



杭州领芯微电子有限公司

LCM32F037 用户手册

文档版本：7.7

发布日期：2023.08.18

适用产品：LCM32F037K6T8

LCM32F037H6S8

LCM32F037K6U8

LCP037A 系列

LCP037B 系列

LCP037C 系列

32 位 RISC CPU，32KB FLASH/4KB RAM，30 个快速 I/O，10 个定时器，5 个通信接口，1 个 ADC，2 个 DAC，3 个比较器，3 个运放，1.8~5.5V

主要特性

- **内核：32位ARM Cortex-M0 MCU**
 - 最高96MHz工作频率
 - 单周期乘法器
 - 硬件除法器
- **存储器**
 - 32KBytes嵌入式Flash（位宽32bit），支持预取功能和读/写保护
 - Flash 擦写次数不低于 10 万次
 - 4KBytes SRAM（位宽32bit），分为两个独立分区，每个分区2KBytes
- **复位和电源管理**
 - 1.8V到5.5V供电和I/O
 - 两个LDO，一个用于低功耗的常开/备份电源域，一个用于系统运行的内核电源域
 - 高精度上电/掉电复位（POR_PDR）
 - 可编程低压复位（LVR），8个低压复位点：1.6V、1.8V、2.0V、2.5V、2.8V、3.0V、3.5V、4.0V
 - 可编程电压监测器（LVD），8个电压监测点：2.0V、2.2V、2.4V、2.7V、2.9V、3.1V、3.6V、4.5V
- **时钟系统**
 - 4MHz到16MHz的高速晶振（OSCH）
 - 内置出厂校准过的16MHz RC振荡器（RCH，1%精度）
 - 32KHz 低速晶振（OSCL）
 - 内置出厂校准过的24KHz RC振荡器（RCL，10%精度）
 - 内置PLL，最高输出144MHz，抖动小于100ps
- **低功耗**
 - 睡眠（SLEEP）模式、停机（STOP）模式、超低功耗停机（ULP STOP）模式
- **调试模式**
 - 串行线调试口（SW-DP）
- **启动模式**
 - 支持从Flash、SRAM、System Memory启动
- **编程模式**
 - 支持串行在系统编程（ISP）
 - 支持在应用升级（IAP），提供UART、I2C、SSP接口支持
- **多达30个快速I/O端口**
 - 所有IO都可映射到16个外部中断
 - 所有IO端口均可容忍5V信号
 - 每个IO支持悬空输入/上拉输入/下拉输入/推挽输出/开漏输出/开源输出
 - 大部分IO支持一到两路模拟通道
 - 每个IO驱动能力和斜率两档可调
- **10个定时器**
 - 1个16位高级控制定时器TIM1，4个通道（带3个互补通道），支持输入捕获/输出比较/PWM输出/单脉冲输出，支持正交增量编码输入，支持死区控制和紧急刹车
 - 1个16位通用定时器TIM3，4个通道，支持输入捕获/输出比较/PWM输出/单脉冲输出，支持正交增量编码输入
 - 1个16位通用定时器TIM14，1个通道，支持输入捕获/输出比较/PWM输出/单脉冲输出
 - 1个16位通用定时器TIM15，2个通道（带2个互补通道），支持输入捕获/输出比较/PWM输出/单脉冲输出，支持死区控制和紧急刹车
 - 2个16位通用定时器TIM16/TIM17，1个通道（带1个互补通道），支持输入捕获/输出比较/PWM输出/单脉冲输出，支持死区控制和紧急刹车
 - 1个16位基本定时器TIM6
 - 1个独立看门狗定时器
 - 1个窗口看门狗定时器
 - 1个24位自减型系统时基定时器

- TIM1、TIM15、TIM16和TIM17支持延时触发和防误触发机制
- **WT钟表定时器**
 - 支持定时、周期性唤醒
- **通用DMA**
 - 4个独立通道
 - 支持的外设包括SSP、I2C、UART、ADC、DAC、Timer
- **CRC计算单元**
 - 8位、16位、32位可配置生成多项式
- **多达5个通信接口**
 - 1个I2C接口，支持主机/从机模式，支持100Kbps、400Kbps和1Mbps速率，支持7位/10位寻址模式，带FIFO和支持DMA
 - 2个UART接口，支持CTS/RTS硬流控，最高波特率为4Mbps，带FIFO和支持DMA
 - 2个SSP接口，支持主机/从机模式，支持Motorola SPI、TI SSI和National Semiconductor Microwire三种接口协议，4到16位的帧大小，最高速率达32Mbps，带FIFO和支持DMA
- **1个12位A/D转换器**
 - 最高转换速率为1.5MSPS
- 18个通道（16个外部通道，2个内部通道）
- 内置温度传感器
- 包含两个独立的采样/保持电路
- 支持内部和外部参考电压：2.5V、3.3V、4V、5V、VDDA
- **高速D/A转换器和模拟比较器（ACMP）**
 - 2个10位D/A转换器，参考电压2.5V、4V、VDDA可选
 - 2个DAC都支持硬件触发和DMA传输，支持噪声波形和三角波形生成
 - 3个ACMP，参考电压来自DAC输出、HALL_MID输出或者外部端口
- **3个运算放大器（OPA）**
 - OPA0/1/2放大倍数：
1/2/6/8/10/16/20/32
- **反电动势采样电路（HALL_MID）**
 - 三路外部电压输入采样，可用于电机控制
- **工作温度**
 - 环境温度：-40°C ~ +125°C
- **96位芯片唯一ID**

目录

主要特性	2
目录	4
图片目录	19
表格目录	24
1. 文档说明	27
2. 系统和存储简述	28
2.1 模块框图	28
2.2 32 位 M0 处理器	29
2.3 存储空间分配	29
2.3.1 简介	29
2.3.2 外设存储空间分配	30
2.4 嵌入式 SRAM	32
2.5 启动模式	32
2.6 Uid 地址	33
3. 嵌入式 32KB Flash 及选项字节说明	34
3.1 Flash 主要特性	34
3.2 Flash 功能描述	34
3.2.1 存储模块的组织形式	34
3.2.2 Flash 的读取	35
3.2.3 Flash 的烧录和擦除	35
3.3 存储器的保护	40
3.3.1 读保护	40
3.3.2 写保护	41
3.3.3 选项字节的写保护	42
3.4 Flash 相关中断	42
3.5 Flash 寄存器描述	42
3.5.1 Flash 访问控制寄存器 (FLASH_ACR)	43
3.5.2 Flash 钥匙寄存器 (FLASH_KEYR)	43
3.5.3 选项字节钥匙寄存器 (FLASH_OPTKEYR)	44
3.5.4 Flash 状态寄存器 (FLASH_SR)	44
3.5.5 Flash 控制寄存器 (FLASH_CR)	45
3.5.6 Flash 地址寄存器 (FLASH_AR)	46
3.5.7 选项字节寄存器 (FLASH_OBR)	47
3.5.8 校准寄存器 1 (FLASH_CAL1)	47
3.5.9 读保护寄存器 (FLASH_RDPR)	48
3.5.10 写保护寄存器 (FLASH_WRPR)	48
3.5.11 校准寄存器 2 (FLASH_CAL2)	49
3.5.12 校准寄存器 3 (FLASH_CAL3)	49
3.5.13 校准寄存器 4 (FLASH_CAL4)	50
3.5.14 系统存储钥匙寄存器 (FLASH_SMKEYR)	50
3.6 选项字节	51
3.6.1 用户及数据选项配置字节	51

3.6.2	用户配置选项字节	52
3.6.3	写保护选项字节	52
3.6.4	读保护选项字节	53
3.7	校准字节	57
3.7.1	出厂校准字节 0/1	57
3.7.2	出厂校准字节 2/3	57
3.7.3	出厂校准字节 4/5	58
3.7.4	出厂校准字节 6/7	58
3.7.5	出厂校准字节 8/9	59
3.7.6	出厂校准字节 10/11	60
3.7.7	出厂校准字节 12/13	60
3.7.8	出厂校准字节 14/15	61
3.7.9	出厂校准字节 16	61
3.7.10	出厂校准字节 17	62
3.7.11	出厂校准字节 18	62
3.7.12	出厂校准字节 19	62
3.7.13	出厂校准字节 20	63
4.	电源管理 (PWR)	64
4.1	供电方案	64
4.1.1	模拟电源 VDDA 供电的相关模块	64
4.1.2	电压调压器 (LDO)	65
4.2	电源检测	65
4.3	低功耗模式	65
4.3.1	降低系统时钟频率	66
4.3.2	外设时钟的关闭	66
4.3.3	睡眠模式 (SLEEP)	67
4.3.4	停机模式 (STOP)	68
4.3.5	超低功耗停机模式 (ULP STOP)	69
4.3.6	低功耗模式的 WT 唤醒	69
4.4	PWR 寄存器描述	70
4.4.1	电源配置寄存器 (PWR_CFG)	70
4.4.2	电源密钥寄存器 (PWR_KEY)	72
4.4.3	电源模式控制寄存器 (PWR_CR)	73
4.4.4	电源 LDO 配置寄存器 (PWR_LDOCR)	73
5.	时钟和复位控制 (RCC)	75
5.1	复位	75
5.1.1	电源复位	75
5.1.2	系统复位	75
5.2	时钟系统	75
5.3	寄存器描述	77
5.3.1	芯片配置寄存器组 (ChipCtrl)	77
5.3.2	系统配置寄存器组 (SysCtrl)	90
6.	系统控制单元	98
6.1	Multi-AHB 总线矩阵	98

6.2 外设互联矩阵	98
6.3 互联配置寄存器描述	99
6.3.1 互联配置寄存器 0 (SysCtrl_EDU_CFG0)	99
6.3.2 互联配置寄存器 1 (SysCtrl_EDU_CFG1)	100
6.3.3 互联配置寄存器 2 (SysCtrl_EDU_CFG2)	102
6.3.4 互联配置寄存器 3 (SysCtrl_EDU_CFG3)	103
6.3.5 互联配置寄存器 4 (SysCtrl_EDU_CFG4)	105
6.3.6 互联配置寄存器 5 (SysCtrl_EDU_CFG5)	106
6.3.7 互联配置寄存器 6 (SysCtrl_EDU_CFG6)	107
6.3.8 互联配置寄存器 7 (SysCtrl_EDU_CFG7)	108
6.3.9 系统密钥寄存器 (SysCtrl_KEY)	109
7. 通用 I/O 端口 (GPIO)	111
7.1 主要特征	111
7.2 功能描述	111
7.3 引脚说明	113
7.4 GPIO 寄存器描述	123
7.4.1 GPIO 端口模式寄存器 (GPIOx_MODER) (x=A,B,F)	124
7.4.2 GPIO 端口开漏设置/清除寄存器 (GPIOx_OD_BSRR) (x=A,B,F)	124
7.4.3 GPIO 端口斜率设置/清除寄存器 (GPIOx_SR_BSRR) (x=A,B,F)	125
7.4.4 GPIO 端口上拉设置/清除寄存器 (GPIOx_PU_BSRR) (x=A,B,F)	125
7.4.5 GPIO 端口下拉设置/清除寄存器 (GPIOx_PD_BSRR) (x=A,B,F)	126
7.4.6 GPIO 端口输入数据寄存器 (GPIOx_IDR) (x=A,B,F)	127
7.4.7 GPIO 端口输出数据寄存器 (GPIOx_ODR) (x=A,B,F)	127
7.4.8 GPIO 端口数据设置/清除寄存器 (GPIOx_BSRR) (x=A,B,F)	128
7.4.9 GPIO 端口配置锁定寄存器 (GPIOx_LCKR) (x=A,B,F)	128
7.4.10 GPIO 端口复用功能配置寄存器低位 (GPIOx_AFR1) (x=A,B,F)	129
7.4.11 GPIO 端口复用功能配置寄存器高位 (GPIOx_AFRH) (x=A,B,F)	130
7.4.12 GPIO 端口输出数据翻转寄存器 (GPIOx_BTGLR) (x=A,B,F)	131
7.4.13 GPIO 端口驱动强度设置/清除寄存器 (GPIOx_DR_BSRR) (x=A,B,F)	131
7.4.14 GPIO 端口施密特触发器输入设置/清除寄存器 (GPIOx_CS_BSRR) (x=A,B,F)	132
7.4.15 GPIO 端口开源模式设置/清除寄存器 (GPIOx_OS_BSRR) (x=A,B,F)	132
8. DMA 控制器	134
8.1 主要特性	134
8.2 DMA 功能描述	135
8.2.1 DMA 处理	135
8.2.2 DMA 仲裁	136
8.2.3 DMA 中断	136
8.2.4 DMA 请求与数据传输	136
8.3 DMA 寄存器描述	138
8.3.1 全局配置和使能寄存器	140
8.3.2 通道寄存器	141
8.3.3 中断寄存器	144
8.4 DMA 请求映射	156
8.5 DMA 参考编程流程	157

9. 中断和事件	158
9.1 嵌套向量中断控制器 (NVIC)	158
9.1.1 NVIC 主要特性	158
9.1.2 中断和异常向量	158
9.2 外部中断/事件控制器 (EXTI)	159
9.2.1 EXTI 主要特性	159
9.2.2 EXTI 事件管理	160
9.2.3 EXTI 功能说明	160
9.2.4 EXTI 外部和内部中断/事件线的映射	161
9.3 EXTI 寄存器描述	161
9.3.1 EXTI 中断屏蔽寄存器 (EXTI_IMR)	162
9.3.2 EXTI 事件屏蔽寄存器 (EXTI_EMR)	162
9.3.3 EXTI 上升沿触发选择寄存器 (EXTI_RTSR)	162
9.3.4 EXTI 下降沿触发选择寄存器 (EXTI_FTSR)	163
9.3.5 EXTI 软件中断事件寄存器 (EXTI_SWIER)	163
9.3.6 EXTI 挂起寄存器 (EXTI_PR)	164
9.3.7 EXTI 滤波寄存器 (EXTI_FR)	164
9.3.8 EXTI 控制寄存器 (EXTI_CR)	164
9.3.9 系统 EXTI 配置寄存器 0 (SysCtrl_EXTI_CFG0)	165
9.3.10 系统 EXTI 配置寄存器 1 (SysCtrl_EXTI_CFG1)	166
10. 模数转换器 (ADC)	167
10.1 ADC 主要特性	167
10.2 ADC 功能描述	168
10.2.1 ADC 开关控制	168
10.2.2 ADC 时钟	169
10.2.3 ADC 寄存器配置	169
10.2.4 ADC 通道选择	169
10.2.5 ADC 可编程采样时间 (SMP)	171
10.2.6 ADC 可编程转换时间 (CONVERT)	171
10.2.7 ADC 转换模式	172
10.2.8 启动转换 (ADSTART)	173
10.2.9 停止转换 (ADSTOP)	174
10.3 ADC 外部触发和触发极性	174
10.3.1 转换结束, 采样阶段结束 (EOC 和 EOSMP 标志)	176
10.3.2 序列转换结束 (EOSEQ 标志)	176
10.3.3 示例时序图 (单次/连续/断续模式, 硬件/软件触发)	176
10.4 ADC 转化的数据管理	181
10.4.1 数据寄存器与数据对齐	181
10.4.2 FIFO	181
10.4.3 不使用 DMA 时管理转换数据	181
10.4.4 使用 DMA 时管理转换数据	181
10.5 低功耗特性	181
10.5.1 自动延迟转换模式	181
10.5.2 低功耗模式	182

10.6 ADC 模拟看门狗	182
10.7 温度传感器及内部参考电压通道转换	183
10.8 ADC 中断	184
10.9 ADC 寄存器描述	185
10.9.1 ADC 中断状态寄存器 (ADC_ISR)	185
10.9.2 ADC 中断使能寄存器 (ADC_IER)	186
10.9.3 ADC 控制寄存器 (ADC_CR)	187
10.9.4 ADC 配置寄存器 1 (ADC_CFGR1)	188
10.9.5 ADC 配置寄存器 2 (ADC_CFGR2)	189
10.9.6 ADC 采样周期寄存器 (ADC_SMPR)	190
10.9.7 ADC 阈值寄存器 (ADC_TR)	191
10.9.8 ADC 通道选择寄存器 (ADC_CHSELR)	191
10.9.9 ADC 数据寄存器 (ADC_DR)	192
10.9.10 ADC 调修寄存器 (ADC_TRIM)	192
10.9.11 ADC 外部配置寄存器 (ADC_EXTCFG)	193
10.9.12 ADC 测试寄存器 (ADC_TESTCFG)	194
11. 数模转换器 (DAC)、模拟比较器 (ACMP) 和反向电动势采样控制器 (HALL_MID)	196
11.1 主要特性	196
11.1.1 DAC 特性	196
11.1.2 ACMP 特性	197
11.2 功能描述	197
11.2.1 DAC 功能描述	197
11.2.2 ACMP 功能描述	200
11.2.3 反电动势采样模块描述	205
11.3 DAC/ACMP/HALL_MID 寄存器描述	205
11.3.1 ACMP0 控制及状态寄存器 (ACMP0_CSR)	206
11.3.2 ACMP0 多路选择控制器 (ACMP0_MUXCR)	207
11.3.3 ACMP0 外部触发寄存器 (ACMP0_EXTCFG)	209
11.3.4 ACMP1 控制及状态寄存器 (ACMP1_CSR)	210
11.3.5 ACMP1 多路选择控制器 (ACMP1_MUXCR)	212
11.3.6 ACMP1 外部触发寄存器 (ACMP1_EXTCFG)	213
11.3.7 ACMP2 控制及状态寄存器 (ACMP2_CSR)	215
11.3.8 ACMP2 多路选择控制器 (ACMP2_MUXCR)	216
11.3.9 ACMP2 外部触发寄存器 (ACMP2_EXTCFG)	218
11.3.10 DAC0 控制寄存器 (DAC0_CR)	219
11.3.11 DAC0 控制及状态寄存器 (DAC0_CSR)	220
11.3.12 DAC0 数据保持寄存器 (DAC0_DHR)	221
11.3.13 DAC0 数据转换输出寄存器 (DAC0_DOR)	222
11.3.14 DAC1 控制寄存器 (DAC1_CR)	222
11.3.15 DAC1 控制及状态寄存器 (DAC1_CSR)	224
11.3.16 DAC1 数据保持寄存器 (DAC1_DHR)	224
11.3.17 DAC1 数据转换输出寄存器 (DAC1_DOR)	225
11.3.18 HALL_MID 控制寄存器 (HALL_CR)	225
11.3.19 模拟控制及状态寄存器 (ANACTRL_CSR)	226

12. 运算放大器 (OPA)	228
12.1 OPA 主要特性	228
12.2 管脚配置	228
12.3 功能描述	228
12.3.1 OPA 校准	230
12.3.2 运放 OPA0 联动与级联	230
12.3.3 运放 OPA1 联动与级联	231
12.3.4 运放 OPA2 联动与级联	231
12.3.5 OPA 运放典型配置	232
12.4 OPA 寄存器描述	233
12.4.1 OPA0 控制状态寄存器 (ANACTRL_OPA0_CSR)	233
12.4.2 OPA1 控制状态寄存器 (ANACTRL_OPA1_CSR)	235
12.4.3 OPA2 控制状态寄存器 (ANACTRL_OPA2_CSR)	236
13. 高级定时器 TIM1	239
13.1 TIM1 主要特性	239
13.2 时基单元	240
13.2.1 自动装载寄存器	241
13.2.2 预分频器	241
13.3 计数器模式	242
13.3.1 向上计数模式	242
13.3.2 向下计数模式	243
13.3.3 中央对齐模式 (向上/向下计数)	244
13.4 重复计数器	245
13.5 时钟选择	246
13.5.1 内部时钟源 (CK_INT)	247
13.5.2 外部时钟模式 1	247
13.5.3 外部时钟模式 2	248
13.6 捕获/比较通道	249
13.6.1 输入模块	250
13.6.2 输入捕获模式	250
13.6.3 PWM 输入	251
13.6.4 输出模块	252
13.6.5 强制输出模式	253
13.6.6 输出比较模式	254
13.6.7 PWM 模式	255
13.6.8 单脉冲模式	256
13.6.9 特殊情况: OCi 快速使能	257
13.6.10 互补输出和死区插入	258
13.6.11 重定向 OCiREF 到 OCi 或 OCiN	259
13.6.12 刹车功能	259
13.6.13 通过外部事件清除 OCiREF	261
13.6.14 6 步 PWM 生成	262
13.6.15 编码器接口模式	262
13.6.16 定时器输入异或功能	265

13.6.17 与霍尔传感器的接口	265
13.7 TIM1 定时器与外部触发的同步	265
13.7.1 标准触发模式	266
13.7.2 复位模式	266
13.7.3 门控模式	267
13.7.4 触发复位模式	268
13.7.5 外部时钟模式 2 及触发模式	268
13.8 TIM1 与其他定时器的联接	268
13.9 TIM1 中断	269
13.10 TIM1 寄存器描述	270
13.10.1 CR1 控制寄存器 1 (TIM1_CR1)	270
13.10.2 CR2 控制寄存器 2 (TIM1_CR2)	272
13.10.3 SMCR 从模式控制寄存器 (TIM1_SMCR)	274
13.10.4 DIER DMA 和中断控制寄存器 (TIM1_DIER)	276
13.10.5 SR 状态寄存器 (TIM1_SR)	278
13.10.6 EGR 事件产生寄存器 (TIM1_EGR)	279
13.10.7 CCMR1 捕获/比较模式寄存器 1 (TIM1_CCMR1)	280
13.10.8 CCMR2 捕获/比较模式寄存器 2 (TIM1_CCMR2)	283
13.10.9 CCER 捕获/比较使能寄存器 (TIM1_CCER)	285
13.10.10 CNT 计数寄存器 (TIM1_CNT)	287
13.10.11 PSC 预分频寄存器 (TIM1_PSC)	288
13.10.12 ARR 自动重装载寄存器 (TIM1_ARR)	288
13.10.13 RCR 重复计数寄存器 (TIM1_RCR)	289
13.10.14 CCR1 捕获/比较寄存器 1 (TIM1_CCR1)	289
13.10.15 CCR2 捕获/比较寄存器 2 (TIM1_CCR2)	290
13.10.16 CCR3 捕获/比较寄存器 3 (TIM1_CCR3)	290
13.10.17 CCR4 捕获/比较寄存器 4 (TIM1_CCR4)	291
13.10.18 BDTR 刹车和死区控制寄存器 (TIM1_BDTR)	292
13.10.19 DLAMT 触发延迟和防触发丢失寄存器 (TIM1_DLAMT)	294
13.10.20 DCR DMA 控制寄存器 (TIM1_DCR)	295
13.10.21 DMAR DMA 传输寄存器 (TIM1_DMAR)	296
13.10.22 CCTR1 比较修调寄存器 1 (TIM1_CCTR1)	296
13.10.23 CCTR2 比较修调寄存器 2 (TIM1_CCTR2)	297
13.10.24 CCTR3 比较修调寄存器 3 (TIM1_CCTR3)	297
13.10.25 CCTR4 比较修调寄存器 4 (TIM1_CCTR4)	298
13.10.26 SysCtrl_EDU_CFG1 寄存器	298
13.10.27 SysCtrl_EDU_CFG4 寄存器	298
14. 通用定时器 TIM3	299
14.1 TIM3 主要特性	299
14.2 时基单元	300
14.2.1 自动装载寄存器	300
14.2.2 预分频器	300
14.3 计数器模式	301
14.3.1 向上计数模式	301

14.3.2	向下计数模式	302
14.3.3	中央对齐模式（向上/向下计数）	303
14.4	时钟选择	304
14.4.1	内部时钟源（CK_INT）	304
14.4.2	外部时钟模式 1	305
14.4.3	外部时钟模式 2	306
14.5	捕获/比较通道	307
14.5.1	输入模块	307
14.5.2	输入捕获模式	308
14.5.3	PWM 输入	309
14.5.4	输出模块	310
14.5.5	强制输出模式	310
14.5.6	输出比较模式	311
14.5.7	PWM 模式	312
14.5.8	单脉冲模式	313
14.5.9	特殊情况：OCi 快速使能	314
14.5.10	通过外部事件清除 OCiREF	315
14.5.11	编码器接口模式	315
14.5.12	定时器输入异或功能	317
14.6	TIM3 定时器与外部触发的同步	317
14.6.1	标准触发模式	317
14.6.2	复位模式	318
14.6.3	门控模式	319
14.6.4	触发复位模式	319
14.6.5	外部时钟模式 2 及触发模式	319
14.7	TIM3 与其他定时器的联接	320
14.8	TIM3 中断	321
14.9	TIM3 寄存器描述	321
14.9.1	CR1 控制寄存器 1（TIM3_CR1）	322
14.9.2	CR2 控制寄存器 2（TIM3_CR2）	324
14.9.3	SMCR 从模式控制寄存器（TIM3_SMCR）	325
14.9.4	DIER DMA 和中断控制寄存器（TIM3_DIER）	327
14.9.5	SR 状态寄存器（TIM3_SR）	329
14.9.6	EGR 事件产生寄存器（TIM3_EGR）	330
14.9.7	CCMR1 捕获/比较模式寄存器 1（TIM3_CCMR1）	331
14.9.8	CCMR2 捕获/比较模式寄存器 2（TIM3_CCMR2）	334
14.9.9	CCER 捕获/比较使能寄存器（TIM3_CCER）	335
14.9.10	CNT 计数寄存器（TIM3_CNT）	336
14.9.11	PSC 预分频寄存器（TIM3_PSC）	337
14.9.12	ARR 自动重装载寄存器（TIM3_ARR）	337
14.9.13	CCR1 捕获/比较寄存器 1（TIM3_CCR1）	337
14.9.14	CCR2 捕获/比较寄存器 2（TIM3_CCR2）	338
14.9.15	CCR3 捕获/比较寄存器 3（TIM3_CCR3）	339
14.9.16	CCR4 捕获/比较寄存器 4（TIM3_CCR4）	339

14.9.17 DCR DMA 控制寄存器 (TIM3_DCR)	340
14.9.18 DMAR DMA 传输寄存器 (TIM3_DMAR)	341
14.9.19 SysCtrl_EDU_CFG2 寄存器	341
14.9.20 SysCtrl_EDU_CFG5 寄存器	341
15. 通用定时器 TIM14	343
15.1 TIM14 主要特性	343
15.2 时基单元	343
15.2.1 自动装载寄存器	344
15.2.2 预分频器	344
15.3 计数器模式	345
15.4 时钟选择	346
15.5 捕获/比较通道	346
15.5.1 输入捕获模式	347
15.5.2 强制输出模式	348
15.5.3 输出比较模式	348
15.5.4 PWM 模式	349
15.6 TIM14 与其他定时器的联接	350
15.7 TIM14 中断	350
15.8 TIM14 寄存器描述	350
15.8.1 CR1 控制寄存器 1 (TIM14_CR1)	351
15.8.2 CR2 控制寄存器 2 (TIM14_CR2)	352
15.8.3 DIER 中断控制寄存器 (TIM14_DIER)	353
15.8.4 SR 状态寄存器 (TIM14_SR)	353
15.8.5 EGR 事件产生寄存器 (TIM14_EGR)	354
15.8.6 CCMR1 捕获/比较模式寄存器 1 (TIM14_CCMR1)	355
15.8.7 CCER 捕获/比较使能寄存器 (TIM14_CCER)	357
15.8.8 CNT 计数寄存器 (TIM14_CNT)	358
15.8.9 PSC 预分频寄存器 (TIM14_PSC)	358
15.8.10 ARR 自动重载寄存器 (TIM14_ARR)	358
15.8.11 CCR1 捕获/比较寄存器 1 (TIM14_CCR1)	359
15.8.12 OR 选项寄存器 (TIM14_OR)	359
16. 通用定时器 TIM15/16/17	361
16.1 TIM15/16/17 主要特性	361
16.1.1 TIM15 主要特性	361
16.1.2 TIM16/17 主要特性	362
16.2 时基单元	362
16.2.1 自动装载寄存器	363
16.2.2 预分频器	363
16.3 计数器模式	364
16.3.1 向上计数模式	364
16.3.2 向下计数模式	365
16.3.3 中央对齐模式 (向上/向下计数)	366
16.4 时钟选择	367
16.4.1 内部时钟源 (CK_INT)	368

16.4.2	外部时钟源模式 1	368
16.4.3	外部时钟模式 2	369
16.5	捕获/比较通道	370
16.5.1	输入捕获模式	371
16.5.2	PWM 输入	371
16.5.3	强制输出模式	373
16.5.4	输出比较模式	373
16.5.5	PWM 模式	374
16.5.6	单脉冲模式	376
16.5.7	特殊情况：OCi 快速使能	377
16.5.8	互补输出和死区插入	377
16.5.9	重定向 OCiREF 到 OCi 或 OCiN	378
16.5.10	刹车功能	379
16.6	TIM15/16/17 定时器与外部触发的同步	380
16.6.1	标准触发模式	380
16.6.2	复位模式	381
16.6.3	门控模式	382
16.6.4	触发复位模式	383
16.6.5	外部时钟模式 2 及触发模式	383
16.7	TIM15/16/17 与其他定时器的联接	383
16.8	TIM15/16/17 中断	385
16.9	TIM15/16/17 寄存器描述	386
16.9.1	CR1 控制寄存器 1 (TIM15/16/17_CR1)	386
16.9.2	CR2 控制寄存器 2 (TIM15/16/17_CR2)	388
16.9.3	SMCR 模式控制寄存器 (TIM15/16/17_SMCR)	389
16.9.4	DIER DMA 和中断控制寄存器 (TIM15/16/17_DIER)	392
16.9.5	SR 状态寄存器 (TIM15/16/17_SR)	393
16.9.6	EGR 事件产生寄存器 (TIM15/16/17_EGR)	394
16.9.7	CCMR1 捕获/比较模式寄存器 1 (TIM15/16/17_CCMR1)	395
16.9.8	CCER 捕获/比较使能寄存器 (TIM15/16/17_CCER)	398
16.9.9	CNT 计数寄存器 (TIM15/16/17_CNT)	401
16.9.10	PSC 预分频寄存器 (TIM15/16/17_PSC)	401
16.9.11	ARR 自动重装载寄存器 (TIM15/16/17_ARR)	401
16.9.12	RCR 重复计数寄存器 (TIM15_RCR)	402
16.9.13	CCR1 捕获/比较寄存器 1 (TIM15/16/17_CCR1)	402
16.9.14	CCR2 捕获/比较寄存器 2 (TIM15_CCR2)	403
16.9.15	BDTR 刹车和死区控制寄存器 (TIM15/16/17_BDTR)	404
16.9.16	DCR DMA 控制寄存器 (TIM15/16/17_DCR)	406
16.9.17	DMAR DMA 传输寄存器 (TIM15/16/17_DMAR)	407
16.9.18	CCTR1 比较修调寄存器 1 (TIM15/16/17_CCTR1)	407
16.9.19	CCTR2 比较修调寄存器 2 (TIM15_CCTR2)	408
16.9.20	SysCtrl_EDU_CFG3 寄存器	408
16.9.21	SysCtrl_EDU_CFG5 寄存器	408
16.9.22	SysCtrl_EDU_CFG6 寄存器	408

16.9.23 SysCtrl_EDU_CFG7 寄存器	408
17. 基础定时器 TIM6	409
17.1 TIM6 主要特性	409
17.2 时基单元	409
17.2.1 自动装载寄存器	409
17.2.2 预分频器	410
17.3 计数器模式	410
17.4 时钟选择	411
17.5 TIM6 中断	412
17.6 TIM6 寄存器描述	412
17.6.1 CR1 控制寄存器 1 (TIM6_CR1)	412
17.6.2 DIER DMA 和中断控制寄存器 (TIM6_DIER)	413
17.6.3 SR 状态寄存器 (TIM6_SR)	414
17.6.4 EGR 事件产生寄存器 (TIM6_EGR)	414
17.6.5 CNT 计数寄存器 (TIM6_CNT)	415
17.6.6 PSC 预分频寄存器 (TIM6_PSC)	415
17.6.7 ARR 自动重载寄存器 (TIM6_ARR)	416
18. 定时器间的互联	417
19. 独立看门狗 (IWDG)	419
19.1 IWDG 主要特性	419
19.2 功能描述	419
19.2.1 当窗口功能使能时配置 IWDG	419
19.2.2 当窗口功能不使能时配置 IWDG	420
19.2.3 寄存器访问保护	420
19.2.4 硬件看门狗功能	420
19.2.5 调试模式	420
19.3 IWDG 寄存器描述	420
19.3.1 IWDG 密钥寄存器 (IWDG_KR)	421
19.3.2 IWDG 预分频寄存器 (IWDG_PR)	421
19.3.3 IWDG 重载寄存器 (IWDG_RLR)	422
19.3.4 IWDG 状态寄存器 (IWDG_SR)	422
19.3.5 IWDG 窗口寄存器 (IWDG_WINR)	423
20. 窗口看门狗 (WWDG)	424
20.1 WWDG 主要特性	424
20.2 功能描述	424
20.2.1 开启看门狗	424
20.2.2 控制递减计数器	425
20.2.3 提前的看门狗中断	425
20.2.4 计算 WWDG 溢出时间	425
20.2.5 调试模式	426
20.3 WWDG 寄存器描述	426
20.3.1 WWDG 控制寄存器 (WWDG_CR)	426
20.3.2 WWDG 配置寄存器 (WWDG_CFR)	427
20.3.3 WWDG 状态寄存器 (WWDG_SR)	427

21. 系统时基定时器 (SysTick)	428
21.1 主要特性	428
21.2 功能描述	428
21.3 SysTick 寄存器描述	428
21.3.1 SysTick 控制及状态寄存器 (SysTick_CTRL)	429
21.3.2 SysTick 重载值寄存器 (SysTick_RELOAD)	429
21.3.3 SysTick 当前值寄存器 (SysTick_VAL)	430
21.3.4 SysTick 校准值寄存器 (SysTick_CALIB)	430
22. 钟表定时器 (WT)	431
22.1 WT 主要特性	431
22.2 功能描述	431
22.3 WT 寄存器描述	432
22.3.1 WT 控制寄存器 (WTCON)	432
22.3.2 WT 定时寄存器 (T8/T8RL)	433
23. I2C 接口	434
23.1 I2C 主要特性	434
23.2 功能描述	434
23.2.1 操作模式	435
23.2.2 超快速模式	441
23.2.3 I2C_CLK 时钟配置	441
23.2.4 DMA 操作	443
23.3 I2C 中断	444
23.4 脉冲滤波	444
23.5 I2C 寄存器描述	445
23.5.1 I2C 控制寄存器 (IC_CON)	446
23.5.2 I2C 目标地址寄存器 (IC_TAR)	447
23.5.3 I2C 从机地址寄存器 (IC_SAR)	448
23.5.4 I2C 数据命令寄存器 (IC_DATA_CMD)	448
23.5.5 I2C 标准模式时钟高字节计数器 (IC_SS_SCL_HCNT)	449
23.5.6 I2C 标准模式时钟低字节计数器 (IC_SS_SCL_LCNT)	450
23.5.7 I2C 快速模式时钟高字节计数器 (IC_FS_SCL_HCNT)	450
23.5.8 I2C 快速模式时钟低字节计数器 (IC_FS_SCL_LCNT)	450
23.5.9 I2C 中断状态寄存器 (IC_INTR_STAT)	451
23.5.10 I2C 中断屏蔽寄存器 (IC_INTR_MASK)	451
23.5.11 I2C 中断原始状态寄存器 (IC_RAW_INTR_STAT)	452
23.5.12 I2C 接收 FIFO 阈值寄存器 (IC_RX_TL)	453
23.5.13 I2C 发送 FIFO 阈值寄存器 (IC_TX_TL)	454
23.5.14 I2C 中断清除寄存器 (IC_CLR_INTR)	454
23.5.15 I2C 清除 RX_UNDER 中断寄存器 (IC_CLR_RX_UNDER)	455
23.5.16 I2C 清除 RX_OVER 中断寄存器 (IC_CLR_RX_OVER)	455
23.5.17 I2C 清除 TX_OVER 中断寄存器 (IC_CLR_TX_OVER)	455
23.5.18 I2C 清除 RD_REQ 中断寄存器 (IC_CLR_RD_REQ)	456
23.5.19 I2C 清除 TX_ABRT 中断寄存器 (IC_CLR_TX_ABRT)	456
23.5.20 I2C 清除 RX_DONE 中断寄存器 (IC_CLR_RX_DONE)	456

23.5.21 I2C 清除 ACTIVITY 中断寄存器 (IC_CLR_ACTIVITY)	457
23.5.22 I2C 清除 STOP_DET 中断寄存器 (IC_CLR_STOP_DET)	457
23.5.23 I2C 清除 START_DET 中断寄存器 (IC_CLR_START_DET)	457
23.5.24 I2C 清除 GEN_CALL 中断寄存器 (CLR_GEN_CALL)	458
23.5.25 I2C 使能寄存器 (IC_ENABLE)	458
23.5.26 I2C 状态寄存器 (IC_STATUS)	459
23.5.27 I2C 发送 FIFO 数据量寄存器 (IC_TXFLR)	460
23.5.28 I2C 接收 FIFO 数据量寄存器 (IC_RXFLR)	460
23.5.29 I2C SDA 维持时间长度寄存器 (IC_SDA_HOLD)	461
23.5.30 I2C 发送丢弃源寄存器 (IC_TX_ABRT_SOURCE)	461
23.5.31 I2C 生成从机数据 NACK 寄存器 (IC_SLV_DATA_NACK_ONLY)	463
23.5.32 I2C DMA 控制寄存器 (IC_DMA_CR)	464
23.5.33 I2C DMA 发送数据水平寄存器 (IC_DMA_TDLR)	464
23.5.34 I2C DMA 接收数据水平寄存器 (IC_DMA_RDLR)	464
23.5.35 I2C SDA 建立寄存器 (IC_SDA_SETUP)	465
23.5.36 I2C 广呼应答寄存器 (IC_ACK_GENERAL_CALL)	465
23.5.37 I2C 使能状态寄存器 (IC_ENABLE_STATUS)	466
23.5.38 I2C 标准模式和快速模式尖峰抑制寄存器 (IC_FS_SPKLEN)	467
23.5.39 I2C 清除 RESTART_DET 寄存器 (IC_CLR_RESTART_DET)	467
24. UART 接口	468
24.1 UART 主要特性	468
24.2 功能描述	469
24.2.1 时钟	469
24.2.2 波特率	470
24.2.3 UART 帧格式	471
24.2.4 数据的发送	471
24.2.5 数据的接收	472
24.2.6 UART 硬件流控制	472
24.2.7 DMA 接口	473
24.3 UART 中断	474
24.4 UART 寄存器描述	476
24.4.1 UART0/1 数据寄存器 (UARTDR)	476
24.4.2 UART0/1 接收状态寄存器 (UARTSR)	477
24.4.3 UART0/1 标志寄存器 (UARTFR)	478
24.4.4 UART0/1 整数波特率寄存器 (UARTIBRD)	479
24.4.5 UART0/1 小数波特率寄存器 (UARTFBRD)	479
24.4.6 UART0/1 线控制寄存器 (UARTLCR_H)	479
24.4.7 UART0/1 控制寄存器 (UARTCR)	481
24.4.8 UART0/1 中断 FIFO 触发水平选择寄存器 (UARTIFLS)	482
24.4.9 UART0/1 中断屏蔽寄存器 (UARTIMSC)	482
24.4.10 UART0/1 原始中断状态寄存器 (UARTRIS)	483
24.4.11 UART0/1 屏蔽后中断状态寄存器 (UARTMIS)	484
24.4.12 UART0/1 中断清除寄存器 (UARTICR)	485
24.4.13 UART0/1 DMA 控制寄存器 (UARTDMACR)	486

25. SSP 接口.....	488
25.1 SSP 主要特征.....	488
25.2 管脚配置.....	488
25.3 功能描述.....	488
25.4 SSP 操作.....	489
25.4.1 时钟比率.....	489
25.4.2 配置 SSPCR0 控制寄存器.....	490
25.4.3 配置 SSPCR1 控制寄存器.....	498
25.4.4 DMA 接口.....	498
25.4.5 SSP 中断.....	499
25.5 SSP 寄存器描述.....	500
25.5.1 SSP0/2 控制寄存器 0 (SSPCR0).....	500
25.5.2 SSP0/2 控制寄存器 1 (SSPCR1).....	501
25.5.3 SSP0/2 数据寄存器 (SSPDR).....	502
25.5.4 SSP0/2 状态寄存器 (SSPSR).....	502
25.5.5 SSP0/2 时钟分频寄存器 (SSPCPSR).....	503
25.5.6 SSP0/2 中断屏蔽寄存器 (SSPIMSC).....	503
25.5.7 SSP0/2 原始中断状态寄存器 (SSPRIS).....	504
25.5.8 SSP0/2 屏蔽后中断状态寄存器 (SSPMIS).....	505
25.5.9 SSP0/2 中断清除寄存器 (SSPICR).....	505
25.5.10 SSP0/2 DMA 控制寄存器 (SSPDMACR).....	506
26. CRC 模块.....	507
26.1 CRC 主要特性.....	507
26.2 功能描述.....	507
26.3 CRC 寄存器描述.....	507
26.3.1 CRC 控制寄存器 (CRC_CTRL).....	507
26.3.2 CRC 数据寄存器 (CRC_DATA).....	508
26.3.3 CRC 结果寄存器 (CRC_RESULT).....	509
27. DIV 除法器.....	510
27.1 DIV 主要特性.....	510
27.2 功能描述.....	510
27.3 DIV 寄存器描述.....	510
27.3.1 DIV 控制寄存器 (DIV_CTRL).....	510
27.3.2 DIV 被除数低位寄存器 (DIV_ALO).....	511
27.3.3 DIV 被除数高位寄存器 (DIV_AHI).....	512
27.3.4 DIV 除数寄存器 (DIV_B).....	512
27.3.5 DIV 商低位寄存器 (DIV_QUOTLO).....	512
27.3.6 DIV 商高位寄存器 (DIV_QUOTHI).....	512
27.3.7 DIV 余数寄存器 (DIV_REM).....	513
28. 调试模块.....	514
28.1 主要特性.....	514
28.2 调试复用接口.....	514
28.2.1 SWD 复用接口.....	514
28.2.2 SW-DP 功能复用说明.....	515

28.2.3 SWD 管脚的内部上拉或下拉.....	515
28.3 SWD 通讯	515
28.3.1 SWD 协议介绍.....	515
28.3.2 SWD 协议时序.....	515
28.3.3 SW-DP 状态机（复位、空闲、ID 码）	516
28.3.4 DP 及 AP 读写操作.....	516
28.3.5 SW-DP 寄存器	516
28.3.6 SW-AP 寄存器.....	517
28.4 内核的调试	517
28.5 断点模块 BPU	518
28.6 数据观测点 DWT	518
28.6.1 数据监控功能说明	518
28.6.2 DWT 的程序指针采样寄存器.....	518
28.7 MCU 调试单元	518
28.7.1 低功耗模式下的调试	518
28.7.2 定时器、看门狗以等模块的调试支持.....	519
28.7.3 调试模式控制寄存器	519
29. 修订历史	521

图片目录

图 2-1 LCM32F037 模块框图	28
图 2-2 系统存储空间分配表	30
图 3-1 FLASH 编程流程	37
图 3-2 FLASH 的页擦除流程	38
图 3-3 FLASH 的块擦除操作流程	39
图 4-1 电源供电结构	64
图 4-2 上电和下电复位图	65
图 5-1 复位框图	75
图 5-2 LCM32F037 时钟框图	76
图 6-1 LCM32F037 总线矩阵	98
图 7-1 GPIO 接口的基本结构	112
图 8-1 DMA 结构框图	134
图 8-2 DMA 连接框图	135
图 8-3 DMA 中断逻辑	136
图 8-4 外设 A 配置 1 且作为目标, 2 次源的传输, 2 次目标的突发事务请求和传输	137
图 8-5 外设 A 配置 2 且作为源, 3 次源的突发事务请求和传输, 2 次目标的传输	137
图 8-6 外设 B 配置 1 且作为目标, 2 次源的传输, 1 次目标的突发事务请求和传输, 2 次目标的单次请求和传输	137
图 8-7 外设 B 配置 2 且作为源, 2 次源的突发事务请求和传输, 2 次源的单次请求和传输, 2 次目标的传输 ..	138
图 8-8 DMA 请求映射	157
图 9-1 外部中断/事件控制器框图	160
图 9-2 外部中断通用 I/O 映射	161
图 10-1 ADC 框图	168
图 10-2 使能/禁用 ADC	168
图 10-3 采样周期图	172
图 10-4 模数转换时序	174
图 10-5 停止当前的转换	174
图 10-6 ADC 外部触发	175
图 10-7 ADC 同时采样时序图 (CON=1)	177
图 10-8 ADC 顺序采样时序图 (CON=0)	177
图 10-9 顺序采样下, 单次数列转换, 软件触发	177
图 10-10 顺序采样下, 连续序列转换, 软件触发	178
图 10-11 顺序采样下, 单次数列转换, 硬件触发	178
图 10-12 顺序采样下, 断续序列转换, 软件触发	178
图 10-13 同时采样下, 单次数列转换, 软件触发	179
图 10-14 同时采样下, 连续序列转换, 软件触发	179
图 10-15 同时采样下, 单次数列转换, 硬件触发	180
图 10-16 同时采样下, 断续序列转换, 软件触发	180
图 10-17 WAIT 模式下的转换图	182
图 10-18 模拟看门狗监控区域	182
图 10-19 VTS 温度曲线图	183
图 11-1 DAC 框图	197

图 11-2 DAC LFSR 计算方式	199
图 11-3 带 LFSR 的 DAC 波形发生过程	199
图 11-4 DAC 三角波波形	199
图 11-5 三角波形转换过程（软件触发）	200
图 11-6 DAC、ACMP 和 HALL_MID 内部框图	201
图 11-7 比较器迟滞	203
图 11-8 ACMP 校准值	204
图 11-9 连续采样模式时序图	204
图 11-10 HALL_MID 框图	205
图 12-1 OPA 内部结构和外部互联	229
图 12-2 OPA 校准过程	230
图 12-3 OPA2 典型应用配置	232
图 13-1 高级定时器 TIM1 框图	240
图 13-2 定时器时基单元	241
图 13-3 预分频系数从 1 变 2 的计数器时序	242
图 13-4 向上计数溢出	243
图 13-5 计数器在二分频时钟频率下的时序图	243
图 13-6 向下计数溢出	244
图 13-7 中央对齐溢出	244
图 13-8 计数器时序图，内部时钟分频为 1，TIMX_ARR=0X6	245
图 13-9 不同模式下更新速率的例子，以及寄存器 TIM1_RCR 的设置	246
图 13-10 时钟/触发控制器框图	247
图 13-11 TI2 外部时钟框图	247
图 13-12 外部时钟模式 1 下的控制电路	248
图 13-13 外部触发时钟框图	248
图 13-14 外部时钟模式 2 下的控制电路	249
图 13-15 捕获/比较通道 1 的主电路	249
图 13-16 输入模块框图	250
图 13-17 TIM1 通道 1 的输入	251
图 13-18 PWM 输入信号测量	252
图 13-19 PWM 输入信号测量实例	252
图 13-20 输出模块框图	253
图 13-21 TIM1 通道 1 的输出	253
图 13-22 输出比较模式，翻转 OC1	254
图 13-23 TIMX_ARR=8 时边沿对齐的 PWM 波形实例	255
图 13-24 中央对齐模式 PWM（ARR=8）	256
图 13-25 单脉冲模式	257
图 13-26 带死区插入的互补输出	258
图 13-27 死区波形延迟大于负脉冲	258
图 13-28 死区波形延迟大于正脉冲	259
图 13-29 刹车对应的输出	261
图 13-30 COM 触发的生成示例	262
图 13-31 编码器接口模式示例	264
图 13-32 TI1FP1 反相的编码器接口模式示例	264

图 13-33 触发输入功能框图	265
图 13-34 标准触发模式下的控制电路	266
图 13-35 复位模式下的控制电路	267
图 13-36 门控模式下的控制电路	267
图 13-37 外部时钟模式 2+触发模式下的控制电路	268
图 14-1 通用定时器 TIM3 框图	299
图 14-2 预分频系数从 1 变 2 的计数器时序	301
图 14-3 向上计数溢出	301
图 14-4 计数器在二分频时钟频率下的时序图	302
图 14-5 向下计数溢出	302
图 14-6 中央对齐溢出	303
图 14-7 计数器时序图, 内部时钟分频为 1, TIMX_ARR=0X6	304
图 14-8 时钟/触发控制器框图	304
图 14-9 TI2 外部时钟框图	305
图 14-10 外部时钟模式 1 下的控制电路	306
图 14-11 外部触发时钟框图	306
图 14-12 外部时钟模式 2 下的控制电路	306
图 14-13 捕获/比较通道 1 的主电路	307
图 14-14 输入模块框图	308
图 14-15 TIM3 通道 1 的输入	308
图 14-16 PWM 输入信号测量	309
图 14-17 PWM 输入信号测量实例	310
图 14-18 输出模块框图	310
图 14-19 输出比较模式, 翻转 OC1	311
图 14-20 TIMX_ARR=8 时边沿对齐的 PWM 波形实例	312
图 14-21 中央对齐模式 PWM (ARR=8)	313
图 14-22 单脉冲模式	314
图 14-23 编码器接口模式示例	316
图 14-24 TI1FP1 反相的编码器接口模式示例	316
图 14-25 触发输入功能框图	317
图 14-26 标准触发模式下的控制电路	318
图 14-27 复位模式下的控制电路	318
图 14-28 门控模式下的控制电路	319
图 14-29 外部时钟模式 2+触发模式下的控制电路	320
图 15-1 通用定时器 TIM14 框图	343
图 15-2 预分频系数从 1 变 2 的计数器时序	345
图 15-3 向上计数溢出	345
图 15-4 计数器在二分频时钟频率下的时序图	346
图 15-5 内部时钟模式	346
图 15-6 捕获/比较通道 1 的主电路	347
图 15-7 输出比较模式, 翻转 OC1	349
图 15-8 TIMX_ARR=8 时边沿对齐的 PWM 波形实例	349
图 15-9 时钟测量和校准	350
图 16-1 通用定时器 TIM15 框图	362

图 16-2 通用定时器 TIM16/17 框图	362
图 16-3 定时器时基单元	363
图 16-4 预分频系数从 1 变 2 的计数器时序	364
图 16-5 向上计数溢出	365
图 16-6 计数器在二分频时钟频率下的时序图	365
图 16-7 向下计数溢出	366
图 16-8 中央对齐溢出	366
图 16-9 计数器时序图, 内部时钟分频为 1, TIMX_ARR=0X6	367
图 16-10 TI2 外部时钟框图	368
图 16-11 外部时钟模式 1 下的控制电路	369
图 16-12 外部触发时钟框图	369
图 16-13 外部时钟模式 2 下的控制电路	370
图 16-14 捕获/比较通道 1 的主电路	370
图 16-15 TIM15 通道 1 的输入	371
图 16-16 PWM 输入信号测量	372
图 16-17 PWM 输入信号测量实例	373
图 16-18 输出比较模式, 翻转 OC1	374
图 16-19 TIMX_ARR=8 时边沿对齐的 PWM 波形实例	375
图 16-20 中央对齐模式 PWM (ARR=8)	376
图 16-21 单脉冲模式	377
图 16-22 带死区插入的互补输出	378
图 16-23 死区波形延迟大于负脉冲	378
图 16-24 死区波形延迟大于正脉冲	378
图 16-25 刹车对应的输出	380
图 16-26 标准触发模式下的控制电路	381
图 16-27 复位模式下的控制电路	382
图 16-28 门控模式下的控制电路	382
图 16-29 外部时钟模式 2 且触发模式下的控制电路	383
图 17-1 基础定时器 TIM6 框图	409
图 17-2 预分频系数从 1 变 2 的计数器时序	410
图 17-3 向上计数溢出	411
图 17-4 计数器在二分频时钟频率下的时序图	411
图 17-5 内部时钟模式	412
图 19-1 独立看门狗 IWDG 框图	419
图 20-1 窗口看门狗 WWDG 逻辑图	424
图 20-2 复位产生时序图	425
图 22-1 WT 框图	431
图 23-1 I2C 框图	434
图 23-2 I2C 基本传输协议	435
图 23-3 I2C 7 位地址格式	438
图 23-4 I2C 10 位地址格式	438
图 23-5 主机发送: 发送 FIFO 空/STOP 产生时序图	439
图 23-6 主机接收: 发送 FIFO 空/STOP 产生时序图	439
图 23-7 主机发送: IC_DATA_CMD 的 RESTART 置位	439

图 23-8 主机接收: IC_DATA_CMD 的 RESTART 置位	440
图 23-9 主机发送: IC_DATA_CMD 的 STOP 置位/发送 FIFO 非空	440
图 23-10 主机发送: FIFO 为空后载入的首字节, RESTART 置位	440
图 23-11 主机接收: IC_DATA_CMD 的 STOP 置位/发送 FIFO 非空	440
图 23-12 主机接收: 允许 TX FIFO 为空/RESTART 置位后, 加载第一个命令	440
图 23-13 两个主机仲裁的时序	441
图 23-14 DMA 连续传输分解	443
图 23-15 DMA 单个和连续传输分解	444
图 23-16 尖峰抑制示例图	445
图 24-1 UART 框图	469
图 24-2 波特率分配器	470
图 24-3 UART 字节帧	471
图 24-4 两个相似设备间的硬件流控	473
图 24-5 DMA 单请求和 DMA 连续请求与 DMACLR 间的时序关系	474
图 25-1 SSP 结构框图	489
图 25-2 单字节传输时序图	491
图 25-3 多字节连续传输时序图	491
图 25-4 SPO=0, SPH=0, 单字节传输时序	492
图 25-5 SPO=0, SPH=0, 多字节传输时序	492
图 25-6 摩托罗拉 SPI 格式的传输信号序列 SPO = 0, SPH = 1, 包括单次和连续传输	493
图 25-7 摩托罗拉 SPI 格式, SPO = 1, SPH = 0 单字节传输	493
图 25-8 摩托罗拉 SPI 格式, SPO = 1, SPH = 0 多字节传输	493
图 25-9 摩托罗拉 SPI 格式的传输信号序列 SPO = 1, SPH = 1, 包括单次和连续传输	494
图 25-10 美国国家半导体 MICROWIRE 帧格式, 单次传输	495
图 25-11 美国国家半导体 MICROWIRE 帧格式, 连续传输	496
图 25-12 MICROWIRE 帧格式, SSPFSSIN 输入建立时间和保持时间要求	496
图 25-13 SSP 主机与 SSP 从机	497
图 25-14 SSP 主机和 SPI 从机	497
图 25-15 SPI 主机与 SSP 从机	498
图 25-16 DMA 传输波形	499
图 28-1 调试模块结构图	514

表格目录

表 2-1 外设存储空间	30
表 2-2 启动模式选择配置	32
表 3-1 FLASH 存储分配表	34
表 3-2 FLASH 读保护	40
表 3-3 保护等级与执行模式对照访问权限	41
表 3-4 FLASH 中断请求列表	42
表 3-5 FLASH 相关寄存器表	42
表 3-6 选项字节的格式	51
表 3-7 选项字节	51
表 4-1 模块电源分配	64
表 4-2 低功耗模式总结	66
表 4-3 立刻进入睡眠	67
表 4-4 退出后睡眠	67
表 4-5 停机模式	68
表 4-6 超低功耗停机模式	69
表 4-7 电源配置相关寄存器表	70
表 5-1 芯片配置相关寄存器表	77
表 5-2 系统配置相关寄存器表	90
表 6-1 LCM32F037 外设互联矩阵	98
表 6-2 互联配置相关寄存器表	99
表 7-1 GPIO 端口说明	112
表 7-2 引脚排列表中使用的图例/缩略语	113
表 7-3 LCM32F037 引脚定义及模拟复用功能 (AN) 映射	113
表 7-4 LCP037A 系列引脚定义	116
表 7-5 LCP037A 系列预驱引脚描述	119
表 7-6 LCP037B 系列引脚定义	119
表 7-7 LCP037B 系列预驱引脚描述	122
表 7-8 端口 A 可选复用功能 (AF) 映射	122
表 7-9 端口 B 可选复用功能 (AF) 映射	122
表 7-10 端口 F 可选复用功能 (AF) 映射	123
表 7-11 GPIO 相关寄存器表	123
表 8-1 DMA 相关寄存器表	138
表 9-1 中断和异常向量表	158
表 9-2 EXTI 相关寄存器表	161
表 10-1 ADC 外部 I/O 通道表	170
表 10-2 通道序列配置	171
表 10-3 触发极性表	175
表 10-4 外部触发表	175
表 10-5 DMA 水平触发配置表	181
表 10-6 看门狗通道选择	182
表 10-7 ADC 中断表	184
表 10-8 ADC 相关寄存器表	185

表 11-1 DAC 管脚	196
表 11-2 外部触发选择	198
表 11-3 比较器输入/输出配置	202
表 11-4 消隐配置	203
表 11-5 DAC/ACMP/HALL_MID 相关寄存器表	205
表 12-1 OPA 管脚配置	228
表 12-2 运放 OPA0 与其他模块的联动/级联关系	230
表 12-3 运放 OPA1 与其他模块的联动/级联关系	231
表 12-4 运放 OPA2 与其他模块的联动/级联关系	231
表 12-5 OPA 相关寄存器表	233
表 13-1 定时器特性比较	239
表 13-2 计数方向与编码器信号的关系	263
表 13-3 TIM1 触发与级联表	269
表 13-4 TIM1 相关寄存器表	270
表 13-5 带刹车功能的互补输出 OCI 和 OCIN 的控制	287
表 13-6 TIMX DMA 请求表	296
表 14-1 计数方向与编码器信号的关系	315
表 14-2 TIM3 触发与级联表	320
表 14-3 TIM3 相关寄存器表	321
表 14-4 TIMX DMA 请求表	341
表 15-1 TIM14 相关寄存器表	350
表 16-1 TIM15 触发与级联表	384
表 16-2 TIM16 触发与级联表	384
表 16-3 TIM17 触发与级联表	385
表 16-4 TIM15/16/17 相关寄存器表	386
表 16-5 带刹车功能的互补输出 OCI 和 OCIN 的控制	400
表 16-6 TIMX DMA 请求表	406
表 17-1 TIM6 相关寄存器表	412
表 18-1 定时器间的互联	417
表 19-1 IWDG 相关寄存器表	420
表 20-1 最小和最大超时值@36 MHz (F_{PCLK0})	426
表 20-2 WWDG 相关寄存器表	426
表 21-1 SYSTICK 相关寄存器表	428
表 22-1 WT 相关寄存器表	432
表 23-1 10 位地址的前五位含义表	438
表 23-2 具有高/低计数值的所有模式的最小 I2C_CLK 值	442
表 23-3 I2C 相关寄存器表	445
表 24-1 UART0/1 支持的功能	468
表 24-2 波特率表 (UART_CLK 为 16MHz)	470
表 24-3 波特率表 (UART_CLK 为 24MHz)	470
表 24-4 接收 FIFO 功能位	472
表 24-5 硬件流控的使能控制位	473
表 24-6 发送接收 FIFO 的 DMA 请求触发点	474
表 24-7 UART 中断请求	475

表 24-8 UART 相关寄存器表	476
表 24-9 奇偶校验位配置真值表	481
表 25-1 SSP 管脚配置	488
表 25-2 DMABREQ 的触发点，用于发送和接收 FIFO	499
表 25-3 SSP 相关寄存器表	500
表 26-1 CRC 相关寄存器表	507
表 27-1 DIV 相关寄存器表	510
表 28-1 SW 调试接口的管脚	515
表 28-2 包传输请求（8 BITS）	515
表 28-3 应答响应（3 BITS）	516
表 28-4 数据传输（33 BITS）	516
表 28-5 SW-DP 寄存器	517
表 28-6 通过输入地址 A[3:2]选择的调试端口寄存器	517
表 28-7 内核调试寄存器	518

1. 文档说明

本用户手册面向应用程序开发人员，提供了如何使用 LCM32F037 系列芯片的完整信息。LCM32F037 系列芯片具有不同封装和预驱特性。本用户手册适用于以下产品型号：

LCM32F037xxxx 系列			
产品名	可用 I/O 数量	封装特性	
LCM32F037K6T8	30	LQFP32	
LCM32F037K6U8	30	QFN32 (5*5)	
LCM32F037H6S8	22	SSOP24	
LCP037A 系列			
产品名	预驱特性	可用 I/O 数量	封装特性
LCP037AK31EU8	三相 P/N MOS 管栅极驱动	20	QFN32 (5*5)
LCP037AK31EV8		20	QFN32 (4*4)
LCP037AH31ES8		14	SSOP24
LCP037B 系列			
产品名	预驱特性	可用 I/O 数量	封装特性
LCP037BT32EU8	三相独立半桥驱动	23	QFN40
LCP037BK32EU8		16	QFN32 (5*5)
LCP037BK32ET8		16	LQFP32
LCP037C 系列			
产品名	预驱特性	可用 I/O 数量	封装特性
LCP037CC32EU8	三相独立半桥驱动	22	QFN38

有关订购信息和电气特性，请参阅相应的产品说明书。

有关 ARM Cortex-M0 内核的信息，请参阅 ARM® Cortex®-M0 技术手册。

2. 系统和存储简述

2.1 模块框图

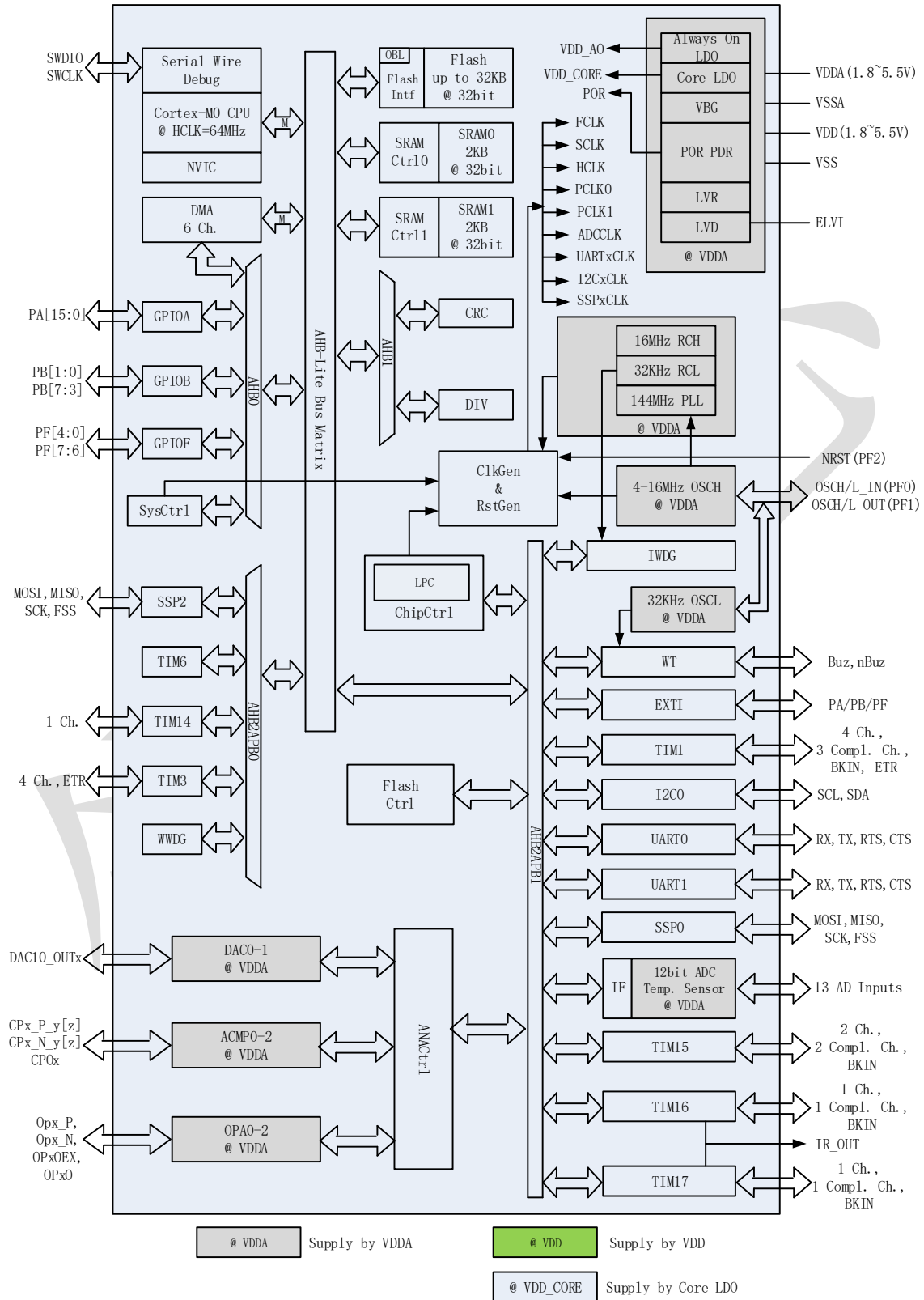


图 2-1 LCM32F037 模块框图

系统的主要结构：

- ◆ 2个主机模块
 - 32 位 CORTEX-M0 内核
 - 4 通道 DMA
- ◆ 7个从机模块
 - Flash 模块及存储总线
 - RAM 接口单元，分成两组各 2KB
 - 两组 AHB 总线，AHB0 直接控制 GPIO 及 DMA，AHB1 用于协处理单元的连接
 - 两组 APB 总线，连接到其他所有外设
- ◆ 总线仲裁单元（BusMatrix）

连接 CPU 和 DMA 两个主机模块的请求到上述的 7 个从机模块。AHB 总线作为仲裁单元及从机模块的连接总线。

- ◆ AHB转APB桥
 - 用于 APB 总线上的模块同步到 AHB 总线
 - 相关模块的地址分配参考章节 2.3.2 外设存储空间分配

注：在复位之后，所有模块（除了 FLASH 和 RAM）的时钟使能处于关闭状态，需要软件配置打开相关模块的时钟使能，详情参考寄存器 SysCtrl_ClkEnR0、SysCtrl_ClkEnR1、SysCtrl_ClkEnR2；模块还支持进行单独的复位操作，系统复位后，默认模块跟着退出复位，详情参考寄存器 SYSCTRL_SW_RSTN0、SYSCTRL_SW_RSTN1。

2.2 32 位 M0 处理器

采用 32 位 ARM Cortex-M0 处理器，它为实现 MCU 的需要提供了低成本的平台、缩减的引脚数目、降低的系统功耗、提供额外的代码效率，同时提供卓越的计算性能和先进的中断系统响应。

LCM32F037 产品系列拥有内置的 ARM 内核，因此它与市面通用的工具和软件兼容。集成的内核主要特征：

- 3 级流水线架构，可兼容 16 位指令集
- SWD 调试接口，兼容通用调试工具
- 支持单周期 32 位乘法指令
- 集成除法协处理器
- 集成系统定时单元
- 内置看门狗，包括窗式看门狗定时器

2.3 存储空间分配

2.3.1 简介

系统包含的程序区、内存（RAM）、外设及寄存器均统一分配在一个总共大小 4GB 的空间，空间地址范围 0x0000_0000 - 0xFFFF_FFFF。

采用小端（Little Endian）组织方式，即最低字节处于一个完整 32 位字的最低 8 位，最高字节处于最高 8 位。

完整地址空间分配见图 2-2：

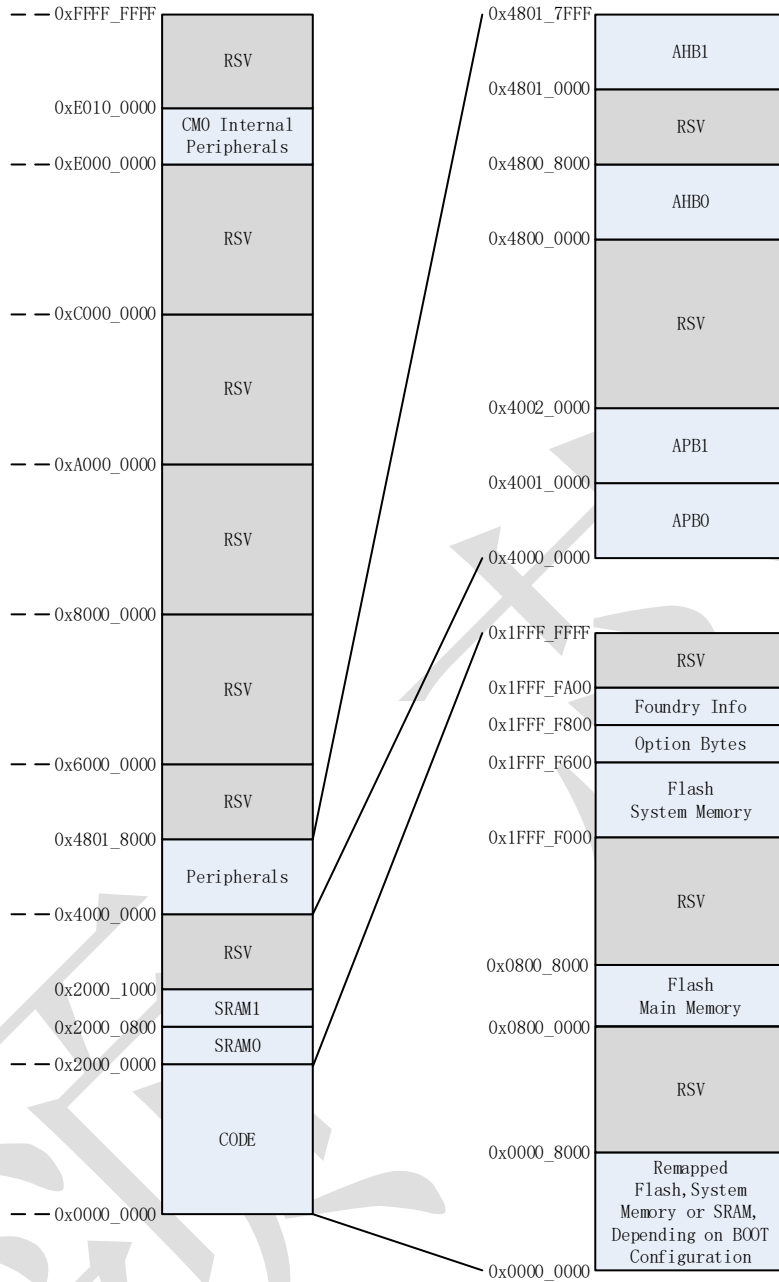


图 2-2 系统存储空间分配表

其中标识为 RSV 的部分为系统预留，禁止软件访问。

2.3.2 外设存储空间分配

表 2-1 外设存储空间

总线	地址范围	大小	外设
APB0	0x4000_0000 - 0x4000_0FFF	4KB	保留
	0x4000_1000 - 0x4000_1FFF	4KB	保留
	0x4000_2000 - 0x4000_2FFF	4KB	保留
	0x4000_3000 - 0x4000_3FFF	4KB	保留

	0x4000_4000 - 0x4000_4FFF	4KB	保留
	0x4000_5000 - 0x4000_5FFF	4KB	保留
	0x4000_6000 - 0x4000_6FFF	4KB	SSP2
	0x4000_7000 - 0x4000_7FFF	4KB	保留
	0x4000_8000 - 0x4000_8FFF	4KB	保留
	0x4000_9000 - 0x4000_9FFF	4KB	TIM6
	0x4000_A000 - 0x4000_AFFF	4KB	TIM14
	0x4000_B000 - 0x4000_BFFF	4KB	TIM3
	0x4000_C000 - 0x4000_CFFF	4KB	保留
	0x4000_D000 - 0x4000_DFFF	4KB	WWDG
	0x4000_E000 - 0x4000_EFFF	4KB	保留
	0x4000_F000 - 0x4000_FFFF	4KB	保留
APB1	0x4001_0000 - 0x4001_0FFF	4KB	TIM1
	0x4001_1000 - 0x4001_1FFF	4KB	EXTI
	0x4001_2000 - 0x4001_2FFF	4KB	保留
	0x4001_3000 - 0x4001_3FFF	4KB	I2C0
	0x4001_4000 - 0x4001_4FFF	4KB	UART0
	0x4001_5000 - 0x4001_5FFF	4KB	UART1
	0x4001_6000 - 0x4001_6FFF	4KB	保留
	0x4001_7000 - 0x4001_73FF	1KB	CHIP CTRL
	0x4001_7400 - 0x4001_77FF	1KB	IWDG
	0x4001_7800 - 0x4001_7BFF	1KB	WT
	0x4001_7C00 - 0x4001_7FFF	1KB	ANA CTRL
	0x4001_8000 - 0x4001_8FFF	4KB	SSP0
	0x4001_9000 - 0x4001_9FFF	4KB	保留
	0x4001_A000 - 0x4001_AFFF	4KB	ADC
	0x4001_B000 - 0x4001_BFFF	4KB	TIM15
	0x4001_C000 - 0x4001_CFFF	4KB	TIM16
	0x4001_D000 - 0x4001_DFFF	4KB	FLASH CTRL
	0x4001_E000 - 0x4001_EFFF	4KB	TIM17
	0x4001_F000 - 0x4001_FFFF	4KB	保留
	0x4002_0000 - 0x47FF_FFFF	~128MB	保留
AHB0	0x4800_0000 - 0x4800_01FF	512B	GPIOA
	0x4800_0200 - 0x4800_03FF	512B	GPIOB
	0x4800_0400 - 0x4800_05FF	512B	保留
	0x4800_0600 - 0x4800_07FF	512B	保留
	0x4800_0800 - 0x4800_09FF	512B	保留
	0x4800_0A00 - 0x4800_0BFF	512B	GPIOF
	0x4800_0C00 - 0x4800_0DFF	512B	保留
	0x4800_0E00 - 0x4800_0FFF	512B	保留
	0x4800_1000 - 0x4800_1FFF	4KB	保留
	0x4800_2000 - 0x4800_2FFF	4KB	保留
	0x4800_3000 - 0x4800_3FFF	4KB	保留

	0x4800_4000 - 0x4800_4FFF	4KB	DMA
	0x4800_5000 - 0x4800_5FFF	4KB	保留
	0x4800_6000 - 0x4800_6FFF	4KB	保留
	0x4800_7000 - 0x4800_7FFF	4KB	SYS CTRL
	0x4800_8000 - 0x4800_FFFF	32KB	保留
AHB1	0x4801_0000 - 0x4801_0FFF	4KB	保留
	0x4801_1000 - 0x4801_1FFF	4KB	CRC
	0x4801_2000 - 0x4801_2FFF	4KB	保留
	0x4801_3000 - 0x4801_3FFF	4KB	保留
	0x4801_4000 - 0x4801_4FFF	4KB	DIV
	0x4801_5000 - 0x4801_5FFF	4KB	保留
	0x4801_6000 - 0x4801_6FFF	4KB	保留
	0x4801_7000 - 0x4801_7FFF	4KB	保留

2.4 嵌入式 SRAM

LCM32F037 包含两个独立的 SRAM Bank, 支持同时读写。每个 Bank 为 2KBytes (位宽 32bit), 一共 4KBytes。CPU 和 DMA 模块能以 0 个等待周期对 SRAM 进行全速读写访问, 访问形式包括字节 (Byte)、半字 (Half word)、全字 (Full word)。CPU 与 DMA 可以分别同时访问不同 Bank 的 RAM, 如果发生同一时刻访问同一个 Bank 的 RAM, 则按照先后顺序占用, 后发者处于等待状态。

LCM32F037 的 RAM 不支持奇偶校验 (Parity)。

2.5 启动模式

启动时, 通过 BOOT0pin 和 BOOT selector option bit 选择三种启动模式:

- 从 Main Memory 启动
- 从 System Memory 启动
- 从 SRAM 启动

BOOT Loader 在 System memory 里, 可以通过 UART/I2C/SSP 通信接口对 Flash 进行在应用编程和升级。三种启动模式的配置方式如下:

表 2-2 启动模式选择配置

BOOT 模式配置				模式
nBOOT1 bit	BOOT0 pin	BOOT_SEL bit	nBOOT0 bit	
x	0	1	x	Main Flash memory 启动
1	1	1	x	System memory 启动
0	1	1	x	SRAM 启动
x	x	0	1	Main Flash memory 启动
1	x	0	0	System memory 启动
0	x	0	0	SRAM 启动

其中 nBOOT1 bit、BOOT_SEL bit、nBOOT0 bit 存放于选项配置字节 (Flash 空间 0x1FFF_F600), 在复位时载入到寄存器 FLASH_OBR 中, 用户可以根据需求通过烧录设备预先设置好相关的 BOOT 模式。

CPU 在开始正式执行程序时, 第一步是将地址 0x0000_0000 的值作为堆栈的顶层取出, 然后从 0x0000_0004 地址处开始运行程序。

三种启动模式: Main Flash memory 启动、SRAM 启动、System memory 启动的寻址说明如下:

- **Main Flash memory 启动:** 主存储 Flash 空间与启动程序空间重叠 (0x0000_0000 - 0x0000_7FFF), 但主存储 Flash 空间也可以通过其原先设定的地址 (0x0800_0000 - 0x0800_7FFF) 来访问。也就是说, 此时主存储 Flash 有两个访问空间, 0x0000_0000 或者 0x0800_0000, 两个地址对应同一物理地址。
- **System memory 启动:** 系统存储器被映射到启动程序空间 (0x1FFF_F000 - 0x1FFF_F5FF), 特别需要注意的是, 因设备及校准字节占用, 实际可操作系统空间的物理地址为 0x1FFF_F000 - 0x1FFF_F5BF, 启动软件 (BOOT Loader) 需小于 1.5KB。避免访问到设备及校准字节、Option Bytes 和 Foundry Info 空间, 地址结构可参考图 2-2。
- **SRAM 启动:** SRAM 存储空间映射到程序启动空间 (0x0000_0000), 原存储空间仍可以正常访问 (0x2000_0000), 与系统存储启动不同, 存放在 SRAM 的启动程序只能在实际 SRAM 大小的空间里从 0 开始被访问, 启动软件不可超出实际 SRAM 的大小。

空程序检测

在加载选项字节的过程中, 如果检测到程序存储空间 0x0800_0600 的值为 0xFFFF_FFFF (全字) 时, 认为主 Flash 程序存储区未经过编程, 选项字节加载单元设置 OB_EMPTY_FLAG 标志位为 1, 并且程序定位到 System memory 或者 RAM 空间开始执行 (由 nBOOT1 bit 确定具体位置)。

向量地址的重定位

LCM32F037 内核只有一个统一的向量寻址, 且复位、中断等向量处于 BOOT 区, 因此, 不同的引导程序启动, 可能带来向量重定义的问题。为此, 硬件设计上提供了可修改映射的寄存器, 通过设置 FLASH_CR 寄存器的 BOOT_MODE, 可动态改变 BOOT 启动区的地址映射。特别是利用 SRAM 的映射, 解决向量重映射的问题:

- 客户应用程序从 0x0800_0000 开始, 将此地址开始对应的向量表拷贝到 0x2000_0000 开始的 SRAM 区中
- 修改上述寄存器 BOOT_MODE, 将 SRAM 区映射到 0x0000_0000 BOOT 区
- 此时一旦有中断等异常发生, CPU 会从重定义后的 SRAM 向量区得到中断入口, 并跳转执行主程序 Flash 中的中断服务程序

2.6 Uid 地址

LCM32F037 内置 1 个 Uid, 位于 0x1FFF_F5B0, 长度 128 位。

3. 嵌入式 32KB Flash 及选项字节说明

3.1 Flash 主要特性

- LCM32F037 内置嵌入式 Flash 用来存储程序和数据。
- Flash 访问时间依赖于 CPU 时钟频率：CPU 时钟频率在 0 到 32MHz 之间时，0 个等待周期；在 32MHz 到 64MHz 时，1 个等待周期；在大于等于 64MHz 时，2 个等待周期。
- CPU 高速运行时，为了提高访问效率，减少等待时间，实现了 3 个 32 位的指令预取 Buffer。
- 写保护防止 Main Memory 被篡改，以 1KBytes（2 个 Page）为单位，一共有 32 个独立的写保护区间。
- 读保护分为三个保护等级；Main Memory 划分为 8 个区间，每个区间可以独立配置为读保护等级 0 或者等级 1；读保护等级 2 针对所有的 Memory 区间。
 - Level 0：没有读保护。
 - Level 1：存储读保护，不能通过 SWD 端口、SRAM/System Memory 里的程序和读保护等级为 0 的 Main Memory 区间里的程序读写 Flash。
 - Level 2：芯片读保护，debug 功能和从 SRAM/System Memory 里执行程序功能被禁止。

3.2 Flash 功能描述

3.2.1 存储模块的组织形式

FLASH 存储单元的位宽为 32-bit，总存储空间 32KB 分成 64 个 pages，每个 page 大小 512B。Flash 在功能上被划分为三个部分：

- 32KBytes 的 Main Memory，主要用于存储用户程序和数据，客户程序可以进行擦写读等操作，受读写字节保护。
- 512Bytes 的 Option Bytes，用于读写保护和用户配置信息等，处于信息区，用户无法直接进行擦写操作，可以直接读取，必须通过特殊设备和特殊流程进行设定和调整。
- 1.5KBytes 的 System Memory，用于存储 BOOT Loader、校准信息和设备配置信息等，用户原则上不能直接操作 System Memory 空间。

表 3-1 FLASH 存储分配表

程序区域	地址范围	大小	名称	写保护区 ¹	读保护区 ²
主程序区	0x0800_0000 - 0x0800_01FF	512B	Page 0	Sector0	RSector 0
	0x0800_0200 - 0x0800_03FF	512B	Page 1		
	0x0800_0400 - 0x0800_05FF	512B	Page 2	Sector1	
	0x0800_0600 - 0x0800_07FF	512B	Page 3		
	0x0800_0800 - 0x0800_09FF	512B	Page 4	Sector2	
	0x0800_0A00 - 0x0800_0BFF	512B	Page 5		
	0x0800_0C00 - 0x0800_0DFF	512B	Page 6	Sector3	
	0x0800_0E00 - 0x0800_0FFF	512B	Page7		
	⋮	⋮	⋮	⋮	⋮
	0x0800_7000 - 0x0800_71FF	512B	Page56	Sector28	RSector 7
0x0800_7200 - 0x0800_73FF	512B	Page57			

	0x0800_7400 - 0x0800_75FF	512B	Page58	Sector29
	0x0800_7600 - 0x0800_77FF	512B	Page59	
	0x0800_7800 - 0x0800_79FF	512B	Page60	Sector30
	0x0800_7A00 - 0x0800_7BFF	512B	Page61	
	0x0800_7C00 - 0x0800_7DFF	512B	Page62	Sector31
	0x0800_7E00 - 0x0800_7FFF	512B	Page63	
信息区	0x1FFF_F600 - 0x1FFF_F7FF	512B		Option Region
	0x1FFF_F000 - 0x1FFF_F5FF	1.5KB		System Memory

注1: Sector 对应写保护分区, 主程序区整体分成 32 个写保护区块, 每个区块 1KB。

注2: 读保护分成 8 个区块, 每个区块 4KB。

3.2.2 Flash 的读取

FLASH 可以直接通过 AHB 总线进行读取, 程序执行和直接的地址读操作均通过同一 AHB 总线进行。由读操作相关寄存器 (FLASH_ACR) 控制:

- 指令预取: 具有指令预驱缓存以加速 CPU 的执行速度
- 延迟: 读存储器操作的延迟等待周期设定 (0~2, 对应等待 0 到 2 个时钟周期)

指令缓存器及控制

LCM32F037 内置了 3 个 32 位的指令缓存, 采用直接映射的方式, 每个缓存可以替代对 Flash 本身的读取访问。CPU 的每次读取不超过 32 位, 读操作会更新缓存, 反馈当前有效指令数据给 CPU, 并经过设定延迟后预取 Flash 的下一个数据更新缓存器的内容。缓存器只有当设置延迟为 1 的时候能起到加速的作用, 而设置 FLASH 读取延迟为 0 则不影响系统的整体性能。

当缓存存在至少一个为空的状态时, 缓存控制器发起一个 Flash 读操作来填充缓存。

初始状态下, 缓存使能, 上电复位完成后即开始预取指令和数据。

缓存的使能一般在初始化的时候确定是否开启, 此时 CPU 的时钟频率运行于默认的内部 RCH 频率, 即为 16MHz。

预取延迟

延迟通过访问控制寄存器 FLASH_CR 的 LATENCY[2:0]来配置, 在这个配置确定的 0~2 个等待周期内, FLASH 的所有控制信号保持读状态。通过此方式, 延长取指周期, 确保访问 Flash 的时序, 保证读取数据的准确性。上电默认状态下, 此控制位为 0, 即读 Flash 无需额外等待周期。

3.2.3 Flash 的烧录和擦除

LCM32F037 内部的 Flash 可以通过内部控制线路来进行直接烧写, 也可以通过应用程序软件控制烧录。在电路编程 (ICP) 方式用于整个 Flash 模块的内容更新, 可使用 SWD 接口, 或者是通过内部 BOOTLoader 确定的接口和协议进行。ICP 模式为芯片出厂自带, 通过固定的接口和协议, 方便用户使用相关烧录工具和软件更新内部 Flash, 一般在芯片第一次烧录时, 必须使用 ICP 方式。

与 ICP 模式不同, 在系统编程 (ISP) 模式则非常灵活, 用户可以根据自己的需求使用任何通讯接口 (例如 I/O, USB, USART, I2C, SPI 等) 来进行 Flash 的更新。

在应用编程 (IAP) 模式需要占用一定的存储空间, 用户可以将程序下载到 BOOT 区 A, 并更新至 BOOT 区 B, 再从 B 区域运行程序。用户可以将此更新程序内嵌于自身的应用程序中, 且自行定义更新协议和细节。

更新控制程序第一次只能先通过 ICP 方式烧录，而且，用户需要小心控制更新协议和过程，一定要有较可靠手段防止更新模式的误触发进入，防止不可逆转的失败更新。

在本品允许的整个工作电压范围内，Flash 都可以进行正常的烧写，相关控制寄存器如下：

- FLASH 访问控制寄存器 (FLASH_ACR)
- FLASH 密钥寄存器 (FLASH_KEYR)
- 选项字节密钥寄存器 (FLASH_OPTKEYR)
- FLASH 控制寄存器 (FLASH_CR)
- FLASH 状态寄存器 (FLASH_SR)
- FLASH 地址寄存器 (FLASH_AR)
- 选项字节寄存器 (FLASH_OBR)
- 写保护寄存器 (FLASH_WRPR)
- 读保护寄存器 (FLASH_RDPR)

以下寄存器为内部使用（原厂设定）：

- 校准寄存器 1 (FLASH_CAL1)
- 校准寄存器 2 (FLASH_CAL2)
- 校准寄存器 3 (FLASH_CAL3)
- 校准寄存器 4 (FLASH_CAL4)
- 外设配置寄存器 (FLASH_DEV)
- 系统存储区密钥寄存器 (FLASH_SMKEYR)

对 FLASH 的直接操作不会中断不对 FLASH 进行访问的 CPU 操作，但如果在 FLASH 烧录过程中发生 CPU 的访问，则此访问将把总线挂起，直到当前烧录过程结束，CPU 才能得到正确的反馈。即烧录过程禁止所有的指令和数据访问。

FLASH 的烧写操作要求内部 RCH 必须处于开启状态。

FLASH 的解锁

复位后，FLASH 会处于锁定状态，以防止误烧写发生。FLASH_CR 寄存器所有位均不可访问。如需解锁 FLASH_CR 寄存器，则需通过写入特定序列的值到密钥寄存器 FLASH_KEYR：

- 写密钥 1: KEY1 = 0x4567_0123
- 写密钥 2: KEY2 = 0xCDEF_89AB

错误的写密钥操作将彻底锁定 FLASH_CR 寄存器，直至重新复位整个系统。发生序列配置错误时，检测电路产生一个总线错误，并发起硬件错误中断。配置错误包括第一次写密钥不符合法定值，以及第一次写正确，但第二次写密钥不符合前述法定值。

FLASH_CR 寄存器可以由用户软件重新锁定，锁定方式为写此寄存器对应的 LOCK 位为 1。

主存储区编程

主程序区的编程以 16bit (Half Word) 作为一个基本单位，通过设置寄存器 FLASH_CR 中的 PG 控制位为 1，一次性编程半字到 FLASH。LCM32F037 也可以接受以一个字 (WORD) 为单位的编程。

非半字或者全字编程的输入，会产生总线错误，并忽略当前烧录操作。

FLASH 存储控制单元首先读取目标地址的存储数据，并检查确认此地址是否已经被擦除。如果没有，则当前编程操作被忽略，并且设置寄存器 FLASH_SR 中的 PGERR 位为 1。

如果目标地址处于寄存器 FLASH_WRPR 规定的写保护区域，则当前编程操作被忽略，且设置寄存器 FLASH_SR 中的 WRPRERR 位为 1。编程结束后，FLASH_SR 寄存器中的 EOP 位也会被设置起来。

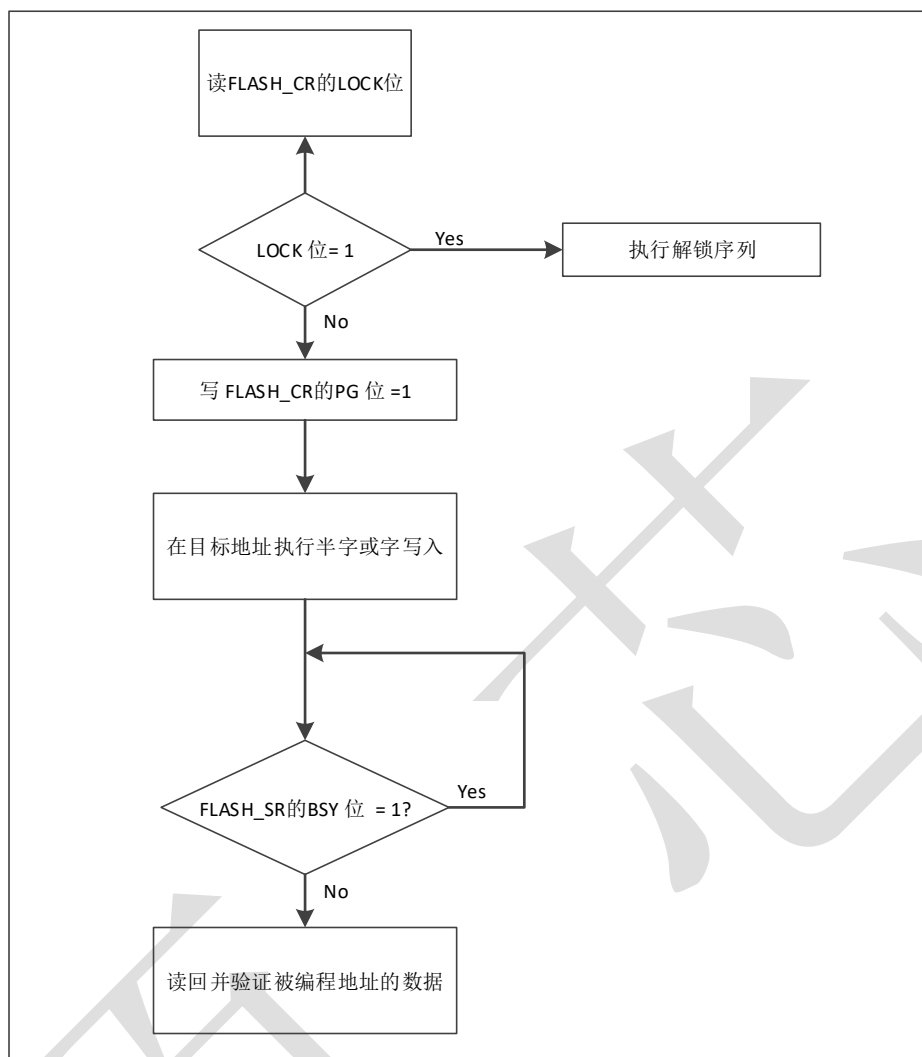


图 3-1 FLASH 编程流程

正常模式下的主程序区烧录流程如下：

1. 检测 FLASH_SR 寄存器的 BSY 位，确认当前没有主程序区的其他操作正在进行；
2. 设置寄存器 FLASH_CR 的 PG 位；
3. 半字或全字方式写入目标地址；
4. 等待 FLASH_SR 寄存器的 BSY 位被清零；
5. 检测 FLASH_SR 寄存器的 EOP 标志位（当烧录成功时，此标志位置起），然后软件清零。

注：以上烧录流程中的寄存器无法在 FLASH_SR 的 BSY 位设置起来后访问。

Flash 存储器的擦除

存储器可以按页擦除或者一次性全部擦除（块擦除）。

页擦除

页擦除的操作过程如下：

1. 通过检测 FLASH_SR 的 BSY 位，确认没有其他操作正在进行；
2. 设置 FLASH_CR 寄存器的 PER 位；
3. 设置寄存器 FLASH_AR，以确定需要擦除哪一页；

4. 将 FLASH_CR 寄存器的 STRT 位设置起来；
5. 等待 BSY 位被清零；
6. 检测 FLASH_SR 寄存器的 EOP 标志（当擦除操作成功后，EOP 标志位置起）；
7. 清除 EOP 标志

注：设置完 STRT 位后，软件必须等待至少一个 CPU 时钟周期才能检测 BSY 是否为 0。

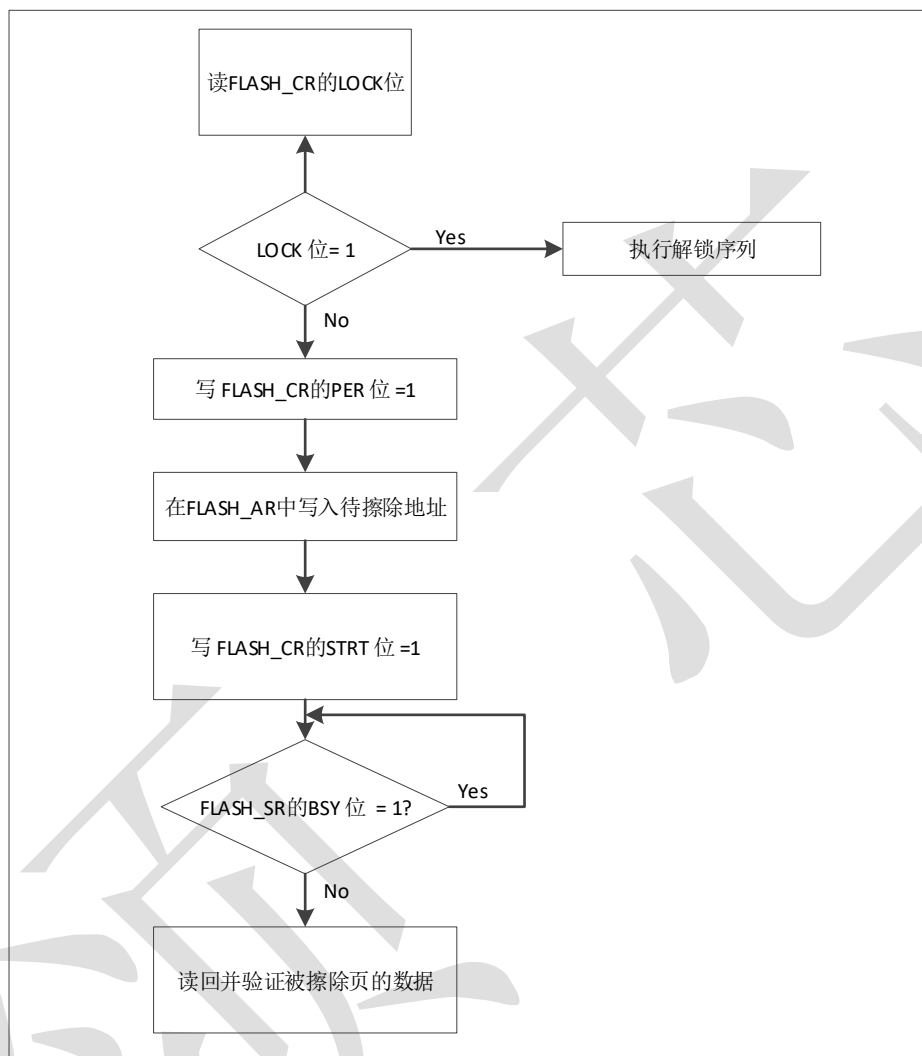


图 3-2 FLASH 的页擦除流程

块擦除

块擦除可以用于擦除主程序区的所有页，但不影响信息区单元。块擦除操作流程如下：

1. 通过检测 FLASH_SR 的 BSY 位，确认没有其他操作正在进行；
2. 设置 FLASH_CR 寄存器的 MER 位；
3. 将 FLASH_CR 寄存器的 STRT 位设置起来；
4. 等待 BSY 位被清零；
5. 检测 FLASH_SR 寄存器的 EOP 标志（当擦除操作成功后，EOP 标志位置起）；
6. 清除 EOP 标志

注：设置完 STRT 位后，软件必须等待至少一个 CPU 时钟周期才能检测 BSY 是否为 0。

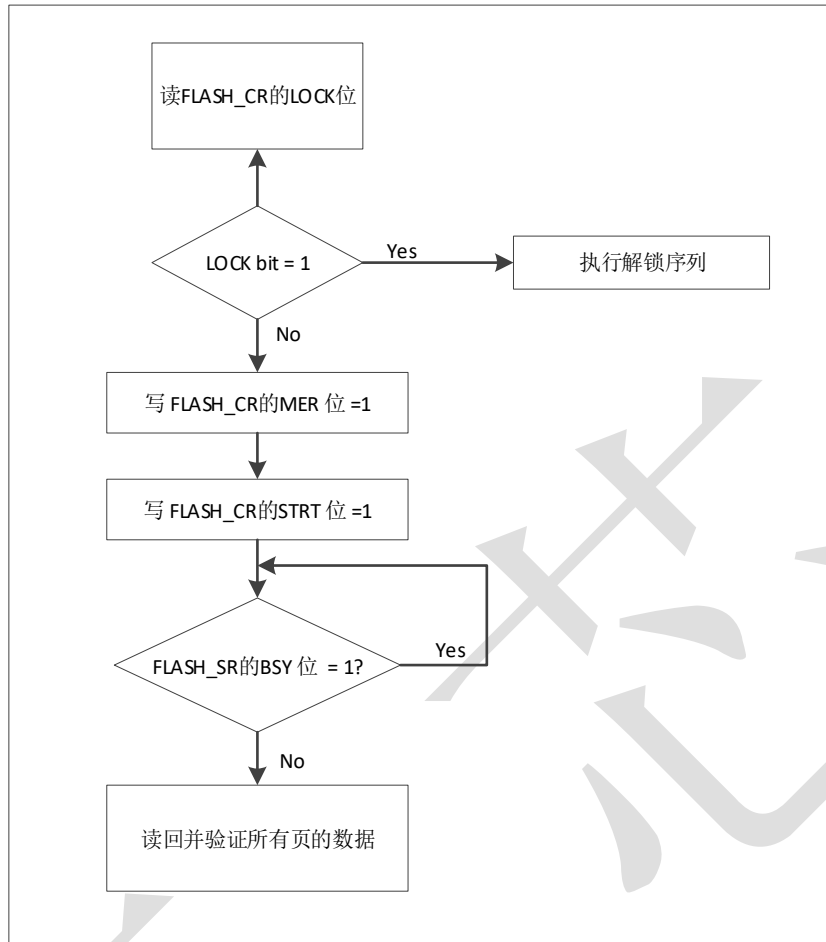


图 3-3 FLASH 的块擦除操作流程

选项字节（Option Byte）的编程

LCM32F037 的选项字节为一个地址段的区间，对应 0x1FFF_F600 至 0x1FFF_F7FF，此段地址空间的编程方式与正常的用户空间的编程方式不同。选项字节空间包括了写保护字，读保护字，硬件配置，用户定义字等内容。用户在解锁完成 FLASH_CR 寄存器的访问后，还需要将解锁密钥写入寄存器 FLASH_OPTKEYR，正确解锁后将 FLASH_CR 寄存器中的 OPTWRE 位置起。此时，用户可以通过设置 FLASH_CR 寄存器中的 OPTPG 位启动选项字节的编程，编程时，以全字方式写入目标存储区地址。

目标地址首先读出确认是否已经被擦除，如果没有，则忽略当前的烧录操作并设置 FLASH_SR 寄存器的 WRPRERR 位。编程结束后，寄存器中的 EOP 标志位被置起。

开始编程之前，选项字节的半字高 8 位分别对应低 8 位的反码，高 8 位会自动扩展（即字地址 bit[15:8] 是 bit[7:0] 的反码，bit[31:24] 是 bit[23:16] 的反码），原写入高 8 位数据忽略，以确保写入字节及其反码的严格对应。

操作流程如下：

1. 检测 FLASH_SR 寄存器中的 BSY 位，确定当前没有其他操作闪存的工作在进行中；
2. 解锁 FLASH_CR 寄存器中的 OPTWRE 位（通过写密钥到 FLASH_OPTKEYR 寄存器）；
3. 设置 FLASH_CR 寄存器中的 OPTPG 位；
4. 将数据写入目标地址（全字方式，实际有效 bit[7:0]，bit[23:16]）；
5. 等待 BSY 位被清零；
6. 读取目标数据进行校验。

选项字节包含读保护字，当前操作如果将读保护等级从保护状态降级到非保护，则在保护字被更新之前，控制模块发起一个块擦除操作，将主程序的数据全部擦除。除此之外对于选项字节的其他操作均不会引发块擦除操作。读保护字被擦除的默认状态下为保护状态（Level 1）。

擦除流程

选项字节的擦除操作过程如下：

1. 检测 FLASH_SR 寄存器中的 BSY 位，确定当前没有其他操作闪存的工作在进行中；
2. 解锁 FLASH_CR 寄存器中的 OPTWRE 位（通过写密钥到 FLASH_OPTKEYR 寄存器）；
3. 设置 FLASH_CR 寄存器中的 OPTER 位；
4. 设置 FLASH_CR 寄存器中的 STRT 位；
5. 等待 BSY 位被清零；
6. 读取目标数据进行校验。

3.3 存储器的保护

处于闪存单元中的用户数据可以设置保护来防止对其进行非法读取，LCM32F037 支持分页的读保护设置。另外，FLASH 按照一定的分页方式可进行写保护，写保护总位数 32 位，主程序区域的 32KB 按照 1KB 一个 Sector，可分别设置写保护。

3.3.1 读保护

读保护状态通过设置选项字节中的 RDPn（0~7）来激活，复位时加载对应读保护选项字节，并确定相应区域的读保护状态。

注：如果 FLASH 被设置了读保护，而调试器通过 SWD 接口进行访问时，系统产生一个总线错误。

读保护有 3 个保护等级，从没有保护（级别 0）到最强保护或者无调试保护状态（级别 2），具体参考表 3-2。LCM32F037 具备分区保护的功能（不包括级别 2，最强保护涵盖全体闪存单元），不同分区之间，低级别的分区不能访问高保护级别的分区，而高保护级别区内的程序可以访问相同或者低于自身保护等级分区内的数据。

读保护通过选项字节 RDPn 来确定保护等级，此选项字节含特定字符及其反码：

表 3-2 FLASH 读保护

RDPn	取值	反码	读保护等级	保护区域
RDP0-7	0xAA	0x55	等级 0	RSector0-7
	除 0xAA 外的任意值	除 0x55 外任意值	等级 1	
RDP8-11	0xCCCC_CCCC	0x3333_3333	等级 2	所有 FLASH 区域

系统存储区域可以被任意操作读取，但用户不能对其进行擦除或者写入的操作，只能通过其他的特殊寄存器来进行擦写。

等级 0：无保护

对应主程序区域可以被读取、擦除以及编程，选项字节可以被任意访问。

等级 1：读保护

默认状态下，FLASH 处于擦除状态，此时所有区域处于读保护状态。实际上，只要选项字节 RDPn 的值不是 0xAA，无须其补码是否正确，相关区域即被设定为读保护。

- 用户程序运行：低级别的分区不能访问高保护级别的分区，而高保护级别区内的程序可以访问相同或者低于自身保护等级分区内的数据；

- 调试模式、动态存储单元 (RAM)、系统存储 (System Memory): 通过 SWD 接口的调试访问, 初始化程序访问等均不能访问被保护的主程序区, 读被保护区的操作会导致发生总线错误以及触发硬件设备故障的中断。同时, 初始引导程序也不能对被保护区进行擦除或者编程操作, 错误操作会被阻止, 并将 FLASH 状态寄存器中的 PGERR 标志位置起。

任意选项字节 RDPn 被重新编程降级为保护等级 0 时, 会触发整个 FLASH 被擦除的块擦除动作。

等级 2: 禁止调试

此模式下, 内核调试功能及接口被禁止, 并且, 禁止从动态存储区 (RAM) 和系统存储区 (System Memory) 加载初始引导程序。

用户程序的运行时, 可以进行正常读取同一保护等级分区的数据, 但还是不能操作前述分区保护等级更高的区域。而且, 用户程序此时对选项字节只能进行读取和编程操作, 而不能进行擦除。选项字节固定不能进行任何擦除的操作。

其中 RDPn 字节同时禁止重新编程, 因此, 读保护等级 2 是不可撤销的状态。任何试图编程 RDPn 字节的操作会置起 FLASH_SR 寄存器的 WRPRERR 位, 并触发中断。

注: 等级 2 将禁止所有调试, 无法通过 SWD 分析内部运行状态, 配置读保护等级 2 操作不可恢复。

表 3-3 保护等级与执行模式对照访问权限

区域	保护等级	用户程序执行			调试模式/RAM 区或系统区引导程序		
		读	写	擦	读	写	擦
主程序区	1	是	是	是	否	否	否 ⁴
	2	是	是	是	N/A ¹	N/A ¹	N/A ¹
系统存储区 (²)	1	是	否	否	是	否	否
	2	是	否	否	N/A ¹	N/A ¹	N/A ¹
选项字节	1	是	是 ³	是	是 ⁴	是 ⁴	是
	2	是	是 ⁵	否	N/A ¹	N/A ¹	N/A ¹

注 1: 读保护等级 2, 调试端口, 从 RAM 区或系统区的初始引导被禁止。

注 2: 系统存储区不能被用户程序擦写, 只能读取, 与具体保护等级无关。

注 3: 选项字节 RDPn 从保护等级 1 变化到等级 0 (0xAA) 时, 主程序区被强制擦除。

注 4: 当 RDPn 为等级 1 时, 内部引导程序不能直接对选项字节操作, 必须先移除 RDPn 的保护 (设置到 0 等级)。

注 5: 除了 RDPn 的其他所有选项字节可以被编程。

读保护等级的修改

保护等级的提升相对比较简单, 只需要将 RDPn 字节按照如表 3-2 所示编程即可, 即从等级 0 可以通过直接编程提升到等级 1, 等级 1 可以提升到等级 2。但从等级 1 降级到等级 0, 则在写入 0xAA 到 RDPn 时会同时擦除所有 FLASH 主存储区。

3.3.2 写保护

写保护按照区块进行分别保护, 对应保护选项 WRPx 字。对写保护区域的编程或者擦除操作会产出一个 WRPRERR 标识, 可通过 FLASH_SR 寄存器来查询此位。

写保护的去除

有以下两种情况:

- 除去写保护, 同时去除读保护:
 - 通过设置寄存器 FLASH_CR 的 OPTER 位来擦除整个选项字节区, 此时 RDPn 被擦除, 读保护升

级为 1

- 将读保护字节全部写成 0xAA 以降级为 0，此操作将擦除整个主程序区
- 去除写保护，但读保护状态维持，此情况一般用于自带引导程序的应用在线编程：
 - 通过设置 OPTER 位擦除选项字节区
 - 通过上电复位或者外部复位 RST 重新加载读写保护字节

3.3.3 选项字节的写保护

选项字节默认写保护，但均可以被读取。此区域的烧写需要先写入特定序列的秘钥（与解锁相同）到 OPTKEYR 寄存器，正确写入后，FLASH_CR 寄存器中的 OPTWRE 位被置起，可查询。将此标志位清零，则选项字节区回到锁定状态。

3.4 Flash 相关中断

表 3-4 Flash 中断请求列表

中断触发事件	触发标志	使能控制位
操作结束	EOP	EOPIE
写保护错误	WRPRTERR	ERRIE
编程错误	PGERR	ERRIE

3.5 Flash 寄存器描述

FLASH 相关的寄存器必须以 32-bit 字访问（不允许半字和字节访问），并通过 APB1 接口接入总线。

表 3-5 Flash 相关寄存器表

名称	说明	读写权限	复位值	字节地址
FLASH_ACR	访问控制寄存器	R/W	0x0000_0010	0x4001D000
FLASH_KEYR	钥匙寄存器	W	0xFFFF_XXXX	0x4001D004
FLASH_OPTKER	选项字节钥匙寄存器	W	0xFFFF_XXXX	0x4001D008
FLASH_SR	状态寄存器	R/W	0x0000_0000	0x4001D00C
FLASH_CR	控制寄存器	R/W	0x0X01_0080	0x4001D010
FLASH_AR	地址寄存器	R/W	0x0000_0000	0x4001D014
FLASH_OBR	选项字节寄存器	R/W	0xFFFF_XXXX	0x4001D018
FLASH_CAL1	校准字节寄存器 1	R	0xFFFF_XXXX	0x4001D01C
FLASH_RDPR	读保护寄存器	R	0xFFFF_XXXX	0x4001D020
FLASH_WRPR	写保护寄存器	R	0xFFFF_XXXX	0x4001D024
FLASH_CAL2	校准字节寄存器 2	R	0xFFFF_XXXX	0x4001D028
FLASH_CAL3	校准字节寄存器 3	R	0xFFFF_XXXX	0x4001D02C
FLASH_CAL4	校准字节寄存器 4	R	0xFFFF_XXXX	0x4001D030
FLASH_SMKEYR	系统存储钥匙寄存器	W	0xFFFF_XXXX	0x4001D03C

注：x 表示不确定；-表示未实现；u 表示不受复位影响；c 表示该位复位值因复位源而异；R 只读；W 只写；R-0 表示只读且读 0；R/W 表示可读写（以后章节同上述）。

3.5.1 Flash 访问控制寄存器 (FLASH_ACR)

地址偏移: 0x00

复位值: 0x0000_0010

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.										PRFTBS	PRFTBE	Res.	LATENCY[2:0]		
										r	rw		rw		

Bits	31:6	保留, 必须保持复位值
Bit	5	PRFTBS: 预取缓存状态
		标示预取缓存当前状态
		0: 当前预取缓存没有使能 (默认)
		1: 当前预取缓存已经使能
		<i>注: 当第一个预取请求完成后, 状态标志位即被设置成 1</i>
Bit	4	PRFTBE: 预取缓存的使能控制
		0: 预取禁止
		1: 预取使能 (默认)
Bit	3	保留, 必须保持复位值
Bits	2:0	LATENCY[2:0]: 读取延迟
		置闪存读取的系统时钟延迟周期数 (SYSCLK)
		000: 0 等待周期, 当 $SYSCLK < 32 \text{ MHz}$ (默认)
		001: 1 个等待周期, 当 $32 \text{ MHz} \leq SYSCLK < 64 \text{ MHz}$
		010: 2 个等待周期, 当 $64 \text{ MHz} \leq SYSCLK$

3.5.2 Flash 钥匙寄存器 (FLASH_KEYR)

地址偏移: 0x04

复位值: 0xxxxx_xxxx

寄存器只写, 读返回 0。

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

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

Bits	31:0	FKEY[31:0]: 闪存密钥
		代表解锁 FLASH_CR 寄存器的写入密钥数据

3.5.3 选项字节钥匙寄存器 (FLASH_OPTKEYR)

地址偏移: 0x08

复位值: 0xFFFF_FFFF

寄存器只写, 读返回 0。

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

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

Bits	31:0	OPTKEY[31:0]: 选项字节密钥
		代表解锁 OPTWRE 寄存器的写入密钥

3.5.4 Flash 状态寄存器 (FLASH_SR)

地址偏移: 0x0C

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.											OBER	OB_E	OB_B	OB_BOOT_	
											R	MPTY_	OOT0		MODE[1:0]
											r	r	r	r	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.										EOP	WRPR	Res.	PGER	Res.	BSY
										rw	rw		rw		r

Bits	31:21	保留, 必须保持复位值
Bit	20	OBERR: 选项字节错误
		0: 表示无异常 (默认)
		1: 表示加载的选项字节与其补码不对应, 相关加载的字节设定为 0xFF
Bit	19	OB_EMPTY_FLAG: 主存储空间空标志
		标志系统复位后主存储空间的状态
		0: 主存储空间已被编程 (默认)
		1: 主存储空间未被编程
Bit	18	OB_BOOT0_PIN: BOOT0 管脚的值
		标识系统复位后所采样到的 BOOT0 管脚的值
		0: BOOT0 管脚低电平
		1: BOOT0 管脚高电平
Bits	17:16	OB_BOOT_MODE [1:0]: 选项字节配置的设备启动模式

	系统复位后设备真正的启动配置模式
	x0: 主存储区地址映射到 0x0000_0000 (默认)
	01: 系统存储区映射到 0x0000_0000
	11: 内部 SRAM 存储映射到 0x0000_0000
Bits 15:6	保留, 必须保持复位值
Bit 5	EOP: FLASH 操作结束标志
	由硬件置位, 完成一次有效的 FLASH 写或擦除操作, 软件写 1 清零
Bit 4	WRPRTERR: 写保护错误
	硬件置位, 当对一个写保护地址进行编程操作时发生, 软件写 1 清零
Bit 3	保留, 必须保持复位值
Bit 2	PGERR: 编程错误
	硬件置位, 当编程写入的目标地址的初值不为 0xFFFF, 或者同一个地址多次写入软件写 1 清零
	<i>注: 开始一次编程操作前, FLASH_CR 寄存器中的 STRT 位须先被清零。</i>
Bit 1	保留, 必须保持复位值
Bit 0	BSY: 忙标识
	表示当前一个 FLASH 操作正在执行过程, 当开始 FLASH 操作时置位, 完成操作或者发生 FLASH 相关错误发生时清零。

3.5.5 Flash 控制寄存器 (FLASH_CR)

地址偏移: 0x10

复位值: 0x0X01_0080 (X 表示由选项启动模式字节规定的启动模式)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.						BOOT_MODE [1:0]		Res.						SMLOCK	
						rw								rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			EOPIE	Res.	ERRIE	OPTWRE	Res.	LOCK	STRT	OPTER	OPTPG	Res.	MER	PER	PG
			rw		rw	rw		rw	rw	rw	rw		rw	rw	rw

Bits 31:26	保留, 必须保持复位值
Bits 25:24	BOOT_MODE[1:0]: 设备启动模式
	软件可以进行读写, 控制哪个存储区映射到 0x0000_0000, 复位后标示实际启动模式的配置
	x0: 主存储区地址映射到 0x0000_0000 (默认)
	01: 系统存储区映射到 0x0000_0000
	11: 内部 SRAM 存储映射到 0x0000_0000
Bits 23:17	保留, 必须保持复位值
Bit 16	SMLOCK: 系统存储区写锁定
	只能写入 1, 表示锁定系统存储区, 当存储区被正确的解锁后清零, 非法进行的解锁操作, 此位被强制设定为 1 直至下一个复位

Bits 15:13	保留，必须保持复位值
Bit 12	OPIE : 操作结束中断使能控制
	当 FLASH_SR 寄存器中的 EOP 置位，控制相关中断是否产生
	0: 不允许产出中断（默认）
	1: 允许产出中断
Bit 11	保留，必须保持复位值
Bit 10	ERRIE : 错误中断使能位
	当 FLASH_SR 寄存器中的 PGERR/WRPRTERR 置位时，此位使能在错误时产生中断。
	0: 不允许产出中断（默认）
	1: 允许产出中断
Bit 9	OPTWRE : 选项字节写使能控制
	只能写 0，为 1 时表示选项字节可以被编程写入，当正确写秘钥入 FLASH_OPTKEYR 寄存器时置 1，可以被软件清零，当解锁操作失败，此标志位保持为清零状态，直至下次复位产生
Bit 8	保留，必须保持复位值
Bit 7	LOCK : 锁定标志
	只可以写 1，为 1 表示 FLASH 操作被锁定，正确的解锁操作可以清零，失败的解锁操作会导致其被锁定为 1，直至下次复位发生
Bit 6	STRT : 开始 FLASH 操作
	写 1 表示启动一次擦除 ERASE 操作，由硬件置位，当 BSY 标志被清零时此位自动清零
Bit 5	OPTER : 选项字节擦除选择
	选择当前擦除操作为选项字节
Bit 4	OPTPG : 选项字节编程选择
	选择当前编程操作为选项字节
Bit 3	保留，必须保持复位值
Bit 2	MER : 块擦除
	擦除所有主存储区
Bit 1	PER : 页擦除
	选择按页擦除
Bit 0	PG : 编程操作
	选择 FLASH 编程

3.5.6 Flash 地址寄存器（FLASH_AR）

地址偏移: 0x14

复位值: 0x0000_0000

由硬件更新标示当前/上次使用的 FLASH 操作地址; 页擦除操作时, 由软件更新指定需要擦除的页地址。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FAR[31:16]															
rw															

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

Bits 31:0	FAR[31:0]: FLASH 地址
	编程操作时选择当前操作的目标地址，页擦除操作时选择擦除页
	当 FALSH_SR 寄存器中的 BSY 位置位时，写入操作被阻止

3.5.7 选项字节寄存器 (FLASH_OBR)

地址偏移: 0x18

复位值: 0xFFFF_FFFF

此寄存器的部分复位值取决于相应选项字节的内容。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA2[7:0]								DATA1[7:0]							
r								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA0[7:0]								Res.				WDG _SW	BOOT _SEL	nBOO T1	nBOO T0
r												r	r	r	r

Bits 31:24	DATA2[7:0]: 用户数据字节 2
Bits 23:16	DATA1[7:0]: 用户数据字节 1
Bits 15:8	DATA0[7:0]: 用户数据字节 0
Bits 7:4	保留，必须保持复位值
Bit 3	WDG_SW
	0: 硬件配置看门狗
	1: 软件配置看门狗
Bit 2	BOOT_SEL
	0: BOOT0 信号由选项字节的 nBOOT0 位确定
	1: BOOT0 信号通过 BOOT0 管脚赋值 (传统模式)
Bit 1	nBOOT1
	与 BOOT0 信号一起定义设备启动模式
Bit 0	nBOOT0
	当 BOOT_SEL 为 0，确定 BOOT0 信号的值用来选择设备的启动模式

3.5.8 校准寄存器 1 (FLASH_CAL1)

地址偏移: 0x1C

复位值: 0xFFFF_FFFF

此寄存器的部分复位值取决于相应选项字节的内容。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.			nOP2TRIM[4:0]					Res.		BGRTRIM[5]	nBGRTRIM[4:0]				
			r							r	r				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				nRCHPT[3:0]				RCHTRIM[7]	nRCHTRIM[6:0]						
				r				r	r						

Bits	31:29	保留，必须保持复位值
Bits	28:24	nOP2TRIM[4:0]: OPA2 校准值反码
Bits	23:22	保留，必须保持复位值
Bit	21	BGRTRIM[5]: VBG 校准值
Bits	20:16	nBGRTRIM[4:0]: VBG 校准值反码
Bits	15:12	保留，必须保持复位值
Bits	11:8	nRCHPT[3:0]: RCH 温度曲线校准值反码
Bit	7	RCHTRIM[7]: RCH 校准值
Bits	6:0	nRCHTRIM[6:0]: RCH 校准值反码

3.5.9 读保护寄存器 (FLASH_RDPR)

地址偏移: 0x20

复位值: 0xXXXX_XXXX

此寄存器的部分复位值取决于相应选项字节的内容。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.							RDPRT2	RDPRT[7:0]							
							r								r

Bits	31:9	保留，必须保持复位值
Bit	8	RDPRT2: FLASH 读保护等级 2 的状态标志位
		0: 没有设置为读保护等级 2
		1: 已被设定为读保护等级 2
Bits	7:0	RDPRT[7:0]: 8 个 FLASH 区域的读保护状态
		0: 读保护等级为 0
		1: 读保护等级为 1

3.5.10 写保护寄存器 (FLASH_WRPR)

地址偏移: 0x24

复位值: 0xXXXX_XXXX

此寄存器的部分复位值取决于相应选项字节的内容。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WRP[31:16]															
r															

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

Bits 31:0	WRP[31:0]: 写保护
	从 OBL（选项字节）中加载的写保护字

3.5.11 校准寄存器 2（FLASH_CAL2）

地址偏移：0x28

复位值：0xXXXX_XXXX

此寄存器的部分复位值取决于相应选项字节的内容。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADC GTRI M[7]	nADC GTRI M[6]	ADC GTRI M[5]	nADC GTRIM[4:2]			ADC GTRIM[1:0]		ADC VTRI M[7]	nADC VTRIM[6:5]		ADC VTRI M[4]	nADC VTRI M[3]	ADC VTRIM[2:0]		
r	r	r	r			r		r	r		r	r	r		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.										RCLTR IM[5]	nRCLTRIM[4:0]				
										r	r				

Bit 31	ADCGTRIM[7]: ADC 增益校准值
Bit 30	nADCGTRIM[6]: ADC 增益校准值反码
Bit 29	ADCGTRIM[5]: ADC 增益校准值
Bits 28:26	nADCGTRIM[4:2]: ADC 增益校准值反码
Bits 25:24	ADCGTRIM[1:0]: ADC 增益校准值
Bit 23	ADCVTRIM[7]: ADC 偏移校准值
Bits 22:21	nADCVTRIM[6:5]: ADC 偏移校准值反码
Bit 20	ADCVTRIM[4]: ADC 偏移校准值
Bit 19	nADCVTRIM[3]: ADC 偏移校准值反码
Bits 18:16	ADCVTRIM[2:0]: ADC 偏移校准值
Bits 15:6	保留，必须保持复位值
Bit 5	RCLTRIM[5]: RCL 校准值
Bits 4:0	nRCLTRIM[4:0]: RCL 校准值反码

3.5.12 校准寄存器 3（FLASH_CAL3）

地址偏移：0x2C

复位值：0xXXXX_XXXX

此寄存器的部分复位值取决于相应选项字节的内容。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.			CP1TRIM[4]	nCP1TRIM[3:0]				Res.			CP0TRIM[4]	nCP0TRIM[3:0]			
			r	r							r	r			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			nOP1TRIM[4:0]				Res.			nOP0TRIM[4:0]					
			r							r					

Bits	31:29	保留，必须保持复位值
Bit	28	CP1TRIM[4] : ACMP1 比较器 1 校准值
Bits	27:24	nCP1TRIM[3:0] : ACMP1 比较器 1 校准值反码
Bits	23:21	保留，必须保持复位值
Bit	20	CP0TRIM[4] : ACMP0 比较器 0 校准值
Bits	19:16	nCP0TRIM[3:0] : ACMP0 比较器 0 校准值反码
Bits	15:13	保留，必须保持复位值
Bits	12:8	nOP1TRIM[4:0] : OPA1 校准值反码
Bits	7:5	保留，必须保持复位值
Bits	4:0	nOP0TRIM[4:0] : OPA0 校准值反码

3.5.13 校准寄存器 4 (FLASH_CAL4)

地址偏移：0x30

复位值：0xXXXX_XXXX

此寄存器的部分复位值取决于相应选项字节的内容。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.											CP2TRIM[4]	nCP2TRIM[3:0]			
											r	r			

Bits	31:5	保留，必须保持复位值
Bit	4	CP2TRIM[4] : ACMP2 比较器 2 校准值
Bits	3:0	nCP2TRIM[3:0] : ACMP2 比较器 2 校准值反码

3.5.14 系统存储钥匙寄存器 (FLASH_SMKEYR)

地址偏移：0x3C

复位值：0xXXXX_XXXX

寄存器只写，读返回 0。

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

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

Bits	31:0	SMKEY[31:0]: 系统存储区秘钥
		解锁系统存储区所需要写入的秘钥

3.6 选项字节

选项字节占用信息区一个 Page，大小为 512B，此区域的信息可以由用户根据实际需求自行配置。例如，决定看门狗是由软件还是硬件开启。

选项字节区域的 32 位字，由如下方式构成：

表 3-6 选项字节的格式

31-24	23-16	15-8	7-0
字节 1 反码	选项字节 1	字节 0 的反码	选项字节 0

信息区里面目前定义的选项字节参考表 3-6。选项字节可以通过表 3-7 所列的地址空间读取，也可以通过寄存器 FLASH_OBR 读取。

注：系统复位后将新编程的选项字节重新装载。

表 3-7 选项字节

地址	[31:24]	[23:16]	[15:8]	[7:0]
0x1FFF_F600	nDATA0	DATA0	nUSER	USER
0x1FFF_F604	nDATA2	DATA2	nDATA1	DATA1
0x1FFF_F608	nWRP1	WRP1	nWRP0	WRP0
0x1FFF_F60C	nWRP3	WRP3	nWRP2	WRP2
0x1FFF_F610	nRDP1	RDP1	nRDP0	RDP0
0x1FFF_F614	nRDP3	RDP3	nRDP2	RDP2
0x1FFF_F618	nRDP5	RDP5	nRDP4	RDP4
0x1FFF_F61C	nRDP7	RDP7	nRDP6	RDP6
0x1FFF_F620	nRDP9	RDP9	nRDP8	RDP8
0x1FFF_F624	nRDP11	RDP11	nRDP10	RDP10

3.6.1 用户及数据选项配置字节

FLASH 存储地址：0x1FFF_F600

默认配置值：0x00FF_00FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
nDATA0								DATA0							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nUSER								USER							
								Res.				WDG _SW	BOOT _SEL	nBOO T1	nBOO T0
rw												rw	rw	rw	rw

Bits 31:24	nDATA0: 用户数据字节 0 的反码
Bits 23:16	DATA0: 用户数据字节 0 (上电后加载到 FLASH_OBR[15:8])
Bits 15:8	nUSER: User 选项字节的反码
Bits 7:0	USER: 用户配置选项字节 (复位加载到 FLASH_OBR[7:0])
Bits 7:4	保留, 必须保持复位值
Bit 3	WDG_SW
	0: 硬件配置看门狗
	1: 软件配置看门狗
Bit 2	BOOT_SEL
	0: BOOT0 信号由选项字节的 nBOOT0 位确定
	1: BOOT0 信号通过 BOOT0 管脚赋值 (传统模式)
Bit 1	nBOOT1
	与 BOOT0 信号一起定义设备启动模式
Bit 0	nBOOT0
	当 BOOT_SEL 为 0, 确定 BOOT0 信号的值用来选择设备的启动模式

3.6.2 用户配置选项字节

FLASH 存储地址: 0x1FFF_F604

默认配置值: 0x00FF_00FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
nDATA2								DATA2							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nDATA1								DATA1							
rw								rw							

Bits 31:24	nDATA2: 用户数据字节 2 的反码
Bits 23:16	DATA2: 用户定义数据字节 2 (复位加载到 FLASH_OBR[31:24])
Bits 15:8	nDATA1: 用户定义数据字节 1 的反码
Bits 7:0	DATA1: 用户定义数据字节 1 (复位加载到 FLASH_OBR[23:16])

3.6.3 写保护选项字节

此系列的选项字节用于定义 FLASH 存储区域的写保护, WRPx 区域的某一位清零 (对应 nWRPx 区域位置 1), 其对应的存储区块被写保护。WRP 的 32 位对应 FLASH 存储的 32 个区域, 每个区域大小为 1KB (2

Pages)。

3.6.3.1 写保护选项字节 0/1

FLASH 存储地址: 0x1FFF_F608

默认配置值: 0x00FF_00FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
nWRP1								WRP1							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nWRP0								WRP0							
rw								rw							

Bits 31:24	nWRP1: FLASH 存储写保护字节 1 反码
Bits 23:16	WRP1: FLASH 存储写保护字节 1 (复位加载到 FLASH_WRP1[15:8])
Bits 15:8	nWRP0: FLASH 存储写保护字节 0 反码
Bits 7:0	WRP0: FLASH 存储写保护字节 0 (复位加载到 FLASH_WRP0[7:0])

3.6.3.2 写保护选项字节 2/3

FLASH 存储地址: 0x1FFF_F60C

默认配置值: 0x00FF_00FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
nWRP3								WRP3							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nWRP2								WRP2							
rw								rw							

Bits 31:24	nWRP3: FLASH 存储写保护字节 3 反码
Bits 23:16	WRP3: FLASH 存储写保护字节 3 (复位加载到 FLASH_WRP3[31:24])
Bits 15:8	nWRP2: FLASH 存储写保护字节 2 反码
Bits 7:0	WRP2: FLASH 存储写保护字节 2 (复位加载到 FLASH_WRP2[23:16])

3.6.4 读保护选项字节

3.6.4.1 读保护选项字节 0/1

FLASH 存储地址: 0x1FFF_F610

默认配置值: 0x55AA_55AA

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
nRDP1								RDP1							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nRDP0								RDPO							
rw								rw							

Bits 31:24	nRDP1: FLASH 存储读保护选项字节 1 反码
Bits 23:16	RDPI: FLASH 存储读保护选项字节 1
	选项字节定义对应 FLASH 存储单元 1 保护等级
	0xAA: 等级 0 (出厂默认配置)
	0xXX (除了 0xAA): 等级 1
Bits 15:8	nRDP0: FLASH 存储读保护选项字节 0 反码
Bits 7:0	RDPO: FLASH 存储读保护选项字节 0
	选项字节定义对应 FLASH 存储单元 0 保护等级
	0xAA: 等级 0 (出厂默认配置)
	0xXX (除了 0xAA): 等级 1
	<i>注: 读保护等级复位加载到 FLASH 读保护寄存器 (FLASH_RDPR) 的 RDPRT[7:0]</i>

3.6.4.2 读保护选项字节 2/3

FLASH 存储地址: 0x1FFF_F614

默认配置值: 0x55AA_55AA

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
nRDP3								RDP3							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nRDP2								RDP2							
rw								rw							

Bits 31:24	nRDP3: FLASH 存储读保护选项字节 3 反码
Bits 23:16	RDPI: FLASH 存储读保护选项字节 3
	选项字节定义对应 FLASH 存储单元 3 保护等级
	0xAA: 等级 0 (出厂默认配置)
	0xXX (除了 0xAA): 等级 1
Bits 15:8	nRDP2: FLASH 存储读保护选项字节 2 反码
Bits 7:0	RDPI: FLASH 存储读保护选项字节 2
	选项字节定义对应 FLASH 存储单元 2 保护等级
	0xAA: 等级 0 (出厂默认配置)
	0xXX (除了 0xAA): 等级 1
	<i>注: 读保护等级复位加载到 FLASH 读保护寄存器 (FLASH_RDPR) 的 RDPRT[7:0]</i>

3.6.4.3 读保护选项字节 4/5

FLASH 存储地址: 0x1FFF_F618

默认配置值: 0x55AA_55AA



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
nRDP5								RDP5							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nRDP4								RDP4							
rw								rw							

Bits 31:24	nRDP5: FLASH 存储读保护选项字节 5 反码
Bits 23:16	RDP5: FLASH 存储读保护选项字节 5
	选项字节定义对应 FLASH 存储单元 5 保护等级
	0xAA: 等级 0 (出厂默认配置)
	0xXX (除了 0xAA): 等级 1
Bits 15:8	nRDP4: FLASH 存储读保护选项字节 4 反码
Bits 7:0	RDP4: FLASH 存储读保护选项字节 4
	选项字节定义对应 FLASH 存储单元 4 保护等级
	0xAA: 等级 0 (出厂默认配置)
	0xXX (除了 0xAA): 等级 1
	注: 读保护等级复位加载到 FLASH 读保护寄存器 (FLASH_RDPR) 的 RDPRT[7:0]

3.6.4.4 读保护选项字节 6/7

FLASH 存储地址: 0x1FFF_F61C

默认配置值: 0x55AA_55AA

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
nRDP7								RDP7							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nRDP6								RDP6							
rw								rw							

Bits 31:24	nRDP7: FLASH 存储读保护选项字节 7 反码
Bits 23:16	RDP7: FLASH 存储读保护选项字节 7
	选项字节定义对应 FLASH 存储单元 7 保护等级
	0xAA: 等级 0 (出厂默认配置)
	0xXX (除了 0xAA): 等级 1
Bits 15:8	nRDP6: FLASH 存储读保护选项字节 6 反码
Bits 7:0	RDP6: FLASH 存储读保护选项字节 6
	选项字节定义对应 FLASH 存储单元 6 保护等级
	0xAA: 等级 0 (出厂默认配置)
	0xXX (除了 0xAA): 等级 1
	注: 读保护等级复位加载到 FLASH 读保护寄存器 (FLASH_RDPR) 的 RDPRT[7:0]

3.6.4.5 读保护选项字节 8/9

FLASH 存储地址: 0x1FFF_F620

默认配置值: 0xFFFF_FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
nRDP9								RDP9							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nRDP8								RDP8							
rw								rw							

Bits 31:24	nRDP9: FLASH 存储读保护选项字节 9 反码
Bits 23:16	RDP9: FLASH 存储读保护选项字节 9
Bits 15:8	nRDP8: FLASH 存储读保护选项字节 8 反码
Bits 7:0	RDP8: FLASH 存储读保护选项字节 8
	RDP8 至 RDP11 的组合定义整个 FLASH 存储单元的读保护等级 2
	0xFFFF_XXXX (除 0xCCCC_CCCC 之外): level 0 (出厂默认配置)
	0xCCCC_CCCC: 等级 2
	注: 读保护等级 2 复位加载到 FLASH 读保护寄存器 (FLASH_RDPR) 的 RDPRT[8]

3.6.4.6 读保护选项字节 10/11

FLASH 存储地址: 0x1FFF_F624

默认配置值: 0xFFFF_FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
nRDP11								RDP11							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nRDP10								RDP10							
rw								rw							

Bits 31:24	nRDP11: FLASH 存储读保护选项字节 11 反码
Bits 23:16	RDP11: FLASH 存储读保护选项字节 11
Bits 15:8	nRDP10: FLASH 存储读保护选项字节 10 反码
Bits 7:0	RDP10: FLASH 存储读保护选项字节 10
	RDP8 至 RDP11 的组合定义整个 FLASH 存储单元的读保护等级 2
	0xFFFF_XXXX (除 0xCCCC_CCCC 之外): level 0 (出厂默认配置)
	0xCCCC_CCCC: 等级 2
	注: 读保护等级 2 复位加载到 FLASH 读保护寄存器 (FLASH_RDPR) 的 RDPRT[8]

注: 保护等级 2 慎用, 如果开启永远无法读写芯片, 彻底锁死。

3.7 校准字节

原厂设定，此区域主要用于确定当前芯片的外设校准信息。

3.7.1 出厂校准字节 0/1

FLASH 存储地址：0x1FFF_F5E0

默认配置值：0xXXXX_XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
nCAL1								CAL1							
								nRCHPT[3:0]							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nCAL0								CAL0							
								RCHTRIM[7]	nRCHTRIM[6:0]						
rw								rw	rw						

Bits	31:24	nCAL1: 出厂校准字节 1 反码
Bits	23:16	CAL1: 出厂校准字节 1 (复位加载到 FLASH_CAL1[15:8])
Bits	23:20	保留, 未做定义
Bits	19:16	nRCHPT[3:0]: RCH 温度曲线校准值反码
Bits	15:8	nCAL0: 出厂校准字节 0 反码
Bits	7:0	CAL0: 出厂校准字节 0 (复位加载到 FLASH_CAL1[7:0])
Bits	7	RCHTRIM[7]: RCH 校准值
Bits	6:0	nRCHTRIM[6:0]: RCH 校准值反码

3.7.2 出厂校准字节 2/3

FLASH 存储地址：0x1FFF_F5E4

默认配置值：0xXXXX_XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
nCAL3								CAL3							
								Res				nOP2TRIM[4:0]			
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nCAL2								CAL2							
								Res		BGRTRIM[5]	nBGRTRIM[4:0]				
rw										rw	rw				

Bits	31:24	nCAL3: 出厂校准字节 3 反码
------	-------	--------------------

Bits 23:16	CAL3: 出厂校准字节 3 (复位加载到 FLASH_CAL1[31:24])
Bits 23:21	保留, 未做定义
Bits 20:16	nOP2TRIM[4:0]: OPA2 校准值反码
Bits 15:8	nCAL2: 出厂校准字节 2 反码
Bits 7:0	CAL2: 出厂校准字节 2 (复位加载到 FLASH_CAL1[23:16])
Bits 7:6	保留, 未做定义
Bits 5	BGRTRIM[5]: BGR 校准值
Bits 4:0	nBGRTRIM[4:0]: BGR 校准值反码

3.7.3 出厂校准字节 4/5

FLASH 存储地址: 0x1FFF_F5E8

默认配置值: 0xXXXX_XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
nCAL5								CAL5							
								Res							
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nCAL4								CAL4							
								Res		RCLTRIM[5]	nRCLTRIM[4:0]				
rw										rw	rw				

Bits 31:24	nCAL5: 出厂校准字节 5 反码
Bits 23:16	CAL5: 出厂校准字节 5 (复位加载到 FLASH_CAL2[15:8])
Bits 23:16	保留, 未做定义
Bits 15:8	nCAL4: 出厂校准字节 4 反码
Bits 7:0	CAL4: 出厂校准字节 4 (复位加载到 FLASH_CAL2[7:0])
Bits 7:6	保留, 未做定义
Bits 5	RCLTRIM[5]: RCL 校准值
Bits 4:0	nRCLTRIM[4:0]: RCL 校准值反码

3.7.4 出厂校准字节 6/7

FLASH 存储地址: 0x1FFF_F5EC

默认配置值: 0xXXXX_XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
nCAL7								CAL7							
								ADC GTRI M[7]	nADC GTRI M[6]	ADC GTRI M[5]	nADC GTRIM[4:2]			ADC GTRIM[1:0]	
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
nCAL6								CAL6							
								ADC VTRI M[7]	nADC VTRIM[6:5]		ADC VTRI M[4]	nADC VTRI M[3]	ADC VTRIM[2:0]		
rw								rw	rw		rw	rw	rw		

Bits 31:24	nCAL7: 出厂校准字节 7 反码
Bits 23:16	CAL7: 出厂校准字节 7 (复位加载到 FLASH_CAL2[31:24])
Bit 23	ADCGTRIM[7]: 参考电压设为 3.3V 时, ADC 增益校准值[7]
Bit 22	nADCGTRIM[6]: 参考电压设为 3.3V 时, ADC 增益校准值[6]反码
Bit 21	ADCGTRIM[5]: 参考电压设为 3.3V 时, ADC 增益校准值[5]
Bits 20:18	nADCGTRIM[4:2]: 参考电压设为 3.3V 时, ADC 增益校准值[4:2]反码
Bits 17:16	ADCGTRIM[1:0]: 参考电压设为 3.3V 时, ADC 增益校准值[1:0]
Bits 15:8	nCAL6: 出厂校准字节 6 反码
Bits 7:0	CAL6: 出厂校准字节 6 (复位加载到 FLASH_CAL2[23:16])
Bit 7	ADCVTRIM[7]: 参考电压设为 3.3V 时, ADC 偏移校准值[7]
Bits 6:5	nADCVTRIM[6:5]: 参考电压设为 3.3V 时, ADC 偏移校准值[6:5]反码
Bits 4	ADCVTRIM[4]: 参考电压设为 3.3V 时, ADC 偏移校准值[4]
Bits 3	nADCVTRIM[3]: 参考电压设为 3.3V 时, ADC 偏移校准值[3]反码
Bits 2:0	ADCVTRIM[2:0]: 参考电压设为 3.3V 时, ADC 偏移校准值[2:0]

3.7.5 出厂校准字节 8/9

FLASH 存储地址: 0x1FFF_F5F0

默认配置值: 0xXXXX_XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
nCAL9								CAL9							
								Res.				nOP1TRIM[4:0]			
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nCAL8								CAL8							
								Res.				nOP0TRIM[4:0]			
rw								rw							

Bits 31:24	nCAL9: 出厂校准字节 9 反码
Bits 23:16	CAL9: 出厂校准字节 9 (复位加载到 FLASH_CAL3[15:8])
Bits 23:21	保留, 未做定义
Bits 20:16	nOP1TRIM[4:0]: OPA1 校准值反码
Bits 15:8	nCAL8: 出厂校准字节 8 反码
Bits 7:0	CAL8: 出厂校准字节 8 (复位加载到 FLASH_CAL3[7:0])
Bits 7:5	保留, 未做定义

Bits 4:0	nOPOTRIM[4:0]: OPA0 校准值反码
----------	---------------------------

3.7.6 出厂校准字节 10/11

FLASH 存储地址: 0x1FFF_F5F4

默认配置值: 0xXXXX_XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
nCAL11								CAL11							
								Res.				CP1TRI M[4]	nCP1TRIM[3:0]		
rw											rw	rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nCAL10								CAL10							
								Res.				CPOTRI M[4]	nCPOTRIM[3:0]		
rw											rw	rw			

Bits 31:24	nCAL11: 出厂校准字节 11 反码
Bits 23:16	CAL11: 出厂校准字节 11 (复位加载到 FLASH_CAL3[31:24])
Bits 23:21	保留, 未做定义
Bit 20	CP1TRIM[4]: ACMP1 校准值
Bits 19:16	nCP1TRIM[3:0]: ACMP1 校准值反码
Bits 15:8	nCAL10: 出厂校准字节 10 反码
Bits 7:0	CAL10: 出厂校准字节 10 (复位加载到 FLASH_CAL3[23:16])
Bits 7:5	保留, 未做定义
Bit 4	CPOTRIM[4]: ACMPO 校准值
Bits 3:0	nCPOTRIM[3:0]: ACMPO 校准值反码

3.7.7 出厂校准字节 12/13

FLASH 存储地址: 0x1FFF_F5F8

默认配置值: 0xXXXX_XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
nCAL13								CAL13							
								Res.							
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nCAL12								CAL12							
								Res.				CP2TRI M[4]	nCP2TRIM[3:0]		
rw											rw	rw			

Bits 31:24	nCAL13: 出厂校准字节 13 反码
Bits 23:16	CAL13: 出厂校准字节 13 (复位加载到 FLASH_CAL4[15:8])
Bits 23:16	保留, 未做定义
Bits 15:8	nCAL12: 出厂校准字节 12 反码
Bits 7:0	CAL12: 出厂校准字节 12 (复位加载到 FLASH_CAL4[7:0])
Bits 7:5	保留, 未做定义
Bit 4	CP2TRIM[4]: ACMP2 校准值
Bits 3:0	nCP2TRIM[3:0]: ACMP2 校准值反码

3.7.8 出厂校准字节 14/15

FLASH 存储地址: 0x1FFF_F5FC

默认配置值: 0xXXXX_XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
nCAL15								CAL15							
								Res.							
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nCAL14								CAL14							
								Res.							
rw															

Bits 31:24	nCAL15: 出厂校准字节 15 反码
Bits 23:16	CAL15: 出厂校准字节 15 (复位加载到 FLASH_CAL4[31:24])
Bits 23:16	保留, 未做定义
Bits 15:8	nCAL14: 出厂校准字节 14 反码
Bits 7:0	CAL14: 出厂校准字节 14 (复位加载到 FLASH_CAL4[23:16])
Bits 7:0	保留, 未做定义

3.7.9 出厂校准字节 16

FLASH 存储地址: 0x1FFF_F5A0

默认配置值: 0x00XX_00XX

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

Bits 31:24	保留, 未做定义
Bits 23:16	VOS_TRIM (5V) [7:0]: 参考电压设为 5V 时, ADC 偏移校准值

Bits 15:8	保留, 未做定义
Bits 7:0	GAIN_TRIM (5V) [7:0]: 参考电压设为 5V 时, ADC 增益校准值

3.7.10 出厂校准字节 17

FLASH 存储地址: 0x1FFF_F5A4

默认配置值: 0x00XX_00XX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.								VOS_TRIM (2.5V) [7:0]							
								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								GAIN_TRIM (2.5V) [7:0]							
								r							

Bits 31:24	保留, 未做定义
Bits 23:16	VOS_TRIM (2.5V) [7:0]: 参考电压设为 2.5V 时, ADC 偏移校准值
Bits 15:8	保留, 未做定义
Bits 7:0	GAIN_TRIM (2.5V) [7:0]: 参考电压设为 2.5V 时, ADC 增益校准值

3.7.11 出厂校准字节 18

FLASH 存储地址: 0x1FFF_F5A8

默认配置值: 0x00XX_00XX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.								VOS_TRIM (4V) [7:0]							
								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								GAIN_TRIM (4V) [7:0]							
								r							

Bits 31:24	保留, 未做定义
Bits 23:16	VOS_TRIM (4V) [7:0]: 参考电压设为 4V 时, ADC 偏移校准值
Bits 15:8	保留, 未做定义
Bits 7:0	GAIN_TRIM (4V) [7:0]: 参考电压设为 4V 时, ADC 增益校准值

3.7.12 出厂校准字节 19

FLASH 存储地址: 0x1FFF_F5AC

默认配置值: 0x0000_0XXX

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				VTS_data (3.3V) [11:0]											
				r											

Bits	31:12	保留, 未做定义
Bits	11:0	VTS_data (3.3V) [11:0]: ADC 在 3.3V 下校准后, 参考电压 3.3V, 测温 25°C 所得 VTS 值

3.7.13 出厂校准字节 20

FLASH 存储地址: 0x1FFF_F5C8

默认配置值: 0x0000_0XXX

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				VBG_data (3.3V) [11:0]											
				r											

Bits	31:12	保留, 未做定义
Bits	11:0	VBG_data (3.3V) [11:0]: ADC 在 3.3V 下校准后, 参考电压 3.3V, 采样所得 VBG 值

4. 电源管理（PWR）

4.1 供电方案

- VSS, VDDH = 1.8~5.5V: VDDH 引脚为 I/O 引脚和 RCH、PLL 等高频模块供电。
- VSSA, VDDA = 1.8~5.5V: 为复位模块、RC 振荡器、PLL、内部调压器（LDO）和模拟模块供电，内核数字部分由调压器供电（LDO），电压 1.5V。VDDA 电压必须大于或等于 VDDH 电压，并且要先于 VDDH 提供。VDDA 和 VSSA 可分别连接到 VDDH 和 VSS。

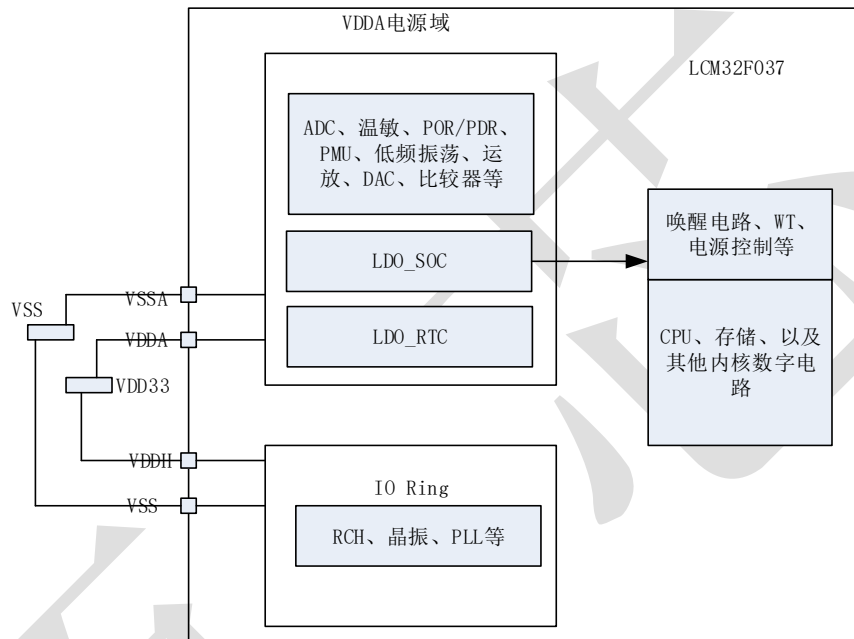


图 4-1 电源供电结构

LCM32F037 集成了两个 LDO 来供电，以达到最大优化电路的功耗，电源分配状态如下：

表 4-1 模块电源分配

	低功耗 LDO (LDO_RTC)	内核 LDO (LDO_SOC)
功能模块	RCL、OSCL	POR/PDR、ADC、DAC、运放、数字逻辑、存储等
管脚	-	所有 I/O

4.1.1 模拟电源 VDDA 供电的相关模块

如图 4-1 所示，内部的模拟模块一般通过电源 VDDA 单独供电，与数字部分的供电 VDDH 相对隔离，以带来更好的抗干扰性能。但是在一些封装当中，VDDA 和数字 VDDH 通过封装打线在一起，统一外部供电 VDD33 管脚输入电压。同样，模拟模块有单独的 VSSA 地线来供电，但一般 VSSA 在封装时会与其他的地线共同连接到同一个封装管脚。

当 VDDA 和 VDDH 单独供电时，要求 VDDA 电压大于等于 VDDH，外部连接到一起时，建议经过一些滤波电路再连接。

为了保证 VDDA 一定高于 VDDH，特别是在上下电过程中，可以在 VDDH 和 VDDA 间直接放一个肖特基二极管，以保证这个电压差。具体参考说明书确定可运行的电压差异。

4.1.2 电压调压器 (LDO)

LCM32F037 里包含两个 LDO，内核 LDO 用于系统运行的内核电源域，低功耗 LDO 用于常开/备份电源域。两个 LDO 在上电复位后处于工作状态。

内核 LDO 有两种操作模式：

- 主模式 (MR)，用于系统正常的运行操作，电压 1.5V。
- 低功耗模式 (LPR)，此时内核 LDO 的输出驱动能力最低可降到 10uA 左右。

如果系统处于超低功耗停机模式，则内核电路由低功耗 LDO 供电，寄存器和 SRAM 的内容保持，低功耗 LDO 的电压四挡可调，分别为 1.0V、1.1V、1.2V 和 1.5V。

4.2 电源检测

LCM32F037 内部集成了一个高精度的上电/掉电复位 (POR/PDR) 电路。这个复位电路在上电时始终处于工作状态，保证系统在供电高于 1.8V 后工作；当供电低于设定的阈值时，设备处于复位状态，而不必使用外部复位电路。

LCM32F037 内部还集成了一个可编程低压复位器 (LVR)，它监视 VDD 供电并与设定的阈值电压比较，当 VDD 低于阈值电压时，置设备于复位状态。LVR 默认是打开的，可通过软件关闭。LVR 支持 8 个低压复位点：1.6V、1.8V、2.0V、2.5V、2.8V、3.0V、3.5V、4.0V。当 MCU 的系统频率超过 72MHz 时，要求 MCU 供电大于 4V，因此建议设置 LVR 档位为 4.0V 作为保护；其他系统频率下，可根据供电配置合适的 LVR 档位。

LCM32F037 内部还集成了一个可编程电压监测器 (LVD)，它监视 VDD 供电并与设定的阈值电压比较，当 VDD 低于或高于阈值电压时产生中断，中断处理程序可以发出警告信息或将设备转入安全模式。LVD 默认是关闭的，需要通过软件开启。LVD 支持 8 个电压监测点：2.0V、2.2V、2.4V、2.7V、2.9V、3.1V、3.6V、4.5V。

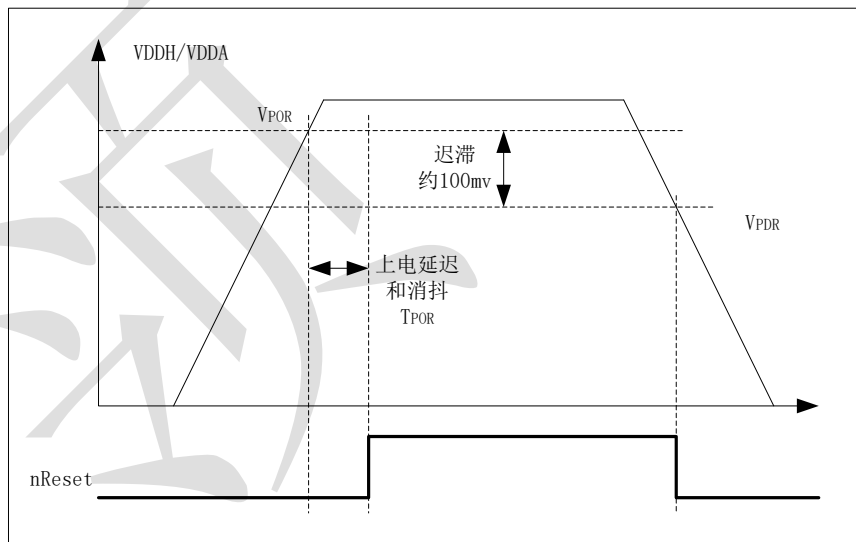


图 4-2 上电和下电复位图

4.3 低功耗模式

系统上电复位或者外部复位之后，默认处于运行模式，用户可以根据自身的环境应用，选择进入不同的低功耗模式来节省 MCU 功耗。LCM32F037 支持三种低功耗模式，可以在低功耗、短启动时间和可唤醒源之间达到最佳的平衡。

◆ 睡眠模式 (SLEEP)

在睡眠模式，只有 CPU 停止工作，所有外设处于工作状态并可在发生中断/事件时唤醒 CPU，内核 LDO 处于主模式。

◆ 停机模式 (STOP)

在保持 SRAM 和寄存器内容不丢失的情况下，停机模式可以达到极低的功耗。在停机模式下，内核 LDO 可以被配置为主模式或者低功耗模式。内核电源域的所有高频时钟全部被关掉，PLL、RCH 和 OSCH 被关闭，内核电源域里的各个模拟模块根据内核 LDO 的状态，可以通过软件关掉或者打开。

可以通过任何 EXTI 把微控制器从停机模式中唤醒，EXTI 可以是 16 个映射的外部 I/O 口之一、LVD、WT 闹钟或模拟比较器 ACMP 的输出。

◆ 超低功耗停机模式 (ULP STOP)

在保持 SRAM 和寄存器内容不丢失的情况下，超低功耗停机模式可以达到最低的功耗。在超低功耗停机模式下，内核电源由低功耗 LDO 提供。内核电源域的所有高频时钟全部被关掉，PLL、RCH 和 OSCH 被关闭，内核电源域里的各个模拟模块也全部被关掉。

超低功耗模式时，低功耗 LDO 输出电压有 4 档可选，当选择 1.0V 时，可进一步降低静态功耗。

可以通过任何 EXTI 把微控制器从超低功耗停机模式中唤醒，EXTI 信号可以是 16 个映射的外部 I/O 口之一、LVD 或 WT 闹钟的输出。

另外，通过降低系统的工作时钟可以显著降低工作电流，且 LCM32F037 中的各外设以及总线均有相应的时钟控制位，在不需要外设模块工作时，可以关闭其时钟以降低工作功耗。

表 4-2 低功耗模式总结

模式	进入方式	唤醒源	内核逻辑部分的状态	高压 VDD33 供电部分	调压器 LDO
睡眠模式 (SLEEP)	WFI	任何中断	CPU 时钟关闭，其他外设等不影响	无关	开启状态不变
	WFE	唤醒事件			
停机模式 (STOP)	ULPDS=0, SLEEPDEEP=1, WFI/WFE	EXTI、LVD、WT、ACMP	所有时钟关闭	模拟模块根据需求，软件配置是否关闭	开启状态，可根据 PWR_CR 的 LPDS 调整驱动能力
超级停机模式 (ULP STOP)	ULPDS=1, SLEEPDEEP=1, WFI/WFE	EXTI、LVD、WT	所有时钟关闭，FLASH 进入 Standby 模式	与内核相关的模拟模块必须关闭	低功耗 LDO 根据软件可配置 1.0V~1.5V

4.3.1 降低系统时钟频率

工作模式的时钟频率 (SYSCLK、HCLK 和 PCLK) 可以通过分频或者选择寄存器来控制，外设时钟也可以通过分频器在进入低功耗模式之前降低频率。

具体的时钟配置请参考时钟和复位控制章节。

4.3.2 外设时钟的关闭

在正常运行模式下，所有不同外设的时钟 (HCLK 或者 PCLK) 在任何时候都可以被直接关闭，以降低整个系统功耗。这些外设的时钟要在系统执行 WFI 或者 WFE 指令进入低功耗模式之前关闭，以节省低功耗模式的功耗。

外设时钟的配置请参考时钟和复位控制章节。

4.3.3 睡眠模式（SLEEP）

睡眠模式的进入

睡眠模式通过执行 WFI（等待中断）或者 WFE（等待事件）这两条指令进入，且通过 CPU 的系统控制寄存器中的 SLEEPONEXIT 位来决定进入方式：

- 立即睡眠：SLEEPONEXIT 设定为 0 时，MCU 在执行完 WFI 或者 WFE 指令后立刻进入睡眠模式；
- 退出后进入睡眠：SLEEPONEXIT 设定为 1 时，MCU 在退出当前最低优先级的中断服务程序后进入睡眠模式。

睡眠模式下，所有输入输出管脚保持其原有状态。

睡眠模式的退出

当使用 WFI 指令来进入睡眠模式时，可被嵌套向量中断控制器（NVIC）判别的中断会将系统从睡眠中唤醒。

当使用 WFE 指令进入睡眠模式时，任何事件均可以唤醒 MCU，有以下两种方式产生事件：

- 设置了 CPU 内部系统控制寄存器中的 SEVONPEND 位，且发生了外设的中断挂起。MCU 从 WFE 的睡眠状态恢复时，必须清除相应外设的中断挂起位，以及相应 NVIC 中的中断请求通道挂起位（位于 NVIC 中断清除挂起寄存器中）；
- 配置了外部或内部 EXTI，从 WFE 状态恢复后，MCU 不必清除外设中断挂起位以及 NVIC 的中断请求通道挂起位，因为相应的挂起位不会被设置。此模式的唤醒时间最短，没有额外的中断进入或者退出的等待时间。

表 4-3 立刻进入睡眠

立刻睡眠模式	描述
进入	WFI（等待中断）或者 WFE（等待事件）： <ul style="list-style-type: none"> ● SLEEPDEEP = 0 且 ● SLEEPONEXIT = 0 详情参考 CPU 的系统控制寄存器说明
退出	当用 WFI 作为进入指令： <ul style="list-style-type: none"> ● 中断：参考中断向量表 当用 WFE 作为进入指令： <ul style="list-style-type: none"> ● 唤醒事件：详情参考事件管理
唤醒延迟	无

表 4-4 退出后睡眠

退出后睡眠模式	描述
进入	WFI（等待中断）： <ul style="list-style-type: none"> ● SLEEPDEEP = 0 且 ● SLEEPONEXIT = 1 详情参考 CPU 的系统控制寄存器说明
退出	中断：参考中断向量表
唤醒延迟	无

4.3.4 停机模式（STOP）

停机模式基于 CPU 进入深度睡眠模式（DEEP SLEEP）以及关闭所有外设的时钟，此时数字电源域的所有时钟均被关闭，其中 PLL、内部 RCH 和外部高频晶振全部被关闭，SRAM 和寄存器的值保持不变。停机模式下的输入输出管脚也与正常运行模式保持相同状态。

停机模式的进入

表 4-5 列出了进入停机模式的方法。为进一步降低功耗，LCM32F037 的内核 LDO 可以调整驱动能力及其本身的静态电流，通过配置 PWR_LDOCR 寄存器中的 DRV[1:0]来调整，总共四挡可调。如果需要更低的功耗，则可以使用超低功耗停机模式。

在 FLASH 编程，或者 APB 外设访问时，停机模式会等待当前操作完成后进入。停机模式下，可以通过配置相关控制位来开启以下模块特性：

- 独立看门狗（IWDG）：IWDG 通过写其密钥寄存器或者硬件配置选项开启，一旦被开启，此模块就不能被关闭，除非复位整个系统。详见 IWDG 相关章节。
- 内部低频振荡（RCL）：通过 RCL 模块控制寄存器的 RCLEN 位配置开启，详情参考后续章节的 CHIPCTRL_RCLCFG 寄存器说明。
- 外部 32.768KHz 低频晶振：由寄存器 CHIPCTRL_BDCR 中的 OSCL_EN 位独立配置，STOP 模式下不强制关闭。
- ADC、DAC、OPA 和 ACMP 等模拟模块在停机模式下不会自动关闭，需要用户根据自身应用在软件中配置相关控制寄存器来关闭其工作状态，以防止静态电流的消耗。

停机模式的退出

表 4-5 列出了退出停机模式的方法。

当停机模式被一个中断或者事件唤醒退出时，系统时钟将被自动切换为内部 RCH 时钟。如果停机模式下，内核 LDO 处于低功耗模式，则唤醒时间会有一定延迟。保持内核 LDO 为主模式时，功耗会增加，但唤醒时间被缩短。

表 4-5 停机模式

停机模式	描述
进入	<p>WFI（等待中断）或者 WFE（等待事件）：</p> <ul style="list-style-type: none"> ● 设置 CPU 核心控制寄存器的 SLEEPDEEP = 1 ● 电源控制寄存器中（PWR_CR）的 ULPDS=0 ● 通过设置 PWR_CR 寄存器中的 LPDS 位选择内核 LDO 的工作模式 <p>注：在进入停机模式时，所有中断请求标志、外部中断标志和 WT 唤醒标志都应被清除，否则进入低功耗行为被忽略，软件继续执行后续指令。另外，如果软件在停机模式下要关闭外部高频晶振时，必须先将系统时钟切换到内部时钟 RCH，然后关闭外部晶振使能 OSCH_EN。如果进入停机模式时，外部时钟使能处于打开状态（OSCH_EN=1），则必须将时钟安全监测 CSS_EN 位打开，防止进入停机模式时外部时钟的发生错误。</p> <p>内核 LDO 在进入停机模式时，根据 LPDS 位的设置，有两种驱动能力选择，即为 0 选择低功耗模式，为 1 选择正常模式；此意味着 LDO 的驱动能力会因进入停机模式而发生改变，软件必须自行根据需求来恢复需要的驱动配置。</p>
退出	<p>当用 WFI 作为进入指令：</p> <ul style="list-style-type: none"> ● 任意被配置为中断模式的 EXTI 输入（NVIC 中对应的 EXTI 中断向量

	必须也被使能), 参见中断向量表 9-1 当用 WFE 作为进入指令: <ul style="list-style-type: none"> ● 任意被配置为事件的 EXTI 输入, 参见事件管理章节 注: 从停机模式被唤醒后, 内核 LDO 的驱动能力自动恢复到驱动能力为正常模式 (50uA 输出驱动)。
唤醒延迟	内部 RCH 起振时间+LDO 从低功耗模式唤醒时间

4.3.5 超低功耗停机模式 (ULP STOP)

此模式是停机模式的进一步降低功耗的补充, 进入方法也基本相同, 通过配置相关寄存器来区分停机模式与超低功耗停机模式, 唤醒方式也基本相同。

与停机模式不同的是, 在超低功耗停机模式中, 系统选择低功耗 LDO 供电, 且此 LDO 通过寄存器 PWR_LDOCR 中的 VSEL[1:0]来设定 1.0~1.5V 的供电范围, 从而进一步降低系统静态功耗。下表列出超低功耗停机模式的工作方式:

表 4-6 超低功耗停机模式

停机模式	描述
进入	WFI (等待中断) 或者 WFE (等待事件): <ul style="list-style-type: none"> ● 设置 CPU 核心控制寄存器的 SLEEPDEEP = 1 且 ● 电源控制寄存器中 (PWR_CR) 的 ULPDS=1 ● MCU 自动切换到低功耗 LDO ● 通过设置 PWR_LDOCR 寄存器中的 VSEL 位来选择电压 注: 数字模块的所有时钟必须全部停止, 模拟模块提前关闭, 以降低功耗
退出	当用 WFI 作为进入指令: <ul style="list-style-type: none"> ● 任意被配置为中断模式的 EXTI 输入 (NVIC 中对应的 EXTI 中断向量必须也被使能), 参见中断向量表 9-1 当用 WFE 作为进入指令: <ul style="list-style-type: none"> ● 任意被配置为事件的 EXTI 输入, 参见事件管理章节 注: 从停机模式被唤醒后, 内核 LDO 的驱动能力自动恢复到驱动能力为正常模式 (50uA 输出驱动)。
唤醒延迟	内部 RCH 起振时间+LDO 从低功耗模式唤醒时间

4.3.6 低功耗模式的 WT 唤醒

WT 可以通过其闹钟设定, 将 MCU 从低功耗模式下唤醒, 通过寄存器 CHIPCTRL_BDCR 的 WT_SEL[1:0]选择 WT 时钟源, 其中的两个时钟源作为唤醒工作时钟:

- 外部低功耗 32.768KHz 晶振 (OSCL)

此时钟模块提供了精准的计数时间, 且功耗极低 (典型条件下, 增加的电流小于 1uA); 为实现起振时间和功耗的均衡, 客户可以调整 CHIPCTRL_BDCR 寄存器的 OSCL_GAIN[1:0]选择外部晶振的驱动, 例如在晶振使能振荡时选择高驱动, 在起振后切换到低驱动。

- 内部低频振荡 (RCL)

使用内部 RCL 可以省去外部 32.768KHz 晶振, 从而节省成本, RCL 的功耗极低 (小于 0.5uA), 但整体精度较差 (不同温度、电压范围下的差异约±10%左右)

通过 WT 从低功耗模式唤醒时, 需要配置:

- 将外部中断 EXTI 的线 17 配置为上升沿有效
- 设置 WT 以及产生闹钟时刻

4.4 PWR 寄存器描述

此寄存器组的基地址为 0x4001_7000，通过 APB1 总线访问，只能通过 WORD 方式读写访问。所有 PWR 寄存器被保护，写操作时需要先解锁密钥寄存器 PWR_KEY (ChipCtrl_KEY)，解锁方式为写 0x8721_4365 到密钥寄存器，且在随后的 32 个时钟周期内完成配置过程，因此需要注意不能被中断打断写入过程，并通过读回确定写入完成。

表 4-7 电源配置相关寄存器表

名称	说明	读写权限	复位值	字节地址
PWR_CFG	电源配置寄存器	R/W	0x0000_0001	0x4001_7004
PWR_KEY	电源密钥寄存器	R/W	0x0000_0002	0x4001_7020
PWR_CR	电源模式控制寄存器	R/W	0x0000_0000	0x4001_7030
PWR_LDOCR	电源 LDO 配置寄存器	R/W	0x0000_0004	0x4001_703C

注：x 表示不确定；-表示未实现；u 表示不受复位影响；c 表示该位复位值因复位源而异；R 只读；W 只写；R-0 表示只读且读 0；R/W 表示可读写（以后章节同上述）。

4.4.1 电源配置寄存器（PWR_CFG）

地址偏移：0x04

复位值：0x0000_0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	OP_ISEL[2:0]			Res.		DAC1_VREF_SEL1	DAC0_VREF_SEL1	DAC1_VREF_SELO	DAC1_VREF_EN	DAC0_VREF_SELO	DAC0_VREF_EN	ADPREF[1:0]		PMU_VRH_SEL1	PMU_VRH_SELO
	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
Res.		VBG_BUFOUT_EN	LVDS[2:0]			LVDES	LVDEN	Res.					LVRS[2:0]		LVREN
		rw	rw			rw	rw						rw		rw

Bit	31	保留，必须保持复位值
Bits	30:28	OP_ISEL[2:0] ：运放偏置电流控制位，Bit[30:28]对应 OPA2/OPA1/OPA0
		0：偏置电流 1uA（默认）
		1：偏置电流 2uA
Bits	27:26	保留，必须保持复位值
Bit	25	与 Bit23 组成 DAC1_VREF_SEL[1:0] ：芯片电源输出给 DAC1 的参考电压选择
		00：2.5V（默认）
		01：4V
		1x：VDDA
Bit	24	与 Bit21 组成 DAC0_VREF_SEL[1:0] ：芯片电源输出给 DAC0 的参考电压选择
		00：2.5V（默认）
		01：4V

	1x: VDDA
Bit 23	参考 Bit25 的描述
Bit 22	DAC1_VREF_EN: 芯片电源输出给 DAC1 的参考电压使能位
	0: 芯片电源输出给 DAC1 的参考电压不使能 (默认)
	1: 芯片电源输出给 DAC1 的参考电压使能
Bit 21	参考 Bit24 的描述
Bit 20	DAC0_VREF_EN: 芯片电源输出给 DAC0 的参考电压使能位
	0: 芯片电源输出给 DAC0 的参考电压不使能 (默认)
	1: 芯片电源输出给 DAC0 的参考电压使能
Bits 19:18	ADPREF[1:0]: 芯片电源输出给 ADC/OPA 的参考电压选择
	00: VDDA (默认)
	01: PMU_VRH
	10: 由外部 I/O (PA0) 输入参考电压
	11: 选择 PMU_VRH 为参考电压, 同时外部 I/O (PA0) 可打开并外挂电容
Bit 17	PMU_VRH_SEL1: 当 PMU_VRH_SEL0 = 0 时, PMU_VRH 的电压选择
	0: PMU_VRH = 2.5V (默认)
	1: PMU_VRH = 4V
Bit 16	PMU_VRH_SEL0: PMU_VRH 的电压选择
	0: PMU_VRH = 2.5V/4V, 由 PMU_VRH_SEL1 确定 (默认)
	1: PMU_VRH = VBG
Bits 15:14	保留, 必须保持复位值
Bit 13	VBG_BUFOUT_EN: VBG 输出使能位
	0: VBG 不输出到 I/O (默认)
	1: VBG 可以从外部 I/O (PB5) 输出
Bits 12:10	LVDS[2:0]: LVD 低压检测电平选择
	000: 2.0V (默认)
	001: 2.2V
	010: 2.4V
	011: 2.7V
	100: 2.9V
	101: 3.1V
	110: 3.6V
	111: 4.5V
Bit 9	LVDES: LVD 电压检测外部输入选择
	0: 选择检测内部电压 (默认)
	1: 选择检测外部电源
Bit 8	LVDEN: LVD 使能控制
	0: LVD 禁止 (默认)
	1: LVD 使能
Bits 7:4	保留, 必须保持复位值
Bits 3:1	LVR[2:0]: LVR 低压复位点选择
	000: 1.6V (默认)
	001: 1.8V

	010: 2.0V
	011: 2.5V
	100: 2.8V
	101: 3.0V
	110: 3.5V
	111: 4.0V
Bit 0	LVREN : LVR 使能位
	0: LVR 禁止
	1: LVR 使能 (默认)

4.4.2 电源密钥寄存器 (PWR_KEY)

地址偏移: 0x20

复位值: 0x0000_0002

所有电源相关寄存器受到密钥寄存器保护, 需要将 16 进制数 0x8721_4365 写入密钥寄存器才能解锁; 当其他数据写入寄存器, 或者 32 个系统时钟周期后, 寄存器恢复到写保护状态。

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			LOCK_CNT[4:0]					Res.						LOCK	WR_ERR
														r	r

写入:

Bits 31:0	KEY[31:0] : 写入密钥以解锁或者锁定寄存器
-----------	-----------------------------------

读出:

Bits 31:13	保留, 必须保持复位值
Bits 12:8	LOCK_CNT[4:0] : 锁定窗口计数值
Bits 7:2	保留, 必须保持复位值
Bit 1	LOCK : 锁定状态
	0: 写保护已被解除
	1: 写保护锁定 (默认)
Bit 0	WR_ERR : 寄存器写错误标志
	0: 没有写保护错误 (解除写保护之后成功写入寄存器) (默认)
	1: 写保护状态下的寄存器写入操作会被忽略

4.4.3 电源模式控制寄存器 (PWR_CR)

地址偏移: 0x30

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.											ULPDS	FPDS	PMUEN	Res.	LPDS
											rw	rw	rw		rw

Bits	31:5	保留, 必须保持复位值
Bit	4	ULPDS: 超低功耗停机模式
		由软件清零和置位, 与控制位 LPDS 协同工作
		0: 不使能超低功耗 (默认)
		1: 使能超低功耗
Bit	3	FPDS: 停机模式下 FLASH 模块的停止控制
		由软件清零和置位, 与控制位 LPDS 协同工作
		0: FLASH 模块停机模式下不进入其深度待机模式 (默认)
		1: FLASH 模块在停机模式下进入深度待机模式
Bit	2	PMUEN: 低功耗模式使能
		由软件清零和置位, 与 CPU 的深度睡眠设定协同工作
		0: CPU 设定进入深度睡眠时, 不进入相应低功耗模式 (默认)
		1: CPU 设定进入深度睡眠时, 进入相应低功耗模式
Bit	1	保留, 必须保持复位值
Bit	0	LPDS: 停机模式 STOP 控制位
		由软件清零和置位, 与控制位 ULPDS 协同工作
		0: 停机模式下, 内核 LDO 处于正常供电模式 (默认)
		1: 停机模式下, 内核 LDO 处于低功耗模式

4.4.4 电源 LDO 配置寄存器 (PWR_LDOCR)

地址偏移: 0x3C

复位值: 0x0000_0004

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.										VSEL[1:0]		DRV[1:0]		Res.	EXTVDD
										rw		rw			rw

Bits 31:6	保留，必须保持复位值
Bits 5:4	VSEL[1:0]: 低功耗 LDO 输出电平选择
	00: 1.5V (默认)
	01: 1.2V
	10: 1.1V
	11: 1.0V
Bits 3:2	DRV[1:0]: 内核 LDO 驱动模式配置
	00: 10uA 自耗电电流低功耗模式
	01: 50uA 自耗电电流运行模式 (默认)
	10: 150uA 自耗电电流运行模式
	11: 200uA 自耗电电流运行模式
Bit 1	保留，必须保持复位值
Bit 0	EXTVDD: 内核 LDO 选择外部输入
	0: 内核电源由内核 LDO 供电 (默认)
	1: 内核电源由外部输入供电

5. 时钟和复位控制（RCC）

5.1 复位

LCM32F037 支持 3 种类型的复位，分别是系统复位、电源复位和模块的软件复位。

5.1.1 电源复位

当以下事件中之一发生时，产生电源复位：

- 上电/掉电复位（POR/PDR）
- 低压检测复位，当电压低于原先设定的值时产生复位

电源复位将复位所有的寄存器。

5.1.2 系统复位

除了芯片控制器中的复位状态寄存器 RSTCSR、BD 控制寄存器 BDCR 和寄存器 BGR，系统复位将复位所有寄存器到默认值，当以下事件发生时，将产生系统复位：

1. NRST 引脚上的低电平（外部复位）
2. 窗式看门狗计数终止（WWDG 复位）
3. 独立看门狗计数终止（IWDG 复位）
4. 软件复位（SW 复位）
5. 电源复位（电源复位也会产生系统复位）

当系统产生复位后，可以通过查询复位状态寄存器 RSTCSR 确认复位源，软件复位通过设置 ARM 处理器中 AIRC 的 SYSRESETREQ 位产生。

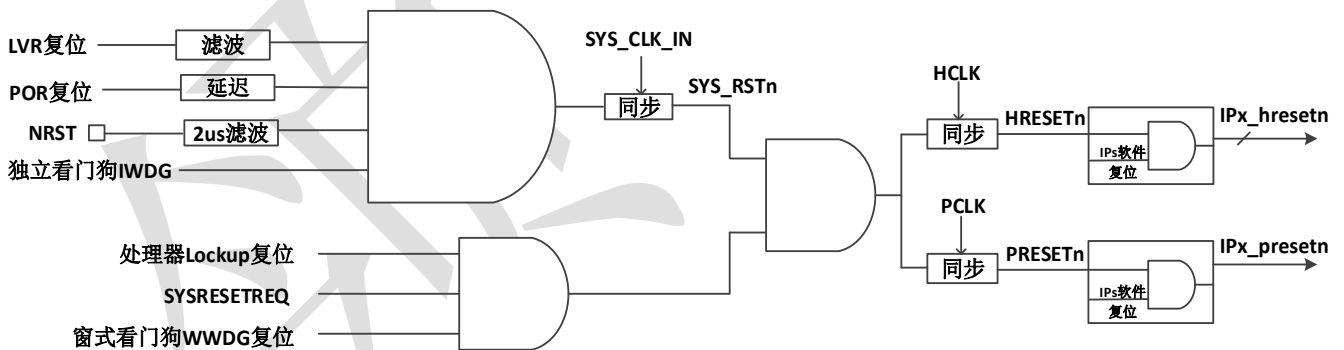


图 5-1 复位框图

5.2 时钟系统

系统时钟源的选择是在启动时进行，复位时内部 16MHz 的 RC 振荡器 RCH 被选为默认时钟。随后可以选择外部的具失效监控的 4~16MHz 的晶振时钟 OSCH 或者 PLL 时钟。时钟安全系统（CSS）可以通过软件配置（ChipCtrl_CTRL 寄存器中的 CSS_EN 位）被激活，一旦被激活，时钟监控器将在 OSCH 时钟启动并稳定后被使能，并在 OSCH 时钟关闭后关闭。当 CSS 使能且 OSCH 时钟出现故障时，CSS 产生中断，它连接到处理器 NMI 中断。NMI 将被不断执行，直到 CSS 安全状态位被清除。因此，在 NMI 的处理程序中必须通过配置 CHIPCTRL_STS 寄存器的 CSS_STS 位来清除 CSS 中断。

如果 OSCH 直接或间接地作为系统时钟（间接指 OSCH 是 PLL 时钟的来源且 PLL 时钟作为系统时钟），

在进入低功耗模式（STOP/ULP STOP）时，系统会自动激活 CSS 并把系统时钟源自动切换到 RCH 时钟，随后再关闭它们（RCH、OSCH 和 PLL 时钟）。

OSCH 和 OSCL 复用相同的 I/O（PF0 和 PF1），因此实际使用时，不能同时打开。

RCH、OSCH、PLL 和几个时钟分频器一起用来配置 AHB 和 APB0/1 总线的频率。AHB 和 APB 总线所支持的最高频率为 96MHz。LCM32F037 的时钟框图如图 5-2 所示。

除了以下模块以外，所有的外设时钟都来自总线时钟：

- Flash 编程接口时钟始终采用内部 RCH 时钟
- ADC 工作时钟可以通过配置选择 APB 总线时钟 PCLK1、PLL 时钟以及它们的分频
- UART、I2C 和 SSP 的工作时钟源可通过软件配置，可选择内部 RCH、外部晶振 OSCH、相应 APB 总线时钟 PCLK0/1 以及系统时钟
- WT 时钟可选择内部低频时钟 RCL、外部低频时钟 OSCL 或者外部 OSCH 的固定分频时钟

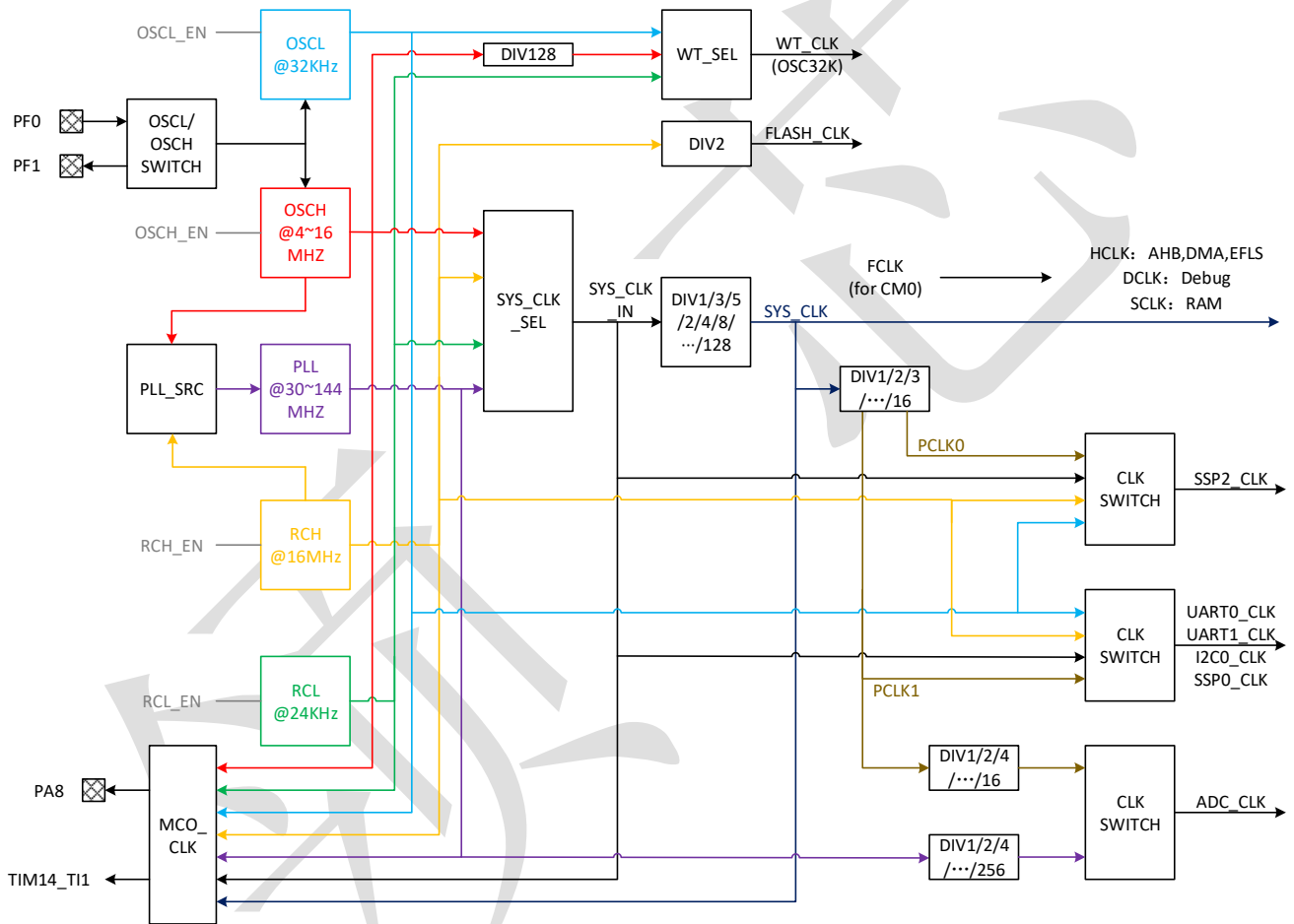


图 5-2 LCM32F037 时钟框图

时钟源切换

复位时内部 RCH 被选为默认的系统时钟源，如果想切换到其他时钟，首先需要进行相应的时钟配置并使能相应的时钟（RCH、OSCH 和 PLL 使能位在时钟配置寄存器 CLKCFG，RCL 使能位在 RCL 配置寄存器 RCLCFG，OSCL 使能位在 BD 控制寄存器 BDCR），等待相应时钟稳定（查看状态寄存器 STS 中高 16 位相关时钟状态）后，再配置系统时钟选择寄存器 SYS_SRC_SEL，等待系统时钟稳定后时钟切换完毕。在选择 PLL 时钟作为系统时钟源时，要提前配置好 PLL 的时钟输入（RCH 或 OSCH）并稳定后，再进行 PLL 时钟的相关配置。下面以系统时钟源从 RCH 时钟切换到以 OSCH 为输入的 PLL 时钟为例，给出配置示例：

1. 配置 OSCHCFG 寄存器中的 OSCH 时钟特性

2. 配置 CLKCFG 寄存器中的 OSCH_EN=1，使能 OSCH
3. 等待 STS 寄存器中的 OSCH_STB=1，表示 OSCH 开启且已经稳定
4. 配置 PLL 寄存器中的 PLL 时钟特性，PLL 输入选择为 OSCH
5. 配置 CLKCFG 寄存器中的 PLL_EN=1，使能 PLL
6. 等待 STS 寄存器中的 PLL_LOCK=1，表示 PLL 开启且已经稳定
7. 配置 CLKCFG 寄存器中的 SYS_SRC_SEL=2'b10，切换系统时钟源为 PLL
8. 等待 STS 寄存器中的 CLKMUX_LOCK=1，表示系统时钟源切换成功

在系统运行过程中，各个时钟稳定状态的变化情况会保存在状态寄存器 STS 中，其中 OSCL、OSCH 和 PLL 时钟的状态极性可配置。此外，时钟使能和系统时钟源切换出现错误时，也会将错误状态保存在状态寄存器 STS 中。以上各个状态位在相应的状态中断使能（芯片控制寄存器 ChipCtrl_CTRL[6:0]）有效后，可以产生对应的中断。

5.3 寄存器描述

时钟与复位相关的控制功能在 LCM32F037 中会被分配到不同的模块来实现，因此，对应寄存器的访问地址较为分散，分布在系统控制（AHB0 基地址 0x4800_7000）、以及独立控制的 ChipCtrl 模块（APB1 基址 0x4001_7000，只允许字访问）。

所有相关寄存器被保护，必须先通过写 ChipCtrl_KEY（PWR_KEY）寄存器或者 SysCtrl_KEY 寄存器来解锁写入操作，且在随后的 32 个时钟周期内完成配置过程，因此需要注意不能被中断打断写入过程，并通过读回确定写入完成。

5.3.1 芯片配置寄存器组（ChipCtrl）

此寄存器组的基地址为 0x4001_7000，通过 APB1 总线访问，只能通过 WORD 方式读写访问。所有 ChipCtrl 寄存器受到密钥保护，写操作时需要先解锁密钥寄存器 ChipCtrl_KEY，解锁方式为写 0x87214365 到密钥寄存器。

表 5-1 芯片配置相关寄存器表

名称	说明	读写权限	复位值	字节地址
ChipCtrl_CLKCFG	时钟配置寄存器	R/W	0x0008_0000	0x4001_7000
ChipCtrl_PWR	低压复位及电源配置寄存器	R/W	0x0000_0001	0x4001_7004
ChipCtrl_CTRL	芯片控制寄存器	R/W	0x0000_0000	0x4001_7008
ChipCtrl_PLL	PLL 配置寄存器	R/W	0x0000_115E	0x4001_700C
ChipCtrl_RCHCFG	RCH 配置寄存器	R/W	0x0000_0XXX	0x4001_7010
ChipCtrl_OSCHCFG	OSCH 配置寄存器	R/W	0x0000_0007	0x4001_7014
ChipCtrl_STS	状态寄存器	R/W	0x00C3_0003	0x4001_7018
ChipCtrl_RSTCSR	复位状态寄存器	R/W	0x0000_000X	0x4001_701C
ChipCtrl_KEY	芯片密钥寄存器	R/W	0x0000_0002	0x4001_7020
ChipCtrl_RCLCFG	RCL 配置寄存器	R/W	0x0000_00XX	0x4001_7024
ChipCtrl_BDCR	BD 控制寄存器	R/W	0x0000_7004	0x4001_7028
ChipCtrl_BGR	BGR 配置寄存器	R/W	0x0000_00XX	0x4001_702C

注：x 表示不确定；- 表示未实现；u 表示不受复位影响；c 表示该位复位值因复位源而异；R 只读；W 只写；R-0 表示只读且读 0；R/W 表示可读写（以后章节同上述）。

5.3.1.1 时钟配置寄存器 (ChipCtrl_CLKCFG)

地址偏移: 0x00

复位值: 0x0008_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.										SYS_SRC_SEL [1:0]		RCH _EN	Res.	OSCH _EN	PLL _EN
										rw		rw		rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
APB1_DIV[3:0]				APB0_DIV[3:0]				SYS_CLK_SEL[7:0]							
rw				rw				rw							

Bits	31:22	保留, 必须保持复位值
Bits	21:20	SYS_SRC_SEL[1:0]: 系统时钟源选择
		00: 选择内部 RCH, SYS_CLK_IN = RCH_CLK (默认)
		01: 选择外部高振 OSCH, SYS_CLK_IN = OSCH_CLK
		10: 选择 PLL 输出, SYS_CLK_IN = PLL_CLK
		11: 选择内部低振 RCL, SYS_CLK_IN = RCL_CLK
Bit	19	RCH_EN: RCH 使能
		0: RCH 关闭
		1: RCH 使能 (默认)
Bit	18	保留, 必须保持复位值
Bit	17	OSCH_EN: 外部高振使能
		0: OSCH 关闭 (默认)
		1: OSCH 使能
Bit	16	PLL_EN: PLL 使能
		0: PLL 关闭 (默认)
		1: PLL 使能
Bits	15:12	APB1_DIV[3:0]: APB1 总线的 PCLK1 时钟分频选择
		$PCLK1 = SYS_CLK / (APB1_DIV[3:0] + 1)$
		0000: PCLK1 = SYS_CLK (默认)
		0001: PCLK1 = SYS_CLK/2
		...
		1110: PCLK1 = SYS_CLK/15
		1111: PCLK1 = SYS_CLK/16
Bits	11:8	APB0_DIV[3:0]: APB0 总线的 PCLK0 时钟分频选择
		$PCLK0 = SYS_CLK / (APB0_DIV[3:0] + 1)$
		0000: PCLK0 = SYS_CLK (默认)
		0001: PCLK0 = SYS_CLK/2
		...
		1110: PCLK0 = SYS_CLK/15
		1111: PCLK0 = SYS_CLK/16

Bits 7:0	SYS_CLK_SEL[7:0]: 系统时钟频率选择
	00xx_xxxx: SYS_CLK = SYS_CLK_IN (默认)
	01xx_xxxx: SYS_CLK = SYS_CLK_IN/(2*(SYS_CLK_SEL[5:0]+1))
	10xx_xxxx: SYS_CLK = SYS_CLK_IN/3
	11xx_xxxx: SYS_CLK = SYS_CLK_IN/5

5.3.1.2 低压复位及电源配置寄存器 (ChipCtrl_PWR)

地址偏移: 0x04

复位值: 0x0000_0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	OP_ISEL[2:0]			Res.		DAC1_VREF_SEL1	DAC0_VREF_SEL1	DAC1_VREF_SELO	DAC1_VREF_EN	DAC0_VREF_SELO	DAC0_VREF_EN	ADPREF[1:0]		PMU_VRH_SEL1	PMU_VRH_SELO
	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
Res.		VBG_BUFOUT_EN		LVDS[2:0]			LVDES	LVDEN	Res.				LVRS[2:0]			LVREN
		rw		rw			rw	rw					rw			rw

Bit 31	保留, 必须保持复位值
Bits 30:28	OP_ISEL[2:0]: 运放偏置电流控制位, Bit[30:28]对应 OPA2/OPA1/OPA0
	0: 偏置电流 1uA (默认)
	1: 偏置电流 2uA
Bits 27:26	保留, 必须保持复位值
Bit 25	与 Bit23 组成 DAC1_VREF_SEL[1:0]: 芯片电源输出给 DAC1 的参考电压选择
	00: 2.5V (默认)
	01: 4V
	1x: VDDA
Bit 24	与 Bit21 组成 DAC0_VREF_SEL[1:0]: 芯片电源输出给 DAC0 的参考电压选择
	00: 2.5V (默认)
	01: 4V
	1x: VDDA
Bit 23	参考 Bit25 的描述
Bit 22	DAC1_VREF_EN: 芯片电源输出给 DAC1 的参考电压使能位
	0: 芯片电源输出给 DAC1 的参考电压不使能 (默认)
	1: 芯片电源输出给 DAC1 的参考电压使能
Bit 21	参考 Bit24 的描述
Bit 20	DAC0_VREF_EN: 芯片电源输出给 DAC0 的参考电压使能位
	0: 芯片电源输出给 DAC0 的参考电压不使能 (默认)
	1: 芯片电源输出给 DAC0 的参考电压使能
Bits 19:18	ADPREF[1:0]: 芯片电源输出给 ADC/OPA 的参考电压选择
	00: VDDA (默认)
	01: PMU_VRH
	10: 由外部 I/O (PA0) 输入参考电压

	11: 选择 PMU_VRH 为参考电压, 同时外部 I/O (PA0) 可打开并外挂电容
Bit 17	PMU_VRH_SEL1: 当 PMU_VRH_SELO = 0 时, PMU_VRH 的电压选择
	0: PMU_VRH = 2.5V (默认)
	1: PMU_VRH = 4V
Bit 16	PMU_VRH_SELO: PMU_VRH 的电压选择
	0: PMU_VRH = 2.5V/4V, 由 PMU_VRH_SEL1 确定 (默认)
	1: PMU_VRH = VBG
Bits 15:14	保留, 必须保持复位值
Bit 13	VBG_BUFOUT_EN: VBG 输出使能位
	0: VBG 不输出到 I/O (默认)
	1: VBG 可以从外部 I/O (PB5) 输出
Bits 12:10	LVDS[2:0]: LVD 低压检测电平选择
	000: 2.0V (默认)
	001: 2.2V
	010: 2.4V
	011: 2.7V
	100: 2.9V
	101: 3.1V
	110: 3.6V
	111: 4.5V
Bit 9	LVDES: LVD 电压检测外部输入选择
	0: 选择检测内部电压 (默认)
	1: 选择检测外部电源
Bit 8	LVDEN: LVD 使能控制
	0: LVD 禁止 (默认)
	1: LVD 使能
Bits 7:4	保留, 必须保持复位值
Bits 3:1	LVR[2:0]: LVR 低压复位点选择
	000: 1.6V (默认)
	001: 1.8V
	010: 2.0V
	011: 2.5V
	100: 2.8V
	101: 3.0V
	110: 3.5V
	111: 4.0V
Bit 0	LVREN: LVR 使能位
	0: LVR 禁止
	1: LVR 使能 (默认)

5.3.1.3 芯片控制寄存器 (ChipCtrl_CTRL)

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	
CSS _EN	LOCK UPEN	LVD_ LOCK	Res.			PLL _IP	OSCH _IP	OSCL _IP	Res.							
rw	rw	rw			rw	rw	rw									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.									CLKEN_ ERR_IE	CLKMUX _ERR_IE	PLL _IE	OSCH _IE	OSCL _IE	RCH _IE	RCL _IE
									rw	rw	rw	rw	rw	rw	rw

Bit	31	CSS_EN: 时钟安全系统使能
		该位由软件设置和清除。当 CSS_EN 被设置为 1 时，如果 OSCH 晶振被直接或者间接作为系统时钟源（间接是指系统 OSCH 作为 PLL 的源时钟，而 PLL 输出时钟作为系统时钟），如果 OSCH 失效，系统时钟将自动切换到内部 RCH 时钟，同时 OSCH 时钟被关闭。如果 OSCH 时钟作为 PLL 的输入时钟，当 OSCH 失效发生时，PLL 也将被关闭。
		0: 时钟安全系统无效（默认）
		1: 时钟安全系统起效
Bit	30	LOCKUPEN: 处理器 LOCKUP（硬件故障）控制 TIM 刹车使能
		该位可由软件设置但必须由系统复位清除。该位用于控制处理器 LOCKUP 输出与 TIM1/15/16/17 的刹车输入的连接关系。
		0: 处理器 LOCKUP 输出与 TIM1/15/16/17 的刹车输入无连接（默认）
		1: 处理器 LOCKUP 输出与 TIM1/15/16/17 的刹车输入连接
Bit	29	LVD_LOCK: LVD 锁存（LOCK）使能
		该位由软件设置但必须由系统复位清除。该位用于控制 LVD 检测结果与 TIM1/15/16/17 的刹车输入的连接关系。此外，当该位为 1 时，ChipCtrl_PWR 寄存器中的 LVD 相关配置位禁止写。
		0: LVD 检测结果与 TIM1/15/16/17 的刹车输入无连接（默认）
		1: LVD 检测结果与 TIM1/15/16/17 的刹车输入连接
Bits	28:27	保留，必须保持复位值
Bit	26	PLL_IP: 设置 PLL 锁存状态寄存器的极性
		0: 当 PLL 由稳定到不稳定时设置 PLL 状态位（默认）
		1: 当 PLL 由不稳定到稳定时设置 PLL 状态位
Bit	25	OSCH_IP: 设置 OSCH 状态寄存器的极性
		0: 当 OSCH 由稳定到不稳定时设置 OSCH 状态位（默认）
		1: 当 OSCH 由不稳定到稳定时设置 OSCH 状态位
Bit	24	OSCL_IP: 设置 OSCL 状态寄存器的极性
		0: 当 OSCL 由稳定到不稳定时设置 OSCL 状态位（默认）
		1: 当 OSCL 由不稳定到稳定时设置 OSCL 状态位
Bits	23:7	保留，必须保持复位值
Bit	6	CLKEN_ERR_IE: 时钟使能错误中断使能
		0: 中断不使能（默认）
		1: 中断使能

Bit 5	CLKMUX_ERR_IE: 系统时钟源切换错误中断使能
	0: 中断不使能 (默认)
	1: 中断使能
Bit 4	PLL_IE: PLL 状态变化中断使能
	0: 中断不使能 (默认)
	1: 中断使能
Bit 3	OSCH_IE: OSCH 状态变化中断使能
	0: 中断不使能 (默认)
	1: 中断使能
Bit 2	OSCL_IE: OSCL 状态变化中断使能
	0: 中断不使能 (默认)
	1: 中断使能
Bit 1	RCH_IE: RCH 状态变化中断使能
	0: 中断不使能 (默认)
	1: 中断使能
Bit 0	RCL_IE: RCL 状态中断使能
	0: 中断不使能 (默认)
	1: 中断使能

5.3.1.4 PLL 配置寄存器 (ChipCtrl_PLL)

地址偏移: 0x0C

复位值: 0x0000_115E

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		PLL _OD	Res.			PLL_DM[1:0]	Res.		PLL_DN[4:0]				PLL _LS	PLL _SRC	
		rw				rw			rw				rw	rw	

Bits 31:13	保留, 必须保持复位值
Bit 12	PLL_OD: PLL 输出时钟二分频
	0: PLL 时钟直接输出
	1: PLL 时钟 2 分频后输出 (默认)
	PLL 输出频率: $F_{PLL_CLK} = (F_{INPUT_CLK} * (DN + 1)) / ((DM + 1) * (OD + 1) * 2)$
Bits 11:10	保留, 必须保持复位值
Bits 9:8	PLL_DM[1:0]: PLL 输入时钟分频系数
	00: 输入时钟 1 分频
	01: 输入时钟 2 分频 (默认)
	10: 输入时钟 3 分频
	11: 输入时钟 4 分频
Bit 7	保留, 必须保持复位值

Bits 6:2	PLL_DN[4:0]: PLL 反馈时钟分频系数 (默认值 10111)
	00000: 反馈时钟 1 分频
	00001: 反馈时钟 2 分频
	00010: 反馈时钟 3 分频
	00011: 反馈时钟 4 分频
	...
	11111: 反馈时钟 32 分频
Bit 1	PLL_LS: PLL 锁存状态选择
	0: PLL 的锁存状态直接由 PLL 产生
	1: PLL 的锁存状态经过 256 个 PLL 时钟之后产生 (默认)
Bit 0	PLL_SRC: PLL 输入时钟源选择
	0: RCH (默认)
	1: OSCH

注: PLL 倍频输出时钟最大为 144Mhz, 之后通过 OD 分频后输出给 SYSCLK, SYSCLK 不能超过 96M, 建议运行最高频率 96Mhz 时, 需要打开相应的取指延迟, 详见 FLASH_LATENCY 的介绍。

5.3.1.5 RCH 配置寄存器 (ChipCtrl_RCHCFG)

地址偏移: 0x10

复位值: 0x0000_0XXX

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

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

Bits 31:12	保留, 必须保持复位值
Bits 11:8	RCH_PT[3:0]: RCH 温度特性调准值
Bits 7:0	RCH_TRIM[7:0]: RCH 频率 TRIM 值
	00000000: RCH 频率最低
	...
	11111111: RCH 频率最高

注: 初始值由存储器配置 (Option Byte) 直接加载, 用户可以根据应用需要配置。

5.3.1.6 OSCH 配置寄存器 (ChipCtrl_OSCHCFG)

地址偏移: 0x14

复位值: 0x0000_0007

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								OSCH _BYP	OSCH _REN	OSCH_TS[2:0]			OSCH_GAIN[2:0]		
								rw	rw	rw			rw		

Bits 31:8	保留，必须保持复位值
Bit 7	OSCH_BYP : OSCH 晶振旁路 (BYPASS)
	该位只有在 OSCH 和 OSCL 不使能的情况下才可以进行配置。
	0: OSCH 晶振不旁路 (默认)
	1: OSCH 晶振旁路
Bit 6	OSCH_REN : OSCH 晶振反馈电阻选择
	0: 选择内部反馈电阻 (默认)
	1: 选择外部反馈电阻
Bits 5:3	OSCH_TS[2:0] : OSCH 晶振稳定时间选择 (假设 OSCH 频率为 16MHz)
	000: 等待 4096us (默认)
	001: 等待 3584us
	010: 等待 3072us
	011: 等待 2556us
	100: 等待 2048us
	101: 等待 1534us
	110: 等待 1024us
	111: 等待 512us
Bits 2:0	OSCH_GAIN[2:0] : 外部晶振增益配置，增益越大，起振时间越快但功耗也越大
	建议在起振的时候配置高增益，起振后配置低增益。
	000: 低增益
	...
	111: 高增益 (默认)

5.3.1.7 状态寄存器 (ChipCtrl_STS)

地址偏移: 0x18

复位值: 0x00C3_0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.							LVD FLG	CLKMUX _LOCK	SYS_CLK _LOCK	Res.	PLL_ LOCK	OSCH _STB	OSCL _STB	RCH_ STB	RCL_ STB
							r	r	r		r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								CSS_ STS	CLKEN_ ERR	CLKMUX_ ERR	PLL _STS	OSCH _STS	OSCL _STS	RCH_ STS	RCL _STS
								rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:25	保留，必须保持复位值
Bit 24	LVDLFG : LVD 低压监测标志

	该位由硬件设置和清除，当 ChipCtrl_PWR.LVDEN 配置后有效。
	0: VDDH/VDDA 电压值高于监测电压（监测电压值由 ChipCtrl_PWR.LVDS[2:0]设置）（默认）
	1: VDDH/VDDA 电压值低于监测电压（监测电压值由 ChipCtrl_PWR.LVDS[2:0]设置）
	当 LVD 使能且 LVDFLG 为 1 时，可以产生中断。
Bit 23	CLKMUX_LOCK: 系统时钟源切换状态
	当寄存器 CLK_SRC_SEL[1:0]的配置发生改变时，该位可以被观察、可以由硬件设置或者清除
	0: 系统时钟源未切换或者系统时钟源切换失败
	1: 系统时钟源切换成功（默认）
Bit 22	SYS_CLK_LOCK: 系统时钟频率切换状态
	当寄存器 SYS_CLK_SEL[7:0]的配置发生改变时，该位可以被观察、可以由硬件设置或者清除
	0: 系统时钟频率未切换或者系统时钟频率切换失败
	1: 系统时钟频率切换成功（默认）
Bit 21	保留，必须保持复位值
Bit 20	PLL_LOCK: PLL 稳定标志，该位由硬件设置和清除
	0: PLL 未稳定（锁存）或者关闭（默认）
	1: PLL 稳定（锁存）
Bit 19	OSCH_STB: OSCH 稳定标志，该位由硬件设置和清除
	0: OSCH 未稳定或者关闭（默认）
	1: OSCH 稳定
Bit 18	OSCL_STB: OSCL 稳定标志，该位由硬件设置和清除
	0: OSCL 未稳定或者关闭（默认）
	1: OSCL 稳定
Bit 17	RCH_STB: RCH 稳定标志，该位由硬件设置和清除
	0: RCH 未稳定或者关闭
	1: RCH 稳定（默认）
Bit 16	RCL_STB: RCL 稳定标志，该位由硬件设置和清除
	0: RCL 未稳定或者关闭
	1: RCL 稳定（默认）
Bits 15:8	保留，必须保持复位值
Bit 7	CSS_STS: 时钟安全状态
	在 CSS_EN 使能后，当发生外部晶振失效，时钟恢复为内部 RCH 事件时，该位置 1，该位写 1 清除。
	0: 未发生时钟恢复事件（默认）
	1: 发生时钟恢复事件
Bit 6	CLKEN_ERR: 时钟使能错误状态，该位写 1 清除
	0: 未发生时钟使能错误（默认）
	1: 发生时钟使能错误
Bit 5	CLKMUX_ERR: 系统时钟源切换错误状态，该位写 1 清除
	0: 未发生系统时钟源切换错误（默认）
	1: 发生系统时钟源切换错误
Bit 4	PLL_STS: PLL 锁存信号的变化状态，该位写 1 清除
	0: PLL 锁存信号未发生过变化（默认）
	1: PLL 锁存信号发生过变化

Bit 3	OSCH_STS: OSCH 稳定信号的变化状态, 该位写 1 清除
	0: OSCH 稳定信号未发生过变化 (默认)
	1: OSCH 稳定信号发生过变化
Bit 2	OSCL_STS: OSCL 稳定信号的变化状态, 该位写 1 清除
	0: OSCL 稳定信号未发生过变化 (默认)
	1: OSCL 稳定信号发生过变化
Bit 1	RCH_STS: RCH 稳定信号的变化状态, 该位写 1 清除
	0: RCH 稳定信号未发生过变化
	1: RCH 稳定信号发生过变化 (默认)
Bit 0	RCL_STS: RCL 稳定信号的变化状态, 该位写 1 清除
	0: RCL 稳定信号未发生过变化
	1: RCL 稳定信号发生过变化 (默认)

5.3.1.8 复位状态寄存器 (ChipCtrl_RSTCSR)

地址偏移: 0x1C

复位值: 0x0000_000X

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			LOCKUP RESET	Res.				SYSRESET _FLAG	LOCKUP_ RST_FLAG	WWDG_ RST_FLAG	IWDG_ RST_FLAG	RSTN _FLAG	STANDBY_RST _FLAG	LVR_ FLAG	POR_ FLAG
			rw					rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:13	保留, 必须保持复位值
Bit 12	LOCKUPRESET: 当发生处理器 LOCKUP 事件时, 产生系统复位使能
	0: 当发生处理器 LOCKUP 事件时, 不产生系统复位 (默认)
	1: 当发生处理器 LOCKUP 事件时, 产生系统复位
Bits 11:8	保留, 必须保持复位值
Bit 7	SYSREST_FLAG: 处理器复位标志, 由硬件设置, 该位写 1 清除
	0: 未发生处理器复位请求 (默认)
	1: 发生过处理器复位请求, 产生系统复位
Bit 6	LOCKUP_RST_FLAG: 在 LOCKUPRESET=1 条件下的处理器 LOCKUP 系统复位标志, 由硬件设置, 该位写 1 清除
	0: 未发生处理器 LOCKUP 系统复位 (默认)
	1: 发生过处理器 LOCKUP 系统复位
Bit 5	保留, 必须保持复位值
Bit 4	IWDG_RST_FLAG: 独立看门狗复位, 由硬件设置, 该位写 1 清除
	0: 未发生独立看门狗 IWDG 复位 (默认)
	1: 发生独立看门狗 IWDG 复位
Bit 3	RSTN_FLAG: 外部复位标志, 由硬件设置, 该位写 1 清除
	0: 未发生外部复位 (默认)

	1: 发生外部复位
Bit 2	保留, 必须保持复位值
Bit 1	LVR_FLAG : 低压监测复位, 由硬件设置, 该位写 1 清除
	0: 未发生低压监测复位 (默认)
	1: 发生低压监测复位
Bit 0	POR_FLAG : 上电/掉电复位标志, 由硬件设置, 该位写 1 清除
	0: 未发生上电/掉电复位 (默认)
	1: 发生上电/掉电复位

5.3.1.9 芯片密钥寄存器 (ChipCtrl_KEY)

地址偏移: 0x20

复位值: 0x0000_0002

所有芯片相关寄存器受到密钥寄存器保护, 需要将 16 进制数 0x8721_4365 写入密钥寄存器才能解锁; 当其他数据写入寄存器, 或者 32 个系统时钟周期后, 寄存器恢复到写保护状态。

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				LOCK_CNT[4:0]				Res.				LOCK	WR_ERR		
				r								r	r		

写入:

Bits 31:0	KEY[31:0] : 写入密钥以解锁或者锁定寄存器
-----------	-----------------------------------

读出:

Bits 31:13	保留, 必须保持复位值
Bits 12:8	LOCK_CNT[4:0] : 锁定窗口计数值
Bits 7:2	保留, 必须保持复位值
Bit 1	LOCK : 锁定状态
	0: 写保护已被解除
	1: 写保护锁定 (默认)
Bit 0	WR_ERR : 寄存器写错误标志
	0: 没有写保护错误 (解除写保护之后成功写入寄存器) (默认)
	1: 写保护状态下的寄存器写入操作会被忽略

5.3.1.10 RCL 配置寄存器 (ChipCtrl_RCLCFG)

地址偏移: 0x24

复位值: 0x0000_00XX

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

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

Bits	31:8	保留，必须保持复位值
Bit	7	RCL_EN : 内部低振 RCL 使能
		0: RCL 关闭
		1: RCL 开启（默认）
Bit	6	保留，必须保持复位值
Bits	5:0	RCL_TRIM[5:0] : RCL 频率 TRIM 值
		000000: RCL 频率最低
		...
		111111: RCL 频率最高
注: RCL_TRIM 初始值由存储器配置 (Option Byte) 直接加载，用户可以根据应用需要配置。		

5.3.1.11 BD 控制寄存器 (ChipCtrl_BDCR)

地址偏移: 0x28

复位值: 0x0000_7004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															WT _EN
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	RST _CFG	Res.		BDRST	WT_SEL[1:0]		Res.		OSCL _REN	OSCL_TS[1:0]		OSCL_GAIN [1:0]		OSCL _BYP	OSCL _EN
	rw			rw	rw				rw	rw	rw	rw	rw	rw	rw

Bits	31:17	保留，必须保持复位值
Bit	16	WT_EN : WT 时钟使能
		0: WT 时钟关闭（默认）
		1: WT 时钟使能
		使能 WT 前要先选择 WT 时钟源，工作过程中不能切换时钟
Bit	15	保留，必须保持复位值
Bit	14	RST_CFG : 复位引脚 PF2 端口配置
		0: PF2 作为通用 GPIO
		1: PF2 作为外部复位引脚（默认）
Bits	13:12	保留，必须保持复位值
Bit	11	BDRST : WT 软件复位，该位由软件配置和清除

	0: 软件复位无效（默认）
	1: 软件复位整个 WT 域和外部低频振荡 OSCL 的配置
Bits 10:9	WT_SEL[1:0]: WT 时钟源选择
	00: 无时钟（默认）
	01: 外部低振 OSCL
	10: 内部低振 RCL
	11: 外部高振 OSCH 的分频后时钟
Bits 8:7	保留，必须保持复位值
Bit 6	OSCL_REN: 外部低振 OSCL 反馈电阻选择
	0: 选择内部反馈电阻（默认）
	1: 选择外部反馈电阻
Bits 5:4	OSCL_TS[1:0]: 外部低振稳定时间选择
	00: 等待 1000ms（默认）
	01: 等待 500ms
	10: 等待 250ms
	11: 等待 125ms
Bits 3:2	OSCL_GAIN[1:0]: OSCL 增益配置，默认配置为 01
	00: 低增益
	...
	11: 高增益
Bit 1	OSCL_BYP: 外部低频晶振旁路
	当 OSCL 旁路以后，可以直接由外部输入时钟，只能在 OSCH 和 OSCL 晶振关闭的情况下才能使用
	0: OSCL 晶振不旁路（默认）
	1: OSCL 晶振旁路
Bit 0	OSCL_EN: 外部低频晶振 OSCL 使能
	0: OSCL 关闭（默认）
	1: OSCL 使能

5.3.1.12 BGR 配置寄存器（ChipCtrl_BGR）

地址偏移：0x2C

复位值：0x0000_00XX

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

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

Bits 31:6	保留，必须保持复位值
Bits 5:0	BGR_TRIM[5:0]: VBG 电压 TRIM 值
	000000: 校准值最小

	...
	111111: 校准值最大
注: 初始值由存储器配置 (Option Byte) 直接加载, 用户可以根据应用需要配置。	

5.3.2 系统配置寄存器组 (SysCtrl)

此寄存器组的基地址为 0x4800_7000, 通过 AHB1 总线访问。所有 SysCtrl 寄存器受到密钥保护, 写操作时需要先解锁密钥寄存器 SysCtrl_KEY, 解锁方式为写 0x05FA_659A 到密钥寄存器。

表 5-2 系统配置相关寄存器表

名称	说明	读写权限	复位值	字节地址
SysCtrl_ClkEnR0	模块时钟配置寄存器 0	R/W	0x0000_0000	0x4800_7000
SysCtrl_ClkEnR1	模块时钟配置寄存器 1	R/W	0x0000_0000	0x4800_7004
SysCtrl_ClkEnR2	模块时钟配置寄存器 2	R/W	0x0000_0000	0x4800_7008
SysCtrl_RST0	软件复位控制寄存器 0	R/W	0xFCBF_7763	0x4800_700C
SysCtrl_RST1	软件复位控制寄存器 1	R/W	0x0000_0060	0x4800_7010

注: x 表示不确定; -表示未实现; u 表示不受复位影响; c 表示该位复位值因复位源而异; R 只读; W 只写; R-0 表示只读且读 0; R/W 表示可读写 (以后章节同上述)。

5.3.2.1 模块时钟配置寄存器 0 (SysCtrl_ClkEnR0)

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
TIM3_	TIM17_	TIM16_	TIM15_	TIM14_	TIM6_	Res.		SSP2_	Res.	GPIOF_	WWDG_	TIM1_	RAM1_	RAM0_	EFLS_
CLKEN	CLKEN	CLKEN	CLKEN	CLKEN	CLKEN			CLKEN		CLKEN	_CLKEN	CLKEN	CLKEN	CLKEN	CLKEN
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
Res.	UART1_	UART0_	SSP0_	Res.	I2C0_	ADC_	DMA_	Res.	CRC_	DIV_	Res.			GPIOB_	GPIOA_
	CLKEN	CLKEN	CLKEN		CLKEN	CLKEN	CLKEN		CLKEN	CLKEN				CLKEN	CLKEN
	rw	rw	rw		rw	rw	rw		rw	rw				rw	rw

Bit	31	TIM3_CLKEN: TIM3 时钟使能位 (由软件进行清零和置位)
		0: TIM3 时钟禁止 (默认)
		1: TIM3 时钟使能
Bit	30	TIM17_CLKEN: TIM17 时钟使能位 (由软件进行清零和置位)
		0: TIM17 时钟禁止 (默认)
		1: TIM17 时钟使能
Bit	29	TIM16_CLKEN: TIM16 时钟使能位 (由软件进行清零和置位)
		0: TIM16 时钟禁止 (默认)
		1: TIM16 时钟使能
Bit	28	TIM15_CLKEN: TIM15 时钟使能位 (由软件进行清零和置位)
		0: TIM15 时钟禁止 (默认)
		1: TIM15 时钟使能

Bit 27	TIM14_CLKEN: TIM14 时钟使能位（由软件进行清零和置位）
	0: TIM14 时钟禁止（默认）
	1: TIM14 时钟使能
Bit 26	TIM6_CLKEN: TIM6 时钟使能位（由软件进行清零和置位）
	0: TIM6 时钟禁止（默认）
	1: TIM6 时钟使能
Bit 25:24	保留，必须保持复位值
Bit 23	SSP2_CLKEN: SSP2 时钟使能位（由软件进行清零和置位）
	0: SSP2 时钟禁止（默认）
	1: SSP2 时钟使能
Bit 22	保留，必须保持复位值
Bit 21	GPIOF_CLKEN: GPIOF 时钟使能位（由软件进行清零和置位）
	0: GPIOF 时钟禁止（默认）
	1: GPIOF 时钟使能
Bit 20	WWDG_CLKEN: WWDG 时钟使能位（由软件进行清零和置位）
	0: WWDG 时钟禁止（默认）
	1: WWDG 时钟使能
Bit 19	TIM1_CLKEN: TIM1 时钟使能位（由软件进行清零和置位）
	0: TIM1 时钟禁止（默认）
	1: TIM1 时钟使能
Bit 18	RAM1_CLKEN: SRAM1 接口时钟使能位
	决定休眠模式下是否打开或者禁止 SRAM1 的时钟，由软件进行清零和置位。
	0: SRAM1 休眠时接口时钟禁止（默认）
	1: SRAM1 休眠时接口时钟使能
Bit 17	RAM0_CLKEN: SRAM0 接口时钟使能位
	决定休眠模式下是否打开或者禁止 SRAM0 的时钟，由软件进行清零和置位。
	0: SRAM0 休眠时接口时钟禁止（默认）
	1: SRAM0 休眠时接口时钟使能
Bit 16	EFLS_CLKEN: Flash 接口时钟使能位
	决定休眠模式下是否打开或者禁止 Flash 的时钟，由软件进行清零和置位。
	0: Flash 睡眠时接口时钟禁止（默认）
	1: Flash 休眠时接口时钟使能
Bit 15	保留，必须保持复位值
Bit 14	UART1_CLKEN: UART1 时钟使能位（由软件进行清零和置位）
	0: UART1 时钟禁止（默认）
	1: UART1 时钟使能
Bit 13	UART0_CLKEN: UART0 时钟使能位（由软件进行清零和置位）
	0: UART0 时钟禁止（默认）
	1: UART0 时钟使能
Bit 12	SSPO_CLKEN: SSPO 时钟使能位（由软件进行清零和置位）
	0: SSPO 时钟禁止（默认）
	1: SSPO 时钟使能
Bit 11	保留，必须保持复位值



Bit 10	I2CO_CLKEN: I2CO 时钟使能位（由软件进行清零和置位）
	0: I2CO 时钟禁止（默认）
	1: I2CO 时钟使能
Bit 9	ADC_CLKEN: ADC 时钟使能位（由软件进行清零和置位）
	0: ADC 时钟禁止（默认）
	1: ADC 时钟使能
Bit 8	DMA_CLKEN: DMA 时钟使能位（由软件进行清零和置位）
	0: DMA 时钟禁止（默认）
	1: DMA 时钟使能
Bit 7	保留，必须保持复位值
Bit 6	CRC_CLKEN: CRC 时钟使能位（由软件进行清零和置位）
	0: CRC 时钟禁止（默认）
	1: CRC 时钟使能
Bit 5	DIV_CLKEN: DIV 时钟使能位（由软件进行清零和置位）
	0: DIV 时钟禁止（默认）
	1: DIV 时钟使能
Bits 4:2	保留，必须保持复位值
Bit 1	GPIOB_CLKEN: GPIOB 时钟使能位（由软件进行清零和置位）
	0: GPIOB 时钟禁止（默认）
	1: GPIOB 时钟使能
Bit 0	GPIOA_CLKEN: GPIOA 时钟使能位（由软件进行清零和置位）
	0: GPIOA 时钟禁止（默认）
	1: GPIOA 时钟使能

5.3.2.2 模块时钟配置寄存器 1 (SysCtrl_ClkEnR1)

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
Res.						SSP2_CLKSEL [1:0]	Res.		SSPO_CLKSEL [1:0]	I2CO_CLKSEL [1:0]	Res.				
						rw			rw	rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UART1_CLKSEL [1:0]		UART0_CLKSEL [1:0]		Res.			OSCH_ CLKDIV _EN	Res.		MCO_SEL[2:0]			MCO_DIV[2:0]		
rw		rw					rw			rw			rw		

Bits 31:26	保留，必须保持复位值
Bits 25:24	SSP2_CLKSEL[1:0]: SSP2 时钟选择
	00: PCLK0（默认）
	01: SYS_CLK_IN
	10: RCH_CLK

	11: OSCL_CLK
Bits 23:22	保留，必须保持复位值
Bits 21:20	SSPO_CLKSEL[1:0]: SSPO 时钟选择
	00: PCLK1 (默认)
	01: SYS_CLK_IN
	10: RCH_CLK
	11: OSCL_CLK
Bits 19:18	I2CO_CLKSEL[1:0]: I2CO 时钟选择
	00: PCLK1 (默认)
	01: SYS_CLK_IN
	10: RCH_CLK
	11: OSCL_CLK
Bits 17:16	保留，必须保持复位值
Bits 15:14	UART1_CLKSEL[1:0]: UART1 时钟选择
	00: PCLK1 (默认)
	01: SYS_CLK_IN
	10: RCH_CLK
	11: OSCL_CLK
Bits 13:12	UART0_CLKSEL[1:0]: UART0 时钟选择
	00: PCLK1 (默认)
	01: SYS_CLK_IN
	10: RCH_CLK
	11: OSCL_CLK
Bits 11:9	保留，必须保持复位值
Bit 8	OSCH_CLKDIV_EN: OSCH 时钟 128 分频使能
	0: OSCH 时钟 128 分频禁止 (默认)
	1: OSCH 时钟 128 分频使能
Bits 7:6	保留，必须保持复位值
Bits 5:3	MCO_SEL[2:0]: MCU 时钟输出选择
	000: 不输出时钟 (默认)
	001: PLL_CLK
	010: SYS_CLK_IN
	011: SYS_CLK
	100: RCL_CLK
	101: OSCL_CLK
	110: RCH_CLK
	111: OSCH_CLK
	在起始阶段或者时钟源切换阶段，时钟输出可能存在短时间的抖动。
Bits 2:0	MCO_DIV[2:0]: MCU 时钟输出分频设置
	000: MCO (默认)
	001: MCO/2
	010: MCO/4
	011: MCO/8

	100: MCO/16
	101: MCO/32
	110: MCO/64
	111: MCO/128

5.3.2.3 模块时钟配置寄存器 2 (SysCtrl_ClkEnR2)

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
Res.															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.									EXTI_	ANA_	Res.				
									CLKEN	CLKEN					
									rw	rw					

Bits 31:7	保留, 必须保持复位值
Bit 6	EXTI_CLKEN: EXTI 时钟使能 (由软件进行清零和置位)
	0: EXTI 时钟禁止 (默认)
	1: EXTI 时钟使能
Bit 5	ANA_CLKEN: 模拟模块控制器 (ACMP/DAC/OPA/HALL_MID) 时钟使能 (由软件进行清零和置位)
	0: 模拟模块控制器时钟禁止 (默认)
	1: 模拟模块控制器时钟使能
Bits 4:0	保留, 必须保持复位值

5.3.2.4 软件复位控制寄存器 0 (SysCtrl_RST0)

地址偏移: 0x0C

复位值: 0xFCBF_7763

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIM3	TIM17	TIM16	TIM15	TIM14	TIM6	Res.		SSP2_	Res.	GPIOF	WWDG	TIM1	RAM1	RAM0	EFLS_
_RST	_RST	_RST	_RST	_RST	_RST			RST		_RST	_RST	_RST	_RST	_RST	RST
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
Res.	UART1_	UART0	SSP0_	Res.	I2CO_	ADC_	DMA_	Res.	CRC_	DIV_	Res.			GPIOB	GPIO
	RST	_RST	RST		RST	RST	RST		RST	RST				_RST	A
	rw	rw	rw		rw	rw	rw		rw	rw				rw	rw

Bit 31	TIM3_RST: TIM3 复位 (由软件进行清零和置位)
	0: 复位 TIM3

	1: 无影响 (默认)
Bit 30	TIM17_RST: TIM17 复位 (由软件置位或者清零)
	0: 复位 TIM17
	1: 无影响 (默认)
Bit 29	TIM16_RST: TIM16 复位 (由软件置位或者清零)
	0: 复位 TIM16
	1: 无影响 (默认)
Bit 28	TIM15_RST: TIM15 复位 (由软件置位或者清零)
	0: 复位 TIM15
	1: 无影响 (默认)
Bit 27	TIM14_RST: TIM14 复位 (由软件置位或者清零)
	0: 复位 TIM14
	1: 无影响 (默认)
Bit 26	TIM6_RST: TIM6 复位 (由软件置位或者清零)
	0: 复位 TIM6
	1: 无影响 (默认)
Bits 25:24	保留, 必须保持复位值
Bit 23	SSP2_RST: SSP2 复位 (由软件置位或者清零)
	0: 复位 SSP2
	1: 无影响 (默认)
Bit 22	保留, 必须保持复位值
Bit 21	GPIOF_RST: GPIOF 复位 (由软件置位或者清零)
	0: 复位 GPIOF
	1: 无影响 (默认)
Bit 20	WWDG_RST: WWDG 复位 (由软件置位或者清零)
	0: 复位 WWDG
	1: 无影响 (默认)
Bit 19	TIM1_RST: TIM1 复位 (由软件置位或者清零)
	0: 复位 TIM1
	1: 无影响 (默认)
Bit 18	RAM1_RST: SRAM1 端口逻辑复位 (由软件置位或者清零)
	0: 复位 SRAM1 端口
	1: 无影响 (默认)
Bit 17	RAM0_RST: SRAM0 端口逻辑复位 (由软件置位或者清零)
	0: 复位 SRAM0 端口
	1: 无影响 (默认)
Bit 16	EFLS_RST: Flash 端口复位 (由软件置位或者清零)
	0: 复位 Flash 端口
	1: 无影响 (默认)
Bit 15	保留, 必须保持复位值
Bit 14	UART1_RST: UART1 复位 (由软件置位或者清零)
	0: 复位 UART1
	1: 无影响 (默认)



Bit 13	UART0_RST: UART0 复位（由软件置位或者清零）
	0: 复位 UART0
	1: 无影响（默认）
Bit 12	SSPO_RST: SSPO 复位（由软件置位或者清零）
	0: 复位 SSPO
	1: 无影响（默认）
Bit 11	保留，必须保持复位值
Bit 10	I2C0_RST: I2C0 复位（由软件置位或者清零）
	0: 复位 I2C0
	1: 无影响（默认）
Bit 9	ADC_RST: ADC 复位（由软件置位或者清零）
	0: 复位 ADC
	1: 无影响（默认）
Bit 8	DMA_RST: DMA 复位（由软件置位或者清零）
	0: 复位 DMA
	1: 无影响（默认）
Bit 7	保留，必须保持复位值
Bit 6	CRC_RST: CRC 复位（由软件置位或者清零）
	0: 复位 CRC
	1: 无影响（默认）
Bit 5	DIV_RST: DIV 复位（由软件置位或者清零）
	0: 复位 DIV
	1: 无影响（默认）
Bits 4:2	保留，必须保持复位值
Bit 1	GPIOB_RST: GPIOB 复位（由软件置位或者清零）
	0: 复位 GPIOB
	1: 无影响（默认）
Bit 0	GPIOA_RST: GPIOA 复位（由软件置位或者清零）
	0: 复位 GPIOA
	1: 无影响（默认）

5.3.2.5 软件复位控制寄存器 1 (SysCtrl_RST1)

地址偏移: 0x10

复位值: 0x0000_0060

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.										EXTI_	ANA_	Res.			
										RST	RST				
										rw	rw				

Bits 31:7	保留，必须保持复位值
Bit 6	EXTI_RST: EXTI 复位（由软件置位或者清零）
	0: 复位 EXTI
	1: 无影响（默认）
Bit 5	ANA_RST: 模拟模块控制器（ACMP/DAC/OPA/HALL_MID）复位（由软件置位或者清零）
	0: 复位模拟模块控制器
	1: 无影响（默认）
Bits 4:0	保留，必须保持复位值

6. 系统控制单元

6.1 Multi-AHB 总线矩阵

32 位的多 AHB 总线矩阵将所有主机模块（CPU、DMA）和从机模块（Flash、SRAM、AHB、APB 外设）互连，确保芯片多个高速外设可以同时高效地工作。

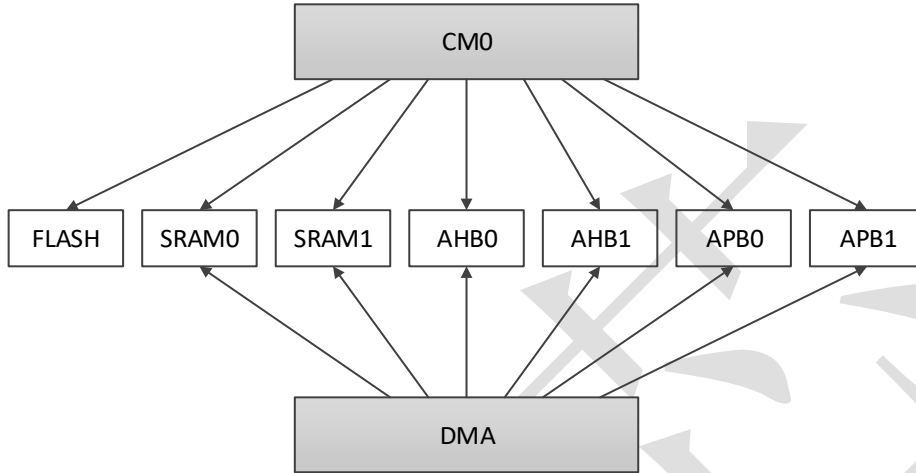


图 6-1 LCM32F037 总线矩阵

6.2 外设互联矩阵

LCM32F037 外设之间存在可配置的互联，允许外设之间自动、灵活地进行通信交互，可以节省 CPU 开销并降低功耗，而且保证了快速、可预测的响应延时。

表 6-1 LCM32F037 外设互联矩阵

互联发起源	互联目标	联动行为
TIMx	TIMx	定时器之间的同步和联动
	ADC	A/D 转换触发
	DACx	D/A 转换触发
	ACMPx	比较器输出消隐控制
	DMA	Memory 传输触发
ADC	TIMx	模拟看门狗触发定时器
GPIO OSCH/128 MCO	TIM14	各时钟源接入定时器的输入通道，用于时钟测量和校准
ACMPx	TIMx	定时器输出控制、输入捕获和输入的触发
	ADC	A/D 转换触发
CSS CPU (Hard Fault) LVD ACMPx GPIO	TIM1 TIM15 TIM16 TIM17	定时器刹车输入
GPIO	TIMx	外部触发，紧急关断

	ADC	A/D 转换外部触发
	DACx	D/A 转换触发
DACx	ACMPx	比较器输入
HALL_MID	ACMPx	比较器输入

6.3 互联配置寄存器描述

外设互联的相关寄存器集中处于 SysCtrl 寄存器组，此寄存器组的基地址为 0x4800_7000，通过 AHB1 总线访问。所有 SysCtrl 寄存器受到密钥保护，写操作时需要先解锁密钥寄存器 SysCtrl_KEY，解锁方式为写 0x05FA_659A 到密钥寄存器，且在随后的 32 个时钟周期内完成配置过程，因此需要注意不能被中断打断写入过程，并通过读回确定写入完成。

表 6-2 互联配置相关寄存器表

名称	说明	读写权限	复位值	字节地址
SysCtrl_EDU_CFG0	互联配置寄存器 0	R/W	0x0000_0000	0x4800_7020
SysCtrl_EDU_CFG1	互联配置寄存器 1	R/W	0x0000_0000	0x4800_7024
SysCtrl_EDU_CFG2	互联配置寄存器 2	R/W	0x0000_0000	0x4800_7028
SysCtrl_EDU_CFG3	互联配置寄存器 3	R/W	0x0000_0000	0x4800_702C
SysCtrl_EDU_CFG4	互联配置寄存器 4	R/W	0x0000_0000	0x4800_7030
SysCtrl_EDU_CFG5	互联配置寄存器 5	R/W	0x0000_0000	0x4800_7034
SysCtrl_EDU_CFG6	互联配置寄存器 6	R/W	0x0000_0000	0x4800_7038
SysCtrl_EDU_CFG7	互联配置寄存器 7	R/W	0x0000_0000	0x4800_703C
SysCtrl_KEY	系统密钥寄存器	R/W	0x0000_0002	0x4800_707C

注：x 表示不确定；-表示未实现；u 表示不受复位影响；c 表示该位复位值因复位源而异；R 只读；W 只写；R-0 表示只读且读 0；R/W 表示可读写（以后章节同上述）。

6.3.1 互联配置寄存器 0 (SysCtrl_EDU_CFG0)

地址偏移：0x20

复位值：0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMA_REQ7_MAP				DMA_REQ6_MAP				DMA_REQ5_MAP				DMA_REQ4_MAP			
rw				rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMA_REQ3_MAP				DMA_REQ2_MAP				DMA_REQ1_MAP				DMA_REQ0_MAP			
rw				rw				rw				rw			

Bits 31:16	DMA_REQx_MAP[3:0]: DMA 请求源通道配置 (x=4 到 7)
	0000: DMA 通道 x 的触发源配置为 TIM1_CH1 (默认)
	0001: DMA 通道 x 的触发源配置为 TIM1_CH2
	0010: DMA 通道 x 的触发源配置为 TIM1_CH3 / TIM1_UP
	0011: DMA 通道 x 的触发源配置为 TIM1_CH4 / TIM1_TRIG / TIM1 COM
	0100: DMA 通道 x 的触发源配置为 ADC
	0101: DMA 通道 x 的触发源配置为 TIM3_CH1 / TIM3_TRIG

	0110: DMA 通道 x 的触发源配置为 TIM3_CH2 / TIM3_CH3
	0111: DMA 通道 x 的触发源配置为 TIM3_CH4 / TIM3_UP
	1000: DMA 通道 x 的触发源配置为 TIM6
	1001: DMA 通道 x 的触发源配置为 TIM15
	1010: DMA 通道 x 的触发源配置为 TIM16
	1011: DMA 通道 x 的触发源配置为 TIM17
	1100: 保留
	1101: 保留
	1110: DMA 通道 x 的触发源配置为 DAC1
	1111: DMA 通道 x 的触发源配置为 DAC0
Bits 15:0	DMA_REQx_MAP[3:0]: DMA 请求源通道配置 (x=0 到 3)
	0000: DMA 通道 x 的触发源配置为 UART0_TX (默认)
	0001: DMA 通道 x 的触发源配置为 UART0_RX
	0010: DMA 通道 x 的触发源配置为 UART1_TX
	0011: DMA 通道 x 的触发源配置为 UART1_RX
	0100: DMA 通道 x 的触发源配置为 I2C0_TX
	0101: DMA 通道 x 的触发源配置为 I2C0_RX
	0110: DMA 通道 x 的触发源配置为 SSP0_TX
	0111: DMA 通道 x 的触发源配置为 SSP0_RX
	1000: 保留
	1101: 保留
	1010: 保留
	1011: 保留
	1100: 保留
	1101: 保留
	1110: DMA 通道 x 的触发源配置为 SSP2_TX
	1111: DMA 通道 x 的触发源配置为 SSP2_RX

6.3.2 互联配置寄存器 1 (SysCtrl_EDU_CFG1)

地址偏移: 0x24

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.													TIM1_ETRS[2:0]		
													rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Res.	TIM1_TI4S[2:0]			Res.	TIM1_TI3S[2:0]			Res.	TIM1_TI2S[2:0]			Res.	TIM1_TI1S[2:0]				
			rw						rw						rw		

Bits 31:19	保留, 必须保持复位值
Bits 18:16	TIM1_ETRS[2:0]: TIM1 ETR 输入触发源选择
	000: 外部 GPIO 输入, 参考相关 I/O 复用说明 (默认)

	001: ADC 模拟模块看门狗输出
	010: ACMP0 输出
	011: ACMP1 输出
	100: ACMP2 输出
	101: TIM15_OC1_TRIG
	110: TIM3_OC1_TRIG
	111: TIM14_OC1_TRIG
Bit 15	保留, 必须保持复位值
Bits 14:12	TIM1_TI4S[2:0]: TIM1 TI4 输入触发源选择
	000: 外部 GPIO 输入, 参考相关 I/O 复用说明 (默认)
	001: ADC 模拟模块看门狗输出
	010: ACMP0 输出
	011: ACMP1 输出
	100: ACMP2 输出
	101: TIM15_OC1_TRIG
	110: TIM3_OC1_TRIG
	111: TIM14_OC1_TRIG
Bit 11	保留, 必须保持复位值
Bits 10:8	TIM1_TI3S[2:0]: TIM1 TI3 输入触发源选择
	000: 外部 GPIO 输入, 参考相关 I/O 复用说明 (默认)
	001: ADC 模拟模块看门狗输出
	010: ACMP0 输出
	011: ACMP1 输出
	100: ACMP2 输出
	101: TIM15_OC1_TRIG
	110: TIM3_OC1_TRIG
	111: TIM14_OC1_TRIG
Bit 7	保留, 必须保持复位值
Bits 6:4	TIM1_TI2S[2:0]: TIM1 TI2 输入触发源选择
	000: 外部 GPIO 输入, 参考相关 I/O 复用说明 (默认)
	001: ADC 模拟模块看门狗输出
	010: ACMP0 输出
	011: ACMP1 输出
	100: ACMP2 输出
	101: TIM15_OC1_TRIG
	110: TIM3_OC1_TRIG
	111: TIM14_OC1_TRIG
Bit 3	保留, 必须保持复位值
Bits 2:0	TIM1_TI1S[2:0]: TIM1 TI1 输入触发源选择
	000: 外部 GPIO 输入, 参考相关 I/O 复用说明 (默认)
	001: ADC 模拟模块看门狗输出
	010: ACMP0 输出
	011: ACMP1 输出

	100: ACMP2 输出
	101: TIM15_OC1_TRIG
	110: TIM3_OC1_TRIG
	111: TIM14_OC1_TRIG
注: TIMx_OC1_TRIG 和 TIMx_TRGO 不同, 详见 TIMx 寄存器说明。	

6.3.3 互联配置寄存器 2 (SysCtrl_EDU_CFG2)

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
Res.													TIM3_ETRS[2:0]		
													rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Res.	TIM3_TI4S[2:0]			Res.	TIM3_TI3S[2:0]			Res.	TIM3_TI2S[2:0]			Res.	TIM3_TI1S[2:0]				
			rw						rw						rw		

Bits 31:19	保留, 必须保持复位值
Bits 18:16	TIM3_ETRS[2:0]: TIM3 ETR 输入触发源选择
	000: 外部 GPIO 输入, 参考相关 I/O 复用说明 (默认)
	001: ADC 模拟模块看门狗输出
	010: ACMP0 输出
	011: ACMP1 输出
	100: ACMP2 输出
	101: TIM1_OC1_TRIG
	110: TIM15_OC1_TRIG
	111: TIM16_OC1_TRIG
Bit 15	保留, 必须保持复位值
Bits 14:12	TIM3_TI4S[2:0]: TIM3 TI4 输入触发源选择
	000: 外部 GPIO 输入, 参考相关 I/O 复用说明 (默认)
	001: ADC 模拟模块看门狗输出
	010: ACMP0 输出
	011: ACMP1 输出
	100: ACMP2 输出
	101: TIM1_OC1_TRIG
	110: TIM15_OC1_TRIG
	111: TIM16_OC1_TRIG
Bit 11	保留, 必须保持复位值
Bits 10:8	TIM3_TI3S[2:0]: TIM3 TI3 输入触发源选择
	000: 外部 GPIO 输入, 参考相关 I/O 复用说明 (默认)
	001: ADC 模拟模块看门狗输出
	010: ACMP0 输出

	011: ACMP1 输出
	100: ACMP2 输出
	101: TIM1_OC1_TRIG
	110: TIM15_OC1_TRIG
	111: TIM16_OC1_TRIG
Bit 7	保留, 必须保持复位值
Bits 6:4	TIM3_TI2S[2:0]: TIM3 TI2 输入触发源选择
	000: 外部 GPIO 输入, 参考相关 I/O 复用说明 (默认)
	001: ADC 模拟模块看门狗输出
	010: ACMP0 输出
	011: ACMP1 输出
	100: ACMP2 输出
	101: TIM1_OC1_TRIG
	110: TIM15_OC1_TRIG
	111: TIM16_OC1_TRIG
Bit 3	保留, 必须保持复位值
Bits 2:0	TIM3_TI1S[2:0]: TIM3 TI1 输入触发源选择
	000: 外部 GPIO 输入, 参考相关 I/O 复用说明 (默认)
	001: ADC 模拟模块看门狗输出
	010: ACMP0 输出
	011: ACMP1 输出
	100: ACMP2 输出
	101: TIM1_OC1_TRIG
	110: TIM15_OC1_TRIG
	111: TIM16_OC1_TRIG
注: TIMx_OCI_TRIG 和 TIMx_TRGO 不同, 详见 TIMx 寄存器说明。	

6.3.4 互联配置寄存器 3 (SysCtrl_EDU_CFG3)

偏移地址: 0x2C

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	TIM17_ETRS[2:0]			Res.	TIM17_TI1S[2:0]			Res.	TIM16_ETRS[2:0]			Res.	TIM16_TI1S[2:0]		
	rw				rw				rw				rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				TIM15_ETRS[2:0]			Res.	TIM15_TI2S[2:0]			Res.	TIM15_TI1S[2:0]			
				rw							rw				

Bit 31	保留, 必须保持复位值
Bits 30:28	TIM17_ETRS[2:0]: TIM17 ETR 输入触发源选择
	000: 保留 (默认)
	001: ADC 模拟模块看门狗输出

	010: ACMP0 输出
	011: ACMP1 输出
	100: ACMP2 输出
	101: TIM1_TRGO
	110: TIM3_TRGO
	111: TIM15_TRGO
Bit 27	保留, 必须保持复位值
Bits 26:24	TIM17_TI1S[2:0]: TIM17 TI1 输入触发源选择
	000: 外部 GPIO 输入, 参考相关 I/O 复用说明 (默认)
	001: ADC 模拟模块看门狗输出
	010: ACMP0 输出
	011: ACMP1 输出
	100: ACMP2 输出
	101: TIM1_TRGO
	110: TIM3_TRGO
	111: TIM15_TRGO
Bit 23	保留, 必须保持复位值
Bits 22:20	TIM16_ETRS[2:0]: TIM16 ETR 输入触发源选择
	000: 保留 (默认)
	001: ADC 模拟模块看门狗输出
	010: ACMP0 输出
	011: ACMP1 输出
	100: ACMP2 输出
	101: TIM1_TRGO
	110: TIM3_TRGO
	111: TIM15_TRGO
Bit 19	保留, 必须保持复位值
Bits 18:16	TIM16_TI1S[2:0]: TIM16 TI1 输入触发源选择
	000: 外部 GPIO 输入, 参考相关 I/O 复用说明 (默认)
	001: ADC 模拟模块看门狗输出
	010: ACMP0 输出
	011: ACMP1 输出
	100: ACMP2 输出
	101: TIM1_TRGO
	110: TIM3_TRGO
	111: TIM15_TRGO
Bits 15:11	保留, 必须保持复位值
Bits 10:8	TIM15_ETRS[2:0]: TIM15 ETR 输入触发源选择
	000: 保留 (默认)
	001: ADC 模拟模块看门狗输出
	010: ACMP0 输出
	011: ACMP1 输出
	100: ACMP2 输出

	101: TIM1_OC1_TRIG
	110: TIM3_OC1_TRIG
	111: TIM14_OC1_TRIG
Bit 7	保留, 必须保持复位值
Bits 6:4	TIM15_TI2S[2:0]: TIM15 TI2 输入触发源选择
	000: 外部 GPIO 输入, 参考相关 I/O 复用说明 (默认)
	001: ADC 模拟模块看门狗输出
	010: ACMP0 输出
	011: ACMP1 输出
	100: ACMP2 输出
	101: TIM1_OC1_TRIG
	110: TIM3_OC1_TRIG
	111: TIM14_OC1_TRIG
Bit 3	保留, 必须保持复位值
Bits 2:0	TIM15_TI1S[2:0]: TIM15 TI1 输入触发源选择
	000: 外部 GPIO 输入, 参考相关 I/O 复用说明 (默认)
	001: ADC 模拟模块看门狗输出
	010: ACMP0 输出
	011: ACMP1 输出
	100: ACMP2 输出
	101: TIM1_OC1_TRIG
	110: TIM3_OC1_TRIG
	111: TIM14_OC1_TRIG
<i>注: TIMx_OC_i_TRIG 和 TIMx_TRGO 不同, 详见 TIMx 寄存器说明。</i>	

6.3.5 互联配置寄存器 4 (SysCtrl_EDU_CFG4)

地址偏移: 0x30

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.		TIM1_OCCSE[5:0]						Res.		TIM1_BK2SE[5:0]					
		rw								rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		TIM1_BKSE[6:0]						Res.		TIM1_SP[6:0]					
		rw								rw					

Bits 31:30	保留, 必须保持复位值
Bits 29:24	TIM1_OCCSE[5:0]: TIM1 OCREF_CLR 输入触发源使能控制, 默认值 000000
	TIM1 OCREF_CLR 输入源为 ADC 模拟模块看门狗输出 / TIM15_TRGO / TIM3_TRGO / ACMP2 / ACMP1 / ACMP0 输出 (从高位到低位)
	所有输入源均可被配置为使能或不使能, 通过控制位 TIM1_OCCSE[x]配置
	所有输入源的有效触发电平, 通过控制位 TIM1_SP[x]配置

	OCREF_CLR 使能的输入源的有效电平触发通过逻辑或的方式组合
	TIM1_OCCSE[x] = 0: 相应触发源不使能
	TIM1_OCCSE[x] = 1: 相应触发源使能
Bits 23:22	保留, 必须保持复位值
Bits 21:16	TIM1_BK2SE[5:0]: TIM1 刹车 2 输入触发源使能控制, 默认值 000000
	TIM1 刹车 2 输入触发源为 ADC 模拟模块看门狗输出/ TIM16_OC1_TRIG / TIM17_OC1_TRIG / ACMP2 / ACMP1 / ACMP0 输出 (从高位到低位)
	所有输入源均可被配置为使能或不使能, 通过控制位 TIM1_BK2SE[x]配置
	所有输入源的有效触发电平, 通过控制位 TIM1_SP[x]配置
	刹车 2 使能的输入源的有效电平触发通过逻辑或的方式组合
	TIM1_BK2SE[x] = 0: 相应触发源不使能
	TIM1_BK2SE[x] = 1: 相应触发源使能
Bit 15	保留, 必须保持复位值
Bits 14:8	TIM1_BKSE[6:0]: TIM1 刹车输入触发源使能控制, 默认值 0000000
	TIM1 刹车输入触发源为外部 GPIO 输入/ ADC 模拟模块看门狗输出/ TIM15_TRGO / TIM3_TRGO / ACMP2 / ACMP1 / ACMP0 输出 (从高位到低位)
	所有输入源均可被配置为使能或不使能, 通过控制位 TIM1_BKSE[x]配置
	所有输入源的有效触发电平, 通过控制位 TIM1_SP[x]配置
	刹车使能的输入源的有效电平触发通过逻辑或的方式组合
	TIM1_BKSE[x] = 0: 相应触发源不使能
	TIM1_BKSE[x] = 1: 相应触发源使能
Bit 7	保留, 必须保持复位值
Bits 6:0	TIM1_SP[6:0]: TIM1 刹车/刹车 2/OCREF_CLR 输入触发源的有效电平选择, 默认值 0000000
	所有输入源的有效触发电平, 通过控制位 TIM1_SP[x]配置 (低位对齐)
	TIM1_SP[x] = 0: 相应触发源的有效电平是高
	TIM1_SP[x] = 1: 相应触发源的有效电平是低
<i>注: TIMx_OC1_TRIG 和 TIMx_TRGO 不同, 详见 TIMx 寄存器说明。</i>	

6.3.6 互联配置寄存器 5 (SysCtrl_EDU_CFG5)

地址偏移: 0x34

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	TIM17_BKSE[6:0]								TIM17_SP[6:0]						
	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TIM3_OCCSE[5:0]							Res.	TIM3_SP[5:0]						
	rw								rw						

Bit 31	保留, 必须保持复位值
Bits 30:24	TIM17_BKSE[6:0]: TIM17 刹车输入触发源使能控制, 默认值 0000000

	TIM17 刹车输入触发源为外部 GPIO 输入/ADC 模拟模块看门狗输出/TIM1_TRGO / TIM3_TRGO / ACMP2/ ACMP1 / ACMP0 输出（从高位到低位）
	所有输入源均可被配置为使能或不使能，通过控制位 TIM17_BKSE[x]配置
	所有输入源的有效触发电平，通过控制位 TIM17_SP[x]配置
	刹车使能的输入源的有效电平触发通过逻辑或的方式组合
	TIM17_BKSE[x] = 0: 相应触发源不使能
	TIM17_BKSE[x] = 1: 相应触发源使能
Bit 23	保留，必须保持复位值
Bits 22:16	TIM17_SP[6:0]: TIM17 刹车输入源的有效电平选择，默认值 0000000
	所有输入源的有效触发电平，通过控制位 TIM17_SP[x]配置（低位对齐）
	TIM17_SP[x] = 0: 相应触发源的有效电平是高
	TIM17_SP[x] = 1: 相应触发源的有效电平是低
Bits 15:14	保留，必须保持复位值
Bits 13:8	TIM3_OCCSE[5:0]: TIM3 OCREF_CLR 输入触发源使能控制，默认值 000000
	TIM3 OCREF_CLR 输入触发源为 ADC 模拟模块看门狗输出/ TIM15_TRGO / TIM1_TRGO / ACMP2 / ACMP1 / ACMP0 输出（从高位到低位）
	所有输入源均可被配置为使能或不使能，通过控制位 TIM3_OCCSE[x]配置
	所有输入源的有效触发电平，通过控制位 TIM3_SP[x]配置
	OCREF_CLR 使能的输入源的有效电平触发通过逻辑或的方式组合
	TIM3_OCCSE[x] = 0: 相应触发源不使能
	TIM3_OCCSE[x] = 1: 相应触发源使能
Bits 7:6	保留，必须保持复位值
Bits 5:0	TIM3_SP[5:0]: TIM3 OCREF_CLR 输入触发源的有效电平选择，默认值 000000
	所有输入源的有效触发电平，通过控制位 TIM3_SP[x]配置（低位对齐）
	TIM3_SP[x] = 0: 相应触发源的有效电平是高
	TIM3_SP[x] = 1: 相应触发源的有效电平是低
<i>注: TIMx_OCI_TRIG 和 TIMx_TRGO 不同，详见 TIMx 寄存器说明。</i>	

6.3.7 互联配置寄存器 6 (SysCtrl_EDU_CFG6)

地址偏移: 0x38

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.										TIM15_BK2SE[5:0]					
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TIM15_BKSE[6:0]							Res.	TIM15_SP[6:0]						
rw								rw							

Bits 31:22	保留，必须保持复位值
Bits 21:16	TIM15_BK2SE[5:0]: TIM15 刹车 2 输入触发源使能控制，默认值 000000
	TIM15 刹车 2 输入触发源为 ADC 模拟模块看门狗输出/ TIM16_OC1_TRIG / TIM17_

	OC1_TRIG / ACMP2 / ACMP1 / ACMP0 输出（从高位到低位）
	所有输入源均可被配置为使能或不使能，通过控制位 TIM15_BK2SE[x]配置
	所有输入源的有效触发电平，通过控制位 TIM15_SP[x]配置
	刹车 2 使能的输入源的有效电平触发通过逻辑或的方式组合
	TIM15_BK2SE[x] = 0: 相应触发源不使能
	TIM15_BK2SE[x] = 1: 相应触发源使能
Bit 15	保留，必须保持复位值
Bits 14:8	TIM15_BKSE[6:0]: TIM15 刹车输入触发源使能控制，默认值 0000000
	TIM15 刹车输入触发源为外部 GPIO 输入/ADC 模拟模块看门狗输出/TIM1_TRGO / TIM3_TRGO / ACMP2 / ACMP1 / ACMP0 输出（从高位到低位）
	所有输入源均可被配置为使能或不使能，通过控制位 TIM15_BKSE[x]配置
	所有输入源的有效触发电平，通过控制位 TIM15_SP[x]配置
	刹车使能的输入源的有效电平触发通过逻辑或的方式组合
	TIM15_BKSE[x] = 0: 相应触发源不使能
	TIM15_BKSE[x] = 1: 相应触发源使能
Bit 7	保留，必须保持复位值
Bits 6:0	TIM15_SP[6:0]: TIM15 刹车/刹车 2 输入触发源的有效电平选择，默认值 0000000
	所有输入源的有效触发电平，通过控制位 TIM15_SP[x]配置（低位对齐）
	TIM15_SP[x] = 0: 相应触发源的有效电平是高
	TIM15_SP[x] = 1: 相应触发源的有效电平是低
<i>注: TIMx_OC1_TRIG 和 TIMx_TRGO 不同, 详见 TIMx 寄存器说明。</i>	

6.3.8 互联配置寄存器 7 (SysCtrl_EDU_CFG7)

偏移地址: 0x3C

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.										TIM16_BK2SE[5:0]					
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TIM16_BKSE[6:0]							Res.	TIM16_SP[6:0]						
rw								rw							

Bits 31:22	保留，必须保持复位值
Bits 21:16	TIM16_BK2SE[5:0]: TIM16 刹车 2 输入触发源使能控制，默认值 000000
	TIM16 刹车 2 输入触发源为 ADC 模拟模块看门狗输出/ TIM15_TRGO / TIM17_OC1_TRIG / ACMP2 / ACMP1 / ACMP0 输出（从高位到低位）
	所有输入源均可被配置为使能或不使能，通过控制位 TIM16_BK2SE[x]配置
	所有输入源的有效触发电平，通过控制位 TIM16_SP[x]配置
	刹车 2 使能的输入源的有效电平触发通过逻辑或的方式组合
	TIM16_BK2SE[x] = 0: 相应触发源不使能
	TIM16_BK2SE[x] = 1: 相应触发源使能

Bit	15	保留，必须保持复位值
Bits	14:8	TIM16_BKSE[6:0]: TIM16 刹车输入触发源使能控制，默认值 0000000
		TIM16 刹车输入触发源为外部 GPIO 输入/ADC 模拟模块看门狗输出/TIM1_TRGO / TIM3_TRGO / ACMP2 / ACMP1 / ACMP0 输出（从高位到低位）
		所有输入源均可被配置为使能或不使能，通过控制位 TIM16_BKSE[x]配置
		所有输入源的有效触发电平，通过控制位 TIM16_SP[x]配置
		刹车使能的输入源的有效电平触发通过逻辑或的方式组合
		TIM16_BKSE[x] = 0: 相应触发源不使能
		TIM16_BKSE[x] = 1: 相应触发源使能
Bit	7	保留，必须保持复位值
Bits	6:0	TIM16_SP[6:0]: TIM16 刹车/刹车 2 输入触发源的有效电平选择，默认值 0000000
		所有输入源的有效触发电平，通过控制位 TIM16_SP[x]配置（低位对齐）
		TIM16_SP[x] = 0: 相应触发源的有效电平是高
		TIM16_SP[x] = 1: 相应触发源的有效电平是低
<i>注: TIMx_OCI_TRIG 和 TIMx_TRGO 不同，详见 TIMx 寄存器说明。</i>		

6.3.9 系统密钥寄存器 (SysCtrl_KEY)

地址偏移: 0x7C

复位值: 0x0000_0002

所有 SysCtrl 相关寄存器受密钥寄存器保护，通过写 0x05FA_659A 到密钥寄存器才能解锁；当其他数据写入寄存器，或者 32 个系统时钟周期后，寄存器恢复到写保护状态。

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

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				LOCK_CNT[4:0]				Res.				LOCK	WR_ERR		
														r	r

写入:

Bits	31:0	KEY[31:0]: 写入密钥以解锁或者锁定寄存器
------	------	----------------------------------

读出:

Bits	31:13	保留，必须保持复位值
Bits	12:8	LOCK_CNT[4:0]: 锁定窗口计数值
Bits	7:2	保留，必须保持复位值
Bit	1	LOCK: 锁定状态
		0: 写保护已被解除

	1: 写保护锁定（默认）
Bit 0	WR_ERR: 寄存器写错误标志
	0: 没有写保护错误（解除写保护之后成功写入寄存器）（默认）
	1: 写保护状态下的寄存器写入操作会被忽略

7. 通用 I/O 端口 (GPIO)

LCM32F037 包含多达 30 个快速 I/O 端口，可以允许 5V 电压。驱动能力和斜率两档可调，所有 I/O 都有大电流的功能，具有速度选择以更好地管理内部噪声、功耗、电磁辐射，每一个 I/O 端口均可以实现按位操作。

7.1 主要特征

- 每个 I/O 端口可被软件配置为悬空输入、上拉输入、下拉输入、推挽输出、开漏输出、开源输出。
- 支持不同的外设引脚功能复用。
- I/O 端口包含一到两路模拟通道，软件可配置关断或者使能。当 I/O 包含两路模拟通道时，两路模拟通道可以同时打开，内部导通。
- 支持外设复用的重映射：I/O 端口支持外设复用的重映射功能，那些复用的外设功能不仅可以通过默认的引脚实现，还可以通过其他可重映射的引脚实现。这使得引脚的选择更加灵活，制板更加方便，用户可以在选定的设备下实现最多数量的外设功能。
- I/O 功能配置可通过执行一个特定的读写操作序列来锁住，防止意外篡改。
- 每个 I/O 端口按位单独配置。

7.2 功能描述

GPIO 接口的基本结构如图 7-1 所示，每一位可以由软件分别配置成多种模式：

- 输入悬空
- 输入上拉
- 输入下拉
- 模拟输入
- 开漏输出
- 推挽式输出
- 推挽式复用功能
- 开漏复用功能

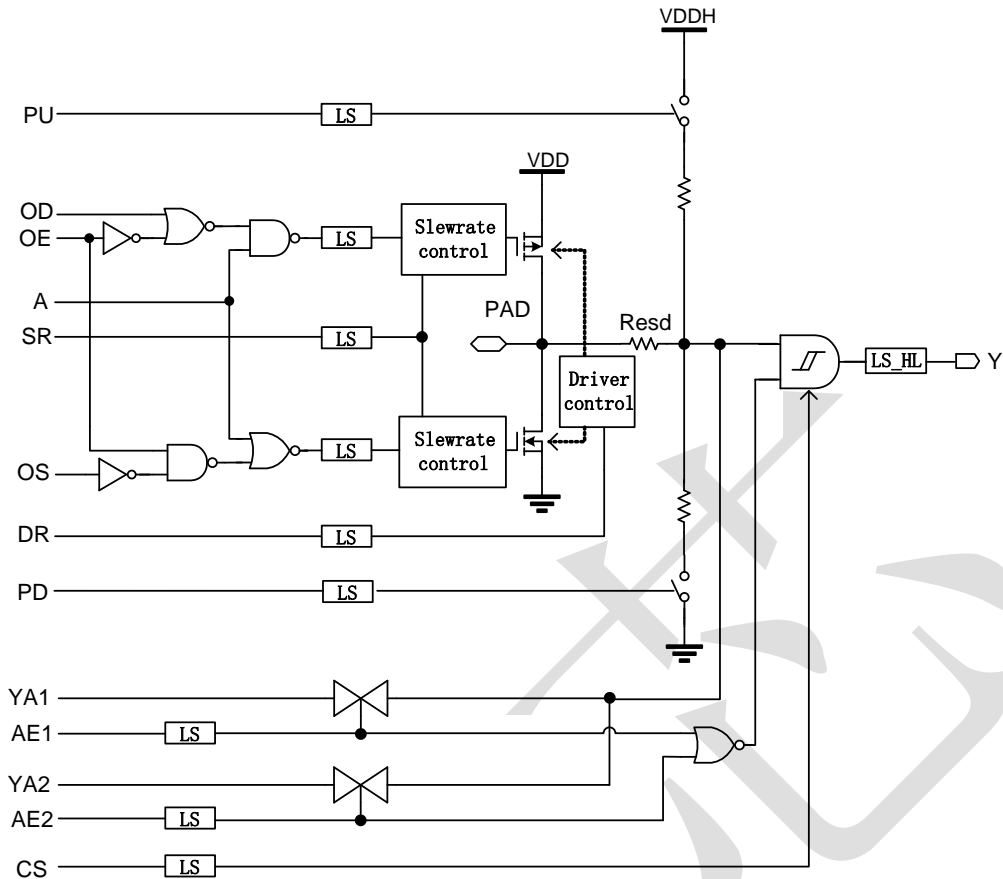


图 7-1 GPIO 接口的基本结构

表 7-1 GPIO 端口说明

端口名	I/O	端口定义	电平
VDDH	P	ESD 电源及 POSTDRIVER 电源	3.3V/5.5V
VSSH	G	模拟地	0
VDD	P	PREDRIVER 电源 1.5V	1.5V
GND	G	数字地	0
PAD	I/O	PAD 端	3.3V/5.5V
Y	I	数据输入信号	1.5V
YA1	I	模拟通道 1 输入端口	A
YA2	I	模拟通道 2 输入端口 (RCMCU_PLBMUX_2ANA_OP 中, YA2 为 OP 的模拟开关)	A
AE1	I	模拟通道选择信号, AE1=1 复用为模拟通道 1, 此时 Y=0, 复用为模拟通道时还要配置 OE=0	1.5V
AE2	I	模拟通道选择信号, AE2=1 复用为模拟通道 2, 此时 Y=0, 复用为模拟通道时还要配置 OE=0; AE1 和 AE2 同时设置为 1 时, 表示 YA1 和 YA2 及 PAD 都相连	1.5V
CS	I	迟滞选择控制: CS = 0: Schmitt 触发器输入 (迟滞输入) CS = 1: CMOS 输入 (无迟滞输入)	1.5V

A	I	推挽输出的输入信号	1.5V
OE(MODER)	I	推挽输出使能， OE=0: 三态输出， OE=1 且 OS=0 且 OD=0: 推挽输出	1.5V
OS	I	开源输出使能 OE=1 且 OS=1, 开源输出	1.5V
OD	I	开漏输出使能 OE=1 且 OD=1, 开漏输出	1.5V
PU	I	弱上拉电阻使能, PU=1 时弱上拉有效	1.5V
PD	I	弱下拉电阻使能, PD=1 时弱下拉有效	1.5V
SR	I	压摆率控制, 0: 高 SR, 1: 低 SR	1.5V
DR	I	驱动强度控制, 0: 大驱动; 1: 小驱动	1.5V

7.3 引脚说明

表 7-2 引脚排列表中使用的图例/缩略语

名称	缩写	定义
引脚名称		除非在引脚名下面的括号中特别说明, 复位期间和复位后的引脚功能与实际引脚名相同
引脚类型	S	电源引脚
	I	仅输入引脚
	I/O	输入/输出引脚
I/O结构	1ANA	只包含一路复用模拟通道
	2ANA	包含两路复用模拟通道, 两路普通模拟开关 (PAD经过ESD电阻后接到模拟开关)
	2OP	包含两路复用模拟通道, 两路低内阻模拟开关 (PAD直接接到模拟开关, 用于运放)
	ANA_OP	包含两路复用模拟通道, 一路普通模拟开关和一路低内阻模拟开关
注释		除非特别注释说明, 否则在复位期间和复位后所有 I/O 都设为浮空输入
引脚功能	可选复用功能	通过 GPIOx_AFL/H、GPIOx_MODE 寄存器选择的功能 (数字复用)
	外部复用功能	通过系统寄存器选择的功能, 优先级高于可选复用功能 (数字复用)
	模拟复用功能1	通过系统寄存器或者 GPIOx_AFL/H、GPIOx_MODE 寄存器选择的模拟功能 1
	模拟复用功能2	通过系统寄存器或者 GPIOx_AFL/H、GPIOx_MODE 寄存器选择的模拟功能 2

表 7-3 LCM32F037 引脚定义及模拟复用功能 (AN) 映射

引脚号		引脚名 (复位后的功能)	引脚类型	I/O 结构	可选复用功能	外部功能	模拟复用功能 (AN)	
LQFP32/ QFN32	SSOP24						YA1	YA2
		TESTEN				测试功能		
1		PF4	I/O	1ANA	SSP2_FSS/TIM1_CH1N/ TIM1_CH3/TIM3_CH1/			
2	21	PF0	I/O	2OP	SSP2_TXD/ I2C0_SDA/ TIM1_CH3/ TIM15_CH1/ UART0_TX/ SSP2_RXD		CPN02/ OP1P3/ CPP07 ¹	OSCL_IN / OSCH_IN
3	22	PF1	I/O	2OP	SSP2_RXD/ I2C0_SCL/ TIM1_CH2/ TIM15_CH1N/		CPP14 / OP1N3/ OPA_NIN2	OSCL_OUT / OSCH_OUT

					URAT0_RX/ SSP2_TXD			
4	23	PF2 (nRST)	I/O	1ANA	TIM1_CH1/ TIM3_CH3/ TIM15_CH1/TIM16_CH1N	nRST ²		
	24	VSS/VSSA	S					
5	1	VDDA	S					
5	1	VDDH	S					
6	2	PA0	I/O	2ANA	UART1_CTS/ TIM1_ETR/ TIM16_CH1/ UART1_RX/ CP0_OUT	ADCIN[0] / CPP04 / CPN00	PMU_VRH	
7	3	PA1	I/O	2ANA	EVENTOUT/ UART1_RTS/ TIM16_CH1N/ UART1_TX/ TIM15_CH1N/ IR_OUT	ADCIN[1]	CPP05/ CPP10	
8		PA2	I/O	2OP	TIM15_CH1/ UART1_TX/ CP1_OUT/ UART1_RX	CPN01/ OPOPO/ CPP06	OP2OEX / ADCIN[2]	
9	4	PA3	I/O	ANA_OP	TIM15_CH2/ UART1_RX/ TIM1_CH1N/ SSP0_TXD/ CP0_OUT/ SSP0_RXD	ADCIN[3] / CPN10/ CPP11	OP00/ CPP17	
10	5	PA4	I/O	2OP	SSP0_FSS/UART0_RTS/ TIM17_CH1/ TIM14_CH1/ WT_BUZ/ CP1_OUT	DAC_OUT0	ADCIN[4] / OP0N0 / CPP00	
11	6	PA5	I/O	2OP	SSP0_SCK/ TIM15_CH1/ TIM1_CH2N/WT_nBuz/ CP2_OUT/ TIM3_ETR	ADCIN[5] / OP2P1 / CPP12	DAC_OUT1	
12	7	PA6	I/O	ANA_OP	SSP0_RXD/TIM3_CH1/ TIM1_BKIN/CP0_OUT/ TIM16_CH1/ EVENTOUT/ SSP0_TXD	ADCIN[6] / CPP01	OP1P0 / OP2N0/ OPA_NIN3	
13	8	PA7	I/O	2OP	SSP0_TXD/ TIM3_CH2/ TIM1_CH1N/CP1_OUT/ TIM14_CH1/ TIM17_CH1/ EVENTOUT/ SSP0_RXD	ADCIN[7] / OP1P1/ OP2N1/ CPP02	OP10/ CPP16	
14	9	PB0	I/O	ANA_OP	EVENTOUT/ TIM3_CH3/ TIM1_CH2N/ UART0_TX/ UART0_RX		ADCIN[8]/ CPP13/ OP2P0/ CPN11/ OP1N1	
15	10	PB1	I/O	2OP	TIM14_CH1/TIM3_CH4/ TIM1_CH3N/UART1_RTS/ UART0_RX/UART0_TX	ADCIN[9]/ CPP20/ OP1N0/ CPN03	OP20/ CPP15	
16		PF6	I/O	ANA_OP	SSP2_FSS/TIM1_CH2N/ TIM1_BKIN/UART1_TX/ TIM15_BKIN/EVENTOUT	ADCIN[11]/ OPOOEX	OP0N3/ CPP21	
17		PF7	I/O	ANA_OP	SSP2_SCK/TIM1_CH1N/ TIM3_CH4/ URAT1_RX	ADCIN[12]/ OP1OEX	OP0P3 / CPN12/ CPP22	
18	11	PA8	I/O	ANA_OP	MCO/UART0_CTS/	ADCIN[10]/	OP0N1/	

					TIM1_CH1/ EVENTOUT/ SSP2_TXD/ SSP2_RXD/ CP0_OUT		ELVI ³ / CPP03	OPA_NIN1
19	12	PA9	I/O	ANA_OP	TIM15_CH2N/ TIM15_BKIN/ UART0_TX/ TIM1_CH2/ I2CO_SCL/ SSP2_RXD/ SSP2_TXD/ UART0_RX		ADCIN[14]	CPN13/ OP0P1
20	13	PA10	I/O	2ANA	TIM15_CH2/ TIM17_BKIN/ UART0_RX/ TIM1_CH3/ I2CO_SDA		ADCIN[15]/ CPP23	DAC_OUT0
21		PA11	I/O	2ANA	EVENTOUT/ UART0_CTS/ TIM1_CH4/ TIM15_CH1N/ CP0_OUT/ I2CO_SCL			
22		PA12	I/O	1ANA	EVENTOUT/ UART0_RTS/ TIM1_ETR/ TIM16_CH1/ TIM15_CH1/ I2CO_SDA/			
23	14	PA13	I/O	1ANA	IR_OUT/ TIM1_CH1N/ TIM1_CH4/ UART1_TX/ WT_BUZ/ CP1_OUT	SWDIO ⁴		
24	15	PA14	I/O	1ANA	UART1_TX/ WT_nBUZ/ CP2_OUT/ UART1_RX	SWCLK ⁴		
25	16	PA15	I/O	1ANA	SSP0_FSS/UART1_RX/ TIM17_CH1/EVENTOUT/ TIM1_CH2/TIM16_CH1N/ TIM1_BKIN/TIM15_BKIN			
26		PB3	I/O	1ANA	SSP0_SCK/TIM1_CH1N/ TIM1_CH2N/TIM16_CH1/ CP1_OUT			
27		PB4	I/O	1ANA	SSP0_RXD/TIM3_CH1/ TIM1_CH2N/UART0_RX/ TIM17_BKIN/SSP0_TXD			
28	17	PB5	I/O	1ANA	SSP0_TXD/TIM3_CH2/ TIM16_BKIN/TIM1_CH3/ UART0_TX/UART1_CTS/ CP2_OUT/ SSP0_RXD		ADCIN[13] VBG_BUFOUT	
29	18	PB6	I/O	ANA_OP	UART0_TX/I2CO_SCL/ TIM16_CH1N/TIM15_CH2/ TIM1_CH3N/ UART0_RX		DAC_OUT0	OP2P3 / CPN22
30	19	PB7	I/O	ANA_OP	UART0_RX/I2CO_SDA/ TIM17_CH1N/TIM1_CH2N/ UART0_TX/TIM1_CH1/ TIM17_CH1/SSP2_FSS		DAC_OUT1	OP2N3/ CPN23
31	20	PF3 (BOOT0)	I/O	1ANA	SSP2_SCK/ TIM1_CH1N/ TIM1_CH2/ TIM3_CH2/	BOOT0 ⁵		

					TIM15_CH2/ TIM16_CH1/ TIM17_CH1N/ TIM3_ETR			
32		VSS/VSSA	S					

注1: ACMP 与 OPA 输入引脚格式。ACMP: CP 序号 + 引脚正负端 + 输入端口; OPA: OP 序号 + 引脚正负端 + 输入端口。例如: CPP11 表示 ACMP1 正端输入 端口 1; OP2N3 表示 OPA2 负端输入 端口 3。

注2: 上电复位后, 这个引脚缺省配置为外部复位引脚 nRST。

注3: ELVI 为外部输入低压检测。

注4: 系统复位后, 这些引脚配置为可选复用功能SWDIO和SWCLK, SWDIO引脚内部上拉, SWCLK引脚内部下拉。

注5: 根据选项字节配置, 在系统复位期间可以作为BOOT0引脚, 以选择启动模式; 后续为正常功能。

注6: I/O驱动强度分为两档, 3.3V供电时为4mA/8mA; 5V供电时为8mA/16mA。

表 7-4 LCP037A 系列引脚定义

引脚名(复位后的功能)		引脚类型	I/O 结构		可选复用功能	外部功能	模拟复用功能(AN)	
QFN32	SSOP24						AN1	AN2
		TESTEN				测试功能		
		PF4	I/O	1ANA	SSP2_FSS/TIM1_CH1N/ TIM1_CH3/TIM3_CH1/			
2	16	PF0	I/O	2OP	SSP2_TXD/ I2C0_SDA/ TIM1_CH3/ TIM15_CH1/ UART0_TX/ SSP2_RXD		CPN02 ¹ / OP1P3/ CPP07	OSCL_IN / OSCH_IN
3	17	PF1	I/O	2OP	SSP2_RXD/ I2C0_SCL/ TIM1_CH2/ TIM15_CH1N/ URAT0_RX/ SSP2_TXD		CPP14 / OP1N3/ OPA_NIN2	OSCL_OUT / OSCH_OUT
4	18	PF2(nRST)	I/O	1ANA	TIM1_CH1/ TIM3_CH3/ TIM15_CH1/TIM16_CH1N	nRST ²		
5		VSS/VSSA	S					
6	6	VDD/VDDA	S					
7	19	PA0	I/O	2ANA	UART1_CTS/ TIM1_ETR/ TIM16_CH1/ UART1_RX/ CP0_OUT		ADCIN[0] / CPP04 / CPN00	PMU_VRH
8	20	PA1	I/O	2ANA	EVENTOUT/ UART1_RTS/ TIM16_CH1N/ UART1_TX/ TIM15_CH1N/ IR_OUT		ADCIN[1]	CPP05/ CPP10
9	21	PA2	I/O	2OP	TIM15_CH1/ UART1_TX/ CP1_OUT/ UART1_RX		CPN01/ OP0P0/ CPP06	OP2OEX / ADCIN[2]
		PA3	I/O	ANA_OP	TIM15_CH2/ UART1_RX/ TIM1_CH1N/ SSP0_TXD/ CP0_OUT/ SSP0_RXD		ADCIN[3] / CPN10/ CPP11	OP00/ CPP17
10	22	PA4	I/O	2OP	SSP0_FSS/UART0_RTS/ TIM17_CH1/ TIM14_CH1/ WT_BUZ/ CP1_OUT		DAC_OUT0	ADCIN[4] / OPON0 / CPP00

11	23	PA5	I/O	2OP	SSPO_SCK/ TIM15_CH1/ TIM1_CH2N/WT_nBuz/ CP2_OUT/ TIM3_ETR		ADCIN[5] / OP2P1 / CPP12	DAC_OUT1
12	24	PA6	I/O	ANA_OP	SSPO_RXD/TIM3_CH1/ TIM1_BKIN/CP0_OUT/ TIM16_CH1/ EVENTOUT/ SSPO_TXD		ADCIN[6] / CPP01	OP1P0 / OP2N0/ OPA_NIN3
13	1	PA7	I/O	2OP	SSPO_TXD/ TIM3_CH2/ TIM1_CH1N/CP1_OUT/ TIM14_CH1/ TIM17_CH1/ EVENTOUT/ SSPO_RXD		ADCIN[7] / OP1P1/ OP2N1/ CPP02	OP1O/ CPP16
14	2	PB0	I/O	ANA_OP	EVENTOUT/ TIM3_CH3/ TIM1_CH2N/ UART0_TX/ UART0_RX			ADCIN[8]/ CPP13/ OP2P0/ CPN11/ OP1N1
15	3	PB1	I/O	2OP	TIM14_CH1/TIM3_CH4/ TIM1_CH3N/UART1_RTS/ UART0_RX/UART0_TX		ADCIN[9]/ CPP20/ OP1N0/ CPN03	OP2O/ CPP15
16		PF6	I/O	ANA_OP	SSP2_FSS/TIM1_CH2N/ TIM1_BKIN/UART1_TX/ TIM15_BKIN/EVENTOUT		ADCIN[11]/ OPO0EX	OP0N3/ CPP21
17		PF7	I/O	ANA_OP	SSP2_SCK/TIM1_CH1N/ TIM3_CH4/ URAT1_RX		ADCIN[12]/ OP1OEX	OP0P3 / CPN12/ CPP22
18	3	PA8	I/O	ANA_OP	MCO/UART0_CTS/ TIM1_CH1/ EVENTOUT/ SSP2_TXD/ SSP2_RXD/ CP0_OUT		ADCIN[10]/ ELVI ³ / CPP03	OP0N1/ OPA_NIN1
19	4	PA9	I/O	ANA_OP	TIM15_CH2N/ TIM15_BKIN/ UART0_TX/ TIM1_CH2/ I2C0_SCL/ SSP2_RXD/ SSP2_TXD/ UART0_RX		ADCIN[14]	CPN13/ OP0P1
20		PA10	I/O	2ANA	TIM15_CH2/ TIM17_BKIN/ UART0_RX/ TIM1_CH3/ I2C0_SDA		ADCIN[15]/ CPP23	DAC_OUT0
		PA11	I/O	2ANA	EVENTOUT/ UART0_CTS/ TIM1_CH4/ TIM15_CH1N/ CP0_OUT/ I2C0_SCL			
		PA12	I/O	1ANA	EVENTOUT/ UART0_RTS/ TIM1_ETR/ TIM16_CH1/ TIM15_CH1/ I2C0_SDA/			
21	4	PA13	I/O	1ANA	IR_OUT/ TIM1_CH1N/ TIM1_CH4/ UART1_TX/ WT_BUZ/ CP1_OUT	SWDIO ⁴		
22	5	PA14	I/O	1ANA	UART1_TX/ WT_nBUZ/	SWCLK ⁴		

					CP2_OUT/ UART1_RX			
23		PA15	I/O	1ANA	SSP0_FSS/UART1_RX/ TIM17_CH1/EVENTOUT/ TIM1_CH2/TIM16_CH1N/ TIM1_BKIN/TIM15_BKIN			
		PB3	I/O	1ANA	SSP0_SCK/TIM1_CH1N/ TIM1_CH2N/TIM16_CH1/ CP1_OUT			
		PB4	I/O	1ANA	SSP0_RXD/TIM3_CH1/ TIM1_CH2N/UART0_RX/ TIM17_BKIN/SSP0_TXD			
		PB5	I/O	1ANA	SSP0_TXD/TIM3_CH2/ TIM16_BKIN/TIM1_CH3/ UART0_TX/UART1_CTS/ CP2_OUT/ SSP0_RXD		ADCIN[13] VBG_BUFOUT	
		PB6	I/O	ANA_OP	UART0_TX/I2C0_SCL/ TIM16_CH1N/TIM15_CH2/ TIM1_CH3N/ UART0_RX		DAC_OUT0	OP2P3 / CPN22
		PB7	I/O	ANA_OP	UART0_RX/I2C0_SDA/ TIM17_CH1N/TIM1_CH2N/ UART0_TX/TIM1_CH1/ TIM17_CH1/SSP2_FSS		DAC_OUT1	OP2N3/ CPN23
		PF3(BOOT0)	I/O	1ANA	SSP2_SCK/ TIM1_CH1N/ TIM1_CH2/ TIM3_CH2/ TIM15_CH2/ TIM16_CH1/ TIM17_CH1N/ TIM3_ETR	BOOT0 ⁵		
24	6	VDD5	P				预驱模块 5V 输出口	
25	7	VCCP	P				预驱模块 5V 电源输入端	
26	8	VCC	O				预驱模块驱动电源输入端	
27	9	HO1	O				预驱模块 1 通道高端输出口, 控制 PMOS 管的导通与截止	
28	10	LO1	O				预驱模块 1 通道低端输出口, 控制 NMOS 管的导通与截止	
29	11	HO2	O				预驱模块 2 通道高端输出口, 控制 PMOS 管的导通与截止	
30	12	LO2	O				预驱模块 2 通道低端输出口, 控制 NMOS 管的导通与截止	
31	13	HO3	O				预驱模块 3 通道高端输出口, 控制 PMOS 管的导通与截止	
32	14	LO3	O				预驱模块 3 通道低端输出口, 控制 NMOS 管的导通与截止	
1	15	VSSP	P				预驱模块地	

注 1: ACMP 与 OPA 输入引脚格式。ACMP: CP 序号 + 引脚正负端 + 输入端口; OPA: OP 序号 + 引脚正负端 + 输入端口。例如: CPP11 表示 ACMP1 正端输入 端口 1; OP2N3 表示 OPA2 负端输入 端口 3。

注2: 上电复位后, 这个引脚缺省配置为外部复位引脚 nRST。

注3: ELVI 为外部输入低压检测。

注4: 系统复位后, 这些引脚配置为可选复用功能SWDIO和SWCLK, SWDIO引脚内部上拉, SWCLK引脚内部下拉。

注5: 根据选项字节配置, 在系统复位期间可以作为BOOT0引脚, 以选择启动模式; 后续为正常功能。

注6: I/O驱动强度分为两档, 3.3V供电时为4mA/8mA; 5V供电时为8mA/16mA。

表 7-5 LCP037A 系列预驱引脚描述

驱动芯片引脚	I/O 类型	引脚描述	引脚位置
HIN1	I	逻辑输入控制信号高电平有效, 控制高端功率 MOS 管的	与主控 MCU PB7 引脚相连
HIN2	I	导通与截止	与主控 MCU PF3 引脚相连
HIN3	I	“0”是关闭功率 MOS 管; “1”是开启功率 MOS 管	与主控 MCU PF4 引脚相连
LIN1	I	逻辑输入控制信号低电平有效, 控制低端功率 MOS 管的	与主控 MCU PB3 引脚相连
LIN2	I	导通与截止	与主控 MCU PB4 引脚相连
LIN3	I	“0”是关闭功率 MOS 管; “1”是开启功率 MOS 管	与主控 MCU PB6 引脚相连

表 7-6 LCP037B 系列引脚定义

引脚名(复位后的功能)			引脚类型	I/O 结构		可选复用功能	外部功能	模拟复用功能(AN)	
QFN40	QFN32	LQFP32						AN1	AN2
			TESTEN				测试功能		
			PF4	I/O	1ANA	SSP2_FSS/TIM1_CH1N/ TIM1_CH3/TIM3_CH1/			
2	2	2	PF0	I/O	2OP	SSP2_TXD/ I2C0_SDA/ TIM1_CH3/ TIM15_CH1/ UART0_TX/ SSP2_RXD		CPN02/ OP1P3/ CPP07 ¹	OSCL_IN / OSCH_IN
3	3	3	PF1	I/O	2OP	SSP2_RXD/ I2C0_SCL/ TIM1_CH2/ TIM15_CH1N/ URAT0_RX/ SSP2_TXD		CPP14 / OP1N3/ OPA_NIN2	OSCL_OUT / OSCH_OUT
4	4	4	PF2 (nRST)	I/O	1ANA	TIM1_CH1/ TIM3_CH3/ TIM15_CH1/TIM16_CH1N	nRST ²		
5	5	5	VSS/VSSA	S					
6	6	6	VDDA/VDDH	S					
7	7	7	PA0	I/O	2ANA	UART1_CTS/ TIM1_ETR/ TIM16_CH1/ UART1_RX/ CP0_OUT		ADCIN[0] / CPP04 / CPN00	PMU_VRH
8	8	8	PA1	I/O	2ANA	EVENTOUT/ UART1_RTS/ TIM16_CH1N/ UART1_TX/ TIM15_CH1N/ IR_OUT		ADCIN[1]	CPP05/ CPP10
9	9	9	PA2	I/O	2OP	TIM15_CH1/ UART1_TX/ CP1_OUT/ UART1_RX		CPN01/ OP0P0/ CPP06	OP2OEX / ADCIN[2]
10			PA3	I/O	ANA_OP	TIM15_CH2/ UART1_RX/		ADCIN[3] /	OP00/ CPP17

						TIM1_CH1N/ SSP0_TXD/ CP0_OUT/ SSP0_RXD		CPN10/ CPP11	
11	10	0	PA4	I/O	2OP	SSP0_FSS/UART0_RTS/ TIM17_CH1/ TIM14_CH1/ WT_BUZ/ CP1_OUT		DAC_OUT0	ADCIN[4] / OP0N0 / CPP00
12	11	11	PA5	I/O	2OP	SSP0_SCK/ TIM15_CH1/ TIM1_CH2N/WT_nBuz/ CP2_OUT/ TIM3_ETR		ADCIN[5] / OP2P1 / CPP12	DAC_OUT1
13	12	12	PA6	I/O	ANA_OP	SSP0_RXD/TIM3_CH1/ TIM1_BKIN/CP0_OUT/ TIM16_CH1/ EVENTOUT/ SSP0_TXD		ADCIN[6] / CPP01	OP1P0 / OP2N0/ OPA_NIN3
14	13	13	PA7	I/O	2OP	SSP0_TXD/ TIM3_CH2/ TIM1_CH1N/CP1_OUT/ TIM14_CH1/ TIM17_CH1/ EVENTOUT/ SSP0_RXD		ADCIN[7] / OP1P1/ OP2N1/ CPP02	OP10/ CPP16
15	14	14	PB0	I/O	ANA_OP	EVENTOUT/ TIM3_CH3/ TIM1_CH2N/ UART0_TX/ UART0_RX			ADCIN[8]/ CPP13/ OP2P0/ CPN11/ OP1N1
16	15	15	PB1	I/O	2OP	TIM14_CH1/TIM3_CH4/ TIM1_CH3N/UART1_RTS/ UART0_RX/UART0_TX		ADCIN[9]/ CPP20/ OP1N0/ CPN03	OP20/ CPP15
17			PF6	I/O	ANA_OP	SSP2_FSS/TIM1_CH2N/ TIM1_BKIN/UART1_TX/ TIM15_BKIN/EVENTOUT		ADCIN[11]/ OP0OEX	OP0N3/ CPP21
18			PF7	I/O	ANA_OP	SSP2_SCK/TIM1_CH1N/ TIM3_CH4/ URAT1_RX		ADCIN[12]/ OP1OEX	OP0P3 / CPN12/ CPP22
19	16	16	PA8	I/O	ANA_OP	MCO/UART0_CTS/ TIM1_CH1/ EVENTOUT/ SSP2_TXD/ SSP2_RXD/ CP0_OUT		ADCIN[10]/ ELVI ³ / CPP03	OP0N1/ OPA_NIN1
20	17	17	PA9	I/O	ANA_OP	TIM15_CH2N/ TIM15_BKIN/ UART0_TX/ TIM1_CH2/ I2C0_SCL/ SSP2_RXD/ SSP2_TXD/ UART0_RX		ADCIN[14]	CPN13/ OP0P1
21	18	18	PA10	I/O	2ANA	TIM15_CH2/ TIM17_BKIN/ UART0_RX/ TIM1_CH3/ I2C0_SDA		ADCIN[15]/ CPP23	DAC_OUT0
22			PA11	I/O	2ANA	EVENTOUT/ UART0_CTS/ TIM1_CH4/ TIM15_CH1N/ CP0_OUT/ I2C0_SCL			
23			PA12	I/O	1ANA	EVENTOUT/ UART0_RTS/ TIM1_ETR/ TIM16_CH1/ TIM15_CH1/ I2C0_SDA/			



24	18	19	PA13	I/O	1ANA	IR_OUT/ TIM1_CH1N/ TIM1_CH4/ UART1_TX/ WT_BUZ/ CP1_OUT	SWDIO ⁴		
25	19	19	PA14	I/O	1ANA	UART1_TX/ WT_nBUZ/ CP2_OUT/ UART1_RX	SWCLK ⁵		
26			PA15	I/O	1ANA	SSPO_FSS/UART1_RX/ TIM17_CH1/EVENTOUT/ TIM1_CH2/TIM16_CH1N/ TIM1_BKIN/TIM15_BKIN			
			PB3	I/O	1ANA	SSPO_SCK/TIM1_CH1N/ TIM1_CH2N/TIM16_CH1/ CP1_OUT			
			PB4	I/O	1ANA	SSPO_RXD/TIM3_CH1/ TIM1_CH2N/UART0_RX/ TIM17_BKIN/SSPO_TXD			
			PB5	I/O	1ANA	SSPO_TXD/TIM3_CH2/ TIM16_BKIN/TIM1_CH3/ UART0_TX/UART1_CTS/ CP2_OUT/ SSPO_RXD		ADCIN[13] VBG_BUFOUT	
			PB6	I/O	ANA_OP	UART0_TX/I2C0_SCL/ TIM16_CH1N/TIM15_CH2/ TIM1_CH3N/ UART0_RX		DAC_OUT0	OP2P3 / CPN22
			PB7	I/O	ANA_OP	UART0_RX/I2C0_SDA/ TIM17_CH1N/TIM1_CH2N/ UART0_TX/TIM1_CH1/ TIM17_CH1/SSP2_FSS		DAC_OUT1	OP2N3/ CPN23
			PF3 (BOOT0)	I/O	1ANA	SSP2_SCK/ TIM1_CH1N/ TIM1_CH2/ TIM3_CH2/ TIM15_CH2/ TIM16_CH1/ TIM17_CH1N/ TIM3_ETR	BOOT0 ⁶		
1	1	1	VDDP	S				预驱模块供电电源	
27	20	20	VSSP	S				预驱模块驱动地	
28	23	23	LO3	O				预驱模块低端驱动输出 3	
29	22	22	LO2	O				预驱模块低端驱动输出 2	
30	21	21	LO1	O				预驱模块低端驱动输出 1	
31			VSSPAD	S				封装框架地	
32	24	24	VS1	-				预驱模块高端悬浮地 1	
33	25	25	HO1	O				预驱模块高端驱动输出 1	
34	26	26	VB1	-				预驱模块高端悬浮电源 1	
35	27	27	VS2	-				预驱模块高端悬浮地 2	
36	28	28	HO2	O				预驱模块高端驱动输出 2	
37	29	29	VB2	-				预驱模块高端悬浮电源 2	
38	30	30	VS3	-				预驱模块高端悬浮地 3	



39	31	31	HO3	O				预驱模块高端驱动输出 3
40	32	32	VB3	-				预驱模块高端悬浮电源 3

注1: ACMP 与 OPA 输入引脚格式。ACMP: CP 序号 + 引脚正负端 + 输入端口; OPA: OP 序号 + 引脚正负端 + 输入端口。例如: CPP11 表示 ACMP1 正端输入 端口 1; OP2N3 表示 OPA2 负端输入 端口 3。

注2: 上电复位后, 这个引脚缺省配置为外部复位引脚 nRST。

注3: ELVI 为外部输入低压检测。

注4: 系统复位后, 这些引脚配置为可选复用功能SWDIO和SWCLK, SWDIO 引脚内部上拉, SWCLK 引脚内部下拉。

注5: 根据选项字节配置, 在系统复位期间可以作为BOOT0引脚, 以选择启动模式; 后续为正常功能。

注6: I/O 驱动强度分为两档, 3.3V 供电时为4mA/8mA; 5V 供电时为8mA/16mA。

表 7-7 LCP037B 系列预驱引脚描述

驱动芯片引脚	I/O 类型	引脚描述	引脚位置
HIN1	I	逻辑输入控制信号高电平有效, 控制高端功率 MOS 管的导通与截止 “0”是关闭功率 MOS 管; “1”是开启功率 MOS 管	与主控 MCU PB7 引脚相连
HIN2	I		与主控 MCU PF3 引脚相连
HIN3	I		与主控 MCU PF4 引脚相连
LIN1	I	逻辑输入控制信号低电平有效, 控制低端功率 MOS 管的导通与截止 “0”是关闭功率 MOS 管; “1”是开启功率 MOS 管	与主控 MCU PB3 引脚相连
LIN2	I		与主控 MCU PB4 引脚相连
LIN3	I		与主控 MCU PB6 引脚相连

表 7-8 端口 A 可选复用功能 (AF) 映射

引脚	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7
PA0	-	UART1_CTS	TIM1_ETR	TIM16_CH1	UART1_RX	-	CP0_OUT	-
PA1	EVENTOUT	UART1_RTS	TIM16_CH1N	-	UART1_TX	TIM15_CH1N	-	IR_OUT
PA2	TIM15_CH1	UART1_TX	-	-	-	-	CP1_OUT	UART1_RX
PA3	TIM15_CH2	UART1_RX	-	-	TIM1_CH1N	SSP0_TXD	CP0_OUT	SSP0_RXD
PA4	SSP0_FSS	UART0_RTS	-	TIM17_CH1	TIM14_CH1	WT_BUZ	CP1_OUT	-
PA5	SSP0_SCK	-	-	TIM15_CH1	TIM1_CH2N	WT_nBuz	CP2_OUT	TIM3_ETR
PA6	SSP0_RXD	TIM3_CH1	TIM1_BKIN	CP0_OUT	-	TIM16_CH1	EVENTOUT	SSP0_TXD
PA7	SSP0_TXD	TIM3_CH2	TIM1_CH1N	CP1_OUT	TIM14_CH1	TIM17_CH1	EVENTOUT	SSP0_RXD
PA8	MCO	UART0_CTS	TIM1_CH1	EVENTOUT	SSP2_TXD	SSP2_RXD	CP0_OUT	-
PA9	TIM15_CH2N	TIM15_BKIN	UART0_TX	TIM1_CH2	I2C0_SCL	SSP2_RXD	SSP2_TXD	UART0_RX
PA10	TIM15_CH2	TIM17_BKIN	UART0_RX	TIM1_CH3	I2C0_SDA	-	-	-
PA11	EVENTOUT	UART0_CTS	TIM1_CH4	TIM15_CH1N	-	-	CP0_OUT	I2C0_SCL
PA12	EVENTOUT	UART0_RTS	TIM1_ETR	TIM16_CH1	TIM15_CH1	-	-	I2C0_SDA
PA13	SWDIO	IR_OUT	TIM1_CH1N	TIM1_CH4	UART1_TX	WT_Buz	CP1_OUT	-
PA14	SWCLK	UART1_TX	-	-	-	WT_nBuz	CP2_OUT	UART1_RX
PA15	SSP0_FSS	UART1_RX	TIM17_CH1	EVENTOUT	TIM1_CH2	TIM16_CH1N	TIM1_BKIN	TIM15_BKIN

表 7-9 端口 B 可选复用功能 (AF) 映射

引脚	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7
PB0	EVENTOUT	TIM3_CH3	TIM1_CH2N	-	UART0_TX	-	UART0_RX	-
PB1	TIM14_CH1	TIM3_CH4	TIM1_CH3N	-	UART1_RTS	UART0_RX	-	UART0_TX
PB3	SSP0_SCK	TIM1_CH1N	TIM1_CH2N	TIM16_CH1	-	-	CP1_OUT	-

PB4	SSPO_RXD	TIM3_CH1	TIM1_CH2N	UART0_RX	-	TIM17_BKIN	-	SSPO_TXD
PB5	SSPO_TXD	TIM3_CH2	TIM16_BKIN	TIM1_CH3	UART0_TX	UART1_CTS	CP2_OUT	SSPO_RXD
PB6	UART0_TX	I2C0_SCL	TIM16_CH1N	TIM15_CH2	TIM1_CH3N	-	-	UART0_RX
PB7	UART0_RX	I2C0_SDA	TIM17_CH1N	TIM1_CH2N	UART0_TX	TIM1_CH1	TIM17_CH1	SSP2_FSS

表 7-10 端口 F 可选复用功能 (AF) 映射

引脚	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7
PF0	SSP2_TXD	I2C0_SDA	TIM1_CH3	TIM15_CH1	UART0_TX	-	-	SSP2_RXD
PF1	SSP2_RXD	I2C0_SCL	TIM1_CH2	TIM15_CH1N	URAT0_RX	-	-	SSP2_TXD
PF2	-	-	TIM1_CH1	TIM3_CH3	TIM15_CH1	TIM16_CH1N	-	-
PF3	SSP2_SCK	TIM1_CH1N	TIM1_CH2	TIM3_CH2	TIM15_CH2	TIM16_CH1	TIM17_CH1N	TIM3_ETR
PF4	SSP2_FSS	TIM1_CH1N	TIM1_CH3	TIM3_CH1	-	-	-	-
PF6	SSP2_FSS	TIM1_CH2N	TIM1_BKIN	-	UART1_TX	TIM15_BKIN	EVENTOUT	-
PF7	SSP2_SCK	TIM1_CH1N	TIM3_CH4	-	URAR1_RX	-	-	-

7.4 GPIO 寄存器描述

GPIO 相关寄存器相关列表如下，其中 GPIOA 基地址为 0x4800_0000，GPIOB 基地址为 0x4800_0200，GPIOF 基地址为 0x4800_0A00。

表 7-11 GPIO 相关寄存器表

名称	说明	读写权限	复位值	地址偏移
GPIOx_MODER	端口模式寄存器	R/W	GPIOA: 0x2800_0000 GPIOB: 0x0000_0000 GPIOF: 0x0000_0000	0x00
GPIOx_OD_BSRR	端口开漏设置/清除寄存器	R/W	0x0000_0000	0x04
GPIOx_SR_BSRR	端口斜率设置/清除寄存器	R/W	0x0000_0000	0x08
GPIOx_PU_BSRR	端口上拉设置/清除寄存器	R/W	GPIOA: 0x0000_2000 GPIOB: 0x0000_0000 GPIOF: 0x0000_0004	0x0C
GPIOx_PD_BSRR	端口下拉设置/清除寄存器	R/W	GPIOA: 0x0000_4000 GPIOB: 0x0000_0000 GPIOF: 0x0000_0000	0x10
GPIOx_IDR	端口输入数据寄存器	R/W	0x0000_XXXX	0x14
GPIOx_ODR	端口输出数据寄存器	R/W	0x0000_0000	0x18
GPIOx_BSRR	端口数据设置/清除寄存器	R/W	0x0000_0000	0x1C
GPIOx_LCKR	端口配置锁定寄存器	R/W	0x0000_0000	0x20
GPIOx_AFRL	端口复用功能配置寄存器低位	R/W	0x0000_0000	0x24
GPIOx_AFRH	端口复用功能配置寄存器高位	R/W	0x0000_0000	0x28
GPIOx_BTGLR	端口输出数据翻转寄存器	R/W	0x0000_0000	0x2C
GPIOx_DR_BSRR	端口驱动强度设置/清除寄存器	R/W	0x0000_FFFF	0x30
GPIOx_CS_BSRR	端口施密特触发器输入设置/清除寄存器	R/W	0x0000_0000	0x34
GPIOx_OS_BSRR	端口开源模式设置/清除寄存器	R/W	0x0000_0000	0x38



注：x 表示不确定；-表示未实现；u 表示不受复位影响；c 表示该位复位值因复位源而异；R 只读；W 只写；R-0 表示只读且读0；R/W 表示可读写（以后章节同上述）。

7.4.1 GPIO 端口模式寄存器（GPIOx_MODER）（x=A,B,F）

地址偏移：0x00

复位值：GPIOA: 0x2800_0000; GPIOB: 0x0000_0000; GPIOF: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[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
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	

Bits	2y+1:2y	MODERy[1:0]: 端口 x 功能配置位 (y=0…15)
		00: 输入模式 (默认)
		01: 通用输出
		10: 复用功能
		11: 模拟功能
注: PA13 和 PA14 在系统复位后自动配置为复用功能 SWDIO 和 SWCLK, SWDIO 引脚内部上拉, SWCLK 引脚内部下拉。		

7.4.2 GPIO 端口开漏设置/清除寄存器（GPIOx_OD_BSRR）（x=A,B,F）

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
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
Res.															
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
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
OD15	OD14	OD13	OD12	OD11	OD10	OD9	OD8	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	31:16	读:
		保留
		写:
		BRy: 端口 x 开漏配置复位 (y=0…15)
		0: 对开漏配置位无操作
		1: 复位开漏配置位
Bits	15:0	读:

	ODy : 端口 x 开漏配置状态
	0: 开漏输出禁止 (默认)
	1: 开漏输出使能
	写:
	BSy : 端口 x 开漏配置位 (y=0…15)
	0: 对开漏配置位无操作
	1: 设置开漏配置位

7.4.3 GPIO 端口斜率设置/清除寄存器 (GPIOx_SR_BSRR) (x=A,B,F)

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
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
Res.															
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
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
SR15	SR14	SR13	SR12	SR11	SR10	SR9	SR8	SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16	读:
	保留
	写:
	BRy : 端口 x 斜率配置复位 (y=0…15)
	0: 对斜率配置位无操作
	1: 复位斜率配置位
Bits 15:0	读:
	SRy : 端口 x 斜率配置状态
	0: 快速斜率状态 (默认)
	1: 缓慢斜率状态
	写:
	BSy : 端口 x 斜率配置位 (y=0…15)
	0: 对斜率配置位无操作
	1: 设置斜率配置位

7.4.4 GPIO 端口上拉设置/清除寄存器 (GPIOx_PU_BSRR) (x=A,B,F)

地址偏移: 0x0C

复位值: GPIOA: 0x0000_2000; GPIOB: 0x0000_0000; GPIOF: 0x0000_0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
Res.															
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
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
PU15	PU14	PU13	PU12	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16	读:
	保留
	写:
	BRy : 端口 x 上拉配置复位 (y=0...15)
	0: 对上拉配置位无操作
	1: 复位上拉配置位
Bits 15:0	读:
	PUy : 端口 x 上拉配置状态
	0: 上拉无效 (默认)
	1: 上拉有效
	写:
	BSy : 端口 x 上拉配置位 (y=0...15)
	0: 对上拉配置位无操作
	1: 设置上拉配置位
注: PA13 在系统复位后自动配置为复用功能 SWDIO, SWDIO 引脚内部上拉。同时 PF2 在上电复位后会自动配置为外部复位引脚 nRST, 引脚内部上拉。	

7.4.5 GPIO 端口下拉设置/清除寄存器 (GPIOx_PD_BSRR) (x=A,B,F)

地址偏移: 0x10

复位值: GPIOA: 0x0000_4000; GPIOB: 0x0000_0000; GPIOF: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
Res.															
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
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16	读:
	保留



	写:
	BRy: 端口 x 下拉配置复位 (y=0…15)
	0: 对下拉配置位无操作
	1: 复位下拉配置位
Bits 15:0	读:
	PDy: 端口 x 下拉配置状态
	0: 下拉无效 (默认)
	1: 下拉有效
	写:
	BSy: 端口 x 下拉配置位 (y=0…15)
	0: 对下拉配置位无操作
	1: 设置下拉配置位
注: PA14 在系统复位后自动配置为复用功能 SWCLK, SWCLK 引脚内部下拉。	

7.4.6 GPIO 端口输入数据寄存器 (GPIOx_IDR) (x=A,B,F)

地址偏移: 0x14

复位值: 0x0000_XXXX

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

Bits 31:16	保留, 必须保持复位值
Bits 15:0	读:
	IDRy: 端口输入数据位 (y=0…15), 与 GPIO 端口上数据的数据一一映射
	写:
	ODRy: 端口输出数据位 (y=0…15), 在 GPIO 位输出情况下, 寄存器数据出现在对应引脚, 与寄存器 GPIOx_ODR 功能一致

7.4.7 GPIO 端口输出数据寄存器 (GPIOx_ODR) (x=A,B,F)

地址偏移: 0x18

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

Bits 31:16	保留，必须保持复位值
Bits 15:0	ODRy : 端口输出数据位 ($y=0\cdots 15$), 在 GPIO 位输出情况下, 寄存器数据出现在对应引脚

7.4.8 GPIO 端口数据设置/清除寄存器 (GPIOx_BSRR) (x=A,B,F)

地址偏移: 0x1C

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
Res.															
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
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

Bits 31:16	读:
	保留
	写:
	BRy : 端口 x 输出数据位复位 ($y=0\cdots 15$)
	0: 对数据输出位无操作
	1: 数据输出位复位
	如果同时设置了 BRy 和 BSy 的对应位, BSy 位起作用。
Bits 15:0	读:
	ODRy : 端口 x 输出数据位对应的值
	写:
	BSy : 端口 x 输出数据配置位 ($y=0\cdots 15$)
	0: 对输出数据位无操作
	1: 设置输出数据位

7.4.9 GPIO 端口配置锁定寄存器 (GPIOx_LCKR) (x=A,B,F)

地址偏移: 0x20

复位值: 0x0000_0000

当执行正确的写序列设置了 Bit 16 (LCKK) 时, 该寄存器用来锁定端口位的配置。Bit[15:0]表示需要锁定的 GPIO 端口位。在规定序列的写入操作期间, 不能改变 LCKP[15:0]。当对应的端口位执行了 LOCK 序列后, 在下次系统复位之前将不能再更改端口位的配置。每个锁定位锁定控制寄存器的对应位。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:17	保留，必须保持复位值
Bit 16	LCKK : 锁键 (Lock Key)
	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 (默认)
	1 : 端口配置锁键位被激活，下次系统复位前 GPIOx_LCKR 寄存器被锁住
	锁键的写入序列:
	①WR LCKR[16:0]={1' b1, LCKR[15:0]}
	②WR LCKR[16:0]={1' b0, LCKR[15:0]}
	③WR LCKR[16:0]={1' b1, LCKR[15:0]}
	④RD LCKR
	⑤RD LCKR[16]=1 (该操作可以忽略，但可以用来确认锁键已被激活)
	<i>注: 在操作锁键的写入序列时，不能改变 LCK[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</i>
Bits 15:0	LCKy : 端口 x 的锁位 y (y=0...15)
	这些位可读写但只能在 LCKK 位为 0 时写入
	0 : 不锁定端口的配置
	1 : 锁定端口的配置

7.4.10 GPIO 端口复用功能配置寄存器低位 (GPIOx_AFRL) (x=A,B,F)

地址偏移: 0x24

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFR7[3:0]				AFR6[3:0]				AFR5[3:0]				AFR4[3:0]			
rw				rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR3[3:0]				AFR2[3:0]				AFR1[3:0]				AFR0[3:0]			
rw				rw											

Bits 31:0	ARFy[3:0] : 端口 x 的 y 位复用功能选择 (y=0...7)
	当端口对应位配置为数字功能复用时，ARFy 用于选择数字功能复用时:
	0000 : 复用功能 0 (AF0) (默认)
	0001 : 复用功能 1 (AF1)
	0010 : 复用功能 2 (AF2)

	0011: 复用功能 3 (AF3)
	0100: 复用功能 4 (AF4)
	0101: 复用功能 5 (AF5)
	0110: 复用功能 6 (AF6)
	0111: 复用功能 7 (AF7)
	当端口配置为模拟复用时, AFRy 用于模拟信号通道的选择:
	AFRy[0]: 模拟通道 1 使能
	0: 模拟通道 1 禁止
	1: 模拟通道 1 使能
	AFRy[1]: 模拟通道 2 使能
	0: 模拟通道 2 禁止
	1: 模拟通道 2 使能
	AFRy[3:2]: 保留

7.4.11 GPIO 端口复用功能配置寄存器高位 (GPIOx_AFRH) (x=A,B,F)

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
AFR15[3:0]				AFR14[3:0]				AFR13[3:0]				AFR12[3:0]			
rw				rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR11[3:0]				AFR10[3:0]				AFR9[3:0]				AFR8[3:0]			
rw				rw											

Bits 31:0	AFRy[3:0]: 端口 x 的 y 位复用功能选择 (y=8···15)
	当端口对应位配置为数字功能复用时, ARFy 用于选择数字功能复用时:
	0000: 复用功能 0 (AF0) (默认)
	0001: 复用功能 1 (AF1)
	0010: 复用功能 2 (AF2)
	0011: 复用功能 3 (AF3)
	0100: 复用功能 4 (AF4)
	0101: 复用功能 5 (AF5)
	0110: 复用功能 6 (AF6)
	0111: 复用功能 7 (AF7)
	当端口配置为模拟复用时, AFRy 用于模拟信号通道的选择:
	AFRy[0]: 模拟通道 1 使能
	0: 模拟通道 1 禁止
	1: 模拟通道 1 使能
	AFRy[1]: 模拟通道 2 使能
	0: 模拟通道 2 禁止
	1: 模拟通道 2 使能

	AFRy[3:2]: 保留
--	---------------

7.4.12 GPIO 端口输出数据翻转寄存器 (GPIOx_BTGLR) (x=A,B,F)

地址偏移: 0x2C

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BTGL15	BTGL14	BTGL13	BTGL12	BTGL11	BTGL10	BTGL9	BTGL8	BTGL7	BTGL6	BTGL5	BTGL4	BTGL3	BTGL2	BTGL1	BTGL0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16	保留, 必须保持复位值
Bits 15:0	读:
	ODRy : 端口 x 输出数据位对应的值 (y=0...15)
	写:
	BTGLy : 端口 x 输出数据翻转配置位 (y=0...15)
	0: 对输出数据翻转配置位无操作
	1: 设置输出数据翻转配置位

7.4.13 GPIO 端口驱动强度设置/清除寄存器 (GPIOx_DR_BSRR) (x=A,B,F)

地址偏移: 0x30

复位值: 0x0000_FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
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
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
DR15	DR14	DR13	DR12	DR11	DR10	DR9	DR8	DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16	读:
	保留
	写:
	BRy : 端口 x 驱动强度配置复位 (y=0...15)
	0: 对驱动强度配置位无操作
	1: 复位驱动强度配置位
Bits 15:0	读:
	DRy : 端口 x 驱动强度配置状态

	0: 高驱动模式
	1: 低驱动模式 (默认)
	写:
	BSy : 端口 x 驱动强度配置位 (y=0...15)
	0: 对驱动强度配置位无操作
	1: 设置驱动强度配置位

7.4.14 GPIO 端口施密特触发器输入设置/清除寄存器(GPIOx_CS_BSRR)(x=A,B,F)

地址偏移: 0x34

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
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
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
CS15	CS14	CS13	CS12	CS11	CS10	CS9	CS8	CS7	CS6	CS5	CS4	CS3	CS2	CS1	CS0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16	读:
	保留
	写:
	BRy : 端口 x CMOS/施密特触发器输入配置复位 (y=0...15)
	0: 对 CMOS/施密特触发器输入配置位无操作
	1: 复位 CMOS/施密特触发器输入配置位
Bits 15:0	读:
	CSy : 端口 x CMOS/施密特触发器输入配置状态
	0: 施密特触发器输入 (默认)
	1: CMOS 输入
	写:
	BSy : 端口 x CMOS/施密特触发器输入配置位 (y=0...15)
	0: 对 CMOS/施密特触发器输入配置位无操作
	1: 设置 CMOS/施密特触发器输入配置位

7.4.15 GPIO 端口开源模式设置/清除寄存器 (GPIOx_OS_BSRR) (x=A,B,F)

地址偏移: 0x38

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
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
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
OS15	OS14	OS13	OS12	OS11	OS10	OS9	OS8	OS7	OS6	OS5	OS4	OS3	OS2	OS1	OS0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16	读:
	保留
	写:
	BRy : 端口 x 开源配置复位 (y=0···15)
	0: 对开源配置位无操作
	1: 复位开源配置位
Bits 15:0	读:
	OSy : 端口 x 开源配置状态
	0: 开源输出禁止 (默认)
	1: 开源输出使能
	写:
	BSy : 端口 x 开源配置位 (y=0···15)
	0: 对开源配置位无操作
	1: 设置开源配置位

8. DMA 控制器

直接存储器访问（DMA）控制器是一个总线主机和系统外围设备。DMA 控制器采用单一的 AHB 主体系统结构。DMA 被用来在脱离 CPU 的控制下进行内存映射的外设和/或内存之间的可编程数据传输。

DMA 通道专用于管理来自一个或多个外设的内存访问请求。每个 DMA 通道都包括一个仲裁器，用于处理 DMA 请求之间的优先级。每个通道的传输长度、传输源地址和目标地址都可以通过软件独立设置。

DMA 可以接收下列外设的硬件请求：SSP、I2C、UART、ADC、DAC 和 TIMER。

8.1 主要特性

- 单 AHB master 接口。可以作为源和目标访问片上存储，包括 SRAM、AHB 和 APB 线上的外设
- AHB slave 接口，用来编程配置 DMA
- 4 个独立可配置的通道，每个通道都包含源和目标对象
- 每个通道优先级可编程（3 为最高优先级，0 为最低优先级）
- 每个通道都直接连接专用的硬件和软件 DMA 请求
- 每个通道都可以自动数据打包和拆包
- 每个通道可独立使能与关闭，不会造成数据丢失
- 每个通道都有 5 个事件标志（传输完成、块传输完成、源传输完成、目标传输完成、传输出错）
- 四种传输模式：存储器和存储器间、外设和存储器间、存储器和外设间、外设和外设间的传输

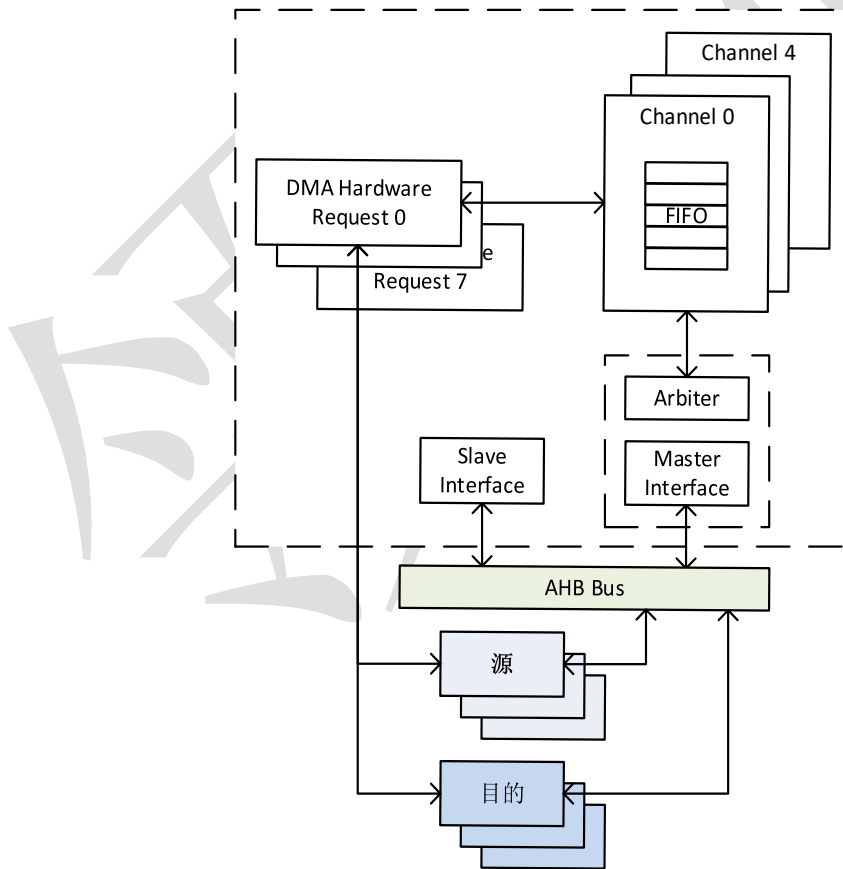


图 8-1 DMA 结构框图

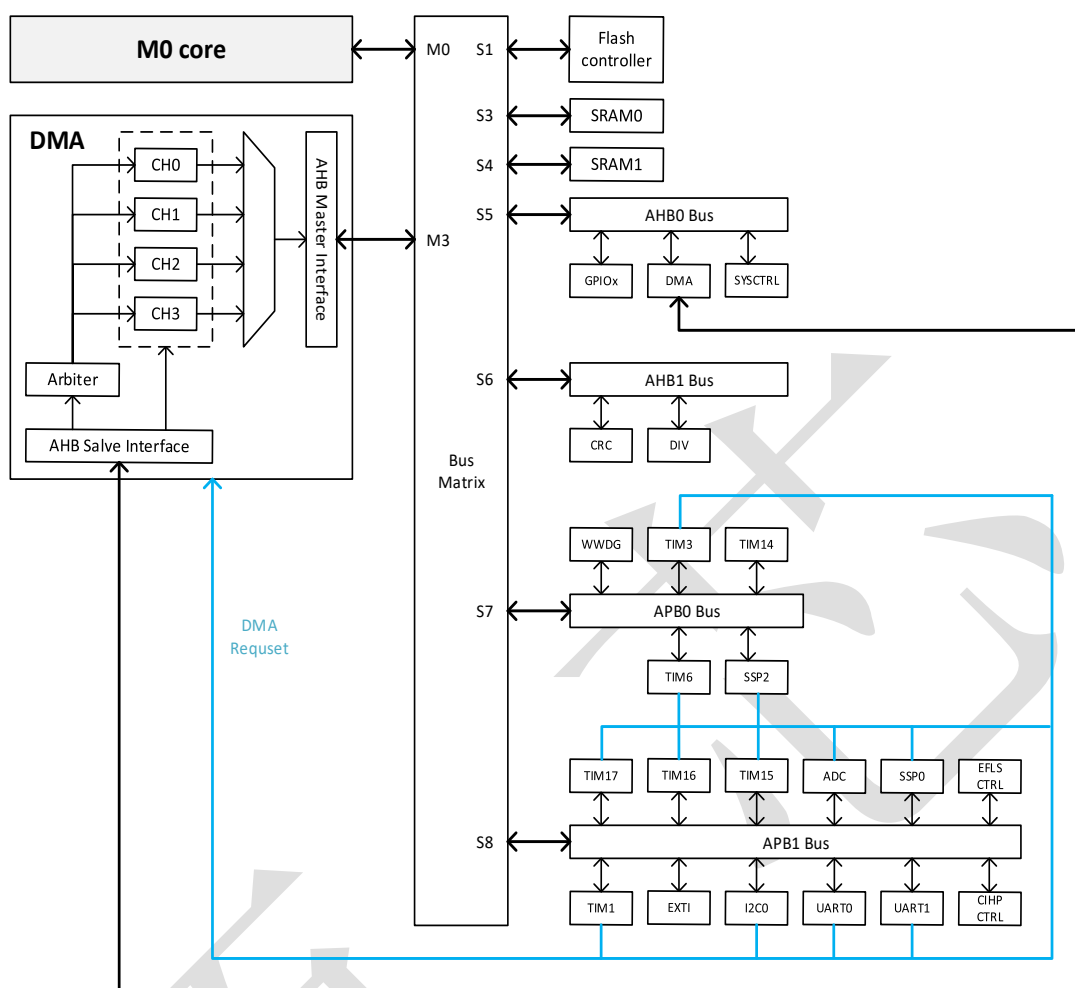


图 8-2 DMA 连接框图

8.2 DMA 功能描述

DMA 控制器和 CPU 核共享系统数据总线执行直接存储器数据传输。当 CPU 和 DMA 同时访问相同的目标（RAM 或外设）时，DMA 请求可能会停止 CPU 访问系统总线达若干个周期，总线仲裁器执行循环调度，以保证 CPU 至少可以得到一半的系统总线（存储器或外设）带宽。

8.2.1 DMA 处理

在发生一个事件后，外设发送一个请求信号到 DMA 控制器。DMA 控制器根据通道的优先级处理请求。当 DMA 控制器完成传输任务后，立即发送给外设一个应答信号。当从 DMA 控制器得到应答信号时，外设立即释放它的请求。一旦外设释放了这个请求，DMA 控制器同时撤销应答信号。如果发生更多的请求时，外设可以启动下次处理。

总之，每次 DMA 传送由 3 个操作组成：

1. 从外设数据寄存器或者从当前外设/存储器地址寄存器，指示的存储器地址取数据，第一次传输时的开始地址是 SARx 寄存器指定的外设基地址或存储器单元。

2. 存数据到外设数据寄存器或者当前外设/存储器地址寄存器指示的存储器地址，第一次传输时的开始地址是 DARx 寄存器指定的外设基地址或存储器单元。
3. 循环以上两步，直到传输结束，产生完成中断信号。

8.2.2 DMA 仲裁

源外设和目标外设的 DMA 请求，通过仲裁器决定哪个请求获得总线访问权，仲裁器根据通道请求的优先级来启动外设/存储器的访问。优先级管理分 2 个阶段：

- 软件：每个通道的优先级可以通过设置通道配置寄存器 CFGx 的 CH_PRIOR 位来设置，有 4 个等级：
 - 最高优先级为 3
 - 最低优先级为 0
- 硬件：如果 2 个请求有相同的软件优先级，则拥有较低编号的通道比拥有较高编号的通道有较高的优先级。例如，通道 2 和通道 3 的软件优先级一样，同时请求的时候，则通道 2 优先于通道 3。

8.2.3 DMA 中断

DMA 的每个通道在传输过程中可以产生以下 5 种中断。每个通道的每个中断支持独立屏蔽。

- 传输中断
- 块传输中断
- 源传输中断
- 目标传输中断
- 错误中断

中断逻辑框图如下所示：

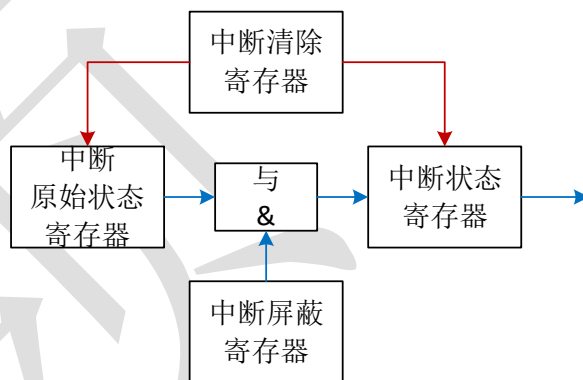


图 8-3 DMA 中断逻辑

8.2.4 DMA 请求与数据传输

DMA 的数据传输过程分为突发事务传输和单次传输两种模式，UART、SSP、I2C 通讯外设支持两种模式请求，TIMER 和 ADC 外设仅支持突发事务传输请求。

突发事务传输：DMA 接收到外设的突发事务传输请求后，进行一次突发事务传输数据数量的搬运。在 block 传输的最后一次突发事务传输时，仅搬运剩余数量的数据。

单次传输：DMA 接收到外设的单次传输请求以及无突发事务传输请求，且为 block 传输中最后一次传输的剩余数量不满足一次突发事务传输数据数量时，则进入单次传输模式。单次传输模式可以将外设 FIFO

内不满一次突发事务传输数据量的数据一个一个搬出或搬入。

下面举例 DMA 和外设的请求与数据传输关系：

DMA 配置位	外设 A 配置值 1	外设 A 配置值 2	外设 B 配置值 1	外设 B 配置值 2
CTLx.TT_FC	001	010	001	010
CTLx.BLOCK_TS	8	10	6	10
CTLx.SRC_TR_WIDTH	32 bits	16 bits	32 bits	32 bits
CTLx.DST_TR_WIDTH	32 bits	32 bits	32 bits <td 32 bits	
CTLx.SRC_MSIZ	001 (长度=4)	001 (长度=4)	001 (长度=4)	001 (长度=4)
CTLx.DEST_MSIZ	001 (长度=4)	001 (长度=4)	001 (长度=4)	001 (长度=4)

注：下图外设 A 代指 TIMER 或 ADC，外设 B 代指 UART、SSP 或 I2C。

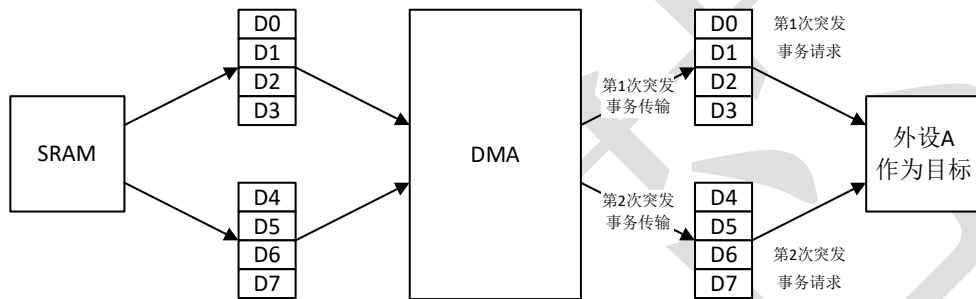


图 8-4 外设 A 配置 1 且作为目标，2 次源的传输，2 次目标的突发事务请求和传输

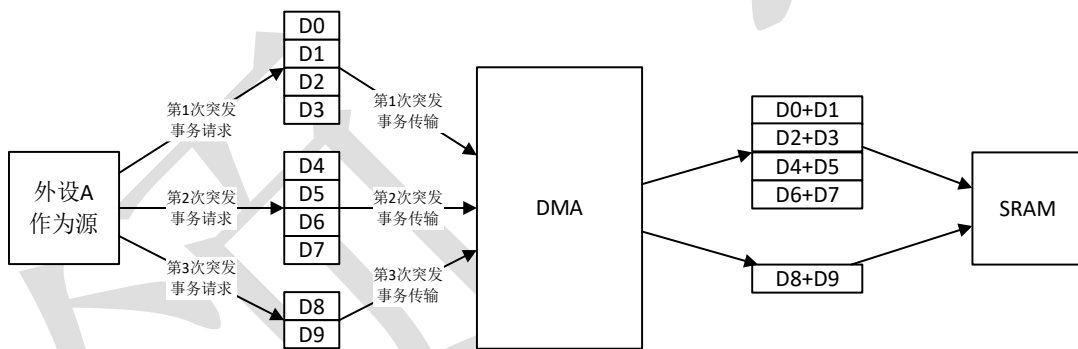


图 8-5 外设 A 配置 2 且作为源，3 次源的突发事务请求和传输，2 次目标的传输

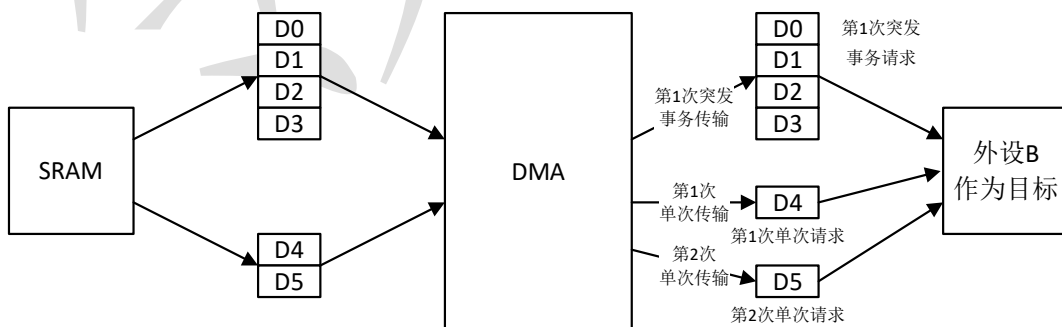


图 8-6 外设 B 配置 1 且作为目标，2 次源的传输，1 次目标的突发事务请求和传输，2 次目标的单次请求和传输

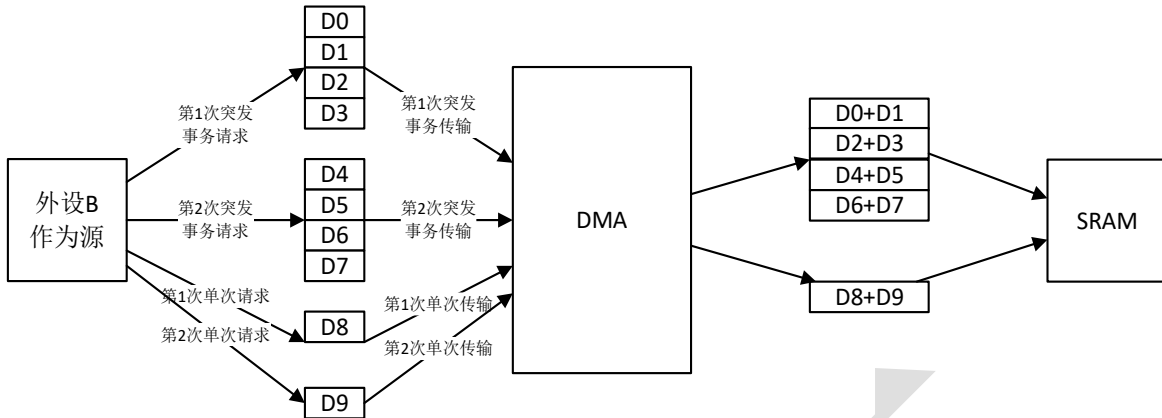


图 8-7 外设 B 配置 2 且作为源，2 次源的突发事务请求和传输，2 次源的单次请求和传输，2 次目标的传输

如上图所示，外设 A 不支持单次请求，外设 B 支持单次请求。如果最后一次传输不满足突发事务传输数据数量，则会有上图传输区别。判断外设 B 是否会进入单次请求和传输可以有以下公式：

1. 外设 B 作为源时：

$$(\text{CTLx.BLOCK_TS}) / (\text{CTLx.SRC_MSIZE}) = n$$

2. 外设 B 作为目标时：

$$(\text{CTLx.SRC_TR_WIDTH} * \text{CTLx.BLOCK_TS}) / (\text{CTLx.DST_TR_WIDTH} * \text{CTLx.DST_MSIZE}) = n$$

当 n 不为整数时，表明最后一次突发事务传输数量不满足配置值。此时，外设 A 依然发出突发事务请求，将最后剩余数据一次性搬运；而外设 B 则会发出单次请求将剩余数据一个一个搬运完。

8.3 DMA 寄存器描述

表 8-1 DMA 相关寄存器表

名称	说明	读写权限	复位值	字节地址
通道 0 寄存器				
SAR0	源地址寄存器 0	R/W	0x0000_0000	0x4800_4000
DAR0	目标地址寄存器 0	R/W	0x0000_0000	0x4800_4008
CTL0	控制寄存器 0	R/W	0x2_0030_4801	0x4800_4018
CFG0	配置寄存器 0	R/W	0x4_0000_0E00	0x4800_4040
通道 1 寄存器				
SAR1	源地址寄存器 1	R/W	0x0000_0000	0x4800_4058
DAR1	目标地址寄存器 1	R/W	0x0000_0000	0x4800_4060
CTL1	控制寄存器 1	R/W	0x2_0030_4801	0x4800_4070
CFG1	配置寄存器 1	R/W	0x4_0000_0E20	0x4800_4098
通道 2 寄存器				
SAR2	源地址寄存器 2	R/W	0x0000_0000	0x4800_40B0
DAR2	目标地址寄存器 2	R/W	0x0000_0000	0x4800_40B8
CTL2	控制寄存器 2	R/W	0x2_0030_4801	0x4800_40C8
CFG2	配置寄存器 2	R/W	0x4_0000_0E40	0x4800_40F0
通道 3 寄存器				
SAR3	源地址寄存器 3	R/W	0x0000_0000	0x4800_4108
DAR3	目标地址寄存器 3	R/W	0x0000_0000	0x4800_4110



CTL3	控制寄存器 3	R/W	0x2_0030_4801	0x4800_4120
CFG3	配置寄存器 3	R/W	0x4_0000_0E60	0x4800_4148
中断寄存器				
中断原始状态寄存器				
RawTfr	传输中断原始状态	R/W	0x0000_0000	0x4800_42C0
RawBlock	块传输中断原始状态	R/W	0x0000_0000	0x4800_42C8
RawSrcTran	源传输中断原始状态	R/W	0x0000_0000	0x4800_42D0
RawDstTran	目标传输中断原始状态	R/W	0x0000_0000	0x4800_42D8
RawErr	传输错误中断原始状态	R/W	0x0000_0000	0x4800_42E0
中断状态寄存器				
StatusTfr	传输中断状态	R/W	0x0000_0000	0x4800_42E8
StatusBlock	块传输中断状态	R/W	0x0000_0000	0x4800_42F0
StatusSrcTran	源传输中断状态	R/W	0x0000_0000	0x4800_42F8
StatusDstTran	目标传输中断状态	R/W	0x0000_0000	0x4800_4300
StatusErr	传输错误中断状态	R/W	0x0000_0000	0x4800_4308
中断屏蔽寄存器				
MaskTfr	传输中断屏蔽	R/W	0x0000_0000	0x4800_4310
MaskBlock	块传输中断屏蔽	R/W	0x0000_0000	0x4800_4318
MaskSrcTran	源传输中断屏蔽	R/W	0x0000_0000	0x4800_4320
MaskDstTran	目标传输中断屏蔽	R/W	0x0000_0000	0x4800_4328
MaskErr	传输错误中断屏蔽	R/W	0x0000_0000	0x4800_4330
中断清除寄存器				
ClearTfr	传输中断清除	W	0x0000_0000	0x4800_4338
ClearBlock	块传输中断清除	W	0x0000_0000	0x4800_4340
ClearSrcTran	源传输中断清除	W	0x0000_0000	0x4800_4348
ClearDstTran	目标传输中断清除	W	0x0000_0000	0x4800_4350
ClearErr	传输错误中断清除	W	0x0000_0000	0x4800_4358
组合中断状态寄存器				
StatusInt	组合中断状态	R	0x0000_0000	0x4800_4360
软件握手寄存器				
ReqSrcReg	源传输软件请求	R/W	0x0000_0000	0x4800_4368
ReqDstReg	目标传输软件请求	R/W	0x0000_0000	0x4800_4370
SglReqSrcReg	源单次传输软件请求	R/W	0x0000_0000	0x4800_4378
SglReqDstReg	目标单次传输软件请求	R/W	0x0000_0000	0x4800_4380
LstSrcReg	源最后传输软件请求	R/W	0x0000_0000	0x4800_4388
LstDstReg	目标最后传输软件请求	R/W	0x0000_0000	0x4800_4390
全局配置和使能寄存器				
DmaCfgReg	DMA 配置寄存器	R/W	0x0000_0000	0x4800_4398
ChEnReg	通道使能寄存器	R/W	0x0000_0000	0x4800_43A0

注: x 表示不确定; - 表示未实现; u 表示不受复位影响; c 表示该位复位值因复位源而异; R 只读; W 只写; R-0 表示只读且读 0; R/W 表示可读写 (以后章节同上述)。



8.3.1 全局配置和使能寄存器

8.3.1.1 DMA 配置寄存器 (DmaCfgReg)

地址偏移: 0x398

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.															DMA_
															EN
															rw

Bits	31:1	保留, 必须保持复位值
Bit	0	DMA_EN : DMA 使能
		0: DMA 禁止 (默认)
		1: DMA 使能
		注1 : 全局使能信号, 使能任何 DMA 通道之前必须使能 DMA。
		注2 : 如果全局通道使能位被清除, 而任何通道仍然活跃, 那么 DmaCfgReg.DMA_EN 仍然为1, 表明有一些通道仍然活跃, 直到硬件终止所有通道的所有活动, 然后 DmaCfgReg.DMA_EN 才被清除为0。

8.3.1.2 DMA 通道使能寄存器 (ChEnReg)

地址偏移: 0x3A0

复位值: 0x0000_0000

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

Bits	31:12	保留, 必须保持复位值
Bits	11:8	CH_EN_WE[3:0] : 通道使能的写使能 (每位对应一个通道)
		0: 通道使能不可写 (默认)
		1: 通道使能可写
Bits	7:4	保留, 必须保持复位值
Bits	3:0	CH_EN[3:0] : DMA 通道使能位
		0: 通道禁止 (默认)
		1: 通道使能
		注 : 当 DMA 使能位为0 时, 所有这些位都为0。

8.3.2 通道寄存器

每个 DMA 通道包含以下的通道寄存器 (x = 0~3):

- SARx – 通道 x 的源地址寄存器
- DARx – 通道 x 的目标地址寄存器
- CTLx – 通道 x 的控制寄存器
- CFGx – 通道 x 的配置寄存器

8.3.2.1 DMA 源地址寄存器 (DMA_SARx (x=0/1/2/3))

地址偏移: 0x00/0x58/0xB0/0x108

复位值: 0x0000_0000

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

Bits	31:0	SARx[31:0]: DMA 传输的当前源地址
		在每次源传输后更新, CTLx.SINC 位决定在整个块传输的每个源传输中地址是递增、递减还是保持不变。

8.3.2.2 DMA 目标地址寄存器 (DMA_DARx (x=0/1/2/3))

地址偏移: 0x08/0x60/0xB8/0x110

复位值: 0x0000_0000

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

Bits	31:0	DARx[31:0]: DMA 传输的当前目标地址
		在每次目标传输后更新, CTLx.DINC 位决定在整个块传输的每个目标传输中地址是递增、递减还是保持不变。

8.3.2.3 DMA 通道控制寄存器 (DMA_CTLx (x=0/1/2/3))

地址偏移: 0x18/0x70/0xC8/0x120

复位值: 0x2_0030_4801

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
Res.															

47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Res.				BLOCK_TS[11:0]											
rw															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.									TT_FC[2:0]			Res.			SRC_M SIZE[2: 0]
									rw						rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRC_MSIZ[2:0]		DEST_MSIZ[2:0]			SINC[1:0]		DINC[1:0]		SRC_TR_WIDTH[2:0]			DST_TR_WIDTH[2:0]			INT_EN
rw		rw			rw		rw		rw			rw			rw

Bits 63:44	保留，必须保持复位值				
Bits 43:32	BLOCK_TS[11:0]: 块传输大小（通道 0 可配置有效位为[7:0]，通道 1 可配置有效位为[6:0]，通道 2/3 可配置有效位为[5:0]）				
	编程到 BLOCK_TS 中的数字表示为每个块传输执行的单个事务的总数				
	单个事务的数据位宽由 CTLx.SRC_TR_WIDTH 决定。				
	一旦传输开始，回读值就是已经从源外设读取的数据总数。				
Bits 31:23	保留，必须保持复位值				
Bits 22:20	TT_FC[2:0]: 传输类型和流控制（复位值：3'b011），DMA 有四种传输类型				
	流控制器固定为 DMA 控制器，源外设或目标外设不能作为流控制器。				
	CTLx.TT_FC	传输类型		流控制器	
	000	存储到存储		DMA 控制器	
	001	存储到外设		DMA 控制器	
	010	外设到存储		DMA 控制器	
	011	外设到外设		DMA 控制器	
Bits 19:17	保留，必须保持复位值				
Bits 16:14	SRC_MSIZ[2:0]: 源突发事务传输长度（同 DEST_MSIZ 位）				
Bits 13:11	DEST_MSIZ[2:0]: 目标突发事务传输长度				
	每个数据项的数量宽度 CTLx.DEST_TR_WIDTH (SRC_TR_WIDTH)，每次从目标（源）读取的目标（源）突发事务请求对应的硬件或软件握手界面。				
	CTLx.SRC_MSIZ/ CTLx.DEST_MSIZ	000	001	010	其他
	事务传输长度	1	4	8	保留
Bits 10:9	SINC[1:0]: 源地址递增位（同 DINC 位）				
Bits 8:7	DINC[1:0]: 目标地址递增位				
	指示在每次目的（源）传输中增加或减少目的地址。				
	如果写入或读取的数据是外设 FIFO 的固定地址，则需要设置为“不变”。				

	00: 递增（默认）				
	01: 递减				
	1x: 不变				
Bits 6:4	SRC_TR_WIDTH[2:0] : 源传输位宽（同 DST_TR_WIDTH 位）				
Bits 3:1	DST_TR_WIDTH[2:0] : 目标传输位宽				
	对非存储外设来说，通常就等于其 FIFO 的位宽。				
	也可以简单的配置为同 AHB、APB 总线的 32 位宽。				
	SRC_TR_WIDTH / DST_TR_WIDTH	000	001	010	其他
	位宽	8	16	32	保留
Bit 0	INT_EN : 通道中断使能位				
	0: 中断禁止				
	1: 中断使能（默认）				

8.3.2.4 DMA 通道配置寄存器（DMA_CFGx (x=0/1/2/3)）

地址偏移：0x40/0x98/0xF0/0x148

复位值：0x4_0000_0e00/0x4_0000_0e20/0x4_0000_0e40/0x4_0000_0e60

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
Res.															

47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Res.		DEST_PER[2:0]			Res.		SRC_PER[2:0]			Res.				FIFO_MODE	Res.
		rw					rw							rw	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RELOAD_D ST	RELOAD_ SRC	Res.										SRC_HS_P OL	DST_HS_P OL	Res.	
Rw	rw											rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				HS_SEL_S RC	HS_SEL_D ST	FIFO_EMP TY	CH_S USP	CH_PRIOR[2:0]			Res.				
				rw	rw	rw	rw	rw							

Bits 63:46	保留，必须保持复位值
Bits 45:43	DEST_PER[2:0] : 目标外设硬件握手接口 0-7（同 SRC_PER 位）
Bit 42	保留，必须保持复位值
Bits 41:39	SRC_PER[2:0] : 源外设硬件握手接口 0-7
	如果 CFGx 为通道 x 的目的地分配一个硬件握手接口（0 - 7），则 HS_SEL_DST 字段为 0；否则该字段将被忽略。然后，通道可以通过指定的硬件握手接口与连接到该接口的目标（源）外设进行通信。
	注： 对于正确的 DMA 操作，同一个握手接口只能分配一个外设（源或目标）。

Bits 38:34	保留，必须保持复位值
Bit 33	FIFO_MODE: FIFO 模式选择
	确定在处理突发事务请求之前，FIFO 中需要有多少空间或数据可用。
	0: 指定传输位宽的单个 AHB 传输可用的空间/数据。(默认)
	1: 可用数据大于或等于用于目标传输的 FIFO 深度的一半，可用空间大于用于源传输的 FIFO 深度的一半。异常发生在突发事务请求或块传输结束时。
Bit 32	保留，必须保持复位值
Bit 31	RELOAD_DST: 自动目标重载 (同 RELOAD_SRC 位)
Bit 30	RELOAD_SRC: 自动源重载
	对于多个块传输，DARx (SARx) 可以在每个块的末尾处自动重新加载它的初始值，然后开始一个新的块传输。
Bits 29:20	保留，必须保持复位值
Bit 19	SRC_HS_POL: 源外设握手接口极性 (同 DST_HS_POL 位)
Bit 18	DST_HS_POL: 目标外设握手接口极性
	0: 高有效 (默认)
	1: 低有效
Bits 17:12	保留，必须保持复位值
Bit 11	HS_SEL_SRC: 源软件或硬件握手选择 (同 HS_SEL_DST 位)
Bit 10	HS_SEL_DST: 目标外设软件或硬件握手接口选择
	选择某个握手接口 (硬件或软件) 对这个通道上的目标 (源) 请求是活跃的。
	0: 硬件握手接口。忽略软件发起的突发事务传输请求
	1: 软件握手接口，忽略硬件发起的突发事务传输请求 (默认)
	如果目标 (源) 外设是 memory，则该 bit 被忽略。
Bit 9	FIFO_EMPTY: 指示是否通道 FIFO 中是否还有数据
	可以与 CFGx.CH_SUSP 结合使用来安全的关闭一个通道。
	0: 通道 FIFO 非空
	1: 通道 FIFO 空 (默认)
Bit 8	CH_SUSP: 通道暂停
	暂停来自源的所有 DMA 数据传输，直到清除该位。不能保证当前事务将完成。也可以与 CFGx.FIFO_EMPTY 一起使用来清除禁用通道而不丢失任何数据。
	0: 通道没有暂停 (默认)
	1: 暂停从源外设来的 DMA 传输
Bits 7:5	CH_PRIOR[2:0]: 通道优先级
	3 为最高优先级，0 是最低优先级 (如果配置超出范围，则会产生错误)。
	复位值: 通道数 (例如: 通道 0 = 0, 通道 1 = 1)。
Bits 4:0	保留，必须保持复位值

8.3.3 中断寄存器

下面的部分描述了与中断有关的寄存器、状态以及如何清除它们。对于每个通道，有五种类型的中断源：

- IntBlock – Block Transfer Complete Interrupt。这个中断是在向目标外设完成 DMA 块传输时产生的。
- IntDstTran – Destination Transaction Complete Interrupt。这个中断是在目标端的握手接口 (硬件或软件握手接口) 完成请求的单个/突发事务的最后一个 AHB 传输之后产生的。
- IntErr – Error Interrupt。当在 DMA 传输期间从 HRESP 总线上的 AHB slave 接收到一个 ERROR 响应

时，产生此中断。此外，DMA 传输被取消，通道被禁用。

- **SrcTran – Source Transaction Complete Interrupt**。这个中断是在源端握手接口（硬件或软件握手接口）完成请求的单个/突发事务的最后一个 AHB 传输之后产生的。
- **IntTfr – DMA Transfer Complete Interrupt**。这个中断是在向目标外设的 DMA 传输完成时产生的。

有几组与中断相关的寄存器：

- RawBlock、RawDstTran、RawErr、RawSrcTran、RawTfr
- StatusBlock、StatusDstTran、StatusErr、StatusSrcTran、StatusTfr
- MaskBlock、MaskDstTran、MaskErr、MaskSrcTran、MaskTfr
- ClearBlock、ClearDstTran、ClearErr、ClearSrcTran、ClearTfr
- StatusInt

当一个通道被使能来生成中断时，则：

- 中断事件存储在原始状态寄存器中。
- 原始状态寄存器的内容被掩码寄存器的内容屏蔽。
- 被屏蔽的中断存储在状态寄存器中。
- 状态寄存器的内容用于驱动 int_* 端口信号。
- 写入到清除寄存器的适当位清除在相同时钟周期的原始状态寄存器和状态寄存器中的中断。

五个状态寄存器中的每一位“或”成一位标志位，对应组合状态寄存器（StatusInt）里的相应一种中断类型。

注1：如果通道的目标是 SRAM，那么该通道将永远不会生成 IntDstTran 中断。因此，对应的位将不会被设置。

注2：如果源或目标是 SRAM，那么 IntSrcTran/IntDstTran 中断应该被忽略，因为内存中没有“DMA 事务级别”的概念。

注3：对于经过 Raw* 中断寄存器阶段的中断，CTLx.INT_EN 必须设置为 1'b1，并且相关的中断必须在 Mask* 中断寄存器中被解除屏蔽。

8.3.3.1 中断原始状态寄存器组

8.3.3.1.1 传输中断原始状态寄存器（DMA_RawTfr）

地址偏移：0x2C0

复位值：0x0000_0000

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

Bits	31:4	保留，必须保持复位值
Bits	3:0	RawTr[3:0] ：传输原始中断状态

8.3.3.1.2 块传输中断原始状态寄存器（DMA_RawBlock）

地址偏移：0x2C8

复位值：0x0000_0000

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

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

Bits 31:4	保留，必须保持复位值
Bits 3:0	RawBlock[3:0] : 块传输原始中断状态

8.3.3.1.3 源传输中断原始状态寄存器 (DMA_RawSrcTran)

地址偏移: 0x2D0

复位值: 0x0000_0000

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

Bits 31:4	保留，必须保持复位值
Bits 3:0	RawSrcTran[3:0] : 源传输原始中断状态

8.3.3.1.4 目标传输中断原始状态寄存器 (DMA_RawDstTran)

地址偏移: 0x2D8

复位值: 0x0000_0000

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

Bits 31:4	保留，必须保持复位值
Bits 3:0	RawDstTran[3:0] : 目标传输原始中断状态

8.3.3.1.5 传输错误中断原始状态寄存器 (DMA_RawErr)

地址偏移: 0x2E0

复位值: 0x0000_0000

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

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

Bits	31:4	保留，必须保持复位值
Bits	3:0	RawErr[3:0] : 错误原始中断状态

8.3.3.2 中断状态寄存器组

8.3.3.2.1 传输中断状态寄存器 (DMA_StatusTfr)

地址偏移: 0x2E8

复位值: 0x0000_0000

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

Bits	31:4	保留，必须保持复位值
Bits	3:0	StatusTfr[3:0] : 传输中断状态

8.3.3.2.2 块传输中断状态寄存器 (DMA_StatusBlock)

地址偏移: 0x2F0

复位值: 0x0000_0000

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

Bits	31:4	保留，必须保持复位值
Bits	3:0	StatusBlock[3:0] : 块传输中断状态

8.3.3.2.3 源传输中断状态寄存器 (DMA_StatusSrcTran)

地址偏移: 0x2F8

复位值: 0x0000_0000

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

Res.															

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

Bits 31:4	保留，必须保持复位值
Bits 3:0	StatusSrcTran[3:0] : 源传输中断状态

8.3.3.2.4 目标传输中断状态寄存器 (DMA_StatusDstTran)

地址偏移: 0x300

复位值: 0x0000_0000

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

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

Bits 31:4	保留，必须保持复位值
Bits 3:0	StatusDstTran[3:0] : 目标传输中断状态

8.3.3.2.5 传输错误中断状态寄存器 (DMA_StatusErr)

地址偏移: 0x308

复位值: 0x0000_0000

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

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

Bits 31:4	保留，必须保持复位值
Bits 3:0	StatusErr[3:0] : 错误中断状态

8.3.3.3 中断屏蔽寄存器组

8.3.3.3.1 传输中断屏蔽寄存器 (DMA_MaskTfr)

地址偏移: 0x310

复位值: 0x0000_0000

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

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

Bits	31:12	保留，必须保持复位值
Bits	11:8	INT_MASK_TFR_WE[3:0] : 传输中断屏蔽写使能
		0: 写禁止（默认）
		1: 写使能
Bits	7:4	保留，必须保持复位值
Bits	3:0	INT_MASK_TFR[3:0] : 传输中断屏蔽
		0: 屏蔽（默认）
		1: 非屏蔽

8.3.3.3.2 块传输中断屏蔽寄存器（DMA_MaskBlock）

地址偏移：0x318

复位值：0x0000_0000

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

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

Bits	31:12	保留，必须保持复位值
Bits	11:8	INT_MASK_BLOCK_WE[3:0] : 块传输中断屏蔽写使能
		0: 写禁止（默认）
		1: 写使能
Bits	7:4	保留，必须保持复位值
Bits	3:0	INT_MASK_BLOCK[3:0] : 块传输中断屏蔽
		0: 屏蔽（默认）
		1: 非屏蔽

8.3.3.3.3 源传输中断屏蔽寄存器（DMA_MaskSrcTran）

地址偏移：0x320

复位值：0x0000_0000

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

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

Bits	31:12	保留，必须保持复位值
Bits	11:8	INT_MASK_SRC_TRAN_WE[3:0] : 源传输中断屏蔽写使能
		0: 写禁止（默认）
		1: 写使能
Bits	7:4	保留，必须保持复位值
Bits	3:0	INT_MASK_SRC_TRAN[3:0] : 源传输中断屏蔽
		0: 屏蔽（默认）
		1: 非屏蔽

8.3.3.3.4 目标传输中断屏蔽寄存器 (DMA_MaskDstTran)

地址偏移: 0x328

复位值: 0x0000_0000

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

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

Bits	31:12	保留，必须保持复位值
Bits	11:8	INT_MASK_DST_TRAN_WE[3:0] : 目标传输中断屏蔽写使能
		0: 写禁止（默认）
		1: 写使能
Bits	7:4	保留，必须保持复位值
Bits	3:0	INT_MASK_DST_TRAN[3:0] : 目标传输中断屏蔽
		0: 屏蔽（默认）
		1: 非屏蔽

8.3.3.3.5 传输错误中断屏蔽寄存器 (DMA_MaskErr)

地址偏移: 0x330

复位值: 0x0000_0000

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

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

				rw								rw			
--	--	--	--	----	--	--	--	--	--	--	--	----	--	--	--

Bits	31:12	保留, 必须保持复位值
Bits	11:8	INT_MASK_Err_WE[3:0]: 错误中断屏蔽写使能
		0: 写禁止 (默认)
		1: 写使能
Bits	7:4	保留, 必须保持复位值
Bits	3:0	INT_MASK_Err[3:0]: 错误中断屏蔽
		0: 屏蔽 (默认)
		1: 非屏蔽

8.3.3.4 中断清除寄存器组

8.3.3.4.1 传输中断清除寄存器 (DMA_ClearTfr)

地址偏移: 0x338

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.												ClearTfr[3:0]			
w															

Bits	31:4	保留, 必须保持复位值
Bits	3:0	ClearTfr[3:0]: 传输中断清除
		0: 无效 (默认)
		1: 清除中断

8.3.3.4.2 块传输中断清除寄存器 (DMA_ClearBlock)

地址偏移: 0x340

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.												ClearBlock[3:0]			
w															

Bits	31:4	保留, 必须保持复位值
Bits	3:0	ClearBlock[3:0]: 块传输中断清除
		0: 无效 (默认)

	1: 清除中断
--	---------

8.3.3.4.3 源传输中断清除寄存器 (DMA_ClearSrcTran)

地址偏移: 0x348

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.												ClearSrcTran[3:0]			
												w			

Bits	31:4	保留, 必须保持复位值
Bits	3:0	ClearSrcTran[3:0]: 源传输中断清除
		0: 无效 (默认)
		1: 清除中断

8.3.3.4.4 目标传输中断清除寄存器 (DMA_ClearDstTran)

地址偏移: 0x350

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.												ClearDstTran[3:0]			
												w			

Bits	31:4	保留, 必须保持复位值
Bits	3:0	ClearDstTran[3:0]: 目标传输中断清除
		0: 无效 (默认)
		1: 清除中断

8.3.3.4.5 传输错误中断清除寄存器 (DMA_ClearErr)

地址偏移: 0x358

复位值: 0x0000_0000

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

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

Res.															

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

Bits 31:12	保留，必须保持复位值
Bits 11:8	SRC_REQ_WE[3:0] : 源传输请求写使能
	0: 写禁止（默认）
	1: 写使能
Bits 7:4	保留，必须保持复位值
Bits 3:0	SRC_REQ[3:0] : 源传输请求
	0: 无操作（默认）
	1: 发出源传输请求

8.3.3.6.2 目标传输软件请求寄存器（DMA_ReqDstReg）

地址偏移：0x370

复位值：0x0000_0000

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

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

Bits 31:12	保留，必须保持复位值
Bits 11:8	DST_REQ_WE[3:0] : 目标传输请求写使能
	0: 写禁止（默认）
	1: 写使能
Bits 7:4	保留，必须保持复位值
Bits 3:0	DST_REQ[3:0] : 目标传输请求
	0: 无操作（默认）
	1: 发出目标传输请求

8.3.3.6.3 源单次传输软件请求寄存器（DMA_SglReqSrcReg）

地址偏移：0x378

复位值：0x0000_0000

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

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

Bits 31:12	保留，必须保持复位值
Bits 11:8	SRC_SGLREQ_WE[3:0] : 源单次传输请求写使能
	0: 写禁止（默认）
	1: 写使能
Bits 7:4	保留，必须保持复位值
Bits 3:0	SRC_SGLREQ[3:0] : 源单次传输请求
	0: 无操作（默认）
	1: 发出源单次传输请求

8.3.3.6.4 目标单次传输软件请求寄存器（DMA_SglReqDstReg）

地址偏移：0x380

复位值：0x0000_0000

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

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

Bits 31:12	保留，必须保持复位值
Bits 11:8	DST_SGLREQ_WE[3:0] : 目标单次传输请求写使能
	0: 写禁止（默认）
	1: 写使能
Bits 7:4	保留，必须保持复位值
Bits 3:0	DST_SGLREQ[3:0] : 目标单次传输请求
	0: 无操作（默认）
	1: 发出目标单次传输请求

8.3.3.6.5 源最后传输软件请求寄存器（DMA_LstSrcReg）

地址偏移：0x388

复位值：0x0000_0000

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

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

Bits 31:12	保留，必须保持复位值
Bits 11:8	LSTSRC_WE[3:0] : 源最后传输请求写使能
	0: 写禁止（默认）
	1: 写使能
Bits 7:4	保留，必须保持复位值
Bits 3:0	LSTSRC_REQ[3:0] : 源最后传输请求
	0: 无操作（默认）
	1: 发出当前块的最后传输请求

8.3.3.6.6 目标最后传输软件请求寄存器 (DMA_LstDstReg)

地址偏移: 0x390

复位值: 0x0000_0000

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

Bits 31:12	保留，必须保持复位值
Bits 11:8	LSTDST_WE[3:0] : 目标最后传输请求写使能
	0: 写禁止（默认）
	1: 写使能
Bits 7:4	保留，必须保持复位值
Bits 3:0	LSTDST[3:0] : 目标最后传输请求
	0: 无操作（默认）
	1: 发出当前块的最后传输请求

8.4 DMA 请求映射

从外设（TIMER、ADC 等）产生的请求信号，映射到 DMA 的 8 个硬件请求接口，再输入到 DMA 的控制器。Reqx_map 从 32 路输入源请求信号中选择一个请求作为 8 路 DMA 的输入请求信号。具体请参考 6.3.1 互联配置寄存器 0（SysCtrl_EDU_CFG0）的内容。

DMA 的所有硬件接口的请求信号，经过 DMA 内部配置后分配给 4 个 DMA 通道。最后通道请求信号经过仲裁之后，决定某个通道获得访问权。

注：只有当应用程序确保不会同时处理这些通道的请求时，才能将同一个外设请求分配给两个不同的通道。换句话说，如果两个不同的通道同时接收到同一个逻辑的外设请求，就会发生不可预测的 DMA 硬件行为。

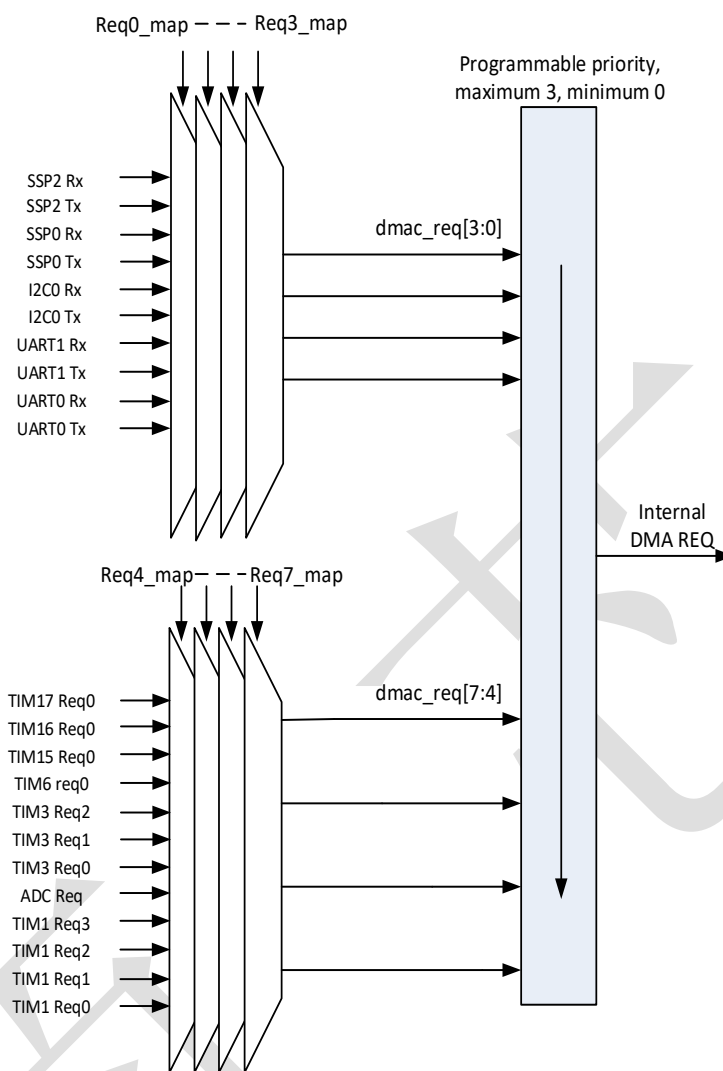


图 8-8 DMA 请求映射

8.5 DMA 参考编程流程

编程参考流程如下：

1. 首先读取通道使能寄存器，选取一个没有使用的通道；
2. 清除该通道上的所有中断标志位，读取中断原始状态寄存器和中断状态寄存器确认所有中断已经清除；
3. 配置源地址SARx和目标地址DARx；
4. 根据需要配置CTLx控制寄存器和CFGx配置寄存器，完成对通道的配置；
5. 开启DMA全局使能（仅首次初始化DMA时需要），设置通道使能寄存器使能该通道；
6. 一旦所有数据传输完成，硬件产生相应中断信号，一次完整的Block传输完成。

9. 中断和事件

9.1 嵌套向量中断控制器（NVIC）

9.1.1 NVIC 主要特性

- 32个可屏蔽中断通道（不包括16个ARM®Cortex®-M0中断线）
- 4个可编程中断优先级
- 低延迟异常和中断处理
- 电源管理控制
- 系统控制寄存器的实现

NVIC 与处理器核心接口紧密耦合，实现了低延迟中断处理和对延迟中断的高效处理。包括核心异常在内的所有中断都由 NVIC 管理。有关异常和 NVIC 编程的更多信息，请参阅针对 Cortex-M0 产品的编程手册。

9.1.2 中断和异常向量

表 9-1 中断和异常向量表

位置	优先级	优先级类型	模块	说明	向量
-	-	-	-	保留	0x0000_0000
-	-3	固定	Reset	Reset	0x0000_0004
-	-2	固定	NMI	不可屏蔽中断。复位时钟控制（RCC）的时钟安全系统（CSS）异常连接到该中断。	0x0000_0008
-	-1	固定	HardFault	任何处理器的内部故障	0x0000_000C
-	3	可配	SVCALL	由 SWI 指令发起的系统服务程序	0x0000_002C
-	5	可配	PendSv	挂起的系统服务	0x0000_0038
-	6	可配	SysTick	系统 Sys Tick 定时器	0x0000_003C
0	7	可配	WWDG	窗口看门狗中断	0x0000_0040
1	8	可配	EXTI16_LVD	EXTI Line[16]中断（低压检测）	0x0000_0044
2	9	可配	EXTI17_18 (WT)	EXTI Line[18:17]中断（WT）	0x0000_0048
3	10	可配	Flash	Flash 全局中断	0x0000_004C
4	11	可配	RCC (CLK_ERR)	复位时钟控制中断	0x0000_0050
5	12	可配	EXTI0_1	EXTI Line[1:0]中断	0x0000_0054
6	13	可配	EXTI2_3	EXTI Line[3:2]中断	0x0000_0058
7	14	可配	EXTI4_15	EXTI Line[15:4]中断	0x0000_005C
8	15	可配	EXTI19_21	EXTI Line[21:19]中断（ACMP0~2）	0x0000_0060
9	16	可配	DMA_CH0	DMA 通道 0 中断	0x0000_0064
10	17	可配	DMA_CH1_2	DMA 通道 1、2 中断	0x0000_0068
11	18	可配	DMA_CH3	DMA 通道 3 中断	0x0000_006C
12	19	可配	ADC	ADC 全局中断	0x0000_0070
13	20	可配	TIM1_NON_CC	TIM1 非捕获比较中断	0x0000_0074
14	21	可配	TIM1_CC	TIM1 捕获比较中断	0x0000_0078
15	-	-	-	-	0x0000_007C

16	23	可配	TIM3	TIM3 全局中断	0x0000_0080
17	24	可配	TIM6_DAC0_1	TIM6 和 DAC0、1 全局中断	0x0000_0084
18	-	-	-	-	0x0000_0088
19	26	可配	TIM14	TIM14 全局中断	0x0000_008C
20	27	可配	TIM15	TIM15 全局中断	0x0000_0090
21	28	可配	TIM16	TIM16 全局中断	0x0000_0094
22	29	可配	TIM17	TIM17 全局中断	0x0000_0098
23	30	可配	I2C0	I2C0 全局中断	0x0000_009C
24	-	-	保留	-	0x0000_00A0
25	32	可配	SSP0	SSP0 全局中断	0x0000_00A4
26	33	可配	SSP2	SSP2 全局中断	0x0000_00A8
27	34	可配	UART0	UART0 全局中断	0x0000_00AC
28	35	可配	UART1	UART1 全局中断	0x0000_00B0
29	-	-	-	-	0x0000_00B4
30	-	-	-	-	0x0000_00B8
31	38	可配	DIV	DIV 中断	0x0000_00BC

9.2 外部中断/事件控制器 (EXTI)

外部中断/事件控制器 (EXTI) 管理外部和内部的异步事件/中断, 并向 CPU/中断控制器生成事件请求, 并向电源管理器生成唤醒请求。

EXTI 允许管理 22 个外部/内部事件线 (多达 30 个通用 I/O 口连接到 16 个外部中断线和 6 个内部事件线)。

每个中断/事件线都可以独立地配置它的触发事件 (上升沿或下降沿或双边沿), 并能够单独地被屏蔽; 一个挂起寄存器维持所有中断请求的状态。

EXTI 可以检测到脉冲宽度小于内部 APB1 的时钟周期, 也可以配置为同步滤波窄脉冲 (依赖于时钟频率, 脉冲宽度可由纳秒级到毫秒级)。

9.2.1 EXTI 主要特性

外部中断和事件控制器 (EXTI) 的主要特性:

- 支持产生 6 个内部和 16 个外部中断/事件的请求
- 每个中断/事件线都有独立的屏蔽位
- 每个中断/事件线都有独立的触发
- 每个中断/事件线都有专用的状态位
- 支持锁存 (Latch) 模式, 可以检测脉冲宽度低于 PCLK1 时钟宽度的信号
- 支持中断/事件信号线的数字滤波
- 当处于低功耗状态时, 支持唤醒事件的自动屏蔽

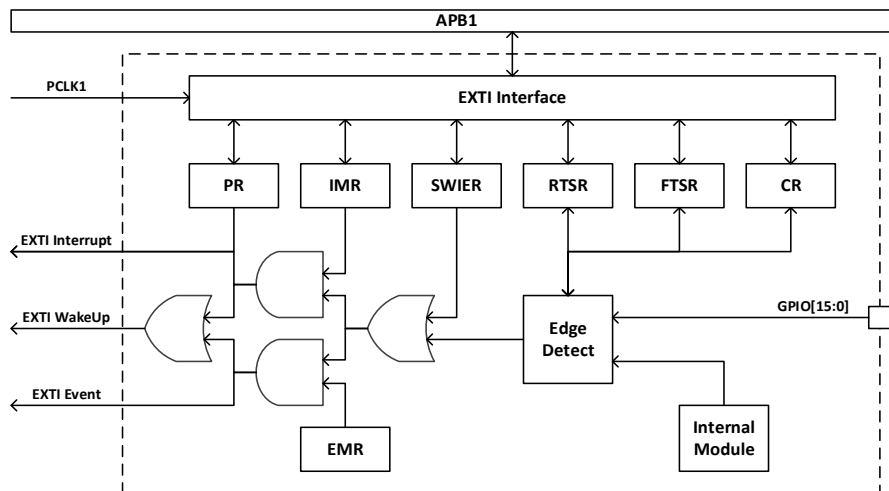


图 9-1 外部中断/事件控制器框图

9.2.2 EXTI 事件管理

MCU 可以处理外部或者内部的事件来唤醒内核（WFE）。唤醒事件可以通过下述配置产生：

- 在外设控制寄存器中使能中断，但在 NVIC 中禁止中断，并在 Cortex-M0 系统控制寄存器中使能 SEVONPEND 位。当 MCU 从 WFE 恢复时，EXTI 外设 EXTI.PR 和外设 NVIC IRQ 通道挂起位（在 NVIC 中断清除挂起寄存器中）必须被清除。
- 或者，在事件模式下配置外部或内部 EXTI 线。当 CPU 从 WFE 恢复时，由于事件行对应的挂起位没有设置，因此不需要清除 EXTI.PR 或 NVIC IRQ 通道挂起位。

9.2.3 EXTI 功能说明

如果需要产生中断，必须先配置好并使能中断线。根据需要的边沿检测设置 2 个触发寄存器，同时在中断屏蔽寄存器的相应位写 1 使能中断请求。当中断线上发生了期待的边沿时，将产生一个中断请求，对应的中断挂起位也随之被置 1。在 PR 寄存器的对应位写 1，将清除中断请求。

如果需要产生事件，必须先配置好并使能事件线。根据需要的边沿检测设置 2 个触发寄存器，同时在事件屏蔽寄存器的相应位写 1 允许事件请求。当事件线上发生了期待的边沿时，将产生一个事件请求，但对应的 PR.PR_x 不被置 1。

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。

硬件中断选择

将某行配置为中断源，操作步骤如下：

- 配置所选事件线的 EXTI_IMR.IM_x 中断屏蔽位
- 配置所选中断线的触发选择位（EXTI_RTSR、EXTI_FTSR 和 EXTI_CR.LR_x）
- 配置控制映射到 EXTI 的 NVIC IRQ 通道的使能位和屏蔽位，这样来自 EXTI 线路之一的中断就可以被正确响应

硬件事件选择

将某一行配置为事件源，操作步骤如下：

- 配置所选事件线的 EXTI_EMR.EMx 事件屏蔽位
- 配置所选事件线的触发选择位 (EXTI_RTSR、EXTI_FTSR 和 EXTI_CR.LRx)

软件中断/事件的选择

任何外部线路都可以配置为软件中断/事件线路。下面是生成软件中断的过程：

- 配置所选事件线的中断/事件屏蔽位 (EXTI_IMR、EXTI_EMR)
- 配置所选事件线的软件中断寄存器的请求位 (EXTI_SWIER)

9.2.4 EXTI 外部和内部中断/事件线的映射

GPIO 与 16 个外部中断/事件线相连接，方式如下：

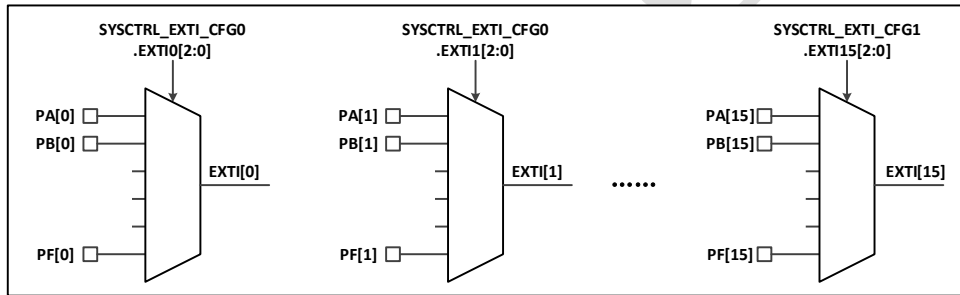


图 9-2 外部中断通用 I/O 映射

其余几行 x ($x = 16 \sim 21$) 按如下方式连接：

- EXTI 线 16 连接到 LVD 输出
- EXTI 线 17 和 18 连接到 WT 闹钟唤醒事件
- EXTI 线 19 连接到模拟比较器 ACMP0 的输出
- EXTI 线 20 连接到模拟比较器 ACMP1 的输出
- EXTI 线 21 连接到模拟比较器 ACMP2 的输出

9.3 EXTI 寄存器描述

表 9-2 EXTI 相关寄存器表

名称	说明	读写权限	复位值	字节地址
IMR	中断屏蔽寄存器	R/W	0x0000_0000	0x4001_1000
EMR	事件屏蔽寄存器	R/W	0x0000_0000	0x4001_1004
RTSR	上升沿触发选择寄存器	R/W	0x0000_0000	0x4001_1008
FTSR	下降沿触发选择寄存器	R/W	0x0000_0000	0x4001_100C
SWIER	软件中断事件寄存器	R/W	0x0000_0000	0x4001_1010
PR	挂起寄存器	R/W	0x0000_0000	0x4001_1014
FR	滤波寄存器	R/W	0x0000_0000	0x4001_1018
CR	控制寄存器	R/W	0x0000_0000	0x4001_101C
SysCtrl_EXTI_CFG0	系统 EXTI 配置寄存器 0	R/W	0x0000_0000	0x4800_7018
SysCtrl_EXTI_CFG1	系统 EXTI 配置寄存器 1	R/W	0x0000_0000	0x4800_701C

注：x 表示不确定；-表示未实现；u 表示不受复位影响；c 表示该位复位值因复位源而异；R 只读；W 只写；R-0 表示只读且读 0；R/W 表示可读写（以后章节同上述）。

9.3.1 EXTI 中断屏蔽寄存器 (EXTI_IMR)

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
Res.															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IM[21:0]															
rw															

Bits	31:22	保留, 必须保持复位值
Bits	21:0	IMx : 线 x 上的中断屏蔽 (x = 0~21)
		0: 屏蔽来自线 x 上的中断请求
		1: 不屏蔽来自线 x 上的中断请求

9.3.2 EXTI 事件屏蔽寄存器 (EXTI_EMR)

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
Res.															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EM[21:0]															
rw															

Bits	31:22	保留, 必须保持复位值
Bits	21:0	EMx : 线 x 上的事件屏蔽 (x = 0~21)
		0: 屏蔽来自线 x 上的事件请求 (默认)
		1: 不屏蔽来自线 x 上的事件请求

9.3.3 EXTI 上升沿触发选择寄存器 (EXTI_RTSR)

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
Res.															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RT[21:0]															
rw															

Bits 31:22	保留，必须保持复位值
Bits 21:0	RTx : 线 x 上的上升沿触发事件配置位 (x = 0~21)
	0: 禁止输入线 x 上的上升沿触发 (中断和事件) (默认)
	1: 使能输入线 x 上的上升沿触发 (中断和事件)
	注 : 同一根输入线既可以被配置为上升沿触发，也可以被配置为下降沿触发。

9.3.4 EXTI 下降沿触发选择寄存器 (EXTI_FTSCR)

地址偏移: 0x0C

复位值: 0x0000_0000

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

Bits 31:22	保留，必须保持复位值
Bits 21:0	FTx : 线 x 上的下降沿触发事件配置位 (x = 0~21)
	0: 禁止输入线 x 上的下降沿触发 (中断和事件) (默认)
	1: 使能输入线 x 上的下降沿触发 (中断和事件)
	注 : 同一根输入线既可以被配置为上升沿触发，也可以被配置为下降沿触发。

9.3.5 EXTI 软件中断事件寄存器 (EXTI_SWIER)

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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIER[21:0]															
rw															

Bits 31:22	保留，必须保持复位值
Bits 21:0	SWIERx : 线 x 上的软件中断和事件 (x = 0~21)
	当该位为 0 时，写 1 将设置 EXTI_PR 中相应的挂起位。如果在 EXTI_IMR 和 EXTI_EMR 中允许产生中断，则此时将产生一个中断。通过清除 EXTI_PR 的对应位 (写入 1)，

硬件自动清除该位为 0。

9.3.6 EXTI 挂起寄存器 (EXTI_PR)

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
Res.															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PR[21:0]															
rw															

Bits	31:22	保留, 必须保持复位值
Bits	21:0	PRx : 线 x 上的挂起位 (x = 0~21)
		0: 没有发生触发请求 (默认)
		1: 已经发生触发请求
		当在外部中断线上发生了触发事件, 该位被硬件置 1。该位软件写 1 清除。

9.3.7 EXTI 滤波寄存器 (EXTI_FR)

地址偏移: 0x18

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FR15[1:0]		FR14[1:0]		FR13[1:0]		FR12[1:0]		FR11[1:0]		FR10[1:0]		FR9[1:0]		FR8[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
FR7[1:0]		FR6[1:0]		FR5[1:0]		FR4[1:0]		FR3[1:0]		FR2[1:0]		FR1[1:0]		FR0[1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	

Bits	31:0	FRx[1:0] : EXTI 输入线 x (x = 0~15) 上的滤波选择
		只有 EXTI 输入线 x 上的事件连续保持 N 个采样时钟以后, 事件才被认为有效
		00: 没有滤波, 以 EXTI 采样时钟的频率 (f _{ck_psc}) 采样 (默认)
		01: N=2
		10: N=4
		11: N=8

9.3.8 EXTI 控制寄存器 (EXTI_CR)

偏移地址: 0x1C

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Res.	PSC[2:0]			Res.												
	rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LR[21:0]															
rw															

Bit	31	保留, 必须保持复位值
Bits	30:28	PSC[2:0]: EXTI 采样时钟分频系数
		000: $f_{CK_PSC} = f_{PCLK1}$ (默认)
		001: $f_{CK_PSC} = f_{PCLK1} / 2$
		010: $f_{CK_PSC} = f_{PCLK1} / 4$
		011: $f_{CK_PSC} = f_{PCLK1} / 8$
		100: $f_{CK_PSC} = f_{PCLK1} / 16$
		101: $f_{CK_PSC} = f_{PCLK1} / 32$
		110: $f_{CK_PSC} = f_{PCLK1} / 64$
		111: 保留
Bits	27:22	保留, 必须保持复位值
Bits	21:0	LRx: 线 x (x = 0~21) Latch 模式使能
		该模式用于待检测信号的脉冲宽度小于一个采样时钟
		0: 同步时钟模式进行采样 (默认)
		1: 使能 Latch 模式进行采样

9.3.9 系统 EXTI 配置寄存器 0 (SysCtrl_EXTI_CFG0)

地址偏移: 0x18 (相对 SYSCTRL 的起始地址 0x4800_7000)

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	EXTI7[2:0]			Res.	EXTI6[2:0]			Res.	EXTI5[2:0]			Res.	EXTI4[2:0]		
	rw				rw				rw				rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	EXTI3[2:0]			Res.	EXTI2[2:0]			Res.	EXTI1[2:0]			Res.	EXTI0[2:0]		
	rw				rw				rw				rw		

Bits	31:0	EXTIx[2:0]: EXTI 外部中断和事件源引脚配置 (x = 0~7)
		000: GPIOAx (默认)
		001: GPIOBx
		010: 保留
		101: GPIOFx
		其他: 保留
		注: 实际 I/O 的选择请参考各端口 I/O 数量。

9.3.10 系统 EXTI 配置寄存器 1 (SysCtrl_EXTI_CFG1)

地址偏移: 0x1C (相对 SYSCTRL 的起始地址 0x4800_7000)

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	EXTI15[2:0]			Res.	EXTI14[2:0]			Res.	EXTI13[2:0]			Res.	EXTI12[2:0]		
	rw				rw				rw				rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	EXTI11[2:0]			Res.	EXTI10[2:0]			Res.	EXTI9[2:0]			Res.	EXTI8[2:0]		
	rw				rw				rw				rw		

Bits 31:0	EXTIx[2:0]: EXTI 外部中断和事件源引脚配置 (x = 8~15)
	000: GPIOAx (默认)
	001: GPIOBx
	010: 保留
	101: GPIOFx
	其他: 保留
	注: 实际 I/O 的选择请参考各端口 I/O 数量。

10. 模数转换器（ADC）

10.1 ADC 主要特性

12 位 ADC 是一种逐次逼近模数转换器。它有多达 18 个多路复用通道，允许它测量来自 16 个外部 I/O 和 2 个内部的电压。

各种通道的 A/D 转换可以在单次、连续或断续模式下进行。ADC 的结果存储在左对齐或右对齐的 16 位数据位宽、16 深度的 FIFO 中。模拟看门狗功能允许应用程序检测输入电压是否超出用户定义的高或低阈值。支持一种高效的低功耗模式，允许在低频率工作时降低功耗。ADC 输入时钟由 PCLK1 或 PLL_CLK 时钟通过预分频产生，且不能超过 24MHz。

- 卓越高效的性能
 - 12 位精度的转换
 - 转换速率高达 1.5MSPS (@24MHz)
 - 可配置的采样时间
 - 自校准
 - 数据对齐与内置数据一致性
 - 内置 16 深度 FIFO
 - 支持 DMA
- 低功耗
 - 独立的工作时钟（ADC_CLK），应用程序可以运行在低频的 PCLK 上不影响 ADC 效率
 - 两种低功耗模式：在能满足性能要求的情况下降低功耗
 - 自动延迟转换模式：防止 ADC 在低频率 PCLK 应用中数据溢出错乱问题
- 模拟输入通道
 - 16 个外部 I/O 通道
 - 1 个内部温度传感器通道（ST）
 - 1 个内部参考电压采样通道（SV）
- 可以触发转换启动：
 - 由软件触发
 - 由硬件触发（包括内部 TIM 的各种事件、EXTI 事件、ACMP 输出）
- 采样模式：ADC 双采样保持电路，支持两个通道同时采样
 - 顺序采样/单采样：在一个采样转换周期内 1 个通道被采样
 - 同步采样/双采样：2 个通道同时被采样
- 转换模式：
 - 单次模式：一次触发转换一个序列中的所有通道
 - 连续模式：连续的转换所有选中的通道
 - 断续模式：一次触发转换一个所选的通道
- 1 个独立的中断向量入口，内部支持多种可配置中断
- 模拟看门狗
- ADC 供电：
 - 内部参考电压（3.3V 或 5V）
 - 外部参考电压（PA0 输入电压、VDDA、2.5V 或 4V）



10.2 ADC 功能描述

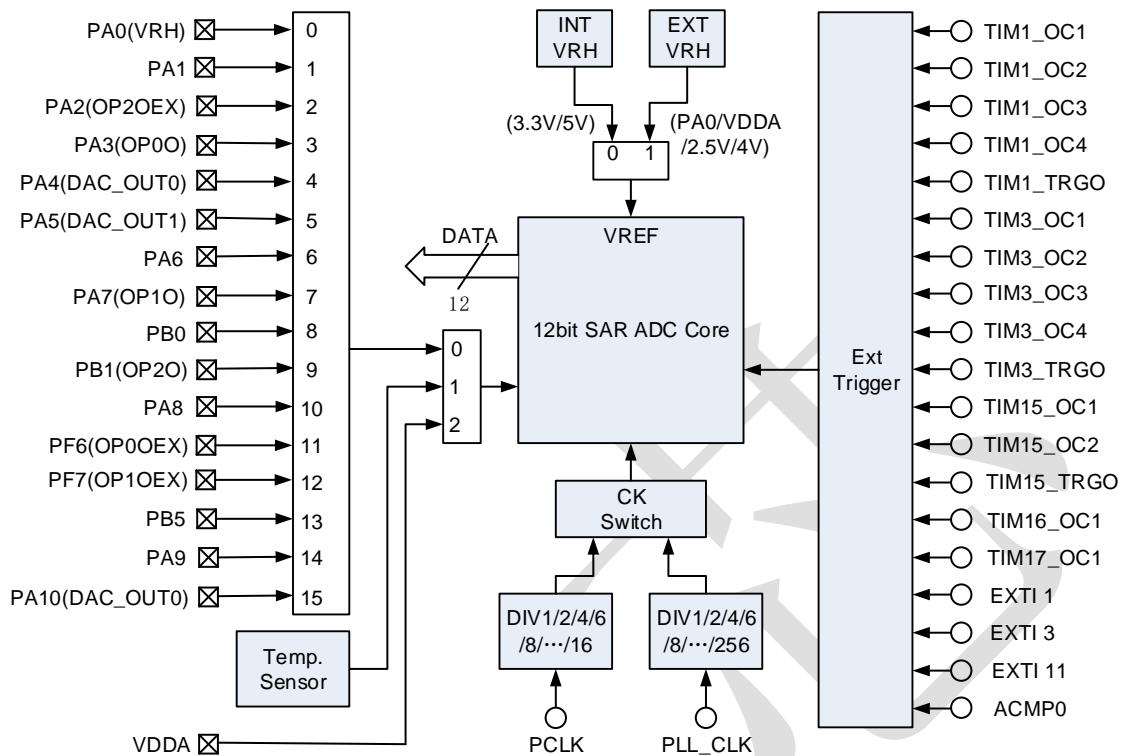


图 10-1 ADC 框图

注1: OP0OEX/OP1OEX/OP2OEX 的输出直接连到 ADC 的输入通道，I/O 无须配置成模拟模式，模拟通道不需要使能；I/O 空闲，可以用作其他功能；

注2: PMU_VRH/OP0O/OP1O/OP2O/ DAC_OUT0/DAC_OUT1 输出可以通过 I/O 的两个模拟通道环路到 ADC 的输入通道。I/O 必须配置成模拟模式，两个模拟通道都要使能；外部高阻：I/O 被占用，不得用作其他功能。

10.2.1 ADC 开关控制

在 MCU 上电的默认情况下，ADC 模块是禁用且处于断电模式（ADEN=0）。

如图 10-2 所示，在 ADC 开始精确转换前需要一个稳定时间 t_{STAB} ，期间会发送一系列脉冲信号用来调整内部电路的共模电压到合理的值。如果刚上电就立刻转换，初始的一些转换结果不可靠。

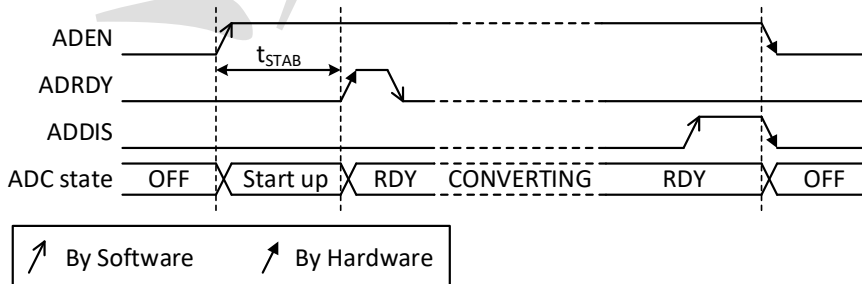


图 10-2 使能/禁用 ADC

两个控制位用于开启或禁用 ADC:

- 设置 ADEN=1 开启 ADC。当 ADC 模块准备好时 ADRDY 标志置 1。
- 设置 ADDIS=1 来关闭 ADC，并使 ADC 处于断电模式。当 ADC 模块全关断后，硬件自动清除 ADEN 和 ADDIS 位。

ADC 可由设置 ADSTART=1 来软件启动或由外部触发事件来触发启动 (ADSTART = 1)。

以下是使能 ADC 的过程:

- 通过写 1 清除 ADC_ISR.ADRDY 位。
- 在 ADC_TESTCFG 寄存器中设置 PD_ADC、PD_REF、PD_SH2、PD_SH1 为 0 启动 ADC 硬件模块。
- 置位 ADC_CR.ADEN=1。
- 等待直到 ADC_ISR.ADRDY=1 (当 ADC 启动完成后 ADRDY 置位)。若 ADC_IER.ADRDYIE=1 设置 ADC Ready 的中断使能时，也可通过中断来处理 ADC Ready。

以下为禁用 ADC 的过程:

- 检查 ADC_CR.ADSTART 是否为 0 以确保 ADC 不在转换过程中。若需要，可对 ADC_CR.ADSTP 置 1 打断正在进行的 ADC 转换，并等待 ADSTP 被硬件清 0 (清 0 表示转换停止完成)。
- 设置 ADC_CR.ADDIS=1。
- 若应用要求，则可等待 ADC_CR.ADEN 位。读此位为 0，其表明 ADC 模块完全关闭 (硬件同时自动将 ADC_CR.ADEN 和 ADC_CR.ADDIS 清 0)。

10.2.2 ADC 时钟

ADC 具有双时钟域架构，ADC 时钟 (ADC_CLK) 独立于 APB1 时钟 (PCLK1)。

1. ADC_CLK 是 ADC 模块工作时使用的时钟；
2. PCLK1 是配置 ADC 寄存器时使用的时钟。

ADC_CLK 可由两种时钟源选择后产生，具体参见 CKMODE[4:0]:

1. PLL_CLK: 与 PCLK1 异步。优势在于：无论系统频率高低，ADC 都能工作在最高频率。
2. PCLK1: 由 PCLK1 分频而得，分频时钟可由 CKMODE[3:0]设置。避免了跨时钟域问题，在 ADC 需要精确触发的情况下具有较大优势，延迟低。(异步时钟会出现触发延迟的情况)。

10.2.3 ADC 寄存器配置

软件必须在 ADC 禁止 (ADC_CR.ADEN 必须为 0) 的情况下设置 ADC_CR.ADEN=1。

软件必须在 ADC 开启且没有禁止请求挂起 (即 ADC_CR.ADEN=1 且 ADC_CR.ADDIS=0) 的情况下设置 ADC_CR.ADSTART 和 ADC_CR.ADDIS 位为 1。

对于以下的这些位 ADC_IER、ADC_CFGRI、ADC_SMPR、ADC_TR、ADC_CHSELR、ADC_TESTCFG、ADC_EXTCFG、ADC_TRIM 寄存器，软件必须在 ADC 开启 (ADC_CR.ADEN = 1) 且不在转换期间 (ADC_CR.ADSTART = 0) 的情况下才能进行写操作。

软件必须在 ADC 开启且无挂起请求 (ADC_CR.ADSTART=1 和 ADC_CR.ADDIS=0) 的情况下才能改写 ADC_CR.ADSTP 位。

注: 在软件违反以上寄存器配置要求时，硬件具有保护机制。假如违反了以上的寄存器配置，ADC 会进入一种不确定状态。为了从这种情况下恢复，ADC 必须被关闭 (ADC_CR.ADEN=0 且 ADC_CR 寄存器清 0)。

10.2.4 ADC 通道选择

ADC 拥有多达 18 个复用通道:

- 16 个外部 I/O 通道（ADC_IN0、ADC_IN1、...、ADC_IN15）
- 2 个内部模拟输入（温度传感器和内部参考电压）

要转换的通道序列必须在 ADC_CHSELR 通道选择寄存器中编程:每个模拟输入通道都有一个专用的选择位（CHSEL0...CHSEL15）。内部通道和外部通道不能同时使能。

10.2.4.1 ADC 外部 I/O 通道

ADC 有 16 个外部 I/O 通道连接,对应 ADC_IN0~15 的 ADC 接口。AD_IN0~7 通道组成通道 A 组,ADC_IN8~15 通道组成通道 B 组。使用外部 I/O 通道,需要配置 ADC_CHSELR.ST=0 和 ADC_CHSELR.SV=x。ADC 和其他模拟模块联动有两种连接方式:

1. ADC 的输入端和其他模拟模块输出端处于同一个 GPIO 模拟通道上,则属于直连。模拟信号从其他模拟模块直接传递到 ADC,不经过 GPIO,此时 GPIO 可以用作输入/输出/数字复用功能;
2. ADC 的输入端和其他模拟模块输出端处于在两个 GPIO 模拟通道上,则属于环路。模拟信号从其他模拟模块先传递到 GPIO,通过 GPIO 再传递到 ADC 形成环路,此时 GPIO 必须打开两路模拟通道,即 GPIO 被占用,不能用于其他用途。

ADC 所有外部 I/O 通道和相关 GPIO 的具体说明如下表:

表 10-1 ADC 外部 I/O 通道表

ADC 外部通道接口	通道组	I/O	GPIO 模拟通道	
			YA1	YA2
ADC_IN0	A 组	PA0_YA1		PMU_VRH
ADC_IN1		PA1_YA1		
ADC_IN2		PA2_YA2		OP2OEX
ADC_IN3		PA3_YA1		OP00
ADC_IN4		PA4_YA2	DAC_OUT0	
ADC_IN5		PA5_YA1		DAC_OUT1
ADC_IN6		PA6_YA1		
ADC_IN7		PA7_YA1		OP10
ADC_IN8	B 组	PB0_YA2		
ADC_IN9		PB1_YA1		OP20
AD_CIN10		PA8_YA1		
ADC_IN11		PF6_YA1	OP00EX	
ADC_IN12		PF7_YA1	OP10EX	
ADC_IN13		PB5_YA1		
ADC_IN14		PA9_YA1		
ADC_IN15		PA10_YA1		DAC_OUT0

10.2.4.2 内部通道

10.2.4.2.1 温度传感器

温度传感器可以用来测量器件的结点温度 (T_J)。温度传感器内部连接到 ADC_IN16 输入通道,可用于转换传感器的电压值。

使用温度传感器,需要配置 ADC_CHSELR.ST=1 和 ADC_CHSELR.SV=0。此时 ADC 连接到 ADC_IN16 通道,其他通道 (ADC_IN0~15, ADC_IN17) 禁用。

注：传感器从断电模式下唤醒时到能正确输出 V_{SENSE} 的值需要一个启动时间，ADC 从启动上电后到准备完毕也有一个启动时间，若要减少这个延时，则需要在同一时间设置 $ADC_CR.ADEN$ 和 $ADC_CHSELR.ST/SV$ 位。

10.2.4.2.2 内部参考电压 V_{DDA}

模拟电源电压内部连接到 ADC_IN17 输入通道。

检测模拟电源电压，需要配置 $ADC_CHSELR.ST=1$ 和 $ADC_CHSELR.SV=1$ 。此时 ADC 连接到 ADC_IN17 通道，其他通道（ $ADC_IN0\sim15$ ， ADC_IN16 ）禁用。

10.2.4.3 通道序列

ADC 可以转换一个单一通道或自动扫描一个序列通道，被转换的通道序列必须在通道选择寄存器 ADC_CHSELR 中编程选择：每个模拟输入通道有专门的一位选择位（ $CHSEL0 \dots CHSEL15$ ， SV/ST ）。ADC 扫描的通道顺序由 $ADC_CFGR1.SCANDIR$ 位的配置来决定：

- $SCANDIR=0$ ：递增扫描，从通道 0 到通道 15
- $SCANDIR=1$ ：递减扫描，从通道 15 到通道 0

表 10-2 通道序列配置

CHB	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8
	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
CHA	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

10.2.5 ADC 可编程采样时间（SMP）

在启动数模转换之前，ADC 需要在被测电压源和内嵌采样电容间建立一个直接连接。采样时间必须足够长以便输入电压源对内嵌电容充电到输入电压的水平。可编程采样时间根据输入电压的输入阻抗来调整转换速度。

ADC 采样输入电压所用的 ADC 时钟个数可用 ADC_SMPR 寄存器中的 $SMP[2:0]$ 位来进行修改。可编程采样时间对所有通道都通用。如有应用需求，则可用软件改变和适应不同通道间的采样时间。采样时间取值为 1~8 个 ADC 时钟周期。

$$T(\text{sample pulse}) = (1 + SMP[2:0]) * ADC_CLK$$

ADC 用设置 $EOSMP$ 标志来表明采样阶段的结束。

10.2.6 ADC 可编程转换时间（CONVERT）

可编程采样转换周期，包括采样时间和等待转换时间。可通过配置 $CONVERT$ 和 SMP ，灵活调整 ADC 的转换速度。不同模式下总采样转换时间的计算公式如下：

- 顺序采样模式（单采样）： $T(\text{采样转换周期}) = (14 + CONVERT[4:0]) * ADC_CLK$
- 同步采样模式（双采样）： $T(\text{采样转换周期}) = (25 + CONVERT[4:0]) * ADC_CLK$

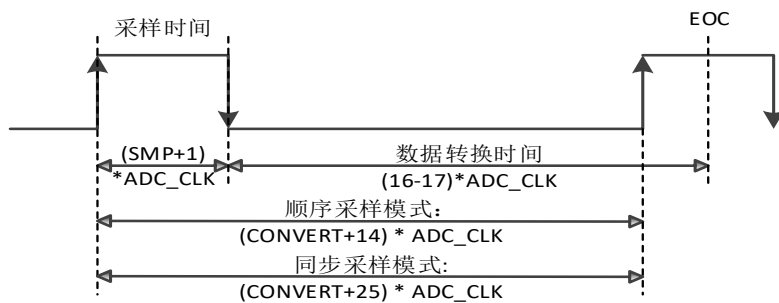


图 10-3 采样周期图

10.2.7 ADC 转换模式

10.2.7.1 单次转换模式 (CONT=0)

单次转换模式下，ADC 执行一次序列转换，转换所有被选的通道。当 ADC_CFGR1 寄存器中 CONT=0 时 ADC 为单次转换模式。ADC 转换可由下述两种方法启动：

- 在 ADC_CR 寄存器中设置 ADSTART 位软件触发
- 硬件触发事件 (ADSTART = 1 时)

在序列通道的转换中，每次转换完成后：

- 转换的数据结果存放到 FIFO 中
- EOC (转换结束标志) 标志置位
- 若 EOCIE 位置位则产生一个中断

通道序列转换完成后：

- EOSEQ (序列结束) 标志置位
- 若 EOSIE 位置位则产生一个中断

转换结束后，ADC 停止直到新的硬件触发事件或 ADSTART 重新置位。

注：若转换单一通道，则可编程一个长度为 1 的一个转换序列。

10.2.7.2 连续转换模式 (CONT=1)

在连续转换模式中，当软件或硬件触发事件产生，ADC 执行一个序列转换。转换所有的通道一次且自动重新开始执行相同的序列转换。当寄存器 ADC_CFGR1 中的 CONT=1 时，ADC 选择为连续转换模式。ADC 转换可由下述两种方法启动：

- 软件触发，在 ADC_CR 寄存器中设置 ADSTART 位软件触发；
- 硬件触发事件 (ADSTART = 1 时)；

在序列通道的转换中，每次转换完成后：

- 转换的数据结果存放到 FIFO 中；
- EOC (转换结束标志) 标志置位
- 若 EOCIE 位置位则产生一个中断。

通道序列转换完成后：

- EOSEQ (序列结束) 标志置位
- 若 EOSIE 位置位则产生一个中断



一次序列转换结束后，ADC 立即重新转换相同的序列通道。

注：不可能同时启用不连续模式和连续模式；禁止同时设置 $DISCEN=1$ 和 $CONT=1$ 。

10.2.7.3 断续转换模式（ $DISCEN=1$ ）

通过在 ADC_CFGR1 寄存器中设置 $DISCEN$ 位使能该模式。在这种模式下（ $DISCEN=1$ ），需要一个硬件或软件触发事件来启动序列中定义的所有转换。相反，如果 $DISCEN=0$ ，单个硬件或软件触发事件依次启动序列中定义的所有转换。

例如：

- $DISCEN=1$ ，需要转换的通道为：0、3、7、10
 1. 第一次触发：通道 0 被转换且一个 EOC 事件产生
 2. 第二次触发：通道 3 被转换且一个 EOC 事件产生
 3. 第三次触发：通道 7 被转换且一个 EOC 事件产生
 4. 第四次触发：通道 10 被转换且产生 EOC 和 EOSEQ 事件
 5. 第五次触发：通道 0 被转换且一个 EOC 事件产生
 6. 第六次触发：通道 3 被转换且一个 EOC 事件产生
 7.
- $DISCEN=0$ ，需要转换的通道为：0、3、7、10
 1. 第一次触发：整个完整的序列转换：依次为通道 0、3、7 和 10。每次转换产生一个 EOC 事件，到最后一通道还产生一个 EOSEQ 事件。
 2. 任何后续的触发事件将重新启动完整的序列。

10.2.8 启动转换（ $ADSTART$ ）

软件通过设置 $ADSTART=1$ 启动 ADC 转换。

$ADSTART$ 设置为 1，则转换：

- 当 $EXTEN=0x0$ （软件触发）时，立即开始
- 当 $EXTEN \neq 0x0$ 时，在下一个所选择硬件触发事件的活动边沿触发

$ADSTART$ 位也用于说明目前 ADC 转换操作是否正在进行。当 ADC 处于空闲时，该位可重新配置为 $ADSTART=0$ 。

$ADSTART$ 位可由硬件清除：

- 单次转换模式由软件触发（ $CONT=0$ ， $EXTSEL=0x00$ ），在序列转换结束后（ $EOSEQ=1$ ）
- 断续转换模式由软件触发（ $CONT=0$ ， $DISCEN=1$ ， $EXTSEL=0x00$ ），在每次转换结束后（ $EOC=1$ ）
- 在所有的情况下（ $CONT=X$ ， $EXTSEL=XX$ ），在软件调用并执行 $ADSTOP$ 过程后

注 1：在连续转换模式下（ $CONT=1$ ），当设置 $EOSEQ$ 标志时，硬件不会清除 $ADSTART$ 位，因为序列会自动重新启动。

注 2：在单次转换模式下选择硬件触发时（ $CONT=0$ ， $EXTEN=01$ ），设置 $EOSEQ$ 标志时，硬件不清除 $ADSTART$ 。这避免了软件必须再次设置 $ADSTART$ 位的需要，并确保不会错过下一个触发事件。

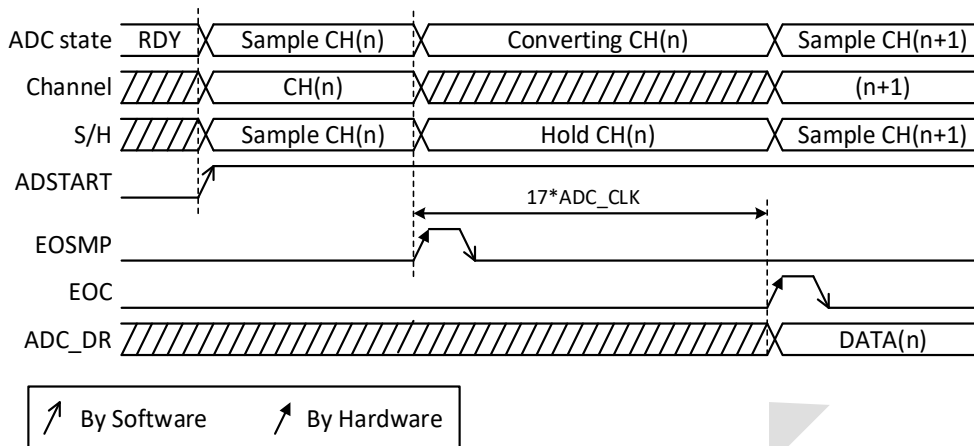


图 10-4 模数转换时序

10.2.9 停止转换 (ADSTOP)

软件可以通过在 ADC_CR 寄存器中设置 ADSTP=1 来决定停止任何正在进行的转换。这将重置 ADC 操作，ADC 将处于空闲状态，准备进行新的操作。当软件设置了 ADSTP 位时，任何正在进行的转换将被中止，结果将被丢弃 (ADC_DR 寄存器不随当前转换而更新)。扫描序列也被中止并重置 (这意味着重新启动 ADC 将重新启动一个新的序列)。一旦这个过程完成，ADSTP 和 ADSTART 位都被硬件清除，软件必须等到 ADSTART=0 才开始新的转换。

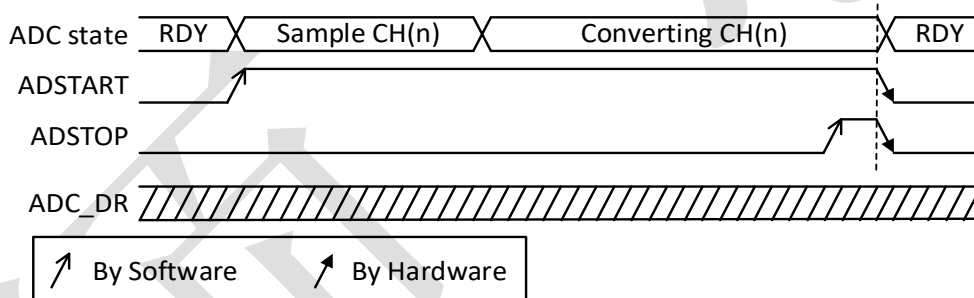


图 10-5 停止当前的转换

10.3 ADC 外部触发和触发极性

转换或转换序列可以由软件或外部事件 (例如 TIMER 捕获) 触发。如果 EXTEN[1:0] 控制位不等于 00，那么外部事件能够触发具有选定极性的转换。一旦软件设置位 ADSTART=1，触发器选择就有效。

在转换过程中发生的任何硬件触发都会被忽略。

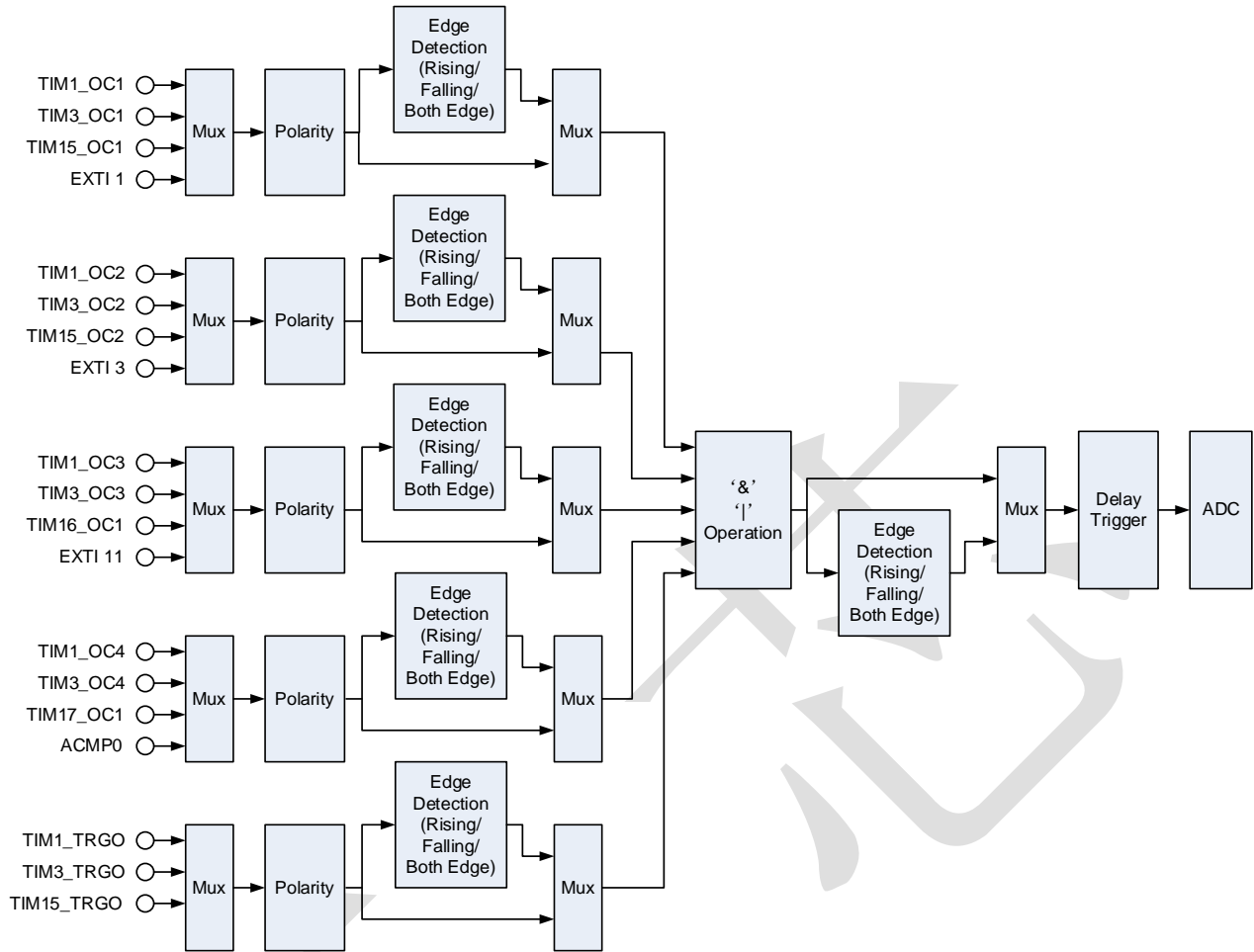


图 10-6 ADC 外部触发

表 10-3 给出 EXTEN[1:0]值与其相对应的极性。

表 10-3 触发极性表

源	EXTEN[1:0]
触发检测禁止（软件触发）	00
在上升沿时检测	01
在下降沿时检测	10
在上升沿及下降沿都检测	11

EXTSELG0~4 控制位用于选择可触发转换的事件。软件源触发事件可由设置 ADC_CR 寄存器中的 ADSTART 位来产生。

表 10-4 外部触发表

名称	来源	类型	条件
EXT0	TIM1_TRGO	定时器产生的内部信号	EXTSELG4 =00
EXT1	TIM3_TRGO		EXTSELG4 =01
EXT2	TIM15_TRGO		EXTSELG4 =10
EXT3	保留		
EXT4	TIM1_OC4	定时器产生的内部信号	EXTSELG3 =00

EXT5	TIM3_OC4		EXTSELG3 =01
EXT6	TIM17_OC1		EXTSELG3 =10
EXT7	ACMP0	比较器产生的内部信号	EXTSELG3 =11
EXT8	TIM1_OC3	定时器产生的内部信号	EXTSELG2 =00
EXT9	TIM3_OC3		EXTSELG2 =01
EXT10	TIM16_OC1		EXTSELG2 =10
EXT11	EXTI11	外部引脚	EXTSELG2 =11
EXT12	TIM1_OC2	定时器产生的内部信号	EXTSELG1 =00
EXT13	TIM3_OC2		EXTSELG1 =01
EXT14	TIM15_OC2		EXTSELG1 =10
EXT15	EXTI3	外部引脚	EXTSELG1 =11
EXT16	TIM1_OC1	定时器产生的内部信号	EXTSELG0 =00
EXT17	TIM3_OC1		EXTSELG0 =01
EXT18	TIM15_OC1		EXTSELG0 =10
EXT19	EXTI1	外部引脚	EXTSELG0 =11

注：触发器选择只能在 ADC 未转换时更改 (ADSTART=0)。

10.3.1 转换结束，采样阶段结束 (EOC 和 EOSMP 标志)

ADC 通知应用每次转换结束 (EOC) 事件。

一旦在 FIFO 中的一个转换数据有效后, ADC 在 ADC_ISR 寄存器中设置 EOC 标志表明转换完成。当 ADC_IER 中的 EOCIE 置为 1 时, 则会产生一个 EOC 中断。EOC 标志由软件写 1 清除或读 ADC_DR 寄存器来清除。

ADC 同样在 ADC_ISR 寄存器中给出采样阶段结束标志 EOSMP。EOSMP 标志可写 1 清除。当在 ADC_IER 寄存器中的 EOSMPIE 置为 1 后, 则会产生一个 EOSMP 中断。

这个中断的目的是允许处理与转换同步。通常模拟多路复用器可以在转换阶段的隐藏时间内访问, 以便在下一个采样开始时定位多路复用器。

10.3.2 序列转换结束 (EOSEQ 标志)

ADC 通知应用每次序列转换结束 (EOSEQ) 事件。

一旦一个转换序列的最后一个通道转换数据有效后, ADC 在 ADC_ISR 寄存器中设置 EOSEQ 标志。当 ADC_IER 中的 EOSIE 位置 1 时, 则会产生 EOC 中断。EOSEQ 标志由软件写 1 清 0。

10.3.3 示例时序图 (单次/连续/断续模式, 硬件/软件触发)

两路采样/维持电路可通过 ADC_CFGR2.CON 选择顺序采样或两路同时采样。相应的时序要求如图所示。信号 EOC 是转换结束标志位, 维持一个 ADC_CLK, 此刻 ADC 转换数据有效。在 Sample 信号变低后, EOC 需 16 或 17 个 ADC_CLK 变高, 最小的采样周期是 16 或 17 时钟。

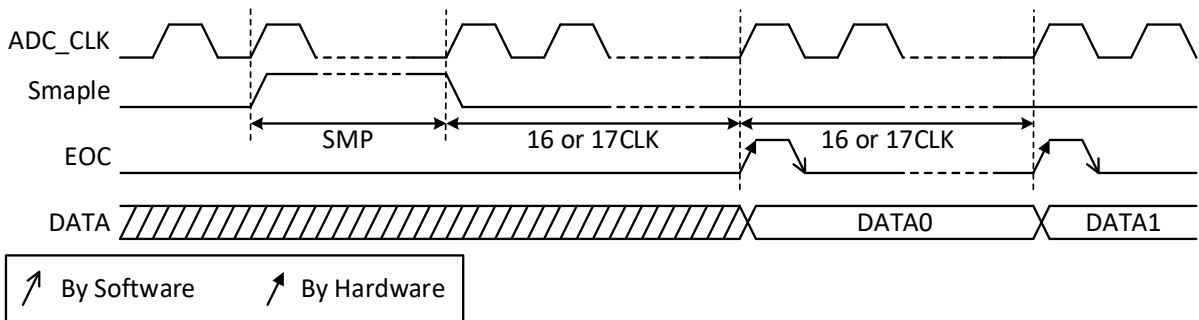


图 10-7 ADC 同时采样时序图 (CON=1)

注：调节采样周期数 *CONVERT* 可提高 ADC 双通道同时采样速率。

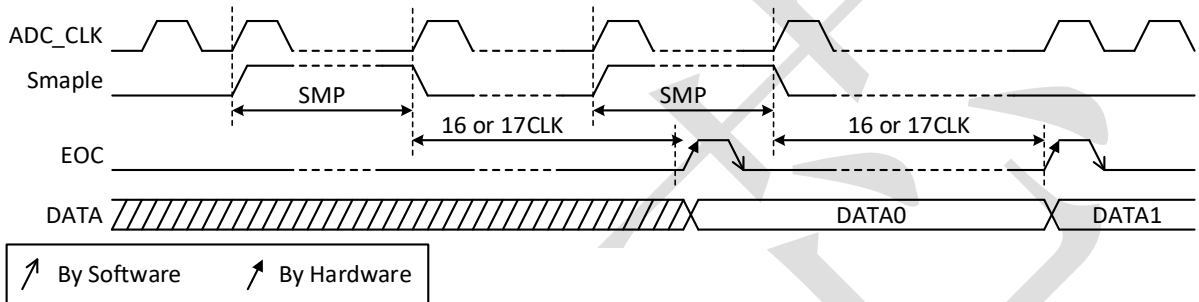


图 10-8 ADC 顺序采样时序图 (CON=0)

注：调节采样周期数 *CONVERT* 可提高 ADC 顺序采样速率。

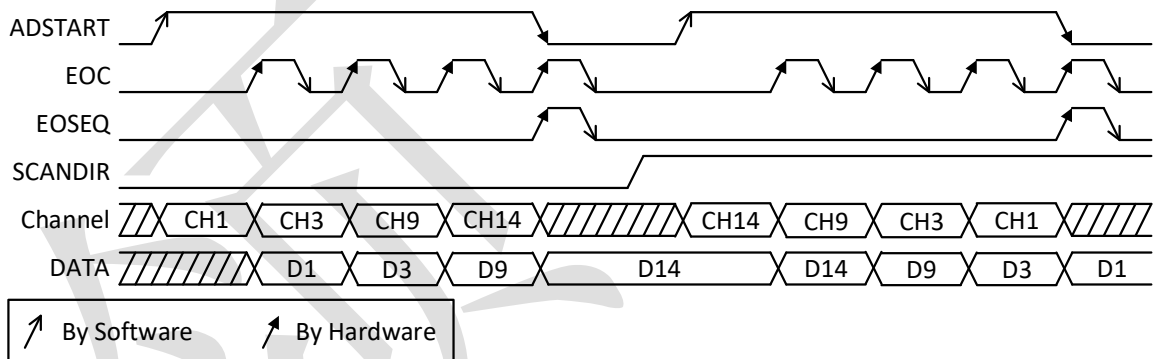


图 10-9 顺序采样下，单序列转换，软件触发

注：EXTEN=0x0, CHSEL=0x420A, WAIT=0, CON=0, CONT=0, DISCEN=0

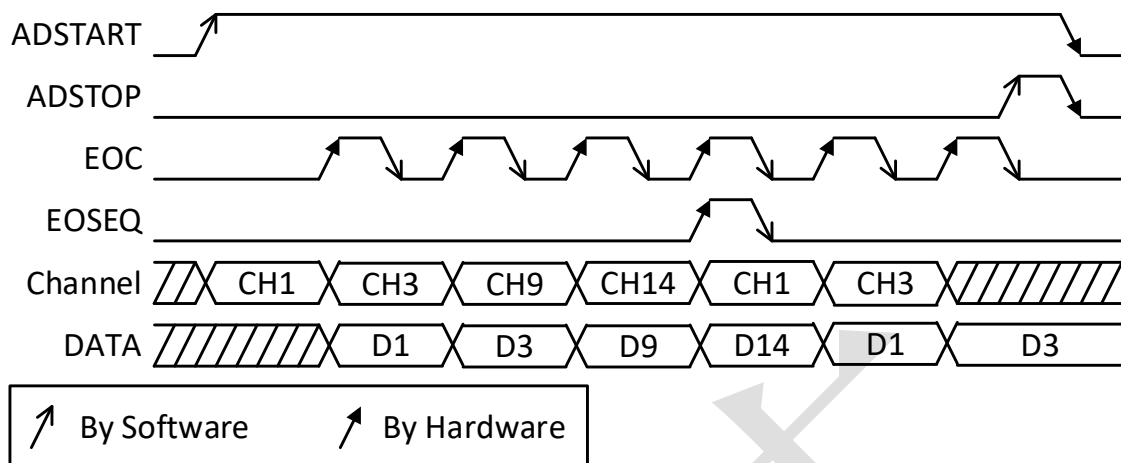


图 10-10 顺序采样下，连续序列转换，软件触发

注: $EXTEN=0x0$, $CHSEL=0x420A$, $WAIT=0$, $SCANDIR=0$, $CONT=1$, $CON=0$, $DISCEN=0$

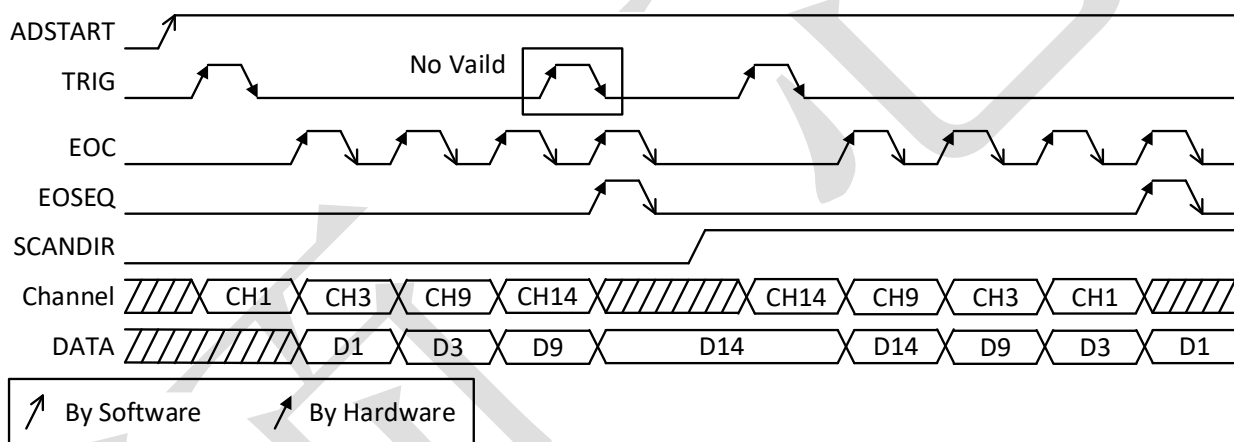


图 10-11 顺序采样下，单次序列转换，硬件触发

注: $EXTEN=0x1$ (上升沿), $CHSEL=0x420A$, $WAIT=0$, $CONT=0$, $CON=0$, $DISCEN=0$

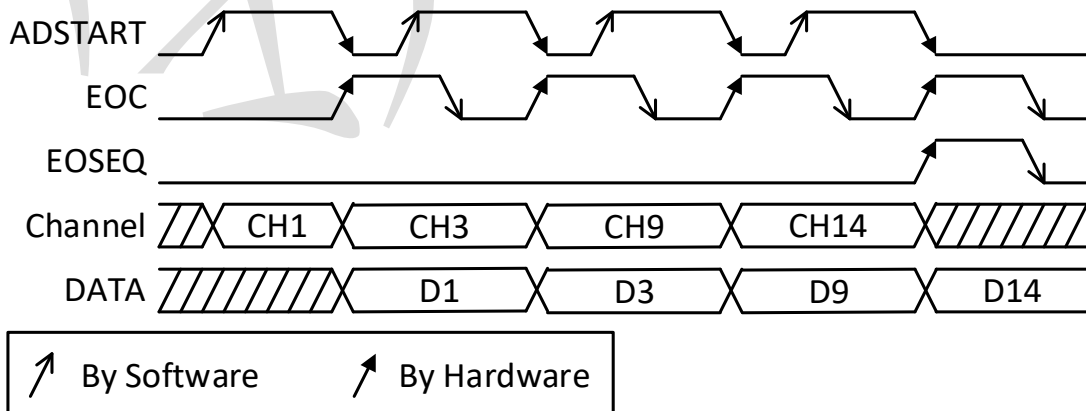


图 10-12 顺序采样下，断续序列转换，软件触发

注: $EXTEN=0x0$, $CHSEL=0x420A$, $SCANDIR=0$, $WAIT=0$, $CONT=0$, $CON=0$, $DISCEN=1$

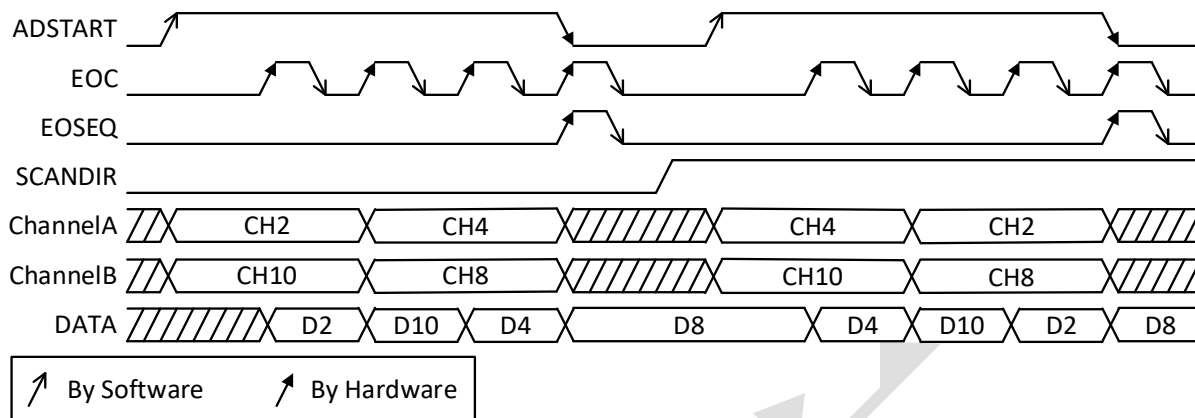


图 10-13 同时采样下，单序列转换，软件触发

注：EXTEN=0x0， SCANDIR=0， WAIT=0， CONT=0， CON=1， DISCEN=0

通道选择（B组缺少一个通道，在配对完毕后默认补齐CH8）

CHB	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8
	0	0	0	0	0	1	0	0
CHA	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
	0	0	0	1	0	1	0	0

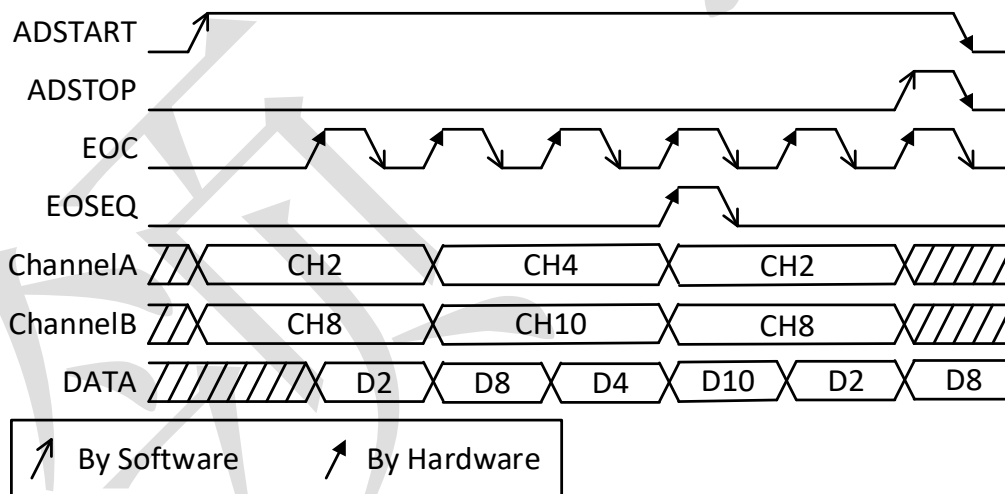


图 10-14 同时采样下，连续序列转换，软件触发

注：EXTEN=0x0， SCANDIR=0， WAIT=0， CONT=1， CON=1， DISCEN=0

通道选择

CHB	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8
	0	0	0	0	0	1	0	1
CHA	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
	0	0	0	1	0	1	0	0

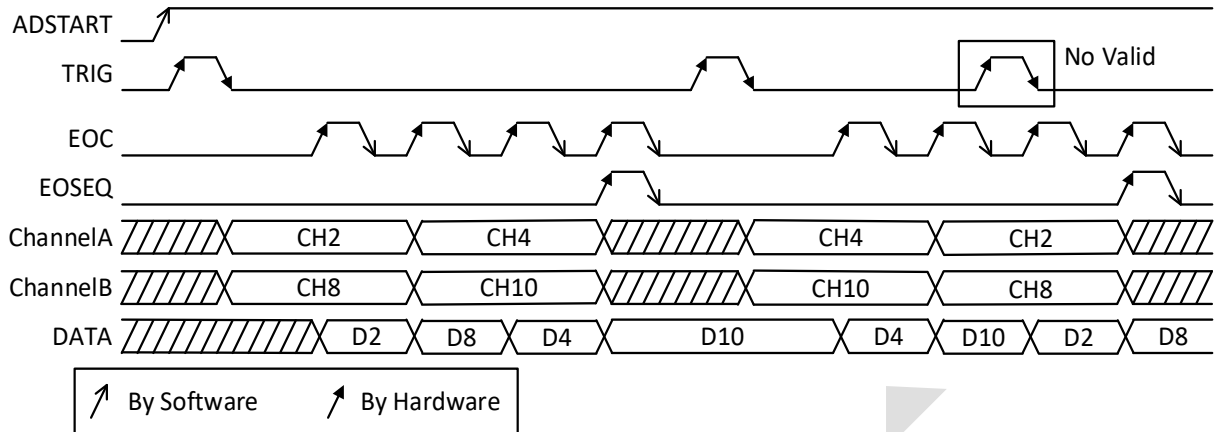


图 10-15 同时采样下，单次序列转换，硬件触发

注：EXTEN=0x2（下升沿），SCANDIR=0，WAIT=0，CONT=0，CON=1，DISCEN=1

通道选择

CHB	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8
	0	0	0	1	1	0	0	0
CHA	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
	0	1	0	0	0	0	0	0

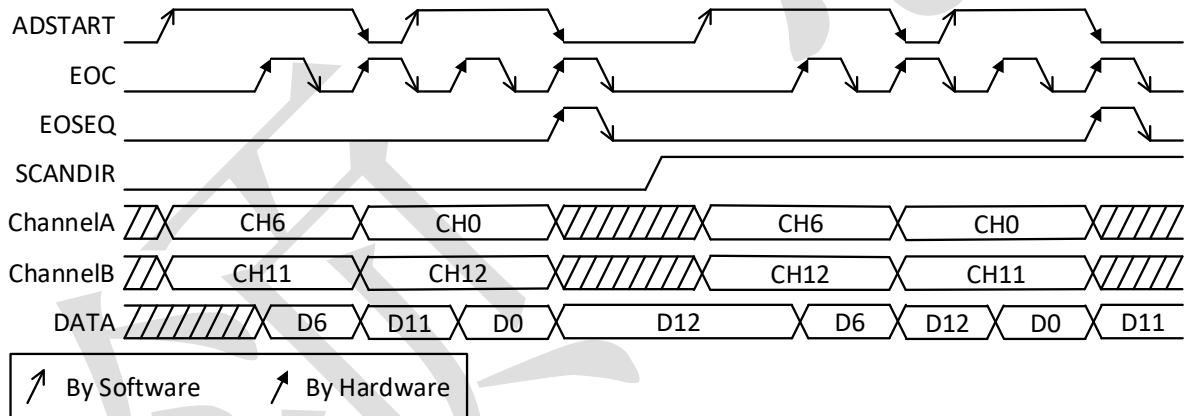


图 10-16 同时采样下，断续序列转换，软件触发

注：EXTEN=0x0，WAIT=0，CONT=0，CON=1，DISCEN=1

通道选择（A组缺少一个通道，在配对完毕后默认补齐CH0）

CHB	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8
	0	0	0	1	1	0	0	0
CHA	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
	0	1	0	0	0	1	0	0

注：内部温度传感器（ADIN_16，VSENSE）和内部参考电压（ADIN_17，VDD）仅支持单通道采样（CON=0）的三种转换模式。

10.4 ADC 转化的数据管理

10.4.1 数据寄存器与数据对齐

在每次转换结束时（当发生 EOC 事件时），转换后的数据结果存储在 16 位宽的 ADC_DR 数据寄存器中。

ADC_DR 的数据格式取决于配置的数据对齐。ADC_CFGR1.ALIGN 位选择转换后存储的数据的对齐方式。数据可以是右对齐（ALIGN=0）或左对齐（ALIGN=1）。

ALIGN	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0x0				DR[11:0]											
1	DR[11:0]												0x0			

10.4.2 FIFO

ADC 有一个 16 深度的 FIFO 缓冲，用来保存采样转换出来的数据。ADC_ISR.fifo_gap 为实时 FIFO 存储的数据量（只读）。ADC_ISR[12:8]位为 FIFO 内数据水平（16、12、8、4、0）中断标志位，当 FIFO 内的数据个数达到对应水平时硬件置位。如果开启对应中断使能位 ADC_IER[12:8]则还将产生中断，软件写 1 清零。

10.4.3 不使用 DMA 时管理转换数据

当 ADC 的采样转化速度过快，并且没有开启 DMA 来管理数据时，会出现 FIFO 内的数据没有及时被读走，导致 FIFO 满，进而不再接收 ADC 新转化出来的数据。为了满足多个通道转换应用对通道顺序的要求，此时 ADC 的采样通道会停止切换，直到 FIFO 内有空余位置接收新数据为止。为了保证在顺序采样和同时采样的正常工作，两种采样模式对重新开启通道切换有不同的要求：

- 顺序采样模式：当 FIFO 内数据水平达到 16 时，通道切换停止；数据水平小于 16 时，通道重新恢复正常切换工作
- 同步采样模式：当 FIFO 内数据水平达到 16 时，通道切换停止；数据水平小于 15 时，通道重新恢复正常切换工作

10.4.4 使用 DMA 时管理转换数据

DMA 模式开启时（ADC_CFGR1.DMAEN = 1），每次达到 FIFO 水平触发时都会产生一个 DMA 请求（ADC_CFGR1.DMACON 设置 FIFO 触发水平）。这样就允许 DMA 将 FIFO 内的转换数据搬运到软件指定的目标地址中。

表 10-5 DMA 水平触发配置表

MDACON[1:0]	0	1	2	3
FIFO 触发水平	>0	>=4	>=8	>=12

10.5 低功耗特性

10.5.1 自动延迟转换模式

自动延迟转换模式可用于在低速运行时简化软件以及优化应用程序的性能。

当在 ADC_CFGR1 寄存器中设置 WAIT 为 1 时，ADC 的采样转换会在 ADC 的 FIFO 缓冲满标志置位或者 ADC 使能 DMA 且发起了一次搬运请求时停止，在数据被读走（例如 ADC_DR 寄存器中的数据被读取或 EOC

标志已被清除)后新一次的采样转换才开始。这是一种自适应 ADC 速度和自适应系统读取 ADC 数据速度的方法。当正在转换中或自动延迟产生的情况下,任一硬件产生的触发或软件触发都会被忽略。

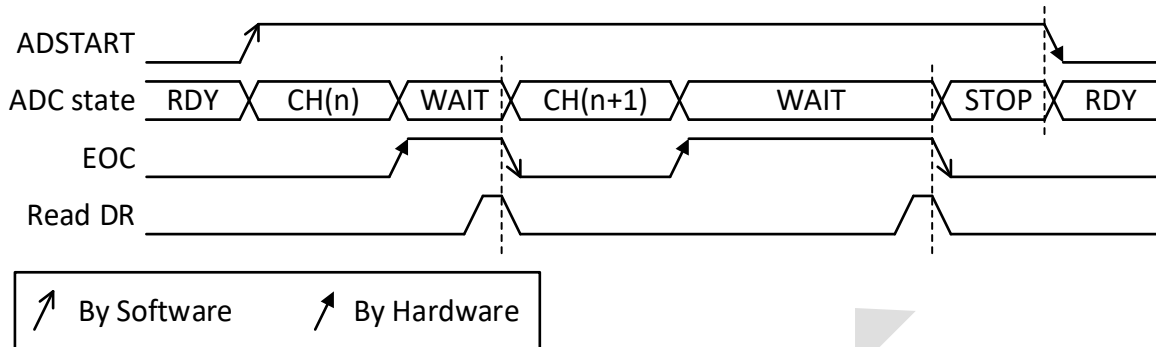


图 10-17 WAIT 模式下的转换图

10.5.2 低功耗模式

ADC 拥有低功耗管理特性,被称为低功耗模式,在能满足性能要求的情况下降低功耗。低功耗模式相关的两个配置位为 ADC_CFGR2.LP_EN 和 ADC_CFGR2.LP。两种低功耗模式具体如下:

1. 自动低功耗模式 (ADC_CFGR2.LP_EN=1, ADC_CFGR2.LP=0): ADC 在每次转换结束后自动进入低功耗模式。此模式适合时钟频率比较高,但转换并不要求很快,即可能好几十个时钟才需要转换一次。
2. 强制低功耗模式 (ADC_CFGR2.LP_EN=1, ADC_CFGR2.LP=1): ADC 直接进入低功耗模式。此模式适合时钟频率已经比较低,转换一次所要求的时钟数可以比较少(实际需要的时间依然是变长)。

10.6 ADC 模拟看门狗

AWD 模拟看门狗的功能由在 ADC_CFGR1 寄存器中的 AWDEN 位置位来开启。它可用于监控所选的单一通道或所有使能通道所配置电压范围(窗口),如图 10-18 所示。

如果 ADC 转换的模拟电压低于较低阈值或高于较高阈值,则设置 AWD 模拟看门狗状态位。这些阈值在 16 位寄存器 ADC_TR.HT 和 ADC_TR.LT 的 12 个最低有效位中编程。通过在 ADC_IER 寄存器中设置 AWDIE 位可以启用中断。

AWD 标志位可用软件写 1 来清除。

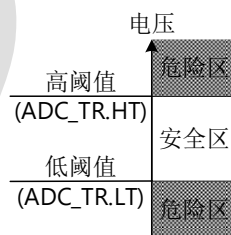


图 10-18 模拟看门狗监控区域

表 10-6 看门狗通道选择

模拟看门狗监控的通道	AWDSGL 位	AWDEN 位
无通道	x	0
所有通道	0	1

单通道	1	1
-----	---	---

注：AWDCH[3:0]选择AWD监控的单通道

10.7 温度传感器及内部参考电压通道转换

温度传感器可以用来测量器件的结点温度 (T_J)。温度传感器内部连接到 ADC_IN16 输入通道, 可用于转换传感器的电压值到一个数值。必须设置 ST 和 SV 位来激活两内部通道: ADC_IN16(温度传感器)和 ADC_IN17 (VDD)。当 ST=1, SV=0 时选用 ADC_IN16, 其他通道 (ADC_IN0~15, ADC_IN17) 禁用。当 ST=1, SV=1 时选用 ADC_IN17 (VDD), 其他通道 (ADC_IN0~15, ADC_IN16) 禁用。

注: 传感器从断电模式下唤醒时到能正确输出 VSENSE 的值要有一个启动时间, ADC 从上电后启动也有一个启动时间, 若要减少这个延时, 则需要在同一时间设置 ADEN 和 ST/SV 位。

下图为 ADC 参考源为内部 3.3V 情况下的 VTS 温度曲线图。

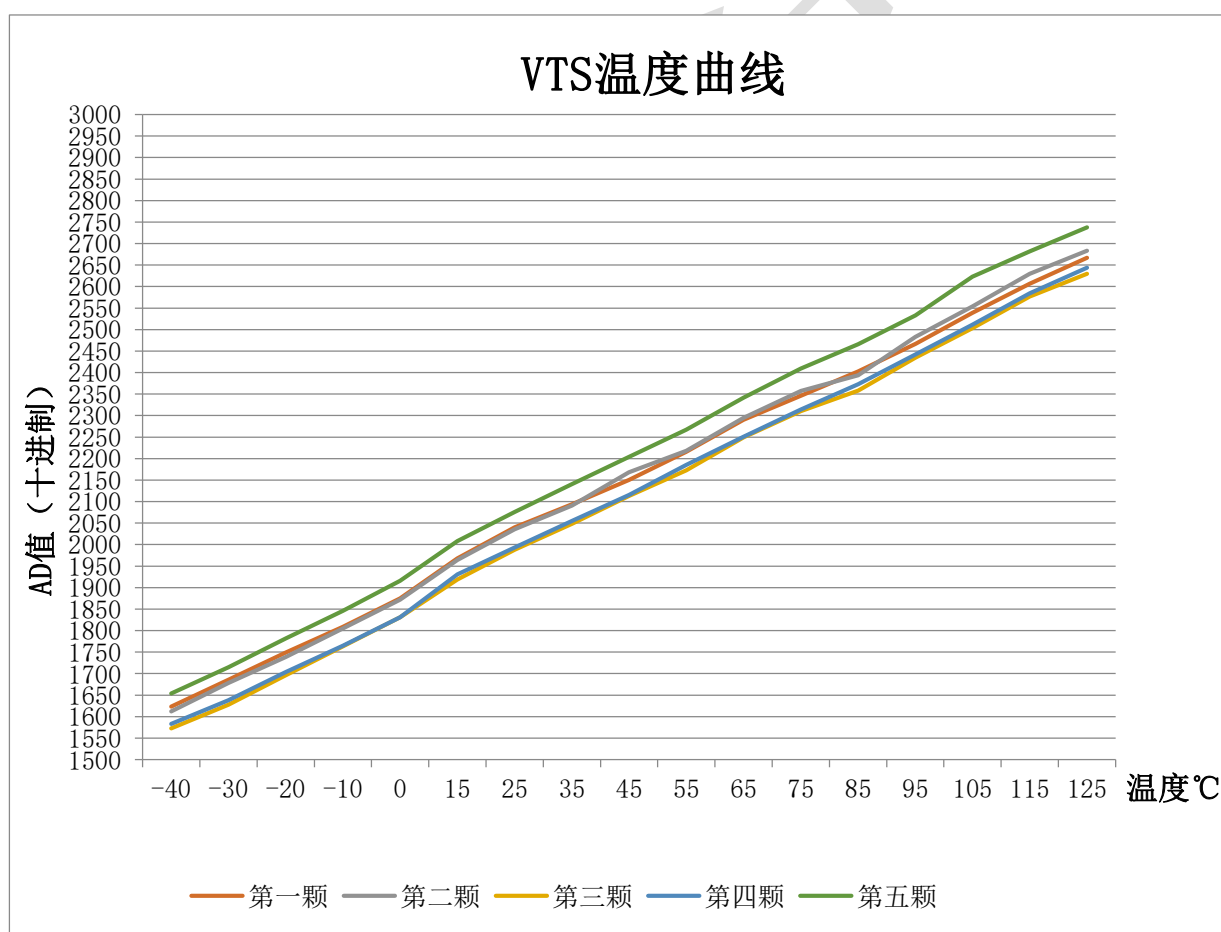


图 10-19 VTS 温度曲线图

芯片出厂时会进行温度标定, 此值会存入信息区中:

- 地址: 0x1FFF_F580 为常温温度 (例如 20.8°C, 存入数据为 208 (十进制));
- 地址: 0x1FFF_F584 为高温温度 (例如 65°C, 存入数据为 650 (十进制));
- 地址: 0x1FFF_F588 为常温下以 3.3V 和 4V 做参考的 VTS 采样 AD 值 (例如 0x864F_07E4 表示 3.3V 做参考时的采样值为 0x07E4 (低 16 位); 4V 做参考时的采样值为 0x64F (高 12 位, 最高位存放测试标志位, 无实际意义);

- 地址：0x1FFF_F58C 为高温下以 3.3V 和 4V 做参考的 VTS 采样 AD 值（数据意义同上）；
- 地址：0x1FFF_F590 为常温下以 2.5V 和 5V 做参考的 VTS 采样 AD 值（低 16 位表示 2.5V 位参考时采样 VTS 的 AD 值，高 12 位表示 5V 做参考时采样 VTS 的 AD 值）；
- 地址：0x1FFF_F594 为高温下以 2.5V 和 5V 做参考的 VTS 采样 AD 值（数据意义同上）；

实测发现 VTS 采样温度与 AD 值为线性关系。为保证 VTS 精度，针对每一种参考源增加高温与常温的标定值，用于计算 VTS 实时温度。保存的温度与 AD 值存在对应关系。根据其线性关系：

- T_{hADC} 表示高温下保存 AD 值（根据参考源的需要去对应的地址取值）；
- T_{nADC} 表示常温下保存 AD 值（根据参考源的需要去对应的地址取值）；
- T_h 表示高温温度值的 10 倍（标定值保存在 0x1FFF_F584）；
- T_n 表示常温温度值的 10 倍（标定值保存在 0x1FFF_F580）；

当前参考源下采样 VTS 的值 C，斜率 $K = (T_{hADC} - T_{nADC}) / (T_h - T_n)$ ，则当前温度 $T = ((C - T_{hADC}) / K + T_h) / 10$ 。

注：可参考库函数 `ADC_GetVTStemperature()`；因为计算过程存在负数，变量需要定义为 `int` 型。

10.8 ADC 中断

ADC 中断可以由以下任何一个事件产生：

- ADC 升级，当 ADC 准备好（ADRDY 标志）
- 任何转换结束（EOC 标志）
- 转换序列结束（EOSEQ 标志）
- 当采样阶段结束时（EOSMP 标志）
- 当发生模拟看门狗检测（AWD 标志）
- 当 FIFO 达到配置水平时：
 - FIFO 空（0 个数据在 FIFO 中，请谨慎使用）
 - FIFO 几乎为空（4 个数据在 FIFO）
 - FIFO 半满（8 个数据在 FIFO）
 - FIFO 几乎满（12 数据在 FIFO）
 - FIFO 全满（16 个数据在 FIFO）

每个中断标志位都有自己的使能位控制，可灵活配置 ADC 中断。

表 10-7 ADC 中断表

中断事件	事件标志	使能控制位
ADC 准备好	ADRDY	ADRDYIE
转换结束	EOC	EOCIE
序列转换结束	EOSEQ	EOSEQIE
模拟看门狗检测	AWD	AWDIE
采样阶段结束	EOSMP	EOSMPIE
FIFO 空，无数据	FE	FEIE
FIFO 有 4 个数据	FAE	FAEIE
FIFO 有 8 个数据	FHF	FHFIE
FIFO 有 12 个数据	FAF	FAFIE
FIFO 有 16 个数据	FF	FFIE

10.9 ADC 寄存器描述

表 10-8 ADC 相关寄存器表

名称	说明	读写权限	复位值	字节地址
ADC_ISR	中断状态寄存器	R/W	0x0000_0000	0x4001_A000
ADC_IER	中断使能寄存器	R/W	0x0000_0000	0x4001_A004
ADC_CR	控制寄存器	R/W	0x0000_0000	0x4001_A008
ADC_CFGR1	配置寄存器 1	R/W	0x0000_0000	0x4001_A00C
ADC_CFGR2	配置寄存器 2	R/W	0x0000_0000	0x4001_A010
ADC_SMPR	采样周期寄存器	R/W	0x0000_0000	0x4001_A014
ADC_TR	阈值寄存器	R/W	0x0FFF_0000	0x4001_A018
ADC_CHSELR	通道选择寄存器	R/W	0x0000_0000	0x4001_A01C
ADC_DR	数据寄存器	R	0x0000_XXXX	0x4001_A020
ADC_TRIM	调修寄存器	R/W	0x0097_00A3	0x4001_A024
ADC_EXTCFG	外部配置寄存器	R/W	0x0000_0000	0x4001_A028
ADC_TESTCFG	测试寄存器	R/W	0x0000_78DC	0x4001_A02C

注：x 表示不确定；-表示未实现；u 表示不受复位影响；c 表示该位复位值因复位源而异；R 只读；W 只写；R-0 表示只读且读 0；R/W 表示可读写（以后章节同上述）。

只有当 ADSTART=0 时（确定无进行中的转换）才允许改写这些寄存器：IER、CFGR1、CFGR2、SMPR、TR、CHSELR、TRIM、EXTCFG 和 TESTCFG。

10.9.1 ADC 中断状态寄存器（ADC_ISR）

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
Res.											FIFO_GAP[4:0]				
											r				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			FF	FAF	FHF	FAE	FE	AWD	Res.			EOSEQ	EOC	EOSMP	ADRDY
			rw	rw	rw	rw	rw	rw				rw	rw	rw	rw

Bits	31:21	保留，必须保持复位值
Bits	20:16	FIFO_GAP[4:0] : FIFO 内的实时数据量
Bits	15:13	保留，必须保持复位值
Bit	12	FF (FIFO_FULL) : FIFO 缓冲全满标志 当 FIFO 内存满 16 个数据，该位由硬件置位。（软件对该位写 1 清除）
Bit	11	FAF (FIFO_ALMOST_FULL) : FIFO 缓冲半全满中断标志 当 FIFO 内存满 12 个数据，该位由硬件置位。（软件对该位写 1 清除）
Bit	10	FHF (FIFO_HALF_FULL) : FIFO 缓冲半满标志 当 FIFO 内存满 8 个数据，该位由硬件置位。（软件对该位写 1 清除）
Bit	9	FAE (FIFO_ALMOST_EMPTY) : FIFO 缓冲半空中断标志

	当 FIFO 内存满 4 个数据，该位由硬件置位。（软件对该位写 1 清除）
Bit 8	FE (FIFO_EMPTY): FIFO 缓冲空标志
	当 FIFO 内没有数据，该位由硬件置位。（软件对该位写 1 清除）
Bit 7	AWD: 模拟看门狗标志
	当转换后的电压超过阈值 (ADC_TR) 时，该位由硬件置位。（软件对该位写 1 清除）
Bits 6:4	保留，必须保持复位值
Bit 3	EOSEQ: 序列转换结束标志
	所选的通道序列转换结束后，该位由硬件置位。（软件对该位写 1 清除）
Bit 2	EOC: 转换结束标志
	当每个通道新转换结果有效时（已经存放在 ADC_DR 中），该位由硬件置位。
	软件对该位写 1 清除或读取 ADC_DR 寄存器来清除
Bit 1	EOSMP: 采样结束标志
	在转换期间的采样阶段结束时，该位由硬件置位。（软件对该位写 1 清除）
Bit 0	ADRDY: ADC 准备完成标志
	在 ADC 上电 (ADEN=1) 稳定结束后，该位由硬件置位。（软件对该位写 1 清除）

10.9.2 ADC 中断使能寄存器 (ADC_IER)

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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			FFIE	FAFIE	FHFIE	FAEIE	FEIE	AWD IE	Res.			EOSIE	EOCIE	EOSM PIE	ADRD YIE
			rw	rw	rw	rw	rw	rw				rw	rw	rw	rw

Bits 31:13	保留，必须保持复位值
Bit 12	FFIE: FIFO 缓冲全满中断使能
	0: 禁止 FIFO 缓冲全满中断（默认）
	1: 使能 FIFO 缓冲全满中断
Bit 11	FAFIE: FIFO 缓冲半全满中断使能
	0: 禁止 FIFO 缓冲半全满中断（默认）
	1: 使能 FIFO 缓冲半全满中断
Bit 10	FHFIE: FIFO 缓冲半满中断使能
	0: 禁止 FIFO 缓冲半满中断（默认）
	1: 使能 FIFO 缓冲半满中断
Bit 9	FAEIE: FIFO 缓冲半空中断使能
	0: 禁止 FIFO 缓冲半空中断（默认）
	1: 使能 FIFO 缓冲半空中断
Bit 8	FEIE: FIFO 缓冲空中断使能

	0: 禁止 FIFO 缓冲空中断 (默认)
	1: 使能 FIFO 缓冲空中断
Bit 7	AWDIE : 模拟看门狗中断使能
	0: 禁止模拟看门狗中断 (默认)
	1: 使能模拟看门狗中断
Bits 6:4	保留, 必须保持复位值
Bit 3	EOSIE : 序列转换结束中断使能
	0: 禁止 EOSEQ 中断 (默认)
	1: 使能 EOSEQ 中断
Bit 2	EOCIE : 转换结束中断使能
	0: 禁止 EOC 中断 (默认)
	1: 使能 EOC 中断
Bit 1	EOSMPIE : 采样结束中断使能
	0: 禁止 EOSMP 中断 (默认)
	1: 使能 EOSMP 中断
Bit 0	ADRDYIE : ADC 准备好中断使能
	0: 禁止 ADRDY 中断 (默认)
	1: 使能 ADRDY 中断

10.9.3 ADC 控制寄存器 (ADC_CR)

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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.											AD STOP	Res.	AD START	ADDIS	ADEN
											rw				

Bits 31:5	保留, 必须保持复位值
Bit 4	ADSTOP : ADC 停止转换命令, (软件置位, 硬件清零)
	当转换停止结束时, 该位由硬件清零且 ADC 已准备好接受新的命令 (新一次 ADC 转化开始, 或者 ADDIS 来关断禁止 ADC)
	写 0 操作无效; 读 0 表示无 ADSTOP 命令正在执行
	写 1 停止 ADC 和丢弃正在进行中的转换; 读 1 表示 ADSTOP 命令执行中
Bit 3	保留, 必须保持复位值
Bit 2	ADSTART : ADC 开始转换命令
	软件触发: 软件置 1 启动一次 ADC 转换, 完成转换后自动硬件清零
	硬件触发: 软件置 1, 等待硬件触发信号后启动一次 ADC 转换 (不会被硬件清零)
	当执行完 ADSTP 命令后, ADSTART 和 ADSTOP 位同时硬件清零;

	读 1 表示 ADC 正在进行转换；读 0 则反之。
Bit 1	ADDIS : ADC 禁止命令（软件置位，硬件清零）
	写 0 无效；读 0 表示无 ADDIS 命令正在执行
	写 1 为关断 ADC，进入掉电状态；读 1 表示 ADDIS 命令执行中
Bit 0	ADEN : ADC 启动命令（软件置位，硬件清零）
	写 0 无效；读 0 表示 ADC 处于关断状态（掉电状态）
	写 1 使能 ADC；读 1 表示 ADC 已经启动
注 1 : 当 ADSTART=1 和 ADDIS=0（ADC 开启且可能正在转换，但无 ADC 关断请求）软件才能置位 ADSTOP	
注 2 : 当 ADEN=1 和 ADDIS=0（ADC 开启且可能正在转换，但无 ADC 关断请求）软件才能置位 ADSTART	
注 3 : 当 ADEN=1 和 ADSTART=0（确定 ADC 不在转换中）软件才能置位 ADDIS 请求关断	
注 4 : 当 ADC_CR 寄存器所有位为 0（ADC 处于关断状态）软件才能置位 ADEN	
注 5 : 当 ADSTART=1（表示 ADC 正在转换中），除了 CR 和 IER 寄存器可以软件配置，其它写操作均无效	

10.9.4 ADC 配置寄存器 1（ADC_CFGR1）

地址偏移：0x0C

复位值：0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	AWDCH[3:0]			Res.			AWD EN	AWD SGL	Res.				DISC EN		
	rw						rw	rw							rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	WAIT	CONT	Res.						ALIGN	DMAON[1:0]	SCANDIR	Res.	DMAEN		
	rw	rw								rw	rw	rw			rw

Bit 31	保留，必须保持复位值
Bits 30:27	AWDCH[3:0] : 模拟看门狗通道选择位
	这些位由软件设置和清除。它们设置模拟看门狗监视的输入通道
	0000: 模拟看门狗监视的 ADC 模拟输入通道 0（默认）
	0001: 模拟看门狗监视的 ADC 模拟输入通道 1
	...
	1111: 模拟看门狗监视的 ADC 模拟输入通道 15
Bits 26:24	保留，必须保持复位值
Bit 23	AWDEN : 模拟看门狗使能位
	0: 禁止模拟看门狗（默认）
	1: 使能模拟看门狗
Bit 22	AWDSGL : 在单一通道或所有通道使能看门狗
	0: 在所有通道上使能模拟看门狗（默认）
	1: 在单一通道上使能模拟看门狗，通道选择为 AWDCH[3:0]
Bits 21:17	保留，必须保持复位值
Bit 16	DISCEN : 断续模式
	0: 禁止断续模式（默认）

	1: 使能断续模式
	注: 不可能同时使能断续模式和连续模式。在这种情况下, ADC 认为连续模式禁止。
Bit 15	保留, 必须保持复位值
Bit 14	WAIT: 自动延迟转换模式
	0: 禁止自动延迟转换模式 (默认)
	1: 使能自动延迟转换模式
Bit 13	CONT: 单次/连续转换模式选择
	0: 单次转换模式 (默认)
	1: 连续转换模式
Bits 12:6	保留, 必须保持复位值
Bit 5	ALIGN: 数据对齐方式
	0: 右对齐, 高位补零 (默认)
	1: 左对齐, 低位补零
Bits 4:3	DMACON[1:0]: DMA 配置
	00: FIFO 内有数据时, 产生一次 DMA 请求 (默认)
	01: FIFO 内有 4 个数据或以上时, 产生一次 DMA 请求
	10: FIFO 内有 8 个数据或以上时, 产生一次 DMA 请求
	11: FIFO 内有 12 个数据或以上时, 产生一次 DMA 请求
Bit 2	SCANDIR: 扫描序列方向
	该位由软件设置和清除, 以选择在序列中扫描通道的方向。
	0: 向上扫描 (从 CHSEL0 到 CHSEL15) (默认)
	1: 向下扫描 (从 CHSEL15 到 CHSEL0)
Bit 1	保留, 必须保持复位值
Bit 0	DMAEN: DMA 使能位
	允许用 DMA 控制器来自动管理转换的结果数据
	0: 禁止 DMA (默认)
	1: 使能 DMA

10.9.5 ADC 配置寄存器 2 (ADC_CFGR2)

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
CKMODE[4:0]					Res.										
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					LP	LP	Res.					PWR_	PULSE	EXT_	CON
					_EN							SET		REF	
					rw	rw						rw	rw	rw	rw

Bit 31	CKMODE[4]: ADC 工作时钟源选择
	0: 同步时钟 PCLK1 (APB1 总线时钟) (默认)

	1: 异步时钟 PLL_CLK
Bits 30:27	CKMODE[3:0]: 时钟分频系数
	0001 PCLK1/2 PLL_CLK/2
	0010 PCLK1/4 PLL_CLK/4
	0011 PCLK1/6 PLL_CLK/6
	0100 PCLK1/8 PLL_CLK/8
	0101 PCLK1/10 PLL_CLK/10
	0110 PCLK1/12 PLL_CLK/12
	0111 PCLK1/16 PLL_CLK/16
	1000 PCLK1 PLL_CLK/32
	1001 PCLK1 PLL_CLK/64
	1010 PCLK1 PLL_CLK/128
	1011 PCLK1 PLL_CLK/256
	其他 PCLK1 PLL_CLK/256
Bits 26:11	保留, 必须保持复位值
Bit 10	LP_EN: 低功耗模式使能位
	0: 禁止低功耗模式 (默认)
	1: 使能低功耗模式
Bit 9	LP: 低功耗模式选择
	LP_EN =1 且 LP=0, 进入自动低功耗模式 (默认)
	LP_EN =1 且 LP=1, 进入强制低功耗模式
Bits 8:4	保留, 必须保持复位值
Bit 3	PWR_SET: 内部参考电压选择
	0: 3.3V (默认)
	1: 5.0V
Bit 2	PULSE: 内部信号脉冲宽度选择
	0: 2*ADC_CLK (默认)
	1: 3*ADC_CLK
	注: 当 PULSE=0 时, 最小采样周期为 16 ADC_CLK; 当 PULSE=1 时, 最小的采样周期为 17 ADC_CLK
Bit 1	EXT_REF: ADC 参考电压选择
	0: 使用内部参考电压 (ADC 模块内部产生, 由 PWR_SET 配置) (默认)
	1: 使用外部参考电压 (由 ChipCtrl_PWR.ADPREF[1:0]配置)
Bit 0	CON: 采样模式选择
	0: 顺序模式/单采样 (在一个采样转换周期内 1 个通道被采样) (默认)
	1: 同步模式/双采样 (2 个通道同时被采样)

10.9.6 ADC 采样周期寄存器 (ADC_SMPR)

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
Res.															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				CONVERT[4:0]				ADJUST[2:0]			Res.	SMP[2:0]			
				rw				rw				rw			

Bits	31:12	保留，必须保持复位值
Bits	11:7	CONVERT[4:0] : 采样周期设定
		顺序采样模式 (CON = 0): 采样转化周期数 $(15 + \text{CONVERT}[4:0]) * \text{ADC_CLK}$
		同时采样模式 (CON = 1): 采样转化周期数 $(24 + \text{CONVERT}[4:0]) * \text{ADC_CLK}$
Bits	6:4	ADJUST[2:0] : AD 上电调整时钟脉冲宽度设定
		<i>注: 建议该位配置为 111。</i>
Bit	3	保留，必须保持复位值
Bits	2:0	SMP[2:0] : AD 采样信号脉冲宽度设定，采样脉冲宽度为 $(\text{SMP}[2:0]+1) * \text{ADC_CLK}$

10.9.7 ADC 阈值寄存器 (ADC_TR)

地址偏移: 0x18

复位值: 0x0FFF_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				HT[11:0]											
				rw											

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

Bits	31:28	保留，必须保持复位值
Bits	27:16	HT[11:0] : 模拟看门狗的高阈值
		这些位由软件配置，用来定义模拟看门狗的高阈值
Bits	15:12	保留，必须保持复位值
Bits	11:0	LT[11:0] : 模拟看门狗的low阈值
		这些位由软件配置，用来定义模拟看门狗的low阈值

10.9.8 ADC 通道选择寄存器 (ADC_CHSELR)

地址偏移: 0x1C

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.														ST	SV
														rw	rw

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

Bits 31:18	保留，必须保持复位值
Bit 17	ST : 温度传感器测量选择，与 SV 配合使用
Bit 16	SV : 模拟电源电压测量选择，与 ST 配合使用
Bits 15:0	CHSELR[15:0] : 这些位可由软件改写，用来定义所要转换序列的通道
	0: 输入通道不被选为转换通道（默认）
	1: 输入通道被选为转换通道

外部通道对应外部 I/O:															
通道组 A								通道组 B							
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PA0	PA1	PA2	PA3	PA4	PA5	PA6	PA7	PB0	PB1	PA8	PF6	PF7	PB5	PA9	PA10

ST	SV	ADC 转换对象
0	X	I/O（通道组 A 和 B）
1	0	温度传感器
1	1	VDDA

10.9.9 ADC 数据寄存器（ADC_DR）

地址偏移: 0x20

复位值: 0x0000_XXXX

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

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

Bits 31:16	保留，必须保持复位值
Bits 15:0	DATA[15:0] : 数据（转换结果值）
	该数据左对齐或右对齐数据格式；每次读走一个，FIFO 内数据减少一个

10.9.10 ADC 调修寄存器（ADC_TRIM）

地址偏移: 0x24

复位值: 0x0097_00A3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.								VOS_TRIM[7:0]							
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								GAIN_TRIM[7:0]							
r															

Bits 31:24	保留，必须保持复位值
Bits 23:16	VOS_TRIM[7:0]: ADC 偏移修调
	默认值为 10010111
Bits 15:8	保留，必须保持复位值
Bits 7:0	GAIN_TRIM[7:0]: ADC 增益修调
	默认值为 10100011
注: VOS_TRIM 和 GAIN_TRIM 在复位完成后，自动载入存储于相应信息区的校准值。	

10.9.11 ADC 外部配置寄存器 (ADC_EXTCFG)

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
EXTSE[4:0]					EXTSP[4:0]					EXTSELG4[1:0]		EXTSELG3[1:0]		EXTSELG2[1:0]	
rw					rw					rw		rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTSELG1[1:0]		EXTSELG0[1:0]		EXTDTG[7:0]								OP	EDGE	EXTEN[1:0]	
rw		rw		rw								rw	rw	rw	

Bits 31:27	EXTSE[4:0]: 5 组触发的使能位，每一位对应一组触发源
	00000: 禁止 (默认)
	1xxxx: 使能 EXTSELG4
	x1xxx: 使能 EXTSELG3
	xx1xx: 使能 EXTSELG2
	xxx1x: 使能 EXTSELG1
	xxxx1: 使能 EXTSELG0
Bits 26:22	EXTSP[4:0]: 5 组触发源极性选择，每一位对应一组触发源
	0: 同相输入 (极性不变) (默认)
	1: 反相输入 (极性反转)
Bits 21:20	EXTSELG4[1:0]: 触发源 4 选择
	00: TIM1_TRGO (默认)
	01: TIM3_TRGO
	10: TIM15_TRGO
	11: 保留
Bits 19:18	EXTSELG3[1:0]: 触发源 3 选择
	00: TIM1_OC4 (默认)
	01: TIM3_OC4
	10: TIM17_OC1
	11: ACMP0
Bits 17:16	EXTSELG2[1:0]: 触发源 2 选择
	00: TIM1_OC3 (默认)

	01: TIM3_OC3
	10: TIM16_OC1
	11: EXTI11
Bits 15:14	EXTSELG1[1:0]: 触发源 1 选择
	00: TIM1_OC2 (默认)
	01: TIM3_OC2
	10: TIM15_OC2
	11: EXTI3
Bits 13:12	EXTSELG0[1:0]: 触发源 0 选择
	00: TIM1_OC1 (默认)
	01: TIM3_OC1
	10: TIM15_OC1
	11: EXTI1
Bits 11:4	EXTDTG[7:0]: 外部触发时延
	延迟模式选择 EXTDTG[7:5]: T_d (延迟时间) 计算公式
	0XX: $T_d = \text{EXTDTG}[6:0] * \text{PCLK}$ (默认)
	10X: $T_d = (\text{EXTDTG}[5:0] + 64) * 2 * \text{PCLK}$
	110: $T_d = (\text{EXTDTG}[4:0] + 32) * 8 * \text{PCLK}$
	111: $T_d = (\text{EXTDTG}[4:0] + 32) * 16 * \text{PCLK}$
Bit 3	OP: ADC 触发源操作模式
	0: 所有组触发信号逻辑“或”处理 (默认)
	1: 所有组触发信号逻辑“与”处理
Bit 2	EDGE: ADC 触发信号模式
	0: 边沿模式 (先进行边沿检测再 OP 处理) (默认)
	1: 电平模式 (先 OP 处理再进行边沿检测)
Bits 1:0	EXTEN[1:0]: ADC 外部硬件触发使能
	00: 禁止外部硬件触发, 仅支持软件触发 (默认)
	01: 外部硬件上升沿触发
	10: 外部硬件下降沿触发
	11: 外部硬件双沿触发

10.9.12 ADC 测试寄存器 (ADC_TESTCFG)

地址偏移: 0x2C

复位值: 0x0000_78DC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	PD_ ADC	PD_ REF	PD_ SH2	PD_ SH1	RMD	RTA	RT[2:0]			RI[3:0]			TT	TEST	
	rw	rw	rw	rw	rw	rw	rw			rw			rw	rw	



Bits	31:15	保留，必须保持复位值
Bit	14	PD_ADC: ADC 内核控制位
		0: ADC 内核上电
		1: ADC 内核掉电（默认）
Bit	13	PD_REF: 参考电压模块控制位
		0: 使能参考电压模块
		1: 关闭参考电压模块（默认）
Bit	12	PD_SH2: 采样/保持电路 2 控制位
		0: 使能采样/保持电路 2
		1: 关闭采样/保持电路 2（默认）
Bit	11	PD_SH1: 采样/保持电路 1 控制位
		0: 使能采样/保持电路 1
		1: 关闭采样/保持电路 1（默认）
Bit	10	RMD: 默认值为 0，必须保持默认值
Bit	9	RTA: 默认值为 0，必须保持默认值
Bits	8:6	RT[2:0]: 默认值为 011，必须保持默认值
Bits	5:2	RI[3:0]: 默认值为 0111，必须保持默认值
Bit	1	TT: 默认值为 0，必须保持默认值
Bit	0	TEST: 仅做测试时使用，必须保持默认值

11. 数模转换器（DAC）、模拟比较器（ACMP）和反向电动势采样控制器（HALL_MID）

LCM32F037 包含 2 个 10 位 DAC 模块，可用于将输入的 2 路数字信号转换成 2 个模拟电压输出到 I/O 或者 3 个模拟比较器的输入。每个 DAC 参考电压可选为 2.5V、4V 或 VDDA。每个 DAC 都支持硬件触发和 DMA 传输功能。每个 DAC 都支持硬件产生可配置的伪随机噪声波形和三角波形。

LCM32F037 包含 3 个快速的轨到轨模拟比较器，比较器的输入来自内部 DAC 输出、HALL_MID 输出或者外部端口。比较器的外部触发、迟滞、速度、滤波、极性和消隐都可以通过软件配置。所有的比较器都可以产生中断，支持将系统从停机模式唤醒。所有的比较器都可以与定时器联动。两个比较器可以组合成一个窗口比较器。

LCM32F037 包含 1 个反向电动势采样控制器（HALL_MID）模块，可以和 DAC、ACMP 联动工作。

11.1 主要特性

11.1.1 DAC 特性

- 响应速度高达 20MHz
- 电平输出范围：0~VREF
- 参考电压 VREF 可选择芯片电源模块输出的 DAC_VREF（2.5 或 4.0V），或者内部 VDDA
- 消耗电流低：在 5V 参考电压下，每个 DAC 电流消耗不超过 150uA
- 电源抑制比（PSRR），1KHz 最差条件下达到-80dB，1MHz 最差条件下达到-30dB
- 积分非线性度（INL）及差分非线性度（DNL）在 0.5LSB 以内
- DAC 输出具有增强驱动 BUFF，可连接到比较器输入、ADC 输入或外部 I/O
- 接入 DMA 通道，配置后可以连续转换，具有 DMA 速度匹配检测及相关中断功能
- 内置波形发生器，可以产生三角波、噪声波等，幅度或者振幅可调
- 可选择多种触发方式，包括软硬件触发以及与其他模块的联动
- DAC 模块通过使能位来控制，关断时无电流消耗

表 11-1 DAC 管脚

名称	信号类型	备注
VDDA	输入电源	外部连接到 VDDA 电源，1.8-5.5V
VSSA	输入地	直接连接到外部 VSSA 脚
VREF	输入参考电压	可选 2.5、4.0V 或 VDDA
DAC_OUT	模拟输出信号	输出可以到管脚或者比较器

DAC 框图如下所示:

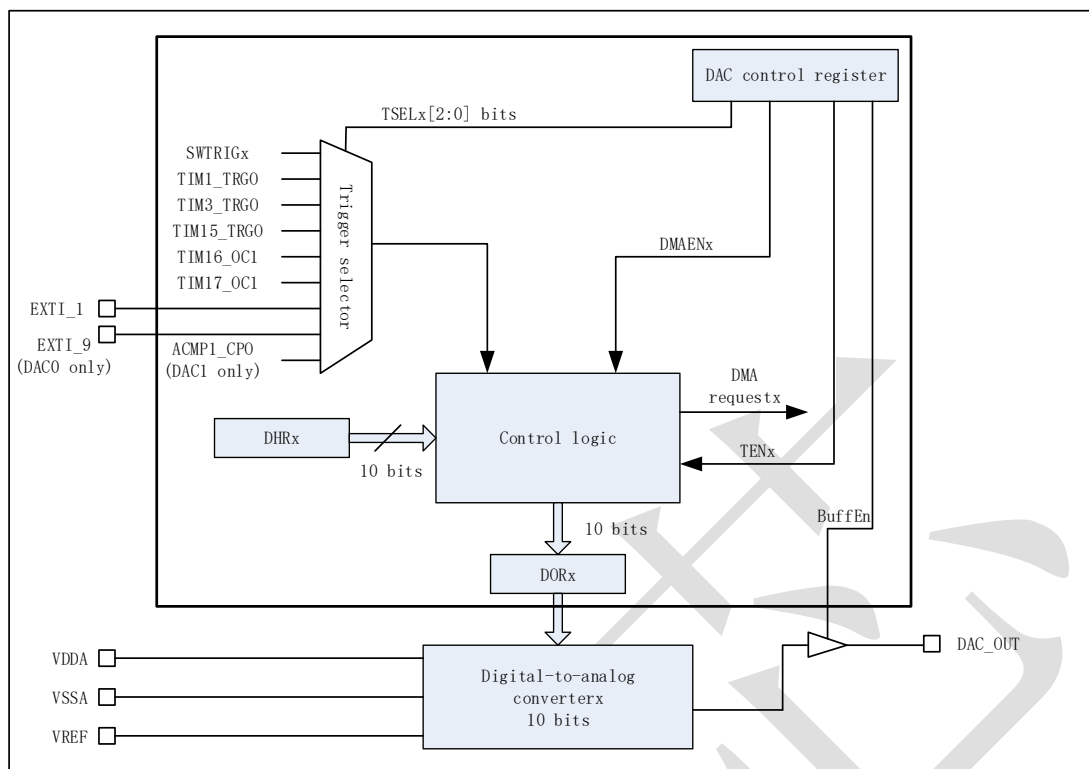


图 11-1 DAC 框图

11.1.2 ACMP 特性

- 集成 3 个模拟比较器，均可单独配置其工作模式
- 轨到轨工作电压范围
- 低输入失调电压，且可以通过内部精确校准
- 可选模拟输出迟滞，可调输出延迟，可选输出滤波
- 最低响应速度 84ns (>10MHz)
- 输出极性可调整
- 支持消隐功能
- 多种联动方式，包括跟 TIMER、ADC 模块等进行联动
- 输入可选，包括 DAC 输出、外部 I/O 以及 HALL_MID 模块输出等
- 可配置作为芯片低功耗模式下的唤醒发起源

11.2 功能描述

11.2.1 DAC 功能描述

LCM32F037 内置两个 10 位高精度 DAC，对应 DAC0/DAC1，响应速度为 20MSPS。

11.2.1.1 使能配置

每个 DAC 的使能通过寄存器 DACx_CR.EN 位来配置，在 DAC 使能之后，触发使能 DAC_TEN、触发选择 DAC_TSEL、波形选择 WAVE 和波形幅值配置 MAMP 寄存器不能修改。

DAC 的输出信号为 DAC_OUT，输出时需要通过配置寄存器 ANA_CTRL.DACx_BUFEN 打开驱动 BUFF。

11.2.1.2 数据转换设置和触发

DAC 的数据格式根据对齐设定，有两种存储方式：

- 当 DACx_CR 寄存器中的对齐控制 ALIGN 设定为 0（右对齐），则转换的数据存储于 DACx_DHR[9:0]
- 如果 ALIGN 被设置为 1（左对齐），则后续载入的数据存储于 DACx_DHR[15:6]

当设定的外部触发发生时，DHRx 的值按照上述的格式载入到实际的转换数据寄存器 DORx 中。

DORx 寄存器不能被直接写入，只有设定 DACx_CR 中的 TEN 为 0 时，在写入 DHRx 的同时，会更新相应 DORx 的内容。而 TEN 不为 0 时，根据 TSEL 所选触发来源，将 DHRx 的内容装载入数据输出寄存器 DORx。这里的软件触发源是通过配置寄存器 DACx_CSR.SWTRIG 位来设置，硬件完成触发后自动清零。

DORx 被更新后，DAC 还需要经过 T_{settling} 时间后输出模拟信号，此时间取决于当前电源电压以及 DAC_OUT 输出负载情况。DAC 的输出电压可以通过如下方式换算：

$$\text{DAC_OUT} = \text{VREF} * \text{DOR}/1024$$

当 TEN 不为 0 时，可以选择外部事件来触发转换（软件配置、定时器、EXTI 输入等）。TSEL[2:0]选择具体的触发源，详见下表：

表 11-2 外部触发选择

触发源	类型	TSEL[2:0]
SWTRIG	软件设置触发	000
TIM1_TRGO	内部定时器输出，片内信号触发	001
TIM3_TRGO		010
TIM15_TRGO		011
TIM16_OC1		100
TIM17_OC1		101
EXTI_1	外部输入触发	110
EXIT_9	外部输入触发（DAC0）	111
ACMP1_CPO	比较器 1 输出触发（DAC1）	

11.2.1.3 波形发生器

DAC 数字部分实现了两种模拟波形发生器，通过配置 DAC 每触发一次，更新一个转换数值，因此，触发频率会影响输出波形周期及频率。

噪声波形发生器

产生一个伪随机的噪声模拟波形，为此实现了一个线性反馈移位寄存器单元（LFSR）。设置 DACx_CR 寄存器的 WAVE[1:0]为 01，设定产生模拟噪声，默认预设 LFSR 寄存器的初值为 0x2AA。此寄存器在硬件触发后加载转换数据寄存器 DACx_DOR，LFSR 的值需要通过特定的换算方式进行更新。

LFSR 的值受寄存器 DACx_CR.MAMP[3:0]控制，可以被部分或者全部屏蔽，此计算值与 DACx_DHR 中的求和，然后在触发控制下存入到寄存器 DACx_DOR 中。

当 LFSR 全 0 时，一个为 1 的值替代移位进入。当 WAVE 寄存器被重新设置，LFSR 寄存器恢复到初值。

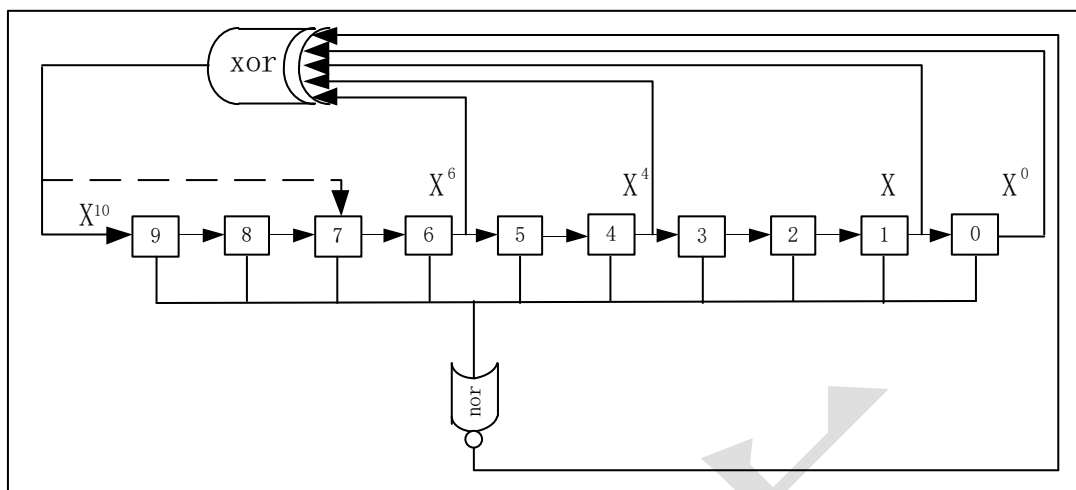


图 11-2 DAC LFSR 计算方式

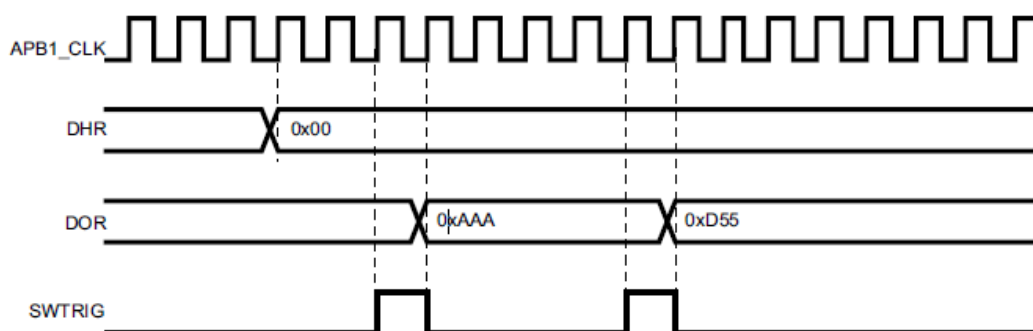


图 11-3 带 LFSR 的 DAC 波形发生过程

注：DACx_CR 寄存器中的 TEN[1:0] 必须为非 0 值才能产生噪声波形

三角波形发生器

DAC 支持产生一个可调整自增减幅度的三角模拟波形，此时需要设置 DACx_CR 寄存器中的 WAVE[1:0] 为 10，幅度通过 DACx_CR 中的 MAMP[3:0] 来控制。内部控制逻辑在每次触发事件时，自动加一基准幅度，与寄存器 DACx_DHR 的值求和后，写入到寄存器 DACx_DOR 中。在计数幅值达到 MAMP 预设的幅值后，内部计数器转而开始每次减 1，直到计数器自减到 0 后再进行自增。

重新设置 WAVE[1:0] 寄存器会复位波形发生器。

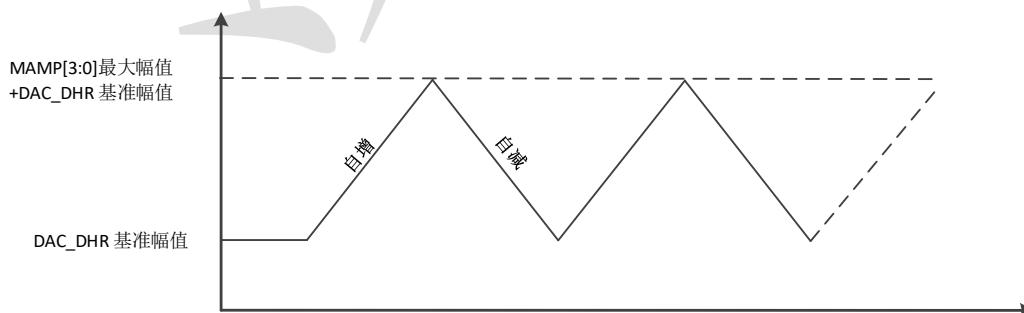


图 11-4 DAC 三角波波形

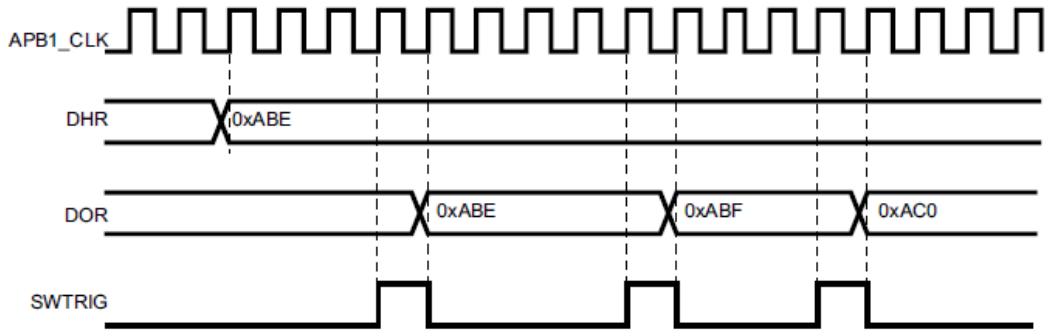


图 11-5 三角波形转换过程（软件触发）

注：DACx_CR 寄存器中的TEN[1:0]必须为非0值才能产生三角波形，MAMP 寄存器必须在DAC使能设置之前配置，否则此寄存器不会更新。

11.2.1.4 DMA 请求

DAC 支持 DMA 通道传输，2 个 DAC 共用一个 DMA 通道，但有可以有各自的 DMA 请求发出，对应不同的 DMA 目标地址。

当收到触发事件时，寄存器 DACx_DHR 的值载入到寄存器 DACx_DOR 中，在寄存器 DACx_CR.DMAEN 使能时，可发出 DMA 请求。

DMA 的欠载

当外部触发事件启动 DAC 数据转换时，如果上一个触发的 DMA 传输还没有完成，则发生一个欠载错误（Underrun），寄存器 DACx_CR.DMAUDR 标志位会被置起。当软件查询到欠载错误发生时，需要重新调整触发事件的发生频率，以匹配 DMA 的传输速率。如果寄存器 DACx_CR.DMAUDRIE 使能，则可以产生相应的 DAC 中断。

注：DAC 控制器本身在欠载发生时，不会屏蔽相应 DMA 传输过程，也不会禁止 DOR 数据的更新，此过程交由 DMA 模块自行控制。

11.2.2 ACMP 功能描述

LCM32F037 内置 3 个比较器，可以用于模拟电压的比较，其内部结构如图 11-6 所示。比较器的输入可选为外部 I/O 端口、DAC 输出或 HALL_MID 输出。比较器的输出是数字信号，可以输出到外部 I/O 端口或用于芯片内部的其他功能：

- 用于触发 ADC
- 用于定时器的 ETR、TI、OCREF_CLR 和刹车输入信号
- 在系统处于睡眠模式或停机模式时唤醒系统

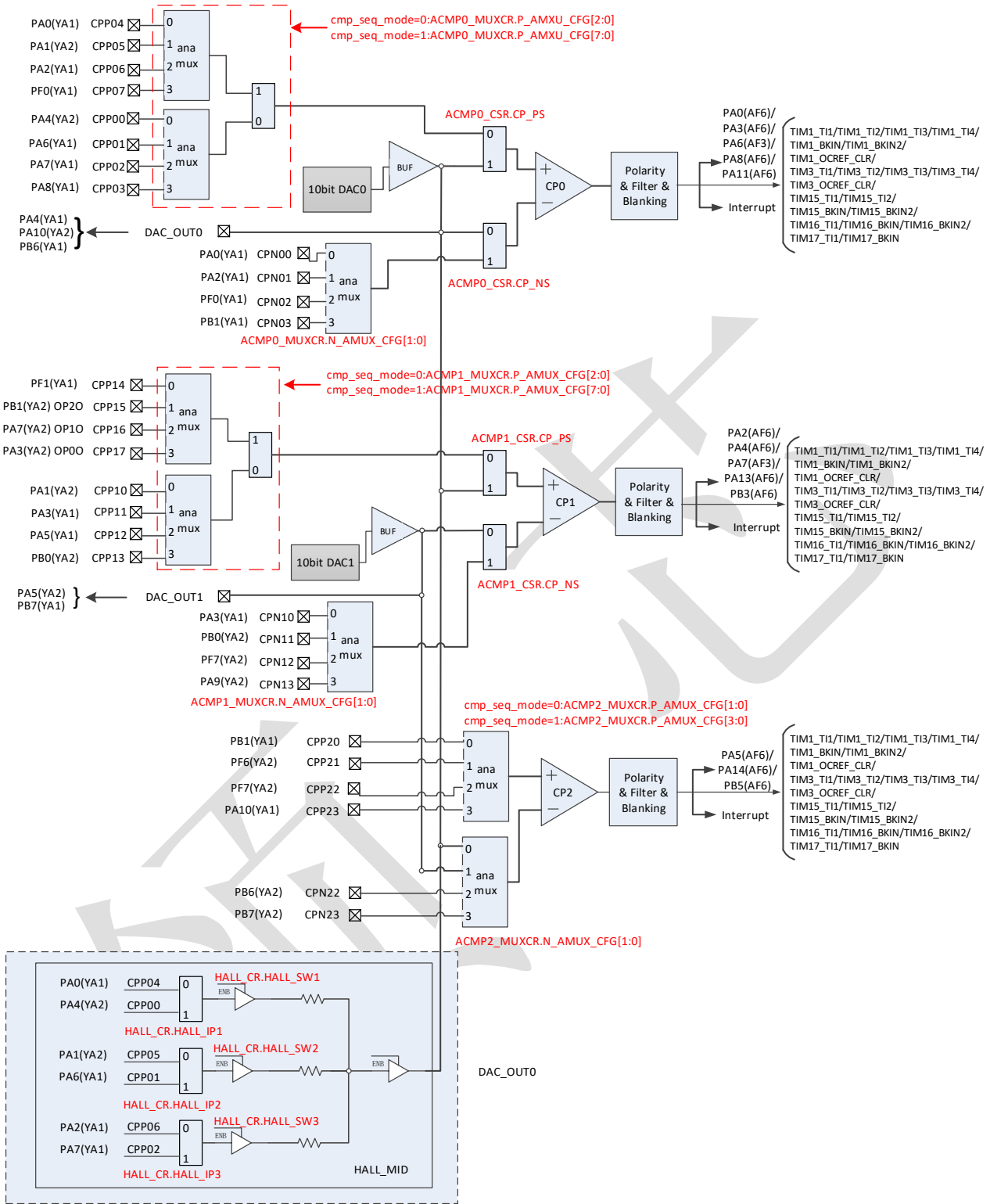


图 11-6 DAC、ACMP 和 HALL_MID 内部框图

注：当模拟通道环路到ADC的输入通道时，I/O 必须配置成模拟模式，两个模拟通道都要使能；外部高阻；I/O 被占用，不得用作其他功能。

表 11-3 比较器输入/输出配置

比较器	控制位及端口	输入来源选择/输出配置	ANA_MUX 输入配置/输出端口配置
比较器 0	控制位	ACMP0_CSR.CP_PS	(CMP_SEQ_MODE=0) : ACMP0_MUXCR.P_AMUX_CFG[2:0] (CMP_SEQ_MODE=1) : ACMP0_MUXCR.P_AMUX_CFG[7:0]
	CPP	0: ANA_MUX 输入 1: DAC0 或 HALL_MID 输出	CPP0 = PA8/PA7/PA6/PA4/PF0/PA2/PA1/PA0 PF0 (YA1) /PA2 (YA1) /PA1 (YA2) /PA0 (YA1) /PA8 (YA1) /PA7 (YA1) /PA6 (YA1) /PA4 (YA2)
	控制位	ACMP0_CSR.CP_NS	ACMP0_MUXCR.N_AMUX_CFG[1:0]
	CPN	0: DAC0 或 HALL_MID 输出 1: ANA_MUX 输入	PB1 (YA1) /PF0 (YA1) /PA2 (YA1) /PA0 (YA1)
	控制位	ACMP0_CSR.IS/DLY/HYSEN/ FS/FREN/BLANKING	-
	CPO	输出相位/延迟/迟滞/消抖/快速 响应/消隐	PA0 (AF6) /PA3 (AF6) /PA6 (AF6) /PA8 (AF6) /PA11 (AF6)
比较器 1	控制位	ACMP1_CSR.CP_PS	(CMP_SEQ_MODE=0) : ACMP1_MUXCR.P_AMUX_CFG[2:0] (CMP_SEQ_MODE=1) : ACMP1_MUXCR.P_AMUX_CFG[7:0]
	CPP	0: ANA_MUX 输入 1: DAC0 或 HALL_MID 输出	PA3 (YA2) /PA7 (YA2) /PB1 (YA2) /PF1 (YA1) /PB0 (YA2) /PA5 (YA1) /PA3 (YA1) /PA1 (YA2)
	控制位	ACMP1_CSR.NS	ACMP1_MUXCR.N_AMUX_CFG[1:0]
	CPN	0: DAC1 输出 1: ANA_MUX 输入	PA9 (YA1) /PF7 (YA2) /PB0 (YA2) /PA3 (YA1)
	控制位	ACMP1_CSR.IS/DLY/HYSEN/ FS/FREN/BLANKING	-
	CPO	输出相位/延迟/迟滞/消抖/快速 响应/消隐	PA2 (AF6) /PA4 (AF6) /PA7 (AF3) /PA13 (AF6) /PB3 (AF6)
比较器 2	控制位	无	(CMP_SEQ_MODE=0) : ACMP2_MUXCR.P_AMUX_CFG[1:0] (CMP_SEQ_MODE=1) : ACMP2_MUXCR.P_AMUX_CFG[3:0]
	CPP	固定 ANA_MUX 输入	PA10 (YA1) /PF7 (YA2) /PF6 (YA2) /PB1 (YA1)
	控制位	无	ACMP2_MUXCR.N_AMUX_CFG[1:0]
	CPN	固定 ANA_MUX 输入	PB7 (YA2) /PB6 (YA2) /DAC1/DAC0
	控制位	ACMP2_CSR.IS/DLY/HYSEN/ FS/FREN/BLANKING	-

	CPO	输出相位/延迟/迟滞/消抖/快速响应/消隐	PA5 (AF6) /PA14 (AF6) /PB5 (AF6)
--	-----	-----------------------	----------------------------------

比较器的时钟和复位

ACMP 控制器使用的时钟与 DAC 控制器相同，其时钟使能通过模块时钟配置寄存器 2 SysCtrl_ClkEnR2.ANA_CLKEN 共同控制，其模块软件复位通过软件复位控制寄存器 1 SysCtrl_RST1.ANA_RST 共同控制。

注：比较器的输出相位选择和复用与 APB 总线时钟无关，即只要设定就可以工作，无论时钟是否关闭，因此即使在系统停机模式下，比较器仍然可以进行转换工作，可以把系统从低功耗模式中唤醒。

比较器迟滞和消隐

比较器迟滞可以通过寄存器 ACMPx_CR.HYSEN 位来决定是否打开，当迟滞使能时，可以滤除一定的输入电平噪声，保证比较器输出的准确性，如图 11-7 所示。

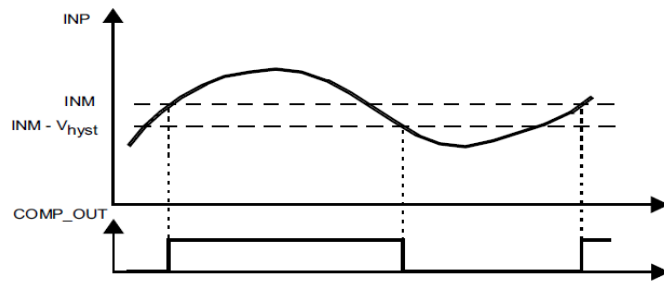


图 11-7 比较器迟滞

另外，比较器支持消隐设定，即在某些内部信号有效的时候，禁止比较器的输出。用户可以通过配置 ACMPx_CSR 寄存器的 BLANKING[2:0]来选择消隐控制输入：

表 11-4 消隐配置

BLANKING[2:0]	消隐输入	BLANKING[2:0]	消隐输入
000	无	001	TIM1_OC4
010	TIM3_OC4	011	TIM15_OC2
100	TIM1_OC3	101	TIM3_OC3
110	TIM15_OC1	111	保留

比较器中断和联动

比较器输出连接到系统 EXTI 线上，可以产生相应的中断和事件，从而用于系统低功耗模式的唤醒。EXTI 线 19 连接到模拟比较器 ACMP0 的输出，EXTI 线 20 连接到模拟比较器 ACMP1 的输出，EXTI 线 21 连接到模拟比较器 ACMP2 的输出。

比较器支持外部触发，触发源可选 EXTI1/3/11、TIM1_OC1/2/3/4 或 TIM3_TRGO，具体配置请参考寄存器 ACMPx_EXTCFG。

比较器的校准

比较器支持校准功能，通过设置 ACMPx_CSR 寄存器的 PF[4:0]位，可以设置比较器的偏移量，默认值 10000，表示默认设计无偏移状态，大于此数值的设定表示比较器负向端增大正向端减小，直至达到 11111，小于此数值则表示正端偏移加大，负端偏移减小，直到 00000，如图 11-8 所示：



图 11-8 ACMP 校准值

PF[4:0]会在系统复位后，由硬件自动从系统存储区（System Memory）中的校准字节 CalByte 10~15 中加载，用户也可以在初始化过程中检查是否加载到位，或者根据自身的应用要求，重新校准比较器偏差。

连续采样模式

图 11-9 描述了 ACMP0 负端选择 CPN00(N_AMUX_CFG [1:0]=0x00)，正端 8 个通道全部使能(P_AMUX_CFG [7:0]=0xFF)，软件触发 (TRIG_MODE = 0) 进行连续采样 (CMP_SEQ_MODE = 1)，并把最终结果存放于寄存器 (ACMP0_MUXCR[15:8])。

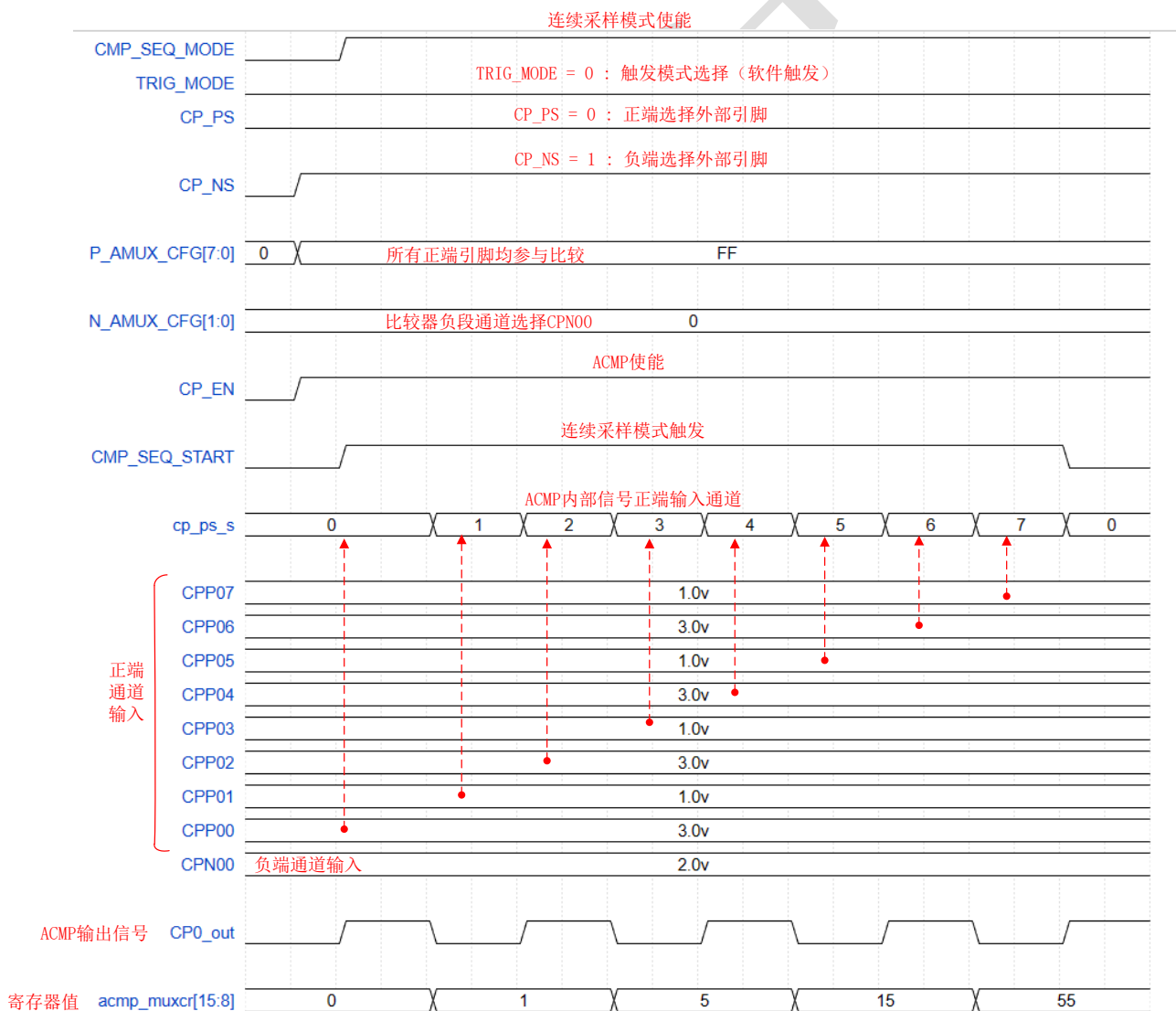


图 11-9 连续采样模式时序图

11.2.3 反电动势采样模块描述

LCM32F037 内置 1 个三端输入的采样电路 HALL_MID，可以用于电机控制时模拟电压的采样。HALL_MID 与 DAC0 共用一个输出信号 DAC_OUT0，编程时须保证两者不同时打开。

HALL_IP1/HALL_IP2/HALL_IP3 连接三个 HALL 信号，HALL_MID 信号是其中两个 HALL 信号的平均值，通过将 HALL 信号与 HALL_MID 信号进行比较，可快速得到 HALL 信号的状态。

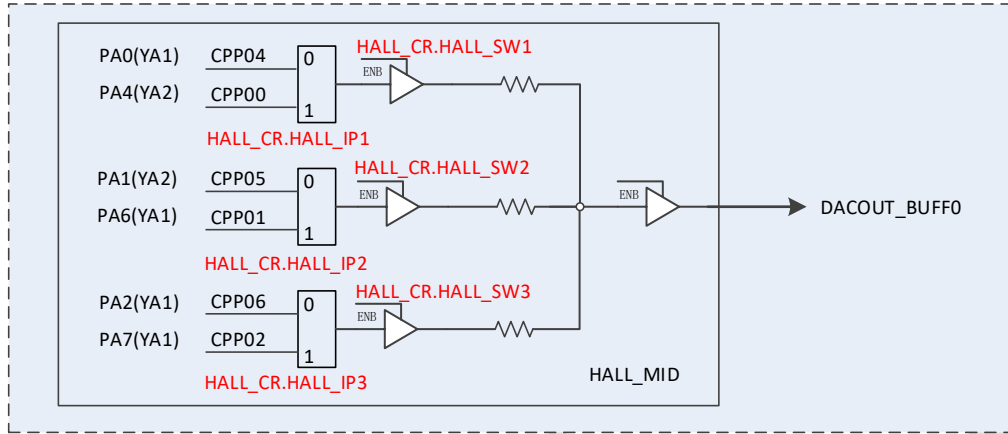


图 11-10 HALL_MID 框图

11.3 DAC/ACMP/HALL_MID 寄存器描述

ACMP、DAC 和 HALL_MID 归属于总线 APB1 管理，共用同一个地址偏移空间，此空间的基地址为 0x4001_7C00，结束地址为 0x4001_7FFF，共 1KB 空间。

表 11-5 DAC/ACMP/HALL_MID 相关寄存器表

名称	说明	读写权限	复位值	字节地址
ACMP0_CSR	ACMP0 控制及状态寄存器	R/W	0x00XX_0000	0x4001_7C20
ACMP0_MUXCR	ACMP0 多路选择控制器	R/W	0x0000_0000	0x4001_7CC0
ACMP0_EXTCFG	ACMP0 外部触发寄存器	R/W	0x0000_0000	0x4001_7CC4
ACMP1_CSR	ACMP1 控制及状态寄存器	R/W	0x00XX_0000	0x4001_7C24
ACMP1_MUXCR	ACMP1 多路选择控制器	R/W	0x0000_0000	0x4001_7CC8
ACMP1_EXTCFG	ACMP1 外部触发寄存器	R/W	0x0000_0000	0x4001_7CCC
ACMP2_CSR	ACMP2 控制及状态寄存器	R/W	0x00XX_0000	0x4001_7C28
ACMP2_MUXCR	ACMP2 多路选择控制器	R/W	0x0000_0000	0x4001_7CD0
ACMP2_EXTCFG	ACMP2 外部触发寄存器	R/W	0x0000_0000	0x4001_7CD4
DAC0_CR	DAC0 控制寄存器	R/W	0x0000_0000	0x4001_7C40
DAC0_CSR	DAC0 控制及状态寄存器	R/W	0x0000_0000	0x4001_7C44
DAC0_DHR	DAC0 数据保持寄存器	R/W	0x0000_0000	0x4001_7C48
DAC0_DOR	DAC0 数据转换输出寄存器	R	0x0000_0000	0x4001_7C4C
DAC1_CR	DAC1 控制寄存器	R/W	0x0000_0000	0x4001_7C50
DAC1_CSR	DAC1 控制及状态寄存器	R/W	0x0000_0000	0x4001_7C54
DAC1_DHR	DAC1 数据保持寄存器	R/W	0x0000_0000	0x4001_7C58
DAC1_DOR	DAC1 数据转换输出寄存器	R	0x0000_0000	0x4001_7C5C
HALL_CR	HALL_MID 控制寄存器	R/W	0x0000_0000	0x4001_7CF4
ANACTRL_CSR	模拟控制及状态寄存器	R/W	0x0000_0000	0x4001_7CF8

注：x 表示不确定；-表示未实现；u 表示不受复位影响；c 表示该位复位值因复位源而异；R 只读；W 只写；R-0 表示只读且读0；R/W 表示可读写（以后章节同上述）。

11.3.1 ACMP0 控制及状态寄存器（ACMP0_CSR）

地址偏移：0x20

复位值：0x00XX_0000

此寄存器的部分复位值取决于相应选项字节的内容。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Res.			CP_OUT_SYNC	Res.							CP_PF[4:0]					
			r													rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	CP_BLANKING[2:0]			Res.	CP_DLY[1:0]	CP_FS	Res.	CP_NS	Res.	CP_PS	CP_IS	CP_HYSEN	CP_FREN	CP_EN	
	rw				rw	rw		rw		rw	rw	rw	rw	rw	rw

Bits	31:29	保留，必须保持复位值																
Bit	28	CP_OUT_SYNC : ACMP0 比较结果同步输出（不受 CP_IS 影响）																
Bits	27:21	保留，必须保持复位值																
Bits	20:16	CP_PF[4:0] : ACMP0 偏移校准值																
		00000: 最大																
		...																
		10000: 0mV																
		...																
		11111: 最小																
Bit	15	保留，必须保持复位值																
Bits	14:12	CP_BLANKING[2:0] : ACMP0 消隐源选择																
		这些位用来控制比较器的输出使用哪个计数器 TIMER 来进行消隐控制																
		000: 无消隐（默认）																
		001: TIM1_OC4 作为消隐控制																
		010: TIM3_OC4 作为消隐控制																
		011: TIM15_OC2 作为消隐控制																
		100: TIM1_OC3 作为消隐控制																
		101: TIM3_OC3 作为消隐控制																
		110: TIM15_OC1 作为消隐控制																
		111: 保留																
Bit	11	保留，必须保持复位值																
Bits	10:9	CP_DLY[1:0] : ACMP0 输出延迟配置																
		<table border="1"> <thead> <tr> <th>CPDLY</th><th>AVDD=2.5</th><th>AVDD=3.6</th><th>AVDD=5</th></tr> </thead> <tbody> <tr> <td>00（默认）</td><td>30ns</td><td>19ns</td><td>13.5ns</td></tr> <tr> <td>01</td><td>140ns</td><td>94ns</td><td>68ns</td></tr> <tr> <td>10</td><td>1.4us</td><td>1000ns</td><td>760ns</td></tr> </tbody> </table>	CPDLY	AVDD=2.5	AVDD=3.6	AVDD=5	00（默认）	30ns	19ns	13.5ns	01	140ns	94ns	68ns	10	1.4us	1000ns	760ns
CPDLY	AVDD=2.5	AVDD=3.6	AVDD=5															
00（默认）	30ns	19ns	13.5ns															
01	140ns	94ns	68ns															
10	1.4us	1000ns	760ns															

	11	2.8us	1.98us	1.5us	
Bit 8	CP_FS: ACMPO 输出滤波使能配置				
	0: 不使能 (默认)				
	1: 使能				
Bit 7	保留, 必须保持复位值				
Bit 6	CP_NS: ACMPO 负端输入选择				
	0: DAC0 或 HALL_MID 的输出 (默认)				
	1: CPNO 外部端口输入				
Bit 5	保留, 必须保持复位值				
Bit 4	CP_PS: ACMPO 正端输入选择				
	0: CPP0 外部端口输入 (默认)				
	1: DAC0 或 HALL_MID 的输出				
Bit 3	CP_IS: ACMPO 极性输出选择				
	0: ACMPO 正相输出 (默认)				
	1: ACMPO 反相输出				
Bit 2	CP_HYSEN: ACMPO 迟滞使能配置				
	0: 无迟滞 (默认)				
	1: 使能, VDD=3.6V 时, 迟滞≈23.4mV				
Bit 1	CP_FREN: ACMPO 快速响应使能				
	0: 非快速响应 (默认), ACMPO 的响应时间大概为 200ns				
	1: 快速响应, ACMPO 的响应时间大概为 60ns				
Bit 0	CP_EN: ACMPO 模块使能				
	0: 禁用 (默认)				
	1: 使能				

11.3.2 ACMPO 多路选择控制器 (ACMPO_MUXCR)

地址偏移: 0xC0

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	CMP_INT_FLG	CMP_INTEN	TRIG_MODE	Res.		N_AMUX_CFG[1:0]		P_AMUX_CFG[7:0]							
	r	rw	rw			rw		rw							

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

Bit 31	保留, 必须保持复位值
Bit 30	CMP_INT_FLG: 连续比较中断标志位 (CMP_INTEN 置 0 后清 0)
	0: 未发生中断 (默认)
	1: 发生中断

Bit	29	CMP_INTEN: 连续比较中断使能屏蔽位
		0: 连续比较结束后不触发中断 (默认)
		1: 连续比较结束后触发中断
Bit	28	TRIG_MODE: 触发模式选择
		0: 软件触发模式 (默认)
		1: 外部硬件触发模式
Bits	27:26	保留, 必须保持复位值
Bits	25:24	N_AMUX_CFG[1:0]: 比较器负端通道选择
		00: CPN00 (默认)
		01: CPN01
		10: CPN02
		11: CPN03
Bits	23:16	P_AMUX_CFG[7:0]: 比较器正端通道选择
		连续比较 (CMP_SEQ_MODE = 1)
		0: 该通道不比较 (默认)
		1: 该通道比较
		Bit23: CPP07
		Bit22: CPP06
		Bit21: CPP05
		Bit20: CPP04
		Bit19: CPP03
		Bit18: CPP02
		Bit17: CPP01
		Bit16: CPP00
		非连续比较 (CMP_SEQ_MODE = 0)
		P_AMUX_CFG[2:0]: 用来选择模拟通道
		111: CPP07
		110: CPP06
		101: CPP05
		100: CPP04
		011: CPP03
		010: CPP02
		001: CPP01
		000: CPP00 (默认)
Bits	15:8	CPOUT_SEQ[7:0]: 连续采样结束后不同通道的比较值
		Bit15: CPP07 与比较器负端的比较值
		Bit14: CPP06 与比较器负端的比较值
		Bit13: CPP05 与比较器负端的比较值
		Bit12: CPP04 与比较器负端的比较值
		Bit11: CPP03 与比较器负端的比较值
		Bit10: CPP02 与比较器负端的比较值
		Bit9: CPP01 与比较器负端的比较值
		Bit8: CPP00 与比较器负端的比较值

Bits 7:2	CHNL_CNT_CFG[5:0]: 连续采样通道间切换时钟周期数选择
	000000: 通道间切换无时间间隔
	...
	111111: 通道间切换间隔 64 个 PCLK
Bit 1	CMP_SEQ_MODE: 连续采样模式选择
	0: 非连续采样模式 (默认)
	1: 连续采样模式
	<i>注: CP_EN 置 1 即开始工作, 无需触发信号</i>
Bit 0	CMP_SEQ_START: 连续模式触发
	0: 空闲状态, 当该位由 1 变为 0 时, 表示一次比较结束 (默认)
	1: 转换触发
	软件模式下 (trig_mode=0), 写 1 触发连续比较, 比较结束后内部清 0, 若需重复比较, 需重新置 1; 硬件模式 (trig_mode=1), 该位由硬件自动设置。

11.3.3 ACMP0 外部触发寄存器 (ACMP0_EXTCFG)

地址偏移: 0xC4

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.			EXTSE[1:0]			Res.			EXTSP[1:0]		SYN_MODE		Res.		
			rw						rw		rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTSELG1[1:0]		EXTSELG0[1:0]		EXTDTG[7:0]								OP_MODE	EDGE_MODE	EXTEN[1:0]	
rw		rw		rw								rw	rw	rw	

Bits 31:29	保留, 必须保持复位值
Bits 28:27	EXTSE[1:0]: 两组触发源使能控制位
	00: 禁止相应触发源触发 ACMP0 (默认)
	01: 使能 EXTSELG0
	1x: 使能 EXTSELG1
Bits 26:24	保留, 必须保持复位值
Bits 23:22	EXTSP[1:0]: 两组触发源极性选择位
	0: 同相输入 (默认)
	1: 反相输入
Bit 21	SYN_MODE: 同/异步模式
	0: 同步模式 (默认)
	1: 异步模式
Bits 20:16	保留, 必须保持复位值
Bits 15:14	EXTSELG1[1:0]: 第 2 组触发源选择
	00: TIM1_OC3 (默认)
	01: TIM1_OC4

	10: EXT13
	11: EXT11
Bits 13:12	EXTSELG0[1:0]: 第 1 组触发源选择
	00: TIM1_OC1 (默认)
	01: TIM1_OC2
	10: TIM3_TRGO
	11: EXT111
Bits 11:4	EXTDTG[7:0]: 触发时延
	延迟模式选择 EXTDTG[7:5]: T_d (延迟时间) 计算公式
	0XX: $T_d = \text{EXTDTG}[6:0] * \text{PCLK}$ (默认)
	10X: $T_d = (\text{EXTDTG}[5:0] + 64) * 2 * \text{PCLK}$
	110: $T_d = (\text{EXTDTG}[4:0] + 32) * 8 * \text{PCLK}$
	111: $T_d = (\text{EXTDTG}[4:0] + 32) * 16 * \text{PCLK}$
Bit 3	OP_MODE: ACMP0 触发方式操作模式
	0: 所有触发信号逻辑或 (默认)
	1: 所有触发信号逻辑与
Bit 2	EDGE_MODE: ACMP0 触发模式
	0: 边沿模式 (默认)
	1: 电平模式
Bits 1:0	EXTEN[1:0]: ACMP0 触发方式
	00: 禁止外部触发, 仅支持软件触发 (默认)
	01: 外部上升沿触发
	10: 外部下降沿触发
	11: 外部双沿触发

11.3.4 ACMP1 控制及状态寄存器 (ACMP1_CSR)

地址偏移: 0x24

复位值: 0x00XX_0000

此寄存器的部分复位值取决于相应选项字节的内容。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Res.			CP_OUT_SYNC	Res.							CP_PF[4:0]					
			r								rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	CP_BLANKING[2:0]			Res.	CP_DLY[1:0]	CP_FS	Res.	CP_NS	Res.	CP_PS	CP_IS	CP_HYSEN	CP_FREN	CP_EN	
rw					rw	rw		rw		rw	rw	rw	rw	rw	

Bits 31:29	保留, 必须保持复位值
Bit 28	CP_OUT_SYNC: ACMP1 比较结果同步输出 (不受 CP_IS 影响)
Bits 27:21	保留, 必须保持复位值
Bits 20:16	CP_PF[4:0]: ACMP1 偏移校准值

	00000: 最大																				
	...																				
	10000: 0mV																				
	...																				
	11111: 最小																				
Bit 15	保留, 必须保持复位值																				
Bits 14:12	CP_BLANKING[2:0]: ACMP1 消隐源选择																				
	这些位用来控制比较器的输出使用哪个计数器 TIMER 来进行消隐控制																				
	000: 无消隐 (默认)																				
	001: TIM1_OC4 作为消隐控制																				
	010: TIM3_OC4 作为消隐控制																				
	011: TIM15_OC2 作为消隐控制																				
	100: TIM1_OC3 作为消隐控制																				
	101: TIM3_OC3 作为消隐控制																				
	110: TIM15_OC1 作为消隐控制																				
	111: 保留																				
Bit 11	保留, 必须保持复位值																				
Bits 10:9	CP_DLY[1:0]: ACMP1 输出延迟配置																				
	<table border="1"> <thead> <tr> <th>CPDLY</th> <th>AVDD=2.5</th> <th>AVDD=3.6</th> <th>AVDD=5</th> </tr> </thead> <tbody> <tr> <td>00 (默认)</td> <td>30ns</td> <td>19ns</td> <td>13.5ns</td> </tr> <tr> <td>01</td> <td>140ns</td> <td>94ns</td> <td>68ns</td> </tr> <tr> <td>10</td> <td>1.4us</td> <td>1000ns</td> <td>760ns</td> </tr> <tr> <td>11</td> <td>2.8us</td> <td>1.98us</td> <td>1.5us</td> </tr> </tbody> </table>	CPDLY	AVDD=2.5	AVDD=3.6	AVDD=5	00 (默认)	30ns	19ns	13.5ns	01	140ns	94ns	68ns	10	1.4us	1000ns	760ns	11	2.8us	1.98us	1.5us
CPDLY	AVDD=2.5	AVDD=3.6	AVDD=5																		
00 (默认)	30ns	19ns	13.5ns																		
01	140ns	94ns	68ns																		
10	1.4us	1000ns	760ns																		
11	2.8us	1.98us	1.5us																		
Bit 8	CP_FS: ACMP1 输出滤波使能配置																				
	0: 不使能 (默认)																				
	1: 使能																				
Bit 7	保留, 必须保持复位值																				
Bit 6	CP_NS: ACMP1 负端输入选择																				
	0: DAC1 的输出 (默认)																				
	1: CPN1 外部端口输入																				
Bit 5	保留, 必须保持复位值																				
Bit 4	CP_PS: ACMP1 正端输入选择																				
	0: CPP1 外部端口输入 (默认)																				
	1: DAC0 或 HALL_MID 的输出																				
Bit 3	CP_IS: ACMP1 极性输出选择																				
	0: ACMP1 正相输出 (默认)																				
	1: ACMP1 反相输出																				
Bit 2	CP_HYSEN: ACMP1 迟滞使能																				
	0: 无迟滞 (默认)																				
	1: 使能, VDD=3.6V 时, 迟滞 \approx 23.4mV																				
Bit 1	CP_FREN: ACMP1 快速响应使能																				
	0: 非快速响应 (默认), ACMP1 的响应时间大概为 200ns																				
	1: 使能快速响应, ACMP1 的响应时间大概为 60ns																				

Bit 0	CP_EN: ACMP1 模块使能
	0: 禁用 (默认)
	1: 使能

11.3.5 ACMP1 多路选择控制器 (ACMP1_MUXCR)

地址偏移: 0xC8

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	CMP_INT_FLG	CMP_INTEN	TRIG_MODE	Res.	N_AMUX_CFG[1:0]			P_AMUX_CFG[7:0]							
	r	rw	rw			rw		rw							

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

Bit 31	保留, 必须保持复位值
Bit 30	CMP_INT_FLG: 连续比较中断标志位 (CMP_INTEN 置 0 后清 0)
	0: 未发生中断 (默认)
	1: 发生中断
Bit 29	CMP_INTEN: 连续比较中断使能屏蔽位
	0: 连续比较结束后不触发中断 (默认)
	1: 连续比较结束后触发中断
Bit 28	TRIG_MODE: 触发模式选择
	0: 软件触发模式 (默认)
	1: 外部硬件触发模式
Bits 27:26	保留, 必须保持复位值
Bits 25:24	N_AMUX_CFG[1:0]: 比较器负端通道选择
	00: CPN10 (默认)
	01: CPN11
	10: CPN12
	11: CPN13
Bits 23:16	P_AMUX_CFG[7:0]: 比较器正端通道选择
	连续比较 (CMP_SEQ_MODE = 1)
	0: 该通道不比较 (默认)
	1: 该通道比较
	Bit23: CPP17
	Bit22: CPP16
	Bit21: CPP15
	Bit20: CPP14
	Bit19: CPP13

	Bit18: CPP12
	Bit17: CPP11
	Bit16: CPP10
	非连续比较 (CMP_SEQ_MODE = 0)
	P_AMUX_CFG[2:0]: 用来选择模拟通道
	111: CPP17
	110: CPP16
	101: CPP15
	100: CPP14
	011: CPP13
	010: CPP12
	001: CPP11
	000: CPP10 (默认)
Bits 15:8	CPOUT_SEQ[7:0]: 连续采样结束后不同通道的比较值
	Bit15: CPP17 与比较器负端的比较值
	Bit14: CPP16 与比较器负端的比较值
	Bit13: CPP15 与比较器负端的比较值
	Bit12: CPP14 与比较器负端的比较值
	Bit11: CPP13 与比较器负端的比较值
	Bit10: CPP12 与比较器负端的比较值
	Bit9: CPP11 与比较器负端的比较值
	Bit8: CPP10 与比较器负端的比较值
Bits 7:2	CHNL_CNT_CFG[5:0]: 连续采样通道间切换时钟周期数选择
	000000: 通道间切换无时间间隔
	...
	111111: 通道间切换间隔 64 个 PCLK
Bit 1	CMP_SEQ_MODE: 连续采样模式选择
	0: 非连续采样模式 (默认)
	1: 连续采样模式
	<i>注: CP_EN 置 1 即开始工作, 无需触发信号</i>
Bit 0	CMP_SEQ_START: 连续模式触发
	0: 空闲状态, 当该位由 1 变为 0 时, 表示一次比较结束 (默认)
	1: 转换触发
	软件模式下 (TRIG_MODE=0), 写 1 触发连续比较, 比较结束后内部清 0, 若需重复比较, 需重新置 1; 硬件模式 (TRIG_MODE=1), 该位由硬件自动设置。

11.3.6 ACMP1 外部触发寄存器 (ACMP1_EXTCFG)

地址偏移: 0xCC

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.			EXTSE[1:0]		Res.			EXTSP[1:0]		SYN_MODE	Res.				
			rw					rw		rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTSELG1[1:0]		EXTSELG0[1:0]		EXTDTG[7:0]								OP_MODE	EDGE_MODE	EXTEN[1:0]	
rw		rw		rw								rw	rw	rw	

Bits	31:29	保留，必须保持复位值
Bits	28:27	EXTSE[1:0]: 两组触发源使能控制位
		00: 禁止相应触发源触发 ACMP1 (默认)
		x1: 使能 EXTSELG0
		1x: 使能 EXTSELG1
Bits	26:24	保留，必须保持复位值
Bits	23:22	EXTSP[1:0]: 两组触发源极性选择位
		0: 同相输入 (默认)
		1: 反相输入
Bit	21	SYN_MODE: 同/异步模式
		0: 同步模式 (默认)
		1: 异步模式
Bits	20:16	保留，必须保持复位值
Bits	15:14	EXTSELG1[1:0]: 第 2 组触发源选择
		00: TIM1_OC3 (默认)
		01: TIM1_OC4
		10: EXT13
		11: EXT11
Bits	13:12	EXTSELG0[1:0]: 第 1 组触发源选择
		00: TIM1_OC1 (默认)
		01: TIM1_OC2
		10: TIM3_TRGO
		11: EXT111
Bits	11:4	EXTDTG[7:0]: 触发时延
		延迟模式选择 EXTDTG[7:5]: T_d (延迟时间) 计算公式
		0XX: $T_d = \text{EXTDTG}[6:0] * \text{PCLK}$ (默认)
		10X: $T_d = (\text{EXTDTG}[5:0] + 64) * 2 * \text{PCLK}$
		110: $T_d = (\text{EXTDTG}[4:0] + 32) * 8 * \text{PCLK}$
		111: $T_d = (\text{EXTDTG}[4:0] + 32) * 16 * \text{PCLK}$
Bit	3	OP_MODE: ACMP1 触发方式操作模式
		0: 所有触发信号逻辑或 (默认)
		1: 所有触发信号逻辑与
Bit	2	EDGE_MODE: ACMP1 触发模式
		0: 边沿模式 (默认)
		1: 电平模式
Bits	1:0	EXTEN[1:0]: ACMP1 触发方式
		00: 禁止外部触发，仅支持软件触发 (默认)

	01: 外部上升沿触发
	10: 外部下降沿触发
	11: 外部双沿触发

11.3.7 ACMP2 控制及状态寄存器 (ACMP2_CSR)

地址偏移: 0x28

复位值: 0x00XX_0000

此寄存器的部分复位值取决于相应选项字节的内容。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Res.			CP_OUT_SYNC	Res.							CP_PF[4:0]					
			r								rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	CP_BLANKING[2:0]			Res.	CP_DLY[1:0]		CP_FS	Res.				CP_IS	CP_HYSEN	CP_FREN	CP_EN
rw				rw		rw					rw	rw	rw	rw	

Bits 31:29	保留, 必须保持复位值		
Bit 28	CP_OUT_SYNC : ACMP2 比较结果同步输出 (不受 CP_IS 影响)		
Bits 27:21	保留, 必须保持复位值		
Bits 20:16	CP_PF[4:0] : ACMP2 偏移校准值		
	00000: 最大		
	...		
	10000: 0mV		
	...		
	11111: 最小		
Bit 15	保留, 必须保持复位值		
Bits 14:12	CP_BLANKING[2:0] : ACMP2 消隐源选择		
	这些位用来控制比较器的输出使用哪个计数器 TIMER 来进行消隐控制		
	000: 无消隐 (默认)		
	001: TIM1_OC4 作为消隐控制		
	010: TIM3_OC4 作为消隐控制		
	011: TIM15_OC2 作为消隐控制		
	100: TIM1_OC3 作为消隐控制		
	101: TIM3_OC3 作为消隐控制		
	110: TIM15_OC1 作为消隐控制		
	111: 保留		
Bit 11	保留, 必须保持复位值		
Bits 10:9	CP_DLY[1:0] : ACMP2 输出延迟配置		
	CPDLY	AVDD=2.5	AVDD=3.6
	00 (默认)	30ns	19ns
	01	140ns	94ns
			AVDD=5
			13.5ns
			68ns

	10	1.4us	1000ns	760ns
	11	2.8us	1.98us	1.5us
Bit 8	CP_FS: ACMP2 输出滤波使能配置			
	0: 不使能 (默认)			
	1: 使能			
Bits 7:4	保留, 必须保持复位值			
Bit 3	CP_IS: ACMP2 极性输出选择			
	0: ACMP2 正相输出 (默认)			
	1: ACMP2 反相输出			
Bit 2	CP_HYSEN: ACMP2 迟滞使能配置			
	0: 无迟滞 (默认)			
	1: 使能, VDD=3.6V 时, 迟滞≈23.4mV			
Bit 1	CP_FREN: ACMP2 快速响应使能			
	0: 非快速响应 (默认), ACMP2 的响应时间大概为 200ns			
	1: 使能快速响应, ACMP2 的响应时间大概为 60ns			
Bit 0	CP_EN: ACMP2 模块使能			
	0: 禁用 (默认)			
	1: 使能			

11.3.8 ACMP2 多路选择控制器 (ACMP2_MUXCR)

地址偏移: 0xD0

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	CMP_INT_FLG	CMP_INTEN	TRIG_MODE	Res.		N_AMUX_CFG[1:0]		Res.							P_AMUX_CFG[3:0]
	r	rw	rw			rw									rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			Res.		CPOUT_SEQ[3:0]									CMP_SEQ_MODE	CMP_SEQ_START
					rw									rw	rw

Bit 31	保留, 必须保持复位值
Bit 30	CMP_INT_FLG: 连续比较中断标志位 (CMP_INTEN 置 0 后清 0)
	0: 未发生中断 (默认)
	1: 发生中断
Bit 29	CMP_INTEN: 连续比较中断使能屏蔽位
	0: 连续比较结束后不触发中断 (默认)
	1: 连续比较结束后触发中断
Bit 28	TRIG_MODE: 触发模式选择
	0: 软件触发模式 (默认)
	1: 外部硬件触发模式

Bits 27:26	保留，必须保持复位值
Bits 25:24	N_AMUX_CFG[1:0]: 比较器负端通道选择
	00: DAC0 或 HALL_MID 的输出（默认）
	01: DAC1
	10: CPN22
	11: CPN23
Bits 23:20	保留，必须保持复位值
Bits 19:16	P_AMUX_CFG[3:0]: 比较器正端通道选择
	连续比较（CMP_SEQ_MODE = 1）
	0: 该通道不比较（默认）
	1: 该通道比较
	Bit19: CPP23
	Bit18: CPP22
	Bit17: CPP21
	Bit16: CPP20
	非连续比较（CMP_SEQ_MODE = 0）
	P_AMUX_CFG[1:0]: 用来选择模拟通道
	00: CPP20（默认）
	01: CPP21
	10: CPP22
	11: CPP23
Bits 15:12	保留，必须保持复位值
Bits 11:8	CPOUT_SEQ[3:0]: 连续采样结束后不同通道的比较值
	Bit11: CPP23 与比较器负端的比较值
	Bit10: CPP22 与比较器负端的比较值
	Bit9: CPP21 与比较器负端的比较值
	Bit8: CPP20 与比较器负端的比较值
Bits 7:2	CHNL_CNT_CFG[5:0]: 连续采样通道间切换时钟周期数选择
	000000: 通道间切换无时间间隔
	...
	111111: 通道间切换间隔 64 个 PCLK
Bit 1	CMP_SEQ_MODE: 连续采样模式选择
	0: 非连续采样模式（默认）
	1: 连续采样模式
	<i>注: CP_EN 置 1 即开始工作，无需触发信号</i>
Bit 0	CMP_SEQ_START: 连续模式触发
	0: 空闲状态，当该位由 1 变为 0 时，表示一次比较结束（默认）
	1: 转换触发
	软件模式下（TRIG_MODE=0），写 1 触发连续比较，比较结束后内部清 0，若需重复比较，需重新置 1；硬件模式（TRIG_MODE=1），该位由硬件自动设置

11.3.9 ACMP2 外部触发寄存器 (ACMP2_EXTCFG)

地址偏移: 0xD4

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.			EXTSE[1:0]		Res.			EXTSP[1:0]		SYN_MODE	Res.				
			rw					rw		rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTSELG1[1:0]		EXTSELG1[1:0]		EXTDTG[7:0]								OP_MODE	EDGE_MODE	EXTEN[1:0]	
rw		rw		rw								rw	rw	rw	

Bits	31:29	保留, 必须保持复位值
Bits	28:27	EXTSE[1:0]: 两组触发源使能控制位
		00: 禁止相应触发源触发 ACMP2 (默认)
		x1: 使能 EXTSELG0
		1x: 使能 EXTSELG1
Bits	26:24	保留, 必须保持复位值
Bits	23:22	EXTSP[1:0]: 两组触发源极性选择位
		0: 同相输入 (默认)
		1: 反相输入
Bit	21	SYN_MODE: 同/异步模式
		0: 同步模式 (默认)
		1: 异步模式
Bits	20:16	保留, 必须保持复位值
Bits	15:14	EXTSELG1[1:0]: 第 2 组触发源选择
		00: TIM1_OC3 (默认)
		01: TIM1_OC4
		10: EXT13
		11: EXT11
Bits	13:12	EXTSELG0[1:0]: 第 1 组触发源选择
		00: TIM1_OC1 (默认)
		01: TIM1_OC2
		10: TIM3_TRGO
		11: EXT111
Bits	11:4	EXTDTG[7:0]: 触发时延
		延迟模式选择 EXTDTG[7:5]: T_d (延迟时间) 计算公式
		0XX: $T_d = \text{EXTDTG}[6:0] * \text{PCLK}$ (默认)
		10X: $T_d = (\text{EXTDTG}[5:0] + 64) * 2 * \text{PCLK}$
		110: $T_d = (\text{EXTDTG}[4:0] + 32) * 8 * \text{PCLK}$
		111: $T_d = (\text{EXTDTG}[4:0] + 32) * 16 * \text{PCLK}$
Bit	3	OP_MODE: ACMP2 触发方式操作模式

	0: 所有触发信号逻辑或（默认）
	1: 所有触发信号逻辑与
Bit 2	EDGE_MODE: ACMP2 触发模式
	0: 边沿模式（默认）
	1: 电平模式
Bits 1:0	EXTEN[1:0]: ACMP2 触发方式
	00: 禁止外部触发，仅支持软件触发（默认）
	01: 外部上升沿触发
	10: 外部下降沿触发
	11: 外部双沿触发

11.3.10 DAC0 控制寄存器（DAC0_CR）

地址偏移：0x40

复位值：0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALIGN	VREF_SEL	DMAUDRIE	DMAEN	MAMP[3:0]				WAVE[1:0]		TSEL[2:0]			TEN[1:0]		EN
rw	rw	rw	rw	rw				rw		rw			rw		rw

Bits 31:16	保留，必须保持复位值
Bit 15	ALIGN: DAC0 的 16 位数据对齐模式
	0: 16 位右对齐（默认）
	1: 16 位左对齐
Bit 14	VREF_SEL: DAC0 参考电压选择
	0: DAC0 内部电压 VDDA（默认）
	1: 芯片电源输出的 DAC0_VREF，具体可参考 ChipCtrl_PWR 寄存器中 DAC0_VREF 的相关配置
Bit 13	DMAUDRIE: DAC0 DMA 欠载中断使能位
	0: DAC0 DMA 欠载中断禁止（默认）
	1: DAC0 DMA 欠载中断使能
Bit 12	DMAEN: DAC0 DMA 使能位
	由软件控制清零和置位
	0: DAC0 DMA 模式禁止（默认）
	1: DAC0 DMA 模式使能
Bits 11:8	MAMP[3:0]: DAC0 屏蔽/放大系数选择因子
	寄存器由软件设置，选择波形自动发生模式中的噪声波形的屏蔽位选择，以及三角波模式中的幅值设定
	0000: 放开序列移位寄存器 LFSR 的第 0 位/三角波形自增幅值为 1（默认）
	0001: 放开序列移位寄存器 LFSR 的[1:0]/三角波形自增幅值为 3

	0010: 放开序列移位寄存器 LFSR 的[2:0]/三角波形自增幅值为 7
	0011: 放开序列移位寄存器 LFSR 的[3:0]/三角波形自增幅值为 15
	0100: 放开序列移位寄存器 LFSR 的[4:0]/三角波形自增幅值为 31
	0101: 放开序列移位寄存器 LFSR 的[5:0]/三角波形自增幅值为 63
	0110: 放开序列移位寄存器 LFSR 的[6:0]/三角波形自增幅值为 127
	≥ 0111: 放开序列移位寄存器 LFSR 的[7:0]/三角波形自增幅值为 255
Bits 7:6	WAVE[1:0]: DAC0 噪声或者三角波形发生器使能控制
	由软件控制清零和置位
	00: 禁止发生波形 (默认)
	01: 使能噪声波形产生
	1x: 三角波形使能
	<i>注: 只能在寄存器 TEN[1:0] 不等于 0 (DAC0 触发使能) 时才会被使用</i>
Bits 5:3	TSEL[2:0]: DAC0 触发选择
	000: 软件触发 (默认)
	001: TIM1_TRGO 事件触发
	010: TIM3_TRGO 事件触发
	011: TIM15_TRGO 事件触发
	100: TIM16_OC1 事件触发
	101: TIM17_OC1 事件触发
	110: 外部中断 1 触发
	111: 外部中断 9 触发
	<i>注: 只能在寄存器 TEN[1:0] 不等于 0 (DAC0 触发使能) 时才会被使用</i>
Bits 2:1	TEN[1:0]: DAC0 触发使能控制
	由软件配置是否允许 DAC0 外部触发
	00: DAC0 外部触发禁止, DAC0_DHR 寄存器的值直接传入转换寄存器 DAC0_DOR (默认)
	01: DAC0 外部触发使能, 且在外触发输入上升沿有效, 触发事件发生时, DAC0_DHR 寄存器的值传入到转换寄存器 DAC0_DOR
	10: DAC0 外部触发使能, 且在外触发输入下降沿有效, 触发事件发生时, DAC0_DHR 寄存器的值传入到转换寄存器 DAC0_DOR
	11: DAC0 外部触发使能, 且在外触发输入上升和下降沿有效, 触发事件发生时, DAC0_DHR 寄存器的值导入到转换寄存器 DAC0_DOR
Bit 0	EN: DAC0 使能位
	0: DAC0 禁止 (默认)
	1: DAC0 使能

11.3.11 DAC0 控制及状态寄存器 (DAC0_CSR)

地址偏移: 0x44

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.							DMAUDR	Res.							SWTRIG
							rw								w

Bits	31:9	保留，必须保持复位值
Bit	8	DMAUDR : DAC0 DMA 欠载标志
		由硬件置位，且通过软件写 1 清零
		0: 没有 DMA 欠载错误发生（默认）
		1: 发生了 DMA 欠载错误（当前选择的触发事件发生频率高于 DMA 传输速率）
Bits	7:1	保留，必须保持复位值
Bit	0	SWTRIG : DAC0 软件触发位
		软件设置此位，触发一次 DAC0 转换（TSEL 选择软件触发）
		0: 无操作（默认）
		1: 软件触发
		<i>注：当触发产生，寄存器 DAC0_DHR 被载入到寄存器 DAC0_DOR 后，此位被硬件自动清除</i>

11.3.12 DAC0 数据保持寄存器（DAC0_DHR）

地址偏移：0x48

复位值：0x0000_0000

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

DAC0_CR 寄存器中的 ALIGN 为 0，16bit 右对齐模式

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

DAC0_CR 寄存器中的 ALIGN 为 1，16bit 左对齐模式

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

Bits	31:16	保留，必须保持复位值
DAC0_CR 寄存器中的 ALIGN 为 0，16-bit 右对齐模式（默认）		
Bits	15:10	保留，必须保持复位值
Bits	9:0	DHR[9:0] : DAC0 16-bit 右对齐的数据
DAC0_CR 寄存器中的 ALIGN 为 1，16-bit 左对齐模式		
Bits	15:6	DHR[9:0] : DAC0 16-bit 左对齐的数据
Bits	5:0	保留，必须保持复位值

11.3.13 DAC0 数据转换输出寄存器 (DAC0_DOR)

地址偏移: 0x4C

复位值: 0x0000_0000

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

DAC0_CR 寄存器中的 ALIGN 为 0, 16-bit 右对齐模式

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

DAC0_CR 寄存器中的 ALIGN 为 1, 16-bit 左对齐模式

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

Bits 31:16	保留, 必须保持复位值
DAC0_CR 寄存器中的 ALIGN 为 0, 16-bit 右对齐模式 (默认)	
Bits 15:10	保留, 必须保持复位值
Bits 9:0	DOR[9:0]: DAC0 16-bit 右对齐的数据
DAC0_CR 寄存器中的 ALIGN 为 1, 16-bit 左对齐模式	
Bits 15:6	DOR[9:0]: DAC0 16-bit 左对齐的数据
Bits 5:0	保留, 必须保持复位值

11.3.14 DAC1 控制寄存器 (DAC1_CR)

地址偏移: 0x50

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALIGN	VREF_SEL	DMAUDRIE	DMAEN	MAMP[3:0]			WAVE[1:0]	TSEL[2:0]		TEN[1:0]		EN			
rw	rw	rw	rw	rw			rw	rw		rw		rw			

Bits 31:16	保留, 必须保持复位值
Bit 15	ALIGN: DAC1 的 16 位数据对齐模式
	0: 16 位右对齐 (默认)
	1: 16 位左对齐
Bit 14	VREF_SEL: DAC1 参考电压选择

	0: DAC1 内部电压 VDDA (默认)
	1: 芯片电源输出的 DAC1_VREF, 具体可参考 ChipCtrl_PWR 寄存器中 DAC1_VREF 的相关配置
Bit 13	DMAUDRIE: DAC1 DMA 欠载中断使能
	0: DAC1 DMA 欠载中断禁止 (默认)
	1: DAC1 DMA 欠载中断使能
Bit 12	DMAEN: DAC1 DMA 使能
	由软件控制清零和置位
	0: DAC1 DMA 模式禁止 (默认)
	1: DAC1 DMA 模式使能
Bits 11:8	MAMP[3:0]: DAC1 屏蔽/放大系数选择因子
	寄存器由软件设置, 选择波形自动发生模式中的噪声波形的屏蔽位选择, 以及三角波模式中的幅值设定
	0000: 放开序列移位寄存器 LFSR 的第 0 位/三角波形自增幅值为 1 (默认)
	0001: 放开序列移位寄存器 LFSR 的[1:0]/三角波形自增幅值为 3
	0010: 放开序列移位寄存器 LFSR 的[2:0]/三角波形自增幅值为 7
	0011: 放开序列移位寄存器 LFSR 的[3:0]/三角波形自增幅值为 15
	0100: 放开序列移位寄存器 LFSR 的[4:0]/三角波形自增幅值为 31
	0101: 放开序列移位寄存器 LFSR 的[5:0]/三角波形自增幅值为 63
	0110: 放开序列移位寄存器 LFSR 的[6:0]/三角波形自增幅值为 127
	≥ 0111: 放开序列移位寄存器 LFSR 的[7:0]/三角波形自增幅值为 255
Bits 7:6	WAVE[1:0]: DAC1 噪声或者三角波形发生器使能控制
	由软件控制清零和置位
	00: 禁止发生波形 (默认)
	01: 使能噪声波形产生
	1x: 三角波形使能
	<i>注: 只能在寄存器 TEN[1:0] 不等于 0 (DAC1 触发使能) 时才会被使用</i>
Bits 5:3	TSEL[2:0]: DAC1 触发选择
	000: 软件触发 (默认)
	001: TIM1_TRGO 事件触发
	010: TIM3_TRGO 事件触发
	011: TIM15_TRGO 事件触发
	100: TIM16_OC1 事件触发
	101: TIM17_OC1 事件触发
	110: 外部中断 1 触发
	111: ACMP1_OUT 触发
	<i>注: 只能在寄存器 TEN[1:0] 不等于 0 (DAC1 触发使能) 时才会被使用</i>
Bits 2:1	TEN[1:0]: DAC1 触发使能控制
	由软件配置是否允许 DAC1 外部触发
	00: DAC1 外部触发禁止, 写入寄存器 DAC1_DHR 的值直接传入转换寄存器 DAC1_DOR (默认)
	01: DAC1 外部触发使能, 且在外触发输入上升沿有效, 触发事件发生时, DAC1_DHR 寄存器的值导入到转换寄存器 DAC1_DOR

	10: DAC1 外部触发使能, 且在外部触发输入下降沿有效, 触发事件发生时, DAC1_DHR 寄存器的值导入到转换寄存器 DAC1_DOR
	11: DAC1 外部触发使能, 且在外部触发输入上升和下降沿有效, 触发事件发生时, DAC1_DHR 寄存器的值导入到转换寄存器 DAC1_DOR
Bit 0	EN: DAC1 使能位
	0: DAC1 禁止 (默认)
	1: DAC1 使能

11.3.15 DAC1 控制及状态寄存器 (DAC1_CSR)

地址偏移: 0x54

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.							DMAUDR	Res.							SWTRIG
							rw								w

Bits 31:9	保留, 必须保持复位值
Bit 8	DMAUDR: DAC1 DMA 欠载标志
	由硬件置位, 且通过软件写 1 清零
	0: 没有 DMA 欠载错误发生 (默认)
	1: 发生了 DMA 欠载错误 (当前选择的触发事件发生频率高于 DMA 传输速率)
Bits 7:1	保留, 必须保持复位值
Bit 0	SWTRIG: DAC1 软件触发位
	软件设置此位, 触发一次 DAC1 转换 (TSEL 选择软件触发)
	0: 无操作 (默认)
	1: 软件触发
	注: 当触发产生, 寄存器 DAC1_DHR 被载入到寄存器 DAC1_DOR 后, 此位被硬件自动清除

11.3.16 DAC1 数据保持寄存器 (DAC1_DHR)

地址偏移: 0x58

复位值: 0x0000_0000

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

DAC1_CR 寄存器中的 ALIGN 为 0, 16-bit 右对齐模式

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

DAC1_CR 寄存器中的 ALIGN 为 1，16-bit 左对齐模式

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

Bits 31:16	保留，必须保持复位值
DAC1_CR 寄存器中的 ALIGN 为 0，16-bit 右对齐模式（默认）	
Bits 15:10	保留，必须保持复位值
Bits 9:0	DHR[9:0] : DAC1 16-bit 右对齐的数据
DAC1_CR 寄存器中的 ALIGN 为 1，16-bit 左对齐模式	
Bits 15:6	DHR[9:0] : DAC1 16-bit 左对齐的数据
Bits 5:0	保留，必须保持复位值

11.3.17 DAC1 数据转换输出寄存器（DAC1_DOR）

地址偏移：0x5C

复位值：0x0000_0000

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

DAC1_CR 寄存器中的 ALIGN 为 0，16-bit 右对齐模式

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						DOR[9:0]									
r															

DAC1_CR 寄存器中的 ALIGN 为 1，16-bit 左对齐模式

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DOR[9:0]										Res.					
r															

Bits 31:16	保留，必须保持复位值
DAC1_CR 寄存器中的 ALIGN 为 0，16-bit 右对齐模式（默认）	
Bits 15:10	保留，必须保持复位值
Bits 9:0	DOR[9:0] : DAC1 16-bit 右对齐的数据
DAC1_CR 寄存器中的 ALIGN 为 1，16-bit 左对齐模式	
Bits 15:6	DOR[9:0] : DAC1 16-bit 左对齐的数据
Bits 5:0	保留，必须保持复位值

11.3.18 HALL_MID 控制寄存器（HALL_CR）

地址偏移：0xF4

复位值：0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.									HALL_ SW3	HALL_ SW2	HALL_ SW1	HALL_ IP3_ SEL	HALL_ IP2_ SEL	HALL_ IP1_ SEL	HALL_ MID_ EN
									rw	rw	rw	rw	rw	rw	rw

Bits	31:7	保留，必须保持复位值
Bit	6	HALL_SW3: HALL_IP3 到 HALL_MID 通路之间的开关
		0: HALL_IP3 到 HALL_MID 关断（默认）
		1: HALL_IP3 到 HALL_MID 导通
Bit	5	HALL_SW2: HALL_IP2 到 HALL_MID 通路之间的开关
		0: HALL_IP2 到 HALL_MID 关断（默认）
		1: HALL_IP2 到 HALL_MID 导通
Bit	4	HALL_SW1: HALL_IP1 到 HALL_MID 通路之间的开关
		0: HALL_IP1 到 HALL_MID 关断（默认）
		1: HALL_IP1 到 HALL_MID 导通
Bit	3	HALL_IP3_SEL: HALL_IP3 通道选择
		0: HALL_IP3 通道选择 CPP06（默认）
		1: HALL_IP3 通道选择 CPP02
Bit	2	HALL_IP2_SEL: HALL_IP2 通道选择
		0: HALL_IP2 通道选择 CPP05（默认）
		1: HALL_IP2 通道选择 CPP01
Bit	1	HALL_IP1_SEL: HALL_IP1 通道选择
		0: HALL_IP1 通道选择 CPP04（默认）
		1: HALL_IP1 通道选择 CPP00
Bit	0	HALL_MID_EN: HALL_MID 信号输出控制开关
		0: HALL_MID 信号不输出（默认）
		1: HALL_MID 信号输出至 DAC_OUT0

11.3.19 模拟控制及状态寄存器（ANACTRL_CSR）

地址偏移：0xF8

复位值：0x0000_0000

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



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						OPA_NIN[1:0]		Res.			DAC0_ BUF_EN	Res.			DAC1_ BUF_EN
						rw					rw				rw

Bits	31:10	保留，必须保持复位值
Bits	9:8	OPA_NIN[1:0]: OP0N/OP1N/ OP2N 输入选择
		00: VSS (默认)
		01: PA8 (YA2) 输入
		10: PF1 (YA1) 输入
		11: PA6 (YA2) 输入
Bits	7:5	保留，必须保持复位值
Bit	4	DACO_BUF_EN: DAC0 输出驱动使能位
		0: 输出驱动禁止 (默认)
		1: 输出驱动使能
Bits	3:1	保留，必须保持复位值
Bit	0	DAC1_BUF_EN: DAC1 输出驱动使能位
		0: 输出驱动禁止 (默认)
		1: 输出驱动使能

12. 运算放大器（OPA）

LCM32F037 内置 3 个运算放大器，放大倍数为 1/2/6/8/10/16/20/32，同时输出偏置为 OPA_VRH/2 或 DAC_OUT1。

12.1 OPA 主要特性

- 输入共模电压：0V ~ VDDA
- 可调增益：1/2/6/8/10/16/20/32
- 输入失调电压：-/+ 5mV（未校准）；-/+ 1mV（校准后）
- 转换速率：5V/us
- 电源电压抑制比：60（min），80（typ）
- 增益系数精度：+/- 1%
- 运放的温度漂移：0.1%
- CMRR：90dB
- 运放的温度特性：温度越高，放大倍数略大，即放大曲线上翘，严禁下曲

12.2 管脚配置

表 12-1 OPA 管脚配置

管脚名称	管脚类型	管脚描述	复用 I/O 口	配置
OP0P	I	OPA0 正端输入	PA2/PA9/PF7	OP0PES
OP0N	I	OPA0 负端输入	PA4/PA8/PF6	OP0NES
OP0O	O	OPA0 输出	PA3	OP0EN 置 1
OP0OEX	O	OPA0EX 输出	PF6	OP0OE 置 1
OP1P	I	OPA1 正端输入	PA6/PA7/PF0	OP1PES
OP1N	I	OPA1 负端输入	PB0/PB1/PF1	OP1NES
OP1O	O	OPA1 输出	PA7	OP1EN 置 1
OP1OEX	O	OPA1EX 输出	PF7	OP1OE 置 1
OP2P	I	OPA2 正端输入	PB0/PA5/PB6	OP2PES
OP2N	I	OPA2 负端输入	PA6/PA7/PB7	OP2NES
OP2O	O	OPA2 输出	PB1	OP2EN 置 1
OP2OEX	O	OPA2EX 输出	PA2	OP2OE 置 1
OPA_NIN	I	OPA NIN 输入	PA6/PA8/PF1	ANACTRL_CSR 寄存器中 OPA_NIN

12.3 功能描述

OPA0/1/2 的功能相同，支持输出偏置，偏置电压为 OPA_VRH/2 或 DAC_OUT1，其中 OPA_VRH 可选择为 PA0 输入电压、VDDA、2.5V、4V（具体参考芯片 ChipCtrl_PWR 寄存器 ADPREF[1:0]配置）。



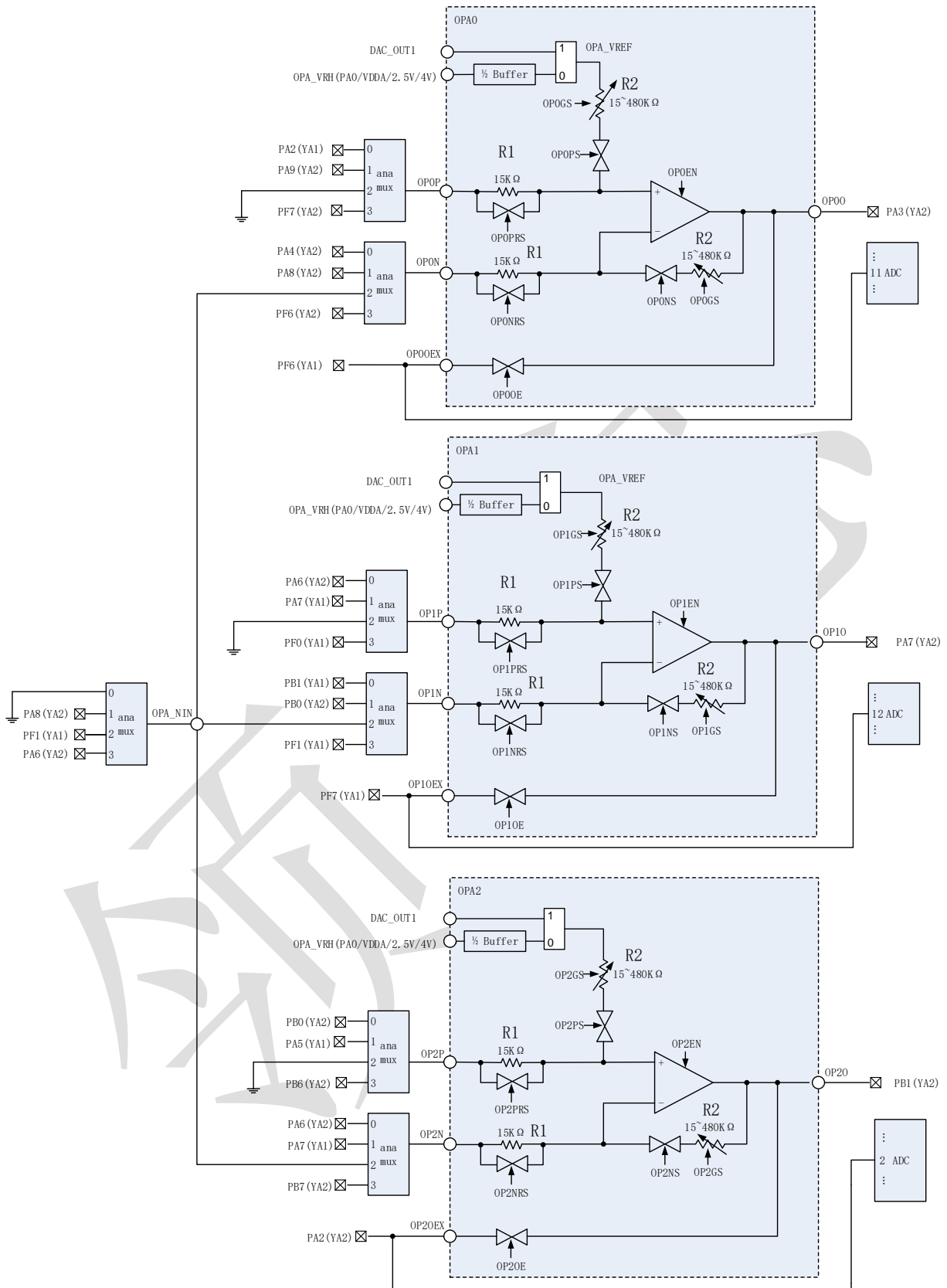


图 12-1 OPA 内部结构和外部互联

12.3.1 OPA 校准

OPA 有一个内置校准模式，校准可大幅度减少因芯片内部差异造成的运放正端输入 OPxP 和负端输入 OPxN 的电压偏差。在出厂前会对运放进行一次校准，并把校准结果存放在 Flash 的 Option Byte 中，在上电复位中会自动加载到 ANACTRL_OPAX_CSR.OPxOS[4:0] (x=1 或 2)，用户可以通过调节 OPxOS[4:0] 的值，进行重新校准，具体步骤如下：

1. 设置寄存器 ANACTRL_OPAX_CSR.OPxNS=0，断开反馈电阻，ANACTRL_OPAX_CSR.OPxPS=1，端口正端偏置，使运放处于开环无限增益状态。
2. 设置寄存器 ANACTRL_OPAX_CSR.OPxPES=10，ANACTRL_OPAX_CSR.OPxNES=10，使得正负端接地，即运放 OPAX 的正负输入端处在相同电压状态（用户也可以根据外围电路的实际选择输入端的设置）；此外，ANACTRL_OPAX_CSR.OPxPRS 的值由用户根据实际中是否使用 R1 串联电阻确定。
3. 设置寄存器 ANACTRL_OPAX_CSR.OPxTRIMEN=1，使能校准过程。
4. 设置寄存器 ANACTRL_OPAX_CSR.OPxEN=1，使能运放。
5. 调准 ANACTRL_OPAX_CSR.OPxOS[4:0] 的值：
 - 将 OPxOS 的值逐步从 11111 减小到 00000，判断 ANACTRL_OPA0_CSR.OPxTRIMO 的值是否从 0 跳变为 1，如果为 1，则保存 OPxOS 的值。
 - 将 OPxOS 的值逐步从 00000 增加到 11111，判断 ANACTRL_OPA0_CSR.OPxTRIMO 的值是否从 1 跳变为 0，如果为 0，则保存 OPxOS 的值；与上一步的保存值做平均，即为实际校准值，校准完成，退出校准。

校准过程如图 12-2 所示：

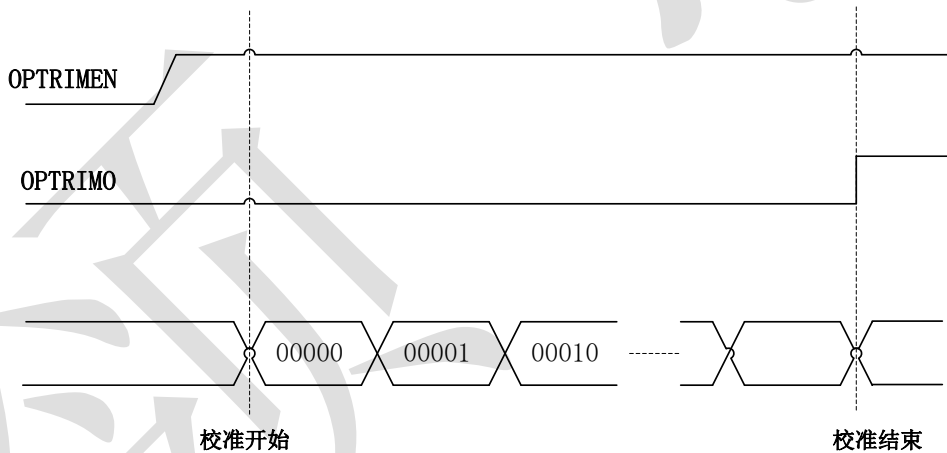


图 12-2 OPA 校准过程

12.3.2 运放 OPA0 联动与级联

运放 OPA0 支持与运放 OPA1 和 OPA2 的级联，运放 OPA1/2 的输出 OP1OEX、OP2OEX 可以通过 I/O 的模拟通道环路到 OPA0 的正向输入端。此外，运放的输出 OP00/OP00EX 可以和通过 I/O 的模拟通道环路到模拟比较器 ACMP 和 ADC（在环路级联情况下 I/O 必须配置成模拟模式，两个模拟通道都要使能；外部高阻；I/O 被占用，不得用作其他功能），具体如表 12-2 所示。

表 12-2 运放 OPA0 与其他模块的联动/级联关系

运放 OPA0 引脚	联动/级联引脚	说明
OP0P3	OP1OEX	运放 OPA1 的输出通过引脚 PF7 环路连接到运放 OPA0 的正端输入

OPOPO	OP2OEX	运放 OPA2 的输出通过引脚 PA2 环路连接到 OPA0 的正端输入
OP00	ADCIN[3]	OP00 通过引脚 PA3 与 ADC 通道 3 环路相连
	CPP17	与 ACMP1 正端直接相连
	CPP11	与 ACMP1 正端环路相连
	CPN11	与 ACMP1 负端环路相连
OPOOEX	ADCIN[11]	OPOOEX 通过引脚 PF6 与 ADC 通道 11 直接相连
	CPP21	与 ACMP2 正端环路相连

12.3.3 运放 OPA1 联动与级联

运放 OPA1 支持与运放 OPA0 和 OPA2 的级联，运放 OPA1 的输出 OP1OEX 可以通过 I/O 的模拟通道环路到 OPA0 的正向输入端，运放 OPA1 的输出 OP1O 可以通过 I/O 的模拟通道环路到 OPA2 的负端，运放 OPA2 的输出 OP2O 可以通过 I/O 的模拟通道环路到运放 OPA1 的负端。此外，运放的输出 OP1O/OP1OEX 可以通过 I/O 的模拟通道环路到比较器 ACMP 和 ADC（在环路级联情况下 I/O 必须配置成模拟模式，两个模拟通道都要使能；外部高阻；I/O 被占用，不得用作其他功能），具体如表 12-3 所示。

表 12-3 运放 OPA1 与其他模块的联动/级联关系

运放 OPA1 引脚	联动/级联引脚	说明
OP1N0	OP2O	运放 OPA2 的输出 OP2O 通过引脚 PB1 环路到 OPA1 的负端
OP1O	ADCIN[7]	OP1O 通过引脚 PA7 与 ADC 通道 7 环路相连
	OP2N1	与运放 OPA2 的负端输入环路相连
	CPP02	与运放 ACMP0 的正端输入环路相连
	CPP16	与运放 ACMP1 的正端输入直接相连
OP1OEX	ADCIN[12]	OP1O 通过引脚 PF7 与 ADC 通道 12 直接相连
	OPOP3	与 OPA0 的正端环路相连
	CPN12	与运放 ACMP1 的负端输入环路相连
	CPP22	与运放 ACMP2 的正端输入环路相连

12.3.4 运放 OPA2 联动与级联

运放 OPA2 支持与运放 OPA0 和 OPA1 的级联，运放 OPA1 的输出 OP1O 可以通过 I/O 的模拟通道环路到 OPA2 的负端，运放 OPA2 的输出端 OP2OEX 可以通过引脚模拟通道环路到运放 OPA0 的正端 OPOP。此外，运放的输出 OP2O/OP2OEX 可以通过 I/O 的模拟通道环路到比较器 ACMP 和 ADC（在环路级联情况下 I/O 必须配置成模拟模式，两个模拟通道都要使能；外部高阻；I/O 被占用，不得用作其他功能），具体如表 12-4 所示。

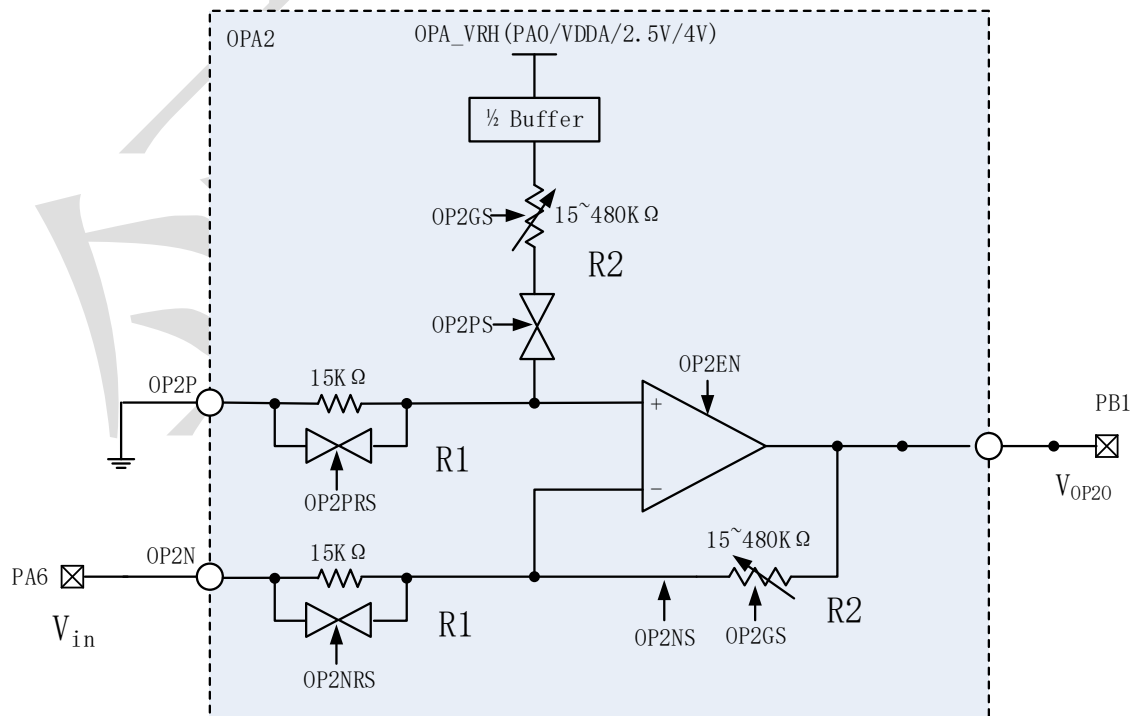
表 12-4 运放 OPA2 与其他模块的联动/级联关系

运放 OPA2 引脚	联动/级联引脚	说明
OP2N1	OP1O	运放 OPA1 的输出 OP1O 通过

		引脚 PA7 环路到 OPA2 的负端
OP2O	ADCIN[9]	运放 OPA2 的输出 OP2O 通过引脚 PB1 与 ADC 的通道 9 环路相连
	OP1N0	与 OPA1 的负端环路相连
	CPN03	与运放 ACMP0 的负端输入环路相连
	CPP20	与运放 ACMP2 的正端输入环路相连
OP2OEX	ADCIN[2]	运放 OPA2 的输出 OP2OEX 通过引脚 PA2 与 ADC 的通道 2 直接相连
	OP0P0	OPA0 的正端 OP0P 环路相连
	CPN03	与运放 ACMP0 的负端输入环路相连
	CPP20	与运放 ACMP2 的正端输入环路相连

12.3.5 OPA 运放典型配置

运放 OPA0/1/2 内部除了集成反馈电阻以外，还具有偏置电压 OPA_VRH/2 或 DAC_OUT1，用户可以根据实际需求，将 OPA 配置成开环模式，由外部提供配置和放大电路；也可以直接将 OPA 配置为内部偏置和反馈，这样可以大幅减少外部器件数目。



$$V_{OP20} = OPA_VRH/2 - (R2/R1)*V_{in}$$

图 12-3 OPA2 典型应用配置

图 12-3 给出了 OPA2 的典型应用方式，内部带偏置和反馈，配置如下：

- ANACTRL_OPA2_CSR.VM_SEL = 0
- ANACTRL_OPA2_CSR.OP2PES = 10
- ANACTRL_OPA2_CSR.OP2NES = 00
- ANACTRL_OPA2_CSR.OP2PS = 1
- ANACTRL_OPA2_CSR.OP2NS = 0

其它配置根据实际的使用情况配置，放大倍数由 ANACTRL_OPA2_CSR.OP2GS 所设置的 R2 决定。

12.4 OPA 寄存器描述

表 12-5 OPA 相关寄存器表

名称	说明	读写权限	复位值	字节地址
ANACTRL_OPA0_CSR	OPA0 控制状态寄存器	R/W	0x0000_0000	0x4001_7C00
ANACTRL_OPA1_CSR	OPA1 控制状态寄存器	R/W	0x0000_0000	0x4001_7C04
ANACTRL_OPA2_CSR	OPA2 控制状态寄存器	R/W	0x0000_0000	0x4001_7C08

注：x 表示不确定；-表示未实现；u 表示不受复位影响；c 表示该位复位值因复位源而异；R 只读；W 只写；R-0 表示只读且读0；R/W 表示可读写

12.4.1 OPA0 控制状态寄存器 (ANACTRL_OPA0_CSR)

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
OPOTRIMO	OPOTRIMEN	Res.		OPOPE[1:0]		OPONES[1:0]		Res.			OPOOS[4:0]				
r	rw			rw		rw					rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				OPOGS[2:0]			Res.	VM_SEL	OPOPS	OPONS	OPOOE	OPOPRS	OPONRS	OP0EN	
				rw				rw	rw	rw	rw	rw	rw	rw	

Bit	31	OPOTRIMO: OPA0 修调状态 (以 OPxOS 从 00000 到 11111 为例)
		0: OPA0 在修调中或者 TRIM 禁止 (默认)
		1: OPA0 修调结束
Bit	30	OPOTRIMEN: OPA0 修调使能
		0: OPA0 修调禁止 (默认)
		1: OPA0 修调使能
Bits	29:28	保留, 必须保持复位值
Bits	27:26	OPOPE[1:0]: OPA0 正端输入 (OPOP) 选择
		00: 引脚 PA2 作为正端输入 (默认)
		01: 引脚 PA9 作为正端输入
		10: VSS 作为正端输入
		11: 引脚 PF7 作为正端输入
Bits	25:24	OPONES[1:0]: OPA0 负端输入选择
		00: 引脚 PA4 作为负端输入 (默认)

	01: 引脚 PA8 作为负端输入
	10: OPA_NIN 作为负端输入 (详见 ANACTRL_CSR Bit[9:8])
	11: 引脚 PF6 作为负端输入
Bits 23:21	保留, 必须保持复位值
Bits 20:16	OPOOS[4:0]: OPA0 电压偏移修调 (TRIM) 值, 10000 为中间值, 增加为正偏, 减小为反偏 (默认值保存在 Flash 的 Option Byte 中)
	00000: OP0N-OP0P 电压偏差最大 (默认)
	...
	11111: OP0P-OP0N 电压偏差最大
	在修调过程中, OPOOS[4:0]的值由 00000->11111 变化, OP0TRIMO 变为 1 即修调结束
Bits 15:11	保留, 必须保持复位值
Bits 10:8	OPOGS[2:0]: 运放反馈电阻值 (R2) 选择, 即运放 OPA0 放大倍数选择 (放大倍数 = R2/R1)
	000: x1 (默认)
	001: x2
	010: x6
	011: x8
	100: x10
	101: x16
	110: x20
	111: x32
Bit 7	保留, 必须保持复位值
Bit 6	VM_SEL: 偏置电压源选择
	0: 运放共模偏置来自 OPA_VRH/2 (默认)
	1: 运放共模偏置来自 DAC_OUT1
Bit 5	OPOPS: OPA0 正端偏置使能
	0: 正端偏置禁止 (默认)
	1: 正端偏置使能
Bit 4	OPONS: OPA0 负端反馈电阻使能
	0: 反馈电阻断开 (默认)
	1: 反馈电阻连接
Bit 3	OPOOE: OPA0 运放输出至 OPOOEX 使能
	0: 连接禁止 (默认)
	1: 连接使能
Bit 2	OPOPRS: OPA0 正端输入电阻短路
	0: 不短路 (默认)
	1: 短路
Bit 1	OPONRS: OPA0 负端输入电阻短路
	0: 不短路 (默认)
	1: 短路
Bit 0	OPOEN: 运放 OPA0 使能
	0: 禁止 (默认)



1: 使能

12.4.2 OPA1 控制状态寄存器 (ANACTRL_OPA1_CSR)

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
OP1TRIMO	OP1TRIMEN	Res.		OP1PES[1:0]		OP1NES[1:0]		Res.			OP1OS[4:0]				
r	rw			rw		rw					rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				OP1GS[2:0]			Res.	VM_SEL	OP1PS	OP1NS	OP1OE	OP1PRS	OP1NRS	OP1EN	
					rw				rw	rw	rw	rw	rw	rw	rw

Bit	31	OP1TRIMO: OPA1 修调状态
		0: OPA1 在修调中或者 TRIM 禁止 (默认)
		1: OPA1 修调结束
Bit	30	OP1TRIMEN: OPA1 修调使能
		0: OPA1 修调禁止 (默认)
		1: OPA1 修调使能
Bits	29:28	保留, 必须保持复位值
Bits	27:26	OP1PES[1:0]: OPA1 正端输入 (OP1P) 选择
		00: 引脚 PA6 作为正端输入 (默认)
		01: 引脚 PA7 作为正端输入
		10: VSS 作为正端输入
		11: 引脚 PF0 作为正端输入
Bits	25:24	OP1NES[1:0]: OPA1 负端输入选择
		00: 引脚 PB1 作为负端输入 (默认)
		01: 引脚 PB0 作为负端输入
		10: OPA_NIN 作为负端输入 (详见 ANACTRL_CSR Bit[9:8])
		11: 引脚 PF1 作为负端输入
Bits	23:21	保留, 必须保持复位值
Bits	20:16	OP1OS[4:0]: OPA1 电压偏移修调 (TRIM) 值, 10000 为中间值, 增加为正偏, 减小为反偏 (默认值保存在 Flash 的 Option Byte 中)
		00000: OP1N-OP1P 电压偏差最大 (默认)
		...
		11111: OP1P-OP1N 电压偏差最大
		在修调过程中, OP1OS[4:0]的值由 00000->11111 变化, OP1TRIMO 变为 1 即修调结束
Bits	15:11	保留, 必须保持复位值
Bits	10:8	OP1GS[2:0]: 运放反馈电阻值 (R2) 选择, 即运放 OPA1 放大倍数选择 (放大倍数 = R2/R1)
		000: x1 (默认)



	001: x2
	010: x6
	011: x8
	100: x10
	101: x16
	110: x20
	111: x32
Bit 7	保留, 必须保持复位值
Bit 6	VM_SEL: 偏置电压源选择
	0: 运放共模偏置来自 OPA_VRH/2 (默认)
	1: 运放共模偏置来自 DAC_OUT1
Bit 5	OP1PS: OPA1 正端偏置使能
	0: 正端偏置禁止 (默认)
	1: 正端偏置使能
Bit 4	OP1NS: OPA1 负端反馈电阻使能
	0: 反馈电阻断开 (默认)
	1: 反馈电阻连接
Bit 3	OP1OE: OPA1 运放输出至 OP1OEX 使能
	0: 连接禁止 (默认)
	1: 连接使能
Bit 2	OP1PRS: OPA1 正端输入电阻短路
	0: 不短路 (默认)
	1: 短路
Bit 1	OP1NRS: OPA1 负端输入电阻短路
	0: 不短路 (默认)
	1: 短路
Bit 0	OP1EN: 运放 OPA1 使能
	0: 禁止 (默认)
	1: 使能

12.4.3 OPA2 控制状态寄存器 (ANACTRL_OPA2_CSR)

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
OP2TRIMO	OP2TRIMEN	Res.		OP2PES[1:0]		OP2NES[1:0]		Res.			OP2OS[4:0]				
r	rw			rw		rw					rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				OP2GS[2:0]			Res.	VM_SEL	OP2PS	OP2NS	OP2OE	OP2PRS	OP2NRS	OP2EN	
					rw				rw	rw	rw	rw	rw	rw	rw

Bit 31	OP2TRIMO: OPA2 修调状态
--------	----------------------------

	0: OPA2 在修调中或者 TRIM 禁止 (默认)
	1: OPA2 修调结束
Bit 30	OP2TRIMEN: OPA2 修调使能
	0: OPA2 修调禁止 (默认)
	1: OPA2 修调使能
Bits 29:28	保留, 必须保持复位值
Bits 27:26	OP2PES[1:0]: OPA2 正端输入 (OP2P) 选择
	00: 引脚 PB0 作为正端输入 (默认)
	01: 引脚 PA5 作为正端输入
	10: VSS 作为正端输入
	11: 引脚 PB6 作为正端输入
Bits 25:24	OP2NES[1:0]: OPA2 负端输入选择
	00: 引脚 PA6 作为负端输入 (默认)
	01: 引脚 PA7 作为负端输入
	10: OPA_NIN 作为负端输入 (详见 ANACTRL_CSR Bit[9:8])
	11: 引脚 PB7 作为负端输入
Bits 23:21	保留, 必须保持复位值
Bits 20:16	OP2OS[4:0]: OPA2 电压偏移修调 (TRIM) 值, 10000 为中间值, 增加为正偏, 减小为反偏 (默认值保存在 Flash 的 Option Byte 中)
	00000: OP2N-OP2P 电压偏差最大 (默认)
	...
	11111: OP2P-OP2N 电压偏差最大
	在修调过程中, OP2OS[4:0]的值由 00000->11111 变化, OP2TRIMO 变为 1 即修调结束
Bits 15:11	保留, 必须保持复位值
Bits 10:8	OP2GS[2:0]: 运放反馈电阻值 (R2) 选择, 即运放 OPA2 放大倍数选择 (放大倍数 = R2/R1)
	000: x1 (默认)
	001: x2
	010: x6
	011: x8
	100: x10
	101: x16
	110: x20
	111: x32
Bit 7	保留, 必须保持复位值
Bit 6	VM_SEL: 偏置电压源选择
	0: 运放共模偏置来自 OPA_VRH/2 (默认)
	1: 运放共模偏置来自 DAC_OUT1
Bit 5	OP2PS: OPA2 正端偏置使能
	0: 正端偏置禁止 (默认)
	1: 正端偏置使能
Bit 4	OP2NS: OPA2 负端反馈电阻使能

	0: 反馈电阻断开 (默认)
	1: 反馈电阻连接
Bit 3	OP2OE: OPA2 运放输出至 OP2OEX 使能
	0: 连接禁止 (默认)
	1: 连接使能
Bit 2	OP2PRS: OPA2 正端输入电阻短路
	0: 不短路 (默认)
	1: 短路
Bit 1	OP2NRS: OPA2 负端输入电阻短路
	0: 不短路 (默认)
	1: 短路
Bit 0	OP2EN: 运放 OPA2 使能
	0: 禁止 (默认)
	1: 使能

13. 高级定时器 TIM1

LCM32F037 包括 1 个高级控制定时器、5 个通用定时器、1 个基本定时器、2 个看门狗定时器和 1 个系统时基定时器。在调试模式下，可以冻结所有定时器计数器。

表 13-1 比较了高级控制定时器、通用定时器和基本定时器的特性。

表 13-1 定时器特性比较

定时器类型	名称	计数器精度	计数器方向	预分频系数	DMA 请求	捕获/比较通道	互补输出
高级定时器	TIM1	16-bit	Up, Down, Up/down	1~65536	Yes	4	3
通用定时器	TIM3	16-bit	Up, Down, Up/down	1~65536	Yes	4	0
	TIM14	16-bit	Up	1~65536	No	1	0
	TIM15	16-bit	Up, Down, Up/down	1~65536	Yes	2	2
	TIM16	16-bit	Up, Down, Up/down	1~65536	Yes	1	1
	TIM17	16-bit	Up, Down, Up/down	1~65536	Yes	1	1
基础定时器	TIM6	16-bit	Up	1~65536	Yes	0	0

13.1 TIM1 主要特性

高级定时器 TIM1 是一个 16 位的定时器/计数器，由一个可编程的预分频器驱动，有四路不同的捕获/比较通道。主要用于基本定时，测量输入信号的脉冲宽度（输入捕获）和产生输出波形（输出比较，PWM 和单脉冲模式），对应于不同事件（捕获、比较、溢出、刹车、触发）的中断以及与其它定时器或外部信号（外部时钟、复位、触发和使能信号）同步。

TIM1 可以看作是一个复用 6 个通道的 3 相 PWM，支持互补的 PWM 输出，能插入可编程的死区时间。也可以看做是一个完整的通用定时器。4 个独立的通道可以被分别用于：

- 输入捕获
- 输出比较
- PWM 生成（边沿或中心对齐模式）
- 单脉冲模式输出

如果配置为一个标准的 16 位定时器，TIM1 具有 TIMx 定时器的所有特性。如果配置为 16 位 PWM 生成器，TIM1 具有全调制能力（0 到 100%）。TIM1 支持延时触发和防误触发机制。在调试模式下，计数器可以被冻结。TIM1 可以通过定时器联动机制与其他定时器共同工作，提供同步或事件联接功能。

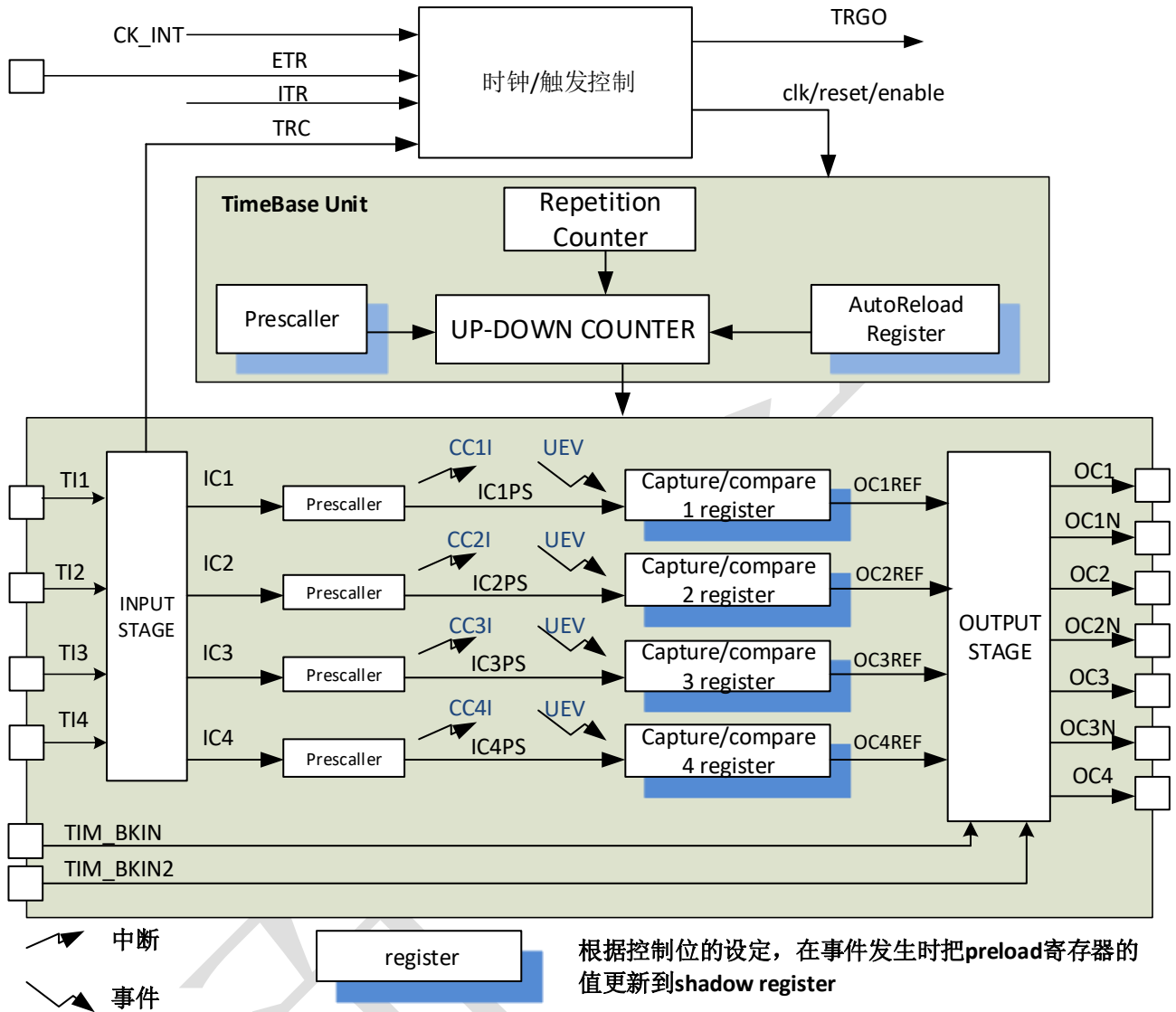


图 13-1 高级定时器 TIM1 框图

13.2 时基单元

时基单元包含：

- 16 位向上/向下计数器 (TIM1_CNT)
- 16 位预分频器 (TIM1_PSC)
- 16 位自动重载寄存器 (TIM1_ARR)
- 8 位重复计数器 (TIM1_RCR)

16 位计数器、预分频器、自动重载寄存器和 8 位重复计数器寄存器都可以通过软件进行读写操作。计数器由预分频器的输出 CK_CNT 驱动，而 CK_CNT 仅在 TIM1_CR1 寄存器的计数器使能位 (CEN) 被置位时才有效。

注：在使能了 CEN 位的一个时钟周期后，计数器才开始计数。

写计数器的操作没有缓存，可以在任何时候写 TIM1_CNT 寄存器，因此建议不要在计数器运行时写入新的数值，以免写入了错误的数值。

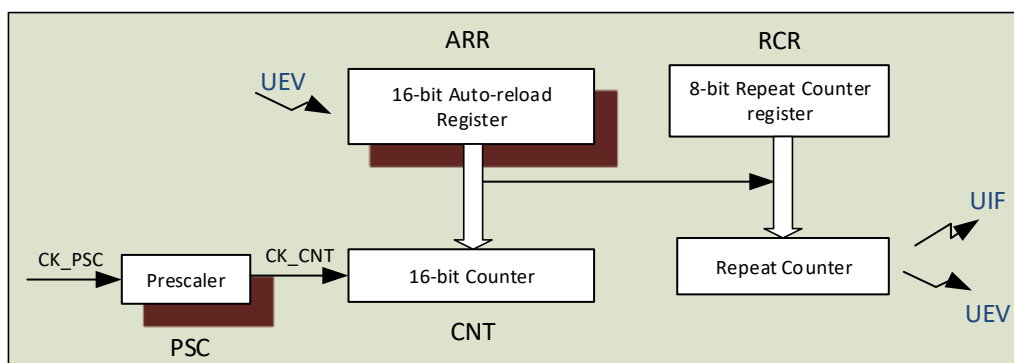


图 13-2 定时器时基单元

13.2.1 自动装载寄存器

自动装载寄存器是预先装载的。写或读自动重载寄存器将访问预装载寄存器。根据在 TIM1_CR1 寄存器中的自动装载预装载使能位 (ARPE) 的设置, 预装载寄存器的内容被立即或在每次的更新事件 UEV 时传送到影子寄存器。当计数器达到溢出条件 (例如向下计数时的下溢条件) 并当 TIM1_CR1 寄存器中的 UDIS=0 时, 产生更新事件。更新事件也可以由软件产生, 随后会详细描述每一种配置下更新事件的产生。注意这里影子寄存器 (Shadow register) 即为有效的工作寄存器 (Active register)。

写自动重载寄存器的两种模式:

- 自动预装载已使能 (TIM1_CR1 寄存器的 ARPE 位置位)。在此模式下, 写入自动重载寄存器的数据将被保存在预装载寄存器中, 并在下一个更新事件 (UEV) 时传送到影子寄存器。
- 自动预装载已禁止 (TIM1_CR1 寄存器的 ARPE 位清除)。在此模式下, 写入自动重载寄存器的数据将立即写入影子寄存器。

更新事件的产生条件:

- 计数器向上或向下溢出
- 软件置位了 TIM1_EGR 寄存器的 UG 位
- 时钟/触发控制器产生硬件复位

在预装载使能时 (ARPE=1), 如果发生了更新事件, 预装载寄存器中的数值 (TIM1_ARR) 将写入影子寄存器中。置位 TIM1_CR1 寄存器的 UDIS 位将禁止更新事件 (UEV)。

13.2.2 预分频器

TIM1 的预分频器基于一个由 16 位寄存器 (TIM1_PSC) 控制的 16 位计数器。由于这个控制寄存器带有缓冲器, 因此它能够在运行时被改变。预分频器可以将计数器的时钟频率按 1 到 65536 之间的任意值分频。计数器的频率可以由下式计算:

$$f_{\text{CK_CNT}} = f_{\text{CK_PSC}} / (\text{PSC}[15:0] + 1)$$

预分频器的值由预装载寄存器写入, 新的预分频器的值在下次更新事件到来时被采用。对 TIM1_PSC 寄存器的读操作通过预装载寄存器完成, 因此不需要特别的关注。

下图给出了在预分频器工作时，更改其参数的情况下计数器操作的例子，其它分频系数类推。

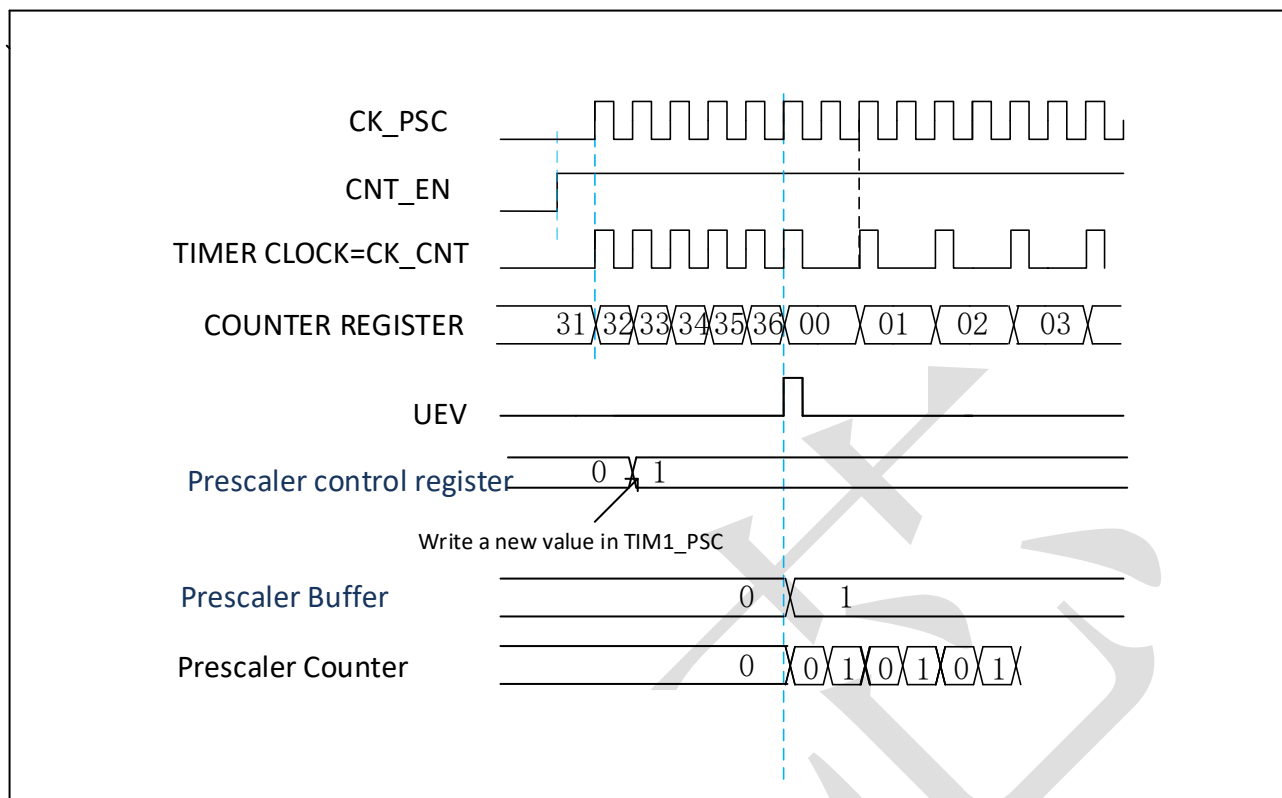


图 13-3 预分频系数从 1 变 2 的计数器时序

13.3 计数器模式

13.3.1 向上计数模式

在向上计数模式中，计数器从 0 计数到用户定义的比较值（TIMx_ARR 寄存器的值），然后重新从 0 开始计数并产生一个计数器溢出事件，同时，如果 TIMx_CR1 寄存器的 UDIS 位是 0，将会产生一个更新事件（UEV）。

如果定时器带有重复计数器，在重复了指定次数（TIMx_RCR 的值）的向上溢出之后会产生更新事件（UEV）。否则每一次的向上溢出都会产生更新事件。

置位 TIMx_EGR 寄存器的 UG 位（通过软件方式或者使用从模式控制器）也同样可以产生一个更新事件。

使用软件置位 TIMx_CR1 寄存器的 UDIS 位，可以禁止更新事件，这样可以避免在更新预装载寄存器时更新影子寄存器。在 UDIS 位被清除之前，将不产生更新事件。但是在应该产生更新事件时，计数器仍会被清 0，同时预分频器的计数也被清 0（但预分频器的数值不变）。此外，如果设置了 TIMx_CR1 寄存器中的 URS 位（选择更新请求源），设置 UG 位将产生一个更新事件 UEV，但硬件不设置 UIF 标志（即不产生中断请求）。这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。

如果定时器带有重复计数器，由于重复寄存器没有双重的缓冲，新的重复数值将立刻生效，因此在修改时需要小心。

当发生一个更新事件时，所有的寄存器都被更新，硬件同时依据 URS 位设置更新标志位（TIMx_SR 寄存器的 UIF 位）：自动装载影子寄存器被重新置入预装载寄存器的值（TIMx_ARR）。预分频器的缓存器被置入预装载寄存器的值（TIMx_PSC）。要注意到如果是因为计数器溢出而产生更新，自动重装载寄存器将在计数器重载入之前被更新，因此下一个计数周期才是预期的值（计数器被装载为新的值）。非自动重装载模式下可以立即更新。

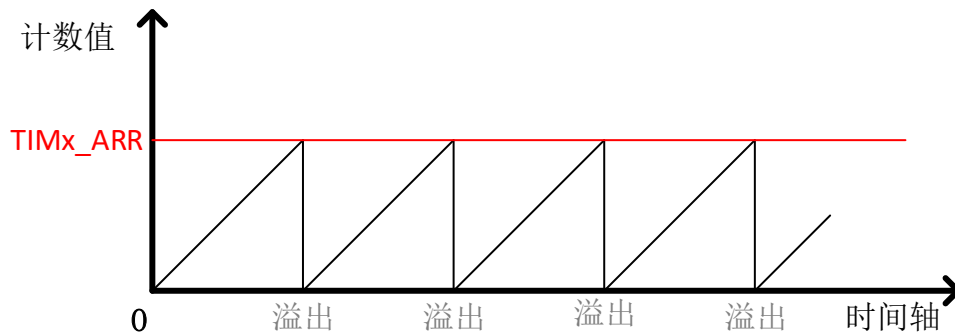


图 13-4 向上计数溢出

下图给出例子，当 $TIMx_ARR=0x36$ 时计数器在二分频时钟频率下的动作。

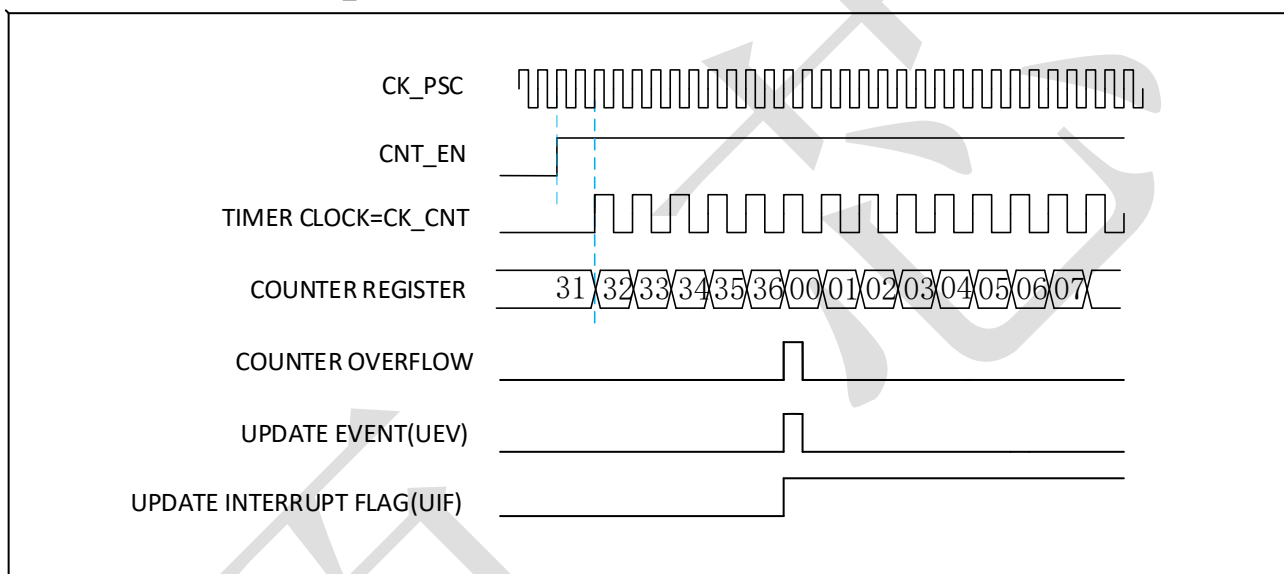


图 13-5 计数器在二分频时钟频率下的时序图

13.3.2 向下计数模式

在向下计数模式中，计数器从自动装载的值（ $TIMx_ARR$ 寄存器的值）开始向下计数到 0，然后再从自动装载的值重新开始计数，并产生一个计数器向下溢出事件。如果 $TIMx_CR1$ 寄存器的 $UDIS$ 位是 0，将会产生一个更新事件（UEV）。

如果定时器带有重复计数器，在重复了指定次数（ $TIMx_RCR$ 的值）的向下溢出之后会产生更新事件（UEV）。否则每一次的向下溢出都会产生更新事件。

置位 $TIMx_EGR$ 寄存器的 UG 位(通过软件方式或者使用从模式控制器)也同样可以产生一个更新事件。

置位 $TIMx_CR1$ 寄存器的 $UDIS$ 位可以禁止 UEV 事件。这样可以避免在更新预装载寄存器时更新影子寄存器。因此 $UDIS$ 位清除之前不会产生更新事件。然而，计数器仍会从当前自动加载值重新开始计数，并且预分频器的计数器重新从 0 开始（但预分频器的数值不变）。此外，如果设置了 $TIMx_CR1$ 寄存器中的 URS 位(选择更新请求源)，设置 UG 位将产生一个更新事件 UEV，但硬件不设置 UIF 标志(即不产生中断请求)。这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。

如果定时器带有重复计数器，由于重复寄存器没有双重的缓冲，新的重复数值将立刻生效，因此在修改时需要小心。

当发生一个更新事件时，所有的寄存器都被更新，硬件同时依据 URS 位设置更新标志位（TIMx_SR 寄存器的 UIF 位）：自动装载影子寄存器被重新置入预装载寄存器的值（TIMx_ARR）。预分频器的缓存器被置入预装载寄存器的值（TIMx_PSC）。要注意到如果是因为计数器溢出而产生更新，自动重装载寄存器将在计数器重载入之前被更新，因此下一个计数周期才是预期的值（计数器被装载为新的值）。非自动重装载模式下可以立即更新。

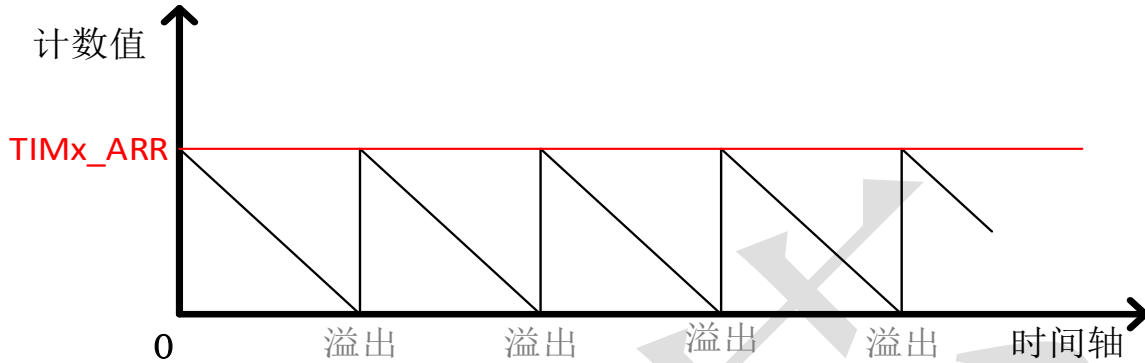


图 13-6 向下计数溢出

13.3.3 中央对齐模式（向上/向下计数）

在中央对齐模式中，计数器从 0 开始计数到自动加载的值（TIMx_ARR-1），产生一个计数器溢出事件，然后向下计数到 1 并且产生一个计数器下溢事件；然后再从 0 开始重新计数。

在此模式下，不能写入 TIMx_CR1 中的 DIR 方向位。它由硬件更新并指示当前的计数方向。

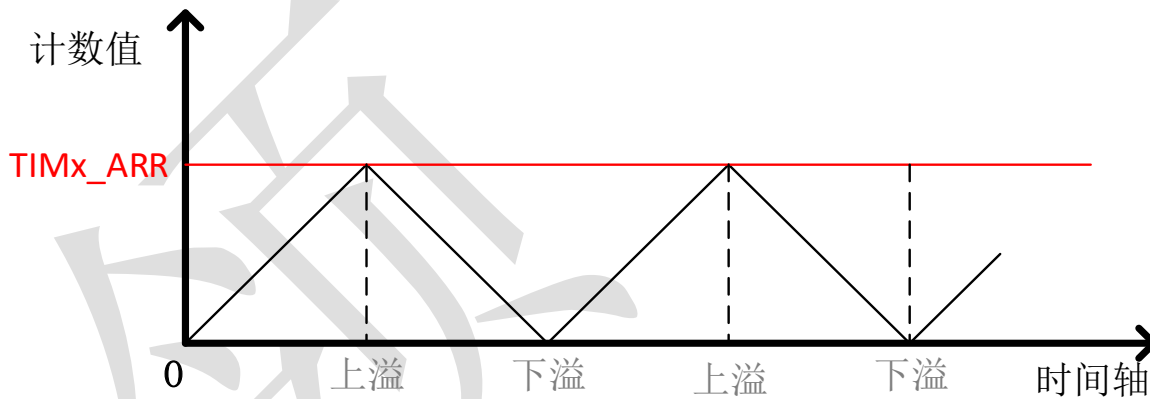


图 13-7 中央对齐溢出

如果定时器带有重复计数器，在重复了指定次数（TIMx_RCR 的值）的向上和向下溢出之后会产生更新事件（UEV）。否则每一次的向上向下溢出都会产生更新事件。

置位 TIMx_EGR 寄存器的 UG 位（通过软件方式或者使用从模式控制器）也同样可以产生一个更新事件。此时，计数器重新从 0 开始计数，预分频器也重新从 0 开始计数。

设置 TIMx_CR1 寄存器中的 UDIS 位可以禁止 UEV 事件，这样可以避免在更新预装载寄存器时更新影子寄存器。因此 UDIS 位被清为 0 之前不会产生更新事件。然而，计数器仍会根据当前自动重加载的值，继续向上或向下计数。

如果定时器带有重复计数器，由于重复寄存器没有双重的缓冲，新的重复数值将立刻生效，因此在修改时需要小心。

此外，如果设置了 TIMx_CR1 寄存器中的 URS 位（选择更新请求源），设置 UG 位将产生一个更新事件 UEV，但硬件不设置 UIF 标志（即不产生中断请求）。这是为了避免在捕获模式下清除计数器时，同时产生更

新和捕获中断。

当发生一个更新事件时，所有的寄存器都被更新，硬件同时依据 **URS** 位设置更新标志位 (**TIMx_SR** 寄存器的 **UIF** 位)：自动装载影子寄存器被重新置入预装载寄存器的值 (**TIMx_ARR**)。预分频器的缓存器被置入预装载寄存器的值 (**TIMx_PSC**)。要注意到如果是因为计数器溢出而产生更新，自动重载寄存器将在计数器重载入之前被更新，因此下一个计数周期才是预期的值 (计数器被装载为新的值)。

使用中央对齐模式：

- 启动中央对齐模式时，计数器将按照原有的向上/向下的配置计数。也就是说 **TIMx_CR1** 寄存器中的 **DIR** 位将决定计数器是向上还是向下计数。此外，软件不能同时修改 **DIR** 位和 **CMS** 位的值。
- 不推荐在中央对齐模式下，计数器正在计数时写计数器的值，这将导致不能预料后果。具体的说：
 - 向计数器写入了比自动装载值更大的数值时 ($TIMx_CNT > TIMx_ARR$)，但计数器的计数方向不发生改变。例如计数器已经向上溢出，但计数器仍然向上计数。
 - 向计数器写入了 0 或者 **TIMx_ARR** 的值，但更新事件不发生。
- 安全使用中央对齐模式的计数器的方法是在启动计数器之前先用软件置位 **TIMx_EGR** 寄存器的 **UG** 位产生一个更新事件，并且不在计数器计数时修改计数器的值。

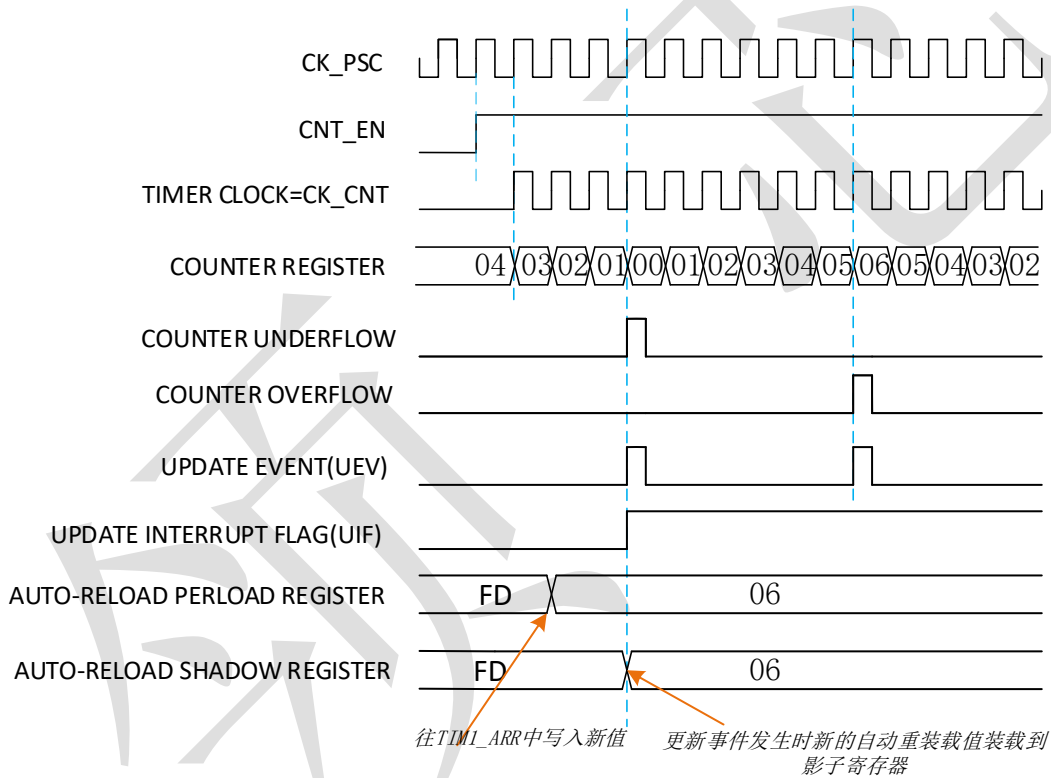


图 13-8 计数器时序图，内部时钟分频为 1，**TIMx_ARR=0x6**

13.4 重复计数器

13.2 章节解释了计数器向上/向下溢出时更新事件 (UEV) 是如何产生的，然而事实上它只能在重复计数器的值达到 0 的时候产生。这个特性对产生 PWM 信号非常有用。

这意味着在每 **N** 次计数上溢或下溢时，数据从预装载寄存器传输到影子寄存器 (**TIMx_ARR** 自动重载入寄存器，**TIMx_PSC** 预装载寄存器，还有在比较模式下的捕获/比较寄存器 **TIMx_CCRx**)，**N** 是 **TIM1_RCR** 重复计数寄存器中的值。

重复计数器在下述任一条件成立时递减：

- 向上计数模式下每次计数器向上溢出时
- 向下计数模式下每次计数器向下溢出时
- 中央对齐模式下每次上溢和每次下溢时

虽然这样限制了 PWM 的最大循环周期为 128，但它能够在每个 PWM 周期中 2 次更新占空比。在中央对齐模式下，因为波形是对称的，如果每个 PWM 周期中仅刷新一次比较寄存器，则最大的分辨率为 $2xt_{CK_PSC}$ 。

重复计数器是通过更新事件自动加载的，重复速率由 TIM1_RCR 寄存器的值定义，参考图 13-9。当更新事件由软件产生（通过设置 TIMx_EGR 中的 UG 位）或者通过硬件的从模式控制器产生，则无论重复计数器的值是多少，立即发生更新事件，并且 TIM1_RCR 寄存器中的内容被重载入到重复计数器。

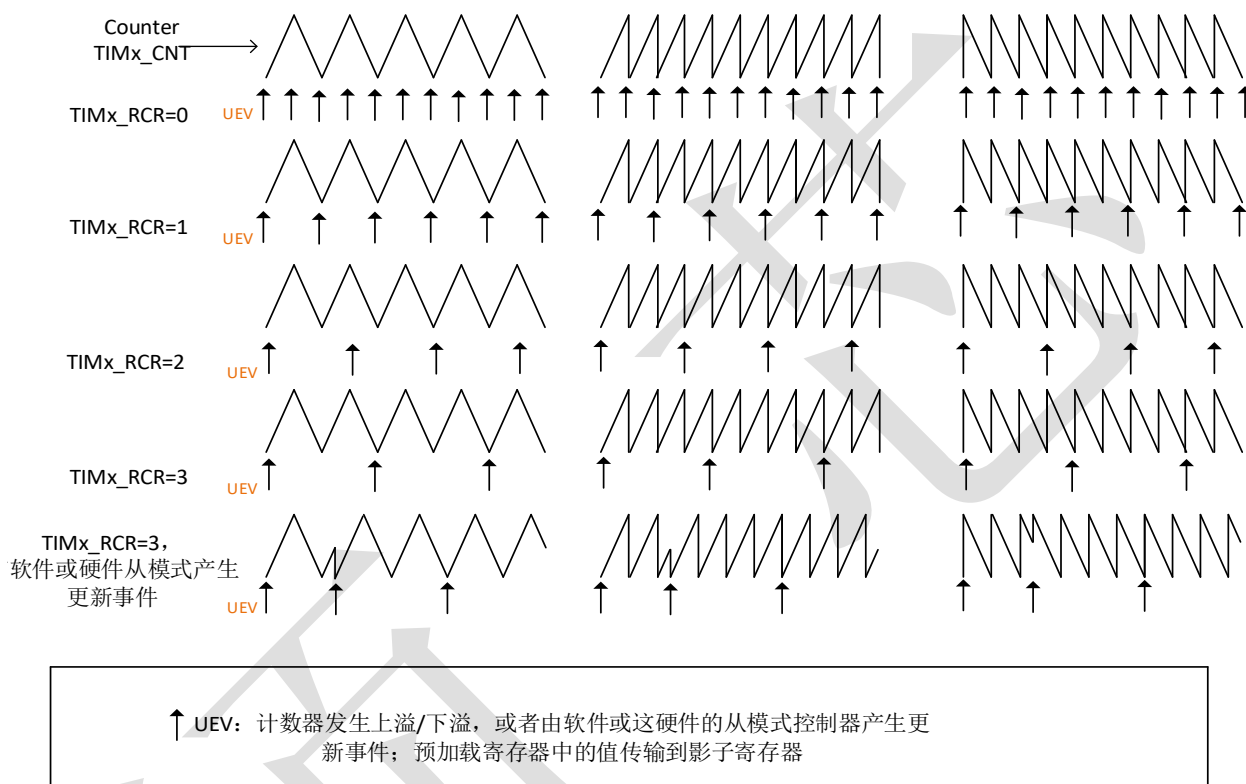


图 13-9 不同模式下更新速率的例子，以及寄存器 TIM1_RCR 的设置

13.5 时钟选择

时钟/触发控制器允许用户选择计数器的时钟源(CK_PSC)、触发输入信号(TRGI)和触发输出信号(TRGO)，其框图如图 13-10 所示。

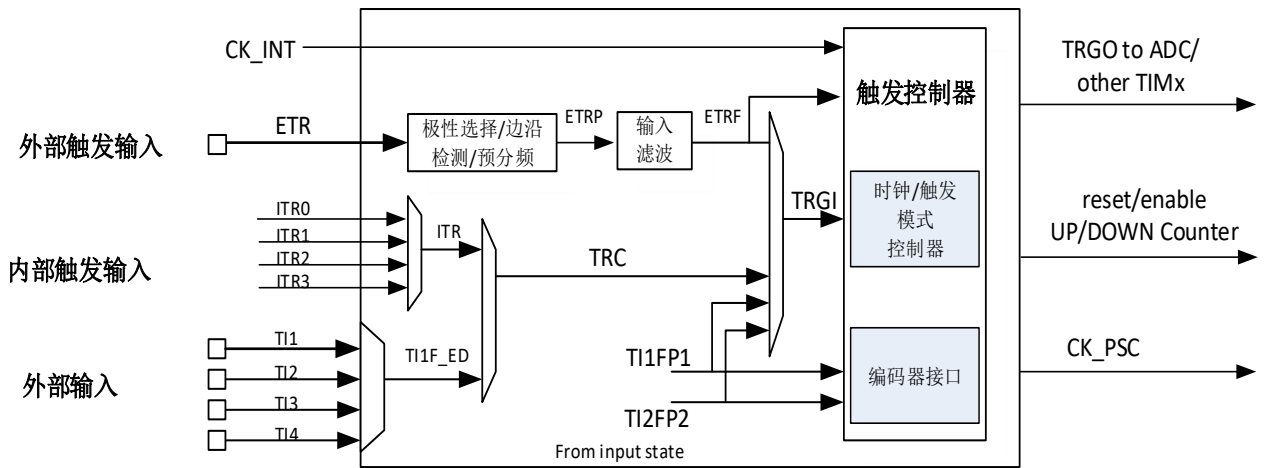


图 13-10 时钟/触发控制器框图

时基单元的预分频时钟（CK_PSC）可以有以下来源：

- 内部时钟（CK_INT）：芯片内部时钟
- 外部时钟模式 1：外部触发输入（Tii）和内部触发输入（ITRx）
- 外部时钟模式 2：外部触发输入（ETR）

13.5.1 内部时钟源（CK_INT）

如果 TIMx_SMCR 寄存器 SMS=000，TIMx_SMCR 寄存器 ECE=0，则 CEN、DIR 和 UG 位是实际上的控制位，并且只能被软件修改（UG 位仍被自动清除）。一旦 CEN 位被写成 1，预分频器的时钟就由内部时钟提供。

13.5.2 外部时钟模式 1

当 TIMx_SMCR 寄存器的 SMS=111 时，此模式被选中。计数器可以在选定输入端的每个上升沿或下降沿计数。

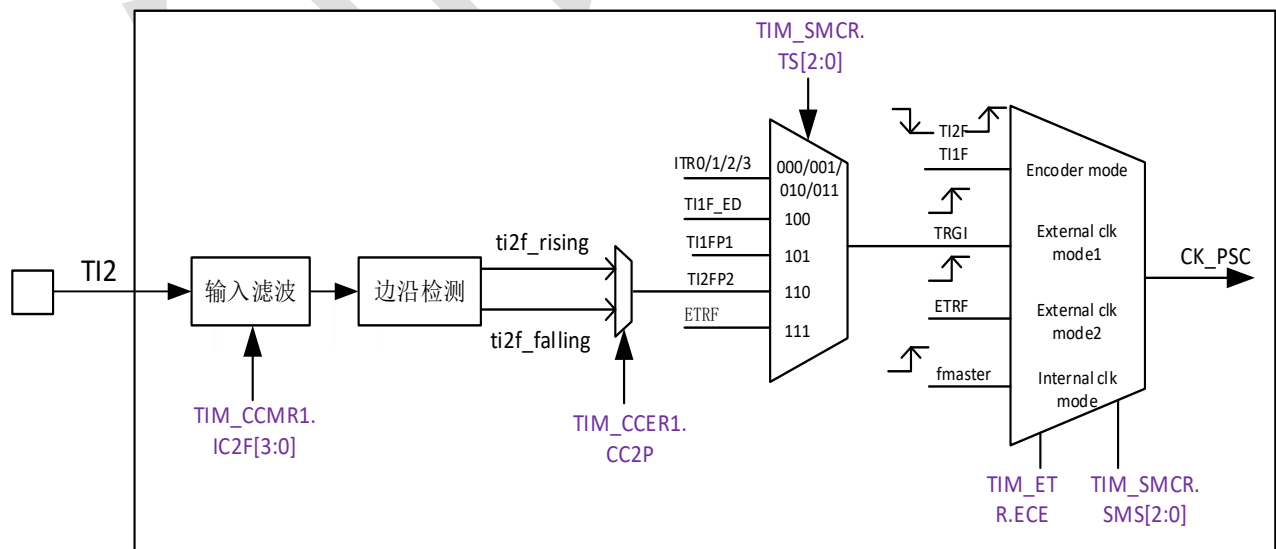


图 13-11 TI2 外部时钟框图

例如，要配置向上计数器在 TI2 输入端的上升沿计数，使用下列步骤：

1. 配置 TIMx_CCMR1 寄存器的 CC2S=01，使用通道 2 检测 TI2 输入的上升沿
2. 配置 TIMx_CCMR1 寄存器的 IC2F[3:0]位，选择输入滤波器带宽（如果不需要滤波器，保持 IC2F=0000）
3. 配置 TIMx_CCER 寄存器的 CC2P=0，选定上升沿极性
4. 配置 TIMx_SMCR 寄存器的 SMS=111，配置计数器使用外部时钟模式 1
5. 配置 TIMx_SMCR 寄存器的 TS=110，选定 TI2 作为输入源
6. 设置 TIMx_CR1 寄存器的 CEN=1，启动计数器

当上升沿出现在 TI2，计数器计数一次，且触发标识位（TIM1_SR 寄存器的 TIF 位）被置 1，如果使能了中断（在 TIMx_DIER 寄存器中配置）则会产生中断请求。

在 TI2 的上升沿和计数器实际时钟之间的延时取决于在 TI2 输入端的重新同步电路（下图预分频系数为 1）。

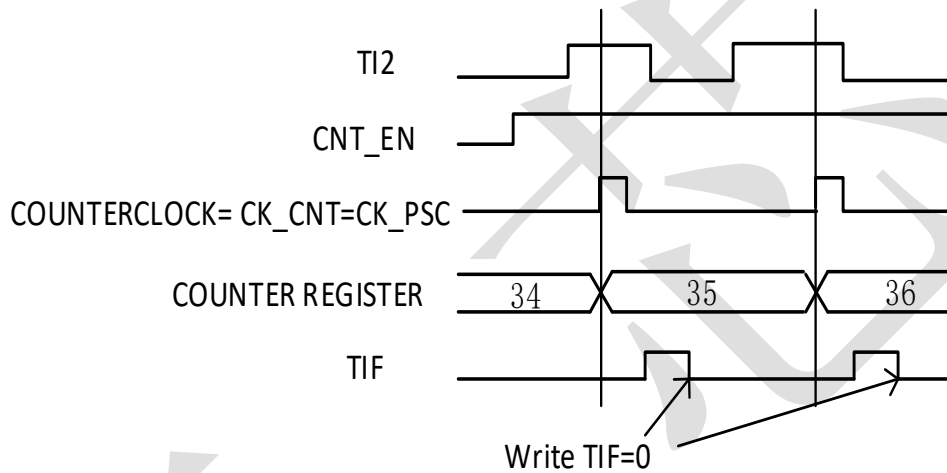


图 13-12 外部时钟模式 1 下的控制电路

13.5.3 外部时钟模式 2

计数器能够在外部触发输入 ETR 信号的每一个上升沿或下降沿计数。将 TIMx_SMCR 寄存器的 ECE 位写 1，即可选定此模式。

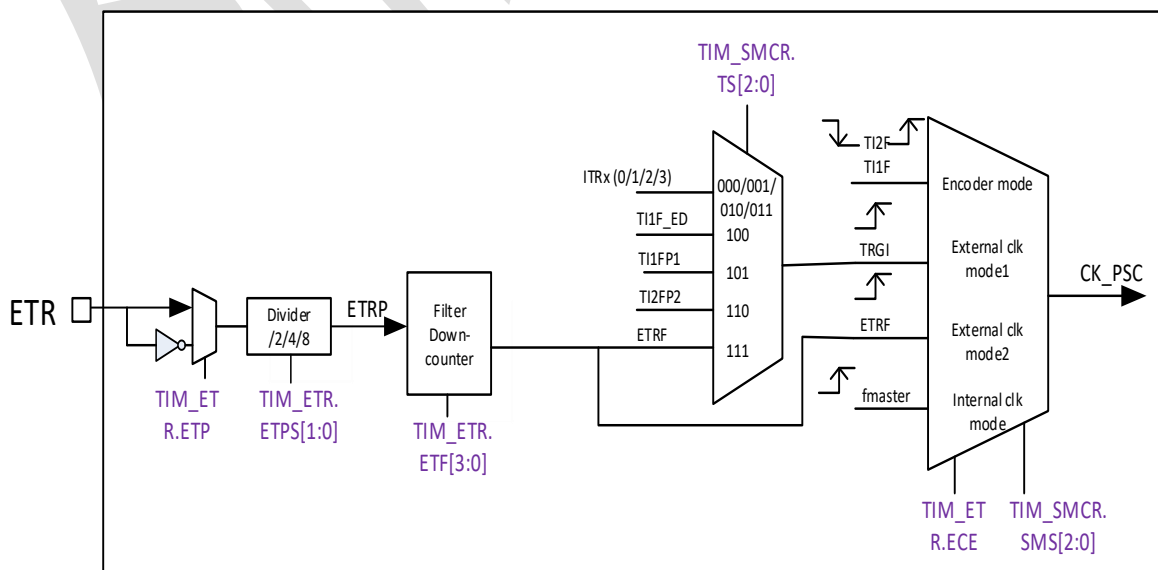


图 13-13 外部触发时钟框图

例如，要配置计数器在 ETR 信号的每 2 个上升沿时向上计数一次，需使用下列步骤：

1. 本例中不需要滤波器，配置 TIMx_SMCR 寄存器的 ETF[3:0]=0000
2. 设置预分频器，配置 TIMx_SMCR 寄存器的 ETPS[1:0]=01
3. 选择 ETR 的上升沿检测，配置 TIMx_SMCR 寄存器的 ETP=0
4. 开启外部时钟模式 2，配置 TIMx_SMCR 寄存器中的 ECE=1
5. 启动计数器，写 TIMx_CR1 寄存器的 CEN=1

计数器在每 2 个 ETR 上升沿计数一次。在 ETR 的上升沿和计数器实际时钟之间的延时取决于在 ETRP 信号端的重新同步电路。

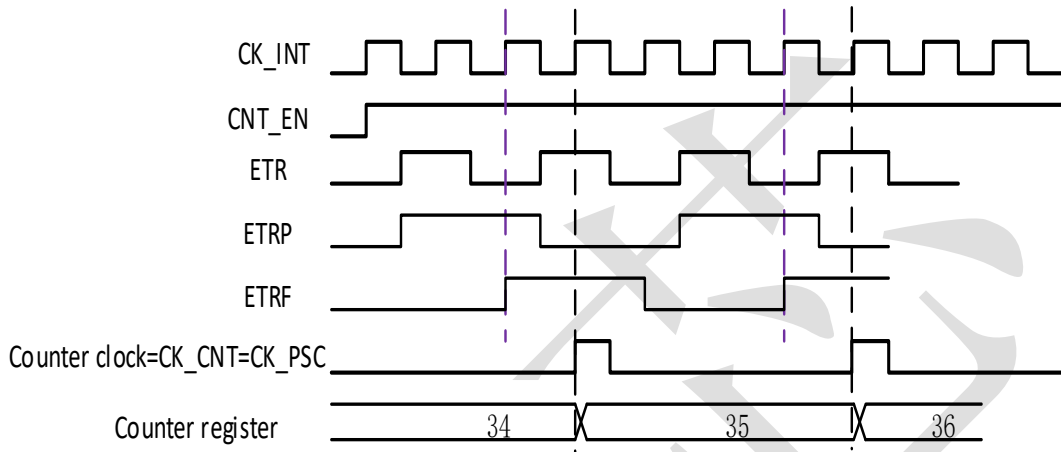


图 13-14 外部时钟模式 2 下的控制电路

13.6 捕获/比较通道

定时器的 I/O 引脚 (TIMx_CCI) 可以用作输入捕获或者输出比较，这个功能可以通过配置捕获/比较通道模式寄存器 (TIMx_CCMRi) 的 CCIS 通道选择位来实现，此处的 i 代表通道数。

每一个捕获/比较通道都是围绕着一个捕获/比较寄存器 (包含影子寄存器) 来构建的，包括捕获的输入部分 (数字滤波、多路复用和预分频器)，和输出部分 (比较器和输出控制)。

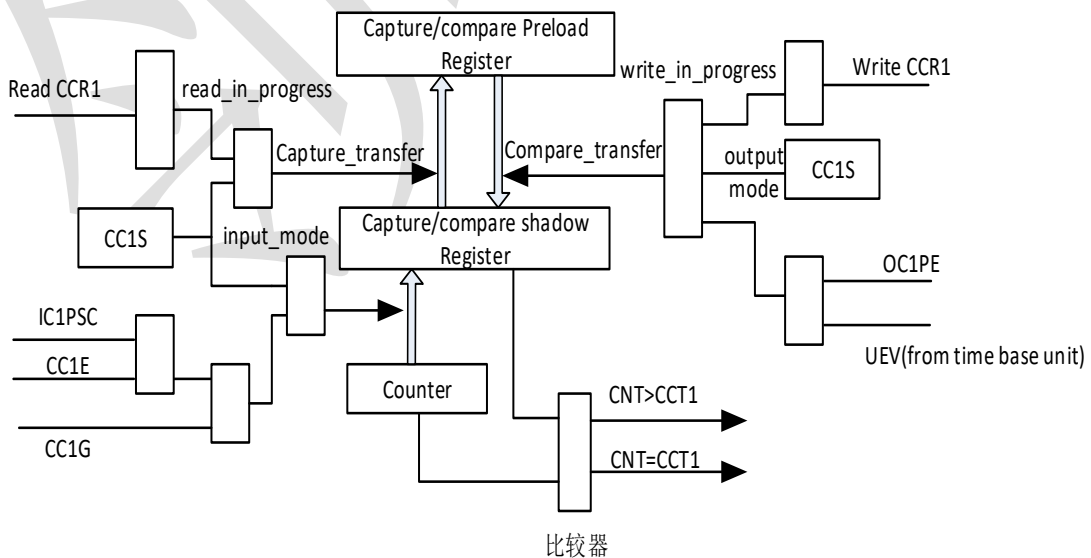


图 13-15 捕获/比较通道 1 的主电路

捕获/比较模块由一个预装载寄存器和一个影子寄存器组成，读写过程仅操作预装载寄存器。在捕获模式下，捕获计数器的值就在影子寄存器上，然后再复制到预装载寄存器中。在比较模式下，预装载寄存器的内容被复制到影子寄存器中，然后影子寄存器内容和计数器进行比较。

当通道被配置成输出模式时（TIMx_CCMRi 寄存器的 CCIS=0），可以随时访问 TIMx_CCRi 寄存器（此处的 i 指通道数）。

当通道被配置成输入模式时，对 TIMx_CCRi 寄存器的读操作类似于计数器的读操作。当捕获发生时，计数器的内容被捕获到 TIMx_CCRi 影子寄存器，然后再复制到预装载寄存器中。在读操作进行中，预装载寄存器是被冻结的（以防读数据过程中预装载寄存器内容被修改）。

13.6.1 输入模块

输入部分对相应的 Ti 输入信号采样，并产生一个滤波后的信号 TiIF。然后，一个带极性选择的边沿检测器产生一个信号（TiIFPx），它可以作为触发模式控制器的输入触发或者作为捕获控制。该信号通过预分频进入捕获寄存器（ICiPS）。

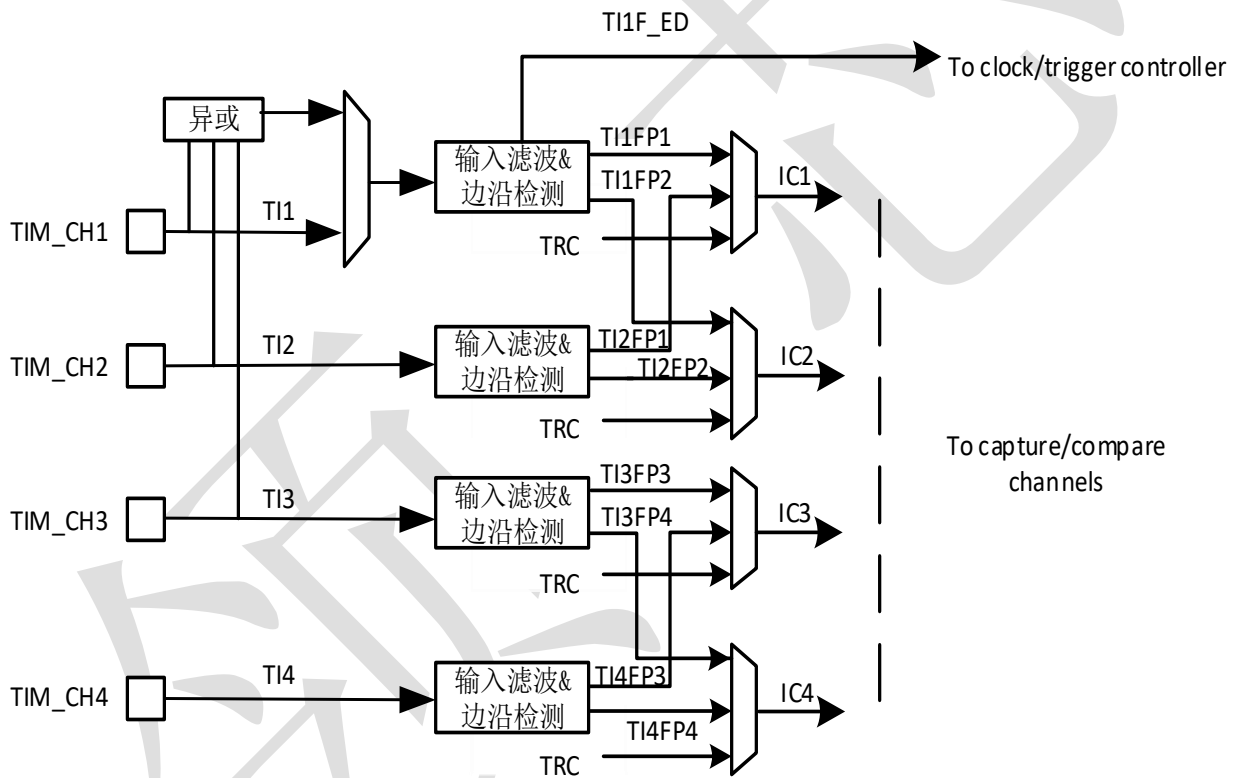


图 13-16 输入模块框图

13.6.2 输入捕获模式

在输入捕获模式下，当检测到 ICi 信号上相应的边沿后，计数器的当前值被锁存到捕获/比较寄存器（TIMx_CCRi）中。当发生捕获事件时，相应的 CCiIF 标志（TIMx_SR 寄存器）被置 1。

如果 TIMx_DIER 寄存器的 CCiIE 位被置位，也就是使能了中断，则将产生中断请求。如果发生捕获事件时 CCiIF 标志已经为高，那么重复捕获标志 CCiOF（TIMx_SR 寄存器）被置 1，写 CCiIF=0 可清除 CCiOF。写 CCiIF=0 或读取存储在 TIMx_CCRi 寄存器中的捕获数据都可清除 CCiIF。

以下例子说明如何在 Ti1 输入的上升沿时捕获计数器的值到 TIMx_CCR1 寄存器中，步骤如下：

1. 选择有效输入端：例如 TIMx_CCR1 连接到 TI1 输入，所以写入 TIMx_CCMR1 寄存器中的 CC1S=01，此时通道被配置为输入，并且 TIMx_CCR1 寄存器变为只读。
2. 根据输入信号 Tii 的特点，可通过配置 TIMx_CCMRi 寄存器中的 ICiF 位来设置相应的输入滤波器的滤波时间。假设输入信号在最多 5 个时钟周期的时间内抖动，我们须配置滤波器的带宽长于 5 个时钟周期；因此我们可以连续采样 8 次，以确认在 TI1 上一次真实的边沿变换，即在 TIMx_CCMR1 寄存器中写入 IC1F=0011，此时，只有连续采样到 8 个相同的 TI1 信号，信号才为有效（采样频率为 DTS 时钟的频率 f_{DTS} ）。
3. 选择 TI1 通道的有效转换边沿，在 TIMx_CCER 寄存器中写入 CC1P=0（上升沿）。
4. 配置输入预分频器。在本例中，我们希望捕获发生在每一个有效的电平转换时刻，因此预分频器被禁止（写 TIMx_CCMR1 寄存器的 IC1PSC=00）。
5. 设置 TIMx_CCER 寄存器的 CC1E=1，允许捕获计数器的值到捕获寄存器中。
6. 如果需要，通过设置 TIMx_DIER 寄存器中的 CC1IE 位允许相关中断请求。

TIM1 通道 1 的输入如下图所示：

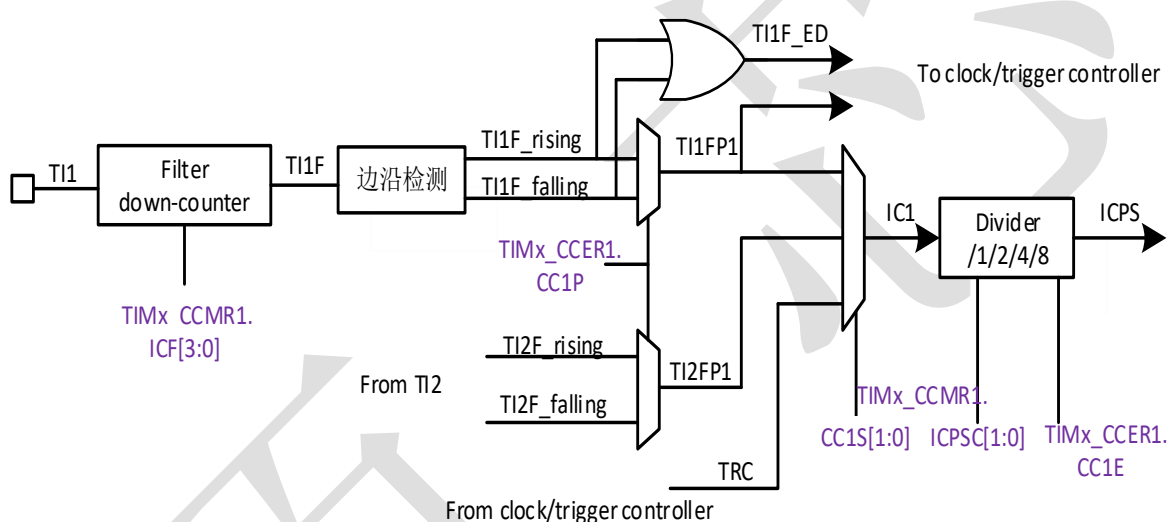


图 13-17 TIM1 通道 1 的输入

当发生一个输入捕获时：

- 当产生有效的电平转换时，计数器的值被传送到 TIMx_CCR1 寄存器。
- CC1IF 标志被设置（中断标志）。当发生至少 2 次连续的捕获，且 CC1IF 未被清除，CC1OF 也被置 1。
- 如设置了 CC1IE 位，则会产生一个中断。
- 为了处理捕获溢出（CC1OF 位），建议在读出重复捕获标志之前读取数据，这是为了避免丢失在读出捕获溢出标志之后和读取数据之前可能产生的重复捕获信息。

注：设置 TIMx_EGR 寄存器中相应的 CCiG 位，可以通过软件产生输入捕获中断。

13.6.3 PWM 输入

该模式是输入捕获模式的一个特例，除下列区别外，操作与输入捕获模式相同：

- 两个 ICi 信号被映射至同一个 Tii 输入
- 这两个 ICi 信号的有效边沿的极性相反
- 两路 TiiFP 信号中的一路被选择作为触发输入信号，并且触发模式控制器被配置成触发复位模式

例如，你可以用以下方式测量 TI1 上输入的 PWM 信号的周期（锁存于 TIMx_CCR1 寄存器）和占空比（锁

存于 TIMx_CCR2 寄存器)。(具体取决于 CK_INT 的频率 f_{CK_INT} 和预分频器的值)

1. 选择 TIMx_CCR1 的有效输入: 置 TIMx_CCMR1 寄存器的 CC1S=01 (选中 TI1)。
2. 选择 TI1FP1 的有效极性 (用来捕获数据到 TIMx_CCR1 中和清除计数器): 置 CC1P=0 (上升沿有效)。
3. 选择 TIMx_CCR2 的有效输入: 置 TIMx_CCMR1 寄存器的 CC2S=10 (选中 TI1FP2)。
4. 选择 TI1FP2 的有效极性 (捕获数据到 TIMx_CCR2): 置 CC2P=1 (下降沿有效)。
5. 选择有效的触发输入信号: 置 TIMx_SMCR 寄存器中的 TS=101 (选择 TI1FP1)。
6. 配置触发模式控制器为复位触发模式: 置 TIMx_SMCR 中的 SMS=100。
7. 使能捕获: 置 TIMx_CCER 寄存器中 CC1E=1, CC2E=1。

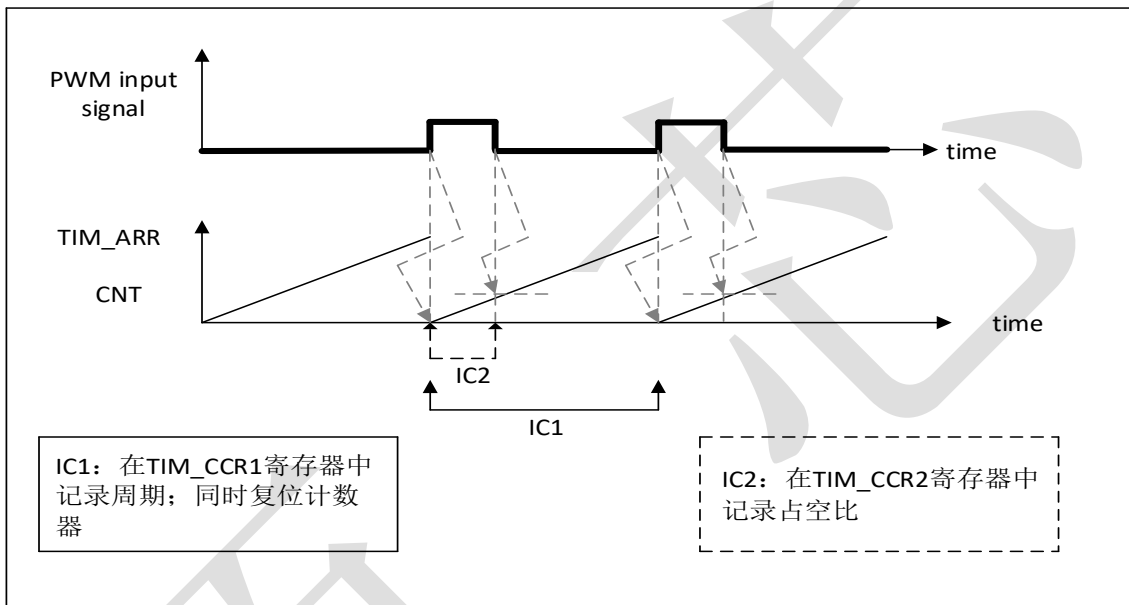


图 13-18 PWM 输入信号测量

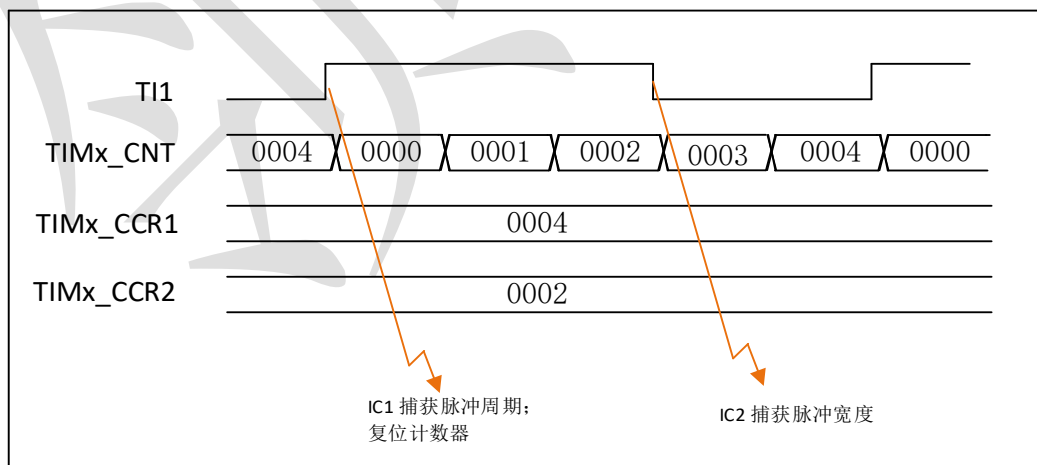


图 13-19 PWM 输入信号测量实例

13.6.4 输出模块

输出模块会产生一个用来做参考的中间波形, 称为 OCiREF (高电平有效)。刹车功能和极性选择都在之

后处理。

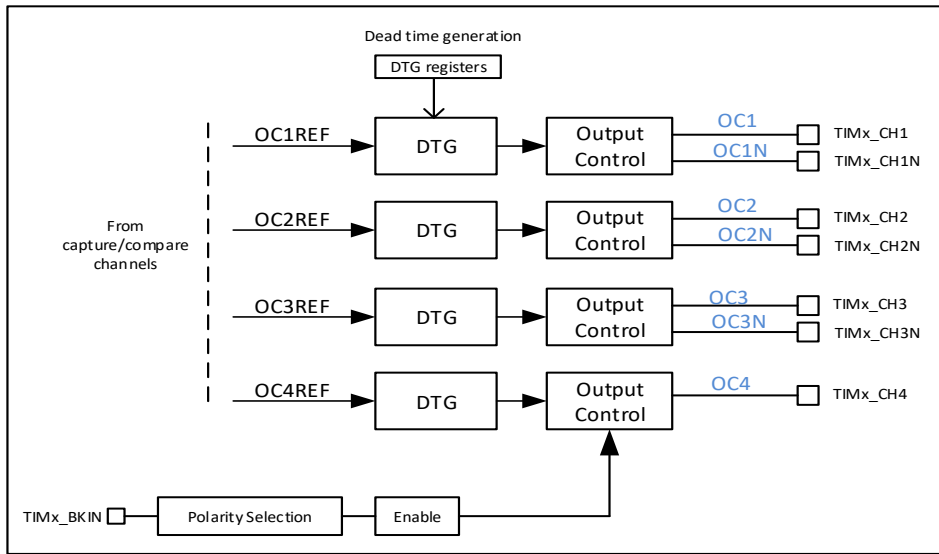


图 13-20 输出模块框图

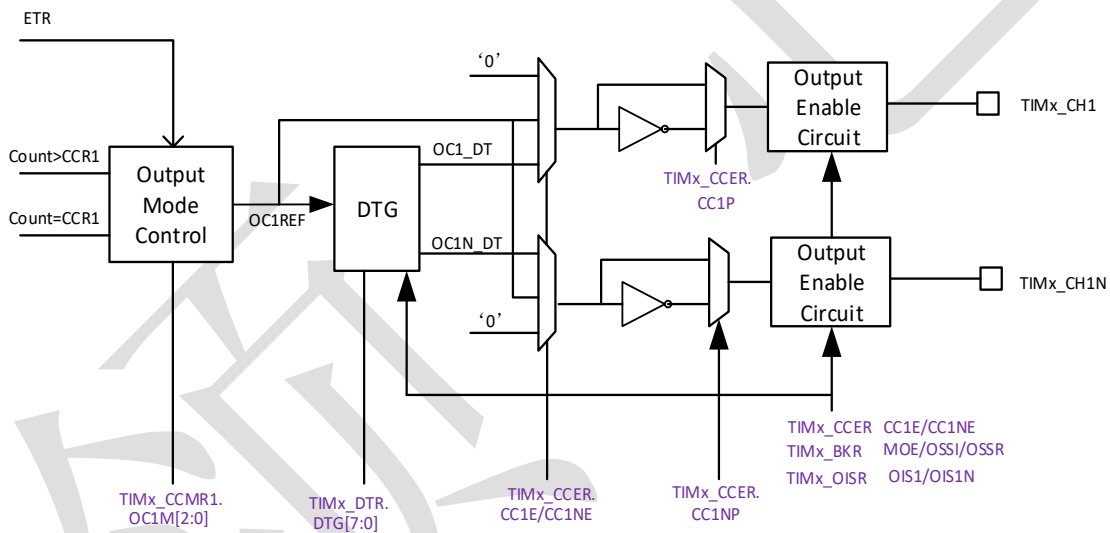


图 13-21 TIM1 通道 1 的输出

13.6.5 强制输出模式

在输出模式 (TIM1_CCMRi 寄存器中 CCIS=00) 下, 输出比较信号能够直接由软件强置为高或低状态, 而不依赖于输出比较寄存器和计数器间的比较结果。

置 TIMx_CCMRi 寄存器中相应的 OCiM=101, 即可强置输出比较信号为有效状态。这样 OCiREF 被强置为高电平 (OCiREF 始终为高电平有效), 而 OCi 的输出是高还是低则取决于 CCiP 极性标志位。例如 CCiP=0 (OCi 高电平有效), 则 OCi 被强置为高电平。

置 TIMx_CCMRi 寄存器的 OCiM=100, 可强置 OCiREF 信号为低。

该模式下, 在 TIMx_CCRi 影子寄存器和计数器之间的比较仍然在进行, 相应的标志也会被修改, 也仍然会产生相应的中断。这将会在下文的输出比较模式一节中介绍。

13.6.6 输出比较模式

此模式用来控制一个输出波形或者指示一段给定的时间已经达到。当计数器与捕获/比较寄存器的内容相同时，有如下操作：

- 根据不同的输出比较模式，相应的 OCiREF 输出信号为：
 - 保持不变 (OCiM=000)
 - 设置为有效电平 (OCiM=001)
 - 设置为无效电平 (OCiM=010)
 - 翻转 (OCiM=011)
- 设置中断状态寄存器中的标志位 (TIMx_SR 寄存器中的 CCIIF 位)
- 若设置了相应的中断使能位 (TIMx_DIER 寄存器中的 CCIIE 位)，则产生一个中断

TIMx_CCMRi 寄存器的 OCiM 位用于选择输出比较模式，而 TIMx_CCER 寄存器的 CCIp 位用于选择有效和无效的电平极性。

TIMx_CCMRi 寄存器的 OCiPE 位用于选择 TIMx_CCRi 寄存器是否需要使用预装载寄存器。

在输出比较模式下，更新事件 UEV 对 OCiREF 和 OCi 输出没有影响。输出比较的时间精度为计数器的一个时钟周期，输出比较模式也能用来输出一个单脉冲。输出比较模式的配置步骤如下：

1. 选择计数器时钟 (内部/外部/预分频器)
2. 将相应的数据写入 TIMx_ARR 和 TIMx_CCRi 寄存器中
3. 如果要产生一个中断请求，设置 CCIIE 位
4. 选择输出模式步骤：
 - 要求计数器与 CCRi 匹配时翻转 OCiM 的输出管脚，设置 OCiM=011
 - 置 OCiPE = 0 禁用预装载寄存器
 - 置 CCIp = 0 选择高电平为有效电平
 - 置 CCIIE = 1 使能输出
5. 设置 TIMx_CR1 寄存器的 CEN 位来启动计数器

在未使用预装载寄存器 (OCiPE=0，否则 TIMx_CCRi 的影子寄存器只能在发生下一次更新事件时被更新) 的情况下，TIMx_CCRi 寄存器能够在任何时候通过软件进行更新以控制输出波形。

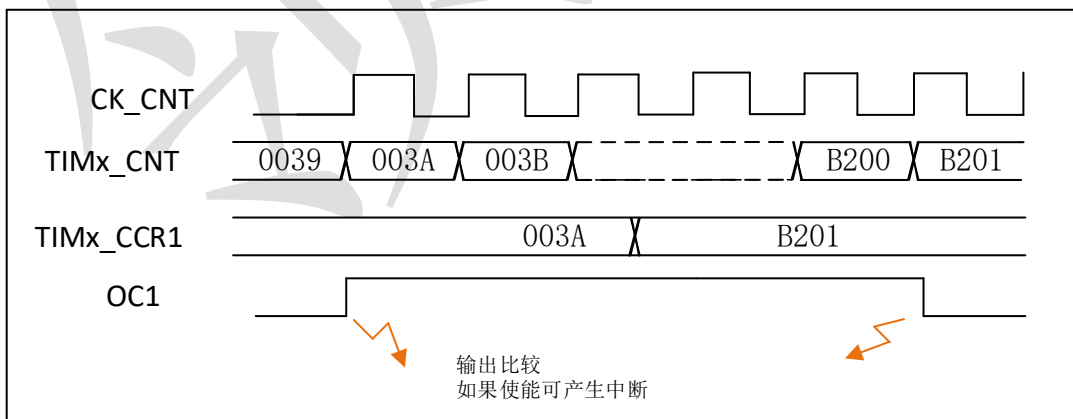


图 13-22 输出比较模式，翻转 OC1

13.6.7 PWM 模式

脉冲宽度调制 (PWM) 模式可以产生一个由 TIMx_ARR 寄存器确定频率, 由 TIMx_CCRi 寄存器确定占空比的信号。

在 TIMx_CCMRi 寄存器中的 OCiM 位写入 110 (PWM 模式 1) 或 111 (PWM 模式 2), 能够独立地设置每个 OCi 输出通道产生一路 PWM。

必须设置 TIMx_CCMRi 寄存器的 OCiPE 位使能相应的预装载寄存器, 也可以设置 TIMx_CR1 寄存器的 ARPE 位使能自动重载的预装载寄存器 (在向上计数模式或中央对齐模式中)。

由于仅当发生一个更新事件的时候, 预装载寄存器才能被传送到影子寄存器, 因此在计数器开始计数之前, 必须通过设置 TIMx_EGR 寄存器的 UG 位来初始化所有的寄存器。

OCi 的极性可以通过软件在 TIMx_CCER 寄存器中的 CCiP 位设置, 它可以设置为高电平有效或低电平有效。OCi 的输出使能通过 TIMx_CCER 和 TIM1_BDTR 寄存器中 CCiE、MOE、OSSR 和 OSSI 位的组合来控制。详见 TIMx_CCER 寄存器的描述。

在 PWM 模式 (模式 1 或模式 2) 下, TIMx_CNT 和 TIM1_CCRi 始终在进行比较, (依据计数器的计数方向) 以确定是否符合 $TIMx_CCRi \leq TIMx_CNT$ 或者 $TIMx_CNT \leq TIMx_CCRi$ 。

根据 TIMx_CR1 寄存器中 CMS 位域的状态, 定时器能够产生边沿对齐的 PWM 信号或中央对齐的 PWM 信号。

13.6.7.1 PWM 边沿对齐模式

向上计数配置

当 TIMx_CR1 寄存器中的 DIR 位为低的时候执行向上计数。下面是一个 PWM 模式 1 的例子。当 $TIMx_CNT < TIMx_CCRi$ 时, PWM 参考信号 OCiREF 为高, 否则为低。如果 TIMx_CCRi 中的比较值大于自动重载值 (TIMx_ARR), 则 OCiREF 保持为 1。如果比较值为 0, 则 OCiREF 保持为 0。下图为 TIMx_ARR=8 时边沿对齐的 PWM 波形实例。

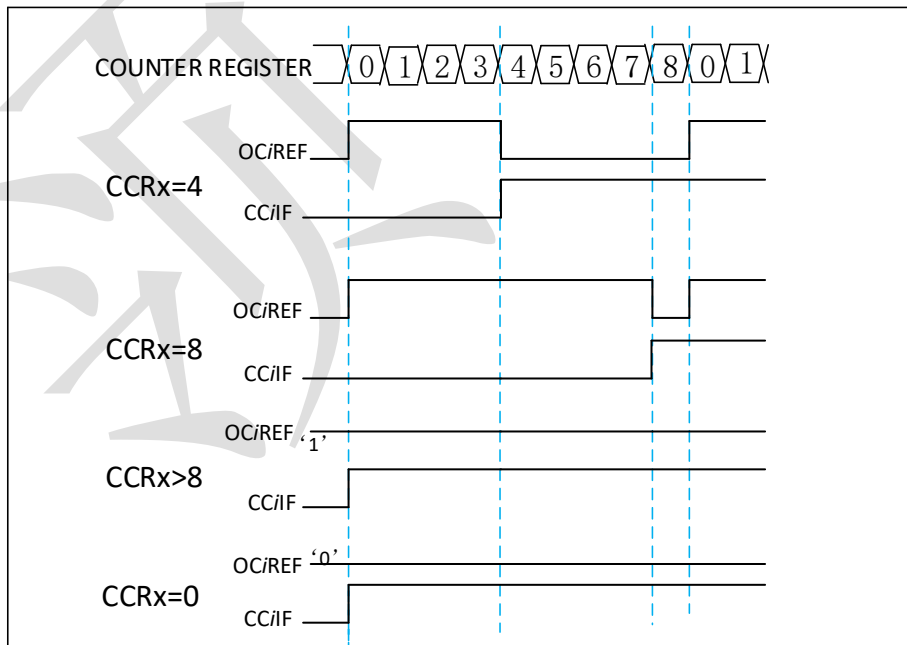


图 13-23 TIMx_ARR=8 时边沿对齐的 PWM 波形实例

向下计数的配置

当 TIMx_CR1 寄存器的 DIR 位为高时执行向下计数。在 PWM 模式 1 时, 当 $TIMx_CNT > TIMx_CCRi$ 时参考信号 OCiREF 为低, 否则为高。如果 TIM1_CCRi 中的比较值大于 TIMx_ARR 中的自动重载值, 则 OCiREF 保

持为 1。该模式下不能产生 0% 的 PWM 波形。

13.6.7.2 PWM 中央对齐模式

当 TIMx_CR1 寄存器中的 CMS 位不为 00 时为中央对齐模式（所有其它的配置对 OCiREF/OCi 信号都有相同的作用）。

根据不同的 CMS 位的设置，比较标志可以在计数器向上计数，向下计数，或向上和向下计数时被置 1。TIMx_CR1 寄存器中的计数方向位（DIR）由硬件更新，不要用软件修改它。

下图给出了一些中央对齐的 PWM 波形的例子：

- TIMx_ARR=8
- PWM 模式 1
- 标志位在以下三种情况下被置位（以箭头形式在图 13-24 中标出）

只有在计数器向下计数时（CMS=01）

只有在计数器向上计数时（CMS=10）

在计数器向上和向下计数时（CMS=11）

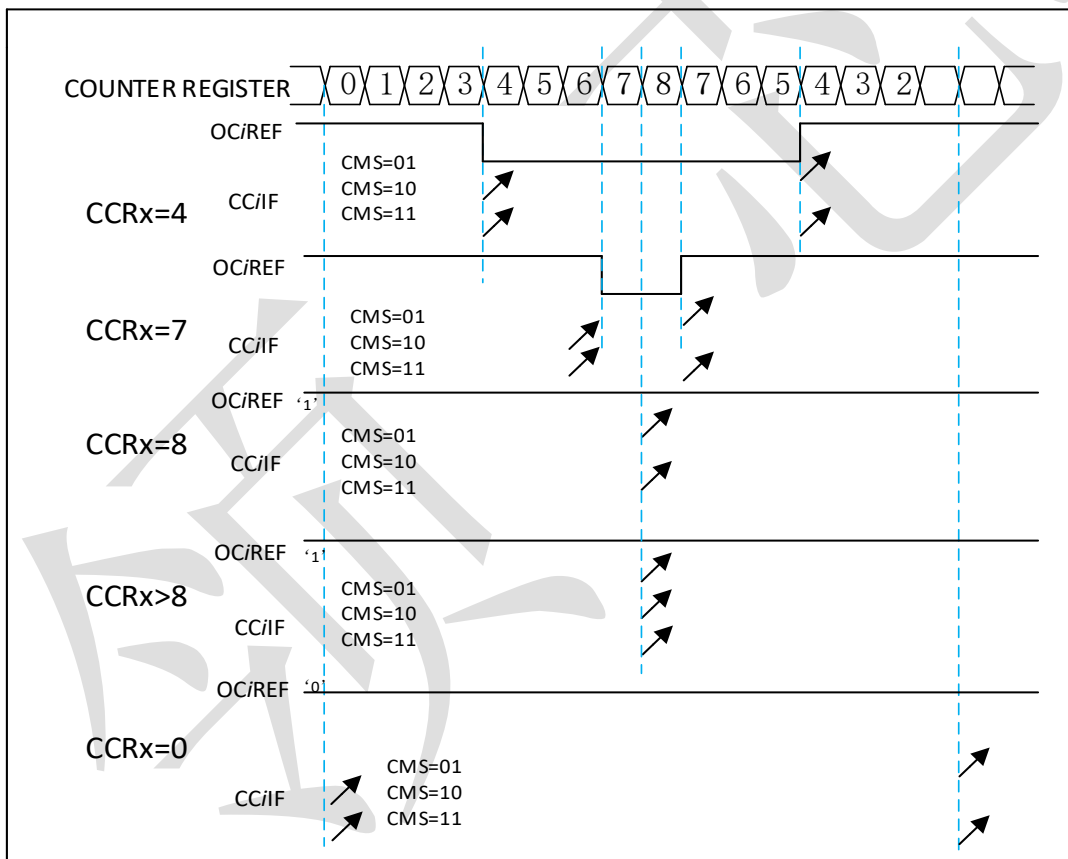


图 13-24 中央对齐模式 PWM（ARR=8）

13.6.8 单脉冲模式

单脉冲模式（OPM）是上述众多模式的一个特例。这种模式允许计数器响应一个激励，并在一个程序可控的延时之后产生一个脉宽可控的脉冲。

可以通过时钟/触发控制器启动计数器，在输出比较模式或者 PWM 模式下产生波形。设置 TIMx_CR1 寄存器的 OPM 位将选择单脉冲模式，此时计数器自动地在下一个更新事件 UEV 时停止。

仅当比较值与计数器的初始值不同时，才能产生一个脉冲。启动之前（当定时器正在等待触发），必须如下配置：

向上计数方式：计数器 $CNT < CCRi \leq ARR$

向下计数方式：计数器 $CNT > CCRi$

单脉冲模式的配置步骤：

如图 13-25 所示，如果用户想在 OC1 生成一个高电平脉冲，它通过 TI2 的输入触发，启动延迟为 t_{DELAY} ，维持时间为 t_{PULSE} 。首先要将 TI2FP2 设定为触发信号，此外单脉冲信号还要通过比较寄存器定义（考虑到时钟频率和预分频器）：

1. TI2FP2 对应到 TI2，配置 TIMx_CCMR1 寄存器中的 CC2S=01
2. TI2FP2 上升沿检测，配置 TIMx_CCER 寄存器中的 CC2P=0
3. 设置 TI2FP2 为从模式下的触发信号（TRGI），配置 TIMx_SMCR 寄存器中的 TS=110
4. 设置 TI2FP2 为计数器的启动信号，配置 TIMx_SMCR 寄存器中的 SMS=110（触发模式）
5. t_{DELAY} 通过 TIMx_CCR1 寄存器配置得到
6. t_{PULSE} 通过 TIMx_ARR - TIMx_CCR1 得到
7. 如果在计数器到达 TIMx_CCR1 时信号由 0 变为 1，到达 TIMx_ARR 时信号由 1 变为 0，则设置 PWM 模式 2，配置 TIMx_CCMR1 寄存器中的 OC1M=111

可选启用预装载功能，配置 TIMx_CCMR1 寄存器中的 OC1PE=1 和 TIMx_CR1 寄存器中的 ARPE=1。如果使用预装载功能，在 TIMx_ARR 和 TIMx_CCR1 配置后要配置 UG 位进行更新，并等待 TI2 的触发。在这个例子里，CC1P=0，TIMx_CR1 寄存器中的 DIR=0，CMS=00。

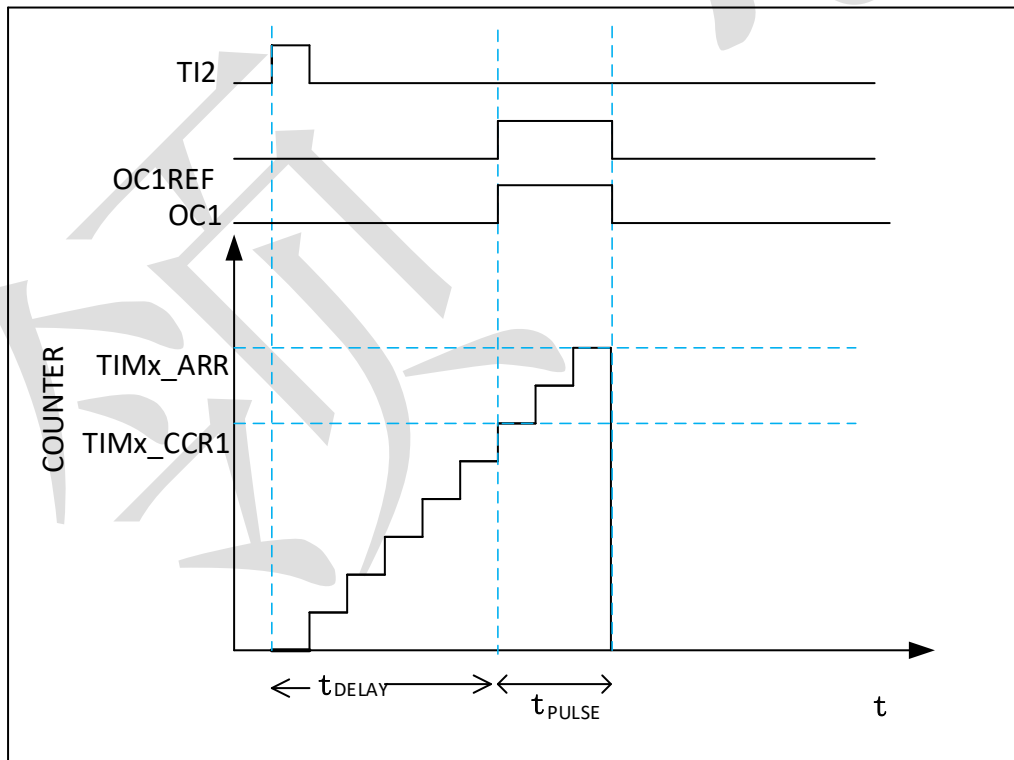


图 13-25 单脉冲模式

13.6.9 特殊情况：OCi 快速使能

在单脉冲模式下，对 Tii 输入脚的边沿检测会设置 CEN 位以启动计数器。然后计数器和比较值间的比较

操作产生了单脉冲的输出。但是这些操作需要一定的时钟周期，因此它限制了可得到的最小延时 t_{DELAY} 。

如果要以最小延时输出波形，可以设置 TIM1_CCMRi 寄存器中的 OCiFE 位；此时强制 OCiREF（和 OCx）直接响应激励而不再依赖比较的结果，输出的波形与比较匹配时的波形一样。OCiFE 只在通道配置为 PWM1 和 PWM2 模式的单脉冲模式时起作用。

13.6.10 互补输出和死区插入

TIM1 能够输出两路互补信号，并且能够管理输出的瞬时关断和接通。这段时间通常被称为死区，用户应该根据连接的输出器件和它们的特性（电平转换的延时、电源开关的延时等）来调整死区时间。

配置 TIMx_CCER 寄存器中的 CCiP 和 CCiNP 位，可以为每一个输出独立地选择极性（主输出 OCi 或互补输出 OCiN）。

互补信号 OCi 和 OCiN 通过下列控制位的组合进行控制：TIMx_CCER 寄存器的 CCiE 和 CCiNE 位，TIMx_BDTR 寄存器中的 MOE、OISi、OISiN、OSSi 和 OSSR 位。

特别是，在转换到 IDLE 状态时（MOE 下降到 0）死区控制被激活。

同时设置 CCiE 和 CCiNE 位将插入死区，如果存在刹车电路，则还要设置 MOE 位。每一个通道都有一个 8 位的死区发生器。如图 13-26 所示，参考信号 OCiREF 可以产生 2 路输出 OCi 和 OCiN。如果 OCi 和 OCiN 为高有效，CCiP=0，CCiNP=0，MOE=1，CCiE=1，CCiNE=1：

OCi 输出信号与参考信号相同，只是它的上升沿相对于参考信号的上升沿有一个延迟。

OCiN 输出信号与参考信号相反，只是它的上升沿相对于参考信号的下降沿有一个延迟。如果延迟大于当前有效的输出宽度（OCi 或者 OCiN），则不会产生相应的脉冲。

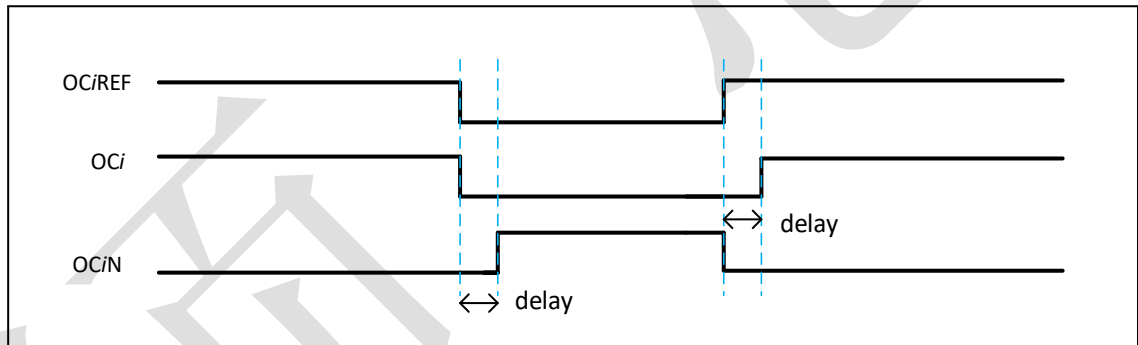


图 13-26 带死区插入的互补输出

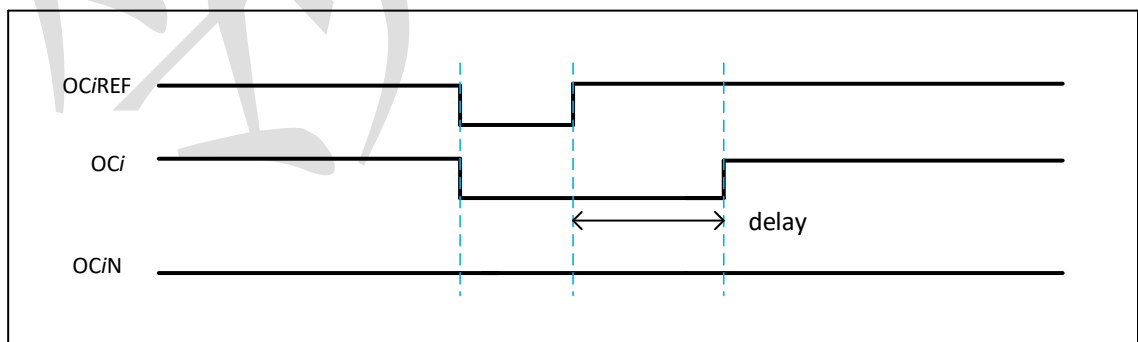


图 13-27 死区波形延迟大于负脉冲

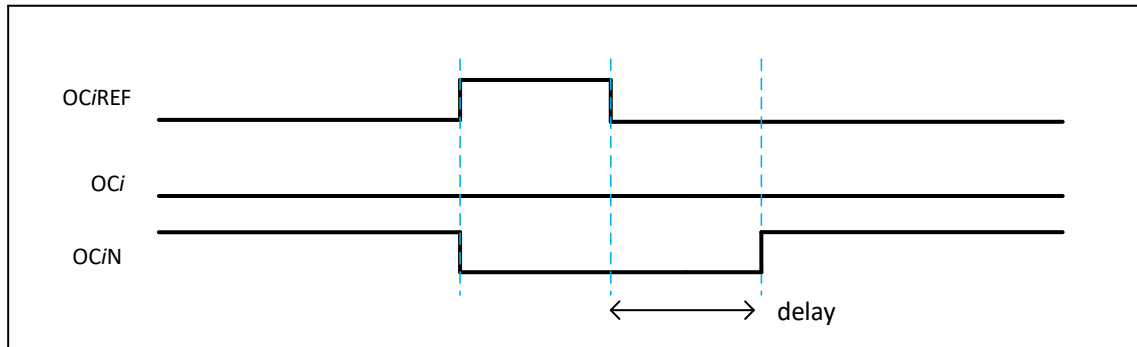


图 13-28 死区波形延迟大于正脉冲

13.6.11 重定向 OCiREF 到 OCi 或 OCiN

在输出模式（强置输出、输出比较或 PWM 模式）下，通过配置 TIMx_CCER 寄存器的 CCiE 和 CCiNE 位，OCiREF 可以被重定向到 OCi 或者 OCiN 的输出。

这个功能可以在互补输出的某一路未使能时，在某个输出上送出一个特殊的波形（例如 PWM 或者静态有效电平）。另一个作用是，让两个输出同时处于无效电平（均未使能），或同时处于有效电平（此时仍然是带死区的互补输出）。

注：当只使能 OCiN (CCiE=0, CCiNE=1) 时，它不会反相，而当 OCiREF 变高时立即有效。例如，如果 CCiNP=0，则 OCiN=OCiREF。另一方面，当 OCi 和 OCiN 都被使能时 (CCiE=CCiNE=1)，当 OCiREF 为高时 OCi 有效；而 OCiN 相反，当 OCiREF 低时 OCiN 变为有效。

13.6.12 刹车功能

刹车功能常用于马达控制中。当使用刹车功能时，依据相应的控制位（TIMx_BDTR 寄存器中的 MOE、OSSI 和 OSSR 位，TIMx_CR2 寄存器中的 OISi 和 OISiN 位），输出使能信号和电平都会被修改。TIM1 有 2 路刹车通道，其输入刹车源不同，分别具有各自的刹车控制位。对于 TIM1 的刹车通道 1，除了互联配置寄存器 4 (SysCtrl_EDU_CFG4) 中给出的刹车源之外，系统中会产生三个特殊的刹车源，分别为发生时钟安全问题、CPU 内核锁死和电压过低触发 LVD 低压检测标志。具体描述请见芯片控制寄存器 (ChipCtrl_CTRL) 寄存器 [31:29] 位。

定时器上有专门的刹车输入信号，在系统复位完成后，刹车电路被禁止，MOE 位为低。配置 TIMx_BDTR 寄存器中的 BKE 位可以使能刹车功能，刹车输入信号的极性可以通过配置其中的 BKP 位选择。BKE 和 BKP 可以被同时修改。

MOE 下降沿相对于时钟模块可以是异步的，因此在实际信号（作用在输出端）和同步控制位（在 TIMx_BDTR 寄存器中）之间设置了一个同步电路。这个电路会导致异步信号和同步信号之间产生延迟。特别的，如果当 MOE 从 0 变为 1，读出它之前必须先插入一个延时（空指令）才能读到正确的值。这是因为写入的是异步信号而读的是同步信号。

当刹车触发时：

1. MOE 位被异步清除，输出可通过 OSSI 位被置为不同状态
2. 当 MOE=0 时，输出可以由 TIMx_CR2 寄存器中的 OISi 和 OISiN 位驱动。如果 OSSI=0，定时器输出不使能，否则输出使能
3. 当使用互补输出时：
 - 输出首先会被置于复位状态（取决于极性配置），这是一个异步行为，所以不需要时钟
 - 如果时钟依旧存在，则会激活死区插入功能，在死区时间后，输出由配置好的 OISi 和 OISiN 位驱动。这里 OCi 和 OCiN 不能同时为有效电平

-
- 如果 $OSSI=0$ ，定时器输出不使能，否则输出使能或在 $CCiE/CCiNE$ 变高时拉高
4. 刹车状态标志 ($TIMx_SR$ 寄存器中的 BIF 位) 被置起。如果 $TIMx_DIER$ 寄存器中的 BIE 位使能，则会产生中断
 5. 如果 $TIMx_BDTR$ 寄存器中的 AOE 位被置位， MOE 位会自动在下一次更新事件 UEV 到来时被置位。否则 MOE 位会一直为低直到被手动写 1
 6. 当 AOE 被置位时，输出比较的 $TIMx_CCRi$ 寄存器的值可以通过相应的 $TIMx_CCTRi$ 修调寄存器来更新：如果 $TIMx_CCMRi$ 寄存器中的 $OCiTE=1$ ，当更新事件 UEV 发生时， $TIMx_CCRi$ 寄存器的影子寄存器被 $TIMx_CCTRi$ 更新；如果 $TIMx_CCMRi$ 寄存器中的 $OCiTE=1$ 且 $OCiTUE=1$ ，当更新事件 UEV 发生时， $TIMx_CCRi$ 寄存器的影子寄存器和预装载寄存器都被 $TIMx_CCTRi$ 更新

注：刹车输入是电平有效，因此当刹车输入保持有效时， MOE 不会被置位（无论自动还是手动）。同时， BIF 标志不能被清除。

有两种方式来产生刹车：

- 通过刹车输入并配合 $TIMx_BDTR$ 寄存器中的 BKE 位和 BKP 位
- 通过软件配置 $TIMx_EGR$ 寄存器中的 BG 位

除了刹车输入和输出管理，刹车电路中有写保护来保证应用的安全。它允许冻结几个参数的配置（死区时长、 $OCi/OCiN$ 极性和状态、 $OCiM$ 配置、刹车使能和极性等）。用户可以在 $TIMx_BDTR$ 寄存器中的 $LOCK$ 位中选择三个等级的保护机制，它们只能在 MCU 复位后写入一次。

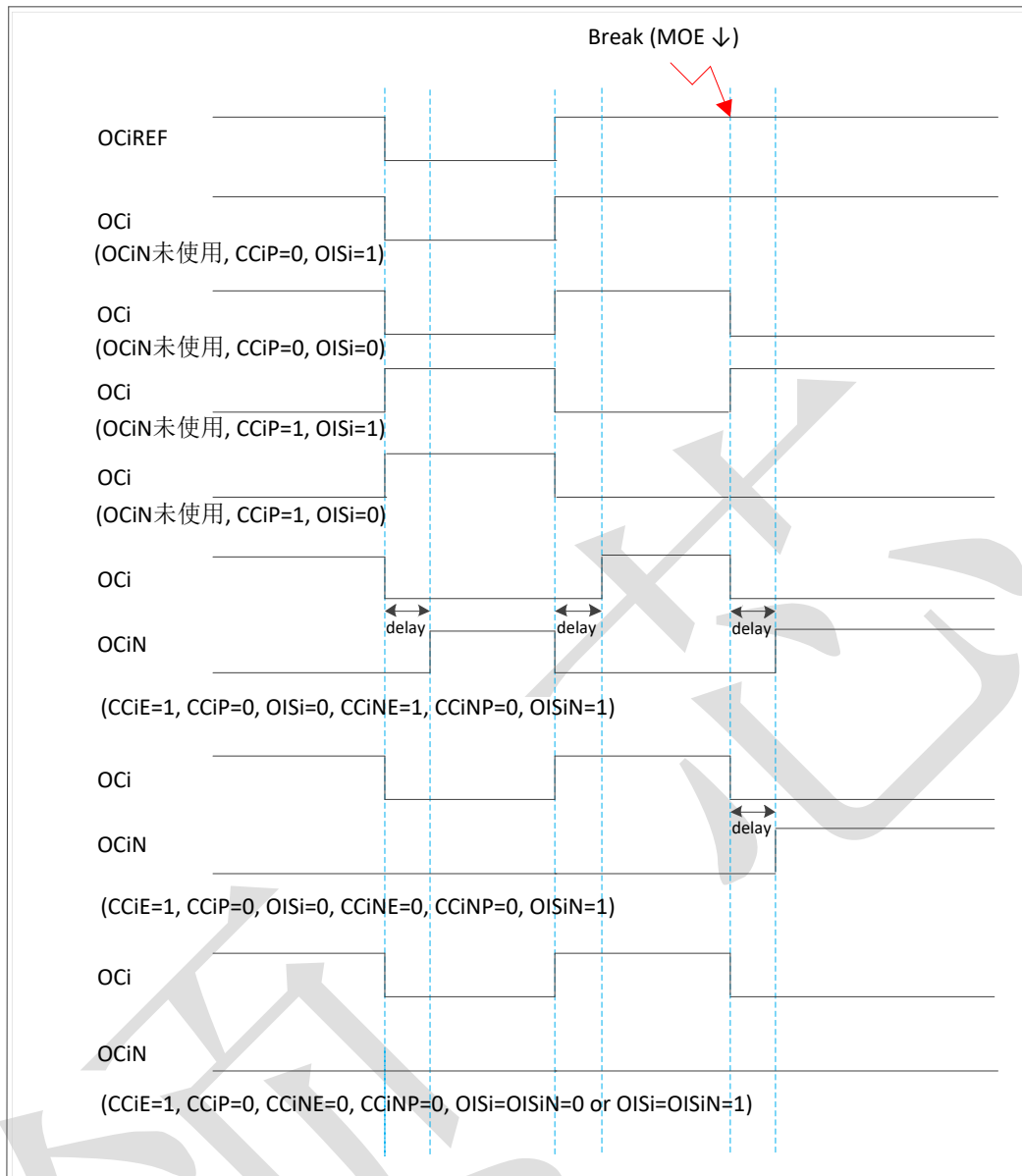


图 13-29 刹车对应的输出

13.6.13 通过外部事件清除 OCiREF

给定通道的 OCiREF 信号可以通过 OCREF_CLR_INT 清除（相应 TIMx_CCMR 寄存器中的使能位 OCiCE 置为 1）。OCiREF 保持低电平，直到下一个更新事件（UEV）发生。此功能只能在输出比较和 PWM 模式下使用，在强制输出模式下不起作用。

通过配置 TIMx_SMCR 寄存器中的 OCCS 位，可以在 OCREF_CLR 输入和 ETRF（滤波器后的 ETR）之间选择 OCREF_CLR_INT。如果选择了 ETRF，ETR 需要如下配置：

1. 外部触发预分频器要保持关闭：TIMx_SMCR 寄存器中 ETPS=00
2. 外部时钟源模式 2 关闭：TIMx_SMCR 寄存器中 ECE=0
3. 外部触发极性（ETP）和外部触发滤波器可以按照用户需求配置

OCREF_CLR 输入的来源为 ADC 看门狗输出、其他定时器的 TRGO 和各个 ACMP 的输出。OCREF_CLR 的极性可以通过 TIMx_SMCR 寄存器中的 OCCP 配置。

13.6.14 6 步 PWM 生成

当使用互补输出时，可以通过 TIMx_CR2 寄存器中的 CCPC 位控制 OCiM、CCiP、CCiNP、CCiE 和 CCiNE 位的预装载功能。这些预装载位在 COM 通信事件发生时才真正生效。用户可以提前为下一步的配置进行编程，并在 COM 发生时同时改变所有通道上的配置。COM 可以通过软件配置 TIMx_EGR 寄存器中的 COMG 位产生，或通过硬件在 TRGI 的上升沿产生，产生方式可以通过 TIMx_CR2 寄存器中的 CCUS 位控制。

当 COM 发生时，TIMx_SR 寄存器中的 COMIF 位置起，它可以产生中断（如果 TIMx_DIER 寄存器中的 COMIE 位被置位），或产生 DMA 请求（如果 TIMx_DIER 寄存器中的 COMDE 位被置位）。

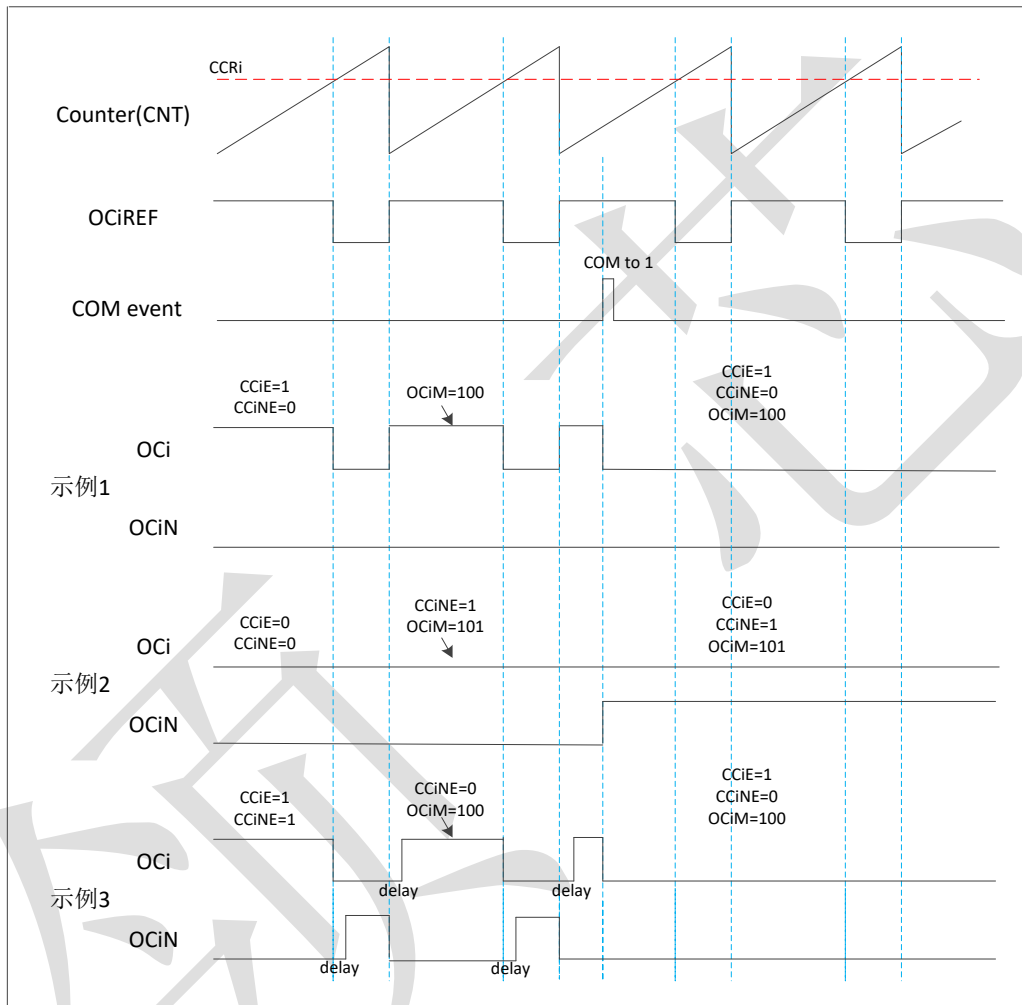


图 13-30 COM 触发的生成示例

13.6.15 编码器接口模式

选择编码器接口模式的方法：如果计数器只在 TI2 的边沿计数，则置 TIMx_SMCR 寄存器中的 SMS=001；如果只在 TI1 边沿计数，则置 SMS=010；如果计数器同时在 TI1 和 TI2 边沿计数，则置 SMS=011。通过设置 TIMx_CCER 寄存器中的 CC1P 和 CC2P 位，可以选择 TI1 和 TI2 极性；如果需要，还可以对输入进行滤波。

两个输入 TI1 和 TI2 被用来作为增量编码器的接口，参照表 13-2，假定计数器已经启动（TIMx_CR1 寄存器中的 CEN=1），则计数器由每次在 TI1FP1 或 TI2FP2 上的有效跳变驱动。TI1FP1 和 TI2FP2 是 TI1 和 TI2 在通过输入滤波器和极性控制后的信号；如果没有滤波和极性变化，则 TI1FP1=TI1，TI2FP2=TI2。根据两个输入信号的跳变顺序，产生了计数脉冲和方向信号。依据两个输入信号的跳变顺序，计数器向上或向下计数。

编码器接口模式相当于使用了一个带有方向选择的外部时钟。这意味着计数器只在 0 到 TIMx_ARR 寄存

器的自动装载值之间连续计数（根据方向决定是 0 到 ARR 计数，或是 ARR 到 0 计数）。所以在开始计数之前必须配置 TIMx_ARR 寄存器；同样，捕获器、预分频器和重复计数器特性等仍工作如常。编码器模式和外部时钟模式 2 不兼容，因此不能同时配置。在这个模式下，计数器依照增量编码器的速度和方向被自动修改，因此计数器的内容始终指示着编码器的位置。计数方向与相连的传感器的旋转方向对应。

假设 TI1 和 TI2 不同时变换，表 13-2 列出了所有可能的组合。这里相对信号是针对正在跳变的信号而言，TI1 在跳变时，相对信号的电平就是 TI2FP2 的电平；TI2 在跳变时，相对信号的电平就是 TI1FP1 的电平。

表 13-2 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

外部的增量编码器可以直接与 MCU 连接而不需要外部接口逻辑。然而，一般会使用比较器将编码器的差分输出转换为数字信号，这会大大增加抗噪声干扰能力。编码器输出的第三个信号表示机械零点，可以把它连接到一个外部中断输入并触发计数器复位。图 13-31 是一个编码器接口模式的示例，显示了计数信号的产生和方向控制。它还显示了当选择了双边沿时，输入抖动是如何被抑制的；抖动可能会在传感器的位置靠近一个转换点时产生。

在这个例子中，我们假定配置如下：

- CC1S=01（TIMx_CCMR1 寄存器，TI1FP1 映射到 TI1）
- CC2S=01（TIMx_CCMR1 寄存器，TI2FP2 映射到 TI2）
- CC1P=0，IC1F=0000（TIMx_CCER 寄存器，TI1FP1 不反相，TI1FP1=TI1）
- CC2P=0，IC2F=0000（TIMx_CCER 寄存器，TI2FP2 不反相，TI2FP2=TI2）
- SMS=011（TIMx_SMCR 寄存器，TI1 和 TI2 输入均在上升沿和下降沿有效）
- CEN=1（TIMx_CR1 寄存器，计数器使能）

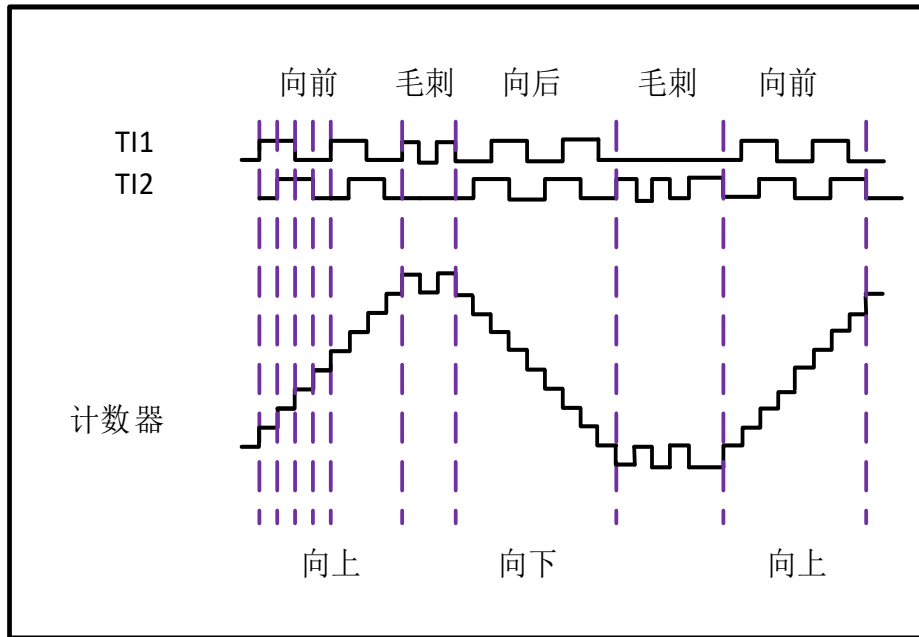


图 13-31 编码器接口模式示例

当其他配置保持一致，TI1FP1 极性反相时的示例：

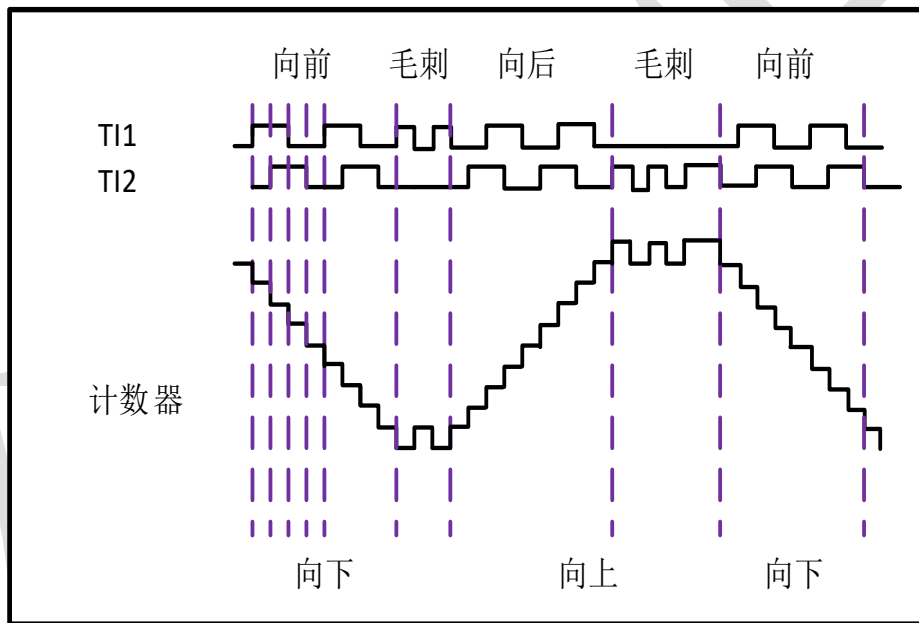


图 13-32 TI1FP1 反相的编码器接口模式示例

当定时器配置成编码器接口模式时，能够提供传感器当前的位置信息。如果使用另一个配置在捕获模式的定时器，我们可以测量两个编码器事件的间隔，并获得动态的信息（速度，加速度，减速度等）。表示机械零点的编码器输出可被用做此目的。根据编码器事件的间隔，我们按照固定的时间读出计数器：可以把计数器的值锁存到捕获寄存器中（捕获信号必须是周期性的，且可以由另一个定时器产生）；也可以通过一个由实时时钟产生的 DMA 请求来读取它的值。

13.6.16 定时器输入异或功能

TIMx_CR2 寄存器中的 TI1S 位，允许通道 1 的输入滤波器连接到一个异或门的输出端，异或门的 3 个输入端为 TIMx_CH1、TIMx_CH2 和 TIMx_CH3。异或输出能够被用于所有定时器的输入功能，如触发或输入捕获。13.6.17 章节给出了此特性用于连接霍尔传感器的例子。

13.6.17 与霍尔传感器的接口

使用 TIM1 产生 PWM 信号驱动马达时，可以用另一个通用 TIMx 定时器作为“接口定时器”来连接霍尔传感器，3 个“接口定时器”输入脚(CC1、CC2、CC3)通过一个异或门连接到 TI1 输入通道(通过设置 TIMx_CR2 寄存器中的 TI1S 位来选择)，“接口定时器”捕获这个信号。

“接口定时器”配置为从模式控制的复位模式，其输入是 TI1F_ED。每当 3 个输入之一变化时，计数器重新从 0 开始计数。这是一个由霍尔输入端的任何变化而触发的定时器。“接口定时器”上的捕获/比较通道 1 配置为捕获模式，捕获信号为 TRC。捕获值反映了两个输入变化间的时间延迟，给出了马达速度的信息。“接口定时器”可以用来在输出模式产生一个脉冲，这个脉冲可以(通过触发一个 COM 事件)用于改变高级定时器 TIM1 各个通道的属性，而 TIM1 产生 PWM 信号驱动马达。因此“接口定时器”通道必须编程为在一个指定的延时(输出比较或 PWM 模式)之后产生一个正脉冲，这个脉冲通过 TRGO 输出被送到高级控制定时器 TIM1。

举例：霍尔输入连接到 TIMx 定时器，要求每次任一霍尔输入上发生变化之后的一个指定的时刻，改变高级控制定时器 TIM1 的 PWM 配置。

- 置 TIMx_CR2 寄存器的 TI1S 位为 1，配置三个定时器输入逻辑或到 TI1 输入。
- 时基编程：置 TIMx_ARR 为其最大值(计数器必须通过 TI1 的变化清零)。设置预分频器得到一个最大的计数器周期，它长于传感器上的两次变化的时间间隔。
- 设置通道 1 为捕获模式(选中 TRC)：置 TIMx_CCMR1 寄存器中 CC1S=11，如果需要，还可以设置数字滤波。
- 设置通道 2 为 PWM2 模式，并具有要求的延时：置 TIMx_CCMR1 寄存器中的 OC2M=111 和 CC2S=00。
- 选择 OC2REF 作为 TRGO 上的触发输出：置 TIMx_CR2 寄存器中的 MMS=101。

在高级控制寄存器 TIM1 中，正确的 ITR 输入必须是触发输入，定时器被编程为产生 PWM 信号，捕获/比较控制信号为预装载的(TIM1_CR2 寄存器中 CCPC=1)，同时触发输入控制 COM 事件(TIM1_CR2 寄存器中 CCUS=1)。在一次 COM 事件后，写入下一次的 PWM 控制位(OCiM、CCiP、CCiNP、CCiE 和 CCiNE)，这可以在处理 OC2REF 上升沿的中断子程序里实现。

13.7 TIM1 定时器与外部触发的同步

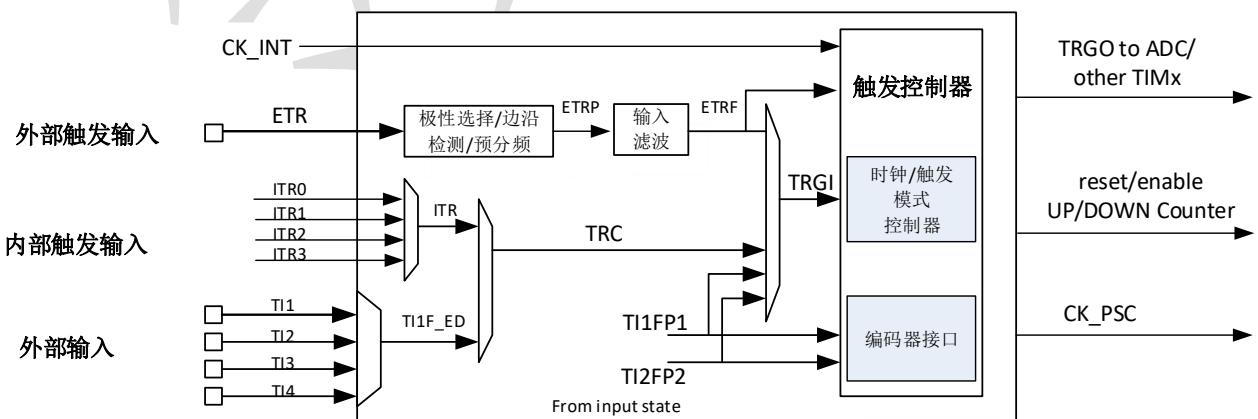


图 13-33 触发输入功能框图

TIM1 定时器能够在多种模式下和一个外部的触发同步：复位模式、门控模式、触发模式和触发复位模式。计数器允许四种触发输入：ETR（外部触发）；TI1 外部输入；TI2 外部输入；来自芯片内部其他模块。

TIM1 使用 4 种模式与外部的触发信号同步：标准触发模式、复位模式、门控模式和触发复位模式。在此基础上，增加了延迟触发和防多次触发功能，为 PWM 输出模式提供增强保障。

13.7.1 标准触发模式

计数器的使能依赖于选中的触发输入事件。在下面的例子中，TIM1 计数器在 TI2 输入的上升沿开始向上计数：

- 配置通道 2 用于检测 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，保持寄存器 CCMR1 中的 IC2F=0000）
- 触发操作中不使用捕获预分频器，不需要配置；寄存器 CCMR1 中的 CC2S=01，用于选择输入捕获源
- 配置 TIMx 的 CCER 寄存器的 CC2P=0，选择上升沿做为触发条件
- 配置 TIMx 的 SMCR 寄存器的 SMS=110，选择定时器为触发模式
- 配置 TIMx 的 SMCR 寄存器的 TS=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 TIMx_SR 寄存器的 TIF 位被置起。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。

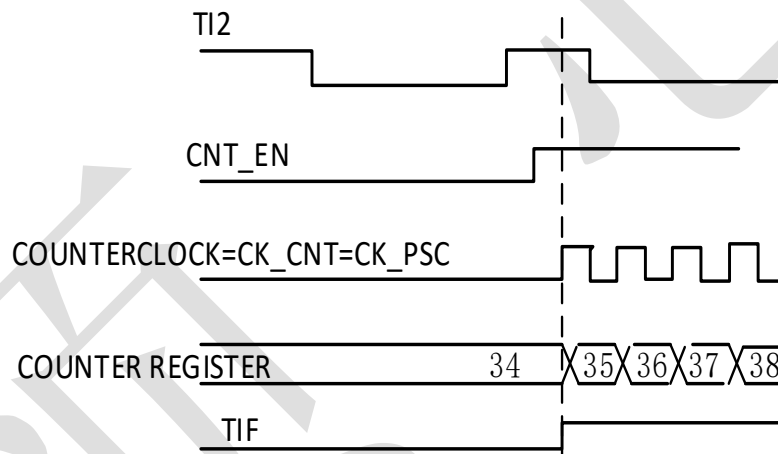


图 13-34 标准触发模式下的控制电路

13.7.2 复位模式

在发生一个触发输入事件时，计数器和它的预分频器能够重新被初始化；同时，如果 TIMx_CR1 寄存器的 URS 位为低，还产生一个更新事件 UEV；然后所有的预装载寄存器（ARR、CCR）都被更新。在以下的例子中，TI1 输入端的上升沿导致向上计数器被清零：

- 配置通道 1 用于检测 TI1 的上升沿；配置输入滤波器的带宽（在本例中，不需要任何滤波器，因此保持 IC1F=0000）
- 触发操作中不使用捕获预分频器，所以不需要配置；寄存器 CCMR1 中的 CC1S=01，用于选择输入捕获源
- 配置 TIMx 的 CCER 寄存器的 CC1P=0 来选择极性（只检测上升沿）
- 配置 TIMx 的 SMCR 寄存器的 SMS=100，选择定时器为复位模式
- 配置 TIMx 的 SMCR 寄存器的 TS=101，选择 TI1 作为输入源

- 配置 TIMx_CR1 寄存器的 CEN=1，启动计数器
- 计数器开始依据内部时钟计数，然后正常计数直到 TI1 出现一个上升沿；此时，计数器被清零然后从 0 重新开始计数。同时，TIMx_SR 寄存器的 TIF 位被置起，如果使能了中断（TIMx_DIER 寄存器的 TIE 位），则产生一个中断请求

下图显示当自动重载寄存器 TIMx_ARR=0x36 时的动作。在 TI1 上升沿和计数器的实际复位之间的延时取决于 TI1 输入端的重同步电路。

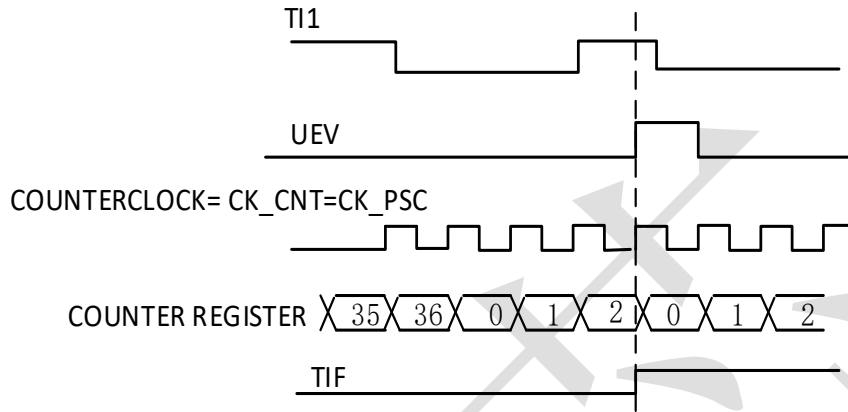


图 13-35 复位模式下的控制电路

13.7.3 门控模式

计数器由选中的输入端信号的电平使能。在以下的例子中，计数器只在 TI1 为低时向上计数：

- 配置通道 1 用于检测 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 IC1F=0000）
- 触发操作中不使用捕获预分频器，所以不需要配置；寄存器 CCMR1 中的 CC1S=01，用于选择输入捕获源
- 配置 TIMx_CCER 寄存器的 CC1P=1 来确定极性（只检测低电平）
- 配置 TIMx_SMCR 寄存器的 SMS=101，选择定时器为门控模式
- 配置 TIMx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源
- 配置 TIMx_CR1 寄存器的 CEN=1，启动计数器（门控模式下，如果 CEN=0，则计数器不能启动，无论触发输入电平如何）

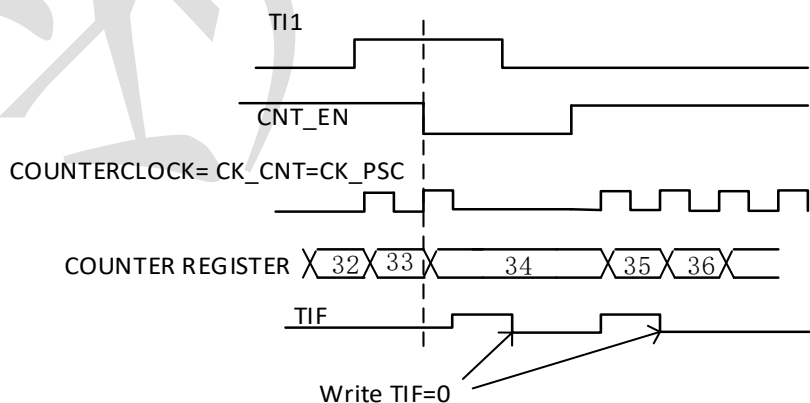


图 13-36 门控模式下的控制电路

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，

TIMx_SR 寄存器的 TIF 位都会被置起。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。

13.7.4 触发复位模式

计数器的使能依赖于选中的触发输入事件，在发生一个触发输入事件时，计数器和它的预分频器能够重新被初始化；同时，如果 TIMx_CR1 寄存器的 URS 位为低，还产生一个更新事件 UEV；然后所有的预装载寄存器（ARR、CCR）都被更新。这种模式可视为是标准触发模式和复位模式的结合，其配置与标准触发模式基本一致，不同之处在于 TIMx 的 SMCR 寄存器的 SMS=1000。

13.7.5 外部时钟模式 2 及触发模式

外部时钟模式 2 可以与另一个输入信号的触发模式一起使用。这时，ETR 信号被用作外部时钟的输入，另一个输入信号可用作触发模式（支持标准触发模式，复位模式、门控模式和触发复位模式）。请注意不能把 ETR 配置成 TRGI（通过 TIMx_SMCR 寄存器的 TS 位）。

在下面的例子中，一旦在 TI1 上出现一个上升沿，计数器即在 ETR 的每一个上升沿向上计数一次，通过 TIMx_SMCR 寄存器配置外部触发输入电路。

- 首先配置 ETR：TIMx_SMCR 寄存器中配置 ETF=0000 禁止滤波器，配置 ETPS=00 禁止预分频，配置 ETP=0 监测 ETR 信号的上升沿，配置 ECE=1 使能外部时钟模式 2
- 配置通道 1 用于检测 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 IC1F=0000）
- 触发操作中不使用捕获预分频器，所以不需要配置；寄存器 CCMR1 中的 CC1S=01，用于选择输入捕获源
- 配置 TIMx 的 CCER 寄存器的 CC1P=0，选择上升沿做为触发条件
- 配置 TIMx 的 SMCR 寄存器的 SMS=110，选择定时器为触发模式
- 配置 TIMx 的 SMCR 寄存器的 TS=101，选择 TI1 作为输入源

当 TI1 上出现一个上升沿时，同时 TIMx_SR 寄存器的 TIF 位被置起，计数器开始在 ETR 的上升沿计数。TI1 信号的上升沿和计数器实际时钟之间的延时取决于 TI1 输入端的重同步电路。ETR 信号的上升沿和计数器实际时钟之间的延时取决于 ETRP 输入端的重同步电路。

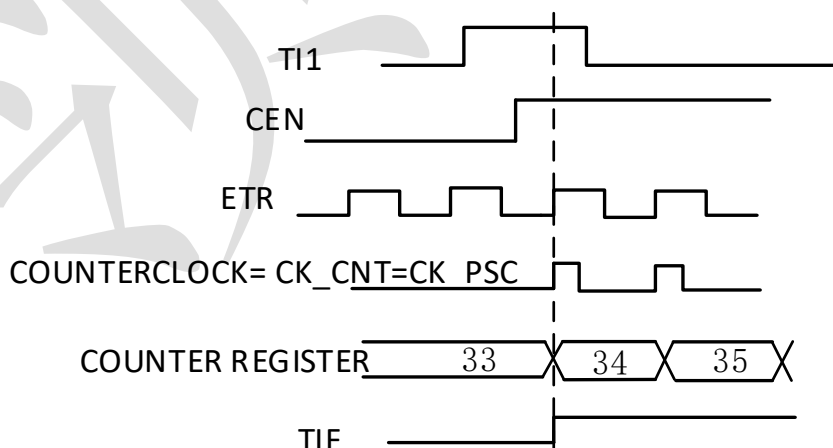


图 13-37 外部时钟模式 2+触发模式下的控制电路

13.8 TIM1 与其他定时器的联接

在芯片中，各个定时器在内部互相联接，用于定时器的同步或连接。当某个定时器配置成主模式时，可

以输出触发信号（TRGO）到那些配置为从模式的定时器来完成复位、启动和停止的操作，或者作为那些定时器的驱动时钟。具体可参考图 13-33。

高级定时器 TIM1 的输入触发可以来自芯片内部的其他定时器、比较器 ACMP0/1/2 输出和 ADC；TIM1 的输出 TRGO 可触发芯片内部的 ADC、DAC 和其他定时器；TIM1 的输出 OC_i 可触发芯片内部的比较器 ACMP0/1/2 和 ADC；TIM1 的特殊输出 TIM1_oc1_trig 可触发其他定时器，它与 CEN 位相关，具体详见 TIM1_CR2 寄存器 TrgIS 位说明。

表 13-3 TIM1 触发与级联表

信号名称	信号描述	来源	选择寄存器
内部触发 ITR			
ITR0	内部触发 0	tim15_trgo	-
ITR1	内部触发 1	tim16_oc1_trig	-
ITR2	内部触发 2	tim3_trgo	-
ITR3	内部触发 3	tim17_oc1_trig	-
外部 Timer 输入 TI			
TI1	外部 Timer 输入 1	tim1_gpio_ti1/adc_awd/ cp0_out/cp1_out/cp2_out/ tim15_oc1_trig / tim3_oc1_trig / tim14_oc1_trig	SysCtrl_EDU_CFG1
TI2	外部 Timer 输入 2	tim1_gpio_ti2/adc_awd/ cp0_out/cp1_out/cp2_out/ tim15_oc1_trig / tim3_oc1_trig / tim14_oc1_trig	
TI3	外部 Timer 输入 3	tim1_gpio_ti3/adc_awd/ cp0_out/cp1_out/cp2_out/ tim15_oc1_trig / tim3_oc1_trig / tim14_oc1_trig	
TI4	外部 Timer 输入 4	tim1_gpio_ti4/adc_awd/ cp0_out/cp1_out/cp2_out/ tim15_oc1_trig / tim3_oc1_trig / tim14_oc1_trig	
外部触发输入 ETR			
ETR	外部触发	tim1_gpio_etr/adc_awd/ cp0_out/cp1_out/cp2_out/ tim15_oc1_trig / tim3_oc1_trig / tim14_oc1_trig	SysCtrl_EDU_CFG1

注：请参考 SysCtrl_EDU_CFG1 寄存器定义。

13.9 TIM1 中断

TIM1 有 8 个中断请求源，分别映射到 2 个中断矢量上：

- 刹车中断
- 更新事件中断
- 触发中断
- COM 事件中断
- 输入捕获/输出比较 1 中断
- 输入捕获/输出比较 2 中断
- 输入捕获/输出比较 3 中断
- 输入捕获/输出比较 4 中断

为了使用中断特性，对每个被使用的中断通道，设置 TIM_x_DIER 寄存器中相应的中断使能位：BIE、TIE、

COMIE、CCIE 和 UIE 位。

通过设置 TIMx_EGR 寄存器中的相应位，也可以用软件产生上述各个中断源。

13.10 TIM1 寄存器描述

表 13-4 TIM1 相关寄存器表

名称	说明	读写权限	复位值	字节地址
CR1	控制寄存器 1	R/W	0x0000_0000	0x4001_0000
CR2	控制寄存器 2	R/W	0x0000_0000	0x4001_0004
SMCR	从模式控制寄存器	R/W	0x0000_0000	0x4001_0008
DIER	DMA 和中断控制寄存器	R/W	0x0000_0000	0x4001_000C
SR	状态寄存器	R/W	0x0000_0000	0x4001_0010
EGR	事件产生寄存器	R/W	0x0000_0000	0x4001_0014
CCMR1	捕获/比较模式寄存器 1	R/W	0x0000_0000	0x4001_0018
CCMR2	捕获/比较模式寄存器 2	R/W	0x0000_0000	0x4001_001C
CCER	捕获/比较使能寄存器	R/W	0x0000_0000	0x4001_0020
CNT	计数寄存器	R/W	0x0000_0000	0x4001_0024
PSC	预分频寄存器	R/W	0x0000_0000	0x4001_0028
ARR	自动重装载寄存器	R/W	0x0000_0000	0x4001_002C
RCR	重复计数寄存器	R/W	0x0000_0000	0x4001_0030
CCR1	捕获/比较寄存器 1	R/W	0x0000_0000	0x4001_0034
CCR2	捕获/比较寄存器 2	R/W	0x0000_0000	0x4001_0038
CCR3	捕获/比较寄存器 3	R/W	0x0000_0000	0x4001_003C
CCR4	捕获/比较寄存器 4	R/W	0x0000_0000	0x4001_0040
BDTR	刹车死区控制寄存器	R/W	0x0000_0000	0x4001_0044
DCR	DMA 控制寄存器	R/W	0x0000_0000	0x4001_0048
DMAR	DMA 传输寄存器	R/W	0x0000_0000	0x4001_004C
CCTR1	比较修调寄存器 1	R/W	0x0000_0000	0x4001_0064
CCTR2	比较修调寄存器 2	R/W	0x0000_0000	0x4001_0068
CCTR3	比较修调寄存器 3	R/W	0x0000_0000	0x4001_006C
CCTR4	比较修调寄存器 4	R/W	0x0000_0000	0x4001_0070
DLAMT	触发延迟和防触发丢失寄存器	R/W	0x0000_0000	0x4001_007C
SysCtrl_EDU_CFG1	互联配置寄存器 1	R/W	0x0000_0000	0x4800_7024
SysCtrl_EDU_CFG4	互联配置寄存器 4	R/W	0x0000_0000	0x4800_7030

注：x 表示不确定；-表示未实现；u 表示不受复位影响；c 表示该位复位值因复位源而异；R 只读；W 只写；R-0 表示只读且读 0；R/W 表示可读写（以后章节同上述）。

13.10.1 CR1 控制寄存器 1 (TIM1_CR1)

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
Res.							AS	OCF	CENCE	TI4E	TI3E	TI2E	TI1E	ETRE	FTE
							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
Res.						CKD[1:0]	ARPE	CMS[1:0]	DIR	OPM	URS	UDIS	CEN		
						rw	rw	rw	rw	rw	rw	rw	rw		

Bits	31:25	保留，必须保持复位值
Bit	24	AS: 影子寄存器读使能
		该位指定当 MCU 读取 TIM1_PSC/TIM1_ARR/TIM1_CCRi 寄存器的时候，是读取影子寄存器还是预装载寄存器
		0: 选择预装载寄存器的值（默认）
		1: 选择影子寄存器的值
Bit	23	OCF: 输出异步抗干扰滤波
		0: 输出滤波禁止（默认）
		1: 输出滤波使能
Bit	22	CENCE: CEN 不使能的时候允许清除 OCREF
		0: CEN 不使能时，OCREF clear 禁止（默认）
		1: CEN 不使能时，OCREF clear 使能
Bit	21	TI4E: TI4 输入检测/滤波使能
		0: TI4 输入检测/滤波禁止（默认）
		1: TI4 输入检测/滤波使能
Bit	20	TI3E: TI3 输入检测/滤波使能
		0: TI3 输入检测/滤波禁止（默认）
		1: TI3 输入检测/滤波使能
Bit	19	TI2E: TI2 输入检测/滤波使能
		0: TI2 输入检测/滤波禁止（默认）
		1: TI2 输入检测/滤波使能
Bit	18	TI1E: TI1 输入检测/滤波使能
		0: TI1 输入检测/滤波禁止（默认）
		1: TI1 输入检测/滤波使能
Bit	17	ETRE: 外部触发输入检测/滤波使能
		0: ETR 输入检测/滤波禁止（默认）
		1: ETR 输入检测/滤波使能
Bit	16	FTE: 数字滤波器（ETR、Tli）的 DTS 时钟使能
		0: 数字滤波器的 DTS 时钟禁止（默认）
		1: 数字滤波器的 DTS 时钟使能
Bits	15:10	保留，必须保持复位值
Bits	9:8	CKD[1:0]: CK_INT 时钟和死区/采样时钟（CK_DTS）的分频系数，DTS 时钟供给死区发生器和数字滤波器（ETR、Tli）使用
		00: $t_{DTS} = t_{CK_INT}$ （默认）
		01: $t_{DTS} = 2 * t_{CK_INT}$
		10: $t_{DTS} = 4 * t_{CK_INT}$
		11: 保留
Bit	7	ARPE: 自动预装载允许位

	0: TIM1_ARR 寄存器可以被直接写入 (默认)
	1: TIM1_ARR 寄存器通过预装载寄存器更新
Bits 6:5	CMS[1:0]: 选择中央对齐模式
	00: 边沿对齐模式, 计数器依据方向位 (DIR) 向上或向下计数。(默认)
	01: 中央对齐模式 1, 计数器交替地向上和向下计数。只有在计数器向下计数时, 输出比较中断标志位 (TIM1_CCMRi 寄存器中 CCIS=00 条件下) 才会被置 1。
	10: 中央对齐模式 2, 计数器交替地向上和向下计数。只有在计数器向上计数时, 输出比较中断标志位 (TIM1_CCMRi 寄存器中 CCIS=00 条件下) 才会被置 1。
	11: 中央对齐模式 3, 计数器交替地向上和向下计数。在计数器向上或向下计数时, 输出比较中断标志位 (TIM1_CCMRi 寄存器中 CCIS=00 条件下) 会被置 1。
	注 1: 在计数器开启时 (CEN=1), 不允许从边沿对齐模式切换到中央对齐模式。
	注 2: 在中央对齐模式下, 编码器模式 (TIM1_SMCR 寄存器中的 SMS=001、010 或 011) 必须被禁止。
Bit 4	DIR: 计数器方向
	0: 计数器向上计数 (默认)
	1: 计数器向下计数
	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。
Bit 3	OPM: 单脉冲模式
	0: 在发生更新事件时, 计数器不停止 (默认)
	1: 在发生下一次更新事件时, 计数器停止 (清除 CEN 位)
	注: TIM1 的 OPM 模式只有在至少有一个通道处于输出模式时才会生效。
Bit 2	URS: 更新请求源
	0: 如果 UDIS 允许产生更新事件, 则下述任一事件产生一个更新中断 (默认):
	- 计数器上溢/下溢
	- 软件设置 UG 位
	- 时钟/触发控制器产生的硬件复位
	1: 如果 UDIS 允许产生更新事件, 则只有当计数器上溢/下溢时才产生更新中断
Bit 1	UDIS: 禁止更新
	0: 一旦下列事件发生, 产生更新事件 (默认):
	- 计数器溢出/下溢
	- 软件设置 UG 位
	- 时钟/触发控制器产生的硬件复位
	1: 不产生更新事件, 影子寄存器 (ARR、PSC、CCR) 保持它们的值。如果 UG 位被配置或时钟/触发控制器发出了一个硬件复位, 则计数器和预分频器被重新初始化
Bit 0	CEN: 计数器使能位
	0: 计数器禁止 (默认)
	1: 计数器使能
	注: 在软件配置了 CEN 后, 外部时钟模式、复位模式、门控模式和编码器模式才能工作。触发模式可以自动通过硬件启动。在门控模式下, 该位读回值表示门控状态。

13.10.2 CR2 控制寄存器 2 (TIM1_CR2)

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
Res.															TrigS
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Res.	CCPC
	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw		rw

Bits	31:17	保留，必须保持复位值
Bit	16	TrigS: TIMx_oc1_trig 触发源选择
		0: 触发源选择为 OC1&CEN (默认)
		1: 触发源选择为 CEN
Bit	15	保留，必须保持复位值
Bit	14	OIS4: 输出空闲状态 4 (OC4 输出)，参考 OIS1
Bit	13	OIS3N: 输出空闲状态 3 (OC3N 输出)，参考 OIS1N
Bit	12	OIS3: 输出空闲状态 3 (OC3 输出)，参考 OIS1
Bit	11	OIS2N: 输出空闲状态 2 (OC2N 输出)，参考 OIS1N
Bit	10	OIS2: 输出空闲状态 2 (OC2 输出)，参考 OIS1
Bit	9	OIS1N: 输出空闲状态 1 (OC1N 输出)
		0: OC1N 的空闲电平为 0 (默认)
		1: OC1N 的空闲电平为 1
		<i>注: 已经设置了 LOCK (TIM1_BDTR 寄存器) 级别 1、2 或 3 后，该位不能被修改。</i>
Bit	8	OIS1: 输出空闲状态 1 (OC1 输出)
		0: OC1 的空闲电平为 0 (默认)
		1: OC1 的空闲电平为 1
		<i>注: 已经设置了 LOCK (TIM1_BDTR 寄存器) 级别 1、2 或 3 后，该位不能被修改。</i>
Bit	7	TI1S: TI1 输入选择
		0: CC1 输入管脚连到 TI1 (默认)
		1: CC1、CC2 和 CC3 管脚经异或后连到 TI1
Bits	6:4	MMS[2:0]: 主模式选择
		用于选择在主模式下送到其他模块的同步信号 (TRGO)
		000: 复位 - 软件设置 UG 位或时钟/触发控制器产生的硬件复位被用作触发输出 (TRGO)。如果触发输入 (时钟/触发控制器配置为复位模式) 产生复位，则 TRGO 上的信号会延迟到与实际的复位同步。(默认)
		001: 使能 - 计数器使能信号被用作触发输出 (TRGO)。其用于同时启动多个定时器，或在一段时间内控制从定时器和其他模块。计数器使能信号是通过 CEN 控制位或门控模式下的触发输入信号产生。除非选择了主/从模式 (见 TIM1_SMCR 寄存器中 MSM 位的描述)，否则当计数器使能信号受控于触发输入时，TRGO 上会有一个延迟。
		010: 更新 - 更新事件被用作触发输出 (TRGO)。在这种模式下，一个主定时器可用作为一个从定时器的预分频器。
		011: 比较脉冲 - 一旦发生一次捕获或一次比较成功，当 CC1IF 标志被置 1 时 (即使它已经为高)，触发输出送出一个正脉冲 (TRGO)。

	100: 比较 - OC1REF 信号被用作触发输出 (TRGO)。
	101: 比较 - OC2REF 信号被用作触发输出 (TRGO)。
	110: 比较 - OC3REF 信号被用作触发输出 (TRGO)。
	111: 比较 - OC4REF 信号被用作触发输出 (TRGO)。
Bit 3	CCDS: 捕获/比较 DMA 选择
	0: 当 CCx 事件发生的时候发送 CCx DMA 请求 (默认)
	1: 当更新事件发生的时候发送 CCx DMA 请求
Bit 2	CCUS: 捕获/比较控制位的更新控制选择
	0: 当捕获/比较的控制位为预装载时 (CCPC=1), 只有在 COMG 位置 1 的时候这些控制位才被更新 (默认)
	1: 当捕获/比较的控制位为预装载时 (CCPC=1), 只有在 COMG 位置 1 或 TRGI 上升沿的时候这些控制位才被更新
	<i>注: 该位只对拥有互补输出的通道有效。</i>
Bit 1	保留, 必须保持复位值
Bit 0	CCPC: 捕获/比较预装载控制位
	0: CCiE、CCiNE、CCiP、CCiNP 位 (TIM1_CCER 寄存器) 和 OCiM 位 (TIM1_CCMRi 寄存器) 不是预装载的 (默认)
	1: CCiE、CCiNE、CCiP、CCiNP 和 OCiM 位是预装载的; 设置该位后, 它们只在设置了 COM 事件 (COMG 位置 1 或 TRGI 上升沿, 由 CCUS 位控制) 发生后被更新
	<i>注: 该位只对拥有互补输出的通道有效。</i>

13.10.3 SMCR 从模式控制寄存器 (TIM1_SMCR)

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	
Res.														OCCP	SMS[3]	
															rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]			MSM	TS[2:0]		OCCS	SMS[2:0]				
rw	rw	rw			rw			rw		rw		rw			rw

Bits 31:18	保留, 必须保持复位值
Bit 17	OCCP: OCREF_CLR 输入极性选择
	0: OCREF_CLR 输入高有效 (默认)
	1: OCREF_CLR 输入低有效
Bit 16	SMS[3]: 从模式选择的 bit 3
	参考 SMS[2:0] 的描述
Bit 15	ETP: 外部触发极性
	0: ETR 不反相, 即高电平或上升沿有效 (默认)
	1: ETR 反相, 即低电平或下降沿有效
Bit 14	ECE: 外部时钟使能, 用于使能外部时钟模式 2

	0: 外部时钟模式 2 禁止 (默认)
	1: 外部时钟模式 2 使能, 计数器的时钟为 ETRF 的有效边沿
	注 1: ECE 位置 1 的效果与选择把 TRGI 连接到 ETRF 的外部时钟模式 1 相同 (TIM1_SMCR 寄存器中, SMS=111, TS=111)。
	注 2: 外部时钟模式 2 可与下列模式同时使用: 标准触发模式、复位模式、门控模式和触发复位模式。但是, 此时 TRGI 不能与 ETRF 相连 (TIM1_SMCR 寄存器中, TS 不能为 111)。
	注 3: 如果外部时钟模式 1 与外部时钟模式 2 同时使能, 外部时钟输入为 ETRF。
Bits 13:12	ETPS: 外部触发预分频器
	外部触发信号 ETRP 的频率最大不能超过定时器时钟 CK_INT 频率的 1/4。可用预分频器来降低 ETRP 的频率, 当 ETRP 的频率很高时非常有用。
	00: 预分频器关闭 (默认)
	01: ETRP 的频率/2
	10: ETRP 的频率/4
	11: ETRP 的频率/8
Bits 11:8	ETF[3:0]: 外部触发滤波器选择
	该位域定义了 ETRP 的采样频率及数字滤波器的长度。数字滤波器由一个事件计数器组成, 只有发生了 N 个连续事件后输出的跳变才被认为有效。这里的 DTS 时钟是 CK_INT 经过 CR1 寄存器中 CKD 配置分频的。
	0000: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}$, 无滤波器 (默认)
	0001: 采样频率 $f_{\text{SAMPLING}} = f_{\text{CK_INT}}$, N=2
	0010: 采样频率 $f_{\text{SAMPLING}} = f_{\text{CK_INT}}$, N=4
	0011: 采样频率 $f_{\text{SAMPLING}} = f_{\text{CK_INT}}$, N=8
	0100: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/2$, N=6
	0101: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/2$, N=8
	0110: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$, N=6
	0111: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$, N=8
	1000: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/8$, N=6
	1001: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/8$, N=8
	1010: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$, N=5
	1011: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$, N=6
	1100: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$, N=8
	1101: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$, N=5
	1110: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$, N=6
	1111: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$, N=8
	注: 请注意采样频率中 $f_{\text{CK_INT}}$ 和 f_{DTS} 的差别。
Bit 7	MSM: 主/从模式
	0: 无作用 (默认)
	1: 触发输入 (TRGI) 上的事件被延迟了, 以允许 TIM1 与它的从定时器间通过 TRGO 完美同步
Bits 6:4	TS[2:0]: 选择同步计数器的触发输入
	000: 内部触发 ITR0, 连接 TIM15_TRGO (默认)
	001: 内部触发 ITR1, 连接 TIM16_oc1_trig
	010: 内部触发 ITR2, 连接 TIM3_TRGO

	011: 内部触发 ITR3, 连接 TIM17_oc1_trig
	100: TI1 的边沿检测器 (TI1F_ED)
	101: 滤波后的定时器输入 1 (TI1FP1)
	110: 滤波后的定时器输入 2 (TI2FP2)
	111: 外部触发输入 (ETRF)
	<i>注: 这些位只能在未用到 (如 SMS=000) 时被改变, 以避免在改变时产生错误的边沿检测。</i>
Bit 3	OCCS: 选择 OCREF 清除源
	0: OCREF_CLR_INT 连接到 OCREF_CLR 输入 (默认)
	1: OCREF_CLR_INT 连接到 ETRF 输入
Bits 2:0	SMS[2:0]: 从模式选择
	当选择外部信号时, 触发信号 (TRGI) 的有效边沿与外部输入的极性相关。
	000: 从模式禁止 - 如果 CEN=1, 则预分频器直接由内部时钟驱动。(默认)
	001: 编码器模式 1 - 根据 TI1FP1 的电平, 计数器在 TI2FP2 的边沿向上/下计数。
	010: 编码器模式 2 - 根据 TI2FP2 的电平, 计数器在 TI1FP1 的边沿向上/下计数。
	011: 编码器模式 3 - 根据另一个输入的电平, 计数器在 TI1FP1 和 TI2FP2 的边沿向上/下计数。
	100: 复位模式 - 在选中的触发输入 (TRGI) 的上升沿时重新初始化计数器, 并且产生一个更新寄存器的信号。
	101: 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。
	110: 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。
	111: 外部时钟模式 1 - 选中的触发输入 (TRGI) 的上升沿驱动计数器。
	1000 (结合 SMS[3]): 触发复位模式 - 计数器在触发输入 (TRGI) 的上升沿启动, 重新初始化计数器, 并且产生一个更新寄存器的信号。
	<i>注: 如果 TI1F_ED 被选为触发输入 (TS=100) 时, 不要使用门控模式。这是因为 TI1F_ED 在每次 TI1F 变化时只是输出一个脉冲, 然而门控模式要检查触发输入的电平。</i>

13.10.4 DIER DMA 和中断控制寄存器 (TIM1_DIER)

地址偏移: 0x0C

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:15	保留, 必须保持复位值
Bit 14	TDE: 触发 DMA 请求使能
	0: 触发 DMA 请求禁止 (默认)

	1: 触发 DMA 请求使能
Bit 13	COMDE: COM 的 DMA 请求使能
	0: COM 的 DMA 请求禁止 (默认)
	1: COM 的 DMA 请求使能
Bit 12	CC4DE: 捕获/比较 4 的 DMA 请求使能
	0: 捕获/比较 4 的 DMA 请求禁止 (默认)
	1: 捕获/比较 4 的 DMA 请求使能
Bit 11	CC3DE: 捕获/比较 3 的 DMA 请求使能
	0: 捕获/比较 3 的 DMA 请求禁止 (默认)
	1: 捕获/比较 3 的 DMA 请求使能
Bit 10	CC2DE: 捕获/比较 2 的 DMA 请求使能
	0: 捕获/比较 2 的 DMA 请求禁止 (默认)
	1: 捕获/比较 2 的 DMA 请求使能
Bit 9	CC1DE: 捕获/比较 1 的 DMA 请求使能
	0: 捕获/比较 1 的 DMA 请求禁止 (默认)
	1: 捕获/比较 1 的 DMA 请求使能
Bit 8	UDE: 更新的 DMA 请求使能
	0: 更新的 DMA 请求禁止 (默认)
	1: 更新的 DMA 请求使能
Bit 7	BIE: 刹车中断使能
	0: 刹车中断禁止 (默认)
	1: 刹车中断使能
Bit 6	TIE: 触发中断使能
	0: 触发中断禁止 (默认)
	1: 触发中断使能
Bit 5	COMIE: COM 中断使能
	0: COM 中断禁止 (默认)
	1: COM 中断使能
Bit 4	CC4IE: 捕获/比较 4 中断使能
	0: 捕获/比较 4 中断禁止 (默认)
	1: 捕获/比较 4 中断使能
Bit 3	CC3IE: 捕获/比较 3 中断使能
	0: 捕获/比较 3 中断禁止 (默认)
	1: 捕获/比较 3 中断使能
Bit 2	CC2IE: 捕获/比较 2 中断使能
	0: 捕获/比较 2 中断禁止 (默认)
	1: 捕获/比较 2 中断使能
Bit 1	CC1IE: 捕获/比较 1 中断使能
	0: 捕获/比较 1 中断禁止 (默认)
	1: 捕获/比较 1 中断使能
Bit 0	UIE: 更新中断使能
	0: 更新中断禁止 (默认)
	1: 更新中断使能



13.10.5 SR 状态寄存器 (TIM1_SR)

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
Res.															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			CC4OF	CC3OF	CC2OF	CC1OF	B2IF	B1F	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	31:13	保留, 必须保持复位值
Bit	12	CC4OF : 捕获/比较 4 过捕获/过比较标志, 参考 CC1OF 位
Bit	11	CC3OF : 捕获/比较 3 过捕获/过比较标志, 参考 CC1OF 位
Bit	10	CC2OF : 捕获/比较 2 过捕获/过比较标志, 参考 CC1OF 位
Bit	9	CC1OF : 捕获/比较 1 过捕获/过比较标志
		该位可由硬件置 1, 软件向 CC1IF 位写 0 可清除该位
		0: 无过捕获/过比较产生 (默认)
		1: 计数器的值被捕获或匹配到 TIM1_CCR1 寄存器时 CC1IF 已经置 1
Bit	8	B2IF : 刹车 2 中断标志
		刹车 2 输入一旦有效, 该位由硬件置 1, 刹车 2 输入无效后可以由软件写 0 清除
		0: 无刹车事件产生 (默认)
		1: 在刹车 2 输入上检测到有效电平
Bit	7	B1F : 刹车中断标志
		刹车输入一旦有效, 该位由硬件置 1, 刹车输入无效后可以由软件写 0 清除
		0: 无刹车事件产生 (默认)
		1: 刹车输入上检测到有效电平
Bit	6	TIF : 触发中断标志
		当发生触发事件(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) 时该位由硬件置 1, 软件写 0 可清除该位
		0: 无触发事件产生 (默认)
		1: 触发中断挂起
Bit	5	COMIF : COM 中断标志
		一旦产生 COM 事件(当捕获/比较控制位 CCiE、CCiNE、OCiM 被更新) 时硬件置位该寄存器位, 软件写 0 可清除该位
		0: 无 COM 事件产生 (默认)
		1: COM 中断挂起
Bit	4	CC4IF : 捕获/比较 4 中断标志, 参考 CC1IF 位
Bit	3	CC3IF : 捕获/比较 3 中断标志, 参考 CC1IF 位
Bit	2	CC2IF : 捕获/比较 2 中断标志, 参考 CC1IF 位
Bit	1	CC1IF : 捕获/比较 1 中断标志
		如果通道 CC1 配置为输出模式:

	当计数器值与比较值匹配时该位由硬件置 1，但在中央对齐模式下除外（参考 TIM1_CR1 寄存器的 CMS 位）。软件写 0 可清除该位，但是当 CC1OF 也为 1 时，需要清除 2 次。
	0: 无匹配发生（默认）
	1: TIM1_CNT 的值与 TIM1_CCR1 的值匹配
	注： 在中央对齐模式下，当计数器值为 0 时，向上计数，当计数器值为 ARR 时，向下计数（它从 0 向上计数到 ARR-1，再由 ARR 向下计数到 1）。因此，对所有的 SMS 位值，这两个值都不会置位 CC1IF。但是，如果 CCR1>ARR，则当 CNT 达到 ARR 值时，CC1IF 置 1。
	如果通道 CC1 配置为输入模式：
	当捕获事件发生时该位由硬件置 1，软件写 0 或通过读 TIM1_CCR1 可清除该位，但是当 CC1OF 也为 1 时，需要清除 2 次。
	0: 无输入捕获产生（默认）
	1: 计数器值已被捕获至 TIM1_CCR1（在 IC1 上检测到与所选极性相同的边沿）
Bit 0	UIF: 更新中断标志
	当产生更新事件时该位由硬件置 1，软件写 0 可清除该位
	0: 无更新事件产生（默认）
	1: 更新中断挂起，当相关寄存器被更新时该位由硬件置 1
	- 若 TIM1_CR1 寄存器的 UDIS=0，当计数器上溢或下溢时
	- 若 TIM1_CR1 寄存器的 UDIS=0、URS=0，当软件设置 TIM1_EGR 寄存器的 UG 位对计数器 CNT 重新初始化时
	- 若 TIM1_CR1 寄存器的 UDIS=0、URS=0，当计数器 CNT 被触发事件重新初始化时

13.10.6 EGR 事件产生寄存器（TIM1_EGR）

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
Res.															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.							B2G	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
							w	w	w	w	w	w	w	w	w

Bits 31:9	保留，必须保持复位值
Bit 8	B2G: 产生刹车 2 事件
	该位由软件置 1，用于产生一个刹车事件，由硬件自动清 0
	0: 无动作（默认）
	1: 产生刹车 2 事件。此时 MOE=0、B2IF=1，若 BIE=1，则产生相应的中断
Bit 7	BG: 产生刹车事件
	该位由软件置 1，用于产生一个刹车事件，由硬件自动清 0
	0: 无动作（默认）
	1: 产生刹车事件。此时 MOE=0、BIF=1，若 BIE=1，则产生相应的中断

Bit 6	TG: 产生触发事件
	该位由软件置 1, 用于产生一个触发事件, 由硬件自动清 0
	0: 无动作 (默认)
	1: 产生触发事件。此时 TIF=1, 若 TIE=1, 则产生相应的中断
Bit 5	COMG: 产生捕获/比较控制更新事件
	该位由软件置 1, 用于产生一个捕获/比较控制更新事件, 由硬件自动清 0
	0: 无动作 (默认)
	1: 当 CCPC=1 时, 允许更新 CCiE、CCiNE、CCiP、CCiNP 和 OCiM 位。此时 COMIF=1, 若 COMIE=1, 则产生相应的中断
	<i>注: 该位只对拥有互补输出的通道有效。</i>
Bit 4	CC4G: 产生捕获/比较 4 事件, 参考 CC1G 位
Bit 3	CC3G: 产生捕获/比较 3 事件, 参考 CC1G 位
Bit 2	CC2G: 产生捕获/比较 2 事件, 参考 CC1G 位
Bit 1	CC1G: 产生捕获/比较 1 事件
	该位由软件置 1, 用于产生一个捕获/比较事件, 由硬件自动清 0
	0: 无动作 (默认)
	1: 在通道 CC1 上产生一个捕获/比较事件
	若通道 CC1 配置为输出:
	设置 CC1IF=1, 若 CC1IE=1, 则产生相应的中断; 若 CC1IF 已经为 1, 则设置 CC1OF=1
	若通道 CC1 配置为输入:
	当前的计数器值被捕获至 TIM1_CCR1 寄存器, 设置 CC1IF=1, 若 CC1IE=1, 则产生相应的中断; 若 CC1IF 已经为 1, 则设置 CC1OF=1
Bit 0	UG: 产生更新事件
	该位由软件置 1, 用于产生一个更新事件, 由硬件自动清 0
	0: 无动作 (默认)
	1: 重新初始化计数器, 并产生一个更新事件。注意预分频器的计数器也被清 0 (但是预分频系数不变)。若中央对齐模式或向上计数 (DIR=0) 则计数器被清 0; 若向下计数 (DIR=1) 则计数器取 TIM1_ARR 的值

13.10.7 CCMR1 捕获/比较模式寄存器 1 (TIM1_CCMR1)

地址偏移: 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 CCiS 位定义。该寄存器部分位的作用在输入和输出模式下不同。OCxx 描述了通道在输出模式下的功能, ICxx 描述了通道在输入模式下的功能。因此必须注意, 同一个寄存器位在输出模式和输入模式下的功能是不同的。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OC2TE	OC2TUE	Res.						OC1TE	OC1TUE	Res.					
rw	rw							rw	rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]		OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]		OC1PE	OC1FE	CC1S[1:0]			
IC2F[3:0]			IC2PSC[1:0]		rw		IC1F[3:0]			IC1PSC[1:0]		rw			
rw			rw		rw		rw			rw		rw			

输出比较模式

Bit	31	OC2TE : 输出比较 2 修调使能, 参考 OC1TE
Bit	30	OC2TUE : 输出比较 2 修调更新使能, 参考 OC1TUE
Bits	29:24	保留, 必须保持复位值
Bit	23	OC1TE : 输出比较 1 修调使能, 与 CCRi 寄存器配合使用
		0: 禁止 (默认)
		1: 使能
Bit	22	OC1TUE : 输出比较 1 修调更新使能, 与 CCRi 寄存器配合使用
		0: 禁止 (默认)
		1: 使能
Bits	21:16	保留, 必须保持复位值
Bit	15	OC2CE : 输出比较 2 清除使能, 参考 OC1CE
Bits	14:12	OC2M[2:0] : 输出比较 2 模式, 参考 OC1M
Bit	11	OC2PE : 输出比较 2 预装载使能, 参考 OC1PE
Bit	10	OC2FE : 输出比较 2 快速使能, 参考 OC1FE
Bits	9:8	CC2S[1:0] : 捕获/比较 2 选择
		该位定义通道的方向 (输入/输出), 以及输入脚的选择
		00: CC2 通道被配置为输出 (默认)
		01: CC2 通道被配置为输入, IC2 映射在 TI2FP2 上
		10: CC2 通道被配置为输入, IC2 映射在 TI1FP2 上
		11: CC2 通道被配置为输入, IC2 映射在 TRC 上。此模式仅工作在触发输入选中 ITRx 或 TI1F_ED 时 (由 TIM1_SMCR 寄存器的 TS 位选择)
		<i>注: CC2S 仅在通道关闭时 (TIM1_CCER 寄存器的 CC2E=0、CC2NE=0 且已被更新) 才是可写的。</i>
Bit	7	OC1CE : 输出比较 1 清零使能
		0: OC1REF 清除禁止 (默认)
		1: OC1REF 清除使能
Bits	6:4	OC1M[2:0] : 输出比较 1 模式选择
		这 3 位定义了输出参考信号 OC1REF 的动作, 而 OC1REF 决定了 OC1 和 OC1N 的值。OC1REF 是高电平有效, 而 OC1 和 OC1N 的有效电平取决于 CC1P 和 CC1NP。
		000: 冻结。输出比较寄存器 TIM1_CCR1 与计数器 TIM1_CNT 间的比较对 OC1REF 不起作用。(默认)
		001: 匹配时设置通道 1 的输出为有效电平。当计数器 TIM1_CNT 的值与捕获/比较寄存器 1 (TIM1_CCR1) 相同时, 强制 OC1REF 为高。
		010: 匹配时设置通道 1 的输出为无效电平。当计数器 TIM1_CNT 的值与捕获/比较寄存器 1 (TIM1_CCR1) 相同时, 强制 OC1REF 为低。
		011: 翻转。当 TIM1_CCR1=TIM1_CNT 时, 翻转 OC1REF 的电平。
		100: 强制为无效电平。强制 OC1REF 为低。
		101: 强制为有效电平。强制 OC1REF 为高。
		110: PWM 模式 1 — 在向上计数时, 一旦 TIM1_CNT<TIM1_CCR1 时通道 1 为有效电平, 否则为无效电平; 在向下计数时, 一旦 TIM1_CNT>TIM1_CCR1 时通道 1 为无效电平, 否则为有效电平。
		111: PWM 模式 2 — 在向上计数时, 一旦 TIM1_CNT<TIM1_CCR1 时通道 1 为无效

	电平，否则为有效电平；在向下计数时，一旦 TIM1_CNT>TIM1_CCR1 时通道 1 为有效电平，否则为无效电平。
	注 1: 一旦 LOCK 级别设为 3 (TIM1_BDTR 寄存器中的 LOCK 位) 并且 CC1S=00 (该通道配置成输出) 则该位不能被修改。
	注 2: 在 PWM 模式 1 或 PWM 模式 2 中，只有当比较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时，OC1REF 电平才改变。(参考章节 13.6.7 PWM 模式)。
	注 3: 在有互补输出的通道上，这些位是预装载的。如果 TIM1_CR2 寄存器的 CCPC=1，OC1M 只有在 COM 事件发生时，才会预装载新的值。
Bit 3	OC1PE: 输出比较 1 预装载使能
	0: 禁止 TIM1_CCR1 寄存器的预装载功能，可随时写入 TIM1_CCR1 寄存器，并且新写入的数值立即起作用 (默认)
	1: 使能 TIM1_CCR1 寄存器的预装载功能，读写操作仅对预装载寄存器操作，TIM1_CCR1 的预装载值在更新事件到来时被加载至当前寄存器中
	注 1: 一旦 LOCK 级别设为 3 (TIM1_BDTR 寄存器中的 LOCK 位) 并且 CC1S=00 (该通道配置成输出) 则该位不能被修改。
	注 2: 为了操作正确，在 PWM 模式下必须使能预装载功能。但在单脉冲模式下 (TIM1_CR1 寄存器的 OPM=1)，它不是必须的。
Bit 2	OC1FE: 输出比较 1 快速使能
	该位用于加快 CC 输出对触发输入的响应
	0: 根据计数器与 CCR1 的值，CC1 正常操作。当触发输入出现一个有效边沿时，通过计数器比较输出 CC1。(默认)
	1: 触发输入的有效边沿的作用就如同发生了一次比较匹配。因此，OC1REF 被直接设置为有效电平，而与比较结果无关。触发输入的有效边沿与 CC1 输出之间的延时被缩短。
	注: OC1FE 只在通道被配置成 PWM 模式 1/2 的单脉冲模式时起作用。
Bits 1:0	CC1S[1:0]: 捕获/比较 1 选择
	该位定义通道的方向 (输入/输出)，以及输入脚的选择
	00: CC1 通道被配置为输出 (默认)
	01: CC1 通道被配置为输入，IC1 映射在 TI1FP1 上
	10: CC1 通道被配置为输入，IC1 映射在 TI2FP1 上
	11: CC1 通道被配置为输入，IC1 映射在 TRC 上。此模式仅工作在触发输入选中 ITRx 或 TI1F_ED 时 (由 TIM1_SMCR 寄存器的 TS 位选择)
	注: CC1S 仅在通道关闭时 (TIM1_CCER 寄存器的 CC1E=0、CC1NE=0 且已被更新) 才是可写的。

输入捕获模式

Bits 31:16	保留，必须保持复位值
Bits 15:12	IC2F[3:0]: 输入捕获 2 滤波器，参考 IC1F
Bits 11:10	IC2PSC[1:0]: 输入捕获 2 预分频器，参考 IC1PSC
Bits 9:8	CC2S[1:0]: 捕获/比较 2 选择
	该位定义通道的方向 (输入/输出)，以及输入脚的选择
	00: CC2 通道被配置为输出 (默认)
	01: CC2 通道被配置为输入，IC2 映射在 TI2FP2 上
	10: CC2 通道被配置为输入，IC2 映射在 TI1FP2 上

	11: CC2 通道被配置为输入, IC2 映射在 TRC 上。此模式仅工作在触发输入选中 ITRx 或 TI1F_ED 时 (由 TIM1_SMCR 寄存器的 TS 位选择)
	注: CC2S 仅在通道关闭时 (TIM1_CCER 寄存器的 CC2E=0、CC2NE=0 且已被更新) 才是可写的。
Bits 7:4	IC1F[3:0]: 输入捕获 1 滤波器
	该位域定义了 TI1 输入的采样频率及数字滤波器的长度。数字滤波器由一个事件计数器组成, 只有发生了 N 个连续事件后输出的跳变才被认为有效。这里的 DTS 时钟是 CK_INT 经过 CR1 寄存器中 CKD 配置分频的。
	0000: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}$, 无滤波器 (默认)
	0001: 采样频率 $f_{\text{SAMPLING}} = f_{\text{CK_INT}}$, N=2
	0010: 采样频率 $f_{\text{SAMPLING}} = f_{\text{CK_INT}}$, N=4
	0011: 采样频率 $f_{\text{SAMPLING}} = f_{\text{CK_INT}}$, N=8
	0100: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/2$, N=6
	0101: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/2$, N=8
	0110: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$, N=6
	0111: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$, N=8
	1000: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/8$, N=6
	1001: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/8$, N=8
	1010: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$, N=5
	1011: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$, N=6
	1100: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$, N=8
	1101: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$, N=5
	1110: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$, N=6
	1111: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$, N=8
	注 1: 请注意采样频率中 $f_{\text{CK_INT}}$ 和 f_{DTS} 的差别。
	注 2: 即使对于带互补输出的通道, 该位域也是非预装载的, 并且不会考虑 CCPC (TIM1_CR2 寄存器) 的值。
Bits 3:2	IC1PSC[1:0]: 输入捕获 1 预分频器
	IC1 的预分频系数, 一旦 CC1E=0 (TIM1_CCER 寄存器中), 则预分频器复位
	00: 无预分频器, TI1 上每 1 个有效边沿作为一次输入 (默认)
	01: TI1 上每 2 个有效边沿作为一次输入
	10: TI1 上每 4 个有效边沿作为一次输入
	11: TI1 上每 8 个有效边沿作为一次输入
Bits 1:0	CC1S[1:0]: 捕获/比较 1 选择
	该位定义通道的方向 (输入/输出), 以及输入脚的选择
	00: CC1 通道被配置为输出 (默认)
	01: CC1 通道被配置为输入, IC1 映射在 TI1FP1 上
	10: CC1 通道被配置为输入, IC1 映射在 TI2FP1 上
	11: CC1 通道被配置为输入, IC1 映射在 TRC 上。此模式仅工作在触发输入选中 ITRx 或 TI1F_ED 时 (由 TIM1_SMCR 寄存器的 TS 位选择)
	注: CC1S 仅在通道关闭时 (TIM1_CCER 寄存器的 CC1E=0、CC1NE=0 且已被更新) 才是可写的。

13.10.8 CCMR2 捕获/比较模式寄存器 2 (TIM1_CCMR2)

地址偏移: 0x1C

复位值：0x0000_0000

参考 CCMR1 寄存器的描述。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OC4TE	OC4TUE	Res.						OC3TE	OC3TUE	Res.					
rw	rw							rw	rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]				IC3PSC[1:0]			
rw				rw			rw	rw				rw		rw	

输出比较模式

Bit	31	OC4TE : 输出比较 4 修调使能, 参考 OC1TE
Bit	30	OC4TUE : 输出比较 4 修调更新使能, 参考 OC1TUE
Bits	29:24	保留, 必须保持复位值
Bit	23	OC3TE : 输出比较 3 修调使能, 参考 OC1TE
Bit	22	OC3TUE : 输出比较 3 修调更新使能, 参考 OC1TUE
Bits	21:16	保留, 必须保持复位值
Bit	15	OC4CE : 输出比较 4 清除使能, 参考 OC1CE
Bits	14:12	OC4M[2:0] : 输出比较 4 模式, 参考 OC1M
Bit	11	OC4PE : 输出比较 4 预装载使能, 参考 OC1PE
Bit	10	OC4FE : 输出比较 4 快速使能, 参考 OC1FE
Bits	9:8	CC4S[1:0] : 捕获/比较 4 选择
		该位定义通道的方向 (输入/输出), 以及输入脚的选择
		00: CC4 通道被配置为输出 (默认)
		01: CC4 通道被配置为输入, IC4 映射在 TI4FP4 上
		10: CC4 通道被配置为输入, IC4 映射在 TI3FP4 上
		11: CC4 通道被配置为输入, IC4 映射在 TRC 上。此模式仅工作在触发输入选中 ITRx 或 TI1F_ED 时 (由 TIM1_SMCR 寄存器的 TS 位选择)
		<i>注: CC4S 仅在通道关闭时 (TIM1_CCER 寄存器的 CC4E=0、CC4NE=0 且已被更新) 才是可写的。</i>
Bit	7	OC3CE : 输出比较 3 清除使能, 参考 OC1CE
Bits	6:4	OC3M[2:0] : 输出比较 3 模式, 参考 OC1M
Bit	3	OC3PE : 输出比较 3 预装载使能, 参考 OC1PE
Bit	2	OC3FE : 输出比较 3 快速使能, 参考 OC1FE
Bits	1:0	CC3S[1:0] : 捕获/比较 3 选择
		该位定义通道的方向 (输入/输出), 以及输入脚的选择
		00: CC3 通道被配置为输出 (默认)
		01: CC3 通道被配置为输入, IC3 映射在 TI3FP3 上
		10: CC3 通道被配置为输入, IC3 映射在 TI4FP3 上
		11: CC3 通道被配置为输入, IC3 映射在 TRC 上。此模式仅工作在触发输入选中 ITRx 或 TI1F_ED 时 (由 TIM1_SMCR 寄存器的 TS 位选择)
		<i>注: CC3S 仅在通道关闭时 (TIM1_CCER 寄存器的 CC3E=0、CC3NE=0 且已被更新) 才是可写的。</i>

输入捕获模式

Bits 31:16	保留，必须保持复位值
Bits 15:12	IC4F[3:0] : 输入捕获 4 滤波器，参考 IC1F
Bits 11:10	IC4PSC[1:0] : 输入捕获 4 预分频器，参考 IC1PSC
Bits 9:8	CC4S[1:0] : 捕获/比较 4 选择
	该位定义通道的方向（输入/输出），以及输入脚的选择
	00: CC4 通道被配置为输出（默认）
	01: CC4 通道被配置为输入，IC4 映射在 TI4FP4 上
	10: CC4 通道被配置为输入，IC4 映射在 TI3FP4 上
	11: CC4 通道被配置为输入，IC4 映射在 TRC 上。此模式仅工作在触发输入选中 ITRx 或 TI1F_ED 时（由 TIM1_SMCR 寄存器的 TS 位选择）
	<i>注: CC4S 仅在通道关闭时（TIM1_CCER 寄存器的 CC4E=0、CC4NE=0 且已被更新）才是可写的。</i>
Bits 7:4	IC3F[3:0] : 输入捕获 3 滤波器，参考 IC1F
Bits 3:2	IC3PSC[1:0] : 输入捕获 3 预分频器，参考 IC1PSC
Bits 1:0	CC3S[1:0] : 捕获/比较 3 选择
	该位定义通道的方向（输入/输出），以及输入脚的选择
	00: CC3 通道被配置为输出（默认）
	01: CC3 通道被配置为输入，IC3 映射在 TI3FP3 上
	10: CC3 通道被配置为输入，IC3 映射在 TI4FP3 上
	11: CC3 通道被配置为输入，IC3 映射在 TRC 上。此模式仅工作在触发输入选中 ITRx 或 TI1F_ED 时（由 TIM1_SMCR 寄存器的 TS 位选择）
	<i>注: CC3S 仅在通道关闭时（TIM1_CCER 寄存器的 CC3E=0、CC3NE=0 且已被更新）才是可写的。</i>

13.10.9 CCER 捕获/比较使能寄存器（TIM1_CCER）

地址偏移: 0x20

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	CC4NE	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16	保留，必须保持复位值
Bit 15	CC4NP : 捕获/比较 4 互补输出极性，参考 CC1NP
Bit 14	CC4NE : 捕获/比较 4 互补输出使能，参考 CC1NE
Bit 13	CC4P : 捕获/比较 4 极性，参考 CC1P
Bit 12	CC4E : 捕获/比较 4 使能，参考 CC1E
Bit 11	CC3NP : 捕获/比较 3 互补输出极性，参考 CC1NP
Bit 10	CC3NE : 捕获/比较 3 互补输出使能，参考 CC1NE
Bit 9	CC3P : 捕获/比较 3 极性，参考 CC1P

Bit 8	CC3E: 捕获/比较 3 使能, 参考 CC1E
Bit 7	CC2NP: 捕获/比较 2 互补输出极性, 参考 CC1NP
Bit 6	CC2NE: 捕获/比较 2 互补输出使能, 参考 CC1NE
Bit 5	CC2P: 捕获/比较 2 极性, 参考 CC1P
Bit 4	CC2E: 捕获/比较 2 使能, 参考 CC1E
Bit 3	CC1NP: 捕获/比较 1 互补输出极性
	CC1 通道作为输出:
	0: OC1N 高电平有效 (默认)
	1: OC1N 低电平有效
	CC1 通道作为输入:
	该位与 CC1P 共同使用, 定义 TI1FP1 和 TI2FP1 的极性, 具体请参考 CC1P
	注 1: 一旦 LOCK 级别设为 3 或 2 (TIM1_BDTR 寄存器中的 LOCK 位) 并且 CC1S=00 (该通道配置成输出) 则该位不能被修改。
	注 2: 对于有互补输出的通道, 该位是预装载的。如果 CCPC=1 (TIM1_CR2 寄存器), 只有在 COM 事件发生时, CC1NP 位才从预装载中读取新的值。
Bit 2	CC1NE: 捕获/比较 1 互补输出使能
	0: 关闭 — OC1N 输出禁止, 因此 OC1N 的输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。(默认)
	1: 开启 — OC1N 信号输出到对应的输出引脚, 其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。
	注: 对于有互补输出的通道, 该位是预装载的。如果 CCPC=1 (TIM1_CR2 寄存器), 只有在 COM 事件发生时, CC1NE 位才从预装载中读取新的值。
Bit 1	CC1P: 捕获/比较 1 极性
	CC1 通道作为输出:
	0: OC1 高电平有效 (默认)
	1: OC1 低电平有效
	CC1 通道作为输入:
	CC1NP 和 CC1P 位共同选择 TI1FP1 和 TI2FP1 作为触发或捕获的极性
	00: 信号不反相, TIiFP1 的上升沿有效 (默认)
	01: 信号反相, TIiFP1 的下降沿有效
	10: 保留
	11: 信号不反相, TIiFP1 的上升沿和下降沿均有效; 这个配置不能出现在编码器接口模式中
	注 1: 一旦 LOCK 级别设为 3 或 2 (TIM1_BDTR 寄存器中的 LOCK 位) 则该位不能被修改。
	注 2: 对于有互补输出的通道, 该位是预装载的。如果 CCPC=1 (TIM1_CR2 寄存器), 只有在 COM 事件发生时, CC1P 位才从预装载中读取新的值。
Bit 0	CC1E: 捕获/比较 1 使能
	CC1 通道作为输出:
	0: 关闭 — OC1 输出禁止, 因此 OC1 的输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。(默认)
	1: 开启 — OC1 信号输出到对应的输出引脚, 其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。
	CC1 通道作为输入:

	该位决定了计数器的值是否能捕获入 TIM1_CCR1 寄存器
	0: 捕获禁止 (默认)
	1: 捕获使能
	注: 对于有互补输出的通道, 该位是预装载的。如果 CCPC=1 (TIM1_CR2 寄存器), 只有在 COM 事件发生时, CC1E 位才从预装载中读取新的值。

表 13-5 带刹车功能的互补输出 OCi 和 OCiN 的控制

控制位					输出状态	
MOE	OSSI	OSSR	CCiE	CCiNE	OCi 输出状态	OCiN 输出状态
1	X	0	0	0	输出禁止 (与定时器断开) OCi=0, OCiN=0	
		0	0	1	输出禁止 (与定时器断开) OCi=0	OCiREF + 极性, OCiN=OCiREF xor CCiNP
		0	1	0	OCiREF + 极性, OCi=OCiREF xor CCiP	输出禁止 (与定时器断开) OCiN=0
		0	1	1	OCiREF + 极性 + 死区	OCiREF 反相 + 极性 + 死区
		1	0	0	输出禁止 (与定时器断开) OCi=CCiP, OCiN=CCiNP	
		1	0	1	关闭状态 (输出使能且为无效电平) OCi=CCiP	OCiREF + 极性, OCiN=OCiREF xor CCiNP
		1	1	0	OCiREF + 极性, OCi=OCiREF xor CCiP	关闭状态 (输出使能且为无效电平) OCiN=CCiNP
		1	1	1	OCiREF + 极性 + 死区	OCiREF 反相 + 极性 + 死区
0	X	0	0	0	输出禁止 (与定时器断开) OCi=CCiP, OCiN=CCiNP	
		0	0	1	首先处于输出禁止状态 (与定时器断开), OCi=CCiP,	
		0	1	0	OCiN=CCiNP, 死区时钟存在且经过死区时间后, 如果 CCiP=OISi	
		0	1	1	且 CCiNP=OISiN, 则继续输出无效电平; 否则输出空闲电平, 即	
		0	1	1	OCi=OISi, OCiN=OISiN	
		1	0	0	输出禁止 (与定时器断开) OCi=CCiP, OCiN=CCiNP	
		1	1	0	首先处于关闭状态 (输出使能且为无效电平), OCi=CCiP,	
		1	0	1	OCiN=CCiNP, 死区时钟存在且经过死区时间后, 如果 CCiP≠OISi	
1	1	1	且 CCiPN≠OISiN, 则继续输出无效电平; 否则输出空闲电平, 即			
1	1	1	OCi=OISi, OCiN=OISiN			

注: 管脚连接到互补的 OCi 和 OCiN 通道的外部 I/O 管脚的状态, 取决于 OCi 和 OCiN 通道状态和 GPIO 寄存器。

13.10.10 CNT 计数寄存器 (TIM1_CNT)

地址偏移: 0x24

复位值: 0x0000_0000

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

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

Bits 31:16	保留，必须保持复位值
Bits 15:0	CNT[15:0] : 计数器的值

13.10.11 PSC 预分频寄存器 (TIM1_PSC)

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
Res.															
rw															

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

Bits 31:16	保留，必须保持复位值
Bits 15:0	PSC[15:0] : 预分频器的值
	计数器的频率可以由下式计算: $f_{CK_CNT} = f_{CK_PSC} / (PSC[15:0] + 1)$ 。 预分频器的值由预装载寄存器写入，新的预分频器的值在下一次更新事件到来时被采用。可以通过 AS 位 (TIM1_CR1 寄存器) 来选择读取 PSC 寄存器的值来自影子寄存器或预装载寄存器。

13.10.12 ARR 自动重载寄存器 (TIM1_ARR)

地址偏移: 0x2C

复位值: 0x0000_0000

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

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

Bits 31:16	保留，必须保持复位值
Bits 15:0	ARR[15:0] : 自动重载寄存器的值
	ARR 是要加载到自动重载寄存器中的值。具体请参考 13.2.1 章节。可以通过 AS 位 (TIM1_CR1 寄存器) 来选择读取 ARR 寄存器的值来自影子寄存器或预装载寄存器。

13.10.13 RCR 重复计数寄存器 (TIM1_RCR)

地址偏移: 0x30

复位值: 0x0000_0000

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

Bits 31:8	保留, 必须保持复位值
Bits 7:0	REP[7:0]: 重复计数器的值
	允许用户设置更新速率; 如果允许产生更新中断, 则会同时影响产生更新中断的速率。
	每次向下计数器 REP_CNT 达到 0, 会产生一个更新事件 UEV, 并且计数器 REP_CNT 重新从 REP 值开始计数。由于 REP_CNT 只有在更新事件发生时才重载 REP 值, 因此对 TIM1_RCR 寄存器写入的新值只在下次更新事件发生时才起作用。
	这意味着在 PWM 模式中, REP+1 对应着:
	<ul style="list-style-type: none"> - 在边沿对齐模式下, PWM 周期的数目 - 在中央对齐模式下, PWM 半周期的数目

13.10.14 CCR1 捕获/比较寄存器 1 (TIM1_CCR1)

地址偏移: 0x34

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw															

Bits 31:16	保留, 必须保持复位值
Bits 15:0	CCR1[15:0]: 捕获/比较 1 的值
	CC1 通道作为输出:
	CCR1 包含了要与计数器进行输出比较的值, 它与计数器 TIM1_CNT 的值相比较, 并在 OC1 端口上产生输出信号
	如果在 OC1PE 位 (TIM1_CCMR1 寄存器) 未选择预装载功能, 写入的数值会立即传输至影子寄存器中; 否则只有当更新事件发生时, 此预装载值才传输至影子寄存器中
	当刹车触发且 AOE=1 (TIM1_BDTR 寄存器) 时, CCR1 的值可以通过相应的



	TIM1_CCTR1 修调寄存器来更新：如果 TIM1_CCMR1 寄存器中的 OC1TE=1，当更新事件 UEV 发生时，CCR1 的影子寄存器被 TIM1_CCTR1 的值更新；如果 TIM1_CCMR1 寄存器中的 OC1TE=1 且 OC1TUE=1，当更新事件 UEV 发生时，CCR1 的影子寄存器和预装载寄存器都被 TIM1_CCTR1 的值更新
	可以通过 AS 位（TIM1_CR1 寄存器）来选择读取 CCR1 寄存器的值来自影子寄存器或预装载寄存器
	CC1 通道作为输入：
	CCR1 包含了上一次输入捕获 1 事件（IC1）发生时的计数器值，此时该寄存器为只读

13.10.15 CCR2 捕获/比较寄存器 2（TIM1_CCR2）

地址偏移：0x38

复位值：0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw															

Bits 31:16	保留，必须保持复位值
Bits 15:0	CCR2[15:0] ：捕获/比较 2 的值
	CC2 通道作为输出：
	CCR2 包含了要与计数器进行输出比较的值，它与计数器 TIM1_CNT 的值相比较，并在 OC2 端口上产生输出信号
	如果在 OC2PE 位（TIM1_CCMR1 寄存器）未选择预装载功能，写入的数值会立即传输至影子寄存器中；否则只有当更新事件发生时，此预装载值才传输至影子寄存器中
	当刹车触发且 AOE=1（TIM1_BDTR 寄存器）时，CCR2 的值可以通过相应的 TIM1_CCTR2 修调寄存器来更新：如果 TIM1_CCMR1 寄存器中的 OC2TE=1，当更新事件 UEV 发生时，CCR2 的影子寄存器被 TIM1_CCTR2 的值更新；如果 TIM1_CCMR1 寄存器中的 OC2TE=1 且 OC2TUE=1，当更新事件 UEV 发生时，CCR2 的影子寄存器和预装载寄存器都被 TIM1_CCTR2 的值更新
	可以通过 AS 位（TIM1_CR1 寄存器）来选择读取 CCR2 寄存器的值来自影子寄存器或预装载寄存器
	CC2 通道作为输入：
	CCR2 包含了上一次输入捕获 2 事件（IC2）发生时的计数器值，此时该寄存器为只读

13.10.16 CCR3 捕获/比较寄存器 3（TIM1_CCR3）

地址偏移：0x3C

复位值：0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
rw															

Bits 31:16	保留，必须保持复位值
Bits 15:0	CCR3[15:0] : 捕获/比较 3 的值
	CC3 通道作为输出:
	CCR3 包含了要与计数器进行输出比较的值，它与计数器 TIM1_CNT 的值相比较，并在 OC3 端口上产生输出信号
	如果在 OC3PE 位 (TIM1_CCMR2 寄存器) 未选择预装载功能，写入的数值会立即传输至影子寄存器中；否则只有当更新事件发生时，此预装载值才传输至影子寄存器中
	当刹车触发且 AOE=1 (TIM1_BDTR 寄存器) 时，CCR3 的值可以通过相应的 TIM1_CCTR3 修调寄存器来更新：如果 TIM1_CCMR2 寄存器中的 OC3TE=1，当更新事件 UEV 发生时，CCR3 的影子寄存器被 TIM1_CCTR3 的值更新；如果 TIM1_CCMR2 寄存器中的 OC3TE=1 且 OC3TUE=1，当更新事件 UEV 发生时，CCR3 的影子寄存器和预装载寄存器都被 TIM1_CCTR3 的值更新
	可以通过 AS 位 (TIM1_CR1 寄存器) 来选择读取 CCR3 寄存器的值来自影子寄存器或预装载寄存器
	CC3 通道作为输入:
	CCR3 包含了上一次输入捕获 3 事件 (IC3) 发生时的计数器值，此时该寄存器为只读

13.10.17 CCR4 捕获/比较寄存器 4 (TIM1_CCR4)

地址偏移：0x40

复位值：0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
rw															

Bits 31:16	保留，必须保持复位值
Bits 15:0	CCR4[15:0] : 捕获/比较 4 的值
	CC4 通道作为输出:
	CCR4 包含了要与计数器进行输出比较的值，它与计数器 TIM1_CNT 的值相比较，

	并在 OC4 端口上产生输出信号
	如果在 OC4PE 位 (TIM1_CCMR2 寄存器) 未选择预装载功能, 写入的数值会立即传输至影子寄存器中; 否则只有当更新事件发生时, 此预装载值才传输至影子寄存器中
	当刹车触发且 AOE=1 (TIM1_BDTR 寄存器) 时, CCR4 的值可以通过相应的 TIM1_CCTR4 修调寄存器来更新: 如果 TIM1_CCMR2 寄存器中的 OC4TE=1, 当更新事件 UEV 发生时, CCR4 的影子寄存器被 TIM1_CCTR4 的值更新; 如果 TIM1_CCMR2 寄存器中的 OC4TE=1 且 OC4TUE=1, 当更新事件 UEV 发生时, CCR4 的影子寄存器和预装载寄存器都被 TIM1_CCTR4 的值更新
	可以通过 AS 位 (TIM1_CR1 寄存器) 来选择读取 CCR4 寄存器的值来自影子寄存器或预装载寄存器
	CC4 通道作为输入:
	CCR4 包含了上一次输入捕获 4 事件 (IC4) 发生时的计数器值, 此时该寄存器为只读

13.10.18 BDTR 刹车和死区控制寄存器 (TIM1_BDTR)

地址偏移: 0x44

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.					AOE2	BK2P	BK2E	Res.							
					rw	rw	rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
rw	rw	rw	rw	rw	rw	rw		rw							

Bits 31:27	保留, 必须保持复位值
Bit 26	AOE2: 自动输出使能 2
	0: MOE 只能被软件置 1 (默认)
	1: MOE 能被软件置 1 或在下一个更新事件被自动置 1 (如果刹车 2 输入无效)
	<i>注:</i> 一旦 LOCK 级别设为 1 (TIM1_BDTR 寄存器中的 LOCK 位) 则该位不能被修改。
Bit 25	BK2P: 刹车 2 输入极性
	0: 刹车 2 输入低电平有效 (默认)
	1: 刹车 2 输入高电平有效
	<i>注:</i> 一旦 LOCK 级别设为 1 (TIM1_BDTR 寄存器中的 LOCK 位) 则该位不能被修改。
Bit 24	BK2E: 刹车 2 功能使能
	0: 刹车 2 功能禁止 (默认)
	1: 刹车 2 功能使能
	<i>注:</i> 一旦 LOCK 级别设为 1 (TIM1_BDTR 寄存器中的 LOCK 位) 则该位不能被修改。
Bits 23:16	保留, 必须保持复位值
Bit 15	MOE: 主输出使能
	一旦刹车输入有效, 该位被硬件异步清 0。根据 AOE/AOE2 位的配置, 该位可以由软件置 1 或被自动置 1。它仅对配置为输出的通道有效。

	0: 禁止 OC _i 和 OC _{iN} 输出或强制为空闲状态 (默认)
	1: 输出使能, 如果设置了相应的使能位 (TIM1_CCER 寄存器的 CC _{iE} 位), 则使能 OC _i 和 OC _{iN} 输出
Bit 14	AOE: 自动输出使能
	0: MOE 只能被软件置 1 (默认)
	1: MOE 能被软件置 1 或在下一个更新事件被自动置 1 (如果刹车输入无效)
	<i>注: 一旦 LOCK 级别设为 1 (TIM1_BDTR 寄存器中的 LOCK 位) 则该位不能被修改。</i>
Bit 13	BKP: 刹车输入极性
	0: 刹车输入低电平有效 (默认)
	1: 刹车输入高电平有效
	<i>注: 一旦 LOCK 级别设为 1 (TIM1_BDTR 寄存器中的 LOCK 位) 则该位不能被修改。</i>
Bit 12	BKE: 刹车功能使能
	0: 刹车功能禁止 (默认)
	1: 刹车功能使能
	<i>注: 一旦 LOCK 级别设为 1 (TIM1_BDTR 寄存器中的 LOCK 位) 则该位不能被修改。</i>
Bit 11	OSSR: 运行模式下的关闭状态选择
	该位用于当 MOE=1 且通道为互补输出时, 具体描述请参考表格 13-5
	0: 当 OC _i /OC _{iN} 未使能时, OC _i /OC _{iN} 输出禁止 (默认)
	1: 当 OC _i /OC _{iN} 未使能时, 如果 CC _{iE} =1 或 CC _{iNE} =1, 则 OC _i /OC _{iN} 输出无效电平
	<i>注: 一旦 LOCK 级别设为 2 (TIM1_BDTR 寄存器中的 LOCK 位) 则该位不能被修改。</i>
Bit 10	OSSI: 空闲模式下的关闭状态选择
	该位用于当 MOE=0 且通道作为输出时, 具体描述请参考表格 13-5
	0: 当 OC _i /OC _{iN} 未使能时, OC _i /OC _{iN} 输出禁止 (默认)
	1: 当 OC _i /OC _{iN} 未使能时, 如果 CC _{iE} =1 或 CC _{iNE} =1, 则 OC _i /OC _{iN} 输出空闲电平
	<i>注: 一旦 LOCK 级别设为 2 (TIM1_BDTR 寄存器中的 LOCK 位) 则该位不能被修改。</i>
Bits 9:8	LOCK[1:0]: 锁定设置, 为防止软件错误而提供写保护
	00: 锁定关闭, 寄存器无写保护 (默认)
	01: 锁定级别 1, 不能写入 TIM1_BDTR 寄存器的 BKE、BKP、AOE、DTG 位, TIM1_CR2 寄存器的 OIS _i /OIS _{iN} 位和 TIM1_DLAMT 寄存器的所有位
	10: 锁定级别 2, 不能写入锁定级别 1, 也不能写入 CC 极性位 (当通道配置为输出时, TIM1_CCER 寄存器的 CC _{iP} /CC _{iNP} 位) 以及 OSSR/OSSI 位
	11: 锁定级别 3, 不能写入锁定级别 2, 也不能写入 CC 控制位 (当通道配置为输出时, TIM1_CCMR _i 寄存器的 OC _{iM} /OC _{iPE} 位)
	<i>注: 在系统复位后, 只能写一次 LOCK 位, 一旦写入 TIM1_BDTR 寄存器, 则其内容保持不变直至复位。</i>
Bits 7:0	DTG[7:0]: 死区发生器设置
	该位定义了插入互补输出之间的死区持续时间。假设 DT 表示其持续时间, t _{DTs} 为死区时钟。
	DTG[7:5]=0xx => DT=DTG[7:0]x t _{dtg} , 其中 t _{dtg} = t _{DTs}
	DTG[7:5]=10x => DT=(64+DTG[5:0])x t _{dtg} , 其中 t _{dtg} = 2x t _{DTs}
	DTG[7:5]=110 => DT=(32+DTG[4:0])x t _{dtg} , 其中 t _{dtg} = 8x t _{DTs}
	DTG[7:5]=111 => DT=(32+DTG[4:0])x t _{dtg} , 其中 t _{dtg} = 16x t _{DTs}
	举例, 如果 t _{DTs} =125 ns (8 MHz), 可能的死区时间为:

	DTG[7:0] = 0 到 7FH, 0 到 15875 ns, 步长时间为 125 ns
	DTG[7:0] = 80H 到 BFH, 16 μ s 到 31750 ns, 步长时间为 250 ns
	DTG[7:0] = C0H 到 DFH, 32 μ s 到 63 μ s, 步长时间为 1 μ s
	DTG[7:0] = E0H 到 FFH, 64 μ s 到 126 μ s, 步长时间为 2 μ s
	注: 一旦 LOCK 级别设为 1、2 或 3 (TIM1_BDTR 寄存器中的 LOCK 位) 则该位不能被修改。

13.10.19 DLAMT 触发延迟和防触发丢失寄存器 (TIM1_DLAMT)

地址偏移: 0x7C

复位值: 0x0000_0000

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

Bits 31:18	保留, 必须保持复位值
Bit 17	DLS: 触发延迟期间清除 OCiREF
	0: 禁止 (默认)
	1: 使能
	注: 一旦 LOCK 级别设为 1、2 或 3 (TIM1_BDTR 寄存器中的 LOCK 位) 则该位不能被修改。
Bit 16	AMTDL: 防触发丢失窗口开始时间
	0: 从延迟的触发输入开始 (默认)
	1: 从触发输入开始
	注: 一旦 LOCK 级别设为 1、2 或 3 (TIM1_BDTR 寄存器中的 LOCK 位) 则该位不能被修改。
Bits 15:8	ATMG[7:0]: 防触发丢失窗口时间设置。如果在触发延迟的过程中有新的触发输入, 可在防触发丢失窗口的时间内把新的触发保留住。
	该位定义了防触发丢失窗口的时间, 假设 AMT 表示其持续时间, t_{DTS} 为死区时钟。
	AMTG[7:5]=0xx => AMT=AMTG[7:0] \times t_{amtg} , 其中 $t_{amtg} = t_{DTS}$
	AMTG[7:5]=10x => AMT=(64+AMTG[5:0]) \times t_{amtg} , 其中 $t_{amtg} = 2x t_{DTS}$
	AMTG[7:5]=110 => AMT=(32+AMTG[4:0]) \times t_{amtg} , 其中 $t_{amtg} = 8x t_{DTS}$
	AMTG[7:5]=111 => AMT=(32+AMTG[4:0]) \times t_{amtg} , 其中 $t_{amtg} = 16x t_{DTS}$
	举例, 如果 $t_{DTS} = 125$ ns (8 MHz), 可能的防触发丢失窗口时间为:
	AMTG[7:0] = 0 到 7FH, 0 到 15875 ns, 步长时间为 125 ns
	AMTG[7:0] = 80H 到 BFH, 16 μ s 到 31750 ns, 步长时间为 250 ns
	AMTG[7:0] = C0H 到 DFH, 32 μ s 到 63 μ s, 步长时间为 1 μ s
	AMTG[7:0] = E0H 到 FFH, 64 μ s 到 126 μ s, 步长时间为 2 μ s
	注: 一旦 LOCK 级别设为 1、2 或 3 (TIM1_BDTR 寄存器中的 LOCK 位) 则该位不能被修改。
Bits 7:0	DLG[7:0]: 触发延迟窗口时间设置
	该位定义了触发延迟窗口的时间, 假设 DLT 表示其持续时间, t_{DTS} 为死区时钟。
	DLG[7:5]=0xx => DLT=DLG[7:0] \times t_{dlg} , 其中 $t_{dlg} = t_{DTS}$

	DLG[7:5]=10x => DLT=(64+DLG[5:0])x t _{dlg} , 其中 t _{dlg} =2x t _{DTS}
	DLG[7:5]=110 => DLT=(32+DLG[4:0])x t _{dlg} , 其中 t _{dlg} =8x t _{DTS}
	DLG[7:5]=111 => DLT=(32+DLG[4:0])x t _{dlg} , 其中 t _{dlg} =16x t _{DTS}
	举例, 如果 t _{DTS} =125 ns (8 MHz), 可能的触发延迟窗口时间为:
	DLG[7:0] = 0 到 7FH, 0 到 15875 ns, 步长时间为 125 ns
	DLG[7:0] = 80H 到 BFH, 16 μs 到 31750 ns, 步长时间为 250 ns
	DLG[7:0] = C0H 到 DFH, 32 μs 到 63 μs, 步长时间为 1 μs
	DLG[7:0] = E0H 到 FFH, 64 μs 到 126 μs, 步长时间为 2 μs
	注: 一旦 LOCK 级别设为 1、2 或 3 (TIM1_BDTR 寄存器中的 LOCK 位) 则该位不能被修改。

13.10.20 DCR DMA 控制寄存器 (TIM1_DCR)

地址偏移: 0x48

复位值: 0x0000_0000

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

Bits 31:13	保留, 必须保持复位值
Bits 12:8	DBL[4:0]: DMA burst 传输长度
	定义了 DMA 在 burst 模式下的传输长度 (当对 TIM1_DMAR 寄存器的地址进行读或写时, TIM1 进行一次 burst 传输), 即定义被传送的字节数目
	00000: 1 字节 (默认)
	00001: 2 字节
	00010: 3 字节
	...
	10001: 18 字节
Bits 7:5	DSEL[2:0]: DMA burst 传输请求选择
	定义了哪个 DMA 请求使用 burst 传输, 其它的未选中的则使用 single 传输
	000: 所有 DMA 请求都使用 single 传输
	001: DMA 请求 0 使用 burst 传输
	010: DMA 请求 1 使用 burst 传输
	011: DMA 请求 2 使用 burst 传输
	100: DMA 请求 3 使用 burst 传输
	101 至 111: 保留
Bits 4:0	DBA[4:0]: DMA 基地址
	该位定义了 DMA 在 burst 模式下的基地址 (当对 TIM1_DMAR 寄存器的地址进行读或写时), DBA 定义为从 TIM1_CR1 寄存器所在地址开始的偏移量
	00000: TIM1_CR1

	00001: TIM1_CR2
	00010: TIM1_SMCR
	...
	举例：当 DBL=7 且 DBA=00000 时，表示从 TIM1_CR1 开始向下读取或发送 7 个寄存器的数据。

表 13-6 TIMx DMA 请求表

	TIM1	TIM3	TIM6	TIM14	TIM15	TIM16	TIM17
DMA 请求 0	TIM1_CH1	TIM3_CH1 TIM3_TRIG	TIM6_UP	N/A	TIM15_CH1 TIM15_CH2 TIM15_UP TIM15_TRIG TIM15_COM	TIM16_CH1 TIM16_UP TIM16_TRIG TIM16_COM	TIM17_CH1 TIM17_UP TIM17_TRIG TIM17_COM
DMA 请求 1	TIM1_CH2	TIM3_CH2 TIM3_CH3	N/A	N/A	N/A	N/A	N/A
DMA 请求 2	TIM1_CH3 TIM1_UP	TIM3_CH4 TIM3_UP	N/A	N/A	N/A	N/A	N/A
DMA 请求 3	TIM1_CH4 TIM1_TRIG TIM1_COM	N/A	N/A	N/A	N/A	N/A	N/A

注：TIM6 仅支持 DMA single 传输。

13.10.21 DMAR DMA 传输寄存器 (TIM1_DMAR)

地址偏移：0x4C

复位值：0x0000_0000

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

Bits 31:0	DMAB[31:0]: DMA 传输寄存器
	对 TIM1_DMAR 寄存器的读或写会导致对以下地址的寄存器的读写操作：
	TIM1_CR1 地址 + (DBA + DMA 指针) × 4，其中“TIM1_CR1 地址”是 TIM1 寄存器的起始地址；DBA 是 TIM1_DCR 寄存器中定义的基地址；DMA 指针是由 DMA 自动控制的偏移量，它的范围是从 0 至 TIM1_DCR 寄存器中定义的 DBL。

13.10.22 CCTR1 比较修调寄存器 1 (TIM1_CCTR1)

地址偏移：0x64

复位值：0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCTR1[15:0]															
rw															

Bits	31:16	保留，必须保持复位值
Bits	15:0	CCTR1[15:0] : 比较 1 修调值
		通道 CC1 作为输出时，如果刹车触发且对应的 AOE 位使能，TIM1_CCMR1 寄存器中的 OC1TE=1，当更新事件 UEV 发生时，TIM1_CCR1 的影子寄存器被 TIM1_CCTR1 的值更新；如果 TIM1_CCMR1 寄存器中的 OC1TE=1 且 OC1TUE=1，当更新事件 UEV 发生时，TIM1_CCR1 的影子寄存器和预装载寄存器都被 TIM1_CCTR1 的值更新。

13.10.23 CCTR2 比较修调寄存器 2 (TIM1_CCTR2)

地址偏移: 0x68

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCTR2[15:0]															
rw															

Bits	31:16	保留，必须保持复位值
Bits	15:0	CCTR2[15:0] : 比较 2 修调值
		通道 CC2 作为输出时，如果刹车触发且对应的 AOE 位使能，TIM1_CCMR1 寄存器中的 OC2TE=1，当更新事件 UEV 发生时，TIM1_CCR2 的影子寄存器被 TIM1_CCTR2 的值更新；如果 TIM1_CCMR1 寄存器中的 OC2TE=1 且 OC2TUE=1，当更新事件 UEV 发生时，TIM1_CCR2 的影子寄存器和预装载寄存器都被 TIM1_CCTR2 的值更新。

13.10.24 CCTR3 比较修调寄存器 3 (TIM1_CCTR3)

地址偏移: 0x6C

复位值: 0x0000_0000

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



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCTR3[15:0]															
rw															

Bits 31:16	保留，必须保持复位值
Bits 15:0	CCTR3[15:0]: 比较 3 修调值
	通道 CC3 作为输出时，如果刹车触发且对应的 AOE 位使能，TIM1_CCMR2 寄存器中的 OC3TE=1，当更新事件 UEV 发生时，TIM1_CCR3 的影子寄存器被 TIM1_CCTR3 的值更新；如果 TIM1_CCMR2 寄存器中的 OC3TE=1 且 OC3TUE=1，当更新事件 UEV 发生时，TIM1_CCR3 的影子寄存器和预装载寄存器都被 TIM1_CCTR3 的值更新。

13.10.25 CCTR4 比较修调寄存器 4 (TIM1_CCTR4)

地址偏移: 0x70

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCTR4[15:0]															
rw															

Bits 31:16	保留，必须保持复位值
Bits 15:0	CCTR4[15:0]: 比较 4 修调值
	通道 CC4 作为输出时，如果刹车触发且对应的 AOE 位使能，TIM1_CCMR2 寄存器中的 OC4TE=1，当更新事件 UEV 发生时，TIM1_CCR4 的影子寄存器被 TIM1_CCTR4 的值更新；如果 TIM1_CCMR2 寄存器中的 OC4TE=1 且 OC4TUE=1，当更新事件 UEV 发生时，TIM1_CCR4 的影子寄存器和预装载寄存器都被 TIM1_CCTR4 的值更新。

13.10.26 SysCtrl_EDU_CFG1 寄存器

具体请参考 6.3.2 互联配置寄存器 1 (SysCtrl_EDU_CFG1) 的内容。

13.10.27 SysCtrl_EDU_CFG4 寄存器

具体请参考 6.3.5 互联配置寄存器 4 (SysCtrl_EDU_CFG4) 的内容。

14.通用定时器 TIM3

LCM32F037 内置有 5 个可同步的通用定时器，具体差异参见上面第 13 章高级定时器 TIM1 的表 13-1。各种定时器的特性区别主要在计数方向、是否具有 DMA 请求、捕获/比较通道数量以及互补输出通道数量上。

14.1 TIM3 主要特性

通用定时器 TIM3 是一个 16 位的定时器/计数器，由一个可编程的预分频器驱动，有四路不同的捕获/比较通道。主要用于基本定时，测量输入信号的脉冲宽度（输入捕获）和产生输出波形（输出比较，PWM 和单脉冲模式），对应于不同事件（捕获、比较、溢出、触发）的中断以及与其它定时器或外部信号（外部时钟、复位、触发和使能信号）同步。

TIM3 的 4 个独立的通道可以被分别用于：

- 输入捕获
- 输出比较
- PWM生成（边沿或中心对齐模式）
- 单脉冲模式输出

TIM3 没有互补输出。如果配置为 16 位 PWM 生成器，TIM3 具有全调制能力（0 到 100%）。在调试模式下，计数器可以被冻结。TIM3 可以通过定时器联动机制与其他定时器共同工作，提供同步或事件联接功能。

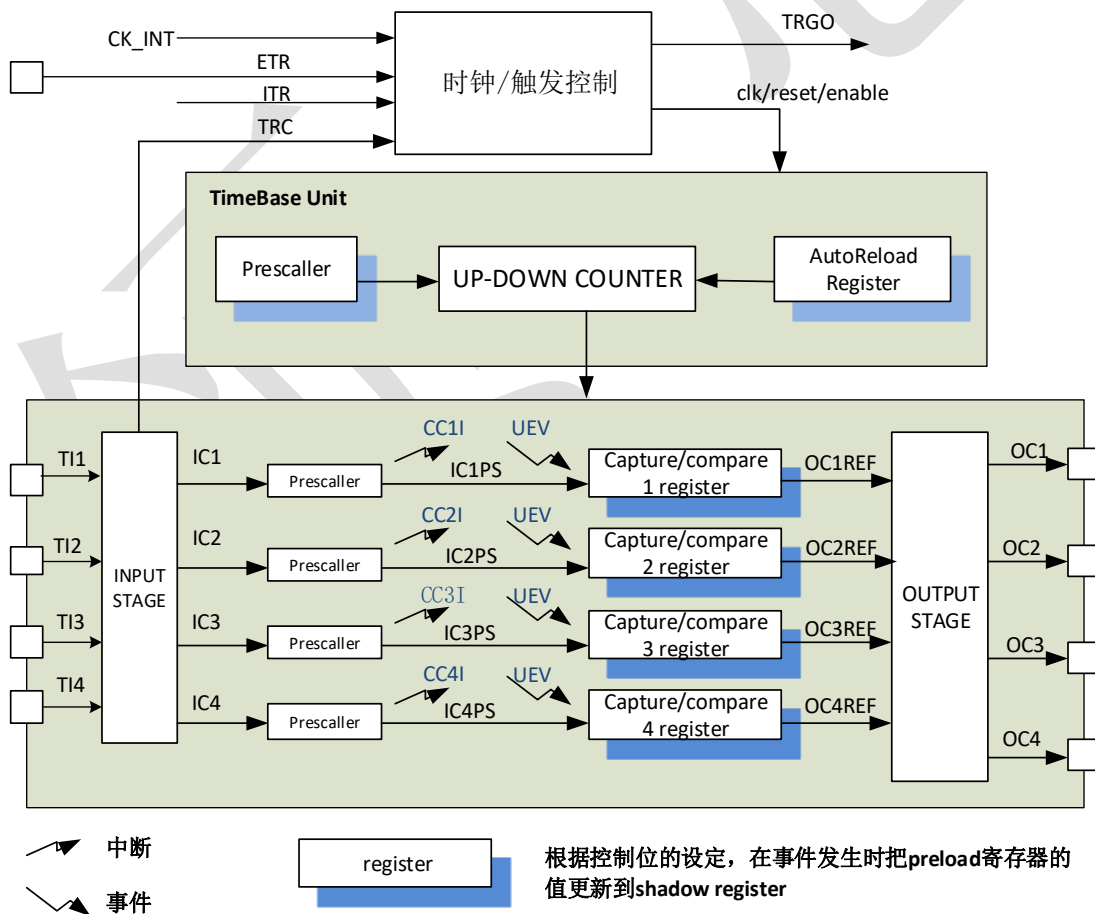


图 14-1 通用定时器 TIM3 框图

14.2 时基单元

时基单元包含：

- 16 位向上/向下计数器 (TIM3_CNT)
- 16 位预分频器 (TIM3_PSC)
- 16 位自动重载寄存器 (TIM3_ARR)

16 位计数器、预分频器和自动重载寄存器都可以通过软件进行读写操作。计数器由预分频器的输出 CK_CNT 驱动，而 CK_CNT 仅在 TIM3_CR1 寄存器的计数器使能位 (CEN) 被置位时才有效。

注：在使能了 CEN 位的一个时钟周期后，计数器才开始计数。

写计数器的操作没有缓存，可以在任何时候写 TIM3_CNT 寄存器，因此建议不要在计数器运行时写入新的数值，以免写入了错误的数值。

14.2.1 自动装载寄存器

自动装载寄存器是预先装载的。写或读自动重载寄存器将访问预装载寄存器。根据在 TIM3_CR1 寄存器中的自动装载预装载使能位 (ARPE) 的设置，预装载寄存器的内容被立即或在每次的更新事件 UEV 时传送到影子寄存器。当计数器达到溢出条件 (例如向下计数时的下溢条件) 并当 TIM3_CR1 寄存器中的 UDIS=0 时，产生更新事件。更新事件也可以由软件产生，随后会详细描述每一种配置下更新事件的产生。注意这里影子寄存器 (Shadow register) 即为有效的工作寄存器 (Active register)。

写自动重载寄存器的两种模式：

- 自动预装载已使能 (TIM3_CR1 寄存器的 ARPE 位置位)。在此模式下，写入自动重载寄存器的数据将被保存在预装载寄存器中，并在下一个更新事件 (UEV) 时传送到影子寄存器。
- 自动预装载已禁止 (TIM3_CR1 寄存器的 ARPE 位清除)。在此模式下，写入自动重载寄存器的数据将立即写入影子寄存器。

更新事件的产生条件：

- 计数器向上或向下溢出
- 软件置位了 TIM3_EGR 寄存器的 UG 位
- 时钟/触发控制器产生硬件复位

在预装载使能时 (ARPE=1)，如果发生了更新事件，预装载寄存器中的数值 (TIM3_ARR) 将写入影子寄存器中。置位 TIM3_CR1 寄存器的 UDIS 位将禁止更新事件 (UEV)。

14.2.2 预分频器

TIM3 的预分频器基于一个由 16 位寄存器 (TIM3_PSC) 控制的 16 位计数器。由于这个控制寄存器带有缓冲器，因此它能够在运行时被改变。预分频器可以将计数器的时钟频率按 1 到 65536 之间的任意值分频。计数器的频率可以由下式计算：

$$f_{CK_CNT} = f_{CK_PSC} / (PSC[15:0] + 1)$$

预分频器的值由预装载寄存器写入，新的预分频器的值在下次更新事件到来时被采用。对 TIM3_PSC 寄存器的读操作通过预装载寄存器完成，因此不需要特别的关注。



下图给出了在预分频器工作时，更改其参数的情况下计数器操作的例子，其它分频系数类推。

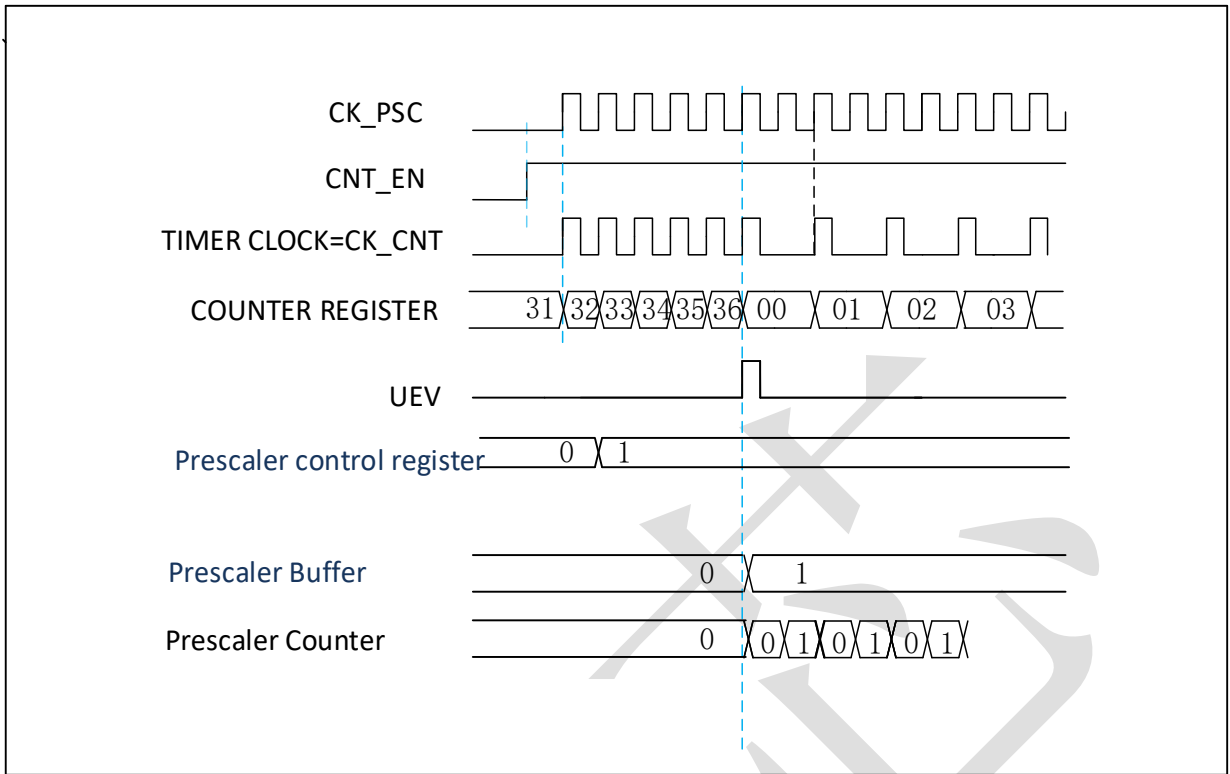


图 14-2 预分频系数从 1 变 2 的计数器时序

14.3 计数器模式

14.3.1 向上计数模式

在向上计数模式中，计数器从 0 计数到用户定义的比较值（TIMx_ARR 寄存器的值），然后重新从 0 开始计数并产生一个计数器溢出事件，同时，如果 TIMx_CR1 寄存器的 UDIS 位是 0，将会产生一个更新事件(UEV)。

置位 TIMx_EGR 寄存器的 UG 位(通过软件方式或者使用从模式控制器)也同样可以产生一个更新事件。

使用软件置位 TIMx_CR1 寄存器的 UDIS 位，可以禁止更新事件，这样可以避免在更新预装载寄存器时更新影子寄存器。在 UDIS 位被清除之前，将不产生更新事件。但是在应该产生更新事件时，计数器仍会被清 0，同时预分频器的计数也被清 0（但预分频器的数值不变）。此外，如果设置了 TIMx_CR1 寄存器中的 URS 位（选择更新请求源），设置 UG 位将产生一个更新事件 UEV，但硬件不设置 UIF 标志（即不产生中断请求）。这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。

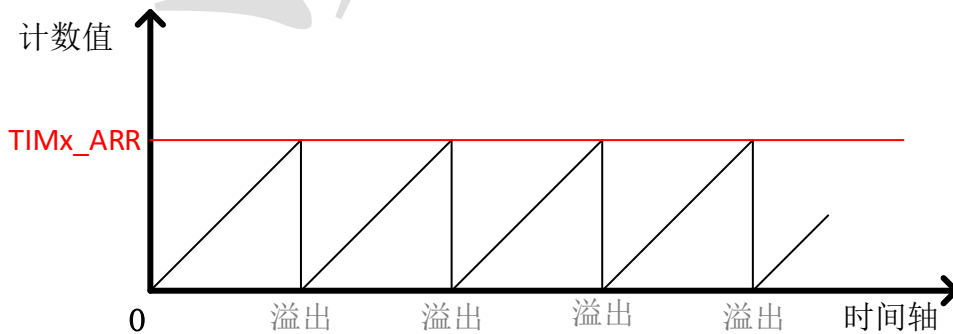


图 14-3 向上计数溢出

当发生一个更新事件时，所有的寄存器都被更新，硬件同时依据 URS 位设置更新标志位（TIMx_SR 寄存器的 UIF 位）：自动装载影子寄存器被重新置入预装载寄存器的值（TIMx_ARR）。预分频器的缓存器被置入预装载寄存器的值（TIMx_PSC）。要注意到如果是因为计数器溢出而产生更新，自动重装载寄存器将在计数器重载入之前被更新，因此下一个计数周期才是预期的值（计数器被装载为新的值）。非自动重装载模式下可以立即更新。

下图给出例子，当 TIMx_ARR=0x36 时计数器在二分频时钟频率下的动作。

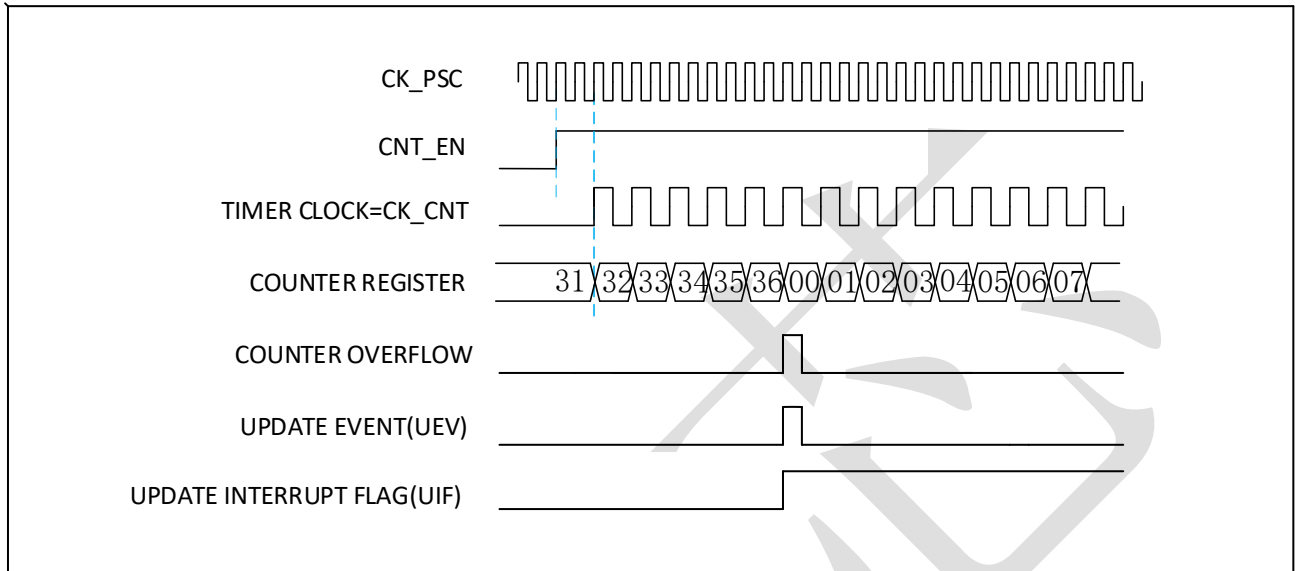


图 14-4 计数器在二分频时钟频率下的时序图

14.3.2 向下计数模式

在向下计数模式中，计数器从自动装载的值（TIMx_ARR 寄存器的值）开始向下计数到 0，然后再从自动装载的值重新开始计数，并产生一个计数器向下溢出事件。如果 TIMx_CR1 寄存器的 UDIS 位是 0，将会产生一个更新事件（UEV）。

置位 TIMx_EGR 寄存器的 UG 位(通过软件方式或者使用从模式控制器)也同样可以产生一个更新事件。

置位 TIMx_CR1 寄存器的 UDIS 位可以禁止 UEV 事件。这样可以避免在更新预装载寄存器时更新影子寄存器。因此 UDIS 位清除之前不会产生更新事件。然而，计数器仍会从当前自动加载值重新开始计数，并且预分频器的计数器重新从 0 开始（但预分频器的数值不变）。此外，如果设置了 TIMx_CR1 寄存器中的 URS 位（选择更新请求源），设置 UG 位将产生一个更新事件 UEV，但硬件不设置 UIF 标志（即不产生中断请求）。这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。

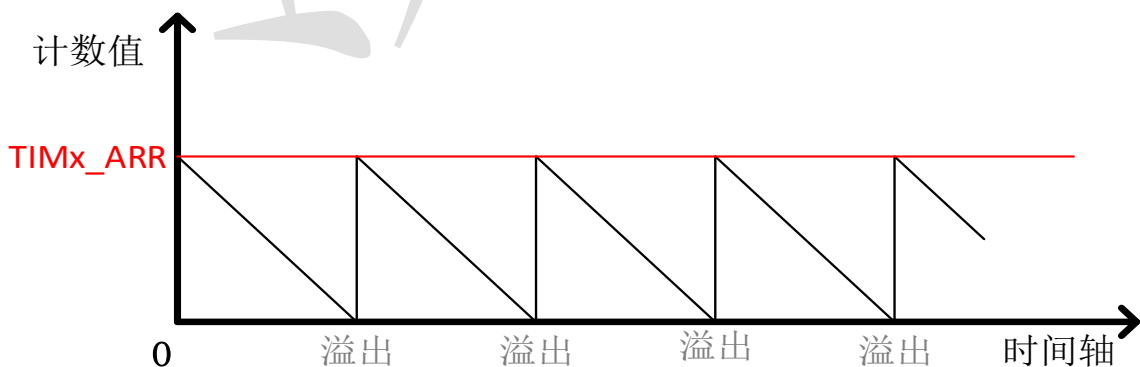


图 14-5 向下计数溢出

当发生一个更新事件时，所有的寄存器都被更新，硬件同时依据 **URS** 位设置更新标志位 (**TIMx_SR** 寄存器的 **UIF** 位)：自动装载影子寄存器被重新置入预装载寄存器的值 (**TIMx_ARR**)。预分频器的缓存器被置入预装载寄存器的值 (**TIMx_PSC**)。要注意到如果是因为计数器溢出而产生更新，自动重装载寄存器将在计数器重载入之前被更新，因此下一个计数周期才是预期的值 (计数器被装载为新的值)。非自动重装载模式下可以立即更新。

14.3.3 中央对齐模式 (向上/向下计数)

在中央对齐模式中，计数器从 0 开始计数到自动加载的值 (**TIMx_ARR-1**)，产生一个计数器溢出事件，然后向下计数到 1 并且产生一个计数器下溢事件；然后再从 0 开始重新计数。

在此模式下，不能写入 **TIMx_CR1** 中的 **DIR** 方向位。它由硬件更新并指示当前的计数方向。

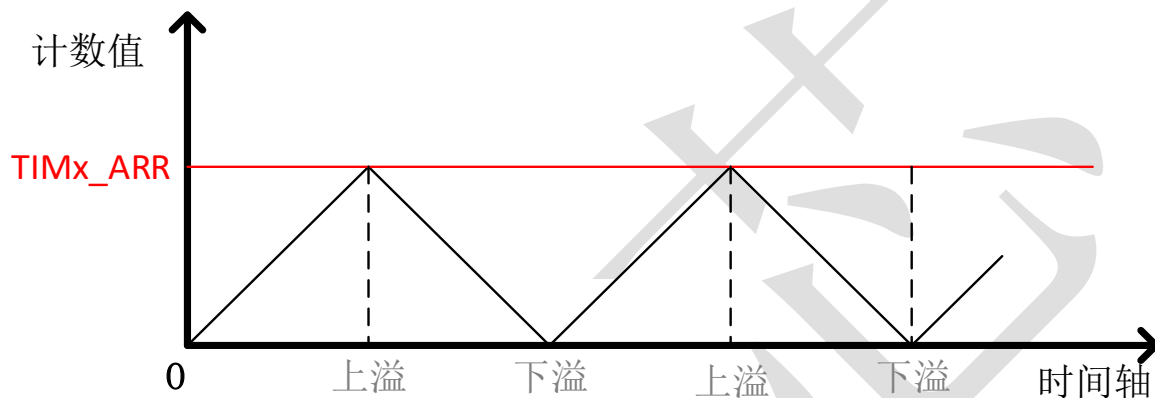


图 14-6 中央对齐溢出

置位 **TIMx_EGR** 寄存器的 **UG** 位(通过软件方式或者使用从模式控制器)也同样可以产生一个更新事件。此时，计数器重新从 0 开始计数，预分频器也重新从 0 开始计数。

设置 **TIMx_CR1** 寄存器中的 **UDIS** 位可以禁止 **UEV** 事件，这样可以避免在更新预装载寄存器时更新影子寄存器。因此 **UDIS** 位被清为 0 之前不会产生更新事件。然而，计数器仍会根据当前自动重加载的值，继续向上或向下计数。

此外，如果设置了 **TIMx_CR1** 寄存器中的 **URS** 位 (选择更新请求源)，设置 **UG** 位将产生一个更新事件 **UEV**，但硬件不设置 **UIF** 标志 (即不产生中断请求)。这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。

当发生一个更新事件时，所有的寄存器都被更新，硬件同时依据 **URS** 位设置更新标志位 (**TIMx_SR** 寄存器的 **UIF** 位)：自动装载影子寄存器被重新置入预装载寄存器的值 (**TIMx_ARR**)。预分频器的缓存器被置入预装载寄存器的值 (**TIMx_PSC**)。要注意到如果是因为计数器溢出而产生更新，自动重装载寄存器将在计数器重载入之前被更新，因此下一个计数周期才是预期的值 (计数器被装载为新的值)。

使用中央对齐模式：

- 启动中央对齐模式时，计数器将按照原有的向上/向下的配置计数。也就是说 **TIMx_CR1** 寄存器中的 **DIR** 位将决定计数器是向上还是向下计数。此外，软件不能同时修改 **DIR** 位和 **CMS** 位的值。
- 不推荐在中央对齐模式下，计数器正在计数时写计数器的值，这将导致不能预料后果。具体的说：
 - 向计数器写入了比自动装载值更大的数值时 ($TIMx_CNT > TIMx_ARR$)，但计数器的计数方向不发生改变。例如计数器已经向上溢出，但计数器仍然向上计数。
 - 向计数器写入了 0 或者 **TIMx_ARR** 的值，但更新事件不发生。
- 安全使用中央对齐模式的计数器的方法是在启动计数器之前先用软件置位 **TIMx_EGR** 寄存器的 **UG** 位产生一个更新事件，并且不在计数器计数时修改计数器的值。

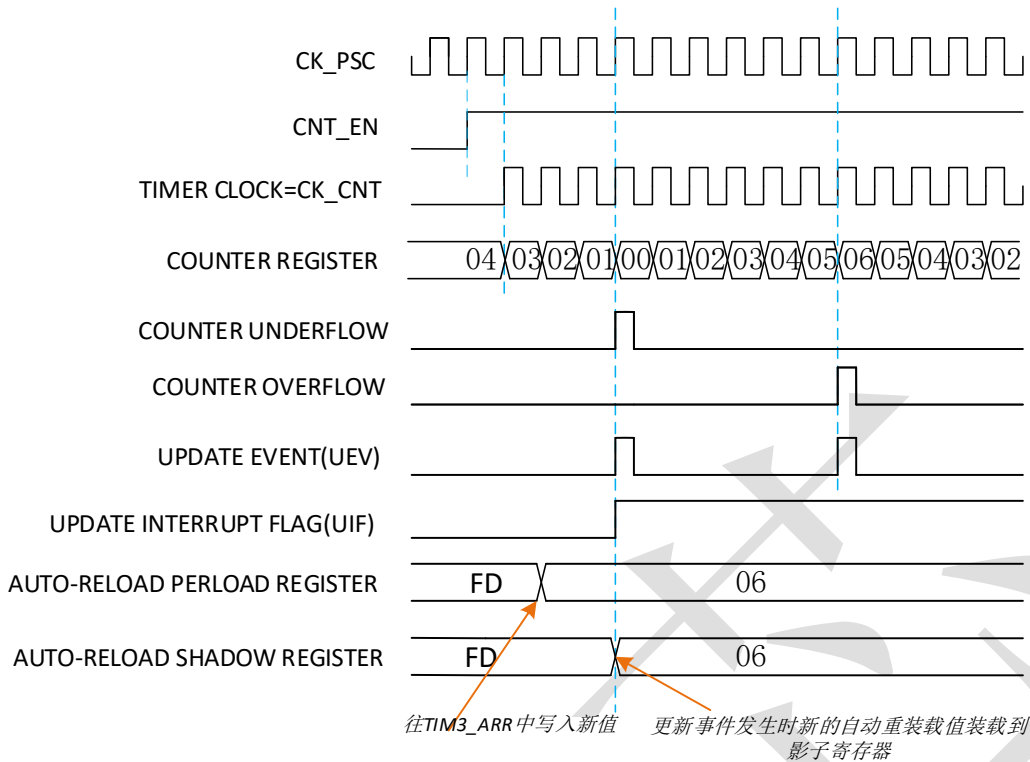


图 14-7 计数器时序图，内部时钟分频为 1，TIMx_ARR=0x6

14.4 时钟选择

时钟/触发控制器允许用户选择计数器的时钟源(CK_PSC)、触发输入信号(TRGI)和触发输出信号(TRGO)，其框图如图 14-8 所示。

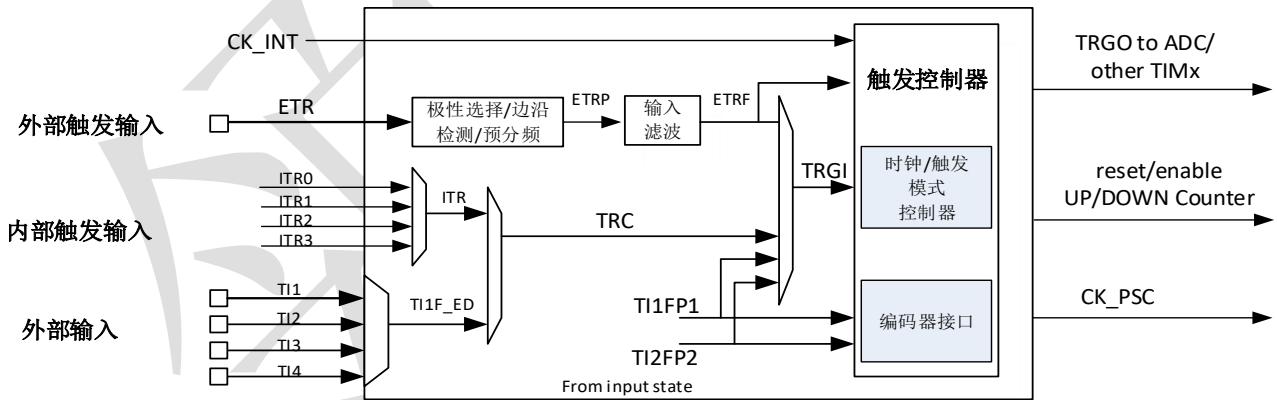


图 14-8 时钟/触发控制器框图

时基单元的预分频时钟 (CK_PSC) 可以有以下来源：

- 内部时钟 (CK_INT)：芯片内部时钟
- 外部时钟模式 1：外部触发输入 (Tii) 和内部触发输入 (ITRx)
- 外部时钟模式 2：外部触发输入 ETR

14.4.1 内部时钟源 (CK_INT)

如果 TIMx_SMCR 寄存器 SMS=000，TIMx_SMCR 寄存器 ECE=0，则 CEN、DIR 和 UG 位是实际上的控制位，并且只能被软件修改 (UG 位仍被自动清除)。一旦 CEN 位被写成 1，预分频器的时钟就由内部时钟提供。

14.4.2 外部时钟模式 1

当 TIMx_SMCR 寄存器的 SMS=111 时，此模式被选中。计数器可以在选定输入端的每个上升沿或下降沿计数。

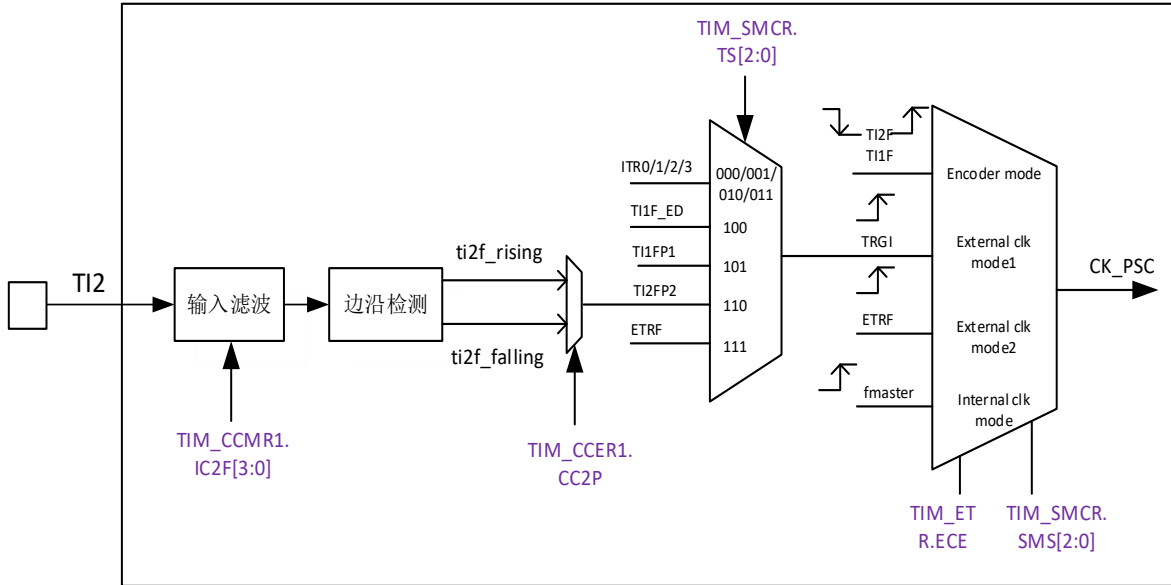


图 14-9 TI2 外部时钟框图

例如，要配置向上计数器在 TI2 输入端的上升沿计数，使用下列步骤：

1. 配置 TIMx_CCMR1 寄存器的 CC2S=01，使用通道 2 检测 TI2 输入的上升沿
2. 配置 TIMx_CCMR1 寄存器的 IC2F[3:0] 位，选择输入滤波器带宽（如果不需要滤波器，保持 IC2F=0000）
3. 配置 TIMx_CCER 寄存器的 CC2P=0，选定上升沿极性
4. 配置 TIMx_SMCR 寄存器的 SMS=111，配置计数器使用外部时钟模式 1
5. 配置 TIMx_SMCR 寄存器的 TS=110，选定 TI2 作为输入源
6. 设置 TIMx_CR1 寄存器的 CEN=1，启动计数器

当上升沿出现在 TI2，计数器计数一次，且触发标识位（TIM3_SR 寄存器的 TIF 位）被置 1，如果使能了中断（在 TIMx_DIER 寄存器中配置）则会产生中断请求。

在 TI2 的上升沿和计数器实际时钟之间的延时取决于在 TI2 输入端的重新同步电路（下图预分频系数为 1）。

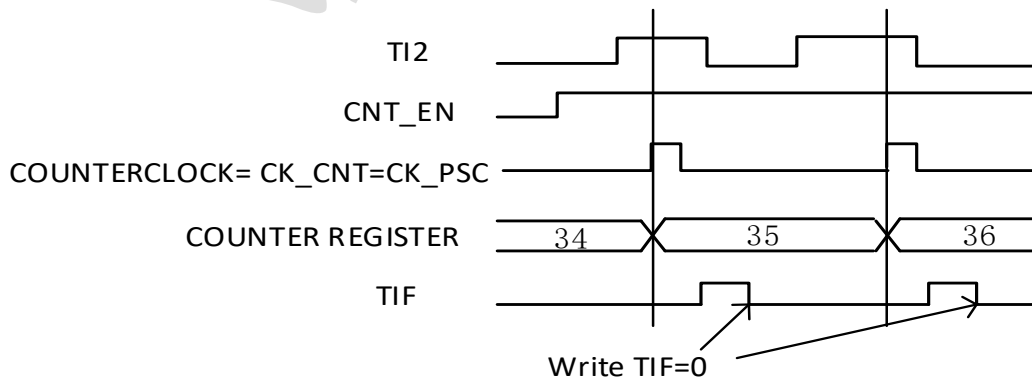


图 14-10 外部时钟模式 1 下的控制电路

14.4.3 外部时钟模式 2

计数器能够在外部触发输入 ETR 信号的每一个上升沿或下降沿计数。将 TIMx_SMCR 寄存器的 ECE 位写 1，即可选定此模式。

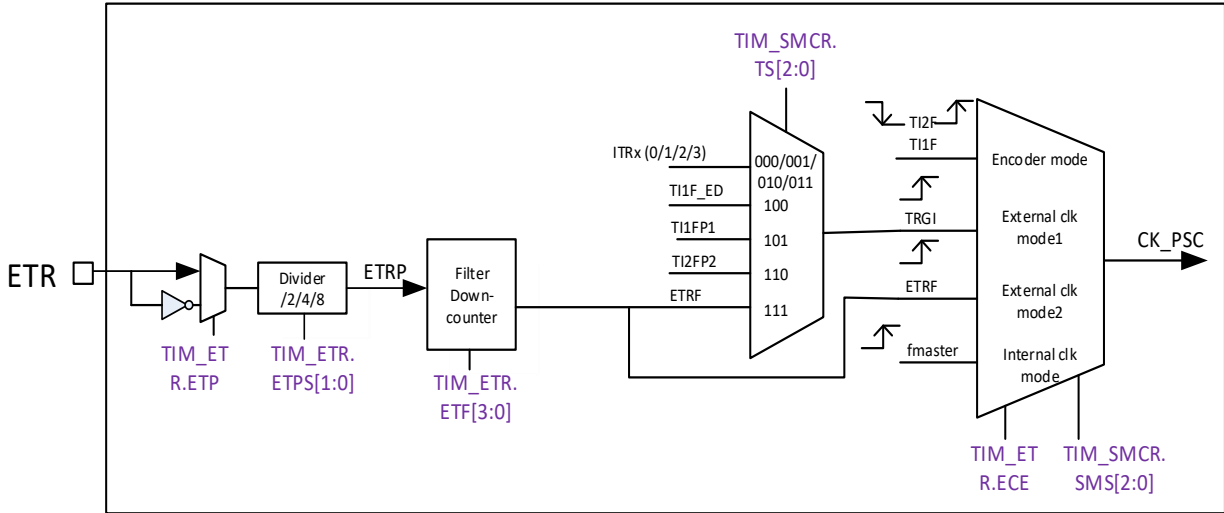


图 14-11 外部触发时钟框图

例如，要配置计数器在 ETR 信号的每 2 个上升沿时向上计数一次，需使用下列步骤：

1. 本例中不需要滤波器，配置 TIMx_SMCR 寄存器的 ETF[3:0]=0000
2. 设置预分频器，配置 TIMx_SMCR 寄存器的 ETPS[1:0]=01
3. 选择 ETR 的上升沿检测，配置 TIMx_SMCR 寄存器的 ETP=0
4. 开启外部时钟模式 2，配置 TIMx_SMCR 寄存器中的 ECE=1
5. 启动计数器，写 TIMx_CR1 寄存器的 CEN=1

计数器在每 2 个 ETR 上升沿计数一次。在 ETR 的上升沿和计数器实际时钟之间的延时取决于在 ETRP 信号端的重新同步电路。

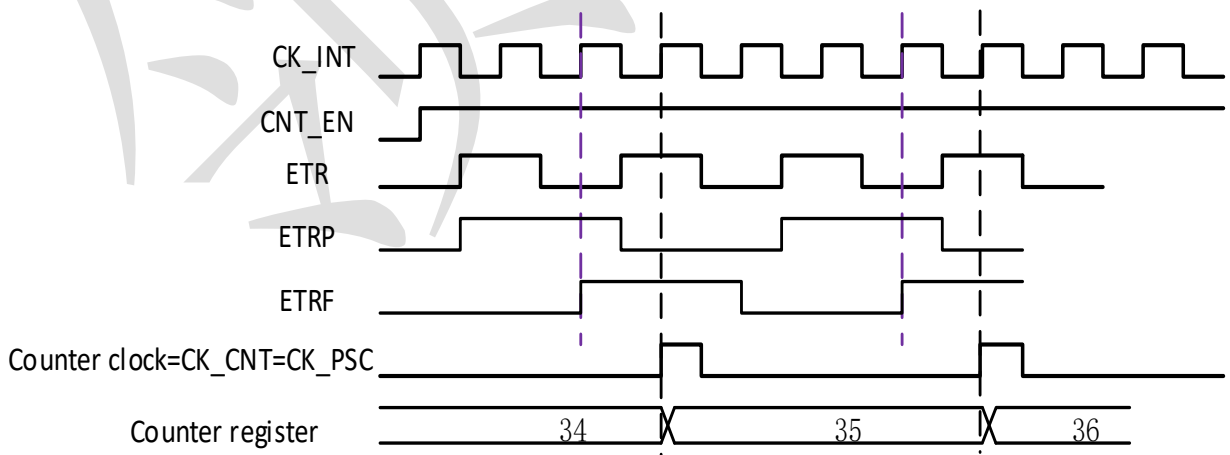


图 14-12 外部时钟模式 2 下的控制电路

14.5 捕获/比较通道

定时器的 I/O 引脚 (TIMx_CCI) 可以用作输入捕获或者输出比较, 这个功能可以通过配置捕获/比较通道模式寄存器 (TIMx_CCMRi) 的 CCIS 通道选择位来实现, 此处的 i 代表通道数。

每一个捕获/比较通道都是围绕着一个捕获/比较寄存器 (包含影子寄存器) 来构建的, 包括捕获的输入部分 (数字滤波、多路复用和预分频器), 和输出部分 (比较器和输出控制)。

捕获/比较模块由一个预装载寄存器和一个影子寄存器组成, 读写过程仅操作预装载寄存器。在捕获模式下, 捕获计数器的值就在影子寄存器上, 然后再复制到预装载寄存器中。在比较模式下, 预装载寄存器的内容被复制到影子寄存器中, 然后影子寄存器内容和计数器进行比较。

当通道被配置成输出模式时 (TIMx_CCMRi 寄存器的 CCIS=0), 可以随时访问 TIMx_CCRi 寄存器 (此处的 i 指通道数)。

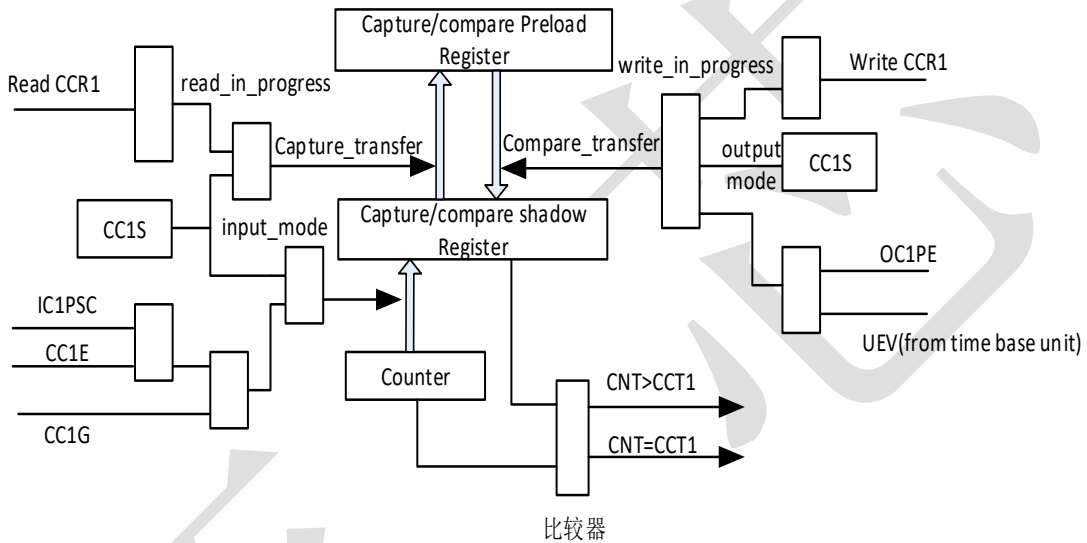


图 14-13 捕获/比较通道 1 的主电路

当通道被配置成输入模式时, 对 TIMx_CCRi 寄存器的读操作类似于计数器的读操作。当捕获发生时, 计数器的内容被捕获到 TIMx_CCRi 影子寄存器, 随后再复制到预装载寄存器中。在读操作进行中, 预装载寄存器是被冻结的 (以防读数据过程中预装载寄存器内容被修改)。

14.5.1 输入模块

输入部分对相应的 Ti_i 输入信号采样, 并产生一个滤波后的信号 Ti_iF 。然后, 一个带极性选择的边沿检测器产生一个信号 (Ti_iFPx), 它可以作为触发模式控制器的输入触发或者作为捕获控制。该信号通过预分频进入捕获寄存器 (ICIPS)。

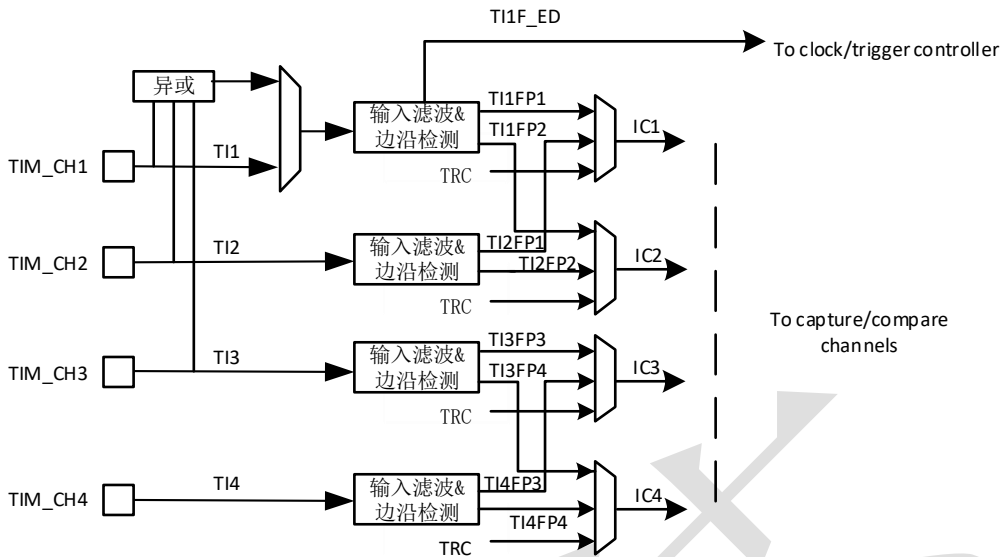


图 14-14 输入模块框图

14.5.2 输入捕获模式

在输入捕获模式下，当检测到 IC_i 信号上相应的边沿后，计数器的当前值被锁存到捕获/比较寄存器（TIM_x_CCR_i）中。当发生捕获事件时，相应的 CCI_iF 标志（TIM_x_SR 寄存器）被置 1。

如果 TIM_x_DIER 寄存器的 CCI_iE 位被置位，也就是使能了中断，则将产生中断请求。如果发生捕获事件时 CCI_iF 标志已经为高，那么重复捕获标志 CCI_iOF（TIM_x_SR 寄存器）被置 1，写 CCI_iF=0 可清除 CCI_iOF。写 CCI_iF=0 或读取存储在 TIM_x_CCR_i 寄存器中的捕获数据都可清除 CCI_iF。

TIM3 通道 1 的输入如下图所示：

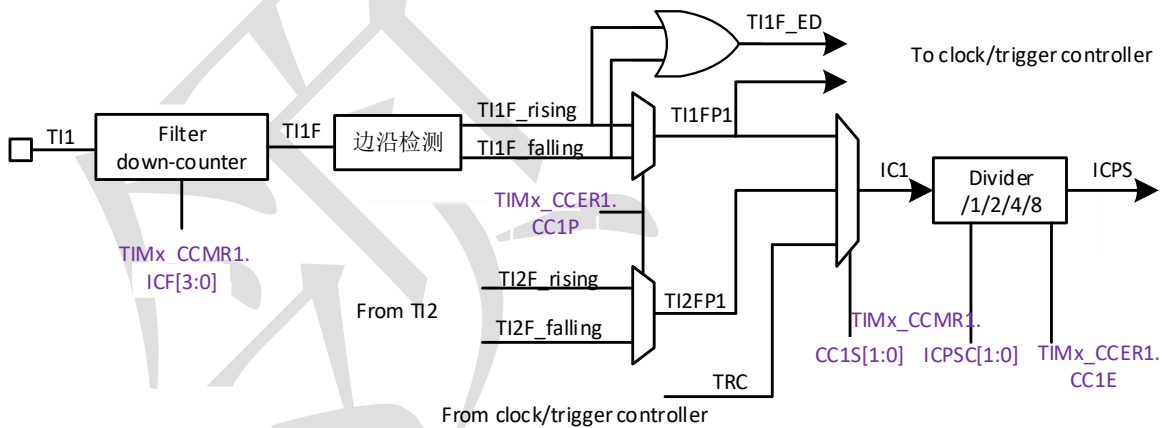


图 14-15 TIM3 通道 1 的输入

以下例子说明如何在 TI1 输入的上升沿时捕获计数器的值到 TIM_x_CCR1 寄存器中，步骤如下：

1. 选择有效输入端：例如 TIM_x_CCR1 连接到 TI1 输入，所将 01 写入 TIM_x_CCMR1 寄存器中的 CC1S，此时通道被配置为输入，并且 TIM_x_CCR1 寄存器变为只读。
2. 根据输入信号 TI_i 的特点，可通过配置 TIM_x_CCMR_i 寄存器中的 IC_iF 位来设置相应的输入滤波器的滤波时间。假设输入信号在最多 5 个时钟周期的时间内抖动，我们须配置滤波器的带宽长于 5 个时钟周期；因此我们可以连续采样 8 次，以确认在 TI1 上一次真实的边沿变换，即在 TIM_x_CCMR1 寄存器中写入 IC1F=0011，此时，只有连续采样到 8 个相同的 TI1 信号，信号才为有效（采样频率

为 DTS 时钟的频率 f_{DTS})。

3. 选择 TI1 通道的有效转换边沿，在 TIMx_CCER 寄存器中写入 CC1P=0（上升沿）。
4. 配置输入预分频器。在本例中，我们希望捕获发生在每一个有效的电平转换时刻，因此预分频器被禁止（写 TIMx_CCMR1 寄存器的 IC1PSC=00）。
5. 设置 TIMx_CCER 寄存器的 CC1E=1，允许捕获计数器的值到捕获寄存器中。
6. 如果需要，通过设置 TIMx_DIER 寄存器中的 CC1IE 位允许相关中断请求。

当发生一个输入捕获时：

- 当产生有效的电平转换时，计数器的值被传送到 TIMx_CCR1 寄存器。
- CC1IF 标志被设置（中断标志）。当发生至少 2 次连续的捕获，且 CC1IF 未被清除，CC1OF 也被置 1。
- 如设置了 CC1IE 位，则会产生一个中断。
- 为了处理捕获溢出（CC1OF 位），建议在读出重复捕获标志之前读取数据，这是为了避免丢失在读出捕获溢出标志之后和读取数据之前可能产生的重复捕获信息。

注：设置 TIM_EGR 寄存器中相应的 CCIg 位，可以通过软件产生输入捕获中断。

14.5.3 PWM 输入

该模式是输入捕获模式的一个特例，除下列区别外，操作与输入捕获模式相同：

- 两个 ICi 信号被映射至同一个 Tii 输入
- 这两个 ICi 信号的有效边沿的极性相反
- 两路 TiiFP 信号中的一路被选择作为触发输入信号，并且触发模式控制器被配置成触发复位模式

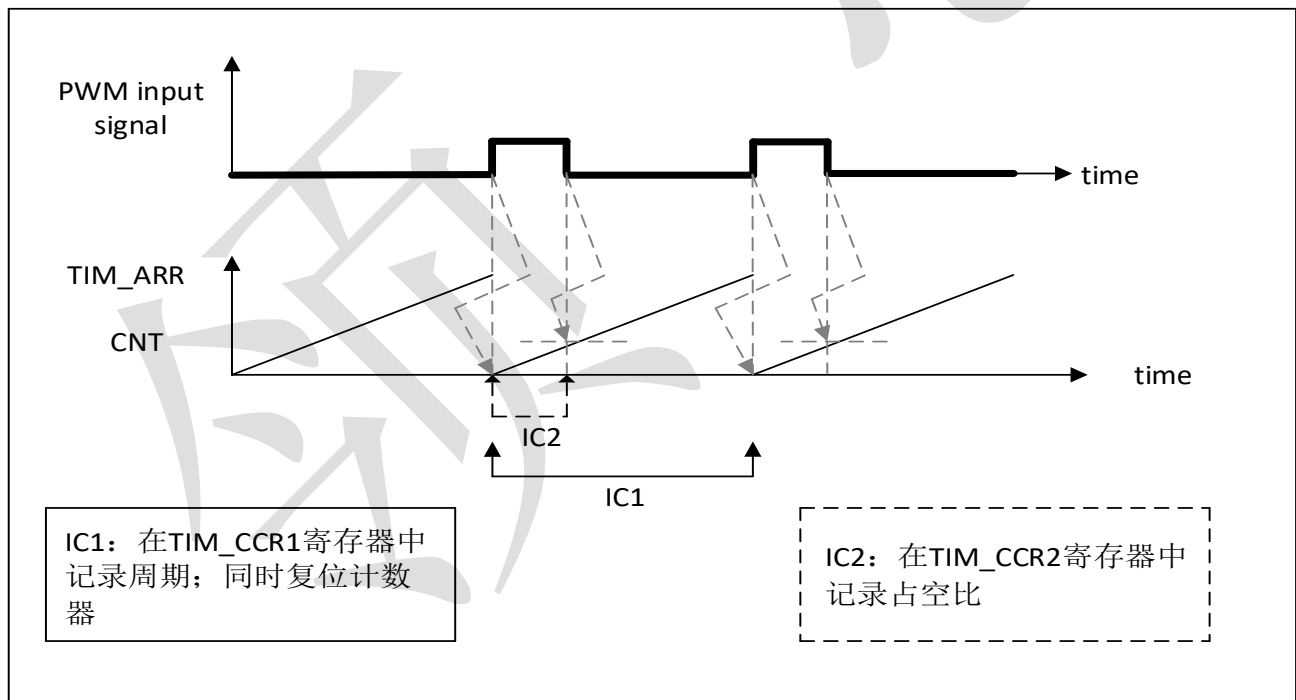


图 14-16 PWM 输入信号测量

例如，你可以用以下方式测量 TI1 上输入的 PWM 信号的周期（锁存于 TIMx_CCR1 寄存器）和占空比（锁存于 TIMx_CCR2 寄存器）。（具体取决于 f_{CK_INT} 的频率 f_{CK_INT} 和预分频器的值）

1. 选择 TIMx_CCR1 的有效输入：置 TIMx_CCMR1 寄存器的 CC1S=01（选中 TI1）。
2. 选择 TI1FP1 的有效极性（用来捕获数据到 TIMx_CCR1 中和清除计数器）：置 CC1P=0（上升沿有效）。

3. 选择 TIMx_CCR2 的有效输入：置 TIMx_CCMR1 寄存器的 CC2S=10（选中 TI1FP2）。
4. 选择 TI1FP2 的有效极性（捕获数据到 TIMx_CCR2）：置 CC2P=1（下降沿有效）。
5. 选择有效的触发输入信号：置 TIMx_SMCR 寄存器中的 TS=101（选择 TI1FP1）。
6. 配置触发模式控制器为复位触发模式：置 TIMx_SMCR 中的 SMS=100。
7. 使能捕获：置 TIMx_CCER 寄存器中 CC1E=1，CC2E=1。

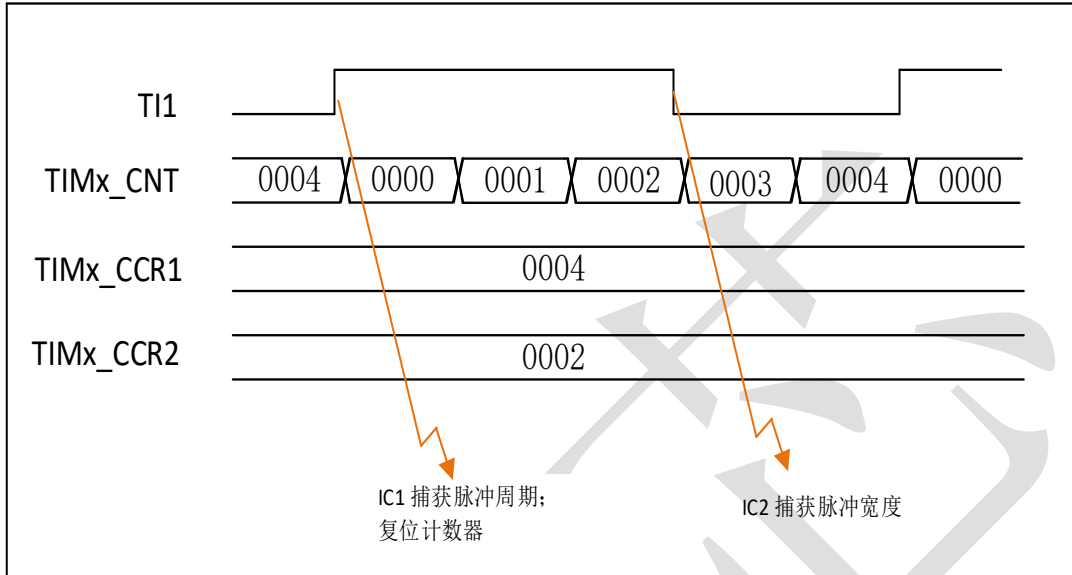


图 14-17 PWM 输入信号测量实例

14.5.4 输出模块

输出模块会产生一个用来做参考的中间波形，称为 OCiREF（高电平有效）。输出极性的处理在模块的最后。TIM3 没有互补输出，没有死区控制和刹车功能。

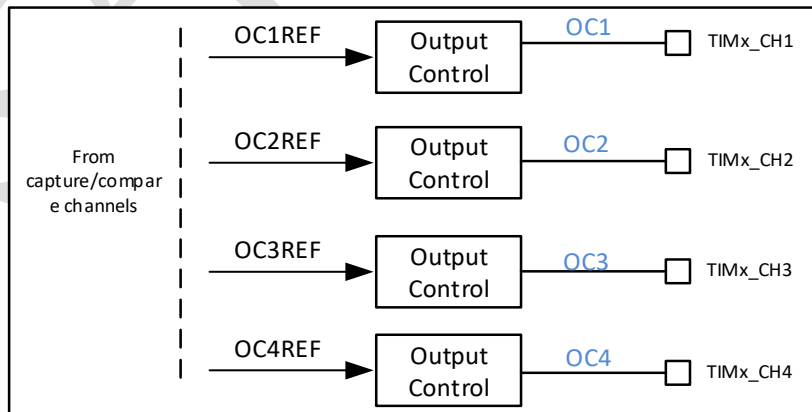


图 14-18 输出模块框图

14.5.5 强制输出模式

在输出模式（TIM3_CCMRi 寄存器中 CCiS=00）下，输出比较信号能够直接由软件强置为高或低状态，而不依赖于输出比较寄存器和计数器间的比较结果。

置 TIMx_CCMRi 寄存器中相应的 OCiM=101，即可强置输出比较信号为有效状态。这样 OCiREF 被强置为高电平（OCiREF 始终为高电平有效），而 OCi 的输出是高还是低则取决于 CCiP 极性标志位。例如 CCiP=0（OCi

高电平有效), 则 OC_i 被强置为高电平。

置 TIM_x_CCMR_i 寄存器的 OC_iM=100, 可强置 OC_iREF 信号为低。

该模式下, 在 TIM_x_CCR_i 影子寄存器和计数器之间的比较仍然在进行, 相应的标志也会被修改, 也会产生相应的中断。这将会在下面的输出比较模式一节中介绍。

14.5.6 输出比较模式

此模式用来控制一个输出波形或者指示一段给定的时间已经达到。当计数器与捕获/比较寄存器的内容相同时, 有如下操作:

- 根据不同的输出比较模式, 相应的 OC_iREF 输出信号为:
 - 保持不变 (OC_iM=000)
 - 设置为有效电平 (OC_iM=001)
 - 设置为无效电平 (OC_iM=010)
 - 翻转 (OC_iM=011)
- 设置中断状态寄存器中的标志位 (TIM_x_SR 寄存器中的 CC_iIF 位)
- 若设置了相应的中断使能位 (TIM_x_DIER 寄存器中的 CC_iIE 位), 则产生一个中断

TIM_x_CCMR_i 寄存器的 OC_iM 位用于选择输出比较模式, 而 TIM_x_CCER 寄存器的 CC_iP 位用于选择有效和无效的电平极性。

TIM_x_CCMR_i 寄存器的 OC_iPE 位用于选择 TIM_x_CCR_i 寄存器是否需要使用预装载寄存器。

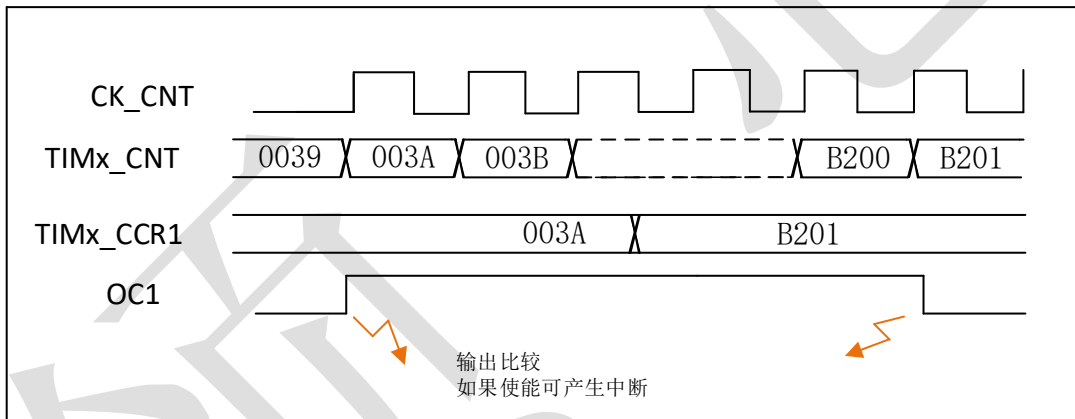


图 14-19 输出比较模式, 翻转 OC1

在输出比较模式下, 更新事件 UEV 对 OC_iREF 和 OC_i 输出没有影响。输出比较的时间精度为计数器的一个时钟周期, 输出比较模式也能用来输出一个单脉冲。输出比较模式的配置步骤如下:

1. 选择计数器时钟 (内部/外部/预分频器)
2. 将相应的数据写入 TIM_x_ARR 和 TIM_x_CCR_i 寄存器中
3. 如果要产生一个中断请求, 设置 CC_iIE 位
4. 选择输出模式步骤:
 - 要求计数器与 CCR_i 匹配时翻转 OC_iM 的输出管脚, 设置 OC_iM=011
 - 置 OC_iPE = 0 禁用预装载寄存器
 - 置 CC_iP = 0 选择高电平为有效电平
 - 置 CC_iE = 1 使能输出
5. 设置 TIM_x_CR1 寄存器的 CEN 位来启动计数器

在未使用预装载寄存器 (OC_iPE=0, 否则 TIM_x_CCR_i 的影子寄存器只能在发生下一次更新事件时被更新)

的情况下，TIMx_CCRi 寄存器能够在任何时候通过软件进行更新以控制输出波形。

14.5.7 PWM 模式

脉冲宽度调制 (PWM) 模式可以产生一个由 TIMx_ARR 寄存器确定频率，由 TIMx_CCRi 寄存器确定占空比的信号。

在 TIMx_CCMRi 寄存器中的 OCiM 位写入 110 (PWM 模式 1) 或 111 (PWM 模式 2)，能够独立地设置每个 OCi 输出通道产生一路 PWM。必须设置 TIMx_CCMRi 寄存器的 OCiPE 位使能相应的预装载寄存器，也可以设置 TIMx_CR1 寄存器的 ARPE 位使能自动重载的预装载寄存器(在向上计数模式或中央对称模式中)。由于仅当发生一个更新事件的时候，预装载寄存器才能被传送到影子寄存器，因此在计数器开始计数之前，必须通过设置 TIMx_EGR 寄存器的 UG 位来初始化所有的寄存器。

OCi 的极性可以通过软件在 TIMx_CCER 寄存器中的 CCiP 位设置，它可以设置为高电平有效或低电平有效。OCi 的输出使能通过 TIMx_CCER 中 CCiE 位控制，详见 TIMx_CCER 寄存器的描述。

在 PWM 模式 (模式 1 或模式 2) 下，TIMx_CNT 和 TIM3_CCRi 始终在进行比较，(依据计数器的计数方向) 以确定是否符合 $TIMx_CCRi \leq TIMx_CNT$ 或者 $TIMx_CNT \leq TIMx_CCRi$ 。根据 TIMx_CR1 寄存器中 CMS 位域的状态，定时器能够产生边沿对齐的 PWM 信号或中央对齐的 PWM 信号。

14.5.7.1 PWM 边沿对齐模式

向上计数配置

当 TIMx_CR1 寄存器中的 DIR 位为低的时候执行向上计数。下面是一个 PWM 模式 1 的例子。当 $TIMx_CNT < TIMx_CCRi$ 时，PWM 参考信号 OCiREF 为高，否则为低。如果 TIMx_CCRi 中的比较值大于自动重载值 (TIMx_ARR)，则 OCiREF 保持为 1。如果比较值为 0，则 OCiREF 保持为 0。下图为 TIMx_ARR=8 时边沿对齐的 PWM 波形实例。

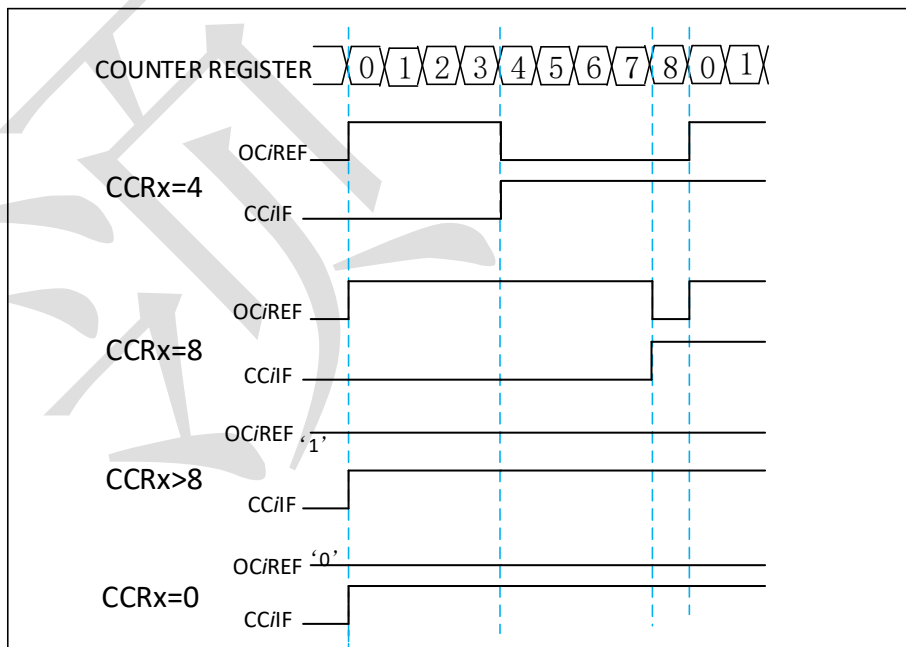


图 14-20 TIMx_ARR=8 时边沿对齐的 PWM 波形实例

向下计数的配置

当 TIMx_CR1 寄存器的 DIR 位为高时执行向下计数。在 PWM 模式 1 时，当 $TIMx_CNT > TIMx_CCRi$ 时参考信号 OCiREF 为低，否则为高。如果 TIM3_CCRi 中的比较值大于 TIMx_ARR 中的自动重载值，则 OCiREF 保

持为 1。该模式下不能产生 0% 的 PWM 波形。

14.5.7.2 PWM 中央对齐模式

当 TIMx_CR1 寄存器中的 CMS 位不为 00 时为中央对齐模式（所有其它的配置对 OCiREF/OCi 信号都有相同的作用）。

根据不同的 CMS 位的设置，比较标志可以在计数器向上计数，向下计数，或向上和向下计数时被置 1。TIMx_CR1 寄存器中的计数方向位（DIR）由硬件更新，不要用软件修改它。

下图给出了一些中央对齐的 PWM 波形的例子：

- TIMx_ARR=8
- PWM 模式 1
- 标志位在以下三种情况下被置位（以箭头形式在图 14-21 中标出）

只有在计数器向下计数时（CMS=01）

只有在计数器向上计数时（CMS=10）

在计数器向上和向下计数时（CMS=11）

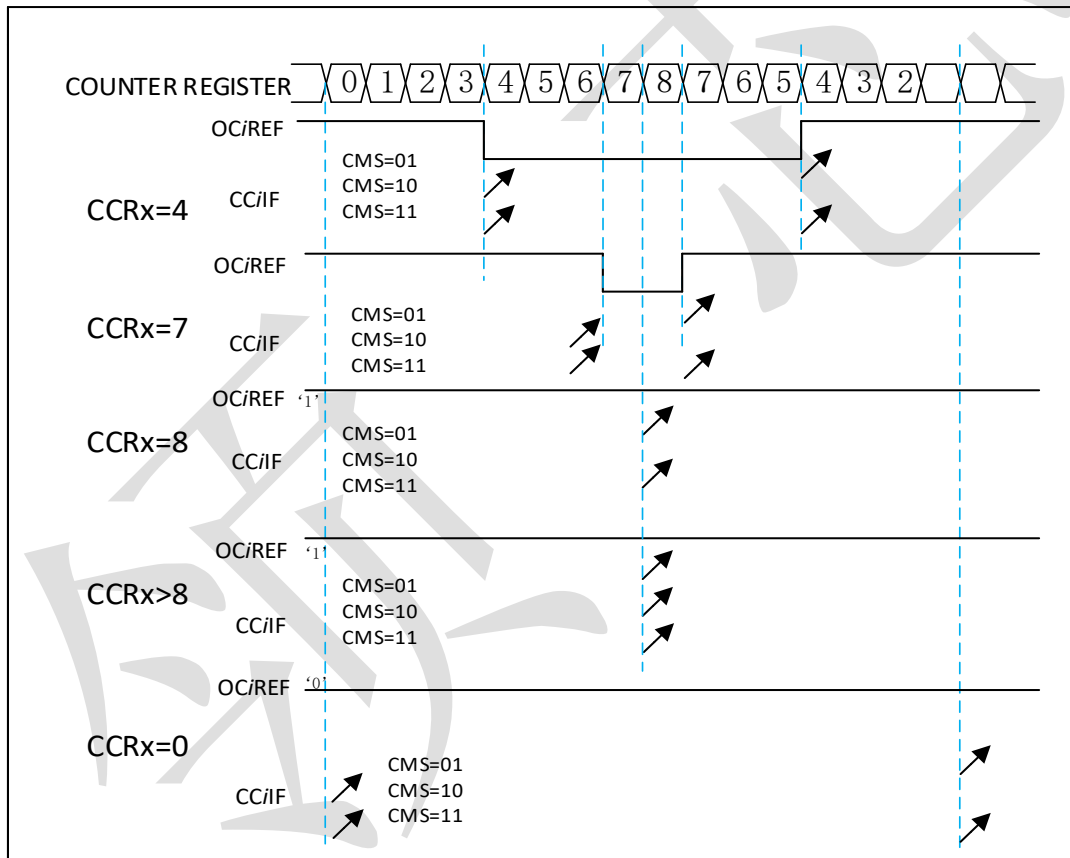


图 14-21 中央对齐模式 PWM (ARR=8)

14.5.8 单脉冲模式

单脉冲模式（OPM）是上述众多模式的一个特例。这种模式允许计数器响应一个激励，并在一个程序可控的延时之后产生一个脉宽可控的脉冲。

可以通过时钟/触发控制器启动计数器，在输出比较模式或者 PWM 模式下产生波形。设置 TIMx_CR1 寄存器的 OPM 位将选择单脉冲模式，此时计数器自动地在下一个更新事件 UEV 时停止。

仅当比较值与计数器的初始值不同时，才能产生一个脉冲。启动之前（当定时器正在等待触发），必须

如下配置：

向上计数方式：计数器 $CNT < CCRi \leq ARR$

向下计数方式：计数器 $CNT > CCRi$

单脉冲模式的配置步骤：

如图 14-22 所示，如果用户想在 OC1 生成一个高电平脉冲，它通过 TI2 的输入触发，启动延迟为 t_{DELAY} ，维持时间为 t_{PULSE} 。首先要把 TI2FP2 设定为触发信号，此外单脉冲信号还要通过比较寄存器定义（考虑到时钟频率和预分频器）：

1. TI2FP2 对应到 TI2，配置 TIMx_CCMR1 寄存器中的 CC2S=01
2. TI2FP2 上升沿检测，配置 TIMx_CCER 寄存器中的 CC2P=0
3. 设置 TI2FP2 为从模式下的触发信号（TRGI），配置 TIMx_SMCR 寄存器中的 TS=110
4. 设置 TI2FP2 为计数器的启动信号，配置 TIMx_SMCR 寄存器中的 SMS=110（触发模式）
5. t_{DELAY} 通过 TIMx_CCR1 寄存器配置得到
6. t_{PULSE} 通过 TIMx_ARR - TIMx_CCR1 得到
7. 如果在计数器到达 TIMx_CCR1 时信号由 0 变为 1，到达 TIMx_ARR 时信号由 1 变为 0，则设置 PWM 模式 2，配置 TIMx_CCMR1 寄存器中的 OC1M=111

可选启用预装载功能，配置 TIMx_CCMR1 寄存器中的 OC1PE=1 和 TIMx_CR1 寄存器中的 ARPE=1。如果使用预装载功能，在 TIMx_ARR 和 TIMx_CCR1 配置后要配置 UG 位进行更新，并等待 TI2 的触发。在这个例子里，CC1P=0，TIMx_CR1 寄存器中的 DIR=0，CMS=00。

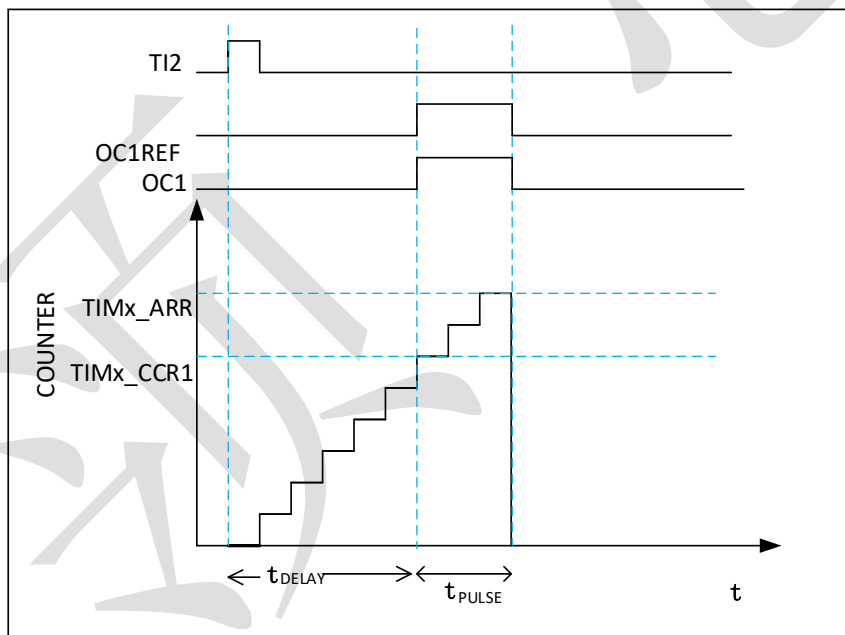


图 14-22 单脉冲模式

14.5.9 特殊情况：OCi 快速使能

在单脉冲模式下，对 Tii 输入脚的边沿检测会设置 CEN 位以启动计数器。然后计数器和比较值间的比较操作产生了单脉冲的输出。但是这些操作需要一定的时钟周期，因此它限制了可得到的最小延时 t_{DELAY} 。

如果要以最小延时输出波形，可以设置 TIM3_CCMRi 寄存器中的 OCiFE 位；此时强制 OCiREF（和 OCx）直接响应激励而不再依赖比较的结果，输出的波形与比较匹配时的波形一样。OCiFE 只在通道配置为 PWM1 和 PWM2 模式的单脉冲模式时起作用。

14.5.10 通过外部事件清除 OCiREF

给定通道的 OCiREF 信号可以通过 OCREF_CLR_INT 清除（相应 TIMx_CCMR 寄存器中的使能位 OCiCE 置为 1）。OCiREF 保持低电平，直到下一个更新事件（UEV）发生。此功能只能在输出比较和 PWM 模式下使用，在强制输出模式下不起作用。

通过配置 TIMx_SMCR 寄存器中的 OCCS 位，可以在 OCREF_CLR 输入和 ETRF（滤波器后的 ETR）之间选择 OCREF_CLR_INT。如果选择了 ETRF，ETR 需要如下配置：

1. 外部触发预分频器要保持关闭：TIMx_SMCR 寄存器中 ETPS=00
2. 外部时钟源模式 2 关闭：TIMx_SMCR 寄存器中 ECE=0
3. 外部触发极性（ETP）和外部触发滤波器可以按照用户需求配置

OCREF_CLR 输入的来源为 ADC 看门狗输出、其他定时器的 TRGO 和各个 ACMP 的输出。OCREF_CLR 的极性可以通过 TIMx_SMCR 寄存器中的 OCCP 配置。

14.5.11 编码器接口模式

选择编码器接口模式的方法：如果计数器只在 TI2 的边沿计数，则置 TIMx_SMCR 寄存器中的 SMS=001；如果只在 TI1 边沿计数，则置 SMS=010；如果计数器同时在 TI1 和 TI2 边沿计数，则置 SMS=011。通过设置 TIMx_CCER 寄存器中的 CC1P 和 CC2P 位，可以选择 TI1 和 TI2 极性；如果需要，还可以对输入进行滤波。

两个输入 TI1 和 TI2 被用来作为增量编码器的接口，参照表 14-1，假定计数器已经启动（TIMx_CR1 寄存器中的 CEN=1），则计数器由每次在 TI1FP1 或 TI2FP2 上的有效跳变驱动。TI1FP1 和 TI2FP2 是 TI1 和 TI2 在通过输入滤波器和极性控制后的信号；如果没有滤波和极性变化，则 TI1FP1=TI1，TI2FP2=TI2。根据两个输入信号的跳变顺序，产生了计数脉冲和方向信号。依据两个输入信号的跳变顺序，计数器向上或向下计数。

编码器接口模式相当于使用了一个带有方向选择的外部时钟。这意味着计数器只在 0 到 TIMx_ARR 寄存器的自动装载值之间连续计数（根据方向决定是 0 到 ARR 计数，或是 ARR 到 0 计数）。所以在开始计数之前必须配置 TIMx_ARR 寄存器；同样，捕获器、预分频器和重复计数器特性等仍工作如常。编码器模式和外部时钟模式 2 不兼容，因此不能同时配置。在这个模式下，计数器依照增量编码器的速度和方向被自动修改，因此计数器的内容始终指示着编码器的位置。计数方向与相连的传感器的旋转方向对应。假设 TI1 和 TI2 不同时变换，表 14-1 列出了所有可能的组合。这里相对信号是针对正在跳变的信号而言，TI1 在跳变时，相对信号的电平就是 TI2FP2 的电平；TI2 在跳变时，相对信号的电平就是 TI1FP1 的电平。

表 14-1 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

外部的增量编码器可以直接与 MCU 连接而不需要外部接口逻辑。然而，一般会使用比较器将编码器的差分输出转换为数字信号，这会大大增加抗噪声干扰能力。编码器输出的第三个信号表示机械零点，可以把

它连接到一个外部中断输入并触发计数器复位。图 14-23 是一个编码器接口模式的示例，显示了计数信号的产生和方向控制。它还显示了当选择了双边沿时，输入抖动是如何被抑制的；抖动可能会在传感器的位置靠近一个转换点时产生。在这个例子中，我们假定配置如下：

- CC1S=01 (TIMx_CCMR1 寄存器, TI1FP1 映射到 TI1)
- CC2S=01 (TIMx_CCMR1 寄存器, TI2FP2 映射到 TI2)
- CC1P=0, IC1F=0000 (TIMx_CCER 寄存器, TI1FP1 不反相, TI1FP1=TI1)
- CC2P=0, IC2F=0000 (TIMx_CCER 寄存器, TI2FP2 不反相, TI2FP2=TI2)
- SMS=011 (TIMx_SMCR 寄存器, TI1 和 TI2 输入均在上升沿和下降沿有效)
- CEN=1 (TIMx_CR1 寄存器, 计数器使能)

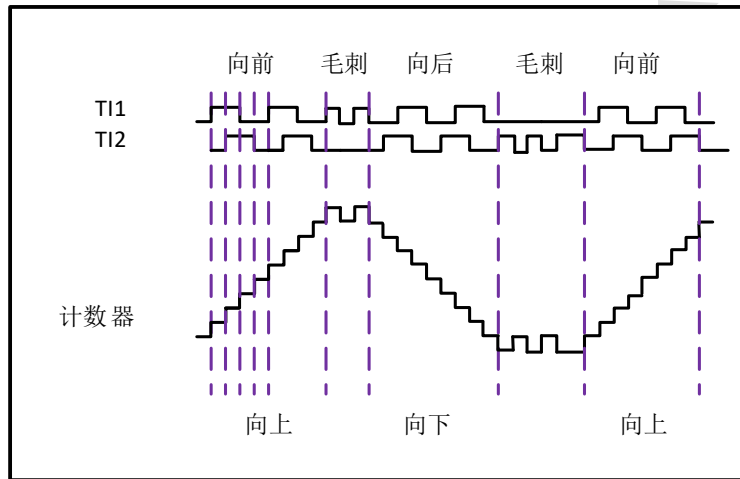


图 14-23 编码器接口模式示例

当其他配置保持一致，TI1FP1 极性反相时的示例：

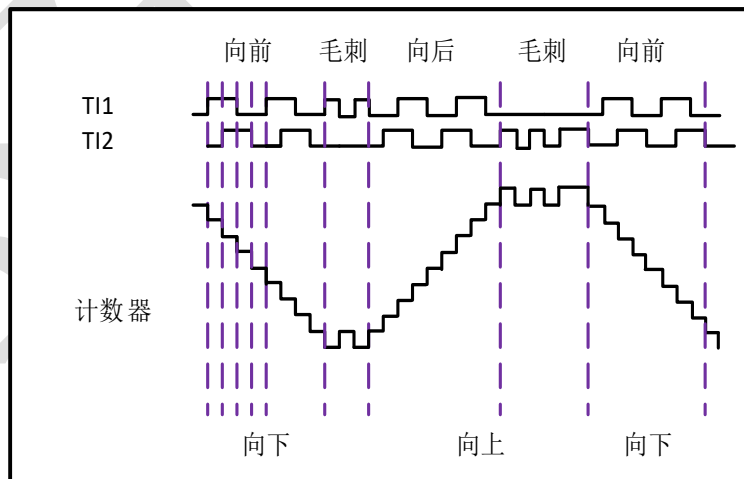


图 14-24 TI1FP1 反相的编码器接口模式示例

当定时器配置成编码器接口模式时，能够提供传感器当前的位置信息。如果使用另一个配置在捕获模式的定时器，我们可以测量两个编码器事件的间隔，并获得动态的信息（速度，加速度，减速度等）。表示机械零点的编码器输出可被用做此目的。根据编码器事件的间隔，我们按照固定的时间读出计数器：可以把计数器的值锁存到捕获寄存器中（捕获信号必须是周期性的，且可以由另一个定时器产生）；也可以通过一个由实时时钟产生的 DMA 请求来读取它的值。

14.5.12 定时器输入异或功能

TIMx_CR2 寄存器中的 TI1S 位，允许通道 1 的输入滤波器连接到一个异或门的输出端，异或门的 3 个输入端为 TIMx_CH1、TIMx_CH2 和 TIMx_CH3。异或输出能够被用于所有定时器的输入功能，如触发或输入捕获。

14.6 TIM3 定时器与外部触发的同步

TIM3 定时器能够在多种模式下和一个外部的触发同步：复位模式、门控模式、触发模式和触发复位模式。计数器允许四种触发输入：ETR（外部触发）；TI1 外部输入；TI2 外部输入；来自芯片内部其他模块。

TIM3 使用 4 种模式与外部的触发信号同步：标准触发模式、复位模式、门控模式和触发复位模式。

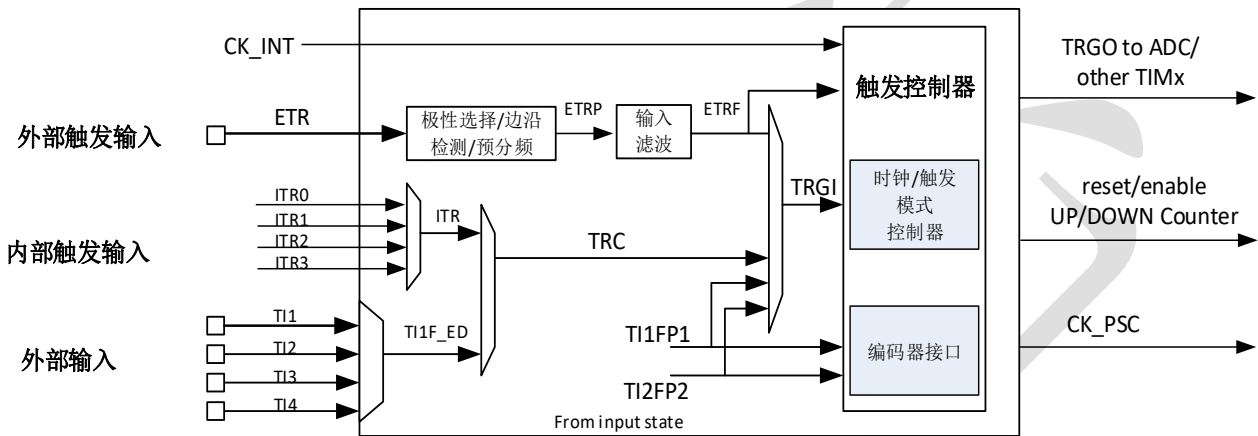


图 14-25 触发输入功能框图

14.6.1 标准触发模式

计数器的使能依赖于选中的触发输入事件。在下面的例子中，TIM3 计数器在 TI2 输入的上升沿开始向上计数：

- 配置通道 2 用于检测 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，保持寄存器 CCMR1 中的 IC2F=0000）
- 触发操作中不使用捕获预分频器，不需要配置；寄存器 CCMR1 中的 CC2S=01，用于选择输入捕获源
- 配置 TIMx 的 CCER 寄存器的 CC2P=0，选择上升沿做为触发条件
- 配置 TIMx 的 SMCR 寄存器的 SMS=110，选择定时器为触发模式
- 配置 TIMx 的 SMCR 寄存器的 TS=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 TIMx_SR 寄存器的 TIF 位被置起。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。

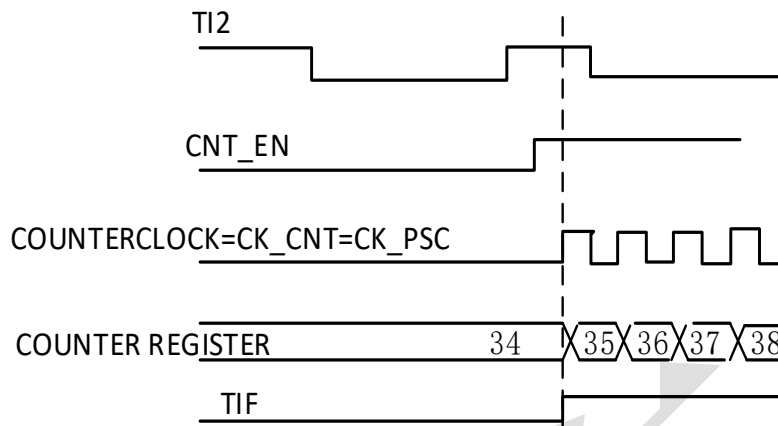


图 14-26 标准触发模式下的控制电路

14.6.2 复位模式

在发生一个触发输入事件时，计数器和它的预分频器能够重新被初始化；同时，如果 `TIMx_CR1` 寄存器的 `URS` 位为低，还产生一个更新事件 `UEV`；然后所有的预装载寄存器（`ARR`、`CCR`）都被更新。在以下的例子中，`TI1` 输入端的上升沿导致向上计数器被清零：

- 配置通道 1 用于检测 `TI1` 的上升沿；配置输入滤波器的带宽（在本例中，不需要任何滤波器，因此保持 `IC1F=0000`）
- 触发操作中不使用捕获预分频器，所以不需要配置；寄存器 `CCMR1` 中的 `CC1S=01`，用于选择输入捕获源
- 配置 `TIMx` 的 `CCER` 寄存器的 `CC1P=0` 来选择极性（只检测上升沿）
- 配置 `TIMx` 的 `SMCR` 寄存器的 `SMS=100`，选择定时器为复位模式
- 配置 `TIMx` 的 `SMCR` 寄存器的 `TS=101`，选择 `TI1` 作为输入源
- 配置 `TIMx_CR1` 寄存器的 `CEN=1`，启动计数器
- 计数器开始依据内部时钟计数，然后正常计数直到 `TI1` 出现一个上升沿；此时，计数器被清零然后从 0 重新开始计数。同时，`TIMx_SR` 寄存器的 `TIF` 位被置起，如果使能了中断（`TIMx_DIER` 寄存器的 `TIE` 位），则产生一个中断请求

下图显示当自动重载寄存器 `TIMx_ARR=0x36` 时的动作。在 `TI1` 上升沿和计数器的实际复位之间的延时取决于 `TI1` 输入端的重同步电路。

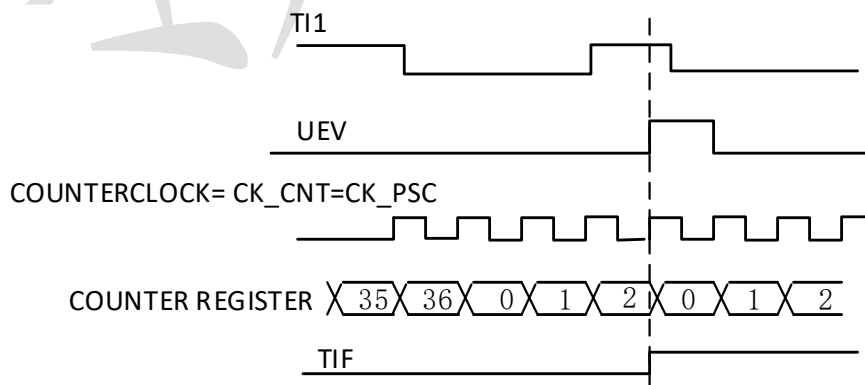


图 14-27 复位模式下的控制电路

14.6.3 门控模式

计数器由选中的输入端信号的电平使能。在以下的例子中，计数器只在 TI1 为低时向上计数：

- 配置通道 1 用于检测 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 IC1F=0000）
- 触发操作中不使用捕获预分频器，所以不需要配置；寄存器 CCMR1 中的 CC1S=01，用于选择输入捕获源
- 配置 TIMx_CCER 寄存器的 CC1P=1 来确定极性（只检测低电平）
- 配置 TIMx_SMCR 寄存器的 SMS=101，选择定时器为门控模式
- 配置 TIMx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源
- 配置 TIMx_CR1 寄存器的 CEN=1，启动计数器（门控模式下，如果 CEN=0，则计数器不能启动，无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，TIMx_SR 寄存器的 TIF 位都会被置起。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。

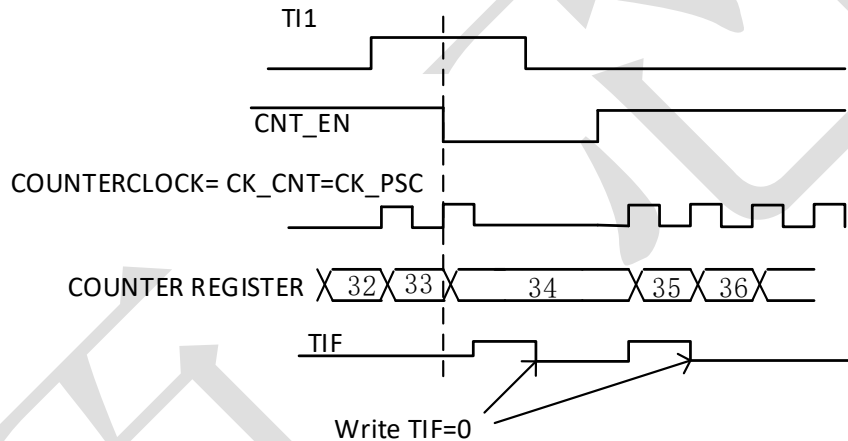


图 14-28 门控模式下的控制电路

14.6.4 触发复位模式

计数器的使能依赖于选中的触发输入事件，在发生一个触发输入事件时，计数器和它的预分频器能够重新被初始化；同时，如果 TIMx_CR1 寄存器的 URS 位为低，还产生一个更新事件 UEV；然后所有的预装载寄存器（ARR、CCR）都被更新。这种模式可视为是标准触发模式和复位模式的结合，其配置与标准触发模式基本一致，不同之处在于 TIMx 的 SMCR 寄存器的 SMS=1000。

14.6.5 外部时钟模式 2 及触发模式

外部时钟模式 2 可以与另一个输入信号的触发模式一起使用。这时，ETR 信号被用作外部时钟的输入，另一个输入信号可用作触发模式（支持标准触发模式，复位模式、门控模式和触发复位模式）。请注意不能把 ETR 配置成 TRGI（通过 TIMx_SMCR 寄存器的 TS 位）。

在下面的例子中，一旦在 TI1 上出现一个上升沿，计数器即在 ETR 的每一个上升沿向上计数一次，通过 TIM3_SMCR 寄存器配置外部触发输入电路。

- 首先配置 ETR：TIMx_SMCR 寄存器中配置 ETF=0000 禁止滤波器，配置 ETPS=00 禁止预分频，配置 ETP=0 监测 ETR 信号的上升沿，配置 ECE=1 使能外部时钟模式 2

- 配置通道 1 用于检测 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 IC1F=0000）
- 触发操作中不使用捕获预分频器，所以不需要配置；寄存器 CCMR1 中的 CC1S=01，用于选择输入捕获源
- 配置 TIMx 的 CCER 寄存器的 CC1P=0，选择上升沿做为触发条件
- 配置 TIMx 的 SMCR 寄存器的 SMS=110，选择定时器为触发模式
- 配置 TIMx 的 SMCR 寄存器的 TS=101，选择 TI1 作为输入源

当 TI1 上出现一个上升沿时，同时 TIMx_SR 寄存器的 TIF 位被置起，计数器开始在 ETR 的上升沿计数。TI1 信号的上升沿和计数器实际时钟之间的延时取决于 TI1 输入端的重同步电路。ETR 信号的上升沿和计数器实际时钟之间的延时取决于 ETRP 输入端的重同步电路。

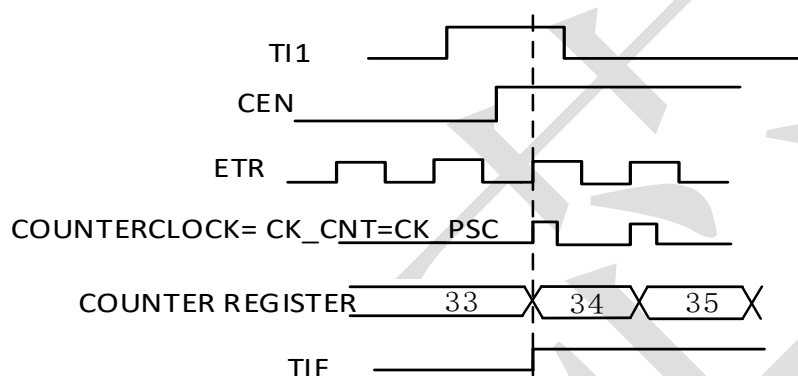


图 14-29 外部时钟模式 2+触发模式下的控制电路

14.7 TIM3 与其他定时器的联接

在芯片中，各个定时器在内部互相联接，用于定时器的同步或连接。当某个定时器配置成主模式时，可以输出触发信号（TRGO）到那些配置为从模式的定时器来完成复位、启动和停止的操作，或者作为那些定时器的驱动时钟。具体可参考图 14-25。

TIM3 的输入触发可以来自芯片内部的其他定时器、比较器 ACMP0/1/2 输出和 ADC；TIM3 的输出 TRGO 可触发芯片内部的 ADC、DAC、比较器 ACMP0/1/2 和其他定时器；TIM3 的输出 OC_i 可触发芯片内部的比较器 ACMP0/1/2 和 ADC；TIM3 的特殊输出 TIM3_oc1_trig 可触发其他定时器，它与 CEN 位相关，具体详见 TIM3_CR2 寄存器 TrgiS 位说明。

表 14-2 TIM3 触发与级联表

信号名称	信号描述	来源	选择寄存器
内部触发 ITR			
ITR0	内部触发 0	tim1_trgo	-
ITR1	内部触发 1	tim17_oc1_trig	-
ITR2	内部触发 2	tim15_trgo	-
ITR3	内部触发 3	tim14_oc1_trig	-
外部 Timer 输入 TI			
TI1	外部 Timer 输入 1	tim3_gpio_ti1/adc_awd/ cp0_out/cp1_out/cp2_out/ tim1_oc1_trig / tim15_oc1_trig / tim16_oc1_trig	SysCtrl_EDU_CFG2
TI2	外部 Timer 输入 2	tim3_gpio_ti2/adc_awd/ cp0_out/cp1_out/cp2_out/ tim1_oc1_trig / tim15_oc1_trig / tim16_oc1_trig	
TI3	外部 Timer 输入 3	tim3_gpio_ti3/adc_awd/ cp0_out/cp1_out/cp2_out/ tim1_oc1_trig / tim15_oc1_trig / tim16_oc1_trig	
TI4	外部 Timer 输入 4	tim3_gpio_ti4/adc_awd/ cp0_out/cp1_out/cp2_out/ tim1_oc1_trig / tim15_oc1_trig / tim16_oc1_trig	
外部触发输入 ETR			
ETR	外部触发	tim3_gpio_etr/adc_awd/ cp0_out/cp1_out/cp2_out/ tim1_oc1_trig / tim15_oc1_trig / tim16_oc1_trig	SysCtrl_EDU_CFG2

注：请参考 SysCtrl_EDU_CFG2 寄存器定义。

14.8 TIM3 中断

TIM3 有 6 个中断请求源，分别映射到 2 个中断矢量上：

- 更新事件中断
- 触发中断
- 输入捕获/输出比较 1 中断
- 输入捕获/输出比较 2 中断
- 输入捕获/输出比较 3 中断
- 输入捕获/输出比较 4 中断

为了使用中断特性，对每个被使用的中断通道，设置 TIMx_DIER 寄存器中相应的中断使能位：TIE、CCiIE 和 UIE 位。

通过设置 TIMx_EGR 寄存器中的相应位，也可以用软件产生上述各个中断源。

14.9 TIM3 寄存器描述

表 14-3 TIM3 相关寄存器表

名称	说明	读写权限	复位值	字节地址
CR1	控制寄存器 1	R/W	0x0000_0000	0x4000_B000
CR2	控制寄存器 2	R/W	0x0000_0000	0x4000_B004

SMCR	从模式控制寄存器	R/W	0x0000_0000	0x4000_B008
DIER	DMA 和中断控制寄存器	R/W	0x0000_0000	0x4000_B00C
SR	状态寄存器	R/W	0x0000_0000	0x4000_B010
EGR	事件产生寄存器	R/W	0x0000_0000	0x4000_B014
CCMR1	捕获/比较模式寄存器 1	R/W	0x0000_0000	0x4000_B018
CCMR2	捕获/比较模式寄存器 2	R/W	0x0000_0000	0x4000_B01C
CCER	捕获/比较使能寄存器	R/W	0x0000_0000	0x4000_B020
CNT	计数寄存器	R/W	0x0000_0000	0x4000_B024
PSC	预分频寄存器	R/W	0x0000_0000	0x4000_B028
ARR	自动重装载寄存器	R/W	0x0000_0000	0x4000_B02C
CCR1	捕获/比较寄存器 1	R/W	0x0000_0000	0x4000_B034
CCR2	捕获/比较寄存器 2	R/W	0x0000_0000	0x4000_B038
CCR3	捕获/比较寄存器 3	R/W	0x0000_0000	0x4000_B03C
CCR4	捕获/比较寄存器 4	R/W	0x0000_0000	0x4000_B040
DCR	DMA 控制寄存器	R/W	0x0000_0000	0x4000_B048
DMAR	DMA 传输寄存器	R/W	0x0000_0000	0x4000_B04C
SysCtrl_EDU_CFG2	互联配置寄存器 2	R/W	0x0000_0000	0x4800_7028
SysCtrl_EDU_CFG5	互联配置寄存器 5	R/W	0x0000_0000	0x4800_7034

注：x 表示不确定；-表示未实现；u 表示不受复位影响；c 表示该位复位值因复位源而异；R 只读；W 只写；R-0 表示只读且读0；R/W 表示可读写（以后章节同上述）。

14.9.1 CR1 控制寄存器 1 (TIM3_CR1)

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
Res.							AS	OCF	CENCE	TI4E	TI3E	TI2E	TI1E	ETRE	FTE
							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
Res.						CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	31:25	保留，必须保持复位值
Bit	24	AS: 影子寄存器读使能
		该位指定当 MCU 读取 TIM3_PSC/TIM3_ARR/TIM3_CCRi 寄存器的时候，是读取影子寄存器还是预装载寄存器
		0: 选择预装载寄存器的值（默认）
		1: 选择影子寄存器的值
Bit	23	OCF: 输出异步抗干扰滤波
		0: 输出滤波禁止（默认）
		1: 输出滤波使能
Bit	22	CENCE: CEN 不使能的时候允许清除 OCREF

	0: CEN 不使能时, OCREF clear 禁止 (默认)
	1: CEN 不使能时, OCREF clear 使能
Bit 21	TI4E: TI4 输入检测/滤波使能
	0: TI4 输入检测/滤波禁止 (默认)
	1: TI4 输入检测/滤波使能
Bit 20	TI3E: TI3 输入检测/滤波使能
	0: TI3 输入检测/滤波禁止 (默认)
	1: TI3 输入检测/滤波使能
Bit 19	TI2E: TI2 输入检测/滤波使能
	0: TI2 输入检测/滤波禁止 (默认)
	1: TI2 输入检测/滤波使能
Bit 18	TI1E: TI1 输入检测/滤波使能
	0: TI1 输入检测/滤波禁止 (默认)
	1: TI1 输入检测/滤波使能
Bit 17	ETRE: 外部触发输入检测/滤波使能
	0: ETR 输入检测/滤波禁止 (默认)
	1: ETR 输入检测/滤波使能
Bit 16	FTE: 数字滤波器 (ETR、Tii) 的 DTS 时钟使能
	0: 数字滤波器的 DTS 时钟禁止 (默认)
	1: 数字滤波器的 DTS 时钟使能
Bits 15:10	保留, 必须保持复位值
Bits 9:8	CKD[1:0]: CK_INT 时钟和死区/采样时钟 (CK_DTS) 的分频系数, DTS 时钟供给死区发生器和数字滤波器 (ETR、Tii) 使用
	00: $t_{DTS} = t_{CK_INT}$ (默认)
	01: $t_{DTS} = 2 * t_{CK_INT}$
	10: $t_{DTS} = 4 * t_{CK_INT}$
	11: 保留
Bit 7	ARPE: 自动预装载允许位
	0: TIM3_ARR 寄存器可以被直接写入 (默认)
	1: TIM3_ARR 寄存器通过预装载寄存器更新
Bits 6:5	CMS[1:0]: 选择中央对齐模式
	00: 边沿对齐模式, 计数器依据方向位 (DIR) 向上或向下计数。(默认)
	01: 中央对齐模式 1, 计数器交替地向上和向下计数。只有在计数器向下计数时, 输出比较中断标志位 (TIM3_CCMRi 寄存器中 CCIS=00 条件下) 才会被置 1。
	10: 中央对齐模式 2, 计数器交替地向上和向下计数。只有在计数器向上计数时, 输出比较中断标志位 (TIM3_CCMRi 寄存器中 CCIS=00 条件下) 才会被置 1。
	11: 中央对齐模式 3, 计数器交替地向上和向下计数。在计数器向上或向下计数时, 输出比较中断标志位 (TIM3_CCMRi 寄存器中 CCIS=00 条件下) 会被置 1。
	注 1: 在计数器开启时 (CEN=1), 不允许从边沿对齐模式切换到中央对齐模式。
	注 2: 在中央对齐模式下, 编码器模式 (TIM3_SMCR 寄存器中的 SMS=001、010 或 011) 必须被禁止。
Bit 4	DIR: 计数器方向
	0: 计数器向上计数 (默认)



	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>
Bit 3	OPM: 单脉冲模式
	0: 在发生更新事件时, 计数器不停止 (默认)
	1: 在发生下一次更新事件时, 计数器停止 (清除 CEN 位)
	<i>注: TIM3 的 OPM 模式只有在至少有一个通道处于输出模式时才会生效。</i>
Bit 2	URS: 更新请求源
	0: 如果 UDIS 允许产生更新事件, 则下述任一事件产生一个更新中断 (默认):
	- 计数器上溢/下溢
	- 软件设置 UG 位
	- 时钟/触发控制器产生的硬件复位
	1: 如果 UDIS 允许产生更新事件, 则只有当计数器上溢/下溢时才产生更新中断
Bit 1	UDIS: 禁止更新
	0: 一旦下列事件发生, 产生更新事件 (默认):
	- 计数器溢出/下溢
	- 软件设置 UG 位
	- 时钟/触发控制器产生的硬件复位
	1: 不产生更新事件, 影子寄存器 (ARR、PSC、CCR) 保持它们的值。如果 UG 位被配置或时钟/触发控制器发出了一个硬件复位, 则计数器和预分频器被重新初始化
Bit 0	CEN: 计数器使能位
	0: 计数器禁止 (默认)
	1: 计数器使能
	<i>注: 在软件配置了 CEN 后, 外部时钟模式、复位模式、门控模式和编码器模式才能工作。触发模式可以自动通过硬件启动。在门控模式下, 该位读回值表示门控状态。</i>

14.9.2 CR2 控制寄存器 2 (TIM3_CR2)

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
Res.															TrigS
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								TI1S	MMS[2:0]			CCDS	Res.		
								rw	rw			rw			

Bits 31:17	保留, 必须保持复位值
Bit 16	TrigS: TIMx_oc1_trig 触发源选择
	0: 触发源选择为 OC1&CEN (默认)
	1: 触发源选择为 CEN
Bit 15:8	保留, 必须保持复位值
Bit 7	TI1S: TI1 输入选择

	0: CC1 输入管脚连到 TI1 (默认)
	1: CC1、CC2 和 CC3 管脚经异或后连到 TI1
Bits 6:4	MMS[2:0]: 主模式选择
	用于选择在主模式下送到其他模块的同步信号 (TRGO)
	000: 复位 - 软件设置 UG 位或时钟/触发控制器产生的硬件复位被用作触发输出 (TRGO)。如果触发输入 (时钟/触发控制器配置为复位模式) 产生复位, 则 TRGO 上的信号会延迟到与实际的复位同步。(默认)
	001: 使能 - 计数器使能信号被用作触发输出 (TRGO)。其用于同时启动多个定时器, 或在一段时间内控制从定时器和其他模块。计数器使能信号是通过 CEN 控制位或门控模式下的触发输入信号产生。除非选择了主/从模式 (见 TIM3_SMCR 寄存器中 MSM 位的描述), 否则当计数器使能信号受控于触发输入时, TRGO 上会有一个延迟。
	010: 更新 - 更新事件被用作触发输出 (TRGO)。在这种模式下, 一个主定时器可用作为一个从定时器的预分频器。
	011: 比较脉冲 - 一旦发生一次捕获或一次比较成功, 当 CC1IF 标志被置 1 时 (即使它已经为高), 触发输出送出一个正脉冲 (TRGO)。
	100: 比较 - OC1REF 信号被用作触发输出 (TRGO)。
	101: 比较 - OC2REF 信号被用作触发输出 (TRGO)。
	110: 比较 - OC3REF 信号被用作触发输出 (TRGO)。
	111: 比较 - OC4REF 信号被用作触发输出 (TRGO)。
Bit 3	CCDS: 捕获/比较 DMA 选择
	0: 当 CCx 事件发生的时候发送 CCx DMA 请求 (默认)
	1: 当更新事件发生的时候发送 CCx DMA 请求
Bit 2:0	保留, 必须保持复位值

14.9.3 SMCR 从模式控制寄存器 (TIM3_SMCR)

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
Res.														OCCP	SMS[3]
														rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]			MSM	TS[2:0]			OCCS	SMS[2:0]			
rw	rw	rw		rw			rw	rw			rw	rw			

Bits 31:18	保留, 必须保持复位值
Bit 17	OCCP: OCREF_CLR 输入极性选择
	0: OCREF_CLR 输入高有效 (默认)
	1: OCREF_CLR 输入低有效
Bit 16	SMS[3]: 从模式选择的 bit 3
	参考 SMS[2:0] 的描述
Bit 15	ETP: 外部触发极性

	0: ETR 不反相, 即高电平或上升沿有效 (默认)
	1: ETR 反相, 即低电平或下降沿有效
Bit 14	ECE : 外部时钟使能, 用于使能外部时钟模式 2
	0: 外部时钟模式 2 禁止 (默认)
	1: 外部时钟模式 2 使能, 计数器的时钟为 ETRF 的有效边沿
	<i>注 1: ECE 位置 1 的效果与选择把 TRGI 连接到 ETRF 的外部时钟模式 1 相同 (TIM3_SMCR 寄存器中, SMS=111, TS=111)。</i>
	<i>注 2: 外部时钟模式 2 可与下列模式同时使用: 标准触发模式、复位模式、门控模式和触发复位模式。但是, 此时 TRGI 决不能与 ETRF 相连 (TIM3_SMCR 寄存器中, TS 不能为 111)。</i>
	<i>注 3: 如果外部时钟模式 1 与外部时钟模式 2 同时使能, 外部时钟输入为 ETRF。</i>
Bits 13:12	ETPS : 外部触发预分频器
	外部触发信号 EPRP 的频率最大不能超过定时器时钟 CK_INT 频率的 1/4。可用预分频器来降低 ETRP 的频率, 当 EPRP 的频率很高时非常有用。
	00: 预分频器关闭 (默认)
	01: EPRP 的频率/2
	10: EPRP 的频率/4
	11: EPRP 的频率/8
Bits 11:8	ETF[3:0] : 外部触发滤波器选择
	该位域定义了 ETRP 的采样频率及数字滤波器的长度。数字滤波器由一个事件计数器组成, 只有发生了 N 个连续事件后输出的跳变才被认为有效。这里的 DTS 时钟是 CK_INT 经过 CR1 寄存器中 CKD 配置分频的。
	0000: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}$, 无滤波器 (默认)
	0001: 采样频率 $f_{\text{SAMPLING}} = f_{\text{CK_INT}}$, N=2
	0010: 采样频率 $f_{\text{SAMPLING}} = f_{\text{CK_INT}}$, N=4
	0011: 采样频率 $f_{\text{SAMPLING}} = f_{\text{CK_INT}}$, N=8
	0100: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/2$, N=6
	0101: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/2$, N=8
	0110: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$, N=6
	0111: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$, N=8
	1000: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/8$, N=6
	1001: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/8$, N=8
	1010: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$, N=5
	1011: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$, N=6
	1100: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$, N=8
	1101: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$, N=5
	1110: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$, N=6
	1111: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$, N=8
	<i>注: 请注意采样频率中 $f_{\text{CK_INT}}$ 和 f_{DTS} 的差别。</i>
Bit 7	MSM : 主/从模式
	0: 无作用 (默认)
	1: 触发输入 (TRGI) 上的事件被延迟了, 以允许 TIM3 与它的从定时器间通过 TRGO 完美同步
Bits 6:4	TS[2:0] : 选择同步计数器的触发输入

	000: 内部触发 ITR0, 连接 TIM1_TRGO (默认)
	001: 内部触发 ITR1, 连接 TIM17_oc1_trig
	010: 内部触发 ITR2, 连接 TIM15_TRGO
	011: 内部触发 ITR3, 连接 TIM14_oc1_trig
	100: TI1 的边沿检测器 (TI1F_ED)
	101: 滤波后的定时器输入 1 (TI1FP1)
	110: 滤波后的定时器输入 2 (TI2FP2)
	111: 外部触发输入 (ETRF)
	<i>注: 这些位只能在未用到 (如 SMS=000) 时被改变, 以避免在改变时产生错误的边沿检测。</i>
Bit 3	OCCS: 选择 OCREF 清除源
	0: OCREF_CLR_INT 连接到 OCREF_CLR 输入 (默认)
	1: OCREF_CLR_INT 连接到 ETRF 输入
Bits 2:0	SMS[2:0]: 从模式选择
	当选择外部信号时, 触发信号 (TRGI) 的有效边沿与外部输入的极性相关。
	000: 从模式禁止 - 如果 CEN=1, 则预分频器直接由内部时钟驱动。(默认)
	001: 编码器模式 1 - 根据 TI1FP1 的电平, 计数器在 TI2FP2 的边沿向上/下计数。
	010: 编码器模式 2 - 根据 TI2FP2 的电平, 计数器在 TI1FP1 的边沿向上/下计数。
	011: 编码器模式 3 - 根据另一个输入的电平, 计数器在 TI1FP1 和 TI2FP2 的边沿向上/下计数。
	100: 复位模式 - 在选中的触发输入 (TRGI) 的上升沿时重新初始化计数器, 并且产生一个更新寄存器的信号。
	101: 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。
	110: 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。
	111: 外部时钟模式 1 - 选中的触发输入 (TRGI) 的上升沿驱动计数器。
	1000 (结合 SMS[3]): 触发复位模式 - 计数器在触发输入 (TRGI) 的上升沿启动, 重新初始化计数器, 并且产生一个更新寄存器的信号。
	<i>注: 如果 TI1F_ED 被选为触发输入 (TS=100) 时, 不要使用门控模式。这是因为 TI1F_ED 在每次 TI1F 变化时只是输出一个脉冲, 然而门控模式要检查触发输入的电平。</i>

14.9.4 DIER DMA 和中断控制寄存器 (TIM3_DIER)

地址偏移: 0x0C

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	Res.	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res.	TIE	Res.	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rw		rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw

Bits 31:15	保留，必须保持复位值
Bit 14	TDE: 触发 DMA 请求使能
	0: 触发 DMA 请求禁止 (默认)
	1: 触发 DMA 请求使能
Bit 13	保留，必须保持复位值
Bit 12	CC4DE: 捕获/比较 4 的 DMA 请求使能
	0: 捕获/比较 4 的 DMA 请求禁止 (默认)
	1: 捕获/比较 4 的 DMA 请求使能
Bit 11	CC3DE: 捕获/比较 3 的 DMA 请求使能
	0: 捕获/比较 3 的 DMA 请求禁止 (默认)
	1: 捕获/比较 3 的 DMA 请求使能
Bit 10	CC2DE: 捕获/比较 2 的 DMA 请求使能
	0: 捕获/比较 2 的 DMA 请求禁止 (默认)
	1: 捕获/比较 2 的 DMA 请求使能
Bit 9	CC1DE: 捕获/比较 1 的 DMA 请求使能
	0: 捕获/比较 1 的 DMA 请求禁止 (默认)
	1: 捕获/比较 1 的 DMA 请求使能
Bit 8	UDE: 更新的 DMA 请求使能
	0: 更新的 DMA 请求禁止 (默认)
	1: 更新的 DMA 请求使能
Bit 7	保留，必须保持复位值
Bit 6	TIE: 触发中断使能
	0: 触发中断禁止 (默认)
	1: 触发中断使能
Bit 5	保留，必须保持复位值
Bit 4	CC4IE: 捕获/比较 4 中断使能
	0: 捕获/比较 4 中断禁止 (默认)
	1: 捕获/比较 4 中断使能
Bit 3	CC3IE: 捕获/比较 3 中断使能
	0: 捕获/比较 3 中断禁止 (默认)
	1: 捕获/比较 3 中断使能
Bit 2	CC2IE: 捕获/比较 2 中断使能
	0: 捕获/比较 2 中断禁止 (默认)
	1: 捕获/比较 2 中断使能
Bit 1	CC1IE: 捕获/比较 1 中断使能
	0: 捕获/比较 1 中断禁止 (默认)
	1: 捕获/比较 1 中断使能
Bit 0	UIE: 更新中断使能
	0: 更新中断禁止 (默认)
	1: 更新中断使能

14.9.5 SR 状态寄存器 (TIM3_SR)

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
Res.															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			CC4OF	CC3OF	CC2OF	CC1OF	Res.		TIF	Res.	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			r	r	r	r			rw		rw	rw	rw	rw	rw

Bits	31:13	保留, 必须保持复位值
Bit	12	CC4OF : 捕获/比较 4 过捕获/过比较标志, 参考 CC1OF 位
Bit	11	CC3OF : 捕获/比较 3 过捕获/过比较标志, 参考 CC1OF 位
Bit	10	CC2OF : 捕获/比较 2 过捕获/过比较标志, 参考 CC1OF 位
Bit	9	CC1OF : 捕获/比较 1 过捕获/过比较标志
		该位可由硬件置 1, 软件向 CC1IF 位写 0 可清除该位
		0: 无过捕获/过比较产生 (默认)
		1: 计数器的值被捕获或匹配到 TIM1_CCR1 寄存器时 CC1IF 已经置 1
Bit	8:7	保留, 必须保持复位值
Bit	6	TIF : 触发中断标志
		当发生触发事件(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)时该位由硬件置 1, 软件写 0 可清除该位
		0: 无触发事件产生 (默认)
		1: 触发中断挂起
Bit	5	保留, 必须保持复位值
Bit	4	CC4IF : 捕获/比较 4 中断标志, 参考 CC1IF 位
Bit	3	CC3IF : 捕获/比较 3 中断标志, 参考 CC1IF 位
Bit	2	CC2IF : 捕获/比较 2 中断标志, 参考 CC1IF 位
Bit	1	CC1IF : 捕获/比较 1 中断标志
		如果通道 CC1 配置为输出模式:
		当计数器值与比较值匹配时该位由硬件置 1, 但在中央对齐模式下除外 (参考 TIM3_CR1 寄存器的 CMS 位)。软件写 0 可清除该位, 但是当 CC1OF 也为 1 时, 需要清除 2 次。
		0: 无匹配发生 (默认)
		1: TIM3_CNT 的值与 TIM3_CCR1 的值匹配
		注 : 在中央对齐模式下, 当计数器值为 0 时, 向上计数, 当计数器值为 ARR 时, 向下计数 (它从 0 向上计数到 ARR-1, 再由 ARR 向下计数到 1)。因此, 对所有的 SMS 位值, 这两个值都不会置位 CC1IF。但是, 如果 CCR1>ARR, 则当 CNT 达到 ARR 值时, CC1IF 置 1。
		如果通道 CC1 配置为输入模式:
		当捕获事件发生时该位由硬件置 1, 软件写 0 或通过读 TIM3_CCR1 可清除该位, 但是当 CC1OF 也为 1 时, 需要清除 2 次。

	0: 无输入捕获产生 (默认)
	1: 计数器值已被捕获至 TIM3_CCR1 (在 IC1 上检测到与所选极性相同的边沿)
Bit 0	UIF : 更新中断标志
	当产生更新事件时该位由硬件置 1, 软件写 0 可清除该位
	0: 无更新事件产生 (默认)
	1: 更新中断挂起, 当相关寄存器被更新时该位由硬件置 1
	- 若 TIM3_CR1 寄存器的 UDIS=0, 当计数器上溢或下溢时
	- 若 TIM3_CR1 寄存器的 UDIS=0、URS=0, 当软件设置 TIM3_EGR 寄存器的 UG 位对计数器 CNT 重新初始化时
	- 若 TIM3_CR1 寄存器的 UDIS=0、URS=0, 当计数器 CNT 被触发事件重新初始化时

14.9.6 EGR 事件产生寄存器 (TIM3_EGR)

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
Res.															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.										TG	Res.	CC4G	CC3G	CC2G	CC1G	UG
									w			w	w	w	w	w

Bits 31:7	保留, 必须保持复位值
Bit 6	TG : 产生触发事件
	该位由软件置 1, 用于产生一个触发事件, 由硬件自动清 0
	0: 无动作 (默认)
	1: 产生触发事件。此时 TIF=1, 若 TIE=1, 则产生相应的中断
Bit 5	保留, 必须保持复位值
Bit 4	CC4G : 产生捕获/比较 4 事件, 参考 CC1G 位
Bit 3	CC3G : 产生捕获/比较 3 事件, 参考 CC1G 位
Bit 2	CC2G : 产生捕获/比较 2 事件, 参考 CC1G 位
Bit 1	CC1G : 产生捕获/比较 1 事件
	该位由软件置 1, 用于产生一个捕获/比较事件, 由硬件自动清 0
	0: 无动作 (默认)
	1: 在通道 CC1 上产生一个捕获/比较事件
	若通道 CC1 配置为输出:
	设置 CC1IF=1, 若 CC1IE=1, 则产生相应的中断; 若 CC1IF 已经为 1, 则设置 CC1OF=1
	若通道 CC1 配置为输入:
	当前的计数器值被捕获至 TIM3_CCR1 寄存器, 设置 CC1IF=1, 若 CC1IE=1, 则产生相应的中断; 若 CC1IF 已经为 1, 则设置 CC1OF=1
Bit 0	UG : 产生更新事件

	该位由软件置 1，用于产生一个更新事件，由硬件自动清 0
	0：无动作（默认）
	1：重新初始化计数器，并产生一个更新事件。注意预分频器的计数器也被清 0（但是预分频系数不变）。若中央对齐模式或向上计数（DIR=0）则计数器被清 0；若向下计数（DIR=1）则计数器取 TIM3_ARR 的值

14.9.7 CCMR1 捕获/比较模式寄存器 1 (TIM3_CCMR1)

地址偏移：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 CCIS 位定义。该寄存器部分位的作用在输入和输出模式下不同。OCxx 描述了通道在输出模式下的功能，ICxx 描述了通道在输入模式下的功能。因此必须注意，同一个寄存器位在输出模式和输入模式下的功能是不同的。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]		OC2PE	OC2FE	CC2S[1:0]			OC1CE	OC1M[2:0]		OC1PE	OC1FE	CC1S[1:0]		
IC2F[3:0]			IC2PSC[1:0]			IC1F[3:0]			IC1PSC[1:0]						
rw			rw			rw			rw			rw			

输出比较模式

Bits	31:16	保留，必须保持复位值
Bit	15	OC2CE : 输出比较 2 清除使能，参考 OC1CE
Bits	14:12	OC2M[2:0] : 输出比较 2 模式，参考 OC1M
Bit	11	OC2PE : 输出比较 2 预装载使能，参考 OC1PE
Bit	10	OC2FE : 输出比较 2 快速使能，参考 OC1FE
Bits	9:8	CC2S[1:0] : 捕获/比较 2 选择
		该位定义通道的方向（输入/输出），以及输入脚的选择
		00: CC2 通道被配置为输出（默认）
		01: CC2 通道被配置为输入，IC2 映射在 TI2FP2 上
		10: CC2 通道被配置为输入，IC2 映射在 TI1FP2 上
		11: CC2 通道被配置为输入，IC2 映射在 TRC 上。此模式仅工作在触发输入选中 ITRx 或 TI1F_ED 时（由 TIM3_SMCR 寄存器的 TS 位选择）
		<i>注：CC2S 仅在通道关闭时（TIM3_CCER 寄存器的 CC2E=0 且已被更新）才是可写的。</i>
Bit	7	OC1CE : 输出比较 1 清零使能
		0: OC1REF 清除禁止（默认）
		1: OC1REF 清除使能
Bits	6:4	OC1M[2:0] : 输出比较 1 模式选择
		这 3 位定义了输出参考信号 OC1REF 的动作，而 OC1REF 决定了 OC1 的值。OC1REF 是高电平有效，而 OC1 的有效电平取决于 CC1P。
		000: 冻结。输出比较寄存器 TIM3_CCR1 与计数器 TIM3_CNT 间的比较对 OC1REF 不起作用。（默认）

	001: 匹配时设置通道 1 的输出为有效电平。当计数器 TIM3_CNT 的值与捕获/比较寄存器 1 (TIM3_CCR1) 相同时, 强制 OC1REF 为高。
	010: 匹配时设置通道 1 的输出为无效电平。当计数器 TIM3_CNT 的值与捕获/比较寄存器 1 (TIM3_CCR1) 相同时, 强制 OC1REF 为低。
	011: 翻转。当 TIM3_CCR1=TIM3_CNT 时, 翻转 OC1REF 的电平。
	100: 强制为无效电平。强制 OC1REF 为低。
	101: 强制为有效电平。强制 OC1REF 为高。
	110: PWM 模式 1 — 在向上计数时, 一旦 TIM3_CNT<TIM3_CCR1 时通道 1 为有效电平, 否则为无效电平; 在向下计数时, 一旦 TIM3_CNT>TIM3_CCR1 时通道 1 为无效电平, 否则为有效电平。
	111: PWM 模式 2 — 在向上计数时, 一旦 TIM3_CNT<TIM3_CCR1 时通道 1 为无效电平, 否则为有效电平; 在向下计数时, 一旦 TIM3_CNT>TIM3_CCR1 时通道 1 为有效电平, 否则为无效电平。
	<i>注: 在 PWM 模式 1 或 PWM 模式 2 中, 只有当比较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时, OC1REF 电平才改变。(参考章节 14.5.7 PWM 模式)。</i>
Bit 3	OC1PE: 输出比较 1 预装载使能
	0: 禁止 TIM3_CCR1 寄存器的预装载功能, 可随时写入 TIM3_CCR1 寄存器, 并且新写入的数值立即起作用 (默认)
	1: 使能 TIM3_CCR1 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIM3_CCR1 的预装载值在更新事件到来时被加载至当前寄存器中
	<i>注: 为了操作正确, 在 PWM 模式下必须使能预装载功能。但在单脉冲模式下 (TIM3_CR1 寄存器的 OPM=1), 它不是必须的。</i>
Bit 2	OC1FE: 输出比较 1 快速使能
	该位用于加快 CC 输出对触发输入的响应
	0: 根据计数器与 CCR1 的值, CC1 正常操作。当触发输入出现一个有效边沿时, 通过计数器比较输出 CC1。(默认)
	1: 触发输入的有效边沿的作用就如同发生了一次比较匹配。因此, OC1REF 被直接设置为有效电平, 而与比较结果无关。触发输入的有效边沿与 CC1 输出之间的延时被缩短。
	<i>注: OC1FE 只在通道被配置成 PWM 模式 1/2 的单脉冲模式时起作用。</i>
Bits 1:0	CC1S[1:0]: 捕获/比较 1 选择
	该位定义通道的方向 (输入/输出), 以及输入脚的选择
	00: CC1 通道被配置为输出 (默认)
	01: CC1 通道被配置为输入, IC1 映射在 TI1FP1 上
	10: CC1 通道被配置为输入, IC1 映射在 TI2FP1 上
	11: CC1 通道被配置为输入, IC1 映射在 TRC 上。此模式仅工作在触发输入选中 ITRx 或 TI1F_ED 时 (由 TIM3_SMCR 寄存器的 TS 位选择)
	<i>注: CC1S 仅在通道关闭时 (TIM3_CCER 寄存器的 CC1E=0 且已被更新) 才是可写的。</i>

输入捕获模式

Bits 31:16	保留, 必须保持复位值
Bits 15:12	IC2F[3:0]: 输入捕获 2 滤波器, 参考 IC1F
Bits 11:10	IC2PSC[1:0]: 输入捕获 2 预分频器, 参考 IC1PSC

Bits 9:8	CC2S[1:0]: 捕获/比较 2 选择
	该位定义通道的方向（输入/输出），以及输入脚的选择
	00: CC2 通道被配置为输出（默认）
	01: CC2 通道被配置为输入，IC2 映射在 TI2FP2 上
	10: CC2 通道被配置为输入，IC2 映射在 TI1FP2 上
	11: CC2 通道被配置为输入，IC2 映射在 TRC 上。此模式仅工作在触发输入选中 ITRx 或 TI1F_ED 时（由 TIM3_SMCR 寄存器的 TS 位选择）
	<i>注：CC2S 仅在通道关闭时（TIM3_CCER 寄存器的 CC2E=0 且已被更新）才是可写的。</i>
Bits 7:4	IC1F[3:0]: 输入捕获 1 滤波器
	该位域定义了 TI1 输入的采样频率及数字滤波器的长度。数字滤波器由一个事件计数器组成，只有发生了 N 个连续事件后输出的跳变才被认为有效。这里的 DTS 时钟是 CK_INT 经过 CR1 寄存器中 CKD 配置分频的。
	0000: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}$ ，无滤波器（默认）
	0001: 采样频率 $f_{\text{SAMPLING}} = f_{\text{CK_INT}}$ ，N=2
	0010: 采样频率 $f_{\text{SAMPLING}} = f_{\text{CK_INT}}$ ，N=4
	0011: 采样频率 $f_{\text{SAMPLING}} = f_{\text{CK_INT}}$ ，N=8
	0100: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/2$ ，N=6
	0101: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/2$ ，N=8
	0110: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$ ，N=6
	0111: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$ ，N=8
	1000: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/8$ ，N=6
	1001: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/8$ ，N=8
	1010: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$ ，N=5
	1011: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$ ，N=6
	1100: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$ ，N=8
	1101: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$ ，N=5
	1110: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$ ，N=6
	1111: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$ ，N=8
	<i>注：请注意采样频率中 $f_{\text{CK_INT}}$ 和 f_{DTS} 的差别。</i>
Bits 3:2	IC1PSC[1:0]: 输入捕获 1 预分频器
	IC1 的预分频系数，一旦 CC1E=0（TIM3_CCER 寄存器中），则预分频器复位
	00: 无预分频器，TI1 上每 1 个有效边沿作为一次输入（默认）
	01: TI1 上每 2 个有效边沿作为一次输入
	10: TI1 上每 4 个有效边沿作为一次输入
	11: TI1 上每 8 个有效边沿作为一次输入
Bits 1:0	CC1S[1:0]: 捕获/比较 1 选择
	该位定义通道的方向（输入/输出），以及输入脚的选择
	00: CC1 通道被配置为输出（默认）
	01: CC1 通道被配置为输入，IC1 映射在 TI1FP1 上
	10: CC1 通道被配置为输入，IC1 映射在 TI2FP1 上
	11: CC1 通道被配置为输入，IC1 映射在 TRC 上。此模式仅工作在触发输入选中 ITRx 或 TI1F_ED 时（由 TIM3_SMCR 寄存器的 TS 位选择）
	<i>注：CC1S 仅在通道关闭时（TIM3_CCER 寄存器的 CC1E=0 且已被更新）才是可写的。</i>



14.9.8 CCMR2 捕获/比较模式寄存器 2 (TIM3_CCMR2)

地址偏移: 0x1C

复位值: 0x0000_0000

参考 CCMR1 寄存器的描述。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]		CC4S[1:0]		IC3F[3:0]				IC3PSC[1:0]		CC3S[1:0]	
rw				rw		rw		rw				rw		rw	

输出比较模式

Bits 31:16	保留, 必须保持复位值
Bit 15	OC4CE : 输出比较 4 清除使能, 参考 OC1CE
Bits 14:12	OC4M[2:0] : 输出比较 4 模式, 参考 OC1M
Bit 11	OC4PE : 输出比较 4 预装载使能, 参考 OC1PE
Bit 10	OC4FE : 输出比较 4 快速使能, 参考 OC1FE
Bits 9:8	CC4S[1:0] : 捕获/比较 4 选择
	该位定义通道的方向 (输入/输出), 以及输入脚的选择
	00: CC4 通道被配置为输出 (默认)
	01: CC4 通道被配置为输入, IC4 映射在 TI4FP4 上
	10: CC4 通道被配置为输入, IC4 映射在 TI3FP4 上
	11: CC4 通道被配置为输入, IC4 映射在 TRC 上。此模式仅工作在触发输入选中 ITRx 或 TI1F_ED 时 (由 TIM3_SMCR 寄存器的 TS 位选择)
	<i>注: CC4S 仅在通道关闭时 (TIM3_CCER 寄存器的 CC4E=0 且已被更新) 才是可写的。</i>
Bit 7	OC3CE : 输出比较 3 清除使能, 参考 OC1CE
Bits 6:4	OC3M[2:0] : 输出比较 3 模式, 参考 OC1M
Bit 3	OC3PE : 输出比较 3 预装载使能, 参考 OC1PE
Bit 2	OC3FE : 输出比较 3 快速使能, 参考 OC1FE
Bits 1:0	CC3S[1:0] : 捕获/比较 3 选择
	该位定义通道的方向 (输入/输出), 以及输入脚的选择
	00: CC3 通道被配置为输出 (默认)
	01: CC3 通道被配置为输入, IC3 映射在 TI3FP3 上
	10: CC3 通道被配置为输入, IC3 映射在 TI4FP3 上
	11: CC3 通道被配置为输入, IC3 映射在 TRC 上。此模式仅工作在触发输入选中 ITRx 或 TI1F_ED 时 (由 TIM3_SMCR 寄存器的 TS 位选择)
	<i>注: CC3S 仅在通道关闭时 (TIM3_CCER 寄存器的 CC3E=0 且已被更新) 才是可写的。</i>

输入捕获模式

Bits 31:16	保留, 必须保持复位值
------------	-------------



Bits 15:12	IC4F[3:0]: 输入捕获 4 滤波器, 参考 IC1F
Bits 11:10	IC4PSC[1:0]: 输入捕获 4 预分频器, 参考 IC1PSC
Bits 9:8	CC4S[1:0]: 捕获/比较 4 选择
	该位定义通道的方向 (输入/输出), 以及输入脚的选择
	00: CC4 通道被配置为输出 (默认)
	01: CC4 通道被配置为输入, IC4 映射在 TI4FP4 上
	10: CC4 通道被配置为输入, IC4 映射在 TI3FP4 上
	11: CC4 通道被配置为输入, IC4 映射在 TRC 上。此模式仅工作在触发输入选中 ITRx 或 TI1F_ED 时 (由 TIM3_SMCR 寄存器的 TS 位选择)
	<i>注: CC4S 仅在通道关闭时 (TIM3_CCER 寄存器的 CC4E=0 且已被更新) 才是可写的。</i>
Bits 7:4	IC3F[3:0]: 输入捕获 3 滤波器, 参考 IC1F
Bits 3:2	IC3PSC[1:0]: 输入捕获 3 预分频器, 参考 IC1PSC
Bits 1:0	CC3S[1:0]: 捕获/比较 3 选择
	该位定义通道的方向 (输入/输出), 以及输入脚的选择
	00: CC3 通道被配置为输出 (默认)
	01: CC3 通道被配置为输入, IC3 映射在 TI3FP3 上
	10: CC3 通道被配置为输入, IC3 映射在 TI4FP3 上
	11: CC3 通道被配置为输入, IC3 映射在 TRC 上。此模式仅工作在触发输入选中 ITRx 或 TI1F_ED 时 (由 TIM3_SMCR 寄存器的 TS 位选择)
	<i>注: CC3S 仅在通道关闭时 (TIM3_CCER 寄存器的 CC3E=0 且已被更新) 才是可写的。</i>

14.9.9 CCER 捕获/比较使能寄存器 (TIM3_CCER)

地址偏移: 0x20

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	Res.	CC4P	CC4E	CC3NP	Res.	CC3P	CC3E	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
rw		rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw

Bits 31:16	保留, 必须保持复位值
Bit 15	CC4NP: 与 CC4P 共同使用, 具体请参考 CC1NP
Bit 14	保留, 必须保持复位值
Bit 13	CC4P: 捕获/比较 4 极性, 参考 CC1P
Bit 12	CC4E: 捕获/比较 4 使能, 参考 CC1E
Bit 11	CC3NP: 与 CC3P 共同使用, 具体请参考 CC1NP
Bit 10	保留, 必须保持复位值
Bit 9	CC3P: 捕获/比较 3 极性, 参考 CC1P
Bit 8	CC3E: 捕获/比较 3 使能, 参考 CC1E
Bit 7	CC2NP: 与 CC2P 共同使用, 具体请参考 CC1NP

Bit 6	保留，必须保持复位值
Bit 5	CC2P : 捕获/比较 2 极性，参考 CC1P
Bit 4	CC2E : 捕获/比较 2 使能，参考 CC1E
Bit 3	CC1NP : 在 CC1 通道作为输入时，该位与 CC1P 共同使用，定义 TI1FP1 和 TI2FP1 的极性，具体请参考 CC1P <i>注：一旦 LOCK 级别设为 3 或 2 (TIM1_BDTR 寄存器中的 LOCK 位) 并且 CC1S=00 (该通道配置成输出) 则该位不能被修改。</i>
Bit 2	保留，必须保持复位值
Bit 1	CC1P : 捕获/比较 1 极性
	CC1 通道作为输出:
	0: OC1 高电平有效 (默认)
	1: OC1 低电平有效
	CC1 通道作为输入:
	CC1NP 和 CC1P 位共同选择 TI1FP1 和 TI2FP1 作为触发或捕获的极性
	00: 信号不反相, TIiFP1 的上升沿有效 (默认)
	01: 信号反相, TIiFP1 的下降沿有效
	10: 保留
	11: 信号不反相, TIiFP1 的上升沿和下降沿均有效; 这个配置不能出现在编码器接口模式中
Bit 0	CC1E : 捕获/比较 1 使能
	CC1 通道作为输出:
	0: 关闭 — OC1 输出禁止 (默认)
	1: 开启 — OC1 信号输出到对应的输出引脚
	CC1 通道作为输入:
	该位决定了计数器的值是否能捕获入 TIM3_CCR1 寄存器
	0: 捕获禁止 (默认)
	1: 捕获使能

14.9.10 CNT 计数寄存器 (TIM3_CNT)

地址偏移: 0x24

复位值: 0x0000_0000

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

Bits 31:16	保留，必须保持复位值
Bits 15:0	CNT[15:0] : 计数器的值

14.9.11 PSC 预分频寄存器 (TIM3_PSC)

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

Bits 31:16	保留, 必须保持复位值
Bits 15:0	PSC[15:0]: 预分频器的值
	计数器的频率可以由下式计算: $f_{CK_CNT} = f_{CK_PSC} / (PSC[15:0] + 1)$ 。 预分频器的值由预装载寄存器写入, 新的预分频器的值在下一次更新事件到来时被采用。可以通过 AS 位 (TIM3_CR1 寄存器) 来选择读取 PSC 寄存器的值来自影子寄存器或预装载寄存器。

14.9.12 ARR 自动重装载寄存器 (TIM3_ARR)

地址偏移: 0x2C

复位值: 0x0000_0000

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

Bits 31:16	保留, 必须保持复位值
Bits 15:0	ARR[15:0]: 自动重装载寄存器的值
	ARR 是要加载到自动重载寄存器中的值。具体请参考 14.2.1 章节。可以通过 AS 位 (TIM3_CR1 寄存器) 来选择读取 ARR 寄存器的值来自影子寄存器或预装载寄存器。

14.9.13 CCR1 捕获/比较寄存器 1 (TIM3_CCR1)

地址偏移: 0x34

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw															

Bits 31:16	保留，必须保持复位值
Bits 15:0	CCR1[15:0] : 捕获/比较 1 的值
	CC1 通道作为输出:
	CCR1 包含了要与计数器进行输出比较的值，它与计数器 TIM3_CNT 的值相比较，并在 OC1 端口上产生输出信号
	如果在 OC1PE 位 (TIM3_CCMR1 寄存器) 未选择预装载功能，写入的数值会立即传输至影子寄存器中；否则只有当更新事件发生时，此预装载值才传输至影子寄存器中
	可以通过 AS 位 (TIM3_CR1 寄存器) 来选择读取 CCR1 寄存器的值来自影子寄存器或预装载寄存器
	CC1 通道作为输入:
	CCR1 包含了上一次输入捕获 1 事件 (IC1) 发生时的计数器值，此时该寄存器为只读

14.9.14 CCR2 捕获/比较寄存器 2 (TIM3_CCR2)

地址偏移: 0x38

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw															

Bits 31:16	保留，必须保持复位值
Bits 15:0	CCR2[15:0] : 捕获/比较 2 的值
	CC2 通道作为输出:
	CCR2 包含了要与计数器进行输出比较的值，它与计数器 TIM3_CNT 的值相比较，并在 OC2 端口上产生输出信号
	如果在 OC2PE 位 (TIM3_CCMR1 寄存器) 未选择预装载功能，写入的数值会立即传输至影子寄存器中；否则只有当更新事件发生时，此预装载值才传输至影子寄存器中
	可以通过 AS 位 (TIM3_CR1 寄存器) 来选择读取 CCR2 寄存器的值来自影子寄存器或预装载寄存器
	CC2 通道作为输入:
	CCR2 包含了上一次输入捕获 2 事件 (IC2) 发生时的计数器值，此时该寄存器为只读

14.9.15 CCR3 捕获/比较寄存器 3 (TIM3_CCR3)

地址偏移: 0x3C

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
rw															

Bits 31:16	保留, 必须保持复位值
Bits 15:0	CCR3[15:0]: 捕获/比较 3 的值
	CC3 通道作为输出:
	CCR3 包含了要与计数器进行输出比较的值, 它与计数器 TIM3_CNT 的值相比较, 并在 OC3 端口上产生输出信号
	如果在 OC3PE 位 (TIM3_CCMR2 寄存器) 未选择预装载功能, 写入的数值会立即传输至影子寄存器中; 否则只有当更新事件发生时, 此预装载值才传输至影子寄存器中
	可以通过 AS 位 (TIM3_CR1 寄存器) 来选择读取 CCR3 寄存器的值来自影子寄存器或预装载寄存器
	CC3 通道作为输入:
	CCR3 包含了上一次输入捕获 3 事件 (IC3) 发生时的计数器值, 此时该寄存器为只读

14.9.16 CCR4 捕获/比较寄存器 4 (TIM3_CCR4)

地址偏移: 0x40

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
rw															

Bits 31:16	保留, 必须保持复位值
Bits 15:0	CCR4[15:0]: 捕获/比较 4 的值
	CC4 通道作为输出:
	CCR4 包含了要与计数器进行输出比较的值, 它与计数器 TIM3_CNT 的值相比较, 并在 OC4 端口上产生输出信号
	如果在 OC4PE 位 (TIM3_CCMR2 寄存器) 未选择预装载功能, 写入的数值会立即



	传输至影子寄存器中；否则只有当更新事件发生时，此预装载值才传输至影子寄存器中
	可以通过 AS 位（TIM3_CR1 寄存器）来选择读取 CCR4 寄存器的值来自影子寄存器或预装载寄存器
	CC4 通道作为输入：
	CCR4 包含了上一次输入捕获 4 事件（IC4）发生时的计数器值，此时该寄存器为只读

14.9.17 DCR DMA 控制寄存器（TIM3_DCR）

地址偏移：0x48

复位值：0x0000_0000

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

Bits 31:13	保留，必须保持复位值
Bits 12:8	DBL[4:0]: DMA burst 传输长度
	定义了 DMA 在 burst 模式下的传输长度（当对 TIM3_DMAR 寄存器的地址进行读或写时，TIM3 进行一次 burst 传输），即定义被传送的字节数目
	00000: 1 字节（默认）
	00001: 2 字节
	00010: 3 字节
	...
	10001: 18 字节
Bits 7:5	DSEL[2:0]: DMA burst 传输请求选择
	定义了哪个 DMA 请求使用 burst 传输，其它的未选中的则使用 single 传输
	000: 所有 DMA 请求都使用 single 传输
	001: DMA 请求 0 使用 burst 传输
	010: DMA 请求 1 使用 burst 传输
	011: DMA 请求 2 使用 burst 传输
	100: DMA 请求 3 使用 burst 传输
	101 至 111: 保留
Bits 4:0	DBA[4:0]: DMA 基地址
	该位定义了 DMA 在 burst 模式下的基地址（当对 TIM3_DMAR 寄存器的地址进行读或写时），DBA 定义为从 TIM3_CR1 寄存器所在地址开始的偏移量
	00000: TIM3_CR1
	00001: TIM3_CR2
	00010: TIM3_SMCR

	...
	举例：当 DBL=7 且 DBA=00000 时，表示从 TIM3_CR1 开始向下读取或发送 7 个寄存器的数据。

表 14-4 TIMx DMA 请求表

	TIM1	TIM3	TIM6	TIM14	TIM15	TIM16	TIM17
DMA 请求 0	TIM1_CH1	TIM3_CH1 TIM3_TRIG	TIM6_UP	N/A	TIM15_CH1 TIM15_CH2 TIM15_UP TIM15_TRIG TIM15_COM	TIM16_CH1 TIM16_UP TIM16_TRIG TIM16_COM	TIM17_CH1 TIM17_UP TIM17_TRIG TIM17_COM
DMA 请求 1	TIM1_CH2	TIM3_CH2 TIM3_CH3	N/A	N/A	N/A	N/A	N/A
DMA 请求 2	TIM1_CH3 TIM1_UP	TIM3_CH4 TIM3_UP	N/A	N/A	N/A	N/A	N/A
DMA 请求 3	TIM1_CH4 TIM1_TRIG TIM1_COM	N/A	N/A	N/A	N/A	N/A	N/A

注：TIM6 仅支持 DMA single 传输。

14.9.18 DMAR DMA 传输寄存器 (TIM3_DMAR)

地址偏移：0x4C

复位值：0x0000_0000

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

Bits 31:0	DMAB[31:0]: DMA 传输寄存器
	对 TIM3_DMAR 寄存器的读或写会导致对以下地址的寄存器的读写操作：
	TIM3_CR1 地址 + (DBA + DMA 指针) x4，其中“TIM3_CR1 地址”是 TIM3 寄存器的起始地址；DBA 是 TIM3_DCR 寄存器中定义的基地址；DMA 指针是由 DMA 自动控制的偏移量，它的范围是从 0 至 TIM3_DCR 寄存器中定义的 DBL。

14.9.19 SysCtrl_EDU_CFG2 寄存器

具体请参考 6.3.3 互联配置寄存器 2 (SysCtrl_EDU_CFG2) 的内容。

14.9.20 SysCtrl_EDU_CFG5 寄存器

具体请参考 6.3.6 互联配置寄存器 5 (SysCtrl_EDU_CFG5) 的内容。

领芯微志

15.通用定时器 TIM14

LCM32F037 内置有 5 个可同步的通用定时器，具体差异参见上面第 13 章高级定时器 TIM1 的表 13-1。各种定时器的特性区别主要在计数方向、是否具有 DMA 请求、捕获/比较通道数量以及互补输出通道数量上。

15.1 TIM14 主要特性

通用定时器 TIM14 是一个 16 位的定时器/计数器，由一个可编程的预分频器驱动，有一路捕获/比较通道。主要用于基本定时，测量输入信号的脉冲宽度（输入捕获）产生输出波形（输出比较和 PWM），对应于不同事件（捕获、比较和溢出）的中断。

TIM14 的 1 个独立的通道可以被用于：

- 输入捕获
- 输出比较
- PWM生成（边沿模式）

TIM14 没有互补输出，也没有 DMA 功能。在调试模式下，计数器可以被冻结。TIM14 可以通过定时器联动机制与其他定时器共同工作，提供同步或事件联接功能。

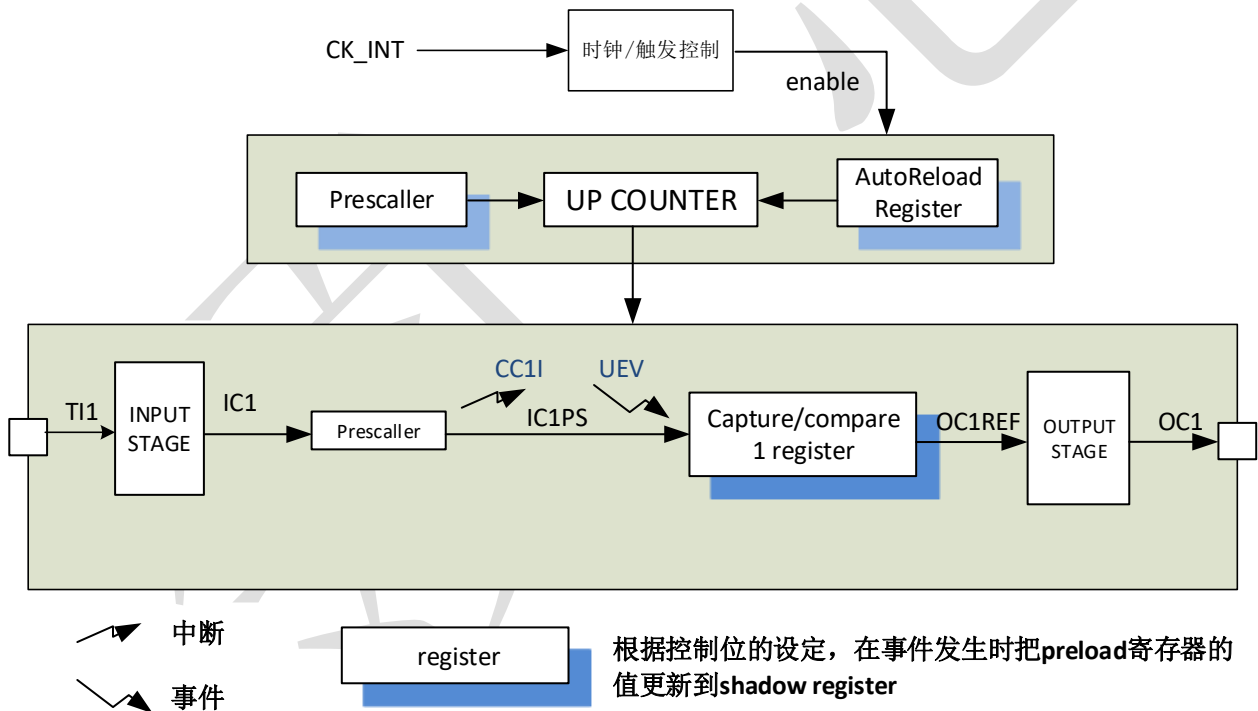


图 15-1 通用定时器 TIM14 框图

15.2 时基单元

时基单元包含：

- 16 位向上计数器（TIM14_CNT）
- 16 位预分频器（TIM14_PSC）
- 16 位自动重载寄存器（TIM14_ARR）

16 位计数器、预分频器和自动重载寄存器都可以通过软件进行读写操作。计数器由预分频器的输出 CK_CNT 驱动，而 CK_CNT 仅在 TIM14_CR1 寄存器的计数器使能位 (CEN) 被置位时才有效。

注：在使能了 CEN 位的一个时钟周期后，计数器才开始计数。

写计数器的操作没有缓存，可以在任何时候写 TIM14_CNT 寄存器，因此建议不要在计数器运行时写入新的数值，以免写入了错误的数值。

15.2.1 自动装载寄存器

自动装载寄存器是预先装载的。写或读自动重载寄存器将访问预装载寄存器。根据在 TIM14_CR1 寄存器中的自动装载预装载使能位 (ARPE) 的设置，预装载寄存器的内容被立即或在每次的更新事件 UEV 时传送到影子寄存器。当计数器达到溢出条件并当 TIM14_CR1 寄存器中的 UDIS=0 时，产生更新事件。更新事件也可以由软件产生，随后会详细描述每一种配置下更新事件的产生。注意这里影子寄存器 (Shadow register) 即为有效的工作寄存器 (Active register)。

写自动重载寄存器的两种模式：

- 自动预装载已使能 (TIM14_CR1 寄存器的 ARPE 位置位)。在此模式下，写入自动重载寄存器的数据将被保存在预装载寄存器中，并在下一个更新事件 (UEV) 时传送到影子寄存器。
- 自动预装载已禁止 (TIM14_CR1 寄存器的 ARPE 位清除)。在此模式下，写入自动重载寄存器的数据将立即写入影子寄存器。

更新事件的产生条件：

- 计数器向上溢出
- 软件置位了 TIM14_EGR 寄存器的 UG 位

在预装载使能时 (ARPE=1)，如果发生了更新事件，预装载寄存器中的数值 (TIM14_ARR) 将写入影子寄存器中。置位 TIM14_CR1 寄存器的 UDIS 位将禁止更新事件 (UEV)。

15.2.2 预分频器

TIM14 的预分频器基于一个由 16 位寄存器 (TIM14_PSC) 控制的 16 位计数器。由于这个控制寄存器带有缓冲器，因此它能够在运行时被改变。预分频器可以将计数器的时钟频率按 1 到 65536 之间的任意值分频。计数器的频率可以由下式计算：

$$f_{CK_CNT} = f_{CK_PSC} / (PSC[15:0] + 1)$$

预分频器的值由预装载寄存器写入，新的预分频器的值在下一次更新事件到来时被采用。对 TIM14_PSC 寄存器的读操作通过预装载寄存器完成，因此不需要特别的关注。

下图给出了在预分频器工作时，更改其参数的情况下计数器操作的例子，其它分频系数类推。

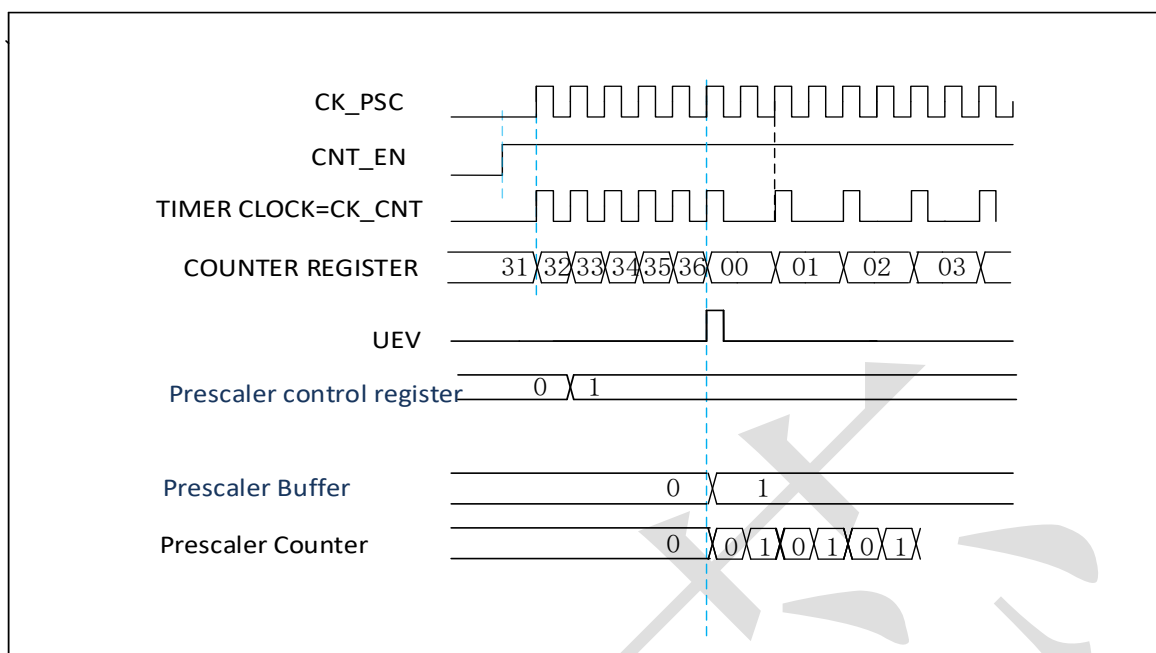


图 15-2 预分频系数从 1 变 2 的计数器时序

15.3 计数器模式

TIM14 只支持向上计数模式。在向上计数模式中，计数器从 0 计数到用户定义的比较值（TIMx_ARR 寄存器的值），然后重新从 0 开始计数并产生一个计数器溢出事件，同时，如果 TIMx_CR1 寄存器的 UDIS 位是 0，将会产生一个更新事件（UEV）。

置位 TIMx_EGR 寄存器的 UG 位（通过软件方式或者使用从模式控制器）也同样可以产生一个更新事件。

使用软件置位 TIMx_CR1 寄存器的 UDIS 位，可以禁止更新事件，这样可以避免在更新预装载寄存器时更新影子寄存器。在 UDIS 位被清除之前，将不产生更新事件。但是在应该产生更新事件时，计数器仍会被清 0，同时预分频器的计数也被清 0（但预分频器的数值不变）。此外，如果设置了 TIMx_CR1 寄存器中的 URS 位（选择更新请求源），设置 UG 位将产生一个更新事件 UEV，但硬件不设置 UIF 标志（即不产生中断请求）。这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。

当发生一个更新事件时，所有的寄存器都被更新，硬件同时依据 URS 位设置更新标志位（TIMx_SR 寄存器的 UIF 位）：自动装载影子寄存器被重新置入预装载寄存器的值（TIMx_ARR）。预分频器的缓存器被置入预装载寄存器的值（TIMx_PSC）。要注意到如果是因为计数器溢出而产生更新，自动重装载寄存器将在计数器重载入之前被更新，因此下一个计数周期才是预期的值（计数器被装载为新的值）。非自动重装载模式下可以立即更新。

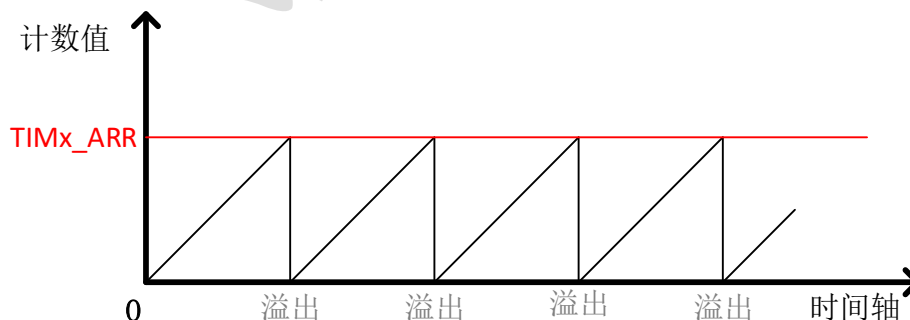


图 15-3 向上计数溢出

下图给出例子，当 TIMx_ARR=0x36 时计数器在二分频时钟频率下的动作。

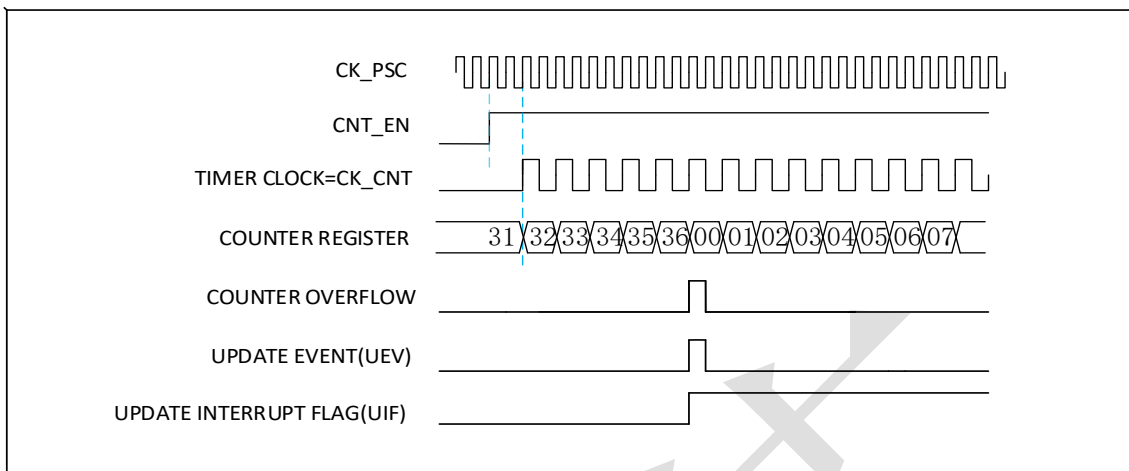


图 15-4 计数器在二分频时钟频率下的时序图

15.4 时钟选择

计数器时钟由内部时钟 (CK_INT) 提供，CEN、DIR 和 UG 位是实际上的控制位，并且只能被软件修改 (UG 位仍被自动清除)。一旦 CEN 位被写成 1，预分频器的时钟就由内部时钟提供。

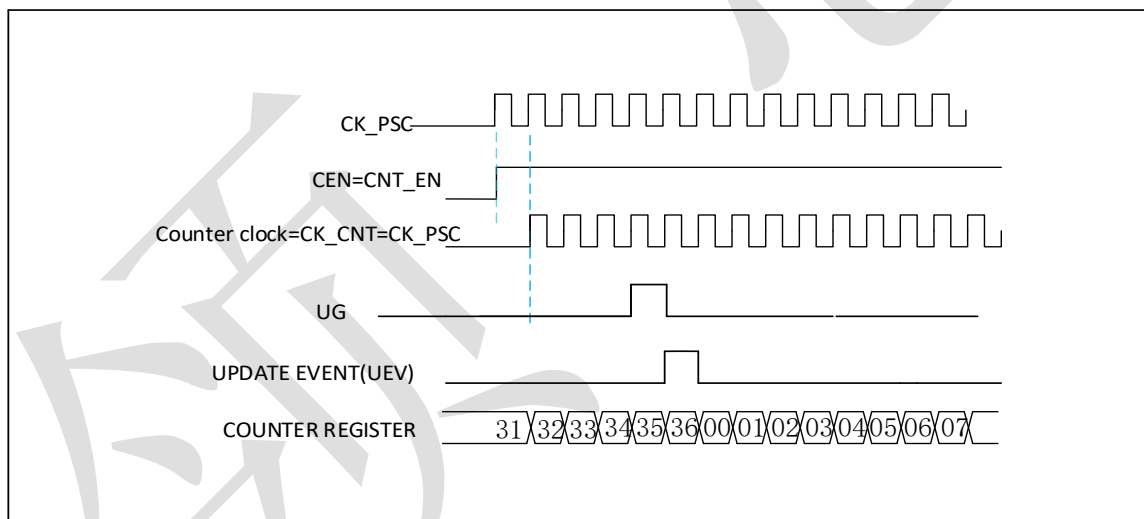


图 15-5 内部时钟模式

15.5 捕获/比较通道

定时器的 I/O 引脚 (TIMx_CCI) 可以用作输入捕获或者输出比较，这个功能可以通过配置捕获/比较通道模式寄存器 (TIMx_CCMRi) 的 CCIS 通道选择位来实现，此处的 i 代表通道数。

每一个捕获/比较通道都是围绕着一个捕获/比较寄存器 (包含影子寄存器) 来构建的，包括捕获的输入部分 (数字滤波、多路复用和预分频器)，和输出部分 (比较器和输出控制)。

捕获/比较模块由一个预装载寄存器和一个影子寄存器组成，读写过程仅操作预装载寄存器。在捕获模式下，捕获计数器的值就在影子寄存器上，然后再复制到预装载寄存器中。在比较模式下，预装载寄存器的内容被复制到影子寄存器中，然后影子寄存器内容和计数器进行比较。

当通道被配置成输出模式时 (TIMx_CCMRi 寄存器的 CCIS=0)，可以随时访问 TIMx_CCRi 寄存器 (此处的

i 指通道数)。

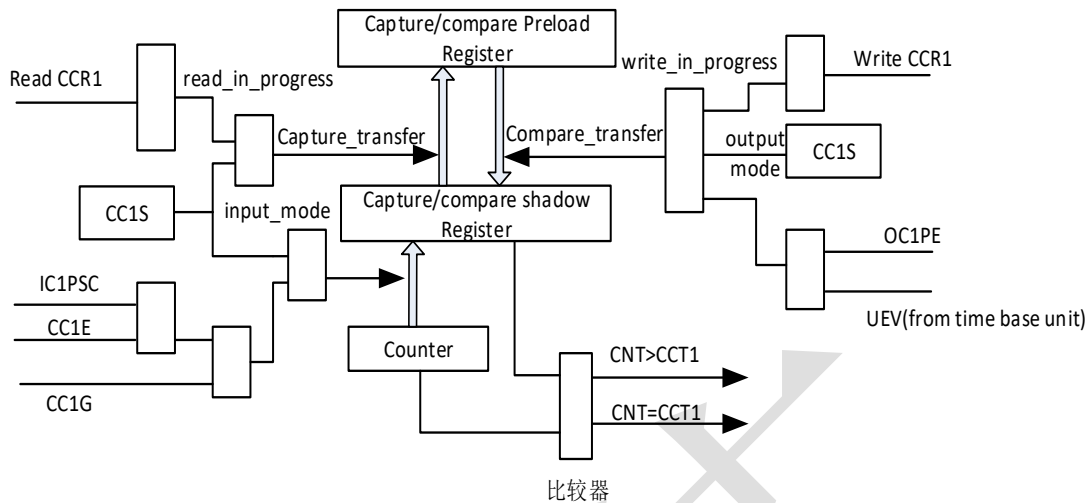


图 15-6 捕获/比较通道 1 的主电路

当通道被配置成输入模式时，对 $TIMx_CCRi$ 寄存器的读操作类似于计数器的读操作。当捕获发生时，计数器的内容被捕获到 $TIMx_CCRi$ 影子寄存器，随后再复制到预装载寄存器中。在读操作进行中，预装载寄存器是被冻结的（以防读数据过程中预装载寄存器内容被修改）。

15.5.1 输入捕获模式

在输入捕获模式下，当检测到 ICi 信号上相应的边沿后，计数器的当前值被锁存到捕获/比较寄存器（ $TIMx_CCRi$ ）中。当发生捕获事件时，相应的 $CCiIF$ 标志（ $TIMx_SR$ 寄存器）被置 1。

如果 $TIMx_DIER$ 寄存器的 $CCiIE$ 位被置位，也就是使能了中断，则将产生中断请求。如果发生捕获事件时 $CCiIF$ 标志已经为高，那么重复捕获标志 $CCiOF$ （ $TIMx_SR$ 寄存器）被置 1，写 $CCiIF=0$ 可清除 $CCiOF$ 。写 $CCiIF=0$ 或读取存储在 $TIMx_CCRi$ 寄存器中的捕获数据都可清除 $CCiIF$ 。

以下例子说明如何在 $TI1$ 输入的上升沿时捕获计数器的值到 $TIMx_CCR1$ 寄存器中，步骤如下：

1. 选择有效输入端：例如 $TIMx_CCR1$ 连接到 $TI1$ 输入，所以写入 $TIMx_CCMR1$ 寄存器中的 $CC1S=01$ ，此时通道被配置为输入，并且 $TIMx_CCR1$ 寄存器变为只读。
2. 根据输入信号 TIi 的特点，可通过配置 $TIMx_CCMRi$ 寄存器中的 $ICiF$ 位来设置相应的输入滤波器的滤波时间。假设输入信号在最多 5 个时钟周期的时间内抖动，我们须配置滤波器的带宽长于 5 个时钟周期；因此我们可以连续采样 8 次，以确认在 $TI1$ 上一次真实的边沿变换，即在 $TIMx_CCMR1$ 寄存器中写入 $IC1F=0011$ ，此时，只有连续采样到 8 个相同的 $TI1$ 信号，信号才为有效（采样频率为 DTS 时钟的频率 f_{DTS} ）。
3. 选择 $TI1$ 通道的有效转换边沿，在 $TIMx_CCER$ 寄存器中写入 $CC1P=0$ （上升沿）。
4. 配置输入预分频器。在本例中，我们希望捕获发生在每一个有效的电平转换时刻，因此预分频器被禁止（写 $TIMx_CCMR1$ 寄存器的 $IC1PSC=00$ ）。
5. 设置 $TIMx_CCER$ 寄存器的 $CC1E=1$ ，允许捕获计数器的值到捕获寄存器中。
6. 如果需要，通过设置 $TIMx_DIER$ 寄存器中的 $CC1IE$ 位允许相关中断请求。

当发生一个输入捕获时：

- 当产生有效的电平转换时，计数器的值被传送到 $TIMx_CCR1$ 寄存器。
- $CC1IF$ 标志被设置（中断标志）。当发生至少 2 次连续的捕获，且 $CC1IF$ 未被清除， $CC1OF$ 也被置 1。

- 如设置了 CC1IE 位，则会产生一个中断。
- 为了处理捕获溢出（CC1OF 位），建议在读出重复捕获标志之前读取数据，这是为了避免丢失在读出捕获溢出标志之后和读取数据之前可能产生的重复捕获信息。

注：设置 TIMx_EGR 寄存器中相应的 CCiG 位，可以通过软件产生输入捕获中断。

15.5.2 强制输出模式

在输出模式（TIM14_CCMRi 寄存器中 CCiS=00）下，输出比较信号能够直接由软件强置为高或低状态，而不依赖于输出比较寄存器和计数器间的比较结果。

置 TIMx_CCMRi 寄存器中相应的 OCiM=101，即可强置输出比较信号为有效状态。这样 OCiREF 被强置为高电平（OCiREF 始终为高电平有效），而 OCi 的输出是高还是低则取决于 CCiP 极性标志位。例如 CCiP=0（OCi 高电平有效），则 OCi 被强置为高电平。

置 TIMx_CCMRi 寄存器的 OCiM=100，可强置 OCiREF 信号为低。

该模式下，在 TIMx_CCRi 影子寄存器和计数器之间的比较仍然在进行，相应的标志也会被修改，也仍然会产生相应的中断。这将会在下面的输出比较模式一节中介绍。

15.5.3 输出比较模式

此模式用来控制一个输出波形或者指示一段给定的时间已经达到。当计数器与捕获/比较寄存器的内容相同时，有如下操作：

- 根据不同的输出比较模式，相应的 OCiREF 输出信号为：
 - 保持不变（OCiM=000）
 - 设置为有效电平（OCiM=001）
 - 设置为无效电平（OCiM=010）
 - 翻转（OCiM=011）
- 设置中断状态寄存器中的标志位（TIMx_SR 寄存器中的 CCiIF 位）
- 若设置了相应的中断使能位（TIMx_DIER 寄存器中的 CCiE 位），则产生一个中断

TIMx_CCMRi 寄存器的 OCiM 位用于选择输出比较模式，而 TIMx_CCER 寄存器的 CCiP 位用于选择有效和无效的电平极性。

TIMx_CCMRi 寄存器的 OCiPE 位用于选择 TIMx_CCRi 寄存器是否需要使用预装载寄存器。

在输出比较模式下，更新事件 UEV 对 OCiREF 和 OCi 输出没有影响。输出比较的时间精度为计数器的一个时钟周期，输出比较模式也能用来输出一个单脉冲。输出比较模式的配置步骤如下：

1. 选择计数器时钟（内部/外部/预分频器）
2. 将相应的数据写入 TIMx_ARR 和 TIMx_CCRi 寄存器中
3. 如果要产生一个中断请求，设置 CCiE 位
4. 选择输出模式步骤：
 - 要求计数器与 CCRi 匹配时翻转 OCiM 的输出管脚，设置 OCiM=011
 - 置 OCiPE = 0 禁用预装载寄存器
 - 置 CCiP = 0 选择高电平为有效电平
 - 置 CCiE = 1 使能输出
5. 设置 TIMx_CR1 寄存器的 CEN 位来启动计数器

在未使用预装载寄存器（OCiPE=0，否则 TIMx_CCRi 的影子寄存器只能在发生下一次更新事件时被更新）的情况下，TIMx_CCRi 寄存器能够在任何时候通过软件进行更新以控制输出波形。

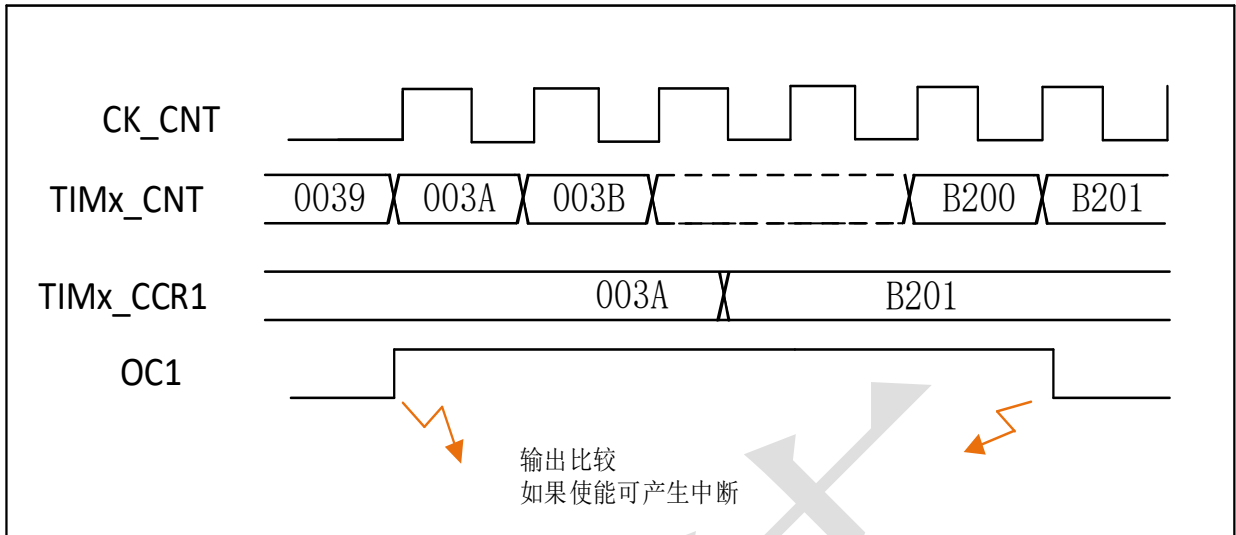


图 15-7 输出比较模式，翻转 OC1

15.5.4 PWM 模式

脉冲宽度调制（PWM）模式可以产生一个由 TIMx_ARR 寄存器确定频率，由 TIMx_CCRi 寄存器确定占空比的信号。

在 TIMx_CCMRi 寄存器中的 OCiM 位写入 110（PWM 模式 1）或 111（PWM 模式 2），能够独立地设置每个 OCi 输出通道产生一路 PWM。

必须设置 TIMx_CCMRi 寄存器的 OCiPE 位使能相应的预装载寄存器，也可以设置 TIMx_CR1 寄存器的 ARPE 位使能自动重载的预装载寄存器（在向上计数模式或中央对称模式中）。

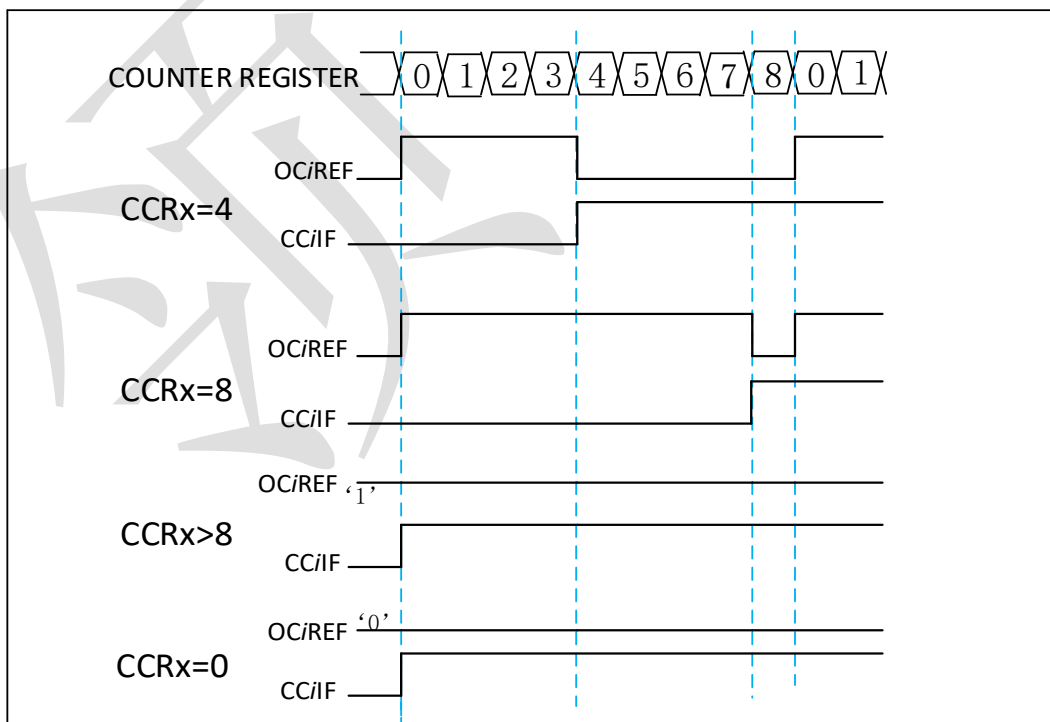


图 15-8 TIMx_ARR=8 时边沿对齐的 PWM 波形实例

由于仅当发生一个更新事件的时候，预装载寄存器才能被传送到影子寄存器，因此在计数器开始计数之前，必须通过设置 TIMx_EGR 寄存器的 UG 位来初始化所有的寄存器。

OCi 的极性可以通过软件在 TIMx_CCER 寄存器中的 CCiP 位设置，它可以设置为高电平有效或低电平有效。OCi 的输出使能通过 TIMx_CCER 中 CCiE 位控制，详见 TIMx_CCER 寄存器的描述。

在 PWM 模式（模式 1 或模式 2）下，TIMx_CNT 和 TIMx_CCRi 始终在进行比较，（依据计数器的计数方向）以确定是否符合 $TIMx_CCRi \leq TIMx_CNT$ 或者 $TIMx_CNT \leq TIMx_CCRi$ 。

TIM14 只能产生边沿对齐的 PWM 信号（因为计数器只有向上计数模式）。

当 TIMx_CR1 寄存器中的 DIR 位为低的时候执行向上计数。图 5-8 是一个 PWM 模式 1 的例子。当 $TIMx_CNT < TIMx_CCRi$ 时，PWM 参考信号 OCiREF 为高，否则为低。如果 TIMx_CCRi 中的比较值大于自动重载值（TIMx_ARR），则 OCiREF 保持为 1。如果比较值为 0，则 OCiREF 保持为 0。下图为 TIMx_ARR=8 时边沿对齐的 PWM 波形实例。

15.6 TIM14 与其他定时器的联接

在芯片中，各个定时器在内部互相联接，用于定时器的同步或连接。TIM14 的没有内部触发 ITR 和外部触发 ETR，只有外部输入 TI1。其中，外部输入 TI 的来源为 tim14_gpio_ti1、WT 时钟、OSCH 时钟的分频时钟和 MCO，通过 TIM14_OR 寄存器配置。可以通过 TIM14 的输入通道 TI1 进行各个时钟源的在线测量和校准。TIM14 的特殊输出 TIM14_oc1_trig 可触发其他定时器，它与 CEN 位相关，具体详见 TIM14_CR2 寄存器 TrgiS 位说明。

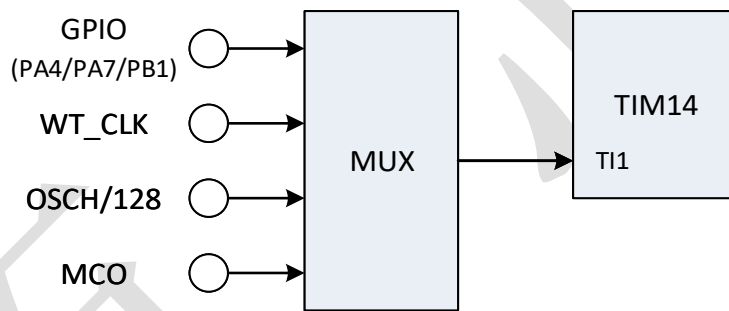


图 15-9 时钟测量和校准

15.7 TIM14 中断

TIM14 有 2 个中断请求源，分别映射到 2 个中断矢量上：

- 更新事件中断
- 输入捕获/输出比较 1 中断

为了使用中断特性，对每个被使用的中断通道，设置 TIMx_DIER 寄存器中相应的中断使能位：CCiE 和 UIE 位。

通过设置 TIMx_EGR 寄存器中的相应位，也可以用软件产生上述各个中断源。

15.8 TIM14 寄存器描述

表 15-1 TIM14 相关寄存器表

名称	说明	读写权限	复位值	字节地址
CR1	控制寄存器 1	R/W	0x0000_0000	0x4000_A000
CR2	控制寄存器 2	R/W	0x0000_0000	0x4000_A004
DIER	中断控制寄存器	R/W	0x0000_0000	0x4000_A00C

SR	状态寄存器	R/W	0x0000_0000	0x4000_A010
EGR	事件产生寄存器	R/W	0x0000_0000	0x4000_A014
CCMR1	捕获/比较模式寄存器 1	R/W	0x0000_0000	0x4000_A018
CCER	捕获比较使能寄存器	R/W	0x0000_0000	0x4000_A020
CNT	计数寄存器	R/W	0x0000_0000	0x4000_A024
PSC	预分频寄存器	R/W	0x0000_0000	0x4000_A028
ARR	自动重装载寄存器	R/W	0x0000_0000	0x4000_A02C
CCR1	捕获/比较寄存器 1	R/W	0x0000_0000	0x4000_A034
OR	选项寄存器	R/W	0x0000_0000	0x4000_A050

注：x 表示不确定；-表示未实现；u 表示不受复位影响；c 表示该位复位值因复位源而异；R 只读；W 只写；R-0 表示只读且读0；R/W 表示可读写（以后章节同上述）。

15.8.1 CR1 控制寄存器 1 (TIM14_CR1)

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
Res.							AS	OCF	Res.				TI1E	Res.	FTE
							rw	rw					rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						CKD[1:0]		ARPE	Res.				URS	UDIS	CEN
						rw	rw						rw	rw	rw

Bits	31:25	保留，必须保持复位值
Bit	24	AS: 影子寄存器读使能 该位指定当 MCU 读取 TIM14_PSC/TIM14_ARR/TIM14_CCRi 寄存器的时候，是读取影子寄存器还是预装载寄存器 0: 选择预装载寄存器的值（默认） 1: 选择影子寄存器的值
Bit	23	OCF: 输出异步抗干扰滤波 0: 输出滤波禁止（默认） 1: 输出滤波使能
Bits	22:19	保留，必须保持复位值
Bit	18	TI1E: TI1 输入检测/滤波使能 0: TI1 输入检测/滤波禁止（默认） 1: TI1 输入检测/滤波使能
Bit	17	保留，必须保持复位值
Bit	16	FTE: 数字滤波器 (Tli) 的 DTS 时钟使能 0: 数字滤波器的 DTS 时钟禁止（默认） 1: 数字滤波器的 DTS 时钟使能
Bits	15:10	保留，必须保持复位值
Bits	9:8	CKD[1:0]: CK_INT 时钟和采样时钟 (CK_DTS) 的分频系数，DTS 时钟供给数字滤波

	器 (Tli) 使用
	00: $t_{DTS} = t_{CK_INT}$ (默认)
	01: $t_{DTS} = 2 * t_{CK_INT}$
	10: $t_{DTS} = 4 * t_{CK_INT}$
	11: 保留
Bit 7	ARPE: 自动预装载允许位
	0: TIM14_ARR 寄存器可以被直接写入 (默认)
	1: TIM14_ARR 寄存器通过预装载寄存器更新
Bits 6:3	保留, 必须保持复位值
Bit 2	URS: 更新请求源
	0: 如果 UDIS 允许产生更新事件, 则下述任一事件产生一个更新中断 (默认):
	- 计数器上溢
	- 软件设置 UG 位
	1: 如果 UDIS 允许产生更新事件, 则只有当计数器上溢时才产生更新中断
Bit 1	UDIS: 禁止更新
	0: 一旦下列事件发生, 产生更新事件 (默认):
	- 计数器溢出
	- 软件设置 UG 位
	1: 不产生更新事件, 影子寄存器 (ARR、PSC、CCR) 保持它们的值。如果 UG 位被配置, 则计数器和预分频器被重新初始化
Bit 0	CEN: 计数器使能位
	0: 计数器禁止 (默认)
	1: 计数器使能

15.8.2 CR2 控制寄存器 2 (TIM14_CR2)

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
Res.															TrigS
															rw

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

Bits 31:17	保留, 必须保持复位值
Bit 16	TrigS: TIMx_oc1_trig 触发源选择
	0: 触发源选择为 OC1&CEN (默认)
	1: 触发源选择为 CEN
Bits 15:0	保留, 必须保持复位值

15.8.3 DIER 中断控制寄存器 (TIM14_DIER)

地址偏移: 0x0C

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.														CC1IE	UIE
														rw	rw

Bits	31:2	保留, 必须保持复位值
Bit	1	CC1IE: 捕获/比较 1 中断使能
		0: 捕获/比较 1 中断禁止 (默认)
		1: 捕获/比较 1 中断使能
Bit	0	UIE: 更新中断使能
		0: 更新中断禁止 (默认)
		1: 更新中断使能

15.8.4 SR 状态寄存器 (TIM14_SR)

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
Res.															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						CC1OF	Res.						CC1IF	UIF	
						rw							rw	rw	

Bits	31:10	保留, 必须保持复位值
Bit	9	CC1OF: 捕获/比较 1 过捕获/过比较标志
		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, 软件向 CC1IF 位写 0 可清除该位
		0: 无过捕获产生 (默认)
		1: 计数器的值被捕获到 TIM14_CCR1 寄存器时 CC1IF 已经置 1
Bits	8:2	保留, 必须保持复位值
Bit	1	CC1IF: 捕获/比较 1 中断标志
		如果通道 CC1 配置为输出模式:
		当计数器值与比较值匹配时该位由硬件置 1, 软件写 0 可清除该位, 但是当 CC1OF 也为 1 时, 需要清除 2 次。



	0: 无匹配发生 (默认)
	1: TIM14_CNT 的值与 TIM14_CCR1 的值匹配
	如果通道 CC1 配置为输入模式:
	当捕获事件发生时该位由硬件置 1, 软件写 0 或通过读 TIM14_CCR1 可清除该位, 但是当 CC1OF 也为 1 时, 需要清除 2 次。
	0: 无输入捕获产生 (默认)
	1: 计数器值已被捕获至 TIM14_CCR1 (在 IC1 上检测到与所选极性相同的边沿)
Bit 0	UIF : 更新中断标志
	当产生更新事件时该位由硬件置 1, 软件写 0 可清除该位
	0: 无更新事件产生 (默认)
	1: 更新中断挂起, 当相关寄存器被更新时该位由硬件置 1
	- 若 TIM14_CR1 寄存器的 UDIS=0, 当计数器上溢时
	- 若 TIM14_CR1 寄存器的 UDIS=0、URS=0, 当软件设置 TIM14_EGR 寄存器的 UG 位对计数器 CNT 重新初始化时
	- 若 TIM14_CR1 寄存器的 UDIS=0、URS=0, 当计数器 CNT 被触发事件重新初始化时

15.8.5 EGR 事件产生寄存器 (TIM14_EGR)

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
Res.															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.														CC1G	UG
														rw	rw

Bits 31:2	保留, 必须保持复位值
Bit 1	CC1G : 产生捕获/比较 1 事件
	该位由软件置 1, 用于产生一个捕获/比较事件, 由硬件自动清 0
	0: 无动作 (默认)
	1: 在通道 CC1 上产生一个捕获/比较事件
	若通道 CC1 配置为输出:
	设置 CC1IF=1, 若 CC1IE=1, 则产生相应的中断; 若 CC1IF 已经为 1, 则设置 CC1OF=1
	若通道 CC1 配置为输入:
	当前的计数器值被捕获至 TIM14_CCR1 寄存器, 设置 CC1IF=1, 若 CC1IE=1, 则产生相应的中断; 若 CC1IF 已经为 1, 则设置 CC1OF=1
Bit 0	UG : 产生更新事件
	该位由软件置 1, 用于产生一个更新事件, 由硬件自动清 0
	0: 无动作 (默认)
	1: 重新初始化计数器, 并产生一个更新事件。注意预分频器的计数器也被清 0 (但

是预分频系数不变)

15.8.6 CCMR1 捕获/比较模式寄存器 1 (TIM14_CCMR1)

地址偏移: 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 CCIS 位定义。该寄存器部分位的作用在输入和输出模式下不同。OCxx 描述了通道在输出模式下的功能, ICxx 描述了通道在输入模式下的功能。因此必须注意, 同一个寄存器位在输出模式和输入模式下的功能是不同的。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								OC1CE	OC1M[2:0]			OC1PE	Res.	CC1S	
								IC1F[3:0]			IC1PSC[1:0]		[1:0]		
								rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式

Bits 31:8	保留, 必须保持复位值
Bit 7	OC1CE: 输出比较 1 清零使能
	0: OC1REF 清除禁止 (默认)
	1: OC1REF 清除使能
Bits 6:4	OC1M[2:0]: 输出比较 1 模式选择
	这 3 位定义了输出参考信号 OC1REF 的动作, 而 OC1REF 决定了 OC1 的值。OC1REF 是高电平有效, 而 OC1 的有效电平取决于 CC1P。
	000: 冻结。输出比较寄存器 TIM14_CCR1 与计数器 TIM14_CNT 间的比较对 OC1REF 不起作用。(默认)
	001: 匹配时设置通道 1 的输出为有效电平。当计数器 TIM14_CNT 的值与捕获/比较寄存器 1 (TIM14_CCR1) 相同时, 强制 OC1REF 为高。
	010: 匹配时设置通道 1 的输出为无效电平。当计数器 TIM14_CNT 的值与捕获/比较寄存器 1 (TIM14_CCR1) 相同时, 强制 OC1REF 为低。
	011: 翻转。当 TIM14_CCR1=TIM14_CNT 时, 翻转 OC1REF 的电平。
	100: 强制为无效电平。强制 OC1REF 为低。
	101: 强制为有效电平。强制 OC1REF 为高。
	110: PWM 模式 1 — 在向上计数时, 一旦 TIM14_CNT<TIM14_CCR1 时通道 1 为有效电平, 否则为无效电平。
	111: PWM 模式 2 — 在向上计数时, 一旦 TIM14_CNT<TIM14_CCR1 时通道 1 为无效电平, 否则为有效电平。
	<i>注: 在 PWM 模式 1 或 PWM 模式 2 中, 只有当比较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时, OC1REF 电平才改变。(参考章节 15.5.4 PWM 模式)。</i>
Bit 3	OC1PE: 输出比较 1 预装载使能
	0: 禁止 TIM14_CCR1 寄存器的预装载功能, 可随时写入 TIM14_CCR1 寄存器, 并且新写入的数值立即起作用 (默认)

	1: 使能 TIM14_CCR1 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIM14_CCR1 的预装载值在更新事件到来时被加载至当前寄存器中
	<i>注: 为了操作正确, 在 PWM 模式下必须使能预装载功能。</i>
Bit 2	保留, 必须保持复位值
Bits 1:0	CC1S[1:0]: 捕获/比较 1 选择
	该位定义通道的方向 (输入/输出), 以及输入脚的选择
	00: CC1 通道被配置为输出 (默认)
	01: CC1 通道被配置为输入, IC1 映射在 TI1FP1 上
	10: 保留
	11: 保留
	<i>注: CC1S 仅在通道关闭时 (TIM14_CCER 寄存器的 CC1E=0 且已被更新) 才是可写的。</i>

输入捕获模式

Bits 31:8	保留, 必须保持复位值
Bits 7:4	IC1F[3:0]: 输入捕获 1 滤波器
	该位域定义了 TI1 输入的采样频率及数字滤波器的长度。数字滤波器由一个事件计数器组成, 只有发生了 N 个连续事件后输出的跳变才被认为有效。这里的 DTS 时钟是 CK_INT 经过 CR1 寄存器中 CKD 配置分频的。
	0000: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}$, 无滤波器 (默认)
	0001: 采样频率 $f_{\text{SAMPLING}} = f_{\text{CK_INT}}$, N=2
	0010: 采样频率 $f_{\text{SAMPLING}} = f_{\text{CK_INT}}$, N=4
	0011: 采样频率 $f_{\text{SAMPLING}} = f_{\text{CK_INT}}$, N=8
	0100: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/2$, N=6
	0101: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/2$, N=8
	0110: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$, N=6
	0111: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$, N=8
	1000: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/8$, N=6
	1001: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/8$, N=8
	1010: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$, N=5
	1011: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$, N=6
	1100: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$, N=8
	1101: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$, N=5
	1110: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$, N=6
	1111: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$, N=8
	<i>注: 请注意采样频率中 $f_{\text{CK_INT}}$ 和 f_{DTS} 的差别。</i>
Bits 3:2	IC1PSC[1:0]: 输入捕获 1 预分频器
	IC1 的预分频系数, 一旦 CC1E=0 (TIM14_CCER 寄存器中), 则预分频器复位
	00: 无预分频器, TI1 上每 1 个有效边沿作为一次输入 (默认)
	01: TI1 上每 2 个有效边沿作为一次输入
	10: TI1 上每 4 个有效边沿作为一次输入
	11: TI1 上每 8 个有效边沿作为一次输入
Bits 1:0	CC1S[1:0]: 捕获/比较 1 选择
	该位定义通道的方向 (输入/输出), 以及输入脚的选择

	00: CC1 通道被配置为输出（默认）
	01: CC1 通道被配置为输入，IC1 映射在 TI1FP1 上
	10: 保留
	11: 保留
	注： CC1S 仅在通道关闭时（TIM14_CCER 寄存器的 CC1E=0 且已被更新）才是可写的。

15.8.7 CCER 捕获/比较使能寄存器（TIM14_CCER）

地址偏移：0x20

复位值：0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.												CC1NP	Res.	CC1P	CC1E
												rw		rw	rw

Bits	31:4	保留，必须保持复位值
Bit	3	CC1NP: 在 CC1 通道作为输入时，该位与 CC1P 共同使用，定义 TI1FP1 的极性，具体请参考 CC1P 注： 一旦 LOCK 级别设为 3 或 2（TIM1_BDTR 寄存器中的 LOCK 位）并且 CC1S=00（该通道配置成输出）则该位不能被修改。
Bit	2	保留，必须保持复位值
Bit	1	CC1P: 捕获/比较 1 极性 CC1 通道作为输出： 0: OC1 高电平有效（默认） 1: OC1 低电平有效 CC1 通道作为输入： CC1NP 和 CC1P 位共同选择 TI1FP1 作为捕获的极性 00: 信号不反相，TI1FP1 的上升沿有效（默认） 01: 信号反相，TI1FP1 的下降沿有效 10: 保留 11: 信号不反相，TI1FP1 的上升沿和下降沿均有效
Bit	0	CC1E: 捕获/比较 1 使能 CC1 通道作为输出： 0: 关闭 — OC1 输出禁止（默认） 1: 开启 — OC1 信号输出到对应的输出引脚 CC1 通道作为输入： 该位决定了计数器的值是否能捕获入 TIM14_CCR1 寄存器 0: 捕获禁止（默认） 1: 捕获使能

15.8.8 CNT 计数寄存器 (TIM14_CNT)

地址偏移: 0x24

复位值: 0x0000_0000

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

Bits	31:16	保留, 必须保持复位值
Bits	15:0	CNT[15:0] : 计数器的值

15.8.9 PSC 预分频寄存器 (TIM14_PSC)

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

Bits	31:16	保留, 必须保持复位值
Bits	15:0	PSC[15:0] : 预分频器的值
计数器的频率可以由下式计算: $f_{CK_CNT} = f_{CK_PSC} / (PSC[15:0] + 1)$ 。 预分频器的值由预装载寄存器写入, 新的预分频器的值在下次更新事件到来时被采用。可以通过 AS 位 (TIM14_CR1 寄存器) 来选择读取 PSC 寄存器的值来自影子寄存器或预装载寄存器。		

15.8.10 ARR 自动重载寄存器 (TIM14_ARR)

地址偏移: 0x2C

复位值: 0x0000_0000

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

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

Bits 31:16	保留，必须保持复位值
Bits 15:0	ARR[15:0] : 自动重装载寄存器的值
	ARR 是要加载到自动重载寄存器中的值。具体请参考 15.2.1 章节。可以通过 AS 位（TIM14_CR1 寄存器）来选择读取 ARR 寄存器的值来自影子寄存器或预装载寄存器。

15.8.11 CCR1 捕获/比较寄存器 1 (TIM14_CCR1)

地址偏移: 0x34

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw															

Bits 31:16	保留，必须保持复位值
Bits 15:0	CCR1[15:0] : 捕获/比较 1 的值
	CC1 通道作为输出:
	CCR1 包含了要与计数器进行输出比较的值，它与计数器 TIM14_CNT 的值相比较，并在 OC1 端口上产生输出信号
	如果在 OC1PE 位（TIM14_CCMR1 寄存器）未选择预装载功能，写入的数值会立即传输至影子寄存器中；否则只有当更新事件发生时，此预装载值才传输至影子寄存器中
	可以通过 AS 位（TIM14_CR1 寄存器）来选择读取 CCR1 寄存器的值来自影子寄存器或预装载寄存器
	CC1 通道作为输入:
	CCR1 包含了上一次输入捕获 1 事件 (IC1) 发生时的计数器值，此时该寄存器为只读

15.8.12 OR 选项寄存器 (TIM14_OR)

地址偏移: 0x50

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.														TI1_RMP[1:0]	
														rw	

Bits	31:2	保留，必须保持复位值
Bits	1:0	TI1_RMP[1:0] : TI1 的输入选择
		00: 连接到 GPIO
		01: 连接到 WT 的时钟源
		10: 连接到 OSCH 的 128 分频时钟
		11: 连接到 MCO

16.通用定时器 TIM15/16/17

LCM32F037 内置有 5 个可同步的通用定时器，具体差异参见上面第 13 章高级定时器 TIM1 的表 13-1。各种定时器的特性区别主要在计数方向、是否具有 DMA 请求、捕获/比较通道数量以及互补输出通道数量上。

16.1 TIM15/16/17 主要特性

通用定时器 TIM15/16/17 是 16 位的定时器/计数器，由一个可编程的预分频器驱动。主要用于基本定时，测量输入信号的脉冲宽度（输入捕获）产生输出波形（输出比较，PWM 和单脉冲模式），对应于不同事件（捕获、比较、溢出、刹车、触发）的中断以及与其它定时器或外部信号（外部时钟、复位、触发和使能信号）同步。

TIM15/16/17 是完全独立的，不共享任何资源。TIM15/16/17 可以通过定时器联动机制与其他定时器共同工作，提供同步或事件联接功能。

16.1.1 TIM15 主要特性

TIM15 支持互补的 PWM 输出，能插入可编程的死区时间。TIM15 的 2 个独立的通道可以被分别用于：

- 输入捕获
- 输出比较
- PWM生成（边沿或中心对齐模式）
- 单脉冲模式输出

如果配置为 16 位 PWM 生成器，TIM15 具有全调制能力（0 到 100%）。在调试模式下，计数器可以被冻结。TIM15 可以通过定时器联动机制与其他定时器共同工作，提供同步或事件联接功能。

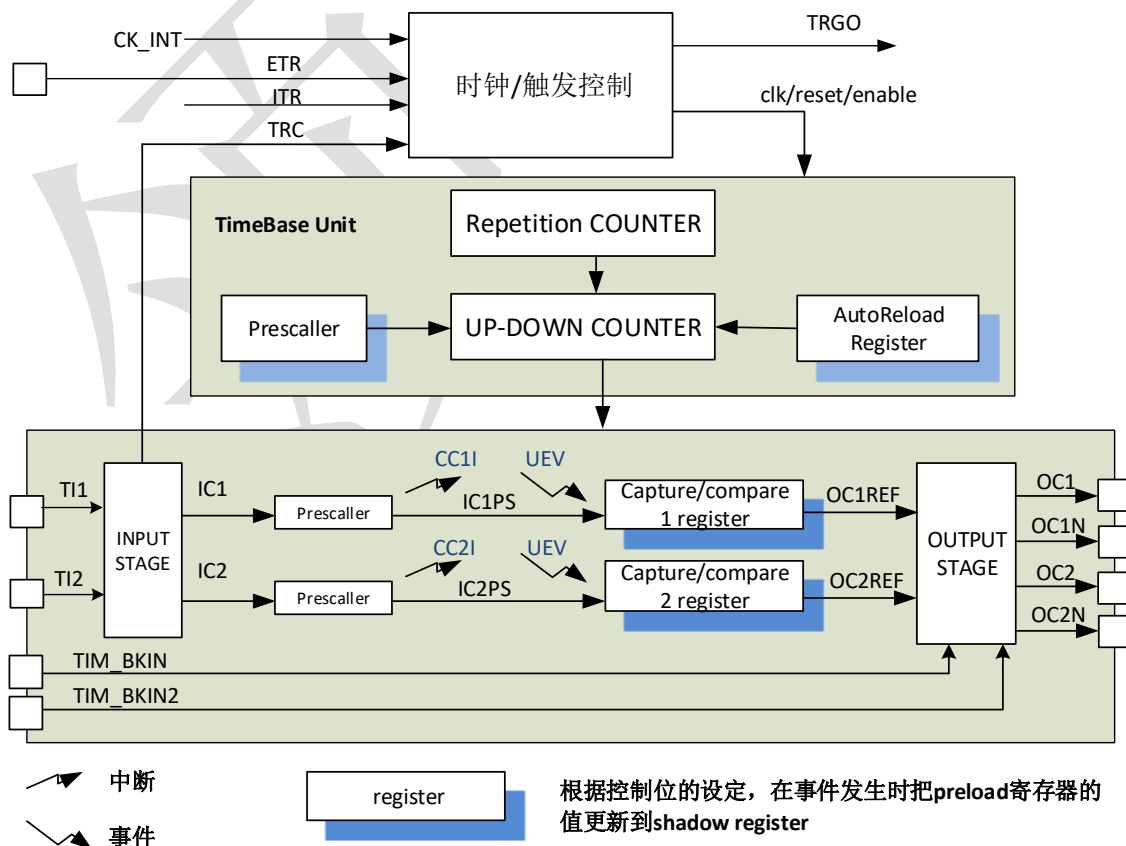


图 16-1 通用定时器 TIM15 框图

16.1.2 TIM16/17 主要特性

TIM16/17 支持互补的 PWM 输出，能插入可编程的死区时间。TIM16/17 的 1 个独立的通道可以被用于：

- 输入捕获
- 输出比较
- PWM生成（边沿或中心对齐模式）
- 单脉冲模式输出

如果配置为 16 位 PWM 生成器，TIM16/17 具有全调制能力（0 到 100%）。在调试模式下，计数器可以被冻结。TIM16/17 可以通过定时器联动机制与其他定时器共同工作，提供同步或事件联接功能。TIM16 和 TIM17 的 OC1 输出可以产生 IR_OUT 信号并从 I/O 输出，用来模拟红外发射功能。

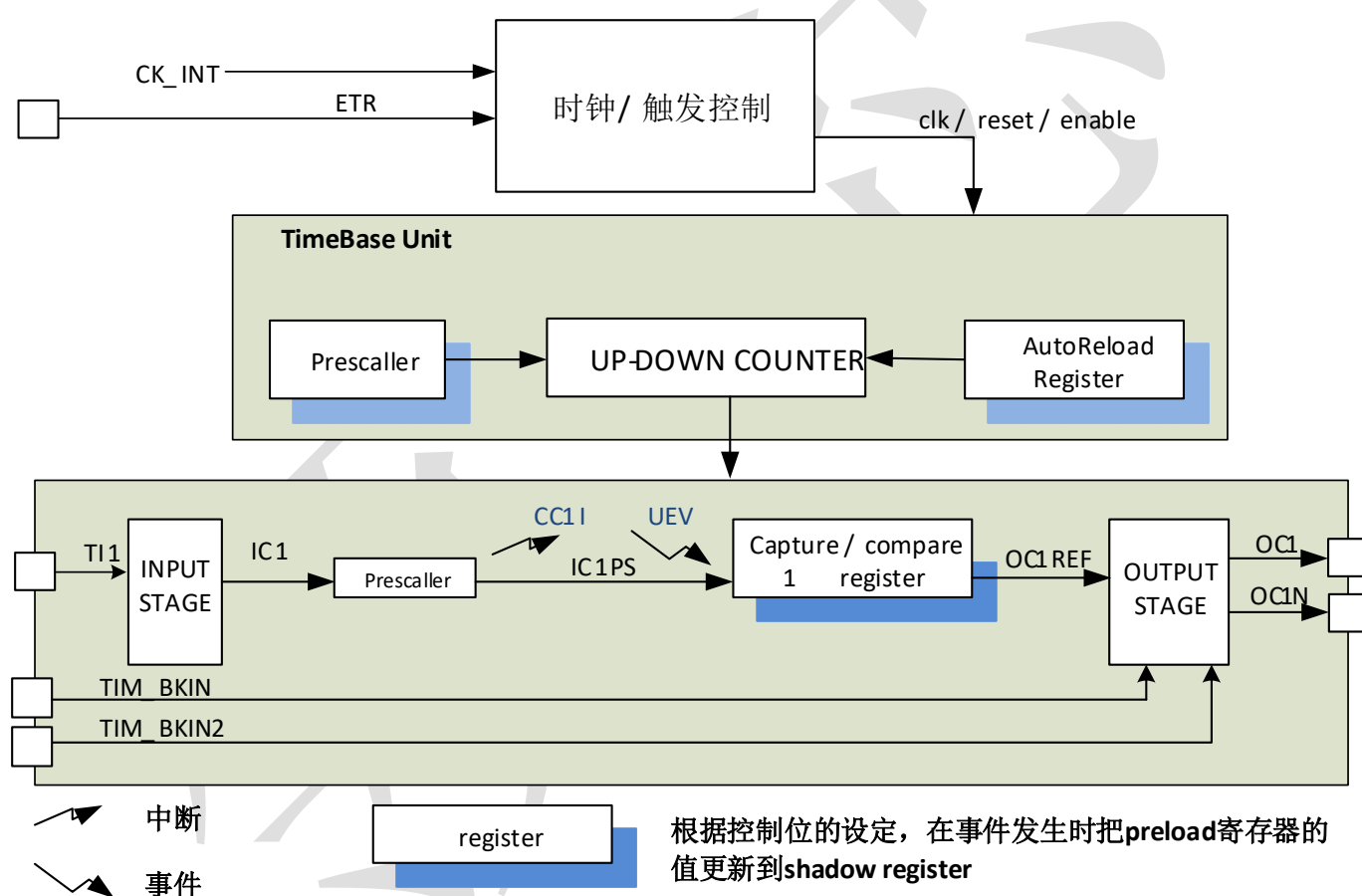


图 16-2 通用定时器 TIM16/17 框图

16.2 时基单元

时基单元包含：

- 16 位向上/向下计数器 (TIMx_CNT)
- 16 位预分频器 (TIMx_PSC)
- 16 位自动重载寄存器 (TIMx_ARR)
- 8 位重复计数器 (TIM15_RCR) (仅 TIM15 具有)

16 位计数器、预分频器、自动重载寄存器和 8 位重复计数器寄存器都可以通过软件进行读写操作。计

计数器由预分频器的输出 CK_CNT 驱动，而 CK_CNT 仅在 TIMx_CR1 寄存器的计数器使能位（CEN）被置位时才有效。

注：在使能了 CEN 位的一个时钟周期后，计数器才开始计数。

写计数器的操作没有缓存，可以在任何时候写 TIMx_CNT 寄存器，因此建议不要在计数器运行时写入新的数值，以免写入了错误的数值。

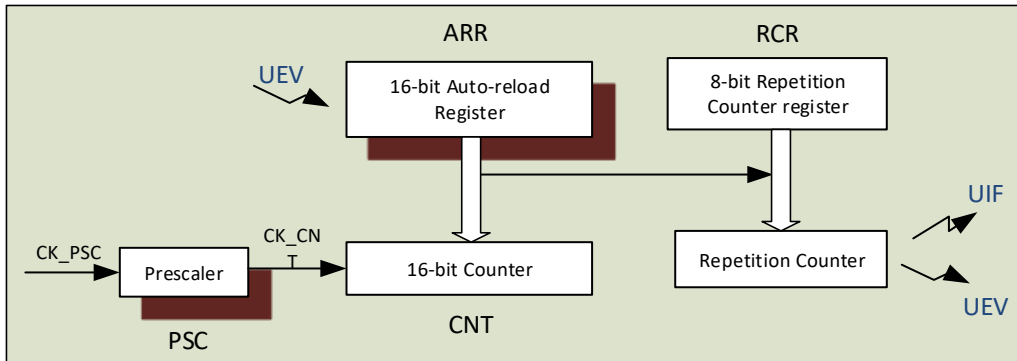


图 16-3 定时器时基单元

16.2.1 自动装载寄存器

自动装载寄存器是预先装载的。写或读自动重载寄存器将访问预装载寄存器。根据在 TIMx_CR1 寄存器中的自动装载预装载使能位（ARPE）的设置，预装载寄存器的内容被立即或在每次的更新事件 UEV 时传送到影子寄存器。当计数器达到溢出条件（例如向下计数时的下溢条件）并当 TIMx_CR1 寄存器中的 UDIS=0 时，产生更新事件。更新事件也可以由软件产生，随后会详细描述每一种配置下更新事件的产生。注意这里影子寄存器（Shadow register）即为有效的工作寄存器（Active register）。

写自动重载寄存器的两种模式：

- 自动预装载已使能（TIMx_CR1 寄存器的 ARPE 位置位）。在此模式下，写入自动重载寄存器的数据将被保存在预装载寄存器中，并在下一个更新事件（UEV）时传送到影子寄存器。
- 自动预装载已禁止（TIMx_CR1 寄存器的 ARPE 位清除）。在此模式下，写入自动重载寄存器的数据将立即写入影子寄存器。

更新事件的产生条件：

- 计数器向上或向下溢出
- 软件置位了 TIMx_EGR 寄存器的 UG 位
- 时钟/触发控制器产生硬件复位

在预装载使能时（ARPE=1），如果发生了更新事件，预装载寄存器中的数值（TIMx_ARR）将写入影子寄存器中。置位 TIMx_CR1 寄存器的 UDIS 位将禁止更新事件（UEV）。

16.2.2 预分频器

TIMx 的预分频器基于一个由 16 位寄存器（TIMx_PSC）控制的 16 位计数器。由于这个控制寄存器带有缓冲器，因此它能够在运行时被改变。预分频器可以将计数器的时钟频率按 1 到 65536 之间的任意值分频。计数器的频率可以由下式计算：

$$f_{CK_CNT} = f_{CK_PSC} / (PSC[15:0] + 1)$$

预分频器的值由预装载寄存器写入，新的预分频器的值在下一次更新事件到来时被采用。对 TIMx_PSC 寄存器的读操作通过预装载寄存器完成，因此不需要特别的关注。

下图给出了在预分频器工作时，更改其参数的情况下计数器操作的例子，其它分频系数类推。

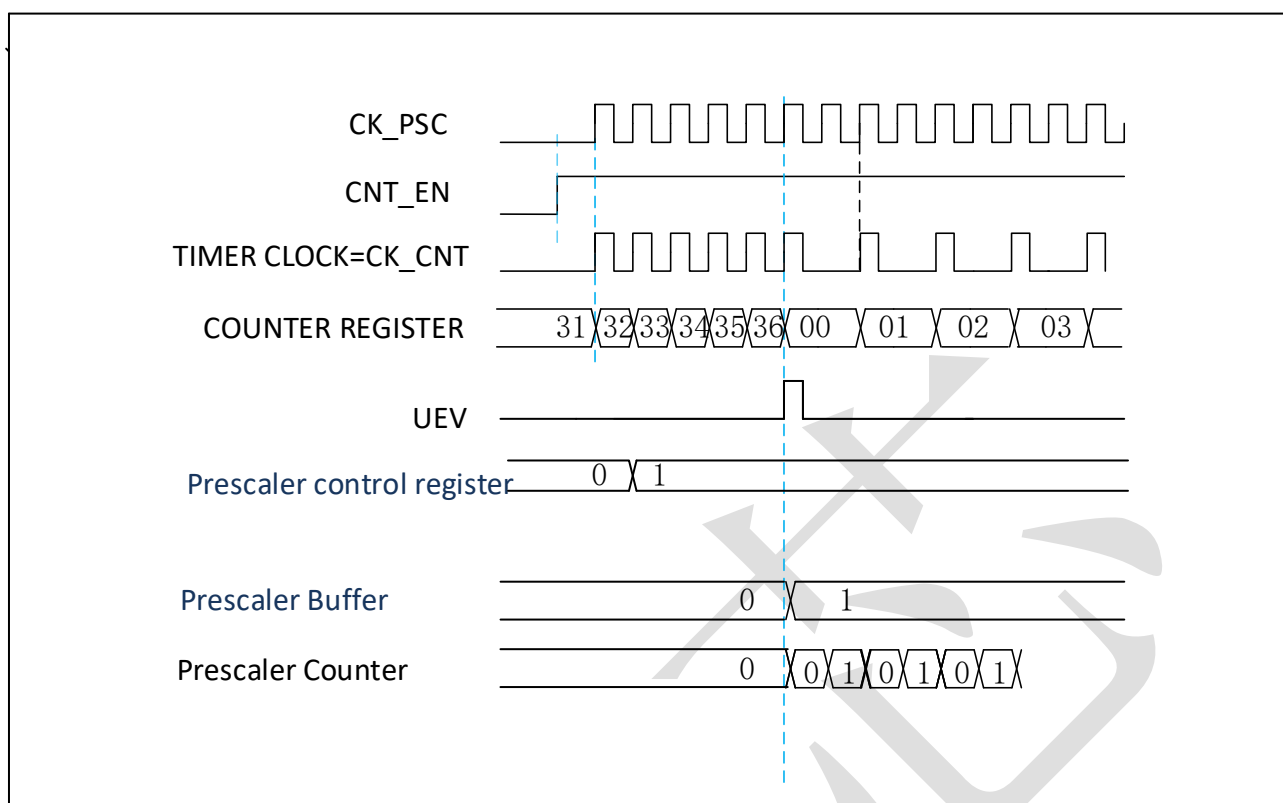


图 16-4 预分频系数从 1 变 2 的计数器时序

16.3 计数器模式

16.3.1 向上计数模式

在向上计数模式中，计数器从 0 计数到用户定义的比较值（TIMx_ARR 寄存器的值），然后重新从 0 开始计数并产生一个计数器溢出事件，同时，如果 TIMx_CR1 寄存器的 UDIS 位是 0，将会产生一个更新事件（UEV）。

如果定时器带有重复计数器，在重复了指定次数（TIMx_RCR 的值）的向上溢出之后会产生更新事件（UEV）。否则每一次的向上溢出都会产生更新事件。

置位 TIMx_EGR 寄存器的 UG 位（通过软件方式或者使用从模式控制器）也同样可以产生一个更新事件。

使用软件置位 TIMx_CR1 寄存器的 UDIS 位，可以禁止更新事件，这样可以避免在更新预装载寄存器时更新影子寄存器。在 UDIS 位被清除之前，将不产生更新事件。但是在应该产生更新事件时，计数器仍会被清 0，同时预分频器的计数也被清 0（但预分频器的数值不变）。此外，如果设置了 TIMx_CR1 寄存器中的 URS 位（选择更新请求源），设置 UG 位将产生一个更新事件 UEV，但硬件不设置 UIF 标志（即不产生中断请求）。这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。

如果定时器带有重复计数器，由于重复寄存器没有双重的缓冲，新的重复数值将立刻生效，因此在修改时需要小心。

当发生一个更新事件时，所有的寄存器都被更新，硬件同时依据 URS 位设置更新标志位（TIMx_SR 寄存器的 UIF 位）：自动装载影子寄存器被重新置入预装载寄存器的值（TIMx_ARR）。预分频器的缓存器被置入预装载寄存器的值（TIMx_PSC）。要注意到如果是因为计数器溢出而产生更新，自动重装载寄存器将在计数器重载入之前被更新，因此下一个计数周期才是预期的值（计数器被装载为新的值）。非自动重装载模式下可以立即更新。

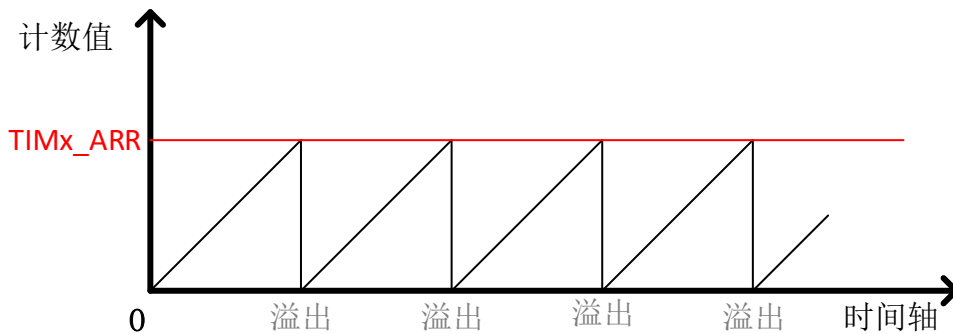


图 16-5 向上计数溢出

下图给出例子，当 TIMx_ARR=0x36 时计数器在二分频时钟频率下的动作。

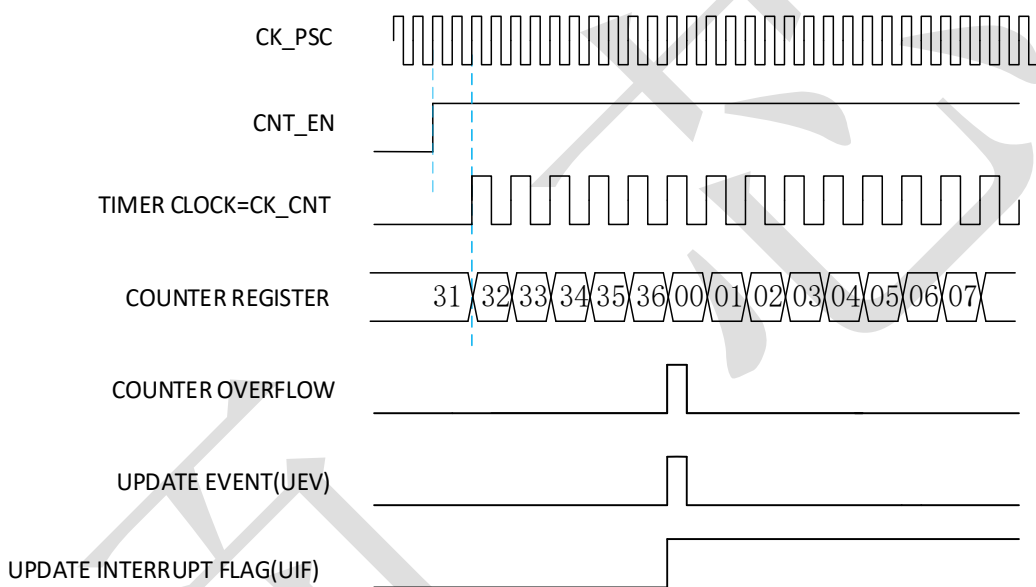


图 16-6 计数器在二分频时钟频率下的时序图

16.3.2 向下计数模式

在向下计数模式中，计数器从自动装载的值（TIMx_ARR 寄存器的值）开始向下计数到 0，然后再从自动装载的值重新开始计数，并产生一个计数器向下溢出事件。如果 TIMx_CR1 寄存器的 UDIS 位是 0，将会产生一个更新事件（UEV）。

如果定时器带有重复计数器，在重复了指定次数（TIMx_RCR 的值）的向下溢出之后会产生更新事件（UEV）。否则每一次的向下溢出都会产生更新事件。

置位 TIMx_EGR 寄存器的 UG 位(通过软件方式或者使用从模式控制器)也同样可以产生一个更新事件。

置位 TIMx_CR1 寄存器的 UDIS 位可以禁止 UEV 事件。这样可以避免在更新预装载寄存器时更新影子寄存器。因此 UDIS 位清除之前不会产生更新事件。然而，计数器仍会从当前自动加载值重新开始计数，并且预分频器的计数器重新从 0 开始（但预分频器的数值不变）。此外，如果设置了 TIMx_CR1 寄存器中的 URS 位（选择更新请求源），设置 UG 位将产生一个更新事件 UEV，但硬件不设置 UIF 标志（即不产生中断请求）。这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。

如果定时器带有重复计数器，由于重复寄存器没有双重的缓冲，新的重复数值将立刻生效，因此在修改

时需要小心。

当发生一个更新事件时，所有的寄存器都被更新，硬件同时依据 URS 位设置更新标志位 (TIMx_SR 寄存器的 UIF 位)：自动装载影子寄存器被重新置入预装载寄存器的值 (TIMx_ARR)。预分频器的缓存器被置入预装载寄存器的值 (TIMx_PSC)。要注意到如果是因为计数器溢出而产生更新，自动重装载寄存器将在计数器重载入之前被更新，因此下一个计数周期才是预期的值 (计数器被装载为新的值)。非自动重装载模式下可以立即更新。

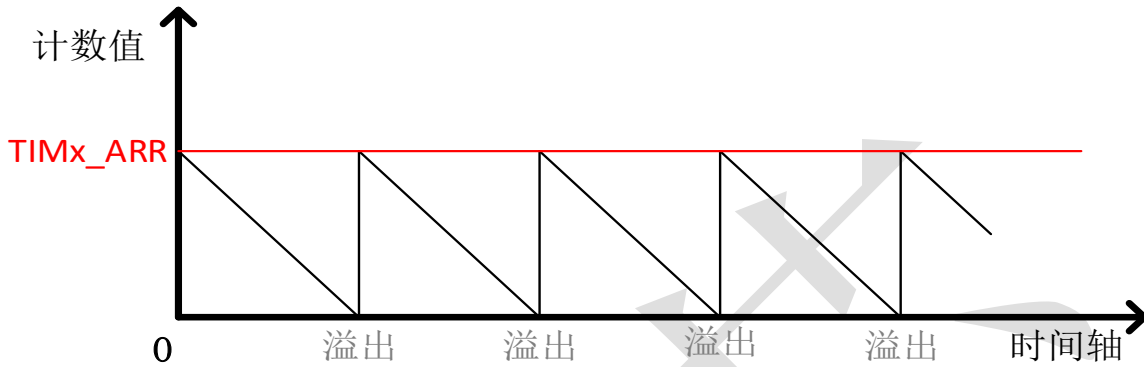


图 16-7 向下计数溢出

16.3.3 中央对齐模式 (向上/向下计数)

在中央对齐模式中，计数器从 0 开始计数到自动加载的值 (TIMx_ARR-1)，产生一个计数器溢出事件，然后向下计数到 1 并且产生一个计数器下溢事件；然后再从 0 开始重新计数。

在此模式下，不能写入 TIMx_CR1 中的 DIR 方向位。它由硬件更新并指示当前的计数方向。

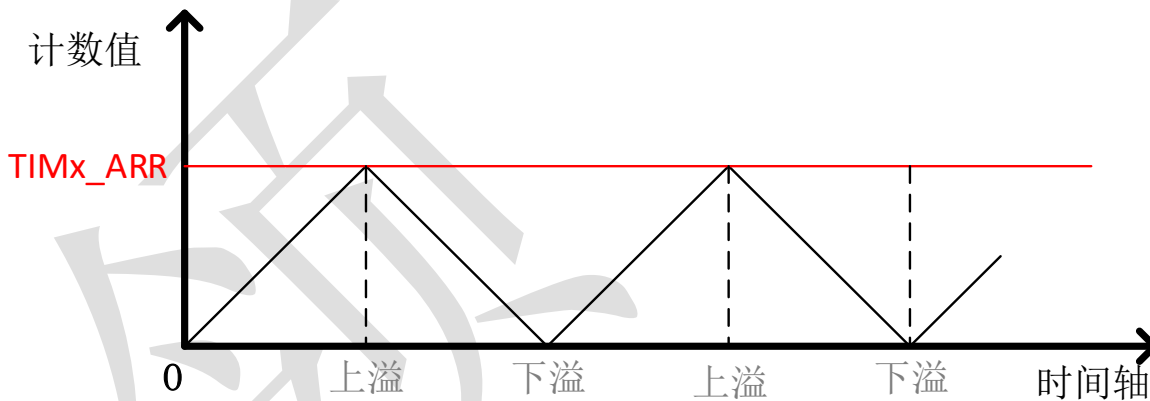


图 16-8 中央对齐溢出

如果定时器带有重复计数器，在重复了指定次数 (TIMx_RCR 的值) 的向上和向下溢出之后会产生更新事件 (UEV)。否则每一次的向上向下溢出都会产生更新事件。

置位 TIMx_EGR 寄存器的 UG 位 (通过软件方式或者使用从模式控制器) 也同样可以产生一个更新事件。此时，计数器重新从 0 开始计数，预分频器也重新从 0 开始计数。

设置 TIMx_CR1 寄存器中的 UDIS 位可以禁止 UEV 事件，这样可以避免在更新预装载寄存器时更新影子寄存器。因此 UDIS 位被清为 0 之前不会产生更新事件。然而，计数器仍会根据当前自动重加载的值，继续向上或向下计数。

如果定时器带有重复计数器，由于重复寄存器没有双重的缓冲，新的重复数值将立刻生效，因此在修改时需要小心。

此外，如果设置了 TIMx_CR1 寄存器中的 URS 位 (选择更新请求源)，设置 UG 位将产生一个更新事件

UEV, 但硬件不设置 UIF 标志 (即不产生中断请求)。这是为了避免在捕获模式下清除计数器时, 同时产生更新和捕获中断。

当发生一个更新事件时, 所有的寄存器都被更新, 硬件同时依据 URS 位设置更新标志位 (TIMx_SR 寄存器的 UIF 位); 自动装载影子寄存器被重新置入预装载寄存器的值 (TIMx_ARR)。预分频器的缓存器被置入预装载寄存器的值 (TIMx_PSC)。要注意到如果是因为计数器溢出而产生更新, 自动重载寄存器将在计数器重载入之前被更新, 因此下一个计数周期才是预期的值 (计数器被装载为新的值)。

使用中央对齐模式:

- 启动中央对齐模式时, 计数器将按照原有的向上/向下的配置计数。也就是说 TIMx_CR1 寄存器中的 DIR 位将决定计数器是向上还是向下计数。此外, 软件不能同时修改 DIR 位和 CMS 位的值。
- 不推荐在中央对齐模式下, 计数器正在计数时写计数器的值, 这将导致不能预料的后果。具体的说:
 - 向计数器写入了比自动装载值更大的数值时 (TIMx_CNT > TIMx_ARR), 但计数器的计数方向不发生改变。例如计数器已经向上溢出, 但计数器仍然向上计数。
 - 向计数器写入了 0 或者 TIMx_ARR 的值, 但更新事件不发生。
- 安全使用中央对齐模式的计数器的方法是在启动计数器之前先用软件置位 TIMx_EGR 寄存器的 UG 位产生一个更新事件, 并且不在计数器计数时修改计数器的值。

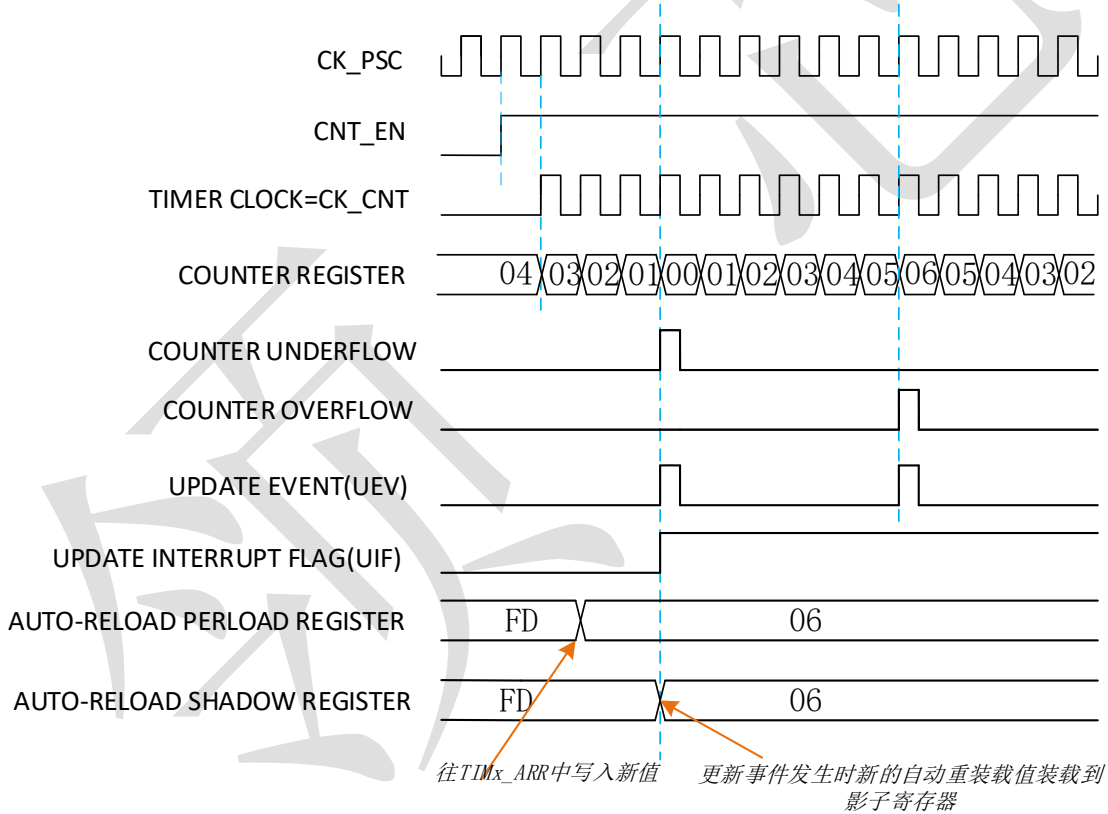


图 16-9 计数器时序图, 内部时钟分频为 1, TIMx_ARR=0x6

16.4 时钟选择

时钟/触发控制器允许用户选择计数器的时钟源 (CK_PSC)、触发输入信号 (TRGI) 和触发输出信号 (TRGO), 其框图如图 16-1 所示。

时基单元的预分频时钟 (CK_PSC) 可以有以下来源:

- 内部时钟 (CK_INT): 芯片内部时钟

- 外部时钟模式 1：外部触发输入（TII）和内部触发输入（ITRx）
 - 外部时钟模式 2：外部触发输入 ETR
- TIM16/17 不支持触发输出信号 TRGO，也不支持内部触发 ITR。

16.4.1 内部时钟源（CK_INT）

如果 TIMx_SMCR 寄存器 SMS=000，TIMx_SMCR 寄存器 ECE=0，则 CEN、DIR 和 UG 位是实际上的控制位，并且只能被软件修改（UG 位仍被自动清除）。一旦 CEN 位被写成 1，预分频器的时钟就由内部时钟提供。

16.4.2 外部时钟源模式 1

当 TIMx_SMCR 寄存器的 SMS=111 时，此模式被选中。计数器可以在选定输入端的每个上升沿或下降沿计数。

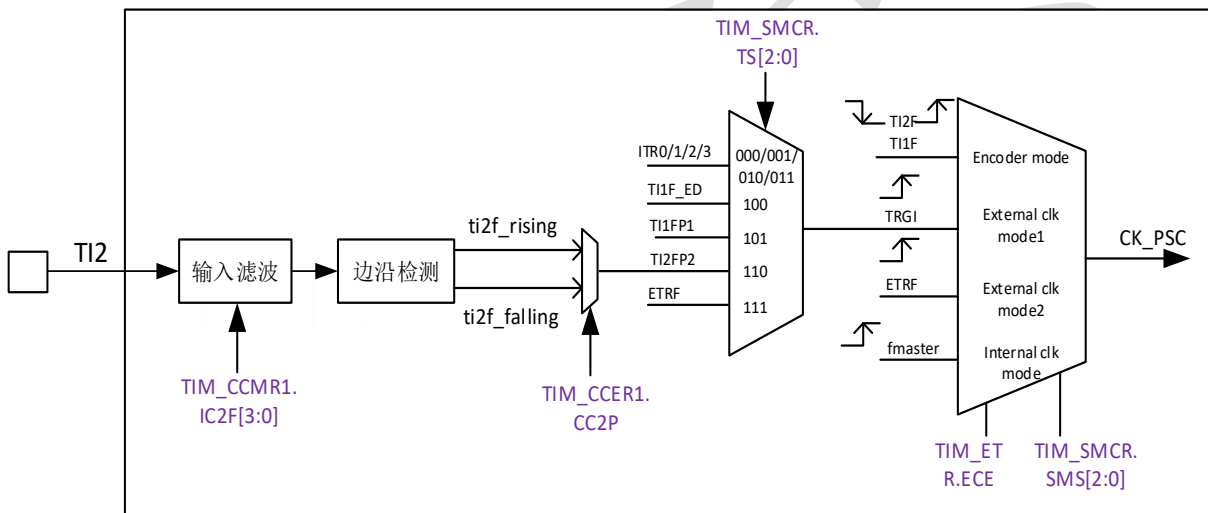


图 16-10 TI2 外部时钟框图

例如，要配置 TIM15 向上计数器在 TI2 输入端的上升沿计数，使用下列步骤：

1. 配置 TIM15_CCMR1 寄存器的 CC2S=01，使用通道 2 检测 TI2 输入的上升沿
2. 配置 TIM15_CCMR1 寄存器的 IC2F[3:0]位，选择输入滤波器带宽（如果不需要滤波器，保持 IC2F=0000）
3. 配置 TIM15_CCER 寄存器的 CC2P=0，选定上升沿极性
4. 配置 TIM15_SMCR 寄存器的 SMS=111，配置计数器使用外部时钟模式 1
5. 配置 TIM15_SMCR 寄存器的 TS=110，选定 TI2 作为输入源
6. 设置 TIM15_CR1 寄存器的 CEN=1，启动计数器

当上升沿出现在 TI2，计数器计数一次，且触发标识位（TIM15_SR 寄存器的 TIF 位）被置 1，如果使能了中断（在 TIM15_DIER 寄存器中配置）则会产生中断请求。

在 TI2 的上升沿和计数器实际时钟之间的延时取决于在 TI2 输入端的重同步电路(下图预分频系数为 1)。

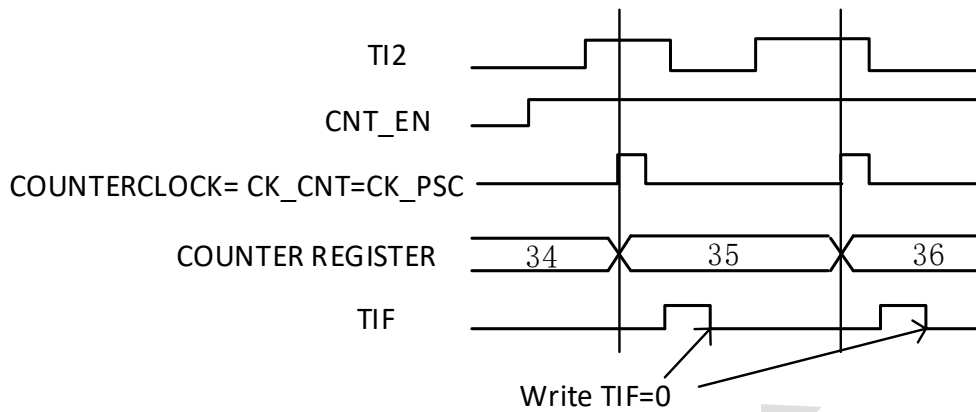


图 16-11 外部时钟模式 1 下的控制电路

16.4.3 外部时钟模式 2

计数器能够在外部触发输入 ETR 信号的每一个上升沿或下降沿计数。将 TIMx_SMCR 寄存器的 ECE 位写 1，即可选定此模式。

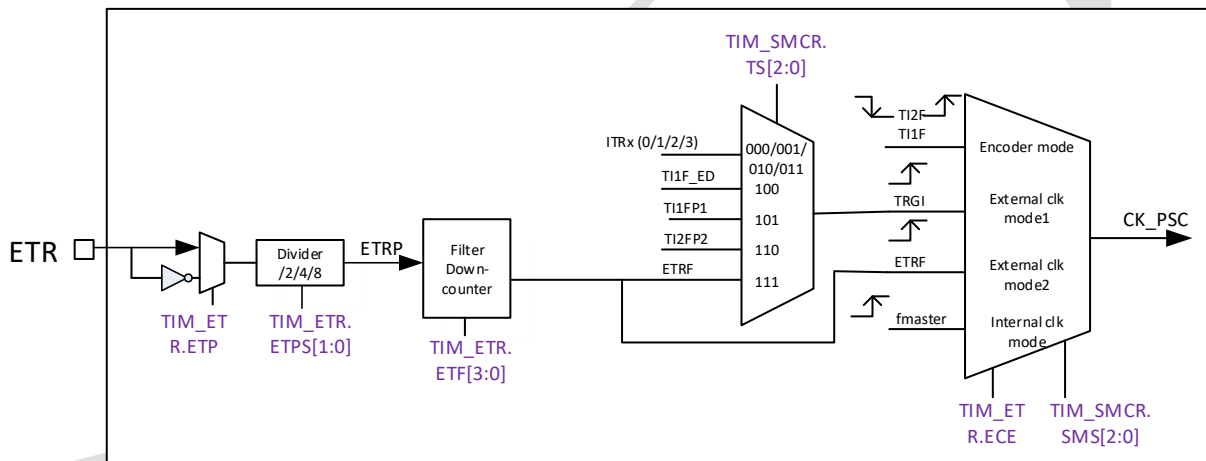


图 16-12 外部触发时钟框图

例如，要配置 TIM15 计数器在 ETR 信号的每 2 个上升沿时向上计数一次，需使用下列步骤：

1. 本例中不需要滤波器，配置 TIM15_SMCR 寄存器的 ETF[3:0]=0000
2. 设置预分频器，配置 TIM15_SMCR 寄存器的 ETPS[1:0]=01
3. 选择 ETR 的上升沿检测，配置 TIM15_SMCR 寄存器的 ETP=0
4. 开启外部时钟模式 2，配置 TIM15_SMCR 寄存器中的 ECE=1
5. 启动计数器，写 TIM15_CR1 寄存器的 CEN=1

计数器在每 2 个 ETR 上升沿计数一次。在 ETR 的上升沿和计数器实际时钟之间的延时取决于在 ETRP 信号端的重同步电路。

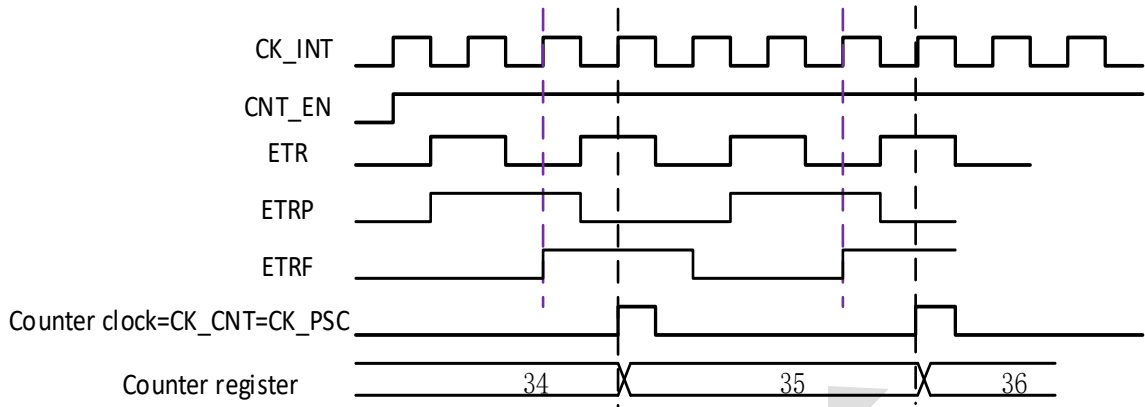


图 16-13 外部时钟模式 2 下的控制电路

16.5 捕获/比较通道

定时器的 I/O 引脚 (TIMx_CCI) 可以用作输入捕获或者输出比较, 这个功能可以通过配置捕获/比较通道模式寄存器 (TIMx_CCMRi) 的 CCIS 通道选择位来实现, 此处的 i 代表通道数。

每一个捕获/比较通道都是围绕着一个捕获/比较寄存器 (包含影子寄存器) 来构建的, 包括捕获的输入部分 (数字滤波、多路复用和预分频器), 和输出部分 (比较器和输出控制)。

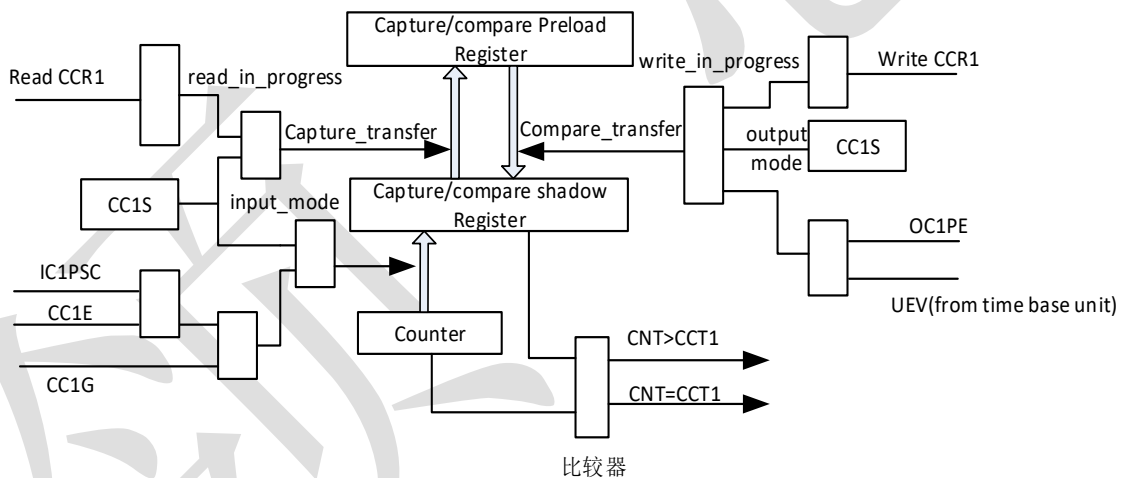


图 16-14 捕获/比较通道 1 的主电路

捕获/比较模块由一个预装载寄存器和一个影子寄存器组成, 读写过程仅操作预装载寄存器。在捕获模式下, 捕获计数器的值就在影子寄存器上, 然后再复制到预装载寄存器中。在比较模式下, 预装载寄存器的内容被复制到影子寄存器中, 然后影子寄存器内容和计数器进行比较。

当通道被配置成输出模式时 (TIMx_CCMRi 寄存器的 CCIS=0), 可以随时访问 TIMx_CCRi 寄存器 (此处的 i 指通道数)。

当通道被配置成输入模式时, 对 TIMx_CCRi 寄存器的读操作类似于计数器的读操作。当捕获发生时, 计数器的内容被捕获到 TIMx_CCRi 影子寄存器, 然后再复制到预装载寄存器中。在读操作进行中, 预装载寄存器是被冻结的 (以防读数据过程中预装载寄存器内容被修改)。

16.5.1 输入捕获模式

在输入捕获模式下，当检测到 ICi 信号上相应的边沿后，计数器的当前值被锁存到捕获/比较寄存器 (TIMx_CCRi) 中。当发生捕获事件时，相应的 CCIIF 标志 (TIMx_SR 寄存器) 被置 1。

如果 TIMx_DIER 寄存器的 CCIIE 位被置位，也就是使能了中断，则将产生中断请求。如果发生捕获事件时 CCIIF 标志已经为高，那么重复捕获标志 CCIOf (TIMx_SR 寄存器) 被置 1，写 CCIIF=0 可清除 CCIOf。写 CCIIF=0 或读取存储在 TIMx_CCRi 寄存器中的捕获数据都可清除 CCIIF。

TIM15 通道 1 的输入如下图所示：

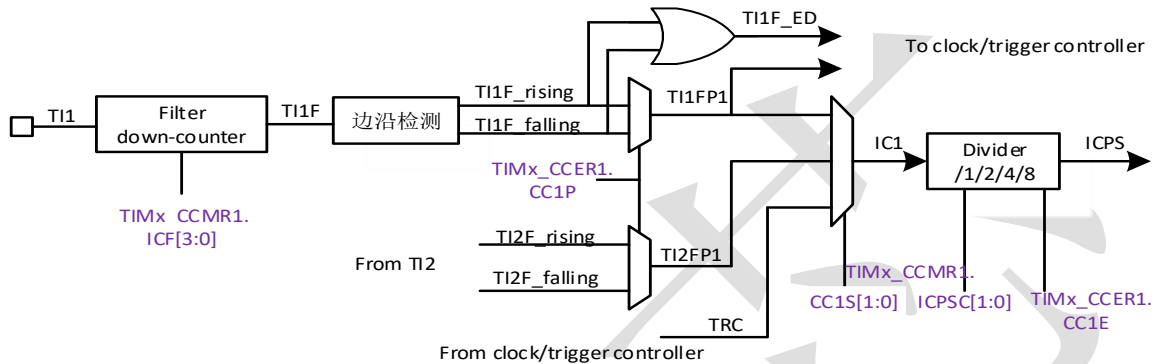


图 16-15 TIM15 通道 1 的输入

以下例子说明如何在 TI1 输入的上升沿时捕获计数器的值到 TIMx_CCR1 寄存器中，步骤如下：

1. 选择有效输入端：例如 TIMx_CCR1 连接到 TI1 输入，所以写入 TIMx_CCMR1 寄存器中的 CC1S=01，此时通道被配置为输入，并且 TIMx_CCR1 寄存器变为只读。
2. 根据输入信号 TIi 的特点，可通过配置 TIMx_CCMRi 寄存器中的 ICiF 位来设置相应的输入滤波器的滤波时间。假设输入信号在最多 5 个时钟周期的时间内抖动，我们须配置滤波器的带宽长于 5 个时钟周期；因此我们可以连续采样 8 次，以确认在 TI1 上一次真实的边沿变换，即在 TIMx_CCMR1 寄存器中写入 IC1F=0011，此时，只有连续采样到 8 个相同的 TI1 信号，信号才为有效（采样频率为 DTS 时钟的频率 f_{DTS} ）。
3. 选择 TI1 通道的有效转换边沿，在 TIMx_CCER 寄存器中写入 CC1P=0（上升沿）。
4. 配置输入预分频器。在本例中，我们希望捕获发生在每一个有效的电平转换时刻，因此预分频器被禁止（写 TIMx_CCMR1 寄存器的 IC1PSC=00）。
5. 设置 TIMx_CCER 寄存器的 CC1E=1，允许捕获计数器的值到捕获寄存器中。
6. 如果需要，通过设置 TIMx_DIER 寄存器中的 CCI1E 位允许相关中断请求。

当发生一个输入捕获时：

- 当产生有效的电平转换时，计数器的值被传送到 TIMx_CCR1 寄存器。
- CC1IF 标志被设置（中断标志）。当发生至少 2 次连续的捕获，且 CC1IF 未被清除，CC1OF 也被置 1。
- 如设置了 CCI1E 位，则会产生一个中断。
- 为了处理捕获溢出（CC1OF 位），建议在读出重复捕获标志之前读取数据，这是为了避免丢失在读出捕获溢出标志之后和读取数据之前可能产生的重复捕获信息。

注：设置 TIMx_EGR 寄存器中相应的 CCIg 位，可以通过软件产生输入捕获中断。

16.5.2 PWM 输入

这里 PWM 输入模式只有 TIM15 能够实现。该模式是输入捕获模式的一个特例，除下列区别外，操作与

输入捕获模式相同：

- 两个 ICi 信号被映射至同一个 Tii 输入
- 这两个 ICi 信号的有效边沿的极性相反
- 两路 TiiFP 信号中的一路被选择作为触发输入信号，并且触发模式控制器被配置成触发复位模式

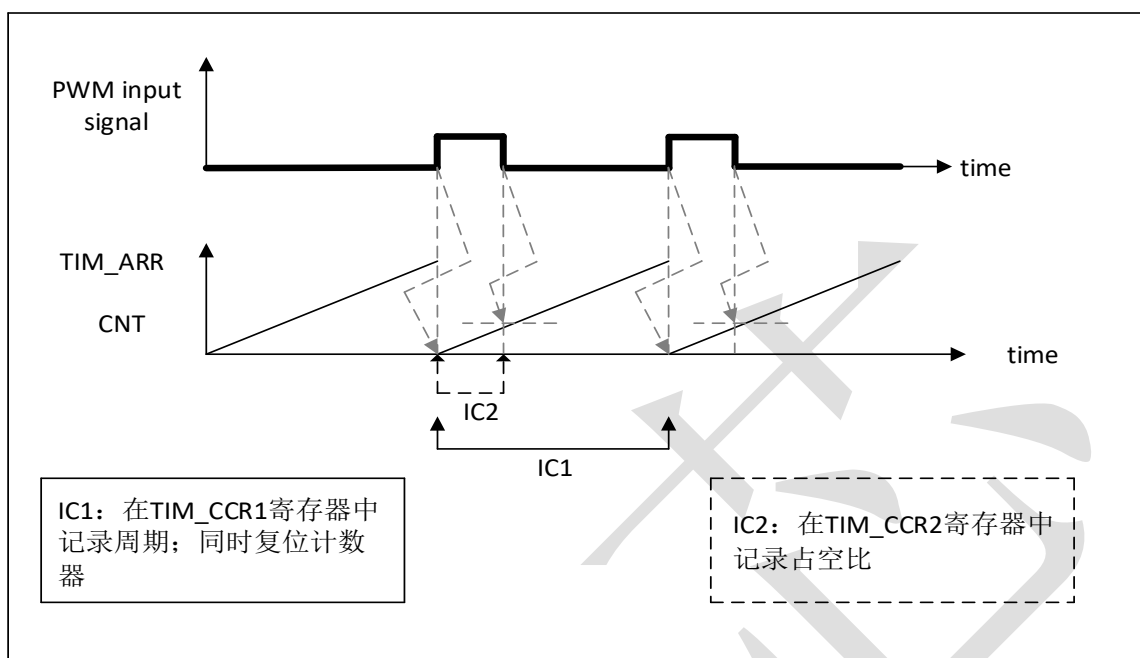


图 16-16 PWM 输入信号测量

例如，你可以用以下方式测量 T11 上输入的 PWM 信号的周期（锁存于 TIMx_CCR1 寄存器）和占空比（锁存于 TIMx_CCR2 寄存器）。（具体取决于 CK_INT 的频率 f_{CK_INT} 和预分频器的值）

1. 选择 TIMx_CCR1 的有效输入：置 TIMx_CCMR1 寄存器的 CC1S=01（选中 TI1）。
2. 选择 TI1FP1 的有效极性（用来捕获数据到 TIMx_CCR1 中和清除计数器）：置 CC1P=0（上升沿有效）。
3. 选择 TIMx_CCR2 的有效输入：置 TIMx_CCMR1 寄存器的 CC2S=10（选中 TI1FP2）。
4. 选择 TI1FP2 的有效极性（捕获数据到 TIMx_CCR2）：置 CC2P=1（下降沿有效）。
5. 选择有效的触发输入信号：置 TIMx_SMCR 寄存器中的 TS=101（选择 TI1FP1）。
6. 配置触发模式控制器为复位触发模式：置 TIMx_SMCR 中的 SMS=100。
7. 使能捕获：置 TIMx_CCER 寄存器中 CC1E=1，CC2E=1。

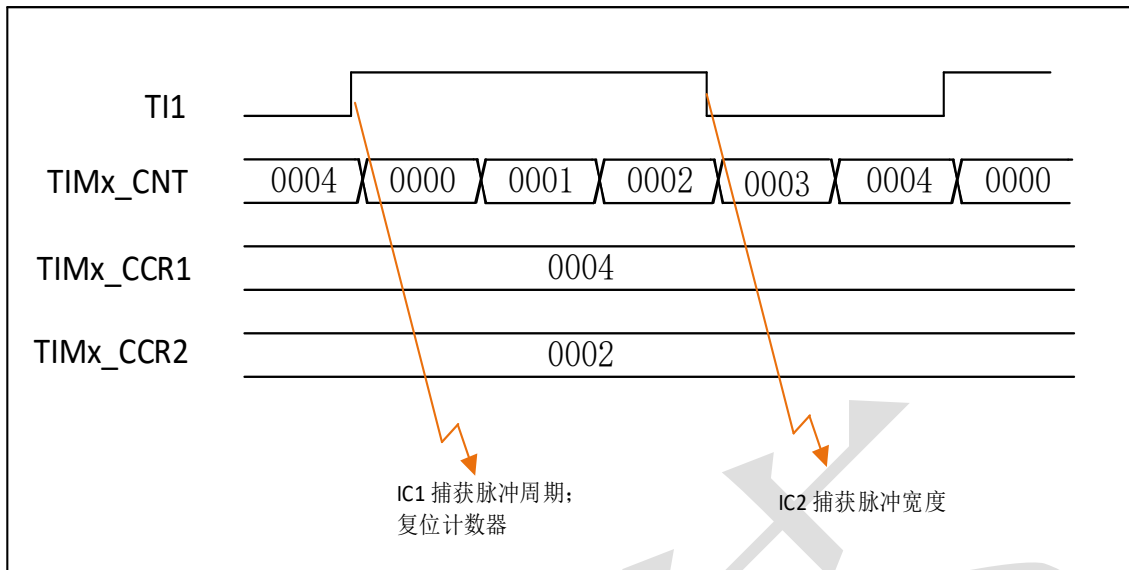


图 16-17 PWM 输入信号测量实例

16.5.3 强制输出模式

输出模块会产生一个用来做参考的中间波形，称为 OCiREF（高有效）。刹车功能和极性选择都在之后处理。在输出模式（TIMx_CCMRi 寄存器中 CCIS=00）下，输出比较信号能够直接由软件强置为高或低状态，而不依赖于输出比较寄存器和计数器间的比较结果。

置 TIMx_CCMRi 寄存器中相应的 OCiM=101，即可强置输出比较信号为有效状态。这样 OCiREF 被强置为高电平（OCiREF 始终为高电平有效），而 OCi 的输出是高还是低则取决于 CCiP 极性标志位。例如 CCiP=0（OCi 高电平有效），则 OCi 被强置为高电平。

置 TIMx_CCMRi 寄存器的 OCiM=100，可强置 OCiREF 信号为低。

该模式下，在 TIMx_CCRi 影子寄存器和计数器之间的比较仍然在进行，相应的标志也会被修改，也仍然会产生相应的中断。这将会在下面的输出比较模式一节中介绍。

16.5.4 输出比较模式

此模式用来控制一个输出波形或者指示一段给定的时间已经达到。当计数器与捕获/比较寄存器的内容相同时，有如下操作：

- 根据不同的输出比较模式，相应的 OCiREF 输出信号为：
 - 保持不变（OCiM=000）
 - 设置为有效电平（OCiM=001）
 - 设置为无效电平（OCiM=010）
 - 翻转（OCiM=011）
 - 设置中断状态寄存器中的标志位（TIMx_SR 寄存器中的 CCiIF 位）
 - 若设置了相应的中断使能位（TIMx_DIER 寄存器中的 CCiIE 位），则产生一个中断
- TIMx_CCMRi 寄存器的 OCiM 位用于选择输出比较模式，而 TIMx_CCER 寄存器的 CCiP 位用于选择有效和无效的电平极性。

TIMx_CCMRi 寄存器的 OCiPE 位用于选择 TIMx_CCRi 寄存器是否需要使用预装载寄存器。

在输出比较模式下，更新事件 UEV 对 OCiREF 和 OCi 输出没有影响。输出比较的时间精度为计数器的一个时钟周期，输出比较模式也能用来输出一个单脉冲。输出比较模式的配置步骤如下：

1. 选择计数器时钟（内部/外部/预分频器）
2. 将相应的数据写入 TIMx_ARR 和 TIMx_CCRi 寄存器中
3. 如果要产生一个中断请求，设置 CCIIE 位
4. 选择输出模式步骤：
 - 要求计数器与 CCRi 匹配时翻转 OCiM 的输出管脚，设置 OCiM=011
 - 置 OCiPE = 0 禁用预装载寄存器
 - 置 CCIp = 0 选择高电平为有效电平
 - 置 CCIe = 1 使能输出
5. 设置 TIMx_CR1 寄存器的 CEN 位来启动计数器

在未使用预装载寄存器（OCiPE=0，否则 TIMx_CCRi 的影子寄存器只能在发生下一次更新事件时被更新）情况下，TIMx_CCRi 寄存器能够在任何时候通过软件进行更新以控制输出波形。

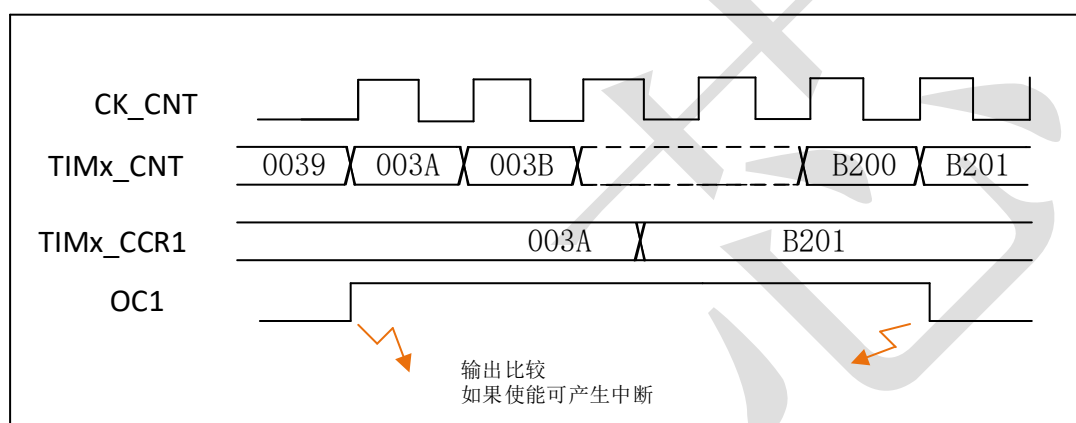


图 16-18 输出比较模式，翻转 OC1

16.5.5 PWM 模式

脉冲宽度调制（PWM）模式可以产生一个由 TIMx_ARR 寄存器确定频率，由 TIMx_CCRi 寄存器确定占空比的信号。

在 TIMx_CCMRi 寄存器中的 OCiM 位写入 110（PWM 模式 1）或 111（PWM 模式 2），能够独立地设置每个 OCi 输出通道产生一路 PWM。

必须设置 TIMx_CCMRi 寄存器的 OCiPE 位使能相应的预装载寄存器，也可以设置 TIMx_CR1 寄存器的 ARPE 位使能自动重载的预装载寄存器（在向上计数模式或中央对称模式中）。

由于仅当发生一个更新事件的时候，预装载寄存器才能被传送到影子寄存器，因此在计数器开始计数之前，必须通过设置 TIMx_EGR 寄存器的 UG 位来初始化所有的寄存器。

OCi 的极性可以通过软件在 TIMx_CCER 寄存器中的 CCIp 位设置，它可以设置为高电平有效或低电平有效。OCi 的输出使能通过 TIMx_CCER 和 TIMx_BDTR 寄存器中 CCIe、MOE、OSSR 和 OSSI 位的组合来控制。详见 TIMx_CCER 寄存器的描述。

在 PWM 模式（模式 1 或模式 2）下，TIMx_CNT 和 TIMx_CCRi 始终在进行比较，（依据计数器的计数方向）以确定是否符合 $TIMx_CCRi \leq TIMx_CNT$ 或者 $TIMx_CNT \leq TIMx_CCRi$ 。

根据 TIMx_CR1 寄存器中 CMS 位域的状态，定时器能够产生边沿对齐的 PWM 信号或中央对齐的 PWM 信号。

16.5.5.1 PWM 边沿对齐模式

向上计数配置

当 TIMx_CR1 寄存器中的 DIR 位为低的时候执行向上计数。下面是一个 PWM 模式 1 的例子。当 TIMx_CNT < TIMx_CCRi 时，PWM 参考信号 OCiREF 为高，否则为低。如果 TIMx_CCRi 中的比较值大于自动重载值 (TIMx_ARR)，则 OCiREF 保持为 1。如果比较值为 0，则 OCiREF 保持为 0。下图为 TIMx_ARR=8 时边沿对齐的 PWM 波形实例。

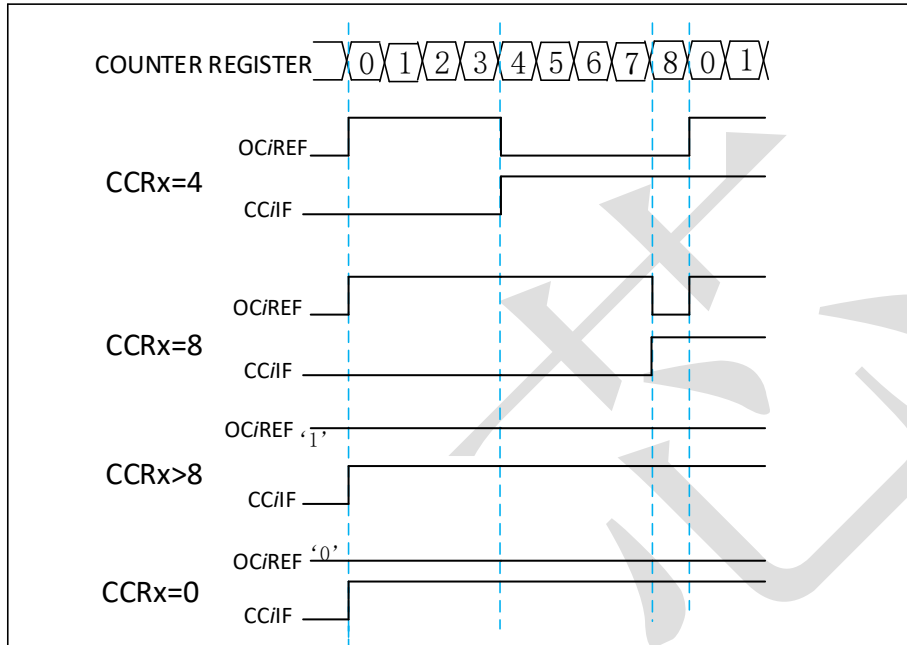


图 16-19 TIMx_ARR=8 时边沿对齐的 PWM 波形实例

向下计数的配置

当 TIMx_CR1 寄存器的 DIR 位为高时执行向下计数。在 PWM 模式 1 时，当 TIMx_CNT > TIMx_CCRi 时参考信号 OCiREF 为低，否则为高。如果 TIMx_CCRi 中的比较值大于 TIMx_ARR 中的自动重载值，则 OCiREF 保持为 1。该模式下不能产生 0% 的 PWM 波形。

16.5.5.2 PWM 中央对齐模式

当 TIMx_CR1 寄存器中的 CMS 位不为 00 时为中央对齐模式（所有其它的配置对 OCiREF/OCi 信号都有相同的作用）。

根据不同的 CMS 位的设置，比较标志可以在计数器向上计数，向下计数，或向上和向下计数时被置 1。TIMx_CR1 寄存器中的计数方向位 (DIR) 由硬件更新，不要用软件修改它。

下图给出了一些中央对齐的 PWM 波形的例子：

- TIMx_ARR=8
- PWM 模式 1
- 标志位在以下三种情况下被置位（以箭头形式在图 16-20 中标出）
 - 只有在计数器向下计数时 (CMS=01)
 - 只有在计数器向上计数时 (CMS=10)
 - 在计数器向上和向下计数时 (CMS=11)

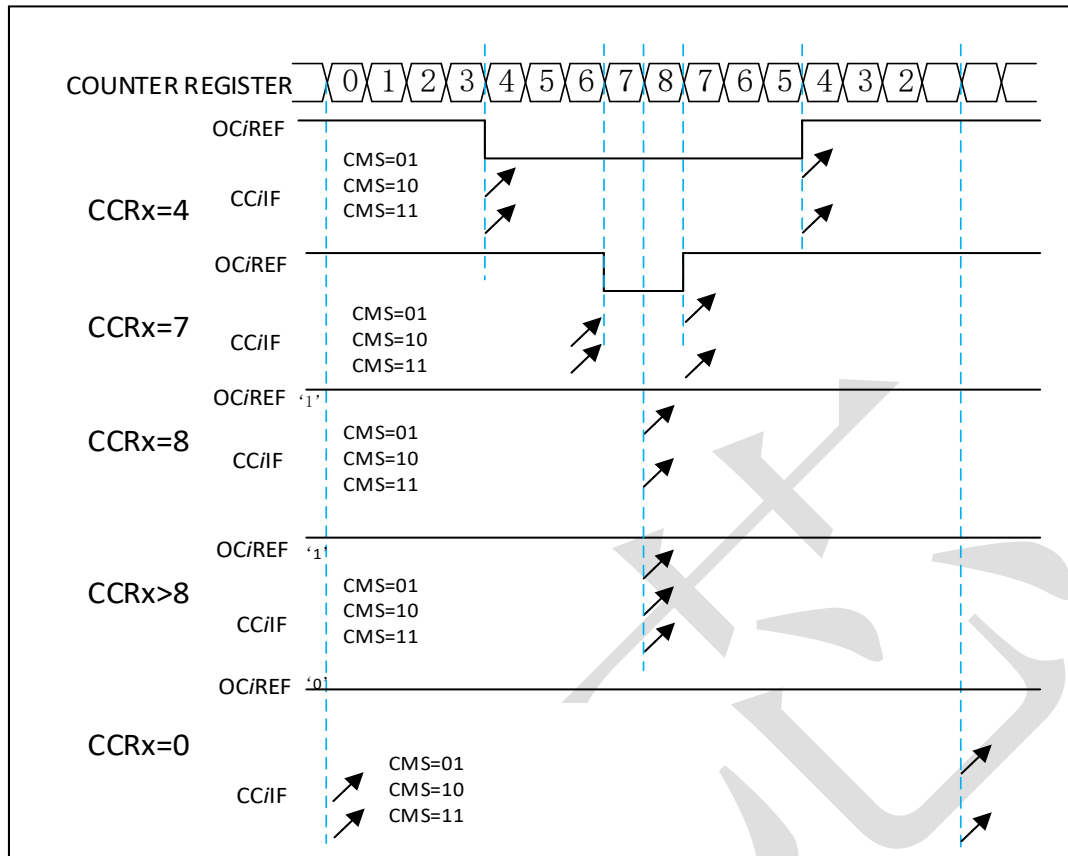


图 16-20 中央对齐模式 PWM (ARR=8)

16.5.6 单脉冲模式

单脉冲模式 (OPM) 是上述众多模式的一个特例。这种模式允许计数器响应一个激励，并在一个程序可控的延时之后产生一个脉宽可控的脉冲。

可以通过时钟/触发控制器启动计数器，在输出比较模式或者 PWM 模式下产生波形。设置 TIMx_CR1 寄存器的 OPM 位将选择单脉冲模式，此时计数器自动地在下一个更新事件 UEV 时停止。

仅当比较值与计数器的初始值不同时，才能产生一个脉冲。启动之前（当定时器正在等待触发），必须如下配置：

向上计数方式：计数器 $CNT < CCRi \leq ARR$

向下计数方式：计数器 $CNT > CCRi$

单脉冲模式的配置步骤：

如图 16-21 所示，如果用户想在 OC1 生成一个高电平脉冲，它通过 TI2 的输入触发，启动延迟为 t_{DELAY} ，维持时间为 t_{PULSE} 。首先要把 TI2FP2 设定为触发信号，此外单脉冲信号还要通过比较寄存器定义（考虑到时钟频率和预分频器）：

1. TI2FP2 对应到 TI2，配置 TIMx_CCMR1 寄存器中的 CC2S=01
2. TI2FP2 上升沿检测，配置 TIMx_CCER 寄存器中的 CC2P=0
3. 设置 TI2FP2 为从模式下的触发信号 (TRGI)，配置 TIMx_SMCR 寄存器中的 TS=110
4. 设置 TI2FP2 为计数器的启动信号，配置 TIMx_SMCR 寄存器中的 SMS=110 (触发模式)
5. t_{DELAY} 通过 TIMx_CCR1 寄存器配置得到
6. t_{PULSE} 通过 $TIMx_ARR - TIMx_CCR1$ 得到

7. 如果在计数器到达 $TIMx_CCR1$ 时信号由 0 变为 1, 到达 $TIMx_ARR$ 时信号由 1 变为 0, 则设置 PWM 模式 2, 配置 $TIMx_CCMR1$ 寄存器中的 $OC1M=111$

可选启用预装载功能, 配置 $TIMx_CCMR1$ 寄存器中的 $OC1PE=1$ 和 $TIMx_CR1$ 寄存器中的 $ARPE=1$ 。如果使用预装载功能, 在 $TIMx_ARR$ 和 $TIMx_CCR1$ 配置后要配置 UG 位进行更新, 并等待 $TI2$ 的触发。在这个例子里, $CC1P=0$, $TIMx_CR1$ 寄存器中的 $DIR=0$, $CMS=00$ 。

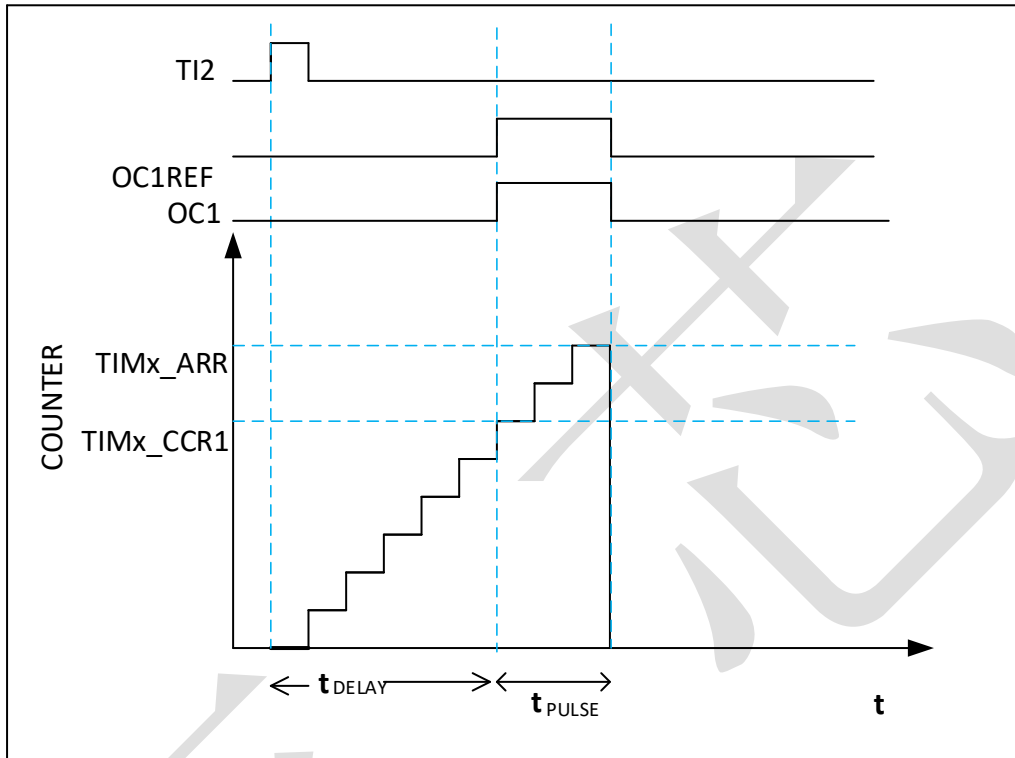


图 16-21 单脉冲模式

16.5.7 特殊情况：OCi 快速使能

在单脉冲模式下, 对 Ti_i 输入脚的边沿检测会设置 CEN 位以启动计数器。然后计数器和比较值间的比较操作产生了单脉冲的输出。但是这些操作需要一定的时钟周期, 因此它限制了可得到的最小延时 t_{DELAY} 。

如果要以最小延时输出波形, 可以设置 $TIMx_CCMRi$ 寄存器中的 $OCiFE$ 位; 此时强制 $OCiREF$ (和 OCx) 直接响应激励而不再依赖比较的结果, 输出的波形与比较匹配时的波形一样。 $OCiFE$ 只在通道配置为 PWM1 和 PWM2 模式的单脉冲模式时起作用。

16.5.8 互补输出和死区插入

$TIM15/16/17$ 能够输出两路互补信号, 并且能够管理输出的瞬时关断和接通。这段时间通常被称为死区, 用户应该根据连接的输出器件和它们的特性 (电平转换的延时、电源开关的延时等) 来调整死区时间。

配置 $TIMx_CCER$ 寄存器中的 $CCiP$ 和 $CCiNP$ 位, 可以为每一个输出独立地选择极性 (主输出 OCi 或互补输出 $OCiN$)。

互补信号 OCi 和 $OCiN$ 通过下列控制位的组合进行控制: $TIMx_CCER$ 寄存器的 $CCiE$ 和 $CCiNE$ 位, $TIMx_BDTR$ 寄存器中的 MOE 、 $OISi$ 、 $OISiN$ 、 $OSSI$ 和 $OSSR$ 位。

特别的是, 在转换到 IDLE 状态时 (MOE 下降到 0) 死区控制被激活。

同时设置 $CCiE$ 和 $CCiNE$ 位将插入死区，如果存在刹车电路，则还要设置 MOE 位。每一个通道都有一个 8 位的死区发生器。如图 16-22 所示，参考信号 $OCiREF$ 可以产生 2 路输出 OCi 和 $OCiN$ 。如果 OCi 和 $OCiN$ 为高有效， $CCiP=0$ ， $CCiNP=0$ ， $MOE=1$ ， $CCiE=1$ ， $CCiNE=1$ ：

OCi 输出信号与参考信号相同，只是它的上升沿相对于参考信号的上升沿有一个延迟。

$OCiN$ 输出信号与参考信号相反，只是它的上升沿相对于参考信号的下降沿有一个延迟。如果延迟大于当前有效的输出宽度（ OCi 或者 $OCiN$ ），则不会产生相应的脉冲。

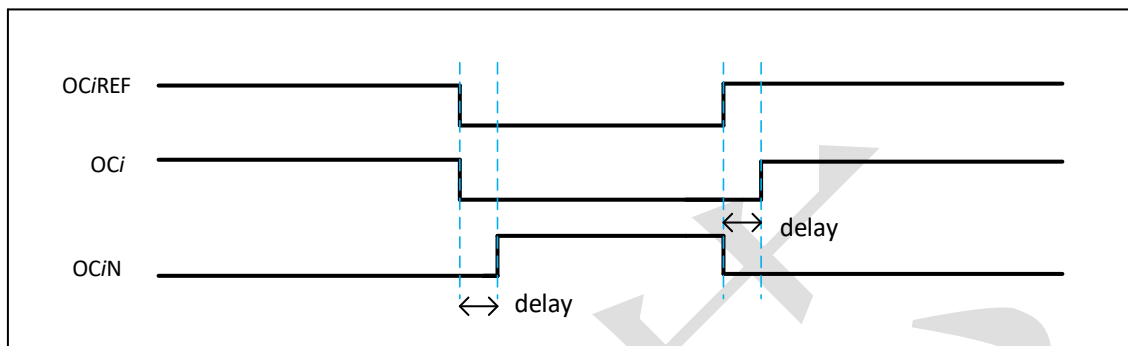


图 16-22 带死区插入的互补输出

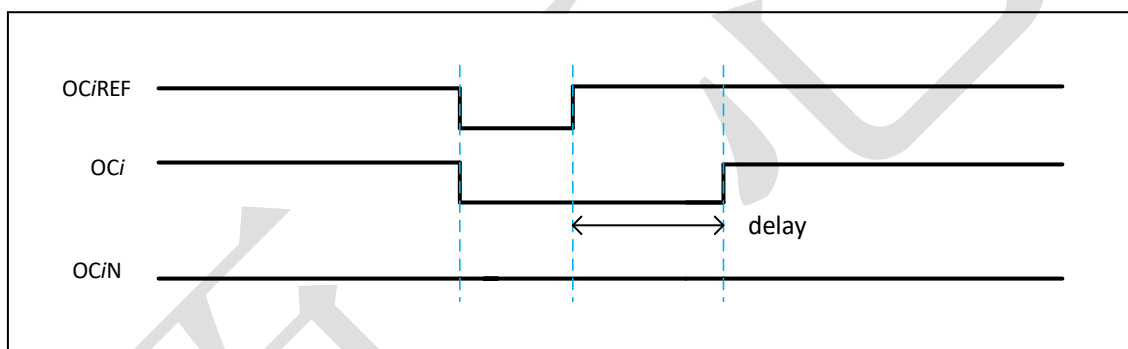


图 16-23 死区波形延迟大于负脉冲

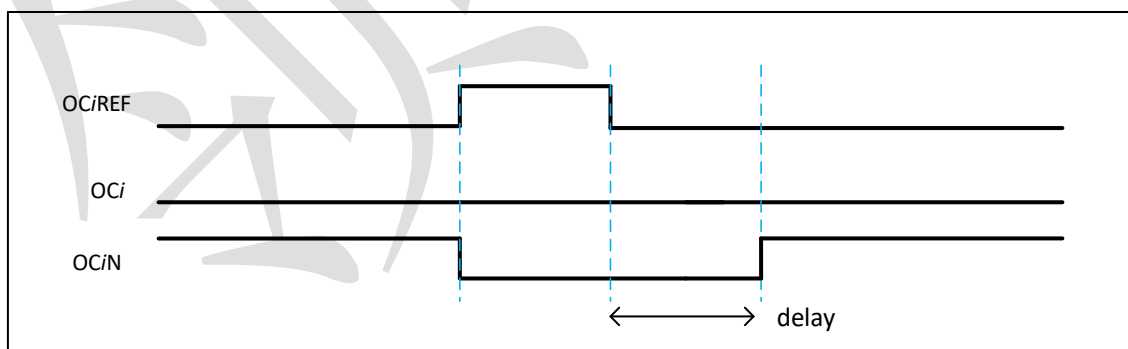


图 16-24 死区波形延迟大于正脉冲

16.5.9 重定向 $OCiREF$ 到 OCi 或 $OCiN$

在输出模式（强置输出、输出比较或 PWM 模式）下，通过配置 $TIMx_CCER$ 寄存器的 $CCiE$ 和 $CCiNE$ 位， $OCiREF$ 可以被重定向到 OCi 或者 $OCiN$ 的输出。

这个功能可以在互补输出的某一路未使能时，在某个输出上送出一个特殊的波形（例如 PWM 或者静态

有效电平)。另一个作用是，让两个输出同时处于无效电平（均未使能），或同时处于有效电平（此时仍然是带死区的互补输出）。

注：当只使能 $OCiN$ ($CCiE=0, CCiNE=1$) 时，它不会反相，而当 $OCiREF$ 变高时立即有效。例如，如果 $CCiNP=0$ ，则 $OCiN=OCiREF$ 。另一方面，当 OCi 和 $OCiN$ 都被使能时 ($CCiE=CCiNE=1$)，当 $OCiREF$ 为高时 OCi 有效；而 $OCiN$ 相反，当 $OCiREF$ 低时 $OCiN$ 变为有效。

16.5.10 刹车功能

刹车功能常用于马达控制中。当使用刹车功能时，依据相应的控制位（ $TIMx_BDTR$ 寄存器中的 MOE 、 $OSSI$ 和 $OSSR$ 位， $TIMx_CR2$ 寄存器中的 $OISi$ 和 $OISiN$ 位），输出使能信号和电平都会被修改。 $TIM15$ 和 $TIM16$ 有 2 路刹车通道，其输入刹车源不同，分别具有各自的刹车控制位； $TIM17$ 有 1 路刹车通道。对于 $TIM15/TIM16$ 的刹车通道 1 和 $TIM17$ 的刹车通道，除了互联配置寄存器 5/6/7（ $SysCtrl_EDU_CFG5/6/7$ ）中给出的刹车源之外，系统中会产生三个特殊的刹车源，分别为发生时钟安全问题、CPU 内核锁死和电压过低触发 LVD 低压检测标志。具体描述请见芯片控制寄存器（ $ChipCtrl_CTRL$ ）寄存器[31:29]位。

定时器上有专门的刹车输入信号，在系统复位完成后，刹车电路被禁止， MOE 位为低。配置 $TIMx_BDTR$ 寄存器中的 BKE 位可以使能刹车功能，刹车输入信号的极性可以通过配置其中的 BKP 位选择。 BKE 和 BKP 可以被同时修改。

MOE 下降沿相对于时钟模块可以是异步的，因此在实际信号（作用在输出端）和同步控制位（在 $TIMx_BDTR$ 寄存器中）之间设置了一个同步电路。这个电路会导致异步信号和同步信号之间产生延迟。特别的，如果当 MOE 从 0 变为 1，读出它之前必须先插入一个延时（空指令）才能读到正确的值。这是因为写入的是异步信号而读的是同步信号。

当刹车触发时：

1. MOE 位被异步清除，输出可通过 $OSSI$ 位被置为不同状态
2. 当 $MOE=0$ 时，输出可以由 $TIMx_CR2$ 寄存器中的 $OISi$ 和 $OISiN$ 位驱动。如果 $OSSI=0$ ，定时器输出不使能，否则输出使能
3. 当使用互补输出时：
 - 输出首先会被置于复位状态（取决于极性配置），这是一个异步行为，所以不需要时钟
 - 如果时钟依旧存在，则会激活死区插入功能，在死区时间后，输出由配置好的 $OISi$ 和 $OISiN$ 位驱动。这里 OCi 和 $OCiN$ 不能同时为有效电平
 - 如果 $OSSI=0$ ，定时器输出不使能，否则输出使能或在 $CCiE/CCiNE$ 变高时拉高
4. 刹车状态标志（ $TIMx_SR$ 寄存器中的 BIF 位）被置起。如果 $TIMx_DIER$ 寄存器中的 BIE 位使能，则会产生中断
5. 如果 $TIMx_BDTR$ 寄存器中的 AOE 位被置位， MOE 位会自动在下一次更新事件 UEV 到来时被置位。否则 MOE 位会一直为低直到被手动写 1
6. 当 AOE 被置位时，输出比较的 $TIMx_CCRi$ 寄存器的值可以通过相应的 $TIMx_CCTRi$ 修调寄存器来更新：如果 $TIMx_CCMRi$ 寄存器中的 $OCiTE=1$ ，当更新事件 UEV 发生时， $TIMx_CCRi$ 寄存器的影子寄存器被 $TIMx_CCTRi$ 更新；如果 $TIMx_CCMRi$ 寄存器中的 $OCiTE=1$ 且 $OCiTUE=1$ ，当更新事件 UEV 发生时， $TIMx_CCRi$ 寄存器的影子寄存器和预装载寄存器都被 $TIMx_CCTRi$ 更新

注：刹车输入是电平有效，因此当刹车输入保持有效时， MOE 不会被置位（无论自动还是手动）。同时， BIF 标志不能被清除。

有两种方式来产生刹车：

- 通过刹车输入并配合 $TIMx_BDTR$ 寄存器中的 BKE 位和 BKP 位
- 通过软件配置 $TIMx_EGR$ 寄存器中的 BG 位

除了刹车输入和输出管理，刹车电路中有写保护来保证应用的安全。它允许冻结几个参数的配置（死区时长、 $OCi/OCiN$ 极性和状态、 $OCiM$ 配置、刹车使能和极性等）。用户可以在 $TIMx_BDTR$ 寄存器中的 $LOCK$ 位中选择三个等级的保护机制，它们只能在 MCU 复位后写入一次。

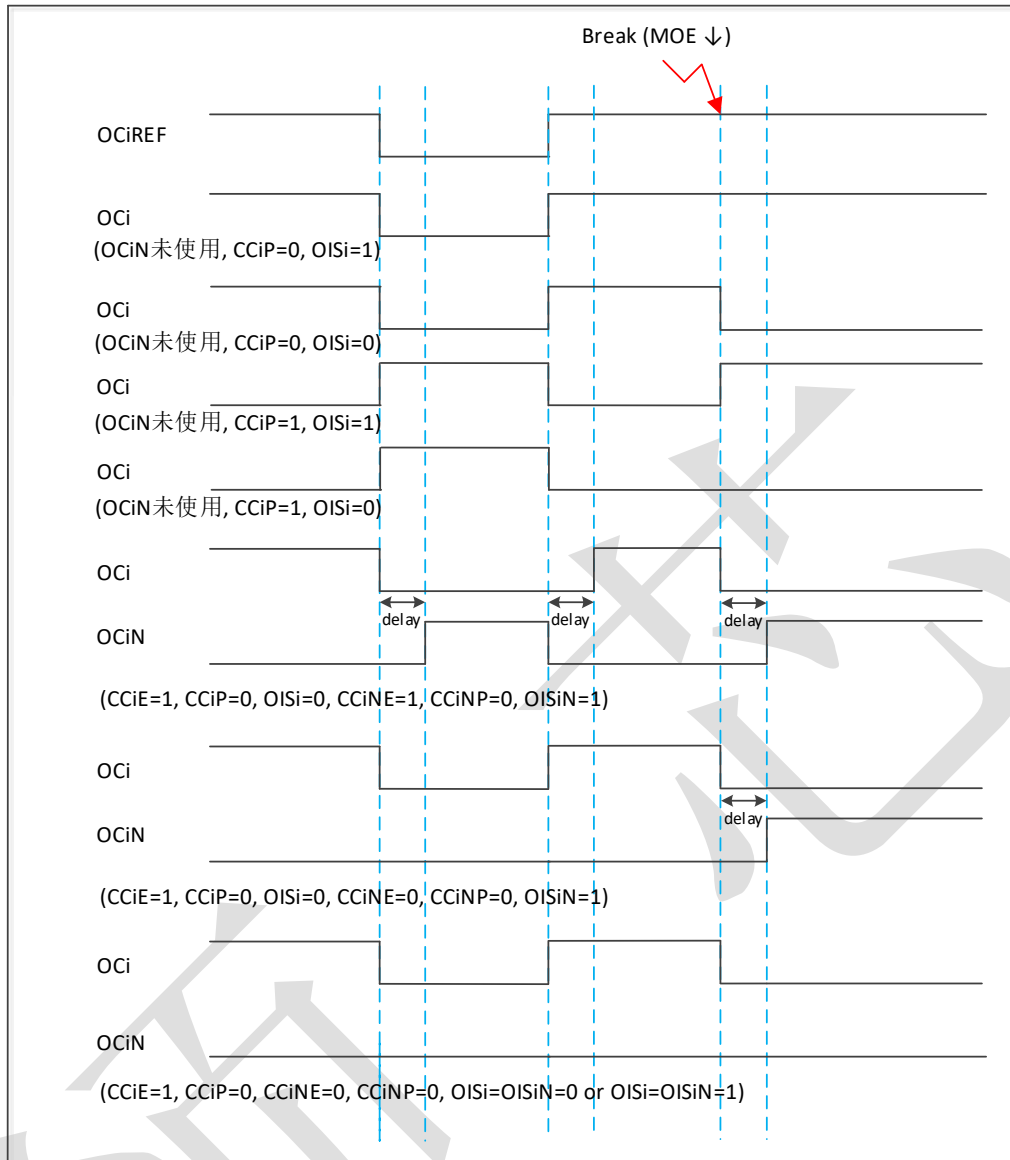


图 16-25 刹车对应的输出

16.6 TIM15/16/17 定时器与外部触发的同步

TIM15/16/17 定时器能够在多种模式下和一个外部的触发同步：复位模式、门控模式、触发模式和触发复位模式。TIM15 计数器允许四种触发输入：ETR（外部触发）；TI1 外部输入；TI2 外部输入；来自芯片内部其他模块。TIM16/17 计数器允许三种触发输入：ETR（外部触发）；TI1 外部输入；来自芯片内部其他模块。TIM16/17 不支持内部触发 ITR。

TIM15/16/17 使用 4 种模式与外部的触发信号同步：标准触发模式、复位模式、门控模式和触发复位模式。

16.6.1 标准触发模式

计数器的使能依赖于选中的触发输入事件。在下面的例子中，TIM15 计数器在 TI2 输入的上升沿开始向上计数：

- 配置通道 2 用于检测 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，保持寄存器 CCMR1 中的 IC2F=0000）
- 触发操作中不使用捕获预分频器，不需要配置；寄存器 CCMR1 中的 CC2S=01，用于选择输入捕获源
- 配置 TIMx 的 CCER 寄存器的 CC2P=0，选择上升沿做为触发条件
- 配置 TIMx 的 SMCR 寄存器的 SMS=110，选择定时器为触发模式
- 配置 TIMx 的 SMCR 寄存器的 TS=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 TIMx_SR 寄存器的 TIF 位被置起。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。

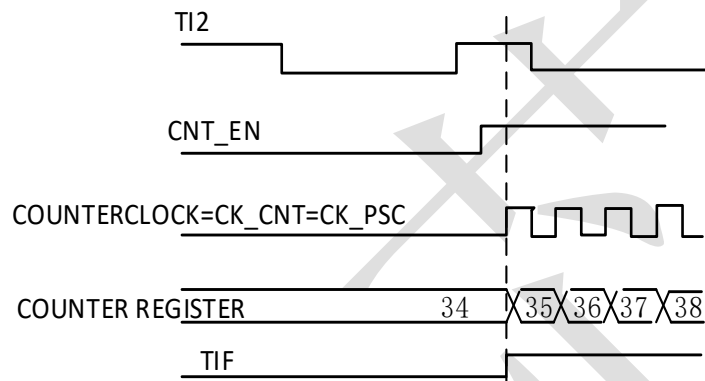


图 16-26 标准触发模式下的控制电路

16.6.2 复位模式

在发生一个触发输入事件时，计数器和它的预分频器能够重新被初始化；同时，如果 TIMx_CR1 寄存器的 URS 位为低，还产生一个更新事件 UEV；然后所有的预装载寄存器（ARR、CCR）都被更新。在以下的例子中，TI1 输入端的上升沿导致向上计数器被清零：

- 配置通道 1 用于检测 TI1 的上升沿；配置输入滤波器的带宽（在本例中，不需要任何滤波器，因此保持 IC1F=0000）
- 触发操作中不使用捕获预分频器，所以不需要配置；寄存器 CCMR1 中的 CC1S=01，用于选择输入捕获源
- 配置 TIMx 的 CCER 寄存器的 CC1P=0 来选择极性（只检测上升沿）
- 配置 TIMx 的 SMCR 寄存器的 SMS=100，选择定时器为复位模式
- 配置 TIMx 的 SMCR 寄存器的 TS=101，选择 TI1 作为输入源
- 配置 TIMx_CR1 寄存器的 CEN=1，启动计数器
- 计数器开始依据内部时钟计数，然后正常计数直到 TI1 出现一个上升沿；此时，计数器被清零然后从 0 重新开始计数。同时，TIMx_SR 寄存器的 TIF 位被置起，如果使能了中断（TIMx_DIER 寄存器的 TIE 位），则产生一个中断请求

下图显示当自动重装载寄存器 TIMx_ARR=0x36 时的动作。在 TI1 上升沿和计数器的实际复位之间的延时取决于 TI1 输入端的重同步电路。

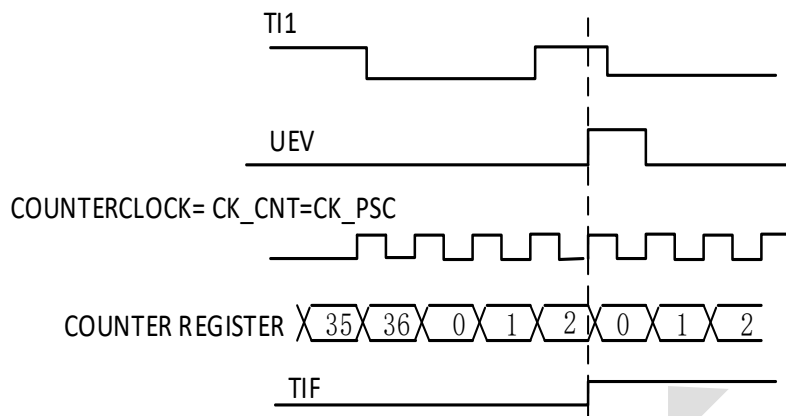


图 16-27 复位模式下的控制电路

16.6.3 门控模式

计数器由选中的输入端信号的电平使能。在以下的例子中，计数器只在 TI1 为低时向上计数：

- 配置通道 1 用于检测 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 IC1F=0000）
- 触发操作中不使用捕获预分频器，所以不需要配置；寄存器 CCMR1 中的 CC1S=01，用于选择输入捕获源
- 配置 TIMx_CCER 寄存器的 CC1P=1 来确定极性（只检测低电平）
- 配置 TIMx_SMCR 寄存器的 SMS=101，选择定时器为门控模式
- 配置 TIMx_SMCR 寄存器中 TS=101，选择 TI1 作为输入源
- 配置 TIMx_CR1 寄存器的 CEN=1，启动计数器（门控模式下，如果 CEN=0，则计数器不能启动，无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，TIMx_SR 寄存器的 TIF 位都会被置起。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。

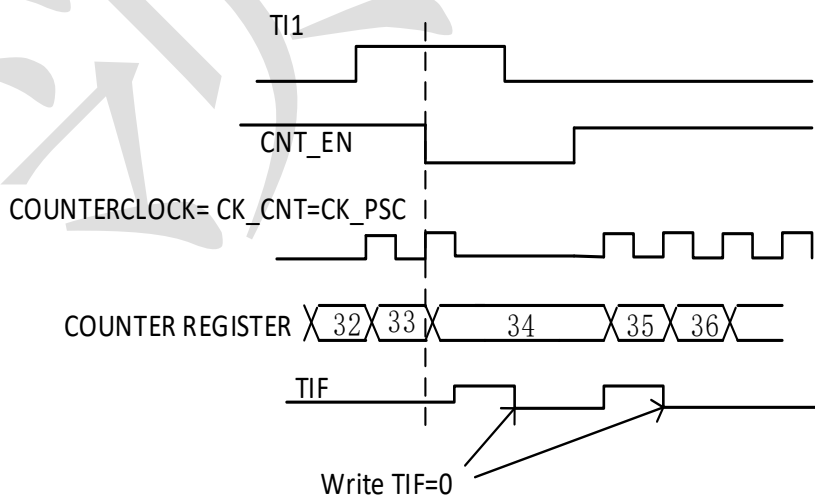


图 16-28 门控模式下的控制电路

16.6.4 触发复位模式

计数器的使能依赖于选中的触发输入事件，在发生一个触发输入事件时，计数器和它的预分频器能够重新被初始化；同时，如果 TIMx_CR1 寄存器的 URS 位为低，还产生一个更新事件 UEV；然后所有的预装载寄存器（ARR、CCR）都被更新。这种模式可视为是标准触发模式和复位模式的结合，其配置与标准触发模式基本一致，不同之处在于 TIMx 的 SMCR 寄存器的 SMS=1000。

16.6.5 外部时钟模式 2 及触发模式

外部时钟模式 2 可以与另一个输入信号的触发模式一起使用。这时，ETR 信号被用作外部时钟的输入，另一个输入信号可用作触发模式（支持标准触发模式，复位模式、门控模式和触发复位模式）。请注意不能把 ETR 配置成 TRGI（通过 TIMx_SMCR 寄存器的 TS 位）。

在下面的例子中，一旦在 TI1 上出现一个上升沿，计数器即在 ETR 的每一个上升沿向上计数一次，通过 TIMx_SMCR 寄存器配置外部触发输入电路。

- 首先配置 ETR：TIMx_SMCR 寄存器中配置 ETF=0000 禁止滤波器，配置 ETPS=00 禁止预分频，配置 ETP=0 监测 ETR 信号的上升沿，配置 ECE=1 使能外部时钟模式 2
- 配置通道 1 用于检测 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 IC1F=0000）
- 触发操作中不使用捕获预分频器，所以不需要配置；寄存器 CCMR1 中的 CC1S=01，用于选择输入捕获源
- 配置 TIMx 的 CCER 寄存器的 CC1P=0，选择上升沿做为触发条件
- 配置 TIMx 的 SMCR 寄存器的 SMS=110，选择定时器为触发模式
- 配置 TIMx 的 SMCR 寄存器的 TS=101，选择 TI1 作为输入源

当 TI1 上出现一个上升沿时，同时 TIMx_SR 寄存器的 TIF 位被置起，计数器开始在 ETR 的上升沿计数。TI1 信号的上升沿和计数器实际时钟之间的延时取决于 TI1 输入端的重同步电路。ETR 信号的上升沿和计数器实际时钟之间的延时取决于 ETRP 输入端的重同步电路。

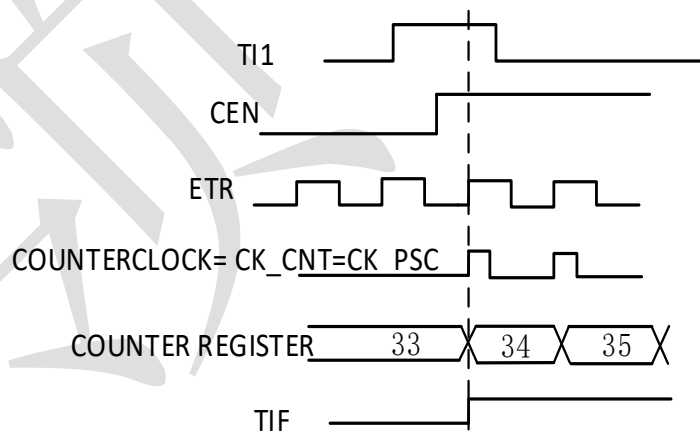


图 16-29 外部时钟模式 2 且触发模式下的控制电路

16.7 TIM15/16/17 与其他定时器的联接

在芯片中，各个定时器在内部互相联接，用于定时器的同步或连接。当某个定时器配置成主模式时，可以输出触发信号（TRGO）到那些配置为从模式的定时器来完成复位、启动和停止的操作，或者作为那些定时器的驱动时钟。TIM16/17 不支持触发输出信号 TRGO。

TIM15 的输入触发可以来自芯片内部的其他定时器、比较器 ACMP0/1/2 输出和 ADC;TIM15 的输出 TRGO 可触发芯片内部的 ADC、DAC 和其他定时器;TIM15 的输出 OC_i 可触发芯片内部的比较器 ACMP0/1/2 和 ADC;TIM15 的特殊输出 TIM15_oc1_trig 可触发其他定时器,它与 CEN 位相关,具体详见 TIM15_CR2 寄存器 TrgIS 位说明。

TIM16/17 的输入触发可以来自芯片内部的其他定时器、比较器 ACMP0/1/2 输出和 ADC;TIM16/17 的输出 OC_i 可触发芯片内部的 ADC;TIM16/17 的特殊输出 TIM16/17_oc1_trig 可触发其他定时器,它与 CEN 位相关,具体详见 TIM16/17_CR2 寄存器 TrgIS 位说明。

表 16-1 TIM15 触发与级联表

信号名称	信号描述	来源	选择寄存器
内部触发 ITR			
ITR0	内部触发 0	tim1_trgo	-
ITR1	内部触发 1	tim3_trgo	-
ITR2	内部触发 2	tim16_oc1_trig	-
ITR3	内部触发 3	tim17_oc1_trig	-
外部 Timer 输入 TI			
TI1	外部 Timer 输入 1	tim15_gpio_ti1/adc_awd/cp0_out/ cp1_out/cp2_out/tim1_oc1_trig/ tim3_oc1_trig/tim14_oc1_trig	SysCtrl_EDU_CFG3
TI2	外部 Timer 输入 2	tim15_gpio_ti2/adc_awd/cp0_out/ cp1_out/cp2_out/tim1_oc1_trig/ tim3_oc1_trig/tim14_oc1_trig	
外部触发输入 ETR			
ETR	外部触发	adc_awd/cp0_out/ cp1_out/cp2_out/tim1_oc1_trig/ tim3_oc1_trig/tim14_oc1_trig	SysCtrl_EDU_CFG3

注: 请参考 SysCtrl_EDU_CFG3 寄存器定义。

表 16-2 TIM16 触发与级联表

信号名称	信号描述	来源	选择寄存器
内部触发 ITR			
ITR0	内部触发 0	-	-
ITR1	内部触发 1	-	-
ITR2	内部触发 2	-	-
ITR3	内部触发 3	-	-
外部 Timer 输入 TI			
TI1	外部 Timer 输入 1	tim16_gpio_ti1/adc_awd/cp0_out/ cp1_out/cp2_out/tim1_trgo/ tim3_trgo/tim15_trgo	SysCtrl_EDU_CFG3
外部触发输入 ETR			
ETR	外部触发	adc_awd/cp0_out/ cp1_out/cp2_out/tim1_trgo/ tim3_trgo/tim15_trgo	SysCtrl_EDU_CFG3

注：请参考 SysCtrl_EDU_CFG3 寄存器定义。

表 16-3 TIM17 触发与级联表

信号名称	信号描述	来源	选择寄存器
内部触发 ITR			
ITR0	内部触发 0	-	-
ITR1	内部触发 1	-	-
ITR2	内部触发 2	-	-
ITR3	内部触发 3	-	-
外部 Timer 输入 TI			
TI1	外部 Timer 输入 1	tim17_gpio_ti1/adc_awd/cp0_out/ cp1_out/cp2_out/tim1_trgo/ tim3_trgo/tim15_trgo	SysCtrl_EDU_CFG3
外部触发输入 ETR			
ETR	外部触发	adc_awd/cp0_out/ cp1_out/cp2_out/tim1_trgo/ tim3_trgo/tim15_trgo	SysCtrl_EDU_CFG3

注：请参考 SysCtrl_EDU_CFG3 寄存器定义。

16.8 TIM15/16/17 中断

TIM15 有 6 个中断请求源，分别映射到 2 个中断矢量上：

- 刹车中断
- 更新事件中断
- 触发中断
- COM 事件中断
- 输入捕获/输出比较 1 中断
- 输入捕获/输出比较 2 中断

TIM16/17 有 4 个中断请求源，分别映射到 2 个中断矢量上：

- 刹车中断
- 更新事件中断
- COM 事件中断
- 输入捕获/输出比较 1 中断

为了使用中断特性，对每个被使用的中断通道，设置 TIMx_DIER 寄存器中相应的中断使能位：BIE、TIE、COMIE、CCiIE 和 UIE 位。

通过设置 TIMx_EGR 寄存器中的相应位，也可以用软件产生上述各个中断源。

16.9 TIM15/16/17 寄存器描述

表 16-4 TIM15/16/17 相关寄存器表

名称	说明	读写权限	复位值	TIM15 字节地址	TIM16 字节地址	TIM17 字节地址
CR1	控制寄存器 1	R/W	0x0000_0000	0x4001_B000	0x4001_C000	0x4001_E000
CR2	控制寄存器 2	R/W	0x0000_0000	0x4001_B004	0x4001_C004	0x4001_E004
SMCR	从模式控制寄存器	R/W	0x0000_0000	0x4001_B008	0x4001_C008	0x4001_E008
DIER	DMA 和中断控制寄存器	R/W	0x0000_0000	0x4001_B00C	0x4001_C00C	0x4001_E00C
SR	状态寄存器	R/W	0x0000_0000	0x4001_B010	0x4001_C010	0x4001_E010
EGR	事件产生寄存器	R/W	0x0000_0000	0x4001_B014	0x4001_C014	0x4001_E014
CCMR1	捕获/比较模式寄存器 1	R/W	0x0000_0000	0x4001_B018	0x4001_C018	0x4001_E018
CCER	捕获/比较使能寄存器	R/W	0x0000_0000	0x4001_B020	0x4001_C020	0x4001_E020
CNT	计数寄存器	R/W	0x0000_0000	0x4001_B024	0x4001_C024	0x4001_E024
PSC	预分频寄存器	R/W	0x0000_0000	0x4001_B028	0x4001_C028	0x4001_E028
ARR	自动重装载寄存器	R/W	0x0000_0000	0x4001_B02C	0x4001_C02C	0x4001_E02C
RCR	重复计数寄存器	R/W	0x0000_0000	0x4001_B030	-	-
CCR1	捕获/比较寄存器 1	R/W	0x0000_0000	0x4001_B034	0x4001_C034	0x4001_E034
CCR2	捕获/比较寄存器 2	R/W	0x0000_0000	0x4001_B038	-	-
BDTR	刹车死区控制寄存器	R/W	0x0000_0000	0x4001_B044	0x4001_C044	0x4001_E044
DCR	DMA 控制寄存器	R/W	0x0000_0000	0x4001_B048	0x4001_C048	0x4001_E048
DMAR	DMA 传输寄存器	R/W	0x0000_0000	0x4001_B04C	0x4001_C04C	0x4001_E04C
CCTR1	比较修调寄存器 1	R/W	0x0000_0000	0x4001_B064	0x4001_C064	0x4001_E064
CCTR2	比较修调寄存器 2	R/W	0x0000_0000	0x4001_B068	-	-

注：x 表示不确定；- 表示未实现；u 表示不受复位影响；c 表示该位复位值因复位源而异；R 只读；W 只写；R-0 表示只读且读 0；R/W 表示可读写（以后章节同上述）。

16.9.1 CR1 控制寄存器 1 (TIM15/16/17_CR1)

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
Res.							AS	OCF	Res.			TI2E	TI1E	ETRE	FTE
							rw	rw				rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
						rw	rw		rw		rw	rw	rw	rw	rw

Bits	31:25	保留，必须保持复位值
Bit	24	AS：影子寄存器读使能
		该位指定当 MCU 读取 TIMx_PSC/TIMx_ARR/TIMx_CCRi 寄存器的时候，是读取影子寄存器还是预装载寄存器

	0: 选择预装载寄存器的值（默认）
	1: 选择影子寄存器的值
Bit 23	OCF: 输出异步抗干扰滤波
	0: 输出滤波禁止（默认）
	1: 输出滤波使能
Bits 22:20	保留，必须保持复位值
Bit 19	TI2E: TI2 输入检测/滤波使能（仅用于 TIM15）
	0: TI2 输入检测/滤波禁止（默认）
	1: TI2 输入检测/滤波使能
Bit 18	TI1E: TI1 输入检测/滤波使能
	0: TI1 输入检测/滤波禁止（默认）
	1: TI1 输入检测/滤波使能
Bit 17	ETRE: 外部触发输入检测/滤波使能
	0: ETR 输入检测/滤波禁止（默认）
	1: ETR 输入检测/滤波使能
Bit 16	FTE: 数字滤波器（ETR、Tii）的 DTS 时钟使能
	0: 数字滤波器的 DTS 时钟禁止（默认）
	1: 数字滤波器的 DTS 时钟使能
Bits 15:10	保留，必须保持复位值
Bits 9:8	CKD[1:0]: CK_INT 时钟和死区/采样时钟（CK_DTS）的分频系数，DTS 时钟供给死区发生器和数字滤波器（ETR、Tii）使用
	00: $t_{DTS} = t_{CK_INT}$ （默认）
	01: $t_{DTS} = 2 * t_{CK_INT}$
	10: $t_{DTS} = 4 * t_{CK_INT}$
	11: 保留
Bit 7	ARPE: 自动预装载允许位
	0: TIMx_ARR 寄存器可以被直接写入（默认）
	1: TIMx_ARR 寄存器通过预装载寄存器更新
Bits 6:5	CMS[1:0]: 选择中央对齐模式
	00: 边沿对齐模式，计数器依据方向位（DIR）向上或向下计数。（默认）
	01: 中央对齐模式 1，计数器交替地向上和向下计数。只有在计数器向下计数时，输出比较中断标志位（TIMx_CCMRi 寄存器中 CCIS=00 条件下）才会被置 1。
	10: 中央对齐模式 2，计数器交替地向上和向下计数。只有在计数器向上计数时，输出比较中断标志位（TIMx_CCMRi 寄存器中 CCIS=00 条件下）才会被置 1。
	11: 中央对齐模式 3，计数器交替地向上和向下计数。在计数器向上或向下计数时，输出比较中断标志位（TIMx_CCMRi 寄存器中 CCIS=00 条件下）会被置 1。
	注: 在计数器开启时（CEN=1），不允许从边沿对齐模式切换到中央对齐模式。
Bit 4	DIR: 计数器方向
	0: 计数器向上计数（默认）
	1: 计数器向下计数
	注: 当计数器配置为中央对齐模式时，该位为只读。
Bit 3	OPM: 单脉冲模式
	0: 在发生更新事件时，计数器不停止（默认）

	1: 在发生下一次更新事件时, 计数器停止 (清除 CEN 位)
	<i>注: TIM15/16/17 的 OPM 模式只有在至少有一个通道处于输出模式时才会生效。</i>
Bit 2	URS: 更新请求源
	0: 如果 UDIS 允许产生更新事件, 则下述任一事件产生一个更新中断 (默认):
	- 计数器上溢/下溢
	- 软件设置 UG 位
	- 时钟/触发控制器产生的硬件复位
	1: 如果 UDIS 允许产生更新事件, 则只有当计数器上溢/下溢时才产生更新中断
Bit 1	UDIS: 禁止更新
	0: 一旦下列事件发生, 产生更新事件 (默认):
	- 计数器溢出/下溢
	- 软件设置 UG 位
	- 时钟/触发控制器产生的硬件复位
	1: 不产生更新事件, 影子寄存器 (ARR、PSC、CCR) 保持它们的值。如果 UG 位被配置或时钟/触发控制器发出了一个硬件复位, 则计数器和预分频器被重新初始化
Bit 0	CEN: 计数器使能位
	0: 计数器禁止 (默认)
	1: 计数器使能
	<i>注: 在软件配置了 CEN 后, 外部时钟模式、复位模式和门控模式才能工作。触发模式可以自动通过硬件启动。在门控模式下, 该位读回值表示门控状态。</i>

16.9.2 CR2 控制寄存器 2 (TIM15/16/17_CR2)

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
Res.															TrigS
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				OIS2N	OIS2	OIS1N	OIS1	Res.	MMS[2:0]			CCDS	CCUS	Res.	CCPC
				rw	rw	rw	rw		rw			rw	rw		rw

Bits 31:17	保留, 必须保持复位值
Bit 16	TrigS: TIMx_oc1_trig 触发源选择
	0: 触发源选择为 OC1&CEN (默认)
	1: 触发源选择为 CEN
Bits 15:12	保留, 必须保持复位值
Bit 11	OIS2N: 输出空闲状态 2 (OC2N 输出), 参考 OIS1N (仅用于 TIM15)
Bit 10	OIS2: 输出空闲状态 2 (OC2 输出), 参考 OIS1 (仅用于 TIM15)
Bit 9	OIS1N: 输出空闲状态 1 (OC1N 输出)
	0: OC1N 的空闲电平为 0 (默认)
	1: OC1N 的空闲电平为 1

	<i>注：已经设置了 LOCK (TIMx_BDTR 寄存器) 级别 1、2 或 3 后，该位不能被修改。</i>
Bit 8	OIS1: 输出空闲状态 1 (OC1 输出)
	0: OC1 的空闲电平为 0 (默认)
	1: OC1 的空闲电平为 1
	<i>注：已经设置了 LOCK (TIMx_BDTR 寄存器) 级别 1、2 或 3 后，该位不能被修改。</i>
Bit 7	保留，必须保持复位值
Bits 6:4	MMS[2:0]: 主模式选择 (仅用于 TIM15)
	用于选择在主模式下送到其他模块的同步信号 (TRGO)
	000: 复位 - 软件设置 UG 位或时钟/触发控制器产生的硬件复位被用作触发输出 (TRGO)。如果触发输入 (时钟/触发控制器配置为复位模式) 产生复位，则 TRGO 上的信号会延迟到与实际的复位同步。(默认)
	001: 使能 - 计数器使能信号被用作触发输出 (TRGO)。其用于同时启动多个定时器，或在一段时间内控制从定时器和其他模块。计数器使能信号是通过 CEN 控制位或门控模式下的触发输入信号产生。除非选择了主/从模式 (见 TIMx_SMCR 寄存器中 MSM 位的描述)，否则当计数器使能信号受控于触发输入时，TRGO 上会有一个延迟。
	010: 更新 - 更新事件被用作触发输出 (TRGO)。在这种模式下，一个主定时器可用作为一个从定时器的预分频器。
	011: 比较脉冲 - 一旦发生一次捕获或一次比较成功，当 CC1IF 标志被置 1 时 (即使它已经为高)，触发输出送出一个正脉冲 (TRGO)。
	100: 比较 - OC1REF 信号被用作触发输出 (TRGO)。
	101: 比较 - OC2REF 信号被用作触发输出 (TRGO)。
	110: 保留
	111: 保留
Bit 3	CCDS: 捕获/比较 DMA 选择
	0: 当 CCx 事件发生的时候发送 CCx DMA 请求 (默认)
	1: 当更新事件发生的时候发送 CCx DMA 请求
Bit 2	CCUS: 捕获/比较控制位的更新控制选择
	0: 当捕获/比较的控制位为预装载时 (CCPC=1)，只有在 COMG 位置 1 的时候这些控制位才被更新 (默认)
	1: 当捕获/比较的控制位为预装载时 (CCPC=1)，只有在 COMG 位置 1 或 TRGI 上升沿的时候这些控制位才被更新
	<i>注：该位只对拥有互补输出的通道有效。</i>
Bit 1	保留，必须保持复位值
Bit 0	CCPC: 捕获/比较预装载控制位
	0: CCiE、CCiNE、CCiP、CCiNP 位 (TIMx_CCER 寄存器) 和 OCiM 位 (TIMx_CCMRi 寄存器) 不是预装载的 (默认)
	1: CCiE、CCiNE、CCiP、CCiNP 和 OCiM 位是预装载的；设置该位后，它们只在设置了 COM 事件 (COMG 位置 1 或 TRGI 上升沿，由 CCUS 位控制) 发生后被更新
	<i>注：该位只对拥有互补输出的通道有效。</i>

16.9.3 SMCR 模式控制寄存器 (TIM15/16/17_SMCR)

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
Res.															SMS[3]
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			Res.	SMS[2:0]		
rw	rw	rw		rw				rw	rw				rw		

Bits 31:17	保留, 必须保持复位值
Bit 16	SMS[3] : 从模式选择的 bit 3 参考 SMS[2:0]的描述
Bit 15	ETP : 外部触发极性 0: ETR 不反相, 即高电平或上升沿有效 (默认) 1: ETR 反相, 即低电平或下降沿有效
Bit 14	ECE : 外部时钟使能, 用于使能外部时钟模式 2 0: 外部时钟模式 2 禁止 (默认) 1: 外部时钟模式 2 使能, 计数器的时钟为 ETRF 的有效边沿 注 1 : ECE 位置 1 的效果与选择把 TRGI 连接到 ETRF 的外部时钟模式 1 相同 (TIMx_SMCR 寄存器中, SMS=111, TS=111)。 注 2 : 外部时钟模式 2 可与下列模式同时使用: 标准触发模式、复位模式、门控模式和触发复位模式。但是, 此时 TRGI 决不能与 ETRF 相连 (TIMx_SMCR 寄存器中, TS 不能为 111)。 注 3 : 如果外部时钟模式 1 与外部时钟模式 2 同时使能, 外部时钟输入为 ETRF。
Bits 13:12	ETPS : 外部触发预分频器 外部触发信号 EPRP 的频率最大不能超过定时器时钟 CK_INT 频率的 1/4。可用预分频器来降低 ETRP 的频率, 当 EPRP 的频率很高时非常有用。 00: 预分频器关闭 (默认) 01: EPRP 的频率/2 10: EPRP 的频率/4 11: EPRP 的频率/8
Bits 11:8	ETF[3:0] : 外部触发滤波器选择 该位域定义了 ETRP 的采样频率及数字滤波器的长度。数字滤波器由一个事件计数器组成, 只有发生了 N 个连续事件后输出的跳变才被认为有效。这里的 DTS 时钟是 CK_INT 经过 CR1 寄存器中 CKD 配置分频的。 0000: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}$, 无滤波器 (默认) 0001: 采样频率 $f_{\text{SAMPLING}} = f_{\text{CK_INT}}$, N=2 0010: 采样频率 $f_{\text{SAMPLING}} = f_{\text{CK_INT}}$, N=4 0011: 采样频率 $f_{\text{SAMPLING}} = f_{\text{CK_INT}}$, N=8 0100: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/2$, N=6 0101: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/2$, N=8 0110: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$, N=6 0111: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$, N=8 1000: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/8$, N=6

	1001: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}} / 8$, N=8
	1010: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}} / 16$, N=5
	1011: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}} / 16$, N=6
	1100: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}} / 16$, N=8
	1101: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}} / 32$, N=5
	1110: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}} / 32$, N=6
	1111: 采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}} / 32$, N=8
	<i>注: 请注意采样频率中 $f_{\text{CK_INT}}$ 和 f_{DTS} 的差别。</i>
Bit 7	MSM: 主/从模式 (仅用于 TIM15)
	0: 无作用 (默认)
	1: 触发输入 (TRGI) 上的事件被延迟了, 以允许 TIMx 与它的从定时器间通过 TRGO 完美同步
Bits 6:4	TS[2:0]: 选择同步计数器的触发输入
	000: 内部触发 ITR0 (默认) (仅用于 TIM15)
	001: 内部触发 ITR1 (仅用于 TIM15)
	010: 内部触发 ITR2 (仅用于 TIM15)
	011: 内部触发 ITR3 (仅用于 TIM15)
	100: TI1 的边沿检测器 (TI1F_ED)
	101: 滤波后的定时器输入 1 (TI1FP1)
	110: 滤波后的定时器输入 2 (TI2FP2) (仅用于 TIM15)
	111: 外部触发输入 (ETRF)
	<i>注: 这些位只能在未用到 (如 SMS=000) 时被改变, 以避免在改变时产生错误的边沿检测。</i>
Bit 3	保留, 必须保持复位值
Bits 2:0	SMS[2:0]: 从模式选择
	当选择外部信号时, 触发信号 (TRGI) 的有效边沿与外部输入的极性相关。
	000: 从模式禁止 - 如果 CEN=1, 则预分频器直接由内部时钟驱动。(默认)
	001: 保留
	010: 保留
	011: 保留
	100: 复位模式 - 在选中的触发输入 (TRGI) 的上升沿时重新初始化计数器, 并且产生一个更新寄存器的信号。
	101: 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。
	110: 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。
	111: 外部时钟模式 1 - 选中的触发输入 (TRGI) 的上升沿驱动计数器。
	1000 (结合 SMS[3]): 触发复位模式 - 计数器在触发输入 (TRGI) 的上升沿启动, 重新初始化计数器, 并且产生一个更新寄存器的信号。
	<i>注: 如果 TI1F_ED 被选为触发输入 (TS=100) 时, 不要使用门控模式。这是因为 TI1F_ED 在每次 TI1F 变化时只是输出一个脉冲, 然而门控模式要检查触发输入的电平。</i>

16.9.4 DIER DMA 和中断控制寄存器 (TIM15/16/17_DIER)

地址偏移: 0x0C

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	COMDE	Res.		CC2DE	CC1DE	UDE	BIE	TIE	COMIE	Res.		CC2IE	CC1IE	UIE
	rw	rw			rw	rw	rw	rw	rw	rw			rw	rw	rw

Bits	31:15	保留, 必须保持复位值
Bit	14	TDE: 触发 DMA 请求使能
		0: 触发 DMA 请求禁止 (默认)
		1: 触发 DMA 请求使能
Bit	13	COMDE: COM 的 DMA 请求使能
		0: COM 的 DMA 请求禁止 (默认)
		1: COM 的 DMA 请求使能
Bits	12:11	保留, 必须保持复位值
Bit	10	CC2DE: 捕获/比较 2 的 DMA 请求使能 (仅用于 TIM15)
		0: 捕获/比较 2 的 DMA 请求禁止 (默认)
		1: 捕获/比较 2 的 DMA 请求使能
Bit	9	CC1DE: 捕获/比较 1 的 DMA 请求使能
		0: 捕获/比较 1 的 DMA 请求禁止 (默认)
		1: 捕获/比较 1 的 DMA 请求使能
Bit	8	UDE: 更新的 DMA 请求使能
		0: 更新的 DMA 请求禁止 (默认)
		1: 更新的 DMA 请求使能
Bit	7	BIE: 刹车中断使能
		0: 刹车中断禁止 (默认)
		1: 刹车中断使能
Bit	6	TIE: 触发中断使能
		0: 触发中断禁止 (默认)
		1: 触发中断使能
Bit	5	COMIE: COM 中断使能
		0: COM 中断禁止 (默认)
		1: COM 中断使能
Bits	4:3	保留, 必须保持复位值
Bit	2	CC2IE: 捕获/比较 2 中断使能 (仅用于 TIM15)
		0: 捕获/比较 2 中断禁止 (默认)
		1: 捕获/比较 2 中断使能
Bit	1	CC1IE: 捕获/比较 1 中断使能

	0: 捕获/比较 1 中断禁止 (默认)
	1: 捕获/比较 1 中断使能
Bit 0	UIE : 更新中断使能
	0: 更新中断禁止 (默认)
	1: 更新中断使能

16.9.5 SR 状态寄存器 (TIM15/16/17_SR)

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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.-					CC2OF	CC1OF	B2IF	BIF	TIF	COMIF	Res.		CC2IF	CC1IF	UIF
					r	r	rw	rw	rw	rw			rw	rw	rw

Bits 31:11	保留, 必须保持复位值
Bit 10	CC2OF : 捕获/比较 2 过捕获/过比较标志, 参考 CC1OF 位 (仅用于 TIM15)
Bit 9	CC1OF : 捕获/比较 1 过捕获/过比较标志 该位可由硬件置 1, 软件向 CC1IF 位写 0 可清除该位 0: 无过捕获/过比较产生 (默认) 1: 计数器的值被捕获或匹配到 TIM1_CCR1 寄存器时 CC1IF 已经置 1
Bit 8	B2IF : 刹车 2 中断标志 (仅用于 TIM15/16) 刹车 2 输入一旦有效, 该位由硬件置 1, 刹车 2 输入无效后可以由软件写 0 清除 0: 无刹车事件产生 (默认) 1: 在刹车 2 输入上检测到有效电平
Bit 7	BIF : 刹车中断标志 刹车输入一旦有效, 该位由硬件置 1, 刹车输入无效后可以由软件写 0 清除 0: 无刹车事件产生 (默认) 1: 刹车输入上检测到有效电平
Bit 6	TIF : 触发中断标志 当发生触发事件(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)时该位由硬件置 1, 软件写 0 可清除该位 0: 无触发事件产生 (默认) 1: 触发中断挂起
Bit 5	COMIF : COM 中断标志 一旦产生 COM 事件(当捕获/比较控制位 CCiE、CCiNE、OCiM 被更新)时硬件置位该寄存器位, 软件写 0 可清除该位 0: 无 COM 事件产生 (默认) 1: COM 中断挂起
Bits 4:3	保留, 必须保持复位值

Bit 2	CC2IF: 捕获/比较 2 中断标志, 参考 CC1IF 位 (仅用于 TIM15)
Bit 1	CC1IF: 捕获/比较 1 中断标志
	如果通道 CC1 配置为输出模式:
	当计数器值与比较值匹配时该位由硬件置 1, 但在中央对齐模式下除外 (参考 TIMx_CR1 寄存器的 CMS 位)。软件写 0 可清除该位, 但是当 CC1OF 也为 1 时, 需要清除 2 次。
	0: 无匹配发生 (默认)
	1: TIMx_CNT 的值与 TIMx_CCR1 的值匹配
	注: 在中央对齐模式下, 当计数器值为 0 时, 向上计数, 当计数器值为 ARR 时, 向下计数 (它从 0 向上计数到 ARR-1, 再由 ARR 向下计数到 1)。因此, 对所有的 SMS 位值, 这两个值都不会置位 CC1IF。但是, 如果 CCR1>ARR, 则当 CNT 达到 ARR 值时, CC1IF 置 1。
	如果通道 CC1 配置为输入模式:
	当捕获事件发生时该位由硬件置 1, 软件写 0 或通过读 TIMx_CCR1 可清除该位, 但是当 CC1OF 也为 1 时, 需要清除 2 次。
	0: 无输入捕获产生 (默认)
	1: 计数器值已被捕获至 TIMx_CCR1 (在 IC1 上检测到与所选极性相同的边沿)
Bit 0	UIF: 更新中断标志
	当产生更新事件时该位由硬件置 1, 软件写 0 可清除该位
	0: 无更新事件产生 (默认)
	1: 更新中断挂起, 当相关寄存器被更新时该位由硬件置 1
	- 若 TIMx_CR1 寄存器的 UDIS=0, 当计数器上溢或下溢时
	- 若 TIMx_CR1 寄存器的 UDIS=0、URS=0, 当软件设置 TIMx_EGR 寄存器的 UG 位对计数器 CNT 重新初始化时
	- 若 TIMx_CR1 寄存器的 UDIS=0、URS=0, 当计数器 CNT 被触发事件重新初始化时

16.9.6 EGR 事件产生寄存器 (TIM15/16/17_EGR)

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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.							B2G	BG	TG	COMG	Res.		CC2G	CC1G	UG
							w	w	w	w			w	w	w

Bits 31:9	保留, 必须保持复位值
Bit 8	B2G: 产生刹车 2 事件 (仅用于 TIM15/16)
	该位由软件置 1, 用于产生一个刹车事件, 由硬件自动清 0
	0: 无动作 (默认)
	1: 产生刹车 2 事件。此时 MOE=0、B2IF=1, 若 BIE=1, 则产生相应的中断
Bit 7	BG: 产生刹车事件

	该位由软件置 1，用于产生一个刹车事件，由硬件自动清 0
	0: 无动作（默认）
	1: 产生刹车事件。此时 MOE=0、BIF=1，若 BIE=1，则产生相应的中断
Bit 6	TG : 产生触发事件
	该位由软件置 1，用于产生一个触发事件，由硬件自动清 0
	0: 无动作（默认）
	1: 产生触发事件。此时 TIF=1，若 TIE=1，则产生相应的中断
Bit 5	COMG : 产生捕获/比较控制更新事件
	该位由软件置 1，用于产生一个捕获/比较控制更新事件，由硬件自动清 0
	0: 无动作（默认）
	1: 当 CCPC=1 时，允许更新 CCiE、CCiNE、CCiP、CCiNP 和 OCiM 位。此时 COMIF=1，若 COMIE=1，则产生相应的中断
	<i>注：该位只对拥有互补输出的通道有效。</i>
Bits 4:3	保留，必须保持复位值
Bit 2	CC2G : 产生捕获/比较 2 事件，参考 CC1G 位（仅用于 TIM15）
Bit 1	CC1G : 产生捕获/比较 1 事件
	该位由软件置 1，用于产生一个捕获/比较事件，由硬件自动清 0
	0: 无动作（默认）
	1: 在通道 CC1 上产生一个捕获/比较事件
	若通道 CC1 配置为输出：
	设置 CC1IF=1，若 CC1IE=1，则产生相应的中断；若 CC1IF 已经为 1，则设置 CC1OF=1
	若通道 CC1 配置为输入：
	当前的计数器值被捕获至 TIMx_CCR1 寄存器，设置 CC1IF=1，若 CC1IE=1，则产生相应的中断；若 CC1IF 已经为 1，则设置 CC1OF=1
Bit 0	UG : 产生更新事件
	该位由软件置 1，用于产生一个更新事件，由硬件自动清 0
	0: 无动作（默认）
	1: 重新初始化计数器，并产生一个更新事件。注意预分频器的计数器也被清 0（但是预分频系数不变）。若中央对齐模式或向上计数（DIR=0）则计数器被清 0；若向下计数（DIR=1）则计数器取 TIMx_ARR 的值

16.9.7 CCMR1 捕获/比较模式寄存器 1 (TIM15/16/17_CCMR1)

地址偏移: 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 CCiS 位定义。该寄存器部分位的作用在输入和输出模式下不同。OCxx 描述了通道在输出模式下的功能，ICxx 描述了通道在输入模式下的功能。因此必须注意，同一个寄存器位在输出模式和输入模式下的功能是不同的。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OC2TE	OC2TUE	Res.						OC1TE	OC1TUE	Res.					
rw	rw							rw	rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		Res.	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	

IC2F[3:0]	IC2PSC[1:0]		IC1F[3:0]	IC1PSC[1:0]	
rw	rw	rw	rw	rw	rw

输出比较模式

Bit	31	OC2TE : 输出比较 2 修调使能, 参考 OC1TE (仅用于 TIM15)
Bit	30	OC2TUE : 输出比较 2 修调更新使能, 参考 OC1TUE (仅用于 TIM15)
Bits	29:24	保留, 必须保持复位值
Bit	23	OC1TE : 输出比较 1 修调使能, 与 CCTRi 寄存器配合使用
		0: 禁止 (默认)
		1: 使能
Bit	22	OC1TUE : 输出比较 1 修调更新使能, 与 CCTRi 寄存器配合使用
		0: 禁止 (默认)
		1: 使能
Bits	21:16	保留, 必须保持复位值
Bits	14:12	OC2M[2:0] : 输出比较 2 模式, 参考 OC1M (仅用于 TIM15)
Bit	11	OC2PE : 输出比较 2 预装载使能, 参考 OC1PE (仅用于 TIM15)
Bit	10	OC2FE : 输出比较 2 快速使能, 参考 OC1FE (仅用于 TIM15)
Bits	9:8	CC2S[1:0] : 捕获/比较 2 选择 (仅用于 TIM15)
		该位定义通道的方向 (输入/输出), 以及输入脚的选择
		00: CC2 通道被配置为输出 (默认)
		01: CC2 通道被配置为输入, IC2 映射在 TI2FP2 上
		10: CC2 通道被配置为输入, IC2 映射在 TI1FP2 上
		11: CC2 通道被配置为输入, IC2 映射在 TRC 上。此模式仅工作在触发输入选中 ITRx 或 TI1F_ED 时 (由 TIMx_SMCR 寄存器的 TS 位选择)
		注 : CC2S 仅在通道关闭时 (TIMx_CCER 寄存器的 CC2E=0、CC2NE=0 且已被更新) 才是可写的。
Bit	7	保留, 必须保持复位值
Bits	6:4	OC1M[2:0] : 输出比较 1 模式选择
		这 3 位定义了输出参考信号 OC1REF 的动作, 而 OC1REF 决定了 OC1 和 OC1N 的值。OC1REF 是高电平有效, 而 OC1 和 OC1N 的有效电平取决于 CC1P 和 CC1NP。
		000: 冻结。输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 间的比较对 OC1REF 不起作用。(默认)
		001: 匹配时设置通道 1 的输出为有效电平。当计数器 TIMx_CNT 的值与捕获/比较寄存器 1 (TIMx_CCR1) 相同时, 强制 OC1REF 为高。
		010: 匹配时设置通道 1 的输出为无效电平。当计数器 TIMx_CNT 的值与捕获/比较寄存器 1 (TIMx_CCR1) 相同时, 强制 OC1REF 为低。
		011: 翻转。当 TIMx_CCR1=TIMx_CNT 时, 翻转 OC1REF 的电平。
		100: 强制为无效电平。强制 OC1REF 为低。
		101: 强制为有效电平。强制 OC1REF 为高。
		110: PWM 模式 1 — 在向上计数时, 一旦 TIMx_CNT<TIMx_CCR1 时通道 1 为有效电平, 否则为无效电平; 在向下计数时, 一旦 TIMx_CNT>TIMx_CCR1 时通道 1 为无效电平, 否则为有效电平。
		111: PWM 模式 2 — 在向上计数时, 一旦 TIMx_CNT<TIMx_CCR1 时通道 1 为无效电平, 否则为有效电平; 在向下计数时, 一旦 TIMx_CNT>TIMx_CCR1 时通道 1 为有

	效电平，否则为无效电平。
	注1: 一旦 LOCK 级别设为3 (TIMx_BDTR 寄存器中的 LOCK 位) 并且 CC1S=00 (该通道配置成输出) 则该位不能被修改。
	注2: 在 PWM 模式1 或 PWM 模式2 中, 只有当比较结果改变了或在输出比较模式中从冻结模式切换到 PWM 模式时, OC1REF 电平才改变。(参考章节 16.5.5 PWM 模式)。
	注3: 在有互补输出的通道上, 这些位是预装载的。如果 TIMx_CR2 寄存器的 CCPC=1, OC1M 只有在 COM 事件发生时, 才会预装载新的值。
Bit 3	OC1PE: 输出比较 1 预装载使能
	0: 禁止 TIMx_CCR1 寄存器的预装载功能, 可随时写入 TIMx_CCR1 寄存器, 并且新写入的数值立即起作用 (默认)
	1: 使能 TIMx_CCR1 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIMx_CCR1 的预装载值在更新事件到来时被加载至当前寄存器中
	注1: 一旦 LOCK 级别设为3 (TIMx_BDTR 寄存器中的 LOCK 位) 并且 CC1S=00 (该通道配置成输出) 则该位不能被修改。
	注2: 为了操作正确, 在 PWM 模式下必须使能预装载功能。但在单脉冲模式下 (TIMx_CR1 寄存器的 OPM=1), 它不是必须的。
Bit 2	OC1FE: 输出比较 1 快速使能
	该位用于加快 CC 输出对触发输入的响应
	0: 根据计数器与 CCR1 的值, CC1 正常操作。当触发输入出现一个有效边沿时, 通过计数器比较输出 CC1。(默认)
	1: 触发输入的有效边沿的作用就如同发生了一次比较匹配。因此, OC1REF 被直接设置为有效电平, 而与比较结果无关。触发输入的有效边沿与 CC1 输出之间的延时被缩短。
	注: OC1FE 只在通道被配置成 PWM 模式 1/2 的单脉冲模式时起作用。
Bits 1:0	CC1S[1:0]: 捕获/比较 1 选择
	该位定义通道的方向 (输入/输出), 以及输入脚的选择
	00: CC1 通道被配置为输出 (默认)
	01: CC1 通道被配置为输入, IC1 映射在 TI1FP1 上
	10: CC1 通道被配置为输入, IC1 映射在 TI2FP1 上 (仅用于 TIM15)
	11: CC1 通道被配置为输入, IC1 映射在 TRC 上。此模式仅工作在触发输入选中 ITRx 或 TI1F_ED 时 (由 TIMx_SMCR 寄存器的 TS 位选择) (仅用于 TIM15)
	注: CC1S 仅在通道关闭时 (TIMx_CCER 寄存器的 CC1E=0、CC1NE=0 且已被更新) 才是可写的。

输入捕获模式

Bits 31:16	保留, 必须保持复位值
Bits 15:12	IC2F[3:0]: 输入捕获 2 滤波器, 参考 IC1F (仅用于 TIM15)
Bits 11:10	IC2PSC[1:0]: 输入捕获 2 预分频器, 参考 IC1PSC (仅用于 TIM15)
Bits 9:8	CC2S[1:0]: 捕获/比较 2 选择 (仅用于 TIM15)
	该位定义通道的方向 (输入/输出), 以及输入脚的选择
	00: CC2 通道被配置为输出 (默认)
	01: CC2 通道被配置为输入, IC2 映射在 TI2FP2 上
	10: CC2 通道被配置为输入, IC2 映射在 TI1FP2 上
	11: CC2 通道被配置为输入, IC2 映射在 TRC 上。此模式仅工作在触发输入选中 ITRx

	或 TI1F_ED 时（由 TIMx_SMCR 寄存器的 TS 位选择）
	注： CC2S 仅在通道关闭时（TIMx_CCER 寄存器的 CC2E=0、CC2NE=0 且已被更新）才是可写的。
Bits 7:4	IC1F[3:0]： 输入捕获 1 滤波器
	该位域定义了 TI1 输入的采样频率及数字滤波器的长度。数字滤波器由一个事件计数器组成，只有发生了 N 个连续事件后输出的跳变才被认为有效。这里的 DTS 时钟是 CK_INT 经过 CR1 寄存器中 CKD 配置分频的。
	0000：采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}$ ，无滤波器（默认）
	0001：采样频率 $f_{\text{SAMPLING}} = f_{\text{CK_INT}}$ ，N=2
	0010：采样频率 $f_{\text{SAMPLING}} = f_{\text{CK_INT}}$ ，N=4
	0011：采样频率 $f_{\text{SAMPLING}} = f_{\text{CK_INT}}$ ，N=8
	0100：采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/2$ ，N=6
	0101：采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/2$ ，N=8
	0110：采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$ ，N=6
	0111：采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/4$ ，N=8
	1000：采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/8$ ，N=6
	1001：采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/8$ ，N=8
	1010：采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$ ，N=5
	1011：采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$ ，N=6
	1100：采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/16$ ，N=8
	1101：采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$ ，N=5
	1110：采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$ ，N=6
	1111：采样频率 $f_{\text{SAMPLING}} = f_{\text{DTS}}/32$ ，N=8
	注 1： 请注意采样频率中 $f_{\text{CK_INT}}$ 和 f_{DTS} 的差别。
	注 2： 即使对于带互补输出的通道，该位域也是非预装载的，并且不会考虑 CCPC（TIMx_CR2 寄存器）的值。
Bits 3:2	IC1PSC[1:0]： 输入捕获 1 预分频器
	IC1 的预分频系数，一旦 CC1E=0（TIMx_CCER 寄存器中），则预分频器复位
	00：无预分频器，TI1 上每 1 个有效边沿作为一次输入（默认）
	01：TI1 上每 2 个有效边沿作为一次输入
	10：TI1 上每 4 个有效边沿作为一次输入
	11：TI1 上每 8 个有效边沿作为一次输入
Bits 1:0	CC1S[1:0]： 捕获/比较 1 选择
	该位定义通道的方向（输入/输出），以及输入脚的选择
	00：CC1 通道被配置为输出（默认）
	01：CC1 通道被配置为输入，IC1 映射在 TI1FP1 上
	10：CC1 通道被配置为输入，IC1 映射在 TI2FP1 上（仅用于 TIM15）
	11：CC1 通道被配置为输入，IC1 映射在 TRC 上。此模式仅工作在触发输入选中 ITRx 或 TI1F_ED 时（由 TIMx_SMCR 寄存器的 TS 位选择）（仅用于 TIM15）
	注： CC1S 仅在通道关闭时（TIMx_CCER 寄存器的 CC1E=0、CC1NE=0 且已被更新）才是可写的。

16.9.8 CCER 捕获/比较使能寄存器（TIM15/16/17_CCER）

地址偏移：0x20

复位值：0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
								rw	rw	rw	rw	rw	rw	rw	rw

Bits	31:8	保留，必须保持复位值
Bit	7	CC2NP : 捕获/比较 2 互补输出极性，参考 CC1NP（仅用于 TIM15）
Bit	6	CC2NE : 捕获/比较 2 互补输出使能，参考 CC1NE（仅用于 TIM15）
Bit	5	CC2P : 捕获/比较 2 极性，参考 CC1P（仅用于 TIM15）
Bit	4	CC2E : 捕获/比较 2 使能，参考 CC1E（仅用于 TIM15）
Bit	3	CC1NP : 捕获/比较 1 互补输出极性
		CC1 通道作为输出:
		0: OC1N 高电平有效（默认）
		1: OC1N 低电平有效
		CC1 通道作为输入:
		该位与 CC1P 共同使用，定义 TIiFP1 的极性，具体请参考 CC1P
		注 1 : 一旦 LOCK 级别设为 3 或 2（TIMx_BDTR 寄存器中的 LOCK 位）并且 CC1S=00（该通道配置成输出）则该位不能被修改。
		注 2 : 对于有互补输出的通道，该位是预装载的。如果 CCPC=1（TIMx_CR2 寄存器），只有在 COM 事件发生时，CC1NP 位才从预装载中读取新的值。
Bit	2	CC1NE : 捕获/比较 1 互补输出使能
		0: 关闭 — OC1N 输出禁止，因此 OC1N 的输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。（默认）
		1: 开启 — OC1N 信号输出到对应的输出引脚，其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。
		注 : 对于有互补输出的通道，该位是预装载的。如果 CCPC=1（TIMx_CR2 寄存器），只有在 COM 事件发生时，CC1NE 位才从预装载中读取新的值。
Bit	1	CC1P : 捕获/比较 1 极性
		CC1 通道作为输出:
		0: OC1 高电平有效（默认）
		1: OC1 低电平有效
		CC1 通道作为输入:
		CC1NP 和 CC1P 位共同选择 TI1FP1 和 TI2FP1 作为触发或捕获的极性（TIM16/17 只有 TI1FP1）
		00: 信号不反相，TIiFP1 的上升沿有效（默认）
		01: 信号反相，TIiFP1 的下降沿有效
		10: 保留
		11: 信号不反相，TIiFP1 的上升沿和下降沿均有效
		注 1 : 一旦 LOCK 级别设为 3 或 2（TIMx_BDTR 寄存器中的 LOCK 位）则该位不能被修改。
		注 2 : 对于有互补输出的通道，该位是预装载的。如果 CCPC=1（TIMx_CR2 寄存器），只有在 COM

	事件发生时, CC1P 位才从预装载中读取新的值。
Bit 0	CC1E: 捕获/比较 1 使能
	CC1 通道作为输出:
	0: 关闭 — OC1 输出禁止, 因此 OC1 的输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。(默认)
	1: 开启 — OC1 信号输出到对应的输出引脚, 其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。
	CC1 通道作为输入:
	该位决定了计数器的值是否能捕获入 TIMx_CCR1 寄存器
	0: 捕获禁止 (默认)
	1: 捕获使能
	注: 对于有互补输出的通道, 该位是预装载的。如果 CCPC=1 (TIMx_CR2 寄存器), 只有在 COM 事件发生时, CC1E 位才从预装载中读取新的值。

表 16-5 带刹车功能的互补输出 OC_i 和 OC_{iN} 的控制

控制位					输出状态	
MOE	OSSI	OSSR	CCiE	CCiNE	OC _i 输出状态	OC _{iN} 输出状态
1	X	0	0	0	输出禁止 (与定时器断开) OC _i =0, OC _{iN} =0	
		0	0	1	输出禁止 (与定时器断开) OC _i =0	OC _i REF + 极性, OC _{iN} =OC _i REF xor CC _i NP
		0	1	0	OC _i REF + 极性, OC _i =OC _i REF xor CC _i P	输出禁止 (与定时器断开) OC _{iN} =0
		0	1	1	OC _i REF + 极性 + 死区	OC _i REF 反相 + 极性 + 死区
		1	0	0	输出禁止 (与定时器断开) OC _i =CC _i P, OC _{iN} =CC _i NP	
		1	0	1	关闭状态 (输出使能且为无效电平) OC _i =CC _i P	OC _i REF + 极性, OC _{iN} =OC _i REF xor CC _i NP
		1	1	0	OC _i REF + 极性, OC _i =OC _i REF xor CC _i P	关闭状态 (输出使能且为无效电平) OC _{iN} =CC _i NP
		1	1	1	OC _i REF + 极性 + 死区	OC _i REF 反相 + 极性 + 死区
0	X	0	0	0	输出禁止 (与定时器断开) OC _i =CC _i P, OC _{iN} =CC _i NP	
		0	0	1	首先处于输出禁止状态 (与定时器断开), OC _i =CC _i P,	
		0	1	0	OC _{iN} =CC _i NP, 死区时钟存在且经过死区时间后, 如果 CC _i P=OIS _i	
		0	1	1	且 CC _i NP=OIS _{iN} , 则继续输出无效电平; 否则输出空闲电平, 即	
		1	0	0	输出禁止 (与定时器断开) OC _i =CC _i P, OC _{iN} =CC _i NP	
		1	1	0	首先处于关闭状态 (输出使能且为无效电平), OC _i =CC _i P,	
		1	0	1	OC _{iN} =CC _i NP, 死区时钟存在且经过死区时间后, 如果 CC _i P≠OIS _i	
		1	1	1	且 CC _i PN≠OIS _{iN} , 则继续输出无效电平; 否则输出空闲电平, 即	
					OC _i =OIS _i , OC _{iN} =OIS _{iN}	

注：管脚连接到互补的 OCi 和 OCiN 通道的外部 I/O 管脚的状态，取决于 OCi 和 OCiN 通道状态和 GPIO 寄存器。

16.9.9 CNT 计数寄存器 (TIM15/16/17_CNT)

地址偏移：0x24

复位值：0x0000_0000

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

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

Bits	31:16	保留，必须保持复位值
Bits	15:0	CNT[15:0] : 计数器的值

16.9.10 PSC 预分频寄存器 (TIM15/16/17_PSC)

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
Res.															

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

Bits	31:16	保留，必须保持复位值
Bits	15:0	PSC[15:0] : 预分频器的值
		计数器的频率可以由下式计算： $f_{CK_CNT} = f_{CK_PSC} / (PSC[15:0] + 1)$ 。 预分频器的值由预装载寄存器写入，新的预分频器的值在下一次更新事件到来时被采用。可以通过 AS 位 (TIMx_CR1 寄存器) 来选择读取 PSC 寄存器的值来自影子寄存器或预装载寄存器。

16.9.11 ARR 自动重装载寄存器 (TIM15/16/17_ARR)

地址偏移：0x2C

复位值：0x0000_0000

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



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

Bits 31:16	保留，必须保持复位值
Bits 15:0	ARR[15:0]: 自动重载寄存器的值
	ARR 是要加载到自动重载寄存器中的值。具体请参考 16.2.1 章节。可以通过 AS 位（TIMx_CR1 寄存器）来选择读取 ARR 寄存器的值来自影子寄存器或预装载寄存器。

16.9.12 RCR 重复计数寄存器 (TIM15_RCR)

地址偏移: 0x30

复位值: 0x0000_0000

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

Bits 31:8	保留，必须保持复位值
Bits 7:0	REP[7:0]: 重复计数器的值（仅用于 TIM15）
	允许用户设置更新速率；如果允许产生更新中断，则会同时影响产生更新中断的速率。
	每次向下计数器 REP_CNT 达到 0，会产生一个更新事件 UEV，并且计数器 REP_CNT 重新从 REP 值开始计数。由于 REP_CNT 只有在更新事件发生时才重载 REP 值，因此对 TIMx_RCR 寄存器写入的新值只在下次更新事件发生时才起作用。
	这意味着在 PWM 模式中，REP+1 对应着：
	- 在边沿对齐模式下，PWM 周期的数目
	- 在中央对齐模式下，PWM 半周期的数目

16.9.13 CCR1 捕获/比较寄存器 1 (TIM15/16/17_CCR1)

地址偏移: 0x34

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw															

Bits 31:16	保留，必须保持复位值
Bits 15:0	CCR1[15:0] : 捕获/比较 1 的值
	CC1 通道作为输出:
	CCR1 包含了要与计数器进行输出比较的值，它与计数器 TIMx_CNT 的值相比较，并在 OC1 端口上产生输出信号
	如果在 OC1PE 位 (TIMx_CCMR1 寄存器) 未选择预装载功能，写入的数值会立即传输至影子寄存器中；否则只有当更新事件发生时，此预装载值才传输至影子寄存器中
	当刹车触发且 AOE=1 (TIMx_BDTR 寄存器) 时，CCR1 的值可以通过相应的 TIMx_CCTR1 修调寄存器来更新：如果 TIMx_CCMR1 寄存器中的 OC1TE=1，当更新事件 UEV 发生时，CCR1 的影子寄存器被 TIMx_CCTR1 的值更新；如果 TIMx_CCMR1 寄存器中的 OC1TE=1 且 OC1TUE=1，当更新事件 UEV 发生时，CCR1 的影子寄存器和预装载寄存器都被 TIMx_CCTR1 的值更新
	可以通过 AS 位 (TIMx_CR1 寄存器) 来选择读取 CCR1 寄存器的值来自影子寄存器或预装载寄存器
	CC1 通道作为输入:
	CCR1 包含了上一次输入捕获 1 事件 (IC1) 发生时的计数器值，此时该寄存器为只读

16.9.14 CCR2 捕获/比较寄存器 2 (TIM15_CCR2)

地址偏移: 0x38

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw															

Bits 31:16	保留，必须保持复位值
Bits 15:0	CCR2[15:0] : 捕获/比较 2 的值 (仅用于 TIM15)
	CC2 通道作为输出:
	CCR2 包含了要与计数器进行输出比较的值，它与计数器 TIMx_CNT 的值相比较，并在 OC2 端口上产生输出信号
	如果在 OC2PE 位 (TIMx_CCMR1 寄存器) 未选择预装载功能，写入的数值会立即传输至影子寄存器中；否则只有当更新事件发生时，此预装载值才传输至影子寄存器中
	当刹车触发且 AOE=1 (TIMx_BDTR 寄存器) 时，CCR2 的值可以通过相应的 TIMx_CCTR2 修调寄存器来更新：如果 TIMx_CCMR1 寄存器中的 OC2TE=1，当更新事件 UEV 发生时，CCR2 的影子寄存器被 TIMx_CCTR2 的值更新；如果 TIMx_CCMR1 寄存器中的 OC2TE=1 且 OC2TUE=1，当更新事件 UEV 发生时，CCR2 的影子寄存器

	和预装载寄存器都被 TIMx_CCTR2 的值更新
	可以通过 AS 位 (TIMx_CR1 寄存器) 来选择读取 CCR2 寄存器的值来自影子寄存器或预装载寄存器
	CC2 通道作为输入:
	CCR2 包含了上一次输入捕获 2 事件 (IC2) 发生时的计数器值, 此时该寄存器为只读

16.9.15 BDTR 刹车和死区控制寄存器 (TIM15/16/17_BDTR)

地址偏移: 0x44

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.					AOE2	BK2P	BK2E	Res.							
					rw	rw	rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
rw	rw	rw	rw	rw	rw	rw		rw							

Bits 31:27	保留, 必须保持复位值
Bit 26	AOE2: 自动输出使能 2 (仅用于 TIM15/16)
	0: MOE 只能被软件置 1 (默认)
	1: MOE 能被软件置 1 或在下一个更新事件被自动置 1 (如果刹车 2 输入无效)
	<i>注:</i> 一旦 LOCK 级别设为 1 (TIMx_BDTR 寄存器中的 LOCK 位) 则该位不能被修改。
Bit 25	BK2P: 刹车 2 输入极性 (仅用于 TIM15/16)
	0: 刹车 2 输入低电平有效 (默认)
	1: 刹车 2 输入高电平有效
	<i>注:</i> 一旦 LOCK 级别设为 1 (TIMx_BDTR 寄存器中的 LOCK 位) 则该位不能被修改。
Bit 24	BK2E: 刹车 2 功能使能 (仅用于 TIM15/16)
	0: 刹车 2 功能禁止 (默认)
	1: 刹车 2 功能使能
	<i>注:</i> 一旦 LOCK 级别设为 1 (TIMx_BDTR 寄存器中的 LOCK 位) 则该位不能被修改。
Bits 23:16	保留, 必须保持复位值
Bit 15	MOE: 主输出使能
	一旦刹车输入有效, 该位被硬件异步清 0。根据 AOE/AOE2 位的配置, 该位可以由软件置 1 或被自动置 1。它仅对配置为输出的通道有效。
	0: 禁止 OC _i 和 OC _{iN} 输出或强制为空闲状态 (默认)
	1: 输出使能, 如果设置了相应的使能位 (TIMx_CCER 寄存器的 CC _{iE} 位), 则使能 OC _i 和 OC _{iN} 输出
Bit 14	AOE: 自动输出使能
	0: MOE 只能被软件置 1 (默认)
	1: MOE 能被软件置 1 或在下一个更新事件被自动置 1 (如果刹车输入无效)
	<i>注:</i> 一旦 LOCK 级别设为 1 (TIMx_BDTR 寄存器中的 LOCK 位) 则该位不能被修改。

Bit 13	BKP: 刹车输入极性
	0: 刹车输入低电平有效 (默认)
	1: 刹车输入高电平有效
	<i>注:</i> 一旦 LOCK 级别设为 1 (TIMx_BDTR 寄存器中的 LOCK 位) 则该位不能被修改。
Bit 12	BKE: 刹车功能使能
	0: 刹车功能禁止 (默认)
	1: 刹车功能使能
	<i>注:</i> 一旦 LOCK 级别设为 1 (TIMx_BDTR 寄存器中的 LOCK 位) 则该位不能被修改。
Bit 11	OSSR: 运行模式下的关闭状态选择
	该位用于当 MOE=1 且通道为互补输出时, 具体描述请参考表格 13-5
	0: 当 OCi/OCiN 未使能时, OCi/OCiN 输出禁止 (默认)
	1: 当 OCi/OCiN 未使能时, 如果 CCiE=1 或 CCiNE=1, 则 OCi/OCiN 输出无效电平
	<i>注:</i> 一旦 LOCK 级别设为 2 (TIMx_BDTR 寄存器中的 LOCK 位) 则该位不能被修改。
Bit 10	OSSI: 空闲模式下的关闭状态选择
	该位用于当 MOE=0 且通道作为输出时, 具体描述请参考表格 13-5
	0: 当 OCi/OCiN 未使能时, OCi/OCiN 输出禁止 (默认)
	1: 当 OCi/OCiN 未使能时, 如果 CCiE=1 或 CCiNE=1, 则 OCi/OCiN 输出空闲电平
	<i>注:</i> 一旦 LOCK 级别设为 2 (TIMx_BDTR 寄存器中的 LOCK 位) 则该位不能被修改。
Bits 9:8	LOCK[1:0]: 锁定设置, 为防止软件错误而提供写保护
	00: 锁定关闭, 寄存器无写保护 (默认)
	01: 锁定级别 1, 不能写入 TIMx_BDTR 寄存器的 BKE、BKP、AOE、DTG 位, TIMx_CR2 寄存器的 OISi/OISiN 位和 TIMx_DLAMT 寄存器的所有位
	10: 锁定级别 2, 不能写入锁定级别 1, 也不能写入 CC 极性位 (当通道配置为输出时, TIMx_CCER 寄存器的 CCiP/CCiNP 位) 以及 OSSR/OSSI 位
	11: 锁定级别 3, 不能写入锁定级别 2, 也不能写入 CC 控制位 (当通道配置为输出时, TIMx_CCMRi 寄存器的 OCiM/OCiPE 位)
	<i>注:</i> 在系统复位后, 只能写一次 LOCK 位, 一旦写入 TIMx_BDTR 寄存器, 则其内容保持不变直至复位。
Bits 7:0	DTG[7:0]: 死区发生器设置
	该位定义了插入互补输出之间的死区持续时间。假设 DT 表示其持续时间, t_{DTs} 为死区时钟。
	DTG[7:5]=0xx => $DT=DTG[7:0] \times t_{dtg}$, 其中 $t_{dtg}=t_{DTs}$
	DTG[7:5]=10x => $DT=(64+DTG[5:0]) \times t_{dtg}$, 其中 $t_{dtg}=2 \times t_{DTs}$
	DTG[7:5]=110 => $DT=(32+DTG[4:0]) \times t_{dtg}$, 其中 $t_{dtg}=8 \times t_{DTs}$
	DTG[7:5]=111 => $DT=(32+DTG[4:0]) \times t_{dtg}$, 其中 $t_{dtg}=16 \times t_{DTs}$
	举例, 如果 $t_{DTs}=125 \text{ ns}$ (8 MHz), 可能的死区时间为:
	DTG[7:0] = 0 到 7FH, 0 到 15875 ns, 步长时间为 125 ns
	DTG[7:0] = 80H 到 BFH, 16 μs 到 31750 ns, 步长时间为 250 ns
	DTG[7:0] = C0H 到 DFH, 32 μs 到 63 μs , 步长时间为 1 μs
	DTG[7:0] = E0H 到 FFH, 64 μs 到 126 μs , 步长时间为 2 μs
	<i>注:</i> 一旦 LOCK 级别设为 1、2 或 3 (TIMx_BDTR 寄存器中的 LOCK 位) 则该位不能被修改。



16.9.16 DCR DMA 控制寄存器 (TIM15/16/17_DCR)

地址偏移: 0x48

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			DBL[4:0]					DSEL[2:0]			DBA[4:0]				
			rw					rw			rw				

Bits 31:13	保留, 必须保持复位值
Bits 12:8	DBL[4:0]: DMA burst 传输长度
	定义了 DMA 在 burst 模式下的传输长度 (当对 TIMx_DMAR 寄存器的地址进行读或写时, TIMx 进行一次 burst 传输), 即定义被传送的字节数目
	00000: 1 字节 (默认)
	00001: 2 字节
	00010: 3 字节
	...
	10001: 18 字节
Bits 7:5	DSEL[2:0]: DMA burst 传输请求选择
	定义了哪个 DMA 请求使用 burst 传输, 其它的未选中的则使用 single 传输
	000: 所有 DMA 请求都使用 single 传输
	001: DMA 请求 0 使用 burst 传输
	010: DMA 请求 1 使用 burst 传输
	011: DMA 请求 2 使用 burst 传输
	100: DMA 请求 3 使用 burst 传输
	101 至 111: 保留
Bits 4:0	DBA[4:0]: DMA 基地址
	该位定义了 DMA 在 burst 模式下的基地址 (当对 TIMx_DMAR 寄存器的地址进行读或写时), DBA 定义为从 TIMx_CR1 寄存器所在地址开始的偏移量
	00000: TIMx_CR1
	00001: TIMx_CR2
	00010: TIMx_SMCR
	...
	举例: 当 DBL=7 且 DBA=00000 时, 表示从 TIMx_CR1 开始向下读取或发送 7 个寄存器的数据。

表 16-6 TIMx DMA 请求表

	TIM1	TIM3	TIM6	TIM14	TIM15	TIM16	TIM17
DMA 请求 0	TIM1_CH1	TIM3_CH1 TIM3_TRIG	TIM6_UP	N/A	TIM15_CH1 TIM15_CH2	TIM16_CH1 TIM16_UP	TIM17_CH1 TIM17_UP

					TIM15_UP TIM15_TRIG TIM15_COM	TIM16_TRIG TIM16_COM	TIM17_TRIG TIM17_COM
DMA 请求 1	TIM1_CH2	TIM3_CH2 TIM3_CH3	N/A	N/A	N/A	N/A	N/A
DMA 请求 2	TIM1_CH3 TIM1_UP	TIM3_CH4 TIM3_UP	N/A	N/A	N/A	N/A	N/A
DMA 请求 3	TIM1_CH4 TIM1_TRIG TIM1_COM	N/A	N/A	N/A	N/A	N/A	N/A

注：TIM6 仅支持 DMA single 传输。

16.9.17 DMAR DMA 传输寄存器 (TIM15/16/17_DMAR)

地址偏移：0x4C

复位值：0x0000_0000

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

Bits 31:0	DMAB[31:0]: DMA 传输寄存器
	对 TIMx_DMAR 寄存器的读或写会导致对以下地址的寄存器的读写操作：
	TIMx_CR1 地址 + (DBA + DMA 指针) x4，其中“TIMx_CR1 地址”是 TIMx 寄存器的起始地址；DBA 是 TIMx_DCR 寄存器中定义的基地址；DMA 指针是由 DMA 自动控制的偏移量，它的范围是从 0 至 TIMx_DCR 寄存器中定义的 DBL。

16.9.18 CCTR1 比较修调寄存器 1 (TIM15/16/17_CCTR1)

地址偏移：0x64

复位值：0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCTR1[15:0]															
rw															

Bits 31:16	保留，必须保持复位值
Bits 15:0	CCTR1[15:0]: 比较 1 修调值

通道 CC1 作为输出时，如果刹车触发且对应的 AOE 位使能，TIMx_CCMR1 寄存器中的 OC1TE=1，当更新事件 UEV 发生时，TIMx_CCR1 的影子寄存器被 TIMx_CCTR1 的值更新；如果 TIMx_CCMR1 寄存器中的 OC1TE=1 且 OC1TUE=1，当更新事件 UEV 发生时，TIMx_CCR1 的影子寄存器和预装载寄存器都被 TIMx_CCTR1 的值更新。
--

16.9.19 CCTR2 比较修调寄存器 2 (TIM15_CCTR2)

地址偏移: 0x68

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCTR2[15:0]															
rw															

Bits 31:16	保留，必须保持复位值
Bits 15:0	CCTR2[15:0]: 比较 2 修调值 (仅用于 TIM15)
	通道 CC2 作为输出时，如果刹车触发且对应的 AOE 位使能，TIMx_CCMR1 寄存器中的 OC2TE=1，当更新事件 UEV 发生时，TIMx_CCR2 的影子寄存器被 TIMx_CCTR2 的值更新；如果 TIMx_CCMR1 寄存器中的 OC2TE=1 且 OC2TUE=1，当更新事件 UEV 发生时，TIMx_CCR2 的影子寄存器和预装载寄存器都被 TIMx_CCTR2 的值更新。

16.9.20 SysCtrl_EDU_CFG3 寄存器

具体请参考 6.3.4 互联配置寄存器 3 (SysCtrl_EDU_CFG3) 的内容。

16.9.21 SysCtrl_EDU_CFG5 寄存器

具体请参考 6.3.6 互联配置寄存器 5 (SysCtrl_EDU_CFG5) 的内容。

16.9.22 SysCtrl_EDU_CFG6 寄存器

具体请参考 6.3.7 互联配置寄存器 6 (SysCtrl_EDU_CFG6) 的内容。

16.9.23 SysCtrl_EDU_CFG7 寄存器

具体请参考 6.3.8 互联配置寄存器 7 (SysCtrl_EDU_CFG7) 的内容。

17.基础定时器 TIM6

17.1 TIM6 主要特性

基础定时器 TIM6 是一个 16 位的定时器/计数器，由一个可编程的预分频器驱动，主要用于基本定时。在计数器上溢时能够产生中断和 DMA 请求。

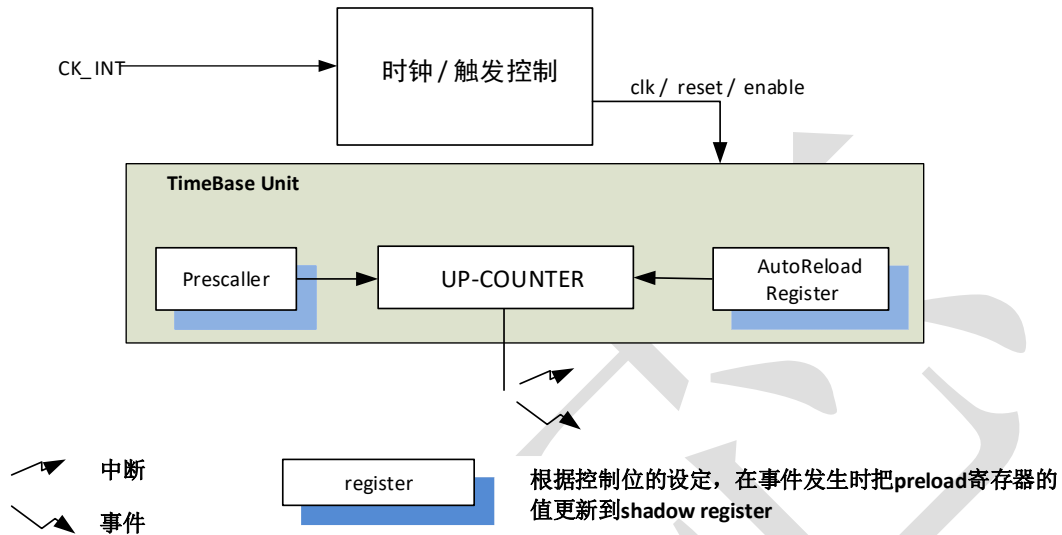


图 17-1 基础定时器 TIM6 框图

17.2 时基单元

时基单元包含：

- 16 位向上计数器 (TIM6_CNT)
- 16 位预分频器 (TIM6_PSC)
- 16 位自动重载寄存器 (TIM6_ARR)

16 位计数器、预分频器和自动重载寄存器都可以通过软件进行读写操作。计数器由预分频器的输出 CK_CNT 驱动，而 CK_CNT 仅在 TIM6_CR1 寄存器的计数器使能位 (CEN) 被置位时才有效。

注：在使能了 CEN 位的一个时钟周期后，计数器才开始计数。

写计数器的操作没有缓存，可以在任何时候写 TIM6_CNT 寄存器，因此建议不要在计数器运行时写入新的数值，以免写入了错误的数值。

17.2.1 自动装载寄存器

自动装载寄存器是预先装载的。写或读自动重载寄存器将访问预装载寄存器。根据在 TIM6_CR1 寄存器中的自动装载预装载使能位 (ARPE) 的设置，预装载寄存器的内容被立即或在每次的更新事件 UEV 时传送到影子寄存器。当计数器达到溢出条件并当 TIM6_CR1 寄存器中的 UDIS=0 时，产生更新事件。更新事件也可以由软件产生，随后会详细描述每一种配置下更新事件的产生。注意这里影子寄存器 (Shadow register) 即为有效的工作寄存器 (Active register)。

写自动重载寄存器的两种模式：

- 自动预装载已使能 (TIM6_CR1 寄存器的 ARPE 位置位)。在此模式下，写入自动重载寄存器的数据将被保存在预装载寄存器中，并在下一个更新事件 (UEV) 时传送到影子寄存器。
- 自动预装载已禁止 (TIM6_CR1 寄存器的 ARPE 位清除)。在此模式下，写入自动重载寄存器的数据

将立即写入影子寄存器。

更新事件的产生条件：

- 计数器向上溢出
- 软件置位了 TIM6_EGR 寄存器的 UG 位

在预装载使能时 (ARPE=1)，如果发生了更新事件，预装载寄存器中的数值 (TIM6_ARR) 将写入影子寄存器中。置位 TIM6_CR1 寄存器的 UDIS 位将禁止更新事件 (UEV)。

17.2.2 预分频器

TIM6 的预分频器基于一个由 16 位寄存器 (TIM6_PSC) 控制的 16 位计数器。由于这个控制寄存器带有缓冲器，因此它能够在运行时被改变。预分频器可以将计数器的时钟频率按 1 到 65536 之间的任意值分频。计数器的频率可以由下式计算：

$$f_{CK_CNT} = f_{CK_PSC} / (PSC[15:0] + 1)$$

预分频器的值由预装载寄存器写入，新的预分频器的值在下次更新事件到来时被采用。对 TIM6_PSC 寄存器的读操作通过预装载寄存器完成，因此不需要特别的关注。

下图给出了在预分频器工作时，更改其参数的情况下计数器操作的例子，其它分频系数类推。

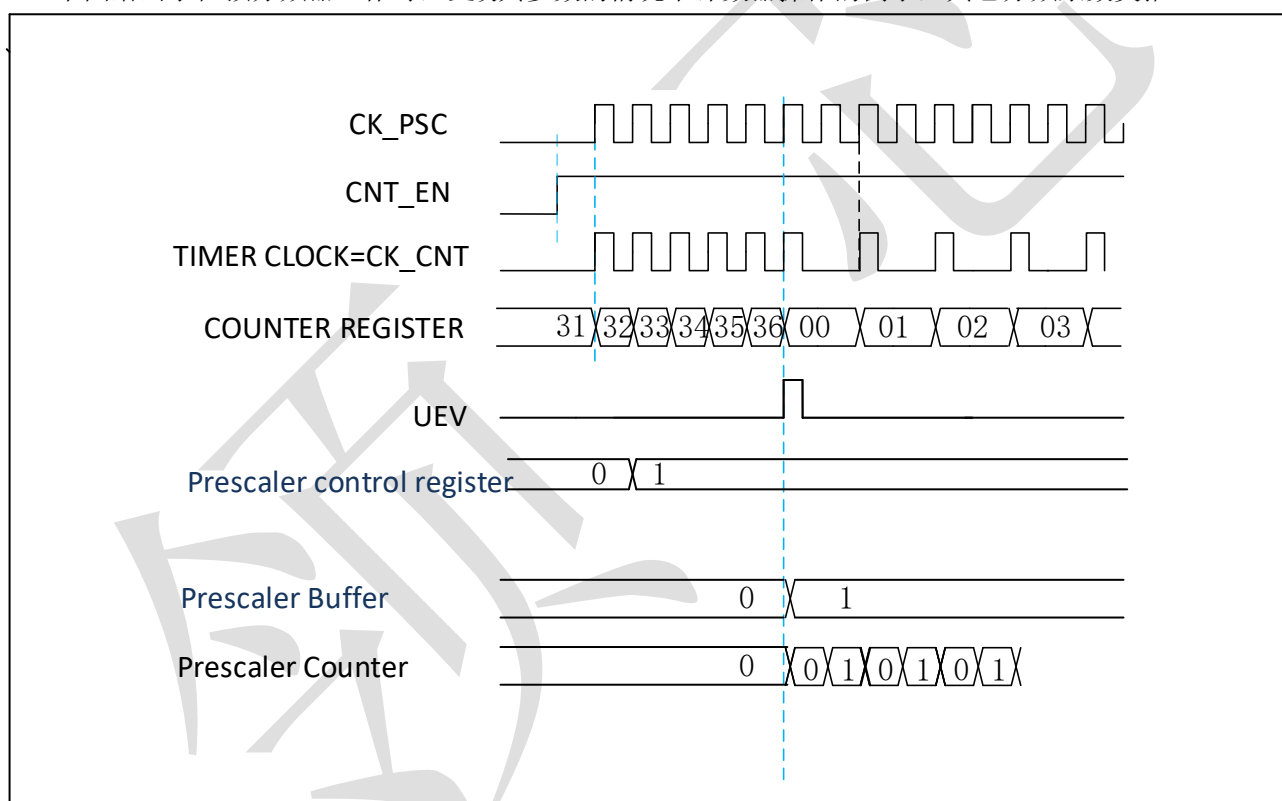


图 17-2 预分频系数从 1 变 2 的计数器时序

17.3 计数器模式

TIM6 只支持向上计数模式。在向上计数模式中，计数器从 0 计数到用户定义的比较值 (TIMx_ARR 寄存器的值)，然后重新从 0 开始计数并产生一个计数器溢出事件，同时，如果 TIMx_CR1 寄存器的 UDIS 位是 0，将会产生一个更新事件 (UEV)。

置位 TIMx_EGR 寄存器的 UG 位(通过软件方式或者使用从模式控制器)也同样可以产生一个更新事件。

使用软件置位 TIMx_CR1 寄存器的 UDIS 位，可以禁止更新事件，这样可以避免在更新预装载寄存器时

更新影子寄存器。在 UDIS 位被清除之前，将不产生更新事件。但是在应该产生更新事件时，计数器仍会被清 0，同时预分频器的计数也被清 0（但预分频器的数值不变）。此外，如果设置了 TIMx_CR1 寄存器中的 URS 位（选择更新请求源），设置 UG 位将产生一个更新事件 UEV，但硬件不设置 UIF 标志（即不产生中断请求）。这是为了避免在捕获模式下清除计数器时，同时产生更新和捕获中断。TIM6 支持单脉冲模式（OPM），设置 TIMx_CR1 寄存器的 OPM 位将选择单脉冲模式，此时计数器自动地在下一个更新事件 UEV 时停止。

当发生一个更新事件时，所有的寄存器都被更新，硬件同时依据 URS 位设置更新标志位（TIMx_SR 寄存器的 UIF 位）：自动装载影子寄存器被重新置入预装载寄存器的值（TIMx_ARR）。预分频器的缓存器被置入预装载寄存器的值（TIMx_PSC）。要注意到如果是因为计数器溢出而产生更新，自动重载寄存器将在计数器重载入之前被更新，因此下一个计数周期才是预期的值（计数器被装载为新的值）。非自动重载模式下可以立即更新。

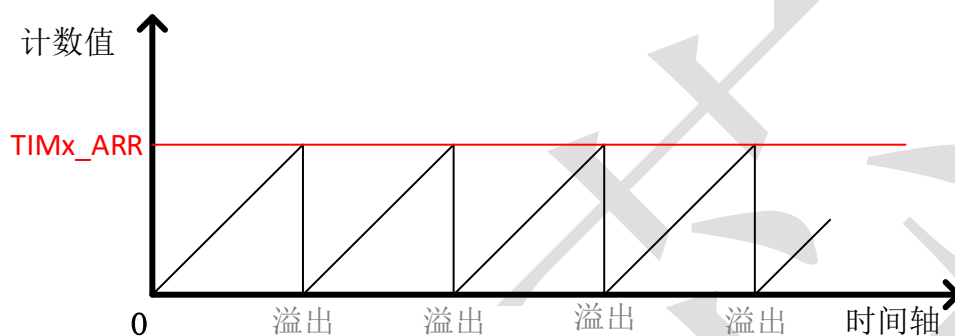


图 17-3 向上计数溢出

下图给出例子，当 TIMx_ARR=0x36 时计数器在二分频时钟频率下的动作。

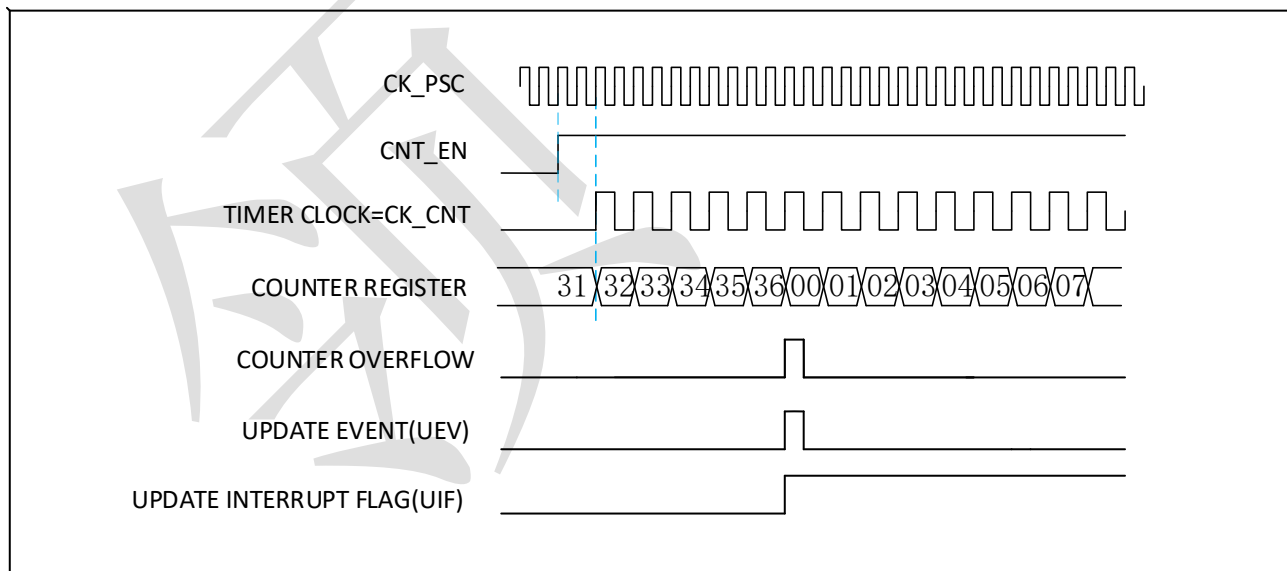


图 17-4 计数器在二分频时钟频率下的时序图

17.4 时钟选择

计数器时钟由内部时钟（CK_INT）提供，CEN、DIR 和 UG 位是实际上的控制位，并且只能被软件修改（UG 位仍被自动清除）。一旦 CEN 位被写成 1，预分频器的时钟就由内部时钟提供。

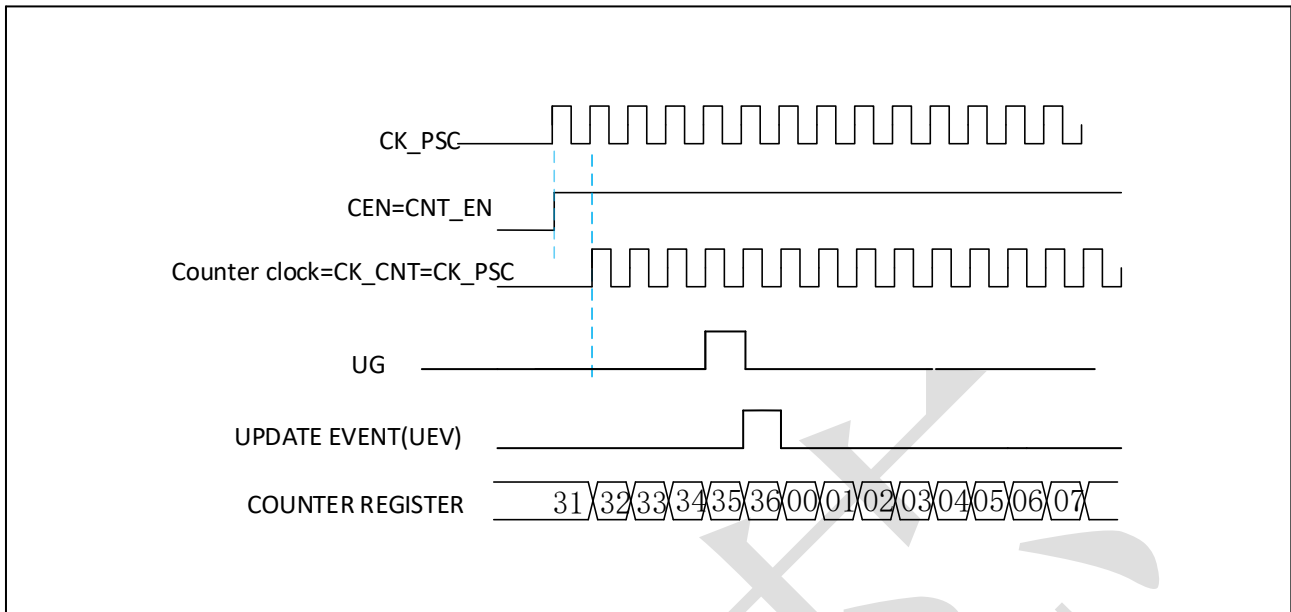


图 17-5 内部时钟模式

17.5 TIM6 中断

TIM6 有 1 个中断请求源，映射到 1 个中断矢量上：

- 更新事件中断

为了使用中断特性，设置 TIMx_DIER 寄存器中相应的中断使能位：UIE 位。

通过设置 TIMx_EGR 寄存器中的相应位，也可以用软件产生上述中断源。

17.6 TIM6 寄存器描述

表 17-1 TIM6 相关寄存器表

名称	说明	读写权限	复位值	字节地址
CR1	控制寄存器 1	R/W	0x0000_0000	0x4000_9000
DIER	DMA 和中断控制寄存器	R/W	0x0000_0000	0x4000_900C
SR	状态寄存器	R/W	0x0000_0000	0x4000_9010
EGR	事件产生寄存器	R/W	0x0000_0000	0x4000_9014
CNT	计数寄存器	R/W	0x0000_0000	0x4000_9024
PSC	预分频寄存器	R/W	0x0000_0000	0x4000_9028
ARR	自动重装载寄存器	R/W	0x0000_0000	0x4000_902C

注：x 表示不确定；-表示未实现；u 表示不受复位影响；c 表示该位复位值因复位源而异；R 只读；W 只写；R-0 表示只读且读 0；R/W 表示可读写（以后章节同上述）。

17.6.1 CR1 控制寄存器 1（TIM6_CR1）

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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								ARPE	Res.			OPM	URS	UDIS	CEN
								rw				rw	rw	rw	rw

Bits 31:9	保留，必须保持复位值
Bit 7	ARPE: 自动预装载允许位
	0: TIM6_ARR 寄存器可以被直接写入（默认）
	1: TIM6_ARR 寄存器通过预装载寄存器更新
Bits 6:4	保留，必须保持复位值
Bit 3	OPM: 单脉冲模式
	0: 在发生更新事件时，计数器不停止（默认）
	1: 在发生下一次更新事件时，计数器停止（清除 CEN 位）
Bit 2	URS: 更新请求源
	0: 如果 UDIS 允许产生更新事件，则下述任一事件产生一个更新中断（默认）：
	– 计数器上溢
	– 软件设置 UG 位
	1: 如果 UDIS 允许产生更新事件，则只有当计数器上溢时才产生更新中断
Bit 1	UDIS: 禁止更新
	0: 一旦下列事件发生，产生更新事件（默认）：
	– 计数器溢出
	– 软件设置 UG 位
	1: 不产生更新事件，影子寄存器（ARR、PSC、CCR）保持它们的值。如果 UG 位被配置，则计数器和预分频器被重新初始化
Bit 0	CEN: 计数器使能位
	0: 计数器禁止（默认）
	1: 计数器使能

17.6.2 DIER DMA 和中断控制寄存器（TIM6_DIER）

地址偏移：0x0C

复位值：0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.							UDE	Res.							UIE
							rw								rw

Bits	31:9	保留，必须保持复位值
Bit	8	UDE : 更新的 DMA 请求使能
		0: 更新的 DMA 请求禁止 (默认)
		1: 更新的 DMA 请求使能
Bits	7:1	保留，必须保持复位值
Bit	0	UIE : 更新中断使能
		0: 更新中断禁止 (默认)
		1: 更新中断使能

17.6.3 SR 状态寄存器 (TIM6_SR)

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
Res.															

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

Bits	31:1	保留，必须保持复位值
Bit	0	UIF : 更新中断标志
		当产生更新事件时该位由硬件置 1，软件写 0 可清除该位
		0: 无更新事件产生 (默认)
		1: 更新中断挂起，当相关寄存器被更新时该位由硬件置 1
		- 若 TIM6_CR1 寄存器的 UDIS=0，当计数器上溢或下溢时
		- 若 TIM6_CR1 寄存器的 UDIS=0、URS=0，当软件设置 TIM6_EGR 寄存器的 UG 位对计数器 CNT 重新初始化时
		- 若 TIM6_CR1 寄存器的 UDIS=0、URS=0，当计数器 CNT 被触发事件重新初始化时

17.6.4 EGR 事件产生寄存器 (TIM6_EGR)

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
Res.															

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

Bits 31:1	保留，必须保持复位值
Bit 0	UG : 产生更新事件
	该位由软件置 1，用于产生一个更新事件，由硬件自动清 0
	0: 无动作（默认）
	1: 重新初始化计数器，并产生一个更新事件。注意预分频器的计数器也被清 0（但是预分频系数不变）

17.6.5 CNT 计数寄存器（TIM6_CNT）

地址偏移: 0x24

复位值: 0x0000_0000

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

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

Bits 31:16	保留，必须保持复位值
Bits 15:0	CNT[15:0] : 计数器的值

17.6.6 PSC 预分频寄存器（TIM6_PSC）

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
Res.															

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

Bits 31:16	保留，必须保持复位值
Bits 15:0	PSC[15:0] : 预分频器的值
	计数器的频率可以由下式计算: $f_{CK_CNT} = f_{CK_PSC} / (PSC[15:0] + 1)$ 。 预分频器的值由预装载寄存器写入，新的预分频器的值在下一次更新事件到来时被采用。可以通过 AS 位（TIM6_CR1 寄存器）来选择读取 PSC 寄存器的值来自影子寄存器或预装载寄存器。

17.5.7 ARR 自动重载寄存器 (TIM6_ARR)

地址偏移: 0x2C

复位值: 0x0000_0000

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

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

Bits	31:16	保留, 必须保持复位值
Bits	15:0	ARR[15:0]: 自动重载寄存器的值
		ARR 是要加载到自动重载寄存器中的值。具体请参考 17.2.1 章节。可以通过 AS 位 (TIM6_CR1 寄存器) 来选择读取 ARR 寄存器的值来自影子寄存器或预装载寄存器。

18. 定时器间的互联

定时器之间可以通过联动机制共同工作，提供同步或事件联接功能，具体如表 18-1 所示。

表 18-1 定时器间的互联

	TRGI ¹		TI	TRGO	BKIN	BKIN2	OCREF_CLR
	ITR(0/1/2/3)	ETR					
TIM1	TIM15_trgo TIM16_oc1_trig TIM3_trgo TIM17_oc1_trig	GPIO_ETR 模拟看门狗 ACMP0 输出 ACMP1 输出 ACMP2 输出 TIM15_oc1_trig TIM3_oc1_trig TIM14_oc1_trig	GPIO_TI 模拟看门狗 ACMP0 输出 ACMP1 输出 ACMP2 输出 TIM15_oc1_trig TIM3_ooc1_trig TIM14_oc1_trig	UG 软件更新位 CEN 计数器使能位 更新事件 比较脉冲 OC1REF 信号 OC2REF 信号 OC3REF 信号 OC4REF 信号	GPIO_BKIN 模拟看门狗 TIM15_TRGO TIM3_TRGO ACMP0 输出 ACMP1 输出 ACMP2 输出	模拟看门狗 TIM16_oc1_trig TIM17_oc1_trig ACMP0 输出 ACMP1 输出 ACMP2 输出	模拟看门狗 TIM15_TRGO TIM3_TRGO ACMP0 输出 ACMP1 输出 ACMP2 输出
TIM15	TIM1_trgo TIM3_trgo TIM16_oc1_trig TIM17_oc1_trig	保留 模拟看门狗 ACMP0 output ACMP1 output ACMP2 output TIM1_oc1_trig TIM3_oc1_trig TIM14_oc1_trig	GPIO_TI 模拟看门狗 ACMP0 output ACMP1 output ACMP2 output TIM1_oc1_trig TIM3_oc1_trig TIM14_oc1_trig	UG 软件更新位 CEN 计数器使能位 更新事件 比较脉冲 OC1REF 信号 OC2REF 信号 OC3REF 信号 OC4REF 信号	GPIO_BKIN 模拟看门狗 TIM1_TRGO TIM3_TRGO ACMP0 输出 ACMP1 输出 ACMP2 输出	模拟看门狗 TIM16_oc1_trig TIM17_oc1_trig ACMP0 输出 ACMP1 输出 ACMP2 输出	无
TIM16	无	保留 模拟看门狗 ACMP0 输出 ACMP1 输出 ACMP2 输出 TIM1_TRGO TIM3_TRGO TIM15_TRGO	GPIO_TI 模拟看门狗 ACMP0 输出 ACMP1 输出 ACMP2 输出 TIM1_TRGO TIM3_TRGO TIM15_TRGO	OC1REF 信号	GPIO_BKIN 模拟看门狗 TIM1_TRGO TIM3_TRGO ACMP0 输出 ACMP1 输出 ACMP2 输出	模拟看门狗 TIM15_TRGO TIM17_oc1_trig ACMP0 输出 ACMP1 输出 ACMP2 输出	无
TIM17	无	保留 模拟看门狗 ACMP0 输出 ACMP1 输出 ACMP2 输出 TIM1_TRGO TIM3_TRGO TIM15_TRGO	GPIO_TI 模拟看门狗 ACMP0 输出 ACMP1 输出 ACMP2 输出 TIM1_TRGO TIM3_TRGO TIM15_TRGO	OC1REF 信号	GPIO_BKIN 模拟看门狗 TIM1_TRGO TIM3_TRGO ACMP0 输出 ACMP1 输出 ACMP2 输出	无	无
TIM6	无	无	无	无	无	无	无

TIM3	TIM1_trgo TIM17_oc1_trig TIM15_trgo TIM14_oc1_trig	GPIO_ETR 模拟看门狗 ACMP0 输出 ACMP1 输出 ACMP2 输出 TIM1_oc1_trig TIM15_oc1_trig TIM16_oc1_trig		UG 软件更新位 CEN 计数器使能位 更新事件 比较脉冲 OC1REF 信号 OC2REF 信号 OC3REF 信号 OC4REF 信号	无	无	模拟看门狗 TIM15_TRGO TIM1_TRGO ACMP0 输出 ACMP1 输出 ACMP2 输出
TIM14	无	无	GPIO_TI rtc_clk osch_clk_div mco	OC1REF 信号	无	无	无

注 1: TIMx_OC_i_trig 信号不是直接输出的通道信号，此信号还有 CEN 位相关，具体详见 TIMx 章节 CR2 寄存器 TrgIS 位说明。

注 2: 模拟看门狗为 ADC 模块的模拟窗口看门狗输出，具体详见第 10 章节 ADC 的模拟看门狗相关说明。

注 3: TRGO 的具体配置详见 TIMx 章节 CR2 寄存器 MMS 位说明。

注 4: ITR 的具体配置详见 TIMx 章节 SMCR 寄存器 TS 位说明。

注 5: ETR / TI / BKIN / BKIN2 / OCREF_CLR 的具体配置详见第 6 章系统控制单元的寄存器描述。



19. 独立看门狗 (IWDG)

19.1 IWDG 主要特性

LCM32F037 有两个嵌入式看门狗（独立看门狗 IWDG 和窗口看门狗 WWDG），提供了高安全性、定时准确性和使用灵活性的组合。两个看门狗外设都用于检测 and 解决由于软件故障引起的故障。

独立的看门狗 (IWDG) 由自己的专用时钟 RCL 进行计时，因此即使主时钟发生故障也能保持活动状态。IWDG 最适合应用于需要看门狗作为一个在主程序之外、能够完全独立工作的进程运行，但对时间精度要求较低的场景。IWDG 还可以实现计时窗口的功能。IWDG 具有下述特性：

- 可编程的自由运行 12 位递减计数器
- 由内部 RCL 提供时钟，可以工作在 STOP 和 ULP STOP 模式下，可将系统从低功耗模式中唤醒
- 有条件的复位（需要开启看门狗）
 - 当递减计数器的数值达到 0x000 时产生复位
 - 当在指定的时间窗口之外重加载递减计数器的数值时产生复位

19.2 功能描述

当在键寄存器 (IWDG_KR) 中写入数值 0xCCCC 后，IWDG 就被启动了，计数器开始从它的复位值 0xFFFF 开始递减计数，当计数减到 0x000 时就会产生一个复位信号 (IWDG RESET)。

每当 0xAAAA 写入到 IWDG_KR 寄存器时，IWDG 将用 IWDG_RLR 的数值重装载到计数器中，从而避免了产生 IWDG 复位。如果开启了窗口值，则在窗口值外重装载计数器的值也会触发 IWDG 的复位。

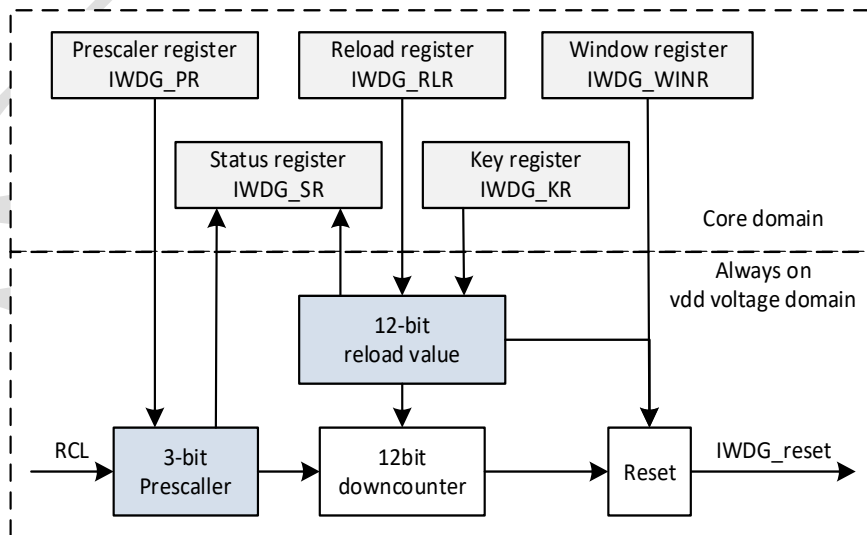


图 19-1 独立看门狗 IWDG 框图

19.2.1 当窗口功能使能时配置 IWDG

1. 通过往 IWDG_KR 寄存器写入 0x0000_CCCC，使能 IWDG；
2. 通过往 IWDG_KR 寄存器写入 0x0000_5555，允许寄存器访问；

3. 通过往 IWDG_PR 寄存器写入 0 到 7 值，编程 IWDG 预分频器；
 4. 通过往 IWDG_RLR 寄存器写入重装载值，设置重装载寄存器；
 5. 通过往 IWDG_WINR 寄存器写入窗口寄值，这将自动将 IWDG_RLR 值载入到计数器中；
 6. 通过往 IWDG_KR 寄存器写入除 0x0000_5555、0x0000_AAAA、0x0000_CCCC 外的其他值，开启写保护；
 7. 等待寄存器被更新（查询 IWDG_SR = 0x0000_0000）；
 8. 在窗口期间内，通过往 IWDG_KR 寄存器写入 0x0000_AAAA，重装载计数器值，避免 IWDG 产生复位。
- 注：**写入窗口值时当 IWDG_SR 为 0x0000_0000 时，会重加载计数器的值为 RLR 的值。

19.2.2 当窗口功能不使能时配置 IWDG

1. 通过往 IWDG_KR 寄存器写入 0x0000_CCCC，使能 IWDG；
 2. 通过往 IWDG_KR 寄存器写入 0x0000_5555，允许寄存器访问；
 3. 通过往 IWDG_PR 寄存器写入 0 到 7 的值，编程 IWDG 预分频器；
 4. 通过往 IWDG_RLR 寄存器写入重装载值，设置重装载寄存器；
 5. 通过往 IWDG_KR 寄存器写入除 0x0000_5555、0x0000_AAAA、0x0000_CCCC 外的其他值，开启写保护；
 6. 等待寄存器被更新（IWDG_SR = 0x0000_0000）；
 7. 通过往 IWDG_KR 寄存器写入 0x0000_AAAA，重装载计数器值，避免 IWDG 产生复位。
- 注：**单次执行动作刷新寄存器值时，最小时间间隔为 1 个 RCL 时钟；如果周期性执行动作，则每次间隔周期至少大于 4 个 RCL 时钟。

19.2.3 寄存器访问保护

保护对 IWDG_PR 和 IWDG_RLR 寄存器的写访问。要修改它们，首先在 IWDG_KR 寄存器中写入 0x5555。如果使用不同的值对该寄存器进行写访问，将打破这个顺序，寄存器访问将再次受到保护。这意味着这是重新加载操作的情况（写入 0xAAAA）。

状态寄存器可用来指示正在进行的预分频器或下计数器重新加载值的更新。

19.2.4 硬件看门狗功能

如果在 WDG_SW 选项字节中使能了硬件看门狗的功能，在芯片上电时看门狗的功能被自动开启，如果软件不能及时操作密钥寄存器，则在计数器达到 0x00 时产生复位。关于选项字节的内容请参考数据手册中的说明。

19.2.5 调试模式

当 MCU 进入调试模式（CPU 停止），IWDG 计数器继续正常工作或停止，取决于 DBG 模块中的 DBG_IWDG_STOP 配置位。

19.3 IWDG 寄存器描述

表 19-1 IWDG 相关寄存器表

名称	说明	读写权限	复位值	字节地址
KR	密钥寄存器	W	0x0000_XXXX	0x4001_7400
PR	预分频寄存器	R/W	0x0000_0000	0x4001_7404
RLR	重装载寄存器	R/W	0x0000_0FFF	0x4001_7408
SR	状态寄存器	R/W	0x0000_0000	0x4001_740C

WINR	窗口寄存器	R/W	0x0000_OFFF	0x4001_7410
------	-------	-----	-------------	-------------

注：x 表示不确定；-表示未实现；u 表示不受复位影响；c 表示该位复位值因复位源而异；R 只读；W 只写；R-0 表示只读且读 0；R/W 表示可读写（以后章节同上述）。

19.3.1 IWDG 密钥寄存器（IWDG_KR）

地址偏移：0x00

复位值：0x0000_XXXX

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

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

Bits 31:16	保留，必须保持复位值
Bits 15:0	KEY[15:0] ：密钥值（只写，读为 0x00）
	写入 0xAAAA 将刷新 IWDG 计数器的计数值，如果超出规定时间则产生复位
	KEY 写入 0xCCCC 将启动 IWDG，除非硬件已经开启了 IWDG
	写入 0x5555 将允许对受保护的 IWDG_PR、IWDG_RLR 和 IWDG_WIN 寄存器的操作；写入非 0x5555 将恢复这些寄存器的写保护状态。

19.3.2 IWDG 预分频寄存器（IWDG_PR）

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
Res.															

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

Bits 31:3	保留，必须保持复位值
Bits 2:0	PR[2:0] ：预分频值
	它们用于指定对计数器时钟分频的分频系数。
	IWDG_SR.PVU 位必须处于 0 时，才能改变预分频值。
	000：分频系数=4（默认）
	001：分频系数=8
	010：分频系数=16
	011：分频系数=32
	100：分频系数=64

	101: 分频系数=128
	110: 分频系数=256
	111: 分频系数=256

19.3.3 IWDG 重载寄存器 (IWDG_RLR)

地址偏移: 0x08

复位值: 0x0000_0FFF

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

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

Bits	31:12	保留, 必须保持复位值
Bits	11:0	RL[11:0]: 看门狗计数器重载值
		每次在 IWDG_KR 寄存器中写入 0xAAAA 时, 重载值会被载入到看门狗的计数器中, 看门狗的计数器将重新从这个值开始计数。超时时间由这个数值和时钟的预分频系数共同决定。
		IWDG_SR.RVU 位必须处于 0 时, 才能改变重载值。

19.3.4 IWDG 状态寄存器 (IWDG_SR)

地址偏移: 0x0C

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.													WVU	RVU	PVU
													r	r	r

Bits	15:3	保留, 必须保持复位值
Bit	2	WVU: 看门狗窗口值更新
		此位由硬件置 1 用来指示窗口值的值的更新正在进行中。当窗口值更新结束后, 此位由硬件清 0。
		看门狗计数器重载值只有在 WVU 位为 0 时才能更新。
Bit	1	RVU: 看门狗计数器重载值更新
		此位由硬件置 1 用来指示重载值的更新正在进行中。当重载更新结束后, 此位由硬件清 0。
		看门狗计数器重载值只有在 RVU 位为 0 时才能更新。

Bit 0	PVU : 看门狗预分频值更新
	此位由硬件置 1 用来指示预分频值的更新正在进行中。当预分频值更新结束后，此位由硬件清 0。
	看门狗预分频值只有在 PVU 位为 0 时才能更新。
注 : 所有位的置 1 或清 0，最多需要 5 个 RCL 时钟周期。	

19.3.5 IWDG 窗口寄存器 (IWDG_WINR)

地址偏移: 0x10

复位值: 0x0000_OFFF

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

Bits 31:12	保留，必须保持复位值
Bits 11:0	WIN[11:0] : 看门狗计数器窗口值
	看门狗计数器必须要在窗口值到 0 之前重装载才能避免发生复位。
	IWDG_SR.WVU 位必须处于 0 时，才能进行写或读操作。

20.窗口看门狗（WWDG）

20.1 WWDG 主要特性

窗口看门狗基于一个 7 位的递减计数器，可以被设置成自由运行，WWDG 计数器达到给定的超时值时触发系统复位或中断。它由时钟 PCLK0 驱动，有一个可配置的时间窗口，可以通过编程来检测异常的延迟或早期应用程序行为，具有早期预警中断功能。在调试模式下，计数器可以被冻结。

窗口看门狗用于监测由于外部干扰或不可预知的逻辑条件所产生的软件错误，这样的软件错误通常会导致应用程序不按照预期的方式运行。除非程序在递减计数器 T[6:0]的 bit 6 变为 0 之前刷新递减计数器，否则看门狗电路将在一个预置的时间间隔后产生系统复位；如果在 7 位的递减计数器数值达到窗口寄存器数值之前刷新递减计数器，同样会产生系统复位。这就意味着只能在一个有限的时间窗口内刷新递减计数器。WWDG 具有下述特性：

- 可编程的自由运行递减计数器
- 有条件的复位（开启看门狗）
 - 当递减计数器的数值小于 0x40 时产生复位
 - 当在指定的时间窗口之外重加载递减计数器的数值时产生复位
- 提前唤醒中断（EWI）：递减计数器的值递减到 0x40 的时候产生中断

20.2 功能描述

如果开启了看门狗（设置了 WDGA=1），当 7 位的递减计数器（T[6:0]位）从 0x40 变为 0x3F 时（即 bit 6 变为 0），看门狗产生一个复位信号并把复位引脚拉低。如果软件刷新计数器时，计数器的数值大于窗口寄存器中的数值，同样会产生复位。

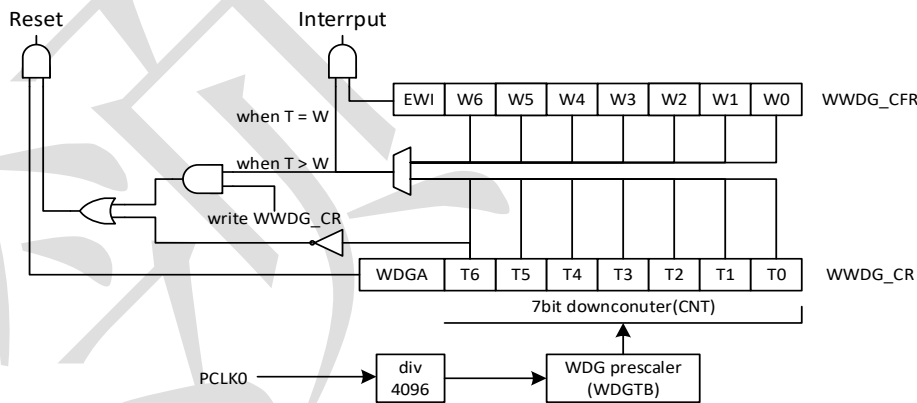


图 20-1 窗口看门狗 WWDG 逻辑图

在正常的操作期间，应用程序必须定期地写入 WWDG_CR 寄存器，以避免产生复位；这个写的动作必须在计数器的数值小于窗口寄存器的数值时进行。写入 WWDG_CR 寄存器的数值必须是介于 0xFF 和 0xE0 之间。

20.2.1 开启看门狗

在系统复位后看门狗处于关闭状态。设置 WWDG_CR 寄存器中的 WDGA 位和 EN 位将开启看门狗复位

功能和计数器，随后在下次复位之前将不能关闭看门狗。

20.2.2 控制递减计数器

递减计数器是可配置的自由运行计数器：即 EN 使能后 WDGA 使未开启看门狗，它依然不断地递减计数。当开启看门狗时，必须设置 T[6:0]的 bit 6 以保证写入值有效。T[5:0]位中包含了看门狗产生复位前允许的时间计数值；因为写入 WWDG_CR 寄存器时，预分频器的状态是不可知的，所以这个时间计数值介乎一个最小和最大数值之间。

窗口配置寄存器 (WWDG_CFR.W) 的数值是指定窗口的高限：为防止复位，必须在递减计数器的数值小于窗口寄存器的数值并大于 0x40 时刷新递减计数器。

注： T[6:0]的 bit 6 可以用于产生一个软件复位（即设置 WDGA 位同时 T[6:0] = 0x40）。

20.2.3 提前的看门狗中断

在产生复位之前需要进行安全处理和数据保存的应用场合，可以使用 EWI (Early Wakup Interrupt)，EWI 通过设置 WWDG_CR 的 EWI 位来使能，当递减计数器的值降低到 0x40 时产生 EWI 中断并且对应 ISR 中断服务程序可以被调用来处理特殊工作（在设备被 WWDG 产生的复位之前）。

有的应用需要利用 EWI 的 ISR 中断服务程序来处理一些任务，但是不需要复位系统，这时候可以在中断服务程序中重载 WWDG 计数器来避免复位产生。

EWI 中断可以通过写 0 到 WWDG_CR 的 EWI 位来清除。

20.2.4 计算 WWDG 溢出时间

WWDG 溢出时间计算公式如下：

$$t_{\text{WWDG}} = t_{\text{PCLK0}} * 2^{\text{WDGTB}[1:0]} * (\text{T}[5:0] + 1) \text{ (ms)}$$

其中： t_{WWDG} 为 WWDG 的超时溢出时间； t_{PCLK0} 为 PCLK0 时钟周期，单位 ms；WDGTB[1:0]为内置分频器系数。

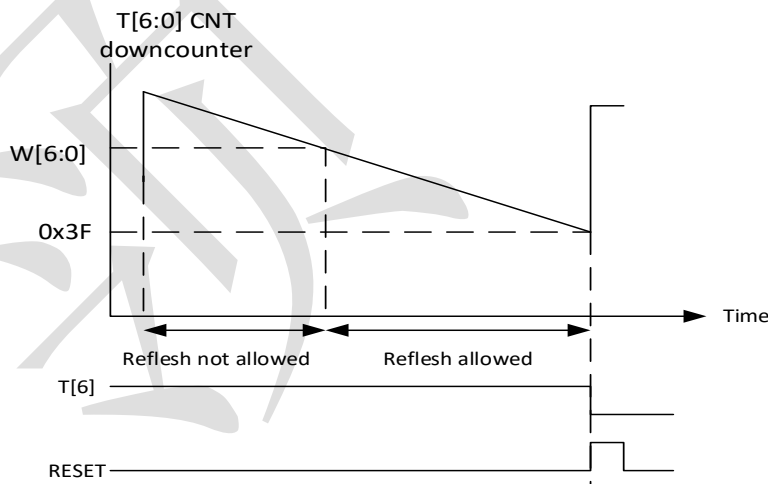


图 20-2 复位产生时序图

例如如果 PCLK0 的频率为 24MHz，WDGTB[1:0]配置为 3，T[5:0]配置为 63：

$$t_{\text{WWDG}} = (1/2400) * 4096 * 2^3 * (63 + 1) = 21.85 \text{ (ms)}$$

t_{WWDG} 的最小值和最大值请参见表 20-1。

表 20-1 最小和最大超时值@36 MHz (F_{PCLK0})

分频系数	WDGTB	最小超时溢出时间	最大超时溢出时间
1	0	113 us	7.28 ms
2	1	227 us	14.56 ms
4	2	445 us	29.12 ms
8	3	910 us	58.25 ms

20.2.5 调试模式

当微控制器进入调试模式（CPU 停止），WWDG 计数器继续正常工作或停止，取决于 DBG 模块中的 DBG_WWDG_STOP 配置位。

20.3 WWDG 寄存器描述

表 20-2 WWDG 相关寄存器表

名称	说明	读写权限	复位值	字节地址
CR	控制寄存器	R/W	0x0000_0000	0x4000_D000
CFR	配置寄存器	R/W	0x0000_0000	0x4000_D004
SR	状态寄存器	R/W	0x0000_0000	0x4000_D008

注：x 表示不确定；-表示未实现；u 表示不受复位影响；c 表示该位复位值因复位源而异；R 只读；W 只写；R-0 表示只读且读0；R/W 表示可读写（以后章节同上述）。

20.3.1 WWDG 控制寄存器（WWDG_CR）

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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.							EN	WDGA	T[6:0]						
							rw	rw	rw						

Bits	31:9	保留，必须保持复位值
Bit	8	EN: WWDG 使能位
		0: WWDG 计数器清零（默认）
		1: WWDG 计数器运行
Bit	7	WDGA: WWDG 激活位
		该位由软件置位，但是只能在复位的时候由硬件自动清零，当 WDGA = 1，WWDG 可以产生复位信号
		0: WWDG 复位禁止（默认）
		1: WWDG 复位使能

Bits 6:0	T[6:0]: 7 位计数器值 (MSB to LSB)
	WWDG 计数器的值。每 (4096 x 2WDGTB[1:0]) 个 PCLK0 周期递减 1, 当其值从 0x40 减到 0x3F 的时候 (T6 被清除) 产生一个复位信号
注: 写 CR 寄存器的时候, 位 31:16 必须写入 KEY 密钥值 (0x36AE)。	

20.3.2 WWDG 配置寄存器 (WWDG_CFR)

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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						EWI	WDGTB[1:0]		W[6:0]						
						rw	rw		rw						

Bits 31:10	保留, 必须保持复位值
Bit 9	EWI: 早期唤醒中断
	设置后, 每当计数器达到 0x40 时, 就会发生中断。这个中断只有在复位后才会被硬件清除。
Bit 8:7	WDGTB[1:0]: 计数器基本设置
	预分频器的计数器基本设置可以做如下修改
	00: 计数器时钟 = PCLK1 / (4096 * 1)
	01: 计数器时钟 = PCLK1 / (4096 * 2)
	10: 计数器时钟 = PCLK1 / (4096 * 4)
	11: 计数器时钟 = PCLK1 / (4096 * 8)
Bits 6:0	W[6:0]: 7 位窗口值
	这些位包含要与递减计数器进行比较的窗口值
注: 写 CR 寄存器的时候, 位 31:16 必须写入 KEY 密钥值 (0x36AE)。	

20.3.3 WWDG 状态寄存器 (WWDG_SR)

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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.															EWIF
r															rw

Bits 31:1	保留, 必须保持复位值
-----------	-------------

Bit 0	EWIF: 早期唤醒中断标志位
	当计数器达到 0x40 值时, 该位由硬件设置。它必须由软件通过写入 0 来清除。写 1 没有效果。如果中断未启用, 此位也可以被置位清除。

21. 系统时基定时器 (SysTick)

21.1 主要特性

系统时基定时器专用于实时操作系统, 也可用作一个标准的递减计数器。它具有下述特性:

- 24 位的递减计数器
- 自动重载功能
- 当计数器计为 0 时, 能产生一个可屏蔽的系统中断
- 可编程时钟源 (HCLK 或者 HCLK/8)

21.2 功能描述

M0 的内核中包含一个 SysTick 时钟, SysTick 为一个 24 位递减计数器; SysTick 设定初值并使能后, 每经过 1 个时钟周期, 计数值就减 1; 计数到 0, SysTick 计数器自动装载初值并继续计数, 同时内部的 COUNTFLAG 标志会置位, 触发中断 (前提是中断使能)。

如果外部晶振 (OSCH) 为 8MHz, 经过 PLL 的 8 倍频, 系统时钟则为 64MHz (即 HCLK 为 64MHz), SysTick 的频率可选为 8MHz 或 64MHz (即 HCLK 的 8 分频或不经过分频)。在选为 8MHz 的时钟频率条件下, 如果设置 SysTick 值为 8000, 则能产生 1ms 的时间基 (即 SysTick 产生 1ms 的中断)。

使用系统计数器的一些不同方法:

- 作为使用处理器时钟的高速报警计时器
- 作为可变速率报警或信号定时器。可用的持续时间范围取决于参考时钟和计数器的动态范围
- 作为一个简单的计数器, 软件可以使用它来测量完成时间和使用的时间
- 作为一个内部时钟源, 控制一段持续的时间。软件可以使用控件和状态寄存器中的 COUNTFLAG 字段, 用于确定操作是否在特定的持续时间内已完成, 作为动态时钟管理控制循环的一部分

当处理器在调试状态下时, 停止计数, VAL 值不再递减。由于计时器在使用参考时钟计时, 参考时钟是处理器时钟 (HCLK) 还是参考时钟源 (HCLK/8) 是用户定义的。如果选择使用参考时钟, 它必须记录处理器时钟和参考时钟之间的关系。这是系统计时器所必需的校准, 需要考虑到亚稳态、时钟偏差和抖动。

21.3 SysTick 寄存器描述

表 21-1 SysTick 相关寄存器表

名称	说明	读写权限	复位值	字节地址
SysTick_CTRL	控制和状态寄存器	R/W	0x0000_0000	0xE000_E010
SysTick_RELOAD	重装载值寄存器	R/W	0xFFFF_FFFF	0xE000_E014
SysTick_VAL	当前值寄存器	R/W	0xFFFF_FFFF	0xE000_E018
SysTick_CALIB	校准值寄存器	R	0x4000_EA5F	0xE000_E01C

注: x 表示不确定; -表示未实现; u 表示不受复位影响; c 表示该位复位值因复位源而异; R 只读; W 只写; R-0 表示只读且读 0; R/W 表示可读写 (以后章节同上述)。

21.3.1 SysTick 控制及状态寄存器 (SysTick_CTRL)

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
Res.															COUNTFLAG
															r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.												CLKSOURCE	CENCE	ENABLE	
												rw	rw	rw	

Bits 31:17	保留, 必须保持复位值
Bit 16	COUNTFLAG: 计数标志位, 当 SysTick 计数到 0 时, 该位变为 1, 读取寄存器会被清零
	0: Systick 计数未到 0 (默认)
	1: Systick 计数到 0
Bits 15:3	保留, 必须保持复位值
Bit 2	CLOCKSOURCE: SysTick 时钟源选择
	0: 参考时钟源 STCLK (HCLK/8) (默认)
	1: 内核时钟源 (HCLK)
Bit 1	TICKINT: SysTick 中断使能
	0: 禁止 (默认)
	1: 使能
Bit 0	ENABLE: SysTick 定时器使能
	0: 禁止 (默认)
	1: 使能

21.3.2 SysTick 重装值寄存器 (SysTick_RELOAD)

地址偏移: 0x04

复位值: 0xFFFF_FFFF

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RELOAD[23:0]															
rw															

Bits 31:24	保留, 必须保持复位值
Bits 23:0	RELOAD[23:0]: SysTick 定时器的重装值, 计数到 0 时该值将被重装

21.3.3 SysTick 当前值寄存器 (SysTick_VAL)

地址偏移: 0x08

复位值: 0xXXXX_XXXX

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

Bits	31:24	保留, 必须保持复位值
Bits	23:0	VAL[23:0] : SysTick 定时器的当前数值, 写入任何值都会清除寄存器, COUNTFLAG 也会清零 (不会引起 SysTick 中断)

21.3.4 SysTick 校准值寄存器 (SysTick_CALIB)

地址偏移: 0x0C

复位值: 0x4000_EA5F

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

Bit	31	NOREF : 表示是否有参考时钟
		0: 有参考时钟可用 (默认)
		1: 无参考时钟
Bit	30	SKEW : 表示 TENMS 域是否准确
		0: TENMS 域准确
		1: TENMS 域不准确 (默认)
Bits	23:0	TENMS[23:0] : 10 毫秒校准值, 默认值为 0xEA5F

22. 钟表定时器 (WT)

22.1 WT 主要特性

WT 在超低功耗停机模式时，由低功耗 LDO 供电；在其他模式下由内核 LDO 供电。只有配置 WT 时钟并使能 WT 时钟后（ChipCtrl_BDCR 寄存器中配置 WT_SEL，且 WT_EN 位置 1），用户才能正常访问 WT。WT 可以被软件复位（ChipCtrl_BDCR 寄存器中的 BDRST 位）。WT 主要用于产生实时中断，同时还作为蜂鸣器输出，主要特性如下：

- WT 的时钟源有 3 种可选：RCL、OSCL、OSCH 的 128 分频（OSCH_DIV）
- 8 位定时器有 4 个频率的时钟源 T8CK：4KHz，64Hz，1Hz，1/60Hz
- 可产生 8 位定时器溢出中断
- 为可输出周期为 0.5s 中断
- 可输出八种 BUZ 信号频率：高频四种 8192Hz、4096Hz、2048Hz、1024Hz，低频四种 2Hz、1Hz、0.5Hz、0.25Hz，并可输出对应的反向 nBUZ 信号

WT 的驱动时钟源如下（工作过程中不能切换时钟源）：

- 32.768KHz 的外部晶体振荡器（OSCL）
- 内部低功耗 RC 振荡器，典型频率为 24KHz（RCL）
- 高速外部时钟 OSCH 经 128 分频（OSCH_DIV）

22.2 功能描述

WT 模块主要用于产生实时中断并提供给其他模块时基。该模块主要包括前级分频电路和一个 8 位定时器 T8。前级分频电路产生 8 位定时器的时钟源并可直接产生 0.5 秒周期中断，同时提供蜂鸣器的时钟。需要使用蜂鸣器功能时必须开启 WT 模块。T8 定时器在上电复位有效的时候载入初值 0xFF，然后由 T8CK 的边沿触发 T8 递减计数，当计数值与 T8RL 的值匹配时，产生中断。最大可产生 256 分钟的定时。

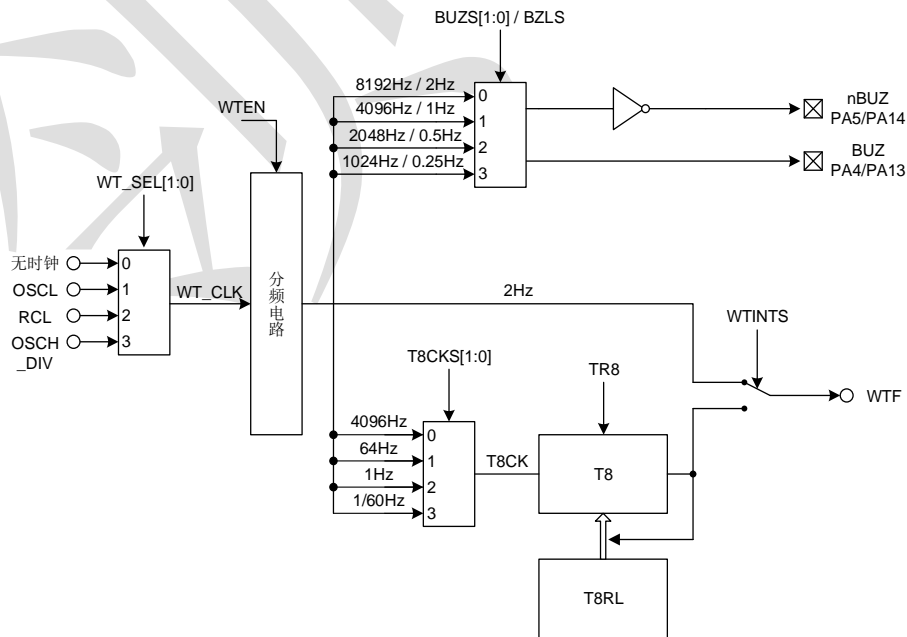


图 22-1 WT 框图

注：在使用WT时，建议选择外接低频晶振OSCL或者RCL作为时钟源。

WT的时钟源为WT_CLK，在低功耗模式下可利用WT模块实现定时或唤醒功能。

- 正常工作模式下，WT可正常工作，产生中断
- 低功耗模式下，由于WT时钟源WT_CLK可选择RCL或OSCL，仍可以工作并产生中断

22.3 WT 寄存器描述

表 22-1 WT 相关寄存器表

名称	说明	读写权限	复位值	字节地址
WTCON	控制寄存器	R/W	0x0000_0000	0x4001_7A00
T8/T8RL	定时寄存器	R/W	0x0000_00FF	0x4001_7A04

注：x表示不确定；-表示未实现；u表示不受复位影响；c表示该位复位值因复位源而异；R只读；W只写；R-0表示只读且读0；R/W表示可读写（以后章节同上述）。

22.3.1 WT 控制寄存器（WTCON）

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
Res.															
R															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.							BZLS	WTEN	WTF	BUZS[1:0]		TR8	T8CKS[1:0]		WTINTS
							rw	rw	rw	rw		rw	rw		rw

Bits 31:9	保留，必须保持复位值
Bit 8	BZLS： 蜂鸣器输出高低频率选择（配合BUZS位进行频率选择）
	0：输出高频（默认）
	1：输出低频
Bit 7	WTEN： WT使能
	0：WT禁止（默认）
	1：WT使能
Bit 6	WTF： WT中断标志位
	0：无中断产生（默认）
	1：产生中断
Bits 5:4	BUZS[1:0]： 蜂鸣器输出频率选择位（高频/低频）
	00：1024Hz/0.25Hz（默认）
	01：2048Hz/0.5Hz
	10：4096Hz/1Hz
	11：8192Hz/2Hz
Bit 3	TR8： T8使能
	0：T8关闭（默认）
	1：T8开启

Bits 2:1	T8CKS[1:0]: T8 时钟源选择
	00: 4096Hz (默认)
	01: 64Hz
	10: 1Hz
	11: 1/60Hz
Bit 0	WTINTS: WT 中断源选择
	0: 2Hz 中断 (默认)
	1: T8 递减溢出中断

22.3.2 WT 定时寄存器 (T8/T8RL)

地址偏移: 0x04

复位值: 0x0000_00FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								T8/T8RL[7:0]							
								rw							

Bits 31:9	保留, 必须保持复位值
Bits 7:0	T8/T8RL[7:0]: T8 定时器值
	T8 和 T8RL 两个寄存器地址共用, 写操作时选择 T8RL 寄存器, 读操作时选择 T8 寄存器。T8 为 8 位下行计数器计数值; T8RL 为 T8 初值寄存器。
	当 T8 计数器从 0xFF 递减到 T8RL 的值时产生匹配中断, 计数器重新置 0xFF 并开始下一计时周期。假设 T8RL 设置为 N, 那么计时间隔为 $(256-N)/T8CLK$ (由 T8CKS 选择位决定)。

23.I2C 接口

LCM32F037 最多支持 1 个 I2C 接口，I2C（芯片间）总线接口连接微控制器和串行 I2C 总线。它提供多主机功能，控制所有 I2C 总线特定的时序、协议、仲裁和定时。它支持标准模式，快速模式和超快速模式，支持 DMA 数据搬运。

23.1 I2C 主要特性

- 能够工作于多主机模式或从机模式
- 支持标准模式（最高 100Kbps）和快速模式（最高 400Kbps）和超快速模式（最高 1Mbps）
- 支持 7 位或 10 位寻址模式
- 支持批量传输模式
- 独立的工作时钟，高达 96MHz
- 带有 8Bytes 的发送和接收 FIFO，支持 DMA 操作
- 可编程的 SDA 保持时间

23.2 功能描述

I2C 模块接收和发送数据，并将数据从串行转换成并行，或者并行转换成串行。可以开启或者禁止中断。接口通过数据引脚（SDA）和时钟引脚（SCL）连接到 I2C 总线。允许连接标准（高达 100 Kb/s），快速模式（高达 400 Kb/s）或超快速模式（高达 1 Mb/s）I2C 总线，I2C 结构框图如下：

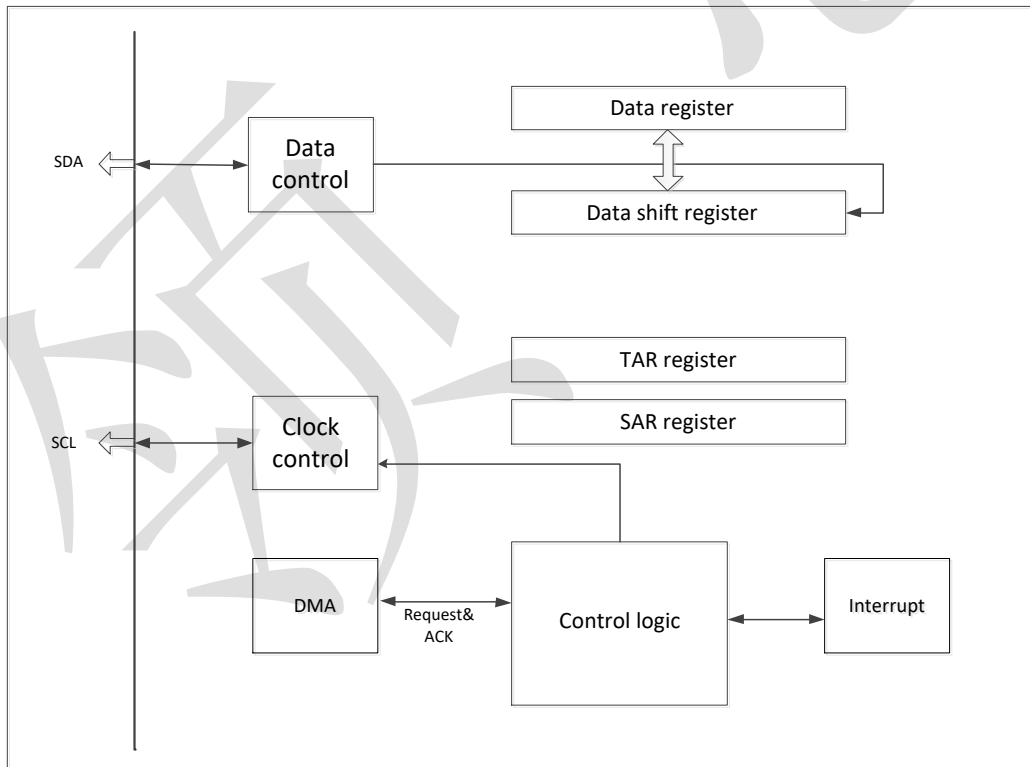


图 23-1 I2C 框图

I2C 有两种工作模式，作为主机与其他从机通讯，或者作为从机与其他主机通讯。作主机时负责产生时钟、控制数据传输。作从机时负责向主机发送或接收数据。每个数据的应答是正在接收数据的设备做出的（可以是主机或从机）。I2C 协议允许多个主机同时挂载在总线上，总线拥有通过仲裁决定权。

每个从机有唯一地址，由用户定义。当主机想跟从机通讯时，主机发送 START/RESTART 信号，紧跟着发送从机的地址和决定数据的方向的控制位（R/W）。然后从机返回 ACK 做出回应。

如果主机向从机进行写操作，从机会收到一字节的数据，传输会持续到主机发出 STOP 来终止。

如果主机向从机进行读操作，从机会向主机发送一字节数据，主机收到数据后返回 ACK，传输会持续到到主机收完最后一字节数据并返回 NACK，随即主机发送 STOP 或发出 RESTART 来寻址其他从机。

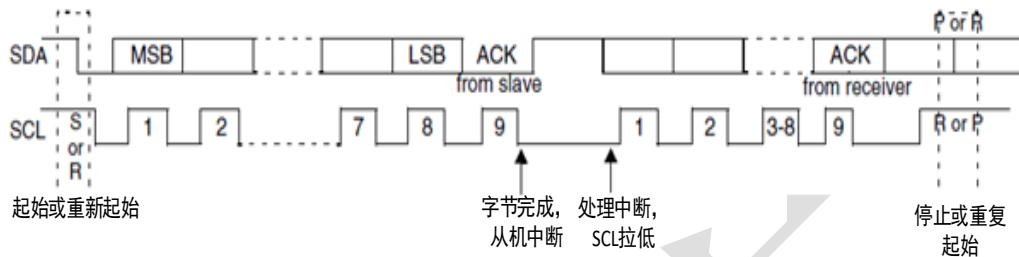


图 23-2 I2C 基本传输协议

I2C 是同步串行接口。双向信号线 SDA 除了特殊条件（STOP/START/RESTART）只能在 SCL 为低电平时改变电平。I2C 输出采用漏极开路或集电极开路实现总线线与功能。总线上所能支持设备的最大个数受 400pF 的最大容限限制。

23.2.1 操作模式

23.2.1.1 从机模式

初始化操作

为使 I2C 执行从机功能，需完成以下几步：

1. 使能关闭 I2C，配置 IC_ENABLE[0] = 0
2. 写 IC_SAR[9:0] 设置从机地址
3. 写 IC_CON 寄存器定义支持地址的类型（bit 3 设置 7 位或 10 位），通过对 bit 6 和 bit 0 写 0，在 slave-only 模式下使能 I2C
4. 置位 IC_ENABLE[0]，使能 I2C

从机单字节发送操作

当其他 I2C 主机设备寻址该 I2C 从机并请求数据，I2C 作为从机发送进行以下步骤：

1. 其他 I2C 主机发送地址，且匹配 I2C 内 IC_SAR 寄存器的地址
2. I2C 发送 ACK 回应发送的地址，识别传输的方向，指示当前 I2C 工作在从机发送模式
3. I2C 置位 RD_REQ 中断（IC_RAW_INTR_STAT 寄存器的 bit 5），并维持 SCL 低电平。I2C 在软件反应之前，进入等待模式

如果 RD_REQ 中断因 IC_INTR_MASK[5] 设置为 0 而被屏蔽，建议执行若干次读 IC_RAW_INTR_STAT 寄存器

- 读到 IC_RAW_INTR_STAT[5] 为 1 必须被认为等同于 RD_REQ 中断被置位
 - 相应软件必须执行，用于满足 I2C 传输
 - 推荐间隔时间应是 SCL 最快时钟周期的 10 倍。例如，对于 400kb/s，时间间隔是 25us
4. 如果在收到读请求之前，Tx FIFO 仍留有数据，I2C 产生 TX_ABRT 中断，刷新 TX FIFO 内的旧数据，如果 TX_ABRT 中断被屏蔽（IC_INTR_MASK[6] 被设置为 0），建议执行若干次读 IC_RAW_INTR_STAT 寄存器。
 - 读到 IC_RAW_INTR_STAT[6]=1，等同于 TX_ABRT 中断被置位

- 软件无需进一步操作
 - 间隔时间的应用同 IC_RAW_INTR_STAT[5]寄存器的操作步骤软件将数据写入 IC_DATA_CMD 寄存器
6. 软件在处理中断之前，必须清除 IC_RAW_INTR_STAT 的 RD_REQ 和 TX_ABRT 中断
 7. 如果 RD_REQ 和 TX_ABRT 中断被屏蔽，清 IC_RAW_INTR_STAT 操作在 R_RD_REQ 或 R_TX_ABRT 被读到 1 时，就已经执行过 I2C 释放 SCL 并传输字节
 9. 主机可以通过发起 RESTART 维持总线，通过发起 STOP 释放总线

单字节从机接收操作

当其他 I2C 主机寻址到当前 I2C 并发送数据时，当前 I2C 作为从机接收，进行以下步骤：

1. 其他 I2C 主机设备通过寻址匹配 IC_SAR 寄存器，发起 I2C 传输
2. I2C 发送 ACK 回应地址，识别传输的方向，指示当前 I2C 工作在从机接收模式
3. I2C 接收传送的字节，并将之放到接收 buffer 中
4. I2C 置位 RX_FULL 中断（IC_RAW_INTR_STAT[2]）

注：如果 RX_FULL 中断被屏蔽，由于将 IC_INTR_MASK[2] 寄存器设置为 0 或设置 IC_TX_TL 为大于 0 的值，则建议定期读取 IC_STATUS 寄存器。如果其中第 3 位（RFNE）设置为 1，则相当于 RX_FULL 中断被置位，则必须由软件进行处理。

5. 软件读取 IC_DATA_CMD 寄存器中的字节[7:0]
6. 另一个 I2C 主机可以通过发出 RESTART 条件来保持 I2C 总线，或者通过发出 STOP 条件释放总线

从机批量发送

在标准 I2C 协议中，所有传输都是单字节传输，用户通过将一个字节写入从机的 TX FIFO 来响应远程主器件读取请求。当从机（发送）响应主机（接收）读请求（RD_REQ）时，从机的 TX FIFO 中至少有一个数据。LCM32F037 的 I2C 的 TX FIFO 可以容纳多个数据，因此当主机有连续的读请求时候，不需要产生中断以让软件再向 TX FIFO 中写入数据，这可以消除中断相应和数据写入时间。

如果主机确认从机发送的数据后从机的 TX FIFO 中没有更多的数据，则 I2C 模块将 SCL 线保持为低电平，同时它会引发读取请求中断（RD_REQ），等待数据写入 TX FIFO 后再向 I2C 主机进一步发送数据。

如果 RD_REQ 中断被屏蔽，由于 IC_INTR_STAT 寄存器的第 5 位（R_RD_REQ）将一直为 0，则建议定时对 IC_RAW_INTR_STAT 寄存器进行周期性读操作。如果返回的 IC_RAW_INTR_STAT 寄存器中的第 5 位（RD_REQ）被设置为 1，则相当于产生 RD_REQ 中断，该操作类似于“从机单字节发送操作”中所述的操作。

RD_REQ 中断在 TX_FIFO 为空，但主机有新的读取请求时立刻产生，与其它中断类似，退出中断服务处理程序（ISR）时必须清除它。在中断服务程序中（ISR）允许将 1 个字节或多个 1 个字节写入 Tx FIFO。在将这些字节传输到主机后，如果主机 ACK 最后一个字节，则表示主机请求读取更多数据，从机必须再次发出 RD_REQ 中断。

如果用户预先知道 I2C 主机请求的数据包的大小（比如 n 个字节），那么当主机寻址 I2C 并请求数据时，Tx FIFO 可以写入 n 个字节，主机以数据流的形式连续接收这些数据，从机无需将 SCL 线保持为低电平或再次发出 RD_REQ。

如果主机要从从机 I2C 中接收 n 个字节，但程序将大于 n 的字节数据写入 Tx FIFO，则当从机完成发送所请求的 n 个字节后，会清除 Tx FIFO 并忽略任何多余的字节，产生中止（TX_ABRT）事件，以表示有清除 Tx FIFO 事件发生。从机在等待 ACK/NACK 时，如果接收到 NACK，则主机已经完成它所需所有数据的接收。此时从机产生一个 TX_ABRT 事件并以清除 Tx FIFO 中的所有剩余数据。

广呼模式

广呼模式是一种特殊的从机接收模式，通过从机地址全为 0 以及数据方向来定址。IC_ACK_GENERAL_CALL 寄存器的 bit 0 要置 1 来保证广呼的应答。

23.2.1.2 主机模式

初始化配置

1. 通过将 0 写入 IC_ENABLE[0] 来禁用 I2C 模块
2. 写入 IC_CON 寄存器以设置支持的速度模式 (bits[2:1]) 和所需的 I2C 主发起传输的速度, 7 位或 10 位寻址 (bit 4)。确保 bit 6 (IC_SLAVE_DISABLE) 写入 1 并且位 0 (MASTER_MODE) 写入 1。向 IC_TAR 寄存器写入要寻址的 I2C 器件的地址 (bits[9:0])。这个寄存器也指示是否将由 I2C 执行广播呼叫或 START BYTE 命令
3. 通过将 1 写入 IC_ENABLE 寄存器的 bit 0 来使能 I2C 模块
4. 向 IC_DATA_CMD 寄存器写入命令或数据。如果 IC_DATA_CMD 寄存器在 I2C 模块使能前写入, 会导致数据和命令丢失, 因为当 I2C 模块被禁用时, 缓冲区保持清除状态。此步骤在 I2C 模块上生成 START 和地址字节。一旦 I2C 使能且 TX FIFO 中有数据, I2C 开始读取 TX 数据并转换成 SCL, SDA 信号向从设备发送

主机发送和主机接收

I2C 模块主机模式支持读和写之间的动态切换。当要传输数据, 将写入的数据写入 I2C Rx/Tx 数据缓冲区和命令寄存器 (IC_DATA_CMD) 的低字节。对于 I2C 写操作, CMD bit 8 应写入 0。随后, 可以通过向 IC_DATA_CMD 寄存器的 CMD 位写入 1 来发出读命令。只要发送 FIFO 中存在命令, I2C 主机就会继续启动传输。如果发送 FIFO 变空, 检查 IC_DATA_CMD [9] 是否设置为 1:

- 如果设置为 0, 则保持 SCL 为低电平, 直到下一个命令写入发送 FIFO
- 如果设置为 1, 则在完成当前传输后发出 STOP 条件

I2C 使能关闭

当 IC_ENABLE 寄存器的第 0 位从 1 设置为 0 时, 软件需要查询寄存器 IC_ENABLE_STATUS 以确定 I2C 是否真正关闭。当 IC_ENABLE 被设置为 0 后, 只有当前命令完成执行, 且当前命令的 STOP 位被设置为 1, I2C 模块才能关闭, 如果正在处理的命令没有设置 STOP 位时尝试关闭 I2C 主机, I2C 将继续保持工作状态, 并保持 SCL 为低电平, 直到在 Tx FIFO 中收到新命令。参考流程如下:

1. 定义定时器间隔 (t_{i2c_poll}) 等于 I2C 模块支持的最高传输周期的 10 倍。例如, 如果 I2C 最高传输速率是 400 kb/s, 则 t_{i2c_poll} 为 25us
2. 定义最大超时参数 MAX_T_POLL_COUNT, 以便在任何重复轮询时使用, 查询操作超过此最大值, 报告错误
3. 保证软件不会在有任何启动 I2C 主机的操作, 但允许完成任何待处理的传输
4. 变量 POLL_COUNT 初始化成 0
5. 设置 IC_ENABLE[0] = 0
6. 读 IC_ENABLE_STATUS 寄存器并测试 IC_EN 位 (bit 0)。POLL_COUNT 加 1。如果 POLL_COUNT >= MAX_T_POLL_COUNT, 则退出并显示相关的错误代码
7. 如果 IC_ENABLE_STATUS [0] 为 1, 则休眠 t_{i2c_poll} 时间并继续执行上一步。否则, 表示 I2C 关闭成功, 退出相关操作

I2C 传输中止

IC_ENABLE 寄存器的 ABORT 控制位允许软件中止传输, 在 I2C 完成从 Tx FIFO 发出的传输命令之前。为了响应 ABORT 请求, 控制器通过 I2C 总线发出 STOP 条件, 然后清空 Tx FIFO。中止传输仅在主操作模式下被允许。具体操作流程如下:

1. 停止向 Tx FIFO (IC_DATA_CMD) 写入新的命令或者数据
2. 如果在 DMA 模式下, 通过将 TDMAE 设置为 0 禁用发送 DMA

3. 将 IC_ENABLE 寄存器 (ABORT) 的第 1 位设置为 1
4. 等待 TX_ABRT 中断
5. 读取 IC_TX_ABRT_SOURCE 寄存器以将源标识为 ABRT_USER_ABRT

寻址方式

I2C 支持两种寻址格式：7 位地址和 10 位地址

- 7 位地址格式

7 位地址格式中，高 7 位 (bits[7:1]) 定义从机的地址，最低位定义操作方向 (R/W)。当 bit 0 设成 0，主机向从机写数据。当 bit 0 设成 1，主机从从机读数据。

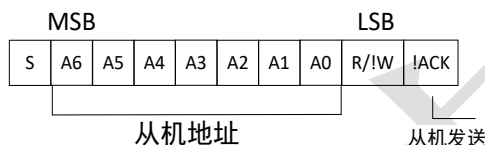


图 23-3 I2C 7 位地址格式

- 10 位地址格式

10 位地址格式中，用 2 个字节来设置 10 位地址。首字节的前 5 位 (bits[7:3]) 通知从机这是 10 位地址传输 (前 5 位含义见图 23-4)，紧跟 2 位 (A9、A8) 设置从机地址的 bits[9:8]，最低位定义 R/W 位。第二个字节设置从机地址的 bits[7:0]。

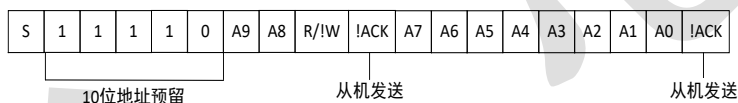


图 23-4 I2C 10 位地址格式

表 23-1 10 位地址的前五位含义表

首字节前五位和 A9、A8	R/W 位	说明
0000 000	0	广呼地址，I2C 把数据放入接收 buffer，并产生广呼中断
0000 000	1	起始字节
0000 001	x	CBUS 地址，I2C 忽略这些访问
0000 010	x	保留
0000 011	x	保留
0000 1xx	x	高速主机编码
1111 1xx	x	保留
1111 0xx	x	10 位从机寻址

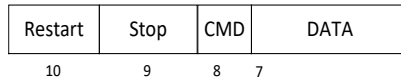
TX FIFO 管理

当 LCM32F037 作为主机操作时，I2C 组件支持两种 Tx FIFO 管理模式。

可以使用 IC_EMPTYFIFO_HOLD_MASTER_EN 参数在这两种模式之间进行选择。

LCM32F037 集成的 I2C 在发送 FIFO 为空的时候，I2C 维持 SCL 低电平，不产生 STOP，在新的数据写入发送 FIFO 前，停止总线。只有当用户写 IC_DATA_CMD 的 bit 9 (STOP 位) 特意请求时，STOP 条件产生。

IC_DATA_CMD



DATA-Read/Write 字段	从中读取从机接收的数据；写入从机即将发送的数据
CMD -Write-only 字段	这个位决定数据的方向，读取（CMD = 1）或写入（CMD = 0）
Stop -Write-only 字段	该位确定在发送或接收数据字节后是否生成 STOP
Restart-Write-only 字段	这个位确定在收到或发送数据字节之前是否重启

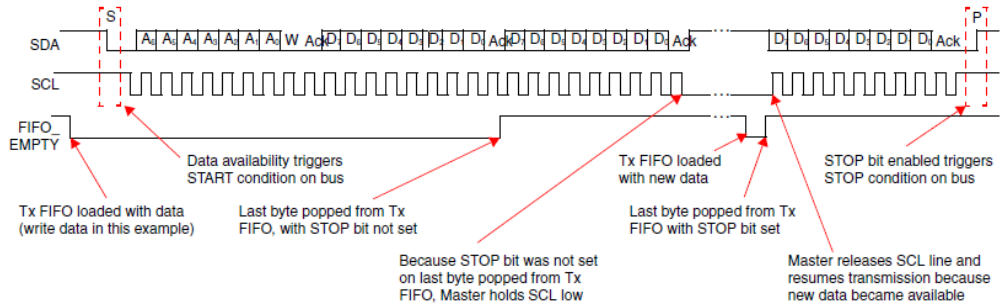


图 23-5 主机发送：发送 FIFO 空/STOP 产生时序图

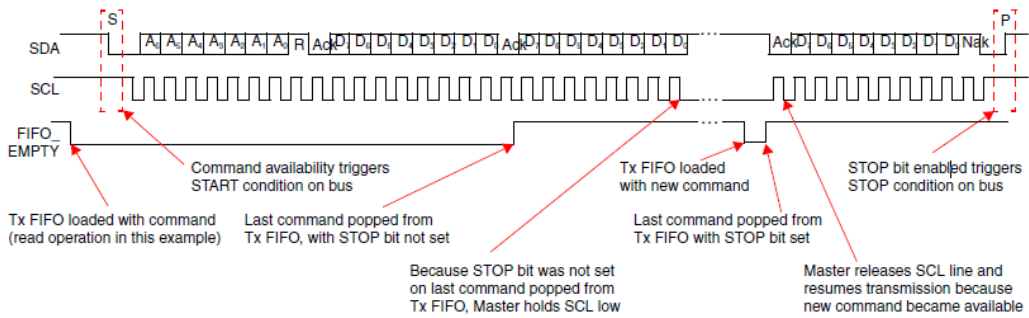


图 23-6 主机接收：发送 FIFO 空/STOP 产生时序图

此外，用户可控制总线上 RESTART 条件的产生。如果 IC_DATA_CMD 的 bit 10 被置位，在数据写入从机或从从机读数据前产生 RESTART 条件。如果 RESTART 不使能，一个停止位紧跟起始位产生。I2C 作为主机发送模式时，在该情形的时序如图 23-7 所示。

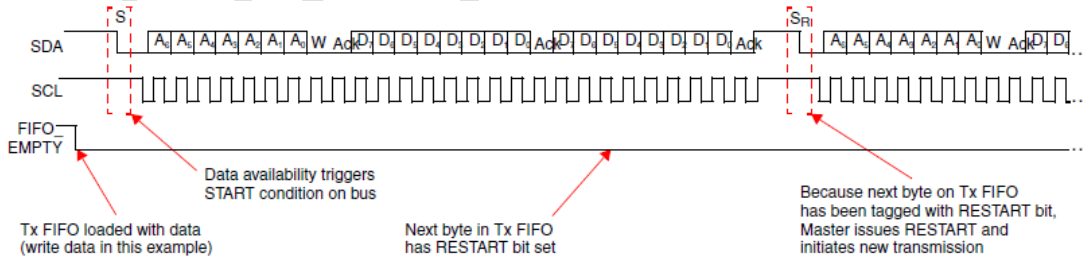


图 23-7 主机发送：IC_DATA_CMD 的 Restart 置位

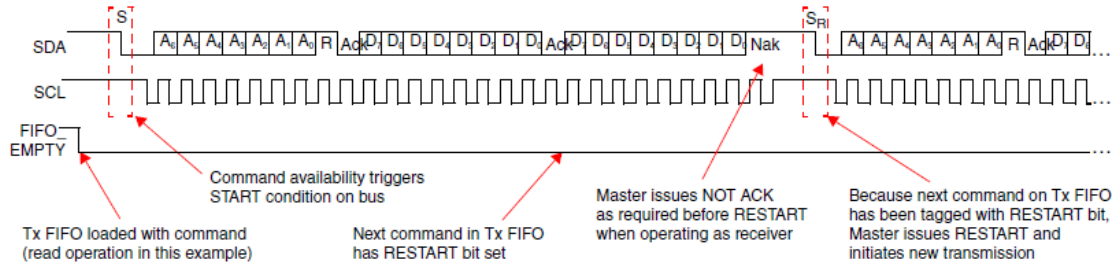


图 23-8 主机接收：IC_DATA_CMD 的 Restart 置位

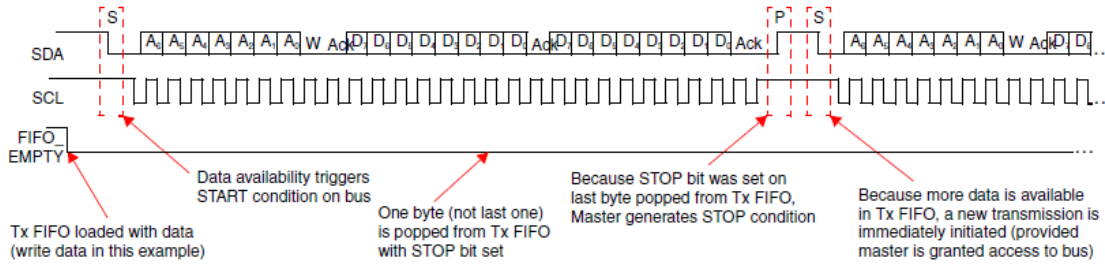


图 23-9 主机发送：IC_DATA_CMD 的 Stop 置位/发送 FIFO 非空

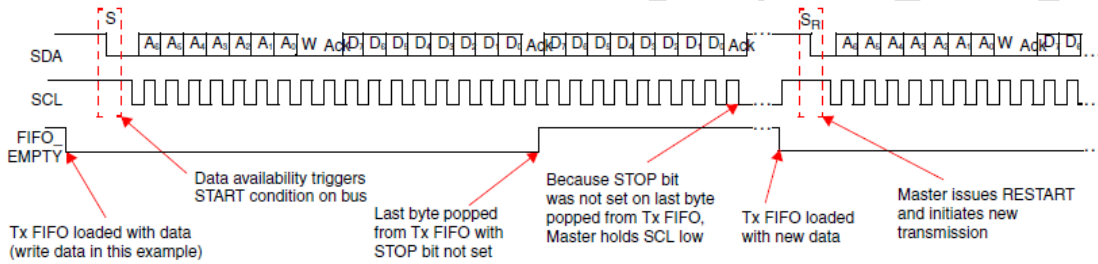


图 23-10 主机发送：FIFO 为空后载入的首字节，Restart 置位

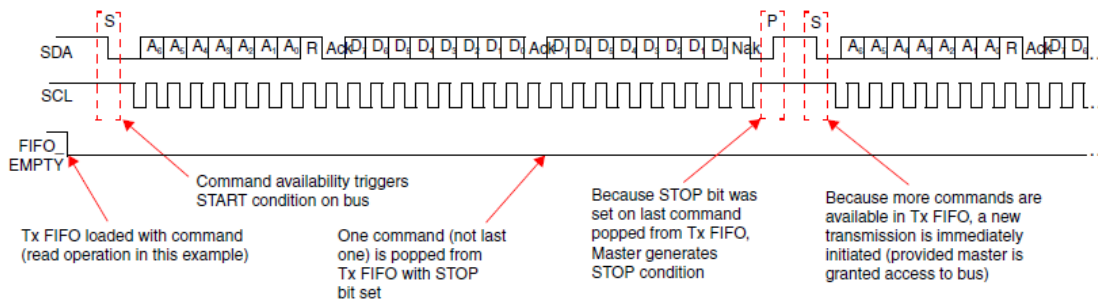


图 23-11 主机接收：IC_DATA_CMD 的 Stop 置位/发送 FIFO 非空

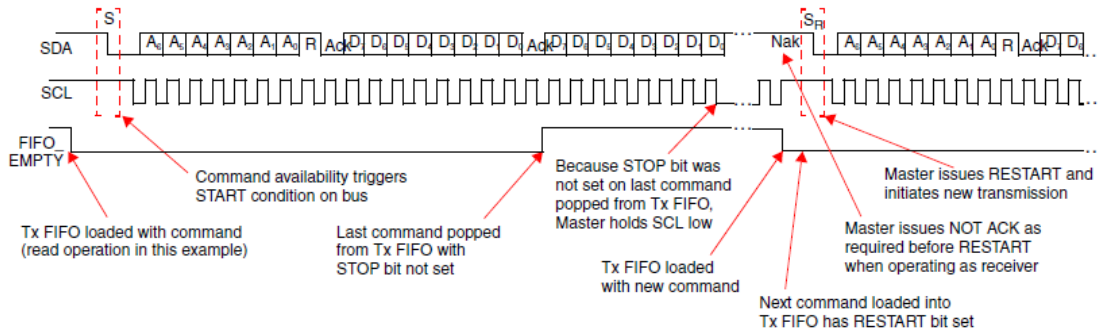


图 23-12 主机接收：允许 Tx FIFO 为空/Restart 置位后，加载第一个命令

多主机仲裁

I2C 协议允许多主机挂载在总线上。如果总线上同时有两个主机，如果两个同时发送 START 位试图占用总线，产生一个仲裁过程。一旦某个主机控制总线，在这个主机发送 STOP 条件释放总线前，其他主机不能接管总线。

当 SCL 为高电平时，在 SDA 线上发生仲裁。当主机发送 1，而其他主机发送 0 的情况下，该主机丢失仲裁并关闭数据输出状态。丢失仲裁的主机可以在该字节结束前持续产生时钟。如果多个主机同时寻址同个从机设备，仲裁可在数据阶段进行。

LCM32F037 的 I2C 一旦侦测到仲裁被其他主机获得，I2C 停止产生 SCL。

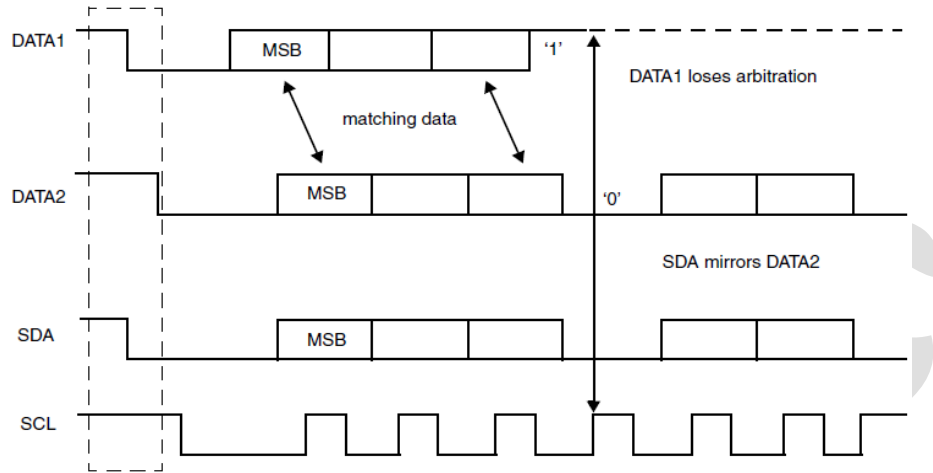


图 23-13 两个主机仲裁的时序

23.2.2 超快速模式

在超快速模式下，I2C 允许快速模式操作扩展到支持高达 1000 Kb/s 的速度。要启用 I2C 以进行超快速模式，请在启动任何数据传输之前执行以下步骤：

1. 将 I2C 主机或从机的最大速度模式配置为超快速模式（`IC_MAX_SPEED_MODE = 2`）
2. 设置 `IC_CLK` 的频率大于等于 32MHz
3. 配置 `IC_CON[2:1] = 2'b10`，进入快速模式或超快速模式
4. 配置 `IC_FS_SCL_LCNT` 和 `IC_FS_SCL_HCNT` 寄存器以满足超快速 SCL
5. 编程 `IC_FS_SPKLEN` 寄存器，以抑制最大尖峰 50ns
6. 编程 `IC_SDA_SETUP` 寄存器，满足最小数据建立时间

23.2.3 I2C_CLK 时钟配置

当 I2C 配置为主机时，必须先设置 *CNT 寄存器，然后才能进行任何 I2C 总线传输，以确保正确的 I/O 时序。*CNT 寄存器是：`IC_SS_SCL_HCNT`、`IC_SS_SCL_LCNT`、`IC_FS_SCL_HCNT` 和 `IC_FS_SCL_LCNT`。

注：*CNT 寄存器仅用于作为 I2C 主设备运行确定 SCL 时序的要求，作为从机时无需配置。

23.2.3.1 最小 I2C_CLK 频率

本节介绍 I2C 每种速度模式支持的 I2C_CLK 最小频率，以及相关的高/低计数值。在从机模式下，需要对 `IC_SDA_HOLD` 和 `IC_SDA_SETUP` 进行编程，以满足 I2C 协议时序要求。

23.2.3.2 标准模式，快速模式和超快速模式

以下方法显示了如何进行快速模式计算，也可以使用相同的方法来进行标准模式和超快速模式的计算。给出快速模式下最小 I2C_CLK 值的条件和计算方式：

- 快速模式的数据速率为 400kb / s: SCL 周期为 $1 / 400\text{kHz} = 2.5\mu\text{s}$
- 最小 hcnt 值为 14, IC_HCNT_FS = 14
- 协议最小 SCL 高低时间:
 - MIN_SCL_LOWtime_FS = 1300ns
 - MIN_SCL_HIGHTime_FS = 600ns

派生方程式：

$$\frac{SCL_PERIOD_FS}{IC_HCNT_FS + IC_LCNT_FS} = IC_CLK_PERIOD$$

$$IC_LCNT_FS \times IC_CLK_PERIOD = MIN_SCL_LOWtime_FS$$

上述等式产生以下：

$$IC_LCNT_FS \times \frac{SCL_PERIOD_FS}{IC_LCNT_FS + IC_HCNT_FS} = MIN_SCL_LOWtime_FS$$

求解 IC_LCNT_FS:

$$IC_LCNT_FS \times \frac{2.5\mu\text{s}}{IC_LCNT_FS + 14} = 1.3\mu\text{s}$$

由以上等式得出：

$$IC_LCNT_FS = \text{roundup}(15.166) = 16$$

这些计算产生 IC_LCNT_FS = 16 和 IC_HCNT_FS = 14, 给出 I2C_CLK 的频率：

$$\frac{2.5\mu\text{s}}{16 + 14} = 83.3\text{ns} = 12\text{Mhz}$$

表 23-2 具有高/低计数值的所有模式的最小 I2C_CLK 值

速度模式	I2C_CLK (MHz)	IC*_SPKLEN 最小值	SCL 低电平 (I2C_CLK)	SCL 低电平编程值	SCL 低电平时间	SCL 高电平 (I2C_CLK)	SCL 高电平编程值	SCL 高电平时间
标准模式	2.7	1	13	12	4.7us	14	6	5.2us
快速模式	12.0	1	17	15	1.33us	14	6	1.16us
超快速模式	32	2	17	15	500ns	16	7	500ns

23.2.3.3 计算高/低计数值

下面的计算显示了如何计算 I2C 中每种速度模式的 SCL 高和低计数。为使计算起作用，所用的 I2C_CLK 频率不得小于表 23-2 中规定的 I2C_CLK 最小频率。

计算设置适当的 SCL 时钟高和低时间所需的正确 I2C_CLK 的公式如下：

$$IC_xCNT = (\text{ROUNDUP}(\text{MIN_SCL_xxtime} * \text{OSCFREQ}, 0))$$

ROUNDUP 是将实数转换为它的等效整数的函数。

MIN_SCL_HIGHTime = 最短高周期: MIN_SCL_HIGHTime = 4000 ns 对应 100 kbps, 400 ns 对应 400 kbps, 260 ns 对应 1000 kbps。

MIN_SCL_LOWtime = 最小低周期: MIN_SCL_LOWtime = 4700 ns 对应 100 kbps, 1300 ns 对应 400 kbps, 500 ns 对应 1000 kbps。

OSCFREQ = I2C_CLK 时钟频率 (Hz)。

例如：

OSCFREQ = 100 MHz

I2Cmode =快速, 400 kbps
 MIN_SCL_HIGHTime = 600 ns
 MIN_SCL_LOWtime = 1300 ns
 IC_xCNT = (ROUNDUP (MIN_SCL_HIGH_LOWtime * OSCFREQ, 0))
 IC_HCNT = (ROUNDUP (600 ns * 100 MHz, 0))
 IC_HCNTSCL PERIOD = 60
 IC_LCNT = (ROUNDUP (1300 ns * 100 MHz, 0))
 IC_LCNTSCL PERIOD = 130
 实际 MIN_SCL_HIGHTime = 60 * (1/100 MHz) = 600 ns
 实际 MIN_SCL_LOWtime = 130 * (1/100 MHz) = 1300 ns

23.2.4 DMA 操作

要启用 I2C 的 DMA 控制器接口, 必须写入 DMA 控制寄存器 (IC_DMA_CR)。将 1 写入 IC_DMA_CR 寄存器的 TDMAE 位, 可启用 I2C 的发送握手接口。将 1 写入 IC_DMA_CR 寄存器的 RDMAE 位, 可启用 I2C 接收握手接口。

作为块流控制设备, DMA 控制器的参数由软件编程, 用户将 I2C 要接收或发送的数据 (块大小) 的数量编程到 DMA 的 CHn_CTRL_A 寄存器的 BLOCK_TL 字段中。该块被分成许多个传输事务, 每个事务都由 I2C 的请求启动, 源和目标的每次传输的数据长度分别编程到 CHn_CTRL_A 寄存器的 SRC_TR_WIDTH / DST_TR_WIDTH 字段中。

下图显示了单块传输, 其中编程到 DMA 控制器中的块大小为 12, 单次突发事务长度设置为 4。在这种情况下, 块大小是突发事务长度的倍数。因此, DMA 块传输一系列突发事务。如果 I2C 向 DMA 通道发送传输请求, 则 DMA 将 4 个数据写入 I2C TX FIFO。类似地, 如果 I2C 向 DMA 发起接收请求, 则 DMA 从 I2C 的 RX FIFO 读取 4 个数据项。要写入或读取所有 12 个数据, I2C 必须向此 DMA 通道发出 3 个单独的请求。

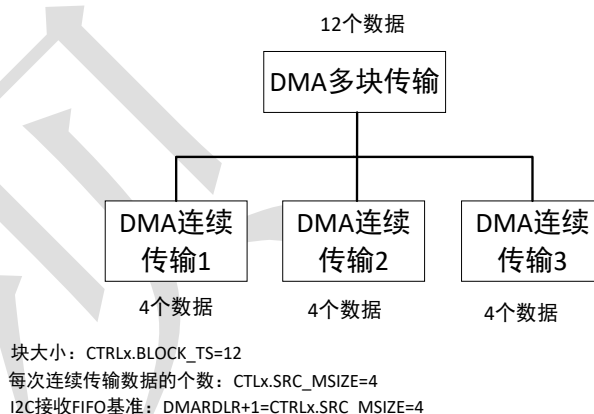


图 23-14 DMA 连续传输分解

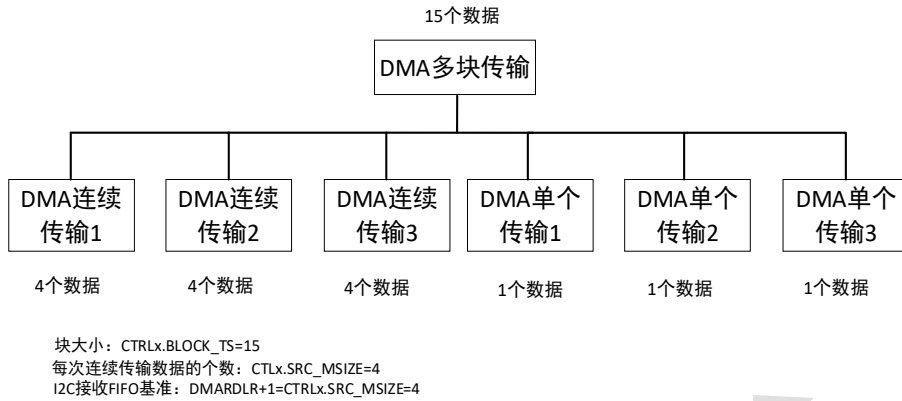


图 23-15 DMA 单个和连续传输分解

块大小不是单次突发事务长度的倍数，则当多次突发传输以后剩余的数据小于突发传输数据大小时，需要多次单个数据传输来完成整个块的传输。

23.3 I2C 中断

I2C 有 9 个可屏蔽中断源，这些独立的中断经过或逻辑生成 1 个中断输出，对应着系统中的 I2C 全局中断向量。

- RX_OVER接收溢出中断：当接收缓冲区的数据溢出时，该位置1
- TX_OVER发送溢出中断：当发送缓冲区的数据溢出时，该位置1
- RD_REQ从机读请求中断：当从机向主机发送读请求时，该位置1
- TX_ABRT 传输中止中断：当传输发生意外中止时，该位置 1
- RX_DONE 接收完成中断：当接收数据完成，接收到 stop 位时，该位置 1
- ACTIVITY I2C 工作中断：当 I2C 处于工作状态时，该位置 1
- STOP_DET STOP 检测中断：当检测到 STOP 时，该位置 1
- START_DET START 检测中断：当检测到 START 时，该位置 1
- GEN_CALL 广呼中断：当广呼响应时，该位置 1

中断状态寄存器 IC_INTR_STAT 的每一位都有对应的屏蔽位，可通过读相应的清除中断寄存器去清除相应中断。

23.4 脉冲滤波

这部分功能逻辑基于监视输入信号（SCL 和 SDA）的计数器，检查它们在内部采样之前是否在预定量的 I2C_CLK 周期内保持稳定。每个信号（SCL 和 SDA）都有一个单独的计数器。用户可以编程 I2C_CLK 周期的数量，并且应该考虑 I2C_CLK 的频率和相关的尖峰长度规范来计算 I2C_CLK 周期的数量。

只要输入信号改变其值，每个计数器就会启动。根据输入信号的行为，会出现以下情况之一：

- 输入信号保持不变，直到计数器达到其计数限值。当这个发生时，信号的内部值用输入值更新，计数器复位并停止。在检测到输入信号的新变化之前，计数器不会重新启动。
- 在计数器达到计数限值之前，输入信号再次发生变化。当这个情况发生时，计数器复位并停止，但信号的内部值不会更新。计数器保持停止直到检测到输入信号的新变化。

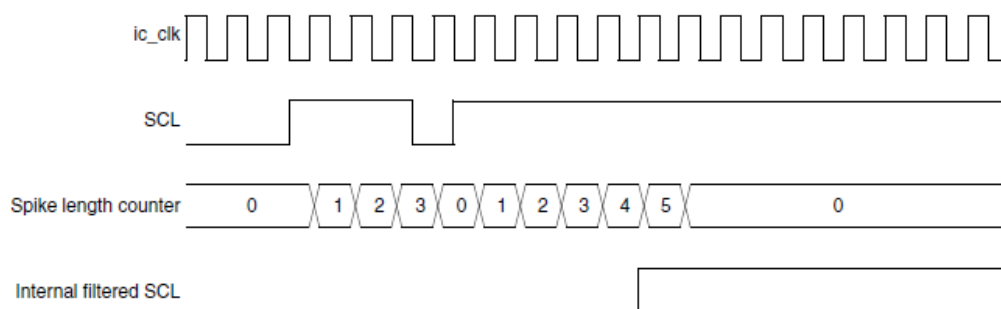


图 23-16 尖峰抑制示例图

图 23-16 中使用的计数限值为 5，在 10 ns 的 `ic_clk` 周期下，标准/快速模式的峰值抑制时间为 50 ns。I2C 总线规范根据工作模式要求不同的最大尖峰长度，对于标准模式和快速模式为 50 ns。寄存器 `IC_FS_SPKLEN` 保持标准模式和快速模式的最大尖峰长度，该寄存器为 8 位，可通过 APB 接口访问，以进行读写操作，但只有在 I2C 禁用时才能写入。`IC_FS_SPKLEN` 可编程的最小值为 1；尝试编程小于 1 的值会导致写入 1。这些寄存器的默认值是根据 `ic_clk` 周期的值自动计算出来的，但是这个值可以被用户在配置组件时覆盖。对于 `ic_clk` 的低频，可以超过峰值长度规范。

23.5 I2C 寄存器描述

表 23-3 I2C 相关寄存器表

名称	说明	读写权限	复位值	字节地址
<code>IC_CON</code>	控制寄存器	R/W	0x0000_007D	0x4001_3000
<code>IC_TAR</code>	I2C 目标地址寄存器	R/W	0x0000_1055	0x4001_3004
<code>IC_SAR</code>	I2C 从机地址寄存器	R/W	0x0000_0055	0x4001_3008
<code>IC_DATA_CMD</code>	I2C 数据命令寄存器	R/W	0x0000_0000	0x4001_3010
<code>IC_SS_SLC_HCNT</code>	I2C 标准模式时钟高字节计数器	R/W	0x0000_011E	0x4001_3014
<code>IC_SS_SLC_LCNT</code>	I2C 标准模式时钟低字节计数器	R/W	0x0000_0150	0x4001_3018
<code>IC_FS_SLC_HCNT</code>	I2C 快速模式时钟高字节计数器	R/W	0x0000_002B	0x4001_301C
<code>IC_FS_SLC_LCNT</code>	I2C 快速模式时钟低字节计数器	R/W	0x0000_005D	0x4001_3020
<code>IC_INTR_STAT</code>	中断状态寄存器	R/W	0x0000_0000	0x4001_302C
<code>IC_INTR_MASK</code>	中断屏蔽寄存器	R/W	0x0000_08FF	0x4001_3030
<code>IC_RAW_INTR_STAT</code>	中断原始状态寄存器	R/W	0x0000_0000	0x4001_3034
<code>IC_RX_TL</code>	接收 FIFO 阈值寄存器	R/W	0x0000_0000	0x4001_3038
<code>IC_TX_TL</code>	发送 FIFO 阈值寄存器	R/W	0x0000_0000	0x4001_303C
<code>IC_CLR_INTR</code>	中断清除寄存器	R/W	0x0000_0000	0x4001_3040
<code>IC_CLR_RX_UNDER</code>	清除 <code>RX_UNDER</code> 中断寄存器	R/W	0x0000_0000	0x4001_3044
<code>IC_CLR_RX_OVER</code>	清除 <code>RX_OVER</code> 中断寄存器	R/W	0x0000_0000	0x4001_3048
<code>IC_CLR_TX_OVER</code>	清除 <code>TX_OVER</code> 中断寄存器	R/W	0x0000_0000	0x4001_304C
<code>IC_CLR_RD_REQ</code>	清除 <code>RD_REQ</code> 中断寄存器	R/W	0x0000_0000	0x4001_3050
<code>IC_CLR_TX_ABRT</code>	清除 <code>TX_ABRT</code> 中断寄存器	R/W	0x0000_0000	0x4001_3054
<code>IC_CLR_RX_DONE</code>	清除 <code>RX_DONE</code> 中断寄存器	R/W	0x0000_0000	0x4001_3058
<code>IC_CLR_ACTIVITY</code>	清除 <code>ACTIVITY</code> 中断寄存器	R/W	0x0000_0000	0x4001_305C
<code>IC_CLR_STOP_DET</code>	清除 <code>STOP_DET</code> 中断寄存器	R/W	0x0000_0000	0x4001_3060
<code>IC_CLR_START_DET</code>	清除 <code>START_DET</code> 中断寄存器	R/W	0x0000_0000	0x4001_3064

CLR_GEN_CALL	清除 GEN_CALL 中断寄存器	R/W	0x0000_0000	0x4001_3068
IC_ENABLE	I2C 使能寄存器	R/W	0x0000_0000	0x4001_306C
IC_STATUS	I2C 状态寄存器	R/W	0x0000_0000	0x4001_3070
IC_TXFLR	I2C 发送 FIFO 数据量寄存器	R/W	0x0000_0000	0x4001_3074
IC_RXFLR	I2C 接收 FIFO 数据量寄存器	R/W	0x0000_0000	0x4001_3078
IC_SDA_HOLD	SDA 维持时间长度寄存器	R/W	0x0000_0000	0x4001_307C
IC_TX_ABRT_SOURCE	I2C 发送丢弃源寄存器	R/W	0x0000_0000	0x4001_3080
IC_SLV_DATA_NACK_ON	I2C 生成从机数据 NACK 寄存器	R/W	0x0000_0000	0x4001_3084
IC_DMA_CR	DMA 控制寄存器	R/W	0x0000_0000	0x4001_3088
IC_DMA_TDLR	DMA 发送数据水平寄存器	R/W	0x0000_0000	0x4001_308C
IC_DMA_RDLR	DMA 接收数据水平寄存器	R/W	0x0000_0000	0x4001_3090
IC_SDA_SETUP	SDA 建立寄存器	R/W	0x0000_0064	0x4001_3094
IC_ACK_GENERAL_CALL	广呼应答寄存器	R/W	0x0000_0001	0x4001_3098
IC_ENABLE_STATUS	I2C 使能状态寄存器	R/W	0x0000_0000	0x4001_309C
IC_FS_SPKLEN	标准模式和快速模式尖峰抑制寄存器	R/W	0x0000_0004	0x4001_30A0
IC_CLR_RESTAR_DET	清除 RESTAR_DET 寄存器	R/W	0x0000_0000	0x4001_30A8

注：x 表示不确定；-表示未实现；u 表示不受复位影响；c 表示该位复位值因复位源而异；R 只读；W 只写；R-0 表示只读且读 0；R/W 表示可读写（以后章节同上述）。

23.5.1 I2C 控制寄存器（IC_CON）

地址偏移：0x00

复位值：0x0000_007D

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					SDIMA	RFFHC	TXEC	SDIAD	SLVDIS	RESTEN	10ADDR_MAS	10ADDR_SLV	SPEED [1:0]	MST_MODE	
					rw	rw	rw	rw	rw	rw	r	rw	rw	rw	

Bits	31:11	保留，必须保持复位值
Bit	10	SDIMA : 主机模式下的 STOP_DET 中断
		0: STOP_DET 中断与主机是否有效无关（默认）
		1: 仅当主机有效时产生 STOP_DET 中断
Bit	9	RFFHC : 决定 I2C 在 RX FIFO 达到 RX_RX_BUFFER_DEPTH 时是否占有总线
		0: 不控制总线（默认）
		1: 控制总线
Bit	8	TXEC : 控制 TX_EMPTY 中断的产生方式

	具体请参考 IC_RAW_INTR_STAT 寄存器
Bit 7	SDIAD: 从机模式下产生 STOP_DET 中断条件
	0: 无论是否寻址匹配均产生 STOP_DET 中断
	1: 当寻址匹配时产生 STOP_DET 中断 (默认)
Bit 6	SLVDIS: 从机关闭控制
	0: 从机使能
	1: 从机禁止 (默认)
Bit 5	RESTEN: 控制主机是否能够发送“重新启动 (Restart)”信号
	0: 禁止
	1: 使能 (默认)
	当 RESTEN=0, 主机不支持以下功能:
	<ul style="list-style-type: none"> ● 发送起始字节 ● 高速模式 ● 在合并格式下不支持方向改变 ● 进行10位地址的读操作 ● 重启操作 (Restart) 将由 STOP 和 START 组合替换
Bit 4	10ADDR_MAS: 只读寄存器, 与 IC_TAR[12]位的状态相同
	0: 7 位地址
	1: 10 位地址 (默认)
Bit 3	10ADDR_SLV: 作为从机时, 定义地址的位宽
	0: 7 位地址, 忽略 10 位地址的传输。7 位地址的传输只比较 IC_SAR 寄存器中的低 7 位
	1: 10 位地址, 从机只回应 10 位地址匹配的传输 (默认)
Bits 2:1	SPEED[1:0]: I2C 模式选择
	01: 标准模式 (0~100Kb/s)
	10: 快速模式 (400Kb/s) 或超快速模式 (1000Kb/s) (默认)
	其他: 保留
Bit 0	MST_MODE: 主机模式使能
	0: 主机模式禁止
	1: 主机模式使能 (默认)

IC_CON[6]	IC_CON[0]	I2C 模式
0	0	从机
0	1	错误
1	0	错误
1	1	主机

23.5.2 I2C 目标地址寄存器 (IC_TAR)

地址偏移: 0x04

复位值: 0000_1055

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			10ADDR _MAS	SPECIAL	GC_ STR	IC_TAR[9:0]									
			rw	rw	rw	rw									

Bits	31:13	保留，必须保持复位值
Bit	12	10ADDR_MAS: 当 I2C 为主机时，决定发送的从机地址位宽
		0: 7 位地址（默认）
		1: 10 位地址
Bit	11	SPECIAL: 决定 General Call 命令还是 START BYTE 命令
		0: 忽略 GC_STR，正常使用 IC_TAR（默认）
		1: 执行 GC_STR 命令
Bit	10	GC_STR: 若 SPECIAL 置 1，该位有效
		0: General Call 模式，只执行发出的写命令，若发出读命令将置位 TX_ABRT
		1: START BYTE（默认）
Bits	9:0	IC_TAR[9:0]: I2C 主机传输的目标地址，执行 General Call 时忽略。生成 START BYTE，只需配置一次
注: 若 I2C 只作为从机使用，该寄存器无需配置，地址不能为 0x00~0x07。		

23.5.3 I2C 从机地址寄存器 (IC_SAR)

地址偏移: 0x08

复位值: 0x0000_0055

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

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

Bits	31:10	保留，必须保持复位值
Bits	9:0	IC_SAR[9:0]: 是 I2C 为从机时存储的从机地址。对于 7 位地址，仅使用 IC_TAR[6:0]，即低位的 7 位地址为有效地址
注: 若 I2C 只作为主机使用，该寄存器无需配置，该寄存器仅在 I2C 禁止时配置有效。		

23.5.4 I2C 数据命令寄存器 (IC_DATA_CMD)

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
Res.															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				FDB	RESTART	STOP	CMD	DAT[7:0]							
				rw	rw	rw	rw	rw							

Bits 31:12	保留，必须保持复位值
Bit 11	FDB (FIRST_DATA_BYTE) : 在主机接收或从机接收模式下收到地址后的第一个数据 该位仅在 FIRST_DATA_BYTE_STATUS=1 时，寄存器中值有效
Bit 10	RESTART : 该位决定在数据发送或接收前是否发送 RESTART 0: 如果 IC_RESTART_EN 为 1，则仅在传输时从上个命令发生改变时发送 RESTART。如果 IC_RESTART_EN 为 0，则发出 STOP 后跟 START (默认) 1: 如果 IC_RESTART_EN 为 1，无论传输方向是不是从上一个命令改变，在数据发送/接收之前发出 RESTART (由 CMD 位决定)
Bit 9	STOP : 该位控制是否在数据发送或接收后发出 STOP 0: 不管发送 FIFO 是否为空，在该字节后 STOP 不发送。如果发送 FIFO 非空，主机根据 CMD 位确定继续发送/接收数据。如果发送 FIFO 为空，主机维持 SCL 为低，停止总线直到发送 FIFO 有新的命令可用 (默认) 1: 不管发送 FIFO 是否为空，在该字节后发出 STOP。如果发送 FIFO 非空，立即通过起始位仲裁总线，用于发起新的数据传输
Bit 8	CMD : 该位控制读/写操作的执行。当 I2C 作为从机时不控制数据传输的方向，仅作为主机时有效 0: 写 (默认) 1: 读 当命令进入发送 FIFO 时，该位识别命令的读写操作。从机接收模式，无需写该寄存器，不关心该位。从机发送模式，0 指示 IC_DATA_CMD 内的数据将被传送。当作为主机操作该位时，应注意以下方面：在广播命令发送后试图发起读操作，将触发 TX_ABRT 中断，除非 IC_TAR 寄存器的 bit 11 被清除；如果在收到 RD_REQ 中断后在该位写入 1，TX_ABRT 中断发生。
Bits 7:0	DAT[7:0] : 该寄存器包含了从总线接收或往总线发送的数据。如果向该寄存器写入数据并执行读操作，I2C 忽略 DAT。然而，当读该寄存器时，DAT 返回 I2C 从总线接收到的数据

23.5.5 I2C 标准模式时钟高字节计数器 (IC_SS_SCL_HCNT)

地址偏移: 0x14

复位值: 0000_011E

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

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

Bits 31:16	保留，必须保持复位值
Bits 15:0	SS_SCL_HCNT[15:0] : 该寄存器必须在 I2C 进行传输前完成配置 (IC_ENABLE[0] = 0)
	SS_SCL_HCNT 最小值为 6，当用户写入的值小于 6 时，硬件写入为 6
	SS_SCL_HCNT 最大值为 65525

23.5.6 I2C 标准模式时钟低字节计数器 (IC_SS_SCL_LCNT)

地址偏移: 0x18

复位值: 0x0000_0150

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

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

Bits 31:16	保留，必须保持复位值
Bits 15:0	SS_SCL_LCNT[15:0] : 该寄存器必须在 I2C 进行传输前完成配置 (IC_ENABLE[0] = 0)
	SS_SCL_LCNT 最小值为 8，当用户写入的值小于 8 时，硬件仍写入 8

25.5.7 I2C 快速模式时钟高字节计数器 (IC_FS_SCL_HCNT)

地址偏移: 0x1C

复位值: 0x0000_002B

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

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

Bits 31:16	保留，必须保持复位值
Bits 15:0	FS_SCL_HCNT[15:0] : 该寄存器必须在 I2C 进行传输前完成配置 (IC_ENABLE[0] = 0)
	FS_SCL_HCNT 最小值为 6，当用户写入的值小于 6 时，硬件写入为 6

23.5.8 I2C 快速模式时钟低字节计数器 (IC_FS_SCL_LCNT)

地址偏移: 0x20

复位值: 0x0000_005D

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

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

Bits 31:16	保留，必须保持复位值
Bits 15:0	FS_SCL_LCNT[15:0] : 该寄存器必须在 I2C 进行传输前完成配置 (IC_ENABLE[0] = 0)
	FS_SCL_HCNT 最小值为 8，当用户写入的值小于 8 时，硬件写入为 8

23.5.9 I2C 中断状态寄存器 (IC_INTR_STAT)

地址偏移: 0x2C

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		R_MST_ON	R_RESTAR	R_GEN	R_STAR	R_STOP	R_	R_RX_	R_TX_	R_RD	R_TX_	R_TX_	R_RX_	R_RX_	R_RX_
		_HOLD	T_DET	_CALL	T_DET	_DET	ACTIVITY	DONE	ABRT	_REQ	EMPTY	OVER	FULL	OVER	UNDER
		r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits 31:14	保留，必须保持复位值
Bit 13	R_MST_ON_HOLD : 详细描述参见 IC_RAW_INTR_STAT
Bit 12	R_RESTART_DET : 详细描述参见 IC_RAW_INTR_STAT
Bit 11	R_GEN_CALL : 详细描述参见 IC_RAW_INTR_STAT
Bit 10	R_START_DET : 详细描述参见 IC_RAW_INTR_STAT
Bit 9	R_STOP_DET : 详细描述参见 IC_RAW_INTR_STAT
Bit 8	R_ACTIVITY : 详细描述参见 IC_RAW_INTR_STAT
Bit 7	R_RX_DONE : 详细描述参见 IC_RAW_INTR_STAT
Bit 6	R_TX_ABRT : 详细描述参见 IC_RAW_INTR_STAT
Bit 5	R_RD_REQ : 详细描述参见 IC_RAW_INTR_STAT
Bit 4	R_TX_EMPTY : 详细描述参见 IC_RAW_INTR_STAT
Bit 3	R_TX_OVER : 详细描述参见 IC_RAW_INTR_STAT
Bit 2	R_RX_FULL : 详细描述参见 IC_RAW_INTR_STAT
Bit 1	R_RX_OVER : 详细描述参见 IC_RAW_INTR_STAT
Bit 0	R_RX_UNDER : 详细描述参见 IC_RAW_INTR_STAT

注: 寄存器的每一位都有相应的 MASK 位 (参见寄存器 IC_INTR_MASK)。通过读对应的中断清寄存器清除这些位。

23.5.10 I2C 中断屏蔽寄存器 (IC_INTR_MASK)

地址偏移: 0x30

复位值: 0x0000_08FF

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		M_MST_O N_HOLD	M_RESTAR T_DET	M_GEN _CALL	M_STAR T_DET	M_STO P_DET	M_ ACTIVITY	M_RX_ DONE	M_TX_ ABRT	M_RD _REQ	M_TX_ EMPTY	M_TX_ OVER	M_RX_ _FULL	M_RX_ OVER	M_RX_ UNDER
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:14	保留，必须保持复位值
Bit 13	M_MST_ON_HOLD: 详细描述参见 IC_RAW_INTR_STAT
Bit 12	M_RESTART_DET: 详细描述参见 IC_RAW_INTR_STAT
Bit 11	M_GEN_CALL: 详细描述参见 IC_RAW_INTR_STAT
Bit 10	M_START_DET: 详细描述参见 IC_RAW_INTR_STAT
Bit 9	M_STOP_DET: 详细描述参见 IC_RAW_INTR_STAT
Bit 8	M_ACTIVITY: 详细描述参见 IC_RAW_INTR_STAT
Bit 7	M_RX_DONE: 详细描述参见 IC_RAW_INTR_STAT
Bit 6	M_TX_ABRT: 详细描述参见 IC_RAW_INTR_STAT
Bit 5	M_RD_REQ: 详细描述参见 IC_RAW_INTR_STAT
Bit 4	M_TX_EMPTY: 详细描述参见 IC_RAW_INTR_STAT
Bit 3	M_TX_OVER: 详细描述参见 IC_RAW_INTR_STAT
Bit 2	M_RX_FULL: 详细描述参见 IC_RAW_INTR_STAT
Bit 1	M_RX_OVER: 详细描述参见 IC_RAW_INTR_STAT
Bit 0	M_RX_UNDER: 详细描述参见 IC_RAW_INTR_STAT

23.5.11 I2C 中断原始状态寄存器 (IC_RAW_INTR_STAT)

地址偏移: 0x34

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		MST_ON _HOLD	RESTART _DET	GEN _CALL	START _DET	STOP _DET	ACTIVITY	RX_ DONE	TX_ ABRT	RD_ REQ	TX_ EMPTY	TX_ OVER	RX_ FULL	RX_ OVER	RX_ UNDER
		r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits 31:14	保留，必须保持复位值
Bit 13	MST_ON_HOLD: 该位指示主机是否占有总线，且 Tx FIFO 为空
Bit 12	RESTART_DET: 指示在从机模式且已被寻址的情况下，检测到一个 RESTART 条件
Bit 11	GEN_CALL: 当 I2C 收到且识别到 General Call 地址，该位置 1 当 I2C 禁止或 CPU 读操作 IC_CLR_GEN_CALL 的 bit 0 时，该位清 0
Bit 10	START_DET: 无论 I2C 作为主机还是从机，当收到 START 或 RESTART 位时该位置 1
Bit 9	STOP_DET: 无论 I2C 作为主机还是从机，当收到 STOP 条件时该位置 1
	从机模式： 如果 IC_CON[7] = 1 (STOP_DET_IF_ADDRESSED)，STOP_DET 中断只发生在从机地址匹配后。注意：一个广播地址，当 STOP_DET_IF_ADDRESSED = 1 时，即使从机返回

	ACK，也不产生 STOP_DET 中断。STOP_DET 中断仅发生在收到的地址和 I2C 从机地址（SAR）匹配的情况。 如果 IC_CON[7] = 0，无论寻址匹配与否，产生 STOP_DET 中断。
	主机模式：
	如果 IC_CON[10] = 1，产生 STOP_DET 中断仅在主机模式工作状态下有效。 如果 IC_CON[10] = 0，产生 STOP_DET 中断无论是否在主机模式在工作状态下。
Bit 8	ACTIVITY ：表示 I2C 模块是否在总线上有操作，在清除之前会一直维持。可通过以下 4 种方式清除该位：
	1. I2C 禁止
	2. 读 IC_CLR_ACTIVITY 寄存器
	3. 读 IC_CLR_INTR 寄存器
	4. 系统复位
Bit 7	RX_DONE ：I2C 作为从机发送时，如果主机不反馈已传送数据，该位置 1。这发生在数据传送的最后一个字节，指示传输完毕
Bit 6	TX_ABRT ：I2C 作为发送器，当无法发送 FIFO 内的数据时，该位置 1。在从机和主机模式下都可能发生。当置为 1 时，读 IC_TX_ABRT_SOURCE 寄存器，可知发送失败的原因。 <i>注：当 TX_ABRT 发生时，清空 TX_FIFO。Tx_FIFO 将维持清空状态直到完成 IC_CLR_TX_ABRT 读操作。读操作一旦完成，Tx_FIFO 可接收更多的数据。</i>
Bit 5	RD_REQ ：I2C 作为从机，当另一主机试图从该从机读取数据时，该位置 1。该 I2C 占用总线（SCL=0）直到中断服务程序执行完毕。即主机请求从机发送数据。CPU 必须响应该中断，并在 IC_DATA_CMD 寄存器中写入待发送数据。该位在读 IC_CLR_RD_REQ 后清 0。
Bit 4	TX_EMPTY ：该中断状态的行为取决于 IC_CON 寄存器的 TX_EMPTY_CTRL 当 TX_EMPTY_CTRL = 0，当发送 buffer 不高于 IC_TX_TL 所设定的阈值时，该位置 1。 当 TX_EMPTY_CTRL = 1，当发送 buffer 不高于 IC_TX_TL 所设定的阈值且内部移位寄存器中的地址或数据发送完毕，该位置 1。当 buffer 超过阈值时，该位硬件自动清 0。
Bit 3	TX_OVER ：I2C 工作在发送模式，但 TX_FIFO 已经满的情况下，软件还向 IC_DATA_CMD 写入命令，该位置 1。在从机和主机模式下都可能发生。
Bit 2	RX_FULL ：I2C 的接收 buffer 超过 IC_RX_TL 所设定的阈值时，该位置为 1。当 buffer 低于阈值时，该位自动清 0。一旦 IC_ENABLE[0]置 0，该位清 0。
Bit 1	RX_OVER ：I2C 的接收 buffer 超过 RX 缓存的深度且从外部 I2C 收到额外的字节，该位置 1。I2C 对此做出响应，但接收的后续字节都被丢弃。一旦 IC_ENABLE[0]置 0，该位清 0。
Bit 0	RX_UNDER ：当接收 buffer 为空时，CPU 通过 IC_DATA_CMD 寄存器读 buffer，该位置 1。一旦 IC_ENABLE[0]置 0，该位清 0。
	<i>注：与 IC_INTR_STAT 寄存器不同，该寄存器不受 Mask 屏蔽的影响，真实反映各个中断的状态信息。</i>

23.5.12 I2C 接收 FIFO 阈值寄存器（IC_RX_TL）

地址偏移：0x38

复位值：0x0000_0000

1	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															

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

Bits	31:3	保留，必须保持复位值
Bits	2:0	RX_TL[2:0] : 接收 FIFO 阈值
		控制 RX_FULL 中断触发的阈值设定，范围 0~7。

23.5.13 I2C 发送 FIFO 阈值寄存器 (IC_TX_TL)

地址偏移: 0x3C

复位值: 0x0000_0000

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

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

Bits	31:3	保留，必须保持复位值
Bits	2:0	TX_TL[2:0] : 发送 FIFO 阈值
		控制 TX_EMPTY 中断触发的阈值设定，范围 0~7。

23.5.14 I2C 中断清除寄存器 (IC_CLR_INTR)

地址偏移: 0x40

复位值: 0x0000_0000

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

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

Bits	31:1	保留，必须保持复位值
Bit	0	CLR_INTR : 读该寄存器清除组合中断，所有独立中断和 IC_TX_ABRT_SOURCE 寄存器
		注: 该位只能清除软件可清除的中断，硬件清除中断该位不能清除，具体参阅 TX_ABRT_SOURCE 寄存器的 bit 9。

23.5.15 I2C 清除 RX_UNDER 中断寄存器 (IC_CLR_RX_UNDER)

地址偏移: 0x44

复位值: 0x0000_0000

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

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

Bits	31:1	保留, 必须保持复位值
Bit	0	CLR_RX_UNDER: 读该寄存器清除 RX_UNDER 中断 (IC_RAW_INTR_STAT 寄存器的 bit 0)

23.5.16 I2C 清除 RX_OVER 中断寄存器 (IC_CLR_RX_OVER)

地址偏移: 0x48

复位值: 0x0000_0000

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

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

Bits	31:1	保留, 必须保持复位值
Bit	0	CLR_RX_OVER: 读该寄存器清除 RX_OVER 中断 (IC_RAW_INTR_STAT 寄存器的 bit 1)

23.5.17 I2C 清除 TX_OVER 中断寄存器 (IC_CLR_TX_OVER)

地址偏移: 0x4C

复位值: 0x0000_0000

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

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

Bits	31:1	保留, 必须保持复位值
Bit	0	CLR_TX_OVER: 读该寄存器清除 TX_OVER 中断 (IC_RAW_INTR_STAT 寄存器的 bit

Res.															

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

Bits	31:1	保留，必须保持复位值
Bit	0	CLR_START_DET: 读该寄存器清除 START_DET 中断 (IC_RAW_INTR_STAT 的 bit 10)

23.5.24 I2C 清除 GEN_CALL 中断寄存器 (CLR_GEN_CALL)

地址偏移: 0x68
 复位值: 0x0000_0000

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

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

Bits	31:1	保留，必须保持复位值
Bit	0	CLR_GEN_CALL: 读该寄存器清除 GEN_CALL 中断 (IC_RAW_INTR_STAT 的 bit 11)

23.5.25 I2C 使能寄存器 (IC_ENABLE)

地址偏移: 0x6C
 复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.													TX_CMD_BLOCK	ABORT	ENABLE
													rw	rw	rw

Bits	31:3	保留，必须保持复位值
Bit	2	TX_CMD_BLOCK: 主机模式有效
		0: 一旦 TX FIFO 有可用的数据, I2C 启动发送 (默认)
		1: 阻止 I2C 发送数据, 即使 TX FIFO 有待发数据
		<i>注:</i> 为阻止主机命令的执行, 只能在 Tx FIFO 为空 (IC_STATUS[2]=1) 且主机处于 IDLE 状态 (IC_STATUS[5]=0) 时置位 TX_CMD_BLOCK。一旦该位置 1, 任何 Tx FIFO 命令都无效。
Bit	1	ABORT: 丢弃 I2C 发送操作
		0: ABORT 操作尚未发起或已经执行完毕 (默认)

	1: ABORT 正在执行
	在主机模式，软件可配置该位来丢弃 I2C 发送操作。ABORT 仅在 ENABLE 为 1 时置位有效。ABORT 一旦置位，软件无法清除。为响应 ABORT 命令，I2C 发出 STOP 且在完成当前数据的传送后清空 Tx FIFO，完成丢弃操作后置位 TX_ABORT 中断。ABORT 位在完成丢弃操作后自动清 0。
Bit 0	ENABLE: I2C 使能位
	0: I2C 禁止（默认）
	1: I2C 使能
	如果 I2C 当前正在发送，待发送完毕后停止并删除发送 buffer 里的内容。如果 I2C 当前正在接收，待接收完毕后停止，并不对当前传送回 ACK。

23.5.26 I2C 状态寄存器（IC_STATUS）

地址偏移：0x70

复位值：0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					SLV_	SLV_	MST_	MST_	SLV_	MST_	RFF	RFNE	TFE	TFNF	ACTIVITY
					HOLD_RX_	HOLD_TX_	HOLD_RX_	HOLD_TX_	ACTIVITY	ACTIVITY					
					FIFO_FULL	FIFO_EMPTY	FIFO_FULL	FIFO_EMPTY							
					r	r	r	r	r	r	r	r	r	r	r

Bits 31:11	保留，必须保持复位值
Bit 10	SLV_HOLD_RX_FIFO_FULL: 表明由于当前 Rx FIFO 满且收到额外字节，I2C 总线被从机控制
Bit 9	SLV_HOLD_TX_FIFO_EMPTY: 表明当前 Tx FIFO 空且收到读请求，I2C 总线被从机控制
Bit 8	MST_HOLD_RX_FIFO_FULL: 表明当前 Rx FIFO 满且收到额外字节，总线被主机控制
Bit 7	MST_HOLD_TX_FIFO_EMPTY: 表明由于当前 Tx FIFO 空且当前传送命令没有停止位，总线被主机接管
Bit 6	SLV_ACTIVITY: 从机状态机活动状态，当状态机不在 IDLE 状态，该位置 1
	0: 从机状态机处于 IDLE 态，从机不活跃（默认）
	1: 从机状态机处于非 IDLE 态，从机活跃
Bit 5	MST_ACTIVITY: 主机状态机活动状态，当状态机不在 IDLE 状态，该位置 1
	0: 主机状态机处于 IDLE 态，主机不活跃（默认）
	1: 主机状态机处于非 IDLE 态，主机活跃
Bit 4	RFF: 接收 FIFO 全满时，该位置 1。当接收 FIFO 含一个或多个空槽时，该位清 0。
	0: 接收 FIFO 非全满（默认）
	1: 接收 FIFO 全满
Bit 3	RFNF: 接收 FIFO 非空。当接收 FIFO 含一个或多个数据时，该位置 1。当接收 FIFO 为空时，该位清 0

	0: 接收 FIFO 空 (默认)
	1: 接收 FIFO 非空
Bit 2	TFE : 发送 FIFO 空。当发送 FIFO 含一个或多个数据时, 该位清 0。该位不产生中断请求
	0: 发送 FIFO 非空 (默认)
	1: 发送 FIFO 空
Bit 1	TFNF : 发送 FIFO 非满。当发送 FIFO 含一个或多个空槽时, 该位置 1。当 FIFO 满时该位清 0
	0: 发送 FIFO 满 (默认)
	1: 发送 FIFO 非满
Bit 0	ACTIVITY : I2C 活跃状态
	0: 不活跃 (默认)
	1: 活跃

23.5.27 I2C 发送 FIFO 数据量寄存器 (IC_TXFLR)

地址偏移: 0x74

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.												TXFLR[3:0]			
												r			

Bits 31:4	保留, 必须保持复位值
Bits 3:0	TXFLR[3:0] : 发送 FIFO 内有效数据的个数。每次把数据放进发送 FIFO, 该寄存器加 1; 每次数据从发送 FIFO 拿出, 该寄存器减 1。
	该寄存器包含发送 FIFO 有效数据的个数, 以下条件可清除该寄存器: <ul style="list-style-type: none"> ● I2C 禁止 ● 发生一个发送丢弃, IC_RAW_INTR_STAT 寄存器的 TX_ABRT 置位 ● 从机发送模式丢弃

23.5.28 I2C 接收 FIFO 数据量寄存器 (IC_RXFLR)

地址偏移: 0x78

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.												RXFLR[3:0]			
												r			

Bits 31:4	保留，必须保持复位值
Bits 3:0	RXFLR[3:0] : 接收 FIFO 内包含有效数据的个数。每次把数据放进接收 FIFO，该寄存器加 1。每次数据从接收 FIFO 拿出，该寄存器减 1。
	该寄存器包含接收 FIFO 有效数据的个数，以下条件可清除该寄存器： <ul style="list-style-type: none"> ● I2C 禁止 ● 发生一个发送丢弃，由 IC_TX_ABRT_SOURCE 的任何事件引起

23.5.29 I2C SDA 维持时间长度寄存器 (IC_SDA_HOLD)

地址偏移: 0x7C

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.								IC_SDA_RX_HOLD[7:0]							
rw															

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

Bits 31:24	保留，必须保持复位值
Bits 23:16	IC_SDA_RX_HOLD[7:0] : 当 I2C 作为接收器时，设置 SDA 维持时间的周期（以 I2C_CLK 为周期）
	用于在主机或从机接收情况下 SCL 为高时，拓展 SDA 时间
Bits 15:0	IC_SDA_TX_HOLD[15:0] : 当 I2C 作为发送器时，设置 SDA 维持时间的周期（以 I2C_CLK 为周期）
	用于控制从机或主机发送时 SDA 的维持时间（在 SCL 从高变为低之后）
注1: 该寄存器仅在 IC_ENABLE[0] 为 0 时写成功。	
注2: 寄存器的值以 I2C_CLK 为计数单位。IC_SDA_TX_HOLD 内的值必须超过每个模式下维持时间的最小值：主机模式 1 个周期，从机模式下 7 个周期。	

23.5.30 I2C 发送丢弃源寄存器 (IC_TX_ABRT_SOURCE)

地址偏移: 0x80

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TX_FLUSH_CNT[8:0]								Res.							ABRT_USER_
r															ABRT
r															r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRT_	ABRT_	ABRT_	ARB_	ABRT_	ABRT_	ABRT_	Re	ABRT_	Re	ABRT_	ABRT_	ABRT_	ABRT_	ABRT_	ABRT_
SLVRD	_SLV_	SLVFLUSH	ARB_	MASTER	10B_RD_	SBYTE_	s.	SBYTE_	s.	GCALL	GCALL	TXDATA	10ADDR2	10ADDR1	7ADDR
_INTX	ARBLOST	_TXFIFO	LOST	_DIS	NORSTR	NORSTR		ACKDET		_READ	_NOACK	_NOACK	_NOACK	_NOACK	_NOACK
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits 31:23	TX_FLUSH_CNT[8:0]: 当 TX_ABORT 中断发生时, Tx FIFO 被 flush 的数据个数。当 I2C 禁止时, 该位清 0。 (主机发送或从机发送)
Bits 22:17	保留, 必须保持复位值
Bit 16	ABRT_USER_ABORT: 该位只在主机模式下使用 (主机发送) 0: 主机未侦测到发送丢失 (默认) 1: 主机侦测到发送丢失
Bit 15	ABRT_SLVRD_INTX: 从机写入数据操作 (从机发送) 0: 无从机写入数据操作 (默认) 1: 用户在 IC_DATA_CMD 寄存器 CMD 位 (bit 8) 写入 1, CPU 响应 I2C 从机写入数据的请求
Bit 14	ABRT_SLV_ARBLOST: 从机发送数据状态 (从机发送) 0: I2C 从机发送数据正常 (默认) 1: I2C 从机往外发送数据时丢失总线, 同时 IC_TX_ABORT_SOURCE[12]置 1
Bit 13	ABRT_SLVFLUSH_TXFIFO: 从机接受数据状态 (从机发送) 0: 从机接受数据正常 (默认) 1: I2C 从机接收到读请求, 但 TX FIFO 中有数据未发送, 从机产生 TX_ABORT 中断来 flush TX FIFO 内的数据
Bit 12	ARB_LOST: 丢失仲裁 (主机发送或从机发送) 0: 主机/从机未丢失仲裁 (默认) 1: I2C 主机丢失仲裁; 如果 IC_TX_ABORT_SOURCE[14]置位, 从机丢失仲裁
Bit 11	ABRT_MASTER_DIS: 主机禁止模式下的主机操作 (主机发送或主机接收) 0: 主机禁止模式下未发生主机操作 (默认) 1: 用户试图在主机模式禁止的情况下发起主机操作
Bit 10	ABRT_10B_RD_NORSTRT: Restart 主机发 10 位地址读命令 (主机接收) 0: Restart 主机可以发 10 位地址的读命令 (默认) 1: Restart 禁止 (IC_RESTART_EN = 0) 主机发 10 位地址的读命令
Bit 9	ABRT_SBYTE_NORSTRT: Restart 主机发送 START Byte 0: Restart 主机可以发送 START Byte (默认) 1: Restart 禁止 (IC_RESTART_EN = 0) 并且用户正在尝试发送 START Byte 清除第 9 位, 首先必须固定 ABRT_SBYTE_NORSTRT 的来源; 重启必须使能 (IC_CON[5]=1), 必须清除 SPECIAL 位 (IC_TAR [11]) 或必须清零 GC_OR_START 位 (IC_TAR[10])。一旦 ABRT_SBYTE_NORSTRT 来源是固定的, 该位的清除方式与该寄存器的其它位一致。 如果在未固定 ABRT_SBYTE_NORSTRT 来源前试图清除这一位, 该位清除一个周期, 然后重新置位。
Bit 8	保留, 必须保持复位值
Bit 7	ABRT_SBYTE_ACKDET: 主机发送起始字节 (主机) 0: 主机未发送起始字节或起始字节无应答 (默认) 1: 主机发送一个起始字节且起始字节被应答
Bit 6	保留, 必须保持复位值
Bit 5	ABRT_GCALL_READ: General Call 操作的情况 (主机发送) 0: General Call 操作和 IC_DATA_CMD[9]设置不冲突 (默认)

	1: 主机在主机模式发送一个 General Call, 但是用户编程的关于 General Call 的字节是从总线读操作 (IC_DATA_CMD[9]设置为 1)
Bit 4	ABRT_GCALL_NOACK: General Call 的响应情况 (主机发送)
	0: 主机未发送 General Call 或者 General Call 有从机响应 (默认)
	1: I2C 在主机模式发送 General Call, 但总线上没有从机响应
Bit 3	ABRT_TXDATA_NOACK: 主机发送模式的响应 (主机发送)
	0: 主机发送模式正常 (默认)
	1: 仅支持主机模式。主机发送地址后收到应答, 但紧跟着发送的数据收不到从机的应答
Bit 2	ABRT_10ADDR2_NOACK: 主机 10 位地址的第二个字节的响应 (主机发送或主机接收)
	0: 在 10 位地址模式, 主机 10 位地址的第二个字节有从机响应。(默认)
	1: 在 10 位地址模式, 主机 10 位地址的第二个字节没有任何从机回应
Bit 1	ABRT_10ADDR1_NOACK: 主机 10 位地址的第一个字节的响应 (主机发送或主机接收)
	0: 在 10 位地址模式, 主机 10 位地址的第一个字节有从机回应 (默认)
	1: 在 10 位地址模式, 主机 10 位地址的第一个字节没有任何从机回应
Bit 0	ABRT_7ADDR_NOACK: 主机 7 位地址的响应 (主机发送或主机接收)
	0: 在 7 位地址模式, 主机地址有从机回应 (默认)
	1: 在 7 位地址模式, 主机地址没有任何从机回应

23.5.31 I2C 生成从机数据 NACK 寄存器 (IC_SLV_DATA_NACK_ONLY)

地址偏移: 0x84

复位值: 0x0000_0000

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

Bits 31:1	保留, 必须保持复位值
Bit 0	NACK: 只有在 I2C 作为从机使用时才能产生 NACK
	如果该位置 1, 在收到数据后产生 NACK。因此, 数据传输完后丢弃, 收到的数据也不放到接收 buffer 中。当该位置为 0, 产生 NACK/ACK 取决于正常标准。
	0: 正常产生 NACK/ACK (默认)
	1: 在收到数据后产生 NACK
	如果以下条件同时满足, 可以发生写该寄存器操作:
	<ul style="list-style-type: none"> ● I2C 禁止 (IC_ENABLE[0] = 0) ● 从机部分不活跃 (IC_STATUS[6] = 0)

23.5.32 I2C DMA 控制寄存器 (IC_DMA_CR)

地址偏移: 0x88

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.														TDMAE	RDMAE
														rw	rw

Bits	31:2	保留, 必须保持复位值
Bit	1	TDMAE : 发送 DMA 使能
		0: 发送 DMA 禁止 (默认)
		1: 发送 DMA 使能
Bit	0	RDMAE : 接收 DMA 使能
		0: 接收 DMA 禁止 (默认)
		1: 接收 DMA 使能

23.5.33 I2C DMA 发送数据水平寄存器 (IC_DMA_TDLR)

地址偏移: 0x8C

复位值: 0x0000_0000

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

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

Bits	31:3	保留, 必须保持复位值
Bits	2:0	DMATDL[2:0] : 发送数据水平。这些位控制发送逻辑何时产生一个 DMA 请求。
		当发送 FIFO 内有效数据的个数小于或等于 DMATDL 且 TDMAE=1 时, 产生一个 dma_tx_req 信号。

23.5.34 I2C DMA 接收数据水平寄存器 (IC_DMA_RDLR)

地址偏移: 0x90

复位值: 0x0000_0000

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

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

Bits 31:3	保留，必须保持复位值
Bits 2:0	DMARDL[2:0] : 接收数据水平。这些位控制接收逻辑何时产生一个 DMA 请求。 当接收 FIFO 内有效数据的个数大于或等于 DMARDL+1 且 RDMAE=1 时，产生一个 dma_rx_req 信号。

23.5.35 I2C SDA 建立寄存器 (IC_SDA_SETUP)

地址偏移: 0x94

复位值: 0x0000_0064

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

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

Bits 31:8	保留，必须保持复位值
Bits 7:0	SDA_SETUP[7:0] : 控制 SCL 上升沿中引入的时间延迟量（就 I2C_CLK 时钟周期数而言）相对于 SDA 变化 该寄存器以 I2C_CLK 时钟周期为单位，当 I2C 在作为从机发送模式读请求时，保持 SCL 为低电平来控制 SCL 上升沿引入的时间延迟，相关的 I2C 要求详见 I2C 总线规范。该寄存器必须编程为等于或大于 2 的值。 推荐：如果需要 100ns 的时延，在 10MHz 的 I2C_CLK 下，IC_SDA_SETUP 应该写入 11。
注 1: 只有当 IC_ENABLE [0] = 0 时，才能成功写入该寄存器。	
注 2: 设置时间的长度使用 $(IC_SDA_SETUP - 1) * (i2c_clk_period)$ 计算，所以如果用户需要 10 个 I2C_CLK 周期的设置时间，编程值应为 11。IC_SDA_SETUP 寄存器仅在 I2C 作为从机发送器时使用。	

23.5.36 I2C 广呼应答寄存器 (IC_ACK_GENERAL_CALL)

地址偏移: 0x98

复位值: 0x0000_0001

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

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

Bits 31:1	保留，必须保持复位值
Bit 0	ACK_GEN_CALL: 控制 I2C 在收到广呼地址后是否返回 ACK/NACK。该寄存器仅在从机模式下使用。
	0: 不会返回 ACK 并且不会产生广呼中断
	1: 当收到 General Call 时，I2C 返回 ACK（默认）

23.5.37 I2C 使能状态寄存器 (IC_ENABLE_STATUS)

地址偏移: 0x9C

复位值: 0x0000_0000

该寄存器保存了 I2C 禁止 (IC_ENABLE[0]从 1 变成 0) 时的硬件信息。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.													SLV_RX_DATA_LOST	SLV_DISABLED_WHILE_BUSY	IC_EN
													r	r	r

Bits 31:3	保留，必须保持复位值
Bit 2	SLV_RX_DATA_LOST: 从机收到数据丢失。该位用于指示当 IC_ENABLE[0]从 1 变为 0 的过程中，从机接收器是否丢弃了至少一个收到的数据。
	0: 数据未丢失（默认）
	1: 数据丢失
	<i>注:</i> 外围 I2C 主机在从机发送 NACK 之前，发出 STOP 条件，IC_ENABLE[0]被设置为 0，该位也被设置为 1。
Bit 1	SLV_DISABLED_WHILE_BUSY: 忙状态时（发送/接收），从机禁止。该位指示是否由于 IC_ENABLE 设置成 0，而造成潜在或活跃的从机操作被丢弃。
	0: 从机操作未被丢弃（默认）
	1: 从机操作被丢弃
	当 IC_EN 被设置 0 的同时有以下两种情形的该位置 1: <ul style="list-style-type: none"> ● I2C 正在接收外围主机发送的地址字节（从机发送操作） ● 从外围主机发送的地址和数据字节（从机接收操作）
	<i>注:</i> 外围 I2C 主机在从机发送 NACK 之前，发出 STOP 条件，IC_ENABLE[0]被设置为 0，该位也被设置为 1。
Bit 0	IC_EN: 该位反映输出端口 IC_EN 的状态信息
	0: I2C 不活跃状态（默认）
	1: I2C 活跃状态
	<i>注:</i> CPU 可在任何时候读该位。当该位读到的值为 0 时，CPU 可安全读取 SLV_RX_DATA_LOST 和 SLV_DISABLED_WHILE_BUSY。
<i>注:</i> 当 IC_ENABLE[0]=0，bit[2:1]固定为 0，bit[0]固定为 1；当 IC_ENABLE[0]=1，bit[2:1]仅在 bit[0]读到 0 时有效。	

23.5.38 I2C 标准模式和快速模式尖峰抑制寄存器 (IC_FS_SPKLEN)

地址偏移: 0xA0

复位值: 0x0000_0004

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

Bits 31:8	保留, 必须保持复位值
Bits 7:0	IC_FS_SPKLEN[7:0]:
	该寄存器须在 I2C 总线传输前配置, 确保操作稳定。该寄存器 (以 I2C_CLK 为计数单位) 设定 SCL 或 SDA 上最长脉冲毛刺的宽度, 该毛刺可由尖峰抑制逻辑滤除。
	该寄存器仅在 IC_ENABLE[0]为 0 时配置有效。
	该寄存器最小有效值为 1.当用户试图写入 0 时, 硬件确保写入仍为 1。

23.5.39 I2C 清除 RESTART_DET 寄存器 (IC_CLR_RESTART_DET)

地址偏移: 0xA8

复位值: 0x0000_0000

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

Bits 31:1	保留, 必须保持复位值
Bit 0	CLR_RESTAR_DET:
	读该寄存器来清除 RESTART_DET 中断 (IC_RAW_INTR_STAT 的 bit 12)

24. UART 接口

LCM32F037 内置了 2 个 UART 接口 (UART0、UART1)，通用异步收发器 (UART) 提供了一个灵活的方式，使 MCU 可以与外部设备通过工业标准 NRZ 的形式实现全双工异步串行数据通讯。支持独立的工作时钟，高达 96MHz。

UART0 和 UART1 实现 modem 流控操作 (CTS/RTS)，所有 UART 均可使用 DMA 实现多缓冲区设置，从而能够支持高速数据通讯，UART0/1 的功能如下表所示：

表 24-1 UART0/1 支持的功能

	DMA 访问	CTS/RTS 硬件流控	IrDA SIR ENDEC	发送接收/FIFO
UART0	支持	支持	不支持	支持
UART1	支持	支持	不支持	支持

24.1 UART 主要特性

- 全双工，异步通讯
- NRZ 标准格式 (mark/space)
- 支持接收和发送单独使能
- 8×8bit 发送 FIFO 和 8×12bit 接收 FIFO，减少 CPU 中断，支持 FIFO 功能关闭，即 FIFO 队列深度变为 1
- 发送和接收 FIFO 中断触发阈值可选：FIFO 1/8、1/4、1/2、3/4、7/8
- 波特率发生器：带有 16 位的整数分频寄存器，可将外设时钟经 (1×16) 到 (65535×16) 分频获得波特率时钟。带有 6 位小数分频寄存器可用频率超过 3.6864MHz 的任意时钟作为参考时钟
- 标准的异步通讯位 (start、stop、parity)，这些位在传输前添加，在接收后移除
- 独立的中断屏蔽位，包括发送 FIFO、接收 FIFO、接收停止、错误条件等中断
- 支持 DMA 连续数据通讯
- 支持硬件流控制
- 支持数据格式的灵活配置
 - 停止位个数可设置：支持 1 个或 2 个停止位
 - 数据字长可编程 (5 或 6 位或 7 或 8 位)
 - 校验控制 (发送产生和接收检查)：不带奇偶校验位，奇校验或偶校验，校验位固定为 0 或 1
- 通讯线上 break 信号的产生和检测
- UART 工作模式下支持的最大波特率：921600 bps
- 五种错误检测：溢出错误、帧错误、校验错误、接收停止错误和 Break 检测错误
- 多个中断源和中断标志：溢出错误、帧错误、奇偶错误、打断错误、接收停止、发送完成和接收完成

24.2 功能描述

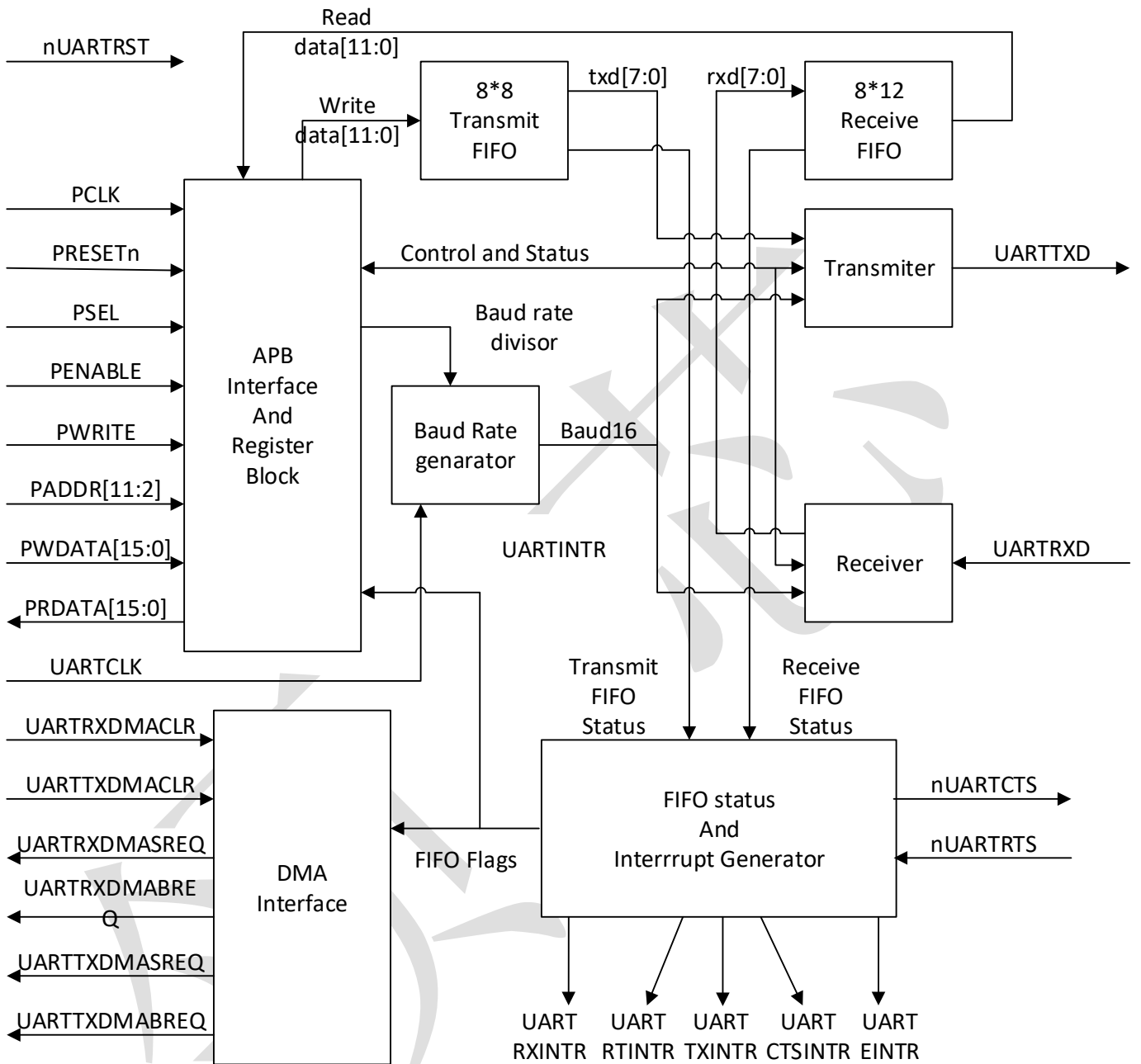


图 24-1 UART 框图

24.2.1 时钟

时钟 UART_CLK 必须适应波特率的设置

- $F_{UART_CLK}(\min) \geq 16 * \text{baud_rate}(\max)$
- $F_{UART_CLK}(\max) \leq 16 * 65535 * \text{baud_rate}(\min)$

例如，波特率的范围为 110bps ~ 460800bps，那么 UART_CLK 频率必须在 7.3728MHz ~ 115.34MHz 间。UART_CLK 和 PCLK 存在一个约束关系：UARTCLK 频率不超过 PCLK 频率的 5/3 倍，即 $F_{UART_CLK} \leq 5/3 * F_{PCLK}$ 。在 UART_CLK 为 14.7456MHz 的条件下产生 921600 的波特率，PCLK 需设置高于 8.85276MHz，以此保证 UART 有足够的时间来将接收到的数据写入接收 FIFO。时钟 UART_CLK 相关配置为：

- 配置系统 SysCtrl_ClkEnR1 寄存器中的 UARTx_CLKSEL[1:0]位，选择 UART_CLK 时钟源（选择相关时钟源需要在 ChipCtrl 寄存器组中使能相应时钟）
- 配置系统 SysCtrl_ClkEnR0 寄存器中的 UARTx_CLKEN 位，使能 UART 时钟

24.2.2 波特率

波特率分频器由 22 位寄存器构成，16 位寄存器存储整数部分，6 位寄存器存储小数部分。利用小数部分可支持频率高于 3.6864MHz 的任意时钟。

$$\text{Baud Rate Divisor} = \text{UART_CLK} / (16 \times \text{Baud Rate}) = \text{BRDI} + \text{BRDF}$$

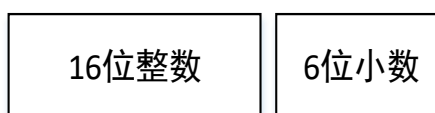


图 24-2 波特率分配器

波特率计算公式如下：

$$\text{BAUDDIV} = (F_{\text{UART_CLK}} / (16 \times \text{Baudrate}))$$

$F_{\text{UART_CLK}}$ 是 UART 的外设时钟。BAUDDIV 由整数值（BAUD DIVINT）和小数值（BAUD DIVFRAC）组成。UARTIBRD 和 UARTFBRD 仅能在发送或接收完成后才能更新配置。BAUDDIV 的范围是 $1 \sim 65535$ ($2^{16} - 1$)，UARTIBRD 最小值为 1，且当 UARTIBRD = 65535 时 UARTFBRD 必须等于 0，否则发送或接收无效。

波特率计算示例（计算主频 4MHz 下波特为 230400 的分频系数）：

$$\text{Baud Rate Divisor} = (4 \times 10^6) / (16 \times 230400) = 1.085$$

于是 BRDI = 1, BRDF = 0.085。小数部分， $m = \text{integer}((0.085 \times 64) + 0.5) = 5$ ，含取整运算和四舍五入。于是 $\text{BDV} = 1 + 5/64 = 1.078$ ，由此推算波特率为 $(4 \times 10^6) / (16 \times 1.078) = 231911$ ，误差为 $(231911 - 230400) / 230400 \times 100 = 0.656\%$ 。利用 6 位 UARTFBRD 计算波特率的最大误差为 $1/64 \times 100 = 1.56\%$ ($m=1$)，64 个计数时钟累积而成。下表中 UART_CLK 为 16MHz，对应的 UARTIBRD，UARTFBRD 以及产生的波特率。

表 24-2 波特率表（UART_CLK 为 16MHz）

UARTIBRD	UARTFBRD	目标波特率 (bps)	获得波特率 (bps)	误差率%
0x2	0xB	460800	460431	0.080
0x4	0x16	230400	230215	0.080
0x8	0x2c	115200	115107	0.080
0xD	0x1	76800	76739	0.079
0x1A	0x3	38400	38392	0.021
0x45	0x1C	14400	14401	0.007
0x1A0	0x2B	2400	2400	0
0x2382	0x3A	110	110	0

下表中 UART_CLK 为 24MHz，对应的 UARTIBRD，UARTFBRD 以及产生的波特率。

表 24-3 波特率表（UART_CLK 为 24MHz）

UARTIBRD	UARTFBRD	目标波特率 (bps)	获得波特率 (bps)	误差率%
0x3	0x10	460800	461538	0.160
0x6	0x21	230400	230216	0.080
0xD	0x1	115200	115246	0.040

0x13	0x22	76800	76800	0
0x27	0x4	38400	38400	0
0x68	0xB	14400	14399	0.007
0x271	0	2400	2400	0
0x3544	0x17	110	110	0

UARTLCR_H、UARTIBRD 和 UARTFBRD 构成了位宽为 30 的寄存器 UARTLCR，这个寄存器通过写 UARTLCR_H 操作更新，所以 UARTLCR_H 的写操作必须放在写 UARTIBRD/UARTFBRD 操作之后进行。同时更新 3 个寄存器的顺序：

- 写 UARTIBRD，写 UARTFBRD，写 UARTLCR_H
- 写 UARTFBRD，写 UARTIBRD，写 UARTLCR_H

仅更新寄存器 UARTIBRD 或 UARTFBRD 的顺序：

- 写 UARTIBRD，写 UARTLCR_H
- 写 UARTFBRD，写 UARTLCR_H

24.2.3 UART 帧格式

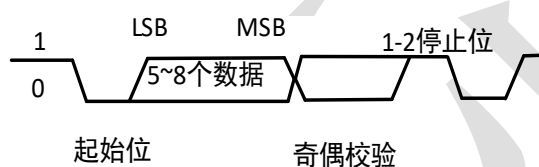


图 24-3 UART 字节帧

通过配置 UARTLCR_H 寄存器的 WLEN[1:0] 位，可选择数据位宽为 5~8 位，通过配置 STP2 位可以选择 1 或者 2 位停止位。

奇偶校验可通过配置 UARTLCR_H 寄存器的 PEN、EPS、SPS 位来实现，具体的配置如下：

1. PEN 位配置成 1，使能奇偶校验功能
2. SPS 位配置成 0，输出的校验值取决于数据内 1 的个数和 EPS 的配置
3. SPS 位配置成 1，固定输出校验值，仅取决于 EPS

24.2.4 数据的发送

发送是对从发送 FIFO 读取的数据到串行转换。按照控制寄存器中的程序配置，在发送起始位后，先发送最低有效位（LSB）开头的的数据位，然后是奇偶校验位，然后是停止位。

发送的数据存储在 8 位宽，深度为 8 的 FIFO 中。对于发送，数据写入到发送 FIFO 中。一旦 UART 使能，触发数据开始发送直到 FIFO 中所有数据发送完毕。数据写入 FIFO 后，忙信号（BUSY）置成高电平，直至 FIFO 为空状态且最后一个字节在移位寄存器发送完毕（Stop 位发送完成）才清 BUSY 状态。

UART 的发送过程如下：

1. 时钟配置
2. 通过 UARTIBRD 和 UARTFBRD 选择合适的波特率
3. 通过 UARTLCR_H 寄存器选择发送数据位宽、停止位、奇偶校验
4. 通过 UARTCR 寄存器选择是否使能硬件控制流和 LBE 回环，并使能 UART 和发送
5. 通过向 UARTDR 寄存器写入数据激活 UART 发送

24.2.5 数据的接收

接收存储在 12 位宽，深度为 8 的 FIFO 中，但接收 FIFO 中每个字节有额外的 4 位来保存状态信息。

当接收器处于空闲态（UARTRXD 持续为 1）且数据线上侦测到低电平，计数器开始工作。在 UART 模式下，每 16 个 Baud16 时钟周期中在第 8 个周期采样数据。接收器检测到数据线低电平后开启计数，在计到第 8 个时钟后 UARTRXD 仍为低电平则识别到有效起始位，否则忽略继续等待接收新的起始位。

有效起始位接收后，接收器开始持续接收数据位，每 16 个 Baud16 采样一个数据位，位宽的长度可配（5~8），数据接收完毕后，接收并检查奇偶校验位（若奇偶校验使能，否则直接接收停止位）。待数据和校验位（若使能）接收完毕，检查 UARTRXD 是否为高电平来确认 Stop 是否有效，若该位为低电平则发生帧错误。

待一帧数据全部接收完毕，数据和状态信息一同写入接收 FIFO 中。

表 24-4 接收 FIFO 功能位

FIFO bit	功能
11	Overrun indicator
10	Break error
9	Parity error
8	Framing error
7:0	Received data

如上表，3 个错误位存储在 bit[10:8]，与当前接收的字节数据的状态相关。Overrun indicator 存在 bit[11]，在当前 FIFO 处于满状态且移位寄存器接收完下个字节试图写入 FIFO 时，发生 Overrun 错误，bit[11]置成 1。发生溢出错误后，移位寄存器中的数据被覆盖，但不会写入到 FIFO 中。当 FIFO 处于非满状态，移位寄存器接收完下个字节，数据携同 Overrun 信息一同写入 FIFO 中。

FIFO 不使能时，FIFO 队列的深度由 8 变成 1（当成 1 字节的保持寄存器）。在之前数据没读走，新的数据又接收完成的情况下 Overrun 置位。在 FIFO 模式下，BE/PE/FE 等异常状态与 FIFO 的顶部字节相关。读取 UARTRSR 前，必须先读取 UARTDR，次序不能颠倒。UARTDR 内的 OE/BE/PE/FE 等信息不能长时间维持。通过读 UARTDR 操作，将接收状态保存到 UARTRSR 中，通过读取 UARTDR 也可以获取当前状态信息。

UART 的接收过程如下：

1. 时钟配置
2. 通过 UARTIBRD 和 UARTFBRD 选择合适的波特率
3. 通过 UARLCCR_H 寄存器选择接收数据位宽、停止位、奇偶校验
4. 通过 UARTCR 寄存器选择是否使能硬件控制流，并使能 UART 和接收
5. 从数据寄存器 UARTDR 中读走接收到的数据

24.2.6 UART 硬件流控制

UART 支持硬件流控通过 nRTS 输出信号和 nCTS 输入信号控制两个 UART 之间的数据。

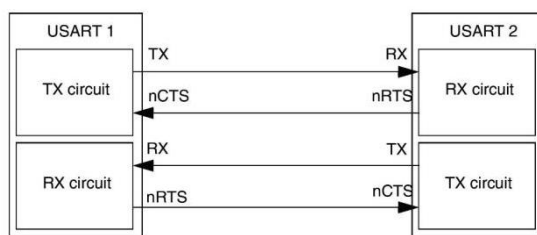


图 24-4 两个相似设备间的硬件流控

当 RTS 流控使能时，nRTS 被拉成高电平，直到接收 FIFO 被填满到水平基准。当 CTS 流控使能时，发送器仅在 nCTS 为高电平时发送数据。硬件流控可用 UARTCR 的 RTSEn 和 CTSEn 控制。表 24-5 描述了这两位在不同配置下的控制流情况。

表 24-5 硬件流控的使能控制位

CTSEn	RTSEn	说明
1	1	RTS 和 CTS 流控使能
1	0	只有 CTS 流控使能
0	1	只有 RTS 流控使能
0	0	RTS 和 CTS 流控禁止

24.2.6.1 RTS 流控制

RTS 流控逻辑与接收 FIFO 的水平基准有关。当 RTSEn 使能时，当 FIFO 内的数据数少于水平基准，nUARTRTS 为高电平。一旦达到基准 nUARTRTS 清 0，表示 FIFO 没有接收数据的空间了，当前字节发送完毕就停止。

当有数据从 FIFO 中读走且 FIFO 内的字节数小于水平基准时，nUARTRTS 置成高电平。如果 RTS 流控使能关闭（UART 使能），数据根据当前 nUARTRTS 的状态，要么一直接收至 FIFO 满，要么不接收数据。

24.2.6.2 CTS 流控制

当 CTSEn 使能时，发送器在传送数据前检查 nUARTCTS 的状态。如果 nUARTCTS 为低电平，发送数据。反之，不发送。当发送 FIFO 非空且 nUARTCTS 为低电平，发送器将持续发送数据。当 CTSEn 使能时，nUARTCTS 被置 0，发送器将发完当前字节数据后停止。如果 CST 流控使能关闭（UART 使能），发送器持续发送数据直至 FIFO 空。

24.2.7 DMA 接口

UART 支持 DMA 访问，提高数据通讯的效率。UART 的 DMA 接口有以下信号：

- UARTRXDMASREQ: 单字节 DMA 读取请求，由 UART 置位。对于接收，一个数据达 12 位。当接收 FIFO 中至少有一个数据时，该信号置位。
- UARTRXDMABREQ: 连续 DMA 读取请求，由 UART 置位。当接收 FIFO 中的数据数超过水平基准时，该信号置 1。用户可通过 UARTIFLS 配置 FIFO 的水平基准。
- UARTRXDMACLR: DMA 读取请求清除信号，由 DMA 控制器置位，用于清除 UART 读取请求信号。如果 UART 向 DMA 控制器发出连续读数据请求（UARTRXDMABREQ），该清除信号将维持高电平到突发传输的最后一个数据。
- UARTRXDMASREQ: 单字节 DMA 写入请求，由 UART 置位。当发送 FIFO 中至少有一个可写入数据

的位置时，该信号置位，向 DMA 发出请求。

- **UARTTXDMABREQ**: 连续 DMA 写入请求，由 UART 置位。当发送 FIFO 中的数据个数少于水平基准时，该信号置 1。用户可通过 UARTIFLS 配置 FIFO 的水平基准。
- **UARTTXDMACLR**: 写入请求清除信号，由 DMA 控制器置位，用于清除 UART 写入请求信号。如果 UART 向 DMA 控制器发出连续写数据请求（UARTTXDMABREQ），该清除信号将维持高电平到突发传输的最后一个数据。

DMA 单请求和连续信号请求不会相互干扰，可同时有效。例如，当 FIFO 内数据数超过水平基准时，两个请求信号同时有效。当 FIFO 内数据数不少于 1 个但不超过水平基准数时，UART 仅发出 DMA 单请求。每个请求信号在相关的 DMACLR 来临前，一直保持有效。在 DMA 清除信号拉低后，如果条件满足，UART 的请求信号又可被再次置位。

所有 DMA 请求信号在 UART 禁止或相关 DMA 使能（TXDMAE、RXDMAE）关闭，UART 请求信号自动清除。如果 FIFO 禁止，仅支持 DMA 单请求。

表 24-6 发送接收 FIFO 的 DMA 请求触发点

基准数	发送（空槽数）	接收（满槽数）
1/8	7	1
1/4	5	2
1/2	4	4
3/4	2	6
7/8	1	7

此外，当接收错误发生时，DMAONERR 置位（UARTDMACR），产生接收错误中断（UARTEINTR），UARTRXDMASREQ 和 UARTRXDMABREQ 将被屏蔽，直至 UARTEINTR 清 0。DMA 的发送请求不受接收错误影响。

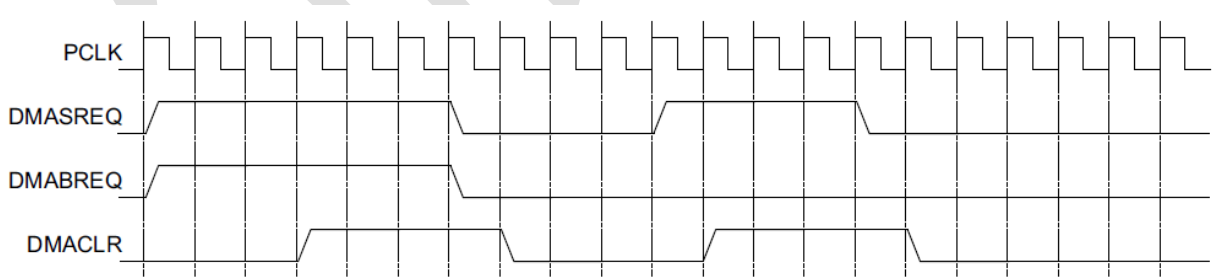


图 24-5 DMA 单请求和 DMA 连续请求与 DMACLR 间的时序关系

24.3 UART 中断

UART 有 5 个可屏蔽中断源，这些独立的中断经过或逻辑产生 1 个中断输出，对应着系统中的 UART 全局中断向量。

- UARTRXINTR
- UARTTXINTR
- UARTRTINTR

- UARTEINTR, 由以下条件触发:
 - UARTOEINTR, FIFO溢出
 - UARTBEINTR, 接收过程中出现打断
 - UARTPEINTR, 接收的字节出现奇偶校验错误
 - UARTFEINTR, 接收的字节出现帧错误

最终UARTEINTR输出由上述4个中断源经过或逻辑输出, 用户可通过UARTIMSC寄存器, 使能或屏蔽每个独立的中断源。每个中断的状态可从UARTRIS或UARTMIS中读取。

UARTRXINTR

接收中断发生在以下两种情形:

- FIFO使能时, 接收FIFO内的数据个数达到设定的触发水平, 接收中断置位。可以由2种方式清中断: 从FIFO中读数据直到数据个数小于触发水平; 写中断清除寄存器相关位清除中断。
- FIFO禁止时, 接收到一个数据, 接收中断置位。可以由2种方式清中断: 对接收FIFO进行单次读数据操作; 写中断清除寄存器相关位清除中断。

UARTTXINTR

发送中断发生在以下两种情形:

- FIFO使能时, 接收FIFO内的数据个数小于等于触发水平, 发送中断置位。可以由2种方式清中断: 往FIFO中写数据直到数据个数大于触发水平; 写中断清除寄存器相关位清除中断。
- FIFO禁止时, 发送FIFO中(深度为1), 发送中断置位。可以由2种方式清中断: 对接收FIFO进行单次写数据操作; 写中断清除寄存器相关位清除中断。

UARTRTINTR

接收停止中断发生在接收FIFO为空且在32位数据的时间窗内没有收到数据。可以由2种方式清中断: 对接收FIFO进行读数据直至空; 写中断清除寄存器相关位清除中断。

UARTEINTR

UART接收数据出错时产生该中断。由以下错误条件产生: 帧错误, 奇偶校验错误, 打断错误, 溢出错误。用户可读UARTMIS或UARTRIS来判断错误源, 写中断清除寄存器相关位清除中断。

UARTCTSINTR

UART启动硬件流控时, 由CTS产生该中断, 写中断清除寄存器相关位清除中断。

表 24-7 UART 中断请求

中断事件	事件标志	使能位控制
发送完成	TXMIS	TXIM
接收完成	RXMIS	RXIM
接收停止	RTMIS	RTIM
奇偶错误	PEMIS	PEIM
帧错误	FEMIS	FEIM
溢出错误	OEMIS	OEIM
打断错误	BEMIS	BEIM
硬控流	CTSMIS	CTSIM

UART中断全部连接到同一个中断向量, 如果设置了相应的使能控制位, 它们都可以引起中断。

24.4 UART 寄存器描述

UART0 基地址为 0x4001_4000，UART1 基地址为 0x4001_5000，UART0 和 UART1 的地址偏移分别对应各自的基地址。

表 24-8 UART 相关寄存器表

名称	说明	读写权限	复位值	地址偏移
UARTDR	数据寄存器	R/W	0x0000_0000	0x00
UARTSR	接收状态寄存器	R/W	0x0000_0000	0x04
UARTFR	标志寄存器	R	0x0000_0090	0x18
UARTIBRD	整数波特率寄存器	R/W	0x0000_0000	0x24
UARTFBRD	小数波特率寄存器	R/W	0x0000_0000	0x28
UARTLCR_H	线控制寄存器	R/W	0x0000_0000	0x2C
UARTCR	控制寄存器	R/W	0x0000_0300	0x30
UARTIFLS	中断 FIFO 触发水平选择寄存器	R/W	0x0000_0012	0x34
UARTIMSC	中断屏蔽寄存器	R/W	0x0000_0000	0x38
UARTRIS	原始中断状态寄存器	R	0x0000_0000	0x3C
UARTMIS	屏蔽后中断状态寄存器	R	0x0000_0000	0x40
UARTICR	中断清除寄存器	W	0x0000_0000	0x44
UARTDMACR	DMA 控制寄存器	R/W	0x0000_0000	0x48

注：x 表示不确定；-表示未实现；u 表示不受复位影响；c 表示该位复位值因复位源而异；R 只读；W 只写；R-0 表示只读且读 0；R/W 表示可读写（以后章节同上述）。

24.4.1 UART0/1 数据寄存器（UARTDR）

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
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				OE	BE	PE	FE	DATA[7:0]							
				rw	rw	rw	rw	rw							

Bits	31:12	保留，必须保持复位值
Bit	11	OE ：溢出错误，写 1 清除标志位
		当接收 FIFO 在满状态又收到新的数据而溢出，该位置 1。一旦 FIFO 有接收新数据的空间该位清 0。
		0：无溢出错误（默认）
		1：溢出错误
Bit	10	BE ：打断错误，写 1 清除标志位
		当接收的数据（起始位+数据位+校验位+停止位）全为 0 则发生打断错误，该位置

	1。
	在 FIFO 模式下，BE 与 FIFO 的顶部字节相关。当 BE 发生时，一个字节 0 载入 FIFO。新的数据仅在数据线变为高电平并且检测到起始位才有效，并载入 FIFO 中。
	0: 无打断错误（默认）
	1: 打断错误
Bit 9	PE: 奇偶校验错误，写 1 清除标志位
	当在接收数据的时候发现校验错误，该位会由硬件置 1。
	0: 无奇偶校验错误（默认）
	1: 奇偶校验错误
Bit 8	FE: 帧格式错误，写 1 清除标志位
	当在接收数据的时候发现无有效停止位，该位会由硬件置 1。
	0: 无帧格式错误（默认）
	1: 帧格式错误
Bits 7:0	DATA[7:0]: 数据寄存器，接收数据字节（读操作），发送数据字节（写操作）
注: 在 FIFO 模式下，BE/PE/FE 等异常状态与 FIFO 的顶部字节相关。	

24.4.2 UART0/1 接收状态寄存器（UARTRSR）

地址偏移：0x04

复位值：0x0000_0000

接收状态可从该寄存器读取，其信息映射了 UARTDR 的 OE/BE/PE/FE 的状态。

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

Bits 31:4	保留，必须保持复位值
Bit 3	OE: 溢出错误，写 1 清除标志位
	当接收 FIFO 在满状态又收到新的数据而溢出，该位置 1。通过读 UARTDR 操作，使 FIFO 处非满状态，该位清 0。
	该位和 UARTDR 的 OE 位的区别是：当 FIFO 满时，队列内的数据有效，新的数据不会覆盖队列内容，仅影响移位寄存器。CPU 必须通过读数据清空 FIFO。
	0: 无溢出错误（默认）
	1: 溢出错误
Bit 2	BE: 打断错误，写 1 清除标志位
	当接收的数据（起始位+数据位+校验位+停止位）全为 0，则发生打断错误，该位置 1。
	0: 无打断错误（默认）
	1: 打断错误
Bit 1	PE: 奇偶校验错误，写 1 清除标志位
	当在接收数据的时候发现校验错误，该位会由硬件置 1。

	0: 无奇偶校验错误（默认）
	1: 奇偶校验错误
Bit 0	FE: 帧格式错误，写 1 清除标志位
	当接收数据的时候发现无有效停止位，该位会由硬件置 1。
	0: 无帧格式错误（默认）
	1: 帧格式错误

24.4.3 UART0/1 标志寄存器（UARTFR）

地址偏移：0x18

复位值：0x0000_0090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								TXFE	RXFF	TXFF	RXFE	BUSY	Res.		nCTS
								r	r	r	r	r			r

Bits 31:8	保留，必须保持复位值
Bit 7	TXFE: 发送 FIFO 空标志位
	当 FIFO 禁止（FEN=0），当发送保持寄存器为空时 TXFE 置 1
	当 FIFO 使能（FEN=1），当发送 FIFO 为空时 TXFE 置 1
	该位并不指示发送移位寄存器中有数据。
	0: 发送 FIFO 非空
	1: 发送 FIFO 空（默认）
Bit 6	RXFF: 接收 FIFO 满标志位
	当 FIFO 禁止（FEN=0），当接收保持寄存器为满时 RXFF 置 1
	当 FIFO 使能（FEN=1），当接收 FIFO 为满时 RXFF 置 1
	0: 接收 FIFO 未满足（默认）
	1: 接收 FIFO 满
Bit 5	TXFF: 发送 FIFO 满标志位
	当 FIFO 禁止（FEN=0），当接收保持寄存器为满时 TXFF 置 1
	当 FIFO 使能（FEN=1），当接收 FIFO 为满时 TXFF 置 1
	0: 发送 FIFO 未满足（默认）
	1: 发送 FIFO 满
Bit 4	RXFE: 接收 FIFO 空标志位
	当 FIFO 禁止（FEN=0），当接收保持寄存器为空时 RXFE 置 1
	当 FIFO 使能（FEN=1），当接收 FIFO 为空时 RXFE 置 1
	0: 接收 FIFO 非空
	1: 接收 FIFO 空（默认）
Bit 3	BUSY: UART 忙标志位
	当 UART 在发送数据时该位置 1。不管 UART 使能与否，一旦 FIFO 非空该位置 1
	当 FIFO 满时，队列内的数据有效，新的数据不会覆盖队列内容，仅影响移位寄存

	器。CPU 必须通过读数据清空 FIFO。
	0: UART 处于空闲状态 (默认)
	1: UART 处于忙状态
Bits 2:1	保留, 必须保持复位值
Bit 0	nCTS : CTS 标志位, 是 nUARTCTS 的互补位
	0: 表示 nUARTCTS 为 1 (默认)
	1: 表示 nUARTCTS 为 0

24.4.4 UART0/1 整数波特率寄存器 (UARTIBRD)

地址偏移: 0x24

复位值: 0x0000_0000

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

Bits 31:16	保留, 必须保持复位值
Bits 15:0	BAUDDIVINT[15:0] : 波特率整数部分的分频系数

24.4.5 UART0/1 小数波特率寄存器 (UARTFBRD)

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

Bits 31:6	保留, 必须保持复位值
Bits 5:0	BAUDDIVFRAC[5:0] : 波特率小数部分的分频系数

24.4.6 UART0/1 线控制寄存器 (UARTLCR_H)

地址偏移: 0x2C

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								SPS	WLEN[1:0]		FEN	STP2	EPS	PEN	BRK
								rw	rw		rw	rw	rw	rw	rw

Bits 31:8	保留，必须保持复位值
Bit 7	SPS: 强制奇偶校验位选择，与 EPS 和 PEN 配合使用。
	0: 强制奇偶校验选择关闭（输出校验位的值取决于数据内 1 的个数和 EPS）（默认）
	1: 强制奇偶校验选择使能（输出校验位为固定的值，仅取决于 EPS）
	若 EPS=0，帧字节中发送时奇偶校验位配置为 1，接收时检查校验位是否为 1
	若 EPS=1，帧字节中发送时奇偶校验位配置为 0，接收时检查校验位是否为 0
Bits 6:5	WLEN[1:0]: 一帧数据位长度选择
	00: 6 位（默认）
	01: 6 位
	10: 7 位
	11: 8 位
Bit 4	FEN: FIFO 使能控制
	0: FIFO 禁止，对列变成 1 字节深度的保持寄存器（默认）
	1: 发送 FIFO 和接收 FIFO 使能
Bit 3	STP2: 2 位 Stop 位选择，为 1 时每帧中包含 2 个停止位，发送时帧末尾带上，而接收则不检查
	0: 1 位停止位（默认）
	1: 2 位停止位
Bit 2	EPS: 奇偶校验位的奇偶选择
	0: 奇校验，UART 生成或检查数据位和校验位中所有 1 的个数，若为奇数个则输出为 1 或 1 正确（默认）
	1: 偶校验，UART 生成或检查数据位和校验位中所有 1 的个数，若为偶数个则输出为 1 或 1 正确
Bit 1	PEN: 奇偶校验使能
	0: 禁止（默认）
	1: 使能
Bit 0	BRK: 发送打断
	当软件配置为 1 时，当前字节数据发送完成后，UARTTXD 仍持续输出低电平。为正确执行 break 命令，BRK 须维持至少 2 个帧的时间。正常使用时，该位必须为 0。
	0: 禁止（默认）
	1: 使能

注: UARTLCR_H、UARTIBRD 和 UARTFBRD 构成了位宽为 30 的寄存器 UARTLCR，这个寄存器通过写 UARTLCR_H 操作更新，所以 UARTLCR_H 的写操作必须放在写 UARTIBRD/UARTFBRD 操作之后进行。

同时更新 3 个寄存器的顺序:

- 写 UARTIBR，写 UARTFBRD，写 UARTLCR_H
- 写 UARTFBRD，写 UARTIBRD，写 UARTLCR_H

仅更新寄存器 UARTIBRD 或 UARTFBRD 的顺序:

- 写 UARTIBRD，写 UARTLCR_H

● 写 UARTFBRD, 写 UARTLCR_H

表 24-9 奇偶校验位配置真值表

PEN	EPS	SPS	校验位
0	x	x	x
1	1	0	偶校验
1	0	0	奇校验
1	0	1	1
1	1	1	0

24.4.7 UART0/1 控制寄存器 (UARTCR)

地址偏移: 0x30

复位值: 0x0000_0300

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTSEn	RTSEn	Res.		RTS	Res.	RXE	TXE	LBE	Res.						UARTEN
rw	rw			rw		rw	rw	rw							rw

Bits 31:16	保留, 必须保持复位值
Bit 15	CTSEn: CTS 硬件流控使能, 为 1 时, 流控使能, 仅在 nUARTCTS 为低电平时发送数据 0: CTS 禁止 (默认) 1: CTS 使能
Bit 14	RTSEn: RTS 硬件流控使能, 为 1 时, 流控使能, 仅在接收 FIFO 有接收新数据的空间时才发起数据读取请求 0: RTS 禁止 (默认) 1: RTS 使能
Bits 13:12	保留, 必须保持复位值
Bit 11	RTS: RTS 流控非使能时, UART 发送请求的互补位 0: 控制 nUARTRTS 为 1 (默认) 1: 控制 nUARTRTS 为 0
Bit 10	保留, 必须保持复位值
Bit 9	RXE: 接收使能, 数据接收过程中关闭 UART 使能, 在停止前仍能接收当前字节 0: 接收禁止 1: 接收使能 (默认)
Bit 8	TXE: 发送使能, 数据发送过程中关闭 UART 使能, 在停止前仍能发送当前字节 0: 发送禁止 1: 发送使能 (默认)
bit 7	LBE: 回环使能 0: UARTTXD 与 UARTRXD 分离 (默认)

	1: UARTTXD 路径插入到 UARTRXD
Bits 6:1	保留, 必须保持复位值
Bit 0	UARTEN : UART 使能
	0: UART 禁止 (默认)
	1: UART 使能
注: 发送时 TXE 和 UARTEN 必须使能; 接收时 RXE 和 UARTEN 必须使能。配置 UARTLCR 流程:	
1. 清 UART 使能	
2. 等待当前字节发送或接收完毕	
3. 通过把 UARTLCR_H 的 FEN 位清 0, 来清空发送 FIFO	
4. 重新配置 UARTCR	
5. 使能 UART	

24.4.8 UART0/1 中断 FIFO 触发水平选择寄存器 (UARTIFLS)

地址偏移: 0x34

复位值: 0x0000_0012

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

Bits 31:6	保留, 必须保持复位值
Bits 5:3	RXIFLSEL[2:0] : 接收中断触发水平选择, 当接收 FIFO 内所包含有效字节数超过预设的触发水平时发相应的中断
	000: 接收 FIFO 中队列数 $\geq 1/8$ 满
	001: 接收 FIFO 中队列数 $\geq 1/4$ 满
	010: 接收 FIFO 中队列数 $\geq 1/2$ 满 (默认)
	011: 接收 FIFO 中队列数 $\geq 3/4$ 满
	100: 接收 FIFO 中队列数 $\geq 7/8$ 满
	其他: 保留
Bits 2:0	TXIFLSEL[2:0] : 发送中断触发水平选择, 当发送 FIFO 内所包含有效字节数超过预设的触发水平时发相应的中断
	000: 发送 FIFO 中队列数 $\geq 1/8$ 满
	001: 发送 FIFO 中队列数 $\geq 1/4$ 满
	010: 发送 FIFO 中队列数 $\geq 1/2$ 满 (默认)
	011: 发送 FIFO 中队列数 $\geq 3/4$ 满
	100: 发送 FIFO 中队列数 $\geq 7/8$ 满
	其他: 保留

24.4.9 UART0/1 中断屏蔽寄存器 (UARTIMSC)

地址偏移: 0x38

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					OEIM	BEIM	PEIM	FEIM	RTIM	TXIM	RXIM	Res.		CTSIM	Res.
					rw	rw	rw	rw	rw	rw	rw			rw	

Bits 31:11	保留, 必须保持复位值
Bit 10	OEIM: 溢出错误中断屏蔽位
	0: 屏蔽 (默认)
	1: 不屏蔽
Bit 9	BEIM: 打断错误中断屏蔽位
	0: 屏蔽 (默认)
	1: 不屏蔽
Bit 8	PEIM: 奇偶校验错误中断屏蔽位
	0: 屏蔽 (默认)
	1: 不屏蔽
Bit 7	FEIM: 帧错误中断屏蔽位 (接收 Stop 位时数据线为低电平)
	0: 屏蔽 (默认)
	1: 不屏蔽
Bit 6	RTIM: 接收停止中断屏蔽位
	0: 屏蔽 (默认)
	1: 不屏蔽
Bit 5	TXIM: 发送中断屏蔽位
	0: 屏蔽 (默认)
	1: 不屏蔽
Bit 4	RXIM: 接收中断屏蔽位
	0: 屏蔽 (默认)
	1: 不屏蔽
Bits 3:2	保留, 必须保持复位值
Bit 1	CTSIM: 硬控流中断屏蔽位
	0: 屏蔽 (默认)
	1: 不屏蔽
Bit 0	保留, 必须保持复位值
注: 复位后上述寄存器为0, 对应的中断将被屏蔽; 当寄存器值为1时, 对应中断被放开。	

24.4.10 UART0/1 原始中断状态寄存器 (UARTRIS)

地址偏移: 0x3C

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					OERIS	BERIS	PERIS	FERIS	RTRIS	TXRIS	RXRIS	Res.		CTSRIS	
					r	r	r	r	r	r	r			r	

Bits	31:11	保留，必须保持复位值
Bit	10	OERIS: 溢出错误中断状态位
		0: 溢出错误中断未发生（默认）
		1: 溢出错误中断发生
Bit	9	BERIS: 打断错误中断状态位
		0: 打断错误中断未发生（默认）
		1: 打断错误中断发生
Bit	8	PERIS: 奇偶校验错误中断状态位
		0: 奇偶校验错误中断未发生（默认）
		1: 奇偶校验错误中断发生
Bit	7	FERIS: 帧错误中断状态位（接收 Stop 位时数据线为低电平）
		0: 帧错误中断未发生（默认）
		1: 帧错误中断发生
Bit	6	RTRIS: 接收停止中断状态位
		0: 接收停止中断未发生（默认）
		1: 接收停止中断发生
Bit	5	TXRIS: 发送中断状态位
		0: 发送中断未发生（默认）
		1: 发送中断发生
Bit	4	RXRIS: 接收中断状态位
		0: 接收中断未发生（默认）
		1: 接收中断发生
Bits	3:2	保留，必须保持复位值
Bit	1	CTSRIS: 硬控流中断状态位
		0: 硬控流中断未发生（默认）
		1: 硬控流中断发生
Bit	0	保留，必须保持复位值
注: 该寄存器只能进行读操作，直接返回对应中断的状态，与中断屏蔽与否无关。		

24.4.11 UART0/1 屏蔽后中断状态寄存器（UARTMIS）

地址偏移：0x40

复位值：0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					OEMIS	BEMIS	PEMIS	FEMIS	RTMIS	TXMIS	RXMIS	Res.		CTSMIS	Res.
					r	r	r	r	r	r	r			r	

Bits	31:11	保留，必须保持复位值
Bit	10	OEMIS : 溢出错误屏蔽中断状态位
		0: 溢出错误中断未发生或被屏蔽（默认）
		1: 溢出错误中断未被屏蔽
Bit	9	BEMIS : 打断错误屏蔽中断状态位
		0: 打断错误中断未发生或被屏蔽（默认）
		1: 打断错误中断未被屏蔽
Bit	8	PEMIS : 奇偶校验屏蔽错误中断状态位
		0: 奇偶校验中断未发生或被屏蔽（默认）
		1: 奇偶校验中断未被屏蔽
Bit	7	FEMIS : 帧错误屏蔽中断状态位（接收 Stop 位时数据线为低电平）
		0: 帧错误中断未发生或被屏蔽（默认）
		1: 帧错误错误中断未被屏蔽
Bit	6	RTMIS : 接收停止屏蔽中断状态位
		0: 接收停止中断未发生或被屏蔽（默认）
		1: 接收停止中断未被屏蔽
Bit	5	TXMIS : 发送屏蔽中断状态位
		0: 发送中断未发生或被屏蔽（默认）
		1: 发送中断未被屏蔽
Bit	4	RXMIS : 接收屏蔽中断状态位
		0: 接收中断未发生或被屏蔽（默认）
		1: 接收中断未被屏蔽
Bits	3: 2	保留，必须保持复位值
Bit	1	CTSMIS : 硬控流屏蔽中断状态位
		0: 硬控流中断未发生或被屏蔽（默认）
		1: 硬控流中断未被屏蔽
Bit	0	保留，必须保持复位值
注: 该寄存器只能进行读操作，当中断屏蔽为 1 时读出对应中断状态，反之读出 0。		

24.4.12 UART0/1 中断清除寄存器（UARTICR）

地址偏移: 0x44

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					OEIC	BEIC	PEIC	FEIC	RTIC	TXIC	RXIC	Res.		CTSIC	Res.
					w	w	w	w	w	w	w			w	

Bits	31:11	保留，必须保持复位值
Bit	10	OEIC : 溢出错误中断清除位
Bit	9	BEIC : 打断错误中断清除位
Bit	8	PEIC : 奇偶校验错误中断清除位
Bit	7	FEIC : 帧错误中断清除位（接收 Stop 位时数据线为低电平）
Bit	6	RTIC : 接收停止中断清除位
Bit	5	TXIC : 发送中断清除位
Bit	4	RXIC : 接收中断清除位
Bits	3:2	保留，必须保持复位值
Bit	1	CTSIC : 硬控流中断清除位
Bit	0	保留，必须保持复位值
注 : 该寄存器只能进行写操作，写1清除中断标志位，写0无效。		

24.4.13 UART0/1 DMA 控制寄存器（UARTDMACR）

地址偏移: 0x48

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.												DMA ONERR	TX DMAE	RX DMAE	
													rw	rw	rw

Bits	31:3	保留，必须保持复位值
Bit	2	DMAONERR : 发送接收响应屏蔽位
		0: 发生溢出错误/奇偶校验错误/帧错误/打断错误时，UARTRXDMASREQ 和 UARTRDSMABREQ 不会被屏蔽（默认）
		1: 发生溢出错误/奇偶校验错误/帧错误/打断错误时，UARTRXDMASREQ 和 UARTRDSMABREQ 被屏蔽
Bit	1	TXDMAE : 发送 FIFO 的 DMA 访问使能
		0: 发送 DMA 访问禁止（默认）
		1: 发送 DMA 访问使能
Bit	0	RXDMAE : 接收 FIFO 的 DMA 访问使能

0:	接收 DMA 访问禁止 (默认)
1:	接收 DMA 访问使能

领芯微

25.SSP 接口

25.1 SSP 主要特征

LCM32F037 内置 2 个 SSP 接口 (SSP0、SSP2)，主要包含如下特性：

- 支持主机和从机模式
- 支持 Motorola SPI、TI SSI、National Semiconductor Microwire 三种接口协议
- 帧长度 4 到 16 位可配
- 独立的工作时钟，高达 96MHz
- 可编程的预分频系数和传输比特率
- 在主机模式时，最高通信速率可达 32Mbps；在从机模式时，最高通信速率可达 8Mbps
- 支持内部回环测试模式
- 独立的 8x16bit 发送和接收 FIFO
- 支持 DMA 操作

25.2 管脚配置

表 25-1 SSP 管脚配置

管脚名称	管脚类型	管脚描述	复用 I/O 口	配置
SSP0_TXD	O	SSP0 发送	PA3/PA6/PA7/PB4/PB5	SSPOSSE 置 1, SSP0SOD 置 0
SSP0_RXD	I	SSP0 接收	PA3/PA6/PA7/PB4/PB5	SSPOSSE 置 1
SSP0_SCK	I/O	SSP0 时钟	PA5/PB3	SSPOSSE 置 1, SSP0SOD 置 0
SSP0_FSS	I/O	SSP0 片选	PA5/PA15	SSPOSSE 置 1
SSP2_TXD	O	SSP2 发送	PA8/PA9/PF0/PF1	SSP2SSE 置 1, SSP2SOD 置 0
SSP2_RXD	I	SSP2 接收	PA8/PA9/PF0/PF1	SSP2SSE 置 1
SSP2_SCK	I/O	SSP2 时钟	PF3/PF7	SSP2SSE 置 1, SSP2SOD 置 0
SSP2_FSS	I/O	SSP2 片选	PB7/PF4/PF6	SSP2SSE 置 1

25.3 功能描述

SSP (Synchronous Serial Port) 是同步串行接口，可作为主设备或从设备，允许与 Motorola SPI, National Semiconductor Microwire 或 TI 协议的主从设备进行同步串行通信。图 25-1 展示了 SSP 的结构框图。

SSP 通过四线实现主从机同步串行通讯，分别为 2 根数据线，1 根时钟线，1 根片选线，具体如下：

- SSPTXD：数据发送端口，在主机和从机模式下都作为数据发送端口
- SSPRXD：数据接收端口，在主机和从机模式下都作为数据接收端口
- SSPCLKOUT：主机时钟输出端口
- SSPFSSOUT：主机片选输出端口
- SSPCLKIN：从机时钟输入端口
- SSPFSSIN：从机片选输入端口

此外，SSP 在背靠背传输等简单应用场景中支持 3 线模式，通过不配置从机 SSPFSS 的 GPIO 数字复用，从机 SSPFSS 会处于默认有效，借此可以取消主从机 FSS 片选线之间的连接。在此模式下，只支持 SPH=1 的传输。

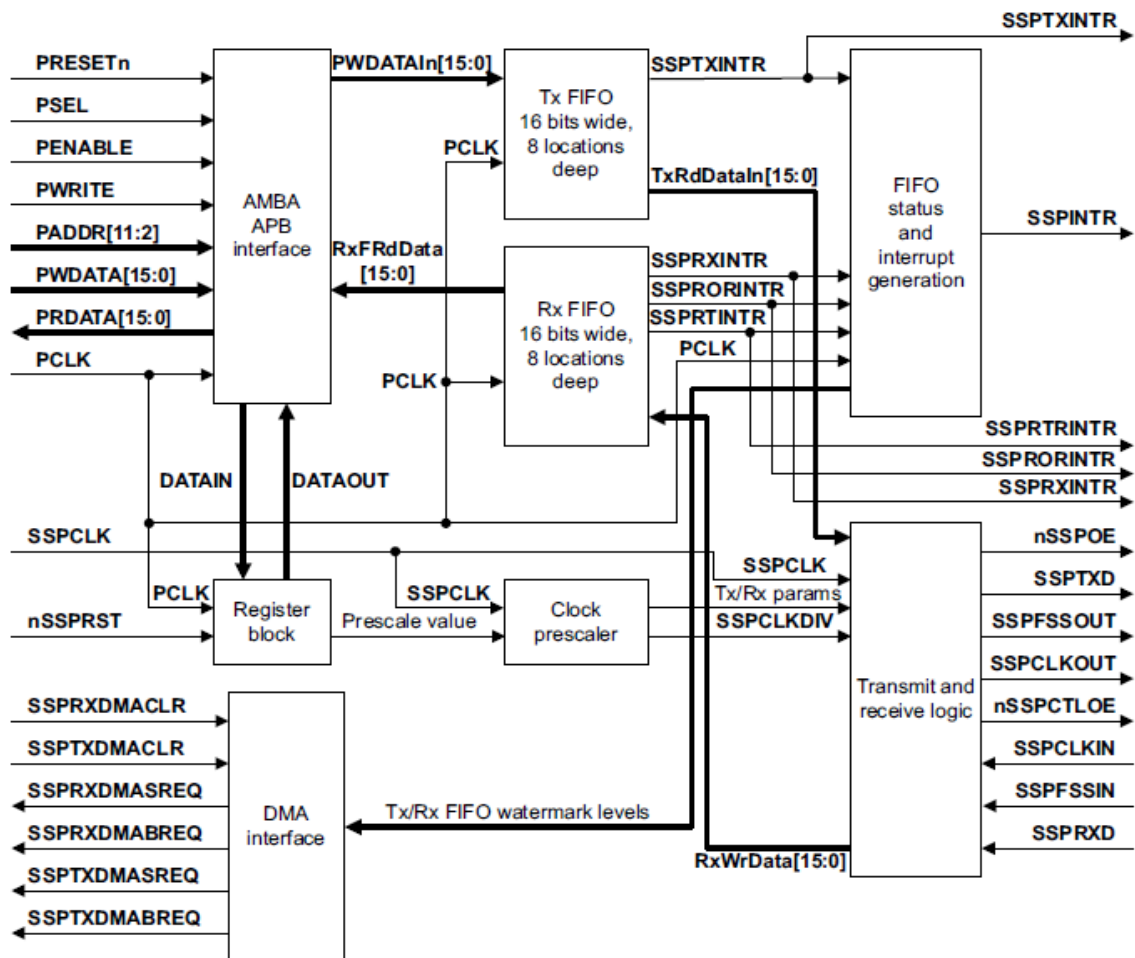


图 25-1 SSP 结构框图

SSP 支持全双工通讯，通讯由主机发起，串行数据在 SSPTXD 上传输，在 SSPRXD 上接收，对从外设接收的数据执行串行到并行转换。通过总线访问数据、控制和状态寄存器，发送和接收路径通过内部 FIFO 存储器启用缓冲，在发送和接收模式下独立存储 8 个 16 位值。

SSP 包含一个可编程比特率时钟分频器和预分频器。用于产生串行输出时钟 SSPCLKOUT，由输入时钟 SSCLK 分频而得。支持比特率 2MHz 或更高，根据 SSCLK 的频率选择，最大传输比特率由外围设备决定。

用户可用控制寄存器 SSPCR0 和 SSPCR1 改变操作模式，帧格式和帧长度。

25.4 SSP 操作

25.4.1 时钟比率

串行比特率是输入时钟 SSCLK 经过分频而来。该时钟先经过偶数分频值 CPSDVSR（范围 2~254）分频，再经过 1+SCR（范围 0~255）分频，SCR 在 SSPCR0 寄存器中配置。

下面等式定义了输出信号位时钟 SSPCLKOUT 频率：

$$F_{SSPCLKOUT} = \frac{F_{SSCLK}}{CPSDVSR \times (1 + SCR)}$$

例如， F_{SSCLK} 为 3.6864MHz， $CPSDVSR=2$ ， $F_{SSPCLKOUT}$ 的频率范围为 7.2KHz ~1.8432MHz。

PCLK0 和 SSCLK 之间的比率存在限制。SSCLK 的频率必须小于或等于 PCLK0。

在主机模式时,主机在 SSPCLKOUT 的采样边沿进行采样(具体是上升沿还是下降沿取决于 SPH 和 SPO),在互补的另一边沿进行数据搬运,因此需要在上升沿和下降沿都进行操作,这要求 SSPCLKOUT 的最高频率不高于 SSPCLK 的 1/2。

在从机模式时,时钟来自片外主机的 SSPCLKIN,经过双同步和延时后侦测边沿。侦测 SSPCLKIN 边沿至少需要 3 个 SSPCLK。主机 SSPRXD 相对于 SSPCLKIN 的建立时间和维持时间须冗余,确保正确采样数据。为确保设备的正确操作,从机的 SSPCLK 须比 SSPCLKIN 的最高期望频率快 12 倍。

SSPCLK 选择的频率必须适应所需的位时钟速率范围。在这种情况下,最小 SSPCLK 频率与 SSPCLKOUT 最大频率之比从模式为 12,主模式为 2。

在主模式下生成最大比特率 1.8432Mbps,即 SSPCLK 的频率必须至少为 3.6864MHz。SSPCLK 频率为 3.6864MHz,SSPCPSR 寄存器必须编程为值 2,以及 SSPCR0 寄存器中的 SCR [7:0]字段必须编程为 0。

要在从机模式下使用最大比特率 1.8432Mbps,SSPCLK 的频率必须至少 22.12MHz。SSPCLK 频率为 22.12MHz,SSPCPSR 寄存器可以使用值 12 编程,SSPCR0 寄存器中的 SCR [7:0]字段可以是编程的值为 0。同样,SSPCLK 的最大频率比为 SSPCLKOUT 最小频率为 254 x 256。

SSPCLK 的最小频率由以下两个等式决定,应同时满足以下条件:

主机模式: $F_{SSPCLK(min)} \Rightarrow 2 \times F_{SSPCLKOUT(max)}$

从机模式: $F_{SSPCLK(min)} \Rightarrow 12 \times F_{SSPCLKIN(max)}$

SSPCLK 的最大频率由以下两个等式决定,应同时满足以下条件:

主机模式: $F_{SSPCLK(max)} \leq 254 \times 256 \times F_{SSPCLKOUT(min)}$

从机模式: $F_{SSPCLK(max)} \leq 254 \times 256 \times F_{SSPCLKIN(min)}$

25.4.2 配置 SSPCR0 控制寄存器

SSPCR0 寄存器用于:

- 配置串行时钟速率
- 选择协议 (3 选 1)
- 选择帧数据长度

串行时钟速率 (SCR) 值,与 SSPCPSR 时钟预分频分频器 CPSDVSR 一起使用,用于从外部 SSPCLK 获取 SSP 发送和接收比特率。

数据帧格式通过 FRF 位选择,数据长度通过 DSS 位配置。

25.4.2.1 帧格式

每个数据帧的长度在 4-16 位之间,具体取决于编程数据的大小。从 MSB 开始传输,可以选择以下基本帧类型:

- 德州仪器 SSI
- 摩托罗拉 SPI
- 美国国家半导体 Microwire

对于所有的格式,SSP 在空闲状态下,SSPCLKOUT 维持无效状态,仅在有效的发送或接收数据期间产生时钟。SSPCLKOUT 的空闲状态用于提供接收超时指示,即当超时周期后接收 FIFO 仍包含数据。

对于 Motorola SPI 和 National Semiconductor Microwire 帧格式,串行帧,SSPFSSOUT 引脚为低电平有效,并在整个帧传输过程中被下拉。

对于德州仪器 (TI) 同步串行帧格式,SSPFSSOUT 引脚脉冲为 1 个串行时钟周期,从每个帧传输之前的上升沿开始。对于这帧格式,SSP 和片外从器件都在 SSPCLKOUT 的上升沿驱动其输出数据,并在下降沿锁存来自另一个器件的数据。

与其他两种帧格式的全双工传输不同,美国国家半导体 Microwire 格式使用特殊的主-从消息传递技术,以半双工方式运行。在此模式下,当帧开始时,将 8 位控制消息发送到片外从器件。在此传输期间,SSP 不

接收传入数据。消息发送后，并在 8 位控制的最后一位发送之后，等待一个串行时钟之后，片外从器件对其进行解码。消息已发送，以请求的数据响应。返回的数据可以是 4~16 位长度，使总帧长度在 13~25 位范围内。

25.4.2.2 德州仪器 SSI 帧格式

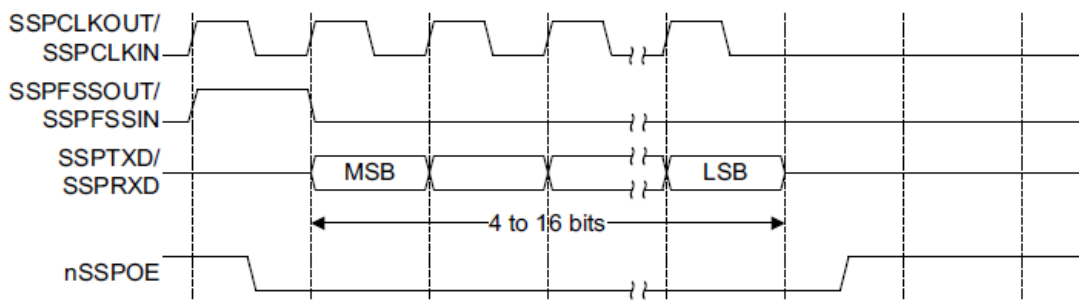


图 25-2 单字节传输时序图

在此模式下，SSPCLKOUT 和 SSPFSSOUT 强制拉低，当 SSP 处于空闲态时，发送数据线 SSPTXD 为三态。当发送 FIFO 的底部入口包含数据，SSPFSSOUT 被拉成一个 SSPCLKOUT 周期的高电平。将要发送的数据经传输逻辑从发送 FIFO 移到到串行移位寄存器内。在 SSPCLKOUT 的下一个上升沿，4~16 位数据帧的 MSB 位在 SSPTXD 引脚上移出。以类似的方式，接收数据的 MSB 是由片外串行从器件移入 SSPRXD 引脚。

SSP 和片外串行从器件在每个 SSPCLKOUT 下降沿将每个数据位移入串行移位器。在接收数据帧的最后一位锁存后，数据在 PCLK 的第一个上升沿存到接收 FIFO。

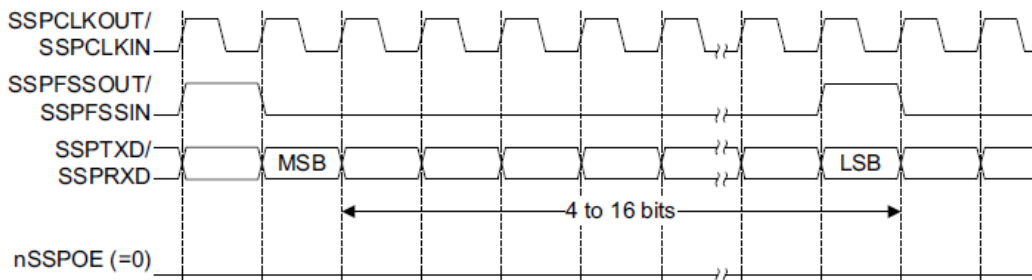


图 25-3 多字节连续传输时序图

25.4.2.3 摩托罗拉 SPI 帧格式

摩托罗拉 SPI 接口是一个四线接口，SSPFSSOUT 信号作为从机选择信号。摩托罗拉 SPI 是全双工通信协议，主要特点是用户可配置 SSPSCR0 的 SPO 和 SPH 位改变在非活动状态下 SSPCLKOUT 信号的状态和相位。

- SPO，时钟的极性：当 SPO 时钟极性控制位为低电平时，它会在 SSPCLKOUT 引脚上产生稳态 LOW 值。如果 SPO 时钟极性控制位为高电平，则当数据未传输时设置 SSPCLKOUT 引脚上的稳态 HIGH 值。
- SPH，时钟的相位：SPH 控制位选择捕获数据的时钟边沿并使其能够改变状态。当 SPH 相位控制位为低电平时，在第一个时钟边沿转换时捕获数据。当 SPH 时钟相位控制位为高电平时，在第二个时钟边沿捕获数据。

25.4.2.3.1 摩托罗拉 SPI 格式，SPO = 0，SPH = 0

在该配置下：

- SSPCLKOUT 信号强制为低电平
- SSPFSSOUT 信号强制为高电平
- 发送数据线 SSPTXD 被强制为低电平
- nSSPOE 引脚使能信号强制为高电平，使发送引脚具有高阻抗
- 当 SSP 配置为主机时，nSSPCTLOE 线路被驱动为低电平，使能 SSPCLKOUT 引脚，使能低电平有效
- 当 SSP 配置为从机时，nSSPCTLOE 线路被驱动为高电平，禁用 SSPCLKOUT 引脚，使能低电平有效

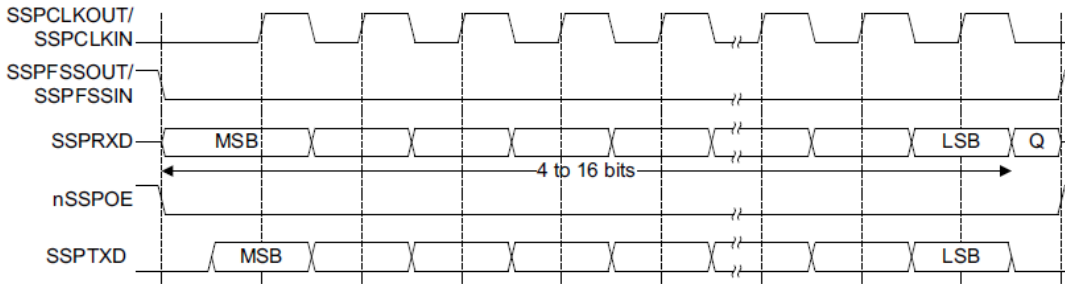


图 25-4 SPO=0, SPH=0, 单字节传输时序

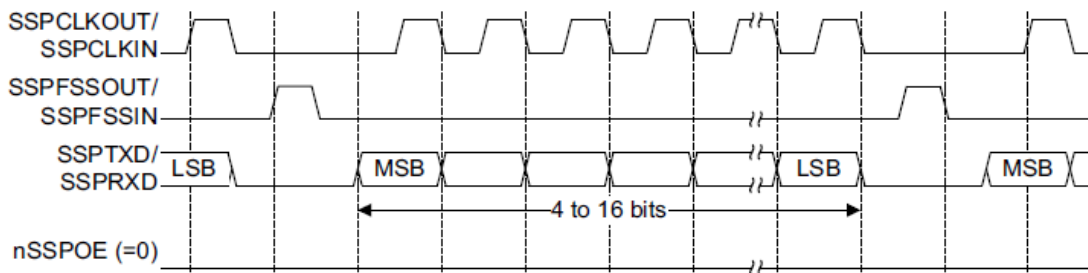


图 25-5 SPO=0, SPH=0, 多字节传输时序

如果 SSP 使能，且发送 FIFO 内含有有效数据，SSPFSSOUT 主信号被驱动为低电平表示传输开始。这使能从机的 SSPTXD 数据输入线。nSSPOE 线被驱动 LOW，使能主机 SSPTXD 输出脚。半个 SSPCLKOUT 周期后，有效主机数据传输到 SSPTXD 引脚。既然主设备和从设备数据都已设置，经过半个 SSPCLKOUT 周期后，SSPCLKOUT 主时钟引脚变为高电平。

现在，数据在上升时捕获，并在 SSPCLKOUT 的下降沿传播信号。

在单字传输的情况下，在传输了所有数据字位之后，SSPFSSOUT 线在最后数据位捕获后，过一个 SSPCLKOUT 周期返回到空闲 HIGH 状态。但是，在连续背靠背传输的情况下，SSPFSSOUT 信号必须在每个数据字传输之间变为高电平。这是因为从选择引脚冻结其串行外围寄存器中的数据，并且如果 SPH 位为逻辑零，则不允许改变它。因此，主设备必须在每次数据传输之间提升从设备的 SSPFSSIN 引脚，以启用串行外围数据写入。完成后连续传输后，SSPFSSOUT 引脚在最后一个 SSPCLKOUT 周期后返回空闲状态。

25.4.2.3.2 摩托罗拉 SPI 格式，SPO = 0，SPH = 1

在此配置中，在空闲期间：

- SSPCLKOUT 信号强制为低电平
- SSPFSSOUT 信号强制为高电平
- 发送数据线 SSPTXD 被强制为低电平
- nSSPOE 引脚使能信号强制为高电平，使发送引脚具有高阻抗

- 当 SSP 配置为主机时，nSSPCTLOE 线路被驱动为低电平，使能 SSPCLKOUT 引脚，使能低电平有效
- 当 SSP 配置为从机时，nSSPCTLOE 线路被驱动为高电平，禁用 SSPCLKOUT 引脚，使能低电平有效

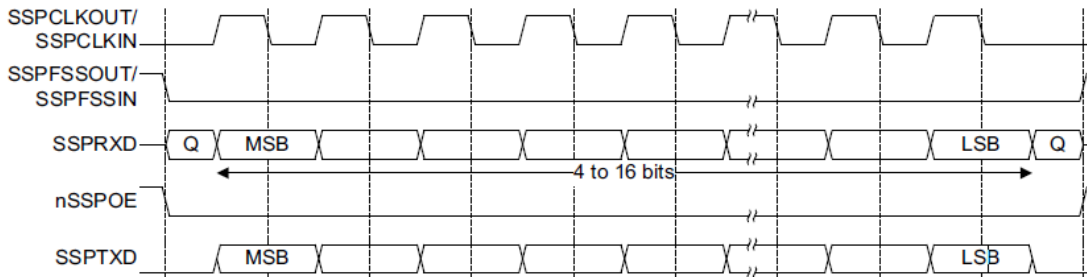


图 25-6 摩托罗拉 SPI 格式的传输信号序列 SPO = 0, SPH = 1, 包括单次和连续传输

如果使能了 SSP，并且发送 FIFO 中有有效数据，则 SSPFSSOUT 主信号被驱动为低电平表示传输启动。nSSPOE 线路被驱动为低电平，使能主 SSPTXD 输出。再过半个 SSPCLKOUT 周期，主机和从机各自的有效数据都被放到各自的传输线上。同时，SSPCLKOUT 通过上升沿转换使能。然后在 SSPCLKOUT 的下降沿捕获数据，并在上升沿传播数据。

在单字传输的情况下，在传输所有位之后，SSPFSSOUT 线在最后一个数据位被捕获后，再经过一个 SSPCLKOUT 周期，返回空闲高电平状态。

对于连续的对背对背传输，SSPFSSOUT 引脚在连续传输之间保持低电平，终止与单字传输相同。

25.4.2.3.3 摩托罗拉 SPI 格式，SPO = 1, SPH = 0

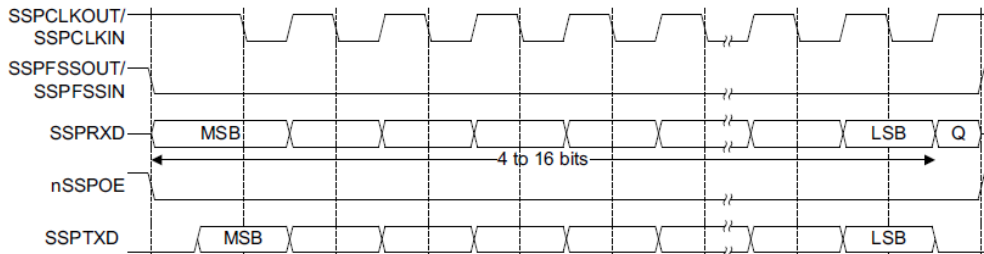


图 25-7 摩托罗拉 SPI 格式，SPO = 1, SPH = 0 单字节传输

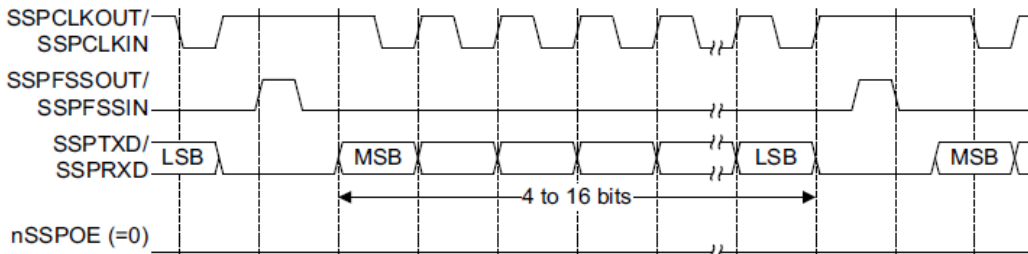


图 25-8 摩托罗拉 SPI 格式，SPO = 1, SPH = 0 多字节传输

在此配置中，在空闲期间：

- SSPCLKOUT 信号强制为高电平
- SSPFSSOUT 信号强制为高电平

- 发送数据线 SSPTXD 被强制为低电平
- nSSPOE 引脚使能信号强制为高电平，使发送引脚具有高阻抗
- 当 SSP 配置为主机时，nSSPCTLOE 线路被驱动为低电平，使能 SSPCLKOUT 引脚，使能低电平有效
- 当 SSP 配置为从机时，nSSPCTLOE 线路被驱动为高电平，禁用 SSPCLKOUT 引脚，使能低电平有效

如果启用了 SSP，并且发送 FIFO 中有有效数据，则启动 SSPFSSOUT 主信号被驱动为低电平表示传输，这会导致从机数据立即传输到主机 SSPTXD 上。nSSPOE 线被驱动为低电平，使能主机 SSPTXD 输出。

半个周期后，有效的主机数据被传输到 SSPTXD 线。既然这两个主机和从机数据都已设置，SSPCLKOUT 主时钟引脚在半个 SSPCLKOUT 周期后变为低电平。这意味着然后在 SSPCLKOUT 的下降沿捕获数据，并在上升沿传播数据。

在单字传输的情况下，在传输数据字的所有位之后，SSPFSSOUT 线在 SSPCLKOUT 周期后返回到空闲 HIGH 状态。

但是，在连续背靠背传输的情况下，SSPFSSOUT 信号必须在每个数据字传输之间变为高电平。这是因为从选择引脚冻结。如果 SPH 位为逻辑零，则串行外设寄存器中的数据不允许更改零。因此，主器件必须在两者之间拉为高电平。SSPFSSIN 引脚每次数据传输都能使串行外设数据写入。完成后连续传输时，SSPFSSOUT 引脚在最后一个 SSPCLKOUT 周期后返回空闲状态。

25.4.2.3.4 摩托罗拉 SPI 格式，SPO = 1，SPH = 1

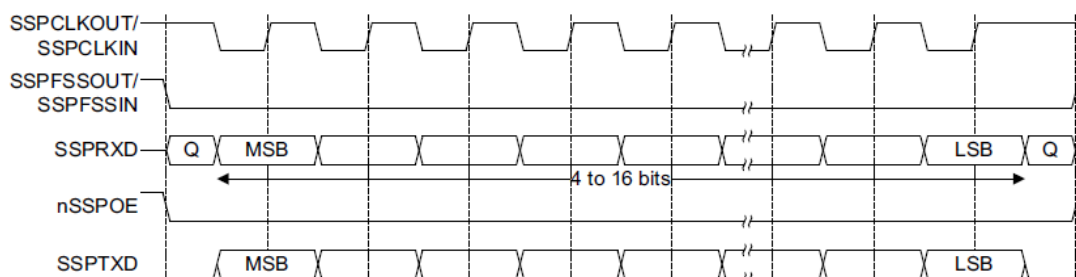


图 25-9 摩托罗拉 SPI 格式的传输信号序列 SPO = 1，SPH = 1，包括单次和连续传输

在此配置中，在空闲期间：

- SSPCLKOUT 信号强制为高电平
- SSPFSSOUT 信号强制为高电平
- 发送数据线 SSPTXD 被任意强制为低电平
- nSSPOE 引脚使能信号强制为高电平，使发送引脚具有高阻抗
- 当 SSP 配置为主机时，nSSPCTLOE 线路被驱动为低电平，使能 SSPCLKOUT 引脚，使能低电平有效
- 当 SSP 配置为从机时，nSSPCTLOE 线路被驱动为高电平，禁用 SSPCLKOUT 引脚，使能低电平有效

如果启用了 SSP，并且发送 FIFO 中有有效数据，则启动 SSPFSSOUT 主信号被驱动为低电平表示传输。nSSPOE 线路被驱动为低电平，使能主 SSPTXD 输出引脚。再过半个 SSPCLKOUT 周期，主数据和从数据都被启用到它们各自的传输上线。同时，SSPCLKOUT 通过下降沿转换启用。然后数据在上升沿捕获并在 SSPCLKOUT 信号的下降沿传播。在传输所有位之后，在单字传输的情况下，SSPFSSOUT 在最后一位捕获并经过一个 SSPCLKOUT 周期后，返回到空闲高电平状态。

对于连续的背靠背传输，SSPFSSOUT 引脚保持低电平有效状态，直到最后一个字的最后一位被捕获，然后返回空闲状态。对于连续的对背传输，SSPFSSOUT 引脚在连续传输之间保持低电平，终止与单字传输相同。

25.4.2.4 美国国家半导体 Microwire 帧格式

Microwire 格式与 SPI 格式非常相似，除了该模式下传输属于半双工，使用主从消息传递技术。SSP 每次向片外从机发送的操作，都以 8 位控制字节开头。在发送期间，SSP 不接收数据。待发送完毕，片外从机译码并在最后数据位发送完成后，等待一个串行时钟，并回馈数据。返回的数据长度为 4~16 位，帧长度为 13~25 位。

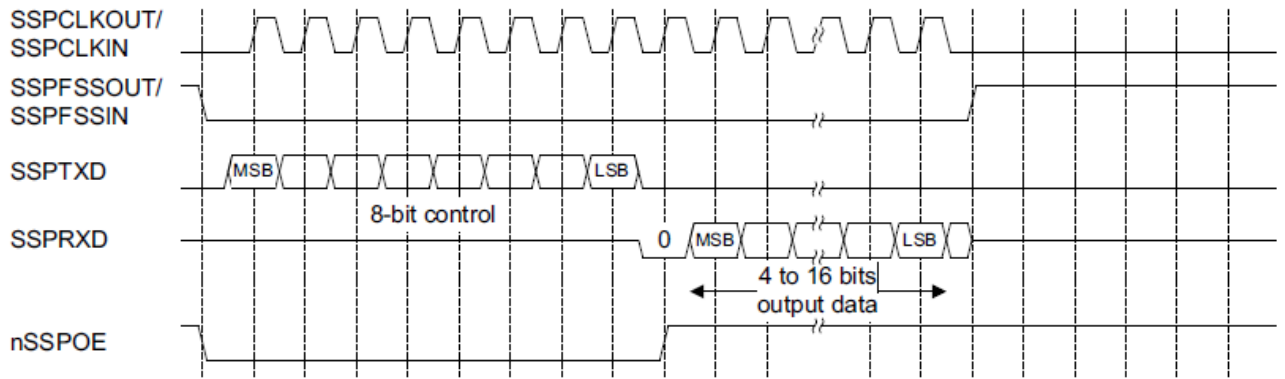


图 25-10 美国国家半导体 Microwire 帧格式，单次传输

在此配置中，在空闲期间：

- SSPCLKOUT 被强制为低电平
- SSPPSSOUT 强制为高电平
- 发送数据线 SSPTXD 被强制为低电平
- nSSPOE 焊盘使能信号强制为高电平，使发送引脚具有高阻抗

对发送 FIFO 写入控制字节触发数据传输。在 SSPPSSOUT 下降沿，发送 FIFO 的底部数据送到移位寄存器，8 位控制帧的 MSB 位移出到 SSPTXD 引脚。SSPPSSOUT 在数据传输期间维持低电平。SSPRXD 引脚在数据传输期间维持三态。

片外串行从机在每个 SSPCLKOUT 的上升沿将控制位锁存进移位寄存器中。待从机锁存最后一位，控制字节在一个时钟的等待期内解码，并发送数据响应 SSP。每个数据位在 SSPCLKOUT 的下降沿被驱动到 SSPRXD 线上。SSP 反过来在 SSPCLKOUT 的上升沿锁存数据的每一位。对于单次传输，SSPPSSOUT 信号在完成最后数据位锁存的一个时钟周期后被拉高，数据传送到接收 FIFO。

注：片外从器件可以在 LSB 被接收移位器锁存后 SSPCLKOUT 的下降沿或 SSPPSSOUT 引脚变为高电平时，将接收线置为三态。

对于连续传输，数据传输以与单个传输相同的方式开始和结束传递。但是，SSPPSSOUT 线连续有效，保持低电平并进行背靠背传输数据。下一帧的控制字节紧跟在当前帧接收的数据的 LSB 之后。每个接收到的值都是从帧的 LSB 被锁存后，在下降沿 SSPCLKOUT 上接收移位器进入 SSP。

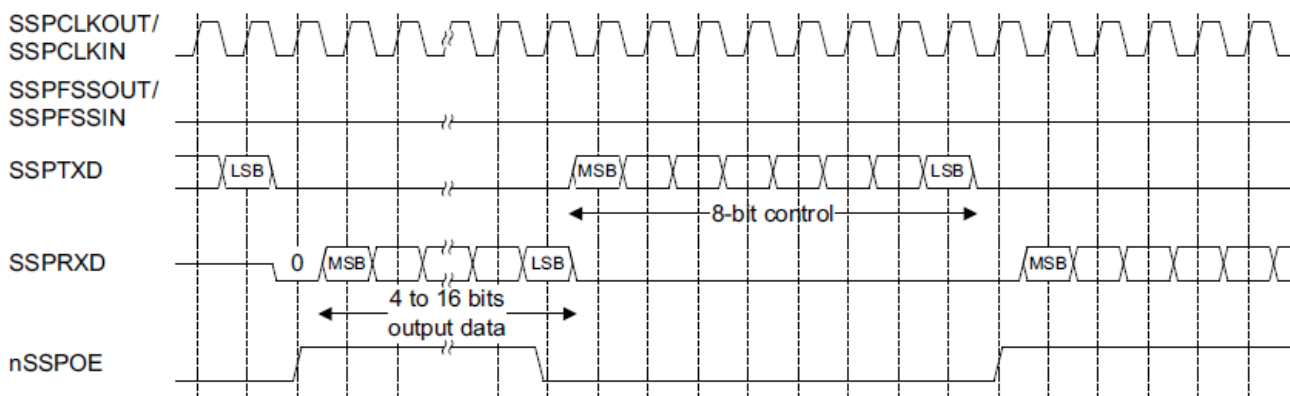


图 25-11 美国国家半导体 Microwire 帧格式，连续传输

在 Microwire 模式下，SSPFSSIN 相对于 SSPCLKIN 有建立和保持时间的要求。

在 Microwire 模式下，SSP 从器件在 SSPFSSIN 变低后的 SSPCLKIN 上升沿采样接收数据的第 1 位。主机驱动 SSPCLKIN，必须确保 SSPFSSIN 有足够的建立时间和保持时间余量。

在 SSP 采样第一个接收数据位时，相较于 SSPCLKIN，SSPFSSIN 的建立时间至少为 SSPCLK 时钟周期的 2 倍。相较于 SSPCLKIN，SSPFSSIN 的保持时间至少为 SSPCLK 时钟周期的 1 倍。

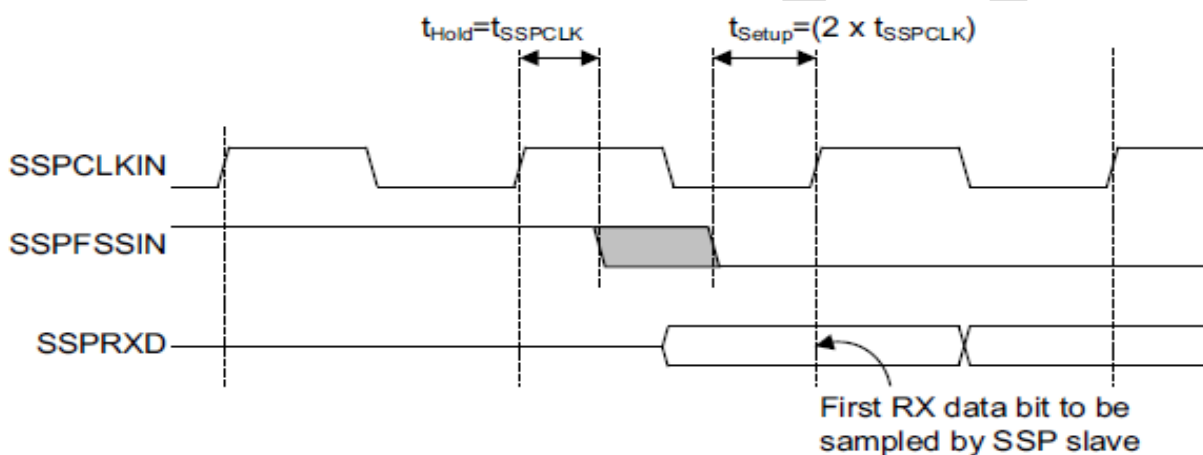


图 25-12 Microwire 帧格式，SSPFSSIN 输入建立时间和保持时间要求

注：该 SSP 不支持在系统中动态切换主机和从机模式。

图 25-13 显示 SSP 一个做从机一个做主机。主机可通过 SSPTXD 访问从机。作为回应，从机驱动 nSSPOE 为高电平，使能从机 SSPTXD 发送数据到主机的 SSPRXD。

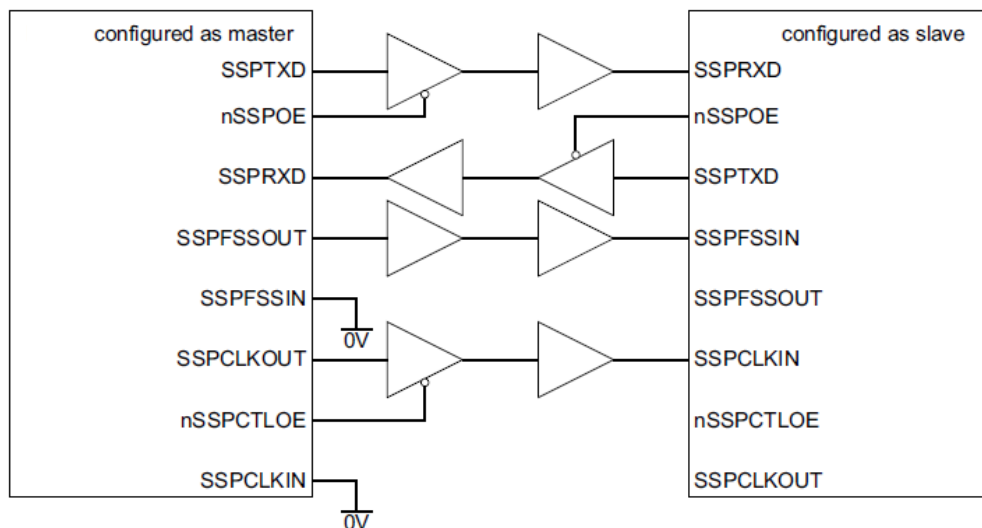


图 25-13 SSP 主机与 SSP 从机

图 25-14 显示 SSP 配置成主机，与外部 SPI 从机相连。SPI 的 SS 信号固定成 0，配置成从机。与上述操作相似，主机通过 SSP SSPTXD 访问从机，从机驱动 SPI MISO 端口到主机的 SSPRXD 线。

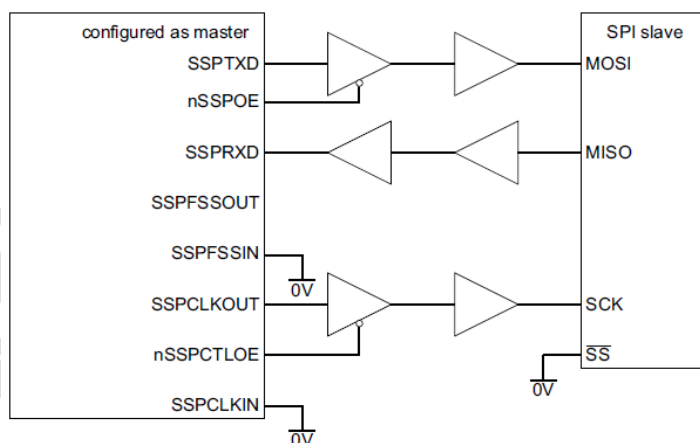


图 25-14 SSP 主机和 SPI 从机

图 25-15 显示外部 SPI 配置成主机，与配成从机的 SSP 相连。在这种情形下，SS 固定成高电平，从机的 SSPFSSIN 接地。主机通过 MOSI 访问从机，从机驱动 nSSPOE 为低电平，这使能 SSPTXD 数据向主机的 MISO 线发送。

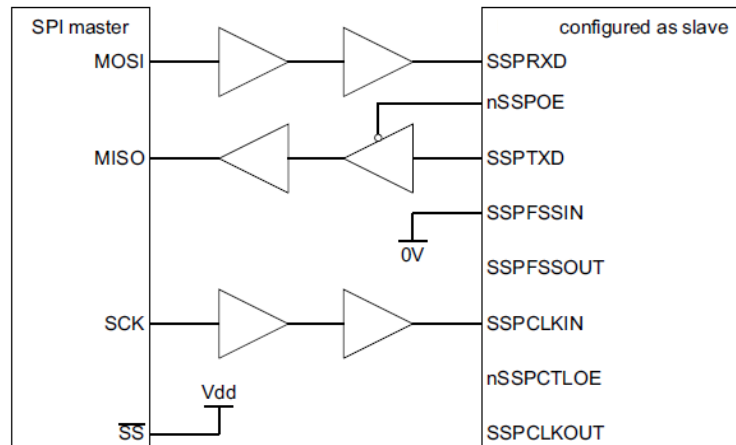


图 25-15 SPI 主机与 SSP 从机

25.4.3 配置 SSPCR1 控制寄存器

SSPCR1 寄存器用于：

- 主机从机选择
- 使能回环测试
- 使能 SSP

为配置主机模式，设置 SSPCR1 寄存器主从机选择位 MS 为 0，即复位值。

为配置从机模式，设置 SSPCR1 寄存器主从机选择位 MS 为 1。当 SSP 配置为从机时，SSPTXD 信号使能与否取决于 SSPCR1 的 SOD 位。这可用在并行广播的多从机环境。

为使能 SSP，置位 SSE。

25.4.4 DMA 接口

SSP 提供一个接口，连接到 DMA 控制器。DMA 控制寄存器控制 SSP 的 DMA 操作。

接收：

对于接收，DMA 接口包含以下信号（括号内为对应 DMA 控制器的信号）：

- SSPRXDMAREQ (REQ 信号)：单字节 DMA 传输请求，由 SSP 置位。当接收 FIFO 含至少 1 个字节该信号置位。
- SSPRXDABREQ (REQ 信号)：连续 DMA 传输请求，由 SSP 置位。当接收 FIFO 含大于等于 4 个数据时该信号置位。
- SSPRXDACLK (ACK 信号)：DMA 请求清除，DMA 控制器置位用于清除 SSP 接收请求信号。如果 DMA 连续传输请求，在传输的最后一个数据置位清除信号。

发送：

对于发送，DMA 接口包含以下信号：

- SSPTXDMAREQ：单字节 DMA 传输请求，由 SSP 置位。在发送 FIFO 至少含一个空槽时该信号置位。
- SSPTXDABREQ：连续 DMA 传输请求，由 SSP 置位。当发送 FIFO 包含小于等于 4 个数据时该信号置位。
- SSPTXDACLK：DMA 请求清除，DMA 控制器置位，清除发送请求信号。如果 DMA 连续传输请求发生，清除信号在最后一个数据后置位。

表 25-2 DMABREQ 的触发点，用于发送和接收 FIFO

触发基准水平	发送（FIFO 空闲数据数）	接收（FIFO 包含数据数）
1/2	4	4

连续传输请求和单个传输请求信号并不互斥。可同时置位。例如，当接收 FIFO 中数据的个数超过 4，这两个请求信号均置位。当接收 FIFO 内包含的数据个数少于基准水平，只置位单个请求信号。

例如，必须要收 19 个字符，DMA 控制器发起 4 次 4 字节的连续读操作，3 次单字节读操作，完成整个任务。

注：对于剩余的 3 个字节，SSP 不会置位连续读请求信号。

每个请求信号保持有效，直到相关的 DMA 清除信号被置位。之后请求清除信号置为无效，请求信号可以再次激活，具体取决于前面部分描述的条件。如果 SSP 使能禁止或 DMA 使能信号禁止，所有请求信号都被置为无效。

图 25-16 显示了单个传输请求和突发传输请求的时序，具有适当的 DMA 清除信号。信号都与 PCLK 同步。

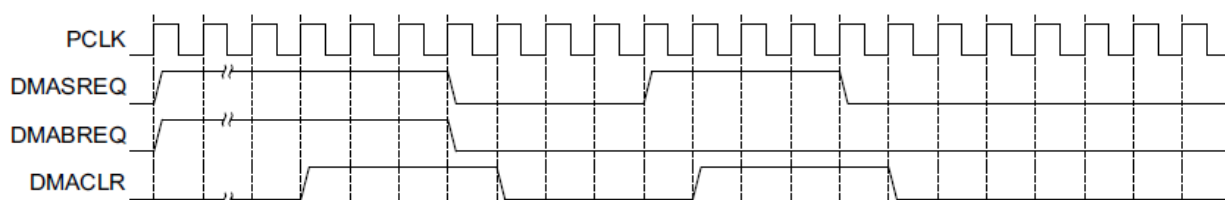


图 25-16 DMA 传输波形

25.4.5 SSP 中断

SSP 有 4 个可屏蔽中断源，这些独立的中断经过或逻辑产生 1 个中断输出 SSPINTR，对应着系统中的 SSP 全局中断向量。

- SSPRXINTR SSP 接收中断
- SSPTXINTR SSP 发送中断
- SSPRORINTR SSP 接收 FIFO 溢出中断
- SSPRTINTR SSP 接收 FIFO 超时中断

用户可以通过设置 SSPIMISC 中对应的位来分别屏蔽四个单独的可屏蔽中断，将相应的屏蔽位设置为高可启用中断。发送和接收动态数据流中断 SSPTXINTR 和 SSPRXINTR 与状态中断分离，因此可以仅由 FIFO 触发进行数据的读取和写入。

可以从 SSPRIS 和 SSPMIS 寄存器中读取各个中断源的状态。

- SSPRXINTR
当接收 FIFO 中有 4 个或更多有效数据时，接收中断被置位。
- SSPTXINTR
当发送 FIFO 中有四个或更少的有效数据时，发送中断被置位。发送中断 SSPTXINTR 因 SSP 使能信号无效时，可以通过以下任一方式进行操作：
 - 在使能 SSP 和中断前，可以将数据写入发送 FIFO 中
 - 可以启用使能 SSP 和中断，以便可以在中断服务程序将数据写入发送 FIFO 中
- SSPRORINTR
当 FIFO 已满，且又收到额外的一帧数据，产生接收溢出中断 SSPRORINTR，FIFO 溢出。新数据覆盖接收移位寄存器的内容，但接收 FIFO 内容不变。通过写入 SSPICR 寄存器中的 RORIC 位可以清除该中断。

- SSPRTINTR

当接收 FIFO 非空且 SSP 在 32 个位周期内保持空闲状态时，接收超时中断被置位。该机制确保用户知道数据仍然存在接收 FIFO 中并需要服务。当通过连续读操作，接收 FIFO 变为空，或者在 SSPRXD 上接收到新数据，该中断清除。此外，通过写入 SSPICR 寄存器中的 RTIC 位也可以清除该中断。

- SSPINTR

以上四个独立中断组合成单个输出 SSPINTR，即，以上中断源（经屏蔽后）经或逻辑产生的中断。如果上面的四个中断中的任何一个中断有效，则组合的 SSP 中断被置位。

25.5 SSP 寄存器描述

SSP0 的基地址为 0X4001_8000，SSP2 的基地址为 0x4000_6000，SSP0 和 SSP2 的地址偏移分别对应各自的基地址。

表 25-3 SSP 相关寄存器表

名称	说明	读写权限	复位值	地址偏移
SSPCR0	SSP 控制寄存器 0	R/W	0x0000_0000	0x00
SSPCR1	SSP 控制寄存器 1	R/W	0x0000_0000	0x04
SSPDR	SSP 数据寄存器	R/W	0x0000_0000	0x08
SSPSR	SSP 状态寄存器	R	0x0000_0003	0x0C
SSPCPSR	SSP 时钟分频寄存器	R/W	0x0000_0000	0x10
SSPIMSC	SSP 中断屏蔽寄存器	R/W	0x0000_0000	0x14
SSPRIS	SSP 原始中断状态寄存器	R	0x0000_0000	0x18
SSPMIS	SSP 屏蔽后中断状态寄存器	R	0x0000_0000	0x1C
SSPICR	SSP 中断清除寄存器	W	0x0000_0000	0x20
SSPDMACR	SSPDMA 控制寄存器	R/W	0x0000_0000	0x24

注：x 表示不确定；-表示未实现；u 表示不受复位影响；c 表示该位复位值因复位源而异；R 只读；W 只写；R-0 表示只读且读 0；R/W 表示可读写（以后章节同上述）。

25.5.1 SSP0/2 控制寄存器 0 (SSPCR0)

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	
Res.																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
SCR[7:0]								SPH	SPO	FRF[1:0]			DSS[3:0]			
rw								rw	rw	rw			rw			

Bits 31:16	保留，必须保持复位值
Bits 15:8	SCR[7:0] : 串行时钟速率（默认为 0）
	SCR 用于生成传输并接收 SSP 的比特率，比特率是：
	$\frac{F_{sspclk}}{CPSDVR \times (1 + SCR)}$
	其中 CPSDVR 是 2-254 的偶数值，通过 SSPCPSR 寄存器编程，SCR 是 0-255 的值

Bit 7	SPH: SSPCLKOUT 相位, 只适用于 Motorola SPI 帧格式
	0: SSP 控制器在帧传输的第一个时钟跳变沿捕获串行数据 (默认)
	1: SSP 控制器在帧传输的第二个时钟跳变沿捕获串行数据
Bit 6	SPO: SSPCLKOUT 极性, 只适用于 Motorola SPI 帧格式
	0: SSP 控制器使总线时钟在每帧传输之间保持低电平 (默认)
	1: SSP 控制器使总线时钟在每帧传输之间保持高电平
Bits 5:4	FRF: 帧格式
	00: Motorola SPI 帧格式 (默认)
	01: TI 同步串行帧格式 (SSI)
	10: National Microwire 帧格式
	11: 保留
Bits 3:0	DSS: 数据长度选择, 该字段控制着每帧传输的位数
	0000~0010: 保留 (默认)
	0011: 4 位数据
	0100: 5 位数据
	0101: 6 位数据
	0110: 7 位数据
	0111: 8 位数据
	1000: 9 位数据
	1001: 10 位数据
	1010: 11 位数据
	1011: 12 位数据
	1100: 13 位数据
	1101: 14 位数据
	1110: 15 位数据
	1111: 16 位数据

25.5.2 SSP0/2 控制寄存器 1 (SSPCR1)

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
Res.															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.												SOD	MS	SSE	LBM
												rw	rw	rw	rw

Bits 31:4	保留, 必须保持复位值
Bit 3	SOD: 从机模式输出使能禁止, 该位只与从机模式有关 (MS=1)
	0: 允许 SSP 控制器驱动发送数据线 TXD (默认)
	1: 禁止 SSP 控制器驱动发送数据线 TXD
Bit 2	MS: 主从模式选择, 该位只能在 SSP 禁止 (SSE=0) 时配置

	0: 主机模式, 驱动 SCK、TXD 和 FSS 并接收 RXD 线 (默认)
	1: 从机模式, SSP 控制器用作总线从机, 驱动 TXD 线和接收 SCK、RXD 和 FSS 管脚
Bit 1	SSE: 同步串行端口使能
	0: SSP 操作禁止 (默认)
	1: SSP 操作使能
	SSP 控制器可与串行总线上的其它器件相互通信。在置位该位前, 软件应将合适的控制器信息写入其它 SSP 寄存器和中断寄存器
Bit 0	LBM: 回写模式
	0: 正常工作模式 (默认)
	1: 正常工作模式串行输入脚可用作串行输出脚 (MOSI 或 MISO), 而不是仅用作串行输入脚 (MISO 或 MOSI 分别起作用)

25.5.3 SSP0/2 数据寄存器 (SSPDR)

地址偏移: 0x08

复位值: 0x0000_0000

写: 当状态寄存器的 TNF 位为 1 指示 Tx FIFO 未滿时, 软件可将要发送的帧数据写入该寄存器。如果 Tx FIFO 以前为空且 SSP 控制器空闲, 则立即开始发送数据。否则, 写入该寄存器的数据要等到所有数据发送 (或接收) 完后才能发送。如果数据长度小于 16 位, 软件必须对数据进行调整后再写入该寄存器。

读: 当状态寄存器的 RNE 位为 1 指示 Rx FIFO 不为空时, 软件可读取该寄存器。软件读取该寄存器时, SSP 控制器将返回 Rx FIFO 中的最早收到的一帧数据。如果数据长度小于 16 位, 该字段的数据必须进行合适的调整, 低位对齐, 高位补零。

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

Bits 31:16	保留, 必须保持复位值
Bits 15:0	DATA[15:0]: 发送/接收 FIFO (默认为 0)
	读: 接收 FIFO
	写: 发送 FIFO

25.5.4 SSP0/2 状态寄存器 (SSPSR)

地址偏移: 0x0C

复位值: 0x0000_0003

SSPSR 是只读状态寄存器, 包含指示 FIFO 填充的位状态和 SSP 忙状态。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.											BSY	RFF	RNE	TNF	TFE
											r	r	r	r	r

Bits 31:5	保留，必须保持复位值
Bit 4	BSY : SSP 忙标志
	0: SSP 为空闲状态（默认）
	1: SSP 当前正在发送/接收，或发送帧 FIFO 不为空
Bit 3	RFF : 接收 FIFO 满
	0: 接收 FIFO 非满（默认）
	1: 接收 FIFO 满
Bit 2	RNE : 接收 FIFO 非空
	0: 接收 FIFO 为空（默认）
	1: 接收 FIFO 非空
Bit 1	TNF : 发送 FIFO 非满
	0: 发送 FIFO 为满
	1: 发送 FIFO 非满（默认）
Bit 0	TFE : 发送 FIFO 为空
	0: 发送 FIFO 非空
	1: 发送 FIFO 为空（默认）

25.5.5 SSP0/2 时钟分频寄存器 (SSPCPSR)

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

Bits 31:8	保留，必须保持复位值
Bits 7:0	CPSDVSR[7:0] : 时钟分频寄存器（默认为 0）
	这是 2~254 中的一个偶数值。它是 SSP_PCLK 的分频因子，SSP_PCLK 通过分频后得到预分频器输出时钟。CPSDVSR[0]读出时总是为 0。
注: 必须适当初始化 SSPnCPSR 值，否则 SSP 控制器将无法正确发送数据。	
<ul style="list-style-type: none"> ● 在从机模式下，主机提供的 SSP 时钟速率不能大于 SSP 外设时钟的 1/12。 ● 在主机模式下，CPSDVSR_{min}=2 或更大的值（只能为偶数）。 	

25.5.6 SSP0/2 中断屏蔽寄存器 (SSPIMSC)

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
Res.															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.												TXIM	RXIM	RTIM	RORIM
												rw	rw	rw	rw

Bits 31:4	保留，必须保持复位值
Bit 3	TXIM: 软件置位该位，允许当 Tx FIFO 至少有一半为空时触发中断（默认为 0）
	0: 屏蔽（默认）
	1: 不屏蔽
Bit 2	RXIM: 软件置位该位，允许当 Rx FIFO 至少有一半为满时触发中断（默认为 0）
	0: 屏蔽（默认）
	1: 不屏蔽
Bit 1	RTIM: 软件置位该位，允许接收超时中断（默认为 0）
	当 Rx FIFO 不为空且在 32 个位时间内既没有接收到新数据有没有从 FIFO 中读出数据时，产生接收超时
	0: 屏蔽（默认）
	1: 不屏蔽
Bit 0	RORIM: 软件置位该位，允许接收溢出中断（默认为 0）
	当 Rx FIFO 满时又完成另一个帧的接收时该位置位。特别指出，发生接收溢出时新数据帧会将前面的数据帧覆盖
	0: 屏蔽（默认）
	1: 不屏蔽

25.5.7 SSP0/2 原始中断状态寄存器（SSPRIS）

地址偏移：0x18

复位值：0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.												TXRIS	RXRIS	RTRIS	RORRIS
												r	r	r	r

Bits 31:4	保留，必须保持复位值
Bit 3	TXRIS: 当 Tx FIFO 至少有一半为空时发生中断
	0: 中断未发生（默认）
	1: 中断发生
Bit 2	RXRIS: 当 Rx FIFO 至少有一半为满时发生中断
	0: 中断未发生（默认）

	1: 中断发生
Bit 1	RTRIS: 如果 Rx FIFO 不为空, 且在“超时周期”中没有被读出时发生中断
	0: 中断未发生 (默认)
	1: 中断发生
Bit 0	RORRIS: 当 Rx FIFO 满时又接收到另一帧数据时发生中断
	0: 中断未发生 (默认)
	1: 中断发生
	注: 发生接收溢出时新数据帧会将前面的数据帧覆盖

25.5.8 SSP0/2 屏蔽后中断状态寄存器 (SSPMIS)

地址偏移: 0x1C

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.												TXMIS	RXMIS	RTMIS	RORMIS
												r	r	r	r

Bits 31:4	保留, 必须保持复位值
Bit 3	TXMIS: 当 Tx FIFO 至少有一半为空, 且中断未屏蔽时发生中断
	0: 中断未发生或被屏蔽 (默认)
	1: 中断未被屏蔽
Bit 2	RXMIS: 当 Rx FIFO 至少有一半为满, 且中断未屏蔽时发生中断
	0: 中断未发生或被屏蔽 (默认)
	1: 中断未被屏蔽
Bit 1	RTMIS: 如果 Rx FIFO 不为空, 且在“超时周期”中没有被读出, 且中断未屏蔽时发生中断
	0: 中断未发生或被屏蔽 (默认)
	1: 中断未被屏蔽
Bit 0	RORMIS: 当 Rx FIFO 满时又接收到另一帧数据, 且中断未屏蔽时发生中断
	0: 中断未发生或被屏蔽 (默认)
	1: 中断未被屏蔽
	注: 发生接收溢出时新数据帧会将前面的数据帧覆盖

25.5.9 SSP0/2 中断清除寄存器 (SSPICR)

地址偏移: 0x20

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.														RTIC	RORIC
														w	w

Bits 31:2	保留，必须保持复位值
Bit 1	RTIC : 接收超时中断清除位
Bit 0	RORIC : 接收溢出中断清除位
注 : 该寄存器只能进行写操作，写1清除中断标志位，写0无效。	

25.5.10 SSP0/2 DMA 控制寄存器 (SSPDMACR)

地址偏移: 0x24

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.														TXDMAE	RXDMAE
														rw	rw

Bits 31:2	保留，必须保持复位值
Bit 1	TXDMAE : 发送 FIFO 的 DMA 访问使能
	0: 发送 DMA 访问禁止 (默认)
	1: 发送 DMA 访问使能
Bit 0	RXDMAE : 接收 FIFO 的 DMA 访问使能
	0: 接收 DMA 访问禁止 (默认)
	1: 接收 DMA 访问使能

26.CRC 模块

26.1 CRC 主要特性

- 支持以下 4 个多项式：
 - CRC8 0x07 生成多项式
 - CRC16 0x8005 生成多项式
 - CRC16 0x1021 生成多项式
 - CRC32 0x04C11DB7 生成多项式
- 每个周期完成一个字节的 CRC 运算
- CRC 的初始值可以设置

26.2 功能描述

CRC（循环冗余校验）计算单元使用一个可配置的多项式发生器，从一个 8 位/16 位/32 位的数据字产生一个 CRC 码。

众多的应用中，基于 CRC 的技术被用于验证数据传输或存储的一致性。在 EN/IEC 60335-1 标准的范围内，它提供了一种检测闪存存储器错误的手段，CRC 计算单元可以用于实时地计算软件的签名，并与在链接和生成该软件时产生的签名对比。

26.3 CRC 寄存器描述

CRC 模块归属于总线 AHB1 管理，此空间的基地址为 0x4801_1000。

表 26-1 CRC 相关寄存器表

名称	说明	读写权限	复位值	字节地址
CRC_CTRL	CRC 控制寄存器	R/W	0x0000_0000	0x4801_1000
CRC_DATA	CRC 数据寄存器	R/W	0x0000_0000	0x4801_1004
CRC_RESULT	CRC 结果寄存器	R/W	0x0000_0000	0x4801_1008

注：x 表示不确定；-表示未实现；u 表示不受复位影响；c 表示该位复位值因复位源而异；R 只读；W 只写；R-0 表示只读且读 0；R/W 表示可读写（以后章节同上述）。

26.3.1 CRC 控制寄存器（CRC_CTRL）

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	
Res.						CRC_GPS[1:0]		Res.						CRC_SWAP[1:0]		CRC_LMS
						rw								rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRC_INV_EN		Res.				CRC_BYTE[1:0]		Res.					CRC_VALID	CRC_SRST	CRC_SRSTALL
rw						rw							r	rw	rw

Bits	31:26	保留，必须保持复位值
Bits	25:24	CRC_GPS[1:0]：CRC 多项式选择

	00: 选择 CRC8, 0x07 生成多项式 (默认)
	01: 选择 CRC16, 0x8005 生成多项式
	10: 选择 CRC16, 0x1021 生成多项式
	11: 选择 CRC32, 0x04C11DB7 生成多项式
Bits 23:19	保留, 必须保持复位值
Bits 18:17	CRC_SWAP[1:0]: 待校验数据字节顺序转换
	00: 待校验数据字节排列不变 (默认)
	01: 待校验数据转换为{bit[23:16], bit[31:24], bit[7:0], bit[15:8]}
	10: 待校验数据转换为{bit[15:0], bit[31:16]}
	00: 待校验数据转换为{bit[7:0], bit[15:8], bit[23:16:], bit[31:24]}
Bit 16	CRC_LMS: CRC 大小端选择
	0: LSB, 低比特先运算 (默认)
	1: MSB, 高比特先运算
Bit 15	CRC_INTV_EN: CRC 运算结果字节大小端选择
	0: LSB 存入 CRC_RESULT 寄存器最低位 (默认)
	1: MSB 存入 CRC_RESULT 寄存器最低位
Bits 14:10	保留, 必须保持复位值
Bits 9:8	CRC_BYTE[1:0]: CRC 有效字节数
	00: 一个字节有效, 即 bit7-0 (默认)
	01: 两个字节有效, 即 bit15-0
	10: 三个字节有效, 即 bit23-0
	11: 四个字节有效, 即 bit31-0
Bits 7:3	保留, 必须保持复位值
Bit 2	CRC_VALID: CRC 有效结果标志位
	0: CRC 校验完毕, CRC_RESULT 数据有效 (默认)
	1: CRC 校验进行中, CRC_RESULT 数据尚未更新
Bit 1	CRC_SRST: CRC 结果复位寄存器
	0: 不复位 (默认)
	1: 复位 CRC 结果寄存器
Bit 0	CRC_SRSTALL: CRC 复位寄存器
	0: 不复位 (默认)
	1: 复位整个 CRC 相关逻辑, 一个周期后自动清零

26.3.2 CRC 数据寄存器 (CRC_DATA)

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
CRC_data[31:16]															
rw															

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

Bits 31:0	CRC_data[31:0]: 多少位有效请参考 CRC_BYTE 设置
	当 CRC_BYTE 为 00 时, bit7-0 有效; 为 01 时, bit15-0 有效; 为 10 时, bit23-0 有效; 为 11 时, bit31-0 有效

26.3.3 CRC 结果寄存器 (CRC_RESULT)

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
CRC_result[31:0]															
rw															

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

Bits 31:0	CRC_result[31:0]: 多少位有效请参考 CRC_GPS 设置
	当 CRC_GPS 为 00 时, bit7-0 有效; 为 01 或 10 时, bit15-0 有效, 为 11 时, bit31-0 有效

27.DIV 除法器

27.1 DIV 主要特性

- 支持 64/32 的有符号操作，32 个周期完成
- 向下支持 32/32 的有符号除法

27.2 功能描述

将数据写入除数寄存器 DIV_B 后，DIV 开始计算，在 32 个周期之后可读取计算结果。除法器操作步骤如下：

1. 置位 DIV_RST 位（DIV_CTRL 寄存器第 0 位），完成除法器逻辑复位
2. 配置 DIV32 位（DIV_CTRL 寄存器第 8 位），选择被除数的位宽 32/64
3. 将被除数按字写入 DIV_AHI 和 DIV_ALO 寄存器
4. 将除数按字写入 DIV_B 寄存器，启动除法运算
5. 等待 32 个时钟周期，分别读 DIV_QUOTHI 寄存器和 DIV_REM 寄存器获取商和余数的值

27.3 DIV 寄存器描述

DIV 除法器归属于总线 AHB1 管理，此空间的基地址为 0x4801_4000。

表 27-1 DIV 相关寄存器表

名称	说明	读写权限	复位值	字节地址
DIV_CTRL	控制寄存器	R/W	0x0000_0000	0x4801_4000
DIV_ALO	被除数低位寄存器	R/W	0x0000_0000	0x4801_4004
DIV_AHI	被除数高位寄存器	R/W	0x0000_0000	0x4801_4008
DIV_B	除数寄存器	R/W	0x0000_0001	0x4801_400C
DIV_QUOTLO	商低位寄存器	R	0x0000_0000	0x4801_4010
DIV_QUOTHI	商高位寄存器	R	0x0000_0000	0x4801_4014
DIV_REM	余数寄存器	R	0x0000_0000	0x4801_4018

注：x 表示不确定；-表示未实现；u 表示不受复位影响；c 表示该位复位值因复位源而异；R 只读；W 只写；R-0 表示只读且读 0；R/W 表示可读写（以后章节同上述）。

27.3.1 DIV 控制寄存器（DIV_CTRL）

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
Res.						DIV0_INTREN	DIV0	Res.							
						rw	r								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.							DIV32	Res.			DIV_B _WRITED	DIV_AHI_ WRITED	DIV_ALO_ WRITED	DIV_IN_ PROCEEED	DIV_ RST
							rw				rw	rw	rw	rw	rw



Bits	31:26	保留，必须保持复位值
Bit	25	DIV0_INTREN : 除 0 中断使能位
		0: 除数为 0 时允许进除 0 中断 (默认)
		1: 除数为 0 时不允许进除 0 中断
Bit	24	DIV0 : 除 0 标志位
		0: 除数不为 0 (默认)
		1: 除数为 0
Bits	23:9	保留，必须保持复位值
Bit	8	DIV32 : 被除数长度配置
		0: 被除数为 64 位 (默认)
		1: 被除数为 32 位
Bits	7:5	保留，必须保持复位值
Bit	4	DIV_B_WRITED : 除数写入标志位，写 1 清除
		0: 除数寄存器未写入 (默认)
		1: 除数寄存器已写入
Bit	3	DIV_AHI_WRITED : 被除数写入标志位，写 1 清除
		0: 被除数高位寄存器 (DIV_AHI) 未写入 (默认)
		1: 被除数高位寄存器 (DIV_AHI) 已写入
Bit	2	DIV_ALO_WRITED : 被除数写入标志位，写 1 清除
		0: 被除数低位寄存器 (DIV_ALO) 未写入 (默认)
		1: 被除数低位寄存器 (DIV_ALO) 已写入
Bit	1	DIV_IN_PROCCEED : 除法器运行标志位，写 1 清除
		0: 除法器空闲 (默认)
		1: 除法器正在使用 (B/AHI/ALO 任一寄存器写入)
Bit	0	DIV_RST : 复位除法逻辑
		0: 无效 (默认)
		1: 复位触发逻辑，一个时钟后清 0，并清除 Bit[4:1]

27.3.2 DIV 被除数低位寄存器 (DIV_ALO)

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
DIV_ALO[31:16]															
rw															

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

Bits	31:0	DIV_ALO[31:0] : 被除数寄存器低 32 位
------	------	-------------------------------------

27.3.3 DIV 被除数高位寄存器 (DIV_AHI)

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
DIV_AHI[31:16]															
rw															

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

Bits 31:0	DIV_AHI[31:0]: 被除数寄存器高 32 位
-----------	-----------------------------

27.3.4 DIV 除数寄存器 (DIV_B)

地址偏移: 0x0C

复位值: 0x0000_0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIV_B[31:16]															
rw															

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

Bits 31:0	DIV_B[31:0]: 除数寄存器
-----------	--------------------

27.3.5 DIV 商低位寄存器 (DIV_QUOTLO)

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
DIV_QUOTLO[31:16]															
r															

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

Bits 31:0	DIV_QUOTLO [31:0]: 商寄存器低 32 位
-----------	-------------------------------

27.3.6 DIV 商高位寄存器 (DIV_QUOTHI)

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
DIV_QUOTHI[31:16]															
r															

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

Bits 31:0	DIV_QUOTHI[31:0]: 商寄存器高 32 位
-----------	-------------------------------------

27.3.7 DIV 余数寄存器 (DIV_REM)

地址偏移: 0x18

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIV_REM[31:16]															
r															

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

Bits 31:0	DIV_REM[31:0]: 余数寄存器
-----------	-----------------------------

28. 调试模块

28.1 主要特性

LCM32F037 集成了一个通用的 32 位调试器模块，此调试模块可以让处理器停止于某条指令执行处（断点），或者数据寻址处（观测点）。处理器停止时，内核及系统状态处于可观测状态，当观测完成，内核和系统从停止处可继续往下执行。

配合外部通用仿真器可以实现调试的相关功能。LCM32F037 使用 SW-DP 调试口，仅使用 2 个 I/O (SWDIO 和 SWCLK) 执行调试，共一组 GPIO 功能复用。

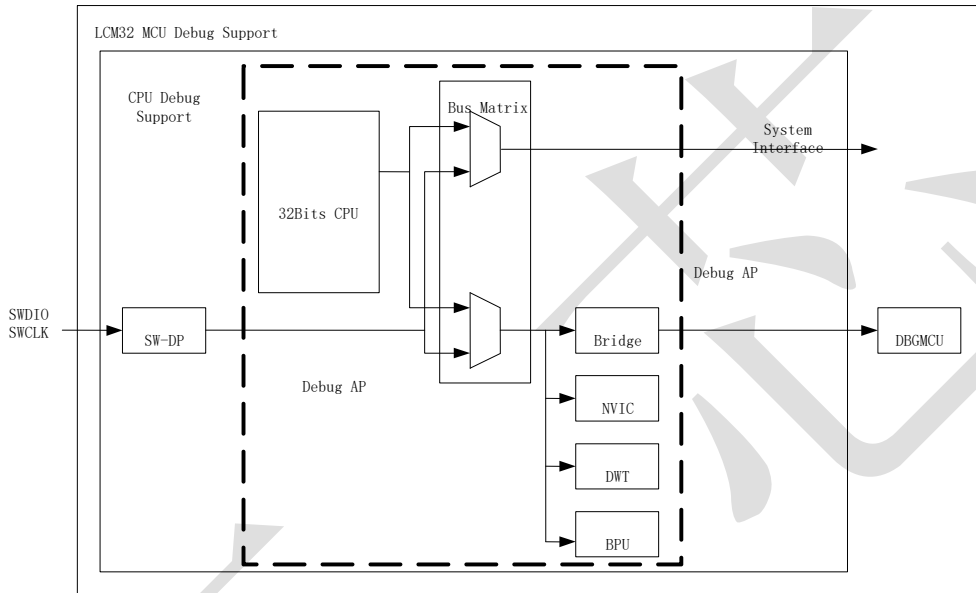


图 28-1 调试模块结构图

CPU 支持的在线调试模式包括以下部分：

- SW-DP：串行调试接口
- BPU：断点模块
- DWT：数据观测点模块

另外，LCM32F037 还包括了其他的一些特性：

- 可以调整的调试复用接口
- MCU 调试模块（支持低功耗模式，外设时钟等）

28.2 调试复用接口

用户可以根据需求，调整 SWD 复用的 I/O，或者调试接口可以配置为普通 I/O 使用。

28.2.1 SWD 复用接口

串行接口 SW-DP 需要两个通用 I/O 口作为其输入、输出通讯，LCM32F037 有一组可复用的 GPIO 供客户配置选择，在所有封装产品中，必须要有相关的调试复用接口。

表 28-1 SW 调试接口的管脚

SW-DP信号	SW调试口说明		外部管脚
	类型	功能	
SWDIO	I/O	串行数据输入输出接口	PA13
SWCLK	I	串行时钟输入	PA14

28.2.2 SW-DP 功能复用说明

上电复位或者系统复位后，调试器可以使用对应的接口进入相关调试模式。

但 MCU 可以通过配置相关寄存器禁止 SWD 管脚的调试功能，并将其作为普通 GPIO 口使用，详情参考 GPIO 相关寄存器的说明。

28.2.3 SWD 管脚的内部上拉或下拉

软件配置 SW 相关管脚为普通 GPIO 后，相关功能由 GPIO 寄存器控制。GPIO 控制寄存器的调试接口复位状态如下：

- SWDIO：输入上拉
- SWCLK：输入下拉

内置的上下拉电阻可以省略外部电阻的使用。

28.3 SWD 通讯

28.3.1 SWD 协议介绍

同步串行通讯协议使用以下两个管脚：

- SWCLK：从主机到目标设备的时钟
- SWDIO：双向数据通道

串行协议允许写入到两组寄存器（DPACC 以及 APACC 寄存器）。传输顺序为 LSB（低位在前）。SWDIO 双向 IO 口外部必须上拉（推荐 100KΩ，LCM32F037 实现了内部上拉）。

协议规定每次 SWDIO 传输方向的改变，都需要插入一个等待周期，此等待周期内，无论主机还是设备都不能驱动数据线。默认情况下，等待周期维持一个 bit 的时钟周期，但可以通过调整 SWCLK 的频率来决定等待的具体时间。

28.3.2 SWD 协议时序

每次传输包含以下三个阶段：

1. 主机发起包传输请求（8 bits）
2. 目标设备反馈应答响应（3 bits）
3. 主机或设备发起数据传输（33 bits）

表 28-2 包传输请求（8 bits）

Bit	名称	描述
0	Start	必须为 1
1	APnDP	0: DP 寻址； 1: AP 寻址
2	RnW	0: 写请求； 1: 读请求
4:3	A[3:2]	DP 或者 AP 寄存器地址（参考后续说明）

5	Parity	校验位
6	Stop	0
7	Park	主机停止驱动，从机必须读到“1”（端口上拉）

包请求传输结束必须有一个等待时间（默认为一个时钟），此时主机和从机都不驱动通讯线。

表 28-3 应答响应（3 bits）

Bit	名称	描述
2:0	ACK	001: 错误 010: 等待 100: OK

当一个读传输，或者等待以及错误响应时，必须在此响应传输后插入一个等待时间。

表 28-4 数据传输（33 bits）

Bit	名称	描述
31:0	WDATA 或 RDATA	写或者读的数据
32	Parity	32 位数据的校验位

读传输结束后必须插入一个等待时间。

28.3.3 SW-DP 状态机（复位、空闲、ID 码）

SW-DP 的控制状态机有一个内部标识 ID 来确定当前传输为 SW-DP，遵循 JEP-106 标准，默认设置为 0x0BB11477。

注：设备只有读到这个ID码才能激活SW-DP状态机。

- SW-DP 在上电复位，或者通讯线维持 50 个时钟的高电平后，处于初始复位状态；复位后，如果通讯线维持两个时钟以上的低电平，则 SW-DP 状态机处于空闲状态；
- 复位状态后，强制要求通讯必须先进入空闲状态，然后发起一个 DP-SW 的 ID 码读请求，否则设备在下次传输时会反馈一个错误响应（FAULT）。

28.3.4 DP 及 AP 读写操作

- 对 DP 的读操作不会被推迟：设备可以马上反馈数据（反馈 ACK=OK），或延迟（如果反馈 ACK=WAIT）；
- 对 AP 的读操作会被推迟，即读操作结果只有下次传输时才会返回。如果紧接着的下次传输不是 AP 访问，则寄存器 DP-RDBUFF 必须被读走以得到正确结果。寄存器 DP-CTRL/STAT 的 READOK 标志在每次 AP 读操作或者 RDBUFF 读操作后更新，以确定 AP 读访问是否成功；
- SW-DP 内置一个写缓存（同时用于 DP 或者 AP 的写入），这样即使在其他操作还没有完成的情况下，系统能够接受对其的写入操作。如果写缓存满，则设备会反馈一个“WAIT”响应。但 IDCODE 读操作、CTRL/STAT 读以及 ABORT 写（写退出）还是可以会被接收，即使是在写缓存满的状态下；
- 由于 SWCLK 和 HCLK 是异步时钟，因此一个写传输（校验位传送之后）之后需要有额外的两个 SWCLK 的时钟周期来保证内部写操作完成，在此额外的周期内，传输线须保持为低电平（空闲 IDLE 状态）。特别重要的是当写入 CTRL/STAT 寄存器一个启动请求，如果接下来的传输马上发起，则会失败。

28.3.5 SW-DP 寄存器

当 APnDP = 0 时，访问如下的寄存器。

表 28-5 SW-DP 寄存器

A[3:2]	R/W	SELECT 寄存器的 CTRLSEL 位	寄存器	备注
00	Read	-	IDCODE	CPU 固定识别码, 标识 SW-DP 传输: 0x0BB11477
00	Write	-	ABORT	
01	R/W	0	DP-CTRL/STAT	有以下功能: <ul style="list-style-type: none"> ● 发起系统或调试装置的上电请求; ● 配置AP访问的传输操作方式; ● 控制堆栈序列比较和校验操作; ● 读取状态标志 (过载, 上电完成等)
01	R/W	1	WIRE CONTROL	配置物理串行通讯参数 (例如等待时间周期数)
10	Read	-	READ RESEND	恢复一个之前被调试器占用的读数据传输, 从而避免重复发送之前的 AP 传输请求
10	Write	-	SELECT	选择当前访问端口及对应 4 个字的寄存器组窗口
11	R/W	-	READ BUFFER	此读缓存作用于 AP 访问推迟 (一个读 AP 的操作在下一个 AP 传输方才有效), 读缓存从 AP 中取得数据, 并作为之前读操作的结果, 不需要重新发起一个新的传输

28.3.6 SW-AP 寄存器

当 APnDP = 1 时, 访问 AP 相关的寄存器。通过以下地址组合来寻址: 1. 传输输入地址 A[3:2]; 2. DP SELECT 寄存器的值。

表 28-6 通过输入地址 A[3:2]选择的调试端口寄存器

地址	A[3:2]	描述
0x0	00	保留, 必须保持复位值
0x4	01	DP CTRL/STAT 寄存器, 用于: <ul style="list-style-type: none"> ● 发起系统或调试装置的上电请求; ● 配置AP访问的传输操作方式; ● 控制堆栈序列比较和校验操作; ● 读取状态标志 (过载, 上电完成等)
0x8	10	DP SELECT 寄存器: 用于选择当前访问端口以及选定的 4 字寄存器窗 <ul style="list-style-type: none"> ● Bits 31:24: APSEL, 选择当前的AP ● Bits 32:8: 保留 ● Bits 7:4: APBANKSEL, 选择当前AP对应的4字的寄存器窗 ● Bits 3:0: 保留
0xC	11	DP RDBUFF 寄存器: 用于调试器经过一系列命令操作后得到的结果 (而无需发起新的 SW-DP 操作)

28.4 内核的调试

内核调试通过读写内核调试寄存器实现, 调试器通过调试访问端口 (AP) 实现对这些寄存器的访问, 包

含以下 4 个寄存器：

表 28-7 内核调试寄存器

寄存器	描述
DHCSR	32 位调试状态下的控制和状态寄存器：提供调试状态下，可停止或者单步的处理器内核的状态信息
DCRSR	17 位调试 CPU 寄存器的选择：选择需要进行读写访问的 CPU 寄存器
DCRDR	32 位调试内核寄存器的数据：保持被寄存器 SCRSR 选中的寄存器需要写入或者读出的数据
DEMCR	32 位调试中断和监控控制寄存器：用于向量抓取以及调试监控和控制

这些寄存器只能通过上电复位回到默认值，系统复位不起作用。若需要复位后停止处理器，须通过以下方式：

- 使能调试和事件监控控制寄存器的 Bit0 (VC_CORRESET)
- 使能调试状态控制和状态寄存器的 Bit0 (C_DEBUGEN)

28.5 断点模块 BPU

处理器集成了 4 个断点及相关寄存器。处理器断点的实现基于内部程序指针 PC，具体参考相关文档。

28.6 数据观测点 DWT

处理器 DWT 单元集成了两个数据监控及寄存器。

28.6.1 数据监控功能说明

处理器的数据监控实现了基于数据地址和程序指针 PC 的观测寄存器，和一个 PC 采样寄存器，支持地址比较，具体参考相关文档。

28.6.2 DWT 的程序指针采样寄存器

实现数据观测功能的处理器包括了可选的 DWT 程序指针采样寄存器 (DWT_PCSR)，此寄存器可以让调试器周期性采样 PC 而不需要将处理器先停止下来，从而可以提供一个较粗粒度的观测。DWT_PCSR 寄存器记录的 PC 不管条件跳转指令条件是否成立，都会被记录下来。

28.7 MCU 调试单元

MCU 调试单元支持调试器的以下状态：

- 低功耗模式
- 复位状态下进入调试模式
- 断点时候的定时器、看门狗等模块的时钟控制

28.7.1 低功耗模式下的调试

必须通过执行 WFI 或 WFE 才能进入低功耗模式。调试模式下，系统时钟 FCLK 和 HCLK 必须保持开启的状态，不同低功耗模式下：

- 睡眠模式下，FCLK 和 HCLK 没有被关闭，不影响调试功能
- 停机模式或超低功耗停机模式下，时钟被关闭，复位状态不能维持，调试器会失去连接



注：客户低功耗停止模式下，会失去调试器和上位机连接。

28.7.2 定时器、看门狗以等模块的调试支持

当 MCU 处于断点停止模式下，需要确定定时器、看门狗等计数器的行为：

- 可以在停止模式下继续保持计数，通常应用于一些 PWM 控制的场合，例如电机控制等
- 也可以在调试状态下停止计数，通常是停止看门狗的计数

28.7.3 调试模式控制寄存器

此寄存器只能被外部复位、IWDG 复位和电源复位来复位，而不能被调试复位，用于控制 MCU 外围相关模块在调试状态下的行为，相关外设设备有：

- TIMER 相关计数器的停止
- 系统窗看门狗以及独立看门狗的计数停止
- 实时时钟模块的计数

寄存器地址：0x4800_7014（由系统密钥寄存器 SysCtrl_KEY 控制）

复位值：0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						DBG_IWDG_STOP	DBG_WWDG_STOP	Res.	DBG_TIM17_STOP	DBG_TIM16_STOP	DBG_TIM15_STOP	DBG_TIM14_STOP	DBG_TIM6_STOP	DBG_TIM3_STOP	DBG_TIM1_STOP
						rw	rw		rw	rw	rw	rw	rw	rw	rw

Bits	31:10	保留，必须保持复位值
Bit	9	DBG_IWDG_STOP ：调试停止状态下 IWDG 的控制
		0：调试状态下 IWDG 继续保持计数（默认）
		1：调试状态下 IWDG 停止计数
Bit	8	DBG_WWDG_STOP ：调试停止状态下 WWDG 的控制
		0：调试状态下 WWDG 继续保持计数（默认）
		1：调试状态下 WWDG 停止计数
Bit	7	保留，必须保持复位值
Bit	6	DBG_TIM17_STOP ：调试停止状态下 TIM17 的控制
		0：调试状态下 TIM17 继续保持计数（默认）
		1：调试状态下 TIM17 停止计数
Bit	5	DBG_TIM16_STOP ：调试停止状态下 TIM16 的控制
		0：调试状态下 TIM16 继续保持计数（默认）
		1：调试状态下 TIM16 停止计数
Bit	4	DBG_TIM15_STOP ：调试停止状态下 TIM15 的控制
		0：调试状态下 TIM15 继续保持计数（默认）
		1：调试状态下 TIM15 停止计数

Bit 3	DBG_TIM14_STOP: 调试停止状态下 TIM14 的控制
	0: 调试状态下 TIM14 继续保持计数 (默认)
	1: 调试状态下 TIM14 停止计数
Bit 2	DBG_TIM6_STOP: 调试停止状态下 TIM6 的控制
	0: 调试状态下 TIM6 继续保持计数 (默认)
	1: 调试状态下 TIM6 停止计数
Bit 1	DBG_TIM3_STOP: 调试停止状态下 TIM3 的控制
	0: 调试状态下 TIM3 继续保持计数 (默认)
	1: 调试状态下 TIM3 停止计数
Bit 0	DBG_TIM1_STOP: 调试停止状态下 TIM1 的控制
	0: 调试状态下 TIM1 继续保持计数 (默认)
	1: 调试状态下 TIM1 停止计数

29. 修订历史

版本	修改日期	修改历史	修改人
0.1	2017.10.13	初始版本	
0.2	2019.05.06	内部版本，内容校对专用	
0.3	2019.05.09	修改 DAC5 寄存器说明，去除 DMA 分散和归集寄存器,版本号变更为 V0.5	
0.4	2019.05.09	删除封装，封装形式请参见对应产品说明书, V0.6	
0.5	2019.05.20	调准目录结构和图片序号	
0.6	2019.06.05	修正比较器正端选择的 0 和 1 互换的错误	
1.0	2019.07.26	增加模块时钟和复位寄存器描述，增加互联矩阵寄存器说明	
1.2	2019.08.13	TIMER 图形错误修正，更正 PLL 设定公式错误	
1.4	2020.06.17	1.增加 RTC 模块寄存器描述 2.增加触发器描述 3. LCM32F037 独立版本文档	
1.8	2020.07.07	更正版本号；	
1.9	2020.07.16	更正运放寄存器描述	
2.0	2020.08.14	ECO 改版产出芯片有效 1.运放 OPA，模拟比较器引脚复用调准 2.ADC 采用通道标准 3.模拟寄存器去除密钥寄存器	
2.1	2020.08.24	1.更新运放寄存器关于引脚选择的描述	
3.0	2020.10.14	1.更新了 TIM15、16、17 向下计数模式和中央计数模式的说明，以及对应的文档目录与图表目录的更新 2.添加 ADC 模块 FIFO 的相关寄存器说明，DMACON 位的寄存器与配置说明 3.添加 ADC 模块 DMA 内 FIFO 功能的说明，针对 DMA 请求和 FIFO 与 ADC->DR 寄存器，重新进行了说明 4.对 OVR、OVRIE 和 OVRMODE 相关部分做了删减或者进行了重新说明 5. uart: 进行了修改，包括对模糊描述重新说明、删减了关于 uart3 模块相关的说明（红外模式以及相关寄存器说明） 6. I2C 与 DMA 相关部分重新说明 7. SPI 添加与 DMA 相关的说明 8. 添加 I2C 的 SPEED 和 UART 奇偶校验的说明 9.修正了 eflash 的地址分配 10.添加了 TIMx 的 CR2[16]位对应 OCTRIG 位，和相关说明 11.添加了 tim15 ch2n，通道 2 的输出互补通道寄存器配置说明 12.对 ADC 的通道进行了补充说明	Hu
3.1	2020.10.19	1.对 ADC 双通道先后顺序做了明确性说明 2.对 ADC 的 ISR 寄存器的修改进行了补充说明 3.修正完善了 ADC 的 wait 模式的说明 4.补充说明了 ADC 的 FIFO 溢出时的说明 5.修正了 ADC 的 FIFO 状态说明	Hu
3.2	2020.10.27	1.修正完善了 ADC 的所有说明和寄存器表格（改版后 ADC 模块修正完毕）	Hu
3.3	2020.10.29	1.修改了若干笔误和描述不清 2.将 RTC 模块代替为 WT 模块	Hu Lee

		3.将寄存器表格格式统一	
3.4	2020.10.30	1.将寄存器表格格式统一 (I2C、UART)	Hu
3.5	2020.11.04	1.统一寄存器说明	Hu/Lee
4.0	2020.12.01	1.寄存器格式与说明重新统一	Hu/Wu
4.1	2020.12.10	1.各部分修订完成	Hu/Wu
4.2	2020.12.10	1.新版修订完成	Hu/Wu
4.3	2021.01.07	1.F03X 正式更名为 F037，并校验完毕	Hu/Wu
4.4	2021.01.11	1.与说明书 V2.5 版同步，表 7-3 增加《可选复用功能》合并《注释》和《外部功能》	Hu Wu
4.5	2021.01.19	1.调整了 TIM 的寄存器说明 2.添加了外设存储空间分配的 DIV	Hu Wu
4.6	2021.05.08	1.修改寄存器描述错误	Wu
4.7	2021.05.11	1.修改 TIM15 描述及框图错误	Fu
4.8	2021.05.12	1.在 WT 功能简介的时钟源处做注释；修改 WT 模块寄存器偏移地址	Fu
4.9	2021.05.14	1.OPTRIMO 从 0 变成 1 的过程改为从 1 变成 0；OPxTRIMO 由 1 到 0 跳变改为由 0 到 1； 2.比较器输入由 IO 口改成对应 OPA0/1/2 的输出	Fu
5.0	2021.05.26	1.修改除法器控制寄存器(DIV_CTRL)Bit24 描述错误 2.除法器除数寄存器(DIV_B)复位值改为 0x00000001	Fu
5.1	2021.05.28	1. 修改 UART 标志寄存器 FR 中 Bit7-4 描述错误	Fu
5.2	2021.06.02	1.ACMP 章节增加中断配置描述，具体见 11.3.2.3 2.ACMP_CSR Bit10~9 增加延迟配置表格说明 3.添加 ACMP 迟滞电压值	Fu
5.3	2021.06.08	1.添加 ADC 采样周期图 2.ADC_SMPR 修改 Bit6~4 描述	Fu
5.4	2021.06.17	1.错别字修改	Fu
5.5	2021.06.21	1. ANACTRL_CSR 寄存器 Bit9~8 描述及对应框图修改	Fu
5.6	2021.06.23	1. ACMP0_EXTCFG 寄存器 Bit28~27 描述修改 2. ADC_EXTCFG 寄存器 Bit31~27 描述修改	Fu
5.7	2021.06.24	1. DAC、ACMP、HALL_MID 内部框图 PB6、PB7 模拟复用修改	Fu
5.8	2021.06.28	1.修正图片索引	Fu
5.9	2021.06.30	1.中断和异常向量表 RCC 添加 () 描述	Fu
6.0	2021.07.07	1.ACMP0 多路选择控制器(ACMP0_MUXCR)Bit1 添加备注 2.芯片低压复位及电源配置寄存器 (ChipCtrl_LVR) 位描述调整 3.TIM16 Bit6~4 描述修改 4.触发源选择 TIM1_CCx 改为 TIM1_OCx	Fu
6.1	2021.07.08	1.修改 TIM15_SMCR 寄存器	Fu
6.2	2021.07.15	1.ACMP 添加连续采样模式介绍 2.ACMP_MUXCR 修改 Bit29 位描述	Fu
6.3	2021.07.29	1.ADC 模拟通道数改为 24 2.工作频率改为 72MHz	Fu
6.4	2021.08.27	1.修改 IWDG 功能描述部分内容 2.修改 ADC 部分关于通道的描述内容	Fu
6.5	2021.08.31	1.统一 OPA、ACMP 通道名称	Fu

6.6	2021.09.01	1. ACMPX_MUXCR Bit0 添加描述	Fu
6.7	2021.09.29	1.更换 OPA 框图 2.修改 ANACTRL_CSR Bit9~8 描述 3.增加 2.6UId 地址, 修正其他地方笔误 4.ETF 描述修改	Fu
6.8	2021.11.09	1.修改启动模式内容描述 2.修改 I2C_STATUS 文字错误 3.工作频率更新为 96MHz	Fu
6.9	2021.12.09	1.修改 ACMPx_CSR Bit28 描述 2.修改 A/D 转换速率	Fu
7.0	2021.12.20	1.修改 ADC_EXTCFG Bit[11:4]描述 2.修改 IC_RX_TL、IC_TX_TL 描述	Fu
7.1	2022.02.16	1.修改 8.2.1DMA 处理文字描述 2.修改 CHIPCTRL_OSCHCFG 寄存器 OSCH_TS[2:0]描述 3.修改 5.2 时钟系统第一段文字描述 4.修改 CHIPCTRL_STS 寄存器 Bit 7 到 Bit 0 的位描述 5.修改低功耗模式里描述不准确的地方 6.10.7 添加图片及文字 7.修改出场校准字节 6/7, 添加出场校准字节 16/17/18/19/20 8.修改 TIM14 SR 寄存器 Bit1 描述 9.修正图 11-10 HALL_MID 内部框图 10.修改 4.4.1 电源配置寄存器(PWR_CFG) Bit3: 1	Fu
7.2	2022.09.08	1.修改 2.6 UID 地址 2.添加 DAC_CR Bit5-3 缺失描述 3.修改 10.7 部分描述 4.检查全文错误并修改	Fu
7.2.1	2022.09.15	1.修改 OPA 内部结构图中 OPA_MIN 的来源	Li
7.2.2	2022.10.09	1.修改 Tn _{ADC} 和 Tn 描述 2.检查全文笔误并修改	Li
7.3	2022.11.07	1.修改定时器中编码器模式的描述 2.检查全文笔误并修改 3.添加文档说明	Li
7.4	2023.04.17	1.添加 LCP037A 系列, LCP037B 系列产品型号 2.检查全文笔误并修改	Li/Yang
7.5	2023.05.31	1.修改定时器空闲电平的条件描述 2.检查全文笔误并修改	Li
7.6	2023.08.04	1.检查全文笔误并修改	Yang
7.7	2023.08.18	1.补充 8.3.2.3 章节 TT_FC 描述 2.增加所有 timer 的 CEN 的注释 3.修正部分笔误	Yang