

**N32G033 系列**

**基于 32 位 ARM Cortex<sup>®</sup>-M0 微控制器**

**用户手册 V1.1.0**

## 目录

目录.....	I
表目录.....	XI
图目录.....	XIV
<b>1 文中的缩写.....</b>	<b>1</b>
1.1 寄存器描述表中使用的缩写列表.....	1
1.2 可用的外设.....	1
<b>2 总线架构.....</b>	<b>2</b>
2.1 系统架构.....	2
2.1.1 总线架构.....	2
2.1.2 总线地址映射.....	3
2.1.3 启动管理.....	5
2.2 存储系统 (Memory system).....	7
2.2.1 FLASH 规格.....	7
2.2.2 SRAM.....	16
2.2.3 FLASH 寄存器描述.....	16
<b>3 电源控制 (PWR).....</b>	<b>24</b>
3.1 通用描述.....	24
3.1.1 电源.....	24
3.1.2 电压监控.....	25
3.2 低功耗模式.....	27
3.2.1 STOP 模式.....	28
3.3 Debug 模式.....	29
3.3.1 外设调试支持.....	29
3.4 PWR 寄存器.....	29
3.4.1 寄存器总览.....	29
3.4.2 电源控制寄存器 (PWR_CTRL).....	30
3.4.3 电源控制状态寄存器 (PWR_CTRLSTS).....	31
3.4.4 电源控制寄存器 2 (PWR_CTRL2).....	31
3.4.5 调试控制寄存器 (DBG_CTRL).....	32
<b>4 复位和时钟控制(RCC).....</b>	<b>34</b>
4.1 复位控制单元.....	34
4.1.1 电源复位.....	34
4.1.2 系统复位.....	34
4.2 时钟控制单元.....	35
4.2.1 时钟树.....	36
4.2.2 HSI 时钟.....	37
4.2.3 LSI 时钟.....	37
4.2.4 系统时钟(SYSCLK)选择.....	37
4.2.5 看门狗时钟.....	37
4.2.6 时钟输出(MCO).....	37
4.2.7 时钟校准.....	37
4.3 RCC 寄存器.....	38
4.3.1 寄存器总览.....	38
4.3.2 时钟控制寄存器 (RCC_CTRL).....	39
4.3.3 时钟配置寄存器 (RCC_CFG).....	40
4.3.4 时钟状态寄存器 (RCC_CLKINT).....	41
4.3.5 APB 外设复位寄存器 (RCC_APBRS).....	42

4.3.6 AHB 外设时钟使能寄存器 (RCC_AHBCLKEN)	44
4.3.7 APB 外设时钟使能寄存器 (RCC_APBCLKEN)	44
4.3.8 控制/状态寄存器 (RCC_CTRLSTS)	46
4.3.9 AHB 外设复位寄存器 (RCC_AHBPRST)	47
4.3.10 时钟配置寄存器 2 (RCC_CFG2)	48
4.3.11 时钟校准计数寄存器 (RCC_LSICAL)	49
4.3.12 EMC 控制寄存器 (RCC_EMCTRL)	49
4.3.13 LSI 校准控制寄存器 (RCC_LSCTRL)	50
4.3.14 滤波时钟配置寄存器 (RCC_TIMFILTCFG)	51
<b>5 中断和事件</b>	<b>53</b>
5.1 嵌套向量中断寄存器	53
5.1.1 SysTick 校准值寄存器	53
5.1.2 中断和异常向量	53
5.2 外部中断/事件控制器 (EXTI)	54
5.2.1 简介	54
5.2.2 主要特性	55
5.2.3 功能描述	55
5.2.4 EXTI 线路映射	56
5.3 EXTI 寄存器	57
5.3.1 EXTI 寄存器总览	57
5.3.2 EXTI 事件屏蔽寄存器 (EXTI_EMASK)	57
5.3.3 EXTI 中断屏蔽寄存器 (EXTI_IMASK)	58
5.3.4 EXTI 下降沿触发配置寄存器 (EXTI_FT_CFG)	58
5.3.5 EXTI 上升沿触发配置寄存器 (EXTI_RT_CFG)	59
5.3.6 EXTI 挂起配置寄存器 (EXTI_PEND)	59
5.3.7 EXTI 软中断使能寄存器 (EXTI_SWIE)	60
<b>6 GPIO 和 AFIO</b>	<b>60</b>
6.1 概述	60
6.2 功能描述	61
6.2.1 IO 模式配置	61
6.2.2 复位后状态	66
6.2.3 单独的位设置和位清除	66
6.2.4 外部中断/唤醒线	66
6.2.5 复用功能	67
6.2.6 外设的 IO 配置	74
6.2.7 GPIO 锁定机制	75
6.3 GPIO 寄存器	76
6.3.1 GPIO 寄存器总览	76
6.3.2 GPIO 端口模式寄存器 (GPIOx_PMODE)	77
6.3.3 GPIO 端口输出类型寄存器 (GPIOx_POTYPE)	78
6.3.4 GPIO 端口上下拉寄存器 (GPIOx_PUPD)	78
6.3.5 GPIO 端口输入数据寄存器 (GPIOx_PID)	79
6.3.6 GPIO 端口输出数据寄存器 (GPIOx_POD)	79
6.3.7 GPIO 端口位设置/清除寄存器 (GPIOx_PBSC)	80
6.3.8 GPIO 端口锁定寄存器 (GPIOx_PLOCK)	81
6.3.9 GPIO 复用功能低配置寄存器 (GPIOx_AFL)	82
6.3.10 GPIO 复用功能高配置寄存器 (GPIOx_AFH)	82
6.3.11 GPIO 端口位清除寄存器 (GPIOx_PBC)	83
6.3.12 GPIO 驱动能力配置寄存器 (GPIOB_DS)	84
6.4 AFIO 寄存器	84
6.4.1 AFIO 寄存器总览	84

6.4.2 AFIO 配置寄存器 (AFIO_CFG) .....	85
6.4.3 AFIO 外部中断配置寄存器 1 (AFIO_EXTI_CFG1) .....	86
6.4.4 AFIO 外部中断配置寄存器 2 (AFIO_EXTI_CFG2) .....	87
6.4.5 数字毛刺滤波器配置寄存器 1 (AFIO_DIGEFT_CFG1) .....	88
6.4.6 数字毛刺滤波器配置寄存器 2 (AFIO_DIGEFT_CFG2) .....	88
6.4.7 数字毛刺滤波器配置寄存器 3 (AFIO_DIGEFT_CFG3) .....	89
<b>7 DMA 控制器.....</b>	<b>89</b>
7.1 简介 .....	89
7.2 主要特性 .....	90
7.3 功能框图 .....	91
7.4 功能描述 .....	91
7.4.1 DMA 操作 .....	91
7.4.2 通道优先级和仲裁器 .....	92
7.4.3 DMA 通道和传输数量 .....	92
7.4.4 可编程的数据位宽 .....	92
7.4.5 外设/内存地址递增 .....	94
7.4.6 通道配置流程 .....	94
7.4.7 流量控制 .....	95
7.4.8 循环模式 .....	95
7.4.9 错误管理 .....	95
7.4.10 中断 .....	95
7.4.11 DMA 请求映射 .....	96
7.5 DMA 寄存器 .....	97
7.5.1 DMA 寄存器总览 .....	97
7.5.2 DMA 中断状态寄存器 (DMA_INTSTS) .....	98
7.5.3 DMA 中断标志清除寄存器 (DMA_INTCLR) .....	99
7.5.4 DMA 通道 x 配置寄存器 (DMA_CHCFGx) .....	100
7.5.5 DMA 通道 x 传输数量寄存器 (DMA_TXNUMx) .....	101
7.5.6 DMA 通道 x 外设基地址寄存器 (DMA_PADDRx) .....	102
7.5.7 DMA 通道 x 存储器基地址寄存器 (DMA_MADDRx) .....	102
7.5.8 DMA 通道 x 请求选择寄存器 (DMA_CHSELx) .....	103
<b>8 内部集成电路总线(I<sup>2</sup>C).....</b>	<b>104</b>
8.1 简介 .....	104
8.2 主要特性 .....	104
8.3 功能描述 .....	104
8.3.1 SDA/SCL 控制 .....	104
8.3.2 软件通讯流程 .....	105
8.3.3 错误条件 .....	114
8.3.4 DMA 应用 .....	115
8.3.5 包错误校验 (PEC) .....	116
8.3.6 超时错误 .....	117
8.3.7 SMBus .....	117
8.3.8 噪声滤波 .....	119
8.4 调试模式 .....	119
8.5 中断请求 .....	119
8.6 I <sup>2</sup> C 寄存器描述 .....	120
8.6.1 I <sup>2</sup> C 寄存器总览 .....	120
8.6.2 I <sup>2</sup> C 控制寄存器 1 (I2C_CTRL1) .....	121
8.6.3 I <sup>2</sup> C 控制寄存器 2 (I2C_CTRL2) .....	123
8.6.4 I <sup>2</sup> C 自身地址寄存器 1 (I2C_OADDR1) .....	125
8.6.5 I <sup>2</sup> C 自身地址寄存器 2 (I2C_OADDR2) .....	126
8.6.6 I <sup>2</sup> C 数据寄存器 (I2C_DAT) .....	126

8.6.7 I <sup>2</sup> C 状态寄存器 1 (I2C_STS1) .....	126
8.6.8 I <sup>2</sup> C 状态寄存器 2 (I2C_STS2) .....	130
8.6.9 I <sup>2</sup> C 时钟控制寄存器 (I2C_CLKCTRL) .....	131
8.6.10 I <sup>2</sup> C 上升时间寄存器 (I2C_TMRISE).....	132
8.6.11 I <sup>2</sup> C 滤波控制寄存器 (I2C_GFLTRCTRL).....	132
<b>9 通用异步收发器(UART) .....</b>	<b>134</b>
9.1 简介 .....	134
9.2 主要特性 .....	134
9.3 功能框图 .....	135
9.4 功能描述 .....	135
9.4.1 UART 帧格式 .....	136
9.4.2 发送器 .....	136
9.4.3 接收器 .....	139
9.4.4 分数波特率计算 .....	141
9.4.5 UART 接收器容忍时钟的变化.....	144
9.4.6 校验控制 .....	145
9.4.7 DMA 通信.....	145
9.4.8 多处理器通信 .....	147
9.4.9 单线半双工模式 .....	149
9.4.10 串行 IrDA 红外编解码模式.....	149
9.4.11 LIN 模式 .....	150
9.4.12 UART3 低功耗唤醒 .....	152
9.5 中断请求 .....	153
9.6 模式配置 .....	153
9.7 UART 寄存器 .....	154
9.7.1 UART 寄存器总览 .....	154
9.7.2 UART 控制寄存器 1(UART_CTRL1).....	154
9.7.3 UART 控制寄存器 2(UART_CTRL2).....	157
9.7.4 UART 控制寄存器 3(UART_CTRL3).....	158
9.7.5 UART 状态寄存器 (UART_STS) .....	159
9.7.6 UART 数据寄存器(UART_DAT).....	161
9.7.7 UART 波特率配置寄存器 (UART_BRCF).....	162
9.7.8 UART 保护时间和预分频寄存器(UART_GTP).....	162
9.7.9 UART 接收超时寄存器(UART_RTO).....	163
9.7.10 UART 低功耗唤醒寄存器(UART_WKUP).....	163
<b>10 HDIV 除法器.....</b>	<b>165</b>
10.1 功能描述 .....	165
10.2 操作说明 .....	165
10.2.1 基本使用流程 .....	165
10.2.2 异常处理 .....	166
10.3 HDIV 寄存器 .....	166
10.3.1 HDIV 寄存器总览 .....	166
10.3.2 HDIV 控制状态寄存器 (HDIV_CTRLSTS) .....	166
10.3.3 HDIV 被除数寄存器 (HDIV_DIVIDEND) .....	167
10.3.4 HDIV 除数寄存器 (HDIV_DIVISOR) .....	168
10.3.5 HDIV 商寄存器 (HDIV_QUOTIENT) .....	168
10.3.6 HDIV 余数寄存器 (HDIV_REMAINDER) .....	168
10.3.7 HDIV 除零异常标志寄存器 (HDIV_DIVBY0) .....	169
<b>11 SQRT 开平方根器.....</b>	<b>170</b>
11.1 功能描述 .....	170
11.2 操作说明 .....	170

11.2.1 基本使用流程 .....	170
11.3 SQRT 寄存器 .....	171
11.3.1 SQRT 寄存器总览 .....	171
11.3.2 SQRT 控制状态寄存器 (SQRT_CTRLSTS) .....	171
11.3.3 SQRT 被开方数寄存器 (SQRT_RADICAND) .....	172
11.3.4 SQRT 根数寄存器 (SQRT_ROOT) .....	172
<b>12 高级定时器 (TIM1) .....</b>	<b>173</b>
12.1 TIM1 简介 .....	173
12.2 TIM1 主要特性 .....	173
12.3 TIM1 功能描述 .....	174
12.3.1 时基单元 .....	174
12.3.2 计数器模式 .....	175
12.3.3 重复计数器 .....	181
12.3.4 时钟选择 .....	184
12.3.5 捕获/比较通道 .....	187
12.3.6 输入捕获模式 .....	190
12.3.7 PWM 输入模式 .....	191
12.3.8 强制输出模式 .....	192
12.3.9 输出比较模式 .....	192
12.3.10 PWM 模式 .....	193
12.3.11 单脉冲模式 .....	195
12.3.12 在外部事件上清除 OCxREF 信号 .....	197
12.3.13 互补输出和死区插入 .....	197
12.3.14 刹车功能 .....	199
12.3.15 调试模式 .....	202
12.3.16 TIMx 定时器和外部触发的同步 .....	202
12.3.17 定时器同步 .....	207
12.3.18 触发 ADC .....	207
12.3.19 产生六步 PWM 输出 .....	207
12.3.20 编码器接口模式 .....	208
12.3.21 与霍尔传感器的接口 .....	210
12.4 TIM1 寄存器描述 .....	212
12.4.1 寄存器总览 .....	212
12.4.2 控制寄存器 1 (TIMx_CTRL1) .....	214
12.4.3 控制寄存器 2 (TIMx_CTRL2) .....	216
12.4.4 状态寄存器 (TIMx_STS) .....	218
12.4.5 事件产生寄存器 (TIMx_EVTGEN) .....	220
12.4.6 从模式控制寄存器 (TIMx_SMCTRL) .....	222
12.4.7 DMA/中断使能寄存器 (TIMx_DINTEN) .....	224
12.4.8 捕获/比较模式寄存器 1 (TIMx_CCMOD1) .....	226
12.4.9 捕获/比较模式寄存器 2 (TIMx_CCMOD2) .....	229
12.4.10 捕获/比较模式寄存器 3 (TIMx_CCMOD3) .....	231
12.4.11 捕获/比较使能寄存器 (TIMx_CCEN) .....	232
12.4.12 捕获/比较寄存器 1 (TIMx_CCDAT1) .....	235
12.4.13 捕获/比较寄存器 2 (TIMx_CCDAT2) .....	236
12.4.14 捕获/比较寄存器 3 (TIMx_CCDAT3) .....	237
12.4.15 捕获/比较寄存器 4 (TIMx_CCDAT4) .....	238
12.4.16 捕获/比较寄存器 5 (TIMx_CCDAT5) .....	239
12.4.17 捕获/比较寄存器 6 (TIMx_CCDAT6) .....	239
12.4.18 预分频器 (TIMx_PSC) .....	240
12.4.19 自动重载寄存器 (TIMx_AR) .....	240
12.4.20 计数器 (TIMx_CNT) .....	241

12.4.21 重复计数寄存器 (TIMx_REPCNT) .....	241
12.4.22 刹车和死区寄存器 (TIMx_BKDT) .....	242
12.4.23 捕获/比较寄存器 7 (TIMx_CCDA7) .....	243
12.4.24 刹车滤波寄存器 (TIMx_BKFR) .....	244
12.4.25 复用功能寄存器 1 (TIMx_AF1) .....	245
12.4.26 DMA 控制寄存器 (TIMx_DCTRL) .....	246
12.4.27 连续模式的 DMA 地址 (TIMx_DADDR) .....	247
<b>13 通用定时器 (TIM3) .....</b>	<b>249</b>
13.1 TIM3 简介 .....	249
13.2 TIM3 主要特性 .....	249
13.3 TIM3 功能描述 .....	250
13.3.1 时基单元 .....	250
13.3.2 计数器模式 .....	251
13.3.3 时钟选择 .....	256
13.3.4 捕获/比较通道 .....	260
13.3.5 输入捕获模式 .....	263
13.3.6 PWM 输入模式 .....	264
13.3.7 强制输出模式 .....	265
13.3.8 输出比较模式 .....	265
13.3.9 PWM 模式 .....	266
13.3.10 单脉冲模式 .....	268
13.3.11 在外部事件上清除 OCxREF 信号 .....	270
13.3.12 刹车功能 .....	270
13.3.13 调试模式 .....	272
13.3.14 外部事件触发装载 LVR .....	272
13.3.15 TIMx 定时器和外部触发的同步 .....	273
13.3.16 定时器同步 .....	273
13.3.17 编码器接口模式 .....	278
13.3.18 与霍尔传感器的接口 .....	280
13.4 TIMx 寄存器描述 (x=3) .....	280
13.4.1 寄存器总览 .....	280
13.4.2 控制寄存器 1 (TIMx_CTRL1) .....	282
13.4.3 控制寄存器 2 (TIMx_CTRL2) .....	284
13.4.4 状态寄存器 (TIMx_STS) .....	285
13.4.5 事件产生寄存器 (TIMx_EVTGEN) .....	287
13.4.6 从模式控制寄存器 (TIMx_SMCTRL) .....	289
13.4.7 DMA/中断使能寄存器 (TIMx_DINTEN) .....	291
13.4.8 捕获/比较模式寄存器 1 (TIMx_CCMOD1) .....	293
13.4.9 捕获/比较模式寄存器 2 (TIMx_CCMOD2) .....	296
13.4.10 捕获/比较使能寄存器 (TIMx_CCEN) .....	298
13.4.11 捕获/比较寄存器 1 (TIMx_CCDA1) .....	299
13.4.12 捕获/比较寄存器 2 (TIMx_CCDA2) .....	300
13.4.13 捕获/比较寄存器 3 (TIMx_CCDA3) .....	300
13.4.14 捕获/比较寄存器 4 (TIMx_CCDA4) .....	301
13.4.15 预分频器 (TIMx_PSC) .....	302
13.4.16 自动重装载寄存器 (TIMx_AR) .....	302
13.4.17 计数器 (TIMx_CNT) .....	303
13.4.18 刹车和死区寄存器 (TIMx_BKDT) .....	303
13.4.19 刹车滤波寄存器 (TIMx_BKFR) .....	304
13.4.20 复用功能寄存器 1 (TIMx_AF1) .....	305
13.4.21 外部事件计数器装载值寄存器 (TIMx_ENCLVR) .....	307
13.4.22 DMA 控制寄存器 (TIMx_DCTRL) .....	308

13.4.23 连续模式的 DMA 地址 (TIMx_DADDR) .....	309
<b>14 通用定时器 (TIM4) .....</b>	<b>310</b>
14.1 TIM4 简介 .....	310
14.2 TIM4 主要特性 .....	310
14.3 TIM4 功能描述 .....	311
14.3.1 时基单元 .....	311
14.3.2 计数器模式 .....	312
14.3.3 时钟选择 .....	317
14.3.4 捕获/比较通道 .....	321
14.3.5 输入捕获模式 .....	324
14.3.6 PWM 输入模式 .....	325
14.3.7 强制输出模式 .....	326
14.3.8 输出比较模式 .....	326
14.3.9 PWM 模式 .....	328
14.3.10 单脉冲模式 .....	330
14.3.11 在外部事件上清除 OCxREF 信号 .....	332
14.3.12 调试模式 .....	332
14.3.13 TI3 事件触发功能 .....	332
14.3.14 TIMx 定时器和外部触发的同步 .....	333
14.3.15 定时器同步 .....	333
14.3.16 编码器接口模式 .....	333
14.3.17 与霍尔传感器的接口 .....	341
14.4 TIMx 寄存器描述 (x=4) .....	341
14.4.1 寄存器总览 .....	342
14.4.2 控制寄存器 1 (TIMx_CTRL1) .....	344
14.4.3 控制寄存器 2 (TIMx_CTRL2) .....	345
14.4.4 状态寄存器 (TIMx_STS) .....	346
14.4.5 事件产生寄存器 (TIMx_EVTGEN) .....	348
14.4.6 从模式控制寄存器 (TIMx_SMCTRL) .....	349
14.4.7 DMA/中断使能寄存器 (TIMx_DINTEN) .....	351
14.4.8 捕获/比较模式寄存器 1 (TIMx_CCMOD1) .....	354
14.4.9 捕获/比较模式寄存器 2 (TIMx_CCMOD2) .....	357
14.4.10 捕获/比较使能寄存器 (TIMx_CCEN) .....	358
14.4.11 捕获/比较寄存器 1 (TIMx_CCDAT1) .....	359
14.4.12 捕获/比较寄存器 2 (TIMx_CCDAT2) .....	360
14.4.13 捕获/比较寄存器 3 (TIMx_CCDAT3) .....	360
14.4.14 预分频器 (TIMx_PSC) .....	361
14.4.15 自动重载寄存器 (TIMx_AR) .....	361
14.4.16 计数器 (TIMx_CNT) .....	362
14.4.17 通道 1 滤波寄存器 (TIMx_C1FILT) .....	362
14.4.18 通道 2 滤波寄存器 (TIMx_C2FILT) .....	363
14.4.19 通道 3 滤波寄存器 (TIMx_C3FILT) .....	364
14.4.20 输入通道滤波输出寄存器 (TIMx_FILTO) .....	365
14.4.21 编码器捕获数据寄存器 (TIMx_ENCDAT) .....	365
14.4.22 编码器模式控制寄存器 (TIMx_ENCMCTRL) .....	366
14.4.23 外部事件计数器装载值寄存器 (TIMx_ENCLVR) .....	367
14.4.24 DMA 控制寄存器 (TIMx_DCTRL) .....	368
14.4.25 连续模式的 DMA 地址 (TIMx_DADDR) .....	369
<b>15 基本定时器 (TIM6).....</b>	<b>370</b>
15.1 基本定时器简介 .....	370
15.2 基本定时器主要特性 .....	370

15.3 基础定时器描述 .....	370
15.3.1 时基单元 .....	370
15.3.2 计数模式 .....	371
15.3.3 时钟选择 .....	374
15.3.4 调试模式 .....	374
15.4 TIM6 寄存器描述 .....	374
15.4.1 寄存器总览 .....	374
15.4.2 控制寄存器 1 (TIMx_CTRL1) .....	375
15.4.3 状态寄存器 (TIMx_STS) .....	376
15.4.4 事件产生寄存器 (TIMx_EVTGEN) .....	377
15.4.5 DMA/中断使能寄存器 (TIMx_DINTEN) .....	377
15.4.6 预分频器 (TIMx_PSC) .....	378
15.4.7 自动重装载寄存器 (TIMx_AR) .....	378
15.4.8 计数器 (TIMx_CNT) .....	379
<b>16 模拟/数字转换(ADC) .....</b>	<b>380</b>
16.1 简述 .....	380
16.2 ADC 主要特征 .....	380
16.3 ADC 功能描述 .....	381
16.3.1 ADC 开关控制 .....	382
16.3.2 ADC 时钟 .....	382
16.3.3 通道选择 .....	382
16.3.4 单次转换模式 .....	383
16.3.5 连续转换模式 .....	383
16.3.6 段采样模式 .....	384
16.3.7 时序图 .....	385
16.3.8 模拟看门狗 .....	385
16.3.9 数据对齐 .....	385
16.3.10 可编程的通道采样时间 .....	386
16.3.11 触发转换 .....	386
16.3.12 ADC 中断 .....	387
16.4 ADC 寄存器 .....	387
16.4.1 ADC 寄存器总览 .....	387
16.4.2 ADC 状态寄存器(ADC_STS) .....	389
16.4.3 ADC 控制寄存器 1(ADC_CTRL1) .....	391
16.4.4 ADC 控制寄存器 2(ADC_CTRL2) .....	391
16.4.5 ADC 控制寄存器 3(ADC_CTRL3) .....	392
16.4.6 ADC 采样时间寄存器 1(ADC_SAMPT1) .....	393
16.4.7 ADC 采样时间寄存器 2(ADC_SAMPT2) .....	394
16.4.8 ADC 看门狗高阈值寄存器(ADC_AWDHIGH) .....	395
16.4.9 ADC 看门狗高阈值寄存器(ADC_AWDLOW) .....	395
16.4.10 ADC 看门狗使能寄存器(ADC_AWDEN) .....	395
16.4.11 ADC 段配置寄存器(ADC_PHCFG) .....	396
16.4.12 ADC 触发选择寄存器(ADC_TRIGSEL) .....	397
16.4.13 ADC 软件配置寄存器(ADC_PHSWTRIG) .....	399
16.4.14 ADC 序列寄存器 1(ADC_SEQ1) .....	399
16.4.15 ADC 序列寄存器 2(ADC_SEQ2) .....	400
16.4.16 ADC 规则数据寄存器 x(ADC_DATx)(x = 0..9) .....	400
<b>17 比较器 (COMP) .....</b>	<b>402</b>
17.1 COMP 系统连接框图 .....	402
17.2 COMP 特性 .....	403
17.3 COMP 配置流程 .....	403

17.4 COMP 工作模式.....	404
17.4.1 独立比较器.....	404
17.5 比较器互联关系.....	404
17.6 中断.....	406
17.7 COMP 寄存器.....	407
17.7.1 COMP 寄存器总览.....	407
17.7.2 COMP 中断使能寄存器(COMP_INTEN).....	408
17.7.3 COMP 中断状态寄存器(COMP_INTSTS).....	408
17.7.4 COMP 锁定寄存器(COMP_LOCK).....	408
17.7.5 COMP 控制寄存器(COMP_CTRL).....	409
17.7.6 COMP 滤波控制寄存器(COMP_FILC).....	410
17.7.7 COMP 滤波时钟寄存器(COMP_FILP).....	411
17.7.8 COMP 控制寄存器(COMP_INVREF).....	411
<b>18 OPAMP.....</b>	<b>413</b>
18.1 OPAMP 特性.....	413
18.2 OPAMP 功能描述.....	414
18.3 OPAMP 工作模式.....	415
18.3.1 OPAMP1 外部放大模式(仅 OPA1 支持).....	415
18.3.2 OPAMP1 跟随模式.....	415
18.3.3 OPAMP1 单端增益 (PGA) 模式.....	416
18.3.4 差分增益模式.....	417
18.3.5 OPAMP 写保护.....	417
18.4 OPAMP 寄存器.....	418
18.4.1 OPAMP 寄存器总览.....	418
18.4.2 OPAMP 控制寄存器 1 (OPAMP1_CS).....	418
18.4.3 OPAMP 控制寄存器 2 (OPAMP2_CS).....	420
18.4.4 OPAMP 控制寄存器 3 (OPAMP3_CS).....	420
18.4.5 OPAMP 锁定寄存器 (OPAMP_LOCK).....	421
18.4.6 OPAMP 参考电压选择寄存器 (OPAMP_VREFSEL).....	421
<b>19 独立看门狗 (IWDG).....</b>	<b>423</b>
19.1 简介.....	423
19.2 主要特性.....	423
19.3 功能描述.....	424
19.3.1 寄存器访问保护.....	424
19.3.2 IWDG 中断.....	424
19.3.3 调试模式.....	425
19.4 用户界面.....	425
19.4.1 操作流程.....	425
19.4.2 IWDG 配置流程.....	425
19.5 IWDG 寄存器.....	426
19.5.1 IWDG 寄存器总览.....	426
19.5.2 IWDG 密钥寄存器 (IWDG_KEY).....	426
19.5.3 IWDG 状态寄存器 (IWDG_STS).....	427
19.5.4 IWDG 预分频寄存器 (IWDG_PREDIV).....	427
19.5.5 IWDG 重装载寄存器 (IWDG_RELV).....	428
19.5.6 IWDG 控制寄存器 (IWDG_CTRL).....	428
<b>20 串行外设接口 (SPI).....</b>	<b>430</b>
20.1 SPI 简介.....	430
20.2 SPI 主要特性.....	430
20.3 SPI 功能描述.....	431
20.3.1 通用描述.....	431

20.3.2 SPI 工作模式 .....	434
20.3.3 状态标志 .....	439
20.3.4 关闭 SPI .....	439
20.3.5 使用 DMA 进行 SPI 通讯 .....	440
20.3.6 CRC 计算 .....	441
20.3.7 错误标志位 .....	442
20.3.8 SPI 中断 .....	443
20.4 SPI 寄存器描述 .....	444
20.4.1 SPI 寄存器总览 .....	444
20.4.2 SPI 控制寄存器 1 (SPI_CTRL1) .....	445
20.4.3 SPI 控制寄存器 2 (SPI_CTRL2) .....	446
20.4.4 SPI 状态寄存器 (SPI_STS) .....	448
20.4.5 SPI 数据寄存器 (SPI_DAT) .....	449
20.4.6 SPI Tx CRC 寄存器 (SPI_CRCTDAT) .....	449
20.4.7 SPI Rx CRC 寄存器 (SPI_CRCRDAT) .....	450
20.4.8 SPI CRC 多项式寄存器 (SPI_CRCPOLY) .....	450
20.4.9 SPI 时钟采样延迟寄存器 (SPI_CTRL3) .....	451
<b>21 调试支持 (DBG) .....</b>	<b>453</b>
21.1 简介 .....	453
21.2 SWD 功能 .....	454
21.2.1 引脚分配 .....	454
<b>22 唯一设备序列号 (UID) .....</b>	<b>455</b>
22.1 简介 .....	455
22.2 UID 寄存器 .....	455
22.3 UCID 寄存器 .....	455
22.4 DBGMCU_ID 寄存器 .....	455
<b>23 版本历史 .....</b>	<b>456</b>
<b>24 声明 .....</b>	<b>457</b>

## 表目录

表 2-1 外设寄存器地址列表.....	4
表 2-2 启动模式列表.....	6
表 2-3 存储总线地址列表.....	7
表 2-4 选项字节列表.....	11
表 2-5 读保护配置列表.....	12
表 2-6 存储区读写擦 <sup>(1)</sup> 权限控制表.....	14
表 2-7 FLASH 寄存器总览.....	16
表 3-1 电源域说明.....	24
表 3-2 电源模式.....	27
表 3-3 外设运行状态.....	28
表 3-4 PWR 寄存器总览.....	29
表 4-1 RCC 寄存器总览.....	38
表 5-1 向量表.....	53
表 5-2 EXTI 寄存器总览.....	57
表 6-1 IO 模式和配置关系.....	62
表 6-2 IO 不同配置的输入输出特性.....	62
表 6-3 外部中断通用 I/O 映像.....	67
表 6-4 SWD 复用功能 I/O 重映射.....	67
表 6-5 TIM1 复用功能 I/O 重映射.....	67
表 6-6 TIM3 复用功能 I/O 重映射.....	69
表 6-7 TIM4 复用功能 I/O 重映射.....	70
表 6-8 UART1 复用功能 I/O 重映射.....	70
表 6-9 UART2 复用功能 I/O 重映射.....	71
表 6-10 UART3 复用功能 I/O 重映射.....	71
表 6-11 I2C1 复用功能 I/O 重映射.....	72
表 6-12 I2C2 复用功能 I/O 重映射.....	72
表 6-13 SPI1 管脚重映射.....	73
表 6-14 SPI2 管脚重映射.....	73
表 6-15 COMPx 复用功能 I/O 重映射.....	73
表 6-16 EVENTOUT 复用功能 I/O 重映射.....	74
表 6-17 RCC 复用功能 I/O 重映射.....	74
表 6-18 ADC.....	74
表 6-19 TIM1/3/4.....	74
表 6-20 UART.....	74

表 6-21 I2C.....	75
表 6-22 SPI.....	75
表 6-23 COMP.....	75
表 6-24 其他.....	75
表 6-25 AFIO 寄存器总览.....	85
表 7-1 可编程的数据宽度和大小端操作(当 PINC = MINC = 1).....	92
表 7-2 流量控制表.....	95
表 7-3 DMA 中断请求.....	96
表 7-4 DMA 请求映射.....	97
表 7-5 DMA 寄存器总览.....	97
表 8-1 SMBus 与 I <sup>2</sup> C 的比较.....	118
表 8-2 I <sup>2</sup> C 中断请求.....	119
表 8-3 I <sup>2</sup> C 寄存器总览.....	120
表 9-1 停止位配置.....	137
表 9-2 噪声检测的数据采样.....	141
表 9-3 设置波特率时的误差计算.....	143
表 9-4 当 DIV_Decimal =0 时, UART 接收器的容忍度.....	144
表 9-5 当 DIV_Decimal !=0 时, UART 接收器的容忍度.....	145
表 9-6 帧格式.....	145
表 9-7 UART 中断请求.....	153
表 9-8 UART 模式设置 <sup>(1)</sup> .....	153
表 9-9 UART 寄存器总览.....	154
表 10-1 HDIV 寄存器总览.....	166
表 11-1 SQRT 寄存器总览.....	171
表 12-1 计数方向与编码器信号的关系.....	209
表 12-2 TIM1 寄存器总览.....	212
表 12-3 TIMx 内部触发连接.....	224
表 12-4 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位.....	234
表 13-1 计数方向与编码器信号的关系.....	279
表 13-2 TIM3 寄存器总览.....	280
表 13-3 TIMx 内部触发连接.....	291
表 13-4 标准 OCx 的输出控制位.....	299
表 14-1 计数方向与编码器信号的关系 (CC1P=CC2P=0).....	334
表 14-2 计数方向与编码器信号和极性设置的关系.....	338
表 14-3 计数方向与编码器信号和极性设置的关系.....	340

表 14-4 TIM4 寄存器总览.....	342
表 14-5 TIMx 内部触发连接.....	351
表 14-6 标准 OCx 的输出控制位.....	359
表 15-1 寄存器总览.....	374
表 16-1 ADC 引脚.....	382
表 16-2 通道对应表.....	382
表 16-3 ADC 触发与通道选择.....	386
表 16-4 ADC 中断.....	387
表 17-1 OUT 引脚.....	404
表 17-2 INP 引脚配置.....	404
表 17-3 INM 引脚配置.....	404
表 17-4 COMP 寄存器总览.....	407
表 18-1 OPAMP 寄存器总览.....	418
表 19-1 IWDG 计数最大和最小复位时间.....	425
表 19-2 IWDG 寄存器总览.....	426
表 20-1 SPI 中断请求.....	443
表 20-2 SPI 寄存器总览.....	444
表 21-1 调试端口引脚.....	454
表 22-1 DBGMCU_ID 位描述.....	455

## 图目录

图 2-1 总线架构图.....	3
图 2-2 总线地址映射图.....	4
图 3-1 电源框图.....	25
图 3-2 上电复位和掉电复位的波形图.....	26
图 3-3 PVD 阈值图.....	27
图 4-1 复位电路.....	35
图 4-2 时钟树.....	36
图 5-1 外部中断/事件控制器框图.....	55
图 5-2 外部中断通用 I/O 映射.....	56
图 6-1 I/O 端口的基本结构（不支持 Fail-safe）.....	61
图 6-2 输入浮空/上拉/下拉模式（不支持 Fail-safe）.....	63
图 6-3 输出模式（不支持 Fail-safe）.....	64
图 6-4 复用功能模式（不支持 Fail-safe）.....	65
图 6-5 高阻抗的模拟功能模式（不支持 Fail-safe）.....	66
图 7-1 DMA 框图.....	91
图 8-1 I <sup>2</sup> C 功能框图.....	106
图 8-2 I <sup>2</sup> C 总线协议.....	106
图 8-3 从发送器传送序列.....	108
图 8-4 从接收器传送序列.....	110
图 8-5 主发送器传送序列.....	112
图 8-6 主接收器传送序列.....	114
图 9-1 UART 框图.....	135
图 9-2 字长=8 设置.....	136
图 9-3 字长=9 设置.....	136
图 9-4 停止位配置.....	137
图 9-5 发送时差.....	138
图 9-6 发送时 TXC/TXDE 的变化情况.....	139
图 9-7 起始位检测.....	140
图 9-8 DMA 发送.....	146
图 9-9 DMA 接收.....	147
图 9-10 静默模式下的空闲总线检测.....	148
图 9-11 静默模式下的地址标识检测.....	148
图 9-12 IrDA SIR ENDEC-框图.....	150
图 9-13 IrDA 数据调制(3/16)-正常模式.....	150

图 9-14 LIN 模式下的断开检测（11 位断开帧长度-设置了 LINBDL 位） .....	151
图 9-15 LIN 模式下的断开检测与帧错误的检测.....	152
图 12-1 TIM1 框图.....	174
图 12-2 当预分频的参数从 1 到 4，计数器的时序图 .....	175
图 12-3 当内部时钟分频因子 = 2/N 时，向上计数的时序图.....	176
图 12-4 当 ARPEN=0/1 产生更新事件时，向上计数的时序图.....	177
图 12-5 内部时钟分频因子 = 2/N 时，向下计数时序图.....	178
图 12-6 内部时钟分频因子 = 2/N，中央对齐时序图.....	179
图 12-7 包含计数器上溢和下溢的中央对齐时序图(ARPEN=1) .....	180
图 12-8 非对称模式对应的输出波形 .....	181
图 12-9 向下计数模式下的重复计数时序图 .....	182
图 12-10 向上计数模式下的重复计数时序列图 .....	183
图 12-11 中央对齐模式下的重复计数时序图 .....	183
图 12-12 正常模式下的控制电路，内部时钟除以 1 .....	184
图 12-13 TI2 外部时钟连接示例 .....	185
图 12-14 外部时钟模式 1 的控制电路 .....	186
图 12-15 外部触发输入框图.....	186
图 12-16 外部时钟模式 2 的控制电路 .....	187
图 12-17 捕获/比较通道（例如：通道 1 输入级） .....	188
图 12-18 捕获/比较通道 1 主电路.....	189
图 12-19 通道 x 的输出部分（以通道 1 为例子） .....	190
图 12-20 PWM 输入模式时序 .....	191
图 12-21 输出比较模式，开启 OC1.....	193
图 12-22 中央对齐的 PWM 波形 (AR=8).....	194
图 12-23 边沿对齐 PWM 波形 (AR=8).....	195
图 12-24 单脉冲模式示例.....	196
图 12-25 清除 TIMx 的 OCxREF.....	197
图 12-26 带死区插入的互补输出 .....	198
图 12-27 刹车输入.....	199
图 12-28 响应刹车的输出行为.....	201
图 12-29 滑动滤波.....	201
图 12-30 复位模式下的控制电路 .....	202
图 12-31 触发器模式下的控制电路 .....	203
图 12-32 门控模式下的控制电路 .....	204
图 12-33 外部时钟模式 2+触发模式下的控制电路 .....	205

图 12-34 组合复位+触发模式下的控制电路.....	206
图 12-35 组合门控+复位模式下的控制电路.....	207
图 12-36 产生六步 PWM, 使用 COM 的例子 (OSSR=1) .....	208
图 12-37 编码器模式下的计数器操作实例 .....	209
图 12-38 IC1FP1 反相的编码器接口模式实例.....	210
图 12-39 霍尔传感器接口的实例.....	211
图 13-1 TIM3 框图.....	250
图 13-2 当预分频的参数从 1 到 4, 计数器的时序图 .....	251
图 13-3 当内部时钟分频因子 = 2/N 时, 向上计数的时序图.....	252
图 13-4 当 ARPEN=0/1 产生更新事件时, 向上计数的时序图.....	253
图 13-5 内部时钟分频因子 = 2/N 时, 向下计数时序图.....	254
图 13-6 内部时钟分频因子 = 2/N, 中央对齐时序图.....	255
图 13-7 包含计数器上溢和下溢的中央对齐时序图(ARPEN=1) .....	256
图 13-8 正常模式下的控制电路, 内部时钟除以 1 .....	257
图 13-9 TI2 外部时钟连接示例 .....	258
图 13-10 外部时钟模式 1 的控制电路 .....	259
图 13-11 外部触发输入框图 .....	259
图 13-12 外部时钟模式 2 的控制电路 .....	260
图 13-13 捕获/比较通道 (例如: 通道 1 输入级) .....	261
图 13-14 捕获/比较通道 1 主电路.....	262
图 13-15 通道 x 的输出部分 (以通道 4 为例子) .....	263
图 13-16 PWM 输入模式时序 .....	264
图 13-17 输出比较模式, 开启 OC1.....	266
图 13-18 中央对齐的 PWM 波形 (AR=8).....	267
图 13-19 边沿对齐 PWM 波形 (AR=8).....	268
图 13-20 单脉冲模式示例.....	269
图 13-21 清除 TIMx 的 OCxREF.....	270
图 13-22 刹车输入.....	271
图 13-23 滑动滤波.....	272
图 13-24 外部事件触发装载 LVR 时, CNT 计数方式.....	273
图 13-25 主/从定时器的例子.....	273
图 13-26 定时器 3 由定时器 1 的 OC1REF 门控.....	275
图 13-27 定时器 3 由定时器 1 的使能门控 .....	276
图 13-28 使用定时器 1 的更新触发定时器 3 .....	277
图 13-29 使用定时器 1 的 TI1 输入触发定时器 1 和定时器 3.....	278

图 13-30 编码器模式下的计数器操作实例 .....	279
图 13-31 IC1FP1 反相的编码器接口模式实例 .....	280
图 14-1 TIM4 框图 .....	311
图 14-2 当预分频的参数从 1 到 4, 计数器的时序图 .....	312
图 14-3 当内部时钟分频因子 = 2/N 时, 向上计数的时序图 .....	313
图 14-4 当 ARPEN=0/1 产生更新事件时, 向上计数的时序图 .....	314
图 14-5 内部时钟分频因子 = 2/N 时, 向下计数时序图 .....	315
图 14-6 内部时钟分频因子 = 2/N, 中央对齐时序图 .....	316
图 14-7 包含计数器上溢和下溢的中央对齐时序图(ARPEN=1) .....	317
图 14-8 正常模式下的控制电路, 内部时钟除以 1 .....	318
图 14-9 TI2 外部时钟连接示例 .....	319
图 14-10 外部时钟模式 1 的控制电路 .....	320
图 14-11 外部触发输入框图 .....	320
图 14-12 外部时钟模式 2 的控制电路 .....	321
图 14-13 捕获/比较通道 (例如: 通道 1 输入级) .....	322
图 14-14 捕获/比较通道 1 主电路 .....	323
图 14-15 通道 x 的输出部分 (以通道 1 为例子) .....	324
图 14-16 滑动滤波 .....	325
图 14-17 PWM 输入模式时序 .....	326
图 14-18 输出比较模式, 开启 OC1 .....	327
图 14-19 中央对齐的 PWM 波形 (AR=8) .....	329
图 14-20 边沿对齐 PWM 波形 (AR=8) .....	330
图 14-21 单脉冲模式示例 .....	331
图 14-22 清除 TIMx 的 OCxREF .....	332
图 14-23 外部事件触发装载 .....	333
图 14-24 编码器仅在 TI1 计数 .....	335
图 14-25 编码器仅在 TI2 计数 .....	335
图 14-26 编码器在 TI1 和 TI2 上计数 .....	335
图 14-27 T2 是高电平时, 计数器只在 TI1 计数 .....	336
图 14-28 T1 是高电平时, 计数器只在 TI2 计数 .....	336
图 14-29 编码器模式下的计数器操作实例 .....	337
图 14-30 IC1FP1 反相的编码器接口模式实例 .....	337
图 14-31 脉冲电平编码模式 (CC1P=CC2P=0) .....	338
图 14-32 双脉冲编码模式 (CC1P = CC2P = 0) .....	339
图 14-33 双脉冲编码模式 (CC1P = CC2P = 1) .....	340

图 14-34 CCW/CW 编码模式 2 .....	341
图 14-35 CCW/CW 编码模式 1 (CC1P=CC2P=0) .....	341
图 15-1 TIM6 框图.....	370
图 15-2 预分频器分频从 1 到 4 的计数器时序图 .....	371
图 15-3 向上计数时序图, 内部时钟分频因子 = 2/N.....	372
图 15-4 ARPEN=0/1 时向上计数、更新事件的时序图 .....	373
图 15-5 正常模式下的控制电路, 内部时钟分频系数为 1 .....	374
图 16-1 ADC 框图.....	381
图 16-2 ADC 段转换图.....	384
图 16-3 ADC 转换.....	385
图 17-1 比较器系统连接图 .....	402
图 18-1 OPAMP 系统连接图.....	414
图 18-2 OPAMP1 外部放大模式.....	415
图 18-3 OPAMP1 跟随模式.....	416
图 18-4 内部增益模式 .....	416
图 18-5 差分增益模式 .....	417
图 19-1 独立看门狗功能框图 .....	424
图 20-1 SPI 框图 .....	431
图 20-2 硬件/软件的从选择管理 .....	432
图 20-3 单主和单从应用 .....	432
图 20-4 数据时钟时序图.....	433
图 20-5 主机全双工模式下连续传输时, SPI_STS.TE/RNE/BUSY 的变化示意图 .....	434
图 20-6 主机单向只发送模式下连续传输时, SPI_STS.TE/BUSY 变化示意图 .....	435
图 20-7 只接收模式 (BIDIRMODE=0 并且 RONLY=1) 下连续传输时, RNE 变化示意图 .....	436
图 20-8 从机全双工模式下连续传输时, SPI_STS.TE/RNE/BUSY 的变化示意图 .....	437
图 20-9 从机单向只发送模式下连续传输时, SPI_STS.TE/BUSY 变化示意图 .....	437
图 20-10 BIDIRMODE = 0, RONLY = 0 非连续传输发送时, SPI_STS.TE/BUSY 变化示意图 .....	439
图 20-11 使用 DMA 发送.....	441
图 20-12 使用 DMA 接收.....	441
图 21-1 级别和 Cortex®-M0 级别的调试框图 .....	453

## 1 文中的缩写

### 1.1 寄存器描述表中使用的缩写列表

在对寄存器的描述中使用了下列缩写：

read/write(rw)	软件能读写此位。
read-only(r)	软件只能读此位。
write-only(w)	软件只能写此位，读此位将返回复位值。
read/clear(rc_w1)	软件可以读此位，也可以通过写‘1’清除此位，写‘0’对此位无影响。
read/clear(rc_w0)	软件可以读此位，也可以通过写‘0’清除此位，写‘1’对此位无影响。
read/clear by read(rc_r)	软件可以读此位，读此位将自动地清除它为‘0’，写‘0’对此位无影响。
read/set(rs)	软件可以读也可以设置此位，写‘0’对此位无影响。
read-only write trigger(rt_w)	软件可以读此位，写‘0’或‘1’触发一个事件但对此位数值没有影响。
toggle(t)	软件只能通过写‘1’来翻转此位，写‘0’对此位无影响。
Reserved(Res.)	保留位，必须保持默认值不变。

### 1.2 可用的外设

有关 N32G033 系列全部型号，某外设存在与否及其数目，请查阅相应型号的数据手册。

## 2 总线架构

### 2.1 系统架构

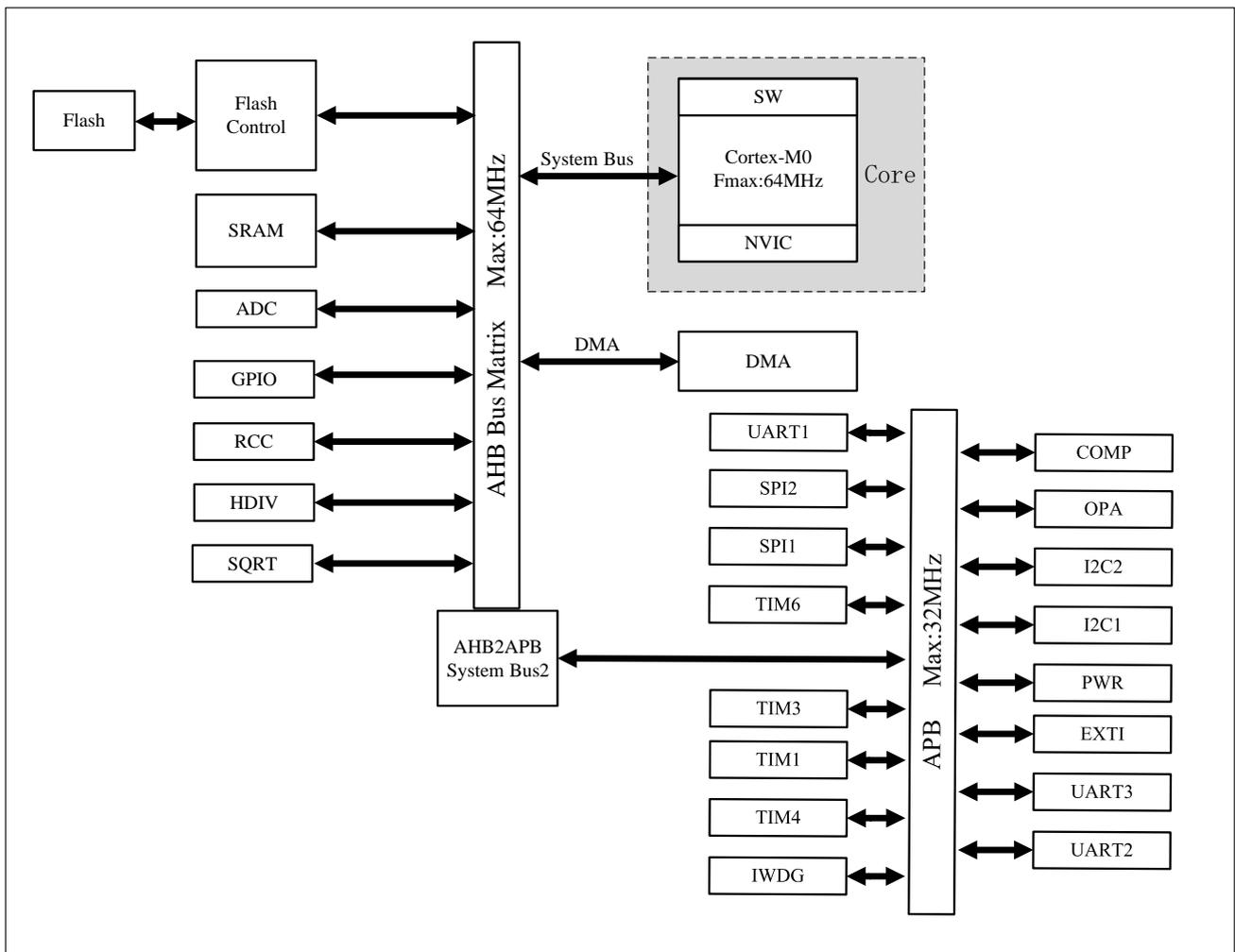
#### 2.1.1 总线架构

主系统由以下部分构成：

- 两个主驱动单元：
  - ◆ Cortex<sup>®</sup>-M0 内核系统总线
  - ◆ 通用 DMA
- 5 个被动单元
  - ◆ 内部 SRAM
  - ◆ 内部闪存存储器
  - ◆ ADC
  - ◆ AHB 到 AHB 的桥，它连接一些 AHB 设备
  - ◆ AHB 到 APB 的桥(AHB2APBx)，它连接所有的 APB 设备

这些都是通过一个多级的 AHB 总线构架相互连接的，如图 2-1 所示：

图 2-1 总线架构图



- CPU 系统总线：连接 Cortex®-M0 内核的 Sbus 总线到总线矩阵，用来指令预取，数据加载（常量加载和调试访问）及 AHB/APB 外设访问。
- DMA 总线：DMA 的 AHB 主控接口连接到总线矩阵，总线矩阵协调着内核和 DMA 到 SRAM、闪存和外设的访问。
- 总线矩阵协调内核系统总线和 DMA 主控总线之间的访问仲裁，仲裁利用轮算法。
- 系统包含 1 个 AHB2APB 桥，包含 17 个 APB 外设，PCLK 的最高速度为 32MHz。

### 2.1.2 总线地址映射

总线地址映射包括所有 AHB 和 APB 外设、Flash、SRAM、SystemMemory 等。具体映射如下

图 2-2 总线地址映射图

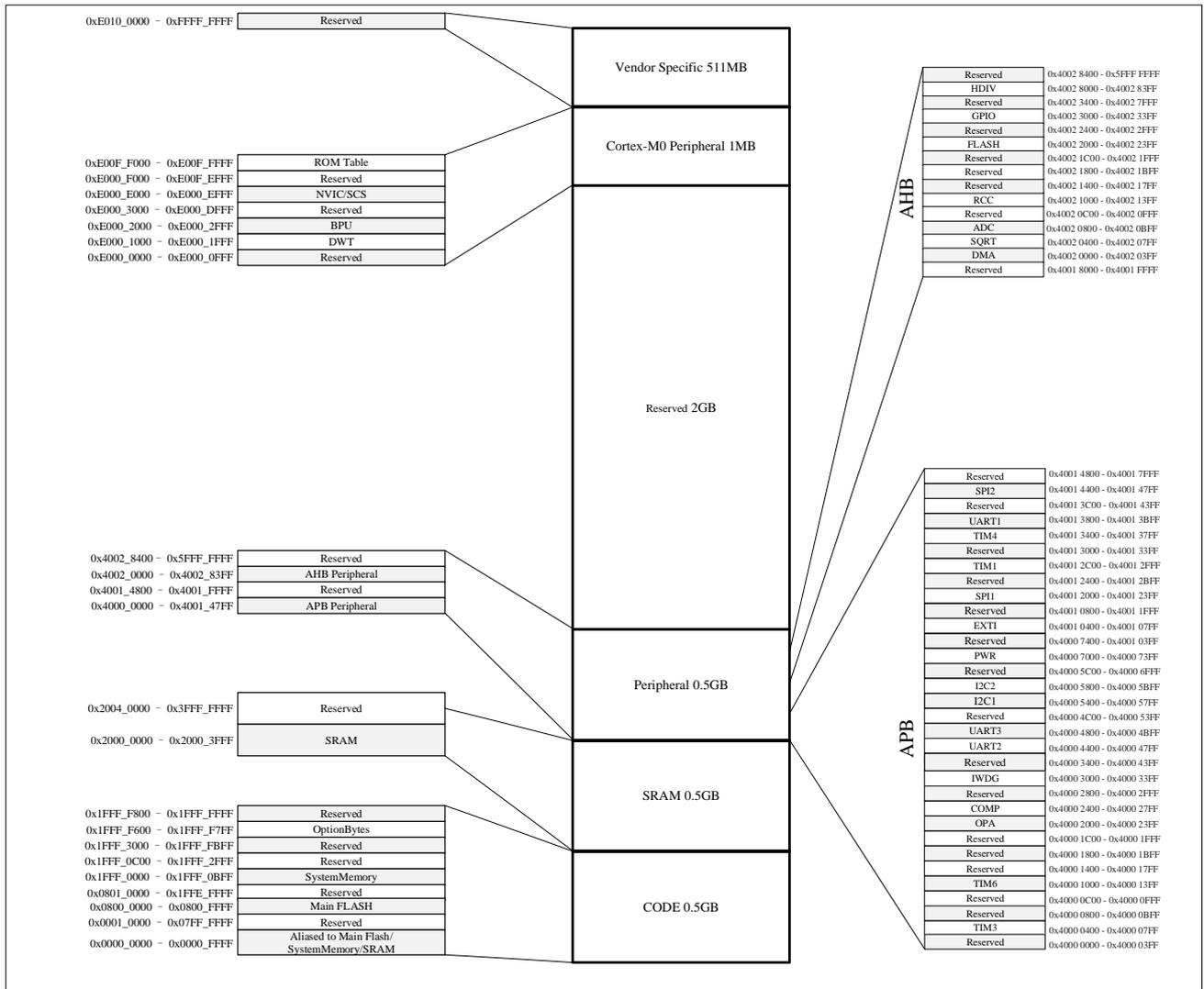


表 2-1 外设寄存器地址列表

地址范围	外设	总线
0x4002 8000 - 0x5FFF FFFF	Reserved	AHB
0x4002 8000 - 0x4002 83FF	HDIV	
0x4002 3400 - 0x4002 7FFF	Reserved	
0x4002 3000 - 0x4002 33FF	IOM	
0x4002 2400 - 0x4002 2FFF	Reserved	
0x4002 2000 - 0x4002 23FF	FLASH	
0x4002 1400 - 0x4002 1FFF	Reserved	
0x4002 1000 - 0x4002 13FF	RCC	
0x4002 0C00 - 0x4002 0FFF	Reserved	
0x4002 0800 - 0x4002 0BFF	ADC	
0x4002 0400 - 0x4002 07FF	SQRT	
0x4002 0000 - 0x4002 03FF	DMA	

地址范围	外设	总线
0x4001 8000 - 0x4001 FFFF	Reserved	
0x4000 0000 - 0x4000 03FF	Reserved	APB
0x4000 0400 - 0x4000 07FF	TIM3	
0x4000 0800 - 0x4000 0BFF	Reserved	
0x4000 0C00 - 0x4000 0FFF	Reserved	
0x4000 1000 - 0x4000 13FF	TIM6	
0x4000 1400 - 0x4000 17FF	Reserved	
0x4000 1800 - 0x4000 1BFF	Reserved	
0x4000 1C00 - 0x4000 1FFF	Reserved	
0x4000 2000 - 0x4000 23FF	OPA	
0x4000 2400 - 0x4000 27FF	COMP	
0x4000 2800 - 0x4000 2FFF	Reserved	
0x4000 3000 - 0x4000 33FF	IWDG	
0x4000 3400 - 0x4000 43FF	Reserved	
0x4000 4400 - 0x4000 47FF	UART2	
0x4000 4800 - 0x4000 4BFF	UART3	
0x4000 4C00 - 0x4000 53FF	Reserved	
0x4000 5400 - 0x4000 57FF	I2C1	
0x4000 5800 - 0x4000 5BFF	I2C2	
0x4000 5C00 - 0x4000 6FFF	Reserved	
0x4000 7000 - 0x4000 73FF	PWR	
0x4000 7400 - 0x4001 03FF	Reserved	
0x4001 0400 - 0x4001 07FF	EXTI	
0x4001 0800 - 0x4001 1FFF	Reserved	
0x4001 2000 - 0x4001 23FF	SPI1	
0x4001 2400 - 0x4001 2BFF	Reserved	
0x4001 2C00 - 0x4001 2FFF	TIM1	
0x4001 3000 - 0x4001 33FF	Reserved	
0x4001 3400 - 0x4001 37FF	TIM4	
0x4001 3800 - 0x4001 3BFF	UART1	
0x4001 3C00 - 0x4001 43FF	Reserved	
0x4001 4400 - 0x4001 47FF	SPI2	
0x4001 4800 - 0x4001 7FFF	Reserved	

## 2.1.3 启动管理

### 2.1.3.1 启动地址

在系统启动时，可以通过 BOOT0 引脚和用户选项字节 BOOT 配置，来选择在复位后的启动模式，在系统复

位后或从掉电模式退出时，BOOT 引脚的值将被重新锁存。经过启动延迟之后，CPU 从地址 0x0000\_0000 获取堆栈顶的地址，并从地址 0x0000\_0004 指示的复位向量地址开始执行代码。由于 Cortex®-M0 始终从地址 0x0000\_0000 和 0x0000\_0004 获取堆栈顶指针和复位向量，所以启动仅适合于从 CODE 代码区开始，设计上需要对启动空间进行地址重映射。有三种启动模式可选：

- 从主闪存存储器(Main Flash)启动：
  - ◆ 主闪存存储器被映射到启动空间（0x0000\_0000）；
  - ◆ 主闪存存储器可在两个地址区域访问，0x0000\_0000 或 0x0800\_0000；
- 从系统存储器(System Memory)启动：
  - ◆ 系统存储器被映射到启动空间（0x0000\_0000）；
  - ◆ 系统存储器可在两个地址区域访问，0x0000\_0000 或 0x1FFF\_0000；
- 从内置 SRAM 启动：
  - ◆ 内置 SRAM 被映射到启动空间（0x0000\_0000）；
  - ◆ 内置 SRAM 可在两个地址区域访问，0x0000\_0000 或 0x2000\_0000；

### 2.1.3.2 启动配置

可以通过 BOOT0 引脚和用户选项字节 BOOT 配置选择三种不同启动模式

表 2-2 启动模式列表

启动模式选择引脚					启动模式	对应启动模式下，访问内存空间的起始地址		
nBOOT1	nBOOT0	BOOT0 引脚	nSWBOOT0	BOOT0_CFG		Main Flash	System Memory	SRAM
X	X	0	1	1	Main Flash 启动	0x0000_0000	0x1FFF_0000	0x2000_0000
X	1	X	0			0x0800_0000		
1	X	1	1		System Memory 启动	0x08000000	0x0000_0000	0x2000_0000
1	0	X	0			0x1FFF_0000		
0	X	1	1		SRAM 启动	0x08000000	0x1FFF_0000	0x0000_0000
0	0	X	0			0x2000_0000		
X	X	1	1	0	Main Flash 启动	0x0000_0000	0x1FFF_0000	0x2000_0000
X	1	X	0			0x0800_0000		
1	X	0	1		System Memory 启动	0x08000000	0x0000_0000	0x2000_0000
1	0	X	0			0x1FFF_0000		
0	X	0	1		SRAM 启动	0x08000000	0x1FFF_0000	0x0000_0000
0	0	X	0			0x2000_0000		

注：其中 BOOT0 和 GPIO 复用，上电默认状态受控于 BOOT0\_CFG 值。

### 2.1.3.3 内嵌启动程序

内嵌的自举程序存放在系统存储器 System Memory 内，用于通过 UART1 对闪存存储器进行重新编程。而 UART1 接口可以依靠内部 64MHz 振荡器（HSI）运行。进一步的细节请查询自举程序手册。

## 2.2 存储系统（Memory system）

程序存储器、数据存储器、寄存器和输入输出端口被组织在同一个 4GB 的线性地址空间内。数据字节以小端格式存放在存储器中，一个字里的最低地址字节被认为是该字的最低有效字节，而最高地址字节是最高有效字节。对程序存储器和数据存储器的规格说明如下。

### 2.2.1 FLASH 规格

Flash 由主存储区、信息区组成，以下分别进行说明：

- 主存储区最大为 64KB，也称作主闪存存储器，包含 128 个 Page，用于用户程序的存放和运行，以及数据存储。
- 系统信息区为 5KB，包含 10 个 Page，由系统存储区（3KB）、系统配置区（1.5KB）、选项字节区（0.5KB）组成：
  - ◆ 系统存储区为 3KB，包含 6 个 Page，也称作 System Memory，用于引导程序（BOOT）的存放和运行。
  - ◆ 系统配置区为 1.5KB，包含 3 个 Page。
  - ◆ 选项字节区为 0.5KB，包含 1 个 Page，也称作 OptionByte，有效空间为 26B，BOOT 程序、用户程序均可以读写擦。

#### 2.2.1.1 存储地址

主存储区、信息区都分配了总线地址空间。

表 2-3 存储总线地址列表

存储区	页名称	地址范围	大小
主存储区	页 0	0x0800_0000 – 0x0800_01FF	0.5KB
	页 1	0x0800_0200 – 0x0800_03FF	0.5KB
	页 2	0x0800_0400 – 0x0800_05FF	0.5KB
	⋮	⋮	⋮
	页 127	0x0800_FE00 – 0x0800_FFFF	0.5KB
信息区	系统存储区	0x1FFF_0000 – 0x1FFF_0BFF	3KB
	系统配置区	0x1FFF_F000 – 0x1FFF_F5FF	1.5KB
	选项字节区	0x1FFF_F600 – 0x1FFF_F619	26B
存储区接口 寄存器	FLASH_AC	0x4002_2000 – 0x4002_2003	4B
	FLASH_KEY	0x4002_2004 – 0x4002_2007	4B
	FLASH_OPTKEY	0x4002_2008 – 0x4002_200B	4B

存储区	页名称	地址范围	大小
	FLASH_STS	0x4002_200C – 0x4002_200F	4B
	FLASH_CTRL	0x4002_2010 – 0x4002_2013	4B
	FLASH_ADD	0x4002_2014 – 0x4002_2017	4B
	FLASH_OB2	0x4002_2018 – 0x4002_201B	4B
	FLASH_OB	0x4002_201C – 0x4002_201F	4B
	FLASH_WRP	0x4002_2020 – 0x4002_2023	4B
	保留	0x4002_2024 – 0x4002_204F	44B
	FLASH_VTOR	0x4002_2050 – 0x4002_2053	4B

闪存存储器被组织成 64 位宽的存储器单元，可以存放代码和数据常数。

信息区分为三个部分：

- 系统存储区是用于存放在系统存储器自举模式下的启动程序，启动程序使用 UART1 接口实现对闪存存储器的编程。
- 系统配置区，包含芯片基本信息。
- 选项字节区。

对主存储器和信息块的写入由内嵌的闪存编程/擦除控制器管理。

主闪存存储器（main flash）有两种保护方式防止非法的访问（读、写、擦除）：

- 页写入保护（WRP）
- 读出保护（RDP）

在执行 FLASH 写操作时，任何对 FLASH 的读操作都会锁住总线，在写操作完成后读操作才能正确地进行；即在进行写或擦除操作时，不能进行代码或数据的读取操作。

进行闪存编程操作时（写或擦除），必须打开内部的 RC 振荡器（HSI）。

*注：在低功耗模式下，所有闪存存储器的操作都被中止。*

### 2.2.1.2 读写操作

Flash 写操作仅支持 32 位操作，写操作之前先擦除 Flash，擦除最小块大小是一个页 0.5KB。写操作分为擦除和编程阶段。

读 Flash 时，读的等待周期数可以通过寄存器配置。使用时，需要结合 SYSCLK 时钟频率进行计算。比如：当  $SYSCLK \leq 32MHz$  时，等待周期数最小为 0；当  $32MHz < SYSCLK \leq 64MHz$  时，等待周期数最小为 1；

### 2.2.1.3 Flash 解锁操作

复位后，Flash 模块是被保护的，不能写入 FLASH\_CTRL 寄存器，以防因电气干扰等原因产生对 Flash 的意外操作。通过写入特定的键值序列到 FLASH\_KEY 寄存器，可以开启对 FLASH\_CTRL 寄存器的操作权限，这个特定的序列是：第一次在 Flash 密钥寄存器（FLASH\_KEY）中写入  $KEY1 = 0x45670123$ ，第二次则在 Flash 密钥寄存器（FLASH\_KEY）中写入  $KEY2 = 0xCDEF89AB$ 。

如果顺序出现错误或键值出现错误，将返回总线错误并锁定 FLASH\_CTRL 寄存器，直到下一次复位，软件可以通过查看 FLASH\_CTRL.LOCK 位来确认 Flash 是否已解锁。若需要进行正常的锁定设置，可以通过软件将 FLASH\_CTRL.LOCK 位置 1 来实现，此后可以通过在 FLASH\_KEY 中写入正确的键值系列来对 Flash 解锁。

## 2.2.1.4 擦除和编程

### 2.2.1.4.1 主存储区擦除

主存储区可以按页擦除或者整片擦除

#### 页擦除

页擦除流程:

- 通过检查 FLASH\_STS.BUSY 位来确保没有正在进行闪存操作;
- 设置 FLASH\_CTRL.PER 为'1';
- 将要擦除的页起始地址写入 FLASH\_ADD 寄存器;
- 设置 FLASH\_CTRL.START 为'1';
- 等待 FLASH\_STS.BUSY 变为'0';
- 读出被擦除页的内容检查是否被擦除。

#### 片擦除

片擦除流程:

- 通过检查 FLASH\_STS.BUSY 位来确保没有正在进行闪存操作;
- 设置 FLASH\_CTRL.MER 为'1';
- 设置 FLASH\_CTRL.START 为'1';
- 等待 FLASH\_STS.BUSY 位变为'0';
- 读出所有被擦除页的内容检查是否被擦除。

### 2.2.1.4.2 主存储区编程

对主存储区编程每次可以写入 32 位。当 FLASH\_CTRL.PG 为'1'时, 在一个闪存地址写入一个字将启动一次编程; 写入任何半字的数据, 都会产生总线错误。在编程过程中(FLASH\_STS.BUSY 为'1'), 任何读写闪存的操作都会使 CPU 暂停, 直到此次闪存编程结束。

主存储区编程流程:

- 通过检查 FLASH\_STS.BUSY 位来确保没有正在进行闪存操作;
- 设置 FLASH\_CTRL.PG 为'1';
- 在指定的地址写入要编程的字;
- 等待 FLASH\_STS.BUSY 变为'0';
- 读出写入地址的数据检查是否正确。

*注意: 当 FLASH\_STS.BUSY 为'1'时, 不能对任何 Flash 寄存器执行写操作。*

*注意: 在编程中向两个地址写入数据后, 需添加 3 个 \_\_NOP(), 防止总线读取 Flash 数据导致编程失败。*

### 2.2.1.4.3 选项字节区擦除和编程

对选项字节区的编程与主存储区不同。选项字节的数目只有 13 个字节(2 个字节作为写保护, 2 个字节作为读保护, 7 个字节为配置选项, 2 个字节存储用户数据)。对 Flash 解锁后, 必须分别写入 KEY1 和 KEY2(见

2.2.1.3)到 FLASH\_OPTKEY 寄存器,再设置 FLASH\_CTRL.OPTWE 为'1',此时可以对选项字节区进行编程:设置 FLASH\_CTRL.OPTPG 为'1'后写入字到指定的地址。

对选项字节区字编程时,使用半字中的低字节并自动地计算出高字节(高字节为低字节的补码),并开始编程操作,这将保证选项字节和它的补码始终是正确的。

选项字节区擦除过程:

- 通过检查 FLASH\_STS.BUSY 位来确保没有正在进行闪存操作;
- 解锁 FLASH\_CTRL.OPTWE;
- 设置 FLASH\_CTRL.OPTER 为'1';
- 设置 FLASH\_CTRL.START 为'1';
- 等待 FLASH\_STS.BUSY 变为'0';
- 读出被擦除选项字节的内容检查是否被擦除。

*注意: 选项字节擦除流程无需配置 FLASH\_ADD 寄存器。若在 FLASH\_ADD 寄存器写入非选项字节区的页起始地址,会产生 FLASH\_STS.WRPERR 错误。*

选项字节区编程流程:

- 通过检查 FLASH\_STS.BUSY 位来确保没有正在进行闪存操作;
- 解锁 FLASH\_CTRL.OPTWE;
- 设置 FLASH\_CTRL.OPTPG 为'1';
- 在指定的地址写入要编程的字;
- 等待 FLASH\_STS.BUSY 变为'0';
- 读出写入地址的数据检查是否正确。

### 2.2.1.5 选项字节

选项字节块主要用于配置读写保护、BOOT 模式配置、软件/硬件看门狗,并分配了总线地址空间,可以进行读写访问。它们由有 13 个选项字节组成: 2 个字节作为写保护, 2 个字节作为读保护, 7 个字节作为配置选项, 2 个字节由用户定义, 这 13 个字节需要通过总线写入。选项字节块同时还包含与这 13 个选项字节相对应的补码, 这些补码需要在总线写入选项字节时, 由硬件自动计算出来, 一起写入 Flash, 并用于选项字节读取时的验证。

默认状态下, 选项字节块始终是可以读且被写保护。要想对选项字节块进行写操作(编程/擦除), 首先要解锁 Flash, 然后解锁选项字节: 在 FLASH\_OPTKEY 中写入正确的键值序列 (KEY1 = 0x45670123, KEY2 = 0xCDEF89AB), 随后对选项字节块的写操作将被允许。如果顺序出现错误或键值出现错误, 将返回总线错误并锁定选项字节, 直到下一次复位。若需要正常进行锁定设置, 可以通过软件将 FLASH\_CTRL.OPTWE 位写 0 来实现, 此后可以通过在 FLASH\_OPTKEY 中写入正确的键值系列来对选项字节解锁。

每次系统复位后, 从 Flash 的选项字节块中读出选项字节数据, 并保存在具有只读属性的选项字节寄存器 (FLASH\_OB/FLASH\_OB2/FLASH\_WRP) 中; 同时一起读出来的选项字节补码数据, 将用于验证选项字节数据是否正确, 如果不匹配, 将产生一个选项字节错误标志 (FLASH\_OB.OBERR)。当发生选项字节错误时, 对应的选项字节被强置为 0xFF。当选项字节和它的补码均为 0xFF 时(擦除后的状态), 则略过上述验证步骤, 无需进行验证。

表 2-4 选项字节列表

地址	[31:24] 相应反码	[23:16] 选项字节	[15:8] 相应反码	[7:0] 选项字节
0x1FFF_F600	nUSER4	USER4	nRDP1	RDP1
0x1FFF_F604	nUSER0[15:8]	USER0[15:8]	nUSER0[7:0]	USER0[7:0]
0x1FFF_F608	nUSER1[15:8]	USER1[15:8]	nUSER1[7:0]	USER1[7:0]
0x1FFF_F60C	nUSER3	USER3	nUSER2	USER2
0x1FFF_F610	nData1	Data1	nData0	Data0
0x1FFF_F614	nWRP1	WRP1	nWRP0	WRP0
0x1FFF_F618	-	-	nRDP2	RDP2

- 读保护 L1 等级：RDP1
  - ◆ 保护存储在闪存中的代码；
  - ◆ 当写入正确的是数值时，将禁止读出闪存存储器；
  - ◆ RDP1 是否开启的结果，可通过 FLASH\_OB[1]查询；
- 用户配置选项 0：USER0
  - ◆ USER0[15:0]，IWDG\_SW 配置选项，可通过 FLASH\_OB[2]查询  
 USER0 = 16'h5aa5：硬件看门狗  
 USER0! = 16'h5aa5：软件看门狗
- 用户配置选项 1：USER1
  - ◆ USER1[15:0]，PF3\_NRST 配置选项，可通过 FLASH\_OB[3]查询  
 USER1 = 16'h5aa5：PF3 为普通 GPIO  
 USER1! = 16'h5aa5：PF3 为 NRST
- 用户配置选项 2：USER2
  - ◆ USER2，上电延时复位控制，可通过 FLASH\_OB[11:4]查询
- 用户配置选项 3：USER3
  - ◆ USER3[7:4]：保留
  - ◆ USER3[3]：BOOT0\_CFG 配置选项，可通过 FLASH\_OB[15]查询  
 0：BOOT0 Pin 拉低有效  
 1：BOOT0 Pin 拉高有效
  - ◆ USER3[2]：nSWBOOT0\_SEL 配置选项，可通过 FLASH\_OB[14]查询  
 0：nBOOT0 配置选型使用为 BOOT 模式选择  
 1：BOOT0 Pin 管脚使用为 BOOT 模式选择
  - ◆ USER3[1]：nBOOT1 配置选项，可通过 FLASH\_OB[13]查询
  - ◆ USER3[0]：nBOOT0 配置选项，可通过 FLASH\_OB[12]查询

- 用户配置选项 4: USER4
  - ◆ USER4[7:2]: 保留
  - ◆ USER4[1:0]: 用于选择 boot 的串口引脚, 可通过 FLASH\_OB [21:20]查询
- 2 字节用户数据: Datax
  - ◆ Data1 (FLASH\_OB2[15:8])
  - ◆ Data0 (FLASH\_OB2 [7:0])
- 写保护选项字节: WRP0 ~ 1, 可通过寄存器 FLASH\_WRP[15:0]查询
  - ◆ WRP0: 第 0~63 页的写保护, bit[0]对应 Page (0~7), ....., bit[7]对应 Page (56~63)
  - ◆ WRP1: 第 64~127 页的写保护, bit[0]对应 Page (64~71), ....., bit[7]对应 Page (120~127)
- 读保护 L2 等级: RDP2
  - ◆ 在 L1 的基础上增加保护功能, 具体见 2.2.1.7 读保护的详细描述;
  - ◆ RDP2 是否开启的结果, 可通过 FLASH\_OB[31]查询;

### 2.2.1.6 写保护

可以对 Flash 主存储区 (最大 64KB) 的所有 Page 配置写保护, 以防在程序跑飞或电气干扰等原因导致的意外写操作, 写保护的基本单位是: 对于 Page0~127, 每 8 页为一个基本保护单元。写保护可以通过设置选项字节块中的 WRP0~1 来进行配置; 每次进行配置后, 需要进行一次系统复位, 配置的值才能生效。如果对一个受保护的页面进行编程或擦除操作, FLASH\_STS 中将会返回一个保护错误标志。

系统信息区中的系统存储块 (3KB), 存放了 BOOT 程序, 不可更改。

系统信息区中的系统配置块 (1.5KB), 存放了芯片基本信息, 不可更改。

系统信息区中的选项字节块 (0.5KB), 存放了用户可配置选项字节信息, 将 FLASH\_CTRL.OPTWE 写 0 使能选项字节块的写保护, 之后通过在 FLASH\_OPTKEY 中写入正确的键值序列, 来对选项字节解除写保护。

### 2.2.1.7 读保护

Flash 中的用户代码可以通过设置读保护来防止被非法读取。读保护通过配置选项字节块中的 RDP 字节进行设置, 可以配置 3 种不同的读保护级别, 如下列表

表 2-5 读保护配置列表

读保护等级	RDP1	nRDP1	nRDP2	RDP2
L0 level	0xA5	0x5A	RDP2! = 0xCC    nRDP2! = 0x33	
L2 level	0xFF	0xFF	0x33	0xCC
L1 level	非以上两种配置			

- L0 等级:
  - ◆ 处于未保护状态, (RDP1 == 0xA5 & nRDP1 == 0x5A) && (RDP2! = 0xCC | nRDP2! = 0x33)
  - ◆ 主存储区和选项字节可以被任意读取
  - ◆ 主存储区和选项字节可以进行编程和擦除, 可配置读写保护
- L1 等级:

- ◆  $\sim ((RDP1 == 0xA5 \& nRDP1 == 0x5A) \&\& (RDP2 != 0xCC \mid nRDP2 != 0x33)) \mid (RDP2 == 0xCC \& nRDP2 == 0x33))$
- ◆ 允许从用户代码/SRAM 代码中对主存储区的读操作
- ◆ 全部主存储区页可以通过在主闪存存储器中执行的代码进行编程（实现 IAP 或数据存储等功能）
- ◆ 全部主存储区页不允许在调试模式下执行写或擦除操作
- ◆ 不能通过 SWD 向内置 SRAM 装载代码
- ◆ 当读保护的选项字节被改写为未保护的 L0 级别时，将会自动擦除全部主存储区，执行的过程如下：  
（擦除选项字节块不会导致自动的擦除操作，因为擦除的结果是 0xFF，相当于仍然处于 L1 级别的保护状态）
  - 在 FLASH\_OPTKEY 中写入正确的键值序列解锁选项字节区；
  - 总线发起命令擦除整个选项字节区（Page 擦）；
  - 总线写入读保护选项字节 0xA5；
  - 内部自动擦除全部主存储区；
  - 内部自动写入 0xA5 到读保护选项字节；
  - 进行系统复位（如软件复位等），选项字节块（包括新的 RDP 值 0xA5）将被重新加载到系统中，读保护被解除；
- L2 等级：除了 SRAM 启动被禁止、调试模式被禁止、选项字节写/页擦被禁止、保护级别不可修改（不可逆）之外，其余特性同 L1 级别。L2 级别通过配置另一个选项字节 RDP2 来实现，不管 RDP1 为何值，只要满足  $(RDP2=0xCC \& nRDP2=0x33)$  即为 L2 级别

### 2.2.1.8 权限保护

- Flash 主存储区权限：
  - ◆ L0 级别下：主存储区都可以被读取；主存储区可以配置各 Page 的写保护属性；
  - ◆ L1/2 级别下：
    - 在 SWD 调试模式下或从内部 SRAM 启动后执行时，所有 Page 不允许（W/R/PE）操作；
    - 所有 Page 可以通过在 Flash 主存储器区中执行的代码进行编程（实现 IAP 或数据存储等功能）；
    - 主存储区所有 Page 可以配置各 Page 的写保护属性；
    - 当 L1 级别改写为 L0 级别时，将会自动擦除全部 Flash 主存储区；
- Flash 选项字节区权限：
  - ◆ L0/L1 级别下，都被允许访问（W/R/PE）；
  - ◆ L2 级别下：除了调试模式被禁止外，都允许只读访问 Flash 选项字节区；
- Flash 系统存储区权限：
  - ◆ System memory 区/用户程序区/Sram 区执行的代码允许读访问（R）；
  - ◆ 通过调试接口不允许访问；

- ◆ 在 L1/L2 级别下，通过调试接口不允许访问；
- ◆ 在 L1/L2 级别下，sram 启动跳转到 system memory 执行代码不允许访问（W/R/PE）。其他情况允许访问（W/R/PE）；

■ Flash 系统配置区

- ◆ 只读

表 2-6 存储区读写擦<sup>(1)</sup>权限控制表

保护级别	启动模式	System Memory/Main Flash		修改保护级别
	执行用户 访问区域	SWD	System Memory/ Main Flash/SRAM	
L0 级别	Flash 主存储区	读写擦	读写擦	允许改为 L1 或 L2
	Flash 主存储区片擦 <sup>(2)</sup>	允许	允许	
	Flash 选项字节区	读写擦	读写擦	
	Flash 系统存储区	禁止	只读	
	系统配置区	只读	只读	
	SRAM (All)	读写	读写	
L1 级别	Flash 主存储区	禁止	读写擦	允许改为 L0 或 L2。 改为 L0 时，主存储区将被自动擦除。
	Flash 主存储区片擦 <sup>(2)</sup>	禁止	允许	
	Flash 选项字节区	读写擦	读写擦	
	Flash 系统存储区	禁止	只读	
	系统配置区	只读	只读	
	SRAM (All)	只读	读写	
L2 级别	Flash 主存储区	SWD 接口被禁止	读写擦	不允许修改。
	Flash 主存储区片擦 <sup>(2)</sup>		允许	
	Flash 选项字节区		只读	

	Flash 系统存储区		只读	
	系统配置区		只读	
	SRAM (All)		读写	
保护级别	启动模式	SRAM		修改保护级别
	执行用户 访问区域	SWD	System Memory/ Main Flash/SRAM	
L0 级别	Flash 主存储区	读写擦	读写擦	允许改为 L1 或 L2
	Flash 主存储区片擦 (2)	允许	允许	
	Flash 选项字节区	读写擦	读写擦	
	Flash 系统存储区	只读/禁止	只读	
	系统配置区	只读	只读	
	SRAM (All)	读写	读写	
L1 级别	Flash 主存储区	禁止	读写擦	允许改为 L0 或 L2。 改为 L0 时，主存储区将被自动擦除。
	Flash 主存储区片擦 (2)	禁止	允许	
	Flash 选项字节区	读写擦	读写擦	
	Flash 系统存储区	禁止	禁止	
	系统配置区	只读	只读	
	SRAM (All)	只读	读写	
L2 级别	Flash 主存储区	L2 保护级别，无法从 SRAM 启动		不允许修改。 SWD 被禁止。
	Flash 主存储区片擦 (2)			
	Flash 选项字节区			

	Flash 系统存储区		
	系统配置区		
	SRAM (All)		

注：1.这里的擦是指 Flash 页擦除；

2.Flash 主存储区片擦除是指 mass erase，仅在 L2 级别调试模式下禁止；。

## 2.2.2 SRAM

SRAM 主要用于代码运行，存放程序执行过程中的变量和数据或堆栈，容量最大为 6KB。

SRAM 支持字节、半字、字的读写访问。

SRAM 支持代码运行，可以在 SRAM 全速运行程序。SRAM 的最大地址范围是 0x2000 0000~0x2000 17FF。

SRAM 在 Run/Stop 工作模式数据可以正常保持。

主要特性如下：

- 容量最大总共为 6KB
- 支持字节/半字/字读写
- CPU/DMA 均可访问
- CPU BUS 可以 Remap 到 SRAM 全速运行程序

## 2.2.3 FLASH 寄存器描述

必须以字（32 位）的方式操作寄存器。

### 2.2.3.1 FLASH 寄存器总览

表 2-7 FLASH 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	FLASH_AC	Reserved															LATENCY																
0x04	FLASH_KEY	FKEY																															
0x08	FLASH_OPTKEY	OPTKEY																															
0x0c	FLASH_STATUS	Reserved															EOP	WRPERR	Reserved	PGERR	Reserved	BUSY											
0x10	FLASH_CTL	Reserved															EOPTIE	Reserved	ERRITE	OPTWE	Reserved	LOCK	START	OPTER	OPTPG	MER	PER	PG					
0x14	FLASH_ADDR	FADD																															
0x18	FLASH_OBS2	Reserved															Data1						Data0										

0x1c	FLASH_OB	RDPRT2	Reserved	BOOT_SEL	Reserved	BOOT0_CF	nSWBOOT0	nBOOT1	nBOOT0	POR_DELAY	NRST_PF3	IWDG_SW	RDPRT1	OBERR	
0x20	FLASH_W RP	Reserved					WRP								
0x50	FLASH_VT OR	VTOR_EN	VTOR_VALUE												

### 2.2.3.2 FLASH 访问控制寄存器 (FLASH\_AC)

偏移地址: 0x00

复位值: 0x0000 0002

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

位域	名称	描述
31:2	Reserved	保留, 必须保持复位值
1:0	LATENCY	时延 这些位表示 SYSCLK (系统时钟) 周期与闪存访问时间的比例 00: 零周期时延, 当 $0 < \text{SYSCLK} \leq 32\text{MHz}$ 01: 一个周期时延, 当 $32\text{MHz} < \text{SYSCLK} \leq 64\text{MHz}$ 10/11: 保留

### 2.2.3.3 FLASH 键寄存器 (FLASH\_KEY)

偏移地址: 0x04

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FKEY[31:16]															
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
FKEY[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位域	名称	描述
31:0	FKEY	用于解锁 FLASH_CTRL.LOCK 位

### 2.2.3.4 FLASH OPTKEY 寄存器 (FLASH\_OPTKEY)

偏移地址: 0x08

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OPTKEY [31:16]															
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
OPTKEY [15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位域	名称	描述
31:0	OPTKEY	用于解锁 FLASH_CTRL.OPTWE 位

### 2.2.3.5 FLASH 状态寄存器 (FLASH\_STS)

偏移地址: 0x0C

复位值: 0x0000 0000

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

位域	名称	描述
31:6	Reserved	保留, 必须保持复位值
5	EOP	操作结束 当闪存操作(编程/擦除)完成时, 硬件设置这位为'1', 写入'1'可以清除这位状态。 <i>注: 每次成功的编程或擦除都会设置 EOP 状态。</i>
4	WRPERR	写保护错误 试图对写保护的闪存地址编程时, 硬件设置这位为'1', 写入'1'可以清除这位状态。
3	Reserved	保留, 必须保持复位值
2	PGERR	编程错误 试图对内容不是'0xFFFF_FFFF'的地址编程时, 硬件设置这位为'1', 写入'1'

位域	名称	描述
		可以清除这位状态。 <i>注：进行编程操作之前，必须先清除 FLASH_CTRL.START 位。</i>
1	Reserved	保留，必须保持复位值
0	BUSY	忙 该位指示闪存操作正在进行。在闪存操作开始时，该位被设置为'1'；在操作结束或发生错误时该位被清除为'0'。

### 2.2.3.6 FLASH 控制寄存器 (FLASH\_CTRL)

偏移地址：0x10

复位值：0x0000 0080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		EOPITE	Reserved	ERRITE	OPTWE	Reserved	LOCK	START	OPTER	OPTPG	Reserved	MER	PER	PG	
		rw		rw	rw		rw	rw	rw	rw		rw	rw	rw	

位域	名称	描述
31:13	Reserved	保留，必须保持复位值
12	EOPITE	允许操作完成中断 该位允许在 FLASH_STS.EOP 位变为'1'时产生中断。 0：禁止产生中断 1：允许产生中断
11	Reserved	保留，必须保持复位值
10	ERRITE	允许错误状态中断 该位允许在发生 Flash 错误时产生中断（当 FLASH_STS.PGERR/WRPERR 置为'1'时）。 0：禁止产生中断 1：允许产生中断
9	OPTWE	允许写选项字节 当该位为'1'时，允许对选项字节进行编程操作。当在 FLASH_OPTKEY 寄存器写入正确的键序列后，该位被置为'1'。 软件可清除此位。
8	Reserved	保留，必须保持复位值
7	LOCK	锁定 只能写'1'。当该位为'1'时表示 Flash 和 FLASH_CTRL 被锁住。在检测到正确的解锁序列后，硬件清除此位为'0'。 在一次不成功的解锁操作后，下次系统复位前，该位不能再被改变。
6	START	开始 当该位为'1'时将触发一次擦除操作。该位只可由软件置为'1'并在

位域	名称	描述
		FLASH_STS.BUSY 变为'1'时清除为'0'。
5	OPTER	擦除选项字节 0: 不开启选项字节擦除模式 1: 开启选项字节擦除模式
4	OPTPG	编程选项字节 0: 不开启选项字节编程模式 1: 开启选项字节编程模式
3	Reserved	保留, 必须保持复位值
2	MER	片擦除 0: 不开启片擦除模式 1: 开启片擦除模式
1	PER	页擦除 0: 不开启页擦除模式 1: 开启页擦除模式
0	PG	编程 0: 不开启编程模式 1: 开启编程模式

注: 关于编程及擦除请参考 2.2.1.4 节。

### 2.2.3.7 FLASH 地址寄存器 (FLASH\_ADD)

偏移地址: 0x14

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FADD[31:16]															
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
FADD[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位域	名称	描述
31:0	FADD	闪存地址 当进行编程时选择要编程的地址, 当进行页擦除时选择要擦除的页。 <i>注意: 当FLASH_STS.BUSY 位为'1'时, 不能写这个寄存器。</i>

### 2.2.3.8 FLASH 选项字节 2 寄存器 (FLASH\_OB2)

偏移地址: 0x18

复位值: 0x0000 FFFF

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

Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data1							Data0								
r	r	r	r	r	r			r	r	r	r	r	r	r	r

位域	名称	描述
31:16	Reserved	保留, 必须保持复位值
15:8	Data1[7:0]	Data1 <i>注: 只读位。</i>
7:0	Data0[7:0]	Data0 <i>注: 只读位。</i>

### 2.2.3.9 FLASH 选项字节寄存器 (FLASH\_OB)

偏移地址: 0x1C

复位值: 0x0FFF FFFC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDPRT2	Reserved										BOOT_SEL[1:0]		Reserved		
r										r	r				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOOT0_CFG	nSWBOOT0	nBOOT1	nBOOT0	POR_DELAY[7:0]								NRST_PF3	IWDG_SW	RDPRT1	OBERR
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

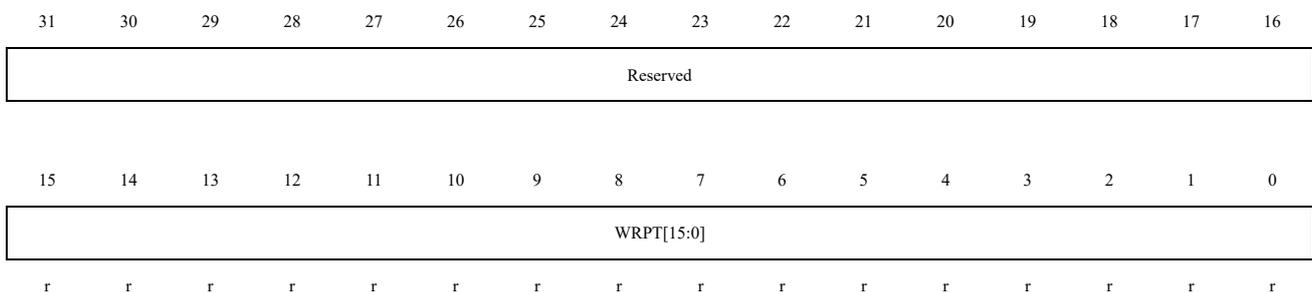
位域	名称	描述
31	RDPRT2	读保护 L2 级别 0: 读保护 L2 级别未使能 1: 读保护 L2 级别使能 <i>注: 只读位。</i>
30:22	Reserved	保留, 必须保持复位值
21:20	BOOT_SEL[1:0]	BOOT 串口通信引脚选择 11b: PA9, PA10, 默认 00b: PA13, PA14 01b: PF0, PF1 10b: PA2, PA3
19:16	Reserved	未使用, 必须保持复位值
15	BOOT0_CFG	BOOT0 上电默认状态 0: 默认高电平, 拉低有效 1: 默认低电平, 拉高有效
14	nSWBOOT0	使用规则见 2.1.3.2 启动配置章节。

位域	名称	描述
13	nBOOT1	使用规则见 2.1.3.2 启动配置章节。
12	nBOOT0	使用规则见 2.1.3.2 启动配置章节。
11:4	POR_DELAY[7:0]	触发 POR 后，CPU 复位的延迟时间。 在系统初始上电完成后，Cortex®-M0 的系统解复位延时长可通过该位配置，用来控制内核的复位延迟时间。 0x00：最大延迟时间 .... 0xFF：无延迟 延迟时间 = (1/f <sub>LST</sub> ) × (0xFF - POR_DELAY)。
3	NRST_PF3	PF3 引脚配置。 0：普通 IO 引脚 1：NRST 引脚 <i>注：只读位。</i>
2	IWDG_SW	看门狗设置 0：硬件看门狗 1：软件看门狗 <i>注：只读位。</i>
1	RDPRT1	读保护 L1 级别 0：读保护 L1 级别未使能 1：读保护 L1 级别使能 <i>注：只读位。</i>
0	OBERR	选项字节错误 当该位为'1'时表示选项字节和它的补码不匹配 <i>注：只读位。</i>

### 2.2.3.10 FLASH 写保护寄存器 (FLASH\_WRP)

偏移地址：0x20

复位值：0xFFFF FFFF



位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:0	WRPT	写保护 该寄存器包含由选项字节区加载的写保护选项字节。 0：写保护生效；

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
		1：写保护失效。 注：只读位。

### 2.2.3.11 FLASH VTOR 寄存器 (FLASH\_VTOR)

偏移地址：0x50

复位值：0x0000 0000

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

位域	名称	描述
31	VTOR_EN	中断向量重映射使能 1：使能 0：禁用
30:0	VTOR_VALUE	用于中断向量重映射，存放中断向量表首地址 这些位在 VTOR_EN = 1 时有效。 中断地址 = VTOR_VALUE + 偏移地址。 注：偏移地址小于 0x100 时，该功能才有效。

## 3 电源控制（PWR）

### 3.1 通用描述

PWR 是电源管理单元，用于控制 MCU 进入不同的电源模式，并在事件或中断发生时唤醒。MCU 支持 RUN 和 STOP 模式。PWR 控制不同电源模式下的电压调节器、时钟源、复位和 Flash/SRAM/GPIO 状态。

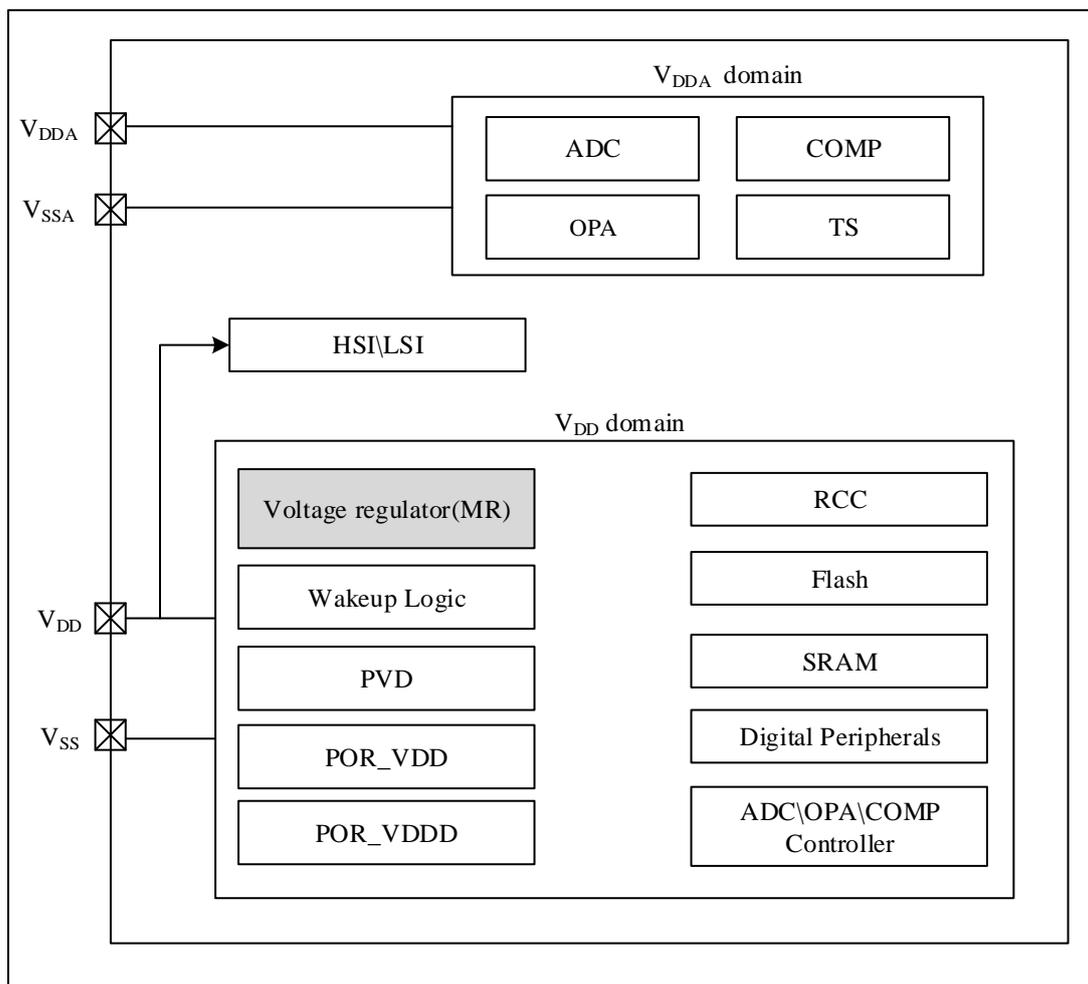
#### 3.1.1 电源

- ◇ MCU 由外部 VDD 供电。嵌入式稳压器用于为内部 1.5V 数字电源供电。稳压器有两种模式，正常模式和低功耗模式。
  - VDD 区域：2.0V~5.5V，主要为 MR、IO 及时钟复位系统提供电源输入。
  - VDDA 区域：2.0V~5.5V，为大部分模拟外设供电，详细信息请参阅相关数据手册电气特性部分。
  - VDDA 和 VSSA 必须分别连接到 VDD 和 VSS。
- ◇ 电压调节器根据应用有几种不同的工作模式：
  - RUN 模式：电压调节器以正常电源模式供电。
  - STOP 模式：电压调节器在低功耗模式下供电，输出电压可通过软件配置为 1.5V 或 1.2V。

表 3-1 电源域说明

电源域	工作模式	电压(v)	被驱动模块	备注
VDDA	External pin	2~5.5V	ADC, COMP, DAC, OPA	连接 PCB 上的 VDD
VDD	External pin	2~5.5V	POR_VDD/PDR_VDD, PVD, MR, LSI, HSI_LDO	-
VDDD	Low power	1.5	-	stop mode
	Active	1.5		run mode
VDDD_HSI	Active	1.5	HSI	-
BG1.2v	Active	1.2	内部基准	全温 2%的精度
BG3.6v	Active	3.6	ADC 参考值使用	全温 2%的精度
BG1.8v	Active	1.8	VBG(1.8V), 1/2AVDD, 1/4AVDD: OPA 的 VCM 使用	全温 2%的精度

图 3-1 电源框图

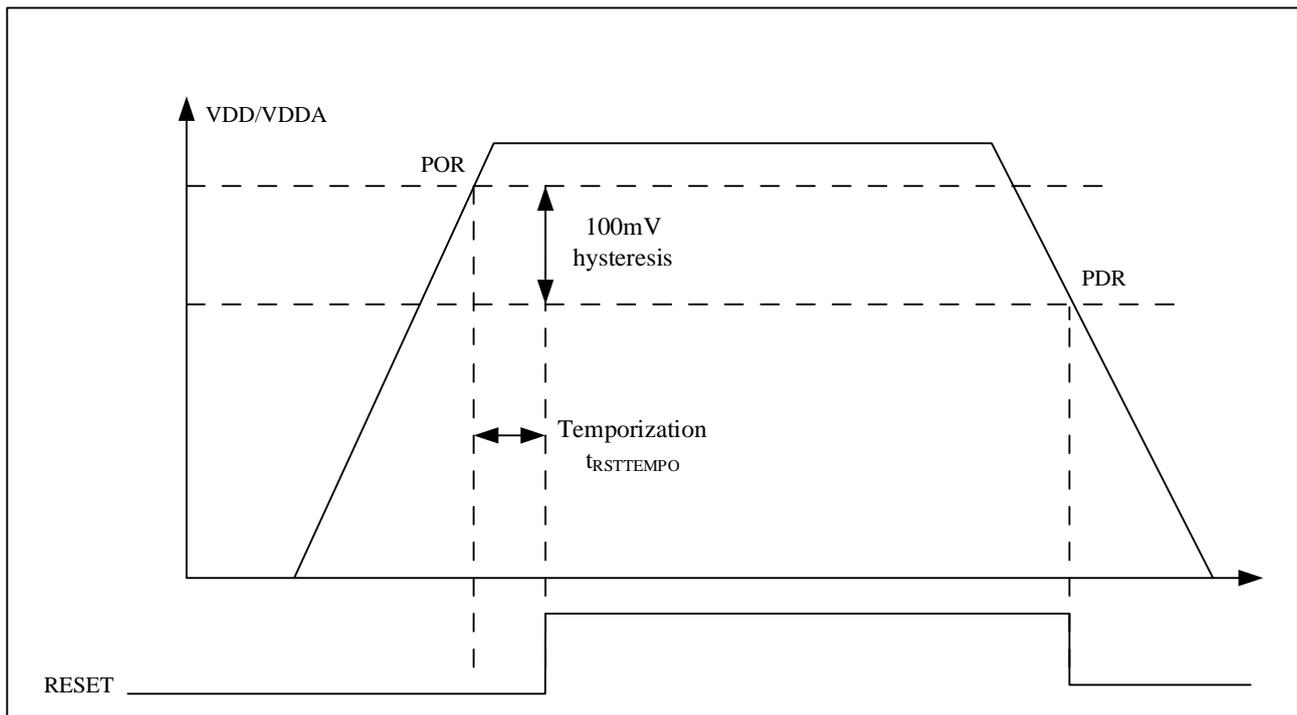


### 3.1.2 电压监控

#### 3.1.2.1 上电复位 (POR) 和下电复位 (PDR)

上电复位(POR)和下电复位(PDR)电路集成在芯片内部。当 $V_{DD}/V_{DDA}$  低于规定的限制电压  $V_{POR}/V_{PDR}$  时，系统保持在复位状态，无需外部复位电路。有关上电和断电复位的详细信息，请参阅数据手册的电气特性部分。

图 3-2 上电复位和掉电复位的波形图



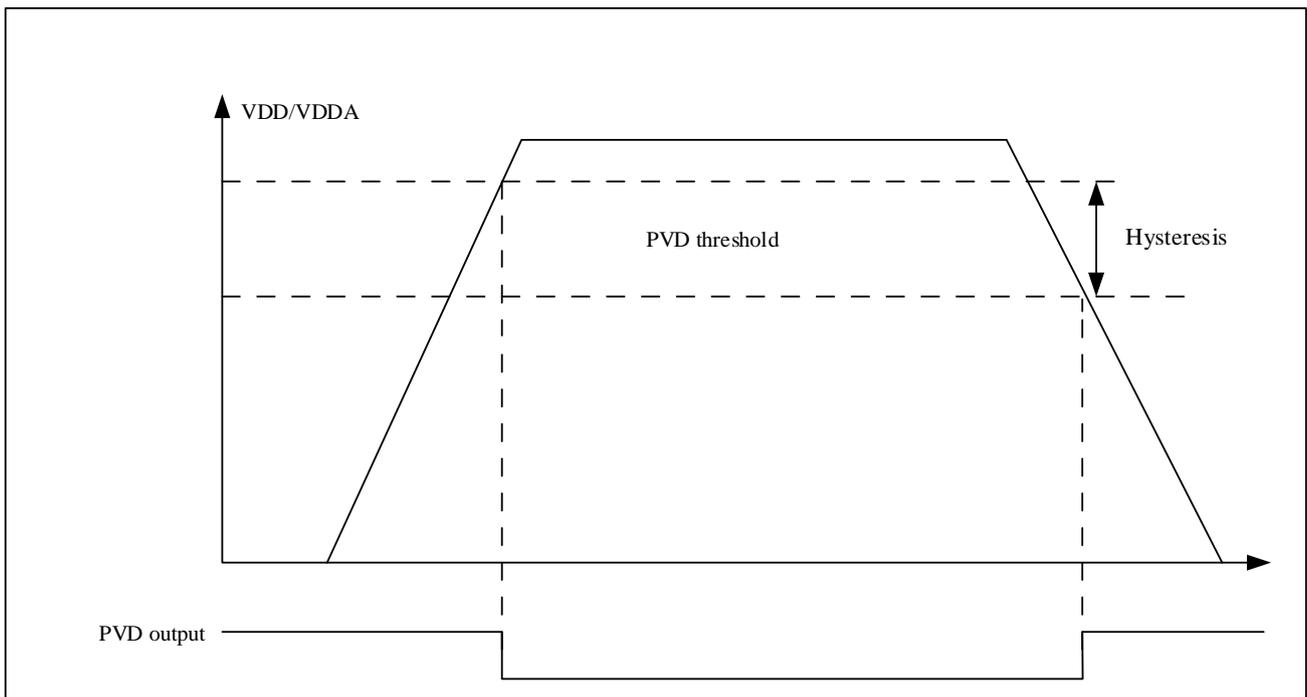
### 3.1.2.2 可编程电压监测器 (PVD)

PVD 通过比较 VDD 电压与 PWR\_CTRL 寄存器中的相关位来产生中断。通过 PWR\_CTRL.PLS[3:0]选择监控电压的阈值，通过设置 PWR\_CTRL.PVDEN 启用 PVD。

PWR\_CTRLSTS.PVDO 标志用于指示 VDD 是否高于 PVD 电压阈值。该事件在内部连接到 EXTI 线 7，如果在外部中断寄存器中启用了中断，则会产生中断。根据 EXTI 线 7 的上升/下降沿触发设置，当 VDD 下降到 PVD 阈值以下和/或 VDD 上升到 PVD 阈值以上时，会发生 PVD 中断。此功能可用于执行紧急关断任务。

- 2.0V 和 5V 之间有 15 个等级，软件可选
- 电压迟滞范围：+/-100mV
- 支持可编程的滤波（基于 LSI 时钟，1~32 个 cycle，30~1024us 可配置，默认值 300us 左右，PWR 内部实现）

图 3-3 PVD 阈值图



### 3.2 低功耗模式

MCU 有 2 种电源模式：RUN 和 STOP，不同的模式有不同的性能和功耗。MCU 功耗模式总结如下所示。

表 3-2 电源模式

模式	状态	进入	退出
RUN	CPU 运行。 所有外设运行可配置。	上电、系统复位或从其他低功耗模式唤醒	进入其他低功耗模式
STOP	CPU 处于深度睡眠模式。 外设时钟关闭，电压调节器处于低功耗模式。 HSI 关闭。 LSI 开启。 Flash 进入深度待机模式。 SRAM、寄存器保持，GPIO 状态保持。 唤醒后，系统时钟配置保持，代码从停止的位置继续执行。	WFI/WFE： 1) SLEEPDEEP = 1，没有挂起的中断/事件。	任何通过 EXTI 的中断唤醒/事件。

注意:

1. IWDG 可配为中断或复位，可中断唤醒,或根据配置选择复位（方式同 NRST）
2. 所有 EXTI 均可唤醒，所有 GPIO 可配置到 EXTI0~6，可记录 7 组唤醒源（PA0~PA3：第 1 组，PA4~PA7：第 2 组，PA8~PA11：第 3 组，PA12~PA15：第 4 组，PB0~PB3：第 5 组，PB4~PB8：第 6 组，PF0~PF3：第 7 组）
3. RUN 模式下，GPIO 输入滤波宽度 0~8 个滤波时钟宽度可配，滤波时钟（RCC 提供）可配，支持 1~16

## 分频

## 4. STOP 模式下, EXTIO~6 输入滤波可使能, 滤波宽度 2 LSI CLK

不同模块在不同功耗模式下的运行使能情况如下表所示:

表 3-3 外设运行状态

Main Blocks	Run/Active	Stop mode	
		Status	Wakeup capability
Cortex-M0	Y	-	-
FLASH	Y	-	-
SRAM	Y	Y (RET)	-
POR/PDR	Y	Y	Y
PVD	O	O	O
DMA	O	-	-
UART1	O	-	-
UART2	O	-	-
UART3	O	O	O
I2C1/2	O	-	-
SPI1/2/3	O	-	-
AD Timer	O	-	-
GP Timer	O	-	-
BS Timer	O	O	O
HSI	O	-	-
LSI	O	Y	-
IWDG	O	O	O
ADC	O	-	-
COMP	O	O	O
OPA	O	-	-
SysTick	O	-	-
GPIOs	O	O	O

注意:

1. Y: 是 (启用), O: 可选 (默认禁用, 软件启用), -: 无效。

### 3.2.1 STOP 模式

STOP 模式基于 Cortex®-M0 深度睡眠模式与外设时钟控制相结合。HSI 被禁用, LSI 开启。所有 GPIO 状态、SRAM 和寄存器内容保持。FLASH 处于深度待机模式。唤醒后, 代码从挂起的地方继续执行。

#### 3.2.1.1 进入 STOP 模式

进入 STOP 模式前, 用户需要设置 SCB\_SCR.SLEEPDEEP=1。

如果正在进行 FLASH 操作，则进入 STOP 模式的时间将延迟到存储器访问完成。

如果正在访问 APB 区域，则进入 STOP 模式的时间将延迟到 APB 访问完成。

在 STOP 模式下，可以使用以下外设：

- 独立看门狗（IWDG）：一旦启动将一直工作，直到产生一个复位。
- TIM6/PVD/COMP 外设可配置唤醒。
- UART3 可选，若使用 UART3 唤醒，则只能使用事件唤醒。

进入 STOP 模式时应禁用 ADC，以避免不必要的功耗。

### 3.2.1.2 退出 STOP 模式

当中断或事件唤醒 STOP 模式时，系统时钟配置保持，代码从挂起位置恢复。

## 3.3 Debug 模式

### 3.3.1 外设调试支持

支持部分外设 在调试状态下停止工作（TIM 1/3/4/6，I2C1/2，IWDG）。具体操作和特性请参考 3.4.5 章节对 DBG\_CTRL 寄存器位域的描述。

## 3.4 PWR 寄存器

### 3.4.1 寄存器总览

表 3-4 PWR 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
000h	PWR_CTRL	Reserved										NRSTCNT				PVD CNT				Reserved	IWDGRSTEN	Reserved				PLS[3:0]			PVDEN	Reserved								
004h	PWR_CTRLSTS	Reserved																												PVDO	Reserved							
008h	PWR_CTRL2	Reserved												MRLPDLY																MRLPEN								
00Ch	DBG_CTRL	Reserved														TIM6STP	Reserved	I2C2TIMOUT	I2C1TIMOUT	Reserved	TIM4STP	TIM3STP	Reserved	TIM1STP	Reserved	IWDGSTP	Reserved											

### 3.4.2 电源控制寄存器 (PWR\_CTRL)

偏移地址: 0x00

复位值: 0x00785000

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

位域	名称	描述																						
31:23	Reserved	保留, 必须保持复位值。																						
22:19	NRSTCNT	NRST Pin 滤波宽度 滤波宽度为 $NRSTCNT * (T_{sysclk} * 2)$ <i>注:</i> 1. all 0 表示 bypass 2. STOP 模式时, bypass																						
18:14	PVDCNT	PVD 滤波宽度 滤波宽度为 $PVDCNT * T_{LSI}$ <i>注:</i> 1. all 0 表示 bypass																						
13	Reserved	保留, 必须保持复位值。																						
12	IWDGRSTEN	IWDG 复位使能控制 0: IWDG 复位请求不会产生系统复位 1: IWDG 复位请求会产生系统复位																						
11:9	Reserved	保留, 必须保持复位值。																						
8:5	PLS[3:0]	PVD 报警电压档位选择 <table border="1" style="margin-left: 20px; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>PWR_CTRL.PLS</th><th>Voltage</th></tr> </thead> <tbody> <tr><td>0000</td><td>保留</td></tr> <tr><td>0001</td><td>2.0V</td></tr> <tr><td>0010</td><td>2.2V</td></tr> <tr><td>0011</td><td>2.4V</td></tr> <tr><td>0100</td><td>2.6V</td></tr> <tr><td>0101</td><td>2.8V</td></tr> <tr><td>0110</td><td>3.0V</td></tr> <tr><td>0111</td><td>3.2V</td></tr> <tr><td>1000</td><td>3.4V</td></tr> <tr><td>1001</td><td>3.6V</td></tr> </tbody> </table>	PWR_CTRL.PLS	Voltage	0000	保留	0001	2.0V	0010	2.2V	0011	2.4V	0100	2.6V	0101	2.8V	0110	3.0V	0111	3.2V	1000	3.4V	1001	3.6V
PWR_CTRL.PLS	Voltage																							
0000	保留																							
0001	2.0V																							
0010	2.2V																							
0011	2.4V																							
0100	2.6V																							
0101	2.8V																							
0110	3.0V																							
0111	3.2V																							
1000	3.4V																							
1001	3.6V																							

位域	名称	描述												
		<table border="1"> <tr><td>1010</td><td>3.8V</td></tr> <tr><td>1011</td><td>4.0V</td></tr> <tr><td>1100</td><td>4.2V</td></tr> <tr><td>1101</td><td>4.4V</td></tr> <tr><td>1110</td><td>4.6V</td></tr> <tr><td>1111</td><td>4.8V</td></tr> </table>	1010	3.8V	1011	4.0V	1100	4.2V	1101	4.4V	1110	4.6V	1111	4.8V
1010	3.8V													
1011	4.0V													
1100	4.2V													
1101	4.4V													
1110	4.6V													
1111	4.8V													
4	PVDEN	PVD 使能控制 0: 禁止 PVD 1: 使能 PVD												
3:0	Reserved	保留，必须保持复位值。												

### 3.4.3 电源控制状态寄存器 (PWR\_CTRLSTS)

偏移地址: 0x04

复位值: 0x00000000

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

r

位域	名称	描述
31:3	Reserved	保留，必须保持复位值。
2	PVDO	PVD 模检测输出 0: 无 PVD 事件 1: PVD 事件发生
1:0	Reserved	保留，必须保持复位值。

### 3.4.4 电源控制寄存器 2 (PWR\_CTRL2)

偏移地址: 0x08

复位值: 0x000007E1

注意: 该寄存器有写保护。修改前需要先向该地址写入 32'h5710\_3616 后, 再进行写操作

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

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

位域	名称	描述
31:13	Reserved	保留，必须保持复位值。
12:1	MRLPDLY	MR LP 控制使能 在系统进入 STOP 后，延迟使能 MR LP。 延迟时间 = MRLPDLY*(T <sub>sysclk</sub> *2)
0	MRLPEN	MR LOWPOWER 模式控制 0: 固定为使能 1: MR LOWPOWER 被自动控制

### 3.4.5 调试控制寄存器 (DBG\_CTRL)

偏移地址: 0x0C

复位值: 0x00000000

注意: 只有 POR/PDR 可以复位该寄存器。只有连上 Debugger 后，软件才可以写访问该寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												TIM6STP	Reserved	I2C2TIMOUT	
												rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C1TIMOUT	Reserved	TIM4STP	TIM3STP	Reserved	TIM1STP	Reserved	IWDGSTP	Reserved							
rw		rw	rw		rw		rw								

位域	名称	描述
31:19	Reserved	保留，必须保持复位值。
18	TIM6STP	当内核停止时停止 TIM6 计数器。 软件置 1(功能生效)或清零。 0: 禁能 1: 使能
17	Reserved	保留，必须保持复位值。
16	I2C2TIMOUT	当内核停止时停止 I2C2 的 SMBUS 超时模式。 软件置 1 或清零。 0: 与正常模式操作相同

位域	名称	描述
		1: 冻结 SMBUS 的超时控制
15	I2C1TIMOUT	当内核停止时停止 I2C1 的 SMBUS 超时模式。 软件置 1 或清零。 0: 与正常模式操作相同 1: 冻结 SMBUS 的超时控制
14	Reserved	保留, 必须保持复位值。
13	TIM4STP	当内核停止时停止 TIM4 计数器。 软件置 1(功能生效)或清零。 0: 禁能 1: 使能
12	TIM3STP	当内核停止时停止 TIM3 计数器。 软件置 1(功能生效)或清零。 0: 禁能 1: 使能
11	Reserved	保留, 必须保持复位值。
10	TIM1STP	当内核停止时停止 TIM1 计数器。 软件置 1(功能生效)或清零。 0: 禁能 1: 使能
9	Reserved	保留, 必须保持复位值。
8	IWDGSTP	当内核进入调试状态时看门狗停止工作。 软件置 1 或清零。 0: 看门狗计数器仍然正常工作 1: 看门狗计数器停止工作
7:0	Reserved	保留, 必须保持复位值。

## 4 复位和时钟控制(RCC)

### 4.1 复位控制单元

支持以下两种复位方式：

- 电源复位
- 系统复位

#### 4.1.1 电源复位

当以下事件发生时，产生电源复位：

- 上电/掉电复位（POR/PDR 复位）

电源复位将复位所有寄存器。（见图 3-1 电源框图）

复位源将最终作用于 NRST 引脚，并在复位过程中保持低电平。复位入口矢量被固定在地址 0x0000\_0004。更多细节，参阅表 5-1 向量表。

#### 4.1.2 系统复位

发生以下事件之一时会产生系统复位：

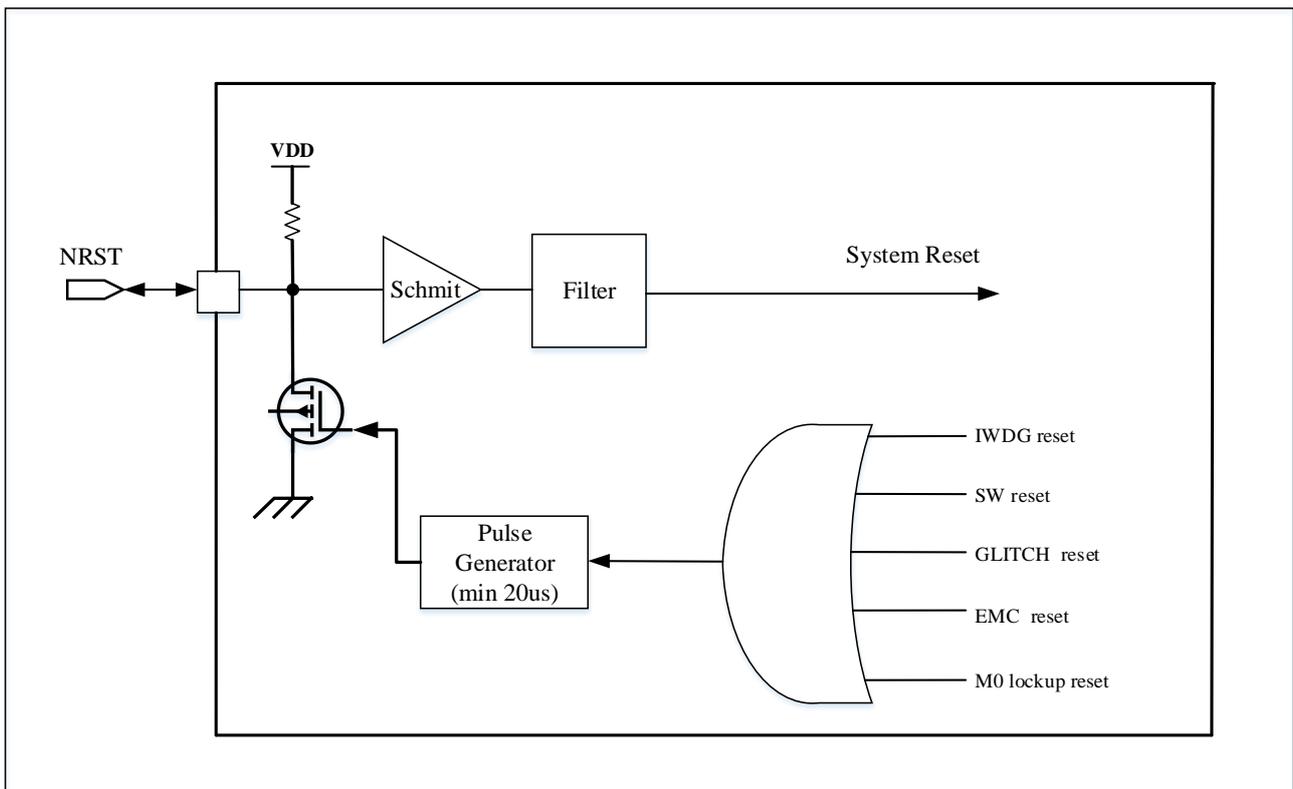
- NRST 引脚上的低电平（外部复位）
- 独立看门狗计数终止（IWDG 复位）
- 软件复位（SW 复位）
- EMC/GLITCH 复位
- M0 内核锁定复位

可以通过检查控制/状态寄存器(RCC\_CTRLSTS)中的复位标志来识别复位源。

提供给芯片的系统复位信号会在 NRST 引脚上输出。脉冲发生器保证每个复位源（外部或内部）的复位脉冲至少持续时间 20 $\mu$ s。对于外部复位，当 NRST 引脚置为低电平时会产生复位脉冲。

下图展示了复位电路：

图 4-1 复位电路



#### 4.1.2.1 软件复位

可以通过设置 Cortex®-M0 应用中断和复位控制寄存器中的 SYSRESETREQ 位来产生软件复位。有关详细信息，请参阅 Cortex®-M0 技术参考手册。

### 4.2 时钟控制单元

可以使用 2 种不同的时钟源来驱动系统时钟(SYSCLK):

- HSI 振荡器时钟
- LSI 振荡器时钟

多个预分频器可用于配置 AHB、APB 的频率。AHB 最大频率为 64MHz，APB 最大频率为 32MHz。

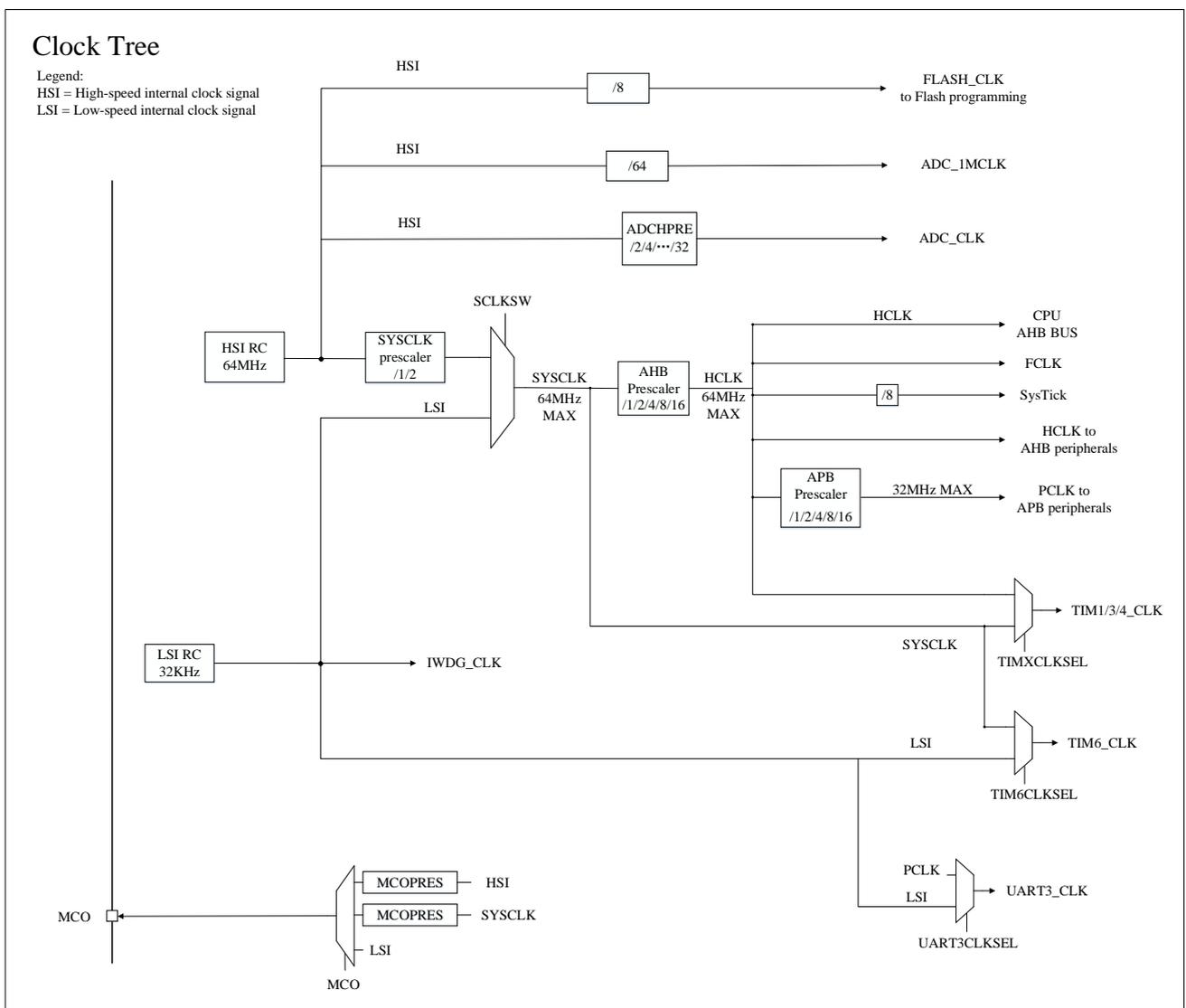
除去以下情况，其他外设时钟都源于 AHB 或 APB:

- 通过配置 RCC\_CFG.TIMXCLKSEL，可选择下述情况之一作为 TIM1/3/4 工作时钟源：
  - ◆ HCLK
  - ◆ SYSCLK
- 通过配置 RCC\_CFG.TIM6CLKSEL，可选择下述情况之一作为 TIM6 时钟源：
  - ◆ SYSCLK
  - ◆ LSI

- 通过配置 RCC\_CFG.UART3CLKSEL，可选择下述情况之一作为 UART3 时钟源：
  - ◆ APB CLK
  - ◆ LSI
- IWDG 的时钟源为 LSI 振荡器
- Flash 存储器编程接口时钟始终是 HSI 时钟
- ADC 工作时钟为 HSI 的 2/4/8/16/32 分频

### 4.2.1 时钟树

图 4-2 时钟树



1. 系统时钟的最大频率为 64MHz。
2. 有关内部时钟源特性的详细信息，请参阅产品数据手册中的“电气特性”部分。

## 4.2.2 HSI 时钟

HSI（高速内部）时钟信号由内部 64MHz RC 振荡器产生，可直接作为系统时钟。HSI RC 振荡器无需任何外部设备即可提供时钟源。

用户可以使用 `RCC_CTRL.HSITRIM[9:0]` 位调整 HSI 频率。

`RCC_CTRL.HSIRDF` 位指示 HSI RC 振荡器是否稳定。在启动时，直到该位被硬件设置，HSI RC 输出时钟才会被释放。可以通过设置 `RCC_CTRL.HSIEN` 位打开和关闭 HSI 时钟。

## 4.2.3 LSI 时钟

LSIRC 可以在 STOP 模式下为 IWDG 提供时钟。LSI 时钟频率约为 32KHz。有关详细信息，请参阅数据表的电气特性部分。

`RCC_CLKINT.LSIRDF` 位标志指示 LSI 时钟是否稳定。

LSI 在上电后不会被关闭。

## 4.2.4 系统时钟(SYSCLK)选择

系统复位后，选择 HSI 振荡器作为系统时钟。

当 LSI 时钟源准备好，可以从一个时钟源切换到另一个时钟源。

通过 `RCC_CFG.SCLKSW` 设置 LSI 作为系统时钟，并且通过 `RCC_CFG.SCLKSTS` 判断当前系统时钟是 HSI 或 LSI。

## 4.2.5 看门狗时钟

如果 IWDG 由硬件选项或软件启动。LSI 振荡器稳定后，时钟被提供给 IWDG。

## 4.2.6 时钟输出(MCO)

微控制器时钟输出(MCO)功能允许将时钟信号输出到外部 MCO 引脚。

对应的 GPIO 口寄存器必须配置为对应的功能。可以选择以下 3 个时钟信号作为 MCO 时钟：

- SYSCLK 分频值
- HSI 分频值
- LSI 值

时钟选择由 `RCC_CFG.MCO[1:0]` 位控制。

## 4.2.7 时钟校准

- 开启条件

支持系统 Active 时，软件启动校准计数，需要先打开对应时钟，并查询到时钟输出稳定

- 时钟校准计数器工作原理

校正开始后，第一个计数器使用 LSI 时钟进行计数，计满 RCC\_CTRL.LSICALLEN[1:0]周期后结束计数，产生计数 Done 信号。

在该计数期间，同时第二个计数器使用 HSI 时钟进行计数，并在计数 Done 时，把计数结果存入寄存器，供 CPU 通过寄存器查询。

● 校准流程

1. 等待 HSI、LSI 处于就绪状态
2. 配置 RCC\_CTRL.LSICALLEN[1:0]确认 LSI 计数周期为 N（N 可配 128、256、512、1024）
3. 配置 RCC\_CTRL.LSICALEN 为 1 使能 LSI 校准计数
4. 等待 RCC\_CTRL.LSICALCF 为 1，计数结束，读取 RCC\_LSICAL 计数结果
5. 通过公式计算 LSI 测量频率=64000000\*N/RCC\_LSICAL，根据测量频率与 32KHz 的偏差，调节 TRIM 值

## 4.3 RCC 寄存器

### 4.3.1 寄存器总览

表 4-1 RCC 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
000h	RCC_CTRL	Reserved										HSITRIM[9:0]										Reserved	LSICALCF	LSICALLEN[1:0]		LSICALEN	Reserved						HSIRDf	HSIEN	
004h	RCC_CFG	Reserved	MCOPRES[2:0]			MCO[1:0]		Reserved										SYSPRES	Reserved		TIMXCLKSEL	TIM6CLKSEL	UART3CLKSEL	APBPRES[2:0]		Reserved		AHBPRES[2:0]		Reserved		SCLKSTS	SCLKSW		
008h	RCC_CLKINT	Reserved														HSIRDCLR	Reserved	LSIRDCLR	Reserved														HSIRDf	Reserved	LSIRDf
00Ch	RCC_APBRS	OPARST	Reserved	COMPST	Reserved										UART3RST	UART2RST	UART1RST	TIM6RST	TIM4RST	TIM3RST	TIM1RST	Reserved	SPI2RST	SPI1RST	Reserved						I2C2RST	I2C1RST	IOMRST		
010h	RCC_AHBCLKEN	Reserved														ADCFEN	Reserved						HDIEN	Reserved	SQRTEN	Reserved				DMAEN					
014h	RCC_APBCLKEN	OPAEN	COMPILTEN	COMPEN	Reserved										UART3EN	UART2EN	UART1EN	TIM6EN	TIM4EN	TIM3EN	TIM1EN	Reserved	SPI2EN	SPI1EN	Reserved						PWREN	I2C2EN	I2C1EN	IOMEN	
01Ch	RCC_CTRLSTS	Reserved														LKUPRSTF	Reserved	EMCGBNF	EMCGBF	GLITCHRSTF	IWDGRSTF	SFTRSTF	PORRSTF	PINRSTF	Reserved		RMRSTF								
020h	RCC_AHBPRST	Reserved														ADCRST	Reserved						HDIRST	Reserved	SQTRST	Reserved									
024h	RCC_CFG2	Reserved																										ADCHPRE[2:0]							
02Ch	RCC_LSICAL	Reserved										HSICALCNT																							

030h	RCC_EMCCTRL	LKUPRSTEN	Reserved	GBNSW	GBNRST	GBNDET	Reserved	GBNDETSEL	GBSW	GBRST	GBDET	Reserved	GBNDETSEL	GVSW	GVRST	GVDDET	GVDDETSEL
034h	RCC_LSCTRL	Reserved											TRIMSEL	LSITRIM[8:0]			
038h	RCC_TIMFILTCFG	IOFLITCLK	Reserved	TIM4FILTCCLK				Reserved	TIM3FILTCCLK				Reserved	TIM1FILTCCLK			

### 4.3.2 时钟控制寄存器 (RCC\_CTRL)

偏移地址: 0x00

复位值: 0x00400603

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

位域	名称	描述
31:23	Reserved	保留, 必须保持复位值
22:13	HSITRIM[9:0]	HSI 时钟 trim 值 默认 10'b10_0000_0000, 由软件写入, 用以校准内部 HSI RC 振荡器的频率。 HSITRIM[9:4]用于粗调, 调节步长 960KHz; HSITRIM[3:0]用于细调, 调节步长 160KHz。 <i>注: HSI trim 的复位值在出厂后每颗 MCU 会不相同, 以 MCU 实际值为准。</i>
12	Reserved	保留, 必须保持复位值
11	LSICALCF	时钟校准完成标志 0: 校准未完成 1: 校准完成
10:9	LSICALLEN[1:0]	时钟校准长度 在 LSICALLEN 个 LSI 周期内, 统计 HSI 时钟的周期数 00: 128 个 LSI 周期 01: 256 个 LSI 周期 10: 512 个 LSI 周期 11: 1024 个 LSI 周期
8	LSICALEN	时钟校准使能 0: 禁能 1: 使能
7:2	Reserved	保留, 必须保持复位值
1	HSIRDF	HSI 时钟准备完成标志

位域	名称	描述
		0: 未准备完成 1: 准备完成
0	HSIEN	HSI 时钟使能 0: 禁能 1: 使能 上电后默认 HSI 使能，且作为系统时钟，当需要切换 LSI 作为系统时钟，在切换完成后，禁能 HSI

### 4.3.3 时钟配置寄存器（RCC\_CFG）

偏移地址：0x04

复位值：0x20000400

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	MCOPRES[2:0]		MCO[1:0]		Reserved										SYSPRES
	rw		rw												rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		TIMXCLKSEL	TIM6CLKSEL	UART3CLKSEL	APBPRES[2:0]		Reserved	AHBPRES[2:0]		Reserved		SCLKSTS	SCLKSW		
		rw	rw	rw	rw			rw				r	rw		

位域	名称	描述
31	Reserved	保留，必须保持复位值
30:28	MCOPRES[2:0]	MCO 输出 SYSCLK、HSI 分频配置 010: 2 分频 011: 3 分频 ... 111: 7 分频 其他配置不被允许
27:26	MCO[1:0]	MCO 输出时钟选择 00: 没有时钟输出 01: 选择输出 LSI 10: 选择输出 SYSCLK 分频值 11: 选择输出 HSI 分频值
25:17	Reserved	保留，必须保持复位值
16	SYSPRES	SYSClk 分频配置 0: 不分频 1: 2 分频
15:14	Reserved	保留，必须保持复位值
13	TIMXCLKSEL	TIM1/3/4 工作时钟的时钟源选择

位域	名称	描述
		0: hclk 1: sysclk
12	TIM6CLKSEL	TIM6 工作时钟的时钟源选择 0: sysclk 1: LSI 如果需要 TIM6 在系统 STOP 状态时工作，必须配置 TIM6 工作时钟的时钟源为 LSI
11	UART3CLKSEL	UART3 工作时钟的时钟源选择 0: APB CLK 1: LSI 如果需要 UART3 在系统 STOP 状态时工作，必须配置 UART3 工作时钟的时钟源为 LSI
10:8	APBPRES[2:0]	APB 时钟分频配置 100: HCLK 的 2 分频 101: HCLK 的 4 分频 110: HCLK 的 8 分频 111: HCLK 的 16 分频 其他值: 不分频 <i>注意: 修改分频系数时, 要确保 APB CLK 频率不超过 32M</i>
7	Reserved	保留, 必须保持复位值
6:4	AHBPRES[2:0]	AHB 时钟分频配置 100: SYSCLK 的 2 分频 101: SYSCLK 的 4 分频 110: SYSCLK 的 8 分频 111: SYSCLK 的 16 分频 其他值: 不分频 <i>注意: 修改分频系数时, 要确保 APB CLK 频率不超过 32M</i>
3:2	Reserved	保留, 必须保持复位值
1	SCLKSTS	系统时钟切换状态 0: HSI 作为当前 SYSCLK 的时钟源 1: LSI 作为当前 SYSCLK 的时钟源
0	SCLKSW	系统时钟切换控制 0: SYSCLK 时钟源选择 HSI 1: SYSCLK 时钟源选择 LSI

### 4.3.4 时钟状态寄存器 (RCC\_CLKINT)

偏移地址: 0x08

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													HSIRDCL R	Reserved	LSIRDCL R
													w		w

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

Reserved	HSIRDF	Reserved	LSIRDF
	r		r

位域	名称	描述
31:19	Reserved	保留，必须保持复位值
18	HSIRDCLR	HSIRDF 清除 0: 无影响 1: 清除 HSIRDF 标志
17	Reserved	保留，必须保持复位值
16	LSIRDCLR	LSIRDF 清除 0: 无影响 1: 清除 LSIRDF 标志
15:3	Reserved	保留，必须保持复位值
2	HSIRDF	HSI 准备完成标志 0: 未准备完成 1: 准备完成
1	Reserved	保留，必须保持复位值
0	LSIRDF	LSI 准备完成标志 0: 未准备完成 1: 准备完成

### 4.3.5 APB 外设复位寄存器 (RCC\_APBRSR)

偏移地址: 0x0C

复位值: 0x00000000

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

OPARST	Reserved	COMPRST	Reserved										UART3RST	UART2RST	UART1RST
rw		rw											rw	rw	rw

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

TIM6RST	TIM4RST	TIM3RST	TIM1RST	Reserved	SPI2RST	SPI1RST	Reserved					I2C2RST	I2C1RST	IOMRST
rw	rw	rw	rw		rw	rw						rw	rw	rw

位域	名称	描述
31	OPARST	OPA 软复位 0: 清除复位 1: 产生复位
30	Reserved	保留，必须保持复位值

位域	名称	描述
29	COMPRST	COMP 软复位 0: 清除复位 1: 产生复位
28:19	Reserved	保留, 必须保持复位值
18	UART3RST	UART3 软复位 0: 清除复位 1: 产生复位
17	UART2RST	UART2 软复位 0: 清除复位 1: 产生复位
16	UART1RST	UART1 软复位 0: 清除复位 1: 产生复位
15	TIM6RST	TIM6 软复位 0: 清除复位 1: 产生复位
14	TIM4RST	TIM4 软复位 0: 清除复位 1: 产生复位
13	TIM3RST	TIM3 软复位 0: 清除复位 1: 产生复位
12	TIM1RST	TIM1 软复位 0: 清除复位 1: 产生复位
11	Reserved	保留, 必须保持复位值
10	SPI2RST	SPI2 软复位 0: 清除复位 1: 产生复位
9	SPI1RST	SPI1 软复位 0: 清除复位 1: 产生复位
8:3	Reserved	保留, 必须保持复位值
2	I2C2RST	I2C2 软复位 0: 清除复位 1: 产生复位
1	I2C1RST	I2C1 软复位 0: 清除复位 1: 产生复位
0	IOMRST	IOM 软复位 0: 清除复位 1: 产生复位

### 4.3.6 AHB 外设时钟使能寄存器 (RCC\_AHBCLKEN)

偏移地址: 0x10

复位值: 0x00000000

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

位域	名称	描述
31:13	Reserved	保留, 必须保持复位值
12	ADCEN	ADC 总线时钟使能 0: 禁能 1: 使能
11:8	Reserved	保留, 必须保持复位值
7	HDIVEN	HDIV 总线时钟使能 0: 禁能 1: 使能
6	Reserved	保留, 必须保持复位值
5	SQRTEN	SQRT 总线时钟使能 0: 禁能 1: 使能
4:1	Reserved	保留, 必须保持复位值
0	DMAEN	DMA 总线时钟使能 0: 禁能 1: 使能

### 4.3.7 APB 外设时钟使能寄存器 (RCC\_APBCLKEN)

偏移地址: 0x14

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OPAEN	COMPFIL TEN	COMPEN	Reserved										UART3EN	UART2EN	UART1EN
rw	rw	rw											rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIM6EN	TIM4EN	TIM3EN	TIM1EN	Reserved	SPI2EN	SPI1EN	Reserved				PWREN	I2C2EN	I2C1EN	IOMEN	
rw	rw	rw	rw		rw	rw					rw	rw	rw	rw	

位域	名称	描述
31	OPAEN	OPA 总线时钟使能 0: 禁能 1: 使能
30	COMPFILTEN	COMP 滤波器时钟使能 0: 禁能 1: 使能
29	COMPEN	COMP 总线时钟使能 0: 禁能 1: 使能
28:19	Reserved	保留, 必须保持复位值
18	UART3EN	UART3 总线时钟使能 0: 禁能 1: 使能
17	UART2EN	UART2 总线时钟使能 0: 禁能 1: 使能
16	UART1EN	UART1 总线时钟使能 0: 禁能 1: 使能
15	TIM6EN	TIM6 总线时钟使能 0: 禁能 1: 使能
14	TIM4EN	TIM4 总线时钟使能 0: 禁能 1: 使能
13	TIM3EN	TIM3 总线时钟使能 0: 禁能 1: 使能
12	TIM1EN	TIM1 总线时钟使能 0: 禁能 1: 使能
11	Reserved	保留, 必须保持复位值
10	SPI2EN	SPI2 总线时钟使能 0: 禁能 1: 使能
9	SPI1EN	SPI1 总线时钟使能 0: 禁能

位域	名称	描述
		1: 使能
8:4	Reserved	保留, 必须保持复位值
3	PWREN	PWR 总线时钟使能 0: 禁能 1: 使能
2	I2C2EN	I2C2 总线时钟使能 0: 禁能 1: 使能
1	I2C1EN	I2C1 总线时钟使能 0: 禁能 1: 使能
0	IOMEN	IOM 总线时钟使能 0: 禁能 1: 使能

### 4.3.8 控制/状态寄存器 (RCC\_CTRLSTS)

偏移地址: 0x1C

复位值: 0x00000018

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				LKUPRST F	Reserved	EMCGBN F	EMCGBF	GLITCHR STF	IWDGRST F	SFTRSTF	PORRSTF	PINRSTF	Reserved		RMRSTF
				r		r	r	r	r	r	r	r			rw

位域	名称	描述
31:12	Reserved	保留, 必须保持复位值
11	LKUPRSTF	CPU Lockup 触发系统复位标志 0: 未产生复位 1: 产生过复位 RMRSTF 写 1, 该 bit 清零
10	Reserved	保留, 必须保持复位值
9	EMCGBNF	EMC GBN 触发系统复位标志 0: 未产生复位 1: 产生过复位 RMRSTF 写 1, 该 bit 清零

位域	名称	描述
8	EMCGBF	EMC GB 触发系统复位标志 0: 未产生复位 1: 产生过复位 RMRSTF 写 1, 该 bit 清零
7	GLITCHRSTF	GLITCH 触发系统复位标志 0: 未产生复位 1: 产生过复位 RMRSTF 写 1, 该 bit 清零
6	IWDGRSTF	IWDG 触发系统复位标志 0: 未产生复位 1: 产生过复位 RMRSTF 写 1, 该 bit 清零
5	SFTRSTF	软件复位触发系统复位标志 0: 未产生复位 1: 产生过复位 RMRSTF 写 1, 该 bit 清零
4	PORRSTF	POR 复位标志 0: 未产生复位 1: 产生过复位
3	PINRSTF	NRST PIN 触发系统复位标志 0: 未产生复位 1: 产生过复位 RMRSTF 写 1, 该 bit 清零
2:1	Reserved	保留, 必须保持复位值
0	RMRSTF	清除复位标志 0: 不清除复位标志 1: 清除 RCC_CTRLSTS 中所有复位标志

### 4.3.9 AHB 外设复位寄存器 (RCC\_AHBPRST)

偏移地址: 0x20

复位值: 0x00000000

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

位域	名称	描述
31:13	Reserved	保留，必须保持复位值
12	ADCRST	ADC 软复位 0: 清除复位 1: 产生复位
11:8	Reserved	保留，必须保持复位值
7	HDIVRST	HDIV 软复位 0: 清除复位 1: 产生复位
6	Reserved	保留，必须保持复位值
5	SQRTRST	SQRT 软复位 0: 清除复位 1: 产生复位
4:0	Reserved	保留，必须保持复位值

### 4.3.10 时钟配置寄存器 2 (RCC\_CFG2)

偏移地址: 0x24

复位值: 0x00000003

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

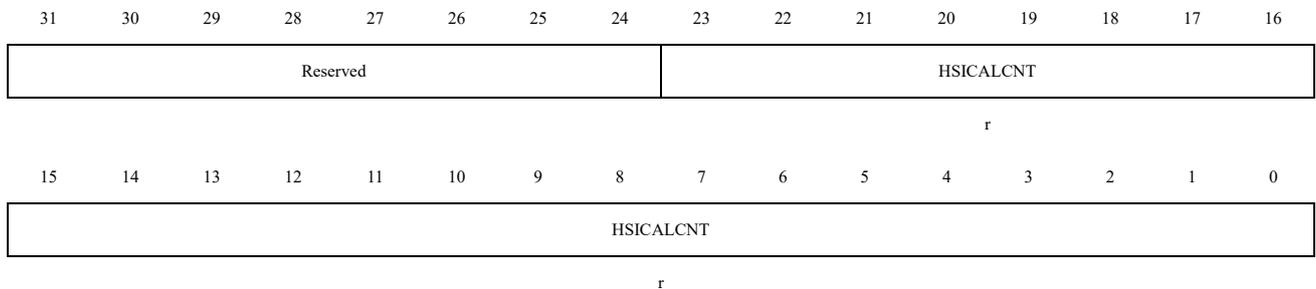
rw

位域	名称	描述
31:3	Reserved	保留，必须保持复位值
2:0	ADCHPRE[2:0]	ADC 工作时钟分频配置 001: HSI 时钟 2 分频 011: HSI 时钟 4 分频 100: HSI 时钟 8 分频 101: HSI 时钟 16 分频 110: HSI 时钟 32 分频 其他: HSI 时钟 32 分频

### 4.3.11 时钟校准计数寄存器 (RCC\_LSICAL)

偏移地址: 0x2C

复位值: 0x00000000



位域	名称	描述
31:24	Reserved	保留, 必须保持复位值
23:0	HSICALCNT	RCC_CTRL.LSICALLEN[1:0]时钟校准周期内 HSI (固定 64MHz) 产生的周期数量

### 4.3.12 EMC 控制寄存器 (RCC\_EMCCTRL)

偏移地址: 0x30

复位值: 0x00000000



位域	名称	描述
31	LKUPRSTEN	CPU Lockup 触发复位使能 0: 禁能 1: 使能
31:24	Reserved	保留, 必须保持复位值
23	GBNSW	EMC GBN 发生时, 系统时钟源由 HSI 切换到 LSI 0: 切换禁能 1: 切换使能
22	GBNRST	EMC GBN 发生时, 产生系统复位 0: 不产生复位

位域	名称	描述
		1: 产生复位
21	GBNDET	EMC GBN 检测使能 0: 禁能 1: 使能
20:18	Reserved	保留, 必须保持复位值
17:16	GBNDETSEL	EMC GBN 检测档位选择 0 是迟滞区间最大, 3 是迟滞区间最小, 依次递减
15	GBSW	EMC GB 发生时, 系统时钟源由 HSI 切换到 LSI 0: 切换禁能 1: 切换使能
14	GBRST	EMC GB 发生时, 产生系统复位 0: 不产生复位 1: 产生复位
13	GBDET	EMC GB 检测使能 0: 禁能 1: 使能
12:10	Reserved	保留, 必须保持复位值
9:8	GBDETSEL	EMC GB 检测档位选择 0 是迟滞区间最大, 3 是迟滞区间最小, 依次递减
7	GVSW	GLITCH 发生时, 系统时钟源由 HSI 切换到 LSI 0: 切换禁能 1: 切换使能
6	GVRST	GLITCH 发生时, 产生系统复位 0: 不产生复位 1: 产生复位
5	GVDET	GLITCH 检测使能 0: 禁能 1: 使能 <i>注: GVDET 位使能后, 需要延时 10us 及以上, 才可以使能 GVRST 位和 GVSW 位。</i>
4:0	GVDETSEL	GLITCH 检查档位选择 bit2 为实际最高位, 0 是灵敏度最高, 0x1F 是灵敏度最低, 依次递减

### 4.3.13 LSI 校准控制寄存器 (RCC\_LSCTRL)

偏移地址: 0x34

复位值: 0x00000140

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

Reserved	TRIMSEL	LSITRIM[8:0]
rw		rw

位域	名称	描述
31:10	Reserved	保留，必须保持复位值
9	TRIMSEL	LSI TRIM 源选择 0: 来源于 AFEC 1: 来源于 RCC_LSCTRL.LSITRIM[8:0]
8:0	LSITRIM[8:0]	LSI 时钟 trim 值 默认 9'b1_0100_0000，由软件写入，用以校准内部 LSI RC 振荡器的频率。 调节步长 100Hz。 <i>注：LSI trim 的复位值在出厂后每颗 MCU 会不相同，以 MCU 实际值为准。</i>

#### 4.3.14 滤波时钟配置寄存器（RCC\_TIMFILTCFG）

偏移地址：0x38

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IOFLITCLK				Reserved								TIM4FILTCLK			
rw												rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			TIM3FILTCLK					Reserved			TIM1FILTCLK				
			rw								rw				

位域	名称	描述
31:28	IOFLITCLK	IOM 滤波单元工作时钟分频控制（基于 HCLK） 0000: 不分频 0001: 2 分频 0010: 3 分频 ... 1110: 15 分频 1111: 16 分频
27:21	Reserved	保留，必须保持复位值
20:16	TIM4FILTCLK	TIM4 滤波单元工作时钟分频控制（基于 TIM4 工作时钟） 00000: 不分频 00001: 2 分频 00010: 3 分频 ...

位域	名称	描述
		11110: 31 分频 11111: 32 分频
15:13	Reserved	保留, 必须保持复位值
12:8	TIM3FILTCLK	TIM3 滤波单元工作时钟分频控制 (基于 TIM3 工作时钟) 00000: 不分频 00001: 2 分频 00010: 3 分频 ... 11110: 31 分频 11111: 32 分频
7:5	Reserved	保留, 必须保持复位值
4:0	TIM1FILTCLK	TIM1 滤波单元工作时钟分频控制 (基于 TIM1 工作时钟) 00000: 不分频 00001: 2 分频 00010: 3 分频 ... 11110: 31 分频 11111: 32 分频

## 5 中断和事件

### 5.1 嵌套向量中断寄存器

#### 特性

- 21 个可屏蔽中断通道。
- 4 个可编程的优先等级（使用了 2 位中断优先级）；
- 低延迟的异常和中断处理；
- 电源管理控制；
- 系统控制寄存器的实现；

嵌套向量中断控制器（NVIC）和处理器核的接口紧密相连，可以实现低延迟的中断处理和高效地处理晚到的中断。嵌套向量中断控制器管理着包括内核异常等中断。

#### 5.1.1 SysTick 校准值寄存器

系统嘀嗒校准值固定为 8000，当系统嘀嗒时钟设定为 8MHz（HCLK/8 的最大值），产生 1ms 时间基准。

#### 5.1.2 中断和异常向量

表 5-1 向量表

位置	优先级	优先级类型	名称	说明	地址
-	-	-	-	保留(Reserved)	0x0000 0000
-	-3	固定	Reset	复位(Reset)	0x0000 0004
-	-2	固定	NMI	不可屏蔽中断。	0x0000 0008
-	-1	固定	HardFault	所有类型的错误(fault)	0x0000 000C
-	3	可设置	SVCall	通过 SWI 指令调用的系统服务	0x0000 002C
-	5	可设置	PendSV	可挂起的系统服务请求	0x0000 0038
-	6	可设置	SysTick	系统嘀嗒定时器	0x0000 003C
0	7	可设置	PVD	PVD 中断（联接 EXTI line 7）	0x0000 0040
1	8	可设置	FLASH	Flash 全局中断	0x0000 0044
2	9	可设置	EXTI0_6	EXTI0_6（ALL GPIO MUX 分组连接到 EXTI line 0~6）中断	0x0000 0048

3	10	可设置	COMP	比较器中断(connected to EXTI line 8)	0x0000 004C
4	11	可设置	UART1	串口中断	0x0000 0050
5	12	可设置	UART2	串口中断	0x0000 0054
6	13	可设置	TIM1_BRK_UP_TRG_COM	TIM1 刹车、更新、触发和通信中断	0x0000 0058
7	14	可设置	TIM1_CC	TIM1 捕获比较中断	0x0000 005C
8	15	可设置	ADC	ADC 全局中断, 分段 ADC 管理	0x0000 0060
9	16	可设置	SPI1	SPI1 全局中断	0x0000 0064
10	17	可设置	IWDG	IWDG 中断	0x0000 0068
11	18	可设置	TIM4	TIM4 全局中断, 32 位 GP	0x0000 006C
12	19	可设置	TIM3	TIM3 全局中断, 16 位 GP	0x0000 0070
13	20	可设置	TIM6	TIM6 定时器, 基本定时器(EXTI line 9)	0x0000 0074
14	21	可设置	UART3	串口中断	0x0000 0078
15	22	可设置	SPI2	SPI2 全局中断	0x0000 007C
16	23	可设置	I2C1_EV	I2C1 事件中断	0x0000 0080
17	24	可设置	I2C1_ER	I2C1 错误中断	0x0000 0084
18	25	可设置	I2C2_EV	I2C2 事件中断	0x0000 0088
19	26	可设置	I2C2_ER	I2C2 错误中断	0x0000 008C
20	27	可设置	DMA	DMA 全局中断	0x0000 0090

## 5.2 外部中断/事件控制器 (EXTI)

### 5.2.1 简介

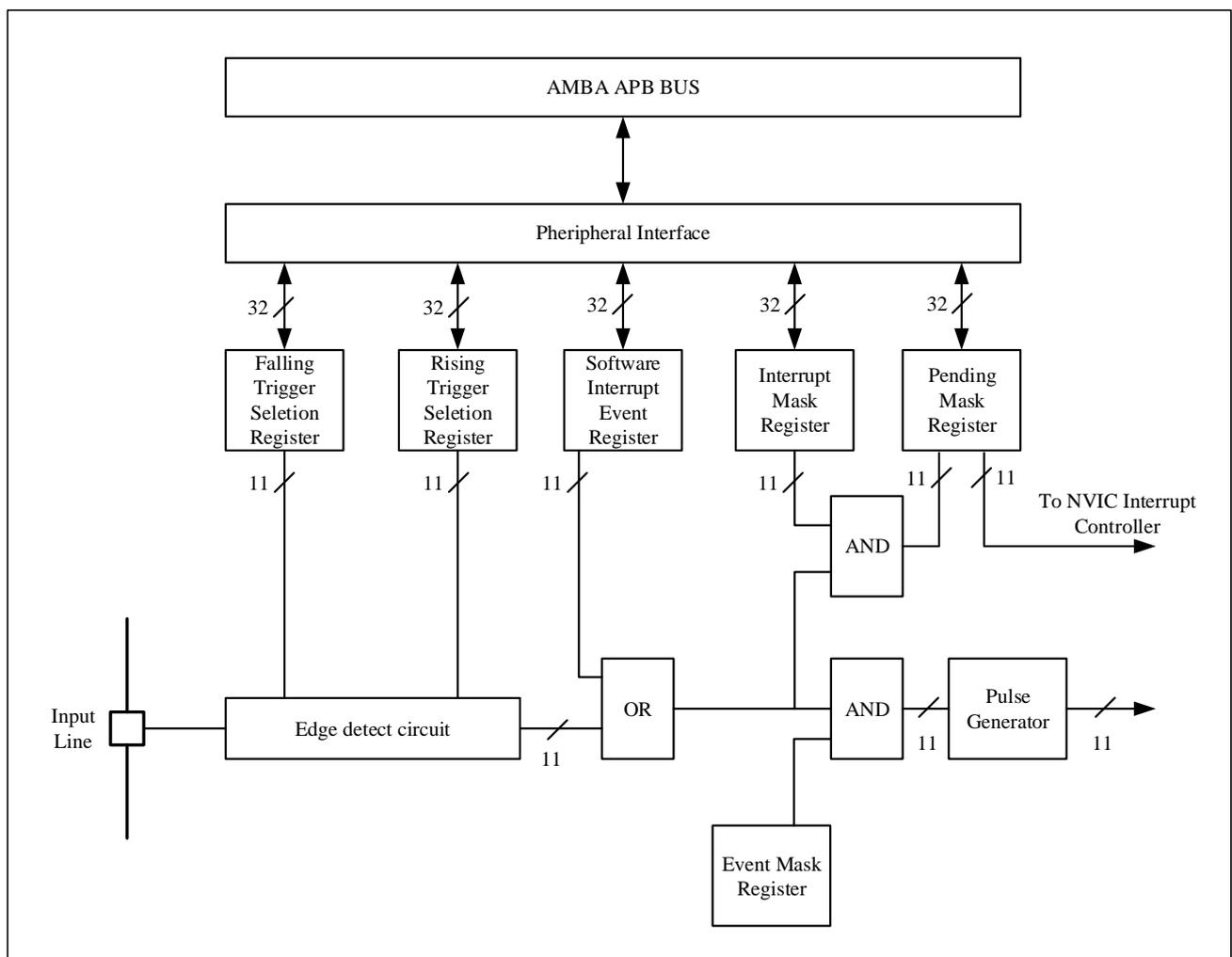
外部中断/事件控制器包含 11 个产生中断/事件触发的边沿检测电路, 每条输入线可以独立地配置脉冲或挂起输入类型, 以及上升沿、下降沿或者双边沿 3 种触发事件类型, 也可以独立地被屏蔽。挂起寄存器保持着状态线的中断请求, 可通过在挂起寄存器的对应位写'1'操作, 清除中断请求。

### 5.2.2 主要特性

EXTI 控制器的主要特性如下：

- 支持 11 个软件中断/事件请求
- 每条输入线对应的中断/事件都能独立配置触发或屏蔽
- 每条中断线都有独立的状态位
- 支持脉冲或挂起输入类型
- 支持上升沿、下降沿或双边沿 3 种触发事件类型
- 可唤醒退出低功耗模式

图 5-1 外部中断/事件控制器框图



### 5.2.3 功能描述

EXTI 包含 11 条中断线。要产生中断，必须配置外部中断控制器的 NVIC 中断通道使能相应的中断线。通过沿触发配置寄存器 EXTI\_RT\_CFG 和 EXTI\_FT\_CFG 选择上升沿、下降沿或双边沿触发事件类型，并将中断屏蔽寄存器 EXTI\_IMASK 的相应位写‘1’开放允许中断请求。当外部中断线上检测到预设的边沿触发极性，

将产生一个中断请求，对应的挂起位也随之被置‘1’。在挂起寄存器的对应位写‘1’，将清除该中断请求。

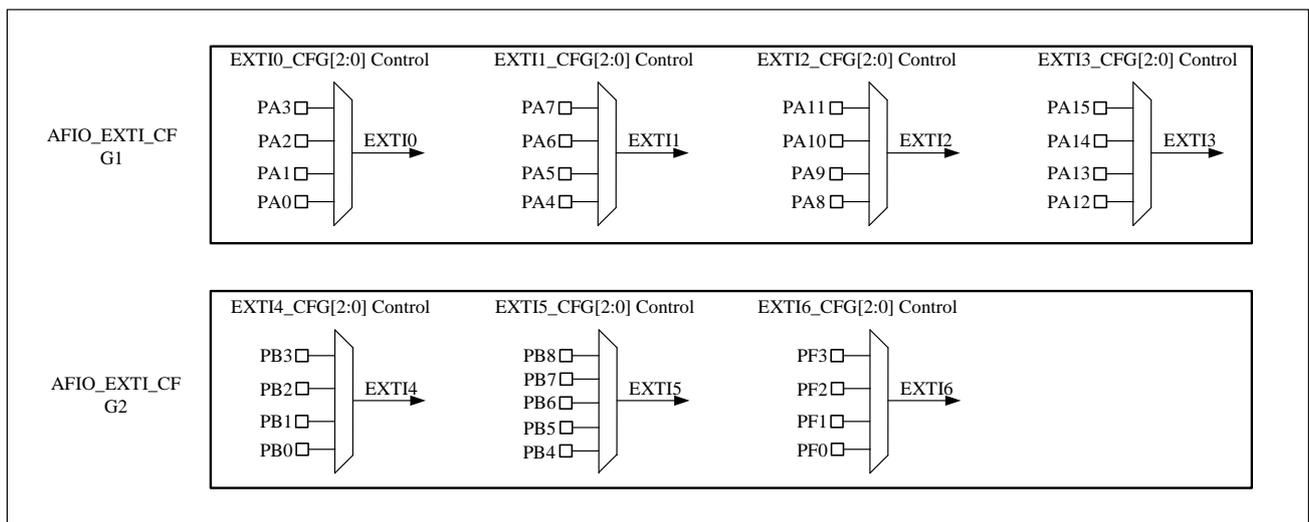
要产生事件，必须配置并使能对应的事件线。根据需要的边沿检测极性，设置上升/下降沿触发配置寄存器，同时在事件屏蔽寄存器的相应位写‘1’允许中断请求。当事件线上发生预设的边沿时，将产生一个事件请求脉冲，对应的挂起位不被置‘1’。

另外，通过在软件中断/事件寄存器写‘1’，也可以通过软件产生中断/事件请求。

- 硬件中断配置，根据需要选择配置 11 条线路作为中断源：
  - ◆ 配置 11 条中断线的屏蔽位 (EXTI\_IMASK)；
  - ◆ 配置所选中断线的触发配置位 (EXTI\_RT\_CFG 和 EXTI\_FT\_CFG)；
  - ◆ 配置对应到外部中断控制器的 NVIC 中断通道的使能和屏蔽位，使 11 条中断线中的请求可以被正确地响应。
- 硬件事件配置，根据需要选择配置 11 条线路作为事件源：
  - ◆ 配置 11 条事件线的屏蔽位 (EXTI\_EMASK)；
  - ◆ 配置所选事件线的触发配置位 (EXTI\_RT\_CFG 和 EXTI\_FT\_CFG)。
- 软件中断/事件配置，根据需要选择配置 11 条线路作为软件中断/事件线：
  - ◆ 配置 11 条中断/事件线屏蔽位 (EXTI\_IMASK,EXTI\_EMASK)；
  - ◆ 配置软件中断事件寄存器的请求位 (EXTI\_SWIE)。

## 5.2.4 EXTI 线路映射

图 5-2 外部中断通用 I/O 映射



配置 GPIO 线上的外部中断/事件，必须先使能 AFIO 时钟。通用 I/O 端口以上图的方式连接到 7 条外部中断/事件线上。另外 4 条 EXTI 线的连接方式如下：

- EXTI line 7: PVD 中断
- EXTI line 8: COMP 中断/唤醒事件
- EXTI line 9: TIM6 中断/唤醒事件

■ EXTI line 10:UART3 唤醒事件

注：使用 COMP 中断和 TIM6 中断时，关联的 EXTI 只能配置成上升沿触发。

## 5.3 EXTI 寄存器

### 5.3.1 EXTI 寄存器总览

表 5-2 EXTI 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	EXTI_EMASK	Reserved																				EMASKx											
0x04	EXTI_IMASK	Reserved																				IMASKx											
0x08	EXTI_FT_CFG	Reserved																				FT_CFGx											
0x0c	EXTI_RT_CFG	Reserved																				RT_CFGx											
0x10	EXTI_PEND	Reserved																				PENDx											
0x14	EXTI_SWIE	Reserved																				SWIEx											

### 5.3.2 EXTI 事件屏蔽寄存器 (EXTI\_EMASK)

偏移地址：0x00

复位值：0x00000000

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

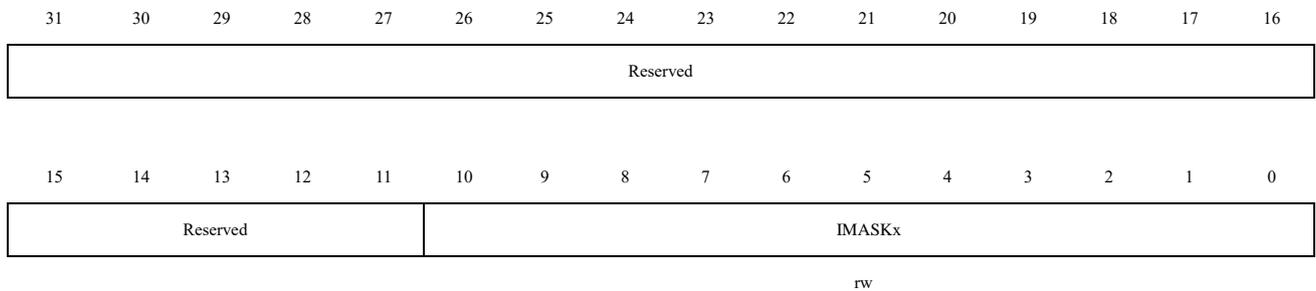
rw

位域	名称	描述
[31:11]	Reserved	保留
[10:0]	EMASKx	EXTI Line x 事件屏蔽 (x=0~10) 0: 事件被屏蔽 1: 事件不被屏蔽

### 5.3.3 EXTI 中断屏蔽寄存器 (EXTI\_IMASK)

偏移地址: 0x04

复位值: 0x00000000

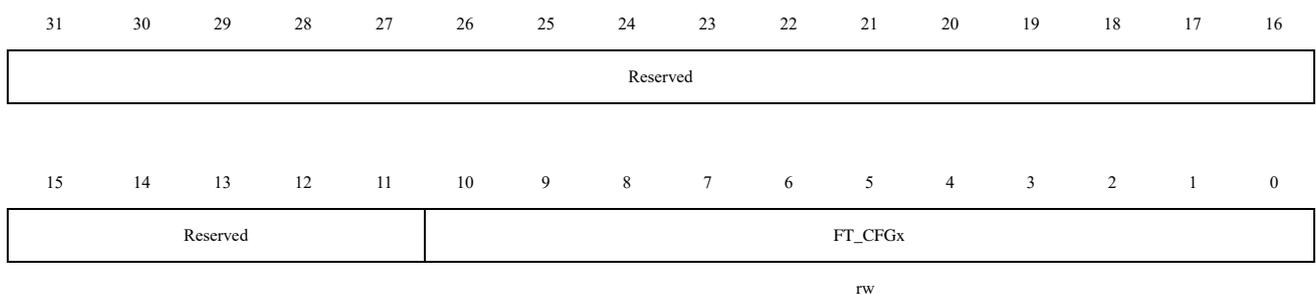


位域	名称	描述
[31:11]	Reserved	保留
[10:0]	IMASKx	EXTI Line x 中断屏蔽 (x=0~10) 0: 中断被屏蔽 1: 中断不被屏蔽

### 5.3.4 EXTI 下降沿触发配置寄存器 (EXTI\_FT\_CFG)

偏移地址: 0x08

复位值: 0x00000000

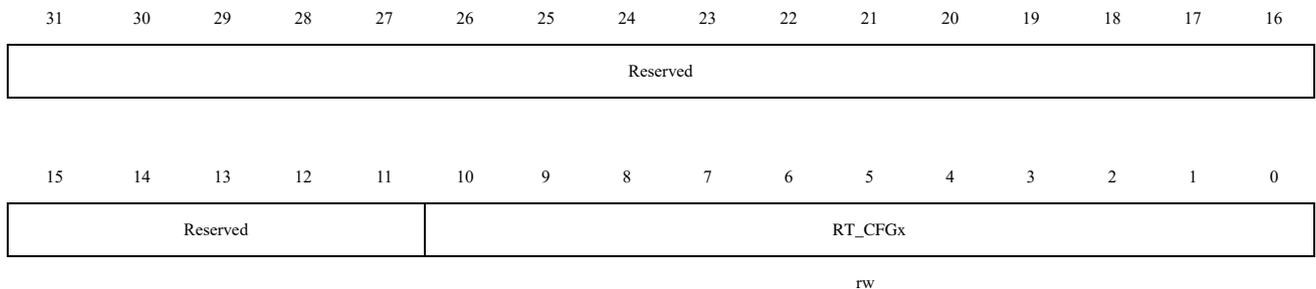


位域	名称	描述
[31:11]	Reserved	保留
[10:0]	FT_CFGx	EXTI Line x 下降沿触发配置 (x=0~10) 0: EXTI Line x 下降沿触发禁能 (事件和中断) 1: EXTI Line x 下降沿触发使能 (事件和中断)

### 5.3.5 EXTI 上升沿触发配置寄存器 (EXTI\_RT\_CFG)

偏移地址: 0x0C

复位值: 0x00000000

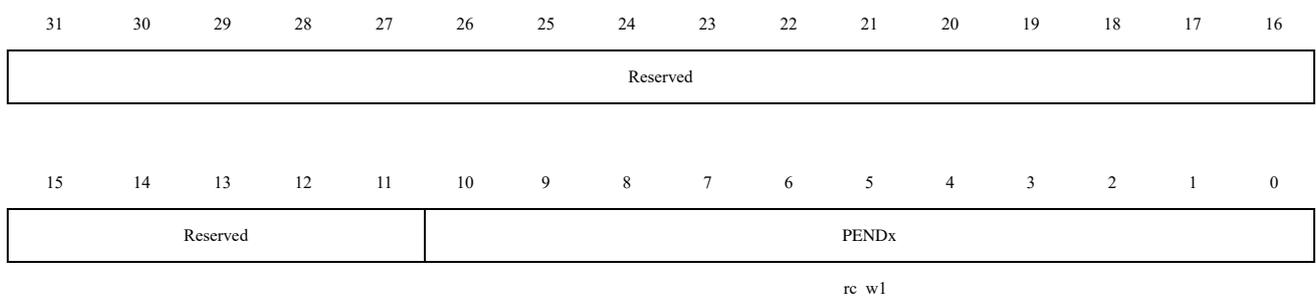


位域	名称	描述
[31:11]	Reserved	保留
[10:0]	RT_CFGx	EXTI Line x 上升沿触发配置 (x=0~10) 0: EXTI Line x 上升沿触发禁能 (事件和中断) 1: EXTI Line x 上升沿触发使能 (事件和中断)

### 5.3.6 EXTI 挂起配置寄存器 (EXTI\_PEND)

偏移地址: 0x10

复位值: 0x00000000

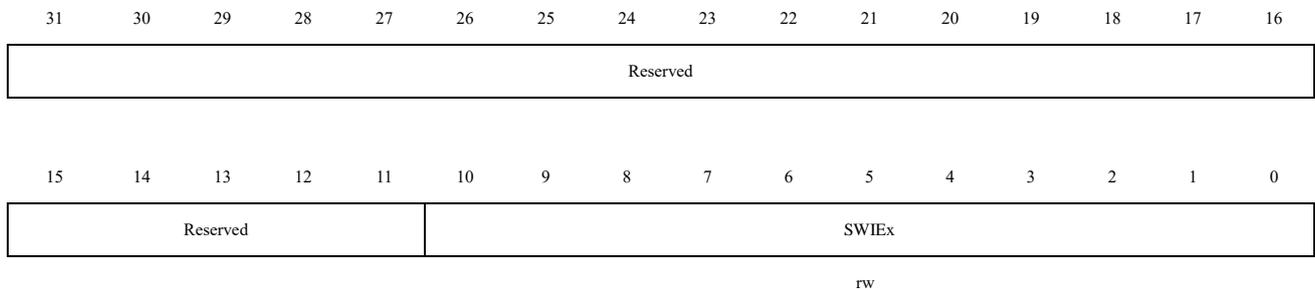


位域	名称	描述
[31:11]	Reserved	保留
[10:0]	PENDx	EXTI Line x 挂起位 (x=0~10) 0: 没有发生挂起请求 1: 发生了挂起触发请求 当外部中断线上发生了选择的边沿触发事件, 该位被置'1'。在该位中写入'1'可以清除它, 也可以通过改变边沿检测的极性清除此位。

### 5.3.7 EXTI 软中断使能寄存器 (EXTI\_SWIE)

偏移地址: 0x14

复位值: 0x00000000



位域	名称	描述
[31:11]	Reserved	保留
[10:0]	SWIE <sub>x</sub>	EXTI Line x 软件触发中断控制 (x=0~10) 通过向相应的 SWIE <sub>x</sub> 写 1 (需确认该位当前为 0), 可以软件触发一个中断, 无需依赖外部硬件信号。 软件中断生成条件: 前提: 对应的 EXTI Line x 必须在 EXTI_IMASK.IMASK <sub>x</sub> 中已经启动 触发: 向相应的 SWIE <sub>x</sub> 写 1, 会设置相应的 EXTI_PEND.PR <sub>x</sub> , 从而生成中断请求 清除: 自动清除, 通过向相应的 EXTI_PEND.PR <sub>x</sub> 写 1, 相应的 SWIE <sub>x</sub> 同步清零

## 6 GPIO 和 AFIO

### 6.1 概述

GPIO (General purpose input/output) 即通用型 I/O, AFIO (Alternate-function input/output) 即复用功能 I/O。芯片最多支持 29 个 GPIO, 共被分为 3 组 (GPIOA/GPIOB/GPIOF), A 组每组 16 个端口, B 组每组 9 个端口, F 组共 4 个。GPIO 端口和其他的复用外设共用引脚, 用户可以根据需求灵活配置。每个 GPIO 引脚都可以独立配置成输出、输入或复用的外设功能端口。除了模拟输入引脚外, 其他的 GPIO 引脚都有大电流通过能力。

GPIO 端口具有以下特性:

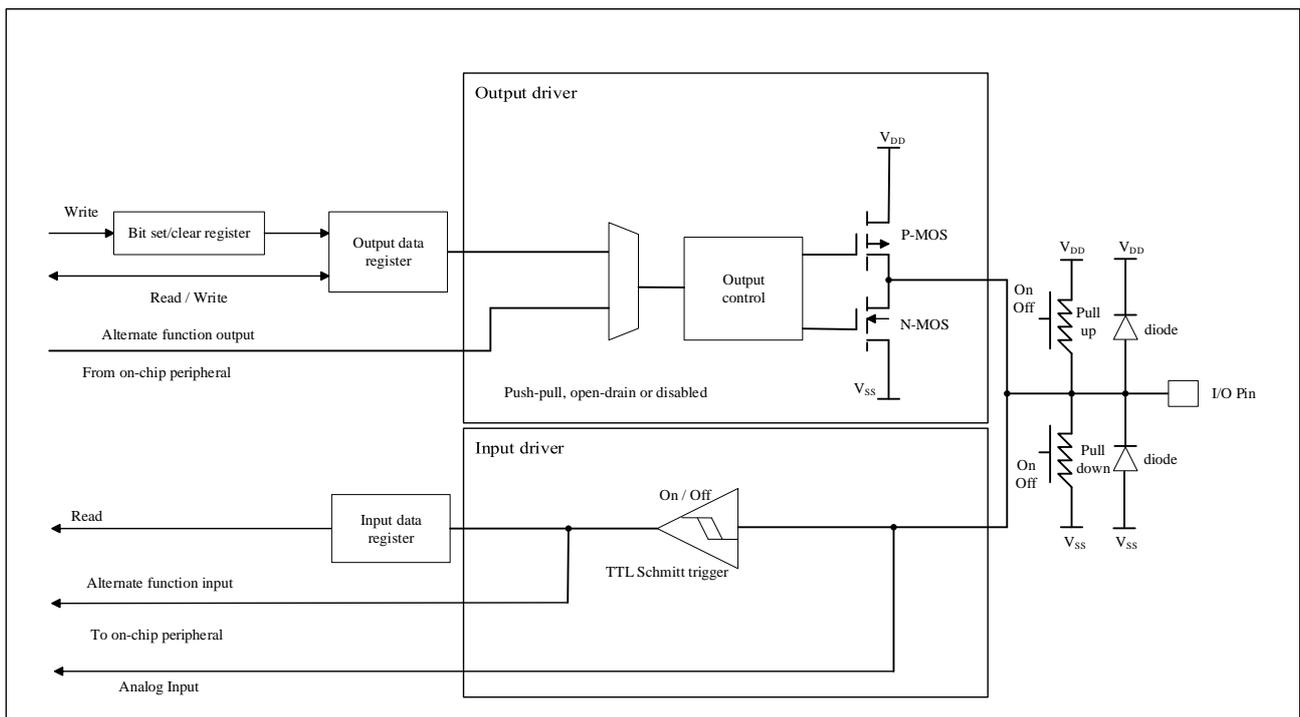
■ GPIO 端口可由软件分别配置成以下模式:

- ◆ 输入浮空
- ◆ 输入上拉
- ◆ 输入下拉

- ◆ 模拟功能
- ◆ 推挽输出及上/下拉可配
- ◆ 推挽复用功能及上/下拉可配
- ◆ 开漏复用功能及上/下拉可配(仅 I2C 相关复用支持)
- 单独的位设置或位清除功能
- 所有 I/O 支持外部中断功能
- 所有 I/O 支持低功耗模式唤醒，上升或下降沿可配置
  - ◆ 所有 EXTI 均可唤醒，所有 GPIO 可配置到 EXTI0~6，可记录 7 组唤醒源（PA0~PA3：第 1 组，PA4~PA7：第 2 组，PA8~PA11：第 3 组，PA12~PA15：第 4 组，PB0~PB3：第 5 组，PB4~PB8：第 6 组，PF0~PF3：第 7 组）
- 支持软件重新映射 I/O 复用功能
- 支持 GPIO 锁定机制，复位方式清除锁定状态

每个 I/O 端口位可以任意编程，但必须按照 32 位字访问 I/O 端口寄存器（不允许 16 位半字或 8 位字节访问）。下图给出了一个 I/O 端口的基本结构。

图 6-1 I/O 端口的基本结构（不支持 Fail-safe）



## 6.2 功能描述

### 6.2.1 IO 模式配置

IO 的模式控制由配置寄存器  $GPIOx\_PMODE$ ， $GPIOx\_POTYPE$  和  $GPIOx\_PUPD$  ( $x=A,B,F$ ) 来设置，不同的操作模式下的配置如下表所示：

**表 6-1 IO 模式和配置关系**

PMODE[1:0]	POTYPE	PUPD[1:0]		I/O 配置
01	0	0	0	通用输出推挽 (Push-Pull)
	0	0	1	通用输出推挽 (Push-Pull) + 上拉
	0	1	0	通用输出推挽 (Push-Pull) + 下拉
	0	1	1	保留
	1	x	x	保留
10	0	0	0	复用功能+推挽 (Push-Pull)
	0	0	1	复用功能+推挽 (Push-Pull) + 上拉
	0	1	0	复用功能+推挽 (Push-Pull) + 下拉
	0	1	1	保留
	1	0	0	复用功能+开漏 (Open-Drain)
	1	0	1	复用功能+开漏 (Open-Drain) + 上拉
	1	1	0	复用功能+开漏 (Open-Drain) + 下拉
	1	1	1	保留
00	x	0	0	浮空输入
	x	0	1	上拉输入
	x	1	0	下拉输入
	x	1	1	保留
11	x	0	0	模拟模式
	x	0	1	保留
	x	1	0	
	x	1	1	

另外 GPIOB\_DS.DS<sub>y</sub> 位可用来配置高/低驱动强度，GPIOB\_SR.SR<sub>y</sub> 位可用来高/低翻转速率的配置。

IO 在不同的配置下的输入输出特性如下表所示：

**表 6-2 IO 不同配置的输入输出特性**

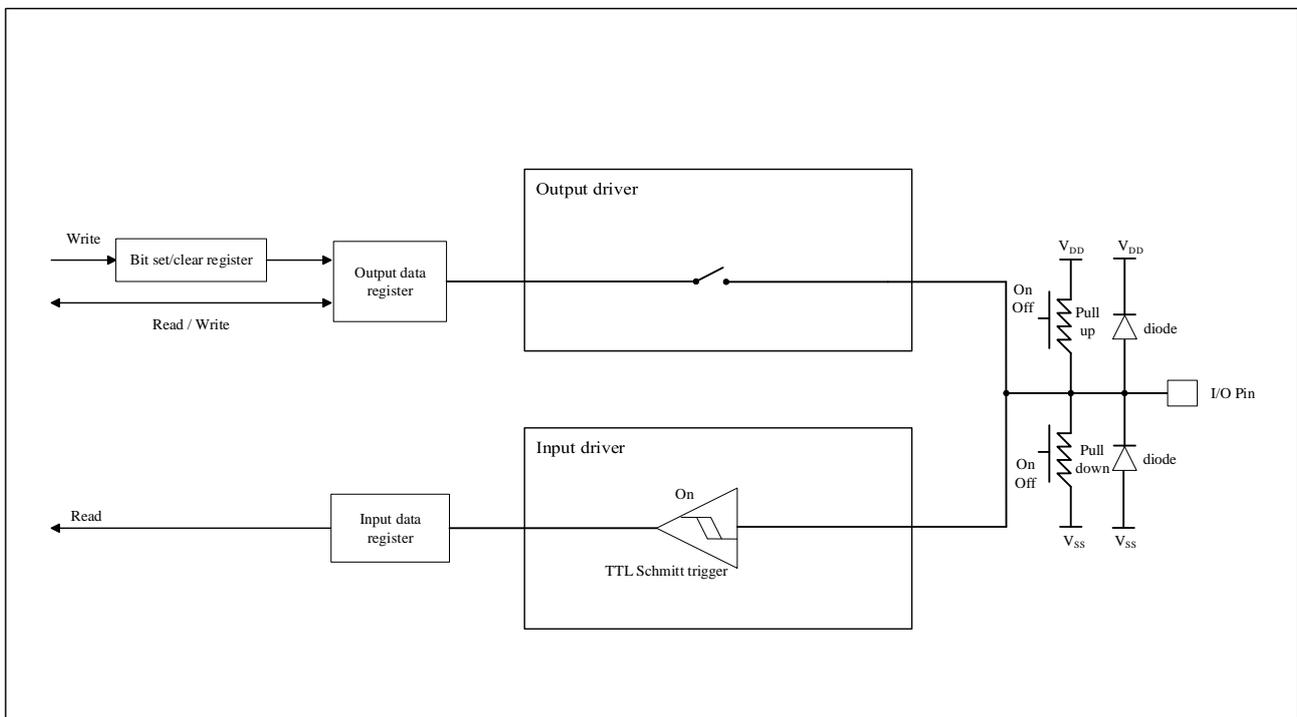
特性	GPIO 输入	GPIO 输出	模拟	外设复用
输出缓冲器	禁能	使能	禁能	根据外设功能配置
施密特触发器	使能	使能	禁能 输出值被强制为 0	使能
上下拉/浮空	可配	可配	禁能	根据外设功能配置
开漏模式	禁能	可配， 输出数据为”0”时 GPIO 输出 0，”1”时 GPIO 高 阻	禁能	可配，输出数据为”0”时 GPIO 输出 0，”1”时 GPIO 高阻
推挽模式	禁能	可配， 输出数据为”0”时 GPIO 输出 0，”1”时 GPIO 输 出 1	禁能	可配，输出数据为”0”时 GPIO 输出 0，”1”时 GPIO 输出 1
输入数据寄存器 (IO 状态)	可读	可读	读出为 0	可读
输出数据寄存器 (输出值)	无效	可读写	无效	可读

### 6.2.1.1 输入模式

当 I/O 端口配置为输入模式时：

- 输出缓冲器被禁止
- 施密特触发输入被激活
- 上拉和下拉电阻是否被连接，取决于 GPIOx\_PUPD 寄存器的配置
- 出现在 I/O 脚上的数据在每个 AHB 时钟被采样到输入数据寄存器
- 对输入数据寄存器的读访问得到 I/O 状态

图 6-2 输入浮空/上拉/下拉模式（不支持 Fail-safe）



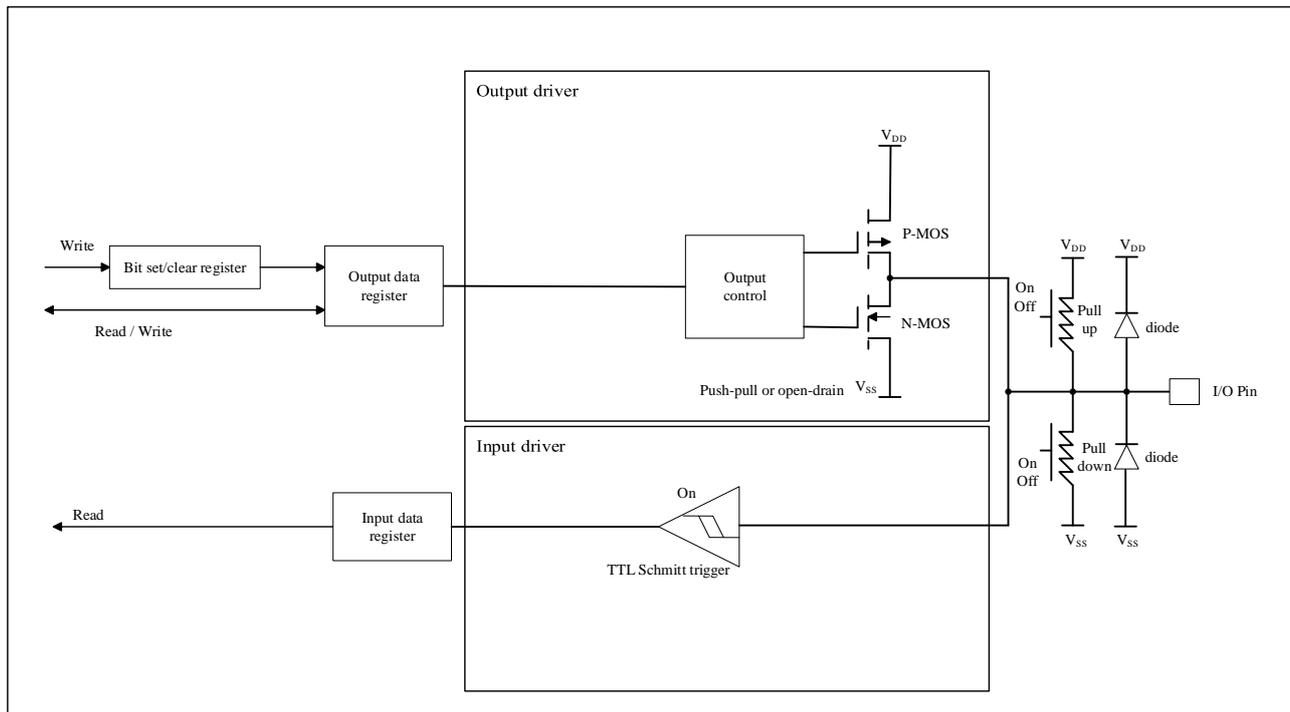
### 6.2.1.2 输出模式

当 I/O 端口配置为输出模式时：

- 施密特触发输入被激活
- 上拉和下拉电阻是否被连接，取决于 GPIOx\_PUPD 寄存器的配置
- 输出缓冲器被激活
  - ◆ 开漏模式(仅 I2C 支持)： 输出数据寄存器上的'0'激活 N-MOS，引脚输出低电平  
输出数据寄存器上的'1'使端口置于高阻状态（P-MOS 从不被激活）
  - ◆ 推挽模式： 输出数据寄存器上的'0'激活 N-MOS，引脚输出低电平  
输出数据寄存器上的'1'激活 P-MOS，引脚输出高电平
- 出现在 I/O 脚上的数据在每个 AHB 时钟被采样到输入数据寄存器
- 对输入数据寄存器的读访问可得到 I/O 状态

- 对输出数据寄存器的读访问得到最后一次写入的值

图 6-3 输出模式（不支持 Fail-safe）

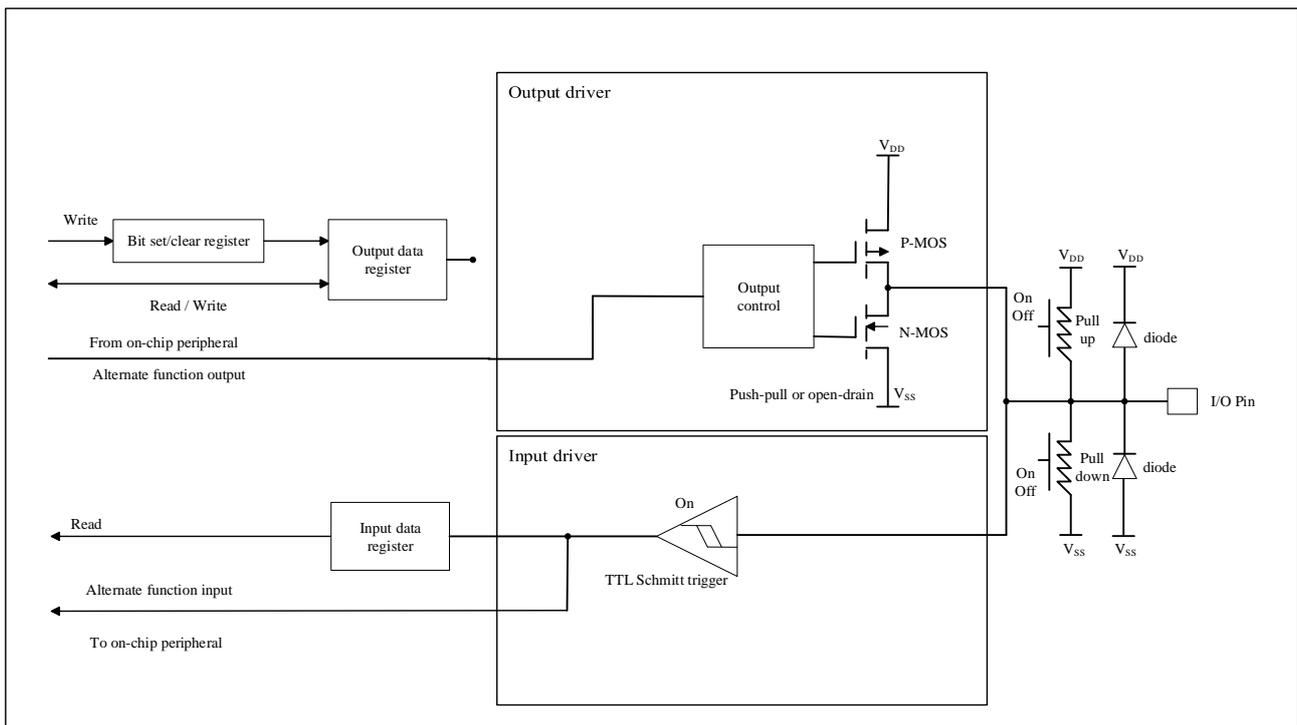


### 6.2.1.3 复用功能模式

当 I/O 端口配置为复用功能模式时：

- 施密特触发输入被激活
- 上拉和下拉电阻是否被连接，取决于 GPIOx\_PUPD 寄存器的配置
- 在开漏或推挽式配置中，输出缓冲器由外设控制
- 内置外设的信号驱动输出缓冲器
- 在每个 AHB 时钟周期，出现在 I/O 脚上的数据被采样到输入数据寄存器
- 对输入数据寄存器的读访问可得到 I/O 状态
- 对输出数据寄存器的读访问得到最后一次写入的值

图 6-4 复用功能模式（不支持 Fail-safe）

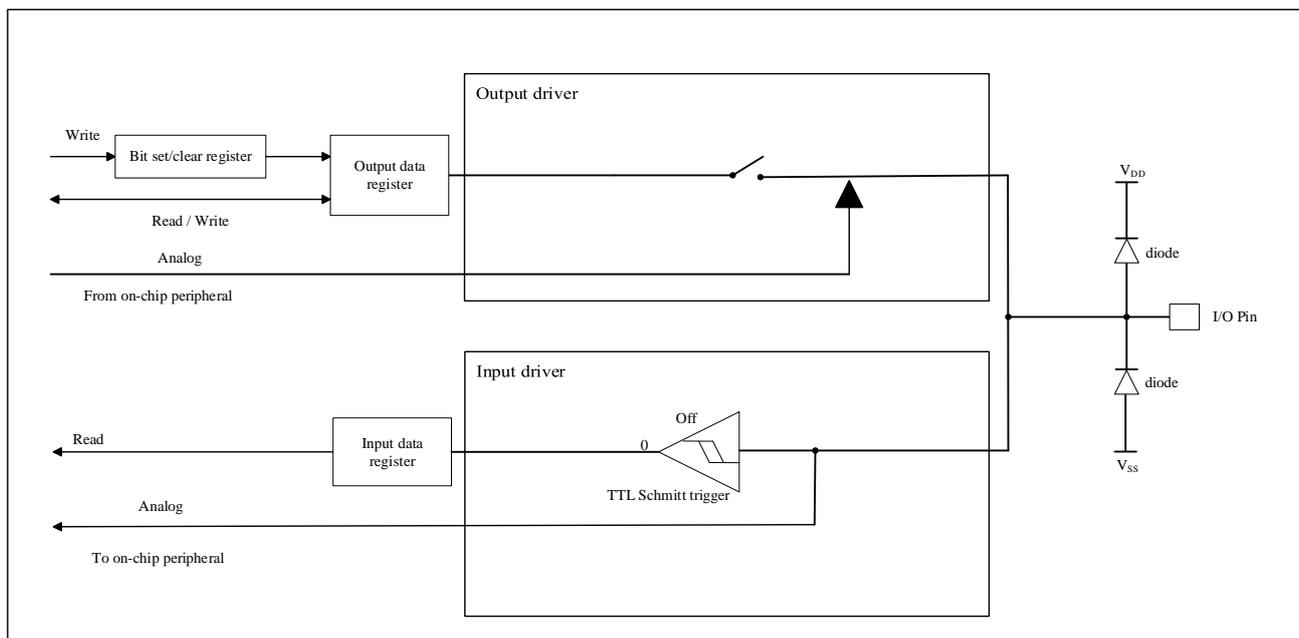


### 6.2.1.4 模拟功能模式

当 I/O 端口被配置为模拟功能模式时：

- 上拉和下拉电阻被禁止
- 读取输入数据寄存器时数值为'0'
- 输出缓存器被禁止
- 施密特触发输入被禁止，输出值被强置为'0'（实现了每个模拟 I/O 引脚上的零消耗）

图 6-5 高阻抗的模拟功能模式（不支持 Fail-safe）



### 6.2.2 复位后状态

复位期间和刚复位后，复用功能未开启，I/O 端口被配置成模拟功能模式（GPIOx\_PMODE.PMODEx[1:0]=2'b11）。但有以下几个例外的信号：

- NRST 默认无 GPIO 功能：
  - ◆ NRST 上拉输入
- 复位后，调试系统相关的引脚默认配置为 SWD 接口 I/O 配置：
  - ◆ PA14: SWCLK 置于输入下拉模式
  - ◆ PA13: SWDIO 置于输入上拉模式
- PF2/BOOT0：
  - ◆ 芯片启动过程中，PF2 作为 BOOT0 功能，默认下拉输入模式，芯片初始化完成后，用作通用 GPIO，被配置成模拟功能模式

### 6.2.3 单独的位设置和位清除

通过对“位设置/清除寄存器（GPIOx\_PBSC）和位清除寄存器（GPIOx\_PBC）”中想要更改的位写‘1’来实现对数据寄存器（GPIOx\_POD）的个别位操作，可以一个或多个位。写‘1’的位被相应的置位或清除，没被写‘1’的位将不被更改。软件不需要禁止中断，在单次 AHB 写操作里完成。

### 6.2.4 外部中断/唤醒线

所有端口都有外部中断能力，可以在 EXTI 模块中配置：

- 端口必须配置成输入模式

- 所有端口可配置用于 STOP 模式唤醒，支持上升或下降沿可配
- 通用 I/O 端口以图 5-2 的方式连接到 7 个外部中断/事件线上，由寄存器 AFIO\_EXTI\_CFGx 配置

## 6.2.5 复用功能

当 I/O 端口被配置为复用功能模式时，使用前必须对端口位配置寄存器（GPIOx\_AFL/ GPIOx\_AFH，GPIOx\_PMODE，GPIOx\_POTYPE 和 GPIOx\_PUPD），复用输入或输出由外设确定。

表 6-3 外部中断通用 I/O 映像

EXTI 线 0 连接到:	PA0、PA1、PA2、PA3
EXTI 线 1 连接到:	PA4、PA5、PA6、PA7
EXTI 线 2 连接到:	PA8、PA9、PA10、PA11
EXTI 线 3 连接到:	PA12、PA13、PA14、PA15
EXTI 线 4 连接到:	PB0、PB1、PB2、PB3
EXTI 线 5 连接到:	PB4、PB5、PB6、PB7、PB8
EXTI 线 6 连接到:	PF0、PF1、PF2、PF3

### 6.2.5.1 软件重新映射 I/O 复用功能

为拓展不同器件封装下的复用外设功能灵活性，可以把一些外设复用功能重新映射到其他引脚上。每个 IO 有多达 16 个可复用的功能 (AF0~ AF15)。复位后，除 PA13 和 PA14 外，AFSELY 默认选择为 AF15。可以通过软件配置相应的寄存器（GPIOx\_AFL/AFH）来重新映射 IO 复用功能。

这时，复用功能就不再映射到它们的原始引脚上(对于外设的 IO 重映射功能，若重映射到不同的引脚，则输入为重映射多选一，输出则接到重映射后的位置，原位置断开)。

#### 6.2.5.1.1 SWD 复用功能 I/O 重映射

表 6-4 SWD 复用功能 I/O 重映射

复用功能	I/O	重映射
SWDIO	PA13	AF0
SWCLK	PA14	AF0

#### 6.2.5.1.2 TIMx 复用功能 I/O 重映射

##### 6.2.5.1.2.1 TIM1 复用功能 I/O 重映射

表 6-5 TIM1 复用功能 I/O 重映射

复用功能	IO	重映射
TIM1_ETR	PA12	AF3
TIM1_BKIN1	PA2	AF2
TIM1_BKIN2	PA6	AF3
TIM1_BKIN3	PA11	AF1
TIM1_BKIN4	PA12	AF1
TIM1_CH1	PA15	AF1
	PA4	AF3
	PA8	AF3

	PB3	AF1
	PB4	AF1
	PB5	AF1
	PB6	AF1
	PB7	AF1
	PB8	AF1
	PF2	AF1
TIM1_CH2	PA3	AF3
	PA9	AF3
	PA15	AF7
	PB3	AF7
	PB4	AF7
	PB5	AF7
	PB6	AF7
	PB7	AF7
	PB8	AF7
	PF2	AF7
TIM1_CH3	PA5	AF3
	PA10	AF3
	PA15	AF9
	PB3	AF9
	PB4	AF9
	PB5	AF9
	PB6	AF9
	PB7	AF9
	PB8	AF9
	PF2	AF9
TIM1_CH4	PA11	AF3
	PA15	AF11
	PB3	AF11
	PB4	AF11
	PB5	AF11
	PB6	AF11
	PB7	AF11
	PB8	AF11
	PF2	AF11
TIM1_CH5	PA0	AF3
TIM1_CH6	PA1	AF3
TIM1_CH7	PA2	AF3
TIM1_CH1N	PA7	AF3
	PA15	AF5
	PB3	AF5
	PB4	AF5

	PB5	AF5
	PB6	AF5
	PB7	AF5
	PB8	AF5
	PF2	AF5
TIM1_CH2N	PA5	AF1
	PA15	AF8
	PB0	AF1
	PB3	AF8
	PB4	AF8
	PB5	AF8
	PB6	AF8
	PB7	AF8
	PB8	AF8
	PF2	AF8
TIM1_CH3N	PA15	AF10
	PB1	AF1
	PB3	AF10
	PB4	AF10
	PB5	AF10
	PB6	AF10
	PB7	AF10
	PB8	AF10
	PF2	AF10
TIM1_CH4N	PA15	AF12
	PB3	AF12
	PB4	AF12
	PB5	AF12
	PB6	AF12
	PB7	AF12
	PB8	AF12
	PF2	AF12

### 6.2.5.1.2.2 TIM3 复用功能 I/O 重映射

表 6-6 TIM3 复用功能 I/O 重映射

复用功能	IO	重映射
TIM3_ETR	PB3	AF2
	PF0	AF11
	PA1	AF10
TIM3_CH1	PA0	AF11
	PA4	AF11
	PA6	AF11
	PA14	AF11

复用功能	IO	重映射
	PB4	AF2
	PF0	AF0
TIM3_CH2	PA1	AF11
	PA7	AF11
	PA13	AF11
	PB5	AF2
	PF1	AF0
TIM3_CH3	PA2	AF11
	PB6	AF2
	PB1	AF10
	PB0	AF11
	PF3	AF0
TIM3_CH4	PB7	AF2
	PB8	AF2
	PB2	AF10
	PB1	AF11

### 6.2.5.1.2.3 TIM4 复用功能 I/O 重映射

表 6-7 TIM4 复用功能 I/O 重映射

复用功能	IO	重映射
TIM4_ETR	PF1	AF12
	PF3	AF12
TIM4_CH1	PA15	AF6
	PA0	AF12
	PA14	AF12
TIM4_CH2	PA1	AF12
	PA13	AF12
	PB3	AF6
TIM4_CH3	PA2	AF12

### 6.2.5.1.3 UARTx 复用功能 I/O 重映射

#### 6.2.5.1.3.1 UART1 复用功能 I/O 重映射

表 6-8 UART1 复用功能 I/O 重映射

复用功能	IO	重映射
UART1_TX	PA1	AF1
	PA2	AF1
	PA9	AF4
	PA14	AF4
	PB6	AF4
	PF0	AF2
UART1_RX	PA0	AF1

复用功能	IO	重映射
	PA3	AF1
	PA10	AF4
	PA13	AF4
	PA15	AF4
	PB7	AF4
	PF1	AF2
UART1_DE	PA4	AF1
	PA8	AF4
	PF3	AF2

### 6.2.5.1.3.2 UART2 复用功能 I/O 重映射

表 6-9 UART2 复用功能 I/O 重映射

复用功能	IO	重映射
UART2_TX	PA14	AF1
	PA1	AF4
	PA2	AF4
	PA9	AF10
	PF0	AF3
UART2_RX	PA13	AF1
	PA15	AF3
	PA0	AF4
	PA3	AF4
	PA10	AF10
	PF1	AF3
UART2_DE	PA4	AF4
	PA8	AF10
	PF3	AF3

### 6.2.5.1.3.3 UART3 复用功能 I/O 重映射

表 6-10 UART3 复用功能 I/O 重映射

复用功能	IO	重映射
UART3_TX	PA1	AF6
	PA2	AF6
	PA4	AF6
	PA6	AF6
	PA14	AF6
	PB3	AF4
	PB5	AF4
	PF0	AF4
UART3_RX	PA0	AF6
	PA3	AF6
	PA7	AF6

复用功能	IO	重映射
	PA13	AF6
	PB7	AF3
	PB4	AF4
	PF1	AF4
UART3_DE	PF3	AF4

#### 6.2.5.1.4 I2Cx 复用功能 I/O 重映射

##### 6.2.5.1.4.1 I2C1 复用功能 I/O 重映射

表 6-11 I2C1 复用功能 I/O 重映射

复用功能	IO	重映射
I2C1_SCL	PA9	AF6
	PA4	AF7
	PB6	AF6
	PB8	AF6
	PF1	AF1
I2C1_SDA	PA10	AF6
	PA5	AF7
	PA13	AF7
	PB7	AF6
	PF0	AF1
I2C1_SMBA	PA1	AF7
	PA14	AF7
	PB2	AF6
	PB5	AF6

##### 6.2.5.1.4.2 I2C2 复用功能 I/O 重映射

表 6-12 I2C2 复用功能 I/O 重映射

复用功能	IO	重映射
I2C2_SCL	PA6	AF7
	PA9	AF7
	PA11	AF7
I2C2_SDA	PA7	AF7
	PA10	AF7
	PA12	AF7
I2C2_SMBA	PB2	AF7

### 6.2.5.1.5 SPIx 复用功能 I/O 重映射

#### 6.2.5.1.5.1 SPI1 复用功能 I/O 重映射

表 6-13 SPI1 管脚重映射

复用功能	IO	重映射
SPI1_NSS	PA1	AF0
	PA4	AF0
	PA15	AF0
SPI1_SCK	PA0	AF0
	PA5	AF0
	PA13	AF5
	PB3	AF0
SPI1_MISO	PA3	AF0
	PA6	AF0
	PA1	AF5
	PA4	AF5
	PA14	AF5
	PB4	AF0
SPI1_MOSI	PA2	AF0
	PA7	AF0
	PA5	AF5
	PB1	AF0
	PB5	AF0

#### 6.2.5.1.5.2 SPI2 复用功能 I/O 重映射

表 6-14 SPI2 管脚重映射

复用功能	IO	重映射
SPI2_NSS	PA8	AF0
	PA7	AF5
SPI2_SCK	PA9	AF0
	PB0	AF0
SPI2_MISO	PA10	AF0
	PA12	AF0
SPI2_MOSI	PA11	AF0
	PB1	AF5

#### 6.2.5.1.6 COMPx 复用功能 I/O 重映射

表 6-15 COMPx 复用功能 I/O 重映射

复用功能	IO	重映射
COMP_OUT	PA0	AF8
	PA6	AF8
	PA11	AF8

复用功能	IO	重映射
	PA12	AF8

### 6.2.5.1.7 EVENT\_OUT 复用功能 I/O 重映射

表 6-16 EVENTOUT 复用功能 I/O 重映射

复用功能	IO	重映射
EVENT_OUT	PA1	AF9
	PA6	AF9
	PA12	AF9
	PB4	AF3

### 6.2.5.1.8 MCO 复用功能 I/O 重映射

#### 6.2.5.1.8.1 MCO 复用功能 I/O 重映射

表 6-17 RCC 复用功能 I/O 重映射

复用功能	IO	重映射
MCO	PA8	AF5
	PA9	AF5

### 6.2.5.1.9 ADC 外部触发复用功能重映射

ADC 规则转换的外部触发源支持所有 GPIOs。

## 6.2.6 外设的 IO 配置

表 6-18 ADC

ADC	GPIO 配置
ADC	模拟模式

表 6-19 TIM1/3/4

TIM1 引脚	配置	GPIO 配置模式
TIM1_CHx	输入捕获通道 x	推挽复用
	输出比较通道 x	推挽复用
TIM1_CHxN	互补输出通道 x	推挽复用
TIM1_BKIN	刹车输入	推挽复用
TIM1_ETR	外部触发时钟输入	推挽复用

表 6-20 UART

UART 引脚	配置	GPIO 配置
UARTx_TX	全双工模式	推挽复用
	半双工模式	推挽复用+上拉
UARTx_RX	全双工模式	推挽复用
	半双工模式	未用，可作为通用 I/O
UARTx_DE	全双工模式	推挽复用
	半双工模式	推挽复用+上拉

**表 6-21 I2C**

I2C 引脚	配置	GPIO 配置
I2Cx_SCL	I2C 时钟	开漏复用
I2Cx_SDA	I2C 数据	开漏复用

**表 6-22 SPI**

SPI 引脚	配置	GPIO 配置
SPIx_SCK	主模式	推挽复用
	从模式	推挽复用
SPIx_MOSI	全双工模式/主模式	推挽复用
	全双工模式/从模式	推挽复用
	单线双向数据线/主模式	推挽复用
	单线双向数据线/从模式	未用, 可作为通用 I/O
SPIx_MISO	全双工模式/主模式	推挽复用
	全双工模式/从模式	推挽复用
	单线双向数据线/主模式	未用, 可作为通用 I/O
	单线双向数据线/从模式	推挽复用
SPIx_NSS	硬件主/从模式	推挽复用
	软件模式	未用, 可作为通用 I/O

**表 6-23 COMP**

COMPx 引脚	配置	GPIO 配置
COMPx_OUT	COMP 输出	推挽复用
COMPx_INM	COMP 负端输入	模拟模式
COMPx_INP	COMP 正端输入	模拟模式

**表 6-24 其他**

引脚	复用功能	GPIO 配置
MCO	时钟输出	推挽复用
EXTI 输入线	外部中断输入	浮空输入+上/下拉

## 6.2.7 GPIO 锁定机制

锁定机制用于冻结 IO 配置以防止被意外更改。当在一个端口位上执行了锁定 (LOCK) 程序, 在下次复位之前, 不能再更改端口的配置, 参考端口配置锁定寄存器 GPIOx\_PLOCK。

- 只有在对 GPIOx\_PLOCK.PLOCKK 按照正确的序列 w1->w0->w1->r0 (此处 r0 必须有) 操作之后, 才会变为 1; 之后只有在进行系统复位才会变为 0。
- 只有在 GPIOx\_PLOCK.PLOCKK = 0 也就是未锁定的时候才能进行 GPIOx\_PLOCK.PLOCK[15:0]修改。
- GPIOx\_PLOCK.PLOCK 只有在和非 0 的 GPIOx\_PLOCK.PLOCK[15:0]同时写入的情况下, 序列 w1->w0->w1->r0 才会有效; 序列写入的过程中, GPIOx\_PLOCK.PLOCK[15:0]必须不能改变;
- 只要 GPIOx\_PLOCK.PLOCKK = 0, GPIOx\_PMODE / GPIOx\_POTYPE / GPIOx\_PUPD / GPIOx\_AFL / GPIOx\_AFH 的位都能修改, 不受 GPIOx\_PLOCK.PLOCK[15:0]配置的影响。

- GPIOx\_PLOCK.PLOCKK= 1,GPIOx\_PMODE/GPIOx\_POTYPE/GPIOx\_PUPD/GPIOx\_AFL/GPIOx\_AFH 受 GPIOx\_PLOCK.PLOCK[15:0]的控制。对应 GPIOx\_PLOCK.PLOCKy (y = 0...15) =1 时, 为锁定配置, 不可修改; PLOCKy = 0, 可以修改。
- 假如序列操作错误, 必须重新进行 w1->w0->w1->r0 才能再次发起锁定操作。

## 6.3 GPIO 寄存器

必须以 32 位字的方式操作这些外设寄存器。

### 6.3.1 GPIO 寄存器总览

GPIO 基地址: 0x40023000

表 1-25 GPIO 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	GPIOA_PMODE	PMODE15		PMODE14		PMODE13		PMODE12		PMODE11		PMODE10		PMODE9		PMODE8		PMODE7		PMODE6		PMODE5		PMODE4		PMODE3		PMODE2		PMODE1		PMODE0		
0x100	GPIOB_PMODE	Reserved														PMODE8	PMODE7	PMODE6	PMODE6	PMODE5	PMODE4	PMODE3	PMODE2	PMODE1	PMODE0									
0x200	GPIOF_PMODE	Reserved																				PMODE3	PMODE2	PMODE1	PMODE0									
0x04	GPIOA_POTYPE	Reserved															POT14	POT13	POT12	POT11	POT10	POT9	Reserved	POT7	POT6	POT5	POT4	Reserved	POT1	Reserved				
0x104	GPIOB_POTYPE	Reserved																				POT8	POT7	POT6	POT5	Reserved	POT2	Reserved						
0x204	GPIOF_POTYPE	Reserved																															POT1	POT0
0x0C	GPIOA_PUPD	PUPD15		PUPD14		PUPD13		PUPD12		PUPD11		PUPD10		PUPD9		PUPD8		PUPD7		PUPD6		PUPD5		PUPD4		PUPD3		PUPD2		PUPD1		PUPD0		
0x10C	GPIOB_PUPD	Reserved														PUPD8	PUPD7	PUPD6	PUPD6	PUPD5	PUPD4	PUPD3	PUPD2	PUPD1	PUPD0									
0x20C	GPIOF_PUPD	Reserved																				PUPD3	PUPD2	PUPD1	PUPD0									
0x10	GPIOA_PID	Reserved															PID15	PID14	PID13	PID12	PID11	PID10	PID9	PID8	PID7	PID6	PID5	PID4	PID3	PID2	PID1	PID0		
0x110	GPIOB_PID	Reserved																					PID8	PID7	PID6	PID5	PID4	PID3	PID2	PID1	PID0			
0x210	GPIOF_PID	Reserved																														PID0		
0x14	GPIOA_POD	Reserved															POD1	POD1	POD1	POD1	POD1	POD1	POD9	POD8	POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0		
0x114	GPIOB_POD	Reserved																					POD	POD	POD	POD	POD	POD	POD	POD	POD			
0x214	GPIOF_POD	Reserved																														POD		
0x18	GPIOA_PBS	PBS1	PBS1	PBS1	PBS1	PBS1	PBS9	PBS8	PBS7	PBS6	PBS5	PBS4	PBS3	PBS2	PBS1	PBS0	PBS15	PBS14	PBS13	PBS12	PBS11	PBS10	PBS9	PBS8	PBS7	PBS6	PBS5	PBS4	PBS3	PBS2	PBS1	PBS0		
0x118	GPIOB_PBS	Reserved							PBS	PBS	PBS	PBS	PBS	PBS	PBS	PBS	PBS	PBS	Reserved															
0x218	GPIOF_PBS	Reserved														PBS	PBS	PBS	PBS	PBS	PBS	PBS	PBS	PBS	PBS	PBS	PBS	PBS	PBS	PBS	PBS	PBS		
0x1C	GPIOA_PLOCK	Reserved														PLOCK	PLOCK1	PLOCK1	PLOCK1	PLOCK1	PLOCK1	PLOCK1	PLOCK9	PLOCK8	PLOCK7	PLOCK6	PLOCK5	PLOCK4	PLOCK3	PLOCK2	PLOCK1	PLOCK0		

0x11C	GPIOB_PLOCK	Reserved				PLOCK	Reserved				PLOCK	PLOCK	PLOCK	PLOCK	PLOCK	PLOCK	PLOCK	PLOCK
0x21C	GPIOF_PLOCK	Reserved				PLOCK	Reserved				PLOCK	PLOCK	PLOCK	PLOCK	PLOCK	PLOCK	PLOCK	PLOCK
0x20	GPIOA_AFL	AFSEL7	AFSEL6	AFSEL5	AFSEL4	AFSEL3	AFSEL2	AFSEL1	AFSEL0									
0x120	GPIOB_AFL	AFSEL7	AFSEL6	AFSEL5	AFSEL4	AFSEL3	AFSEL2	AFSEL1	AFSEL0									
0x220	GPIOF_AFL	Reserved				AFSEL3	AFSEL2	AFSEL1	AFSEL0									
0x24	GPIOA_AFH	AFSEL15	AFSEL14	AFSEL13	AFSEL12	AFSEL11	AFSEL10	AFSEL9	AFSEL8									
0x124	GPIOA_AFH	Reserved							AFSEL8									
0x28	GPIOA_PBC	Reserved				PBC1	PBC1	PBC1	PBC1	PBC1	PBC1	PBC9	PBC8	PBC7	PBC6	PBC5	PBC4	PBC3
0x128	GPIOB_PBC	Reserved				Reserved				PBC	PBC	PBC	PBC	PBC	PBC	PBC	PBC	PBC
0x228	GPIOF_PBC	Reserved				Reserved				PBC	PBC	PBC	PBC	PBC	PBC	PBC	PBC	PBC
0x012C	GPIOB_DS	Reserved					SR5	SR4	SR3	Reserved	DSR5	DSR4	DSR3	Reserved				

### 6.3.2 GPIO 端口模式寄存器 (GPIOx\_PMODE)

偏移地址: 0x0000 (x=A); 0x0100 (x=B); 0x0200 (x=F);

复位值: 0xCBFF FFFF (x=A); 0x3 FFFF (x=B); 0xFF (x=F);

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PMODE15[1:0]		PMODE14[1:0]		PMODE13[1:0]		PMODE12[1:0]		PMODE11[1:0]		PMODE10[1:0]		PMODE9[1:0]		PMODE8[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
PMODE7[1:0]		PMODE6[1:0]		PMODE5[1:0]		PMODE4[1:0]		PMODE3[1:0]		PMODE2[1:0]		PMODE1[1:0]		PMODE0[1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	

位域	名称	描述
31:30	PMODEy [1:0]	端口 GPIOx (x = A,B,F) 引脚 PINy 的模式: 00: 输入模式 01: 通用输出模式 10: 复用功能模式 11: 模拟功能模式 <i>注: x = A 时, y = 0...15; x = B 时, y = 0...8; x = F 时, y = 0...3; 其余位为保留, 保留位为只读;</i>
29:28		
27:26		
25:24		
23:22		
21:20		
19:18		
17:16		
15:14		
13:12		
11:10		
9:8		
7:6		

位域	名称	描述
5:4 3:2 1:0		

### 6.3.3 GPIO 端口输出类型寄存器 (GPIOx\_POTYPE)

偏移地址: 0x0004 (x=A); 0x0104 (x=B); 0x0204 (x=F);

复位值: 0x0000 0000 (x=A,B,F)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POT15	POT14	POT13	POT12	POT11	POT10	POT9	POT8	POT7	POT6	POT5	POT4	POT3	POT2	POT1	POT0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:16	Reserved	保留, 必须保持复位值。
15:0	POTy	端口 GPIOx (x = A,B,F) 引脚 PINy 的输出类型: 0: 输出推挽模式 (复位后的状态) 1: 输出开漏模式 (仅 I2C 复用功能时可配置开漏模式) 注: x = A 时, y = 1,4,5,6,7,9,10,11,12,13,14; x = B 时, y = 2,5,6,7,8; x = F 时, y = 0,1; 其余位为保留, 保留位为只读。

### 6.3.4 GPIO 端口上下拉寄存器 (GPIOx\_PUPD)

偏移地址: 0x000C (x=A); 0x010C (x=B); 0x020C (x=F);

复位值: 0x2400 0000 (x=A); 0x0000 0000 (x=B,F)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUPD15[1:0]		PUPD14[1:0]		PUPD13[1:0]		PUPD12[1:0]		PUPD11[1:0]		PUPD10[1:0]		PUPD9[1:0]		PUPD8[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
PUPD7[1:0]		PUPD6[1:0]		PUPD5[1:0]		PUPD4[1:0]		PUPD3[1:0]		PUPD2[1:0]		PUPD1[1:0]		PUPD0[1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	

位域	名称	描述
31:30 29:28 27:26 25:24 23:22 21:20 19:18 17:16 15:14 13:12 11:10 9:8 7:6 5:4 3:2 1:0	PUPDy[1:0]	端口 GPIOx (x = A,B,F) 引脚 PINy 的上拉下拉模式: 00: 无上/下拉 01: 上拉 10: 下拉 11: 保留 注: x = A 时, y = 0...15; x = B 时, y = 0...8; x = F 时, y = 0...3; 其余位为保留, 保留位为只读;

### 6.3.5 GPIO 端口输入数据寄存器 (GPIOx\_PID)

偏移地址: 0x0010 (x=A); 0x0110 (x=B); 0x0210 (x=F);

复位值: 0x0000 0000 (x= A,B,F)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PID15	PID14	PID13	PID12	PID11	PID10	PID9	PID8	PID7	PID6	PID5	PID4	PID3	PID2	PID1	PID0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位域	名称	描述
31:16	Reserved	保留, 必须保持复位值。
15:0	PIDy	端口 GPIOx (x = A,B,F) 引脚 PINy 的输入数据 这些位为只读, 读出的值为对应 I/O 口的状态。 注: x = A 时, y = 0...15; x = B 时, y = 0...8; x = F 时, y = 0...3; 其余位为保留, 保留位为只读;

### 6.3.6 GPIO 端口输出数据寄存器 (GPIOx\_POD)

偏移地址: 0x0014 (x=A); 0x0114 (x=B); 0x0214 (x=F);

复位值：0x0000 0000 (x=A,B,F)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8	POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:16	Reserved	保留，必须保持复位值。
15:0	PODy	端口 GPIOx (x = A,B,F) 引脚 PINy 的输出数据 这些位可由软件读出或写入，可对相应的 POD 位进行独立的设置/清除。 注：x = A 时，y = 0...15； x = B 时，y = 0...8； x = F 时，y = 0...3； 其余位为保留，保留位为只读；

### 6.3.7 GPIO 端口位设置/清除寄存器 (GPIOx\_PBSC)

偏移地址：0x0018 (x=A)；0x0118 (x=B)；0x0218 (x=F)；

复位值：0x0000 0000 (x=A,B,F)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PBC15	PBC14	PBC13	PBC12	PBC11	PBC10	PBC9	PBC8	PBC7	PBC6	PBC5	PBC4	PBC3	PBC2	PBC1	PBC0
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
PBS15	PBS14	PBS13	PBS12	PBS11	PBS10	PBS9	PBS8	PBS7	PBS6	PBS5	PBS4	PBS3	PBS2	PBS1	PBS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位域	名称	描述
31:16	PBCy	清除端口 GPIOx (x = A,B,F) 的位 y 这些位只能写入。 0：对相应的 PODy 位不产生影响 1：清除对应的 PODy 位为 0 注：如果同时设置了 PBSy 和 PBCy 的对应位，PBSy 位起作用。 注：x = A 时，y = 0...15； x = B 时，y = 0...8； x = F 时，y = 0...3； 其余位为保留，保留位为只读；

位域	名称	描述
15:0	PBSy	设置端口 GPIOx (x = A,B,F) 的位 y 这些位只能写入。 0: 对相应的 PODy 位不产生影响 1: 设置对应的 PODy 位为 1 注: x = A 时, y = 0...15; x = B 时, y = 0...8; x = F 时, y = 0...3; 其余位为保留, 保留位为只读;

### 6.3.8 GPIO 端口锁定置寄存器 (GPIOx\_PLOCK)

偏移地址: 0x001C (x=A); 0x011C (x=B); 0x021C (x=F);

复位值: 0x0000 0000 (x = A,B,F)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															PLOCKK
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLOCK15	PLOCK14	PLOCK13	PLOCK12	PLOCK11	PLOCK10	PLOCK9	PLOCK8	PLOCK7	PLOCK6	PLOCK5	PLOCK4	PLOCK3	PLOCK2	PLOCK1	PLOCK0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:17	Reserved	保留, 必须保持复位值。
16	PLOCKK	锁键 该位可随时读出, 它只可通过锁键写入序列修改。 0: 端口配置锁键位未激活 1: 端口配置锁键位被激活, 下次系统复位前 GPIOx_PLOCK 寄存器被锁住。 锁键写入序列: 写 1 -> 写 0 -> 写 1 -> 读 0 -> (读 1) 最后一个读 1 可省略, 但可以用来确认锁键已被激活。 注: 在操作锁键写入序列时, 不能改变 PLOCK[15:0] 的值。操作锁键写入序列中的任何错误将不能激活锁键。
15:0	PLOCKy	端口 GPIOx (x = A,B,F) 引脚 PINy 的配置锁定位 这些位可读可写但只能在 PLOCKK 位为 0 时写入。 0: 不锁定端口的配置 1: 锁定端口的配置 注: x = A 时, y = 0...15; x = B 时, y = 0...8; x = F 时, y = 0...3; 其余位为保留, 保留位为只读;

### 6.3.9 GPIO 复用功能低配置寄存器 (GPIOx\_AFL)

偏移地址: 0x0020 (x=A); 0x0120 (x=B); 0x0220 (x=F);

复位值: 0xFFFF FFFF (x=A,B), 0x FFFF (x=F)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFSEL7[3:0]				AFSEL6[3:0]				AFSEL5[3:0]				AFSEL4[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFSEL3[3:0]				AFSEL2[3:0]				AFSEL1[3:0]				AFSEL0[3:0]			
rw				rw				rw				rw			

位域	名称	描述
31:28	AFSELY[3:0]	端口 GPIOx (x = A,B,F) 引脚 PINy (y = 0...7) 的复用功能配置位
27:24		0000: AF0
23:20		0001: AF1
19:16		0010: AF2
15:12		0011: AF3
11:8		0100: AF4
7:4		0101: AF5
3:0		0110: AF6
		0111: AF7
		1000: AF8
		1001: AF9
		1010: AF10
		1011: AF11
		1100: AF12
		1101: AF13
	1110: AF14	
	1111: AF15	
	注: x = A 时, y = 0...7;	
	x = B 时, y = 0...7;	
	x = F 时, y = 0...3;	
	其余位为保留, 保留位为只读;	
	注: AF15 是 GPIO, 没有复用功能	

### 6.3.10 GPIO 复用功能高配置寄存器 (GPIOx\_AFH)

偏移地址: 0x0024 (x=A); 0x0124 (x=B);

复位值: 0xF00F FFFF (x=A), 0xF (x=B)

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

AFSEL15[3:0]				AFSEL14[3:0]				AFSEL13[3:0]				AFSEL12[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFSEL11[3:0]				AFSEL10[3:0]				AFSEL9[3:0]				AFSEL8[3:0]			
rw				rw				rw				rw			

位域	名称	描述	
31:28	AFSELY[3:0]	端口 GPIOx (x = A,B) 引脚 PINy (y = 8...15) 的复用功能配置位	
27:24			0000: AF0
23:20			0001: AF1
19:16			0010: AF2
15:12			0011: AF3
11:8			0100: AF4
7:4			0101: AF5
3:0			0110: AF6
			0111: AF7
			1000: AF8
			1001: AF9
			1010: AF10
			1011: AF11
			1100: AF12
			1101: AF13
	1110: AF14		
	1111: AF15		
	注: x = A 时, y = 8...15;		
	x = B 时, y = 8;		
	其余位为保留, 保留位为只读;		
	注: AF15 是 GPIO, 没有复用功能		

### 6.3.11 GPIO 端口位清除寄存器 (GPIOx\_PBC)

偏移地址: 0x0028 (x=A); 0x0128 (x=B); 0x0228 (x=F);

复位值: 0x0000 0000 (x = A,B,F)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PBC15	PBC14	PBC13	PBC12	PBC11	PBC10	PBC9	PBC8	PBC7	PBC6	PBC5	PBC4	PBC3	PBC2	PBC1	PBC0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位域	名称	描述
31:16	Reserved	保留，必须保持复位值。
15:0	PBCy	清除端口 GPIOx 的位 y (y = 0...15) 这些位只能写入。 0: 对相应的 PODy 位不产生影响 1: 清除对应的 PODy 位为 0 注: x = A 时, y = 0...15; x = B 时, y = 0...8; x = F 时, y = 0...3; 其余位为保留, 保留位为只读。

### 6.3.12 GPIOB 驱动能力配置寄存器 (GPIOB\_DS)

偏移地址: 0x12C

复位值: 0x0000 0738

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				SR5	SR4	SR3	Reserved			DS5	DS4	DS3	Reserved		
				rw	rw	rw				rw	rw	rw			

位域	名称	描述
31:11	Reserved	保留，必须保持复位值。
10:8	SRy	端口 GPIOB 引脚 PINy (y = 3...5) 的翻转率配置位 0: 快速翻转 1: 慢速翻转 注: y = 3,4,5;
7:6	保留	必须保持复位值。
5:3	DSy	端口 GPIOB 引脚 PINy (y = 3...5) 的驱动能力配置位 0: 高驱动能力 (16mA(5V)/8mA(3.3V)/4mA(2.0V)) 1: 低驱动能力 (8mA(5V)/4mA(3.3V)/2mA(2.0V)) 注: y = 3,4,5; 注: 其他 GPIO, VDD=5/3.3/2V 对应的驱动能力 12/6/3mA
2:0	Reserved	保留，必须保持复位值。

## 6.4 AFIO 寄存器

### 6.4.1 AFIO 寄存器总览

AFIO 基地址: 0x40023000

**表 6-25 AFIO 寄存器总览**

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0300	AFIO_CFG	Reserved								SPI2_NSS	SPI1_NSS	Reserved								EXTI_ETRR	Reserved								EXTIFLITE	IOFLITCFG			
0x0308	AFIO_EXTI_CFG1	Reserved										EXTI3_CFG	Reserved	EXTI2_CFG	Reserved	EXTI1_CFG	Reserved	EXTI0_CFG															
0x030c	AFIO_EXTI_CFG2	Reserved										EXTI6_CFG	Reserved	EXTI5_CFG	Reserved	EXTI4_CFG																	
0x0310	AFIO_DIGEFT_CFG1	Reserved										P <sub>Ay</sub> DIGEFTEN																					
0x0314	AFIO_DIGEFT_CFG2	Reserved										P <sub>By</sub> DIGEFTEN																					
0x0318	AFIO_DIGEFT_CFG3	Reserved										P <sub>Fy</sub> DIGEFTEN																					

## 6.4.2 AFIO 配置寄存器 (AFIO\_CFG)

偏移地址: 0x0300

复位值: 0x0000 3800

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
Reserved								SPI2_NSS	SPI1_NSS	Reserved							
								rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reserved		EXTI_ETRR[2:0]				Reserved						EXTIFLITEN	IOFLITCFG[2:0]				
		rw										rw	rw				

位域	名称	描述
31:24	Reserved	保留, 必须保持复位值。
23	SPI2_NSS	SPI2 的 NSS 模式选择位 (NSS 配置为 AFIO 推挽模式)。 0: NSS 空闲时为高阻态 1: NSS 空闲时为高电平 <i>注: SPI2 作为从机时, 此位必须配置为 0。</i>
22	SPI1_NSS	SPI1 的 NSS 模式选择位 (NSS 配置为 AFIO 推挽模式)。 0: NSS 空闲时为高阻态 1: N3SS 空闲时为高电平 <i>注: SPI1 作为从机时, 此位必须配置为 0。</i>
21:14	Reserved	保留, 必须保持复位值。
13:11	EXTI_ETRR	选择中断线规则转换外部触发重映射 000b: 选择 EXTI0 规则转换外部触发 001b: 选择 EXTI1 规则转换外部触发

位域	名称	描述
		..... 110b: 选择 EXTI6 规则转换外部触发
10:4	Reserved	保留, 必须保持复位值。
3	EXTIFLITEN	EXTI 中断线过滤使能 0: 不使能 1: 使能 <i>注: 滤波使能打开之后, 固定滤波 2 个 LSI CLK</i>
2:0	IOFILTCFG	IO 滤波控制 000b: 旁路滤波器 001b: IO 滤波时间为 1 个 HCLK 时钟周期 010b: IO 滤波时间为 2 个 HCLK 时钟周期 ..... 111b: IO 滤波时间为 7 个 HCLK 时钟周期

### 6.4.3 AFIO 外部中断配置寄存器 1 (AFIO\_EXTI\_CFG1)

偏移地址: 0x0308

复位值: 0x0000 7777

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI3_CFG[3:0]				EXTI2_CFG[3:0]				EXTI1_CFG[3:0]				EXTI0_CFG[3:0]			
rw				rw				rw				rw			

位域	名称	描述
31:16	Reserved	保留, 必须保持复位值。
15:12	EXTI3_CFG	EXTI3 配置 这些位可由软件读写, 用于选择 EXTI3 外部中断的输入源。 0000b: PA12 引脚 0001b: PA13 引脚 0010b: PA14 引脚 0011b: PA15 引脚 其他: 保留
11:8	EXTI2_CFG	EXTI2 配置 这些位可由软件读写, 用于选择 EXTI2 外部中断的输入源。 0000b: PA8 引脚 0001b: PA9 引脚 0010b: PA10 引脚 0011b: PA11 引脚

位域	名称	描述
		其他: 保留
7:4	EXTI1_CFG	EXTI1 配置 这些位可由软件读写, 用于选择 EXTI1 外部中断的输入源。 0000b: PA4 引脚 0001b: PA5 引脚 0010b: PA6 引脚 0011b: PA7 引脚 其他: 保留
3:0	EXTI0_CFG	EXTI0 配置 这些位可由软件读写, 用于选择 EXTI0 外部中断的输入源。 0000b: PA0 引脚 0001b: PA1 引脚 0010b: PA2 引脚 0011b: PA3 引脚 其他: 保留

#### 6.4.4 AFIO 外部中断配置寄存器 2 (AFIO\_EXTI\_CFG2)

偏移地址: 0x030C

复位值: 0x0000 0777

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				EXTI6_CFG[3:0]				EXTI5_CFG[3:0]				EXTI4_CFG[3:0]			
				rw				rw				rw			

位域	名称	描述
31:12	Reserved	保留, 必须保持复位值。
11:8	EXTI6	EXTI6 配置 这些位可由软件读写, 用于选择 EXTI6 外部中断的输入源。 0000b: PF0 引脚 0001b: PF1 引脚 0010b: PF2 引脚 0011b: PF3 引脚 其他: 保留
7:4	EXTI5	EXTI5 配置 这些位可由软件读写, 用于选择 EXTI5 外部中断的输入源。 0000b: PB4 引脚 0001b: PB5 引脚

位域	名称	描述
		0010b: PB6 引脚 0011b: PB7 引脚 0100b: PB8 引脚 其他: 保留
3:0	EXTI4	EXTI4 配置 这些位可由软件读写, 用于选择 EXTI4 外部中断的输入源。 0000b: PB0 引脚 0001b: PB1 引脚 0010b: PB2 引脚 0011b: PB3 引脚 其他: 保留

### 6.4.5 数字毛刺滤波器配置寄存器 1 (AFIO\_DIGEFT\_CFG1)

偏移地址: 0x0310

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PA15DIGEFTEN	PA14DIGEFTEN	PA13DIGEFTEN	PA12DIGEFTEN	PA11DIGEFTEN	PA10DIGEFTEN	PA9DIGEFTEN	PA8DIGEFTEN	PA7DIGEFTEN	PA6DIGEFTEN	PA5DIGEFTEN	PA4DIGEFTEN	PA3DIGEFTEN	PA2DIGEFTEN	PA1DIGEFTEN	PA0DIGEFTEN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:16	Reserved	保留, 必须保持复位值。
15:0	PAyDIGEFTEN	引脚 PAy 数字滤波使能位(y=0...15) 0: 失能 EFT 1: 使能 EFT

### 6.4.6 数字毛刺滤波器配置寄存器 2 (AFIO\_DIGEFT\_CFG2)

偏移地址: 0x0314

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							PB8DIGEFTEN	PB7DIGEFTEN	PB6DIGEFTEN	PB5DIGEFTEN	PB4DIGEFTEN	PB3DIGEFTEN	PB2DIGEFTEN	PB1DIGEFTEN	PB0DIGEFTEN

rw    rw    rw    rw    rw    rw    rw    rw    rw

位域	名称	描述
31:9	Reserved	保留，必须保持复位值。
8:0	PByDIGEFTEN	引脚 PBy 数字滤波使能位(y=0...8) 0: 失能 EFT 1: 使能 EFT

### 6.4.7 数字毛刺滤波器配置寄存器 3 (AFIO\_DIGEFT\_CFG3)

偏移地址: 0x0318

复位值: 0x0000 0000



位域	名称	描述
31:4	Reserved	保留，必须保持复位值。
3:0	PFyDIGEFTEN	引脚 PFy 数字滤波使能位(y=0...3) 0: 失能 EFT 1: 使能 EFT

## 7 DMA 控制器

### 7.1 简介

DMA 控制器总共可以访问 3 个 AHB 从机: Flash、SRAM 和 APB。DMA 控制器由 CPU 控制以执行从源到目的的快速数据移动。配置完成后，无需 CPU 干预即可传输数据。因此，可以释放 CPU 用于其他计算/控制任务或节省整体系统功耗。

MCU 的主要架构是具有循环仲裁方案的多层 AHB-Lite 总线结构。DMA 和 CPU 内核可以并行访问不同的从机，也可以顺序访问相同的从机。

DMA 控制器有 3 个逻辑通道。每个逻辑通道用于服务来自单个或多个外设的内存访问请求。内部仲裁器控制不同 DMA 通道的优先级。

支持外设请求通道用户可配置。

## 7.2 主要特性

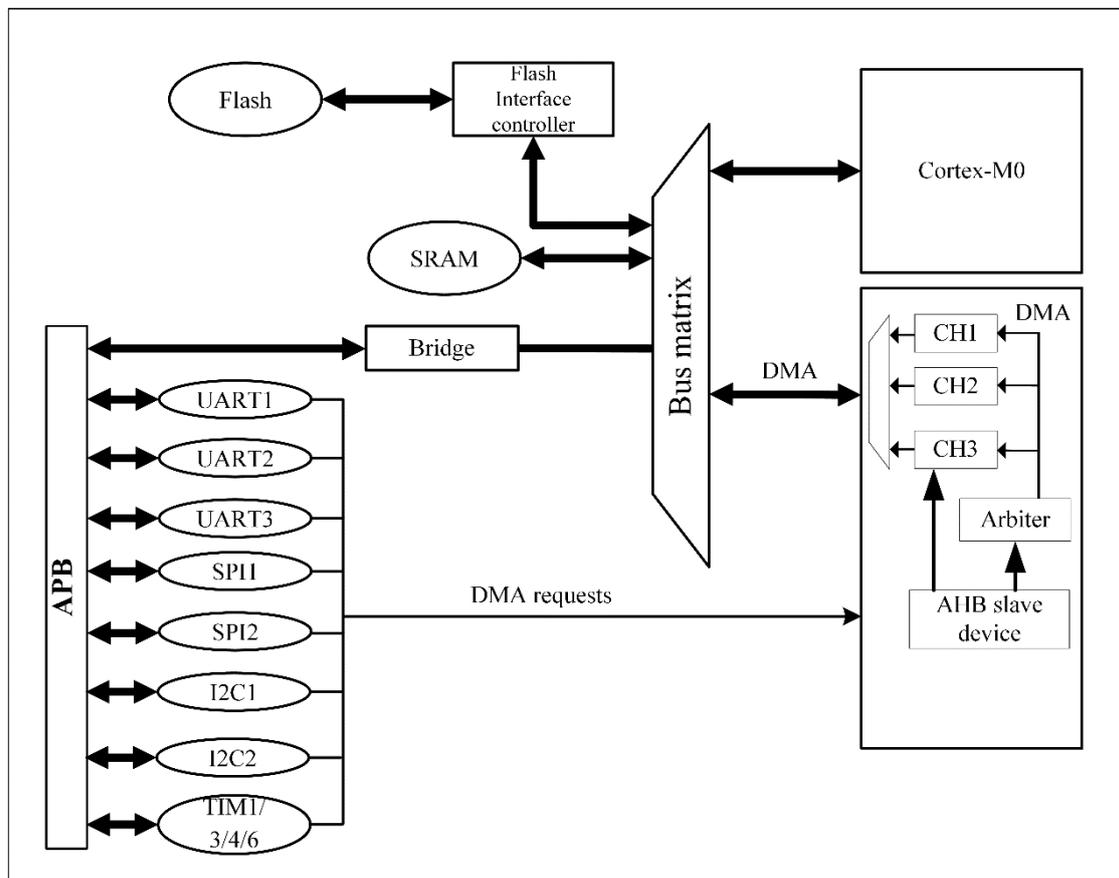
DMA 主要特性：

- 3 个可独立配置的 DMA 通道。
- 每个 DMA 通道支持硬件请求和软件触发来启动传输，并由软件配置。
- 每个 DMA 通道都有专用的软件优先级（DMA\_CHCFGx.PRIOLVL[1:0]位，对应 4 个优先级），可以单独配置。具有相同软件优先级的通道将进一步比较硬件索引（通道号）以确定最终优先级（索引号越低的通道优先级越高）。
- 可配置的源和目标大小。地址设置应与数据大小相对应。
- 每个通道可配置循环传输模式。
- 每个通道有 3 个独立的事件标志和中断（传输完成、半传输、传输错误）和 1 个全局中断标志（由 3 个事件的逻辑或设置）。
- 支持内存到内存、内存到外设和外设到内存三种传输类型。  
*注：也可以支持外设 SPI 到外设 DAC 的搬运（模式配置为外设到内存模式）。*
- 共访问 3 个 AHB 从机：Flash、SRAM 和 APB。
- 可配置数据传输数（0~65535）。

*注：不建议使用 DMA 对 FLASH 进行编程，以避免时序错误。*

## 7.3 功能框图

图 7-1 DMA 框图



## 7.4 功能描述

DMA 控制器和 Cortex™-M0 内核共享相同的系统数据总线。当 CPU 和 DMA 同时访问同一个目标（RAM 或外设）时，DMA 请求会暂停 CPU 访问系统总线几个周期，由总线仲裁器进行循环调度。这允许 CPU 获得至少一半的系统总线（内存或外围设备）带宽。

### 7.4.1 DMA 操作

DMA 请求可以由硬件外设或软件触发，DMA 控制器根据通道的优先级处理请求。根据配置的传输地址和位宽从源地址读取数据，然后将读取的数据存储在目的地址空间中。一次操作后，控制器计算剩余传输次数，并更新下一次传输的源地址和目的地址。

每个 DMA 数据传输包括三个操作：

- 数据访问：根据传输方向确定源地址（DMA\_PADDR<sub>x</sub> 或 DMA\_MADDR<sub>x</sub>），从源地址读取数据。
- 数据存储：根据传输方向确定目的地址（DMA\_PADDR<sub>x</sub> 或 DMA\_MADDR<sub>x</sub>），将读取的数据存储到目的地址空间。
- 计算未完成操作的数量，对 DMA\_TXNUM<sub>x</sub> 寄存器进行减量操作，更新下一个操作的源地址和目的地

址。

## 7.4.2 通道优先级和仲裁器

DMA 使用仲裁策略来处理来自不同通道的多个请求。每个通道的优先级可在通道控制寄存器 (DMA\_CHCFGx) 中进行编程。

4 个优先级：

- ◆ 非常高优先级
- ◆ 高优先级
- ◆ 中优先级
- ◆ 低优先级

默认情况下，如果编程的优先级相同，则索引较低的通道具有较高的优先级。

对于内存到内存的传输，在 4 次传输操作后进行重新仲裁。

对于与外设相关的传输，每次传输操作后都会进行重新仲裁。

## 7.4.3 DMA 通道和传输数量

每个通道都可以在指定地址的外设寄存器和内存地址之间进行 DMA 传输。DMA 传输的数据数量是可编程的，最大支持值为 65535。DMA\_TXNUM 寄存器在每次传输后递减。

## 7.4.4 可编程的数据位宽

外设和内存传输数据位宽支持字节、半字和字，可以通过 DMA\_CHCFGx.PSIZE 和 DMA\_CHCFGx.MSIZE 进行编程。

当 DMA\_CHCFGx.PSIZE 和 DMA\_CHCFGx.MSIZE 不同时，DMA 模块根据表 7-1 对齐数据。

表 7-1 可编程的数据宽度和大小端操作(当 PINC = MINC = 1)

Source width (bit)	Destination width (bit)	Number of transfer (Byte)	Source: Address / data	Transfer operations (R: Read, W: Write)	Destination: Address / data
8	8	4	0x0 / B0 0x1 / B1 0x2 / B2 0x3 / B3	1: R B0 [7:0] @0x0, W B0 [7:0] @0x0 2: R B1 [7:0] @0x1, W B1 [7:0] @0x1 3: R B2 [7:0] @0x2, W B2 [7:0] @0x2 4: R B3 [7:0] @0x3, W B3 [7:0] @0x3	0x0 / B0 0x1 / B1 0x2 / B2 0x3 / B3
8	16	4	0x0 / B0 0x1 / B1 0x2 / B2 0x3 / B3	1: R B0 [7:0] @0x0, W 00B0 [15:0] @0x0 2: R B1 [7:0] @0x1, W 00B1 [15:0] @0x2 3: R B2 [7:0] @0x2, W 00B2 [15:0] @0x4 4: R B3 [7:0] @0x3, W 00B3 [15:0] @0x6	0x0 / 00B0 0x2 / 00B1 0x4 / 00B2 0x6 / 00B3

Source width (bit)	Destination width (bit)	Number of transfer (Byte)	Source: Address / data	Transfer operations (R: Read, W: Write)	Destination: Address / data
8	32	4	0x0 / B0 0x1 / B1 0x2 / B2 0x3 / B3	1: R B0 [7:0] @0x0, W 000000B0 [31:0] @0x0 2: R B1 [7:0] @0x1, W 000000B1 [31:0] @0x4 3: R B2 [7:0] @0x2, W 000000B2 [31:0] @0x8 4: R B3 [7:0] @0x3, W 000000B3 [31:0] @0xC	0x0 / 000000B0 0x4 / 000000B1 0x8 / 000000B2 0xC / 000000B3
16	8	4	0x0 / B1B0 0x2 / B3B2 0x4 / B5B4 0x6 / B7B6	1: R B1B0 [15:0] @0x0, W B0 [7:0] @0x0 2: R B3B2 [15:0] @0x2, W B2 [7:0] @0x1 3: R B5B4 [15:0] @0x4, W B4 [7:0] @0x2 4: R B7B6 [15:0] @0x6, W B6 [7:0] @0x3	0x0 / B0 0x1 / B2 0x2 / B4 0x3 / B6
16	16	4	0x0 / B1B0 0x2 / B3B2 0x4 / B5B4 0x6 / B7B6	1: R B1B0 [15:0] @0x0, W B1B0 [15:0] @0x0 2: R B3B2 [15:0] @0x2, W B3B2 [15:0] @0x2 3: R B5B4 [15:0] @0x4, W B5B4 [15:0] @0x4 4: R B7B6 [15:0] @0x6, W B7B6 [15:0] @0x6	0x0 / B1B0 0x2 / B3B2 0x4 / B5B4 0x6 / B7B6
16	32	4	0x0 / B1B0 0x2 / B3B2 0x4 / B5B4 0x6 / B7B6	1: R B1B0 [15:0] @0x0, W 0000B1B0 [31:0] @0x0 2: R B3B2 [15:0] @0x2, W 0000B3B2 [31:0] @0x4 3: R B5B4 [15:0] @0x4, W 0000B5B4 [31:0] @0x8 4: R B7B6 [15:0] @0x6, W 0000B7B6 [31:0] @0xC	0x0 / 0000B1B0 0x4 / 0000B3B2 0x8 / 0000B5B4 0xC / 0000B7B6
32	8	4	0x0 / B3B2B1B0 0x4 / B7B6B5B4 0x8 / BBBAB9B8 0xC / BFBEBDBC	1: R B3B2B1B0 [31:0] @0x0, W B0 [7:0] @0x0 2: R B7B6B5B4 [31:0] @0x4, W B4 [7:0] @0x1 3: R BBBAB9B8 [31:0] @0x8, W B8 [7:0] @0x2 4: R BFBEBDBC [31:0] @0xC, W BC [7:0] @0x3	0x0 / B0 0x1 / B4 0x2 / B8 0x3 / BC
32	16	4	0x0 / B3B2B1B0 0x4 / B7B6B5B4 0x8 / BBBAB9B8 0xC / BFBEBDBC	1: R B3B2B1B0 [31:0] @0x0, W B1B0 [15:0] @0x0 2: R B7B6B5B4 [31:0] @0x4, W B5B4 [15:0] @0x2 3: R BBBAB9B8 [31:0] @0x8, W B9B8 [15:0] @0x4 4: R BFBEBDBC [31:0] @0xC, W BDBC [15:0] @0x6	0x0 / B1B0 0x2 / B5B4 0x4 / B9B8 0x6 / BDBC
32	32	4	0x0 / B3B2B1B0 0x4 / B7B6B5B4 0x8 / BBBAB9B8 0xC / BFBEBDBC	1: R B3B2B1B0 [31:0] @0x0, W B3B2B1B0 [31:0] @0x0 2: R B7B6B5B4 [31:0] @0x4, W B7B6B5B4 [31:0] @0x4 3: R BBBAB9B8 [31:0] @0x8, W BBBAB9B8 [31:0] @0x8 4: R BFBEBDBC [31:0] @0xC, W BFBEBDBC [31:0] @0xC	0x0 / B3B2B1B0 0x4 / B7B6B5B4 0x8 / BBBAB9B8 0xC / BFBEBDBC

注意:

DMA 总是向 HWDATA[31:0]提供完整的 32 位数据, 无论它是什么目标大小 (HSIZE 仍然遵循设备支持字节/半字操作的目标大小设置)。它提供的 HWDATA[31:0]遵循以下规则:

- 当源大小小于目标大小时, DMA 用 0 填充 MSB, 直到它们的大小匹配并将其复制为 32 位。例如, 源是 8 位数据 0x55, 目标大小是 16 位。DMA 用 0 填充源数据使其成为 16 位 0x0055, 然后将其复制为 32 位数据 0x0055\_0055 并提供给 HWDATA[31:0]; (如果目标大小为 32 位, 则 DMA 只会用 0 填充源数据)。
- 当源大小大于或等于目标大小且小于 32 位时, DMA 将源数据复制到 32 位数据。例如, 源数据为 8 位

数据 0x1F, HWDATA[31:0] = 0x1F1F\_1F1F。如果源数据是 16 位数据 0x2345, 则 HWDATA[31:0] = 0x2345\_2345。

这保证了仅支持字操作的外设不会产生总线错误, 并且所需的数据仍然可以通过额外的位 (即 0 填充) 移动到我们想要的位置。如果用户想要配置一个 8 位寄存器但与 32 位地址边界对齐, 则源大小应设置为 8 位, 目标大小应设置为 32 位, 因此额外的位将用 0 填充。

## 7.4.5 外设/内存地址递增

DMA\_CHCFGx.PINC 和 DMA\_CHCFGx.MINC 分别控制外设地址和内存地址是否使能自动递增模式。软件在传输过程中不能写 (可以读) 地址寄存器。

- 在自动递增模式下, 下一个要传输的地址在每次传输后根据数据位宽 (1、2 或 4) 自动增加。第一次传输的地址存储在 DMA\_PADDRx 或 DMA\_MADDRx 寄存器中。
- 在固定模式下, 地址始终固定为初始地址。

在传输结束时 (即传输计数变为 0), 将根据当前是否工作于循环模式进行不同的处理。

- 在非循环模式下, DMA 在传输完成后停止。要开始新的 DMA 传输, 需要在禁用 DMA 通道的情况下重写 DMA\_TXNUMx 寄存器中的传输数量。
- 在循环模式下, 在传输结束时, DMA\_TXNUMx 寄存器的内容会自动重新加载其初始值, 并且当前内部外设或内存地址也会重新加载 DMA\_PADDRx 或 DMA\_MADDRx 寄存器设置的初始基地址。

## 7.4.6 通道配置流程

详细配置流程如下:

1. 配置中断屏蔽位, 1: 启用中断, 0: 禁用中断。
2. 配置通道外设地址和内存地址以及传输方向。
3. 配置通道优先级, 0: 最低, 3: 最高。
4. 配置外设和内存地址增量。
5. 配置通道传输数据的宽度和数量。
6. 如有必要, 配置循环模式。
7. 如果是存储器到存储器, 配置 MEM2MEM 模式 (注: 要配置 DMA 工作在 M2M 模式, 用户需要将相应的通道选择值设置为保留值, 例如 47)。
8. 在通道 1~3 上重复第 1~8 步。
9. 最后使能相应通道。

如果使用软件提供中断服务, 则软件必须查询中断状态寄存器以检查发生了哪个中断 (软件需要向中断标志清除位写 1 来清除相应的中断)。在使能通道之前, 应清除该通道对应的所有中断。

如果中断是传输完成中断, 软件可以配置下一次传输, 或者向用户报告该通道传输完成。

## 7.4.7 流量控制

支持三种主要的流量控制：

- 存储器到存储器
- 存储器到外设
- 外设到存储器

流控制由每个 DMA 通道配置寄存器中的两个寄存器位控制。流控制用于控制 DMA 通道的源/目标和方向。

表 7-2 流量控制表

DMA_CHCFGx.MEM2MEM	DMA_CHCFGx.DIR	Source	Destination	Transfer
1	x	Memory	Memory	AHB read to AHB write, can do back2back transfer
0	1	Memory	AHB Peripheral	AHB read to AHB write, single transfer
			APB Peripheral	AHB read to APB write, single transfer
0	0	AHB Peripheral	Memory	AHB read to AHB write, single transfer
		APB Peripheral		APB read to AHB write, single transfer

## 7.4.8 循环模式

循环模式用于处理循环缓冲区和连续数据传输。DMA\_CHCFGx.CIRC 用于启用此功能。激活循环模式时，如果要传输的数据数变为 0，则在配置通道时会自动恢复到初始值，继续进行 DMA 操作。

如果用户想关闭循环模式，用户需要向 DMA\_CHCFGx.CHEN 写入 0 以禁用 DMA 通道，然后向 DMA\_CHCFGx.CIRC 写入 0（当 DMA\_CHCFGx.CHEN 为 1 时，DMA\_CHCFGx 寄存器中的其他位不能被改写）。

## 7.4.9 错误管理

对保留地址区域的 DMA 访问会导致 DMA 传输错误。发生错误时，设置传输错误标志，硬件自动清除当前 DMA 通道使能位（DMA\_CHCFGx.CHEN），通道操作停止。如果在 DMA\_CHCFGx 寄存器中设置了传输错误中断使能位，则会产生中断。

## 7.4.10 中断

- 传输完成中断：

通道数据传输完成时会产生中断。中断是一个电平信号。每个通道都有其专用的中断、中断屏蔽控制和中断状态位。当中断标志清除位被设置时，中断状态位被清除。

- 半传输中断：

当传输了一半的通道数据时会产生中断。中断是一个电平信号。每个通道都有其专用的中断、中断屏蔽控制和中断状态位。当中断标志清除位被设置时，中断状态位被清除。

- 传输错误中断：

总线返回错误时产生中断。中断是一个电平信号。每个通道都有其专用的中断、中断屏蔽控制和中断状态位。当中断标志清除位被设置时，中断状态位被清除。

表 7-3 DMA 中断请求

中断事件	事件标志位	使能控制位
半传输	HTXF	HTXIE
传输完成	TXCF	TXCIE
传输错误	ERRF	ERRIE

### 7.4.11 DMA 请求映射

总共有来自所有外设的 33 个 DMA 请求。为了获得更好的支持和完全的灵活性，可以使用寄存器位来选择将哪个 DMA 请求映射到哪个 DMA 通道。下表显示了外设的 DMA 请求到 DMA 控制器的 DMA 通道的映射方案。

表 7-4 DMA 请求映射

DMA request source select	Peripheral DMA request	DMA request source select	Peripheral DMA request
sel = 0	UART1_TX	sel = 21	TIM3_CH1
sel = 1	UART1_RX	sel = 22	TIM3_CH2
sel = 2	UART2_TX	sel = 23	TIM3_CH3
sel = 3	UART2_RX	sel = 24	TIM3_CH4
sel = 4	UART3_TX	sel = 25	TIM3_UP
sel = 5	UART3_RX	sel = 26	TIM3_TRIG
sel = 6	SPI1_TX	sel = 27	TIM4_CH1
sel = 7	SPI1_RX	sel = 28	TIM4_CH2
sel = 8	SPI2_TX	sel = 29	TIM4_CH3
sel = 9	SPI2_RX	sel = 30	TIM4_UP
sel = 10	I2C1_TX	sel = 31	TIM4_TRIG
sel = 11	I2C1_RX	sel = 32	TIM6_UP
sel = 12	I2C2_TX		
sel = 13	I2C2_RX		
sel = 14	TIM1_CH1		
sel = 15	TIM1_CH2		
sel = 16	TIM1_CH3		
sel = 17	TIM1_CH4		
sel = 18	TIM1_COM		
sel = 19	TIM1_UP		
sel = 20	TIM1_TRIG		

注意：DMA 不同通道不能使用同一请求源，否则在多个通道都使能的情况下，只有优先级较高的通道会被触发。

## 7.5 DMA 寄存器

### 7.5.1 DMA 寄存器总览

表 7-5 DMA 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																	
000h	DMA_INTSTS	Reserved																				ERRF3	HTXF3	TXCF3	GLBF3	ERRF2	HTXF2	TXCF2	GLBF2	ERRF1	HTXF1	TXCF1	GLBF1																	
	Reset Value	0																				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
004h	DMA_INTCLR	Reserved																				CERRF3	CHTXF3	CTXCF3	CGLBF3	CERRF2	CHTXF2	CTXCF2	CGLBF2	CERRF1	CHTXF1	CTXCF1	CGLBF1																	
	Reset Value	0																				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
008h	DMA_CHCFG1	Reserved												MEMMEM	PRIOLVL[1:0]	MSIZE[1:0]	PSIZE[1:0]	MINC	PINC	CIRC	DIR	ERRIE	HTXIE	TXCIE	CHEN																									
	Reset Value	0												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																
00Ch	DMA_TXNUM1	Reserved															NDTX[15:0]																																	
	Reset Value	0															0																																	
010h	DMA_PADDR1	ADDR[31:0]																																																
	Reset Value	0																																																
014h	DMA_MADDR1	ADDR[31:0]																																																
	Reset Value	0																																																

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
018h	DMA_CHSEL1	Reserved																								CH_SEL[5:0]							
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
01Ch	DMA_CHCFG2	Reserved														MEMMEM	PRIOLVL[1:0]	MSIZE[1:0]	PSIZE[1:0]	MINC	PINC	CIRC	DIR	ERRIE	HTXIE	TXCIE	CHEN						
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
020h	DMA_TXNUM2	Reserved														NDTX[15:0]																	
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
024h	DMA_PADDR2	ADDR[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
028h	DMA_MADDR2	ADDR[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
02Ch	DMA_CHSEL2	Reserved																								CH_SEL[5:0]							
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
030h	DMA_CHCFG3	Reserved														MEMMEM	PRIOLVL[1:0]	MSIZE[1:0]	PSIZE[1:0]	MINC	PINC	CIRC	DIR	ERRIE	HTXIE	TXCIE	CHEN						
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
034h	DMA_TXNUM3	Reserved														NDTX[15:0]																	
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
038h	DMA_PADDR3	ADDR[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
03Ch	DMA_MADDR3	ADDR[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
040h	DMA_CHSEL3	Reserved																								CH_SEL[5:0]							
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

## 7.5.2 DMA 中断状态寄存器 (DMA\_INTSTS)

偏移地址: 0x00

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved																
													r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved				ERRF3	HTXF3	TXCF3	GLBF3	ERRF2	HTXF2	TXCF2	GLBF2	ERRF1	HTXF1	TXCF1	GLBF1	
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	

位域	名称	描述
31:12	Reserved	保留, 必须保持复位值。
11/7/3	ERRFx	通道 x(x = 1...3)的传输错误标志。 发生传输错误时, 硬件设置该位。该位由软件通过向 DMA_INTCLR.CERRFx 位写 1 清零。 0: 通道 x 上没有发生传输错误。 1: 通道 x 上发生传输错误。
10/6/2	HTXFx	通道 x(x = 1...3)的半传输标志。 当半传输完成时, 硬件设置该位。该位由软件通过向 DMA_INTCLR.CHTXFx 位写 1 清零。 0: 通道 x 上的半传输尚未完成。 1: 通道 x 上的半传输已完成。

位域	名称	描述
9/5/1	TXCF <sub>x</sub>	通道 $x(x = 1\dots1)$ 的传输完成标志。 当传输完成时，硬件设置该位。该位由软件通过向 DMA_INTCLR.CTXCF <sub>x</sub> 位写 1 清零。 0: 通道 $x$ 上的传输尚未完成。 1: 通道 $x$ 上的传输已完成。
8/4/0	GLBF <sub>x</sub>	通道 $x(x = 1\dots3)$ 的全局标志。 当该通道中发生任何中断事件时，硬件设置该位。该位由软件通过向 DMA_INTCLR.CGLBF <sub>x</sub> 位写 1 清零。 0: 通道 $x$ 上没有发生传输错误、半传输或传输完成事件。 1: 通道 $x$ 上发生传输错误、半传输或传输完成事件之一。

### 7.5.3 DMA 中断标志清除寄存器 (DMA\_INTCLR)

偏移地址: 0x04

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
												w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CERRF3	CHTXF3	CTXCF3	CGLBF3	CERRF2	CHTXF2	CTXCF2	CGLBF2	CERRF1	CHTXF1	CTXCF1	CGLBF1
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位域	名称	描述
31:12	Reserved	保留，必须保持复位值。
11/7/3	CERRF <sub>x</sub>	清除通道 $x(x = 1\dots3)$ 的传输错误标志。 软件可以设置该位来清除相应通道的 ERRF。 0: 无动作。 1: 复位相应通道的 DMA_INTSTS.ERRF 位。
10/6/2	CHTXF <sub>x</sub>	清除通道 $x(x = 1\dots3)$ 的半传输标志。 软件可以设置该位来清除相应通道的 HTXF。 0: 无动作。 1: 复位相应通道的 DMA_INTSTS.HTXF 位。
9/5/1	CTXCF <sub>x</sub>	清除通道 $x(x = 1\dots3)$ 的传输完成标志。 软件可以设置该位来清除相应通道的 TXCF。 0: 无动作。 1: 复位相应通道的 DMA_INTSTS.TXCF 位。
8/4/0	CGLBF <sub>x</sub>	清除通道 $x(x = 1\dots3)$ 的全局事件标志。 软件可以设置该位来清除相应通道的 GLBF。 0: 无动作。

位域	名称	描述
31:12	Reserved	保留，必须保持复位值。
		1: 复位相应通道的 DMA_INTSTS.GLBF 位。

## 7.5.4 DMA 通道 x 配置寄存器 (DMA\_CHCFGx)

注: x 为通道号, x = 1...3

偏移地址:  $0x08+20 * (x-1)$

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	MEM2 MEM	PRIOLVL[1:0]	MSIZE[1:0]	PSIZE[1:0]	MINC	PINC	CIRC	DIR	ERRIE	HTXIE	TXCIE	CHEN			
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:15	Reserved	保留，必须保持复位值。
14	MEM2MEM	存储器到存储器模式。 当通道尚未使能时，软件可以将此通道配置为存储器到存储器传输。 0: 存储器和外设之间的通道传输。 1: 通道设置为存储器到存储器间的传输。
13:12	PRIOLVL[1:0]	通道优先级。 当通道未使能时，软件可以编程通道优先级。 00: 低 01: 中 10: 高 11: 非常高
11:10	MSIZE[1:0]	存储器数据大小。 软件可以配置从/向存储器地址读取/写入的数据大小。 00: 8 位 01: 16 位 10: 32 位 11: 保留
9:8	PSIZE[1:0]	外设数据大小。 软件可以配置从/向外设地址读取/写入的数据大小。 00: 8 位 01: 16 位 10: 32 位 11: 保留
7	MINC	存储器地址递增模式。

位域	名称	描述
		软件可以使能/禁能存储器地址递增模式。 0: 内存地址不会随着每次传输而递增。 1: 内存地址随着每次传输而递增。
6	PINC	外设地址增量模式。 软件可以使能/禁能外设地址递增模式。 0: 外设地址不会随着每次传输而递增。 1: 外设地址随每次传输而递增。
5	CIRC	循环模式。 软件可以设置/清除该位。 0: 经过一轮传输后通道停止。 1: 通道配置为循环模式。
4	DIR	数据传输方向 软件可以设置/清除该位。 0: 从外设到存储器的数据传输 1: 从存储器到外设的数据传输。
3	ERRIE	传输错误中断使能。 软件可以使能/禁能传输错误中断。 0: 禁止通道 x 的传输错误中断。 1: 使能通道 x 的传输错误中断。
2	HTXIE	半传输中断使能。 软件可以使能/禁能半传输中断。 0: 禁止通道 x 的半传输中断。 1: 使能通道 x 的半传输中断。
1	TXCIE	传输完成中断使能。 软件可以使能/禁能传输完成中断。 0: 禁止通道 x 的传输完成中断。 1: 使能通道 x 的传输完成中断。
0	CHEN	通道使能。 软件可以设置/复位该位。 0: 禁用通道。 1: 使能通道。

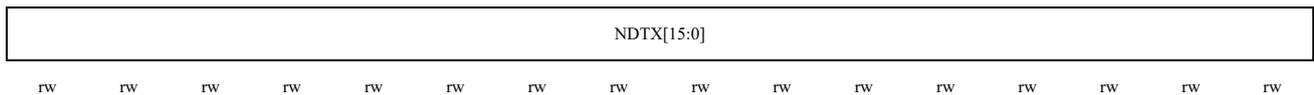
### 7.5.5 DMA 通道 x 传输数量寄存器 (DMA\_TXNUM<sub>x</sub>)

注: x 为通道号, x = 1...3

偏移地址:  $0x0c+20 * (x-1)$

复位值: 0x0000 0000

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



位域	名称	描述
31:16	Reserved	保留，必须保持复位值。
15:0	NDTX	数据传输数量。 要传输的数据数量（0~65535）。软件可以在通道禁用时写入/读出传输数量，并且通道使能后该位为只读。相应的 DMA 通道每次成功传输后，该寄存器就会减 1。如果使能循环模式，它会在达到零时自动重新加载预设值。否则它将保持为零并复位通道使能位。

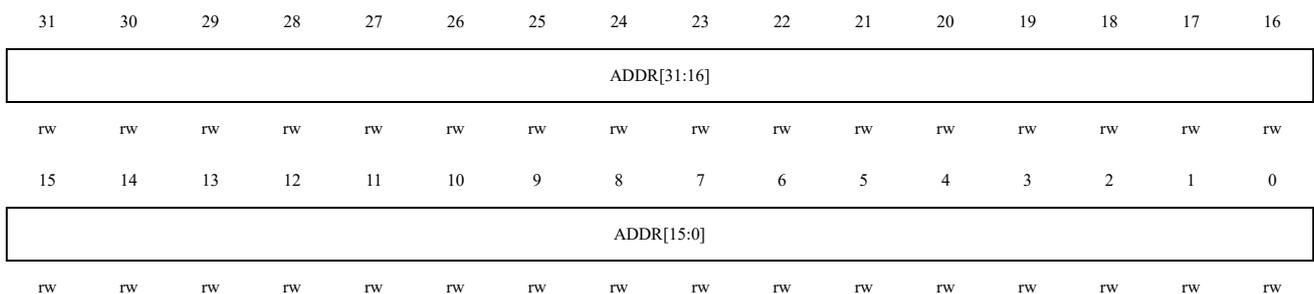
### 7.5.6 DMA 通道 x 外设基地址寄存器 (DMA\_PADDRx)

注：x 为通道号，x = 1...3

偏移地址：0x10+20 \* (x-1)

复位值：0x0000 0000

只有在禁用通道(DMA\_CHCFGx.CHEN = 0)时才能写该寄存器。



位域	名称	描述
31:0	ADDR	外设基地址。 DMA 读取/写入的外设起始地址。 地址的递增由 DMA_CHCFGx.PSIZE 决定。DMA_CHCFGx.PSIZE 等于‘01’，DMA 忽略 PADDR 的第 0 位，如果 DMA_CHCFGx.PSIZE 等于‘10’，DMA 将忽略 PADDR 的第 0 位和第 1 位。

### 7.5.7 DMA 通道 x 存储器基地址寄存器 (DMA\_MADDRx)

注：x 为通道号，x = 1...3

偏移地址：0x14+20 \* (x-1)

复位值：0x0000 0000

只有在禁用通道(DMA\_CHCFGx.CHEN = 0)时才能写该寄存器。

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

位域	名称	描述
31:0	ADDR	存储器基地址。 DMA 读取/写入的存储器起始地址。 地址的递增由 DMA_CHCFGx.MSIZE 决定。DMA_CHCFGx.MSIZE 等于‘01’，DMA 忽略 MADDR 的第 0 位，如果 DMA_CHCFGx.MSIZE 等于‘10’，DMA 将忽略 MADDR 的第 0 位和第 1 位。

## 7.5.8 DMA 通道 x 请求选择寄存器 (DMA\_CHSELx)

注：x 为通道号，x = 1...3

偏移地址：0x18+20 \* (x-1)

复位值：0x0000 0000

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

位域	名称	描述
31:6	Reserved	保留，必须保持复位值。
5:0	CH_SEL[5:0]	DMA 通道请求源选择 0x00: UART1_TX ..... 0x32: TIM6_UP 外设 DMA 请求映射到 DMA 输入请求通道号请参考表 7-4 DMA 请求映射。

## 8 内部集成电路总线(I<sup>2</sup>C)

### 8.1 简介

I<sup>2</sup>C(inter-integrated circuit)总线是一种广泛应用的总线结构，它只有两根双向线，即数据总线 SDA 和时钟总线 SCL，通过这两根线，所有与 I<sup>2</sup>C 总线兼容的设备都可以通过 I<sup>2</sup>C 总线彼此直接通信。

I<sup>2</sup>C 接口连接微控制器和串行 I<sup>2</sup>C 总线，可用于 MCU 和外部 I<sup>2</sup>C 设备的通讯。I<sup>2</sup>C 接口模块实现了 I<sup>2</sup>C 协议的标准模式和快速模式，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)，此外它提供多主机功能，控制所有 I<sup>2</sup>C 总线特定的时序、协议、仲裁。I<sup>2</sup>C 接口模块也支持 DMA 模式，可有效减轻 CPU 的负担。

### 8.2 主要特性

- 同一接口既可实现主机功能又可实现从机功能
- 是并行总线到 I<sup>2</sup>C 总线协议的转换器
- 支持 7 位和 10 位的地址模式和广播寻址
- 作为 I<sup>2</sup>C 主设备可以产生时钟、起始信号和停止信号
- 作为 I<sup>2</sup>C 从设备具有可编程的 I<sup>2</sup>C 地址检测、停止位检测的功能
- 支持标速(最高 100kHz)、快速模式(最高 400kHz)、快速+模式(最高 1MHz)
- 支持中断向量，事件中断和错误中断共用一个中断向量
- 可选的时钟延展功能
- 支持 DMA 模式
- 可选择的 PEC(报文错误检测)生成和校验
- 兼容 SMBus 2.0 和 PMBus
- 可编程的模拟滤波和数字滤波

*注：不是所有产品中都包含上述所有特性，请参考相关的数据手册，确认该产品支持的 I<sup>2</sup>C 功能。*

### 8.3 功能描述

I<sup>2</sup>C 接口通过数据引脚(SDA)和时钟引脚(SCL)连接到 I<sup>2</sup>C 总线与外部设备进行通信，可以连接到标准(高达 100kHz)或快速(400kHz、1MHz)的 I<sup>2</sup>C 总线。I<sup>2</sup>C 模块接收时将数据从串行转换成并行，发送时将数据从并行转换成串行。支持中断模式，用户可以根据需要开启或禁止中断。

#### 8.3.1 SDA/SCL 控制

I<sup>2</sup>C 模块有两条接口线：串行数据线(SDA)和串行时钟线(SCL)。连接到总线上的设备通过这两根线互相传递信息。SDA 和 SCL 都是双向线，通过一个电流源或者上拉电阻接到电源正极。当总线空闲时，两条线都是高电平。连接到总线的设备输出极必须带开漏或者开集，以提供线与功能。I<sup>2</sup>C 总线上的数据在标准模式下可以达到 100 kbit/s，在快速模式下可以达到 1000kbit/s。由于 I<sup>2</sup>C 总线上可能会连接不同工艺的设备，

逻辑‘0’和逻辑‘1’的电平并不是固定的，取决于 VDD 的实际电平。

如果允许时钟延长，即 SCL 线拉低，就可以避免在接收时发生过载错误以及在发送时发生欠载错误。

比如，在发送模式时，如果发送数据寄存器为空且字节发送结束位置起（I2C\_STS1.TXDATE = 1, I2C\_STS1.BSF = 1），I<sup>2</sup>C 接口在传输前保持时钟线为低，以等待软件读取 STS1 后把数据写进数据寄存器（缓冲器和移位寄存器都是空的）；在接收模式时，如果数据寄存器非空且字节发送结束位置起（I2C\_STS1.RXDATNE = 1, I2C\_STS1.BSF = 1），I<sup>2</sup>C 接口在接收到数据字节后保持时钟线为低，以等待软件读 STS1，然后读数据寄存器 DAT（缓冲器和移位寄存器都是满的）。

如果从模式中禁止时钟延长，在接收模式时，如果接收数据寄存器非空（I2C\_STS1.RXDATNE = 1），在接收到下个字节前数据还没有被读出，则发生过载错误，最后一字节也将被丢弃。在发送模式时，如果发送数据寄存器空（I2C\_STS1.TXDATE = 1），在必须发送下个字节之前还没有新数据写进数据寄存器，则发生欠载错误。相同的字节将被重复发出。这种情况下不控制重复写冲突。

### 8.3.2 软件通讯流程

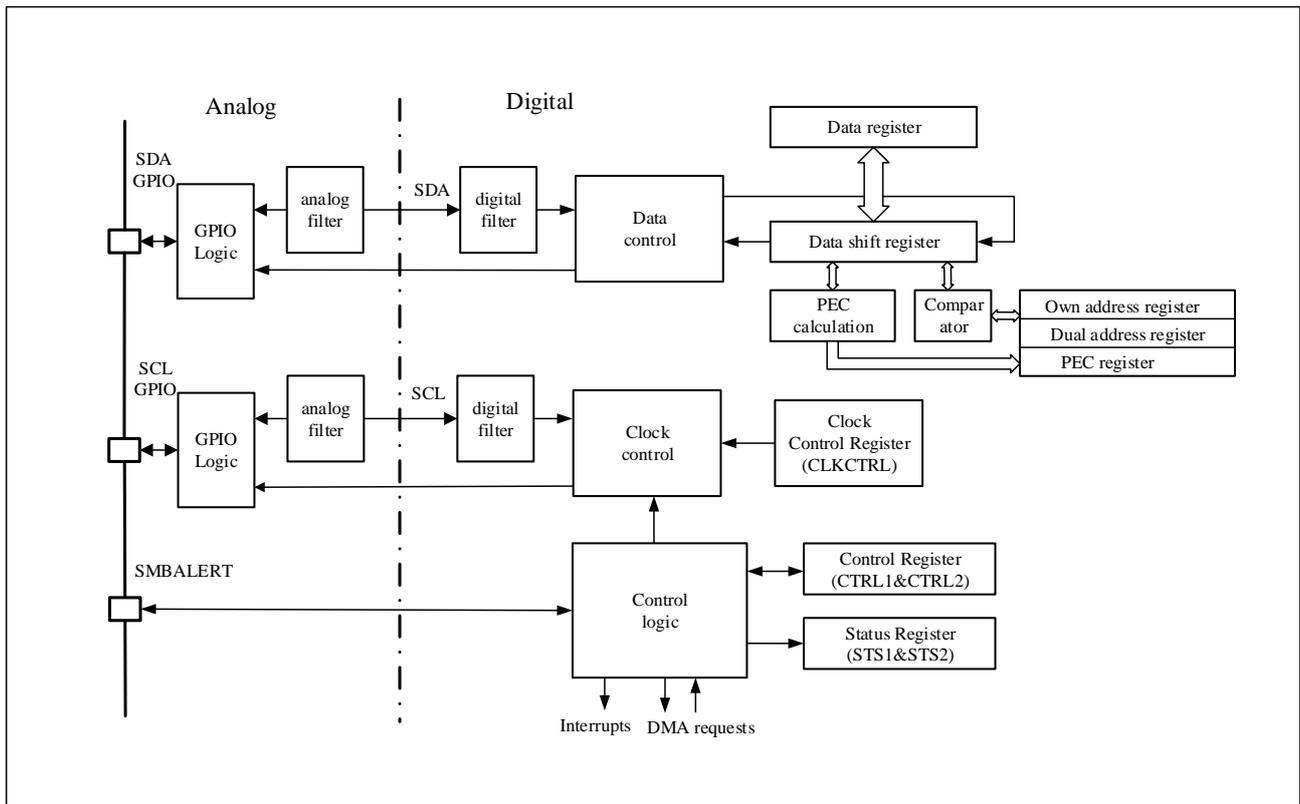
I<sup>2</sup>C 设备的数据传输分为主机和从机。主机是指负责初始化总线上数据的传输并产生时钟信号的设备，此时任何被寻址的设备都是从机。不管 I<sup>2</sup>C 设备是主机还是从机，都可以发送或接收数据。I<sup>2</sup>C 接口支持 4 种运行模式：

- 从机发送器模式
- 从机接收器模式
- 主机发送器模式
- 主机接收器模式

系统复位后，I<sup>2</sup>C 默认工作在从机模式下。通过软件配置 I<sup>2</sup>C 接口在总线上发送一个起始位，随后接口自动地从从模式切换到主模式；当仲裁丢失或产生停止信号时，则从主模式切换到从模式。

I<sup>2</sup>C 接口的功能框图如下：

图 8-1 I<sup>2</sup>C 功能框图

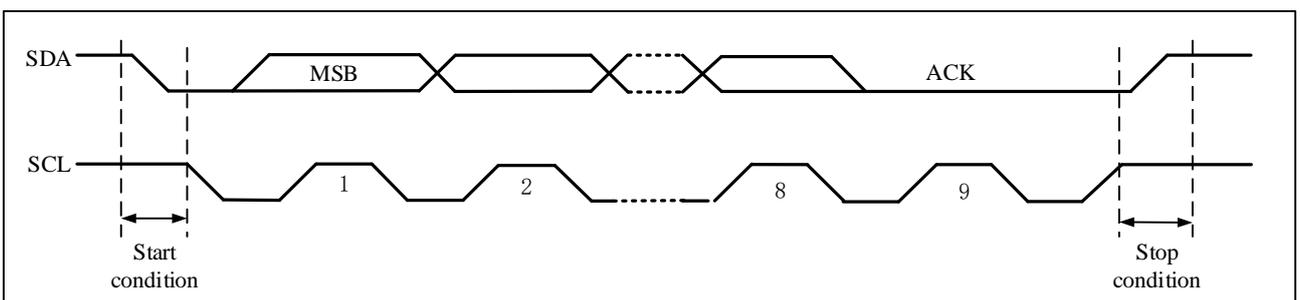


注：在 SMBus 模式下，SMBALERT 是可选信号。如果禁止了 SMBus，则不能使用该信号。

### 8.3.2.1 开始和停止条件

所有的数据传输总是以起始位开始并以停止位结束。起始条件和停止条件都是在主模式下由软件控制产生。起始位 (START) 是指：在 SCL 为高时，SDA 线上出现一个从高到低的电平转换。停止位 (STOP) 是指在 SCL 为高时，SDA 线上出现一个从低到高的电平转换。如下图所示：

图 8-2 I<sup>2</sup>C 总线协议



### 8.3.2.2 时钟同步与仲裁

I<sup>2</sup>C 接口支持多主机仲裁，即两个主机可以同时在空闲总线上开始传送数据，因此就必须通过一些机制来决定哪个主机获取总线的控制权，这一般是通过时钟同步和仲裁来完成的。

I<sup>2</sup>C 电路具有两个关键特点：

- SDA 和 SCL 为漏极开路结构，通过外部的上拉电阻实现了信号的“线与”逻辑；

- SDA 和 SCL 引脚在输出信号的同时还将引脚上的电平进行检测，检测是否与刚才输出一致。这为“时钟同步”和“总线仲裁”提供硬件基础。

I<sup>2</sup>C 设备对总线的操作是通过“把线路接地”来输出逻辑 0。基于 I<sup>2</sup>C 总线的特点，如果一设备发送逻辑 0，其他发送逻辑 1，那么线路看到的只有逻辑 0，所以线路上不可能出现电平冲突的现象。

总线的物理接法允许主设备往总线写数据的同时读取数据。这样两主设备争总线的时候发送逻辑 0 的并不知道竞争的发生，只有发送逻辑 1 的才会发现冲突（当写一个逻辑 1，却读到了 0）从而退出竞争。

### 时钟同步

多个主机可以同时空闲总线上产生时钟。SCL 线的高到低切换会使器件开始数它们的低电平周期，而且一旦器件的时钟变低电平，它会使 SCL 线保持这种状态直到到达时钟的高电平。但是，如果另一个时钟仍处于低电平周期，这个时钟的低到高切换不会改变 SCL 线的状态，因此，SCL 线被有最长低电平周期的器件保持低电平，此时，低电平周期短的器件会进入高电平的等待状态。

当所有有关的器件数完了它们的低电平周期后，时钟线被释放并且变成高电平，之后器件时钟和 SCL 线的状态没有差别，而且所有器件会开始数它们的高电平周期，首先完成高电平周期的器件会再次将 SCL 线拉低。

这样，产生的同步 SCL 时钟的低电平周期由低电平时钟周期最长的器件决定，而高电平周期由高电平时钟周期最短的器件决定。

### 仲裁

仲裁和同步一样，都是为了解决多主机情况下的总线控制冲突。仲裁的过程与从机无关。当两个主机在总线空闲的时候都产生了一个有效的起始位，这种情况就需要决定由哪个主机来完成数据传输，这就是仲裁的过程。

各个主控制器没有对总线实施控制的优先级别，都是由仲裁决定的。总线控制随即而定且逐位进行，他们遵循“低电平优先”的原则，即谁先发送低电平谁就会掌握对总线的控制权。在每一位的仲裁期间，当 SCL 为高时，每个主机都检查自己的 SDA 电平是否和自己发送的相同。理论上讲，如果两个主机所传输的内容完全相同，那么他们能够成功传输而不出现错误。如果一个主机发送高电平但检测到 SDA 电平为低，则认为自己仲裁失败并关闭自己的 SDA 输出驱动，而另一个主机则继续完成自己的传输。

### 8.3.2.3 I<sup>2</sup>C 数据通信流程

每个 I<sup>2</sup>C 设备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。

I<sup>2</sup>C 主机负责产生起始位和结束位来开始和结束一次传输，并且负责产生 SCL 时钟。

I<sup>2</sup>C 模块支持 7 位和 10 位的地址，用户可以通过软件配置 I<sup>2</sup>C 从机的地址。I<sup>2</sup>C 从机检测到 I<sup>2</sup>C 总线上的起始位之后，就开始从总线上接收地址，并把接收到的地址和自身的地址进行比较，一旦两个地址相同，I<sup>2</sup>C 从机将发送一个确认应答(ACK)，并响应总线的后续命令：发送或接受所要求的数据。此外，如果软件开启了广播呼叫，则 I<sup>2</sup>C 从机始终对一个广播地址(0x00)发送确认应答。

数据和地址按 8 位字节进行传输，高位在前。跟在起始条件后的 1 或 2 个字节是地址（7 位模式为 1 个字节，10 位模式为 2 个字节）。地址只在主模式发送。在一个字节传输的 8 个时钟后的第 9 个时钟期间，接收器必须回送一个应答位（ACK）给发送器。如图 8-2 I<sup>2</sup>C 总线协议所示。

软件可以开启或禁能应答（ACK），并可以设置 I<sup>2</sup>C 接口的地址（7 位、10 位地址或广播呼叫地址）。

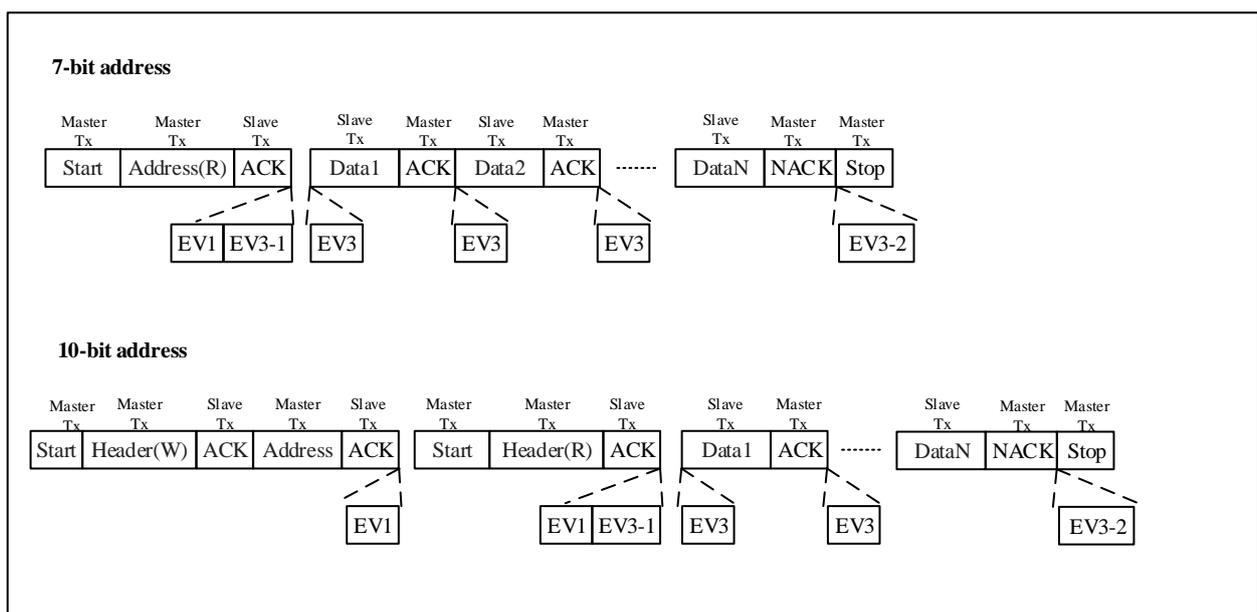
### 8.3.2.4 I<sup>2</sup>C 从机发送模式

在从机模式下发送接收标记位（I2C\_STS2.TRF）指示当前是处于接收器模式还是发送器模式。当在发送器

模式下要发送数据到 I<sup>2</sup>C 总线，软件应该按照下面的步骤操作：

1. 首先，使能 I<sup>2</sup>C 外设时钟，配置 I2C\_CTRL1 中相关寄存器，确保输出正确的 I<sup>2</sup>C 时序。当这两步都完成以后，I<sup>2</sup>C 运行在从机模式下，等待接收起始位和地址。
2. I<sup>2</sup>C 从机先收到一个起始位，随后收到匹配的 7 位或 10 地址，I<sup>2</sup>C 硬件将 I2C\_STS1.ADDRF 位（接收到了地址并且和自身的地址匹配）置 1，软件应该定期查询此位或者中断监视此位，发现置位后，软件读 I2C\_STS1 寄存器然后读 I2C\_STS2 寄存器来清除 I2C\_STS1.ADDRF 位。如果地址是 10 位格式，I<sup>2</sup>C 主机应该接着再产生一个 START 并发送一个地址头到 I<sup>2</sup>C 总线。从机在检测到 START 和紧接着的地址头之后会继续将 I2C\_STS1.ADDRF 位置 1。软件继续通过读 I2C\_STS1 寄存器和接着读 I2C\_STS2 寄存器来第二次清除 I2C\_STS1.ADDRF 位。
3. I<sup>2</sup>C 进入数据发送状态，现在移位寄存器和数据寄存器 I2C\_DAT 都是空，所以硬件将 I2C\_STS1.TXDATE（发送数据空）位置 1。此时软件可以写入第一个字节数据到 I2C\_DAT 寄存器，但是，因为写入 I2C\_DAT 寄存器的字节被立即移入内部移位寄存器了，所以 I2C\_STS1.TXDATE 位并没有被清 0。当移位寄存器非空的时候，I<sup>2</sup>C 开始发送数据到 I<sup>2</sup>C 总线。
4. 第一个字节的发送期间，软件写第二个字节到 I2C\_DAT，此时 I2C\_DAT 寄存器和移位寄存器都不是空。I2C\_STS1.TXDATE 位被清 0。
5. 第一个字节发送完成之后，I2C\_STS1.TXDATE 再次被置起，软件写第三个字节到 I2C\_DAT，同时 I2C\_STS1.TXDATE 位被清 0。在此之后，只要依然有数据待等待被发送且 I2C\_STS1.TXDATE 被置 1，软件都可以写入一个字节到 I2C\_DAT 寄存器。
6. 倒数第二个字节发送期间，软件写最后一个数据到 I2C\_DAT 寄存器来清除 I2C\_STS1.TXDATE 标志位，之后就再也不用关心 I2C\_STS1.TXDATE 的状态。I2C\_STS1.TXDATE 位会在倒数第二个字节发送完成后置起，直到检测到 STOP 结束位时被清 0。
7. 根据 I<sup>2</sup>C 协议，I<sup>2</sup>C 主机不会对接收到的最后一个字节发送应答，所以在最后一个字节发送结束后，I<sup>2</sup>C 从机的 ACKFAIL 位（应答出错）会置起以通知软件发送结束。软件写 0 到 I2C\_STS1.ACKFAIL 位可以清除此位。

图 8-3 从发送器传送序列



说明:

1. EV1: I2C\_STS1.ADDRF = 1, 读 STS1 然后读 STS2 将清除该事件。
2. EV3-1: I2C\_STS1.TXDATE=1, 移位寄存器空,数据寄存器空, 写 DAT。
3. EV3: I2C\_STS1.TXDATE=1, 移位寄存器非空, 数据寄存器空, 写 DAT 将清除该事件。
4. EV3-2: I2C\_STS1.ACKFAIL=1, 在 STS1 的 ACKFAIL 位写“0”可清除该事件。

注:

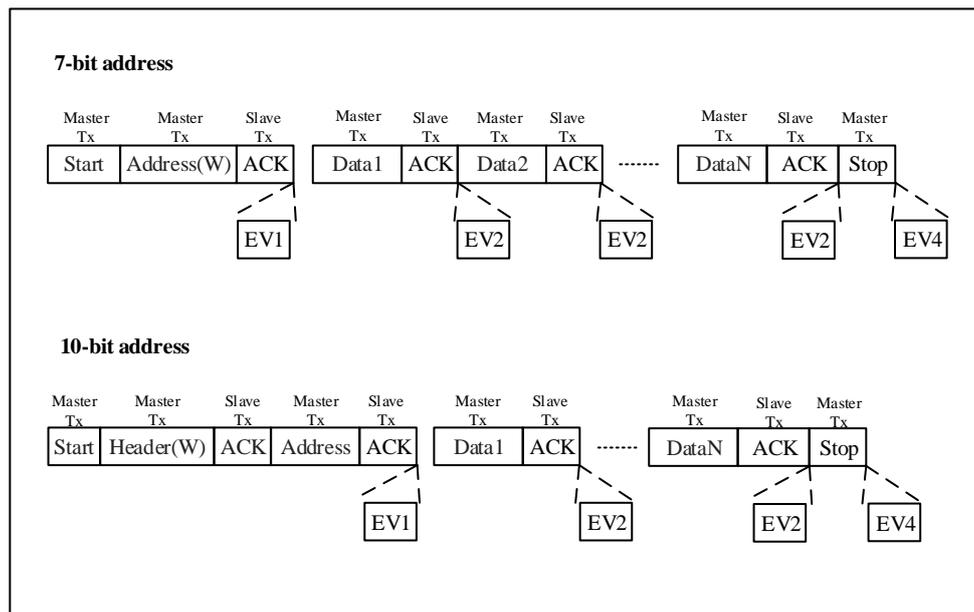
- a) EV1 和 EV3\_1 事件拉长 SCL 低的时间, 直到对应的软件序列结束。
- b) EV3 的软件序列必须在当前字节传输结束之前完成。

### 8.3.2.5 I2C 从机接收模式

在从机模式下接收数据时, 软件应该按如下步骤操作:

1. 首先, 使能 I<sup>2</sup>C 外设时钟, 配置 I2C\_CTRL1 中相关寄存器, 确保输出正确的 I<sup>2</sup>C 时序。当这两步都完成以后, I<sup>2</sup>C 运行在从机模式下, 等待接收起始位和地址。
2. 在接收到 START 起始条件和匹配的 7 位或 10 地址之后, I<sup>2</sup>C 硬件将 I2C\_STS1.ADDRF 位 (接收到了地址并且和自身的地址匹配) 置 1, 此位应该通过软件轮询或者中断来检测, 发现置起后, 软件通过先读 I2C\_STS1 寄存器然后再读 I2C\_STS2 寄存器来清除 I2C\_STS1.ADDRF 位。一旦 I2C\_STS1.ADDRF 位被清 0, I<sup>2</sup>C 从机就开始接收来自 I<sup>2</sup>C 总线的的数据。
3. 当接收到第一个字节后, I2C\_STS1.RXDATNE 位 (接收数据非空) 被硬件置 1, 如果设置了 I2C\_CTRL2.EVTINTEN 和 I2C\_CTRL2.BUFINTEN 位, 则产生一个中断。软件应该通过轮询或者中断来检测该位, 一旦发现置起后, 软件可以读取 I2C\_DAT 寄存器的第一个字节, 此时 I2C\_STS1.RXDATNE 位被清 0。注意, 如果设置了 I2C\_CTRL1.ACKEN 位, 则在接收到一个字节后, 从机应该产生一个应答脉冲。
4. 任何时候, 只要 I2C\_STS1.RXDATNE 位被置 1, 软件均可以从 I2C\_DAT 寄存器读取一个字节。当接收到最后一个字节后, I2C\_STS1.RXDATNE 被置 1, 软件读取最后一个字节
5. 当从机检测到 I<sup>2</sup>C 总线上的停止位 (STOP) 后, 将 I2C\_STS1.STOPF 位置 1, 如果设置了 I2C\_CTRL2.EVTINTEN 位, 则产生一个中断。软件通过先读 I2C\_STS1 寄存器再写 I2C\_CTRL1 寄存器来清除 I2C\_STS1.STOPF 位 ((见下图的 EV4))。

图 8-4 从接收器传送序列



说明:

1. EV1: I2C\_STS1.ADDRF = 1, 读 STS1 然后读 STS2 将清除该事件。
2. EV2: I2C\_STS1.RXDATNE =1, 读 DAT 将清除该事件。
3. EV4: I2C\_STS1.STOPF=1, 读 STS1 然后写 CTRL1 寄存器将清除该事件。

注:

- a) EV1 事件拉长 SCL 低的时间, 直到对应的软件序列结束。
- b) EV2 的软件序列必须在当前字节传输结束之前完成。

### 8.3.2.6 I2C 主机发送模式

在主模式时, I<sup>2</sup>C 接口启动数据传输并产生时钟信号。串行数据传输总是以起始条件开始并以停止条件结束。当通过 START 位在总线上产生了起始条件, 设备就进入了主模式。

在主机模式下发送数据到 I<sup>2</sup>C 总线时, 软件应该按如下步骤操作:

1. 首先, 使能 I<sup>2</sup>C 外设时钟, 配置 I2C\_CTRL1 中相关寄存器, 确保输出正确的 I<sup>2</sup>C 时序。当这两步都完成以后, I<sup>2</sup>C 默认运行在从机模式下, 等待接收起始位和地址。
2. 当 BUSY=0 时, 设置 I2C\_CTRL1.STARTGEN 位为 1, I<sup>2</sup>C 接口将产生一个开始条件并切换至主模式 (I2C\_STS2.MSMODE 位置位)。
3. 一旦发出开始条件, I<sup>2</sup>C 硬件将 I2C\_STS1.STARTBF 位 (起始位标志) 置 1 然后进入主机模式, 如果设置了 I2C\_CTRL2.EVTINTEN 位, 则会产生一个中断。接着软件读 I2C\_STS1 寄存器然后写一个 7 位地址位或带有地址头的 10 位地址位到 I2C\_DAT 寄存器来清除 I2C\_STS1.STARTBF 位。I2C\_STS1.STARTBF 位被清 0 后 I<sup>2</sup>C 就开始发送地址或者地址头到 I<sup>2</sup>C 总线。

在 10 位地址模式时, 发送一个头序列会产生以下事件:

- ◆ I2C\_STS1.ADDR10F 位被硬件置位, 如果设置了 I2C\_CTRL2.EVTINTEN 位, 则产生一个中断。然后主设备读 STS1 寄存器, 再将第二个地址字节写入 DAT 寄存器。

- ◆ I2C\_STS1.ADDRF 位被硬件置位，如果设置了 I2C\_CTRL2.EVTINTEN 位，则产生一个中断。随后主设备读 STS1 寄存器，跟着读 STS2 寄存器

*注：在发送器模式，主设备先发送头字节 (11110xx0)，然后发送从地址的低 8 位。(这里 xx 代表 10 位地址中的最高 2 位)。*

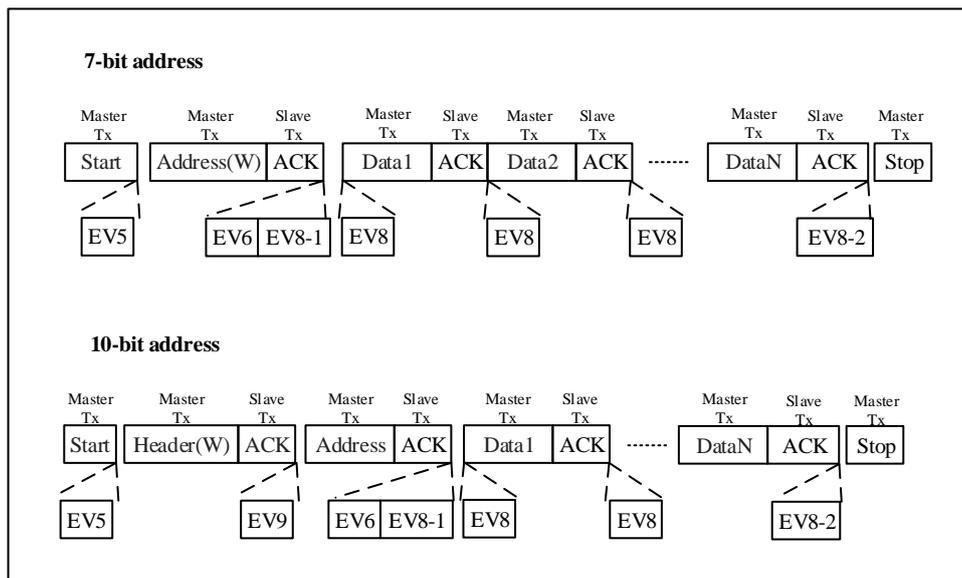
在 7 位地址模式时，只需送出一个地址字节，一旦该地址字节被送出：

- ◆ ADDRf 位被硬件置位，如果设置了 I2C\_CTRL2.EVTINTEN 位，则产生一个中断。随后主设备等待一次读 STS1 寄存器，跟着读 STS2 寄存器。

*注：在发送器模式，主设备发送从地址时置最低位为‘0’。*

4. 7 位或 10 位的地址位发送完成后，I<sup>2</sup>C 硬件将 I2C\_STS1.ADDRF 位（地址已被发送）置 1，如果设置了 I2C\_CTRL2.EVTINTEN 位，则产生一个中断，软件通过读 I2C\_STS1 寄存器然后读 I2C\_STS2 寄存器来清除 I2C\_STS1.ADDRF
5. I<sup>2</sup>C 进入数据发送状态，因为移位寄存器和数据寄存器（I2C\_DAT）都是空的，所以硬件将 I2C\_STS1.TXDATE 位（发送数据空）置 1，接着软件写第一个字节数据到 I2C\_DAT 寄存器，但是，因为写入 I2C\_DAT 寄存器的字节被立即移入内部移位寄存器，所以 I2C\_STS1.TXDATE 位此时不会被清零。一旦移位寄存器非空，I<sup>2</sup>C 就开始发送数据到总线。
6. 在第一个字节的发送过程中，软件写第二个字节到 I2C\_DAT，此时 I2C\_STS1.TXDATE 被清零。任何时候，只要还有数据等待被发送，且 I2C\_STS1.TXDATE 位被置 1，软件都可以向 I2C\_DAT 寄存器写入一个字节。
7. 在倒数第二个字节发送过程中，软件写入最后一个字节数据到 I2C\_DAT 来清除 I2C\_STS1.TXDATE 标志位，此后就不用关心 I2C\_STS1.TXDATE 位的状态。I2C\_STS1.TXDATE 位会在倒数第二个字节发送完成后被置起，直到发送停止位（STOP）时被清零。
8. 最后一个字节发送结束后，因为移位寄存器和 I2C\_DAT 寄存器此时都为空，I<sup>2</sup>C 主机将 I2C\_STS1.BSF 位（字节发送结束）置位，在清除 I2C\_STS1.BSF 位之前 I<sup>2</sup>C 接口将保持 SCL 为低电平；读出 I2C\_STS1 之后再写入 I2C\_DAT 寄存器将清除 I2C\_STS1.BSF 位。软件此时设置 I2C\_CTRL1.STOPGEN 位产生一个停止条件，然后 I<sup>2</sup>C 接口将自动回到从模式（I2C\_STS2.MSMODE 位清除）。

图 8-5 主发送器传送序列



说明:

1. EV5: I2C\_STS1.STARTBF = 1, 读 STS1 然后将地址写如 DAT 寄存器将清除该事件。
2. EV6: I2C\_STS1.ADDRF = 1, 读 STS1 然后读 STS2 将清除该事件。
3. EV8\_1: I2C\_STS1.TXDATE = 1, 移位寄存器空, 数据寄存器空, 写 DAT 寄存器。
4. EV8\_2: I2C\_STS1.TXDATE = 1, 移位寄存器非空, 数据寄存器空, 写 DAT 寄存器将清除该事件。
5. EV8\_2: I2C\_STS1.TXDATE = 1, I2C\_STS1.BSF = 1, 请求设置停止位。这两个事件由硬件在产生停止条件时清除。
6. EV9: I2C\_STS1.ADDR10F = 1, 读 STS1 然后写入 DAT 寄存器将清除该事件。

注:

- a) EV5、EV6、EV9、EV8\_1 和 EV8\_2 事件拉长 SCL 低的时间, 直到对应的软件序列结束。
- b) EV8 的软件序列必须在当前字节传输结束之前完成。
- c) 当 TXDATE 或 BSF 位置位时, 停止条件应安排在出现 EV8\_2 事件时。

### 8.3.2.7 I2C 主机接收模式

在主机模式下从 I<sup>2</sup>C 总线接收数据软件应该按如下步骤操作:

1. 首先, 使能 I<sup>2</sup>C 外设时钟, 配置 I2C\_CTRL1 中相关寄存器, 确保输出正确的 I<sup>2</sup>C 时序。使能和配置以后, I<sup>2</sup>C 默认运行在从机模式下, 等待接收起始位和地址。
2. 当 BUSY=0 时, 设置 I2C\_CTRL1.STARTGEN 位为 1, I<sup>2</sup>C 接口将产生一个开始条件并切换至主模式 (I2C\_STS2.MSMODE 位置位)。
3. 一旦发出开始条件, I<sup>2</sup>C 硬件将 I2C\_STS1.STARTBF 位 (起始位标志) 置 1 然后进入主机模式, 如果设置了 I2C\_CTRL2.EVTINTEN 位, 则会产生一个中断。接着软件读 I2C\_STS1 寄存器然后写一个 7 位地址位或带有地址头的 10 位地址位到 I2C\_DAT 寄存器来清除 I2C\_STS1.STARTBF 位。I2C\_STS1.STARTBF 位被清 0 后 I<sup>2</sup>C 就开始发送地址或者地址头到 I<sup>2</sup>C 总线。

在 10 位地址模式时，发送一个头序列会产生以下事件：

- I2C\_STS1.ADDR10F 位被硬件置位，如果设置了 I2C\_CTRL2.EVTINTEN 位，则产生一个中断。然后主设备读 STS1 寄存器，再将第二个地址字节写入 DAT 寄存器。
- I2C\_STS1.ADDRF 位被硬件置位，如果设置了 I2C\_CTRL2.EVTINTEN 位，则产生一个中断。随后主设备读 STS1 寄存器，跟着读 STS2 寄存器。

*注：在接收器模式，主设备先发送头字节 (11110xx0)，然后发送从地址的低 8 位，然后再重新发送一个开始条件，后面跟着头字节 (11110xx1) (这里 xx 代表 10 位地址中的最高 2 位)。*

在 7 位地址模式时，只需送出一个地址字节，一旦该地址字节被送出：

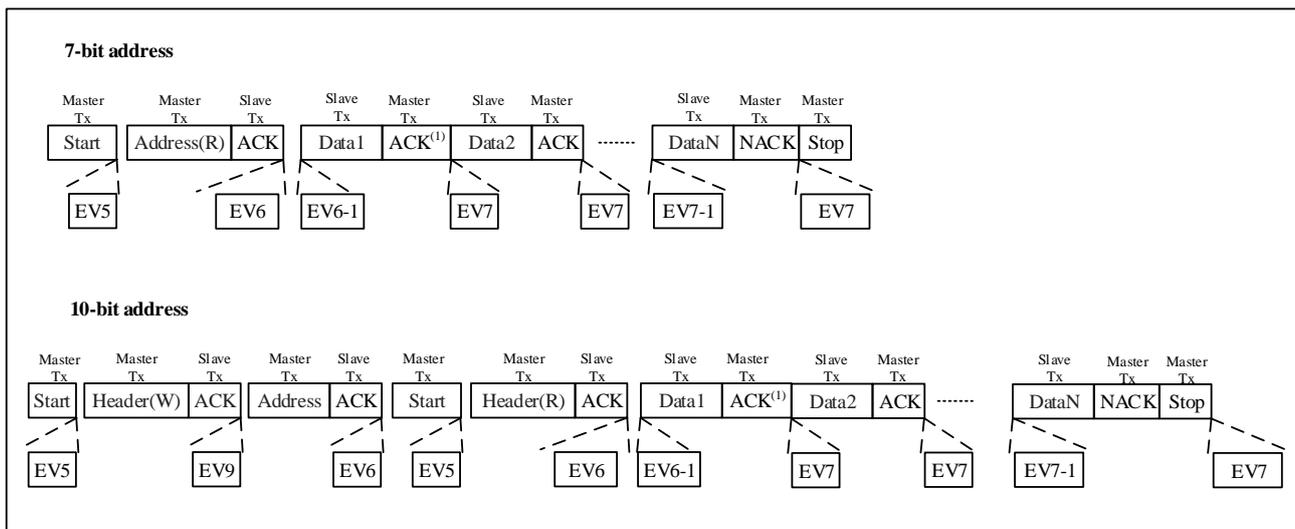
- I2C\_STS1.ADDRF 位被硬件置位，如果设置了 I2C\_CTRL2.EVTINTEN 位，则产生一个中断。随后主设备等待一次读 STS1 寄存器，跟着读 STS2 寄存器。

*注：在接收器模式，主设备发送从地址时置最低位为‘1’。*

4. 7 位或 10 位的地址位发送完成后，I<sup>2</sup>C 硬件将 I2C\_STS1.ADDRF 位（地址已被发送）置 1，如果设置了 I2C\_CTRL2.EVTINTEN 位，则产生一个中断，软件通过读 I2C\_STS1 寄存器然后读 I2C\_STS2 寄存器来清除 I2C\_STS1.ADDRF，在发送地址和清除 I2C\_STS1.ADDRF 之后，如果地址是 10 位格式，软件应该再次将 STARTGEN 位置 1 来重新产生一个 START (S<sub>r</sub>)。在 START 产生后，I2C\_STS1.STARTBF 位会被置 1。软件应该通过先读 I2C\_STS1 然后写地址头到 I2C\_DAT 来清除 I2C\_STS1.STARTBF 位，然后地址头被发到 I<sup>2</sup>C 总线，ADDRF 再次被置 1。软件应该再次通过先读 I2C\_STS1 然后读 I2C\_STS2 来清除 I2C\_STS1.ADDRF 位。
5. 在发送完地址和清除 I2C\_STS1.ADDRF 之后，I<sup>2</sup>C 接口进入主机接收器模式。在此模式下，I<sup>2</sup>C 接口从 SDA 线接收数据字节，并通过内部移位寄存器送至 DAT 寄存器。一旦接收到第一个字节，硬件会将 I2C\_STS1.RXDATNE 位（接收数据非空标志位）置 1，如果 ACKEN 位被置位，发出一个应答脉冲。此时软件可以从 I2C\_DAT 寄存器读取第一个字节，之后 I2C\_STS1.RXDATNE 位被清 0。此后，只要 I2C\_STS1.RXDATNE 被置 1，软件就可以从 I2C\_DAT 寄存器读取一个字节。
6. 主设备在从从设备接收到最后一个字节后发送一个 NACK。接收到 NACK 后，从设备释放对 SCL 和 SDA 线的控制；主设备就可以发送一个停止/重新起始条件。为了在收到最后一个字节后产生一个 NACK 脉冲，接收完倒数第二个字节(N-1)数据之后，软件应该立即清除 ACKEN 位。为了产生一个停止/重新起始条件，软件必须在读倒数第二个数据字节之后将 I2C\_CTRL1.STOPGEN 位或者 I2C\_CTRL1.STARTGEN 置 1，这一过程需要在最后一个字节接收完毕之前完成，以确保 NACK 发送给最后一个字节。
7. 最后一个字节接收完毕后，I2C\_STS1.RXDATNE 位被置 1，软件可以读取最后一个字节。由于 I2C\_CTRL1.ACKEN 已经在前一步骤中被清 0，I<sup>2</sup>C 不再为最后一个字节发送 ACK，并在最后一个字节发送完毕后产生一个 STOP 停止位。

*注意：以上步骤要求字节数目 N>1，如果 N=1，步骤 6 应该在步骤 4 之后就执行，且需要在字节接收完成之前完成。*

图 8-6 主接收器传送序列



说明:

1. EV5: I2C\_STS1.STARTBF=1, 读 STS1 然后将地址写入 DAT 寄存器清除该事件。
2. EV6: I2C\_STS1.ADDRF=1, 读 STS1 然后读 STS2 将清除该事件。在 10 位主接收模式下, 该事件后应设置 CTRL1 的 STARTGEN=1。
3. EV6\_1: 没有对应的事件标志, 至适于接收 1 个字节的情况。恰好在 EV6 之后 (即清除了 ADDRDF 之后), 要清除响应和停止条件的产生位。
4. EV7: I2C\_STS1.RXDATNE=1, 读 DAT 寄存器消除该事件。
5. EV7\_1: I2C\_STS1.RXDATNE =1, 读 DAT 寄存器清除该事件。设置 I2C\_CTRL1.ACKEN=0 和 I2C\_CTRL1.STOPGEN=1。
6. EV9: I2C\_STS1.ADDR10F=1, 读 STS1 然后写入 DAT 寄存器将清除该事件。

注:

- a) 如果收到一个单独的字节, 则是 NA。
- b) EV5、EV6 和 EV9 事件拉长 SCL 低电平, 直到对应的软件序列结束。
- c) EV7 的软件序列必须在当前字节传输结束前完成。
- d) EV6\_1 或 EV7\_1 的软件序列必须在当前传输字节的 ACK 脉冲之前完成。

### 8.3.3 错误条件

PC 的错误主要有总线错误、应答错误、仲裁丢失、过载/欠载错误。这些错误都可能造成通讯的失败。

#### 8.3.3.1 应答错误 (ACKFAIL)

当接口检测到应答位与期望不符时, 将产生应答错误。此时 I2C\_STS1.ACKFAIL 被置位, 如果设置了 I2C\_CTRL2.ERRINTEN 位, 则产生一个中断。当发送器接收到一个 NACK 时, 必须复位通讯; 如果处于从模式, 硬件会释放总线。如果是处于主模式, 软件必须生产一个停止条件。

### 8.3.3.2 总线错误 (BUSERR)

在一个地址或数据字节传输期间,当 I<sup>2</sup>C 接口检测到一个外部的停止或起始条件则产生总线错误,I2C\_STS1.BUSERR 被置位。如果设置了 I2C\_CTRL2.ERRINTEN 位为“1”,则产生一个中断。

在主模式情况下,硬件不释放总线,同时不影响当前的传输状态。此时由软件决定是否要中止当前的传输。

在从模式情况下,数据被丢弃,硬件释放总线。此时有两种情况:如果检测到错误的开始条件,从设备认为是一个重启动,并等待地址或停止条件。如果检测到错误的停止条件,从设备按正常的停止条件操作,同时硬件释放总线。

### 8.3.3.3 仲裁丢失 (ARLOST)

当 I<sup>2</sup>C 接口检测到仲裁丢失时产生仲裁丢失错误,硬件释放总线,I2C\_STS1.ARLOST 位被置位。如果设置了 I2C\_CTRL2.ERRINTEN 位为“1”,则产生一个中断。

I<sup>2</sup>C 接口自动回到从模式(I2C\_STS2.MSMODE 位被清除)。当 I<sup>2</sup>C 接口丢失了仲裁,则它无法在同一次传输中作为从机响应它的从地址,但它可以在赢得了仲裁的主设备重新发送起始条件之后响应。

### 8.3.3.4 过载/欠载错误 (OVERRUN)

在从机接收模式下,如果禁止时钟延长,容易发生过载/欠载错误。

当它已经接收到一个字节(I2C\_STS1.RXDATNE=1),但在 DAT 寄存器中前一个字节数据还没有被读出,则发生过载错误,数据寄存器最后接收的数据被丢弃;同时软件应清除 I2C\_STS1.RXDATNE 位,发送器重新发送最后一次发送的字节。

在从机发送模式下,如果禁止时钟延长,在当前字节已经发送完成,而 DAT 仍然为空(I2C\_STS1.TXDATE=1),则发生欠载错误。此时,在 DAT 寄存器中的前一个字节将被重复发出;用户应该确定在发生欠载错时,接收端应丢弃重复接收到的数据。发送端应按 I<sup>2</sup>C 总线标准在规定的更新时间更新 I2C\_DAT 寄存器。

在发送第一个字节时,必须在清除 I2C\_STS1.ADDRF 之后并且第一个 SCL 上升沿之前写入 I2C\_DAT 寄存器;如果不能做到这点,则接收方应该丢弃第一个数据。

## 8.3.4 DMA 应用

在传输时,当数据寄存器为空或满时,能够生成 DMA 请求。DMA 能够写数据到 I<sup>2</sup>C 数据寄存器,或从 I<sup>2</sup>C 数据寄存器读出数据以减少 CPU 的开销。

DMA 请求必须在当前字节传输结束之前被响应。当相应 DMA 通道设置的数据传输已经完成时,DMA 控制器发送传输结束信号 EOT 到 I<sup>2</sup>C 接口,并且在中断使能时产生一个中断。

在主机发送模式,在 EOT 中断服务程序中,需禁止 DMA 请求,然后在等到 I2C\_STS1.BSF 事件后设置停止条件。

在主机接收模式,当要接收的数据数目大于或等于 2 时,DMA 控制器发送一个硬件信号 EOT\_1,它对应 DMA 传输(字节数-1)。如果设置了 I2C\_CTRL2.DMALAST 位,硬件在发送完 EOT\_1 后的下一个字节,将自动发送 NACK。在中断允许的情况下,用户可以在 DMA 传输完成的中断服务程序中产生一个停止条件。

### 8.3.4.1 发送流程

DMA 模式通过设置 I2C\_CTRL2.DMAEN 位使能。只要 I2C\_STS1.TXDATE 位被置位,数据将由 DMA 从预置的存储区装载进 I2C\_DAT 寄存器。设置 DMA 通道进行 I<sup>2</sup>C 发送,须执行以下步骤(x 是通道号):

1. 在 DMA\_PADDR<sub>x</sub> 寄存器中设置 I2C\_DAT 寄存器地址。数据将在每个 I2C\_STS1.TXDATE 事件后从存

存储器传送至这个地址。

2. 在 DMA\_MADDRx 寄存器中设置存储器地址。数据在每个 I2C\_STS1.TXDATE 事件后从这个存储区传送到 I2C\_DAT。
3. 在 DMA\_TXNUMx 寄存器中设置所需传输的字节数。在每个 I2C\_STS1.TXDATE 事件后，此值递减，直到 0。
4. 利用 DMA\_CHCFGx 寄存器中的 PRIOLVL[1:0]位配置通道优先级。
5. 设置 DMA\_CHCFGx 寄存器中的 DIR 位，并根据应用要求可以配置在整个传输完成一半或全部完成时发出中断请求。
6. 通过设置 DMA\_CHCFGx 寄存器上的 CHEN 位激活通道。
7. 当 DMA 控制器中设置的数据传输数目已经完成时，DMA 控制器给 I2C 接口发送一个传输结束的 EOT/EOT\_1 信号。在中断允许的情况下，将产生一个 DMA 中断。

*注：使用 DMA 进行发送时，不要设置 I2C\_CTRL2.BUFINTEN 位。*

### 8.3.4.2 接收流程

DMA 模式通过设置 I2C\_CTRL2.DMAEN 位使能。每次接收到数据字节时，将由 DMA 把 I2C\_DAT 寄存器的数据传送到设置的存储区。设置 DMA 通道进行 I2C 接收，须执行以下步骤（x 是通道号）

1. 在 DMA\_PADDRx 寄存器中设置 I2C\_DAT 寄存器的地址。数据将在每次 I2C\_STS1.RXDATNE 事件后从此地址传送到存储区。
2. 在 DMA\_MADDRx 寄存器中设置存储区地址。数据将在每次 I2C\_STS1.RXDATNE 事件后从 I2C\_DAT 寄存器传送到此存储区。
3. 在 DMA\_TXNUMx 寄存器中设置所需的传输字节数。在每个 I2C\_STS1.RXDATNE 事件后，此值递减，直到 0。
4. 用 DMA\_CHCFGx 寄存器中的 PRIOLVL[1:0]配置通道优先级。
5. 清除 DMA\_CHCFGx 寄存器中的 DIR 位，根据应用要求可以设置在数据传输完成一半或全部完成时发出中断请求。
6. 设置 DMA\_CHCFGx 寄存器中的 CHEN 位激活该通道。
7. 当 DMA 控制器中设置的数据传输数目已经完成时，DMA 控制器给 I2C 接口发送一个传输结束的 EOT/EOT\_1 信号。在中断允许的情况下，将产生一个 DMA 中断。

*注：使用 DMA 进行接收时，不要设置 I2C\_CTRL2.BUFINTEN 位。*

### 8.3.5 包错误校验 (PEC)

将 I2C\_CTRL1.PECEN 位置 1 就可以使能 PEC 功能，PEC 使用 CRC-8 算法对包括地址和读/写位在内所有信息字节进行计算，从而提高通信的可靠性。包错误校验 (PEC) 计算器使用的 CRC-8 多项式为  $C(x) = x^8 + x^2 + x + 1$ 。

在发送模式时，软件可以在最后一个 I2C\_STS1.TXDATE 事件时设置 I2C\_CTRL1.PEC 传输位，PEC 将在最后一个字节后被发送。在接收模式时，软件在最后一个 I2C\_STS1.RXDATNE 事件之后设置 I2C\_CTRL1.PEC 位，然后接收 PEC 字节，并将接收到的 PEC 字节与内部计算的 PEC 值进行比较。如果不等于内部计算的 PEC，接收器发送一个 NACK。如果是主机接收器模式，不管校对的结果如何，PEC 后都将发送 NACK。需

要注意，I2C\_CTRL1.PEC 位必须在接收当前字节的 ACK 脉冲之前设置。

如果 DMA 和 PEC 计算器都被激活，I<sup>2</sup>C 将自动发送或者检查 PEC 值。

在发送模式时，当 I<sup>2</sup>C 接口从 DMA 控制器处接收到 EOT 信号时，它在最后一个字节后自动发送 PEC。在接收模式时，当 I<sup>2</sup>C 接口从 DMA 处接收到一个 EOT\_1 信号时，它将自动把下一个字节作为 PEC，并且和内部计算的 PEC 进行比较。在接收到 PEC 后产生一个 DMA 请求。

为了允许中间 PEC 传输，I2C\_CTRL2.DMALAST 位用于判别是否真是最后一个 DMA 传输。如果确实是最后一个主接收器的 DMA 请求，在接收到最后一个字节后自动发送 NACK。

当仲裁丢失的时候，PEC 计算失效。

### 8.3.6 超时错误

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。这就是为什么 SMBus 有最小传输速率的要求——为了防止超时后长时间锁死总线。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 会话。

N32G033 系列芯片无论是 I2C 模式还是 SMBUS 模式均支持超时时间配置，包含 SCL 低超时判断、SCL 高超时判断、SDA 低超时判断：

其中 SCL 低电平超时功能通过 I2C\_CTRL1.LTOSEL 配置超时阈值，I2C\_CTRL2.LTOEN 使能超时判断，SCL 为低的时间超过设置阈值时 I2C\_STS1.SCLLTO 置 1，此时若 I2C\_CTRL2.SCLLTOINTEN 为 1 则产生中断；

SCL 高电平超时功能通过 I2C\_CTRL1.HTOSEL 配置超时阈值，I2C\_CTRL2.HTOEN 使能超时判断，SCL 为高的时间超过设置阈值时 I2C\_STS1.SCLHTO 置 1，此时若 I2C\_CTRL2.SCLHTOINTEN 为 1 则产生中断；

SDA 低电平超时功能通过 I2C\_CTRL1.LTOSEL 配置超时阈值，I2C\_CTRL2.LTOEN 使能超时判断，SDA 为低的时间超过设置阈值时 I2C\_STS1.SDALTO 置 1，此时若 I2C\_CTRL2.SDALTOINTEN 为 1 则产生中断；

另外 SMBUS 模式下，会自动开启 SCL 累积低电平超时功能，超时阈值主机固定为 10ms，从机固定为 25ms，SCL 累积为低的时间超过阈值时 I2C\_STS1.SCLHTO 置 1，此时若 I2C\_CTRL2.SCLHTOINTEN 为 1 则产生中断；

从模式超时：从设备复位通讯，硬件释放总线。主模式超时：硬件发出停止条件。

### 8.3.7 SMBus

#### 8.3.7.1 介绍

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。通过它，各设备之间以及设备与系统的其他部分之间可以互相通信。SMBus 是 I2C 的一种衍生总线形式，为系统和电源管理相关的任务提供一条控制总线。SMBus 基于 I2C 通信标准，是一个与系统管理和电源管理相关的控制总线。想要了解更多信息，请参考 SMBus 规范 V2.0(<http://smbus.org/specs/>)。

SMBus 有三类设备标准：

- 主设备：发送命令、产生时钟和终止发送设备；
- 从设备：接收或响应命令设备；

- **主机：**一个系统仅有一个主机，它提供与系统 CPU 的主接口。主机具有主-从机功能并必须支持 SMBus 提醒协议。

SMBus 与 I2C 的相似点：

- 总线协议都是两条线（一个时钟线 SCL 和一个数据线 SDA），再加一条可选的 SMBus 提醒线；
- 数据格式相似，SMBus 数据格式类似于 I2C 的 7 位地址格式（见图 8-2）；
- 都是主-从通信模式，且主设备提供时钟；
- 都支持多主机功能；

SMBus 和 I2C 的不同点：

**表 8-1 SMBus 与 I<sup>2</sup>C 的比较**

SMBus	I <sup>2</sup> C
最大传输速度 100KHz	最大传输速度 1MHz
最小传输速度 10KHz	无最小传输速度
固定逻辑电平	逻辑电平由 VDD 决定
不同的地址类型(保留的, 动态的等)	7 位、10 位和广播呼叫从地址类型
不同的总线协议(快速命令, 处理呼叫等)	无总线协议

### 8.3.7.2 SMBus 用途

SMBus 利用系统管理总线，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，为系统和电源管理相关的任务提供控制总线。

### 8.3.7.3 设备标识

在 SMBus 中，任意一个设备作为从设备时都有一个地址，叫从地址。

为了给每一个设备分配地址，必须有一个唯一的设备标识（UDID）分配给设备。

### 8.3.7.4 总线协议

SMBus 技术规范包括 8 个总线协议。想要了解关于 SMBus 的详细信息和地址类型请参考 SMBus 技术规范 V2.0(<http://smbus.org/specs/>)。用户可以决定使用哪些协议。

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。

*注：SMBus 不支持 Quick command 协议。*

### 8.3.7.5 地址解析协议 (ARP)

通过 SMBus 协议动态分配新的唯一地址给每个从设备来解决地址冲突，这是地址解析协议（ARP）。地址解析协议具有一下特性：

任何一个 SMBus 主设备都可以遍历总线；

使用 SMBus 物理层仲裁机制分配地址。当设备维持供电期间，分配的地址保持不变，协议也允许再断电后保留其地址。

地址分配之后，没有额外的 SMBus 打包开销（访问分配地址的设备与访问固定地址的设备所用的时间是一样的）。

### 8.3.7.6 SMBus 提醒模式

SMBus 提供了一个可选的中断信号 SMBALERT（与 SCL 和 SDA 一样，是一种有线与信号），设备使用该信号来扩展其控制能力，但需要牺牲一个引脚。SMBALERT 通常和 SMBus 广播呼叫地址结合使用。关于 SMBus 的消息有 2 个字节。

仅具有从机功能的设备可以设置 I2C\_CTRL1.SMBALERT 位以指示它要与主机通信。主机处理该中断并通过提醒响应地址 ARA（Alert Response Address，地址值为 0001100x）访问所有 SMBALERT 设备。只有那些将 SMBALERT 拉低的设备能应答 ARA。此状态是由 I2C\_STS1.SMBALERT 来标识的。从发送设备提供的 7 位设备地址被放在字节的 7 个最高位上，第八个位可以是‘0’或‘1’。

当多个设备的 SMBALERT 为低时，在地址传输过程中，最高优先级（地址越小优先级越高）可以通过标准仲裁赢得总线通信。如果确认从机地址，则设备的 SMBALERT 不再为低。如果报文传输完毕，设备的 SMBALERT 仍为低，表示主机将再次读取 ARA。没有采用 SMBALERT 信号时主机可以定期访问 ARA。

### 8.3.7.7 SMBus 通信流程

SMBus 的通讯流程和标准 I<sup>2</sup>C 的流程相似。如果要使用 SMBus 模式，还需要在程序中配置 SMBus 特定寄存器、并响应 SMBus 特定标志位、实现在 SMBus 手册中介绍的上层协议。

1. 首先，设置 I2C\_CTRL1.SMBMODE 位；并按照应用要求配置 I2C\_CTRL1.SMBTYPE 和 I2C\_CTRL1.ARPEN 位。如果 I2C\_CTRL1.ARPEN=1 且 I2C\_CTRL1.SMBTYPE=0，使用 SMB 设备默认地址；如果 I2C\_CTRL1.ARPEN=1 且 I2C\_CTRL1.SMBTYPE=1，使用 SMB 主设备头字段。
2. 为了支持 ARP 协议（I2C\_CTRL1.ARPEN=1），在 SMBus 主机模式下（I2C\_CTRL1.SMBTYPE=1），软件需要响应标志位 I2C\_STS2.SMBHADDR（在 SMBus 从机模式下，响应 I2C\_STS2.SMBDADDR 标志位），并实现 ARP 协议中的功能。
3. 为了支持 SMBus 警告模式，软件应该响应 I2C\_STS1.SMBALERT 标志位，并实现相应的功能。

### 8.3.8 噪声滤波

I<sup>2</sup>C 接口标准要求能够过滤在 SCL/SDA 线上的 50ns 的毛刺，因此设计了模拟滤波器和数字滤波器。模拟滤波器默认开启，也可以通过设置 I2C\_GFLTRCTRL.SCLAFENN/SDAAFENN 位禁用此功能。模拟滤波器通过配置 I2C\_GFLTRCTRL.SCLAFE/SDAAF 设置过滤毛刺的宽度 5ns,15ns,25ns,35ns。数字滤波器通过配置 I2C\_GFLTRCTRL.SCLDFW/SDADFW 为非 0 值去使能。其最大滤波宽度为(SCLDEW[3:0]或 SDADFW[3:0])\*T<sub>PCLK</sub>。使能数字滤波器将增加 SDA 线的保持时间，增加值为(SDADFW[3:0]+1)\*T<sub>PCLK</sub>。

## 8.4 调试模式

当微控制器进入调试模式（Cortex<sup>®</sup>-M0 核心处于停止状态）时，根据 PWR 模块中的 DBG\_CTRL.I2C1TIMOUT 和 DBG\_CTRL.I2C2TIMOUT 配置位，SMBUS 超时控制或者继续正常工作或者可以停止。详见 3.3.1 节。

## 8.5 中断请求

下表列出了所有的 I<sup>2</sup>C 中断请求：

表 8-2 I<sup>2</sup>C 中断请求

中断函数	中断事件	事件标志	设置控制位
I2C 全局中断	起始位已发送 (主)	STARTBF	EVTINTEN

中断函数	中断事件	事件标志	设置控制位
	地址已发送 (主) 或地址匹配 (从)	ADDRF	EVTINTEN 和 BUFINTEN
	10 位头段地址已发送 (主)	ADDR10F	
	收到停止位 (从)	STOPF	
	数据字节传输完成	BSF	
	接收缓冲区非空	RXDATNE	EVTINTEN 和 BUFINTEN
	发送缓冲区为空	TXDATE	
	总线错误	BUSERR	ERRINTEN
	仲裁丢失 (主)	ARLOST	
	应答失败	ACKFAIL	
	过载/欠载	OVERRUN	
	PEC 错误	PECERR	
	SMBus 提醒	SMBALERT	SCLLTOINTEN
	SCL 低超时	SCLLTO	
	SCL 高超时	SCLHTO	SCLHTOINTEN
	SDA 低超时	SDALTO	SDALTOINTEN

## 8.6 I2C 寄存器描述

可以用半字（16 位）或字（32 位）的方式操作这些外设寄存器。

### 8.6.1 I<sup>2</sup>C 寄存器总览

表 8-3 I<sup>2</sup>C 寄存器总览

Offset	Register	32	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
000h	I2C_CTRL1	Reserved	LTOSEL	HTOSEL	Reserved												Reserved	SWRESET	SMBALERT	PEC	ACKPOS	ACKEN	STOPGEN	STARTGEN	NOEXTEND	GCEN	PECEN	ARPEN	SMBTYPE	SMBMODE	EN					
004h	I2C_CTRL2	Reserved	LTOEN	HTOEN	SCLLTOINTEN	SCLHTOINTEN	SDALTOINTEN	Reserved												DMAEN	ERRINTEN	EVTINTEN	BUFINTEN	Reserved	Reserved	DMALAST	Reserved	CLKFREQ[6:0]								
008h	I2C_OADDR1	Reserved												ADDRMODE	Reserved						ADDR[9:8]			ADDR[7:1]				ADDR0								
00Ch	I2C_OADDR2	Reserved												Reserved						ADDR2[7:1]				DUALE												
010h	I2C_DAT	Reserved												Reserved						DATA[7:0]																
014h	I2C_STS1	Reserved				SCLLTO	SCLHTO	SDALTO	Reserved												Reserved	SMBALERT	TIMEOUT	PECERR	OVERRUN	BUSERR	ARLOST	ACKFAIL	Reserved	ADDR[0F]	TXDATE	RXDATNE	STOPF	BSF	ADDRF	STARTBF

018h	I2C_STS2	Reserved	PECVAL[7:0]			SMBHADDR	SMBDADDR	DUALFLAG	GCALLADDR	Reserved	TRF	MSMODE	BUSY
01Ch	I2C_CLKCTRL	Reserved	DUTY	FSMODE	Reserved	CLKCTRL[11:0]							
020h	I2C_TMRISE	Reserved	Reserved						TMRISE[5:0]				
028h	I2C_GFLTRCTRL	Reserved	SCLAFENN	Reserved	SCLAFW[1:0]	SDAAFENN	Reserved	SDAAFV[1:0]	SCLDFW[3:0]			SDADFW[3:0]	

## 8.6.2 I<sup>2</sup>C 控制寄存器 1 (I2C\_CTRL1)

地址偏移：0x00

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	LTOSEL		HTOSEL		Reserved										
	rw		rw												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SW RESET	SMB ALERT	PEC	ACKPOS	ACKEN	STOP GEN	START GEN	NO EXTEND	GCEN	PECEN	ARPEN	SMB TYPE	SMB MODE	EN	
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

位域	名称	描述
31	Reserved	保留，必须保持复位值
30:29	LTOSEL	低超时阈值选择（low timeout threshold selection） 00: 25ms 01: 100ms 10: 1s 11: 4s
28:27	HTOSEL	高超时阈值选择（hightimeout threshold selection） 00: 256us 01: 512us 10: 1ms 11: 128us
26:14	Reserved	保留，必须保持复位值
13	SWRESET	软件复位（Software reset） 在复位该位前确认 I <sup>2</sup> C 的引脚被释放，总线是空的。 0: I <sup>2</sup> C 模块不处于复位状态； 1: I <sup>2</sup> C 模块处于复位状态。 <i>注：该位可以用于 I2C_STS2.BUSY 位为‘1’，在总线上又没有检测到停止条件时。</i>
12	SMBALERT	SMBus 提醒（SMBus alert）

位域	名称	描述
		软件可以设置或清除该位；当 I2C_CTRL1.EN=0 时，由硬件清除。 0：释放 SMBAlert 引脚使其变高。提醒响应地址头紧跟在 NACK 信号后面； 1：驱动 SMBAlert 引脚使其变低。提醒响应地址头紧跟在 ACK 信号后面。
11	PEC	数据包出错检测（Packet error checking） 软件可以设置或清除该位；当传送完 PEC 后或检测到起始或停止条件时或当 I2C_CTRL1.EN=0 时，硬件均会将其清除。 0：无 PEC 传输 1：PEC 传输 <i>注：仲裁丢失时，PEC 的计算失效。</i>
10	ACKPOS	应答/PEC 位置（用于数据接收）（Acknowledge/PEC Position（for data reception）） 软件可以设置或清除该位，或当 I2C_CTRL1.EN=0 时，由硬件清除。 0：I2C_CTRL1.ACKEN 位决定是否向当前正在接收的字节发送 ACK；I2C_CTRL1.PEC 位表示当前移位寄存器中的字节为 PEC。 1：I2C_CTRL1.ACKEN 位决定是否向下一个接收到的字节发送 ACK；I2C_CTRL1.PEC 位指示移位寄存器中接收到的下一个字节是 PEC。 <i>注：ACKPOS 位只能用在 2 字节的接收配置中，必须在接收数据之前配置。</i> 为了 NACK 第 2 个字节，I2C_CTRL1.ACKEN 位必须在 I2C_STS1.ADDRF 位清零后清零。 为了检测第 2 个字节的 PEC，必须在配置 ACKPOS 位之后，ADDR 延展事件时设置 I2C_CTRL1.PEC 位。
9	ACKEN	应答使能（Acknowledge enable） 软件可以设置或清除该位，或当 I2C_CTRL1.EN=0 时，由硬件清除。 0：无应答返回； 1：在接收到一个字节后返回一个应答（匹配的地址或数据）。
8	STOPGEN	停止条件产生（Stop generation） 软件可以设置或清除该位；或当检测到停止条件时，由硬件清除；当检测到 SMBus 超时错误时，硬件将其置位。 在主模式下： 0：无停止条件产生； 1：在当前字节传输或在当前起始条件发出后产生停止条件。 在从模式下： 0：无停止条件产生； 1：在当前字节传输后释放 SCL 和 SDA 线。 <i>注：当设置了 STOPGEN、STARTGEN 或 PEC 位，在硬件清除这个位之前，软件不要执行任何对 I2C_CTRL1 的写操作；否则有可能会第 2 次设置 STOPGEN、STARTGEN 或 PEC 位。</i>
7	STARTGEN	起始条件产生（Start generation） 软件可以设置或清除该位，当起始条件发出后或 I2C_CTRL1.EN=0 时，由硬件清除。 0：无起始条件产生； 1：产生起始条件。
6	NOEXTEND	禁止时钟延长（从模式）（Clock stretching disable（Slave mode）） 该位决定在从机模式下数据未就绪（I2C_STS1.ADDRF 或 I2C_STS1.BSF 标志置位）时是否拉低 SCL。通过软件复位清零。 0：允许时钟延长； 1：禁止时钟延长。

位域	名称	描述
5	GCEN	广播呼叫使能 (General call enable) 0: 禁止广播呼叫。不应答(NACK)地址 00h; 1: 允许广播呼叫。以应答(ACK)地址 00h。
4	PECEN	PEC 使能 (PEC enable) 0: 禁止 PEC 模式; 1: 开启 PEC 模式。
3	ARPEN	ARP 使能 (ARP enable) 0: 禁止 ARP; 1: 使能 ARP。 如果 I2C_CTRL1.SMBTYPE=0, 使用 SMBus 设备的默认地址。 如果 I2C_CTRL1.SMBTYPE=1, 使用 SMBus 的主地址。
2	SMBTYPE	SMBus 类型 (SMBus type) 0: SMBus 设备; 1: SMBus 主机。
1	SMBMODE	SMBus 模式 (SMBus mode) 0: I2C 模式; 1: SMBus 模式。
0	EN	I <sup>2</sup> C 模块使能 (Peripheral enable) 0: 禁用 I <sup>2</sup> C 模块; 1: 启用 I <sup>2</sup> C 模块。 <i>注: 如果清除该位时通讯正在进行, 在当前通讯结束后, I2C 模块被禁用并返回空闲状态。 由于在通讯结束后发生 EN=0, 所有的位被清除。 在主模式下, 通讯结束之前, 绝不能清除该位。</i>

### 8.6.3 I<sup>2</sup>C 控制寄存器 2 (I2C\_CTRL2)

地址偏移: 0x04

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved	LTOEN	HTOEN	SCLLTO INTEN	SCLHTO INTEN	SDALTO INTEN	Reserved										
	rw	rw	rw	rw	rw											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
DMAEN	ERRINT EN	EVTINT EN	BUFINT EN	Reserved			DMA LAST	Reserved	CLKFREQ[6:0]							
rw	rw	rw	rw				rw		rw							

位域	名称	描述
31	Reserved	保留, 必须保持复位值
30	LTOEN	低超时功能使能 (low timeout function enable) 0: 禁用; 1: 使能;

位域	名称	描述
		注: I2C 模式下不需要超时报错时可禁用此位。
29	HTOEN	高超时功能使能 (high timeout function enable) 0: 禁用; 1: 使能; 注: I2C 模式下不需要超时报错时可禁用此位。
28	SCLLTOINTEN	SCL 低超时错误中断使能 (SCL low timeout error interrupt enable) 0: I2C_STS1.SCLLTO = 1 时, 不产生中断; 1: I2C_STS1.SCLLTO = 1 时, 产生中断;
27	SCLHTOINTEN	SCL 高超时错误中断使能 (SCL high timeout error interrupt enable) 该位由软件写'0'清除, 或在 I2C_CTRL1.EN=0=0 时由硬件清除。 0: I2C_STS1.SCLHTO = 1 时, 不产生中断; 1: I2C_STS1.SCLHTO = 1 时, 产生中断; 注: SMBUS 模式下, 该位同时用作 SCL 累计低电平超时中断使能
26	SDALTOINTEN	SDA 低超时错误中断使能 (SDA low timeout error interrupt enable) 该位由软件写'0'清除, 或在 I2C_CTRL1.EN=0=0 时由硬件清除。 0: I2C_STS1.SDALTO = 1 时, 不产生中断; 1: I2C_STS1.SDALTO = 1 时, 产生中断;
25:16	Reserved	保留, 必须保持复位值
15	DMAEN	DMA 请求使能 (DMA requests enable) 0: 禁止 DMA 请求; 1: 使能 DMA 请求。
14	ERRINTEN	出错中断使能 (Error interrupt enable) 0: 禁止出错中断; 1: 允许出错中断。 在下列条件下, 将产生该中断: I2C_STS1.BUSERR = 1 I2C_STS1.ARLOST = 1 I2C_STS1.ACKFAIL = 1 I2C_STS1.OVERRUN = 1 I2C_STS1.PECERR = 1 I2C_STS1.SMBALERT = 1
13	EVTINTEN	事件中断使能 (Event interrupt enable) 0: 禁止事件中断 1: 允许事件中断 在下列条件下, 将产生该中断: I2C_STS1.STARTBF = 1 (主模式) I2C_STS1.ADDRF = 1 (主/从模式) I2C_STS1.ADD10F = 1 (主模式) I2C_STS1.STOPF = 1 (从模式) I2C_STS1.BSF = 1, 但是没有 I2C_STS1.TXDATE 或 I2C_STS1.RXDATNE 事件 如果 I2C_STS1.BUFINTEN = 1, I2C_STS1.TXDATE 标志为 1 如果 I2C_STS1.BUFINTEN = 1, I2C_STS1.RXDATNE 标志为 1
12	BUFINTEN	缓冲器中断使能 (Buffer interrupt enable)

位域	名称	描述
		0: 当 I2C_STS1.TXDATE=1 或 I2C_STS1.RXDATNE=1 时, 不产生任何中断; 1: 当 I2C_STS1.TXDATE=1 或 I2C_STS1.RXDATNE=1 时 (I2C_CTRL2.EVTINTEN=1), 产生事件中断 (不管 DMAEN 是何种状态)。
11:9	Reserved	保留, 必须保持复位值
8	DMALAST	DMA 最后一次传输 (DMA last transfer) 0: 下一次 DMA 的 EOT 不是最后的传输; 1: 下一次 DMA 的 EOT 是最后的传输。 <i>注: 该位在主接收模式使用, 使得在最后一次接收数据时可以产生一个 NACK。</i>
7	Reserved	保留, 必须保持复位值
6:0	CLKFREQ[6:0]	I <sup>2</sup> C 模块时钟频率 (Peripheral clock frequency) CLKFREQ[6:0] 应该为 APB 时钟频率以产生正确的时序: 0000000: 禁用 0000001: 禁用 0000010: 2MHz 0000011: 3MHz ... 1000000: 64MHz 1000001~1111111: 禁用

## 8.6.4 I<sup>2</sup>C 自身地址寄存器 1 (I2C\_OADDR1)

地址偏移: 0x08

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR MODE	Reserved	Reserved				ADDR[9:8]		ADDR[7:1]					ADDR0		
rw						rw		rw					rw		

位域	名称	描述
15	ADDRMODE	寻址模式 (从模式) (Addressing mode (slave mode)) 0: 7 位从地址 (不响应 10 位地址); 1: 10 位从地址 (不响应 7 位地址)。
14	Reserved	必须始终由软件保持为‘1’。
13:10	Reserved	保留, 必须保持复位值
9:8	ADDR[9:8]	接口地址 (Interface address) 地址的 9~8 位。 <i>注: 7 位地址模式时不用关心</i>
7:1	ADDR[7:1]	接口地址 (Interface address) 地址的 7~1 位。
0	ADDR0	接口地址 (Interface address) 地址第 0 位。 <i>注: 7 位地址模式时不用关心</i>

## 8.6.5 I<sup>2</sup>C 自身地址寄存器 2 (I2C\_OADDR2)

地址偏移: 0x0C

复位值: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ADDR2[7:1]							DUALEN
								rw							rw

位域	名称	描述
15:8	Reserved	保留, 必须保持复位值
7:1	ADDR2[7:1]	接口地址 (Interface address) 在双地址模式下地址的 7~1 位。
0	DUALEN	双地址模式使能位 (Dual addressing mode enable) 0: 不使能双地址模式, 只有 OADDR1 被识别; 1: 使能双地址模式, OADDR1 和 OADDR2 都被识别。 <i>注: 仅 7 位地址模式有效</i>

## 8.6.6 I<sup>2</sup>C 数据寄存器 (I2C\_DAT)

地址偏移: 0x10

复位值: 0x0000

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

位域	名称	描述
15:8	Reserved	保留, 必须保持复位值
7:0	DATA[7:0]	8 位数据寄存器 (8-bit data register) 发送或接收数据 <i>注意: 从模式下。地址不会被拷贝到数据寄存器</i> <i>注意: 如果 I2C_STS1.TXDATE = 0, 数据仍然会被写入数据寄存器</i> <i>注意: 如果在处理 ACK 时, 发生了 ARLOST 事件, 接收到的字节不会被拷贝到数据寄存器, 因此不能读到它。</i>

## 8.6.7 I<sup>2</sup>C 状态寄存器 1 (I2C\_STS1)

地址偏移: 0x14

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						SCLLTO	SCLHTO	SDALTO	Reserved						
						rc_w0	rc_w0	rc_w0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SMBALERT	TIMEOUT	PECERR	OVERRUN	BUSERR	ARLOST	ACKFAIL	Reserved	ADDR10F	TXDATE	RXDATNE	STOPF	BSF	ADDRF	STARTBF
rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0		r	r	r	r	r	r	r

位域	名称	描述
31:26	Reserved	保留，必须保持复位值
25	SCLLTO	SCL 低超时错误（SCL low timeout error） 该位由软件写‘0’清除，或在 I2C_CTRL1.EN=0=0 时由硬件清除。 0：无超时错误； 1：超时错误发生；
24	SCLHTO	SCL 高超时错误（SCL high timeout error） 该位由软件写‘0’清除，或在 I2C_CTRL1.EN=0=0 时由硬件清除。 0：无超时错误； 1：超时错误发生； <i>注：SMBUS 模式下，该位同时用作 SCL 累计低电平超时标志</i>
23	SDALTO	SDA 低超时错误（SDA low timeout error） 该位由软件写‘0’清除，或在 I2C_CTRL1.EN=0=0 时由硬件清除。 0：无超时错误； 1：超时错误发生；
22:15	Reserved	保留，必须保持复位值
14	SMBALERT	SMBus 提醒（SMBus alert） 该位由软件写‘0’清除，或在 I2C_CTRL1.EN =0 时由硬件清除。 0：无 SMBus 提醒（主模式）或没有 SMBAlert 响应地址头序列（从模式）； 1：在引脚上产生 SMBAlert 提醒事件（主模式）或收到 SMBAlert 响应地址（从模式）；
13	TIMEOUT	超时标志位 任何一种超时的发生都会使该位被置 1，该位由软件写‘0’清除，或在 I2C_CTRL1.EN =0 时由硬件清除。 0：无超时错误； 1：有超时错误发生；
12	PECERR	在接收时发生 PEC 错误（PEC Error in reception） 该位由软件写‘0’清除，或在 I2C_CTRL1.EN =0 时由硬件清除。 0：无 PEC 错误：接收到 PEC 后接收器返回 ACK（如果 I2C_CTRL1.ACKEN=1） 1：PEC 错误：接收到 PEC 后接收器返回 NACK（不管 I2C_CTRL1.ACKEN 是否置位）
11	OVERRUN	过载/欠载（Overrun/Underrun） 该位由软件写‘0’清除，或在 I2C_CTRL1.EN=0 时由硬件清除。 0：无过载/欠载； 1：出现过载/欠载。 当 I2C_CTRL1.NOEXTEND=1 时，在从模式下该位被硬件置位。同时：

位域	名称	描述
		在接收模式中当接收到一个新的字节时，如果数据寄存器里的内容还未被读出，则发生过载错误，新接收的字节将会丢失。 在发送模式中要发送一个新的字节时，却没有新的数据写入数寄存器，将发生欠载错误，同样的数据将会被发送两次。
10	BUSERR	总线错误（Bus error） 该位由软件写‘0’清除，或在 I2C_CTRL1.EN=0 时由硬件清除。 0：无起始或停止条件出错； 1：起始或停止条件出错。
9	ARLOST	仲裁丢失（主模式） 该位由软件写‘0’清除，或在 I2C_CTRL1.EN=0 时由硬件清除。 0：没有仲裁丢失； 1：仲裁丢失。 当接口失去对总线的控制时，硬件将置该位为‘1’。在 ARLOST 事件之后，I <sup>2</sup> C 接口自动切换回从模式(I2C_STS2.MSMODE=0)。 <i>注：在 SMBUS 模式下，在从模式下对数据的仲裁仅仅发生在数据阶段，或应答传输区间（不包括地址的应答）。</i>
8	ACKFAIL	应答失败（Acknowledge failure） 该位由软件写‘0’清除，或在 I2C_CTRL1.EN=0 时由硬件清除。 0：没有应答失败； 1：应答失败。
7	Reserved	保留，必须保持复位值
6	ADDR10F	10 位头序列已发送（主模式） 软件读取 STS1 寄存器后，对 DAT 寄存器的写操作将清除该位，或当 I2C_CTRL1.EN=0 时，硬件清除该位。 0：没有 ADD10F 事件发生； 1：主设备已经将第一个地址字节发送出去。 在 10 位地址模式下，当主设备已经将第一个地址字节发送出去时，硬件将该位置‘1’。 <i>注：收到一个 NACK 后，I2C_STS1.ADD10F 位不被置位。</i>
5	TXDATE	数据寄存器为空（发送时） 软件写数据到 DAT 寄存器可清除该位；或在发生一个起始或停止条件后，或当 I2C_CTRL1.EN=0 时由硬件自动清除。 0：数据寄存器非空； 1：数据寄存器空。 在发送数据时，数据寄存器为空时该位被置‘1’，在发送地址阶段不设置该位。 如果收到一个 NACK，或下一个要发送的字节是 PEC（I2C_CTRL1.PEC=1），该位不被置位。 <i>注：在写入第 1 个要发送的数据后，或设置了 BSF 时写入数据，都不能清除 TXDATE 位，这是因为数据寄存器仍然为空。</i>
4	RXDATNE	数据寄存器非空（接收时） 软件对数据寄存器的读写操作清除该位，或当 I2C_CTRL1.EN=0 时由硬件清除。 0：数据寄存器为空； 1：数据寄存器非空。 在接收时，当数据寄存器不为空，该位被置‘1’。在接收地址阶段，该位不被置位。

位域	名称	描述
		在发生 ARLOST 事件时, RXDATNE 不被置位。 <i>注: 当设置了 BSF 时, 读取数据不能清除 RXDATNE 位, 因为数据寄存器仍然为满。</i>
3	STOPF	停止条件检测位 (从模式) 软件读取 STS1 寄存器后, 对 I2C_CTRL1 寄存器的写操作将清除该位, 或当 I2C_CTRL1.EN=0 时, 硬件清除该位。 0: 没有检测到停止条件; 1: 检测到停止条件。 在一个应答之后, 当从设备在总线上检测到停止条件时, 硬件将该位置‘1’。 <i>注: 在收到 NACK 后, I2C_STS1.STOPF 位不被置位。</i>
2	BSF	字节传输结束 (Byte transfer finished) 在软件读取 STS1 寄存器后, 对数据寄存器的读或写操作将清除该位; 或在传输中发送一个起始或停止条件后, 或当 I2C_CTRL1.EN =0 时, 由硬件清除该位。 0: 字节传输未完成; 1: 字节传输结束。 当 I2C_CTRL1.NOEXTEND =0 时, 在下列情况下硬件将该位置‘1’: 在接收时, 当收到一个新字节 (包括 ACK 脉冲) 且数据寄存器还未被读取 (I2C_STS1.RXDATNE =1)。 在发送时, 当一个新数据将被发送且数据寄存器还未被写入新的数据 (I2C_STS1.TXDATE =1)。 <i>注: 在收到一个 NACK 后, BSF 位不会被置位。</i> 如果下一个要传输的字节是 PEC (I2C_STS2.TRF 为‘1’, 同时 I2C_CTRL1.PEC 为‘1’), BSF 位不会被置位。
1	ADDRF	地址已被发送 (主模式) /地址匹配 (从模式) 在软件读取 STS1 寄存器后, 对 STS2 寄存器的读操作将清除该位, 或当 I2C_CTRL1.EN=0 时, 由硬件清除该位。 0: 地址不匹配或没有收到地址 (从模式), 地址发送未完成 (主模式); 1: 收到的地址匹配 (从模式), 地址发送完成 (主模式)。 在主模式下: 7 位地址模式时, 当收到地址的 ACK 后该位被置‘1’, 10 位地址模式时, 当收到地址的第二个字节的 ACK 后该位被置‘1’。 在从模式下: 当收到的从地址与 OADDR 寄存器中的内容相匹配, 或广播呼叫或 SMBus 设备默认地址或 SMBus 主机或 SMBus 提醒被识别时, 硬件就将该位置‘1’ (当对应的设置被使能时)。 <i>注: 在收到 NACK 后, I2C_STS1.ADDRF 位不会被置位。</i>
0	STARTBF	起始位 (主模式) 软件读取 I2C_STS1 寄存器后, 写数据寄存器的操作将清除该位, 或当 I2C_CTRL1.EN=0 时, 硬件清除该位。 0: 未发送起始条件; 1: 起始条件已发送。 当发送出起始条件时该位被置‘1’。

## 8.6.8 I<sup>2</sup>C 状态寄存器 2 (I2C\_STS2)

地址偏移: 0x18

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PECVAL[7:0]							SMBH ADDR	SMBD ADDR	DUAL FLAG	GCALL ADDR	Reserved	TRF	MS MODE	BUSY	
r							r	r	r	r		r	r	r	

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:8	PECVAL[7:0]	数据包出错检测（Packet error checking register） 当 I2C_CTRL1.PECEN=1 时，PECVAL[7:0]存放内部的 PEC 值。
7	SMBHADDR	SMBus 主机头地址（从模式） 在产生一个停止条件或一个重复的起始条件时，或 I2C_CTRL1.EN =0 时，硬件将该位清除。 0: 未收到 SMBus 主机的地址； 1: 当 I2C_CTRL1.SMBTYPE=1 且 I2C_CTRL1.ARPEN=1 时，收到 SMBus 主机地址。
6	SMBDADDR	SMBus 设备默认地址（从模式） 在产生一个停止条件或一个重复的起始条件时，或 I2C_CTRL1.EN=0 时，硬件将该位清除。 0: 未收到 SMBus 设备的默认地址； 1: 当 I2C_CTRL1.ARPEN=1 时，收到 SMBus 设备的默认地址。
5	DUALFLAG	双标志（从模式） 在产生一个停止条件或一个重复的起始条件时，或 I2C_CTRL1.EN =0 时，硬件将该位清除。 0: 接收到的地址与 OADDR1 内的内容相匹配； 1: 接收到的地址与 OADDR2 内的内容相匹配。
4	GCALLADDR	广播呼叫地址（从模式） 在产生一个停止条件或一个重复的起始条件时，或 I2C_CTRL1.EN=0 时，硬件将该位清除。 0: 未收到广播呼叫地址； 1: 当 I2C_CTRL1.GCEN =1 时，收到广播呼叫的地址。
3	Reserved	保留，必须保持复位值
2	TRF	发送/接收（Transmitter/receiver） 在检测到停止条件（I2C_STS1.STOPF =1）、重复的起始条件或总线仲裁丢失（I2C_STS1.ARLOST =1）后，或当 I2C_CTRL1.EN=0 时，硬件将其清除。 0: 数据接收模式；

位域	名称	描述
		1: 数据发送模式; 在整个地址传输阶段的结尾, 该位根据地址字节的 R/W 位来设定。
1	MSMODE	主从模式 (Master/slave) 当总线上检测到一个停止条件、仲裁丢失 (I2C_STS1.ARLOST =1) 时、或当 I2C_CTRL1.EN =0 时, 硬件清除该位。 0: 从模式; 1: 主模式。 当接口处于主模式 (I2C_CTRL1.STARTBF=1) 时, 硬件将该位置位;
0	BUSY	总线忙 (Bus busy) 当检测到一个停止条件时, 硬件将该位清除。 0: 在总线上无数据通讯; 1: 在总线上正在进行数据通讯。 在检测到 SDA 或 SCL 为低电平时, 硬件将该位置‘1’; <i>注: 该位指示当前正在进行的总线通讯, 当接口被禁用 (I2C_CTRL1.EN =0) 时该信息仍然被更新。</i>

## 8.6.9 I<sup>2</sup>C 时钟控制寄存器 (I2C\_CLKCTRL)

地址偏移: 0x1C

复位值: 0x0000

*注: CLKCTRL 寄存器只有在关闭 I<sup>2</sup>C 时 (I2C\_CTRL1.EN =0) 才能设置。*

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

位域	名称	描述
15	DUTY	快速模式占空比 0: Tlow/Thigh= 2; 1: Tlow/Thigh= 16/9
14	FSMODE	I2C 主模式选择 0: I2C 标准模式(占空比默认 1/1); 1: I2C 快速模式(占空比可配置)
13:12	Reserved	保留, 必须保持复位值
11:0	CLKCTRL[11:0]	快速/标准模式下的时钟控制分频系数 (主模式) 该分频系数用于设置主模式下的 SCL 时钟。 <ul style="list-style-type: none"> <li>■ 如果 duty cycle = Tlow/Thigh = 1/1:  <math display="block">\text{CLKCTRL} = f_{\text{PCLK1}}(\text{Hz})/100000/2</math> <math display="block">\text{Tlow} = \text{CLKCTRL} \times \text{TPCLK1}</math> <math display="block">\text{Thigh} = \text{CLKCTRL} \times \text{TPCLK1}</math> </li> <li>■ 如果 duty cycle = Tlow/Thigh = 2/1:</li> </ul>

位域	名称	描述
		$CLKCTRL = f_{PCLK1}(Hz)/100000/3$ $T_{low} = 2 \times CLKCTRL \times TPCLK1$ $T_{high} = CLKCTRL \times TPCLK1$ ■ 如果 duty cycle = $T_{low}/T_{high} = 16/9$ : $CLKCTRL = f_{PCLK1}(Hz)/100000/25$ $T_{low} = 16 \times CLKCTRL \times TPCLK1$ $T_{high} = 9 \times CLKCTRL \times TPCLK1$ 例如, 如果 $f_{PCLK1}(Hz) = 8MHz$ , duty cycle = $1/1$ , $CLKCTRL = 8000000/100000/2 = 0x28$ 。 注意: 1. 允许设定的最小值为 $0x04$ , 在快速 DUTY 模式下允许的最小值为 $0x01$ 。 2. $T_{high} = T_{r(SCL)} + T_{w(SCLH)}$ , 详见数据手册中对这些参数的定义。 3. $T_{low} = T_{f(SCL)} + T_{w(SCLL)}$ , 详见数据手册中对这些参数的定义。

### 8.6.10 I<sup>2</sup>C 上升时间寄存器 (I2C\_TMRISE)

地址偏移: 0x20

复位值: 0x0002

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

位域	名称	描述
15:6	Reserved	保留, 必须保持复位值
5:0	TMRISE[5:0]	在快速/标准模式下的最大上升时间 (主模式) 这些位必须设置为 I <sup>2</sup> C 总线规范里给出的最大的 SCL 上升时间, 增长步幅为 1。 例如, 标准模式中最大允许 SCL 上升时间为 1000ns。如果 I2C_CTRL2.CLKFREQ[5:0] 中的值等于 0x08(8MHz)且 $TPCLK1=125ns$ , 故 TMRISE[5:0] 中必须写入 09h ( $1000ns/125ns + 1$ )。 如果结果不是一个整数, 则将整数部分写入 TMRISE[5:0] 以确保 $t_{HIGH}$ 参数。 注: 只有当 I2C 禁用 ( $I2C_CTRL1.EN=0$ ) 时才能配置 TMRISE[5:0]

### 8.6.11 I<sup>2</sup>C 滤波控制寄存器 (I2C\_GFLTRCTRL)

地址偏移: 0x28

复位值: 0x0000

 注: GFLTRCTRL 寄存器只有在关闭 I<sup>2</sup>C 时 ( $I2C_CTRL1.EN=0$ ) 才能设置。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCLAF ENN	Reserved	SCLAFW[1:0]		SDAAF ENN	Reserved	SDAAFW[1:0]		SCLDFW[3:0]			SDADFW[3:0]				
rw		rw		rw		rw		rw		rw		rw		rw	

位域	名称	描述
15	SCLAFENN	SCL 模拟滤波器使能 0: 使能 1: 禁用
14	Reserved	Reserved
13:12	SCLAFW[1:0]	SCL 模拟滤波器宽度选择 00: 5ns 01: 15ns 10: 25ns 11: 35ns
11	SDAAFENN	SDA 模拟滤波器使能 0: 使能 1: 禁用
10	Reserved	Reserved
9:8	SDAAFV[1:0]	SDA 模拟滤波器宽度选择: 00: 5ns 01: 15ns 10: 25ns 11: 35ns
7:4	SCLDFW[3:0]	SCL 数字滤波器宽度选择 0000: 禁用 SCL 数字滤波器 其他值: 滤波宽度为 SCLDFW * T <sub>PCLK</sub>
3:0	SDADFW[3:0]	SDA 数字滤波器宽度选择 0000: 禁用 SDA 数字滤波器 其他值: 滤波宽度为 SDADFW * T <sub>PCLK</sub>

## 9 通用异步收发器(UART)

### 9.1 简介

通用异步收发器（UART）是一种全双工串行数据交换接口，支持异步通信。可灵活配置，以便于与多种外部设备进行全双工数据交换。

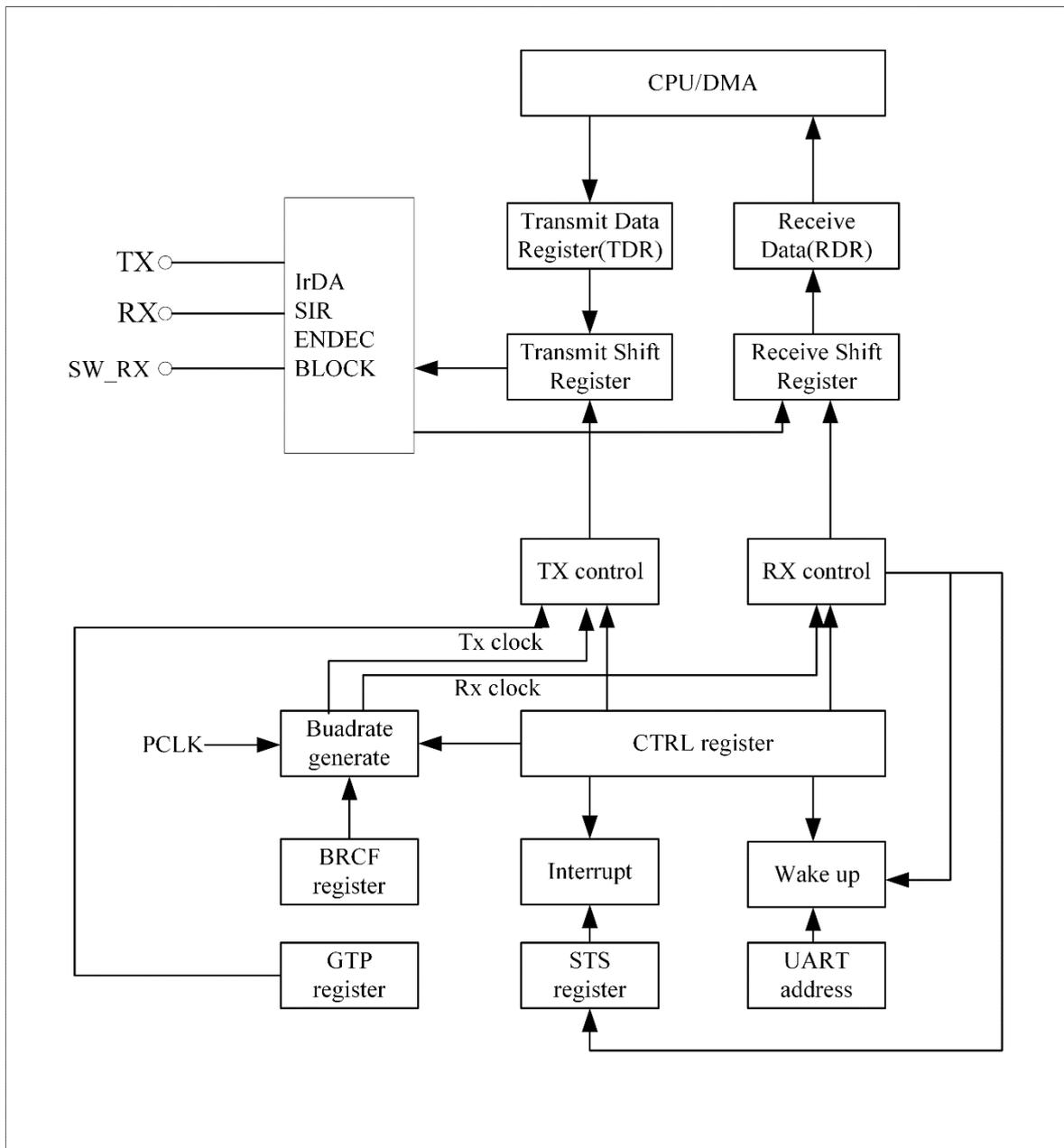
UART 接口发送与接收波特率可配置，也支持通过 DMA 进行连续通信。UART 还支持多处理器通信、LIN 模式、单线半双工通信、IrDA SIR ENDEC 功能。

### 9.2 主要特性

- 支持全双工通信
- 支持单线半双工通信
- 波特率可配置，最高波特率可达 4Mbit/s
- 支持 8bit 或 9bit 数据帧
- 支持 1bit 或 2bit 停止位
- 支持硬件生成校验位及校验位检查
- 支持 TX/RX swap 功能
- 支持 DMA 收发
- 支持 RS-485
- UART3 支持低功耗唤醒
- 支持多处理器通信：如果地址不匹配，则进入静默模式，可通过空闲总线检测或地址标识唤醒
- 支持串行红外协议（IrDA SIR）编码与解码，提供正常与低功耗两种运行模式
- 支持 LIN 模式
- 支持多钟错误检测：数据溢出错误、帧错误、噪声错误、检验错误
- 支持多个中断请求：发送数据寄存器为空、发送完成、数据已接收、数据溢出、总线空闲、检验错误、LIN 模式断开帧检测、以及多缓冲区通信中的噪声标志/溢出错误/帧错误、接收超时

### 9.3 功能框图

图 9-1 UART 框图



### 9.4 功能描述

如图 9-1 所示, UART 的双向通信都需要使用 RX 和 TX 引脚与外部器件连接。其中 TX 为数据发送引脚(输出), 当发送功能使能但没有发送数据时, TX 引脚输出高电平, 当发送功能被禁用时, TX 引脚为普通 IO 端口, 状态由应 IO 配置决定。RX 为数据接收引脚(输入), 接收数据时采用了过采样技术。

当设备作为发送端时, 通过 TX 引脚发送数据, 作为接收端时则通过 RX 引脚接收数据。当没有数据收发时, 总线处于空闲状态。数据帧格式为: 1 个起始位+8 或 9 位数据 (最低有效位在前)+1 个检验位 (可选) + 0.5, 1, 1.5 或 2 个停止位。

使用分数波特率发生器来配置发送与接收波特率。

### 9.4.1 UART 帧格式

起始位：1 位，低电平有效

数据位：可通过 UART\_CTRL1.WL 配置为 8 或 9 位，最低有效位在前。

停止位：高电平有效。

空闲帧：全部由‘1’组成的一个完整的数据帧，包括起始位。后跟包含数据的数据帧的起始位。

断开帧：全部由‘0’组成的一个完整的数据帧，包括停止位。在断开帧结束后，发送端再插入 1 或 2 个停止位来应答起始位。

图 9-2 字长=8 设置

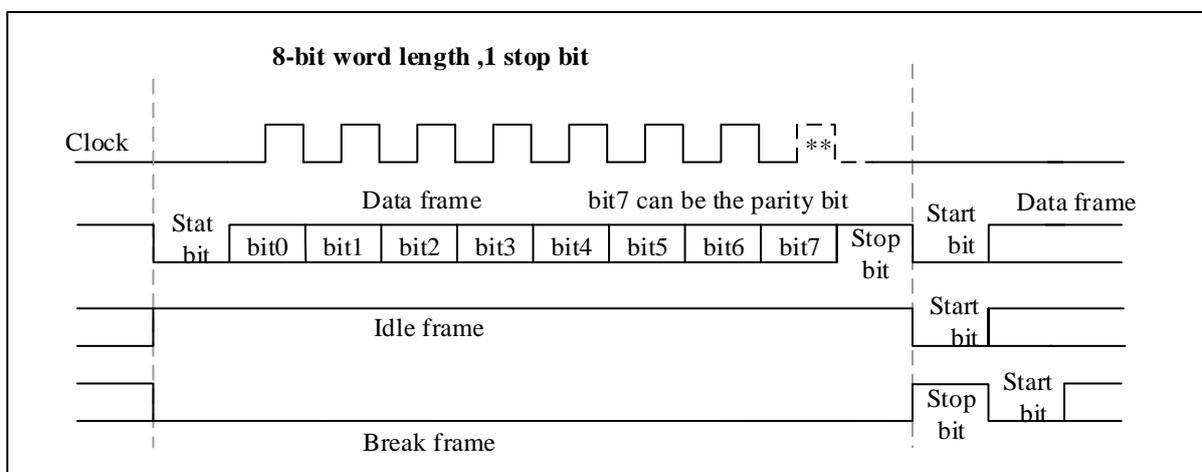
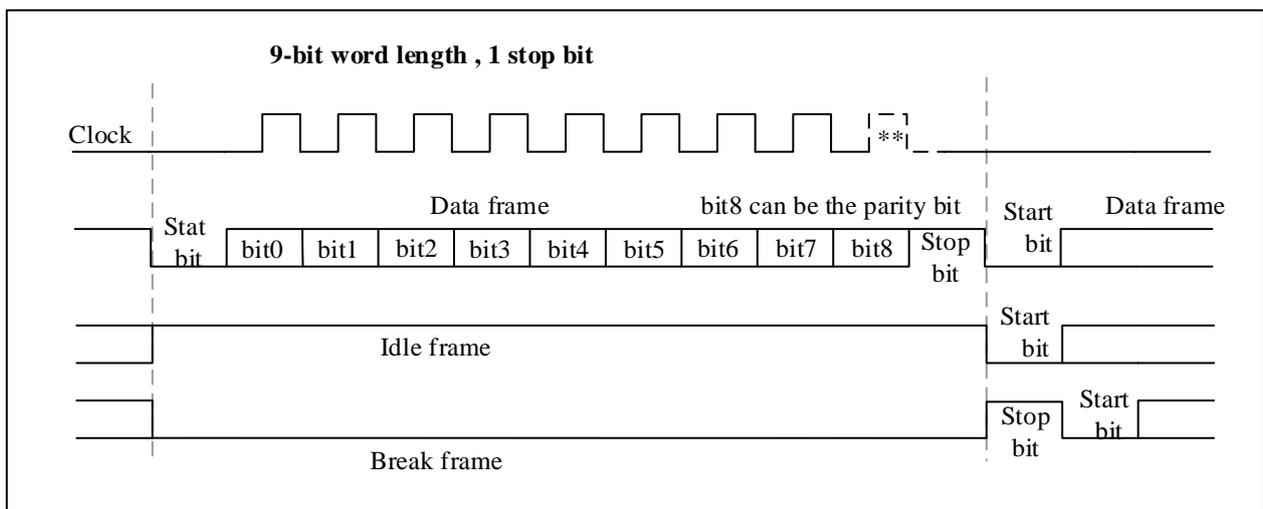


图 9-3 字长=9 设置



### 9.4.2 发送器

当发送功能使能后，进入移位寄存器中的数据通过 TX 引脚输出。

### 9.4.2.1 空闲帧

UART\_CTRL1.TXEN 置 1 后，UART 会在发送数据之前发送一个空闲帧。

### 9.4.2.2 字符发送

在空闲帧结束后，字符可正常发送。在每个字符发送前，先发送一个起始位（低电平）。发送器根据数据长度配置发送 8 位或 9 位数据，其中最低有效位先发送。如果在数据传输时 UART\_CTRL1.TXEN 被清零，将导致波特率计数器停止计数，从而破坏正在传输的数据。

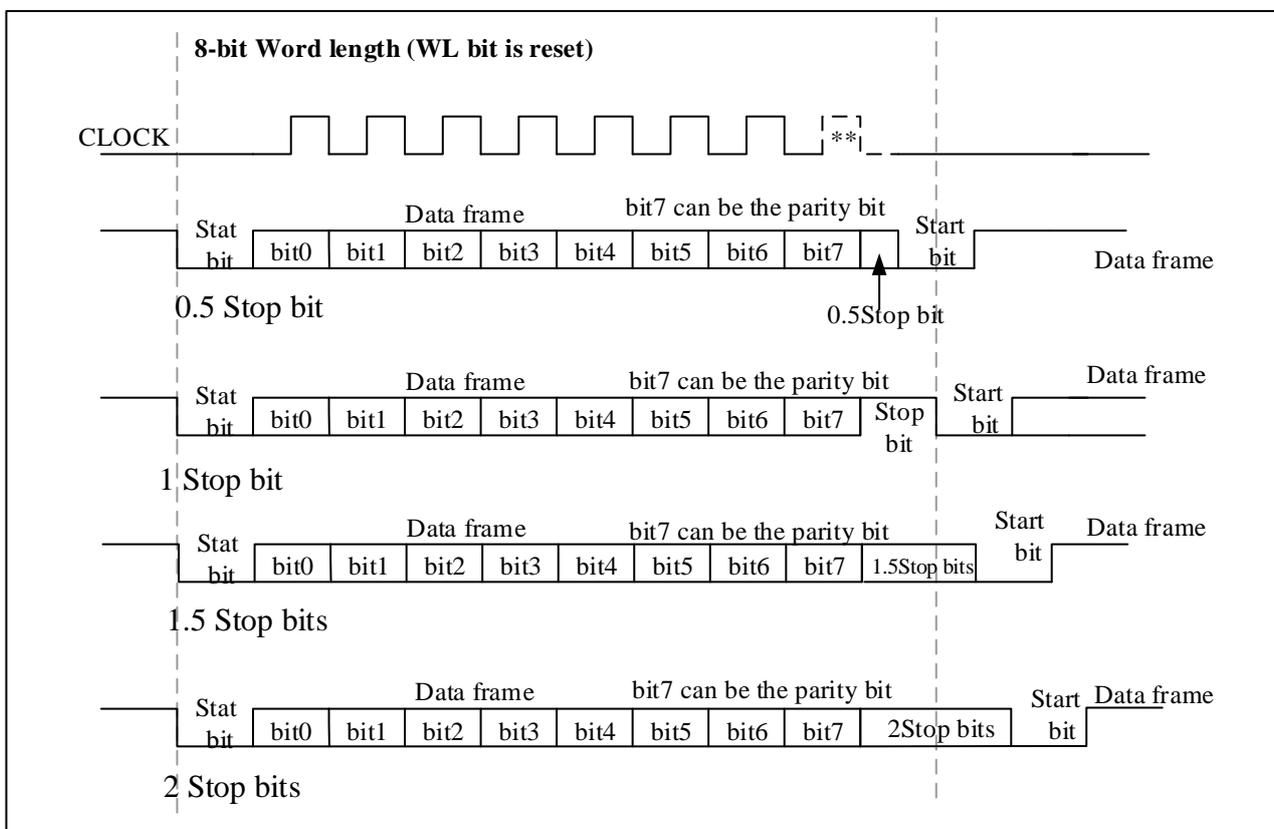
### 9.4.2.3 停止位

字符发送完成后，发送器自动发送停止位。停止位位数可通过 UART\_CTRL2.STPB[1:0]配置。

表 9-1 停止位配置

UART_CTRL2.STPB[1:0]	停止位长度 (位)	功能描述
00	1	默认
01	0.5	
10	2	用于常规 UART 模式、单线模式以及调制解调器模式。
11	1.5	

图 9-4 停止位配置



### 9.4.2.4 断开帧

可通过置位 `UART_CTRL1.SDBRK` 来发送 1 个断开帧。当数据长度为 8 位时，断开帧由 10 位低电平组成，当数据长度为 9 位时，断开帧由 11 位低电平组成。断开帧结束后将插入一位停止位（高电平）。

断开帧发送完成后，`UART_CTRL1.SDBRK` 被硬件清零，同时自动发送停止位。因此，如果要连续发送断开帧，必须在前一个断开帧与停止位发送完成后再次置位 `UART_CTRL1.SDBRK`。

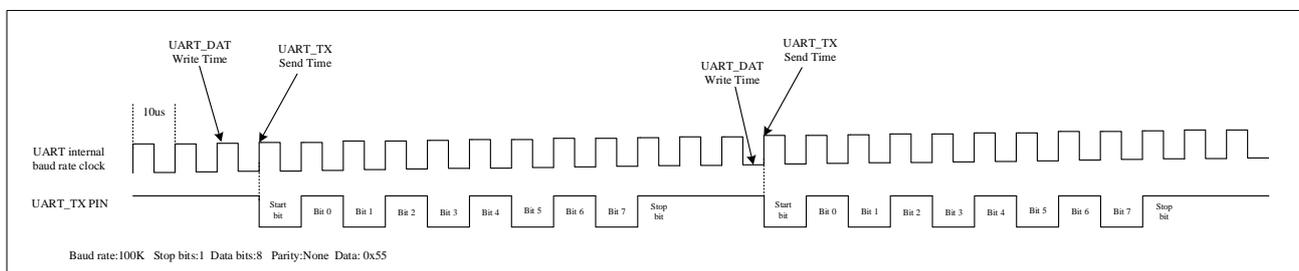
如果在断开帧开始发送前软件清零 `UART_CTRL1.SDBRK`，当前断开帧不会发送。

### 9.4.2.5 发送流程

1. 置位 `UART_CTRL1.UEN` 来使能 UART；
2. 配置波特率、数据长度、校验位、停止位长度、以及根据需要配置相关 DMA；
3. 使能发送功能 (`UART_CTRL1.TXEN`)；
4. 通过 CPU 或 DMA 将要发送的数据依次写入数据寄存器 `UART_DAT`，当数据写入数据寄存器时将清零 `UART_STS.TXDE`；
5. 当所有数据已写入到数据寄存器 `UART_DAT` 后，等待发送完成标志位 `UART_STS.TXC` 置 1，数据发送完成。

*注意：向 `UART_DAT` 写入数据，到数据到 `UART_TX` 引脚会存在 0~1 波特率周期的延时。例如下图 100K 波特率，在一个波特率周期任意时刻写入数据到 `UART_DAT`，会在下一个波特率周期开始时传输到 `UART_TX` 引脚。*

图 9-5 发送时差



### 9.4.2.6 单字节通信

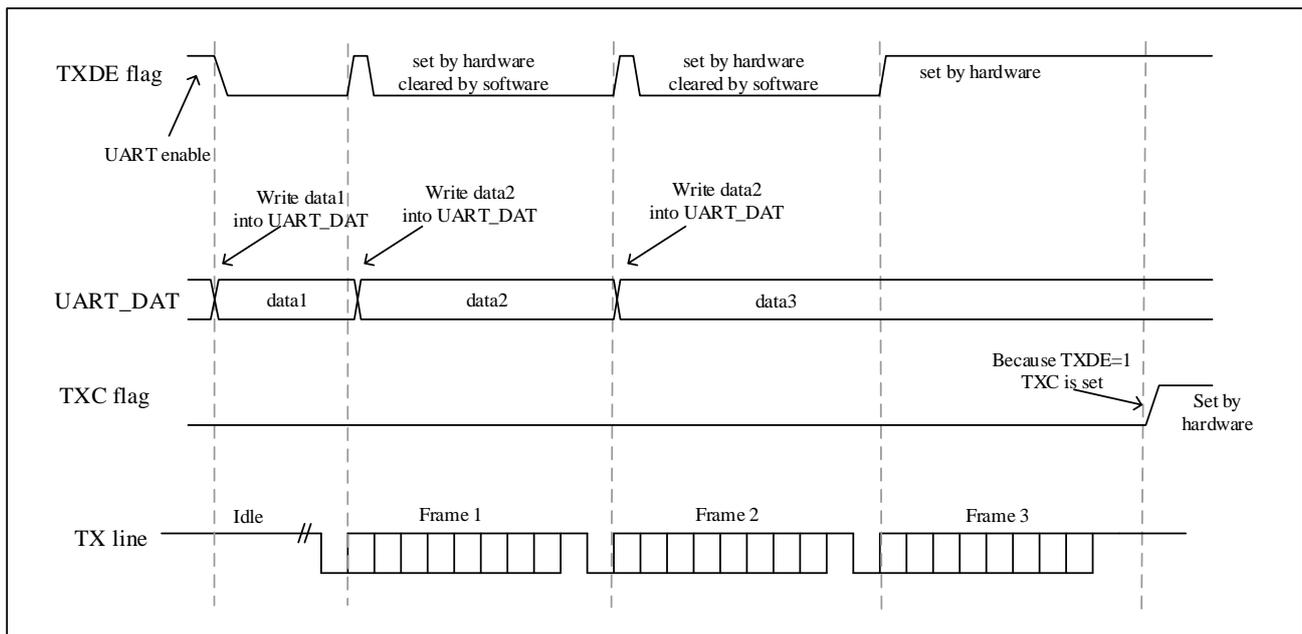
对数据寄存器 `UART_DAT` 的写操作将清零标志位 `UART_STS.TXDE`。

当数据已从发送数据寄存器送到移位寄存器时，`UART_STS.TXDE` 位由硬件置 1，表示数据开始发送。如果 `UART_CTRL1.TXDEIEN` 已置 1，将产生一个中断。此时，可将下一个数据写入数据寄存器 `UART_DAT`。

对数据寄存器 `UART_DAT` 进行写操作时：

- 如果移位寄存器空闲，数据将直接送到移位寄存器，同时 `UART_STS.TXDE` 硬件置 1
  - 如果移位寄存器正在发送数据，数据保存在数据寄存器，待上一个数据发送完成后，再送到移位寄存器
- 当一帧数据发送完成后并且 `UART_STS.TXDE` 置 1，`UART_STS.TXC` 被硬件置 1。如果 `UART_CTRL1.TXCIEN` 已置 1，将产生一个中断。`UART_STS.TXC` 通过以下软件操作清零：先读一次 `UART_STS` 寄存器，再写一次 `UART_DAT` 寄存器。

图 9-6 发送时 TXC/TXDE 的变化情况



## 9.4.3 接收器

### 9.4.3.1 起始位检测

在 UART 中，如果识别到一个特殊的采样序列 1 1 1 0 X 0 X 0 X 0 0 0 0，就认为检测到一个起始位。

在第 3、5、7 位的采样，以及在第 8、9、10 位的采样都为 '0'（也即 6 个 '0'），则确认收到起始位，并将 UART\_STS.RXDNE 置 1，但不会置位 NEF 噪声标志。如果 UART\_CTRL1.RXDNEIEN 已置 1，则产生一个中断。

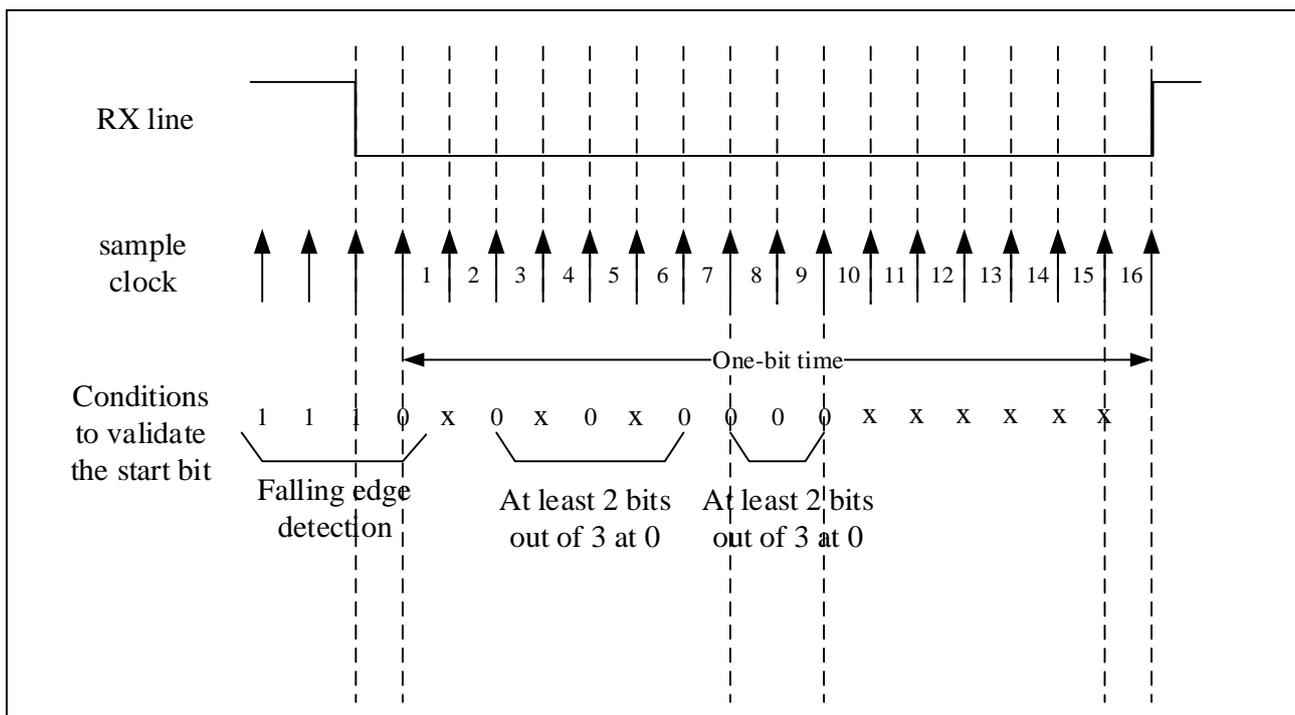
第 3、5、7 位的采样有两个 '0'，与此同时，第 8、9、10 位的采样有两个 '0' 点，也确认收到起始位，但是会置位 NEF 噪声标志位。

第 3、5、7 位的采样有三个 '0'，与此同时，第 8、9、10 位的采样有两个 '0' 点，也确认收到起始位，并置位 NEF 噪声标志位。

第 3、5、7 位的采样有两个 '0'，与此同时，第 8、9、10 位的采样有三个 '0' 点，也确认收到起始位，并置位 NEF 噪声标志位。

如果在第 3、5、7、8、9、10 位的采样值满足不了上面四种要求，UART 接收器认为没有接受到正确的起始位，将退出起始位侦测并回到空闲状态等待下降沿。

图 9-7 起始位检测



### 9.4.3.2 停止位

停止位长度可通过 `UART_CTRL2.STPB[1:0]` 配置。常规模式下，可配置为 1 位或 2 位。

- 0.5 位停止位：不对停止位进行采样。因此，此时不能检测帧错误和断开帧。
- 1 个停止位：默认情况下通过三个点对 1 个停止位的采样，选择第 8，第 9 和第 10 采样位上进行。
- 1.5 个停止位：对 1.5 个停止位的采样是在第 16，第 17 和第 18 采样点进行的，可分成两个部分：0.5 个数据位周期，接收器不做任何处理；然后是 1 个数据位周期，接收器对其进行采样。
- 2 个停止位：对 2 个停止位的采样是在第一停止位的第 8，第 9 和第 10 个采样点完成的。如果第一个停止位期间检测到一个帧错误，帧错误标志将被置起。在第一个停止位结束时 `UART_STS.RXDNE` 标志将被设置。第二个停止位将不会检测帧错误。

### 9.4.3.3 接收流程

- 将 `UART_CTRL1.UEN` 置 1 来使能 UART；
- 配置波特率、数据长度、校验位、停止位长度、以及根据需要配置相关 DMA；
- 使能接收器 (`UART_CTRL1.RXEN`)，开始起始位检测；
- 接收 8 位或 9 位数据，通过 `RX` 引脚送往接收移位寄存器，最低有效位在前；
- 当数据由接收移位寄存器送到 `RDR` 寄存器，`UART_STS.RXDNE` 被置 1，表示数据可以被读出。如果 `UART_CTRL2.RXNEIEN` 已置 1，将产生一个中断；
- 当接收过程中检测到帧错误、噪音或溢出错误，这样错误标志将被置 1。如果在数据传输过程中 `UART_CTRL1.RXEN` 被清零，当前接收数据丢失；
- `UART_STS.RXDNE` 通过对 `UART_DAT` 寄存器进行读操作清零；

- 在多缓冲器通信模式, UART\_STS.RXDNE 通过 DMA 对数据寄存器的读操作清零。
- 在单缓冲器通信模式, UART\_STS.RXDNE 通过软件对数据寄存器的读操作清零。

#### 9.4.3.4 空闲帧检测

当一空闲帧被检测到时, UART\_STS.IDLEIF 置 1。此时如果 UART\_CTRL1.IDLEIEN 已置 1, 将产生一个中断。UART\_STS.IDLEIF 可通过以下软件操作清零: 先读 UART\_STS 寄存器, 再读 UART\_DAT 寄存器。

#### 9.4.3.5 断开帧检测

当一断开帧被检测到时, 帧错误标志 UART\_STS.FEF 被硬件置 1, 可通过以下软件操作清零: 先读 UART\_STS 寄存器, 再读 UART\_DAT 寄存器。

#### 9.4.3.6 帧错误

如果在预期的时间内没有接收和识别到停止位, 产生一个帧错误, 标志位 UART\_STS.FEF 置 1, 同时无效数据将从移位寄存器送到 UART\_DAT 寄存器。在单字节通信时, 没有帧错误中断产生, 因为此时 UART\_STS.RXDNE 位同时置 1, 后者将产生中断。在多缓冲器通信(DMA)情况下, 如果 UART\_CTRL3.ERRIEN 已置 1, 将产生一个中断。

#### 9.4.3.7 溢出错误

如果 UART\_STS.RXDNE 已被置 1, 而接收移位寄存器又有数据需要送入数据寄存器, 则发生溢出错误, 同时标志位 UART\_STS.OREF 硬件置 1。此时数据寄存器中的数据不会丢失, 但移位寄存器中的数据将被覆盖。UART\_STS.OREF 可通过以下软件操作清零: 先读 UART\_STS 寄存器, 再读 UART\_DAT 寄存器。

当产生溢出错误时, 若 UART\_STS.RXDNEIEN 已置 1, 将产生一个接收中断。多缓冲器通信模式(DMA)下, 如果 UART\_CTRL3.ERRIEN 已置 1, 将产生一个错误中断。

#### 9.4.3.8 噪声错误

当接收器检测到噪声错误时, UART\_STS.NEF 被置 1, 可通过以下软件操作清零: 先读 UART\_STS 寄存器, 再读 UART\_DAT 寄存器。在单字节通信模式下不会产生噪声中断, 因为此时 UART\_STS.RXDNE 也被置 1 并产生接收中断。在多缓冲器通信模式(DMA), 如果 UART\_CTRL3.ERRIEN 已置 1, 将产生一个错误中断。

表 9-2 噪声检测的数据采样

采样值	NE 状态	接收的位	数据有效性
000	0	0	有效
001	1	0	无效
010	1	0	无效
011	1	1	无效
100	1	0	无效
101	1	1	无效
110	1	1	无效
111	0	1	有效

### 9.4.4 分数波特率计算

波特率通过 UART\_BRCF 寄存器配置, 分频系数由整数部分和小数部分组成, 同时适用于发送器与接收器。在写入 UART\_BRCF 之后, 波特率计数器会被波特率寄存器的新值替换。因此, 不要在通信过程中改变波

特率寄存器的数值。

$$\text{TX/RX 波特率} = f_{\text{PCLK}} / (16 * \text{UARTDIV})$$

其中  $f_{\text{PCLK}}$  为 UART 外设时钟：

■ PCLK1 用于 UART1/UART2/UART3，最高 64MHz。

UARTDIV 为无符号分频系数

#### 9.4.4.1 分频系数 UARTDIV 与 UART\_BRCF 寄存器配置

过采样设置为 16：

示例 1：

如果  $\text{UARTDIV} = 27.75$ ，则：

$$\text{DIV\_Decimal} = 16 * 0.75 = 12 = 0x0C$$

$$\text{DIV\_Integer} = 27 = 0x1B$$

因此  $\text{UART\_BRCF} = 0x1BC$

示例 2：

如果  $\text{UARTDIV} = 20.98$ ，则：

$$\text{DIV\_Decimal} = 16 * 0.98 = 15.68$$

取最接近的整数  $\text{DIV\_Decimal} = 16 = 0x10$ ，超出可配置范围，因此需要向整数位进位

从而  $\text{DIV\_Integer} = 20 + 1 = 21 = 0x15$

$$\text{DIV\_Decimal} = 0x0$$

因此  $\text{UART\_BRCF} = 0x150$

示例 3：

如果  $\text{UART\_BRCF} = 0x19B$ ：

$$\text{DIV\_Integer} = 0x19 = 25$$

$$\text{DIV\_Decimal} = 0x0B = 11$$

$$\text{UARTDIV} = 25 + 11/16 = 25.6875$$

过采样设置为 8：

示例 1：

如果  $\text{UARTDIV} = 27.75$ ，则：

$$\text{DIV\_Decimal} = 8 * 0.75 = 6 = 0x06$$

$$\text{DIV\_Integer} = 27 = 0x1B$$

因此  $\text{UART\_BRCF} = 0x1B6$

示例 2：

如果  $\text{UARTDIV} = 20.98$ ，则：

$$\text{DIV\_Decimal} = 8 * 0.98 = 7.84$$

取最接近的整数  $\text{DIV\_Decimal} = 8 = 0x08$ ，超出可配置范围，因此需要向整数位进位

$$\text{从而 } \text{DIV\_Integer} = 20 + 1 = 21 = 0x15$$

$$\text{DIV\_Decimal} = 0x0$$

$$\text{因此 } \text{UART\_BRCF} = 0x150$$

示例 3:

$$\text{如果 } \text{UART\_BRCF} = 0x196:$$

$$\text{DIV\_Integer} = 0x19 = 25$$

$$\text{DIV\_Decimal} = 0x06 = 6$$

$$\text{UARTDIV} = 25 + 6/8 = 25.75$$

表 9-3 设置波特率时的误差计算

16 倍过采样时									
波特率		f <sub>PCLK</sub> =32MHz				f <sub>PCLK</sub> =64MHz			
序号	Kbps	实际	过采样值	寄存器设置值	误差%	实际	过采样值	寄存器设置值	误差%
1	2.4	2.400	16	833.3125	0%	2.400	16	1666.6875	0%
2	9.6	9.601	16	208.3215	0.01%	9.600	16	416.6875	0%
3	19.2	19.196	16	104.1875	0.02%	19.202	16	208.3125	0.01%
4	57.6	57.554	16	34.750	0.08%	57.606	16	69.4375	0.01%
5	115.2	115.108	16	17.375	0.08%	115.108	16	34.75	0.08%
6	230.4	230.216	16	8.6875	0.08%	230.216	16	17.375	0.08%
7	460.8	463.768	16	4.3125	0.64%	460.432	16	8.6875	0.08%
8	921.6	914.286	16	2.1875	0.79%	927.536	16	4.3125	0.64%
9	2250	不可能	16	不可能	不可能	2285.714	16	1.75	1.59%
10	3000	不可能	16	不可能	不可能	3047.619	16	1.3125	1.59%
11	4000	不可能	16	不可能	不可能	4000	16	1	0%

8 倍过采样时

波特率		f <sub>PCLK</sub> =32MHz				f <sub>PCLK</sub> =64MHz			
序号	Kbps	实际	过采样值	寄存器设置值	误差%	实际	过采样值	寄存器设置值	误差%
1	2.4	2.400	8	1666.6875	0%	2.400	8	3333.3125	0%
2	9.6	9.600	8	416.6875	0%	9.600	8	833.3125	0%
3	19.2	19.202	8	208.3125	0.01%	19.199	8	416.6875	0%
4	57.6	57.606	8	69.4375	0.01%	57.606	8	138.875	0.01%
5	115.2	115.108	8	34.75	0.08%	115.212	8	69.4375	0.01%
6	230.4	230.216	8	17.375	0.08%	230.216	8	34.75	0.08%
7	460.8	460.432	8	8.6875	0.08%	460.432	8	17.375	0.08%
8	921.6	927.536	8	4.3125	0.64%	920.863	8	8.6875	0.08%
9	2250	2285.714	8	1.75	1.59%	2245.614	8	3.5625	0.19%
10	3000	3047.619	8	1.3125	1.59%	2976.744	8	2.6875	0.78%
11	4000	4000	8	1	0%	4000	8	2	0%

注意: CPU 的时钟频率越低, 则某一特定波特率的误差也越低。

## 9.4.5 UART 接收器容忍时钟的变化

应用中可能会出现发送误差(包括发射端时钟的变化)、接收端波特率误差及振荡器变化、传输线变化(通常由数据上升沿和下降沿时序不一致引起)。这些因素都会影响整个时钟系统的变化。只有当上述四个变化之和小于 UART 接收机的容差时, UART 异步接收机才能正常工作。

正常接收数据时, UART 接收器的容忍度为最大能容忍的变化, 取决于数据位长度的选择, 以及是否使用分数波特率分频系数。

表 9-4 当 DIV\_Decimal =0 时, UART 接收器的容忍度

WL 位	认为 NEF 是错误	不认为 NEF 是错误
0	3.75%	4.375%
1	3.41%	3.97%

表 9-5 当 DIV\_Decimal !=0 时，UART 接收器的容忍度

WL 位	认为 NEF 是错误	不认为 NEF 是错误
0	3.33%	4.0%
1	3.03%	3.63%

## 9.4.6 校验控制

通过设置 UART\_CTRL1.PCEN 来使能奇偶校验功能。

使能后，在发送数据时自动生成并发送校验位，接收数据时对校验位进行检查。

表 9-6 帧格式

WL 位	PCEN 位	UART 帧
0	0	起始位   8 位数据   停止位
0	1	起始位   7 位数据   奇偶校验位   停止位
1	0	起始位   9 位数据   停止位
1	1	起始位   8 位数据   奇偶校验位   停止位

### 偶校验

UART\_CTRL1.PSEL 设置为 0，使能偶校验

偶校验表示一帧数据（包括校验位）中‘1’的个数为偶数。例如：数据=11000101，有 4 个‘1’，则发送端偶校验位为‘0’（总共 4 个‘1’）。接收端对数据中‘1’个数进行确认：如果是偶数，校验通过；如果是奇数，表示产生了校验错误，UART\_STS.PEF 标志位置 1，此时如果 UART\_CTRL1.PEIEEN 已置 1，产生一个中断。

### 奇校验

UART\_CTRL1.PSEL 设置为 1，使能奇校验

奇校验表示一帧数据（包括校验位）中‘1’的个数为奇数。例如：数据=11000101，有 4 个‘1’，则发送端奇校验位为‘1’（总共 5 个‘1’）。接收端对数据中‘1’个数进行确认：如果是奇数，校验通过；如果是偶数，表示产生了校验错误，UART\_STS.PEF 标志位置 1，此时如果 UART\_CTRL1.PEIEEN 已置 1，产生一个中断。

## 9.4.7 DMA 通信

UART 支持 DMA 通信，此时采用多缓冲模式可达到较高的通信效率。

### 9.4.7.1 DMA 发送

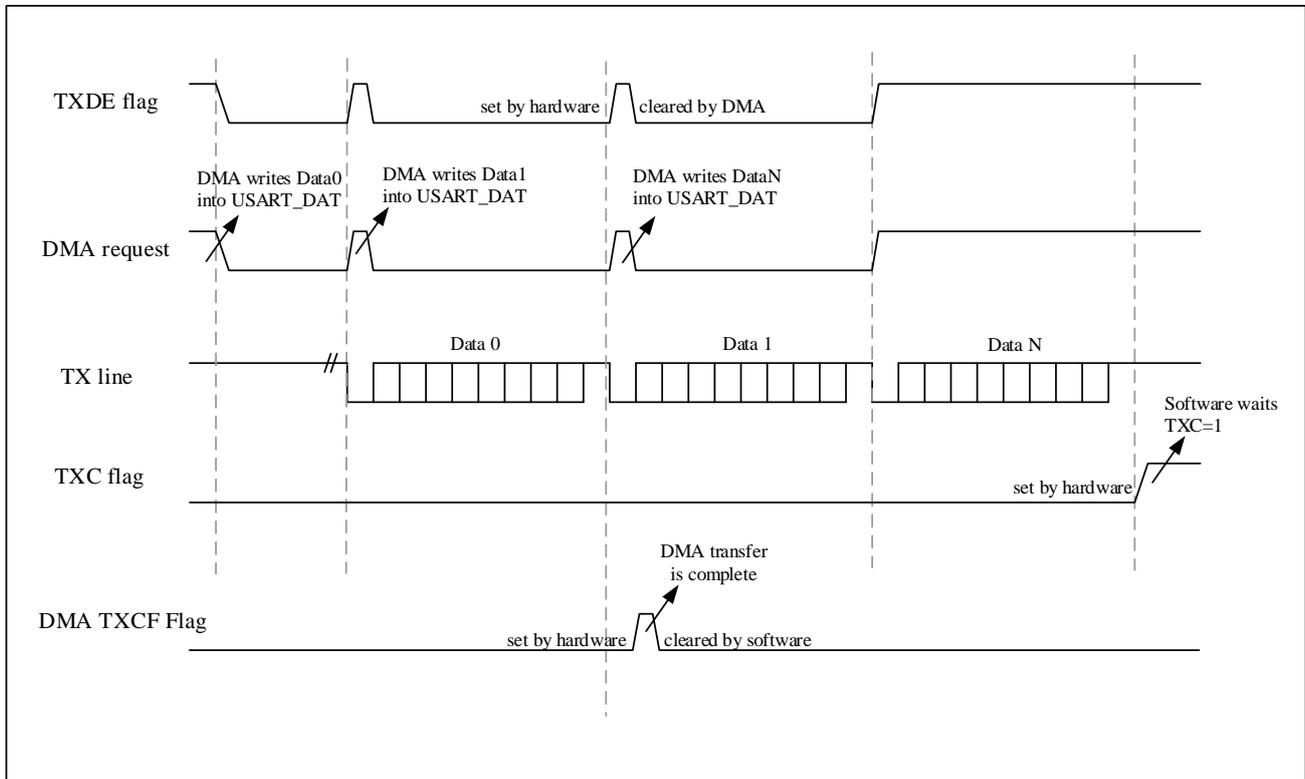
发送器通过将 UART\_CTRL3.DMATXEN 置 1 来使能 DMA 发送。当发送移位寄存器为空时 (UART\_STS.TXDE=1)，DMA 将数据由 SRAM 送到数据寄存器 UART\_DAT。

使用 DMA 发送功能时，按照以下流程对 DMA 进行配置：

1. 设置 DMA 传输的源地址，DMA 传输时从此地址读取要发送的数据
2. 设置 DMA 传输的目的地地址为 UART\_DAT 寄存器地址

3. 设置要传输的总的字节数.
4. 设置 DMA 通道优先级、循环模式、地址增加模式、传输数据宽度、中断（传输完成一半还是全部完成时）
5. 激活当前 DMA 通道
6. 传输完成后，标志位 DMA\_INTSTS.TXCFx 被置 1

图 9-8 DMA 发送



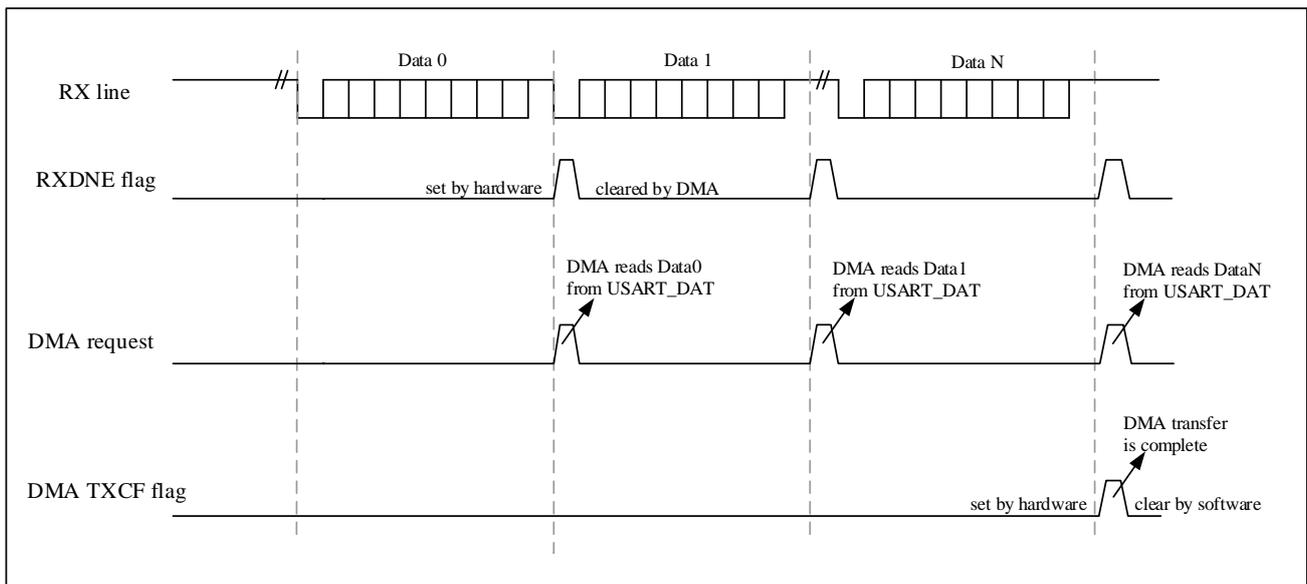
### 9.4.7.2 DMA 接收

接收器通过将 UART\_CTRL3.DMATXEN 置 1 来使能 DMA 接收。当收到 1 字节数据时 (UART\_STS.RXDNE=1)，DMA 将数据从数据寄存器 USART\_DAT 读出数据，送到 SRAM。

使用 DMA 接收功能时，按照以下流程对 DMA 进行配置：

1. 设置 DMA 传输的源地址为 USART\_DAT 寄存器地址，DMA 传输时从此地址读取要发送的数据
2. 设置 DMA 传输的目的地地址，DMA 传输时将数据送到此地址。
3. 设置要传输的总的字节数.
4. 设置 DMA 通道优先级、循环模式、地址增加模式、传输数据宽度、中断（传输完成一半还是全部完成时）
5. 激活当前 DMA 通道

图 9-9 DMA 接收



在多缓冲器通信模式，当检测到帧错误、溢出错误、噪声错误时，相应标志位置 1。如果此时 UART\_CTRL3.ERRIEN 已置 1，产生一个错误中断。

## 9.4.8 多处理器通信

UART 支持多处理器通信：多个设备同时连接到 UART 进行通信，因此必须判定哪一个设备作为主设备,其他设备自动做为从设备。主机的 TX 引脚直接连接到其他从设备的 RX 引脚，所有从设备的 TX 引脚通过逻辑与的方式合并，再连接到主设备的 RX 引脚。

在多处理器通信模式下，从设备处于静默模式，主设备在需要时通过通过指定方式唤醒某一个从设备，从而从设备可以和主设备进行正常通信。

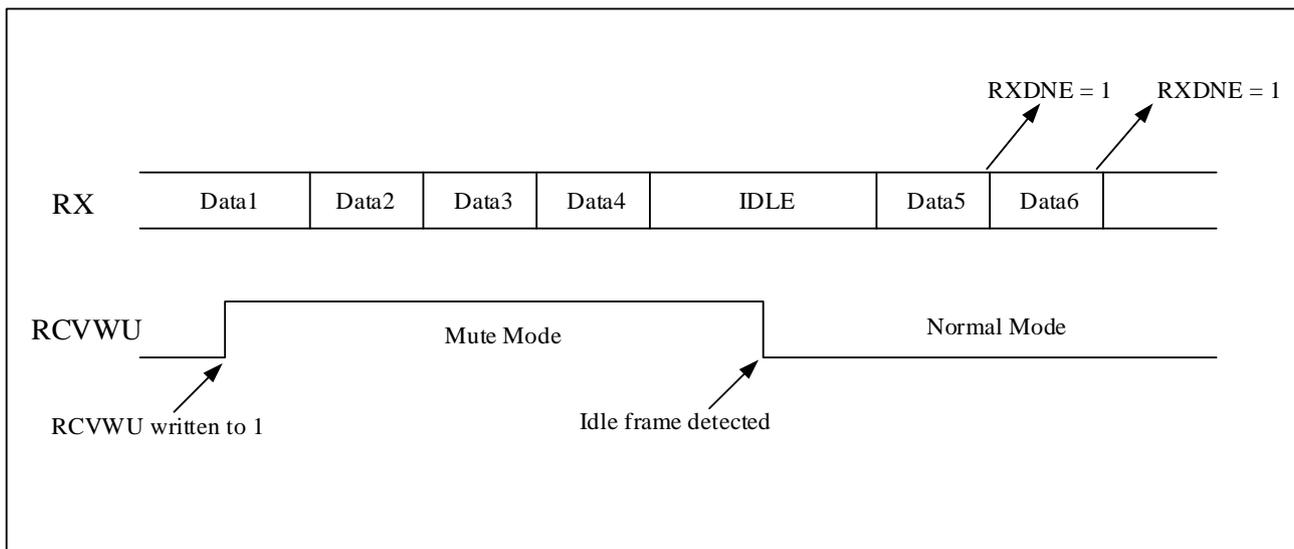
UART 可通过空闲总线检测或地址标识检测的方式从静默模式唤醒。

### 9.4.8.1 空闲总线检测

空闲总线检测流程如下：

1. 清零 UART\_CTRL1.WUM 位，UART 启用空闲总线检测功能。
2. 当 UART\_CTRL1.RCVWU 已置 1(可通过硬件自动控制或由在特定条件下由软件配置)，UART 进入静默模式，此时接收状态标志位不会置位，同时接收中断被禁用。
3. 如图 9-10 所示，当检测到空闲帧时，UART 被唤醒,同时 UART\_CTRL1.RCVWU 被硬件清零,此时 UART\_STS.IDLEF 标志位不会被置 1。

图 9-10 静默模式下的空闲总线检测



### 9.4.8.2 地址标识检测

当 UART\_CTRL1.WUM 置 1 时，UART 启用地址标识检测功能。标识地址通过 UART\_CTRL2.ADDR[3:0] 来配置。如果接收的数据最高有效位（MSB）为 1，当前数据为地址，低 4 位有效；如果 MSB = 0，则当前数据为普通数据。

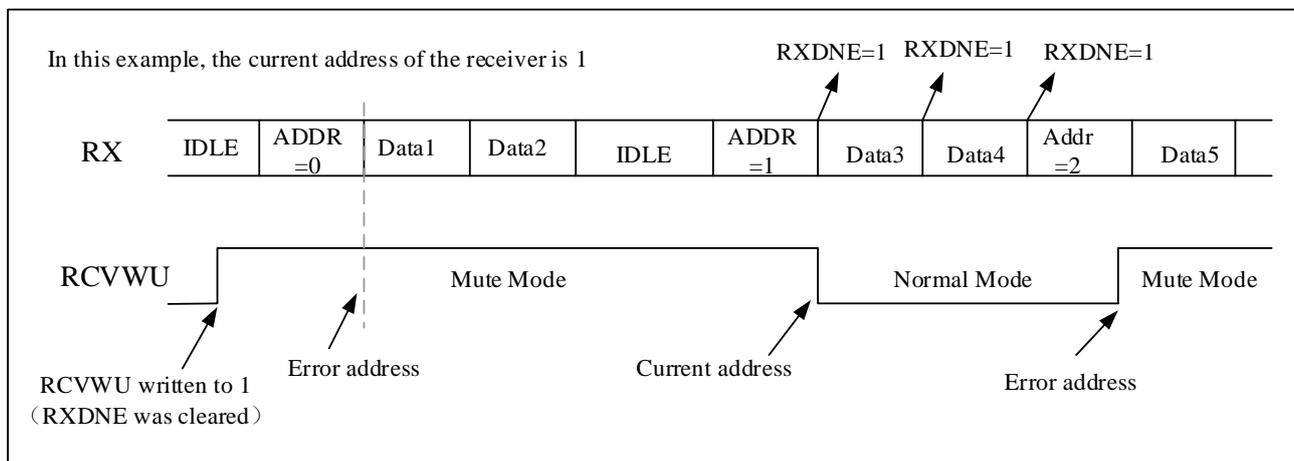
此模式下，UART 可通过以下方式进入静默模式：

1. 当接收器没有数据处理时，可通过软件将 UART\_CTRL1.RCVWU 置 1，使 UART 进入静默模式。  
*注意：当接收数据寄存器为空时，(UART\_STS.RXNE=0)，UART\_CTRL1.RCVWU 位可通过软件写 0 或写 1。否则，对 UART\_CTRL1.RCVWU 的写操作被忽略。*
2. 当接收器收到的地址与预设的地址标识不匹配时，UART\_CTRL1.RCVWU 由硬件置 1。

静默模式下，所有接收状态标志位不会置位，同时所有接收中断被禁用。

当接收器收到的地址与预设的地址标识相同时，UART 从静默模式唤醒，UART\_CTRL1.RCVWU 被硬件清零，同时 UART\_STS.RXDNE 位置 1，此时可进行正常的传输。

图 9-11 静默模式下的地址标识检测



## 9.4.9 单线半双工模式

UART 支持单线半双工通信模式，允许数据双方向收发，但同一时间只能单向接收数据或发送数据，数据通信的冲突由软件控制。

通过设置 UART\_CTRL3.HDMEN 位来选择单线半双工模式，此时以下控制位必须全部清零：UART\_CTRL2.LINMEN、UART\_CTRL3.IRDAMEN。

启用单线半双工通信模式后，TX 引脚与 RX 引脚在芯片内部相连，外部 RX 引脚不再使用。当没有数据发送时，TX 引脚被释放。因此，TX 引脚未被 UART 使用时，必须配置为浮空输入或开漏输出高电平。

## 9.4.10 串行 IrDA 红外编解码模式

UART 支持 IrDA (Infrared Data Association) SIR ENDEC 规范。

通过设置 UART\_CTRL3.IRDAMEN 位来选择是否使用 IrDA 模式。当启用 IrDA 模式时，以下配置位必须全部清零：UART\_CTRL2.STPB[1:0]、UART\_CTRL2.LINMEN、UART\_CTRL3.HDMEN。

通过设置 UART\_CTRL3.IRDALP 位，可选择 IrDA 的正常工作模式或低功耗模式。

### 9.4.10.1 IrDA 正常模式

当 UART\_CTRL3.IRDALP=0，IrDA 工作在正常模式。

IrDA 是一个半双工通信接口，因此在发送和接收之间最小要有 10ms 的延时。数据采用反相归零(RZI)调制，即采用红外光源脉冲表示逻辑 0。脉冲宽度规定为一个位周期的 3/16，如图 9-13 所示。最大波特率为 115200bps。

UART 将数据送到 SIR 编码器进行调制后输出。调制后的数据流输出给外部红外发送器进行发送。接收时，先通过外部红外接收器接收数据并解调后，发送到 SIR 解码器，解码后再将数据送给 UART。

发送编码器与解码器输入极性相反。空闲时，编码器输出为低电平，而解码器输入为高电平。编码器输出高脉冲表示逻辑 0，输出低电平作为逻辑 1。解码器输入则与之相反。

当 UART 正在发送数据给 IrDA 编码器时，解码器将忽略数据线上的所有数据。当 UART 正在从解码器接收数据时，发送到编码器的数据也被忽略，不进行编码操作。

脉冲宽度可软件配置。IrDA 规范要求脉冲宽度大于 1.41us。如果脉冲宽度小于 2 个周期，数据被过滤而丢失。PSCV 是在 UART\_GTP 寄存器配置的预分频值。

### 9.4.10.2 IrDA 低功耗模式

当 UART\_CTRL3.IRDALP=1，IrDA 工作在低功耗模式。

在低功耗模式下发送数据时，脉冲宽度为 3 倍 PSCV 周期。经 PSCV 分频后的时钟频率最小值为 1.42MHz，典型值为 1.8432MHz，(1.42 MHz < 时钟频率 < 2.12 MHz)。

接收数据时，有效低电平信号宽度必须大于 2 个 PSCV 周期。

图 9-12 IrDA SIR ENDEC-框图

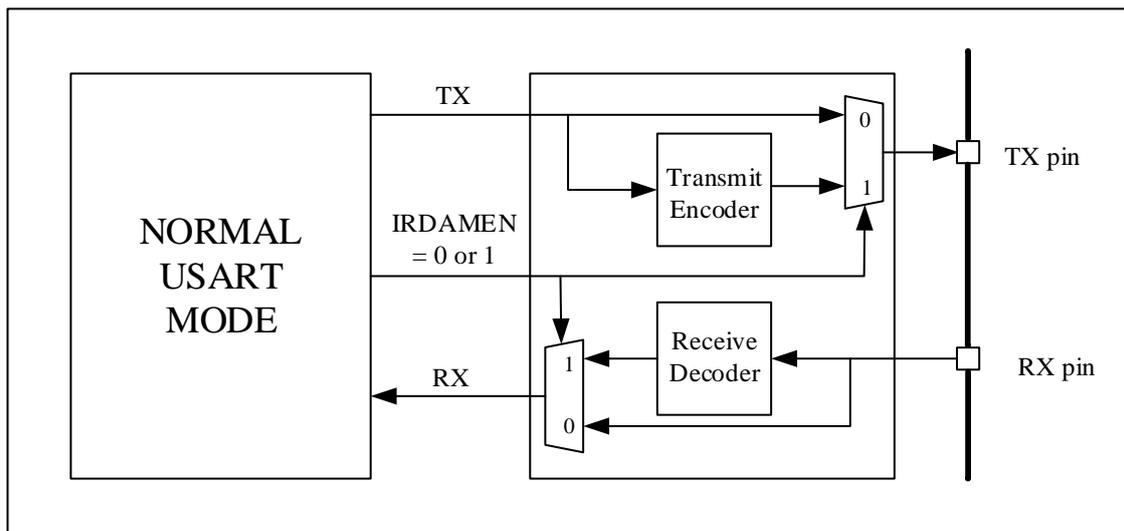
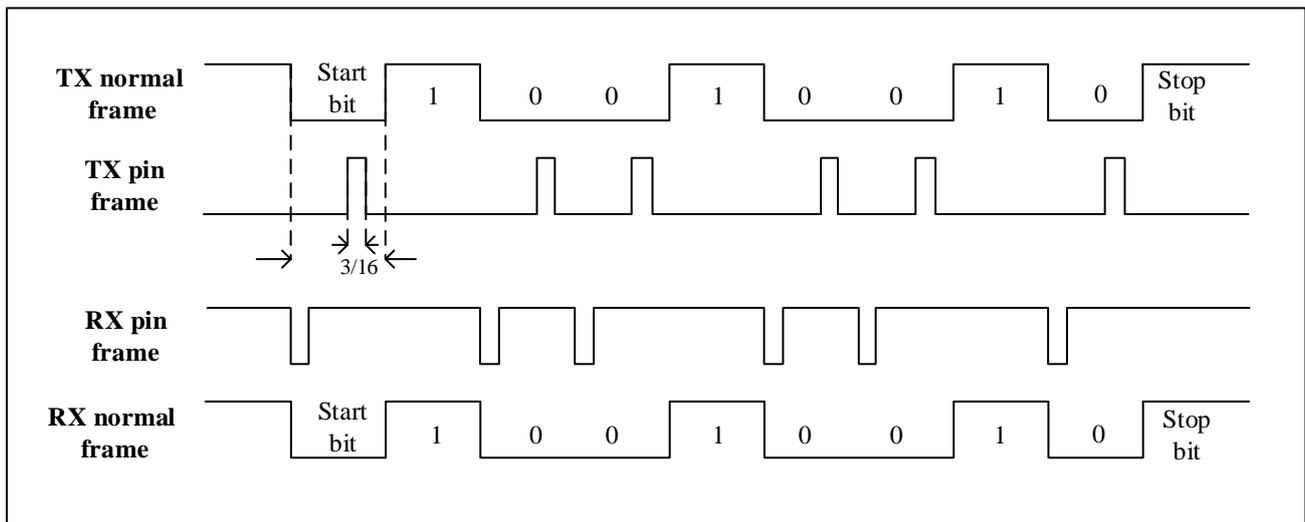


图 9-13 IrDA 数据调制(3/16)-正常模式



### 9.4.11 LIN 模式

UART 支持 LIN (Local interconnection Network)模式，支持作为主机时发送同步断开帧，也支持作为从机检测断开帧。通过设置 `UART_CTRL2.LINMEN` 位来使能 LIN 模式。

注意：当使用 LIN 模式时，以下配置位必须全部被清零：`UART_CTRL2.STPB[1:0]`、`UART_CTRL3.HDMEN`、`UART_CTRL3.IRDAMEN`。

#### 9.4.11.1 LIN 发送

在 LIN 模式下发送数据时，数据长度只能配置为 8 位。将 `UART_CTRL1.SDBRK` 置 1 将发送一个 13 位“0”断开帧，并插入一个停止位。

#### 9.4.11.2 LIN 接收

当总线空闲或数据传输过程中均可检测断开帧。断开帧检测机制独立于 UART 接收器。

通过配置 UART\_CTRL2.LINBDL 位，断开帧检测有效低电平位可选择 10 位或 11 位。

当接收器检测到一个起始位，采样电路在每个位的第 8、9、10 个过采样时钟点进行过采样。如果 10 个或 11 个位都是 '0'，并且又跟着一个定界符，表示检测到一个断开帧，UART\_STS.LINBDF 位置 1。在确认为断开帧前，必须检测定界符，意味着 RX 线已经回归空闲状态（高电平）。此时如果 UART\_CTRL2.LINBDIEN 已置 1，将产生一个中断。

如果在 10 个或 11 个位前收到了 '1'，当前断开帧检测被取消，并重新寻找起始位。

图 9-14 LIN 模式下的断开检测（11 位断开帧长度-设置了 LINBDL 位）

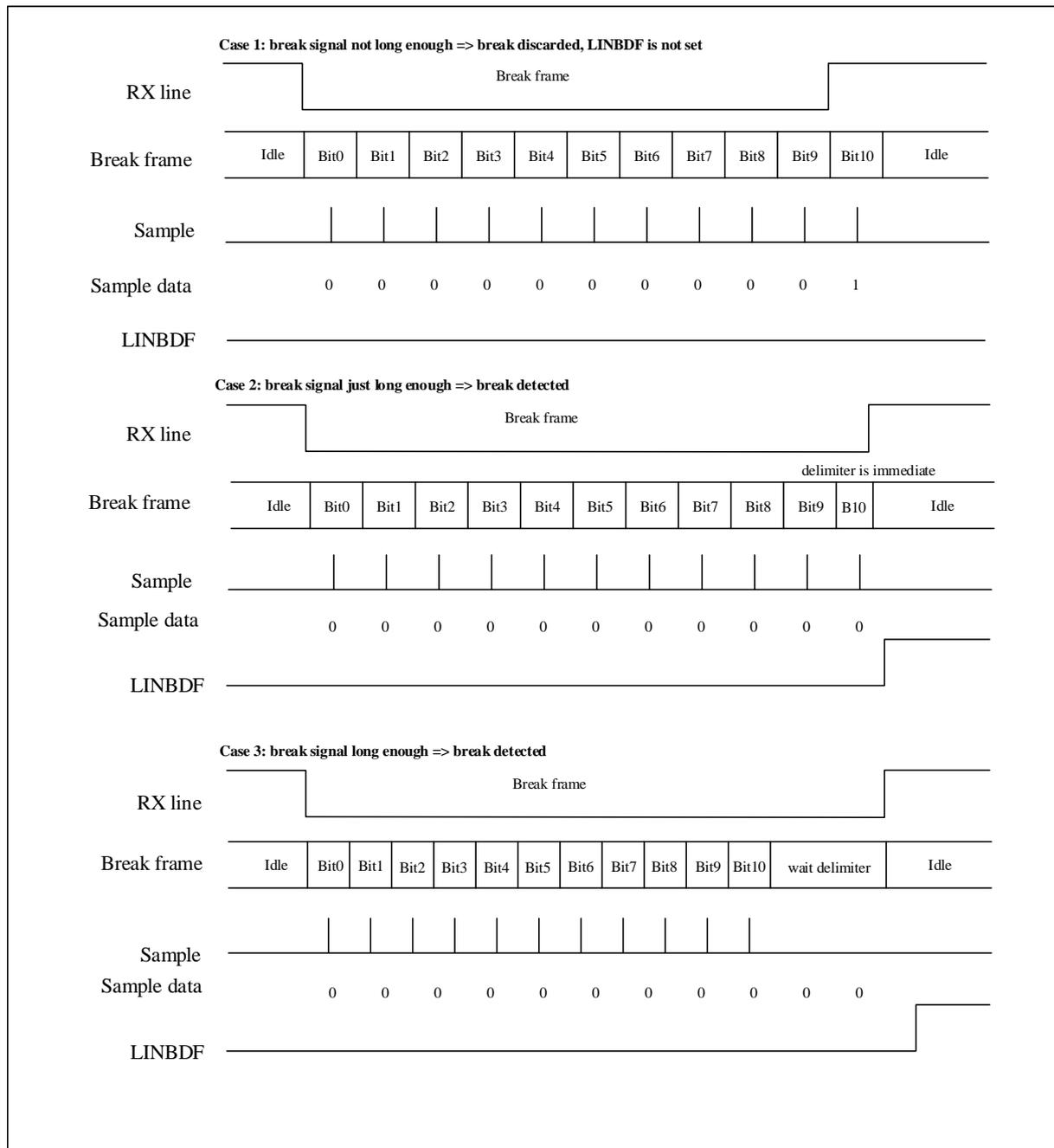
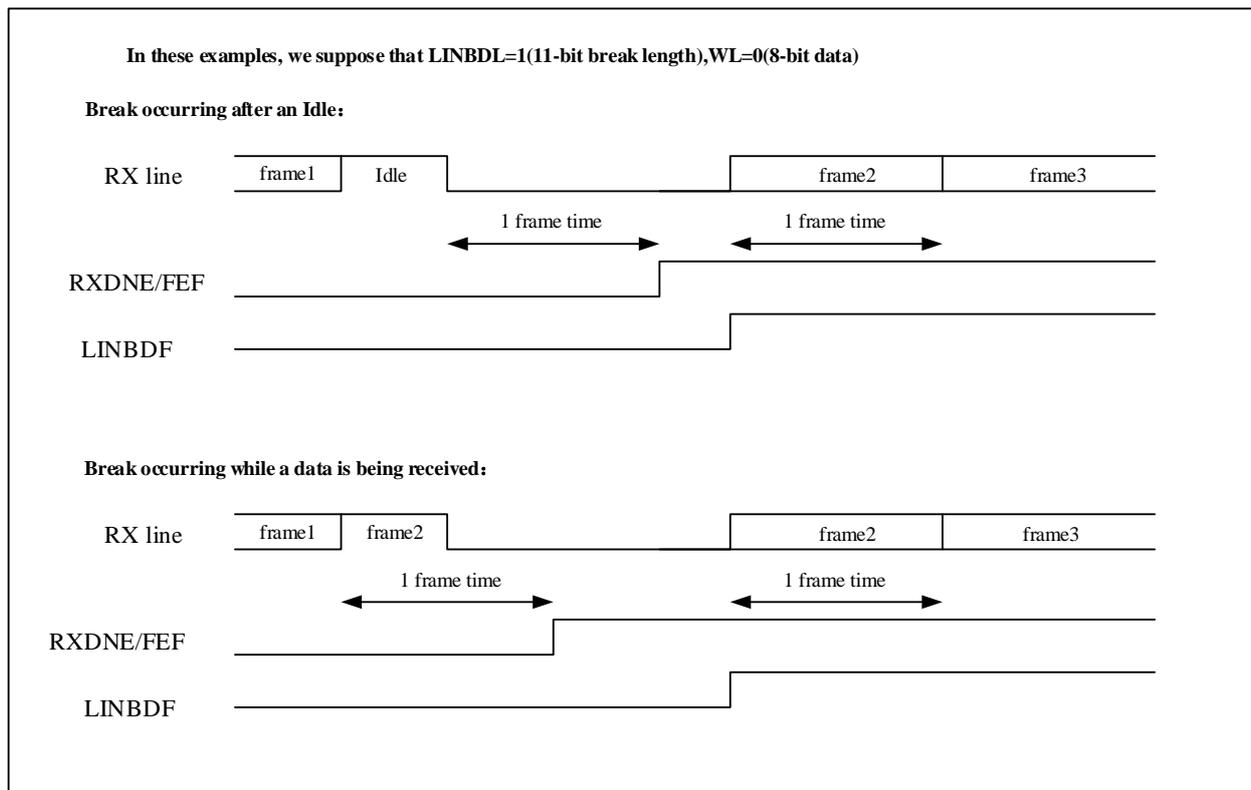


图 9-15 LIN 模式下的断开检测与帧错误的检测



### 9.4.12 UART3 低功耗唤醒

UART3 支持低功耗唤醒功能，当芯片处于低功耗模式时，UART 可以在低功耗模式下继续工作。

配置流程如下：

1. 使能 EXTI10 事件触发触发
2. 切换系统时钟为 LSI
3. 切换 UART3 时钟源为 LSI
4. 配置 UART3
5. 进入 STOP 模式前使能 UART3 唤醒模式
6. 配置选择进入 STOP 模式类型为 WFE
7. 退出 STOP 模式后对时钟和 UART3 进行重新配置

*注意：系统时钟切换到 LSI 后，其他外设需要注意时钟主频的变化。*

## 9.5 中断请求

UART 的各种中断事件是逻辑或的关系。如果某个事件对应的中断使能位已置 1，将产生一个相应的中断。但同一个时间只产生一个中断请求。

表 9-7 UART 中断请求

中断函数	中断事件	事件标志	使能
UART 全局中断	发送数据寄存器空	TXDE	TXDEIEN
	接收器超时	RTOF	RTOIE
	发送完成	TXC	TXCIEN
	接收数据就绪可读	RXDNE	RXDNEIEN
	检测到数据溢出	ORERR	
	检测到空闲线路	IDLEF	IDLEIEN
	奇偶检验错	PEF	PEIEN
	断开标志	LINBDF	LINBDIEN
	噪声标志，多缓冲通信（DMA）中的溢出错误和帧错误 <sup>(1)</sup>	NEF/OREF/FEF	ERRIEN <sup>(1)</sup>

(1) 仅当使用 DMA 接收数据(UART\_CTRL3.DMARXEN=1)时，才使用这个标志位。

## 9.6 模式配置

表 9-8 UART 模式设置<sup>(1)</sup>

通信模式	UART1	UART2	UART3
异步模式	Y	Y	Y
DMA 通讯模式	Y	Y	Y
多处理器	Y	Y	Y
单线半双工模式	Y	Y	Y
IrDA 红外模式	Y	Y	Y
LIN	Y	Y	Y

(1) Y = 支持该模式，N = 不支持该模式

## 9.7 UART 寄存器

### 9.7.1 UART 寄存器总览

表 9-9 UART 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	UART_CTRL1	Reserved				SWAP	OSPM	DEAT				DEDT				DEP	DEM	SDBRK	PEIEN	TXCIEN	TXDEIEN	RXDNEIEN	IDLEIEN	WUM	RCVWU	WL	PCEN	PSEL	TXEN	RXEN	UEN		
0x04	UART_CTRL2	Reserved													RTOITE	RTOCF	RTOEN	LINBDL	LINBDIE	LINMEN	Reserved					STPB	Reserved	ADDR					
0x08	UART_CTRL3	Reserved																							IRDALP	IRDAME	ERRIEN	DMARXE	DMATXE	HDMEN	Reserved		
0x0C	UART_STATUS	Reserved													RTOF	FEF	NEF	OREF	PEF	LINBDF	Reserved	RXDNE	TXC	TXDE	IDLEF	Reserved							
0x10	UART_DATA	Reserved																			DATV												
0x14	UART_BRCF	Reserved											DIV_Integer						DIV_Decimal														
0x18	UART_GTP	Reserved																			PSCV												
0x24	UART_RT0	Reserved	TIME																														
0x28	UART_WKUP	Reserved																										DATCLR	EN				

### 9.7.2 UART 控制寄存器 1(UART\_CTRL1)

偏移地址: 0x00

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved				SWAP	OSPM	DEAT				DEDT						
				rw	rw	rw				rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
DEP	DEM	SDBRK	PEIEN	TXCIEN	TXDEIEN	RXDNEIEN	IDLEIEN	WUM	RCVWU	WL	PCEN	PSEL	TXEN	RXEN	UEN	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

位域	名称	描述
31:28	Reserved	保留, 必需保持复位值。
27	SWAP	交换 TX/RX 引脚 (Swap TX/RX pins) 此位由软件置 1 和清零。

位域	名称	描述
		0: 按标准引脚排列定义使用 TX/RX 引脚 1: 交换 TX 和 RX 引脚功能。允许在与另一个 UART 的交叉连接时工作 只有在禁止 UART (UEN="0") 时才能写入此位域。
26	OSPM	过采样模式 (Oversampling mode) 0: 16 倍过采样 1: 8 倍过采样 只有在禁止 UART (UE="0") 时才能写入该位。 注: 在 LIN、IrDA 模式下, 此位必须保持清零。
25:21	DEAT	驱动器使能时间 (Driver Enable assertion time) 该 5 位值用于定义激活 DE (驱动器使能) 信号与起始位开始间的时间。此时间以采样时间单位表示 (1/8 或 1/16 位时间, 具体取决于过采样速率)。 只有在禁止 UART (UE="0") 时才能写入此位域。
20:16	DEDT	驱动器使能禁止时间 (Driver Enable deassertion time) 该 5 位值用于定义发送的消息中最后一个停止位结束与取消激活 DE (驱动器使能) 信号间的时间。此时间以采样时间单位表示 (1/8 或 1/16 位时间, 具体取决于过采样速率)。 如果在 DEDT 时间内对 UART_TDR 寄存器执行写操作, 则新数据仅在经过 DEDT 和 DEAT 时间后才会发送。 只有在禁止 UART (UE="0") 时才能写入此位域。
15	DEP	驱动器使能极性选择 (Driver enable polarity selection) 0: DE 信号高电平有效。 1: DE 信号低电平有效。 只有在禁止 UART (UE="0") 时才能写入该位。
14	DEM	驱动器使能模式 (Driver enable mode) 此位用于通过 DE 信号激活外部收发器控制。 0: 禁止 DE 功能。 1: 使能 DE 功能。 只有在禁止 UART (UEN="0") 时才能写入该位。
13	SDBRK	发送断开帧 (Send break)。 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 该位。 0: 没有发送断开帧。 1: 发送断开帧。
12	PEIEN	校验错误中断使能 (PE interrupt enable)。 如果该位置 1, UART_STS.PEF 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。
11	TXCIEN	发送完成中断使能 (Transmission complete interrupt enable)。 如果该位置 1, UART_STS.TXC 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。
10	TXDEIEN	发送缓冲区空中断使能 (TXDE interrupt enable)。 如果该位置 1, UART_STS.TXDE 被置位时产生中断。

位域	名称	描述
		0: 发送缓冲区空中断禁止。 1: 发送缓冲区空中断使能。
9	RXDNEIEN	读数据缓冲区非空中断和过载错误中断使能 (RXDNE interrupt enable)。 如果该位置 1, UART_STS.RXDNE 或 UART_STS.OREF 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。
8	IDLEIEN	IDLE 线检测中断使能 (IDLE interrupt enable)。 如果该位置 1, UART_STS.IDLEF 被置位时产生中断。 0: IDLE 线检测中断禁用。 1: IDLE 线检测中断使能。
7	WUM	从静默模式唤醒方法 (Wake up mode)。 0: 空闲帧唤醒。 1: 地址标识唤醒。
6	RCVWU	接收器从静默模式中唤醒 (Receiver wakeup) 软件可以通过将该位置 1 使得 UART 进入静默模式, 将该位清 0 唤醒 UART。 空闲帧唤醒模式下 (UART_CTRL1.WUM=0), 当检测到空闲帧时, 该位由硬件清 0。地址标识唤醒模式下 (UART_CTRL1.WUM=1), 当接收到一个地址匹配帧时, 该位由硬件清 0; 或接收到一个地址非匹配帧时, 由硬件置 1。 0: 接收器处于普通工作模式。 1: 接收器处于静默模式。
5	WL	字长 (Word length)。 0: 8 数据位。 1: 9 数据位。 <i>注意: 在数据传输过程中 (发送或者接收时), 不能修改这个位。</i>
4	PCEN	校验控制使能 (Parity control enable)。 0: 校验控制禁用。 1: 校验控制被使能。
3	PSEL	校验模式 (Parity selection)。 0: 偶校验。 1: 奇校验。
2	TXEN	发送器使能 (Transmitter enable)。 0: 发送器禁用。 1: 发送器使能。
1	RXEN	接收器使能 (Receiver enable)。 0: 接收器禁用。 1: 接收器使能。
0	UEN	UART 使能 (UART enable)。 当该位被清零, 在当前字节传输完成后 UART 的分频器和输出停止工作, 以减少功耗。该位由软件设置和清零。 0: UART 禁用。 1: UART 使能。

### 9.7.3 UART 控制寄存器 2(UART\_CTRL2)

偏移地址：0x04

复位值：0x0000 0000

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

位域	名称	描述
31:18	Reserved	保留，必需保持复位值。
17	RTOITE	接收器超时中断使能 (Receiver timeout interrupt enable) 此位由软件置 1 和清零。 0: 禁止中断 1: UART_STS 寄存器中的 RTOF 位置 1 时生成 UART 中断。 注：如果 UART 不支持接收器超时功能，该位保留并由硬件强制清零。
16	RTOCF	接收器超时清零标志 (Receiver timeout clear flag) 向此位写入“1”时，UART_STS 寄存器中的 RTOF 标志将清零。 注：如果 UART 不支持接收器超时功能，该位保留并由硬件强制清零。
15	RTOEN	接收器超时使能 (Receiver timeout enable) 此位由软件置 1 和清零。 0: 禁止接收器超时功能。 1: 使能接收器超时功能。 使能此功能后，如果 RX 线路在 RTOR（接收器超时寄存器）中编程的持续时间内处于空闲状态（无接收），则 UART_STS 寄存器中的 RTOF 标志置 1。 注：如果 UART 不支持接收器超时功能，该位保留并由硬件强制清零。
14	LINBDL	LIN 断开帧检测长度 (LIN break detection length)。 该位用来设定在断开帧长度。 0: 10 位 1: 11 位 注意: LINBDL 可用于 LIN 模式及其他模式下的断开帧的检测长度控制，且检测长度和 LIN 模式相同。
13	LINBDIEN	LIN 断开帧检测中断使能 (LIN break detection interrupt enable)。 如果该位置 1，当 UART_STS.LINBDF 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
12	LINMEN	LIN 模式使能 (LIN mode enable)

位域	名称	描述
		0: LIN 模式禁用 1: LIN 模式使能
11:7	Reserved	保留, 必需保持复位值。
6:5	STPB[1:0]	停止位长 (STOP bits)。 00: 1 停止位。 01: 0.5 停止位。 10: 2 停止位。 11: 1.5 停止位。 <i>注: UART 推荐使用 1/2 停止位, 0.5/1.5 停止位一般用于智能卡模式</i>
4	Reserved	保留, 必需保持复位值。
3:0	ADDR[3:0]	UART 地址。 在多处理器通信下的静默模式中使用的, 使用地址标识来唤醒某个 UART 设备。 地址标识唤醒模式下 (UART_CTRL1.WUM=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。

## 9.7.4 UART 控制寄存器 3(UART\_CTRL3)

偏移地址: 0x08

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						IRDALP	IRDAMEN	ERRIEN	DMA RXEN	DMA TXEN	HDMEN	Reserved			
						rw	rw	rw	rw	rw	rw				

位域	名称	描述
31:9	Reserved	保留, 必需保持复位值。
8	IRDALP	IrDA 低功耗模式 (IrDA low-power)。 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。
7	IRDAMEN	IrDA 模式使能 (IrDA mode enable)。 0: IrDA 禁用。 1: IrDA 使能。
6	ERRIEN	错误中断使能 (Error interrupt enable)。 当 DMA 接收模式 (UART_CTRL3.DMARXEN=1) 使能时, 如果该位被置 1, UART_STS.FEF、UART_STS.OREF、UART_STS.NEF 被置位将产生中断。

位域	名称	描述
		0: 错误中断禁用。 1: 错误中断使能。
5	DMARXEN	DMA 接收使能 (DMA receiver enable)。 0: DMA 接收模式禁用。 1: DMA 接收模式使能。
4	DMATXEN	DMA 发送使能 (DMA transmitter enable)。 0: DMA 发送模式禁用。 1: DMA 发送模式使能。
3	HDMEN	半双工模式使能 (Half-duplex mode enable)。 该位用于使能半双工模式。 0: 半双工模式禁用。 1: 半双工模式使能。
2:0	Reserved	保留, 必需保持复位值。

## 9.7.5 UART 状态寄存器 (UART\_STS)

偏移地址: 0x0C

复位值: 0x0000 0180

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															RTOF
															r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FEF	NEF	OREF	PEF	LINBDF	Reserved	RXDNE	TXC	TXDE	IDLEF	Reserved					
r	r	r	r	rc_w0		rc_w0	rc_w0	r	r						

位域	名称	描述
31:17	Reserved	保留, 必需保持复位值。
16	RTOF	接收超时 (recevier timeout) 已经超过在 RTO 寄存器中编程的超时值后, 若无任何通信, 此位由硬件置 1。此位由软件清零, 方法是向 UART_CTRL2 寄存器中的 RTOCF 位写入“1”。如果 UART_CTRL2 寄存器中的 RTOITE=“1”, 则会生成中断。 0: 未达到超值 1: 已达到超值, 未接收到任何数据
15	FEF	帧错误 (Framing error)。 当检测到同步错位、过多的噪声或者检测到断开符 (即没有检测到预期的停止位), 该位被硬件置位。由软件序列将其清零 (先读 UART_STS, 再读 UART_DAT)。 0: 未检测到帧错误。 1: 检测到帧错误或者断开帧 (break frame)。 <i>注意: 该位不会产生中断, 因为它和 UART_STS.RXDNE 一起出现, 硬件会在</i>

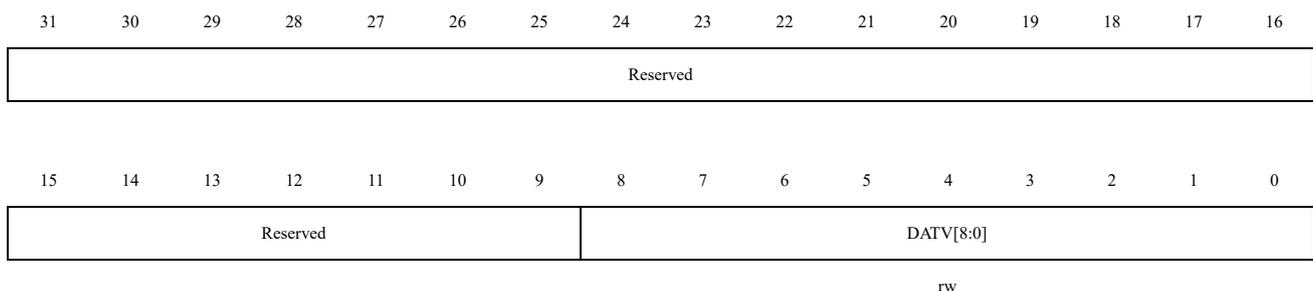
位域	名称	描述
		<p>设置 <code>UART_STS.RXDNE</code> 标志时产生中断。如果当前传输的数据既产生了帧错误, 又产生了过载错误, 硬件还是会继续该数据的传输, 并且只设置 <code>OREF</code> 标志位。</p> <p>在多缓冲区通信模式 (DMA) 下, 如果设置了 <code>UART_CTRL3.ERRIEN</code> 位, 则设置 <code>FEF</code> 标志时会产生中断。</p>
14	NEF	<p>噪声错误标志 (Noise error flag)。</p> <p>在接收到的帧检测到噪音时, 由硬件对该位置位。由软件序列对其清零 (先读 <code>UART_STS</code>, 再读 <code>UART_DAT</code>)。</p> <p>0: 没检测到噪声错误。 1: 检测到噪声错误。</p> <p>注意: 该位不会产生中断, 因为它和 <code>UART_STS.RXDNE</code> 一起出现, 硬件会在设置 <code>UART_STS.RXDNE</code> 标志时产生中断。在多缓冲区通信模式(DMA)下, 如果设置了 <code>UART_CTRL3.ERRIEN</code> 位, 则设置 <code>NEF</code> 标志时会产生中断。</p>
13	OREF	<p>溢出错误 (Overrun error)。</p> <p>在 <code>RXDNE</code> 置位的情况下, 如果 <code>UART_DAT</code> 寄存器接收到来自移位寄存器的数据, 该位置 1。当寄存器 <code>UART_CTRL3.RXDNEIEN</code> 位被置位, 将会有中断产生。</p> <p>软件先读 <code>UART_STS</code>, 再读 <code>UART_DAT</code> 可清除该位。</p> <p>0: 没有检测到溢出错误。 1: 检测到溢出错误。</p> <p>注意: 在多缓冲区通信模式(DMA)下, 如果设置了 <code>UART_CTRL3.ERRIEN</code> 位, 则设置 <code>OREF</code> 标志时会产生中断。</p> <p>注意: 当 <code>OREF</code> 置位后, <code>UART_DAT</code> 不会再更新数据; 如果此时 <code>RXDNE</code> 为 0, 因为数据不再更新, 故 <code>RXDNE</code> 不会重新置 1。</p>
12	PEF	<p>校验错误 (Parity error)。</p> <p>当接收到的数据帧校验位与预期校验值不同时, 该位置位。</p> <p>软件先读 <code>UART_STS</code>, 再读 <code>UART_DAT</code> 可清除该位。</p> <p>0: 没检测到校验错误。 1: 检测到校验错误。</p>
11	LINBDF	<p>LIN 断开检测标志 (LIN break detection flag)。</p> <p>如果设置了 <code>UART_CTRL2.LINMEN</code> 位, 当检测到 LIN 断开, 该位由硬件置位。如果 <code>UART_CTRL2.LINBDIEN</code> 被置位时, 将产生中断。</p> <p>该位由软件清 0。</p> <p>0: 没有检测到 LIN 断开字符。 1: 检测到 LIN 断开字符。</p>
10	Reserved	保留, 必需保持复位值。
9	RXDNE	<p>读数据缓冲区非空 (Read data register not empty)。</p> <p>当读数据缓冲区接收到来自移位寄存器的数据时, 该位置 1。当寄存器 <code>UART_CTRL1.RXDNEIEN</code> 位被置位, 将会有中断产生。</p> <p>软件可以通过对该位写 0 或读 <code>UART_DAT</code> 寄存器来将该位清 0。</p> <p>0: 读数据缓冲区为空。 1: 读数据缓冲区不为空。</p>
8	TXC	发送完成 (Transmission complete)。

位域	名称	描述
		<p>上电复位后，该位被置 1。如果 UART_STS.TXDE 置位，在当前数据发送完成时该位置 1。</p> <p>UART_CTRL1.TXCIEN 被置位将产生中断。</p> <p>该位由软件清 0。</p> <p>0：发送没有完成。</p> <p>1：发送完成。</p>
7	TXDE	<p>发送数据缓冲区空 (Transmit data register empty)。</p> <p>上电复位或待发送数据已发送至移位寄存器后，该位置 1。</p> <p>UART_CTRL1.TXDEIEN 被置位将产生中断。</p> <p>该位在软件将待发送数据写入 UART_DAT 时被清 0。</p> <p>0：发送数据缓冲区不为空。</p> <p>1：发送数据缓冲区空。</p>
6	IDLEF	<p>空闲线检测标志 (IDLE line detected)。</p> <p>在一个帧时间内，在 RX 引脚检测到空闲状态，该位置 1。当寄存器 UART_CTRL1.IDLEIEN 位被置位，将会有中断产生。</p> <p>软件先读 UART_STS，再读 UART_DAT 可清除该位。</p> <p>0：未检测到空闲帧。</p> <p>1：检测到空闲帧。</p> <p><i>注意：IDLEF 位不会再次被置高直到 RXDNE 位被置起 (即又检测到一次空闲总线)。</i></p>
5:0	Reserved	保留，必需保持复位值。

## 9.7.6 UART 数据寄存器(UART\_DAT)

偏移地址：0x10

复位值：未定义 (不确定值)



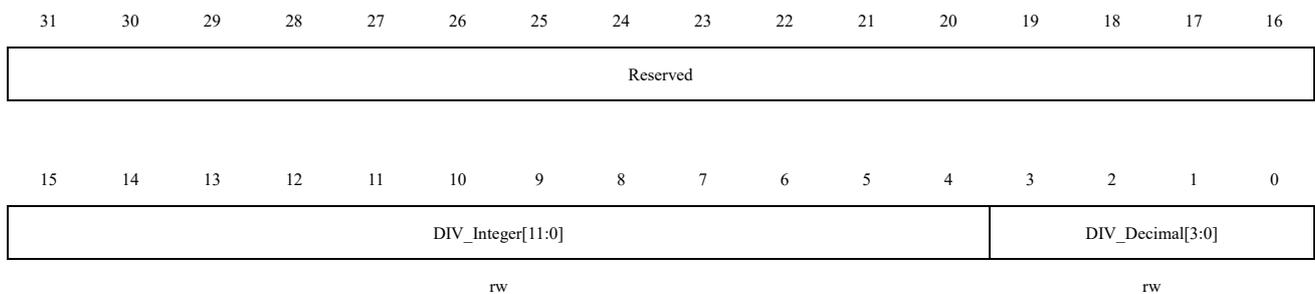
位域	名称	描述
31:9	Reserved	保留，必需保持复位值。
8:0	DATV[8:0]	<p>数据值 (Data value)</p> <p>包含了发送或接收的数据；软件可以通过写这些位来改变发送数据，或读这些位的值来获取接收数据。</p> <p>如果使能了奇偶校验，当发送数据被写入寄存器，数据的最高位 (第 7 位或第 8 位取决于 UART_CTRL1.WL 位) 将被校验位取代。</p>

## 9.7.7 UART 波特率配置寄存器 (UART\_BRCF)

偏移地址： 0x14

复位值： 0x0000 0000

注意： UART\_CTRL1.UEN=1 时，不能写该寄存器；如果 UART\_CTRL1.TXNE 或 UART\_CTRL1.RXNE 被分别禁止，波特计数器停止计数。

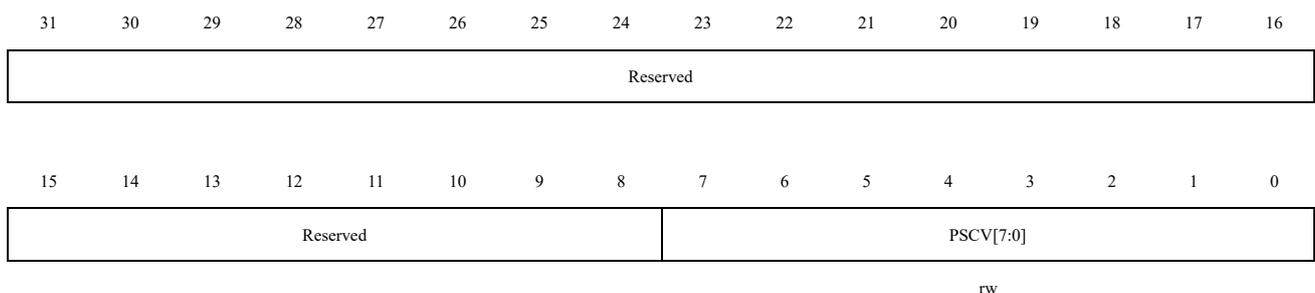


位域	名称	描述
31:16	Reserved	保留，必需保持复位值。
15:4	DIV_Integer [11:0]	波特率分频器的整数部分。
3:0	DIV_Decimal[3:0]	波特率分频器的小数部分。

## 9.7.8 UART 保护时间和预分频寄存器(UART\_GTP)

偏移地址： 0x18

复位值： 0x0000 0000



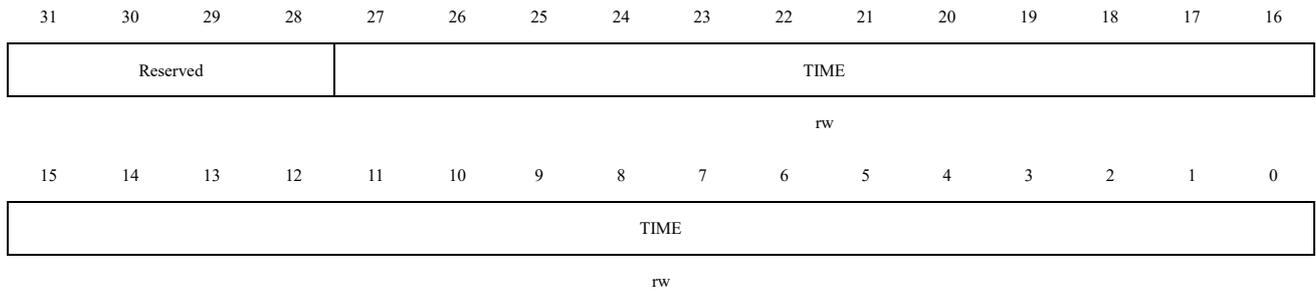
位域	名称	描述
31:8	Reserved	保留，必需保持复位值。
7:0	PSCV[7:0]	预分频器值 (Prescaler value)。 在 IrDA 低功耗模式下，这些位用来设定将外设时钟 (PCLK1/PCLK2) 分频产生低功耗频率的分频系数。 00000000：保留 – 不要写入该值 00000001：对源时钟 1 分频 ...

位域	名称	描述
		11111111: 对源时钟 255 分频 在 IrDA 正常模式下, PSCV 只能设置成 00000001。

### 9.7.9 UART 接收超时寄存器(UART\_RTO)

偏移地址: 0x24

复位值: 0x0000 0000



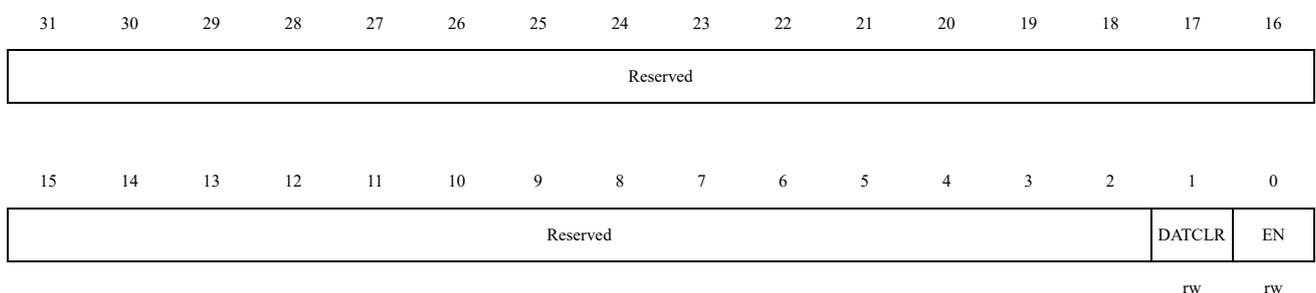
位域	名称	描述
31:28	Reserved	保留, 必需保持复位值。
27:0	TIME	接收器超时值 <i>注: 以波特率时间为单位时间</i>

### 9.7.10 UART 低功耗唤醒寄存器(UART\_WKUP)

偏移地址: 0x28

复位值: 0x0000 0000

*注意: 仅 UART3 支持低功耗唤醒功能*



位域	名称	描述
31:2	Reserved	保留, 必需保持复位值。
1	DATCLR	是否清除接收唤醒数据 0: 清除接收唤醒数据, 且清除 RXDNE 标志位

位域	名称	描述
		1: 不清除接收唤醒数据, RXDNE 标志位置 1
0	EN	低功耗唤醒使能寄存器 0: 关闭 UART 低功耗唤醒功能 1: 开启 UART 低功耗唤醒功能

## 10 HDIV 除法器

### 10.1 功能描述

- 只支持 32 位操作，支持有符号或无符号运算
- 输入：32 位被除数、32 位除数
- 输出：32 位商、32 位余数
- 8 个时钟周期完成一次有/无符号整数除法运算
- 除数为零，警告标志位，数据固定返回 0
- 支持可选硬件自动使能（写入除数自动使能）
- 读商或余数寄存器，无需查询状态，可立即读（使能）

### 10.2 操作说明

#### 10.2.1 基本使用流程

##### 步骤 1：初始化配置

检查 HDIV\_CTRLSTS.HDIVDF，确保为 1(第一次使用除外，默认值为 0)，即模块处于空闲状态

设置 HDIV\_CTRLSTS.HDIVEN=1

根据场景需要配置 HDIV\_CTRLSTS.TYPESEL ， HDIV\_CTRLSTS.BUSHOLD ， HDIV\_CTRLSTS.AUTOSTART

##### 步骤 2：写入被除数和除数

被除数写入 HDIV\_DIVIDEND

除数写入 HDIV\_DIVISOR

##### 步骤 3：启动计算

若 HDIV\_CTRLSTS.AUTOSTART=1，则在写入 HDIV\_DIVISOR 后，自动开始计算

否则，在 HDIV\_CTRLSTS.HDIVEN 写入 1 后，开始计算

##### 步骤 4：等待计算完成

如果 HDIV\_CTRLSTS.BUSHOLD=1,则无需查询 HDIV\_CTRLSTS.HDIVDF

如果 HDIV\_CTRLSTS.BUSHOLD=0, 则需要查询 HDIV\_CTRLSTS.HDIVDF ， 直到 HDIV\_CTRLSTS.HDIVDF=1

##### 步骤 5：读取结果

得数需要读取 HDIV\_QUOTIENT

余数需要读取 HDIV\_REMAINDER

### 步骤 6: 循环操作

配置改变, 需要从步骤 1 开始

配置不变, 需要从步骤 2 开始

## 10.2.2 异常处理

除零错误, 即 HDIV\_DIVISOR 配置为 0, 此时发生除零异常, HDIV\_DIVBY0.HDIV\_DIVBY0 自动置 1, 用户需要在每次计算完成后, 查询该位, 确认本次计算是否发生除零异常。该位会在下一次计算开始时自动清零。

## 10.3 HDIV 寄存器

### 10.3.1 HDIV 寄存器总览

表 10-1 HDIV 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0x00	HDIV_CTLSTS	Reserved																						TYPESEL	AUTOSTA	BUSHOLD	Reserved	HDIVDF	Reserved	HDIVSTAR	HDIVEN						
0x04	HDIV_DIVIDEND	DIVIDEND																																			
0x08	HDIV_DIVISOR	DIVISOR																																			
0x0c	HDIV_QUOTIENT	QUOTIENT																																			
0x10	HDIV_REMAINDER	REMAINDER																																			
0x14	HDIV_DIVBY0	Reserved																																	DIVBY0		

### 10.3.2 HDIV 控制状态寄存器 (HDIV\_CTRLSTS)

偏移地址: 0x00

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved							TYPESEL	AUTO START	BUS HOLD	Reserved			HDIVDF	Reserved	HDIV START	HDIVEN

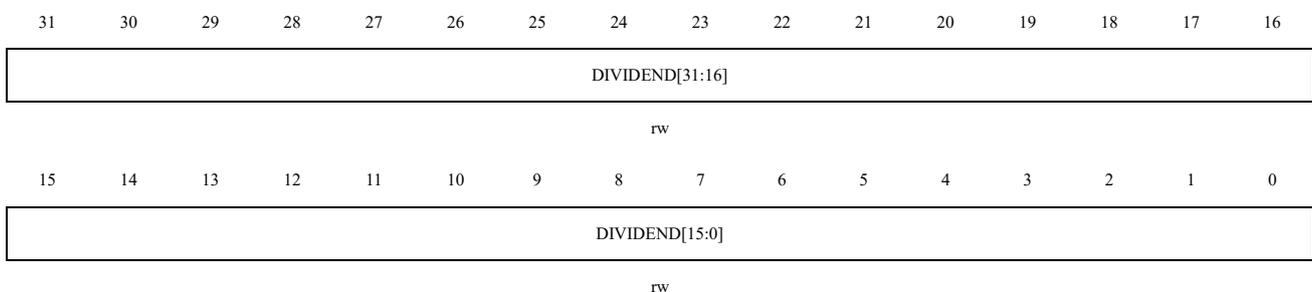
rw      rw      rw                      rw                      rw      rw

位域	名称	描述
[31:9]	Reserved	保留，必须保持复位值。
[8]	TYPESSEL	被除数、除数数据类型定义 0: 无符号数 1: 有符号数
[7]	AUTOSTART	自动计算使能，自动计算是指，每写入一次除数，即开始一次除法运算 0: 禁能 1: 使能
[6]	BUSHOLD	计算过程中总线保持模式 0: 当计算开始，未完成本次计算过程中，软件读取得数或余数，总线将不会保持 1: 当计算开始，未完成本次计算过程中，软件读取得数或余数，总线保持
[5:4]	Reserved	保留，必须保持复位值。
[3]	HDIVDF	本次计算完成标志 0: 未完成 1: 已完成
[2]	Reserved	保留，必须保持复位值。
[1]	HDIVSTART	单次计算使能，使能一次该控制位，进行一次除法运算 0: 禁能 1: 使能
[0]	HDIVEN	HDIV 使能，只有该控制位使能后，才可以进行单次或自动计算 0: 禁能 1: 使能

### 10.3.3 HDIV 被除数寄存器 (HDIV\_DIVIDEND)

偏移地址: 0x04

复位值: 0x00000000

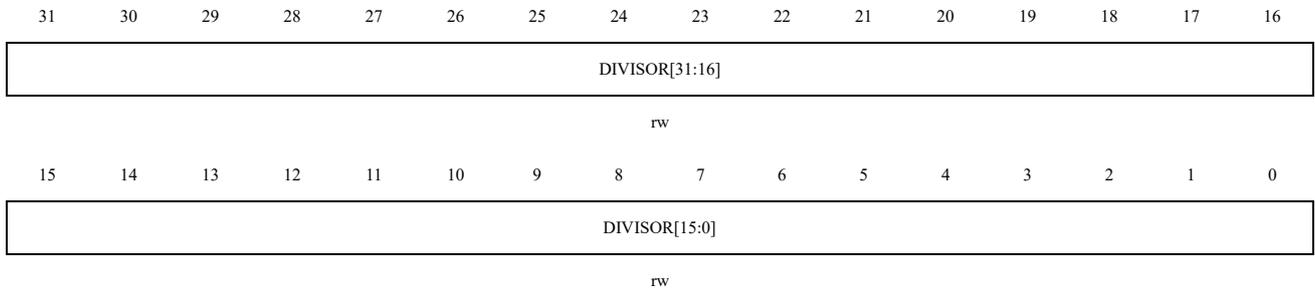


位域	名称	描述
[31:0]	DIVIDEND	32bit 被除数

### 10.3.4 HDIV 除数寄存器 (HDIV\_DIVISOR)

偏移地址: 0x08

复位值: 0x00000000

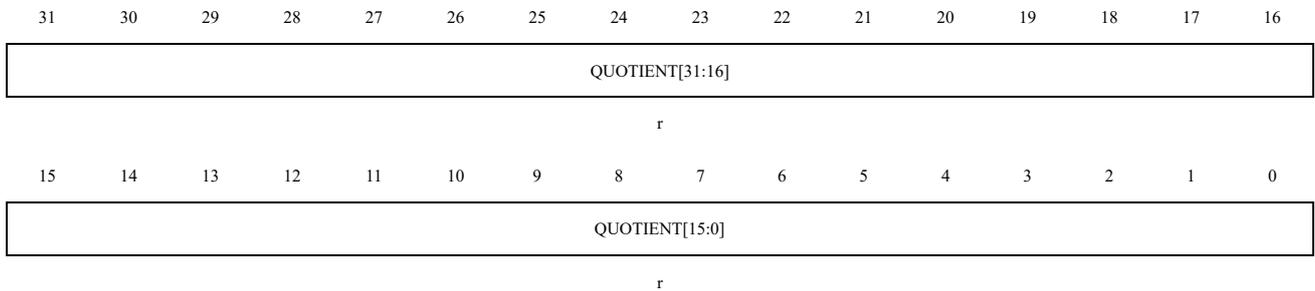


位域	名称	描述
[31:0]	DIVIDEND	32bit 除数

### 10.3.5 HDIV 商寄存器 (HDIV\_QUOTIENT)

偏移地址: 0x0C

复位值: 0x00000000

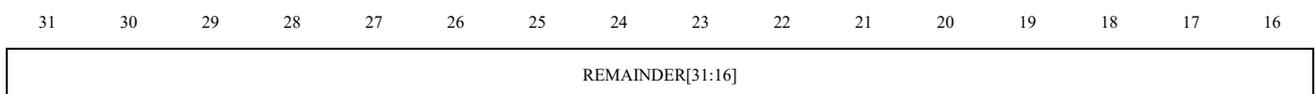


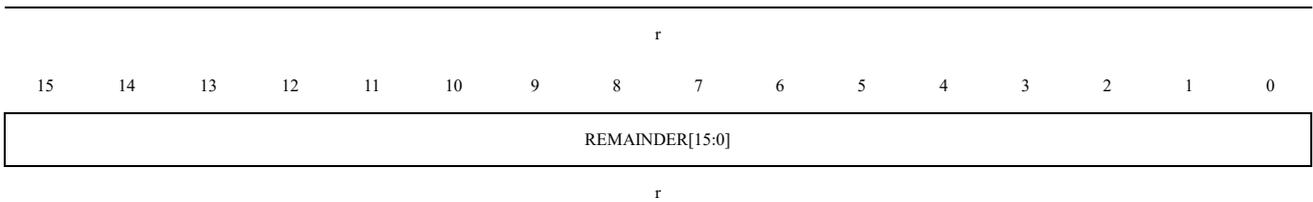
位域	名称	描述
[31:0]	QUOTIENT	32bit 商 如果除数写入 0，则得数为 0

### 10.3.6 HDIV 余数寄存器 (HDIV\_REMAINDER)

偏移地址: 0x10

复位值: 0x00000000



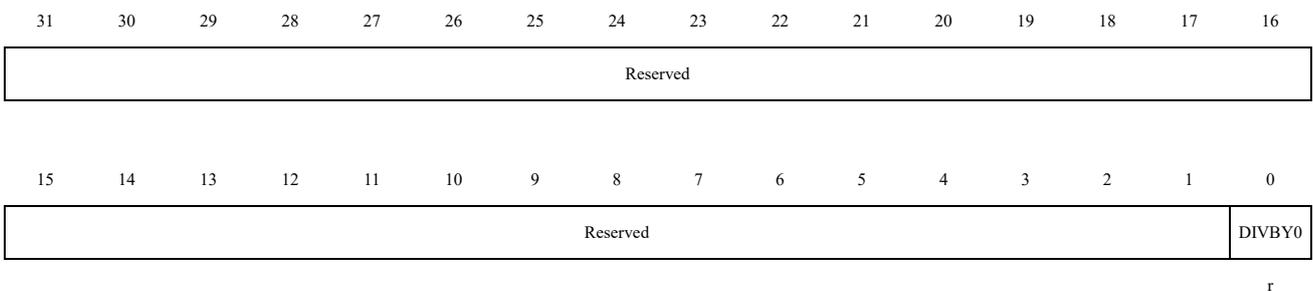


位域	名称	描述
[31:0]	REMAINDER	32bit 余数 如果除数写入 0，则得数为 0

### 10.3.7 HDIV 除零异常标志寄存器 (HDIV\_DIVBY0)

偏移地址: 0x14

复位值: 0x00000000



位域	名称	描述
[31:1]	Reserved	保留，必须保持复位值。
[0]	DIVBY0	除数为 0 标志位 0: 除数非 0 1: 除数为 0

## 11 SQRT 开平方根器

### 11.1 功能描述

- 只支持 32 位操作
- 输入：32 位无符号被开方整数
- 输出：16 位开方根
- 8 个时钟周期完成一次无符号整数开方运算
- 支持可选硬件自动使能（写入被开方整数自动使能）
- 读结果寄存器，无需查询状态，可立即读

### 11.2 操作说明

#### 11.2.1 基本使用流程

##### 步骤 1：初始化配置

检查 `SQRT_CTRLSTS.SQRTF`，确保为 1(第一次使用除外，默认值为 0)，即模块处于空闲状态

设置 `SQRT_CTRLSTS.SQRTEN=1`

根据场景需要配置 `HDIV_CTRLSTS.BUSHOLD`，`HDIV_CTRLSTS.AUTOSTART`

##### 步骤 2：写入被开方数

被开方数写入 `SQRT_RADICAND`

##### 步骤 3：启动计算

若 `SQRT_CTRLSTS.AUTOSTART=1`，则在 `SQRT_RADICAND` 写入后，自动开始计算

否则，在 `SQRT_CTRLSTS.SQRTEN` 写入 1 后，开始计算

##### 步骤 4：等待计算完成

如果 `SQRT_CTRLSTS.BUSHOLD=1`，则无需查询 `SQRT_CTRLSTS.SQRTF`

如果 `SQRT_CTRLSTS.BUSHOLD=0`，则需要查询 `SQRT_CTRLSTS.SQRTF`，直到 `SQRT_CTRLSTS.SQRTF=1`

##### 步骤 5：读取结果

平方根数需要读取 `SQRT_ROOT`

##### 步骤 6：循环操作

配置改变，需要从步骤 1 开始

配置不变，需要从步骤 2 开始

## 11.3 SQRT 寄存器

### 11.3.1 SQRT 寄存器总览

表 11-1 SQRT 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	SQRT_CTRLSTS TS	Reserved																								AUTOSTART	BUSHOLD	Reserved	SQRTDF	Reserved	SQRTSTART	SQRTEN	
0x04	SQRT_RADICAND AND	RADICAND																															
0x08	SQRT_ROOT	Reserved																ROOT															

### 11.3.2 SQRT 控制状态寄存器 (SQRT\_CTRLSTS)

偏移地址: 0x00

复位值: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved								AUTO START	BUS HOLD	Reserved			SQRTDF	Reserved	SQRT START	SQRTEN
								rw	rw				r		rw	rw

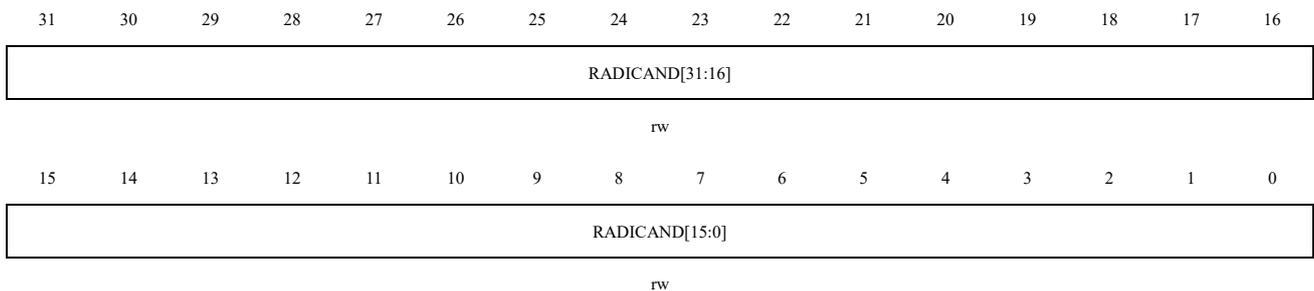
位域	名称	描述
[31:8]	Reserved	保留, 必须保持复位值。
[7]	AUTOSTART	自动计算使能, 自动计算是指, 每写入一次被开方数, 即开始一次开平方根运算 0: 禁能 1: 使能
[6]	BUSHOLD	计算过程中总线保持模式 0: 当计算开始, 未完成本次计算过程中, 软件读取平方根根数, 总线将不会保持 1: 当计算开始, 未完成本次计算过程中, 软件读取平方根根数, 总线保持
[5:4]	Reserved	保留, 必须保持复位值。
[3]	SQRTDF	本次计算完成标志 0: 未完成 1: 已完成
[2]	Reserved	保留, 必须保持复位值。
[1]	SQRTSTART	单次计算使能, 使能一次该控制位, 进行一次开平方根运算 0: 禁能 1: 使能

位域	名称	描述
[0]	SQRTEEN	SQRT 使能，只有该控制位使能后，才可以进行单次或自动计算 0: 禁能 1: 使能

### 11.3.3 SQRT 被开方数寄存器 (SQRT\_RADICAND)

偏移地址: 0x04

复位值: 0x00000000

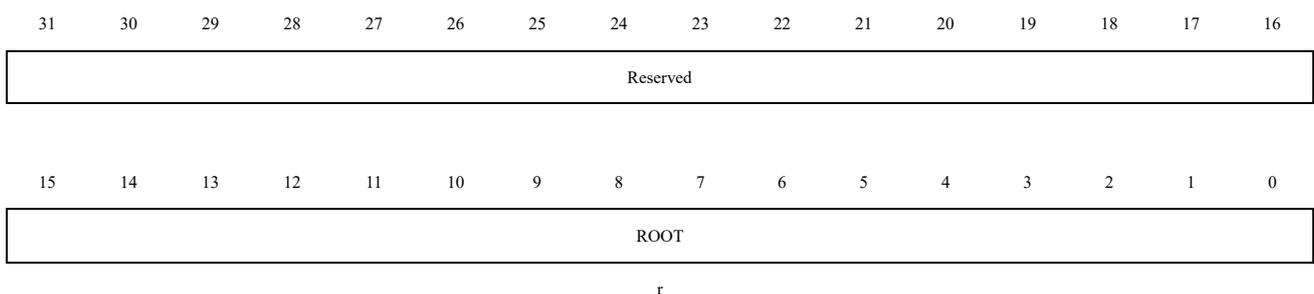


位域	名称	描述
[31:0]	RADICAND	32bit 被开方数

### 11.3.4 SQRT 根数寄存器 (SQRT\_ROOT)

偏移地址: 0x08

复位值: 0x00000000



位域	名称	描述
[31:16]	Reserved	保留，必须保持复位值。
[15:0]	ROOT	16 位开方根输出

## 12 高级定时器（TIM1）

### 12.1 TIM1 简介

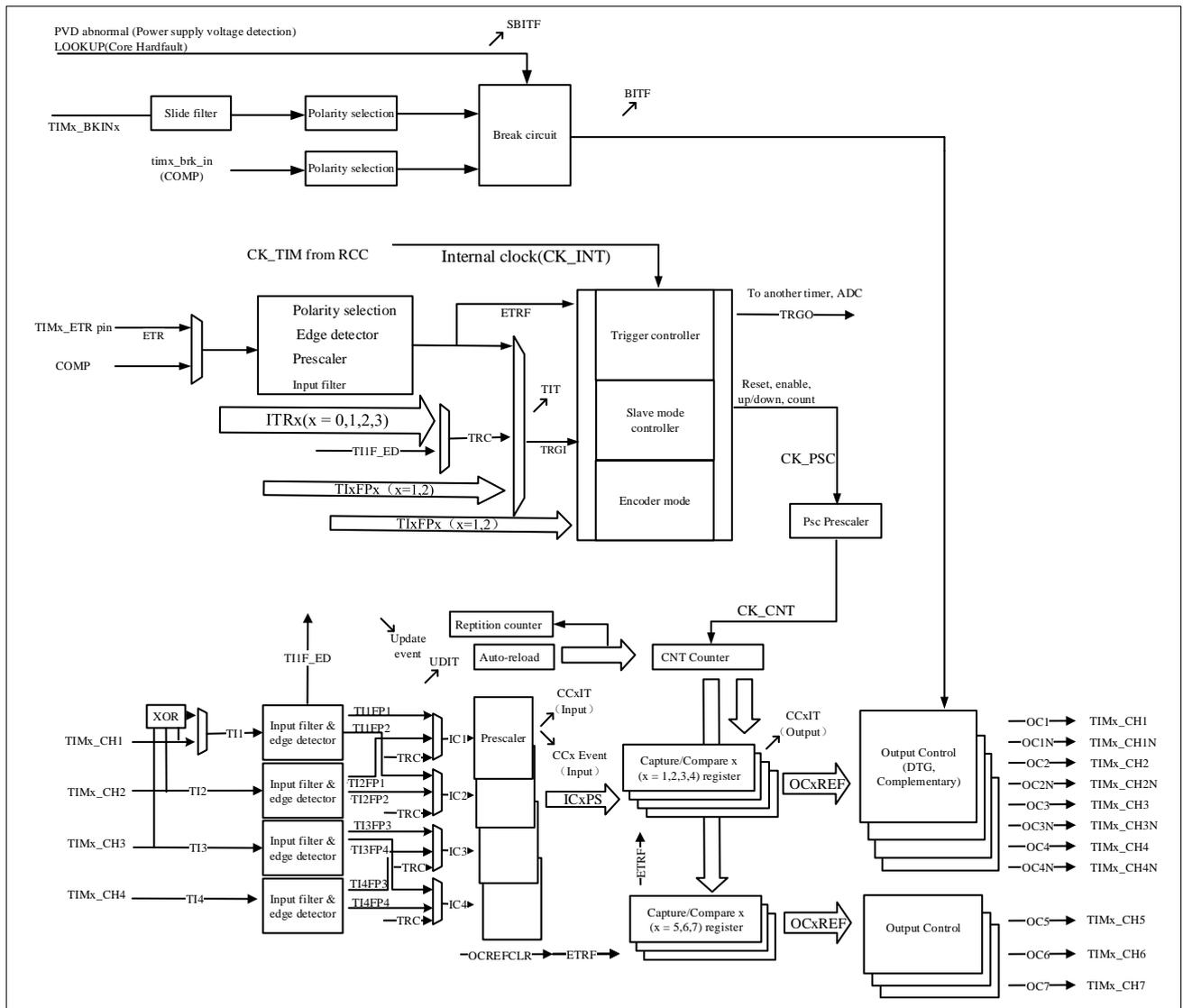
高级控制定时器（TIM1）主要用于以下场合：对输入信号进行计数、测量输入信号的脉冲宽度和产生输出波形等。

高级定时器具有互补输出功能、死区插入和刹车功能。适用于电机控制。

### 12.2 TIM1 主要特性

- 16 位自动装载计数器。（可实现向上计数、向下计数、向上/下计数）
- 16 位可编程预分频器。（分频系数可配置为 1 到 65536 之间的任意值）
- 可编程重复计数器。
- TIM1 最多 7 个通道。
- 7 个比较通道（CH1/2/3/4/5/6/7），工作模式为：PWM 输出、输出比较、单脉冲模式输出。
- 4 个捕获通道（CH1/2/3/4），工作模式为：输入捕获。
- 如下事件发生时产生中断/DMA：
  - ◆ 更新事件。
  - ◆ 触发事件。
  - ◆ 输入捕获。
  - ◆ 输出比较。
  - ◆ 刹车信号输入。
- 死区时间可编程的互补输出。
  - 对于 TIM1，通道 1、2、3、4 支持此功能。
- 可通过外部信号控制定时器。
- 多个定时器内部连接在一起，以实现定时器的同步或链接。
- TIM1\_CC5 用于比较器消隐。
- 增量（正交）编码器接口：用于追踪运行轨迹和解析旋转方位。
- 霍尔传感器接口：用于三相电机控制。

图 12-1 TIM1 框图



↓ 事件                      ↑ 中断和DMA 输出

捕获通道 1 输入可以来自 IOM 或比较器输出

## 12.3 TIM1 功能描述

### 12.3.1 时基单元

高级控制器的时基单元主要包括：预分频器、计数器、自动重装载寄存器和重复计数器。当时基单元工作时，软件可以随时读取和写入相应的寄存器（TIMx\_PSC、TIMx\_CNT、TIMx\_AR 和 TIMx\_REPCNT）。

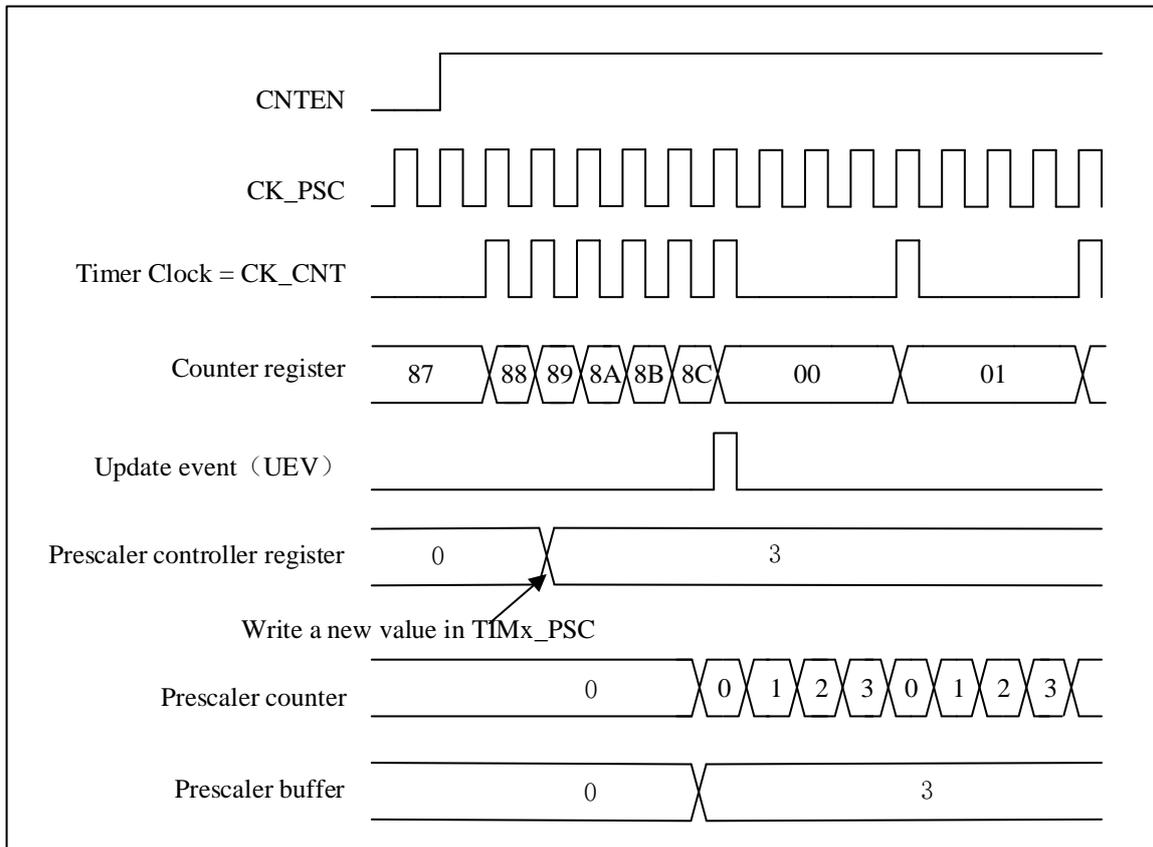
根据自动重装载预装载使能位（TIMx\_CTRL1.ARPEN）的设置，预装载寄存器的值会立即或在每次更新事件 UEV 时传输到影子寄存器。TIMx\_CTRL1.UPDIS=0 时，计数器上溢/下溢或软件设置 TIMx\_EVTGEN.UDGN 将生成更新事件。计数器 CK\_CNT 仅在 TIMx\_CTRL1.CNTEN 位被设置时有效。计

数器在 TIMx\_CTRL.CNTEN 位被设置后一个时钟周期之后开始计数。

### 12.3.1.1 预分频器描述

TIMx\_PSC 寄存器由一个 16 位计数器组成，可用于计数器时钟频率按 1 和 65536 之间的任意分频。因为这个控制器带有缓冲器，可以在运行时动态改变。新的预分频器值只有在下次更新事件中才会被采用。

图 12-2 当预分频的参数从 1 到 4，计数器的时序图



## 12.3.2 计数器模式

### 12.3.2.1 向上计数模式

使用向上计数模式，计数器将从 0 计数到寄存器 TIMx\_AR 的值，然后重置为 0。并产生一个计数器溢出事件。

如果设置 TIMx\_CTRL1.UPRS 位(选择更新请求)和 TIMx\_EVTGEN.UDGN 位,将产生一个更新事件(UEV)。但是 TIMx\_STS.UDITF 不会被硬件置起，因此不会产生更新中断或 DMA 更新请求。这是为了避免清除计数器时产生更新中断。

取决于 TIMx\_CTRL1.UPRS 的配置，当发生更新事件时，TIMx\_STS.UDITF 被设置，所有寄存器都会更新：

- 重复计数器被重新加载为 TIMx\_REPCNT 的内容
- 当 TIMx\_CTRL1.ARPEN = 1，预装载寄存器(TIMx\_AR)的值被更新到自动装载影子寄存器
- 预加载值 (TIMx\_PSC) 被重新加载到预分频器影子寄存器中

为了避免在向预装载寄存器中写入新值时更新影子寄存器，可以通过设置 TIMx\_CTRL1.UPDIS=1 来禁止更新事件。

当产生一个更新事件时，计数器仍将被清除，预分频器计数器也将被设置为 0（但预分频器值将保持不变）。

下图给出一些示例，展示了向上计数模式计数器在不同分频因子下的动作。

图 12-3 当内部时钟分频因子 = 2/N 时，向上计数的时序图

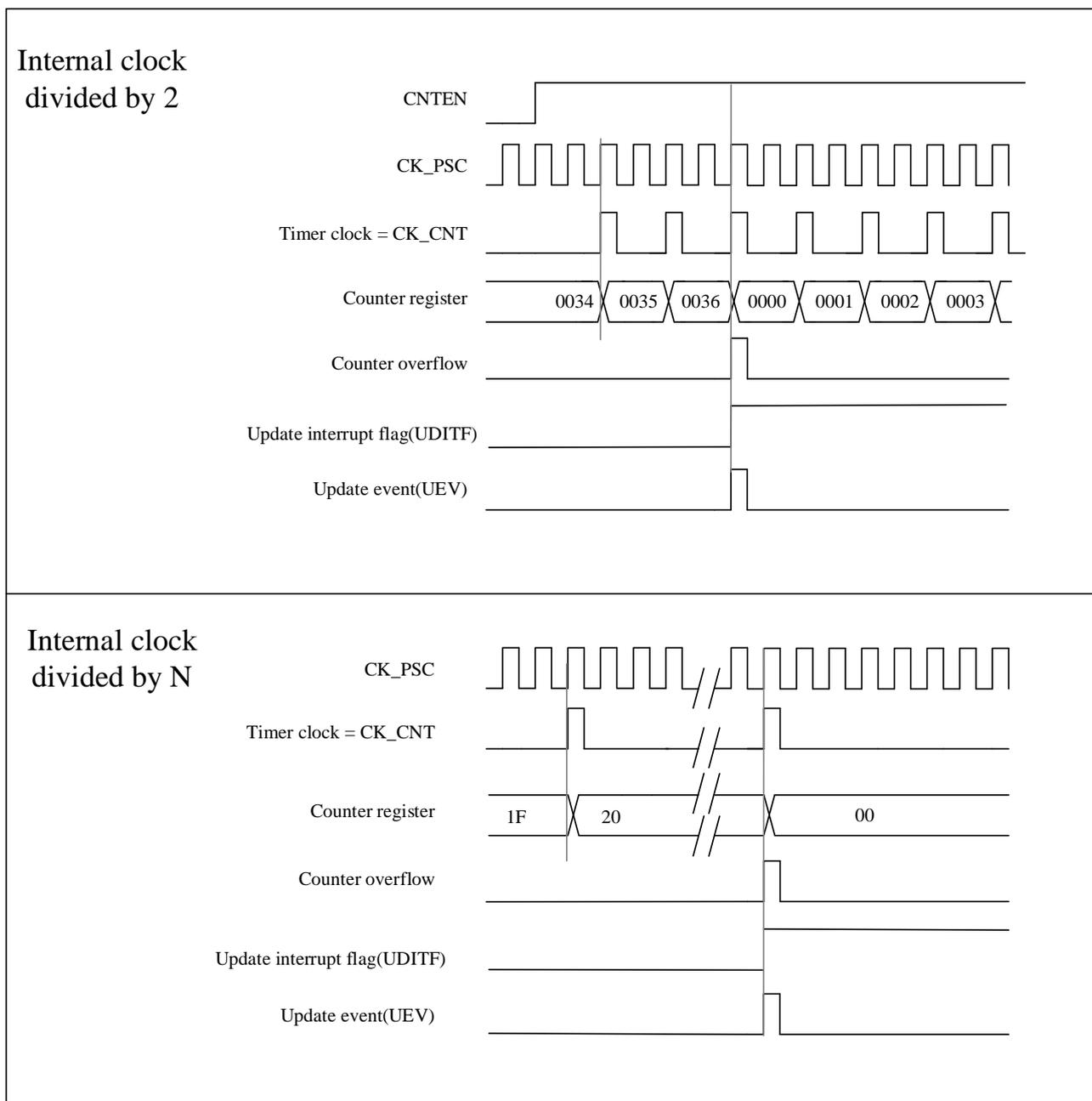
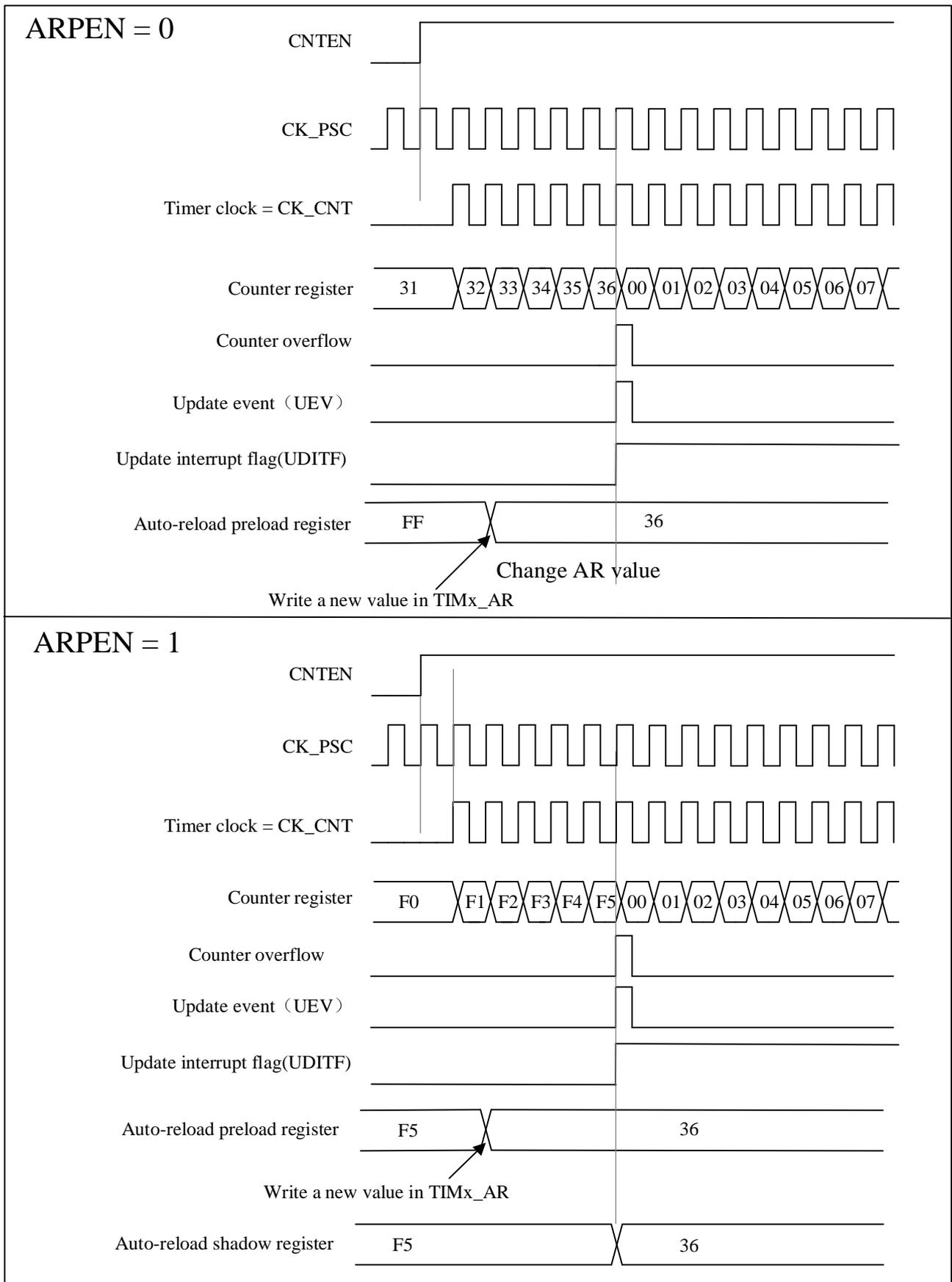


图 12-4 当 ARPEN=0/1 产生更新事件时，向上计数的时序图



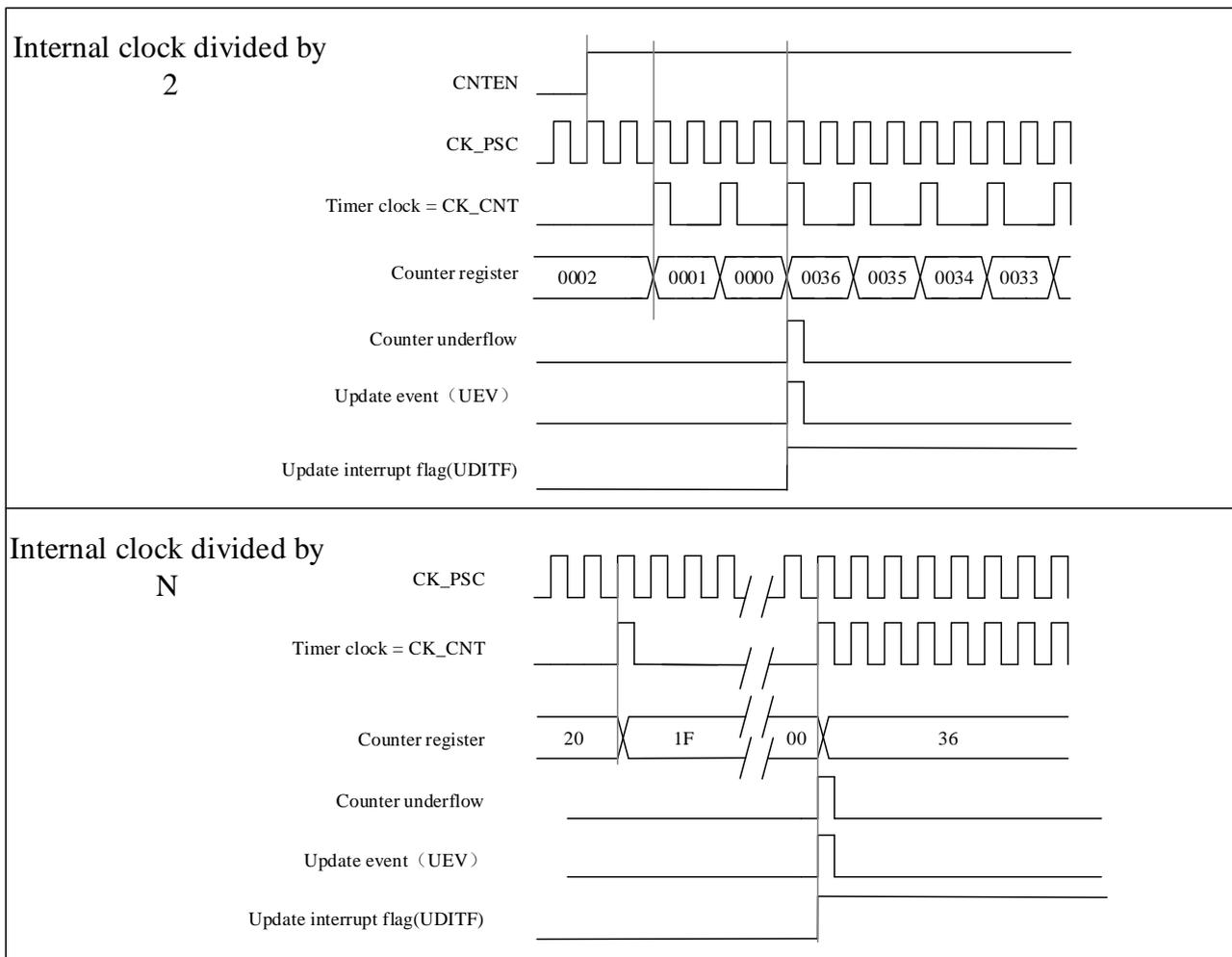
### 12.3.2.2 向下计数模式

向下计数模式，计数器将从寄存器 TIMx\_AR 的值减至 0，然后从自动重装载值重新开始，并产生计数器向下溢出事件

向下计数模式和向上计数模式配置更新事件和更新寄存器的过程相同，请查阅 12.3.2.1 章节。

下图给出一些示例，展示了向下计数模式计数器在不同分频因子下的动作。

图 12-5 内部时钟分频因子 = 2/N 时，向下计数时序图



### 12.3.2.3 中央对齐模式

#### 12.3.2.3.1 中央对齐对称模式

在中央对齐模式下，计数器从 0 增加到值 (TIMx\_AR) - 1，产生计数器溢出事件。然后，它从自动重装载值 (TIMx\_AR) 向下计数到 1，并生成一个计数器向下溢出事件。然后计数器重置为 0 并再次开始计数。

在这种模式下，TIMx\_CTRL1.DIR 方向位无效，由硬件更新和指定当前计数方向。当 TIMx\_CTRL1.CAMSEL 位不等于“00”时，中央对齐模式有效。

每次计数上溢和计数下溢时都会生成更新事件。或者，也可以通过设置 TIMx\_EVTGEN.UDGN 位（通过软件或使用从模式控制器）来生成更新事件。在这种情况下，计数器从 0 重新开始计数，预分频器的计数器也从 0 重新开始计数。

注：如果因为计数器溢出而产生更新，自动重载将在计数器重新载入之前被更新。

图 12-6 内部时钟分频因子 = 2/N，中央对齐时序图

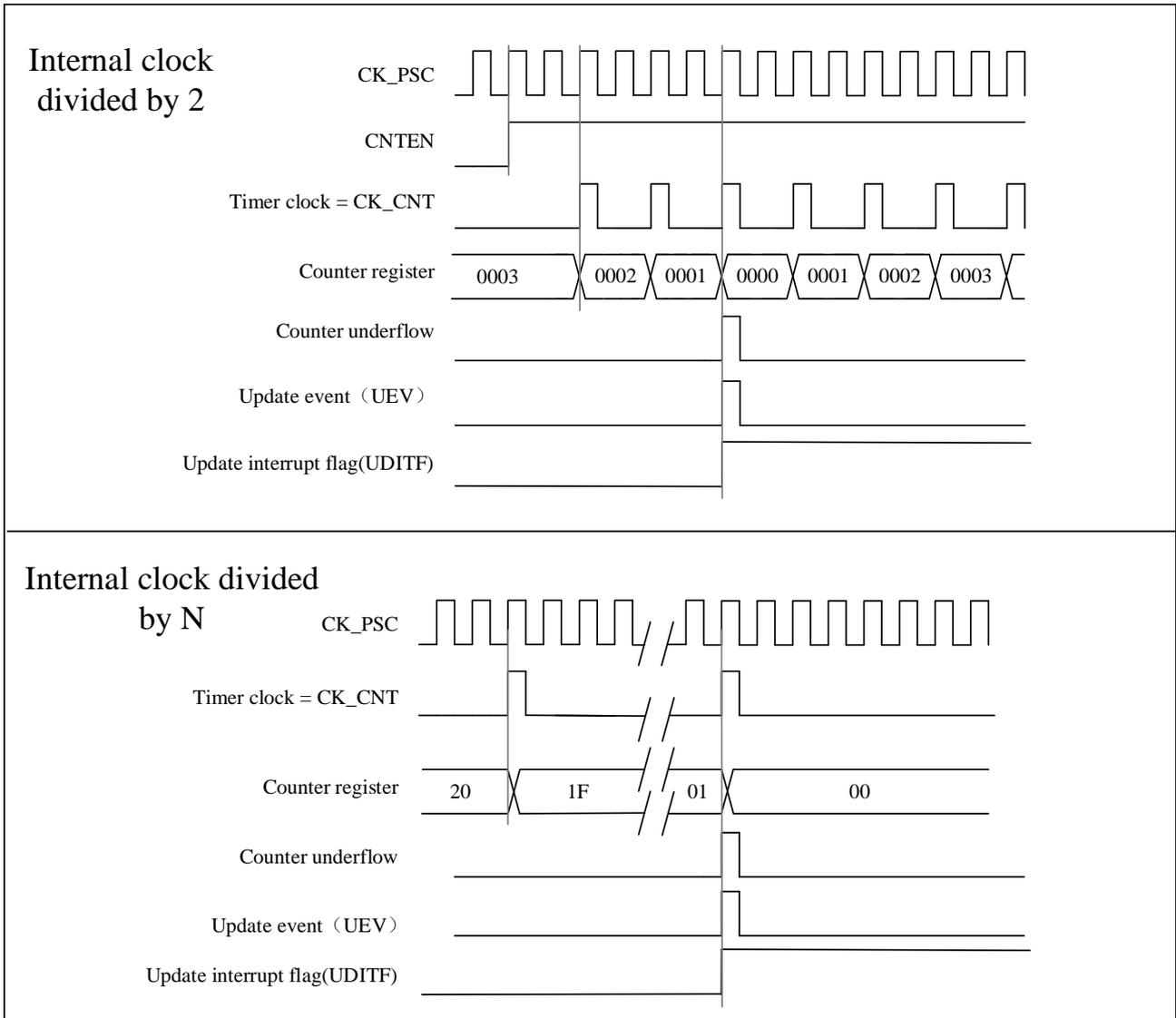
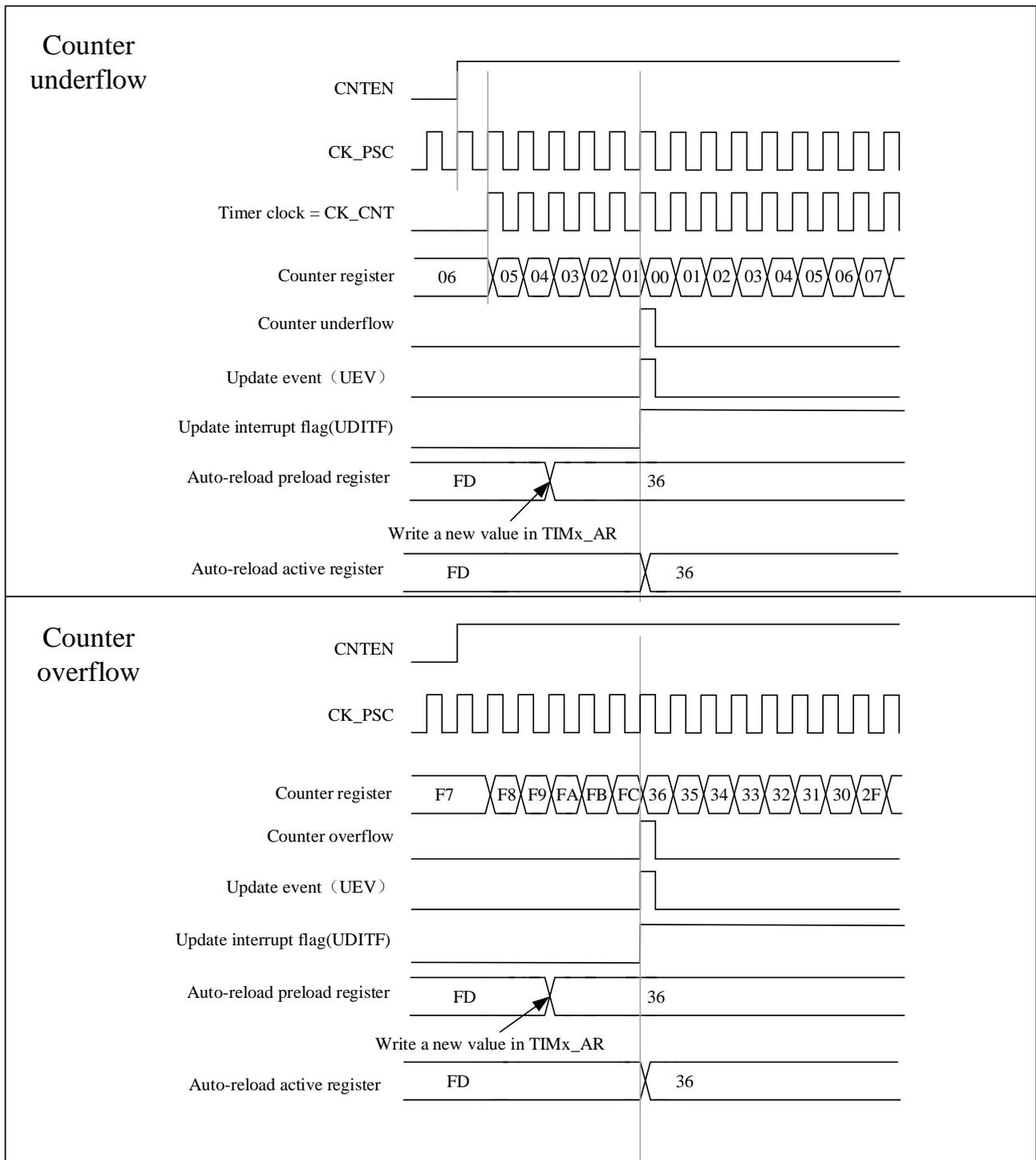


图 12-7 包含计数器上溢和下溢的中央对齐时序图(ARPEN=1)



### 12.3.2.3.2 中央对齐非对称模式

在中央对齐非对称模式下（TIMx\_CTRL1.ASYMMETRIC 为 1，TIMx\_CTRL1.CAMSEL[1:0]为非零），计数器从 0 计数到自动重载值（TIMx\_AR）-1，并产生计数器溢出事件，然后从自动重载值计数到 1，并产生计数器向下溢出事件，然后从 0 重新开始计数。

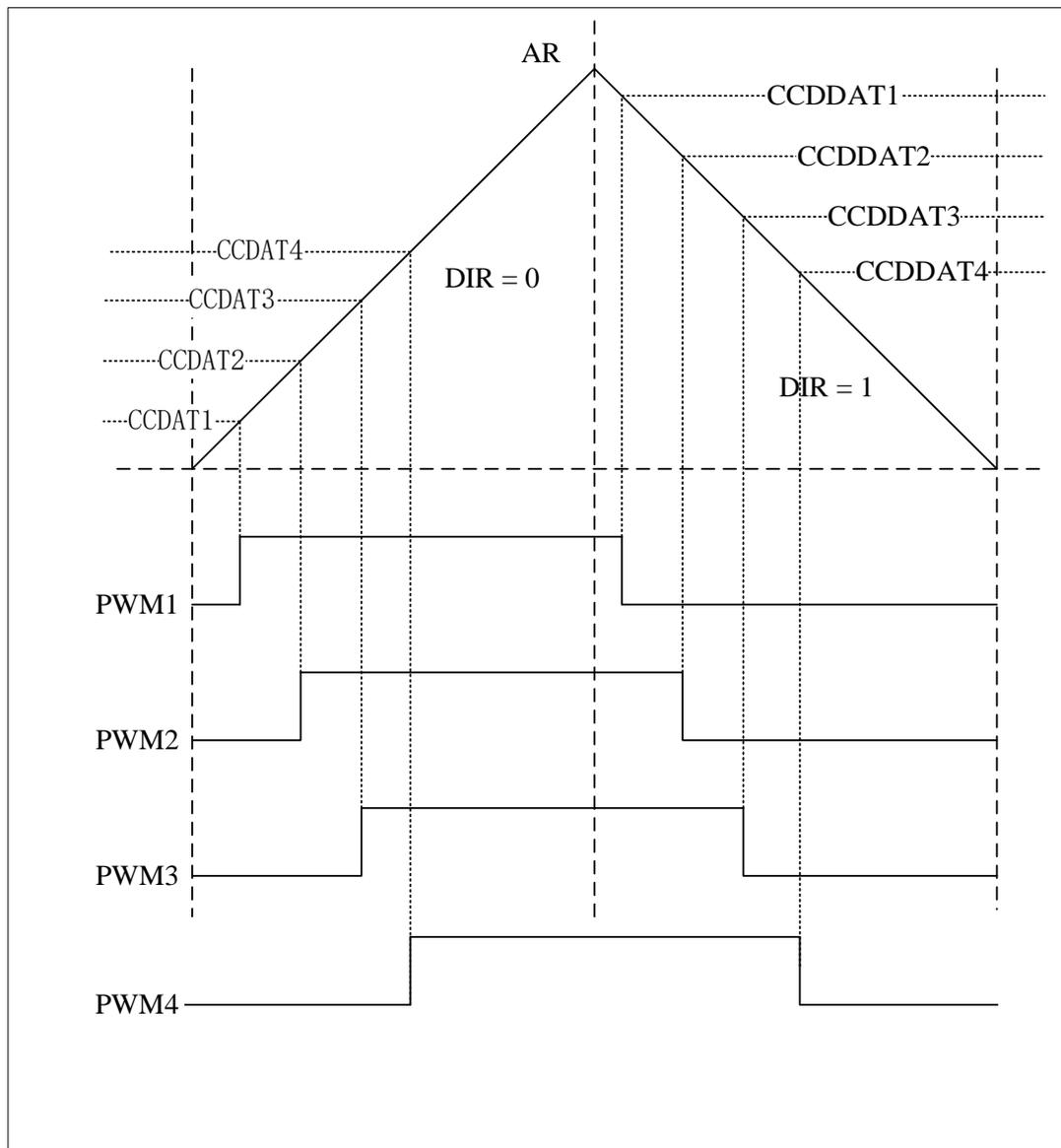
TIMx\_CTRL1.DIR 值不能在此模式下写入，它由硬件更新和指定当前计数方向。

当通道不是 1,2,3,4 时，比较值是 CCDDAT<sub>x</sub>。当死区时间发生器打开时，请注意，当 DIR=0 时，死区时间插入点是计数器值等于 CCDDAT<sub>x</sub> (x=1,2,3,4)，当 DIR=1 时，死区时间插入点是计数器值等于 CCDDAT<sub>x</sub> (x=1,2,3,4)。

每次计数上溢和计数下溢时都会产生更新事件。或者，也可以通过设置 TIM<sub>x</sub>\_EVTGEN.UDGN 位（通过软件或使用从模式控制器）产生更新事件。在这种情况下，在这种情况下，计数器从 0 重新开始计数，预分频器的计数器也从 0 重新开始计数。

注：如果因为计数器溢出而产生更新，自动重装载将在计数器重新载入之前被更新。

图 12-8 非对称模式对应的输出波形



### 12.3.3 重复计数器

第 12.3.1 章节的基本单元描述了生成更新事件 (UEV) 的条件。更新事件 (UEV) 实际上仅在重复计数器达到零时生成，这对于生成 PWM 信号非常有用。

这意味着每 N+1 计数器溢出或下溢一次，数据就会从预加载寄存器传输到影子寄存器，其中 N 是 TIMx\_REPCNT 中的值。

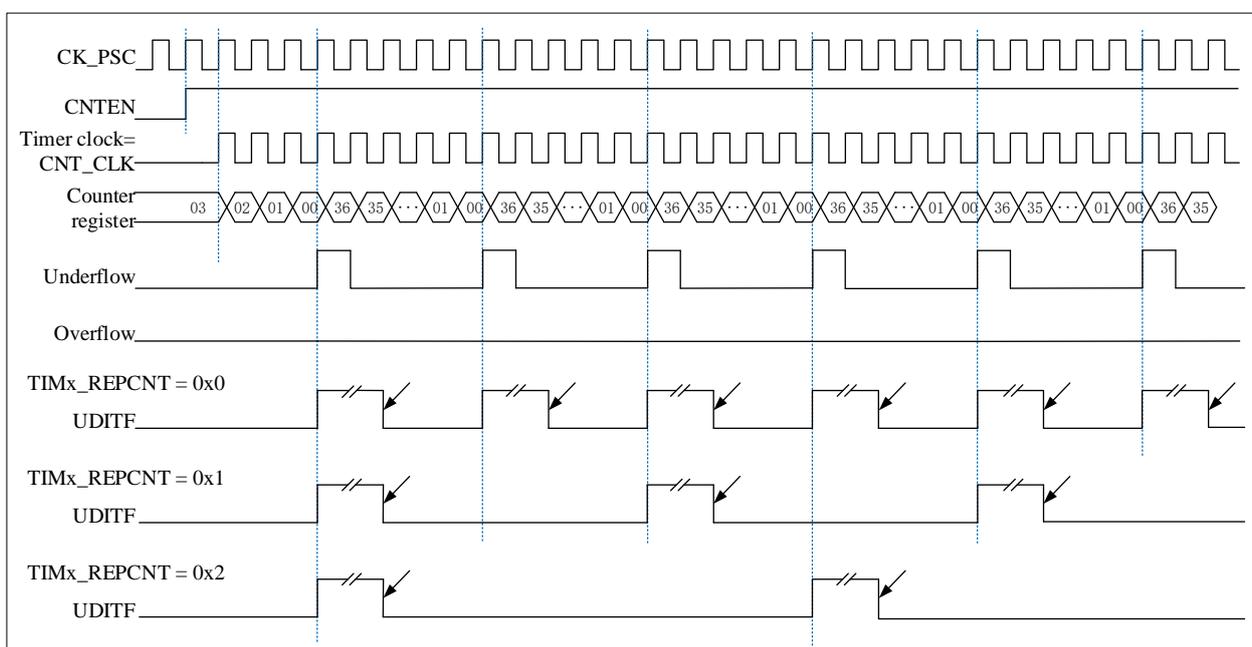
重复计数器递减：

- 在向上计数模式下，每次计数器达到最大值时，都会发生溢出
- 在向下计数模式下，每次计数器减至最小值时，都会发生下溢
- 在中央对齐模式下，每次计数上溢或下溢时

其重复率由 TIMx\_REPCNT 寄存器的值定。

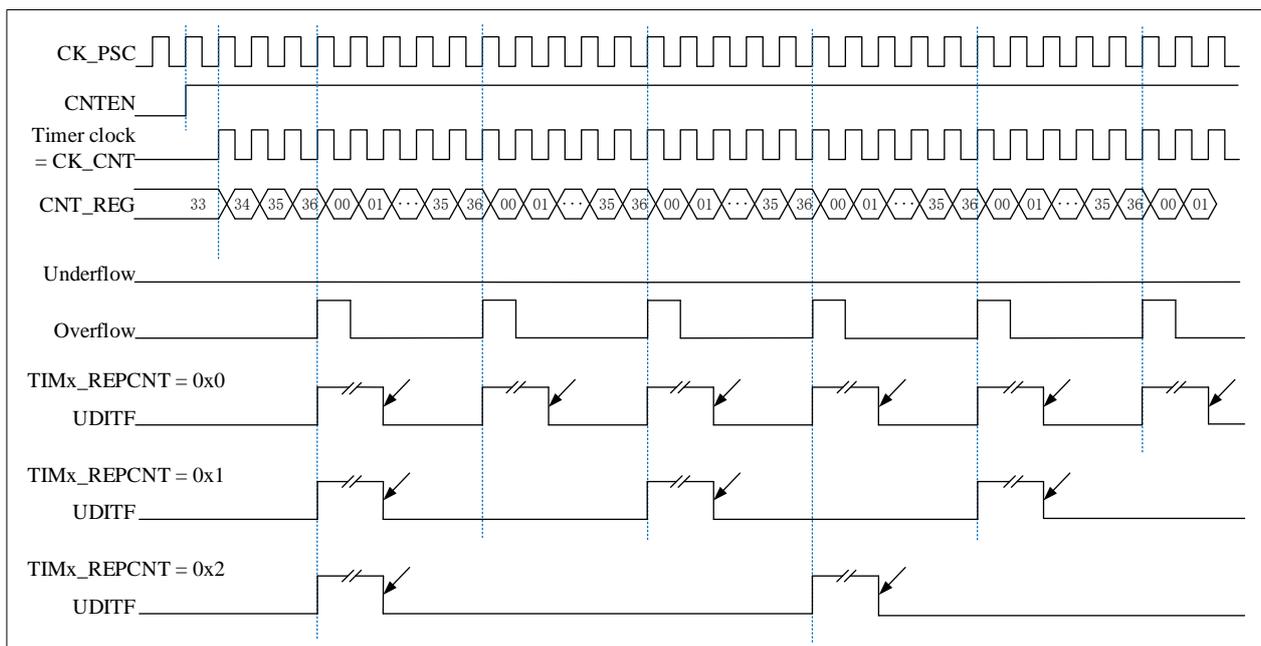
重复计数器具有自动重新加载功能。无论重复计数器的值如何，更新事件（通过从模式控制器设置 TIMx\_EVTGEN.UDGN 或硬件生成）都会立即发生。

图 12-9 向下计数模式下的重复计数时序图



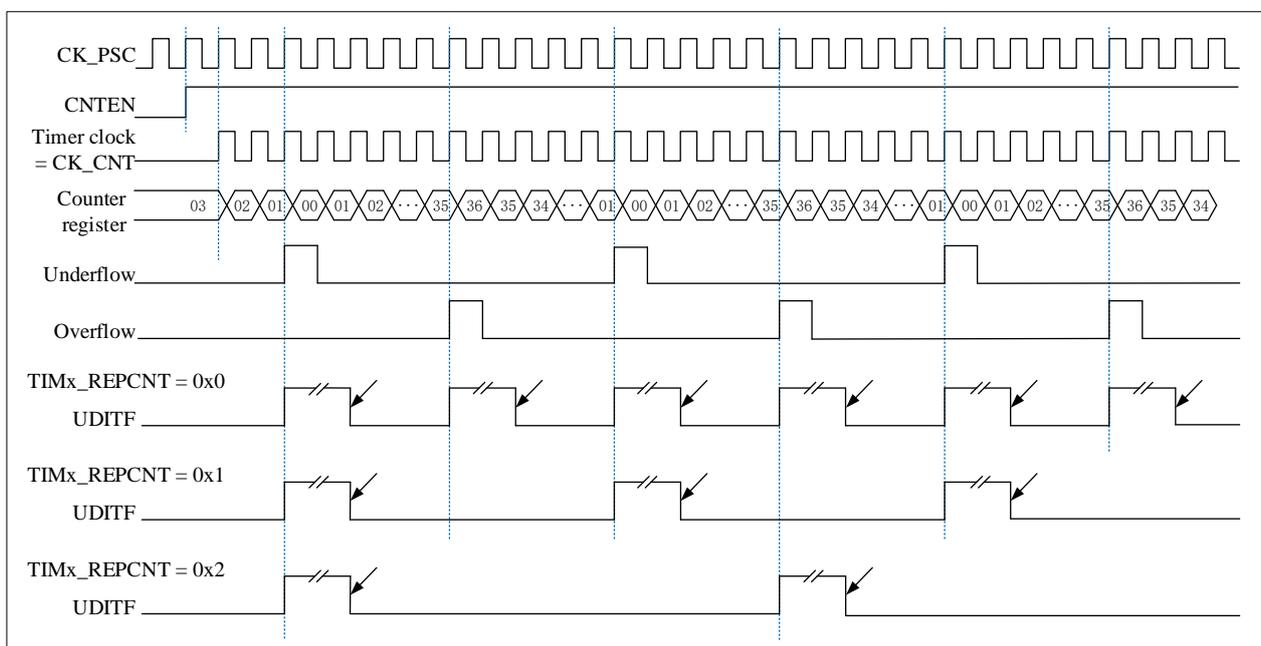
软件清除

图 12-10 向上计数模式下的重复计数时序图



软件清除

图 12-11 中央对齐模式下的重复计数时序图



软件清除

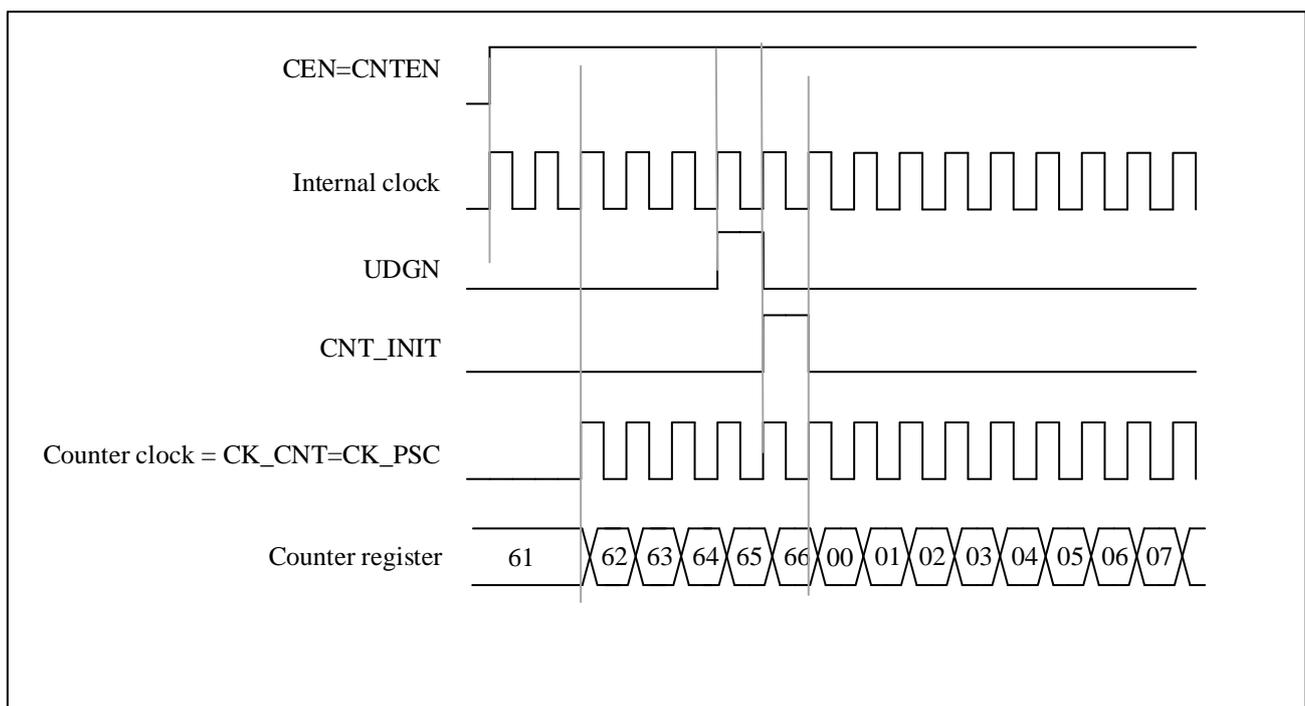
### 12.3.4 时钟选择

- CK\_INT 高级控制定时器的内部时钟：CK\_INT：
- 两种外部时钟模式：
  - 外部输入引脚
  - 外部触发输入 ETR
- 内部触发输入（ITRx）：一个定时器用作另一个定时器的预分频器

#### 12.3.4.1 内部时钟源(CK\_INT)

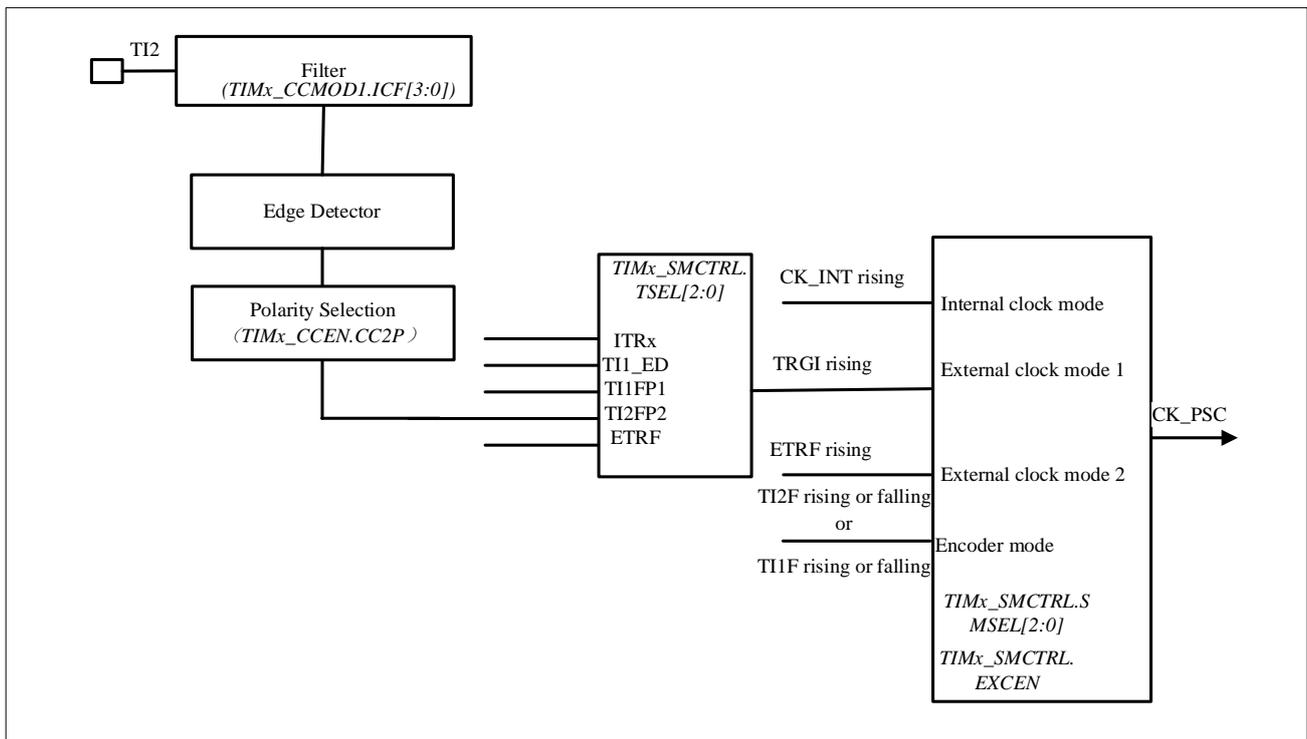
当 TIMx\_SMCTRL.SMSEL 等于“0000”时，从模式控制器被禁用。这三个控制位（TIMx\_CTRL1.CNTEN、TIMx\_CTRL1.DIR、TIMx\_EVTGEN.UDGN）只能由软件改变（TIMx\_EVTGEN.UDGN 除外，它保持自动清零）。前提是 TIMx\_CTRL1.CNTEN 位被软写为'1'，预分频器的时钟源由内部时钟 CK\_INT 提供。

图 12-12 正常模式下的控制电路，内部时钟除以 1



### 12.3.4.2 外部时钟源模式 1

图 12-13 TI2 外部时钟连接示例



通过配置 `TIMx_SMCTRL.SMSEL=0111` 选择该模式。计数器可以配置为在所选输入的时钟上升沿或下降沿进行计数。

例如，配置向上计数模式在 `TI2` 输入的时钟上升沿计数，配置步骤如下：

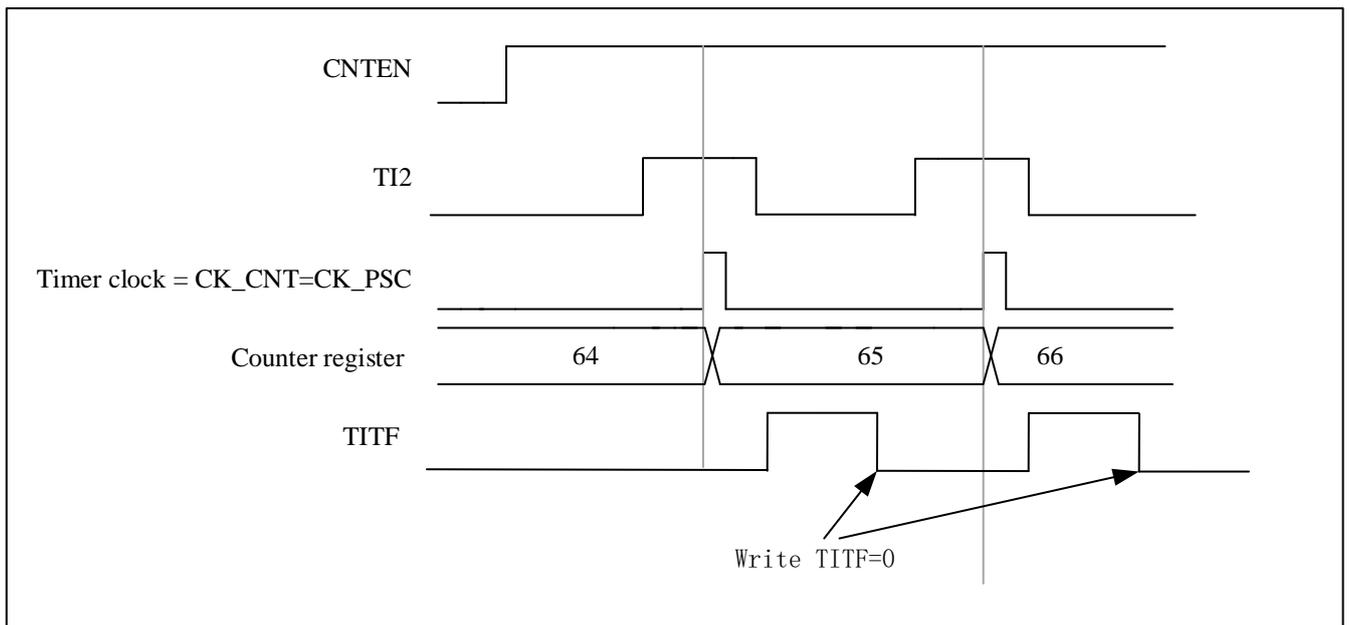
- 配置 `TIMx_CCMOD1.CC2SEL` 等于‘01’，`CC2` 通道配置为输入，`IC2` 映射到 `TI2`
- 配置 `TIMx_CCEN.CC2P` 等于‘0’，选择时钟上升沿极性
- 通过配置 `TIMx_CCMOD1.IC2F[3:0]` 选择输入滤波器带宽（如果不需要滤波器，保持 `IC2F` 位为‘0000’）
- 配置 `TIMx_SMCTRL.SMSEL` 等于‘0111’，选择定时器外部时钟模式 1
- 配置 `TIMx_SMCTRL.TSEL` 等于‘110’，选择 `TI2` 作为触发输入源
- 配置 `TIMx_CTRL1.CNTEN` 等于‘1’以启动计数器

注意：捕获预分频器不用于触发，所以不需要配置

当定时器时钟的上升沿出现在 `TI2=1` 时，计数器计数一次并且 `TIMx_STS.TITF` 标志被拉高。

`TI2` 的上升沿与计数器实际时钟之间的延迟取决于 `TI2` 输入端的再同步电路。

图 12-14 外部时钟模式 1 的控制电路

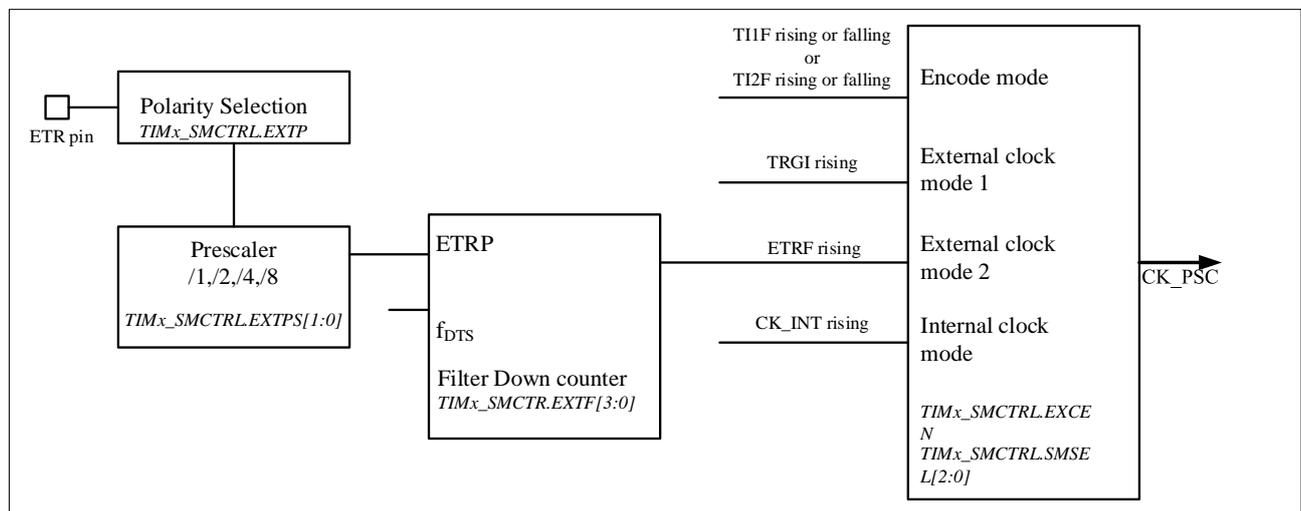


### 12.3.4.3 外部时钟源模式 2

此模式由 TIMx\_SMCTRL.EXCEN 选择等于 1。计数器可以在外部触发输入 ETR 的每个上升沿或下降沿计数。

下图为外部时钟源模式 2 的外部触发输入模块示意图。

图 12-15 外部触发输入框图



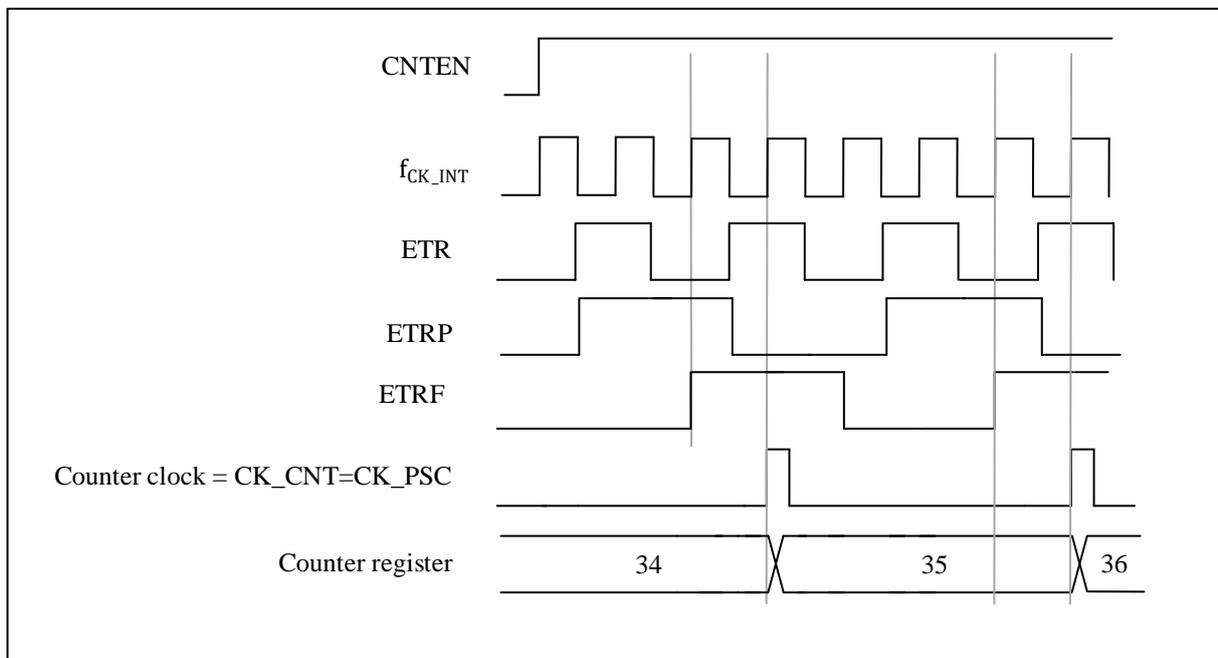
例如，使用以下配置步骤使向上计数器在 ETR 上每 2 个上升沿计数一次。

- 由于在这种情况下不需要过滤器，因此使 TIMx\_SMCTRL.EXTF[3:0] 等于‘0000’
- 通过使 TIMx\_SMCTRL.EXTPS[1:0] 等于 ‘01’ 来配置预分频器
- 通过设置 TIMx\_SMCTRL.EXTP 等于‘0’来选择 ETR 引脚的极性，ETR 的上升沿有效
- 外部时钟模式 2 通过设置 TIMx\_SMCTRL.EXCEN 等于‘1’来选择

- 通过设置 TIMx\_CTRL1.CNTEN 等于“1”启动计数器。

计数器每 2 个 ETR 上升沿计数一次。ETR 的上升沿与计数器的实际时钟之间的延迟是由于 ETRP 信号上的再同步电路造成的。

图 12-16 外部时钟模式 2 的控制电路

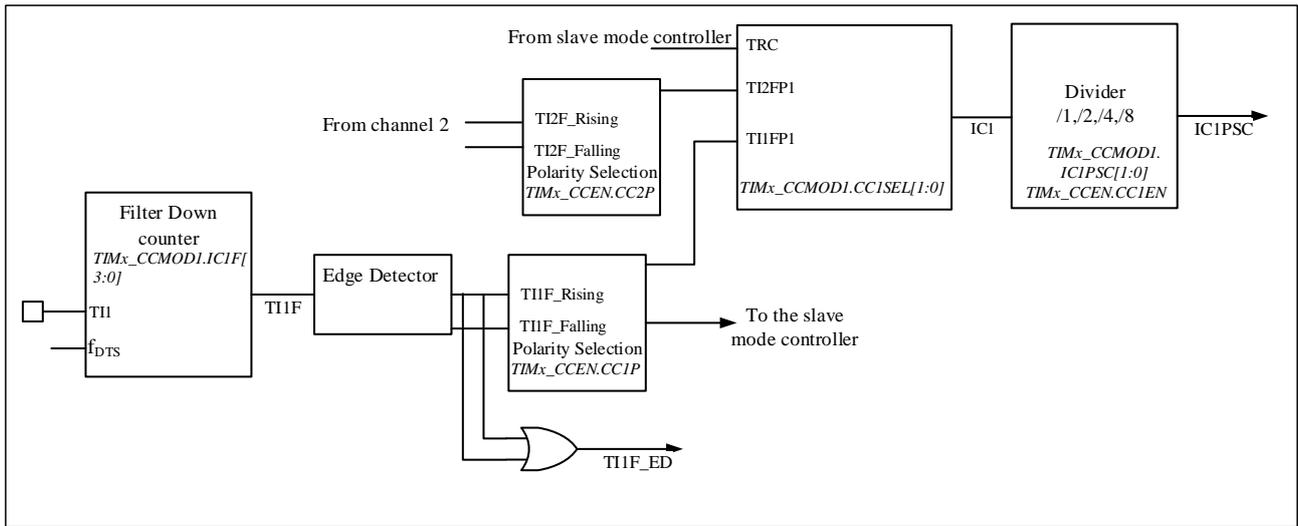


### 12.3.5 捕获/比较通道

捕获/比较通道包括捕获/比较寄存器和影子寄存器。输入部分由数字滤波器、多路复用器和预分频器组成。输出部分包括比较器和输出控制。

输入信号 T<sub>Ix</sub> 被采样和滤波以产生信号 T<sub>IxF</sub>。然后由极性选择功能的边沿检测器生成信号 (T<sub>IxF\_rising</sub> 或 T<sub>IxF\_falling</sub>)，其极性由 TIMx\_CCEN.CCxP 位选择。该信号可用作从模式控制器的触发输入。同时，信号 IC<sub>x</sub> 经过分频后送入捕获寄存器。下图显示了捕获/比较通道的框图。

图 12-17 捕获/比较通道（例如：通道 1 输入级）



输出部分生成一个中间波形 OCxRef（高电平有效）作为参考。极性作用在链的末端。

图 12-18 捕获/比较通道 1 主电路

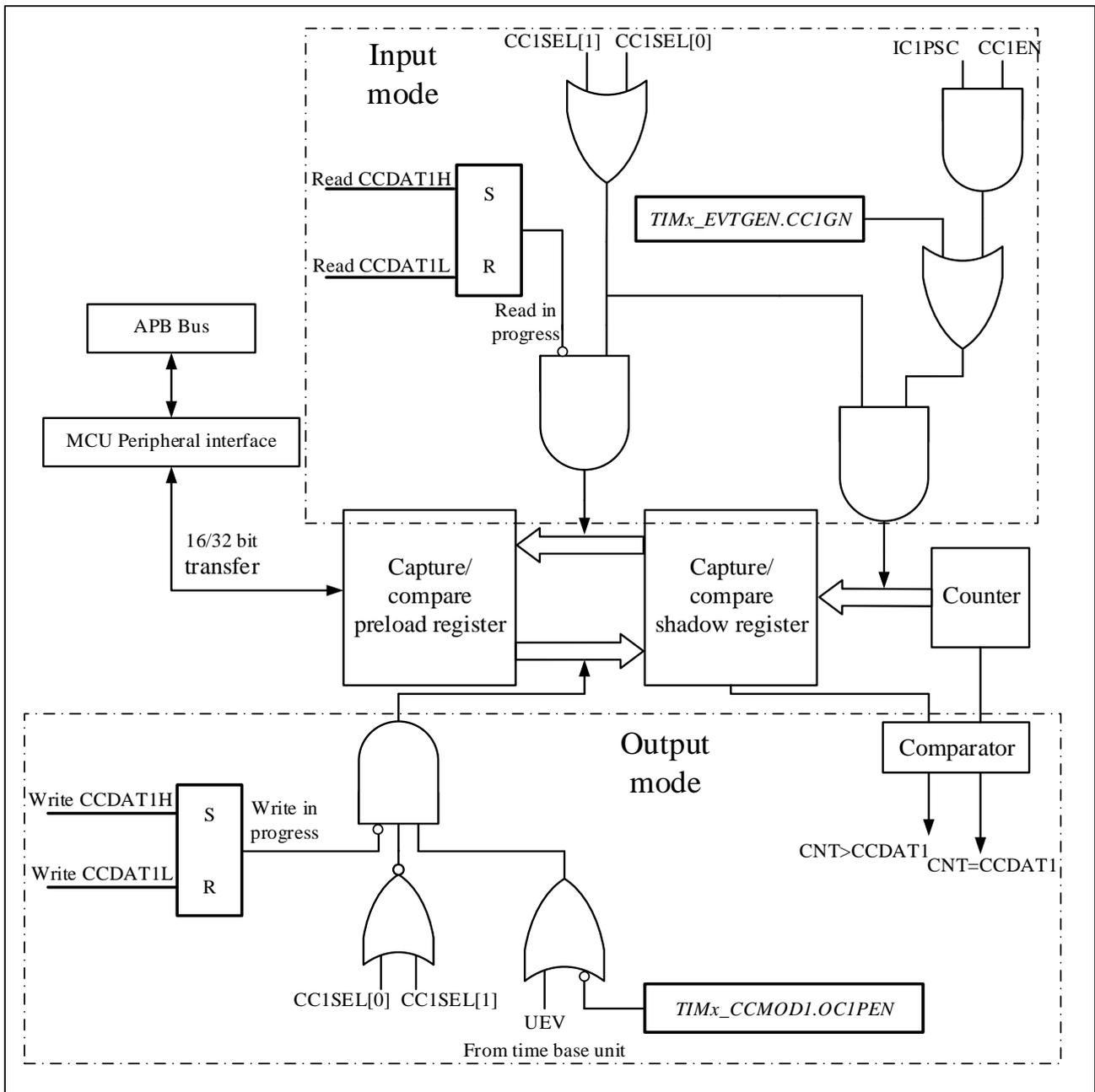
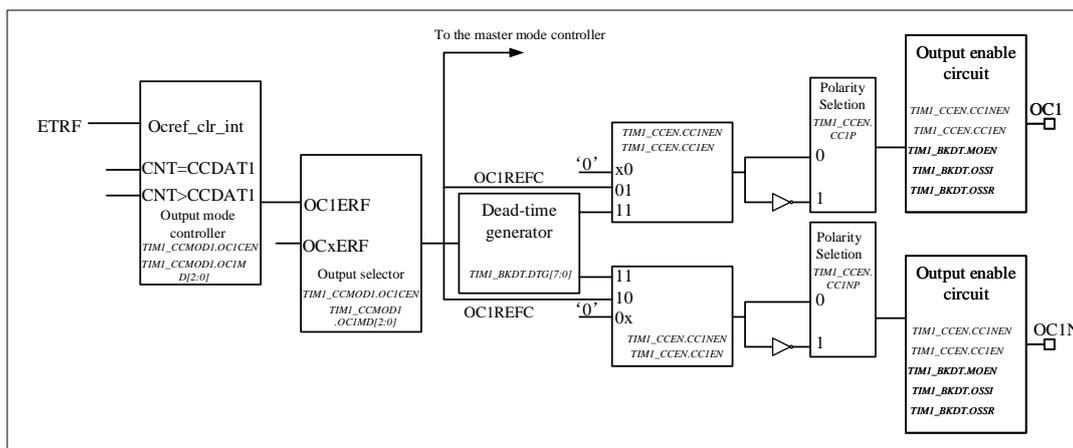


图 12-19 通道 x 的输出部分（以通道 1 为例子）



在捕获/比较时，读取和写入始终访问预加载的寄存器。两个具体工作流程如下：

在捕获模式下，捕获实际上是在影子寄存器中完成的，然后将影子寄存器中的值复制到预加载寄存器中。

在比较模式下，与捕获模式相反，预加载寄存器的值被复制到影子寄存器中，并与计数器进行比较。

### 12.3.6 输入捕获模式

在捕获模式下，TIMx\_CCDATx 寄存器用于在检测到 ICx 信号后锁存计数器值。

有一个捕获中断标志 TIMx\_STS.CCxITF，如果相应的中断使能被拉高，它可以发出中断或 DMA 请求。

TIMx\_STS.CCxITF 位在发生捕获事件时由硬件设置，并由软件或读取 TIMx\_CCDATx 寄存器清零。

当 TIMx\_CCDATx 寄存器中的计数器值被捕获并且 TIMx\_STS.CCxITF 被拉高时，重复捕获标志 TIMx\_STS.CCxOCF 设置为 1。与前者不同，TIMx\_STS.CCxOCF 通过向其写入 0 来清除。

为实现 TI1 输入的上升沿将计数器值捕获到 TIMx\_CCDAT1 寄存器中，配置流程如下：

- 选择有效输入：

将 TIMx\_CCMOD1.CC1SEL 配置为“01”。此时输入为 CC1 通道，IC1 映射到 TI1。

- 编程所需的输入滤波器持续时间：

通过配置 TIMx\_CCMOD1.ICx F 位来定义 TI1 输入的采样频率和数字滤波器的长度。示例：如果输入信号抖动多达 5 个内部时钟周期，我们必须选择比这 5 个时钟周期更长的滤波器持续时间。当检测到具有新电平的 8 个连续样本（以 f<sub>TIM1FILTCLK</sub> 频率采样）时，我们可以验证 TI1 上的转换。然后配置 TIMx\_CCMOD1.IC1F 到“1xxx”

- 通过配置 TIMx\_CCEN.CC1P=0，选择上升沿作为 TI1 通道的有效跳变极性

- 配置输入预分频器。在本例中，配置 TIMx\_CCMOD1.IC1PSC=‘00’以禁用预分频器，因为我们想要捕获每个有效转换

- 通过配置 TIMx\_CCEN.CC1EN=‘1’启用捕获。

如果要使能 DMA 请求，可以配置 TIMx\_DINTEN.CC1DEN=1。如果要使能相关中断请求，可以配置 TIMx\_DINTEN.CC1IEN=1。

### 12.3.7 PWM 输入模式

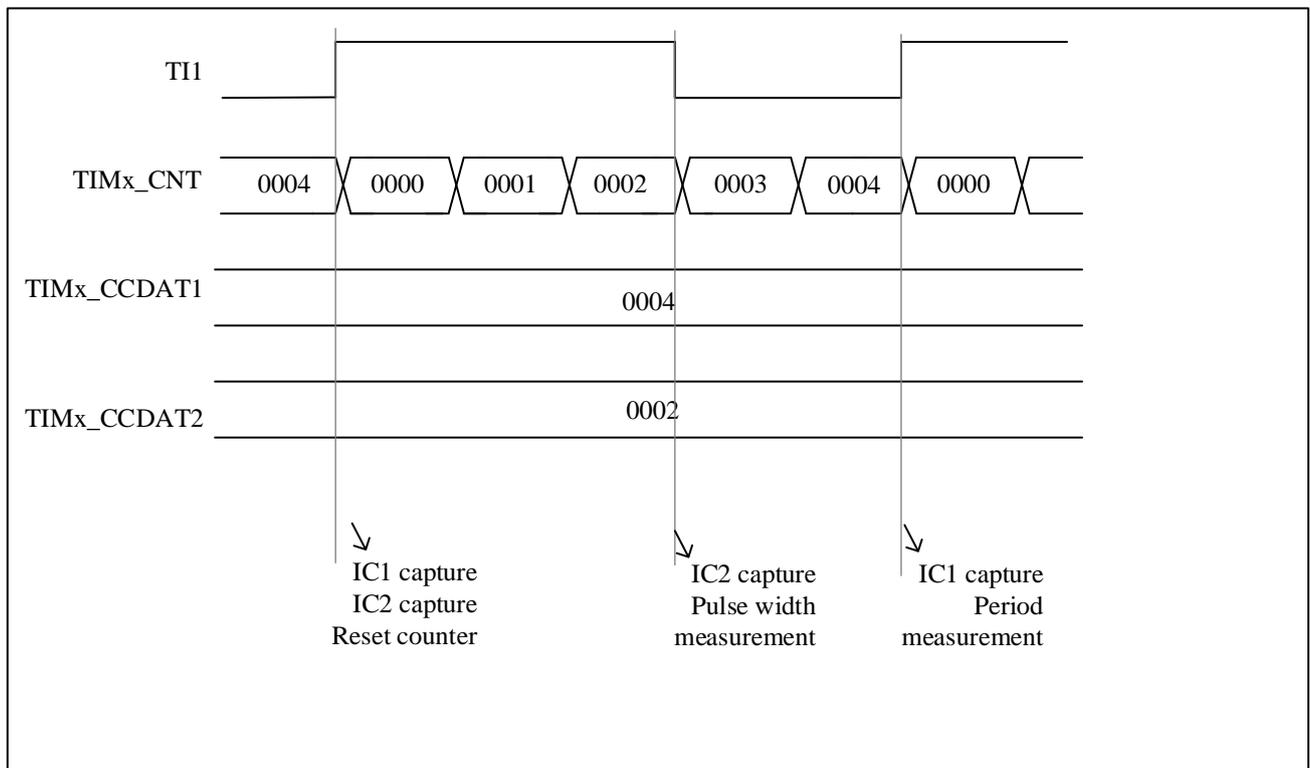
PWM 输入模式和普通输入捕获模式有一些区别，包括：

- 两个 ICx 信号映射到同一个 TIx 输入
- 两个 ICx 信号在极性相反的边沿有效
- 选择两个 TIxFP 信号之一作为触发输入
- 从机模式控制器配置为复位模式

例如，下面的配置流程可以用来知道 TI1 上 PWM 信号的周期和占空比（这取决于 CK\_INT 的频率和预分频器的值）。

- 配置 TIMx\_CCMOD1.CC1SEL 等于 ‘01’ 以选择 TI1 作为 TIMx\_CC1 的有效输入
- 配置 TIMx\_CCEN.CC1P 等于 ‘0’ 选择滤波定时器输入 1(TI1FP1) 的有效极性，在上升沿有效
- 配置 TIMx\_CCMOD1.CC2SEL 等于 ‘10’ 选择 TI1 作为 TIMx\_CC2 的有效输入
- 配置 TIMx\_CCEN.CC2P 等于 1 选择滤波定时器输入 2(TI1FP2) 的有效极性，下降沿有效
- 配置 TIMx\_SMCTRL.TSEL=101 选择 Filtered timer input 1 (TI1FP1) 作为有效触发输入
- 配置 TIMx\_SMCTRL.SMSEL=0100 配置从模式控制器为复位模式
- 配置 TIMx\_CCEN.CC1EN=1 和 TIMx\_CCEN.CC2EN=1 以启用捕获

图 12-20 PWM 输入模式时序



由于只有滤波器定时器输入 1 (TI1FP1) 和滤波器定时器输入 2 (TI2FP2) 连接到从模式控制器，因此 PWM 输入模式只能与 TIMx\_CH1/TIMx\_CH2 信号一起使用。

### 12.3.8 强制输出模式

在输出模式（TIMx\_CCMODx.CCxSEL=00）下，软件可以直接将输出比较信号强制为有效或无效电平。

用户可以设置 TIMx\_CCMODx.OCxMD=101 强制输出比较信号为有效电平。OCxREF 将被强制为高电平，OCx 得到与 CCxP 极性位相反的值。另一方面，用户可以设置 TIMx\_CCMODx.OCxMD=100 强制输出比较信号为无效电平，即 OCxREF 被强制为低电平。

在此模式下，TIMx\_CCDATx 影子寄存器和计数器的值仍然相互比较。

输出比较寄存器 TIMx\_CCDATx 和计数器 TIMx\_CNT 之间的比较对 OCxREF 没有影响。并且仍然可以设置标志。因此，仍然可以发送中断和 DMA 请求。

### 12.3.9 输出比较模式

用户可以使用此模式来控制输出波形，或指示一段时间已过。

当捕获/比较寄存器和计数器的值相同时，输出比较函数的操作如下：

- TIMx\_CCMODx.OCxMD 为输出比较模式，TIMx\_CCEN.CCxP 为输出极性。当比较匹配时，如果设置 TIMx\_CCMODx.OCxMD=000，则输出管脚将保持其电平；如果设置 TIMx\_CCMODx.OCxMD=001，则设置输出管脚有效；如果设置 TIMx\_CCMODx.OCxMD=010，则输出管脚将为 设置为无效；如果设置 TIMx\_CCMODx.OCxMD=011，则输出引脚将设置为翻转。
- 设置 TIMx\_STS.CCxITF
- 如果用户设置了 TIMx\_DINTEN.CCxIEN，将产生相应的中断
- 如果用户设置 TIMx\_DINTEN.CCxDEN 并设置 TIMx\_CTRL2.CCDSEL 选择 DMA 请求，将发送 DMA 请求

用户可以设置 TIMx\_CCMODx.OCxPEN 来选择是否使用捕获/比较预加载寄存器（TIMx\_CCDATx）来选择捕获/比较影子寄存器。

时间分辨率是计数器的一个计数周期。

在单脉冲模式下，输出比较模式也可用于输出单脉冲。

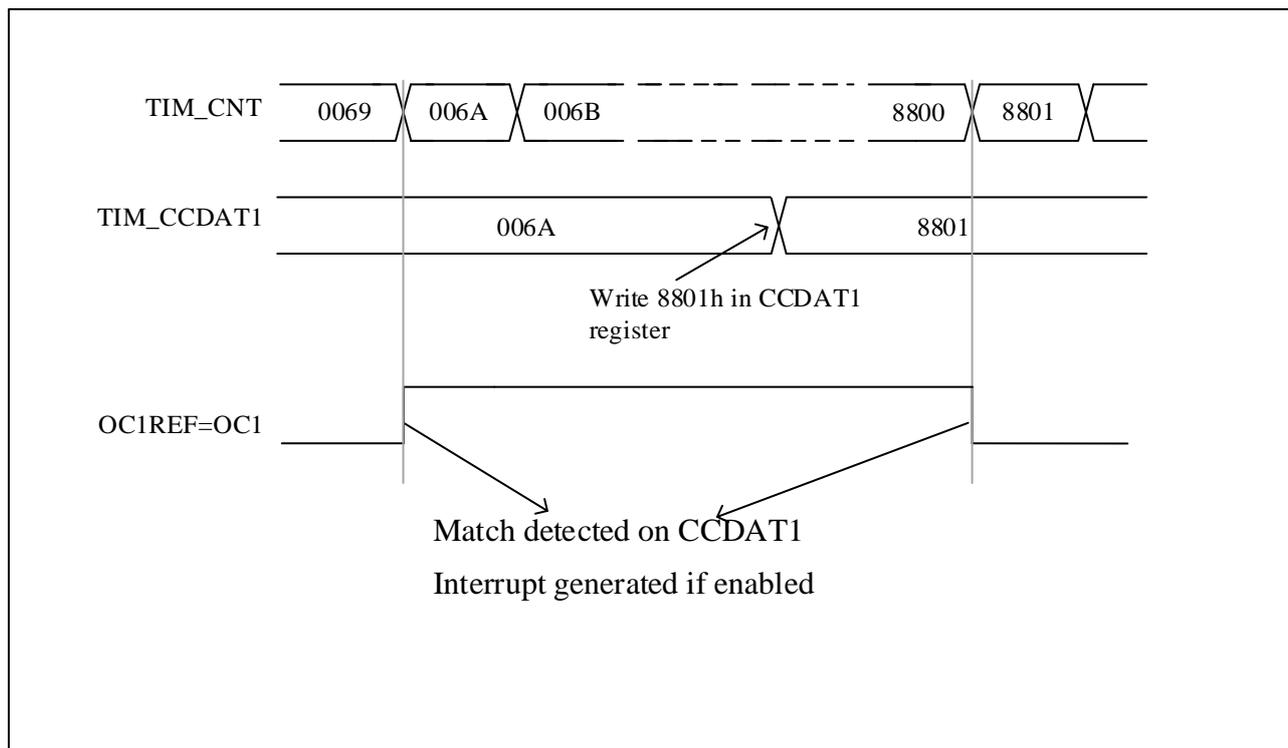
以下是输出比较模式的配置步骤：

- 首先，用户应该选择计数器时钟
- 其次，用所需数据设置 TIMx\_AR 和 TIMx\_CCDATx
- 如果用户需要产生中断，设置 TIMx\_DINTEN.CCxIEN
- 然后通过设置 TIMx\_CCEN.CCxP、TIMx\_CCMODx.OCxMD、TIMx\_CCEN.CCxEN 等选择输出模式
- 最后，设置 TIMx\_CTRL1.CNTEN 启用计数器

用户可以随时通过设置 TIMx\_CCDATx 来更新输出波形，只要不启用预加载寄存器。否则，TIMx\_CCDATx 影子寄存器将在下一次更新事件中更新。

例如：

图 12-21 输出比较模式，开启 OC1



### 12.3.10 PWM 模式

用户可以使用 PWM 模式产生一个信号，其占空比由 TIMx\_CCDATx 寄存器的值决定，其频率由 TIMx\_AR 寄存器的值决定。并且取决于 TIMx\_CTRL1.CAMSEL 的值，TIM 可以在边沿对齐模式或中央对齐模式下产生 PWM 信号。

用户可以通过设置 TIMx\_CCMODx.OCxMD=110 或设置 TIMx\_CCMODx.OCxMD=111 来设置 PWM 模式 1 或 PWM 模式 2。要启用预加载寄存器，用户必须设置相应的 TIMx\_CCMODx.OCxPEN。然后设置 TIMx\_CTRL1.ARPEN 自动重装载预加载寄存器。

用户可以通过设置 TIMx\_CCEN.CCxP 来设置 OCx 的极性。另一方面，要启用 OCx 的输出，用户需要在 TIMx\_CCEN 和 TIMx\_BKDT 中设置 CCxEN、CCxNEN、MOEN、OSSI 和 OSSR 的值的组合。

当 TIM 处于 PWM 模式时，TIMx\_CNT 和 TIMx\_CCDATx 的值总是相互比较。

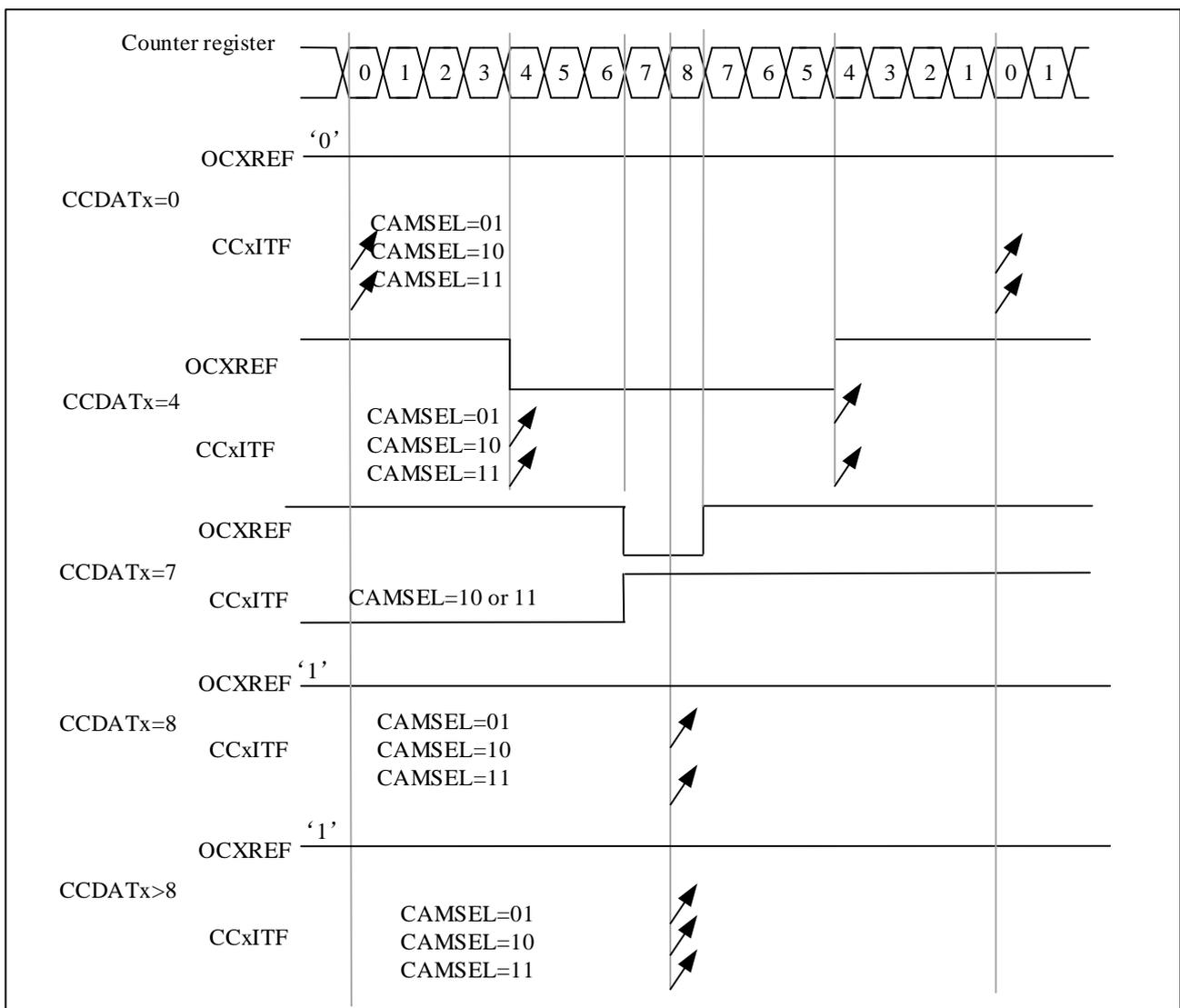
只有当更新事件发生时，预加载寄存器才会转移到影子寄存器。因此，用户必须在计数器开始计数之前通过设置 TIMx\_EVTGEN.UDGN 来复位所有寄存器。

#### 12.3.10.1 PWM 中央对齐模式

如果用户设置 TIMx\_CTRL1.CAMSEL 等于 01、10 或 11，PWM 中央对齐模式将被激活。比较标志的设置取决于 TIMx\_CTRL1.CAMSEL 的值。设置比较标志的情况有 3 种，仅当计数器向上计数时，仅当计数器向下计数时，或当计数器向上计数和向下计数时。用户不应通过软件修改 TIMx\_CTRL1.DIR，它是由硬件更新的。

中央对齐 PWM 波形示例如下，波形设置为：TIMx\_AR=8，PWM 模式 1，当计数器向下计数对应 TIMx\_CTRL1.CAMSEL=01 时设置比较标志。

图 12-22 中央对齐的 PWM 波形 (AR=8)



使用中央对齐模式时用户应注意的事项如下：

- 计数器向上或向下计数取决于 TIMx\_CTRL1.DIR 的值。注意不要同时更改 DIR 和 CAMSEL 位
- 用户在中央对齐模式下不要写计数器，否则会导致意想不到的结果。例如：
  - ◆ 如果写入计数器的值为 0 或者是 TIMx\_AR 的值，则方向会被更新，但不会产生更新事件
  - ◆ 如果写入计数器的值大于自动重载的值，则方向不会更新
- 为了安全起见，建议用户在启动计数器之前设置 TIMx\_EVTGEN.UDGN 以通过软件生成更新，并且在计数器运行时不要写入计数器

### 12.3.10.2 PWM 中央对齐非对称模式

关于 PWM 中央对齐非对称模式请查阅 12.3.2.3.2。

### 12.3.10.3 PWM 边沿对齐模式

边沿对齐模式有两种配置，向上计数和向下计数。

- 向上计数

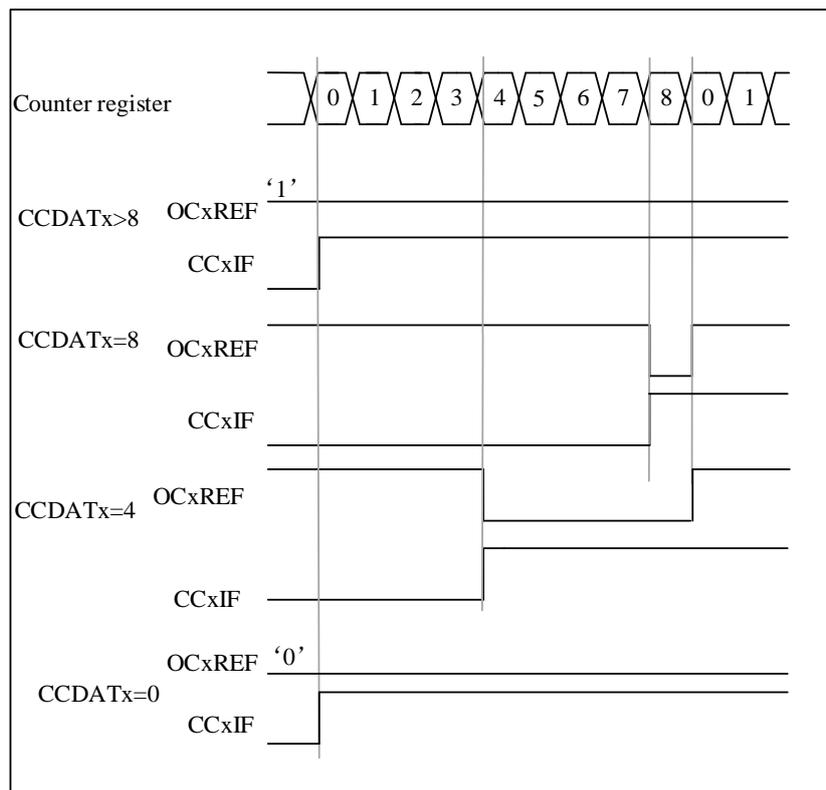
用户可以设置 TIMx\_CTRL1.DIR=0 使计数器向上计数。

PWM 模式 1 的示例：

当 TIMx\_CNT < TIMx\_CCxDATx 时，OCxREF 为高电平，否则为低电平。如果 TIMx\_CCxDATx 中的比较值大于自动重载值，则 OCxREF 将保持为 1。相反，如果比较值为 0，则 OCxREF 将保持为 0。

当 TIMx\_AR=8 时，PWM 波形如下：

图 12-23 边沿对齐 PWM 波形 (AR=8)



● 向下计数

用户可以设置 TIMx\_CTRL1.DIR=1 使计数器向下计数。

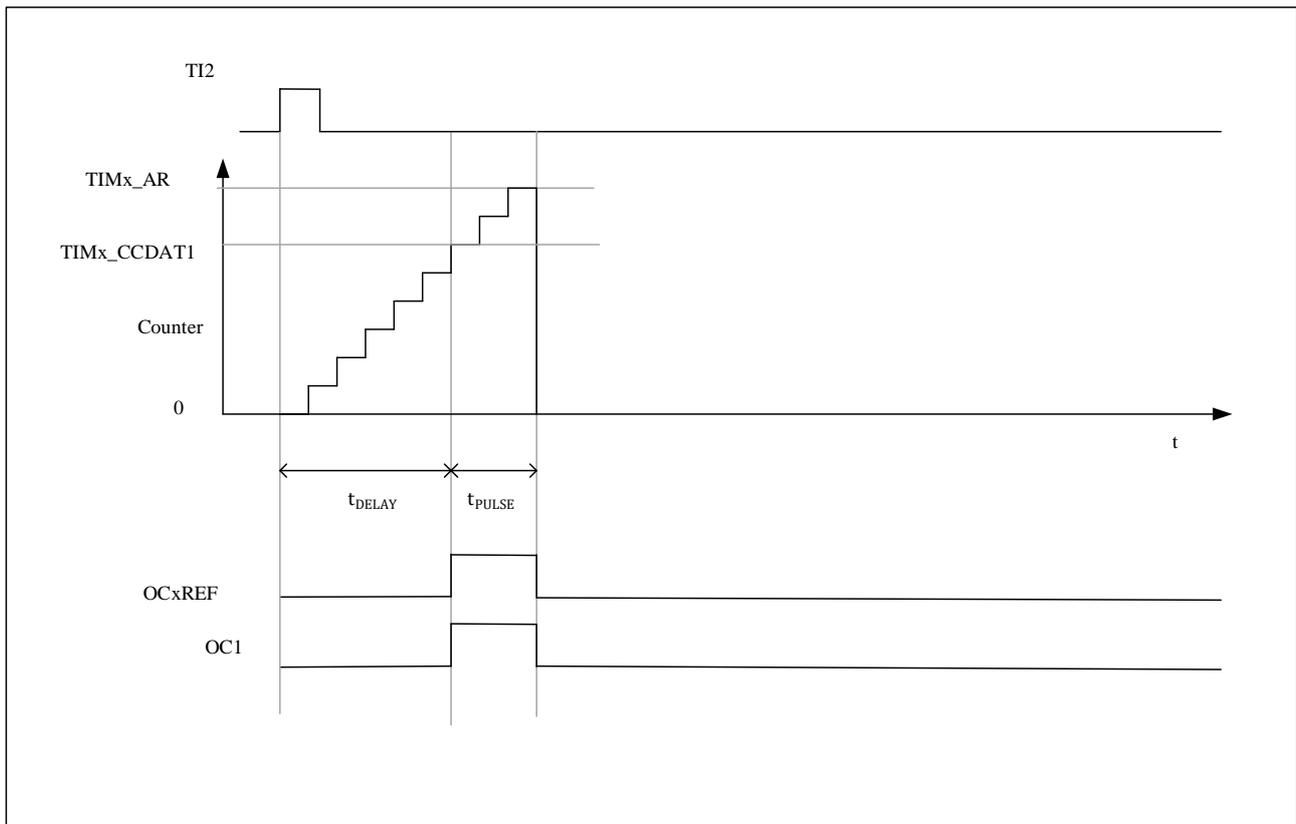
PWM 模式 1 的示例：

当 TIMx\_CNT > TIMx\_CCxDATx 时，OCxREF 为低电平，否则为高电平。如果 TIMx\_CCxDATx 中的比较值大于自动重载值，则 OCxREF 将保持为 1。

注：若第 n 个 PWM 周期 CCDATx 影子寄存器 ≥ AR 值，第 n+1 个 PWM 周期 CCDATx 的影子寄存器值是 0。在第 n+1 个 PWM 周期的计数器为 0 的时刻，虽然计数器 = CCDATx 影子寄存器的值 = 0，OCxREF = ‘0’，但不会产生比较事件。

### 12.3.11 单脉冲模式

在单脉冲模式(ONEPM)中，接收到触发信号，经过可控延迟 t<sub>DELAY</sub> 后产生脉宽可控的脉冲 t<sub>PULSE</sub>。输出模式需要配置为输出比较模式或 PWM 模式。选择单脉冲模式后，计数器会在更新事件 UEV 产生后停止计数。

**图 12-24 单脉冲模式示例**


以下是单脉冲模式的示例：

从 TI2 输入检测到上升沿触发，延迟  $t_{\text{DELAY}}$  后在 OC1 上产生宽度为  $t_{\text{PULSE}}$  的脉冲。

1. 计数器配置：向上计数，计数器  $\text{TIMx\_CNT} < \text{TIMx\_CCDAT1} \leq \text{TIMx\_AR}$ ；
2. TI2FP2 映射到 TI2,  $\text{TIMx\_CCMOD1.CC2SEL} = '01'$ ；TI2FP2 配置为上升沿检测,  $\text{TIMx\_CCEN.CC2P} = '0'$ ；
3. TI2FP2 充当从模式控制器的触发器（TRGI）并启动计数器， $\text{TIMx\_SMCTRL.TSEL} = '110'$ ， $\text{TIMx\_SMCTRL.SMSEL} = '0110'$ （触发模式）；
4.  $\text{TIMx\_CCDAT1}$  写入要延迟的计数值（ $t_{\text{DELAY}}$ ）， $\text{TIMx\_AR} - \text{TIMx\_CCDAT1}$  为脉宽  $t_{\text{PULSE}}$  的计数值；
5. 配置  $\text{TIMx\_CTRL1.ONEPM} = 1$  使能单脉冲模式，配置  $\text{TIMx\_CCMOD1.OC1MD} = '111'$  选择 PWM2 模式；
6. 等待 TI2 有外部触发事件，OC1 输出一个单脉冲波形；

### 12.3.11.1 特殊情况：OCx 快速使能：

在单脉冲模式下，通过 TIx 输入检测到一个边沿，并触发计数器开始计数到比较值，然后输出一个脉冲。这些操作限制了可以达到的最小延迟  $t_{\text{DELAY}}$ 。

您可以设置  $\text{TIMx\_CCMODx.OCxFEN} = 1$  开启 OCx 快速使能，在触发上升沿后，OCxREF 信号将被强制转换为与比较匹配立即发生的电平相同的电平，而不管比较结果如何。OCxFEN 快速使能仅在通道模式配置为 PWM1 和 PWM2 模式时生效。

### 12.3.12 在外部事件上清除 OCxREF 信号

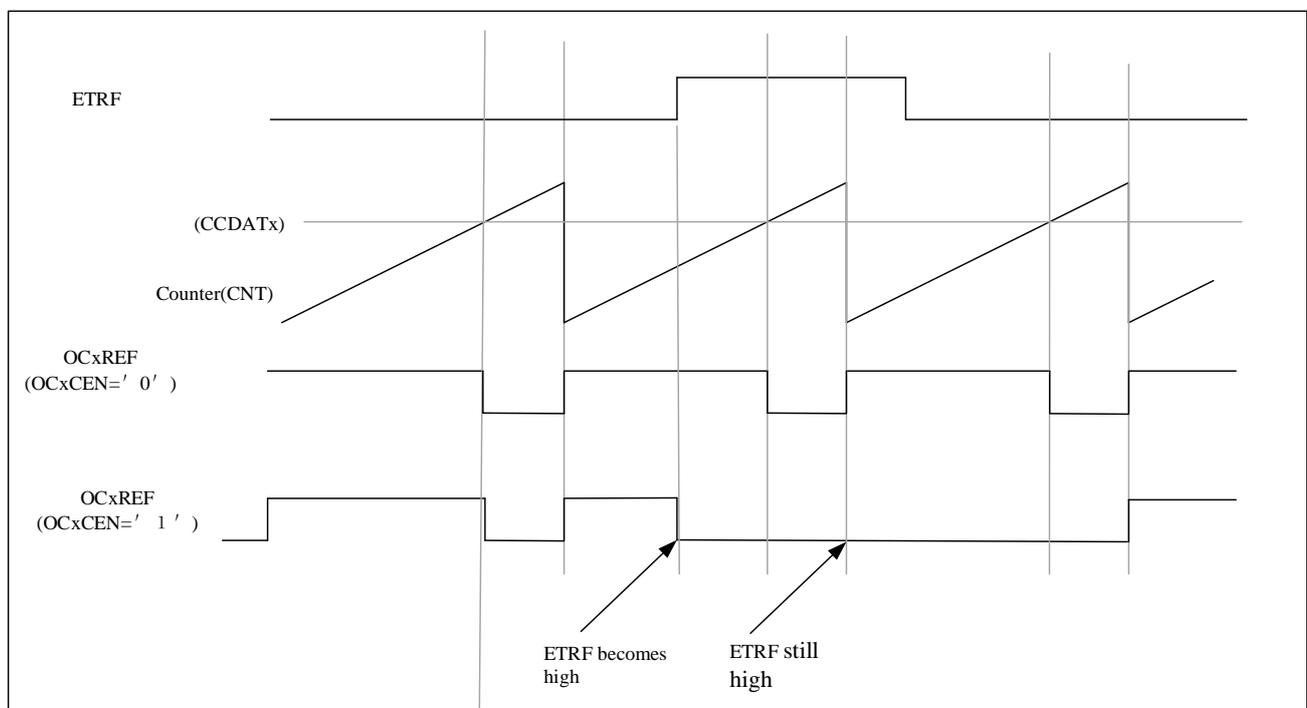
如果用户设置 `TIMx_CCMODx.OCxCEN=1`，ETRF 输入的高电平可用于驱动 OCxREF 信号为低电平，OCxREF 信号将保持低电平，直到下一次 UEV 发生。只有输出比较和 PWM 模式可以使用该功能。在强制模式下不能使用。

例如：为了控制电流，用户可以将 ETR 信号连接到比较器的输出端，ETR 的操作如下：

- 设置 `TIMx_SMCTRL.EXTPS=00` 禁用外部触发预分频器。
- 设置 `TIMx_SMCTRL.EXCEN=0` 禁用外部时钟模式 2。
- 设置 `TIMx_SMCTRL.EXTP` 和 `TIMx_SMCTRL.EXTF`，根据需要配置外触发极性和外触发滤波器。

例：当 ETRF 输入变高时，OCxREF 信号对于不同的 OCxCEN 值的行为。在这种情况下，定时器设置为 PWM 模式。

图 12-25 清除 TIMx 的 OCxREF



### 12.3.13 互补输出和死区插入

高级控制定时器可以输出两个互补信号，并管理输出的关闭和打开。这称为死区时间。用户应根据连接到输出的设备及其特性调整死区时间。

用户可以通过设置 `TIMx_CCEN.CCxP` 和 `TIMx_CCEN.CCxNP` 来选择输出的极性。并且此选择对于每个输出都是独立的。

用户可以通过设置几个控制位的组合来控制互补信号 `OCx` 和 `OCxN`，它们分别是 `TIMx_CCEN.CCxEN`、`TIMx_CCEN.CCxNEN`、`TIMx_BKDT.MOEN`、`TIMx_CTRL2.OIx`、`TIMx_CTRL2.OIxN`、`TIMx_BKDT.OSSI` 和 `TIMx_BKDT.OSSR`。当切换到空闲状态时，死区时间将被激活。

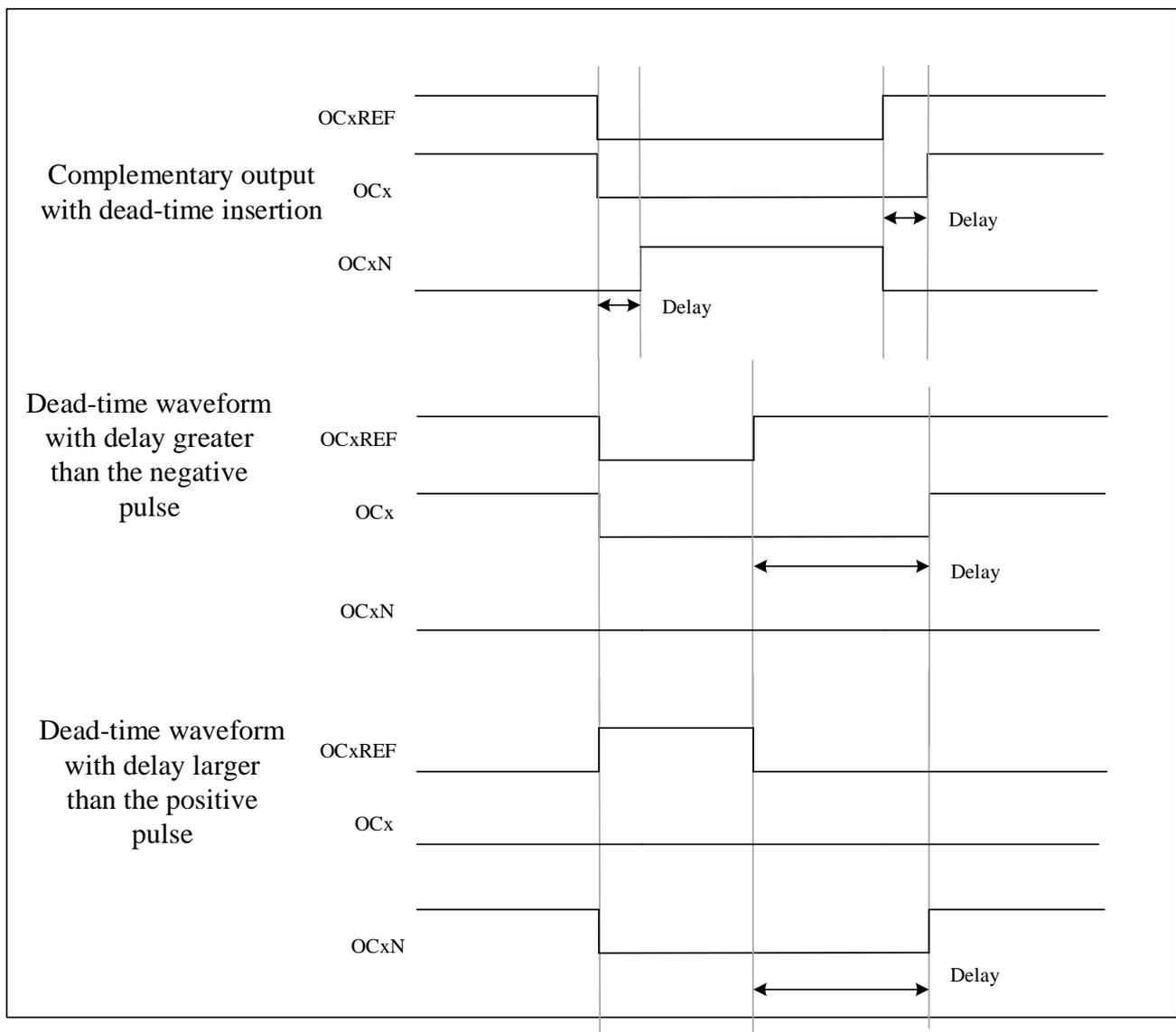
如果用户同时设置 TIMx\_CCEN.CCxEN 和 TIMx\_CCEN.CCxNEN，则会插入死区时间。如果有刹车，还要设置 TIMx\_BKDT.MOEN。每个通道都有 10 位死区时间发生器。

参考波形 OCxREF 可以生成 2 个输出 OCx 和 OCxN。如果 OCx 和 OCxN 为高电平有效，则 OCx 输出信号与参考信号相同，而 OCxN 输出信号与参考信号相反。但是，OCx 输出信号将相对于参考上升沿延迟，而 OCxN 输出信号将相对于参考下降沿延迟。如果延迟大于有效 OCx 或 OCxN 输出的宽度，则不会产生相应的脉冲。

死区时间发生器的输出信号与参考信号 OCxREF 之间的关系如下。

假设 TIMx\_CCEN.CCxP=0，TIMx\_CCEN.CCxNP=0，TIMx\_BKDT.MOEN=1，TIMx\_CCEN.CCxEN=1，TIMx\_CCEN.CCxNEN=1。

图 12-26 带死区插入的互补输出



用户可以设置 TIMx\_BKDT.DTGN 来编程每个通道的死区时间延迟。

### 12.3.13.1 重定向 OCxREF 到 OCx 或 OCxN

在输出模式下，用户可以设置 TIMx\_CCEN.CCxEN 和 TIMx\_CCEN.CCxNEN 以将 OCxREF 重定向到

OCx 输出或 OCxN 输出。

这里有两种使用这个功能的方法。当互补保持在其无效电平时，用户可以使用此功能发送特定波形，例如 PWM 或静态有效电平。用户还可以使用此功能将两个输出设置为无效电平，或将两个输出都设置为有效，两者互补且带死区。

如果用户设置 TIMx\_CCEN.CCxEN=0 和 TIMx\_CCEN.CCxNEN=1，两者不互补，当 OCxREF 为高电平时 OCxN 将变为有效。另一方面，如果用户设置 TIMx\_CCEN.CCxEN=1 和 TIMx\_CCEN.CCxNEN=1，当 OCxREF 为高电平时，OCx 将变为有效。相反，当 OCxREF 为低电平时，OCxN 将变为有效。

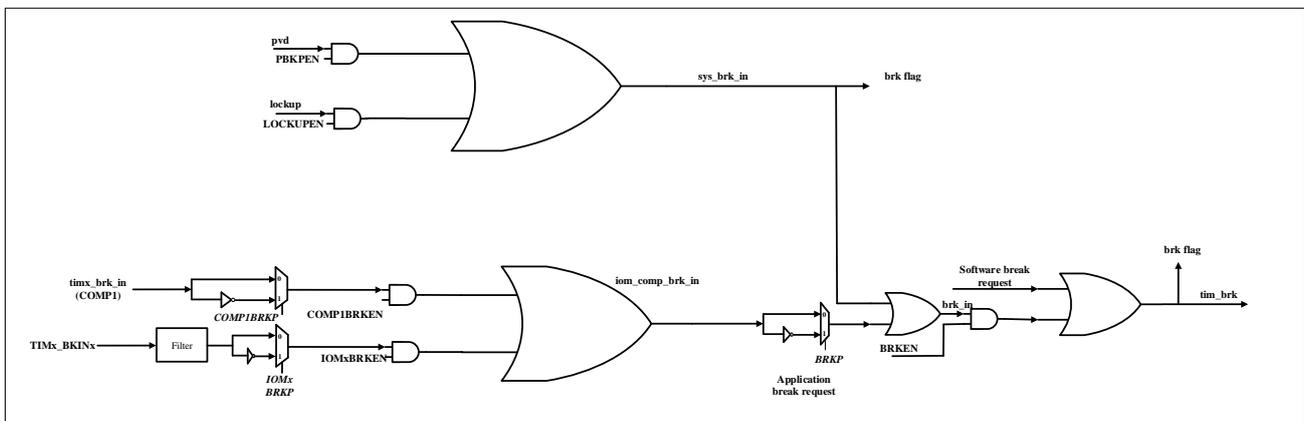
### 12.3.14 刹车功能

使用刹车功能时，设置相应的控制位时会修改输出使能信号和无效电平。但是，无论何时，OCx 和 OCxN 的输出都不能同时处于有效电平，即需要满足  $(CCxP \wedge OIx) \wedge (CCxNP \wedge OIxN) \neq 0$ 。TIM1 刹车功能仅对 OC1~OC4 及其互补通道有效，对 OC5~OC7 无效。

当启用多个刹车信号时，每个刹车信号构成一个 OR 逻辑。这里有一些信号可能是刹车的来源。

- 刹车输入引脚（4 个 IO 引脚）
- PVD 事件。
- 内核 Hardfault 事件。
- 比较器的输出信号。
- 软件设置 TIMx\_EVTGEN.BGN。

图 12-27 刹车输入



复位后刹车电路将被禁用。MOEN 位将为低电平。用户可以设置 TIMx\_BKDT.BKEN 来启用刹车功能。通过设置 TIMx\_BKDT.BKP 可以选择刹车输入信号的极性。用户可以同时修改 TIMx\_BKDT.BKEN 和 TIMx\_BKDT.BKP。用户设置 TIMx\_BKDT.BKEN 和 TIMx\_BKDT.BKP 后，生效前有 1 个 APB 时钟周期延迟。因此，用户需要等待 1 个 APB 时钟周期才能读回写入位的值。

MOEN 的下降沿可以是异步的，所以在实际信号和同步控制位之间设置了一个再同步电路。该电路将导致异步和同步信号之间的延迟。当用户设置 TIMx\_BKDT.MOEN 为低电平时，用户需要在读取该值之前插入一个延迟。因为写入了异步信号，但用户读取了同步信号。

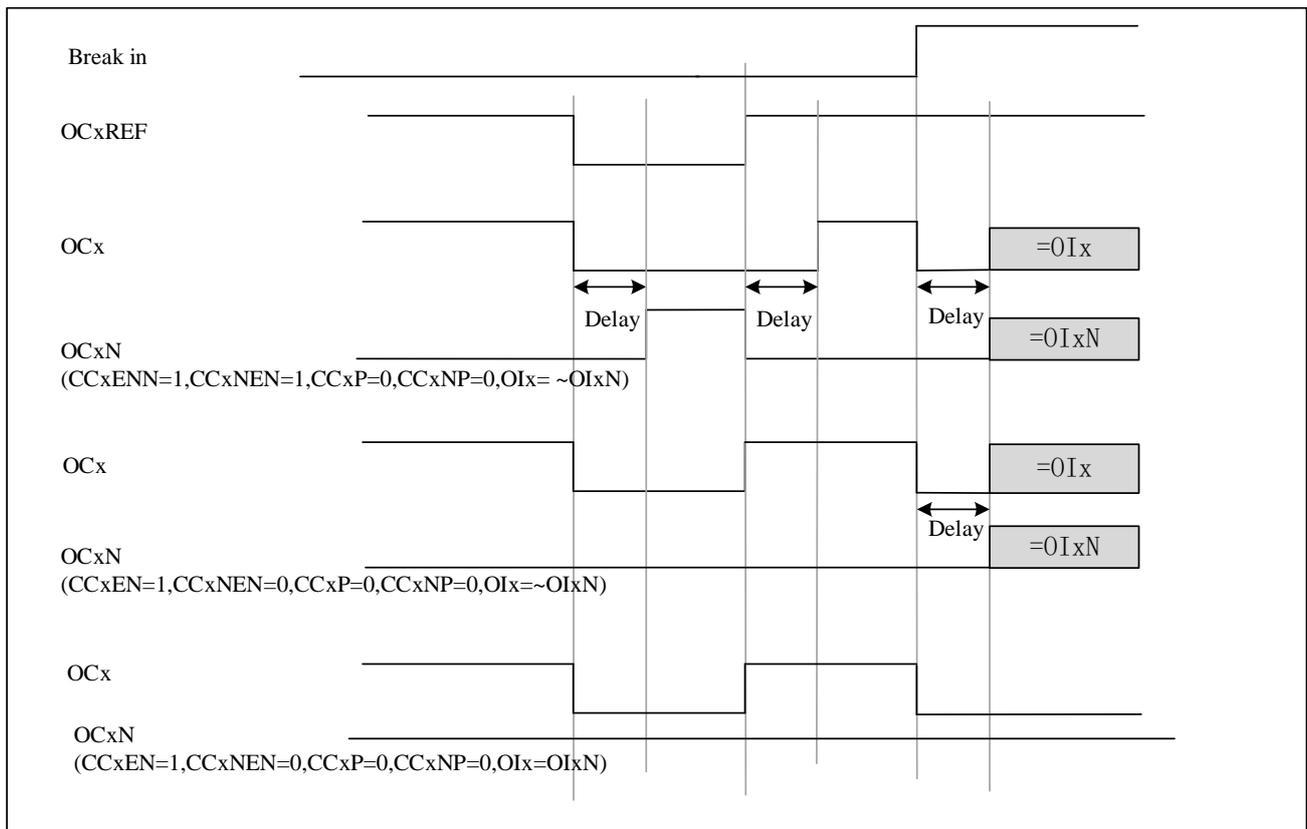
刹车发生后的行为如下：

- `TIMx_BKDT.MOEN` 将被异步清除，然后输出将进入无效状态、空闲状态或复位状态。通过设置 `TIMx_BKDT.OSSI` 选择输出状态。即使 MCU 振荡器关闭，这也会生效。
- 一旦 `TIMx_BKDT.MOEN=0`，每个输出通道的输出将使用 `TIMx_CTRL2.OIx` 中编程的电平驱动。如果 `TIMx_BKDT.OSSI=0`，定时器将释放使能输出（由 GPIO 控制器接管），否则将保持高电平。
- 如果用户选择使用互补输出，TIM 的行为如下
  - 取决于极性，输出将首先设置为复位状态。它是一个异步选项，因此即使没有为计时器提供时钟，它仍然可以工作。
  - 如果仍然提供定时器时钟，死区发生器将重新激活，当  $(CCxP \wedge OIx) \wedge (CCxNP \wedge OIxN) \neq 0$ ，即 `OCx` 和 `OCxN` 仍然不能同时被驱动到有效电平，在死区时间后根据 `TIMx_CTRL2.OIx` 和 `TIMx_CTRL2.OIxN` 的值驱动输出。请注意，由于 `MOEN` 上的重新同步（大概 2 个 `ck_tim` 周期），死区时间将比平时长。
  - 如果 `TIMx_BKDT.OSSI=0`，定时器将释放输出控制。否则，如果使能输出为高电平，它将保持为高电平。如果为低电平，则在 `TIMx_CCEN.CCxEN` 或 `TIMx_CCEN.CCxNEN` 为高电平时变为高电平。
- 如果 `TIMx_DINTEN.BIEN=1`，当 `TIMx_STS.BITF=1` 时，会产生中断。
- 如果用户设置了 `TIMx_BKDT.AOEN`，`TIMx_BKDT.MOEN` 将在下一次 UEV 发生时自动设置。用户可以使用它来调节。如果用户未设置 `TIMx_BKDT.AOEN`，则 `TIMx_BKDT.MOEN` 将保持低电平，直到再次设置为 1。在这种情况下，用户可以使用它来保证安全。用户可以将刹车输入连接到热传感器、电源驱动器警报或其他安全组件。
- 刹车输入有效时，`TIMx_BKDT.MOEN` 不能自动置位或软件同时置位，`TIMx_STS.BITF` 也不能清零。因为刹车输入在电平上处于有效状态。

为保证应用安全，刹车电路具有写保护功能，并有刹车输入输出管理。它允许用户冻结一些参数，例如死区持续时间、`OCx/OCxN` 极性和禁用时的状态、`OCxMD` 配置、刹车启用和极性。用户可以通过设置 `TIMx_BKDT.LCKCFG` 选择使用 3 种保护级别之一。但是，`TIMx_BKDT.LCKCFG` 只能在 MCU 复位后写入一次。

响应刹车的输出行为示例如下

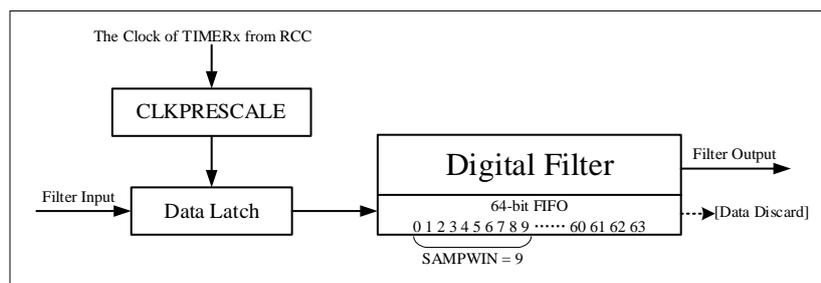
图 12-28 响应刹车的输出行为



### 12.3.14.1 刹车滤波

寄存器 TIM1\_BKFR 描述如下：

图 12-29 滑动滤波



- 数字滤波器通过 RCC 的 TIM1 时钟采样刹车信号，在 64 位 FIFO 中累积采样。仅在 TIM1\_BKFR.WSIZE [5:0] 中定义的窗口大小内采样数据，最大大小为 64。
- 过滤器输出采样窗口内的多数值，该值由 TIM1\_BKFR.THRESH [5:0] 中的阈值定义，最大阈值为 63。此值应等于或大于窗口大小的一半。如果采样窗口内的逻辑 1 和逻辑 0 计数均不大于阈值，则数字滤波器保持先前的输出值。
- RCC\_TIMFILTCFG.TIM1FILTCLK[4:0] 寄存器决定相应数字滤波器的采样率。过滤器 FIFO 在每个采样时钟从输入中捕获一个采样值。
- 如果数字滤波器关闭，滤波器输入将像电线一样绕过输出。

### 12.3.15 调试模式

当微控制器处于调试模式（Cortex-M0 内核停止）时，根据 DBG\_CTRL.TIMx\_STOP 配置，TIMx 计数器可以继续正常工作或停止。详见 3.3.1。

### 12.3.16 TIMx 定时器和外部触发的同步

TIMx 定时器可以通过从模式（复位、触发和门控）中的触发器进行同步。

#### 12.3.16.1 从模式：复位模式

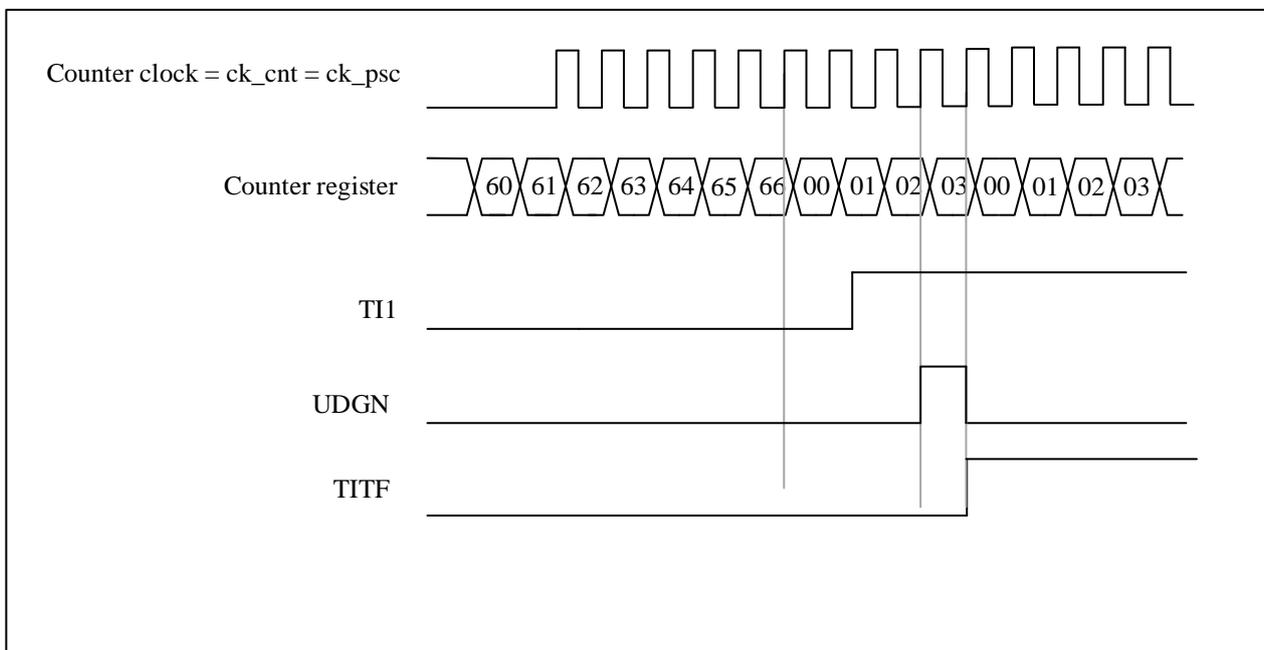
在复位模式下，触发事件可以复位计数器和预分频器。更新预加载寄存器 TIMx\_AR、TIMx\_CCDATx，并产生更新事件 UEV（TIMx\_CTRL1.UPRS=0）。

以下是复位模式的示例：

1. 通道 1 配置为输入检测 TI1 的上升沿（TIMx\_CCMOD1.CC1SEL=01，TIMx\_CCEN.CC1P=0）；
2. 从模式选择为复位模式（TIMx\_SMCTRL.SMSEL=0100），触发输入选择为 TI1（TIMx\_SMCTRL.TSEL=101）；
3. 启动计数器（TIMx\_CTRL1.CNTEN = 1）

启动定时器后，当 TI1 检测到上升沿时，计数器复位并重新开始计数，并设置触发标志（TIMx\_STS.TITF=1）；TI1 的上升沿与实际计数器复位之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 12-30 复位模式下的控制电路



#### 12.3.16.2 从模式：触发模式

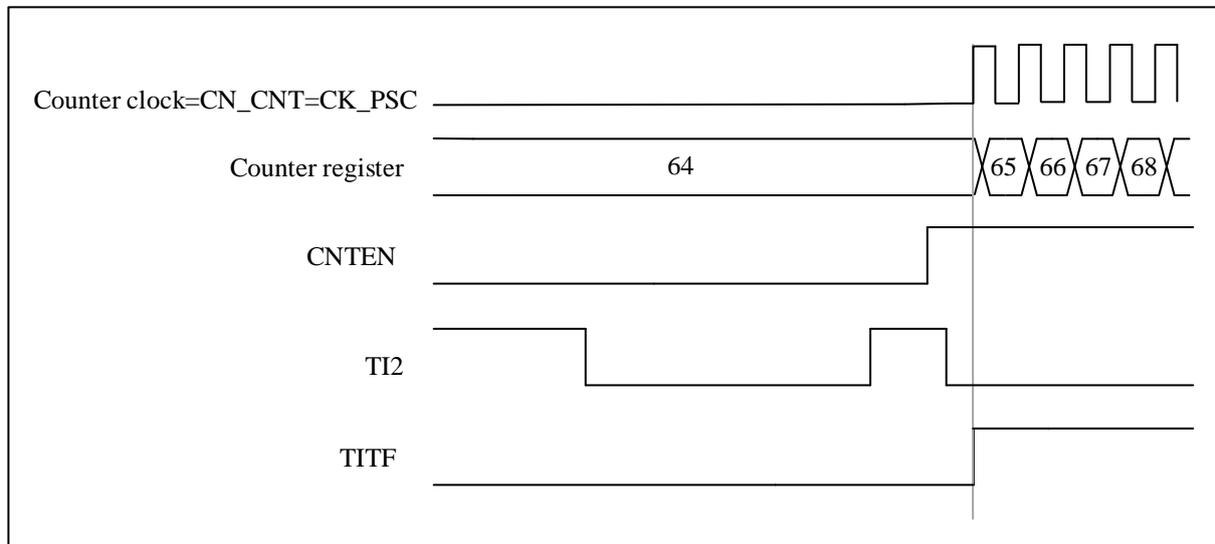
在触发模式下，输入端口的触发事件（上升沿/下降沿）可以触发计数器开始计数。

以下是触发模式的示例：

1. 通道 2 配置为输入，检测 TI2 的上升沿（TIMx\_CCMOD1.CC2SEL=01，TIMx\_CCEN.CC2P=0）；

- 选择从模式为触发模式( $TIMx\_SMCTRL.SMSEL=0110$ ), 触发输入选择 TI2( $TIMx\_SMCTRL.TSEL=110$ ); 当 TI2 检测到上升沿时, 计数器开始计数, 触发标志置位 ( $TIMx\_STS.TITF=1$ );
- TI2 的上升沿与实际计数器启动之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 12-31 触发器模式下的控制电路



### 12.3.16.3 从模式：门控模式

在门控模式下, 输入端口的电平极性可以控制计数器是否计数。

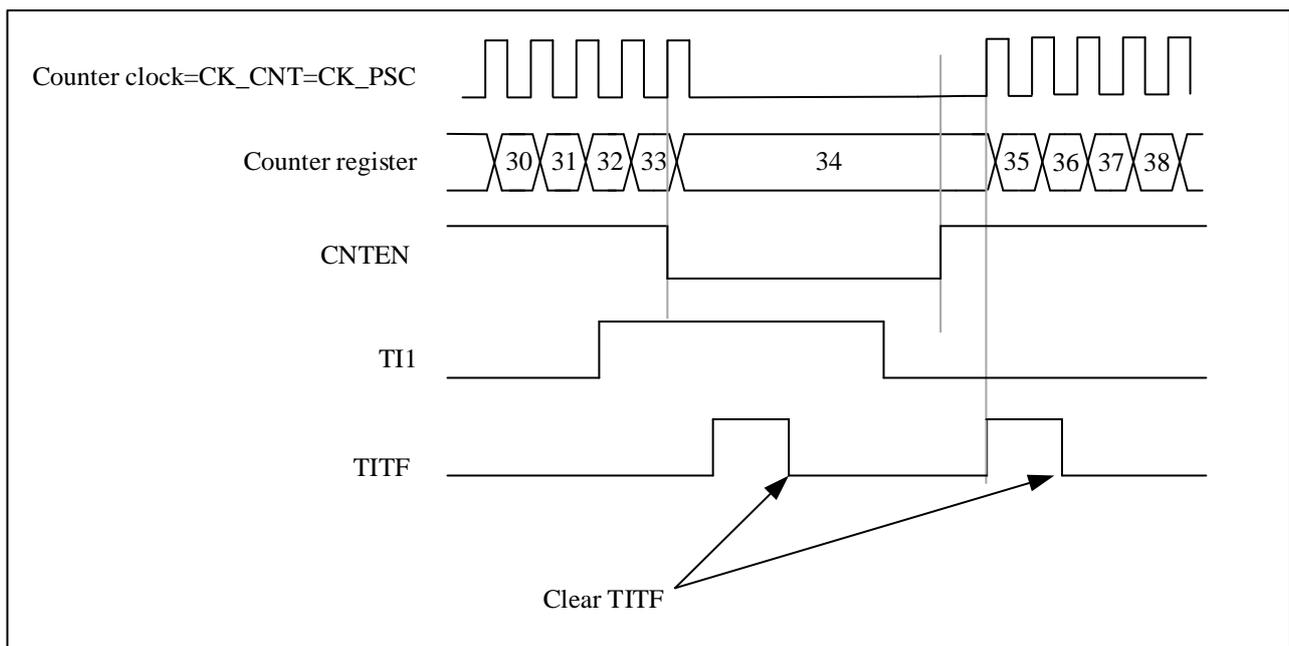
以下是门控模式的示例:

- 通道 1 配置为 TI1 上的输入检测低电平有效 ( $TIMx\_CCMOD1.CC1SEL=01$ ,  $TIMx\_CCEN.CC1P=1$ );
- 选择从模式为门控模式 ( $TIMx\_SMCTRL.SMSEL=0101$ ), 选择 TI1 作为 TRGI ( $TIMx\_SMCTRL.TSEL=101$ );
- 启动计数器 ( $TIMx\_CTRL1.CNTEN = 1$ );

当 TI1 检测到电平由低变高时, 计数器停止计数, 当 TI1 检测到电平由高变低时, 计数器开始计数, 开始或停止计数时触发标志置位 ( $TIMx\_STS.TITF=1$ )。

TI1 的上升沿与实际计数器停止之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 12-32 门控模式下的控制电路



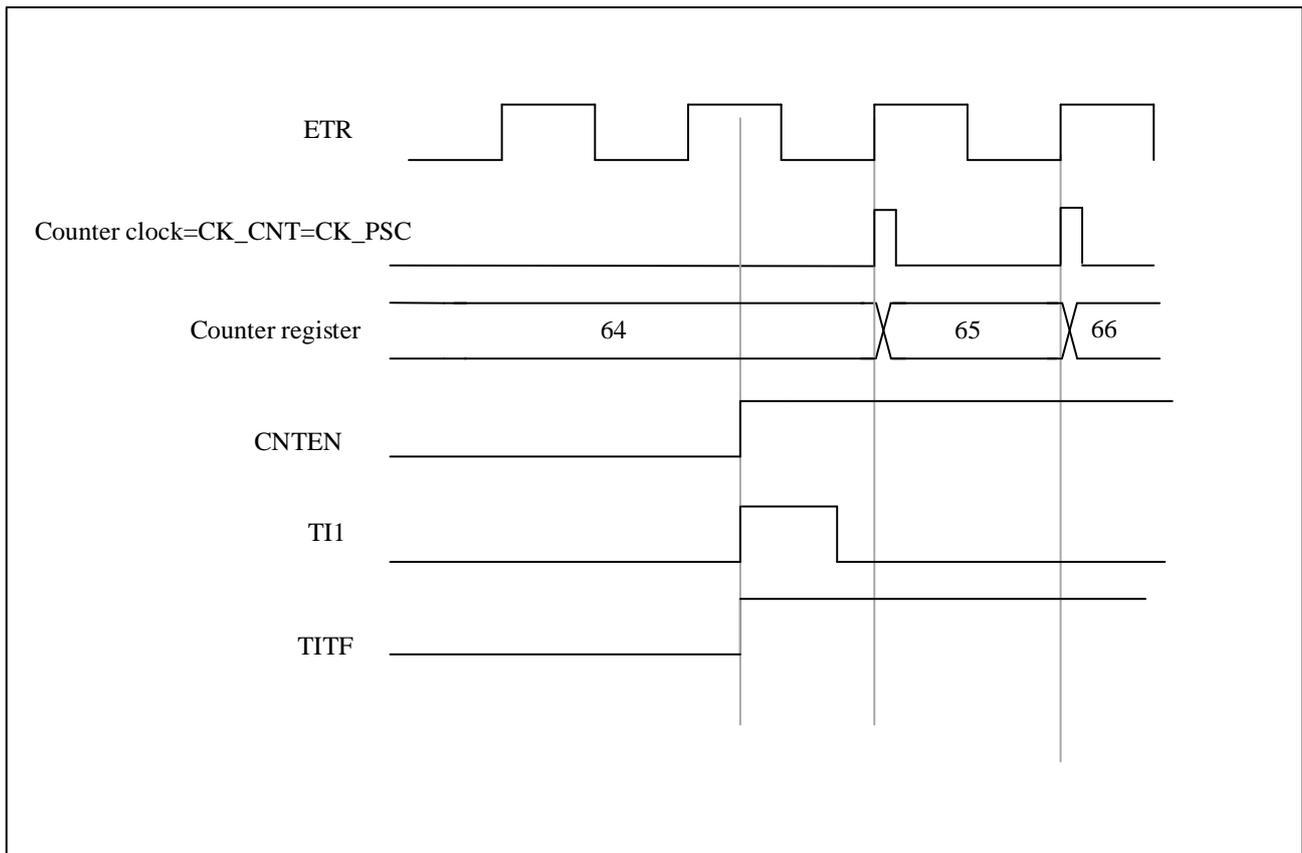
### 12.3.16.4 从模式：触发模式 +外部时钟模式 2

在复位模式、触发模式和门控模式下，计数器时钟可选择为外部时钟模式 2，ETR 信号作为外部时钟源输入。这时候触发选择需要选择非 ETRF (TIMx\_SMCTRL.TSEL=111)。

这是一个例子：

1. 通道 1 配置为输入检测 TI1 的上升沿 (TIMx\_CCMOD1.CC1SEL=01, TIMx\_CCEN.CC1P=0);
  2. 使能外部时钟模式 2(TIMx\_SMCTRL.EXCEN=1), 外部触发极性选择上升沿(TIMx\_SMCTRL.EXTP=0), 触发模式作为从模式 (TIMx\_SMCTRL.SMSEL=0110), TRGI 选择 TI1 (TIMx\_SMCTRL.TSEL=101);
- 当 TI1 检测到上升沿时，计数器在 ETR 的上升沿开始计数，并设置触发标志 (TIMx\_STS.TITF=1);

图 12-33 外部时钟模式 2+触发模式下的控制电路



### 12.3.16.5 从模式：组合复位+触发模式

在这种情况下，选定的触发器输入（trgi）的上升沿会重新初始化计数器，生成寄存器的更新，并启动计数器。

这种模式用于单脉冲模式。

输入端上选中的事件复位并使能计数器。

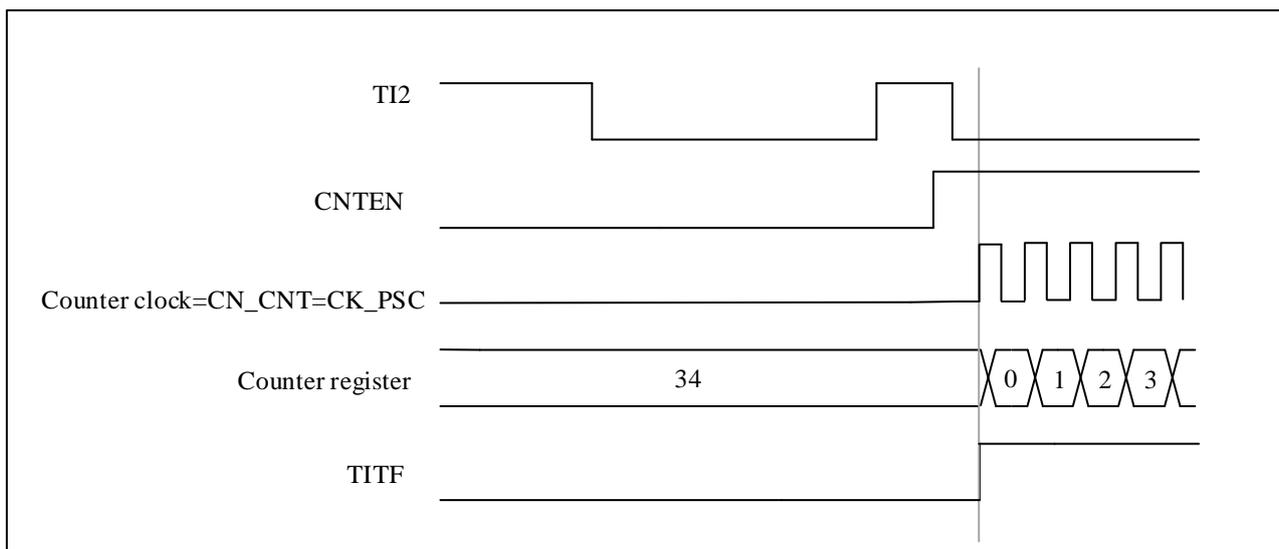
在下面的例子中，计数器在 TI2 输入的上升沿复位并开始向上计数：

- 配置通道 2 检测 TI2 的上升沿。配置输入滤波器带宽(本例中，不需要任何滤波器，保持 TIMx\_CCMOD1.IC2F=0000)。触发操作中不使用捕获预分频器，不需要配置。TIMx\_CCMOD1.CC2SEL 位只用于选择输入捕获源，置 TIMx\_CCMOD1.CC2SEL=01。置 TIMx\_CCEN.CC2P=1 以确定极性(只检测低电平)
- 置 TIMx\_SMCTRL.SMSEL=1110，配置定时器为组合复位+触发模式；置 TIMx\_SMCTRL.TSEL=110，选择 TI2 作为输入源。

当 TI2 出现一个上升沿时，计数器开始在内部时钟驱动下计数，同时设置 TITF 标志。

TI2 上升沿和计数器启动计数之间的延时，取决于 TI2 输入端的重同步电路。

图 12-34 组合复位+触发模式下的控制电路



### 12.3.16.6 从模式：组合门控+复位模式

当触发器输入（trgi）为高电平时，计数器时钟被启用。一旦触发器变为低电平，计数器就会停止，并被复位）。计数器的启动和停止都受到控制。

这种模式可以检测出超范围的 PWM 信号（占空比超过最大预期值）。

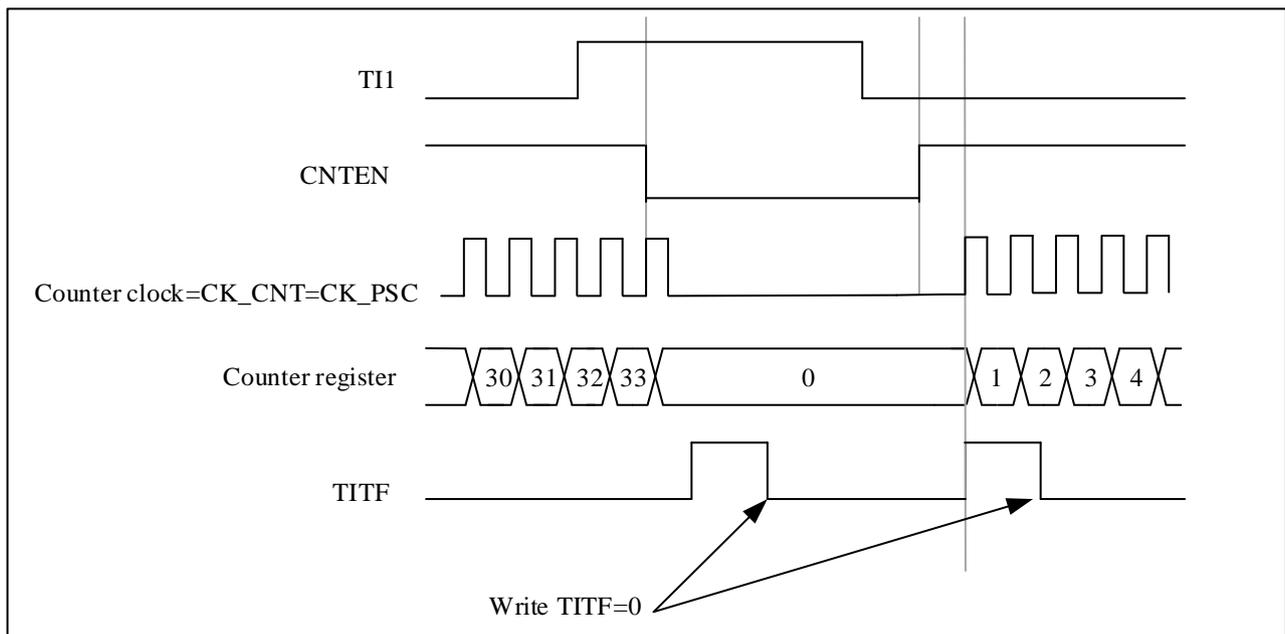
在如下的例子中，计数器只在 TI1 为低时向上计数，在 TI1 变为高时计数器停止并复位：

- 配置通道 1 以检测 TI1 上的低电平。配置输入滤波器带宽(本例中，不需要滤波，所以保持 TIMx\_CCMOD1.IC1F=0000)。触发操作中不使用捕获预分频器，所以不需要配置。TIMx\_CCMOD1.CC1SEL 位用于选择输入捕获源，置 TIMx\_CCMOD1.CC1SEL=01。置 TIMx\_CCEN.CC1P=1 以确定极性(只检测低电平)。
- 置 TIMx\_SMCTRL.SMSEL=1101，配置定时器为门控+复位模式；置 TIMx\_SMCTRL.TSEL=101，选择 TI1 作为输入源。
- 置 TIMx\_CTRL1 寄存器中 CNTEN=1，启动计数器。在门控+复位模式下，如果 CNTEN=0，则计数器不能启动，不论触发输入电平如何。

只要 TI1 为低，计数器开始依据内部时钟计数，一旦 TI1 变高则停止计数。当计数器开始或停止时都设置 TIMx\_STS 中的 TITF 标置。

TI1 上升沿和计数器实际停止之间的延时取决于 TI1 输入端的重同步电路。

图 12-35 组合门控+复位模式下的控制电路



### 12.3.17 定时器同步

所有 TIM 定时器在内部相连，用于定时器同步或链接。详见 13.3.16 章节。

### 12.3.18 触发 ADC

定时器可通过多种内部信号产生 ADC 触发事件，例如复位、使能或比较事件，也可生成由内部边沿检测器发出的脉冲触发。在重定向到 ADC 的 TRGO 内部线路上发出触发信号可通过 TIMx\_CTRL2 寄存器中的 MMSEL[3:0] 位选择。也可以通过通道 1/2/3/4/5/6/7 触发 ADC。

### 12.3.19 产生六步 PWM 输出

为了同时修改所有通道的配置，可以提前设置下一步的配置（预加载位为 OCxMD、CCxEN 和 CCxNEN）。当发生 COM 换相事件时，OCxMD、CCxEN 和 CCxNEN 预加载位被传送到影子寄存器位。

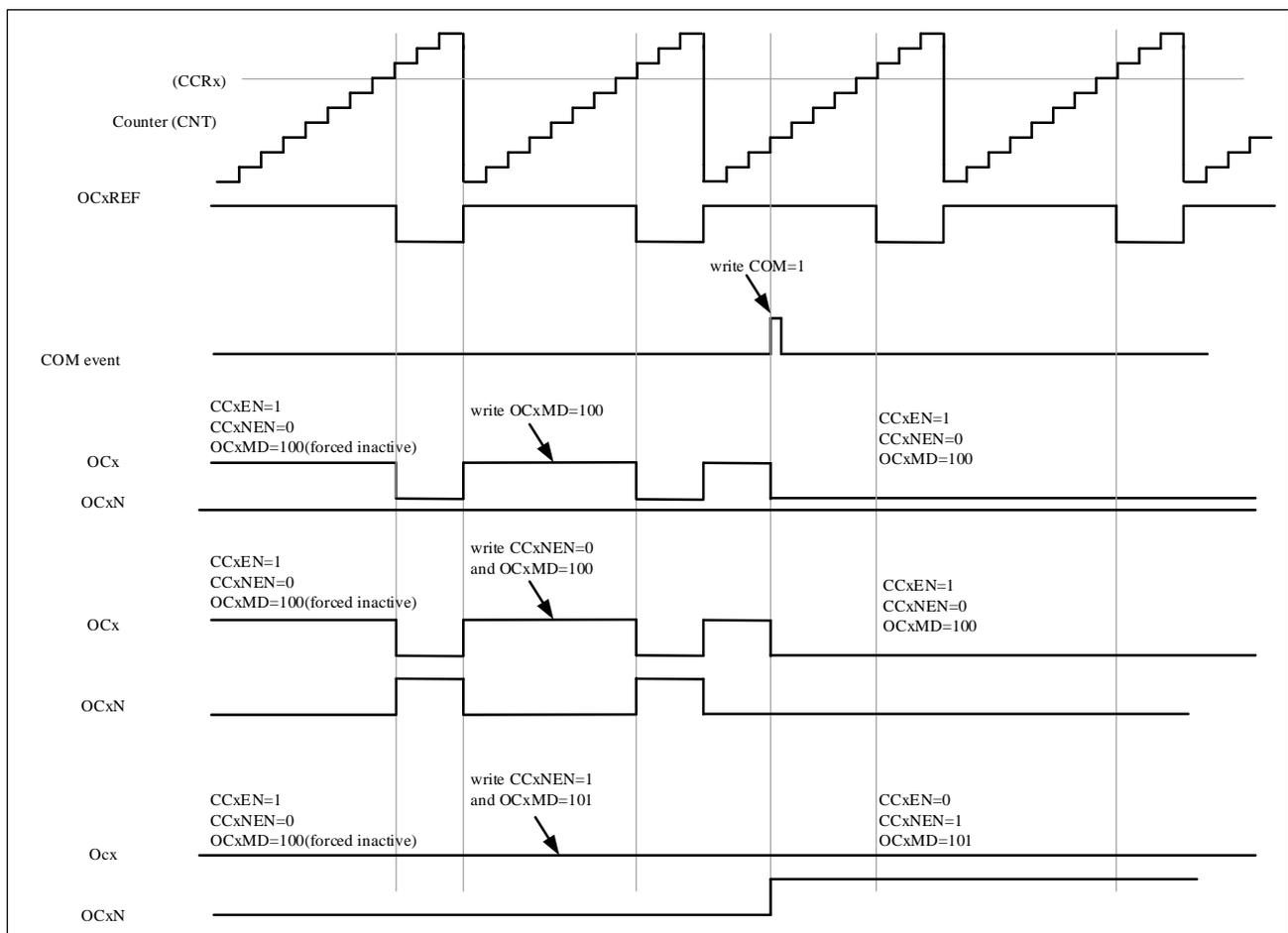
COM 换相事件生成方法：

1. 软件设置 TIMx\_EVTGEN.CCUDGN；
2. 在 TRGI 的上升沿由硬件产生；

当 COM 换相事件发生时，TIMx\_STS.COMITF 标志将被设置，启用中断 (TIMx\_DINTEN.COMIEN) 将产生中断，启用 DMA 请求 (TIMx\_DINTEN.COMDEN) 将产生 DMA 请求。

下图显示了三种不同配置下发生 COM 换向事件时 OCx 和 OCxN 的输出时序图：

图 12-36 产生六步 PWM，使用 COM 的例子 (OSSR=1)



### 12.3.20 编码器接口模式

编码器使用两个输入 TI1 和 TI2 作为接口，计数器对 TI1FP1 或 TI2FP2 上的每个边沿变化进行计数。计数方向由硬件 TIMx\_CTRL1.DIR 自动控制。编码器计数模式共有三种：

1. 计数器只在 TI1 的边沿计数，TIMx\_SMCTRL.SMSEL = '0001'；
2. 计数器只在 TI2 的边沿计数，TIMx\_SMCTRL.SMSEL = '0010'；
3. 计数器同时在 TI1 和 TI2 的边沿计数，TIMx\_SMCTRL.SMSEL = '0011'；

编码器接口相当于使用带方向选择的外部时钟，计数器只在 0 和自动重载值(TIMx\_AR.AR [15:0])之间连续计数。因此，需要提前配置自动重载寄存器 TIMx\_AR。

*注意：编码器模式和外部时钟模式 2 不兼容，不能同时选择。*

计数方向与编码器信号的关系如下表：

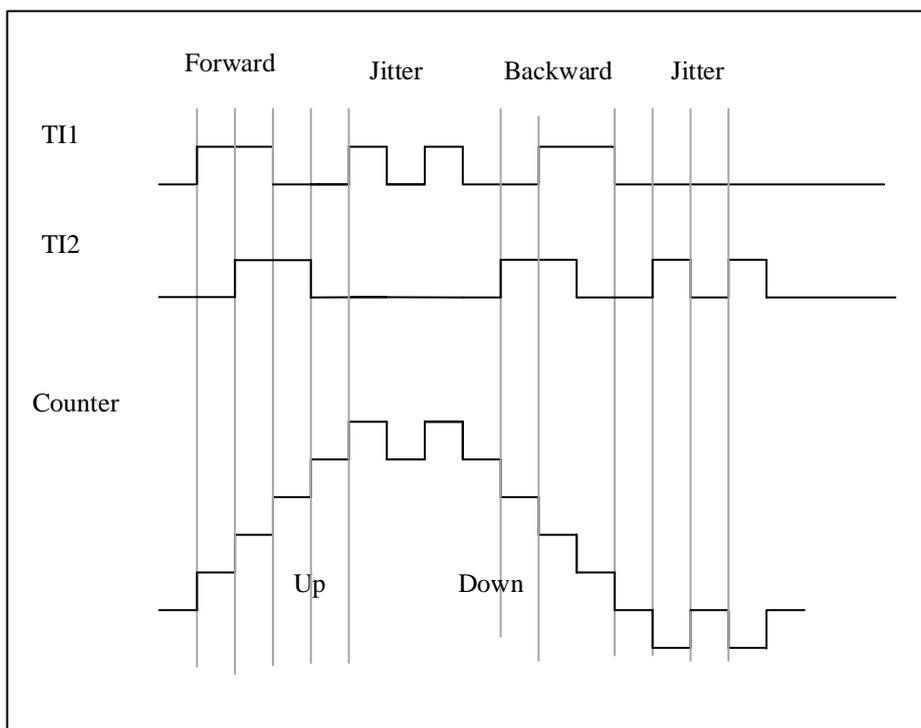
表 12-1 计数方向与编码器信号的关系

有效边沿	相对信号的电平 (TI1FP1对应TI2, TI2FP2对应TI1)	TI1FP1信号		TI2FP2信号	
		上升	下降	上升	下降
仅在TI1计数	高	向下计数	向上计数	不计数	不计数
	低	向上计数	向下计数	不计数	不计数
仅在TI2计数	高	不计数	不计数	向上计数	向下计数
	低	不计数	不计数	向下计数	向上计数
在TI1和TI2上计数	高	向下计数	向上计数	向上计数	向下计数
	低	向上计数	向下计数	向下计数	向上计数

以下是选择了双边沿触发以抑制输入抖动的编码器示例：

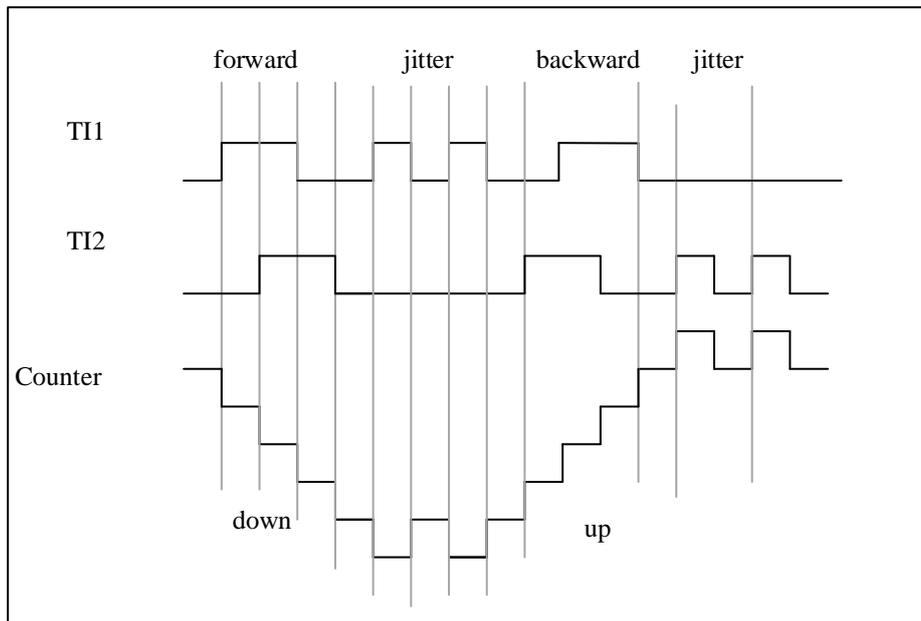
1. IC1FP1 映射到 TI1 (TIMx\_CCMOD1.CC1SEL = '01')，IC1FP1 不反相 (TIMx\_CCEN.CC1P = '0')；
2. IC1FP2 映射到 TI2 (TIMx\_CCMOD2.CC2SEL = '01')，IC2FP2 不反相 (TIMx\_CCEN.CC2P = '0')；
3. 输入在上升沿和下降沿均有效 (TIMx\_SMCTRL.SMSEL = '0011')；
4. 启用计数器 TIMx\_CTRL1.CNTEN = '1'；

图 12-37 编码器模式下的计数器操作实例



下图为 IC1FP1 极性反转时的计数器行为示例 (CC1P = '1', 其他配置同上)

图 12-38 IC1FP1 反相的编码器接口模式实例



### 12.3.21 与霍尔传感器的接口

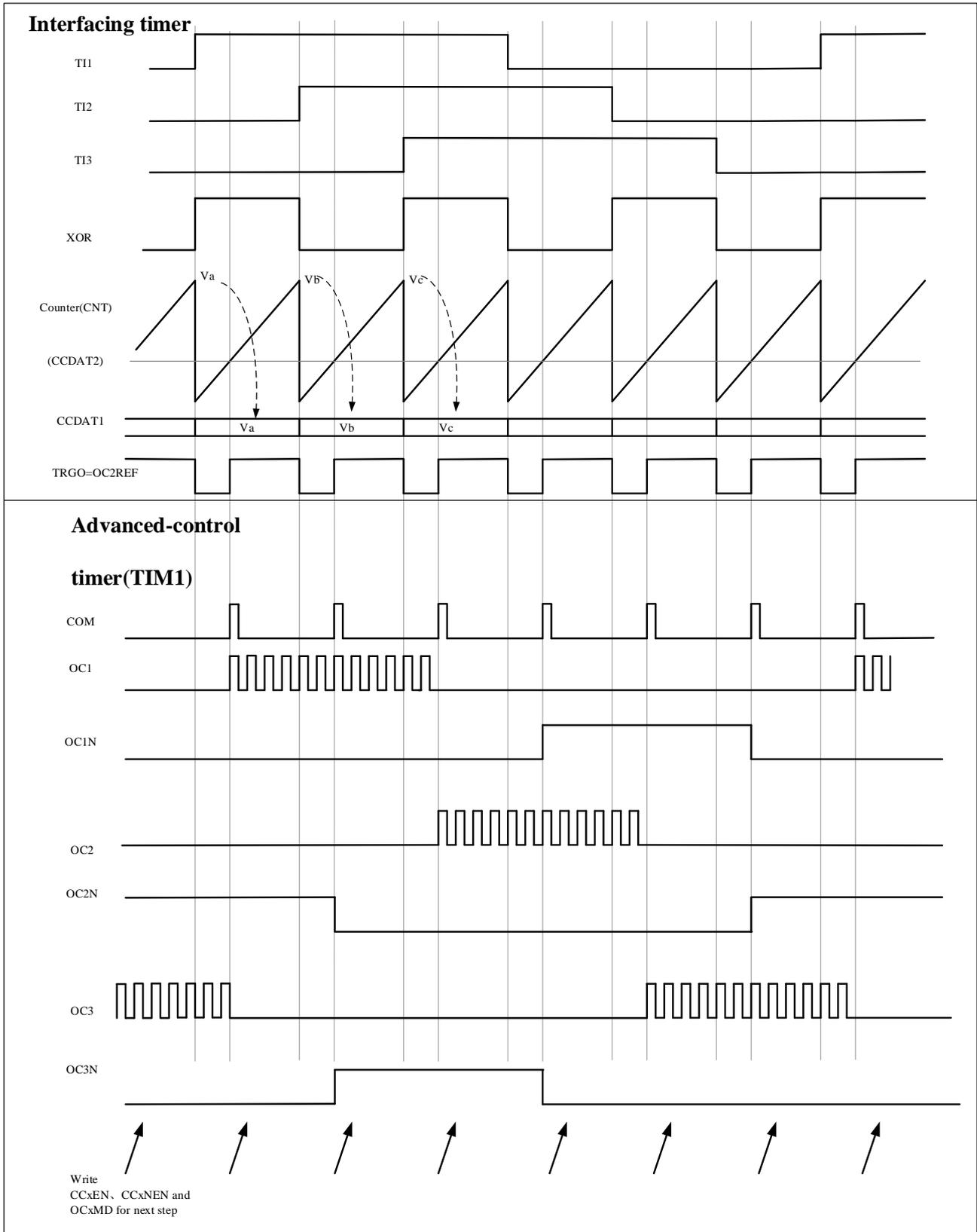
将霍尔传感器连接到定时器的三个输入引脚 (CC1、CC2 和 CC3)，然后选择异或功能将 TIMx\_CH1、TIMx\_CH2 和 TIMx\_CH3 的输入通过异或门作为 TI1 的输出到通道 1 进行捕捉信号。

定时器需要配置为从模式下的复位模式 (TIMx\_SMCTRL.SMSEL='0100')；触发选择 TI1 的边沿触发 TI1F\_ED (TIMx\_SMCTRL.TSEL='100')，霍尔 3 输入的任何变化都会触发计数器重新计数，因此用作时间参考；捕获/比较通道 1 配置为捕获模式下的 TRC 信号 (TIMx\_CCMOD1.CC1SEL='11')，用于计算两个输入时间间隔，从而反映电机速度。

选择定时器通道 2 向高级定时器输出脉冲，触发高级定时器的 COM 事件，更新输出 PWM 的控制位。高级定时器的触发选择需要选择对应的内部触发信号 (TIMx\_SMCTRL.TSEL="ITRx")，捕获/比较预加载控制位需要配置为支持预加载 (TIMx\_CTRL2.CCPCTL=1) 并支持上升沿 TRGI 边沿触发更新 (TIMx\_CTRL2.CCUSEL=1)。

此示例如下图所示。

图 12-39 霍尔传感器接口的实例



## 12.4 TIM1 寄存器描述

关于在寄存器描述里面所用到的缩写，详见 1.1 节。

可以用半字（16 位）或字（32 位）的方式操作这些外设寄存器。

### 12.4.1 寄存器总览

表 12-2 TIM1 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
000h	TIM_CTRL1	Reserve								ASYMMETRIC	Reserve								C1SEL	Reserve	CLRSEL	Reserve	PBKPEN	LBKPEN	ARPEN	ONEPM	CLKD[1:0]	UPDIS	UPRS	CAMSEL[1:0]			DIR	CNTEN				
004h	TIM_CTRL2	Reserve												T1SEL	CCPCTL	CCDSEL	CCUSEL	MMSEL[3:0]			Reserve				O14N	O14	O1EN	O13	O12N	O12	O11N	O11						
008h	TIM_STS	Reserve	IOMB1TF	Reserve	CC7ITF	COMPBITF	PBKPT1F	LBKPT1F	Reserve	BITF	T1TF	COM1TF	UD1TF	Reserve								CC4OCF	CC3OCF	CC2OCF	CC1OCF	Reserve	CC6ITF	CC5ITF	CC4ITF	CC3ITF	CC2ITF	CC1ITF						
00Ch	TIM_EVTGEN	Reserve																		BGN	TGN	CCUDGN	UDGN	Reserve											CC4GN	CC3GN	CC2GN	CC1GN
010h	TIM_SMCTRL	Reserve														MSMD	EXTF[3:0]			EXTP	EXCEN	EXTPS[1:0]			SMSEL[3:0]			Reserve	TSEL[2:0]									
014h	TIM_DINTEN	Reserve								COM1EN	TDEN	COMDEN	UDEN	BIEN	T1EN	U1EN	Reserve								CC4DEN	CC3DEN	CC2DEN	CC1DEN	Reserve	CC71EN	CC61EN	CC51EN	CC41EN	CC31EN	CC21EN	CC11EN		
018h	TIM_CCMOD1_OUT	Reserve																		OC2MD[2:0]		OC2CEN	OC2FEN	OC2PEN	CC2SEL[1:0]			OC1MD[2:0]			OC1CEN	OC2FEN	OC1PEN	CC1SEL[1:0]				
	TIM_CCMOD1_IN	Reserve																		IC2F[3:0]			IC2PSC[1:0]			CC2SEL[1:0]			IC1F[3:0]			IC1PSC[1:0]			CC1SEL[1:0]			
01Ch	TIM_CCMOD2_OUT	Reserve																		OC4MD[2:0]		OC4CEN	OC4FEN	OC4PEN	CC4SEL[1:0]			OC3MD[2:0]			OC3CEN	OC3FEN	OC3PEN	CC3SEL[1:0]				
	TIM_CCMOD2_IN	Reserve																		IC4F[3:0]			IC4PSC[1:0]			CC4SEL[1:0]			IC3F[3:0]			IC3PSC[1:0]			CC3SEL[1:0]			
020h	TIM_CCMOD3_OUT	Reserve	OC7MD[2:0]			Reserve								OC7CEN	OC7FEN	OC7PEN	OC6MD[2:0]			OC6CEN	OC6FEN	OC6PEN	Reserve	OC5MD[2:0]			OC5CEN	OC5FEN	OC5PEN	Reserve								
024h	TIM_CCEN	Reserve								CC7P	CC7EN	CC6P	CC6EN	Reserve	CC5P	CC5EN	Reserve	CC4P	CC4EN	CC4NP	CC4NEN	CC3P	CC3EN	CC3NP	CC3NEN	CC2P	CC2EN	CC2NP	CC2NEN	CC1P	CC1EN	CC1NP	CC1NEN					

028h	TIM_CC DAT1	CCDDAT1[15:0]				CCDAT1[15:0]											
02Ch	TIM_CC DAT2	CCDDAT2[15:0]				CCDAT2[15:0]											
030h	TIM_CC DAT3	CCDDAT3[15:0]				CCDAT3[15:0]											
034h	TIM_CC DAT4	CCDDAT4[15:0]				CCDAT4[15:0]											
038h	TIM_CC DAT5	Reserve				CCDAT5[15:0]											
03Ch	TIM_CC DAT6	Reserve				CCDAT6[15:0]											
040h	TIM_PSC	Reserve				PSC[15:0]											
044h	TIM_AR	Reserve				AR[15:0]											
048h	TIM_CNT	Reserve				CNT[15:0]											
04Ch	TIM_REPCNT	Reserve								REPCNT[7:0]							
050h	TIM_BKDT	Reserve				LCKCFG[1:0]	OSSR	OSSI	BKEN	BKP	AOEN	MOEN	DTGN[7:0]				
054h	TIM_CC DAT7	Reserve				CCDAT7[15:0]											
060h	TIM_BKFR	Reserve	THRESH[5:0]	Reserve	WSIZE[5:0]	FIL/TEN	Reserve										
07Ch	TIM_AF1	Reserve				IOM4BRKP	IOM3BRKP	IOM2BRKP	IOM4BRKEN	IOM3BRKEN	IOM2BRKEN	Reserve	COMP1BRKP	IOM1BRKP	Reserve	COMP1BRKEN	IOM1BRKEN
094h	TIM_DCTRL	Reserve						DBADDR[5:0]		Reserve	DBLEN[5:0]						
098h	TIM_DADDR	BURST[31:0]															

## 12.4.2 控制寄存器 1 (TIMx\_CTRL1)

偏移地址: 0x00

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								ASYMMETRIC	Reserved						C1SEL
								rw							rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		CLRSEL	Reserved	PBKPEN	LBKPEN	ARPEN	ONEPM	CLKD[1:0]		UPDIS	UPRS	CAMSEL[1:0]		DIR	CNTEN
		rw		rw	rw	rw	rw	rw		rw	rw	rw		rw	rw

位域	名称	描述
31:24	Reserved	保留, 必须保持复位值
23	ASYMMETRIC	中央对齐非对称模式使能 0: 禁用 1: 使能 (当TIMx_CTRL1.CAMSEL[1:0]非零时有效, 每个通道向上计数时与CCDATx比较, 向下计数时与CCDDATx比较) 注意: 此功能仅适用于 TIM1
22:17	Reserved	保留, 必须保持复位值
16	C1SEL	通道1选择 (Channel 1 selection) 0: 选择外部CH1 (来自IOM) 信号 1: 选择内部CH1 (来自COMP) 信号
15: 14	Reserved	保留, 必须保持复位值
13	CLRSEL	OCxRef选择 (OCxRef selection) 0: 选择外部Ocxclr (ETR) 信号 1: 选择内部Ocxclr (来自COMP) 信号
12	Reserved	保留, 必须保持复位值
11	PBKPEN	PVD作为BRK启用 (PVD as brk Enable) 0: 禁止 1: 使能
10	LBKPEN	锁存作为BRK使能 (LockUp as brk Enable) (Core Hardfault) 0: 禁止 1: 使能
9	ARPEN	自动重装载预装载允许位 (Auto-reload preload enable) 0: TIMx_AR 寄存器的影子寄存器禁用 1: TIMx_AR 寄存器的影子寄存器使能

位域	名称	描述
8	ONEPM	单脉冲模式 (One pulse mode) 0: 禁用单脉冲模式, 发生更新事件时不影响计数器计数。 1: 使能单脉冲模式, 下次更新事件发生时计数器停止计数
7:6	CLKD[1:0]	时钟分频因子 (Clock division) CLKD[1:0] 表示 CK_INT (定时器时钟) 和 DTS (用于死区时间发生器和数字滤波器 (ETR、TIx) 的时钟) 之间的分频比。 00: $t_{DTS} = t_{CK\_INT}$ 01: $t_{DTS} = 2 \times t_{CK\_INT}$ 10: $t_{DTS} = 4 \times t_{CK\_INT}$ 11: 保留, 不要使用这个配置
5	UPDIS	更新禁用 (Update disable) 该位用于启用/禁用软件生成的更新事件 (UEV) 事件。 0: 启用。如果满足以下条件之一, 将生成 UEV: - 计数器上溢/下溢 - TIMx_EVTGEN.UDGN 位被设置 - 从模式控制器的更新生成 影子寄存器将使用预加载值进行更新。 1: UEV 禁用。不生成更新事件, 影子寄存器 (AR、PSC 和 CCDATx) 保持它们的值。如果 TIMx_EVTGEN.UDGN 位置位或从模式控制器发出硬件复位, 则重新初始化计数器和预分频器。
4	UPRS	更新请求源 (Update request source) 该位用于通过软件选择 UEV 事件源。 0: 如果更新中断或 DMA 请求使能, 以下任何事件都会产生更新中断或 DMA 请求: - 计数器上溢/下溢 - TIMx_EVTGEN.UDGN 位被设置 - 从模式控制器的更新生成 1: 如果更新中断或 DMA 请求使能, 只有计数器上溢/下溢会产生更新中断或 DMA 请求。
3:2	CAMSEL[1:0]	选择中央对齐模式 (Center-aligned mode selection) 00: 边缘对齐模式。TIMx_CTRL1.DIR 指定向上计数或向下计数。 01: 中央对齐模式1。计数器在中央对齐模式下计数, 向下计数时输出比较中断标志位设置为 1。 10: 中央对齐模式2。计数器在中央对齐模式下计数, 向上计数时输出比较中断标志位设置为1。 11: 中央对齐模式3。计数器在中央对齐模式下计数, 向上计数或向下计数时输出比较中断标志位设置为 1。 注意: 当计数器仍然启用时 (TIMx_CTRL1.CNTEN = 1), 不允许从边缘对齐模式切换到中央对齐模式。

位域	名称	描述
1	DIR	方向 (Direction) 0: 计数器向上计数; 1: 计数器向下计数。 <i>注: 当计数器配置为中央对齐模式或编码器模式时, 该位为只读。</i>
0	CNTEN	使能计数器 (Counter enable) 0: 禁止计数器; 1: 使能计数器。 <i>注: 在软件设置了CNTEN位后, 外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置CNTEN位。</i> <i>注: 软件设置了CNTEN位后, 需要等待至少两个TIMx_CLK, CNTEN才能从TIMx_PCLK同步到TIMx_CLK生效。</i>

### 12.4.3 控制寄存器 2 (TIMx\_CTRL2)

偏移地址: 0x04

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved												TI1SEL	CCPCTL	CCDSEL	CCUSEL	
												rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
MMSEL[3:0]			Reserved				OI4N	OI4	OI3N	OI3	OI2N	OI2	OI1N	OI1		
rw							rw	rw	rw	rw	rw	rw	rw	rw		

位域	名称	描述
31:20	Reserved	保留, 必须保持复位值
19	TI1SEL	TI1选择 (TI1 selection) 0: TIMx_CH1引脚连到TI1输入; 1: TIMx_CH1、TIMx_CH2和TIMx_CH3引脚经异或后连到TI1输入。
18	CCPCTL	捕获/比较预装载控制位 (Capture/compare preloaded control) 0: CCxEN, CCxNEN和OCxMD位不是预装载的; 1: CCxEN, CCxNEN和OCxMD位是预装载的; 设置该位后, 它们只在设置了CCUDGN位后被更新。 <i>注: 该位只对具有互补输出的通道起作用。</i>
17	CCDSEL	捕获/比较的DMA选择 (Capture/compare DMA selection) 0: 当发生CCx事件时, 送出CCx的DMA请求; 1: 当发生更新事件时, 送出CCx的DMA请求。
16	CCUSEL	捕获/比较控制更新选择 (Capture/compare control update selection) 0: 如果捕获/比较控制位是预装载的 (CCPCTL=1), 只能通过设置CCUDGN位更新它

位域	名称	描述
		<p>们:</p> <p>1: 如果捕获/比较控制位是预装载的 (CCPCTL =1), 可以通过设置CCUDGN位或TRGI上的一个上升沿更新它们。</p> <p><i>注: 该位只对具有互补输出的通道起作用。</i></p>
15:12	MMSEL[3:0]	<p>主模式选择</p> <p>这 4 位用于选择在主模式下发送到从定时器的同步信息 (TRGO)。可能的组合如下:</p> <p>0000: 复位 - 当 TIMx_EVTGEN.UDGN 置位或从模式控制器产生复位时, 将出现 TRGO 脉冲。在后一种情况下, TRGO 上的信号与实际复位相比有所延迟。</p> <p>0001: 使能 - TIMx_CTRL1.CNTEN 位用作触发输出 (TRGO)。有时需要同时启动多个定时器或者在一段时间内开启从定时器。</p> <p>当 TIMx_CTRL1.CNTEN 位置位或门控模式下的触发输入为高电平时, 计数器使能信号置位。</p> <p>当计数器使能信号由触发输入控制时, TRGO 上有一个延迟, 除非选择了主/从模式 (参见 TIMx_SMCTRL.MSMD 位的说明)。</p> <p>0010: 更新 - 选择更新事件作为触发输出 (TRGO)。例如, 主定时器时钟可用作从定时器预分频器。</p> <p>0011: 比较脉冲 - 当 TIMx_STS.CC1ITF 被设置时 (即使它已经是高电平), 即捕获或比较成功时, 触发输出发送一个正脉冲 (TRGO)。</p> <p>0100: 比较 - OC1REF 信号用作触发输出 (TRGO)。</p> <p>0101: 比较 - OC2REF 信号用作触发输出 (TRGO)。</p> <p>0110: 比较 - OC3REF 信号用作触发输出 (TRGO)。</p> <p>0111: 比较 - OC4REF 信号用作触发输出 (TRGO)。</p> <p>1000: 比较 - OC5REF 信号用作触发输出 (TRGO)。</p> <p>1001: 比较 - OC6REF 信号用作触发输出 (TRGO)。</p> <p>1010: 比较 - OC7REF 信号用作触发输出 (TRGO)。</p> <p>1011~1111: 保留。</p>
11:8	Reserved	保留, 必须保持复位值
7	OI4N	输出空闲状态4 (OC4N输出)。参见OI1N位。
6	OI4	输出空闲状态4 (OC4输出)。参见OI1位。
5	OI3N	输出空闲状态3 (OC3N输出)。参见OI1N位。
4	OI3	输出空闲状态3 (OC3输出)。参见OI1位。
3	OI2N	输出空闲状态2 (OC2N输出)。参见OI1N位。
2	OI2	输出空闲状态2 (OC2输出)。参见OI1位。
1	OI1N	<p>输出空闲状态1 (OC1N输出) (Output Idle state 1N)</p> <p>0: 当MOEN=0时, 死区后OC1N=0;</p> <p>1: 当MOEN=0时, 死区后OC1N=1。</p> <p><i>注: 已经设置了LCKCFG (TIMx_BKR 寄存器) 级别1、2或3后, 该位不能被修改。</i></p>
0	OI1	<p>输出空闲状态1 (OC1输出) (Output Idle state 1)</p> <p>0: 当MOEN=0时, 如果实现了OC1N, 则死区后OC1=0;</p>

位域	名称	描述
		1: 当MOEN=0时, 如果实现了OC1N, 则死区后OC1=1。 注: 已经设置了LCKCFG (TIMx_BKR寄存器) 级别1、2或3后, 该位不能被修改。

## 12.4.4 状态寄存器 (TIMx\_STS)

偏移地址: 0x08

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
Reserved				IOMBITF	Reserved			CC7ITF	COMP BITF	PBKPITF	LBKPITF	Reserved	BITF	TITF	COMITF	UDITF	
				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		
Reserved				CC4OCF	CC3OCF	CC2OCF	CC1OCF	Reserved			CC6ITF	CC5ITF	CC4ITF	CC3ITF	CC2ITF	CC1ITF	
				rc_w0	rc_w0	rc_w0	rc_w0				rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	

位域	名称	描述
31:28	Reserved	保留, 必须保持复位值
27	IOMBITF	IOM 刹车中断标记 (IOM Break interrupt flag) 一旦 IOM 刹车输入有效, 由硬件对该位置'1'。如果刹车输入无效, 则该位可由软件清'0'。 0: 无刹车事件产生; 1: IOM 刹车输入上检测到有效电平。
26:25	Reserved	保留, 必须保持复位值
24	CC7ITF	捕获/比较7中断标记 (Capture/Compare 7 interrupt flag) 参考CC7ITF描述。
23	COMPBITF	COMP 刹车中断标记 (COMP Break interrupt flag) 一旦 COMP 刹车输入有效, 由硬件对该位置'1'。如果刹车输入无效, 则该位可由软件清'0'。 0: 无刹车事件产生; 1: COMP 刹车输入上检测到有效电平。
22	PBKPITF	PVD 刹车中断标记 (PVD Break interrupt flag) 一旦PVD刹车输入有效, 由硬件对该位置'1'。如果刹车输入无效, 则该位可由软件清'0'。 0: 无刹车事件产生; 1: PVD 刹车输入上检测到有效电平。
21	LBKPITF	Lockup 刹车中断标记 (Lockup Break interrupt flag) 一旦 Lockup 刹车输入有效, 由硬件对该位置'1'。如果刹车输入无效, 则该位可由软件清'0'。 0: 无刹车事件产生;

位域	名称	描述
		1: Lockup 刹车输入上检测到有效电平。
20	Reserved	保留, 必须保持复位值
19	BITF	刹车中断标记 (Break interrupt flag) 一旦刹车输入有效, 由硬件对该位置'1'。如果刹车输入无效, 则该位可由软件清'0'。 0: 无刹车事件产生; 1: 刹车输入上检测到有效电平。
18	TITF	触发器中断标记 (Trigger interrupt flag) 当发生触发事件 (当从模式控制器处于除门控模式外的其它模式时, 在TRGI输入端检测到有效边沿, 或门控模式下的任一边沿) 时由硬件对该位置'1'。它由软件清'0'。 0: 无触发器事件产生; 1: 触发中断等待响应。
17	COMITF	COM中断标记 (COM interrupt flag) 一旦产生COM事件 (当捕获/比较控制位: CCxEN、CCxNEN、OCxMD已被更新) 该位由硬件置'1'。它由软件清'0'。 0: 无COM事件产生; 1: COM中断等待响应。
16	UDITF	更新中断标志 (Update interrupt flag) 当在以下条件下发生更新事件时, 该位由硬件设置: - 当 TIMx_CTRL1.UPDIS = 0 时, 并且重复计数器值上溢或下溢 (当重复计数器等于 0 时生成更新事件UEV)。 - 当 TIMx_CTRL1.UPRS = 0 时, TIMx_CTRL1.UPDIS = 0, 并通过软件设置 TIMx_EVTGEN.UDGN 位以重新初始化 CNT。 - 当 TIMx_CTRL1.UPRS = 0 时, TIMx_CTRL1.UPDIS = 0, 并且计数器 CNT 由触发事件重新初始化。(参见 TIMx_SMCTRL 寄存器说明) 该位由软件清零。 0: 未发生更新事件 1: 发生更新中断
15:12	Reserved	保留, 必须保持复位值
11	CC4OCF	捕获/比较4重复捕获标记 (Capture/Compare 4 overcapture flag) 参见CC1OCF描述。
10	CC3OCF	捕获/比较3重复捕获标记 (Capture/Compare 3 overcapture flag) 参见CC1OCF描述。
9	CC2OCF	捕获/比较2重复捕获标记 (Capture/Compare 2 overcapture flag) 参见CC1OCF描述。
8	CC1OCF	捕获/比较1重复捕获标记 (Capture/Compare 1 overcapture flag) 仅当相应的通道被配置为输入捕获时, 该标记可由硬件置1。写0可清除该位。 0: 无重复捕获产生; 1: 计数器的值被捕获到TIMx_CCDAT1寄存器时, CC1ITF的状态已经为'1'。
7:6	Reserved	保留, 必须保持复位值

位域	名称	描述
5	CC6ITF	捕获/比较6中断标记 (Capture/Compare 6 interrupt flag) 参考CC1ITF描述。
4	CC5ITF	捕获/比较5中断标记 (Capture/Compare 5 interrupt flag) 参考CC1ITF描述。
3	CC4ITF	捕获/比较4中断标记 (Capture/Compare 4 interrupt flag) 参考CC1ITF描述。
2	CC3ITF	捕获/比较3中断标记 (Capture/Compare 3 interrupt flag) 参考CC1ITF描述。
1	CC2ITF	捕获/比较2中断标记 (Capture/Compare 2 interrupt flag) 参考CC1ITF描述。
0	CC1ITF	捕获/比较1中断标记 (Capture/Compare 1 interrupt flag) <b>如果通道CC1配置为输出模式:</b> 除中央对齐模式外, 当计数器值与比较值相同时, 该位由硬件设置 (参见TIMx_CTRL1.CAMSEL 位描述)。 该位由软件清零。 0: 未发生匹配。 1: TIMx_CNT 的值与 TIMx_CCDA1 的值相同。 当 TIMx_CCDA1 的值大于 TIMx_AR 的值时, 如果计数器溢出 (在向上计数和向上/向下计数模式下) 和向下计数模式下溢, 则 TIMx_STS.CC1ITF 位将变为高电平。 <b>如果通道CC1配置为输入模式:</b> 当捕捉事件发生时, 该位由硬件设置。 该位由软件或读取 TIMx_CCDA1 清零。 0: 未发生输入捕捉。 1: 发生输入捕捉。 计数器值已在 TIMx_CCDA1 中捕获。 在 IC1 上检测到与所选极性相同的边沿。

## 12.4.5 事件产生寄存器 (TIMx\_EVTGEN)

偏移地址:0x0C

复位值:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				BGN	TGN	CCUDGN	UDGN	Reserved				CC4GN	CC3GN	CC2GN	CC1GN
				w	w	w	w					w	w	w	w

位域	名称	描述
31:12	Reserved	保留, 必须保持复位值

位域	名称	描述
11	BGN	产生刹车事件 (Break generation) 当由软件设置时, 该位可以产生一个刹车事件。而此时TIMx_BKDT.MOEN = 0, TIMx_STS.BITF = 1, 如果相应的中断和DMA被使能, 就会产生相应的中断和DMA。该位由硬件自动清零。 0: 无动作 1: 产生刹车事件
10	TGN	产生触发事件 (Trigger generation) 当由软件置位时, 该位可以产生一个触发事件。而此时TIMx_STS.TITF = 1, 如果相应的中断和DMA被使能, 就会产生相应的中断和DMA。该位由硬件自动清零。 0: 无动作 1: 产生触发事件
9	CCUDGN	捕获/比较事件, 产生控制更新 (Capture/Compare control update generation) 该位由软件设置。如果此时TIMx_CTRL2.CCCTL = 1, 则允许更新CCxEN、CCxNEN和OCxMD位。该位由硬件自动清零。 0: 无动作 1: 产生一个COM事件 <i>注意: 该位仅对具有互补输出的通道有效。</i>
8	UDGN	产生更新事件 (Update generation) 该位由软件置'1', 由硬件自动清'0'。 当由软件设置时, 该位可以生成更新事件。而此时计数器会重新初始化, 预分频计数器会被清零, 计数器在中央对齐或向上计数模式下会被清零, 但在向下计数模式下取TIMx_AR寄存器的值。该位由硬件自动清零。 0: 无动作 1: 生成更新事件
7:4	Reserved	保留, 必须保持复位值
3	CC4GN	产生捕获/比较4事件 (Capture/Compare 4 generation) 参考CC1GN描述。
2	CC3GN	产生捕获/比较3事件 (Capture/Compare 3 generation) 参考CC1GN描述。
1	CC2GN	产生捕获/比较2事件 (Capture/Compare 2 generation) 参考CC1GN描述。
0	CC1GN	产生捕获/比较1事件 (Capture/Compare 1 generation) 当由软件设置时, 该位可以产生一个捕获/比较事件。该位由硬件自动清零。 <b>CC1对应通道为输出模式时:</b> TIMx_STS.CC1ITF 标志将被拉高, 如果相应的中断和DMA被使能, 就会产生相应的中断和DMA。 <b>CC1对应通道为输入模式时:</b> TIMx_CCDAT1 将捕获当前计数器值, 并将TIMx_STS.CC1ITF标志拉高, 如果相应的中断和DMA被使能, 则会产生相应的中断和DMA。如果TIMx_STS.CC1ITF已经拉高, 则拉高TIMx_STS.CC1OCF。 0: 无动作

位域	名称	描述
		1: 生成 CC1 捕获/比较事件

## 12.4.6 从模式控制寄存器 (TIMx\_SMCTRL)

偏移地址: 0x10

复位值: 0x0000 0000

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

位域	名称	描述
31:17	Reserved	保留, 必须保持复位值
16	MSMD	主/从模式 (Master/slave mode) 0: 无作用; 1: 触发输入 (TRGI) 上的事件被延迟了, 以允许在当前定时器 (通过TRGO) 与它的从定时器间的完美同步。这对要求把几个定时器同步到一个单一的外部事件时是非常有用的。
15:12	EXTF[3:0]	外部触发滤波 (External trigger filter) 这些位定义了ETRP数字滤波器长度(RCC_TIMFILTCFG.TIM1FILTCLK[4:0] 寄存器决定相应数字滤波器的采样频率)。数字滤波器由一个事件计数器组成, 它记录到N个事件后会产生一个输出的跳变: 0000: 无滤波器, 以TIM内部工作时钟采样 0001: N=1 0010: N=2 0011: N=3 0100: N=4 0101: N=5 0110: N=6 0111: N=7 1xxx: N=8

11	EXTP	<p>外部触发极性 (External trigger polarity)</p> <p>该位选择是用ETR还是ETR的反相来作为触发操作</p> <p>0: ETR高电平或上升沿有效;</p> <p>1: ETR低电平或下降沿有效。</p>
10	EXCEN	<p>外部时钟使能位 (External clock enable) 该位启用外部时钟模式2。启用后, 计数器由ETRF信号上的任意有效边沿驱动。</p> <p>0: 禁止外部时钟模式2;</p> <p>1: 使能外部时钟模式2。</p> <p><i>注 1: 当同时使能外部时钟模式 1 和外部时钟模式 2 时, 外部时钟的输入为 ETRF。</i></p> <p><i>注2: 以下从机模式可以与外部时钟模式2同时使用: 复位模式、门控模式和触发模式; 但是, TRGI 无法连接到 ETRF (TIMx_SMCTRL.TSEL ≠ '111')。</i></p> <p><i>注 3: 设置 TIMx_SMCTRL.EXCEN 位与选择外部时钟模式 1 并将 TRGI 连接到 ETRF (TIMx_SMCTRL.SMSEL = 0111 和 TIMx_SMCTRL.TSEL = 111) 的效果相同</i></p>
9:8	EXTPS[1:0]	<p>外部触发预分频 (External trigger prescaler)</p> <p>外部触发信号 ETRP 的频率必须最多为 TIMxCLK 频率的 1/4。当输入更快的外部时钟时, 可以使用预分频器来降低 ETRP 的频率。</p> <p>00: 关闭预分频;</p> <p>01: ETRP频率除以2;</p> <p>10: ETRP频率除以4;</p> <p>11: ETRP频率除以8。</p>
7:4	SMSEL[3:0]	<p>从模式选择 (Slave mode selection)</p> <p>当选择了外部信号, 触发信号 (TRGI) 的有效边沿与选中的外部输入极性相关 (见输入控制寄存器和控制寄存器的说明)</p> <p>0000: 关闭从模式 – 如果CNTEN=1, 则预分频器直接由内部时钟驱动。</p> <p>0001: 编码器模式1 – 根据TI2FP2的电平, 计数器在TI1FP1的边沿向上/下计数。</p> <p>0010: 编码器模式2 – 根据TI1FP1的电平, 计数器在TI2FP2的边沿向上/下计数。</p> <p>0011: 编码器模式3 – 根据另一个信号的输入电平, 计数器在TI1FP1和TI2FP2的边沿向上/下计数。</p> <p>0100: 复位模式 – 在选定触发输入 (TRGI) 的上升沿, 计数器重新初始化并更新影子寄存器。</p> <p>0101: 门控模式 – 当触发输入 (TRGI) 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 (但不复位)。计数器的启动和停止都是受控的。</p> <p>0110: 触发模式 – 计数器在触发输入TRGI的上升沿启动 (但不复位), 只有计数器的启动是受控的。</p> <p>0111: 外部时钟模式1 – 选中的触发输入 (TRGI) 的上升沿驱动计数器。</p> <p>1000~1100: 保留。</p> <p>1101: 组合门控+复位模式 – 当触发输入 (TRGI) 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 (且复位)。计数器的启动和停止都是受控的。</p> <p>1110: 组合复位+触发模式 – 计数器在触发输入TRGI的上升沿启动 (且复位), 只有计数器的启动是受控的。</p> <p>1111: 保留。</p> <p><i>注: 如果TI1F_ED被选为触发输入 (TSEL=100) 时, 不要使用门控模式。这是因为, TI1F_ED在</i></p>

		每次TIF变化时输出一个脉冲，然而门控模式是要检查触发输入的电平。
3	Reserved	保留，必须保持复位值
2:0	TSEL[2:0]	触发选择（Trigger selection） 这3位选择用于同步计数器的触发输入。 000：内部触发0（ITR0） 100：TI1的边沿检测器（TI1F_ED） 001：内部触发1（ITR1） 101：滤波后的定时器输入1（TI1FP1） 010：内部触发2（ITR2） 110：滤波后的定时器输入2（TI2FP2） 011：内部触发3（ITR3） 111：外部触发输入（ETRF） 更多有关ITRx的细节，参见表12-3。 注：这些位只能在未用到（如SMSEL=0000）时被改变，以避免在改变时产生错误的边沿检测。

**表 12-3 TIMx 内部触发连接**

Slave timer	ITR0 (TSEL = 000)	ITR1 (TSEL = 001)	ITR2 (TSEL = 010)	ITR3 (TSEL = 011)
<b>TIM1</b>	TIM3	NA	NA	NA
<b>TIM3</b>	TIM1	NA	NA	NA
<b>TIM4</b>	NA	NA	NA	NA

## 12.4.7 DMA/中断使能寄存器（TIMx\_DINTEN）

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									COMIEN	TDEN	COMDEN	UDEN	BIEN	TIEN	UIEN
									rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CC4DEN	CC3DEN	CC2DEN	CC1DEN	Reserved	CC7IEN	CC6IEN	CC5IEN	CC4IEN	CC3IEN	CC2IEN	CC1IEN
				rw	rw	rw	rw		rw						

位域	名称	描述
31:23	Reserved	保留，必须保持复位值
22	COMIEN	允许COM中断（COM interrupt enable） 0：禁止COM中断； 1：允许COM中断。
21	TDEN	允许触发DMA请求（Trigger DMA request enable） 0：禁止触发DMA请求； 1：允许触发DMA请求。
20	COMDEN	允许COM的DMA请求（COM DMA request enable） 0：禁止COM的DMA请求；

位域	名称	描述
		1: 允许COM的DMA请求。
19	UDEN	允许更新的DMA请求 (Update DMA request enable) 0: 禁止更新的DMA请求; 1: 允许更新的DMA请求。
18	BIEN	允许刹车中断 (Break interrupt enable) 0: 禁止刹车中断; 1: 允许刹车中断。
17	TIEN	触发中断使能 (Trigger interrupt enable) 0: 禁止触发中断; 1: 使能触发中断。
16	UIEN	允许更新中断 (Update interrupt enable) 0: 禁止更新中断; 1: 允许更新中断。
15:12	Reserved	保留, 必须保持复位值
11	CC4DEN	允许捕获/比较4的DMA请求 (Capture/Compare 4 DMA request enable) 0: 禁止捕获/比较4的DMA请求; 1: 允许捕获/比较4的DMA请求。
10	CC3DEN	允许捕获/比较3的DMA请求 (Capture/Compare 3 DMA request enable) 0: 禁止捕获/比较3的DMA请求; 1: 允许捕获/比较3的DMA请求。
9	CC2DEN	允许捕获/比较2的DMA请求 (Capture/Compare 2 DMA request enable) 0: 禁止捕获/比较2的DMA请求; 1: 允许捕获/比较2的DMA请求。
8	CC1DEN	允许捕获/比较1的DMA请求 (Capture/Compare 1 DMA request enable) 0: 禁止捕获/比较1的DMA请求; 1: 允许捕获/比较1的DMA请求。
7	Reserved	保留, 必须保持复位值
6	CC7IEN	允许捕获/比较7中断 (Capture/Compare 7 interrupt enable) 0: 禁止捕获/比较7中断; 1: 允许捕获/比较7中断。
5	CC6IEN	允许捕获/比较6中断 (Capture/Compare 6 interrupt enable) 0: 禁止捕获/比较6中断; 1: 允许捕获/比较6中断。
4	CC5IEN	允许捕获/比较5中断 (Capture/Compare 5 interrupt enable) 0: 禁止捕获/比较5中断; 1: 允许捕获/比较5中断。

位域	名称	描述
3	CC4IEN	允许捕获/比较4中断 (Capture/Compare 4 interrupt enable) 0: 禁止捕获/比较4中断; 1: 允许捕获/比较4中断。
2	CC3IEN	允许捕获/比较3中断 (Capture/Compare 3 interrupt enable) 0: 禁止捕获/比较3中断; 1: 允许捕获/比较3中断。
1	CC2IEN	允许捕获/比较2中断 (Capture/Compare 2 interrupt enable) 0: 禁止捕获/比较2中断; 1: 允许捕获/比较2中断。
0	CC1IEN	允许捕获/比较1中断 (Capture/Compare 1 interrupt enable) 0: 禁止捕获/比较1中断; 1: 允许捕获/比较1中断。

## 12.4.8 捕获/比较模式寄存器 1 (TIMx\_CCMOD1)

偏移地址: 0x18

复位值: 0x0000 0000

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

输出比较模式:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2MD[2:0]		OC2CEN	OC2FEN	OC2PEN	CC2SEL[1:0]		OC1MD[2:0]		OC1CEN	OC1FEN	OC1PEN	CC1SEL[1:0]			
rw		rw	rw	rw	rw		rw		rw	rw	rw	rw			

位域	名称	描述
31:16	Reserved	保留, 必须保持复位值
15:13	OC2MD[2:0]	输出比较2模式 (Output Compare 2 mode)
12	OC2CEN	输出比较2清0使能 (Output Compare 2 clear enable)
11	OC2FEN	输出比较2快速使能 (Output Compare 2 fast enable)
10	OC2PEN	输出比较2预装载使能 (Output Compare 2 preload enable)
9:8	CC2SEL[1:0]	捕获/比较2选择。(Capture/Compare 2 selection) 该位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC2通道被配置为输出; 01: CC2通道被配置为输入, IC2映射在TI2上; 10: CC2通道被配置为输入, IC2映射在TI1上;

位域	名称	描述
		11: CC2通道被配置为输入, IC2映射在TRC上。此模式仅工作在内部触发输入被选中时(由TIMx_SMCTRL寄存器的TSEL位选择)。 <i>注: CC2SEL仅在通道关闭时(TIMx_CCEN寄存器的CC2EN=0)才是可写的。</i>
7:5	OC1MD[2:0]	输出比较1模式 (Output Compare 1 mode) 这些位用于管理输出参考信号 OC1REF, 它决定了 OC1 和 OC1N 的值, 在高电平有效, 而 OC1 和 OC1N 的有效电平取决于 TIMx_CCEN.CC1P 和 TIMx_CCEN.CC1NP 位。 000: 冻结。TIMx_CCDAT1 寄存器和计数器 TIMx_CNT 之间的比较对 OC1REF 信号没有影响。 001: 将通道 1 设置为匹配时的有效电平。当 TIMx_CCDAT1 = TIMx_CNT 时, OC1REF 信号将被强制为高电平。 010: 将通道 1 设置为匹配时的无效电平。当 TIMx_CCDAT1 = TIMx_CNT 时, OC1REF 信号将被强制为低电平。 011: 翻转。当 TIMx_CCDAT1 = TIMx_CNT 时, OC1REF 信号将被翻转。 100: 强制无效电平。OC1REF 信号被强制为低电平。 101: 强制有效电平。OC1REF 信号被强制为高电平。 110: PWM 模式 1 - 在向上计数模式下, 如果 TIMx_CNT < TIMx_CCDAT1, 则通道 1 的 OC1REF 信号为高电平, 否则为低电平。在向下计数模式下, 如果 TIMx_CNT > TIMx_CCDAT1, 则通道 1 的 OC1REF 信号为低电平, 否则为高电平。 111: PWM 模式 2 - 在向上计数模式下, 如果 TIMx_CNT < TIMx_CCDAT1, 则通道 1 的 OC1REF 信号为低电平, 否则为高电平。在向下计数模式下, 如果 TIMx_CNT > TIMx_CCDAT1, 则通道 1 的 OC1REF 信号为高电平, 否则为低电平。 <i>注 1: 在 PWM 模式 1 或 PWM 模式 2 中, OC1REF 电平仅在比较结果改变或输出比较模式从冻结模式切换到 PWM 模式时才会改变。</i>
4	OC1CEN	输出比较1清'0'使能 (Output Compare 1 clear enable) 0: OC1REF 不受ETRF输入的影响; 1: 一旦检测到ETRF输入高电平, 清除OC1REF=0。
3	OC1FEN	输出比较1 快速使能 (Output Compare 1 fast enable) 该位用于加快CC输出对触发输入事件的响应。 0: 根据计数器与CCDAT1的值, CC1正常操作, 即使触发器是打开的。当触发器的输入有一个有效沿时, 激活CC1输出的最小延时为5个时钟周期。 1: 输入到触发器的有效沿的作用就像发生了一次比较匹配。因此, OC1被设置为比较电平而与比较结果无关。采样触发器的有效沿和CC1输出间的延时被缩短为3个时钟周期。 OCxFEN只在通道被配置成PWM1或PWM2模式时起作用。
2	OC1PEN	输出比较 1 预加载使能 (Output Compare 1 preload enable) 0: 禁用 TIMx_CCDAT1 寄存器的预加载功能。支持随时对TIMx_CCDAT1寄存器进行写操作, 写入的值立即生效。 1: 使能 TIMx_CCDAT1 寄存器的预加载功能。仅对预加载寄存器进行读写操作。当更新事件发生时, TIMx_CCDAT1 的值被加载到影子寄存器中。

位域	名称	描述
		<i>注 1: 只有当 TIMx_CTRL1.ONEPM = 1 (在单脉冲模式下) 时, 才能使用 PWM 模式而不验证预加载寄存器, 否则无法预测其他行为。</i>
1:0	CC1SEL[1:0]	捕获/比较1 选择。(Capture/Compare 1 selection) 这2位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC1通道被配置为输出; 01: CC1通道被配置为输入, IC1映射在TI1上; 10: CC1通道被配置为输入, IC1映射在TI2上; 11: CC1通道被配置为输入, IC1映射在TRC上。此模式仅工作在内部触发输入被选中时(由TIMx_SMCTRL寄存器的TSEL位选择)。 <i>注: CC1SEL 仅在通道关闭时 (TIMx_CCEN寄存器的CC1EN=0) 才是可写的。</i>

**输入捕获模式:**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
IC2F[3:0]				IC2PSC[1:0]			CC2SEL[1:0]			IC1F[3:0]			IC1PSC[1:0]		CC1SEL[1:0]	
rw				rw			rw			rw			rw		rw	

位域	名称	描述
31:16	Reserved	保留, 必须保持复位值
15:12	IC2F[3:0]	输入捕获2滤波器 (Input capture 2 filter)
11:10	IC2PSC[1:0]	输入/捕获2预分频器 (Input capture 2 prescaler)
9:8	CC2SEL[1:0]	捕获/比较2选择 (Capture/Compare 2 selection) 这2位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC2通道被配置为输出; 01: CC2通道被配置为输入, IC2映射在TI2上; 10: CC2通道被配置为输入, IC2映射在TI1上; 11: CC2通道被配置为输入, IC2映射在TRC上。此模式仅工作在内部触发输入被选中时(由TIMx_SMCTRL寄存器的TSEL位选择)。 <i>注: CC2SEL 仅在通道关闭时 (TIMx_CCEN寄存器的CC2EN=0) 才是可写的。</i>

7:4	IC1F[3:0]	<p>输入捕获1滤波器 (Input capture 1 filter)</p> <p>这几位定义了TIM1数字滤波器长度(RCC_TIMFILTCFG.TIM1FILTCLK[4:0] 寄存器决定相应数字滤波器的采样频率)。数字滤波器由一个事件计数器组成, 它记录到N个事件后会产生一个输出的跳变:</p> <p>0000: 无滤波器, 以TIM内部工作时钟采样</p> <p>0001: N=1</p> <p>0010: N=2</p> <p>0011: N=3</p> <p>0100: N=4</p> <p>0101: N=5</p> <p>0110: N=6</p> <p>0111: N=7</p> <p>1xxx: N=8</p>
3:2	IC1PSC[1:0]	<p>输入/捕获1预分频器 (Input capture 1 prescaler)</p> <p>这2位定义了CC1输入 (IC1) 的预分频系数。</p> <p>一旦TIMx_CCEN.CC1EN=0, 则预分频器复位。</p> <p>00: 无预分频器, 捕获输入口上检测到的每一个边沿都触发一次捕获;</p> <p>01: 每2个事件触发一次捕获;</p> <p>10: 每4个事件触发一次捕获;</p> <p>11: 每8个事件触发一次捕获。</p>
1:0	CC1SEL[1:0]	<p>捕获/比较1选择 (Capture/Compare 1 Selection)</p> <p>这2位定义通道的方向 (输入/输出), 及输入脚的选择:</p> <p>00: CC1通道被配置为输出;</p> <p>01: CC1通道被配置为输入, IC1映射在TI1上;</p> <p>10: CC1通道被配置为输入, IC1映射在TI2上;</p> <p>11: CC1通道被配置为输入, IC1映射在TRC上。此模式仅工作在内部触发输入被选中时 (由TIMx_SMCTRL寄存器的TSEL位选择)。</p> <p><i>注: CC1SEL仅在通道关闭时 (TIMx_CCEN寄存器的CC1EN=0) 才是可写的。</i></p>

## 12.4.9 捕获/比较模式寄存器 2 (TIMx\_CCMOD2)

偏移地址: 0x1C

复位值: 0x0000 0000

参看以上 CCMOD1 寄存器的描述

输出比较模式:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4MD[2:0]		OC4CEN	OC4FEN	OC4PEN	CC4SEL[1:0]		OC3MD[2:0]		OC3CEN	OC3FEN	OC3PEN	CC3SEL[1:0]			

rw      rw      rw      rw      rw      rw      rw      rw      rw      rw

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:13	OC4MD[2:0]	输出比较4模式（Output compare 4 mode）
12	OC4CEN	输出比较4清0使能（Output compare 4 clear enable）
11	OC4FEN	输出比较4快速使能（Output compare 4 fast enable）
10	OC4PEN	输出比较4预装载使能（Output compare 4 preload enable）
9:8	CC4SEL[1:0]	捕获/比较4选择（Capture/Compare 4 selection） 该2位定义通道的方向（输入/输出），及输入脚的选择： 00：CC4通道被配置为输出； 01：CC4通道被配置为输入，IC4映射在TI4上； 10：CC4通道被配置为输入，IC4映射在TI3上； 11：CC4通道被配置为输入，IC4映射在TRC上。此模式仅工作在内部触发输入被选中时（由TIMx_SMCTRL寄存器的TSEL位选择）。 <i>注：CC4SEL仅在通道关闭时（TIMx_CCEN寄存器的CC4EN=0）才是可写的。</i>
7:5	OC3MD[2:0]	输出比较3模式（Output compare 3 mode）
4	OC3CEN	输出比较3清0使能（Output compare 3 clear enable）
3	OC3FEN	输出比较3快速使能（Output compare 3 fast enable）
2	OC3PEN	输出比较3预装载使能（Output compare 3 preload enable）
1:0	CC3SEL[1:0]	捕获/比较3选择（Capture/Compare 3 selection） 这2位定义通道的方向（输入/输出），及输入脚的选择： 00：CC3通道被配置为输出； 01：CC3通道被配置为输入，IC3映射在TI3上； 10：CC3通道被配置为输入，IC3映射在TI4上； 11：CC3通道被配置为输入，IC3映射在TRC上。此模式仅工作在内部触发输入被选中时（由TIMx_SMCTRL寄存器的TSEL位选择）。 <i>注：CC3SEL仅在通道关闭时（TIMx_CCEN寄存器的CC3EN=0）才是可写的。</i>

#### 输入捕获模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
IC4F[3:0]				IC4PSC[1:0]			CC4SEL[1:0]			IC3F[3:0]			IC3PSC[1:0]		CC3SEL[1:0]	
rw				rw			rw			rw			rw		rw	

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:12	IC4F[3:0]	输入捕获4滤波器 (Input capture 4 filter)
11:10	IC4PSC[1:0]	输入/捕获4预分频器 (Input capture 4 prescaler)
9:8	CC4SEL[1:0]	捕获/比较4选择 (Capture/Compare 4 selection) 这2位定义通道的方向(输入/输出)，及输入脚的选择： 00：CC4通道被配置为输出； 01：CC4通道被配置为输入，IC4映射在TI4上； 10：CC4通道被配置为输入，IC4映射在TI3上； 11：CC4通道被配置为输入，IC4映射在TRC上。此模式仅工作在内部触发输入被选中时(由TIMx_SMCTRL寄存器的TSEL位选择)。 <i>注：CC4SEL仅在通道关闭时(TIMx_CCEN寄存器的CC4EN=0)才是可写的。</i>
7:4	IC3F[3:0]	输入捕获3滤波器 (Input capture 3 filter)
3:2	IC3PSC[1:0]	输入/捕获3预分频器 (Input capture 3 prescaler)
1:0	CC3SEL[1:0]	捕获/比较3选择 (Capture/compare 3 selection) 这2位定义通道的方向(输入/输出)，及输入脚的选择： 00：CC3通道被配置为输出； 01：CC3通道被配置为输入，IC3映射在TI3上； 10：CC3通道被配置为输入，IC3映射在TI4上； 11：CC3通道被配置为输入，IC3映射在TRC上。此模式仅工作在内部触发输入被选中时(由TIMx_SMCTRL寄存器的TSEL位选择)。 <i>注：CC3SEL仅在通道关闭时(TIMx_CCEN寄存器的CC3EN=0)才是可写的。</i>

### 12.4.10 捕获/比较模式寄存器 3 (TIMx\_CCMOD3)

偏移地址：0x20

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				OC7MD[2:0]			Reserved				OC7CEN	OC7FEN	OC7PEN		
				rw							rw	rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC6MD[2:0]		OC6CEN	OC6FEN	OC6PEN	Reserved			OC5MD[2:0]		OC5CEN	OC5FEN	OC5PEN	Reserved		
rw		rw	rw	rw				rw		rw	rw	rw			

位域	名称	描述
31:28	Reserved	保留，必须保持复位值
27:25	OC7MD[2:0]	输出比较7模式 (Output compare 7 mode)
24:19	Reserved	保留，必须保持复位值
18	OC7CEN	输出比较7清0使能 (Output compare 7 clear enable)

位域	名称	描述
17	OC7FEN	输出比较7快速使能 (Output compare 7 fast enable)
16	OC7PEN	输出比较7预装载使能 (Output compare 7 preload enable)
15:13	OC6MD[2:0]	输出比较6模式 (Output compare 6 mode)
12	OC6CEN	输出比较6清0使能 (Output compare 6 clear enable)
11	OC6FEN	输出比较6快速使能 (Output compare 6 fast enable)
10	OC6PEN	输出比较6预装载使能 (Output compare 6 preload enable)
9:8	Reserved	保留, 必须保持复位值
7:5	OC5MD[2:0]	输出比较5模式 (Output compare 5 mode)
4	OC5CEN	输出比较5清0使能 (Output compare 5 clear enable)
3	OC5FEN	输出比较5快速使能 (Output compare 5 fast enable)
2	OC5PEN	输出比较5预装载使能 (Output compare 5 preload enable)
1:0	Reserved	保留, 必须保持复位值

### 12.4.11 捕获/比较使能寄存器 (TIMx\_CCEN)

偏移地址: 0x24

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved				CC7P	CC7EN	CC6P	CC6EN	Reserved				CC5P	CC5EN	Reserved		
				rw	rw	rw	rw					rw	rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
CC4P	CC4EN	CC4NP	CC4NEN	CC3P	CC3EN	CC3NP	CC3NEN	CC2P	CC2EN	CC2NP	CC2NEN	CC1P	CC1EN	CC1NP	CC1NEN	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

位域	名称	描述
31:26	Reserved	保留, 必须保持复位值
25	CC7P	捕获/比较7输出极性 (Capture/Compare 7 output polarity) 参考TIMx_CCEN.CC1P的描述。
24	CC7EN	捕获/比较7输出使能 (Capture/Compare 7 output enable) 参考TIMx_CCEN.CC1EN 的描述
23	CC6P	捕获/比较6输出极性 (Capture/Compare 6 output polarity) 参考TIMx_CCEN.CC1P的描述。
22	CC6EN	捕获/比较6输出使能 (Capture/Compare 6 output enable) 参考TIMx_CCEN.CC1EN 的描述
21:20	Reserved	保留, 必须保持复位值
19	CC5P	捕获/比较5输出极性 (Capture/Compare 5 output polarity) 参考TIMx_CCEN.CC1P的描述。

位域	名称	描述
18	CC5EN	捕获/比较5输出使能 (Capture/Compare 5 output enable) 参考TIMx_CCEN.CC1EN 的描述
17: 16	Reserved	保留, 必须保持复位值
15	CC4P	捕获/比较4输出极性 (Capture/Compare 4 output polarity) 参考TIMx_CCEN.CC1P的描述。
14	CC4EN	捕获/比较4输出使能 (Capture/Compare 4 output enable) 参考TIMx_CCEN.CC1EN 的描述。
13	CC4NP	捕获/比较4互补输出极性 (Capture/Compare 4 complementary output polarity) 参考TIMx_CCEN.CC1NP的描述。
12	CC4NEN	捕获/比较4互补输出使能 (Capture/Compare 4 complementary output enable) 参考TIMx_CCEN.CC1NEN的描述。
11	CC3P	捕获/比较3输出极性 (Capture/Compare 3 output polarity) 参考TIMx_CCEN.CC1P的描述。
10	CC3EN	捕获/比较3输出使能 (Capture/Compare 3 output enable) 参考TIMx_CCEN.CC1E 的描述。
9	CC3NP	捕获/比较3互补输出极性 (Capture/Compare 3 complementary output polarity) 参考TIMx_CCEN.CC1NP的描述。
8	CC3NEN	捕获/比较3互补输出使能 (Capture/Compare 3 complementary output enable) 参考TIMx_CCEN.CC1NEN的描述。
7	CC2P	捕获/比较2输出极性 (Capture/Compare 2 output polarity) 参考TIMx_CCEN.CC1P的描述。
6	CC2EN	捕获/比较2输出使能 (Capture/Compare 2 output enable) 参考TIMx_CCEN.CC1EN的描述。
5	CC2NP	捕获/比较2互补输出极性 (Capture/Compare 2 complementary output polarity) 参考TIMx_CCEN.CC1NP的描述。
4	CC2NEN	捕获/比较2互补输出使能 (Capture/Compare 2 complementary output enable) 参考TIMx_CCEN.CC1NEN的描述。
3	CC1P	捕获/比较1输出极性 (Capture/Compare 1 output polarity) <b>CC1对应通道为输出模式时:</b> 0: OC1 高电平有效 1: OC1 低电平有效 <b>CC1对应通道为输入模式时:</b> 此时, 该位用于选择是使用IC1还是IC1的反相信号作为触发信号或捕捉信号。 0: 非反相: 当 IC1 产生上升沿时发生捕获动作。 当用作外部触发时, IC1 是非反相的。 1: 反相: 当 IC1 产生下降沿时发生捕获动作。 当用作外部触发时, IC1 被反相。
2	CC1EN	捕获/比较1输出使能 (Capture/Compare 1 output enable) <b>CC1通道配置为输出:</b> 0: 关闭— OC1禁止输出, 因此OC1的输出电平依赖于MOEN、OSSI、OSSR、OI1、OI1N和CC1NEN位的值。

位域	名称	描述
		1: 开启 - OC1信号输出到对应的输出引脚, 其输出电平依赖于MOEN、OSSI、OSSR、OI1、OI1N和CC1NEN位的值。 <b>CC1通道配置为输入:</b> 该位决定了计数器的值是否能捕获入TIMx_CC1N寄存器。 0: 捕获禁止; 1: 捕获使能。
1	CC1NP	捕获/比较1互补输出极性 (Capture/Compare 1 complementary output polarity) 0: OC1N高电平有效; 1: OC1N低电平有效。
0	CC1NEN	捕获/比较1互补输出使能 (Capture/Compare 1 complementary output enable) 0: 禁用 - 禁用输出 OC1N 信号。OC1N 的电平取决于TIMx_BKDT.MOEN、TIMx_BKDT.OSSI、TIMx_BKDT.OSSR、TIMx_CTRL2.OI1、TIMx_CTRL2.OI1N 和 TIMx_CCEN.CC1EN 的值。 1: 使能 - 使能输出 OC1N 信号。OC1N 的电平取决于TIMx_BKDT.MOEN、TIMx_BKDT.OSSI、TIMx_BKDT.OSSR、TIMx_CTRL2.OI1、TIMx_CTRL2.OI1N 和 TIMx_CCEN.CC1EN 的值。

**表 12-4 带刹车功能的互补输出通道 OCx 和 OCxN 的控制位**

控制位					输出状态 <sup>(1)</sup>		
MOEN	OSSI	OSSR	CCxEN	CCxNEN	OCx 输出状态	OCxN 输出状态	
1X		0	0	0	输出禁止 (与定时器断开)	输出禁止 (与定时器断开)	
					OCx=0, OCx_EN=0	OCxN=0, OCxN_EN=0	
		0	0	1	1	输出禁止 (与定时器断开)	OCxREF + 极性,
						OCx=0, OCx_EN=0	OCxN= OCxREF xor CCxNP, OCxN_EN=1
		0	1	1	0	OCxREF + 极性,	输出禁止 (与定时器断开)
						OCx= OCxREF xor CCxP, OCx_EN=1	OCxN=0, OCxN_EN=0
		0	1	1	1	OCxREF + 极性 + 死区, OCx_EN=1	OCxREF 反相 + 极性 + 死区, OCxN_EN=1
						输出禁止 (与定时器断开)	输出禁止 (与定时器断开)
		1	0	0	0	OCx=CCxP, OCx_EN=0	OCxN=CCxNP, OCxN_EN=0
						关闭状态 (输出使能且为无效电平)	OCxREF + 极性,
1	0	1	1	OCx=CCxP, OCx_EN=1	OCxN= OCxREF xor CCxNP, OCxN_EN=1		
				OCxREF + 极性,	关闭状态 (输出使能且为无效电平)		

					OCx= OCxREF xor CCxP, OCx_EN=1	OCxN=CCxNP, OCxN_EN=1
		1	1	1	OCxREF + 极性 + 死区, OCx_EN=1	OCxREF 反相 + 极性 + 死区, OCxN_EN=1
0	X	0	0	0	输出禁止（与定时器断开）	
		0	0	1	异步：OCx=CCxP, OCx_EN=0, OCxN=CCxNP, OCxN_EN=0;	
		0	1	0	若时钟存在：假设 $(CCxP \wedge OIx) \wedge (CCxNP \wedge OIxN) \neq 0$ ,	
		0	1	1	经过一个死区时间后 OCx=OIx, OCxN=OIxN	
		1	0	0	关闭状态（输出使能且为无效电平）	
		1	0	1	异步：OCx=CCxP, OCx_EN=1, OCxN=CCxNP, OCxN_EN=1;	
		1	1	0	若时钟存在：假设 $(CCxP \wedge OIx) \wedge (CCxNP \wedge OIxN) \neq 0$ ,	
		1	1	1	经过一个死区时间后 OCx=OIx, OCxN=OIxN,	

1. 如果一个通道的 2 个输出都没有使用（CCxEN = CCxNEN = 0），那么 OIx, OIxN, CCxP 和 CCxNP 都必须清零。

注：引脚连接到互补的 OCx 和 OCxN 通道的外部 I/O 引脚的状态，取决于 OCx 和 OCxN 通道状态和 GPIO 以及 AFIO 寄存器。

## 12.4.12 捕获/比较寄存器 1（TIMx\_CC DAT1）

偏移地址：0x28

复位值：0x0000 0000

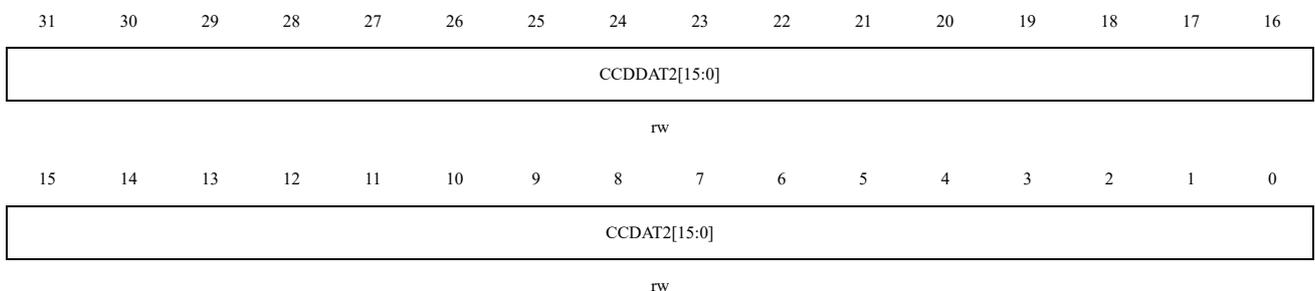
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCDDAT1[15:0]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCDAT1[15:0]															
rw															

位域	名称	描述
31:16	CCDDAT1[15:0]	捕获/比较通道1向下计数值(Capture/Compare 1 down-counting value), 专用于中央对齐非对称模式。 ■ CC1 通道只能配置为输出: CCDDAT1 包含要与计数器 TIMx_CNT 比较的值 (仅当 TIMx_CTRL1.DIR = 1 且处于非对称模式时), 在 OC1 输出上发出信号。 如果未在 TIMx_CCMOD1.OC1PEN 位中选择预加载功能, 则写入的值会立即传输到有效寄存器。 否则, 仅当更新事件发生时, 此预加载值才会传输到活动寄存器。
15:0	CCDAT1[15:0]	捕获/比较通道1的值 (Capture/Compare 1 value) ■ CC1 通道配置为输出: CCDAT1 包含要与计数器 TIMx_CNT 比较的值, 在 OC1 输出上发出信号。 如果未在 TIMx_CCMOD1.OC1PEN 位中选择预加载功能, 则写入的值会立即传输到有效寄存器。 否则, 仅当更新事件发生时, 此预加载值才会传输到活动寄存器。 ■ CC1 通道配置为输入: CCDAT1 包含由最后一个输入捕获 1 事件 (IC1) 传输的计数器值。 当配置为输入模式时, 寄存器 CCDAT1 和 CCDDAT1 只能读取。 当配置为输出模式时, 寄存器 CCDAT1 和 CCDDAT1 是可读写的。

### 12.4.13 捕获/比较寄存器 2 (TIMx\_CCDA2)

偏移地址: 0x2C

复位值: 0x0000 0000

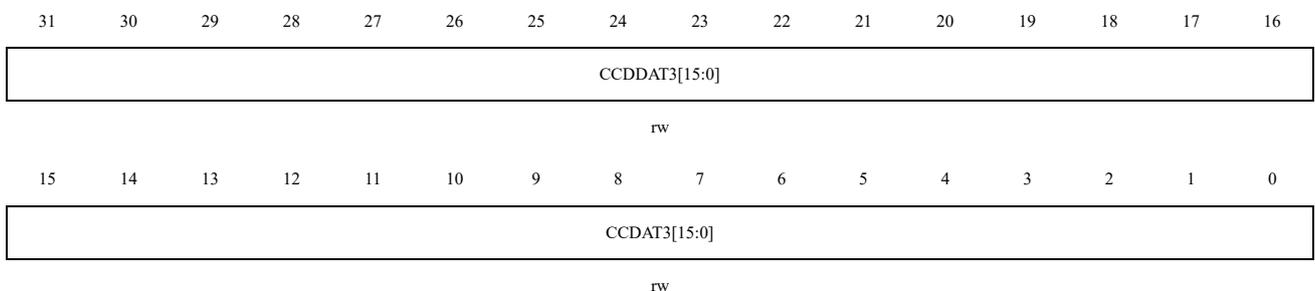


位域	名称	描述
31:16	CCDDAT2[15:0]	捕获/比较通道2向下计数值(Capture/Compare 2 down-counting value), 专用于中央对齐非对称模式。 ■ CC2 通道只能配置为输出: CCDDAT2 包含要与计数器 TIMx_CNT 比较的值 (仅当 TIMx_CTRL1.DIR = 1 且处于非对称模式时), 在 OC2 输出上发出信号。 如果未在 TIMx_CCMOD1.OC2PEN 位中选择预加载功能, 则写入的值会立即传输到有效寄存器。 否则, 仅当更新事件发生时, 此预加载值才会传输到活动寄存器。
15:0	CCDAT2[15:0]	捕获/比较通道2的值 (Capture/Compare 2 value) ■ CC2 通道配置为输出: CCDAT2 包含要与计数器 TIMx_CNT 比较的值, 在 OC2 输出上发出信号。 如果未在 TIMx_CCMOD1.OC2PEN 位中选择预加载功能, 则写入的值会立即传输到有效寄存器。 否则, 仅当更新事件发生时, 此预加载值才会传输到活动寄存器。 ■ CC2 通道配置为输入: CCDAT2 包含由最后一个输入捕获 2 事件 (IC2) 传输的计数器值。 当配置为输入模式时, 寄存器 CCDAT2 和 CCDDAT2 只能读取。 当配置为输出模式时, 寄存器 CCDAT2 和 CCDDAT2 是可读写的。

### 12.4.14 捕获/比较寄存器 3 (TIMx\_CCDA3)

偏移地址: 0x30

复位值: 0x0000 0000



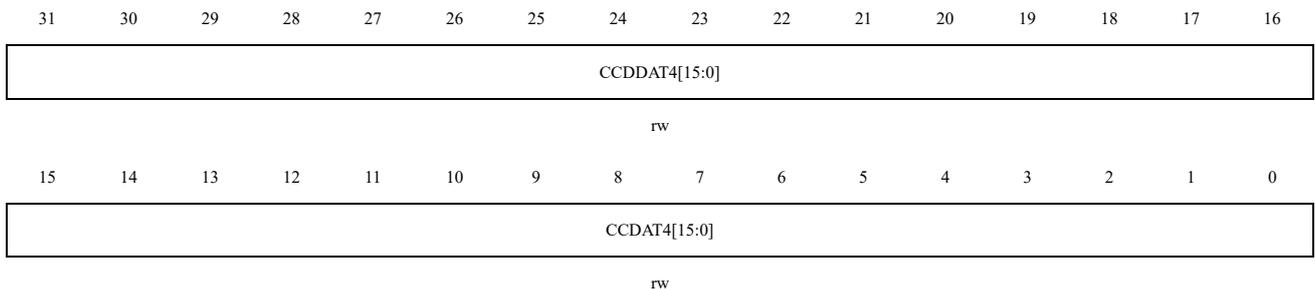
位域	名称	描述
31:16	CCDDAT3[15:0]	捕获/比较通道3向下计数值(Capture/Compare 3 down-counting value), 专用于中央对齐非对称模式。 ■ CC3 通道只能配置为输出: CCDDAT3 包含要与计数器 TIMx_CNT 比较的值 (仅当 TIMx_CTRL1.DIR = 1 且处于非对称模式时), 在 OC3 输出上发出信号。 如果未在 TIMx_CCMOD2.OC3PEN 位中选择预加载功能, 则写入的值会立即传输到有效寄存器。 否则, 仅当更新事件发生时, 此预加载值才会传输到活动寄存器。
15:0	CCDAT3[15:0]	捕获/比较通道3的值 (Capture/Compare 3 value) ■ CC3 通道配置为输出: CCDAT3 包含要与计数器 TIMx_CNT 比较的值, 在 OC3 输出上发出信号。

位域	名称	描述
31:16	CCDDAT3[15:0]	捕获/比较通道3向下计数值(Capture/Compare 3 down-counting value), 专用于中央对齐非对称模式。 ■ CC3 通道只能配置为输出: CCDDAT3 包含要与计数器 TIMx_CNT 比较的值 (仅当 TIMx_CTRL1.DIR = 1 且处于非对称模式时), 在 OC3 输出上发出信号。 如果未在 TIMx_CCMOD2.OC3PEN 位中选择预加载功能, 则写入的值会立即传输到有效寄存器。 否则, 仅当更新事件发生时, 此预加载值才会传输到活动寄存器。
		如果未在 TIMx_CCMOD2.OC3PEN 位中选择预加载功能, 则写入的值会立即传输到有效寄存器。 否则, 仅当更新事件发生时, 此预加载值才会传输到活动寄存器。 ■ CC3 通道配置为输入: CCDAT3 包含由最后一个输入捕获 3 事件 (IC3) 传输的计数器值。 当配置为输入模式时, 寄存器 CCDAT3 和 CCDDAT3 只能读取。 当配置为输出模式时, 寄存器 CCDAT3 和 CCDDAT3 是可读写的。

### 12.4.15 捕获/比较寄存器 4 (TIMx\_CCDA4)

偏移地址: 0x34

复位值: 0x0000 0000



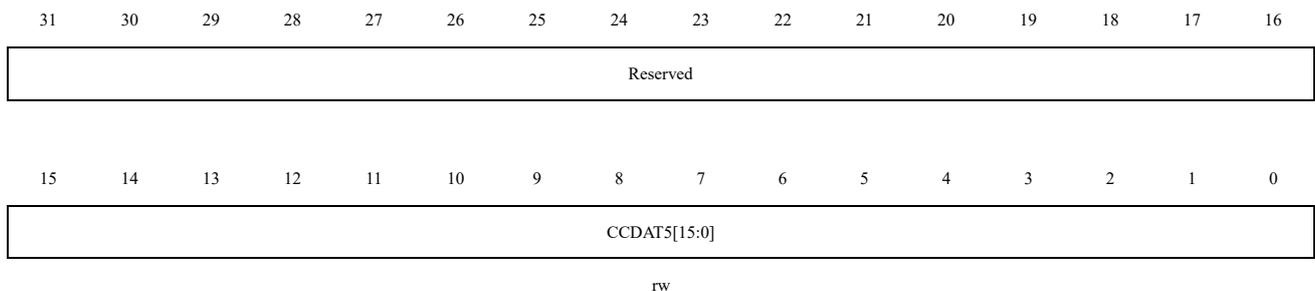
位域	名称	描述
31:16	CCDDAT4[15:0]	捕获/比较通道4向下计数值(Capture/Compare 4 down-counting value), 专用于中央对齐非对称模式。 ■ CC4 通道只能配置为输出: CCDDAT4 包含要与计数器 TIMx_CNT 比较的值 (仅当 TIMx_CTRL1.DIR = 1 且处于非对称模式时), 在 OC4 输出上发出信号。 如果未在 TIMx_CCMOD2.OC4PEN 位中选择预加载功能, 则写入的值会立即传输到有效寄存器。 否则, 仅当更新事件发生时, 此预加载值才会传输到活动寄存器。
15:0	CCDAT4[15:0]	捕获/比较通道4的值 (Capture/Compare 4 value) ■ CC4 通道配置为输出: CCDAT4 包含要与计数器 TIMx_CNT 比较的值, 在 OC4 输出上发出信号。 如果未在 TIMx_CCMOD2.OC4PEN 位中选择预加载功能, 则写入的值会立即传输到有效寄存器。 否则, 仅当更新事件发生时, 此预加载值才会传输到活动寄存器。 ■ CC4 通道配置为输入:

		CCDAT4 包含由最后一个输入捕获 4 事件 (IC4) 传输的计数器值。 当配置为输入模式时, 寄存器 CCDAT4 和 CCDDAT4 只能读取。 当配置为输出模式时, 寄存器 CCDAT4 和 CCDDAT4 是可读写的。
--	--	---

## 12.4.16 捕获/比较寄存器 5 (TIMx\_CC DAT5)

偏移地址: 0x38

复位值: 0x0000 0000

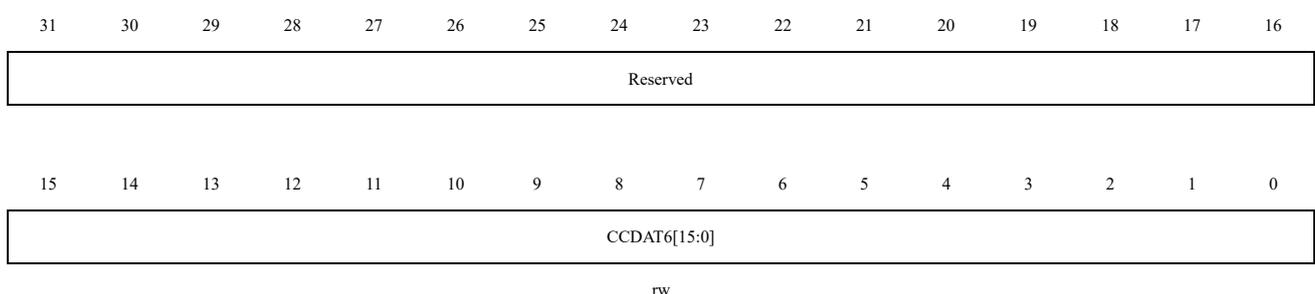


位域	名称	描述
31:16	Reserved	保留, 必须保持复位值
15:0	CCDAT5[15:0]	捕获/比较通道5的值 (Capture/Compare 5 value) <ul style="list-style-type: none"> <li>■ CC5 通道只能配置为输出:</li> </ul> CCDAT5 包含要与计数器 TIMx_CNT 比较的值, 在 OC5 输出上发出信号。 如果未在 TIMx_CCMOD3.OC5PEN 位中选择预加载功能, 则写入的值会立即传输到有效寄存器。否则, 仅当更新事件发生时, 此预加载值才会传输到活动寄存器。 CC5 用于比较器消隐。

## 12.4.17 捕获/比较寄存器 6 (TIMx\_CC DAT6)

偏移地址: 0x3C

复位值: 0x0000 0000



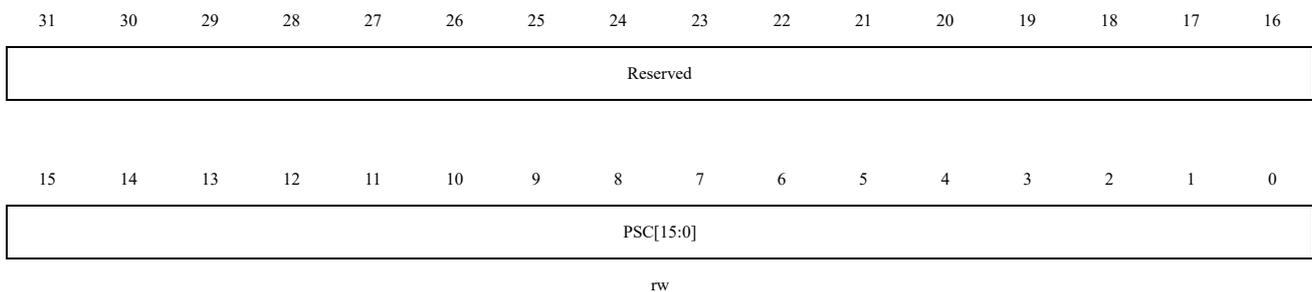
位域	名称	描述
31:16	Reserved	保留, 必须保持复位值

15:0	CCDAT6[15:0]	捕获/比较通道6的值 (Capture/Compare 6 value) ■ CC6 通道只能配置为输出： CCDAT6 包含要与计数器 TIMx_CNT 比较的值，在 OC6 输出上发出信号。 如果未在 TIMx_CCMOD3.OC6PEN 位中选择预加载功能，则写入的值会立即传输到有效寄存器。 否则，仅当更新事件发生时，此预加载值才会传输到活动寄存器。
------	--------------	--

### 12.4.18 预分频器 (TIMx\_PSC)

偏移地址: 0x40

复位值: 0x0000 0000

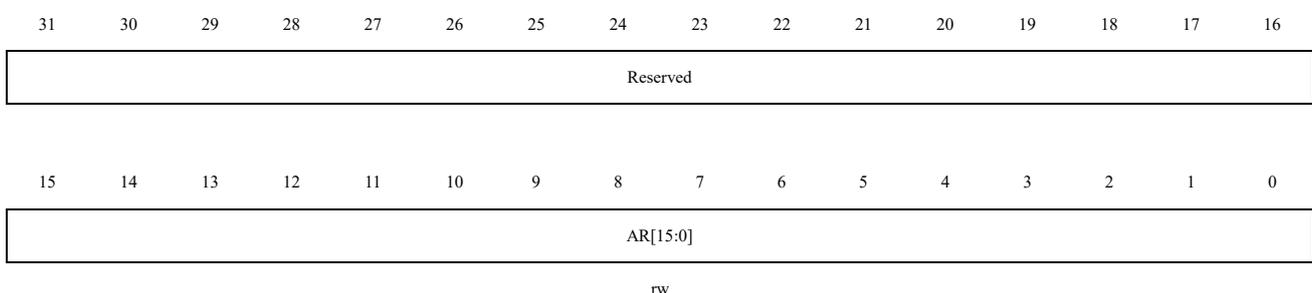


位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:0	PSC[15:0]	预分频器的值 (Prescaler value) 计数器时钟 $f_{CK\_CNT} = f_{CK\_PSC} / (PSC [15:0] + 1)$ 。 每次发生更新事件时，PSC 值都会加载到预分频器的影子寄存器中。

### 12.4.19 自动重载寄存器 (TIMx\_AR)

偏移地址: 0x44

复位值: 0x0000 FFFF



位域	名称	描述
31:16	Reserved	保留，必须保持复位值

15:0	AR[15:0]	自动重载的值（Auto-reload value） AR包含了将要装载入实际的自动重载寄存器的值。详细参考12.3.1节：有关AR的更新和动作。 当自动重载的值为空时，计数器不工作。
------	----------	---

## 12.4.20 计数器（TIMx\_CNT）

偏移地址：0x48

复位值：0x0000 0000

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

rw

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:0	CNT[15:0]	计数器的值（Counter value）

## 12.4.21 重复计数寄存器（TIMx\_REPCNT）

偏移地址：0x4C

复位值：0x0000 0000

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

rw

位域	名称	描述
31:8	Reserved	保留，必须保持复位值
7:0	REPCNT[7:0]	重复计数器的值（Repetition counter value） 重复计数器仅在给定数量 (N+1) 个计数器周期后用于生成更新事件或更新定时器寄存器，其中 N 是 TIMx_REPCNT.REPCNT 的值。在向上计数模式下，每次计数器溢出，向下计数模式下每次计数器下溢或中央对齐模式下每次计数器溢出和每次计数器下溢时，重复计数器都会递减。设置 TIMx_EVTGEN.UDGN 位将重新加载 TIMx_REPCNT.REPCNT

位域	名称	描述
		的内容并生成更新事件。

### 12.4.22 刹车和死区寄存器 (TIMx\_BKDT)

偏移地址: 0x50

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCKCFG[1:0]		OSSR	OSSI	BKEN	BKP	AOEN	MOEN	DTGN[7:0]							
rw		rw	rw	rw	rw	rw	rw	rw							

注释: 根据锁定设置, AOEN、BKP、BKEN、OSSI、OSSR 和 DTGN[7:0] 位均可被写保护, 有必要在第一次写入 TIMx\_BKDT 寄存器时对它们进行配置。

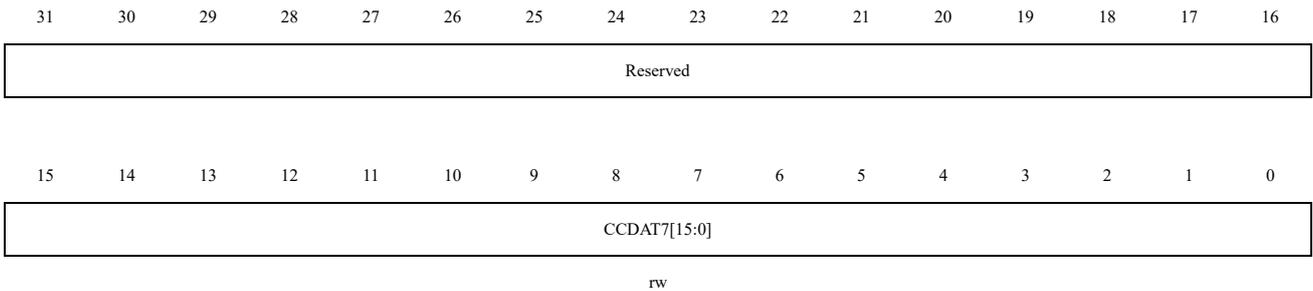
位域	名称	描述
31:16	Reserved	保留, 必须保持复位值
15:14	LCKCFG[1:0]	<p>锁定设置 (Lock configuration) 该位为防止软件错误而提供写保护。这些位提供针对软件错误的写保护。</p> <p>00: – 没有写保护。</p> <p>01: – 锁定级别 1 TIMx_BKDT.DTGN、TIMx_BKDT.BKEN、TIMx_BKDT.BKP、TIMx_BKDT.AOEN、TIMx_CTRL2.OIx、TIMx_CTRL2.OIxN、TIMx_AF1位启用写保护。</p> <p>10: – 锁定 2 级 除了 LOCK Level 1 模式下的寄存器写保护外, TIMx_CCEN.CCxP 和 TIMx_CCEN.CCxNP (如果相应通道配置为输出模式), TIMx_BKDT.OSSR 和 TIMx_BKDT.OSSI 位也能写保护。</p> <p>11: – 锁定 3 级 除了 LOCK Level 2 中的寄存器写保护外, TIMx_CCMODx.OCxMD 和 TIMx_CCMODx.OCxPEN 位 (如果相应通道配置为输出模式) 也启用写保护。 注意: 定时器复位后, LCKCFG 位只能写一次。一旦写入 TIMx_BKDT 寄存器, LCKCFG 将受到保护, 直到下一次复位。</p>

13	OSSR	<p>当 TIMx_BKDT.MOEN=1 且通道为互补输出时使用该位。</p> <p>没有互补输出的定时器中不存在 OSSR 位。</p> <p>0: 当定时器不工作时, 禁止OC/OCN输出 (OC/OCN使能输出信号=0);</p> <p>1: 当定时器不工作时, 一旦CCxEN=1或CCxNEN=1, 首先开启OC/OCN并输出无效电平, 然后置OC/OCN使能输出信号=1。</p> <p>有关更多详细信息, 请参见第12.4.11节, 捕获/比较使能寄存器 (TIMx_CCEN)。</p>
12	OSSI	<p>空闲模式下“关闭状态”选择 (Off-state selection for Idle mode)</p> <p>当 TIMx_BKDT.MOEN=0 且通道配置为输出时使用该位。</p> <p>0: 当定时器不工作时, 禁止OC/OCN输出 (OC/OCN使能输出信号=0);</p> <p>1: 当定时器不工作时, 一旦CCxEN=1 或CCxNEN=1, OC/OCN首先输出其空闲电平, 然后OC/OCN使能输出信号=1。</p> <p>有关更多详细信息, 请参见第12.4.11节, 捕获/比较启用寄存器 (TIMx_CCEN)。</p>
11	BKEN	<p>刹车功能使能 (Break enable)</p> <p>0: 禁止刹车输入 (BRK及CCS时钟失效事件);</p> <p>1: 开启刹车输入 (BRK及CCS时钟失效事件)。</p> <p><i>注: 任何对该位的写操作都需要一个APB时钟的延迟以后才能起作用。</i></p>
10	BKP	<p>刹车输入极性 (Break polarity)</p> <p>0: 刹车输入低电平有效;</p> <p>1: 刹车输入高电平有效。</p> <p><i>注: 任何对该位的写操作都需要一个APB时钟的延迟以后才能起作用。</i></p>
9	AOEN	<p>自动输出使能 (Automatic output enable)</p> <p>0: 只有软件可以设置TIMx_BKDT.MOEN;</p> <p>1: 软件设置TIMx_BKDT.MOEN; 或者如果刹车输入未激活, 则在下一次更新事件发生时, 硬件自动设置 TIMx_BKDT.MOEN。</p>
8	MOEN	<p>主输出使能 (Main output enable)</p> <p>该位可由软件或硬件根据 TIMx_BKDT.AOEN 位设置, 一旦刹车输入有效, 该位由硬件异步清零。它仅对配置为输出的通道有效。</p> <p>0: OC 和 OCN 输出被禁用或强制进入空闲状态。</p> <p>1: 如果设置了 TIMx_CCEN.CCxEN 或 TIMx_CCEN.CCxNEN 位, 则使能 OC 和 OCN 输出。有关更多详细信息, 请参见第 12.4.11节捕获/比较使能寄存器 (TIMx_CCEN)。</p>
7:0	DTGN[7:0]	<p>死区发生器设置 (Dead-time generator setup)</p> <p>这些位定义插入的互补输出之间的死区持续时间。DTGN值与死区时间的关系如下:</p> <p><math>DTGN[7:5]=0xx \Rightarrow DT=DTGN[7:0] \times T_{dtgn}, T_{dtgn} = T_{DTS};</math></p> <p><math>DTGN[7:5]=10x \Rightarrow DT=(64+DTGN[5:0]) \times T_{dtgn}, T_{dtgn} = 2 \times T_{DTS};</math></p> <p><math>DTGN[7:5]=110 \Rightarrow DT=(32+DTGN[4:0]) \times T_{dtgn}, T_{dtgn} = 8 \times T_{DTS};</math></p> <p><math>DTGN[7:5]=111 \Rightarrow DT=(32+DTGN[4:0]) \times T_{dtgn}, T_{dtgn} = 16 \times T_{DTS};</math></p>

### 12.4.23 捕获/比较寄存器 7 (TIMx\_CCDA7)

偏移地址: 0x54

复位值: 0x0000 0000



位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:0	CCDAT7[15:0]	捕获/比较通道7的值（Capture/Compare 7 value） <ul style="list-style-type: none"> <li>■ CC7 通道只能配置为输出：</li> </ul> CCDAT7 包含要与计数器 TIMx_CNT 比较的值，在 OC7 输出上发出信号。 如果未在 TIMx_CCMOD3.OC7PEN 位中选择预加载功能，则写入的值会立即传输到有效寄存器。否则，仅当更新事件发生时，此预加载值才会传输到活动寄存器。

## 12.4.24 刹车滤波寄存器（TIMx\_BKFR）

偏移地址：0x60

复位值：0x0000 0000



位域	名称	描述
31:30	Reserved	保留，必须保持复位值
29:24	THRESH[5:0]	采样逻辑电平有效的阈值数(Threshold)，最大 63： 有效逻辑电平的阈值。在采样窗口内，如果逻辑高的数量大于或等于阈值，则下一个逻辑电平将为逻辑高。同样的规则适用于逻辑低。如果窗口内 1 和 0 的数量都小于阈值，则过滤器输出保持不变。阈值应设置为大于或等于 Window 值的一半。 推荐阈值范围为： <b>最小值：</b> 比最大毛刺大小的上限（预分频时钟周期）多 1 个预分频时钟周期，并且需要大于窗口大小的一半。 例如，如果毛刺大小为 3.2*（预分频时钟周期），则阈值应为 $\lceil 3.2 \rceil = 4 + 1 = 5$ <b>最大值：</b> 有效信号最小尺寸的底值（在预分频时钟周期内），需要小于窗口尺寸。

		例如，如果最小信号大小为 $3.2 * (\text{预分频时钟周期})$ ，则阈值应为下限 $(3.2) = 3$ 。
23	Reserved	保留，必须保持复位值
22:17	WSIZE[5:0]	逻辑电平检查的窗口大小值（Window size），最大 63： 窗口大小决定了在获得下一个逻辑级别时将考虑多少采样值。内置 FIFO 为 64 位，最大索引为 63，只能将窗口大小设置为 63。
16	FILTEN	滤波器使能（Filter enable）： 0: 滤波器禁能 1: 滤波器使能
15:0	Reserved	保留，必须保持复位值

### 12.4.25 复用功能寄存器 1（TIMx\_AF1）

偏移地址：0x7C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										IOM4 BRKP	IOM3 BRKP	IOM2 BRKP	IOM4 BRKEN	IOM3 BRKEN	IOM2 BRKEN
										rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					COMP1 BRKP	IOM BRKP	Reserved					COMP1 BRKEN	IOM BRKEN		
					rw	rw						rw	rw		

位域	名称	描述
31:22	Reserved	保留，必须保持复位值
21	IOM4BRKP	TIMx_BKIN IO4刹车输入极性选择 0: TIMx_BKIN IO4刹车输入极性不翻转（如果BKP=0,则低电平有效；如果BKP=1,则高电平有效） 1: TIMx_BKIN IO4刹车输入极性翻转（如果BKP=0,则高电平有效；如果BKP=1,则低电平有效）
20	IOM3BRKP	TIMx_BKIN IO3刹车输入极性选择 0: TIMx_BKIN IO3刹车输入极性不翻转（如果BKP=0,则低电平有效；如果BKP=1,则高电平有效） 1: TIMx_BKIN IO3刹车输入极性翻转（如果BKP=0,则高电平有效；如果BKP=1,则低电平有效）
19	IOM2BRKP	TIMx_BKIN IO2刹车输入极性选择 0: TIMx_BKIN IO2刹车输入极性不翻转（如果BKP=0,则低电平有效；如果BKP=1,则高电平有效） 1: TIMx_BKIN IO2刹车输入极性翻转（如果BKP=0,则高电平有效；如果BKP=1,则低电平有效）
18	IOM4BRKEN	TIMx_BKIN IO4刹车输入使能

		0: TIMx_BKIN IO4刹车输入禁止 1: TIMx_BKIN IO4刹车输入使能
17	IOM3BRKEN	TIMx_BKIN IO3刹车输入使能 0: TIMx_BKIN IO3刹车输入禁止 1: TIMx_BKIN IO3刹车输入使能
16	IOM2BRKEN	TIMx_BKIN IO2刹车输入使能 0: TIMx_BKIN IO2刹车输入禁止 1: TIMx_BKIN IO2刹车输入使能
15:11	Reserved	保留, 必须保持复位值
10	COMP1BRKP	tim_brk_comp1 刹车输入极性选择 0: tim_brk_comp1 刹车输入极性不翻转 (如果BKP=0,则低电平有效; 如果BKP=1,则高电平有效) 1: tim_brk_comp1 刹车输入极性翻转 (如果BKP=0,则高电平有效; 如果BKP=1,则低电平有效)
9	IOMBRKP	TIMx_BKIN IO1 刹车输入极性选择 0: TIMx_BKIN IO1 刹车输入极性不翻转 (如果BKP=0,则低电平有效; 如果BKP=1,则高电平有效) 1: TIMx_BKIN IO1 刹车输入极性翻转 (如果BKP=0,则高电平有效; 如果BKP=1,则低电平有效)
8:2	Reserved	保留, 必须保持复位值
1	COMP1BRKEN	tim_brk_comp1 刹车输入使能 0: tim_brk_comp1 刹车输入禁止 1: tim_brk_comp1 刹车输入使能
0	IOMBRKEN	TIMx_BKIN IO1 刹车输入使能 0: TIMx_BKIN IO1 刹车输入禁止 1: TIMx_BKIN IO1 刹车输入使能

## 12.4.26 DMA 控制寄存器 (TIMx\_DCTRL)

偏移地址: 0x94

复位值: 0x0000 0000

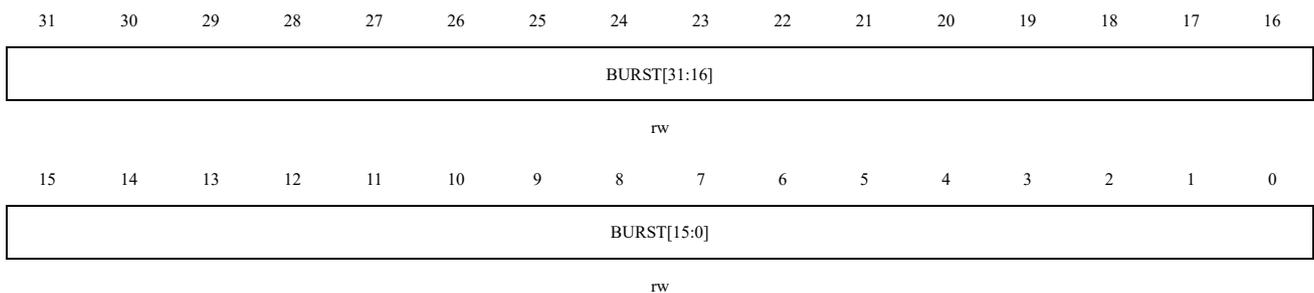
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DBADDR[5:0]				Reserved				DBLEN[5:0]			
rw								rw							

位域	名称	描述
31:14	Reserved	保留，必须保持复位值
13:8	DBADDR[5:0]	DMA基地址（DMA base address） 该位字段定义 DMA 访问 TIMx_DADDR 寄存器的第一个地址。 当第一次通过 TIMx_DADDR 完成访问时，该位域指定您刚刚访问的地址。然后第二次访问TIMx_DADDR，会访问到“DMA Base Address + 4”的地址 000000: TIMx_CTRL1, 000001: TIMx_CTRL2, 000010: TIMx_STS, ..... 100101: TIMx_DCTRL
7:6	Reserved	保留，必须保持复位值
5:0	DBLEN[5:0]	DMA连续传送长度（DMA burst length） 该位字段定义 DMA 将访问（写入/读取）TIMx_DADDR 寄存器的次数。 000000: 1次传输 000001: 2次传输 000010: 3次传输 ... 010001: 18次传输 ... 100010: 35次传输

### 12.4.27 连续模式的 DMA 地址（TIMx\_DADDR）

偏移地址：0x98

复位值：0x0000



位域	名称	描述
31:0	BURST[31:0]	DMA 访问缓冲区。 当对该寄存器分配读或写操作时，将访问位于地址范围（DMA base address + DMA burst length × 4）的寄存器。 DMA base address = The address of TIM_CTRL1 + TIMx_DCTRL.DBADDR * 4; DMA burst len = TIMx_DCTRL.DBLEN + 1. 例子:

位域	名称	描述
		<p>如果 <code>TIMx_DCTRL.DBLEN = 0x3</code> (4 次传输), <code>TIMx_DCTRL.DBADDR = 0xD</code> (<code>TIMx_CC DAT1</code>), DMA 数据长度 = 半字, DMA 存储器地址 = SRAM 中的缓冲区地址, DMA 外设地址 = <code>TIMx_DADDR</code> 地址。</p> <p>当事件发生时, <code>TIMx</code> 将向 DMA 发送请求, 并传输 4 次数据。</p> <p>第一次, 对 <code>TIMx_DADDR</code> 寄存器的 DMA 访问将映射到访问 <code>TIMx_CC DAT1</code> 寄存器:</p> <p>第二次, 对 <code>TIMx_DADDR</code> 寄存器的 DMA 访问将映射到访问 <code>TIMx_CC DAT2</code> 寄存器:</p> <p>.....</p> <p>第四次, 对 <code>TIMx_DADDR</code> 寄存器的 DMA 访问将映射到访问 <code>TIMx_CC DAT4</code> 寄存器:</p>

## 13 通用定时器（TIM3）

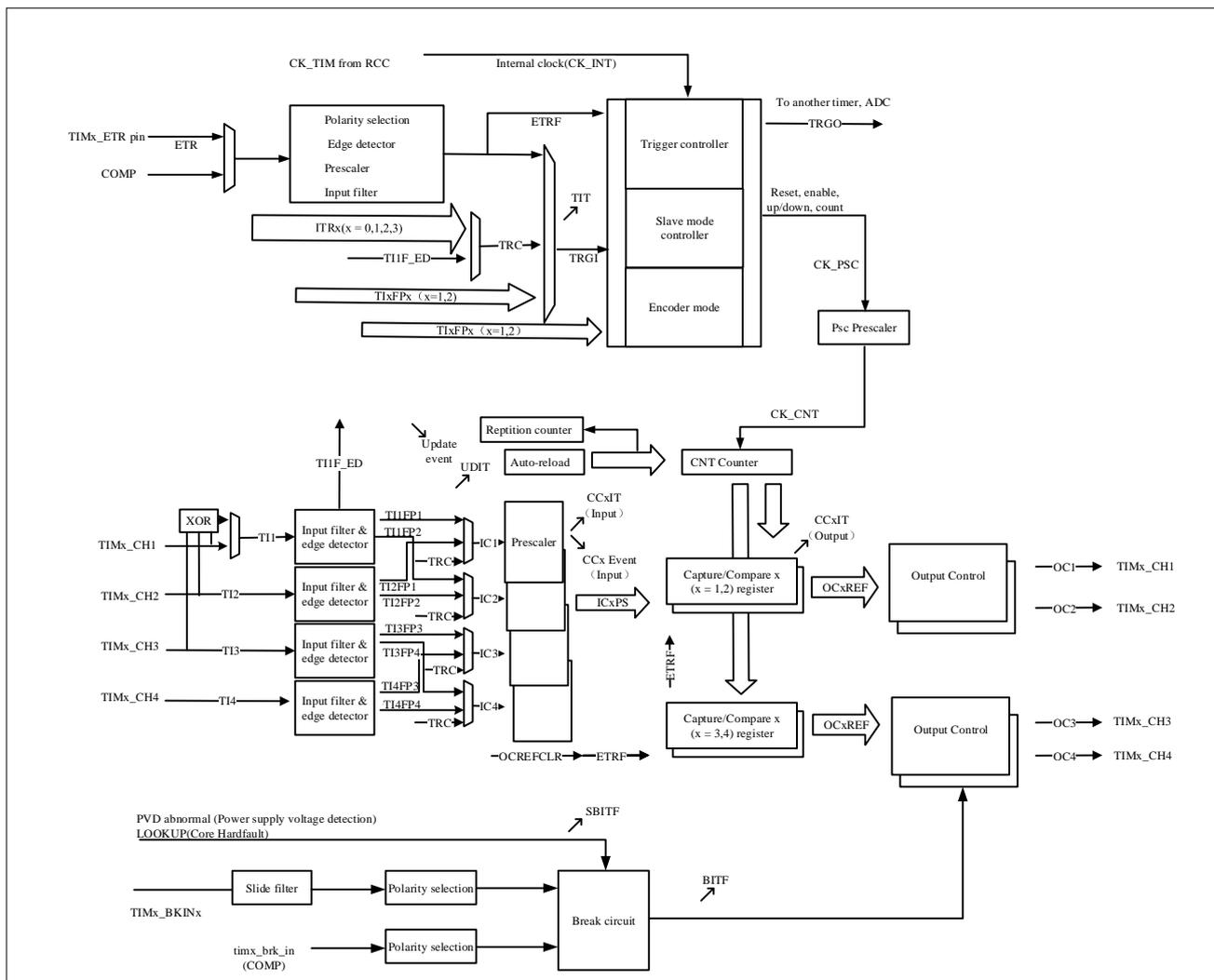
### 13.1 TIM3 简介

通用定时器（TIM3）主要用于以下场合：对输入信号进行计数、测量输入信号的脉冲宽度和产生输出波形等。

### 13.2 TIM3 主要特性

- 16 位自动装载计数器。（可实现向上计数、向下计数、向上/下计数）。
- 16 位可编程预分频器。（分频系数可配置为 1 到 65536 之间的任意值）
- TIM3 最多支持 4 个通道
- 通道工作模式：PWM 输出、输出比较、单脉冲模式输出、输入捕获
- 如下事件发生时产生中断/DMA：
  - ◆ 更新事件
  - ◆ 触发事件
  - ◆ 输入捕获
  - ◆ 输出比较
- 可通过外部信号控制定时器
- 多个定时器内部连接在一起，以实现定时器的同步或链接
- 增量（正交）编码器接口：用于追踪运行轨迹和解析旋转方位
- 霍尔传感器接口：用于三相电机控制
- 支持捕获内部比较器输出信号。

图 13-1 TIM3 框图



↓ 事件                      ↗ 中断和 DMA 输出

捕获通道 1 输入可以来自 IOM 或比较器输出

## 13.3 TIM3 功能描述

### 13.3.1 时基单元

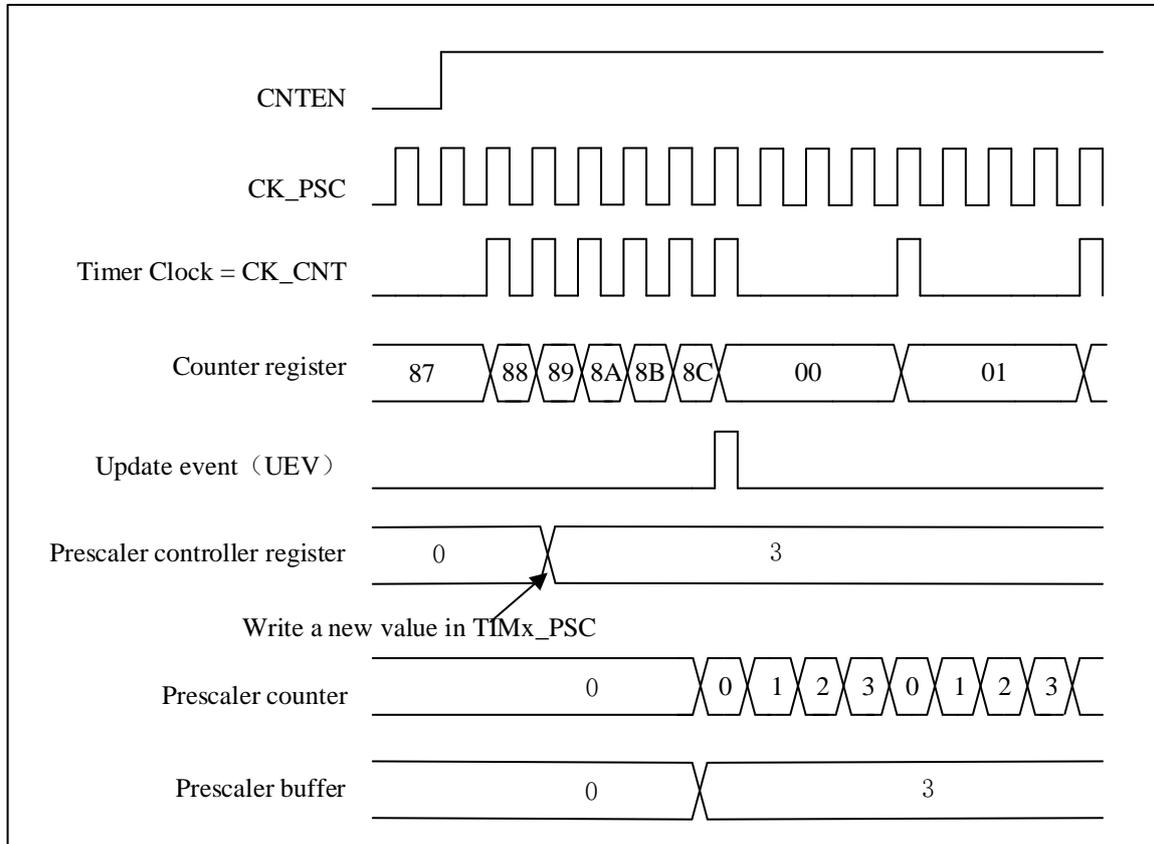
通用器的时基单元主要包括：预分频器、计数器和自动重载寄存器。当时基单元工作时，软件可以随时读取和写入相应的寄存器（TIMx\_PSC、TIMx\_CNT 和 TIMx\_AR）。

根据自动重载预装载使能位（TIMx\_CTRL1.ARPEN）的设置，预装载寄存器的值会立即或在每次更新事件 UEV 时传输到影子寄存器。TIMx\_CTRL1.UPDIS=0 时，当计数器上溢/下溢或软件设置 TIMx\_EVTGEN.UDGN，将生成更新事件。计数器 CK\_CNT 仅在 TIMx\_CTRL1.CNTEN 位被设置时有效。计数器在 TIMx\_CTRL.CNTEN 位被设置后一个时钟周期之后开始计数。

### 13.3.1.1 预分频器描述

TIMx\_PSC 寄存器由一个 16 位计数器组成，可用于计数器时钟频率按 1 和 65536 之间的任意分频。因为这个控制器带有缓冲器，可以在运行时动态改变。新的预分频器值只有在下次更新事件中才会被采用。

图 13-2 当预分频的参数从 1 到 4，计数器的时序图



## 13.3.2 计数器模式

### 13.3.2.1 向上计数模式

使用向上计数模式，计数器将从 0 计数到寄存器 TIMx\_AR 的值，然后重置为 0。并产生一个计数器溢出事件。

如果设置 TIMx\_CTRL1.UPRS 位(选择更新请求)和 TIMx\_EVTGEN.UDGN 位，将产生一个更新事件(UEV)。但是 TIMx\_STS.UDITF 不会被硬件置起，因此不会产生更新中断或 DMA 更新请求。这是为了避免清除计数器时产生更新中断。

取决于 TIMx\_CTRL1.UPRS 的配置，当发生更新事件时，TIMx\_STS.UDITF 被设置，所有寄存器都会更新：

- 当 TIMx\_CTRL1.ARPEN = 1，预装载寄存器(TIMx\_AR)的值被更新到自动装载影子寄存器
- 预加载值 (TIMx\_PSC) 被重新加载到预分频器影子寄存器中

为了避免在向预装载寄存器中写入新值时更新影子寄存器，可以通过设置 TIMx\_CTRL1.UPDIS=1 来禁止更新事件。

当产生一个更新事件时，计数器仍将被清除，预分频器计数器也将被设置为 0（但预分频器值将保持不变）。  
 下图给出一些示例，展示了向上计数模式计数器在不同分频因子下的动作。

图 13-3 当内部时钟分频因子 = 2/N 时，向上计数的时序图

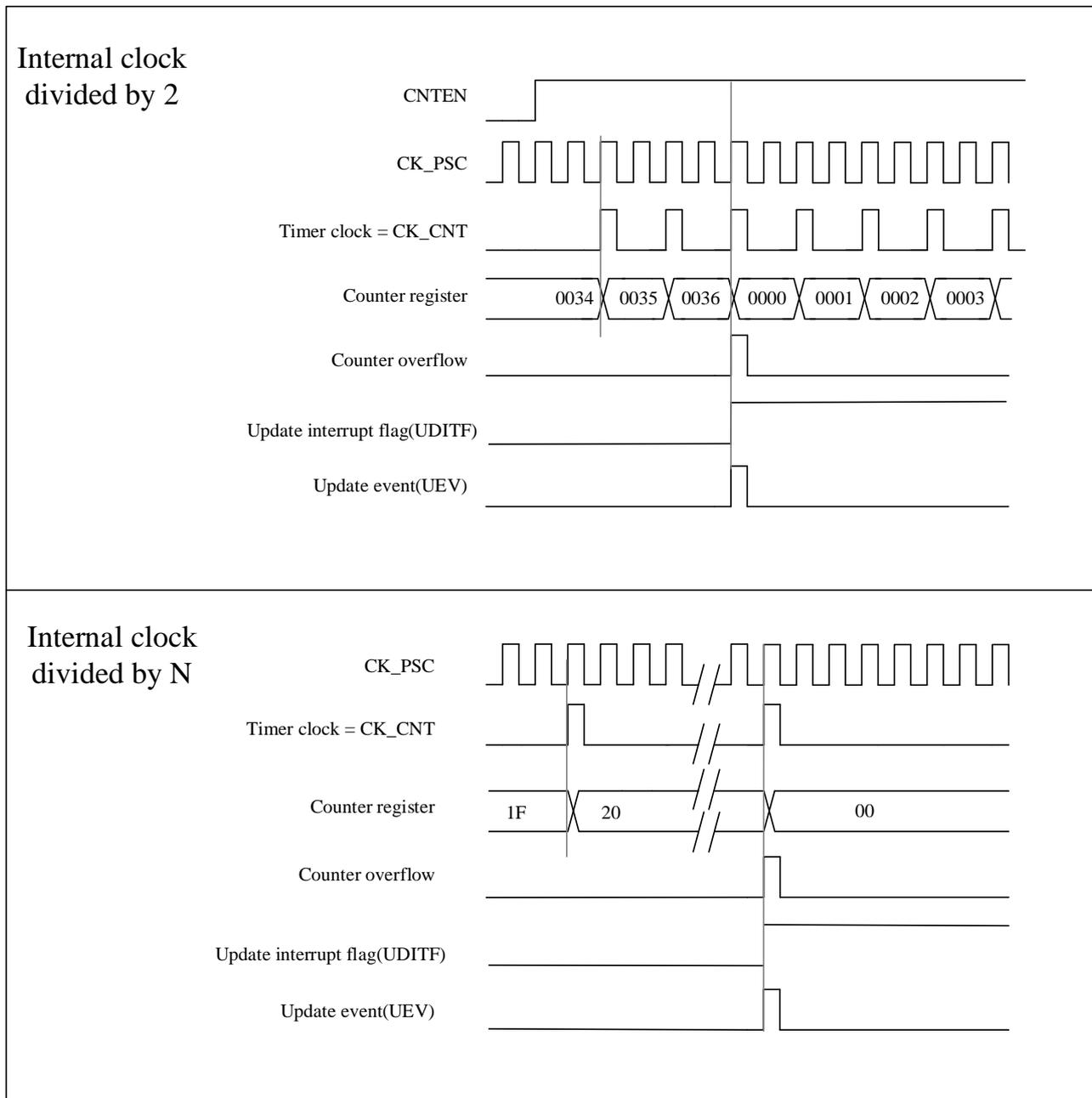
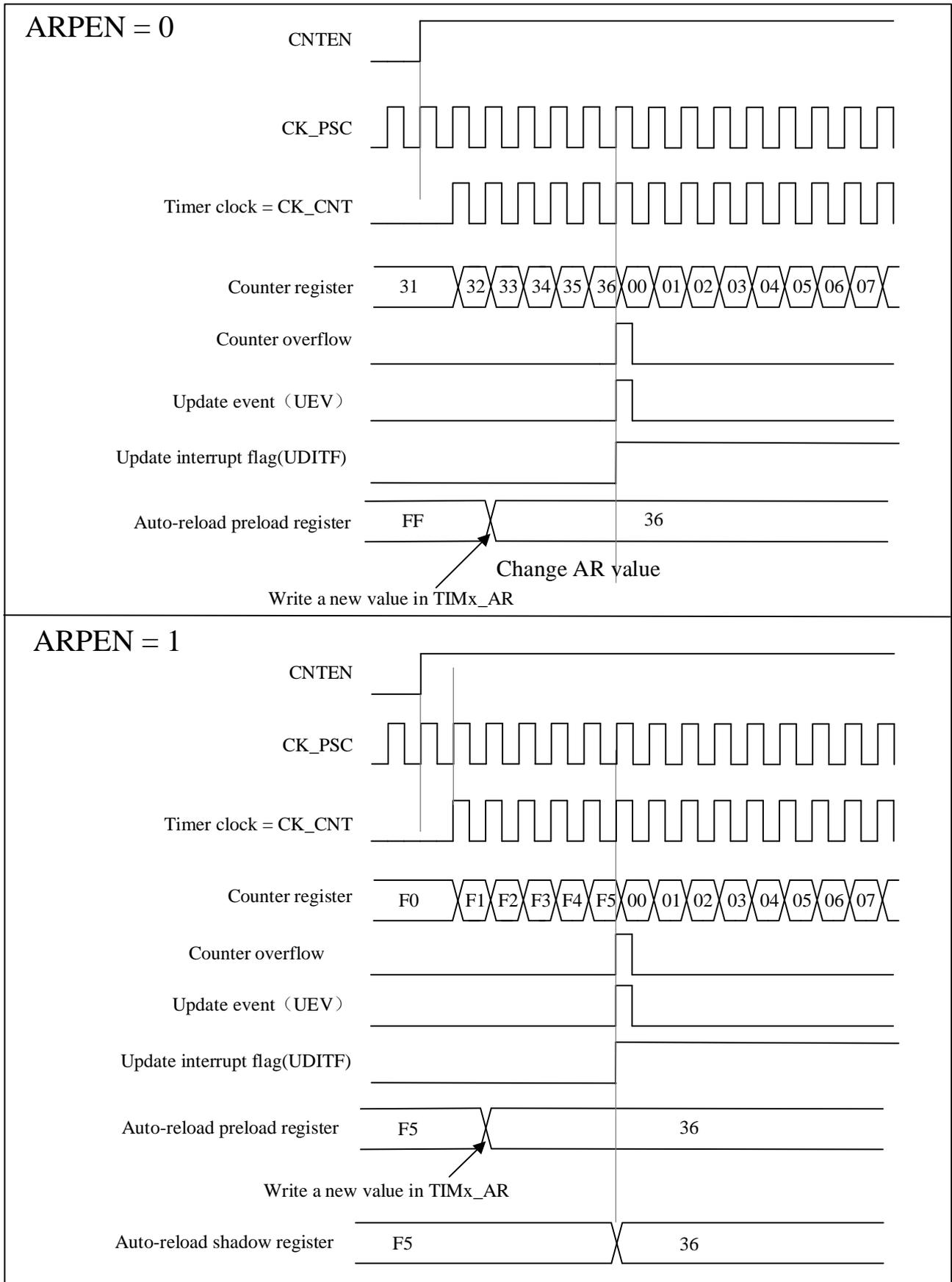


图 13-4 当 ARPEN=0/1 产生更新事件时，向上计数的时序图



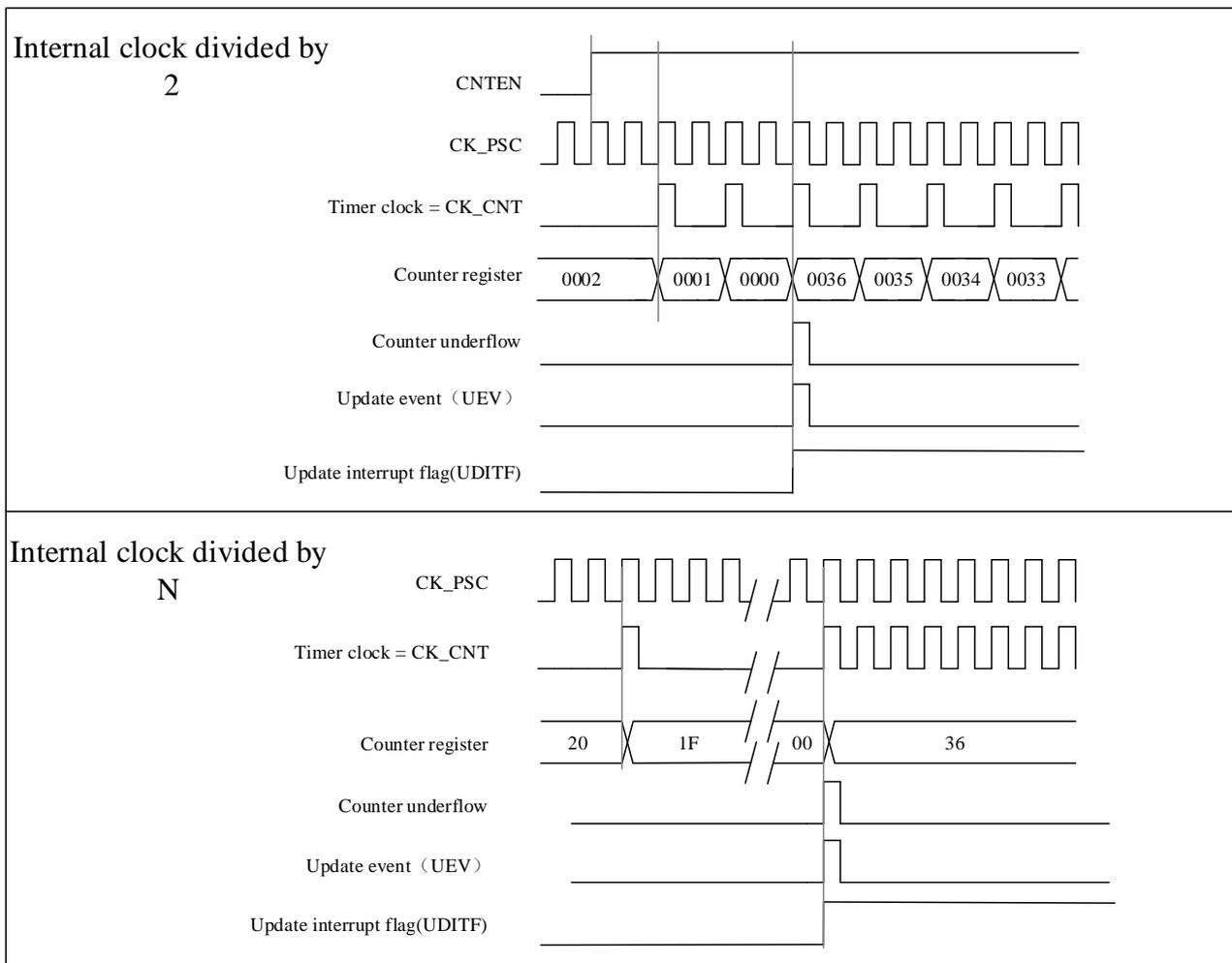
### 13.3.2.2 向下计数模式

向下计数模式，计数器将从寄存器 TIMx\_AR 的值减至 0，然后从自动重载值重新开始，并产生计数器向下溢出事件

向下计数模式和向上计数模式配置更新事件和更新寄存器的过程相同，请查阅 13.3.2.1 章节。

下图给出一些示例，展示了向下计数模式计数器在不同分频因子下的动作。

图 13-5 内部时钟分频因子 = 2/N 时，向下计数时序图



### 13.3.2.3 中央对齐模式

在中央对齐模式下，计数器从 0 增加到值 (TIMx\_AR) - 1，产生计数器溢出事件。然后，它从自动重载值 (TIMx\_AR) 向下计数到 1，并生成一个计数器向下溢出事件。然后计数器重置为 0 并再次开始计数。

在这种模式下，TIMx\_CTRL1.DIR 方向位无效，由硬件更新和指定当前计数方向。当 TIMx\_CTRL1.CAMSEL 位不等于“00”时，中央对齐模式有效。

每次计数上溢和计数下溢时都会生成更新事件。或者，也可以通过设置 TIMx\_EVTGEN.UDGN 位（通过软件或使用从模式控制器）来生成更新事件。在这种情况下，计数器从 0 重新开始计数，预分频器的计数器也从 0 重新开始计数。

注：如果因为计数器溢出而产生更新，自动重载将在计数器重新载入之前被更新。

图 13-6 内部时钟分频因子 = 2/N，中央对齐时序图

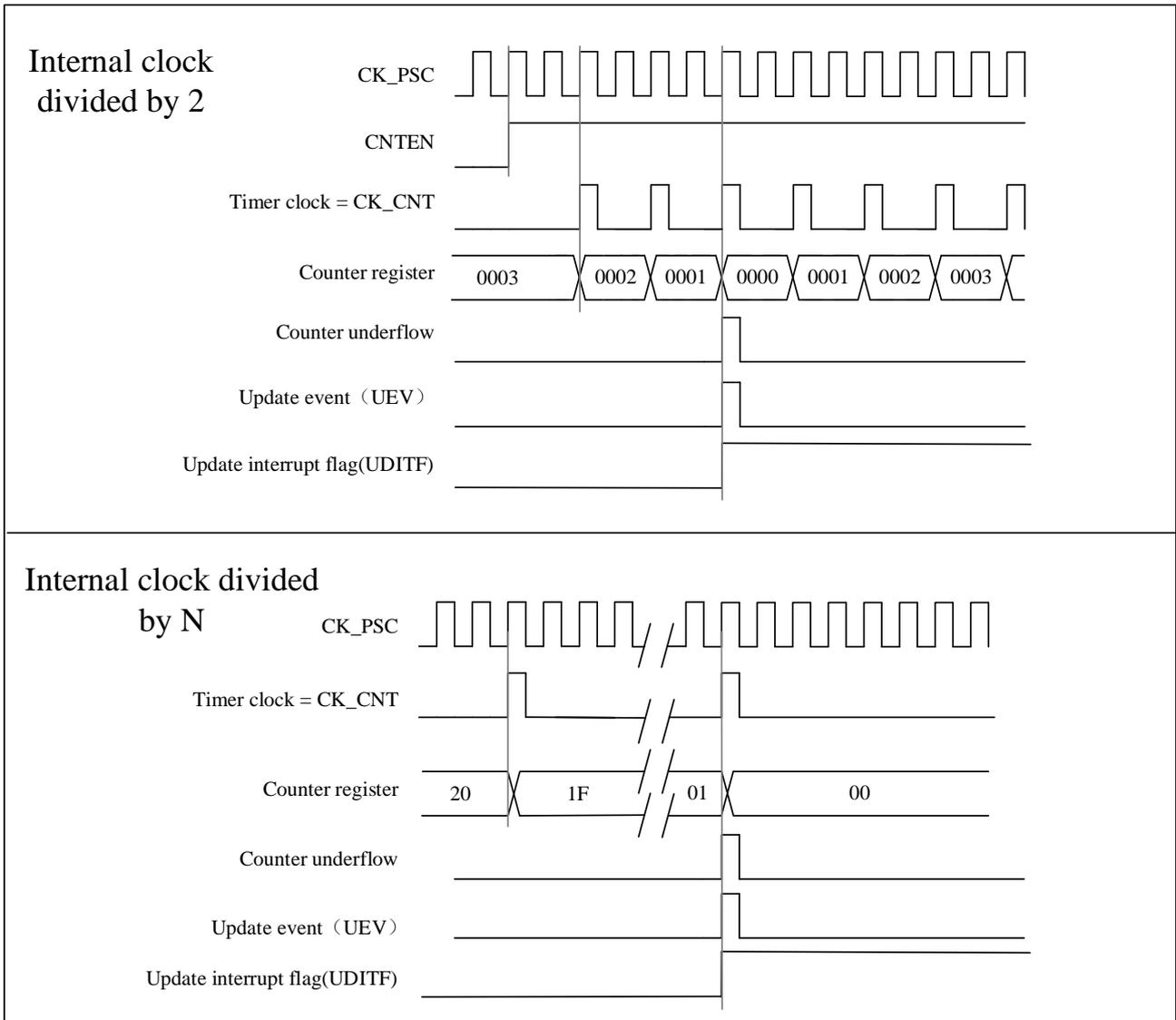
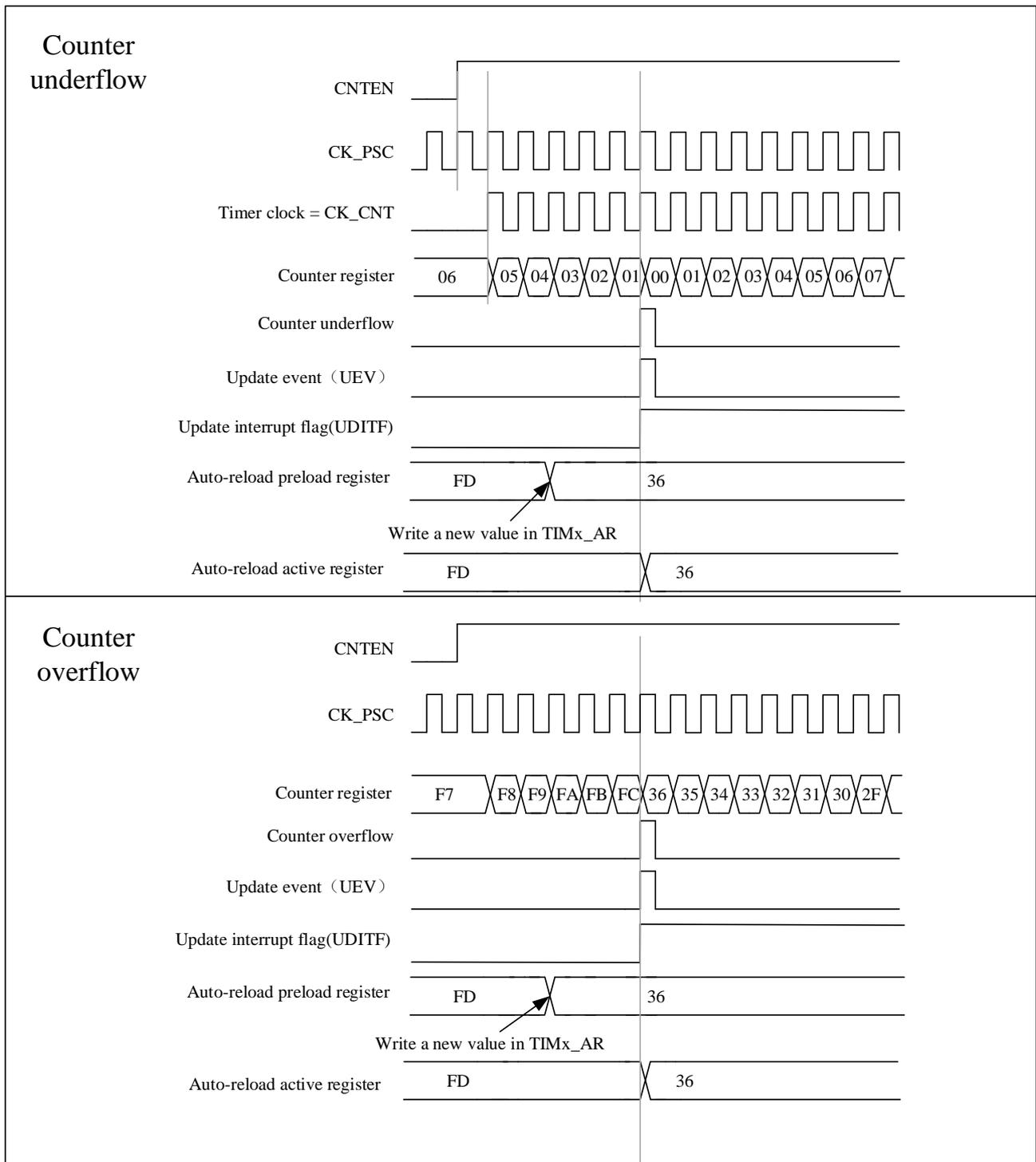


图 13-7 包含计数器上溢和下溢的中央对齐时序图(ARPEN=1)



### 13.3.3 时钟选择

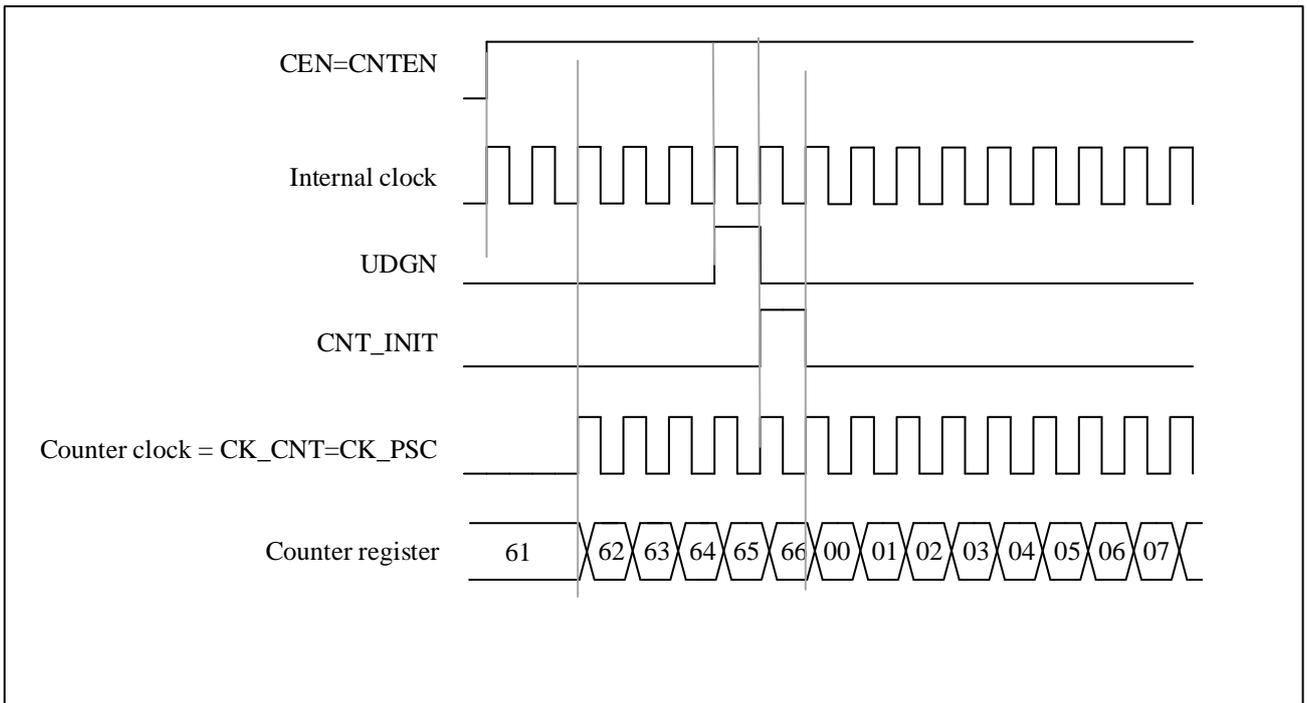
- 通用定时器的内部时钟：CK\_INT；
- 两种外部时钟模式：
  - 外部输入引脚

- 外部触发输入 ETR
- 内部触发输入 (ITRx): 一个定时器用作另一个定时器的预分频器

### 13.3.3.1 内部时钟源(CK\_INT)

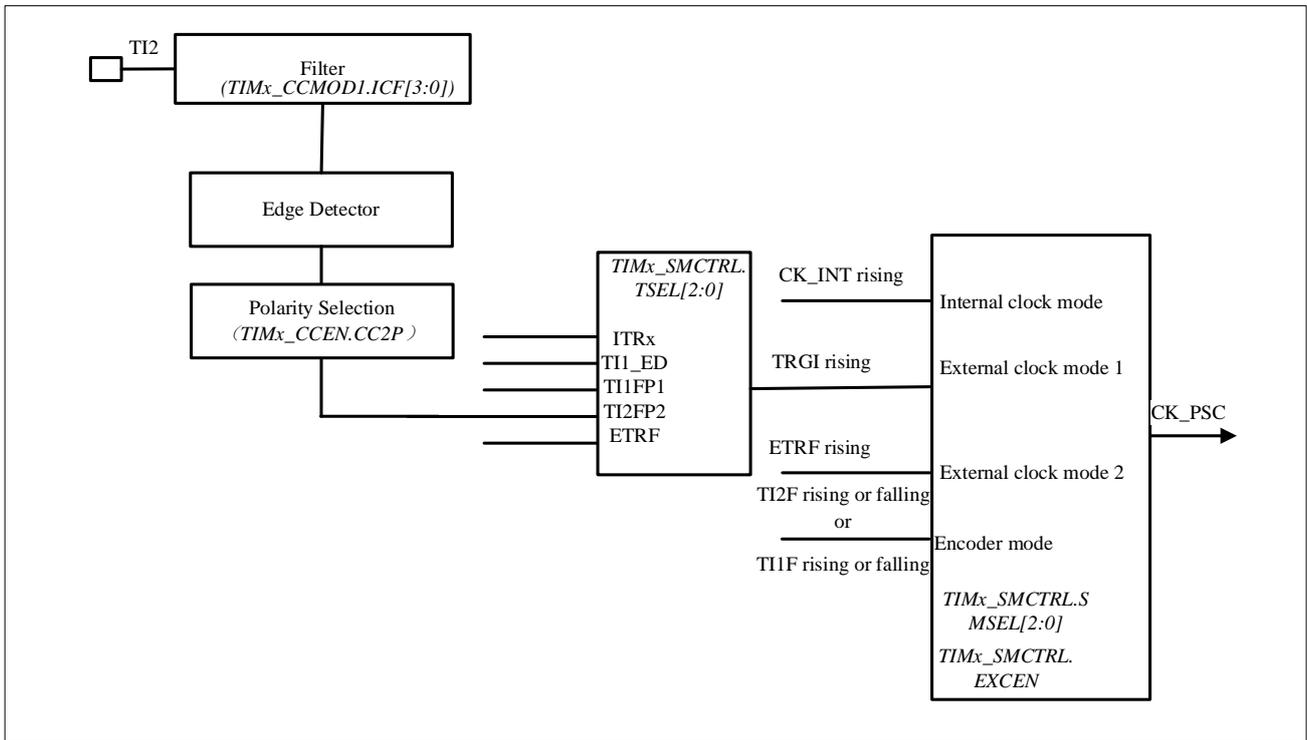
当 TIMx\_SMCTRL.SMSEL 等于“0000”时，从模式控制器被禁用。这三个控制位 (TIMx\_CTRL1.CNTEN、TIMx\_CTRL1.DIR、TIMx\_EVTGEN.UDGN) 只能由软件改变 (TIMx\_EVTGEN.UDGN 除外，它保持自动清零)。前提是 TIMx\_CTRL1.CNTEN 位被软写为'1'，预分频器的时钟源由内部时钟 CK\_INT 提供。

图 13-8 正常模式下的控制电路，内部时钟除以 1



### 13.3.3.2 外部时钟源模式 1

图 13-9 TI2 外部时钟连接示例



通过配置 `TIMx_SMCTRL.SMSEL=0111` 选择该模式。计数器可以配置为在所选输入的时钟上升沿或下降沿进行计数。

例如，配置向上计数模式在 `TI2` 输入的时钟上升沿计数，配置步骤如下：

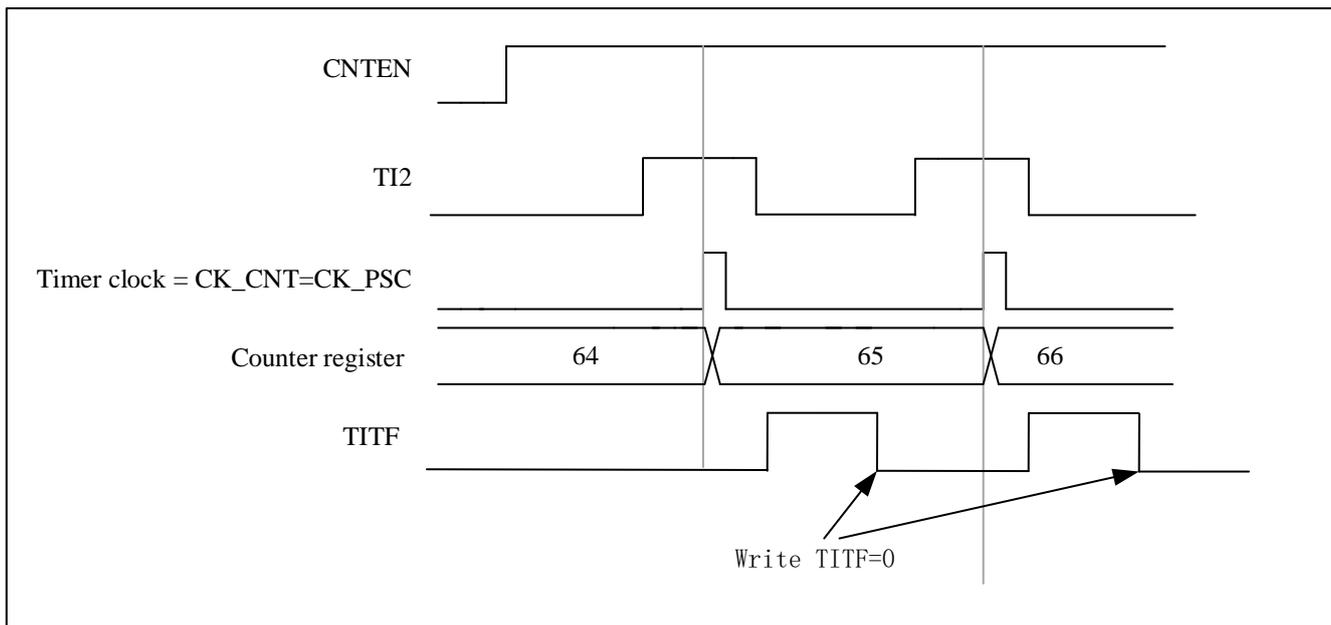
- 配置 `TIMx_CCMOD1.CC2SEL` 等于‘01’，`CC2` 通道配置为输入，`IC2` 映射到 `TI2`
- 配置 `TIMx_CCEN.CC2P` 等于‘0’，选择时钟上升沿极性
- 通过配置 `TIMx_CCMOD1.IC2F[3:0]` 选择输入滤波器带宽（如果不需要滤波器，保持 `IC2F` 位为‘0000’）
- 配置 `TIMx_SMCTRL.SMSEL` 等于‘0111’，选择定时器外部时钟模式 1
- 配置 `TIMx_SMCTRL.TSEL` 等于‘110’，选择 `TI2` 作为触发输入源
- 配置 `TIMx_CTRL1.CNTEN` 等于‘1’以启动计数器

注意：捕获预分频器不用于触发，所以不需要配置

当定时器时钟的上升沿出现在 `TI2=1` 时，计数器计数一次并且 `TIMx_STS.TITF` 标志被拉高。

`TI2` 的上升沿与计数器实际时钟之间的延迟取决于 `TI2` 输入端的再同步电路。

图 13-10 外部时钟模式 1 的控制电路

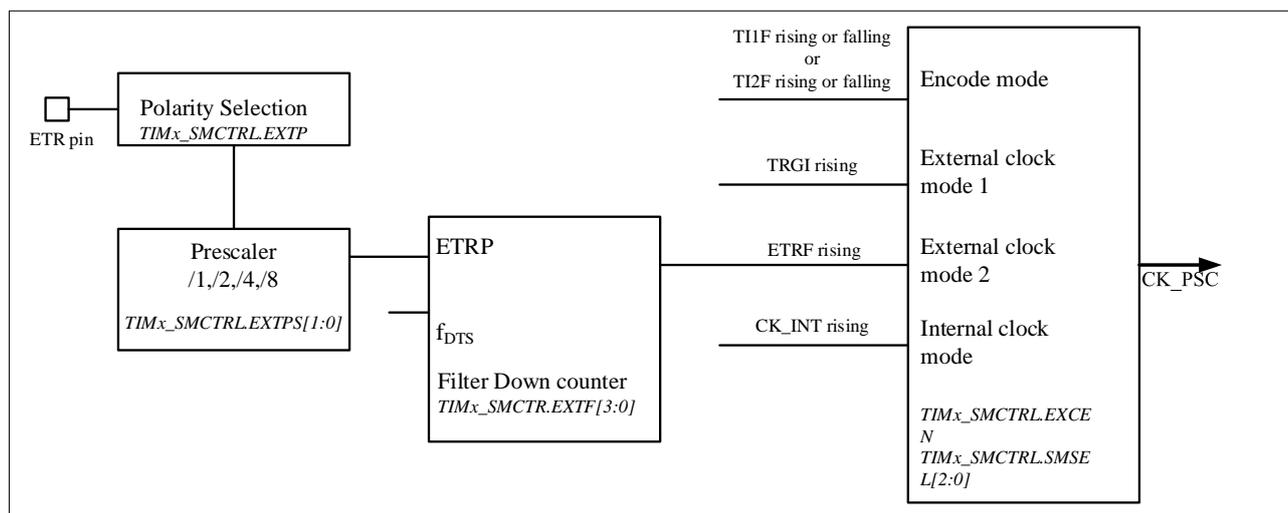


### 13.3.3.3 外部时钟源模式 2

此模式由 `TIMx_SMCTRL.EXCEN` 选择等于 1。计数器可以在外部触发输入 ETR 的每个上升沿或下降沿计数。

下图为外部时钟源模式 2 的外部触发输入模块示意图。

图 13-11 外部触发输入框图



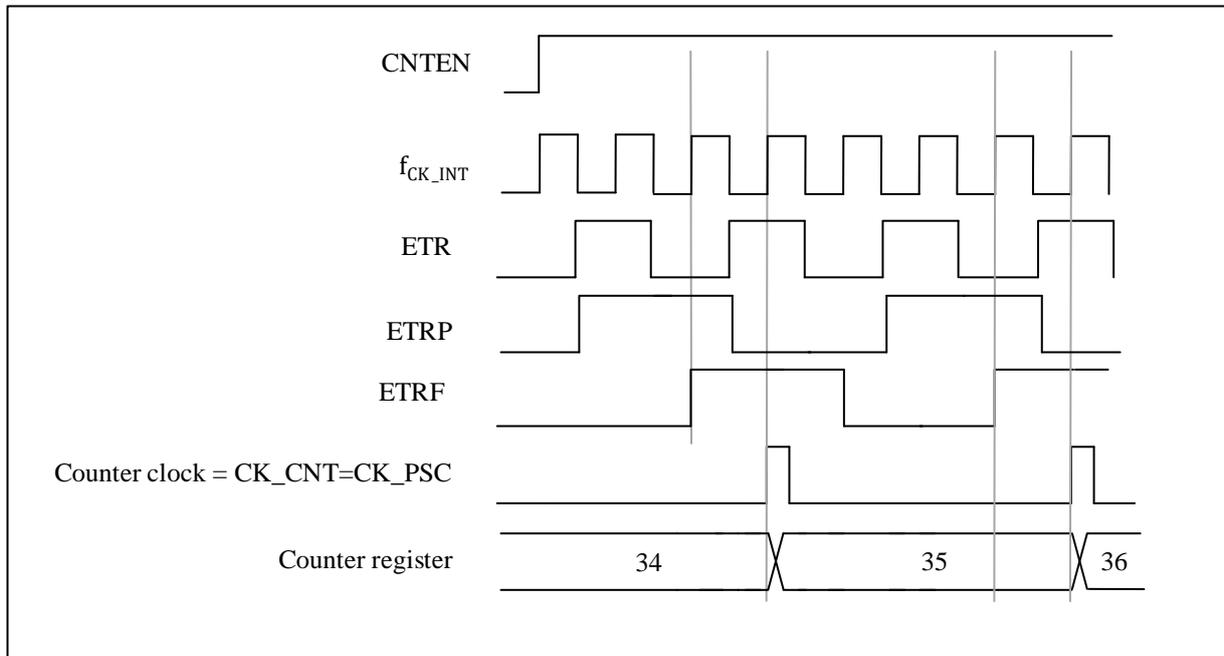
例如，使用以下配置步骤使向上计数器在 ETR 上每 2 个上升沿计数一次。

- 由于在这种情况下不需要过滤器，因此使 `TIMx_SMCTRL.EXTF[3:0]` 等于‘0000’
- 通过使 `TIMx_SMCTRL.EXTPS[1:0]` 等于 ‘01’ 来配置预分频器
- 通过设置 `TIMx_SMCTRL.EXTP` 等于‘0’来选择 ETR 引脚的极性，ETR 的上升沿有效

- 外部时钟模式 2 通过设置 TIMx\_SMCTRL.EXCEN 等于‘1’来选择
- 通过设置 TIMx\_CTRL1.CNTEN 等于“1”启动计数器。

计数器每 2 个 ETR 上升沿计数一次。ETR 的上升沿与计数器的实际时钟之间的延迟是由于 ETRP 信号上的再同步电路造成的。

图 13-12 外部时钟模式 2 的控制电路

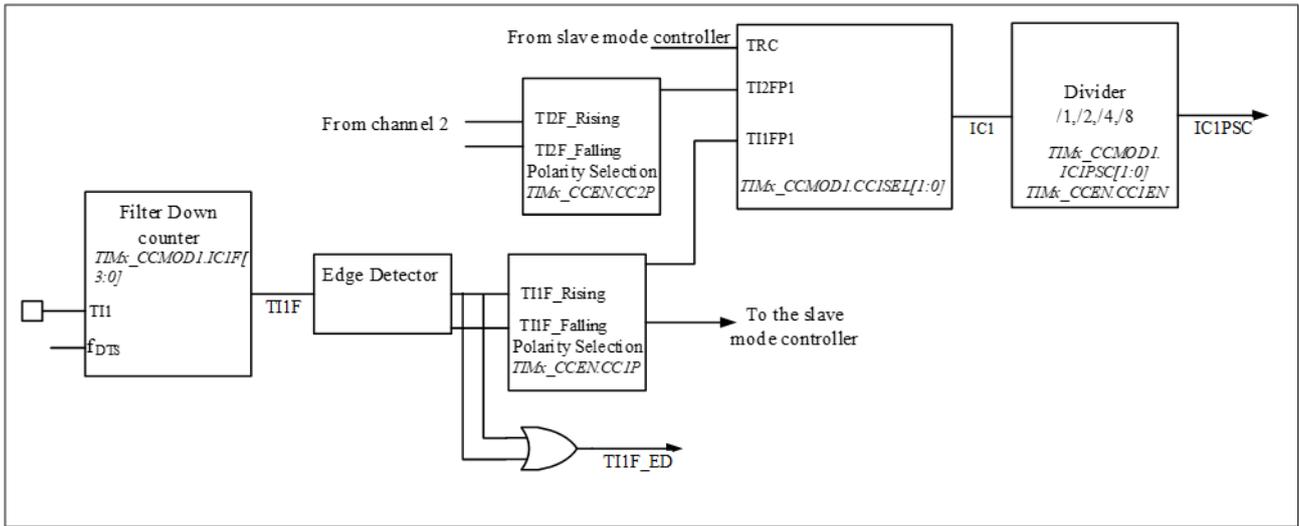


### 13.3.4 捕获/比较通道

捕获/比较通道包括捕获/比较寄存器和影子寄存器。输入部分由数字滤波器、多路复用器和预分频器组成。输出部分包括比较器和输出控制。

输入信号 TIX 被采样和滤波以产生信号 TIXF。然后由极性选择功能的边沿检测器生成信号 (TIXF\_rising 或 TIXF\_falling)，其极性由 TIMx\_CCEN.CCxP 位选择。该信号可用作从模式控制器的触发输入。同时，信号 ICx 经过分频后送入捕获寄存器。下图显示了捕获/比较通道的框图。

图 13-13 捕获/比较通道（例如：通道 1 输入级）



输出部分生成一个中间波形 OCxRef（高电平有效）作为参考。极性作用在链的末端。

图 13-14 捕获/比较通道 1 主电路

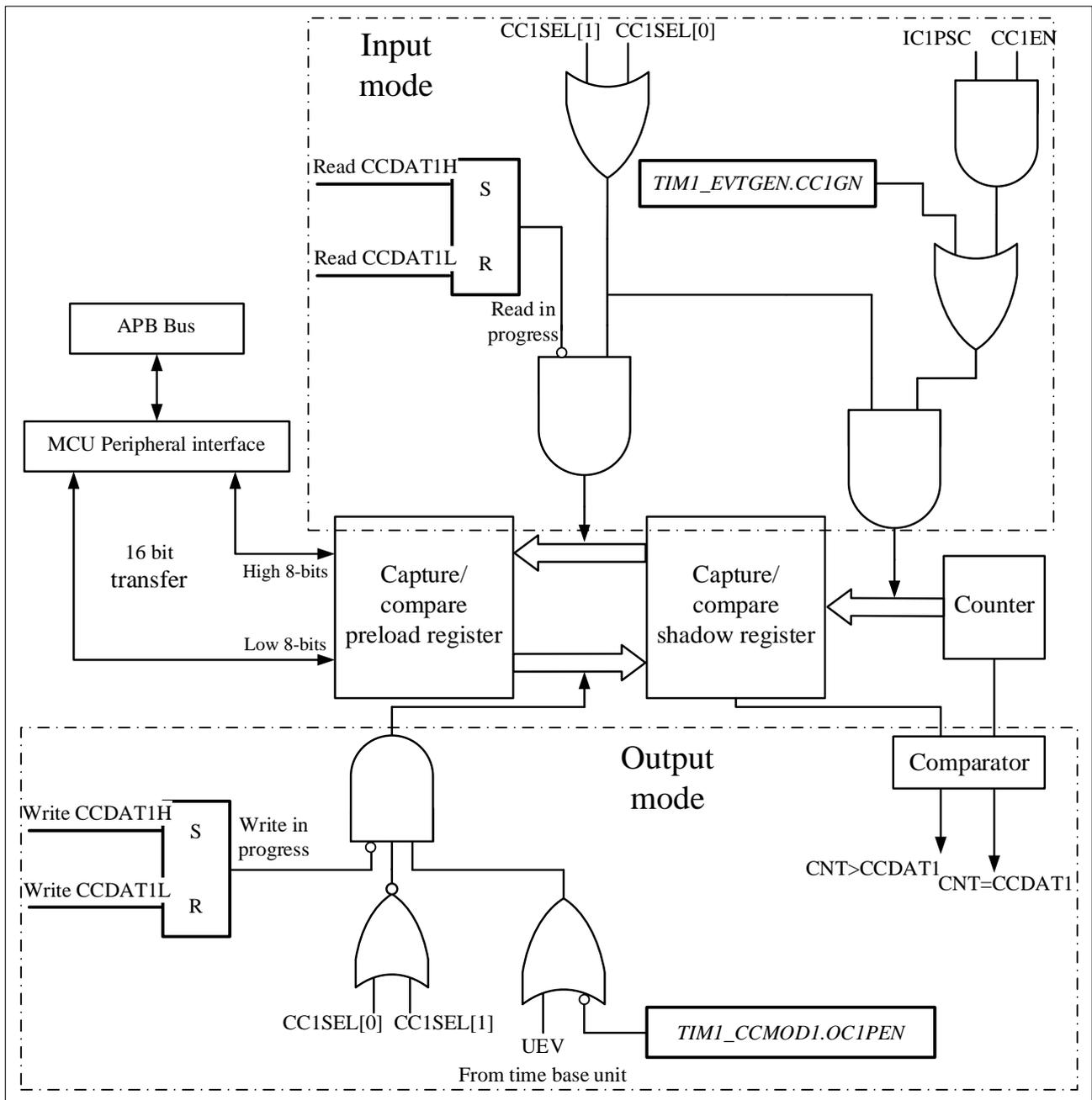
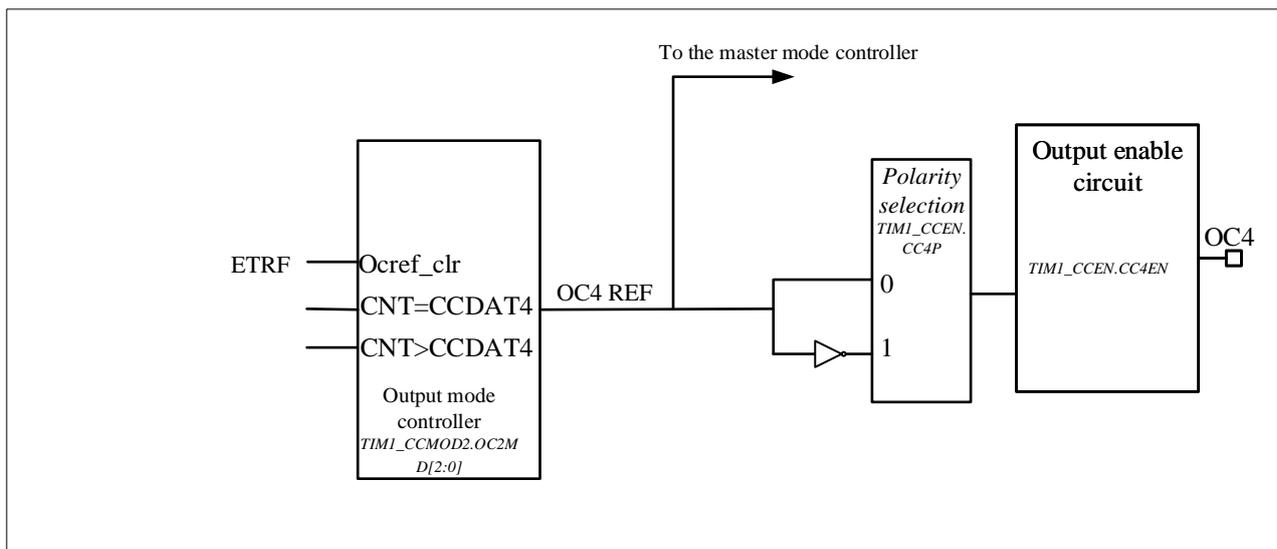


图 13-15 通道 x 的输出部分（以通道 4 为例子）



在捕获/比较时，读取和写入始终访问预加载的寄存器。两个具体工作流程如下：

在捕获模式下，捕获实际上是在影子寄存器中完成的，然后将影子寄存器中的值复制到预加载寄存器中。

在比较模式下，与捕获模式相反，预加载寄存器的值被复制到影子寄存器中，并与计数器进行比较。

### 13.3.5 输入捕获模式

在捕获模式下，TIMx\_CCxDATx 寄存器用于在检测到 ICx 信号后锁存计数器值。

有一个捕获中断标志 TIMx\_STS.CCxITF，如果相应的中断使能被拉高，它可以发出中断或 DMA 请求。

TIMx\_STS.CCxITF 位在发生捕获事件时由硬件设置，并由软件或读取 TIMx\_CCxDATx 寄存器清零。

当 TIMx\_CCxDATx 寄存器中的计数器值被捕获并且 TIMx\_STS.CCxITF 已经被拉高时，重复捕获标志 TIMx\_STS.CCxOCF 设置为 1。与前者不同，TIMx\_STS.CCxOCF 通过向其写入 0 来清除。

为实现 TI1 输入的上升沿将计数器值捕获到 TIMx\_CCxDAT1 寄存器中，配置流程如下：

■ 选择有效输入：

将 TIMx\_CCMOD1.CC1SEL 配置为“01”。此时输入为 CC1 通道，IC1 映射到 TI1。

■ 编程所需的输入滤波器持续时间：

通过配置 TIMx\_CCMODx.ICxF 位来定义 TI1 输入的采样频率和数字滤波器的长度。示例：如果输入信号抖动多达 5 个内部时钟周期，我们必须选择比这 5 个时钟周期更长的滤波器持续时间。当检测到具有新电平的 8 个连续样本（以  $f_{TIM4FILTCLK}$  频率采样）时，我们可以验证 TI1 上的转换。然后配置 TIMx\_CCMOD1.IC1F 到“1xxx”

■ 通过配置 TIMx\_CCEN.CC1P=0，选择上升沿作为 TI1 通道的有效跳变极性

■ 配置输入预分频器。在本例中，配置 TIMx\_CCMOD1.IC1PSC= ‘00’ 以禁用预分频器，因为我们想要捕获每个有效转换

■ 通过配置 TIMx\_CCEN.CC1EN = ‘1’ 启用捕获。

如果要使能 DMA 请求，可以配置 TIMx\_DINTEN.CC1DEN=1。如果要使能相关中断请求，可以配置 TIMx\_DINTEN.CC1IEN =1。

### 13.3.6 PWM 输入模式

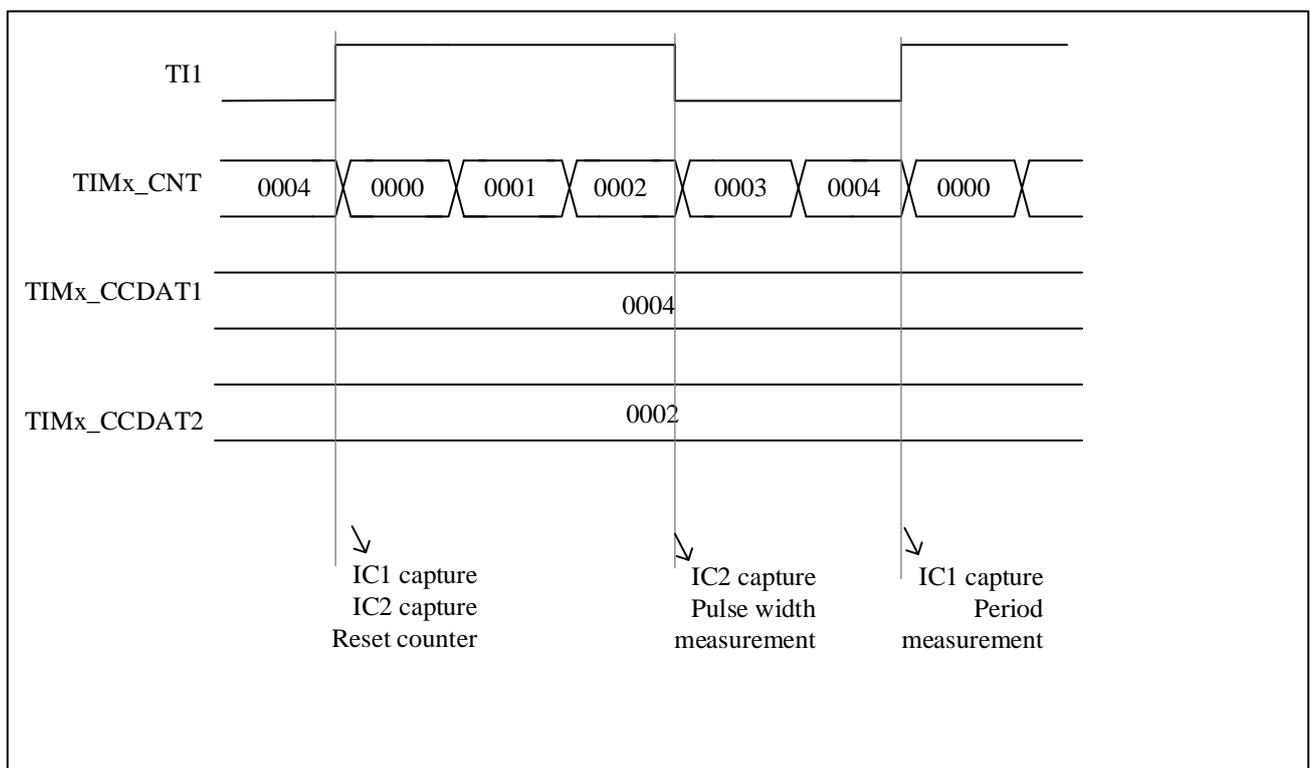
PWM 输入模式和普通输入捕获模式有一些区别，包括：

- 两个 ICx 信号映射到同一个 TIx 输入
- 两个 ICx 信号在极性相反的边沿有效
- 选择两个 TIxFP 信号之一作为触发输入
- 从机模式控制器配置为复位模式

例如，下面的配置流程可以用来知道 TI1 上 PWM 信号的周期和占空比（这取决于 CK\_INT 的频率和预分频器的值）。

- 配置 TIMx\_CCMOD1.CC1SEL 等于 ‘01’ 以选择 TI1 作为 TIMx\_CC1 的有效输入
- 配置 TIMx\_CCEN.CC1P 等于 ‘0’ 选择滤波定时器输入 1(TI1FP1) 的有效极性，在上升沿有效
- 配置 TIMx\_CCMOD1.CC2SEL 等于 ‘10’ 选择 TI1 作为 TIMx\_CC2 的有效输入
- 配置 TIMx\_CCEN.CC2P 等于 1 选择滤波定时器输入 2(TI1FP2) 的有效极性，下降沿有效
- 配置 TIMx\_SMCTRL.TSEL=101 选择 Filtered timer input 1 (TI1FP1) 作为有效触发输入
- 配置 TIMx\_SMCTRL.SMSEL=0100 配置从模式控制器为复位模式
- 配置 TIMx\_CCEN.CC1EN=1 和 TIMx\_CCEN.CC2EN=1 以启用捕获

图 13-16 PWM 输入模式时序



由于只有滤波器定时器输入 1 (TI1FP1) 和滤波器定时器输入 2 (TI2FP2) 连接到从模式控制器, 因此 PWM 输入模式只能与 TIMx\_CH1/TIMx\_CH2 信号一起使用。

### 13.3.7 强制输出模式

在输出模式 (TIMx\_CCMODx.CCxSEL=00) 下, 软件可以直接将输出比较信号强制为有效或无效电平。

用户可以设置 TIMx\_CCMODx.OCxMD=101 强制输出比较信号为有效电平。OCxREF 将被强制为高电平, OCx 得到与 CCxP 极性相反的值。另一方面, 用户可以设置 TIMx\_CCMODx.OCxMD=100 强制输出比较信号为无效电平, 即 OCxREF 被强制为低电平。

在此模式下, TIMx\_CCxDATx 影子寄存器和计数器的值仍然相互比较。

输出比较寄存器 TIMx\_CCxDATx 和计数器 TIMx\_CNT 之间的比较对 OCxREF 没有影响。并且仍然可以设置标志。因此, 仍然可以发送中断和 DMA 请求。

### 13.3.8 输出比较模式

用户可以使用此模式来控制输出波形, 或指示一段时间已过。

当捕获/比较寄存器和计数器的值相同时, 输出比较函数的操作如下:

- TIMx\_CCMODx.OCxMD 为输出比较模式, TIMx\_CCEN.CCxP 为输出极性。当比较匹配时, 如果设置 TIMx\_CCMODx.OCxMD=000, 则输出管脚将保持其电平; 如果设置 TIMx\_CCMODx.OCxMD=001, 则设置输出管脚有效; 如果设置 TIMx\_CCMODx.OCxMD=010, 则输出管脚将为 设置为无效; 如果设置 TIMx\_CCMODx.OCxMD=011, 则输出引脚将设置为翻转。
- 设置 TIMx\_STS.CCxITF
- 如果用户设置了 TIMx\_DINTEN.CCxIEN, 将产生相应的中断
- 如果用户设置 TIMx\_DINTEN.CCxDEN 并设置 TIMx\_CTRL2.CCDSEL 选择 DMA 请求, 将发送 DMA 请求

用户可以设置 TIMx\_CCMODx.OCxPEN 来选择是否使用捕获/比较预加载寄存器 (TIMx\_CCxDATx) 来选择捕获/比较影子寄存器。

时间分辨率是计数器的一个计数周期。

在单脉冲模式下, 输出比较模式也可用于输出单脉冲。

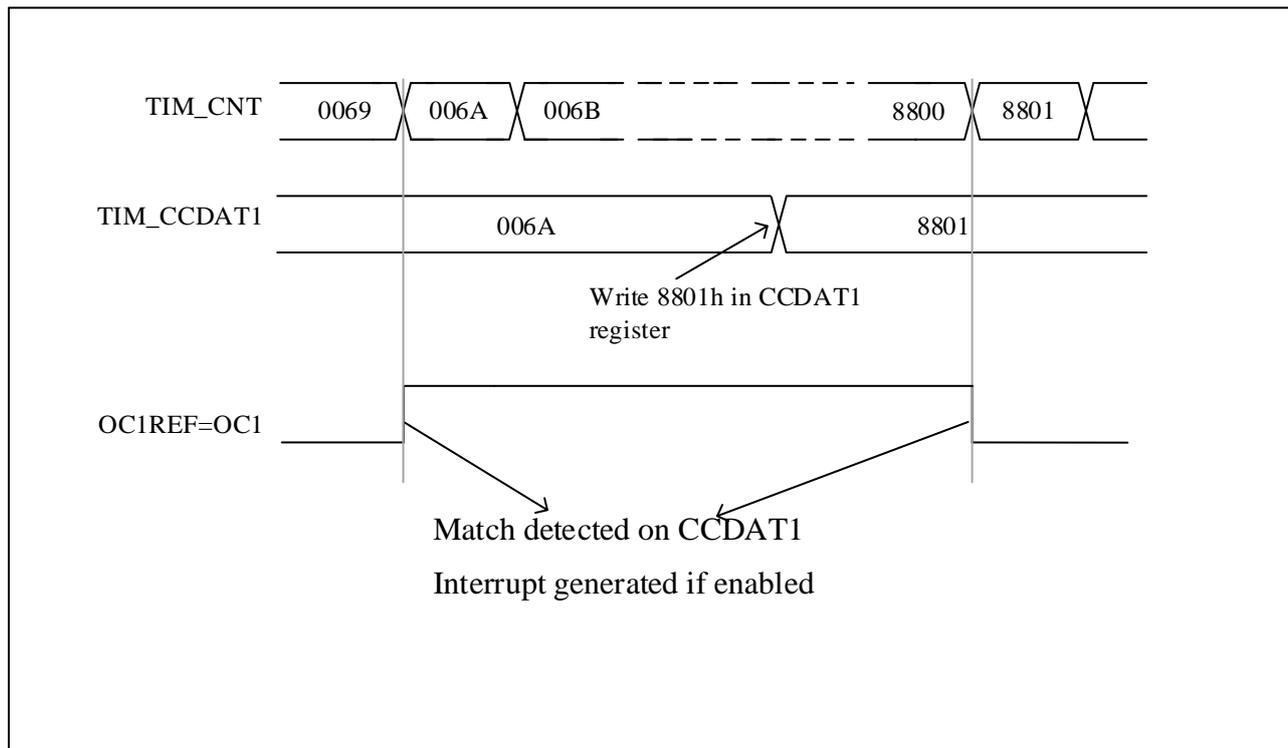
以下是输出比较模式的配置步骤:

- 首先, 用户应该选择计数器时钟
- 其次, 用所需数据设置 TIMx\_AR 和 TIMx\_CCxDATx
- 如果用户需要产生中断, 设置 TIMx\_DINTEN.CCxIEN
- 然后通过设置 TIMx\_CCEN.CCxP、TIMx\_CCMODx.OCxMD、TIMx\_CCEN.CCxEN 等选择输出模式
- 最后, 设置 TIMx\_CTRL1.CNTEN 启用计数器

用户可以随时通过设置 TIMx\_CCxDATx 来更新输出波形, 只要不启用预加载寄存器。否则, TIMx\_CCxDATx 影子寄存器将在下一次更新事件中更新。

例如：

图 13-17 输出比较模式，开启 OC1



### 13.3.9 PWM 模式

用户可以使用 PWM 模式产生一个信号，其占空比由 TIMx\_CCDATx 寄存器的值决定，其频率由 TIMx\_AR 寄存器的值决定。并且取决于 TIMx\_CTRL1.CAMSEL 的值，TIM 可以在边沿对齐模式或中央对齐模式下产生 PWM 信号。

用户可以通过设置 TIMx\_CCMODx.OCxMD=110 或设置 TIMx\_CCMODx.OCxMD=111 来设置 PWM 模式 1 或 PWM 模式 2。要启用预加载寄存器，用户必须设置相应的 TIMx\_CCMODx.OCxPEN。然后设置 TIMx\_CTRL1.ARPEN 自动重装载预加载寄存器。

用户可以通过设置 TIMx\_CCEN.CCxP 来设置 OCx 的极性。

当 TIM 处于 PWM 模式时，TIMx\_CNT 和 TIMx\_CCDATx 的值总是相互比较。

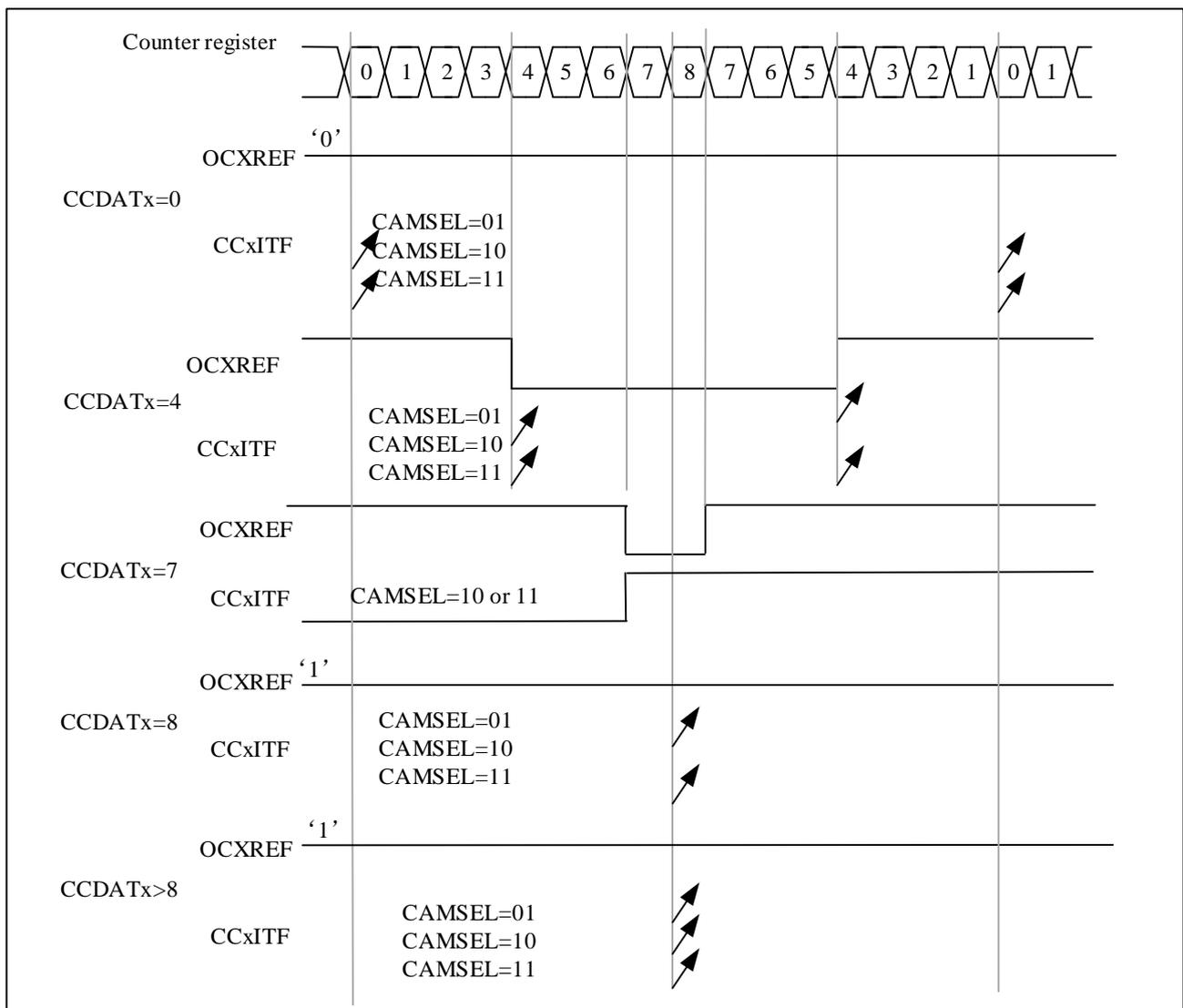
只有当更新事件发生时，预加载寄存器才会转移到影子寄存器。因此，用户必须在计数器开始计数之前通过设置 TIMx\_EVTGEN.UDGN 来复位所有寄存器。

#### 13.3.9.1 PWM 中央对齐模式

如果用户设置 TIMx\_CTRL1.CAMSEL 等于 01、10 或 11，PWM 中央对齐模式将被激活。比较标志的设置取决于 TIMx\_CTRL1.CAMSEL 的值。设置比较标志的情况有 3 种，仅当计数器向上计数时，仅当计数器向下计数时，或当计数器向上计数和向下计数时。用户不应通过软件修改 TIMx\_CTRL1.DIR，它是由硬件更新的。

中央对齐 PWM 波形示例如下，波形设置为：TIMx\_AR=8，PWM 模式 1，当计数器向下计数对应 TIMx\_CTRL1.CAMSEL=01 时设置比较标志。

图 13-18 中央对齐的 PWM 波形 (AR=8)



使用中央对齐模式时用户应注意的事项如下：

- 计数器向上或向下计数取决于 TIMx\_CTRL1.DIR 的值。注意不要同时更改 DIR 和 CAMSEL 位
- 用户在中央对齐模式下不要写计数器，否则会导致意想不到的结果。例如：
  - ◆ 如果写入计数器的值为 0 或者是 TIMx\_AR 的值，则方向会被更新，但不会产生更新事件
  - ◆ 如果写入计数器的值大于自动重载的值，则方向不会更新
- 为了安全起见，建议用户在启动计数器之前设置 TIMx\_EVTGEN.UDGN 以通过软件生成更新，并且在计数器运行时不要写入计数器

### 13.3.9.2 PWM 边沿对齐模式

边沿对齐模式有两种配置，向上计数和向下计数。

#### ● 向上计数

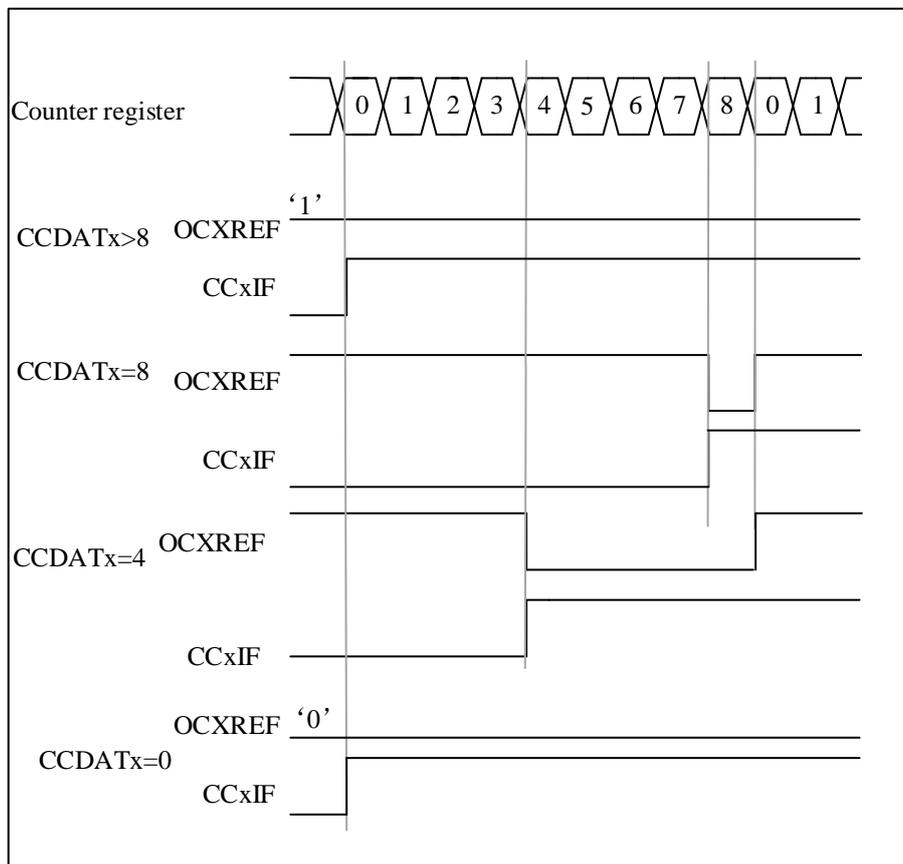
用户可以设置 TIMx\_CTRL1.DIR=0 使计数器向上计数。

PWM 模式 1 的示例：

当  $TIMx\_CNT < TIMx\_CCDATx$  时， $OCxREF$  为高电平，否则为低电平。如果  $TIMx\_CCDATx$  中的比较值大于自动重载值，则  $OCxREF$  将保持为 1。相反，如果比较值为 0，则  $OCxREF$  将保持为 0。

当  $TIMx\_AR=8$  时，PWM 波形如下：

图 13-19 边沿对齐 PWM 波形 (AR=8)



● 向下计数

用户可以设置  $TIMx\_CTRL1.DIR=1$  使计数器向下计数。

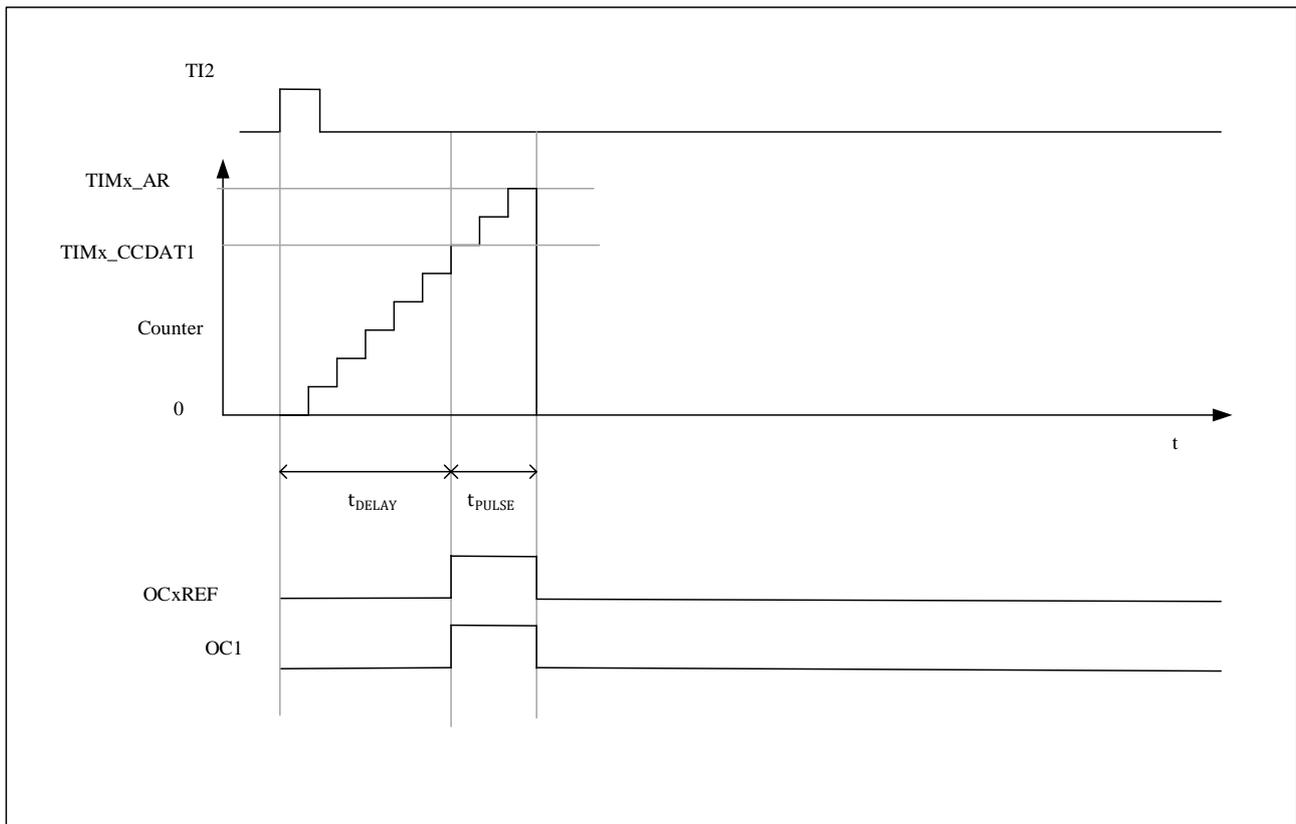
PWM 模式 1 的示例：

当  $TIMx\_CNT > TIMx\_CCDATx$  时， $OCxREF$  为低电平，否则为高电平。如果  $TIMx\_CCDATx$  中的比较值大于自动重载值，则  $OCxREF$  将保持为 1。

注：若第  $n$  个 PWM 周期  $CCDATx$  影子寄存器  $\geq AR$  值，第  $n+1$  个 PWM 周期  $CCDATx$  的影子寄存器值是 0。在第  $n+1$  个 PWM 周期的计数器为 0 的时刻，虽然计数器 =  $CCDATx$  影子寄存器的值 = 0， $OCxREF = '0'$ ，但不会产生比较事件。

### 13.3.10 单脉冲模式

在单脉冲模式(ONEPM)中，接收到触发信号，经过可控延迟  $t_{DELAY}$  后产生脉宽可控的脉冲  $t_{PULSE}$ 。输出模式需要配置为输出比较模式或 PWM 模式。选择单脉冲模式后，计数器会在更新事件 UEV 产生后停止计数。

**图 13-20 单脉冲模式示例**


以下是单脉冲模式的示例：

从 TI2 输入检测到上升沿触发，延迟  $t_{DELAY}$  后在 OC1 上产生宽度为  $t_{PULSE}$  的脉冲。

1. 计数器配置：向上计数，计数器  $TIMx\_CNT < TIMx\_CCDAT1 \leq TIMx\_AR$ ；
2. TI2FP2 映射到 TI2,  $TIMx\_CCMOD1.CC2SEL = '01'$ ；TI2FP2 配置为上升沿检测,  $TIMx\_CCEN.CC2P = '0'$ ；
3. TI2FP2 充当从模式控制器的触发器（TRGI）并启动计数器,  $TIMx\_SMCTRL.TSEL = '110'$ ,  $TIMx\_SMCTRL.SMSEL = '0110'$ （触发模式）；
4.  $TIMx\_CCDAT1$  写入要延迟的计数值 ( $t_{DELAY}$ ),  $TIMx\_AR - TIMx\_CCDAT1$  为脉宽  $t_{PULSE}$  的计数值；
5. 配置  $TIMx\_CTRL1.ONEPM = 1$  使能单脉冲模式, 配置  $TIMx\_CCMOD1.OC1MD = '111'$  选择 PWM2 模式；
6. 等待 TI2 有外部触发事件, OC1 输出一个单脉冲波形；

### 13.3.10.1 特殊情况：OCx 快速使能：

在单脉冲模式下，通过  $TIx$  输入检测到一个边沿，并触发计数器开始计数到比较值，然后输出一个脉冲。这些操作限制了可以达到的最小延迟  $t_{DELAY}$ 。

您可以设置  $TIMx\_CCMODx.OCxFEN = 1$  开启 OCx 快速使能，在触发上升沿后，OCxREF 信号将被强制转换为与比较匹配立即发生的电平相同的电平，而不管比较结果如何。OCxFEN 快速使能仅在通道模式配置为 PWM1 和 PWM2 模式时生效。

### 13.3.11 在外部事件上清除 OCxREF 信号

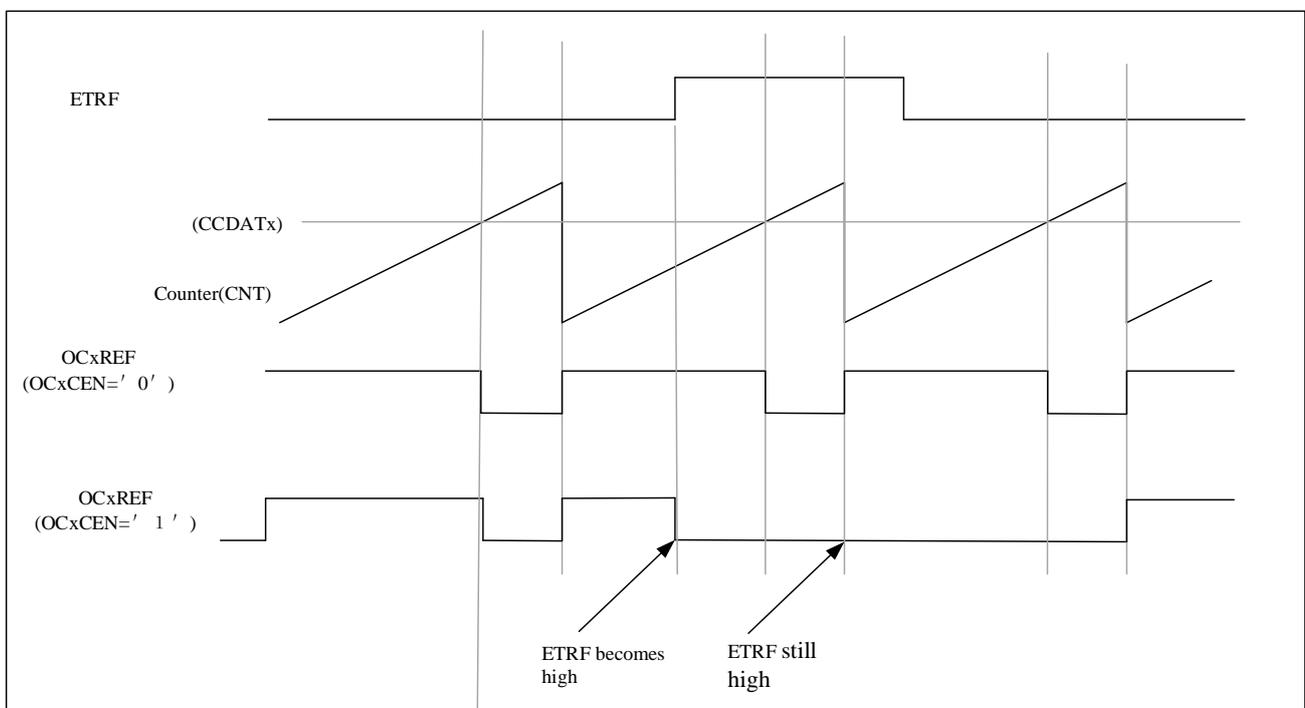
如果用户设置 TIMx\_CCMODx.OCxCEN=1, ETRF 输入的高电平可用于驱动 OCxREF 信号为低电平, OCxREF 信号将保持低电平, 直到下一次 UEV 发生。只有输出比较和 PWM 模式可以使用该功能。在强制模式下不能使用。

例如: 为了控制电流, 用户可以将 ETR 信号连接到比较器的输出端, ETR 的操作如下:

- 设置 TIMx\_SMCTRL.EXTPS=00 禁用外部触发预分频器。
- 设置 TIMx\_SMCTRL.EXCCEN=0 禁用外部时钟模式 2。
- 设置 TIMx\_SMCTRL.EXTP 和 TIMx\_SMCTRL.EXTF, 根据需要配置外触发极性和外触发滤波器。

例: 当 ETRF 输入变高时, OCxREF 信号对于不同的 OCxCEN 值的行为。在这种情况下, 定时器设置为 PWM 模式。

图 13-21 清除 TIMx 的 OCxREF



### 13.3.12 刹车功能

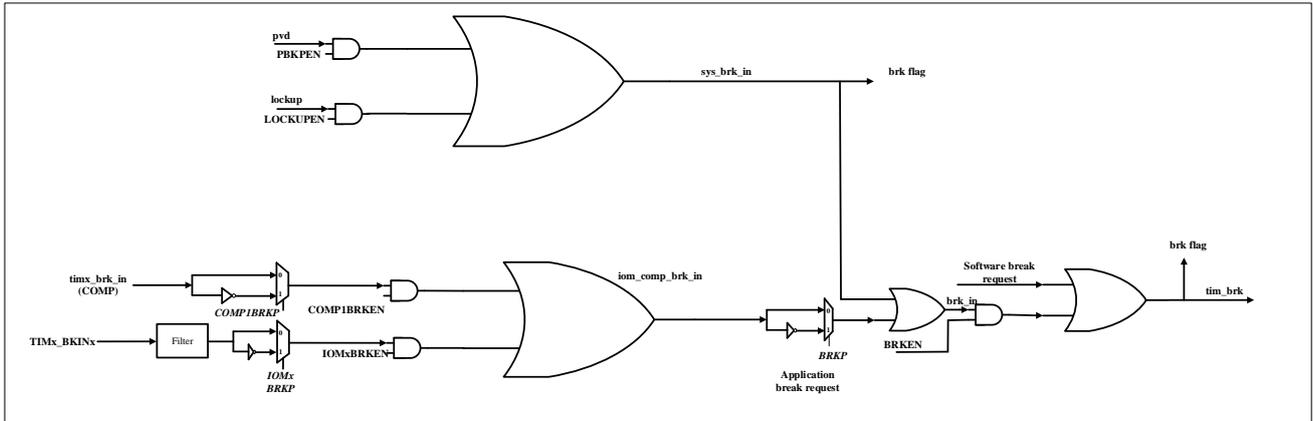
使用刹车功能时, 设置相应的控制位时会修改输出使能信号并将 OCx 输出置为 OIx 电平。TIM3 刹车功能仅对 OC3 和 OC4 有效, 对 OC1 和 OC2 无效。

当启用多个刹车信号时, 每个刹车信号构成一个 OR 逻辑。这里有一些信号可能是刹车的来源。

- 刹车输入引脚 (4 个 IO 引脚)
- PVD 事件。
- 内核 Hardfault 事件。
- 比较器的输出信号。

■ 软件设置 TIMx\_EVTGEN.BGN。

图 13-22 刹车输入



复位后刹车电路将被禁用。MOEN 位将为低电平。用户可以设置 TIMx\_BKDT.BKEN 来启用刹车功能。通过设置 TIMx\_BKDT.BKP 可以选择刹车输入信号的极性。用户可以同时修改 TIMx\_BKDT.BKEN 和 TIMx\_BKDT.BKP。用户设置 TIMx\_BKDT.BKEN 和 TIMx\_BKDT.BKP 后，生效前有 1 个 APB 时钟周期延迟。因此，用户需要等待 1 个 APB 时钟周期才能读回写入位的值。

MOEN 的下降沿可以是异步的，所以在实际信号和同步控制位之间设置了一个再同步电路。该电路将导致异步和同步信号之间的延迟。当用户设置 TIMx\_BKDT.MOEN 为低电平时，用户需要在读取该值之前插入一个延迟。因为写入了异步信号，但用户读取了同步信号。

刹车发生后的行为如下：

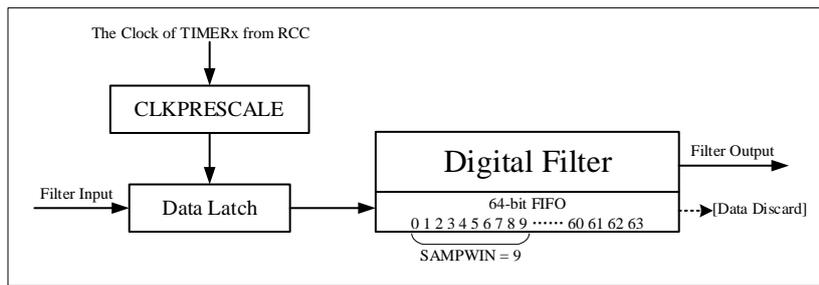
- TIMx\_BKDT.MOEN 将被异步清除，输出立即进入空闲状态 (OIx)。即使 MCU 振荡器关闭，这也会生效。
- 一旦 TIMx\_BKDT.MOEN=0，每个输出通道的输出将使用 TIMx\_CTRL2.OIx 中编程的电平驱动。
- 如果 TIMx\_DINTEN.BIEN=1，当 TIMx\_STS.BITF=1 时，会产生中断。
- 如果用户设置了 TIMx\_BKDT.AOEN，TIMx\_BKDT.MOEN 将在下一次 UEV 发生时自动设置。用户可以使用它来调节。如果用户未设置 TIMx\_BKDT.AOEN，则 TIMx\_BKDT.MOEN 将保持低电平，直到再次设置为 1。在这种情况下，用户可以使用它来保证安全。用户可以将刹车输入连接到热传感器、电源驱动器警报或其他安全组件。
- 刹车输入有效时，TIMx\_BKDT.MOEN 不能自动置位或软件同时置位，TIMx\_STS.BITF 也不能清零。因为刹车输入在电平上处于有效状态。

为保证应用安全，刹车电路具有写保护功能，并有刹车输入输出管理。它允许用户冻结一些参数，例如 OCx 极性和禁用时的状态、OCxMD 配置、刹车启用和极性。用户可以通过设置 TIMx\_BKDT.LCKCFG 选择使用 3 种保护级别之一。但是，TIMx\_BKDT.LCKCFG 只能在 MCU 复位后写入一次。

### 13.3.12.1 刹车滤波

寄存器 TIM1\_BKFR 描述如下：

图 13-23 滑动滤波



- 数字滤波器通过 RCC 的 TIM1 时钟采样刹车信号，在 64 位 FIFO 中累积采样。仅在 TIM1\_BKFR.WSIZE [5:0] 中定义的窗口大小内采样数据，最大大小为 64。
- 过滤器输出采样窗口内的多数值，该值由 TIM1\_BKFR.THRESH [5:0] 中的阈值定义，最大阈值为 63。此值应等于或大于窗口大小的一半。如果采样窗口内的逻辑 1 和逻辑 0 计数均不大于阈值，则数字滤波器保持先前的输出值。
- RCC\_TIMFILTCFG.TIM3FILTCLK[4:0] 寄存器决定相应数字滤波器的采样率。过滤器 FIFO 在每个采样时钟从输入中捕获一个采样值。
- 如果数字滤波器关闭，滤波器输入将像电线一样绕过输出。

### 13.3.13 调试模式

当微控制器处于调试模式（Cortex-M0 内核停止）时，根据 DBG\_CTRL.TIMx\_STOP 配置，TIMx 计数器可以继续正常工作或停止。详见 3.3.1 章节。

### 13.3.14 外部事件触发装载 LVR

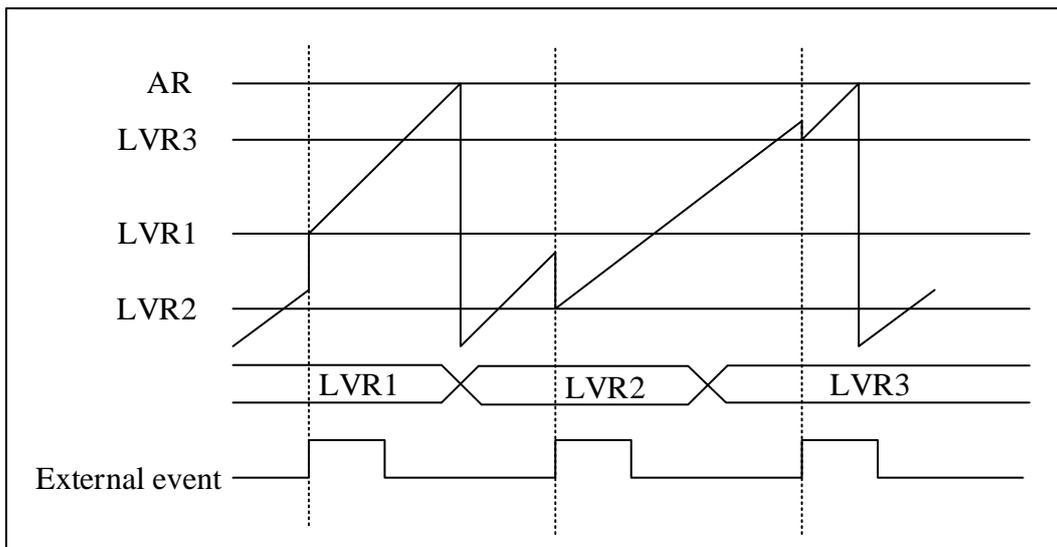
用户可以选择一个外部信号作为触发器，用外部事件计数器装载值寄存器（TIMx\_ENCLVR.LVR[31:0]）的当前值重新加载定时器的值。通过在 TIMx\_SMCTRL.TSEL[2:0] 寄存器中设置所需的值，可以选择不同的外部触发信号，该功能通过 TIMx\_CTRL1.TRGLDCNTEN 位启用。

注意，选择触发源为 TI1 的边沿检测器时（TSEL = '100'），该触发信号是上升沿/下降沿均有效。选择其他触发源时（TSEL != '100'），该触发信号是上升沿有效的，用户也可配置 CCxP=1 或者 EXTP=1 选择对应的触发信号下降沿有效。

*注：开启外部事件（TRGI）触发计数器装载值功能时，如果同时使用输出比较模式，必须选择 ETRF 或者 ITRx 作为触发源（TSEL 只能配置为“0xx 或者 111”）。*

在下图中，外部信号被用作触发器，在上升沿激活。每次检测到外部信号的上升沿时，2 个 TIM\_CLK 时钟后，定时器的值就会加载为 TIMx\_ENCLVR.LVR[31:0]。

图 13-24 外部事件触发装载 LVR 时，CNT 计数方式



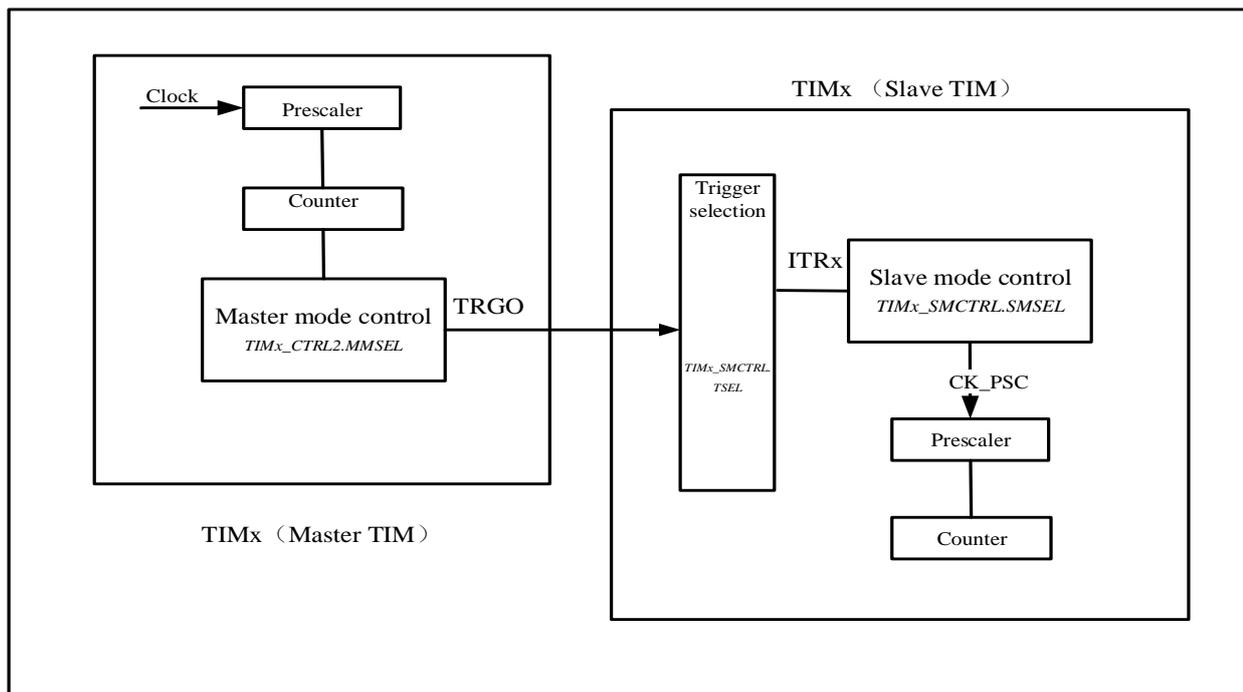
### 13.3.15 TIMx 定时器和外部触发的同步

与高级定时器相同，见 12.3.16。

### 13.3.16 定时器同步

所有 TIMx 定时器都在内部相互连接。该实现允许任何主定时器提供触发以复位、启动、停止或为其他从定时器提供时钟。主时钟用于内部计数器，可以预分频。下图为定时器互连框图。同步功能不支持连接的动态变化。用户应在启用主定时器的触发器或时钟之前配置并启用从定时器。

图 13-25 主/从定时器的例子



### 13.3.16.1 主定时器作为另一个定时器的预分频器

定时器 1 作为定时器 3 的预分频器。TIM1 是主，TIM3 是从。

用户需要为此配置执行以下步骤。

- 设置 TIM1\_CTRL2.MMSEL='010' 以使用 TIM1 的更新事件作为触发输出。
- 配置 TIM3\_SMCTRL.TSEL='000'，将 TIM1 的 TRGO 连接到 TIM3。
- 配置 TIM3\_SMCTRL.SMSEL = '0111'，从模式控制器将配置为外部时钟模式 1。
- 通过设置 TIM3\_CTRL1.CNTEN = "1"，启动 TIM3。
- 通过设置 TIM1\_CTRL1.CNTEN = "1"，启动 TIM1。

*注：如果用户通过配置 MMSEL = '1xx' 选择 OCx 作为 TIM1 的触发输出，则 OCx 上升沿将用于驱动 TIM3。*

### 13.3.16.2 主定时器使能另一个定时器

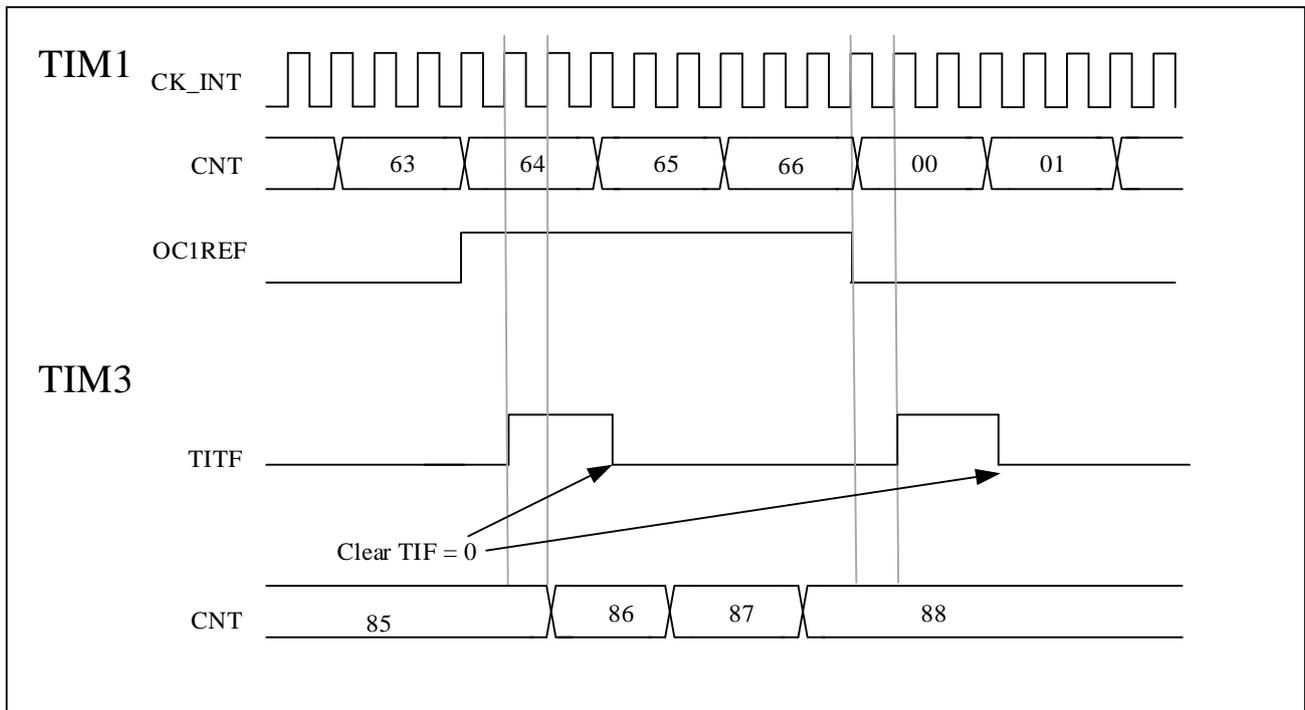
在本例中，TIM3 通过 TIM1 的输出比较使能。TIM1 的 OC1REF 输出为高电平后，TIM3 计数器将开始计数。两个计数器的时钟均基于 CK\_INT，通过预分频器除以 3 ( $f_{CK\_CNT} = f_{CK\_INT}/3$ )。

配置步骤如下所示。

- 设置 TIM1\_CTRL2.MMSEL='100' 以使用 TIM1 的 OC1REF 作为触发输出。
- 配置 TIM1\_CCMOD1 寄存器来配置 OC1REF 输出波形。
- 设置 TIM3\_SMCTRL.TSEL = '000' 将 TIM1 触发输出连接到 TIM3。
- 设置 TIM3\_SMCTRL.SMSEL = '0101' 将 TIM3 设置为门控模式。
- 设置 TIM3\_CTRL1.CNTEN = '1' 来启动 TIM3。
- 设置 TIM1\_CTRL1.CNTEN = '1' 以启动 TIM1。

*注：TIM3 时钟与 TIM1 时钟不同步，该模式仅影响 TIM3 计数器使能信号。*

图 13-26 定时器 3 由定时器 1 的 OC1REF 门控



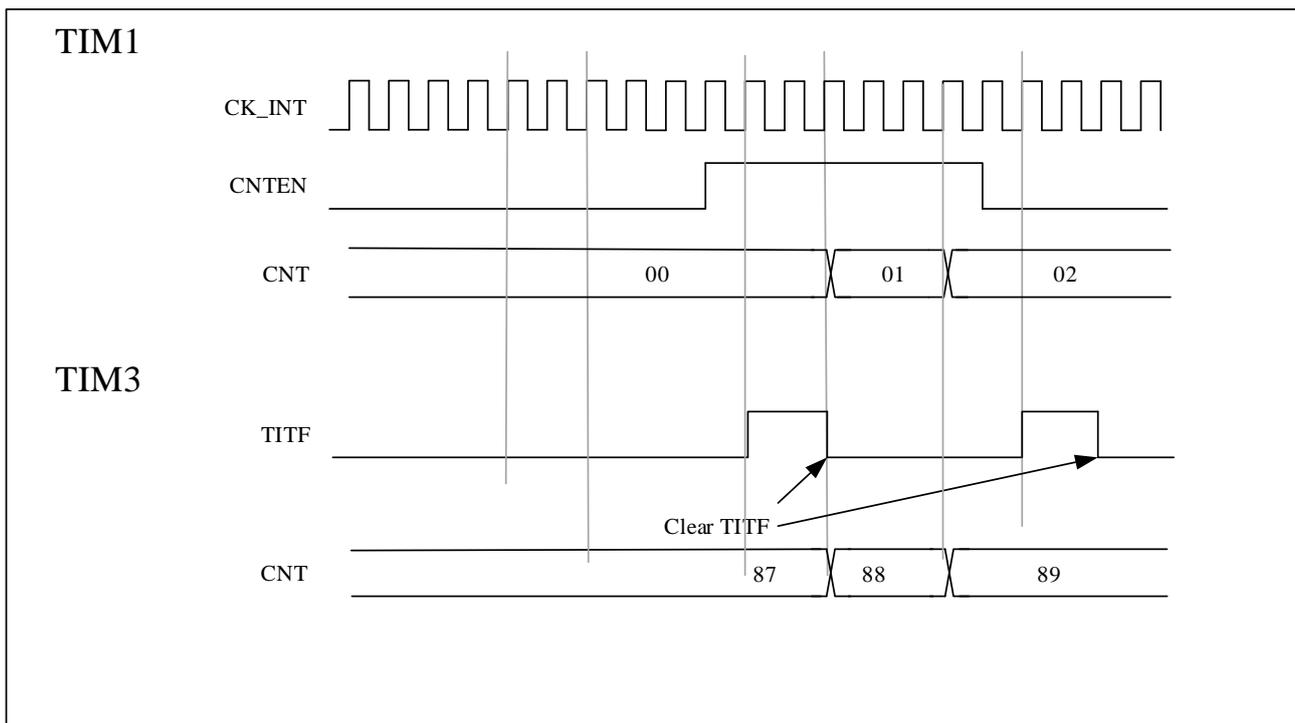
在下一个示例中，用 TIM1 的使能信号门控 TIM3，设置 TIM1\_CTRL1.CNTEN= '0' 以停止 TIM1。

仅当 TIM1 使能时，TIM3 才基于分频的内部时钟计数。两个计数器的时钟均基于 CK\_INT，通过预分频器除以 3 ( $f_{CK\_CNT} = f_{CK\_INT}/3$ )。

配置步骤如下所示

- 设置 TIM1\_CTRL2.MMSEL='001' 使用 TIM1 的使能信号作为触发输出
- 设置 TIM3\_SMCTRL.TSEL = '000' 配置 TIM3 从 TIM1 获取触发输入
- 设置 TIM3\_SMCTRL.SMSEL = '0101' 将 TIM3 配置为门控模式。
- 设置 TIM3\_CTRL1.CNTEN= '1' 来启动 TIM3。
- 设置 TIM1\_CTRL1.CNTEN= '1' 以启动 TIM1。
- 设置 TIM1\_CTRL1.CNTEN= '0' 以停止 TIM1。

图 13-27 定时器 3 由定时器 1 的使能门控



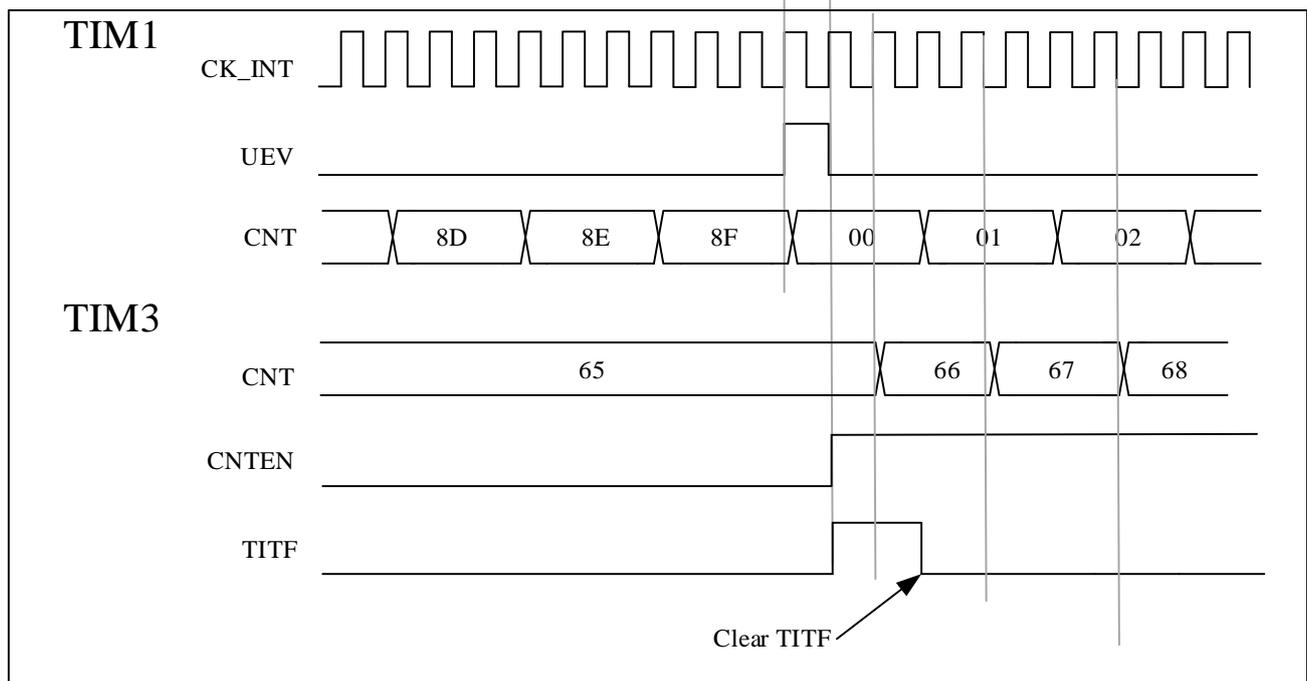
### 13.3.16.3 主定时器启动另一个定时器

在这个例子中，我们可以使用更新事件作为触发源。TIM1 是主，TIM3 是从。

配置步骤如下图所示：

- 设置 TIM1\_CTRL2.MMSEL='010' 使用 TIM1 的更新事件作为触发输出
- 配置 TIM1\_AR 寄存器设置输出周期。
- 设置 TIM3\_SMCTRL.TSEL='000' 将 TIM1 触发输出连接到 TIM3。
- 设置 TIM3\_SMCTRL.SMSEL='0110' 将 TIM3 设置为触发模式。
- 设置 TIM1\_CTRL1.CNTEN=1 启动 TIM1。

图 13-28 使用定时器 1 的更新触发定时器 3



### 13.3.16.4 使用一个外部触发同步地启动 2 个定时器

在本例中，TIM1 的 TI1 输入上升时使能 TIM1，使能 TIM1 时使能 TIM3。为确保计数器对齐，TIM1 必须配置为主/从模式。对于 TI1，TIM1 是从；对于 TIM3，TIM1 是主。

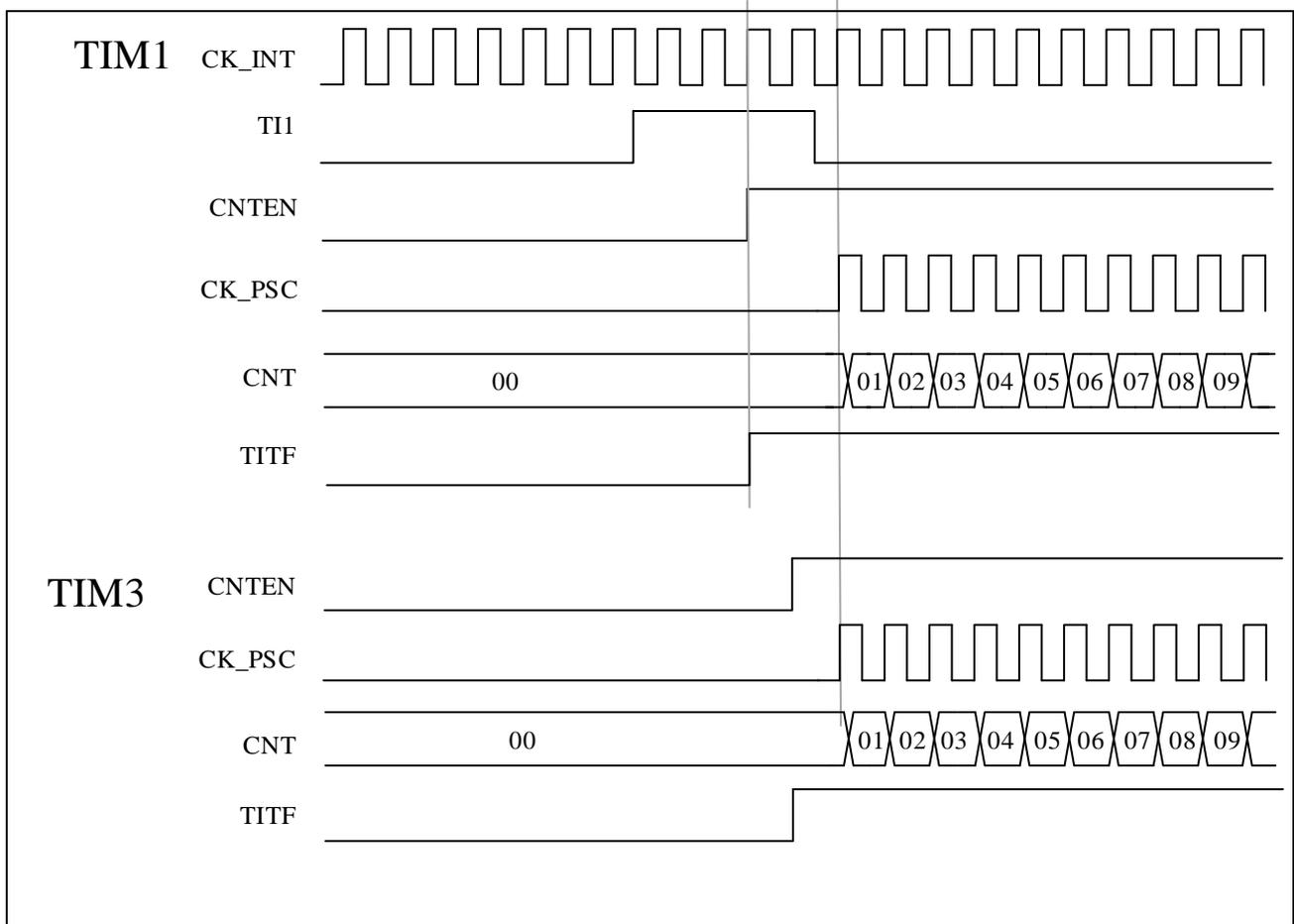
配置步骤如下图所示：

- 设置 TIM1\_MMSEL = '001' 使用使能信号作为触发输出
- 设置 TIM1\_SMCTRL.TSEL = '100' 将 TIM1 配置为从模式并接收 TI1 的触发输入。
- 设置 TIM1\_SMCTRL.SMSEL = '0110' 将 TIM1 配置为触发模式。
- 设置 TIM1\_SMCTRL.MSMD = '1' 将 TIM1 配置为主/从模式。
- 设置 TIM3\_SMCTRL.TSEL = '000' 将 TIM1 触发输出连接到 TIM3。
- 设置 TIM3\_SMCTRL.SMSEL = '0110' 将 TIM3 配置为触发模式。

当 TI1 上升沿到来时，两个定时器开始根据内部时钟同步计数，两个 TITF 标志同时置位。

注：下图显示了在主/从模式下 CNTEN 和 TIM1 的 CK\_PSC 之间的延迟。

图 13-29 使用定时器 1 的 TI1 输入触发定时器 1 和定时器 3



### 13.3.17 编码器接口模式

编码器使用两个输入 TI1 和 TI2 作为接口，计数器对 TI1FP1 或 TI2FP2 上的每个边沿变化进行计数。计数方向由硬件 TIMx\_CTRL1.DIR 自动控制。编码器计数模式共有三种：

1. 计数器只在 TI1 的边沿计数，TIMx\_SMCTRL.SMSEL = '0001'；
2. 计数器只在 TI2 的边沿计数，TIMx\_SMCTRL.SMSEL = '0010'；
3. 计数器同时在 TI1 和 TI2 的边沿计数，TIMx\_SMCTRL.SMSEL = '0011'；

编码器接口相当于使用带方向选择的外部时钟，计数器只在 0 和自动重载值 (TIMx\_AR.AR [15:0]) 之间连续计数。因此，需要提前配置自动重载寄存器 TIMx\_AR。

*注意：编码器模式和外部时钟模式 2 不兼容，不能同时选择。*

计数方向与编码器信号的关系如下表：

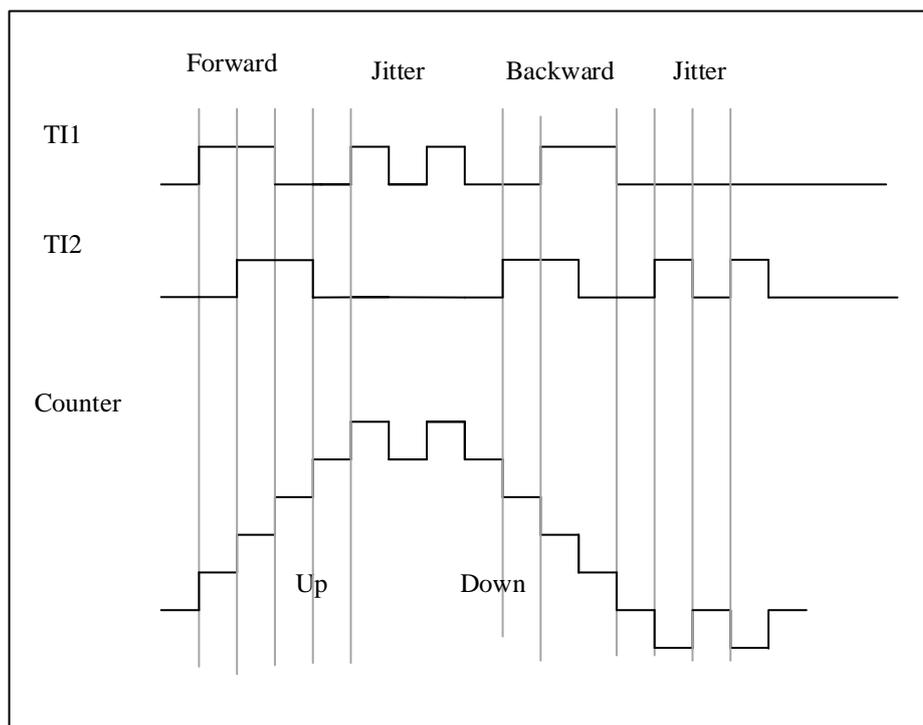
表 13-1 计数方向与编码器信号的关系

有效边沿	相对信号的电平 (TI1FP1对应TI2, TI2FP2对应TI1)	TI1FP1信号		TI2FP2信号	
		上升	下降	上升	下降
仅在TI1计数	高	向下计数	向上计数	不计数	不计数
	低	向上计数	向下计数	不计数	不计数
仅在TI2计数	高	不计数	不计数	向上计数	向下计数
	低	不计数	不计数	向下计数	向上计数
在TI1和TI2上计数	高	向下计数	向上计数	向上计数	向下计数
	低	向上计数	向下计数	向下计数	向上计数

以下是选择了双边沿触发以抑制输入抖动的编码器示例：

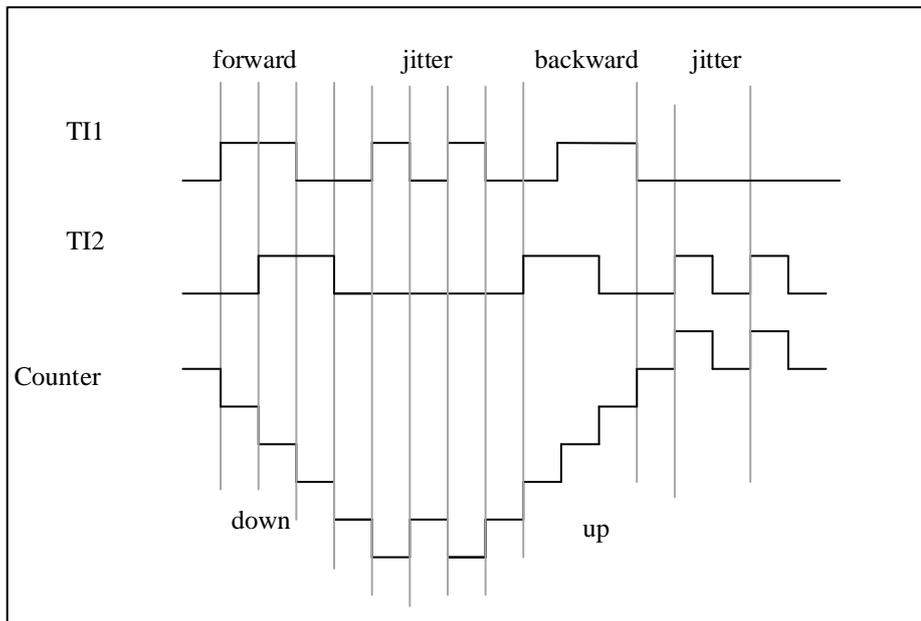
1. IC1FP1 映射到 TI1 (TIMx\_CCMOD1.CC1SEL= '01'), IC1FP1 不反相 (TIMx\_CCEN.CC1P= '0');
2. IC1FP2 映射到 TI2 (TIMx\_CCMOD2.CC2SEL= '01'), IC2FP2 不反相 (TIMx\_CCEN.CC2P= '0');
3. 输入在上升沿和下降沿均有效 (TIMx\_SMCTRL.SMSEL = '0011');
4. 启用计数器 TIMx\_CTRL1.CNTEN= '1';

图 13-30 编码器模式下的计数器操作实例



下图为 IC1FP1 极性反转时的计数器行为示例 (CC1P='1', 其他配置同上)

图 13-31 IC1FP1 反相的编码器接口模式实例



### 13.3.18 与霍尔传感器的接口

请查阅 12.3.21 节

## 13.4 TIMx 寄存器描述 (x=3)

关于在寄存器描述里面所用到的缩写, 详见 1.1 节。

可以用半字 (16 位) 或字 (32 位) 的方式操作这些外设寄存器。

### 13.4.1 寄存器总览

表 13-2 TIM3 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	TIM_CTRL1	TRGRSTCNTEN	Reserve													CISEL	Reserve	CLRSEL	Reserve	PBKPEN	LBPEN	ARPEN	ONEPM	Reserve	UPDIS	UPRS	CAMSEL[1:0]	DIR	CNTEN				
004h	TIM_CTRL2	Reserve											TISEL	Reserve	CCDSEL	Reserve	MMSEL[2:0]		Reserve				O14	Reserve	O13	Reserve							
008h	TIM_STS	Reserve		IOMBITF	Reserve			COMPBITF	PBKPTIF	LBPPTIF	Reserve	BITF	TITF	Reserve	UDITF	Reserve				CC4OCF	CC3OCF	CC2OCF	CC1OCF	Reserve				CC4ITF	CC3ITF	CC2ITF	CC1ITF		
00Ch	TIM_EVTGEN	Reserve																			BGN	TGN	Reserve	UDGN	Reserve				CC4GN	CC3GN	CC2GN	CC1GN	

010h	TIM_SMCTRL	Reserve										MSMD	EXTF[3:0]			EXTP	EXCEN	EXTPS[1:0]			SMSEL[3:0]			Reserve	TSEL[2:0]			
014h	TIM_DINTEN	Reserve							TDEN	Reserve	UDEN	BIEN	TIEN	UIEN	Reserve			CC4DEN	CC3DEN	CC2DEN	CC1DEN	Reserve			CC4IEN	CC3IEN	CC2IEN	CC1IEN
018h	TIM_CCMOD1_OUT	Reserve										OC2MD[2:0]		OC2CEN	OC2PEN	OC2PEN	CC2SEL[1:0]			OC1MD[2:0]		OC1CEN	OC2PEN	OC1PEN	CC1SEL[1:0]			
	TIM_CCMOD1_IN	Reserve										IC2F[3:0]			IC2PSC[1:0]			CC2SEL[1:0]			IC1F[3:0]			IC1PSC[1:0]			CC1SEL[1:0]	
01Ch	TIM_CCMOD2_OUT	Reserve										OC4MD[2:0]		OC4CEN	OC4PEN	OC4PEN	CC4SEL[1:0]			OC3MD[2:0]		OC3CEN	OC3PEN	OC3PEN	CC3SEL[1:0]			
	TIM_CCMOD2_IN	Reserve										IC4F[3:0]			IC4PSC[1:0]			CC4SEL[1:0]			IC3F[3:0]			IC3PSC[1:0]			CC3SEL[1:0]	
024h	TIM_CCEN	Reserve										CC4P	CC4EN	Reserve	CC3P	CC3EN	Reserve	CC2P	CC2EN	Reserve	CC1P	CC1EN	Reserve	Reserve				
028h	TIM_CCDAT1	Reserve										CCDAT1[15:0]																
02Ch	TIM_CCDAT2	Reserve										CCDAT2[15:0]																
030h	TIM_CCDAT3	Reserve										CCDAT3[15:0]																
034h	TIM_CCDAT4	Reserve										CCDAT4[15:0]																
040h	TIM_PSC	Reserve										PSC[15:0]																
044h	TIM_AR	Reserve										AR[15:0]																
048h	TIM_CNT	Reserve										CNT[15:0]																
04Ch	TIM_REPCNT	Reserve										REPCNT[7:0]																
050h	TIM_BKDT	Reserve										LCKCFG[1:0]	Reserve	BKEN	BKP	AOEN	MOEN	Reserve										
060h	TIM_BKFR	Reserve	THRESH[5:0]				Reserve	WSIZE[5:0]				FILTEN	Reserve															
07Ch	TIM_AF1	Reserve							IOM4BRKP	IOM3BRKP	IOM2BRKP	IOM4BRKEN	IOM3BRKEN	IOM2BRKEN	Reserve			COMP1BRKP	IOM1BRKP	Reserve					COMP1BRKEN	IOM1BRKEN		
090h	TIM_ENCLVR	Reserve										LVR[15:0]																
094h	TIM_DCTRL	Reserve										DBADDR[5:0]					Reserve	DBLEN[5:0]										
098h	TIM_DADDR	BURST[31:0]																										

### 13.4.2 控制寄存器 1 (TIMx\_CTRL1)

偏移地址: 0x00

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16											
TRGLDCNTEN															Reserved											C1SEL
rw																										rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0											
Reserved			CLRSEL	Reserved		PBKPEN	LBKPEN	ARPEN	ONEPM	Reserved			UPDIS	UPRS	CAMSEL[1:0]	DIR	CNTEN									
			rw			rw	rw	rw	rw				rw	rw	rw	rw	rw									

位域	名称	描述
31	TRGLDCNTEN	TRGI有效事件触发计数器装载为LVR值使能。 当检测到TRGI事件（取决于TIMx_SMCTRL.TSEL[2:0]寄存器）时，计数器将装载为TIMx_ENCLVR.LVR值。 0: TRGI有效事件不能触发计数器装载为LVR值 1: TRGI有效事件能触发计数器装载为LVR值
30:17	Reserved	保留，必须保持复位值
16	C1SEL	通道1选择（Channel 1 selection） 0: 选择外部CH1（来自IOM）信号 1: 选择内部CH1（来自COMP）信号
15:14	Reserved	保留，必须保持复位值
13	CLRSEL	OCxRef选择（OCxRef selection） 0: 选择外部Ocxclr（ETR）信号 1: 选择内部OCxclr（来自COMP）信号
12	Reserved	保留，必须保持复位值
11	PBKPEN	PVD作为BRK启用（PVD as brk Enable） 0: 禁止 1: 使能
10	LBKPEN	锁存作为BRK使能（LockUp as brk Enable）（Core Hardfault） 0: 禁止 1: 使能
9	ARPEN	自动重载预装载允许位（Auto-reload preload enable） 0: TIMx_AR寄存器的影子寄存器禁用 1: TIMx_AR寄存器的影子寄存器使能
8	ONEPM	单脉冲模式（One pulse mode） 0: 禁用单脉冲模式，发生更新事件时不影响计数器计数。 1: 使能单脉冲模式，下次更新事件发生时计数器停止计数
7:6	Reserved	保留，必须保持复位值

位域	名称	描述
31	TRGLDCNTEN	<p>TRGI有效事件触发计数器装载为LVR值使能。</p> <p>当检测到TRGI事件（取决于TIMx_SMCTRL.TSEL[2:0]寄存器）时，计数器将装载为TIMx_ENCLVR.LVR值。</p> <p>0: TRGI有效事件不能触发计数器装载为LVR值</p> <p>1: TRGI有效事件能触发计数器装载为LVR值</p>
5	UPDIS	<p>更新禁用（Update disable）</p> <p>该位用于启用/禁用软件生成的更新事件（UEV）事件。</p> <p>0: 启用。如果满足以下条件之一，将生成 UEV：</p> <ul style="list-style-type: none"> <li>– 计数器上溢/下溢</li> <li>– TIMx_EVTGEN.UDGN 位被设置</li> <li>– 从模式控制器的更新生成</li> </ul> <p>影子寄存器将使用预加载值进行更新。</p> <p>1: UEV 禁用。不生成更新事件，影子寄存器（AR、PSC 和 CCDATx）保持它们的值。如果 TIMx_EVTGEN.UDGN 位置位或从模式控制器发出硬件复位，则重新初始化计数器和预分频器。</p>
4	UPRS	<p>更新请求源（Update request source）</p> <p>该位用于通过软件选择 UEV 事件源。</p> <p>0: 如果更新中断或 DMA 请求使能，以下任何事件都会产生更新中断或 DMA 请求：</p> <ul style="list-style-type: none"> <li>– 计数器上溢/下溢</li> <li>– TIMx_EVTGEN.UDGN 位被设置</li> <li>– 从模式控制器的更新生成</li> </ul> <p>1: 如果更新中断或 DMA 请求使能，只有计数器上溢/下溢会产生更新中断或 DMA 请求。</p>
3:2	CAMSEL[1:0]	<p>选择中央对齐模式（Center-aligned mode selection）</p> <p>00: 边缘对齐模式。TIMx_CTRL1.DIR 指定向上计数或向下计数。</p> <p>01: 中央对齐模式1。计数器在中央对齐模式下计数，向下计数时输出比较中断标志位设置为 1。</p> <p>10: 中央对齐模式2。计数器在中央对齐模式下计数，向上计数时输出比较中断标志位设置为1。</p> <p>11: 中央对齐模式3。计数器在中央对齐模式下计数，向上计数或向下计数时输出比较中断标志位设置为 1。</p> <p><i>注意：当计数器仍然启用时（TIMx_CTRL1.CNTEN = 1），不允许从边缘对齐模式切换到中央对齐模式。</i></p>
1	DIR	<p>方向（Direction）</p> <p>0: 计数器向上计数；</p> <p>1: 计数器向下计数。</p> <p><i>注：当计数器配置为中央对齐模式或编码器模式时，该位为只读。</i></p>

位域	名称	描述
31	TRGLDCNTEN	TRGI有效事件触发计数器装载为LVR值使能。 当检测到TRGI事件（取决于TIMx_SMCTRL.TSEL[2:0]寄存器）时，计数器将装载为TIMx_ENCLVR.LVR值。 0: TRGI有效事件不能触发计数器装载为LVR值 1: TRGI有效事件能触发计数器装载为LVR值
0	CNTEN	使能计数器（Counter enable） 0: 禁止计数器； 1: 使能计数器。 <i>注：在软件设置了CNTEN位后，外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置CNTEN位。</i> <i>注：软件设置了CNTEN位后，需要等待至少两个TIMx_CLK，CNTEN才能从TIMx_PCLK同步到TIMx_CLK生效。</i>

### 13.4.3 控制寄存器 2（TIMx\_CTRL2）

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved											TI1SEL	Reserved	CCDSEL	Reserved		
											rw		rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
MMSEL[3:0]				Reserved				OI4	Reserved	OI3	Reserved					
rw								rw		rw						

位域	名称	描述
31:20	Reserved	保留，必须保持复位值
19	TI1SEL	TI1选择（TI1 selection） 0: TIMx_CH1引脚连到TI1输入； 1: TIMx_CH1、TIMx_CH2和TIMx_CH3引脚经异或后连到TI1输入。
18	Reserved	保留，必须保持复位值
17	CCDSEL	捕获/比较的DMA选择（Capture/compare DMA selection） 0: 当发生CCx事件时，送出CCx的DMA请求； 1: 当发生更新事件时，送出CCx的DMA请求。
16	Reserved	保留，必须保持复位值
15:12	MMSEL[3:0]	主模式选择 这4位用于选择在主模式下发送到从定时器的同步信息（TRGO）。可能的组合如下： x000: 复位 - 当TIMx_EVTGEN.UDGN置位或从模式控制器产生复位时，将出现TRGO脉冲。在后一种情况下，TRGO上的信号与实际复位相比有所延迟。

位域	名称	描述
		x001: 使能 - TIMx_CTRL1.CNTEN 位用作触发输出 (TRGO)。有时需要同时启动多个定时器或者在一段时间内开启从定时器。 当 TIMx_CTRL1.CNTEN 位置位或门控模式下的触发输入为高电平时, 计数器使能信号置位。 当计数器使能信号由触发输入控制时, TRGO 上有一个延迟, 除非选择了主/从模式 (参见 TIMx_SMCTRL.MSMD 位的说明)。 x010: 更新 - 选择更新事件作为触发输出 (TRGO)。例如, 主定时器时钟可用作从定时器预分频器。 x011: 比较脉冲 - 当 TIMx_STS.CC1ITF 被设置时 (即使它已经是高电平), 即捕获或比较成功时, 触发输出发送一个正脉冲 (TRGO)。 x100: 比较 - OC1REF 信号用作触发输出 (TRGO)。 x101: 比较 - OC2REF 信号用作触发输出 (TRGO)。 x110: 比较 - OC3REF 信号用作触发输出 (TRGO)。 x111: 比较 - OC4REF 信号用作触发输出 (TRGO)。
11:7	Reserved	保留, 必须保持复位值
6	OI4	输出空闲状态4 (OC4输出) (Output Idle state 4) 0: 当MOEN=0时, 如果实现了OC4, 则OC4=0; 1: 当MOEN=0时, 如果实现了OC4, 则OC4=1。 <i>注: 已经设置了LCKCFG (TIMx_BKDT寄存器) 级别1、2或3后, 该位不能被修改。</i>
5	Reserved	保留, 必须保持复位值
4	OI3	输出空闲状态3 (OC3输出) (Output Idle state 3) 0: 当MOEN=0时, 如果实现了OC3, 则OC3=0; 1: 当MOEN=0时, 如果实现了OC3, 则OC3=1。 <i>注: 已经设置了LCKCFG (TIMx_BKDT寄存器) 级别1、2或3后, 该位不能被修改。</i>
3:0	Reserved	保留, 必须保持复位值

### 13.4.4 状态寄存器 (TIMx\_STS)

偏移地址: 0x08

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved				IOMBITF	Reserved				COMPBITF	PBKPITF	LBKPITF	Reserved	BITF	TITF	Reserved	UDITF
				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	
Reserved				CC4OCF	CC3OCF	CC2OCF	CC1OCF	Reserved				CC4ITF	CC3ITF	CC2ITF	CC1ITF	
				rc_w0	rc_w0	rc_w0	rc_w0					rc_w0	rc_w0	rc_w0	rc_w0	

位域	名称	描述
31:28	Reserved	保留, 必须保持复位值
27	IOMBITF	IOM 刹车中断标记 (IOM Break interrupt flag) 一旦 IOM 刹车输入有效, 由硬件对该位置'1'。如果刹车输入无效, 则该位可由软件清'0'。 0: 无刹车事件产生; 1: IOM 刹车输入上检测到有效电平。
26:24	Reserved	保留, 必须保持复位值
23	COMPBITF	COMP 刹车中断标记 (COMP Break interrupt flag) 一旦 COMP 刹车输入有效, 由硬件对该位置'1'。如果刹车输入无效, 则该位可由软件清'0'。 0: 无刹车事件产生; 1: COMP 刹车输入上检测到有效电平。
22	PBKPITF	PVD 刹车中断标记 (PVD Break interrupt flag) 一旦PVD刹车输入有效, 由硬件对该位置'1'。如果刹车输入无效, 则该位可由软件清'0'。 0: 无刹车事件产生; 1: PVD 刹车输入上检测到有效电平。
21	LBKPITF	Lockup 刹车中断标记 (Lockup Break interrupt flag) 一旦 Lockup 刹车输入有效, 由硬件对该位置'1'。如果刹车输入无效, 则该位可由软件清'0'。 0: 无刹车事件产生; 1: Lockup 刹车输入上检测到有效电平。
20	Reserved	保留, 必须保持复位值
19	BITF	刹车中断标记 (Break interrupt flag) 一旦刹车输入有效, 由硬件对该位置'1'。如果刹车输入无效, 则该位可由软件清'0'。 0: 无刹车事件产生; 1: 刹车输入上检测到有效电平。
18	TITF	触发器中断标记 (Trigger interrupt flag) 当发生触发事件 (当从模式控制器处于除门控模式外的其它模式时, 在TRGI输入端检测到有效边沿, 或门控模式下的任一边沿) 时由硬件对该位置'1'。它由软件清'0'。 0: 无触发器事件产生; 1: 触发中断等待响应。
17	Reserved	保留, 必须保持复位值
16	UDITF	更新中断标志 (Update interrupt flag) 当在以下条件下发生更新事件时, 该位由硬件设置: - 当 TIMx_CTRL1.UPDIS = 0 时, 并且重复计数器值上溢或下溢 (当重复计数器等于 0 时生成更新事件UEV)。 - 当 TIMx_CTRL1.UPRS = 0 时, TIMx_CTRL1.UPDIS = 0, 并通过软件设置 TIMx_EVTGEN.UDGN 位以重新初始化 CNT。 - 当 TIMx_CTRL1.UPRS = 0 时, TIMx_CTRL1.UPDIS = 0, 并且计数器 CNT 由触发

位域	名称	描述
		事件重新初始化。（参见 TIMx_SMCTRL 寄存器说明） 该位由软件清零。 0：未发生更新事件 1：发生更新中断
15:12	Reserved	保留，必须保持复位值
11	CC4OCF	捕获/比较4重复捕获标记（Capture/Compare 4 overcapture flag） 参见CC1OCF描述。
10	CC3OCF	捕获/比较3重复捕获标记（Capture/Compare 3 overcapture flag） 参见CC1OCF描述。
9	CC2OCF	捕获/比较2重复捕获标记（Capture/Compare 2 overcapture flag） 参见CC1OCF描述。
8	CC1OCF	捕获/比较1重复捕获标记（Capture/Compare 1 overcapture flag） 仅当相应的通道被配置为输入捕获时，该标记可由硬件置1。写0可清除该位。 0：无重复捕获产生； 1：计数器的值被捕获到TIMx_CCDAT1寄存器时，CC1ITF的状态已经为‘1’。
7:4	Reserved	保留，必须保持复位值
3	CC4ITF	捕获/比较4中断标记（Capture/Compare 4 interrupt flag） 参考CC1ITF描述。
2	CC3ITF	捕获/比较3中断标记（Capture/Compare 3 interrupt flag） 参考CC1ITF描述。
1	CC2ITF	捕获/比较2中断标记（Capture/Compare 2 interrupt flag） 参考CC1ITF描述。
0	CC1ITF	捕获/比较1中断标记（Capture/Compare 1 interrupt flag） <b>如果通道CC1配置为输出模式：</b> 除中央对齐模式外，当计数器值与比较值相同时，该位由硬件设置（参见TIMx_CTRL1.CAMSEL 位描述）。该位由软件清零。 0：未发生匹配。 1：TIMx_CNT 的值与 TIMx_CCDAT1 的值相同。 当 TIMx_CCDAT1 的值大于 TIMx_AR 的值时，如果计数器溢出（在向上计数和向上/向下计数模式下）和向下计数模式下溢，则 TIMx_STS.CC1ITF 位将变为高电平。 <b>如果通道CC1配置为输入模式：</b> 当捕捉事件发生时，该位由硬件设置。该位由软件或读取 TIMx_CCDAT1 清零。 0：未发生输入捕捉。 1：发生输入捕捉。计数器值已在 TIMx_CCDAT1 中捕获。在 IC1 上检测到与所选极性相同的边沿。

### 13.4.5 事件产生寄存器（TIMx\_EVTGEN）

偏移地址:0x0C

复位值:0x0000 0000

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

Reserved															
----------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

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

Reserved	TGN	Reserved	UDGN	Reserved	CC4GN	CC3GN	CC2GN	CC1GN
	w		w		w	w	w	w

位域	名称	描述
31:11	Reserved	保留，必须保持复位值
10	TGN	产生触发事件（Trigger generation） 当由软件置位时，该位可以产生一个触发事件。而此时TIMx_STS.TITF = 1，如果相应的中断和DMA被使能，就会产生相应的中断和DMA。该位由硬件自动清零。 0：无动作 1：产生触发事件
9	Reserved	保留，必须保持复位值
8	UDGN	产生更新事件（Update generation）该位由软件置‘1’，由硬件自动清‘0’。 当由软件设置时，该位可以生成更新事件。而此时计数器会重新初始化，预分频计数器会被清零，计数器在中央对齐或向上计数模式下会被清零，但在向下计数模式下取TIMx_AR寄存器的值。该位由硬件自动清零。 0：无动作 1：生成更新事件
7:4	Reserved	保留，必须保持复位值
3	CC4GN	产生捕获/比较4事件（Capture/Compare 4 generation） 参考CC1GN描述。
2	CC3GN	产生捕获/比较3事件（Capture/Compare 3 generation） 参考CC1GN描述。
1	CC2GN	产生捕获/比较2事件（Capture/Compare 2 generation） 参考CC1GN描述。
0	CC1GN	产生捕获/比较1事件（Capture/Compare 1 generation） 当由软件设置时，该位可以产生一个捕获/比较事件。该位由硬件自动清零。 <b>CC1对应通道为输出模式时：</b> TIMx_STS.CC1ITF 标志将被拉高，如果相应的中断和 DMA 被使能，就会产生相应的中断和 DMA。 <b>CC1对应通道为输入模式时：</b> TIMx_CC1DAT1 将捕获当前计数值，并将 TIMx_STS.CC1ITF 标志拉高，如果相应的中断和 DMA 被使能，则会产生相应的中断和 DMA。如果 TIMx_STS.CC1ITF 已经拉高，则拉高 TIMx_STS.CC1OCF。 0：无动作 1：生成 CC1 捕获/比较事件

### 13.4.6 从模式控制寄存器 (TIMx\_SMCTRL)

偏移地址: 0x10

复位值: 0x0000 0000

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

位域	名称	描述
31:17	Reserved	保留, 必须保持复位值
16	MSMD	主/从模式 (Master/slave mode) 0: 无作用; 1: 触发输入 (TRGI) 上的事件被延迟了, 以允许在当前定时器 (通过TRGO) 与它的从定时器间的完美同步。这对要求把几个定时器同步到一个单一的外部事件时是非常有用的。
15:12	EXTF[3:0]	外部触发滤波 (External trigger filter) 这些位定义了ETRP数字滤波器长度(RCC_TIMFILTCFG.TIM3FILTCLK[4:0] 寄存器决定相应数字滤波器的采样频率)。数字滤波器由一个事件计数器组成, 它记录到N个事件后会产生一个输出的跳变: 0000: 无滤波器, 以TIM内部工作时钟采样 0001: N=1 0010: N=2 0011: N=3 0100: N=4 0101: N=5 0110: N=6 0111: N=7 1xxx: N=8
11	EXTP	外部触发极性 (External trigger polarity) 该位选择是用ETR还是ETR的反相来作为触发操作 0: ETR高电平或上升沿有效; 1: ETR低电平或下降沿有效。
10	EXCEN	外部时钟使能位 (External clock enable) 该位启用外部时钟模式2。启用后, 计数器由ETRF信号上的任意有效边沿驱动。 0: 禁止外部时钟模式2; 1: 使能外部时钟模式2。 <i>注 1: 当同时使能外部时钟模式 1 和外部时钟模式 2 时, 外部时钟的输入为 ETRF。</i>

		<p>注2：以下从机模式可以与外部时钟模式2同时使用：复位模式、门控模式和触发模式；但是，TRGI 无法连接到 ETRF (<math>TIMx\_SMCTRL.TSEL \neq '111'</math>)。</p> <p>注 3：设置 <math>TIMx\_SMCTRL.EXCEN</math> 位与选择外部时钟模式 1 并将 TRGI 连接到 ETRF (<math>TIMx\_SMCTRL.SMSEL = 0111</math> 和 <math>TIMx\_SMCTRL.TSEL = 111</math>) 的效果相同</p>
9:8	EXTPS[1:0]	<p>外部触发预分频 (External trigger prescaler)</p> <p>外部触发信号 ETRP 的频率必须最多为 TIMxCLK 频率的 1/4。当输入更快的外部时钟时，可以使用预分频器来降低 ETRP 的频率。</p> <p>00：关闭预分频；</p> <p>01：ETRP频率除以2；</p> <p>10：ETRP频率除以4；</p> <p>11：ETRP频率除以8。</p>
7:4	SMSEL[3:0]	<p>从模式选择 (Slave mode selection)</p> <p>当选择了外部信号，触发信号 (TRGI) 的有效边沿与选中的外部输入极性相关 (见输入控制寄存器和控制寄存器的说明)</p> <p>0000：关闭从模式 – 如果CNTEN=1，则预分频器直接由内部时钟驱动。</p> <p>0001：编码器模式1 – 根据TI2FP2的电平，计数器在TI1FP1的边沿向上/下计数。</p> <p>0010：编码器模式2 – 根据TI1FP1的电平，计数器在TI2FP2的边沿向上/下计数。011：编码器模式3 – 根据另一个信号的输入电平，计数器在TI1FP1和TI2FP2的边沿向上/下计数。</p> <p>0100：复位模式 – 在选定触发输入 (TRGI) 的上升沿，计数器重新初始化并更新影子寄存器。</p> <p>0101：门控模式 – 当触发输入 (TRGI) 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 (但不复位)。计数器的启动和停止都是受控的。</p> <p>0110：触发模式 – 计数器在触发输入TRGI的上升沿启动 (但不复位)，只有计数器的启动是受控的。</p> <p>0111：外部时钟模式1 – 选中的触发输入 (TRGI) 的上升沿驱动计数器。</p> <p>1000~1101：保留。</p> <p>1110：组合复位+触发模式 – 计数器在触发输入TRGI的上升沿启动 (且复位)，只有计数器的启动是受控的。</p> <p>1111：保留。</p> <p>注：如果TI1F_ED被选为触发输入 (<math>TSEL=100</math>) 时，不要使用门控模式。这是因为，TI1F_ED在每次TI1F变化时输出一个脉冲，然而门控模式是要检查触发输入的电平。</p>
3	Reserved	保留，必须保持复位值
2:0	TSEL[2:0]	<p>触发选择 (Trigger selection)</p> <p>这3位选择用于同步计数器的触发输入。</p> <p>000：内部触发0 (ITR0) 100：TI1的边沿检测器 (TI1F_ED)</p> <p>001：内部触发1 (ITR1) 101：滤波后的定时器输入1 (TI1FP1)</p> <p>010：内部触发2 (ITR2) 110：滤波后的定时器输入2 (TI2FP2)</p> <p>011：内部触发3 (ITR3) 111：外部触发输入 (ETRF)</p> <p>更多有关ITRx的细节，参见表13-3。</p> <p>注：这些位只能在未用到 (如SMSEL=0000) 时被改变，以避免在改变时产生错误的边沿检测。</p> <p>注：开启外部事件 (TRGI) 触发计数器装载值功能时，如果同时使用输出比较模式，必须选择 ETRF 或者ITRx作为触发源。</p>

表 13-3 TIMx 内部触发连接

Slave timer	ITR0 (TSEL = 000)	ITR1 (TSEL = 001)	ITR2 (TSEL = 010)	ITR3 (TSEL = 011)
TIM1	TIM3	NA	NA	NA
TIM3	TIM1	NA	NA	NA
TIM4	NA	NA	NA	NA

### 13.4.7 DMA/中断使能寄存器 (TIMx\_DINTEN)

偏移地址: 0x14

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										TDEN	Reserved	UDEN	BIEN	TIEN	UIEN
										rw		rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CC4DEN	CC3DEN	CC2DEN	CC1DEN	Reserved				CC4IEN	CC3IEN	CC2IEN	CC1IEN
				rw	rw	rw	rw					rw	rw	rw	rw

位域	名称	描述
31:22	Reserved	保留, 必须保持复位值
21	TDEN	允许触发DMA请求 (Trigger DMA request enable) 0: 禁止触发DMA请求; 1: 允许触发DMA请求。
20	Reserved	保留, 必须保持复位值
19	UDEN	允许更新的DMA请求 (Update DMA request enable) 0: 禁止更新的DMA请求; 1: 允许更新的DMA请求。
18	BIEN	允许刹车中断 (Break interrupt enable) 0: 禁止刹车中断; 1: 允许刹车中断。
17	TIEN	触发中断使能 (Trigger interrupt enable) 0: 禁止触发中断; 1: 使能触发中断。
16	UIEN	允许更新中断 (Update interrupt enable) 0: 禁止更新中断; 1: 允许更新中断。
15:12	Reserved	保留, 必须保持复位值

位域	名称	描述
11	CC4DEN	允许捕获/比较4的DMA请求 (Capture/Compare 4 DMA request enable) 0: 禁止捕获/比较4的DMA请求; 1: 允许捕获/比较4的DMA请求。
10	CC3DEN	允许捕获/比较3的DMA请求 (Capture/Compare 3 DMA request enable) 0: 禁止捕获/比较3的DMA请求; 1: 允许捕获/比较3的DMA请求。
9	CC2DEN	允许捕获/比较2的DMA请求 (Capture/Compare 2 DMA request enable) 0: 禁止捕获/比较2的DMA请求; 1: 允许捕获/比较2的DMA请求。
8	CC1DEN	允许捕获/比较1的DMA请求 (Capture/Compare 1 DMA request enable) 0: 禁止捕获/比较1的DMA请求; 1: 允许捕获/比较1的DMA请求。
7:4	Reserved	保留, 必须保持复位值
3	CC4IEN	允许捕获/比较4中断 (Capture/Compare 4 interrupt enable) 0: 禁止捕获/比较4中断; 1: 允许捕获/比较4中断。
2	CC3IEN	允许捕获/比较3中断 (Capture/Compare 3 interrupt enable) 0: 禁止捕获/比较3中断; 1: 允许捕获/比较3中断。
1	CC2IEN	允许捕获/比较2中断 (Capture/Compare 2 interrupt enable) 0: 禁止捕获/比较2中断; 1: 允许捕获/比较2中断。
0	CC1IEN	允许捕获/比较1中断 (Capture/Compare 1 interrupt enable) 0: 禁止捕获/比较1中断; 1: 允许捕获/比较1中断。

### 13.4.8 捕获/比较模式寄存器 1 (TIMx\_CCMOD1)

偏移地址：0x18

复位值：0x0000 0000

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

**输出比较模式：**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2MD[2:0]			OC2CEN	OC2FEN	OC2PEN	CC2SEL[1:0]		OC1MD[2:0]			OC1CEN	OC1FEN	OC1PEN	CC1SEL[1:0]	
rw			rw	rw	rw	rw		rw			rw	rw	rw	rw	

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:13	OC2MD[2:0]	输出比较2模式（Output Compare 2 mode）
12	OC2CEN	输出比较2清0使能（Output Compare 2 clear enable）
11	OC2FEN	输出比较2快速使能（Output Compare 2 fast enable）
10	OC2PEN	输出比较2预装载使能（Output Compare 2 preload enable）
9:8	CC2SEL[1:0]	捕获/比较2选择。（Capture/Compare 2 selection） 该位定义通道的方向（输入/输出），及输入脚的选择： 00：CC2通道被配置为输出； 01：CC2通道被配置为输入，IC2映射在TI2上； 10：CC2通道被配置为输入，IC2映射在TI1上； 11：CC2通道被配置为输入，IC2映射在TRC上。此模式仅工作在内部触发输入被选中时（由TIMx_SMCTRL寄存器的TSEL位选择）。 <i>注：CC2SEL仅在通道关闭时（TIMx_CCEN寄存器的CC2EN=0）才是可写的。</i>
7:5	OC1MD[2:0]	输出比较1模式（Output Compare 1 mode） 这些位用于管理输出参考信号 OC1REF，它决定了 OC1 和 OC1N 的值，在高电平有效，而 OC1 和 OC1N 的有效电平取决于 TIMx_CCEN.CC1P 和 TIMx_CCEN.CC1NP 位。 000：冻结。TIMx_CCDAT1 寄存器和计数器 TIMx_CNT 之间的比较对 OC1REF 信号没有影响。 001：将通道 1 设置为匹配时的有效电平。当 TIMx_CCDAT1 = TIMx_CNT 时，OC1REF 信号将被强制为高电平。 010：将通道 1 设置为匹配时的无效电平。当 TIMx_CCDAT1 = TIMx_CNT 时，OC1REF 信号将被强制为低电平。

位域	名称	描述
		011: 翻转。当 $TIMx\_CCDAT1 = TIMx\_CNT$ 时, OC1REF 信号将被翻转。 100: 强制无效电平。 OC1REF 信号被强制为低电平。 101: 强制有效电平。 OC1REF 信号被强制为高电平。 110: PWM 模式 1 - 在向上计数模式下, 如果 $TIMx\_CNT < TIMx\_CCDAT1$ , 则通道 1 的 OC1REF 信号为高电平, 否则为低电平。在向下计数模式下, 如果 $TIMx\_CNT > TIMx\_CCDAT1$ , 则通道 1 的 OC1REF 信号为低电平, 否则为高电平。 111: PWM 模式 2 - 在向上计数模式下, 如果 $TIMx\_CNT < TIMx\_CCDAT1$ , 则通道 1 的 OC1REF 信号为低电平, 否则为高电平。在向下计数模式下, 如果 $TIMx\_CNT > TIMx\_CCDAT1$ , 则通道 1 的 OC1REF 信号为高电平, 否则为低电平。 <i>注 1: 在 PWM 模式 1 或 PWM 模式 2 中, OC1REF 电平仅在比较结果改变或输出比较模式从冻结模式切换到 PWM 模式时才会改变。</i>
4	OC1CEN	输出比较1清'0'使能 (Output Compare 1 clear enable) 0: OC1REF 不受ETRF输入的影响; 1: 一旦检测到ETRF输入高电平, 清除OC1REF=0。
3	OC1FEN	输出比较1 快速使能 (Output Compare 1 fast enable) 该位用于加快CC输出对触发输入事件的响应。 0: 根据计数器与CCDAT1的值, CC1正常操作, 即使触发器是打开的。当触发器的输入有一个有效沿时, 激活CC1输出的最小延时为5个时钟周期。 1: 输入到触发器的有效沿的作用就像发生了一次比较匹配。因此, OC1被设置为比较电平而与比较结果无关。采样触发器的有效沿和CC1输出间的延时被缩短为3个时钟周期。 OCxPEN只在通道被配置成PWM1或PWM2模式时起作用。
2	OC1PEN	输出比较 1 预加载使能 (Output Compare 1 preload enable) 0: 禁用 $TIMx\_CCDAT1$ 寄存器的预加载功能。 支持随时对 $TIMx\_CCDAT1$ 寄存器进行写操作, 写入的值立即生效。 1: 使能 $TIMx\_CCDAT1$ 寄存器的预加载功能。 仅对预加载寄存器进行读写操作。当更新事件发生时, $TIMx\_CCDAT1$ 的值被加载到影子寄存器中。 <i>注 1: 只有当 <math>TIMx\_CTRL1.ONEPM = 1</math> (在单脉冲模式下) 时, 才能使用 PWM 模式而不验证预加载寄存器, 否则无法预测其他行为。</i>
1:0	CC1SEL[1:0]	捕获/比较1 选择。(Capture/Compare 1 selection) 这2位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC1通道被配置为输出; 01: CC1通道被配置为输入, IC1映射在TI1上; 10: CC1通道被配置为输入, IC1映射在TI2上; 11: CC1通道被配置为输入, IC1映射在TRC上。此模式仅工作在内部触发输入被选中时(由 $TIMx\_SMCTRL$ 寄存器的TSEL位选择)。 <i>注: CC1SEL 仅在通道关闭时 (<math>TIMx\_CCEN</math>寄存器的CC1EN=0) 才是可写的。</i>

**输入捕获模式：**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC2F[3:0]				IC2PSC[1:0]		CC2SEL[1:0]		IC1F[3:0]			IC1PSC[1:0]		CC1SEL[1:0]		
rw				rw		rw		rw			rw		rw		

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:12	IC2F[3:0]	输入捕获2滤波器（Input capture 2 filter）
11:10	IC2PSC[1:0]	输入/捕获2预分频器（Input capture 2 prescaler）
9:8	CC2SEL[1:0]	捕获/比较2选择（Capture/Compare 2 selection） 这2位定义通道的方向（输入/输出），及输入脚的选择： 00：CC2通道被配置为输出； 01：CC2通道被配置为输入，IC2映射在TI2上； 10：CC2通道被配置为输入，IC2映射在TI1上； 11：CC2通道被配置为输入，IC2映射在TRC上。此模式仅工作在内部触发输入被选中时（由TIMx_SMCTRL寄存器的TSEL位选择）。 <i>注：CC2SEL仅在通道关闭时（TIMx_CCEN寄存器的CC2EN=0）才是可写的。</i>
7:4	IC1F[3:0]	输入捕获1滤波器（Input capture 1 filter） 这几位定义了TI1数字滤波器长度(RCC_TIMFILTCFG.TIM3FILTCLK[4:0] 寄存器决定相应数字滤波器的采样频率)。数字滤波器由一个事件计数器组成，它记录到N个事件后会产生一个输出的跳变： 0000：无滤波器，以TIM内部工作时钟采样 0001：N=1 0010：N=2 0011：N=3 0100：N=4 0101：N=5 0110：N=6 0111：N=7 1xxx：N=8
3:2	IC1PSC[1:0]	输入/捕获1预分频器（Input capture 1 prescaler） 这2位定义了CC1输入（IC1）的预分频系数。 一旦TIMx_CCEN.CC1EN=0，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	CC1SEL[1:0]	捕获/比较1选择 (Capture/Compare 1 Selection) 这2位定义通道的方向 (输入/输出), 及输入脚的选择: 00: CC1通道被配置为输出; 01: CC1通道被配置为输入, IC1映射在TI1上; 10: CC1通道被配置为输入, IC1映射在TI2上; 11: CC1通道被配置为输入, IC1映射在TRC上。此模式仅工作在内部触发输入被选中时 (由TIMx_SMCTRL寄存器的TSEL位选择)。 <i>注: CC1SEL仅在通道关闭时 (TIMx_CCEN寄存器的CCIEN=0) 才是可写的。</i>

### 13.4.9 捕获/比较模式寄存器 2 (TIMx\_CCMOD2)

偏移地址: 0x1C

复位值: 0x0000 0000

参看以上 CCMOD1 寄存器的描述

输出比较模式:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4MD[2:0]		OC4CEN	OC4FEN	OC4PEN	CC4SEL[1:0]		OC3MD[2:0]		OC3CEN	OC3FEN	OC3PEN	CC3SEL[1:0]			
rw		rw	rw	rw	rw		rw		rw	rw	rw	rw			

位域	名称	描述
31:16	Reserved	保留, 必须保持复位值
15:13	OC4MD[2:0]	输出比较4模式 (Output compare 4 mode)
12	OC4CEN	输出比较4清0使能 (Output compare 4 clear enable)
11	OC4FEN	输出比较4快速使能 (Output compare 4 fast enable)
10	OC4PEN	输出比较4预装载使能 (Output compare 4 preload enable)
9:8	CC4SEL[1:0]	捕获/比较4选择 (Capture/Compare 4 selection) 该2位定义通道的方向 (输入/输出), 及输入脚的选择: 00: CC4通道被配置为输出; 01: CC4通道被配置为输入, IC4映射在TI4上; 10: CC4通道被配置为输入, IC4映射在TI3上; 11: CC4通道被配置为输入, IC4映射在TRC上。此模式仅工作在内部触发输入被选中时 (由TIMx_SMCTRL寄存器的TSEL位选择)。 <i>注: CC4SEL仅在通道关闭时 (TIMx_CCEN寄存器的CC4EN=0) 才是可写的。</i>
7:5	OC3MD[2:0]	输出比较3模式 (Output compare 3 mode)
4	OC3CEN	输出比较3清0使能 (Output compare 3 clear enable)

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
3	OC3FEN	输出比较3快速使能 (Output compare 3 fast enable)
2	OC3PEN	输出比较3预装载使能 (Output compare 3 preload enable)
1:0	CC3SEL[1:0]	捕获/比较3选择 (Capture/Compare 3 selection) 这2位定义通道的方向(输入/输出)，及输入脚的选择： 00: CC3通道被配置为输出； 01: CC3通道被配置为输入，IC3映射在TI3上； 10: CC3通道被配置为输入，IC3映射在TI4上； 11: CC3通道被配置为输入，IC3映射在TRC上。此模式仅工作在内部触发输入被选中时(由TIMx_SMCTRL寄存器的TSEL位选择)。 注: CC3SEL仅在通道关闭时(TIMx_CCEN寄存器的CC3EN=0)才是可写的。

**输入捕获模式:**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
IC4F[3:0]				IC4PSC[1:0]			CC4SEL[1:0]			IC3F[3:0]			IC3PSC[1:0]		CC3SEL[1:0]	
rw				rw			rw			rw			rw		rw	

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:12	IC4F[3:0]	输入捕获4滤波器 (Input capture 4 filter)
11:10	IC4PSC[1:0]	输入/捕获4预分频器 (Input capture 4 prescaler)
9:8	CC4SEL[1:0]	捕获/比较4选择 (Capture/Compare 4 selection) 这2位定义通道的方向(输入/输出)，及输入脚的选择： 00: CC4通道被配置为输出； 01: CC4通道被配置为输入，IC4映射在TI4上； 10: CC4通道被配置为输入，IC4映射在TI3上； 11: CC4通道被配置为输入，IC4映射在TRC上。此模式仅工作在内部触发输入被选中时(由TIMx_SMCTRL寄存器的TSEL位选择)。 注: CC4SEL仅在通道关闭时(TIMx_CCEN寄存器的CC4EN=0)才是可写的。
7:4	IC3F[3:0]	输入捕获3滤波器 (Input capture 3 filter)
3:2	IC3PSC[1:0]	输入/捕获3预分频器 (Input capture 3 prescaler)

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
1:0	CC3SEL[1:0]	捕获/比较3选择 (Capture/compare 3 selection) 这2位定义通道的方向(输入/输出)，及输入脚的选择： 00：CC3通道被配置为输出； 01：CC3通道被配置为输入，IC3映射在TI3上； 10：CC3通道被配置为输入，IC3映射在TI4上； 11：CC3通道被配置为输入，IC3映射在TRC上。此模式仅工作在内部触发输入被选中时(由TIMx_SMCTRL寄存器的TSEL位选择)。 注：CC3SEL仅在通道关闭时(TIMx_CCEN寄存器的CC3EN=0)才是可写的。

### 13.4.10 捕获/比较使能寄存器 (TIMx\_CCEN)

偏移地址：0x24

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4P	CC4EN	Reserved		CC3P	CC3EN	Reserved		CC2P	CC2EN	Reserved		CC1P	CC1EN	Reserved	
rw	rw			rw	rw			rw	rw			rw	rw		

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15	CC4P	捕获/比较4输出极性 (Capture/Compare 4 output polarity) 参考TIMx_CCEN.CC1P的描述。
14	CC4EN	捕获/比较4输出使能 (Capture/Compare 4 output enable) 参考TIMx_CCEN.CC1EN 的描述。
13:12	Reserved	保留，必须保持复位值
11	CC3P	捕获/比较3输出极性 (Capture/Compare 3 output polarity) 参考TIMx_CCEN.CC1P的描述。
10	CC3EN	捕获/比较3输出使能 (Capture/Compare 3 output enable) 参考TIMx_CCEN.CC1E 的描述。
9:8	Reserved	保留，必须保持复位值
7	CC2P	捕获/比较2输出极性 (Capture/Compare 2 output polarity) 参考TIMx_CCEN.CC1P的描述。
6	CC2EN	捕获/比较2输出使能 (Capture/Compare 2 output enable) 参考TIMx_CCEN.CC1EN的描述。
5:4	Reserved	保留，必须保持复位值
3	CC1P	捕获/比较1输出极性 (Capture/Compare 1 output polarity) CC1对应通道为输出模式时：

位域	名称	描述
		0: OC1 高电平有效 1: OC1 低电平有效 <b>CC1对应通道为输入模式时:</b> 此时, 该位用于选择是使用IC1还是IC1的反相信号作为触发信号或捕捉信号。 0: 非反相: 当 IC1 产生上升沿时发生捕获动作。 当用作外部触发时, IC1 是非反相的。 1: 反相: 当 IC1 产生下降沿时发生捕获动作。 当用作外部触发时, IC1 被反相。
2	CC1EN	捕获/比较1输出使能 (Capture/Compare 1 output enable) <b>CC1通道配置为输出:</b> 0: 关闭— OC1禁止输出, 因此OC1的输出电平依赖于MOEN、OSSI、OSSR、OI1、OI1N和CC1NEN位的值。 1: 开启— OC1信号输出到对应的输出引脚, 其输出电平依赖于MOEN、OSSI、OSSR、OI1、OI1N和CC1NEN位的值。 <b>CC1通道配置为输入:</b> 该位决定了计数器的值是否能捕获入TIMx_CC DAT1寄存器。 0: 捕获禁止; 1: 捕获使能。
1:0	Reserved	保留, 必须保持复位值

表 13-4 标准 OCx 的输出控制位

CCxEN	OCx output status
0	Disable output (OCx=0)
1	OCx = OCxREF + polarity

注: 连接到标准 OCx 通道的外部 I/O 引脚的状态取决于 OCx 通道状态以及 GPIO 和 AFIO 寄存器。

### 13.4.11 捕获/比较寄存器 1 (TIMx\_CC DAT1)

偏移地址: 0x28

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCDAT1[15:0]															

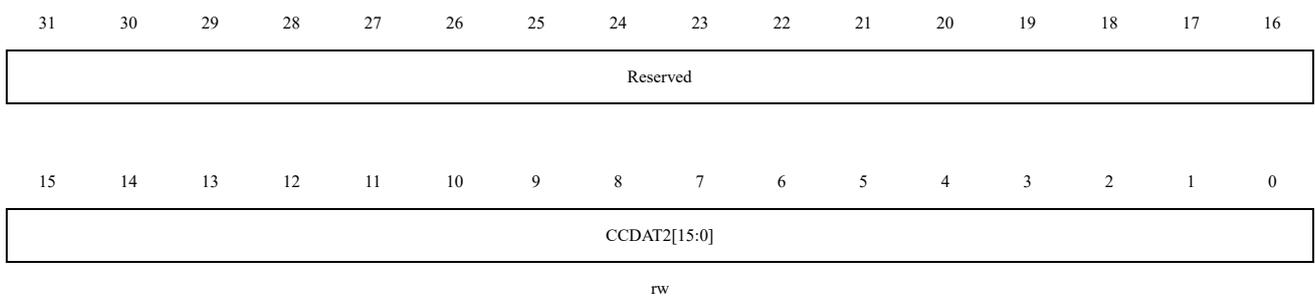
rw

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:0	CCDAT1[15:0]	捕获/比较通道1的值（Capture/Compare 1 value） <ul style="list-style-type: none"> <li>■ CC1 通道配置为输出： CCDAT1 包含要与计数器 TIMx_CNT 比较的值，在 OC1 输出上发出信号。 如果未在 TIMx_CCMOD1.OC1PEN 位中选择预加载功能，则写入的值会立即传输到有效寄存器。否则，仅当更新事件发生时，此预加载值才会传输到活动寄存器。</li> <li>■ CC1 通道配置为输入： CCDAT1 包含由最后一个输入捕获 1 事件 (IC1) 传输的计数器值。 当配置为输入模式时，寄存器 CCDAT1 和 CCDDAT1 只能读取。 当配置为输出模式时，寄存器 CCDAT1 和 CCDDAT1 是可读写的。</li> </ul>

### 13.4.12 捕获/比较寄存器 2 (TIMx\_CCDAT2)

偏移地址：0x2C

复位值：0x0000 0000



位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:0	CCDAT2[15:0]	捕获/比较通道2的值（Capture/Compare 2 value） <ul style="list-style-type: none"> <li>■ CC2 通道配置为输出： CCDAT2 包含要与计数器 TIMx_CNT 比较的值，在 OC2 输出上发出信号。 如果未在 TIMx_CCMOD1.OC2PEN 位中选择预加载功能，则写入的值会立即传输到有效寄存器。否则，仅当更新事件发生时，此预加载值才会传输到活动寄存器。</li> <li>■ CC2 通道配置为输入： CCDAT2 包含由最后一个输入捕获 2 事件 (IC2) 传输的计数器值。 当配置为输入模式时，寄存器 CCDAT2 和 CCDDAT2 只能读取。 当配置为输出模式时，寄存器 CCDAT2 和 CCDDAT2 是可读写的。</li> </ul>

### 13.4.13 捕获/比较寄存器 3 (TIMx\_CCDAT3)

偏移地址：0x30

复位值：0x0000 0000

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

Reserved

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

CCDAT3[15:0]

rw

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:0	CCDAT3[15:0]	捕获/比较通道3的值（Capture/Compare 3 value） <ul style="list-style-type: none"> <li>■ CC3 通道配置为输出： CCDAT3 包含要与计数器 TIMx_CNT 比较的值，在 OC3 输出上发出信号。 如果未在 TIMx_CCMOD2.OC3PEN 位中选择预加载功能，则写入的值会立即传输到有效寄存器。否则，仅当更新事件发生时，此预加载值才会传输到活动寄存器。</li> <li>■ CC3 通道配置为输入： CCDAT3 包含由最后一个输入捕获 3 事件 (IC3) 传输的计数器值。 当配置为输入模式时，寄存器 CCDAT3 和 CCDDAT3 只能读取。 当配置为输出模式时，寄存器 CCDAT3 和 CCDDAT3 是可读写的。</li> </ul>

### 13.4.14 捕获/比较寄存器 4 (TIMx\_CCDA4)

偏移地址：0x34

复位值：0x0000 0000

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

Reserved

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

CCDAT4[15:0]

rw

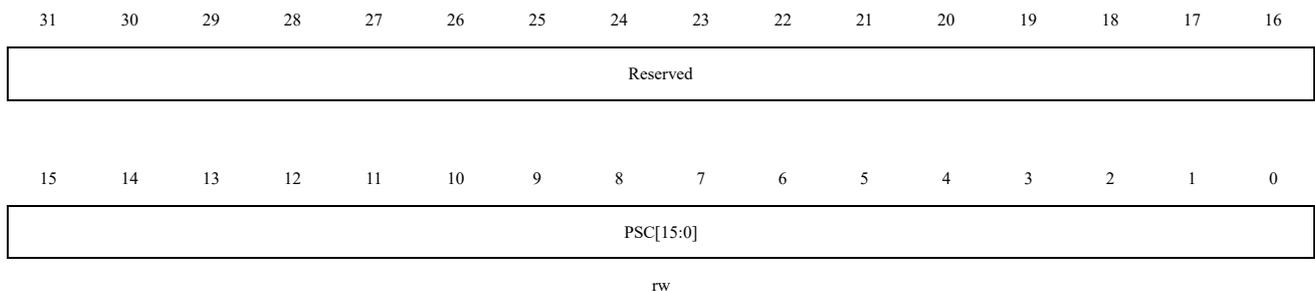
位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:0	CCDAT4[15:0]	捕获/比较通道4的值（Capture/Compare 4 value） <ul style="list-style-type: none"> <li>■ CC4 通道配置为输出： CCDAT4 包含要与计数器 TIMx_CNT 比较的值，在 OC4 输出上发出信号。 如果未在 TIMx_CCMOD2.OC4PEN 位中选择预加载功能，则写入的值会立即传输到有效寄存器。否则，仅当更新事件发生时，此预加载值才会传输到活动寄存器。</li> <li>■ CC4 通道配置为输入：</li> </ul>

		CCDAT4 包含由最后一个输入捕获 4 事件 (IC4) 传输的计数器值。 当配置为输入模式时, 寄存器 CCDAT4 和 CCDDAT4 只能读取。 当配置为输出模式时, 寄存器 CCDAT4 和 CCDDAT4 是可读写的。
--	--	---

### 13.4.15 预分频器 (TIMx\_PSC)

偏移地址: 0x40

复位值: 0x0000 0000

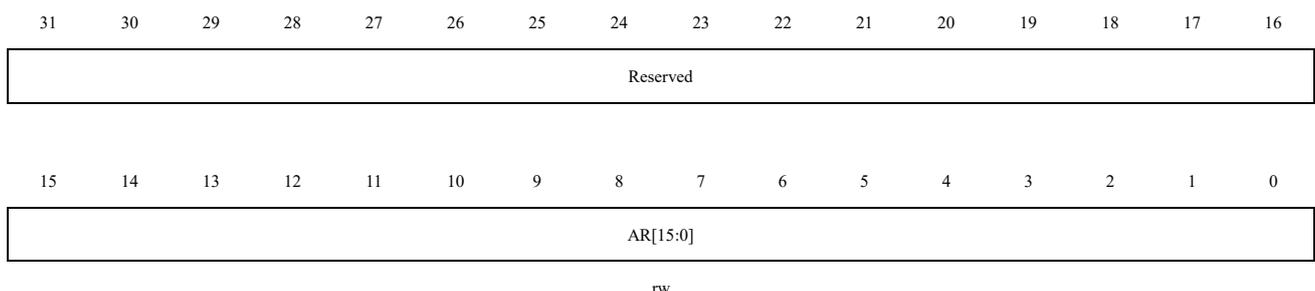


位域	名称	描述
31:16	Reserved	保留, 必须保持复位值
15:0	PSC[15:0]	预分频器的值 (Prescaler value) 计数器时钟 $f_{CK\_CNT} = f_{CK\_PSC} / (PSC [15:0] + 1)$ 。 每次发生更新事件时, PSC 值都会加载到预分频器的影子寄存器中。

### 13.4.16 自动重载寄存器 (TIMx\_AR)

偏移地址: 0x44

复位值: 0x0000 FFFF

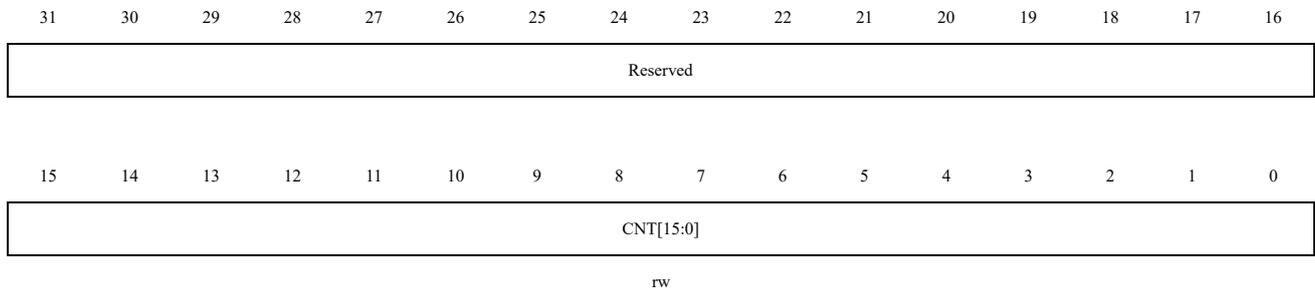


位域	名称	描述
31:16	Reserved	保留, 必须保持复位值
15:0	AR[15:0]	自动重载的值 (Auto-reload value) AR包含了将要装载入实际的自动重载寄存器的值。详细参考13.3.1节: 有关AR的更新和动作。 当自动重载的值为空时, 计数器不工作。

### 13.4.17 计数器 (TIMx\_CNT)

偏移地址: 0x48

复位值: 0x0000 0000

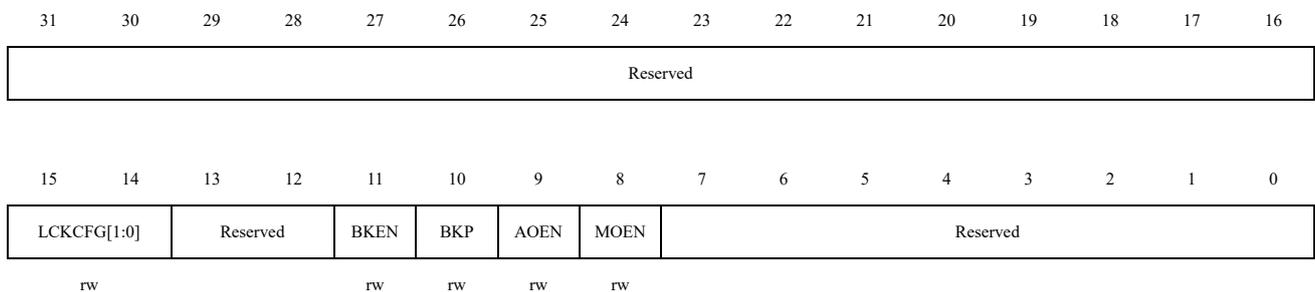


位域	名称	描述
31:16	Reserved	保留, 必须保持复位值
15:0	CNT[15:0]	计数器的值 (Counter value)

### 13.4.18 刹车和死区寄存器 (TIMx\_BKDT)

偏移地址: 0x50

复位值: 0x0000 0000



注意: 根据锁定设置, AOEN、BKP、BKEN、OSSI、OSSR 和 DTGN[7:0]位均可被写保护, 有必要在第一次写入 TIMx\_BKDT 寄存器时对它们进行配置。

位域	名称	描述
31:16	Reserved	保留, 必须保持复位值
15:14	LCKCFG[1:0]	锁定设置 (Lock configuration) 该位为防止软件错误而提供写保护。 这些位提供针对软件错误的写保护。 00: – 没有写保护。 01: – 锁定级别 1 TIMx_BKDT.DTGN、TIMx_BKDT.BKEN、TIMx_BKDT.BKP、TIMx_BKDT.AOEN、 TIMx_CTRL2.OIx、TIMx_CTRL2.OIxN、TIMx_AF1位启用写保护。 10:

		– 锁定 2 级 除了 LOCK Level 1 模式下的寄存器写保护外，TIMx_CCEN.CCxP 和 TIMx_CCEN.CCxNP（如果相应通道配置为输出模式），TIMx_BKDT.OSSR 和 TIMx_BKDT.OSSI 位也使能写保护。 11: – 锁定 3 级 除了 LOCK Level 2 中的寄存器写保护外，TIMx_CCMODx.OCxMD 和 TIMx_CCMODx.OCxPEN 位（如果相应通道配置为输出模式）也启用写保护。 注意：定时器复位后，LCKCFG 位只能写一次。一旦写入 TIMx_BKDT 寄存器，LCKCFG 将受到保护，直到下一次复位。
13:12	Reserved	保留，必须保持复位值
11	BKEN	刹车1功能使能（Break1 enable） 0：禁止刹车1输入； 1：开启刹车1输入。 注意：任何对该位的写操作都需要一个APB时钟的延迟以后才能起作用。
10	BKP	刹车1输入极性（Break1 polarity） 0：刹车1输入低电平有效； 1：刹车1输入高电平有效。 注意：任何对该位的写操作都需要一个APB时钟的延迟以后才能起作用。
9	AOEN	自动输出使能（Automatic output enable） 0：只有软件可以设置TIMx_BKDT.MOEN； 1：软件设置TIMx_BKDT.MOEN；或者如果刹车输入未激活，则在下一次更新事件发生时，硬件自动设置 TIMx_BKDT.MOEN。
8	MOEN	主输出使能（Main output enable） 该位可由软件或硬件根据 TIMx_BKDT.AOEN 位设置，一旦刹车输入有效，该位由硬件异步清零。它仅对配置为输出的通道3和通道4有效，对通道1和通道2无效。 0：OC 和 OCN 输出被禁用或强制进入空闲状态。 1：如果设置了 TIMx_CCEN.CCxEN 或 TIMx_CCEN.CCxNEN 位，则使能 OC 和 OCN 输出。有关更多详细信息，请参见第13.4.10节捕获/比较使能寄存器 (TIMx_CCEN)。
7:0	Reserved	保留，必须保持复位值

### 13.4.19 刹车滤波寄存器（TIMx\_BKFR）

偏移地址：0x60

复位值：0x0000 0000

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

位域	名称	描述
31:30	Reserved	保留，必须保持复位值
29:24	THRESH[5:0]	采样逻辑电平有效的阈值数(Threshold)，最大 63： 有效逻辑电平的阈值。在采样窗口内，如果逻辑高的数量大于或等于阈值，则下一个逻辑电平将为逻辑高。同样的规则适用于逻辑低。如果窗口内 1 和 0 的数量都小于阈值，则过滤器输出保持不变。阈值应设置为大于或等于 Window 值的一半。 推荐阈值范围为： <b>最小值：</b> 比最大毛刺大小的上限（预分频时钟周期）多 1 个预分频时钟周期，并且需要大于窗口大小的一半。 例如，如果毛刺大小为 3.2*（预分频时钟周期），则阈值应为 $\lceil 3.2 \rceil = 4 + 1 = 5$ <b>最大值：</b> 有效信号最小尺寸的底值（在预分频时钟周期内），需要小于窗口尺寸。 例如，如果最小信号大小为 3.2*（预分频时钟周期），则阈值应为下限 $(3.2) = 3$ 。
23	Reserved	保留，必须保持复位值
22:17	WSIZE[5:0]	逻辑电平检查的窗口大小值（Window size），最大 63： 窗口大小决定了在获得下一个逻辑级别时将考虑多少采样值。内置 FIFO 为 64 位，最大索引为 63，只能将窗口大小设置为 63。
16	FILTEN	滤波器使能（Filter enable）： 0: 滤波器禁能 1: 滤波器使能
15:0	Reserved	保留，必须保持复位值

### 13.4.20 复用功能寄存器 1（TIMx\_AF1）

偏移地址：0x7C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										IOM4BRK P	IOM3BRK P	IOM2BRK P	IOM4BRK EN	IOM3BRK EN	IOM2BRK EN
										rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					COMP1 BRKP	IOM BRKP	Reserved					COMP1 BRKEN	IOM BRKEN		
					rw	rw						rw	rw		

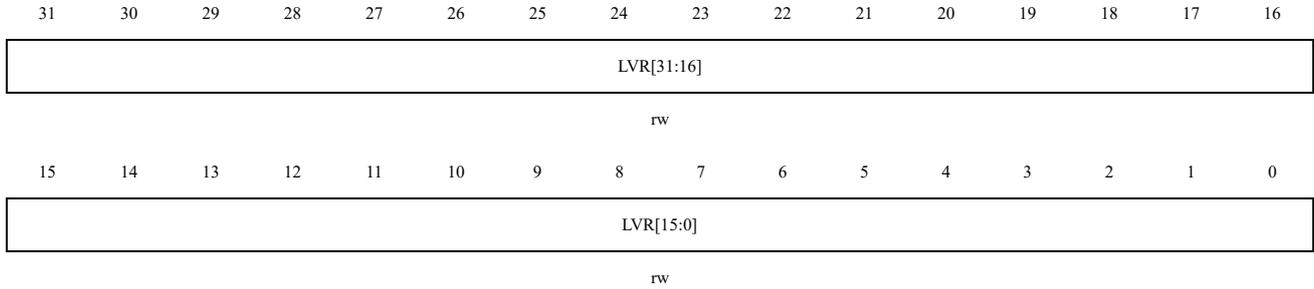
位域	名称	描述
31:22	Reserved	保留，必须保持复位值
21	IOM4BRKP	TIMx_BKIN IO4刹车输入极性选择 0: TIMx_BKIN IO4刹车输入极性不翻转（如果BKP=0,则低电平有效；如果BKP=1,则高电平有效）

		1: TIMx_BKIN IO4刹车输入极性翻转（如果BKP=0,则高电平有效；如果BKP=1,则低电平有效）
20	IOM3BRKP	TIMx_BKIN IO3刹车输入极性选择 0: TIMx_BKIN IO3刹车输入极性不翻转（如果BKP=0,则低电平有效；如果BKP=1,则高电平有效） 1: TIMx_BKIN IO3刹车输入极性翻转（如果BKP=0,则高电平有效；如果BKP=1,则低电平有效）
19	IOM2BRKP	TIMx_BKIN IO2刹车输入极性选择 0: TIMx_BKIN IO2刹车输入极性不翻转（如果BKP=0,则低电平有效；如果BKP=1,则高电平有效） 1: TIMx_BKIN IO2刹车输入极性翻转（如果BKP=0,则高电平有效；如果BKP=1,则低电平有效）
18	IOM4BRKEN	TIMx_BKIN IO4刹车输入使能 0: TIMx_BKIN IO4刹车输入禁止 1: TIMx_BKIN IO4刹车输入使能
17	IOM3BRKEN	TIMx_BKIN IO3刹车输入使能 0: TIMx_BKIN IO3刹车输入禁止 1: TIMx_BKIN IO3刹车输入使能
16	IOM2BRKEN	TIMx_BKIN IO2刹车输入使能 0: TIMx_BKIN IO2刹车输入禁止 1: TIMx_BKIN IO2刹车输入使能
15:11	Reserved	保留，必须保持复位值
10	COMP1BRKP	tim_brk_comp1刹车输入极性选择 0: tim_brk_comp1刹车输入极性不翻转（如果BKP=0,则低电平有效；如果BKP=1,则高电平有效） 1: tim_brk_comp1刹车输入极性翻转（如果BKP=0,则高电平有效；如果BKP=1,则低电平有效）
9	IOMBRKP	TIMx_BKIN IO1刹车输入极性选择 0: TIMx_BKIN IO1刹车输入极性不翻转（如果BKP=0,则低电平有效；如果BKP=1,则高电平有效） 1: TIMx_BKIN IO1刹车输入极性翻转（如果BKP=0,则高电平有效；如果BKP=1,则低电平有效）
8:2	Reserved	保留，必须保持复位值
1	COMP1BRKEN	tim_brk_comp1刹车输入使能 0: tim_brk_comp1刹车输入禁止 1: tim_brk_comp1刹车输入使能
0	IOMBRKEN	TIMx_BKIN IO1刹车输入使能 0: TIMx_BKIN IO1刹车输入禁止 1: TIMx_BKIN IO1刹车输入使能

### 13.4.21 外部事件计数器装载值寄存器 (TIMx\_ENCLVR)

偏移地址: 0x90

复位值: 0x0000 0000

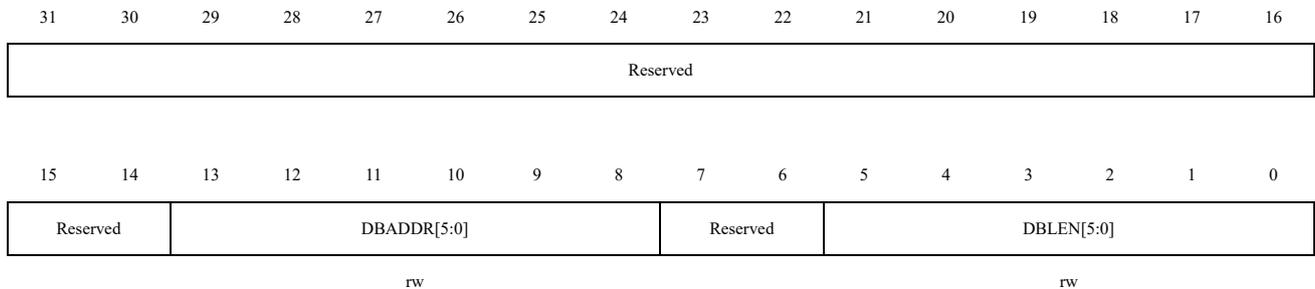


位域	名称	描述
31:0	LVR[31:0]	外部事件 (TRGI) 触发计数器装载值。 当检测到TRGI事件 (取决于TIMx_SMCTRL.TSEL[2:0]寄存器) 时, 计数器将装载为LVR值。

### 13.4.22 DMA 控制寄存器 (TIMx\_DCTRL)

偏移地址: 0x94

复位值: 0x0000 0000

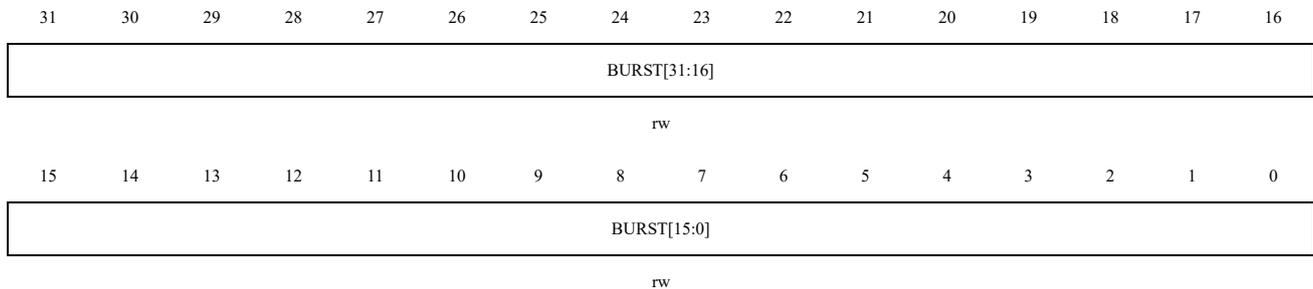


位域	名称	描述
31:14	Reserved	保留, 必须保持复位值
13:8	DBADDR[5:0]	DMA基地址 (DMA base address) 该位字段定义 DMA 访问 TIMx_DADDR 寄存器的第一个地址。 当第一次通过 TIMx_DADDR 完成访问时, 该位域指定您刚刚访问的地址。 然后第二次访问TIMx_DADDR, 会访问到“DMA Base Address + 4”的地址  000000: TIMx_CTRL1, 000001: TIMx_CTRL2, 000010: TIMx_STS, ..... 100101: TIMx_DCTRL
7:6	Reserved	保留, 必须保持复位值
5:0	DBLEN[5:0]	DMA连续传送长度 (DMA burst length) 该位字段定义 DMA 将访问 (写入/读取) TIMx_DADDR 寄存器的次数。  000000: 1次传输 000001: 2次传输 000010: 3次传输 ... 010001: 18次传输 ... 100010: 35次传输

### 13.4.23 连续模式的 DMA 地址 (TIMx\_DADDR)

偏移地址: 0x98

复位值: 0x0000 0000



位域	名称	描述
31:0	BURST[31:0]	<p>DMA 访问缓冲区。</p> <p>当对该寄存器分配读或写操作时，将访问位于地址范围 (DMA base address + DMA burst length × 4) 的寄存器。</p> <p>DMA base address = The address of TIM_CTRL1 + TIMx_DCTRL.DBADDR * 4;</p> <p>DMA burst len = TIMx_DCTRL.DBLEN + 1.</p> <p>例子:</p> <p>如果 TIMx_DCTRL.DBLEN = 0x3 (4 次传输)，TIMx_DCTRL.DBADDR = 0xD (TIMx_CC DAT1)，DMA 数据长度 = 半字，DMA 存储器地址 = SRAM 中的缓冲区地址，DMA 外设地址 = TIMx_DADDR 地址。</p> <p>当事件发生时，TIMx 将向 DMA 发送请求，并传输 4 次数据。</p> <p>第一次，对 TIMx_DADDR 寄存器的 DMA 访问将映射到访问 TIMx_CC DAT1 寄存器；</p> <p>第二次，对 TIMx_DADDR 寄存器的 DMA 访问将映射到访问 TIMx_CC DAT2 寄存器；</p> <p>.....</p> <p>第四次，对 TIMx_DADDR 寄存器的 DMA 访问将映射到访问 TIMx_CC DAT4 寄存器；</p>

## 14 通用定时器（TIM4）

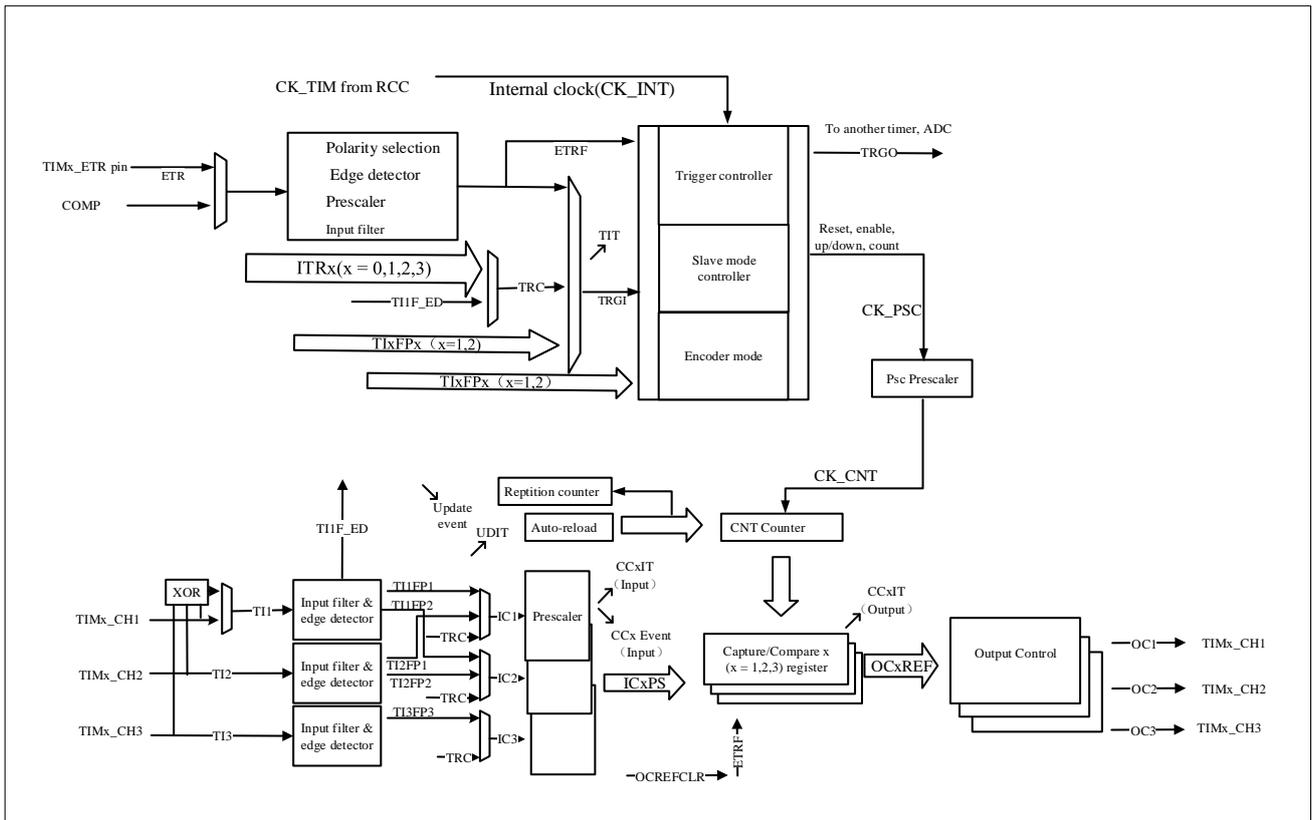
### 14.1 TIM4 简介

通用定时器（TIM4）主要用于以下场合：对输入信号进行计数、测量输入信号的脉冲宽度和产生输出波形等。

### 14.2 TIM4 主要特性

- 32 位自动装载计数器。（可实现向上计数、向下计数、向上/下计数）。
- 16 位可编程预分频器。（分频系数可配置为 1 到 65536 之间的任意值）
- TIM4 最多支持 3 个通道
- 通道工作模式：PWM 输出、输出比较、单脉冲模式输出、输入捕获
- 如下事件发生时产生中断/DMA：
  - ◆ 更新事件
  - ◆ 触发事件
  - ◆ 输入捕获
  - ◆ 输出比较
- 可通过外部信号控制定时器
- 多个定时器内部连接在一起，以实现定时器的同步或链接
- 增量（正交），脉冲+符号，CCW/CW 编码器接口：用于追踪运行轨迹和解析旋转方位
- 霍尔传感器接口：用于三相电机控制
- 支持捕获内部比较器输出信号。

图 14-1 TIM4 框图



↓ 事件                      ↗ 中断和 DMA 输出

捕获通道 1 输入可以来自 IOM 或比较器输出

## 14.3 TIM4 功能描述

### 14.3.1 时基单元

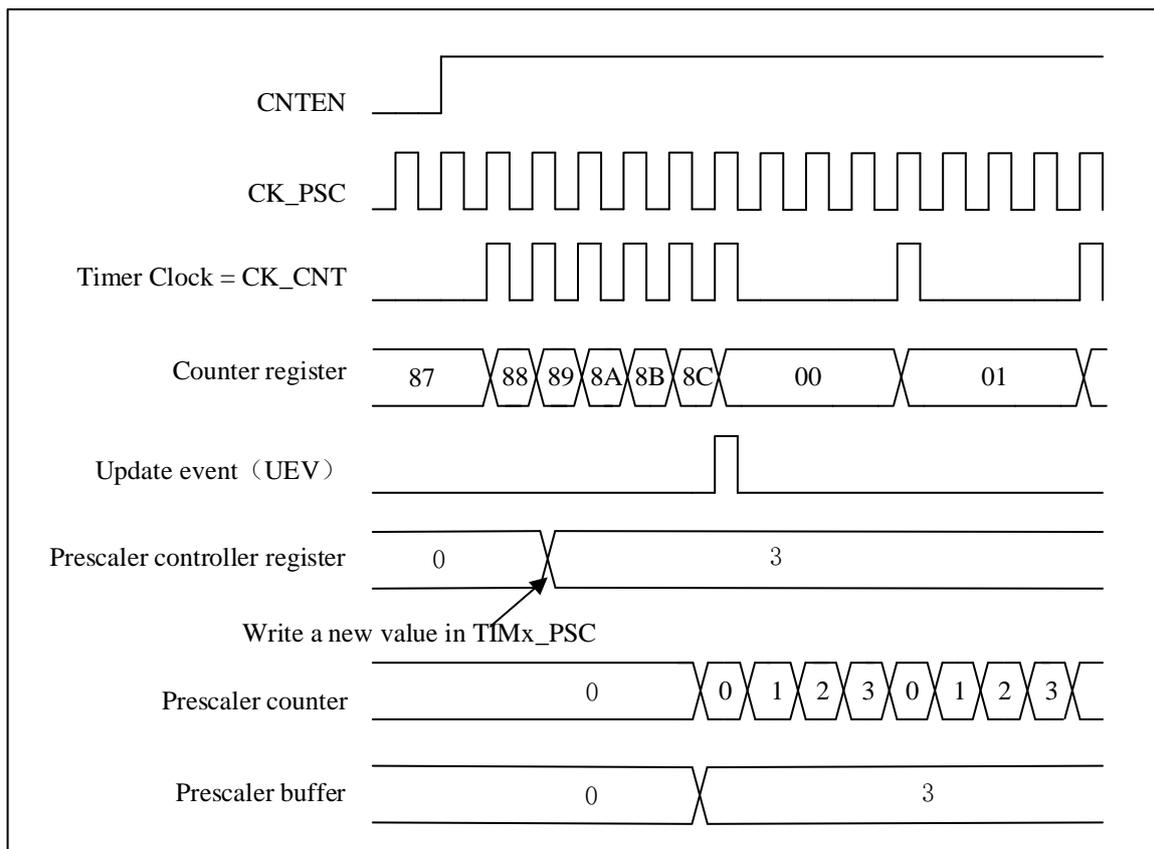
通用器的时基单元主要包括：预分频器、计数器和自动重载寄存器。当时基单元工作时，软件可以随时读取和写入相应的寄存器（TIMx\_PSC、TIMx\_CNT 和 TIMx\_AR）。

根据自动重载预装载使能位（TIMx\_CTRL1.ARPEN）的设置，预装载寄存器的值会立即或在每次更新事件 UEV 时传输到影子寄存器。TIMx\_CTRL1.UPDIS=0 时，当计数器上溢/下溢或软件设置 TIMx\_EVTGEN.UDGN，将生成更新事件。计数器 CK\_CNT 仅在 TIMx\_CTRL1.CNTEN 位被设置时有效。计数器在 TIMx\_CTRL1.CNTEN 位被设置后一个时钟周期之后开始计数。

#### 14.3.1.1 预分频器描述

TIMx\_PSC 寄存器由一个 16 位计数器组成，可用于计数器时钟频率按 1 和 65536 之间的任意分频。因为这个控制器带有缓冲器，可以在运行时动态改变。新的预分频器值只有在下次更新事件中才会被采用。

图 14-2 当预分频的参数从 1 到 4，计数器的时序图



## 14.3.2 计数器模式

### 14.3.2.1 向上计数模式

使用向上计数模式，计数器将从 0 计数到寄存器 TIMx\_AR 的值，然后重置为 0。并产生一个计数器溢出事件。

如果设置 TIMx\_CTRL1.UPRS 位(选择更新请求)和 TIMx\_EVTGEN.UDGN 位,将产生一个更新事件(UEV)。但是 TIMx\_STS.UDITF 不会被硬件置起，因此不会产生更新中断或 DMA 更新请求。这是为了避免清除计数器时产生更新中断。

取决于 TIMx\_CTRL1.UPRS 的配置，当发生更新事件时，TIMx\_STS.UDITF 被设置，所有寄存器都会更新：

- 当 TIMx\_CTRL1.ARPEN = 1，预装载寄存器(TIMx\_AR)的值被更新到自动装载影子寄存器
- 预加载值 (TIMx\_PSC) 被重新加载到预分频器影子寄存器中

为了避免在向预装载寄存器中写入新值时更新影子寄存器，可以通过设置 TIMx\_CTRL1.UPDIS=1 来禁止更新事件。

当产生一个更新事件时，计数器仍将被清除，预分频器计数器也将被设置为 0 (但预分频器值将保持不变)。

下图给出一些示例，展示了向上计数模式计数器在不同分频因子下的动作。

图 14-3 当内部时钟分频因子 = 2/N 时，向上计数的时序图

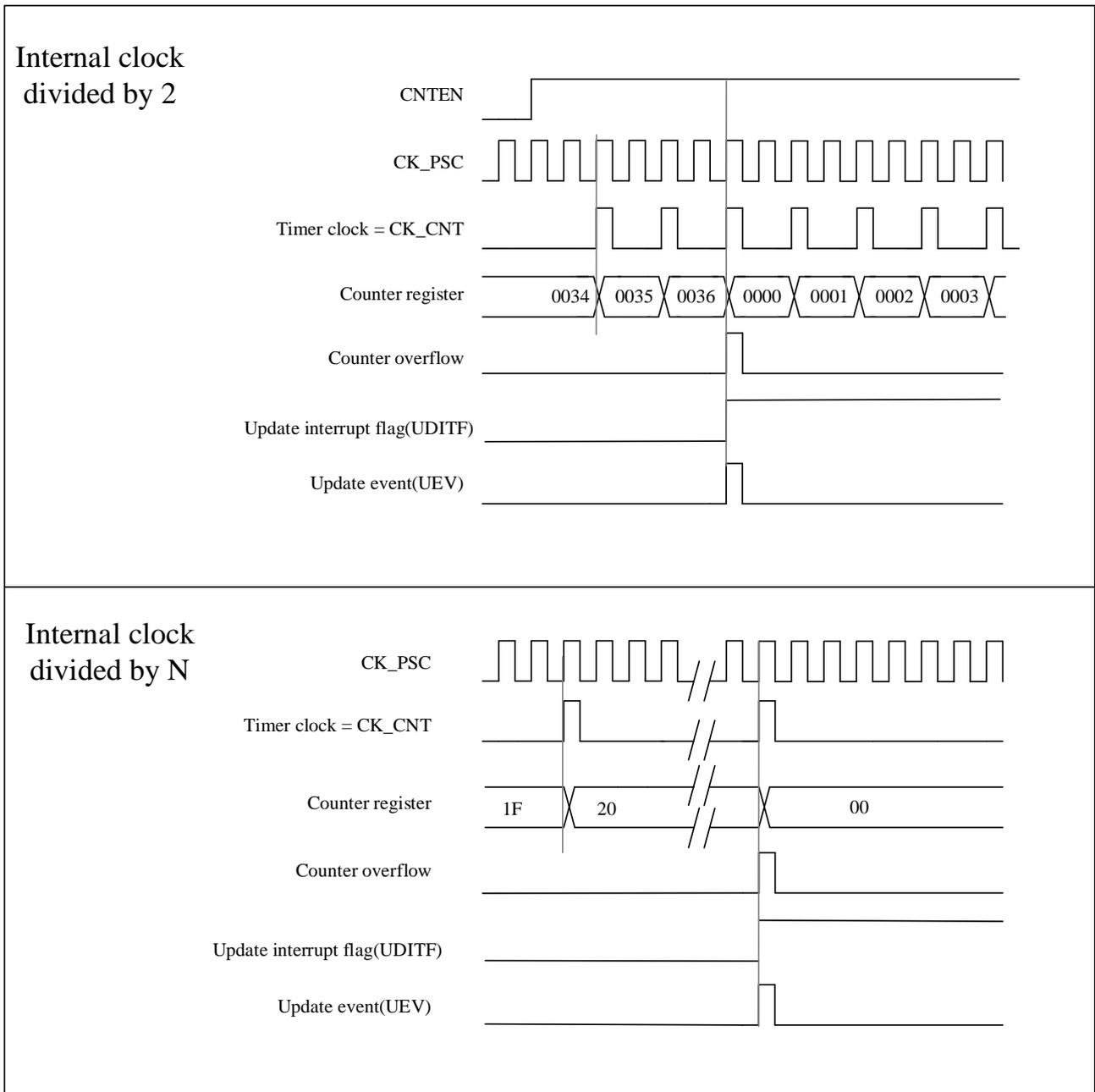
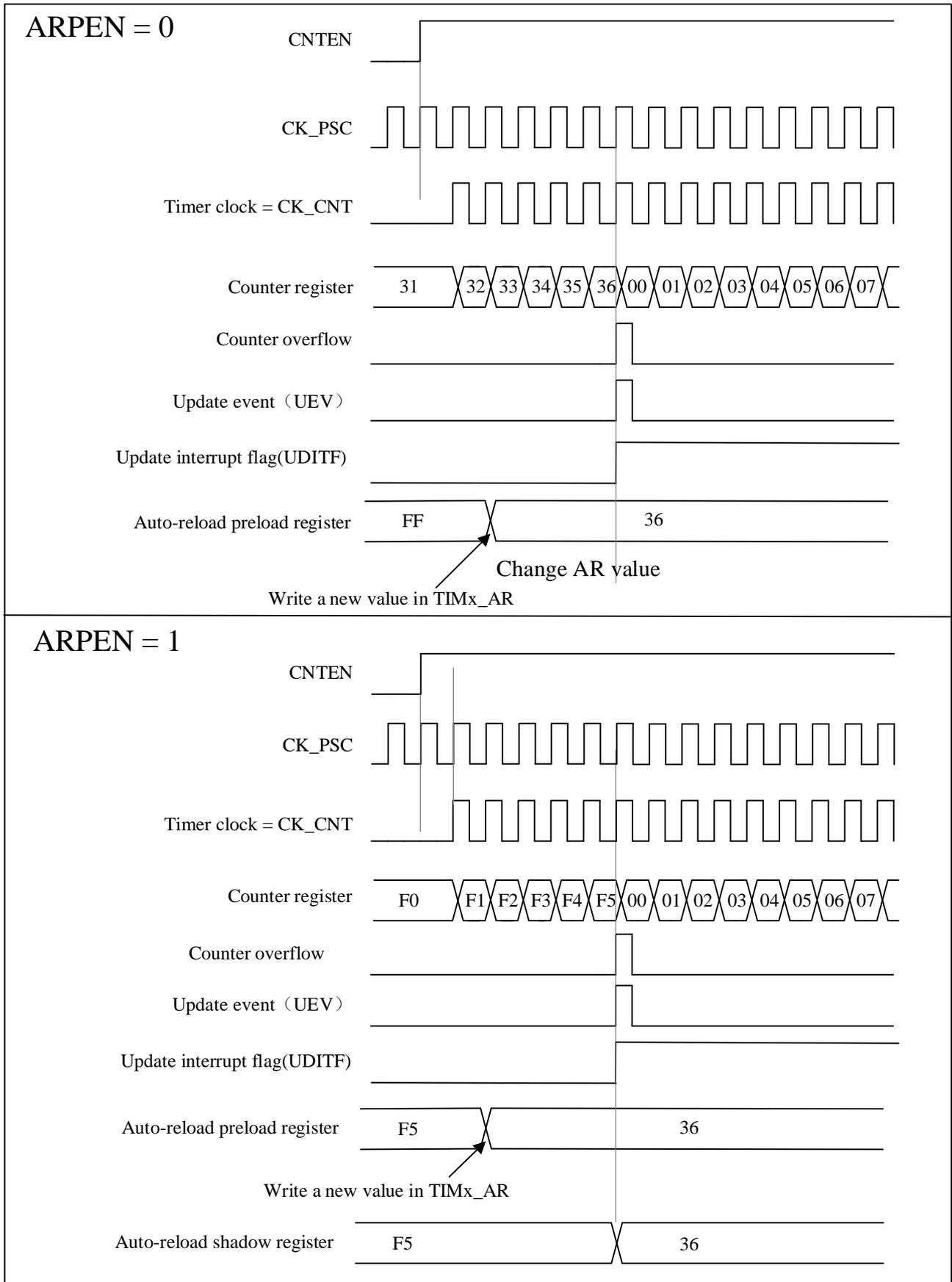


图 14-4 当 ARPEN=0/1 产生更新事件时，向上计数的时序图



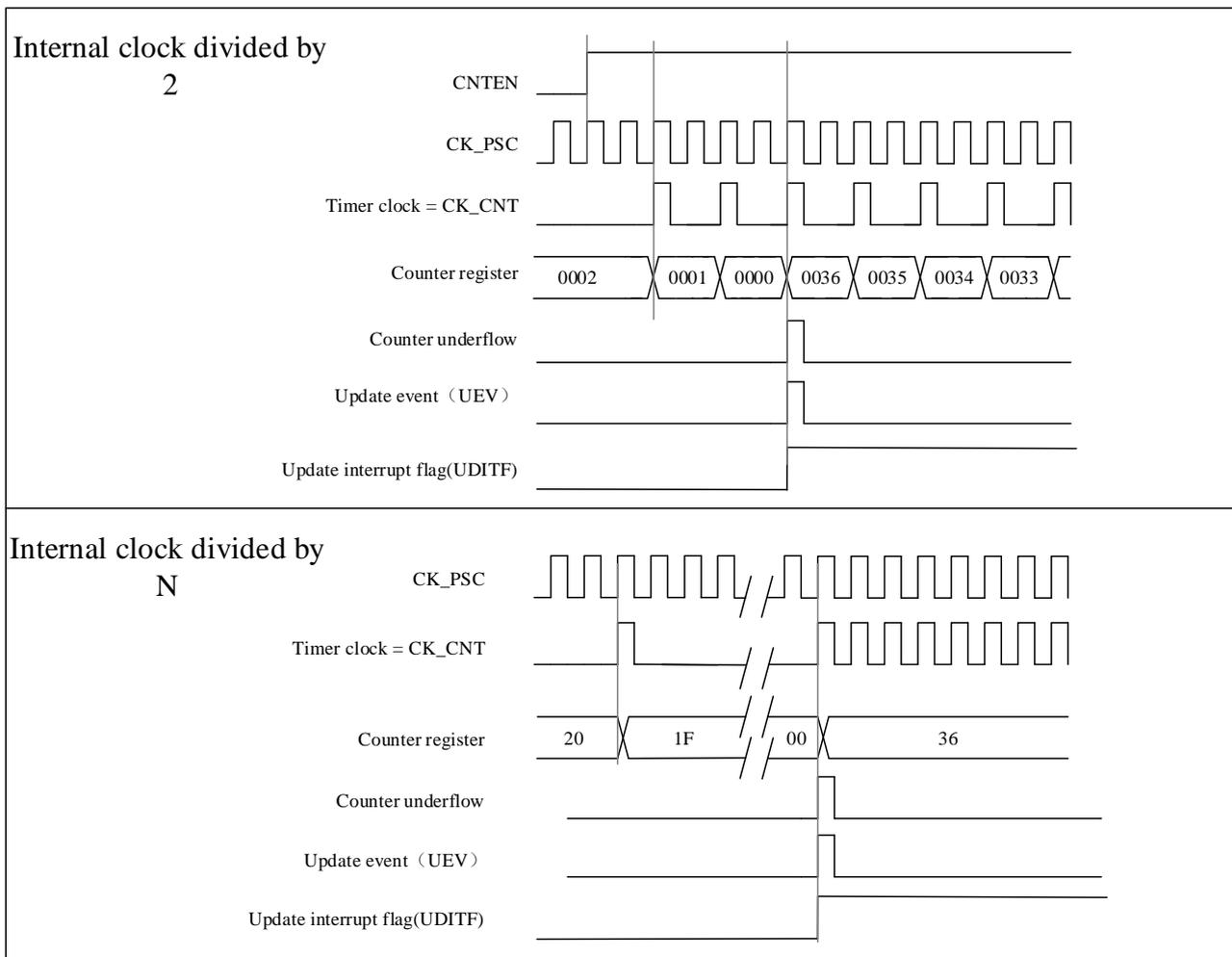
### 14.3.2.2 向下计数模式

向下计数模式，计数器将从寄存器 TIMx\_AR 的值减至 0，然后从自动重载值重新开始，并产生计数器向下溢出事件

向下计数模式和向上计数模式配置更新事件和更新寄存器的过程相同，请查阅 13.3.2.1 章节。

下图给出一些示例，展示了向下计数模式计数器在不同分频因子下的动作。

图 14-5 内部时钟分频因子 = 2/N 时，向下计数时序图



### 14.3.2.3 中央对齐模式

在中央对齐模式下，计数器从 0 增加到值 (TIMx\_AR) - 1，产生计数器溢出事件。然后，它从自动重载值 (TIMx\_AR) 向下计数到 1，并生成一个计数器向下溢出事件。然后计数器重置为 0 并再次开始计数。

在这种模式下，TIMx\_CTRL1.DIR 方向位无效，由硬件更新和指定当前计数方向。当 TIMx\_CTRL1.CAMSEL 位不等于“00”时，中央对齐模式有效。

每次计数上溢和计数下溢时都会生成更新事件。或者，也可以通过设置 TIMx\_EVTGEN.UDGN 位（通过软件或使用从模式控制器）来生成更新事件。在这种情况下，计数器从 0 重新开始计数，预分频器的计数器也从 0 重新开始计数。

*注：如果因为计数器溢出而产生更新，自动重载将在计数器重新载入之前被更新。*

图 14-6 内部时钟分频因子 = 2/N，中央对齐时序图

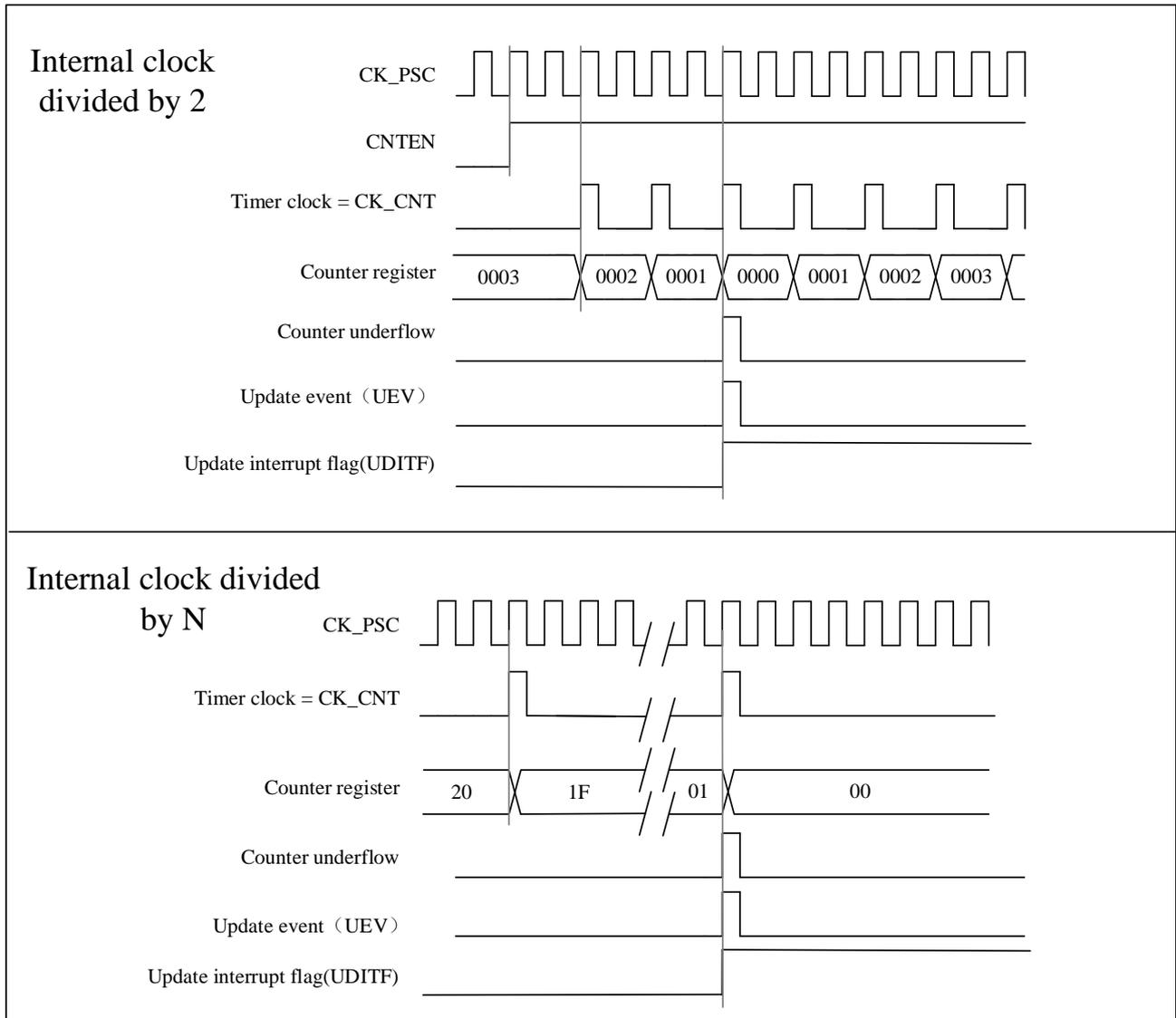
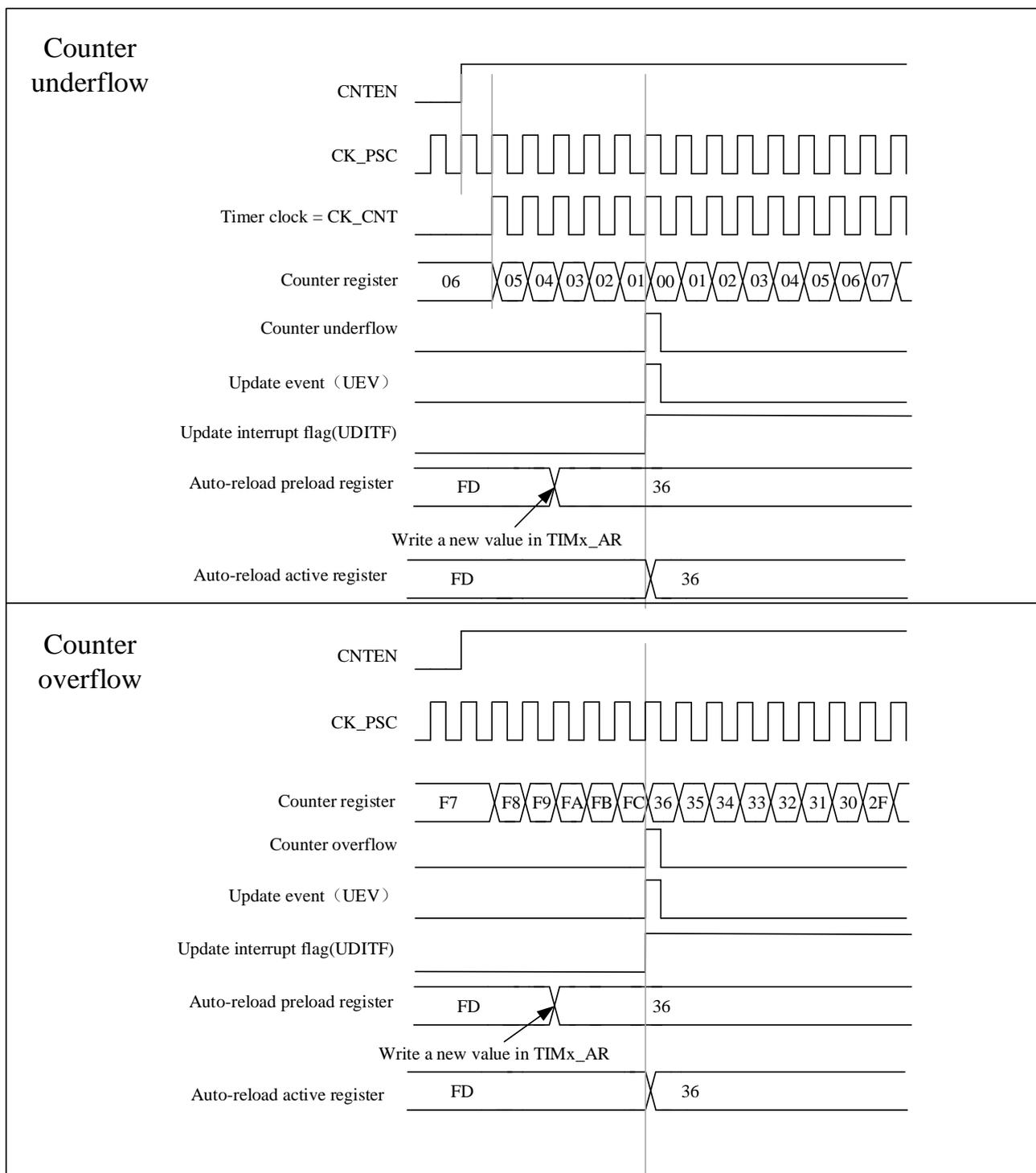


图 14-7 包含计数器上溢和下溢的中央对齐时序图(ARPEN=1)



### 14.3.3 时钟选择

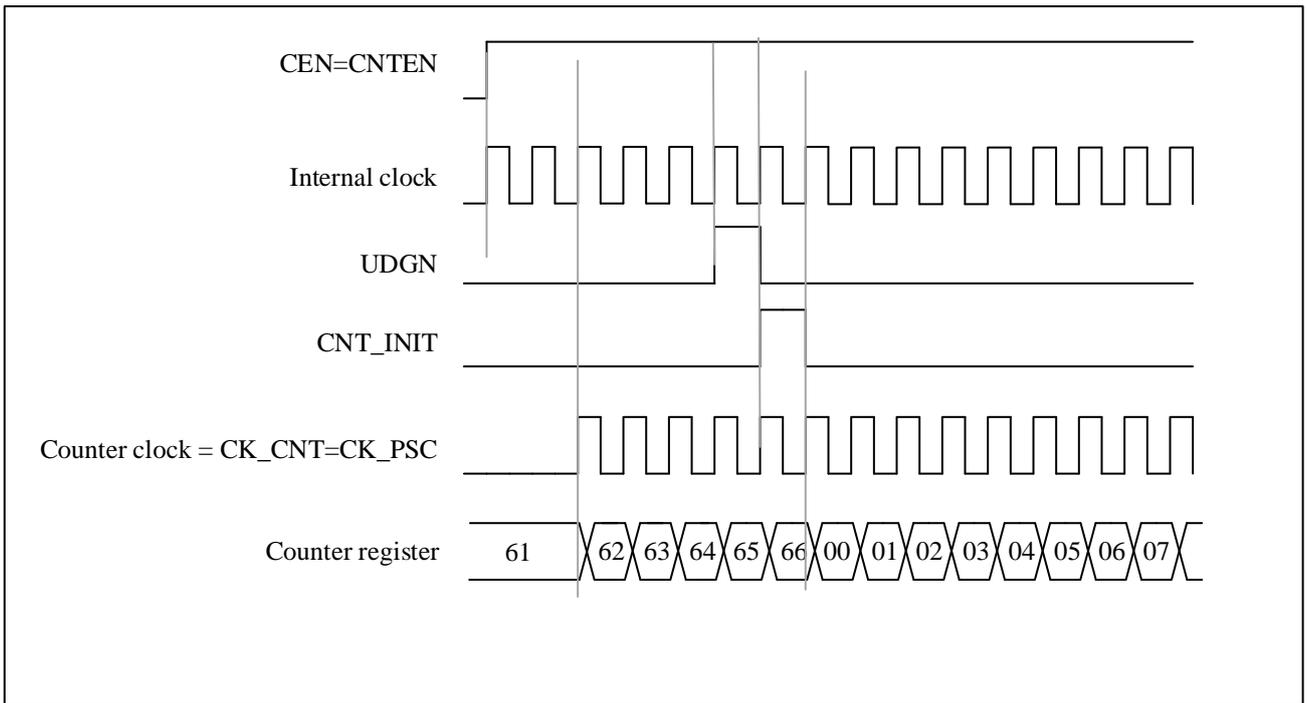
- 通用定时器的内部时钟：CK\_INT；
- 两种外部时钟模式：
  - 外部输入引脚

- 外部触发输入 ETR
- 内部触发输入 (ITRx): 一个定时器用作另一个定时器的预分频器

### 14.3.3.1 内部时钟源(CK\_INT)

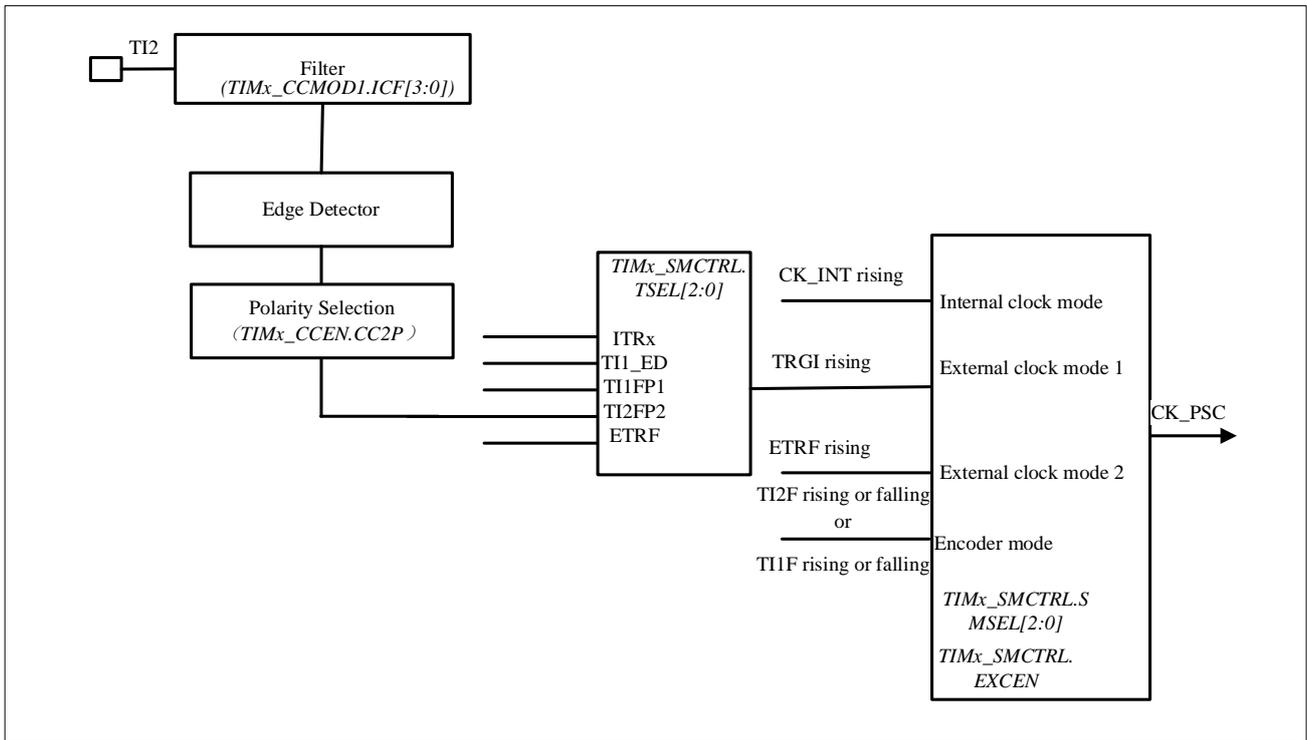
当 TIMx\_SMCTRL.SMSEL 等于“0000”时，从模式控制器被禁用。这三个控制位 (TIMx\_CTRL1.CNTEN、TIMx\_CTRL1.DIR、TIMx\_EVTGEN.UDGN) 只能由软件改变 (TIMx\_EVTGEN.UDGN 除外，它保持自动清零)。前提是 TIMx\_CTRL1.CNTEN 位被软写为'1'，预分频器的时钟源由内部时钟 CK\_INT 提供。

图 14-8 正常模式下的控制电路，内部时钟除以 1



### 14.3.3.2 外部时钟源模式 1

图 14-9 TI2 外部时钟连接示例



通过配置 `TIMx_SMCTRL.SMSEL=0111` 选择该模式。计数器可以配置为在所选输入的时钟上升沿或下降沿进行计数。

例如，配置向上计数模式在 `TI2` 输入的时钟上升沿计数，配置步骤如下：

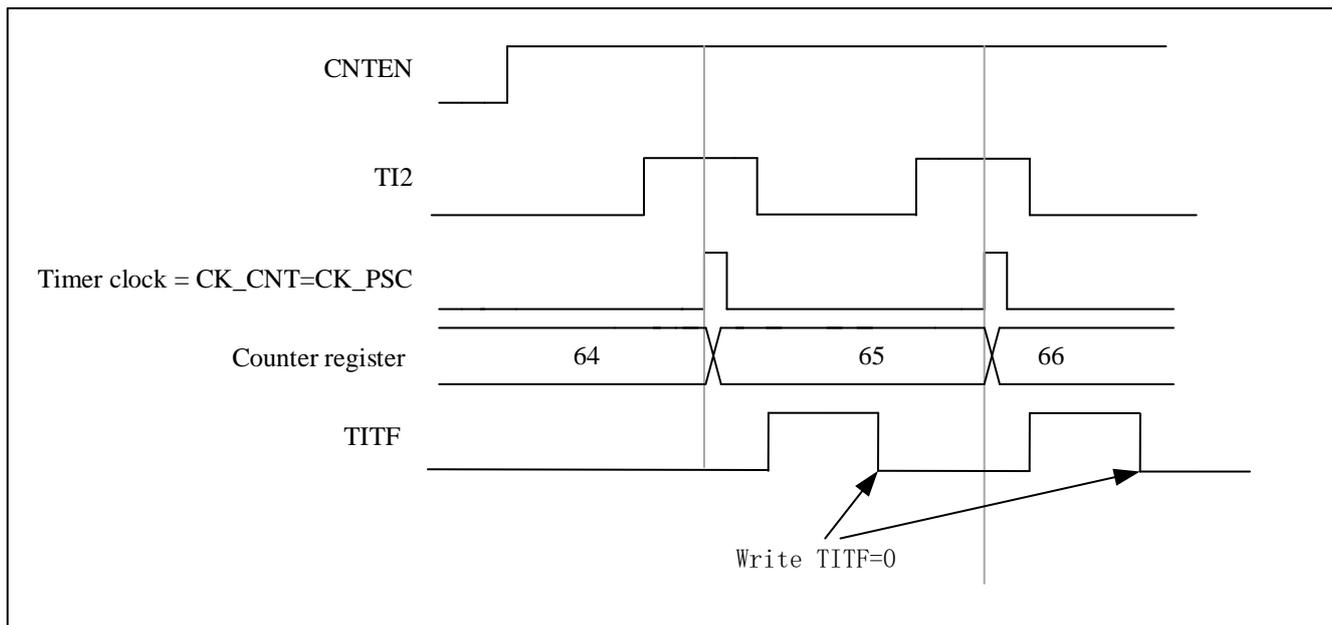
- 配置 `TIMx_CCMOD1.CC2SEL` 等于‘01’，`CC2` 通道配置为输入，`IC2` 映射到 `TI2`
- 配置 `TIMx_CCEN.CC2P` 等于‘0’，选择时钟上升沿极性
- 通过配置 `TIMx_CCMOD1.IC2F[3:0]` 选择输入滤波器带宽（如果不需要滤波器，保持 `IC2F` 位为‘0000’）
- 配置 `TIMx_SMCTRL.SMSEL` 等于‘0111’，选择定时器外部时钟模式 1
- 配置 `TIMx_SMCTRL.TSEL` 等于‘110’，选择 `TI2` 作为触发输入源
- 配置 `TIMx_CTRL1.CNTEN` 等于‘1’以启动计数器

注意：捕获预分频器不用于触发，所以不需要配置

当定时器时钟的上升沿出现在 `TI2=1` 时，计数器计数一次并且 `TIMx_STS.TITF` 标志被拉高。

`TI2` 的上升沿与计数器实际时钟之间的延迟取决于 `TI2` 输入端的再同步电路。

图 14-10 外部时钟模式 1 的控制电路

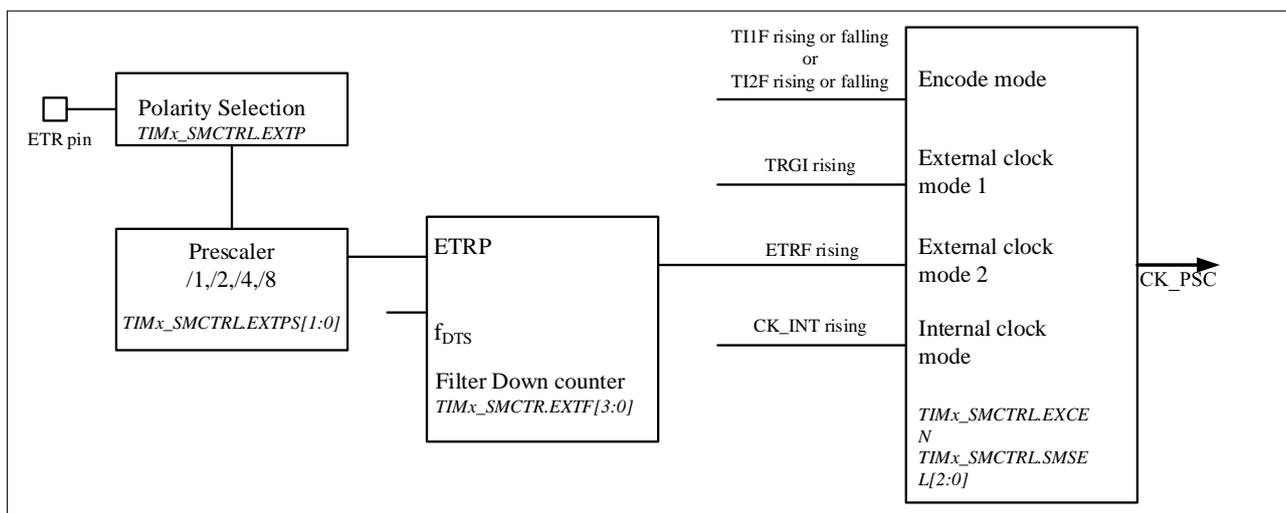


### 14.3.3.3 外部时钟源模式 2

此模式由 `TIMx_SMCTRL.EXCEN` 选择等于 1。计数器可以在外部触发输入 ETR 的每个上升沿或下降沿计数。

下图为外部时钟源模式 2 的外部触发输入模块示意图。

图 14-11 外部触发输入框图



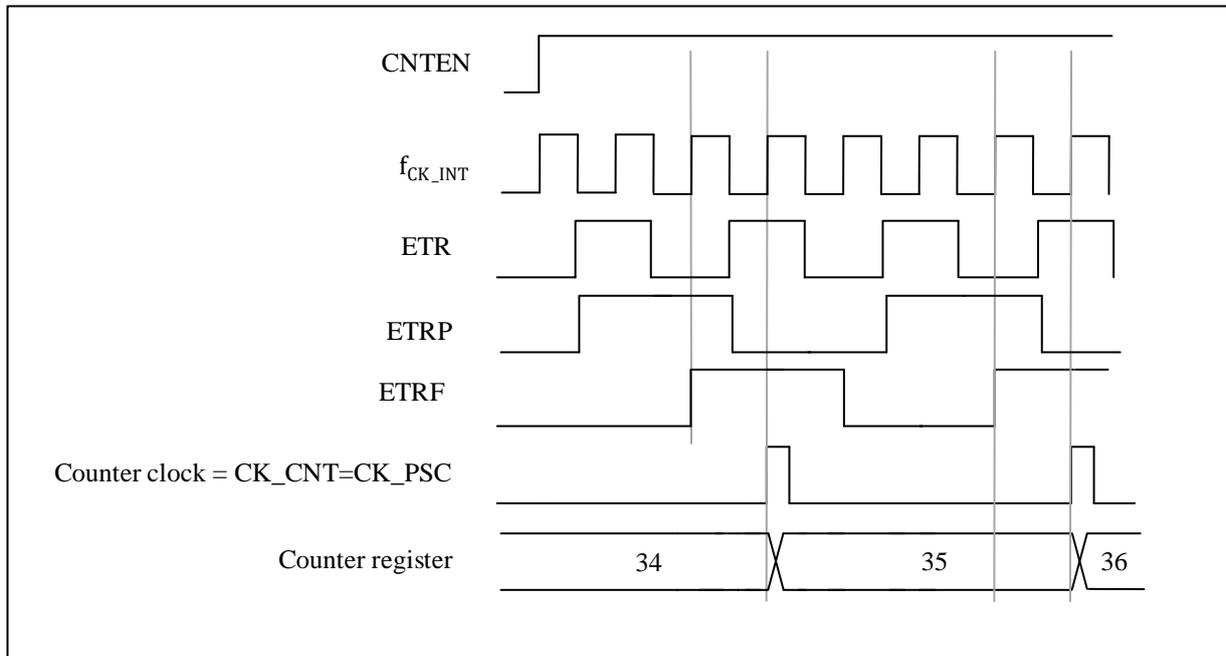
例如，使用以下配置步骤使向上计数器在 ETR 上每 2 个上升沿计数一次。

- 由于在这种情况下不需要过滤器，因此使 `TIMx_SMCTRL.EXTF[3:0]` 等于 '0000'
- 通过使 `TIMx_SMCTRL.EXTPS[1:0]` 等于 '01' 来配置预分频器
- 通过设置 `TIMx_SMCTRL.EXTP` 等于 '0' 来选择 ETR 引脚的极性，ETR 的上升沿有效

- 外部时钟模式 2 通过设置 TIMx\_SMCTRL.EXCEN 等于‘1’来选择
- 通过设置 TIMx\_CTRL1.CNTEN 等于“1”启动计数器。

计数器每 2 个 ETR 上升沿计数一次。ETR 的上升沿与计数器的实际时钟之间的延迟是由于 ETRP 信号上的再同步电路造成的。

图 14-12 外部时钟模式 2 的控制电路

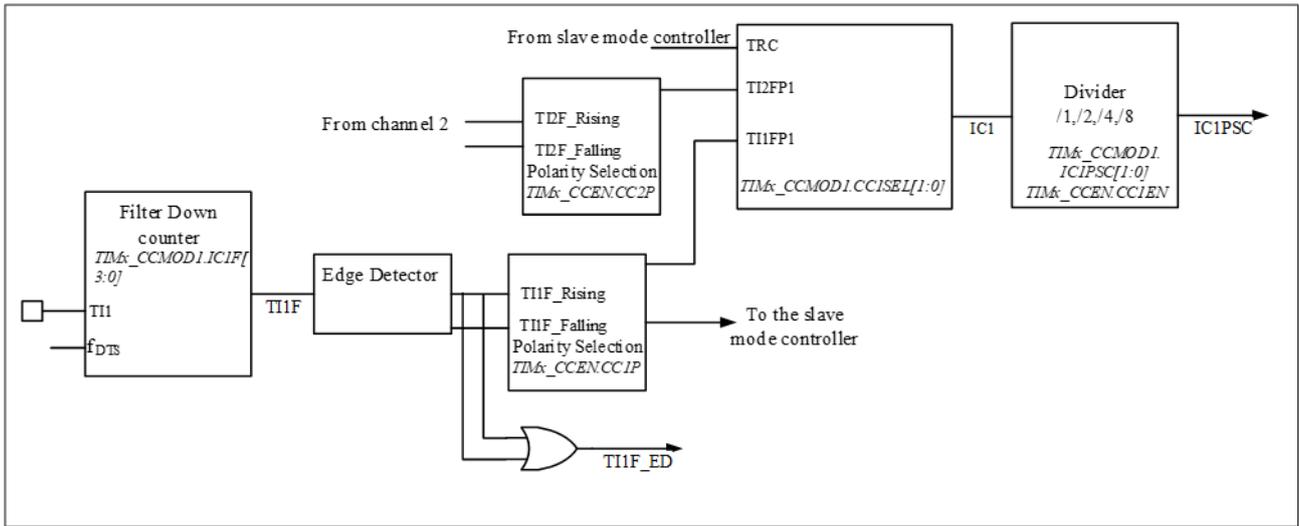


### 14.3.4 捕获/比较通道

捕获/比较通道包括捕获/比较寄存器和影子寄存器。输入部分由数字滤波器、多路复用器和预分频器组成。输出部分包括比较器和输出控制。

输入信号 TIx 被采样和滤波以产生信号 TIxF。然后由极性选择功能的边沿检测器生成信号 (TIxF\_rising 或 TIxF\_falling)，其极性由 TIMx\_CCEN.CCxP 位选择。该信号可用作从模式控制器的触发输入。同时，信号 ICx 经过分频后送入捕获寄存器。下图显示了捕获/比较通道的框图。

图 14-13 捕获/比较通道（例如：通道 1 输入级）



输出部分生成一个中间波形 OCxRef（高电平有效）作为参考。极性作用在链的末端。

图 14-14 捕获/比较通道 1 主电路

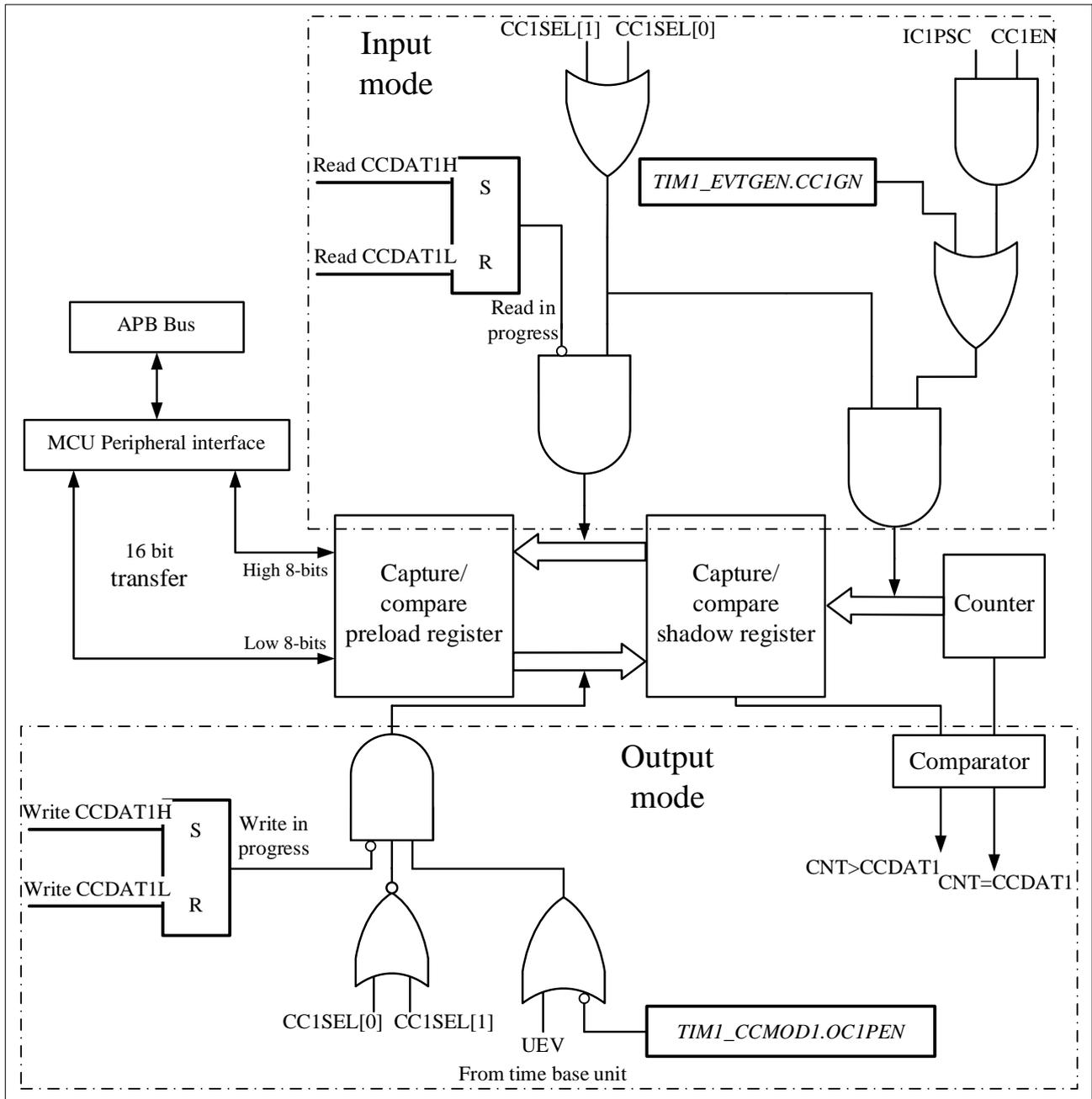
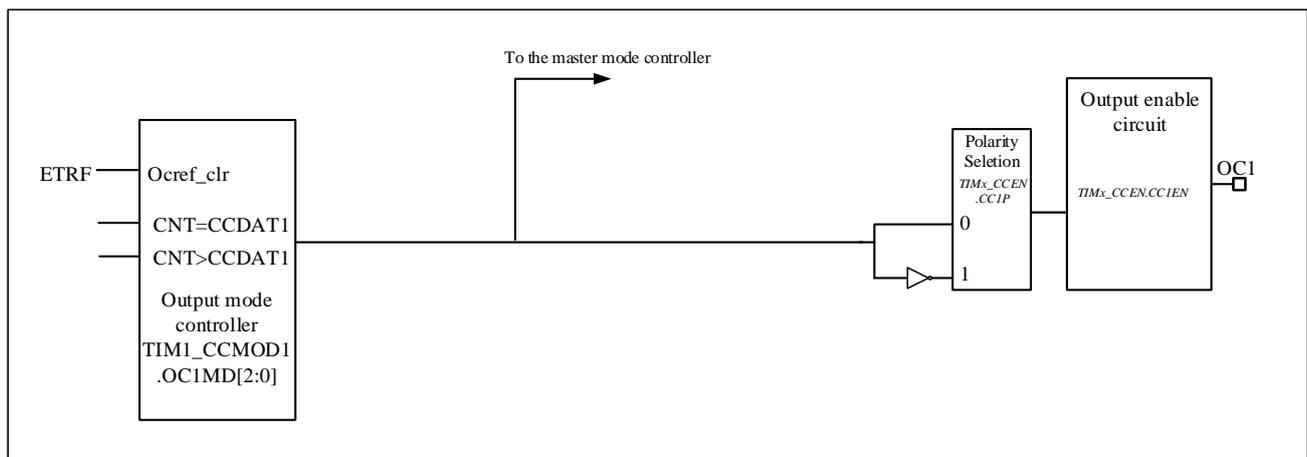


图 14-15 通道 x 的输出部分（以通道 1 为例子）



在捕获/比较时，读取和写入始终访问预加载的寄存器。两个具体工作流程如下：

在捕获模式下，捕获实际上是在影子寄存器中完成的，然后将影子寄存器中的值复制到预加载寄存器中。

在比较模式下，与捕获模式相反，预加载寄存器的值被复制到影子寄存器中，并与计数器进行比较。

### 14.3.5 输入捕获模式

在捕获模式下，TIMx\_CCDATx 寄存器用于在检测到 ICx 信号后锁存计数器值。

有一个捕获中断标志 TIMx\_STS.CCxITF，如果相应的中断使能被拉高，它可以发出中断或 DMA 请求。

TIMx\_STS.CCxITF 位在发生捕获事件时由硬件设置，并由软件或读取 TIMx\_CCDATx 寄存器清零。

当 TIMx\_CCDATx 寄存器中的计数器值被捕获并且 TIMx\_STS.CC1ITF 已经被拉高时，重复捕获标志 TIMx\_STS.CCxOCF 设置为 1。与前者不同，TIMx\_STS.CCxOCF 通过向其写入 0 来清除。

为实现 TI1 输入的上升沿将计数器值捕获到 TIMx\_CCDAT1 寄存器中，配置流程如下：

■ 选择有效输入：

将 TIMx\_CCMOD1.CC1SEL 配置为“01”。此时输入为 CC1 通道，IC1 映射到 TI1。

■ 编程所需的输入滤波器持续时间：

通过配置 TIMx\_CCMODx.ICx F 位来定义 TI1 输入的采样频率和数字滤波器的长度。示例：如果输入信号抖动多达 5 个内部时钟周期，我们必须选择比这 5 个时钟周期更长的滤波器持续时间。当检测到具有新电平的 8 个连续样本（以  $f_{TIM4FILTCLK}$  频率采样）时，我们可以验证 TI1 上的转换。然后配置 TIMx\_CCMOD1.IC1F 到“1xxx”

■ 通过配置 TIMx\_CCEN.CC1P=0，选择上升沿作为 TI1 通道的有效跳变极性

■ 配置输入预分频器。在本例中，配置 TIMx\_CCMOD1.IC1PSC= ‘00’ 以禁用预分频器，因为我们想要捕获每个有效转换

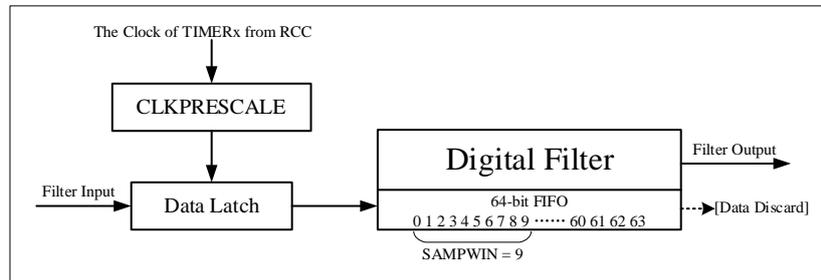
■ 通过配置 TIMx\_CCEN.CC1EN = ‘1’ 启用捕获。

如果要使能 DMA 请求，可以配置 TIMx\_DINTEN.CC1DEN=1。如果要使能相关中断请求，可以配置 TIMx\_DINTEN.CC1IEN =1。

### 14.3.5.1 通道输入滤波

寄存器 TIMx\_CxFILT(x = 1, 2, 3, 4) 描述如下:

图 14-16 滑动滤波



- 数字滤波器通过 RCC 的 TIMx 时钟采样通道输入信号，在 64 位 FIFO 中累积采样。仅在 TIMx\_CxFILT.WSIZE [5:0] 中定义的窗口大小内采样数据，最大大小为 64。
- 过滤器输出采样窗口内的多数值，该值由 TIMx\_CxFILT.THRESH [5:0] 中的阈值定义，最大阈值为 63。此值应等于或大于窗口大小的一半。如果采样窗口内的逻辑 1 和逻辑 0 计数均不大于阈值，则数字滤波器保持先前的输出值。
- RCC\_TIMFILTCFG.TIM4FILTCLK[4:0] 寄存器决定相应数字滤波器的采样率。过滤器 FIFO 在每个采样时钟从输入中捕获一个采样值。
- 如果数字滤波器关闭，滤波器输入将像电线一样绕过输出。

### 14.3.6 PWM 输入模式

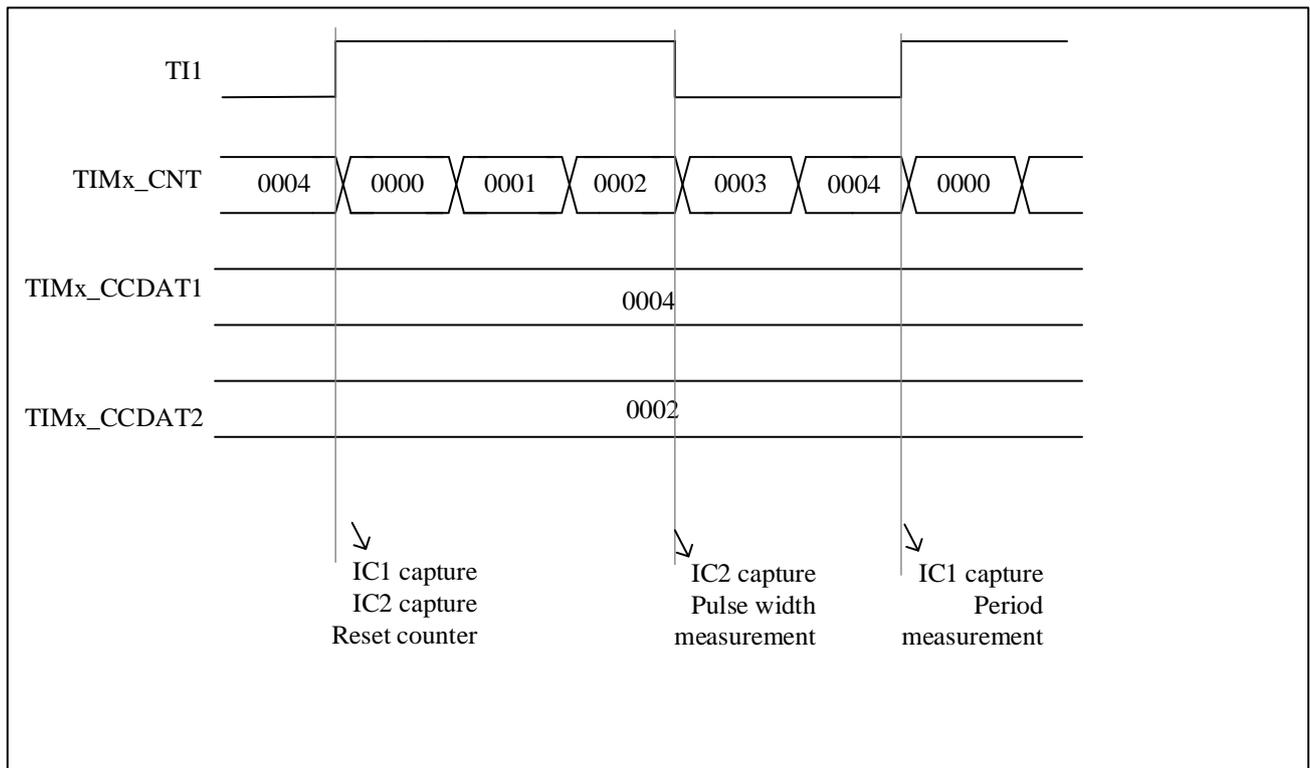
PWM 输入模式和普通输入捕获模式有一些区别，包括:

- 两个 ICx 信号映射到同一个 TIx 输入
- 两个 ICx 信号在极性相反的边沿有效
- 选择两个 TIxFP 信号之一作为触发输入
- 从机模式控制器配置为复位模式

例如，下面的配置流程可以用来知道 TI1 上 PWM 信号的周期和占空比（这取决于 CK\_INT 的频率和预分频器的值）。

- 配置 TIMx\_CCMOD1.CC1SEL 等于 '01' 以选择 TI1 作为 TIMx\_CCDAT1 的有效输入
- 配置 TIMx\_CCEN.CC1P 等于 '0' 选择滤波定时器输入 1(TI1FP1) 的有效极性，在上升沿有效
- 配置 TIMx\_CCMOD1.CC2SEL 等于 '10' 选择 TI1 作为 TIMx\_CCDAT2 的有效输入
- 配置 TIMx\_CCEN.CC2P 等于 1 选择滤波定时器输入 2(TI1FP2)的有效极性，下降沿有效
- 配置 TIMx\_SMCTRL.TSEL=101 选择 Filtered timer input 1 (TI1FP1) 作为有效触发输入
- 配置 TIMx\_SMCTRL.SMSEL=0100 配置从模式控制器为复位模式
- 配置 TIMx\_CCEN.CC1EN=1 和 TIMx\_CCEN.CC2EN=1 以启用捕获

图 14-17 PWM 输入模式时序



由于只有滤波器定时器输入 1 (TI1FP1) 和滤波器定时器输入 2 (TI2FP2) 连接到从模式控制器, 因此 PWM 输入模式只能与 TIMx\_CH1/TIMx\_CH2 信号一起使用。

### 14.3.7 强制输出模式

在输出模式 (TIMx\_CCMODx.CCxSEL=00) 下, 软件可以直接将输出比较信号强制为有效或无效电平。

用户可以设置 TIMx\_CCMODx.OCxMD=101 强制输出比较信号为有效电平。OCxREF 将被强制为高电平, OCx 得到与 CcxP 极性位相反的值。另一方面, 用户可以设置 TIMx\_CCMODx.OCxMD=100 强制输出比较信号为无效电平, 即 OCxREF 被强制为低电平。

在此模式下, TIMx\_CCDATx 影子寄存器和计数器的值仍然相互比较。

输出比较寄存器 TIMx\_CCDATx 和计数器 TIMx\_CNT 之间的比较对 OCxREF 没有影响。并且仍然可以设置标志。因此, 仍然可以发送中断和 DMA 请求。

### 14.3.8 输出比较模式

用户可以使用此模式来控制输出波形, 或指示一段时间已过。

当捕获/比较寄存器和计数器的值相同时, 输出比较函数的操作如下:

- TIMx\_CCMODx.OCxMD 为输出比较模式, TIMx\_CCEN.CCxP 为输出极性。当比较匹配时, 如果设置 TIMx\_CCMODx.OCxMD=000, 则输出管脚将保持其电平; 如果设置 TIMx\_CCMODx.OCxMD=001, 则设置输出管脚有效; 如果设置 TIMx\_CCMODx.OCxMD=010, 则输出管脚将为 设置为无效; 如果设置 TIMx\_CCMODx.OCxMD=011, 则输出引脚将设置为翻转。
- 设置 TIMx\_STS.CCxITF

- 如果用户设置了 TIMx\_DINTEN.CCxIEN，将产生相应的中断
- 如果用户设置 TIMx\_DINTEN.CCxDEN 并设置 TIMx\_CTRL2.CCDSEL 选择 DMA 请求，将发送 DMA 请求

用户可以设置 TIMx\_CCMODx.OCxPEN 来选择是否使用捕获/比较预加载寄存器 (TIMx\_CCDATx) 来选择捕获/比较影子寄存器。

时间分辨率是计数器的一个计数周期。

在单脉冲模式下，输出比较模式也可用于输出单脉冲。

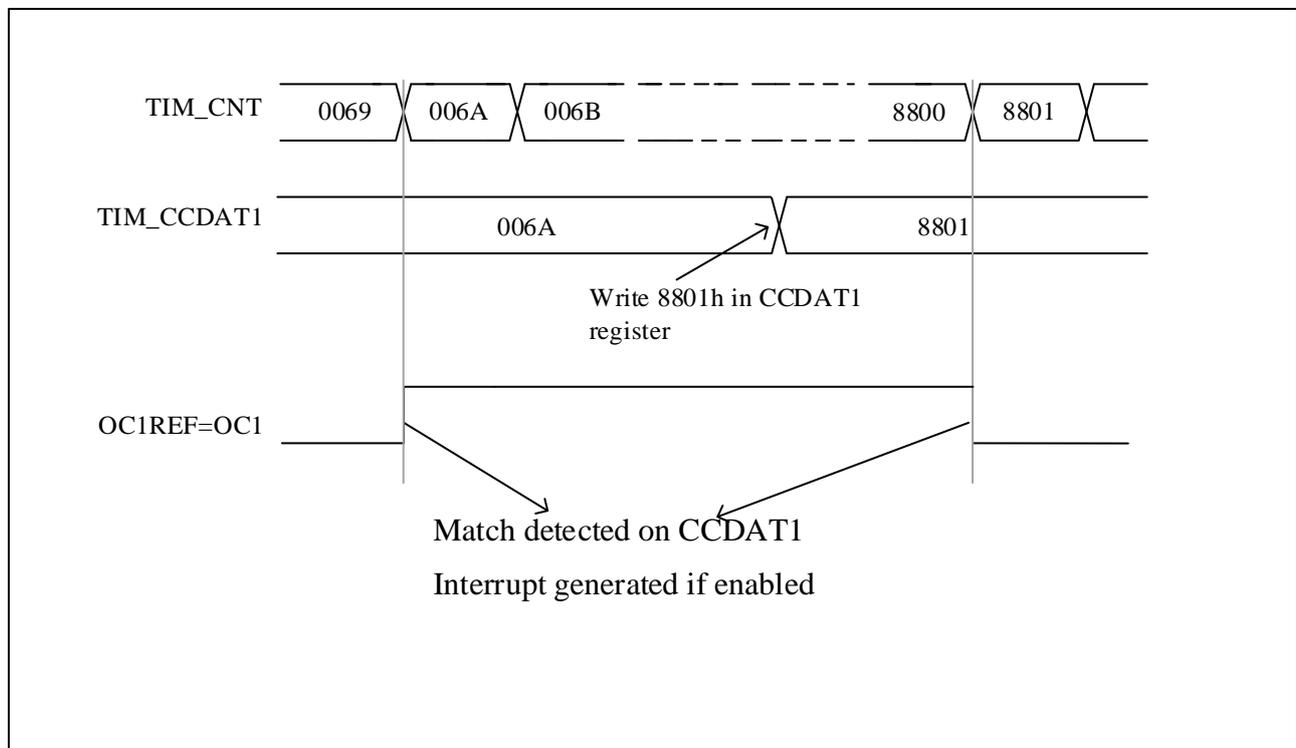
以下是输出比较模式的配置步骤：

- 首先，用户应该选择计数器时钟
- 其次，用所需数据设置 TIMx\_AR 和 TIMx\_CCDATx
- 如果用户需要产生中断，设置 TIMx\_DINTEN.CCxIEN
- 然后通过设置 TIMx\_CCEN.CCxP、TIMx\_CCMODx.OCxMD、TIMx\_CCEN.CCxEN 等选择输出模式
- 最后，设置 TIMx\_CTRL1.CNTEN 启用计数器

用户可以随时通过设置 TIMx\_CCDATx 来更新输出波形，只要不启用预加载寄存器。否则，TIMx\_CCDATx 影子寄存器将在下一次更新事件中更新。

例如：

图 14-18 输出比较模式，开启 OC1



### 14.3.9 PWM 模式

用户可以使用 PWM 模式产生一个信号，其占空比由 TIMx\_CCxDATx 寄存器的值决定，其频率由 TIMx\_AR 寄存器的值决定。并且取决于 TIMx\_CTRL1.CAMSEL 的值，TIM 可以在边沿对齐模式或中央对齐模式下产生 PWM 信号。

用户可以通过设置 TIMx\_CCMODx.OCxMD=110 或设置 TIMx\_CCMODx.OCxMD=111 来设置 PWM 模式 1 或 PWM 模式 2。要使能预加载寄存器，用户必须设置相应的 TIMx\_CCMODx.OCxPEN。然后设置 TIMx\_CTRL1.ARPEN 自动重装载预加载寄存器。

用户可以通过设置 TIMx\_CCEN.CCxP 来设置 OCx 的极性。

当 TIM 处于 PWM 模式时，TIMx\_CNT 和 TIMx\_CCxDATx 的值总是相互比较。

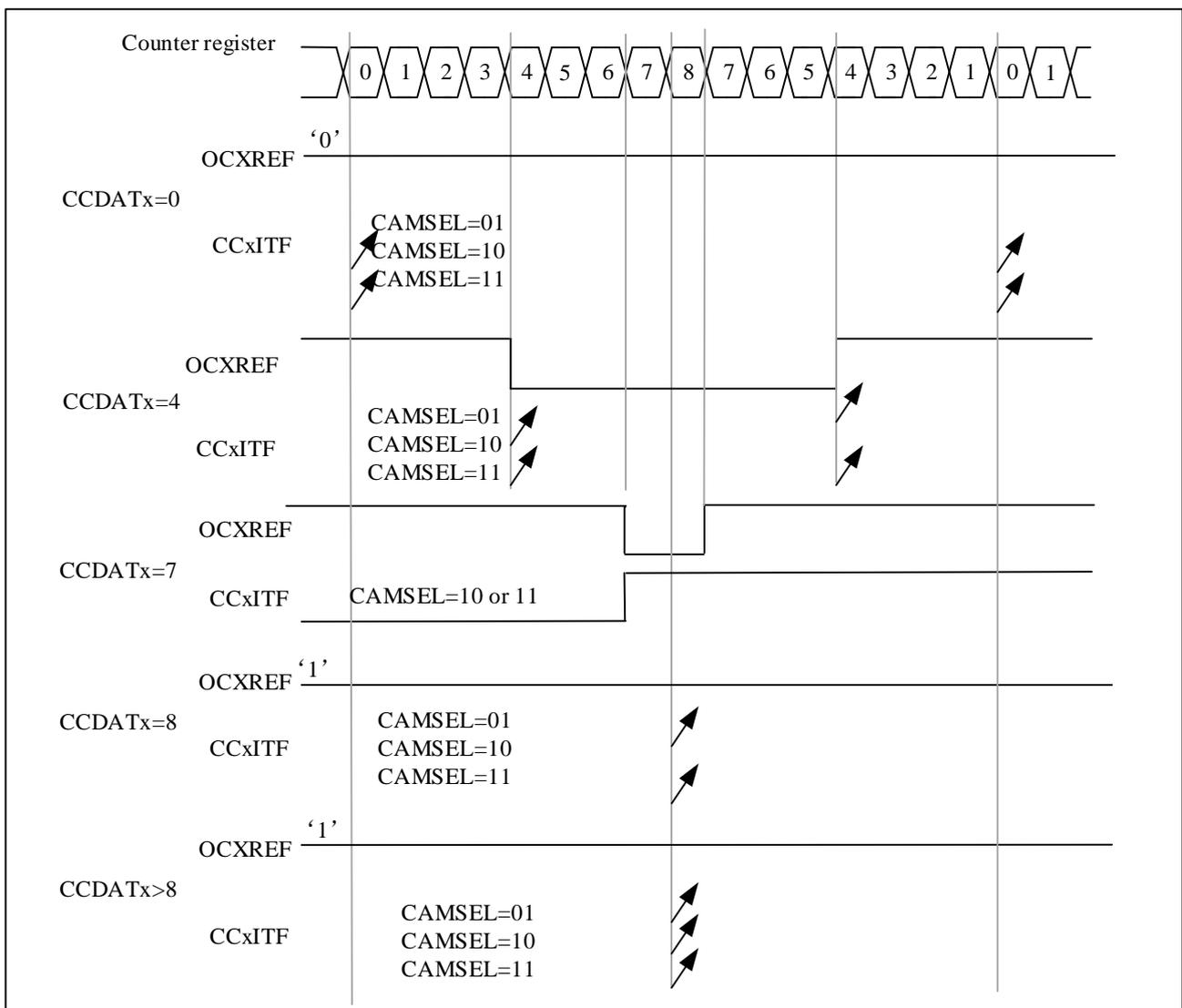
只有当更新事件发生时，预加载寄存器才会转移到影子寄存器。因此，用户必须在计数器开始计数之前通过设置 TIMx\_EVTGEN.UDGN 来复位所有寄存器。

#### 14.3.9.1 PWM 中央对齐模式

如果用户设置 TIMx\_CTRL1.CAMSEL 等于 01、10 或 11，PWM 中央对齐模式将被激活。比较标志的设置取决于 TIMx\_CTRL1.CAMSEL 的值。设置比较标志的情况有 3 种，仅当计数器向上计数时，仅当计数器向下计数时，或当计数器向上计数和向下计数时。用户不应通过软件修改 TIMx\_CTRL1.DIR，它是由硬件更新的。

中央对齐 PWM 波形示例如下，波形设置为：TIMx\_AR=8，PWM 模式 1，当计数器向下计数对应 TIMx\_CTRL1.CAMSEL=01 时设置比较标志。

图 14-19 中央对齐的 PWM 波形 (AR=8)



使用中央对齐模式时用户应注意的事项如下：

- 计数器向上或向下计数取决于 TIMx\_CTRL1.DIR 的值。注意不要同时更改 DIR 和 CAMSEL 位
- 用户在中央对齐模式下不要写计数器，否则会导致意想不到的结果。例如：
  - ◆ 如果写入计数器的值为 0 或者是 TIMx\_AR 的值，则方向会被更新，但不会产生更新事件
  - ◆ 如果写入计数器的值大于自动重载的值，则方向不会更新
- 为了安全起见，建议用户在启动计数器之前设置 TIMx\_EVTGEN.UDGN 以通过软件生成更新，并且在计数器运行时不要写入计数器

### 14.3.9.2 PWM 边沿对齐模式

边沿对齐模式有两种配置，向上计数和向下计数。

#### ● 向上计数

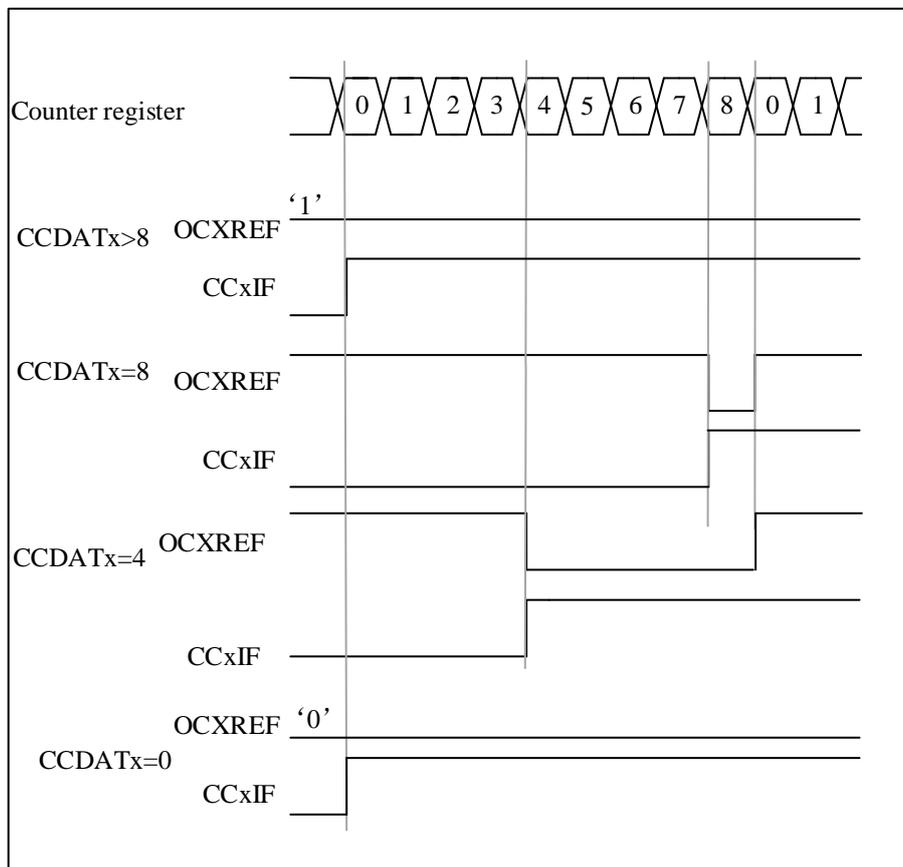
用户可以设置 TIMx\_CTRL1.DIR=0 使计数器向上计数。

PWM 模式 1 的示例：

当  $TIMx\_CNT < TIMx\_CCDATx$  时， $OCxREF$  为高电平，否则为低电平。如果  $TIMx\_CCDATx$  中的比较值大于自动重载值，则  $OCxREF$  将保持为 1。相反，如果比较值为 0，则  $OCxREF$  将保持为 0。

当  $TIMx\_AR=8$  时，PWM 波形如下：

图 14-20 边沿对齐 PWM 波形 (AR=8)



### ● 向下计数

用户可以设置  $TIMx\_CTRL1.DIR=1$  使计数器向下计数。

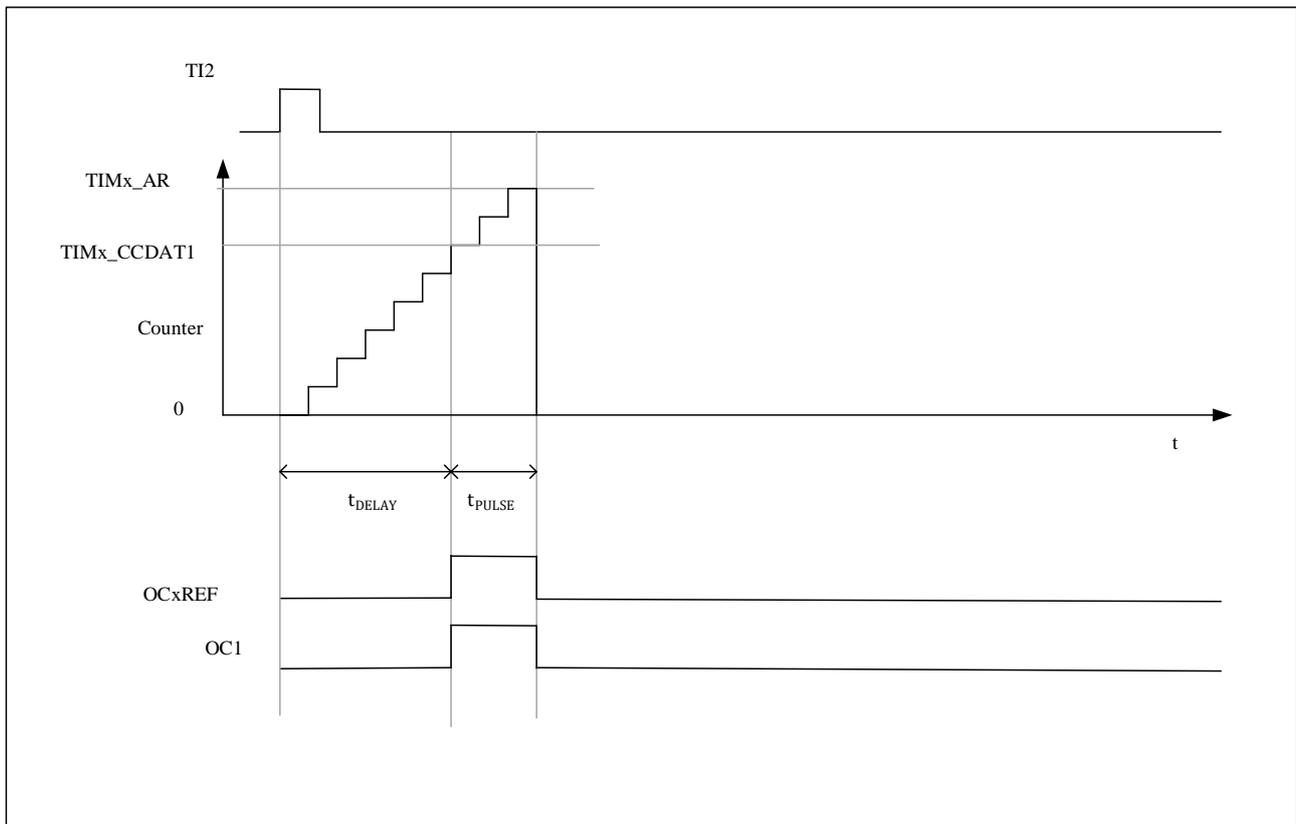
PWM 模式 1 的示例：

当  $TIMx\_CNT > TIMx\_CCDATx$  时， $OCxREF$  为低电平，否则为高电平。如果  $TIMx\_CCDATx$  中的比较值大于自动重载值，则  $OCxREF$  将保持为 1。

注：若第  $n$  个 PWM 周期  $CCDATx$  影子寄存器  $\geq AR$  值，第  $n+1$  个 PWM 周期  $CCDATx$  的影子寄存器值是 0。在第  $n+1$  个 PWM 周期的计数器为 0 的时刻，虽然计数器 =  $CCDATx$  影子寄存器的值 = 0， $OCxREF = '0'$ ，但不会产生比较事件。

### 14.3.10 单脉冲模式

在单脉冲模式(ONEPM)中，接收到触发信号，经过可控延迟  $t_{DELAY}$  后产生脉宽可控的脉冲  $t_{PULSE}$ 。输出模式需要配置为输出比较模式或 PWM 模式。选择单脉冲模式后，计数器会在更新事件 UEV 产生后停止计数。

**图 14-21 单脉冲模式示例**


以下是单脉冲模式的示例：

从 TI2 输入检测到上升沿触发，延迟  $t_{\text{DELAY}}$  后在 OC1 上产生宽度为  $t_{\text{PULSE}}$  的脉冲。

7. 计数器配置：向上计数，计数器  $\text{TIMx\_CNT} < \text{TIMx\_CCDAT1} \leq \text{TIMx\_AR}$ ；
8. TI2FP2 映射到 TI2， $\text{TIMx\_CCMOD1.CC2SEL} = '01'$ ；TI2FP2 配置为上升沿检测， $\text{TIMx\_CCEN.CC2P} = '0'$ ；
9. TI2FP2 充当从模式控制器的触发器（TRGI）并启动计数器， $\text{TIMx\_SMCTRL.TSEL} = '110'$ ， $\text{TIMx\_SMCTRL.SMSEL} = '0110'$ （触发模式）；
10. TIMx\_CCDAT1 写入要延迟的计数值（ $t_{\text{DELAY}}$ ）， $\text{TIMx\_AR} - \text{TIMx\_CCDAT1}$  为脉宽  $t_{\text{PULSE}}$  的计数值；
11. 配置  $\text{TIMx\_CTRL1.ONEPM} = 1$  使能单脉冲模式，配置  $\text{TIMx\_CCMOD1.OC1MD} = '111'$  选择 PWM2 模式；
12. 等待 TI2 有外部触发事件，OC1 输出一个单脉冲波形；

#### 14.3.10.1 特殊情况：OCx 快速使能：

在单脉冲模式下，通过 TIx 输入检测到一个边沿，并触发计数器开始计数到比较值，然后输出一个脉冲。这些操作限制了可以达到的最小延迟  $t_{\text{DELAY}}$ 。

您可以设置  $\text{TIMx\_CCMODx.OCxFEN} = 1$  开启 OCx 快速使能，在触发上升沿后，OCxREF 信号将被强制转换为与比较匹配立即发生的电平相同的电平，而不管比较结果如何。OCxFEN 快速使能仅在通道模式配置为 PWM1 和 PWM2 模式时生效。

### 14.3.11 在外部事件上清除 OCxREF 信号

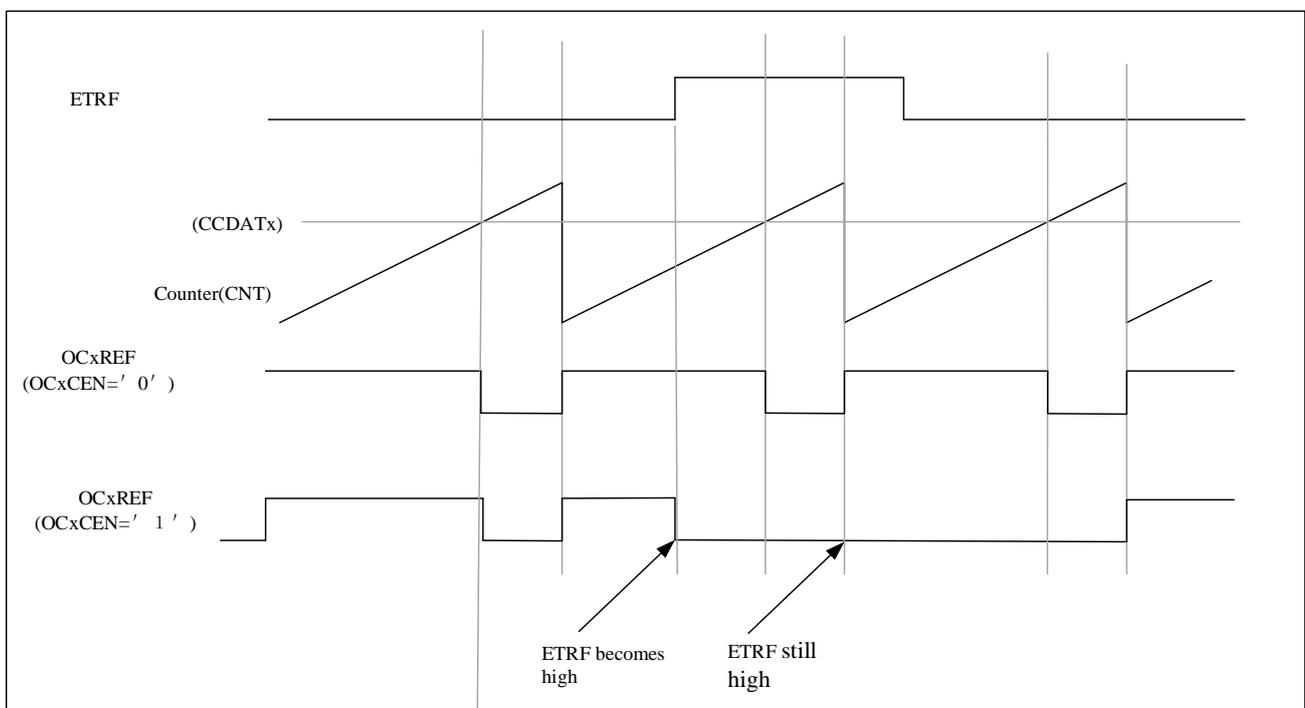
如果用户设置 TIMx\_CCMODx.OCxCEN=1, ETRF 输入的高电平可用于驱动 OCxREF 信号为低电平, OCxREF 信号将保持低电平, 直到下一次 UEV 发生。只有输出比较和 PWM 模式可以使用该功能。在强制模式下不能使用。

例如: 为了控制电流, 用户可以将 ETR 信号连接到比较器的输出端, ETR 的操作如下:

- 设置 TIMx\_SMCTRL.EXTPS=00 禁用外部触发预分频器。
- 设置 TIMx\_SMCTRL.EXCCEN=0 禁用外部时钟模式 2。
- 设置 TIMx\_SMCTRL.EXTP 和 TIMx\_SMCTRL.EXTF, 根据需要配置外触发极性和外触发滤波器。

例: 当 ETRF 输入变高时, OCxREF 信号对于不同的 OCxCEN 值的行为。在这种情况下, 定时器设置为 PWM 模式。

图 14-22 清除 TIMx 的 OCxREF



### 14.3.12 调试模式

当微控制器处于调试模式 (Cortex-M0 内核停止) 时, 根据 DBG\_CTRL.TIMx\_STOP 配置, TIMx 计数器可以继续正常工作或停止。详见 3.3.1 章节。

### 14.3.13 TI3 事件触发功能

#### 14.3.13.1 TI3 事件触发装载

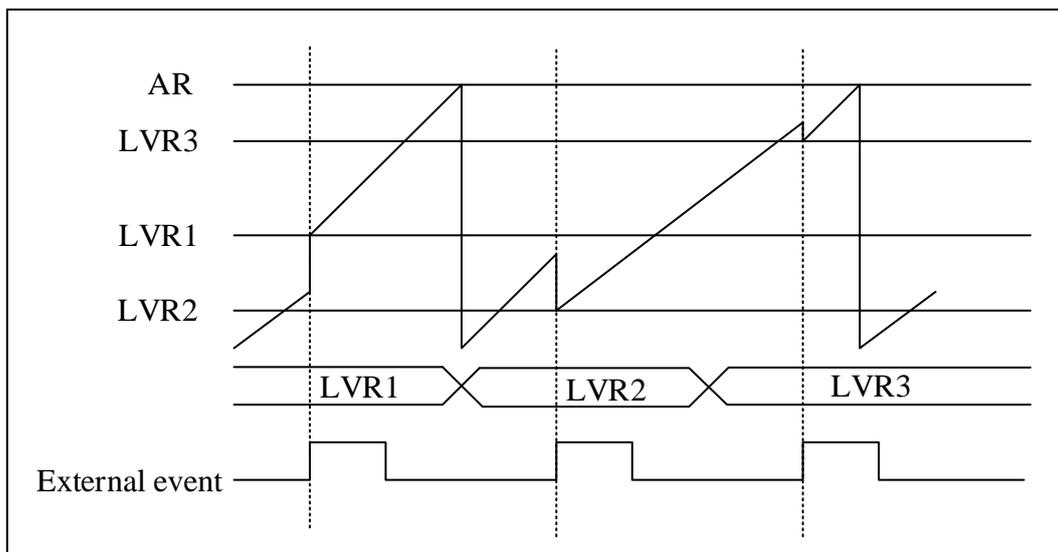
用户可以通过通道 3 的输入信号 TI3 作为触发器, 用外部事件计数器装载值寄存器 (TIMx\_ENCLVR.LVR[31:0]) 的当前值重新加载定时器的值。该功能通过 TIMx\_ENCMCTRL.C3LDCNTEN 位启用。TI3 有效事件触发装

载时，可通过 TIMx\_DINTEN.C3LDCNTIEN 使能开启 TI3 有效事件触发计数器装载为 LVR 值中断。

注意，该 TI3 触发信号可以是边沿有效或者电平有效，用户可配置 TIMx\_ENCMCTRL.C3LDCNTSEL[1:0] 进行选择。

如下图所示，外部信号被用作触发器，在上升沿激活。每次检测到外部信号的上升沿时，定时器的值就会加载为 TIMx\_ENCLVR.LVR[31:0]。图中外部事件的同步延时未画出，实际外部事件会有延时。

图 14-23 外部事件触发装载



### 14.3.13.2 TI3 事件触发捕获差值

当检测到 TI3 事件的有效边沿（当 TIMx\_ENCMCTRL.C3LDCNTSEL[1:0]=2'b0 时，上升沿有效；当 TIMx\_ENCMCTRL.C3LDCNTSEL[1:0]=2'b1 时，下降沿有效）时，TIMx\_ENCDAT.ENC DAT 将记录当前计数器值与 TIMx\_ENCLVR.LVR 值之间差值的绝对值，即

$$TIMx\_ENCDAT.ENC DAT = |TIMx\_CNT.CNT - TIMx\_ENCLVR.LVR|。$$

同时，TIMx\_ENCMCTRL.ENC DATS 将记录当前计数器值与 LVR 值之间差值的符号。例如，如果 ENCDATS 等于 0，则表示 ENCDAT 为正值，即 TIMx\_CNT.CNT - TIMx\_ENCLVR.LVR 大于或等于 0；如果 ENCDATS 等于 1，则表示 ENCDAT 为负值，即 TIMx\_CNT.CNT - TIMx\_ENCLVR.LVR 小于 0。

### 14.3.14 TIMx 定时器和外部触发的同步

与高级定时器相同，见 12.3.16。

### 14.3.15 定时器同步

所有 TIM 定时器在内部相连，用于定时器同步或链接。详见 13.3.16 章节。

### 14.3.16 编码器接口模式

#### 14.3.16.1 正交编码模式

编码器使用两个输入 TI1 和 TI2 作为接口，计数器对 TI1FP1 或 TI2FP2 上的每个边沿变化进行计数。计数方向由硬件 TIMx\_CTRL1.DIR 自动控制。正交编码器计数模式共有五种：

- 编码器模式 1：计数器只在 TI1 的边沿计数，TIMx\_SMCTRL.SMSEL = ‘0001’ 或者 TIMx\_ENCMCTRL.ENCMD = ‘0001’；
- 编码器模式 2：计数器只在 TI2 的边沿计数，TIMx\_SMCTRL.SMSEL = ‘0010’ 或者 TIMx\_ENCMCTRL.ENCMD = ‘0010’；
- 编码器模式 3：计数器同时在 TI1 和 TI2 的边沿计数，TIMx\_SMCTRL.SMSEL = ‘0011’ 或者 TIMx\_ENCMCTRL.ENCMD = ‘0011’；
- 编码器模式 4：T2 是高电平时，计数器只在 TI1 的边沿计数，TIMx\_SMCTRL.SMSEL = ‘1001’ 或者 TIMx\_ENCMCTRL.ENCMD = ‘0100’；
- 编码器模式 5：T1 是高电平时，计数器只在 TI2 的边沿计数，TIMx\_SMCTRL.SMSEL = ‘1010’ 或者 TIMx\_ENCMCTRL.ENCMD = ‘0101’；

编码器接口相当于使用带方向选择的外部时钟，计数器只在 0 和自动重载值(TIMx\_AR.AR [15:0])之间连续计数。因此，需要提前配置自动重载寄存器 TIMx\_AR。

*注意：编码器模式和外部时钟模式 2 不兼容，不能同时选择。*

计数方向与编码器信号的关系如下表：

表 14-1 计数方向与编码器信号的关系 (CC1P=CC2P=0)

有效边沿	SMSEL[3:0] 或者 ENCMD[3:0]	相对信号的电平 (TI1FP1对应 TI2, TI2FP2对应 TI1)	TI1FP1信号		TI2FP2信号	
			上升	下降	上升	下降
仅在TI1计数	SMSEL = 0001或 ENCMD = 0001	高	向下计数	向上计数	不计数	不计数
		低	向上计数	向下计数	不计数	不计数
仅在TI2计数	SMSEL = 0010或 ENCMD = 0010	高	不计数	不计数	向上计数	向下计数
		低	不计数	不计数	向下计数	向上计数
在TI1和TI2上计数	SMSEL = 0011或 ENCMD = 0011	高	向下计数	向上计数	向上计数	向下计数
		低	向上计数	向下计数	向下计数	向上计数
仅在TI1计数且T2为 高电平	SMSEL = 1001或 ENCMD = 0100	高	向下计数	向上计数	不计数	不计数
		低	不计数	不计数	不计数	不计数
仅在TI2计数且T1为 高电平	SMSEL = 1010或 ENCMD = 0101	高	不计数	不计数	向上计数	向下计数
		低	不计数	不计数	不计数	不计数

计数器在各个模式下时计数器值的变化如下：

图 14-24 编码器仅在 TI1 计数

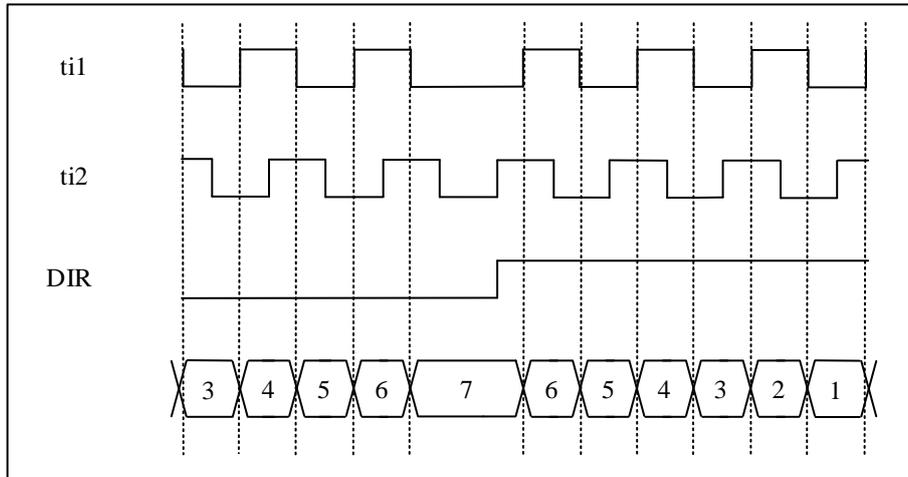


图 14-25 编码器仅在 TI2 计数

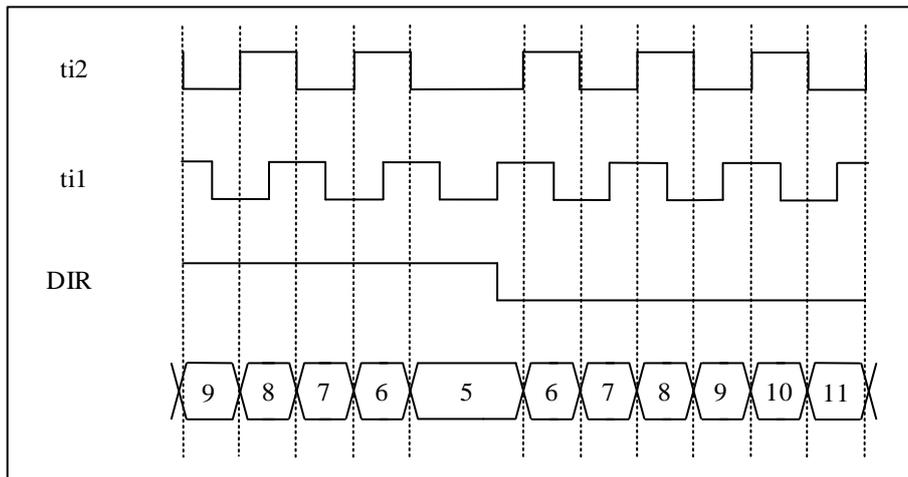


图 14-26 编码器在 TI1 和 TI2 上计数

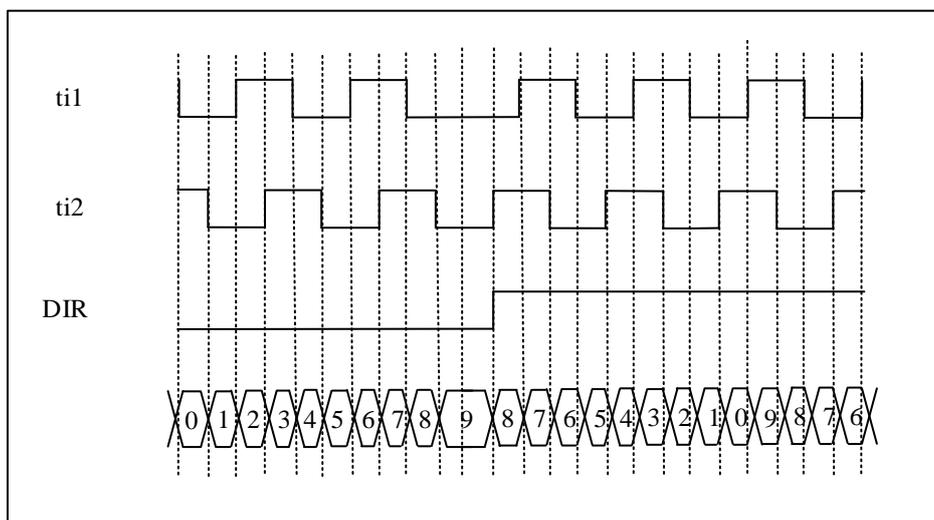


图 14-27 T2 是高电平时，计数器只在 TI1 计数

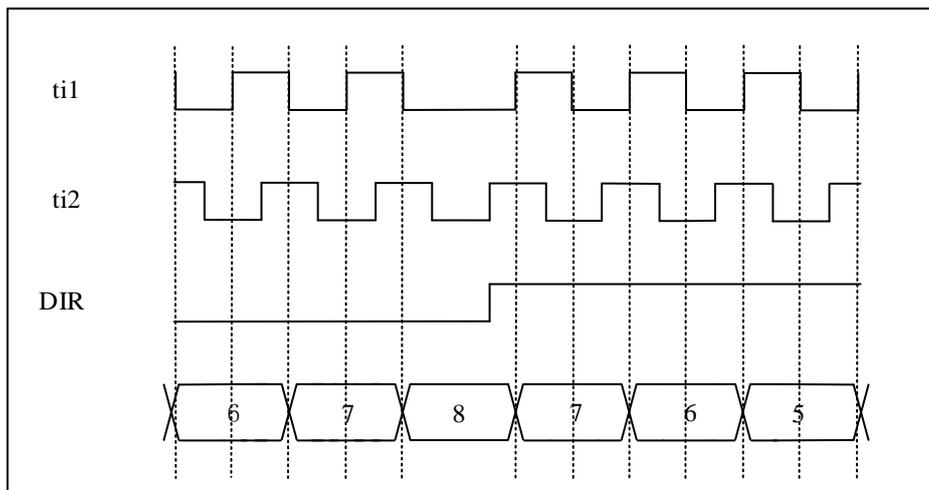
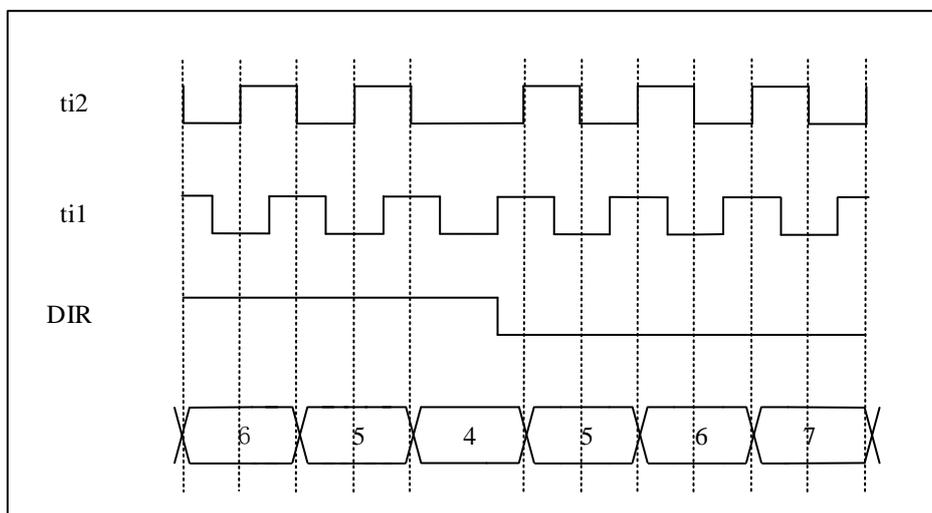


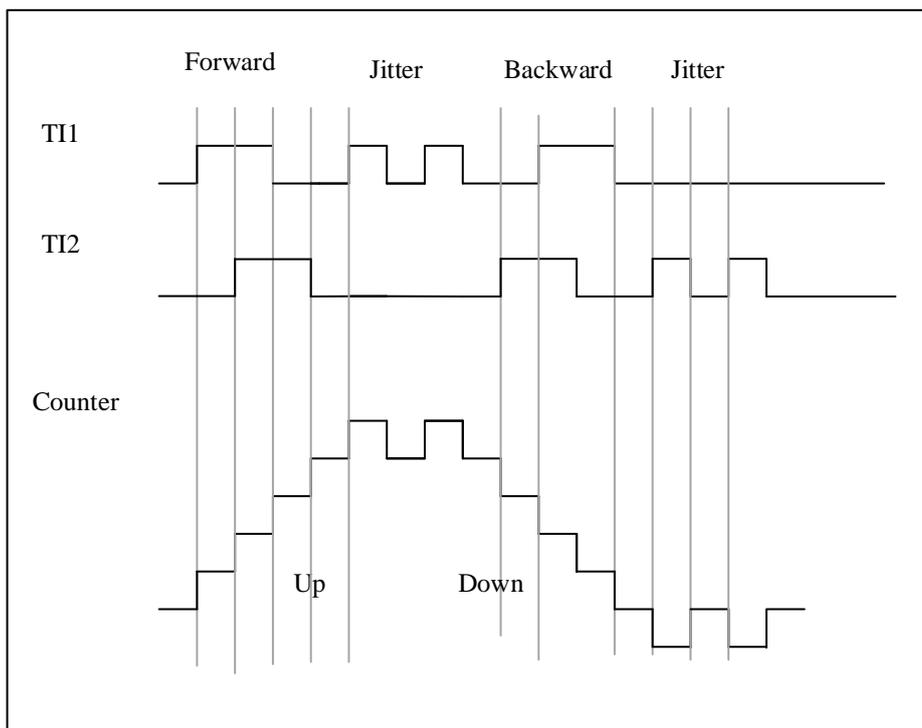
图 14-28 T1 是高电平时，计数器只在 TI2 计数



以下是选择了双边沿触发以抑制输入抖动的编码器示例：

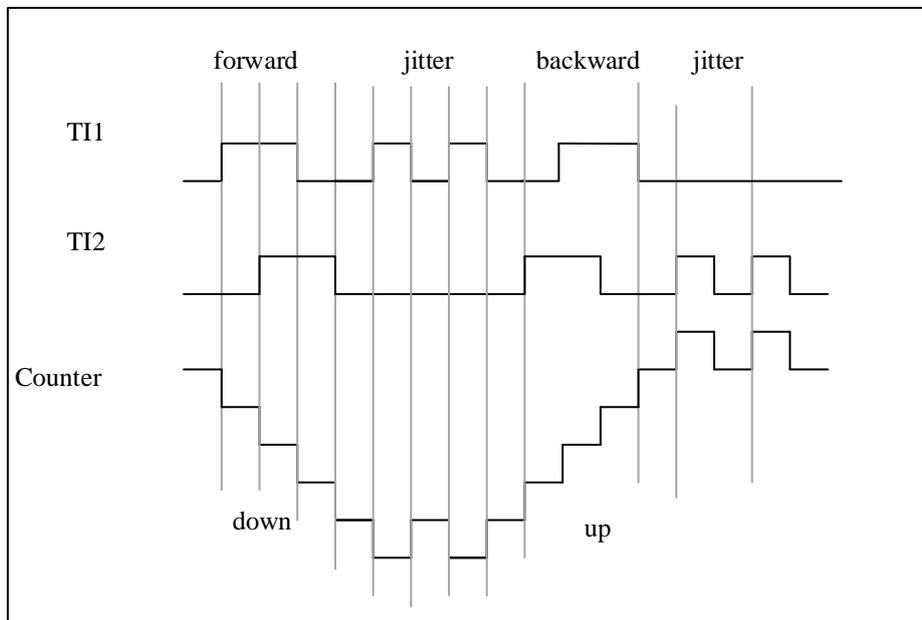
5. IC1FP1 映射到 TI1 (TIMx\_CCMOD1.CC1SEL = '01')，IC1FP1 不反相 (TIMx\_CCEN.CC1P = '0')；
6. IC1FP2 映射到 TI2 (TIMx\_CCMOD2.CC2SEL = '01')，IC2FP2 不反相 (TIMx\_CCEN.CC2P = '0')；
7. 输入在上升沿和下降沿均有效 (TIMx\_SMCTRL.SMSEL = '0011')；
8. 启用计数器 TIMx\_CTRL1.CNTEN = '1'；

图 14-29 编码器模式下的计数器操作实例



下图为 IC1FP1 极性反转时的计数器行为示例 (CC1P = '1', 其他配置同上)

图 14-30 IC1FP1 反相的编码器接口模式实例



### 14.3.16.2 脉冲电平编码模式

脉冲电平编码模式中，时钟是在 TI2 上单线上提供的，而计数方向是 TI1 输入提供的。

该模式通过 TIMx\_SMCTRL 寄存器中的 SMSEL[3:0]启用，具体如下。

1011：脉冲电平编码模式 2，计数器在时钟的上升沿和下降沿都被更新。该模式也可通过

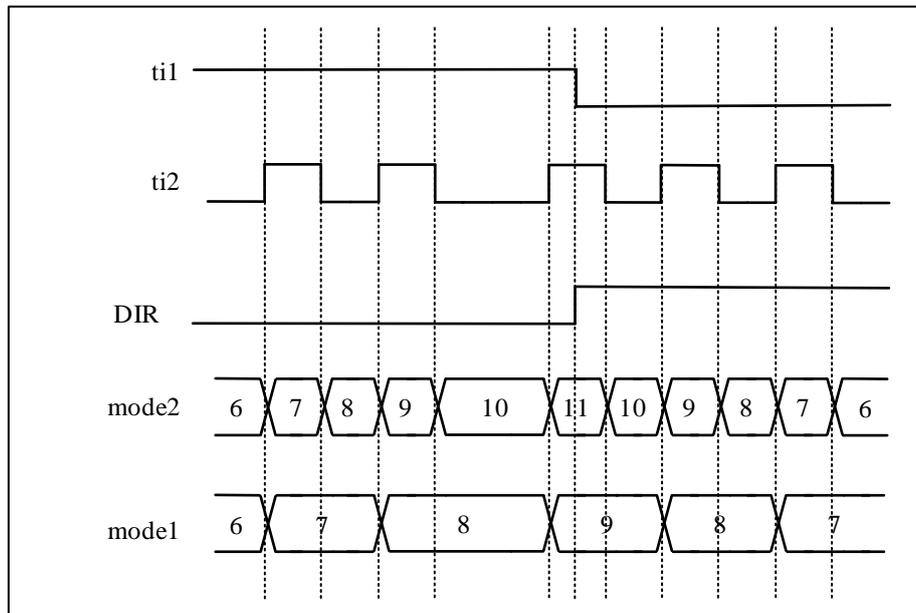
TIMx\_ENCMCTRL.ENCMD = ‘0111’启用。

1100：脉冲电平编码模式 1，根据 CC2P 值，计数器在单个时钟沿上更新。CC2P = 0 对应于上升沿计数，CC2P = 1 对应于下降沿计数。该模式也可通过 TIMx\_ENCMCTRL.ENCMD = ‘0110’启用。

TI1 的方向信号的极性是通过 CC1P 位来设置的。CC2P=0 时当 TI1 为高电平时向上计数，当 TI1 为低电平时向下计数；CC1P=1 时当 TI1 为低电平时向上计数，TI1 为高电平时向下计数。

下图以 CC1P=CC2P=0 为例：

图 14-31 脉冲电平编码模式 (CC1P=CC2P=0)



### 14.3.16.3 双脉冲编码模式

双脉冲编码模式中，时钟在两条线上被提供，根据不同的方向，一次只能提供一条，这样就有一条向上计数的时钟线和一条向下计数的时钟线。

该模式通过 TIMx\_SMCTRL 寄存器中的 SMSEL[3:0]位域启用，具体如下。

- 1000：双脉冲编码模式 2，计数器在两条时钟线中任何一条的上升沿和下降沿都被更新。CC1P 和 CC2P 位是对时钟空闲状态的编码。CCxP=0 对应于高电平空闲状态，CCxP=1 对应于低电平空闲状态。该模式也可通过 TIMx\_ENCMCTRL.ENCMD = ‘1011’启用。
- 1111：双脉冲编码模式 1，根据 CC1P 和 CC2P 位值，计数器在单个时钟沿上更新。CCxP=0 对应下降沿和高电平状态，CCxP=1 对应上升沿和低电平状态。该模式也可通过 TIMx\_ENCMCTRL.ENCMD = ‘1010’启用。

下表描述了计数方向与编码器信号和极性设置的关系

表 14-2 计数方向与编码器信号和极性设置的关系

双脉冲编码	SMSEL[3:0]	相对信号的电	TI1FP1 信号	TI2FP2 信号
-------	------------	--------	-----------	-----------

模式		平(TI1FP1 对应 TI2, TI2FP2 对应 TI1)	上升	下降	上升	下降
模式 2 CCxP=0	1000	高	向下计数	向下计数	向上计数	向上计数
		低	不计数	不计数	不计数	不计数
模式 2 CCxP=1	1000	高	不计数	不计数	不计数	不计数
		低	向下计数	向下计数	向上计数	向上计数
模式 1 CCxP=0	1111	高	不计数	向下计数	不计数	向上计数
		低	不计数	不计数	不计数	不计数
模式 1 CCxP=1	1111	高	不计数	不计数	不计数	不计数
		低	向下计数	不计数	向上计数	不计数

下图显示了双脉冲编码模式计数器计数方式

图 14-32 双脉冲编码模式 (CC1P = CC2P = 0)

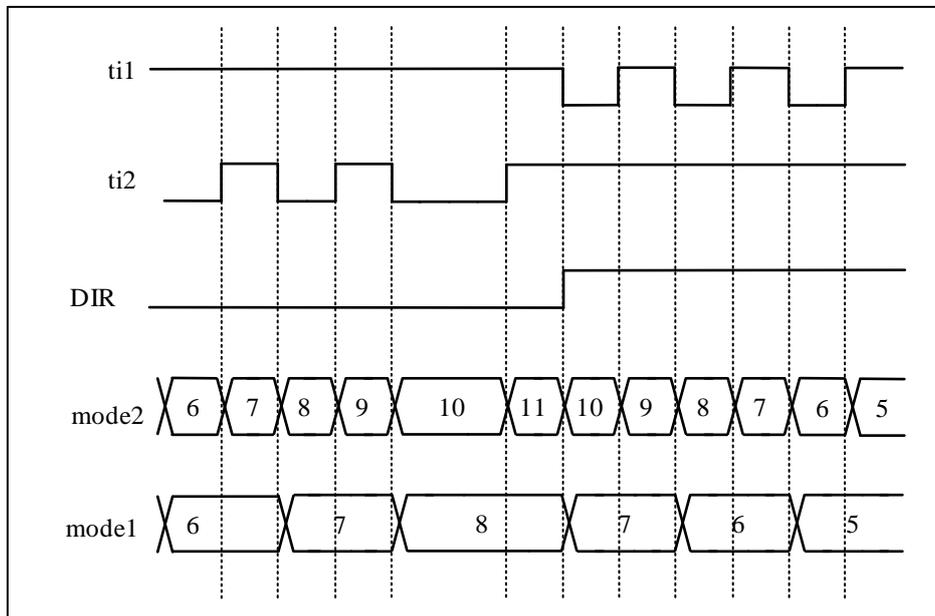
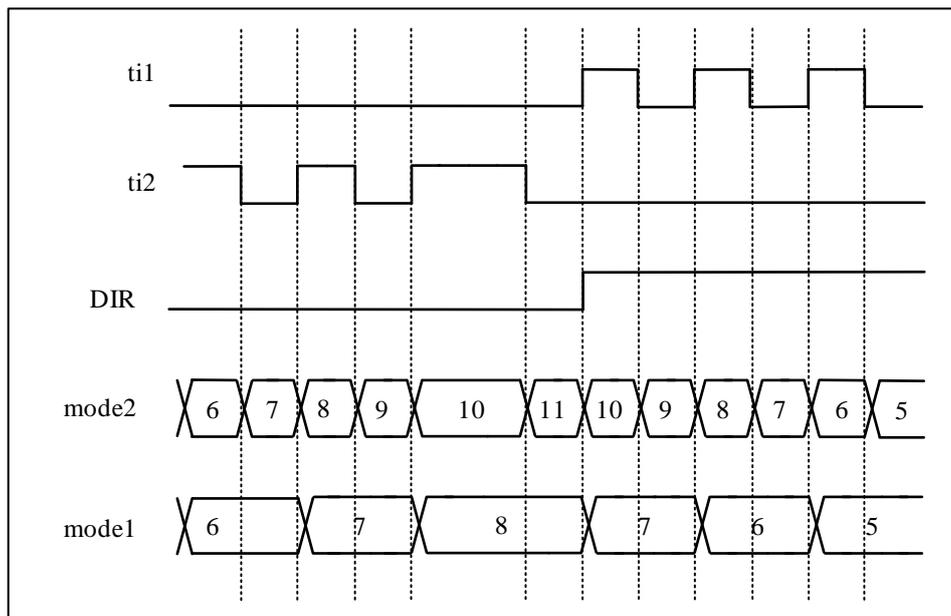


图 14-33 双脉冲编码模式 (CC1P = CC2P = 1)



### 14.3.16.4 CCW/CW 编码模式

CCW/CW 编码模式，在 T1 跳变时计数器递增，在 T2 跳变时计数器递减。可以配置计数器仅在上升沿变化或者在上升下降沿都变化。

该模式通过 TIMx\_ENCCTRL 寄存器中的 ENCMD [3:0]位域启用，具体如下。

- 1001: CCW/CW 编码模式 2，计数器在 TI1 的上升沿和下降沿递增计数，在 TI2 的上升沿和下降沿递减计数。
- 1000: CCW/CW 编码模式 1，计数器在 TI1 的上升沿递增计数，在 TI2 的上升沿递减计数。可通过 CCxP 选择 TI1 和 TI2 的边沿极性。

表 14-3 计数方向与编码器信号和极性设置的关系

计数模式	ENCMD[3:0]	变化边沿状态			
		TI1 上升沿	TI1 下降沿	TI2 上升沿	TI2 下降沿
CCW/CW 编码模式 1	1000	向上计数	不计数	向下计数	不计数
CCW/CW 编码模式 2	1001	向上计数	向上计数	向下计数	向下计数

图 14-34 CCW/CW 编码模式 2

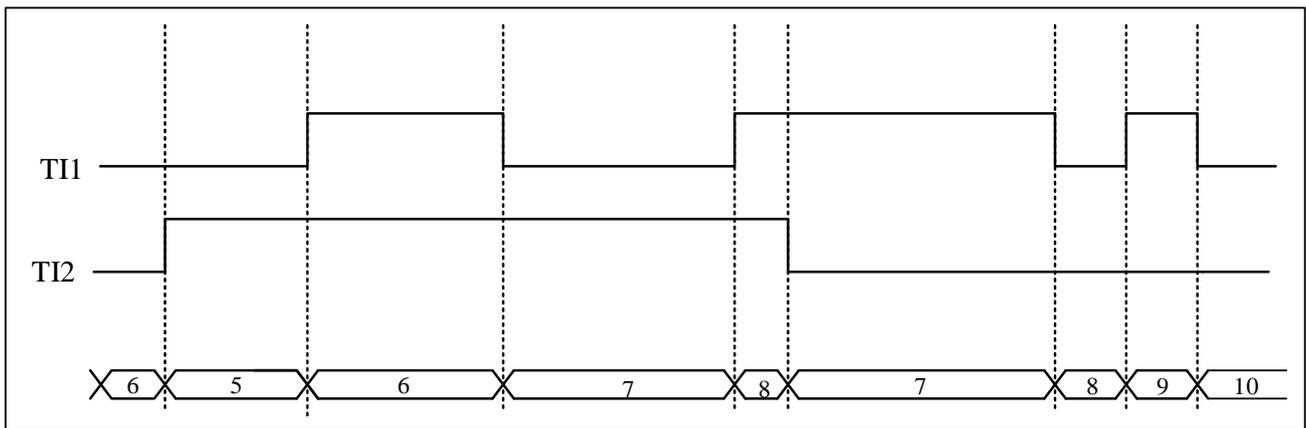
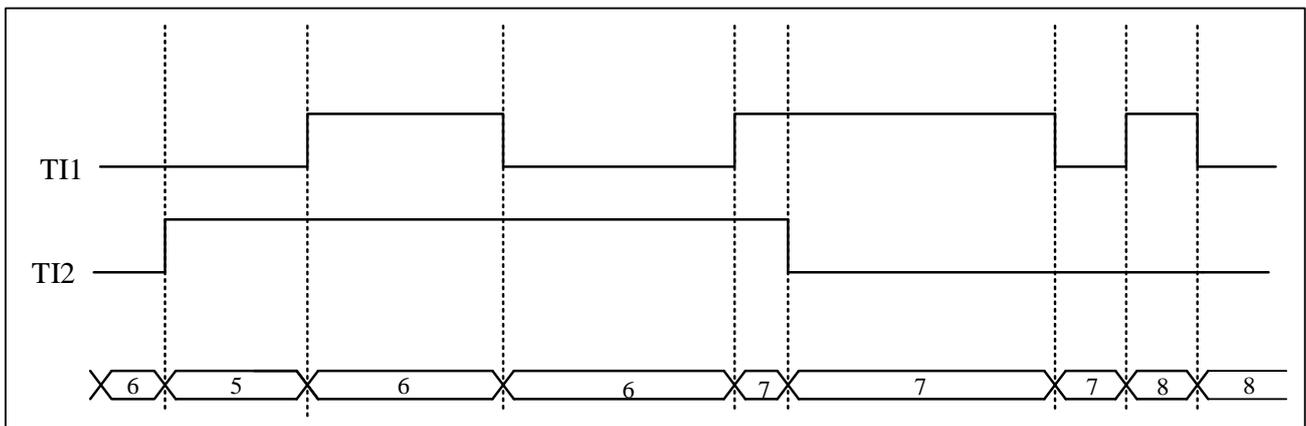


图 14-35 CCW/CW 编码模式 1 (CC1P=CC2P=0)



### 14.3.16.5 编码器 Z 相信号

编码器每旋转一周发一个脉冲，称之为零位脉冲或标识脉冲（即 Z 相信号），零位脉冲用于决定零位置或标识位置。编码器 Z 相信号可接入到 TI3 事件，实现 TI3 事件触发装载功能和 TI3 事件触发捕获差值功能，参考 [TI3 事件触发功能](#) 章节。

### 14.3.17 与霍尔传感器的接口

请查阅 12.3.21 节

## 14.4 TIMx 寄存器描述 (x=4)

关于在寄存器描述里面所用到的缩写，详见 1.1 节。

可以用半字（16 位）或字（32 位）的方式操作这些外设寄存器。

## 14.4.1 寄存器总览

**表 14-4 TIM4 寄存器总览**

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	TIM_CTRL1	Reserve															CISEL	Reserve	CLRSEL	Reserve			ARPEN	ONEPM	Reserve	UPDIS	UPRS	CAMSEL[1:0]	DIR	CNTEN			
004h	TIM_CTRL2	Reserve												TISEL	Reserve	CCDSEL	Reserve	MMSEL[2:0]			Reserve												
008h	TIM_STS	Reserve														TITF	Reserve	UDITF	Reserve				CC3OCF	CC2OCF	CC1OCF	C3LDCN/TITF	Reserve			CC3ITF	CC2ITF	CC1ITF	
00Ch	TIM_EVTGEN	Reserve																		TGN	Reserve	UDGN	Reserve					CC3GN	CC2GN	CC1GN			
010h	TIM_SMCTRL	Reserve															MSMD	EXTF[3:0]			EXTP	EXCEN	EXTPS[1:0]		SMSEL[3:0]			Reserve	TSEL[2:0]				
014h	TIM_DINTEN	Reserve												TDEN	Reserve	UDEN	Reserve	T1EN	U1EN	Reserve				CC3DEN	CC2DEN	CC1DEN	C3RSTCN/T1EN	Reserve			CC3IEN	CC2IEN	CC1IEN
018h	TIM_CCMOD1_OUT	Reserve																		OC2MD[2:0]		OC2CEN	OC2FEN	OC2PEN	CC2SEL[1:0]		OC1MD[2:0]		OC1CEN	OC2FEN	OC1PEN	CC1SEL[1:0]	
	TIM_CCMOD1_IN	Reserve																		IC2F[3:0]			IC2PSC[1:0]		CC2SEL[1:0]		IC1F[3:0]			IC1PSC[1:0]		CC1SEL[1:0]	
01Ch	TIM_CCMOD2_OUT	Reserve																		OC4MD[2:0]		OC4CEN	OC4FEN	OC4PEN	CC4SEL[1:0]		OC3MD[2:0]		OC3CEN	OC3FEN	OC3PEN	CC3SEL[1:0]	
	TIM_CCMOD2_IN	Reserve																		IC3F[3:0]			IC3PSC[1:0]		CC3SEL[1:0]								
024h	TIM_CCEN	Reserve																		CC3P	CC3EN	Reserve	CC2P	CC2EN	Reserve	CC1P	CC1EN	Reserve					
028h	TIM_CCDAT1	CCDAT1[31:0]																															
02Ch	TIM_CCDAT2	CCDAT2[31:0]																															
030h	TIM_CCDAT3	CCDAT3[31:0]																															
040h	TIM_PSC	Reserve															PSC[15:0]																
044h	TIM_AR	AR[31:0]																															
048h	TIM_CNT	CNT[31:0]																															
064h	TIM_C1FILT	Reserve	THRESH[5:0]				Reserve	WSIZE[5:0]				FILTEN	Reserve																				
068h	TIM_C2FILT	Reserve	THRESH[5:0]				Reserve	WSIZE[5:0]				FILTEN	Reserve																				
06Ch	TIM_C3FILT	Reserve	THRESH[5:0]				Reserve	WSIZE[5:0]				FILTEN	Reserve																				

074h	TIM_FILTO	Reserve				C3FILTO	C2FILTO	C1FILTO
088h	TIM_ENCDAT	ENCDAT[31:0]						
08Ch	TIM_ENCMCTRL	ENCDATS	Reserve			ENCMD[3:0]	C3LDCNTSEL[1:0]	C3LDCNTEN
090h	TIM_ENCLVR	LVR[31:0]						
094h	TIM_DCTRL	Reserve			DBADDR[5:0]	Reserve	DBLEN[5:0]	
098h	TIM_DADDR	BURST[31:0]						

## 14.4.2 控制寄存器 1 (TIMx\_CTRL1)

偏移地址: 0x00

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															C1SEL
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		CLRSEL	Reserved			ARPEN	ONEPM	Reserved		UPDIS	UPRS	CAMSEL[1:0]		DIR	CNTEN
		rw				rw	rw			rw	rw	rw		rw	rw

位域	名称	描述
31:17	Reserved	保留, 必须保持复位值
16	C1SEL	通道1选择 (Channel 1 selection) 0: 选择外部CH1 (来自IOM) 信号 1: 选择内部CH1 (来自COMP) 信号
15:14	Reserved	保留, 必须保持复位值
13	CLRSEL	OCxRef选择 (OCxRef selection) 0: 选择外部Ocxclr (ETR) 信号 1: 选择内部Ocxclr (来自COMP) 信号
12:10	Reserved	保留, 必须保持复位值
9	ARPEN	自动重载预装载允许位 (Auto-reload preload enable) 0: TIMx_AR 寄存器的影子寄存器禁用 1: TIMx_AR 寄存器的影子寄存器使能
8	ONEPM	单脉冲模式 (One pulse mode) 0: 禁用单脉冲模式, 发生更新事件时不影响计数器计数。 1: 使能单脉冲模式, 下次更新事件发生时计数器停止计数
7:6	Reserved	保留, 必须保持复位值
5	UPDIS	更新禁用 (Update disable) 该位用于启用/禁用软件生成的更新事件 (UEV) 事件。 0: 启用。 如果满足以下条件之一, 将生成 UEV: – 计数器上溢/下溢 – TIMx_EVTGEN.UDGN 位被设置 – 从模式控制器的更新生成 影子寄存器将使用预加载值进行更新。 1: UEV 禁用。 不生成更新事件, 影子寄存器 (AR、PSC 和 CC DATx) 保持它们的值。 如果 TIMx_EVTGEN.UDGN 位置位或从模式控制器发出硬件复位, 则重新初始化计数器和预分频器。

位域	名称	描述
4	UPRS	更新请求源 (Update request source) 该位用于通过软件选择 UEV 事件源。 0: 如果更新中断或 DMA 请求使能, 以下任何事件都会产生更新中断或 DMA 请求: – 计数器上溢/下溢 – TIMx_EVTGEN.UDGN 位被设置 – 从模式控制器的更新生成 1: 如果更新中断或 DMA 请求使能, 只有计数器上溢/下溢会产生更新中断或 DMA 请求。
3:2	CAMSEL[1:0]	选择中央对齐模式 (Center-aligned mode selection) 00: 边缘对齐模式。TIMx_CTRL1.DIR 指定向上计数或向下计数。 01: 中央对齐模式1。计数器在中央对齐模式下计数, 向下计数时输出比较中断标志位设置为 1。 10: 中央对齐模式2。计数器在中央对齐模式下计数, 向上计数时输出比较中断标志位设置为1。 11: 中央对齐模式3。计数器在中央对齐模式下计数, 向上计数或向下计数时输出比较中断标志位设置为 1。 <i>注意: 当计数器仍然启用时 (TIMx_CTRL1.CNTEN = 1), 不允许从边缘对齐模式切换到中央对齐模式。</i>
1	DIR	方向 (Direction) 0: 计数器向上计数; 1: 计数器向下计数。 <i>注: 当计数器配置为中央对齐模式或编码器模式时, 该位为只读。</i>
0	CNTEN	使能计数器 (Counter enable) 0: 禁止计数器; 1: 使能计数器。 <i>注: 在软件设置了CNTEN位后, 外部时钟、门控模式和编码器模式才能工作。触发模式可以自动地通过硬件设置CNTEN位。</i> <i>注: 软件设置了CNTEN位后, 需要等待至少两个TIMx_CLK, CNTEN才能从TIMx_PCLK同步到TIMx_CLK生效。</i>

### 14.4.3 控制寄存器 2 (TIMx\_CTRL2)

偏移地址: 0x04

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												TI1SEL	Reserved	CCDSEL	Reserved
												rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

MMSEL[3:0]	Reserved
------------	----------

rw

位域	名称	描述
31:20	Reserved	保留，必须保持复位值
19	TI1SEL	TI1选择（TI1 selection） 0：TIMx_CH1引脚连到TI1输入； 1：TIMx_CH1、TIMx_CH2和TIMx_CH3引脚经异或后连到TI1输入。
18	Reserved	保留，必须保持复位值
17	CCDSEL	捕获/比较的DMA选择（Capture/compare DMA selection） 0：当发生CCx事件时，送出CCx的DMA请求； 1：当发生更新事件时，送出CCx的DMA请求。
16	Reserved	保留，必须保持复位值
15:12	MMSEL[3:0]	主模式选择 这4位用于选择在主模式下发送到从定时器的同步信息（TRGO）。可能的组合如下： x000：复位 - 当TIMx_EVTGEN.UDGN置位或从模式控制器产生复位时，将出现TRGO脉冲。在后一种情况下，TRGO上的信号与实际复位相比有所延迟。 x001：使能 - TIMx_CTRL1.CNTEN位用作触发输出（TRGO）。有时需要同时启动多个定时器或者在一段时间内开启从定时器。 当TIMx_CTRL1.CNTEN位置位或门控模式下的触发输入为高电平时，计数器使能信号置位。 当计数器使能信号由触发输入控制时，TRGO上有一个延迟，除非选择了主/从模式（参见TIMx_SMCTRL.MSMD位的说明）。 x010：更新 - 选择更新事件作为触发输出（TRGO）。例如，主定时器时钟可用作从定时器预分频器。 x011：比较脉冲 - 当TIMx_STS.CC1ITF被设置时（即使它已经是高电平），即捕获或比较成功时，触发输出发送一个正脉冲（TRGO）。 x100：比较 - OC1REF信号用作触发输出（TRGO）。 x101：比较 - OC2REF信号用作触发输出（TRGO）。 x110：比较 - OC3REF信号用作触发输出（TRGO）。 其它：保留。
11:0	Reserved	保留，必须保持复位值

#### 14.4.4 状态寄存器（TIMx\_STS）

偏移地址：0x08

复位值：0x0000 0000

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

											rc_w0	rc_w0			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CC3OCF	CC2OCF	CC1OCF	C3LDCNT ITF	Reserved				CC3ITF	CC2ITF	CC1ITF	
				rc_w0	rc_w0	rc_w0	rc_w0					rc_w0	rc_w0	rc_w0	

位域	名称	描述
31:19	Reserved	保留，必须保持复位值
18	TITF	触发器中断标记（Trigger interrupt flag） 当发生触发事件（当从模式控制器处于除门控模式外的其它模式时，在TRGI输入端检测到有效边沿，或门控模式下的任一边沿）时由硬件对该位置‘1’。它由软件清‘0’。 0：无触发器事件产生； 1：触发中断等待响应。
17	Reserved	保留，必须保持复位值
16	UDITF	更新中断标志（Update interrupt flag） 当在以下条件下发生更新事件时，该位由硬件设置： - 当 TIMx_CTRL1.UPDIS = 0 时，并且重复计数器值上溢或下溢（当重复计数器等于 0 时生成更新事件UEV）。 - 当 TIMx_CTRL1.UPRS = 0 时，TIMx_CTRL1.UPDIS = 0，并通过软件设置 TIMx_EVTGEN.UDGN 位以重新初始化 CNT。 - 当 TIMx_CTRL1.UPRS = 0 时，TIMx_CTRL1.UPDIS = 0，并且计数器 CNT 由触发事件重新初始化。（参见 TIMx_SMCTRL 寄存器说明） 该位由软件清零。 0：未发生更新事件 1：发生更新中断
15:12	Reserved	保留，必须保持复位值
11	Reserved	保留，必须保持复位值
10	CC3OCF	捕获/比较3重复捕获标记（Capture/Compare 3 overcapture flag） 参见CC1OCF描述。
9	CC2OCF	捕获/比较2重复捕获标记（Capture/Compare 2 overcapture flag） 参见CC1OCF描述。
8	CC1OCF	捕获/比较1重复捕获标记（Capture/Compare 1 overcapture flag） 仅当相应的通道被配置为输入捕获时，该标记可由硬件置1。写0可清除该位。 0：无重复捕获产生； 1：计数器的值被捕获到TIMx_CCDAT1寄存器时，CCIITF的状态已经为‘1’。
7	C3LDCNTITF	TI3有效事件触发计数器装载为LVR值中断标记 0：无TI3有效事件触发计数器装载为LVR值发生 1：发生TI3有效事件触发计数器装载为LVR值
6:3	Reserved	保留，必须保持复位值

位域	名称	描述
2	CC3ITF	捕获/比较3中断标记 (Capture/Compare 3 interrupt flag) 参考CC1ITF描述。
1	CC2ITF	捕获/比较2中断标记 (Capture/Compare 2 interrupt flag) 参考CC1ITF描述。
0	CC1ITF	捕获/比较1中断标记 (Capture/Compare 1 interrupt flag) <b>如果通道CC1配置为输出模式:</b> 除中央对齐模式外, 当计数器值与比较值相同时, 该位由硬件设置 (参见TIMx_CTRL1.CAMSEL 位描述)。 该位由软件清零。 0: 未发生匹配。 1: TIMx_CNT 的值与 TIMx_CC1 的值相同。 当 TIMx_CC1 的值大于 TIMx_ARR 的值时, 如果计数器溢出 (在向上计数和向上/向下计数模式下) 和向下计数模式下溢, 则 TIMx_STS.CC1ITF 位将变为高电平。 <b>如果通道CC1配置为输入模式:</b> 当捕捉事件发生时, 该位由硬件设置。 该位由软件或读取 TIMx_CC1 清零。 0: 未发生输入捕捉。 1: 发生输入捕捉。 计数器值已在 TIMx_CC1 中捕获。 在 IC1 上检测到与所选极性相同的边沿。

### 14.4.5 事件产生寄存器 (TIMx\_EVTGEN)

偏移地址:0x0C

复位值:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				TGN	Reserved	UDGN		Reserved				CC3GN	CC2GN	CC1GN	
				w			w						w	w	w

位域	名称	描述
31:11	Reserved	保留, 必须保持复位值
10	TGN	产生触发事件 (Trigger generation) 当由软件置位时, 该位可以产生一个触发事件。 而此时TIMx_STS.TITF = 1, 如果相应的中断和DMA被使能, 就会产生相应的中断和DMA。 该位由硬件自动清零。 0: 无动作 1: 产生触发事件
9	Reserved	保留, 必须保持复位值
8	UDGN	产生更新事件 (Update generation) 该位由软件置'1', 由硬件自动清'0'。

位域	名称	描述
		当由软件设置时，该位可以生成更新事件。而此时计数器会重新初始化，预分频计数器会被清零，计数器在中央对齐或向上计数模式下会被清零，但在向下计数模式下取 TIMx_AR 寄存器的值。该位由硬件自动清零。 0: 无动作 1: 生成更新事件
7:3	Reserved	保留，必须保持复位值
2	CC3GN	产生捕获/比较3事件 (Capture/Compare 3 generation) 参考CC1GN描述。
1	CC2GN	产生捕获/比较2事件 (Capture/Compare 2 generation) 参考CC1GN描述。
0	CC1GN	产生捕获/比较1事件 (Capture/Compare 1 generation) 当由软件设置时，该位可以产生一个捕获/比较事件。该位由硬件自动清零。 <b>CC1对应通道为输出模式时：</b> TIMx_STS.CC1ITF 标志将被拉高，如果相应的中断和 DMA 被使能，就会产生相应的中断和 DMA。 <b>CC1对应通道为输入模式时：</b> TIMx_CC DAT1 将捕获当前计数器值，并将 TIMx_STS.CC1ITF 标志拉高，如果相应的中断和 DMA 被使能，则会产生相应的中断和 DMA。如果 TIMx_STS.CC1ITF 已经拉高，则拉高 TIMx_STS.CC1OCF。 0: 无动作 1: 生成 CC1 捕获/比较事件

### 14.4.6 从模式控制寄存器 (TIMx\_SMCTRL)

偏移地址: 0x10

复位值: 0x0000 0000

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

位域	名称	描述
31:17	Reserved	保留，必须保持复位值
16	MSMD	主/从模式 (Master/slave mode) 0: 无作用; 1: 触发输入 (TRGI) 上的事件被延迟了，以允许在当前定时器 (通过TRGO) 与它的从定时器间

		的完美同步。这对要求把几个定时器同步到一个单一的外部事件时是非常有用的。
15:12	EXTF[3:0]	<p>外部触发滤波 (External trigger filter)</p> <p>这些位定义了ETRP数字滤波器长度(RCC_TIMFILTCFG.TIM4FILTCLK[4:0] 寄存器决定相应数字滤波器的采样频率)。数字滤波器由一个事件计数器组成, 它记录到N个事件后会产生一个输出的跳变:</p> <p>0000: 无滤波器, 以TIM内部工作时钟采样</p> <p>0001: N=1</p> <p>0010: N=2</p> <p>0011: N=3</p> <p>0100: N=4</p> <p>0101: N=5</p> <p>0110: N=6</p> <p>0111: N=7</p> <p>1xxx: N=8</p>
11	EXTP	<p>外部触发极性 (External trigger polarity)</p> <p>该位选择是用ETR还是ETR的反相来作为触发操作</p> <p>0: ETR高电平或上升沿有效;</p> <p>1: ETR低电平或下降沿有效。</p>
10	EXCEN	<p>外部时钟使能位 (External clock enable) 该位启用外部时钟模式2。启用后, 计数器由ETRF信号上的任意有效边沿驱动。</p> <p>0: 禁止外部时钟模式2;</p> <p>1: 使能外部时钟模式2。</p> <p><i>注 1: 当同时使能外部时钟模式 1 和外部时钟模式 2 时, 外部时钟的输入为 ETRF。</i></p> <p><i>注2: 以下从机模式可以与外部时钟模式2同时使用: 复位模式、门控模式和触发模式; 但是, TRGI 无法连接到 ETRF (TIMx_SMCTRL.TSEL ≠ '111')。</i></p> <p><i>注 3: 设置 TIMx_SMCTRL.EXCEN 位与选择外部时钟模式 1 并将 TRGI 连接到 ETRF (TIMx_SMCTRL.SMSEL = 0111 和 TIMx_SMCTRL.TSEL = 111) 的效果相同</i></p>
9:8	EXTPS[1:0]	<p>外部触发预分频 (External trigger prescaler)</p> <p>外部触发信号 ETRP 的频率必须最多为 TIMxCLK 频率的 1/4。当输入更快的外部时钟时, 可以使用预分频器来降低 ETRP 的频率。</p> <p>00: 关闭预分频;</p> <p>01: ETRP频率除以2;</p> <p>10: ETRP频率除以4;</p> <p>11: ETRP频率除以8。</p>
7:4	SMSEL[3:0]	<p>从模式选择 (Slave mode selection)</p> <p>当选择了外部信号, 触发信号 (TRGI) 的有效边沿与选中的外部输入极性相关 (见输入控制寄存器和控制寄存器的说明)</p> <p>0000: 关闭从模式 – 如果CNTEN=1, 则预分频器直接由内部时钟驱动。</p> <p>0001: 编码器模式1 – 根据TI2FP2的电平, 计数器在TI1FP1的边沿向上/下计数。</p> <p>0010: 编码器模式2 – 根据TI1FP1的电平, 计数器在TI2FP2的边沿向上/下计数。011: 编码器模式3 – 根据另一个信号的输入电平, 计数器在TI1FP1和TI2FP2的边沿向上/下计数。</p>

		<p>0100: 复位模式 – 在选定触发输入 (TRGI) 的上升沿, 计数器重新初始化并更新影子寄存器。</p> <p>0101: 门控模式 – 当触发输入 (TRGI) 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 (但不复位)。计数器的启动和停止都是受控的。</p> <p>0110: 触发模式 – 计数器在触发输入TRGI的上升沿启动 (但不复位), 只有计数器的启动是受控的。</p> <p>0111: 外部时钟模式1 – 选中的触发输入 (TRGI) 的上升沿驱动计数器。</p> <p>1000: 双脉冲编码模式2。</p> <p>1001: 正交编码器模式4 – 根据TI2FP2的电平, 计数器在TI1FP1的边沿向上/下计数。通过CC1P选择计数边沿。</p> <p>1010: 正交编码器模式5 – 根据TI1FP1的电平, 计数器在TI2FP2的边沿向上/下计数。通过CC2P选择计数边沿。</p> <p>1011: 脉冲电平编码模式2。</p> <p>1100: 脉冲电平编码模式1。通过CC2P设置TI2FP2的计数边沿。</p> <p>1101: 保留。</p> <p>1110: 保留。</p> <p>1111: 双脉冲编码模式1。通过CC1P和CC2P设置TI1FP1和TI2FP2的计数敏感边沿。</p> <p><i>注: 如果TIIF_ED被选为触发输入 (TSEL=100) 时, 不要使用门控模式。这是因为, TIIF_ED在每次TIIF变化时输出一个脉冲, 然而门控模式是要检查触发输入的电平。</i></p>
3	Reserved	保留, 必须保持复位值
2:0	TSEL[2:0]	<p>触发选择 (Trigger selection)</p> <p>这3位选择用于同步计数器的触发输入。</p> <p>000: 内部触发0 (ITR0)    100: TI1的边沿检测器 (TIIF_ED)</p> <p>001: 内部触发1 (ITR1)    101: 滤波后的定时器输入1 (TI1FP1)</p> <p>010: 内部触发2 (ITR2)    110: 滤波后的定时器输入2 (TI2FP2)</p> <p>011: 内部触发3 (ITR3)    111: 外部触发输入 (ETRF)</p> <p>更多有关ITRx的细节, 参见表14-5。</p> <p><i>注: 这些位只能在未用到 (如SMSEL=0000) 时被改变, 以避免在改变时产生错误的边沿检测。</i></p>

**表 14-5 TIMx 内部触发连接**

Slave timer	ITR0 (TSEL = 000)	ITR1 (TSEL = 001)	ITR2 (TSEL = 010)	ITR3 (TSEL = 011)
<b>TIM1</b>	TIM3	NA	NA	NA
<b>TIM3</b>	TIM1	NA	NA	NA
<b>TIM4</b>	NA	NA	NA	NA

### 14.4.7 DMA/中断使能寄存器 (TIMx\_DINTEN)

偏移地址: 0x14

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										TDEN	Reserved	UDEN	Reserved	TIEN	UIEN

										rw				rw				rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
Reserved					CC3DEN	CC2DEN	CC1DEN	C3LDCNT IEN	Reserved					CC3IEN	CC2IEN	CC1IEN				
					rw	rw	rw						rw	rw	rw					

位域	名称	描述
31:22	Reserved	保留，必须保持复位值
21	TDEN	允许触发DMA请求（Trigger DMA request enable） 0：禁止触发DMA请求； 1：允许触发DMA请求。
20	Reserved	保留，必须保持复位值
19	UDEN	允许更新的DMA请求（Update DMA request enable） 0：禁止更新的DMA请求； 1：允许更新的DMA请求。
18	Reserved	保留，必须保持复位值
17	TIEN	触发中断使能（Trigger interrupt enable） 0：禁止触发中断； 1：使能触发中断。
16	UIEN	允许更新中断（Update interrupt enable） 0：禁止更新中断； 1：允许更新中断。
15:11	Reserved	保留，必须保持复位值
10	CC3DEN	允许捕获/比较3的DMA请求（Capture/Compare 3 DMA request enable） 0：禁止捕获/比较3的DMA请求； 1：允许捕获/比较3的DMA请求。
9	CC2DEN	允许捕获/比较2的DMA请求（Capture/Compare 2 DMA request enable） 0：禁止捕获/比较2的DMA请求； 1：允许捕获/比较2的DMA请求。
8	CC1DEN	允许捕获/比较1的DMA请求（Capture/Compare 1 DMA request enable） 0：禁止捕获/比较1的DMA请求； 1：允许捕获/比较1的DMA请求。
7	C3LDCNTIEN	允许TI3有效事件触发计数器装载为LVR值中断 0：禁止TI3有效事件触发计数器装载为LVR值中断 1：允许TI3有效事件触发计数器装载为LVR值中断
6:3	Reserved	保留，必须保持复位值
2	CC3IEN	允许捕获/比较3中断（Capture/Compare 3 interrupt enable） 0：禁止捕获/比较3中断； 1：允许捕获/比较3中断。

位域	名称	描述
1	CC2IEN	允许捕获/比较2中断 (Capture/Compare 2 interrupt enable) 0: 禁止捕获/比较2中断; 1: 允许捕获/比较2中断。
0	CC1IEN	允许捕获/比较1中断 (Capture/Compare 1 interrupt enable) 0: 禁止捕获/比较1中断; 1: 允许捕获/比较1中断。

## 14.4.8 捕获/比较模式寄存器 1 (TIMx\_CCMOD1)

偏移地址：0x18

复位值：0x0000 0000

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

**输出比较模式：**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2MD[2:0]			OC2CEN	OC2FEN	OC2PEN	CC2SEL[1:0]		OC1MD[2:0]			OC1CEN	OC1FEN	OC1PEN	CC1SEL[1:0]	
rw			rw	rw	rw	rw		rw			rw	rw	rw	rw	

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:13	OC2MD[2:0]	输出比较2模式（Output Compare 2 mode）
12	OC2CEN	输出比较2清0使能（Output Compare 2 clear enable）
11	OC2FEN	输出比较2快速使能（Output Compare 2 fast enable）
10	OC2PEN	输出比较2预装载使能（Output Compare 2 preload enable）
9:8	CC2SEL[1:0]	捕获/比较2选择。（Capture/Compare 2 selection） 该位定义通道的方向（输入/输出），及输入脚的选择： 00：CC2通道被配置为输出； 01：CC2通道被配置为输入，IC2映射在TI2上； 10：CC2通道被配置为输入，IC2映射在TI1上； 11：CC2通道被配置为输入，IC2映射在TRC上。此模式仅工作在内部触发输入被选中时（由TIMx_SMCTRL寄存器的TSEL位选择）。 <i>注：CC2SEL仅在通道关闭时（TIMx_CCEN寄存器的CC2EN=0）才是可写的。</i>
7:5	OC1MD[2:0]	输出比较1模式（Output Compare 1 mode） 这些位用于管理输出参考信号 OC1REF，它决定了 OC1 和 OC1N 的值，在高电平有效，而 OC1 和 OC1N 的有效电平取决于 TIMx_CCEN.CC1P 和 TIMx_CCEN.CC1NP 位。 000：冻结。TIMx_CCDAT1 寄存器和计数器 TIMx_CNT 之间的比较对 OC1REF 信号没有影响。 001：将通道 1 设置为匹配时的有效电平。当 TIMx_CCDAT1 = TIMx_CNT 时，OC1REF 信号将被强制为高电平。 010：将通道 1 设置为匹配时的无效电平。当 TIMx_CCDAT1 = TIMx_CNT 时，OC1REF 信号将被强制为低电平。

位域	名称	描述
		011: 翻转。当 $TIMx\_CCDAT1 = TIMx\_CNT$ 时, OC1REF 信号将被翻转。 100: 强制无效电平。 OC1REF 信号被强制为低电平。 101: 强制有效电平。 OC1REF 信号被强制为高电平。 110: PWM 模式 1 - 在向上计数模式下, 如果 $TIMx\_CNT < TIMx\_CCDAT1$ , 则通道 1 的 OC1REF 信号为高电平, 否则为低电平。在向下计数模式下, 如果 $TIMx\_CNT > TIMx\_CCDAT1$ , 则通道 1 的 OC1REF 信号为低电平, 否则为高电平。 111: PWM 模式 2 - 在向上计数模式下, 如果 $TIMx\_CNT < TIMx\_CCDAT1$ , 则通道 1 的 OC1REF 信号为低电平, 否则为高电平。在向下计数模式下, 如果 $TIMx\_CNT > TIMx\_CCDAT1$ , 则通道 1 的 OC1REF 信号为高电平, 否则为低电平。 <i>注 1: 在 PWM 模式 1 或 PWM 模式 2 中, OC1REF 电平仅在比较结果改变或输出比较模式从冻结模式切换到 PWM 模式时才会改变。</i>
4	OC1CEN	输出比较1清'0'使能 (Output Compare 1 clear enable) 0: OC1REF 不受ETRF输入的影响; 1: 一旦检测到ETRF输入高电平, 清除OC1REF=0。
3	OC1FEN	输出比较1 快速使能 (Output Compare 1 fast enable) 该位用于加快CC输出对触发输入事件的响应。 0: 根据计数器与CCDAT1的值, CC1正常操作, 即使触发器是打开的。当触发器的输入有一个有效沿时, 激活CC1输出的最小延时为5个时钟周期。 1: 输入到触发器的有效沿的作用就像发生了一次比较匹配。因此, OC1被设置为比较电平而与比较结果无关。采样触发器的有效沿和CC1输出间的延时被缩短为3个时钟周期。 OCxPEN只在通道被配置成PWM1或PWM2模式时起作用。
2	OC1PEN	输出比较 1 预加载使能 (Output Compare 1 preload enable) 0: 禁用 $TIMx\_CCDAT1$ 寄存器的预加载功能。 支持随时对 $TIMx\_CCDAT1$ 寄存器进行写操作, 写入的值立即生效。 1: 使能 $TIMx\_CCDAT1$ 寄存器的预加载功能。 仅对预加载寄存器进行读写操作。当更新事件发生时, $TIMx\_CCDAT1$ 的值被加载到影子寄存器中。 <i>注 1: 只有当 <math>TIMx\_CTRL1.ONEPM = 1</math> (在单脉冲模式下) 时, 才能使用 PWM 模式而不验证预加载寄存器, 否则无法预测其他行为。</i>
1:0	CC1SEL[1:0]	捕获/比较1 选择。(Capture/Compare 1 selection) 这2位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC1通道被配置为输出; 01: CC1通道被配置为输入, IC1映射在TI1上; 10: CC1通道被配置为输入, IC1映射在TI2上; 11: CC1通道被配置为输入, IC1映射在TRC上。此模式仅工作在内部触发输入被选中时(由 $TIMx\_SMCTRL$ 寄存器的TSEL位选择)。 <i>注: CC1SEL 仅在通道关闭时 (<math>TIMx\_CCEN</math>寄存器的CC1EN=0) 才是可写的。</i>

**输入捕获模式：**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC2F[3:0]				IC2PSC[1:0]		CC2SEL[1:0]		IC1F[3:0]				IC1PSC[1:0]		CC1SEL[1:0]	
rw				rw		rw		rw				rw		rw	

位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:12	IC2F[3:0]	输入捕获2滤波器（Input capture 2 filter）
11:10	IC2PSC[1:0]	输入/捕获2预分频器（Input capture 2 prescaler）
9:8	CC2SEL[1:0]	捕获/比较2选择（Capture/Compare 2 selection） 这2位定义通道的方向（输入/输出），及输入脚的选择： 00：CC2通道被配置为输出； 01：CC2通道被配置为输入，IC2映射在TI2上； 10：CC2通道被配置为输入，IC2映射在TI1上； 11：CC2通道被配置为输入，IC2映射在TRC上。此模式仅工作在内部触发输入被选中时（由TIMx_SMCTRL寄存器的TSEL位选择）。 <i>注：CC2SEL仅在通道关闭时（TIMx_CCEN寄存器的CC2EN=0）才是可写的。</i>
7:4	IC1F[3:0]	输入捕获1滤波器（Input capture 1 filter） 这几位定义了TI1数字滤波器长度(RCC_TIMFILTCFG.TIM4FILTCLK[4:0] 寄存器决定相应数字滤波器的采样频率)。数字滤波器由一个事件计数器组成，它记录到N个事件后会产生一个输出的跳变： 0000：无滤波器，以TIM内部工作时钟采样 0001：N=1 0010：N=2 0011：N=3 0100：N=4 0101：N=5 0110：N=6 0111：N=7 1xxx：N=8
3:2	IC1PSC[1:0]	输入/捕获1预分频器（Input capture 1 prescaler） 这2位定义了CC1输入（IC1）的预分频系数。 一旦TIMx_CCEN.CC1EN=0，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	CC1SEL[1:0]	捕获/比较1选择 (Capture/Compare 1 Selection) 这2位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC1通道被配置为输出; 01: CC1通道被配置为输入, IC1映射在TI1上; 10: CC1通道被配置为输入, IC1映射在TI2上; 11: CC1通道被配置为输入, IC1映射在TRC上。此模式仅工作在内部触发输入被选中时 (由TIMx_SMCTRL寄存器的TSEL位选择)。 <i>注: CC1SEL仅在通道关闭时(TIMx_CCEN寄存器的CCIEN=0)才是可写的。</i>

### 14.4.9 捕获/比较模式寄存器 2 (TIMx\_CCMOD2)

偏移地址: 0x1C

复位值: 0x0000 0000

参看以上 CCMOD1 寄存器的描述

输出比较模式:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							OC3MD[2:0]	OC3CEN	OC3FEN	OC3PEN	CC3SEL[1:0]				
							rw	rw	rw	rw	rw				

位域	名称	描述
31:8	Reserved	保留, 必须保持复位值
7:5	OC3MD[2:0]	输出比较3模式 (Output compare 3 mode)
4	OC3CEN	输出比较3清0使能 (Output compare 3 clear enable)
3	OC3FEN	输出比较3快速使能 (Output compare 3 fast enable)
2	OC3PEN	输出比较3预装载使能 (Output compare 3 preload enable)
1:0	CC3SEL[1:0]	捕获/比较3选择 (Capture/Compare 3 selection) 这2位定义通道的方向(输入/输出), 及输入脚的选择: 00: CC3通道被配置为输出; 01: CC3通道被配置为输入, IC3映射在TI3上; 10: 保留 11: CC3通道被配置为输入, IC3映射在TRC上。此模式仅工作在内部触发输入被选中时 (由TIMx_SMCTRL寄存器的TSEL位选择)。 <i>注: CC3SEL仅在通道关闭时(TIMx_CCEN寄存器的CC3EN=0)才是可写的。</i>

输入捕获模式:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						IC3F[3:0]			IC3PSC[1:0]			CC3SEL[1:0]			
						rw			rw			rw			

位域	名称	描述
31:8	Reserved	保留，必须保持复位值
7:4	IC3F[3:0]	输入捕获3滤波器 (Input capture 3 filter)
3:2	IC3PSC[1:0]	输入/捕获3预分频器 (Input capture 3 prescaler)
1:0	CC3SEL[1:0]	捕获/比较3选择 (Capture/compare 3 selection) 这2位定义通道的方向(输入/输出)，及输入脚的选择： 00：CC3通道被配置为输出； 01：CC3通道被配置为输入，IC3映射在TI3上； 10：CC3通道被配置为输入，IC3映射在TI4上； 11：CC3通道被配置为输入，IC3映射在TRC上。此模式仅工作在内部触发输入被选中时(由TIMx_SMCTRL寄存器的TSEL位选择)。 注：CC3SEL仅在通道关闭时(TIMx_CCEN寄存器的CC3EN=0)才是可写的。

### 14.4.10 捕获/比较使能寄存器 (TIMx\_CCEN)

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved				CC3P	CC3EN	Reserved		CC2P	CC2EN	Reserved			CC1P	CC1EN	Reserved	
				rw	rw			rw	rw				rw	rw		

位域	名称	描述
31:12	Reserved	保留，必须保持复位值
11	CC3P	捕获/比较3输出极性 (Capture/Compare 3 output polarity) 参考TIMx_CCEN.CC1P的描述。
10	CC3EN	捕获/比较3输出使能 (Capture/Compare 3 output enable) 参考TIMx_CCEN.CC1E 的描述。
9:8	Reserved	保留，必须保持复位值
7	CC2P	捕获/比较2输出极性 (Capture/Compare 2 output polarity) 参考TIMx_CCEN.CC1P的描述。

位域	名称	描述
6	CC2EN	捕获/比较2输出使能 (Capture/Compare 2 output enable) 参考TIMx_CCEN.CC1EN的描述。
5:4	Reserved	保留, 必须保持复位值
3	CC1P	捕获/比较1输出极性 (Capture/Compare 1 output polarity) <b>CC1对应通道为输出模式时:</b> 0: OC1 高电平有效 1: OC1 低电平有效 <b>CC1对应通道为输入模式时:</b> 此时, 该位用于选择是使用IC1还是IC1的反相信号作为触发信号或捕捉信号。 0: 非反相: 当 IC1 产生上升沿时发生捕获动作。 当用作外部触发时, IC1 是非反相的。 1: 反相: 当 IC1 产生下降沿时发生捕获动作。 当用作外部触发时, IC1 被反相。
2	CC1EN	捕获/比较1输出使能 (Capture/Compare 1 output enable) <b>CC1通道配置为输出:</b> 0: 关闭— OC1禁止输出, 因此OC1的输出电平依赖于MOEN、OSSI、OSSR、OI1、OI1N和CC1NEN位的值。 1: 开启— OC1信号输出到对应的输出引脚, 其输出电平依赖于MOEN、OSSI、OSSR、OI1、OI1N和CC1NEN位的值。 <b>CC1通道配置为输入:</b> 该位决定了计数器的值是否能捕获入TIMx_CC1DAT1寄存器。 0: 捕获禁止; 1: 捕获使能。
1:0	Reserved	保留, 必须保持复位值

**表 14-6 标准 OCx 的输出控制位**

CCxEN	OCx output status
0	Disable output (OCx=0)
1	OCx = OCxREF + polarity

注: 连接到标准 OCx 通道的外部 I/O 引脚的状态取决于 OCx 通道状态以及 GPIO 和 AFIO 寄存器。

### 14.4.11 捕获/比较寄存器 1 (TIMx\_CC1DAT1)

偏移地址: 0x28

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCDAT1[31:0]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCDAT1[31:0]															

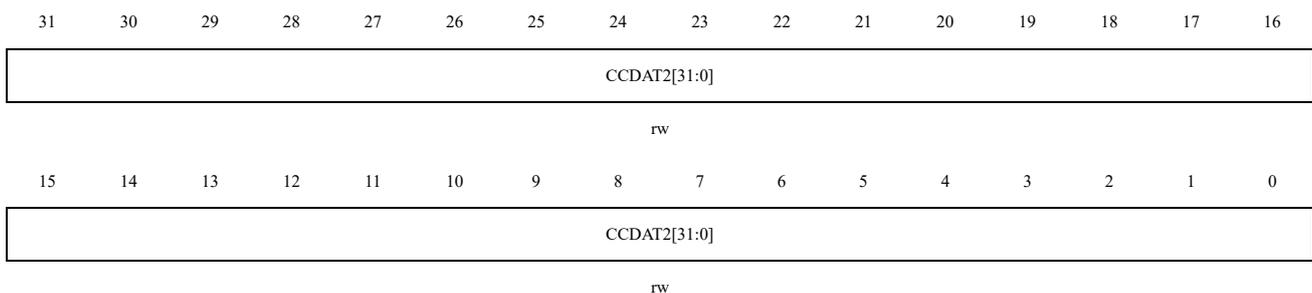
rw

位域	名称	描述
31:0	CCDAT1[31:0]	捕获/比较通道1的值 (Capture/Compare 1 value) <ul style="list-style-type: none"> <li>■ CC1 通道配置为输出： CCDAT1 包含要与计数器 TIMx_CNT 比较的值，在 OC1 输出上发出信号。 如果未在 TIMx_CCMOD1.OC1PEN 位中选择预加载功能，则写入的值会立即传输到有效寄存器。 否则，仅当更新事件发生时，此预加载值才会传输到活动寄存器。</li> <li>■ CC1 通道配置为输入： CCDAT1 包含由最后一个输入捕获 1 事件 (IC1) 传输的计数器值。 当配置为输入模式时，寄存器 CCDAT1 和 CCDDAT1 只能读取。 当配置为输出模式时，寄存器 CCDAT1 和 CCDDAT1 是可读写的。</li> </ul>

### 14.4.12 捕获/比较寄存器 2 (TIMx\_CCDAT2)

偏移地址：0x2C

复位值：0x0000 0000

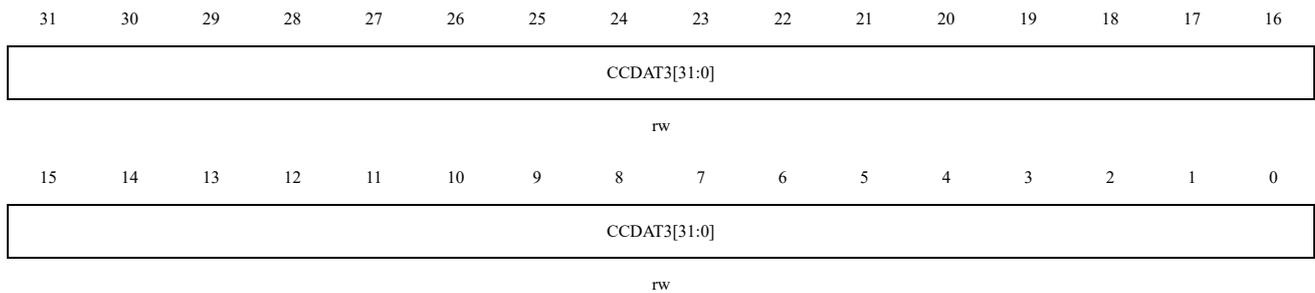


位域	名称	描述
31:0	CCDAT2[31:0]	捕获/比较通道2的值 (Capture/Compare 2 value) <ul style="list-style-type: none"> <li>■ CC2 通道配置为输出： CCDAT2 包含要与计数器 TIMx_CNT 比较的值，在 OC2 输出上发出信号。 如果未在 TIMx_CCMOD1.OC2PEN 位中选择预加载功能，则写入的值会立即传输到有效寄存器。 否则，仅当更新事件发生时，此预加载值才会传输到活动寄存器。</li> <li>■ CC2 通道配置为输入： CCDAT2 包含由最后一个输入捕获 2 事件 (IC2) 传输的计数器值。 当配置为输入模式时，寄存器 CCDAT2 和 CCDDAT2 只能读取。 当配置为输出模式时，寄存器 CCDAT2 和 CCDDAT2 是可读写的。</li> </ul>

### 14.4.13 捕获/比较寄存器 3 (TIMx\_CCDAT3)

偏移地址：0x30

复位值：0x0000 0000

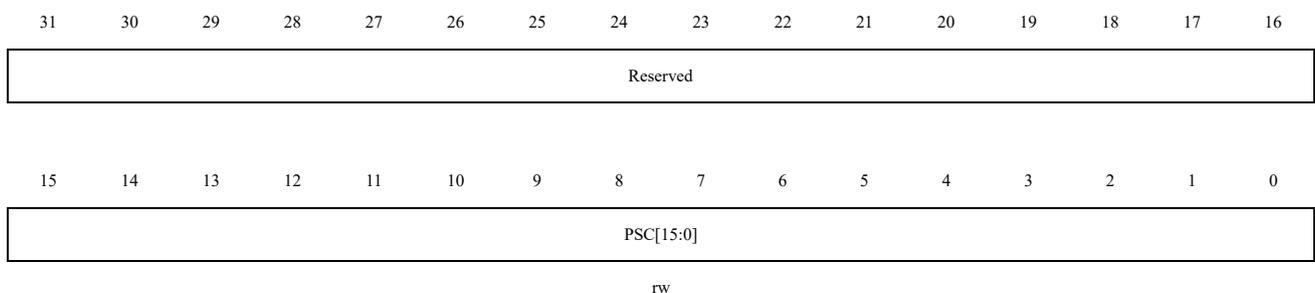


位域	名称	描述
31:0	CCDAT3[31:0]	捕获/比较通道3的值（Capture/Compare 3 value） <ul style="list-style-type: none"> <li>■ CC3 通道配置为输出： CCDAT3 包含要与计数器 TIMx_CNT 比较的值，在 OC3 输出上发出信号。 如果未在 TIMx_CCMOD2.OC3PEN 位中选择预加载功能，则写入的值会立即传输到有效寄存器。否则，仅当更新事件发生时，此预加载值才会传输到活动寄存器。</li> <li>■ CC3 通道配置为输入： CCDAT3 包含由最后一个输入捕获 3 事件 (IC3) 传输的计数器值。 当配置为输入模式时，寄存器 CCDAT3 和 CCDDAT3 只能读取。 当配置为输出模式时，寄存器 CCDAT3 和 CCDDAT3 是可读写的。</li> </ul>

#### 14.4.14 预分频器（TIMx\_PSC）

偏移地址：0x40

复位值：0x0000 0000

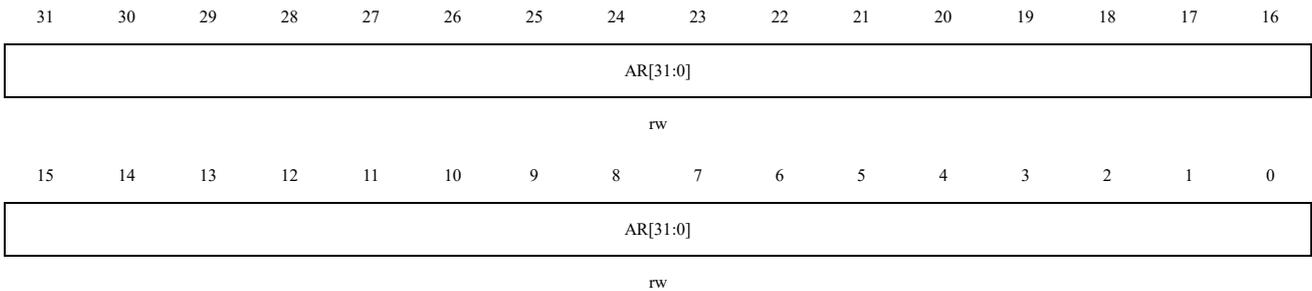


位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:0	PSC[15:0]	预分频器的值（Prescaler value） 计数器时钟 $f_{CK\_CNT} = f_{CK\_PSC} / (PSC [15:0] + 1)$ 。 每次发生更新事件时，PSC 值都会加载到预分频器的影子寄存器中。

#### 14.4.15 自动重装载寄存器（TIMx\_AR）

偏移地址:0x44

复位值: 0xFFFF FFFF

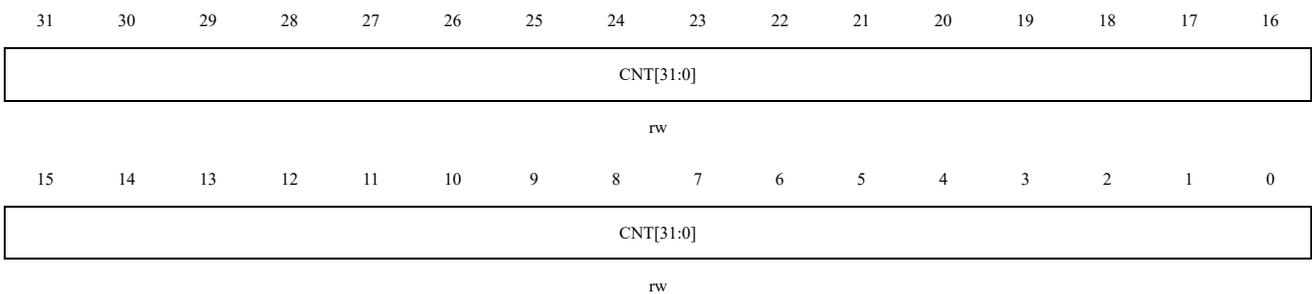


位域	名称	描述
31:0	AR[31:0]	自动重载的值（Auto-reload value） AR包含了将要装载入实际的自动重载寄存器的值。详细参考12.3.1节：有关AR的更新和动作。 当自动重载的值为空时，计数器不工作。

### 14.4.16 计数器（TIMx\_CNT）

偏移地址：0x48

复位值：0x0000 0000



位域	名称	描述
31:0	CNT[31:0]	计数器的值（Counter value）

### 14.4.17 通道 1 滤波寄存器（TIMx\_C1FILT）

偏移地址：0x64

复位值：0x0000 0000



位域	名称	描述
31:30	Reserved	保留, 必须保持复位值
29:24	THRESH[5:0]	采样逻辑电平有效的阈值数(Threshold), 最大 63: 有效逻辑电平的阈值。在采样窗口内, 如果逻辑高的数量大于或等于阈值, 则下一个逻辑电平将为逻辑高。同样的规则适用于逻辑低。如果窗口内 1 和 0 的数量都小于阈值, 则过滤器输出保持不变。阈值应设置为大于或等于 Window 值的一半。 推荐阈值范围为: <b>最小值:</b> 比最大毛刺大小的上限 (预分频时钟周期) 多 1 个预分频时钟周期, 并且需要大于窗口大小的一半。 例如, 如果毛刺大小为 $3.2 * (\text{预分频时钟周期})$ , 则阈值应为 $\lceil 3.2 \rceil = 4 + 1 = 5$ <b>最大值:</b> 有效信号最小尺寸的底值 (在预分频时钟周期内), 需要小于窗口尺寸。 例如, 如果最小信号大小为 $3.2 * (\text{预分频时钟周期})$ , 则阈值应为下限 $(3.2) = 3$ 。
23	Reserved	保留, 必须保持复位值
22:17	WSIZE[5:0]	逻辑电平检查的窗口大小值 (Window size), 最大 63: 窗口大小决定了在获得下一个逻辑级别时将考虑多少采样值。内置 FIFO 为 64 位, 最大索引为 63, 只能将窗口大小设置为 63。
16	FILTEN	滤波器使能 (Filter enable) : 0: 滤波器禁能 1: 滤波器使能
15:0	Reserved	保留, 必须保持复位值

### 14.4.18 通道 2 滤波寄存器 (TIMx\_C2FILT)

偏移地址: 0x68

复位值: 0x0000 0000

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

位域	名称	描述
31:30	Reserved	保留, 必须保持复位值
29:24	THRESH[5:0]	采样逻辑电平有效的阈值数(Threshold), 最大 63: 有效逻辑电平的阈值。在采样窗口内, 如果逻辑高的数量大于或等于阈值, 则下一个逻辑电平将为逻辑高。同样的规则适用于逻辑低。如果窗口内 1 和 0 的数量都小于阈

		值，则过滤器输出保持不变。 阈值应设置为大于或等于 Window 值的一半。 推荐阈值范围为： <b>最小值：</b> 比最大毛刺大小的上限（预分频时钟周期）多 1 个预分频时钟周期，并且需要大于窗口大小的一半。 例如，如果毛刺大小为 3.2*（预分频时钟周期），则阈值应为 $\lceil 3.2 \rceil = 4 + 1 = 5$ <b>最大值：</b> 有效信号最小尺寸的底值（在预分频时钟周期内），需要小于窗口尺寸。 例如，如果最小信号大小为 3.2*（预分频时钟周期），则阈值应为下限 $(3.2) = 3$ 。
23	Reserved	保留，必须保持复位值
22:17	WSIZE[5:0]	逻辑电平检查的窗口大小值（Window size），最大 63： 窗口大小决定了在获得下一个逻辑级别时将考虑多少采样值。 内置 FIFO 为 64 位，最大索引为 63，只能将窗口大小设置为 63。
16	FILTEN	滤波器使能（Filter enable）： 0: 滤波器禁能 1: 滤波器使能
15:0	Reserved	保留，必须保持复位值

### 14.4.19 通道 3 滤波寄存器（TIMx\_C3FILT）

偏移地址：0x6C

复位值：0x0000 0000

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

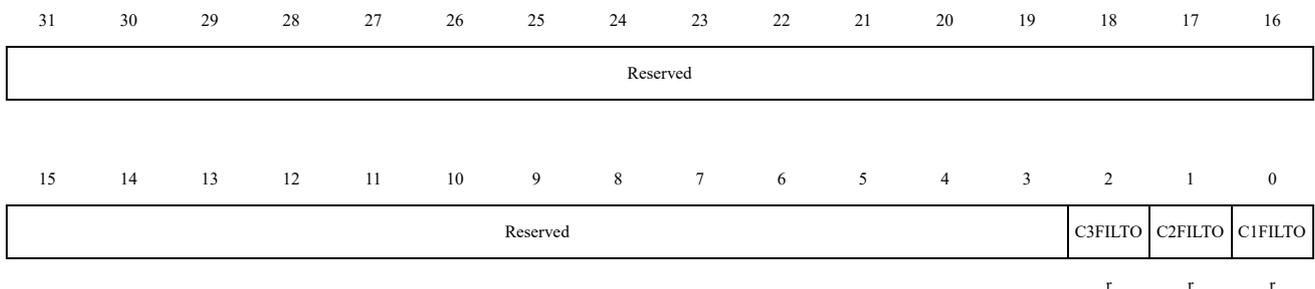
位域	名称	描述
31:30	Reserved	保留，必须保持复位值
29:24	THRESH[5:0]	采样逻辑电平有效的阈值数(Threshold)，最大 63： 有效逻辑电平的阈值。 在采样窗口内，如果逻辑高的数量大于或等于阈值，则下一个逻辑电平将为逻辑高。 同样的规则适用于逻辑低。 如果窗口内 1 和 0 的数量都小于阈值，则过滤器输出保持不变。 阈值应设置为大于或等于 Window 值的一半。 推荐阈值范围为： <b>最小值：</b> 比最大毛刺大小的上限（预分频时钟周期）多 1 个预分频时钟周期，并且需要大于窗口大小的一半。 例如，如果毛刺大小为 3.2*（预分频时钟周期），则阈值应为 $\lceil 3.2 \rceil = 4 + 1 = 5$ <b>最大值：</b> 有效信号最小尺寸的底值（在预分频时钟周期内），需要小于窗口尺寸。 例如，如果最小信号大小为 3.2*（预分频时钟周期），则阈值应为下限 $(3.2) = 3$ 。

23	Reserved	保留，必须保持复位值
22:17	WSIZE[5:0]	逻辑电平检查的窗口大小值（Window size），最大 63： 窗口大小决定了在获得下一个逻辑级别时将考虑多少采样值。 内置 FIFO 为 64 位，最大索引为 63，只能将窗口大小设置为 63。
16	FILTEN	滤波器使能（Filter enable）： 0: 滤波器禁能 1: 滤波器使能
15:0	Reserved	保留，必须保持复位值

### 14.4.20 输入通道滤波输出寄存器（TIMx\_FILTO）

偏移地址：0x74

复位值：0x0000 0000

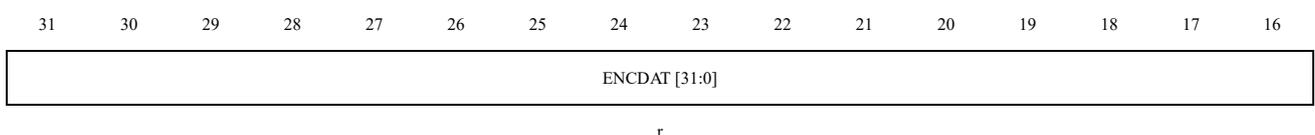


位域	名称	描述
31:3	Reserved	保留，必须保持复位值
2	C3FILTO	通道3滤波输出状态 0: 输出低电平； 1: 输出高电平；
1	C2FILTO	通道2滤波输出状态 0: 输出低电平； 1: 输出高电平；
0	C1FILTO	通道1滤波输出状态 0: 输出低电平； 1: 输出高电平；

### 14.4.21 编码器捕获数据寄存器（TIMx\_ENC DAT）

偏移地址：0x88

复位值：0x0000 0000



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

ENC DAT [31:0]
----------------

r

位域	名称	描述
31:0	ENC DAT[31:0]	编码器捕获数据寄存器。 当检测到TI3事件的有效边沿（当TIMx_ENCMCTRL.C3LDCNTSEL[1:0]=2'bx0时，上升沿有效；当TIMx_ENCMCTRL.C3LDCNTSEL[1:0]=2'bx1时，下降沿有效）时，该寄存器将记录当前计数器值与LVR值之间差值的绝对值，即TIMx_ENC DAT.ENC DAT= TIMx_CNT.CNT-TIMx_ENCLVR.LVR 。

### 14.4.22 编码器模式控制寄存器（TIMx\_ENCMCTRL）

偏移地址：0x8C

复位值：0x0000 0000

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

ENC DATS	Reserved
----------	----------

r

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

Reserved	ENCMD[3:0]	C3LDCNTSEL[1:0]	C3LDCNT EN
----------	------------	-----------------	------------

rw

rw

rw

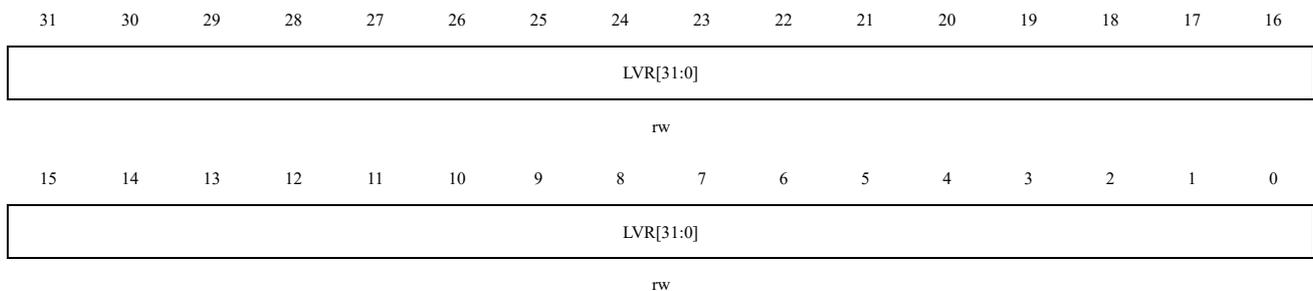
位域	名称	描述
31	ENC DATS	编码器捕获数据寄存器的符号。 当检测到TI3事件的有效边沿（当TIMx_ENCMCTRL.C3LDCNTSEL[1:0]=2'bx0时，上升沿有效；当TIMx_ENCMCTRL.C3LDCNTSEL[1:0]=2'bx1时，下降沿有效）时，该寄存器将记录当前计数器值与LVR值之间差值的符号，即 0：TIMx_ENC DAT.ENC DAT为正值，即TIMx_CNT.CNT-TIMx_ENCLVR.LVR 大于或等于0。 1：TIMx_ENC DAT.ENC DAT为负值，即TIMx_CNT.CNT-TIMx_ENCLVR.LVR 小于0。
30:7	Reserved	保留，必须保持复位值
6:3	ENCMD[3:0]	编码器模式选择。 0000：无 0001：正交编码器模式1 – 根据TI2FP2的电平，计数器在TI1FP1的边沿向上/下计数。 0010：正交编码器模式2 – 根据TI1FP1的电平，计数器在TI2FP2的边沿向上/下计数。 0011：正交编码器模式3 – 根据另一个信号的输入电平，计数器在TI1FP1和TI2FP2的边沿向上/下计数。 0100：正交编码器模式4 – 根据TI2FP2的电平，计数器在TI1FP1的边沿向上/下计数。通过CC1P选择计数边沿。 0101：正交编码器模式5 – 根据TI1FP1的电平，计数器在TI2FP2的边沿向上/下计数。通过CC2P选择计数边沿。

位域	名称	描述
		0110: 脉冲电平编码模式1。通过CC2P设置TI2FP2的计数边沿。 0111: 脉冲电平编码模式2。 1000: CCW/CW编码模式1, 计数器在TI1FP1的边沿向上/下计数。 1001: CCW/CW编码模式2 1010: 双脉冲编码模式1。通过CC1P和CC2P设置TI1FP1和TI2FP2的计数敏感边沿。 1011: 双脉冲编码模式2。 1100~1111: 保留 <i>注: 这些bit位只能在TIMx_SMCTRL.SMSEL[3:0]=4'b0000时使用。</i> <i>注: 这些编码器模式和TIMx_SMCTRL.SMSEL[3:0]中的编码器模式效果相同, 例如ENCMD[3:0]=4'b0001和TIMx_SMCTRL.SMSEL[3:0]=4'b0001都是正交编码器模式1。</i>
2:1	C3LDCNTSEL[1:0]	TI3有效事件选择。 00: TI3事件高有效 01: TI3事件低有效 10: TI3事件上升沿有效 11: TI3事件下降沿有效
0	C3LDCNTEN	TI3有效事件触发计数器装载为LVR值使能。 0: TI3有效事件不能触发计数器装载为LVR值 1: TI3有效事件能触发计数器装载为LVR值

### 14.4.23 外部事件计数器装载值寄存器 (TIMx\_ENCLVR)

偏移地址: 0x90

复位值: 0x0000 0000

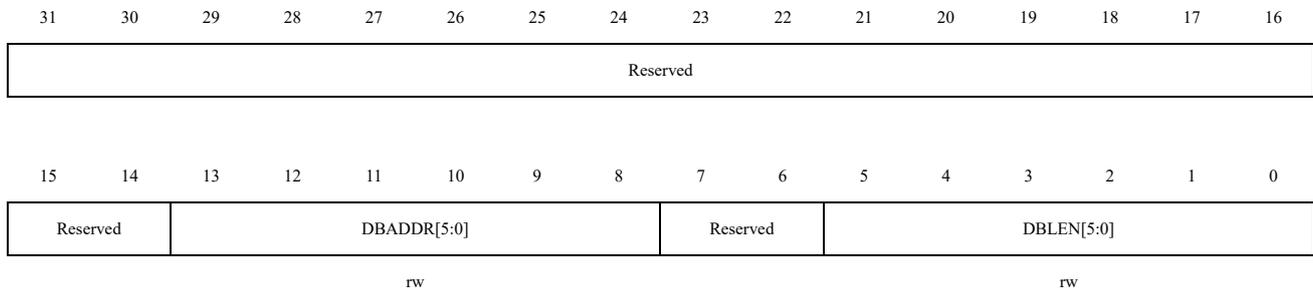


位域	名称	描述
31:0	LVR[31:0]	外部事件 (TI3) 触发计数器装载值。 当检测到TI3事件有效 (可选择高电平有效、低电平有效、上升沿有效或者下降沿有效, 取决于TIMx_ENCCTRL.C3LDCNTSEL[1:0]寄存器) 时, 计数器将装载为LVR值。

## 14.4.24 DMA 控制寄存器 (TIMx\_DCTRL)

偏移地址: 0x94

复位值: 0x0000 0000

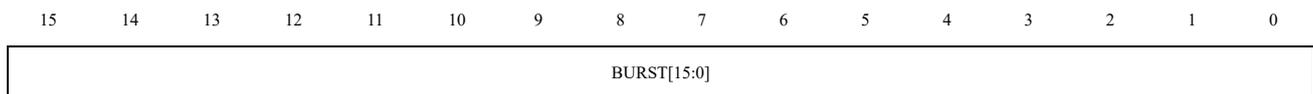
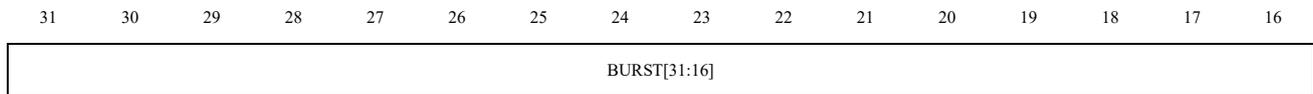


位域	名称	描述
31:14	Reserved	保留, 必须保持复位值
13:8	DBADDR[5:0]	DMA基地址 (DMA base address) 该位字段定义 DMA 访问 TIMx_DADDR 寄存器的第一个地址。 当第一次通过 TIMx_DADDR 完成访问时, 该位域指定您刚刚访问的地址。 然后第二次访问TIMx_DADDR, 会访问到“DMA Base Address + 4”的地址  000000: TIMx_CTRL1, 000001: TIMx_CTRL2, 000010: TIMx_STS, ..... 100101: TIMx_DCTRL
7:6	Reserved	保留, 必须保持复位值
5:0	DBLEN[5:0]	DMA连续传送长度 (DMA burst length) 该位字段定义 DMA 将访问 (写入/读取) TIMx_DADDR 寄存器的次数。  000000: 1次传输 000001: 2次传输 000010: 3次传输 ... 010001: 18次传输 ... 100010: 35次传输

## 14.4.25 连续模式的 DMA 地址 (TIMx\_DADDR)

偏移地址: 0x98

复位值: 0x0000 0000



rw

位域	名称	描述
31:0	BURST[31:0]	<p>DMA 访问缓冲区。</p> <p>当对该寄存器分配读或写操作时，将访问位于地址范围 (DMA base address + DMA burst length × 4) 的寄存器。</p> <p>DMA base address = The address of TIMx_CTRL1 + TIMx_DCTRL.DBADDR * 4;</p> <p>DMA burst len = TIMx_DCTRL.DBLEN + 1.</p> <p>例子:</p> <p>如果 TIMx_DCTRL.DBLEN = 0x3 (4 次传输)，TIMx_DCTRL.DBADDR = 0xD (TIMx_CC DAT1)，DMA 数据长度 = 半字，DMA 存储器地址 = SRAM 中的缓冲区地址，DMA 外设地址 = TIMx_DADDR 地址。</p> <p>当事件发生时，TIMx 将向 DMA 发送请求，并传输 4 次数据。</p> <p>第一次，对 TIMx_DADDR 寄存器的 DMA 访问将映射到访问 TIMx_CC DAT1 寄存器；</p> <p>第二次，对 TIMx_DADDR 寄存器的 DMA 访问将映射到访问 TIMx_CC DAT2 寄存器；</p> <p>.....</p> <p>第四次，对 TIMx_DADDR 寄存器的 DMA 访问将映射到访问 TIMx_CC DAT4 寄存器；</p>

## 15 基本定时器 (TIM6)

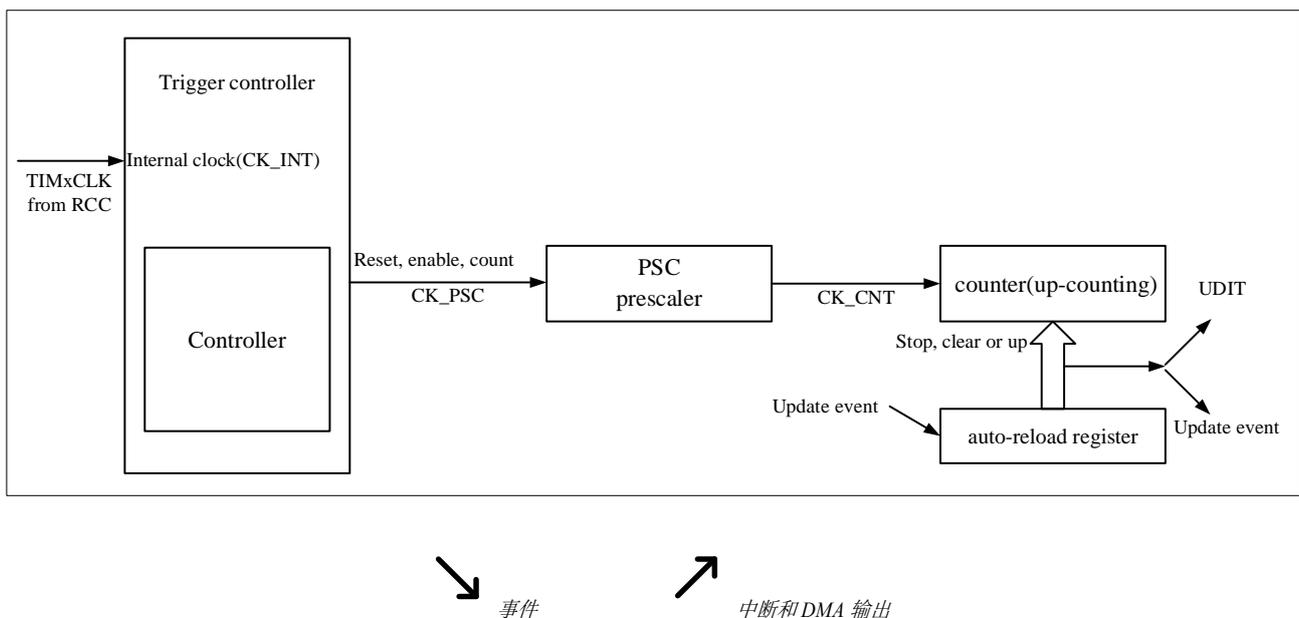
### 15.1 基本定时器简介

基本定时器 TIM6 包含一个 32 位自动装载计数器。

### 15.2 基本定时器主要特性

- 32 位自动重载向上计数计数器。
- 16 位可编程预分频器。(分频系数可配置为 1 到 65536 之间的任意值)
- 产生中断/DMA 的事件如下：
  - ◆ 更新事件
- 支持 STOP 模式唤醒：时钟源配置为 LSI 时，可通过更新中断（联接到 EXTI 9）唤醒 STOP 模式

图 15-1 TIM6 框图



↓ 事件      ↗ 中断和DMA 输出

### 15.3 基础定时器描述

#### 15.3.1 时基单元

时基单元主要包括：预分频器、计数器、自动重载和重复计数器。当时基单元工作时，软件可以随时读写相应的寄存器（TIMx\_PSC、TIMx\_CNT 和 TIMx\_AR）。

*注意：时钟源配置为 LSI 时，TIMx\_CNT 不支持写入。*

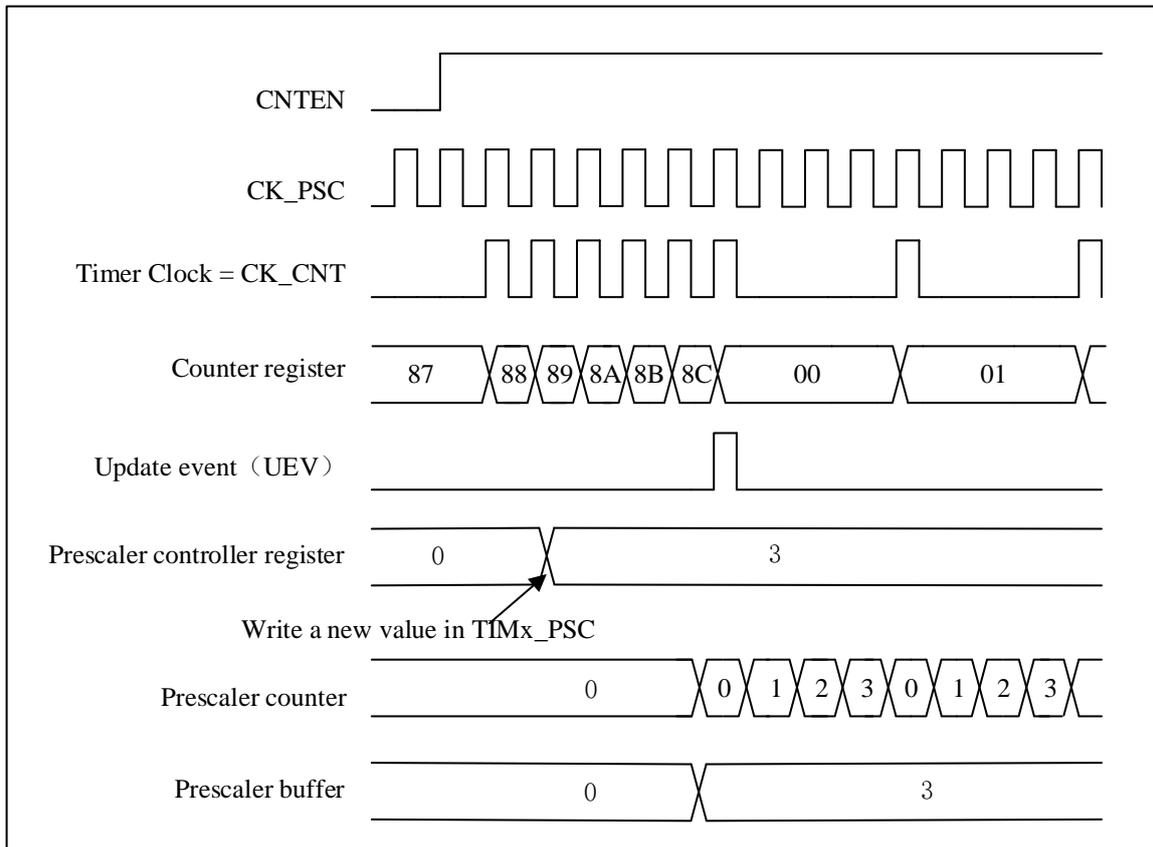
根据自动重载预加载使能位(TIMx\_CTRL1.ARPEN)的设置，预加载寄存器的值会立即或在每次更新事件 UEV 时传输到影子寄存器。TIMx\_CTRL1.UPDIS=0 时，当计数器达到上溢条件或软件设置 TIMx\_EVTGEN.UDGN 位，将生成更新事件。仅当 TIMx\_CTRL1.CNTEN 位置位时，计数器 CK\_CNT 才有

效。计数器在 TIMx\_CTRL1.CNTEN 位置位后一个时钟周期开始计数。

### 15.3.1.1 预分频器描述

TIMx\_PSC 寄存器包含一个 16 位计数器，可用于将计数器时钟频率除以 1 到 65536 之间的任何因子。它可以在缓冲时动态更改。仅在下一次更新事件时才考虑预分频器值。

图 15-2 预分频器分频从 1 到 4 的计数器时序图



## 15.3.2 计数模式

### 15.3.2.1 向上计数模式

在向上计数模式下，计数器会从 0 计数到寄存器 TIMx\_AR 的值，然后复位为 0。并产生计数器溢出事件。

如果设置了 TIMx\_CTRL1.UPRS 位(选择更新请求)和 TIMx\_EVTGEN.UDGN 位，则会生成更新事件(UEV)，并且不会由硬件设置 TIMx\_STS.UDITF。因此，不会产生更新中断或更新 DMA 请求。此设置用于您想要清除计数器但不想产生更新中断的场景。

取决于 TIMx\_CTRL1.UPRS 的配置，当更新事件发生时，TIMx\_STS.UDITF 被设置，所有寄存器都被更新：

- 当 TIMx\_CTRL1.ARPEN =1 时，使用预加载值(TIMx\_AR)更新自动重载影子寄存器。
- 预分频器影子寄存器重新加载预加载值(TIMx\_PSC)。

为避免在将新值写入预加载寄存器时更新影子寄存器，您可以通过设置 TIMx\_CTRL1.UPDIS=1 来禁用更新。

当更新事件发生时，计数器仍将被清零，预分频器计数器也将设置为 0（但预分频器值将保持不变）。

下图显示了向上计数模式下不同除法因子的计数器行为和更新标志的一些示例。

图 15-3 向上计数时序图，内部时钟分频因子 = 2/N

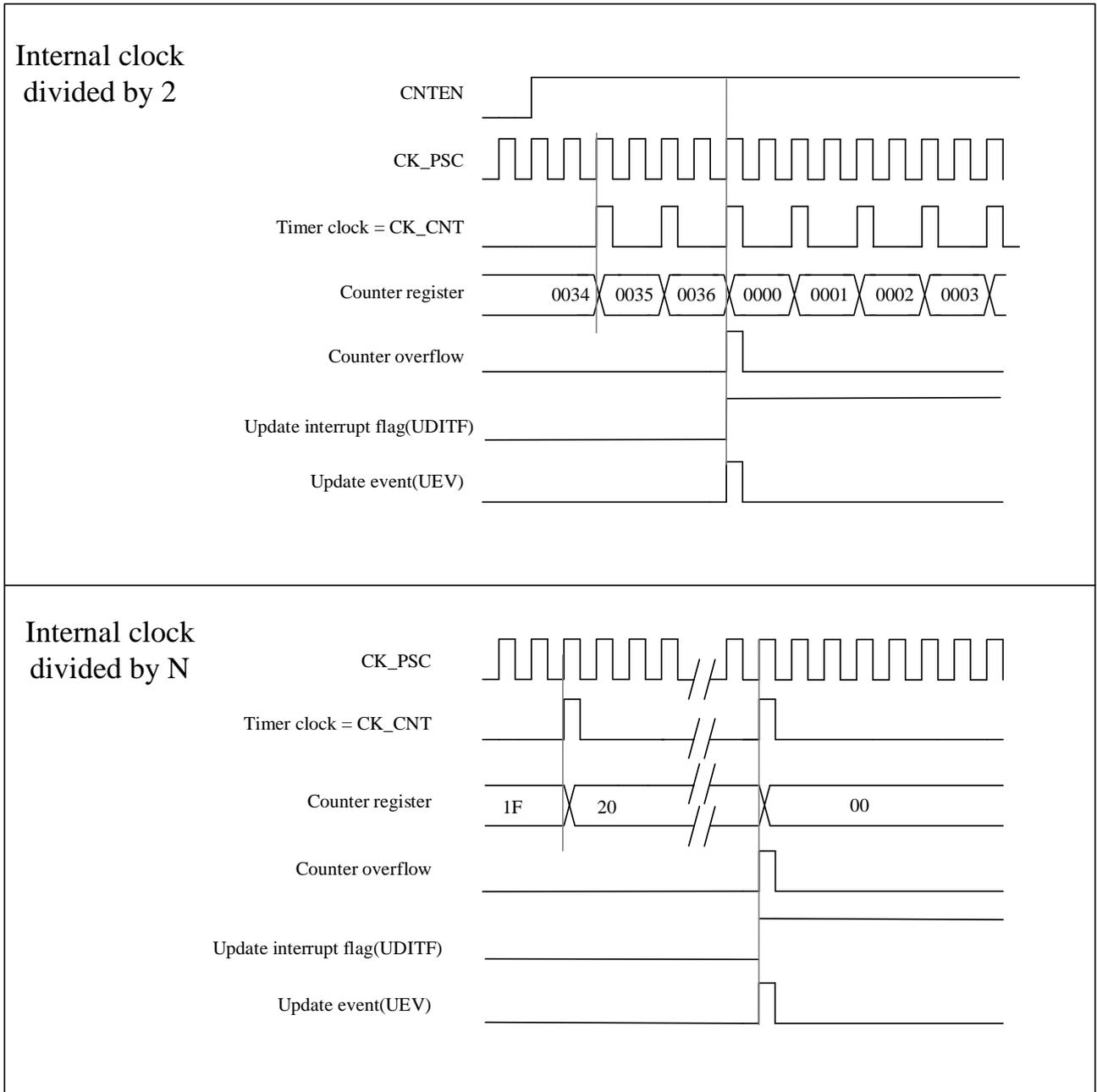
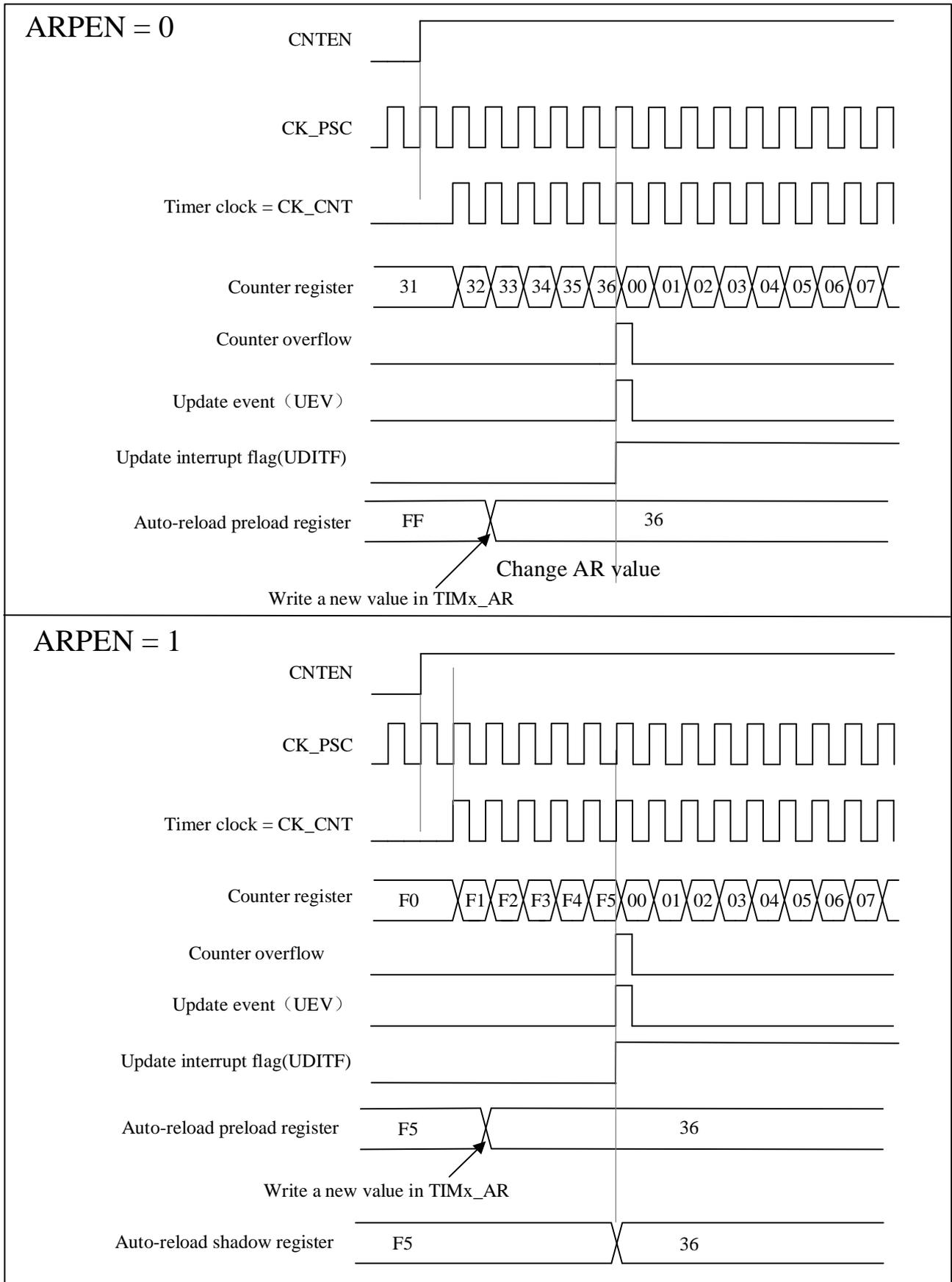


图 15-4 ARPEN=0/1 时向上计数、更新事件的时序图



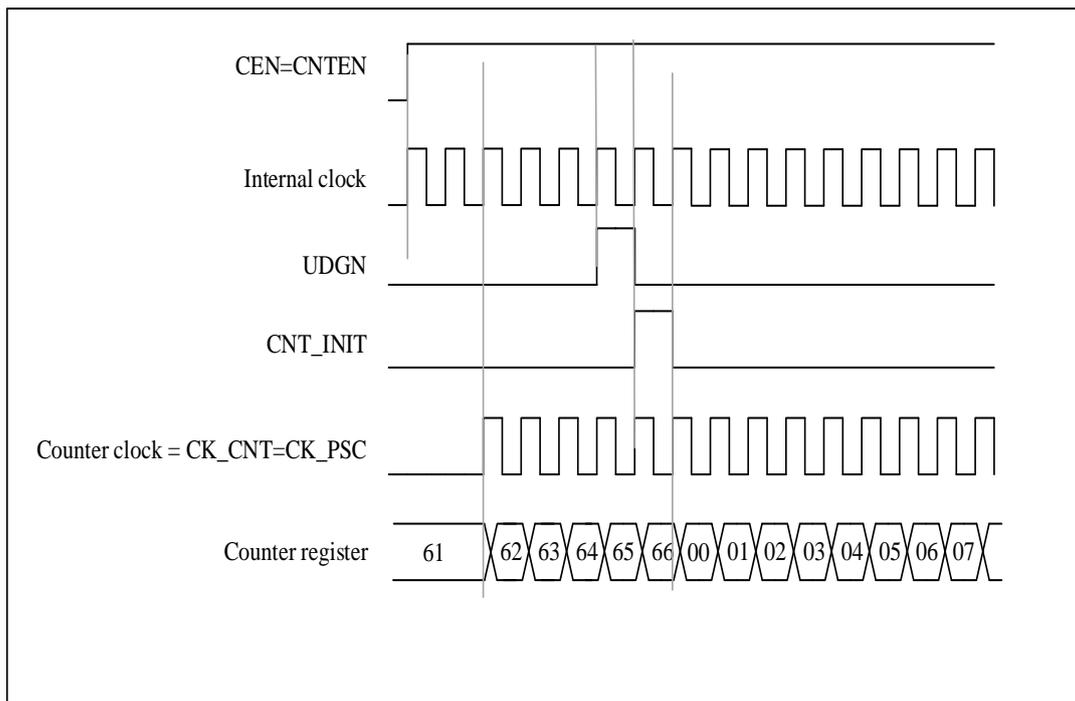
### 15.3.3 时钟选择

■ 定时器内部时钟: CK\_INT

#### 15.3.3.1 内部时钟源 (CK\_INT)

前提是 TIMx\_CTRL1.CNTEN 位由软件写为'1', 预分频器的时钟源由内部时钟 CK\_INT 提供。

图 15-5 正常模式下的控制电路, 内部时钟分频系数为 1



### 15.3.4 调试模式

当微控制器处于调试模式 (Cortex-M0 内核停止) 时, 根据 DBG\_CTRL.TIM6STP 位配置, TIM6 计数器可以继续正常工作或停止。有关详细信息, 请参阅 3.3.1 节。

## 15.4 TIM6 寄存器描述

有关寄存器中使用的缩写, 请参阅第 1.1 节

这些外设寄存器可以作为半字 (16 位) 或一个字 (32 位) 操作。

### 15.4.1 寄存器总览

表 15-1 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000h	TIM_CTRL1	Reserve														ARPE	ONEPI	Reserve	UPDIS	UPRS	Reserve	CNTEEN											

008h	TIM_STS	Reserve	UDITF	Reserve		
00Ch	TIM_EVTGEN	Reserve			UDGN	Reserve
014h	TIM_DINTEN	Reserve	UDEN	Reserve	UIEN	Reserve
040h	TIM_PSC	Reserve			PSC[15:0]	
044h	TIM_AR	AR[31:0]				
048h	TIM_CNT	CNT[31:0]				

## 15.4.2 控制寄存器 1 (TIMx\_CTRL1)

地址偏移: 0x00

复位值: 0x0000 0000

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

位域	名称	描述
31:10	Reserved	保留，必须保持复位值
9	ARPEN	自动重载预装载允许位（Auto-reload preload enable） 0: TIMx_AR 寄存器的影子寄存器禁用 1: TIMx_AR 寄存器的影子寄存器使能
8	ONEPM	单脉冲模式（One pulse mode） 0: 禁用单脉冲模式，发生更新事件时不影响计数器计数。 1: 使能单脉冲模式，计数器在下次更新事件发生时停止计数（清 TIMx_CTRL1.CNTEN 位）。

7:6	Reserved	保留，必须保持复位值
5	UPDIS	禁止更新（Update disable） 该位用于启用/禁用软件生成的更新事件（UEV）事件。 0：启用UEV。如果满足以下条件之一，将生成UEV： – 计数器溢出 – TIMx_EVTGEN.UDGN 位被设置 影子寄存器将使用预加载值进行更新。 1：UEV禁用。不生成更新事件，影子寄存器（AR、PSC）保持其值。如果设置了TIMx_EVTGEN.UDGN位，则重新初始化计数器和预分频器。
4	UPRS	更新请求源（Update request source） 该位用于通过软件选择 UEV 事件源。 0：如果更新中断或 DMA 请求使能，以下任何事件都会产生更新中断或 DMA 请求： – 计数器溢出 – TIMx_EVTGEN.UDGN 位被设置 1：如果更新中断或 DMA 请求使能，只有计数器溢出会产生更新中断或 DMA 请求
3:1	Reserved	保留，必须保持复位值
0	CNTEN	使能计数器（Counter enable） 0：禁止计数器； 1：使能计数器。

### 15.4.3 状态寄存器 (TIMx\_STS)

地址偏移: 0x08

复位值: 0x0000 0000

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

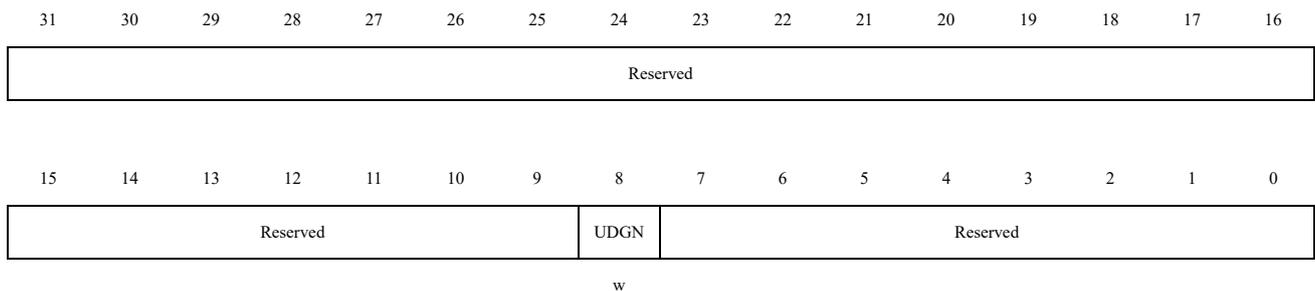
位域	名称	描述
31:17	Reserved	保留，必须保持复位值
16	UDITF	更新中断标志（Update interrupt flag） 当在以下条件下发生更新事件时，该位由硬件设置： – 当 TIMx_CTRL1.UPDIS = 0 且计数器值溢出时。 – 当 TIMx_CTRL1.UPRS = 0 时，TIMx_CTRL1.UPDIS = 0，并通过软件设置TIMx_EVTGEN.UDGN 位以重新初始化 CNT。

		该位由软件清零。 0: 未发生更新事件 1: 发生更新中断
15:0	Reserved	保留, 必须保持复位值

### 15.4.4 事件产生寄存器 (TIMx\_EVTGEN)

地址偏移: 0x0C

复位值: 0x0000 0000

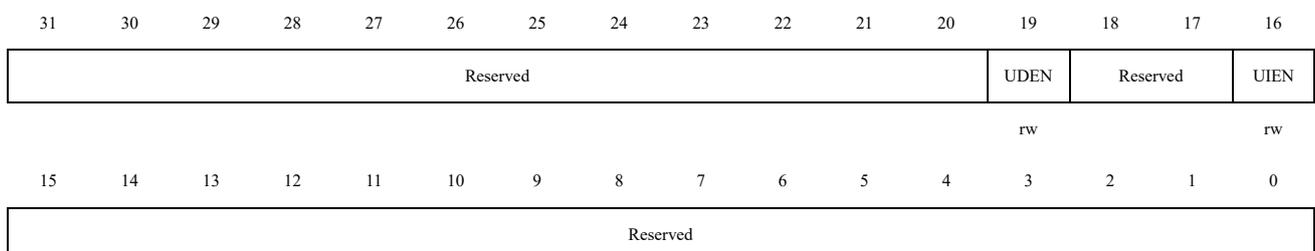


位域	名称	描述
31:9	Reserved	保留, 必须保持复位值
8	UDGN	产生更新事件 (Update generation) 软件可以设置该位来更新配置寄存器的值, 硬件会自动清除它。 0: 无效果。 1: 定时器计数器将重新启动, 所有影子寄存器将被更新。 它也将重新启动预分频器计数器。
7:0	Reserved	保留, 必须保持复位值

### 15.4.5 DMA/中断使能寄存器 (TIMx\_DINTEN)

地址偏移: 0x14

复位值: 0x0000 0000

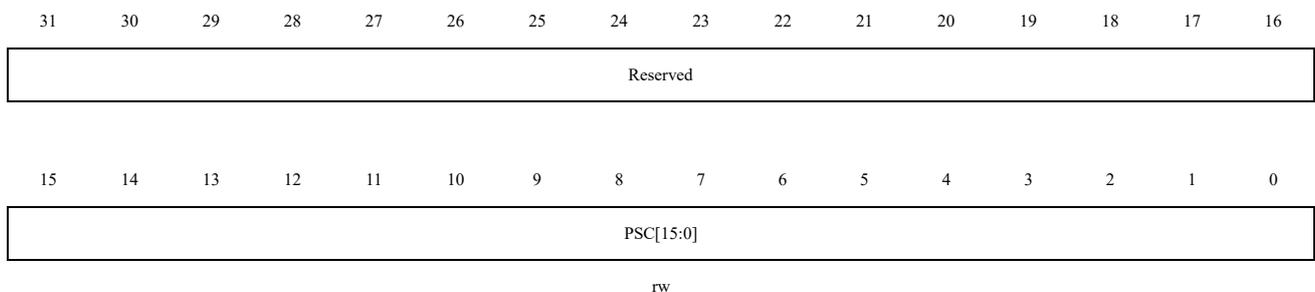


位域	名称	描述
31:20	Reserved	保留，必须保持复位值
19	UDEN	更新DMA请求使能（Update DMA request enable） 0：禁止更新DMA请求 1：使能更新DMA请求
18:17	Reserved	保留，必须保持复位值
16	UIEN	更新中断使能（Update interrupt enable） 0：禁止更新中断 1：使能更新中断
15:0	Reserved	保留，必须保持复位值

### 15.4.6 预分频器 (TIMx\_PSC)

地址偏移: 0x40

复位值: 0x0000 0000



位域	名称	描述
31:16	Reserved	保留，必须保持复位值
15:0	PSC[15:0]	预分频器数值（Prescaler value） PSC寄存器值将在更新事件时更新到预分频器寄存器。计数器时钟频率是输入时钟分频 PSC+1。

### 15.4.7 自动重载寄存器 (TIMx\_AR)

地址偏移: 0x44

复位值: 0x0000 FFFF



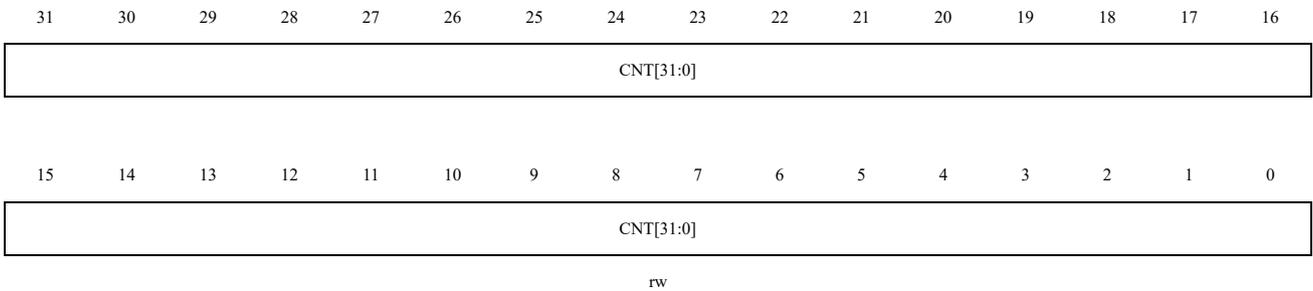
rw

位域	名称	描述
31:0	AR[31:0]	自动重载数值 (Auto-reload value) 这些位定义将加载到实际自动重载寄存器中的值。 有关详细信息, 请参阅 15.3.1。 当TIMx_AR.AR [31:0]值为空时, 计数器不工作。

### 15.4.8 计数器 (TIMx\_CNT)

地址偏移: 0x48

复位值: 0x0000 0000



位域	名称	描述
31:0	CNT[31:0]	计数器数值 (Counter value)

## 16 模拟/数字转换(ADC)

### 16.1 简述

12 位 ADC 是使用逐次逼近的高速模数转换器。共有 16 个通道，可测 11 个外部和 5 个内部信号源。各个通道的 A/D 转换通道可以在单次、连续或段模式下执行。ADC 转换值存储（左对齐/右对齐）在 16 位数据寄存器中。可以通过模拟看门狗检测输入电压是否在用户定义的高/低阈值内，并且 ADC 的输入时钟的最大频率为 32MHz。

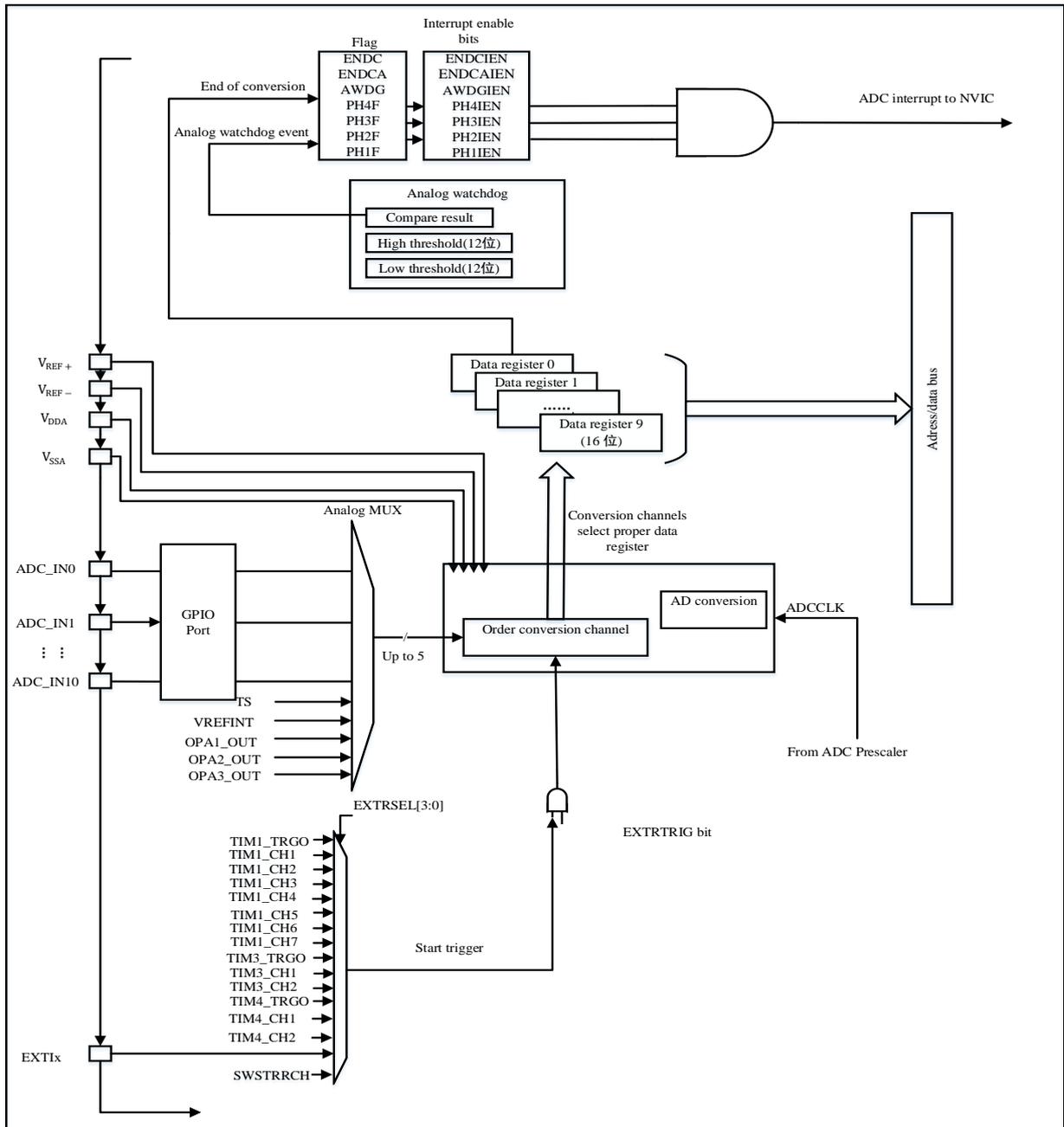
### 16.2 ADC 主要特征

- 仅支持 12 位分辨率
- 仅支持单端输入
- 转换结束、发生模拟看门狗事件时产生中断
- 单次和连续转换模式
- 10 个结果寄存器，通道号可配
- 16（11 外部 + 5 内部）个通道设置，采样时间单独可配
- 单段、双段、三段、四段自定义采样序列采样，序列次数和通道号可灵活配置
- 转换完成中断，每个触发源转换完成后，自动产生转换完成标志，硬件置位软件清零；打开中断使能产生中断，包括段中断，任意通道完成中断，全部转换完成中断和看门狗中断
- 带内嵌数据一致性的数据对齐
- 规则转换有内部触发和外部触发选项
- 支持最大 1MSPS
- 支持 ADC 联动：支持可选 Trigger 源 16 个（包括 TIM Trigger 源 14 个，EXTI 和软件触发），触发后可以产生可配长度和指定通道的转换
- ADC 供电要求：2.4V 到 5.5V
- ADC 输入范围： $0 \leq V_{IN} \leq V_{DDA}$

### 16.3 ADC 功能描述

下图为一个 ADC 模块的框图，表 16-1 为 ADC 引脚的说明。

图 16-1 ADC 框图



**表 16-1 ADC 引脚**

名称	信号类型	注解
VREF+	输入, 模拟参考正极	ADC使用的高端/正极参考电压, $2.4V \leq V_{REF+} \leq V_{DDA}$
VDDA <sup>(1)</sup>	输入, 模拟电源	等效于V <sub>DD</sub> 的模拟电源且: $2.4V \leq V_{DDA} \leq V_{DD}(5.5V)$
VREF-	输入, 模拟参考负极	ADC使用的低端/负极参考电压, $V_{REF-} = V_{SSA}$
VSSA <sup>(1)</sup>	输入, 模拟电源地	等效于VSS的模拟电源地
ADC_IN[10:0]	模拟输入信号	11个模拟外部输入通道

1. V<sub>DDA</sub> 和 V<sub>SSA</sub> 应该分别连接到 V<sub>DD</sub> 和 V<sub>SS</sub>。

### 16.3.1 ADC 开关控制

用户必须等待 PowerUp 过程完成才可以进入下一步的操作, 可以通过查询 ADC\_STS 里面的 RDY 确认是否上电完成。

通过设置 ADC\_CTRL2 寄存器的 ON 位可给 ADC 上电。当第一次设置 ON 位时, 它将 ADC 从断电状态下唤醒。ADC 上电延迟一段时间后, 当接收到触发信号后进行 ADC 转换。

通过清除 ON 位可以停止转换, 并将 ADC 置于断电模式。在这个模式中, ADC 几乎不耗电 (仅几个  $\mu A$ )。用户可以通过查询 ADC\_STS 里面的 PDRDY 确认是否下电完成。在 ADC Disable 的时候默认都是 PowerDown 模式。

### 16.3.2 ADC 时钟

ADC 需要三个时钟, ADC\_CLK, HCLK, ADC\_1MCLK。

- HCLK 用于寄存器的访问时钟。
- ADC\_CLK 为 ADC 的工作时钟, 最大支持 32M。
- ADC\_1MCLK 用于内部计时功能, 在 RCC 中配置, 频率大小必须配置成 1M

注:

1 配置 ADC\_CLK (源自 SYS\_CLK) 作为工作时钟最高可到 32M, 支持分频 2,4,8,16,32

### 16.3.3 通道选择

ADC 支持 16 路通道输入, 11 路外部通道, 5 路内部通道, 连接关系如下表:

**表 16-2 通道对应表**

Channel	Source
CH0	PA0
CH1	PA1
CH2	PA2
CH3	PA3

CH4	PA4
CH5	PA5
CH6	PA6
CH7	PA7
CH8	PB0
CH9	PB1
CH10	PB2
CH11	Temp Seneor
CH12	VREFINT
CH13	OPA1.OUT
CH14	OPA2.OUT
CH15	OPA3.OUT

注:

1. 测试 TS 需要打开寄存器 `ADC_CTRL3.TEMPEN = 1'b1`;
2. 测试内部参考电压 VREFINT 需要打开寄存器 `ADC_CTRL3.REFSEL = 1'b1`, 等待 `ADC_STS.VREFRDY = 1` 时, 再进行 adc 转换。
3. 测试 OPAMP1 的输出需要打开对应的 `OPAMP1_CS.MOD[1:0] = 2'b01`;
4. 测试 OPAMP2 的输出需要打开对应的 `OPAMP2_CS.MOD = 1'b0`;
5. 测试 OPAMP3 的输出需要打开对应的 `OPAMP3_CS.MOD = 1'b0`;

### 16.3.4 单次转换模式

ADC 可以通过配置 `ADC_CTRL2.CTU` 为 0 进入单次转换模式;

当转换一个通道时, 转换完成时, 任意通道转换结束标志 (`ADC_STS.ENDCA`) 将被置 1。如果任意规则通道转换结束中断使能 (`ADC_CTRL1.ENDCAIEN`) 位被置 1, 则一个中断将生成, 转换后的数据将存储在 `ADC_DATx` 寄存器中

### 16.3.5 连续转换模式

ADC 可以通过配置 `ADC_CTRL2.CTU` 为 1 进入连续转换模式。在该模式下, 软件或者外部事件都可以启动 ADC 开始转换, ADC 会持续转换选择的通道。

转换开始后, 当一组序列转换完成时, 通道转换结束标志位 (`ADC_STS.ENDC`) 将设置为 1。如果通道转换结束中断使能位 (`ADC_INTEN.ENDCIEN`) 设置为 1, 将产生一个中断。每次转换后的数据将存储在 `ADC_DATx` 寄存器中;

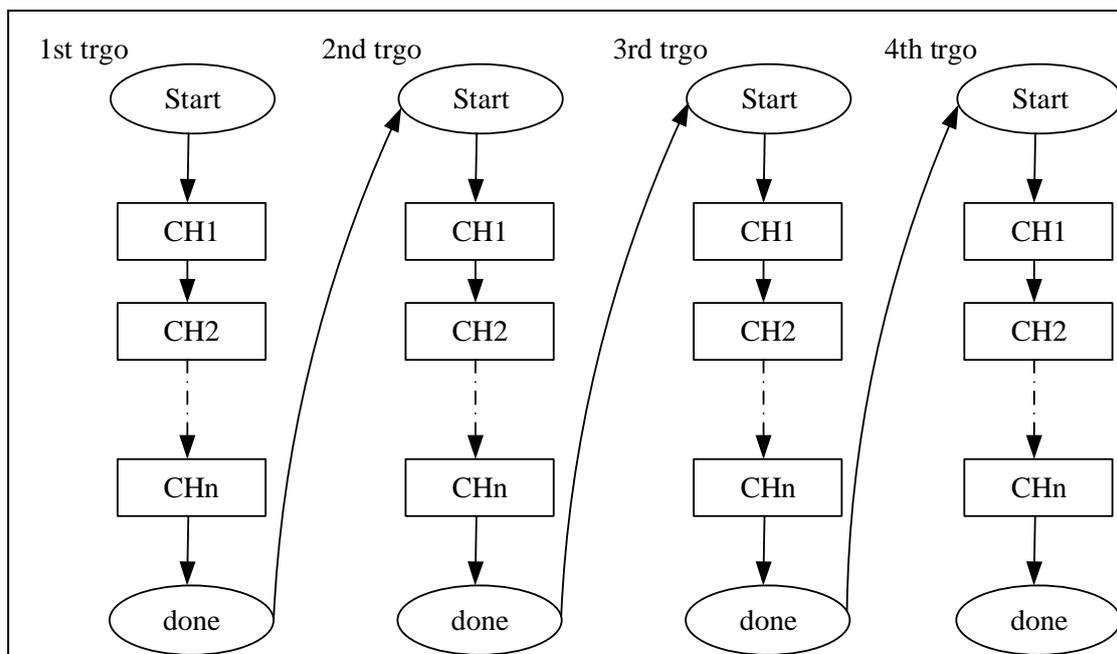
注: 连续转换只适用于第一段转换。

### 16.3.6 段采样模式

ADC 支持一段、两段、三段、四段采样模式，每段采样需要特定的外部事件来触发开始，每段采样支持不同采样次数和采样信号通道配置，段转换为一次性转换完成，不支持循环转换。

- 一段采样是指 ADC 收到一次触发完成一段采样动作，一段采样可能包含多次对模拟信号的采样，次数由分段采样次数寄存器配 ADC\_PHCFG 进行配置，寄存器数值为 0~9 时（最大 9），对应的采样次数为 1~10（不能超过 10 个）。
- 两段触发需要两次触发才能完成完整的一轮采样。第一个触发到达时进行第一段采样，第二个触发到达时进行第二段采样。
- 三段触发需要三次触发才能完成完整的一轮采样。第一个触发到达时进行第一段采样，第二个触发到达时进行第二段采样，第三个触发到达时进行第三段采样。
- 四段触发需要四次触发才能完成完整的一轮采样。第一个触发到达时进行第一段采样，第二个触发到达时进行第二段采样，第三个触发到达时进行第三段采样，第四个触发到达时进行第四段采样。

图 16-2 ADC 段转换图



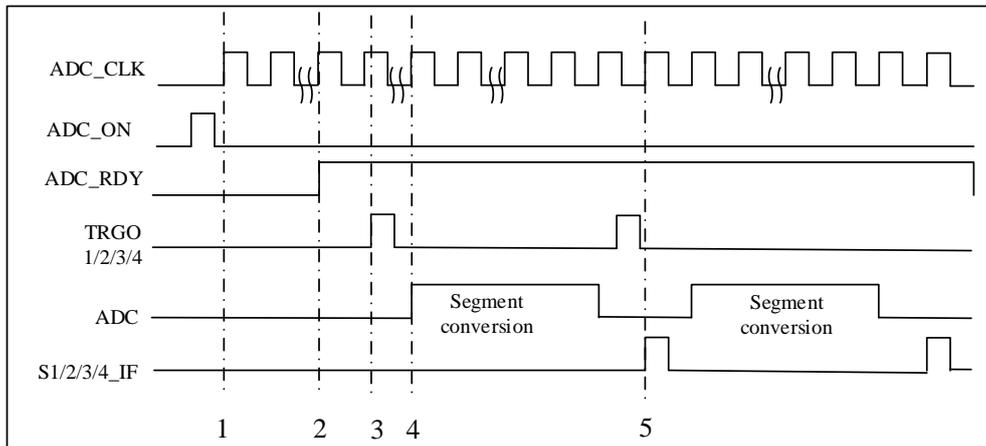
注:

1. 目前的 trgo 分成 4 段各自触发，每段仅需一个 trgo 触发多个通道（寄存器配置）采样，每次最多可接收 4 个段采样 trgo，段转换完成后自动清零。
2. ADC 四段采样之和不能超过寄存器数（10 个数据寄存器），10 个通道对应 10 个寄存器，比如第一段是 1、2、4 通道，第二段是 1、3、4 通道。存放到寄存器的顺序是 1，2，3，4，5，6；
3. 四段之和不要超过 10 个通道，配置超过 10 个通道后，只会转换 10 个通道，多出部分会被直接忽略且产生转换通道的段完成标志。
4. 第一段触发源抢占功能，例如配置:1, 2, 3 段触发，1, 2 段触发先后已经完成，第 3 段触发未发生。然后此时第 1 段触发到来，说明上一轮（1, 2, 3）的采样是不完整的。需要重新触发 1, 2, 3 段进行采样；

### 16.3.7 时序图

如下图所示，设置 ADCCTRL2.ON 为 1，等待 ADC\_STS.RDY 置位，开始 ADC 转换，标志被设置，转换的结果保存在 ADC 数据寄存器中。

图 16-3 ADC 转换



注:

- 1: 打开 ADC 使能, 开始 ADC 上电
- 2: ADC 上电完成
- 3: ADC 触发开始
- 4: ADC 段转换开始
- 5: ADC 段转换完成

### 16.3.8 模拟看门狗

支持 10 通道单独看门狗中断使能，模拟看门狗的阈值与数据对齐的方式无关，因为 ADC 的转换值与阈值的比较是在对齐之前完成。当 ADC 转换的值高于模拟看门狗的高阈值或低于模拟看门狗的低阈值时，则模拟看门狗标志(ADC\_STS.AWDG)将被置为 1。如果开启了相应通道的模拟看门狗，此时会产生中断。

注:

1. 看门狗中断或标志只有在 ADC 转换的时候才会置起，即转换结束后数据寄存器内的值不会产生中断或者标志。

### 16.3.9 数据对齐

ADC\_CTRL2 寄存器中的 ALIG 位选择转换后数据储存的对齐方式。数据可以左对齐或右对齐。

数据右对齐

0	0	0	0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
---	---	---	---	-----	-----	----	----	----	----	----	----	----	----	----	----

数据左对齐

D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0
-----	-----	----	----	----	----	----	----	----	----	----	----	---	---	---	---

### 16.3.10 可编程的通道采样时间

ADC 使用若干个 ADC\_CLK 周期对输入电压采样，采样周期数目可以通过 ADC\_SAMPTx 寄存器中的 SAMPx[3:0]位更改。每个通道可以分别用不同的时间采样。总转换时间如下计算：

$$T_{CONV} = \text{采样时间} + 12 \text{ 个周期}$$

例如：

当 ADCCLK=16MHz，采样时间为 4 周期

$$T_{CONV} = 4 + 12 = 16 \text{ 周期}$$

### 16.3.11 触发转换

此子模块主要实现对外来的 trgo 信号的握手同步，并选择一个当前配置需要的 trgo。

每个采样的信号源通过寄存器配置选定，信号源的选定需在触发前完成，且在一次采样过程完成前不应该改变。

表 16-3 ADC 触发与通道选择

TRIGSEL[3:0]	触发源	类型
0000	TIM1.CC1 事件	来自片上定时器的内部信号
0001	TIM1.CC2 事件	
0010	TIM1.CC3 事件	
0011	TIM1.CC4 事件	
0100	TIM1.CC5 事件	
0101	TIM1.CC6 事件	
0110	TIM1.CC7 事件	
0111	TIM1.TRGO 事件	
1000	TIM3.CC1 事件	
1001	TIM3.CC2 事件	
1010	TIM3.TRGO 事件	
1011	TIM4.CC1 事件	
1100	TIM4.CC2 事件	
1101	TIM4.TRGO 事件	
1110	EXTI 线 0~6 事件	外部引脚/来自片上定时器的内部信号
1111	SWSTRCH	软件控制位

注：当 2 次触发用同一个触发源时，两次触发间隔需要至少 5 个 ADC\_CLK。

## 16.3.12 ADC 中断

支持 7 个中断包括：看门狗中断(AWDG)，所有通道（段）转换结束中断(ENDC)，任一通道转换结束中断(ENDCA)，第一段采样转换完成中断(PH1F)，第二段采样转换完成中断(PH2F)，第三段采样转换完成中断(PH3F)，第四段采样转换完成中断(PH4F)。

单段触发采样完成中断：采样完成时 ADC\_STS.PH1F 置位。

两段触发采样完成中断：第一段采样完成时 ADC\_STS.PH1F 置位，第二段采样完成时 ADC\_STS.PH2F 置位。

三段触发采样完成中断：第一段采样完成时 ADC\_STS.PH1F 置位，第二段采样完成时 ADC\_STS.PH2F 置位，第三段采样完成时 ADC\_STS.PH3F 置位。

四段触发采样完成中断：第一段采样完成时 ADC\_STS.PH1F 置位，第二段采样完成时 ADC\_STS.PH2F 置位，第三段采样完成时 ADC\_STS.PH3F 置位，第四段采样完成时 ADC\_STS.PH4F 置位。

如果使能了相关的中断位，将产生相应的中断。

表 16-4 ADC 中断

中断事件	事件标志	使能控制位
全部通路转换结束使能	ADC_STS.ENDC	ADC_CTRL1.ENDCIEN
设置模拟看门狗状态位使能	ADC_STS.AWDG	ADC_CTRL1.AWDGIEN
任何规则通道中断使能	ADC_STS.ENDCA	ADC_CTRL1.ENDCAIEN
第一段采样转换完成使能	ADC_STS.PH1F	ADC_CTRL1.PH1IEN
第二段采样转换完成使能	ADC_STS.PH2F	ADC_CTRL1.PH2IEN
第三段采样转换完成使能	ADC_STS.PH3F	ADC_CTRL1.PH3IEN
第四段采样转换完成使能	ADC_STS.PH4F	ADC_CTRL1.PH4IEN

## 16.4 ADC 寄存器

### 16.4.1 ADC 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
0x00	ADC_STS	Reserved											PDRDY	RDY	VREFRDY	STR	Reserved	AWDG	ENDCA	ENDC	PH4F	PH3F	PH2F	PH1F															
0x04	ADC_CTRL1	Reserved																										AWDGIEN	ENDCAIEN	ENDCIEN	PH4IEN	PH3IEN	PH2IEN	SIIE					
0x08	ADC_CTRL2	Reserved																																					

0xC	ADC_CTRL3	Reserved										TEMPEN	Reserved	REFSEL	VREFEN	Reserved					
0x10	ADC_SAMPT 1	SAMP7[3:0]	SAMP6[3:0]	SAMP5[3:0]	SAMP4[3:0]	SAMP3[3:0]	SAMP2[3:0]	SAMP1[3:0]	SAMP0[3:0]												
0x14	ADC_SAMPT 2	SAMP15[3:0]	SAMP14[3:0]	SAMP13[3:0]	SAMP12[3:0]	SAMP11[3:0]	SAMP10[3:0]	SAMP9[3:0]	SAMP8[3:0]												
0x18	ADC_AWDHI GH	Reserved						HTH[11:0]													
0x1C	ADC_AWDL OW	Reserved						LTH[11:0]													
0x20	ADC_AWDE N	Reserved						AWDENx[9:0]													
0x24	ADC_PHCFG	Reserved				TRIGMD	PH4CHNUM		PH3CHNUM		PH2CHNUM		PH1CHNUM								
0x28	ADC_TRIGSE L	PH4EXTSEL	PH3EXTSEL	PH2EXTSEL	PH1EXTSEL	SWSTRCH	EXTTRIG	TIM4_TRGOEN	TIM4_CC2EN	TIM4_CC1EN	TIM3_TRGOEN	TIM3_CC2EN	TIM3_CC1EN	TIM1_TRGOEN	TIM1_CC7EN	TIM1_CC6EN	TIM1_CC5EN	TIM1_CC4EN	TIM1_CC3EN	TIM1_CC2EN	TIM1_CC1EN
0x2C	ADC_PHSWT RIG	Reserved				PHSWSTART															
0x30	ADC_SEQ1	SEQ8[3:0]	SEQ7[3:0]	SEQ6[3:0]	SEQ5[3:0]	SEQ4[3:0]	SEQ3[3:0]	SEQ2[3:0]	SEQ1[3:0]												
0x34	ADC_SEQ2	Reserved							SEQ10[3:0]	SEQ9[3:0]											
0x38	ADC_DAT0	Reserved						DAT0[11:0]													
0x3C	ADC_DAT1	Reserved						DAT1[11:0]													

0x40	ADC_DAT2	Reserved	DAT2[11:0]
0x44	ADC_DAT3	Reserved	DAT3[11:0]
0x48	ADC_DAT4	Reserved	DAT4[11:0]
0x4C	ADC_DAT5	Reserved	DAT5[11:0]
0x50	ADC_DAT6	Reserved	DAT6[11:0]
0x54	ADC_DAT7	Reserved	DAT7[11:0]
0x58	ADC_DAT8	Reserved	DAT8[11:0]
0x5C	ADC_DAT9	Reserved	DAT9[11:0]

## 16.4.2 ADC 状态寄存器(ADC\_STS)

地址偏移: 0x00

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				PDRDY	RDY	VREFRD Y	STR	Reserved	AWDG	ENDCA	ENDC	PH4F	PH3F	PH2F	PH1F
				r	r	r	re-w1		re-w1						

位域	名称	描述
31:12	Reserved	保留, 必须保持复位值。

11	PDRDY	ADC掉电准备 (Power down ready) 0: 没有准备好; 1: 准备好。
10	RDY	ADC准备(Ready) 0: 没有准备好; 1: 准备好。
9	VREFRDY	VREFINT准备好 (VREFINT_READY) ADC内部输入buffer准备状态, 在测量VREFINT之前软件必须检测该状态位 0: VREFINT 没有准备好; 1: VREFINT准备好。
8	STR	通道开始位 (channel Start flag) 通道转换开始时该位被硬件设置为1, 由软件清除。 0: 通道转换未开始; 1: 通道转换已开始。
7	Reserved	保留, 必须保持复位值。
6	AWDG	模拟看门狗标志位 (Analog watchdog flag) 转换的电压值超出了ADC_AWDHIGH.HTH和ADC_AWDLOW.LTH寄存器定义的范围 时被硬件设置为1, 由软件清除 0: 没有发生模拟看门狗事件; 1: 发生模拟看门狗事件。
5	ENDCA	任意通道转换结束位 (Any End of conversion flag) 任意规则通道转换结束时被硬件设置为1。 0: 转换没有完成; 1: 转换完成。
4	ENDC	转换结束位 (End of conversion) 所有通道序列转换结束时被硬件设置为1。 0: 转换未完成; 1: 转换完成。
3	PH4F	第四段采样完成标志 0: 转换未完成; 1: 转换完成。
2	PH3F	第三段采样完成标志 0: 转换未完成; 1: 转换完成。
1	PH2F	第二段采样完成标志 0: 转换未完成; 1: 转换完成。
0	PH1F	第一段采样完成标志 0: 转换未完成; 1: 转换完成。

### 16.4.3 ADC 控制寄存器 1(ADC\_CTRL1)

地址偏移: 0x04

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved										AWDG IEN	ENDCA IEN	ENDCIEN	PH4IEN	PH3IEN	PH2IEN	PH1IEN
										rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:7	Reserved	保留，必须保持复位值。
6	AWDGIEN	模拟看门狗中断使能（Analog watchdog interrupt enable） 该位由软件设置和清除以禁止或允许模拟看门狗产生中断。 0：禁止模拟看门狗中断； 1：使能模拟看门狗中断。
5	ENDCAIEN	ENDCA的中断使能（Any End of conversion flag） 该位由软件设置和清除，以禁止或允许在规则转换序列转换结束后发生中断。 0：禁止ENDCA中断； 1：使能ENDCA中断。
4	ENDCIEN	ENDC的中断使能（Interrupt enable for ENDC） 该位由软件设置和清除，以禁止或允许在规则转换序列转换结束后发生中断。 0：禁止ENDC中断； 1：使能ENDC中断。
3	PH4IEN	第四段采样完成中断使能
2	PH3IEN	第三段采样完成中断使能
1	PH2IEN	第二段采样完成中断使能
0	PH1IEN	第一段采样完成中断使能

### 16.4.4 ADC 控制寄存器 2(ADC\_CTRL2)

地址偏移: 0x08

复位值: 0x0000 0000

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

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

Reserved											ALIG	CTU	ON
											rw	rw	rw

位域	名称	描述
31:3	Reserved	保留，必须保持复位值。
2	ALIG	数据对齐（Data alignment） 该位由软件设置和清除。 0：右对齐； 1：左对齐。
1	CTU	连续转换（Continuous conversion） 该位由软件设置和清除。如果该位置位，则转换将继续，直到该位被清除。 0：单次转换模式； 1：连续转换模式。
0	ON	A/D 转换器开/关（A/D converter ON/OFF） 该位由软件设置和清除。当该位为“0”时，写入“1”会将ADC从断电模式中唤醒。 应注意，转换器上电与转换开始之间存在延迟 $t_{STAB}$ 。 0：关闭ADC转换/校准并进入掉电模式； 1：启动ADC并开始转换。 注意：如果该寄存器中的其他位随着ON发生变化，则不会触发转换。这是为了防止触发错误的转换。

### 16.4.5 ADC 控制寄存器 3(ADC\_CTRL3)

地址偏移：0x0C

复位值：0x0000 0400

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

Reserved															
----------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

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

Reserved											TEMPEN	Reserved	REFSEL	VREFEN	Reserved
											rw		rw	rw	

位域	名称	描述
31:6	Reserved	保留，必须保持复位值。
5	TEMPEN	温度传感器使能（Temperature sensor enable） 该位由软件设置和清除，用于开启或禁止温度传感器通道 0：禁止温度传感器测量

位域	名称	描述
		1: 使能温度传感器测量
4	Reserved	保留, 必须保持复位值。
3	REFSEL	adc的vrefp的开关选择信号 0: 选择VDDA 1: 选择内部3.6V
2	VREFEN	VREFINT使能 (VREFINT_EN) ADC内部输入buffer使能, 在测量VREFINT之前软件必须使能该位 0: 禁止VREFINT测量 1: 使能VREFINT测量
1:0	Reserved	保留, 必须保持复位值。

### 16.4.6 ADC 采样时间寄存器 1(ADC\_SAMPT1)

地址偏移: 0x10

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAMP7[3:0]				SAMP6[3:0]				SAMP5[3:0]				SAMP4[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAMP3[3:0]				SAMP2[3:0]				SAMP1[3:0]				SAMP0[3:0]			
rw				rw				rw				rw			

位域	名称	描述
31:0	SAMPx[3:0]	通道x采样时间选择 (Channel x Sample time selection) 这些位用于独立选择每个通道的采样时间。通道选择位在采样期间必须保持不变。 0000: 4 周期 0001: 6 周期 0010: 14 周期 0011: 20 周期 0100: 29 周期 0101: 42 周期 0110: 56 周期 0111: 72 周期 1000: 88 周期 1001: 120 周期 1010: 182 周期

位域	名称	描述
		1011: 240 周期 1100: 300 周期 1101: 400 周期 1110: 480 周期 1111: 600 周期

### 16.4.7 ADC 采样时间寄存器 2(ADC\_SAMPT2)

地址偏移: 0x14

复位值: 0x0000 0000

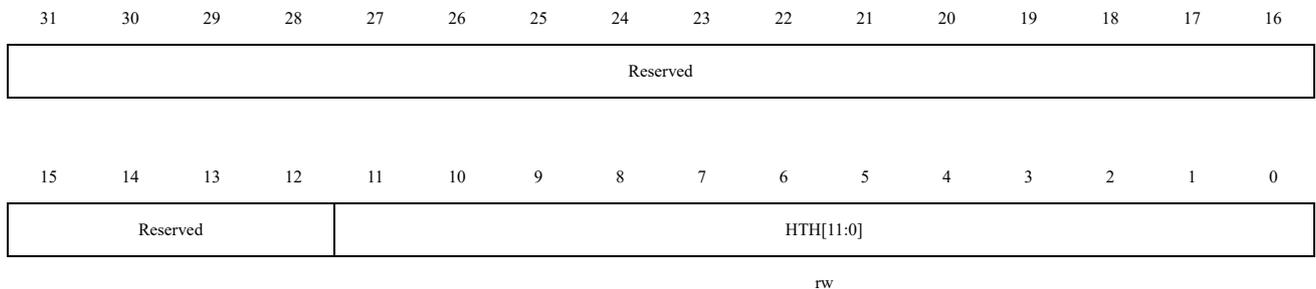
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAMP15[3:0]				SAMP14[3:0]				SAMP13[3:0]				SAMP12[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAMP11[3:0]				SAMP10[3:0]				SAMP9[3:0]				SAMP8[3:0]			
rw				rw				rw				rw			

位域	名称	描述
31:0	SAMPx[3:0]	通道x采样时间选择 (Channel x Sample time selection) 这些位用于独立选择每个通道的采样时间。通道选择位在采样期间必须保持不变。 0000: 4 周期 0001: 6 周期 0010: 14 周期 0011: 20 周期 0100: 29 周期 0101: 42 周期 0110: 56 周期 0111: 72 周期 1000: 88 周期 1001: 120 周期 1010: 182 周期 1011: 240 周期 1100: 300 周期 1101: 400 周期 1110: 480 周期 1111: 600 周期

### 16.4.8 ADC 看门狗高阈值寄存器(ADC\_AWDHIGH)

地址偏移: 0x18

复位值: 0x0000 0FFF

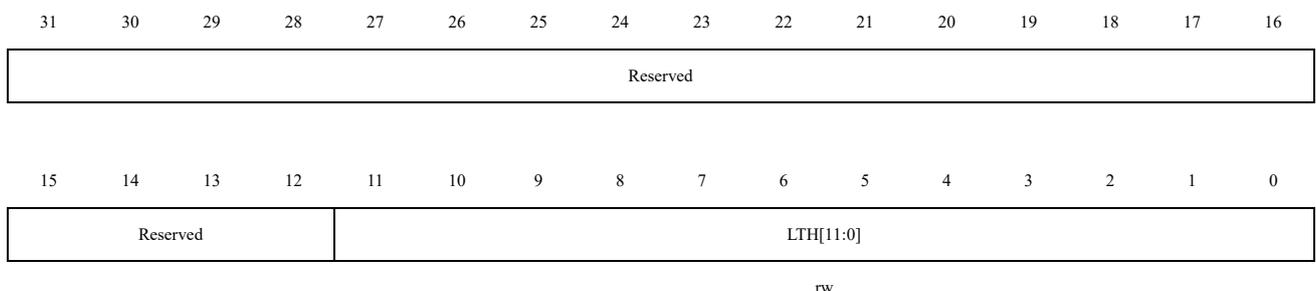


位域	名称	描述
31:12	Reserved	保留, 必须保持复位值。
11:0	HTH[11:0]	模拟看门狗高阈值 (Analog watchdog high threshold) 这些位定义模拟看门狗的高阈值。

### 16.4.9 ADC 看门狗高阈值寄存器(ADC\_AWDLOW)

地址偏移: 0x1C

复位值: 0x0000 0000

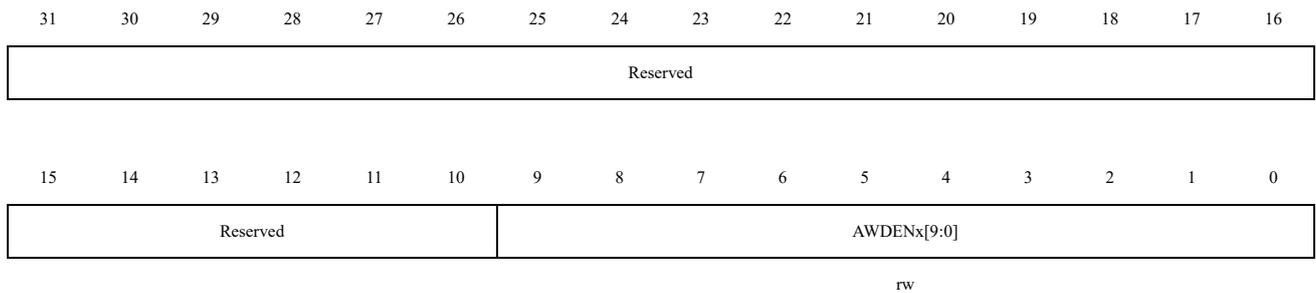


位域	名称	描述
31:12	Reserved	保留, 必须保持复位值。
11:0	LTH[11:0]	模拟看门狗低阈值 (Analog watchdog low threshold) 这些位定义模拟看门狗的低阈值。

### 16.4.10 ADC 看门狗使能寄存器(ADC\_AWDEN)

地址偏移: 0x20

复位值: 0x0000 0000

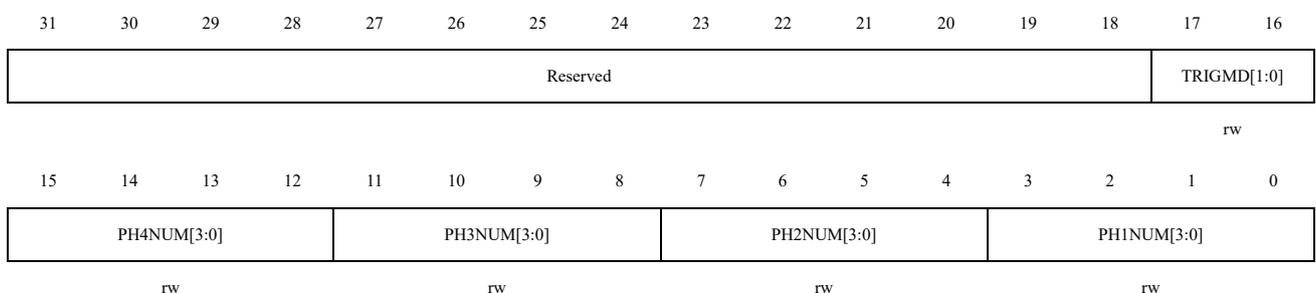


位域	名称	描述
31:10	Reserved	保留，必须保持复位值。
9:0	AWDENx[9:0]	ADC 模拟看门狗监测数据使能位 BIT0: DAT0 看门狗监测使能 BIT1: DAT1 看门狗监测使能 ..... BIT8: DAT8 看门狗监测使能 BIT9: DAT9 看门狗监测使能

### 16.4.11 ADC 段配置寄存器(ADC\_PHCFG)

地址偏移: 0x24

复位值: 0x0000 0000



位域	名称	描述
31:18	Reserved	保留，必须保持复位值。
17:16	TRIGMD[1:0]	触发模式选择 0:单段触发； 1:两段触发； 2:三段采样； 3:四段触发
15:12	PH4NUM[3:0]	四段采样模式下第四段采样次数
11:8	PH3NUM[3:0]	四段采样模式下第三段采样次数
7:4	PH2NUM[3:0]	四段采样模式下第二段采样次数
3:0	PH1NUM[3:0]	四段采样模式下第一段采样次数

## 16.4.12 ADC 触发选择寄存器(ADC\_TRIGSEL)

地址偏移: 0x28

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PH4EXTSEL[3:0]				PH3EXTSEL[3:0]				PH2EXTSEL[3:0]				PH1EXTSEL[3:0]			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWEN	EXTEN	TIM4_TRGOEN	TIM4_CC2EN	TIM4_CC1EN	TIM3_TRGOEN	TIM3_CC2EN	TIM3_CC1EN	TIM1_TRGOEN	TIM1_CC7EN	TIM1_CC6EN	TIM1_CC5EN	TIM1_CC4EN	TIM1_CC3EN	TIM1_CC2EN	TIM1_CC1EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

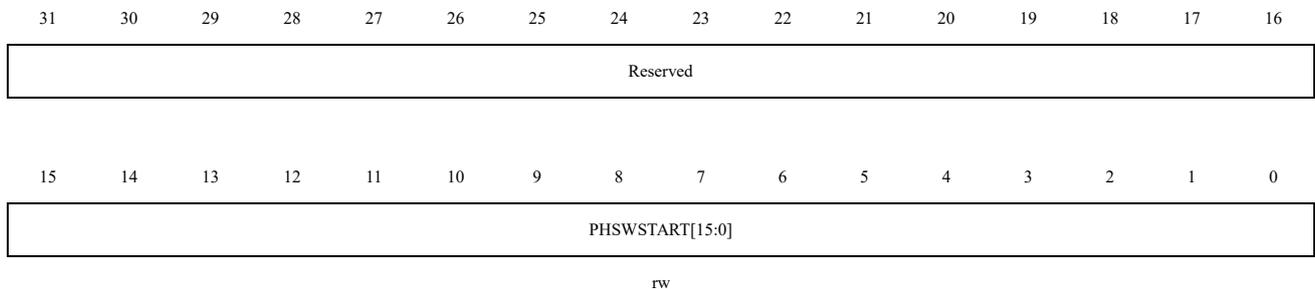
位域	名称	描述
31:16	PHxEXTSEL[3:0]	选择启动通道序列转换的外部事件 (External event select for regular group) 这些位选择外部事件以启动规则序列转换。  0000: 定时器1的 CC1 事件 0001: 定时器1的 CC2 事件 0010: 定时器1的 CC3 事件 0011: 定时器1的 CC4 事件 0100: 定时器1的 CC5 事件 0101: 定时器1的 CC6 事件 0110: 定时器1的 CC7 事件 0111: 定时器1的 TRGO 事件 1000: 定时器3的 CC1 事件 1001: 定时器3的 CC2 事件 1010: 定时器3的 TRGO 事件 1011: 定时器4的 CC1 事件 1100: 定时器4的 CC2 事件 1101: 定时器4的 TRGO 事件 1110: EXTI线 1111: SWSTART
15	SWEN	软件触发事件 0: 软件触发不使能; 1: 软件触发使能。
14	EXTIEN	外部EXTI线事件 0: 外部EXTI线不使能; 1: 外部EXTI线使能。
13	TIM4_TRGOEN	定时器4的 TRGO 事件 0: 转换事件不使能 1: 转换事件使能

位域	名称	描述
12	TIM4_CC2EN	定时器4的 CC2 事件 0: 转换事件不使能 1: 转换事件使能
11	TIM4_CC1EN	定时器4的 CC1 事件 0: 转换事件不使能 1: 转换事件使能
10	TIM3_TRGOEN	定时器3的 TRGO 事件 0: 转换事件不使能 1: 转换事件使能
9	TIM3_CC2EN	定时器3的 CC2 事件 0: 转换事件不使能 1: 转换事件使能
8	TIM3_CC1EN	定时器3的 CC1 事件 0: 转换事件不使能 1: 转换事件使能
7	TIM1_TRGOEN	定时器1的 TRGO 事件 0: 转换事件不使能 1: 转换事件使能
6	TIM1_CC7EN	定时器1的 CC7 事件 0: 转换事件不使能 1: 转换事件使能
5	TIM1_CC6EN	定时器1的 CC6 事件 0: 转换事件不使能 1: 转换事件使能
4	TIM1_CC5EN	定时器1的 CC5 事件 0: 转换事件不使能 1: 转换事件使能
3	TIM1_CC4EN	定时器1的 CC4 事件 0: 转换事件不使能 1: 转换事件使能
2	TIM1_CC3EN	定时器1的 CC3 事件 0: 转换事件不使能 1: 转换事件使能
1	TIM1_CC2EN	定时器1的 CC2 事件 0: 转换事件不使能 1: 转换事件使能
0	TIM1_CC1EN	定时器1的 CC1 事件 0: 转换事件不使能 1: 转换事件使能

### 16.4.13 ADC 软件配置寄存器(ADC\_PHSWTRIG)

地址偏移: 0x2C

复位值: 0x0000 0000

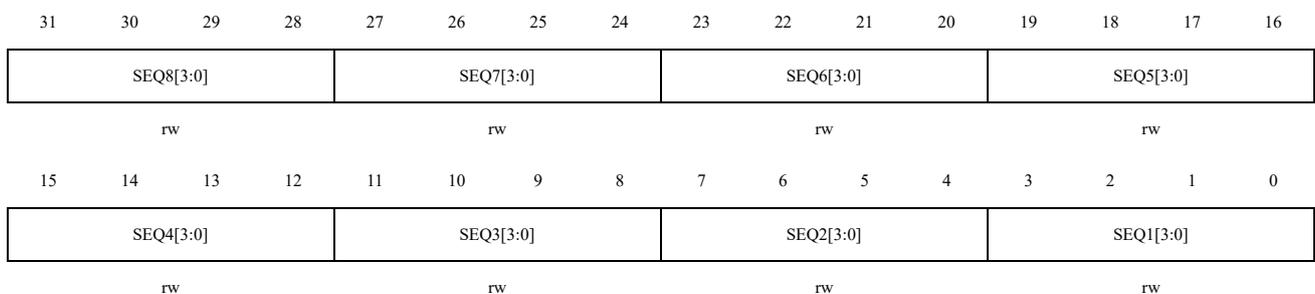


位域	名称	描述
31:16	Reserved	保留, 必须保持复位值。
15:0	PHSWSTART	软件触发模式的启动转换 第一段软件写0xA5A1, 第二段软件写0xF0F2, 第三段写0x5A53, 第四段写0x0FA4

### 16.4.14 ADC 序列寄存器 1(ADC\_SEQ1)

地址偏移: 0x30

复位值: 0x0000 0000



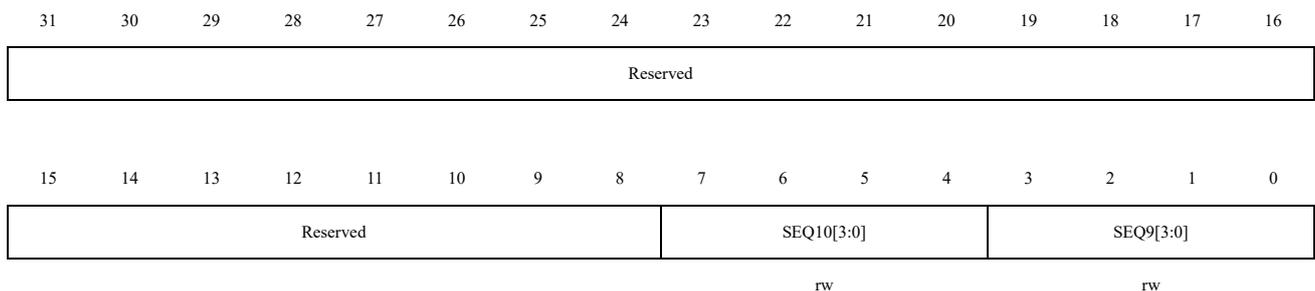
位域	名称	描述
31:28	SEQ8[3:0]	序列中的第8个转换的通道号 (8th conversion channel number in sequence)
27:24	SEQ7[3:0]	序列中的第7个转换的通道号 (7th conversion channel number in sequence)
23:20	SEQ6[3:0]	序列中的第6个转换的通道号 (6th conversion channel number in sequence)
19:16	SEQ5[3:0]	序列中的第5个转换的通道号 (5th conversion channel number in sequence)
15:12	SEQ4[3:0]	序列中的第4个转换的通道号 (4th conversion channel number in sequence)
11:8	SEQ3[3:0]	序列中的第3个转换的通道号 (3rd conversion channel number in sequence)
7:4	SEQ2[3:0]	序列中的第2个转换的通道号 (2nd conversion channel number in sequence)

位域	名称	描述
3:0	SEQ1[3:0]	序列中的第1个转换的通道号 (1st conversion channel number in sequence) 0000:通道0 ... 1111: 通道15

### 16.4.15 ADC 序列寄存器 2(ADC\_SEQ2)

地址偏移: 0x34

复位值: 0x0000 0000

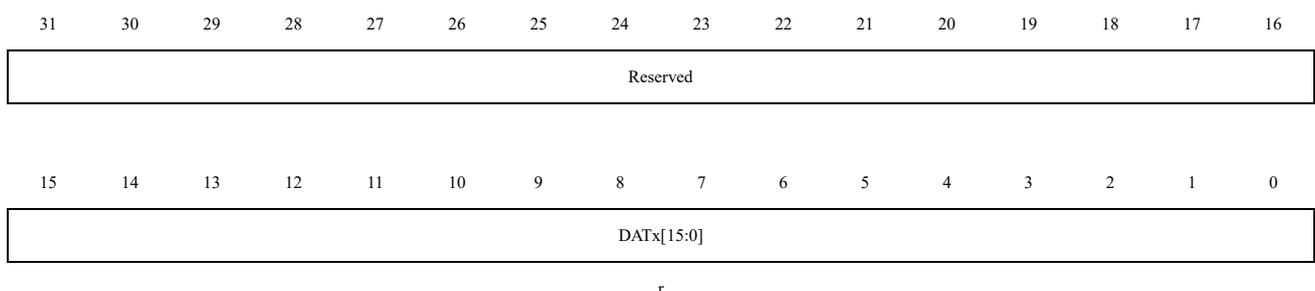


位域	名称	描述
31:8	Reserved	保留, 必须保持复位值。
7:4	SEQ10[3:0]	序列中的第10个转换的通道号 (10th conversion channel number in sequence)
3:0	SEQ9[3:0]	序列中的第9个转换的通道号 (9th conversion channel number in sequence) 0000:通道0 ... 1111: 通道15

### 16.4.16 ADC 规则数据寄存器 x(ADC\_DATx)(x = 0..9)

地址偏移: 0x38 – 0x5C

复位值: 0x0000 0000



位域	名称	描述
31:16	Reserved	保留，必须保持复位值。
15:0	DATx[15:0]	规则转换的数据（Regular data） 这些位是只读的，包含规则通道的转换结果。数据左对齐或右对齐。

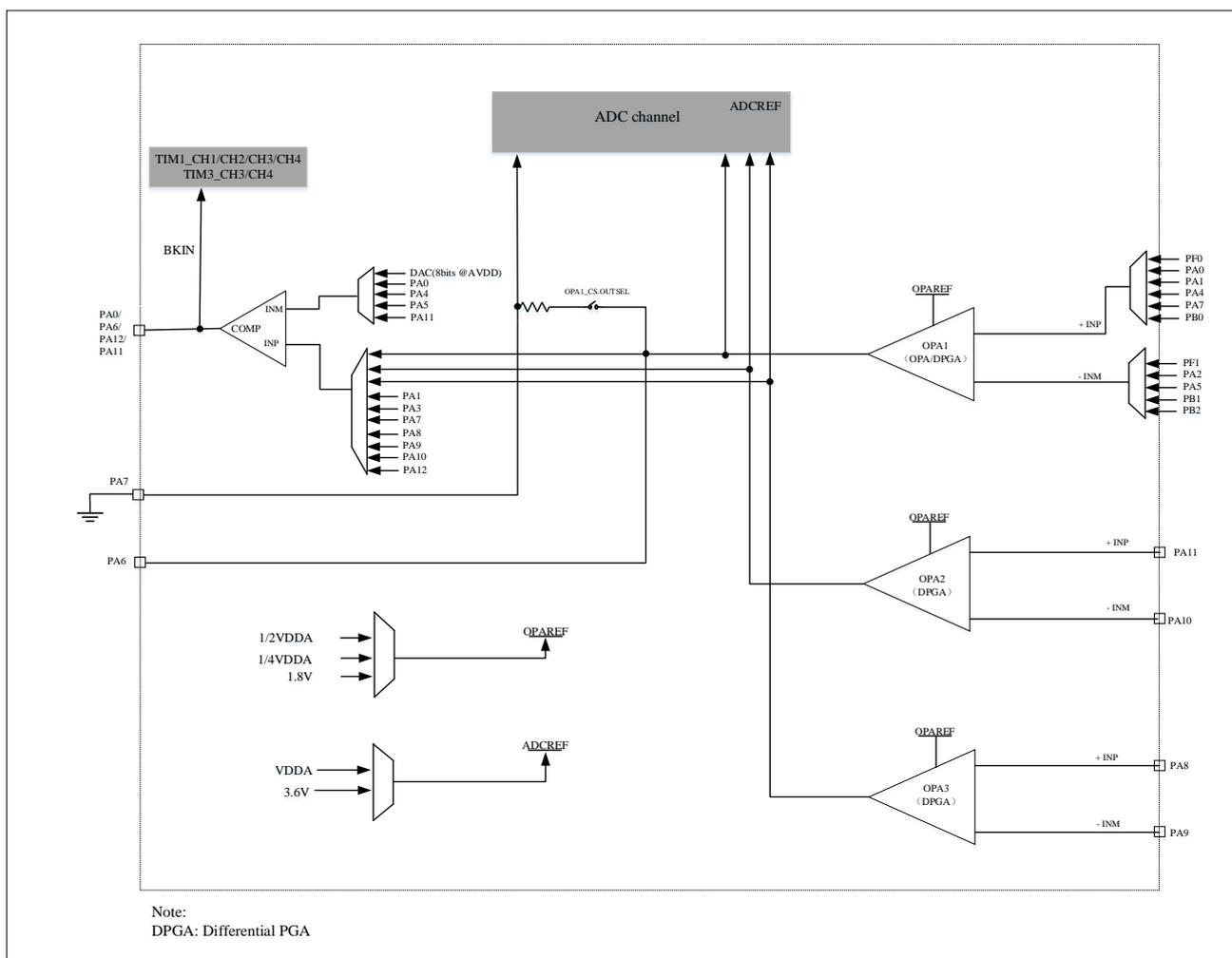
## 17 比较器 (COMP)

COMP 模块用于比较两个输入模拟电压的大小，并根据比较结果输出高/低电平。当“INP”输入端电压高于“INM”输入端电压时，比较器输出为高电平，当“INP”输入端电压低于“INM”输入端电压时，比较器输出为低电平。

### 17.1 COMP 系统连接框图

COMP 模块支持 1 个独立比较器，挂接在 APB1 总线上。

图 17-1 比较器系统连接图



## 17.2 COMP 特性

- 1 个独立的比较器
- 独立的 8bit DAC 的内部参考输入
- 支持滤波时钟
- 输出极性可配置高、低
- 迟滞配置可配置无、低、中、高
- 比较结果可输出到 I/O 端口或触发定时器，用于捕获事件、OCREF\_CLR 事件、刹车事件、产生中断
- 输入通道可复选 I/O 端口、专用的 8bit DAC
- COMP\_CTRL 寄存器可配只读或读写，在锁定的情况下需要系统复位或模块复位才能解锁
- 支持消隐 (Blanking)，可配置产生 Blanking 的消隐源
- 可配置滤波窗口大小
- 可配置滤波阈值大小
- 可配置用于滤波的采样频率

## 17.3 COMP 配置流程

完整的配置项包括如下所示，某些项目如果采用系统默认配置，跳过相应的配置项。

- 可配置的迟滞等级 COMP\_CTRL.HYST[1:0]。
- 配置输出极性 COMP\_CTRL.POL。
- 配置输入选择，比较器正极 COMP\_CTRL.INPSEL[3:0]，负极 COMPxCTRL.INMSEL[2:0]。
- 配置消隐源 COMP\_CTRL.BLANKING[2:0]。
- 配置滤波器采样窗口 COMP\_FILC.SAMPW[4:0]。
- 配置阈值 COMP\_FILC.THRESH[4:0]（阈值应当大于 COMP\_FILC.SAMPW[4:0]/2）。
- 配置滤波器采样频率。
- 打开滤波器使能 COMP\_FILC.FILEN。
- 打开比较器使能 COMP\_CTRL.EN。

**注意：**

1: 对于以上步骤，需先打开滤波器使能，再打开比较器使能，比较器使能需要在滤波（若启用）配置、使能完成后启用，此外在比较器控制寄存器锁定 LOCK 的情况下，只有通过复位才能取消锁定。

2: 对于比较器输入信号可能存在的毛刺，根据实际毛刺大小配置滤波长度和迟滞等级进行过滤。

## 17.4 COMP 工作模式

### 17.4.1 独立比较器

1 个比较器可独立配置，完成比较器功能。比较器的输出可以输出到 IO 端口，每一个比较器都有不同的重映射端口，通过配置可以选择比较器的输出，连接到相应的端口。

比较器输出，支持触发事件，比如可以配置成定时器 1 的刹车功能。

*注：具体配置参考比较器互联关系*

## 17.5 比较器互联关系

比较器输出端口的互联，可以参考 GPIO 的复用功能章节，定义了比较器 OUT 重映射的值。

比较器 OUT 引脚如下：

表 17-1 OUT 引脚

COMP_OUT
PA0
PA6
PA11
PA12

比较器 INP 引脚有如下配置：

表 17-2 INP 引脚配置

INPSEL	COMP
0000	OPA1.out
0001	OPA2.out
0010	OPA3.out
0011	PA1
0100	PA3
0101	PA7
0110	PA8
0111	PA9
1000	PA10
1001	PA12
Other:Reserved	

比较器 INM 引脚有如下配置：

表 17-3 INM 引脚配置

INMSEL	COMP1
000	VREF1

001	PA0
010	PA4
011	PA5
100	PA11

## 17.6 中断

COMP 支持中断响应。中断产生有如下 2 种情况。

- COMP\_CTRL.POL 极性不反转，中断使能，当 INPSEL > INMSEL 时，COMP\_CTRL.OUT 由硬件置为 1 时即产生比较器中断。
- COMP\_CTRL.POL 极性反转，中断使能，当 INPSEL < INMSEL 时，COMP\_CTRL.OUT 由硬件置为 1 时即产生比较器中断。

*注意：COMP 中断使用需先配置 EXTI line，参考 NVIC 章节。*

## 17.7 COMP 寄存器

### 17.7.1 COMP 寄存器总览

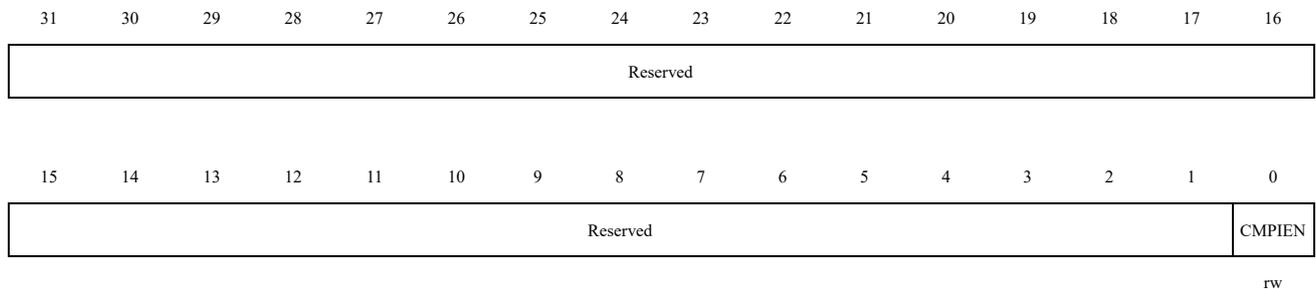
表 17-4 COMP 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00h	COMP_INT EN	Reserved																												CMPIEN			
04h	COMP_INTS TS	Reserved																												CMPIIS			
08h		Reserved																															
0Ch	COMP_LOK	Reserved																												CMPLK			
10h	COMP_CTL	Reserved										CLKSEL	PWRMD	Reserved	OUT	Reserved	BLKING	HYST	POL	Reserved			INPSEL[2:0]	INMSEL[3:0]	EN								
14h	COMP_FILC	Reserved															SAMPW [4:0]			THRESH [4:0]			FILEN										
18h	COMP_FILP	Reserved										CLKPSC[15:0]																					
1Ch ~ 3Ch		Reserved																															
40h	COMP_INV REF	Reserved															VREFSEL[7:0]							VREFEN									

### 17.7.2 COMP 中断使能寄存器(COMP\_INTEN)

偏移地址:0x00

复位值:0x0000 0000

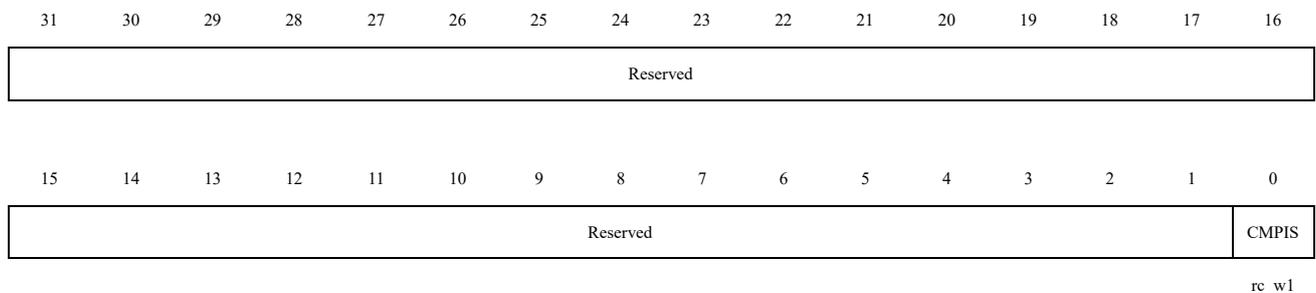


位域	名称	描述
31:1	Reserved	保留，必须保持复位值。
0	CMPIEN	比较器中断使能 0: 比较器中断使能关闭 1: 比较器中断使能开启。

### 17.7.3 COMP 中断状态寄存器(COMP\_INTSTS)

偏移地址:0x04

复位值:0x0000 0000



位域	名称	描述
31:1	Reserved	保留，必须保持复位值。
0	CMPIS	比较器中断状态位 写1 清零

### 17.7.4 COMP 锁定寄存器(COMP\_LOCK)

偏移地址:0x0C

复位值:0x0000 0000

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

位域	名称	描述
31:1	Reserved	保留，必须保持复位值。
0	CMPLK	仅写一次，是由软件控制的，只能通过系统重置/模块复位来清除 设置此位可以将 COMP_CTRL 寄存器设置为只读 0: COMP_CTRL 是可读写的 1: COMP_CTRL 是只读的...

### 17.7.5 COMP 控制寄存器(COMP\_CTRL)

偏移地址:0x10

复位值:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved											CLKSEL	PWRMD	Reserved	OUT	Reserved	
											rw	rw				rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved	BLKING	HYST[1:0]			POL	Reserved				INPSEL[3:0]			INMSEL[2:0]		EN	
		rw			rw				rw			rw			rw	

位域	名称	描述
31:21	Reserved	保留，必须保持复位值。
20	CLKSEL	时钟选择 0: COMP 时钟选择来源于 HSI (64M); 1: COMP 时钟选择来源于 LSI (32K),常应用于低功耗模式下;
19	PWRMD	指示比较器控制模式选择 0: 低功耗模式 1: 正常模式
18	Reserved	保留，必须保持复位值。
17	OUT	指示比较器输出的状态 0: 输出低 1: 输出高

位域	名称	描述
16:15	Reserved	保留，必须保持复位值。
14	BLKING	比较器输出消隐。 0: 不消隐 1: 选择 TIM1_CC5 作为消隐源
13:12	HYST[1:0]	这些位选择比较器的迟滞等级。 00: 无迟滞; 01: 低迟滞; 10: 中等迟滞; 11: 高迟滞。
11	POL	该位用于反转比较器的输出 0:输出未反转; 1:输出反转。
10:8	Reserved	保留，必须保持复位值。
7:4	INPSEL[3:0]	比较器正端选择位 4'b0000: OPAMP1 的输出; 4'b0001: OPAMP2 的输出; 4'b0010: OPAMP3 的输出; 4'b0011: PA1; 4'b0100: PA3; 4'b0101: PA7; 4'b0110: PA8; 4'b0111: PA9; 4'b1000: PA10; 4'b1001: PA12; other: Reserved
3:1	INMSEL[2:0]	比较器负端输入选择位 3'b000: VREF1(8-bit DAC); 3'b001: PA0; 3'b010: PA4; 3'b011: PA5; 3'b100: PA11
0	EN	该位打开/关闭 COMP 0: 比较器已禁用; 1: 比较器已启用。

### 17.7.6 COMP 滤波控制寄存器(COMP\_FILC)

偏移地址:0x14

复位值:0x0000 0000

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

Reserved

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

位域	名称	描述
31:11	Reserved	保留，必须保持复位值。
10:6	SAMPW[4:0]	低通滤波器采样窗口大小，采样窗口 = SAMPW + 1。
5:1	THRESH[4:0]	低通滤波器门限置，样本窗口中至少出现相反状态的采样阈值，才能改变输出状态，此值要求大于 SAMPW / 2。
0	FILEN	滤波器使能位 0：关闭； 1：使能。

### 17.7.7 COMP 滤波时钟寄存器(COMP\_FILP)

偏移地址:0x18

复位值:0x0000 0000

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

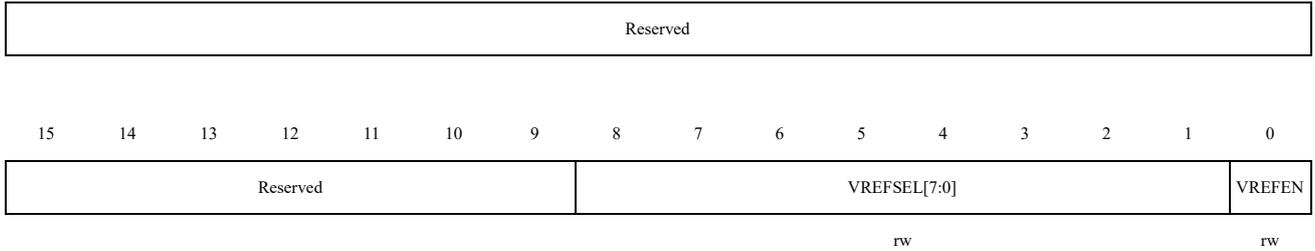
位域	名称	描述
31:16	Reserved	保留，必须保持复位值。
15:0	CLKPSC[15:0]	低通滤波采样时钟预分频，系统时钟分频数 = CLKPSC + 1。 0：每 1 个时钟； 1：每 2 个时钟； 2：每 3 个时钟； ... 65535：每 65536 个时钟；

### 17.7.8 COMP 参考输入比较电压寄存器(COMP\_INVREF)

偏移地址:0x40

复位值:0x0000 0000

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



位域	名称	描述
31:9	Reserved	保留，必须保持复位值。
8:1	VREFSEL[7:0]	电压控制 内部比较器参考输入比较电压 VREF1 档位选择 0~0b'11111111 对应输出电压范围 0~VREF1+，共 255 个档位，例如 51 则代表 (51) * VREF1+/255 = 1/5 VREF1+
0	VREFEN	内部 DAC 使能 0: 禁用； 1: 启用。

## 18 OPAMP

OPAMP 模块可以灵活配置,适用于独立运放 PGA 和跟随等模式应用。OPAMP 的输入范围是 0V 到 VDDA, 输出范围是 0.4V 到 VDDA-0.4V

### 18.1 OPAMP 特性

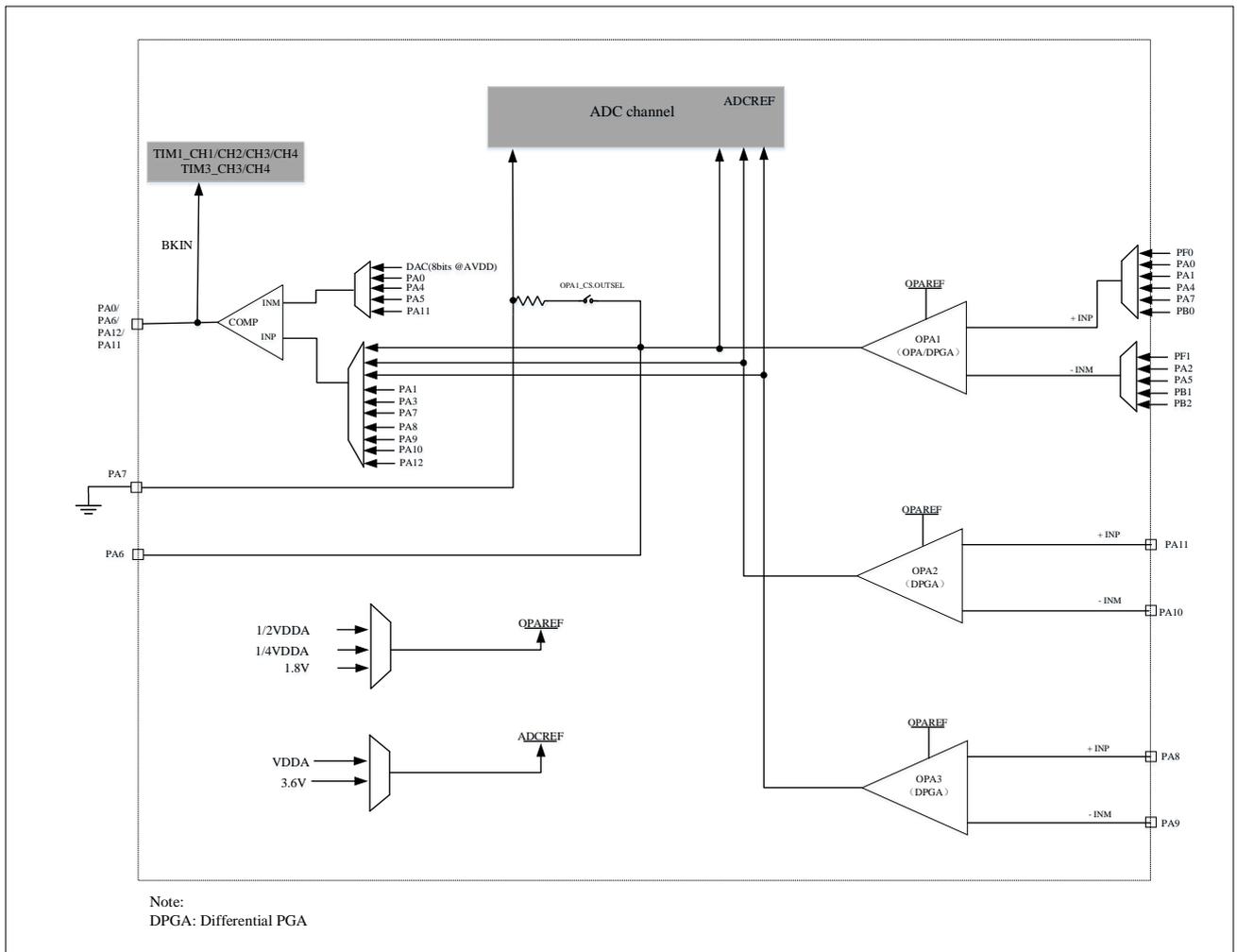
- OPA1 支持独立运放模式, 跟随模式, 差分 PGA 模式, 单端 PGA 模式; OPA2、OPA3 差分 PGA 模式和跟随模式
- 支持轨到轨输入, 输入范围是 0 到 VDDA, 输出范围是 0.4 到 VDDA-0.4 可编程增益
- OPAMP 通过外部电阻连接可配置为仪表放大器
- 内部电阻反馈网络可配置, 2%精度
- 可编程增益设置
- OPA1 单端 PGA 增益: 2X、4X、8X、16X、32X 倍
- OPA1 差分 PGA 增益: 1X、2X、4X、8X、16X、32X 倍
- OPA2、OPA3 差分 PGA 增益: 1X、2X、4X、8X、16X 倍
- 增益带宽: 5MHz
- 支持独立写保护

## 18.2 OPAMP 功能描述

OPAMP 有两个输入 (VP,VM)，一个输出，VP/VM 输入端可以选择接不同的 GPIO ,OPAMP 的输出可作为 COMP 或 ADC 的输入。

具体连接如下图

图 18-1 OPAMP 系统连接图

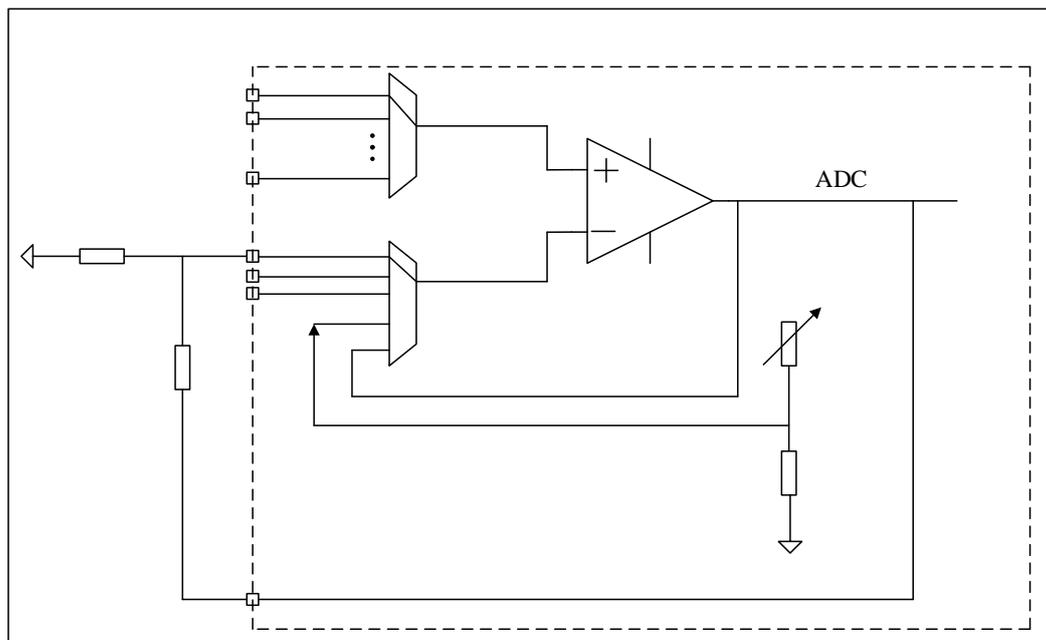


## 18.3 OPAMP 工作模式

### 18.3.1 OPAMP1 外部放大模式(仅 OPA1 支持)

外部放大模式即放大倍数由连接的电阻电容决定。OPAMP1\_CS.MOD = 00 通用模式，OPAMP1\_CS.VPSEL 选择正端输入，OPAMP1\_CS.VMSEL 选择负端输入。使用外部电阻组成闭环放大系统。

图 18-2 OPAMP1 外部放大模式



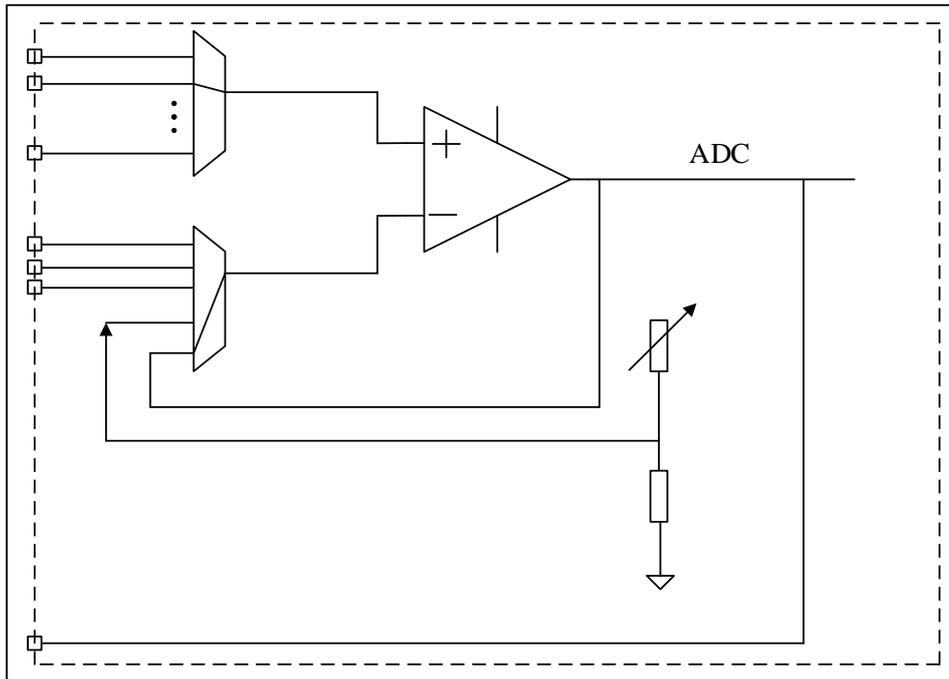
### 18.3.2 OPAMP1 跟随模式

跟随模式，电压是直接跟随，VMSEL 端必须配置为和 OPAMP 输出端口直连。

OPAMP1\_CS.MOD = “11”为内部跟随功能，OPAMP1\_CS.VPSEL 选择正端输入，OPAMP1\_CS.VMSEL 由芯片内部连接到输出端口。OPAMP2\_CS.MOD = ‘1’为内部跟随功能，OPAMP3\_CS.MOD = ‘1’为内部跟随功能。

没有占用的 VM 引脚可以作为其他 GPIO 使用。

图 18-3 OPAMP1 跟随模式



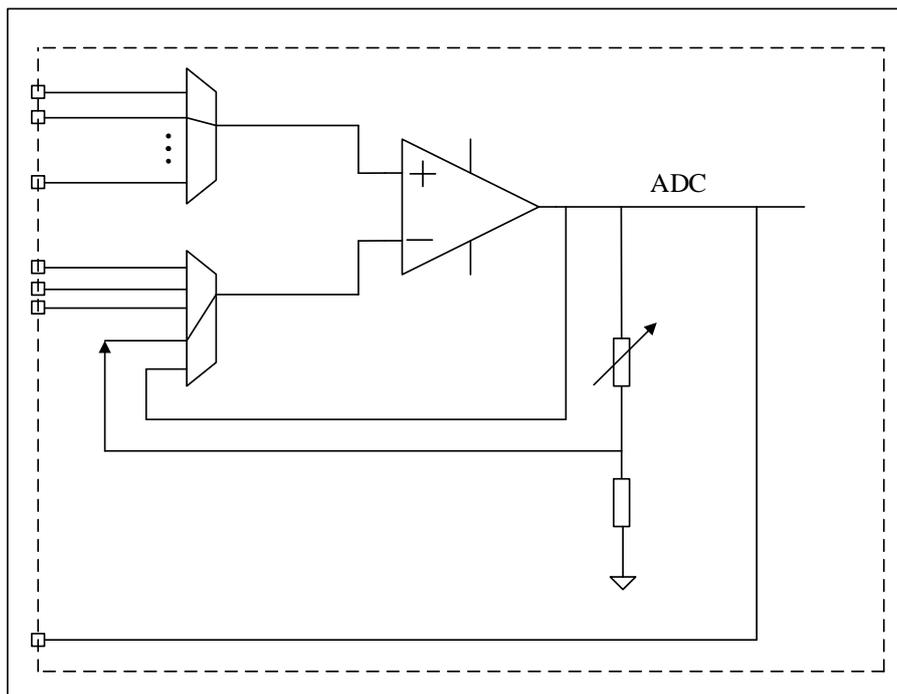
### 18.3.3 OPAMP1 单端增益 (PGA) 模式

单端增益放大模式，即 PGA 模式，通过内置的电阻反馈网络对输入电压进行放大。

OPAMP1\_CS.MOD = “10”为 PGA 功能，支持 2/4/8/16/32 放大倍数，OPAMP1\_CS.VMSEL 引脚必须设置为浮空。OPAMP1\_CS.VPSEL 选择正端输入。

OPAMP1 的 VM、VP 未使用的输入引脚可以作为普通 GPIO 使用

图 18-4 内部增益模式

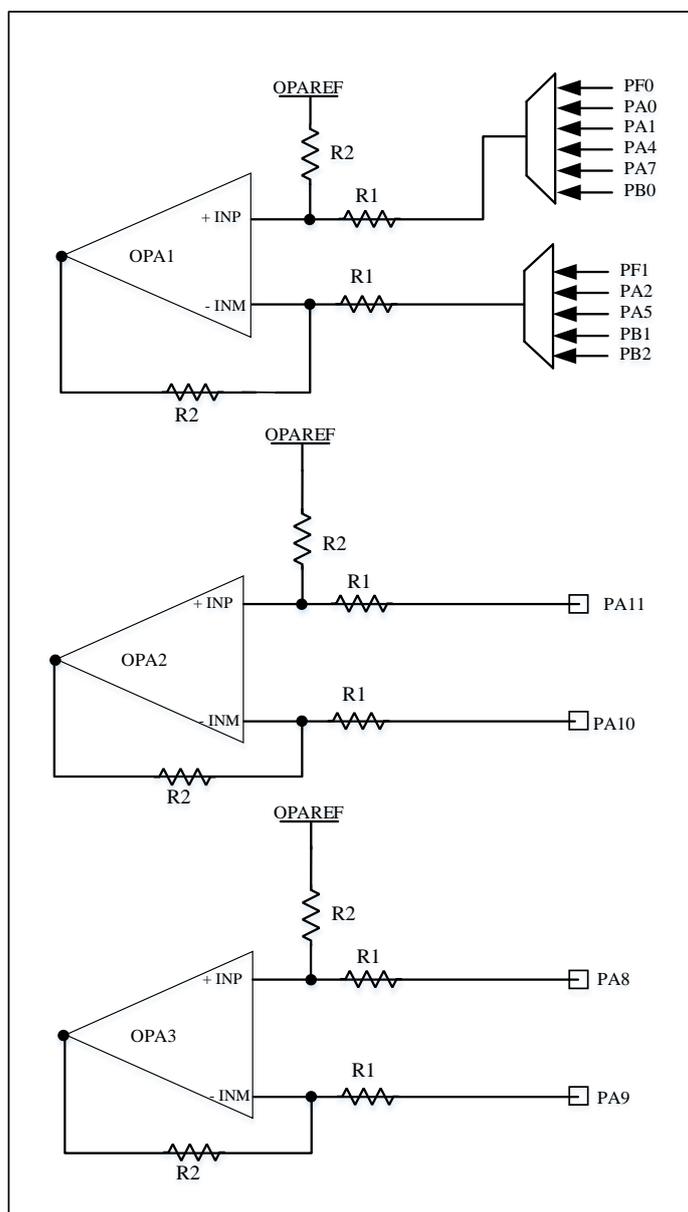


### 18.3.4 差分增益模式

差分放大模式，放大电压可调，OPAMP1 支持 1/2/4/8/16/32，OPAMP2、OPAMP3 支持 1/2/4/8/16；根据需求选择合适的 VP 和 VM，OPAMP1 输出管脚可作放大输出，OPAMP2、OPAMP3 无输出管脚。

VP/VM 未使用输入的引脚可作普通 GPIO 使用。

图 18-5 差分增益模式



### 18.3.5 OPAMP 写保护

通过配置 OPAMP\_LOCK 寄存器，可以对 OPAMP1、2、3 的写保护进行设置。当设置了写保护后，软件将不能对相应 OPAMP1、2、3 控制寄存器进行写操作，芯片复位或 OPA 模块复位，才能取消写保护功能。

## 18.4 OPAMP 寄存器

### 18.4.1 OPAMP 寄存器总览

表 18-1 OPAMP 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	OPAMP1_CS	Reserved														OUTSEL	VMSEL[2:0]			Reserved	VPSEL[2:0]			Reserved			GAIN[2:0]			MOD[1:0]		EN		
0x04	OPAMP2_CS	Reserved																								GAIN[2:0]			Reserved	MOD	EN			
0x08	OPAMP3_CS	Reserved																								GAIN[2:0]			Reserved	MOD	EN			
0xC	OPAMP_LOCK	Reserved																								OPAMPOLK								
0x10	OPAMP_VREFSEL	Reserved																								OPA3SEL[1:0]			OPA2SEL[1:0]			OPA1SEL[1:0]		

### 18.4.2 OPAMP 控制寄存器 1 (OPAMP1\_CS)

偏移地址:0x00

复位值:0x0000 0000

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

位域	名称	描述
31:16	Reserved	保留，必须保持复位值。
15	OUTSEL	1: OPAMP1 输出外加 40k 电阻，

位域	名称	描述
		0: OPAMP1 输出不加 40k 电阻
14:12	VMSEL [2:0]	OPAMP1 反向输入端从选择 (Inverted input selection) 000: PF1 001: PA2 010: PA5 011: PB1 100: PB2 默认值 =3'b000
11	Reserved	保留, 必须保持复位值。
10:8	VPSEL[2:0]	OPAMP1 正向输入端从选择 (Non inverted input selection) 000: PF0 001: PA0 010: PA1 011: PA4 100: PA7 101: PB0 default value =0000
7:6	Reserved	保留, 必须保持复位值。
5:3	GAIN[2:0]	OPAMP1 增益控制 当 OPAMP1_CS.MOD[1:0] = 2b'01 时: 000: 差分 PGA 增益 = 1; 001: 差分 PGA 增益= 2 010: 差分 PGA 增益= 4 011: 差分 PGA 增益= 8 100: 差分 PGA 增益= 16 101: 差分 PGA 增益= 32 当 OPAMP1_CS.MOD[1:0] = 2b'10 时: 000: 内部 PGA 增益= 2; 001: 内部 PGA 增益= 4 010: 内部 PGA 增益 = 8 011: 内部 PGA 增益= 16 100: 内部 PGA 增益= 32
2:1	MOD[1:0]	OPAMP1 工作模式选择 00: 通用 OPA 模式 ; 01: 差分 PGA 模式; 10: 内部 PGA 模式; 11: 内部 BUFFER 模式.
0	EN	OPAMP1 使能 0: 不使能 1: 使能

### 18.4.3 OPAMP 控制寄存器 2 (OPAMP2\_CS)

偏移地址:0x04

复位值:0x0000 0000

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

位域	名称	描述
31:6	Reserved	保留, 必须保持复位值。
5:3	GAIN[2:0]	OPAMP2 增益控制 000: 差分 PGA2 增益=1; 001: 差分 PGA2 增益=2 010: 差分 PGA2 增益=4 011: 差分 PGA2 增益=8 100: 差分 PGA2 增益=16
2	Reserved	保留, 必须保持复位值。
1	MOD	OPAMP2 工作模式控制信号 0: 差分 PGA 模式 1: 内部 BUFFER 模式.
0	EN	1: OPAMP2 使能 0: OPAMP2 禁能

### 18.4.4 OPAMP 控制寄存器 3 (OPAMP3\_CS)

偏移地址:0x08

复位值:0x0000 0000

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

位域	名称	描述
31:6	Reserved	保留，必须保持复位值。
5:3	GAIN[2:0]	OPAMP3 增益控制 000: 差分 PGA 增益= 1; 001: 差分 PGA 增益= 2 010: 差分 PGA 增益= 4 011: 差分 PGA 增益= 8 100: 差分 PGA 增益= 16
2	Reserved	保留，必须保持复位值。
1	MOD	OPAMP3 工作模式控制信号 0: 差分 PGA 模式 1: 内部 BUFFER 模式
0	EN	1: OPAMP3 使能 0: OPAMP3 禁能

### 18.4.5 OPAMP 锁定寄存器 (OPAMP\_LOCK)

偏移地址:0x0C

复位值:0x0000 0000

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

rw

位域	名称	描述
31:1	Reserved	保留，必须保持复位值。
0	OPALK	仅写一次，是由软件控制的，只能通过系统重置/模块复位来清除 设置此位可以将 OPAMP1_CS/OPAMP2_CS/OPAMP3_CS 寄存器设置为只读 0: OPAMP1_CS/OPAMP2_CS/OPAMP3_CS 寄存器是可读写的 1: OPAMP1_CS/OPAMP2_CS/OPAMP3_CS 寄存器是只读的...

### 18.4.6 OPAMP 参考电压选择寄存器 (OPAMP\_VREFSEL)

偏移地址:0x10

复位值:0x0000 0000

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

Reserved															
----------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

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

Reserved										OPA3SEL[2:0]		OPA2SEL[2:0]		OPA1SEL[2:0]	
										rw		rw		rw	

位域	名称	描述
31:6	Reserved	保留，必须保持复位值。
5:4	OPA3SEL[1:0]	OPAMP3 参考电压选择 00: 参考电压选择 0.9V，输出 1.8V， 01: 参考电压选择 VDD/8，输出 VDD/4 10: 参考电压选择 VDD/4，输出 VDD/2 11: 参考电压选择 0.9V，输出 1.8V
3:2	OPA2SEL[1:0]	OPAMP2 参考电压选择 00: 参考电压选择 0.9V，输出 1.8V， 01: 参考电压选择 VDD/8，输出 VDD/4 10: 参考电压选择 VDD/4，输出 VDD/2 11: 参考电压选择 0.9V，输出 1.8V
1:0	OPA1SEL[1:0]	OPAMP1 参考电压选择 00: 参考电压选择 0.9V，输出 1.8V， 01: 参考电压选择 VDD/8，输出 VDD/4 10: 参考电压选择 VDD/4，输出 VDD/2 11: 参考电压选择 0.9V，输出 1.8V

## 19 独立看门狗（IWDG）

### 19.1 简介

内置独立看门狗（IWDG），解决软件错误导致的问题。看门狗定时器使用非常灵活，提高了系统的安全性和定时控制的准确性。

独立看门狗（IWDG）由运行在 32KHz 的低速内部时钟（LSI 时钟）驱动，在死循环事件或 MCU 卡死发生时，它仍然可以运行。这可以提供更高的安全级别、定时精度和看门狗的灵活性。它可以通过重置来解决由于软件故障引起的系统故障。IWDG 最适合需要看门狗在主应用程序之外作为完全独立进程运行但时序精度限制较低的应用程序。

当电源控制寄存器 PWR\_CTRL.IWDGRSTEN 位置‘1’，IWDG 计数器达到 0 时，会产生系统复位（若该位置‘0’，IWDG 会计数但不产生复位）。IWDG 复位也可用于低功耗唤醒。

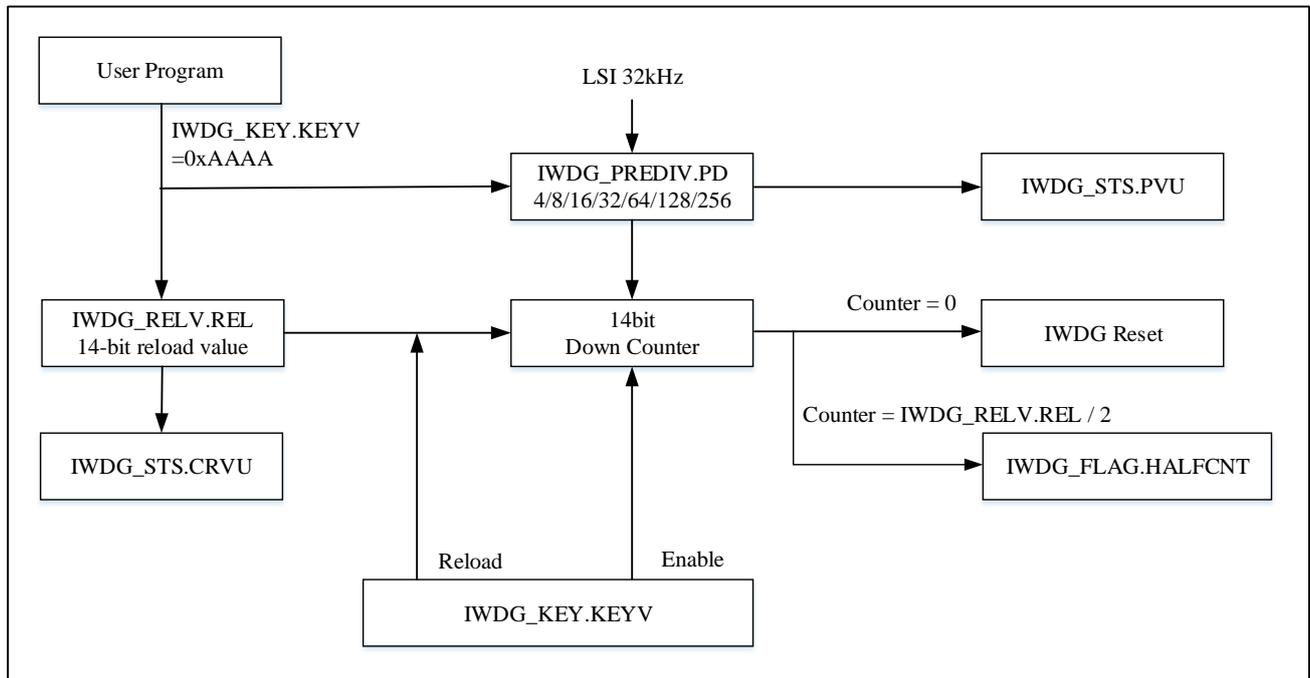
*注：本章基于系统默认值 PWR\_CTRL.IWDGRSTEN=1 描述。*

### 19.2 主要特性

- 独立运行的 14 位递减计数器
- RC 振荡器提供独立时钟源，在 STOP 模式下仍能正常工作
- 可以匹配复位和低功耗唤醒
- 当递减计数器达到 0x000 时，系统复位（如果激活了看门狗）

## 19.3 功能描述

图 19-1 独立看门狗功能框图



要启用 IWDG，我们需要将 0xCCCC 写入 IWDG\_KEY.KEYV[15:0]位，计数器开始递减计数。当计数器计数到 0x000 时，它会向 MCU 产生一个复位信号 (IWDG\_RESET)。除此之外，只要在复位前将 0xAAAA（重载请求）写入 IWDG\_KEY.KEYV[15:0]位，计数器值就会设置为 IWDG\_RELV.REL[13:0]位中的重载值并防止看门狗复位整个设备。

如果通过选项字节使能“硬件看门狗定时器”功能，则看门狗将在系统上电后自动开始运行并产生系统复位，除非软件在计数器到达‘0’之前重新加载计数器。

如果使能“中断功能”，则看门狗将在计数器到达“IWDG\_RELV.REL/2”之后产生中断。

### 19.3.1 寄存器访问保护

IWDG\_PREDIV 和 IWDG\_RELV 寄存器具有写保护功能。在修改这两个寄存器数据之前，必须先配置 IWDG\_KEY 寄存器为 0x5555。配置成其他任何数据，都将再次启动寄存器写保护。IWDG\_STS.PVU 指示预分频器值更新是否正在进行。IWDG\_STS.CRVU 指示 IWDG 是否正在更新重载值。当预分频器值和/或重载值更新时，硬件设置 IWDG\_STS.PVU 位和/或 IWDG\_STS.CRVU 位。预分频器值和/或重载值更新完成后，硬件清除 IWDG\_STS.PVU 位和/或 IWDG\_STS.CRVU 位。

重载操作（IWDG\_KEY 配置 0xAAAA）也会启动写保护功能。

### 19.3.2 IWDG 中断

若配置 IWDG\_CTRL.ITE=1，当 IWDG 的 DOWN COUNTER 小于等于 IWDG\_RELV.RVL[13:0]/2 时，则会产生对应的 IWDG 中断，并需要通过喂狗退出中断。

### 19.3.3 调试模式

在调试模式下（Cortex-M0 内核停止），IWDG 计数器将继续正常工作或停止，具体取决于调试模块中的 DBG\_CTRL.IWDG\_STOP 位。如果该位设置为“1”，则计数器停止。该位为“0”时，计数器正常工作。

## 19.4 用户界面

IWDG 模块用户界面包含 4 个寄存器：密钥寄存器（IWDG\_KEY）、预分频寄存器（IWDG\_PREDIV）、重装载寄存器（IWDG\_RELV）和状态寄存器（IWDG\_STS）。

### 19.4.1 操作流程

当 IWDG 从软件（将 0xAAAA 写入 IWDG\_KEY.KEYV[15:0]位）或硬件（清零 FLASH\_OB.WDG\_SW 位）复位启用时。它从 0xFFFF 开始递减计数。向下计数间隙由预分频 LSI 时钟确定。重新加载计数器后，新一轮递减计数器的值将从 IWDG\_RELV.REL[13:0]中的值开始，而不是 0xFFFF。

程序正常运行时，软件需要在计数器到达 0 前喂狗，开始新一轮的递减计数。当计数器达到 0 时，表示程序故障。IWDG 在这种情况下产生复位信号。

如果用户想要配置 IWDG 预分频和重装载值寄存器，需要先将 0x5555 写入 IWDG\_KEY.KEYV[15:0]。然后确认 IWDG\_STS.CRVU 位和 IWDG\_STS.PVU 位。IWDG\_STS.CRVU 位指示重装载值更新正在进行，IWDG\_STS.PVU 表示预分频值更新正在进行。只有当这两位为 0 时，用户才能更新相应的值。当更新正在进行时，硬件将相应位设置为 1。此时，读取 IWDG\_PREDIV.PD[2:0]或 IWDG\_RELV.REL[13:0]无效，因为数据需要同步到 LSI 时钟域。从 IWDG\_PREDIV.PD[2:0]或 IWDG\_RELV.REL[13:0]读取的值将在硬件清除 IWDG\_STS.PVU 位或 IWDG\_STS.CRVU 位后才有效。

如果应用程序使用多个重装载值或预分频值，则必须等到 IWDG\_STS.CRVU 位复位后才能更改重装载值，IWDG\_STS.PVU 位复位后才能更改预分频值。但是，在更新预分频值和重装载值后，或只更新预分频值后，或只更新重装载值后，无需等到 IWDG\_STS.CRVU 位或 IWDG\_STS.PVU 位复位后才能继续执行代码（即使在进低功耗模式的情况下，写入操作也会被考虑并完成）。

预分频寄存器和重装载寄存器控制产生复位的时间，如表 19-1。

表 19-1 IWDG 计数最大和最小复位时间

预分频系数	PD [2:0]	最短时间 (ms) REL[13:0]=0x000	最长时间 (ms) REL[13:0]=0x3FFF
/4	000	0.125	2048
/8	001	0.25	4096
/16	010	0.5	8192
/32	011	1.0	16384
/64	100	2	32768
/128	101	4	65536
/256	11x	8	131072

### 19.4.2 IWDG 配置流程

软件配置流程：

1. 将 0x5555 写入 IWDG\_KEY.KEYV[15:0]位以启用对 IWDG\_PREDIV 和 IWDG\_RELV 寄存器的写访问；
2. 检查 IWDG\_STS.PVU 位或 IWDG\_STS.CRVU 位，如果为 0，则继续下一步；
3. 配置 IWDG\_PREDIV.PD[2:0]位以选择预分频值；
4. 配置 IWDG\_RELV.REL[13:0]位重装载值；
5. 将 0xAAAA 写入 IWDG\_KEY.KEYV[15:0]位，用重装载值更新计数器；
6. 通过软件或硬件将 0xCCCC 写入 IWDG\_KEY.KEYV[15:0]位来启用看门狗。  
如果用户想改变预分频值和重装载值，重复步骤 1~5。如果没有，只需按照第 5 步喂狗。

## 19.5 IWDG 寄存器

### 19.5.1 IWDG 寄存器总览

表 19-2 IWDG 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00h	IWDG_KEY	Reserved																KEYV[15:0]															
04h	IWDG_STS	Reserved																								CRVU	PVU						
08h	IWDG_PREDIV	Reserved																								PD[2:0]							
0Ch	IWDG_RELV	Reserved													REL[13:0]																		
10h	IWDG_CTRL	Reserved																								ITE							

### 19.5.2 IWDG 密钥寄存器 (IWDG\_KEY)

偏移地址：0x00

复位值：0x00000000

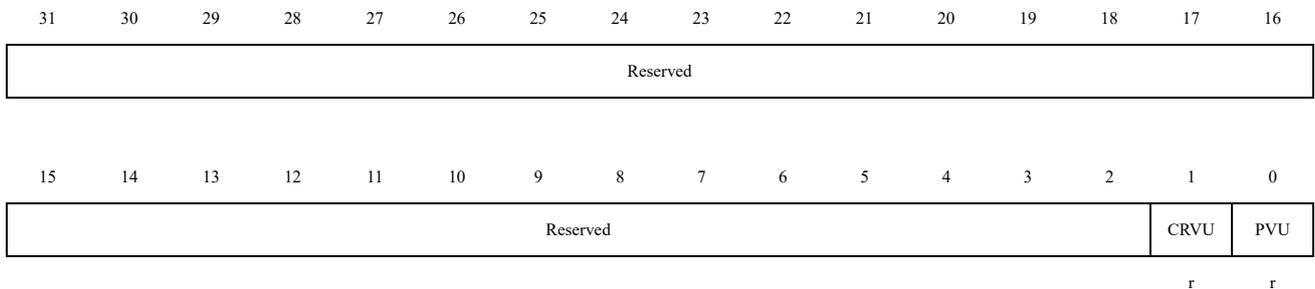
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEYV[15:0]															
rc_w0															

位域	名称	描述
31:16	保留	保留，必须保持复位值。
15:0	KEYV[15:0]	密钥寄存器：只有特定的值才能发挥特定的作用 0xCCCC：启动看门狗计数器，如果硬件看门狗使能则无效（启动） 0xAAAA：用 IWDG_RELV 寄存器中的 REL 值重新加载计数器以防止复位（喂狗） 0x5555：禁用 IWDG_PREDIV 和 IWDG_RELV 寄存器的写保护（解除保护）

### 19.5.3 IWDG 状态寄存器 (IWDG\_STS)

偏移地址: 0x04

复位值: 0x00000000

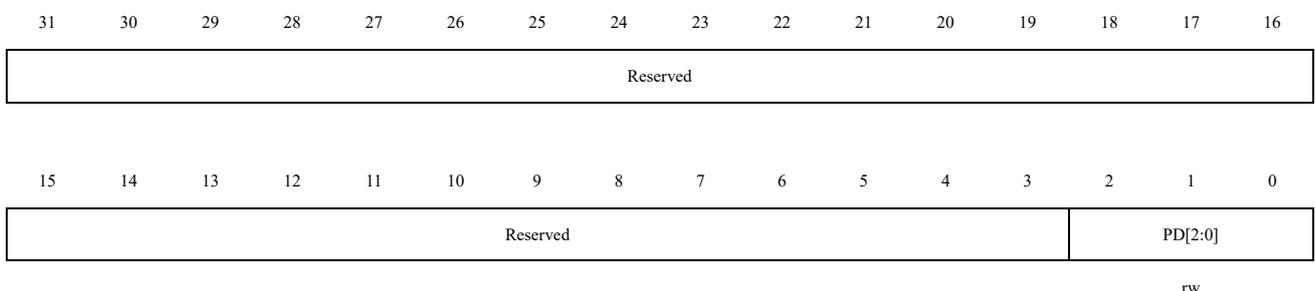


位域	名称	描述
31:2	保留	保留, 必须保持复位值。
1	CRVU	看门狗重装载值更新 重装载值更新: 该位表示正在更新重装载值。硬件置位, 硬件清零。软件只能在 IWDG_KEY.KEYV[15:0]位的值为 0x5555 且该位为 0 时尝试更改 IWDG_RELV.REL[13:0]的值。
0	PVU	看门狗预分频值更新 预分频值更新: 该位表示正在更新预分频值。硬件置位, 硬件清零。软件只能在 IWDG_KEY.KEYV[15:0]位的值为 0x5555 且该位为 0 时尝试更改 IWDG_PREDIV.PD[2:0]的值。

### 19.5.4 IWDG 预分频寄存器 (IWDG\_PREDIV)

偏移地址: 0x08

复位值: 0x00000000



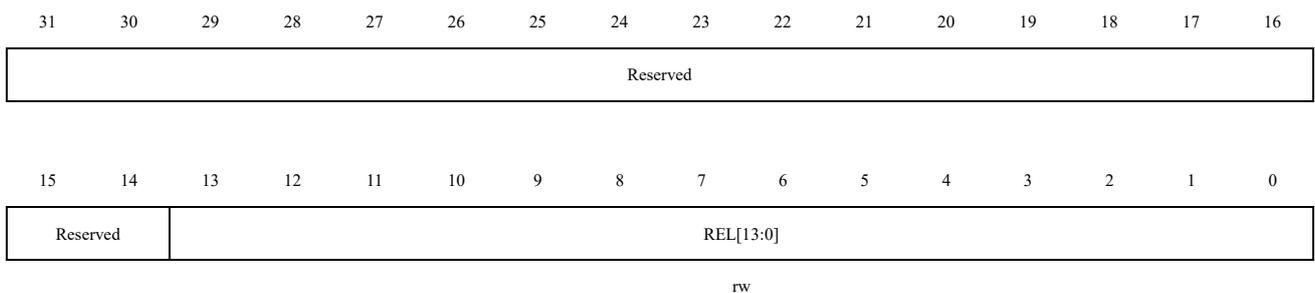
位域	名称	描述
31:3	保留	保留, 必须保持复位值。
2:0	PD[2:0]	预分频因子

位域	名称	描述
		当 IWDG_KEY.KEYV[15:0] 不是 0x5555 时具有写访问保护。IWDG_STS.PVU 位必须为 0，否则 PD[2:0] 值无法更改。分频系数如下： 000: 预分频因子=4 001: 预分频因子=8 010: 预分频因子=16 011: 预分频因子=32 100: 预分频因子=64 101: 预分频因子=128 其他: 预分频因子=256 注意: 读取该寄存器将返回来自 VDD 电压域的预分频值。如果正在进行写操作, 则回读值可能无效。因此, 读取值仅在 IWDG_STS.PVU 位为 0 时有效。

### 19.5.5 IWDG 重装载寄存器 (IWDG\_RELV)

偏移地址: 0x0C

复位值: 0x00003FFF

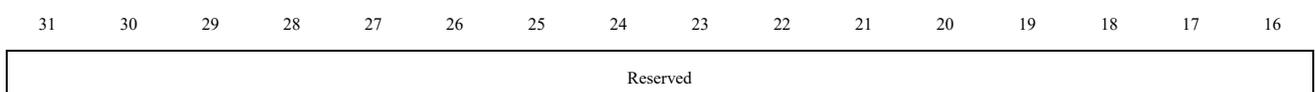


位域	名称	描述
31:14	保留	保留, 必须保持复位值。
13:0	REL[13:0]	看门狗计数器重装载值。 带写保护。定义看门狗计数器的重装载值, 每次将 0xAAAA 写入 IWDG_KEY.KEYV[15:0] 位时将其加载到计数器。然后计数器从该值开始倒计时。看门狗超时周期可以根据这个重装载值和时钟预分频值计算, 参考表 19-1。 该寄存器只能在 IWDG_STS.CRVU 位为 0 时修改。 注意: 读取该寄存器将返回来自 VDD 电压域的重装载值。如果正在进行写操作, 则回读值可能无效。因此, 读取值仅在 IWDG_STS.CRVU 位为 0 时有效。

### 19.5.6 IWDG 控制寄存器 (IWDG\_CTRL)

偏移地址: 0x10

复位值: 0x00000000



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

Reserved	ITE
----------	-----

rw

位域	名称	描述
31:1	保留	保留，必须保持复位值。
0	ITE	IWDG 定时中断使能，当 IWDG 中 DOWN COUNTER 小于等于 IWDG_REVL.RVL[13:0]/2， IWDG 输出中断 0：中断禁能 1：中断使能 <i>注：通过喂狗退出中断</i>

## 20 串行外设接口（SPI）

### 20.1 SPI 简介

SPI 可以工作在主模式或从模式，支持全双工和单工高速通讯模式，并且具有硬件 CRC 计算能力且可配置多主模式。

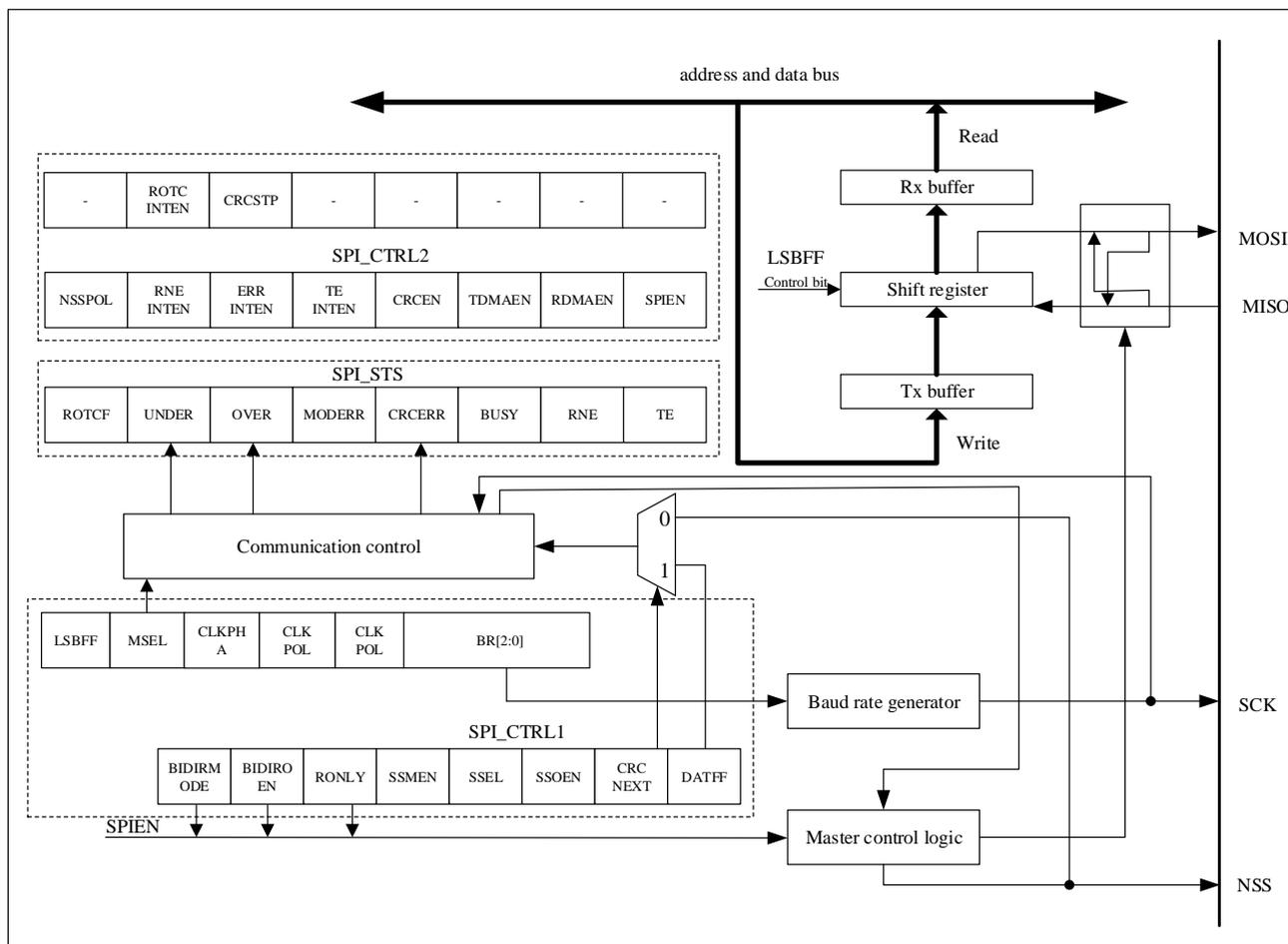
### 20.2 SPI 主要特性

- 全双工和单工同步模式
- 支持主模式、从模式和多主模式
- 支持 8bit 或 16bit 数据帧格式
- 数据位顺序可编程
- 硬件或软件片选管理
- 时钟极性和时钟相位可配置
- 发送和接收支持硬件 CRC 计算及校验
- 支持 DMA 传输功能

## 20.3 SPI 功能描述

### 20.3.1 通用描述

图 20-1 SPI 框图



为了连接外部设备，SPI 接口有 4 个引脚与外设器件连接，具体如下：

- SCLK: 串行时钟引脚，该信号从主设备 SCLK 引脚输出，由从设备 SCLK 引脚输入
- MISO: 主输入/从输出引脚，数据从主设备的 MISO 引脚输入，由从设备的 MISO 引脚输出
- MOSI: 主输出/从输入引脚，数据从主设备的 MOSI 引脚输出，由从设备的 MOSI 引脚输入
- NSS: 片选引脚，有两种 NSS 引脚类型，外部引脚和内部引脚。如果内部引脚检测到高电平，SPI 工作在主模式，相反，SPI 工作在从模式。用户可以使用主设备的一个标准 I/O 引脚控制从设备的 NSS 引脚

#### 软件 NSS 模式

当 SPI\_CTRL1.SSMEN = 1（图 20-2），软件从设备管理被使能。

软件 NSS 模式时，不需要使用 NSS 引脚。在这种模式下，通过写 SPI\_CTRL1.SSEL 位（主模式 SPI\_CTRL1.SSEL = 1，从模式 SPI\_CTRL1.SSEL = 0），驱动内部 NSS 信号电平。

#### 硬件 NSS 模式

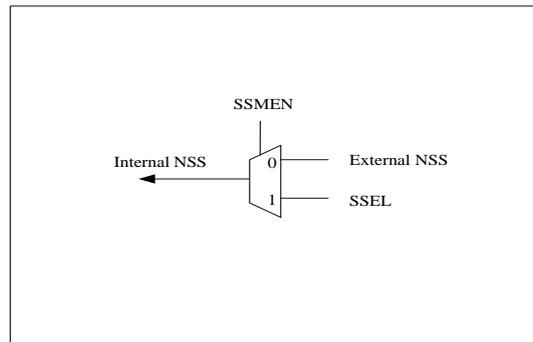
当 SPI\_CTRL1.SSMEN = 0（图 20-2），软件从设备管理被禁能。

NSS 输入模式：主设备的 NSS 输出被禁止 (SPI\_CTRL1.MSEL = 1, SPI\_CTRL1.SSOEN = 0)，允许操作在多主模式下。在整个数据帧传输期间主机应该连接 NSS 到高电平，从机应该连接 NSS 到低电平。

NSS 输出模式：主设备的 NSS 输出被使能 (SPI\_CTRL1.MSEL = 1, SPI\_CTRL1.SSOEN = 1)，主设备必须驱动 NSS 到低电平，所有与主设备连接并且设置为硬件 NSS 模式的设备将会检测到低电平，并自动进入从模式。当主设备的 NSS 没有被驱动到低电平，设备进入从模式，并产生主模式失效错误。

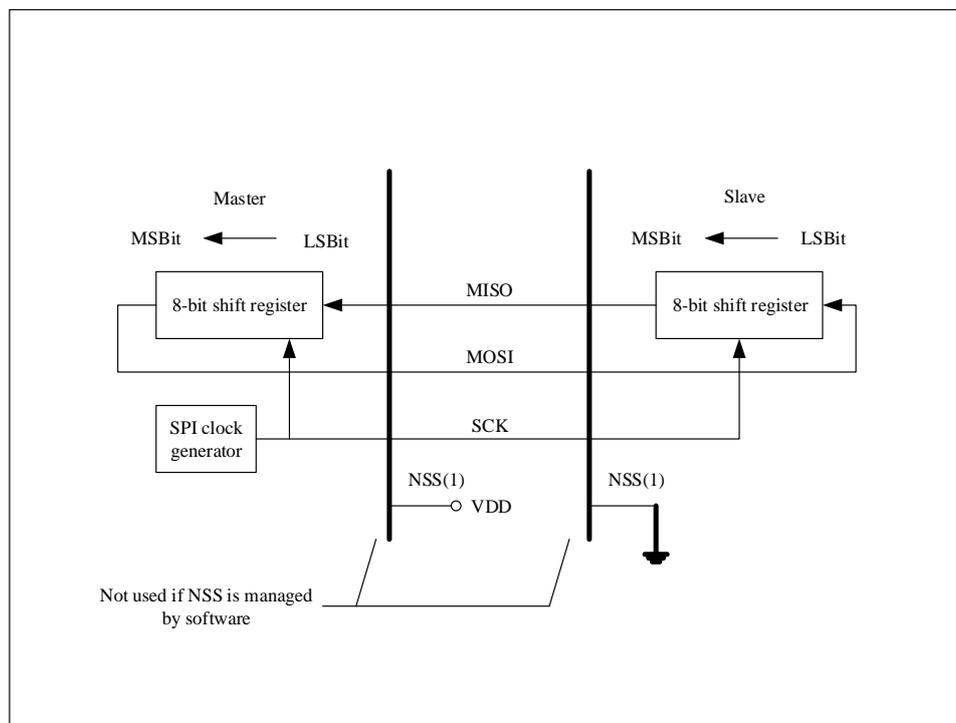
注：软件模式或硬件模式的选择，取决于通讯协议中是否需要 NSS 控制。如果不需要，可以选择软件模式，释放一个 GPIO 管脚另作他用。

图 20-2 硬件/软件的从选择管理



下图是一个单主和单从设备互连的例子。

图 20-3 单主和单从应用



注：NSS 引脚设置为输入

主设备通过 SCLK 引脚输出同步时钟信号，主设备的 MOSI 引脚连接到从设备的 MOSI 引脚，并且主设备的 MISO 引脚连接到从设备的 MISO 引脚，以便数据可以在设备之间传输。主设备和从设备之间的连续数据传输，通过 MOSI 引脚发送数据到从设备，而从设备通过 MISO 引脚发送数据到主设备。

### SPI 时序模式

通过设置 SPI\_CTRL1.CLKPOL 位和 SPI\_CTRL1.CLKPHA 位，用户可以选择数据捕获的时钟沿。

当 CLKPOL = 0, CLKPHA = 0，空闲时 SCLK 引脚将保持低电平，数据将在第一个时钟沿被采样，即上升沿。

当 CLKPOL = 0, CLKPHA = 1，空闲时 SCLK 引脚将保持低电平，数据将在第二个时钟沿被采样，即下降沿。

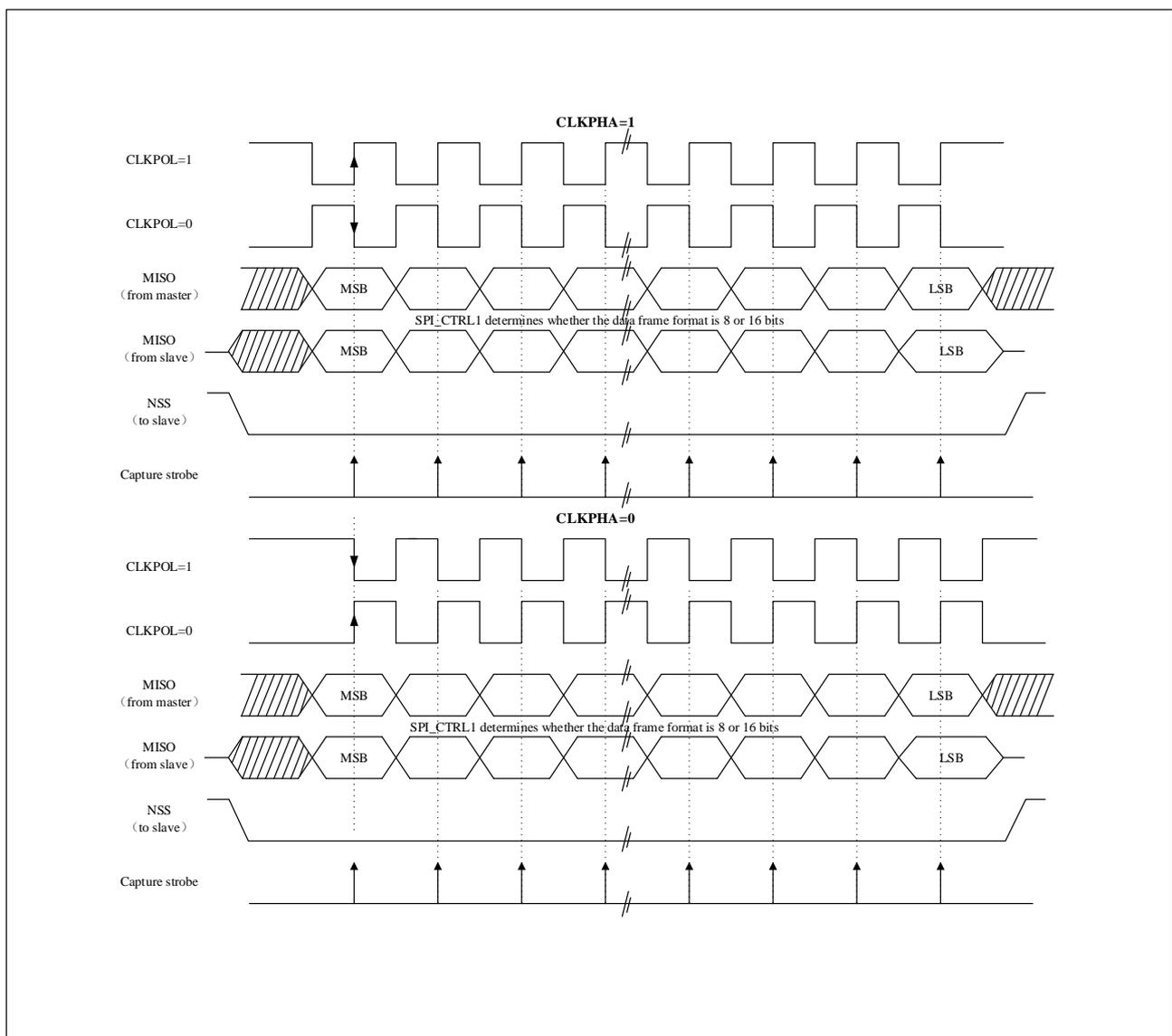
当 CLKPOL = 1, CLKPHA = 0，空闲时 SCLK 引脚将保持高电平，数据将在第一个时钟沿被采样，即下降沿。

当 CLKPOL = 1, CLKPHA = 1，空闲时 SCLK 引脚将保持高电平，数据将在第二个时钟沿被采样，即上升沿。

不管选择哪种时序模式，主设备和从设备的时序模式配置必须相同。

图 20-4 是当 SPI\_CTRL1.LSBFF = 0 时，SPI 传输的 4 种 CLKPHA 和 CLKPOL 位组合时序。

图 20-4 数据时钟时序图



## 数据格式

通过设置 SPI\_CTRL1.LSBFF 位，用户可以选择数据的位顺序，当 SPI\_CTRL1.LSBFF = 0，SPI 将先发送数据的高位（MSB），当 SPI\_CTRL1.LSBFF = 1，SPI 将先发送数据的低位（LSB）。

通过设置 SPI\_CTRL1.DATFF 位，用户可以选择数据帧格式。

## 20.3.2 SPI 工作模式

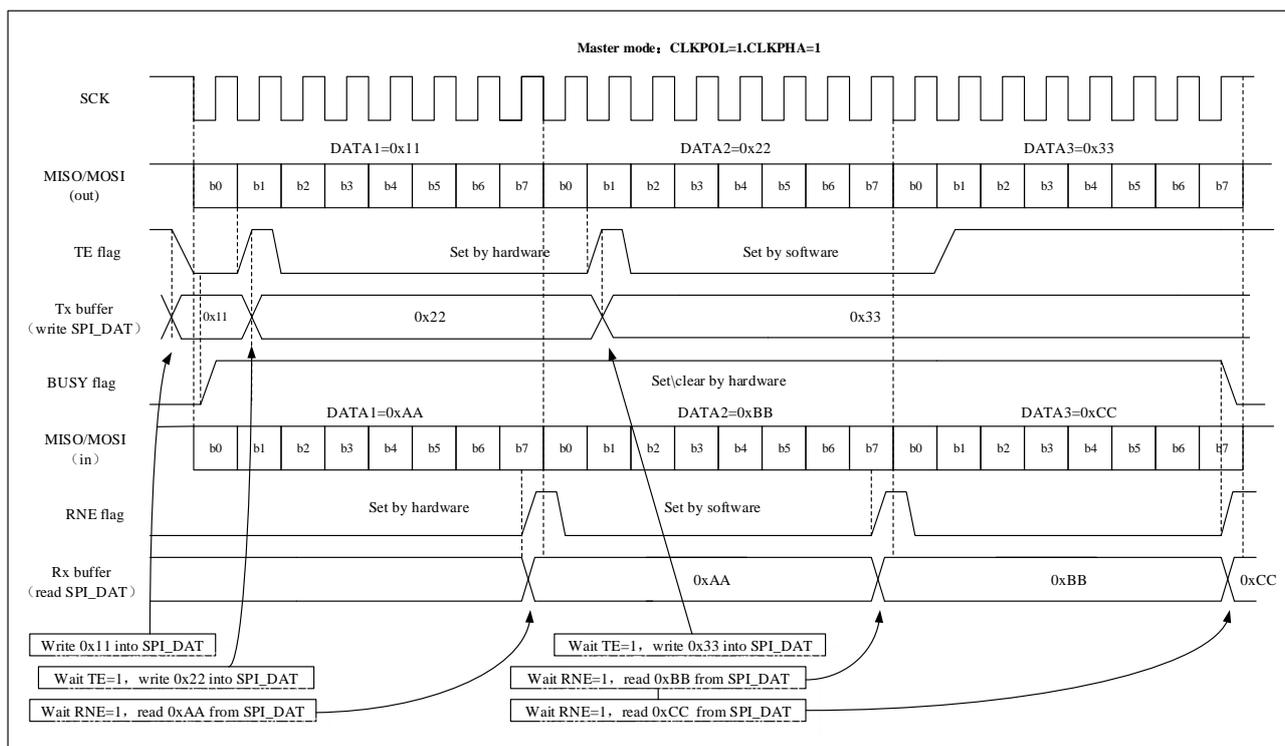
### ■ 主机全双工模式（SPI\_CTRL1.MSEL = 1，SPI\_CTRL1.BIDIRMODE = 0，SPI\_CTRL1.RONLY = 0）

第一个数据被写到 SPI\_DAT 寄存器后，将会开始传输，数据第一个位被发送时，数据字节并行从数据寄存器装载进入移位寄存器，然后数据位按照 SPI\_CTRL1.LSBFF 位的配置，数据位按照 MSB 或 LSB 顺序被串行移位进入 MOSI 引脚。与此同时，在 MISO 引脚上接收到的数据，按照同样顺序被串行地移位进入移位寄存器，然后并行装载入 SPI\_DAT 寄存器。

1. 设置 SPI\_CTRL2.SPIEN 位为 1，使能 SPI 模块；
2. 写待发送的第一个数据到 SPI\_DAT（这个写操作会清除 SPI\_STS.TE 标志位）；
3. 等待 SPI\_STS.TE 标志位置 1 后，再写入第二个待发送的数据到 SPI\_DAT 寄存器，等待 SPI\_STS.RNE 标志位置 1 后，读取 SPI\_DAT 寄存器获得第一个接收的数据，读取 SPI\_DAT 寄存器，SPI\_STS.RNE 标志位会清 0。重复上述操作，发送后续的数据，同时接收第 n-1 个数据；
4. 等待 SPI\_STS.RNE 置 1 后，读取最后一个数据；
5. 等待 SPI\_STS.TE 标志位置 1，等待 SPI\_STS.BUSY 标志位清除后再关闭 SPI 模块。

数据的发送和接收处理可以在 SPI\_STS.RNE 标志位或 SPI\_STS.TE 标志位的上升沿产生的中断处理程序中实现。

图 20-5 主机全双工模式下连续传输时，SPI\_STS.TE/RNE/BUSY 的变化示意图



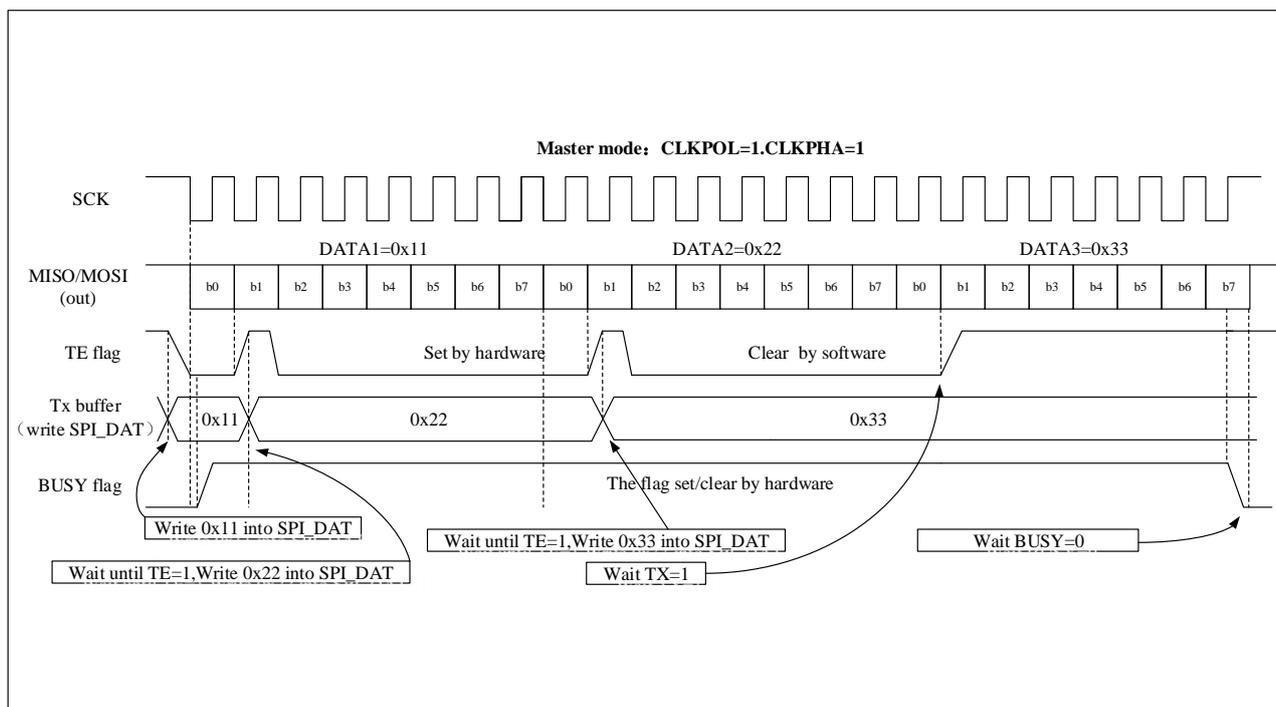
■ 主机双线单向仅发送模式 (SPI\_CTRL1.MSEL = 1, SPI\_CTRL1.BIDIRMODE = 0, SPI\_CTRL1.ONLY = 0)

双线单向仅发送模式与全双工模式硬件配置一致，但是在双线单向仅发送模式时，接收的数据将不会被读取，因此 SPI\_STS.OVER 标志位将会置位，软件应该忽略这个位。软件操作流程如下（图 20-6 主机单向只发送模式下连续传输时，SPI\_STS.TE/BUSY 变化示意图）：

1. 设置 SPI\_CTRL2.SPIEN 位为 1，使能 SPI 模块；
2. 写待发送的第一个数据到 SPI\_DAT 寄存器（该操作会清除 SPI\_STS.TE 标志位）；
3. 等待 SPI\_STS.TE 标志位置 1，写待发送的第二个数据到 SPI\_DAT 寄存器，重复这个操作发送后续的数据；
4. 写最后一个数据到 SPI\_DAT 寄存器，等待 SPI\_STS.TE 标志位置 1，然后等待 SPI\_STS.BUSY 位清除，完成所有数据的发送。

数据发送可以在 SPI\_STS.TE 标志位上升沿产生的中断处理程序里实现。

图 20-6 主机单向只发送模式下连续传输时，SPI\_STS.TE/BUSY 变化示意图



■ 主机双线单向仅接收模式 (SPI\_CTRL1.MSEL = 1, SPI\_CTRL1.BIDIRMODE = 0, SPI\_CTRL1.ONLY = 1)

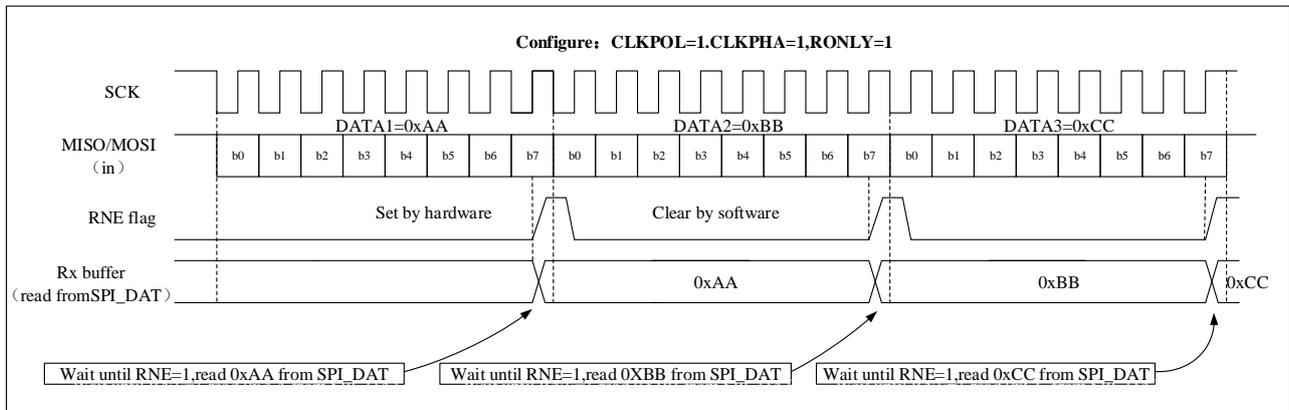
当 SPI\_CTRL2.SPIEN = 1，开始接收过程。来自 MISO 引脚的数据位依次连续移位进入移位寄存器，然后并行传送数据到 SPI\_DAT 寄存器。软件操作流程如下（见图 20-7）：

1. 设置 SPI\_CTRL1.ONLY = 1，使能仅接收模式；
2. 主机模式下，设置 SPIEN 位为 1，使能 SPI 模块，SCLK 信号会立即产生，在 SPI 关闭前 (SPIEN=0)，数据连续被接收。从机模式下，当主设备驱动 NSS 信号低电平并且产生 SCLK，数据持续被接收；
3. 等待 SPI\_STS.RNE 位置 1，读取 SPI\_DAT 寄存器获得接收的数据，当读取 SPI\_DAT 寄存器，

SPI\_STS.RNE 位将会清除。重复这个操作接收所有数据。

数据处理可以在 SPI\_STS.RNE 标志位产生的中断处理程序里实现。

图 20-7 只接收模式 (BIDIRMODE=0 并且 RONLY=1) 下连续传输时, RNE 变化示意图



■ 主机单线双向发送模式 (SPI\_CTRL1.MSEL = 1, SPI\_CTRL1.BIDIRMODE = 1, SPI\_CTRL1.BIDIROEN = 1, SPI\_CTRL1.RONLY = 0)

数据写进 SPI\_DAT 寄存器后, 传输过程开始。这个模式不接收数据。发送第一个数据位的同时, 被发送的数据并行装载进移位寄存器, 然后根据 LSBFF 位的配置, SPI 按照 MSB 或 LSB 顺序将数据位串行移位到 MOSI 引脚。

主机单线双向发送的软件操作流程和仅发送模式的流程相同。

■ 主机单线双向接收模式 (SPI\_CTRL1.MSEL = 1, SPI\_CTRL1.BIDIRMODE = 1, SPI\_CTRL1.BIDIROEN = 0, SPI\_CTRL1.RONLY = 0)

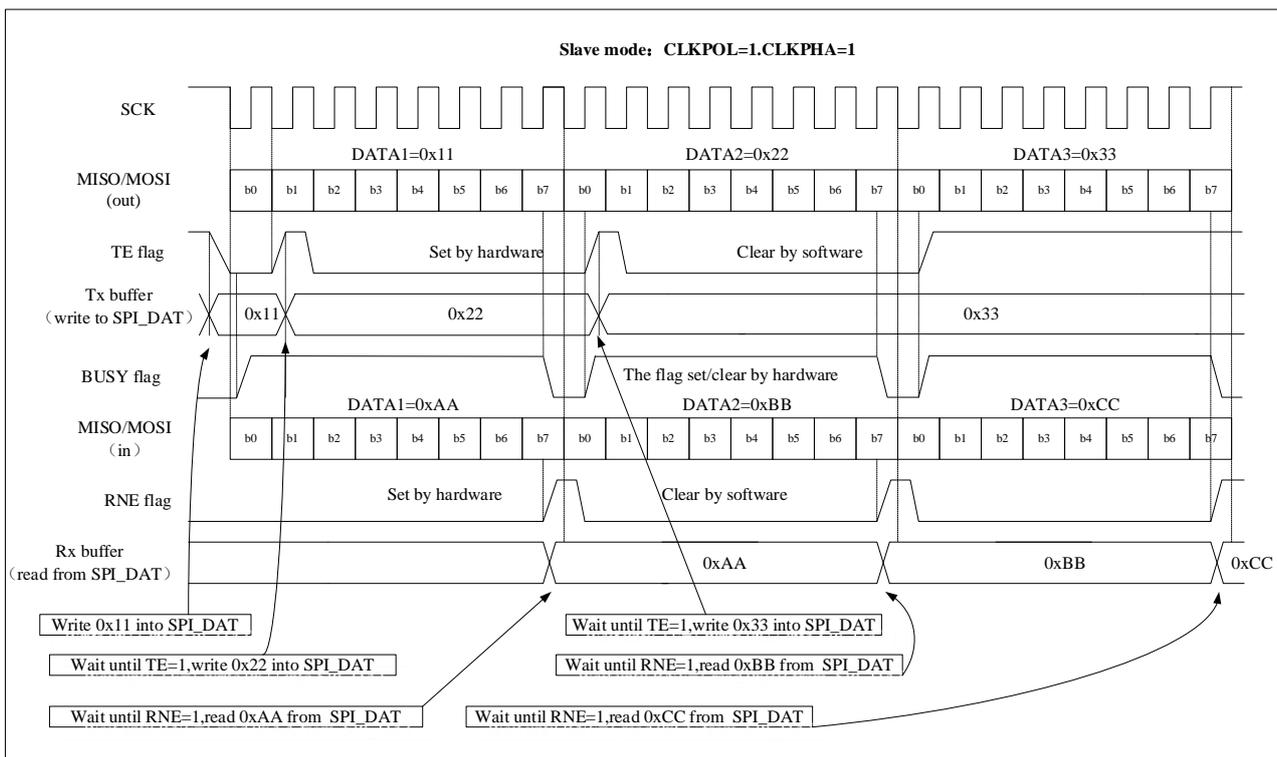
该模式下, 当 SPI 使能 (SPI\_CTRL2.SPIEN = 1), 接收过程开始。该模式下, 没有数据输出, 接收到的数据位顺序且连续移位进入移位寄存器, 并行的传输进 SPI\_DAT 寄存器 (接收缓存)。

主机单线双向接收模式的软件操作流程和仅接收模式一样。

■ 从机全双工模式 (SPI\_CTRL1.MSEL = 0, SPI\_CTRL1.BIDIRMODE = 0, SPI\_CTRL1.RONLY = 0)

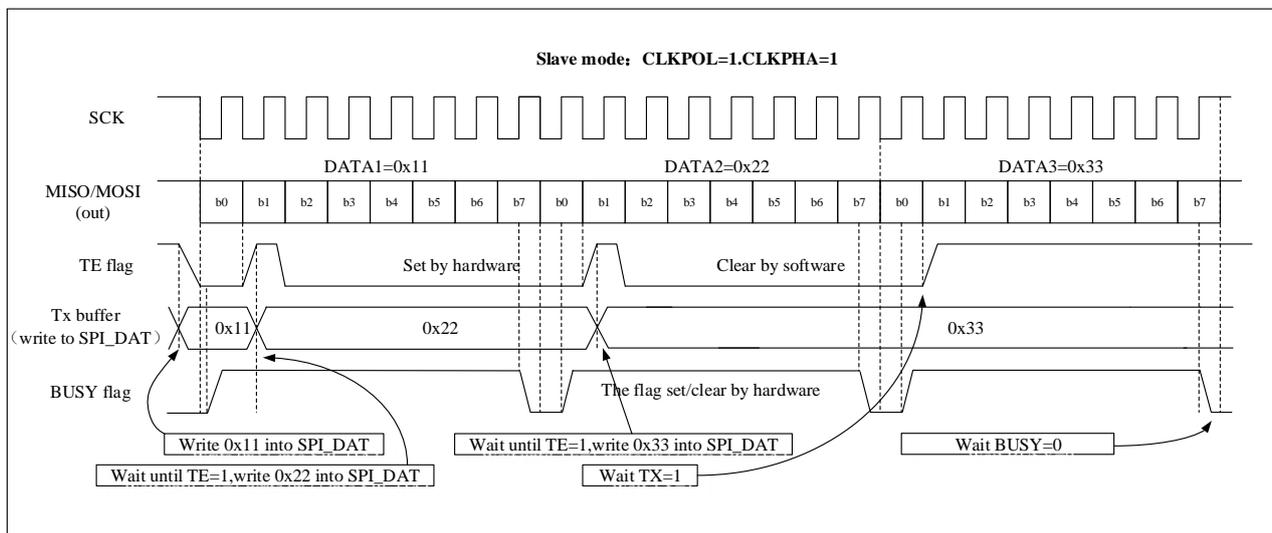
当从设备接收到第一个时钟沿, 数据传输过程开始。主设备开始数据传输之前, 软件必须确保待发送的数据写入 SPI\_DAT 寄存器。

图 20-8 从机全双工模式下连续传输时，SPI\_STS.TE/RNE/BUSY 的变化示意图



■ 从机双线单向仅发送模式 (SPI\_CTRL1.MSEL = 0, SPI\_CTRL1.BIDIRMODE = 0, SPI\_CTRL1.ROONLY = 0)

图 20-9 从机单向只发送模式下连续传输时，SPI\_STS.TE/BUSY 变化示意图



■ 从机双线单向仅接收模式 (SPI\_CTRL1.MSEL = 0, SPI\_CTRL1.BIDIRMODE = 0, SPI\_CTRL1.ROONLY = 1)

当从设备接收到时钟信号和来自 MOSI 引脚的第一个数据位，数据接收过程开始。接收到的数据位顺序且连续地串行移位到移位寄存器，然后并行地装载到 SPI\_DAT 寄存器（接收缓存）。

■ 从机单线双向发送模式 (SPI\_CTRL1.MSEL = 0, SPI\_CTRL1.BIDIRMODE = 1, SPI\_CTRL1.BIDIROEN = 1)

当从设备接收到第一个时钟沿，数据发送过程开始。该模式没有数据接收，SPI 主机开始数据传输前，软件必须确保待发送数据已经被写进 SPI\_DAT 寄存器。

■ **从机单线双向接收模式 (SPI\_CTRL1.MSEL = 0, SPI\_CTRL1.BIDIRMODE = 1, SPI\_CTRL1.BIDIROEN = 0)**

当从设备接收到第一个时钟沿和来自 MOSI 引脚的数据位时，数据接收开始。该模式没有数据输出，接收到的数据位顺序且连续地串行移位到移位寄存器，然后并行地装载到 SPI\_DAT 寄存器（接收缓存）。

*注意：从机的软件操作流程参考主机的。*

### SPI 初始化流程

1. 通过设置 SPI\_CTRL1.BR[2:0]位配置数据传输的波特率（如果工作在从模式，则忽略该步骤）；
2. 选择时钟极性（SPI\_CTRL1.CLKPOL）和时钟相位（SPI\_CTRL1.CLKPHA），定义数据传输和时钟的相位关系；
3. 设置 SPI\_CTRL1.DATFF 位定义帧格式为 8bit 还是 16bit；
4. 配置 SPI\_CTRL1.LSBFF 定义数据位发送的顺序是 LSB 还是 MSB；
5. 配置 NSS 模式；
6. 配置 SPI\_CTRL1.MSEL、SPI\_CTRL1.BIDIRMODE、SPI\_CTRL1.BIDIROEN 和 SPI\_CTRL1.ROONLY 位；
7. 设置 SPI\_CTRL2.SPIEN 位使能 SPI 模块。

### SPI 协议基本的发送和接收处理

当 SPI 发送 1 个数据帧，首先，数据帧从数据缓存装载进移位寄存器，然后装载的数据被发送。当来自发送缓存的数据传输进移位寄存器，发送缓存器为空，SPI\_STS.TE 标志位置 1，然后下一个数据可装载进入发送缓存。如果 SPI\_CTRL2.TEINTEN 位置 1，中断将会产生。写 SPI\_DAT 寄存器可以对 SPI\_STS.TE 标志位清 0。

采样时钟的最后一个边沿，当数据从移位寄存器传输进接收缓存，SPI\_STS.RNE 标志位置 1，数据准备就绪，可以从 SPI\_DAT 寄存器读取。如果 SPI\_CTRL2.RNEINTEN 标志位置 1，中断将会产生。读 SPI\_DAT 寄存器可以对 SPI\_STS.RNE 标志位清 0。

主模式下，当数据写进发送缓存，发送过程开始。当前数据帧发送完成前，如果下个数据写进 SPI\_DAT 寄存器，连续发送可以实现。

从机模式下，NSS 引脚为低，当第一个时钟沿到来，发送过程开始。为了避免意外的数据传输，数据发送前（主机发送时钟前，建议先使能 SPI 模块）软件必须写数据到发送缓存。

在有些配置里，当发送最后数据时，SPI\_STS.BUSY 标志位可以用于等待数据发送结束。

### 连续和非连续传输

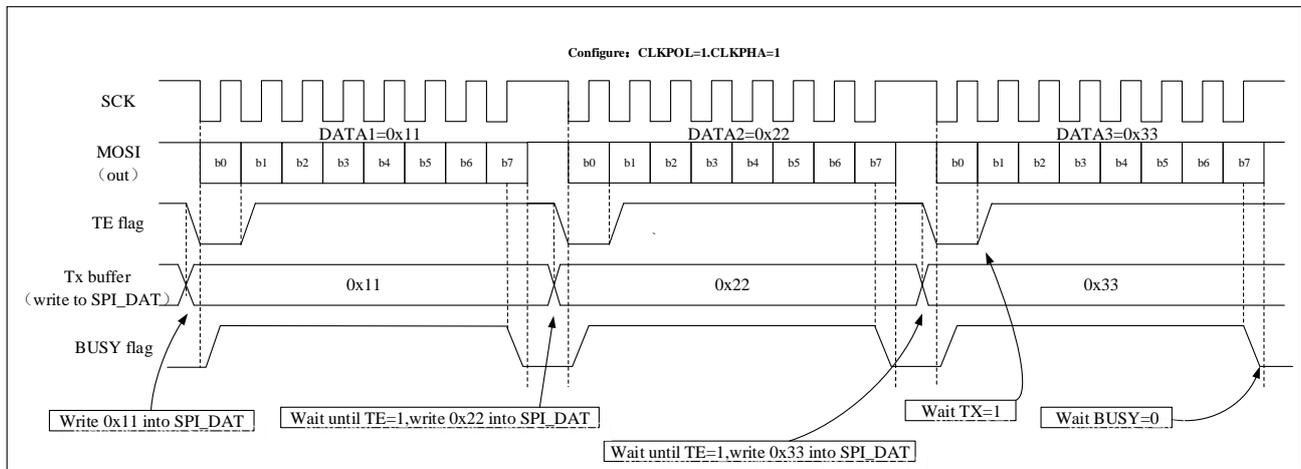
当主模式下发送数据，如果软件足够快，检测到每个 TE 上升沿（或 TE 中断），且正进行的传输结束前，立即将数据写入 SPI\_DAT 寄存器，此时，每个数据项之间的 SPI 时钟保持连续，SPI\_STS.BUSY 标志位将不会被清除，连续通讯可以实现。

如果软件不够快，将导致不连续的通讯，此时，每个数据传输之间 SPI\_STS.BUSY 标志位会被清除（图 20-10 BIDIRMODE = 0, ROONLY = 0 非连续传输发送时，SPI\_STS.TE/BUSY 变化示意图）。

主机仅接收模式下（SPI\_CTRL1.ROONLY = 1），通讯总是连续的，并且 BUSY 标志位总是为高。

从模式下，通讯的连续性由主设备决定，任何情况下，即使通讯是连续的，每个数据项之间，BUSY 标志位将至少有 1 个 SPI 时钟周期为低（见图 20-9）。

图 20-10 BIDIRMODE = 0, RONLY = 0 非连续传输发送时，SPI\_STS.TE/BUSY 变化示意图



### 20.3.3 状态标志

SPI\_STS 寄存器中有以下 3 个标志位，用以监视 SPI 总线的状态。

#### 发送缓存空标志位 (TE)

当发送缓存空，TE 标志位置 1，意味着可以将新数据写进 SPI\_DAT 寄存器。当发送缓存非空，该标志位将硬件清 0。

#### 接收缓存非空标志位 (RNE)

当接收缓存非空，RNE 标志位置 1，因此用户知道接收缓存有数据。读取 SPI\_DAT 寄存器后，该标志位将硬件清 0。

#### 忙标志位 (BUSY)

当传输开始，BUSY 标志位置 1，传输结束后 BUSY 标志位硬件清 0。

仅当设备在主机单线双向接收模式，当通讯进行中，BUSY 标志位将会设置为 0。

下面情况，BUSY 标志位将会清 0：

- 传输结束（主模式下连续通信的情况除外）；
- 关闭 SPI 模块（SPI\_CTRL2.SPIEN = 0）；
- 产生主模式失效（SPI\_STS.MODERR = 1）。

当通讯是不连续的：每个数据项传输之间，BUSY 标志位清 0。

当通讯是连续的：在主机模式，整个传输过程，BUSY 标志位保持为高。在从机模式，每个数据项传输之间 BUSY 标志位会有 1 个 SPI 时钟周期为低。因此不要使用 BUSY 标志位处理每个数据项的发送和接收。

### 20.3.4 关闭 SPI

为了关闭 SPI 模块，不同的操作模式需要采用不同的操作步骤：

### 在主机或从机全双工模式

1. 等待 SPI\_STS.RNE 标志位置 1，并且接收到最后一个字节；
2. 等待 SPI\_STS.TE 标志位置 1；
3. 等待 SPI\_STS.BUSY 标志位清 0；
4. 关闭 SPI 模块（SPI\_CTRL2.SPIEN = 0）。

### 在主机或从机单向发送模式

1. 向 SPI\_DAT 寄存器写完最后一个字节后，等待 SPI\_STS.TE 标志位置 1；
2. 等待 SPI\_STS.BUSY 标志位清 0；
3. 关闭 SPI 模块（SPI\_CTRL2.SPIEN = 0）。

### 在主机单向只接收模式

1. 等待倒数第二个 SPI\_STS.RNE 置 1；
2. 关闭 SPI 模块前（SPI\_CTRL2.SPIEN = 0），等待 1 个 SPI 时钟周期（使用软件延时）；
3. 进入关机模式前（或关闭 SPI 模块时钟），等待最后一个 SPI\_STS.RNE 置 1。

### 从机单向只接收模式

1. 可以在任意时间关闭 SPI 模块（SPI\_CTRL2.SPIEN = 0），并且当前传输结束后，SPI 模块将被关闭；
2. 如果想进入关机模式，进入关机模式之前（或关闭 SPI 模块时钟），必须等待 SPI\_STS.BUSY 标志位为 0。

## 20.3.5 使用 DMA 进行 SPI 通讯

用户可以选择 DMA 进行 SPI 数据传输，应用程序可以得到释放，系统效率可以大大提升。

当发送缓存 DMA 使能（SPI\_CTRL2.TDMAEN 位置 1），每次 SPI\_STS.TE 标志位置 1，会产生 DMA 请求，DMA 自动将数据写入 SPI\_DAT 寄存器，这将会清除 TE 标志位。当接收缓存 DMA 使能（SPI\_CTRL2.RDMAEN 位置 1），每次 SPI\_STS.RNE 标志位置 1，会产生 DMA 请求，DMA 自动读取 SPI\_DAT 寄存器，这将会清除 SPI\_STS.RNE 标志位。

当 SPI 仅用于数据发送，仅需要使能 SPI 的发送 DMA 通道（SPI\_CTRL2.TDMAEN 位置 1）。此时，由于接收到的数据还没有被读取，OVER 标志被设置为“1”（软件不需要注意这个标志）。

当 SPI 仅用于数据接收，仅需要使能 SPI 的接收 DMA 通道（SPI\_CTRL2.RDMAEN 位置 1）。

在发送模式，DMA 已经发送完所有待发送的数据后（DMA\_INTSTS.TXCF 标志位变为 1），SPI\_STS.BUSY 标志位可以被监视确认 SPI 通讯结束，这样可以避免当 SPI 关闭或进入停机模式，破坏最后数据的发送。因此，软件需要等待 SPI\_STS.TE 标志位置 1，并且等待 SPI\_STS.BUSY 标志位为 0。

图 20-11 使用 DMA 发送

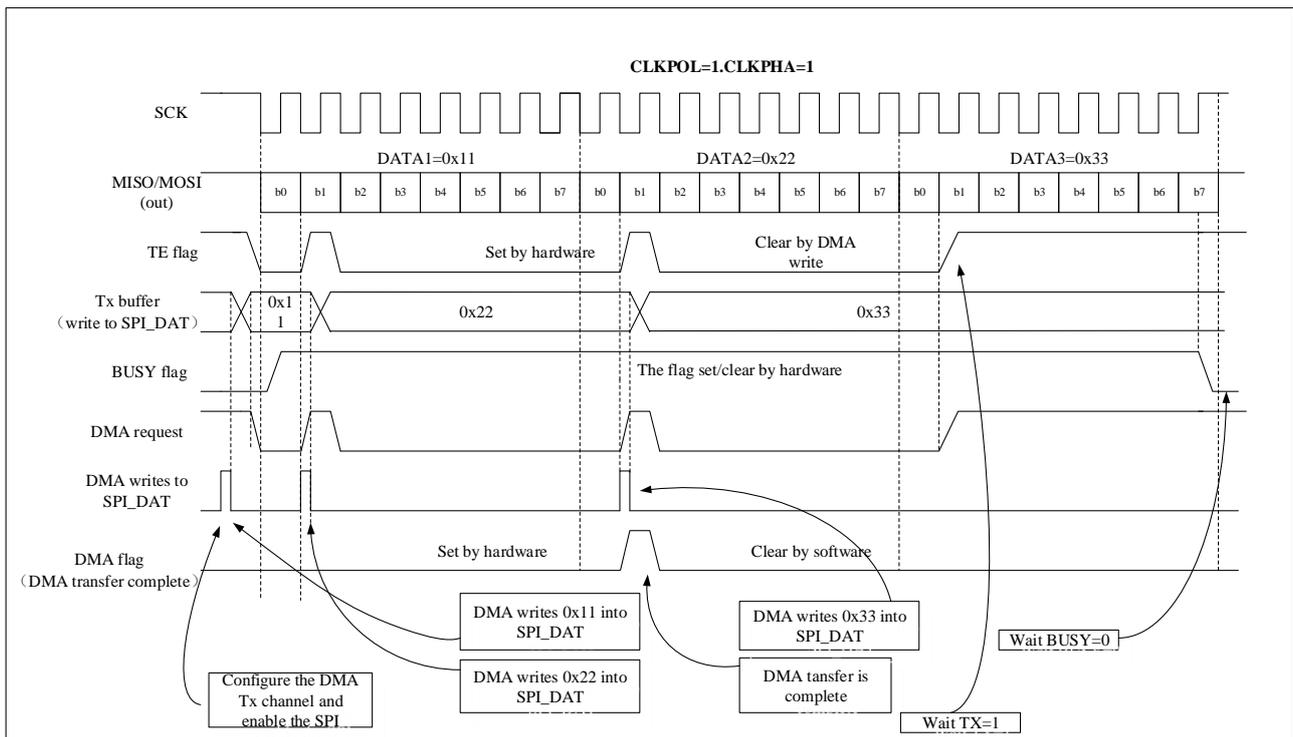
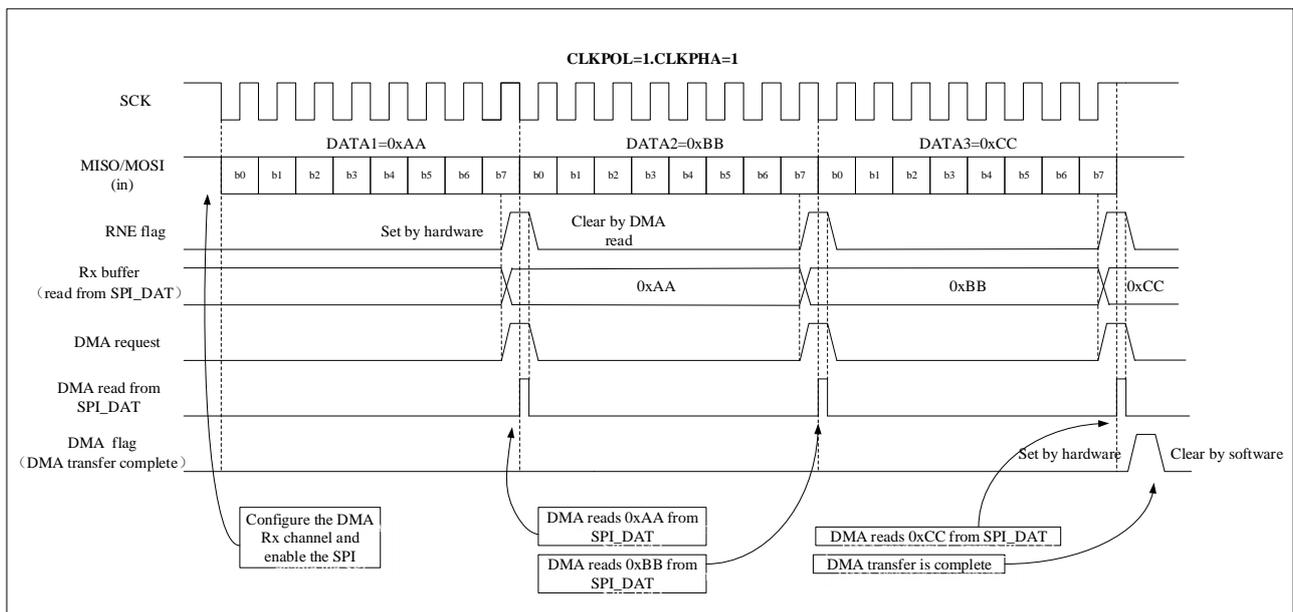


图 20-12 使用 DMA 接收



### 20.3.6 CRC 计算

SPI 包含两个独立的 CRC 计算器，用于数据发送和数据接收，以确保数据传输的正确性。根据发送和接收数据帧格式，CRC 采用不同的计算方法，8 位数据帧格式采用 CRC8，16 位数据帧格式采用 CRC16。SPI CRC 计算使用的多项式由 SPI\_CRCPOLY 寄存器设置，用户设置 SPI\_CTRL2.CRCEN 位使能 CRC 计算。

在发送模式，最后的数据写进发送缓存后，设置 SPI\_CTRL1.CRCNEXT 位为 1，这指示发送完数据后硬件

将开始发送 CRC 值 (SPI\_CRCTDAT 值)。发送 CRC 时, CRC 计算将停止。

在接收模式, 倒数第二个数据帧接收到后, 设置 SPI\_CTRL1.CRCNEXT 位为 1。接收到的 CRC 和 SPI\_CRCRDAT 值进行比较, 如果他们不同, SPI\_STS.CRCERR 位置 1, 当 SPI\_CTRL2.ERRINTEN 位置 1, 中断产生。

为了保持主设备-从设备下次的 CRC 计算结果的同步, 用户应清除主设备-从设备的 CRC 值。SPI\_CTRL2.CRCEN 位置 1 会复位 SPI\_CRCRDAT 寄存器和 SPI\_CRCTDAT 寄存器。按顺序采用下面步骤: SPI\_CTRL2.SPIEN = 0; SPI\_CTRL2.CRCEN = 0; SPI\_CTRL2.CRCEN = 1; SPI\_CTRL2.SPIEN = 1。

最重要的是, 当 SPI 配置为从模式且 CRC 使能, 只要 SCLK 引脚有时钟脉冲, 即使 NSS 引脚为高, CRC 计算将仍然被执行。这种情况常见于当主设备和多个从设备交替通讯时, 因此这需要避免 CRC 误操作。

当 SPI 硬件 CRC 检查使能 (SPI\_CTRL2.CRCEN = 1) 且 DMA 使能, 通讯结束时硬件自动完成 CRC 字节发送和接收。

## 20.3.7 错误标志位

### 主模式失效错误 (MODERR)

以下两种情况下会发生主模式失效错误:

- NSS 引脚硬件管理模式, 主设备 NSS 引脚被驱动低电平;
- NSS 引脚软件模式管理, SSEL 位被置 0。
- 当主模式失效错误发生, SPI\_STS.MODERR 标志位置 1。如果用户允许相应的中断, 则产生中断。SPI\_CTRL2.SPIEN 位和 SPI\_CTRL1.MSEL 将写保护, 且硬件清除。SPI 关闭且强制进入从模式。
- 软件执行 SPI\_STS 寄存器读或写操作, 然后写 SPI\_CTRL1 寄存器可以清除 SPI\_STS.MODERR 位 (在多主配置下, 主机的 NSS 引脚必须先拉高)。
- 通常, 从机的 SPI\_STS.MODERR 位不能设置为 1。然而, 在多主配置下, 从设备的 SPI\_STS.MODERR 位可能置位。这种情况下, SPI\_STS.MODERR 位指示存在多主冲突。中断程序可以执行复位或返回默认状态从错误状态恢复。

### 上溢标志位 (OVER)

当 SPI\_STS.RNE 位置 1, 但是仍然有数据发送进入接收缓存, 上溢错误将发生, 此时, 上溢标志 SPI\_STS.OVER 置 1。如果用户使能相应的中断, 则产生中断。所有接收到的数据丢失, 且 SPI\_DAT 寄存器仅保留之前未读的数据。

依次读 SPI\_DAT 寄存器和 SPI\_STS 寄存器可以清除 SPI\_STS.OVER 位。

### CRC 错误 (CRCERR)

CRC 错误标志用于检查接收数据的有效性。当接收到的 CRC 值和 SPI\_CRCRDAT 值不匹配, CRC 错误发生。此时, SPI\_STS.CRCERR 标志位被设置为“1”, 如果用户使能相应的中断 (SPI\_CTRL2.ERRINTEN = 1), 则产生中断。

## 20.3.8 SPI 中断

表 20-1 SPI 中断请求

中断事件	事件标志位	使能控制位
发送缓存空标志	TE	TEINTEN
接收缓存非空标志	RNE	RNEINTEN
主模式失效事件	MODERR	ERRINTEN
溢出错误	OVER	
CRC 错误标志	CRCERR	

## 20.4 SPI 寄存器描述

### 20.4.1 SPI 寄存器总览

表 20-2 SPI 寄存器总览

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0											
000h	SPI_CTRL1	Reserved																BIDROEN	BIDIRMODE	RONLY	SSMEN	SSEL	SSOEN	CRCNEXT	DATFF	LSBFF	MSEL	CLKPHA	CLKPOL	Reserved	BR[2:0]													
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
004h	SPI_CTRL2	Reserved																CRCSTP	Reserved							NSSPOL	ERRINTEN	RNEINTEN	TEINTEN	CRCEN	TDMAEN	RDMAEN	SPIEN											
	Reset Value																	0								0	0	0	0	0	0	0	0	0										
008h	SPI_STS	Reserved																								OVER	MODERR	CRCERR	BUSY	RNE	TE													
	Reset Value																									0	0	0	0	1	0													
00Ch	SPI_DAT	Reserved																DAT[15:0]																										
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
010h	SPI_CRCTDAT	Reserved																CRCTDAT[15:0]																										
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
014h	SPI_CRCRDAT	Reserved																CRCRDAT[15:0]																										
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
018h	SPI_CRCPOLY	Reserved																CRCPOLY[15:0]																										
	Reset Value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	
038h	SPI_CTRL3	Reserved																								DELAYTIME[3:0]																		
	Reset Value																									0	0	0	0															

## 20.4.2 SPI 控制寄存器 1 (SPI\_CTRL1)

地址偏移: 0x00

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIDIR MODE	BIDIR OEN	RONLY	SSMEN	SSEL	SSOEN	CRC NEXT	DATFF	LSBFF	MSEL	CLKPHA	CLKPOL	Reserved	BR[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:16	Reserved	保留, 必须保持复位值
15	BIDIRMODE	双向数据模式使能 0: 选择“双线双向”模式; 1: 选择“单线双向”模式。
14	BIDIROEN	“单线双向”模式下输出使能 该位和 SPI_CTRL1.BIDIRMODE 位一起决定“单线双向”模式的传输方向。 0: 输出禁止 (仅接收模式)。 1: 输出使能 (仅发送模式)。 在主机模式下, “单线”数据线是 MOSI 引脚, 在从机模式下, “单线”数据线是 MISO 引脚。
13	RONLY	“双线双向”模式下输出使能 该位和 SPI_CTRL1.BIDIRMODE 位一起决定“双线双向”模式的传输方向。在多个从设备的应用场景, 该位仅被访问的从设备置 1, 仅被访问的从设备可以输出, 从而避免数据线冲突。 0: 全双工 (发送和接收模式)。 1: 输出禁能 (仅接收模式)。
12	SSMEN	软件从设备管理 当 SPI_CTRL1.SSMEN 被置位时, NSS 引脚上的电平由 SPI_CTRL1.SSEL 位的值决定。 0: 禁止软件从设备管理; 1: 启用软件从设备管理。
11	SSEL	内部从设备选择 该位仅在 SPI_CTRL1.SSMEN 位置 1 时有意义。它决定了 NSS 电平, 且 NSS 引脚的 I/O 操作无效。
10	SSOEN	NSS 输出使能 0: 禁止在主模式下 NSS 输出, 该设备可以工作在主设备模式; 1: 设备开启时, 开启主模式下 NSS 输出, 该设备不能工作在主设备模式。
9	CRCNEXT	下一个发送 CRC 0: 下一个发送的值来自发送缓存。

位域	名称	描述
		1: 下一个发送的值来自发送 CRC 寄存器。 <i>注意: 在 SPI_DAT 寄存器写入最后一个数据后应立即设置该位。</i>
8	DATFF	数据帧格式 0: 使用 8 位数据帧格式进行发送/接收; 1: 使用 16 位数据帧格式进行发送/接收。 <i>注意: 只有当 SPI 禁止 (SPI_CTRL2.SPIEN = 0) 时, 才能写该位, 否则出错。</i>
7	LSBFF	帧格式 0: 先发送 MSB。 1: 先发送 LSB。 <i>注意: 通讯过程中该位不能被改变。</i>
6	MSEL	主设备选择 0: 配置为从设备; 1: 配置为主设备。 <i>注意: 当通信正在进行的时候, 不能修改该位。</i>
5	CLKPHA	时钟相位 0: 第一个时钟沿采样数据 1: 第二个时钟沿采样数据。 <i>注意: 当通信正在进行的时候, 不能修改该位。</i>
4	CLKPOL	时钟极性 0: 空闲状态时, SCLK 保持低电平; 1: 空闲状态时, SCLK 保持高电平。 <i>注意:</i> 1. 当通信正在进行的时候, 不能修改该位。 2. CLKPOL 为 1 配置对应 IO 为上拉, CLKPOL 为 0 配置对应 IO 为下拉。
3	Reserved	保留, 必需保持复位值。
2:0	BR[2:0]	波特率控制 000: $f_{CLK}/2$ 001: $f_{CLK}/4$ 010: $f_{CLK}/8$ 011: $f_{CLK}/16$ 100: $f_{CLK}/32$ 101: $f_{CLK}/64$ 110: $f_{CLK}/128$ 111: $f_{CLK}/256$ <i>注意: 当通信正在进行的时候, 不能修改这些位。</i>

### 20.4.3 SPI 控制寄存器 2 (SPI\_CTRL2)

地址偏移: 0x04

复位值: 0x0000 0000

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

Reserved													
----------	--	--	--	--	--	--	--	--	--	--	--	--	--

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved	CRCSTP	Reserved					NSSPOL	ERRINTEN	RNEINTEN	TEINTEN	CRCEN	TDMAEN	RDMAEN	SPIEN		
	rw						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位域	名称	描述
31:14	Reserved	保留，必须保持复位值
13	CRCSTP	NSS 失效后，CRC 计算控制 0：停止计算 1：在时钟存在的情况下，仍然进行计算
12:8	Reserved	保留，必须保持复位值
7	NSSPOL	NSS 极性控制 0：NSS 低电平有效； 1：NSS 高电平有效。
6	ERRINTEN	错误中断使能 当错误（SPI_STS.CRCERR、SPI_STS.OVER、SPI_STS.MODERR）产生时，该位控制是否产生中断 0：禁止错误中断； 1：允许错误中断。
5	RNEINTEN	接收缓存非空中断使能 0：禁止 RNE 中断； 1：允许 RNE 中断，当 SPI_STS.RNE 标志置位时产生中断请求。
4	TEINTEN	发送缓存空中断使能 0：禁止 TE 中断； 1：允许 TE 中断，当 SPI_STS.TE 标志置位为‘1’时产生中断请求。
3	CRCEN	硬件 CRC 校验使能 0：禁止 CRC 计算； 1：启动 CRC 计算。 该位只能用于全双工模式。 <i>注意：只有在禁止 SPI 时（SPI_CTRL2.SPIEN = 0），才能写该位，否则出错。</i>
2	TDMAEN	发送缓存 DMA 使能 当该位被设置时，TE 标志一旦被置位就发出 DMA 请求 0：禁止发送缓存 DMA； 1：启动发送缓存 DMA。
1	RDMAEN	接收缓存 DMA 使能 当该位被设置时，RNE 标志一旦被置位就发出 DMA 请求 0：禁止接收缓存 DMA； 1：启动接收缓存 DMA。

位域	名称	描述
0	SPIEN	SPI 使能 0: 禁能 SPI 模块。 1: 使能 SPI 模块。 注意: 当关闭 SPI 设备时, 请遵循 0 的流程操作。

## 20.4.4 SPI 状态寄存器 (SPI\_STS)

地址偏移: 0x08

复位值: 0x0000 0101

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved											OVER	MODERR	CRCERR	BUSY	RNE	TE
											r	r	rc_w0	r	r	r

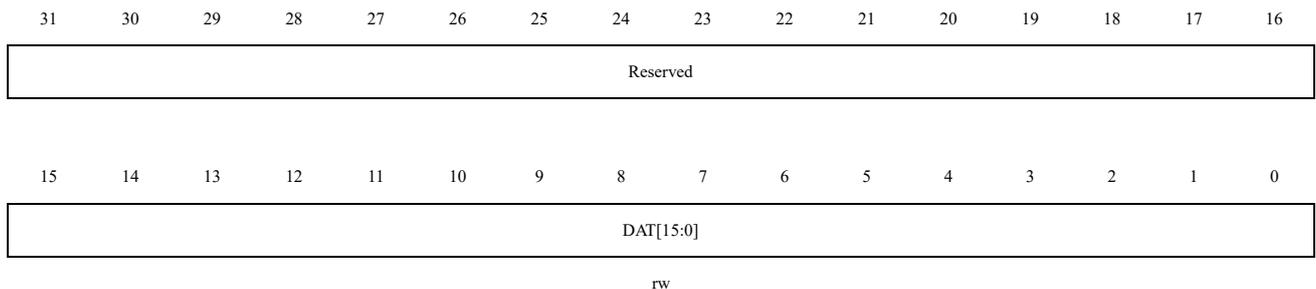
位域	名称	描述
31:6	Reserved	保留, 必须保持复位值。
5	OVER	溢出标志 0: 没有出现溢出错误; 1: 出现溢出错误。 该位由硬件置位, 由软件序列复位。关于软件序列的详细信息, 参考 20.3.7 节。
4	MODERR	模式错误 0: 没有出现模式错误; 1: 出现模式错误。 该位由硬件置位, 由软件序列复位。关于软件序列的详细信息, 参考 20.3.7 节。
3	CRCERR	CRC 错误标志 0: 收到的 CRC 值和 SPI_CRCRDAT 寄存器中的值匹配; 1: 收到的 CRC 值和 SPI_CRCRDAT 寄存器中的值不匹配。 该位由硬件置位, 由软件写'0'而复位。
2	BUSY	忙标志 0: SPI 不忙; 1: SPI 正忙于通信, 或者发送缓冲非空。 该位由硬件置位或者复位。 注意: 使用这个标志时需要特别注意, 详见第 20.3.3 和 20.3.4 章节。
1	RNE	接收缓冲非空 0: 接收缓冲为空; 1: 接收缓冲非空。
0	TE	发送缓冲为空 0: 发送缓冲非空;

位域	名称	描述
		1: 发送缓冲为空。

### 20.4.5 SPI 数据寄存器 (SPI\_DAT)

地址偏移: 0x0C

复位值: 0x0000 0000



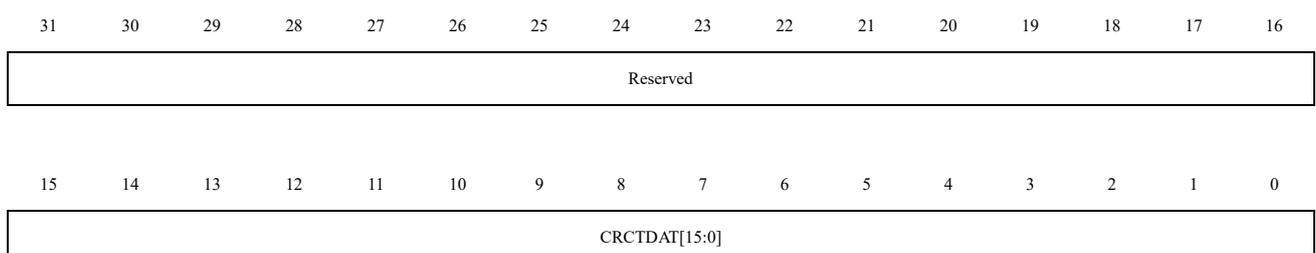
rw

位域	名称	描述
31:16	Reserved	保留, 必须保持复位值。
15:0	DAT[15:0]	数据寄存器 待发送或者已经收到的数据 数据寄存器对应两个缓存: 一个用于写 (发送缓存); 另外一个用于读 (接收缓存)。写操作将数据写到发送缓存; 读操作将返回接收缓存里的数据。 对 SPI 模式的注释: 根据 SPI_CTRL1.DATFF 位对数据帧格式的选择, 数据的发送和接收可以是 8 位或者 16 位的。为保证正确的操作, 需要在启用 SPI 之前就确定好数据帧格式。 对于 8 位的数据, 缓冲器是 8 位的, 发送和接收时只会用到 SPI_DAT[7:0]。在接收时, SPI_DAT[15:8]被强制为 0。 对于 16 位的数据, 缓冲器是 16 位的, 发送和接收时会用到整个数据寄存器, 即 SPI_DAT[15:0]。

### 20.4.6 SPI Tx CRC 寄存器 (SPI\_CRCTDAT)

地址偏移: 0x10

复位值: 0x0000



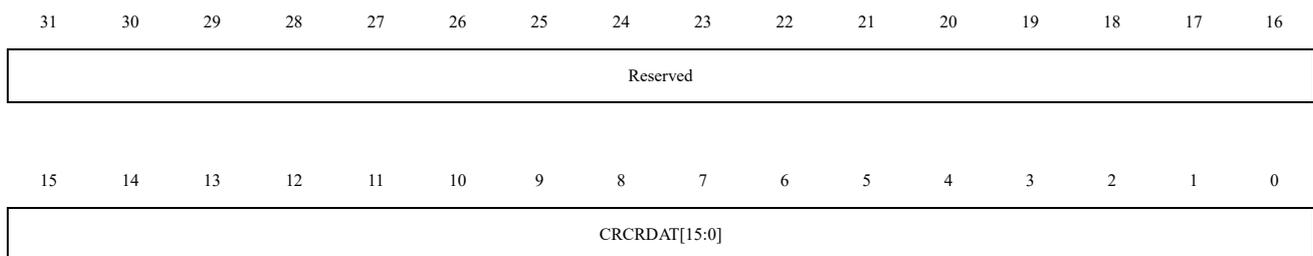
r

位域	名称	描述
31:16	Reserved	保留，必须保持复位值。
15:0	CRCTDAT	发送 CRC 寄存器 在启用 CRC 计算时，CRCTDAT[15:0]中包含了依据将要发送的字节计算的 CRC 数值。 当在 SPI_CTRL2 中的 CRCEN 位写入‘1’时，该寄存器被复位。CRC 计算使用 SPI_CRCPOLY 中的多项式。 当数据帧格式被设置为 8 位时，仅低 8 位参与计算，并且按照 CRC8 的方法进行；当数据帧格式为 16 位时，寄存器中的所有 16 个位都参与计算，并且按照 CRC16 的标准。 注意：当 BUSY 标志为‘1’时读该寄存器，将可能读到不正确的数值。

### 20.4.7 SPI Rx CRC 寄存器 (SPI\_CRCDAT)

地址偏移：0x14

复位值：0x0000



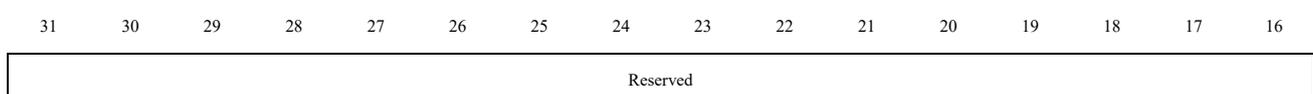
r

位域	名称	描述
31:16	Reserved	保留，必须保持复位值。
15:0	CRCDAT	接收 CRC 寄存器 在启用 CRC 计算时，CRCDAT[15:0]中包含了依据收到的字节计算的 CRC 数值。当在 SPI_CTRL2.CRCEN 位写入‘1’时，该寄存器被复位。CRC 计算使用 SPI_CRCPOLY 中的多项式。 当数据帧格式被设置为 8 位时，仅低 8 位参与计算，并且按照 CRC8 的方法进行；当数据帧格式为 16 位时，寄存器中的所有 16 位都参与计算，并且按照 CRC16 的标准。 注意：当 BUSY 标志为‘1’时读该寄存器，将可能读到不正确的数值。

### 20.4.8 SPI CRC 多项式寄存器 (SPI\_CRCPOLY)

地址偏移：0x18

复位值：0x0007



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

CRCPOLY[15:0]
---------------

rw

位域	名称	描述
31:16	Reserved	保留，必须保持复位值。
15:0	CRCPOLY [15:0]	CRC 多项式寄存器 该寄存器包含了 CRC 计算时用到的多项式。 其复位值为 0x0007，根据应用可以设置其他数值。

### 20.4.9 SPI 时钟采样延迟寄存器 (SPI\_CTRL3)

地址偏移：0x38

复位值：0x0000

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

Reserved
----------

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

Reserved	DELAYTIME[3:0]
----------	----------------

rw

位域	名称	描述
31:4	Reserved	保留，必须保持复位值。
3:0	DELAYTIME [15:0]	SPI 主机时钟延迟时间配置 SPI 作为主机模式下，用于配置 MISO 与 SCK 之间的延时，用于规避总线上带来的接收延时 4'b0000: bypass，对主机生成的时钟不做延迟处理； 4'b0001: 用 APB2 时钟下降沿对 SPI SCLK 采 1 个 PCLK2，延迟为 1/2 个 PCLK2； 4'b0010: 用 APB2 时钟下降沿对 SPI SCLK 采 1 个 PCLK2，延迟为 2/2 个 PCLK2； 4'b0011: 用 APB2 时钟下降沿对 SPI SCLK 采 2 个 PCLK2，延迟为 3/2 个 PCLK2； 4'b0100: 用 APB2 时钟下降沿对 SPI SCLK 采 2 个 PCLK2，延迟为 4/2 个 PCLK2； 4'b0101: 用 APB2 时钟下降沿对 SPI SCLK 采 3 个 PCLK2，延迟为 5/2 个 PCLK2； 4'b0110: 用 APB2 时钟下降沿对 SPI SCLK 采 3 个 PCLK2，延迟为 6/2 个 PCLK2； 4'b0111: 用 APB2 时钟下降沿对 SPI SCLK 采 4 个 PCLK2，延迟为 7/2 个 PCLK2； 4'b1000: 用 APB2 时钟下降沿对 SPI SCLK 采 4 个 PCLK2，延迟为 8/2 个 PCLK2； 4'b1001: 用 APB2 时钟下降沿对 SPI SCLK 采 5 个 PCLK2，延迟为 9/2 个 PCLK2； 4'b1010: 用 APB2 时钟下降沿对 SPI SCLK 采 5 个 PCLK2，延迟为 10/2 个 PCLK2； 4'b1011: 用 APB2 时钟下降沿对 SPI SCLK 采 6 个 PCLK2，延迟为 11/2 个 PCLK2；

位域	名称	描述
		4'b1100: 用 APB2 时钟下降沿对 SPI SCLK 采 6 个 PCLK2, 延迟为 12/2 个 PCLK2; 4'b1101: 用 APB2 时钟下降沿对 SPI SCLK 采 7 个 PCLK2, 延迟为 13/2 个 PCLK2; 4'b1110: 用 APB2 时钟下降沿对 SPI SCLK 采 7 个 PCLK2, 延迟为 14/2 个 PCLK2; 4'b1111: 用 APB2 时钟下降沿对 SPI SCLK 采 8 个 PCLK2, 延迟为 15/2 个 PCLK2; 注: 该寄存器只可在 SPI 作为主机模式下可配置, SPI 作为从机模式下配置该位无效

## 21 调试支持 (DBG)

### 21.1 简介

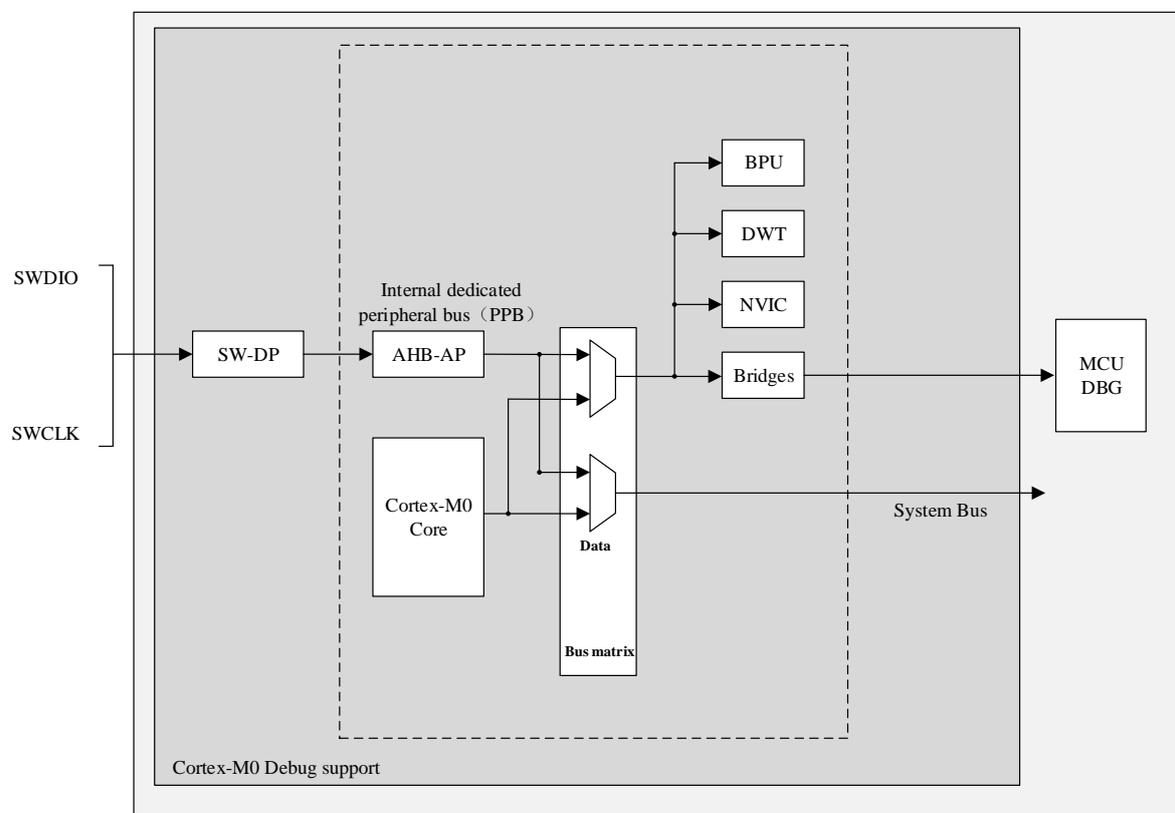
使用 Cortex®-M0 内核，内核集成硬件调试模块。支持指令断点（指令取值时停止）和数据断点（数据访问时停止）。当内核停止时，用户可以查看内核的内部状态和系统的外部状态。用户查询操作完成后，可以使内核和外设恢复，并继续执行相应程序。

内核的硬件调试模块在连接到调试器时即可被使用（在未被禁止情况下）。

支持以下调试接口：

- 串行接口

图 21-1N32G033 级别和 Cortex®-M0 级别的调试框图



ARM Cortex®-M0 内核硬件调试模块可提供如下调试功能：

- SW-DP：串行调试端口
- AHP-AP：AHB 访问端口
- BPU：断点产生
- DWT：数据触发

可参考：

- Cortex®-M0 技术参考手册 (TRM)
- ARM 调试接口 V5 结构规范

- ARM CoreSight 开发工具集（r1p0 版）技术参考手册

## 21.2 SWD 功能

调试工具可以通过上述的 SWD 调试接口来调用调试功能。

### 21.2.1 引脚分配

SWD（串行调试）接口包含 2 个管脚：SWCLK（时钟管脚）和 SWDIO（数据输入输出管脚）。

SWD 调试接口管脚分配见下表：

表 21-1 调试端口引脚

调试端口	引脚分配
SWDIO	PA13
SWCLK	PA14

## 22 唯一设备序列号 (UID)

### 22.1 简介

MCU 系列产品内置两个不同长度的唯一设备序列号，分别为 96 位的 UID(Unique device ID)和 128 位的 UCID(Unique Customer ID)，这两个设备序列号存放在闪存存储器的系统配置块中，它们所包含的信息在出厂时编写，并保证对任意一个 MCU 微控制器在任何情况下都是唯一的，用户应用程序或外部设备可以通过 CPU 或 SWD 接口读取，不可被修改。

UID 为 96 位，通常用来作为序列号或作为密码，在编写闪存时，将此唯一标识与软件加解密算法相结合，进一步提高代码在闪存存储器内的安全性，也可用于激活带安全功能的自举程序(Secure Bootloader)。

UCID 为 128 位，遵守国民技术芯片序列号定义，它包含芯片生产及版本相关信息。

除以上两个设备序列号外，还有一个 32 位的 DBGMCU\_ID，它包含了芯片版本号、芯片型号、Flash/SRAM 容量信息。

### 22.2 UID 寄存器

起始地址： 0x1FFF\_F500 长度 96 位。

### 22.3 UCID 寄存器

起始地址： 0x1FFF\_F4D0 长度 128 位。

### 22.4 DBGMCU\_ID 寄存器

起始地址：0x1FFFF510，长度 32 位。不同字节，低字节在前，高字节在后；同一字节，高位在前，低位在后。

表 22-1 DBGMCU\_ID 位描述

描述	位数	备注
芯片版本号	4bit	芯片版本号低 4 位。
	4bit	芯片版本号高 4 位。
芯片型号	4bit	设备型号的高 4 位。 设备型号由高、中、低共 12 位组成，代表芯片的型号。
	4bit	设备型号的中 4 位。
	4bit	设备型号的低 4 位。
Flash 容量	4bit	FLASH 容量指示位。 固定 64 KB
SRAM 容量	4bit	SRAM 容量指示位。 固定 6KB
系列	4bit	系列指示位。 1: G 系列

## 23 版本历史

日期	版本	修改点
2025.10.11	V1.0.0	初始版本
2025.11.27	V1.1.0	<ol style="list-style-type: none"> <li>1. FLASH 编程增加注意事项，写入数据后，需要添加 3 个 NOP</li> <li>2. 5.2.4 章节增加注意事项，使用 COMP 中断和 TIM6 中断时，关联的 EXTI 只能配置成上升沿触发。</li> <li>3. 复用寄存器增加注意事项：AF15 是 GPIO，没有复用功能</li> <li>4. GPIOB_DS.SDy 寄存器增加注意事项：其他 GPIO，VDD=5/3.3/2V 对应的驱动能力 12/6/3mA</li> <li>5. DMA 主要特性增加注意事项：不建议使用 DMA 对 FLASH 进行编程，以避免时序错误</li> <li>6. SQRT&amp;HDIV 初始化流程增加描述：SQRT_CTRLSTS.SQRTF 第一次使用默认值为 0</li> <li>7. ADC_PHSWTRIG 描述更新</li> <li>8. 17.3 COMP 配置流程删除 COMP_CTRL.OUTSEL 配置</li> <li>9. 新增表 17-1 COMP_OUT 引脚</li> <li>10. OPA 写保护取消增加通过 OPA 模块复位描述</li> <li>11. 表 19-1 IWDG 计数最大和最小复位时间更新</li> <li>12. HSI Trim 和 LSI Trim 寄存器描述添加注意事项</li> <li>13. RCC_EMCCTRL.GVDET 位添加注意事项</li> <li>14. 删除通用输出开漏模式描述，不支持</li> </ol>

## 24 声明

国民技术股份有限公司（下称“国民技术”）对此文档拥有专属产权。依据中华人民共和国的法律、条约以及世界其他法域相适用的管辖，此文档及其中描述的国民技术产品（下称“产品”）为公司所有。

国民技术在此并未授予专利权、著作权、商标权或其他任何知识产权许可。所提到或引用的第三方名称或品牌（如有）仅用作区别之目的。

国民技术保留随时变更、订正、增强、修改和改良此文档的权利，恕不另行通知。请使用者在下单购买前联系国民技术获取此文档的最新版本。

国民技术竭力提供准确可信的资讯，但即便如此，并不推定国民技术对此文档准确性和可靠性承担责任。

使用此文档信息以及生成产品时，使用者应当进行合理的设计、编程并测试其功能性和安全性，国民技术不对任何因使用此文档或本产品而产生的任何直接、间接、意外、特殊、惩罚性或衍生性损害结果承担责任。

国民技术对于产品在系统或设备中的应用效果没有任何故意或保证，如有任何应用在其发生操作不当或故障情况下，有可能致使人员伤亡、人身伤害或严重财产损失，则此类应用被视为“不安全使用”。

不安全使用包括但不限于：外科手术设备、原子能控制仪器、飞机或宇宙飞船仪器、所有类型的安全装置以及其他旨在支持或维持生命的应用。

所有不安全使用的风险应由使用人承担，同时使用人应使国民技术免于因为这类不安全使用而导致被诉、支付费用、发生损害或承担责任时的赔偿。

对于此文档和产品的任何明示、默示之保证，包括但不限于适销性、特定用途适用性和不侵权的保证，国民技术可在法律允许范围内进行免责。

未经明确许可，任何人不得以任何理由对此文档的全部或部分进行使用、复制、修改、抄录和传播。