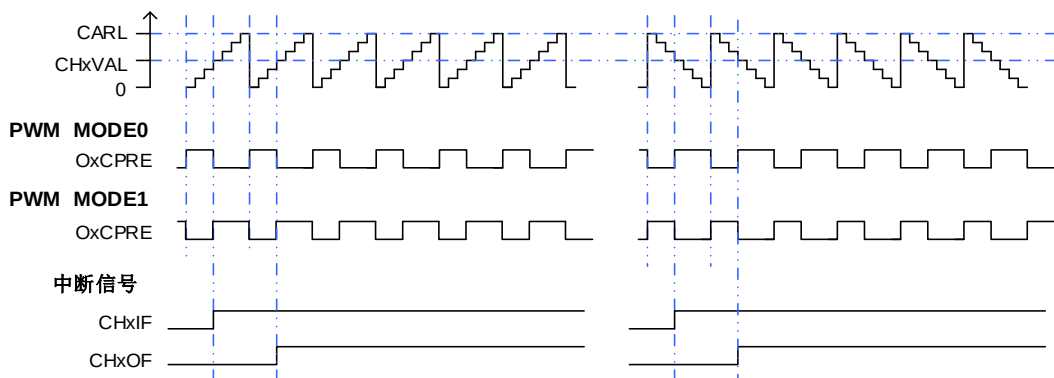
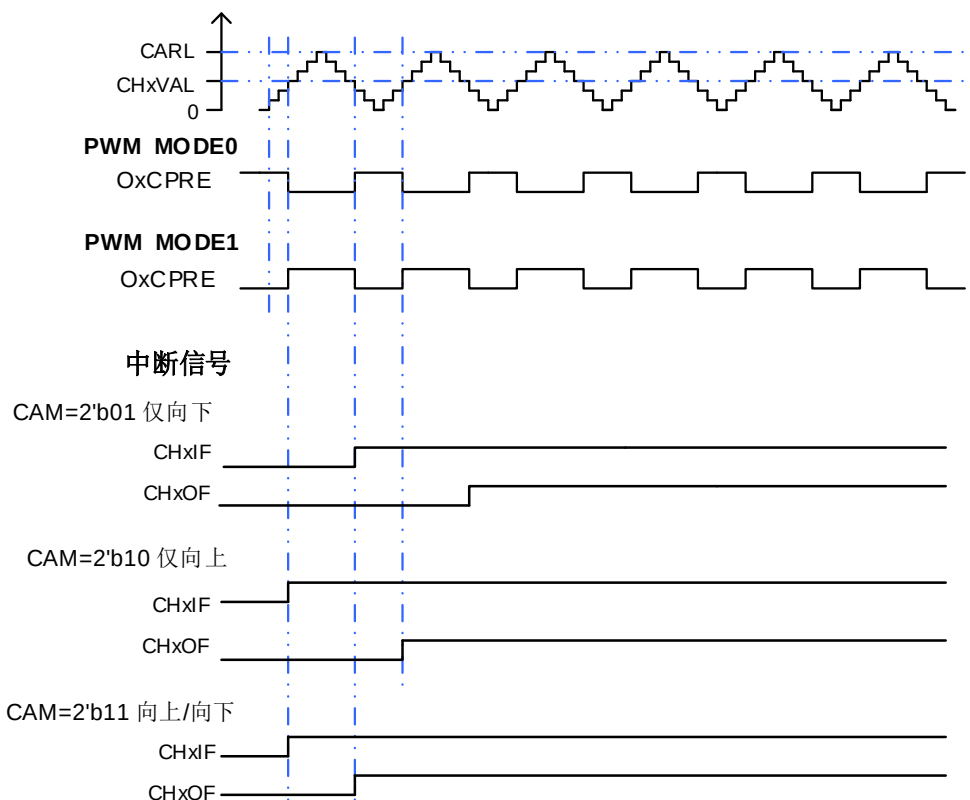


图 21-22. CAPWM 时序图



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 $CHxCOMCTL = 4'b0110$ (复合 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）。

中央对齐模式下，复合 PWM 波形为边沿对齐计数模式下的对称波形，以边沿对齐模式为例进行分析。PWM 的周期取决于（CARL + 0x0001），PWM 脉冲宽度可以下[表 21-3 复合 PWM 脉冲宽度](#)计算。

表 21-3 复合 PWM 脉冲宽度

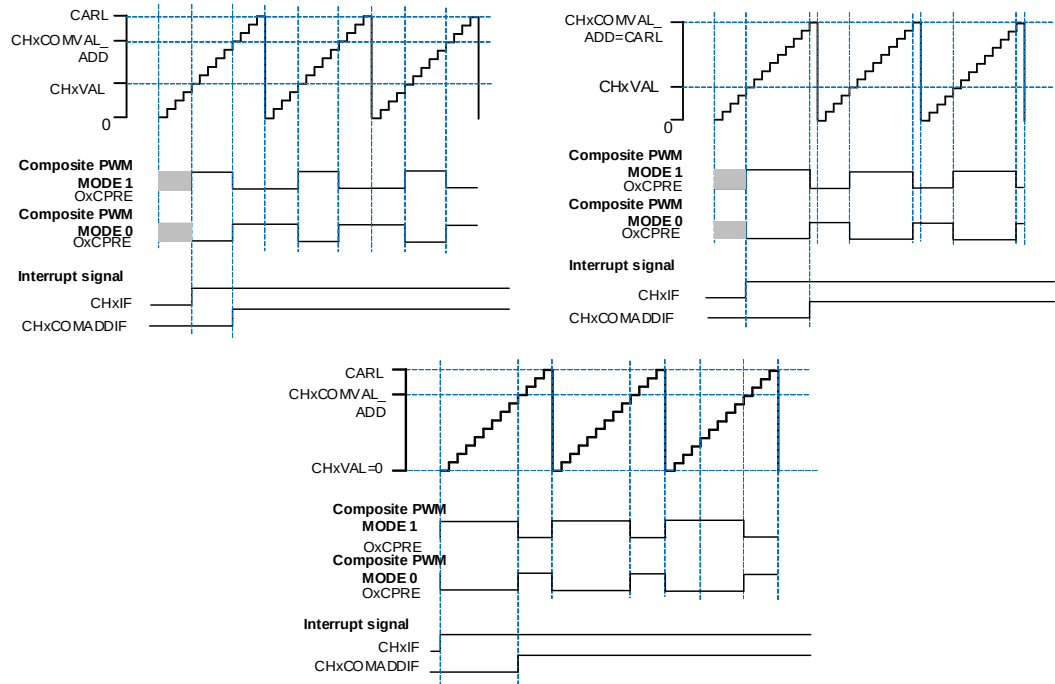
条件	模式	PWM 脉冲宽度
CHxVAL < CHxCOMVAL_ADD ≤ CARL	复合 PWM 模式 0	(CARL + 0x0001) + (CHxVAL - CHxCOMVAL_ADD)
	复合 PWM 模式 1	(CHxCOMVAL_ADD - CHxVAL)
CHxCOMVAL_ADD < CHxVAL ≤ CARL	复合 PWM 模式 0	(CHxVAL - CHxCOMVAL_ADD)
	复合 PWM 模式 1	(CARL + 0x0001) + (CHxCOMVAL_ADD - CHxVAL)
(CHxVAL = CHxCOMVAL_ADD ≤ CARL) 或 (CHxVAL > CARL > CHxCOMVAL_ADD)	复合 PWM 模式 0（向上计数）或 复合 PWM 模式 1（向下计数）	100%
	复合 PWM 模式 0（向下计数）或 复合 PWM 模式 1（向上计数）	0%
CHxCOMVAL_ADD > CARL > CHxVAL	复合 PWM 模式 0（向上计数）或 复合 PWM 模式 1（向下计数）	0%
	复合 PWM 模式 0（向下计数）或 复合 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：

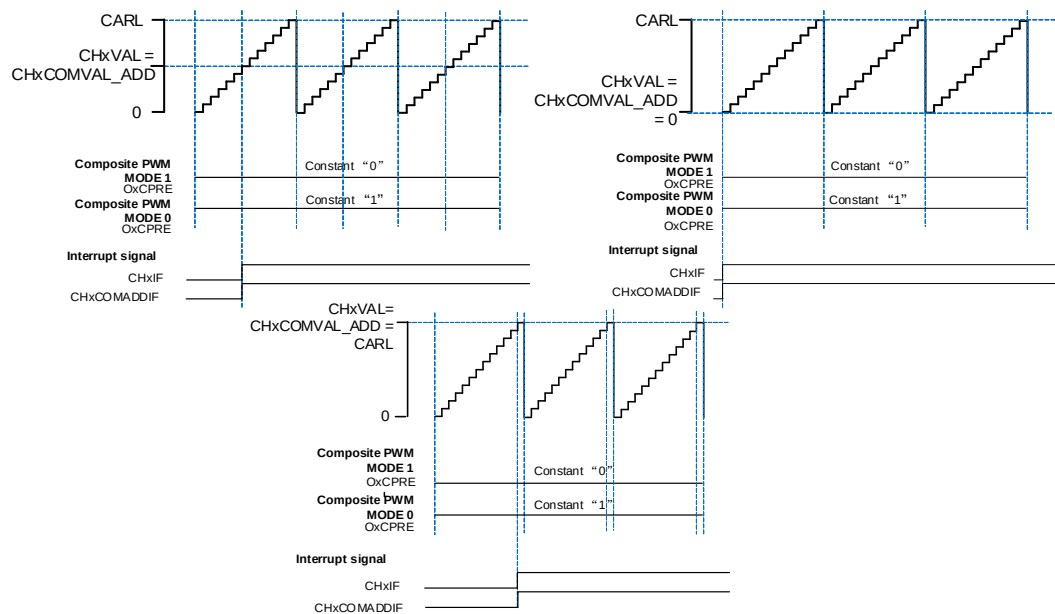
- 1) CHxVAL < CHxCOMVAL_ADD，CHxVAL和CHxCOMVAL_ADD值介于0和CARL之间。

图 21-23 通道 x 输出 PWM (CHxVAL < CHxCOMVAL_ADD)



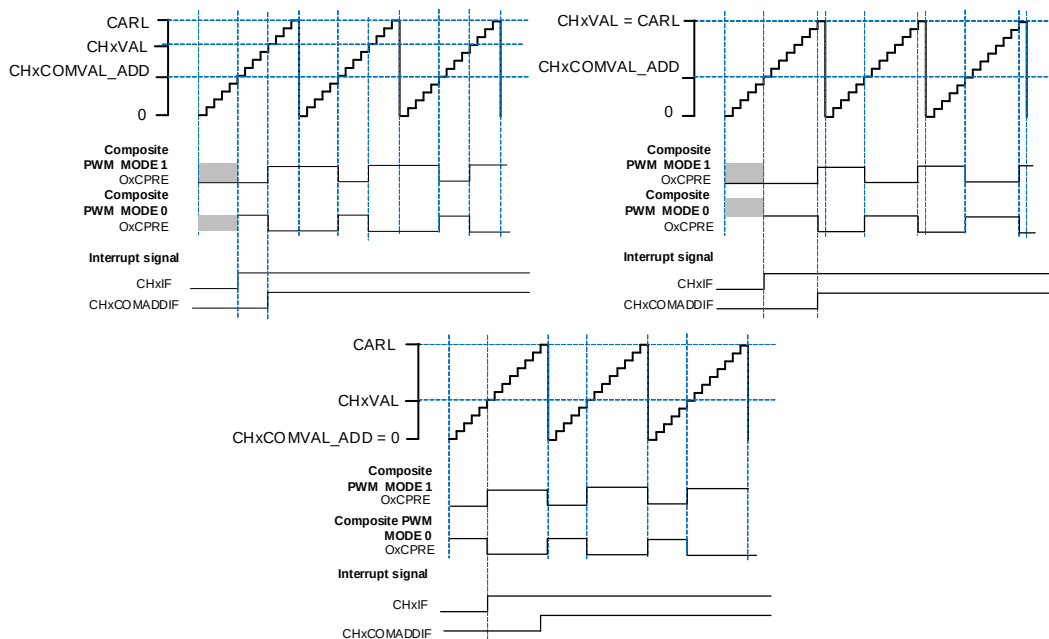
2) $CHxVAL = CHxCOMVAL\_ADD$, $CHxVAL$ 和 $CHxCOMVAL\_ADD$ 值介于0和 $CARL$ 之间。

图 21-24 通道 x 输出 PWM ($CHxVAL = CHxCOMVAL\_ADD$)

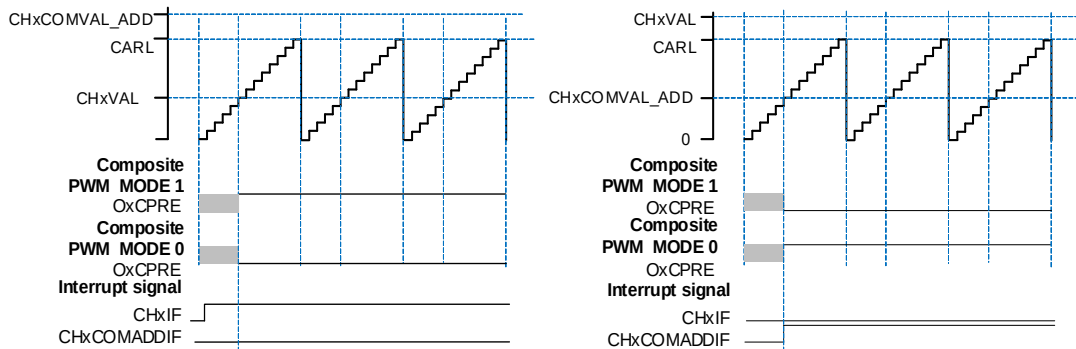


3) $CHxVAL > CHxCOMVAL\_ADD$, $CHxVAL$ 和 $CHxCOMVAL\_ADD$ 值介于0和 $CARL$ 之间。

图 21-25. 通道 x 输出 PWM (CHxVAL > CHxCOMVAL_ADD)



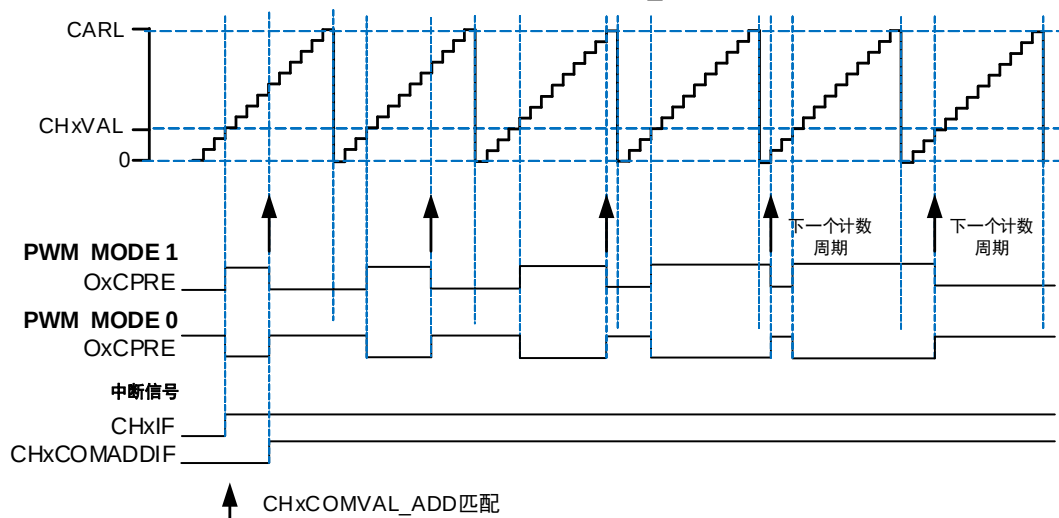
4) $CHxVAL$ 或 $CHxCOMVAL\_ADD$ 值大于 $CARL$ 。

图 21-26. 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 21-27. 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。

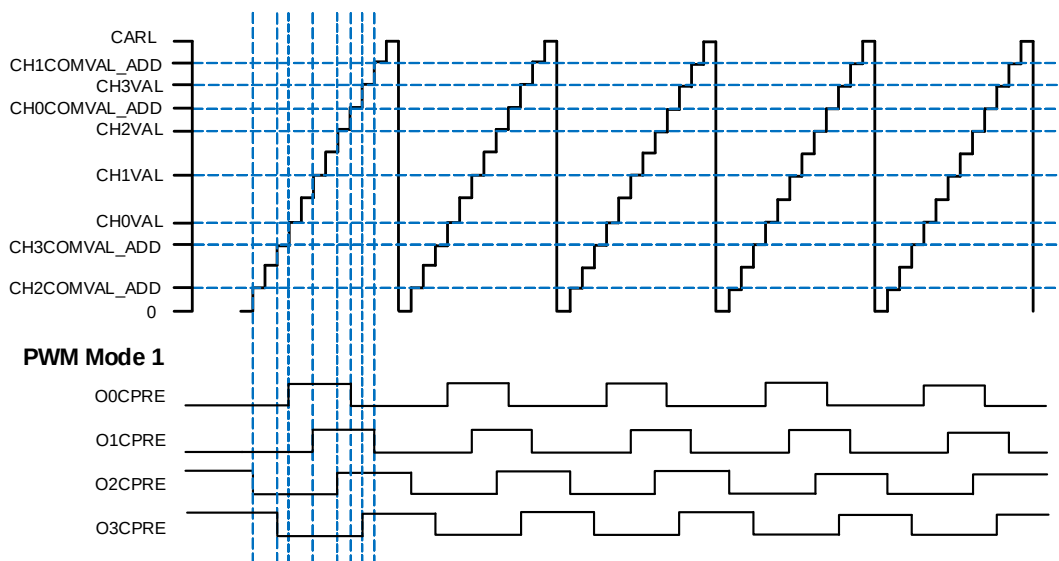
在某些情况下， $CHxCOMVAL\_ADD$ 的匹配事件可以发生在下一个计数周期 ($CHxCOMVAL\_ADD$ 值在计数器到达 $CHxVAL$ 值之后被写入，且 $CHxCOMVAL\_ADD$ 值小于或者等于 $CHxVAL$ 值)。

图 21-27. 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 21-28. 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，输出强制为低电平。

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 $TIMERx\_CHCTL2$ 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 $2CARL$ ，PWM 脉冲宽度可以下 [表 21-4 非对称 PWM 脉冲宽度](#) 计算。

表 21-4 非对称 PWM 脉冲宽度

条件	模式	PWM 脉冲宽度
$CHxVAL < CHxCOMVAL\_ADD \leq CARL$ 或者 $CHxCOMVAL\_ADD < CHxVAL \leq CARL$ 或者 $CHxVAL = CHxCOMVAL\_ADD$	非对称 PWM 模式 0	$CHxCOMVAL\_ADD + CHxVAL$
	非对称 PWM 模式 1	$2CARL - CHxCOMVAL\_ADD - CHxVAL$
$CHxVAL > CARL > CHxCOMVAL\_ADD$	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

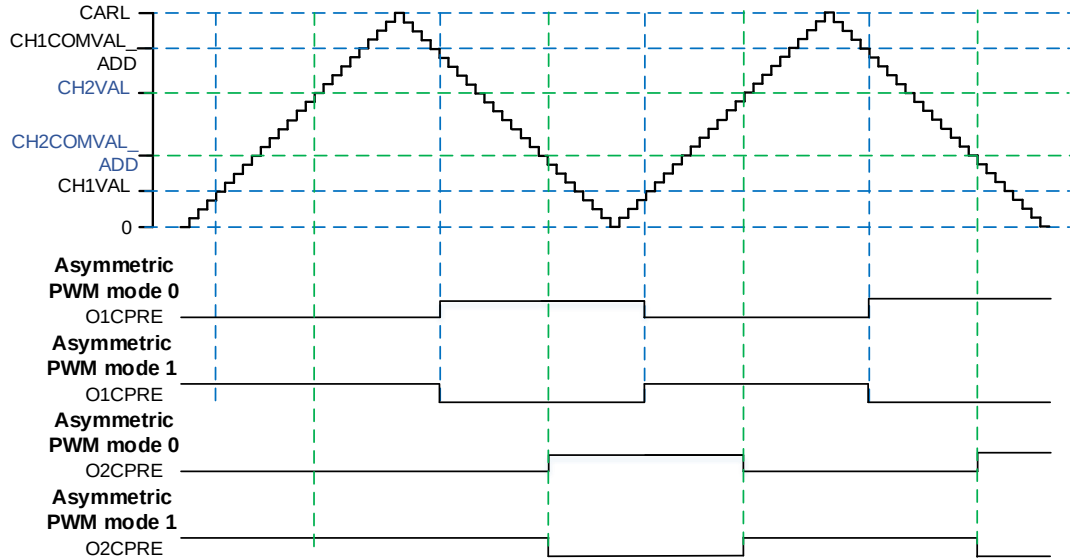
当计数器计数到 $CHxVAL$ ， $CHxIF$ 位置 1 且如果 $CHxIE=1$ 通道 x 产生中断，如果 $CHxDEN=1$ ，则产生 DMA 请求。当计数器计数到 $CHxCOMVAL\_ADD$ 时， $CHxCOMADDIF$ 位置 1（该中断标志位只在复合 PWM 模式和非对称 PWM 模式有效，当 $CHxCOMCTL[3:2] = 2'b10$ ），如果 $CHxCOMADDIE = 1$ 通道 x 附加比较中断产生（只有中断产生，没有 DMA 请求响应）。

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

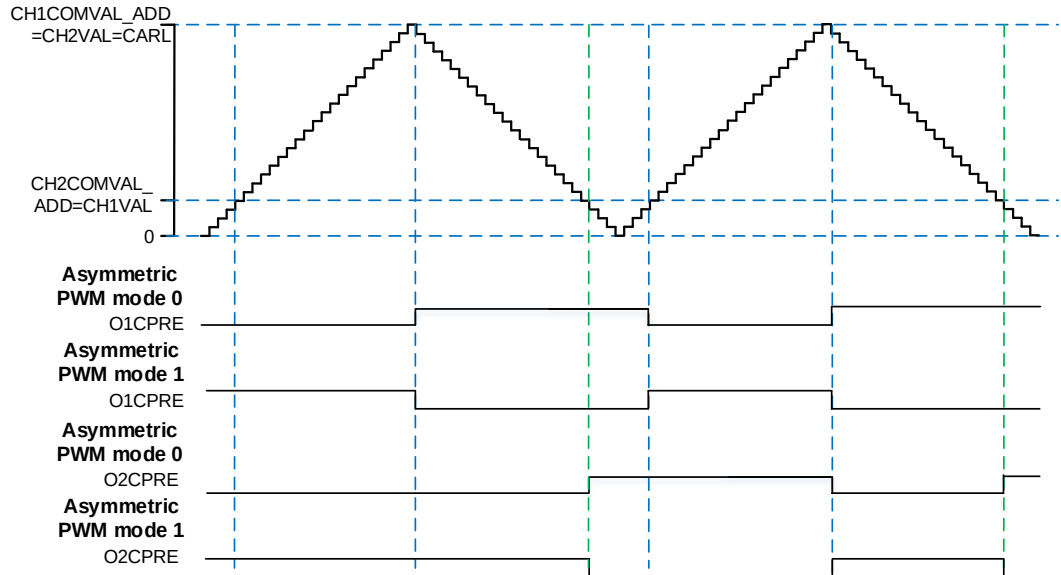
1) $CHxVAL < CHxCOMVAL\_ADD$, $CHxVAL$ 和 $CHxCOMVAL\_ADD$ 值介于0和 $CARL$ 之间。

图 21.29 通道 x 输出 PWM($CHxVAL < CHxCOMVAL\_ADD / CHxCOMVAL\_ADD < CHxVAL$)

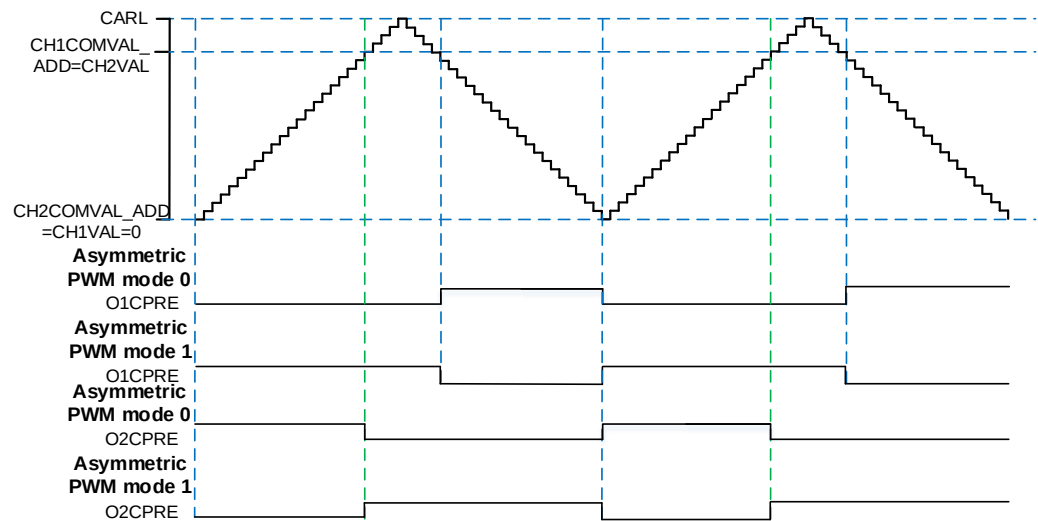
a. $0 < CH1VAL < CH1COMVAL\_ADD < CARL / 0 < CH2COMVAL\_ADD < CH2VAL < CARL$



b. $0 < CH1VAL < CH1COMVAL\_ADD = CARL / 0 < CH2COMVAL\_ADD < CH2VAL = CARL$, 同时 $CH1VAL = CH2COMVAL\_ADD$



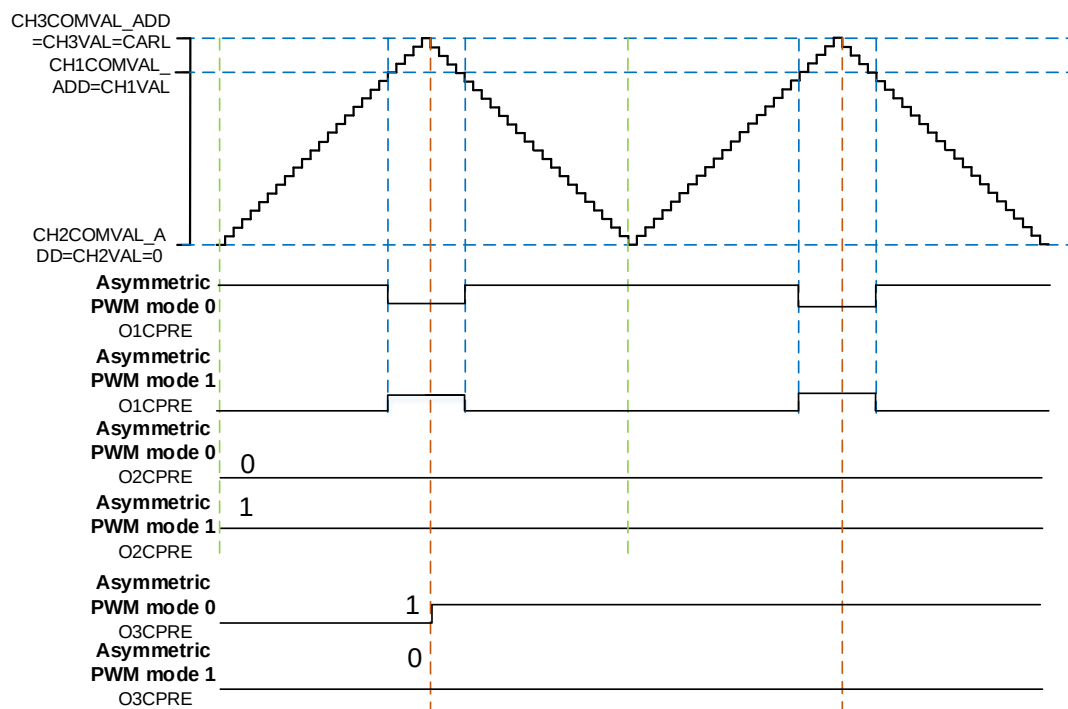
c. $0 = CH1VAL < CH1COMVAL\_ADD < CARL / 0 = CH2COMVAL\_ADD < CH2VAL < CARL$, 同时 $CH1COMVAL\_ADD = CH2VAL$



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

图 21.30. 通道 x 输出 PWM ($CHxVAL = CHxCOMVAL\_ADD$)

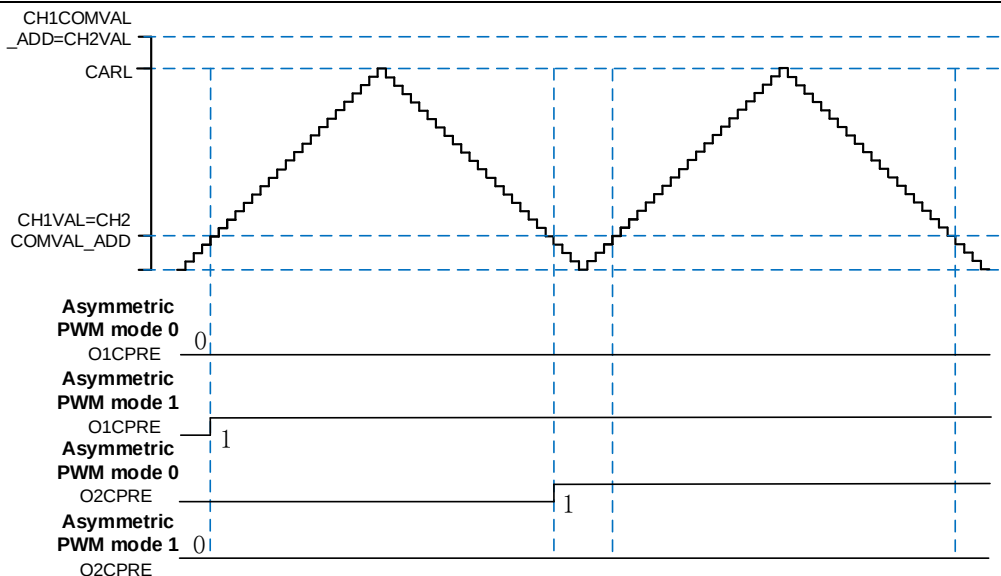
$0 < CH1VAL = CH1COMVAL\_ADD < CARL / CH2VAL = CH2COMVAL\_ADD = 0 / CH3VAL = CH3COMVAL\_ADD = CARL$



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

图 21.31. 通道 x 输出 PWM ($CHxVAL$ 或 $CHxCOMVAL\_ADD > CARL$)

$0 < CH1VAL < CARL < CH1COMVAL\_ADD / 0 < CH2COMVAL\_ADD < CARL < CH2VAL$



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[2:0]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 21.32. 边沿对齐模式下 CHx_O 输出脉冲 (CHxOMPSEL≠2'b00)

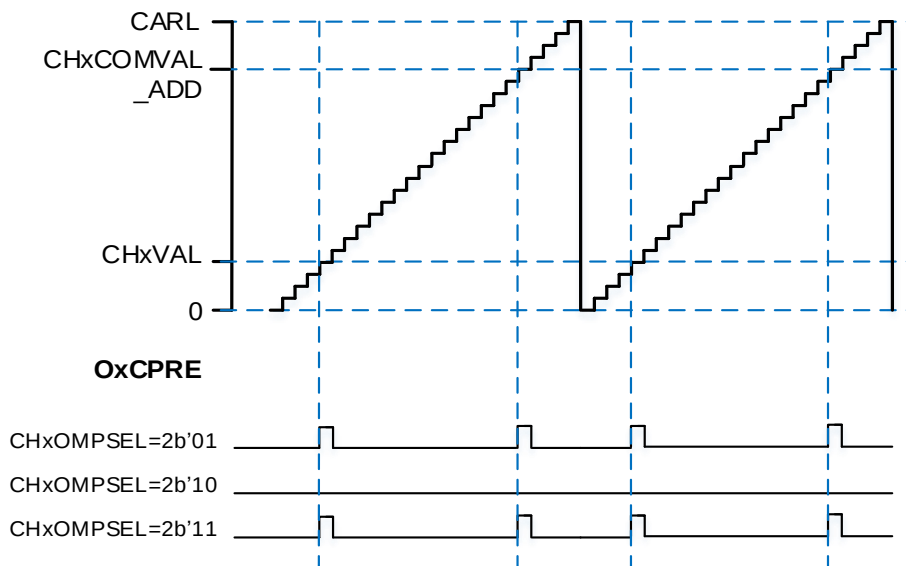
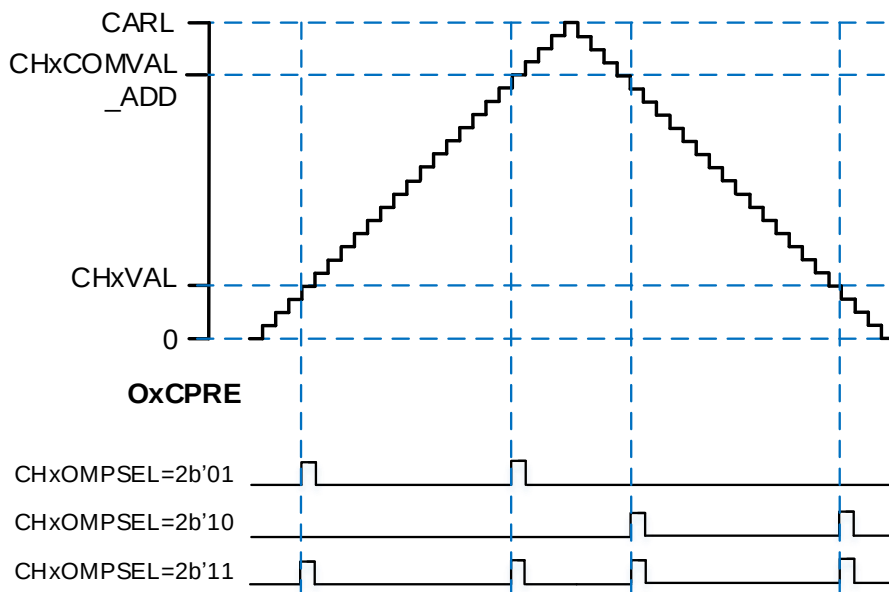


图 21.33. 中央对齐模式下 CHx_O 输出脉冲 (CHxOMPSEL#2'b00)



通道输出准备信号

如[图 21-18. 输出比较逻辑 \(当 MCHxMSEL = 2'b00 时, x = 0,1,2,3\)](#)和[图 21-19. 输出比较逻辑 \(当 MCHxMSEL = 2'b11 时, x = 0,1,2,3\)](#)所示, 当 TIMERx 用于输出匹配比较模式下, 在通道输出信号之前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 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。

复合 PWM 模式 0/复合 PWM 模式 1 输出是 OxCPRE 的一种输出类型, 通过将 CHxCOMCTL 位域配置为 0x08/0x09 来设置。在这些模式中, OxCPRE 信号电平会根据计数方向以及计数值与 TIMERx_CHxCV/TIMERx_CHxCOMVAL_ADD 之间的关系而变化。有关更多详细信息, 请参阅相关位的定义。

非对称 PWM 模式 0/非对称 PWM 模式 1 输出是 OxCPRE 的一种输出类型, 通过将 CHxCOMCTL 位域配置为 0x0a/0x0b 来设置。在这些模式中, OxCPRE 信号电平会根据计数方向以及计数值与 TIMERx_CHxCV/TIMERx_CHxCOMVAL_ADD 之间的关系而变化。有关更多详细信息, 请参阅相关位的定义。

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。

清除通道输出准备信号

当 $CHxCOMCEN$ 位或 $MCHxCOMCEN$ 位（在 $TIMERx\_CHCTLy$ / $TIMERx\_MCHCTLy$ 寄存器中）置 1 时， $OxCPRE$ 和 $MOxCPRE$ 信号可以由 $OCPRE\_CLR\_INT$ 信号清除。该功能用于 $CHxCOMCTL[3:0]$ 位域（4'b0100 和 4'b0101 除外）或 $MCHxCOMCTL[2:0]$ 位域（3'b100 和 3'b101 除外）中配置的比较输出模式。

可以通过 $TIMERx\_SMCFG$ 寄存器中的 $OCRC$ 位来选择 $OCPRE\_CLR\_INT$ 的信号源。

$OCRC$ 位清 0 时， $OCPRE\_CLR\_INT$ 连接到 $OCPRE\_CLR$ 输入。 $OxCPRE$ / $MOxCPRE$ 信号被 $OCPRE\_CLR\_INT$ 信号的高电平清除，直到下一次有效电平发生时才会恢复输出。在 $TIMERx\_AFCTL$ 寄存器的 $OCRINSEL[2:0]$ 位域中选择 $OCPRE\_CLR$ 的输入。

$OCRC$ 位置 1 时， $OCPRE$ / $MOCPRE$ 连接到 $ETIF$ 。由 $TIMERx\_SMCFG$ 寄存器中的 ETP 位配置 $OCPRE\_CLR\_INT$ 的输入极性。此时， $SMC1$ 位置 1， $ETPSC[1:0]$ 位域必须设置为 2'b00。

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。 $TIMERx$ 有 4 对通道，所有 4 对通道都具有此功能。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， $TIMERx\_CCHP$ 寄存器中和 $TIMERx\_CTL1$ 寄存器中的 $POEN$ 、 ROS 、 IOS 、 $ISOx$ 和 $ISOxN$ 位。输出极性由 $TIMERx\_CHCTL2$ 寄存器中的 $CHxP$ 和 $MCHxP$ 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 21-5. 由参数控制的互补输出表 \(\$MCHxMSEL = 2'b11\$ \)](#)。

表 21-5. 由参数控制的互补输出表 ($MCHxMSEL = 2'b11$)

互补参数					输出状态	
POEN	ROS	IOS	CHxEN	MCHxEN	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 = CHxNP；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		

互补参数					输出状态	
POEN	ROS	IOS	CHxEN	MCHxEN	CHx_O	MCHx_O
		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 = CHxNP）；如果死区产生时钟未失效，在死区时间之后： CHx_O = ISOx，MCHx_O = ISOxN	
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 ⊕ ⁽²⁾ 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) ⁽³⁾ ⊕ 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 / MCHx_O 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 / MCHx_O 输出无效电平（CHx_O = 0[⊕] CHxP = CHxP）；
- (3) 详情见中止模式章节。
- (4) ⊕：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\(TIMERx_CCHP\)](#)。

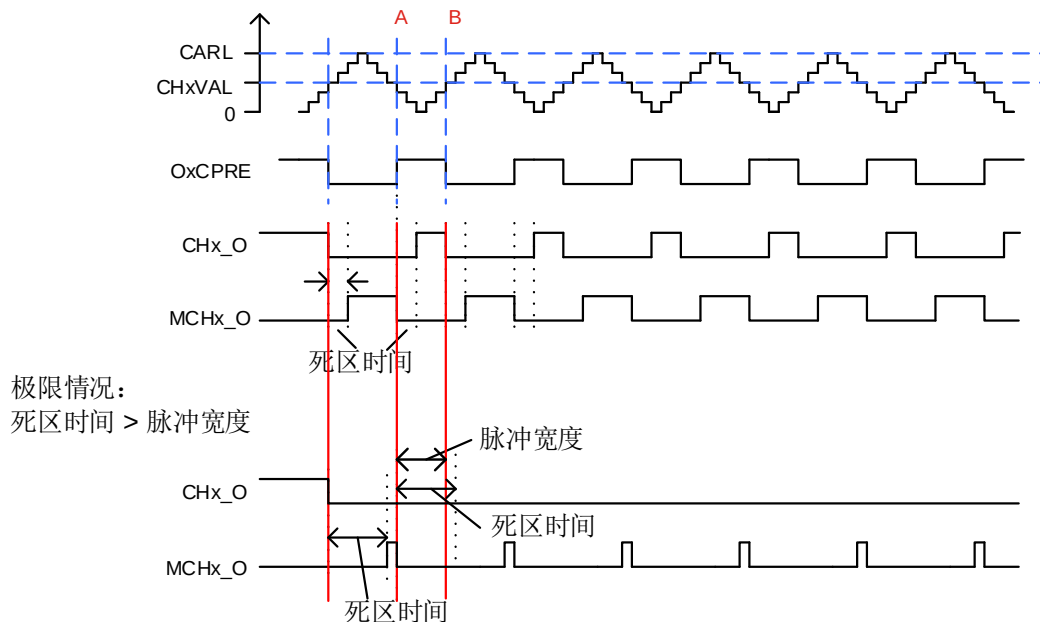
死区时间的插入，确保了通道互补的两路信号不会同时有效。

在 PWM0 模式，当通道 x 匹配发生时（TIMERx 计数器 = TIMERx_CHxCV），OxCPRE 翻转。在[图 21-34. 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 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 21-34. 带死区时间的互补输出](#) 所示。

图 21-34. 带死区时间的互补输出



中止功能

使用中止功能时, CHx_O 和 MCHx_O 信号的输出电平由以下位控制: TIMERx_CCHP 寄存器的 POEN、IOS 和 ROS 位, TIMERx_CTL1 寄存器的 ISOx 和 ISOxN 位。

当 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。

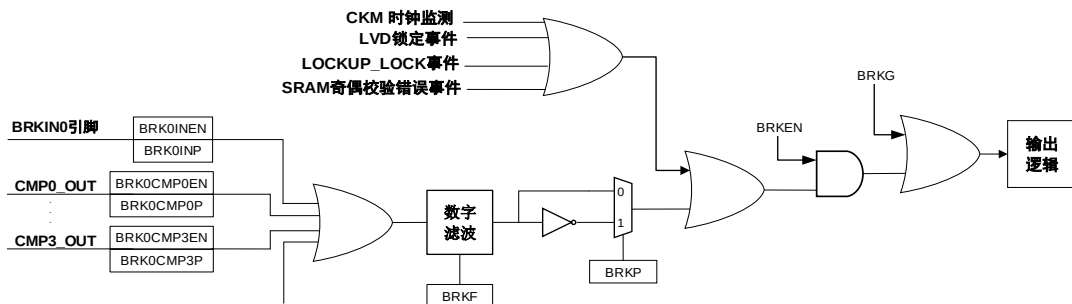
中止源是系统事件和其他事件。当发生中止事件时, 在死区持续时间之后, 输出将以预定的电平 (有效或无效) 强制输出。中止功能通过将 TIMERx_CCHP 寄存器中的 BRKEN 位置 1 来使能。中止输入极性由 TIMERx_CCHP 寄存器中的 BRKP 位配置。

中止事件是所有源逻辑或运算的结果。中止功能可以处理两种类型的事件源:

- 外部信号源: 来自外部引脚 BRKIN0 输入, 并具有数字滤波器和极性选择;
- 内部源: 系统源 (如: RCU 时钟监控器 CKM 产生的 HXTAL 卡死事件、LVD 锁事件、LOCKUP_LOCK 事件、SRAM 奇偶错误事件)、片上比较器事件 (四个比较器经过数字滤波和极性选择后的四个输出作为 TIMERx 输入)。

中止事件也可以由软件置位 TIMERx_SWEVG 寄存器中的 BRKG 位产生。参考 [图 21-35. BREAK0 中止功能逻辑图](#)。

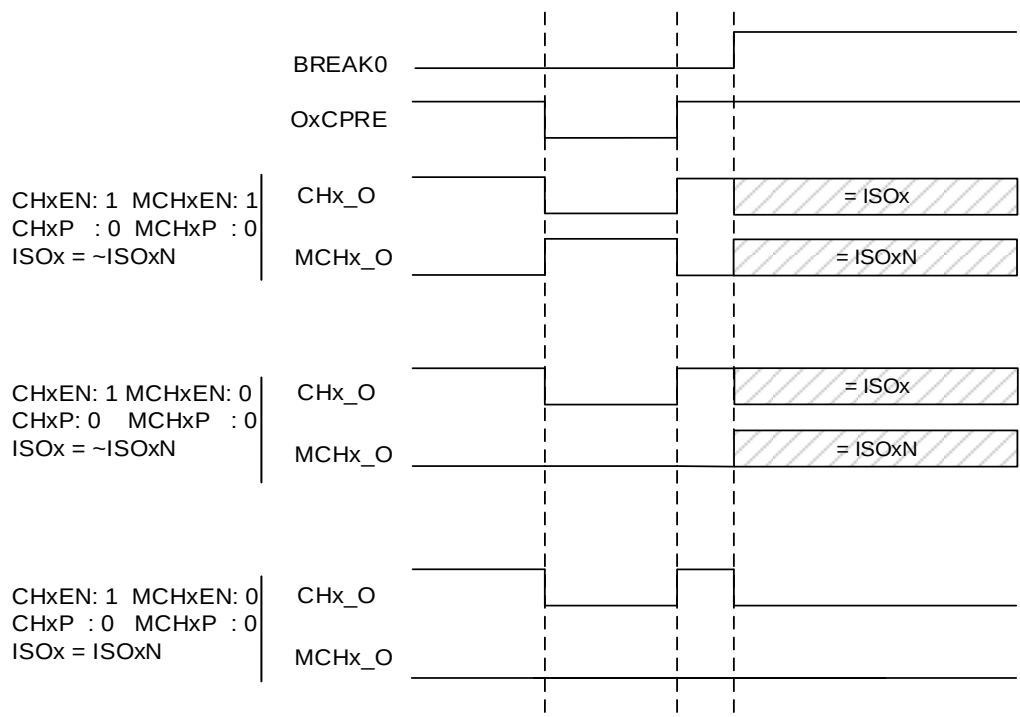
图 21-35. BREAK0 中止功能逻辑图



中止功能图中的 BRKIN0 引脚选自 GPIO，BRKCMPIx 选自比较器，由 BRKCMPIxEN 使能位和 BRKCMPIxP 极性选择控制。当系统中止源激活时，由硬件设置 SYSBIF 标志。软件等待系统中止源变为非激活状态，然后清除 SYSBIF 标志。

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 21-36. 通道响应 BREAK0 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 21-36. 通道响应 BREAK0 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。

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..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 21-6. [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 21-6. 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 21-37. 译码器接口模式下计数器运行例子

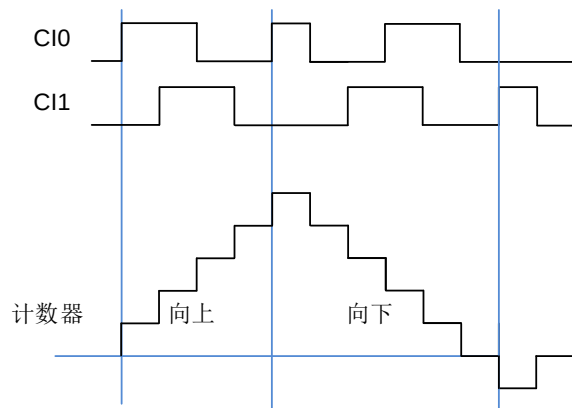
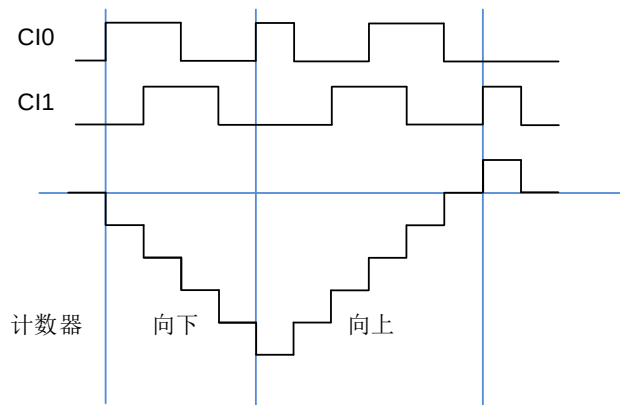


图 21-38. CI0FE0 极性反相的译码器接口模式下的例子



非正交译码器

非正交译码器功能有 4 种模式：非正交译码器模式 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。

使用非正交译码器模式 0/1 时，CI0FE0 作为计数方向信号，CI1FE1 作为计数脉冲。

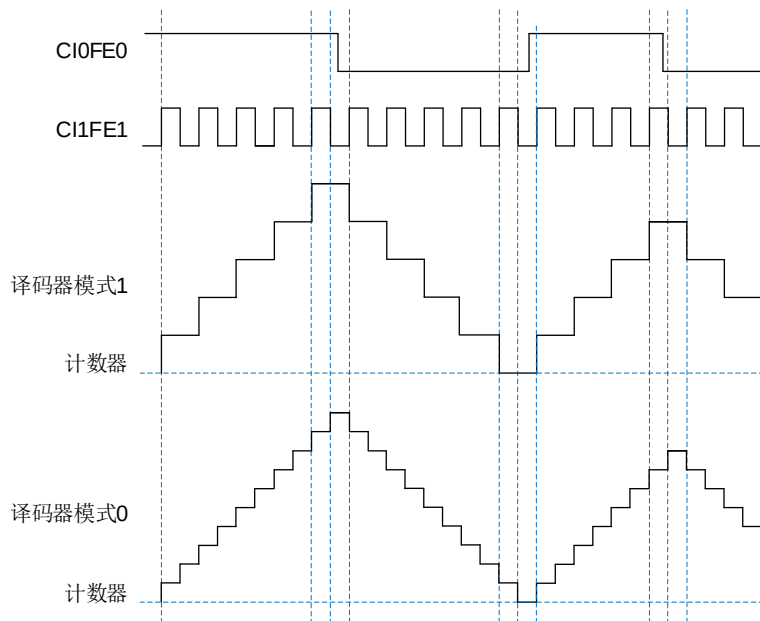
其中，CH0P 用于计数方向选择：当 CH0P=0 时，CI0FE0 为高电平时向上计数，CI0FE0 为低电平时向下计数；当 CH0P=1 时，CI0FE0 为高电平时向下计数，CI0FE0 为低电平时向上计数。

CH1P 用于选择 CI1FE1 信号的计数边沿：非正交译码器模式 0 时，计数器在 CI1FE1 信号的上升沿和下降沿进行计数；非正交译码器模式 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。更多非正交译码器模式 1 的细节如[表 21-7. 非正交译码器模式 1 的计数情况](#)和[图 21-39. 非正交译码器模式 0/1 计数器运行实例 \(CH1P=0\)](#)所示。

表 21-7. 非正交译码器模式 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 21-39. 非正交译码器模式 0/1 计数器运行实例 (CH1P=0)



非正交译码器模式 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。

非正交译码器模式 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；非正交译码器模式 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 21-40. 非正交译码器模式 2 / 3 计数器运行实例 \(CH0P / CH1P=0\)](#) 和 [表 21-8. 非正交译码器模式 2 / 3 下的计数方向](#)。

图 21-40. 非正交译码器模式 2 / 3 计数器运行实例 (CH0P / CH1P=0)

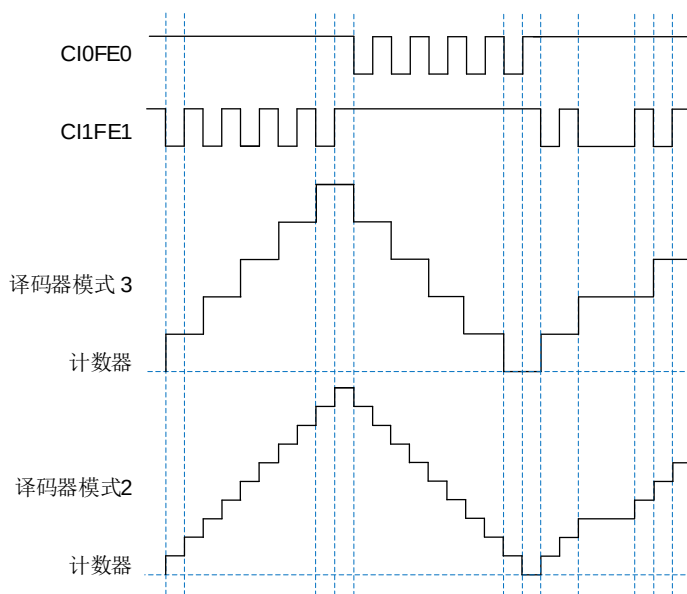


表 21-8. 非正交译码器模式 2 / 3 下的计数方向

计数模式	极性	电平	CI0FE0		CI1FE1	
			上升	下降	上升	下降
非正交译码器模式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
非正交译码器模式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	-

当非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 位改变，同时 TIMERx_INTF 寄存器中的 DIRTRANIF 标志位置 1。若 TIMERx_DMAINTEN 寄存器中的 DIRTRANIE 位置 1，则产生相应的中断。

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 21-41. 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，基于内部连接关系，选择互连的定时器，例如：

TIMER_in (TIMER0) -> TIMER_out (TIMER7 ITI0)

TIMER_in (TIMER1) -> TIMER_out (TIMER0 ITIO)

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，CHOVAL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 21-41. 霍尔传感器用在 BLDC 电机控制中

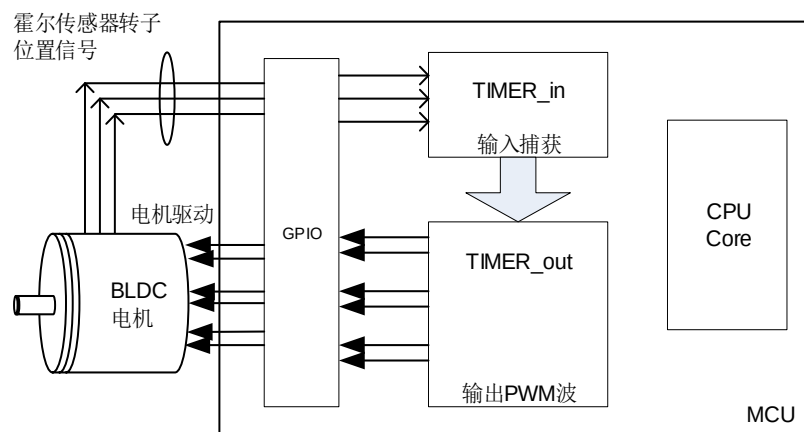
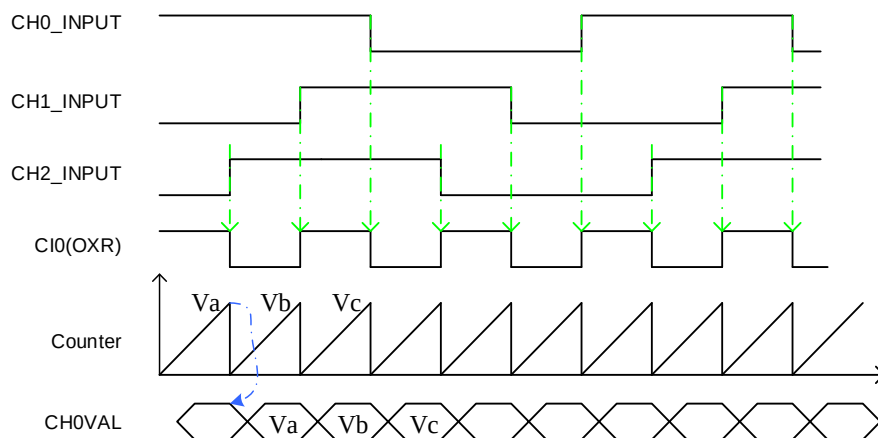
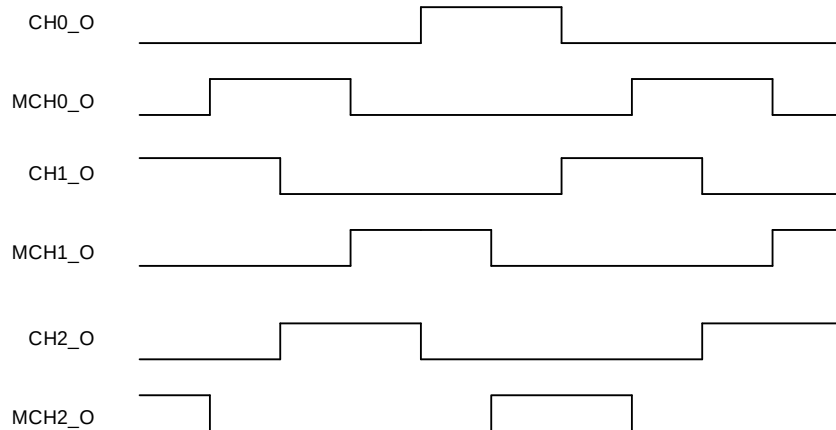


图 21-42. 两个定时器之间的霍尔传感器时序图

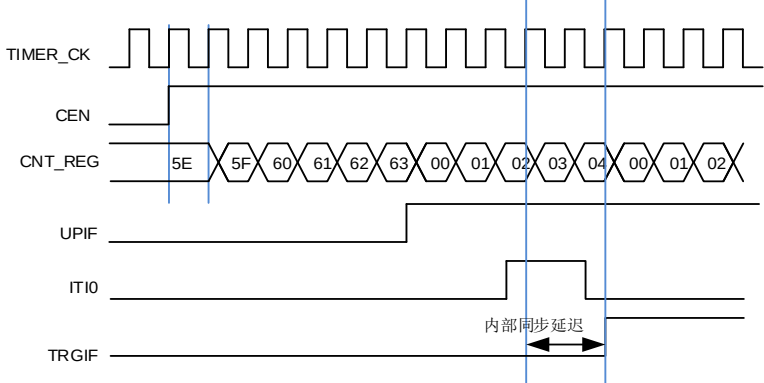
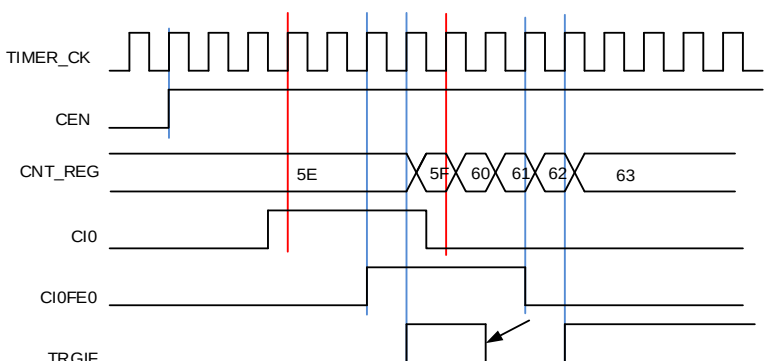
高级/通用 L0 定时器 **TIMER_in** 工作在输入捕获模式高级定时器 **TIMER_out** 工作在输出比较模式(带有死区的PWM)

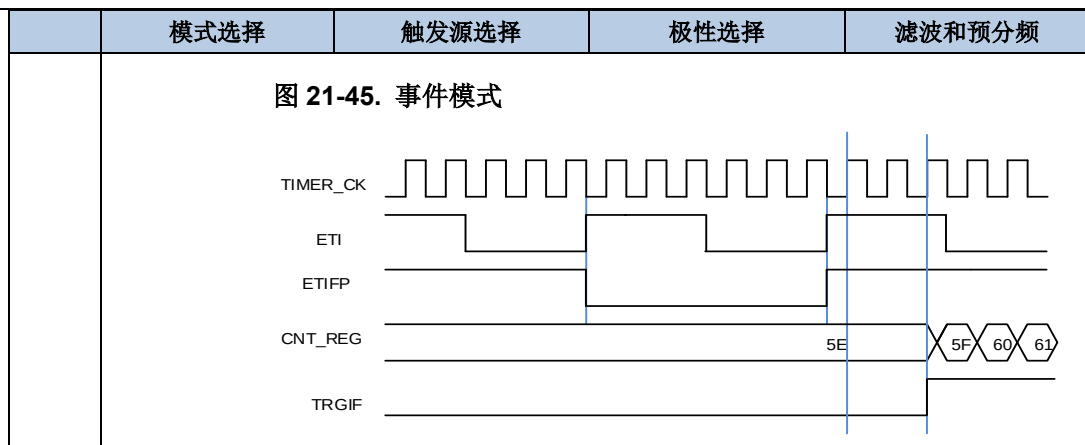
主-从管理

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..7) 位域来确定，具体的输入触发源可以通过 TSCFGy[4:0] (y = 3..7) 位域值来选择。

表 21-9. 从模式例子列表

	模式选择	触发源选择	极性选择	滤波和预分频
列举	TSCFGy[4:0]	TSCFGy[4:0]	如果触发源是 C1xFEx(x = 0...3) 或者 MCIxFEMx(x = 0...3)，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。
	y = 3: 复位模式	00000: 模式禁能		触发源 C1x/MCIx，配置 CHxCAPFLT/
	y = 4: 暂停模式	00001: ITI0		MCHxCAPFLT 设置滤波，分频不可用。
	y = 5: 事件模式	00010: ITI1		触发源是 ETIFP，滤波和预分频不可用。
	y = 6: 外部时钟模式	00011: ITI2		
	0	00100: ITI3		
	y = 7: 复位 + 事件模式	00101: CI0F_ED 00110: CI0FE0	如果触发源是 ETIFP(滤波后的 ETI 外部触发输	

	模式选择	触发源选择	极性选择	滤波和预分频
		00111: CI1FE1 01000: ETIFP 01001: CI2FE2 01010: CI3FE3 01011: MCI0FEM0 01100: MCI1FEM1 01101: MCI2FEM2 01110: MCI3FEM3	入), 配置ETP选择极性和反相。	
例1	复位模式 当触发输入上升沿, 计数器清零重启	TSCFG3[4:0] = 5'b00001, 选择ITIO为触发源	触发源是ITIO, 极性选择不可用	触发源是ITIO, 滤波和预分频不可用
	图 21-43. 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	TIOS=0 (非异或) [MCH0P=0, CH0P=0] 在这个例子中滤波被旁路 CI0FE0不反相, 在上升沿捕获	
	图 21-44. 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

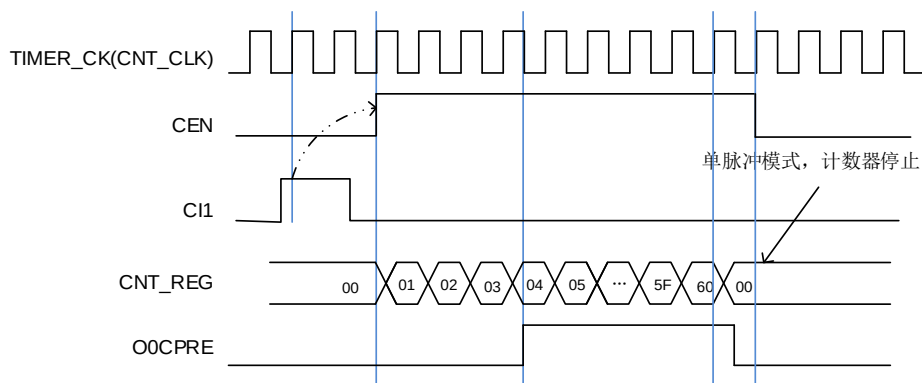
单脉冲模式与重复模式是相反的，设置 `TIMERx_CTL0` 寄存器的 `SPM` 位置 1，可使得单脉冲模式。当 `SPM` 置 1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置 `CHxCOMCTL` / `MCHxCOMCTL` 配置 `TIMERx` 为 PWM 模式或者比较模式。

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 模式和非对称 PWM 模式。

图 21-46. 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。互连的例子如下：

■ 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 = 3'b010）。定时器2在每次计数器溢出产生更新事件时，输出一个周期信号；
2. 配置定时器2周期（TIMER2_CAR寄存器）；
3. 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TRCFG6[4:0] = 5'b00011）；
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）。

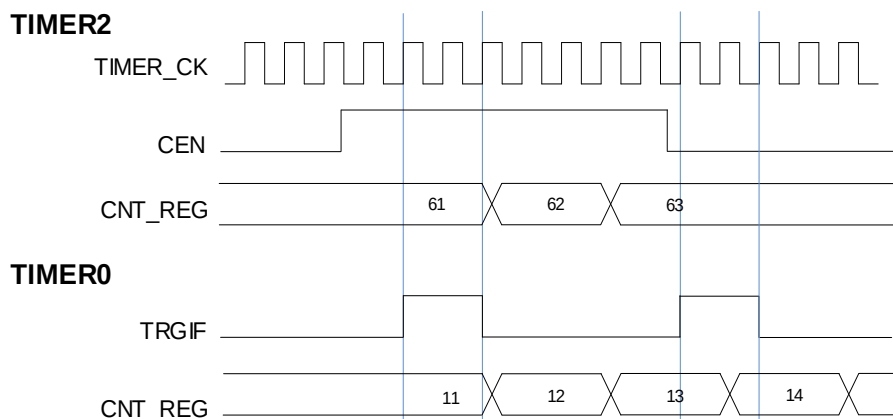
■ 用定时器2的使能/更新信号来启动定时器0

用定时器2的使能信号来启动定时器0，见[图 21-47. 用定时器2的使能信号触发定时器0](#)。在定时器2使能信号输出后，定时器0按照分频后的内部时钟从当前值开始计数。

当定时器0接收到触发信号，它的CEN位被自动置1，计数器计数直到禁能定时器0。两个定时器的计数器频率都是TIMER_CK经过预分频器3分频后频率（ $f_{CNT\_CLK} = f_{TIMER\_CK}/3$ ）。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 = 3'b001）；
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TRCFG5[4:0] = 5'b00011）；
3. 写1到CEN来开启定时器2（TIMER2_CTL0寄存器）。

图 21-47. 用定时器2的使能信号触发定时器0



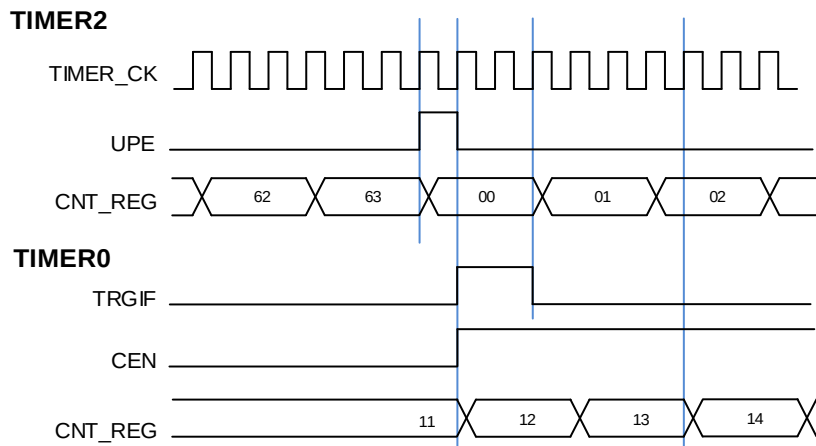
在这个例子中，我们也可以使用更新事件代替使能信号作为触发源。见[图 21-48. 用定时器2的更新事件来触发定时器0](#)，按以下步骤进行：

1. 配置定时器2为主模式，发送它的更新事件（UPE）作为触发输出（配置TIMER2_CTL1寄存器的MMC = 3'b010）；
2. 配置定时器2的周期（TIMER2_CARL寄存器）；
3. 配置定时器0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TRCFG5[4:0] = 5'b00011)

4. 写1到CEN来开启定时器2 (TIMER2_CTL0寄存器)。

图 21-48. 用定时器 2 的更新事件来触发定时器 0

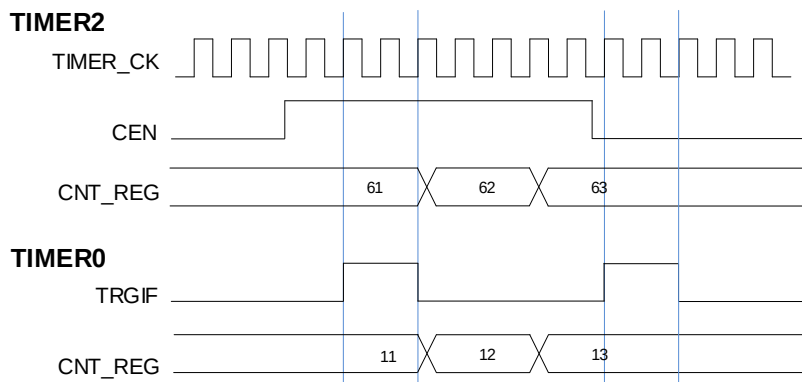


- 使用定时器2的使能/O0CPRE参考信号来使能定时器0计数。

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 21-49. 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 = 3'b001）；
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TRCFG4[4:0] = 5'b00011）；
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）。

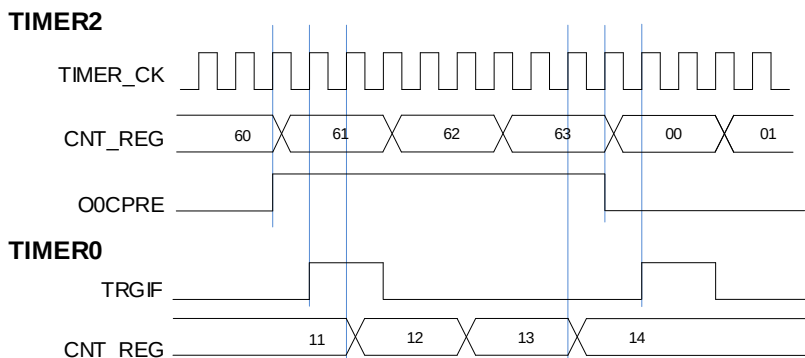
图 21-49. 用定时器 2 的使能来选通定时器 0



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MMC=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TRCFG4[4:0] = 5'b00011）；
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 21-50. 用定时器 2 的 O0CPRE 信号选通定时器 1



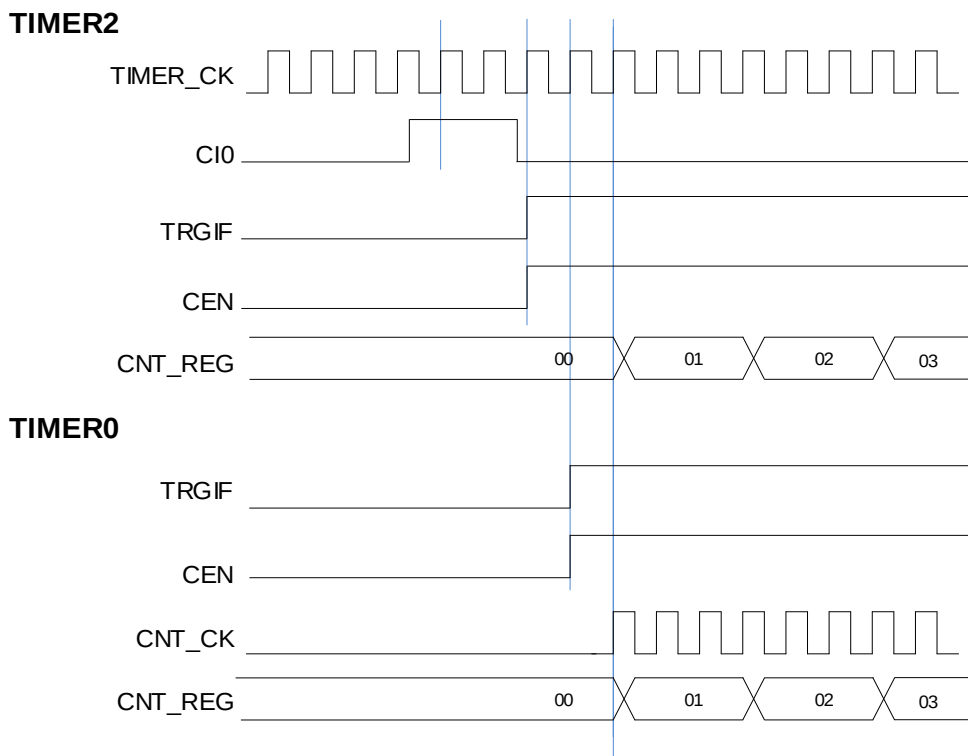
■ 使用一个外部触发来同步两个定时器

配置定时器 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TRCFG5[4:0] = 5'b00101）；
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TRCFG5[4:0] = 5'b00011）。

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 21-51. 用定时器 2 的 CI0 输入来触发定时器 0 和定时器 2



ADC 触发请求

定时器可以通过各种内部信号产生触发事件，例如复位、使能、更新或比较事件。也可生成由内部边沿检测器发出的脉冲。

此外，还可以使用专用的 ADC 比较事件和计数器溢出来生成各自的触发信号并分别触发 ADC。下面将分别介绍几种触发 ADC 的方法。

1) 计数器溢出触发模式

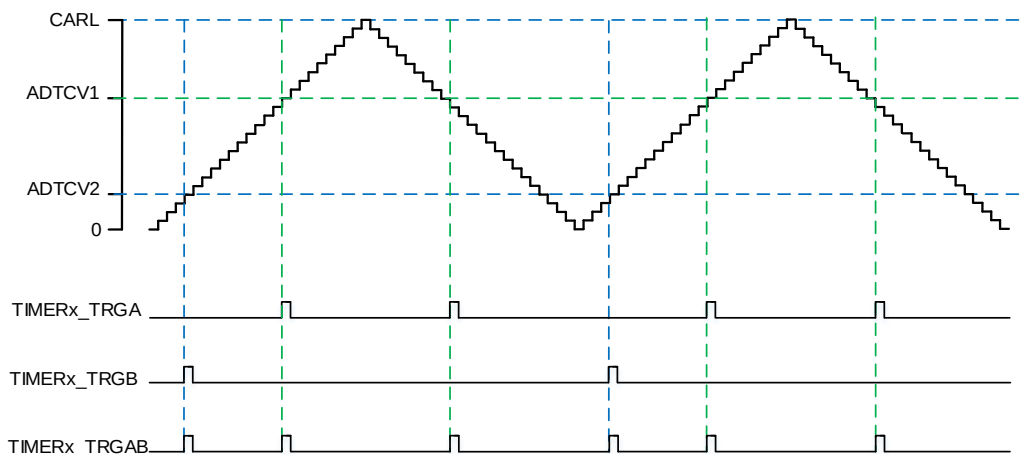
当置位 `TIMERx_ADCTL` 寄存器中的 `ADTSOF` 位时，计数器溢出将产生一个单独的触发信号 `TIMERx_TRGOF`。当置位 `TIMERx_ADCTL` 寄存器中的 `ADTSUF` 位时，计数器溢出将产生一个单独的触发信号 `TIMERx_TRGUF`。

2) ADC 比较匹配触发模式

定时器有两个专门的事件触发比较寄存器(`TIMERx_ADCCR1` 和 `TIMERx_ADCCR2`)。当发生比较匹配时，可以生成触发信号 `TIMERx_TRGA` 和 `TIMERx_TRGB` 来触发 ADC，它们还可以共同作用来生成触发信号 `TIMERx_TRGAB`。可配置 `TIMERx_TRGA` 和 `TIMERx_TRGB` 的向上计数时比较匹配使能(`ADT1UPGTEN`, `ADT2UPGTEN`)或向下计数时比较匹配使能(`ADT1DWGTEN`, `ADT2DWGTEN`)寄存器，可分别选择上升沿、下降沿、上升沿和下降沿产生触发信号或不产生触发信号。如 [图 21-52. ADC 触发信号产生在 ADC 不同比较匹配触发模式 \(`ADT1UPGTEN=1`, `ADT2UPGTEN=1`, `ADT1DWGTEN=1`, `ADT2DWGTEN=0`\)](#)所示。

图 21-52. ADC 触发信号产生在 ADC 不同比较匹配触发模式 (`ADT1UPGTEN=1`,

ADT2UPGTEN=1, ADT1DWGTEN=1, ADT2DWGTEN=0)



注意：中央对齐模式下，当 $ADTCVx = 0$ 时，只有当向下计数 ADC 触发信号使能时才产生一个 ADC 触发脉冲($ADT1DWGTEN = 1 / ADT2DWGTEN = 1$)。在中央对齐模式下，如果 $ADTCVx = CARL$ ，则只有当向上计数 ADC 触发信号使能时才会产生一个 ADC 触发脉冲($ADT1UPGTEN = 1 / ADT2UPGTEN = 1$)。

ADTCVx 有一个对应的缓冲区，根据影子寄存器加载功能的相关配置更新比较寄存器。

ADC 触发信号表及其对应的信号产生方法如[表 21-10. TIMER0/TIMER7 ADC 触发器信号及其操作](#)所示。

表 21-10. TIMER0/TIMER7 ADC 触发器信号及其操作

ADC触发信号	产生模式
TIMERx_TRGOF	计数器上溢触发
TIMERx_TRGUF	计数器下溢触发
TIMERx_TRGA	ADC触发比较值1比较匹配且满足专用重复计数器功能
TIMERx_TRGB	ADC触发比较值2比较匹配且满足专用重复计数器功能
TIMERx_TRGAB	ADC触发比较值1或2比较匹配且满足专用重复计数器功能
TIMERx_TRGAORB	ADC触发比较值1或2比较匹配且需要满足TIMERx_TRGA、TIMERx_TRGB重复计数器功能

注意：除专门产生 ADC 触发信号外，复位、使能、更新或比较事件和内部边缘检测器产生的 TRGO 信号也可以触发 ADC(根据 MMC 寄存器选择，只能输出一个触发信号 TRGO)。

3) ADC 重复计数器功能

ADC 比较匹配触发模式具有两种重复计数器模式，周期重复计数器模式，比较匹配重复计数器模式。

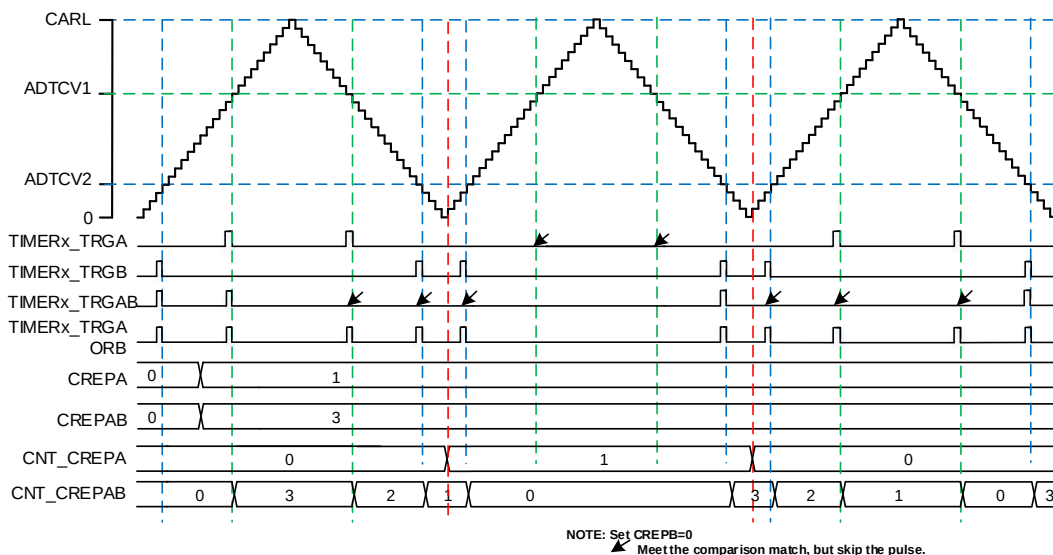
周期重复计数器模式应用 TIMERx_TRGA 或者 TIMERx_TRGB 信号。根据设定的 4bit ADC 重复计数器值 ADCREPA、ADCREPB，来分别实现 ADC 的 TIMERx_TRGA 或者 TIMERx_TRGB 信号的重复计数器功能。当每次通过 TIMER 的上溢或下溢来生成 TIMERx_TRGA 或 TIMERx_TRGB 时，在生成事件触发信号之前，计数器将不断减小，直到达到 0 时生效。当重复计数器为 0 时，不使用重复计数器功能。

比较匹配重复计数器模式应用于 $TIMERx\_TRGAB$ 信号。根据设定的 4bit ADC 重复计数器值 $ADCREPAB$ ，可以实现 ADC 的 $TIMERx\_TRGAB$ 信号的重复计数器功能。根据有效的事件触发信号 $TIMERx\_TRGA$ 、 $TIMERx\_TRGB$ ，用于 $TIMERx\_TRGAB$ 信号的重复计数器从 $ADCREPAB$ 重复计数值开始递减，直到重复计数器为 0 时，才会生成事件触发信号 $TIMERx\_TRGAB$ 。当重复计数器为 0 时，不使用重复计数器功能。

比较匹配触发模式有单独的重复计数器功能。根据设置的 4 位域 ADC 重复计数器值 ($TIMERx\_ADCREP$ 寄存器中的 $ADCREPA$ 、 $ADCREPB$ 、 $ADCREPAB$ 位)，可以分别实现 ADC 的 $TIMERx\_TRGA$ 或 $TIMERx\_TRGB$ 信号的重复计数器功能。当每次通过比较和匹配触发器来生成 $TIMERx\_TRGA$ 或 $TIMERx\_TRGB$ 时，在生成事件触发信号之前，计数器将不断减小，直到达到 0。当重复计数器为 0 时，不使用重复计数器功能。

如 [图 21-53. ADC 触发模式下的 ADC 重复计数功能\(ADT1UPGTEN =1, ADT2UPGTEN =1, ADT1DWGTEN =1, ADT2DWGTEN =1\)](#) 所示。

图 21-53. ADC 触发模式下的 ADC 重复计数功能(ADT1UPGTEN =1, ADT2UPGTEN =1, ADT1DWGTEN =1, ADT2DWGTEN =1)



注意：当修改 $ADCREPA$ 、 $ADCREPB$ 和 $ADCREPAB$ 值时，不会立刻更新，而是在相应的计数器计数到 0 时，更新重复计数器的值。若 $TIMERx\_ADCTL$ 寄存器的 $ADREPRLD$ 位置位，则立即更新重复计数器的值。

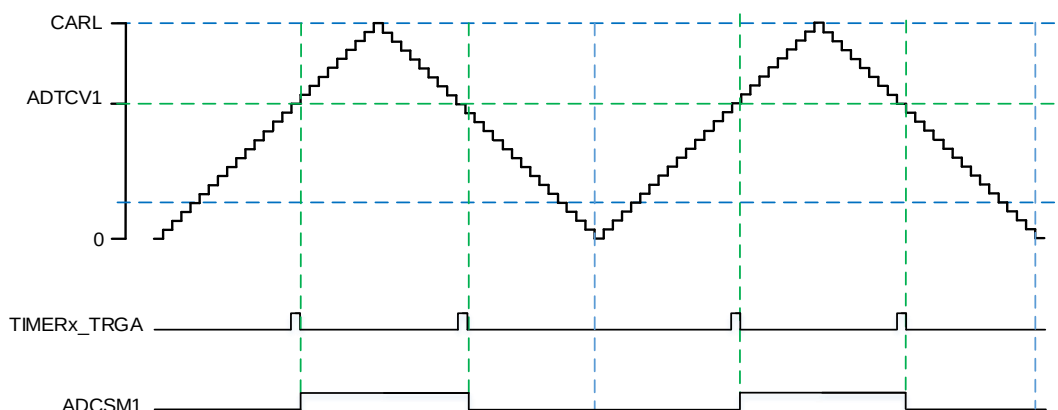
使用 ADC 重复计数器进行 ADC 同步的具体操作步骤如下：

1. 在 ADC 触发器比较值寄存器($ADTCV1$ 和 $ADTCV2$)中设置比较值；
2. 设置影子寄存器加载功能相关寄存器，并设置比较值更新时刻；
3. 分别配置 ADC 重复计数器的值；
4. 配置 $ADCREPA$ 、 $ADCREPB$ 在定时器上溢或下溢时递减，选配重复计数器立即重载功能；
5. 设置边沿比较匹配使能($ADT1UPGTEN$, $ADT2UPGTEN$, $ADT1DWGTEN$, $ADT2DWGTEN$)使能 ADC 触发信号的产生。

4) ADC 触发信号监视功能

使用外部引脚 ADCSM1/2 监控 ADC 触发器启动信号。ADC 触发信号监测选择寄存器 (ADCSM1SEL、ADCSM2SEL) 分别选择 ADCSM1/2 要监测的 ADC 触发信号, ADCSM1/2 将其输出到外部引脚。当检测到 ADC 触发器启动信号时, 电平翻转。如 [图 21-54. ADC 触发器同步监控器信号时序图\(ADCSM1SEL = 3'b010\)](#) 所示。

图 21-54. ADC 触发器同步监控器信号时序图(ADCSM1SEL = 3'b010)



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_DMACHCFG` 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_DMACHCFG` 寄存器中的 `DMATA` 来指定。如果 `TIMERx_DMACHCFG` 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_DMACHCFG` 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。

输出 DIR 位

DIR 位可以在 CH2 和 CH3 通道上输出, 配置在 `TIMERx_CHCTL1` 寄存器中的 `CH2COMCTL[3:0]` 或 `CH3COMCTL[3:0]` 位域值为 4b'1011 来使能该功能。

当计数器工作在中央对齐模式时, 该功能可用于指示计数器的计数方向。当计数器工作在译码器模式时, 此功能可用于指示外部信号的旋转方向。

UPIF 位备份功能

可以通过配置 `TIMERx_CTL0` 寄存器中的 `UPIFBUEN` 位来使能 UPIF 位的备份功能, UPIF 和

UPIFBU 位之间没有延迟，两者完全同步。

使能该功能后，TIMERx_INTF 寄存器中的 UPIF 位将会被实时备份到 TIMERx_CNT 寄存器中的 UPIFBU 位。这可以避免在读计数器和中断处理时产生冲突的情况。

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。

21.1.5. TIMERx 寄存器 (x = 0, 7)

TIMER0 基地址: 0x4001 2C00

TIMER7 基地址: 0x4001 3400

控制寄存器 0 (TIMERx_CTL0)

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UPSSEL[1:0]		INTCOSEL[1:0]		保留		ETISEL[1:0]		保留	INTREPR	保留					
rw		rw				rw			rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				UPIFBUE	保留	CKDIV[1:0]		ARSE	CAM[1:0]		DIR	SPM	UPS	UPDIS	CEN
				rw		rw		rw	rw		rw	rw	rw	rw	rw

位/位域	名称	描述
31:30	UPSSEL[1:0]	更新重复计数器溢出源选择 00: 计数器上溢/下溢触发更新重复计数器递减, 当更新重复计数器减到 0 时产生更新事件。 01: 计数器上溢触发更新重复计数器递减, 当更新重复计数器减到 0 时产生更新事件。 10: 计数器下溢触发更新重复计数器递减, 当更新重复计数器减到 0 时产生更新事件。 11: 保留 注: 在单独的向上计数或者向下计数模式下, 配置 01 或者 10, 重复计数器将在溢出时递减, 且在重复计数器递减到 0 时, 计数器溢出时产生更新事件。
29:28	INTCOSEL[1:0]	中断重复计数器溢出方式选择 00: 计数器上溢/下溢触发中断重复计数器递减, 当中断重复计数器递减到 0 时产生中断事件。 01: 计数器上溢会触发中断重复计数器递减, 且在中断重复计数器递减到 0 时产生中断事件。 10: 计数器下溢会触发中断重复计数器递减, 且在中断重复计数器递减到 0 时产生中断事件。 11: 保留 注: 在单独的向上计数或者向下计数模式下, 配置 01 或者 10, 重复计数器将在溢出时递减, 且在中断重复计数器递减到 0 时, 计数器溢出时产生中断溢出事件。
27:26	保留	必须保持复位值。
25:24	ETISEL[1:0]	TIMER 外部触发输入选择

		00: 从 GPIO 选择 TIMER_ETI0 引脚
		01: 从 GPIO 选择 TIMER_ETI1 引脚
		10~11: 保留
23	保留	必须保持复位值。
22	INTREPRD	中断重复计数器立即重新加载 在该比特位设置为1时，中断重复计数器将重新加载到相应的重复计数设置值。 在重新加载完成后，该比特位由硬件自动清除。
21:12	保留	必须保持复位值。
11	UPIFBUEN	UPIF位备份使能 0: 备份禁能。UPIF位没有备份到TIMERx_CNT寄存器中的UPIFBU位 1: 备份使能。UPIF 位备份到 TIMERx_CNT 寄存器中的 UPIFBU 位
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=000), 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=000), 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=000), 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。事件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													
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。

29:16	保留	必须保持复位值。
15	ISO3N	多模式通道 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	多模式通道 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	多模式通道 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	多模式通道 0 的互补通道空闲状态输出 0: 当 POEN 复位, MCH0_O 输出低电平 1: 当 POEN 复位, MCH0_O 输出高电平 此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]位为 00 的时候可以被更改。
8	ISO0	通道 0 的空闲状态输出 0: 当 POEN 复位, CH0_O 输出低电平 1: 当 POEN 复位, CH0_O 输出高电平 如果 MCH0_O 生效, 一个死区时间后 CH0_O 输出改变。此位只有在 TIMERx_CCHP 寄存器的 PROT [1:0]位为 00 的时候可以被更改。
7	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 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。 000: 复位。TIMERx_SWEVG 寄存器的 UPG 位被置 1 或从模式控制器产生复位触发一次 TRGO 脉冲, 后一种情况下, TRGO 上的信号相对实际的复位会有一个延迟。 001: 使能。此模式可用于同时启动多个定时器或控制在一段时间内使能从定时器。主模式控制器选择计数器使能信号作为触发输出 TRGO。当 CEN 控制位被置 1 或者暂停模式下触发输入为高电平时, 计数器使能信号被置 1。在暂停模式下, 计数器使能信号受控于触发输入, 在触发输入和 TRGO 上会有一个延迟, 除非选择了主/从模式。 010: 更新。主模式控制器选择更新事件作为 TRGO。 011: 捕获/比较脉冲。通道 0 在发生一次捕获或一次比较成功时, 主模式控制器产生一个 TRGO 脉冲。 100: 比较。在这种模式下主模式控制器选择 O0CPRE 信号作为触发输出 TRGO。

		101: 比较。在这种模式下主模式控制器选择 O1CPRE 信号作为触发输出 TRGO。
		110: 比较。在这种模式下主模式控制器选择 O2CPRE 信号作为触发输出 TRGO。
		111: 比较。在这种模式下主模式控制器选择 O3CPRE 信号作为触发输出 TRGO。
3	DMAS	DMA 请求源选择 0: 当通道捕获 / 比较事件发生时, 发送通道 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 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 (CHxEN、MCHxEN 和 CHxCOMCTL 位) 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、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

该寄存器只能按字 (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			OCRC	保留		
rw	rw	rw		rw				rw				rw			

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	ETP	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 5'b01000。 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，TRGI 用做启动事件。 0: 主从模式禁能

1: 主从模式使能

6:4	保留	必须保持复位值。
3	OCRC	OxCPRE / MOxCPRE清除源选择 该位用于选择OxCPRE / MOxCPRE清除源。 0: OCPRE_CLR_INT连接OCPRE_CLR输入 1: OCPRE_CLR_INT连接到ETIF
2:0	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 ADDIE	CH2COM ADDIE	CH1COM ADDIE	CH0COM ADDIE	MCH3 DEN	MCH2 DEN	MCH1 DEN	MCH0 DEN	MCH3IE	MCH2IE	MCH1IE	MCH0IE	保留	FLOWIE	保留	
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
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	MCH3DEN	多模式通道 3 比较/捕获 DMA 请求使能

		0: 禁止多模式通道 3 比较/捕获 DMA 请求 1: 使能多模式通道 3 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 MCH3MSEL[1:0] = 2'b00 时)。
26	MCH2DEN	多模式通道 2 比较/捕获 DMA 请求使能 0: 禁止多模式通道 2 比较/捕获 DMA 请求 1: 使能多模式通道 2 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 MCH2MSEL[1:0] = 2'b00 时)。
25	MCH1DEN	多模式通道 1 比较/捕获 DMA 请求使能 0: 禁止多模式通道 1 比较/捕获 DMA 请求 1: 使能多模式通道 1 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 MCH1MSEL[1:0] = 2'b00 时)。
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] = 2'b00 时)。
23	MCH3IE	多模式通道 3 比较/捕获中断使能 0: 禁止多模式通道 3 中断 1: 使能多模式通道 3 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 MCH3MSEL[1:0] = 2'b00 时)。
22	MCH2IE	多模式通道 2 比较/捕获中断使能 0: 禁止多模式通道 2 中断 1: 使能多模式通道 2 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 MCH2MSEL[1:0] = 2'b00 时)。
21	MCH1IE	多模式通道 1 比较/捕获中断使能 0: 禁止多模式通道 1 中断 1: 使能多模式通道 1 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 MCH1MSEL[1:0] = 2'b00 时)。
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] = 2'b00 时)。

19	保留	必须保持复位值。
18	FLOWIE	上溢或下溢中断使能 0: 上溢或下溢中断禁能 1: 上溢或下溢中断禁能
17:15	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	MCH3OF	MCH2OF	MCH1OF	MCH0OF	MCH3IF	MCH2IF	MCH1IF	MCH0IF	OVERFL OWIF	UNDERF LOWIF	Reserved	
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		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		SYSBIF	CH3OF	CH2OF	CH1OF	CH0OF	Reserved	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

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	MCH3OF	多模式通道 3 捕获溢出标志 参见 MCH0OF 描述。
26	MCH2OF	多模式通道 2 捕获溢出标志 参见 MCH0OF 描述。
25	MCH1OF	多模式通道 1 捕获溢出标志 参见 MCH0OF 描述。
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	MCH3IF	多模式通道 3 比较/捕获中断标志 参见 MCH0IF 描述。
22	MCH2IF	多模式通道 2 比较/捕获中断标志 参见 MCH0IF 描述。
21	MCH1IF	多模式通道 1 比较/捕获中断标志 参见 MCH0IF 描述。
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 TIMEx_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	OVERFLOWIF	上溢中断标志 在中断重复计数器上溢事件(计数值 = IREP)发生时，硬件会设置该标志。软件通过将其写入`0`来清除它。 0：无上溢中断 1：发生上溢中断 注意： 中断的上溢或下溢由INTCOSEL位发生。
18	UNDERFLOWIF	下溢中断标志 在中断重复计数器下溢事件(计数值 = IREP)发生时，硬件会设置该标志。软件通过将其写入`0`来清除它。 0：无下溢中断 1：发生下溢中断 注意： 中断的上溢或下溢由INTCOSEL位发生。
17:14	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 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
CH3COM ADDG	CH2COM ADDG	CH1COM ADDG	CH0COM ADDG	保留				MCH3G	MCH2G	MCH1G	MCH0G	保留			
W	W	W	W					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 模式和非对称 PWM 模式。

27:24	保留	必须保持复位值。
23	MCH3G	多模式通道 3 捕获或比较事件发生 参见 MCH0G 描述。
22	MCH2G	多模式通道 2 捕获或比较事件发生 参见 MCH0G 描述。
21	MCH1G	多模式通道 1 捕获或比较事件发生 参见 MCH0G 描述。
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	BRK0G	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 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 [2]	CH0MS [2]	CH1COM	CH0COM	保留			CH1COM	保留							CH0COM
		ADDSSEN	ADDSSEN				CTL[3]								CTL[3]
		保留	保留				保留								保留
rw	rw	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					

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，TIMERx_CH0COMV_ADD 寄存器的影子寄存器被使能，根据影子寄存器加载功能的相关配置在设置时刻进行更新。

		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（TIMERx_CTL0 寄存器的 SPM =1），可以在未确认预装载寄存器情况下使用 PWM 模式。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =000 时此位不能被改变。
27:25	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 MCH1MSEL[1:0] = 2'b00 时，TIMERx_CHCTL2 寄存器的 CH1EN 位清 0；当 MCH1MSEL[1:0] = 2'b01 或 2'b11 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 = 0, 7)寄存器中的 TSCFG15[4:0]位域选择）。 100: 通道 1 配置为输入，IS1 映射在 MCI1FE1 上 101~111: 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 OCPRE 清除源信号输入高电平时，O0CPRE 参考信号被清 0。OCPRE 清除源由 TIMERx_SMCFG 寄存器的 OCRC 位选择。 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 的值。O0CPRE 高电平有效，而 CH0_O 的有效电平取决于 CH0P 位。

注意：当多模式通道 0 配置为输出模式，且 MCH0MSEL[1:0] = 2'b11 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: 复合 PWM 模式 0，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，一旦计数器值等于 TIMERx_CH0CV_ADD 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 时，输出强制为无效电平，一旦计数器值等于 TIMERx_CH0CV_ADD 时，输出强制为有效电平。

1001: 复合 PWM 模式 1，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，一旦计数器值等于 TIMERx_CH0CV_ADD 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 时，输出强制为有效电平，一旦计数器值等于 TIMERx_CH0CV_ADD 时，输出强制为无效电平。

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（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](#)和[非对称 PWM 模式](#)在 PWM 模式（包括普通 PWM、复合 PWM 和非对称 PWM）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_CCHP 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，根据影子寄存器加载功能的相关配置在设置时刻进行更新。 0：禁止通道 0 输出比较影子寄存器 1：使能通道 0 输出比较影子寄存器 仅在单脉冲模式下（TIMERx_CTL0 寄存器的 SPM=1），可以在未确认预装载寄存器情况下使用 PWM 模式。 当 TIMERx_CCHP 寄存器的 PROT [1:0]=11 且 CH0MS =000 时此位不能被改变。
2	CH0COMFEN	通道0输出比较快速使能 0：通道0输出比较快速禁能。仅比较结果输入作为有效边沿时会产生比较匹配，并将 CH0_O 设置为比较电平，激活CH0_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。触发输入信号的有效边沿和比较结果都会产生比较匹配，并将CH0_O 设置为比较电平。当触发信号的输入作为有效边沿时，激活CH0_O 输出的最小延时为3个时钟周期。
1:0	CH0MS[1:0]	通道 0 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭（当 MCH0MSEL[1:0] = 2'b00 时，TIMERx_CHCTL2 寄存器的 CH1EN 位清 0；当 MCH0MSEL[1:0] = 2'b01 或 2'b11 时，TIMERx_CHCTL2 寄存器的 CH0EN、MCH0EN 位清 0）时，CH0MS[2:0]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 CI0FE0 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：通道 0 配置为输入，IS0 映射在 MCI0FE0 上 101~111：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 CIO 输入信号的采样频率和数字滤波器的长度。 0000: 无滤波器, $f_{SAMP} = f_{DTS}$, $N=1$ 0001: $f_{SAMP} = f_{CK\_TIMER}$, $N=2$ 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 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 ADDSEN	CH2COM ADDSEN	保留			CH3COM CTL[3]	保留						CH2COM CTL[3]	
		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
CH3COM CEN	CH3COMCTL[2:0]			CH3COM SEN	CH3COM FEN	CH3MS[1:0]		CH2COM CEN	CH2COMCTL[2:0]			CH2COM SEN	CH2COM FEN	CH2MS[1:0]	
CH3CAPFLT[3:0]				CH3CAPPSC[1:0]				CH2CAPFLT[3:0]			CH2CAPPSC[1:0]				
rw				rw		rw		rw			rw		rw		

输出比较模式:

位/位域	名称	描述
31	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, 根据影子寄存器加载功能的相关配置在设置时刻进行更新。 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。
27:25	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 输出比较控制 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方向和输入信号的选择。只有当通道关闭（当 MCH3MSEL[1:0] = 2'b00 时，TIMERx_CHCTL2 寄存器的 CH3EN 位清 0；当 MCH3MSEL[1:0] = 2'b01 或 2'b11 时，TIMERx_CHCTL2 寄存器的 CH3EN、MCH3EN 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，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 = 0, 7)寄存器中的 TSCFG15[4:0]位域选择）。 100：通道3配置为输入，IS3映射在MCI3FE3上。 101~111：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 OCPRE 清除源信号输入高电平时，O2CPRE 参考信号被清 0。OCPRE 清除源由 TIMERx_SMCFG 寄存器的 OCRC 位选择。 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 输出比较控制 参考 CH0COMCTL[2:0]描述。
3	CH2COMSEN	通道 0 输出比较影子寄存器使能 参考 CH0COMSEN 描述。
2	CH2COMFEN	通道2输出比较快速使能 参考 CH0COMFEN 描述。
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 MCH2MSEL[1:0] = 2'b00 时，TIMERx_CHCTL2 寄存器的 CH2EN 位清 0；当 MCH1MSEL[1:0] = 2'b01 或 2'b11 时，TIMERx_CHCTL2 寄存器的 CH2EN、MCH2EN 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：通道2配置为输入，IS2映射在MCI2FE2上。 101~111：保留

输入捕获模式：

位/位域	名称	描述
31	CH3MS[2]	通道 3 模式选择

		与输出模式相同。
30	CH2MS[2]	通道 2 模式选择 与输出模式相同。
29:16	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

该寄存器只能按字 (32 位) 访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3PERFOREN	CH2PERFOREN	CH1PERFOREN	CH0PERFOREN	保留											
rw	rw	rw	rw												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCH3P	MCH3EN	CH3P	CH3EN	MCH2P	MCH2EN	CH2P	CH2EN	MCH1P	MCH1EN	CH1P	CH1EN	MCH0P	MCH0EN	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	MCH3P	多模式通道 3 捕获 / 比较极性 参考 MCH0P 描述。
14	MCH3EN	多模式通道 3 捕获 / 比较使能 参考 MCH0EN 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	MCH2P	多模式通道 2 捕获 / 比较极性 参考 MCH0P 描述。
10	MCH2EN	多模式通道 2 捕获 / 比较使能 参考 MCH0EN 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	MCH1P	多模式通道 1 捕获 / 比较极性 参考 MCH0P 描述。
6	MCH1EN	多模式通道 1 捕获 / 比较使能 参考 MCH0EN 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	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] = 2'b11 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_CCHP 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_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

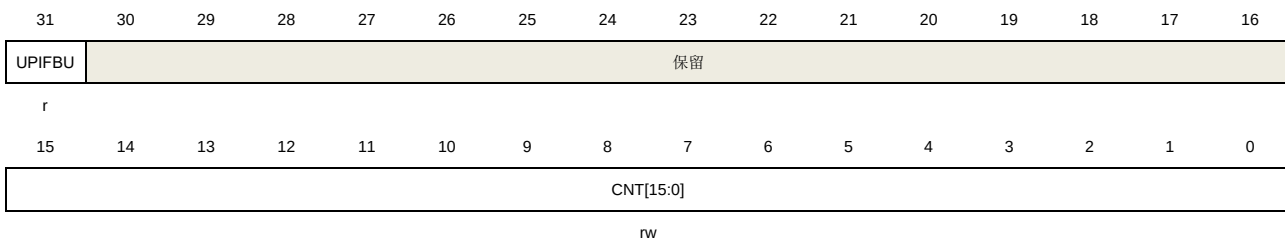
1: 使能通道 0

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。



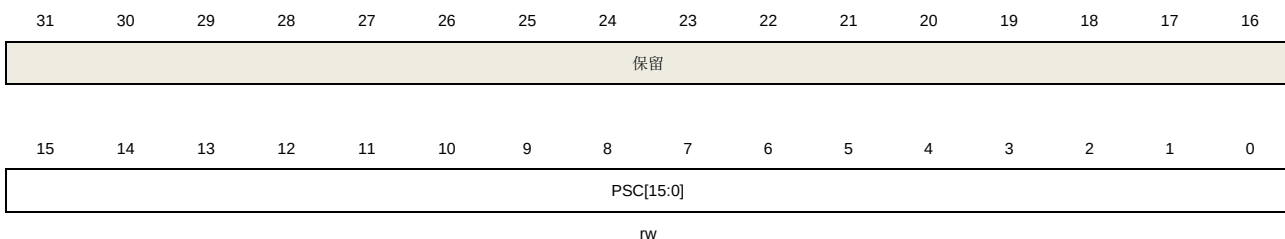
位/位域	名称	描述
31	UPIFBU	UPIF 位备份 该位只读，是 <code>TIMERx_INTF</code> 寄存器的 <code>UPIF</code> 位的备份值。当 <code>UPIFBUEN = 1</code> 时，该位有效，若 <code>UPIFBUEN = 0</code> ，该位保留，读取该位值为零。
30:16	保留	必须保持复位值。
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

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。



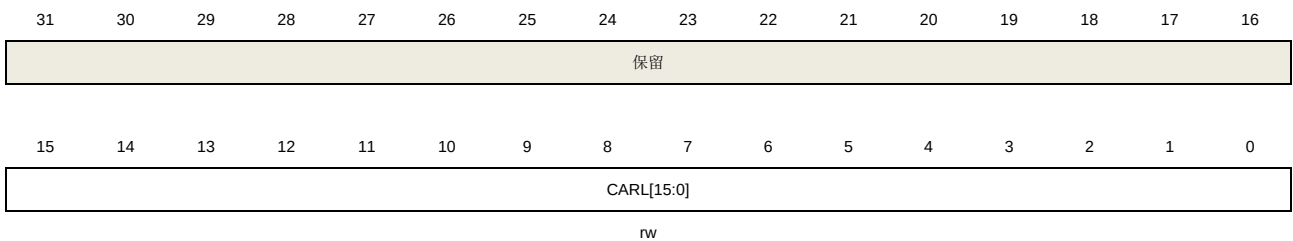
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 PSC 时钟除以 (PSC+1)，每次当更新事件产生时，PSC 的值被装入当前预分频寄存器。

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。



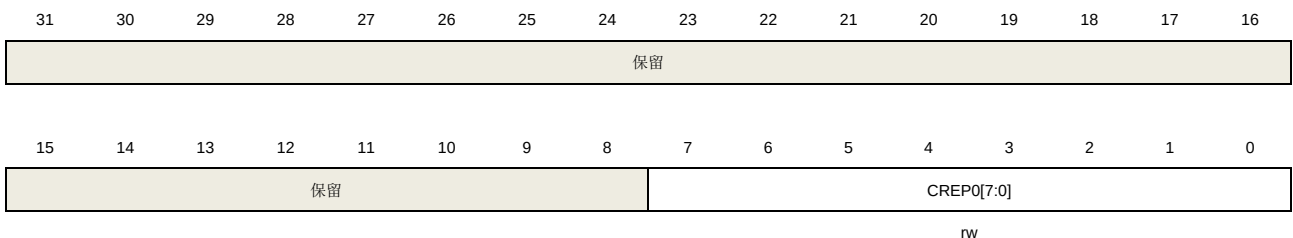
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CARL[15:0]	计数器自动重载值 (0 位至 15 位) 这些位定义了计数器的自动重载值。

重复计数寄存器 (TIMERx_CREP)

地址偏移: 0x30

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	CREP[7:0]	重复计数器的值 这些位定义了更新事件的产生速率。重复计数器计数值减为 0 时产生更新事件。影子寄存器的更新速率也会受这些位影响 (前提是影子寄存器被使能)。

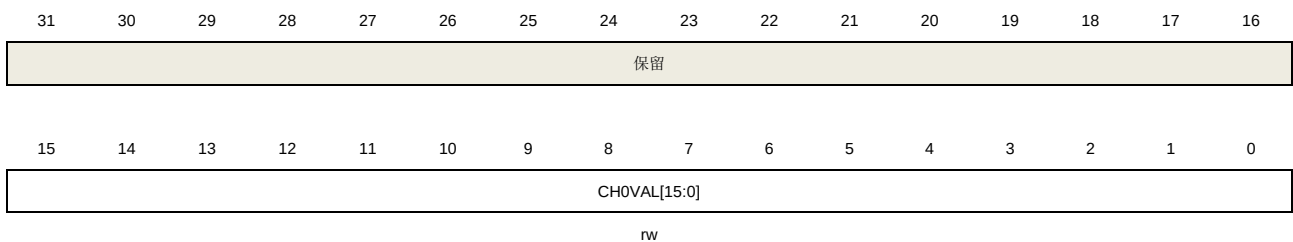
注意：当 TIMERx_CFG 寄存器中的 CREPSEL =0 时，使用该位域。

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。



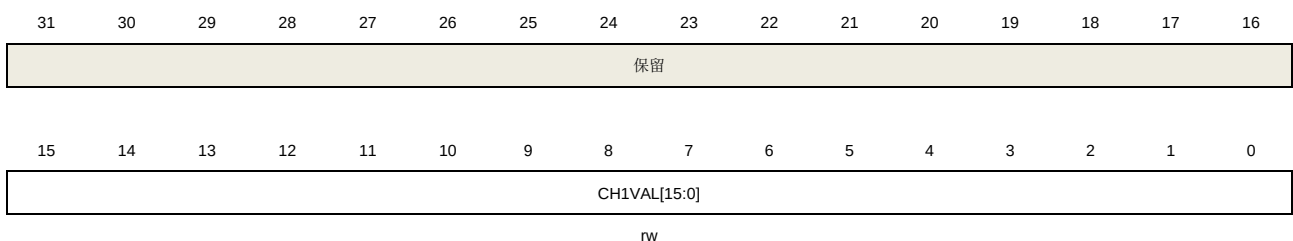
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH0VAL[15:0]	通道 0 的捕获或比较值 当通道 0 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器根据影子寄存器加载功能的相关配置在设定的时刻更新。

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。



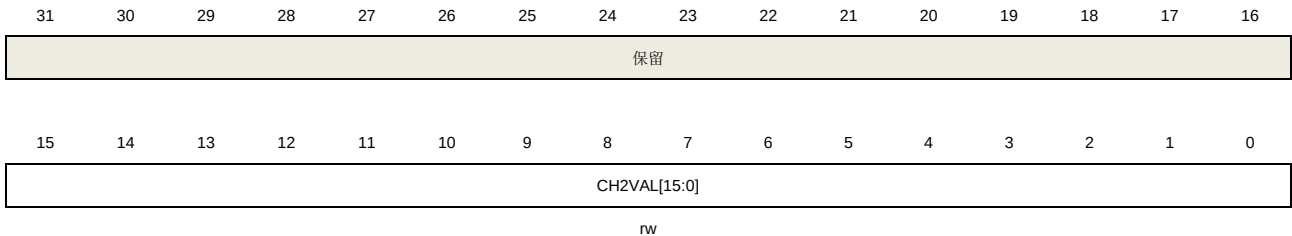
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH1VAL[15:0]	通道 1 的捕获或比较值 当通道 1 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 1 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器根据影子寄存器加载功能的相关配置在设定的时刻更新。

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。



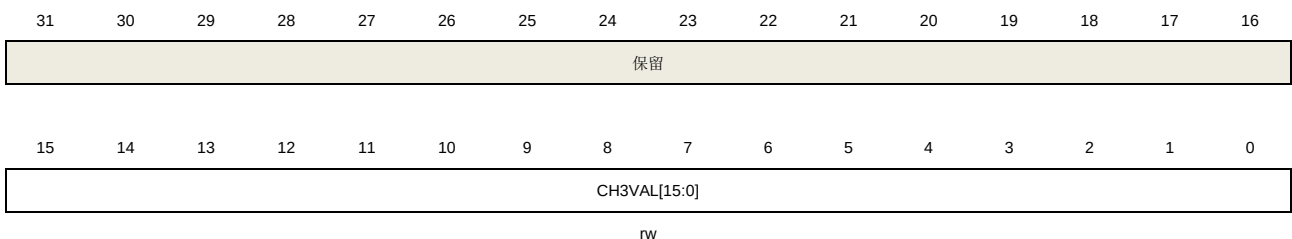
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH2VAL[15:0]	通道 2 的捕获或比较值 当通道 2 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 2 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器根据影子寄存器加载功能的相关配置在设定的时刻更新。

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，影子寄存器根据影子寄存器加载功能的相关配置在设定的时刻更新。

互补通道保护寄存器(TIMERx_CCHP)

地址偏移: 0x44

复位值: 0x0000 0000

该寄存器只能按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留												BRKF[3:0]			
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
31:20	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 BREAK0 输入信号的采样频率和数字滤波器的长度。 0000: 无滤波器，BREAK0 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_CCHP 寄存器的 PROT [1:0] = 00 时才可修改。
15	POEN	所有的通道输出使能 –该位由软件置位，或由硬件根据 OAEN 位自动设置 该位通过以下方式清 0: –写 0 清 0 –有效的中止输入（异步） 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 位，MCHxEN 位)，则使能 CHx_O 和 MCHx_O 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 该位决定了 POEN 位是否可以由硬件自动设置。 0: POEN 位只能使用软件方式置 1 1: 根据自动输出功能模式选择(BRKAUMS), 当中止输入无效时, 硬件可在相应的计数器上溢或下溢时置位 POEN。 此位只有在 TIMEx_CCHP 寄存器的 PROT [1:0] =00 时才可修改。
13	BRKP	BREAK 输入信号极性 此位定义了中止输入信号 BREAK 的极性。 0: BREAK 中止输入低电平有效 1: BREAK 中止输入高电平有效 此位只有在 TIMEx_CCHP 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 中止事件。 0: 中止输入禁能 1: 中止输入使能 此位只有在 TIMEx_CCHP 寄存器的 PROT [1:0] =00 时才可修改。
11	ROS	运行模式下“关闭状态”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 21-5. 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 CHxNEN 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 CHxNEN 位被清零, 对应通道为输出“关闭状态”。 此位在 TIMEx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。
10	IOS	空闲模式下“关闭状态”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 21-5. 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 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 TIMEx_CCHP 寄存器的 PROT [1:0]=10 或 11 时不能被更改。
9:8	PROT[1:0]	互补寄存器保护控制 这两位定义了寄存器的写保护特性。 00: 禁能保护模式。无写保护。 01: PROT 模式 0。TIMEx_CTL1 寄存器中 ISOx/ISOxN 位, TIMEx_CCHP 寄存器中 BRKEN/ BRKP/ OAEN/ DTCFG 位写保护。

10: PROT 模式 1。除了 PROT 模式 0 下的寄存器写保护外, 还有

TIMERx_CHCTL2 寄存器中 CHxP/ MCHxP 位 (如果相应通道配置为输出模式),
TIMERx_CCHP 寄存器中 ROS/IOS 位。

11: PROT 模式 2。除了 PROT 模式 1 下的寄存器写保护外, 还有

TIMERx_CHCTL0/ 1 及 TIMERx_MCHCTL0/ 1 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_CCHP 寄存器被写入, 这两位被写
保护。

7:0 DTCFG[7:0]

死区时间控制

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

$DTCFG[7:5] = 3'b0xx: DTvalue = DTCFG[7:0] \times t_{DT}, t_{DT} = t_{DTS}$.

$DTCFG[7:5] = 3'b10x: DTvalue = (64 + DTCFG[5:0]) \times t_{DT}, t_{DT} = t_{DTS} \times 2$.

$DTCFG[7:5] = 3'b110: DTvalue = (32 + DTCFG[4:0]) \times t_{DT}, t_{DT} = t_{DTS} \times 8$.

$DTCFG[7:5] = 3'b111: DTvalue = (32 + DTCFG[4:0]) \times t_{DT}, t_{DT} = t_{DTS} \times 16$.

此位只有在 TIMERx_CCHP 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。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MCH1	MCH0	保留													
MS[2]	MS[2]														
rw		rw													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCH1CO MCEN	MCH1COMCTL[2:0]			MCH1CO MSEN	MCH1CO MFEN	MCH1MS[1:0]		MCH0CO MCEN	MCH0COMCTL[2:0]			MCH0CO MSEN	MCH0CO MFEN	MCH0MS[1:0]	
MCH1CAPFLT[3:0]				MCH1CAPPSC[1:0]				MCH0CAPFLT[3:0]			MCH0CAPPSC[1:0]				
rw				rw		rw		rw			rw			rw	

输出比较模式:

位/位域	名称	描述
31	MCH1MS[2]	多模式通道 1 I/O 模式选择 参考 MCH1MS[1:0]描述。
30	MCH0MS[2]	多模式通道 0 I/O 模式选择 参考 MCH0MS[1:0]描述。
29:16	保留	必须保持复位值。
15	MCH1COMCEN	多模式通道 1 输出比较清 0 使能 参见 MCH0COMCEN 描述。

14:12	MCH1COMCTL[2:0]	多模式通道 1 输出比较控制 参见 MCH0COMCTL 描述。
11	MCH1COMSEN	多模式通道 1 输出比较影子寄存器使能 参见 MCH0COMSEN 描述。
10	MCH1COMFEN	多模式通道1输出比较快速使能 参见MCH0COMSEN描述
9:8	MCH1MS[1:0]	多模式通道 1 I/O 模式选择 这些位定义了通道的方向和输入信号的选择。只有当通道关闭 (TIMERx_CHCTL2 寄存器的 MCH1EN 位清 0) 时这些位才可以写。 000: 多模式通道 1 配置为输出 001: 多模式通道1配置为输入, MIS1映射在MCI1FEM1上 010: 多模式通道1配置为输入, MIS1映射在MCI0FEM1上 011: 多模式通道 1 配置为输入, MIS1 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: 多模式通道1配置为输入, MIS1映射在CI1FEM1上。 101~111: 保留
7	MCH0COMCEN	多模式通道 0 输出比较清 0 使能 当此位被置 1, 当检测到 OCPRE 清除源信号输入高电平时, MO0CPRE 参考信号被清 0。OCPRE 清除源由 TIMERx_SMCFG 寄存器的 OCRC 位选择。 0: 多模式通道 0 输出比较清零禁止 1: 多模式通道 0 输出比较清零使能
6:4	MCH0COMCTL[2:0]	多模式通道 0 输出比较模式 当多模式通道 0 配置为输出模式, 并且 MCH0MSEL[1:0] = 2b'00, 此位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[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。 000: 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 001: 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。 010: 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。 011: 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。 100: 强制为低。强制 MO0CPRE 为低电平 101: 强制为高。强制 MO0CPRE 为高电平 110: 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 <code>TIMERx_MCH0CV</code> 时，MO0CPRE 为无效电平，否则为有效电平。 111：PWM 模式 1。在向上计数时，一旦计数器值小于 <code>TIMERx_MCH0CV</code> 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 <code>TIMERx_MCH0CV</code> 时，MO0CPRE 为有效电平，否则为无效电平。 如果配置在 PWM 模式下，只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时，MO0CPRE 电平才改变。 当 CH0 和 MCH0 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_CCHP</code> 寄存器的 <code>PROT[1:0]=11</code> 且 <code>MCH0MS = 000</code>（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，<code>TIMERx_MCH0CV</code> 寄存器的影子寄存器被使能，根据影子寄存器加载功能的相关配置在设置时刻进行更新。 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（<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>MCH0MS = 000</code> 时此位不能被改变。
2	MCH0COMFEN	多模式通道 0 输出比较快速使能 0：多模式通道 0 输出比较快速禁能。仅比较结果输入作为有效边沿时会产生比较匹配，并将 <code>MCH0_O</code> 设置为比较电平，激活 <code>MCH0_O</code> 输出的最小延时为 5 个时钟周期。 1：多模式通道 0 输出比较快速使能。触发输入信号的有效边沿和比较结果都会产生比较匹配，并将 <code>MCH0_O</code> 设置为比较电平。当触发信号的输入作为有效边沿时，激活 <code>MCH0_O</code> 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（<code>TIMERx_CHCTL2</code> 寄存器的 <code>MCH0EN</code> 位清 0）时，<code>MCH0MS[2:0]</code>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 <code>MCIOFEM0</code> 上 010：多模式通道 0 配置为输入，MIS0 映射在 <code>MCI1FEM0</code> 上 011：多模式通道 0 配置为输入，MIS0 映射在 <code>ITS</code> 上。此模式仅工作在内部触发输入被选中时（由 <code>SYSCFG_TIMERxCFG2(x = 0, 7)</code>寄存器中的 <code>TSCFG15[4:0]</code>位域选择）。 100：多模式通道0配置为输入，MIS0映射在<code>CI0FEM0</code>上。 101~111：保留

输入捕获模式：

位/位域	名称	描述
31	MCH1MS[2]	多模式通道 1 I/O 模式选择

		参考 MCH1MS[1:0]描述。
30	MCH0MS[2]	多模式通道 0 I/O 模式选择 参考 MCH0MS[1:0]描述。
29:16	保留	必须保持复位值。
15:12	MCH1CAPFLT[3:0]	多模式通道 1 输入捕获滤波控制 参见 MCH0CAPFLT 描述。
11:10	MCH1CAPPSC[1:0]	多模式通道 1 输入捕获预分频器 参见 MCH0CAPPSC 描述。
9:8	MCH1MS[1:0]	多模式通道 1 I/O 模式选择 与输出模式相同。
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 MCIO 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	MCH0CAPPSC[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 TIMERx_CHCTL2 寄存器中的 MCH0EN =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	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 1 (TIMERx_MCHCTL1)

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
MCH3MS [2]	MCH2MS [2]	保留													
rw	rw														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCH3CO MCEN	MCH3COMCTL[2:0]			MCH3CO MSEN	MCH3CO MFEN	MCH3MS[1:0]		MCH2CO MCEN	MCH2COMCTL[2:0]			MCH2CO MSEN	MCH2CO MFEN	MCH2MS[1:0]	
MCH3CAPFLT[3:0]				MCH3CAPPSC[1:0]				MCH2CAPFLT[3:0]			MCH2CAPPSC[1:0]				
rw		rw			rw			rw			rw			rw	

输出比较模式:

位/位域	名称	描述
31	MCH3MS[2]	多模式通道 1 I/O 模式选择 参考 MCH3MS[1:0]描述。
30	MCH2MS[2]	多模式通道 0 I/O 模式选择 参考 MCH2MS[1:0]描述。
29:16	保留	必须保持复位值。
15	MCH3COMCEN	多模式通道 3 输出比较清 0 使能 参见 MCH2COMCEN 描述。
14:12	MCH3COMCTL[2:0]	多模式通道 3 输出比较控制 参见 MCH2COMCTL 描述。
11	MCH3COMSEN	多模式通道 3 输出比较影子寄存器使能 参见 MCH2COMSEN 描述。
10	MCH3COMFEN	多模式通道3输出比较快速使能 参见 MCH2COMSEN 描述
9:8	MCH3MS[1:0]	多模式通道 3 I/O 模式选择 这些位定义了通道的方向和输入信号的选择。只有当通道关闭 (TIMERx_CHCTL2 寄存器的 MCH3EN 位清 0) 时, 这些位才可以写。 000: 多模式通道 3 配置为输出 001: 多模式通道 3 配置为输入, MIS3 映射在 MCI3FEM3 上 010: 多模式通道 3 配置为输入, MIS3 映射在 MCI2FEM3 上 011: 多模式通道 3 配置为输入, MIS3 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: 多模式通道3配置为输入, MIS3映射在CI3FEM3上。 101~111: 保留

7	MCH2COMCEN	多模式通道 2 输出比较清 0 使能 当此位被置 1，当检测到 OCPRE 清除源信号输入高电平时，MO2CPRE 参考信号被清 0。OCPRE 清除源由 TIMERx_SMCFG 寄存器的 OCRC 位选择。 0：多模式通道 2 输出比较清零禁止 1：多模式通道 2 输出比较清零使能
6:4	MCH2COMCTL[2:0]	多模式通道 2 输出比较模式 当多模式通道 2 配置为输出模式，并且 MCH2MSEL[1:0] = 2b'00，这些位定义了输出准备信号 MO2CPRE 的动作，而 MO2CPRE 决定了 MCH2_O 的值。MO2CPRE 高电平有效，而 MCH2_O 的有效电平取决于 MCH2FP[1:0]位。 注意：当多模式通道 2 配置为输出模式，且 MCH2MSEL[1:0] = 2b'11 时，CH2COMCTL[2:0]位定义了输出准备信号 O2CPRE 的动作，而 O2CPRE 决定了 CH2_O、MCH2_O 的值。O2CPRE 高电平有效，CH2_O、MCH2_O 的有效电平取决于 CH2P、MCH2P 位。 000：时基。输出比较寄存器 TIMERx_CHN2CV 与计数器 TIMERx_CNT 间的比较对 MO2CPRE 不起作用 001：匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_MCH2CV 相同时，强制 MO2CPRE 为高。 010：匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_MCH2CV 相同时，强制 MO2CPRE 为低。 011：匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_MCH2CV 相同时，强制 MO2CPRE 翻转。 100：强制为低。强制 MO2CPRE 为低电平 101：强制为高。强制 MO2CPRE 为高电平 110：PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_MCH2CV 时，MO2CPRE 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_MCH2CV 时，MO2CPRE 为无效电平，否则为有效电平。 111：PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_MCH2CV 时，MO2CPRE 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_MCH2CV 时，MO2CPRE 为有效电平，否则为无效电平。 如果配置在 PWM 模式下，只有当输出比较模式从时基模式变为 PWM 模式或者比较结果改变时，MO2CPRE 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_CCHP 寄存器的 PROT [1:0]=11 且 MCH2MS =000（比较模式）时此位不能被改变。
3	MCH2COMSEN	多模式通道 2 输出比较影子寄存器使能 当此位被置 1，TIMERx_MCH2CV 寄存器的影子寄存器被使能，根据影子寄存器加载功能的相关配置在设置时刻进行更新。 0：禁止多模式通道 2 输出/比较影子寄存器 1：使能多模式通道 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 `MCH2MS=000` 时此位不能被改变。

2	MCH2COMFEN	多模式通道 2 输出比较快速使能 0: 多模式通道 2 输出比较快速禁能。仅比较结果输入作为有效边沿时会产生比较匹配, 并将 <code>MCH2_O</code> 设置为比较电平, 激活 <code>MCH2_O</code> 输出的最小延时为 5 个时钟周期。 1: 多模式通道 2 输出比较快速使能。触发输入信号的有效边沿和比较结果都会产生比较匹配, 并将 <code>MCH2_O</code> 设置为比较电平。当触发信号的输入作为有效边沿时, 激活 <code>MCH2_O</code> 输出的最小延时为 3 个时钟周期。
1:0	MCH2MS[1:0]	多模式通道 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 (<code>TIMERx_CHCTL2</code> 寄存器的 <code>MCH2EN</code> 位清 0) 时, 这些位才可写。 000: 多模式通道 2 配置为输出 001: 多模式通道 2 配置为输入, <code>MIS2</code> 映射在 <code>MCI2FEM2</code> 上 010: 多模式通道 2 配置为输入, <code>MIS2</code> 映射在 <code>MCI3FE2M</code> 上 011: 多模式通道 2 配置为输入, <code>MIS2</code> 映射在 <code>ITS</code> 上。此模式仅工作在内部触发输入被选中时 (由 <code>SYSCFG_TIMERxCFG2(x=0,7)</code> 寄存器中的 <code>TSCFG15[4:0]</code> 位域选择)。 100: 多模式通道 2 配置为输入, <code>MIS3</code> 映射在 <code>CI2FEM2</code> 上。 101~111: 保留

输入捕获模式:

位/位域	名称	描述
31	MCH3MS[2]	多模式通道 1 I/O 模式选择 参考 <code>MCH3MS[1:0]</code> 描述。
30	MCH2MS[2]	多模式通道 0 I/O 模式选择 参考 <code>MCH2MS[1:0]</code> 描述。
29:16	保留	必须保持复位值。
15:12	MCH3CAPFLT[3:0]	多模式通道 3 输入捕获滤波控制 参见 <code>MCH2CAPFLT</code> 描述。
11:10	MCH3CAPPSC[1:0]	多模式通道 3 输入捕获预分频器 参见 <code>MCH2CAPPSC</code> 描述。
9:8	MCH3MS[1:0]	多模式通道 3 I/O 模式选择 与输出模式相同。
7:4	MCH2CAPFLT[3:0]	多模式通道 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 <code>N</code> 个输入事件后会产生一个输出的跳变。这些位定义了 <code>MCI2</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	MCH2CAPPSC[1:0]	多模式通道 2 输入捕获预分频器 这 2 位定义了多模式通道 2 输入的预分频系数。当 <code>TIMERx_CHCTL2</code> 寄存器中的 <code>MCH2EN=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	MCH2MS[1:0]	多模式通道 2 I/O 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。

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								MCH3FP[1:0]		MCH2FP[1:0]		MCH1FP[1:0]		MCH0FP[1:0]	
								rw		rw		rw		rw	

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:6	MCH3FP[1:0]	多模式通道 3 捕获 / 比较独立极性控制 参考 MCH0FP[1:0]描述。
5:4	MCH2FP[1:0]	多模式通道 2 捕获 / 比较独立极性控制

参考 MCH0FP[1:0]描述。

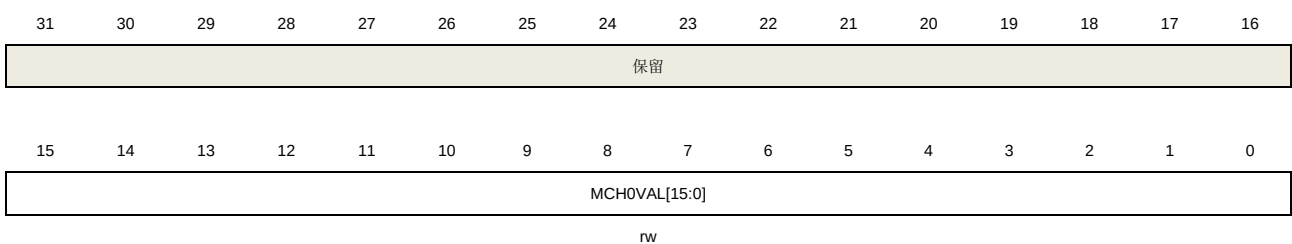
3:2	MCH1FP[1:0]	多模式通道 1 捕获 / 比较独立极性控制 参考 MCH0FP[1:0]描述。
1:0	MCH0FP[1:0]	多模式通道 0 捕获 / 比较独立极性控制 当多模式通道 0 配置为输出模式时，且 MCH0MSEL[1:0] = 2'b00，此位定义了输出信号极性。 00：多模式通道 0 高电平有效 01：多模式通道 0 低电平有效 10：保留 11：保留 当通道 0 配置为输入模式时，此位定义了多模式通道 0 输入信号的极性。 MCH0FP[1:0]将选择多模式通道 0 输入信号的有效边沿或者捕获极性。 00：把多模式通道 0 输入信号的上升沿作为捕获或者从模式下触发的有效信号，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_CCHP 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。



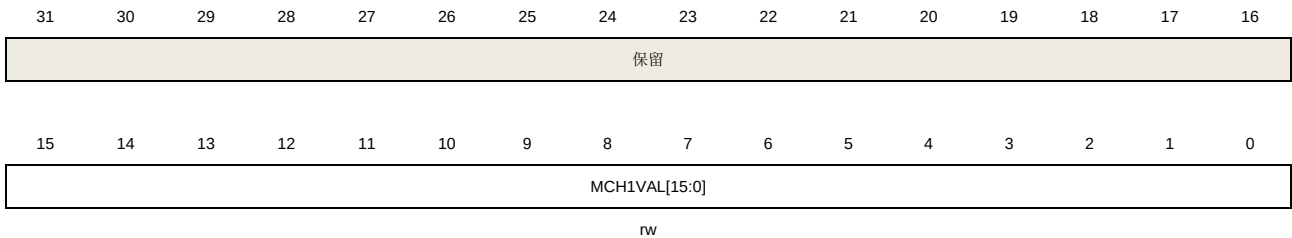
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	MCH0VAL[15:0]	多模式通道 0 的捕获或比较值 当多模式通道 0 配置为输入模式时，这些位决定了上次捕获事件的计数器值，且本寄存器为只读。 当多模式通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器根据影子寄存器加载功能的相关配置在设定的时刻更新。

多模式通道 1 捕获 / 比较寄存器 (TIMERx_MCH1CV)

地址偏移: 0x58

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



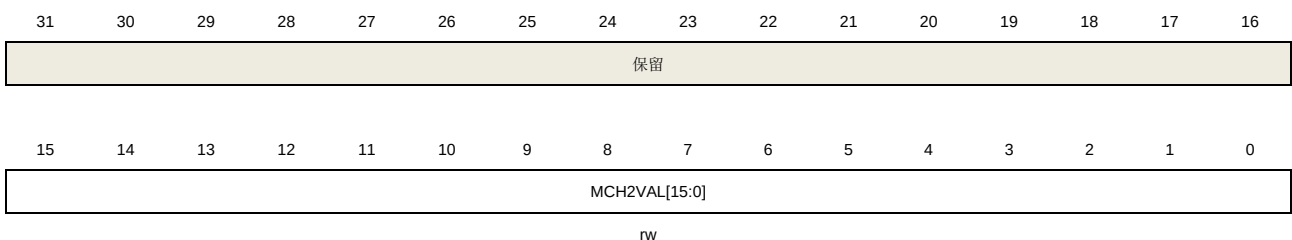
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	MCH1VAL[15:0]	多模式通道 1 的捕获或比较值 当多模式通道 1 配置为输入模式时, 这些位决定了上次捕获事件的计数器值, 且本寄存器为只读。 当多模式通道 1 配置为输出模式时, 这些位包含了即将和计数器比较的值。使能相应影子寄存器后, 影子寄存器根据影子寄存器加载功能的相关配置在设定的时刻更新。

多模式通道 2 捕获 / 比较寄存器 (TIMERx_MCH2CV)

地址偏移: 0x5C

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



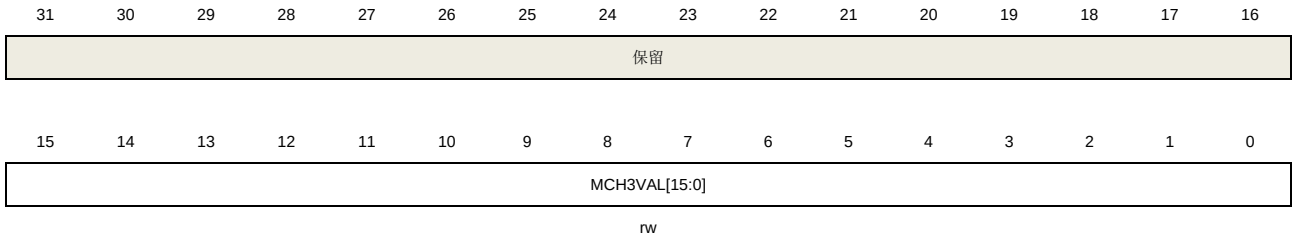
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	MCH2VAL[15:0]	多模式通道 2 的捕获或比较值 当多模式通道 2 配置为输入模式时, 这些位决定了上次捕获事件的计数器值, 且本寄存器为只读。 当多模式通道 2 配置为输出模式时, 这些位包含了即将和计数器比较的值。使能相应影子寄存器后, 影子寄存器根据影子寄存器加载功能的相关配置在设定的时刻更新。

多模式通道 3 捕获 / 比较寄存器 (TIMERx_MCH3CV)

地址偏移: 0x60

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



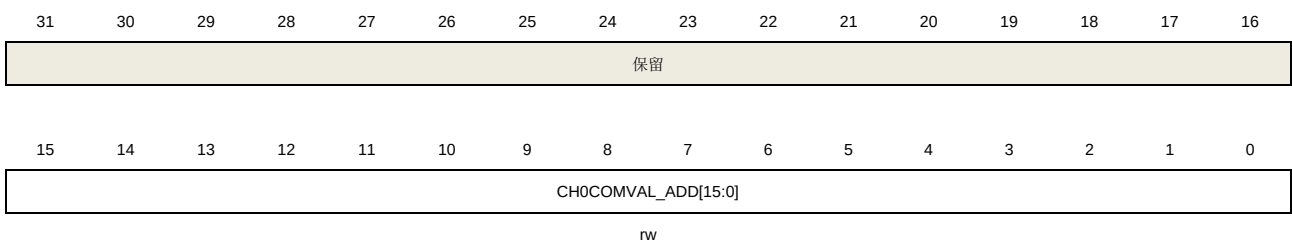
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	MCH3VAL[15:0]	多模式通道 3 的捕获或比较值 当多模式通道 3 配置为输入模式时, 这些位决定了上次捕获事件的计数器值, 且本寄存器为只读。 当多模式通道 3 配置为输出模式时, 这些位包含了即将和计数器比较的值。使能相应影子寄存器后, 影子寄存器根据影子寄存器加载功能的相关配置在设定的时刻更新。

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。



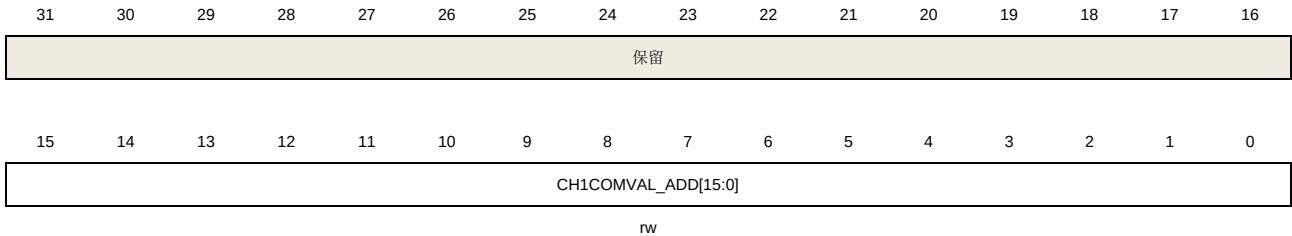
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH0COMVAL_ADD [15:0]	通道0附加比较值 当通道0配置为输出模式时, 这些位包含了即将和计数器比较的值。使能相应影子寄存器后, 影子寄存器根据影子寄存器加载功能的相关配置在设定的时刻更新。 注意: 该寄存器仅用于复合PWM模式和非对称模式 (当CH0COMCTL[3:2] = 2'b10和CH0MS[2:0] = 3'b000时)。

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。



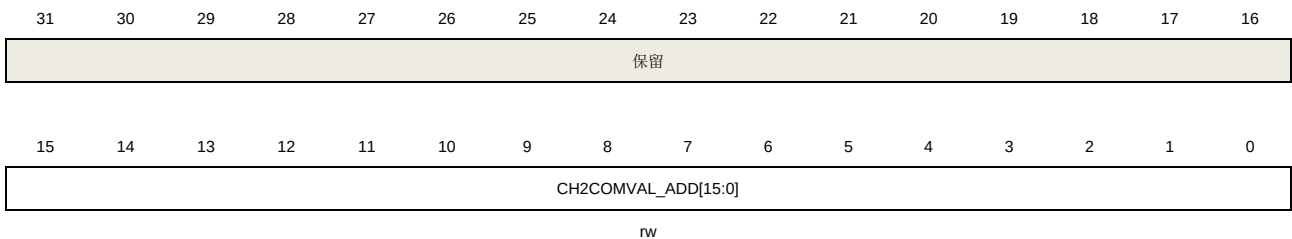
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH1COMVAL_ADD [15:0]	通道1附加比较值 当通道1附加配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器根据影子寄存器加载功能的相关配置在设定的时刻更新。 注意：该寄存器仅用于复合PWM模式和非对称模式（当CH1COMCTL[3:2] = 2'b10和CH1MS[2:0] = 3'b000时）。

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。



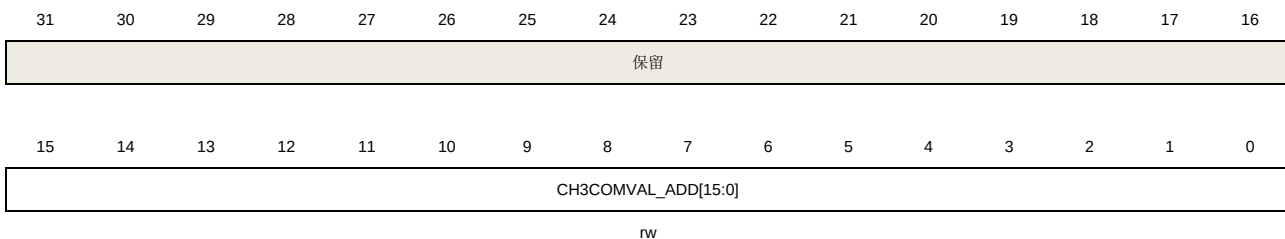
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH2COMVAL_ADD [15:0]	通道2附加比较值 当通道2附加配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器根据影子寄存器加载功能的相关配置在设定的时刻更新。 注意：该寄存器仅用于复合PWM模式和非对称模式（当CH2COMCTL[3:2] = 2'b10和CH2MS[2:0] = 3'b000时）。

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。



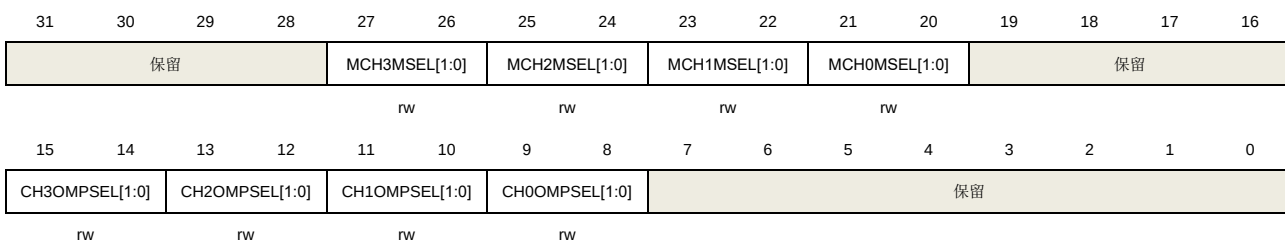
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH3COMVAL_ADD [15:0]	通道3附加比较值 当通道3附加配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器根据影子寄存器加载功能的相关配置在设定的时刻更新。 注意： 该寄存器仅用于复合PWM模式和非对称模式（当CH3COMCTL[3:2] = 2'b10 和 CH3MS[2:0] = 3'b000时）。

控制寄存器 2 (TIMERx_CTL2)

地址偏移: 0x74

复位值: 0x0FF0 0000

该寄存器只能按字 (32 位) 访问。



位/位域	名称	描述
31:28	保留	必须保持复位值。
27:26	MCH3MSEL[1:0]	多模式通道 3 模式选择 00: 独立模式，MCH3 独立于 CH3 01: 保留 10: 保留 11: 互补模式，只有 CH3 可用于输入，MCH3 输出与 CH3 输出互补
25:24	MMCH2SEL[1:0]	多模式通道 2 模式选择 00: 独立模式，MCH2 独立于 CH2

		01: 保留
		10: 保留
		11: 互补模式，只有 CH2 可用于输入，MCH2 输出与 CH2 输出互补
23:22	MCH1MSEL[1:0]	多模式通道 1 模式选择 00: 独立模式，MCH1 输出独立于 CH1 输出 01: 保留 10: 保留 11: 互补模式，只有 CH1 可用于输入，MCH1 输出与 CH1 输出互补
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
19:16	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[3:0]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[3:0]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[3:0]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[3:0]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。

备用功能控制寄存器 (TIMERx_AFCTL)

地址偏移: 0x8C

复位值: 0x0000 0007

该寄存器只能按字 (32 位) 访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
保留													OCRINSEL[2:0]			
rw																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
保留			BRKCOMP	BRKCOMP	BRKCOMP	BRKCOMP	BRKIN0P	保留				BRKCOMP	BRKCOMP	BRKCOMP	BRKCOMP	BRKIN0E
			3P	2P	1P	0P						3EN	2EN	1EN	0EN	N
			rw	rw	rw	rw	rw					rw	rw	rw	rw	rw

位/位域	名称	描述
31:19	保留	必须保持复位值。
18:16	OCRINSEL[2:0]	OCPRE_CLR输入选择 000: OCPRE_CLR0(CMP0 output) 001: OCPRE_CLR1(CMP1 output) 010: OCPRE_CLR2(CMP2 output) 011: OCPRE_CLR3(CMP3 output) 其它位保留。 此位只有在TIMERx_CCHP寄存器的PROT [1:0] = 00时才可修改。
15:13	保留	必须保持复位值。
12	BRKCOMP3P	BREAK CMP3输入极性 参考BRKCOMP0P描述。

11	BRKCMP2P	BREAK CMP2输入极性 参考BRKCMP0P描述。
10	BRKCMP1P	BREAK CMP1输入极性 参考BRKCMP0P描述。
9	BRKCMP0P	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: CMP0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_CCHP寄存器的PROT [1:0] = 00时才可修改。
8	BRKIN0P	BRKIN0输入信号极性 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_CCHP寄存器的PROT [1:0] = 00时才可修改。
7:5	保留	必须保持复位值。
4	BRKCMP3EN	BREAK CMP3输入使能 参考BRKCMP0EN。
3	BRKCMP2EN	BREAK CMP2输入使能 参考BRKCMP0EN。
2	BRKCMP1EN	BREAK CMP1输入使能 参考BRKCMP0EN。
1	BRKCMP0EN	BREAK CMP0输入使能 0: CMP0输入禁能 1: CMP0输入使能 此位只有在TIMERx_CCHP寄存器的PROT [1:0] = 00时才可修改。
0	BRKIN0EN	BREAK0输入信号使能 0: BRKIN0输入禁能 1: BRKIN0输入使能 此位只有在TIMERx_CCHP寄存器的PROT [1:0] = 00时才可修改。

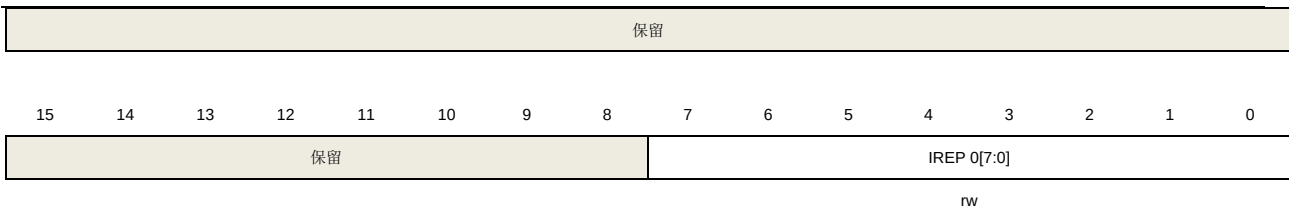
中断重复计数寄存器 (TIMERx_IREP)

地址偏移: 0x90

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



位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	IREP[7:0]	中断重复计数器的值 该位域指定了上溢或下溢中断产生速率。每次重复计数器计数到 0 时，都会产生一个溢出中断。

ADC 转换请求控制寄存器 (TIMERx_ADCRCTL)

地址偏移: 0x94

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



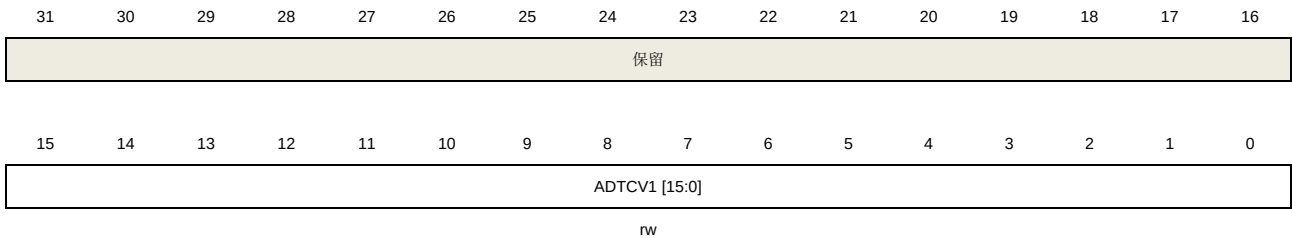
位/位域	名称	描述
31:8	保留	必须保持复位值。
7	ADT1UPGTEN	ADTCV1[15:0]比较匹配 (仅向上计数) 生成触发使能 0: 禁能 1: 使能
6	ADT1DWGTEN	ADTCV1[15:0]比较匹配 (仅向下计数) 生成触发使能 0: 禁能 1: 使能
5	ADT2UPGTEN	ADTCV2[15:0]比较匹配 (仅向上计数) 生成触发使能 0: 禁能 1: 使能
4	ADT2DWGTEN	ADTCV2[15:0]比较匹配 (仅向下计数) 生成触发使能 0: 禁能 1: 使能
3:0	保留	必须保持复位值。

ADC 触发比较值寄存器 1 (TIMERx_ADCCR1)

地址偏移: 0x98

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



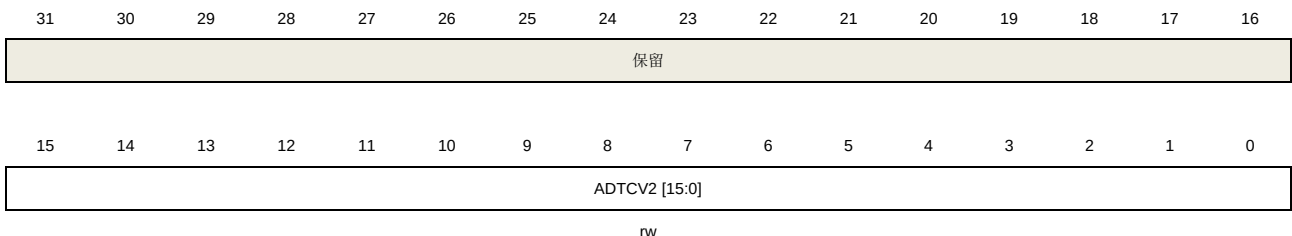
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	ADTCV1 [15:0]	ADC触发比较值1 这些位包含了与计数器比较的值。使能相应影子寄存器后，影子寄存器根据影子寄存器加载功能的相关配置在设定的时刻更新。

ADC 触发比较值寄存器 2 (TIMERx_ADCCR2)

地址偏移: 0x9C

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	ADTCV2 [15:0]	ADC触发比较值2 这些位包含了与计数器比较的值。使能相应影子寄存器后，影子寄存器根据影子寄存器加载功能的相关配置在设定的时刻更新。

ADC 重复计数器寄存器 (TIMERx_ADCREP)

地址偏移: 0xA0

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				ADCREPAB [3:0]				ADCREPB [3:0]				ADCREPA [3:0]			
rw				rw				rw				rw			

位/位域	名称	描述
31:12	保留	必须保持复位值。
11:8	ADCREPAB[3:0]	ADC TIMEx_TRGAB 重复计数值设置 这些比特位指定了 TIMEx_TRGAB 重复计数，范围从 0 到 15。 在比较触发器生成TIMEx_TRGAB时，计数器从这些比特位计数值开始递减，直到计数器为0时，才会生成TIMEx_TRGAB事件。
7:4	ADCREPB[3:0]	ADC TIMEx_TRGB 重复计数值设置 这些比特位指定了 TIMEx_TRGB 重复计数，范围从 0 到 15。 在比较触发器生成TIMEx_TRGB时，计数器从这些比特位计数值开始递减，直到计数器为0时，才会生成TIMEx_TRGB事件。
3:0	ADCREPA[3:0]	ADC TIMEx_TRGA 重复计数值设置 这些比特位指定了 TIMEx_TRGA 重复计数，范围从 0 到 15。 在比较触发器生成TIMEx_TRGA时，计数器从这些比特位计数值开始递减，直到计数器为0时，才会生成TIMEx_TRGA事件。

ADC 触发控制寄存器（TIMEx_ADCTL）

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADREPR LD	保留					ADTPRE EN2	ADTPRE EN1	ADTSOF	ADTSUF	保留				ADCREP BMD	ADCREP AMD
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	ADREPRLD	ADC重复计数器立即重新加载 当该比特位置为1时，TIMEx_TRGA重复计数器、TIMEx_TRGB重复计数器和TIMEx_TRGAB重复计数器将重新加载各自的重复计数设置值。

在重新加载完成后，该比特位由硬件自动清零。

14:10	保留	必须保持复位值。
9	ADTPREEN2	ADTCV2[15:0]预装载寄存器使能 0: 禁能ADTCV2[15:0]影子寄存器 1: 使能ADTCV2[15:0]影子寄存器 注意： 该位和影子寄存器加载功能相链接。
8	ADTPREEN1	ADTCV1[15:0]预装载寄存器使能 0: 禁能ADTCV1[15:0]影子寄存器 1: 使能ADTCV1[15:0]影子寄存器 注意： 该位和影子寄存器加载功能相链接。
7	ADTSOF	当计数器上溢时，ADC触发信号TIMERx_TRGOF使能 0: TIMERx_TRGOF触发信号禁能 1: TIMERx_TRGOF触发信号使能
6	ADTSUF	当计数器下溢时，ADC触发信号TIMERx_TRGUF使能 0: TIMERx_TRGUF触发信号禁能 1: TIMERx_TRGUF触发信号使能
5:2	保留	必须保持复位值。
1	ADCREPBMD	ADC TIMERx_TRGB重复计数器递减模式设置。 0: 定时器下溢时，TIMERx_TRGB重复计数器从ADCREPB开始递减，直到重复计数器计数为0，触发信号TIMERx_TRGB发生。 1: 定时器上溢时，TIMERx_TRGB重复计数器从ADCREPB开始递减，直到重复计数器计数为0，触发信号TIMERx_TRGB发生。
0	ADCREPAMD	ADC TIMERx_TRGA重复计数器递减模式设置。 0: 定时器下溢时，TIMERx_TRGA重复计数器从ADCREPA开始递减，直到重复计数器计数为0，触发信号TIMERx_TRGA发生。 1: 定时器上溢时，TIMERx_TRGA重复计数器从ADCREPA开始递减，直到重复计数器计数为0，触发信号TIMERx_TRGA发生。

影子更新模式寄存器 0 (TIMERx_SHUPM0)

地址偏移: 0xC4

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
CARUPSEL[1:0]		ADCCR2UPSEL[1:0]		ADCCR1UPSEL[1:0]		保留	CH3COMV_ADDUPSEL[2:0]		CH3CVUPSEL[2:0]		CH2COMV_ADDUPSEL[2:0]				
rw		rw		rw					rw		rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	CH2CVUPSEL[2:0]		CH1COMV_ADDUPSEL[2:0]		CH1CVUPSEL[2:0]		CH0COMV_ADDUPSEL[2:0]		CH0CVUPSEL[2:0]						
	rw		rw		rw		rw		rw		rw				

位/位域	名称	描述
31:30	CARUPSEL[1:0]	计数器自动重装载值更新模式选择，用于设置自动重装载值何时自动重载。 00：全局更新功能使能。 01：自动重装载值在下一个计数器上溢、UPG = 1 或主从模式(restart, restart + event)更新比较值。 10：自动重装载值在下一个计数器下溢，UPG = 1 或主从模式(restart, restart + event) 更新比较值。 11：自动重装载值在下一个计数器上溢或下溢，UPG = 1 或主从模式(restart, restart + event) 更新比较值。 注意： 1.在向上计数模式下，比较值在计数器上溢时更新，而在向下计数模式下，比较值在计数器下溢时更新； 2.这些比特位由 TIMERx_CTL0 寄存器中的 UPDIS 位控制； 3.全局更新功能是指统一在更新事件进行更新。
29:28	ADCCR2UPSEL[1:0]	ADC 触发比较值 2 更新模式选择，用于设置触发比较值何时自动重载。 参考 ADCCR1UPSEL[1:0]描述。
27:26	ADCCR1UPSEL[1:0]	ADC 触发比较值 1 更新模式选择，用于设置触发比较值何时自动重载。 00：全局更新功能使能。 01：触发比较值在下一个计数器上溢、UPG = 1 或主从模式(restart, restart + event)更新比较值。 10：触发比较值在下一个计数器下溢，UPG = 1 或主从模式(restart, restart + event) 更新比较值。 11：触发比较值在下一个计数器上溢或下溢，UPG = 1 或主从模式(restart, restart + event) 更新比较值。 注意： 1.在向上计数模式下，比较值在计数器上溢时更新，而在向下计数模式下，比较值在计数器下溢时更新； 2.这些比特位由 TIMERx_CTL0 寄存器中的 UPDIS 位控制； 3.全局更新功能是指统一在更新事件进行更新。
25	保留	必须保持复位值。
24:22	CH3COMV_ADDUPSEL[2:0]	通道 3 附加比较值更新模式选择，用于设置比较值何时自动重载。 参考 CH0COMV_ADDUPSEL[2:0]描述。
21:19	CH3CVUPSEL[2:0]	通道 3 捕获/比较值更新模式选择，用于设置比较值何时自动重载。 参考 CH0CVUPSEL[2:0]描述。
18:16	CH2COMV_ADDUPSEL[2:0]	通道 2 附加比较值更新模式选择，用于设置比较值何时自动重载。 参考 CH0COMV_ADDUPSEL[2:0]描述。
15	保留	必须保持复位值。
14:12	CH2CVUPSEL[2:0]	通道 2 捕获/比较值更新模式选择，用于设置比较值何时自动重载。 参考 CH0CVUPSEL[2:0]描述。
11:9	CH1COMV_ADDUP	通道 1 附加比较值更新模式选择，用于设置比较值何时自动重载。

	SEL[2:0]	参考 CH0COMV_ADDUPSEL[2:0]描述。
8:6	CH1CVUPSEL[2:0]	通道 1 捕获/比较值更新模式选择，用于设置比较值何时自动重载。 参考 CH0CVUPSEL[2:0]描述。
5:3	CH0COMV_ADDUPSEL[2:0]	通道 0 附加比较值更新模式选择，用于设置比较值何时自动重载。 000：全局更新功能使能。 001：附加比较值在下一个计数器上溢、UPG = 1 或主从模式(restart, restart + event)更新比较值。 010：附加比较值在下一个计数器下溢，UPG = 1 或主从模式(restart, restart + event) 更新比较值。 011：附加比较值在下一个计数器上溢或下溢，UPG = 1 或主从模式(restart, restart + event) 更新比较值。 100：附加比较值在下一个计数周期和该通道比较值匹配时刻更新。 注意： 1.在向上计数模式下，比较值在计数器上溢时更新，而在向下计数模式下，比较值在计数器下溢时更新； 2.这些比特位由 TIMERx_CTL0 寄存器中的 UPDIS 位控制； 3.全局更新功能是指统一在更新事件进行更新。
2:0	CH0CVUPSEL[2:0]	通道 0 捕获/比较值更新模式选择，用于设置比较值何时自动重载。 000：全局更新功能使能。 001：比较值在下一个计数器上溢、UPG = 1 或主从模式(restart, restart + event)更新比较值。 010：比较值在下一个计数器下溢，UPG = 1 或主从模式(restart, restart + event) 更新比较值。 011：比较值在下一个计数器上溢或下溢，UPG = 1 或主从模式(restart, restart + event) 更新比较值。 注意： 1.在向上计数模式下，比较值在计数器上溢时更新，而在向下计数模式下，比较值在计数器下溢时更新； 2.这些比特位由 TIMERx_CTL0 寄存器中的 UPDIS 位控制； 3.全局更新功能是指统一在更新事件进行更新。

影子更新模式寄存器 1 (TIMERx_SHUPM1)

地址偏移：0xC8

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								MCH3CVUPSEL[1:0]	MCH2CVUPSEL[1:0]	MCH1CVUPSEL[1:0]	MCH0CVUPSEL[1:0]				
								r/w	r/w	r/w	r/w				

位/位域	名称	描述
31:8	保留	必须保持复位值。
7:6	MCH3CVUPSEL[1:0]	多模式通道 3 捕获/比较值更新模式选择，用于设置比较值何时自动重载。 参考 MCH0CVUPSEL[1:0]描述。
5:4	MCH2CVUPSEL[1:0]	多模式通道 2 捕获/比较值更新模式选择，用于设置比较值何时自动重载。 参考 MCH0CVUPSEL[1:0]描述。
3:2	MCH1CVUPSEL[1:0]	多模式通道 1 捕获/比较值更新模式选择，用于设置比较值何时自动重载。 参考 MCH0CVUPSEL[1:0]描述。
1:0	MCH0CVUPSEL[1:0]	多模式通道 0 捕获/比较值更新模式选择，用于设置比较值何时自动重载。 00：全局更新功能使能。 01：比较值在下一个计数器上溢、UPG = 1 或主从模式(restart, restart + event)更新比较值。 10：比较值在下一个计数器下溢，UPG = 1 或主从模式(restart, restart + event)更新比较值。 11：比较值在下一个计数器上溢或下溢，UPG = 1 或主从模式(restart, restart + event)更新比较值。 注意： 1.在向上计数模式下，比较值在计数器上溢时更新，而在向下计数模式下，比较值在计数器下溢时更新。 2.这些比特位由 TIMERx_CTL0 寄存器中的 UPDIS 位控制。 3.全局更新功能是指统一在更新事件进行更新。

计数器上溢或下溢模式选择寄存器 (TIMERx_COUNSEL)

地址偏移：0xCC

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												BRKAUMS[1:0]		保留	
rw															

位/位域	名称	描述
31:4	保留	必须保持复位值。
3:2	BRKAUMS[1:0]	Break 自动输出功能模式选择。如果 break 输入无效，则可以设置 poen。 00：POEN 在下一个计数器上溢或下溢时置位 01：POEN 在下一个计数器上溢时置位 10：POEN 在下一个计数器下溢时置位 11：POEN 在下次更新时置位

只有当 `TIMERx_CCHP` 寄存器中的 `PROT[1:0]` 位为 `00` 时，该比特位才能修改。

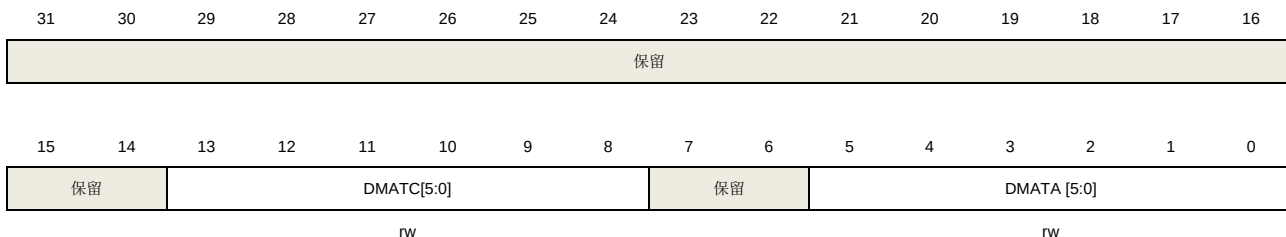
1:0 保留 必须保持复位值。

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。



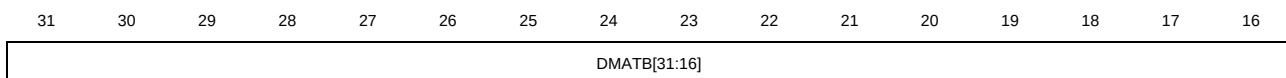
位/位域	名称	描述
31:14	保留	必须保持复位值。
13:8	DMATC [5:0]	DMA 传输计数 该位域定义了 DMA 访问 (读/写) <code>TIMERx_DMATB</code> 寄存器的次数。 6'b000000: 传输1次 6'b000001: 传输2次 ... 6'b110000: 传输 52 次
7:6	保留	必须保持复位值。
5:0	DMATA [5:0]	DMA 传输起始地址 该位域定义了 DMA 访问 <code>TIMERx_DMATB</code> 寄存器的第一个地址。当通过 <code>TIMERx_DMA</code> 第一次访问时，访问的就是该位域指定的地址。第二次访问 <code>TIMERx_DMATB</code> 时，将访问起始地址+0x4。 6'b0_0000: <code>TIMERx_CTL0</code> 6'b0_0001: <code>TIMERx_CTL1</code> ... 6'b110011: <code>TIMERx_COUNSEL</code> 总之: 起始地址 = <code>TIMERx_CTL0 + DMATA*4</code>

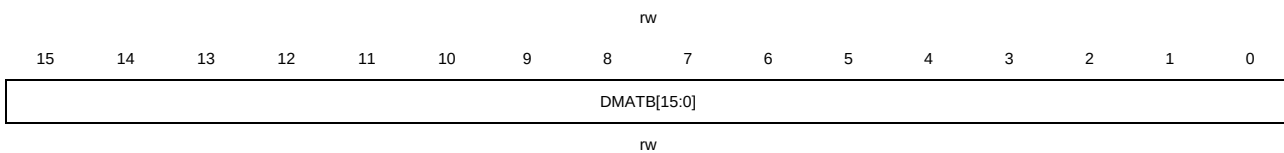
DMA 发送缓冲区寄存器 (TIMERx_DMATB)

地址偏移: `0xE4`

复位值: `0x0000 0000`

该寄存器只能按字 (32 位) 访问。





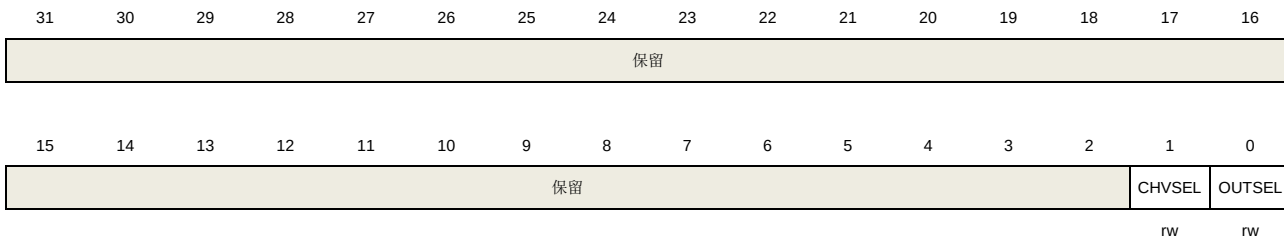
位/位域	名称	描述
31:0	DMATB [31:0]	DMA 发送缓冲 对这个寄存器的读或写，（起始地址+传输次数*4）地址范围内的寄存器会被访问 传输次数由硬件计算，范围为 0 到 DMATC。

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	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

21.2. 通用定时器 L0 (TIMERx, x = 1, 2)

21.2.1. 简介

通用定时器 L0 (TIMER1 / 2) 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。

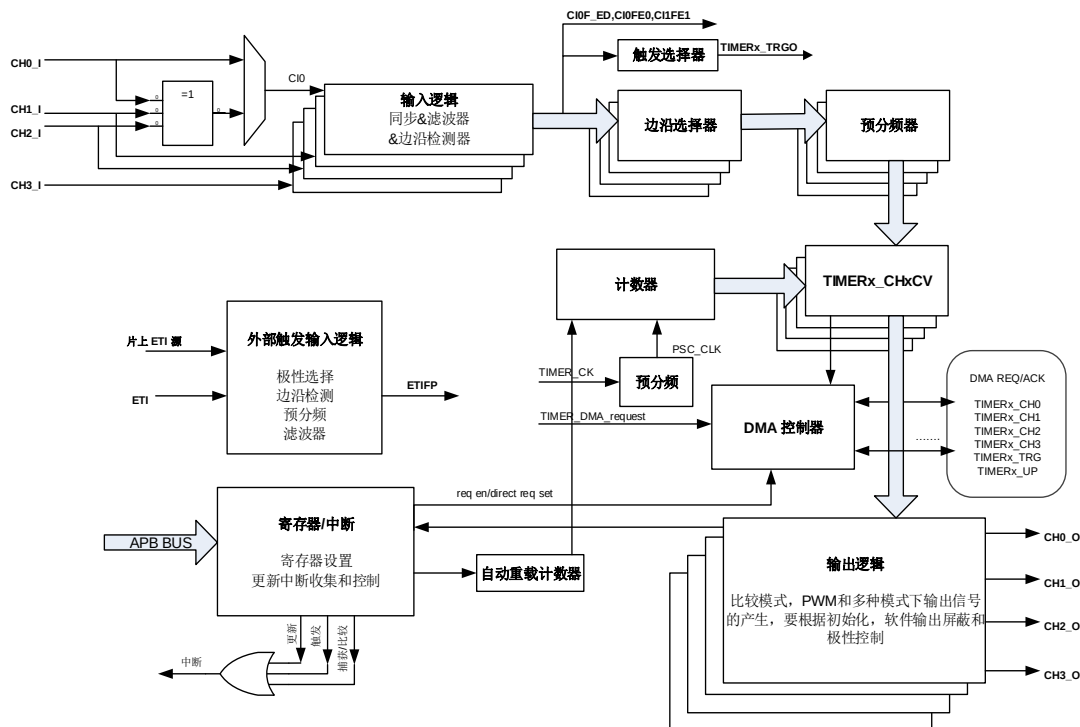
21.2.2. 主要特征

- 总通道数：4；
- 计数器宽度：16位；
- 时钟源可选：内部时钟，内部触发，外部输入，外部触发；
- 多种计数模式：向上计数，向下计数和中央计数；
- 正交译码器接口：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。

21.2.3. 结构框图

[图 21-55. 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 21-55. 通用定时器 L0 结构框图



21.2.4. 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, 2)寄存器中的 TSCFGy[4:0] (y=0..7,9..12,15)位域控制的复用时钟源驱动。

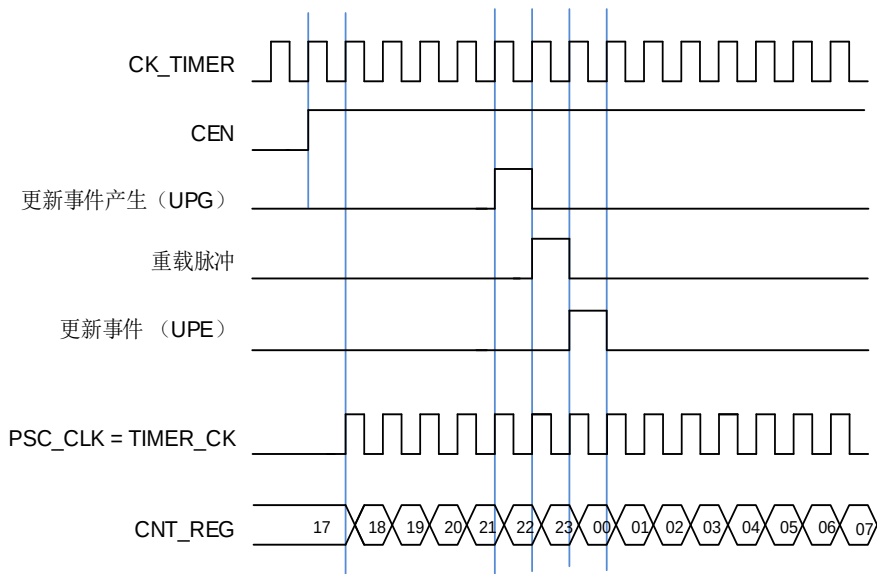
- 当SYSCFG_TIMERxCFG(x = 0, 7)寄存器中的TSCFGy[4:0] = 5'b000000(y=0..7,9..12,15)时，定时器选择内部时钟源（连接到RCU模块的CK_TIMER）

如果 SYSCFG_TIMERxCFG(x = 1, 2)寄存器中的 TSCFGy[4:0] = 5'b000000(y=0..7,9..12,15)，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。

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 = 1, 2)寄存器中的 TSCFGy[4:0] (y=0..2,6,9..12)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, 7)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 21-56. 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` ($n=0, 1$) 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 0x9~0xE 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

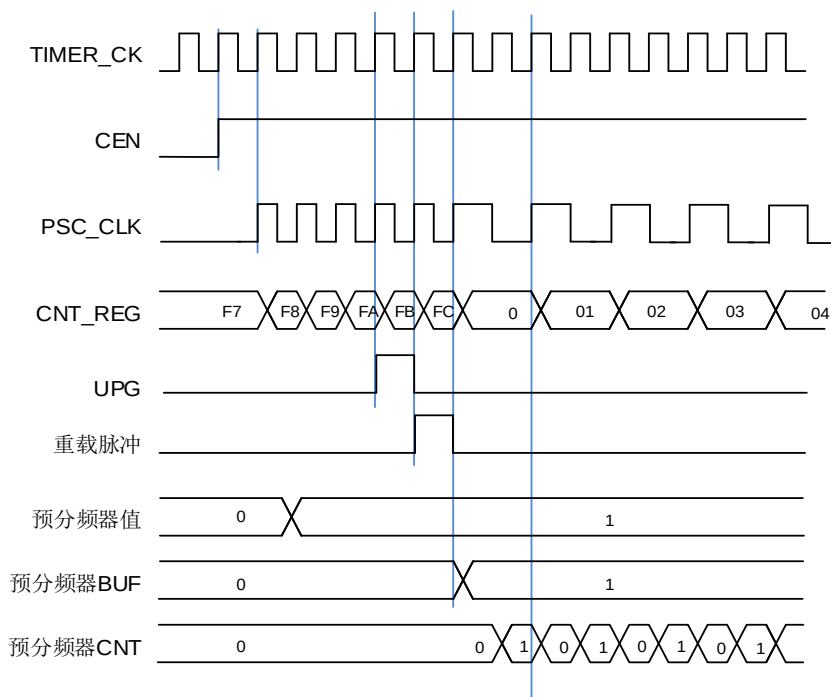
计数器预分频器可以在外部引脚 `ETI` 的每个上升沿或下降沿计数。这种模式可以通过设置 `TIMERx_SMCFG` 寄存器中的 `SMC1` 位为 1 来选择。另一种选择 `ETI` 信号作为时钟源方式是，设置 TSCFG6[4:0] 为 0x8。注意 `ETI` 信号是通过数字滤波器采样 `ETI` 引脚得到的。如果选择 `ETI` 信号为时钟源，触发控制器包括边沿监测电路将在每个 `ETI` 信号上升沿产生一个时钟脉冲来为计数器预分频器提供时钟。

注意： `ETI` 信号可以从外部 `ETI` 引脚输入，也可由片上外设提供，可以参考 `TIMERx_CTL0` 寄存器的 `ETISEL[1:0]` 位域。

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次更新事件到来时被采用。

图 21-57. 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。

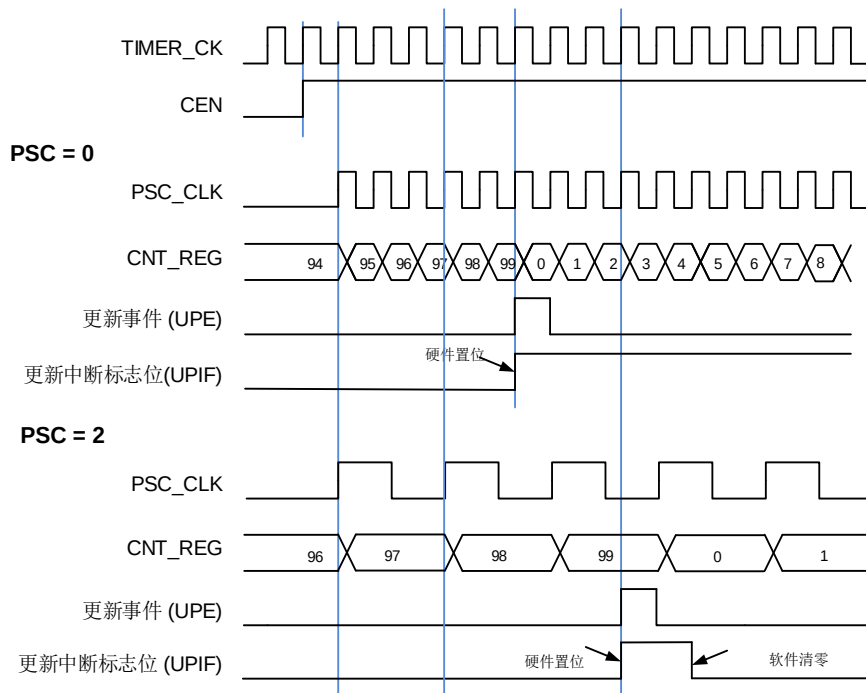
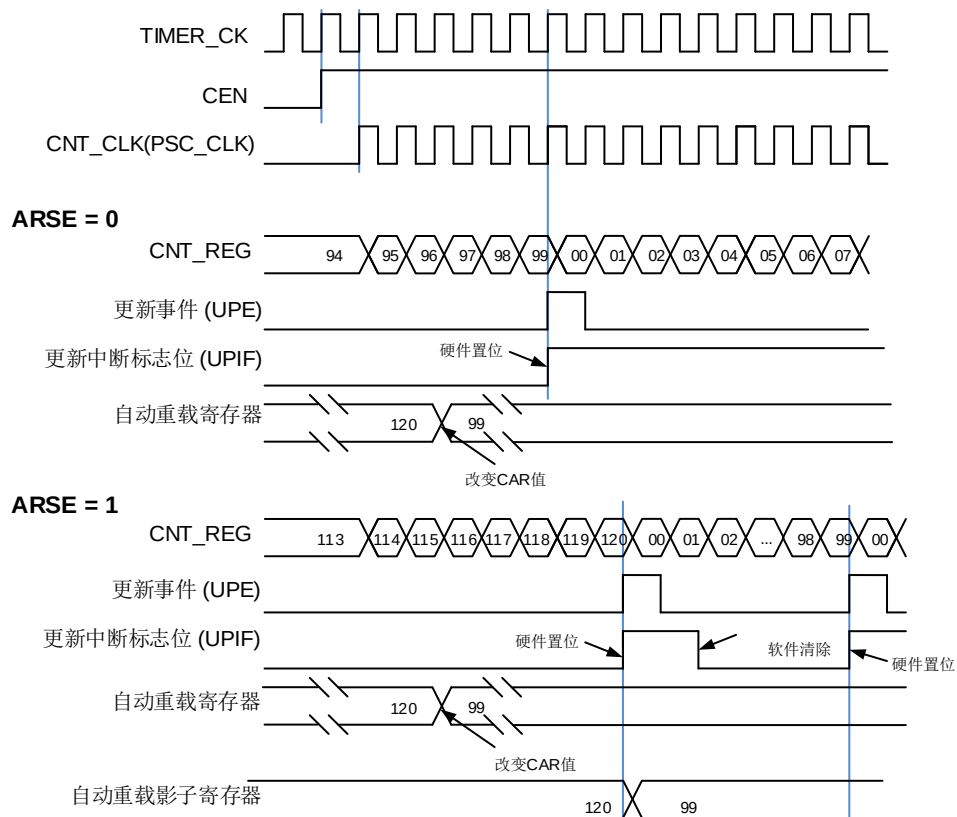
当通过 `TIMERx_SWEVG` 寄存器的 `UPG` 位置 1 来设置更新事件时，计数值会被清 0，并产生更新事件。

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 21-58. 向上计数时序图，PSC=0 / 2](#) 和 [图 21-59. 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 21-58. 向上计数时序图, PSC=0 / 2

图 21-59. 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 21-60. 向下计数时序图, `PSC=0 / 2`](#) 和 [图 21-61. 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 21-60. 向下计数时序图, `PSC=0 / 2`

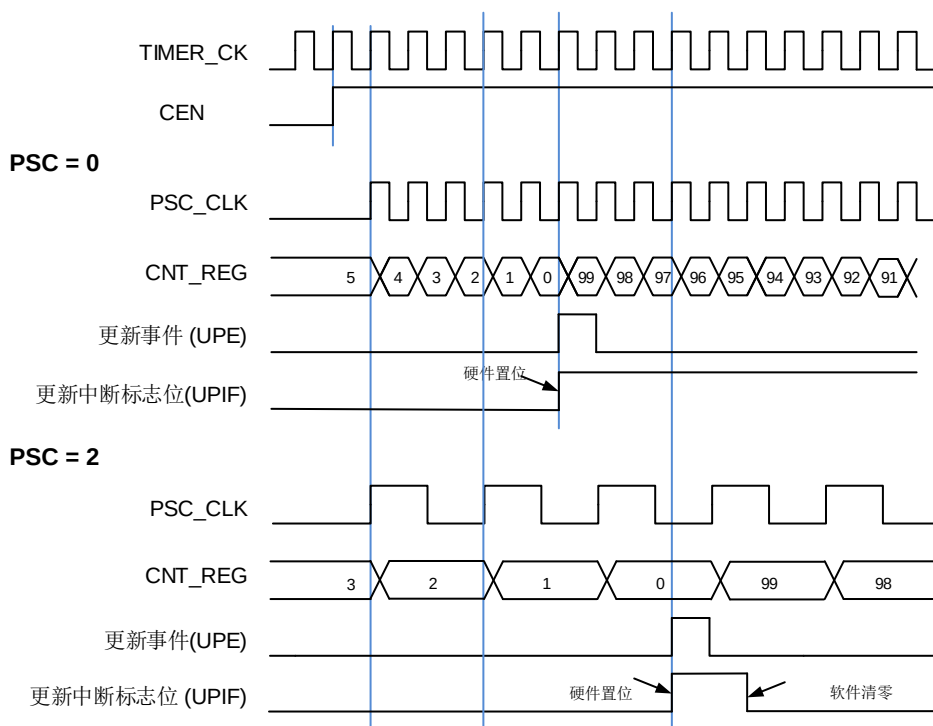
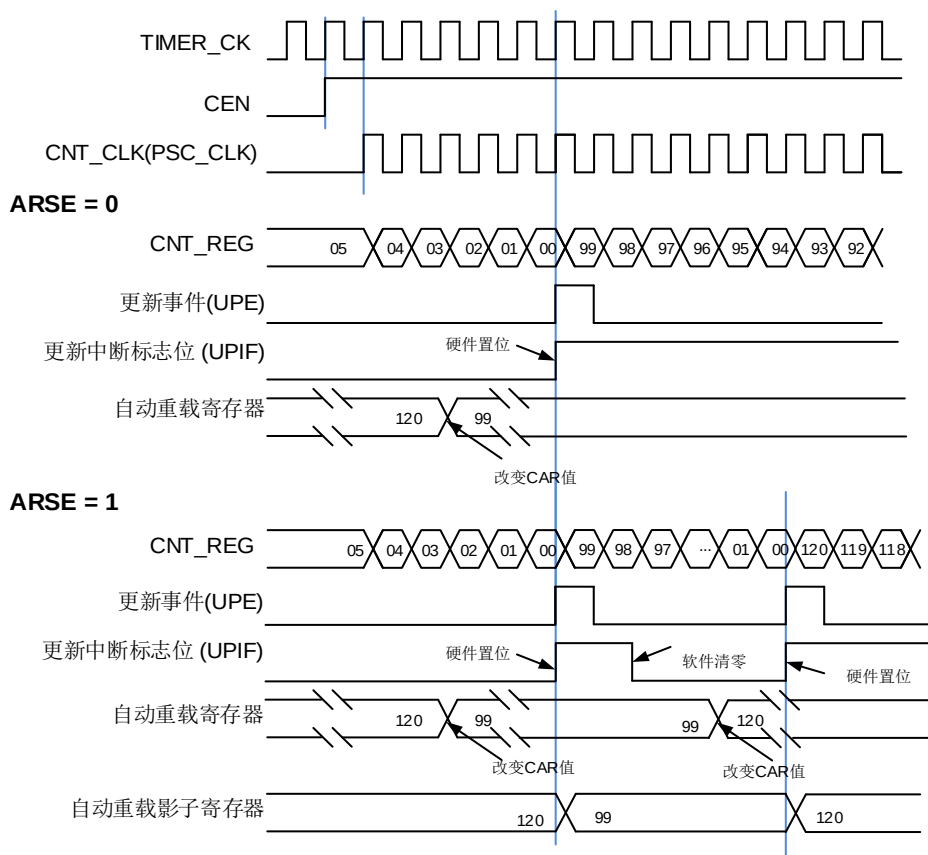


图 21-61. 向下计数时序图，在运行时改变 **TIMERx_CAR** 寄存器值

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。

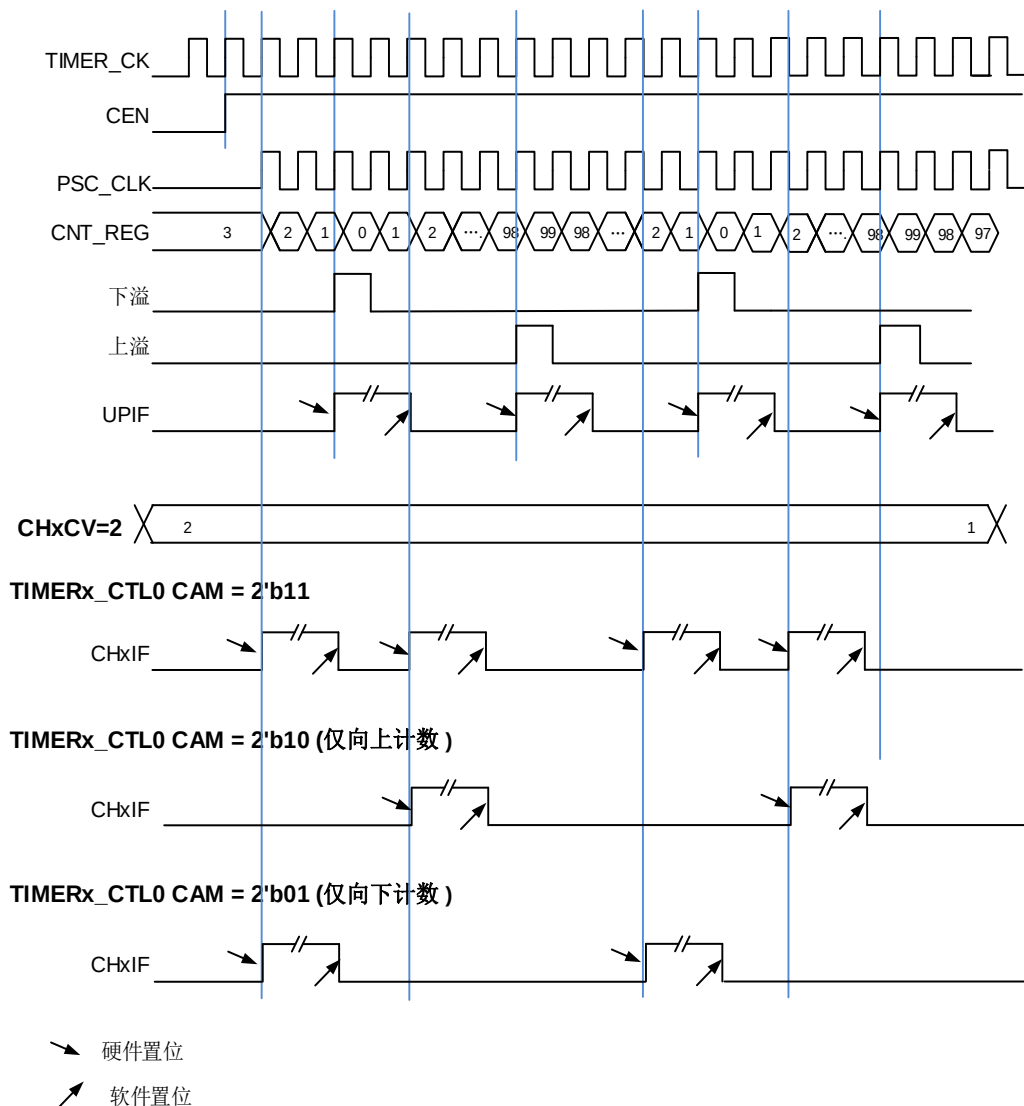
将 **TIMERx_SWEVG** 寄存器的 **UPG** 位置 1 可以初始化计数值为 0，并产生一个更新事件，而无需考虑计数器在中央模式下是向上计数还是向下计数。

上溢或者下溢时，**TIMERx_INTF** 寄存器中的 **UIP** 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 21-62. 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 21-62. 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 21-62. 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。

[illegible]

配置步骤如下：

根据输入信号和请求信号的质量，配置相应的 CHxCAPFLT。

配置 CHxP 选择上升沿或者下降沿。

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

使能相应中断，可以获得中断和 DMA 请求。

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。

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 CIO。配置 TIMERx_CHCTL0 寄存器中 CH0MS 为 2'b01，选择通道 0 的捕获信号为 CIO 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 CIO 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 21-64. 输出比较逻辑 (x = 0, 1, 2, 3)

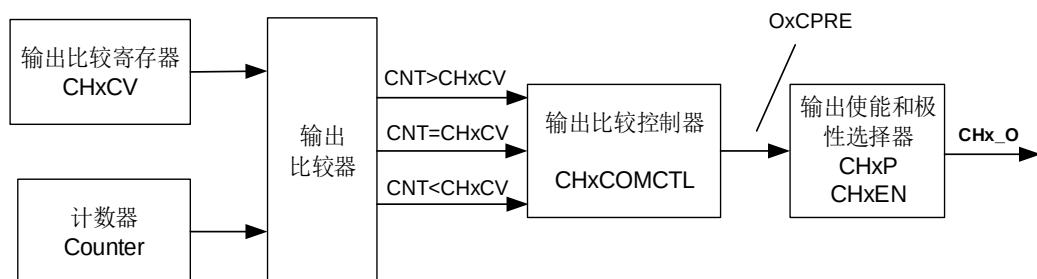


图 21-64. 输出比较逻辑 (x = 0, 1, 2, 3) 给出了输出比较的逻辑电路。通道输出信号 CHx_O 与 OxCPRE 信号（详情请见[通道输出准备信号](#)）的关系描述如下：OxCPRE 信号高电平有效，CHx_O 的输出情况与 OxCPRE 信号，CHxP 位和 CHxEN 位有关（具体情况请见 TIMERx_CHCTL2 寄存器中的描述）。例如，当设置 CHxP=0（CHx_O 高电平有效，与 OxCPRE 输出极性相同）、CHxEN=1（CHx_O 输出使能）时：

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。

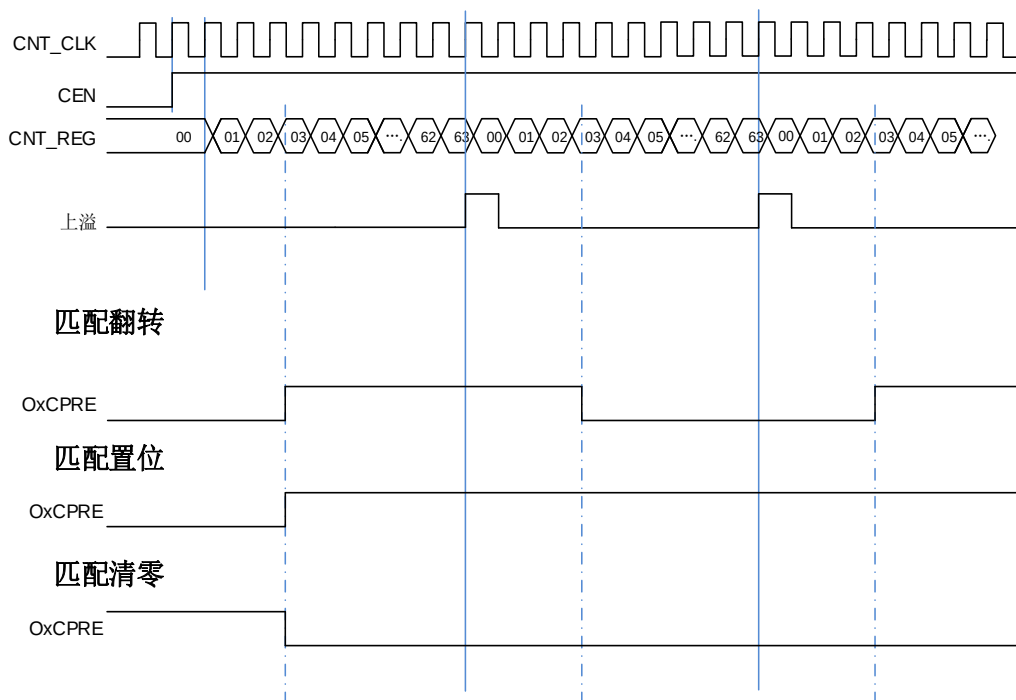
第四步：通过 `TIMERx_CAR` 寄存器和 `TIMERx_CHxCV` 寄存器配置输出比较时基：

`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 21-65. 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 21-65. 三种输出比较模式



PWM 模式

在 PWM 输出模式下（PWM 模式 0 是配置 `CHxCOMCTL` 为 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 21-66. EAPWM 时序图](#)显示了 EAPWM 的输出波形和中断。

CAPWM 的周期由（ $2 \times \text{TIMERx\_CAR}$ 寄存器值）决定，占空比由（ $2 \times \text{TIMERx\_CHxCV}$ 寄存器值）决定。[图 21-67. 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 21-66. EAPWM 时序图

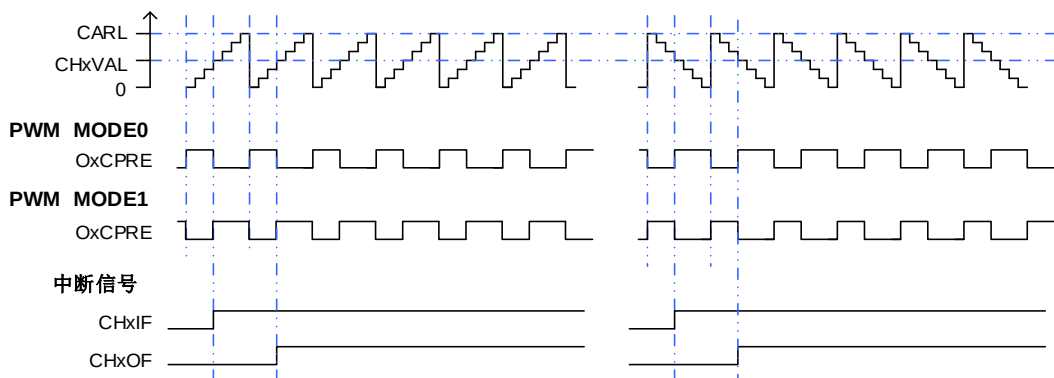
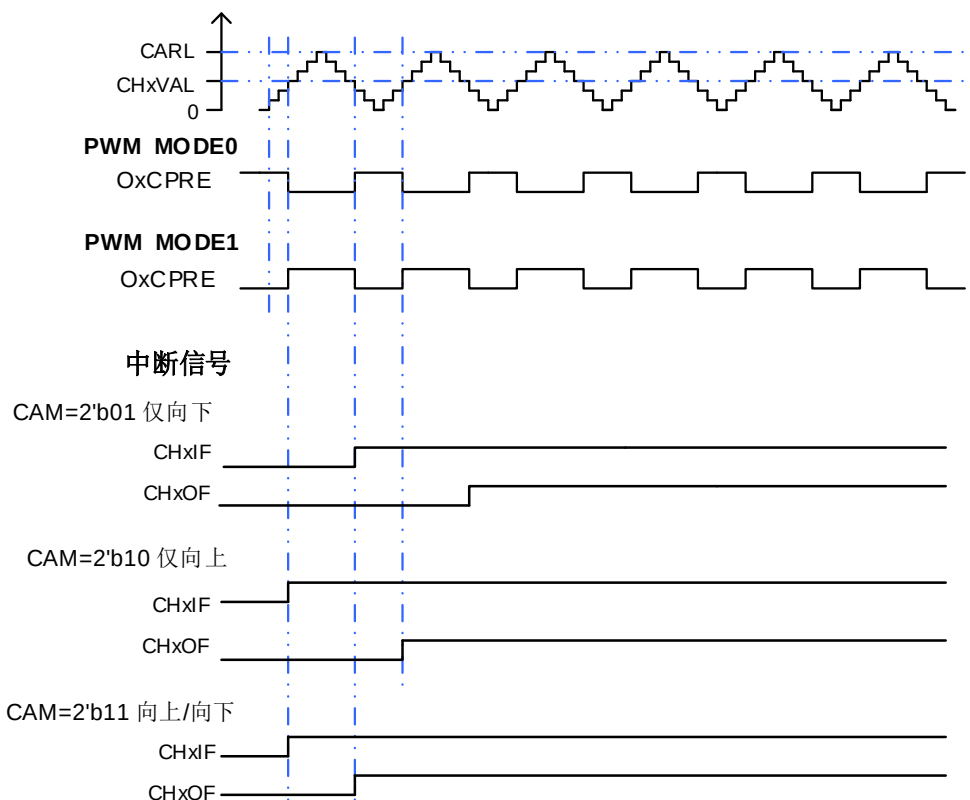


图 21-67. CAPWM 时序图



通道输出准备信号

根据图21-64. 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 TIMEx_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。直到下一次有效电平发生时，OxCPRE 信号才会回到有效电平状态。

正交译码器

参考[高级定时器\(TIMEx, x = 0, 7\)正交译码器](#)。

非正交译码器

参考[高级定时器\(TIMEx, x = 0, 7\)非正交译码器](#)。

霍尔传感器接口功能

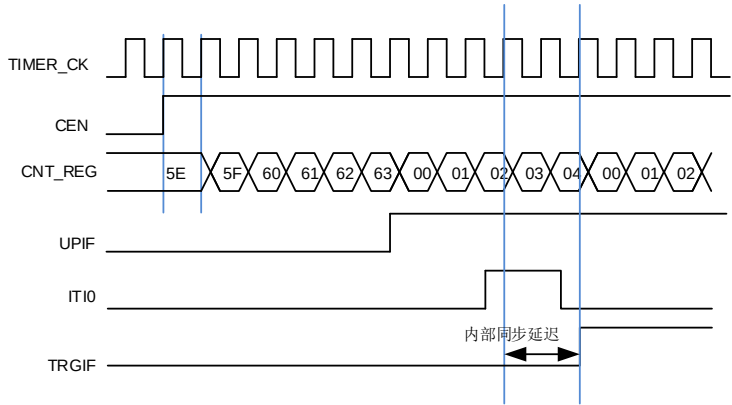
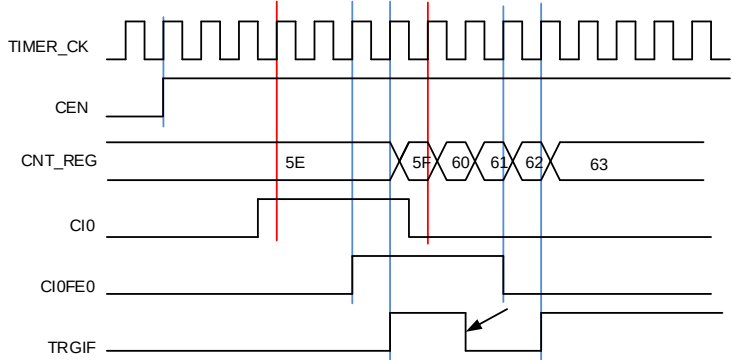
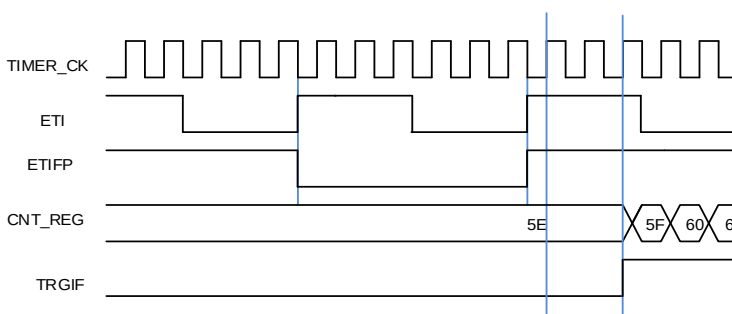
参考[高级定时器\(TIMEx, x = 0, 7\)霍尔传感器接口功能](#)。

主-从管理

TIMEx 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_TIMExCFG(x = 1, 2)寄存器中的 TSCFGy[4:0] (y = 3..7)位域来确定，具体的输入触发源可以通过 TSCFGy[4:0] (y = 3..7)位域值来选择。

表 21-11. 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 y = 6: 外部时钟模式 0 y = 7: 复位 + 事件模式	TSCFGy[4:0] 00000: 模式禁能 00001: ITI0 00010: ITI1 00011: ITI2 00100: ITI3 00101: CI0F_ED 00110: CI0FE0 00111: CI1FE1 01000: ETIFP	如果触发源是 CI0FE0 或者 CI1FE1，配置 CHxP 和 CHxNP 来选择极性和反相。 如果触发源是 ETIFP（滤波后的 ETI 外部触发输入），配置 ETP 选择极性和反相	触发源 ITIx，滤波和预分频不可用 触发源 Cix，配置 CHxCAPFLT 设置滤波，分频不可用 触发源是 ETIFP，滤波和预分频不可用
例1	复位模式 当触发输入上升沿，计数器清零重启	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

	模式选择	触发源选择	极性选择	滤波和预分频
	图 21-68. 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 21-69. 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
	图 21-70. 事件模式 			
例 4	复位 + 事件模式 当触发输入的上升沿到来时，计数器被重新初始化并开始计数。该模式仅用于可延时的单脉冲模式。			
例 5	暂停 + 复位模式 当触发输入的上升沿或下降沿（由 TIMERx_SMCFG 寄存器中的 PRMRPSEL 位配置）到来			

模式选择	触发源选择	极性选择	滤波和预分频
时，计数器将复位。当触发输入高时计数器计数，当触发输入低时计数器停止。在这种模式下，计数器的开始和停止可以被控制。			

单脉冲模式

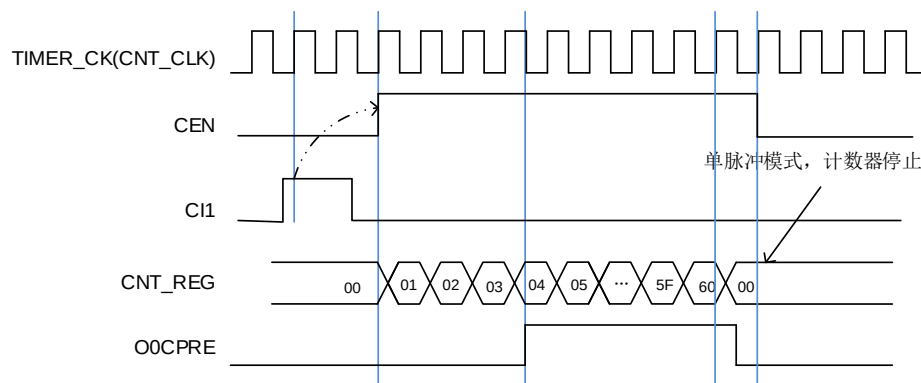
单脉冲模式与重复模式是相反的，设置 `TIMERx_CTL0` 寄存器的 `SPM` 位置 1，则使能单脉冲模式。当 `SPM` 置 1，计数器在下次更新事件到来后清零并停止计数。为了得到脉冲波，可以通过设置 `CHxCOMCTL` 配置 `TIMERx` 为 PWM 模式或者比较模式。

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。

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 21-71. 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。

输出 DIR 位

DIR 位可以在 CH2 和 CH3 通道上输出，配置在 TIMERx_CHCTL1 寄存器中的 CH2COMCTL[3:0]或 CH3COMCTL[3:0]位域值为 4b'1011 来使能该功能。

当计数器工作在中央对齐模式时，该功能可用于指示计数器的计数方向。当计数器工作在译码器模式时，此功能可用于指示外部信号的旋转方向。

UPIF 位备份功能

可以通过配置 TIMERx_CTL0 寄存器中的 UPIFBUEN 位来使能 UPIF 位的备份功能，UPIF 和 UPIFBU 位之间没有延迟，两者完全同步。

使能该功能后，TIMERx_INTF 寄存器中的 UPIF 位将会被实时备份到 TIMERx_CNT 寄存器中的 UPIFBU 位。这可以避免在读计数器和中断处理时产生冲突的情况。

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。

21.2.5. TIMERx 寄存器 (x = 1, 2)

TIMER1 基地址: 0x4000 0000

TIMER2 基地址: 0x4000 0400

控制寄存器 0 (TIMERx_CTL0)

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留						ETISEL[1:0]		保留							
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				UPIFBUE	保留	CKDIV[1:0]		ARSE	CAM[1:0]		DIR	SPM	UPS	UPDIS	CEN
				N											
				rw		rw		rw	rw		rw	rw	rw	rw	rw

位/位域	名称	描述
31:26	保留	必须保持复位值。
25:24	ETISEL[1:0]	TIMER 外部触发输入选择 00: 从 GPIO 选择 TIMER_ETI0 引脚 01: 从 GPIO 选择 TIMER_ETI1 引脚 10: 从 GPIO 选择 TIMER_ETI2 引脚 11: 从 GPIO 选择 TIMER_ETI3 引脚
23:12	保留	必须保持复位值。
11	UPIFBUE	UPIF位备份使能 0: 备份禁能。UPIF位没有备份到TIMERx_CNT寄存器中的UPIFBU位 1: 备份使能。UPIF位备份到TIMERx_CNT寄存器中的UPIFBU位
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=00），只有在向下计数时，通道的比较中断标志置 1 10: 中央对齐向上计数置 1 模式。计数器在中央计数模式计数，通道被配置在输出模式（TIMERx_CHCTL0 寄存器中 CHxMS=00），只有在向上计数时，通道的比较中断标志置 1 11: 中央对齐上下计数置 1 模式。计数器在中央计数模式计数，通道被配置在输出模式（TIMERx_CHCTL0 寄存器中 CHxMS=00），在向上和向下计数时，通道的比较中断标志都会置 1 当计数器使能以后，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
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								TI0S	MMC[2:0]			DMAS	保留		
								rw	rw			rw			

位/位域	名称	描述
31:8	保留	必须保持复位值。
7	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。 000: 复位。TIMERx_SWEVG 寄存器的 UPG 位被置 1 或从模式控制器产生复位触发一次 TRGO 脉冲, 后一种情况下, TRGO 上的信号相对实际的复位会有一个延迟。 001: 使能。此模式可用于同时启动多个定时器或控制在一段时间内使能从定时器。主模式控制器选择计数器使能信号作为触发输出 TRGO。当 CEN 控制位被置 1 或者暂停模式下触发输入为高电平时, 计数器使能信号被置 1。在暂停模式下, 计数器使能信号受控于触发输入, 在触发输入和 TRGO 上会有一个延迟, 除非选择了主/从模式。 010: 更新。主模式控制器选择更新事件作为 TRGO。 011: 捕获/比较脉冲。通道 0 在发生一次捕获或一次比较成功时, 主模式控制器产生一个 TRGO 脉冲 100: 比较。在这种模式下主模式控制器选择 O0CPRE 信号被用于作为触发输出 TRGO 101: 比较。在这种模式下主模式控制器选择 O1CPRE 信号被用于作为触发输出 TRGO 110: 比较。在这种模式下主模式控制器选择 O2CPRE 信号被用于作为触发输出 TRGO 111: 比较。在这种模式下主模式控制器选择 O3CPRE 信号被用于作为触发输出 TRGO

3	DMAS	DMA 请求源选择 0: 当通道捕获/比较事件发生时, 发送通道 CHx 的 DMA 请求 1: 当更新事件发生时, 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 5'b01000。 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，TRGI 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															
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
	RW		RW	RW	RW	RW	RW		RW		RW	RW	RW	RW	RW

位/位域	名称	描述
31:15	保留	必须保持复位值。
14	TRGDEN	触发 DMA 请求使能 0: 禁止触发 DMA 请求

		1: 使能触发 DMA 请求
13	保留	必须保持复位值。
12	CH3DEN	通道 3 比较/捕获 DMA 请求使能 0: 禁止通道 3 比较/捕获 DMA 请求 1: 使能通道 3 比较/捕获 DMA 请求
11	CH2DEN	通道 2 比较/捕获 DMA 请求使能 0: 禁止通道 2 比较/捕获 DMA 请求 1: 使能通道 2 比较/捕获 DMA 请求
10	CH1DEN	通道 1 比较/捕获 DMA 请求使能 0: 禁止通道 1 比较/捕获 DMA 请求 1: 使能通道 1 比较/捕获 DMA 请求
9	CH0DEN	通道 0 比较/捕获 DMA 请求使能 0: 禁止通道 0 比较/捕获 DMA 请求 1: 使能通道 0 比较/捕获 DMA 请求
8	UPDEN	更新 DMA 请求使能 0: 禁止更新 DMA 请求 1: 使能更新 DMA 请求
7	保留	必须保持复位值。
6	TRGIE	触发中断使能 0: 禁止触发中断 1: 使能触发中断
5	保留	必须保持复位值。
4	CH3IE	通道 3 比较/捕获中断使能 0: 禁止通道 3 中断 1: 使能通道 3 中断
3	CH2IE	通道 2 比较/捕获中断使能 0: 禁止通道 2 中断 1: 使能通道 2 中断
2	CH1IE	通道 1 比较/捕获中断使能 0: 禁止通道 1 中断 1: 使能通道 1 中断
1	CH0IE	通道 0 比较/捕获中断使能 0: 禁止通道 0 中断 1: 使能通道 0 中断
0	UPIE	更新中断使能 0: 禁止更新中断

1: 使能更新中断

中断标志寄存器 (TIMERx_INTF)

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		CH3OF	CH2OF	CH1OF	CH0OF	保留		TRGIF	保留	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

位/位域	名称	描述
31:13	保留	必须保持复位值。
12	CH3OF	通道 3 捕获溢出标志 参见 CH0OF 描述。
11	CH2OF	通道 2 捕获溢出标志 参见 CH0OF 描述。
10	CH1OF	通道 1 捕获溢出标志 参见 CH0OF 描述。
9	CH0OF	通道 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
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1COM CEN	CH1COMCTL[2:0]			CH1COM SEN	CH1COM FEN	CH1MS[1:0]		CH0COM CEN	CH0COMCTL[2:0]			CH0COM SEN	CH0COM FEN	CH0MS[1:0]	
CH1CAPFLT[3:0]				CH1CAPPSC[1:0]				CH0CAPFLT[3:0]				CH0CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

输出比较模式：

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	CH1COMCEN	通道 1 输出比较清 0 使能 参见 CH0COMCEN 描述。
14:12	CH1COMCTL[2:0]	通道 1 输出比较模式 参见 CH0COMCTL 描述。
11	CH1COMSEN	通道 1 输出比较影子寄存器使能 参见 CH0COMSEN 描述。
10	CH1COMFEN	通道 1 输出比较快速使能 参见 CH0COMFEN 描述。

9:8	CH1MS[1:0]	通道 1 模式选择 CH1MS[1:0] 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。 00: 通道 1 配置为输出 01: 通道 1 配置为输入，IS1 映射在 CI1FE1 上 10: 通道 1 配置为输入，IS1 映射在 CI0FE1 上 11: 通道 1 配置为输入，IS1 映射在 ITS 上，此模式仅工作在内部触发器输入被选中时（由 SYSCFG_TIMERxCFG2(x = 1,2)寄存器中的 TSCFG15[4:0]位域选择）。
7	CH0COMCEN	通道 0 输出比较清 0 使能 当此位被置 1，当检测到 OCPRE 清除源信号输入高电平时，O0CPRE 参考信号被清 0。OCPRE 清除源由 TIMERx_SMCFG 寄存器的 OCRC 位选择。 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[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。 000: 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 001: 匹配时设置为高。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为高。 010: 匹配时设置为低。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 为低。 011: 匹配时翻转。当计数器的值与捕获/比较值寄存器 TIMERx_CH0CV 相同时，强制 O0CPRE 翻转。 100: 强制为低。强制 O0CPRE 为低电平 101: 强制为高。强制 O0CPRE 为高电平 110: PWM 模式 0。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为有效电平，否则为无效电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为无效电平，否则为有效电平。 111: PWM 模式 1。在向上计数时，一旦计数器值小于 TIMERx_CH0CV 时，O0CPRE 为无效电平，否则为有效电平。在向下计数时，一旦计数器的值大于 TIMERx_CH0CV 时，O0CPRE 为有效电平，否则为无效电平。 在 PWM 模式中，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，TIMERx_CH0CV 寄存器的影子寄存器被使能，根据影子寄存器加载功能的相关配置在设置时刻进行更新。 0: 禁止通道 0 输出/比较影子寄存器 1: 使能通道 0 输出/比较影子寄存器 仅在单脉冲模式下（TIMERx_CTL0 寄存器的 SPM = 1），可以在未确认预装载寄存器情况下使用 PWM 模式。

2	CH0COMFEN	通道0输出比较快速使能 0: 通道0输出比较快速禁能。仅比较结果输入作为有效边沿时会产生比较匹配, 并将CH0_O设置为比较电平, 激活CH0_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。触发输入信号的有效边沿和比较结果都会产生比较匹配, 并将CH0_O设置为比较电平。当触发信号的输入作为有效边沿时, 激活CH0_O输出的最小延时为3个时钟周期。
1:0	CH0MS[1:0]	通道0 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭 (TIMERx_CHCTL2 寄存器的CH0EN位被清0) 时, CH0MS[2:0]位才可以写。 00: 通道0配置为输出 01: 通道0配置为输入, IS0映射在CI0FE0上 10: 通道0配置为输入, IS0映射在CI1FE0上 11: 通道0配置为输入, IS0映射在ITS上。此模式仅工作在内部触发输入被选中时 (由SYSCFG_TIMERxCFG2(x = 1,2)寄存器中的TSCFG15[4:0]位域选择)。

输入捕获模式:

位/位域	名称	描述
31:16	保留	必须保持复位值。
15:12	CH1CAPFLT[3:0]	通道1输入捕获滤波控制 参见CH0CAPFLT描述。
11:10	CH1CAPPSC[1:0]	通道1输入捕获预分频器 参见CH0CAPPSC描述。
9:8	CH1MS[1:0]	通道1模式选择 与输出模式相同。
7:4	CH0CAPFLT[3:0]	通道0输入捕获滤波控制 数字滤波器由一个事件计数器组成, 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 `TIMERx_CHCTL2` 寄存器中的 `CH0EN = 0` 时，则预分频器复位。
00: 无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获
01: 每 2 个事件触发一次捕获
10: 每 4 个事件触发一次捕获
11: 每 8 个事件触发一次捕获
- 1:0 CH0MS[1:0] 通道 0 模式选择
与输出比较模式相同。

通道控制寄存器 1 (TIMERx_CHCTL1)

地址偏移: 0x1C

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3COM CEN	CH3COMCTL[2:0]			CH3COM SEN	CH3COM FEN	CH3MS[1:0]		CH2COM CEN	CH2COMCTL[2:0]			CH2COM SEN	CH2COM FEN	CH2MS[1:0]	
CH3CAPFLT[3:0]				CH3CAPPSC[1:0]				CH2CAPFLT[3:0]			CH2CAPPSC[1:0]				
rw				rw		rw		rw			rw		rw		

输出比较模式:

位/位域	名称	描述
31:16	保留	必须保持复位值。
15	CH3COMCEN	通道 3 输出比较清 0 使能 参见 CH0COMCEN 描述。
14:12	CH3COMCTL[2:0]	通道 3 输出比较模式 参见 CH0COMCTL 描述。
11	CH3COMSEN	通道 3 输出比较影子寄存器使能 参见 CH0COMSEN 描述。
10	CH3COMFEN	通道 3 输出比较快速使能 参见 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。 00: 通道 3 配置为输出

		01: 通道 3 配置为输入, IS3 映射在 CI3FE3 上
		10: 通道 3 配置为输入, IS3 映射在 CI2FE3 上
		11: 通道 3 配置为输入, IS3 映射在 ITS 上, 此模式仅工作在内部触发器输入被选中时 (由 SYSCFG_TIMERxCFG2(x = 1,2)寄存器中的 TSCFG15[4:0]位域选择)。
7	CH2COMCEN	通道 2 输出比较清 0 使能 当此位被置 1, 当检测到 OCPRE 清除源信号输入高电平时, O2CPRE 参考信号被清 0。OCPRE 清除源由 TIMERx_SMCFG 寄存器的 OCRC 位选择。 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 参考 CH0COMCTL[2:0]描述。
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, 根据影子寄存器加载功能的相关配置在设置时刻进行更新。 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 参考 CH0COMFEN 描述。
1:0	CH2MS[1:0]	通道 2 I/O 模式选择 这些位定义了通道的工作模式和输入信号的选择。只有当通道关闭 (TIMERx_CHCTL2 寄存器的 CH2EN 位被清 0) 时这些位才可写。 00: 通道 2 配置为输出 01: 通道 2 配置为输入, IS2 映射在 CI2FE2 上 10: 通道 2 配置为输入, IS2 映射在 CI3FE2 上 11: 通道 2 配置为输入, IS2 映射在 ITS 上。此模式仅工作在内部触发输入被选中时 (由 SYSCFG_TIMERxCFG2(x = 1,2)寄存器中的 TSCFG15[4:0]位域选择)。

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 $CH2$ 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 $TIMERx\_CHCTL2$ 寄存器中的 $CH2EN = 0$ 时，则预分频器复位。 00: 无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获 01: 每 2 个事件触发一次捕获 10: 每 4 个事件触发一次捕获 11: 每 8 个事件触发一次捕获
1:0	CH2MS[1:0]	通道 2 模式选择 与输出比较模式相同。

通道控制寄存器 2 (TIMERx_CHCTL2)

地址偏移: $0x20$

复位值: $0x0000\ 0000$

该寄存器只能按字 (32 位) 访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3NP	保留	CH3P	CH3EN	CH2NP	保留	CH2P	CH2EN	CH1NP	保留	CH1P	CH1EN	CH0NP	保留	CH0P	CH0EN
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	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 CI0 的极性选择控制信号。
2	保留	必须保持复位值。
1	CH0P	通道 0 极性 当通道 0 配置为输出模式时，此位定义了输出信号极性。 0: 通道 0 高电平有效 1: 通道 0 低电平有效 当通道 0 配置为输入模式时，此位定义了通道 0 输入信号极性。[CH0NP, CH0P]将选择 CI0FE0 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 C1xFE0 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 C1xFE0 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

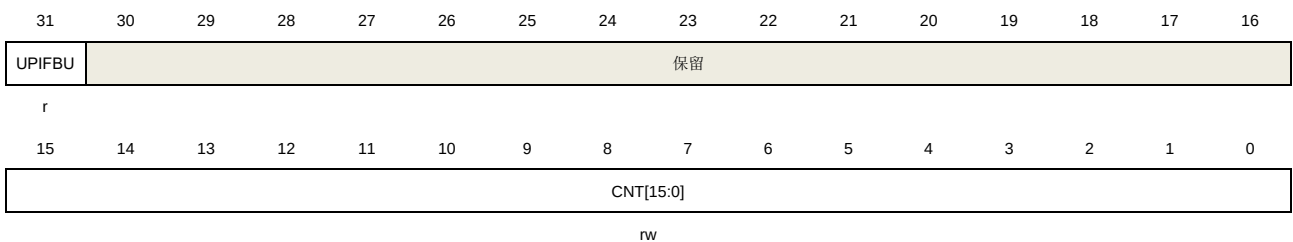
1: 使能通道 0

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。



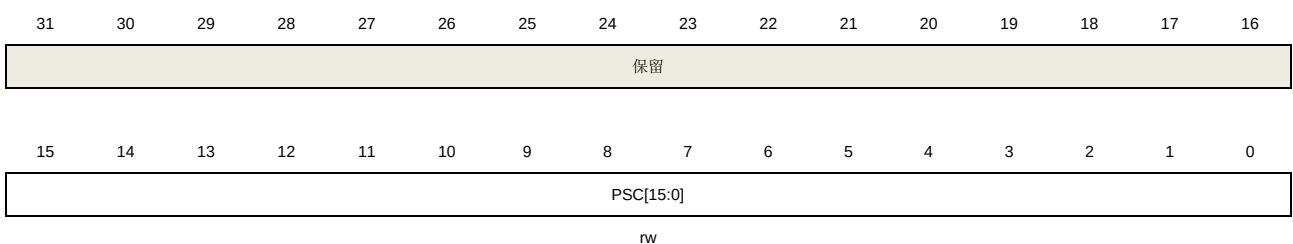
位/位域	名称	描述
31	UPIFBU	UPIF位备份 该位只读，是 TIMERx_INTF 寄存器的 UPIF 位的备份值。当 UPIFBUEN = 1 时，该位有效，若 UPIFBUEN = 0，该位保留，读取该位值为零。
30:16	保留	必须保持复位值。
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

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。



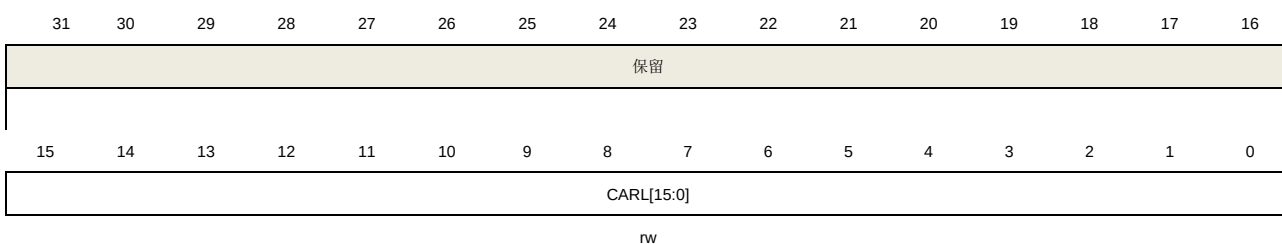
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于 PSC 时钟除以 (PSC+1)，每次当更新事件产生时，PSC 的值被装入当前预分频寄存器。

计数器自动重载寄存器 (TIMERx_CAR)

地址偏移: 0x2C

复位值: 0x 0000 FFFF

该寄存器只能按字 (32 位) 访问。



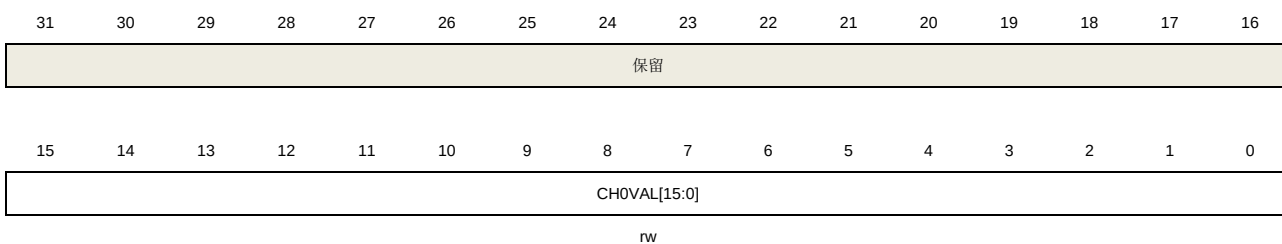
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CARL[15:0]	计数器自动重载值 (0 位至 15 位) 这些位定义了计数器的自动重载值。

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

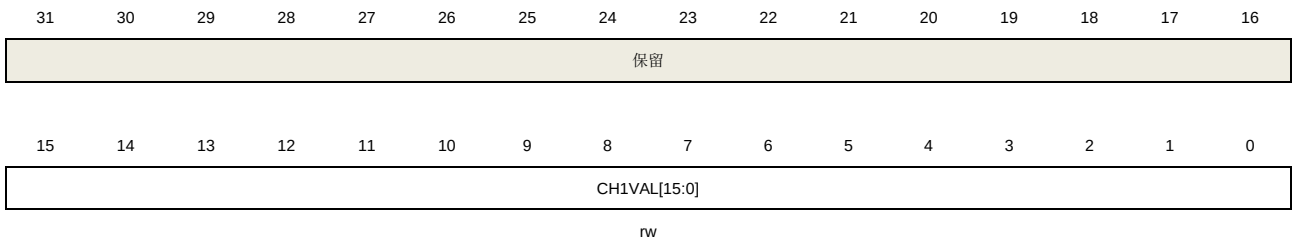
寄存器后，影子寄存器值随每次更新事件更新。

通道 1 捕获/比较值寄存器(TIMERx_CH1CV)

地址偏移：0x38

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



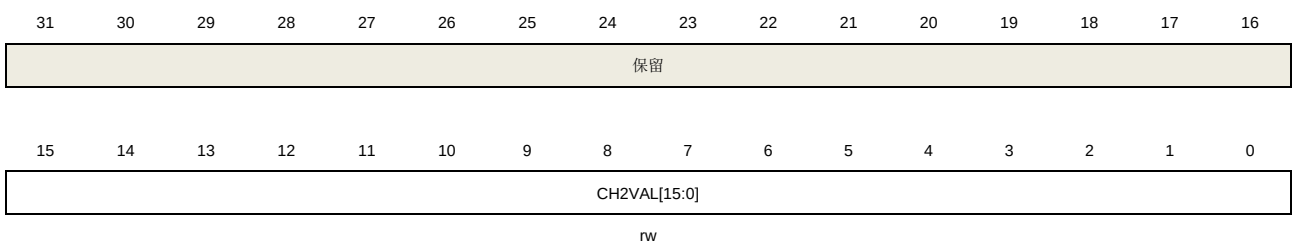
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH1VAL[15:0]	通道 1 的捕获或比较值 当通道 1 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 1 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 2 捕获/比较值寄存器(TIMERx_CH2CV)

地址偏移：0x3C

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



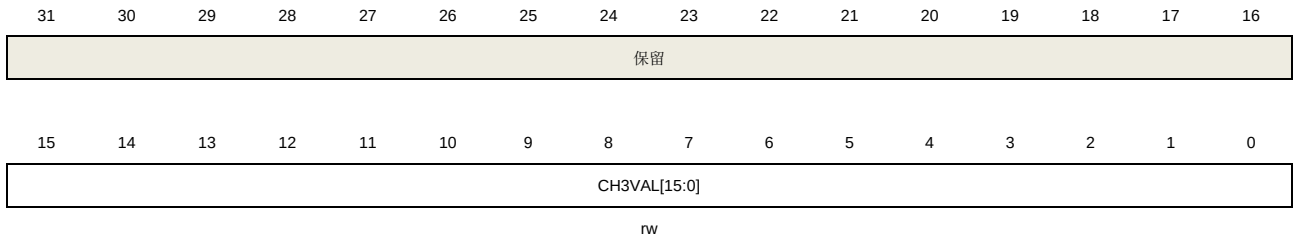
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH2VAL[15:0]	通道 2 的捕获或比较值 当通道 2 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 2 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

通道 3 捕获/比较值寄存器(TIMERx_CH3CV)

地址偏移: 0x40

复位值: 0x0000 0000

该寄存器只能按字（32 位）访问。



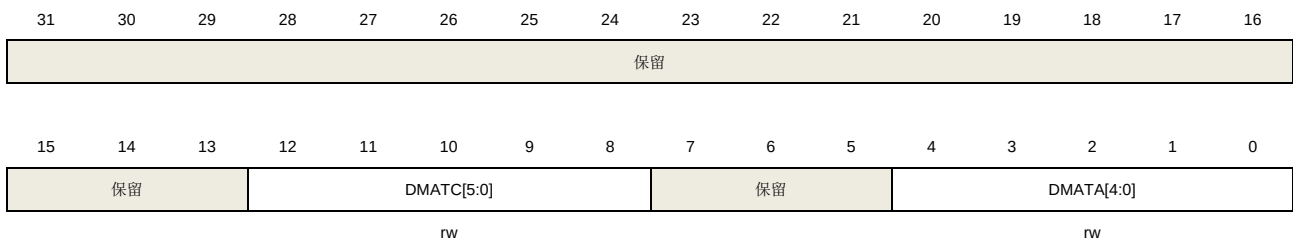
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH3VAL[15:0]	通道 3 的捕获或比较值 当通道 3 配置为输入模式时，这些位决定了上次捕获事件的计数器值。并且本寄存器为只读。 当通道 3 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应影子寄存器后，影子寄存器值随每次更新事件更新。

DMA 配置寄存器 (TIMERx_DMACFG)

地址偏移: 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
31:13	保留	必须保持复位值。
12:8	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。 5'b00000: 传输1次 5'b00001: 传输2次 ... 5'b10000: 传输17次
7:5	保留	必须保持复位值。

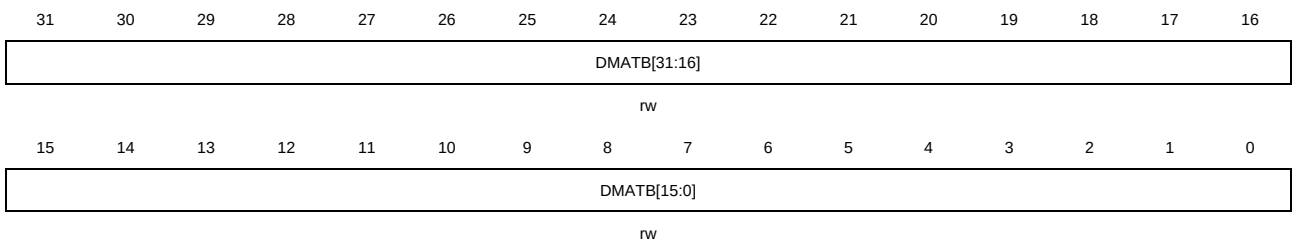
4:0	DMATA [4:0]	DMA 传输起始地址 该位域定义了 DMA 访问 TIMERx_DMATB 寄存器的第一个地址。当通过 TIMERx_DMA 第一次访问时，访问的就是该位域指定的地址。第二次访问 TIMERx_DMATB 时，将访问起始地址+0x4。 5'b00000: TIMERx_CTL0 5'b00001: TIMERx_CTL1 ... 5'b10000: TIMERx_CH3CV 总之：起始地址 = TIMERx_CTL0 + DMATA*4。

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。



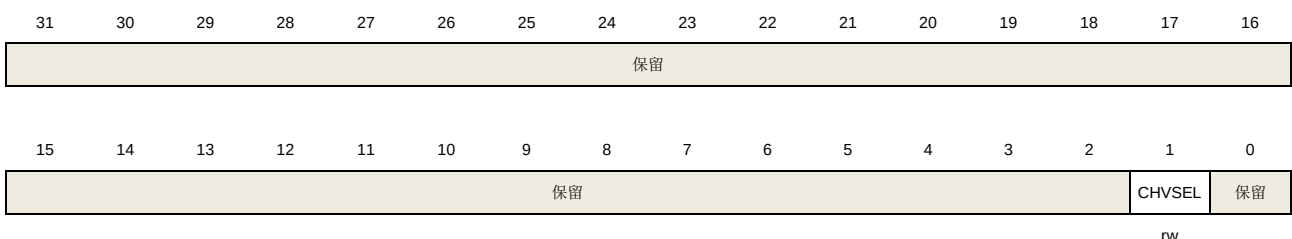
位/位域	名称	描述
31:0	DMATB [31:0]	DMA 发送缓冲 对这个寄存器的读或写，（起始地址+传输次数*4）地址范围内的寄存器会被访问 传输次数由硬件计算，范围为 0 到 DMATC。

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。

22. 通用定时器（GPTIMER）

22.1. 简介

通用定时器（GPTIMER0/1）是 2 通道定时器，支持输入捕获和输出比较。可以产生 PWM 信号控制功率因数校正（PFC）。通用定时器含有一个 16 位无符号计数器。

通用定时器是可编程的，可以用于计数，其外部事件可以驱动其他定时器。

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（通过 EVIC 实现）。

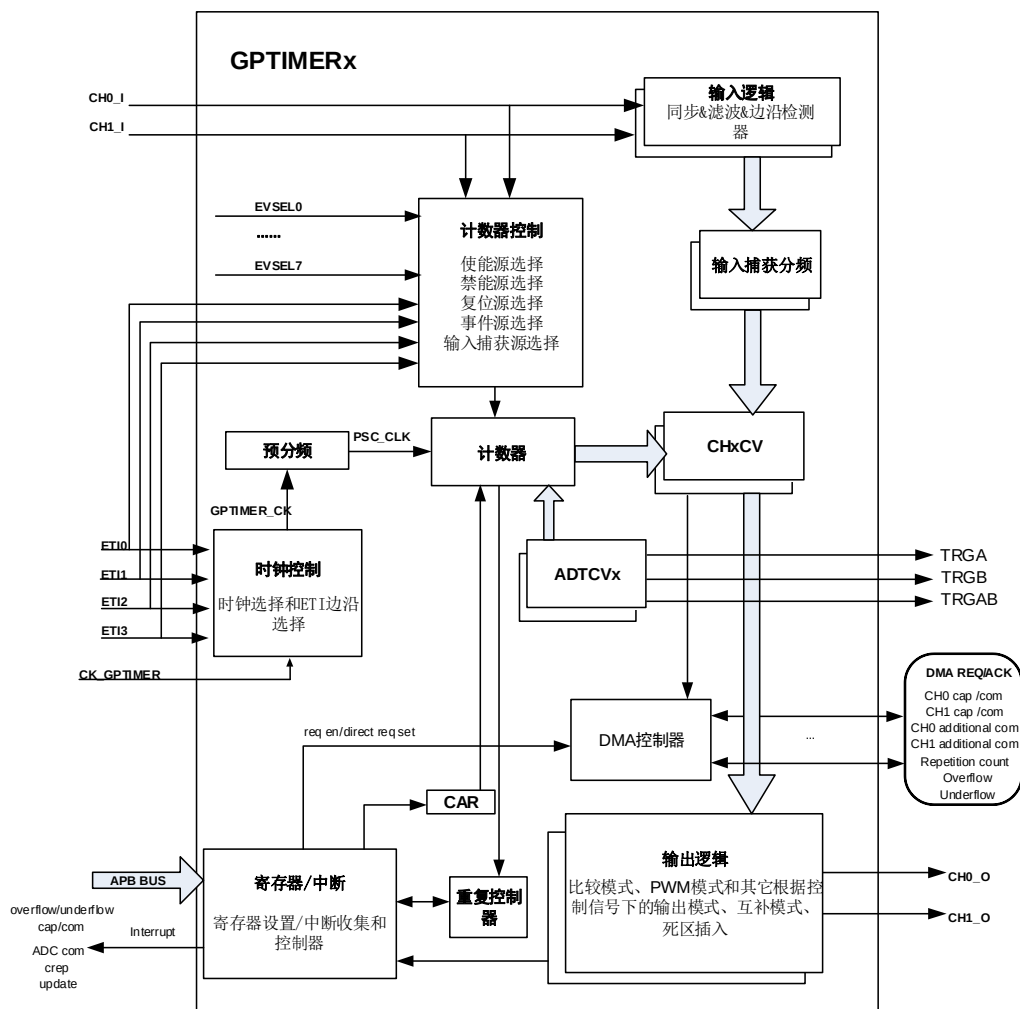
22.2. 主要特征

- 总通道数：2；
- 计数器宽度：16位；
- 时钟源可选：内部时钟，外部触发；
- 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模式和单脉冲模式；
- 自动重载功能；
- 可编程的计数器重复功能；
- 中断输出和DMA请求：计数器复位事件，触发事件，比较/捕获事件；
- 多个定时器的同步启动/停止/复位，可通过EVIC、通道输入、ETI输入触发；
- 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；
- ADC触发请求输出及引脚检测，且有重复跳过功能；
- 全局更新；
- 互补输出和死区插入（死区时间可非对称）；
- 与GTOC协同工作实现停止输出功能；
- 外部脉冲宽度测量。

22.3. 结构框图

图 21-72. GPTIMER 结构框图提供了低功耗定时器的内部配置细节。

图 21-72. GPTIMER 结构框图



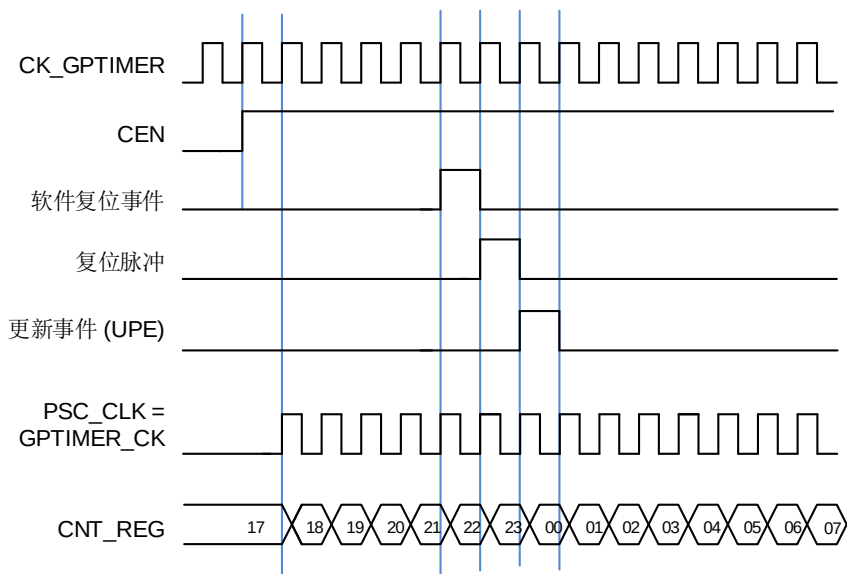
22.4. 功能描述

22.4.1. 时钟源配置

GPTIMER0 / 1 可以选择内部时钟源 CK_GPTIMER 或者外部引脚输入 ETI0/ETI1/ETI2/ETI3（通过 GTOC 滤波和极性选择后，提供给 GPTIMER）通过 GPTIMERx_CTL0 寄存器 CKSSEL[2:0] 位域选择，在使用外部输入的时钟源时，可以通过寄存器选择有效边沿。

- 当 GPTIMERx_CTL0 寄存器中 CKSSEL[2:0] = 3'b000 时，内部时钟来自于 RCU 模块的 CK_GPTIMER。当 CEN 置位，CK_GPTIMER 经过预分频器（预分频值由 GPTIMERx_PSC 寄存器确定）产生 PSC_CLK。

图 21-73. 内部时钟分频为 1 时正常模式下的控制电路



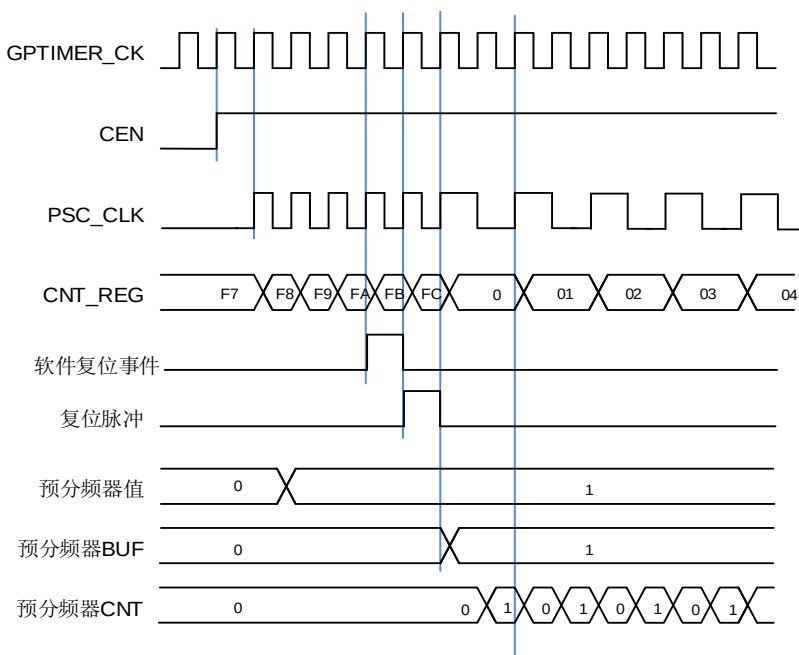
- 当GPTIMERx_CTL0寄存器中CKSSEL[2:0] = 3'b001/3'b010/3'b011/3'b100时，内部时钟来自于外部输入引脚。

GPTIMER_CK 可以通过外部触发引脚 ETI0/ETI1/ETI2/ETI3 的上升沿或下降沿计数。

22.4.2. 时钟预分频器

预分频器可以将定时器的时钟（GPTIMER_CK）频率按 1 到 65536 之间的任意值分频，分频后的时钟 PSC_CLK 驱动计数器计数。分频系数受预分频寄存器 GPTIMERx_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次更新事件到来时被采用。

图 21-74. 当预分频器的参数从 1 变到 2 时，计数器的时序图



22.4.3. 定时器使能/禁能/复位

定时器可以通过设置 GPTIMERx_CTL0 寄存器中的 CEN 位使能，通过复位 CEN 位禁能。CEN 位可以从以下来源更改：

- 设置 GPTIMERx_CTL0 寄存器中的 CEN 位；
- 当 GPTIMERx_ESSEL 寄存器中的 SWENCNT = 1 时，置位 GPTIMERx_SWEN 寄存器中 GPTIMERxSWEN 位。
- 当 GPTIMERx_DSSEL 寄存器中的 SWDISCNT = 1 时，置位 GPTIMERx_SWDIS 寄存器中 GPTIMERxSWDIS 位；
- 当 GPTIMERx_ESSEL 寄存器选择的硬件源事件（ETI 输入、EVIC 事件或通道输入）发生时；
- 当由 GPTIMERx_DSSEL 寄存器选择的硬件源事件（ETI 输入，EVIC 事件或通道输入）发生时。

定时器可以通过以下来源重置：

- 将 0 写入 GPTIMERx_CNT 寄存器。
- 当 GPTIMERx_RSSEL 寄存器中的 SWRSTCNT = 1 时，置位 GPTIMERx_SWRST 寄存器中的 GPTIMERxSWDIS 位。
- 当 GPTIMERx_RSSEL 寄存器选择的硬件源事件（ETI 输入，EVIC 事件或通道输入）发生时。

在定时器计数期间，不允许写入寄存器 GPTIMERx_CNT。当定时器计数或停止时，可以写入寄存器 GPTIMERx_SWRST。

在向下计数的边沿对齐模式下，GPTIMERx_CNT 的值在复位后变为周期值 GPTIMERx_CAR，在其他情况下，GPTIMERx_CNT 在复位后变为 0。

在事件计数模式下，当写入 GPTIMERx_SWRST 寄存器后，计数器会在一个 CK_GPTIMER 时钟后重置。

更新定时器优先级

GPTIMERx_CNT 更新操作的优先级（从高到低递减）：

1. CPU 写入行为：写入 CNT 寄存器或写入 GPTIMERx_SWRST 寄存器；
2. 由 GPTIMERx_REESEL 选定的硬件复位源复位；
3. 由 GPTIMERx_CUPEVSSEL 和 GPTIMERx_CDNEVSSEL 选定的硬件源执行上/下计数；
4. 周期性计数行为。

定时器控制优先级

CPU 在 GPTIMERx_CTL1 寄存器中写入 CEN 位具有最高优先级。GPTIMERxSWEN 寄存器中的 GPTIMERxSWEN 位和 GPTIMERx_SWDIS 寄存器中的 GPTIMERxSWDIS 位具有比硬件源的启动-停止控制更高的优先级。

当硬件源选择的开始和停止事件同时发生时，CEN 不会改变。

当 CEN 改变时，CPU 读取寄存器，读到的是改变之前的值。

在单脉冲模式下，硬件在周期结束时关闭 CEN 具有最高优先级。

22.4.4. 周期计数模式

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 GPTIMERx_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（GPTIMERx_CREP + 1）次上溢后产生更新事件，否则在每次上溢时都会产生更新事件。在向上计数模式中，GPTIMERx_CTL0 寄存器中的计数方向控制位 DIR 应该被设置成 0。

当 GPTIMERx_REESEL 选择的硬件复位源发生时，或者选择软件复位源并且设置了 GPTIMERx_SWRST 寄存器的相应位时，计数器值将被初始化为 0，并生成一个更新事件。

如果 GPTIMERx_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 21-75. 向上计数时序图，PSC = 0/2](#) 和 [图 21-76. 向上计数时序图，在运行时改变 GPTIMERx_CAR 寄存器的值](#) 给出一些例子，当 GPTIMERx_CAR = 0x99 时，计数器在不同预分频因子下的行为。

图 21-75. 向上计数时序图，PSC = 0/2

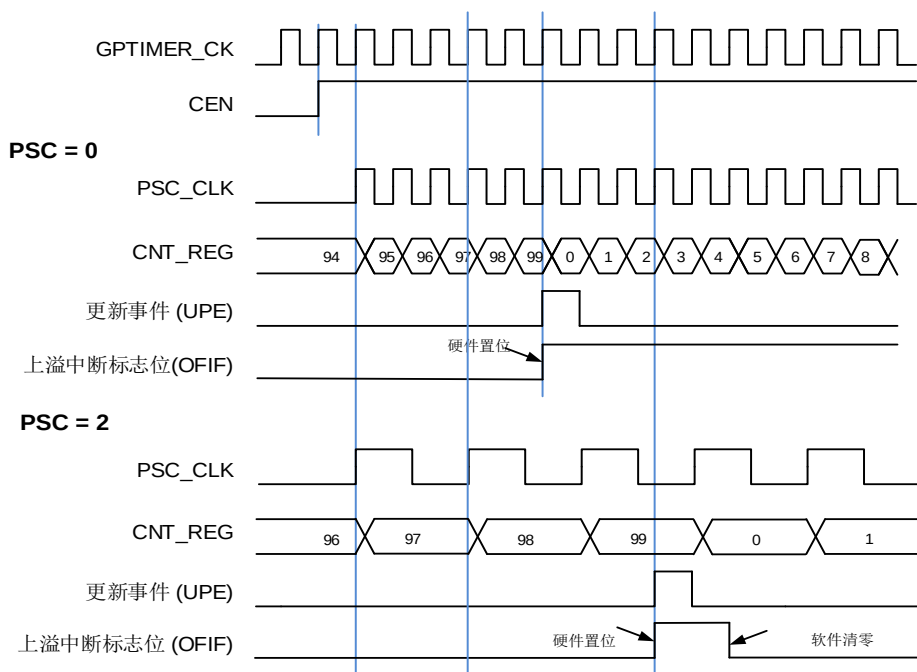
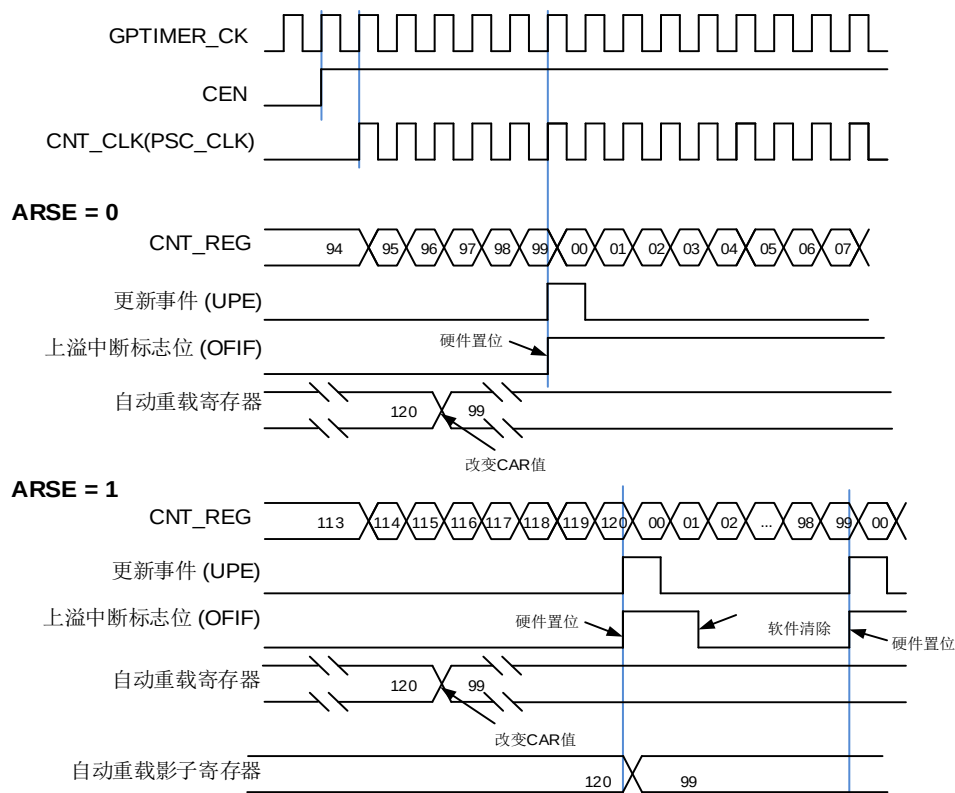


图 21-76. 向上计数时序图，在运行时改变 GPTIMERx_CAR 寄存器的值



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 **GPTIMERx_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 $(\text{TIMERx\_CREP} + 1)$ 次下溢后产生更新事件，否则在每次下溢时都会产生更新事件。在向下计数模式中，**GPTIMERx_CTL0** 寄存器中的计数方向控制位 **DIR** 应该被设置成 1。

当 **GPTIMERx_REESEL** 选定的硬件复位源发生时，或者选定软件复位源并且设置了 **GPTIMERx_SWRST** 寄存器的相应位时，计数器值将被初始化为计数器重载值，并生成一个更新事件。

如果 **GPTIMERx_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 21-77. 向下计数时序图，PSC = 0/2](#) 和 [图 21-78. 向下计数时序图，在运行时改变 GPTIMERx_CAR 寄存器值](#)给出了一些例子，当 **GPTIMERx_CAR** = 0x99 时，计数器在不同时钟频率下的行为。

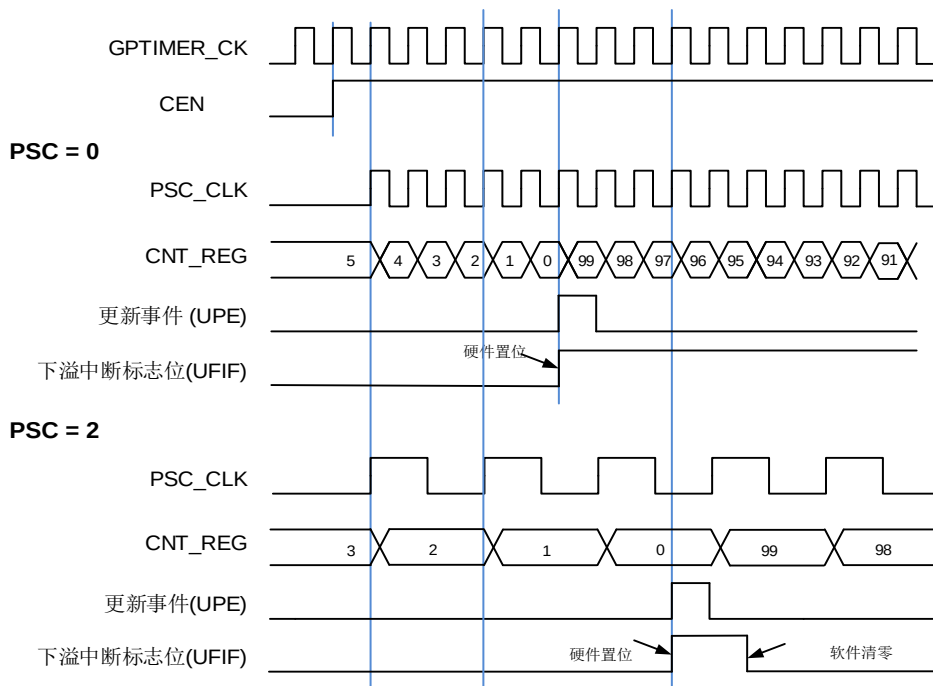
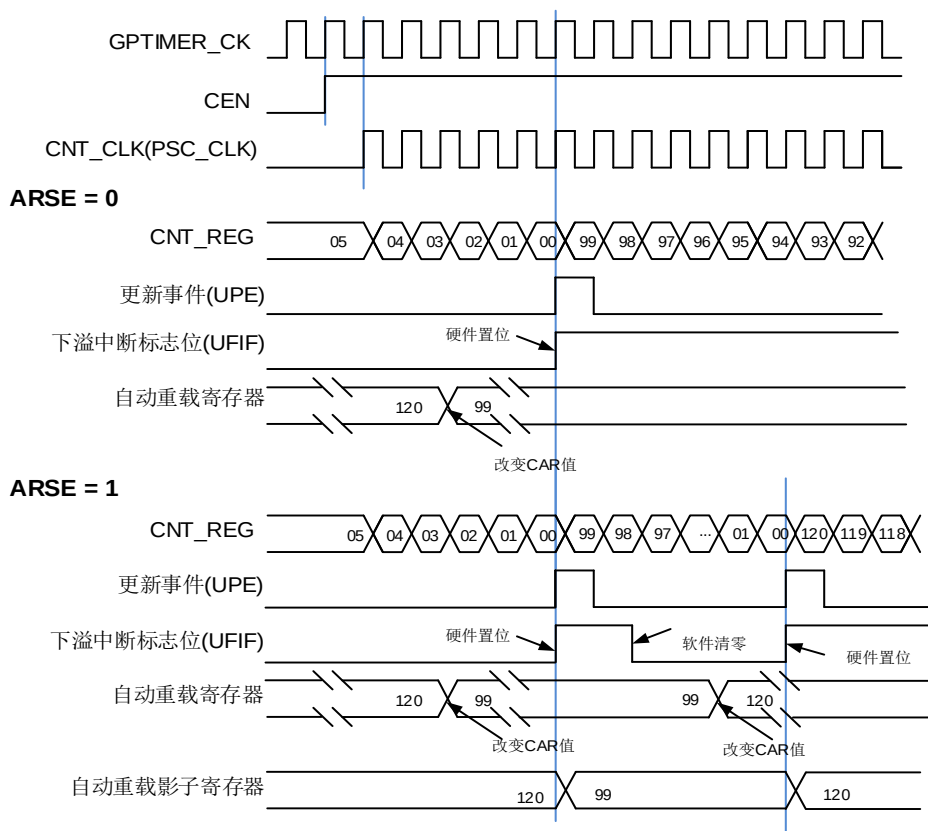
图 21-77. 向下计数时序图, $PSC = 0/2$ 

图 21-78. 向下计数时序图, 在运行时改变 GPTIMERx_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（GPTIMERx_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，GPTIMERx_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。

当由 GPTIMERx_REESEL 选定的硬件复位源发生，或者选定了软件复位源并且设置了 GPTIMERx_SWRST 寄存器的相应位时，计数器值将被初始化为 0，并在中心对齐计数模式下，无论计数器是向上还是向下计数，都会生成一个更新事件。

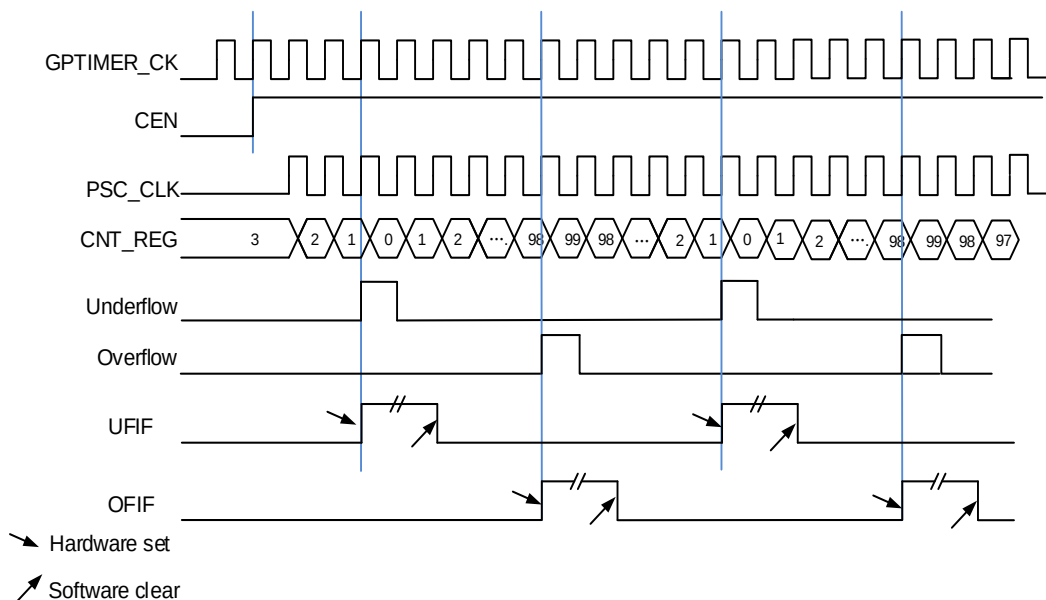
如果 GPTIMERx_CTL0 寄存器的 UPDIS 置 1，则禁止更新事件。

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。

在输出比较模式或 ADC 触发模式中，用于比较匹配的比较寄存器和附加比较寄存器（CHxCV/CHxCOMVAL_ADD(x=0,1)），以及专用于生成 ADC 触发的两个比较寄存器（GPTIMERx_ADCCR1 / GPTIMERx_ADCCR2），也由全局更新功能控制。[影子寄存器加载功能](#)介绍了影子功能的详细信息。

[图 21-79. 中央计数模式计数器时序图](#)给出了一些例子，当 GPTIMERx_CAR = 0x99，GPTIMERx_PSC = 0x0 时，计数器的行为。

图 21-79. 中央计数模式计数器时序图



22.4.5. 事件计数模式

计数器可以对 GPTIMERx_CUPEVSSEL 和 GPTIMERx_CDNEVSSEL 寄存器选择的硬件源进行事件计数。当这两个寄存器被配置，GPTIMERx_CTL0 寄存器中的 CKSSEL[2:0]位选择的时钟源和 GPTIMERx_CTL0 中的 DIR 位选择的计数方向无效。如果同时发生上升计数源选

择事件和下降计数源选择事件，计数器的值保持不变。

上溢事件和下溢事件的生成方式与周期计数相同。

GPTIMERx_CTL0 中的 CEN 位被设置为通过硬件源启动事件计数，计时器将在一个 CK_GPTIMER 周期后启动。

在事件计数模式下，DIR 的值由硬件控制，当一个向上计数事件发生时置为 0，当下一个向下计数事件发生时置为 1。当在事件计数模式下发生更新事件时，GPTIMERx_CNT 的值会根据当前的 DIR 值进行重置。

外部脉冲宽度测量

定时器可以测量通道输入或 ETI 输入的脉冲宽度。要测量的脉冲（高/低电平，计数方向）是通过 GPTIMERx_CUPEVSSEL 寄存器中的 INTLCUPS[3:0]位或 GPTIMERx_CDNEVSSEL 寄存器中的 INTLCDWS[3:0]位来选择的。

向上计数和向下计数脉冲禁止选择相同的。

在此模式下，时钟源由 GPTIMERx_CTL0 寄存器中的 CKSSEL[2:0]位选择，计数周期由 GPTIMERx_CAR 设置，计数器的初始值由 GPTIMERx_CNT 设置。

22.4.6. 计数方向

计数器的方向由配置 GPTIMERx_CTL0 寄存器中的 DIR 位来指定。

在边沿对齐模式下，如果在计数过程中改变 DIR，计数方向将在上溢（向上计数时）或下溢（向下计数时）时改变。当 GPTIMERx_CTL0 寄存器中的 FDIR 位置 0，如果 DIR 在计数停止时改变，计数方向将在计数重新启动后的上溢/下溢时改变。当 GPTIMERx_CTL0 寄存器中的 FDIR 位置 1，计数方向将在计数重新启动时立即改变。

在中心对齐模式下，如果在计数过程中 DIR 被改变，计数方向不会改变。当 GPTIMERx_CTL0 寄存器中的 FDIR 位置 0，如果计数停止时 DIR 被改变，计数方向不会改变。当 GPTIMERx_CTL0 寄存器中的 FDIR 位置 1，一旦计数重新启动，计数方向就会改变。

当从边缘对齐模式切换到中心对齐模式时，只有在计数器达到 0 时，计数方向才会强制向上；只有在计数器达到周期值时，计数方向才会强制向下。

22.4.7. 重复计数器

重复计数器用于在 (N+1) 个周期后生成更新事件，更新定时器的寄存器，其中 N 是 GPTIMERx_CREP 寄存器中 CREP 位的值。在向上计数模式中，每次计数器上溢时重复计数器会递减，在向下计数模式中，每次计数器下溢时重复计数器会递减，在中央对齐计数模式中，每次计数器下溢时重复计数器会递减。当 GPTIMERx_CREP 寄存器的 CREPCNT 值减少到 0 时，GPTIMERx_INTF 寄存器中的 RCENDIF 位置位，并且如果 GPTIMERx_DMAINTEN 寄存器中的相应使能位置位，则生成相应的中断信号。

当重复计数器被启用且 GPTIMER 计数器停止时，CREPCNT 保持当前值，并且在计数器重新

启动时，CREPCNT 从当前值继续递减。

图 21-80. 中央计数模式下计数器重复时序图

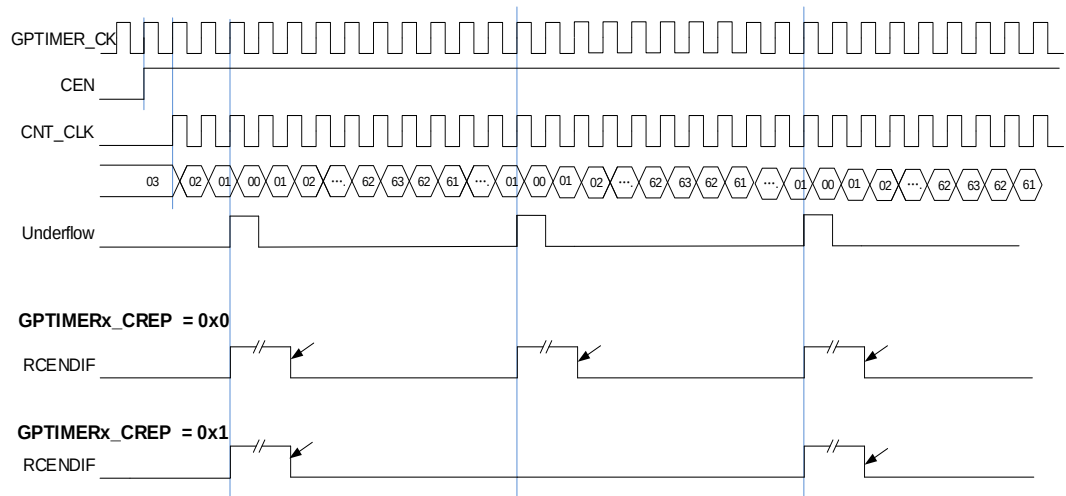


图 21-81. 在向上计数模式下计数器重复时序图

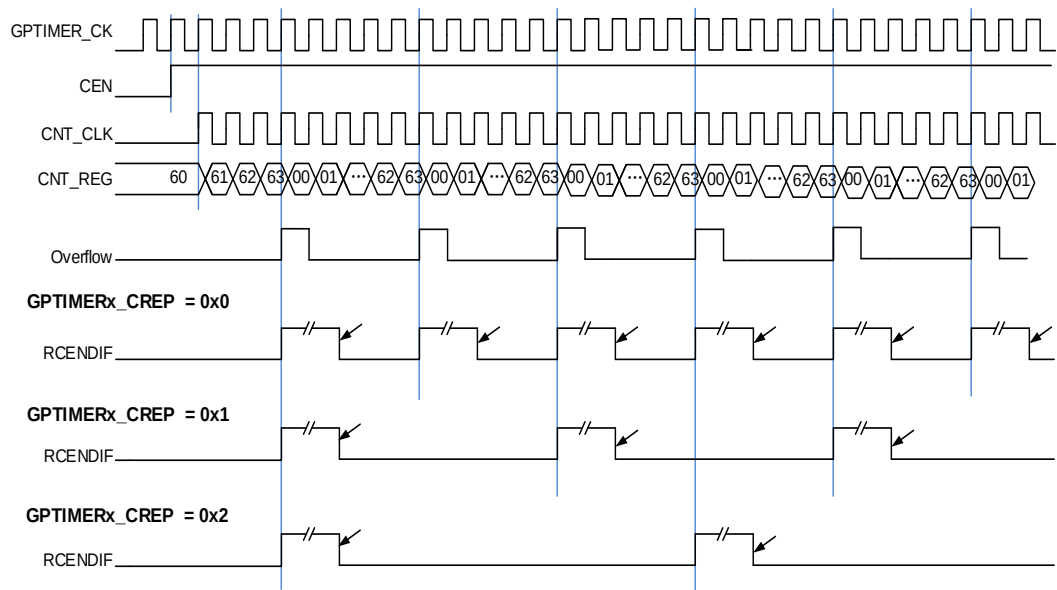
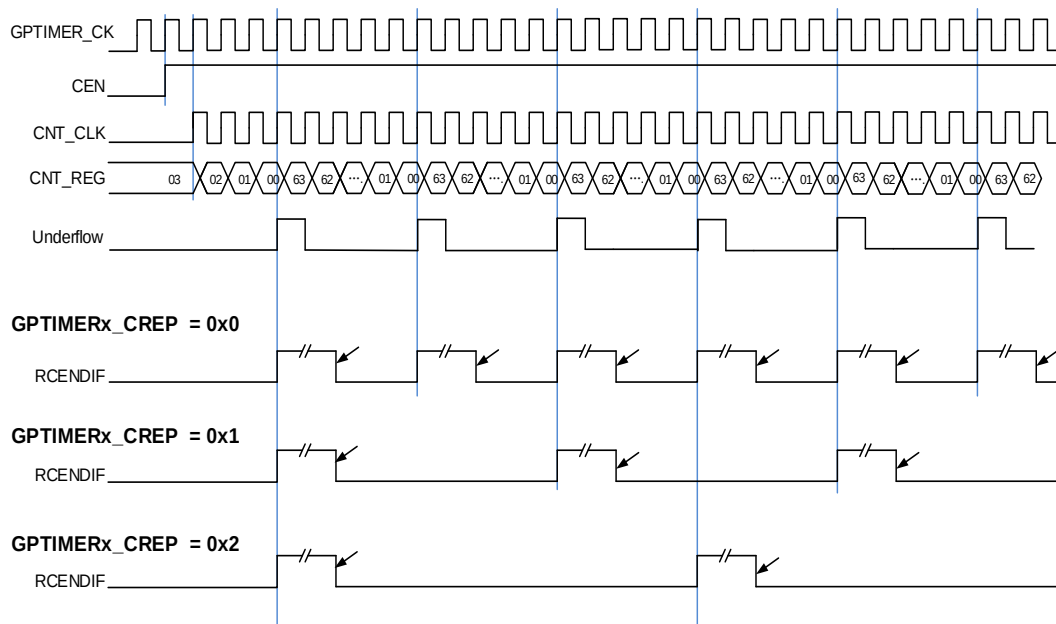


图 21-82. 在向下计数模式下计数器重复时序图



22.4.8. 影子寄存器加载功能

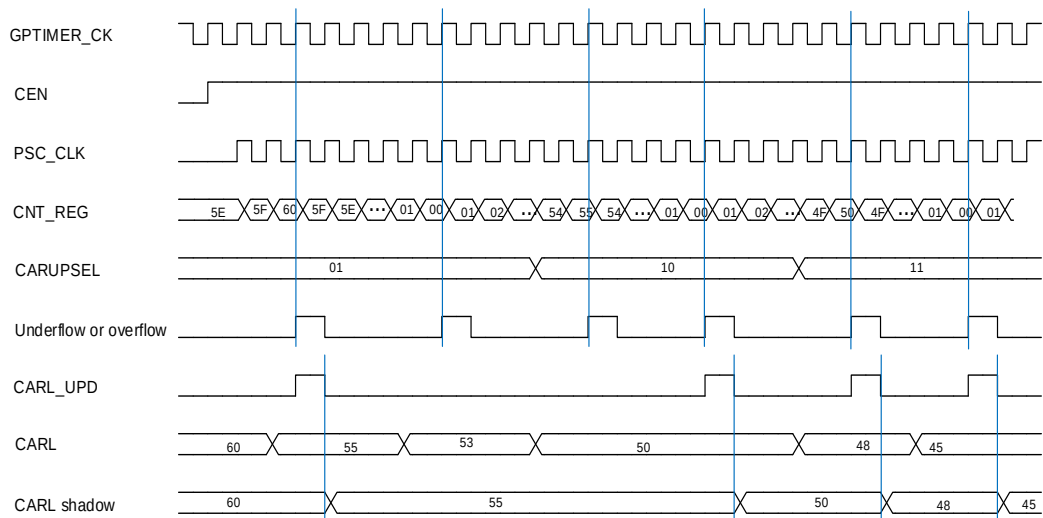
GPTIMERx_CAR、GPTIMERx_CHxCV、GPTIMERx_CHxCOMV_ADD、GPTIMERx_ADCCR1 和 GPTIMERx_ADCCR2 ($x = 0, 1$) 都具有独立的影子功能启用位。当影子功能被禁用时，具有影子寄存器功能的相应寄存器将立即被新配置的值覆盖，并且定时器将使用新配置的值进行操作。具有影子寄存器功能的寄存器同时在 GPTIMERx_UPSSEL 寄存器中具有单独的 2 位更新模式选择。当启用影子功能时，每个影子寄存器可以选择启用全局更新功能、上溢更新、下溢更新或上下溢更新。全局更新功能可以在设定的时间实现所选影子寄存器统一更新。

全局一次性重载模式

如果在 GPTIMERx_UPSSEL 寄存器中的 GLBUPSEL[1:0] $\neq 2'b11$ ，则启用全局一次性重载更新。通过 GLBUPSEL[1:0] 位选择一个统一的时刻加载影子寄存器值。

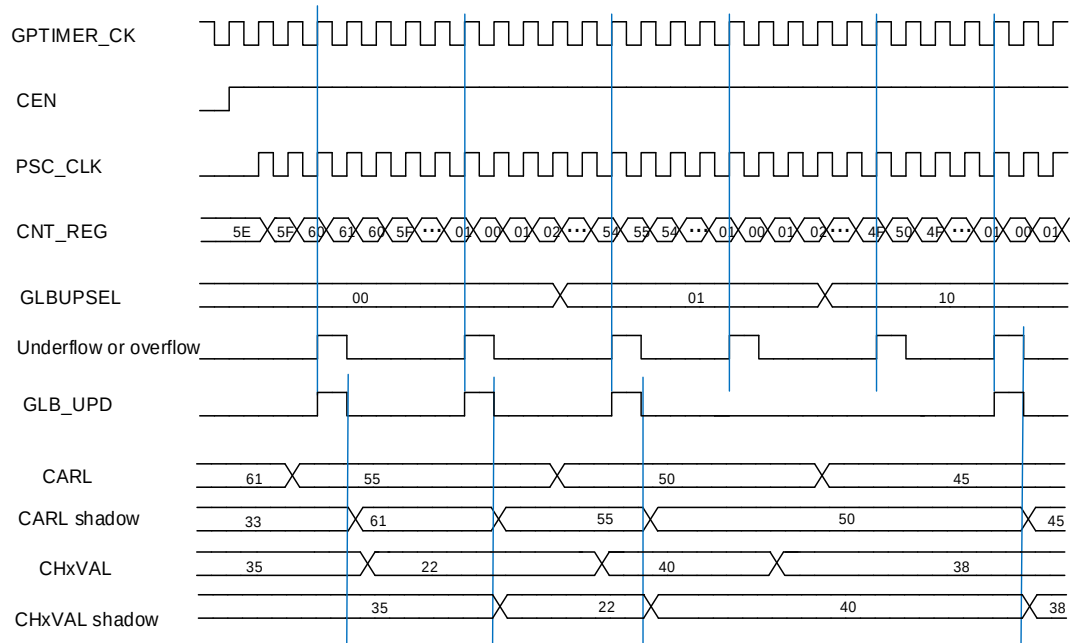
如 [图 21-83. 非全局一次性重载模式下更新事件的更新时序图](#) 所示，当更新事件发生时，CAR 不使用全局一次性重载模式 (CARUPSEL[1:0] $\neq 2'b00$)，并启用自动重载影子寄存器 (ARSE=1)。当 CARUPSEL[1:0] = 2'b01 时，CAR 在下一个上溢时刻重载；当 CARUPSEL[1:0] = 2'b10 时，CAR 在下一个下溢时刻重载；当 CARUPSEL[1:0] = 2'b11 时，CAR 在下一个上溢或下溢时刻重载。

图 21-83. 非全局一次性重载模式下更新事件的更新时序图



如 [图 21-84. 全局一次性重载模式下更新事件的更新时序图 \(GLBUPSEL\[1:0\] ≠ 2'b11\)](#) 所示，显示了在全局更新模式下的 CARL 和 CHxCV(x = 0..1) 更新时刻图 (ARSE = 1 和 CHxCOMSEN = 1)。

图 21-84. 全局一次性重载模式下更新事件的更新时序图 (GLBUPSEL[1:0] ≠ 2'b11)



如 [图 21-85. 全局一次性重载模式下更新事件的更新时序图 \(GLBUPSEL\[1:0\] = 2'b11\)](#) 所示，显示了 CARL 和 CHxCV(x = 0..1) 使能了全局更新模式，但禁能一次性全局更新模式时的更新时序图，CARL 和 CHxCV(x = 0..1) 将在下一个计数器复位时重载 (CARUPSEL[1:0] = 2'b00, CHxCVUPSEL[1:0] = 2'b00, ARSE = 1, CHxCOMSEN = 1)。

The timing diagram illustrates the operation of the GPTIMER module. The signals shown are:

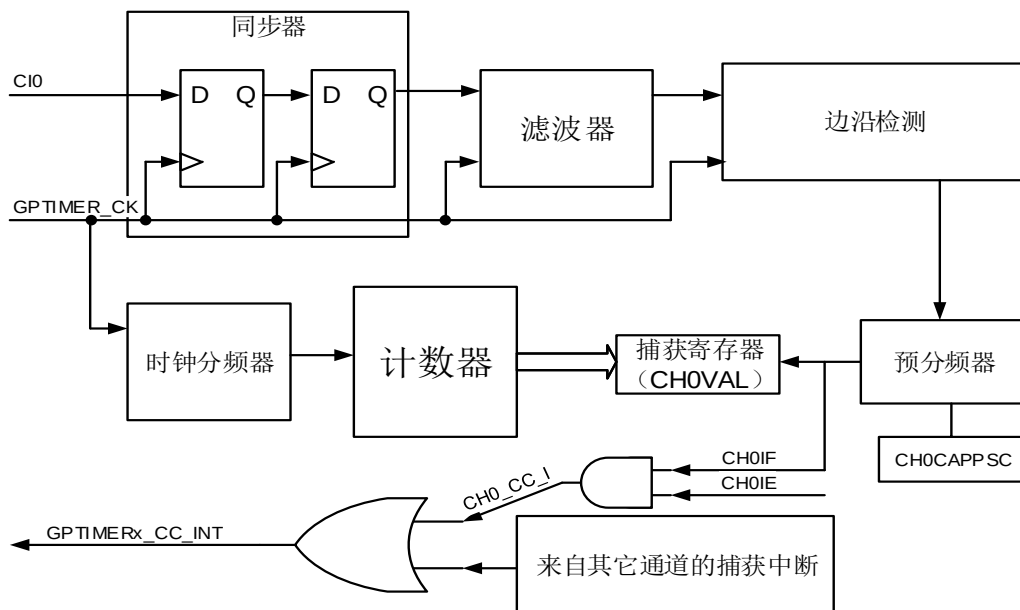
- GPTIMER_CK**: The main clock signal, shown as a periodic square wave.
- CEN**: The enable signal, which transitions from low to high at the start of the timer operation.
- PSC_CLK**: The prescaler clock signal, shown as a periodic square wave with a higher frequency than GPTIMER_CK.
- CNT_REG**: The counter register value, shown as a sequence of hexadecimal values: 5E, 5F, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 6A, 6B, 6C, 6D, 6E, 6F, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 7A, 7B, 7C, 7D, 7E, 7F, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 8A, 8B, 8C, 8D, 8E, 8F, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 9A, 9B, 9C, 9D, 9E, 9F, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E, 2F, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 3A, 3B, 3C, 3D, 3E, 3F, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 4A, 4B, 4C, 4D, 4E, 4F, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 5A, 5B, 5C, 5D, 5E, 5F, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 6A, 6B, 6C, 6D, 6E, 6F, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 7A, 7B, 7C, 7D, 7E, 7F, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 8A, 8B, 8C, 8D, 8E, 8F, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 9A, 9B, 9C, 9D, 9E, 9F, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E, 2F, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 3A, 3B, 3C, 3D, 3E, 3F, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 4A, 4B, 4C, 4D, 4E, 4F, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 5A, 5B, 5C, 5D, 5E, 5F, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 6A, 6B, 6C, 6D, 6E, 6F, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 7A, 7B, 7C, 7D, 7E, 7F, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 8A, 8B, 8C, 8D, 8E, 8F, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 9A, 9B, 9C, 9D, 9E, 9F, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E, 2F, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 3A, 3B, 3C, 3D, 3E, 3F, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 4A, 4B, 4C, 4D, 4E, 4F, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 5A, 5B, 5C, 5D, 5E, 5F, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 6A, 6B, 6C, 6D, 6E, 6F, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 7A, 7B, 7C, 7D, 7E, 7F, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 8A, 8B, 8C, 8D, 8E, 8F, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 9A, 9B, 9C, 9D, 9E, 9F, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E, 2F, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 3A, 3B, 3C, 3D, 3E, 3F, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 4A, 4B, 4C, 4D, 4E, 4F, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 5A, 5B, 5C, 5D, 5E, 5F, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 6A, 6B, 6C, 6D, 6E, 6F, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 7A, 7B, 7C, 7D, 7E, 7F, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 8A, 8B, 8C, 8D, 8E, 8F, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 9A, 9B, 9C, 9D, 9E, 9F, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E, 2F, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 3A, 3B, 3C, 3D, 3E, 3F, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 4A, 4B, 4C, 4D, 4E, 4F, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 5A, 5B, 5C, 5D, 5E, 5F, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 6A, 6B, 6C, 6D, 6E, 6F, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 7A, 7B, 7C, 7D, 7E, 7F, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 8A, 8B, 8C, 8D, 8E, 8F, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 9A, 9B, 9C, 9D, 9E, 9F, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E, 2F, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 3A, 3B, 3C, 3D, 3E, 3F, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 4A, 4B, 4C, 4D, 4E, 4F, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 5A, 5B, 5C, 5D, 5E, 5F, 60, 61, 62, 63, 64, 65, 66, 67,

通用定时器拥有 2 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。

输入捕获模式允许通道执行诸如脉冲定时、频率、周期、占空比等测量。输入阶段包括数字滤波器、边沿检测和通道预分频器。当通道输入上发生选定的边沿时，当前计数器的值被捕获到 GPTIMERx_CHxCV (x=0,1) 寄存器中，同时设置 CHxIF (x=0,1) 位，并在 CHxIE=1 (x=0,1) 时生成通道中断（如果启用）。

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，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，GPTIMERx_CHxCV (x=0,1) 寄存器会捕获计数器当前的值，同时 CHxIF 位置 1，如果 ChxIE = 1，则产生通道中断。

图 21-86. 通道 0 输入捕获逻辑



通道输入信号源由 GPTIMERx_CH0CSSEL 和 GPTIMERx_CH1CSSEL 寄存器选择，包括外部输入信号 ETI0/1/2/3、通道输入信号 channel0/1 以及 EVIC 事件源。ETI 和通道可以选择有效的边沿（上升沿、下降沿、上升/下降沿）。

输入捕获预分频器使得多个输入事件生成一个有效的捕获事件。在捕获事件上，GPTIMERx_CHxCV 将存储计数器的值。

配置步骤如下：

步骤 1：滤波器配置（GPTIMERx_CHnCTL(x=0,1 和 n=0,1)寄存器中的 CHxCAPFLT[3:0]位域）。根据输入信号和请求信号的质量，配置 CHxCAPFLT[3:0]。

步骤 2：捕获源选择（GPTIMERx_CHnCSSEL 寄存器）。一旦通过 GPTIMERx_CHnCSSEL 寄存器选择了一个输入捕获源，通道必须设置为输入模式，且 GPTIMERx_CHxCV 不能再被写入了。

步骤 3：中断使能（GPTIMERx_DMAINTEN 寄存器中的 CHxIE 和 CHxDEN 位）。使能相关位以获取中断和 DMA 请求。

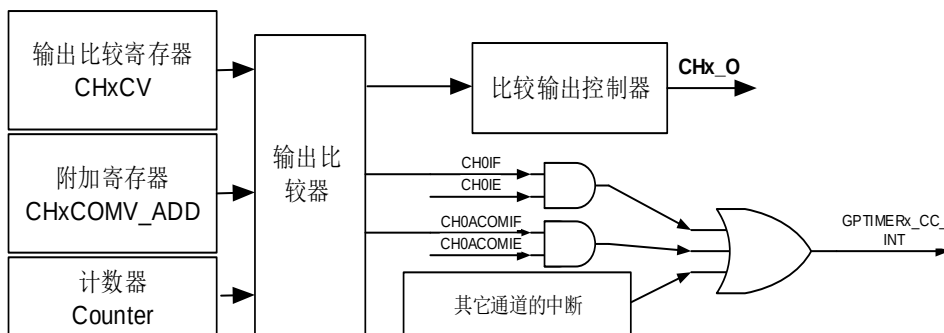
步骤 4：捕获使能（GPTIMERx_CHCTL 寄存器中的 CHxEN 位）。

当捕获到所需的输入信号时，GPTIMERx_CHxCV 将被设置成当前计数器的值，并且 CHxIF 位置位。中断和 DMA 请求将根据 GPTIMERx_DMAINTEN 寄存器中 CHxIE 和 CHxDEN 位的配置决定。

输出比较模式

[图 21-87. 输出比较逻辑](#)显示了输出比较模式的逻辑电路。

图 21-87. 输出比较逻辑



在输出比较模式下，GPTIMERx 可以生成具有可编程位置、极性、持续时间和频率的定时脉冲。当计数器匹配到输出比较通道的 GPTIMERx_CHxCV / GPTIMERx_CHxCOMV_ADD 寄存器中的值时，基于 GPTIMERx_CHnCTL 寄存器相关位，channel n 的输出可以被置高电平、低电平、保持或翻转。当计数器达到 GPTIMERx_CHxCV / GPTIMERx_CHxCOMV_ADD 寄存器中的值时，CHxIF / CH1COMADDIF 位将被置位，如果 CHxIE / CHxCOMADDIE = 1，则生成 channel n 中断。如果 CHxDEN / CH0COMADDEN = 1，则会产生 DMA 请求。

配置步骤如下：

步骤 1：时钟配置。例如时钟源、预分频器等。

步骤 2：比较模式配置。

- 通过 CHxCOMSEN / CH0COMADDSSEN 配置输出比较影子寄存器；
- 通过 GPTIMERx_CHnCTL[7:0]和 GPTIMERx_CHnCTL[18:16]位设置输出模式；
- 使能 CHxEN 的输出。

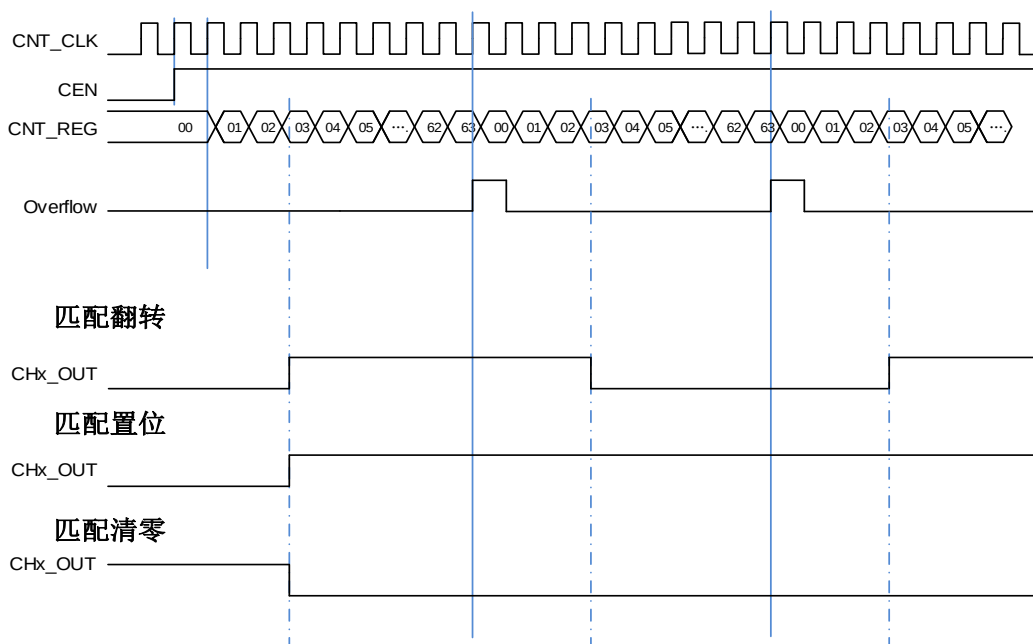
步骤 3：通过 CHxIE / CHxCOMADDIE / CHxDEN 配置中断/DMA 请求使能。

步骤 4：通过 GPTIMERx_CAR、GPTIMERx_CHxCV 和 GPTIMERx_CHxCOMV_ADD 寄存器配置比较输出时基。GPTIMERx_CHxCV / GPTIMERx_CHxCOMV_ADD 可以根据需要改变以满足预期的波形。

步骤 5：设置 CEN 位使能定时器。

图 21-88. 三种输出比较模式显示了三种比较模式：翻转、置位、清零电平。CARL=0x63, CHxVAL=0x3, CH0COMVAL_ADD 无效。

图 21-88. 三种输出比较模式



22.4.10. PWM 模式

在 PWM 输出模式下，通道可以根据 GPTIMERx_CAR 寄存器和 GPTIMERx_CHxCV/GPTIMERx_CHxCOMV_ADD 寄存器输出 PWM 波形。

CHxCV 和 CHxCOMV_ADD 的值应在 0-CARL 的范围内，当 CHxCV / CHxCOMV_ADD = 0 或 CARL 时，在一个周期内只实现一次比较匹配，如果 CHxCV / CHxCOMV_ADD 配置的值超出范围，则不发生比较匹配。

根据 GPTIMERx_CHCTL 寄存器中的 CH1MSEL 位选择两个通道独立输出或互补输出，死区时间由 GPTIMERx_DTCTL 寄存器控制。

基于计数模式，PWM 还可以分为 EAPWM（边沿对齐 PWM）和 CAPWM（中央对齐 PWM）。

如果 GPTIMERx_CTL1 寄存器中的 CCSOEN 位置位，与 PWM 周期同步的 CPPO 信号将输出到 GPIO。CPPO 的初始值为 0，当计数器被启用时，CPPO 变为 1，并且在溢出/下溢/计数器被重置时 CPPO 被翻转。只有在至少一个通道处于输出模式时，才会产生 CPPO。

通道的输出电平由 GPTIMERx_CHnCTL 寄存器的[18:16]位和[7:0]位控制，当计数器启动时（高电平/低电平），在周期结束时，CHxCV 与匹配比较或 CHxCOMV_ADD 与匹配比较时，输出电平被置高/置低/保持/翻转。通过上述配置，可以实现所有常用的 PWM 波形，包括 PWM 模式 0/1、复合 PWM 模式 0/1、非对称 PWM 模式 0/1、0% 占空比和 100% 占空比波形。

例如，复合 PWM 模式周期由 $2 \times \text{GPTIMERx\_CAR}$ 决定，占空比由 CHxCV / CHxCOMV_ADD 决定。[图 21-89. 复合 PWM 时序图](#)显示了复合 PWM 模式的输出波形图。复合 PWM0 模式可以通过 GPTIMERx_CHnCTL 寄存器中的 CH0COMDW[1:0] = 2'b10, CH0COMUP[1:0] = 2'b01, CH0ACOMDW[1:0] = 2'b01, CH0COMUP[1:0] = 2'b10 位实现，复合 PWM1 模式的配置也是类似的。

例如,非对称 PWM 模式周期由 $2 \times \text{GPTIMERx\_CAR}$ 决定,占空比由 $\text{CHxCV} / \text{CHxCOMV\_ADD}$ 决定。[图 21-90. 非对称 PWM 时序图](#)显示了非对称 PWM 模式的输出波形图。非对称 PWM0 模式可以通过设置 GPTIMERx_CHnCTL 寄存器中的 $\text{CH0COMDW}[1:0] = 2'b01$, $\text{CH0COMUP}[1:0] = 2'b00$, $\text{CH0ACOMDW}[1:0] = 2'b10$, $\text{CH0COMUP}[1:0] = 2'b00$ 位来实现,非对称 PWM1 模式的配置也是类似的。

客户还可以根据需要配置不同的波形,包括边沿对齐模式。

图 21-89. 复合 PWM 时序图

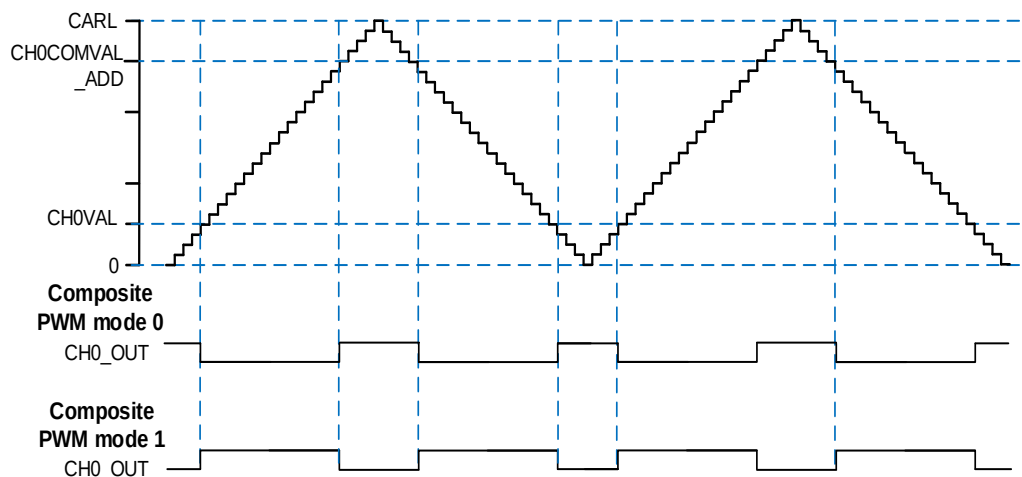
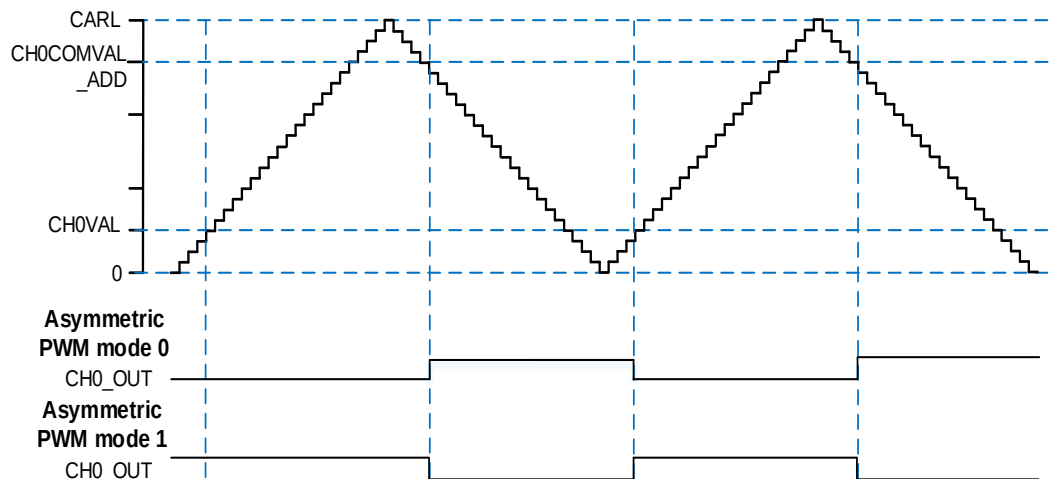


图 21-90. 非对称 PWM 时序图



22.4.11. 互补输出和死区时间插入

两个通道输出包括两种情况:

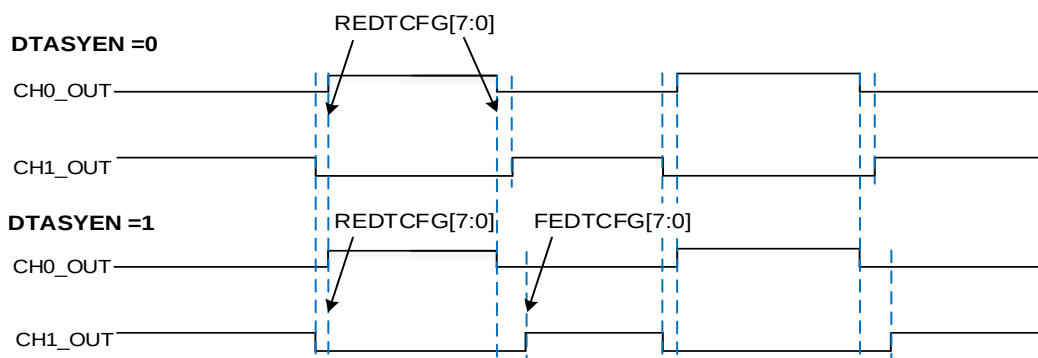
- 当 GPTIMERx_CHCTL 寄存器中的 CH1MSEL 位被复位时,通道 0 和通道 1 独立输出 PWM 波形;
- 当 GPTIMERx_CHCTL 寄存器中的 CH1MSEL 位被置位时,通道 0 和通道 1 配合输出互补的 PWM 波形。

当输出互补波形时,两个通道不能同时输出相同电平。通过配置 GPTIMERx_DMAINTEN 寄存

器中 CHHOUTDET 位和 CHLOUTDET 位，可使得能通道的同时输出高/低电平检测功能。

当 GPTIMERx_DMAINTEN 寄存器中的 DTEN 位被设置时，使能死区时间插入。插入死区时间是为了确保两个通道的互补输出不会同时处于高/低电平，相关的死区时间配置通过 GPTIMERx_DTCTL 寄存器实现。可以通过 GPTIMERx_DTCTL 寄存器中的 DTASYEN 位选择上升沿和下降沿是否使用不对称的死区时间。当 DTASYEN=1 时，上升沿和下降沿的死区时间分别由 REDTCFG[7:0]位和 FEDTCFG[7:0]位的值配置。DTASYEN 位必须在使能计数器之前配置，并且在 CEN=1 时不能修改。

图 21-91. 非对称死区时间



两个死区时间寄存器都有影子寄存器。当设置 REDTSEN 位和 FEDTSEN 位时，REDTCFG[7:0] 位和 FEDTCFG[7:0]位的值将预加载到相应的影子寄存器中。在下一个计数器复位事件时，预加载的值将更新到寄存器中。当在计数过程中启用影子寄存器时，将应用当前的死区时间值，并丢弃新配置的值。

注意：当死区时间大于输出波形的占空比时，该占空比消失，并且 CHx_OUT 输出低电平。

22.4.12. 占空比 0%或 100%输出

可以通过在 GPTIMERx_CHnCTL 寄存器中配置 CHxOUTM[1:0]位来定义一个具有 0%或 100% 占空比的波形输出。

在边沿对齐模式下，如果在计数过程中更改了 CHxOUTM[1:0]位，则在上溢（向上计数）或下溢（向下计数）时输出波形的占空比会发生变化。当计数器停止时，如果 CHxOUTCTL = 0 且更改了 CHxOUTM[1:0]位，则在计数开始后的上溢（向上计数）或下溢（向下计数）时输出波形的占空比会发生变化；当计数器停止时，如果 CHxOUTCTL = 1 且更改了 CHxOUTM[1:0]位，则在计数开始后立即改变输出波形的占空比。

在中央对齐模式下，如果在计数过程中更改了 CHxOUTM[1:0]位，则在下溢时输出波形的占空比会发生变化。当计数器停止时，如果 CHxOUTCTL = 0，并且在停止后更改了 CHxOUTM[1:0]位，则在计数开始后的下溢时输出波形的占空比会发生变化；如果计数器停止时 CHxOUTCTL = 1，并且更改了 CHxOUTM[1:0]位，则在计数开始后立即改变输出波形的占空比。

当输出 0%或 100%波形时，比较匹配中断仍然产生。

当输出波形的占空比从 0%或 100%释放时，输出电平值由 GPTIMERx_CHnCTL 寄存器中的 CHxREDUTY 位和 CHxENDOUT[1:0]位确定。当 CHxENDOUT[1:0] = 2'b01 时，在周期结束

时输出低电平。当 $CHxENDOUT[1:0] = 2'b10$ 时，在周期结束时输出高电平。当 $CHxENDOUT[1:0] = 2'b00$ 或 $2'b11$ 时，在周期结束时保持或翻转电平，并由 $CHxREDUTY$ 选择基准。当 $CHxREDUTY = 0$ 时，以 0% 或 100% 为基准，当 $CHxREDUTY = 1$ 时，以最后一次比较匹配的结果为基准。如 [表 21-12. 波形输出示意](#) 所示：

表 21-12. 波形输出示意表

CHxENDOUT[1:0]	被占空比0%或100%覆盖的比较匹配时电平值	CHxREDUTY在0%占空比时		CHxREDUTY在100%占空比时	
		0	1	0	1
00: 输出保持	0	0	0	1	0
	1	0	1	1	1
01: 输出低电平	-	0	0	0	0
10: 输出高电平	-	1	1	1	1
11: 输出翻转	0	1	1	0	1
	1	1	0	0	0

注意：若在两通道互补输出且使能死区插入时，插入死区会导致最终输出的占空比不是 100% 或 0%，使用时需注意配合。

22.4.13. GPTIMERx 之间的同步

软件源同步

GPTIMERx 之间的计数器同步使能/禁能/复位行为可以通过同时在 $GPTIMERx\_SWEN$ / $GPTIMERx\_SWDIS$ / $GPTIMERx\_SWRST$ 寄存器中将对应 GPTIMER 编号的位配置为 1 来指定。在启动计数器之前，可以通过预设 CNT 值来实现相位偏移。如 [图 21-92. 软件同步操作无初始值](#) 和 [图 21-93. 软件同步操作有初始值](#) 所示：

图 21-92. 软件同步操作无初始值

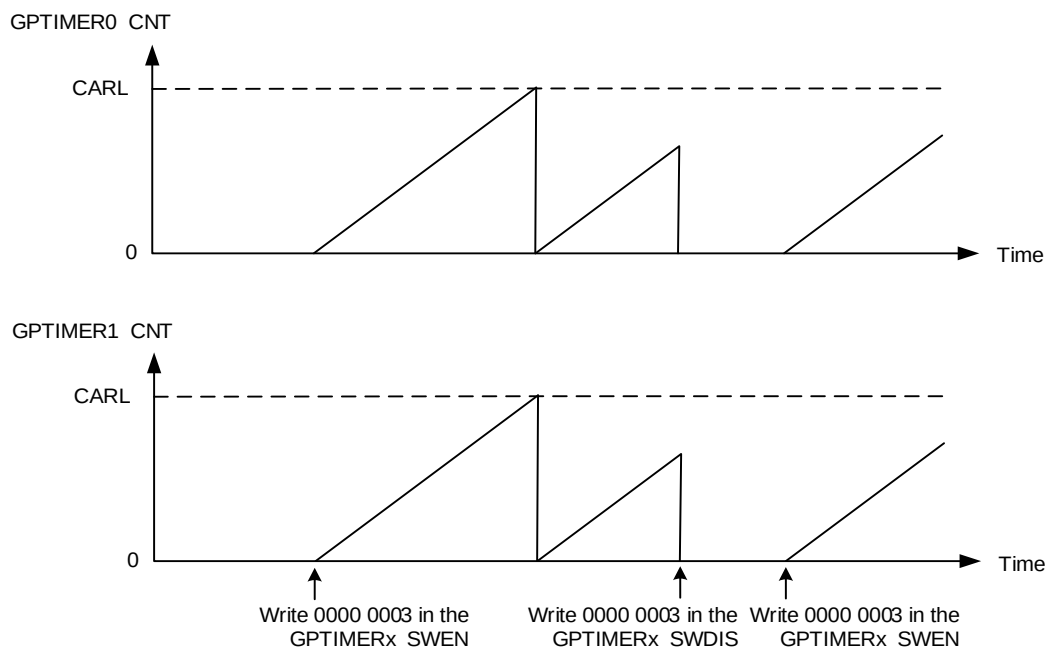
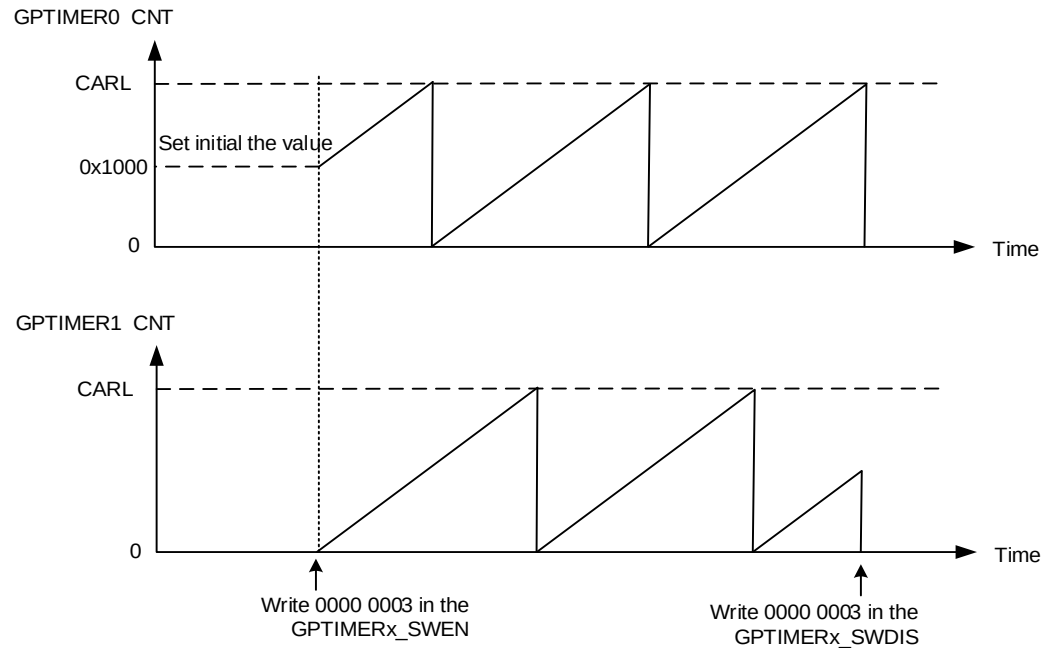


图 21-93. 软件同步操作有初始值

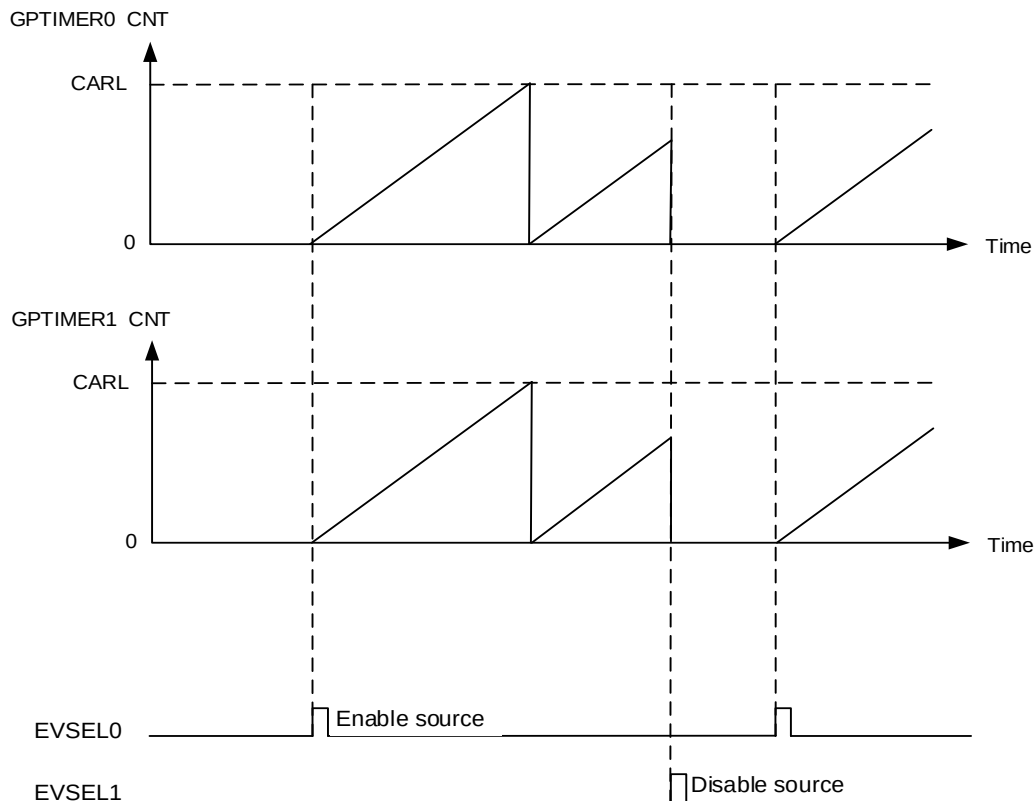


注意：如果同步操作的 GPTIMER 时钟分频系数不同，则计数器行为的生效时间也不同

硬件源同步

硬件同步源包括：ETI 输入、EVIC 输入和通道输入。由通道输入触发的同步行为是通过将通道输入捕获设置为 EVIC 源来实现的。如 [图 21-94. 硬件同步操作源](#) 所示：

图 21-94. 硬件同步操作源



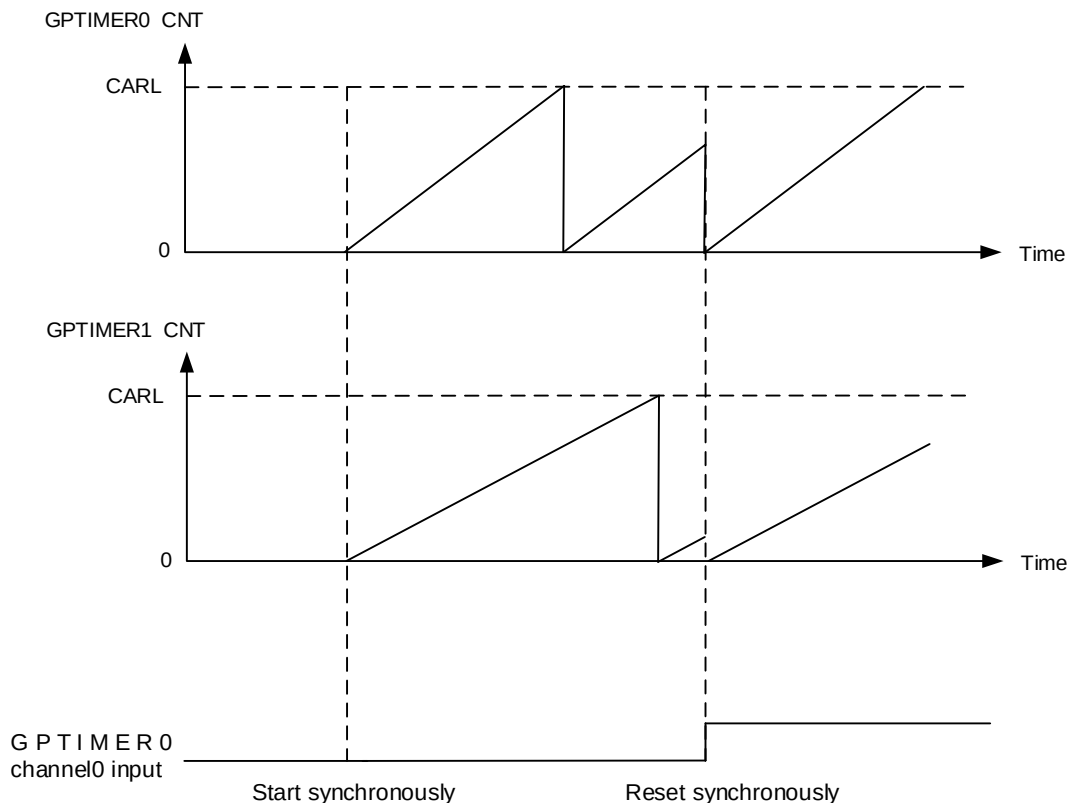
同步复位

一个 GPTIMER 的计数器复位行为可以触发其他 GPTIMER 的同步复位，计数器复位可以由比较匹配、输入捕获、上溢/下溢或在 GPTIMERx_RSSEL 寄存器中选定的输入源触发。将生成同步复位触发的 GPTIMER 和需要同步复位的 GPTIMER 设置在同一个组中，也可以实现 GPTIMER 之间的同步复位行为（选择相同的集合，配置相应的 GPTIMER 中的 SYNCSSSEL 位在 GPTIMERx_SYNRRSTCTL 寄存器中）。寄存器配置如下：

- GPTIMER 复位源：配置 GPTIMERx_SYNRRSTCTL 寄存器，选择同步重置源；
- 同步 GPTIMER：在 GPTIMERx_RSSEL 寄存器中配置 CSIRSTCNT[2:0] = 3'b101；
- 将源 GPTIMER 和目标 GPTIMER 设置为同一集合：在相应的 GPTIMER 的 GPTIMERx_SYNRRSTCTL 寄存器中配置 SYNCSSSEL 位写相同值。

当源 GPTIMER 产生一个复位事件时，目标 GPTIMER 会同时被复位，如[图 21-95. 通过通道同步复位源](#)所示。

图 21-95. 通过通道同步复位源



22.4.14. 单脉冲模式

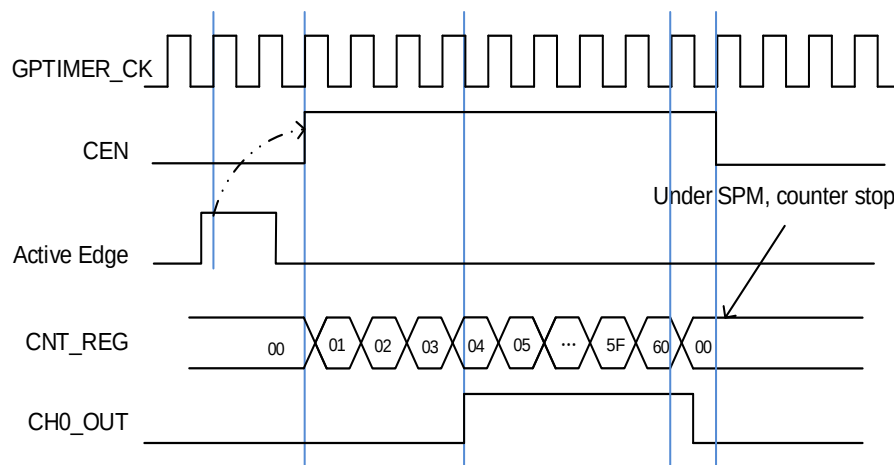
单脉冲模式与重复模式是相反的，设置 GPTIMERx_CTL0 寄存器的 SPM 位置 1，可启用单脉冲模式。当 SPM 置 1，计数器在下次更新事件到来后清零并停止计数。

一旦计时器设置为单脉冲模式，将 CEN 位置为 1 或触发信号沿都可以产生一个脉冲，然后保持 CEN 位在高电平状态，直到更新事件发生或软件将 CEN 位写为 0。如果 CEN 位被软件清除为 0，则计数器将停止，其值将保持。如果 CEN 位通过硬件/软件更新事件自动清 0，则计数器将重新初始化。硬件在周期结束时关闭 CEN 具有最高优先级。

在单脉冲模式下，有效的外部触发边沿将 CEN 位设置为 1，从而启用计数器。在单脉冲模式下，当触发上升沿发生后，如果计数值与 CHxCV 值比较，CHx_OUT 将立即根据通道输出配置改变。如 [图 21-96. 单脉冲模式，GPTIMERx_CHxCV=0x04, GPTIMERx_CAR=0x60, CH0COMUP\[1:0\] = 2'b10](#) 所示。

图 21-96. 单脉冲模式，GPTIMERx_CHxCV=0x04, GPTIMERx_CAR=0x60,

CH0COMUP[1:0] = 2'b10



22.4.15. 中断跳过功能

中断跳过

可以通过配置 GPTIMERx_IADCTSS 寄存器来设置要跳过的计数器上溢/下溢中断的数量。CHxCV 捕获/比较事件中断、CHxCOMV_ADD 比较事件中断和 ADC 触发请求信号可以链接到上溢/下溢中断跳过功能，实现中断跳过。

跳过的中断不会更新寄存器 GPTIMERx_INTF 中的相应标志位。中断跳过功能仅与寄存器 GPTIMERx_IADCTSS 有关，与 GPTIMERx_DMAINTEN 中的中断使能位无关，后者仅控制未跳过的中断信号是否使能。

在边沿对齐模式下，上溢（向上计数）和下溢（向下计数）将被跳过，而中心对齐模式的上溢/下溢将由 GPTIMERx_IADCTSS 寄存器中的 OUFINTSSEL[1:0]位选择，并且跳过的数目选择应与跳过中断源的选择结合使用。边沿对齐模式下改变计数方向时尤其需要注意。如 [图 21-97. 中心对齐模式下中断跳过的示例（跳过值为 2）](#)和 [图 21-98. 边沿对齐模式下中断跳过的示例（改变方向和跳过值为 2）](#)所示。

当计数器停止计数时，跳过计数器会被重置。只有在中断跳过计数器停止时（OUFINTSSEL = 2'b00），才能更改中断跳过值。

图 21-97. 中心对齐模式下中断跳过的示例（跳过值为 2）

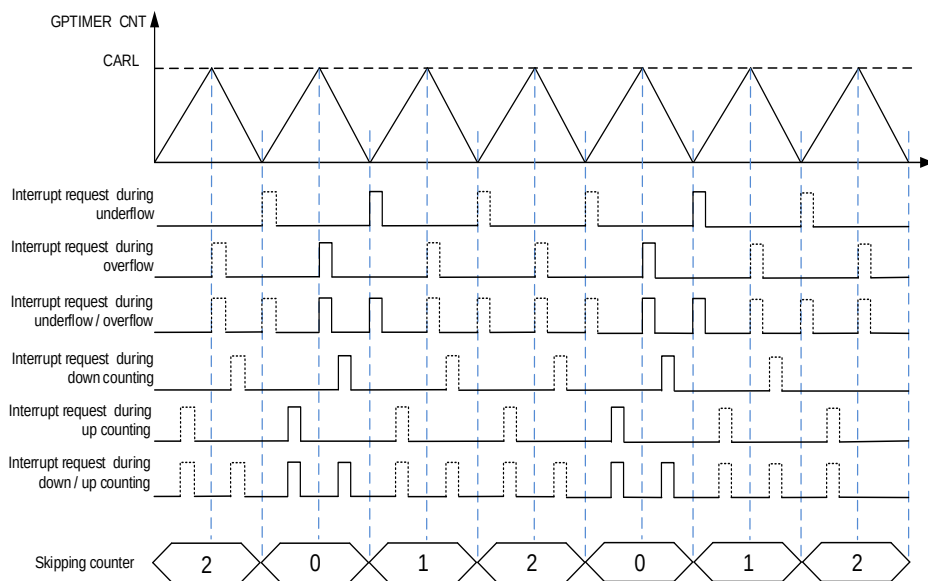
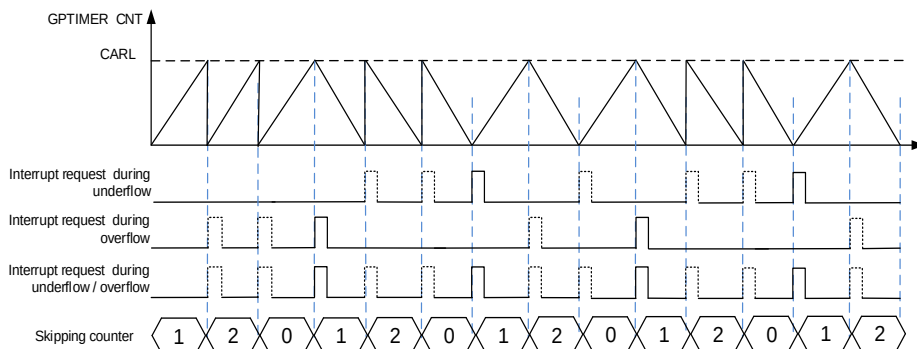


图 21-98. 边沿对齐模式下中断跳过的示例（改变方向和跳过值为 2）



附加中断跳过

通过配置 `GPTIMERx_ADDINTSCTL0` 和 `GPTIMERx_ADDINTSCTL1` 寄存器实现附加中断跳过功能。附加中断跳过具有两个独立的重复计数器，通过 `GPTIMERx_ADDINTSCTL0` 寄存器可以选择两个重复计数器的计数源、初始值和跳过次数，附加中断跳过重复计数器对 `GPTIMER` 计数器的上溢/下溢中断进行计数。两个计数器都会递增。

`CHxCV` 捕获/比较事件中断、`CHxCOMV_ADD` 比较事件中断、`ADC` 触发请求信号以及 `GPTIMER` 计数器上溢/下溢中断的扩展跳过功能是通过 `GPTIMERx_ADDINTSCTL1` 寄存器选择的。跳过的中断对应的 `EVIC` 事件也将被跳过。跳过的中断不会更新寄存器 `GPTIMERx_INTF` 中相应的标志位。中断跳过功能独立于 `GPTIMERx_DMAINTEN` 中的中断使能位，后者仅控制未跳过的中断信号是否使能。

当 `CNT` 停止计数时，附加中断跳过的两个计数器保留它们的当前值。附加中断跳过值只能在附加中断跳过计数器停止时更改（`ADDCxSKSEL[1:0] = 2'b00`）。在不跳过状态变为跳过状态（`ADDCxSKSEL[1:0]` 从 `00` 变为其他值）后的第一个计数时钟，`ADDCxVAL[2:0]` 加 1。初始值寄存器 `ADDCxIVAL[2:0]` 只有在 `ADDCxSKSEL[1:0] ≠ 2'b00` 时才能写入，并且在写入

ADCCxSKSEL[1:0] = 2'b00 时不会被重置。

ADC 触发请求中断跳过

ADC 触发请求的跳过功能是通过配置寄存器 GPTIMERx_ADCTL 和 GPTIMERx_ADCTRGS 实现的。ADC 触发请求跳过有两个独立的重复计数器，通过寄存器 GPTIMERx_ADCTRGS 可以选择两个重复计数器的计数源、初始值和跳过次数，ADC 触发请求跳过重复计数器通过与 ADTCV1/2[15:0] 的比较匹配事件计数。

当 CNT 停止计数时，跳过计数器保留其当前值。ADC 触发请求跳过值只能在 ADC 触发请求跳过计数器停止时更改 (ADCCxSKSEL[1:0]=00)。ADCC2VAL[1:0]+1 是 ADC 触发请求从未跳过变为跳过后第一个计数时钟 (ADCCxSKSEL[1:0] 从 00 变为其他值)。初始值寄存器 ADCCxIVAL[2:0] 只有在 ADCCxSKSEL[1:0] 从 00 配置为其他值时才能被写入，并且在写入 ADCCxSKSEL[1:0]=2'b00 时不会被重置。

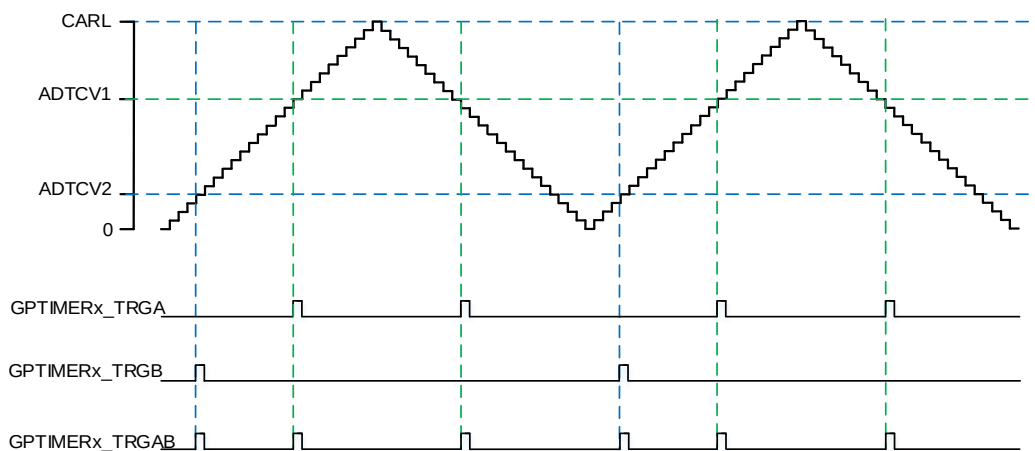
22.4.16. ADC 触发请求

ADC 比较匹配触发模式

ADC 触发请求信号可以通过 GPTIMER 比较匹配事件单独或组合触发 ADC。在事件计数模式下，不能生成任何 ADC 触发请求。

通用定时器有两个 ADC 触发比较值寄存器 (GPTIMERx_ADCCRy, x=0 或 1, y=1 或 2)，当比较匹配发生时，它们会生成 ADC 触发请求信号 GPTIMERx_TRGA 和 GPTIMERx_TRGB。匹配使能寄存器 (ADT1DWGTEN, ADT1UPGTEN, ADT2DWGTEN, ADT2UPGTEN) 可以配置，在向上计数、向下计数、向上计数+向下计数过程中生成或不生成 ADC 触发请求信号。如 [图 21-99. ADC 触发信号产生在 ADC 不同比较匹配触发模式 \(ADT1UPGTEN=1, ADT2UPGTEN=1, ADT1DWGTEN=1, ADT2DWGTEN=0\)](#) 所示。

图 21-99. ADC 触发信号产生在 ADC 不同比较匹配触发模式 (ADT1UPGTEN=1, ADT2UPGTEN=1, ADT1DWGTEN=1, ADT2DWGTEN=0)



GPTIMERx_ADCCRy 寄存器的 ADTCV1/2[15:0] 位域有对应的影子寄存器，根据影子寄存器的相关配置，更新比较寄存器。在中央对齐模式下，ADC 触发比较值寄存器的更新由

GPTIMERx_UPSSEL 寄存器中的 ADCCRnUPSEL[1:0]位域控制。

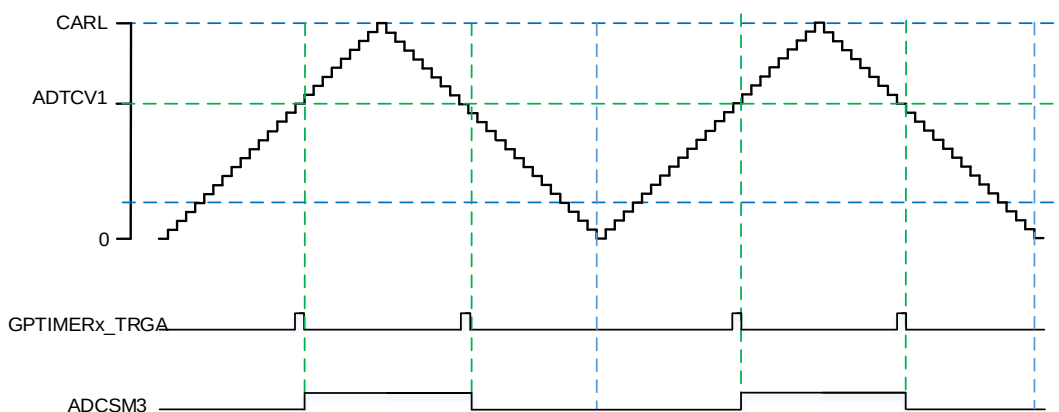
表 21-13. GPTIMER0/1 ADC 触发器信号及其操作

ADC触发信号	产生模式
GPTIMERx_TRGA	与ADC触发器比较值1 (ADTCV1)匹配
GPTIMERx_TRGB	与ADC触发器比较值2 (ADTCV2)匹配
GPTIMERx_TRGAB	与ADC触发器比较值1或2(ADTCV1或ADTCV2)匹配

监控 ADC 触发请求信号

外部引脚 ADCSM3/4 用于监控 ADC 触发请求信号。通过 GPTIMERx_ADCTL 寄存器中的 ADCMS3/4EN 位选择是否使能两个监控输出 ADCSM3/4。通过 ADCMS3/4SEL 位, 选择 ADCSM3/4 要监控的 ADC 触发请求, 并输出到外部引脚。当使能关闭时, 输出为低电平, 每次生成 ADC 触发请求时监控信号翻转一次。如[图21-100. ADC 触发器同步监控器信号时序图 \(ADCSM3SEL = 0\)](#)所示。

图 21-100. ADC 触发器同步监控器信号时序图(ADCSM3SEL = 0)



22.4.17. 与 EVIC 的交互

GPTIMER 事件输出

GPTIMER 可以通过 EVIC 与其他模块交互。输出到 EVIC 的中断信号不受 GPTIMERx_DMAINTEN 寄存器中相应使能位的影响。GPTIMER 输出的 EVIC 事件如下:

- CH0CV 比较匹配事件
- CH1CV 比较匹配事件
- CH0COMV_ADD 比较匹配事件
- CH1COMV_ADD 比较匹配事件
- GPTIMER 计数器上溢事件
- GPTIMER 计数器下溢事件

- 由 ADTCV1 匹配生成的 ADC 触发事件
- 由 ADTCV2 匹配生成的 ADC 触发事件
- 重复计数结束事件

GPTIMER 接收 EVIC 事件

GPTIMER 接收到 EVIC 事件有八个，分别为 EVSELn (n=0...7)，根据配置寄存器，GPTIMER 的行为如下：

- 计数器使能、禁能、复位
- 向上计数、向下计数
- 输入捕获

22.4.18. 保护功能

寄存器写保护

为了防止误写寄存器，所有寄存器都具有写保护功能，这是通过 GPTIMERx_WP 寄存器实现的。三个寄存器 GPTIMERx_SWEN、GPTIMERx_SWDIS 和 GPTIMERx_SWRST 具有独立的位 SWEWPN、SWDWPEN 和 SWRWPN 用于写保护。所有其他寄存器都由 WPEN 位进行写保护。

若要更改是否使能写保护，需要将指定的序列“0xA5”写入 WPKEY[15:8]位域。

停止输出保护

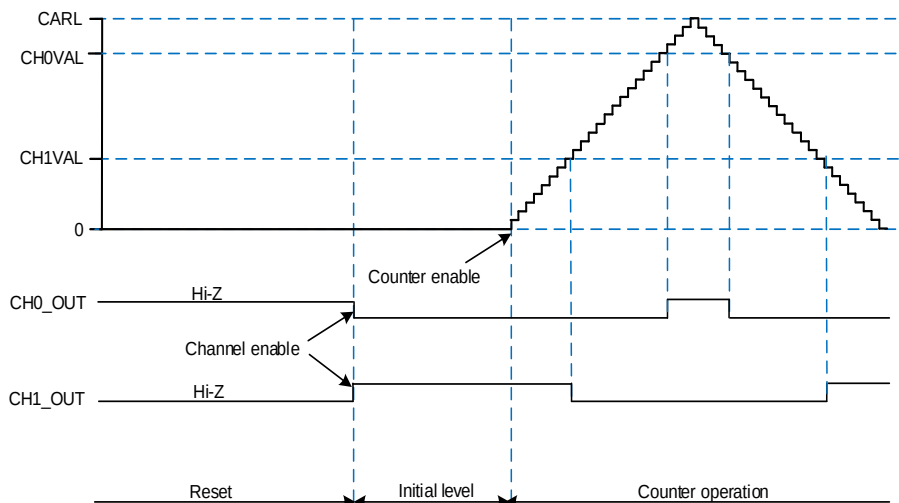
当从 GTOC 接收到停止输出请求时，GPTIMER 的输出可以被强制停止在端口上。当两个通道同时输出相同电平的信号时，输出停止保护非常有用。在接收到停止输出请求时，可以通过 GPTIMERx_CH0CTL 中的 CH0INVSOUT[1:0] 位和 GPTIMERx_CH1CTL 中的 CH1INVSOUT[1:0]位来选择 GPTIMER 的输出电平。

当在周期结束时释放停止输出请求，停止输出将生效，直到下一个计数周期结束。

22.4.19. 输出引脚初始化

GPTIMER 寄存器在复位后被初始化，输出端口由 GPTIMERx_CHnCTL 寄存器中的 CHnENDOUT[1:0]位和 CHnENOUT 位初始化。如 [图 21-101. 复位后的通道输出](#)所示，复位后的通道输出，CHnENDOUT = 2'b11，CH0ENOUT = 0，CH1ENOUT = 1。

图 21-101. 复位后的通道输出



如果在 GPTIMER 工作时出现错误，有几种方法可以设置输出：

- 将 GPTIMERx_CHnCTL 寄存器中的 CHnOUT 位置位，输出保持在当前电平；
- 复位 GPTIMERx_CHnCTL 寄存器中的 CHnOUT 位，在计数器停止时选择 CHnDISOUT 位在 GPTIMERx_CHnCTL 寄存器中设置的输出电平；
- 关闭 GPTIMERx_CHCTL 寄存器中的通道使能位 CHnEN，通过 GPIO 设置输出电平；
- 通过 GTOC 输出 Hi-Z 配置。

22.4.20. 定时器 DMA 模式

定时器 DMA 模式是通过 DMA 模块配置定时器寄存器的功能。有两个跟定时器 DMA 模式相关的寄存器：GPTIMERx_DMACFG 和 GPTIMERx_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。当中断事件发生时，GPTIMERx 会给 DMA 发送请求。DMA 配置为 M2P（内存到外设）模式，PADDR 是 GPTIMERx_DMATB 寄存器地址，DMA 就会访问 GPTIMERx_DMATB 寄存器。实际上，GPTIMERx_DMATB 寄存器仅是一个缓冲，定时器将 GPTIMERx_DMATB 映射到由 GPTIMERx_DMACFG 中的 DMATA 指定的内部寄存器。如果 GPTIMERx_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 GPTIMERx_DMACFG 寄存器的 DMATC 位域值不为 0，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 GPTIMERx_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。

如果再发生一次 DMA 请求事件，GPTIMERx 将重复上述过程。

22.4.21. 定时器调试模式

当 Cortex®-M33 停止时，如果 DBG_CTL 寄存器中的 GPTIMERx_HOLD 配置位设置为 1，则 GPTIMERx 计数器停止。

22.5. GPTIMERx 寄存器(x = 0, 1)

GPTIMER0 基地址: 0x4001 6000

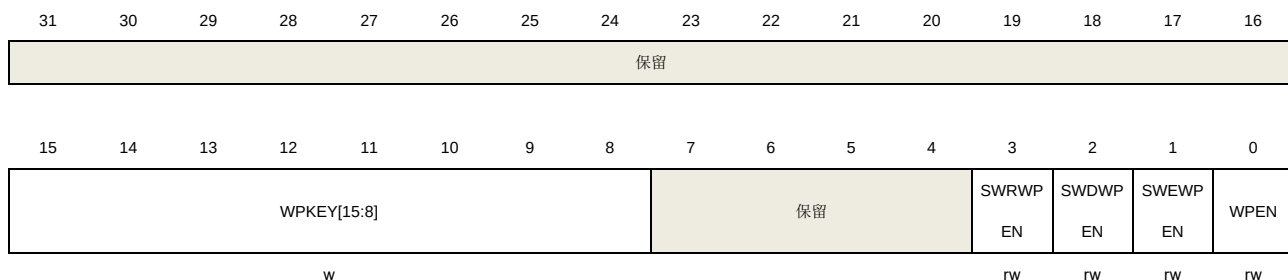
GPTIMER1 基地址: 0x4001 6100

22.5.1. 寄存器写保护(GPTIMERx_WP)

地址偏移: 0x00

复位值: 0x0000 0000

该寄存器只能按字（32 位）访问。



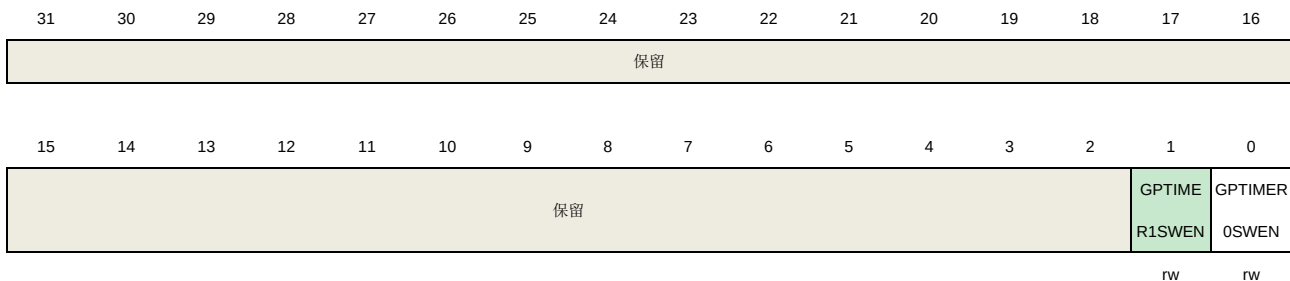
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:8	WPKEY[15:8]	写保护 如果要修改WPEN, SWEWPEN, SWDWPEN, SWRWPEN位, 需要该位域同时写入0xA5。
7:4	保留	必须保持复位值。
3	SWRWPEN	GPTIMERx_SWRST寄存器写保护使能 0: 写保护禁能 1: 写保护使能
2	SWDWPEN	GPTIMERx_SWDIS寄存器写保护使能 0: 写保护禁能 1: 写保护使能
1	SWEWPEN	GPTIMERx_SWEN寄存器写保护使能 0: 写保护禁能 1: 写保护使能
0	WPEN	寄存器写保护使能 目标寄存器是除SWEWPEN、SWDWPEN和SWRWPEN保护的寄存器之外所有寄存器。 0: 写保护禁能 1: 写保护使能

22.5.2. 软件使能寄存器(GPTIMERx_SWEN)

地址偏移：0x04

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



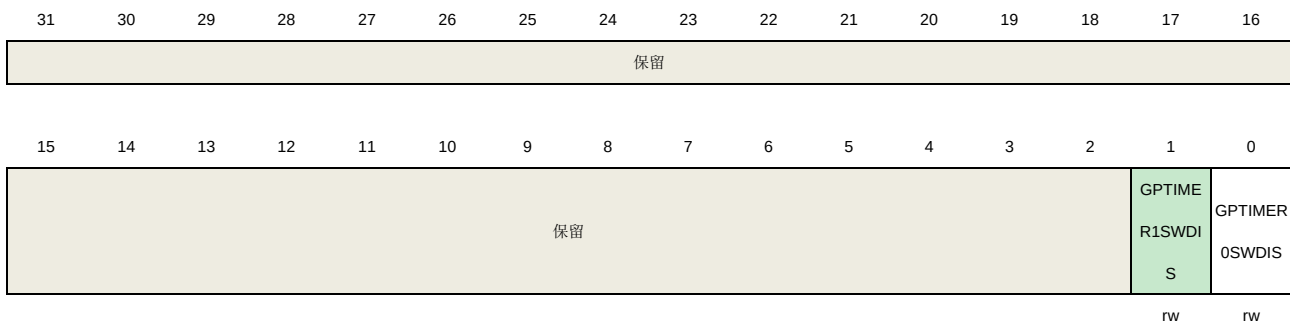
位/位域	名称	描述
31:2	保留	必须保持复位值。
1	GPTIMER1SWEN	GPTIMER1计数器软件使能 参考GPTIMER0SWEN描述。
0	GPTIMER0SWEN	GPTIMER0计数器软件使能 当读该位时： 0：计数器状态停止 1：计数器正在工作 当写该位时： 0：无影响 1：计数器开始工作

22.5.3. 软件禁能寄存器(GPTIMERx_SWDIS)

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

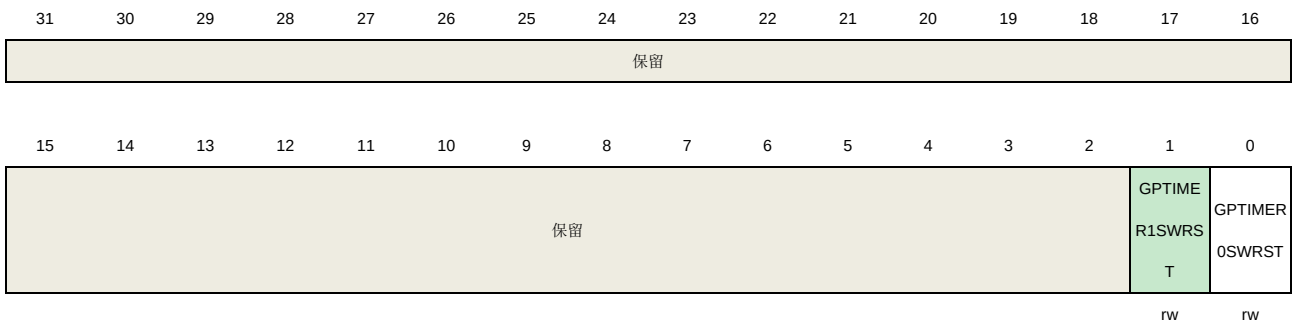
31:2	保留	必须保持复位值。
1	GPTIMER1SWDIS	GPTIMER1计数器软件禁能 参考GPTIMER0SWDIS描述。
0	GPTIMER0SWDIS	GPTIMER0计数器软件禁能 当读该位时： 0：计数器正在工作 1：计数器状态停止 当写该位时： 0：无影响 1：计数器停止工作

22.5.4. 软件复位寄存器(GPTIMERx_SWRST)

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
31:2	保留	必须保持复位值。
1	GPTIMER1SWRST	GPTIMER1计数器软件复位 参考GPTIMER0SWRST描述。
0	GPTIMER0SWRST	GPTIMER0计数器软件复位 0：无影响 1：计数器复位

22.5.5. 计数器使能源选择寄存器(GPTIMERx_ESSEL)

地址偏移：0x10

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



SWENCN	保留							EVSEL7E	EVSEL6E	EVSEL5E	EVSEL4E	EVSEL3E	EVSEL2E	EVSEL1E	EVSEL0E
T								NCNT	NCNT	NCNT	NCNT	NCNT	NCNT	NCNT	NCNT
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
保留				CH1ENCNT[1:0]		CH0ENCNT[1:0]		ETI3ENCNT[1:0]		ETI2ENCNT[1:0]		ETI1ENCNT[1:0]		ETI0ENCNT[1:0]	
				rw		rw		rw		rw		rw		rw	

位/位域	名称	描述
31	SWENCNT	软件信号使能计数器功能使能 0: 禁能 1: 使能
30:24	保留	必须保持复位值。
23	EVSEL7ENCNT	EVSEL7信号使能计数器功能使能 参考EVSEL0ENCNT描述。
22	EVSEL6ENCNT	EVSEL6信号使能计数器功能使能 参考EVSEL0ENCNT描述。
21	EVSEL5ENCNT	EVSEL5信号使能计数器功能使能 参考EVSEL0ENCNT描述。
20	EVSEL4ENCNT	EVSEL4信号使能计数器功能使能 参考EVSEL0ENCNT描述。
19	EVSEL3ENCNT	EVSEL3信号使能计数器功能使能 参考EVSEL0ENCNT描述。
18	EVSEL2ENCNT	EVSEL2信号使能计数器功能使能 参考EVSEL0ENCNT描述。
17	EVSEL1ENCNT	EVSEL1信号使能计数器功能使能 参考EVSEL0ENCNT描述。
16	EVSEL0ENCNT	EVSEL0信号使能计数器功能使能 0: 禁能 1: 使能
15:12	保留	必须保持复位值。
11:10	CH1ENCNT[1:0]	通道1输入信号边沿使能计数器功能使能 参考CH0ENCNT[1:0]描述。
9:8	CH0ENCNT[1:0]	通道0输入信号边沿使能计数器功能使能 00: 禁能 01: 上升沿 10: 下降沿 11: 上升沿/下降沿

7:6	ETI3ENCNT[1:0]	ETI3信号边沿使能计数器功能使能 参考ETI0ENCNT[1:0]描述。
5:4	ETI2ENCNT[1:0]	ETI2信号边沿使能计数器功能使能 参考ETI0ENCNT[1:0]描述。
3:2	ETI1ENCNT[1:0]	ETI1信号边沿使能计数器功能使能 参考ETI0ENCNT[1:0]描述。
1:0	ETI0ENCNT[1:0]	ETI0信号边沿使能计数器功能使能 00: 禁能 01: 上升沿 10: 下降沿 11: 上升沿/下降沿

22.5.6. 计数器禁能源选择寄存器(GPTIMERx_DSSEL)

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		
SWDISC	保留								EVSEL7D	EVSEL6D	EVSEL5D	EVSEL4D	EVSEL3D	EVSEL2D	EVSEL1D	EVSEL0D	
NT									ISCNT	ISCNT	ISCNT	ISCNT	ISCNT	ISCNT	ISCNT	ISCNT	
rw								rw		rw		rw		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
保留				CH1DISCNT[1:0]		CH0DISCNT[1:0]		ETI3DISCNT[1:0]		ETI2DISCNT[1:0]		ETI1DISCNT[1:0]		ETI0DISCNT[1:0]			
rw				rw		rw		rw		rw		rw		rw			

位/位域	名称	描述
31	SWDISCNT	软件信号禁能计数器使能 0: 禁能 1: 使能
30:24	保留	必须保持复位值。
23	EVSEL7DISCNT	EVSEL7信号禁能计数器功能使能 参考EVSEL0DISCNT描述。
22	EVSEL6DISCNT	EVSEL6信号禁能计数器功能使能 参考EVSEL0DISCNT描述。
21	EVSEL5DISCNT	EVSEL5信号禁能计数器功能使能 参考EVSEL0DISCNT描述。
20	EVSEL4DISCNT	EVSEL4信号禁能计数器功能使能 参考EVSEL0DISCNT描述。

19	EVSEL3DISCNT	EVSEL3信号禁能计数器功能使能 参考EVSEL0DISCNT描述。
18	EVSEL2DISCNT	EVSEL2信号禁能计数器功能使能 参考EVSEL0DISCNT描述。
17	EVSEL1DISCNT	EVSEL1信号禁能计数器功能使能 参考EVSEL0DISCNT描述。
16	EVSEL0DISCNT	EVSEL0信号禁能计数器功能使能 0: 禁能 1: 使能
15:12	保留	必须保持复位值。
11:10	CH1DISCNT[1:0]	通道1输入信号边沿禁能计数器功能使能 参考CH0DISCNT[1:0]描述。
9:8	CH0DISCNT[1:0]	通道0输入信号边沿禁能计数器功能使能 00: 禁能 01: 上升沿 10: 下降沿 11: 上升沿/下降沿
7:6	ETI3DISCNT[1:0]	ETI3信号边沿禁能计数器功能使能 参考ETI0DISCNT[1:0]描述。
5:4	ETI2DISCNT[1:0]	ETI2信号边沿禁能计数器功能使能 参考ETI0DISCNT[1:0]描述。
3:2	ETI1DISCNT[1:0]	ETI1信号边沿禁能计数器功能使能 参考ETI0DISCNT[1:0]描述。
1:0	ETI0DISCNT[1:0]	ETI0信号边沿禁能计数器功能使能 00: 禁能 01: 上升沿 10: 下降沿 11: 上升沿/下降沿

22.5.7. 计数器复位源选择寄存器(GPTIMERx_RSSEL)

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
SWRSTC	保留					CSIRSTCNT[2:0]		EVSEL7R	EVSEL6R	EVSEL5R	EVSEL4R	EVSEL3R	EVSEL2R	EVSEL1R	EVSEL0R
NT								STCNT	STCNT	STCNT	STCNT	STCNT	STCNT	STCNT	STCNT
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
保留				CH1RSTCNT[1:0]		CH0RSTCNT[1:0]		ETI3RSTCNT[1:0]		ETI2RSTCNT[1:0]		ETI1RSTCNT[1:0]		ETI0RSTCNT[1:0]	
				rw		rw		rw		rw		rw		rw	

位/位域	名称	描述
31	SWRSTCNT	软件信号复位计数器使能 0: 禁能 1: 使能
30:27	保留	必须保持复位值。
26:24	CSIRSTCNT[2:0]	比较匹配、同步复位或者输入捕获复位计数器使能 000: 禁能 001: CH1VAL比较匹配或输入捕获 010: CH0VAL比较匹配或输入捕获 010: CH1COMVAL_ADD比较匹配 011: CH0COMVAL_ADD比较匹配 101: 同步复位 其它: 保留
23	EVSEL7RSTCNT	EVSEL7信号复位计数器功能使能 参考EVSEL0RSTCNT描述。
22	EVSEL6RSTCNT	EVSEL6信号复位计数器功能使能 参考EVSEL0RSTCNT描述。
21	EVSEL5RSTCNT	EVSEL5信号复位计数器功能使能 参考EVSEL0RSTCNT描述。
20	EVSEL4RSTCNT	EVSEL4信号复位计数器功能使能 参考EVSEL0RSTCNT描述。
19	EVSEL3RSTCNT	EVSEL3信号复位计数器功能使能 参考EVSEL0RSTCNT描述。
18	EVSEL2RSTCNT	EVSEL2信号复位计数器功能使能 参考EVSEL0RSTCNT描述。
17	EVSEL1RSTCNT	EVSEL1信号复位计数器功能使能 参考EVSEL0RSTCNT描述。
16	EVSEL0RSTCNT	EVSEL0信号复位计数器功能使能 0: 禁能 1: 使能
15:12	保留	必须保持复位值。
11:10	CH1RSTCNT[1:0]	通道1输入信号边沿复位计数器功能使能 参考CH0RSTCNT[1:0]描述。

9:8	CH0RSTCNT[1:0]	通道0输入信号边沿复位计数器功能使能 00: 禁能 01: 上升沿 10: 下降沿 11: 上升沿/下降沿
7:6	ETI3RSTCNT[1:0]	ETI3信号边沿复位计数器功能使能 参考ETI0RSTCNT[1:0]描述。
5:4	ETI2RSTCNT[1:0]	ETI2信号边沿复位计数器功能使能 参考ETI0RSTCNT[1:0]描述。
3:2	ETI1RSTCNT[1:0]	ETI1信号边沿复位计数器功能使能 参考ETI0RSTCNT[1:0]描述。
1:0	ETI0RSTCNT[1:0]	ETI0信号边沿复位计数器功能使能 00: 禁能 01: 上升沿 10: 下降沿 11: 上升沿/下降沿

22.5.8. 通道 0 捕获源选择寄存器(GPTIMERx_CH0CSSEL)

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								EVSEL7C	EVSEL6C	EVSEL5C	EVSEL4C	EVSEL3C	EVSEL2C	EVSEL1C	EVSEL0C
								H0IC	H0IC	H0IC	H0IC	H0IC	H0IC	H0IC	H0IC
								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				CH1CH0IC[1:0]		CH0CH0IC[1:0]		ETI3CH0IC[1:0]		ETI2CH0IC[1:0]		ETI1CH0IC[1:0]		ETI0CH0IC[1:0]	
				rw		rw		rw		rw		rw		rw	

位/位域	名称	描述
31:24	保留	必须保持复位值。
23	EVSEL7CH0IC	EVSEL7信号作为通道0输入捕获源使能 参考EVSEL0CH0IC描述。
22	EVSEL6CH0IC	EVSEL6信号作为通道0输入捕获源使能 参考EVSEL0CH0IC描述。
21	EVSEL5CH0IC	EVSEL5信号作为通道0输入捕获源使能 参考EVSEL0CH0IC描述。
20	EVSEL4CH0IC	EVSEL4信号作为通道0输入捕获源使能

		参考EVSEL0CH0IC描述。
19	EVSEL3CH0IC	EVSEL3信号作为通道0输入捕获源使能 参考EVSEL0CH0IC描述。
18	EVSEL2CH0IC	EVSEL2信号作为通道0输入捕获源使能 参考EVSEL0CH0IC描述。
17	EVSEL1CH0IC	EVSEL1信号作为通道0输入捕获源使能 参考EVSEL0CH0IC描述。
16	EVSEL0CH0IC	EVSEL0信号作为通道0输入捕获源使能 0: 禁能 1: 使能
15:12	保留	必须保持复位值。
11:10	CH1CH0IC[1:0]	通道1输入信号边沿作为通道0输入捕获源使能 参考CH0CH0IC[1:0]描述。
9:8	CH0CH0IC[1:0]	通道0输入信号边沿作为通道0输入捕获源使能 00: 禁能 01: 上升沿 10: 下降沿 11: 上升沿/下降沿
7:6	ETI3CH0IC[1:0]	ETI3信号边沿作为通道0输入捕获源使能 参考ETI0CH0IC[1:0]描述。
5:4	ETI2CH0IC[1:0]	ETI2信号边沿作为通道0输入捕获源使能 参考ETI0CH0IC[1:0]描述。
3:2	ETI1CH0IC[1:0]	ETI1信号边沿作为通道0输入捕获源使能 参考ETI0CH0IC[1:0]描述。
1:0	ETI0CH0IC[1:0]	ETI0信号边沿作为通道0输入捕获源使能 00: 禁能 01: 上升沿 10: 下降沿 11: 上升沿/下降沿

22.5.9. 通道 1 捕获源选择寄存器(GPTIMERx_CH1CSSEL)

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								EVSEL7C	EVSEL6C	EVSEL5C	EVSEL4C	EVSEL3C	EVSEL2C	EVSEL1C	EVSEL0C
								H1IC	H1IC	H1IC	H1IC	H1IC	H1IC	H1IC	H1IC
								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				CH1CH1IC[1:0]		CH0CH1IC[1:0]		ETI3CH1IC[1:0]		ETI2CH1IC[1:0]		ETI1CH1IC[1:0]		ETI0CH1IC[1:0]	
				rw		rw		rw		rw		rw		rw	

位/位域	名称	描述
31:24	保留	必须保持复位值。
23	EVSEL7CH1IC	EVSEL7信号作为通道1输入捕获源使能 参考EVSEL0CH1IC描述。
22	EVSEL6CH1IC	EVSEL6信号作为通道1输入捕获源使能 参考EVSEL0CH1IC描述。
21	EVSEL5CH1IC	EVSEL5信号作为通道1输入捕获源使能 参考EVSEL0CH1IC描述。
20	EVSEL4CH1IC	EVSEL4信号作为通道1输入捕获源使能 参考EVSEL0CH1IC描述。
19	EVSEL3CH1IC	EVSEL3信号作为通道1输入捕获源使能 参考EVSEL0CH1IC描述。
18	EVSEL2CH1IC	EVSEL2信号作为通道1输入捕获源使能 参考EVSEL0CH1IC描述。
17	EVSEL1CH1IC	EVSEL1信号作为通道1输入捕获源使能 参考EVSEL0CH1IC描述。
16	EVSEL0CH1IC	EVSEL0信号作为通道1输入捕获源使能 0: 禁能 1: 使能
15:12	保留	必须保持复位值。
11:10	CH1CH1IC[1:0]	通道1输入信号边沿作为通道1输入捕获源使能 参考CH0CH1IC[1:0]描述。
9:8	CH0CH1IC[1:0]	通道0输入信号边沿作为通道1输入捕获源使能 00: 禁能 01: 上升沿 10: 下降沿 11: 上升沿/下降沿
7:6	ETI3CH1IC[1:0]	ETI3信号边沿作为通道1输入捕获源使能 参考ETI0CH1IC[1:0]描述。
5:4	ETI2CH1IC[1:0]	ETI2信号边沿作为通道1输入捕获源使能

参考ETI0CH1IC[1:0]描述。

3:2	ETI1CH1IC[1:0]	ETI1信号边沿作为通道1输入捕获源使能 参考ETI0CH1IC[1:0]描述。
1:0	ETI0CH1IC[1:0]	ETI0信号边沿作为通道1输入捕获源使能 00: 禁能 01: 上升沿 10: 下降沿 11: 上升沿/下降沿

22.5.10. 控制寄存器 0(GPTIMERx_CTL0)

地址偏移: 0x24

复位值: 0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	保留	CKESEL[1:0]	保留	保留	CKSSEL[2:0]	保留	保留	保留	保留	保留	保留	保留	保留	保留	CCSOEN
rw				rw				rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	保留	保留	保留	保留	保留	保留	保留	CKDIV[1:0]	ARSE	CAM	FDIR	DIR	SPM	保留	CEN
rw						rw		rw	rw	rw	rw	rw	rw	r	rw

位/位域 名称 描述

31:30	保留	必须保持复位值。
29:28	CKESEL[1:0]	当选择ETI _x (x = 0, 1, 2, 3)作为时钟源时, 选择有效边沿 00: 上升沿 01: 下降沿 10/11: 上升沿/下降沿
27	保留	必须保持复位值。
26:24	CKSSEL[2:0]	时钟源选择 000: CK_GPTIMER 001: ETI0 010: ETI1 011: ETI2 100: ETI3 101-111: 保留
23:17	保留	必须保持复位值。
16	CCSOEN	计数器周期同步信号的输出使能 0: 禁能 1: 使能

15:10	保留	必须保持复位值。
9:8	CKDIV[1:0]	时钟分频 通过软件配置CKDIV，规定定时器时钟（CK_GPTIMER）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: $f_{DTS} = f_{CK\_GPTIMER}$ 01: $f_{DTS} = f_{CK\_GPTIMER} / 2$ 10: $f_{DTS} = f_{CK\_GPTIMER} / 4$ 11: 保留
7	ARSE	自动重载影子寄存器使能 0: 禁能 GPTIMERx_CAR 寄存器的影子寄存器 1: 使能GPTIMERx_CAR寄存器的影子寄存器
6	CAM	计数器对齐模式选择 0: 无中央对齐模式（边沿对齐模式），DIR 位指定了计数方向。 1: 中央对齐模式。 注意：此位只能在CEN=0时修改。
5	FDIR	强制设置计数方向 0: 边沿对齐模式下，在上溢/下溢时根据DIR位的配置改变计数方向，在中央对齐模式下无效。 1: 在重新开始计数时，立即改变为DIR位配置的计数方向。 注意：只能在计数器停止时配置此位。
4	DIR	方向 0: 向上计数 1: 向下计数 注意：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	保留	必须保持复位值。
1	UPDIS	禁止更新 该位用来使能或禁能更新事件的产生。 0: 计数器更新事件使能。当计数器上溢/下溢（边沿对齐模式）时，计数器下溢（中央对齐模式）或其它计数器清零事件发生时，具有缓存的寄存器被装入它们的预装载值。 1: 计数器更新事件禁能。带有缓存的寄存器保持原有值，如果计数器清零事件发生，计数器、预分频器和重复计数器值被重新初始化。
0	CEN	计数器使能 0: 计数器禁能 1: 计数器使能

22.5.11. 向上计数事件源选择寄存器(GPTIMERx_CUPEVSSEL)

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				INTLCUPS[3:0]				EVSEL7C	EVSEL6C	EVSEL5C	EVSEL4C	EVSEL3C	EVSEL2C	EVSEL1C	EVSEL0C
								UPS	UPS	UPS	UPS	UPS	UPS	UPS	UPS
				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
保留				CH1CUPS[1:0]		CH0CUPS[1:0]		ETI3CUPS[1:0]		ETI2CUPS[1:0]		ETI1CUPS[1:0]		ETI0CUPS[1:0]	
				rw		rw		rw		rw		rw		rw	

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:24	INTLCUPS[3:0]	输入电平作为向上计数事件源功能使能 0000: 禁能 0010: 通道0输入低电平 0011: 通道0输入高电平 0100: 通道1输入低电平 0101: 通道1输入高电平 1000: ETI0输入低电平 1001: ETI0输入高电平 1010: ETI1输入低电平 1011: ETI1输入高电平 1100: ETI2输入低电平 1101: ETI2输入高电平 1110: ETI3输入低电平 1111: ETI3输入高电平 其它: 保留。
23	EVSEL7CUPS	EVSEL7作为向上计数事件源功能使能 参考EVSEL0CUPS描述。
22	EVSEL6CUPS	EVSEL6作为向上计数事件源功能使能 参考EVSEL0CUPS描述。
21	EVSEL5CUPS	EVSEL5作为向上计数事件源功能使能 参考EVSEL0CUPS描述。
20	EVSEL4CUPS	EVSEL4作为向上计数事件源功能使能 参考EVSEL0CUPS描述。
19	EVSEL3CUPS	EVSEL3作为向上计数事件源功能使能

参考EVSEL0CUPS描述。

18	EVSEL2CUPS	EVSEL2作为向上计数事件源功能使能 参考EVSEL0CUPS描述。
17	EVSEL1CUPS	EVSEL1作为向上计数事件源功能使能 参考EVSEL0CUPS描述。
16	EVSEL0CUPS	EVSEL0作为向上计数事件源功能使能 0: 禁能 1: 使能
15:12	保留	必须保持复位值。
11:10	CH1CUPS[1:0]	通道1输入信号边沿作为向上计数事件源功能使能 参考CH0CUPS[1:0]描述。
9:8	CH0CUPS[1:0]	通道0输入信号边沿作为向上计数事件源功能使能 00: 禁能 01: 上升沿 10: 下降沿 11: 上升沿/下降沿
7:6	ETI3CUPS[1:0]	ETI3信号边沿作为向上计数事件源功能使能 参考ETI0UPS[1:0]描述。
5:4	ETI2CUPS[1:0]	ETI2信号边沿作为向上计数事件源功能使能 参考ETI0UPS[1:0]描述。
3:2	ETI1CUPS[1:0]	ETI1信号边沿作为向上计数事件源功能使能 参考ETI0UPS[1:0]描述。
1:0	ETI0CUPS[1:0]	ETI0信号边沿作为向上计数事件源功能使能 00: 禁能 01: 上升沿 10: 下降沿 11: 上升沿/下降沿

22.5.12. 向下计数事件源选择寄存器(GPTIMERx_CDNEVSSEL)

地址偏移: 0x2C

复位值: 0x0000 FFFF

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				INTLCDWS[3:0]				EVSEL7C	EVSEL6C	EVSEL5C	EVSEL4C	EVSEL3C	EVSEL2C	EVSEL1C	EVSEL0C
								DWS	DWS	DWS	DWS	DWS	DWS	DWS	DWS
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

保留	CH1CDWS[1:0]	CH0CDWS[1:0]	ETI3CDWS[1:0]	ETI2CDWS[1:0]	ETI1CDWS[1:0]	ETI0CDWS[1:0]
	rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:28	保留	必须保持复位值。
27:24	INTLCDWS[3:0]	输入电平作为向下计数事件源功能使能 0000: 禁能 0010: 通道0输入低电平 0011: 通道0输入高电平 0100: 通道1输入低电平 0101: 通道1输入高电平 1000: ETI0输入低电平 1001: ETI0输入高电平 1010: ETI1输入低电平 1011: ETI1输入高电平 1100: ETI2输入低电平 1101: ETI2输入高电平 1110: ETI3输入低电平 1111: ETI3输入高电平 其它: 保留。
23	EVSEL7CDWS	EVSEL7作为向下计数事件源功能使能 参考EVSEL0CDWS描述。
22	EVSEL6CDWS	EVSEL6作为向下计数事件源功能使能 参考EVSEL0CDWS描述。
21	EVSEL5CDWS	EVSEL5作为向下计数事件源功能使能 参考EVSEL0CDWS描述。
20	EVSEL4CDWS	EVSEL4作为向下计数事件源功能使能 参考EVSEL0CDWS描述。
19	EVSEL3CDWS	EVSEL3作为向下计数事件源功能使能 参考EVSEL0CDWS描述。
18	EVSEL2CDWS	EVSEL2作为向下计数事件源功能使能 参考EVSEL0CDWS描述。
17	EVSEL1CDWS	EVSEL1作为向下计数事件源功能使能 参考EVSEL0CDWS描述。
16	EVSEL0CDWS	EVSEL0作为向下计数事件源功能使能 0: 禁能 1: 使能
15:12	保留	必须保持复位值。

11:10	CH1CDWS[1:0]	通道1输入信号边沿作为向下计数事件源功能使能 参考CH0CDWS[1:0]描述。
9:8	CH0CDWS[1:0]	通道0输入信号边沿作为向下计数事件源功能使能 00: 禁能 01: 上升沿 10: 下降沿 11: 上升沿/下降沿
7:6	ETI3CDWS[1:0]	ETI3信号边沿作为向下计数事件源功能使能 参考ETI0CUPS[1:0]描述。
5:4	ETI2CDWS[1:0]	ETI2信号边沿作为向下计数事件源功能使能 参考ETI0CUPS[1:0]描述。
3:2	ETI1CDWS[1:0]	ETI1信号边沿作为向下计数事件源功能使能 参考ETI0CUPS[1:0]描述。
1:0	ETI0CDWS[1:0]	ETI0信号边沿作为向下计数事件源功能使能 00: 禁能 01: 上升沿 10: 下降沿 11: 上升沿/下降沿

22.5.13. 通道 0 控制寄存器(GPTIMERx_CH0CTL)

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
CH0CAPFLT[3:0]				CH0CAPPSC[1:0]		保留		CH0INVSOUT[1:0]		CH0OUT	CH0DISO UT	CH0ENO UT	CH0ENDOUT[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
CH0RED UTY	CH0OUT CTL	CH0OUTM[1:0]		保留		CH0EFF	CH0ACOMUP[1:0]		CH0ACOMDW[1:0]		CH0COMUP[1:0]		CH0COMDW[1:0]		
rw	rw	rw				rw	rw		rw		rw		rw		

位/位域	名称	描述
31:28	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: $f_{SAMP} = f_{DTS}/32$, $N=8$
27:26	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 GPTIMERx_CHCTL 寄存器中的 CH0EN =0 时，则预分频器复位。 00: 无预分频器，捕获输入口上检测到的每一个边沿都触发一次捕获 01: 每 2 个事件触发一次捕获 10: 每 4 个事件触发一次捕获 11: 每 8 个事件触发一次捕获
25:23	保留	必须保持复位值。
22:21	CH0INVSOUT[1:0]	当 GTOC 发送停止输出请求时，通道 0 的无效电平设置 00: 没有影响 01: 输出 Hi-Z 10: 输出低电平 11: 输出高电平
20	CH0OUT	当计数器使能或禁能时，通道 0 输出电平选择 0: 输出电平由 CH0ENOUT 位或 CH0DISOUT 位决定 1: 输出电平保持 注意：如果在停止时修改此位，在 CH0DISOUT 改变或者计数器状态改变时生效。
19	CH0DISOUT	当计数器禁能时，通道 0 输出电平选择 0: 输出低电平 1: 输出高电平 注意：此位只能在计数器停止时配置(CEN=0)，立即生效。
18	CH0ENOUT	当计数器使能时，通道 0 初始输出电平选择 0: 输出低电平 1: 输出高电平
17:16	CH0ENDOUT[1:0]	当周期结束时，通道 0 输出电平选择 00: 输出电平保持 01: 输出低电平

		10: 输出高电平
		11: 输出电平反转
		注意: 周期结束包括上溢(向上计数)和下溢(向下计数)在边沿对齐模式、下溢在中央对齐模式和复位计数器。
15	CH0REDUTY	当计数器从 0%或 100%占空比周期释放时, 通道 0 输出电平选择 0: CH0ENDOUT 选择的电平以 0%或 100%的电平值为基准 1: CH0ENDOUT 选择的电平以 0%或 100%周期内比较匹配定义的电平值为基准
14	CH0OUTCTL	通道 0 输出控制设置 0: 开始计数后CH0OUTM[1:0]设置的模式在第一个上溢(向上计数边沿对齐)或下溢(向下计数边沿对齐和中央对齐)时生效 1: 开始计数后CH0OUTM[1:0]设置的模式立即生效 注意: 此位只能在计数器停止时写 1, 开始计数后第一个周期结束时硬件清零。
13:12	CH0OUTM[1:0]	通道 0 输出模式 0x: 比较匹配结果定义通道输出波形的占空比 10: 通道输出波形的占空比为 0% 11: 通道输出波形的占空比为 100%
11:9	保留	必须保持复位值。
8	CH0EFF	当比较匹配与周期结束同时发生时, 通道 0 周期结束和比较匹配的输出控制是否生效 0: 通道 0 周期结束和比较匹配的输出控制有效, 且比较匹配优先级较高 1: 通道 0 的输出控制无效, 输出电平保持
7:6	CH0ACOMUP[1:0]	当计数器向上计数时, 通道 0 附加比较匹配输出电平控制 参考 CH0COMDW[1:0]描述。
5:4	CH0ACOMDW[1:0]	当计数器向下计数时, 通道 0 附加比较匹配输出电平控制 参考 CH0COMDW[1:0]描述。
3:2	CH0COMUP[1:0]	当计数器向上计数时, 通道 0 比较匹配输出电平控制 参考 CH0COMDW[1:0]描述。
1:0	CH0COMDW[1:0]	当计数器向下计数时, 通道 0 比较匹配输出电平控制 00: 保持 01: 低电平 10: 高电平 11: 翻转电平

22.5.14. 通道 1 控制寄存器(GPTIMERx_CH1CTL)

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
CH1CAPFLT[3:0]				CH1CAPPSC[1:0]		保留		CH1INVSOUT[1:0]		CH1OUT	CH1DISO UT	CH1ENO UT	CH1ENDOUT[1:0]		
rw				rw				rw		rw	rw	rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1RED UTY	CH1OUT CTL	CH1OUTM[1:0]		保留		CH1EFF	CH1ACOMUP[1:0]		CH1ACOMDW[1:0]		CH1COMUP[1:0]		CH1COMDW[1:0]		
rw	rw	rw				rw	rw		rw		rw		rw		

位/位域	名称	描述
31:28	CH1CAPFLT[3:0]	通道 1 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 CH1 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: $f_{SAMP} = f_{DTS}/32$, $N=8$
27:26	CH1CAPPSC[1:0]	通道 1 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 GPTIMERx_CHCTL 寄存器中的 CH1EN = 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
25:23	保留	必须保持复位值。
22:21	CH1INVSOUT[1:0]	当 GTOC 发送停止输出请求时，通道 1 的无效电平设置 00: 没有影响 01: 输出 Hi-Z 10: 输出低电平

		11: 输出高电平
20	CH1OUT	当计数器使能或禁能时，通道 1 输出电平选择 0: 输出电平由 CH1ENOUT 位或 CH1DISOUT 位决定 1: 输出电平保持 注意：计数器停止时修改此位，在 CH1DISOUT 改变或者计数器状态改变时生效。
19	CH1DISOUT	当计数器禁能时，通道 1 输出电平选择 0: 输出低电平 1: 输出高电平 注意：此位只能在计数器停止时配置(CEN=0)，立即生效。
18	CH1ENOUT	当计数器使能时，通道 1 初始输出电平选择 0: 输出低电平 1: 输出高电平
17:16	CH1ENDOUT[1:0]	当周期结束时，通道 1 输出电平选择 00: 输出电平保持 01: 输出低电平 10: 输出高电平 11: 输出电平反转 注意：周期结束包括上溢(向上计数)和下溢(向下计数)在边沿对齐模式、下溢在中央对齐模式和复位计数器。
15	CH1REDUTY	当计数器从 0%或 100%占空比周期释放时，通道 1 输出电平选择 0: CH1ENDOUT 选择的电平以 0%或 100%的电平值为基准 1: CH1ENDOUT 选择的电平以 0%或 100%周期内比较匹配定义的电平值为基准
14	CH1OUTCTL	通道 1 输出控制设置 0: 开始计数后CH1OUTM[1:0]设置的模式在第一个上溢（向上计数边沿对齐）或下溢（向下计数边沿对齐和中央对齐）时生效 1: 开始计数后CH1OUTM[1:0]设置的模式立即生效 注意：此位只能在计数器停止时写 1，开始计数后第一个周期结束时硬件清零。
13:12	CH1OUTM[1:0]	通道 1 输出模式 0x: 比较匹配结果定义通道输出波形的占空比 10: 通道输出波形的占空比为 0% 11: 通道输出波形的占空比为 100%
11:9	保留	必须保持复位值。
8	CH1EFF	当比较匹配与周期结束同时发生时，通道 1 周期结束和比较匹配的输出控制是否生效 0: 通道 1 周期结束和比较匹配的输出控制有效，且比较匹配优先级较高 1: 通道 1 的输出控制无效，输出电平保持
7:6	CH1ACOMUP[1:0]	当计数器向上计数时，通道 1 附加比较匹配输出电平控制 参考 CH1COMDW[1:0]描述。

5:4	CH1ACOMDW[1:0]	当计数器向下计数时，通道 1 附加比较匹配输出电平控制 参考 CH1COMDW[1:0]描述。
3:2	CH1COMUP[1:0]	当计数器向上计数时，通道 1 比较匹配输出电平控制 参考 CH1COMDW[1:0]描述。
1:0	CH1COMDW[1:0]	当计数器向下计数时，通道 1 比较匹配输出电平控制 00: 保持 01: 低电平 10: 高电平 11: 翻转电平

22.5.15. 通道控制寄存器(GPTIMERx_CHCTL)

地址偏移: 0x38

复位值: 0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留											STORES EL	保留		DTEN	CH1MSE L
											rw			rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	CH1SEN	CH1COM ADDSEN	CH1COM SEN	保留		CH1SEL	CH1EN	保留	CH0SEN	CH0COM ADDSEN	CH0COM SEN	保留		CH0SEL	CH0EN
rw		rw	rw			rw	rw	rw		rw	rw			rw	rw

位/位域	名称	描述
31:21	保留	必须保持复位值。
20	STORESEL	停止输出恢复时间选择 0: 在撤销 GTOC 停止输出请求后，第一个计数周期结束恢复 1: 在撤销 GTOC 停止输出请求后，第一个重复计数周期结束恢复
19:18	保留	必须保持复位值。
17	DTEN	死区时间插入使能 0: 禁能 1: 使能
16	CH1MSEL	通道 1 模式选择 0: 独立模式，通道 1 独立输出 1: 互补模式，通道 1 输出与通道 0 互补
15	保留	必须保持复位值。
14	CH1SEN	通道1输出影子寄存器使能 当此位被置 1，GPTIMERx_CH1CTL 寄存器中 CH1ENDOUT[1:0]、

		CH1ACOMUP[1:0]、CH1ACOMDW[1:0]、CH1COMUP[1:0]和CH1COMDW[1:0]位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: 禁能通道 1 输出影子寄存器 1: 使能通道1输出影子寄存器
13	CH1COMADDSEN	通道 1 附加输出比较影子寄存器使能 当此位被置 1，GPTIMERx_CH1COMV_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: 禁能通道 1 附加比较输出影子寄存器 1: 使能通道 1 附加比较输出影子寄存器
12	CH1COMSEN	通道1输出比较影子寄存器使能 当此位被置1，GPTIMERx_CH1CV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: 禁能通道 1 输出比较影子寄存器 1: 使能通道 1 输出比较影子寄存器
11:10	保留	必须保持复位值。
9	CH1SEL	通道 1 捕获/比较模式选择 0: 通道 1 配置为输出 1: 通道 1 配置为输入
8	CH1EN	通道1捕获、比较使能 当通道1配置为输出模式时，将此位置1使能通道1的输出。当通道1配置为输入模式时，将此位置1使能通道1上的捕获事件。 0: 禁能通道1 1: 使能通道 1
7	保留	必须保持复位值。
6	CH0SEN	通道0输出影子寄存器使能 当 此 位 被 置 1 ， GPTIMERx_CH0CTL 寄 存 器 中 CH0ENDOUT[1:0] 、CH0ACOMUP[1:0]、CH0ACOMDW[1:0]、CH0COMUP[1:0]和CH0COMDW[1:0]位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: 禁能通道 0 输出影子寄存器 1: 使能通道 0 输出影子寄存器
5	CH0COMADDSEN	通道 0 附加输出比较影子寄存器使能 当此位被置 0，GPTIMERx_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: 禁能通道 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
4	CH0COMSEN	通道0输出比较影子寄存器使能 当此位被置1，GPTIMERx_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: 禁能通道 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

3:2	保留	必须保持复位值。
1	CH0SEL	通道 0 捕获/比较模式选择 0: 通道 0 配置为输出 1: 通道 0 配置为输入
0	CH0EN	通道0捕获、比较使能 当通道1配置为输出模式时，将此位置1使能通道0的输出。当通道1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: 禁能通道1 1: 使能通道 1

22.5.16. DMA 和中断使能寄存器(GPTIMERx_DMAINTEN)

地址偏移: 0x3C

复位值: 0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留						CLOSSEL[1:0]		保留	RCENDE N	CH1COM ADDEN	CH0COM ADDEN	OVUNDEN[1:0]		CH1DEN	CH0DEN
rw						rw			rw	rw	rw	rw		rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCENDIE	CHLOUT DET	CHHOUT DET	保留					RSIE	保留	CH1COM ADDIE	CH0COM ADDIE	UFIE	OFIE	CH1IE	CH0IE
rw	rw	rw						rw		rw	rw	rw	rw	rw	rw

位/位域	名称	描述
31:26	保留	必须保持复位值。
25:24	CLOSSEL[1:0]	GTOC 停止输出源选择 00: GTOC0 01: GTOC1 10: GTOC2 11: GTOC3
23	保留	必须保持复位值。
22	RCENDEN	重复计数结束DMA请求使能 0: 禁能 1: 使能
21	CH1COMADDEN	通道1附加比较DMA请求使能 0: 禁能 1: 使能

20	CH0COMADDEN	通道0附加比较DMA请求使能 0: 禁能 1: 使能
19:18	OVUNDEN[1:0]	计数器上溢/下溢 DMA 请求使能 00: 禁能 01: 上溢 DMA 请求使能 10: 下溢 DMA 请求使能 11: 上溢/下溢 DMA 请求使能
17	CH1DEN	通道 1 比较/捕获 DMA 请求使能 0: 禁能 1: 使能
16	CH0DEN	通道 0 比较/捕获 DMA 请求使能 0: 禁能 1: 使能
15	RCENDIE	重复计数结束中断使能 0: 禁能 1: 使能
14	CHLOUTDET	通道0和通道1同时输出低电平停止输出检测使能 0: 禁能 1: 使能
13	CHHOUTDET	通道 0 和通道 1 同时输出高电平停止输出检测使能 0: 禁能 1: 使能
12:8	保留	必须保持复位值。
7	RSIE	复位中断使能 0: 禁能 1: 使能
6	保留	必须保持复位值。
5	CH1COMADDIE	通道1附加比较中断使能 0: 禁能 1: 使能
4	CH0COMADDIE	通道0附加比较中断使能 0: 禁能 1: 使能
3	UFIE	计数器下溢中断使能 0: 禁能 1: 使能

2	OFIE	计数器上溢中断使能 0: 禁能 1: 使能
1	CH1IE	通道 1 比较/捕获中断使能 0: 禁能 1: 使能
0	CH0IE	通道 0 比较/捕获中断使能 0: 禁能 1: 使能

22.5.17. 中断标志寄存器(GPTIMERx_INTF)

地址偏移: 0x40

复位值: 0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留							OSTF	保留							DIRF
r							r							r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCENDIF	CHLOUT	CHHOUT	保留	ADT2CM	ADT2CM	ADT1CM	ADT1CM	RSIF	保留	CH1COM	CH0COM	UFIF	OFIF	CH1IF	CH0IF
rc_w0	r	r		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
31:25	保留	必须保持复位值。
24	OSTF	输出停止标志位（根据 CLOSSEL[1:0]位域选择的停止输出源） 0: 未收到 GTOC 发来的停止输出请求 1: 收到 GTOC 发来的停止输出请求 注意: 当收到停止输出请求后，即使停止输出请求位被刷新为 0，无效输出也要持续到当前 PWM 周期结束。
23:17	保留	必须保持复位值。
16	DIRF	计数方向标志位 0: 定时器向上计数 1: 定时器向下计数
15	RCENDIF	重复计数器结束中断标志位 0: 无重复计数器结束中断发生 1: 发生重复计数器结束中断
14	CHLOUTF	通道 0 和通道 1 同时输出低电平检测标志 0: 通道 0 和通道 1 均使能，且不同时输出低电平，或者至少有一个通道不使能

		1: 通道 0 和通道 1 均使能，并同时输出低电平
13	CHHOUTF	通道 0 和通道 1 同时输出高电平检测标志 0: 通道 0 和通道 1 均使能，且不同时输出高电平，或者至少有一个通道不使能 1: 通道 0 和通道 1 均使能，并同时输出高电平 注意：当 CHHOUTDET 位使能时，对应的标志位会输出到 GTOC 模块，不产生中断信号，需要与 GTOC 的中断信号结合使用。
12	保留	必须保持复位值。
11	ADT2CMDIF	ADTCV2[15:0]比较匹配（仅向下计数时）标志 0: 无ADTCV2[15:0]比较匹配中断发生 1: ADTCV2[15:0]比较匹配中断发生
10	ADT2CMUF	ADTCV2[15:0]比较匹配（仅向上计数时）标志 0: 无ADTCV2[15:0]比较匹配中断发生 1: ADTCV2[15:0]比较匹配中断发生
9	ADT1CMDIF	ADTCV1[15:0]比较匹配（仅向下计数时）标志 0: 无ADTCV1[15:0]比较匹配中断发生 1: ADTCV1[15:0]比较匹配中断发生
8	ADT1CMUF	ADTCV1[15:0]比较匹配（仅向上计数时）标志 0: 无ADTCV1[15:0]比较匹配中断发生 1: ADTCV1[15:0]比较匹配中断发生
7	RSIF	复位中断标志 此位在任何计数器清零事件（包括软/硬件计数器清零事件，GPTIMER 组间同步清零）发生时由硬件置 1，软件清 0。 0: 无复位中断发生 1: 发生复位中断
6	保留	必须保持复位值。
5	CH1COMADDIF	通道1附加比较中断 参考 CH0COMADDIF 描述。
4	CH0COMADDIF	通道0附加比较中断 此标志由硬件置 1 软件清 0。当通道 0 在输出模式下时，此标志位在一个比较事件发生时被置 1。 0: 无通道 0 中断发生 1: 通道0中断发生
3	UFIF	计数器下溢中断标志 0: 无下溢中断发生 1: 下溢中断发生
2	OFIF	计数器上溢中断标志 0: 无上溢中断发生

1: 上溢中断发生

1	CH1IF	通道 1 比较/捕获中断标志 参考 CH0IF 描述。
0	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 GPTIMERx_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

22.5.18. 更新源选择寄存器(GPTIMERx_UPSSEL)

地址偏移: 0x44

复位值: 0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				CH1UPSEL[1:0]		CH0UPSEL[1:0]		保留				GLBUPSEL[1:0]			
rw				rw		rw		rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARUPSEL[1:0]		ADCCR2UPSEL[1:0]		ADCCR1UPSEL[1:0]		CH1COMV_ADDUPSEL[1:0]		CH1CVUPSEL[1:0]		CH0COMV_ADDUPSEL[1:0]		CH0CVUPSEL[1:0]			
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。
27:26	CH1UPSEL[1:0]	通道 1 输出模式更新源选择 参考 CH0UPSEL[1:0]描述。
25:24	CH0UPSEL[1:0]	通道 0 输出模式更新源选择 00: 全局更新 01: 上溢更新或计数器清零更新 10: 下溢更新或计数器清零更新 11: 上溢或下溢更新或计数器清零更新 注意: 这些位由 GPTIMERx_CTL0 寄存器中的 UPDIS 位控制。
23:18	保留	必须保持复位值。
17:16	GLBUPSEL[1:0]	全局更新模式更新源选择 00: 在上溢/下溢时递减重复计数器，当重复计数器递减到 0 时产生全局更新事件 01: 在上溢时递减重复计数器，当重复计数器递减到 0 时产生全局更新事件 10: 在下溢时递减重复计数器，当重复计数器递减到 0 时产生全局更新事件 11: 保留 注意: 这些位由 GPTIMERx_CTL0 寄存器中的 UPDIS 位控制。

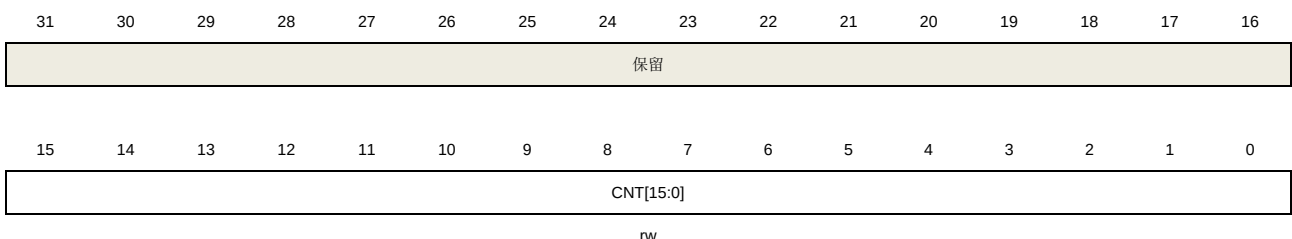
15:14	CARUPSEL[1:0]	计数器自动重载值更新模式选择，用于设置计数器自动重载值何时将被自动覆盖 00：全局更新 01：上溢更新或计数器清零更新 10：下溢更新或计数器清零更新 11：上溢或下溢更新或计数器清零更新 注意： 这些位由 GPTIMERx_CTL0 寄存器中的 UPDIS 位控制。
13:12	ADCCR2UPSEL[1:0]	ADTCV2[15:0]更新模式选择 参考 CARUPSEL[1:0]描述。
11:10	ADCCR1UPSEL[1:0]	ADTCV1[15:0]更新模式选择 参考 CARUPSEL[1:0]描述。
9:7	CH1COMV_ADDUPSEL[1:0]	通道 1 附加比较值更新模式选择 参考 CH0COMV_ADDUPSEL[1:0]描述。
6:5	CH1CVUPSEL[1:0]	通道 1 捕获/比较值更新模式选择 参考 CARUPSEL[1:0]描述。
4:2	CH0COMV_ADDUPSEL[1:0]	通道 0 附加比较值更新模式选择 000：全局更新 001：上溢更新或计数器清零更新 010：下溢更新或计数器清零更新 011：上溢或下溢更新或计数器清零更新 100：通道 0 比较匹配 注意： 这些位由 GPTIMERx_CTL0 寄存器中的 UPDIS 位控制。
1:0	CH0CVUPSEL[1:0]	通道 0 捕获/比较值更新模式选择 参考 CARUPSEL[1:0]描述。

22.5.19. 计数器寄存器(GPTIMERx_CNT)

地址偏移：0x48

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述

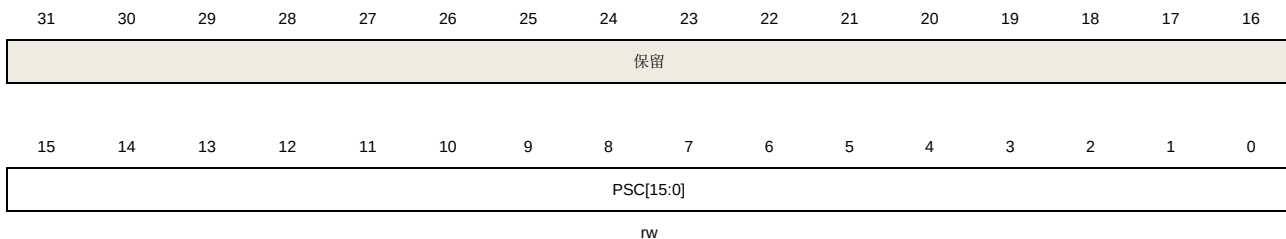
31:16	保留	必须保持复位值。
15:0	CNT[15:0]	这些位是当前的计数值。写操作能改变计数器值。

22.5.20. 预分频寄存器(GPTIMERx_PSC)

地址偏移: 0x4C

复位值: 0x0000 0000

该寄存器只能按字（32 位）访问。



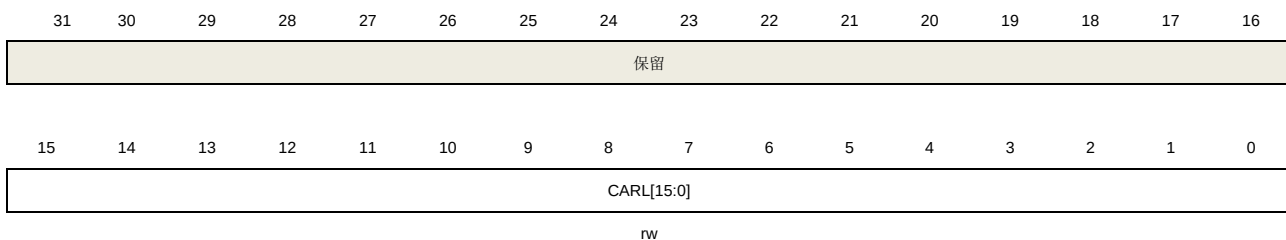
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	PSC[15:0]	计数器时钟预分频值 计数器时钟等于PSC时钟除以（PSC+1），每次当更新事件产生时，PSC的值被装入当前预分频寄存器。

22.5.21. 计数器自动重载寄存器(GPTIMERx_CAR)

地址偏移: 0x50

复位值: 0x0000 0000

This register can be accessed by half-word(16-bit) or word(32-bit).



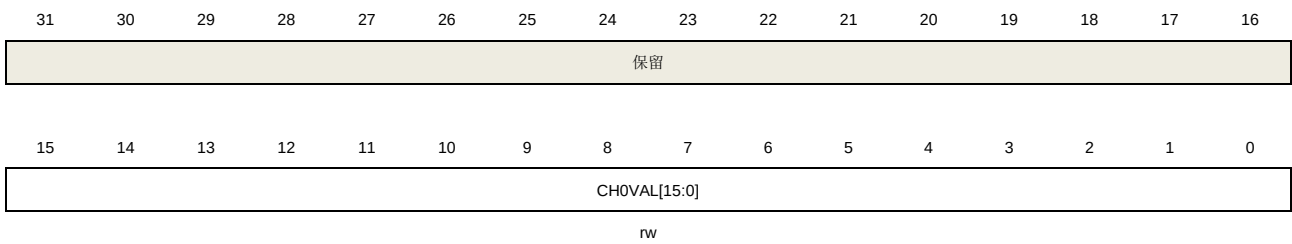
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。

22.5.22. 通道 0 捕获/比较寄存器(GPTIMERx_CH0CV)

地址偏移: 0x54

复位值: 0x0000 0000

该寄存器只能按字（32 位）访问。



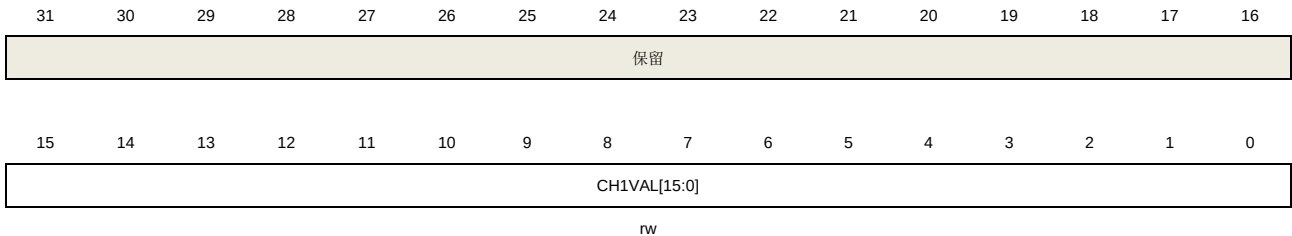
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH0VAL[15:0]	通道 0 的捕获或比较值 当通道 0 配置为输入模式时，这些位决定了上次捕获事件的计数器值，且该位域只读。 当通道0配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。

22.5.23. 通道 1 捕获/比较寄存器(GPTIMERx_CH1CV)

地址偏移：0x58

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，且该位域只读。 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。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。

22.5.24. 通道 0 附加比较寄存器(GPTIMERx_CH0COMV_ADD)

地址偏移：0x5C

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。





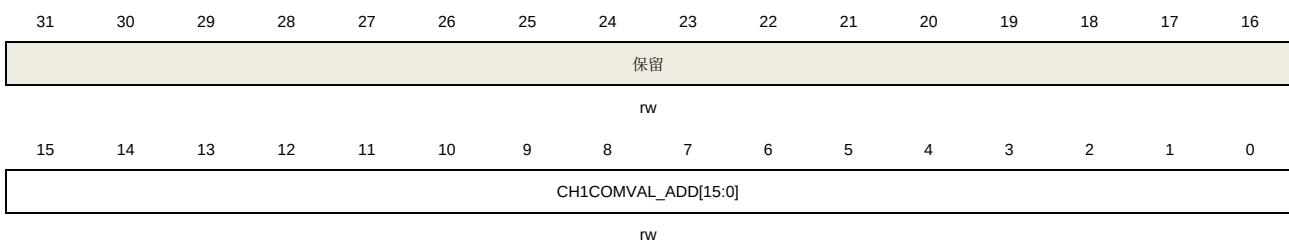
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH0COMVAL_ADD [15:0]	通道 0 附加比较值 当通道 0 配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。

22.5.25. 通道 1 附加比较寄存器(GPTIMERx_CH1COMV_ADD)

地址偏移：0x60

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



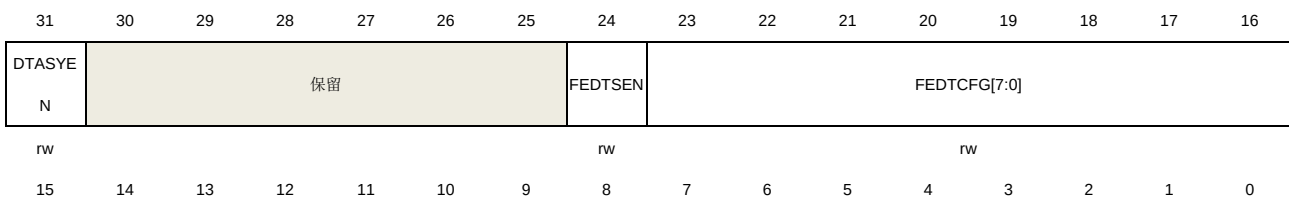
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CH1COMVAL_ADD [15:0]	通道 1 附加比较值 当通道1配置为输出模式时，这些位包含了即将和计数器比较的值。使能相应的影子寄存器后，影子寄存器值随每次更新事件更新。

22.5.26. 死区控制寄存器(GPTIMERx_DTCTL)

地址偏移：0x64

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



保留	REDTSEN	REDTCFG[7:0]
	rw	rw

位/位域	名称	描述
31	DTASYEN	死区时间不对称模式使能 0: 上升沿和下降沿死区时间均使用 REDTCFG[7:0] 1: 上升沿死区使用 REDTCFG[7:0], 下降沿死区使用 FEDTCFG[7:0] 注意: 此位不能在 CEN=1 时更改。
30:25	保留	必须保持复位值。
24	FEDTSEN	下降沿死区时间影子寄存器使能 当该位置位时, 可支持动态改变死区时间值, 死区时间在全局计数器更新事件到来时生效。 0: 禁能 1: 使能
23:16	FEDTCFG[7:0]	下降沿死区时间控制 这些位定义了插入互补输出之间的死区持续时间。FEDTCFG 值和死区时间的关系如下: $FEDTCFG[7:5] = 3'b0xx: DTvalue = FEDTCFG[7:0] \times t_{DT}, t_{DT} = t_{DTS}$ $FEDTCFG[7:5] = 3'b10x: DTvalue = (64 + FEDTCFG[5:0]) \times t_{DT}, t_{DT} = t_{DTS} \times 2$ $FEDTCFG[7:5] = 3'b110: DTvalue = (32 + FEDTCFG[4:0]) \times t_{DT}, t_{DT} = t_{DTS} \times 8$ $FEDTCFG[7:5] = 3'b111: DTvalue = (32 + FEDTCFG[4:0]) \times t_{DT}, t_{DT} = t_{DTS} \times 16$
15:9	保留	必须保持复位值。
8	REDTSEN	上升沿死区时间影子寄存器使能 当该位置位时, 可支持动态改变死区时间值, 死区时间在全局计数器更新事件到来时生效。 0: 禁能 1: 使能
7:0	REDTCFG[7:0]	上升沿死区时间控制 这些位定义了插入互补输出之间的死区持续时间。REDTCFG 值和死区时间的关系如下: $REDTCFG[7:5] = 3'b0xx: DTvalue = REDTCFG[7:0] \times t_{DT}, t_{DT} = t_{DTS}$ $REDTCFG[7:5] = 3'b10x: DTvalue = (64 + REDTCFG[5:0]) \times t_{DT}, t_{DT} = t_{DTS} \times 2$ $REDTCFG[7:5] = 3'b110: DTvalue = (32 + REDTCFG[4:0]) \times t_{DT}, t_{DT} = t_{DTS} \times 8$ $REDTCFG[7:5] = 3'b111: DTvalue = (32 + REDTCFG[4:0]) \times t_{DT}, t_{DT} = t_{DTS} \times 16$

22.5.27. ADC 触发控制寄存器(GPTIMERx_ADCTL)

地址偏移: 0x68

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				ADCMS4	保留				ADCMS4	保留				ADCMS3	ADCMS3
				EN					SEL					EN	SEL
rw					rw					rw					rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADT2DW	ADT2UP	ADT1DW	ADT1UP	保留		ADTCV2	ADTCV1	保留	ADTCV2SSEL[2:0]			保留	ADTCV1SSEL[2:0]		
GTEN	GTEN	GTEN	GTEN			SEN	SEN		rw				rw		
rw	rw	rw	rw			rw	rw		rw				rw		

位/位域	名称	描述
31:29	保留	必须保持复位值。
28	ADCMS4EN	ADC触发监测信号4使能 0: 禁能 1: 使能
27:25	保留	必须保持复位值。
24	ADCMS4SEL	ADC触发监测信号4选择 0: 由ADTCV1[15:0]比较匹配产生的触发信号 1: 由ADTCV2[15:0]比较匹配产生的触发信号
23:21	保留	必须保持复位值。
20	ADCMS3EN	ADC触发监测信号3使能 0: 禁能 1: 使能
19:17	保留	必须保持复位值。
16	ADCMS3SEL	ADC触发监测信号3选择 0: 由ADTCV1[15:0]比较匹配产生的触发信号 1: 由ADTCV2[15:0]比较匹配产生的触发信号
15	ADT2DWGTEN	ADTCV2[15:0]比较匹配（仅向下计数）生成触发使能 0: 禁能 1: 使能
14	ADT2UPGTEN	ADTCV2[15:0]比较匹配（仅向上计数）生成触发使能 0: 禁能 1: 使能
13	ADT1DWGTEN	ADTCV1[15:0]比较匹配（仅向下计数）生成触发使能 0: 禁能 1: 使能
12	ADT1UPGTEN	ADTCV1[15:0]比较匹配（仅向上计数）生成触发使能 0: 禁能

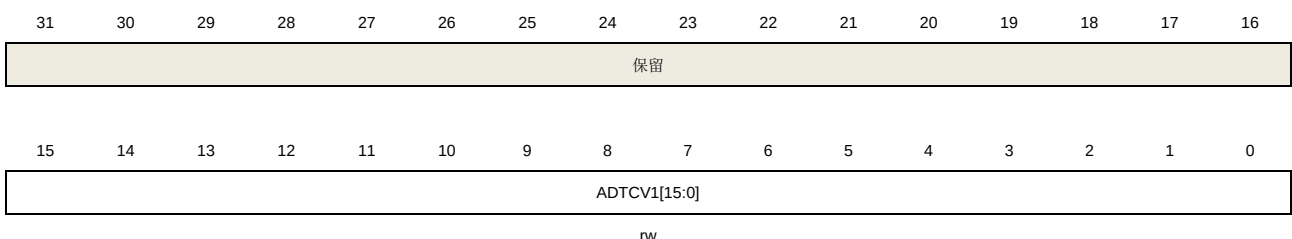
		1: 使能
11:10	保留	必须保持复位值。
9	ADTCV2SEN	ADTCV2[15:0]影子寄存器使能 0: 禁能 1: 使能
8	ADTCV1SEN	ADTCV1[15:0]影子寄存器使能 0: 禁能 1: 使能
7	保留	必须保持复位值。
6:4	ADTCV2SSEL[2:0]	由ADTCV2[15:0]生成的触发信号产生ADC转换请求跳过源选择 参考ADTCV1SSEL[2:0]描述。
3	保留	必须保持复位值。
2:0	ADTCV1SSEL[2:0]	由ADTCV1[15:0]生成的触发信号产生ADC转换请求跳过源选择 000: 不进行比较匹配跳过 001: 当ADC跳过计数器1的值不是4'b0000时跳过ADC转换请求（在ADCC1VAL[3:0] = 4'b0000期间输出ADC转换请求） 010: 当ADC跳过计数器2的值不是4'b0000时跳过ADC转换请求（在ADCC2VAL[3:0] = 4'b0000期间输出ADC转换请求） 011: 当ADC跳过计数器1/2的值不是4'b0000时跳过ADC转换请求（在ADCC1VAL[3:0] = 4'b0000和ADCC2VAL[3:0] = 4'b0000期间输出ADC转换请求） 100: 保留 101: 当ADC跳过计数器1的值不是跳过值时跳过ADC转换请求（在ADCC1VAL[3:0] = ADCC1SVAL[3:0]期间输出ADC转换请求） 110: 当ADC跳过计数器2的值不是跳过值时跳过ADC转换请求（在ADCC2VAL[3:0] = ADCC2SVAL[3:0]期间输出ADC转换请求） 111: 当ADC跳过计数器1/2的值不是跳过值时跳过ADC转换请求（在ADCC1VAL[3:0] = ADCC1SVAL[3:0]和ADCC2VAL[3:0] = ADCC2SVAL[3:0]期间输出ADC转换请求）

22.5.28. ADC 触发比较值 1 寄存器(GPTIMERx_ADCCR1)

地址偏移: 0x6C

复位值: 0x0000 0000

该寄存器只能按字（32 位）访问。



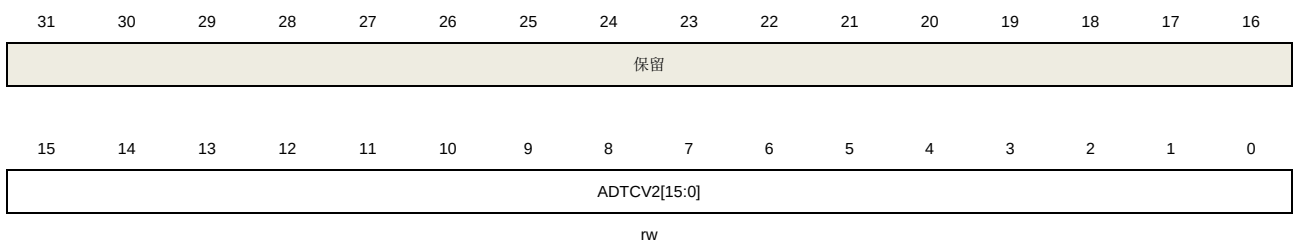
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	ADTCV1[15:0]	ADC触发比较值1 这些位包含与计数器比较的值。在相应的影子寄存器被启用后，寄存器值会根据影子寄存器的配置进行更新。

22.5.29. ADC 触发比较值 2 寄存器(GPTIMERx_ADCCR2)

地址偏移：0x70

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



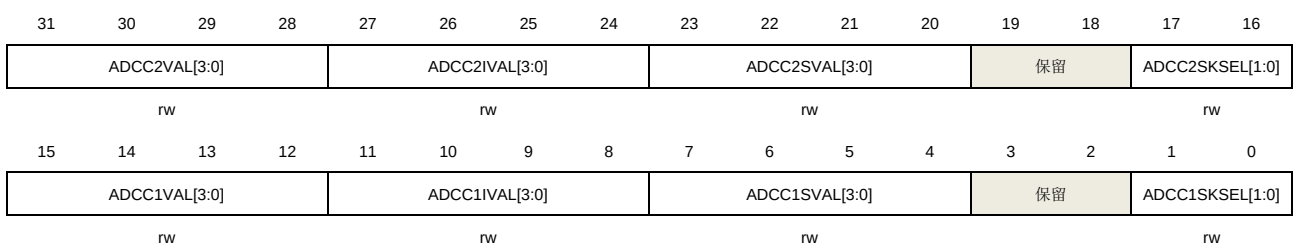
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	ADTCV2[15:0]	ADC触发比较值2 这些位包含与计数器比较的值。在相应的影子寄存器被启用后，寄存器值会根据影子寄存器的配置进行更新。

22.5.30. ADC 触发跳过寄存器(GPTIMERx_ADCTRGs)

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
31:28	ADCC2VAL[3:0]	当前ADC跳过计数器2的值。 范围：0 - ADCC2SVAL[3:0] 注意： 当写入ADCC2SKSEL[1:0] = 2'b00或ADCC2SVAL[3:0] = 4'b0000时，该位域

被重置。在ADCC2SVAL[3:0]位中设置非零后的第一个计数时钟，该计数器+1。

27:24	ADCC2IVAL[3:0]	ADC跳过计数器2的初始值 范围：0-15 注意：只有当ADCC2SKSEL[1:0] != 2'b00时，该位域才可写，并且该位域不会被重置。
23:20	ADCC2SVAL[3:0]	ADC跳过计数器2的跳过值 范围：0-15
19:18	保留	必须保持复位值。
17:16	ADCC2SKSEL[1:0]	ADC跳过计数器2跳过源选择 00：不计数、不跳过 01：ADTCV1[15:0]比较匹配 10：ADTCV2[15:0]比较匹配 11：ADTCV1[15:0]或ADTCV2[15:0]比较匹配
15:12	ADCC1IVAL[3:0]	当前ADC跳过计数器1的值。 范围：0 – ADCC1SVAL[3:0] 注意： 当写入ADCC1SKSEL[1:0] = 2'b00或ADCC1SVAL[3:0] = 4'b0000时，该位域被重置。在ADCC1SVAL[3:0]位中设置非零后的第一个计数时钟，该计数器+1。
11:8	ADCC1IVAL[3:0]	ADC跳过计数器1的初始值 范围：0-15 注意：只有当ADCC1SKSEL[1:0] != 2'b00时，该位域才可写，并且该位域不会被重置。
7:4	ADCC1SVAL[3:0]	ADC跳过计数器2的跳过值 范围：0-15
3:2	保留	必须保持复位值。
1:0	ADCC1SKSEL[1:0]	ADC跳过计数器1跳过源选择 00：不计数、不跳过 01：ADTCV1[15:0]比较匹配 10：ADTCV2[15:0]比较匹配 11：ADTCV1[15:0]或ADTCV2[15:0]比较匹配

22.5.31. 附加中断跳过控制寄存器 0(GPTIMERx_ADDINTSCTL0)

地址偏移：0x78

复位值：0x0FF0 00FF

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADCC2VAL[3:0]				ADCC2IVAL[3:0]				ADCC2SVAL[3:0]				保留		ADCC2SKSEL[1:0]	
rw				rw				rw						rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDC1VAL[3:0]				ADDC1IVAL[3:0]				ADDC1SVAL[3:0]				保留		ADDC1SKSEL[1:0]	
rw				rw				rw				rw			

位/位域	名称	描述
31:28	ADDC2VAL[3:0]	当前附加中断跳过计数器2的值。 范围：0 - ADDC2SVAL[3:0] 注意： 当写入ADDC2SKSEL[1:0] = 2'b00或ADDC2SVAL[3:0] = 4'b0000时，该位域被重置。在ADDC2SVAL[3:0]位中设置非零后的第一个计数时钟，该计数器+1。
27:24	ADDC2IVAL[3:0]	附加中断跳过计数器2的初始值 范围：0-15 注意：只有当ADDC2SKSEL[1:0] != 2'b00时，该位域才可写，并且该位域不会被重置。
23:20	ADDC2SVAL[3:0]	附加中断跳过计数器2的跳过值 范围：0-15
19:18	保留	必须保持复位值。
17:16	ADDC2SKSEL[1:0]	附加中断跳过计数器2跳过源选择 00：不计数、不跳过 01：边沿对齐时上溢(向上计数)，下溢(向下计数)，中央对齐时上溢 10：边沿对齐时上溢(向上计数)，下溢(向下计数)，中央对齐时下溢 11：边沿对齐时上溢(向上计数)，下溢(向下计数)，中央对齐时上溢或下溢
15:12	ADDC1VAL[3:0]	当前附加中断跳过计数器1的值。 范围：0 – ADDC1SVAL[3:0] 注意： 当写入ADDC1SKSEL[1:0] = 2'b00或ADDC1SVAL[3:0] = 4'b0000时，该位域被重置。在ADDC1SVAL[3:0]位中设置非零后的第一个计数时钟，该计数器+1。
11:8	ADDC1IVAL[3:0]	附加中断跳过计数器2的初始值 范围：0-15 注意：只有当ADDC1SKSEL[1:0] != 2'b00时，该位域才可写，并且该位域不会被重置。
7:4	ADDC1SVAL[3:0]	附加中断跳过计数器2的跳过值 范围：0-15
3:2	保留	必须保持复位值。
1:0	ADDC1SKSEL[1:0]	附加中断跳过计数器2跳过源选择 00：不计数、不跳过 01：边沿对齐时上溢(向上计数)，下溢(向下计数)，中央对齐时上溢 10：边沿对齐时上溢(向上计数)，下溢(向下计数)，中央对齐时下溢 11：边沿对齐时上溢(向上计数)，下溢(向下计数)，中央对齐时上溢或下溢

22.5.32. 附加中断跳过控制寄存器 1(GPTIMERx_ADDINTSCTL1)

地址偏移：0x7C

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	ADT2AISSEL[2:0]			保留	ADT1AISSEL[2:0]			保留	OFAISSEL[2:0]			保留	UFAISSEL[2:0]		
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	CH1ADDAISSEL[2:0]			保留	CH0ADDAISSEL[2:0]			保留	CH1AISSEL[2:0]			保留	CH0AISSEL[2:0]		
rw				rw				rw				rw			

位/位域	名称	描述
31	保留	必须保持复位值。
30:28	ADT2AISSEL[2:0]	ADTCV2[15:0]产生的触发信号附加中断跳过功能选择 参考CH0AISSEL[2:0]描述。
27	保留	必须保持复位值。
26:24	ADT1AISSEL[2:0]	ADTCV1[15:0]产生的触发信号附加中断跳过功能选择 参考CH0AISSEL[2:0]描述。
23	保留	必须保持复位值。
22:20	OFAISSEL[2:0]	上溢附加中断跳过功能选择 参考CH0AISSEL[2:0]描述。
19	保留	必须保持复位值。
18:16	UFAISSEL[2:0]	下溢附加中断跳过功能选择 参考CH0AISSEL[2:0]描述。
15	保留	必须保持复位值。
14:12	CH1ADDAISSEL[2:0]	通道1附加比较值比较匹配附加中断跳过功能选择 参考CH0AISSEL[2:0]描述。
11	保留	必须保持复位值。
10:8	CH0ADDAISSEL[2:0]	通道0附加比较值比较匹配附加中断跳过功能选择 参考CH0AISSEL[2:0]描述。
7	保留	必须保持复位值。
6:4	CH1AISSEL[2:0]	通道1输入捕获/比较匹配附加中断跳过功能选择 参考CH0AISSEL[2:0]描述。
3	保留	必须保持复位值。

2:0	CH0AISSEL[2:0]	通道0输入捕获/比较匹配附加中断跳过功能选择
		000: 不跳过
		001: 当附加中断跳过计数器1的值不是4'b0000时跳过该中断请求(当ADDC1VAL[3:0] = 4'b0000时输出中断请求)
		010: 当附加中断跳过计数器2的值不是4'b0000时跳过该中断请求(当ADDC2VAL[3:0] = 4'b0000时输出中断请求)
		011: 当附加中断跳过计数器1/2的值不是4'b0000时跳过该中断请求(当ADDC1VAL[3:0] = 4'b0000和ADDC2VAL[3:0] = 4'b0000时输出中断请求)
		100: 保留
		101: 当附加中断跳转计数器1的值不是跳过值时跳过该中断请求(当ADDC1VAL[3:0] = ADDC1SVAL[3:0]时输出中断请求)
		110: 当附加中断跳转计数器2的值不是跳过值时跳过该中断请求(当ADDC2VAL[3:0] = ADDC2SVAL[3:0]时输出中断请求)
		111: 当附加中断跳转计数器1/2的值不是跳过值时跳过该中断请求(当ADDC1VAL[3:0] = ADDC1SVAL[3:0]和ADDC2VAL[3:0] = ADDC2SVAL[3:0]时输出中断请求)

22.5.33. 中断和 ADC 触发信号跳过寄存器(GPTIMERx_IADCTSS)

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													OUFINTSC[2:0]		
													rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		ADT2INT	ADT1INT	保留		OUFINTSVAL[2:0]		保留		OUFINTSSEL[1:0]		CH1AINT	CH0AINT	CH1INTS	CH0INTS
		SL	SL									SL	SL	L	L
rw		rw	rw	rw		rw		rw		rw		rw	rw	rw	rw

位/位域	名称	描述
31:19	保留	必须保持复位值。
18:16	OUFINTSC[2:0]	上溢/下溢中断跳过计数器 范围: 0-7 注意: 当写入 OUFINTSSEL[1:0] = 2'b00 或 OUFINTSVAL[2:0] = 3'b000 时, 该位域被重置。当计数器停止时, 该位域清零, 附加中断跳过的两个计数器保持当前值。
15:14	保留	必须保持复位值。
13	ADT2INTSL	ADTCV2[15:0]产生的触发信号中断跳过链接 参考 CH0INTSL 描述。
12	ADT1INTSL	ADTCV1[15:0]产生的触发信号中断跳过链接

参考 CH0INTSL 描述。

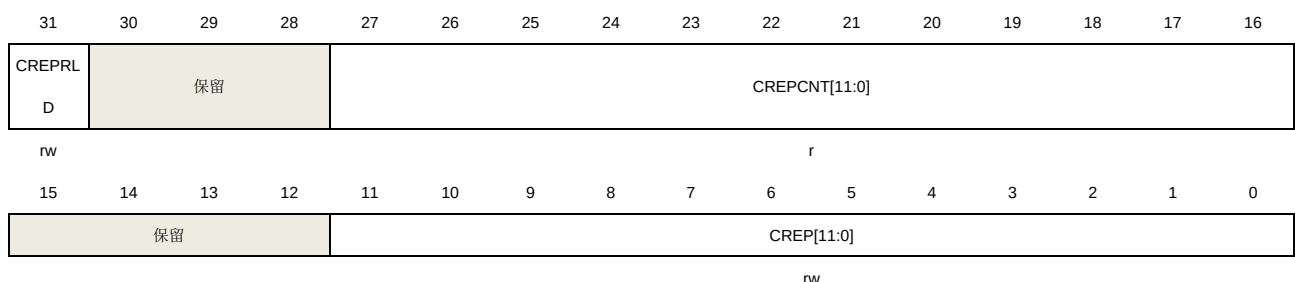
11	保留	必须保持复位值。
10:8	OUFINTSVAL[2:0]	上/下溢中断跳过数量选择 范围：0-7
7:6	保留	必须保持复位值。
5:4	OUFINTSSEL[1:0]	上/下溢中断跳过源选择 00：不计数、不跳过 01：边沿对齐时上溢(向上计数)，下溢(向下计数)，中央对齐时上溢 10：边沿对齐时上溢(向上计数)，下溢(向下计数)，中央对齐时下溢 11：边沿对齐时上溢(向上计数)，下溢(向下计数)，中央对齐时上溢或下溢
3	CH1AINTSL	通道 1 附加比较值比较匹配中断跳过链接 参考 CH0INTSL 描述。
2	CH0AINTSL	通道 0 附加比较值比较匹配中断跳过链接 参考 CH0INTSL 描述。
1	CH1INTSL	通道 1 输入捕获/比较匹配中断跳过链接 参考 CH0INTSL 描述。
0	CH0INTSL	通道 0 输入捕获/比较匹配中断跳过链接 0：不与上/下溢中断跳过功能链接 1：与上/下溢中断跳过功能链接

22.5.34. 重复计数器寄存器(GPTIMERx_CREP)

地址偏移：0x84

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	CREPRLD	重复计数器立即加载 该位由软件设置为 1，当软件置位时，CREPCNT[11:0]和 CREP[11:0]加载新的值，加载完成后，硬件清除该位。
30:28	保留	必须保持复位值。

27:16	CREPCNT[11:0]	重复计数器当前值 重复计数器的数值，可读回重复计数递减到多少。
15:12	保留	必须保持复位值。
11:0	CREP[11:0]	重复计数器值 这些位定义了更新事件的产生速率。重复计数器计数值减为 0 时产生更新事件。影子寄存器的更新速率也会受这些位影响（前提是影子寄存器被使能）。

22.5.35. 同步复位控制寄存器(GPTIMERx_SYN_RSTCTL)

地址偏移：0x88

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										SYNCSS EL	保留				CHSRET EN		
										rw						rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
保留								UNDRET EN	OVRETE N	保留		CH1ARE TEN	CH0ARE TEN	CH1RET EN	CH0RET EN		
								rw		rw				rw		rw	

位/位域	名称	描述
31:22	保留	必须保持复位值。
21	SYNCSSEL	GPTIMER 同步控制集合选择 当选择同一集合时，两个 GPTIMER 同步控制 00: SET0 01: SET1
20:17	保留	必须保持复位值。
16	CHSRETEEN	通道输入作为其它 GPTIMER 同步复位源使能 0: 禁止使用 GPTIMERx_RSSEL 寄存器中选定的通道作为其它 GPTIMER 的同步复位源 1: 使用 GPTIMERx_RSSEL 寄存器中选择的通道作为其他 GPTIMER 的同步复位源
15:8	保留	必须保持复位值。
7	UNDRETEEN	下溢作为其它 GPTIMER 同步复位源使能 0: 禁能 1: 使能
6	OVRETEEN	上溢作为其它 GPTIMER 同步复位源使能 0: 禁能 1: 使能

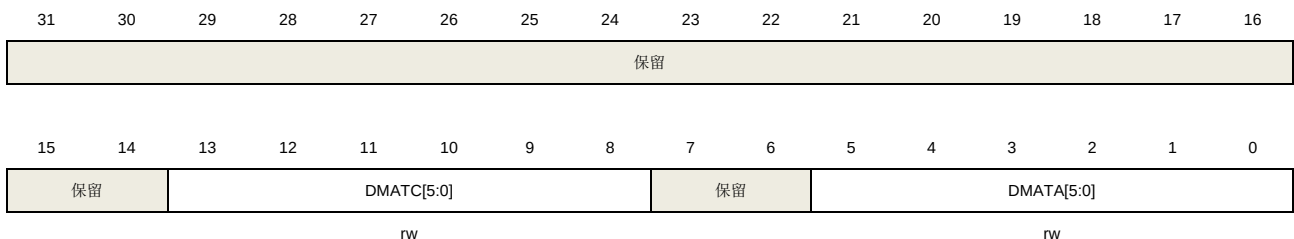
5:4	保留	必须保持复位值。
3	CH1ARETEN	通道 1 附加值比较匹配作为其它 GPTIMER 同步复位源使能 0: 禁能 1: 使能
2	CH0ARETEN	通道 0 附加值比较匹配作为其它 GPTIMER 同步复位源使能 0: 禁能 1: 使能
1	CH1RETEN	通道 1 比较匹配/输入捕获作为其它 GPTIMER 同步复位源使能 0: 禁能 1: 使能
0	CH0RETEN	通道 0 比较匹配/输入捕获作为其它 GPTIMER 同步复位源使能 0: 禁能 1: 使能

22.5.36. DMA 配置寄存器(GPTIMERx_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）GPTIMERx_DMATB寄存器的次数。 6'b000000: 传输1次 6'b000001: 传输2次 ... 6'b100011: 传输35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GPTIMERx_DMATB寄存器的第一个地址。当通过GPTIMERx_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GPTIMERx_DMATB时，将访问起始地址+0x4。

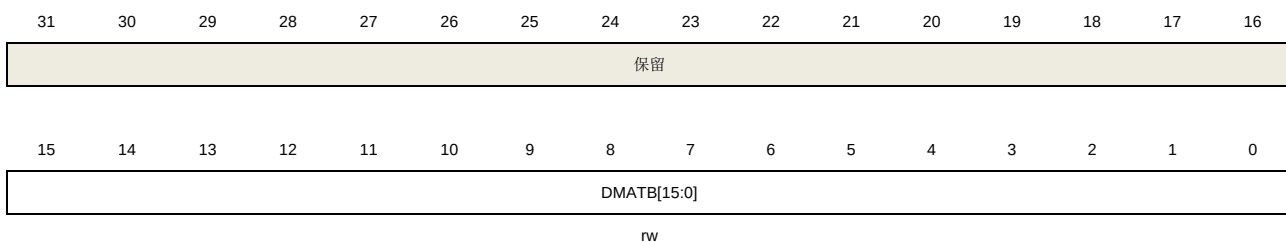
6'b000000: GPTIMERx_WP
 6'b000001: GPTIMERx_SWEN
 ...
 6'b100011: GPTIMERx_SYNRSCTL
 总之：起始地址 = GPTIMERx_WP + DMATA*4

22.5.37. DMA 发送缓冲区寄存器(GPTIMERx_DMATB)

地址偏移：0xE4

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



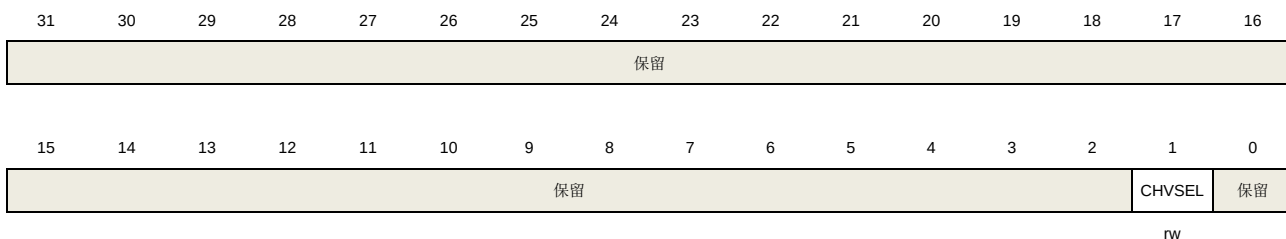
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	DMATB[15:0]	DMA 发送缓冲 对这个寄存器的读或写，（起始地址+传输次数*4）地址范围内的寄存器会被访问 传输次数由硬件计算，范围为0到DMATC。

22.5.38. 配置寄存器(GPTIMERx_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。

23. 比较匹配定时器 W (CPTIMERW)

23.1. 简介

比较匹配定时器 W 模块包含一个 32 位无符号计数器，可以配置为 16 位。比较匹配定时器 W 有两个输入捕获通道和四个比较匹配通道（其中两通道支持脉冲输出）。所有通道和计数器更新事件都可以配置以设定周期产生中断或 EVIC 事件。

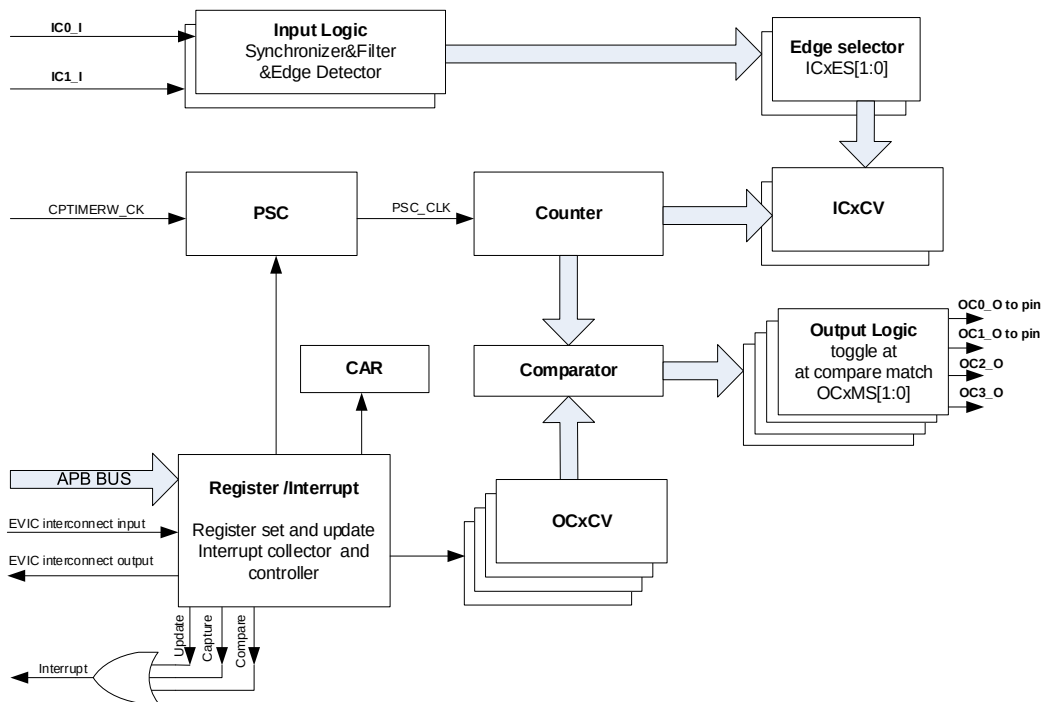
23.2. 主要特征

- 计数器宽度：32 位或 16 位可配置；
- 时钟源：仅限内部时钟；
- 计数模式：仅向上计数；
- 自动重装载功能；
- 可编程分频器：16 位。可以在运行中更改分频系数；
- 计数器清除功能：输入捕获事件发生时清除计数器；
- 输入捕获：边沿可配置的 2 个输入捕获通道；
- 输出比较：4 个输出比较通道，其中 2 个通道支持脉冲输出到引脚；
- EVIC 互连输入：计数器使能、计数器重启和事件计数；
- EVIC 互连输出：比较匹配事件输出到 EVIC，无需使能中断；
- 在输入捕获、比较匹配和更新事件时产生中断；

23.3. 功能说明

[图 23-1. 比较匹配定时器 W 框图](#)提供了比较匹配定时器 W 内部配置的详细信息。

图 23-1. 比较匹配定时器 W 框图



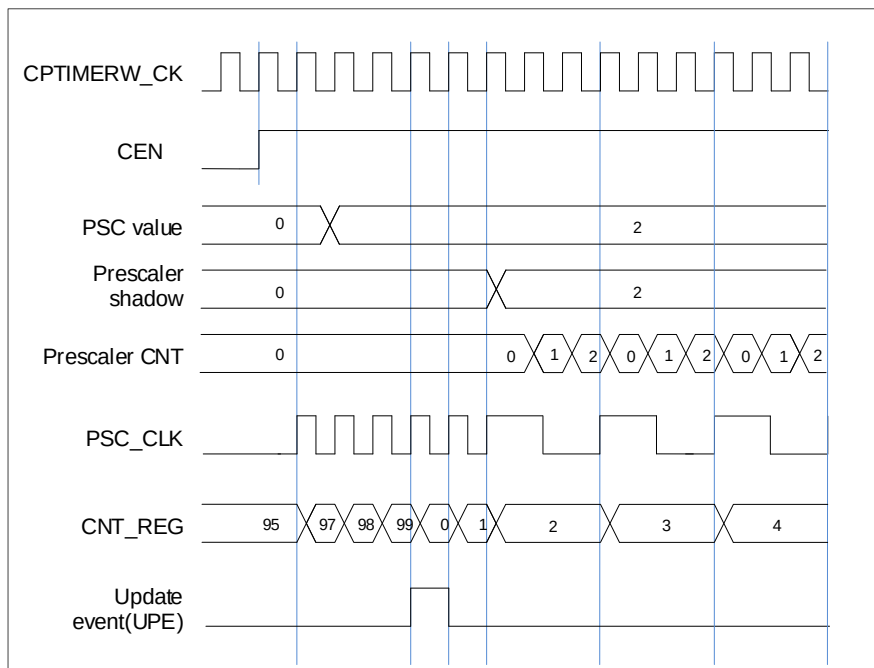
23.3.1. 时钟源

CPTIMERW 只能由内部定时器时钟驱动，该时钟来自 RCU 中的名为 CK_CPTIMERW 的源。驱动计数器预分频器的 CPTIMERW_CK 等于 CK_CPTIMERW。当 CEN 被设置时，CPTIMERW_CK 经过预分频器（预分频值由 PSC 寄存器确定）产生 PSC_CLK。

23.3.2. 时钟预分频器

分频器可以将定时器时钟（CPTIMERW_CK）频率按 1 到 65536 之间的任意值分频，分频后的时钟 PSC_CLK 驱动计数器计数。分频系数受预分频寄存器（CPTIMERW_PSC）控制，这个控制寄存器带有影子寄存器，它能够在运行时被改变。新的预分频器的值在下一个更新事件到来时被采用。

图 23-2. PSC 值从 0 变化到 2 的时序图，CAR=99

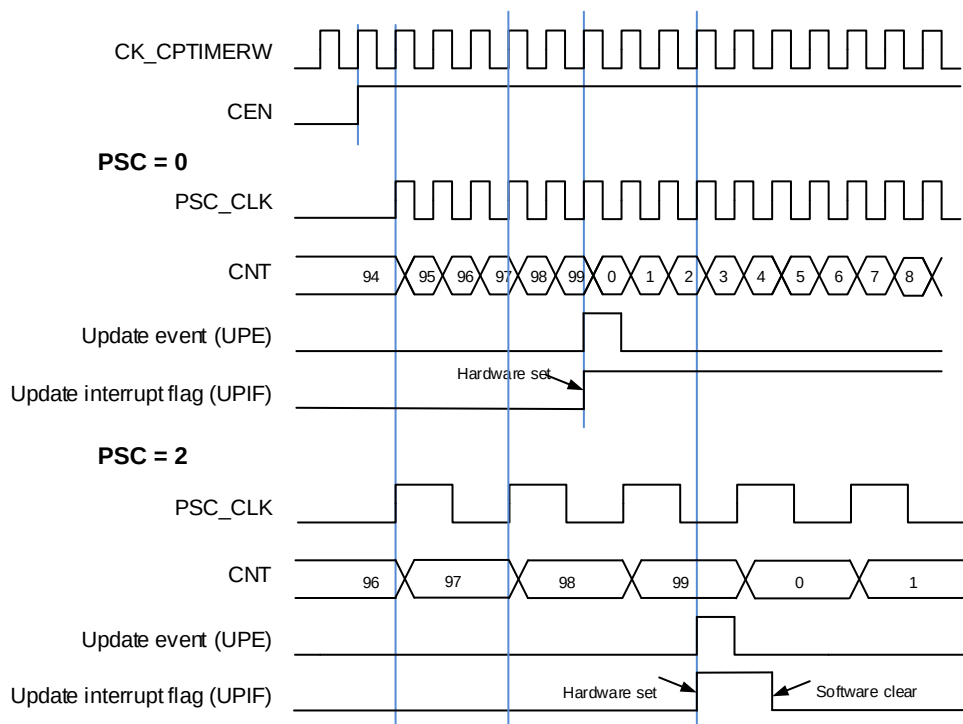


23.3.3. 计数器向上计数

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 **CPTIMERW_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，每次计数器溢出时都会生成更新事件。

下面这些图给出了一些例子，当 **CPTIMERW_CAR=0x99** 时，计数器在不同预分频因子下的行为。

图 23-3. 向上计数时序图, PSC=0/2



23.3.4. 计数器位宽配置

CPTIMERW 的计数器位宽可以通过在 CPTIMERW_CTL0 寄存器中的 CNTWSEL 位设置为 32 位或 16 位。当 CNTWSEL 设置为 1 时, 计数器宽度为 16 位, CPTIMERW_CAR 寄存器的最高 16 位应设置为 0000h, 而计数器溢出值为 0000_FFFFh。复位后, 默认的计数器宽度为 32 位。

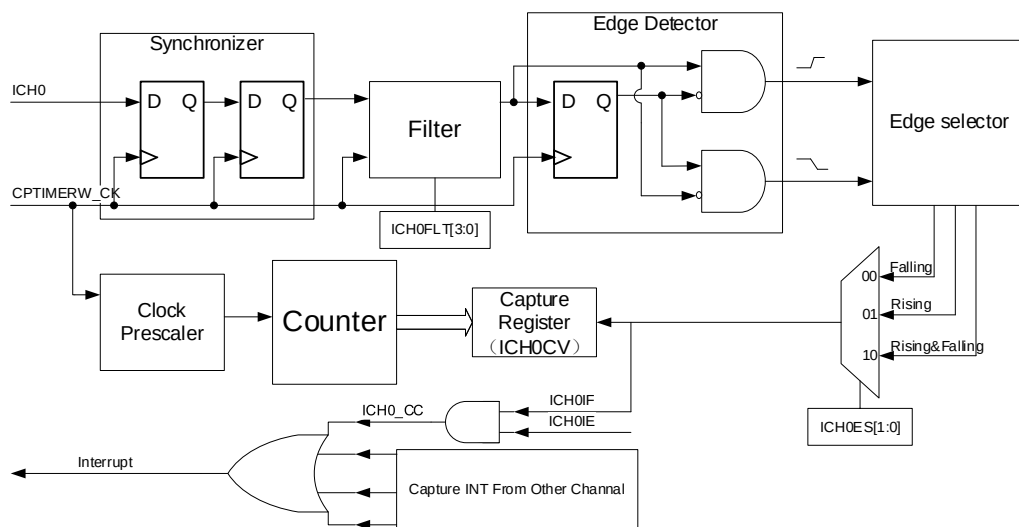
23.3.5. 输入捕获和输出比较

CPTIMERW 具有两个输入捕获通道和四个输出比较通道。每个通道都由相应的寄存器独立控制。每个输入捕获通道包括一个输入引脚, 而输出比较只有四个通道中的两个 (通道 0/1) 可以输出脉冲到引脚。

■ 通道输入捕获功能

通道输入捕获功能允许通道执行诸如脉冲计时、频率、周期、占空比等测量。输入阶段由数字滤波器、通道边沿检测和通道预分频器组成。当通道输入上发生选定的边沿时, 计数器的当前值被捕获到 CPTIMERW_ICHxCV 寄存器中, 同时 ICHxIF 位被置位, 并且如果由 ICHxIE = 1 启用, 则产生通道中断。

图 23-4. 通道输入捕获原理



其中一个通道的输入信号 (ICHx) 可以选择来自 CPTIMERW_ICHx 信号。首先，通道输入信号 (ICHx) 被同步到 CK_CPTIMERW 域，然后通过数字滤波器采样以生成滤波后的输入信号。然后通过边沿检测器，检测上升沿和下降沿。用户可以通过 ICHxES[1:0] 位选择其中一个边沿或双边沿。预分频器使多个输入边沿事件生成一个有效的捕获事件。在捕获事件发生时，ICHxCV 将存储当前计数器的值。

因此，这个过程可以被划分为以下几个步骤：

第一步：滤波配置 (CPTIMERW_CTL1 中的 ICHxFLT)。

根据输入信号和需求的信号质量，配置合适的 ICHxFLT。

第二步：边沿选择 (CPTIMERW_CTL1 中 ICHxES)。

选择上升沿、下降沿或两者，通过 ICHxES 选择一个。

第三步：中断使能 (CPTIMERW_INTEN 中的 ICHxIE)。

根据实际应用需求启用或禁用相关的中断使能。

第四步：捕获使能 (CPTIMERW_CTL1 中的 ICHxEN)。

结果：当检测到选定的输入信号边沿时，当前计数器的值将被加载到 CPTIMERW_ICHxCV 中，并且 ICHxIF 将被置位。根据 CPTIMERW_CTL0 中 ICHxIE 的配置，将使能、禁止中断请求产生。

■ 通道输出比较功能

在通道输出比较功能中，CPTIMERW 可以生成具有可编程的位置、极性、持续时间和频率的定时脉冲。当计数器与输出比较通道的 OCHxCV 寄存器中的值匹配时，通道 x (x=0,1) 的输出基于 OCHxMS 可以翻转或保持不变。当计数器达到 OCHxCV 寄存器中的值时，将置位 OCHxIF 位，并且如果 OCHxIE = 1，则产生通道 x (x=0...3) 中断。

因此，这个过程可以被划分为以下几个步骤：

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。

通过CPTIMERW_CTL1中的OCHxMS[1:0]位设置输出模式。

CPTIMERW_CTL1中的OCHxEN启用输出。

第三步：通CPTIMERW_INTEN的OCHxIE位配置中断请求使能。

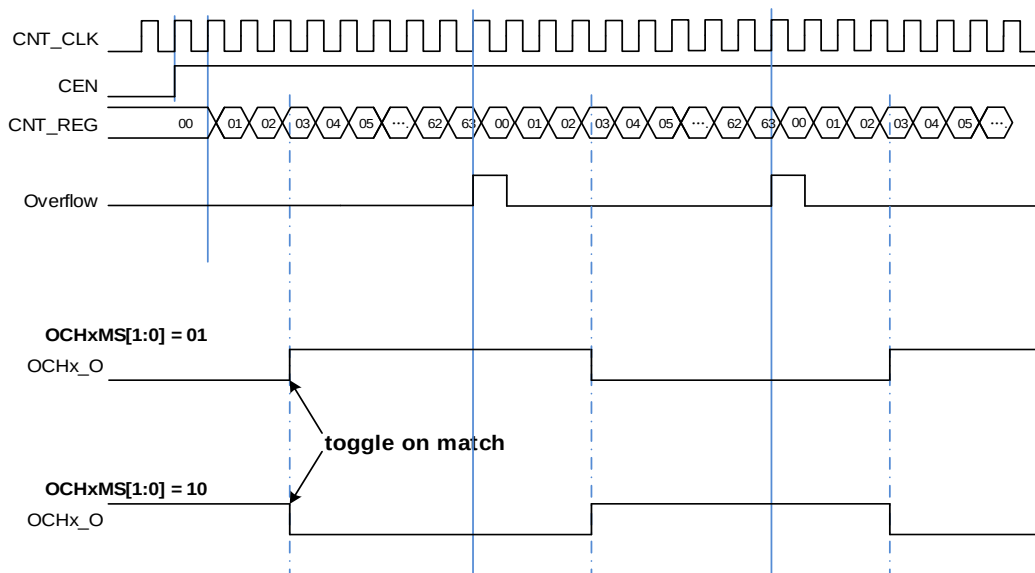
第四步：通过CPTIMERW_CAR和CPTIMERW_OCHxCV比较输出时序配置。

关于OCHxCV可以在运行时根据你所期望的波形而改变。

第五步：通过CEN启动计数器。

下面的时序图展示了两种比较模式在初始低电平或高电平状态下与比较匹配时的切换。CAR = 0x63, OCHxCV = 0x3。

图 23-5. 两种模式下的输出比较



注意：只有输出比较通道0/1可以在外部引脚上输出脉冲。输出比较通道0/1/2/3可以用来生成比较匹配中断或事件。

23.3.6. EVIC 互连

CPTIMERW 通过 EVIC 互连支持向其他模块输出事件信号或从其他模块输入事件信号。

EVIC 互连输出

EVIC 互连输出功能用于将 CPTIMERW 的中断请求作为事件信号输出给其他模块。

(1) 比较匹配事件

CPTIMERW 输出比较通道 0/1/2/3 的比较匹配事件信号，无论中断使能位的设置如何，都

可以通过 EVIC 输出到其他模块。有关更多详细信息，请参考 EVIC 章节。

EVIC 互连输入

EVIC 互连输入功能是 CPTIMERW 从其他模块接收事件信号以触发不同操作。根据 EVIC 中 EVIC_SMCFG1 和 EVIC_CPTIMERW 寄存器设置，操作和事件源可以是以下之一。

(1) 计数启用

当 EVIC 中的 EVIC_SMCFG1 寄存器选择 CPTIMERW 计数使能模式，并且由 EVIC_CPTIMERW 寄存器指定的事件源发生时，CPTIMERW 将被使能进行计数，并且 CPTIMERW_CTL0 中的 CEN 位将被设置为 1。

(2) 计数重启

当 EVIC 中的 EVIC_SMCFG1 寄存器选择 CPTIMERW 计数重启模式，并且由 EVIC_CPTIMERW 寄存器指定的事件发生时，当前计数器值 CPTIMERW_CNT 将被重置为 0。如果通道的计数器被启用，计数器将再次从 0 开始计数。

(3) 事件计数

当 EVIC 中的 EVIC_SMCFG1 寄存器选择 CPTIMERW 事件计数模式，并且计数器被启用时，CPTIMERW 计数器将在 EVIC_CPTIMERW 寄存器指定的事件每次发生时向上计数，无视计数时钟源。

23.4. CPTIMERW 寄存器

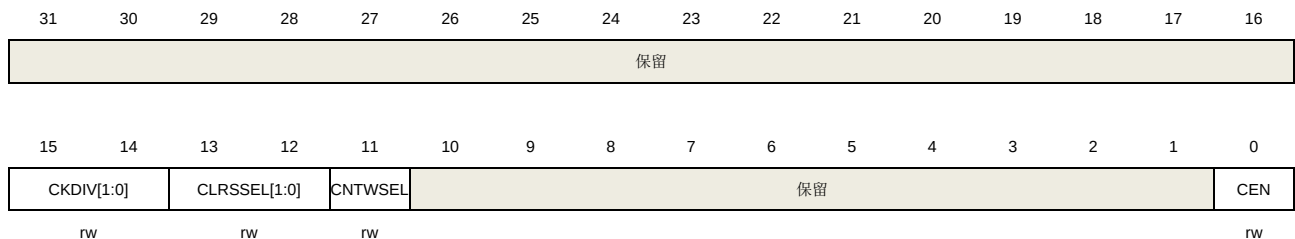
CPTIMERW 基地址: 0x4000 E000

23.4.1. 控制寄存器 0 (CPTIMERW_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
31:16	保留	必须保持复位值。
15:14	CKDIV[1:0]	输入捕获采样时钟分频 CKDIV 位可以由软件配置来指定 CK_CPTIMERW 和输入捕获数字滤波采样时钟 (ICSCK) 间的分频系数 00: $f_{ICSCK} = f_{CK\_CPTIMERW}$ 01: $f_{ICSCK} = f_{CK\_CPTIMERW} / 2$ 10: $f_{ICSCK} = f_{CK\_CPTIMERW} / 4$ 11: 保留
13:12	CLRSSEL[1:0]	计数器清零源选择 01: 计数器在输入捕获通道0捕获事件清零 10: 计数器在输入捕获通道1捕获事件清零 00 / 11: 禁用
11	CNTWSEL	计数器位宽选择 0: 32位计数模式 1: 16位计数模式
10:1	保留	必须保持复位值。
0	CEN	计数器 0 使能 0: 计数器禁能 1: 计数器使能 通过软件将CEN位置1/清零来启动/停止计数器。当在计数器使能模式下时, 也可通过EVIC事件硬件自动触发使CEN置位并启动计数器。

23.4.2. 控制寄存器 1 (CPTIMERW_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						ICH1FLT[3:0]				保留		ICH0FLT[3:0]			
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		OCH3EN	OCH2EN	OCH1EN	OCH0EN	OCH1MS[1:0]		OCH0MS[1:0]		ICH1EN	ICH0EN	ICH1ES[1:0]		ICH0ES[1:0]	
rw		rw	rw	rw	rw	rw		rw		rw	rw	rw		rw	

位/位域	名称	描述
31:26	保留	必须保持复位值。
25:22	ICH1FLT[3:0]	输入捕获通道1滤波控制 参见ICH0FLT描述
21:20	保留	必须保持复位值。
19:16	ICH0FLT[3:0]	输入捕获通道0滤波控制 一个事件计数器被用于数字滤波, N次有效采样电平将产生一次有效电平输出。此位指定了采样ICH0的时钟频率 f_{SAMP} 和滤波长度N。 0000: 滤波器禁用, $f_{SAMP}=f_{ICSCCK}$, N=1. 0001: $f_{SAMP}=f_{CK\_CPTIMERW}$, N=2. 0010: $f_{SAMP}=f_{CK\_CPTIMERW}$, N=4. 0011: $f_{SAMP}=f_{CK\_CPTIMERW}$, N=8. 0100: $f_{SAMP}=f_{ICSCCK}/2$, N=6. 0101: $f_{SAMP}=f_{ICSCCK}/2$, N=8. 0110: $f_{SAMP}=f_{ICSCCK}/4$, N=6. 0111: $f_{SAMP}=f_{ICSCCK}/4$, N=8. 1000: $f_{SAMP}=f_{ICSCCK}/8$, N=6. 1001: $f_{SAMP}=f_{ICSCCK}/8$, N=8. 1010: $f_{SAMP}=f_{ICSCCK}/16$, N=5. 1011: $f_{SAMP}=f_{ICSCCK}/16$, N=6. 1100: $f_{SAMP}=f_{ICSCCK}/16$, N=8. 1101: $f_{SAMP}=f_{ICSCCK}/32$, N=5. 1110: $f_{SAMP}=f_{ICSCCK}/32$, N=6. 1111: $f_{SAMP}=f_{ICSCCK}/32$, N=8.
15:14	保留	必须保持复位值。
13	OCH3EN	输出比较通道3使能 参见OCH0EN描述
12	OCH2EN	输出比较通道2使能

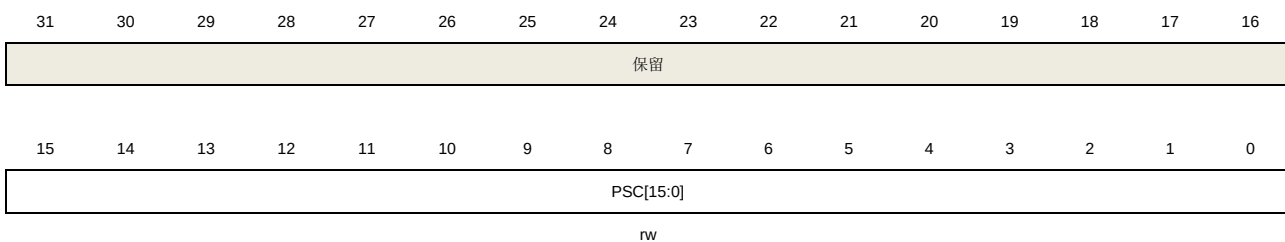
		参见OCH0EN描述
11	OCH1EN	输出比较通道1使能 参见OCH0EN描述
10	OCH0EN	输出比较通道0使能 0: 禁止 1: 使能
9:8	OCH1MS[1:0]	输出比较通道1模式选择 此位指定输出比较通道的输出模式 00: 输出电平保持不变 01: 初始电平为低, 比较匹配时翻转 10: 初始电平为高, 比较匹配时翻转 11: 输出禁止
7:6	OCH0MS[1:0]	输出比较通道0模式选择 此位指定输出比较通道的输出模式 00: 输出电平保持不变 01: 初始电平为低, 比较匹配时翻转 10: 初始电平为高, 比较匹配时翻转 11: 输出禁止
5	ICH1EN	输入捕获通道1使能 0: 禁止 1: 使能
4	ICH0EN	输入捕获通道0使能 0: 禁止 1: 使能
3:2	ICH1ES[1:0]	输入捕获通道1边沿选择 00: 上升沿 01: 下降沿 10: 双边沿 11: 禁止
1:0	ICH0ES[1:0]	输入捕获通道0边沿选择 00: 上升沿 01: 下降沿 10: 双边沿 11: 禁止

23.4.3. 预分频器寄存器 (CPTIMERW_PSC)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器只能按字（32 位）访问。



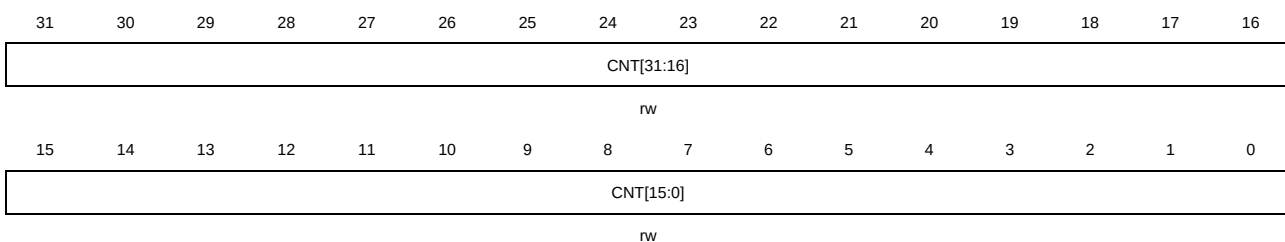
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	PSC[15:0]	时钟预分频值 计数器时钟等于CPTIMERW_CK时钟除以（PSC+1），每次当更新事件产生时，PSC的值被装入到对应的影子寄存器。

23.4.4. 计数器寄存器（CPTIMERW_CNT）

地址偏移：0x0C

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



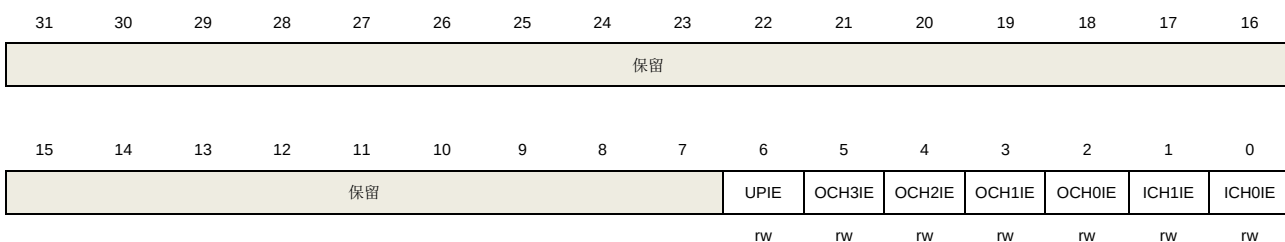
位/位域	名称	描述
31:0	CNT[31:0]	这些位是计数器当前的计数值。写操作能改变计数器值。

23.4.5. 中断使能寄存器（CPTIMERW_INTEN）

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
31:7	保留	必须保持复位值。
6	UPIE	更新中断使能 0: 禁止更新中断 1: 使能更新中断
5	OCH3IE	输出比较通道3中断使能 参见OCH0IE描述
4	OCH2IE	输出比较通道2中断使能 参见OCH0IE描述
3	OCH1IE	输出比较通道1中断使能 参见OCH0IE描述
2	OCH0IE	输出比较通道0中断使能 0: 禁止中断 1: 使能中断
1	ICH1IE	输入捕获通道1中断使能 参见ICH0IE描述
0	ICH0IE	输入捕获通道0中断使能 0: 禁止中断 1: 使能中断

23.4.6. 中断标志寄存器（CPTIMERW_INTF）

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									UPIF	OCH3IF	OCH2IF	OCH1IF	OCH0IF	ICH1IF	ICH0IF
									rc_w0	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	UPIF	更新中断标志 此位在更新事件发生时由硬件置 1，软件可以写 0 清 0。 更新事件包含计数器清零，计数器溢出，EVIC 计数器重启动和软件更新事件。 0: 无更新中断发生

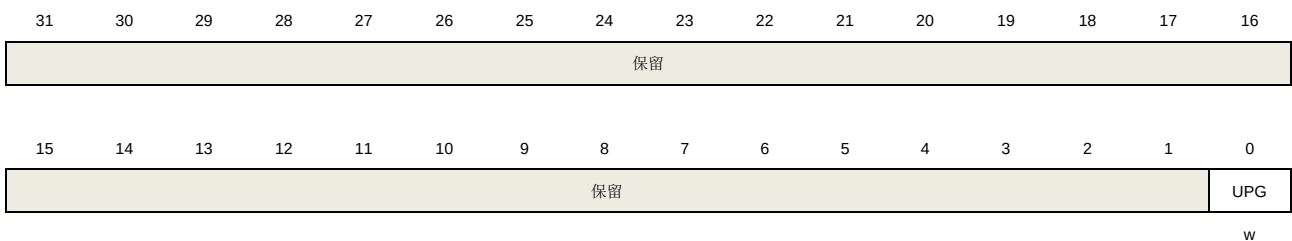
		1: 发生更新中断
5	OCH3IF	输出比较通道3中断标志 参见OCH0IF描述
4	OCH2IF	输出比较通道2中断标志 参见OCH0IF描述
3	OCH1IF	输出比较通道1中断标志 参见OCH0IF描述
2	OCH0IF	输出比较通道0中断标志 此位在比较匹配事件发生时由硬件置 1，软件可以写 0 清 0。 0: 无比较匹配中断发生 1: 发生了比较匹配中断
1	ICH1IF	输入捕获通道1中断标志 参见ICH0IF描述
0	ICH0IF	输入捕获通道0中断标志 此位在比较匹配事件发生时由硬件置 1，软件可以写 0 清 0。 0: 无捕获中断发生 1: 发生了捕获中断

23.4.7. 软件事件生成寄存器（CPTIMERW_SWEVG）

地址偏移：0x18

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



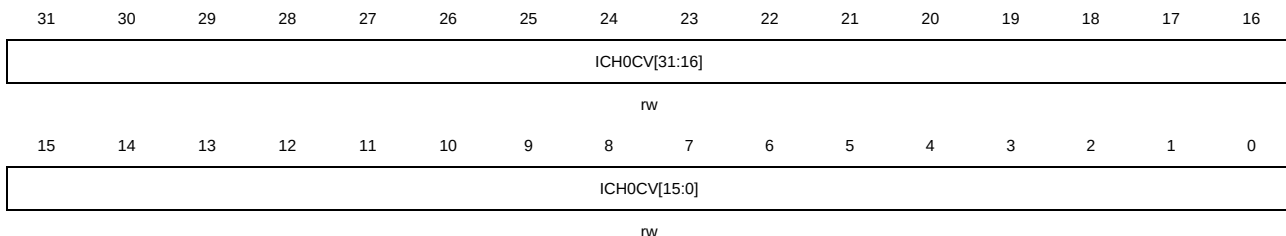
位/位域	名称	描述
31:1	保留	必须保持复位值。
0	UPG	更新事件产生 此位由软件置 1，被硬件自动清 0。当此位被置 1，计数器被清 0，预分频计数器将同时被清除。 0: 无更新事件产生 1: 产生更新事件

23.4.8. 输入捕获通道 0 捕获值寄存器 (CPTIMERW_ICH0CV)

地址偏移: 0x1C

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



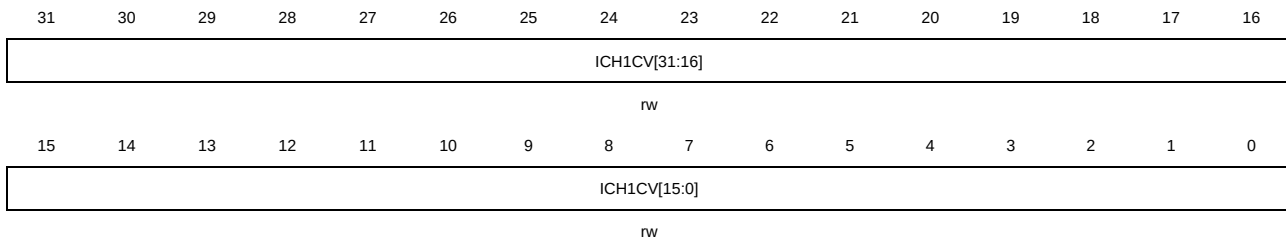
位/位域	名称	描述
31:0	ICH0CV[31:0]	输入捕获通道0的捕获值 此位域为输入捕获通道0捕获事件发生时的捕获的计数值

23.4.9. 输入捕获通道 1 捕获值寄存器 (CPTIMERW_ICH1CV)

地址偏移: 0x20

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



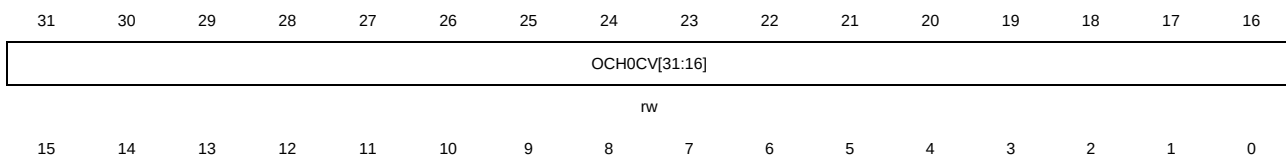
位/位域	名称	描述
31:0	ICH1CV[31:0]	输入捕获通道1的捕获值 此位域为输入捕获通道1捕获事件发生时的捕获的计数值

23.4.10. 输出比较通道 0 比较值寄存器 (CPTIMERW_OCH0CV)

地址偏移: 0x24

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问。



OCH0CV[15:0]		
rw		

位/位域	名称	描述
31:0	OCH0CV[31:0]	输出比较通道0比较值 此位域指定比较值用于和计数器值进行比较产生比较匹配事件。当计数器位宽是16位时，[31:16]位应该被设为0。

23.4.11. 输出比较通道 1 比较值寄存器（CPTIMERW_OCH1CV）

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
OCH1CV[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OCH1CV[15:0]															
rw															

位/位域	名称	描述
31:0	OCH1CV[31:0]	输出比较通道1比较值 此位域指定比较值用于和计数器值进行比较产生比较匹配事件。当计数器位宽是16位时，[31:16]位应该被设为0。

23.4.12. 输出比较通道 2 比较值寄存器（CPTIMERW_OCH2CV）

地址偏移：0x2C

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OCH2CV[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OCH2CV[15:0]															
rw															

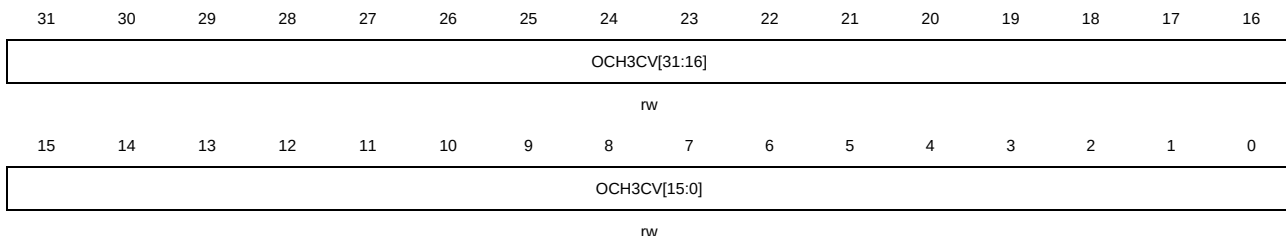
位/位域	名称	描述
31:0	OCH2CV[31:0]	输出比较通道2比较值 此位域指定比较值用于和计数器值进行比较产生比较匹配事件。当计数器位宽是16位时，[31:16]位应该被设为0。

23.4.13. 输出比较通道 3 比较值寄存器 (CPTIMERW_OCH3CV)

地址偏移: 0x30

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



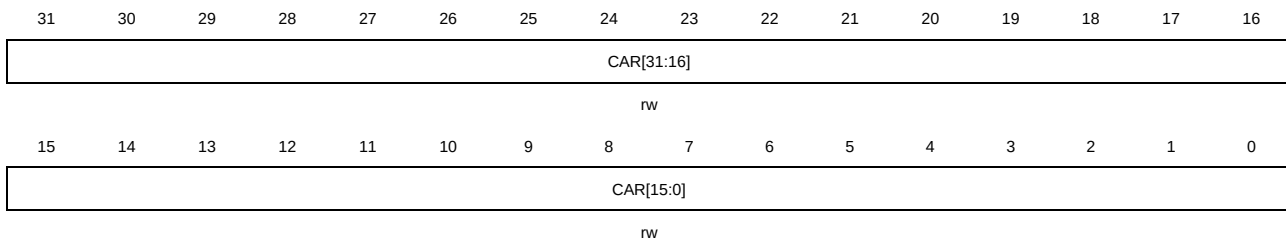
位/位域	名称	描述
31:0	OCH3CV[31:0]	输出比较通道3比较值 此位域指定比较值用于和计数器值进行比较产生比较匹配事件。当计数器位宽是16位时，位[31:16]应被设为0。

23.4.14. 计数器自动重载寄存器 (CPTIMERW_CAR)

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
31:0	CAR[31:0]	计数器自动重载值 此位域指定计数器自动重载值。当计数器位宽是16位时，位[31:16]应被设为0。

24. 比较匹配定时器（CPTIMER）

24.1. 简介

比较匹配定时器模块（CPTIMER0/1）有两个独立的 16 位无符号计数器，两个模块总共有 4 个计数器。比较匹配定时器可以被配置以设定的周期生成中断或 EVIC 请求。

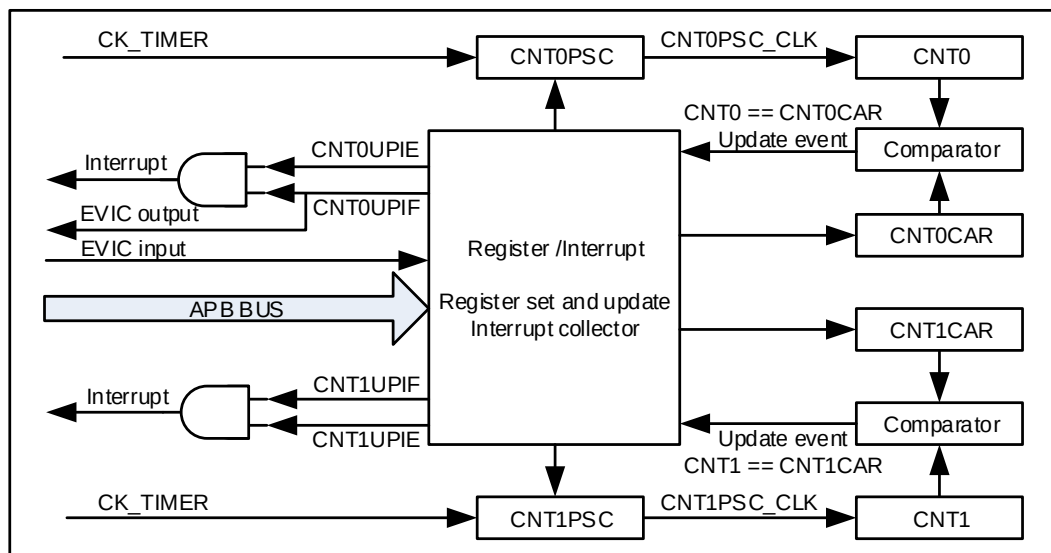
24.2. 主要特征

- 总计数器：每个模块 2 个计数器；
- 计数器宽度：16 位；
- 时钟源：仅限内部时钟；
- 计数模式：仅向上计数；
- 可编程预分频器：16 位。分频系数可以实时更改；
- 自动重装功能；
- 计数器溢出事件可输出到 EVIC 作为触发源；
- 更新事件可产生中断；

24.3. 功能说明

[图24-1. 比较匹配定时器框图](#)提供了比较匹配定时器内部配置的详细信息。

图 24-1. 比较匹配定时器框图



24.3.1. 时钟源

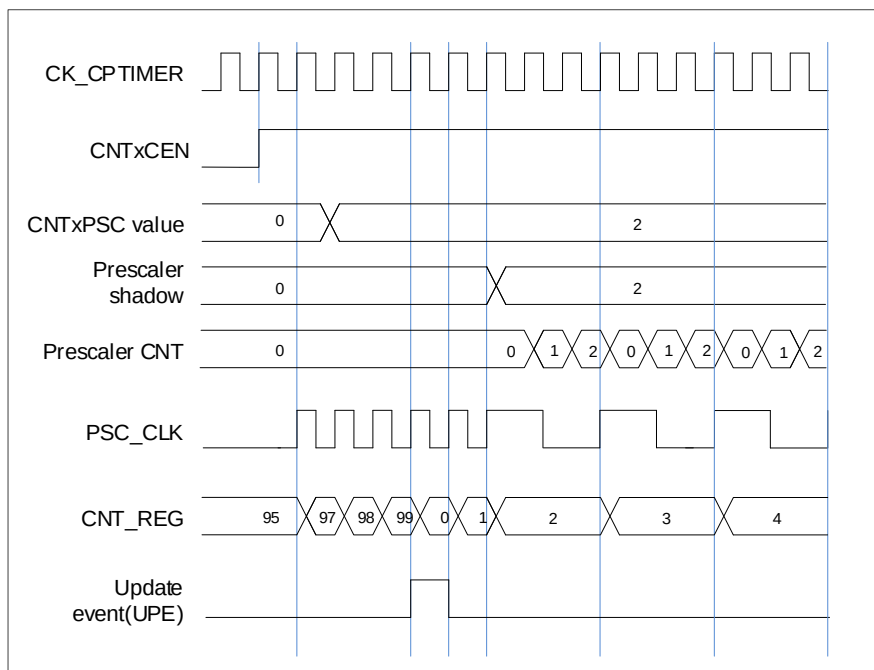
CPTIMER 只能由内部定时器时钟驱动，该时钟来自 RCU 中的名为 CK_CPTIMERx 的源。驱

动计数器预分频器的 $CPTIMER\_CK$ 等于 $CK\_CPTIMER$ 。当 $CENx$ 被设置时， $CPTIMER\_CK$ 经过预分频器（预分频值由 $CNTxPSC$ 寄存器确定）产生 $CNTxPSC\_CLK$ 。

24.3.2. 时钟预分频器

分频器可以将定时器时钟（ $CPTIMER\_CK$ ）频率按 1 到 65536 之间的任意值分频，分频后的时钟 $CNTxPSC\_CLK$ 驱动计数器计数。分频系数受预分频寄存器（ $CPTIMER\_CNTxPSC$ ）控制，这个控制寄存器带有影子寄存器，它能够在运行时被改变。新的预分频器的值在下一个更新事件到来时被采用。

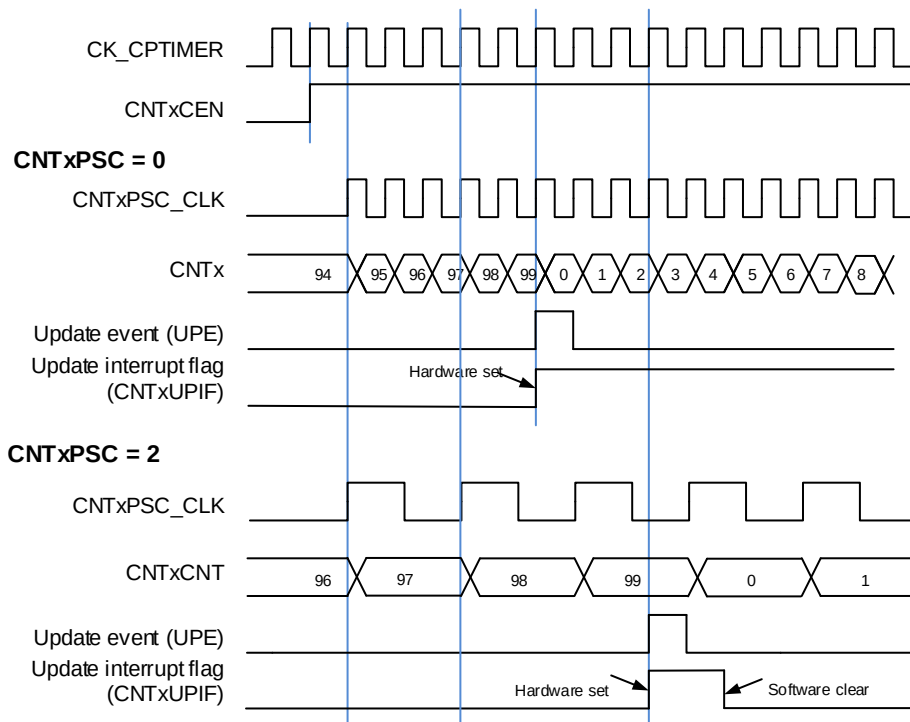
图 24-2. $CNTxPSC$ 值从 0 变化到 2 的时序图， $CNTxCAR=99$



24.3.3. 计数器向上计数模式

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 $CPTIMER\_CNTxCAR$ 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，每次计数器溢出时都会生成更新事件。

下面这些图给出了一些例子，当 $CPTIMER\_CNTxCAR=0x99$ 时，计数器在不同预分频因子下的行为。

图 24-3. 向上计数时序图, $CNTxPSC=0/2$ 

24.3.4. EVIC 互连

CPTIMER 通过 EVIC 互连支持事件信号输出到其他模块或从其他模块输入。

注意：只有 CPTIMER0 的计数器 0 支持 EVIC 互连。

EVIC 互连输出

EVIC 互连输出功能用于将 CPTIMER0 的中断请求作为事件信号输出给其他模块。CPTIMER0 的计数器 0 溢出事件信号可以通过 EVIC 输出到其他模块，而不受中断使能位的设置影响。有关更多详细信息，请参考 EVIC 章节。

EVIC 互连输入

EVIC 互连输入功能是 CPTIMER0 从其他模块接收事件信号以触发不同操作。基于 EVIC 模块 EVIC_SMCFG0 和 EVIC_CPTIMER0 寄存器的设置，操作和事件源可以是以下之一。

(1) 计数使能

当 EVIC 中的 EVIC_SMCFG0 寄存器选择 CPTIMER0 计数使能模式，并且 EVIC_CPTIMER0 寄存器指定的事件发生时，CPTIMER0 计数器 0 将被启用进行计数，并且 CPTIMER_CTL0 中的 CEN0 位被设置为 1。

(2) 计数重启

当 EVIC 中的 EVIC_SMCFG0 寄存器选择 CPTIMER0 计数器的重启模式，并且 EVIC_CPTIMER0 寄存器指定的事件发生时，当前计数器值 CNT0 将被重置为 0，并生成

一个更新事件。如果计数器被启用，计数器将从 0 重新开始计数。

(3) 事件计数

当 EVIC 中的 EVIC_SMCFG0 寄存器选择 CPTIMER0 事件计数模式，并且计数器被启用时，CPTIMER 计数器 0 将在 EVIC_CPTIMER0 寄存器指定的事件每次发生时向上计数，无视计数时钟源。

24.4. CPTIMER 寄存器

CPTIMER0 基地址: 0x4000 0000

CPTIMER1 基地址: 0x4000 0400

24.4.1. 控制寄存器 0 (CPTIMER_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															CEN1
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															CEN0
															rw

位/位域	名称	描述
31:17	保留	必须保持复位值。
16	CEN1	计数器 1 使能 0: 计数器禁能 1: 计数器使能 只能通过软件将CEN1位置1/清零来启动/停止计数器1。
15:1	保留	必须保持复位值。
0	CEN0	计数器 0 使能 0: 计数器禁能 1: 计数器使能 通过软件将CEN0位置1/清零来启动/停止计数器。当在计数器使能模式下时, 也可通过EVIC事件硬件自动触发使CEN0置位并启动计数器0。

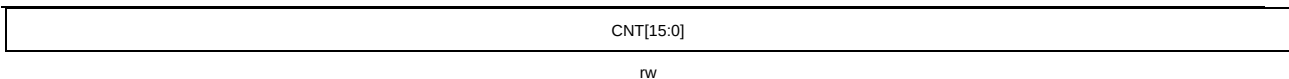
24.4.2. 计数器 0 寄存器 (CPTIMER_CNT0)

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



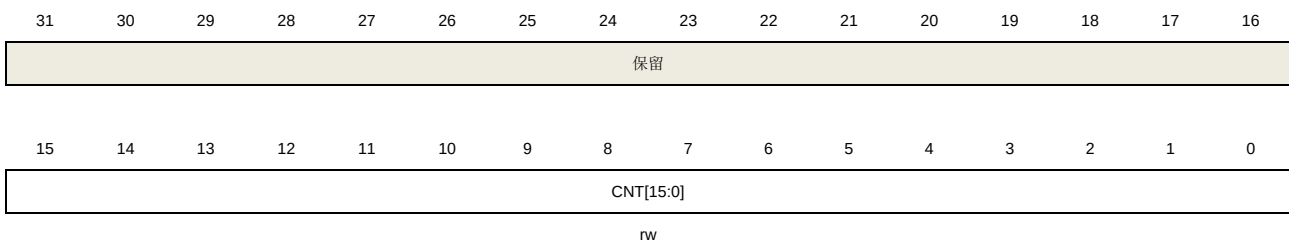
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CNT[15:0]	这些位是计数器0当前的计数值。写操作能改变计数器值。

24.4.3. 计数器 1 寄存器（CPTIMER_CNT1）

地址偏移：0x08

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



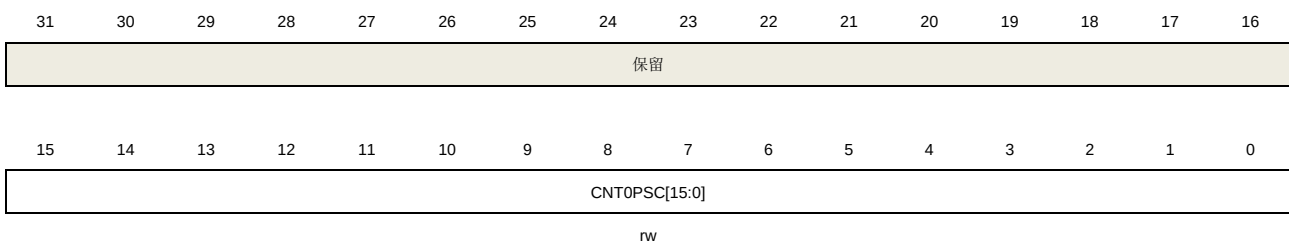
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CNT[15:0]	这些位是计数器1当前的计数值。写操作能改变计数器值。

24.4.4. 计数器 0 预分频器寄存器（CPTIMER_CNT0PSC）

地址偏移量：0x0C

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



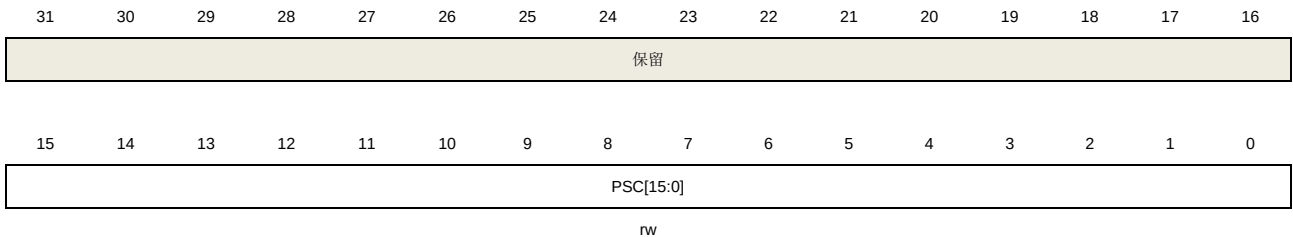
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CNT0PSC[15:0]	计数器 0 时钟预分频值 计数器0时钟等于CPTIMER_CK时钟除以（CNT0PSC+1），每次当更新事件产生时，CNT0PSC的值被装入到对应的影子寄存器。

24.4.5. 计数器 1 预分频器寄存器 (CPTIMER_CNT1PSC)

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。



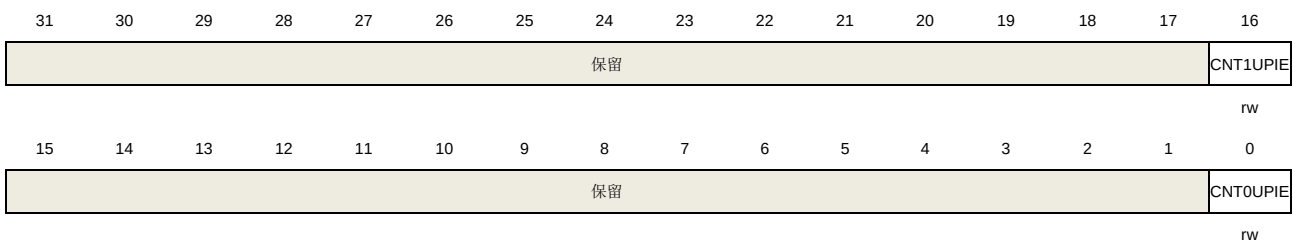
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CNT1PSC[15:0]	计数器 1 时钟预分频值 计数器1时钟等于CPTIMER_CK时钟除以 (CNT1PSC+1)，每次当更新事件产生时，CNT1PSC的值被装入到对应的影子寄存器。

24.4.6. 中断使能寄存器 (CPTIMER_INTEN)

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
31:17	保留	必须保持复位值。
16	CNT1UIPIE	计数器 1 更新中断使能 0: 禁止更新中断 1: 使能更新中断
15:1	保留	必须保持复位值。
0	CNT0UIPIE	计数器 0 更新中断使能 0: 禁止更新中断 1: 使能更新中断

24.4.7. 中断标志寄存器（CPTIMER_INTF）

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															CNT1UPIF
															rc_w0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															CNT0UPIF
															rc_w0

位/位域	名称	描述
31:17	保留	必须保持复位值。
16	CNT1UPIF	计数器 1 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。更新事件包含计数器溢出，软件更新事件。 0：无更新中断发生 1：发生更新中断
15:1	保留	必须保持复位值。
0	CNT0UPIF	计数器 0 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。更新事件包含计数器溢出，软件更新事件和EVIC重启动事件。 0：无更新中断发生 1：发生更新中断

24.4.8. 软件事件生成寄存器（CPTIMER_SWEVG）

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															CNT1UPG
															w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留															CNT0UPG
															w

位/位域	名称	描述
31:17	保留	必须保持复位值。

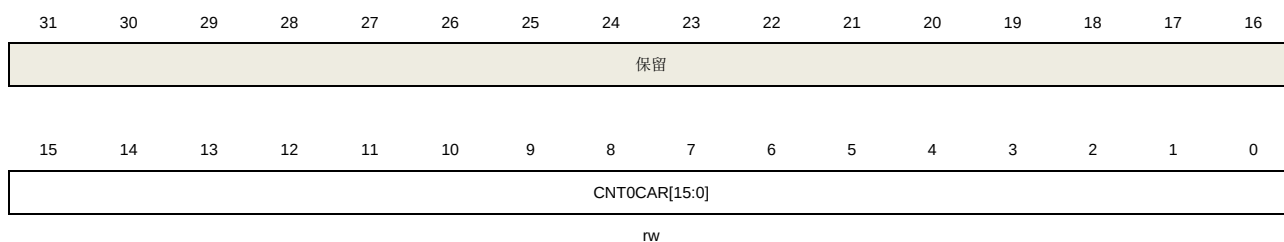
16	CNT1UPG	计数器 1 更新事件产生 此位由软件置 1，被硬件自动清 0。当此位被置 1，计数器被清 0，预分频计数器将同时被清除。 0：无更新事件产生 1：产生更新事件
15:1	保留	必须保持复位值。
0	CNT0UPG	计数器 0 更新事件产生 此位由软件置 1，被硬件自动清 0。当此位被置 1，计数器被清 0，预分频计数器将同时被清除。 0：无更新事件产生 1：产生更新事件

24.4.9. 计数器 0 自动重载寄存器 (CPTIMER_CNT0CAR)

地址偏移：0x20

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



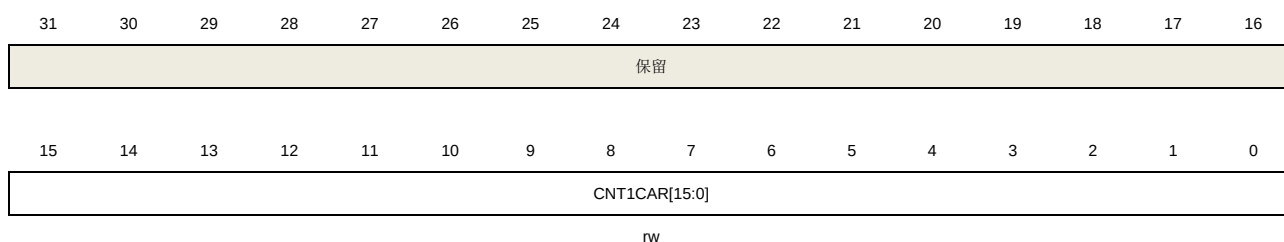
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CNT0CAR[15:0]	计数器 0 自动重载值 这些位定义了计数器的自动重载值。

24.4.10. 计数器 1 自动重载寄存器 (CPTIMER_CNT1CAR)

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	CNT1CAR[15:0]	计数器 0 自动重载值 这些位定义了计数器的自动重载值。

25. 通用异步收发器（UART）

25.1. 简介

通用异步收发器（UART）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异步的方式进行传输。UART提供了可编程的波特率发生器，能对PCLK1进行分频产生UART发送和接收所需的特定频率。

UART不仅支持标准的异步收发模式，还实现了一些其它类型的串行数据交换模式，如红外编码规范，SIR，LIN模式等。它还支持多处理器通信。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。

UART支持DMA功能，以实现高速率的数据通信。

25.2. 主要特征

- NRZ标准格式；
- 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，UART0/1/2/3由PCLK1分频得到；
 - 8倍或16倍过采样；
 - 最高速度可到12Mbits/s（当PCLK1时钟频率为96M，8倍过采样模式）；
- 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（7，8，9或10位）；
 - 产生1或者2个停止位；
- 发送器和接收器可分别使能；
- 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）。
- 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检测从静默模式唤醒；
- 多种状态标志：
 - 传输检测标志：接收缓冲区不为空（RBNE），发送缓冲区为空（TBE），传输完成（TC），忙（BSY）；
 - 错误检测标志：过载错误（ORERR），噪声错误（NERR），帧格式错误（FERR），奇偶校验错误（PERR）；
 - 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）；
 - 若相应的中断使能，这些事件发生将会触发中断；

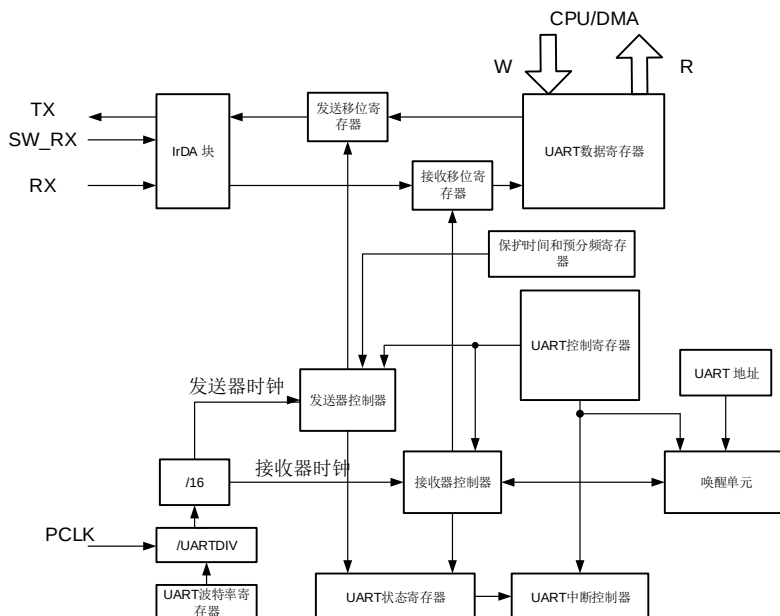
25.3. 功能说明

UART 接口通过[表 25-1. UART 重要引脚描述](#)中主要引脚从外部连接到其他设备。

表 25-1. UART 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）	发送数据。当 UART 使能后，若无数据发送，默认为高电平

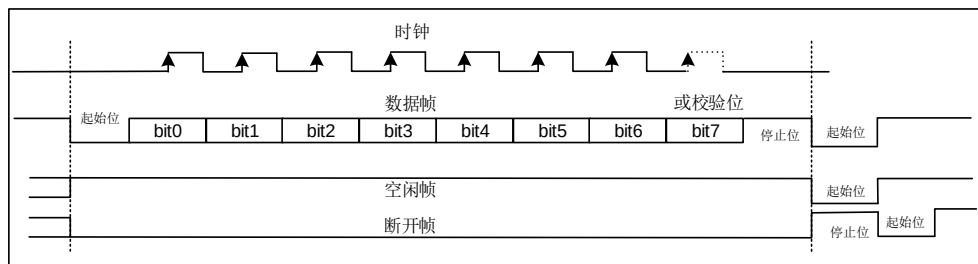
图 25-1. UART 模块内部框图



25.3.1. UART 帧格式

UART 数据帧开始于起始位，结束于停止位。UART_CTL0 寄存器中 WL 位可以设置数据长度。将 UART_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 0b00，第七位为校验位。若 WL 位置 0b01，第八位为校验位。若 WL 位为 0b10，第六位为校验位。若 WL 位置 0b11，第九位为校验位。UART_CTL0 寄存器中 PM 位用于选择校验位的计算方法。

图 25-2. UART 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 UART_CTL1 寄存器中 STB[1:0] 位域配置。

表 25-2. 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	/
10	2	标准 UART，单线以及调制解调模式
11	1.5	/

在一个空闲帧中，所有位都为1。数据帧长度与正常UART数据帧长度相同。

紧随停止位后多个低电平为中断帧。UART数据帧的传输速度由PCLK时钟频率，波特率发生器的配置，以及过采样模式共同决定。

25.3.2. 波特率发生

波特率分频系数是一个20位的数字，包含16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UART能够产生所有标准波特率。

波特率分频系数（UARTDIV）与系统时钟具有如下关系：

如果过采样率是16，公式为：

$$\text{UARTDIV} = \frac{\text{PCLK}}{16 \times \text{Baud Rate}} \quad (25-1)$$

如果过采样率是8，公式为：

$$\text{UARTDIV} = \frac{\text{PCLK}}{8 \times \text{Baud Rate}} \quad (25-2)$$

例如，当过采样率是16时：

1. 由UART_BAUD寄存器的值得到UARTDIV：
假设UART_BAUD=0x21D，则INTDIV=33（0x21），FRADIV=13（0xD）。
UASRTDIV=33+13/16=33.81。
2. 由UARTDIV得到UART_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）
UART_BAUD=0x1E6。

注意：若取整后FRADIV=16（溢出），则进位必须加到整数部分。

25.3.3. UART 发送器

如果UART_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UART_CTL2寄存器中TINV位来配置。

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。

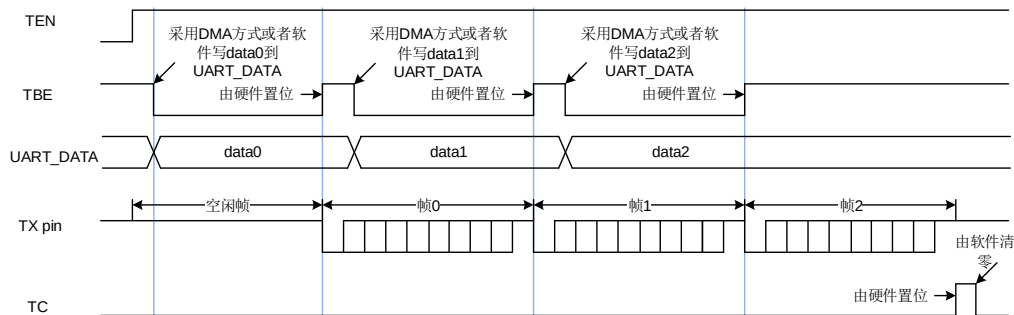
系统上电后，TBE默认为高电平。在UART_STAT0寄存器中TBE置位时，数据可以在不覆盖前一个数据的情况下写入UART_DATA寄存器。当数据写入UART_DATA寄存器，TBE位将被清0。在数据由UART_DATA移入移位寄存器后，该位由硬件置1。如果数据在一个发送过程正在进行时被写入UART_DATA寄存器，它将首先被存入发送缓冲区，在当前发送过程完成时传输到发送移位寄存器中。如果数据在写入UART_DATA寄存器时，没有发送过程正在进行，TBE位将被清零然后迅速置位，原因是数据将立刻传输到发送移位寄存器。

假如一帧数据已经发送出去，并且TBE位已经置位，那么UART_STAT0寄存器中TC位将被置1。如果UART_CTL0寄存器中的中断使能位（TCIE）为1，将会产生中断。

图 25-3. UART 发送步骤给出了 UART 发送步骤。软件操作按以下流程进行：

1. 通过UART_CTL0中的WL0和WL1设置字长；
2. 在UART_CTL1寄存器中写STB[1:0]位来设置停止位的长度；
3. 如果选择了多级缓存通信方式，应该在UART_CTL2寄存器中使能DMA（DENT位）；
4. 在UART_BAUD寄存器中设置波特率；
5. 在UART_CTL0寄存器中置位UEN位，使能UART；
6. 在UART_CTL0寄存器中设置TEN位；
7. 等待TBE置位；
8. 向UART_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 25-3. UART 发送步骤



在禁用UART或进入低功耗状态之前，必须等待TC置位。先读UART_STAT0然后再写UART_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。

25.3.4. UART 接收器

上电后，UART接收器使能按以下步骤进行：

1. 通过UART_CTL0中的WL0和WL1设置字长；
2. 在UART_CTL1寄存器中写STB[1:0]位来设置停止位的长度；
3. 如果选择了多级缓存通信方式，应该在UART_CTL2寄存器中使能DMA（DENR位）；
4. 在UART_BAUD寄存器中设置波特率
5. 在UART_CTL0寄存器中置位UEN位，使能UART；
6. 在UART_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。

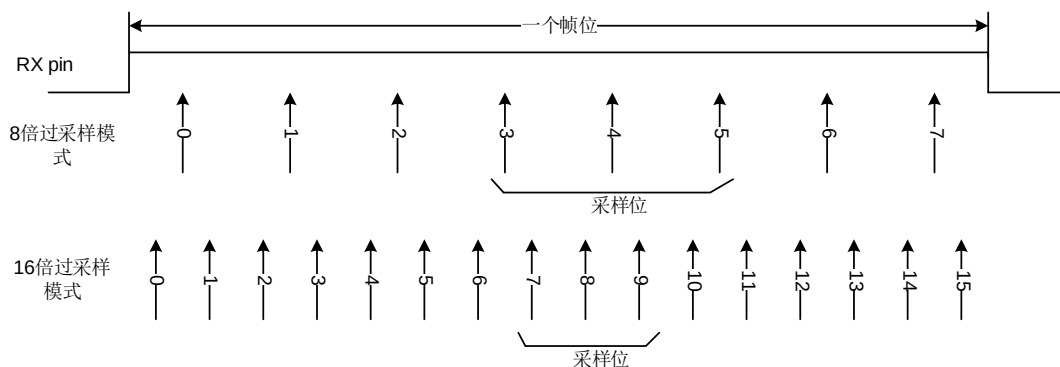
当接收到一个数据帧，UART_STAT0寄存器中的RBNE置位，如果设置了UART_CTL0寄存器中相应的中断使能位RBNEIE，将会产生中断。在UART_STAT0寄存器中可以观察接收状态标志。

软件可以通过读UART_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UART_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。如果启用了8倍过采样模式，则使用第1，第2和第3个采样点；而在16倍过采样模式下，则使用第6，第7和第8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UART_CTL2寄存器中ERRIE，将会产生中断。如果UART_CTL2寄存器中的OSB位被置位，接收器将只读取一个样本来评估比特值。在这种情况下，不会检测到噪声错误。

图 25-4. 过采样方式接收一个数据位（OSB = 0）



通过置位UART_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，UART_STAT0寄存器中PERR被置位。如果设置了UART_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，UART_STAT0寄存器中FERR置位。如果使能DMA并置位UART_CTL2寄存器中ERRIE位，将产生中断。

在RBNE位还没有被清零情况下接收到一帧数据，随后的数据帧将不会存储在数据接收缓冲区中。UART_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UART_CTL2寄存器中ERRIE位或者置位RBNEIE，将产生中断。

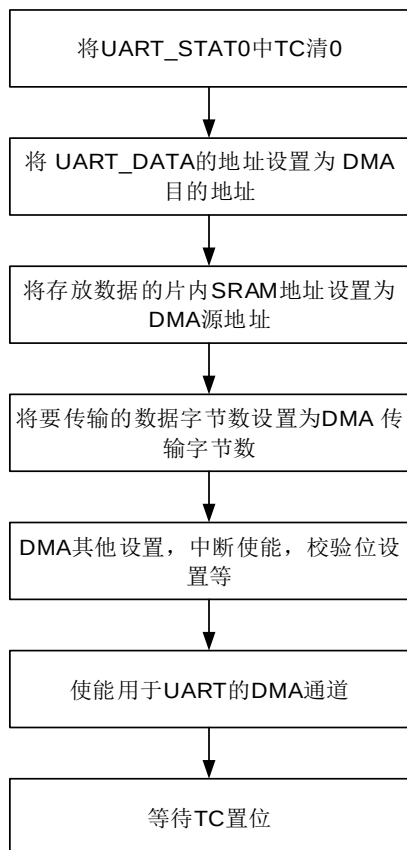
若接收过程中，产生了噪声错误（NERR）、校验错误（PERR）、帧错误（FERR）或溢出错误（ORERR），则NERR、PERR、FERR或ORERR将和RBNE同时置位。如果没有使能DMA，RBNE中断发生时，软件需检查是否有噪声错误、校验错误、帧错误或溢出错误产生。

25.3.5. 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UART_CTL2寄存器中DENT位可以使能DMA发送，置位UART_CTL2寄存器中DENR位可以使能DMA接收。

当DMA用于UART发送时，DMA将数据从片内SRAM传送到UART的数据缓冲区。配置步骤如[图25-5. 采用DMA方式实现UART数据发送配置步骤](#)所示。

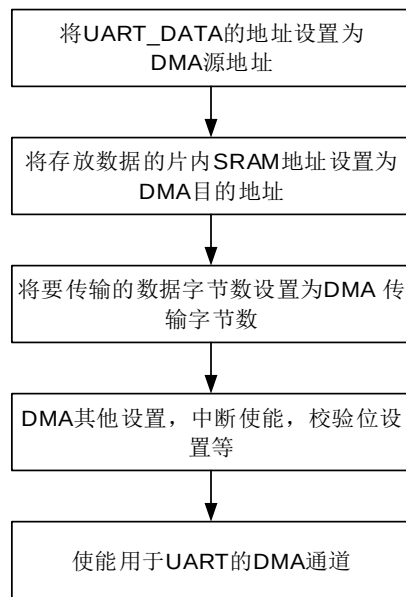
图 25-5. 采用 DMA 方式实现 UART 数据发送配置步骤



所有数据帧都传输完成后，UART_STAT0寄存器中TC位置1。如果UART_CTL0寄存器中TCIE置位，将产生中断。

当DMA用于UART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25-6. 采用DMA方式实现UART数据接收配置步骤](#)所示。如果将UART_CTL2寄存器中ERRIE位置1，UART_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 25-6. 采用 DMA 方式实现 UART 数据接收配置步骤



当UART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。

25.3.6. 多处理器通信

在多处理器通信中，多个UART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UART_CTL0寄存器中RWU位置1使一个UART进入静默模式。

如果UART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UART。

此外，UART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UART。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。

当UART_CTL0寄存器中WM被置位，数据最高位会被认为是地址标志位。如果地址标志位为1，该字节被认为是地址字节。如果地址标志位为0，该字节被认为是数据字节。如果地址字节的低4位与UART_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UART唤醒的数据帧，RBNE将置位。状态标志可以从UART_STAT0寄存器中获取。如果地址字节的低4位与UART_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，接收器对地址字节不做奇偶校验。如果UART_CTL0寄存器中PCEN位被置位。地址字节最高位被视为校验位，其余位被视为地址位。

当接收器缓冲区不含数据（UART_STAT0寄存器中的RBNE=0）时，RWU位可以被写为0或1。

25.3.7. LIN 模式

将UART_CTL1寄存器的LMEN置位即可使能本地互连网络模式。

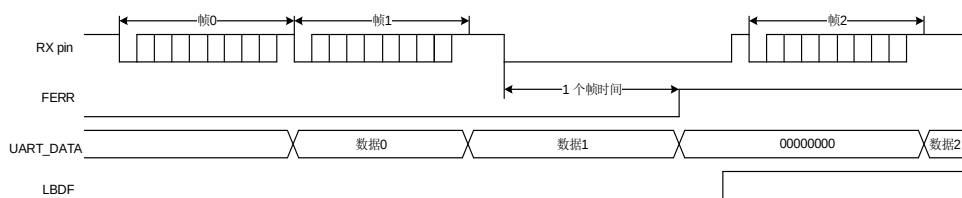
在LIN模式下，UART_CTL1寄存器中WL0，WL1，STB[1:0]以及UART_CTL2的HDEN，IREN位都应该被清0。

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UART_CTL0寄存器中SBKCMD置位时，UART在连续发送13个0，再发送完一个停止位。

断开检测功能完全独立于普通UART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。UART_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），UART_STAT0寄存器中LBDF置位。如果UART_CTL1寄存器中LBDIE被置位，将产生中断。

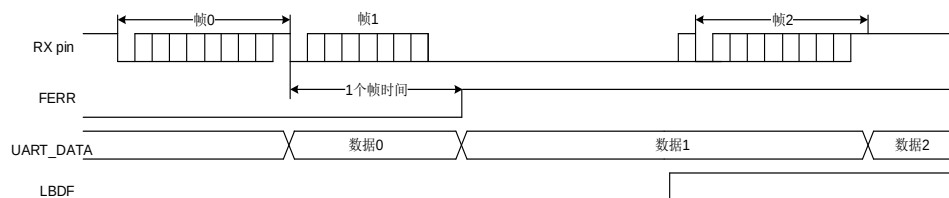
如[图25-7. 空闲状态下检测断开帧](#)所示，如果断开帧发生在空闲状态下，UART接收器会接收到一个全0数据帧，同时FERR置位。

图 25-7. 空闲状态下检测断开帧



如[图25-8. 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 25-8. 数据传输过程中检测断开帧

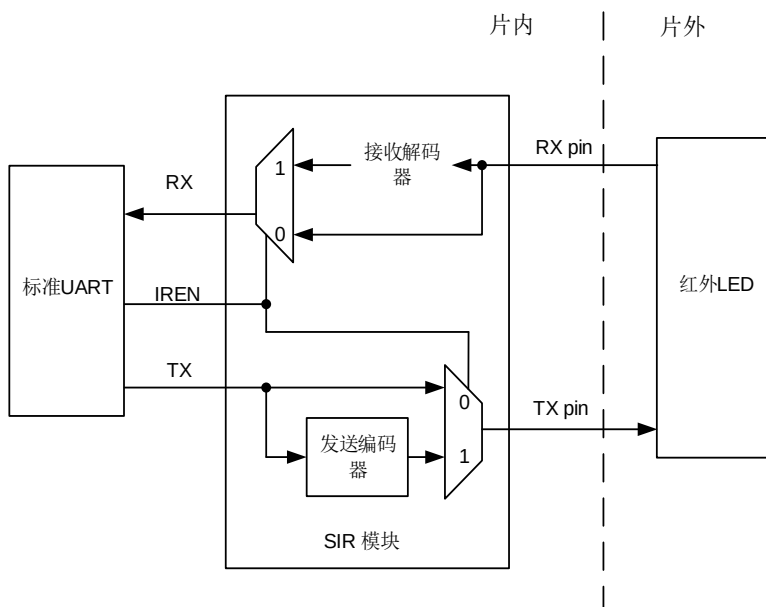


25.3.8. 串行红外（IrDA SIR）编解码功能模块

串行红外编解码功能通过置位UART_CTL2寄存器中IREN使能。在IrDA模式下，UART_CTL1寄存器的LMEN，STB[1:0]位和UART_CTL2寄存器的HDEN位将被清0。

在IrDA模式下，UART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UART接收器。对于编码器而言，波特率应小于115200。

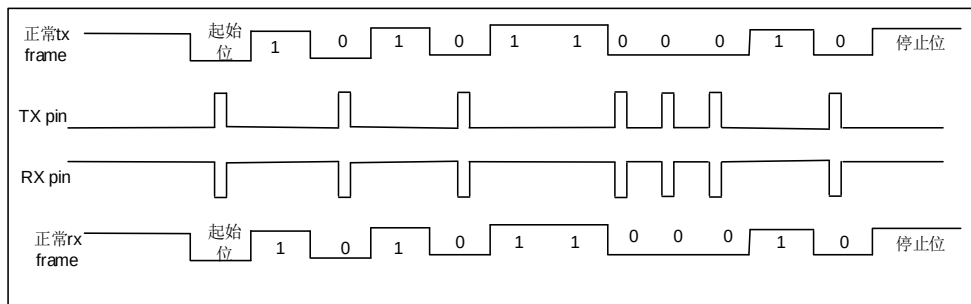
图 25-9. IrDA SIR ENDEC 模块



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 25-10. IrDA 数据调制



将UART_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UART_GP寄存器中PSC[7:0]位配置，UART_BAUD寄存器需配置为 $16 \times \text{PSC}[7:0]$ 。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。

25.3.9. 半双工通信模式

通过设置UART_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，UART_CTL1寄存器的LMEN位和UART_CTL2寄存器的IREN位清零。

半双工模式下，TX引脚和RX引脚将从内部连接到一起，RX引脚不再使用。TX引脚应该被配置

为开漏输出模式。通信冲突由软件处理。

25.3.10. UART 中断

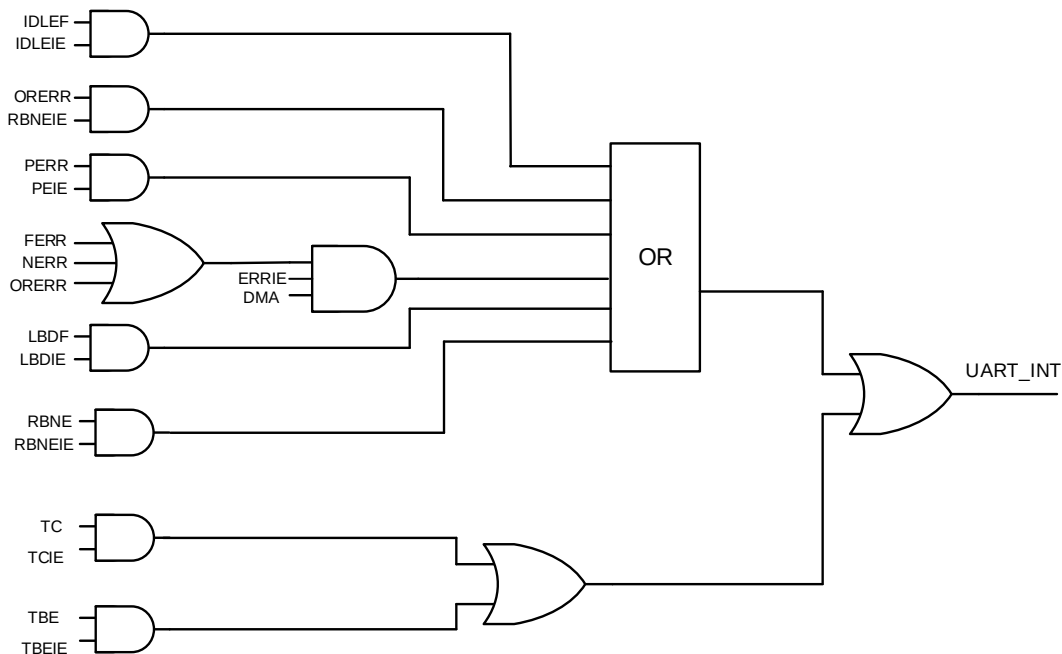
UART 中断事件和标志如[表 25-3. UART 中断请求](#)所示：

表 25-3. UART 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	UART_CTL0	TBEIE
发送结束	TC	UART_CTL0	TCIE
接收到的数据可以读取	RBNE	UART_CTL0	RBNEIE
检测到过载错误	ORERR		
检测到线路空闲	IDLEF	UART_CTL0	IDLEIE
奇偶校验错误	PERR	UART_CTL0	PERRIE
LIN模式下，检测到断开标志	LBDF	UART_CTL1	LBDIE
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	UART_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 UART 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 25-11. UART 中断映射框图



25.4. UART 寄存器

UART0 基地址: 0x4001 4C00

UART1 基地址: 0x4001 5000

UART2 基地址: 0x4001 7800

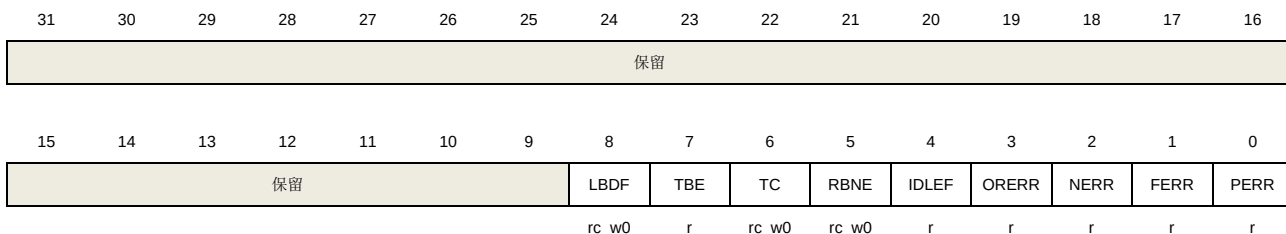
UART3 基地址: 0x4000 7C00

25.4.1. 状态寄存器 0 (UART_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
31:9	保留	必须保持复位值。
8	LBDF	LIN断开检测标志 寄存器UART_CTL1寄存器中LMEN置位，说明检测到LIN断开。如果UART_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。UART_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UART_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。UART_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。

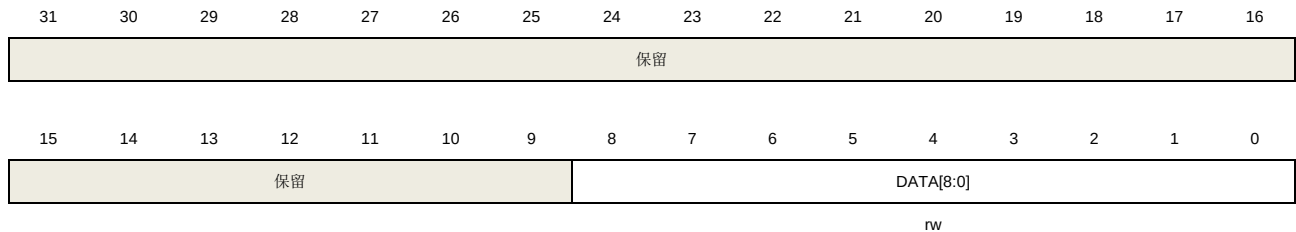
5	RBNE	读数据缓冲区非空 当读数据缓冲区接收到来自移位寄存器的数据时，该位置1。当寄存器UART_CTL0的RBNEIE位被置位，将会有中断产生。 软件可以通过对该位写0或读UART_DATA寄存器来将该位清0。 0：读数据缓冲区为空。 1：读数据缓冲区不为空。
4	IDLEF	空闲线检测标志 在一个帧时间内，在RX引脚检测到空闲状态，该位置1。当寄存器UART_CTL0的IDLEIE位被置位，将会有中断产生。 软件先读UART_STAT0，再读UART_DATA可清除该位。 0：未检测到空闲帧。 1：检测到空闲帧。 注意：当UART_CTL0寄存器中的MEN位被置1时，当RWU位为0时，此位总是1。而当RWU位为1时，此位总是0。
3	ORERR	溢出错误 在RBNE置位的情况下，如果UART_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UART_CTL2的ERRIE位被置位，将会有中断产生。 软件先读UART_STAT0，再读UART_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 当UART_CTL2中的OSB位被置为0时，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UART_CTL2的ERRIE位被置位，将会有中断产生。 软件先读UART_STAT0，再读UART_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UART_CTL2的ERRIE位被置位，将会有中断产生。软件先读UART_STAT0，再读UART_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。如果UART_CTL0中的PERRIE位被置位，则会产生中断。软件先读UART_STAT0，再读UART_DATA可清除该位。 0：没检测到校验错误。 1：检测到校验错误。

25.4.2. 数据寄存器 (UART_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7，8，9或10位，取决于UART_CTL0寄存器的WL0和WL1位）将被校验位取代。

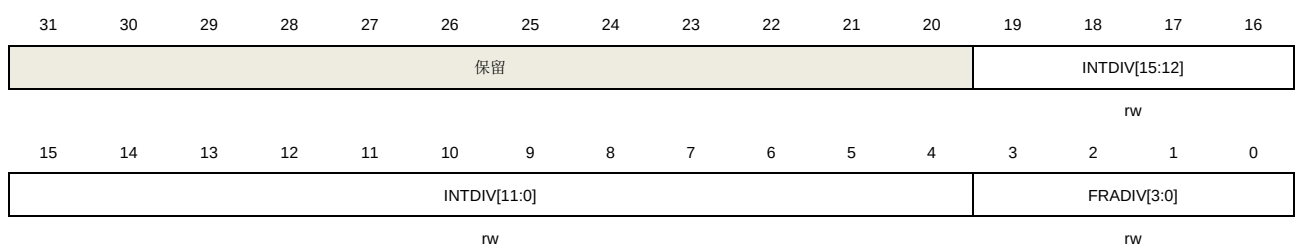
25.4.3. 波特率寄存器 (UART_BAUD)

地址偏移: 0x08

复位值: 0x0000 0000

使能UART (UEN=1) 时，不能写该寄存器。

该寄存器只能按字 (32位) 访问。



位/位域	名称	描述
31:20	保留	必须保持复位值。
19:4	INTDIV[15:0]	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。

25.4.4. 控制寄存器 0 (UART_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			WL1	保留											
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVSMOD	UEN	MEN	WL0	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	rw1	rw

位/位域	名称	描述
31:29	保留	必须保持复位值。
28	WL1	字长位1 该位必须与第12位 (WL0) 一起使用, 以确定字长。该位由软件设置或清除 00: 8数据位 01: 9数据位 10: 7数据位 11: 10数据位 注意: 当UART使能时 (UEN=1), 此位字段不能被写入。
27:16	保留	必须保持复位值。
15	OVSMOD	过采样模式 0: 16倍过采样 1: 8倍过采样 如果IREN = 1或LMEN = 1, 硬件将强制OVSMOD为0。 注意: 当UART使能时 (UEN=1), 此位字段不能被写入。
14	UEN	UART使能 0: 禁用UART 1: 使能UART
13	MEN	静音模式使能 0: 失能静音模式 1: 使能静音模式 注意: 1. 当UART使能时 (UEN=1), 此位字段不能被写入。 2. 当WM为0时 (通过检测总线空闲线唤醒静默模式), 在总线空闲的情况下, 不允许进入静默模式。
12	WL0	字长位0 请参阅此寄存器中的WL1位描述
11	WM	从静默模式唤醒方法

		0: 空闲线 1: 地址掩码
10	PCEN	校验控制使能 0: 校验控制禁用 1: 校验控制被使能
9	PM	校验模式 0: 偶校验 1: 奇校验
8	PERRIE	校验错误中断使能 如果该位置1, UART_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, UART_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, UART_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, UART_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, UART_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 该位用于指示UART是否处于静默模式。 0: 接收器处于正常工作模式。 1: 接收器处于静默模式。

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。空闲帧唤醒模式下（WM=0），UART在总线空闲状态下不允许通过设置RWU位进入静默模式

注意：当MEN位为1且UEN位为1时，该位可以被写为1。通过将UART_CTL0寄存器中MEN位或UEN位写0来清除。

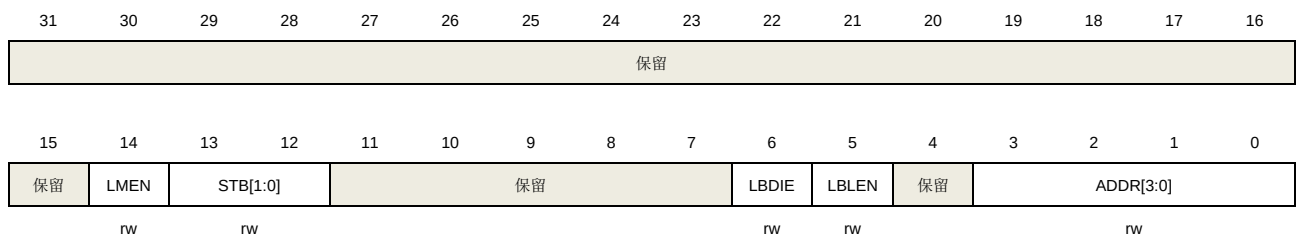
0	SBKCMD	发送断开帧 软件通过将该位置1发送断开帧。 断开帧传输结束由硬件清0。 0：没有发送断开帧。 1：发送断开帧。

25.4.5. 控制寄存器 1（UART_CTL1）

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
31:15	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 注意： 当UART使能时（UEN=1），此位字段不能被写入。
13:12	STB[1:0]	STOP位长 00：1停止位 01：保留 10：2停止位 11：保留 注意： 当UART使能时（UEN=1），此位字段不能被写入。
11:7	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UART_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 注意: 当UART使能时 (UEN=1), 此位字段不能被写入。
4	保留	必须保持复位值。
3:0	ADDR[3:0]	UART地址 地址掩码唤醒模式下 (WM=1), 如果接收到的数据帧低四位与ADDR[3:0]值不相等, UART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, UART就会被唤醒。 注意: 当UART使能时 (UEN=1), 此位字段不能被写入。

25.4.6. 控制寄存器 2 (UART_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	
MSBF	DINV	TINV	RINV	OSB	保留			DENT	DENR	保留			HDEN	IRLP	IREN	ERRIE
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	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。 注意: 当UART使能时 (UEN=1), 此位字段不能被写入。
14	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。 注意: 当UART使能时 (UEN=1), 此位字段不能被写入。
13	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。 注意: 当UART使能时 (UEN=1), 此位字段不能被写入。
12	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。 注意: 当UART使能时 (UEN=1), 此位字段不能被写入。。
11	OSB	单次采样法 该位用于选择采样方式, 当此位被设置时, UART对每个数据位只进行一次采样, 而不是每个位进行3次采样。当选择单次采样时, 噪声错误标志 (NERR) 被禁用。 0: 三次采样。 1: 单次采样。 注意: 当UART使能时 (UEN=1), 此位字段不能被写入。
10:8	保留	必须保持复位值。
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。
5:4	保留	必须保持复位值。
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。 注意: 当UART使能时 (UEN=1), 此位字段不能被写入。
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 注意: 当UART使能时 (UEN=1), 此位字段不能被写入。
1	IREN	IrDA模式使能 0: IrDA禁用 1: IrDA使能 注意: 当UART使能时 (UEN=1), 此位字段不能被写入。
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, UART_STAT0寄存器中FERR, ORERR, NERR被置位将产生中断。 0: 错误中断禁用

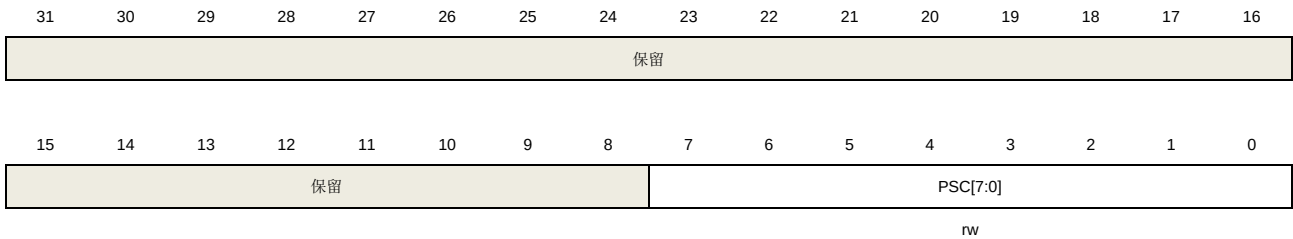
1: 错误中断使能。

25.4.7. 保护时间和预分频器寄存器 (UART_GP)

地址偏移: 0x18

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问。



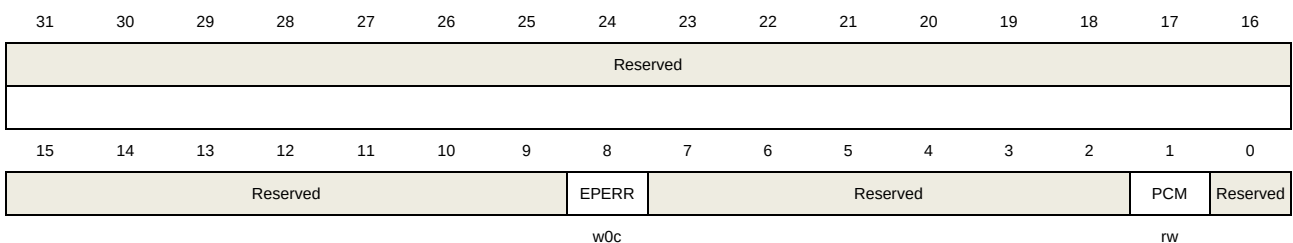
位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	PSC[7:0]	使能UART IrDA低功耗模式，这些位用来设定将外设时钟PCLK1分频产生低功耗频率的分频系数。 00000000: 保留 – 不要写入该值。 00000001: 对源时钟1分频。 ... 11111111: 对源时钟255分频。 在IrDA正常模式下，PSC只能设置成00000001。

25.4.8. 兼容性控制寄存器 (UART_CHC)

偏移地址: 0xC0

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	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。
7:2	保留	必须保持复位值。
1	PCM	奇偶校验一致性模式 0: 在通过地址匹配唤醒的情况下，数据的最高有效位（MSB）被考虑用于识别地址，但不包括奇偶校验位。并且接收器不检查地址数据的奇偶校验（在奇偶校验错误的情况下不设置PERR）。 1: 在通过地址匹配唤醒的情况下，接收器检查地址数据的奇偶校验，并且在奇偶校验错误的情况下设置PERR。
0	保留	必须保持复位值。

26. 内部集成电路总线接口（I2C）

26.1. 简介

I2C（内部集成电路总线）模块提供了符合工业标准的两线串行制接口，可用于 MCU 和外部 I2C 设备的通讯。I2C 总线使用两条串行线：串行数据线 SDA 和串行时钟线 SCL。

I2C 接口模块实现了 I2C 协议的标速模式，快速模式以及快速+ 模式，具备 CRC 计算和校验功能、支持 SMBus（系统管理总线）和 PMBus（电源管理总线）。此外，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 的负载。

26.2. 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，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）和快速+ 模式（最高 1MHz，该模式必须在 SYSCFG_CFG1 中使能 I2CFMPEN）；
- 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；
- 配置 I2C_CTL0 中 SMBEN 位，决定 I2C 是否工作在 SMBus 模式；
- 可选的 PEC（报文错误校验）生成和校验；
- 可编程数字过滤器；
- I2C 地址匹配时，由睡眠模式，深度睡眠模式唤醒；
- 独立于 PCLK 的时钟。

26.3. 功能说明

I2C 接口的内部结构如 [图 26-1. I2C 模块框图](#) 所示。

图 26-1. I2C 模块框图

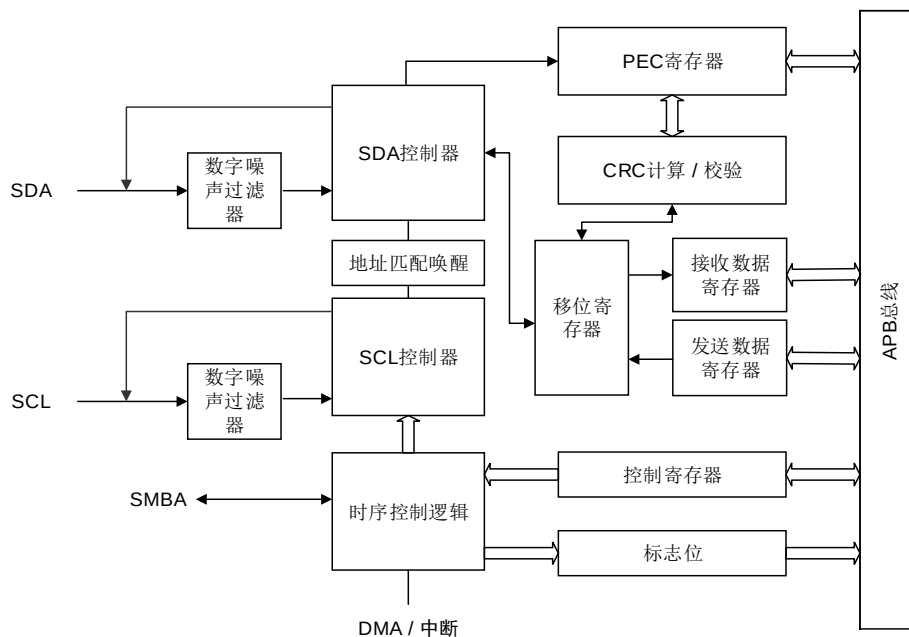


表 26-1. I2C 总线术语说明（参考飞利浦 I2C 规范）

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

26.3.1. 时钟要求

I2C 时钟独立于 PCLK 时钟，因此可以独立操作 I2C。

I2C 时钟（I2CCLK）可以从以下四个时钟源中选择：

- APB1 时钟 PCLK1（默认值）
- PLL：锁相环
- IRC32M：内部 32M RC 时钟
- HXTAL（1~24M）

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, 表示数字滤波器产生的延时。数字滤波器产生的延时为 $\text{DNF}[3:0] \times t_{\text{I2CCLK}}$ 。

PCLK 时钟周期 t_{PCLK} 必须满足以下条件:

- $t_{\text{PCLK}} < 4/3 \times t_{\text{SCL}}$

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。

26.3.2. I2C 通讯流程

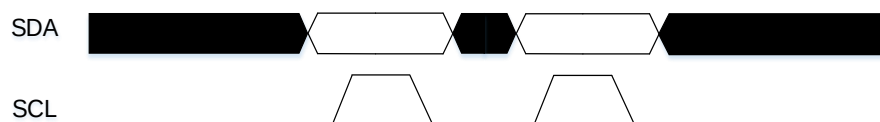
主机和从机都能实现数据收发, 因此, I2C 可以实现四种工作模式:

- 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 26-2. 数据有效性](#)。每个数据比特传输需要一个时钟脉冲。

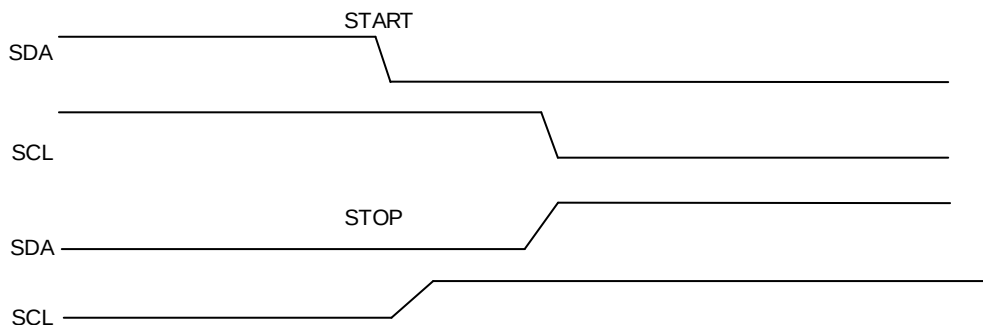
图 26-2. 数据有效性



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 26-3. 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 26-3. 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 26-4. 10 位地址的 I2C 通讯流程（主机发送）

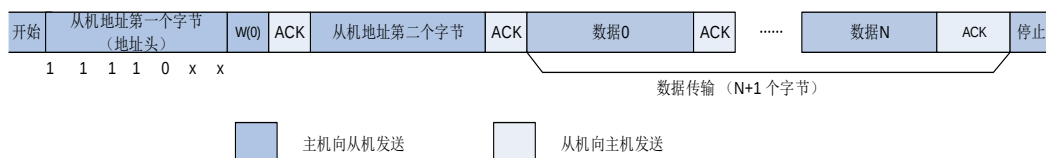


图 26-5. 7 位地址的 I2C 通讯流程（主机发送）

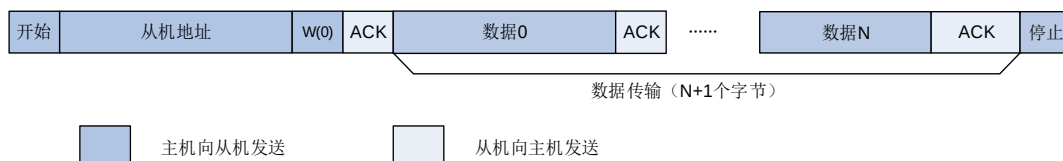
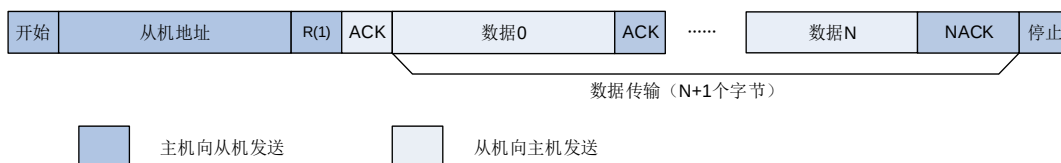


图 26-6. 7 位地址的 I2C 通讯流程（主机接收）



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 26-7. 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，如果主机接收是在主机发送结束后执行，读寻址序列可以是 RESTART+10 位地址头（读），如 [图 26-8. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R=1）](#) 所示。

图 26-7. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R=0）

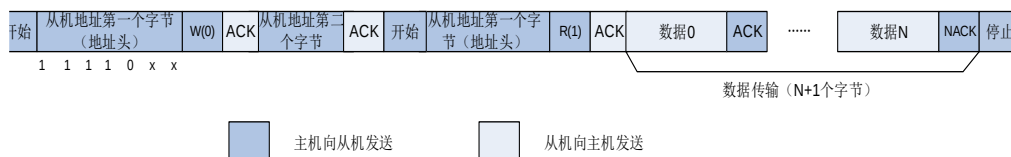
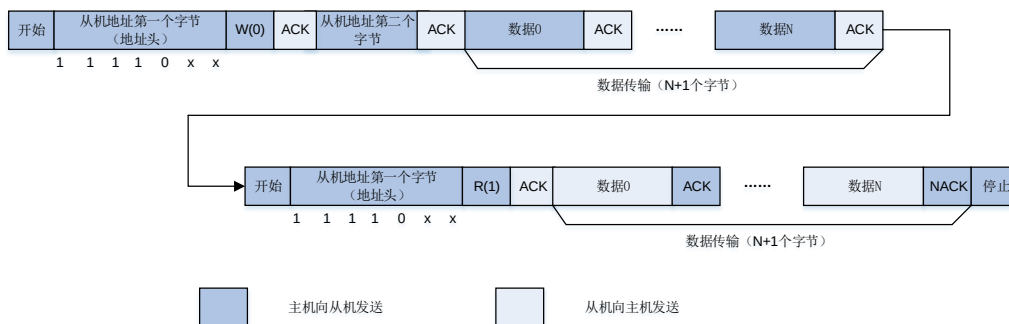


图 26-8. 10 位地址的 I2C 通讯流程（主机接收，HEAD10R=1）



26.3.3. 数字噪声滤波器

I2C 外设集成了数字噪声滤波器，噪声滤波器可根据实际需要在 I2C 外设启用前进行配置。

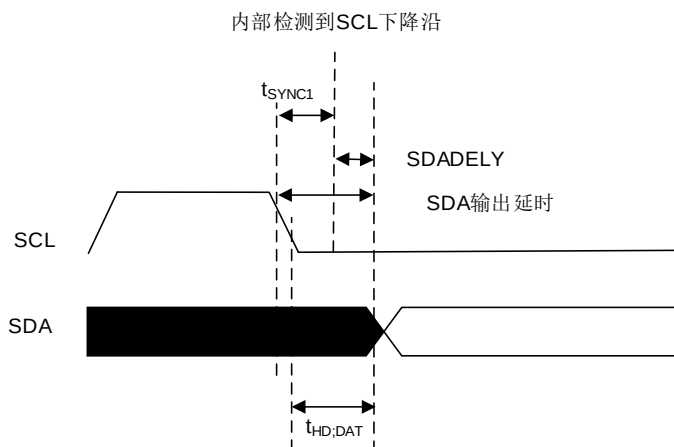
数字滤波器由 I2C_CTL0 寄存器中 DNF[3:0] 位来配置。当数字滤波器使能时，SCL 和 SDA 电平保持稳定的时间大于 $DNF[3:0] \times t_{I2CCLK}$ 才会发生内部变化。抑制峰值宽度可由 DNF[3:0] 配置。

26.3.4. 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。

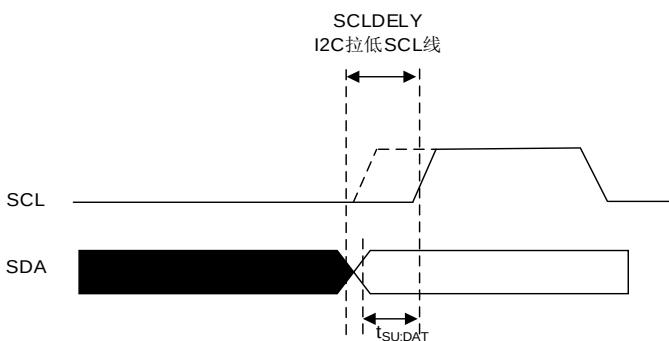
如果数据已经在 I2C_TDATA 寄存器中，在经历 SDADELY 延时后，数据由 SDA 发送，如 [图 26-9. 数据保持时间](#) 所示。

图 26-9. 数据保持时间



当数据经过 SDA 发送时，SCLDELY 计数器开启。如 [图 26-10. 数据建立时间](#) 所示。

图 26-10. 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 $t_{SDADELY} = SDADELY * t_{PSC} + t_{I2CCLK}$ ，其中 $t_{PSC} = (PSC + 1) * t_{I2CCLK}$ 。 $t_{SDADELY}$ 会影响 $t_{HD;DAT}$ 。 SDA 输出总延时为 $t_{SYNC1} + \{[SDADELY * (PSC + 1) + 1] * t_{I2CCLK}\}$ 。 t_{SYNC1} 由 SCL 下降斜率，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：

- $SDADELY \geq \{t_f(\max) + t_{HD;DAT}(\min) - [(DNF + 3) * t_{I2CCLK}]\} / [(PSC + 1) * t_{I2CCLK}]$
- $SDADELY \leq \{t_{HD;DAT}(\max) - [(DNF + 4) * t_{I2CCLK}]\} / [(PSC + 1) * t_{I2CCLK}]$

注意： $t_{HD;DAT}$ 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 $t_{SCLDELY} = (SCLDELY + 1) * t_{PSC}$ 。 $t_{SCLDELY}$ 影响 $t_{SU;DAT}$ 。

SCLDELY 必须符合以下条件：

$$\blacksquare \quad SCLDELY \geq \lceil [t_r(\max) + t_{SU,DAT}(\min)] / [(PSC+1) * t_{I2CCLK}] \rceil - 1$$

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 $t_{SCLL} = (SCLL+1) * t_{PSC}$, 其中 $t_{PSC} = (PSC+1) * t_{I2CCLK}$ 。 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 $t_{SCLH} = (SCLH+1) * t_{PSC}$, 其中 $t_{PSC} = (PSC+1) * t_{I2CCLK}$ 。 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 26-2. 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	

26.3.5. 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

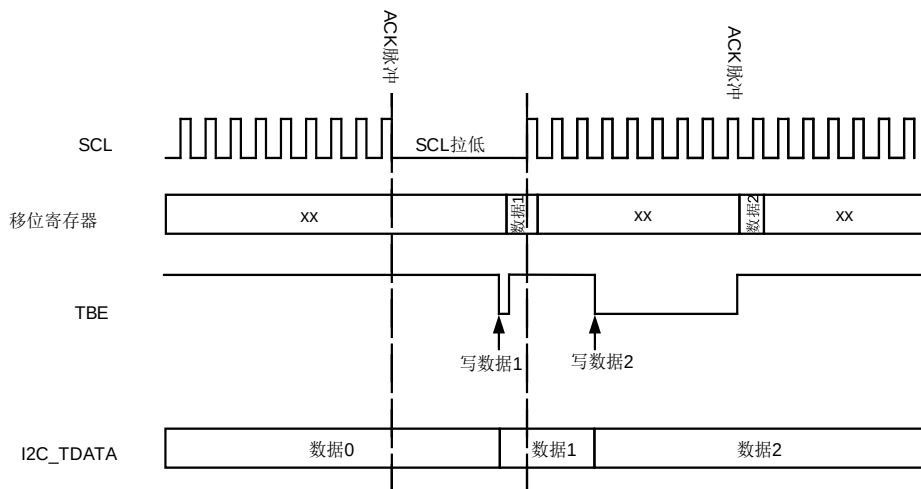
26.3.6. 数据传输

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。

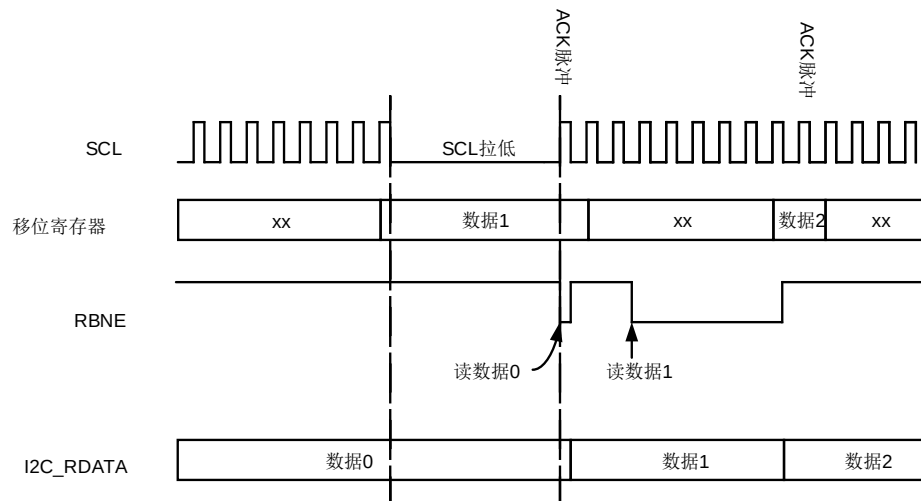
图 26-11. 数据发送



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 26-12. 数据接收



硬件传输管理重载和自动结束模式

为了管理字节传输和中断如[表 26-3. 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 26-3. 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。

26.3.7. 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 $[(SDADELY+SCLDELY+1)*(PSC+1)+1]*t_{I2CCLK}$ 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。

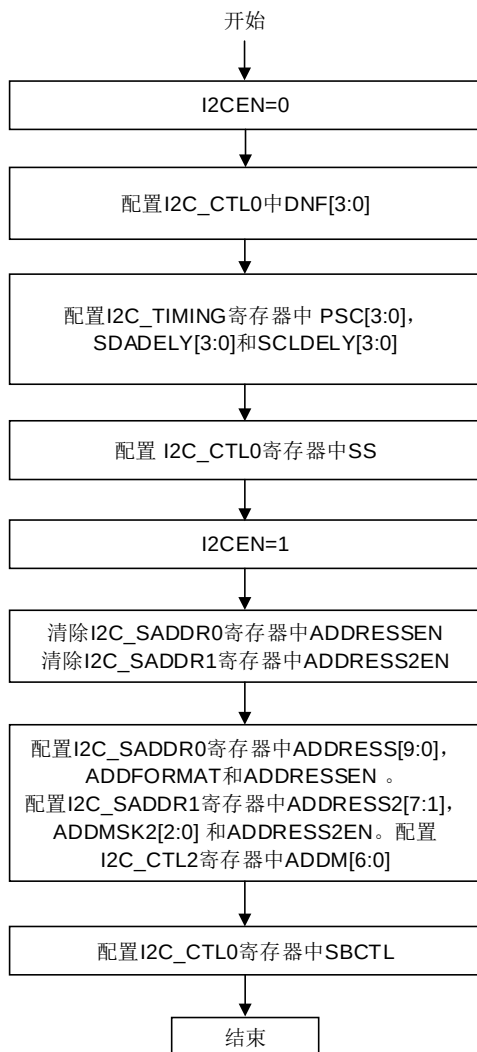
当 BYTENUM[7:0] 大于 0x1 时，在 BYTENUM[7:0] 数据接收期间，数据流是连续的。

注意：在下列情况下，可以配置 SBCTL 位：

1. I2CEN=0。
2. 从机还未被寻址。
3. ADDSEND=1。

当 ADDSEND=1，或者 TCR=1 时，RELOAD 才可以被修改。

图 26-13. 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 26-14. I2C 从机发送编程模型 (SS=0)

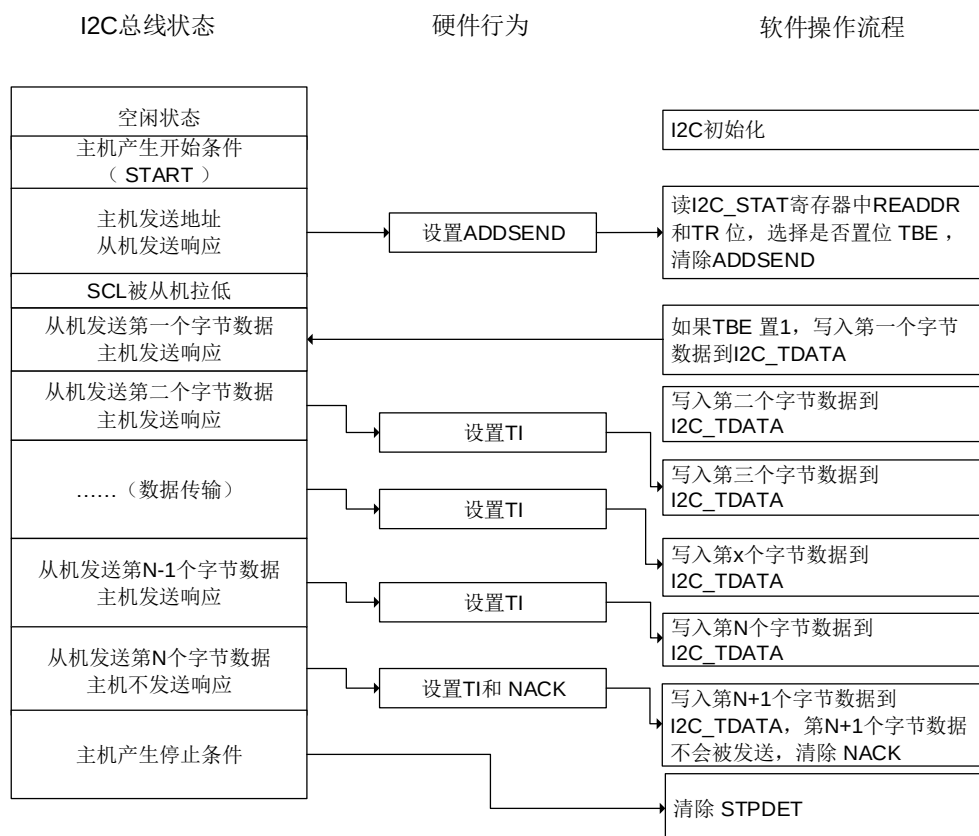
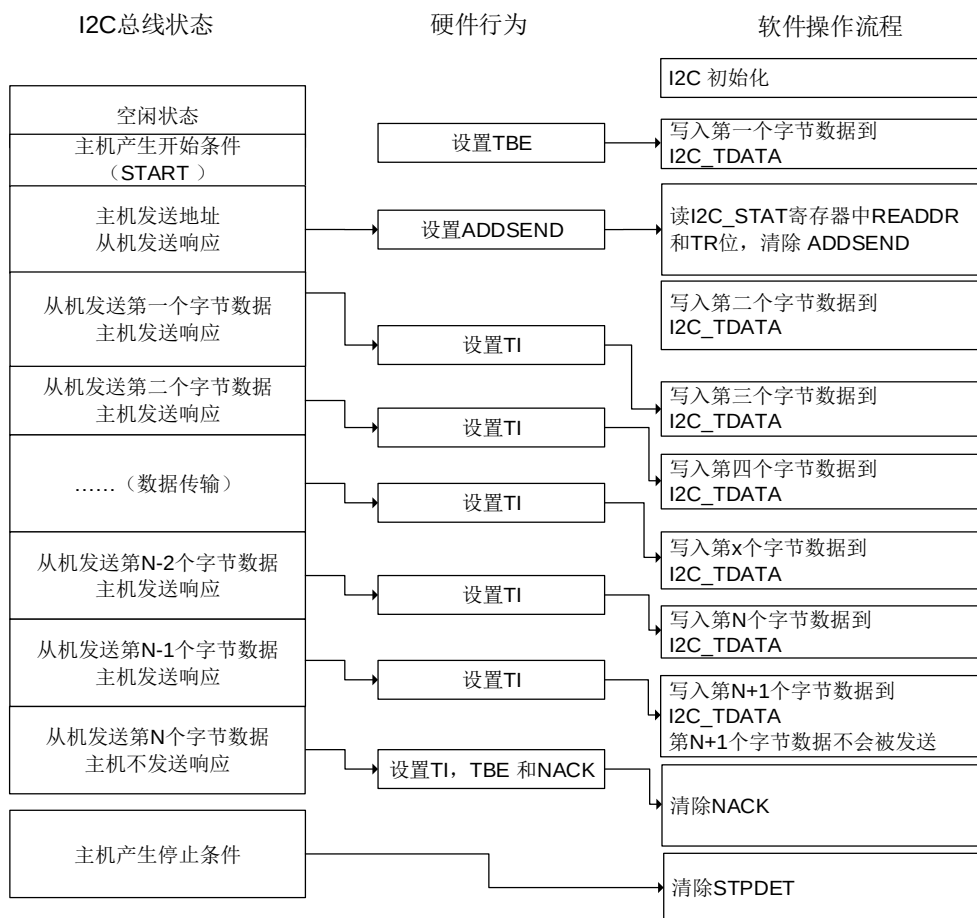


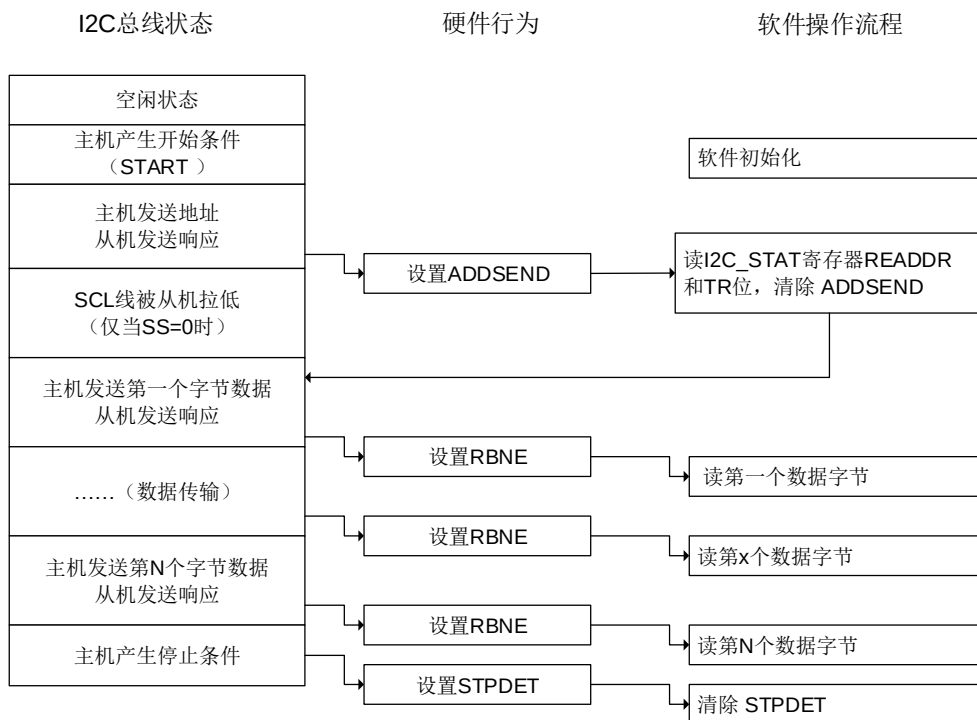
图 26-15. I2C 从机发送编程模型 (SS=1)



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 26-16. I2C 从机接收编程模型



26.3.8. 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

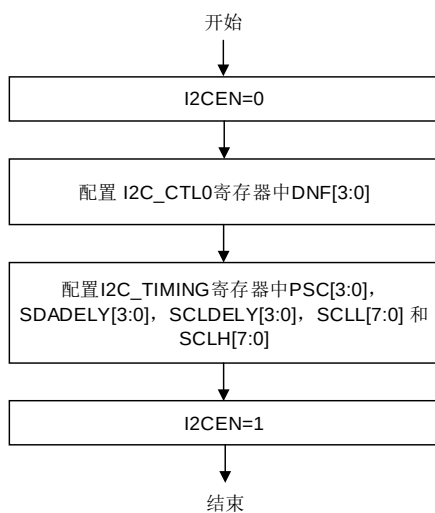
由硬件清零。当从机地址发送完成时，START 位由硬件清零。

注意：BYTENUM[7:0]不能配置为 0。

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 26-17. 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 26-18. I2C 主机发送编程模型 (N<=255)

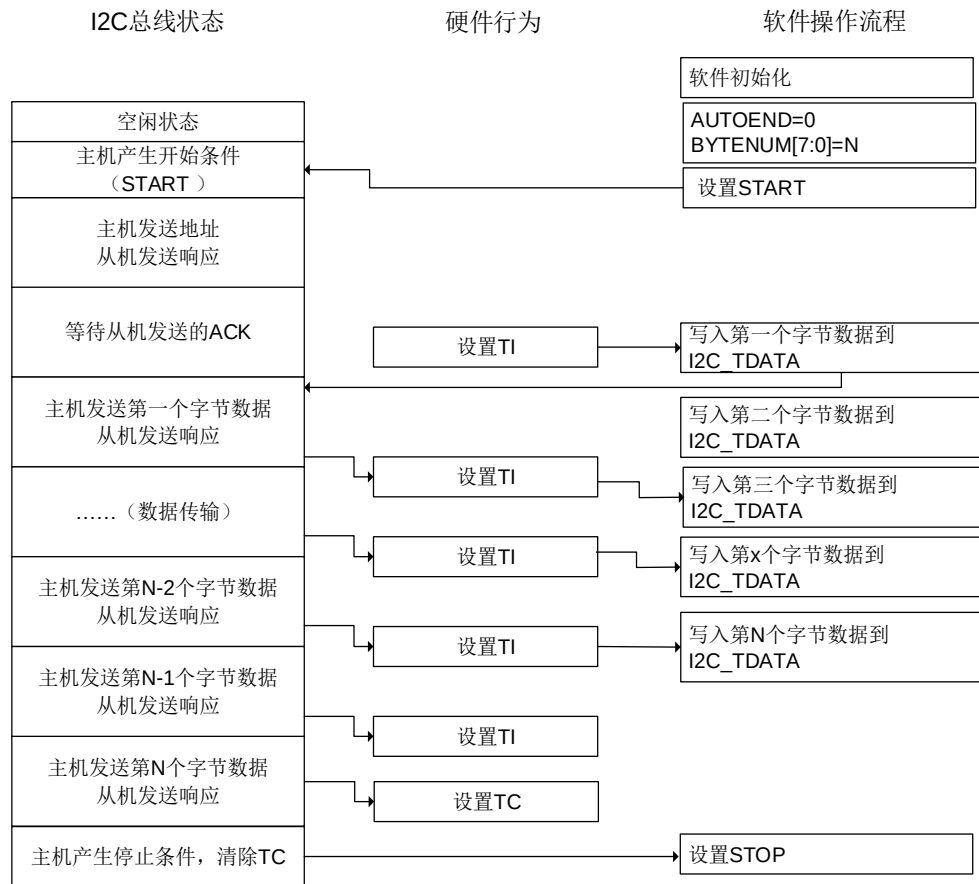


图 26-19. I2C 主机发送编程模型 (N>255)



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 26-20. I2C 主机接收编程模型 (N<=255)

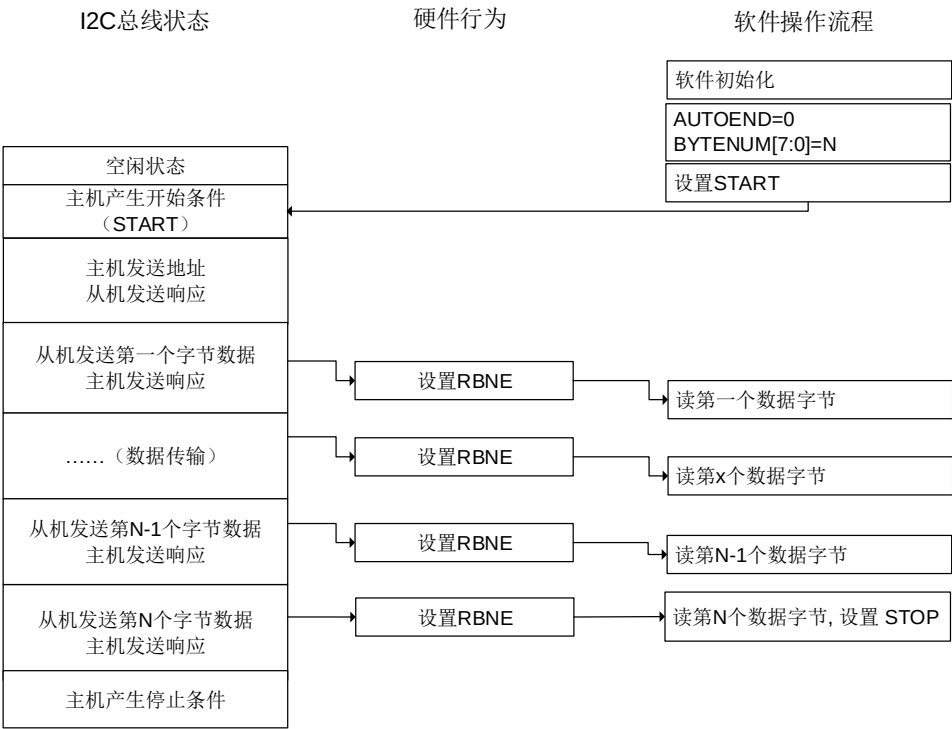


图 26-21. I2C 主机接收编程模型 (N>255)



26.3.9. 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，将可以继续 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 EXTOEN 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 $(\text{BUSTOA}+1)*2048*t_{\text{I2CCLK}}$ ，I2C_STAT 寄存器中 TIMEOUT 位将置 1。

BUSTOB[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 EXTOEN 位置 1 来使能定时器。在 EXTOEN 置 1 之后，BUSTOB[11:0]不能被修改。如果 SMBus 外设 SCL 拉低时间大于 $(\text{BUSTOB}+1)*2048*t_{\text{I2CCLK}}$ ，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 $t_{\text{HIGH,MAX}}$, 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 $(\text{BUSTOA}+1)*4*t_{\text{I2CCCLK}}$, 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)。

26.3.10. SMBus 模式

SMBus 主机发送器和从机接收器

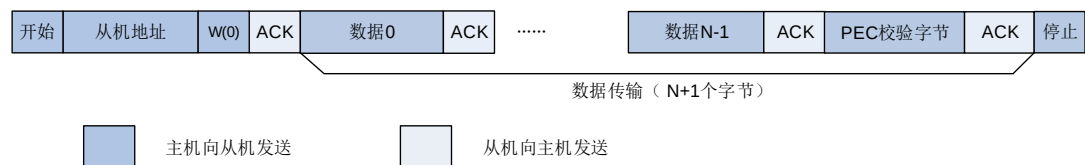
当 SMBus 主机发送 PEC 时, 必须在 START 位置 1 前, 将 PECTRANS 位置 1 并在

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 26-22. 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：

1. PECTRANS 位在 RELOAD 置 1 之后不能被修改。
2. SMBus 作为从机发送器，并且使用 PEC 功能时，必须在检测到 START 信号前配置 BYTENUM[7:0]。

图 26-23. SMBus 主机接收器和从机发送器通信流程



26.3.11. 从省电模式唤醒

当 I2C 地址匹配成功时，MCU 从睡眠模式，深度睡眠模式被唤醒。为了将 MCU 从这些省电模式唤醒，I2C_CTL0 寄存器中 WUEN 位必须置 1，同时 I2CCLK 时钟源选择 IRC32M。在深度睡眠模式下，IRC32M 关闭。当 I2C 检测到 START 信号时，IRC32M 打开，I2C 会将 SCL 拉低直到 IRC32M 被唤醒。在接收地址期间，IRC32M 为 I2C 提供时钟。当地址匹配时，在 MCU 唤醒期间，I2C 的 SCL 线被拉低。当 ADDSEND 清除时，SCL 线被释放，数据传输过程恢复正常。如果检测到的地址不匹配，IRC32M 会再次关闭，MCU 将不会被唤醒。

只有地址匹配中断(ADDMIE=1)能唤醒 MCU。如果 I2C 的时钟源是系统时钟，或者 WUEN=0，IRC32M 在接收到 START 信号之后将不会打开。当从省电模式唤醒使能时，数字滤波器必须禁能，I2C_CTL0 寄存器中 SS 位也必须清 0。如果禁止从省电模式唤醒（WUEN=0），则在进入省电模式之前必须禁能 I2C 外设（I2CEN=0）。

26.3.12. 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。

26.3.13. I2C 错误和中断

I2C 错误标志如[表 26-4. I2C 错误标志](#)所示。

表 26-4. 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 26-5. I2C 中断事件](#)所示。

表 26-5. 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
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	

26.3.14. I2C 调试模式

当为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 I2C_HOLD 配置位选择继续正常工作还是停止工作。

26.3.15. 事件链接功能（输出）

I2C 处理事件输出到事件互连单元（EVIC），包含以下来源的事件：

- 通讯错误/通讯事件
- 在接收期间 I2C_RDATA 非空
- 发送中断
- 传输完成

I2C 有四种类型中断事件：通信错误或通信事件（仲裁丢失、接收到 NACK、超时错误或从机模式下检测到 STOP 信号）、在接收期间 I2C_RDATA 非空、发送中断和传输完成。无论中断使能位是否配置，当满足中断触发条件时，通过 EVIC 将相应的事件信号发送给其他模块。

26.4. I2C 寄存器

I2C基地址：0x4000 5400

26.4.1. 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							SMBEN	PECEN	SMBALT EN	SMBDAE N	SMBHAE N	GCEN	WUEN	SS	SBCTL
							r/w	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
DENR	DENT	保留		DNF[3:0]				ERRIE	TCIE	STPDETI E	NACKIE	ADDIE	RBNEIE	TIE	I2CEN
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
31:25	保留	必须保持复位值。
24	SMBEN	SMBUS 模式使能。 在 I2C 模式下工作，该位必须设置为 0；工在 SMBus 模式下工作，该位必须设置为 1。 0: I2C 模式。 1: SMBus 模式，I2C 引脚电压被修改为 TTL 电平。 注意： 该位只能在 I2CEN=0 时被修改。
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	WUEN	使能从省电模式中唤醒，包含睡眠模式和深度睡眠模式。 当 MCU 从省电模式唤醒时该位清零。 0: 禁止从省电模式中唤醒。 1: 使能从省电模式中唤醒。 注意： 当 DNF[3:0]=0 时，WUEN 才能被置 1。
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
13:12	保留	必须保持复位值。
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。

26.4.2. 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, 该位不能被修改, 传输过程中软件不能修改该位。
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, 传输过程中软件不能修改该位。
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, 传输过程中软件不能修改该位。
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, 传输过程中软件不能修改该位。

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, 传输过程中软件不能修改该位。

26.4.3. 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，该位域不能被改写。 2. 当从机地址设置为保留地址 0b1111000 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。

26.4.4. 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，该位域不能被改写。 2. 当从机地址设置为保留地址 0b1111000 时，如果地址匹配，从机不响应。
0	保留	必须保持复位值。

26.4.5. 时序寄存器 (I2C_TIMING)

地址偏移: 0x10

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
PSC[3:0]				保留				SCLDELY[3:0]				SDADELY[3:0]			
rw								rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCLH[7:0]								SCLL[7:0]							
rw								rw							

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	SCLDELY[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$t_{SCLDELY}$。在主机模式下和在从机模式下 SS=0 时, 在$t_{SCLDELY}$期间 SCL 线被拉低。 $t_{SCLDELY} = (SCLDELY + 1) * t_{PSC}$。
19:16	SDADELY[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$t_{SDADELY}$。在主机模式下和在从机模式下 SS=0 时, 在$t_{SDADELY}$期间 SCL 线被拉低。 0x0: $t_{SDADELY} = 1 * t_{PSC}$ 0x1-0xF: $t_{SDADELY} = SDADELY * t_{PSC}$
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。

26.4.6. 超时寄存器 (I2C_TIMEOUT)

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
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， $t_{TIMEOUT}=(BUSTOA+1)*2048*t_{I2CCLK}$ 当 TOIDLE=1 时， $t_{IDLE}=(BUSTOA+1)*4*t_{I2CCLK}$ 注意： 该位域仅在 TOEN=0 时可以被改写。

26.4.7. 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。

26.4.8. 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。

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	SMBALT	TIMEOUT	PECERR	OUERRC	LOSTAR	BERRC	保留	STPDET	NACKC	ADDSEN	保留				
	C	C	C	W	BC	W		C	W	DC					
	W	W	W	W	W	W		W	W	W					

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。

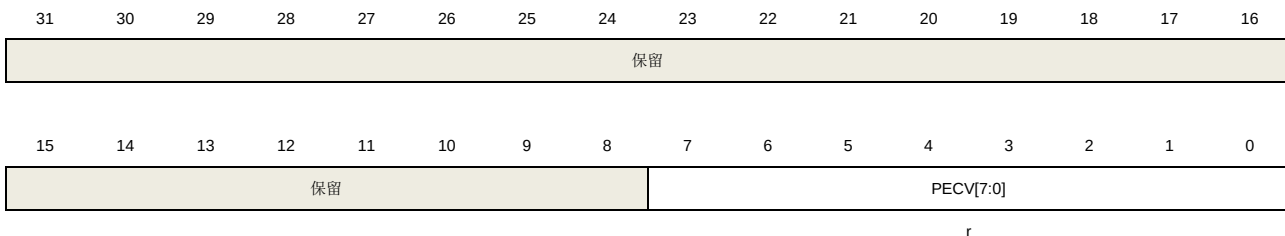
10	OUERRC	上溢/下溢标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 OUERR 位。
9	LOSTARBC	仲裁丢失标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 LOSTARB 位。
8	BERRC	总线错误标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 BERR 位。
7:6	保留	必须保持复位值。
5	STPDETC	停止位检测标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 STPDET 位。
4	NACKC	NACK 标志清零 软件对该位写 1 可以清除 I2C_STAT 寄存器中 NACK 位。
3	ADDSENDC	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。

26.4.9. 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。

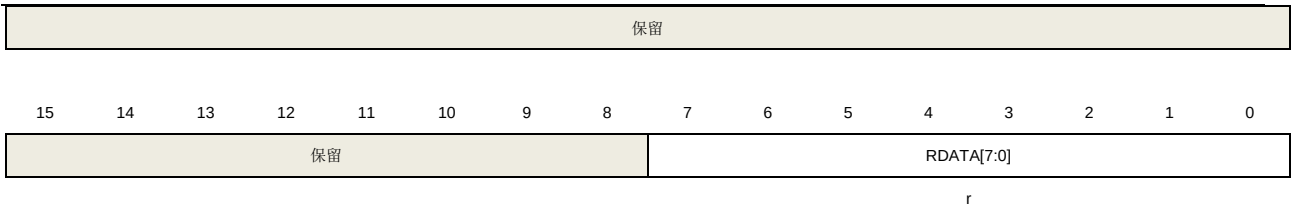
26.4.10. 接收数据寄存器 (I2C_RDATA)

地址偏移: 0x24

复位值: 0x0000 0000

该寄存器只能按字 (32 位) 访问。





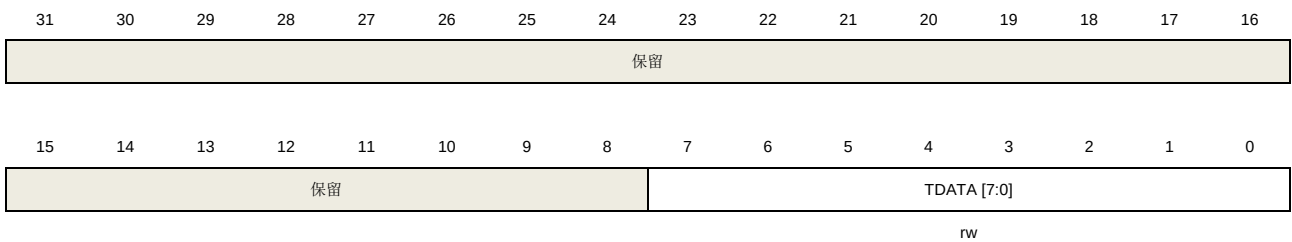
位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	RDATA[7:0]	接收到的数据

26.4.11. 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。



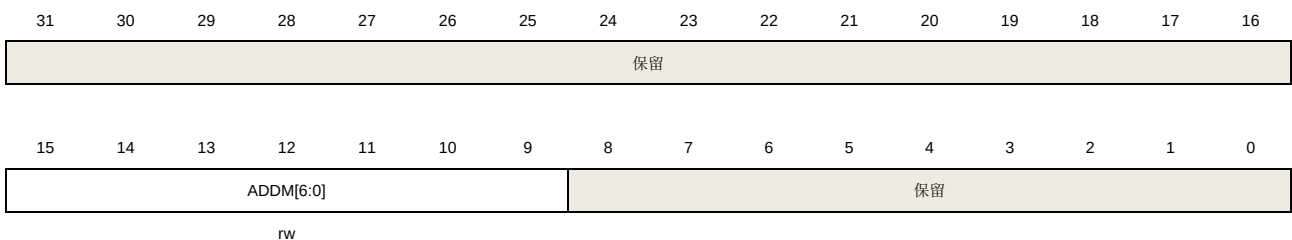
位/位域	名称	描述
31:8	保留	必须保持复位值。
7:0	TDATA[7:0]	发送的数据

26.4.12. 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。

27. 串行外设接口（SPI）

27.1. 简介

SPI模块可以通过SPI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，还支持SPI四线主机模式。

27.2. 主要特性

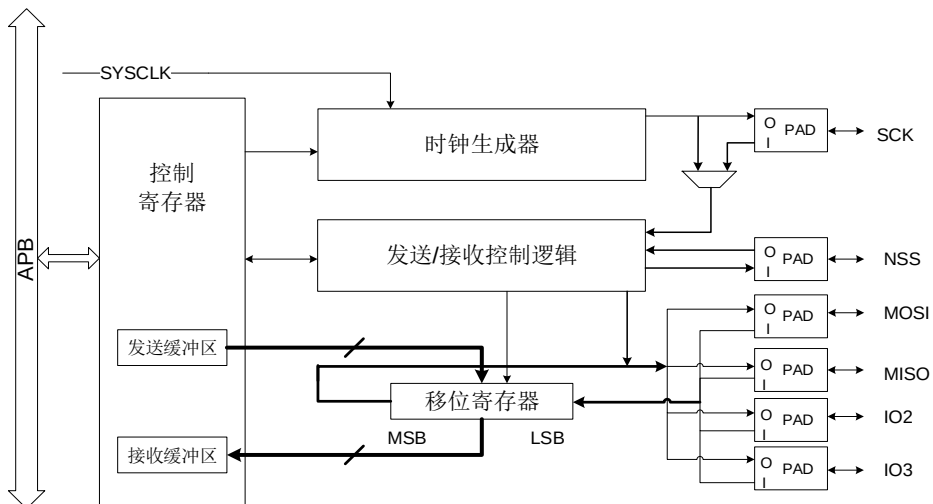
27.2.1. SPI 主要特性

- 具有全双工、半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。

27.3. SPI 功能说明

27.3.1. SPI 结构框图

图 26-24. SPI 结构框图



27.3.2. SPI 信号线描述

常规配置（非 SPI 四线模式）

表 26-6. 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。 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默认配置为单线模式，当SPI_QCTL中的QMOD位置1时，配置为SPI四线模式。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 26-7. SPI 四线信号描述

引脚名称	方向	描述
SCK	O	SPI 时钟输出
MOSI	I/O	发送或接收数据 0
MISO	I/O	发送或接收数据 1
IO2	I/O	发送或接收数据 2
IO3	I/O	发送或接收数据 3
NSS	O	NSS 输出

27.3.3. 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 26-25. 常规模式下的 SPI 时序图

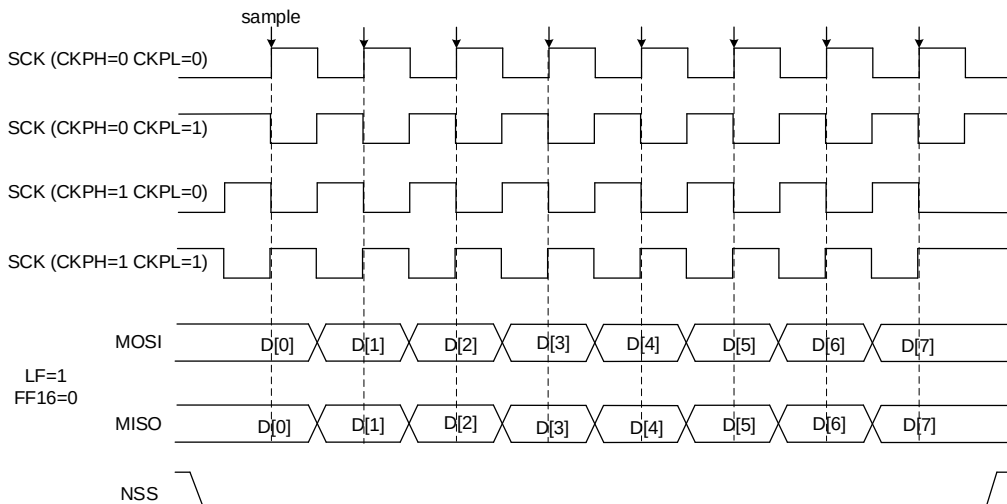
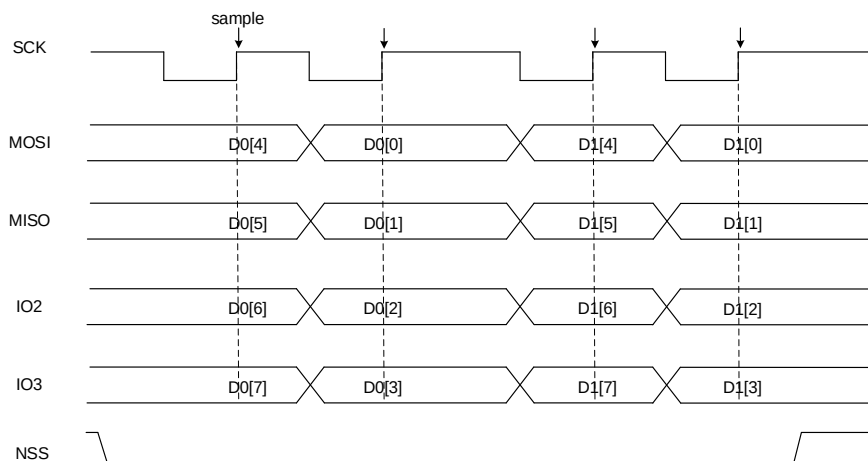


图 26-26. SPI 四线模式下的 SPI 时序图 (CKPL=1, CKPH=1, LF=0)



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。

27.3.4. NSS 功能

从机模式

当配置为从机模式 (MSTMOD=0) 时，在硬件NSS模式 (SWNSSSEN = 0) 下，SPI从NSS引脚获取NSS电平，在软件NSS (SWNSSSEN = 1) 下，SPI根据SWNSS位得到NSS电平。只有当NSS为低电平时，发送或接收数据。在软件NSS模式下，不使用NSS引脚。

表 26-8. 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 SWNSSSEN = 0	SPI 从机 NSS 电平从 NSS 引脚获取。
从机软件 NSS 模式	MSTMOD = 0 SWNSSSEN = 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 (SWNSSSEN=0, NSSDRV=0) 或者软件模式 (SWNSSSEN=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（SWNSSEN=0，NSSDRV=1）。使能SPI之后，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 26-9. 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 SWNSSEN = 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 SWNSSEN = 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 SWNSSEN = 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 SWNSSEN = 1 SWNSS = 1 NSSDRV: 不要求	从机可以使用硬件或软件 NSS 模式

27.3.5. SPI 运行模式

表 26-10. SPI 运行模式

模式	描述	寄存器配置	数据引脚用法
MFD	全双工主机模式	MSTMOD = 1 RO = 0 BDEN = 0 BDOEN: 不要求	MOSI: 发送 MISO: 接收
MTU	单向线连接主机发送模式	MSTMOD = 1 RO = 0 BDEN = 0 BDOEN: 不要求	MOSI: 发送 MISO: 不使用
MRU	单向线连接主机接收模式	MSTMOD = 1 RO = 1 BDEN = 0 BDOEN: 不要求	MOSI: 不使用 MISO: 接收
MTB	双向线连接主机发送模式	MSTMOD = 1 RO = 0 BDEN = 1 BDOEN = 1	MOSI: 发送 MISO: 不使用

模式	描述	寄存器配置	数据引脚用法
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 26-27. 典型的全双工模式连接

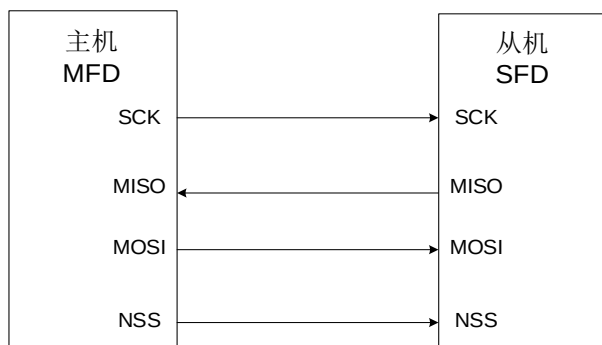


图 26-28. 典型的单工模式连接（主机：接收，从机：发送）

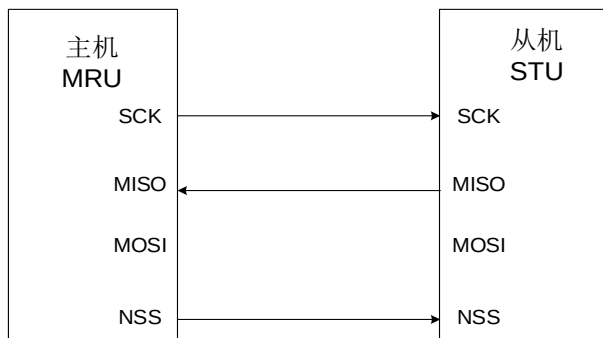


图 26-29. 典型的单工模式连接（主机：只发送，从机：接收）

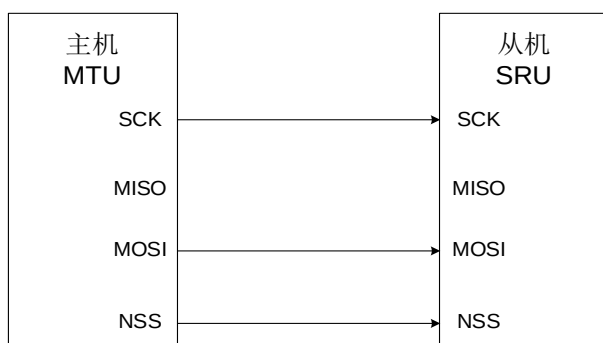
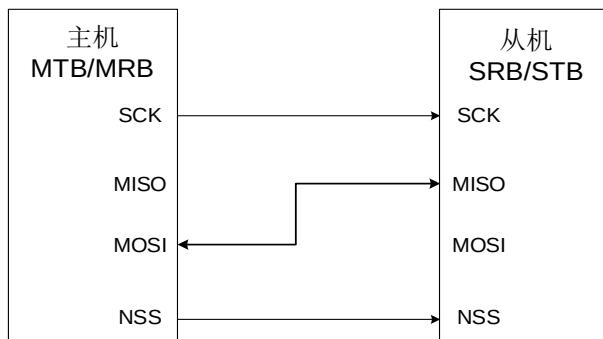


图 26-30. 典型的双向线连接



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[表26-10. SPI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）。

注意：在通信过程中，不应更改CKPH、CKPL、MSTMOD、PSC[2:0]、LF位。

SPI 基本发送和接收流程

发送流程

在完成初始化过程之后，SPI模块使能并保持在空闲状态。在主机模式下，当软件写一个数据到发送缓冲区时，发送过程开始。在从机模式下，当SCK引脚上的SCK信号开始翻转，且NSS引脚电平为低，发送过程开始。所以，在从机模式下，应用程序必须确保在数据发送开始前，数据已经写入发送缓冲区中。

当SPI开始发送一个数据帧时，首先将这个数据帧从数据缓冲区加载到移位寄存器中，然后开始发送加载的数据。在数据帧的第一位发送之后，TBE（发送缓冲区空）位置1。TBE标志位置1，说明发送缓冲区为空，此时如果需要发送更多数据，软件应该继续写SPI_DATA寄存器。

在主机模式下，若想要实现连续发送功能，那么在当前数据帧发送完成前，软件应该将下一个数据写入SPI_DATA寄存器中。

接收流程

在最后一个采样时钟边沿之后，接收到的数据将从移位寄存器存入到接收缓冲区，且RBNE（接收缓冲区非空）位置1。软件通过读SPI_DATA寄存器获得接收的数据，此操作会自动清除RBNE标志位。在MRU和MRB模式中，为了接收下一个数据帧，硬件需要连续发送时钟信号，而在全双工主机模式（MFD）中，仅当发送缓冲区非空时，硬件才接收下一个数据帧。

SPI 不同模式下的操作流程（非 SPI 四线模式，TI 模式或 NSSP 模式）

在全双工模式下，无论是MFD模式或者SFD模式，应用程序都应该监视RBNE标志位和TBE标志位，并且遵循上文描述的操作流程。

发送模式（MTU，MTB，STU或STB）与全双工模式中的发送流程类似，不同的是需要忽略RBNE位和RXORERR位。

相比于发送模式的情况，主机接收模式（MRU或MRB）与全双工的接收流程大不相同。在MRU模式或MRB模式下，在SPI使能后，SPI产生连续的SCK信号，直到SPI停止。所以，软件应该忽略TBE标志位，并且在RBNE位置1后，读出接收缓冲区内的数据，否则，将会产生接收过载错误。

除了忽略TBE标志位，且只执行上述的接收流程之外，从机接收模式（SRU或SRB）与全双工模式类似。

SPI TI 模式

SPI TI模式将NSS作为一种特殊的帧头标志信号，它的操作流程与上文描述的常规模式类似。上文描述的模式（MFD，MTU，MRU，MTB，MRB，SFD，STU，SRU，STB和SRB）都支

持TI模式。但是，在TI模式中，SPI_CTL0中的CKPL位和CKPH位是没有意义的，SCK信号的采样边沿为下降沿。

图 26-31. 主机 TI 模式在不连续发送时的时序图

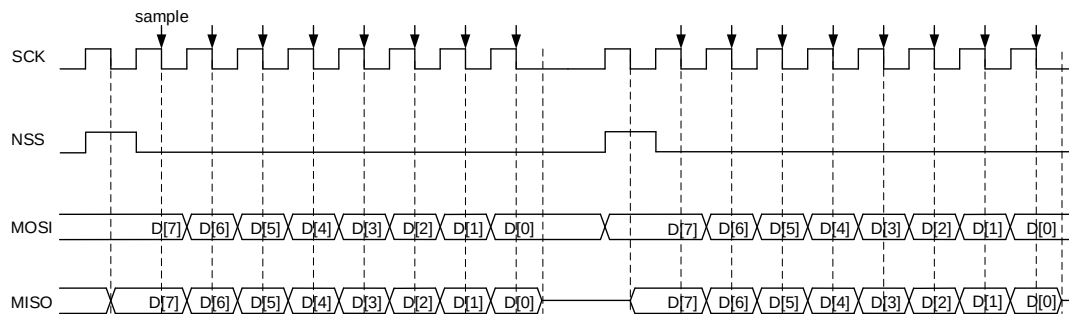
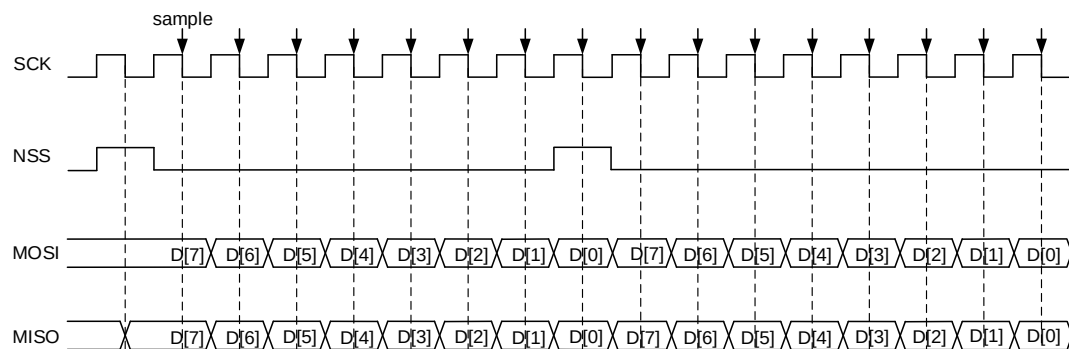
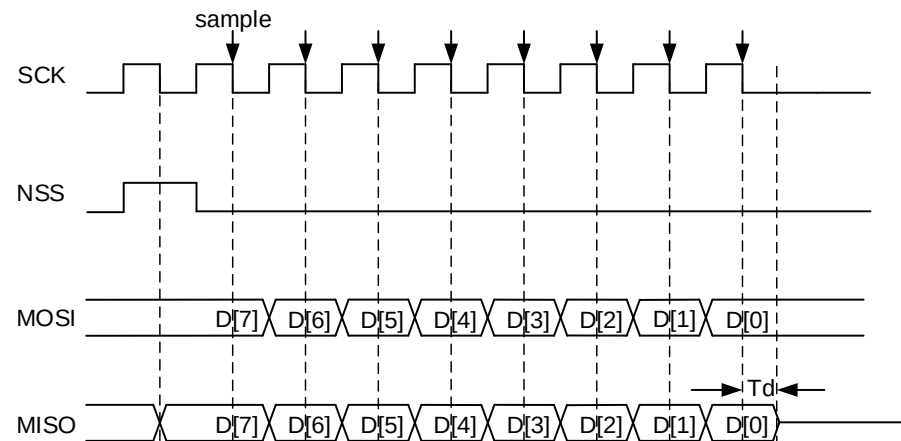


图 26-32. 主机 TI 模式在连续发送时的时序图



在主机TI模式下，SPI模块可实现连续传输或者不连续传输。如果主机写SPI_DATA的速度很快，那么就是连续传输，否则，为不连续传输。在不连续传输中，在每个字节传输前需要一个额外的时钟周期。在连续传输中，额外的时钟周期只存在于第一个字节之前，随后字节的起始时钟周期被前一个字节的最后一位的时钟周期覆盖。

图 26-33. 从机 TI 模式时序图



在从机TI模式中，在SCK信号的最后一个上升沿，从机开始发送最后一个字节的LSB位，在半位的时间之后，主机开始采集数据。为了确保主机采集到正确的数据，在释放该引脚之前，从机需要在SCK信号的下降沿之后继续驱动该位一段时间，这段时间称为 T_d ， T_d 通过SPI_CTL0寄存器中的PSC[2:0]位来设置。

$$T_d = \frac{T_{bit}}{2} + 5 * T_{pclk} \quad (19-1)$$

例如，如果PSC[2:0] = 010，那么 T_d 数值为 $9 * T_{pclk}$ 。

在从机模式下，从机需要监视NSS信号，如果检测到错误的NSS信号，将会置位FERR标志位。
例如，NSS信号在一个字节的中间位发生翻转。

NSS 脉冲模式操作流程

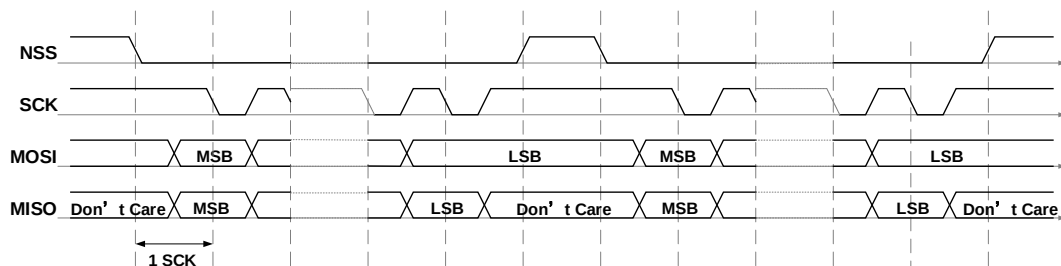
配置SPI_CTL1寄存器中的NSSP位使能该功能，为了确保使用该功能实现，需满足以下几个条件：配置设备为主机模式，使用普通SPI协议的数据帧格式，同时在第一个时钟跳变沿采样数据。

总之：MSTMOD = 1，NSSP = 1，CKPH = 0。

当使用NSS脉冲模式时，根据内部数据发送缓冲区的状态，NSS脉冲会在两个连续的数据帧之间产生，且持续时间至少为1个SCK时钟周期。如果数据发送缓冲区保持为空，可能会持续多个SCK时钟周期。NSS脉冲功能专为单一的主从应用设计，支持从机锁存数据。

下图描述了NSS脉冲模式在主机连续发送时的时序图。

图 26-34. NSS 脉冲模式时序图（主机连续发送）



SPI 四线模式操作流程

SPI四线模式用于控制四线SPI flash外设。

要配置成SPI四线模式，首先要确认TBE位置1，且TRANS位清零，然后将SPI_QCTL寄存器中的QMOD位置1。在SPI四线模式，SPI_CTL0寄存器中BDEN位、BDOEN位、CRCEN位、CRCNT位、FF16位、RO位和LF位保持清零，且MSTMOD位置1，以保证SPI工作于主机模式。SPIEN位、PSC位、CKPL位和CKPH位根据需要进行配置。

SPI四线模式有两种运行模式：四线写模式和四线读模式，通过SPI_QCTL寄存器中的QRD位进行配置。

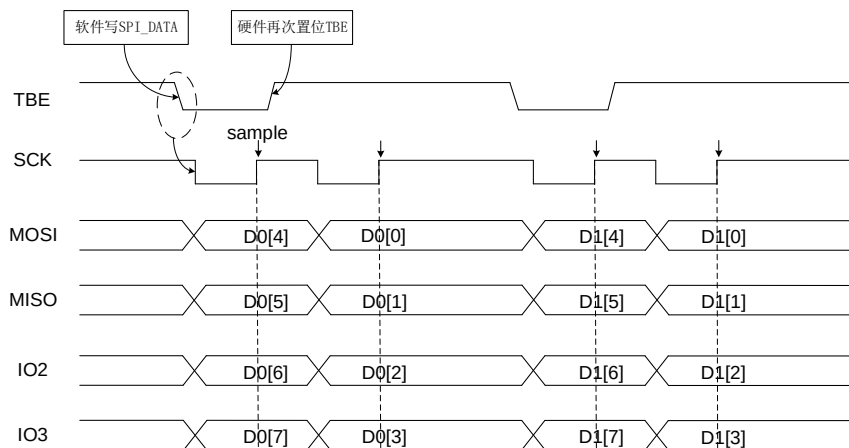
四线写模式

当SPI_QCTL寄存器中的QMOD位置1且QRD位清零时，SPI工作在四线写模式。在四线写模式中，MOSI、MISO、IO2和IO3都用作输出引脚，在SCK产生时钟信号后，一旦数据写入SPI_DATA寄存器（TBE位清零）且SPIEN位置1时，将会通过这四个引脚发送写入的数据。SPI开始数据传输之后，每发送一个数据帧都要检测TBE标志位，若不能满足条件则停止传输。

四线模式下发送操作流程：

1. 根据应用需求，配置SPI_CTL0和SPI_CTL1中的时钟预分频、时钟极性、相位等参数；
2. 将SPI_QCTL中的QMOD位置1，然后将SPI_CTL0中的SPIEN位置1来使能SPI功能；
3. 向SPI_DATA寄存器中写入一个字节的的数据，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 26-35. SPI 四线模式四线写操作时序图



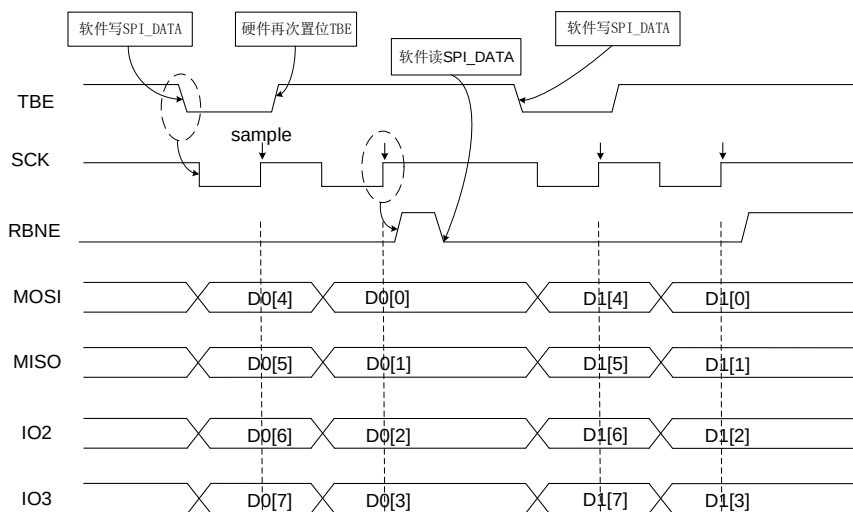
四线读模式

当SPI_QCTL寄存器中的QMOD位都置1时，SPI工作在四线读模式。在四线读模式中，MOSI、MISO、IO2和IO3都用作输入引脚，一旦数据写入SPI_DATA寄存器（TBE位清零）且SPIEN位置1时，在SCK信号线产生时钟信号。写数据到SPI_DATA寄存器只是为了产生SCK时钟信号，所以可以写入任何数据。SPI开始数据传输之后，每发送一个数据帧都要检测SPIEN位和TBE位，若条件不满足则停止传输。所以软件需要一直向SPI_DATA写空闲数据，以产生SCK时钟信号。

四线模式下接收操作流程：

1. 根据应用需求，配置SPI_CTL0和SPI_CTL1中时钟预分频、时钟极性、相位等参数；
2. 将SPI_QCTL中的QMOD位和QRD位置1，然后将SPI_CTL0中的SPIEN位置1来使能SPI功能；
3. 写任意数据（例如0xFF）到SPI_DATA寄存器；
4. 等待RBNE位置1，然后读SPI_DATA寄存器来获取接收的数据；
5. 写任意数据（例如0xFF）到SPI_DATA寄存器，以接收下一个字节数据。

图 26-36. SPI 四线模式四线读操作时序图



SPI 停止流程

不同运行模式下采用不同的流程来停止SPI功能。

MFD SFD

等待最后一个RBNE位并接收最后一个数据，等待TBE=1和TRANS=0，最后，通过清零SPIEN位关闭SPI。

MTU MTB STU STB

将最后一个数据写入SPI_DATA寄存器，等待TBE位置1，等待TRANS位清零，通过清零SPIEN位关闭SPI。

MRU MRB

等待倒数第二个RBNE位置1，从SPI_DATA寄存器读数据，等待一个SCK时钟周期，然后通过清零SPIEN位关闭SPI。等待最后一个RBNE位置1，并从SPI_DATA读数据。

SRU SRB

应用程序可以在任何时候关闭SPI功能，然后等待TRANS=0以确保当前通信过程结束。

TI模式

TI模式的停止流程与上面描述过程相同。

NSS脉冲模式

NSS脉冲模式的停止流程与上面描述过程相同。

SPI四线模式

在禁用SPI四线模式和关闭SPI功能之前，软件应该先检查：TBE位置1，TRANS位清零，SPI_QCTL中的QMOD位和SPI_CTL0中的SPIEN位清零。

27.3.6. 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，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。

27.3.7. 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，软件将CRCNT位置1。在CRC校验失败时，CRCERR错误标志位将会置1。

如果使能了DMA功能，软件不需要设置CRCNT位，硬件将会自动处理CRC传输和校验。

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）。

27.3.8. 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寄存器中接收到的数据的CRC计算值将会和紧随着最后一帧数据接收到的CRC值进行比较，当两者不同时，CRCERR位将会置1。

表 26-11. 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	

27.4. SPI 寄存器

SPI基地址：0x4001 3000

27.4.1. 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
BDEN	BDOEN	CRCEN	CRCNT	FF16	RO	SWNSS EN	SWNSS	LF	SPIEN	PSC[2:0]		MSTMOD	CKPL	CKPH	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw

位/位域	名称	描述
31:16	保留	必须保持复位值
15	BDEN	双向数据模式使能 0：2线单向传输模式 1：1线双向传输模式。数据在主机的主MOSI引脚和从机的MISO引脚之间传输。
14	BDOEN	双向传输输出使能 当BDEN置位时，该位决定了数据的传输方向。 0：工作在只接收模式 1：工作在只发送模式
13	CRCEN	CRC计算使能 0：CRC计算禁止 1：CRC计算使能
12	CRCNT	下一次传输CRC 0：下一次传输值为数据 1：下一次传输值为CRC值（TCRC） 当数据传输由DMA管理时，CRC值由硬件传输，该位应该被清零。 在全双工和只发送模式下，当最后一个数据写入SPI_DATA寄存器后应将该位置1。 在只接收模式下，在接收完倒数第二个数据后应将该位置1。
11	FF16	数据帧格式 0：8位数据帧格式 1：16位数据帧格式
10	RO	只接收模式

		当BDEN清零时，该位决定了数据的传输方向。 0：全双工模式 1：只接收模式
9	SWNSSEN	NSS软件模式使能 0：NSS硬件模式，NSS电平取决于NSS引脚 1：NSS软件模式，NSS电平取决于SWNSS位 该位在SPI TI模式下没有意义。
8	SWNSS	NSS软件模式下NSS引脚选择 0：NSS引脚拉低 1：NSS引脚拉高 只有在SWNSSEN置位时，该位有效。 该位在SPI TI模式下没有意义。
7	LF	最低有效位先发模式 0：先发送最高有效位 1：先发送最低有效位 该位在SPI TI模式下没有意义。
6	SPIEN	SPI使能 0：SPI设备禁止 1：SPI设备使能
5:3	PSC[2:0]	主时钟预分频选择 000：PCLK/2 001：PCLK/4 010：PCLK/8 011：PCLK/16 100：PCLK/32 101：PCLK/64 110：PCLK/128 111：PCLK/256 当使用SPI时，PCLK=PCLK2。
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

27.4.2. 控制寄存器 1 (SPI_CTL1)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								TBEIE	RBNEIE	ERRIE	TMOD	NSSP	NSSDRV	DMATEN	DMAREN
								r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

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。

27.4.3. 状态寄存器（SPI_STAT）

地址偏移：0x08

复位值：0x0000 0002

该寄存器可以按半字（16位）或字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								FERR	TRANS	RXORERR	CONFERR	CRCERR	保留	TBE	RBNE
								rc_w0	r	r	r	rc_w0	r	r	r

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 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空闲 1: SPI当前正在发送且/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。

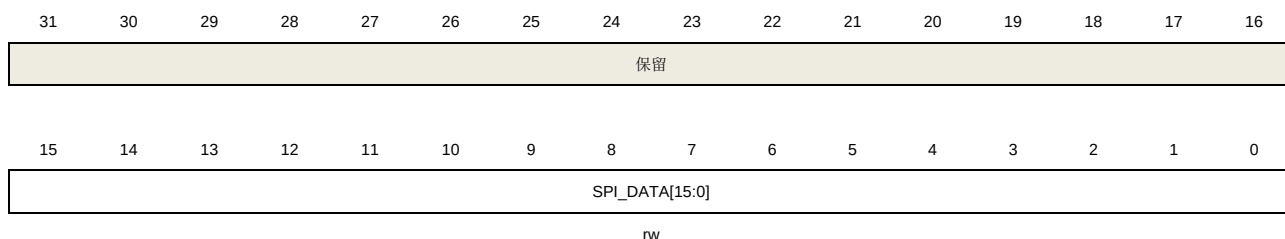
4	CRCERR	SPI CRC错误标志 0: SPI_RCRC值等于最后接收到的CRC值 1: SPI_RCRC值不等于最后接收到的CRC值 该位由硬件置位，可以通过写0清除。
3:2	保留	必须保持复位值
1	TBE	发送缓冲区空 0: 发送缓冲区非空 1: 发送缓冲区空
0	RBNE	接收缓冲区非空 0: 接收缓冲区空 1: 接收缓冲区非空

27.4.4. 数据寄存器（SPI_DATA）

地址偏移：0x0C

复位值：0x0000 0000

该寄存器可以按半字（16位）或字（32位）访问。



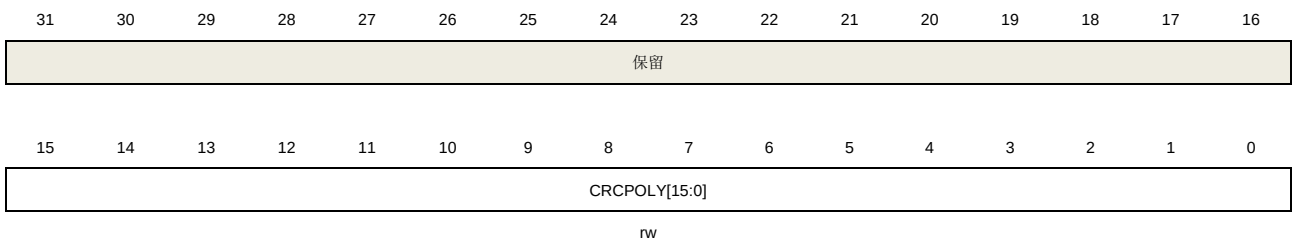
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	SPI_DATA[15:0]	数据传输寄存器值 硬件有两个缓冲区：发送缓冲区和接收缓冲区。向SPI_DATA写数据将会把数据存入发送缓冲区，从SPI_DATA读数据，将从接收缓冲区获得数据。 当数据帧格式为8位时，SPI_DATA[15:8]强制为0，SPI_DATA[7:0]用来发送和接收数据，发送和接收缓冲区都是8位。如果数据帧格式为16位，SPI_DATA[15:0]用于发送和接收数据，发送和接收缓冲区也是16位。

27.4.5. 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。



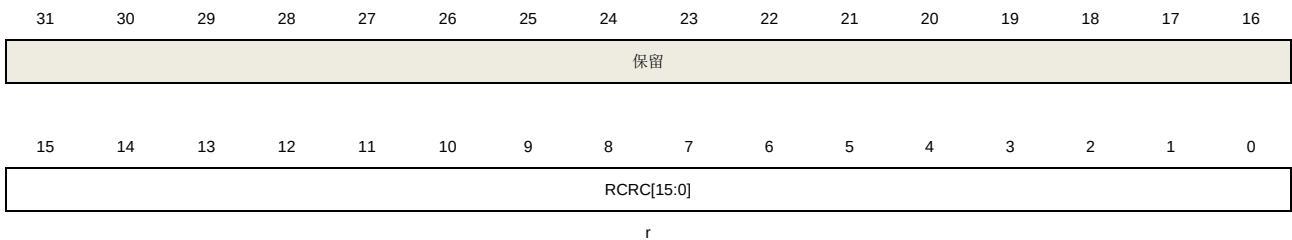
位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	CRCPOLY[15:0]	CRC多项式寄存器值 该值包含了CRC多项式，用于CRC计算，默认值为0007h。

27.4.6. 接收 CRC 寄存器（SPI_RCRC）

地址偏移：0x14

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	RCRC[15:0]	接收CRC寄存器值 当SPI_CTL0中的CRCEN置位时，硬件计算接收数据的CRC值，并保存到RCRC寄存器中。如果是8位数据帧格式，CRC计算基于CRC8标准进行，保存数据到RCRC[7:0]。如果是16位数据帧格式，CRC计算基于CRC16标准进行，保存数据到RCRC[15:0]。硬件在接收到每个数据位后都会计算CRC值，当TRANS置位时，读该寄存器将返回一个中间值。 当SPI_CTL0寄存器中的CRCEN位或RCU复位寄存器中的SPIxRST位置位时，该寄存器复位。

27.4.7. 发送 CRC 寄存器（SPI_TCRC）

地址偏移：0x18

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
TCRC[15:0]															
r															

位/位域	名称	描述
31:16	保留	必须保持复位值
15:0	TCRC[15:0]	发送CRC寄存器值 当SPI_CTL0中的CRCEN置位时，硬件计算发送数据的CRC值，并保存到TCRC寄存器中。如果是8位数据帧格式，CRC计算基于CRC8标准进行，保存数据到TCRC[7:0]。如果是16位数据帧格式，CRC计算基于CRC16标准进行，保存数据到TCRC[15:0]。硬件在发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位或RCU复位寄存器中的SPIxRST位置位时，该寄存器复位。

27.4.8. SPI 四线 SPI 控制寄存器（SPI_QCTL）

地址偏移：0x80

复位值：0x0000

该寄存器可以按半字（16位）或字（32位）访问。

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

位/位域	名称	描述
31:2	保留	必须保持复位值
1	QRD	四线SPI模式读选择 0: SPI四线模式写操作 1: SPI四线模式读操作 该位仅能在SPI未通信时配置（TRANS位清零）。
0	QMOD	四线SPI模式使能 0: SPI工作在单线模式 1: SPI工作在四线模式 该位仅能在SPI未通信时配置（TRANS位清零）。

28. 控制器局域网网络（CAN）

28.1. 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 和 2.0B。CAN 总线控制器可以处理总线上的数据收发，在 GD32M531 系列产品中，CAN 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。

28.2. 主要特征

- 支持 CAN 总线协议 2.0A 和 2.0B；
- 通信波特率最大为 1Mbit/s；
- 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；
- 在 GD32M531 系列产品中，具有 28 个过滤器；
- FIFO 锁定功能。

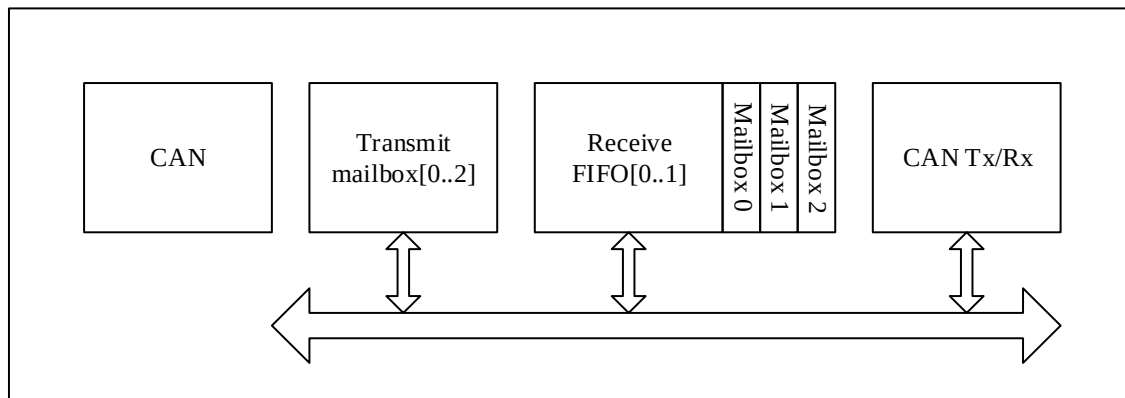
时间触发通信

- 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。

28.3. 功能说明

CAN 模块结构框图如 [图 28-1. CAN 模块结构框图](#) 所示。

图 28-1. CAN 模块结构框图



28.3.1. 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。

28.3.2. 通信模式

CAN 总线控制器有 4 种通信模式：

- 静默（Silent）通信模式；
- 回环（Loopback）通信模式；
- 回环静默（Loopback and Silent）通信模式；
- 正常（Normal）通信模式。

静默（Silent）通信模式

在静默通信模式下，可以从 CAN 总线接收数据，但不向总线发送任何数据。将 CAN_BT 寄存器中的 SCMOD 位置 1，使 CAN 总线控制器进入静默通信模式，将其清 0 可以退出静默通信模式。

静默通信模式可以用来监控 CAN 网络上的数据传输。

回环（Loopback）通信模式

在回环通信模式下，由 CAN 总线控制器发送的数据可以被自己接收并存入接收 FIFO，同时这些发送数据也送至 CAN 网络。将 CAN_BT 寄存器中的 LCMOD 位置 1，使 CAN 总线控制器进入回环通信模式，将其清 0 可以退出回环通信模式。

回环通信模式通常用来进行 CAN 通信自测。

回环静默（Loopback and Silent）通信模式

在回环静默通信模式下，CAN 的 RX 和 TX 引脚与 CAN 网络断开。CAN 总线控制器既不从 CAN 网络接收数据，也不向 CAN 网络发送数据，其发送的数据仅可以被自己接收。将 CAN_BT 寄存器中的 LCMOD 位和 SCMOD 位置 1，使 CAN 总线控制器进入回环静默通信模式，将它们清 0 可以退出回环静默通信模式。

回环静默通信模式通常用来进行 CAN 通信自测。对外 TX 引脚保持隐性状态（逻辑 1），RX 引脚保持高阻态。

正常（Normal）通信模式

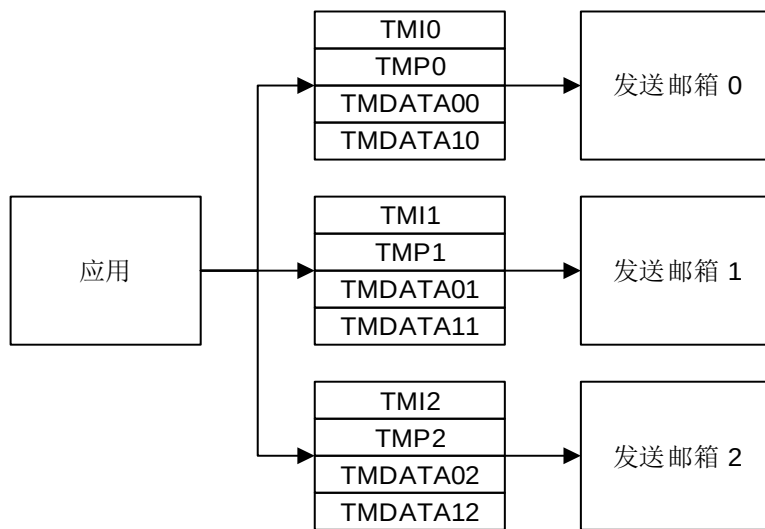
CAN 总线控制器通常工作在正常通信模式下，可以从 CAN 总线接收数据，也可以向 CAN 总线发送数据。这时需要将 CAN_BT 寄存器的 LCMOD 位和 SCMOD 位清 0。

28.3.3. 数据发送

发送寄存器

数据发送通过 3 个发送邮箱进行，可以通过寄存器 CAN_TMIx，CAN_TMPx，CAN_TMDATA0x 和 CAN_TMDATA1x 对发送邮箱进行配置。如[图 28-2. 发送寄存器](#)所示。

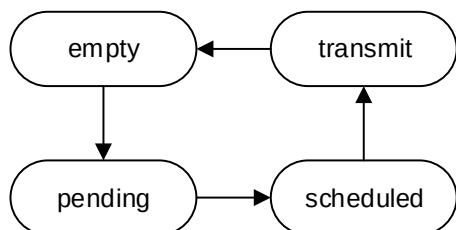
图 28-2. 发送寄存器



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 28-3. 发送邮箱状态转换](#)所示。

图 28-3. 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。

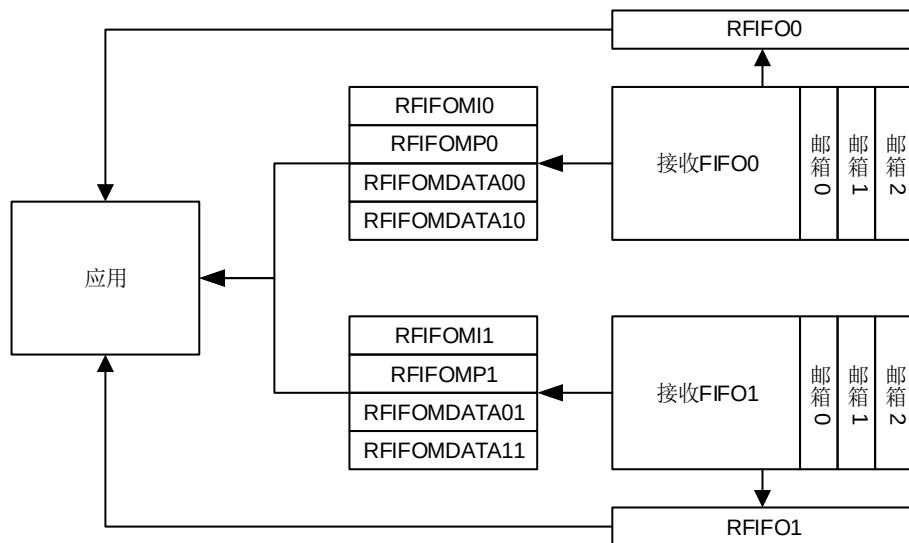
28.3.4. 数据接收

接收寄存器

应用程序通过2个深度为3的FIFO接收来自CAN网络的数据。

寄存器 CAN_RFIFOx 可以操作 FIFO, 也包含 FIFO 状态。寄存器 CAN_RFIFOMIx, CAN_RFIFOMPx, CAN_RFIFOMDATA0x和CAN_RFIFOMDATA1x用于接收数据帧。如 [图 28-4. 接收寄存器](#)所示。

图 28-4. 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置1释放邮箱。

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。

28.3.5. 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。

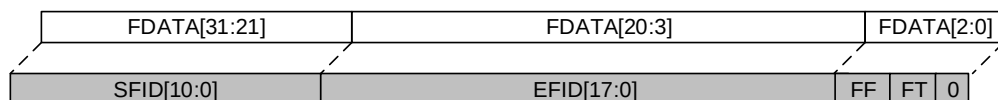
过滤器位宽

在GD32M531系列产品中，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。

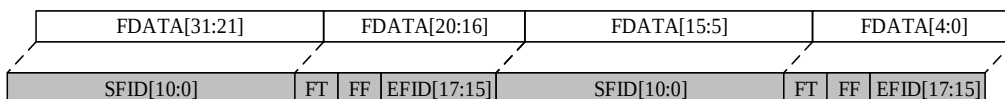
32-bit 位宽 CAN_FDATAx 包含字段：SFID[10:0]，EFID[17:0]，FF 和 FT。如[图 28-5. 32-bit 位宽过滤器](#)所示。

图 28-5. 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 28-6. 16-bit 位宽过滤器](#)所示。

图 28-6. 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 28-7. 32-bit 位宽掩码模式过滤器](#)所示。

图 28-7. 32-bit 位宽掩码模式过滤器

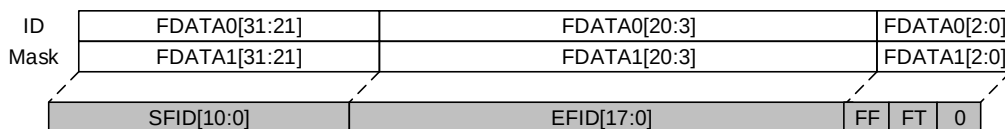
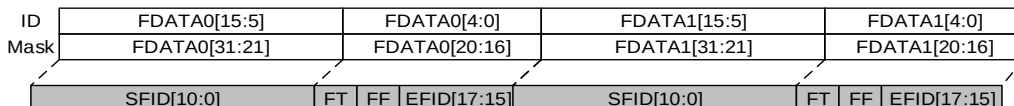


图 28-8. 16-bit 位宽掩码模式过滤器



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 28-9. 32-bit 位宽列表模式过滤器](#)所示。

图 28-9. 32-bit 位宽列表模式过滤器

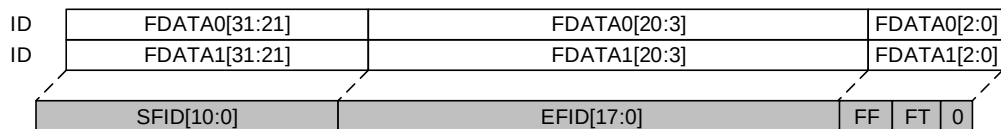
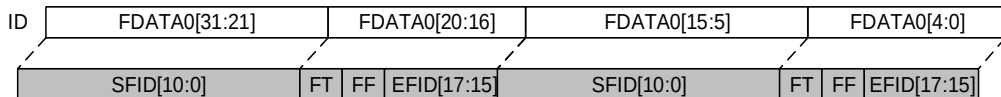


图 28-10. 16-bit 位宽列表模式过滤器



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 28-1. 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 28-1. 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。[表 28-2. 过滤索引](#)是一个过滤索引的例子。

在[表 28-2. 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 28-2. 过滤索引

过 滤 单元	FIFO0	激活	过 滤 序 号	过 滤 单元	FIFO1	激活	过 滤 序 号
0	F0DATA0-32bits-ID	是	0	2	F2DATA0[15:0]-16bits-ID	是	0
	F0DATA1-32bits-Mask				F2DATA0[31:16]-16bits-Mask		
1	F1DATA0-32bits-ID	是	1		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-Mask				F13DATA1-32bits-ID		16

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。

28.3.6. 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。

28.3.7. 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。

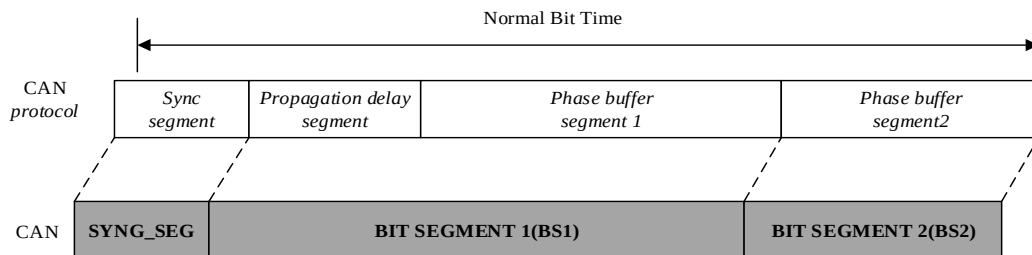
同步段（Synchronization segment），记为SYNC_SEG。该段占用1个时间单元（ $1 \times t_q$ ）。

位段1（Bit segment 1），记为BS1。该段占用1到16个时间单元。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。该段占用1到8个时间单元。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 28-11. 位时序](#)所示。

图 28-11. 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，占用1到4个时间单元。

有效跳变沿定义为只有当在前一个采样点检测到的总线状态为隐性（recessive）时，一个位时间内从隐性位到显性位的第一次转变。

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。

注：有关硬同步和再同步的详细说明，请参考ISO 11898标准。

波特率

波特率计算公式如下：

$$\text{BaudRate} = \frac{1}{\text{Normal Bit Time}} \quad (\text{式28-1})$$

$$\text{Normal Bit Time} = t_{\text{SYNC\_SEG}} + t_{\text{BS1}} + t_{\text{BS2}} \quad (\text{式28-2})$$

其中：

$$t_{\text{SYNC\_SEG}} = 1 \times t_q \quad (\text{式28-3})$$

$$t_{\text{BS1}} = (1 + \text{BT.BS1}) \times t_q \quad (\text{式28-4})$$

$$t_{\text{BS2}} = (1 + \text{BT.BS2}) \times t_q \quad (\text{式28-5})$$

$$t_q = (1 + \text{BT.BAUDPSC}) \times t_{\text{PCLK2}} \quad (\text{式28-6})$$

28.3.8. 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。

通过使能CAN_INTEN寄存器中的相应位（ERRIE等），软件可以在检测到错误时产生相应中断。

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。

28.3.9. 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28-3. CAN事件/中断标志](#)。

表 28-3. 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	

28.4. CAN 寄存器

CAN0基地址: 0x4001 A000

28.4.1. 控制寄存器 (CAN_CTL)

地址偏移: 0x00

复位值: 0x0001 0702

该寄存器只能按字 (32位) 访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															DFZ
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWRST	保留					FIIDE	BSPMOD	BSSSEL	TTC	ABOR	AWU	ARD	RFOD	TFO	SLPWMO D
rs						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	DFZ	调试冻结 如果 DBG_CTL0 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_CTL0 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
14:11	保留	必须保持复位值。
10	FIIDE	帧间隔到帧 ID 帧间隔的第三位接收到显性位后, 直接发送帧 ID 的第一个 bit, 略过发送 SOF。 0: 帧间隔到帧 ID 功能关闭 1: 帧间隔到帧 ID 功能开启
9	BSPMOD	位采样模式 0: 一个采样点 1: 三个采样点
8	BSSSEL	位采样同步选择 两个周期的同步会使信号的采集更加准确, 一个周期的同步可以减少信号采集的时间。

		0: 对CAN总线的采样进行一个周期的同步 1: 对 CAN 总线的采样进行两个周期的同步
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

28.4.2. 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。

保留															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				RXL	LASTRX	RS	TS	保留			SLPIF	WUIF	ERRIF	SLPWS	IWS
				r	r	r	r				rc_w1	rc_w1	rc_w1	r	r

位/位域	名称	描述
31:12	保留	必须保持复位值。
11	RXL	RX 引脚电平
10	LASTRX	RX 引脚最近一次的采样值
9	RS	接收状态 0: CAN 当前不是接收器 1: CAN 当前是接收器
8	TS	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

28.4.3. 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 当发生仲裁失败时, 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。 当发生仲裁失败时该位被置 1。
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。也可以在没有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_TMI2 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

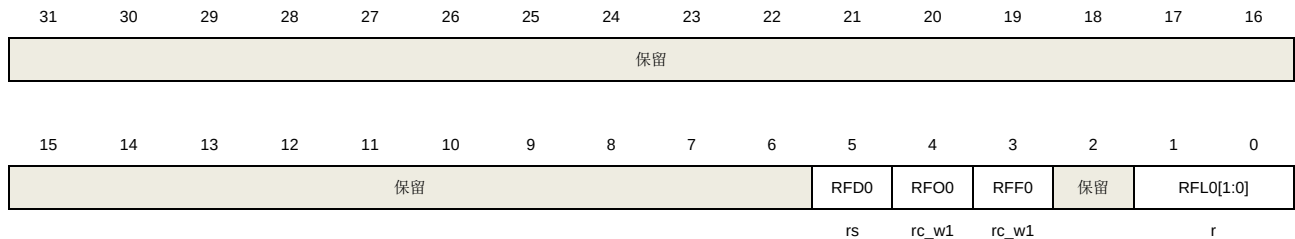
		MTF1 写 1 清 0。也可以在下一发送开始时由硬件清 0。 当发生错误时该位被置 1。
10	MAL1	邮箱 1 仲裁失败 当发生仲裁失败时，该位由硬件置 1。由软件写 1 清 0 或对 CAN_TSTAT 寄存器中 MTF1 写 1 清 0。也可以在下一发送开始时由硬件清 0。 当发生仲裁失败时该位被置 1。
9	MTFNERR1	邮箱 1 无错发送完成 当发送结束并且没有错误产生时，该位由硬件置 1。由软件写 1 清 0 或对 CAN_TSTAT 寄存器中 MTF1 写 1 清 0。也可以在无错传输结束时由硬件清 0。 0: 传输结束时发生了错误 1: 传输结束且没有错误
8	MTF1	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_TMI1 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 当发生仲裁失败时，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_TMI0 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

28.4.4. 接收 FIFO0 寄存器 (CAN_RFIFO0)

地址偏移: 0x0C

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问。



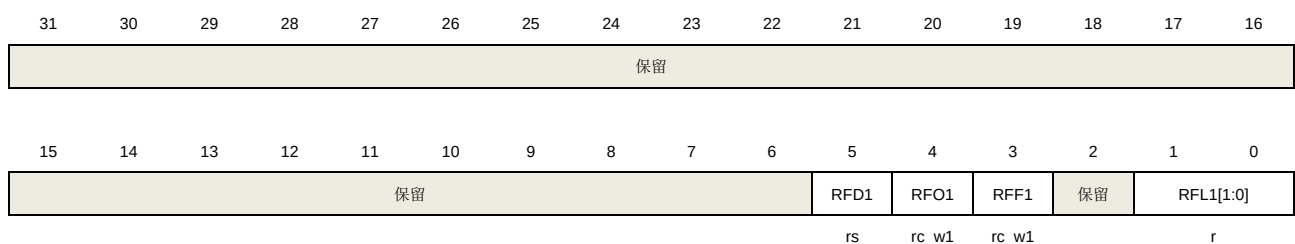
位/位域	名称	描述
31:6	保留	必须保持复位值。
5	RFD0	释放一次 FIFO0 中的数据 该位被置 1，将释放 FIFO0 中的一帧数据。 FIFO 释放相应的数据空间后，该位被清 0。
4	RFO0	接收 FIFO0 溢出 当接收 FIFO0 溢出时被置位，由软件写 1 清 0。 0: 接收 FIFO0 没有溢出 1: 接收 FIFO0 溢出
3	RFF0	接收 FIFO0 满 当接收 FIFO0 满时被置位，由软件写 1 清 0。 0: 接收 FIFO0 不满 1: 接收 FIFO0 满
2	保留	必须保持复位值。
1:0	RFL0[1:0]	接收 FIFO0 中帧的数量

28.4.5. 接收 FIFO1 寄存器 (CAN_RFIFO1)

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问。



位/位域	名称	描述
31:6	保留	必须保持复位值。
5	RFD1	释放一次 FIFO1 中的数据 该位被置 1，将释放 FIFO1 中的一帧数据。 FIFO 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

28.4.6. 中断使能寄存器（CAN_INTEN）

地址偏移：0x14

复位值：0x0000 0000

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留														SLPWIE	WIE
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIE	保留			ERRNIE	BOIE	PERRIE	WERRIE	保留	RFOIE1	RFFIE1	RFNEIE1	RFOIE0	RFFIE0	RFNEIE0	TMEIE
rw				rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

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

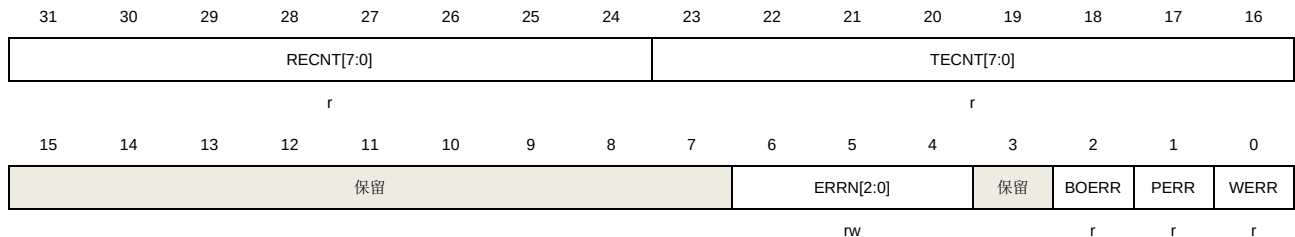
		1: 使能错误中断
14:12	保留	必须保持复位值。
11	ERRNIE	错误种类中断使能 0: 禁用错误种类中断 1: 使能错误种类中断
10	BOIE	离线中断使能 0: 禁用离线中断 1: 使能离线中断
9	PERRIE	被动错误中断使能 0: 禁用被动错误 1: 使能被动错误
8	WERRIE	警告错误中断使能 0: 禁用警告错误中断 1: 使能警告错误中断
7	保留	必须保持复位值。
6	RFOIE1	接收 FIFO1 溢出中断使能 0: 禁用接收 FIFO1 溢出中断 1: 使能接收 FIFO1 溢出中断
5	RFFIE1	接收 FIFO1 满中断使能 0: 禁用接收 FIFO1 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

28.4.7. 错误寄存器（CAN_ERR）

地址偏移：0x18

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:24	RECNT[7:0]	接收错误计数值
23:16	TECNT[7:0]	发送错误计数值
15:7	保留	必须保持复位值。
6:4	ERRN[2:0]	错误种类 ERRN 由硬件更新，可以反映位传输过程中的错误情况。当位传输成功没有错误时，ERRN 为 0。软件可以设置 ERRN 为 0b111。 000：无错误 001：填充错误 010：格式错误 011：ACK 错误 100：位隐性错 101：位显性错误 110：CRC 错误 111：软件设置值
3	保留	必须保持复位值。
2	BOERR	离线错误 当 TEC 上溢（超过 255）时，CAN 总线控制器进入离线状态，该位被置 1。
1	PERR	被动错误 当 TECNT 或者 RECNT 大于 127 时，该位由硬件置 1。
0	WERR	警告错误 当 TECNT 或 RECNT 大于等于 96 时，该位由硬件置 1。

28.4.8. 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[1:0]		保留	BS2[2:0]			BS1[3:0]			
rw	rw	rw				rw			rw			rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留						BAUDPSC[9:0]									
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
29:26	保留	必须保持复位值。
25:24	SJW[1:0]	再同步补偿宽度 再同步补偿占用的时间单元数量=SJW[1:0]+1
23	保留	必须保持复位值。
22:20	BS2[2:0]	位段 2 位段 2 占用的时间单元数量=BS2[2:0]+1
19:16	BS1[3:0]	位段 1 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	BAUDPSC[9:0]	波特率分频系数

28.4.9. 发送邮箱标识符寄存器（CAN_TMIx）（x = 0...2）

地址偏移：0x180，0x190，0x1A0

复位值：0xFFFF XXXX（bit0 = 0）

该寄存器只能按字（32位）访问。

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

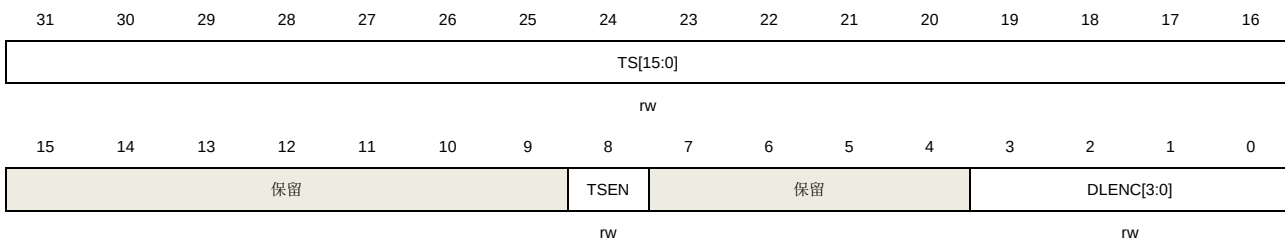
位/位域	名称	描述
31:21	SFID[10:0]/EFID[28:18]	标识符 SFID[10:0]: 标准格式帧标识符 EFID[28:18]: 扩展格式帧标识符
20:16	EFID[17:13]	标识符 EFID[17:13]: 扩展格式帧标识符
15:3	EFID[12:0]	标识符 EFID[12:0]: 扩展格式帧标识符
2	FF	帧格式 0: 标准格式帧 1: 扩展格式帧
1	FT	帧种类 0: 数据帧 1: 遥控帧
0	TEN	发送使能 当应用程序想要发送数据时, 该位被置 1 将启动发送过程。当发送结束, 发送邮箱为空时, 该位由硬件清 0。 0: 禁用发送 1: 使能发送

28.4.10. 发送邮箱属性寄存器 (CAN_TMPx) (x = 0..2)

地址偏移: 0x184, 0x194, 0x1A4

复位值: 0xFFFF XXXX

该寄存器只能按字 (32位) 访问。



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。

7:4	保留	必须保持复位值。
3:0	DLENC[3:0]	数据长度，DLENC[3:0]表示帧内数据长度。

28.4.11. 发送邮箱 data0 寄存器 (CAN_TMDATA0x) (x = 0..2)

地址偏移：0x188, 0x198, 0x1A8

复位值：0xFFFF XXXX

该寄存器只能按字（32位）访问。



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

28.4.12. 发送邮箱 data1 寄存器 (CAN_TMDATA1x) (x = 0..2)

地址偏移：0x18C, 0x19C, 0x1AC

复位值：0xFFFF XXXX

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31:24	DB7[7:0]	字节 7
23:16	DB6[7:0]	字节 6

15:8	DB5[7:0]	字节 5
7:0	DB4[7:0]	字节 4

28.4.13. 接收 FIFO 邮箱标识符寄存器 (CAN_RFIFOMIx) (x = 0, 1)

地址偏移: 0x1B0, 0x1C0

复位值: 0xFFFF XXXX

该寄存器只能按字 (32位) 访问。



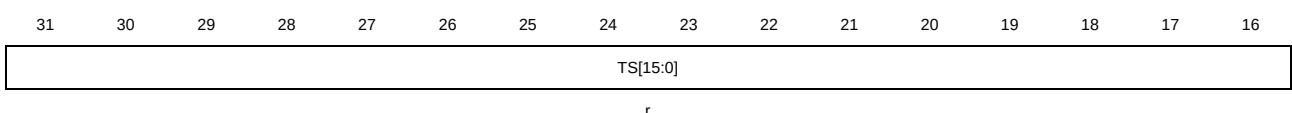
位	区域	说明
31:21	SFID[10:0]/EFID[28:18]	标识符
	8]	SFID[10:0]: 标准格式帧标识符 EFID[28:18]: 扩展格式帧标识符
20:16	EFID[17:13]	标识符 EFID[17:13]: 扩展格式帧标识符
15:3	EFID[12:0]	标识符 EFID[12:0]: 扩展格式帧标识符
2	FF	帧格式 0: 标准格式帧 1: 扩展格式帧
1	FT	帧种类 0: 数据帧 1: 遥控帧
0	保留	必须保持复位值。

28.4.14. 接收 FIFO 邮箱属性寄存器 (CAN_RFIFOMPx) (x = 0, 1)

地址偏移: 0x1B4, 0x1C4

复位值: 0xFFFF XXXX

该寄存器只能按字 (32位) 访问。



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

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
7:4	保留	必须保持复位值。
3:0	DLENC[3:0]	数据长度 DLENC[3:0]表示帧内数据长度。

28.4.15. 接收 FIFO 邮箱 data0 寄存器（CAN_RFIFOMDATA0x）（x = 0, 1）

地址偏移：0x1B8，0x1C8

复位值：0XXXXX XXXX

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DB3[7:0]								DB2[7:0]							
r								r							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB1[7:0]								DB0[7:0]							
r								r							

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

28.4.16. 接收 FIFO 邮箱 data1 寄存器（CAN_RFIFOMDATA1x）（x = 0, 1）

地址偏移：0x1BC，0x1CC

复位值：0XXXXX XXXX

该寄存器只能按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DB7[7:0]								DB6[7:0]							
r								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB5[7:0]								DB4[7:0]							
r								r							

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

28.4.17. 过滤器控制寄存器（CAN_FCTL）

地址偏移：0x200

复位值：0x2A1C 1C01

该寄存器只能按字（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															FLD
rw															

位/位域	名称	描述
31:1	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

28.4.18. 过滤器模式配置寄存器（CAN_FMCFG）

地址偏移：0x204

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

28.4.19. 过滤器位宽配置寄存器 (CAN_FSCFG)

地址偏移: 0x20C

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

28.4.20. 过滤器关联 FIFO 寄存器 (CAN_FAFIFO)

地址偏移: 0x214

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

28.4.21. 过滤器激活寄存器 (CAN_FW)

地址偏移: 0x21C

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
				r/w	r/w	r/w	r/w	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
FW15	FW14	FW13	FW12	FW11	FW10	FW9	FW8	FW7	FW6	FW5	FW4	FW3	FW2	FW1	FW0
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
31:28	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

28.4.22. 过滤器 (x) 数据 (y) 寄存器 (CAN_FxDATAy) (x = 0...27, y = 0, 1)地址偏移: $0x240 + 8 * x + 4 * y$, (x = 0...27, y = 0, 1)

复位值: 0xFFFF XXXX

该寄存器只能按字 (32位) 访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FD31	FD30	FD29	FD28	FD27	FD26	FD25	FD24	FD23	FD22	FD21	FD20	FD19	FD18	FD17	FD16
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
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
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
31:0	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

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
读/写(rw)	软件可以对这个位进行读写。
只读(r)	软件只能对这个位进行读。
只写(w)	软件只能对这个位进行写。读取该位将返回复位值。
读/写 1 清零(rc_w1)	软件可以读该位，对该位写入 1 可以清除这个位。写入 0 对位值没有影响。
读/写 0 清零(rc_w0)	软件可以读该位，对该位写入 0 可以清除这个位。写入 1 对位值没有影响。
翻转(t)	软件可以通过写 1 来翻转该位。写入 0 对位值没有效果。
只读/写 1 触发 (rt_w1)	软件可以读该位，写入 1 触发事件，但对位值没有影响。

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。
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.	说明	日期
1.0	首次发布	2026 年 3 月 4 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.