



杭州领芯微电子有限公司

LCM32F067/LCM32F063 /LCP067 用户手册

文档版本：2.3

发布日期：2025.02.14

适用产品：LCM32F067 系列

LCM32F063 系列

LCP067 系列

内部版本



32 位 ARM Cortex-M0 MCU, 64KB Flash/10KB RAM, 高达 25 个快速 I/O, 7 个定时器, 4 个通信接口, 支持 LIN, 1 个 ADC, 1 个 DAC, 2 个比较器, 3 个运放, 2.0~5.5V

主要特性

- 内核: 32位ARM Cortex-M0 MCU
 - LCM32F067最高工作频率108MHz, LCM32F063最高工作频率96MHz
 - CORDIC协处理器
 - 硬件除法器
 - 内置电机加速算法
- 存储器
 - 64KBytes 嵌入式 Flash (位宽 32bit), 支持预取功能和读/写保护, 支持 Parity 校验
 - Flash 擦写次数不低于 10 万次
 - 10KBytes SRAM (位宽 32bit), 分为两个独立分区, 分别为 4KBytes 和 6KBytes, 支持 Parity 校验
- 复位和电源管理
 - 2.0V到5.5V供电和I/O
 - 内置两个LDO: 正常LDO和低功耗LDO
 - 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
 - 内置出厂校准过的16MHz RC振荡器 (RCH, 1%精度)
 - 32KHz 的低速晶振 (OSCL)
 - 内置出厂校准过的32KHz RC振荡器 (RCL, 10%精度)
 - 内置PLL, 最高输出256MHz, 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或SRAM启动
- 编程模式
 - 支持串行在系统编程 (ISP)
- 多达25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
- 7个定时器
 - 1个16位高级控制定时器TIM1, 5路通道 (带4路互补通道), 支持输出比较/PWM输出/单脉冲输出, 支持死区控制和紧急刹车
 - 1个16位通用定时器TIM2, 4路通道, 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、霍尔检测和紧急刹车
 - 1个16位通用定时器TIM15, 2路通道 (带2路互补通道), 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, 支持中央调制模式
 - 1个16位通用定时器TIM16, 2路通道 (带1路互补通道), 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, 支持中央调制模式
 - 1个16位通用定时器TIM17, 1路通道 (带1路互补通道), 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, 支持中央调制模式
 - 1个独立看门狗定时器
 - 1个24位自减型系统时基定时器
 - TIM1、TIM15和TIM16支持延时触发和防误触发机制



- **WT 钟表定时器**
 - 支持闹钟、周期性唤醒
 - 可配置频率的蜂鸣信号输出
- **通用 DMA**
 - 2路独立通道，8个握手信号
 - 支持的外设包括SPI、I2C、USART、ADC、DAC和Timer
- **CRC 计算单元**
 - 8位、16位、32位可配置生成多项式
- **DIV 计算单元**
 - 1个32/32除法器，8运算周期，支持有符号运算，向下兼容16/16
- **CORDIC 计算单元**
 - 支持三角函数、反三角函数、双曲正切函数和开方等计算
- **多达 4 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位双地址模式，支持SMBus，带FIFO和支持DMA
 - 2 个 USART 接口，支持 CTS/RTS 硬流控，最高波特率为 4Mbps，带 FIFO 和支持DMA；2 个 USART 均支持低功耗模式；均硬件支持 LIN 协议规范 1.3、2.0、2.1、2.2 和 J2602
 - 1个SPI接口，支持主机/从机模式，4到16位的帧大小，主机最高速率达32Mbps，从机最高速率达24Mbps，带FIFO和支持DMA
- **1 个 12 位 A/D 转换器**
 - 最高转换速率为2MSPS
 - 18个通道（16个外部通道，2个内部通道）
 - 内置温度传感器VTS
 - 支持内部和外部参考电压：3.2V（VDDA>2.2V）、4.9V（VDDA>4V）或VDDA
 - 双采样保持电路，灵活可配的序列模式，支持ADC通道挂起和注入
- **1 个 10 位 D/A 转换器**
 - 1个10位D/A转换器，参考电压可配置
 - 支持硬件触发和DMA传输，支持噪声波形和三角波形生成
- **2 个模拟比较器（ACMP）**
 - 2个模拟比较器，比较器的输入可选择内部或者外部输入
 - 内置HALL中心点还原模块，输出可观测
- **3 个运算放大器（OPA）**
 - OPA0/1/2放大倍数：1/2/4/6/8/10/16/32
 - 运放的所有输入均支持外部2路输入，或内部接地
- **工作温度**
 - 环境温度：-40°C ~ +125°C
- **128 位芯片唯一 ID**



目录

主要特性	2
目录	4
表格目录	19
图片目录	22
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64KB Flash及选项字节说明	34
3.1 Flash的主要特性	34
3.2 Flash模块的功能性说明	34
3.2.1 存储模块的组织形式	34
3.2.2 Flash的读操作	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）	42
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）	43
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 读保护寄存器（FLASH_RDPR）	47
3.5.9 写保护寄存器（FLASH_WRPR）	48
3.6 选项字节	48
3.6.1 用户及数据选项配置字节	49
3.6.2 用户配置选项字节	50
3.6.3 写保护选项字节	50
3.6.4 读保护选项字节	51
3.7 校准字节	54
4. 电源管理（PWR）	57



4.1	供电方案	57
4.1.1	模拟电源VDDA供电的相关模块	57
4.1.2	电压调压器（LDO）	58
4.2	电源检测	58
4.3	低功耗模式	58
4.3.1	降低系统时钟频率	59
4.3.2	外设时钟的关闭	59
4.3.3	睡眠模式（SLEEP）	59
4.3.4	停机模式（STOP）	60
4.3.5	超低功耗停机模式（ULP STOP）	61
4.3.6	低功耗模式的WT唤醒	62
4.4	PWR寄存器描述	62
4.4.1	电源配置寄存器（PWR_CFG）	63
4.4.2	电源密钥寄存器（PWR_KEY）	64
4.4.3	电源模式控制寄存器（PWR_CR）	65
4.4.4	电源LDO配置寄存器（PWR_LDOCR）	65
5.	时钟和复位控制（RCC）	67
5.1	复位	67
5.1.1	电源复位	67
5.1.2	系统复位	67
5.2	时钟系统	67
5.3	寄存器描述	69
5.3.1	芯片配置寄存器组（ChipCtrl）	69
5.3.2	系统配置寄存器组（SysCtrl）	82
6.	系统控制单元	90
6.1	Multi-AHB总线矩阵	90
6.2	外设互联矩阵	90
6.3	互联配置寄存器描述	90
6.3.1	互联配置寄存器0（SysCtrl_EDU_CFG0）	91
6.3.2	互联配置寄存器1（SysCtrl_EDU_CFG1）	92
6.3.3	互联配置寄存器3（SysCtrl_EDU_CFG3）	93
6.3.4	互联配置寄存器4（SysCtrl_EDU_CFG4）	95
6.3.5	互联配置寄存器5（SysCtrl_EDU_CFG5）	96
6.3.6	互联配置寄存器6（SysCtrl_EDU_CFG6）	97
6.3.7	互联配置寄存器7（SysCtrl_EDU_CFG7）	98
6.3.8	互联配置寄存器8（SysCtrl_EDU_CFG8）	98
6.3.9	互联配置寄存器9（SysCtrl_EDU_CFG9）	100
6.3.10	互联配置寄存器11（SysCtrl_EDU_CFG11）	101
6.3.11	互联配置寄存器12（SysCtrl_EDU_CFG12）	102
6.3.12	互联配置寄存器15（SysCtrl_EDU_CFG15）	103
6.3.13	互联配置寄存器16（SysCtrl_EDU_CFG16）	104
6.3.14	互联配置寄存器17（SysCtrl_EDU_CFG17）	105
6.3.15	系统密钥寄存器（SysCtrl_KEY）	106
7.	通用I/O端口（GPIO）	107

7.1 主要特征	107
7.2 功能描述	107
7.3 引脚说明	109
7.4 GPIO寄存器描述	115
7.4.1 GPIO端口模式寄存器 (GPIOx_MODER) (x=A,B)	115
7.4.2 GPIO端口开漏设置/清除寄存器 (GPIOx_OD_BSRR) (x=A,B)	116
7.4.3 GPIO端口上拉设置/清除寄存器 (GPIOx_PU_BSRR) (x=A,B)	117
7.4.4 GPIO端口下拉设置/清除寄存器 (GPIOx_PD_BSRR) (x=A,B)	117
7.4.5 GPIO端口输入数据寄存器 (GPIOx_IDR) (x=A,B)	118
7.4.6 GPIO端口输出数据寄存器 (GPIOx_ODR) (x=A,B)	118
7.4.7 GPIO端口数据设置/清除寄存器 (GPIOx_BSRR) (x=A,B)	119
7.4.8 GPIO端口配置锁定寄存器 (GPIOx_LCKR) (x=A,B)	119
7.4.9 GPIO端口复用功能配置寄存器低位 (GPIOx_AFR1) (x=A,B)	120
7.4.10 GPIO端口复用功能配置寄存器高位 (GPIOx_AFR2) (x=A,B)	121
7.4.11 GPIO端口输出数据翻转寄存器 (GPIOx_BTLR) (x=A,B)	122
7.4.12 GPIO端口驱动强度设置/清除寄存器 (GPIOx_DR_BSRR) (x=A,B)	122
7.4.13 GPIO端口斜率和施密特触发器输入设置/清除寄存器 (GPIOx_CS_SR_BSRR) (x=A,B) ..	123
7.4.14 GPIO端口开源模式设置/清除寄存器 (GPIOx_OS_BSRR) (x=A,B)	123
8. DMA控制器	125
8.1 主要特性	125
8.2 DMA功能描述	126
8.2.1 DMA处理	126
8.2.2 DMA仲裁	126
8.2.3 DMA中断	126
8.2.4 DMA请求与数据传输	127
8.2.5 DMA请求映射	129
8.3 DMA寄存器描述	131
8.3.1 全局配置和使能寄存器	132
8.3.2 通道寄存器	133
8.3.3 中断寄存器	136
8.3.4 DMA互联配置寄存器 (SysCtrl_EDU_CFG0)	147
8.4 DMA参考编程流程	148
9. 中断和事件	150
9.1 嵌套向量中断控制器 (NVIC)	150
9.1.1 NVIC主要特性	150
9.1.2 中断和异常向量	150
9.2 外部中断/事件控制器 (EXTI)	151
9.2.1 EXTI主要特性	151
9.2.2 EXTI事件管理	152
9.2.3 EXTI功能说明	152
9.2.4 EXTI外部和内部中断/事件线的映射	153
9.3 EXTI寄存器描述	153
9.3.1 EXTI中断屏蔽寄存器 (EXTI_IMR)	154
9.3.2 EXTI事件屏蔽寄存器 (EXTI_EMR)	154

9.3.3	EXTI上升沿触发选择寄存器 (EXTI_RTISR)	154
9.3.4	EXTI下降沿触发选择寄存器 (EXTI_FTSR)	155
9.3.5	EXTI软件中断事件寄存器 (EXTI_SWIER)	155
9.3.6	EXTI挂起寄存器 (EXTI_PR)	156
9.3.7	EXTI滤波寄存器 (EXTI_FR)	156
9.3.8	EXTI控制寄存器 (EXTI_CR)	157
9.3.9	系统EXTI配置寄存器0 (SysCtrl_EXTI_CFG0)	157
9.3.10	系统EXTI配置寄存器1 (SysCtrl_EXTI_CFG1)	158
10.	模数转换器 (ADC)	159
10.1	ADC主要特性	159
10.2	ADC功能描述	159
10.2.1	时钟	160
10.2.2	开关控制 (EN、DIS)	160
10.2.3	工作控制 (START、STOP)	161
10.2.4	通道	162
10.2.5	可编程采样转换周期	165
10.2.6	单次采样转换 (SOC)	168
10.2.7	采样模式	170
10.2.8	触发单元	171
10.2.9	模拟窗口看门狗 (AWD)	172
10.3	DMA搬运	173
10.4	ADC中断	174
10.5	ADC寄存器描述	174
10.5.1	ADC状态寄存器 (ADC_SR)	175
10.5.2	ADC中断状态寄存器 (ADC_ISR)	176
10.5.3	ADC中断使能寄存器 (ADC_IER)	176
10.5.4	ADC_SOC中断状态寄存器 (ADC_SOC_ISR)	177
10.5.5	ADC_SOC中断使能寄存器 (ADC_SOC_IER)	177
10.5.6	ADC控制寄存器 (ADC_CR)	177
10.5.7	ADC配置寄存器 (ADC_CFGR)	179
10.5.8	ADC采样周期寄存器 (ADC_SMPR)	180
10.5.9	ADC校准寄存器 (ADC_TRIM)	181
10.5.10	ADC模拟寄存器 (ADC_ANA)	182
10.5.11	ADC DMA配置寄存器 (ADC_DMCFG)	182
10.5.12	ADC DMA传输寄存器 (ADC_DADDR)	183
10.5.13	ADC模拟看门狗[x]配置寄存器 (ADC_AWD[x]CFGR) (x=0~1)	183
10.5.14	ADC模拟看门狗[x]阈值寄存器 (ADC_AWD[x]TR) (x=0~1)	184
10.5.15	ADC外部触发配置寄存器0 (ADC_EXTCFG0)	184
10.5.16	ADC_SOC[x]配置寄存器 (ADC_SOC[x]CFG) (x=0~15)	184
10.5.17	ADC数据[x]寄存器 (ADC_DR[x]) (x=0~15)	185
10.5.18	ADC互联配置寄存器11 (SysCtrl_EDU_CFG11)	185
10.5.19	ADC互联配置寄存器12 (SysCtrl_EDU_CFG12)	187
11.	数模转换器 (DAC) 和模拟比较器 (ACMP) 和反向电动势采样控制器 (HALL_MID)	189
11.1	主要特性	189

11.1.1	DAC特性	189
11.1.2	ACMP特性	189
11.2	功能描述	190
11.2.1	DAC功能描述	190
11.2.2	ACMP功能描述	193
11.2.3	HALL_MID功能描述	205
11.3	DAC/ACMP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多路选择控制寄存器 (ACMP0_MUXCSR)	207
11.3.3	ACMP0硬件触发寄存器 (ACMP0_EXTCFG)	209
11.3.4	ACMP0轮询通道选择寄存器 (ACMP0_CHNLSEL)	209
11.3.5	ACMP1控制及状态寄存器 (ACMP1_CSR)	210
11.3.6	ACMP1多路选择控制寄存器 (ACMP1_MUXCSR)	212
11.3.7	ACMP1硬件触发寄存器 (ACMP1_EXTCFG)	213
11.3.8	ACMP1轮询通道选择寄存器 (ACMP1_CHNLSEL)	214
11.3.9	DAC控制寄存器 (DAC_CR)	215
11.3.10	DAC状态寄存器 (DAC_SR)	216
11.3.11	DAC数据保持寄存器 (DAC_DHR)	217
11.3.12	DAC数据转换输出寄存器 (DAC_DOR)	217
11.3.13	HALL_MID控制寄存器 (HALL_CR)	218
11.3.14	模拟控制及状态寄存器 (ANA_CTRL)	219
11.3.15	ACMP互联配置寄存器15 (SysCtrl_EDU_CFG15)	220
11.3.16	ACMP互联配置寄存器16 (SysCtrl_EDU_CFG16)	220
11.3.17	DAC互联配置寄存器17 (SysCtrl_EDU_CFG17)	222
12.	运算放大器 (OPA)	224
12.1	OPA主要特性	224
12.2	管脚配置	224
12.3	功能描述	227
12.3.1	OPA校准	228
12.3.2	运放OPA与ADC联动	229
12.3.3	OPA运放典型配置	229
12.4	OPA寄存器描述	230
12.4.1	OPA0控制状态寄存器 (OPA0_CSR)	230
12.4.2	OPA1控制状态寄存器 (OPA1_CSR)	232
12.4.3	OPA2控制状态寄存器 (OPA2_CSR)	234
13.	PWM专用定时器TIM1	237
13.1	TIM1主要特性	237
13.2	管脚配置	238
13.3	时基单元	239
13.3.1	自动重装载寄存器	239
13.3.2	预分频器	240
13.4	计数器模式	240
13.4.1	向上计数模式	240
13.4.2	向下计数模式	241



13.4.3 中央对齐模式（向上/向下计数）	242
13.5 重复计数器	243
13.6 时钟选择	244
13.6.1 内部时钟源（TIM1_CLK）	244
13.6.2 外部时钟模式1	245
13.6.3 外部时钟模式2	245
13.7 比较通道	246
13.7.1 输出模块	247
13.7.2 强制输出模式	248
13.7.3 输出比较模式	248
13.7.4 PWM模式	249
13.7.5 单脉冲模式	252
13.7.6 特殊情况：OCi快速使能	253
13.7.7 互补输出和死区插入	253
13.7.8 重定向OCiREF到OCi或OCiN	254
13.7.9 刹车功能	254
13.7.10 通过外部事件清除OCiREF	257
13.7.11 PWM生成	257
13.7.12 与霍尔传感器的接口	258
13.8 TIM1定时器与外部触发的同步	259
13.8.1 标准触发模式	259
13.8.2 复位模式	260
13.8.3 门控模式	260
13.8.4 触发复位模式	261
13.8.5 外部时钟模式2及触发模式	261
13.8.6 触发延迟和防误触发功能	261
13.9 TIM1与其他定时器的联接	262
13.10 TIM1中断	263
13.11 TIM1寄存器描述	263
13.11.1 CR1控制寄存器1（TIM1_CR1）	264
13.11.2 CR2控制寄存器2（TIM1_CR2）	266
13.11.3 SMCR从模式控制寄存器（TIM1_SMCR）	268
13.11.4 DIER DMA和中断控制寄存器（TIM1_DIER）	270
13.11.5 SR状态寄存器（TIM1_SR）	271
13.11.6 EGR事件产生寄存器（TIM1_EGR）	273
13.11.7 CCMR1比较模式寄存器1（TIM1_CCMR1）	274
13.11.8 CCMR2比较模式寄存器2（TIM1_CCMR2）	276
13.11.9 CCMR3比较模式寄存器3（TIM1_CCMR3）	277
13.11.10 CCER比较使能寄存器（TIM1_CCER）	277
13.11.11 CNT计数寄存器（TIM1_CNT）	280
13.11.12 PSC预分频寄存器（TIM1_PSC）	281
13.11.13 ARR自动重装载寄存器（TIM1_ARR）	281
13.11.14 RCR重复计数寄存器（TIM1_RCR）	281
13.11.15 CCR1比较寄存器1（TIM1_CCR1）	282

13.11.16	CCR2比较寄存器2 (TIM1_CCR2)	283
13.11.17	CCR3比较寄存器3 (TIM1_CCR3)	283
13.11.18	CCR4比较寄存器4 (TIM1_CCR4)	284
13.11.19	CCR5比较寄存器5 (TIM1_CCR5)	284
13.11.20	BDTR刹车和死区控制寄存器 (TIM1_BDTR)	285
13.11.21	DLAMT触发延迟和防误触发寄存器 (TIM1_DLAMT)	287
13.11.22	DCR DMA控制寄存器 (TIM1_DCR)	288
13.11.23	DMAR DMA传输寄存器 (TIM1_DMAR)	289
13.11.24	CCTR1比较修调寄存器1 (TIM1_CCTR1)	290
13.11.25	CCTR2比较修调寄存器2 (TIM1_CCTR2)	290
13.11.26	CCTR3比较修调寄存器3 (TIM1_CCTR3)	290
13.11.27	CCTR4比较修调寄存器4 (TIM1_CCTR4)	291
13.11.28	CCTR5比较修调寄存器5 (TIM1_CCTR5)	291
13.11.29	CCOR1比较偏移寄存器1 (TIM1_CCOR1)	292
13.11.30	CCOR2比较偏移寄存器2 (TIM1_CCOR2)	292
13.11.31	CCOR3比较偏移寄存器3 (TIM1_CCOR3)	293
13.11.32	CCOR4比较偏移寄存器4 (TIM1_CCOR4)	293
13.11.33	CCOR5比较偏移寄存器5 (TIM1_CCOR5)	294
13.11.34	OCR比较偏移控制寄存器 (TIM1_OCR)	294
13.11.35	OMR输出模式寄存器 (TIM1_OMR)	295
13.11.36	TIM1互联配置寄存器1 (SysCtrl_EDU_CFG1)	296
13.11.37	TIM1互联配置寄存器4 (SysCtrl_EDU_CFG4)	296
14.	通用定时器TIM2	298
14.1	TIM2主要特性	298
14.2	管脚配置	299
14.3	时基单元	299
14.3.1	自动重装载寄存器	299
14.3.2	预分频器	300
14.4	计数器模式	300
14.4.1	向上计数模式	300
14.4.2	向下计数模式	301
14.4.3	中央对齐模式 (向上/向下计数)	302
14.5	时钟选择	303
14.5.1	内部时钟源 (TIM2_CLK)	304
14.5.2	外部时钟模式1	304
14.5.3	外部时钟模式2	305
14.6	捕获/比较通道	305
14.6.1	输入模块	306
14.6.2	输入捕获模式	306
14.6.3	PWM输入	307
14.6.4	输出模块	308
14.6.5	强制输出模式	309
14.6.6	输出比较模式	309
14.6.7	PWM模式	310

14.6.8 单脉冲模式	312
14.6.9 特殊情况：OCi快速使能	313
14.6.10 刹车功能	313
14.6.11 通过外部事件清除OCiREF	315
14.6.12 编码器接口模式	315
14.6.13 定时器输入异或功能	317
14.7 TIM2定时器与外部触发的同步	318
14.7.1 标准触发模式	318
14.7.2 复位模式	319
14.7.3 门控模式	319
14.7.4 触发复位模式	320
14.7.5 外部时钟模式2及触发模式	320
14.8 TIM2与其他定时器的联接	321
14.9 TIM2中断	322
14.10 TIM2寄存器描述	322
14.10.1 CR1控制寄存器1 (TIM2_CR1)	323
14.10.2 CR2控制寄存器2 (TIM2_CR2)	325
14.10.3 SMCR从模式控制寄存器 (TIM2_SMCR)	326
14.10.4 DIER DMA和中断控制寄存器 (TIM2_DIER)	328
14.10.5 SR状态寄存器 (TIM2_SR)	329
14.10.6 EGR事件产生寄存器 (TIM2_EGR)	331
14.10.7 CCMR1捕获模式寄存器1 (TIM2_CCMR1)	332
14.10.8 CCMR2捕获模式寄存器2 (TIM2_CCMR2)	335
14.10.9 CCER捕获使能寄存器 (TIM2_CCER)	336
14.10.10 CNT计数寄存器 (TIM2_CNT)	337
14.10.11 PSC预分频寄存器 (TIM2_PSC)	338
14.10.12 ARR自动重装载寄存器 (TIM2_ARR)	338
14.10.13 CCR1捕获寄存器1 (TIM2_CCR1)	339
14.10.14 CCR2捕获寄存器2 (TIM2_CCR2)	339
14.10.15 CCR3捕获寄存器3 (TIM2_CCR3)	340
14.10.16 CCR4捕获寄存器4 (TIM2_CCR4)	340
14.10.17 BDTR刹车和死区控制寄存器 (TIM2_BDTR)	341
14.10.18 DCR DMA控制寄存器 (TIM2_DCR)	342
14.10.19 DMAR DMA传输寄存器 (TIM2_DMAR)	343
14.10.20 TIM2互联配置寄存器8 (SysCtrl_EDU_CFG8)	344
14.10.21 TIM2互联配置寄存器9 (SysCtrl_EDU_CFG9)	345
15. 通用定时器TIM15/16/17	347
15.1 TIM15/16/17主要特性	347
15.1.1 TIM15主要特性	347
15.1.2 TIM16主要特性	348
15.1.3 TIM17主要特性	349
15.2 管脚配置	350
15.3 时基单元	351
15.3.1 自动重装载寄存器	351

15.3.2 预分频器	352
15.4 计数器模式	352
15.4.1 向上计数模式	352
15.4.2 向下计数模式	353
15.4.3 中央对齐模式（向上/向下计数）	354
15.5 时钟选择	355
15.5.1 内部时钟源（TIMx_CLK）	356
15.5.2 外部时钟模式1	356
15.5.3 外部时钟模式2	357
15.6 捕获/比较通道	358
15.6.1 输入捕获模式	359
15.6.2 PWM输入	359
15.6.3 强制输出模式	361
15.6.4 输出比较模式	361
15.6.5 PWM模式	362
15.6.6 单脉冲模式	364
15.6.7 特殊情况：OCi快速使能	365
15.6.8 编码器接口模式	365
15.6.9 互补输出和死区插入	367
15.6.10 重定向OCiREF到OCi或OCiN	368
15.6.11 刹车功能	368
15.7 TIM15/16/17定时器与外部触发的同步	371
15.7.1 标准触发模式	371
15.7.2 复位模式	371
15.7.3 门控模式	372
15.7.4 触发复位模式	373
15.7.5 外部时钟模式2及触发模式	373
15.7.6 触发延迟和防误触发功能	373
15.8 TIM15/16/17与其他定时器的联接	375
15.9 TIM15/16/17中断	376
15.10 TIM15/16/17寄存器描述	377
15.10.1 CR1控制寄存器1（TIM15/16/17_CR1）	377
15.10.2 CR2控制寄存器2（TIM15/16/17_CR2）	379
15.10.3 SMCR模式控制寄存器（TIM15/16/17_SMCR）	381
15.10.4 DIER DMA和中断控制寄存器（TIM15/16/17_DIER）	383
15.10.5 SR状态寄存器（TIM15/16/17_SR）	384
15.10.6 EGR事件产生寄存器（TIM15/16/17_EGR）	385
15.10.7 CCMR1捕获/比较模式寄存器1（TIM15/16/17_CCMR1）	387
15.10.8 CCER捕获/比较使能寄存器（TIM15/16/17_CCER）	390
15.10.9 CNT计数寄存器（TIM15/16/17_CNT）	392
15.10.10 PSC预分频寄存器（TIM15/16/17_PSC）	393
15.10.11 ARR自动重装载寄存器（TIM15/16/17_ARR）	393
15.10.12 RCR重复计数寄存器（TIM15_RCR）	393
15.10.13 CCR1捕获/比较寄存器1（TIM15/16/17_CCR1）	394

15.10.14	CCR2捕获/比较寄存器2 (TIM15/16_CCR2)	395
15.10.15	BDTR刹车和死区控制寄存器 (TIM15/16/17_BDTR)	395
15.10.16	DLAMT触发延迟和防误触发寄存器 (TIM15/16/17_DLAMT)	397
15.10.17	DCR DMA控制寄存器 (TIM15/16/17_DCR)	399
15.10.18	DMAR DMA传输寄存器 (TIM15/16/17_DMAR)	399
15.10.19	CCTR1比较修调寄存器1 (TIM15/16/17_CCTR1)	400
15.10.20	CCTR2比较修调寄存器2 (TIM15/16_CCTR2)	400
15.10.21	TIM15/16/17互联配置寄存器3 (SysCtrl_EDU_CFG3)	401
15.10.22	TIM17互联配置寄存器5 (SysCtrl_EDU_CFG5)	403
15.10.23	TIM15互联配置寄存器6 (SysCtrl_EDU_CFG6)	404
15.10.24	TIM16互联配置寄存器7 (SysCtrl_EDU_CFG7)	405
16.	定时器间的互联	406
17.	独立看门狗 (IWDG)	408
17.1	IWDG主要特性	408
17.2	功能描述	408
17.2.1	当窗口功能使能时配置IWDG	408
17.2.2	当窗口功能不使能时配置IWDG	409
17.2.3	寄存器访问保护	409
17.2.4	硬件看门狗功能	409
17.2.5	调试模式	409
17.3	IWDG寄存器描述	409
17.3.1	IWDG密钥寄存器 (IWDG_KR)	410
17.3.2	IWDG预分频寄存器 (IWDG_PR)	410
17.3.3	IWDG重装载寄存器 (IWDG_RLR)	411
17.3.4	IWDG状态寄存器 (IWDG_SR)	411
17.3.5	IWDG窗口寄存器 (IWDG_WINR)	412
18.	系统时基定时器 (SysTick)	413
18.1	主要特性	413
18.2	功能描述	413
18.3	SysTick寄存器描述	413
18.3.1	SysTick控制及状态寄存器 (SysTick_CTRL)	413
18.3.2	SysTick重装载值寄存器 (SysTick_RELOAD)	414
18.3.3	SysTick当前值寄存器 (SysTick_VAL)	414
18.3.4	SysTick校准值寄存器 (SysTick_CALIB)	415
19.	钟表定时器 (WT)	416
19.1	WT主要特性	416
19.2	管脚配置	416
19.3	功能描述	416
19.4	WT寄存器描述	417
19.4.1	WT控制寄存器 (WTCON)	417
19.4.2	WT定时寄存器 (T8/T8RL)	418
20.	I2C接口	420
20.1	I2C主要特性	420
20.2	管脚配置	420

20.3 功能描述	420
20.3.1 操作模式	422
20.3.2 尖峰抑制	429
20.3.3 超快速模式	429
20.3.4 I2C_CLK时钟配置	429
20.3.5 DMA操作	431
20.3.6 SMBus模式	432
20.4 I2C中断	436
20.5 I2C寄存器描述	437
20.5.1 I2C控制寄存器 (IC_CON)	438
20.5.2 I2C目标地址寄存器 (IC_TAR)	440
20.5.3 I2C从机地址寄存器 (IC_SAR)	440
20.5.4 I2C数据命令寄存器 (IC_DATA_CMD)	441
20.5.5 I2C标准模式时钟高字节计数器 (IC_SS_SCL_HCNT)	442
20.5.6 I2C标准模式时钟低字节计数器 (IC_SS_SCL_LCNT)	442
20.5.7 I2C快速模式时钟高字节计数器 (IC_FS_SCL_HCNT)	442
20.5.8 I2C快速模式时钟低字节计数器 (IC_FS_SCL_LCNT)	443
20.5.9 I2C中断状态寄存器 (IC_INTR_STAT)	443
20.5.10 I2C中断屏蔽寄存器 (IC_INTR_MASK)	444
20.5.11 I2C中断原始状态寄存器 (IC_RAW_INTR_STAT)	445
20.5.12 I2C接收FIFO阈值寄存器 (IC_RX_TL)	446
20.5.13 I2C发送FIFO阈值寄存器 (IC_TX_TL)	447
20.5.14 I2C中断清除寄存器 (IC_CLR_INTR)	447
20.5.15 I2C清除RX_UNDER中断寄存器 (IC_CLR_RX_UNDER)	448
20.5.16 I2C清除RX_OVER中断寄存器 (IC_CLR_RX_OVER)	448
20.5.17 I2C清除TX_OVER中断寄存器 (IC_CLR_TX_OVER)	448
20.5.18 I2C清除RD_REQ中断寄存器 (IC_CLR_RD_REQ)	449
20.5.19 I2C清除TX_ABRT中断寄存器 (IC_CLR_TX_ABRT)	449
20.5.20 I2C清除RX_DONE中断寄存器 (IC_CLR_RX_DONE)	449
20.5.21 I2C清除ACTIVITY中断寄存器 (IC_CLR_ACTIVITY)	450
20.5.22 I2C清除STOP_DET中断寄存器 (IC_CLR_STOP_DET)	450
20.5.23 I2C清除START_DET中断寄存器 (IC_CLR_START_DET)	450
20.5.24 I2C清除GEN_CALL中断寄存器 (CLR_GEN_CALL)	451
20.5.25 I2C使能寄存器 (IC_ENABLE)	451
20.5.26 I2C状态寄存器 (IC_STATUS)	452
20.5.27 I2C发送FIFO数据量寄存器 (IC_TXFLR)	453
20.5.28 I2C接收FIFO数据量寄存器 (IC_RXFLR)	453
20.5.29 I2C SDA维持时间长度寄存器 (IC_SDA_HOLD)	454
20.5.30 I2C发送丢弃源寄存器 (IC_TX_ABRT_SOURCE)	454
20.5.31 I2C生成从机数据NACK寄存器 (IC_SLV_DATA_NACK_ONLY)	456
20.5.32 I2C DMA控制寄存器 (IC_DMA_CR)	457
20.5.33 I2C DMA发送数据水平寄存器 (IC_DMA_TDLR)	457
20.5.34 I2C DMA接收数据水平寄存器 (IC_DMA_RDLR)	457
20.5.35 I2C SDA建立寄存器 (IC_SDA_SETUP)	458

20.5.36 I2C广呼应答寄存器 (IC_ACK_GENERAL_CALL)	458
20.5.37 I2C使能状态寄存器 (IC_ENABLE_STATUS)	459
20.5.38 I2C标准模式和快速模式尖峰抑制寄存器 (IC_FS_SPKLEN)	460
20.5.39 I2C清除RESTART_DET寄存器 (IC_CLR_RESTART_DET)	460
20.5.40 I2C清除TIMEOUT寄存器 (IC_CLR_TIMEOUT)	460
20.5.41 I2C清除SMBUS_ALERT寄存器 (IC_CLR_SMBUS_ALERT)	461
20.5.42 I2C清除PEC_ERROR寄存器 (IC_CLR_PEC_ERROR)	461
20.5.43 I2CTIMEOUT计时寄存器 (IC_TIMEOUT_DATA)	461
20.5.44 I2C PEC数据寄存器 (IC_PEC_DATA)	462
20.5.45 I2C清除ADDR0_MATCH寄存器 (IC_CLR_ADDR0_MATCH)	462
20.5.46 I2C清除ADDR1_MATCH寄存器 (IC_CLR_ADDR1_MATCH)	463
21. USART/LIN接口	464
21.1 USART主要特性	464
21.2 管脚配置	464
21.3 功能描述	465
21.3.1 时钟	465
21.3.2 波特率	466
21.3.3 USART帧格式	467
21.3.4 数据的发送	467
21.3.5 数据的接收	468
21.3.6 USART硬件流控制	468
21.3.7 单线半双工模式	469
21.3.8 同步发送模式	469
21.3.9 低功耗唤醒模式 (LPUSART)	469
21.3.10 LIN模式	470
21.3.11 冲突检测	475
21.3.12 DMA接口	475
21.4 USART中断	475
21.5 USART寄存器描述	477
21.5.1 USART0/1数据寄存器 (USARTDR)	478
21.5.2 USART0/1接收状态寄存器 (USARTRSR)	479
21.5.3 USART0/1控制寄存器2 (USARTCR2)	479
21.5.4 USART0/1控制寄存器3 (USARTCR3)	480
21.5.5 USART0/1标志寄存器 (USARTFR)	481
21.5.6 USART0/1整数波特率寄存器 (USARTIBRD)	482
21.5.7 USART0/1小数波特率寄存器 (USARTFBRD)	482
21.5.8 USART0/1线控制寄存器 (USARTLCR_H)	483
21.5.9 USART0/1控制寄存器 (USARTCR)	484
21.5.10 USART0/1中断FIFO触发水平选择寄存器 (USARTIFLS)	485
21.5.11 USART0/1中断屏蔽寄存器 (USARTIMSC)	486
21.5.11 USART0/1原始中断状态寄存器 (USARTRIS)	487
21.5.12 USART0/1屏蔽后中断状态寄存器 (USARTMIS)	488
21.5.13 USART0/1中断清除寄存器 (USARTICR)	489
21.5.14 USART0/1 DMA控制寄存器 (USARTDMACR)	490



21.5.15	USART0/1 LIN控制寄存器1 (LINCON1)	490
21.5.16	USART0/1 LIN控制寄存器2 (LINCON2)	492
21.5.17	USART0/1 帧控制寄存器 (FRAMECON)	493
21.5.18	USART0/1 LIN状态寄存器 (LINSTS)	493
21.5.19	USART0/1数据配置寄存器 (DATCON)	495
21.5.20	USART0/1波特率侦测寄存器 (BRDMSU)	496
21.5.21	USART0/1波特率侦测上限寄存器 (BRDUPPER)	496
21.5.22	USART0/1波特率侦测下限寄存器 (BRDLOWER)	496
21.5.23	USART0/1 LIN BREAK定时寄存器 (LINBTIMER)	497
21.5.24	USART0/1 LIN帧头定时寄存器 (LINHTIMER)	497
21.5.25	USART0/1中断屏蔽寄存器1 (USARTIMSC1)	498
21.5.26	USART0/1原始中断状态寄存器1 (USARTRIS1)	498
21.5.27	USART0/1屏蔽中断状态寄存器1 (USARTMIS1)	499
21.5.28	USART0/1中断清除寄存器1 (USARTICR1)	500
22.	SPI接口	501
22.1	SPI主要特性	501
22.2	管脚配置	501
22.3	功能描述	501
22.4	SPI操作	502
22.4.1	串行时钟的生成	502
22.4.2	时钟频率与模式选择	503
22.4.3	配置SPICR0控制寄存器	503
22.4.4	配置SPICR1控制寄存器	503
22.4.5	配置SPICR2控制寄存器	503
22.4.6	SPI帧格式	504
22.4.7	主从配置实例	507
22.4.8	DMA接口	508
22.5	SPI中断	509
22.6	SPI寄存器描述	510
22.6.1	SPI控制寄存器0 (SPICR0)	510
22.6.2	SPI控制寄存器1 (SPICR1)	511
22.6.3	SPI控制寄存器2 (SPICR2)	512
22.6.4	SPI数据寄存器 (SPIDR)	513
22.6.5	SPI状态寄存器 (SPISR)	513
22.6.6	SPI时钟分频寄存器 (SPICPSR)	514
22.6.7	SPI中断屏蔽寄存器 (SPIIMSC)	514
22.6.8	SPI原始中断状态寄存器 (SPIRIS)	515
22.6.9	SPI屏蔽后中断状态寄存器 (SPIMIS)	516
22.6.10	SPI中断清除寄存器 (SPIICR)	516
22.6.11	SPI DMA控制寄存器 (SPIDMACR)	517
23.	CRC模块	518
23.1	CRC主要特性	518
23.2	功能描述	518
23.3	CRC寄存器描述	518

23.3.1	CRC控制寄存器 (CRC_CTRL)	518
23.3.2	CRC数据寄存器 (CRC_DATA)	519
23.3.3	CRC结果寄存器 (CRC_RESULT)	520
24.	DIV除法器	521
24.1	DIV主要特性	521
24.2	功能描述	521
24.3	DIV中断	521
24.4	DIV寄存器描述	521
24.4.1	DIV控制寄存器 (DIV_CTRL)	521
24.4.2	DIV被除数寄存器 (DIV_ALO)	522
24.4.3	DIV除数寄存器 (DIV_B)	523
24.4.4	DIV商寄存器 (DIV_QUOT)	523
24.4.5	DIV余数寄存器 (DIV_REM)	523
25.	CORDIC模块	524
25.1	CORDIC主要特性	524
25.2	功能描述	524
25.3	CORDIC中断	526
25.4	CORDIC寄存器描述	526
25.4.1	CORDIC控制寄存器 (CORDIC_CTRL)	526
25.4.2	CORDIC配置寄存器 (CORDIC_CFG)	527
25.4.3	CORDIC X输入寄存器 (CORDIC_Xi)	528
25.4.4	CORDIC Y输入寄存器 (CORDIC_Yi)	528
25.4.5	CORDIC Z输入寄存器 (CORDIC_Zi)	528
25.4.6	CORDIC X输出寄存器 (CORDIC_Xo)	529
25.4.7	CORDIC Y输出寄存器 (CORDIC_Yo)	529
25.4.8	CORDIC Z输出寄存器 (CORDIC_Zo)	529
26.	调试模块	530
26.1	主要特性	530
26.2	调试复用接口	530
26.2.1	SWD复用接口	530
26.2.2	SW-DP功能复用说明	531
26.2.3	SWD管脚的内部上拉或下拉	531
26.3	SWD通讯	531
26.3.1	SWD协议介绍	531
26.3.2	SWD协议时序	531
26.3.3	SW-DP状态机 (复位、空闲、ID码)	532
26.3.4	DP及AP读写操作	532
26.3.5	SW-DP寄存器	533
26.3.6	SW-AP寄存器	533
26.4	内核的调试	534
26.5	断点模块BPU	534
26.6	数据观测点DWT	534
26.6.1	数据监控功能说明	534
26.6.2	DWT的程序指针采样寄存器	534

26.7	MCU调试单元	535
26.7.1	低功耗模式下的调试	535
26.7.2	定时器、看门狗以等模块的调试支持.....	535
26.7.3	调试模式控制寄存器	535
27.	预驱概述	537
27.1	预驱LND31A01	537
27.1.1	LND31A01开关特性和死区时间	538
27.1.2	LND31A01 VCC端电源电压.....	539
27.1.3	LND31A01逻辑信号要求和特性	539
27.2	预驱LND31A02	540
27.3	预驱LND32B01	542
27.3.1	LND32B01开关特性和死区时间	543
27.3.2	LND32B01 VCC端电源电压.....	545
27.3.3	LND32B01逻辑信号要求和特性	545
27.4	预驱LND33A04	546
27.4.1	LND33A04低边电源VCC和欠压锁定	547
27.4.2	LND33A04高边电源VBS和欠压锁定	548
27.4.3	LND33A04低边和高边逻辑输入控制：HIN和LIN	548
27.4.4	LND33A04功率管直通保护	548
27.4.5	LND33A04死区时间	549
27.5	预驱LND36C01	549
27.5.1	LND36C01开关特性和死区时间	551
27.5.2	LND36C01 VCC端电源电压.....	551
27.5.3	LND36C01逻辑信号要求和特性	551
27.6	预驱LND33A01	552
27.6.1	LND33A01开关特性和死区时间	554
27.6.2	LND33A01 VCC端电源电压.....	555
27.6.3	LND33A01 VB端电源电压.....	556
27.6.4	LND33A01逻辑信号要求和特性	557
28.	修订历史	559



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	48
表3-7 FLASH选项字节	49
表3-8 FLASH校准字节	54
表4-1 模块电源分配	57
表4-2 低功耗模式总结	59
表4-3 立刻进入睡眠	60
表4-4 退出后睡眠	60
表4-5 停机模式	61
表4-6 超低功耗停机模式	62
表4-7 电源配置相关寄存器表	62
表5-1 芯片配置相关寄存器表	69
表5-2 系统配置相关寄存器表	82
表6-1 LCM32F06X外设互联矩阵	90
表6-2 互联配置相关寄存器表	91
表7-1 GPIO端口说明	108
表7-2 引脚排列表中使用的图例/缩略语	109
表7-3 LCM32F06X引脚定义及模拟复用功能（AN）映射	109
表7-4 端口A可选复用功能（AF）映射	114
表7-5 端口B可选复用功能（AF）映射	115
表7-6 GPIO相关寄存器表	115
表8-1 DMA相关寄存器表	131
表9-1 中断和异常向量表	150
表9-2 EXTI相关寄存器表	153
表10-1 ADC外部I/O通道表	162
表10-2 硬件触发连接表	172
表10-3 AWD通道选择	173
表10-4 ADC中断事件	174
表10-5 ADC相关寄存器表	174
表11-1 DAC管脚信号	189
表11-2 DAC触发选择	191
表11-3 比较器管脚配置	195
表11-4 HALL_MID端口复用表	205
表11-5 DAC/ACMP相关寄存器表	205
表12-1 OPA管脚配置	224
表12-2 OPA相关寄存器表	230



表13-1 定时器特性比较	237
表13-2 TIM1管脚配置	238
表13-3 TIM1触发与级联表	263
表13-4 TIM1相关寄存器表	263
表13-5 带刹车功能的互补输出OCI和OCIN的控制	280
表14-1 TIM2管脚配置	299
表14-2 计数方向与编码器信号的关系	316
表14-3 TIM2触发与级联表	321
表14-4 TIM2相关寄存器表	322
表15-1 TIM15/16/17管脚配置	350
表15-2 计数方向与编码器信号的关系	366
表15-3 TIM15触发与级联表	375
表15-4 TIM16触发与级联表	375
表15-5 TIM17触发与级联表	376
表15-6 TIM15/16/17相关寄存器表	377
表15-7 带刹车功能的互补输出OCI和OCIN的控制	392
表16-1 定时器间的互联	406
表17-1 IWDG相关寄存器表	409
表18-1 SYSTICK相关寄存器表	413
表19-1 WT管脚配置	416
表19-2 WT相关寄存器表	417
表20-1 I2C管脚配置	420
表20-2 10位地址的前五位含义表	425
表20-3 具有高/低计数值的所有模式的最小I2C_CLK值	431
表20-4 SMBUS超时规范	434
表20-5 各种IC_CLK频率的IC_TIMEOUT_DATA配置示例1	434
表20-6 各种IC_CLK频率的IC_TIMEOUT_DATA配置示例2	434
表20-7 各种IC_CLK频率的IC_TIMEOUT_DATA配置示例3	435
表20-8 I2C相关寄存器表	437
表21-1 USART0/1支持的功能	464
表21-2 USART管脚配置	464
表21-3 波特率表（USART_CLK为16MHZ）	466
表21-4 波特率表（USART_CLK为24MHZ）	466
表21-5 接收FIFO功能位	468
表21-6 硬件流控的使能控制位	469
表21-7 USARTRIS1寄存器功能位	474
表21-8 USART中断请求	477
表21-9 USART相关寄存器表	477
表21-10 奇偶校验位配置真值表	484
表22-1 SPI管脚配置	501
表22-2 SSP DMABREQ的触发点，用于发送和接收FIFO	509
表22-3 SPI相关寄存器表	510
表23-1 CRC相关寄存器表	518
表24-1 DIV相关寄存器表	521

表25-1 CORDIC旋转系统详细结果.....	524
表25-2 CORDIC相关寄存器表.....	526
表26-1 SW调试接口的管脚.....	531
表26-2 包传输请求（8 BITS）.....	532
表26-3 应答响应（3 BITS）.....	532
表26-4 数据传输（33 BITS）.....	532
表26-5 SW-DP寄存器.....	533
表26-6 通过输入地址A[3:2]选择的调试端口寄存器.....	533
表26-7 内核调试寄存器.....	534
表27-1 LND31A01信号逻辑真值表.....	540
表27-2 LND32B01信号逻辑真值表.....	546
表27-3 LND36C01信号逻辑真值表.....	552
表27-4 LND33A01信号逻辑真值表.....	558



图片目录

图2-1 LCM32F06X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/半页擦除流程	38
图3-3 FLASH的块擦除操作流程	39
图4-1 电源供电结构	57
图4-2 上电和下电复位图	58
图5-1 复位框图	67
图5-2 LCM32F06X时钟框图	68
图6-1 LCM32F06X总线矩阵	90
图7-1 GPIO接口的基本结构	108
图8-1 DMA结构框图	125
图8-2 DMA连接框图	126
图8-3 DMA中断逻辑	127
图8-4 示例一：外设A配置1且作为目标，2次源的传输，2次目标的连续请求和传输	128
图8-5 示例二：外设A配置2且作为源，3次源的连续请求和传输，2次目标的传输	128
图8-6 示例三：外设B配置1且作为目标，3次源的传输，1次目标的连续请求和传输，2次目标的单次请求和传输	129
图8-7 示例四：外设B配置2且作为源，2次源的连续请求和传输，2次源的单次请求和传输，2次目标的传输 ..	129
图8-8 示例五：SRAM 0作为源，SRAM 1作为目标，2次源的传输，2次目标的传输	129
图8-9 DMA请求映射	130
图9-1 外部中断/事件控制器框图	152
图9-2 外部中断通用I/O映射	153
图10-1 ADC框图	159
图10-2 ADC启动与关闭时序图	160
图10-3 START和STOP工作时序图	161
图10-4 START和强制STOP工作时序图	161
图10-5 VTS温度曲线图	164
图10-6 调整脉冲时序图	165
图10-7 单次采样转换时序图	165
图10-8 连续采样转换时序图	167
图10-9 SOC优先级排序的时序	168
图10-10 SOC优先级抢占的时序图	168
图10-11 SOC和ADC通道的触发关系图	169
图10-12 SOC状态跳转图	169
图10-13 SOC状态时序图	170
图10-14 SOC挂起溢出	170
图10-15 硬件触发源框图	172
图10-16 模拟看门狗监控区域	173
图10-17 DMA搬运ADC数据的过程	173
图11-1 DAC框图	190
图11-2 DAC LFSR计算方式	191

图11-3 带LFSR的DAC波形发生过程	192
图11-4 DAC三角波波形	192
图11-5 三角波形转换过程（软件触发）	192
图11-6 DAC和ACMP内部框图	194
图11-7 ACMP触发示意图	200
图11-8 ACMP0单次序列模式时序图	201
图11-9 ACMP0断续序列模式时序图	202
图11-10 ACMP1连续序列模式时序图	203
图11-11 比较器迟滞	203
图11-12 比较器输出示意图	204
图11-13 ACMP内部分压电阻串模块结构图	204
图11-14 HALL_MID框图	205
图12-1 OPA内部结构和外部互联	228
图12-2 OPA2典型应用配置	229
图13-1 PWM专用定时器TIM1框图	238
图13-2 定时器时基单元	239
图13-3 预分频系数从1变2的计数器时序	240
图13-4 向上计数溢出	241
图13-5 计数器在二分频时钟频率下的时序图	241
图13-6 向下计数溢出	242
图13-7 中央对齐溢出	242
图13-8 计数器时序图，内部时钟分频为1，TIM1_ARR=0X6	243
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	244
图13-10 时钟/触发控制器框图	244
图13-11 TIM1外部时钟框图	245
图13-12 外部时钟模式1下的控制电路	245
图13-13 外部触发时钟框图	246
图13-14 外部时钟模式2下的控制电路	246
图13-15 比较通道1的主电路	247
图13-16 输出模块框图	247
图14-1 通用定时器TIM2框图	298
图14-2 预分频系数从1变2的计数器时序	300
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	301
图14-5 向下计数溢出	302
图14-6 中央对齐溢出	302
图14-7 计数器时序图，内部时钟分频为1，TIM2_ARR=0X6	303
图14-8 时钟/触发控制器框图	303
图14-9 TIM2外部时钟框图	304
图14-10 外部时钟模式1下的控制电路	304
图14-11 外部触发时钟框图	305
图14-12 外部时钟模式2下的控制电路	305
图14-13 捕获/比较通道1的主电路	306
图14-14 输入模块框图	306



图14-15 TIM2通道1的输入	307
图14-16 PWM输入信号测量	308
图14-17 PWM输入信号测量实例	308
图14-18 输出模块框图	309
图14-19 输出比较模式，翻转OC1	310
图14-20 TIM2_ARR=8时边沿对齐的PWM波形实例	311
图14-21 中央对齐模式PWM（TIM2_ARR=8）	312
图14-22 单脉冲模式	313
图14-23 TIM2刹车通道框图	314
图14-24 刹车对应的输出波形	315
图14-25 编码器接口模式示例	317
图14-26 TI1FP1反相的编码器接口模式示例	317
图14-27 TIM2触发输入功能框图	318
图14-28 标准触发模式下的时序图	318
图14-29 门控模式下的时序图	320
图14-30 外部时钟模式2+触发模式下的时序图	321
图15-1 通用定时器TIM15框图	348
图15-2 通用定时器TIM16框图	349
图15-3 通用定时器TIM17框图	350
图15-4 定时器时基单元	351
图15-5 预分频系数从1变2的计数器时序	352
图15-6 向上计数溢出	353
图15-7 计数器在二分频时钟频率下的时序图	353
图15-8 向下计数溢出	354
图15-9 中央对齐溢出	354
图15-10 计数器时序图，内部时钟分频为1，TIMX_ARR=0X6	355
图15-11 时钟/触发控制器框图	355
图15-12 TI2外部时钟框图	356
图15-13 外部时钟模式1下的控制电路	357
图15-14 外部触发时钟框图	357
图15-15 外部时钟模式2下的控制电路	358
图15-16 捕获/比较通道1的主电路	358
图15-17 TIM15通道1的输入	359
图15-18 PWM输入信号测量	360
图15-19 PWM输入信号测量实例	360
图15-20 输出比较模式，翻转OC1	361
图15-21 TIMX_ARR=8时边沿对齐的PWM波形实例	363
图15-22 中央对齐模式PWM（TIMX_ARR=8）	364
图15-23 单脉冲模式	365
图15-24 编码器接口模式示例	366
图15-25 TI1FP1反相的编码器接口模式示例	367
图15-26 带死区插入的互补输出	367
图15-27 死区波形延迟大于负脉冲	368
图15-28 死区波形延迟大于正脉冲	368



图15-29 TIMX刹车通道框图.....	369
图15-30 刹车对应的输出	370
图15-31 标准触发模式下的时序图	371
图15-32 复位模式下的时序图	372
图15-33 门控模式下的时序图	372
图15-34 外部时钟模式2且触发模式下的时序图	373
图15-35 防误触发窗口从延迟的触发输入开始	374
图15-36 防误触发窗口从触发输入开始	374
图17-1 独立看门狗IWDG框图.....	408
图19-1 WT框图.....	417
图20-1 I2C框图	421
图20-2 I2C基本传输协议	421
图20-3 I2C 7位地址格式.....	425
图20-4 I2C 10位地址格式	425
图20-5 主机发送：发送FIFO空/STOP产生时序图	426
图20-6 主机接收：发送FIFO空/STOP产生时序图	426
图20-7 主机发送：IC_DATA_CMD的RESTART置位	427
图20-8 主机接收：IC_DATA_CMD的RESTART置位	427
图20-9 主机发送：IC_DATA_CMD的STOP置位/发送FIFO非空	427
图20-10 主机发送：FIFO为空后载入的首字节，RESTART置位	427
图20-11 主机接收：IC_DATA_CMD的STOP置位/发送FIFO非空	428
图20-12 主机接收：允许TX FIFO为空/RESTART置位后，加载第一个命令	428
图20-13 两个主机仲裁的时序	428
图20-14 尖峰抑制示例图	429
图20-15 DMA连续传输分解	432
图20-16 DMA单个和连续传输分解	432
图20-17 SMBUS数据传输示例	433
图20-18 超时时间定义	433
图20-19 时钟扩展时间定义	434
图20-20 带PEC的数据读/写传输示例.....	435
图20-21 7位可寻址设备响应ARA	435
图20-22 UDID组成.....	436
图21-1 USART框图	465
图21-2 波特率分配器	466
图21-3 USART字节帧	467
图21-4 两个相似设备间的硬件流控	468
图21-5 LIN的帧结构.....	470
图21-6 帧头发送中LIN主机操作	471
图21-7 响应传输操作	471
图21-8 波特率同步示例	472
图21-9 网络唤醒示意图	472
图21-10 环回模式	473
图21-11 自测模式	473
图21-12 DMA单请求和DMA连续请求与DMACLR间的时序关系	475

图22-1 SPI结构框图	502
图22-2 SPO=0, SPH=0, 单字节传输时序	504
图22-3 SPO=0, SPH=0, 多字节传输时序	504
图22-4 SPO=0, SPH=1, 包括单次和连续传输	505
图22-5 SPO=1, SPH=0单字节传输	506
图22-6 SPO=1, SPH=0多字节传输	506
图22-7 SPO=1, SPH=1, 包括单次和连续传输	507
图22-8 SPI主机与SPI从机	507
图22-9 SPI主机和外部SPI从机	508
图22-10 外部SPI主机与SPI从机	508
图22-11 DMA传输波形	509
图26-1 调试模块结构图	530
图27-1 预驱LND31A01结构框图	538
图27-2 LND31A01输出LO开关时间	538
图27-3 LND31A01输出HO开关时间	539
图27-4 LND31A01死区时间	539
图27-5 LND31A01信号的逻辑功能	540
图27-6 预驱LND31A02结构框图	541
图27-7 LND31A02输入输出时序	542
图27-8 预驱LND32B01结构框图	543
图27-9 LND32B01输出LO开关时间	544
图27-10 LND32B01输出HO开关时间	544
图27-11 LND32B01死区时间	544
图27-12 LND32B01自举电路结构	545
图27-13 LND32B01信号的逻辑功能	546
图27-14 LND33A04 VCC电源UVLO工作区域	547
图27-15 LND33A04 VBS电源UVLO工作区域	548
图27-16 LND33A04功率管直通保护	549
图27-17 LND33A04死区保护	549
图27-18 预驱LND36C01结构框图	550
图27-19 LND36C01输出LO开关时间	551
图27-20 LND36C01输出HO开关时间	551
图27-21 LND36C01信号的逻辑功能	552
图27-22 预驱LND33A01结构框图	554
图27-23 LND33A01输出LO开关时间	555
图27-24 LND33A01输出HO开关时间	555
图27-25 LND33A01死区时间	555
图27-26 LND33A01 VCC电源UVLO工作区域	556
图27-27 LND33A01 VBS电源UVLO工作区域	557
图27-28 LND33A01信号的逻辑功能	557



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 LCM32F06X 系列芯片的完整信息。LCM32F06X 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：

LCM32F067 系列				
产品名		可用 I/O 数量		封装特性
LCM32F067H8S8		22		SSOP24
LCM32F067H8W8		22		QFN24（3*3）
LCM32F067I8P8		25		TSSOP28
LCM32F067I8V8		25		QFN28（4*4）
LCM32F067B8N8		6		SOP8
LCM32F067E8N8		14		SOP16
LCM32F063 系列				
产品名		可用 I/O 数量		封装特性
LCM32F063H8S8		22		SSOP24
LCM32F063H8V8		22		QFN24（4*4）
LCP067 NP 预驱系列				
产品名	预驱特性	预驱名称	可用 I/O 数量	封装特性
LCP067AK31GU8	三相 P/N MOS 管栅极驱动	LND31A02	19	QFN32（5*5）
LCP067AH31ES8		LND31A01	14	SSOP24
LCP067AH31GS8		LND31A02	14	SSOP24
LCP067AE30GN8		LND31A02	7	SOP16
LCP067AK31EV8		LND31A01	19	QFN32（4*4）
LCP067 NN 预驱系列				
产品名	预驱特性	预驱名称	可用 I/O 数量	封装特性
LCP067BK32EU8	三相独立半桥驱动	LND32B01	16	QFN32（5*5）
LCP067BU32EV8		LND32B01	19	QFN36（4*4）
LCP067AT33EU8	三相独立半桥驱动+LDO	LND33A01	19	QFN40（5*5）
LCP067AT33XU8		LND33A04	18	QFN40（5*5）
LCP067CC36EU8		LND36C01	16	QFN38（7*7）

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

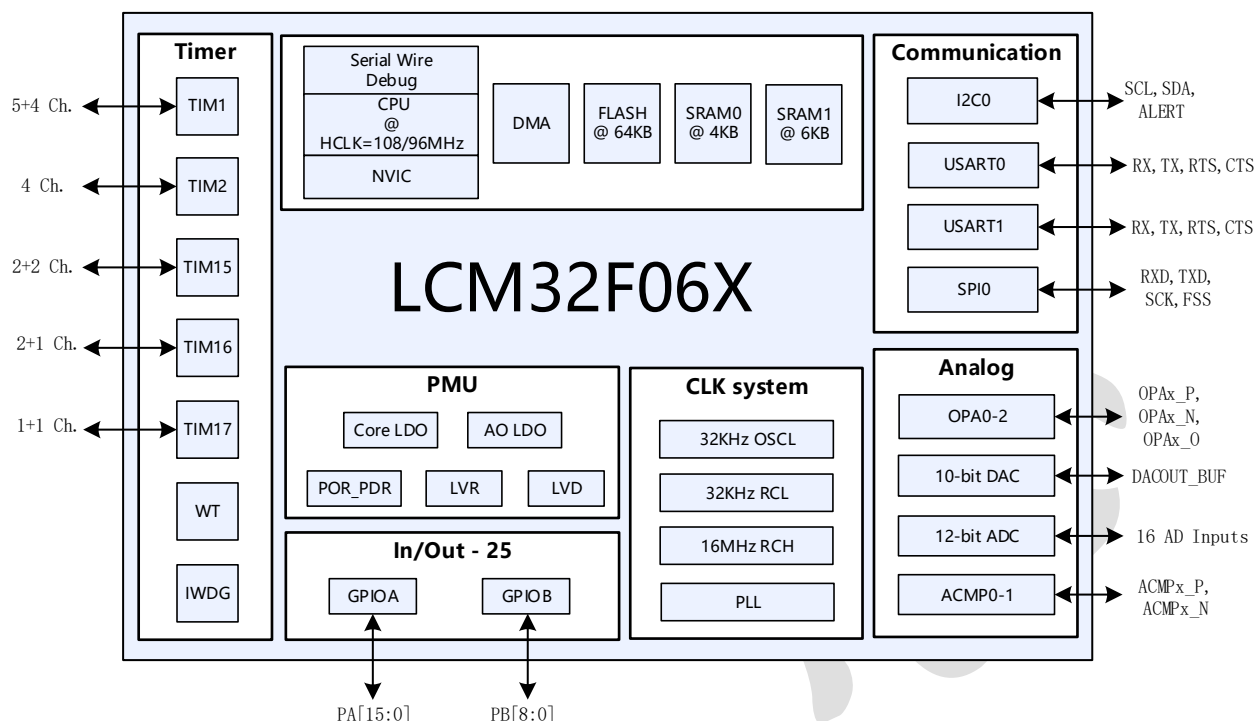


图 2-1 LCM32F06X 模块框图

系统的主要结构：

- 2 个主机模块
 - 32 位 CORTEX-M0 内核
 - 2 通道 DMA
- 6 个从机模块
 - Flash 模块及存储总线
 - RAM 接口单元，分成两组，4KB 和 6KB
 - 两组 AHB 总线，AHB0 直接控制 GPIO 及 DMA，AHB1 用于协处理单元的连接
 - 一组 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 6 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。

LCM32F06X 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

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：

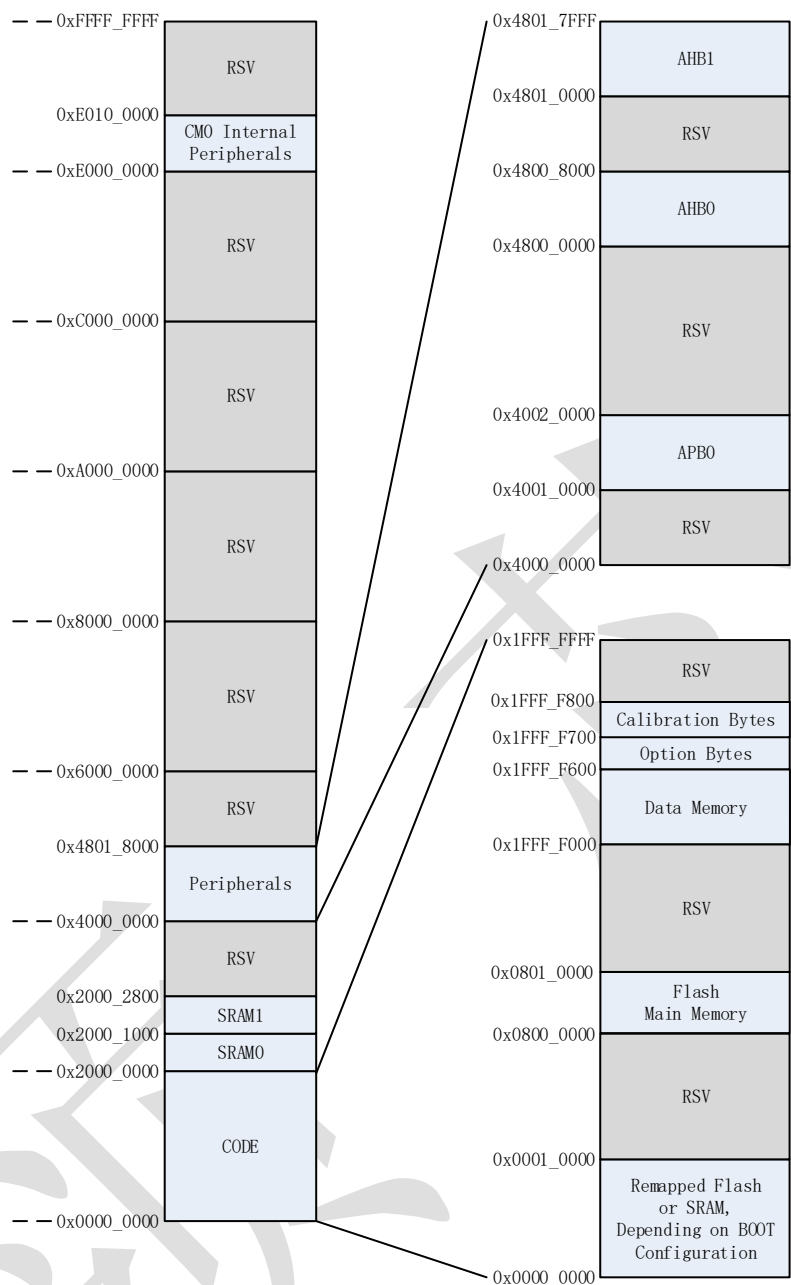


图 2-2 系统存储空间分配表

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	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	USART0
	0x4001_5000 - 0x4001_5FFF	4KB	USART1
	0x4001_6000 - 0x4001_6FFF	4KB	保留
	0x4001_7000 - 0x4001_73FF	1KB	CHIPCTRL
	0x4001_7400 - 0x4001_77FF	1KB	IWDG
	0x4001_7800 - 0x4001_7BFF	1KB	WT
	0x4001_7C00 - 0x4001_7FFF	1KB	ANA CTRL
	0x4001_8000 - 0x4001_8FFF	4KB	SPIO
	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	TIM2
AHB0	0x4002_0000 - 0x47FF_FFFF	~128MB	保留
	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	保留
	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	CORDIC
	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	保留
	0x4801_5000 - 0x4801_5FFF	4KB	DIV
	0x4801_6000 - 0x4801_6FFF	4KB	保留
	0x4801_7000 - 0x4801_7FFF	4KB	保留

2.4 嵌入式 SRAM

LCM32F06X 包含两个独立的 SRAM Bank，支持同时读写。两个 Bank 分别为 4KBytes 和 6KBytes（位宽 32bit），一共 10KBytes。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。

LCM32F06X 的 RAM 支持奇偶校验（Parity），当 SRAM 发生奇偶校验错误时，可进入普通中断或进入 HardFault 异常。注意，如果使用奇偶校验功能，LCM32F06X 可达到的最高频率会下降。

2.5 启动模式

启动时，通过 BOOT0 pin 和 BOOT_SEL bit 选择 2 种启动模式：

- 从 Main Memory 启动
- 从 SRAM 启动

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
0	1	1	x	SRAM 启动
x	x	0	1	Main Flash 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启动模式。

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。

2 种启动模式：Main Flash memory 启动、SRAM 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_FFFF），但主存储 Flash 空间也可以通过其原先设定的地址（0x0800_0000 - 0x0800_FFFF）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。
- **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_0000 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 SRAM 空间开始执行（由 nBOOT1 bit 确定具体位置）。

向量地址的重定位

LCM32F06X 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

LCM32F06X 内置 1 个 UID，位于 `0x1FFF_F7B0`，长度 128 位。

3. 嵌入式 64KB Flash 及选项字节说明

3.1 Flash 的主要特性

- 该系列产品内置嵌入式 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 24MHz 之间时，0 个等待周期；在 24MHz 到 48MHz 之间时，1 个等待周期；在 48MHz 到 72MHz 之间时，2 个等待周期；在 72MHz 到 96MHz 之间时，3 个等待周期；在大于 96MHz 时，4 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，可实现指令预取。
- 写保护防止 Main Memory 被篡改，以 2KBytes（4 个 Page）为一个区间，一共有 32 个独立的写保护区间。
- 读保护分为三个保护等级；以 8KBytes（16 个 Page）为一个区间，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，不能通过调试端口、SRAM 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，调试功能和从 SRAM 里执行程序功能被禁止。
- Flash 支持硬件奇偶校验
 - 全片范围都支持校验，使用奇校验，1Byte 信息对应 1 位校验位。
 - 检验错误可触发中断：CPU 取指令校验错误默认进入 Hardfault 中断；取数据校验错误默认进入普通 Flash 中断，可配置进入 Hardfault 中断。
 - 可通过注入寄存器在 Flash 编程时人为注入错误，以此检验校验功能是否正常。
 - 发生奇偶校验错误后可从寄存器内读取出错的地址、数据、校验位等信息。
 - 奇偶校验相关寄存器参见章节 5.3。

3.2 Flash 模块的功能性说明

3.2.1 存储模块的组织形式

Flash 存储单元的位宽为 32 位，在功能上被划分为三个部分：

- 64KBytes 的 Main Memory（主存储区），分成 128 个 Page，每个 Page 大小为 512Bytes；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。
- 256Bytes 的 Option Bytes（选项字节区），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。
- 256Bytes 的 Calibration Bytes（校准字节区），用于存储校准信息和设备配置信息等。
- 1.5 KBytes 的 Data Memory（数据存储区），用于存储用户数据，可作为模拟 EEPROM 使用。

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
主存储区	0x0800 0000 - 0x0800 01FF	512B	Page 0	WSector0	RSector 0
	0x0800 0200 - 0x0800 03FF	512B	Page 1		
	0x0800 0400 - 0x0800 05FF	512B	Page 2		
	0x0800 0600 - 0x0800 07FF	512B	Page 3		
	0x0800 0800 - 0x0800 09FF	512B	Page 4	WSector1	

	0x0800 0A00 - 0x0800 0BFF	512B	Page 5			
	0x0800 0C00 - 0x0800 0DFF	512B	Page 6			
	0x0800 0E00 - 0x0800 0FFF	512B	Page 7			
	⋮	⋮	⋮	⋮	⋮	
	0x0800 E000 - 0x0800 E1FF	512B	Page120	WSector30	RSector 7	
	0x0800 E200 - 0x0800 E3FF	512B	Page121			
	0x0800 E400 - 0x0800 E5FF	512B	Page122			
	0x0800 E600 - 0x0800 E7FF	512B	Page123			
	0x0800 F800 - 0x0800 F9FF	512B	Page124	WSector31		
	0x0800 FA00 - 0x0800 FBFF	512B	Page125			
	0x0800 FC00 - 0x0800 FDFF	512B	Page126			
	0x0800 FE00 - 0x0800 FFFF	512B	Page127			
信息区	0x1FFF F000 - 0x1FFF F1FF	512B	Page 0	Data Memory		
	0x1FFF F200 - 0x1FFF F3FF	512B	Page 1			
	0x1FFF F400 - 0x1FFF F5FF	512B	Page 2			
	0x1FFF F600 - 0x1FFF F6FF	256B	Page 3	Option Bytes		
	0x1FFF F700 - 0x1FFF F7FF	256B		Calibration Memory		

注 1: WSector 对应写保护分区, 主存储区整体分成 32 个写保护区间, 每个区间 2KB。

注 2: RSector 对应读保护分区, 主存储区整体分成 8 个读保护区间, 每个区块 8KB。

3.2.2 Flash 的读操作

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, 读操作相关寄存器 (FLASH_ACR) :

- 指令预取: 具有指令预取缓存以加速 CPU 的执行速度
- 多周期访问: 读存储器操作的等待周期设定 (0~4, 对应等待 0 到 4 个时钟周期)

指令预取

本系列产品内置指令缓存, CPU 读操作会更新缓存, 反馈当前有效指令数据给 CPU。在使能指令预取后, 会自动读取 Flash 的下一个数据, 并更新指令缓存的内容, 以便加速 CPU 执行速度。

初始状态下, 默认指令预取开启, 上电复位完成后即开始预取指令和数据。

指令预取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。

多周期访问

多周期访问通过 FLASH_ACR.LATENCY 来配置, 通过此方式, 延长取指周期, 确保访问 Flash 的时序正确, 保证读取数据的准确性。上电后默认状态下, FLASH_ACR.LATENCY 为 0, 即读 Flash 无需额外等待周期。

3.2.3 Flash 的烧录和擦除

Flash 可以通过内部控制线路进行直接烧录, 也可以通过应用程序软件控制烧录。

内部控制线路 (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 BOOT Loader 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，在线应用程序编程（IAP）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、USART、I2C 或 SPI 等）来进行 Flash 的更新。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。但 IAP 模式需要占用一定的存储空间，且更新控制程序第一次只能先通过 ICP 方式烧录。用户需要小心控制更新协议和过程，一定要有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录，相关控制寄存器如下：

- 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_WRPFR）
- 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）

Flash 的烧录要求内部 RCH 时钟必须处于开启状态，同时烧录过程中禁止所有对 Flash 的指令和数据访问。

Flash 的解锁

复位后，Flash 会处于只读模式，控制寄存器（FLASH_CR）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。发生写密钥错误时，会产生 Hardfault 中断。写密钥错误包括第一次写密钥错误，以及第一次写正确，但第二次写密钥错误。

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

主存储区的编程以一个字节（8-bits）作为基本单位，通过设置寄存器 FLASH_CR.PG 位为 1，一次性编程一个字节到 Flash；也允许以半字（16-bits）或字（32-bits）为单位的编程。

Flash 存储控制模块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，且 FLASH_SR.PGERR 位被置 1。

如果目标地址处于寄存器 FLASH_WRPFR 规定的写保护区域，则当前编程操作被忽略，且寄存器 FLASH_SR.WRPRTERR 位被置 1。编程结束后，FLASH_SR.EOP 位也会被置 1。

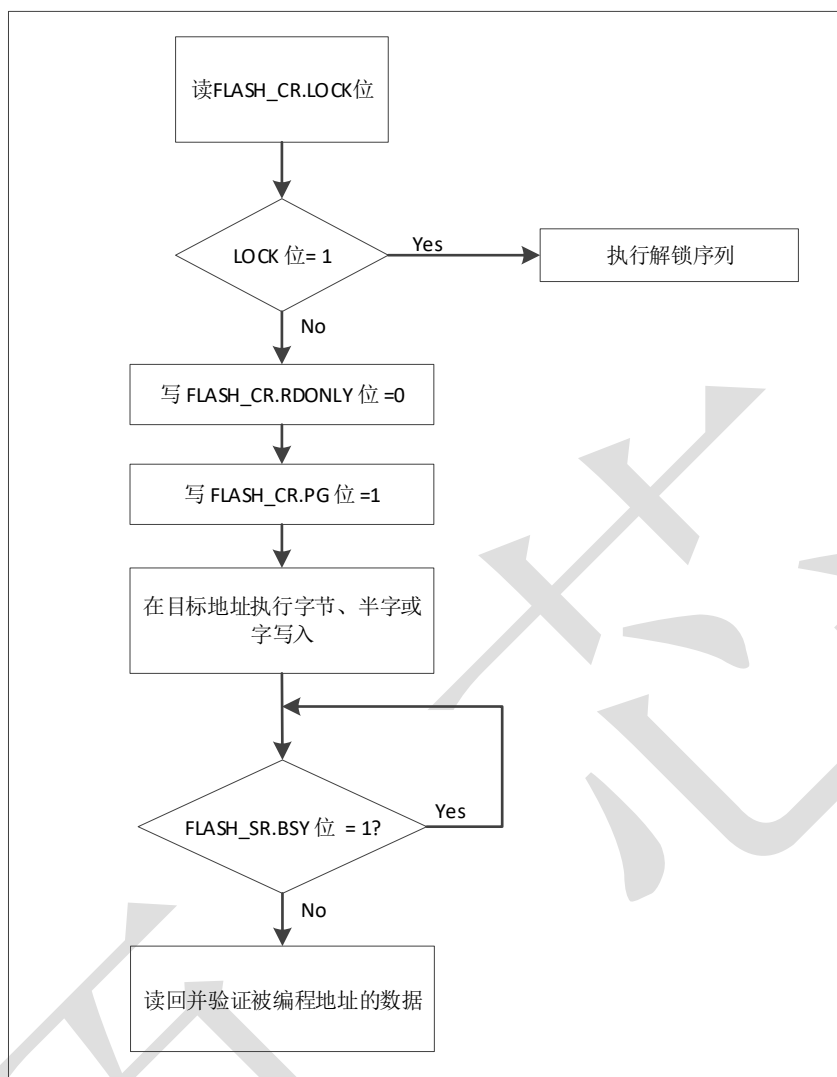


图 3-1 FLASH 编程流程

正常模式下的主存储区烧录流程如下：

1. 检测 FLASH_SR.BSY 位，确认当前没有主存储区的其他操作正在进行中；
2. 清除寄存器 FLASH_CR.RDONLY 位，使 Flash 退出只读状态；
3. 设置寄存器 FLASH_CR.PG 位；
4. 字节、半字或全字方式写入目标地址；
5. 等待 FLASH_SR.BSY 位被清零；
6. 检测 FLASH_SR.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.BSY 位设置起来后访问

Flash 存储器的擦除

存储器可以按半页擦除、页擦除或者一次性全部擦除（块擦除），Data Memory、Option Bytes 区域的擦除方式固定为半页擦除。

页/半页擦除

页/半页擦除的操作过程如下：

1. 通过检测 FLASH_SR.BSY 位，确认没有其他操作正在进行；
2. 清除寄存器 FLASH_CR.RDONLY 位，使 FLASH 退出只读状态；

3. 设置 FLASH_CR.PER 位（页擦除）/ HALFER（半页擦除）；
4. 设置寄存器 FLASH_AR，以确定需要擦除哪一页/半页；
5. 将 FLASH_CR.STRT 位设置为 1；
6. 等待 FLASH_SR.BSY 位被清零；
7. 检测 FLASH_SR.EOP 标志（当擦除操作成功后，EOP 标志位置起）；
8. 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 FLASH_SR.BSY 是否为 0。

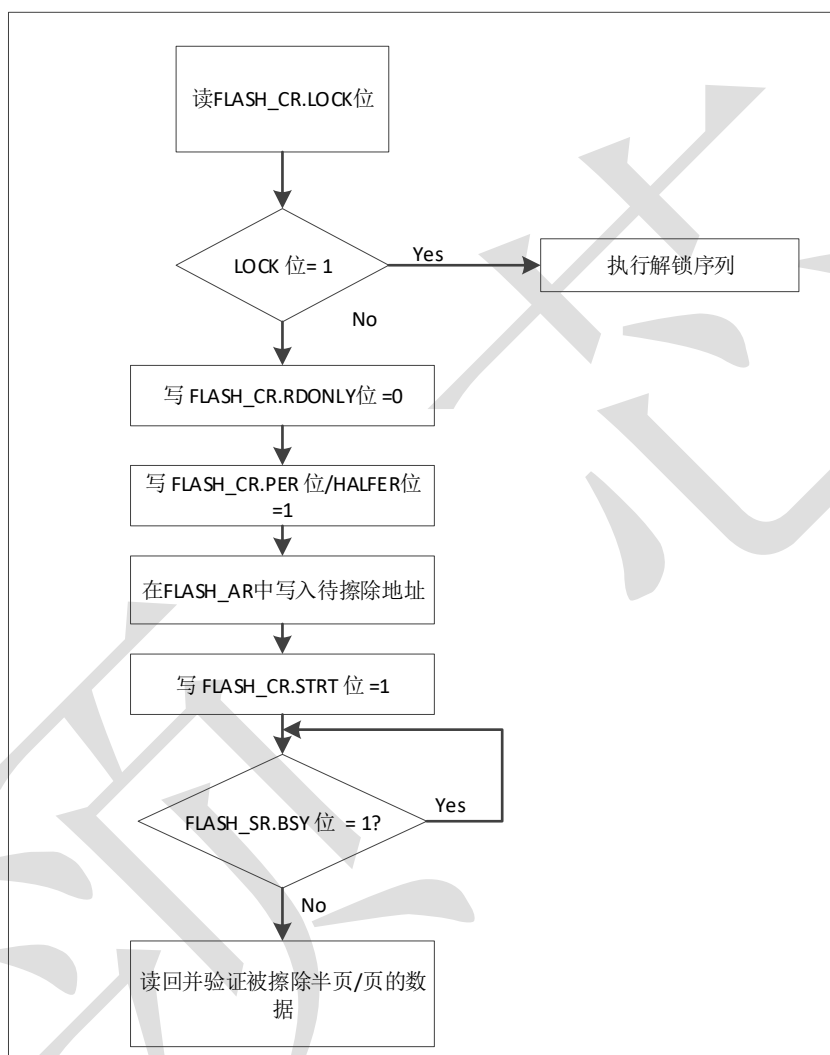


图 3-2 FLASH 的页/半页擦除流程

块擦除

块擦除可以用于擦除主存储区的所有页，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.BSY 位，确认没有其他操作正在进行；
2. 清除寄存器 FLASH_CR.RDONLY 位，使 Flash 退出只读状态；
3. 设置 FLASH_CR.MER 位；
4. 将 FLASH_CR.STRT 位设置为 1；
5. 等待 FLASH_SR.BSY 位被清零；
6. 检测 FLASH_SR.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 FLASH_SR.BSY 是否为 0。

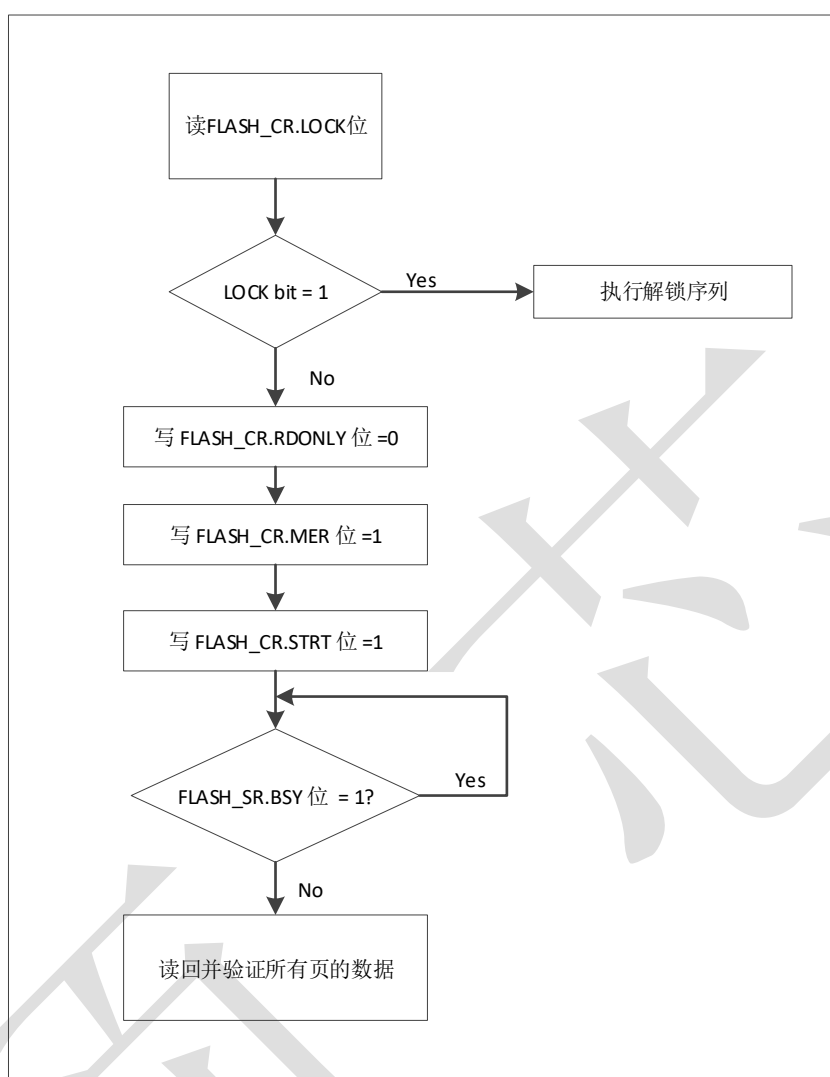


图 3-3 FLASH 的块擦除操作流程

选项字节（Option Byte）的编程

该系列产品的选项字节为一个地址段的区间，对应 0x1FFF_F400~0x1FFF_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，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，正确解锁将 FLASH_CR.OPTWRE 位置起。此时，用户才可以通过设置 FLASH_CR.OTPG 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，且 FLASH_SR.WRPRTERR 位被置 1。编程结束后，FLASH_SR.EOP 标志位被置 1。

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），原写入的高 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.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.OPTWRE 位（通过写密钥到 FLASH_OPTKEYR 寄存器）；
3. 清除寄存器 FLASH_CR.RDONLY 位，使 Flash 退出只读状态；

4. 设置 FLASH_CR.OPTPG 位;
5. 将数据写入目标地址 (全字方式, 实际有效 bit[7:0], bit[23:16]);
6. 等待 FLASH_SR.BSY 位被清零;
7. 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, 会发起一个块擦除操作, 将主存储区的数据全部擦除。除此之外对于选项字节的其它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.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.OPTWRE 位 (通过写密钥到 FLASH_OPTKEYR 寄存器);
3. 设置 FLASH_CR.OPTER 位;
4. 设置 FLASH_CR.STRT 位;
5. 等待 FLASH_SR.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 Flash 中的用户数据可通过硬件保护来防止非法读取, LCM32F06X 支持 8 个区间的读保护设置。另外, Flash 可进行写保护, 主存储区以 2KB 为一个写保护区间, 一共划分为 32 个区间, 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, 并确定相应区块的读保护状态。

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 Hardfault 错误。

读保护有 3 个保护等级: 没有保护 (等级 0), 正常读保护 (等级 1) 和最强保护 (无法调试) (等级 2), 具体参考表 3-2。该系列产品具备分区保护的功能 (等级 2 涵盖所有 Flash 区域), 不同分区之间, 低保护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

校准字节区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。

等级 0: 无保护

对应主存储区可以被读取、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，不需要判断其补码是否正确，相关区域即被设定为读保护。

- 用户程序运行：低保护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；
- 引导程序和 SRAM：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存储区，访问读保护区的操作会触发 Hardfault 错误。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_SR.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，所有 Flash 区域会被擦除。

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 SRAM 加载初始引导程序。

用户程序运行时，可正常读取保护等级相同的数据，但不能操作分区保护等级更高的区域。用户程序此时只能对选项字节进行读取，不能进行擦除和编程操作。

RDPn 字节禁止重新编程，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.WRPRTERR 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			调试模式/RAM 区程序		
		读	写	擦	读	写	擦
主存储区	1	是	是	是	否	否	否 ⁴
	2	是	是	是	N/A ¹	N/A	N/A
校准字节 ²	1	是	否	否	是	否	否
	2	是	否	否	N/A	N/A	N/A
选项字节	1	是	是 ³	是	是 ⁴	是	是
	2	是	否	否	N/A	N/A	N/A

注1：读保护等级 2，调试端口和 RAM 区程序执行被禁止。

注2：校准字节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，主存储区被强制擦除。

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）。

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 3-3 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，FLASH_SR.WRPRTERR 位被置 1。

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.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存储区；

- 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 FLASH_CR.OPTER 位擦除选项字节区；
 - 通过上电复位或者外部复位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密钥（与解锁相同）到 OPTKEYR 寄存器，正确写入后，FLASH_CR.OPTWRE 位被置 1。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	中断类型
操作结束	EOP	EOPIE	Flash 中断
写保护错误	WRPRTERR	ERRIE	Flash 中断
编程错误	PGERR	ERRIE	Flash 中断

3.5 Flash 寄存器描述

Flash 相关的寄存器必须以字（32-bit）访问（不允许半字和字节访问），并通过 APB0 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	0x00XX_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_RDPR	读保护寄存器	R	0xFFFF_XXXX	0x4001D020
FLASH_WRPR	写保护寄存器	R	0xFFFF_XXXX	0x4001D024

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 ：当前预取状态
		0：当前预取没有使能（默认）
		1：当前预取已经使能
Bit	4	PRFTBE ：预取使能
		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 Flash 的等待周期
		000：0 个等待周期，当 CPU 时钟频率≤24MHz（默认）
		001：1 个等待周期，当 24 MHz < CPU 时钟频率 ≤ 48 MHz
		010：2 个等待周期，当 48 MHz < CPU 时钟频率 ≤ 72 MHz
		011：3 个等待周期，当 72 MHz < CPU 时钟频率 ≤ 96 MHz
		100：4 个等待周期，当 96 MHz < CPU 时钟频率
		其他：保留

3.5.2 Flash 钥匙寄存器（FLASH_KEYR）

地址偏移：0x04

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
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] ：Flash 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 FLASH_CR.OPTWRE 位的写入密钥

3.5.4 Flash 状态寄存器（FLASH_SR）

地址偏移：0x0C

复位值：0x00XX_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.											OBERR	OB_EMPTY_FLAG	OB_BOOT0_PIN	OB_BOOT_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	WRPRERR	Res.	PGERR	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
	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，表示系统复位后主存储空间的状态
	0: 主存储空间已被编程
	1: 主存储空间未被编程
Bit 18	OB_BOOT0_PIN: 复位后所采样到的 BOOT0 管脚的值
	0: BOOT0 管脚为低电平
	1: BOOT0 管脚为高电平
Bits 17:16	OB_BOOT_MODE: 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
	01: 保留
	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，软件写 1 清零
	0: 表示 Flash 空闲或编程/擦除操作未结束（默认）
	1: 由硬件置位，表示完成一次有效的编程/擦除操作
Bit 4	WRPRERR: 写保护错误标志，软件写 1 清零
	0: 未发生写保护错误（默认）
	1: 对一个写保护地址进行编程/擦除操作时，由硬件置位
Bit 3	保留，必须保持复位值
Bit 2	PGERR: 编程错误标志，软件写 1 清零

	0: 未发生编程错误（默认）
	1: 硬件置位，当编程写入的目标地址的初值不为 0xFFFF、同一个地址多次写入或者 FLASH_CR.STRT 位置起时同时存在多个擦除选择项时置 1
	注：开始一次编程操作前，FLASH_CR 寄存器中的 STRT 位须先被清零
Bit 1	保留，必须保持复位值
Bit 0	BSY: 忙标志
	0: 表示 Flash 空闲（默认）
	1: 硬件置位，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 相关错误发生时硬件自动清零

3.5.5 Flash 控制寄存器（FLASH_CR）

地址偏移：0x10

复位值：0x1X01_0080

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

Bits 31:29	保留，必须保持复位值
Bit 28	RDONLY: 只读模式使能
	0: 只读模式关闭
	1: 只读模式使能（默认）
Bits 27:26	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
	01: 保留
	11: 内部 SRAM 存储映射到 0x0000_0000
Bits 23:13	保留，必须保持复位值
Bit 12	EOPIE: 操作结束中断使能
	当 FLASH_SR.EOP 置位，控制相关中断是否产生
	0: EOP 中断禁止（默认）
	1: EOP 中断使能
Bit 11	保留，必须保持复位值
Bit 10	ERRIE: 错误中断使能
	当 FLASH_SR.PGERR/WRPRERR 置位时，控制相关中断是否产生
	0: Flash 错误中断禁止（默认）

	1: Flash 错误中断使能
Bit 9	OPTWRE: 选项字节编程标志
	只允许软件写 0。当正确写秘钥 FLASH_OPTKEYR 寄存器时置 1，为 1 时表示选项字节可以被编程，可以被软件清零；当解锁操作失败，此标志位保持为 0，直至下次复位
	0: 禁止编程选项字节（默认）
	1: 允许编程选项字节
Bit 8	保留，必须保持复位值
Bit 7	LOCK: FLASH_CR 寄存器锁定标志
	只允许软件写 1。当正确写秘钥 FLASH_KEYR 寄存器时置 0，解锁后可以被软件置 1，为 1 表示 FLASH_CR 寄存器被锁定；当解锁操作失败，此标志位保持为 1，直至下次复位
	0: 允许配置 FLASH_CR 寄存器
	1: 禁止 FLASH_CR 寄存器（默认）
Bit 6	STRT: 开始 Flash 操作
	只允许软件写 1，写 1 表示启动一次擦除 ERASE 操作，STRT 位在 FLASH_SR.BSY 标志清零时自动清零。OPTER/HALFER/MER/PER 在 STRT 置 1 时只能置起其中一个，否则将触发错误，并将 FLASH_SR.PGERR 置 1。
Bit 5	OPTER: 选项字节和数据存储区擦除选择
	0: 不选择擦除选项字节和数据存储区（默认）
	1: 选择擦除选项字节或者数据存储区，单次擦除大小为半页
Bit 4	OPTPG: 选项字节和数据存储区编程选择
	0: 不选择编程选项字节和数据存储区（默认）
	1: 选择编程选项字节或者数据存储区
Bit 3	HALFER: 主存储区半页擦除
	0: 不选择按半页擦除主存储区（默认）
	1: 选择按半页擦除主存储区
Bit 2	MER: 主存储区块擦除
	0: 不选择所有主存储区擦除（默认）
	1: 选择所有主存储区擦除
Bit 1	PER: 主存储区页擦除
	0: 不选择按页擦除主存储区（默认）
	1: 选择按页擦除主存储区
Bit 0	PG: 编程操作
	0: 不选择编程 Flash 主存储区（默认）
	1: 选择编程 Flash 主存储区

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/半页擦除操作时选择擦除页/半页
	当 FALSH_SR.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 引脚赋值（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 读保护寄存器（FLASH_RDPR）

地址偏移：0x20

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.															

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

3.5.9 写保护寄存器（FLASH_WRPRT）

地址偏移: 0x24

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
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

3.6 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 3-7。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 Flash 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								Res.				USER			
												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 7:4	USER: 用户配置选项字节 (复位加载到 FLASH_OBR[7:0])
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
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							

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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 2KB (WSectors)。

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								RDP0							
rw								rw							

Bits 31:24	nRDP1: Flash 存储读保护选项字节 1 反码
Bits 23:16	RDP1: 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	RDP0: 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
	注: 读保护等级复位加载到 Flash 读保护寄存器 (FLASH_RDPR) 的 RDPRT[7:0]

3.6.4.2 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	RDP3: 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)
	0xFF (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	RDP2: 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)
	0xFF (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]

3.6.4.3 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)
	0xFF (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)
	0xFF (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
	<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 8/9

FLASH 存储地址: 0x1FFF_F620

默认配置值: 0x55AA_55AA

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 0xFFFF_CCCC 之外): level 0 (出厂默认配置)
	0xFFFF_CCCC: 等级 2
	<i>注: 读保护等级 2 复位加载到 Flash 读保护寄存器 (FLASH_RDPR) 的 RDPRT[8]</i>

3.6.4.5 读保护选项字节 10/11

FLASH 存储地址: 0x1FFF_F624

默认配置值: 0x55AA_55AA

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
	0XXXXX_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

Flash 校准字节区地址范围：0x1FFFF700~0x1FFFF7FF 共计 256Bytes。此区域校准字节用于确定当前芯片的外设校准信息，内容权限为只读，原厂设定，不可更改。

表 3-8 Flash 校准字节

Flash 地址	校准字节内容
0x1FFFF700~0x1FFFF708	保留
0x1FFFF70C	ADC 校准值@3.2V 参考
0x1FFFF710	OPA0 过零校准 TRIM 值
0x1FFFF712	OPA1 过零校准 TRIM 值
0x1FFFF714	ACMP0 过零校准 TRIM 值
0x1FFFF716	ACMP1 过零校准 TRIM 值
0x1FFFF718	OPA2 过零校准 TRIM 值
0x1FFFF71C	保留
0x1FFFF720	OPA0 常温共模零点偏置采样值
0x1FFFF724	OPA0 常温 2x 计算值
0x1FFFF726	OPA0 常温 4x 计算值
0x1FFFF728	OPA0 常温 6x 计算值
0x1FFFF72A	OPA0 常温 8x 计算值
0x1FFFF72C	OPA0 常温 10x 计算值
0x1FFFF72E	OPA0 常温 16x 计算值
0x1FFFF730	OPA1 常温共模零点偏置采样值
0x1FFFF734	OPA1 常温 2x 计算值
0x1FFFF736	OPA1 常温 4x 计算值

0x1FFFF738	OPA1 常温 6x 计算值
0x1FFFF73A	OPA1 常温 8x 计算值
0x1FFFF73C	OPA1 常温 10x 计算值
0x1FFFF73E	OPA1 常温 16x 计算值
0x1FFFF740	OPA2 常温共模零点偏置采样值
0x1FFFF744	OPA2 常温 2x 计算值
0x1FFFF746	OPA2 常温 4x 计算值
0x1FFFF748	OPA2 常温 6x 计算值
0x1FFFF74A	OPA2 常温 8x 计算值
0x1FFFF74C	OPA2 常温 10x 计算值
0x1FFFF74E	OPA2 常温 16x 计算值
0x1FFFF750	OPA0 高温共模零点偏置采样值
0x1FFFF754	OPA0 高温 2x 计算值
0x1FFFF756	OPA0 高温 4x 计算值
0x1FFFF758	OPA0 高温 6x 计算值
0x1FFFF75A	OPA0 高温 8x 计算值
0x1FFFF75C	OPA0 高温 10x 计算值
0x1FFFF75E	OPA0 高温 16x 计算值
0x1FFFF760	高温 OPA1 共模零点偏置采样值
0x1FFFF764	OPA1 高温 2x 计算值
0x1FFFF766	OPA1 高温 4x 计算值
0x1FFFF768	OPA1 高温 6x 计算值
0x1FFFF76A	OPA1 高温 8x 计算值
0x1FFFF76C	OPA1 高温 10x 计算值
0x1FFFF76E	OPA1 高温 16x 计算值
0x1FFFF770	高温 OPA2 共模零点偏置采样值
0x1FFFF774	OPA2 高温 2x 计算值
0x1FFFF776	OPA2 高温 4x 计算值
0x1FFFF778	OPA2 高温 6x 计算值
0x1FFFF77A	OPA2 高温 8x 计算值
0x1FFFF77C	OPA2 高温 10x 计算值
0x1FFFF77E	OPA2 高温 16x 计算值
0x1FFFF780	常温 VTS 温度标定值
0x1FFFF784	高温 VTS 温度标定值
0x1FFFF788~0x1FFFF79C	保留
0x1FFFF7A0~0x1FFFF7AC	ADC 校准值@4.9V 参考
0x1FFFF7B0~0x1FFFF7BC	128 位唯一 UID
0x1FFFF7C0~0x1FFFF7DC	保留

0x1FFFF7E0	常温 VTS 采样 AD 值
0x1FFFF7E4	高温 VTS 采样 AD 值
0x1FFFF7E8~0x1FFFF7FC	保留

4. 电源管理（PWR）

4.1 供电方案

- VSS, VDDH = 2.0~5.5V: VDDH 引脚为 I/O 引脚和 RCH、PLL 等高频模块供电。
- VSSA, VDDA = 2.0~5.5V: 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 1.2V。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。

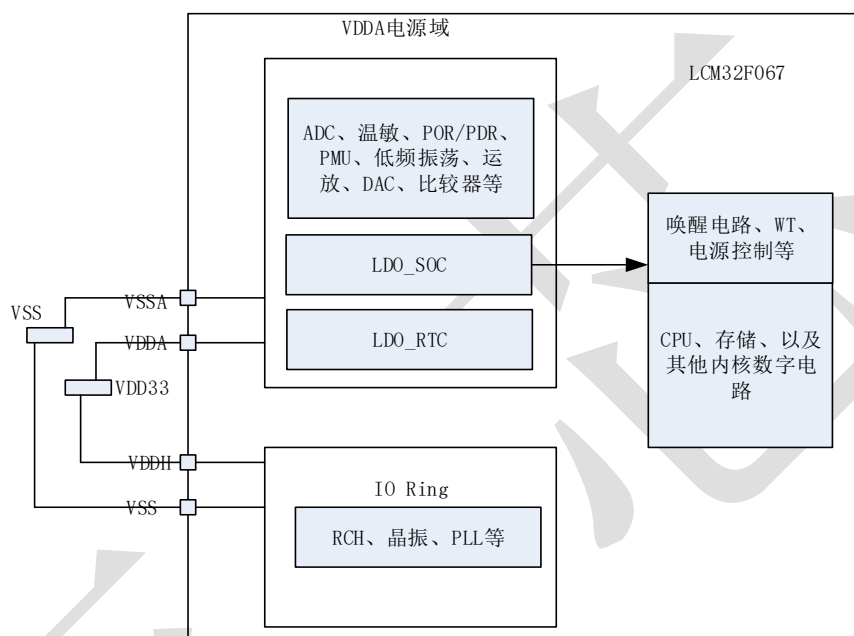


图 4-1 电源供电结构

LCM32F06X 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）

LCM32F06X 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 1.2V
- 低功耗模式（LPR），此时内核 LDO 自身功耗最低可降到 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。

4.2 电源检测

LCM32F06X 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。

LCM32F06X 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 和 4.0V。可根据供电配置合适的 LVR 档位。

LCM32F06X 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 和 4.5V。

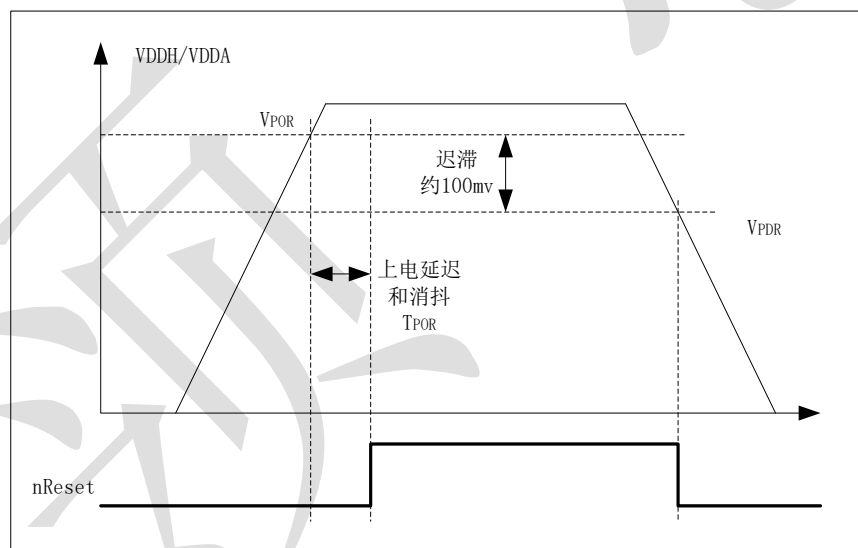


图 4-2 上电和下电复位图

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。LCM32F06X 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 和 RCH 被关闭，内核电源域里的各个模拟模块根据 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、USART、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 和 RCH 被关闭，内核电源域里的各个模拟模块也全部被关掉。

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、USART 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 LCM32F06X 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、USART	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 LPDS 寄存器调整驱动能力
超级停机模式（ULP STOP）	ULPDS=1, SLEEPDEEP=1, WFI/WFE	EXTI、WT、USART	所有时钟关闭	与内核相关的模拟模块必须关闭	切换到低功耗模式

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）： ● 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： ● 中断：参考中断向量表 当用 WFE 作为进入指令： ● 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）： ● 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 和内部 RCH 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，LCM32F06X 的 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	WFI（等待中断）或者 WFE（等待事件）： ● 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 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。 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。
退出	当用 WFI 作为进入指令： ● 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： ● 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，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

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，LDO 会切换到低功耗模式供电，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）： ● 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 ● LDO 自动切换到低功耗模式 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： ● 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： ● 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，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 从低功耗模式切换时间

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 APB0 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 0x87214365 到密钥寄存器，且在随后的 64 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.											
	rw														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		AOUT_EN	LVDS[2:0]			LVDES	LV DEN	AOUT_SEL[2:0]			Res.	LVRS[2:0]			LVREN
		rw	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 2uA（默认）
		1：偏置电流 4uA
Bits	27:14	保留，必须保持复位值
Bit	13	AOUT_EN ：PMU_AOUT 输出使能
		0：PMU_AOUT 输出高阻态（默认）
		1：PMU_AOUT 输出 AOUT_SEL 选择的电压，可从 PA14/PA15 输出
		注： 该输出只支持 VDDA-1.5V 以下的电压输出。
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：选择检测外部电压 ELVI
Bit	8	LV DEN ：LVD 使能控制
		0：LVD 禁止（默认）
		1：LVD 使能
Bits	7:5	AOUT_SEL[2:0] ：PMU_AOUT 输出选择
		000：高精度基准电压
		001：Flash 参考电压
		010：低功耗基准电压
		011：ACMP 的内部参考电压 CVREF

	100: HALL_MID 输出
	101: VDDA/2 电压输出
	110: 保留
	111: 保留（默认）
Bit 4	保留，必须保持复位值
Bits 3:1	LVRS[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 64 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

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: 低功耗模式下 Flash 模块的深度待机控制
	由软件清零和置位，与控制位 ULPDS/LPDS 协同工作
	0: Flash 模块在停机模式和超低功耗停机模式下不进入深度待机模式（默认）
	1: Flash 模块在停机模式和超低功耗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.									STOP	Res.		DRV[1:0]		PD	EXTVDD
									rw			rw		rw	rw

Bits	31:4	保留，必须保持复位值
Bit	6	STOP: 内核 LDO 使能位，与 PD 位配合使用 仅当系统时钟选为 RCL 且 RCH 时钟使能关闭时可配置，否则该位由硬件配置
Bits	5:4	保留，必须保持复位值
Bits	3:2	DRV[1:0]: LDO 驱动模式配置 00: 10uA 自耗电流低功耗模式 01: 50uA 自耗电流运行模式（默认） 10: 150uA 自耗电流运行模式 11: 200uA 自耗电流运行模式
Bit	1	PD: 内核 LDO 使能位，与 STOP 位配合使用；在 STOP=1 的前提下 0: 内核 LDO 使能（默认） 1: 内核 LDO 关闭，芯片由低功耗 LDO 供电
Bit	0	EXTVDD: 内核电源选择 0: 内核电源由 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

LCM32F06X 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，同时电源复位也会产生系统复位。

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 和记录奇偶校验相关内容的寄存器，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. 独立看门狗计数终止（IWDG 复位）
3. 软件复位（SW 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 AIRC 寄存器的 SYSRESETREQ 位产生。

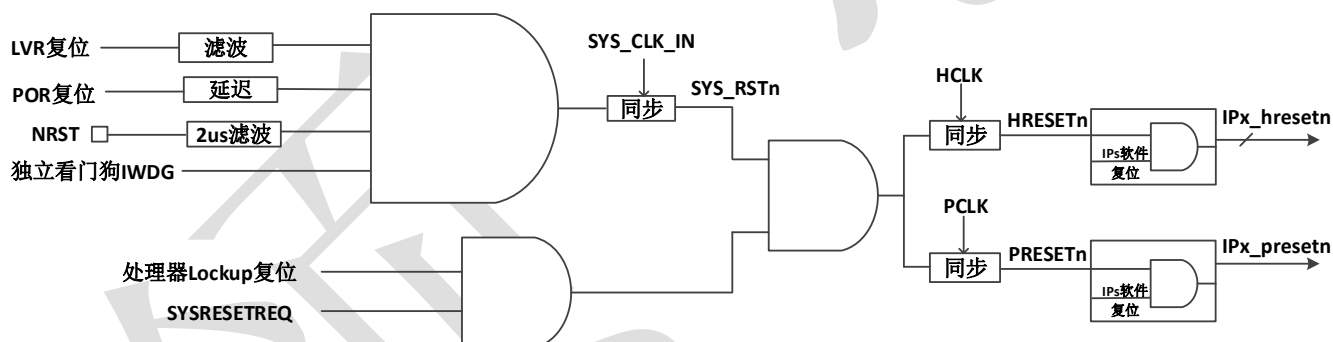


图 5-1 复位框图

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 PLL 时钟。RCH、PLL 和几个时钟分频器一起用来配置 AHB 和 APB 总线的频率。AHB 总线支持的最高频率为 108MHz，APB 总线支持的最高频率为 48MHz。LCM32F06X 的时钟框图如图 5-2 所示。

- Flash 编程接口时钟始终采用内部 RCH 的二分频时钟
- ADC 工作时钟可以通过配置选择相应 APB 总线时钟 PCLK0、PLL 时钟以及它们的分频
- Timer 工作时钟源可通过软件配置，可选择相应 APB 总线时钟 PCLK0 或系统时钟
- USART、I2C 和 SPI 的工作时钟源可通过软件配置，可选择内部 RCH、外部低频时钟 OSCL、相应 APB 总线时钟 PCLK0 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 RCH 的固定分频时钟

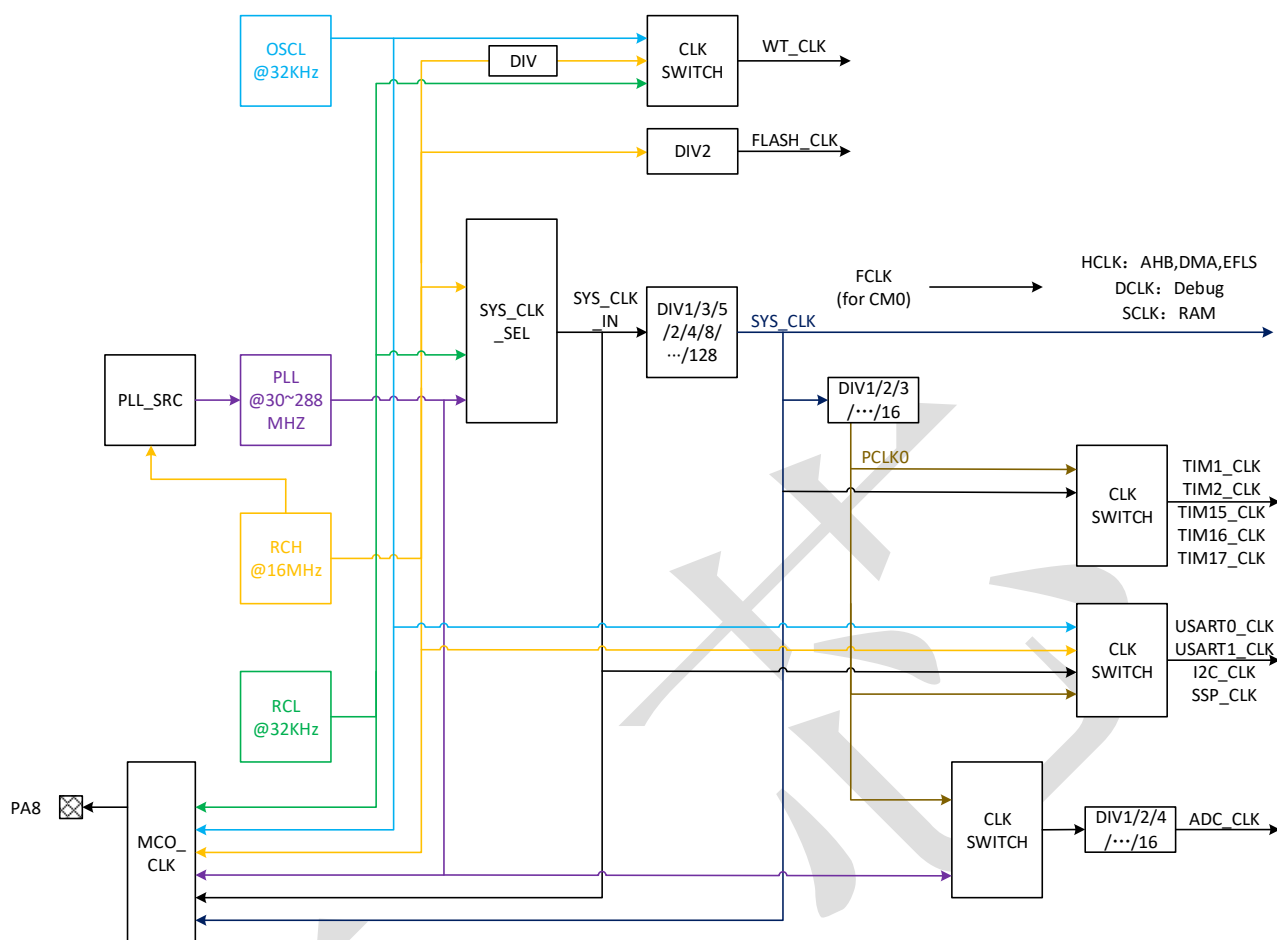


图 5-2 LCM32F06X 时钟框图

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 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）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 PLL 时钟为例，给出配置示例：

1. 配置 PLL 寄存器中的 PLL 时钟特性，PLL 输入选择为 RCH
2. 配置 CLKCFG 寄存器中的 PLL_EN=1，使能 PLL
3. 等待 STS 寄存器中的 PLL_LOCK=1，表示 PLL 开启且已经稳定
4. 配置 CLKCFG 寄存器中的 SYS_SRC_SEL=2'b10，切换系统时钟源为 PLL
5. 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 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 LCM32F06X 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（APB0 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 64 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 APB0 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_1040	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_015E	0x4001_700C
ChipCtrl_RCHCFG	RCH 配置寄存器	R/W	0x0000_0XXX	0x4001_7010
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_F004	0x4001_7028
ChipCtrl_FLASH_P ARITY_ADDR	Flash 奇偶校验地址寄存器	R/W	0x0000_0000	0x4001_7040
ChipCtrl_FLASH_P ARITY_DATA	Flash 奇偶校验数据寄存器	R/W	0x0000_0000	0x4001_7044
ChipCtrl_FLASH_P ARITY_INJ	Flash 奇偶校验注入寄存器	R/W	0x0000_0000	0x4001_7048
ChipCtrl_SRAM_PA RITY_ADDR	SRAM 奇偶校验地址寄存器	R/W	0x0000_0000	0x4001_704C
ChipCtrl_SRAM_PA RITY_DATA	SRAM 奇偶校验数据寄存器	R/W	0x0000_0000	0x4001_7050
ChipCtrl_SRAM_PA RITY_INJ	SRAM 奇偶校验注入寄存器	R/W	0x0000_0000	0x4001_7054

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_1040

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.	Res.	PLL _EN
										rw		rw			rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
APB0_DIV[3:0]				Res.				SYS_CLK_SEL[7:0]							
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: 保留
	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)
Bits 18:17	保留，必须保持复位值
Bit 16	PLL_EN : PLL 使能
	0: PLL 关闭 (默认)
	1: PLL 使能
Bits 15:12	APB0_DIV[3:0] : APB0 总线的 PCLK0 时钟分频选择
	$PCLK0 = SYS_CLK / (APB0_DIV[3:0] + 1)$
	0000: $PCLK0 = SYS_CLK$
	0001: $PCLK0 = SYS_CLK / 2$ (默认)
	...
	1110: $PCLK0 = SYS_CLK / 15$
	1111: $PCLK0 = SYS_CLK / 16$
Bits 11:8	保留，必须保持复位值
Bits 7:0	SYS_CLK_SEL[7:0] : 系统时钟频率选择
	00xx_xxxx: $SYS_CLK = SYS_CLK_IN$
	01xx_xxxx: $SYS_CLK = SYS_CLK_IN / (2 * (SYS_CLK_SEL[5:0] + 1))$ (默认)
	10xx_xxxx: $SYS_CLK = SYS_CLK_IN / 3$
	11xx_xxxx: $SYS_CLK = SYS_CLK_IN / 5$

5.3.1.2 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.											
	rw														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		AOUT_EN	LVDS[2:0]			LVDES	LVDEN	AOUT_SEL[2:0]			Res.	LVRS[2:0]			LVREN
		rw	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 2uA (默认)
		1: 偏置电流 4uA
Bits	27:14	保留，必须保持复位值
Bit	13	AOUT_EN : PMU_AOUT 输出使能
		0: PMU_AOUT 输出高阻态 (默认)
		1: PMU_AOUT 输出 AOUT_SEL 选择的电压，可从 PA14/PA15 输出
		注 : 该输出只支持 VDDA-1.5V 以下的电压输出。
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: 选择检测外部电压 ELVI
Bit	8	LVDEN : LVD 使能控制
		0: LVD 禁止 (默认)
		1: LVD 使能
Bits	7:5	AOUT_SEL[2:0] : PMU_AOUT 输出选择
		000: 高精度基准电压
		001: Flash 参考电压
		010: 低功耗基准电压
		011: ACMP 的内部参考电压 CVREF
		100: HALL_MID 输出
		101: VDDA/2 电压输出
		110: 保留
		111: 保留 (默认)
Bit	4	保留，必须保持复位值
Bits	3:1	LVRS[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
Res.	LOCKUPEN	LVD_LOCK	Res.		PLL_IP	Res.	OSCL_IP	Res.							
	rw	rw			rw		rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SRAM_IE	DF_IE	IF_IE	Res.	HF_ROM_EN	SRAM_P_EN	FLASH_P_EN	Res.	CLKEN_ERR_IE	CLKMUX_ERR_IE	PLL_IE	Res.	OSCL_IE	RCH_IE	RCL_IE
	rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw

Bit 31	保留，必须保持复位值
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 TIM1/2/15/16/17 的刹车输入的连接关系。 0: 处理器 LOCKUP 输出与 TIM1/2/15/16/17 的刹车输入无连接（默认） 1: 处理器 LOCKUP 输出与 TIM1/2/15/16/17 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 TIM1/2/15/16/17 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 的刹车输入无连接（默认） 1: LVD 检测结果与 TIM1/15/16 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 由不稳定到稳定时设置 PLL 状态位（默认） 1: 当 PLL 由稳定到不稳定时设置 PLL 状态位
Bit 25	保留，必须保持复位值
Bit 24	OSCL_IP: 设置 OSCL 状态寄存器的极性 0: 当 OSCL 由不稳定到稳定时设置 OSCL 状态位（默认） 1: 当 OSCL 由稳定到不稳定时设置 OSCL 状态位
Bits 23:15	保留，必须保持复位值
Bit 14	SRAM_IE: 读取 SRAM 时出现奇偶校验错误的中断配置

	0: 进入 SRAM 中断（默认）
	1: 进入 HardFault 异常
Bit 13	DF_IE: 通过 Flash 取数据时出现奇偶校验错误的中断配置
	0: 进入 Flash 中断（默认）
	1: 进入 HardFault 异常
Bit 12:10	保留，必须保持复位值
Bit 9	SRAM_P_EN: SRAM 奇偶校验使能
	0: SRAM 不进行奇偶校验检测（默认）
	1: SRAM 进行奇偶校验检测
Bit 8	FLASH_P_EN: Flash 奇偶校验使能
	0: Flash 不进行奇偶校验检测（默认）
	1: Flash 进行奇偶校验检测
Bit 7	保留，必须保持复位值
Bit 6	CLKEN_ERR_IE: 时钟使能错误使能 RCC 中断
	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使能 RCC 中断
	0: 中断不使能（默认）
	1: 中断使能
Bit 4	PLL_IE: PLL 状态变化使能 RCC 中断
	0: 中断不使能（默认）
	1: 中断使能
Bit 3	保留，必须保持复位值
Bit 2	OSCL_IE: OSCL 状态变化使能 RCC 中断
	0: 中断不使能（默认）
	1: 中断使能
Bit 1	RCH_IE: RCH 状态变化使能 RCC 中断
	0: 中断不使能（默认）
	1: 中断使能
Bit 0	RCL_IE: RCL 状态变化使能 RCC 中断
	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_015E

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	PLL_SEL	PLL_MODE	PLL_DM[1:0]		Res.	PLL_DN[4:0]				PLL_LS	Res.	
			rw	rw	rw	rw			rw				rw		

Bits 31:13	保留，必须保持复位值
Bit 12	PLL_OD : PLL 输出时钟二分频使能
	0: PLL 输出时钟（默认）
	1: PLL 输出时钟二分频
Bit 11	PLL_SEL : PLL 输出时钟选择
	0: 输出时钟频率 $F_{PLL_CLK} = F_{PLL_VCO} / ((OD + 1) * 4)$ （默认）
	1: 输出时钟频率 $F_{PLL_CLK} = F_{PLL_VCO} / ((OD + 1) * 2)$
	此处 $F_{PLL_VCO} = F_{INPUT_CLK} * (DN + 1) / (DM + 1)$
注 : PLL_VCO 频率最高可配置为 256MHz。	
Bit 10	PLL_MODE : PLL_VCO 增益控制
	0: PLL_VCO 正常产生（默认）
	1: PLL_VCO 增益变大，减少 PLL 锁存需要的稳定时间
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	保留，必须保持复位值
注 : PLL 输出时钟作为系统时钟源时，系统时钟不能超过 108MHz，运行最高频率 108MHz 时，需要打开相应的取指延迟，详见 FLASH_ACR.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.										RCH_FJ_RANGE[1:0]		RCH_FJ_INV[2:0]		RCH_FJ_EN	
										rw		rw		rw	

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

Bits	31:22	保留，必须保持复位值
Bits	21:20	RCH_FJ_RANGE[1:0]: RCH 抖频变化范围
		00: 抖频变化范围最小（默认）
		...
		11: 抖频变化范围最大
Bits	19:17	RCH_FJ_INV[2:0]: RCH 抖频中每次频率变化的时间间隔
		000: 64 个 RCH 周期变化一次（默认）
		001: 128 个 RCH 周期变化一次
		...
		111: 8192 个 RCH 周期变化一次
Bit	16	RCH_FJ_EN: RCH 抖频使能一旦抖频使能开启，则会在当前已校准好的 RCH 上进行抖频，当抖频使能关闭时，则自动切换回到之前已校准的 RCH 频率
		0: RCH 抖频禁止（默认）
		1: RCH 抖频使能
Bits	15:0	保留，必须保持复位值

5.3.1.6 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	Res.	OSCL _STB	RCH_ STB	RCL_ STB
							r	r	r		r		r	r	r

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

Bits	31:25	保留，必须保持复位值
Bit	24	LVDFLG: 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: 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: 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	保留，必须保持复位值
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	15:7	保留，必须保持复位值
Bit	6	CLKEN_ERR: 时钟使能错误状态，配合 CLKEN_ERR_IE 可产生 RCC 中断，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，配合 CLKMUX_ERR_IE 可产生 RCC 中断，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，配合 PLL_IE 可产生 RCC 中断，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	保留，必须保持复位值
Bit	2	OSCL_STS: OSCL 稳定信号的变化状态，配合 OSCL_IE 可产生 RCC 中断，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，配合 RCH_IE 可产生 RCC 中断，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，配合 RCL_IE 可产生 RCC 中断，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）

5.3.1.7 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	Res.	IWDG_ RST_FLAG	RSTN _FLAG	Res.	LVR_ FLAG	POR_ FLAG
			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的使能
		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	SYSRESET_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

5.3.1.8 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 64 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

5.3.1.9 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.						
								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 6:0	保留, 必须保持复位值
注: 该寄存器只在上电复位时复位。	

5.3.1.10 BD 控制寄存器 (ChipCtrl_BDCR)

地址偏移: 0x28

复位值: 0x0000_F004

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

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 PB2 端口配置
	0: PB2 作为通用 GPIO
	1: PB2 作为外部复位引脚（默认）
注 : 上电复位后, PB2 引脚缺省配置为外部复位引脚 nRST; 如果通过选项字节配置 PB2 为 SWCLK, 则在芯片开始工作时 PB2 会作为 SWCLK, 此时外部复位配置自动关闭, 之后由软件配置。	
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 WT 模块和 ChipCtrl_BDCR 寄存器
Bits 10:9	WT_SEL[1:0] : WT 时钟源选择
	00: 根据 SysCtrl_ClkEnR1.RCH_DIV 使能的 RCH 的分频时钟（默认）
	01: 外部低振 OSCL
	10: 内部低振 RCL
	11: 保留
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 OSCL 未使能时才能使用
	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
注： 该寄存器只在上电复位时复位。	

5.3.1.11 Flash 奇偶校验地址寄存器（ChipCtrl_FLASH_PARITY_ADDR）

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.			FLASH_PARITY_ADDR[16:0]												
			r												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				FLASH_PARITY_BIT[3:0]				Res.						DF_P_FL	IF_P_FL
														AG	G
				r										rw	rw

Bits 31:29	保留，必须保持复位值
Bits 28:12	FLASH_PARITY_ADDR[16:0] : Flash 发生奇偶校验错误时的错误地址的低 17 位，该位域只读，只在上电复位时复位
Bits 11:8	FLASH_PARITY_BIT[3:0] : Flash 发生奇偶校验错误时的校验位，该位域只读，只在上电复位时复位
Bits 7:2	保留，必须保持复位值
Bit 1	DF_P_FLAG : 通过 Flash 取数据时出现奇偶校验错误的中断标志位，写 1 清零，只在上电复位时复位
	0: Flash 取数据时未出现奇偶校验错误（默认）
	1: Flash 取数据时出现奇偶校验错误
Bit 0	IF_P_FLAG : 通过 Flash 取指令时出现奇偶校验错误的中断标志位，写 1 清零，只在上电复位时复位
	0: Flash 取指令时未出现奇偶校验错误（默认）
	1: Flash 取指令时出现奇偶校验错误

5.3.1.12 Flash 奇偶校验数据寄存器（ChipCtrl_FLASH_PARITY_DATA）

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
FLASH_PARITY_DATA[31:0]															
r															

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

Bits 31:0	FLASH_PARITY_DATA[31:0] : Flash 发生奇偶校验错误时的错误数据, 该位域只读, 只在上电复位时复位
-----------	---

5.3.1.13 Flash 奇偶校验注入寄存器 (ChipCtrl_FLASH_PARITY_INJ)

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
								FLASH_P_INJ_EN[3:0]				FLASH_P_INJ_BIT[3:0]			
								rw				rw			

Bits 31:8	保留, 必须保持复位值
Bits 7:4	FLASH_P_INJ_EN[3:0] : Flash 奇偶校验注入使能位, 当配置为 1010 时, 允许下一次编程 flash 时将奇偶校验位注入为 FLASH_P_INJ_BIT, 当该次编程结束时使能位由硬件自动清零; 每次配置后只能对一次 Flash 的编程操作生效
Bits 3:0	FLASH_P_INJ_BIT[3:0] : Flash 奇偶校验注入位, 从低到高对应 4 个字节的奇偶校验位

5.3.1.14 SRAM 奇偶校验地址寄存器 (ChipCtrl_SRAM_PARITY_ADDR)

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.								SRAM_PARITY_CHECK[3:0]				SRAM_PARITY_BIT[3:0]			
								r				r			

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

Bits 31:24	保留, 必须保持复位值
Bits 23:20	SRAM_PARITY_CHECK[3:0] : SRAM 发生奇偶校验错误时的实际字节选择, SRAM 是支持按字节读写的, 因此需要指明具体读取的字节位置 该位域只读, 只在上电复位时复位
Bits 19:16	SRAM_PARITY_BIT[3:0] : SRAM 发生奇偶校验错误时的校验位, 该位域只读, 只在上电复位时复位
Bits 15:2	SRAM_PARITY_ADDR[13:0] : SRAM 发生奇偶校验错误时的错误地址低 14 位, 该

	位域只读，只在上电复位时复位
Bit 1	保留，必须保持复位值
Bit 0	SRAM_P_FLAG : SRAM 出现奇偶校验错误的中断标志位，写 1 清零，只在上电复位时复位

5.3.1.15 SRAM 奇偶校验数据寄存器（ChipCtrl_SRAM_PARITY_DATA）

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
SRAM_PARITY_DATA[31:0]															
r															

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

Bits 31:0	SRAM_PARITY_DATA[31:0] : SRAM 发生奇偶校验错误时的错误数据，该位域只读，只在上电复位时复位
-----------	---

5.3.1.16 SRAM 奇偶校验注入寄存器（ChipCtrl_SRAM_PARITY_INJ）

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
								SRAM_P_INJ_EN[3:0]				SRAM_P_INJ_BIT[3:0]			
								rw				rw			

Bits 31:8	保留，必须保持复位值
Bits 7:4	SRAM_P_INJ_EN[3:0] : SRAM 奇偶校验注入使能位，当配置为 1010 时，允许下一次写 SRAM 时将奇偶校验位注入为 SRAM_P_INJ_BIT，当该次编程结束时使能位由硬件自动清零；每次配置后只能对一次 SRAM 写操作生效
Bits 3:0	SRAM_P_INJ_BIT[3:0] : SRAM 奇偶校验注入位，从低到高对应 4 个字节的奇偶校验位

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



名称	说明	读写权限	复位值	字节地址
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	0xF00F_7763	0x4800_700C
SysCtrl_RST1	软件复位控制寄存器 1	R/W	0x0000_00E0	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
TIM2_	TIM17_	TIM16_	TIM15_	Res.								TIM1_	RAM1_	RAM0_	EFLS_
CLKEN	CLKEN	CLKEN	CLKEN									CLKEN	CLKEN	CLKEN	CLKEN
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.	USART1_	USART0_	SPI0_	Res.	I2C0_	ADC_	DMA_	CORDIC_	CRC_	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	TIM2_CLKEN: TIM2 时钟使能位，由软件进行清零和置位
		0: TIM2 时钟禁止（默认）
		1: TIM2 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
Bits	27:20	保留，必须保持复位值
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	USART1_CLKEN : USART1 时钟使能位，由软件进行清零和置位
	0: USART1 时钟禁止（默认）
	1: USART1 时钟使能
Bit 13	USART0_CLKEN : USART0 时钟使能位，由软件进行清零和置位
	0: USART0 时钟禁止（默认）
	1: USART0 时钟使能
Bit 12	SPIO_CLKEN : SPIO 时钟使能位，由软件进行清零和置位
	0: SPIO 时钟禁止（默认）
	1: SPIO 时钟使能
Bit 11	保留，必须保持复位值
Bit 10	I2C0_CLKEN : I2C0 时钟使能位，由软件进行清零和置位
	0: I2C0 时钟禁止（默认）
	1: I2C0 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	CORDIC_CLKEN : CORDIC 时钟使能位，由软件进行清零和置位
	0: CORDIC 时钟禁止（默认）
	1: CORDIC 时钟使能
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
Bits 5:2	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.	TIM17_ CLKSEL	TIM16_ CLKSEL	TIM15_ CLKSEL	TIM2_C LKSEL	TIM1_C LKSEL	Res.				SPIO_CLKSEL [1:0]	SPIO_CLK DIV	I2CO_CLKSEL [1:0]	I2CO_CL KDIV		
	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
USART1_ CLKSEL[1:0]		USART1_ CLKDIV	USART0_ CLKSEL[1:0]	USART0_ CLKDIV	Res.			RCH_D IV	Res.	MCO_SEL[2:0]			MCO_DIV[2:0]		
rw		rw	rw	rw				rw		rw			rw		

Bit	31	保留, 必须保持复位值
Bit	30	TIM17_CLKSEL: TIM17 时钟选择
		0: 系统时钟 SYS_CLK (默认)
		1: APB 时钟 PCLK0
Bit	29	TIM16_CLKSEL: TIM16 时钟选择
		0: 系统时钟 SYS_CLK (默认)
		1: APB 时钟 PCLK0
Bit	28	TIM15_CLKSEL: TIM15 时钟选择
		0: 系统时钟 SYS_CLK (默认)
		1: APB 时钟 PCLK0
Bit	27	TIM2_CLKSEL: TIM2 时钟选择
		0: 系统时钟 SYS_CLK (默认)
		1: APB 时钟 PCLK0
Bit	26	TIM1_CLKSEL: TIM1 时钟选择
		0: 系统时钟 SYS_CLK (默认)
		1: APB 时钟 PCLK0
Bits	25:22	保留, 必须保持复位值
Bits	21:20	SPIO_CLKSEL[1:0]: SPIO 时钟选择
		00: PCLK0 (默认)
		01: SYS_CLK_IN
		10: RCH_CLK
		11: 保留
Bit	19	SPIO_CLKDIV: SPIO 时钟二分频使能
		0: SPIO 时钟不分频 (默认)
		1: SPIO 时钟二分频
Bits	18:17	I2CO_CLKSEL[1:0]: I2CO 时钟选择
		00: PCLK0 (默认)
		01: SYS_CLK_IN
		10: RCH_CLK
		11: OSCL_CLK
Bit	16	I2CO_CLKDIV: I2CO 时钟二分频使能
		0: I2CO 时钟不分频 (默认)



	1: I2C0 时钟二分频
Bits 15:14	USART1_CLKSEL[1:0]: USART1 时钟选择
	00: PCLK0 (默认)
	01: SYS_CLK_IN
	10: RCH_CLK
	11: OSCL_CLK
Bit 13	USART1_CLKDIV: USART1 时钟二分频使能
	0: USART1 时钟不分频 (默认)
	1: USART1 时钟二分频
Bits 12:11	USART0_CLKSEL[1:0]: USART0 时钟选择
	00: PCLK0 (默认)
	01: SYS_CLK_IN
	10: RCH_CLK
	11: OSCL_CLK
Bit 10	USART0_CLKDIV: USART0 时钟二分频使能
	0: USART0 时钟不分频 (默认)
	1: USART0 时钟二分频
Bits 9:8	保留, 必须保持复位值
Bit 7	RCH_DIV: RCH 时钟分频使能, 分频后时钟可作为 WT 工作时钟
	0: RCH 时钟分频禁止 (默认)
	1: RCH 时钟分频使能
Bit 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: RCH_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.								DIV_ CLKEN	EXTI_ CLKEN	ANA_ CLKEN	Res.				
								rw	rw	rw					

Bits	31:8	保留, 必须保持复位值
Bit	7	DIV_CLKEN: DIV 时钟使能, 由软件进行清零和置位
		0: DIV 时钟禁止 (默认)
		1: DIV 时钟使能
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: 0xF00F_7763

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIM2 _RST	TIM17 _RST	TIM16 _RST	TIM15 _RST	Res.								TIM1 _RST	RAM1 _RST	RAM0 _RST	EFLS_ RST
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.	USART1 _RST	USART0 _RST	SPIO_ RST	Res.	I2C0_ RST	ADC_ RST	DMA_ RST	CORDIC _RST	CRC_ RST	Res.				GPIOB _RST	GPIOA _RST
	rw	rw	rw		rw	rw	rw	rw	rw					rw	rw

Bit	31	TIM2_RST: TIM2 复位, 由软件进行清零和置位
		0: 复位 TIM2
		1: 无影响 (默认)
Bit	30	TIM17_RST: TIM17 复位, 由软件进行清零和置位
		0: 复位 TIM17

	1: 无影响（默认）
Bit 29	TIM16_RST : TIM16 复位，由软件进行清零和置位
	0: 复位 TIM16
	1: 无影响（默认）
Bit 28	TIM15_RST : TIM15 复位，由软件进行清零和置位
	0: 复位 TIM15
	1: 无影响（默认）
Bits 27:20	保留，必须保持复位值
Bit 19	TIM1_RST : TIM1 复位，由软件进行清零和置位
	0: 复位 TIM1
	1: 无影响（默认）
Bit 18	RAM1_RST : SRAM1 端口逻辑复位，由软件进行清零和置位
	0: 复位 SRAM1 端口
	1: 无影响（默认）
Bit 17	RAM0_RST : SRAM0 端口逻辑复位，由软件进行清零和置位
	0: 复位 SRAM0 端口
	1: 无影响（默认）
Bit 16	EFSL_RST : Flash 端口复位，由软件进行清零和置位
	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	USART1_RST : USART1 复位，由软件进行清零和置位
	0: 复位 USART1
	1: 无影响（默认）
Bit 13	USART0_RST : USART0 复位，由软件进行清零和置位
	0: 复位 USART0
	1: 无影响（默认）
Bit 12	SPIO_RST : SPIO 复位，由软件进行清零和置位
	0: 复位 SPIO
	1: 无影响（默认）
Bit 11	保留，必须保持复位值
Bit 10	I2C0_RST : I2C0 复位，由软件进行清零和置位
	0: 复位 I2C0
	1: 无影响（默认）
Bit 9	ADC_RST : ADC 复位，由软件进行清零和置位
	0: 复位 ADC
	1: 无影响（默认）
Bit 8	DMA_RST : DMA 复位，由软件进行清零和置位
	0: 复位 DMA
	1: 无影响（默认）
Bit 7	CORDIC_RST : CORDIC 复位，由软件进行清零和置位
	0: 复位 CORDIC
	1: 无影响（默认）

Bit 6	CRC_RST: CRC 复位，由软件进行清零和置位
	0: 复位 CRC
	1: 无影响（默认）
Bits 5:2	保留，必须保持复位值
Bit 1	GPIOB_RST: GPIOB 复位，由软件进行清零和置位
	0: 复位 GPIOB
	1: 无影响（默认）
Bit 0	GPIOA_RST: GPIOA 复位，由软件进行清零和置位
	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_00E0

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.								DIV_ RST	EXTI_ RST	ANA_ RST	Res.				
								rw	rw	rw					

Bits 31:8	保留，必须保持复位值
Bit 7	DIV_RST: DIV 复位，由软件进行清零和置位
	0: 复位 DIV
	1: 无影响（默认）
Bit 6	EXTI_RST: EXTI 复位，由软件进行清零和置位
	0: 复位 EXTI
	1: 无影响（默认）
Bit 5	ANA_RST: 模拟模块控制器复位，由软件进行清零和置位
	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 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。

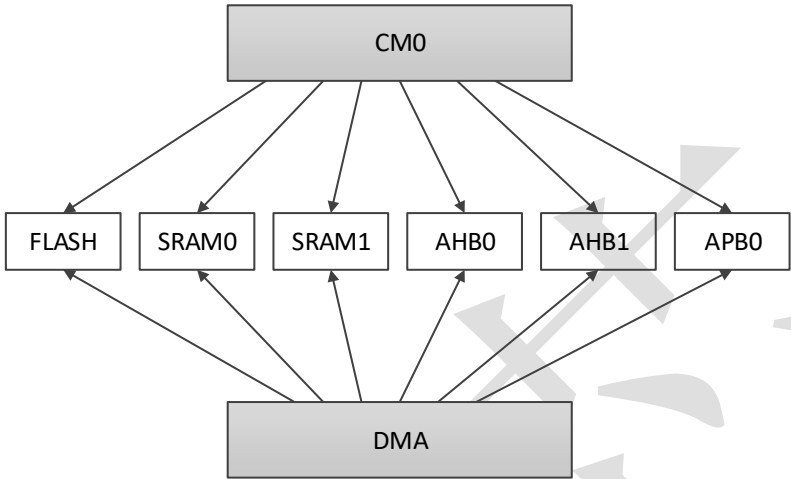


图 6-1 LCM32F06X 总线矩阵

6.2 外设互联矩阵

LCM32F06X 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 LCM32F06X 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
	DAC	D/A 转换触发
	ACMPx	比较器触发和输出消隐控制
ACMPx	TIMx	定时器输出控制、输入触发和刹车
	ADC	A/D 转换触发
CPU（HardFault） LVD ACMPx GPIO	TIM1 TIM15 TIM16 TIM17	定时器刹车输入
GPIO	TIMx	定时器外部触发和刹车
	ADC	A/D 转换外部触发
	DAC	D/A 转换触发
ADC	TIMx	定时器外部触发和刹车

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 0x05FA659A 到密钥寄存器，且在随后的 64 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_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_EDU_CFG8	互联配置寄存器 8	R/W	0x0000_0000	0x4800_7040
SysCtrl_EDU_CFG9	互联配置寄存器 9	R/W	0x0000_0000	0x4800_7044
SysCtrl_EDU_CFG11	互联配置寄存器 11	R/W	0x0000_0000	0x4800_704C
SysCtrl_EDU_CFG12	互联配置寄存器 12	R/W	0x0000_0000	0x4800_7050
SysCtrl_EDU_CFG15	互联配置寄存器 15	R/W	0x0000_0000	0x4800_705C
SysCtrl_EDU_CFG16	互联配置寄存器 16	R/W	0x0000_0000	0x4800_7060
SysCtrl_EDU_CFG17	互联配置寄存器 17	R/W	0x0000_0000	0x4800_7064
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 请求源通道 x 的触发源配置为 TIM1 DMA 请求 1 (默认)
	0001: DMA 请求源通道 x 的触发源配置为 TIM1 DMA 请求 2
	0010: DMA 请求源通道 x 的触发源配置为 TIM1 DMA 请求 3
	0011: DMA 请求源通道 x 的触发源配置为 TIM1 DMA 请求 4
	0100: DMA 请求源通道 x 的触发源配置为 ADC
	0101: DMA 请求源通道 x 的触发源配置为 TIM2_CH1 / TIM2_TRIG
	0110: DMA 请求源通道 x 的触发源配置为 TIM2_CH2 / TIM2_CH3

	0111: DMA 请求源通道 x 的触发源配置为 TIM2_CH4 / TIM2_UP
	1000: 保留
	1001: DMA 请求源通道 x 的触发源配置为 TIM15
	1010: DMA 请求源通道 x 的触发源配置为 TIM16
	1011: DMA 请求源通道 x 的触发源配置为 TIM17
	1100: 保留
	1101: 保留
	1110: 保留
	1111: DMA 请求源通道 x 的触发源配置为 DAC
Bits 15:0	DMA_REQx_MAP[3:0]: DMA 请求源通道配置 (x=0 到 3)
	0000: DMA 请求源通道 x 的触发源配置为 USART0_TX (默认)
	0001: DMA 请求源通道 x 的触发源配置为 USART0_RX
	0010: DMA 请求源通道 x 的触发源配置为 USART1_TX
	0011: DMA 请求源通道 x 的触发源配置为 USART1_RX
	0100: DMA 请求源通道 x 的触发源配置为 I2C0_TX
	0101: DMA 请求源通道 x 的触发源配置为 I2C0_RX
	0110: DMA 请求源通道 x 的触发源配置为 SPI0_TX
	0111: DMA 请求源通道 x 的触发源配置为 SPI0_RX
	1000: 保留
	1101: 保留
	1010: 保留
	1011: 保留
	1100: 保留
	1101: 保留
	1110: 保留
	1111: 保留

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.															

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: TIM2_TRGO

	010: ACMP0 输出
	011: ACMP1 输出
	100: 保留
	101: TIM15_TRIG_OC1
	110: TIM16_TRIG_OC1
	111: 保留
Bits 15:0	保留, 必须保持复位值
注: TIMx_TRIG_OC _i 和 TIMx_TRGO 不同, 详见 TIM 章节说明。	

6.3.3 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_TI2S[2:0]		
	rw				rw				rw				rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TIM16_TI1S[2:0]			Res.	TIM15_ETRS[2:0]			Res.	TIM15_TI2S[2:0]			Res.	TIM15_TI1S[2:0]		
	rw				rw				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: 外部 GPIO 输入, 参考相关 I/O 复用说明 (默认)
	001: TIM2_TRGO
	010: ACMP0 输出
	011: ACMP1 输出
	100: 保留
	101: TIM1_TRGO
	110: TIM1_TRIG_OC5
	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 窗口看门狗 1
	010: ACMP0 输出
	011: ACMP1 输出
	100: TIM2_TRGO
	101: TIM1_TRGO
	110: TIM1_TRIG_OC5
	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: 外部 GPIO 输入, 参考相关 I/O 复用说明 (默认)
	001: TIM2_TRGO
	010: ACMP0 输出
	011: ACMP1 输出
	100: 保留
	101: TIM1_TRGO
	110: TIM1_TRIG_OC5
	111: TIM15_TRGO
Bit 19	保留, 必须保持复位值
Bits 18:16	TIM16_TI2S[2:0]: TIM16 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 窗口看门狗 1
	010: ACMP0 输出
	011: ACMP1 输出
	100: TIM2_TRGO
	101: TIM1_TRGO
	110: TIM1_TRIG_OC5
	111: TIM15_TRGO
Bit 15	保留, 必须保持复位值
Bits 14:12	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 窗口看门狗 0
	010: ACMP0 输出
	011: ACMP1 输出
	100: TIM2_TRGO
	101: TIM1_TRGO
	110: TIM1_TRIG_OC5
	111: TIM15_TRGO
Bit 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: 外部 GPIO 输入, 参考相关 I/O 复用说明 (默认)
	001: TIM2_TRGO
	010: ACMP0 输出
	011: ACMP1 输出
	100: 保留
	101: TIM1_TRIG_OC1
	110: TIM1_TRIG_OC5
	111: TIM16_TRIG_OC1
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 窗口看门狗 0
	010: MCO

	011: ACMP0 输出
	100: ACMP1 输出
	101: TIM1_TRIG_OC1
	110: TIM1_TRIG_OC5
	111: TIM2_TRIG_OC1
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 窗口看门狗 0
	010: ADC 窗口看门狗 1
	011: ACMP0 输出
	100: ACMP1 输出
	101: TIM1_TRIG_OC1
	110: TIM1_TRIG_OC5
	111: TIM2_TRIG_OC1
注: TIMx_TRIG_OC _i 和 TIMx_TRGO 不同, 详见 TIM 章节说明。	

6.3.4 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 窗口看门狗 1/ADC 窗口看门狗 0/ TIM15_TRGO / TIM2_TRGO / 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 窗口看门狗 1/ADC 窗口看门狗 0/ TIM16_TRIG_OC1 /TIM17_TRIG_OC1/ 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 窗口看门狗 1/ADC 窗口看门狗 0/ TIM15_TRGO / TIM2_TRGO / 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
注: TIMx_TRIG_OCI 和 TIMx_TRGO 不同，详见 TIM 章节说明。	

6.3.5 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]							Res.	TIM17_SP[6:0]						
	rw								rw						

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

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 窗口看门狗 1/ADC 窗口看门狗 0/ TIM1_TRGO / TIM2_TRGO / 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
	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）
	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 15:0	保留，必须保持复位值
注: TIMx_TRIG_OC1 和 TIMx_TRGO 不同，详见 TIM 章节说明。	

6.3.6 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 窗口看门狗 1/ADC 窗口看门狗 0/TIM16_TRIG_OC1 / TIM17_TRIG_OC1 /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 窗口看门狗 1/ADC 窗口看门狗 0/TIM1_TRGO / TIM2_TRGO / 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

注：TIMx_TRIG_OC_i 和 TIMx_TRGO 不同，详见 TIM 章节说明。

6.3.7 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								

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 窗口看门狗 1/ADC 窗口看门狗 0/ TIM15_TRGO /TIM17_TRIG_OC1/ 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 窗口看门狗 1/ADC 窗口看门狗 0/ TIM1_TRGO / TIM2_TRGO / 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

注：TIMx_TRIG_OC_i 和 TIMx_TRGO 不同，详见 TIM 章节说明。

6.3.8 互联配置寄存器 8（SysCtrl_EDU_CFG8）

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.												TIM2_ETRS[2:0]			
												rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TIM2_TI4S[2:0]			Res.	TIM2_TI3S[2:0]			Res.	TIM2_TI2S[2:0]			Res.	TIM2_TI1S[2:0]		
	rw				rw				rw				rw		

Bits	31:19	保留，必须保持复位值
Bits	18:16	TIM2_ETRS[2:0] : TIM2 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: 保留
		010: ACMP0 输出
		011: ACMP1 输出
		100: TIM1_TRIG_OC1
		101: TIM1_TRIG_OC5
		110: TIM15_TRIG_OC1
		111: TIM16_TRIG_OC1
Bit	15	保留，必须保持复位值
Bits	14:12	TIM2_TI4S[2:0] : TIM2 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 窗口看门狗
		010: ACMP0 输出
		011: ACMP1 输出
		100: TIM1_TRIG_OC1
		101: TIM1_TRIG_OC5
		110: TIM15_TRIG_OC1
		111: TIM16_TRIG_OC1
Bit	11	保留，必须保持复位值
Bits	10:8	TIM2_TI3S[2:0] : TIM2 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 窗口看门狗
		010: ACMP0 输出
		011: ACMP1 输出
		100: TIM1_TRIG_OC1
		101: TIM1_TRIG_OC5
		110: TIM15_TRIG_OC1
		111: TIM16_TRIG_OC1
Bit	7	保留，必须保持复位值
Bits	6:4	TIM2_TI2S[2:0] : TIM2 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 窗口看门狗
		010: ACMP0 输出

	011: ACMP1 输出
	100: TIM1_TRIG_OC1
	101: TIM1_TRIG_OC5
	110: TIM15_TRIG_OC1
	111: TIM16_TRIG_OC1
Bit 3	保留, 必须保持复位值
Bits 2:0	TIM2_TI1S[2:0]: TIM2 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 窗口看门狗
	010: ACMP0 输出
	011: ACMP1 输出
	100: TIM1_TRIG_OC1
	101: TIM1_TRIG_OC5
	110: TIM15_TRIG_OC1
	111: TIM16_TRIG_OC1
注: TIMx_TRIG_OC _i 和 TIMx_TRGO 不同, 详见 TIM 章节说明。	

6.3.9 互联配置寄存器 9 (SysCtrl_EDU_CFG9)

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.		TIM2_OCCSE[5:0]						Res.							
		rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		TIM2_BKSE[6:0]						Res.	TIM2_SP[6:0]						
		rw							rw						

Bits 31:30	保留, 必须保持复位值
Bits 29:24	TIM2_OCCSE[5:0]: TIM2 OCREF_CLR 输入触发源使能控制, 默认值 000000
	TIM2 OCREF_CLR 输入源为 ADC 窗口看门狗 1/ADC 窗口看门狗 0/ TIM15_TRGO / TIM1_TRGO / 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 TIM2_OCCSE[x]配置
	所有输入源的有效触发电平, 通过控制位 TIM2_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
	TIM2_OCCSE[x] = 0: 相应触发源不使能
	TIM2_OCCSE[x] = 1: 相应触发源使能
Bits 23:15	保留, 必须保持复位值
Bits 14:8	TIM2_BKSE[6:0]: TIM2 刹车输入触发源使能控制, 默认值 0000000
	TIM2 刹车输入触发源为外部 GPIO 输入/ADC 窗口看门狗 1/ADC 窗口看门狗 0/ TIM1_TRGO / TIM15_TRGO / 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 TIM2_BKSE[x]配置

	所有输入源的有效触发电平，通过控制位 TIM2_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
	TIM2_BKSE[x] = 0: 相应触发源不使能
	TIM2_BKSE[x] = 1: 相应触发源使能
Bit 7	保留，必须保持复位值
Bits 6:0	TIM2_SP[6:0]: TIM2 刹车/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 TIM2_SP[x]配置（低位对齐）
	TIM2_SP[x] = 0: 相应触发源的有效电平是高
	TIM2_SP[x] = 1: 相应触发源的有效电平是低
注: TIMx_TRIG_OCI 和 TIMx_TRGO 不同，详见 TIM 章节说明。	

6.3.10 互联配置寄存器 11（SysCtrl_EDU_CFG11）

偏移地址：0x4C

复位值：0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.						ADC_EXTEN1[1:0]		ADC_EXTSEL1[7:0]							
						rw		rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						ADC_EXTEN0[1:0]		ADC_EXTSEL0[7:0]							
						rw		rw							

Bits 31:26	保留，必须保持复位值
Bits 25:24	ADC_EXTEN1[1:0]: ADC 触发源 1 使能控制
	00: 外部硬件上升沿触发（默认）
	01: 外部硬件下降沿触发
	10: 外部硬件双沿触发
	11: 禁止外部硬件触发，仅支持软件触发
Bits 23:16	ADC_EXTSEL1[7:0]: ADC 触发源 1 的选择，默认值 00000000
	ADC 触发源 1 可选择为 ACMP0 /TIM2_TRIG_OC4/TIM2_TRIG_OC3/ TIM2_TRIG_OC1 /EXTI3/ TIM15_TRIG_OC2 /TIM2_TRIG_OC2 /TIM1_TRIG_OC2 输出（从高位到低位）
	所有触发源均可被配置为使能或不使能，通过控制位 ADC_EXTSEL1[x]配置
	所有使能的触发源通过逻辑或的方式组合，有效触发方式通过控制 ADC_EXTEN1[1:0]配置
	ADC_EXTSEL1[x] = 0: 相应触发源不选择
	ADC_EXTSEL1[x] = 1: 相应触发源选择
Bits 15:10	保留，必须保持复位值
Bits 9:8	ADC_EXTEN0[1:0]: ADC 触发源 0 使能控制
	00: 外部硬件上升沿触发（默认）
	01: 外部硬件下降沿触发

	10: 外部硬件双沿触发
	11: 禁止外部硬件触发, 仅支持软件触发
Bits 7:0	ADC_EXTSEL0[7:0]: ADC 触发源 0 的选择, 默认值 00000000
	ADC 触发源 0 可选择为 TIM1_TRIG_OC5 / TIM1_TRIG_OC4 / TIM1_TRIG_OC3 / TIM1_TRIG_OC2 / EXT11 / TIM15_TRIG_OC1 / TIM2_TRIG_OC1 / TIM1_TRIG_OC1 输出 (从高位到低位)
	所有触发源均可被配置为使能或不使能, 通过控制位 ADC_EXTSEL0[x]配置
	所有使能的触发源通过逻辑或的方式组合, 有效触发方式通过控制 ADC_EXTEN0[1:0]配置
	ADC_EXTSEL0[x] = 0: 相应触发源不选择
	ADC_EXTSEL0[x] = 1: 相应触发源选择
注: TIMx_TRIG_OC _i 和 TIMx_TRGO 不同, 详见 TIM 章节说明。	

6.3.11 互联配置寄存器 12 (SysCtrl_EDU_CFG12)

偏移地址: 0x50

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.						ADC_EXTEN3[1:0]		ADC_EXTSEL3[7:0]							
						rw		rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						ADC_EXTEN2[1:0]		ADC_EXTSEL2[7:0]							
						rw		rw							

Bits 31:26	保留, 必须保持复位值
Bits 25:24	ADC_EXTEN3[1:0]: ADC 触发源 3 使能控制
	00: 外部硬件上升沿触发 (默认)
	01: 外部硬件下降沿触发
	10: 外部硬件双沿触发
	11: 禁止外部硬件触发, 仅支持软件触发
Bits 23:16	ADC_EXTSEL3[7:0]: ADC 触发源 3 的选择, 默认值 00000000
	ADC 触发源 3 可选择为 TIM1_TRIG_OC5/TIM15_TRGO/TIM2_TRGO/ TIM1_TRGO /EXT11 / TIM17_TRIG_OC1 /TIM2_TRIG_OC4 /TIM1_TRIG_OC4 输出 (从高位到低位)
	所有触发源均可被配置为使能或不使能, 通过控制位 ADC_EXTSEL3[x]配置
	所有使能的触发源通过逻辑或的方式组合, 有效触发方式通过控制 ADC_EXTEN3[1:0]配置
	ADC_EXTSEL3[x] = 0: 相应触发源不选择
	ADC_EXTSEL3[x] = 1: 相应触发源选择
Bits 15:10	保留, 必须保持复位值
Bits 9:8	ADC_EXTEN2[1:0]: ADC 触发源 2 使能控制
	00: 外部硬件上升沿触发 (默认)

	01: 外部硬件下降沿触发
	10: 外部硬件双沿触发
	11: 禁止外部硬件触发, 仅支持软件触发
Bits 7:0	ADC_EXTSEL2[7:0]: ADC 触发源 2 的选择, 默认值 00000000
	ADC 触发源 2 可选择为 ACMP1 / ACMP1 / TIM17_TRIG_OC1 / TIM16_TRIG_OC1 / EXTI9 / TIM16_TRIG_OC1 / TIM2_TRIG_OC3 / TIM1_TRIG_OC3 输出 (从高位到低位)
	所有触发源均可被配置为使能或不使能, 通过控制位 ADC_EXTSEL2[x]配置
	所有使能的触发源通过逻辑或的方式组合, 有效触发方式通过控制 ADC_EXTEN2[1:0]配置
	ADC_EXTSEL2[x] = 0: 相应触发源不选择
	ADC_EXTSEL2[x] = 1: 相应触发源选择
注: TIMx_TRIG_OC _i 和 TIMx_TRGO 不同, 详见 TIM 章节说明。	

6.3.12 互联配置寄存器 15 (SysCtrl_EDU_CFG15)

偏移地址: 0x5C

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
CP1_BLANKING[7:0]								CP0_BLANKING[7:0]							
rw								rw							

Bits 31:16	保留, 必须保持复位值
Bits 15:8	CP1_BLANKING[7:0]: ACMP1 消隐源选择
	ACMP1 消隐源可选择为软件/ TIM2_TRIG_OC4/ TIM15_TRIG_OC2/ TIM15_TRIG_OC1 / TIM1_TRIG_OC5 / TIM1_TRIG_OC3 / TIM1_TRIG_OC2/ TIM1_TRIG_OC1 输出 (从高位到低位)
	所有消隐源均可被配置为使能或不使能, 通过控制位 CP1_BLANKING[x]配置: 当 CP1_BLANKING[7]配置为 1 时为软件消隐, 会直接触发消隐
	所有使能的消隐源通过逻辑或的方式组合
	CP1_BLANKING[x] = 0: 相应消隐源不选择 (默认)
	CP1_BLANKING[x] = 1: 相应消隐源选择
Bits 7:0	CP0_BLANKING[7:0]: ACMP0 消隐源选择
	ACMP0 消隐源可选择为软件/ TIM2_TRIG_OC4/ TIM15_TRIG_OC2/ TIM15_TRIG_OC1 / TIM1_TRIG_OC5 / TIM1_TRIG_OC3 / TIM1_TRIG_OC2/ TIM1_TRIG_OC1 输出 (从高位到低位)
	所有消隐源均可被配置为使能或不使能, 通过控制位 CP0_BLANKING[x]配置: 当 CP0_BLANKING[7]配置为 1 时为软件消隐, 会直接触发消隐
	所有使能的消隐源通过逻辑或的方式组合
	CP0_BLANKING[x] = 0: 相应消隐源不选择 (默认)

	CP0_BLANKING[x] = 1: 相应消隐源选择
注: TIMx_TRIG_OC _i 和 TIMx_TRGO 不同, 详见 TIM 章节说明。	

6.3.13 互联配置寄存器 16 (SysCtrl_EDU_CFG16)

偏移地址: 0x60

复位值: 0x0000_0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CP1_EXTSP1	CP1_EXTSP0	CP0_EXTSP1	CP0_EXTSP0	CP1_EXTSEL1 [1:0]	CP1_EXTSEL0 [1:0]	CP0_EXTSEL1 [1:0]	CP0_EXTSEL0 [1:0]	CP1_EXTEN [1:0]	CP0_EXTEN [1:0]						
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	CP1_EXTSP1: ACMP1 第 2 组触发源极性选择位
		0: 同相输入 (默认)
		1: 反相输入
Bit	14	CP1_EXTSP0: ACMP1 第 1 组触发源极性选择位
		0: 同相输入 (默认)
		1: 反相输入
Bit	13	CP0_EXTSP1: ACMP0 第 2 组触发源极性选择位
		0: 同相输入 (默认)
		1: 反相输入
Bit	12	CP0_EXTSP0: ACMP0 第 1 组触发源极性选择位
		0: 同相输入 (默认)
		1: 反相输入
Bits	11:10	CP1_EXTSEL1[1:0]: ACMP1 第 2 组触发源选择
		00: TIM1_TRIG_OC3 (默认)
		01: TIM1_TRIG_OC5
		10: TIM16_TRIG_OC1
		11: TIM17_TRIG_OC1
Bits	9:8	CP1_EXTSEL0[1:0]: ACMP1 第 1 组触发源选择
		00: TIM1_TRIG_OC1 (默认)
		01: TIM1_TRIG_OC2
		10: TIM2_TRGO
		11: TIM15_TRGO
Bits	7:6	CP0_EXTSEL1[1:0]: ACMP0 第 2 组触发源选择
		00: TIM1_TRIG_OC3 (默认)
		01: TIM1_TRIG_OC5
		10: TIM16_TRIG_OC1

	11: TIM17_TRIG_OC1
Bits 5:4	CP0_EXTSEL0[1:0]: ACMP0 第 1 组触发源选择
	00: TIM1_TRIG_OC1 (默认)
	01: TIM1_TRIG_OC2
	10: TIM2_TRGO
	11: TIM15_TRGO
Bits 3:2	CP1_EXTEN[1:0]: ACMP1 触发方式
	00: 外部上升沿触发 (默认)
	01: 外部下降沿触发
	10: 外部双沿触发
	11: 禁止外部触发, 仅支持软件触发
Bits 1:0	CP0_EXTEN[1:0]: ACMP0 触发方式
	00: 外部上升沿触发 (默认)
	01: 外部下降沿触发
	10: 外部双沿触发
	11: 禁止外部触发, 仅支持软件触发

6.3.14 互联配置寄存器 17 (SysCtrl_EDU_CFG17)

偏移地址: 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
Res.									DAC_TSEL[2:0]			Res.		DAC_TEN[1:0]	
									rw					rw	

Bits 31:7	保留, 必须保持复位值
Bits 6:4	DAC_TSEL[2:0]: DAC 触发源选择
	000: 软件触发 (默认)
	001: TIM1_TRGO
	010: TIM2_TRGO
	011: TIM15_TRGO
	100: TIM16_TRIG_OC1
	101: TIM17_TRIG_OC1
	110: EXTI1
	111: EXTI9
Bits 3:2	保留, 必须保持复位值
Bits 1:0	DAC_TEN[1:0]: DAC 触发源使能控制
	00: 外部上升沿触发, 触发事件发生时, DAC_DHR 寄存器的值导入到转换寄存器 DAC_DOR (默认)

	01: 外部下降沿触发, 触发事件发生时, DAC_DHR 寄存器的值导入到转换寄存器 DAC_DOR
	10: 外部双沿触发, 触发事件发生时, DAC_DHR 寄存器的值导入到转换寄存器 DAC_DOR
	11: 禁止外部触发, 写入寄存器 DAC_DHR 的值直接传入转换寄存器 DAC_DOR
注: TIMx_TRIG_OC _i 和 TIMx_TRGO 不同, 详见 TIM 章节说明。	

6.3.15 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 64 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

7. 通用 I/O 端口（GPIO）

LCM32F06X 包含多达 25 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，使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
- 推挽输出
- 开源输出
- 推挽复用功能
- 开漏复用功能

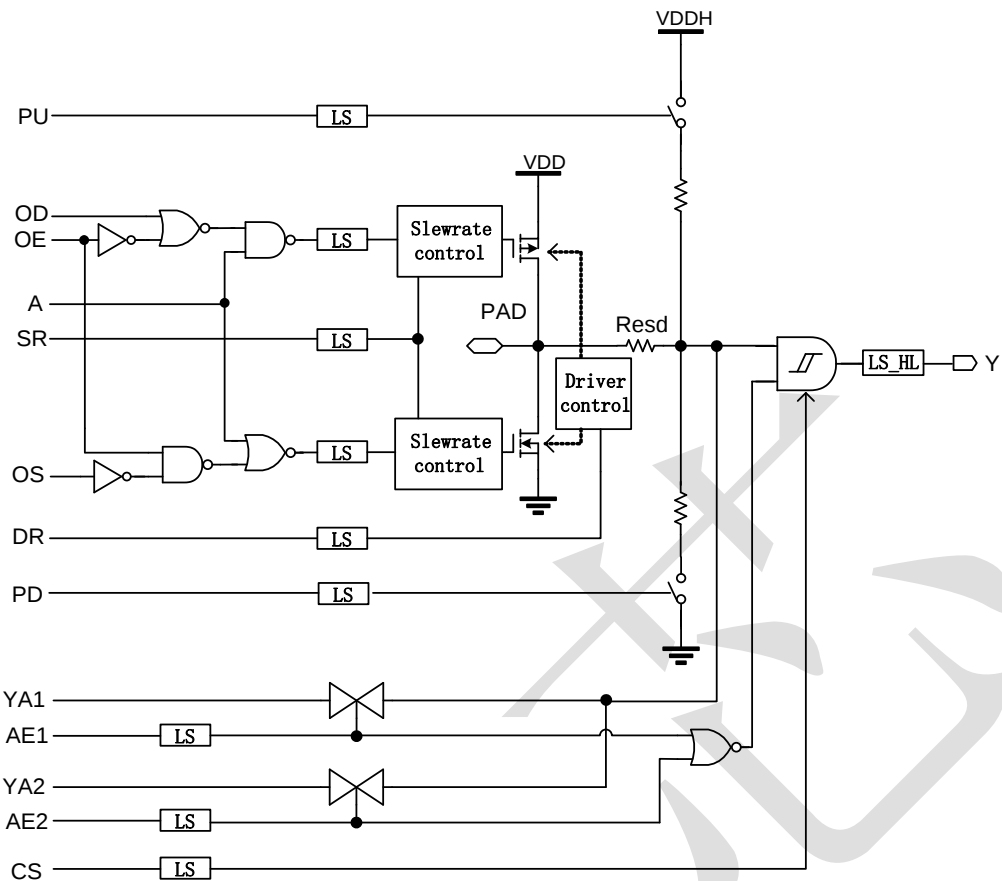


图 7-1 GPIO 接口的基本结构

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 1.2V	1.2V
GND	G	数字地	0
PAD	I/O	PAD 端	3.3V/5.5V
Y	I	数据输入信号	1.2V
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	1.2V
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	1.2V
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）	1.2V



A	I	推挽输出的输入信号	1.2V
OE(MODER)	I	推挽输出使能, OE=0: 高阻态输出 OE=1 且 OS=0 且 OD=0, 推挽输出	1.2V
OS	I	开源输出使能 OE=1 且 OS=1, 开源输出 (高电平接 VDD, 低电平为高阻态)	1.2V
OD	I	开漏输出使能 OE=1 且 OD=1, 开漏输出	1.2V
PU	I	弱上拉电阻使能, PU=1 时弱上拉有效	1.2V
PD	I	弱下拉电阻使能, PD=1 时弱下拉有效	1.2V
SR	I	压摆率控制, 0: 高压摆率, 1: 低压摆率	1.2V
DR	I	驱动强度控制, 0: 大驱动; 1: 小驱动	1.2V

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
	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)
	2ANA_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
	2OP	包含两路复用模拟通道, 两路低内阻模拟开关
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 LCM32F06X 引脚定义及模拟复用功能 (AN) 映射

引脚名	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)	
					YA1	YA2
PB7	I/O	2ANA	TIM1_PWM2 TIM1_PWM3 TIM1_PWM6 TIM17_CH1 TIM17_CH1N I2C0_SDA USART0_RX USART0_TX		ADCIN[8] OPA1_N0 ACMP0_N4 ACMP1_P3 ACMP1_N1 (PB0_YA2)	
PB8	I/O	2ANA	TIM1_PWM2	BOOT ¹	ADCIN[13]	ADCIN[14]

(BOOT0)			TIM1_PWM3 TIM1_PWM5 TIM15_CH2 TIM16_CH1 TIM17_CH1N TIM17_ETR I2CO_ALERT		(PA2_YA1/ PA12_YA2/ PB5_YA2)	(PA9_YA2/ PA11_YA2)
PA11	I/O	2OP	TIM1_PWM3 TIM15_CH1 I2CO_SDA USART0_TX USART0_CTS WT_Buz		OPA1_P1 ACMP0_P7 ACMP1_N2	ADCIN[14] OSCL_OUT (PA9_YA2/ PB8_YA2)
PA12	I/O	2OP	TIM1_PWM2 TIM15_CH1N I2CO_SCL USART0_RX USART0_RTS WT_nBuz		OPA0_N1 OPA1_N1 OPA2_N1 ACMP1_P4 (PA6_YA2)	ADCIN[13] OSCL_IN (PA2_YA1/ PB5_YA2/ PB8_YA1)
PB2 (nRST)	I/O	2ANA	TIM1_PWM1 TIM2_CH3 TIM15_CH2 TIM16_CH1N TIM1/2/15/16/17_GPIO_BKIN USART0_TX USART1_TX	SWCLK ² nRST ³		
VSS	S					
VSSA	S					
PA2	I/O	2ANA_OP	TIM1_PWM4 TIM1_PWM5 TIM15_CH2 USART1_RX USART1_TX SPI0_RXD SPI0_TXD ACMP0_out		ADCIN[13] (PA12_YA2/ PB5_YA2/ PB8_YA1)	ADCIN[2] OPA2_O ACMP1_P5 (PB1_YA2)
VDDH	S					
VDDA	S					
PA0	I/O	2ANA_OP	TIM2_CH1 TIM16_CH1 TIM1_ETR TIM15_ETR I2CO_ALERT USART1_RX		ADCIN[0] ACMP0_P4 ACMP0_N0 (PA13_YA1)	OPA2_P1

			USART1_CTS ACMP0_out			
PA1	I/O	2ANA_OP	TIM2_CH2 TIM15_CH1N TIM16_CH1N TIM16_CH2 TIM16_ETR I2C0_SDA USART1_RTS USART1_TX		ADCIN[1] ACMP0_P5 ACMP0_N1 OPA1_OX (PA15_YA1)	OPA0_N0 OPA2_N0 ACMP1_N3 (PA4_YA2/ PA9_YA1)
PA3	I/O	2ANA_OP	TIM2_CH3 TIM15_CH1 I2C0_SCL USART1_TX USART1_RX SPI0_RXD SPI0_TXD ACMP1_out		ADCIN[3] ACMP0_N2 ACMP0_P6 ACMP1_N0	OPA0_P1
PA5	I/O	2ANA_OP	TIM1_PWM6 TIM15_CH1 TIM16_CH2 TIM2_ETR TIM15_ETR I2C0_SCL SPI0_SCK WT_nBuz		ADCIN[4] ACMP0_P0 ACMP1_P1 (PA4_YA1/ PB0_YA1/ PB4_YA1)	ADCIN[5] OPA2_P0 ACMP1_P2 (PB3_YA1)
PA4	I/O	2ANA_OP	TIM15_CH2 TIM17_CH1 USART0_RTS USART1_TX USART1_CK SPI0_FSS ACMP1_out WT_Buz		ADCIN[4] ACMP0_P0 ACMP1_P1 (PA5_YA1/ PB0_YA1/ PB4_YA1)	OPA0_N0 OPA2_N0 ACMP1_N3 (PA1_YA2/ PA9_YA1)
PA7	I/O	2OP	TIM1_PWM3 TIM1_PWM5 TIM2_CH2 TIM17_CH1 USART0_RX SPI0_RXD SPI0_TXD ACMP1_out		ADCIN[7] OPA1_P0 ACMP0_P2 (PB6_YA1)	ADCIN[12] OPA1_O ACMP1_P6
PA6	I/O	2ANA_OP	TIM1_PWM5 TIM2_CH1		ADCIN[6]	OPA0_N1 OPA1_N1

			TIM16_CH1 TIM1_GPIO_BKIN USART0_CK SPI0_RXD SPI0_TXD ACMP0_out		ACMP0_P1 ACMP1_N4 (PB5_YA1)	OPA2_N1 ACMP1_P4 (PA12_YA1)
PB0	I/O	2ANA_OP	TIM1_PWM6 TIM2_CH3 USART0_RX USART0_TX EVENTOUT		ADCIN[4] ACMP0_P0 ACMP1_P1 (PA4_YA1/ PA5_YA1/ PB4_YA1)	ADCIN[8] OPA1_N0 ACMP1_P3 ACMP1_N1 ACMP0_N4 (PB7_YA1)
PB1	I/O	2ANA_OP	TIM1_PWM7 TIM2_CH4 TIM15_CH1 I2C0_ALERT USART0_RX USART0_TX USART1_RTS		ADCIN[9] ACMP0_N3 ACMP1_P0 OPA2_OX (PB6_YA2)	ADCIN[2] OPA2_O ACMP1_P5 (PA2_YA2)
PA8	I/O	2ANA_OP	TIM1_PWM1 TIM15_CH2 TIM16_GPIO_BKIN TIM17_ETR USART0_TX USART0_CTS ACMP0_out MCO		ADCIN[10] ACMP0_P3 OPA0_OX ELV1 ⁴ (PA14_YA1)	ADCIN[11] OPA0_O ACMP1_P7
PA9	I/O	2ANA_OP	TIM1_PWM2 TIM15_CH1 TIM15_CH2N TIM15_GPIO_BKIN TIM16_GPIO_BKIN I2C0_SCL USART0_RX USART0_TX		OPA0_N0 OPA2_N0 ACMP1_N3 (PA1_YA2/ PA4_YA2)	ADCIN[14] (PA11_YA2/ PB8_YA2)
PA10	I/O	2ANA_OP	TIM1_PWM3 TIM1_PWM8 TIM15_CH2 TIM1_GPIO_BKIN TIM17_GPIO_BKIN I2C0_SDA USART0_RX USART0_TX		ADCIN[15] OPA0_P0	DACOUT_BUF (PA13_YA2)
PA13	I/O	2ANA	TIM1_PWM4	SWDIO ²	ADCIN[0]	DACOUT_BUF

			TIM1_PWM5 TIM16_CH1 I2CO_SDA USART1_RX ACMP1_out IR_OUT		ACMP0_P4 ACMP0_N0 (PA0_YA1)	(PA10_YA2)
PA14	I/O	2ANA	TIM1_PWM8 TIM1_CH5 TIM17_CH1 TIM1_GPIO_BKIN I2CO_SCL USART1_RX USART1_TX	SWCLK ²	ADCIN[10] ACMP0_P3 OPA0_OX ELVI ⁴ (PA8_YA1)	PMU_AOUT (PA15_YA2)
PA15	I/O	2ANA	TIM1_PWM1 TIM1_PWM4 TIM16_CH1N TIM17_CH1 TIM1_GPIO_BKIN TIM15_GPIO_BKIN USART1_RX SPI0_FSS		ADCIN[1] ACMP0_P5 ACMP0_N1 OPA1_OX (PA1_YA1)	PMU_AOUT ⁶ (PA14_YA2)
PB3	I/O	2ANA	TIM1_PWM1 TIM1_PWM5 TIM1_PWM8 TIM1_PWM7 TIM16_CH1 TIM16_ETR USART1_TX SPI0_SCK		ADCIN[5] OPA2_P0 ACMP1_P2 (PA5_YA2)	
PB4	I/O	2ANA	TIM1_PWM6 TIM1_PWM7 TIM2_CH1 TIM16_CH2 TIM17_GPIO_BKIN USART0_RX SPI0_RXD SPI0_TXD		ADCIN[4] ACMP0_P0 ACMP1_P1 (PA4_YA1/ PA5_YA1/ PB0_YA1)	
PB5	I/O	2ANA_OP	TIM1_PWM3 TIM1_PWM4 TIM1_PWM6 TIM1_PWM8 TIM2_CH2 USART0_TX SPI0_RXD		ADCIN[6] ACMP0_P1 ACMP1_N4 (PA6_YA1)	ADCIN[13] (PA2_YA1/ PA12_YA2/ PB8_YA1)

			SPIO_TXD			
PB6	I/O	2ANA	TIM1_PWM1 TIM1_PWM5 TIM2_CH3 TIM15_CH2 TIM16_CH1N I2C0_SCL USART0_RX USART0_TX		ADCIN[7] OPA1_P0 ACMP0_P2 (PA7_YA1)	ADCIN[9] ACMP0_N3 ACMP1_P0 OPA2_OX (PB1_YA1)

注1：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2：根据选项字节配置，在系统复位期间可选择两组SWCLK（PB2或PA14）引脚中的一组和SWDIO（PA13）生效，生效后该组SWDIO引脚内部上拉，SWCLK引脚内部下拉。

注3：上电复位后，PB2引脚缺省配置为外部复位引脚nRST；如果通过选项字节配置为SWCLK，则在芯片开始工作时PB2会作为SWCLK，此时外部复位配置自动关闭，之后由软件配置。

注4：ELVI为可被LVD进行电压检测的外部电压。

注5：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。

注6：PMU_AOUT为芯片的电压观测输出。

表 7-4 端口 A 可选复用功能（AF）映射

引脚	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7
PA0	TIM15_ETR	USART1_CTS	TIM1_ETR	TIM16_CH1	USART1_RX	TIM2_CH1	ACMP0_out	I2C0_ALERT
PA1	TIM16_ETR	USART1_RTS	TIM16_CH1N	TIM2_CH2	USART1_TX	TIM15_CH1N	TIM16_CH2	I2C0_SDA
PA2	TIM15_CH2	USART1_RX	TIM1_PWM4	USART1_TX	TIM1_PWM5	SPIO_TXD	ACMP0_out	SPIO_RXD
PA3	TIM15_CH1	USART1_TX	SPIO_RXD	TIM2_CH3	SPIO_TXD	I2C0_SCL	ACMP1_out	USART1_RX
PA4	SPIO_FSS	USART0_RTS	TIM15_CH2	TIM17_CH1	USART1_CK	Wt_Buz	ACMP1_out	USART1_TX
PA5	SPIO_SCK	TIM16_CH2	I2C0_SCL	TIM15_CH1	TIM1_PWM6	Wt_nBuz	TIM15_ETR	TIM2_ETR
PA6	SPIO_RXD	TIM2_CH1	TIM1_GPIO_BKIN	ACMP0_out	TIM1_PWM5	TIM16_CH1	USART0_CK	SPIO_TXD
PA7	SPIO_TXD	TIM2_CH2	TIM1_PWM5	ACMP1_out	USART0_RX	TIM17_CH1	TIM1_PWM3	SPIO_RXD
PA8	MCO	USART0_CTS	TIM1_PWM1	TIM15_CH2	USART0_TX	TIM16_GPIO_BKIN	ACMP0_out	TIM17_ETR
PA9	TIM15_CH2N	TIM15_GPIO_BKIN	USART0_TX	TIM1_PWM2	I2C0_SCL	TIM16_GPIO_BKIN	TIM15_CH1	USART0_RX
PA10	TIM15_CH2	TIM17_GPIO_BKIN	USART0_RX	TIM1_PWM3	I2C0_SDA	USART0_TX	TIM1_PWM8	TIM1_GPIO_BKIN
PA11	-	I2C0_SDA	TIM1_PWM3	TIM15_CH1	USART0_TX	Wt_Buz	USART0_CTS	-
PA12	-	I2C0_SCL	TIM1_PWM2	TIM15_CH1N	USART0_RX	Wt_nBuz	USART0_RTS	-
PA13	SWDIO	IR_OUT	TIM1_PWM5	TIM1_PWM4	USART1_RX	TIM16_CH1	ACMP1_out	I2C0_SDA
PA14	SWCLK	USART1_TX	TIM1_CH5	TIM1_GPIO_BKIN	TIM17_CH1	TIM1_PWM8	I2C0_SCL	USART1_RX
PA15	SPIO_FSS	USART1_RX	TIM17_CH1	TIM15_GPIO_BKIN	TIM1_PWM1	TIM16_CH1N	TIM1_GPIO_BKIN	TIM1_PWM4

表 7-5 端口 B 可选复用功能（AF）映射

引脚	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7
PB0	EVENTOUT	TIM2_CH3	TIM1_PWM6	-	USART0_TX	-	USART0_RX	-
PB1	TIM15_CH1	TIM2_CH4	TIM1_PWM7	-	USART1_RTS	USART0_RX	I2C0_ALERT	USART0_TX
PB2	SWCLK	TIM1/2/15/16/17_GPIO_BKIN	TIM1_PWM1	TIM2_CH3	TIM15_CH2	TIM16_CH1N	USART1_TX	USART0_TX
PB3	SPI0_SCK	TIM1_PWM5	TIM1_PWM7	TIM16_CH1	TIM16_ETR	TIM1_PWM8	TIM1_PWM1	USART1_TX
PB4	SPI0_RXD	TIM2_CH1	TIM1_PWM6	USART0_RX	TIM1_PWM7	TIM17_GPIO_BKIN	TIM16_CH2	SPI0_TXD
PB5	SPI0_TXD	TIM2_CH2	TIM1_PWM8	TIM1_PWM3	USART0_TX	TIM1_PWM4	TIM1_PWM6	SPI0_RXD
PB6	USART0_TX	I2C0_SCL	TIM16_CH1N	TIM15_CH2	TIM1_PWM5	TIM1_PWM1	TIM2_CH3	USART0_RX
PB7	USART0_RX	I2C0_SDA	TIM17_CH1N	TIM1_PWM6	USART0_TX	TIM1_PWM3	TIM17_CH1	TIM1_PWM2
PB8	TIM17_ETR	TIM1_PWM5	TIM1_PWM2	TIM1_PWM3	TIM15_CH2	TIM16_CH1	TIM17_CH1N	I2C0_ALERT

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。

表 7-6 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	0x00
GPIOx_OD_BSRR	端口开漏设置/清除寄存器	R/W	0x0000_0000	0x04
GPIOx_PU_BSRR	端口上拉设置/清除寄存器	R/W	GPIOA: 0x0000_2000 GPIOB: 0x0000_0000	0x0C
GPIOx_PD_BSRR	端口下拉设置/清除寄存器	R/W	GPIOA: 0x0000_X000 GPIOB: 0x0000_000X	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_AFR1	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_SR_BSRR	端口斜率和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）

地址偏移：0x00

复位值: GPIOA: 0x2800_0000; GPIOB: 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: 根据选项字节配置, 在系统复位期间 PA13 会被选为 SWDIO, PA14 或 PB2 会有一组引脚被选为 SWCLK, 生效后该组 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)

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 端口上拉设置/清除寄存器 (GPIOx_PU_BSRR) (x=A,B)

地址偏移: 0x0C

复位值: GPIOA: 0x0000_2000; GPIOB: 0x0000_000X

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: 根据选项字节配置, 在系统复位期间 PA13 会被选为 SWDIO, PA14 或 PB2 会有一组引脚被选为 SWCLK, 生效后该组 SWDIO 引脚内部上拉, SWCLK 引脚内部下拉。	

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

地址偏移: 0x10

复位值: GPIOA: 0x0000_X000; GPIOB: 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: 根据选项字节配置, 在系统复位期间 PA13 会被选为 SWDIO, PA14 或 PB2 会有一组引脚被选为 SWCLK, 生效后该组 SWDIO 引脚内部上拉, SWCLK 引脚内部下拉。	

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

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

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

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...15), 在 GPIO 位输出情况下, 寄存器数据出现在对应引脚

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

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...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...15)
	0: 对输出数据位无操作
	1: 设置输出数据位

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

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。位[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 LCK[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）
	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
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

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

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

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

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	ARFy[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

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

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

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

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

7.4.13 GPIO 端口斜率和施密特触发器输入设置 / 清除寄存器

(GPIOx_CS_SR_BSRR) (x=A,B)

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 斜率和 CMOS/施密特触发器输入配置复位 (y=0...15)
	0: 对斜率和 CMOS/施密特触发器输入配置位无操作
	1: 复位斜率和 CMOS/施密特触发器输入配置位
Bits 15:0	读:
	CSy : 端口 x 斜率和 CMOS/施密特触发器输入配置状态
	0: 快速斜率, 施密特触发器输入 (默认)
	1: 缓慢斜率, CMOS 输入
	写:
	BSy : 端口 x 斜率和 CMOS/施密特触发器输入配置位 (y=0...15)
	0: 对斜率和 CMOS/施密特触发器输入配置位无操作
	1: 设置斜率和 CMOS/施密特触发器输入配置位

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

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：SPI、I2C、USART、ADC、DAC 和 TIMER。

8.1 主要特性

- 2 个独立可配置的通道，每个通道都包含源和目标对象
- 每个通道拥有不同的优先级
- 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

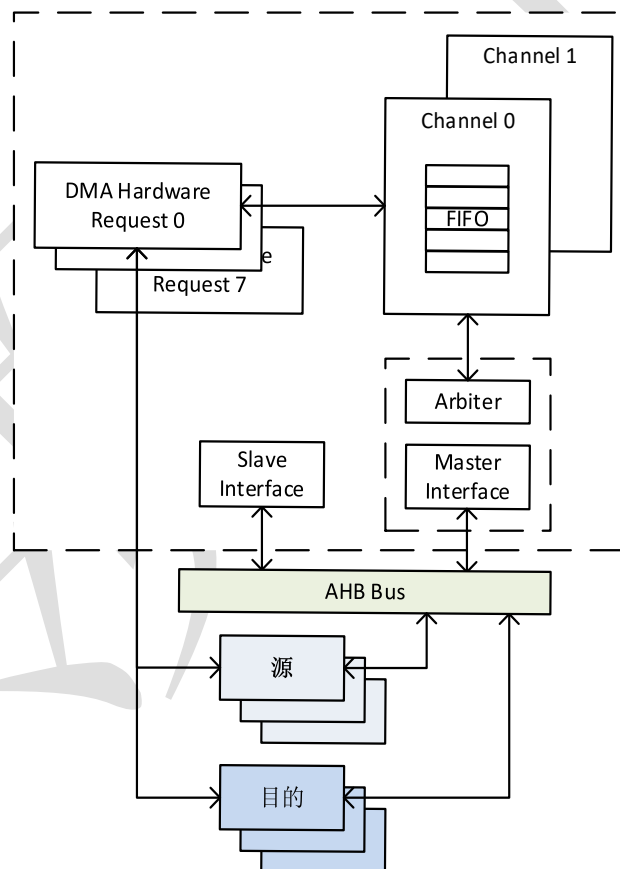


图 8-1 DMA 结构框图

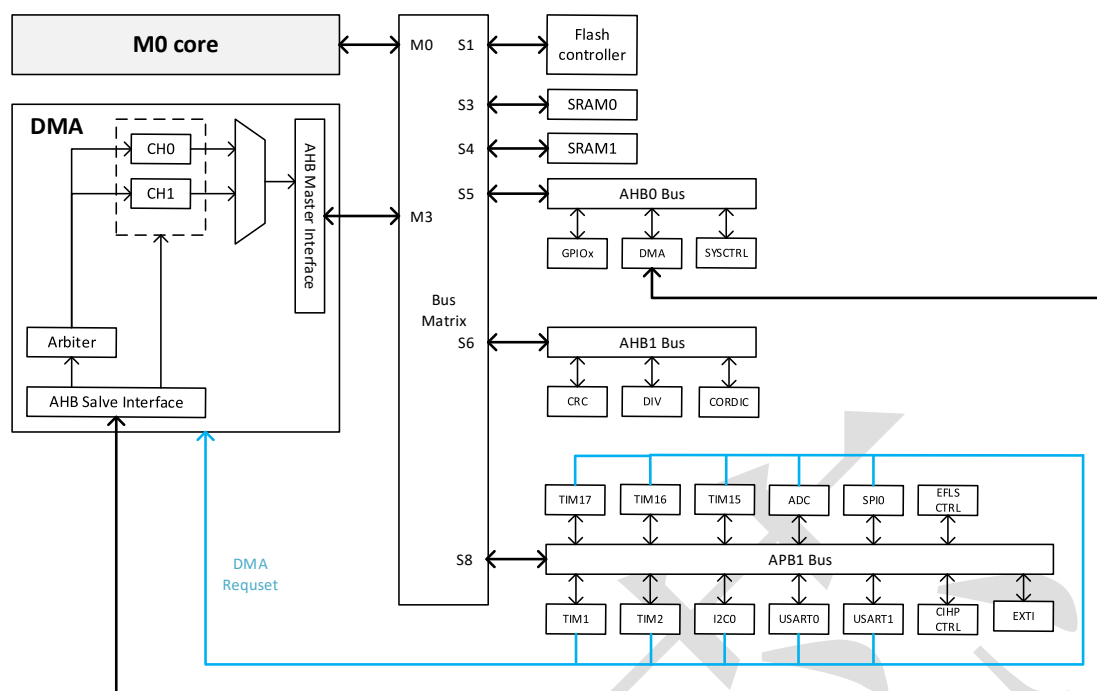


图 8-2 DMA 连接框图

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。

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。

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。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权。例如，通道 0 的优先级 > 通道 1，同时请求的时候，通道 0 的请求优先于通道 1 被响应。

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：

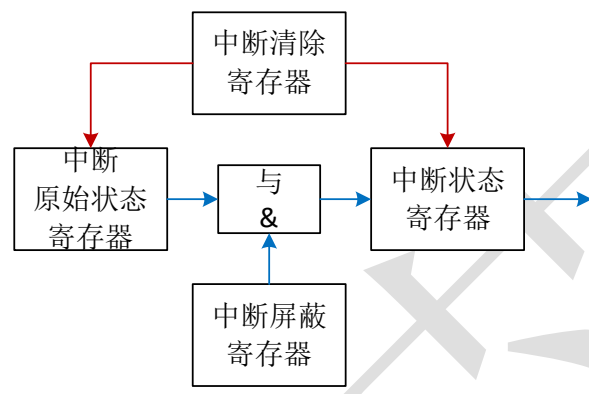


图 8-3 DMA 中断逻辑

8.2.4 DMA 请求与数据传输

DMA 的数据传输过程分为连续传输和单次传输两种模式，USART、SPI、I2C 通讯外设支持两种模式请求，TIMER 和 ADC 外设仅支持连续传输请求。

连续传输：DMA 接收到外设的连续传输请求后，根据寄存器设定的数据长度进行一次连续传输。在 block 传输的最后一次连续传输时，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连续传输请求，且为 block 传输中最后一次传输的剩余数量不满足一次连续传输数据数量时，则进入单次传输模式。单次传输模式可以将外设 FIFO 内不满一次连续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：

	示例一	示例二	示例三	示例四	示例五
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	SRAM
CTLx.TT_FC	001	010	001	010	000
CTLx.BLOCK_TS	8	10	6	10	8
CTLx.SRC_TR_WIDTH	32 bits	16 bits	32 bits	16 bits	32 bits
CTLx.DST_TR_WIDTH	32 bits	32 bits	32 bits	32 bits	32 bits
CTLx.SRC_MSIZE	001（长度=4）	001（长度=4）	001（长度=4）	001（长度=4）	001（长度=4）
CTLx.DST_MSIZE	001（长度=4）	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（不支持单次传输），外设 B 代指 USART、SPI 或 I2C（支持单次传输）。

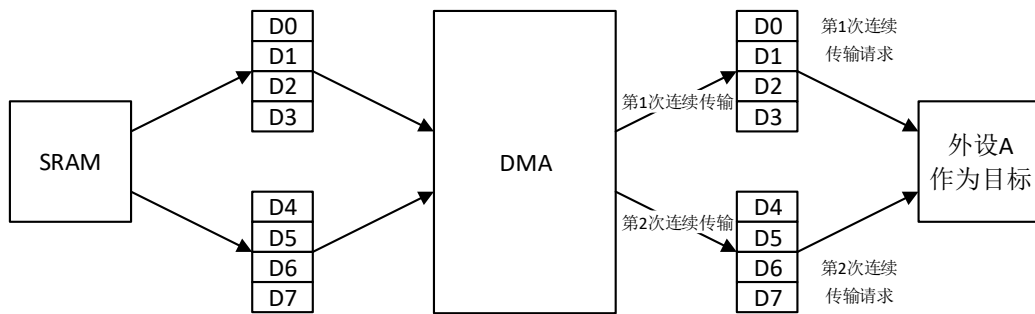


图 8-4 示例一：外设 A 配置 1 且作为目标，2 次源的传输，2 次目标的连续请求和传输

如示例一所示，由寄存器配置可知，以外设 A 为目标，SRAM 为源，传输的 BLOCK 数量为 8，源和目标的位宽为 32bit，一次连续传输的数据长度为 4。以 A 为目标时，由公式 $(CTLx.SRC_TR_WIDTH * CTLx.BLOCK_TS) / (CTLx.DST_TR_WIDTH * CTLx.DST_MSIZE) = n$ ，计算得 $n=2$ ，所以需要进行 2 次连续传输，每次连续传输搬运 4 个 word。

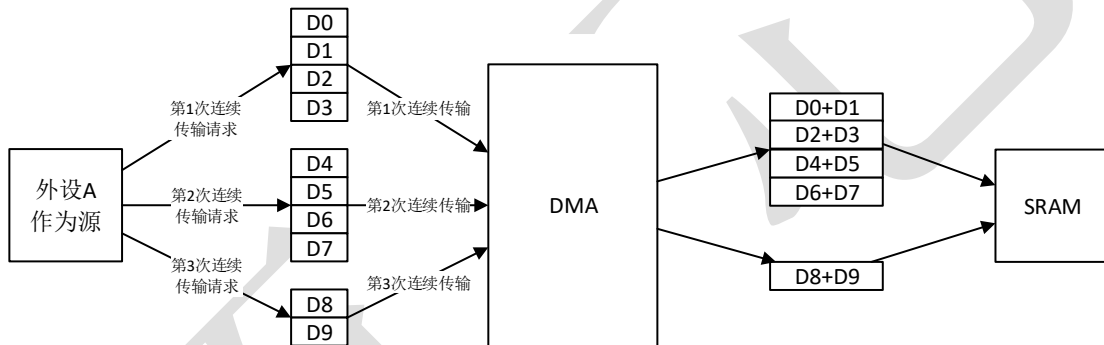


图 8-5 示例二：外设 A 配置 2 且作为源，3 次源的连续请求和传输，2 次目标的传输

如示例二所示，由寄存器配置可知，以外设 A 为源，SRAM 为目标，传输的 BLOCK 数量为 10，源的位宽为 16bit，目标的位宽为 32bit，一次连续传输的数据长度为 4。以 A 为源时，由公式 $(CTLx.BLOCK_TS) / (CTLx.SRC_MSIZE) = n$ ，计算得 $n=2$ ，所以需要先进行 2 次连续传输，次连续传输搬运数量为 4。由于数据未全部搬运完成，所以需要第三次传输搬运剩余数据，因为 A 只支持连续传输且剩余数据量不足 4，所以第三次传输为连续传输且仅搬运剩余数量的数据。由于目标位宽和源位宽不一致，所以目标的数据在整合后进行 2 次搬运即可完成。

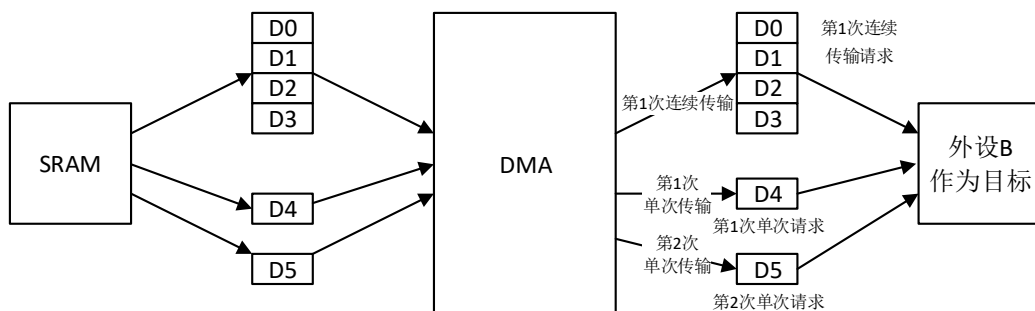


图 8-6 示例三：外设 B 配置 1 且作为目标，3 次源的传输，1 次目标的连续请求和传输，2 次目标的单次请求和传输

如示例三所示，由寄存器配置可知，以外设 B 为目标，SRAM 为源，传输的 BLOCK 数量为 6，源和目标的位宽为 32bit，一次连续传输的数据长度为 4。由公式 $(CTLx.SRC_TR_WIDTH * CTLx.BLOCK_TS) / (CTLx.DST_TR_WIDTH * CTLx.DST_MSIZE) = n$ 计算得 $n=1$ 。1 次连续传输后，由于数据未全部传输完成，所以需要再次请求传输剩余数据，因为 B 支持单次传输，剩余数据量为 2，所以 B 进行 2 次单次传输请求传输剩余数据。

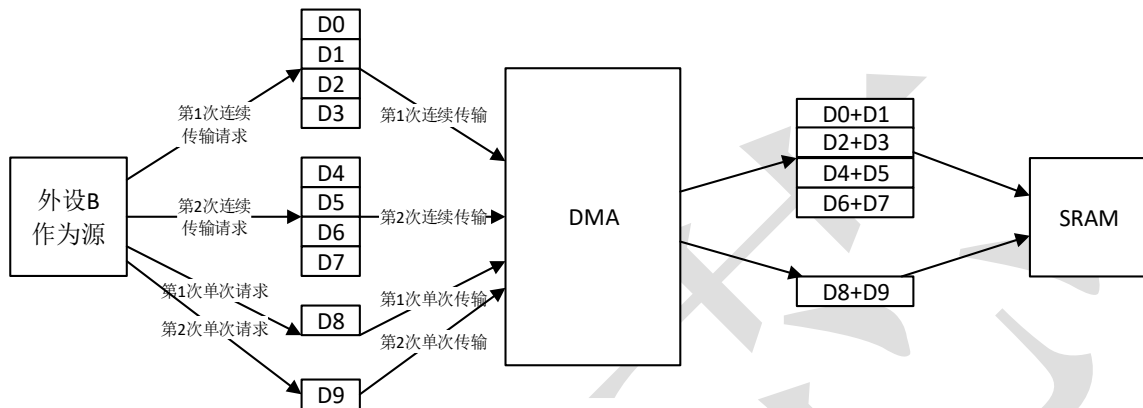


图 8-7 示例四：外设 B 配置 2 且作为源，2 次源的连续请求和传输，2 次源的单次请求和传输，2 次目标的传输

如示例四所示，由寄存器配置可知，以外设 B 为源，SRAM 为目标，传输的 BLOCK 数量为 10，源和目标的位宽为 32bit，一次连续传输的数据长度为 4。由公式 $(CTLx.BLOCK_TS) / (CTLx.SRC_MSIZE) = n$ 计算得 $n=2$ 。2 次连续传输后，由于数据未全部传输完成，所以需要再次请求传输剩余数据，因为 B 支持单次传输，剩余数据量为 2，所以 B 进行 2 次单次传输请求传输剩余数据。由于目标位宽和源位宽不一致，所以目标的数据在整合后进行 2 次传输即可完成。

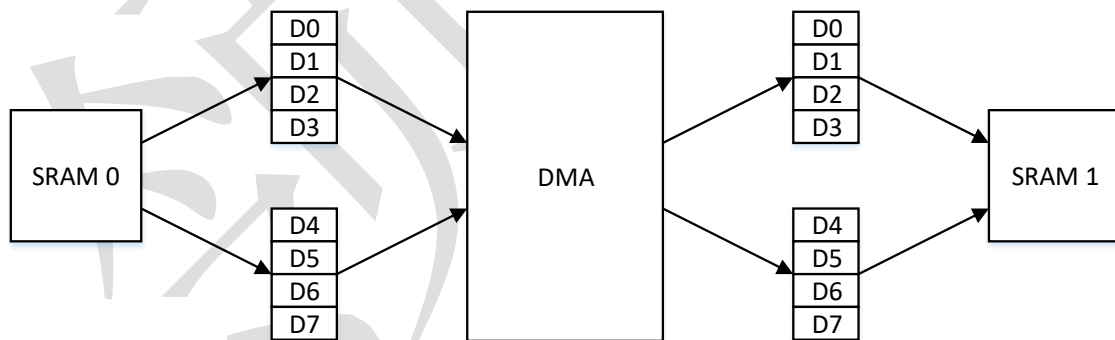


图 8-8 示例五：SRAM 0 作为源，SRAM 1 作为目标，2 次源的传输，2 次目标的传输

如示例五所示，由寄存器配置可知，以 SRAM 0 为源，SRAM 1 为目标，传输的 BLOCK 数量为 8，源和目标的位宽为 32bit，一次传输的数据长度为 4。由于存储器作为源和目标时不会发起传输请求，所以当寄存器配置完成后，两个 SRAM 之间就会根据寄存器的配置直接进行数据传输。

8.2.5 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。具体配置请参

考 DMA 互联配置寄存器 0 (SysCtrl_EDU_CFG0)。

DMA 的所有硬件接口的请求信号，经过 DMA 内部配置后分配给 2 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。

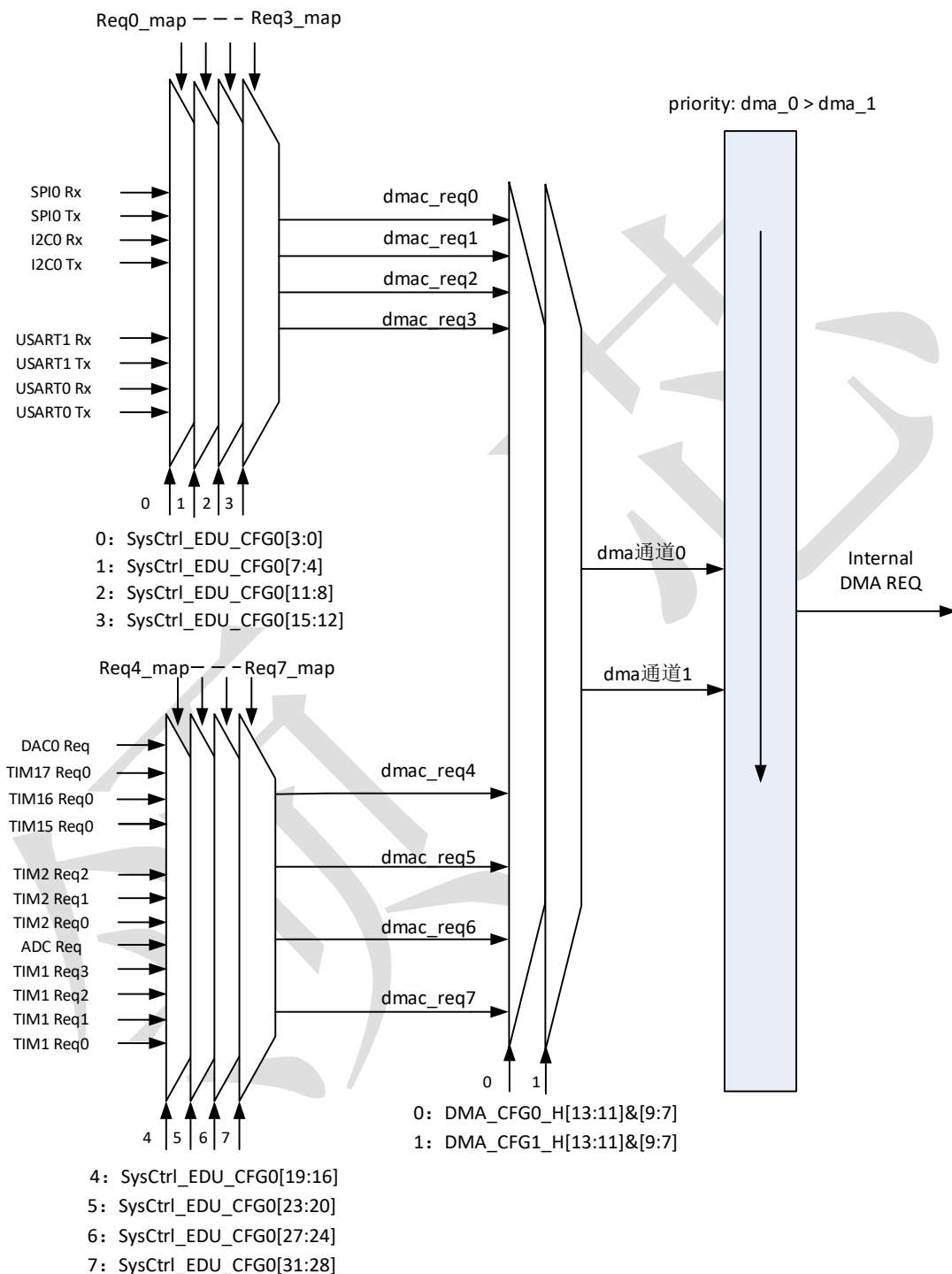


图 8-9 DMA 请求映射

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
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
软件触发寄存器				
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
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
DMA 互联寄存器				
SysCtrl_EDU_CFG0	DMA 互联配置寄存器	R/W	0x0000_0000	0x4800_7020

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[1:0]	Res.						CH_EN[1:0]
						w							rw

Bits	31:10	保留，必须保持复位值
Bits	9:8	CH_EN_WE[1:0] : 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	7:2	保留，必须保持复位值
Bits	1:0	CH_EN[1:0] : DMA 通道使能位
		0: 通道禁止（默认）
		1: 通道使能
		<i>注：当 DMA 使能位为 0 时，所有这些位都为 0。</i>

8.3.2 通道寄存器

每个 DMA 通道包含以下的通道寄存器（ $x = 0 \sim 1$ ）：

- 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（ $x=0/1$ ））

地址偏移：0x00/0x58

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（ $x=0/1$ ））

地址偏移：0x08/0x60

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_H (x=0/1)

地址偏移: 0x1C/0x74

复位值: 0x0000_0002

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.				BLOCK_TS[11:0]											
				rw											

Bits 31:12	保留, 必须保持复位值
Bits 11:0	BLOCK_TS[11:0]: 块传输大小 (通道 0/1 可配置有效位为[4:0])
	编程到 BLOCK_TS 中的数字表示为每个块传输执行的总传输数据量
	连续传输时的数据位宽由 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。

DMA_CTLx_L (x=0/1)

地址偏移: 0x18/0x70

复位值: 0x0030_4801

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

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

Bits	31:23	保留，必须保持复位值	
Bits	22:20	TT_FC: DMA 传输类型（复位值：3'b011）	
		CTLx.TT_FC	传输类型
		000	存储到存储
		001	存储到外设
		010	外设到存储
		011	外设到外设

Bits 19:17	保留，必须保持复位值				
Bits 16:14	SRC_MSIZ[2:0] : 源连续传输长度（同 DST_MSIZ 位）				
Bits 13:11	DST_MSIZ[2:0] : 目标连续传输长度				
	一次连续传输的数据量，其中每个数据的数据宽度为 CTLx.DST_TR_WIDTH (SRC_TR_WIDTH)				
	CTLx.SRC_MSIZ/ CTLx.DST_MSIZ	000	001	010	其他
	连续传输长度	1	4	8	保留
	例: BLOCK_TS =8, 代表总传输数据量为8, DST_MSIZ=1, 代表目标的一次连续传输的数据量为4, 需要两次目标连续传输完成全部数据的搬运				
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_H (x=0/1)

地址偏移: 0x44/0x9C

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_PER[2:0]			Res.	SRC_PER[2:0]			Res.						
		rw				rw									

Bits 31:14	保留，必须保持复位值				
Bits 13:11	DST_PER[2:0] : 目标外设选触发接口 0-7 中的 DMA 请求（同 SRC_PER 位）				
Bit 10	保留，必须保持复位值				
Bits 9:7	SRC_PER[2:0] : 源外设选择触发接口 0-7 中的 DMA 请求				
	如果 CFGx 为通道 x 的目的地分配一个硬件触发接口（0 - 7），则 HS_SEL_DST 字				

	段为 0；否则该字段将被忽略。通道可以通过指定的硬件触发接口与连接到该接口的目标（源）外设。
	注： 对于正确的 DMA 操作，同一个触发接口只能分配一个外设（源或目标）。
Bits 6:0	保留，必须保持复位值

DMA_CFGx_L (x=0/1)

地址偏移：0x40/0x98

复位值：0x0000_0C00

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.				HS_SEL_S	HS_SEL_D	Res.									
				RC	ST										
				rw	rw										

Bits 31:12	保留，必须保持复位值
Bit 11	HS_SEL_SRC： 源软件或硬件触发选择（同 HS_SEL_DST 位）
Bit 10	HS_SEL_DST： 目标外设软件或硬件触发接口选择
	选择某个触发接口（硬件或软件）对这个通道上的目标（源）请求是活跃的。
	0：硬件触发接口。忽略软件发起的连续传输请求
	1：软件触发接口，忽略硬件发起的连续传输请求（默认）
	如果目标（源）外设是存储器，则该 bit 被忽略。
Bits 9:0	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：

- 传输中断（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。
- 块传输中断（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。
- 目标传输中断（Destination Transaction Complete Interrupt），这个中断是在目标端的触发接口（硬件或软件触发接口）完成请求的单个/连续的最后一次传输之后产生的。
- 源传输中断（Source Transaction Complete Interrupt），这个中断是在源端触发接口（硬件或软件触发接口）完成请求的单个/连续的最后一次传输之后产生的。
- 错误中断（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。

与中断相关的寄存器（联系中断逻辑理解，中断逻辑参考章节 8.2.3 和图 8-3）：

- RawBlock、RawDstTran、RawErr、RawSrcTran、RawTfr（中断原始状态寄存器组）
- StatusBlock、StatusDstTran、StatusErr、StatusSrcTran、StatusTfr（中断状态寄存器组）
- MaskBlock、MaskDstTran、MaskErr、MaskSrcTran、MaskTfr（中断屏蔽寄存器组）
- ClearBlock、ClearDstTran、ClearErr、ClearSrcTran、ClearTfr（中断清除寄存器组）
- StatusInt（组合中断状态寄存器）

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 连续传输级别”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[1:0]	
														rw	

Bits	31:2	保留，必须保持复位值
Bits	1:0	RawTr[1:0]：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[1:0]	
														rw	

Bits	31:2	保留，必须保持复位值
Bits	1:0	RawBlock[1:0]：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[1:0]	
														rw	

Bits	31:2	保留，必须保持复位值
Bits	1:0	RawSrcTran[1:0] ：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[1:0]	
														rw	

Bits	31:2	保留，必须保持复位值
Bits	1:0	RawDstTran[1:0] ：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[1:0]	
														rw	

Bits	31:2	保留，必须保持复位值
Bits	1:0	RawErr[1:0] ：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[1:0]	
														rw	

Bits	31:2	保留, 必须保持复位值
Bits	1:0	StatusTfr[1:0]: 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[1:0]	
														rw	

Bits	31:2	保留, 必须保持复位值
Bits	1:0	StatusBlock[1:0]: 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[1:0]	
														rw	

Bits	31:2	保留, 必须保持复位值
Bits	1:0	StatusSrcTran[1:0]: 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[1:0]	
														rw	

Bits	31:2	保留, 必须保持复位值
Bits	1:0	StatusDstTran[1:0]: 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[1:0]	
														rw	

Bits	31:2	保留, 必须保持复位值
Bits	1:0	StatusErr[1:0]: 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[1:0]		Res.						INT_MASK_TFR[1:0]	
						w								rw	

Bits	31:10	保留, 必须保持复位值
Bits	9:8	INT_MASK_TFR_WE[1:0]: 传输中断屏蔽写使能

	0: 写禁止（默认）
	1: 写使能
Bits 7:2	保留，必须保持复位值
Bits 1:0	INT_MASK_TFR[1:0]: 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[1:0]		Res.						INT_MASK_BLOCK[1:0]	
						w								rw	

Bits 31:10	保留，必须保持复位值
Bits 9:8	INT_MASK_BLOCK_WE[1:0]: 块传输中断屏蔽写使能
	0: 写禁止（默认）
	1: 写使能
Bits 7:2	保留，必须保持复位值
Bits 1:0	INT_MASK_BLOCK[1:0]: 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[1:0]		Res.						INT_MASK_SRC_TRAN[1:0]	
						w								rw	

Bits 31:10	保留，必须保持复位值
Bits 9:8	INT_MASK_SRC_TRAN_WE[1:0]: 源传输中断屏蔽写使能
	0: 写禁止（默认）
	1: 写使能

Bits 7:2	保留，必须保持复位值
Bits 1:0	INT_MASK_SRC_TRAN[1:0] : 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[1:0]		Res.						INT_MASK_DST _TRAN[1:0]	
						w								rw	

Bits 31:10	保留，必须保持复位值
Bits 9:8	INT_MASK_DST_TRAN_WE[1:0] : 目标传输中断屏蔽写使能
	0: 写禁止（默认）
	1: 写使能
Bits 7:2	保留，必须保持复位值
Bits 1:0	INT_MASK_DST_TRAN[1:0] : 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[1:0]		Res.						INT_MASK_Err [1:0]	
						w								rw	

Bits 31:10	保留，必须保持复位值
Bits 9:8	INT_MASK_Err_WE[1:0] : 错误中断屏蔽写使能
	0: 写禁止（默认）
	1: 写使能
Bits 7:2	保留，必须保持复位值

Bits 1:0	INT_MASK_Err[1:0]: 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[1:0]	
														w	

Bits 31:2	保留，必须保持复位值
Bits 1:0	ClearTfr[1:0]: 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[1:0]	
														w	

Bits 31:2	保留，必须保持复位值
Bits 1:0	ClearBlock[1:0]: 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.															

8.3.3.5 组合中断状态寄存器（DMA_StatusInt）

地址偏移：0x360

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.											ERR	DSCT	SRCT	BLOCK	TFR
											r	r	r	r	r

Bits	31:5	保留，必须保持复位值
Bit	4	ERR : StatusErr 中断状态寄存器所有位的“或”组合
Bit	3	DSCT : StatusDst 中断状态寄存器所有位的“或”组合
Bit	2	SRCT : StatusSrcTran 中断状态寄存器所有位的“或”组合
Bit	1	BLOCK : StatusBlock 中断状态寄存器所有位的“或”组合
Bit	0	TFR : StatusTfr 中断状态寄存器所有位的“或”组合

8.3.3.6 软件触发寄存器组

组成软件触发寄存器的寄存器允许软件以触发接口信号在硬件中所做的相同方式发起单个或连续传输请求。软件触发寄存器如下：

- ReqDstReg – Destination Software Transaction Request Register，目标传输软件请求寄存器
- ReqSrcReg – Source Software Transaction Request Register，源传输软件请求寄存器寄存器
- SglReqDstReg – Single Destination Transaction Request Register，目标单次传输软件请求寄存器
- SglReqSrcReg – Single Source Transaction Request Register，源单次传输软件请求寄存器

设置 CFGx.HS_SEL_SRC 为 1 使能通道 x 源上的软件触发；设置 CFGx.HS_SEL_DST 为 1 使能通道 x 目的上的软件触发。

8.3.3.6.1 源传输软件请求寄存器寄存器（DMA_ReqSrcReg）

地址偏移：0x368

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_REQ_WE[1:0]		Res.						SRC_REQ[1:0]	
						w								rw	

Bits	31:10	保留，必须保持复位值
Bits	9:8	SRC_REQ_WE[1:0] : 源传输请求写使能

	0: 写禁止（默认）
	1: 写使能
Bits 7:2	保留，必须保持复位值
Bits 1:0	SRC_REQ[1:0]: 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[1:0]		Res.						DST_REQ[1:0]	
						w								rw	

Bits 31:10	保留，必须保持复位值
Bits 9:8	DST_REQ_WE[1:0]: 目标传输请求写使能
	0: 写禁止（默认）
	1: 写使能
Bits 7:2	保留，必须保持复位值
Bits 1:0	DST_REQ[1:0]: 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[1:0]		Res.						SRC_SGLREQ[1:0]	
						w								rw	

Bits 31:10	保留，必须保持复位值
Bits 9:8	SRC_SGLREQ_WE[1:0]: 源单次传输请求写使能
	0: 写禁止（默认）
	1: 写使能
Bits 7:2	保留，必须保持复位值

Bits 1:0	SRC_SGLREQ[1:0]: 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[1:0]		Res.						DST_SGLREQ[1:0]	
						w								rw	

Bits 31:10	保留，必须保持复位值
Bits 9:8	DST_SGLREQ_WE[1:0]: 目标单次传输请求写使能
	0: 写禁止（默认）
	1: 写使能
Bits 7:2	保留，必须保持复位值
Bits 1:0	DST_SGLREQ[1:0]: 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

8.3.4 DMA 互联配置寄存器（SysCtrl_EDU_CFG0）

SysCtrl_EDU_CFG0 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 0x05FA659A 到 SysCtrl_KEY 密钥寄存器，且在随后的 64 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。

地址：0x4800_7020

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 请求源通道 x 的触发源配置为 TIM1 DMA 请求 1（默认）
	0001: DMA 请求源通道 x 的触发源配置为 TIM1 DMA 请求 2
	0010: DMA 请求源通道 x 的触发源配置为 TIM1 DMA 请求 3

	0011: DMA 请求源通道 x 的触发源配置为 TIM1 DMA 请求 4
	0100: DMA 请求源通道 x 的触发源配置为 ADC
	0101: DMA 请求源通道 x 的触发源配置为 TIM2_CH1 / TIM2_TRIG
	0110: DMA 请求源通道 x 的触发源配置为 TIM2_CH2 / TIM2_CH3
	0111: DMA 请求源通道 x 的触发源配置为 TIM2_CH4 / TIM2_UP
	1000: 保留
	1001: DMA 请求源通道 x 的触发源配置为 TIM15
	1010: DMA 请求源通道 x 的触发源配置为 TIM16
	1011: DMA 请求源通道 x 的触发源配置为 TIM17
	1100: 保留
	1101: 保留
	1110: 保留
	1111: DMA 请求源通道 x 的触发源配置为 DAC
Bits 15:0	DMA_REQx_MAP[3:0]: DMA 请求源通道配置 (x=0 到 3)
	0000: DMA 请求源通道 x 的触发源配置为 USART0_TX (默认)
	0001: DMA 请求源通道 x 的触发源配置为 USART0_RX
	0010: DMA 请求源通道 x 的触发源配置为 USART1_TX
	0011: DMA 请求源通道 x 的触发源配置为 USART1_RX
	0100: DMA 请求源通道 x 的触发源配置为 I2C0_TX
	0101: DMA 请求源通道 x 的触发源配置为 I2C0_RX
	0110: DMA 请求源通道 x 的触发源配置为 SPI0_TX
	0111: DMA 请求源通道 x 的触发源配置为 SPI0_RX
	1000: 保留
	1101: 保留
	1010: 保留
	1011: 保留
	1100: 保留
	1101: 保留
	1110: 保留
	1111: 保留

8.4 DMA 参考编程流程

编程参考流程如下：

1. 开启DMA全局使能（仅首次初始化DMA时需要），读取通道使能寄存器，查看通道使能，如果通道使能开启，代表通道正在被使用，选取另一个未被使用的通道进行块传输；
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. 设置通道使能寄存器使能该通道；
6. 数据传输；
7. 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。

例：以 ADC 为源，位宽 32，连续传输长度为 4，通过 DMA 通道 0 向位宽为 32，连续传输长度为 4 的 SRAM 中传递 10 组数据。

1. 配置 `DmaCfgRe.DMA_EN = 1`（仅首次初始化 DMA 时需要），使能 DMA，读取 `ChEnReg.CH_EN[3:0]`，检查通道使能是否被开启，例：`ChEnReg.CH_EN[0] = 0`，代表通道 0 此时空闲，选取通道 0 进行块传输；
2. 检查 `DMA_CTL0_L.INT_EN`，确认通道 0 的中断使能是否开启，开启后通过配置 `ClearBlock`、`ClearDstTran`、`ClearErr`、`ClearSrcTran`、`ClearTfr` 寄存器清除通道 0 的中断标志位，读取 `RawBlock`、`RawDstTran`、`RawErr`、`RawSrcTran`、`RawTfr` 以及 `StatusBlock`、`StatusDstTran`、`StatusErr`、`StatusSrcTran`、`StatusTfr` 确认通道 0 的中断已被清除，块传输中断配置为不屏蔽；
3. 向 `DMA_SAR0[31:0]` 中写入 DMA 传输的源地址，向 `DMA_DAR0[31:0]` 中写入 DMA 传输的目标地址；
4. 配置 `DMA_CTL0_L` 及 `DMA_CTL0_H`，`BLOCK_TS=10`，`SRC_MSIZE` 与 `DST_MSIZE=1`，`SRC_TR_WIDTH` 与 `DST_TR_WIDTH=2`，`SINC` 与 `DINC=0`；配置 `DMA_CFG0_L` 及 `DMA_CFG0_H`，通过配置 `HS_SEL_DST` 与 `HS_SEL_SRC`，选择源和目标的触发方式为软件触发或硬件触发，由于存储器无法发出请求信号，所以用源使用两种触发时的配置为例：
硬件触发配置方式：参考章节 8.2.5 和图 8-9 配置 `SysCtrl_EDU_CFG0[31:16]` 和 `DMA_CFG0_H.SRC_PER` 来选择通道 0 响应 ADC 的硬件请求。
软件触发配置方式：前两次连续传输时通过配置 `DMA_ReqSrcReg[8]&[0]=1` 发出源传输请求，传输 8 组数据后，剩余两组数据。可以通过配置 `DMA_ReqSrcReg[8]&[0]=1` 发出源传输请求，连续传输完最后两组数据，也可以通过配置 `DMA_SglReqSrcReg[8]&[0]=1` 发出源单次传输请求，2 次单次传输完最后两组数据。
5. 配置 `ChEnReg[8]&[0]=1` 开启通道 0 的使能。
6. 结合 8.2.4 DMA 请求和数据传输中的示例；
7. 块传输完成后，`RawBlock.Raw_Tr=1`，通道 0 产生原始中断状态，参考图 8-3，从而使 `StatusBlock[0]= 1`，通道 0 产生中断状态，进而使 `DMA_StatusInt.BLOCK=1`，DMA 中断产生一次完整的 Block 传输完成。
8. 第一次块传输完成后，通道使能会被硬件关闭，在收到新的请求开启第二次 Block 传输前需要软件开启通道使能。

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	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 SysTick 定时器	0x0000_003C
0	-	-	-	-	0x0000_0040
1	8	可配	EXTI16 (LVD)	EXTI Line[16]（低压检测）中断	0x0000_0044
2	9	可配	EXTI17 (WT)	EXTI Line[17]（WT）中断	0x0000_0048
3	10	可配	Flash	Flash 全局中断	0x0000_004C
4	11	可配	RCC	RCC 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	EXTI18_19 (ACMP0/1)	EXTI Line[19:18]（ACMP0/1 输出）中断	0x0000_0060
9	16	可配	DMA_CH0	DMA 通道 0 中断	0x0000_0064
10	17	可配	DMA_CH1	DMA 通道 1 中断	0x0000_0068
11	18	可配	SRAM	SRAM 全局中断	0x0000_006C
12	19	可配	ADC	ADC 全局中断	0x0000_0070
13	20	可配	TIM1_NON_CC	TIM1 非比较中断	0x0000_0074
14	21	可配	TIM1_CC	TIM1 比较中断	0x0000_0078
15	22	可配	TIM2	TIM2 全局中断	0x0000_007C
16	23	可配	EXTI22_24	EXTI Line[24:22]（USART0/1 的低功耗唤醒	0x0000_0080

			(LP_USART)	醒) 中断	
17	24	可配	DAC	DAC 全局中断	0x0000_0084
18	25	可配	EXTI20_21 (ACMP0/1)	EXTI Line[21:20] (ACMP0/1) 中断	0x0000_0088
19	-	-	-	-	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	-	-	-	-	0x0000_00A0
25	32	可配	SPI0	SPI0 全局中断	0x0000_00A4
26	-	-	-	-	0x0000_00A8
27	34	可配	USART0	USART0 全局中断	0x0000_00AC
28	35	可配	USART1	USART1 全局中断	0x0000_00B0
29	-	-	-	-	0x0000_00B4
30	-	-	-	-	0x0000_00B8
31	38	可配	DIV/CORDIC	DIV/CORDIC 全局中断	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 24 个外部/内部事件线（多达 25 个通用 I/O 口连接到 16 个外部中断线和 8 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 APB0 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8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PCLK0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

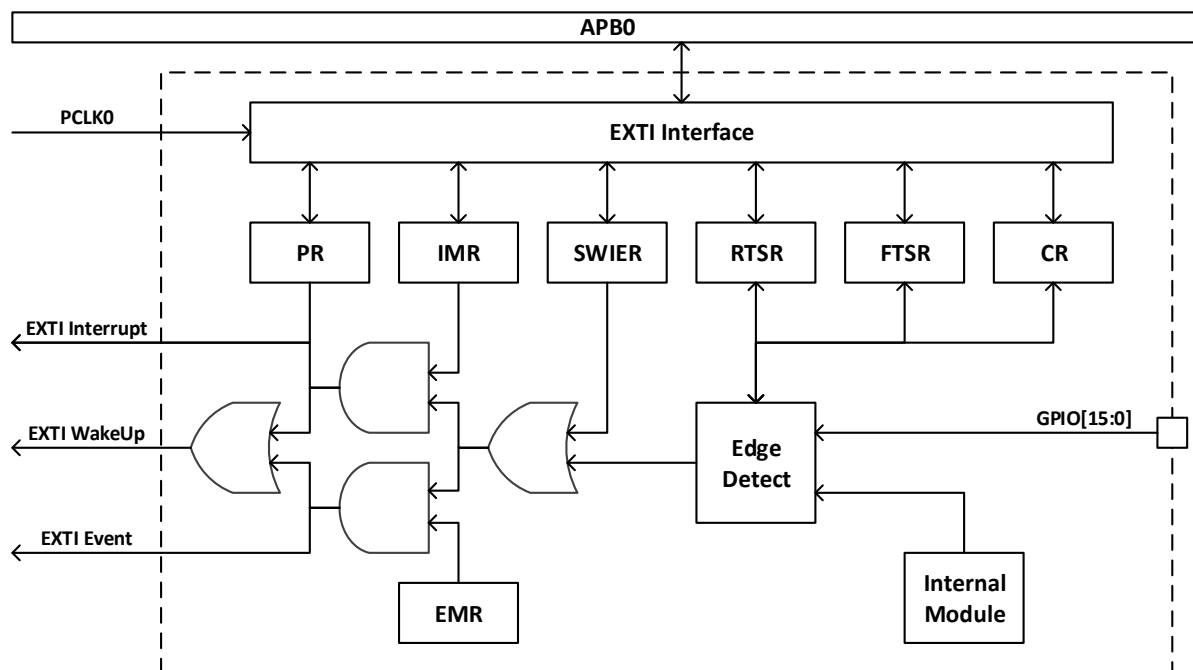


图 9-1 外部中断/事件控制器框图

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 pending 位（在 NVIC 中断清除 pending 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 pending 位没有设置，因此不需要清除 EXTI.PR 或 NVIC IRQ 通道 pending 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：

- 配置所选中断线的 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_RTSTR、EXTI_FTSTR 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：

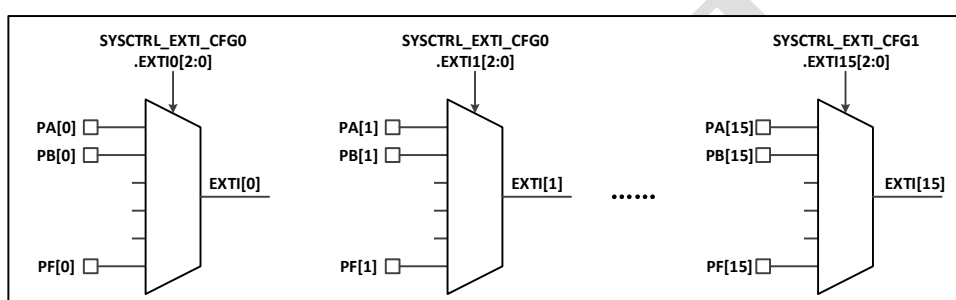


图 9-2 外部中断通用 I/O 映射

其余 EXTIx（x = 16~23）按如下方式连接：

- EXTI 线 16 连接到 LVD 输出
- EXTI 线 17 连接到 WT 唤醒事件
- EXTI 线 18 连接到模拟比较器 ACMP0 的输出事件
- EXTI 线 19 连接到模拟比较器 ACMP1 的输出事件
- EXTI 线 20 连接到模拟比较器 ACMP0 的序列比较结束中断
- EXTI 线 21 连接到模拟比较器 ACMP1 的序列比较结束中断
- EXTI 线 22 连接到 USART0 低功耗唤醒事件
- EXTI 线 23 连接到 USART1 低功耗唤醒事件

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

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	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[23:0]															
rw															

Bits 31:24	保留，必须保持复位值
Bits 23:0	IMx：线 x 上的中断屏蔽（x = 0~23）
	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[23:0]															
rw															

Bits 31:24	保留，必须保持复位值
Bits 23:0	EMx：线 x 上的事件屏蔽（x = 0~23）
	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[23:0]															
rw															

Bits 31:24	保留，必须保持复位值
Bits 23:0	RTx : 线 x 上的上升沿触发事件配置位 (x = 0~23)
	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_FTSR)

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[23:0]															
rw															

Bits 31:24	保留，必须保持复位值
Bits 23:0	FTx : 线 x 上的下降沿触发事件配置位 (x = 0~23)
	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 : 同一根输入线即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[23:0]															
rw															

Bits 31:24	保留，必须保持复位值
Bits 23:0	SWIERx : 线 x 上的软件中断和事件 (x = 0~23)
	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[23:0]															
rw															

Bits 31:24	保留，必须保持复位值
Bits 23:0	PRx : 线 x 上的挂起位 (x = 0~23)
	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[23:0]															
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	27:24	保留, 必须保持复位值
Bits	23:0	LRx: 线 x (x = 0~23) 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



	其它：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
	其它：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）

LCM32F06X 内置有 1 个 12 位模拟/数字转换器（ADC），一共有 16 个外部通道和 1 个内部通道（内部电压通道）；包含 2 路独立的采样/保持电路；最高转换速率达 2MSPS。DAC 和 OPA 的输出也可以通过 I/O 端口送给 ADC 的输入通道。

10.1 ADC 主要特性

- 转换速率达 2 MSPS
- 精度为 12-bit
- ADC 支持使用内部参考电压 3.2V（芯片供电 $VDDA > 2.2V$ ）、4.9V（芯片供电 $VDDA > 4V$ ）或者外部参考电压 $VDDA$
- ADC 可以实现 4 组硬件触发的序列和任意可配的软件触发
- ADC 内置有一个温度通道，温度传感器产生一个随温度线性变化的电压，在内部连接到温度通道上
- ADC 内置有一个电压通道，可以选择一个内部电压，通过 PMU_AOUT ，在内部连接到 ADC 电压通道上
- ADC 内置两个模拟看门狗（AWD）。模拟看门狗功能允许监视任意路选中的通道，当被监视的信号超出预置的阈值范围时，将产生标志和中断
- 定时器（TIM）、模拟比较器（ACMP）和外部 I/O 可以通过各种逻辑组合硬件触发 ADC
- ADC 工作时钟为 APB0 总线时钟（PCLK0）的分频，或者 PLL 时钟的分频
- ADC 支持 DMA 操作，可配置不同搬运策略
- 内部含有 2 路采样/保持电路，支持单/双采样，可编程采样、转换时间

10.2 ADC 功能描述

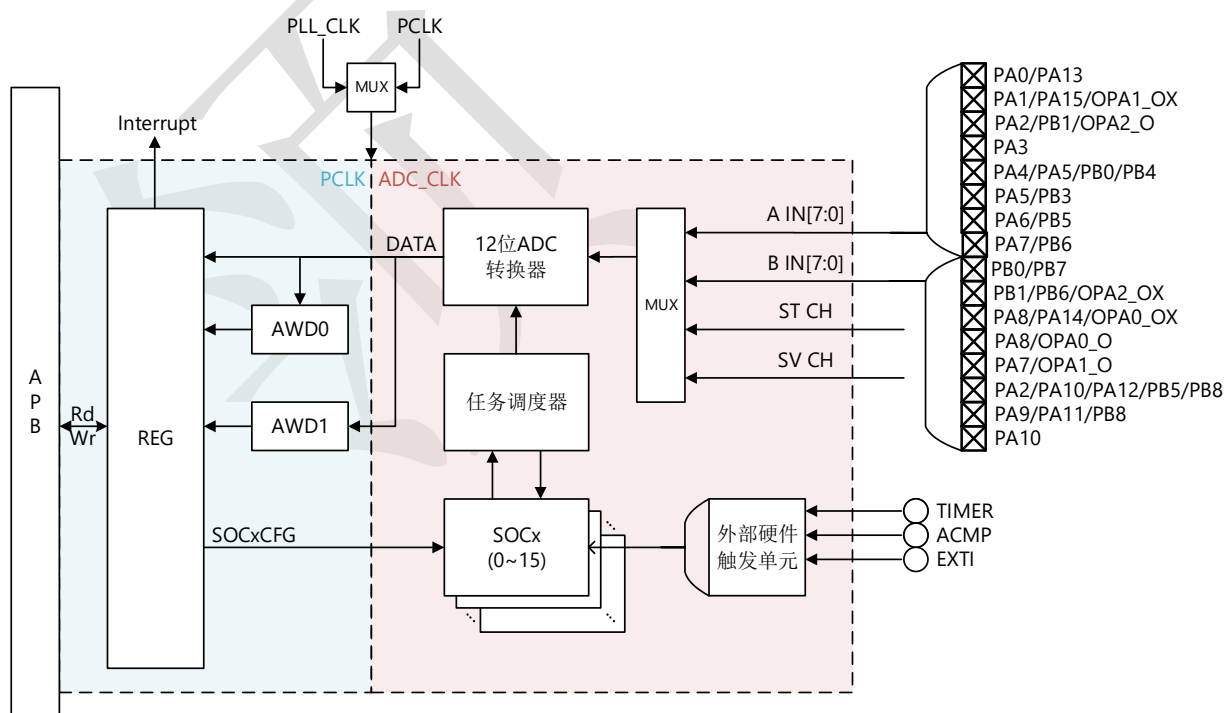


图 10-1 ADC 框图

10.2.1 时钟

ADC 具有双时钟域架构，ADC 时钟（ADC_CLK）独立于 APB 时钟（PCLK0）。ADC_CLK 是 ADC 模块工作时使用的时钟；PCLK 是处理器配置 ADC 寄存器时使用的时钟。ADC_CLK 可由两个时钟源分频产生，具体参见寄存器 ADC_CFGR.CKMODE 和 ADC_CFGR.CKMODE[3:0]。

- PLL 时钟，与 APB 时钟（PCLK0）异步；这种配置的优势在于无论系统频率高低，ADC 都能工作在最佳频率。
- APB 时钟（PCLK）避免了跨时钟域问题，在 ADC 由定时器触发或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）。

10.2.2 开关控制（EN、DIS）

默认情况下，ADC 模块处于掉电状态（EN=0），如图 10-2 所示。

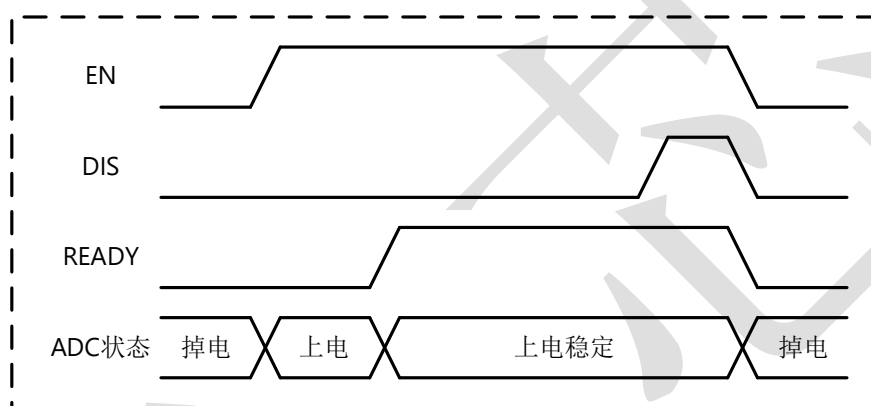


图 10-2 ADC 启动与关闭时序图

两个控制位用于开启或关闭 ADC：

- 设置寄存器 ADC_CR. EN=1 开启 ADC。当 ADC 模块上电稳定后，寄存器 ADC_SR.ADRDY 状态位由硬件置 1
- 设置寄存器 ADC_CR. DIS=1 关闭 ADC，并使 ADC 进入掉电状态。当 ADC 模块掉电后，硬件自动清除 ADC_CR.EN、ADC_CR.DIS、ADC_SR.ADRDY 位

以下为 ADC 开启过程：

1. 配置 ADC_TRIM 寄存器，加载 TRIM 值（如果没有 TRIM 值，请加载一遍复位值）；
2. 配置 ADC_ANA 寄存器中的 PD_ADC、PD_REF、PD_SHA、PD_SHB = 0（默认为 1）；
3. 配置 ADC_CR.EN = 1，发起 ADC 使能命令，启动 ADC 模块上电；
4. 等待 ADC_SR.READY 硬件置 1，表示 ADC 模块已经上电且稳定；
5. ADC 启动完成，可以开始工作进行转换数据。

以下为 ADC 关闭过程：

1. 检查 ADC_CR.START 是否为 0，以确保 ADC 不在转换过程中。
2. 如果 ADC_CR.START = 1，可以通过配置 ADC_CR.STOP = 1，停止 ADC 模块工作；
3. 如果 ADC_CR.START = 0，表示 ADC 已经停止工作，可以；
4. 配置 ADC_CR.DIS = 1，发起 ADC 关闭命令；
5. 等待 ADC_CR.EN 或 ADC_CR.DIS = 0，表示 ADC 模块完全关闭且掉电。

10.2.3 工作控制（START、STOP）

START 设置：

软件配置 ADC_CR.START 位，发起 START 命令，使 ADC 从非工作状态进入工作状态。进入工作状态后，ADC 可以接收触发。在接收到触发后，ADC 进入采样转换的工作状态，并且 ADC_SR.BUSY 位由硬件置 1，表示当前 ADC 忙状态。

STOP 设置：

软件配置 ADC_CR.STOP 位，发起 STOP 命令，停止或强制停止 ADC 工作。在完成当前的 SOC 采样转换后停止 ADC，使其进入非工作状态。随后 ADC_CR.STOP、ADC_CR.START、ADC_SR.BUSY 位都由硬件清 0。

在两种工作状态下，发起 STOP 命令后执行有所不同：

ADC_SR.BUSY = 0 时，ADC 并未采样转换，会立即响应 STOP 命令。

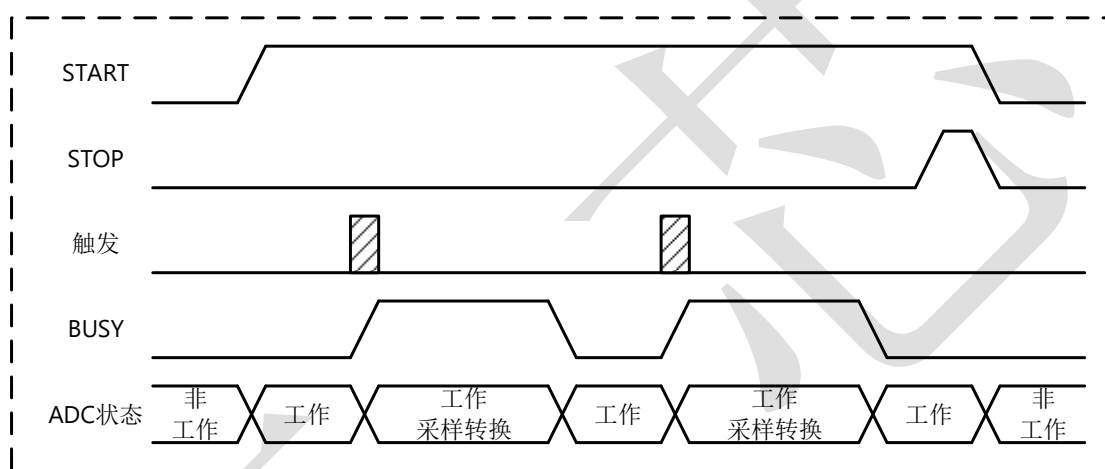


图 10-3 START 和 STOP 工作时序图

ADC_SR.BUSY = 1 时，ADC 处于采样转换，如图 10-4 所示，会在完成 SOC[8]的采样转换后才会响应 STOP 命令，ADC 进入到非工作状态。

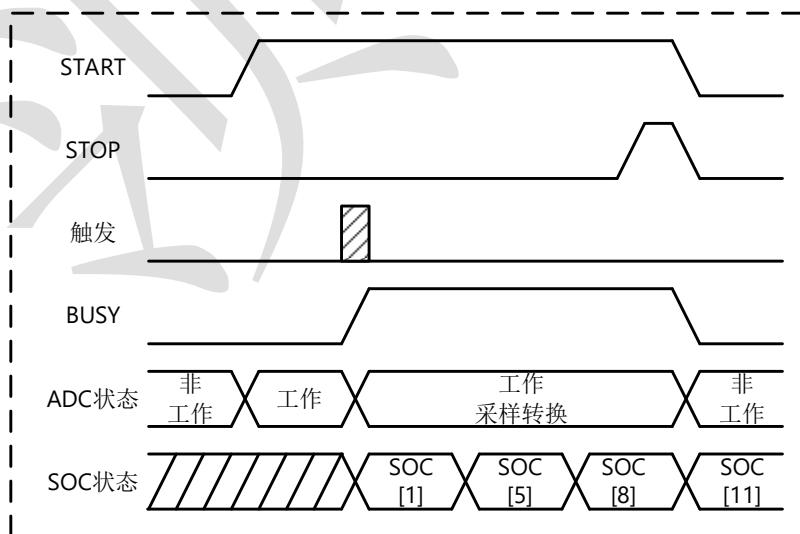


图 10-4 START 和强制 STOP 工作时序图

10.2.4 通道

ADC 拥有 16 个外部通道和 2 个内部通道。

10.2.4.1 外部通道

ADC 有 16 个外部通道，分别对应 ADCIN[15:0]。每一个外部通道连接到不同的 I/O 上。

ADC 和其他模拟模块联动有两种连接方式：

- ADC 的输入端和其他模拟模块输出端处于同一个 GPIO 的同一个模拟通道上，则属于直连。模拟信号从其他模拟模块直接连接到 ADC，不经过 GPIO，此时 GPIO 可以用作输入/输出/数字复用功能。
- ADC 的输入端和其他模拟模块输出端处于在同一个 GPIO 的不同模拟通道上，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 所有外部通道与相关 I/O 关系具体说明如下表：

表 10-1 ADC 外部 I/O 通道表

A 组 通道	I/O 序号	GPIO 模拟通道		B 组 通道	I/O 序号	GPIO 模拟通道	
		YA1	YA2			YA1	YA2
ADCIN [0]	PA0	ADCIN[0] ACMP0_P4 ACMP0_N0	OPA2_P1	ADCIN [8]	PB0	ADCIN[4] ACMP0_P0 ACMP1_P1	ADCIN[8] OPA1_N0 ACMP1_P3 ACMP1_N1 ACMP0_N4
	PA13		DACOUT_BUF		PB7	ADCIN[8] OPA1_N0 ACMP1_P3 ACMP1_N1 ACMP0_N4	-
ADCIN [1]	PA1	ADCIN[1] ACMP0_P5 ACMP0_N1 OPA1_OX	OPA2_N0 OPA0_N0 ACMP1_N3	ADCIN [9]	PB1	ADCIN[9] ACMP1_P0 ACMP0_N3 OPA2_OX	ADCIN[2] OPA2_O ACMP1_P5
	PA15		PMU_AOUT		PB6	ADCIN[7] OPA1_P0 ACMP0_P2	ADCIN[9] ACMP1_P0 ACMP0_N3 OPA2_OX
ADCIN [2]	PA2	ADCIN[13]	ADCIN[2] OPA2_O ACMP1_P5	ADCIN [10]	PA8	ADCIN[10] OPA0_OX ACMP0_P3 ELVI	ADCIN[11] OPA0_O ACMP1_P7
	PB1	ADCIN[9] ACMP1_P0 ACMP0_N3 OPA2_OX			PA14		PMU_AOUT

ADCIN [3]	PA3	ADCIN[3] ACMP0_N2 ACMP0_P6 ACMP1_N0	OPA0_P1	ADCIN [11]	PA8	ADCIN[10] OPA0_OX ACMP0_P3 ELVI	ADCIN[11] OPA0_O ACMP1_P7	
ADCIN [4]	PA4	ADCIN[4] ACMP0_P0 ACMP1_P1	OPA0_N0 OPA2_N0 ACMP1_N3	ADCIN [12]	PA7	ADCIN[7] OPA1_P1 ACMP0_P2	ADCIN[12] OPA1_O ACMP1_P6	
	PA5		ADCIN[5] OPA2_P0 ACMP1_P2					
	PB0		ADCIN[8] OPA1_N0 ACMP1_P3 ACMP1_N1 ACMP0_N4					
	PB4		-					
ADCIN [5]	PA5	ADCIN[4] ACMP0_P0 ACMP1_P1	ADCIN[5] OPA2_P0 ACMP1_P2	ADCIN [13]	PA2	ADCIN[13]	ADCIN[2] OPA2_O ACMP1_P5	
					PB8		ADCIN[14]	
	PB3	ADCIN[5] OPA2_P0 ACMP1_P2	-		PA12	OPA0_N1 OPA1_N1 OPA2_N1 ACMP1_P4	ADCIN[13] OSCL_IN	
					PB5	ADCIN[6] ACMP0_P1 ACMP1_N4	ADCIN[13]	
ADCIN [6]	PA6	ADCIN[6] ACMP0_P1 ACMP1_N4	ACMP1_P4 OPA0_N1 OPA1_N1 OPA2_N1	ADCIN [14]	PA9	OPA0_N0 OPA2_N0 ACMP1_N3	ADCIN[14]	
	PB5		ADCIN[13]		PA11	OPA1_P1 ACMP1_N2 ACMP0_P7	ADCIN[14] OSCL_OUT	
					PB8	ADCIN[13]	ADCIN[14]	
ADCIN [7]	PA7	ADCIN[7] OPA1_P0 ACMP0_P2	ADCIN[12] OPA1_O ACMP1_P6	ADCIN [15]	PA10	ADCIN[15] OPA0_P0	DACOUT_BUF	
	PB6		ADCIN[9] ACMP1_P0					



			ACMP0_N3 OPA2_OX				
--	--	--	---------------------	--	--	--	--

10.2.4.2 内部电压通道（SV）

ADC 模块配置了一个内部电压通道，连接在芯片内部 PMU_AOUT 信号上，需要检测的电压请参考第 4 章中的电源配置寄存器 PWR_CFG.AOUT_SEL。这个内部通道直接与 ADC 相连接，可以将电压通过 ADC 进行转换。配置 ADC_CFGR.SV 来启用内部电压监控。当启用了这个模式后，ADC_CFGR.SVS=0 时，会自动把内部电压通道映射到 SOC[15]上；ADC_CFGR.SVS=1 时，会自动把内部电压通道映射到 SOC[7]上。启用内部电压通道的监控模式后，被映射的 SOC 将忽略原有外部通道的信息，专用于内部电压通道。

10.2.4.3 内部温度通道（ST）

ADC 模块配置了一个内部温度传感器，可以用来测量器件的结点温度。内部温度传感器直接与 ADC 相连接，可以将测量到的温度转化为电压值，然后由 ADC 进行数字化转换。配置 ADC_CFGR.ST 来启用内部温度监控。当启用了这个模式后，ADC_CFGR.STS=0 时，会自动把内部温度通道映射到 SOC[15]上；ADC_CFGR.STS=1 时，会自动把内部温度通道映射到 SOC[7]上。启用内部温度通道的监控模式后，被映射的 SOC 将忽略原有外部通道的信息，专用于内部温度通道。

芯片出厂时会在不同温度下进行 VTS 采样，采样的 AD 值会存入 Flash 对应地址：

- 地址 0x1FFFF7E0 为常温下以 3.2V 和 4.9V 做参考的 VTS 采样 AD 值，其中低 16 位为 3.2V 做参考时的 AD 采样值，高 16 位为 4.9V 做参考时的 AD 采样值。
- 地址 0x1FFFF7E4 为高温下以 3.2V 和 4.9V 做参考的 VTS 采样 AD 值，其中低 16 位为 3.2V 做参考时的 AD 采样值，高 16 位为 4.9V 做参考时的 AD 采样值。

此外，VTS 对应的采样温度会转换成 16 进制数字，并存入 Flash 对应地址：地址 0x1FFFF780 为常温的温度标定值；地址 0x1FFFF784 为高温的温度标定值。

图 10-5 为 ADC 参考源为内部 3.2V 情况下 VTS 曲线图，横坐标是温度值，纵坐标是采样的十进制 AD 值。

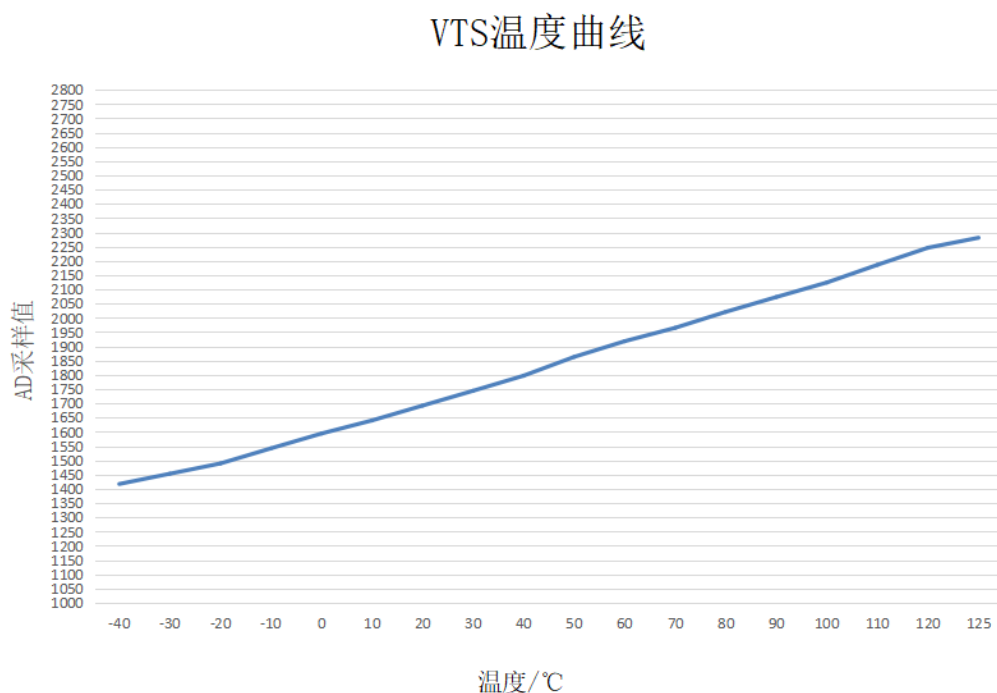


图 10-5 VTS 温度曲线图

实测发现 VTS 采样温度与 AD 值为线性关系；这里给出采样温度和 AD 值的计算公式，用于计算 VTS 实时温度：

$$\text{斜率 } K = (ThA - TnA) * 100 / (Th - Tn);$$

$$\text{当前温度 } T = (C - ThA) * 100 / K + Th;$$

此处 C 为 VTS 实时采样值，Th 表示高温的温度标定值，Tn 表示常温的温度标定值，ThA 表示高温下的 VTS 采样 AD 值（根据参考源选择对应的高 16 位或低 16 位），TnA 表示常温下的 VTS 采样 AD 值（根据参考源选择对应的高 16 位或低 16 位）。

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。

10.2.5 可编程采样转换周期

- 在 ADC 开始高精度转换前需要一个稳定时间 T_{STAB} ，在此期间会发送一系列调整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。在上电稳定后，`ADC_SR.READY` 由硬件置 1，可以发起 `START` 命令让 ADC 模块进入工作状态。
- ADC 在上述阶段内的调整脉冲，周期为 $T_{AP} = (ADC_SMPR.APW[2:0] + 1) * 2 * ADC_CLK$ 。

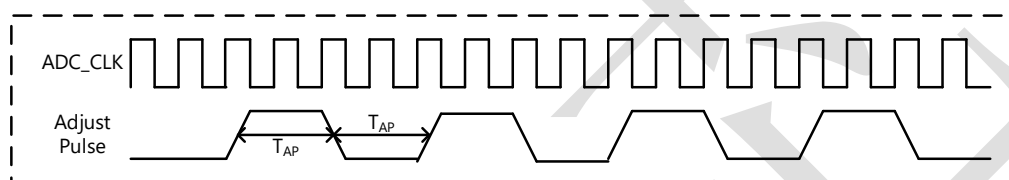


图 10-6 调整脉冲时序图

温度通道的采样需要额外的采样时间，建议采样时间为 $1\mu s$ 。在 $24MHz$ 工作频率下，需要配置寄存器 `ADC_SMPR.STSPW=01`。如果 ADC 工作频率降低，可以配置更短的时间周期。

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足够长的直接采样脉冲，以便输入电压源对内嵌电容充电到输入电压的水平。可编程采样脉冲（Sample Pulse）根据输入电压的阻抗来调整转换速度。完成采样后，采样保持（Hold）电路会保持该电压值，等待被转换。ADC 模块接收到转换脉冲（Convert Pulse）后才能开始将采样的电压信息转换为数值。ADC 模块的状态和可编程采样转换周期信号的时序图如下。

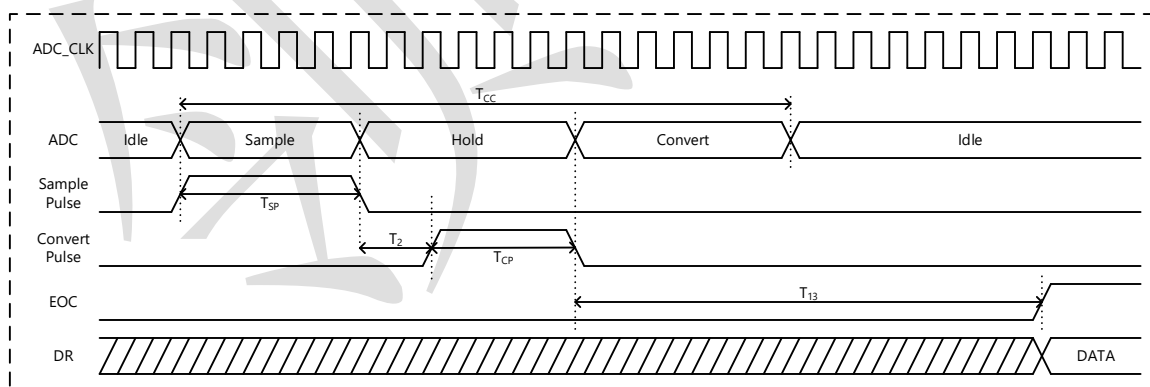


图 10-7 单次采样转换时序图

- 其中采样脉冲的宽度 $T_{SP} = (ADC_SMPR.SPW[3:0] + 1) * ADC_CLK$ ；ADC 模块在 Hold 阶段中，有一个 $2 * ADC_CLK$ 的固定保持时间；转换脉冲（Convert Pulse）的宽度 $T_{CP} = (ADC_SMPR.CPW[1:0] + 1) * ADC_CLK$ ；采样转换周期（Conversion Cycle） $T_{CC} = (ADC_SMPR.CC[3:0] + 13) * ADC_CLK + T_{CP}$ 。

当转换脉冲的下降沿到来时，转换开始，13 个 `ADC_CLK` 后，ADC 的转换过程完成。这个过程结束时，ADC 会生成一个数据，并同时设置对应的结束转换位（EOC，End of Conversion），EOC 置位表示该数据已

经被成功转换。

如图 10-8 所示，对于单次采样转换，从采样到转换数据产生的耗时为 $T_{SP} + T_{CP} + 15 * ADC_CLK$ ；对于连续采样转换，下一次采样可以在前一次采样转换周期结束后立即开始，形成类似于流水线的执行，实际从采样到转换数据产生的耗时为 T_{CC} ，最小为 14 个 ADC_CLK 。

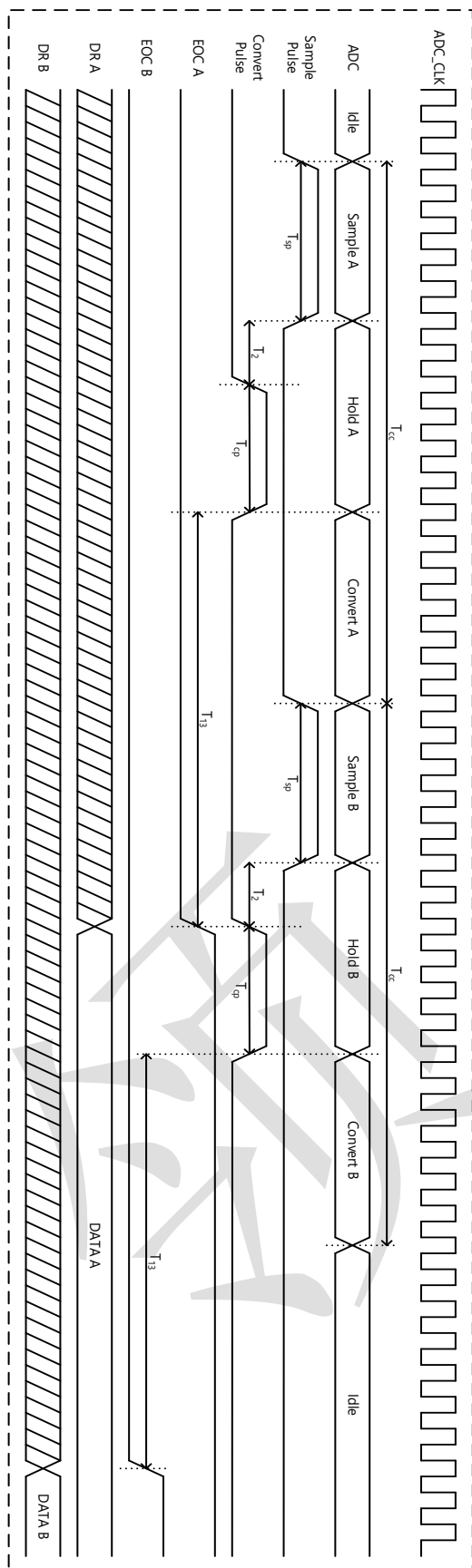


图 10-8 连续采样转换时序图

10.2.6 单次采样转换（SOC）

单次采样转换，Single of Convert（简称为 SOC，SOC[x]为具体的某一个单次采样转换）。ADC 的采样转换就是由多个 SOC 所构成的。每一个 SOC 都是独立可配置的，它们可以灵活的构成任意序列，满足各种复杂需求。

ADC 中一共包含 16 个可配置 SOC，其中 SOC[7]和 SOC[15]除了绑定外部 I/O 通道外，还可以绑定到内部温度 ST 或内部电压 SV 通道上。

10.2.6.1 SOC 优先级

每个 SOC 都有一个固定优先级，依据 SOC 序号从小到大。在组成序列的时候，按照优先级自动排序。优先级决定同时被触发挂起的 SOC 进行序列排序，不同触发的 SOC 之间也通过优先级的顺序进行抢占。

当一个硬件触发信号挂起多个 SOC 时，SOC 会依据优先级从小到大的顺序排序并组成一段序列。ADC 模块会按照序列的顺序依次采样转换这些 SOC。

当多个硬件触发信号挂起多个 SOC 时，情况与一个硬件触发相同。多个硬件序列会组合成一个序列。

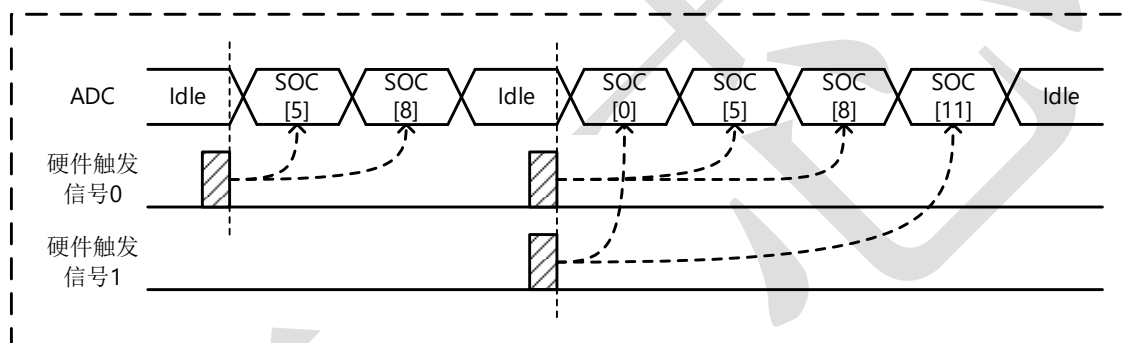


图 10-9 SOC 优先级排序的时序

当 ADC 模块正在采样转换工作中，有新的硬件触发信号。新的序列会依据优先级从小到大的插入到序列中。其中，当前正在采样转换的 SOC 不受影响。

可以通过分配不同硬件序列的优先级，实现序列之间的分离、抢占、注入等功能。

- 分离：一段序列内的优先级连续，可以保证序列与其他序列分离。
- 抢占：分配高优先级的 SOC 给序列，可以使序列在下次就进行采样转换。
- 注入：分配合适的中等优先级，可以将一段序列插入到另一段序列的固定位置中。

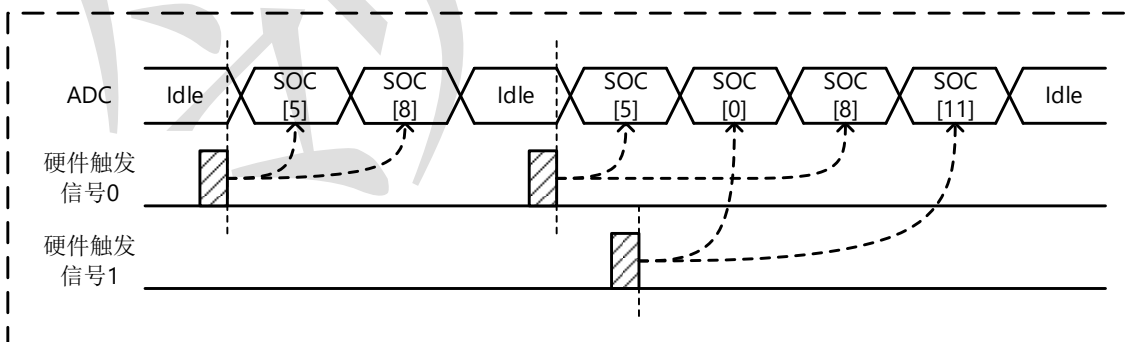


图 10-10 SOC 优先级抢占的时序图

10.2.6.2 SOC 配置

SOC 的相关配置都在 SOC[x]CFG 寄存器中，每个 SOC[x]CFG 寄存器都包含 TRIGSEL、CHSEL、CON、FP 共 4 个部分。

TRIGSEL 是 SOC 绑定硬件触发的配置位。默认值为 3'b111，即没有触发。不同的 SOC[x]可以绑定相同的硬件触发，绑定在同一个硬件触发的所有 SOC 会自动分为一组，并按照序号从小到大排序。

CHSEL 是 SOC 绑定外部 IO 通道的配置位。不同的 SOC[x]可以绑定相同的 IO 通道，这样可以通过不同的触发在不同的时间点上采样同一个外部 IO 通道上的电压。

CON 是 SOC 的双采样的配置位。一组 SOC 的 CON 配置位启用后，该组 SOC 会变为双采样模式。

FP 是强制挂起位，当 POF 事件发生时，FP = 1 的 SOC 依旧会被硬件触发，而 FP = 0 的 SOC 则会忽略。

这里举例说明，配置 SOC[0]、SOC[5]、SOC[8]和 SOC[11]为硬件触发 0，配置 SOC[3]和 SOC[4]为硬件触发 1，则 SOC[0]、SOC[5]、SOC[8]和 SOC[11]形成一组序列，而 SOC[3]和 SOC[4]形成另一组序列。SOC[0]选择 ADCIN[7]，SOC[5]选择 ADCIN[5]，SOC[8]选择 ADCIN[3]，SOC[11]选择 ADCIN[5]，SOC[3]选择 ADCIN[3]，SOC[4]选择 ADCIN[5]。ADC 先后接收到触发 0 和触发 1 的工作示意图如下，在触发 0 到来时 SOC[0]、SOC[5]、SOC[8]和 SOC[11]组成的序列进行连续采样转换，在触发 1 到来时 SOC[3]和 SOC[4]组成的序列进行连续采样转换。

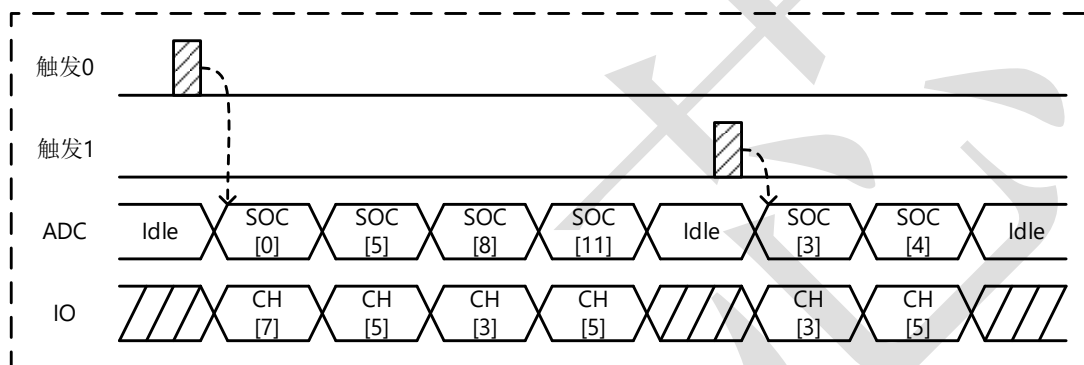


图 10-11 SOC 和 ADC 通道的触发关系图

10.2.6.3 SOC 状态

每个 SOC 都有三个状态，分别是空闲状态、采样转换状态和挂起状态。当 SOC 未被触发时，处于空闲状态。SOC 在接收到触发后，变为挂起状态，等待被采样转换。所有挂起的 SOC 中，优先级最高的 SOC 进入采样转换状态，完成数据的采样转换后变回空闲状态。

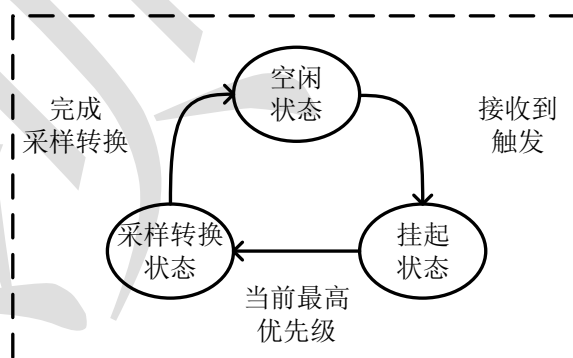


图 10-12 SOC 状态跳转图

其中 SOC[0]是触发 0 触发的所有 SOC 中优先级最高的，所以它在进入挂起状态后立马又进入到了采样转换状态，可以看作直接从空闲状态进入到采样转换状态。当 SOC[0]完成了采样转换后，下一个从挂起状态进入到采样转换状态的 SOC 就是剩余所有挂起 SOC 中优先级最高的 SOC[5]。依次类推，接下来就是 SOC[8]、SOC[11]。最后完成了所有 SOC 的采样转换。

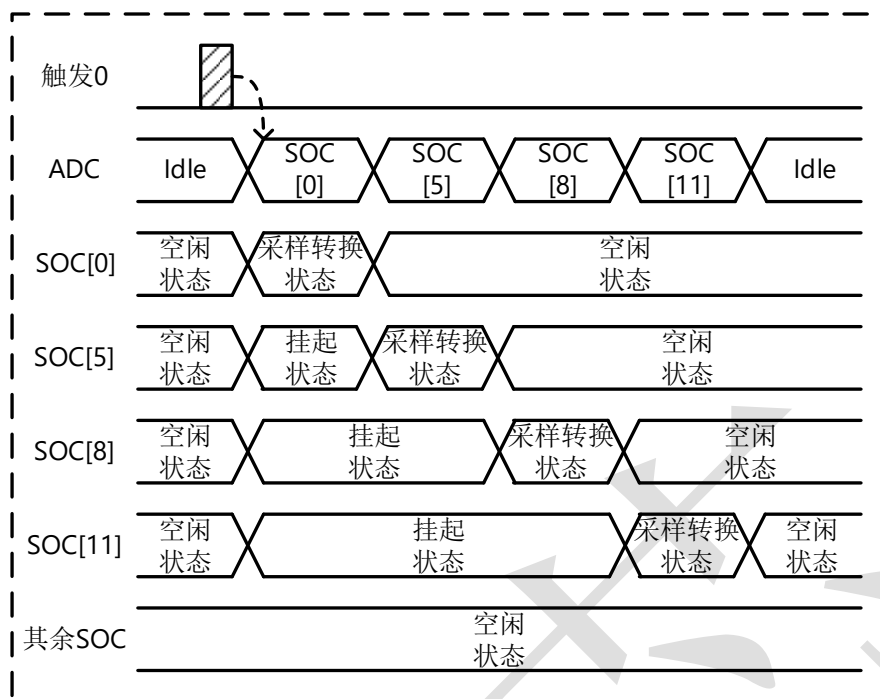


图 10-13 SOC 状态时序图

当有 SOC 处在非空闲状态（挂起状态或采样转换状态）时，又接收到一次新的触发，则会产生挂起溢出事件。

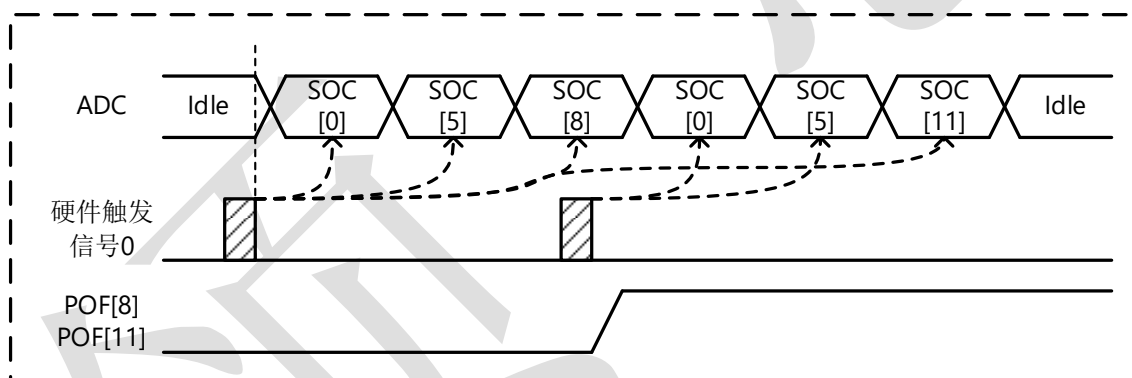


图 10-14 SOC 挂起溢出

其中，SOC[8]和 SOC[11]并未进入空闲状态，当硬件触发信号 0 再次触发时，对应的 POF[8]和 POF[11]标志位置起。

例如，配置了 SOC[0]的 FP=1，SOC[5]的 FP=0，则 POF 事件发生后，挂起的 SOC 的行为将会发生改变。SOC[0]会重新被挂起，而 SOC[5]则忽略的本次硬件触发。

10.2.7 采样模式

ADC 模块拥有两路独立的采样/保持电路，和一个电压转换电路。每路独立的采样/保持电路分别对应通道 A 组和通道 B 组。值得注意的是，采样通道如果有 300us 以上的空闲时 ADC 会自动进入休眠模式，此时需要重新启动 ADC，即依次配置 ADC_CR.STOP 位、ADC_CR.DIS 位和 ADC_CR.EN 位，其他的 ADC 配置寄存器不会受到影响。

通过配置 ADC_SOC[x]CFGR.CON 决定 SOC[x]被采样转换时的采样模式：

1. 单采样模式：每次转换周期内只有一路采样/维持电路被激活，进入工作。此模式采样对于 SOC 的通道选择没有任何限制，可以选择 A 组通道内的 IO 或 B 组通道内的 IO。

2. 双采样模式：每次转换周期内两路路采样/保持电路都会被被激活，进入工作。此模式用于在两次采样点的时间在同一时刻时，两个 SOC 必须分配一个在通道 A 组和另一个在通道 B 组。

A 组通道	B 组通道
ADCIN[00]	ADCIN[08]
ADCIN[01]	ADCIN[09]
ADCIN[02]	ADCIN[10]
ADCIN[03]	ADCIN[11]
ADCIN[04]	ADCIN[12]
ADCIN[05]	ADCIN[13]
ADCIN[06]	ADCIN[14]
ADCIN[07]	ADCIN[15]

当两个通道需要配置同时采样时，需要配置两个为一组双采样 SOC 的 $ADC_SOC[x]CFGR.CON = 1$ ， $ADC_SOC[x]CFGR.FP = 0$ ，并配置相同的 $ADC_SOC[x]CFGR.TRIGSEL$ 硬件触发。

第 1 组双采样	SOC[00]	SOC[01]
第 2 组双采样	SOC[02]	SOC[03]
第 3 组双采样	SOC[04]	SOC[05]
第 4 组双采样	SOC[06]	SOC[07]
第 5 组双采样	SOC[08]	SOC[09]
第 6 组双采样	SOC[10]	SOC[11]
第 7 组双采样	SOC[12]	SOC[13]
第 8 组双采样	SOC[14]	SOC[15]

注：两种采样模式下 ADC 的转换速率没有区别。

10.2.8 触发单元

ADC 模块同时支持软件触发和硬件触发。

软件触发形成的序列，完全由软件本身控制，可以实现任意的序列组合。

硬件触发形成的序列，数量等同于触发源数量，为 4 个。四个硬件触发的序列共使用 16 个 SOC 组成，不能重合，SOC 内的通道绑定可以灵活配置。

10.2.8.1 软件触发

ADC 模块支持软件触发，通过写 $ADC_CR.SW$ 可以单独触发每一个 SOC。软件触发也会产生 POF 事件，但是不受到 $ADC_SOC[x]CFGR.FP$ 位影响。

软件 触发	SW [0]	SW [1]	SW [2]	SW [3]	SW [4]	SW [5]	SW [6]	SW [7]	SW [8]	SW [9]	SW [10]	SW [11]	SW [12]	SW [13]	SW [14]	SW [15]
SOC	SOC [0]	SOC [1]	SOC [2]	SOC [3]	SOC [4]	SOC [5]	SOC [6]	SOC [7]	SOC [8]	SOC [9]	SOC [10]	SOC [11]	SOC [12]	SOC [13]	SOC [14]	SOC [15]

10.2.8.2 硬件触发

ADC 模块配备了 4 个硬件触发源，这 4 个硬件触发源被标识为硬件触发 0、硬件触发 1、硬件触发 2 和硬件触发 3，具体配置请参考 ADC 互联配置寄存器 11/12（ $SysCtrl_EDU_CFG11/12$ ）。在这 4 个硬件触发源各自均带有一个延迟单元，可以是实现 0 到 1008 个 ADC_CLK 范围内的触发延迟。触发连接关系如下表：



表 10-2 硬件触发连接表

互联配置 ADC_EXTSELx	硬件触发源 0	硬件触发源 1	硬件触发源 2	硬件触发源 3
0	TIM1_trig_oc1	TIM1_trig_oc2	TIM1_trig_oc3	TIM1_trig_oc4
1	TIM2_trig_oc1	TIM2_trig_oc2	TIM2_trig_oc3	TIM2_trig_oc4
2	TIM15_trig_oc1	TIM15_trig_oc2	TIM16_trig_oc1	TIM17_trig_oc1
3	EXTI1	EXTI3	EXTI9	EXTI11
4	TIM1_trig_oc2	TIM2_trig_oc1	TIM16_trig_oc1	TIM1_trgo
5	TIM1_trig_oc3	TIM2_trig_oc3	TIM17_trig_oc1	TIM2_trgo
6	TIM1_trig_oc4	TIM2_trig_oc4	ACMP1_out	TIM15_trgo
7	TIM1_trig_oc5	ACMP0_out	ACMP1_out	TIM1_trig_oc5

硬件触发框图如下所示。

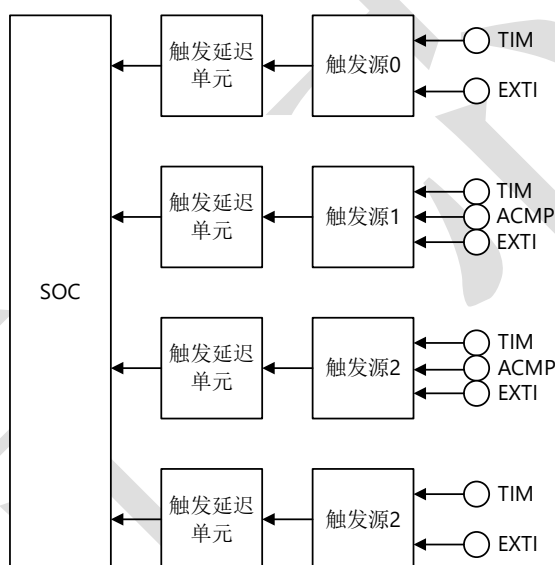


图 10-15 硬件触发源框图

10.2.9 模拟窗口看门狗（AWD）

模拟看门狗 Analog-Watch-Dog（简称 AWD），是内建在 ADC 模块内的一个独立功能单元，专用于监控 SOC 的通道数据值。ADC 内有两个独立的 AWD 模块，分别为 AWD[0]和 AWD[1]。

AWD 的警报脉冲拥有两个输出，一个用于产生中断标志位，另一个用于和其他外设进行联动。AWD 的功能有三部分配置：

1. ADC_AWD[x]CFGR.AWDSLE[15:0]，监控 SOC 通道配置，可以同时监控任意个 SOC 的通道（0~15），全为 0 则关闭 AWD。每一位对应每个 SOC，即 ADC_AWD[x]CFGR.AWDSLE[x]对应 SOC[x]。
2. ADC_AWD[x]TR，AWD 阈值寄存器，设定高低阈值进行电压窗口的监控，如图 10-16 所示。
3. ADC_ISR.AWD[x]、ADC_IER.AWD[x]IE，AWD 相关中断配置。

如果使能的 SOC[x]通道内模拟电压转换出的数据值低于低阈值或高于高阈值时，AWD 标志位被硬件置位并输出一个单脉冲的 awd_out 信号。如果此时 ADC_IER.AWDIE = 1，则还会产生一个 AWD 中断。

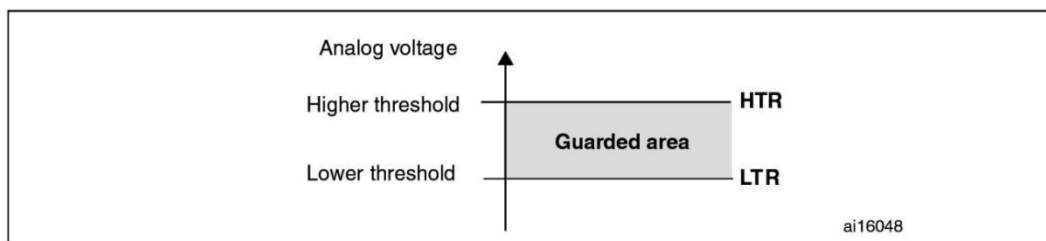


图 10-16 模拟看门狗监控区域

下表列出配置 ADC_AWD[x]CFGR.AWDSEL[15:0]开启 AWD 任意 SOC 监控。

表 10-3 AWD 通道选择

AWD[0]SEL[x]	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWD[1]SEL[x]	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOC[x]	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

10.3 DMA 搬运

ADC 模块支持 DMA 搬运功能，这是一个高效的数据传输方式，可以在不涉及处理器的情况下，将数据直接从 ADC 模块搬运到内存。

DMA 搬运功能通过寄存器 DMACFG（DMA 配置寄存器）和 DADDR（DMA 地址寄存器）进行配置。在 DMACFG 寄存器中有两个关键的配置位，分别是 DEN[x]和 REQ[x]。其中 DMACFG.DEN[x]用于标识哪些 DR[x]内的数据需要被 DMA 搬运；DMACFG.DREQ[x]决定在哪个 SOC[x]采样转换完成后触发 DMA 的请求。具体操作流程如下：

1. 首先将需要 DMA 搬运的数据的 SOC[x]的 DEN[x]配置为 1，这样当 DMA 访问 DADDR 时，就能按照从小到大的顺序，依次映射到对应的 DR[x]寄存器中。
2. 然后在最后一个需要传输的数据上配置 DREQ[x]，使其在数据产生后发出 DMA 请求，也可以根据 DMA 搬运规则设置不同的请求。
3. 最后在 DMA 的配置中也需要选择 DADDR 作为目标地址，并设置固定地址偏移。

举例来说，如图 10-17 所示，需要设置 DEN[0]、DEN[5]、DEN[8]、DEN[11]和 DREQ[11]，并且 DMA 配置了一次连续传输 4 个数据。当 SOC[0]、SOC[5]、SOC[8]、SOC[11]触发后，ADC 将依次采样转换并产生数据。最后，当 DATA[11]产生后，ADC 会发起 DMA 搬运请求，DMA 将会访问 DADDR 寄存器并搬运这四个数据。

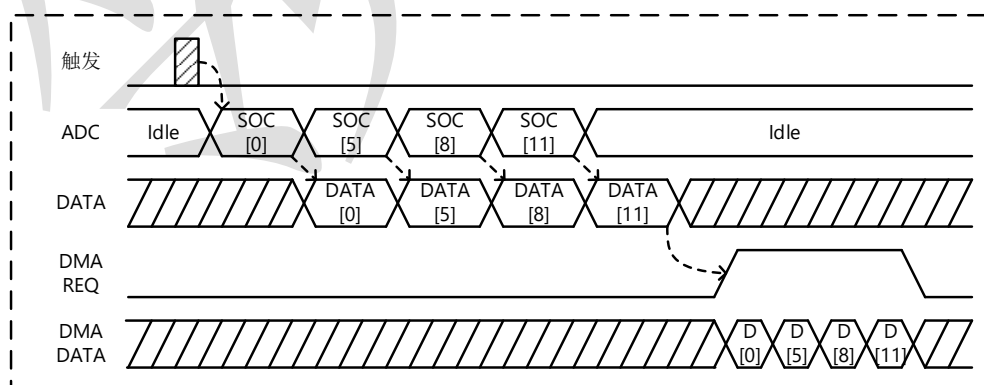


图 10-17 DMA 搬运 ADC 数据的过程

10.4 ADC 中断

ADC 具有一个中断向量，其中分为两类：

- ADC 普通中断，可由数据溢出 DOF 事件或 AWD 事件产生中断标志位
 - DOF：数据溢出，有数据寄存器 DR[x]的 EOC[x]标志位没有清除时，SOC[x]又产生一个新的数据时产生
 - AWD：模拟看门狗警报，模拟看门狗检测的通道中有数据值超过阈值时产生
- ADC_SOC 中断，可由 EOC 转换结束事件或 POF 挂起溢出事件产生中断标志位
 - EOC[x]：对应 SOC[x]完成一次采样转换并产生一个新的数据
 - POF[x]：对应 SOC[x]在挂起状态状态下，又接收到一次触发

表 10-4 ADC 中断事件

中断事件	中断标志位	中断使能位
模拟看门狗事件 (x=0~1) (AWD[x] monitor data exceeds the threshold)	AWD[x]	AWDIE[x]
数据溢出事件 (Data over flow)	DOF	DOFIE
挂起溢出事件 (x=0~15) (Pending over flow on the SOC[x])	POF[x]	POFIE[x]
数据转换完成事件 (x=0~15) (End of conversion on the SOC[15:0])	EOC[x]	EOCIE[x]

每一个中断都有自己的中断使能控制 (ADC_IER、ADC_SOC_IER)，最后“或”成一个总的 ADC 中断信号，接入到对应的 ADC 全局中断向量中。在中断服务程序中，可以通过寄存器 ADC_ISR、ADC_SOC_ISR 查看详细的中断情况。

AWD[x]在监控到配置的 SOC[x]产生的数据超过设定上下阈值时，置位 ADC_ISR[1:0].AWD[x]中断标志位。

当发生 DOF 事件时，除了寄存器 ADC_ISR.DOOF 标志位置位外，用户可以通过查询寄存器 ADC_ISR.SDOF[15:0]来确认具体溢出的 DR[x]寄存器。在写 1 清除 ADC_ISR[15].DOOF 标志位时，硬件同时清除寄存器 ADC_ISR.SDOF[15:0]标志位。

SOC[x]处于挂起状态时，又接收到新的触发，会将相应 POF[x]置位。

SOC[x]完成采样转换并产生新数据时，对应的 EOC[x]标志位硬件置位。对相应标志位写 1 或读相应 DR[x]时，清除 EOC[x]标志位。当发起 STOP 命令后，ADC 仍有可能产生一次 EOC 标志。

10.5 ADC 寄存器描述

表 10-5 ADC 相关寄存器表

名称	说明	读写权限	复位值	字节地址
ADC_SR	ADC 状态寄存器	R	0x0000_0000	0x4001_A000
ADC_ISR	ADC 中断状态寄存器	R/W	0x0000_0000	0x4001_A004
ADC_IER	ADC 中断使能寄存器	R/W	0x0000_0000	0x4001_A008
ADC_SOC_ISR	ADC_SOC 中断状态寄存器	R/W	0x0000_0000	0x4001_A00C
ADC_SOC_IER	ADC_SOC 中断使能寄存器	R/W	0x0000_0000	0x4001_A010
ADC_CR	ADC 控制寄存器	R/W	0x0000_0000	0x4001_A014
ADC_CFGR	ADC 配置寄存器	R/W	0x0800_0000	0x4001_A018

ADC_SMPR	ADC 采样周期寄存器	R/W	0x0000_0247	0x4001_A01C
ADC_TRIM	ADC 校准寄存器	R/W	0x0097_00A3	0x4001_A020
ADC_ANA	ADC 模拟寄存器	R/W	0x0000_F0DC	0x4001_A024
ADC_DMACHCFG	ADC DMA 配置寄存器	R/W	0x0000_0000	0x4001_A028
ADC_DADDR	ADC DMA 传输寄存器	R	0x0000_0000	0x4001_A02C
ADC_AWD[x]CFGR	ADC 模拟看门狗[x]配置寄存器	R/W	0x0000_0000	0x4001_A040~ 0x4001_A044
ADC_AWD[x]TR	ADC 模拟看门狗[x]阈值寄存器	R/W	0x0FFF_0000	0x4001_A050~ 0x4001_A054
ADC_EXTCFGRO	ADC 外部触发配置寄存器 0	R/W	0x0000_0000	0x4001_A060
ADC_SOC[x]CFG	ADC_SOC 配置寄存器	R/W	0x0000_0F00	0x4001_A080 ~ 0x4001_A0BC
ADC_DR[x]	ADC 数据寄存器	R	0x0000_0000	0x4001_A0C0 ~ 0x4001_A0FC
SysCtrl_EDU_CFG11	ADC 互联配置寄存器 11	R/W	0x0000_0000	0x4800_704C
SysCtrl_EDU_CFG12	ADC 互联配置寄存器 12	R/W	0x0000_0000	0x4800_7050

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）。

10.5.1 ADC 状态寄存器（ADC_SR）

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

Bits 31:16	PENDING[15:0] : SOC[x]挂起状态位（每次通道切换后更新），该位域只读
	0: 表示 SOC[x]的处于空闲状态（默认）
	1: 表示 SOC[x]的处于挂起状态（表示处于正在采样转换或等待采样转换）
	此状态位比实际状态延迟 2 个 PCLK，软件读取此位可以忽略这个延迟
Bits 15:2	保留，必须保持复位值
Bit 1	BUSY : 忙状态位，该位只读
	0: 表示 ADC 模块不处在采样转化的工作状态中（默认）
	1: 表示 ADC 模块处在采样转化的工作状态中
Bit 0	READY : 准备完成状态位，该位只读
	0: 表示 ADC 模块处在掉电状态或上电稳定中，无法进行工作（默认）
	1: 表示 ADC 模块已经上电完成且稳定，可以开始工作

10.5.2 ADC 中断状态寄存器 (ADC_ISR)

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
SDOF[15:0]															
r															

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

Bits	31:16	SDOF[15:0]: 单数据溢出状态位, 该位域只读, 由硬件置位和清除
		0: DR[x]寄存器未发生数据溢出 (默认)
		1: DR[x]寄存器发生了数据溢出
Bit	15	DOF: 数据溢出标志位, 由硬件置位, 软件写 1 清除
		0: 无数据溢出事件 (默认)
		1: 有 EOC[x]标志位未清除时同一个 SOC[x]又产生一个新的数据
Bits	14:2	保留, 必须保持复位值
Bits	1:0	AWD[1:0]: 模拟看门狗标志位, 由硬件置位, 软件写 1 清除
		0: 无 AWD 事件 (默认)
		1: AWD[x]检测到 SOC 产生的数据超出阈值

10.5.3 ADC 中断使能寄存器 (ADC_IER)

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
DOFIE	Res.													AWDIE[1:0]	
rw														rw	

Bits	31:16	保留, 必须保持复位值
Bit	15	DOFIE: DOF 中断使能位
		0: DOF 中断禁止 (默认)
		1: DOF 中断使能, 当 ADC_ISR.DO F 置位时产生中断
Bits	14:2	保留, 必须保持复位值
Bits	1:0	AWDIE[1:0]: AWD 中断使能位
		0: AWD[x]中断不使能 (默认)
		1: AWD[x]中断使能, 当 ADC_ISR.AWD[x]置位时产生中断

注: 只有当 ADC_CR.START=0 时 (ADC 不处于工作状态) 才允许改写 ADC_IER 寄存器, 否则写操作无效。

10.5.4 ADC_SOC 中断状态寄存器 (ADC_SOC_ISR)

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
POF[15:0]															
rw															

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

Bits	31:16	POF[15:0]: 挂起溢出标志位, 由硬件置位, 软件写 1 清除
		0: 无挂起溢出事件 (默认)
		1: 当 SOC[x]处于挂起状态时又接收到触发
Bits	15:0	EOC[15:0]: ADC 转换完成标志位, 由硬件置位, 软件写 1 清除
		0: 无新数据产生 (默认)
		1: SOC[x]完成采样转换且对应 DR[x]中有新的数据
		读取 DR[x]内的数据也能清除该标志位

10.5.5 ADC_SOC 中断使能寄存器 (ADC_SOC_IER)

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
POFIE[15:0]															
rw															

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

Bits	31:16	POFIE[15:0]: POF[x]中断使能位
		0: POF[x]中断禁止 (默认)
		1: POF[x]中断使能, 当 ADC_SOC_ISR.POF[x]置位时产生中断
Bits	15:0	EOCIE[15:0]: EOC[x]中断使能位
		0: EOC[x]中断禁止 (默认)
		1: EOC[x]中断使能, 当 ADC_SOC_ISR.EOC[x]置位时产生中断
注: 只有当 ADC_CR.START=0 时 (ADC 不处于工作状态) 才允许改写 ADC_SOC_IER 寄存器, 否则写操作无效。		

10.5.6 ADC 控制寄存器 (ADC_CR)

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
SW[15:0]															
w															

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

Bits 31:16	SW[15:0]: SOC 软件触发位，该位域只写，读回值固定为 0
	写：
	0: 无效操作
	1: 软件触发 SOC[x]并使其变为挂起状态
	注：当前 SOC[x] 状态通过 ADC_SR.PENDING[x] 位查询
Bits 15:4	保留，必须保持复位值
Bit 3	STOP: 停止工作命令
	写：
	0: 无效操作
	1: 发起 STOP 命令，在完成当前进行中 SOC[x]后，丢弃所有挂起的 SOC
	读：
	0: 表示无 STOP 命令进行中（默认）
	1: 表示正在停止 ADC 模块的工作
	发起 STOP 命令且 ADC 模块停止工作后，ADC_CR.STOP 和 ADC_CR.START 同时由硬件清 0
	注：只有当 ADC_CR.START=1 时写 1 有效。
Bit 2	START: 开始工作命令
	写：
	0: 无效操作
	1: 发起 STSRT 命令，即 ADC 模块进入工作的状态，锁定所有配置且允许接收触发
	读：
	0: 表示 ADC 模块不在工作中，可以进行配置且忽略所有触发（默认）
	1: 表示 ADC 模块正在工作中，锁定所有配置且允许接收触发
	注：只有当 ADC_SR.READY=1 且 ADC_CR.DIS=0 时写 1 有效。
Bit 1	DIS: ADC 关闭
	写：
	0: 无效操作
	1: 发起 DIS 命令，即关闭 ADC 控制器
	读：
	0: 表示无 DIS 命令进行中（默认）
	1: 表示 DIS 命令进行中，正在关闭 ADC
	发起 DIS 命令且完成后，ADC_CR.DIS 和 ADC_CR.EN 同时由硬件清 0
	注：只有当 ADC_SR.EN=1 且 ADC_CR.START=0 时写 1 有效。
Bit 0	EN: ADC 使能

	写:
	0: 无效操作
	1: 发起 EN 命令, 即使能 ADC 控制器
	读:
	0: 表示 ADC 控制器处于关闭状态 (默认)
	1: 表示 ADC 控制器处于使能状态
	注: 只有对 ADC_TRIM 写入校准值后写 1 有效。

10.5.7 ADC 配置寄存器 (ADC_CFGR)

地址偏移: 0x18

复位值: 0x0800_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CKMODE[4:0]					Res.							MD2	VREF	VFS	LP
rw												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SVS	SV	STS	ST	Res.										OVF	ALIGN
rw	rw	rw	rw											rw	rw

Bit	31	CKMODE[4]: ADC 时钟源选择
		0: ADC_CLK 使用 PCLK 作为时钟源 (默认)
		1: ADC_CLK 使用 PLL_CLK 作为时钟源
Bits	30:27	CKMODE[3:0]: ADC 时钟分频
		0000: ADC_CLK = 时钟源
		0001: ADC_CLK = 时钟源/2 (默认)
		0010: ADC_CLK = 时钟源/3
		0011: ADC_CLK = 时钟源/4
		0100: ADC_CLK = 时钟源/5
		0101: ADC_CLK = 时钟源/6
		0110: ADC_CLK = 时钟源/8
		0111: ADC_CLK = 时钟源/10
		1000: ADC_CLK = 时钟源/12
		1001: ADC_CLK = 时钟源/14
		其他: ADC_CLK = 时钟源/16
Bits	27:18	保留, 必须保持复位值
Bit	19	MD2: ADC 工作模式
		0: 正常模式 (默认)
		1: 稳定模式, 采样转换周期 T _{cc} 会比正常模式多一个 ADC_CLK 周期
		注: 建议在较低电压、高温等极限条件下, 且外部输入信号驱动能力较强时, 开启稳定模式, 以保持 ADC 采样精准度。
Bit	18	VREF: 参考电压选择
		0: 内部电压 (3.2V、4.9V) (默认)

	1: 外部电压 (VDDA)
Bit 17	VFS: 当选择内部电压作为参考电压时, 满量程电压选择
	0: 满量程电压为 3.2V (默认)
	1: 满量程电压为 4.9V
Bit 16	LP: 低功耗模式 (在低频率转换下, 可以开启 LP 降低 ADC 功耗)
	0: 正常工作模式 (默认)
	1: 开启低功耗模式
Bit 15	SVS: 内部电压监控映射 SOC[x] (被映射 SOC 的通道选择无效)
	0: 将 SOC[15]映射为内部电压通道 (默认)
	1: 将 SOC[7]映射为内部电压通道
Bit 14	SV: 内部电压监控通道使能位
	0: 不使能内部电压监控模式 (默认)
	1: 使能内部电压监控模式, 并将电压监控映射到 SOC[x]上 (具体详见 SVS)
Bit 13	STS: 温度监控映射 SOC[x] (被映射 SOC 的外部通道选择无效)
	0: 将 SOC[15]映射为温度通道 (默认)
	1: 将 SOC[7]映射为温度通道
Bit 12	ST: 温度监控通道使能位
	0: 不使能温度监控模式 (默认)
	1: 使能温度监控模式, 并将温度监控映射到 SOC[x]上 (具体详见 STS)
注: ST 和 SV 禁止同时映射在同一个 Soc[x]上, 否则采样值错误。	
Bits 11:2	保留, 必须保持复位值
Bit 1	OVF: 数据溢出模式
	0: 发生 DOF 事件时, 保留新数据并写入 ADC_DR[x] (默认)
	1: 发生 DOF 事件时, 保留老数据并舍弃新数据
Bit 0	ALIGN: 数据对齐模式
	0: 数据右对齐 16 位, 高位补零 (默认)
	1: 数据左对齐 16 位, 低位补零
注: 只有当 ADC_CR.START=0 时 (ADC 不处于工作状态) 才允许改写 ADC_CFGR 寄存器, 否则写操作无效。	

10.5.8 ADC 采样周期寄存器 (ADC_SMPR)

地址偏移: 0x1C

复位值: 0x0000_0247

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

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

Bits 31:22	保留, 必须保持复位值
Bits 21:20	STSPW[1:0]: 温度采样脉冲宽度 (ST sample pulse width)

	00: 温度采样脉冲宽度 $T_{ST_SP} = 0.5\mu s$ (默认)
	01: 温度采样脉冲宽度 $T_{ST_SP} = 1.0\mu s$
	10: 温度采样脉冲宽度 $T_{ST_SP} = 1.5\mu s$
	11: 温度采样脉冲宽度 $T_{ST_SP} = 2.0\mu s$
	注: 以上时间均为 ADC_CLK 为 $24MHz$ 时计算得出。
Bits 19:16	保留, 必须保持复位值
Bits 15:12	CC[3:0]: 采样周期 (Conversion cycle), 默认为 0
	当 $T_{SPW} < 12 * ADC_CLK$, 采样周期 $T_{CC} = (CC + CPW + 14) * ADC_CLK$
	当 $T_{SPW} \geq 12 * ADC_CLK$, 采样周期 $T_{CC} = (CC + CPW + SPW + 2) * ADC_CLK$
Bits 11:10	保留, 必须保持复位值
Bits 9:8	CPW[1:0]: 转换脉冲宽度 (Convert pulse width), 默认值为 10
	转换脉冲宽度 $T_{CP} = (CPW + 1) * ADC_CLK$
Bits 7:4	SPW[3:0]: 采样脉冲宽度 (sample pulse width), 默认值为 0100
	采样脉冲宽度 $T_{SP} = (SPW + 1) * ADC_CLK$
	注: SPW 配置要求 $T_{SP} < T_{CC}$ 。
Bit 3	保留, 必须保持复位值
Bits 2:0	APW[2:0]: 上电调整脉冲宽度 (adjust pulse width), 默认值为 111
	上电调整脉冲宽度 $T_{AP} = (APW + 1) * ADC_CLK$
	如果需要减少 ADC 上电稳定的时间, 可以适当减少 APW 值
注 1: 在 ADC_CLK 为 $24MHz$ 时, $SMPR$ 寄存器默认值为 ADC 典型工作时序, 也是推荐配置值。	
注 2: $STSPW$ 的脉冲宽度依据 ADC_CLK 线性变化。	
注 3: 只有当 $ADC_CR.START=0$ 时 (ADC 不处于工作状态) 才允许改写 ADC_SMPR 寄存器, 否则写操作无效。	

10.5.9 ADC 校准寄存器 (ADC_TRIM)

地址偏移: $0x20$

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]							
								rw							

Bits 31:24	保留, 必须保持复位值
Bits 23:16	VOS_TRIM[7:0]: ADC 偏移校准 (默认值 10010111)
Bits 15:8	保留, 必须保持复位值
Bits 7:0	GAIN_TRIM[7:0]: ADC 增益校准 (默认值为 10100011)
注 1: vos_trim 和 $gain_trim$ 在 ADC 上电启动前需要写入校准值, 否则硬件禁止启动 ADC。	
注 2: 如果无相关校准值, 请通过软件重新载一次复位值。	
注 3: 只有当 $ADC_CR.START=0$ 时 (ADC 不处于工作状态) 才允许改写 ADC_TRIM 寄存器, 否则写操作无效。	

10.5.10 ADC 模拟寄存器 (ADC_ANA)

地址偏移: 0x24

复位值: 0x0000_F0DC

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
PD_ ADC	PD_ REF	PD_ SHB	PD_ SHA	Res.											
rw	rw	rw	rw												

Bits	31:16	保留, 必须保持为复位值
Bit	15	PD_ADC: 关闭 ADC 内核
		0: ADC 内核模块上电
		1: ADC 内核模块掉电 (默认)
Bit	14	PD_REF: 关闭参考电压产生模块, 并整体关闭 ADC
		0: ADC 参考电压模块上电
		1: ADC 参考电压模块掉电 (默认)
		注: PD_REF=1 时, ADC 模块整体掉电, 包括 ADC 内核、Sample/Hold 模块和参考电压模块; PD_REF=0 时, ADC 模块整体上电, 同时可以给其他模块提供参考电压, 但需要额外配置 PD_ADC、PD_SHA 和 PD_SHB 上电后 ADC 才能正常工作。
Bit	13	PD_SHB: 关闭 Sample/Hold B 模块
		0: ADC 采样/保持电路 B 上电
		1: ADC 采样/保持电路 B 掉电 (默认)
Bit	12	PD_SHA: 关闭 Sample/Hold A 模块
		0: ADC 采样/保持电路 A 上电
		1: ADC 采样/保持电路 A 掉电 (默认)
Bits	11:0	保留, 必须保持为复位值
注 1: 只有当 ADC_CR.START=0 时 (ADC 不处于工作状态) 才允许改写 ADC_ANA 寄存器, 否则写操作无效。		
注 2: 当进入超低功耗停机模式时, ADC 所有模块会自动关闭; 其他情况下, 可以通过分别关闭 ADC 内部模块来减小功耗。但要注意关闭 PD_REF 后, ADC 参考电压关闭, 会影响其他模块的参考电压。		

10.5.11 ADC DMA 配置寄存器 (ADC_DMACFG)

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



rw

Bits 31:16	DEN[15:0]: SOC 对应的 DMA 数据传输使能位
	0: SOC[x]的 DMA 数据传输禁止（默认）
	1: SOC[x]的 DMA 数据传输使能, 对应的 DR[x]能通过 ADC_DADDR 读访问
Bits 15:0	DREQ[15:0]: SOC 对应的 DMA 数据传输请求位
	0: 当 SOC[x]的 EOC 置位时, 不向 DMA 发起连续传输请求（默认）
	1: 当 SOC[x]的 EOC 置位时, 向 DMA 发起连续传输请求
注: 只有当 ADC_CR.START=0 时 (ADC 不处于工作状态) 才允许改写 ADC_DMACFG 寄存器, 否则写操作无效。	

10.5.12 ADC DMA 传输寄存器 (ADC_DADDR)

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
DMA_DATA [15:0]															
r															

Bits 31:16	保留, 必须保持复位值
Bits 15:0	DMA_DATA[15:0]: DMA 传输寄存器, 该位域只读
	对 ADC_DADDR 寄存器的读会导致对以下地址的寄存器的读访问:
	ADC_DR[x], 其中 x 为 DEN[x]为 1 且 EOC 产生数据存入此 DEN[x]后的序号

10.5.13 ADC 模拟看门狗[x]配置寄存器 (ADC_AWD[x]CFGR) (x=0~1)

地址偏移: 0x40, 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
AWDSEL[15:0]															
rw															

Bits 31:16	保留, 必须保持为复位值
Bits 15:0	AWDSEL[15:0]: 模拟看门狗监控使能位, 每一位 AWDSEL[x]对应每一个 SOC[x]通道
	0: 不使能 AWD[x]监控 SOC[x]通道数据值（默认）
	1: 使能 AWD[x]监控 SOC[x]通道数据值

10.5.14 ADC 模拟看门狗[x]阈值寄存器（ADC_AWD[x]TR）（x=0~1）

地址偏移：0x50，0x54

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，必须保持为复位值
Bits	27:16	HT[11:0] ：模拟看门狗监控高阈值（默认值 0xFFFF）
Bits	15:12	保留，必须保持为复位值
Bits	11:0	LT[11:0] ：模拟看门狗监控低阈值（默认值 0x000）

10.5.15 ADC 外部触发配置寄存器 0（ADC_EXTCFG0）

地址偏移：0x60

复位值：0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DLY3[7:0]								DLY2[7:0]							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DLY1[7:0]								DLY0[7:0]							
rw								rw							

Bits	31:24	DLY3[7:0] ：外部触发 3 的延迟时间，具体详见 DLY0（默认为 0x00）
Bits	23:16	DLY2[7:0] ：外部触发 2 的延迟时间，具体详见 DLY0（默认为 0x00）
Bits	15:8	DLY1[7:0] ：外部触发 1 的延迟时间，具体详见 DLY0（默认为 0x00）
Bits	7:0	DLY0[7:0] ：外部触发 0 的延迟时间，默认为 0x00
		DLY0[7:5] = 0xx: $T_{\text{delay}} = \text{DLY0}[6:0] * \text{ADC_CLK}$
		DLY0[7:5] = 10x: $T_{\text{delay}} = (64 + \text{DLY0}[5:0]) * 2 * \text{ADC_CLK}$
		DLY0[7:5] = 110: $T_{\text{delay}} = (32 + \text{DLY0}[4:0]) * 8 * \text{ADC_CLK}$
		DLY0[7:5] = 111: $T_{\text{delay}} = (32 + \text{DLY0}[4:0]) * 16 * \text{ADC_CLK}$

10.5.16 ADC_SOC[x]配置寄存器（ADC_SOC[x]CFG）（x=0~15）

地址偏移：0x80~0xBC

复位值：0x0000_00F0

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.							CON	FP	TRIGSEL[2:0]			CHSEL[3:0]			
							rw	rw	rw			rw			

Bits 31:9	保留，必须保持复位值
Bit 8	CON: 双采样位
	0: SOC[x]为单采样（默认）
	1: SOC[x]为双采样
	<i>注：使用双采样时，需要将配对的两个SOC都置位CON位，否则ADC工作异常。</i>
Bit 7	FP: 强制挂起位
	0: POF事件发生时，SOC[x]不可以被硬件触发进入挂起状态
	1: POF事件发生时，SOC[x]可以被硬件触发进入挂起状态（默认）
Bits 6:4	TRIGSEL[2:0]: SOC[x]触发源选择，默认为111
	000: 硬件触发为外部触发0
	001: 硬件触发为外部触发1
	010: 硬件触发为外部触发2
	011: 硬件触发为外部触发3
	其他: 无触发
Bits 3:0	CHSEL[3:0]: SOC[x]通道选择
	SOC[x]选择外部ADC_IN[15:0]中的一个作为采样转化通道，默认为0000
<i>注：只有当ADC_CR.START=0时（ADC处于不工作状态）才允许改写ADC_SOC[x]CFG寄存器，否则写操作无效。</i>	

10.5.17 ADC数据[x]寄存器（ADC_DR[x]）（x=0~15）

地址偏移：0xC0~0xFC

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
DATA[x][15:0]															
r															

Bits 31:17	保留，必须保持复位值
Bits 15:0	DATA[x][15:0]: ADC_SOC[x]的通道转换值，该位域只读
	16位数据仅12位有效位，数据对齐格式通过ADC_CFGR.ALIGN配置

10.5.18 ADC互联配置寄存器11（SysCtrl_EDU_CFG11）

SysCtrl_EDU_CFG11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

0x05FA659A 到 SysCtrl_KEY 密钥寄存器，且在随后的 64 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。

地址：0x4800_704C

复位值：0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.						ADC_EXTEN1[1:0]		ADC_EXTSEL1[7:0]							
						rw		rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						ADC_EXTEN0[1:0]		ADC_EXTSEL0[7:0]							
						rw		rw							

Bits 31:26	保留，必须保持复位值
Bits 25:24	ADC_EXTEN1[1:0]: ADC 触发源 1 使能控制
	00: 外部硬件上升沿触发（默认）
	01: 外部硬件下降沿触发
	10: 外部硬件双沿触发
	11: 禁止外部硬件触发，仅支持软件触发
Bits 23:16	ADC_EXTSEL1[7:0]: ADC 触发源 1 的选择，默认值 00000000
	ADC 触发源 1 可选择为 ACMP0 /TIM2_TRIG_OC4/TIM2_TRIG_OC3/ TIM2_TRIG_OC1 /EXTI3/ TIM15_TRIG_OC2 /TIM2_TRIG_OC2 /TIM1_TRIG_OC2 输出（从高位到低位）
	所有触发源均可被配置为使能或不使能，通过控制位 ADC_EXTSEL1[x]配置
	所有使能的触发源通过逻辑或的方式组合，有效触发方式通过控制 ADC_EXTEN1[1:0]配置
	ADC_EXTSEL1[x] = 0: 相应触发源不选择
	ADC_EXTSEL1[x] = 1: 相应触发源选择
Bits 15:10	保留，必须保持复位值
Bits 9:8	ADC_EXTEN0[1:0]: ADC 触发源 0 使能控制
	00: 外部硬件上升沿触发（默认）
	01: 外部硬件下降沿触发
	10: 外部硬件双沿触发
	11: 禁止外部硬件触发，仅支持软件触发
Bits 7:0	ADC_EXTSEL0[7:0]: ADC 触发源 0 的选择，默认值 00000000
	ADC 触发源 0 可选择为 TIM1_TRIG_OC5 / TIM1_TRIG_OC4 / TIM1_TRIG_OC3 / TIM1_TRIG_OC2 / EXTI1 / TIM15_TRIG_OC1 / TIM2_TRIG_OC1 / TIM1_TRIG_OC1 输出（从高位到低位）
	所有触发源均可被配置为使能或不使能，通过控制位 ADC_EXTSEL0[x]配置
	所有使能的触发源通过逻辑或的方式组合，有效触发方式通过控制 ADC_EXTEN0[1:0]配置
	ADC_EXTSEL0[x] = 0: 相应触发源不选择
	ADC_EXTSEL0[x] = 1: 相应触发源选择

注：TIMx_TRIG_OCi 和 TIMx_TRGO 不同，详见 TIM 章节说明。

10.5.19 ADC 互联配置寄存器 12 (SysCtrl_EDU_CFG12)

SysCtrl_EDU_CFG12 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 0x05FA659A 到 SysCtrl_KEY 密钥寄存器，且在随后的 64 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。

地址：0x4800_7050

复位值：0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.						ADC_EXTEN3[1:0]		ADC_EXTSEL3[7:0]							
						rw		rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						ADC_EXTEN2[1:0]		ADC_EXTSEL2[7:0]							
						rw		rw							

Bits 31:26	保留，必须保持复位值
Bits 25:24	ADC_EXTEN3[1:0]: ADC 触发源 3 使能控制
	00: 外部硬件上升沿触发（默认）
	01: 外部硬件下降沿触发
	10: 外部硬件双沿触发
	11: 禁止外部硬件触发，仅支持软件触发
Bits 23:16	ADC_EXTSEL3[7:0]: ADC 触发源 3 的选择，默认值 00000000
	ADC 触发源 3 可选择为 TIM1_TRIG_OC5/TIM15_TRGO/TIM2_TRGO/ TIM1_TRGO /EXTI11 / TIM17_TRIG_OC1 /TIM2_TRIG_OC4 /TIM1_TRIG_OC4 输出（从高位到低位）
	所有触发源均可被配置为使能或不使能，通过控制位 ADC_EXTSEL3[x]配置
	所有使能的触发源通过逻辑或的方式组合，有效触发方式通过控制 ADC_EXTEN3[1:0]配置
	ADC_EXTSEL3[x] = 0: 相应触发源不选择
	ADC_EXTSEL3[x] = 1: 相应触发源选择
Bits 15:10	保留，必须保持复位值
Bits 9:8	ADC_EXTEN2[1:0]: ADC 触发源 2 使能控制
	00: 外部硬件上升沿触发（默认）
	01: 外部硬件下降沿触发
	10: 外部硬件双沿触发
	11: 禁止外部硬件触发，仅支持软件触发
Bits 7:0	ADC_EXTSEL2[7:0]: ADC 触发源 2 的选择，默认值 00000000
	ADC 触发源 2 可选择为 ACMP1 / ACMP1 / TIM17_TRIG_OC1 / TIM16_TRIG_OC1 / EXTI9 / TIM16_TRIG_OC1 / TIM2_TRIG_OC3 / TIM1_TRIG_OC3 输出（从高位到低位）
	所有触发源均可被配置为使能或不使能，通过控制位 ADC_EXTSEL2[x]配置
	所有使能的触发源通过逻辑或的方式组合，有效触发方式通过控制 ADC_EXTEN2[1:0]配置

	ADC_EXTSEL2[x] = 0: 相应触发源不选择
	ADC_EXTSEL2[x] = 1: 相应触发源选择
注: TIMx_TRIG_OC _i 和 TIMx_TRGO 不同, 详见 TIM 章节说明。	

11. 数模转换器（DAC）和模拟比较器（ACMP）和反向电动势采样控制器（HALL_MID）

LCM32F06X 包含 1 个 10 位 DAC 模块，可用于将输入的 1 路数字信号转换成 1 个模拟电压输出到 I/O 或者 2 个模拟比较器的输入。DAC 参考电压可选为 ADC 参考电压或 VDDA。DAC 支持硬件触发和 DMA 传输功能。DAC 支持硬件产生可配置的伪随机噪声波形和三角波形。

LCM32F06X 包含 2 个快速的轨到轨模拟比较器 ACMP，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。ACMP 具有 CVREF 功能，CVREF 参考电压可选为 ADC 参考电压或 VDDA。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、ADC 联动。两个比较器可以组合成一个窗口比较器。

LCM32F06X 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

- 参考电压 DAC_VREF 可选为 ADC 参考电压或者 VDDA
- DAC 输出具有可单独使能的驱动 BUFF，可以输出到外部 I/O 或比较器的输入
- 具有 DMA 功能，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 管脚信号

名称	信号类型	备注
DAC_VREF	输入参考电压	可选 ADC 参考电压或 VDDA
DACOUT_BUF	经过 BUFF 的 DAC 输出信号	输出可以到管脚（PA10_YA2/PA13_YA2）或者比较器

11.1.2 ACMP 特性

- 集成 2 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
- CVREF 的参考电压 ACMP_VREF 可选为 ADC 参考电压或者 VDDA
- 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
- 响应速度可配置
- 输出极性可配置
- 支持消隐功能
- 多种联动方式，包括跟 TIMER、ADC 等进行联动
- 输入可选，包括 DAC 输出、外部 I/O 的输入或比较器内部电压 CVREF
- 比较器内部电压 CVREF 可连接到 PMU_AOUT 进行观测
- 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

LCM32F06X 内置 1 个 10 位高精度 DAC。需要注意的是，DACOUT_BUF 输出需要配置 ANA_CTRL.DAC_BUFEN=1。

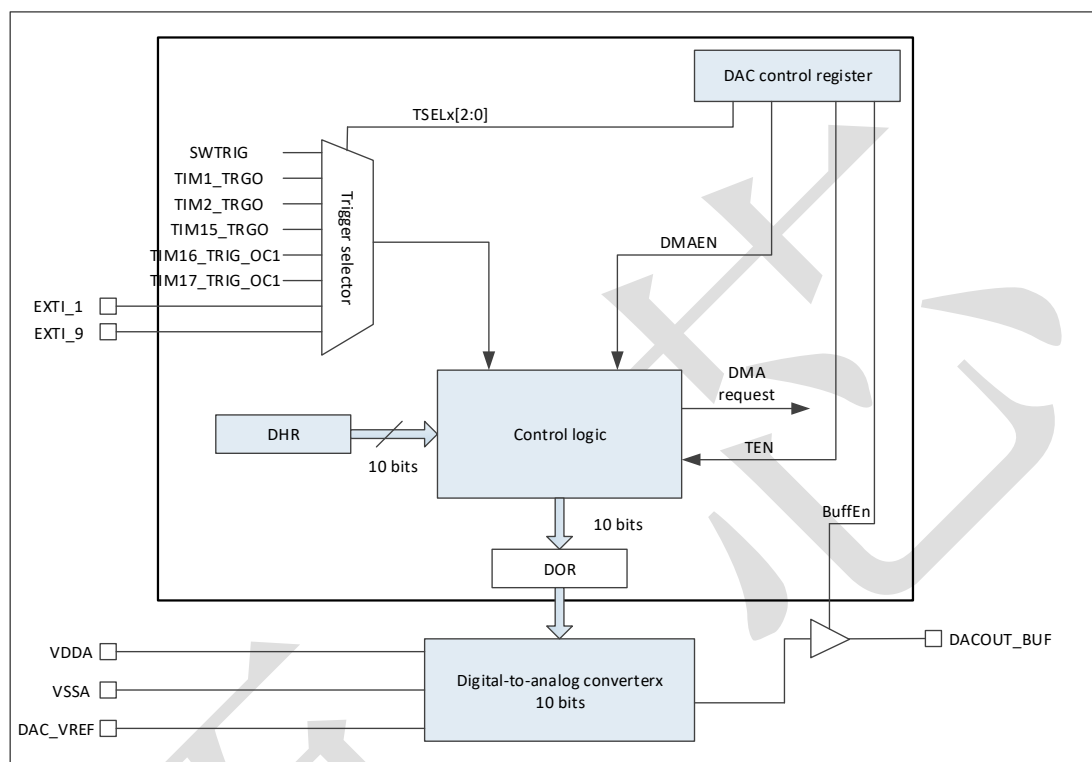


图 11-1 DAC 框图

11.2.1.1 使能配置

DAC 支持软件或硬件触发，即在设定的触发发生时，进行 D/A 转换；可配置触发选择 SysCtrl_EDU_CFG17.DAC_TSEL 和触发使能 SysCtrl_EDU_CFG17.DAC_TEN，具体内容请参考 DAC 互联配置寄存器 17（SysCtrl_EDU_CFG17）。DAC 的使能通过寄存器 DAC_CR.EN 来配置，在 DAC 使能之后，触发相关寄存器、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 DACOUT_BUF，输出时需要通过配置寄存器 ANA_CTRL.DAC_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 DAC_CR.ALIGN=0（右对齐），则转换的数据存储于 DAC_DHR[9:0]
- 当 DAC_CR.ALIGN=1（左对齐），则转换的数据存储于 DAC_DHR[15:6]

当设定的外部触发发生时，DAC_DHR 的值按照上述的格式载入到实际的转换数据寄存器 DAC_DOR 中。

DAC_DOR 寄存器不能被直接写入，只有 SysCtrl_EDU_CFG17.DAC_TEN=2'b11 时，在写入 DHR 的同时，会更新相应 DOR 的内容。而当 SysCtrl_EDU_CFG17.DAC_TEN!=2'b11 时，根据 SysCtrl_EDU_CFG17.DAC_TSEL 所选触发来源，将 DHR 的内容装载入寄存器 DOR。这里的软件触发源是通过配置寄存器 DAC_CSR.SWTRIG 位来设置，硬件完成触发后自动清零。

DAC_DOR 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输

出负载情况。DAC 的输出电压为 $DAC_VREF * DOR/1024$ 。

当 $SysCtrl_EDU_CFG17.DAC_TEN!=2'b11$ 时，可以选择触发事件（软件配置、定时器、EXTI 输入等）来触发 DAC 转换。通过 $SysCtrl_EDU_CFG17.DAC_TSEL$ 选择具体的触发源，详见下表：

表 11-2 DAC 触发选择

触发源	类型	DAC_TSEL[2:0]
SWTRIG	软件设置触发	000
TIM1_TRGO	内部计时器输出，片内信号触发	001
TIM2_TRGO		010
TIM15_TRGO		011
TIM16_TRIG_OC1		100
TIM17_TRIG_OC1		101
EXTI_1	外部输入触发	110
EXTI_9	外部输入触发	111

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 $DAC_CR.WAVE=2'b01$ ，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 DAC_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 $DAC_CR.MAMP$ 控制，可以被部分或者全部屏蔽，此计算值与 DAC_DHR 中的值求和，然后在触发控制下写入到寄存器 DAC_DOR 中。

当 LFSR 全 0 时，一个为 1 的值替代移位进入。当 $WAVE$ 寄存器被重新设置时，LFSR 寄存器恢复到初值。

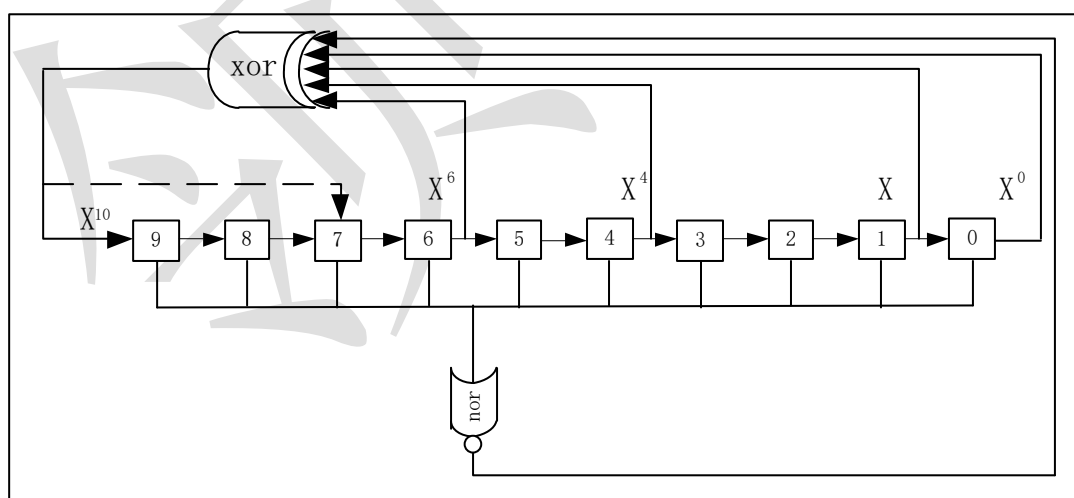


图 11-2 DAC LFSR 计算方式

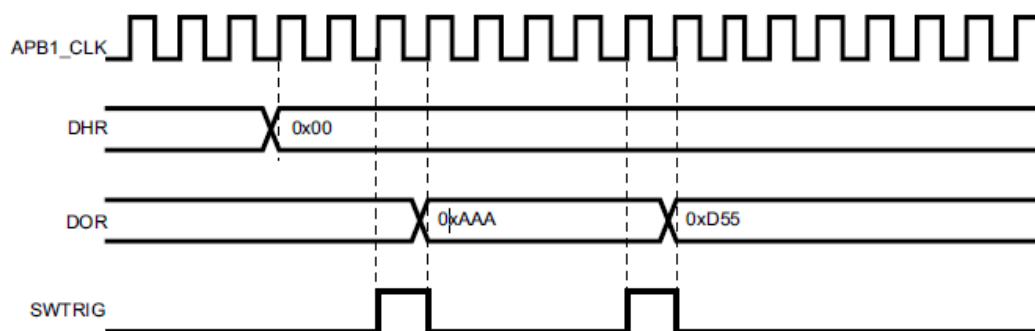


图 11-3 带 LFSR 的 DAC 波形发生过程

注: SysCtrl_EDU_CFG17.DAC_TEN!=2'b11 时才能产生噪声波形。

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 DAC_CR.WAVE=2'b10，幅度通过 DAC_CR.MAMP 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 DAC_DHR 的值求和后，写入到 DAC_DOR 中。在计数幅值达到 DAC_CR.MAMAP 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 DAC_CR.MAMAP 寄存器会复位波形发生器。

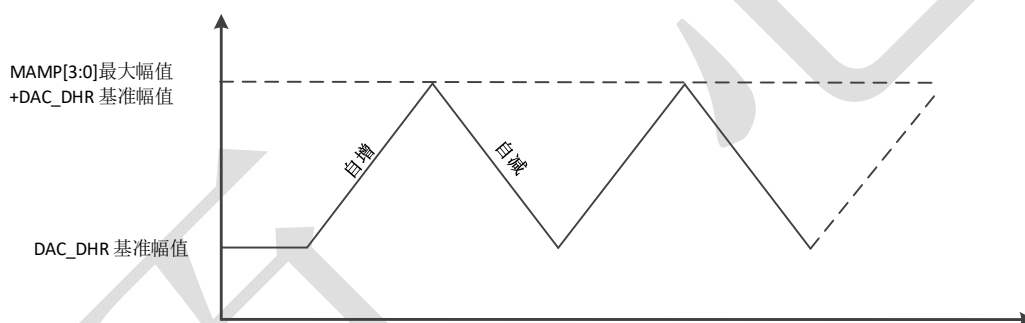


图 11-4 DAC 三角波波形

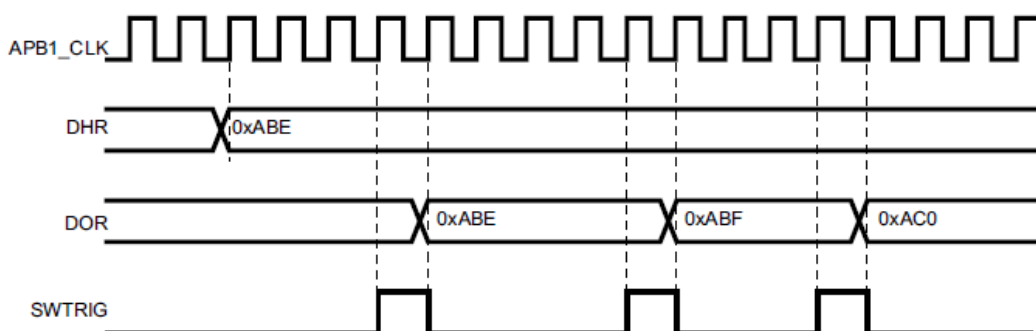


图 11-5 三角波形转换过程（软件触发）

注: SysCtrl_EDU_CFG17.DAC_TEN!=2'b11 时才能产生三角波形，MAMP 寄存器必须在 DAC 使能前配置，否则此寄存器不会更新。

11.2.1.4 DMA 请求和中断

DAC 支持 DMA 通道传输，当收到触发事件时，寄存器 DAC_DHR 的值载入到寄存器 DAC_DOR 中，在寄存器 DAC_CR.DMAEN 使能时，可发出 DMA 请求。

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 DAC_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。如果将 DAC_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

LCM32F06X 内置了 2 个比较器，可以用于模拟电压的比较，比较器 ACMP 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、DACOUT_BUF、HALL_MID 输出或比较器内部电压 CVREF。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
- 具有单次比较模式、单次数列模式、断续序列模式和连续序列模式共四种工作模式

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。

11.2.2.1 结构框图

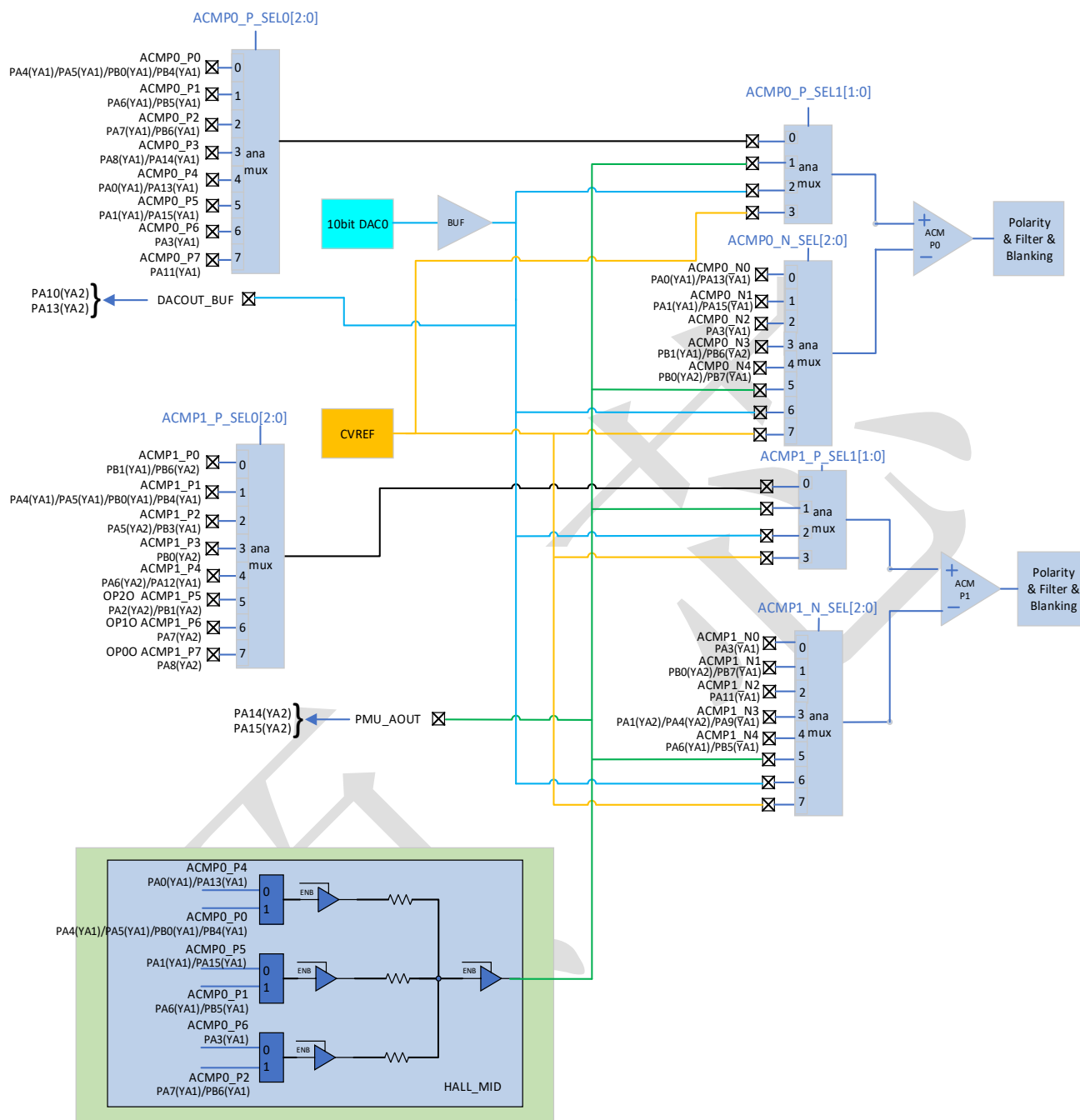


图 11-6 DAC 和 ACMP 内部框图

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。

11.2.2.2 管脚配置

表 11-3 比较器管脚配置

管脚名称	管脚描述	I/O 序号	GPIO 模拟通道		备注
			YA1	YA2	
ACMP0_P0	ACMP0 正端 IO 输入 0	PA4	ADCIN[4] ACMP0_P0 ACMP1_P1	OPA0_N0 OPA2_N0 ACMP1_N3	在单次比较模式下选择 IO 输入需要配置 ACMP0_MUXCR.ACMP_P_SEL1 = 2'b00, 并通过 ACMP0_MUXCR.ACMP_P_SELO 选择对应 IO 输入; 在序列模式下, 可以直接通过 ACMP0_MUXCSR.CHNLSEL0~3 选择轮询的 IO 通道
		PA5	ADCIN[4] ACMP0_P0 ACMP1_P1	ADCIN[5] OPA2_P0 ACMP1_P2	
		PB0	ADCIN[4] ACMP0_P0 ACMP1_P1	ADCIN[8] OPA1_N0 ACMP1_P3 ACMP1_N1 ACMP0_N4	
		PB4	ADCIN[4] ACMP0_P0 ACMP1_P1	-	
ACMP0_P1	ACMP0 正端 IO 输入 1	PA6	ADCIN[6] ACMP0_P1 ACMP1_N4	OPA0_N1 OPA1_N1 OPA2_N1 ACMP1_P4	
		PB5	ADCIN[6] ACMP0_P1 ACMP1_N4	ADCIN[13]	
ACMP0_P2	ACMP0 正端 IO 输入 2	PA7	ADCIN[7] OPA1_P0 ACMP0_P2	ADCIN[12] OPA1_O ACMP1_P6	
		PB6	ADCIN[7] OPA1_P0 ACMP0_P2	ADCIN[9] ACMP0_N3 ACMP1_P0 OPA2_OX	
ACMP0_P3	ACMP0 正端 IO 输入 3	PA8	ADCIN[10] ACMP0_P3 OPA0_OX ELVI	ADCIN[11] OPA0_O ACMP1_P7	
		PA14	ADCIN[10] ACMP0_P3 OPA0_OX ELVI	PMU_AOUT	
ACMP0_P4	ACMP0 正端 IO 输入 4	PA0	ADCIN[0] ACMP0_P4 ACMP0_N0	OPA2_P1	

		PA13	ADCIN[0] ACMP0_P4 ACMP0_N0	DACOUT_BUF	
ACMP0_P5	ACMP0 正端 IO 输入 5	PA1	ADCIN[1] ACMP0_P5 ACMP0_N1 OPA1_OX	OPA0_N0 OPA2_N0 ACMP1_N3	
		PA15	ADCIN[1] ACMP0_P5 ACMP0_N1 OPA1_OX	PMU_AOUT	
ACMP0_P6	ACMP0 正端 IO 输入 6	PA3	ADCIN[3] ACMP0_P6 ACMP0_N2 ACMP1_N0	OPA0_P1	
ACMP0_P7	ACMP0 正端 IO 输入 7	PA11	OPA1_P1 ACMP0_P7 ACMP1_N2	ADCIN[14] OSCL_OUT	
ACMP0_N0	ACMP0 负端 IO 输入 0	PA0	ADCIN[0] ACMP0_P4 ACMP0_N0	OPA2_P1	在单次比较模式下通过 ACMP0_MUXCR.ACMP_N_SEL 选择对应 IO 输入； 在序列模式下，可以直接通过 ACMP0_MUXCSR.CHNLSEL0~3 选择轮询的 IO 通道
		PA13	ADCIN[0] ACMP0_P4 ACMP0_N0	DACOUT_BUF	
ACMP0_N1	ACMP0 负端 IO 输入 1	PA1	ADCIN[1] ACMP0_P5 ACMP0_N1 OPA1_OX	OPA0_N0 OPA2_N0 ACMP1_N3	
		PA15	ADCIN[1] ACMP0_P5 ACMP0_N1 OPA1_OX	PMU_AOUT	
ACMP0_N2	ACMP0 负端 IO 输入 2	PA3	ADCIN[3] ACMP0_N2 ACMP0_P6 ACMP1_N0	OPA0_P1	
ACMP0_N3	ACMP0 负端 IO 输入 3	PB1	ADCIN[9] ACMP0_N3 ACMP1_P0 OPA2_OX	ADCIN[2] OPA2_O ACMP1_P5	
		PB6	ADCIN[7] OPA1_P0 ACMP0_P2	ADCIN[9] ACMP0_N3 ACMP1_P0 OPA2_OX	



ACMP0_N4	ACMP0 负端 IO 输入 4	PB0	ADCIN[4] ACMP0_P0 ACMP1_P1 ACMP0_N4	ADCIN[8] OPA1_N0 ACMP1_P3 ACMP1_N1 ACMP0_N4	
		PB7	ADCIN[8] OPA1_N0 ACMP1_P3 ACMP1_N1 ACMP0_N4	-	
ACMP0_out	ACMP0 输出	PA0	AF6		需要配置对应 IO 的 AF 功能
		PA2	AF6		
		PA6	AF3		
		PA8	AF6		
ACMP1_P0	ACMP1 正端 IO 输入 0	PB1	ADCIN[9] ACMP0_N3 ACMP1_P0 OPA2_OX	ADCIN[2] OPA2_O ACMP1_P5	
		PB6	ADCIN[7] OPA1_P0 ACMP0_P2	ADCIN[9] ACMP0_N3 ACMP1_P0 OPA2_OX	
ACMP1_P1	ACMP1 正端 IO 输入 1	PA4	ADCIN[4] ACMP0_P0 ACMP1_P1	OPA0_N0 OPA2_N0 ACMP1_N3	在单次比较模式下选择 IO 输入需要配置 ACMP1_MUXCR.ACMP_P_SEL1=2'b00, 并通过 ACMP1_MUXCR.ACMP_P_SELO 选择对应 IO 输入; 在序列模式下, 可以直接通过 ACMP1_MUXCSR.CHNLSELO~3 选择轮询的 IO 通道
		PA5	ADCIN[4] ACMP0_P0 ACMP1_P1	ADCIN[5] OPA2_P0 ACMP1_P2	
		PB0	ADCIN[4] ACMP0_P0 ACMP1_P1	ADCIN[8] OPA1_N0 ACMP1_P3 ACMP1_N1 ACMP0_N4	
		PB4	ADCIN[4] ACMP0_P0 ACMP1_P1	-	
ACMP1_P2	ACMP1 正端 IO 输入 2	PA5	ADCIN[4] ACMP0_P0 ACMP1_P1	ADCIN[5] OPA2_P0 ACMP1_P2	
		PB3	ADCIN[5] OPA2_P0 ACMP1_P2	-	



ACMP1_P3	ACMP1 正端 IO 输入 3	PB0	ADCIN[4] ACMP0_P0 ACMP1_P1	ADCIN[8] OPA1_N0 ACMP1_P3 ACMP1_N1 ACMP0_N4	
		PB7	ADCIN[8] OPA1_N0 ACMP1_P3 ACMP1_N1 ACMP0_N4	-	
ACMP1_P4	ACMP1 正端 IO 输入 4	PA6	ADCIN[6] ACMP0_P1 ACMP1_N4	OPA0_N1 OPA1_N1 OPA2_N1 ACMP1_P4	
		PA12	OPA0_N1 OPA1_N1 OPA2_N1 ACMP1_P4	ADCIN[13] OSCL_IN	
ACMP1_P5	ACMP1 正端 IO 输入 5	PA2	ADCIN[13]	ADCIN[2] OPA2_O ACMP1_P5	
		PB1	ADCIN[9] ACMP0_N3 ACMP1_P0 OPA2_OX	ADCIN[2] OPA2_O ACMP1_P5	
ACMP1_P6	ACMP1 正端 IO 输入 6	PA7	ADCIN[7] OPA1_P0 ACMP0_P2	ADCIN[12] OPA1_O ACMP1_P6	
ACMP1_P7	ACMP1 正端 IO 输入 7	PA8	ADCIN[10] ACMP0_P3 OPA0_OX ELVI	ADCIN[11] OPA0_O ACMP1_P7	
ACMP1_N0	ACMP1 负端 IO 输入 0	PA3	ADCIN[3] ACMP0_P6 ACMP0_N2 ACMP1_N0	OPA0_P1	在单次比较模式下通过 ACMP1_MUXCR.ACMP_N_SEL 选择对应 IO 输入； 在序列模式下，可以直接通过 ACMP1_MUXCSR.CHNLSEL0~3 选择轮询的 IO 通道
ACMP1_N1	ACMP1 负端 IO 输入 1	PB0	ADCIN[4] ACMP0_P0 ACMP1_P1	ADCIN[8] OPA1_N0 ACMP1_P3 ACMP1_N1 ACMP0_N4	
		PB7	ADCIN[8] OPA1_N0 ACMP1_P3	-	



			ACMP1_N1 ACMP0_N4		
ACMP1_N2	ACMP1 负端 IO 输入 2	PA11	OPA1_P1 ACMP0_P7 ACMP1_N2	ADCIN[14] OSCL_OUT	
ACMP1_N3	ACMP1 负端 IO 输入 3	PA1	ADCIN[1] ACMP0_P5 ACMP0_N1 OPA1_OX	OPA0_N0 OPA2_N0 ACMP1_N3	
		PA4	ADCIN[4] ACMP0_P0 ACMP1_P1	OPA0_N0 OPA2_N0 ACMP1_N3	
		PA9	OPA0_N0 OPA2_N0 ACMP1_N3	ADCIN[14]	
ACMP1_N4	ACMP1 负端 IO 输入 4	PA6	ADCIN[6] ACMP0_P1 ACMP1_N4	OPA0_N1 OPA1_N1 OPA2_N1 ACMP1_P4	
		PB5	ADCIN[6] ACMP0_P1 ACMP1_N4	ADCIN[13]	
ACMP1_out	ACMP1 输出	PA3	AF6		需要配置对应 IO 的 AF 功能
		PA4	AF6		
		PA7	AF3		
		PA13	AF6		

11.2.2.3 工作模式

比较器 ACMP0/1 支持软件或硬件触发比较，通过 ACMPx_MUXCSR.TRIG_MODE 配置。当选择软件触发模式时（ACMPx_MUXCSR.TRIG_MODE=0），可通过软件写 ACMPx_MUXCSR.CMP_SEQ_START=1 触发 ACMP 比较。当选择硬件触发模式时（ACMPx_MUXCSR.TRIG_MODE=1），可选 2 组不同的触发源，每组中的触发属性可单独配置（来源、极性和边沿），具体请参考 ACMP 互联配置寄存器 SysCtrl_EDU_CFG16。此外，硬件触发可以通过 ACMPx_EXTCFG.EXTDTG 配置触发时延，即触发信号会延迟一定的时间后到达 ACMP。

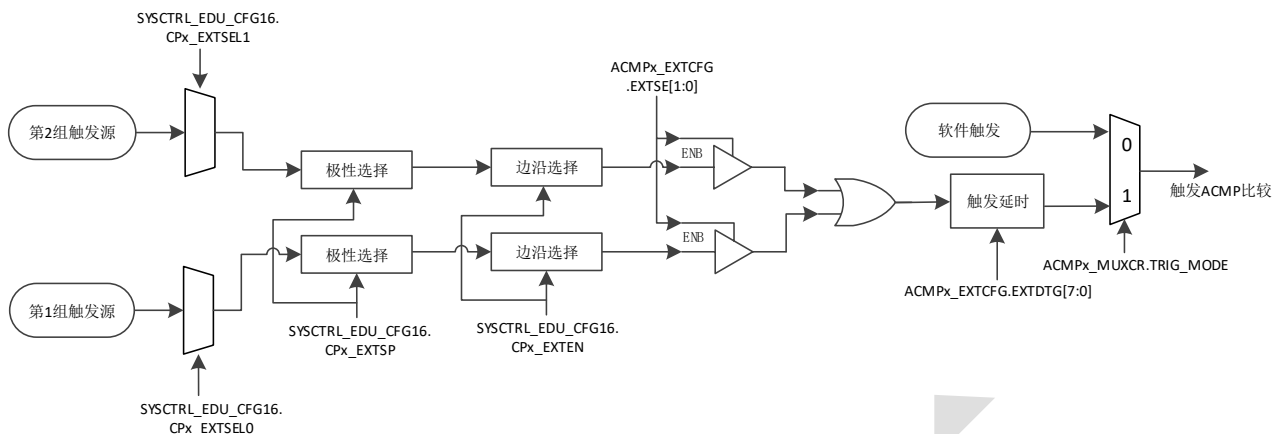


图 11-7 ACMP 触发示意图

根据触发后 ACMP 不同的比较行为，可以通过 ACMPx_MUXCSR.CMP_MODE 配置 ACMP 的 4 种工作模式：单次比较模式、单次数列模式、断续序列模式和连续序列模式。

单次比较模式：通过配置相应 ACMP 寄存器的正负端通道选择位来选择需要被比较的两路输入通道，直接比较后输出。单次比较模式是默认的工作模式，不需要触发也能正常工作。

单次数列模式：配置 ACMP 正端或负端的某一通道为固定比较通道，可以配置另一端的某些通道（正端可轮询 ACMPx_PO~7，负端可轮询 ACMPx_NO~4、HALL_MID 输出、DACBUF_OUT 输出和 CVREF）为轮询比较通道（最多可配置 4 个）。当有一个内部或外部触发到来时，ACMP 会按照可配置的时间间隔（ACMPx_MUXCSR.CHNL_CNT_CFG），按照 ACMPx_MUXCSR.CHNLSEL0~3 的顺序将配置的轮询比较通道依次与固定通道进行比较，当轮询通道都比较完毕后，ACMP 停止工作，将比较结果保存至寄存器 ACMPx_MUXCSR.CPOUT_SEQ 的相应位。

举例说明，ACMP0 选择正端通道固定为通道 0，负端通道 4/2/7/3 依次发起轮询比较，通过软件触发，基本配置如下：

- 配置 ACMP0_MUXCSR.CMP_MODE=1，配置 ACMP 为单次数列模式，ACMP0_MUXCSR.TRIG_MODE=0，默认为软件触发
- 配置 ACMP0_CHNLSEL.POLL_PS=1，选择正端为固定通道，负端轮询比较
- 配置 ACMP0_MUXCSR.ACMP_P_SEL1=0，ACMP0_MUXCSR.ACMP_P_SEL0=0，选择正端为 P0 通道
- 配置 ACMP0_CHNLSEL.CHNL_NUM=3，负端有 4 个通道进行轮询
- 配置 ACMP0_CHNLSEL.CHNLSEL0=4，ACMP0_CHNLSEL.CHNLSEL1=2，ACMP0_CHNLSEL.CHNLSEL1=7，ACMP0_CHNLSEL.CHNLSEL3=3，定义负端通道 4/2/7/3 依次发起轮询比较

配置 ACMP0_CSR.CP_EN=1 后，写入 ACMP0_MUXCSR.CMP_SEQ_START=1，即可实现软件触发单次数列比较，相应的寄存器配置和单次数列时序如图 11-8 所示。比较结束后，ACMP0_MUXCSR.CPOUT_SEQ=0x90，只有 CPOUT_SEQ[4]、CPOUT_SEQ[2]、CPOUT_SEQ[7]和 CPOUT_SEQ[3]位实际生效。

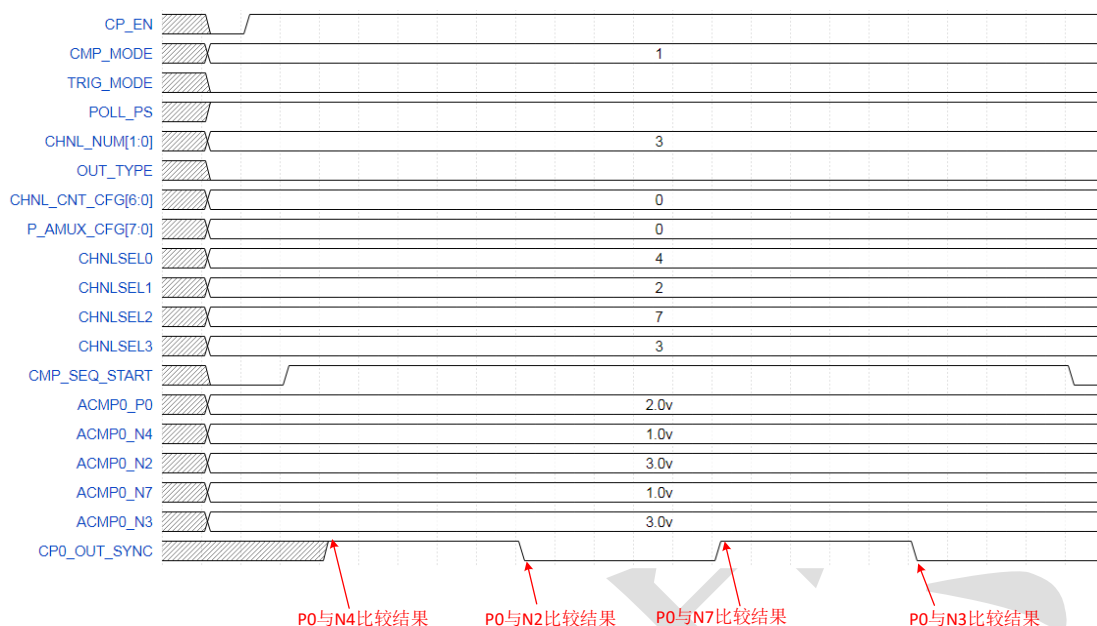


图 11-8 ACMP0 单次序列模式时序图

断续序列模式：配置 ACMP 正端或负端的某一通道为固定比较通道，可以配置另一端的某些通道（正端可轮询 ACMPx_P0~7，负端可轮询 ACMPx_N0~4、HALL_MID 输出、DACBUF_OUT 输出和 CVREF）为轮询比较通道（最多可配置 4 个）。当内部或外部触发到来时，ACMP 会按照可配置的时间间隔（ACMPx_MUXCSR.CHNL_CNT_CFG），按照 ACMPx_MUXCSR.CHNLSEL0~3 的顺序将轮询比较通道与固定通道进行比较。断续序列模式与单次序列模式的差别是，断续序列模式下每个触发只会触发一次比较，如果配置了 4 个轮询比较通道，则需要 4 个触发才能完成比较过程。当轮询通道都比较完毕后，ACMP 停止工作，将比较结果保存至寄存器 ACMPx_MUXCSR.CPOUT_SEQ 的相应位。

举例说明，ACMP0 选择负端通道固定为通道 1，正端通道 3/5/2/6 依次发起轮询比较，通过硬件触发，基本配置如下：

- 配置 ACMP0_MUXCSR.CMP_MODE=2，配置 ACMP 为断续序列模式
- 配置 ACMP0_MUXCSR.TRIG_MODE=1，为硬件触发
- 配置 ACMP0_EXTCFG.EXTSE=1，其他硬件触发配置为默认，即使用 TIM1_TRIG_OC1 上升沿触发 ACMP0
- 配置 ACMP0_CHNLSEL.POLL_PS=0，选择负端为固定通道，正端轮询比较
- 配置 ACMP0_MUXCSR.ACMP_N_SEL=1，选择负端为 P1 通道
- 配置 ACMP0_CHNLSEL.CHNL_NUM=3，正端有 4 个通道进行轮询
- 配置 ACMP0_CHNLSEL.CHNLSEL0=3，ACMP0_CHNLSEL.CHNLSEL1=5，ACMP0_CHNLSEL.CHNLSEL2=2，ACMP0_CHNLSEL.CHNLSEL3=6，定义正端通道 3/5/2/6 依次发起轮询比较

配置 ACMP0_CSR.CP_EN=1 后，每当 TIM1_TRIG_OC1 上升沿产生时，即可实现硬件触发断续序列比较，相应的寄存器配置和断续序列时序如图 11-9 所示。比较结束后，ACMP0_MUXCSR.CPOUT_SEQ =0x0C，只有 CPOUT_SEQ[3]、CPOUT_SEQ[5]、CPOUT_SEQ[2]和 CPOUT_SEQ[6]位实际生效。值得注意的是，断续序列模式下不支持软件触发。

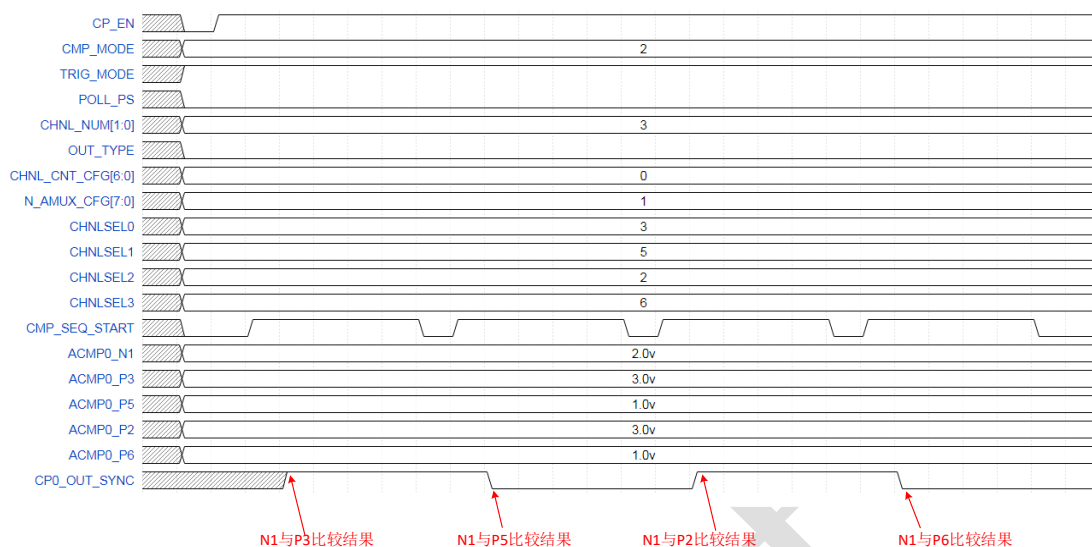


图 11-9 ACMP0 断续序列模式时序图

连续序列模式：配置 ACMP 正端或负端的某一通道为固定比较通道，可以配置另一端的某些通道（正端可轮询 ACMPx_P0~7，负端可轮询 ACMPx_N0~4、HALL_MID 输出、DACBUF_OUT 输出和 CVREF）为轮询比较通道（最多可配置 4 个）。当有一个内部或外部触发到来时，ACMP 会按照可配置的时间间隔（ACMPx_MUXCSR.CHNL_CNT_CFG），按照 ACMPx_MUXCSR.CHNLSEL0~3 的顺序将轮询比较通道与固定通道进行比较。连续序列模式与单次序列模式的差别是，连续序列模式在轮询通道都比较完毕后，并不会停止 ACMP 工作，而是重新发起轮询比较，持续将比较结果保存至寄存器 ACMPx_MUXCSR.CPOUT_SEQ 的相应位。

举例说明，ACMP1 选择正端通道固定为通道 0，负端通道 5/1 依次发起轮询比较，通过软件触发，基本配置如下：

- 配置 ACMP1_MUXCSR.CMP_MODE=3，配置 ACMP 为连续序列模式，ACMP0_MUXCSR.TRIG_MODE=0，默认为软件触发
- 配置 ACMP1_CHNLSEL.POLL_PS=1，选择正端为固定通道，负端轮询比较
- 配置 ACMP1_MUXCSR.ACMP_P_SEL1=0，ACMP1_MUXCSR.ACMP_P_SEL0=0，选择正端为 P0 通道
- 配置 ACMP1_CHNLSEL.CHNL_NUM=1，负端有 2 个通道进行轮询
- 配置 ACMP1_CHNLSEL.CHNLSEL0=5，ACMP1_CHNLSEL.CHNLSEL1=1，定义负端通道 5/1 依次发起轮询比较

配置 ACMP1_CSR.CP_EN=1 后，写入 ACMP1_MUXCSR.CMP_SEQ_START=1，即可实现软件触发连续序列比较，相应的寄存器配置和连续序列时序如图 11-10 所示。图中显示了两轮连续比较的过程，第一轮序列比较完成后，ACMP1_MUXCSR.CPOUT_SEQ =0x22；连续序列模式会继续工作，第一轮序列比较完成后，ACMP1_MUXCSR.CPOUT_SEQ =0x02。连续序列模式会持续进行，只有关闭 ACMP 使能（ACMP1_CSR.CP_EN=0）才能停止连续序列模式的比较。

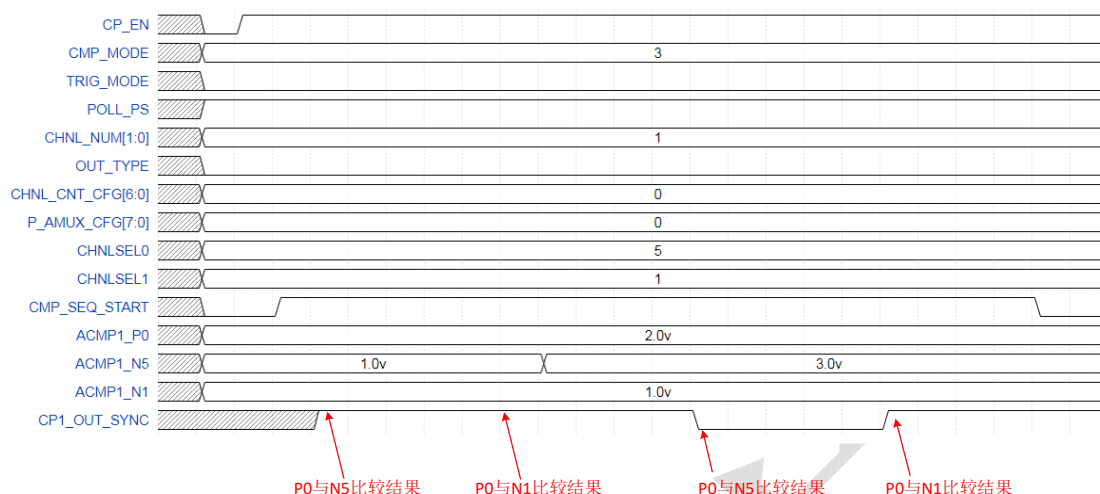


图 11-10 ACMP1 连续序列模式时序图

单次序列模式、断续序列模式和连续序列模式下，允许在 ACMP 工作状态下配置 ACMP 轮询通道选择寄存器 ACMPx_CHNLSEL，配置完成后不会立即切换新配置的比较通道序列，而是在上一个比较通道序列全部比较完成后，再切换到新的比较通道序列开始比较。通过配置轮询比较的通道个数 ACMPx_MUXCSR.CHNL_NUM 和对各个通道 ACMPx_MUXCSR.CHNLSEL0~3 的定义，可以灵活实现不同的比较序列。

ACMP 的结果可以进行数字滤波和消隐。数字滤波时长通过 ACMPx_CSR.FILTER_SEL 来配置。消隐可以通过配置 ACMP 互联配置寄存器 15 (SysCtrl_EDU_CFG15) 来选择消隐控制输入，即在某些内部信号有效的时候，禁止比较器的输出。

此外，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 11-11 所示。

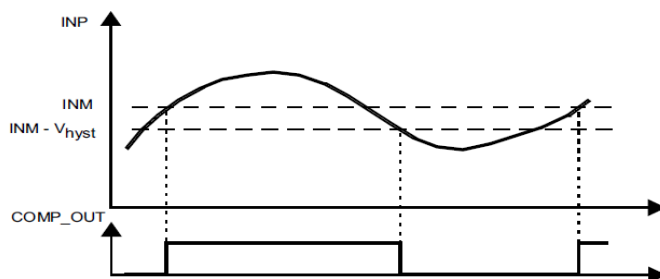


图 11-11 比较器迟滞

ACMP 的迟滞、快速响应、输出极性配置均由模拟电路实现，ACMPx_CSR.CPFS 配置的滤波时长由模拟电路实现。此外，ACMP 最终的输出信号由寄存器 ACMPx_CHNLSEL.CPOUT_CHNL 和 ACMPx_CHNLSEL.OUT_TYPE 配置，可输出实时比较结果或比较序列中某个指定通道的比较结果。ACMP0/1 的输出信号连接到系统 EXTI 线 18/19 上，可产生事件和中断；也可以联动触发 TIMER、ADC 等模块。ACMP 的输出示意图如下：

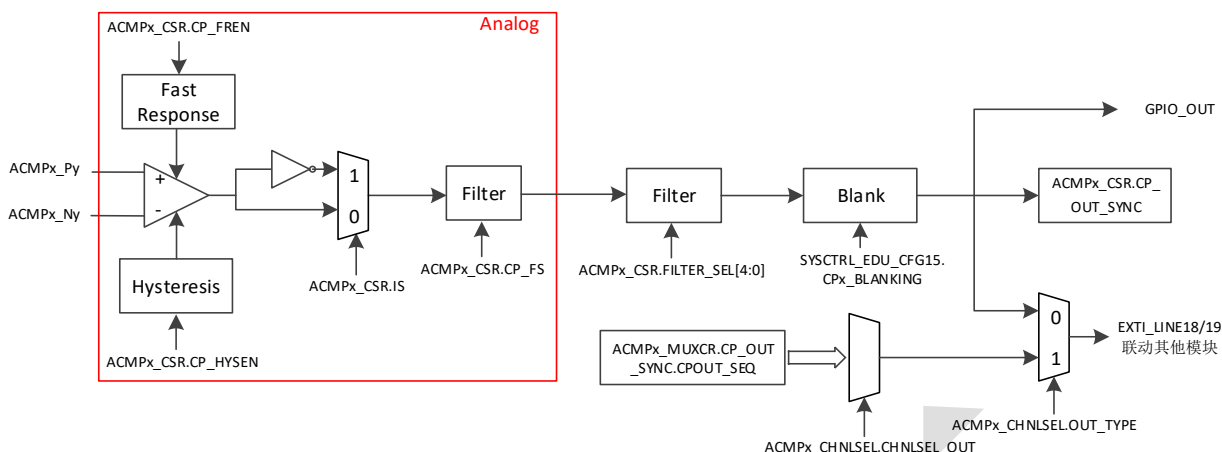


图 11-12 比较器输出示意图

11.2.2.4 CVREF 和校准

比较器的输入端可选择内部电压 CVREF，CVREF 来自 ACMP 内部分压电阻串模块，CVREF 的参考电压 ACMP_VREF 可选为 ADC 参考电压或 VDDA，具体配置方式请参考 ANA_CTRL 寄存器。ACMP 内部分压电阻串模块结构如下：

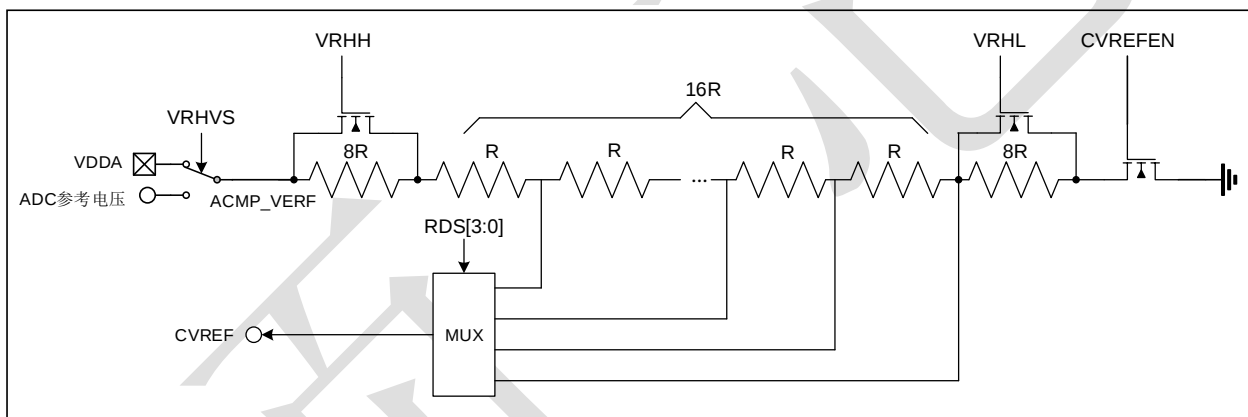


图 11-13 ACMP 内部分压电阻串模块结构图

比较器支持校准功能，通过设置寄存器 ACMPx_CSR.PF 进行校准，默认值 5'b10000，默认表示比较器无偏差。ACMPx_CSR.PF 大于 5'b10000，表示比较器负向端增大正向端减小，最大可达 5'b11111；ACMPx_CSR.PF 小于 5'b10000，表示比较器正向端增大负向端减小，最小可达 5'b00000。ACMPx_CSR.PF 在系统复位后，由软件从校准字节中加载，用户可以在初始化过程中检查是否加载成功，或者根据自身的应用要求，重新校准比较器偏差。

11.2.2.5 比较器中断

比较器可以在系统中产生两类中断：输出中断和序列结束中断，它们均可用于系统低功耗模式的唤醒。比较器输出连接在系统 EXT_I 线上，可以产生事件和中断，EXT_I 线 18 连接到 ACMP0 的输出，EXT_I 线 19 连接到 ACMP1 的输出，它们以逻辑或的关系对应系统 EXT_I Line[19:18]（ACMP0/1 输出）中断向量。比较器在单次序列模式或断续序列模式下，可以产生序列比较结束的中断标志，在相应中断使能的情况下产生事件和中断，EXT_I 线 20 连接到 ACMP0 的序列结束事件，EXT_I 线 21 连接到 ACMP1 的序列结束事件，它们以逻辑或的关系对应系统 EXT_I Line[21:20]（ACMP0/1）中断向量。

11.2.3 HALL_MID 功能描述

LCM32F06X 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。如图 11-14 所示，HALL_IP1、HALL_IP2 和 HALL_IP3 为 HALL_MID 的三路平均信号，分别通过 HALL_CR.HALL_SW1/2/3 配置导通，其中导通的信号在做平均后可以输出到 PMU_AOUT 进行观测，也可以作为比较器的输入。

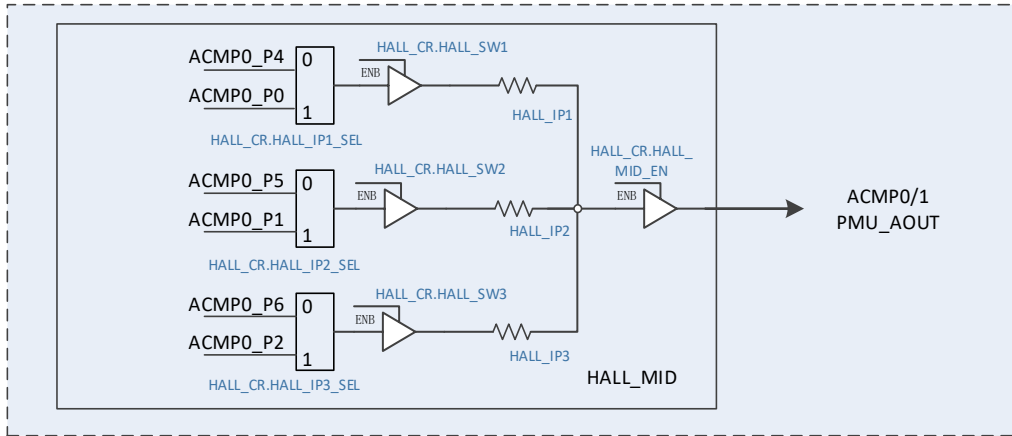


图 11-14 HALL_MID 框图

HALL_IP1、HALL_IP2 和 HALL_IP3 的来源可以分别来自 ACMP0 不同的正端输入，具体如下表。HALL_CR.HALL_MID_EN 为 HALL_MID 输出控制使能。

表 11-4 HALL_MID 端口复用表

名称	管脚类型	复用 I/O 口		备注
HALL_IP1	I	ACMP0_P0	PA4_YA1/PA5_YA1/ PB0_YA1/PB4_YA1	HALL_CR.HALL_IP1_SEL=1'b1
	I	ACMP0_P4	PA0_YA1/PA13_YA1	HALL_CR.HALL_IP1_SEL=1'b0
HALL_IP2	I	ACMP0_P1	PA6_YA1/PB5_YA1	HALL_CR.HALL_IP2_SEL=1'b1
	I	ACMP0_P5	PA1_YA1/PA15_YA1	HALL_CR.HALL_IP2_SEL=1'b0
HALL_IP3	I	ACMP0_P2	PA7_YA1/PB6_YA1	HALL_CR.HALL_IP3_SEL=1'b1
	I	ACMP0_P6	PA3_YA1	HALL_CR.HALL_IP3_SEL=1'b0
HALL_MID	O	-		可输出到 PMU_AOUT 和 ACMP0/1 的正端和负端

11.3 DAC/ACMP 寄存器描述

ACMP 和 DAC 属于总线 APB0，共用同一个地址偏移空间 0x4001_7C00。

表 11-5 DAC/ACMP 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	0x0008_0000	0x4001_7C20
ACMP1_CSR	ACMP1 控制及状态寄存器	R/W	0x0008_0000	0x4001_7C24
DAC_CR	DAC 控制寄存器	R/W	0x0000_0000	0x4001_7C40
DAC_SR	DAC 状态寄存器	R/W	0x0000_0000	0x4001_7C44



DAC_DHR	DAC 数据保持寄存器	R/W	0x0000_0000	0x4001_7C48
DAC_DOR	DAC 数据转换输出寄存器	R	0x0000_0000	0x4001_7C4C
ACMP0_MUXCSR	ACMP0 多路选择控制器	R/W	0x0000_0000	0x4001_7CC0
ACMP0_EXTCFG	ACMP0 硬件触发寄存器	R/W	0x0000_0000	0x4001_7CC4
ACMP1_MUXCSR	ACMP1 多路选择控制器	R/W	0x0000_0000	0x4001_7CC8
ACMP1_EXTCFG	ACMP1 硬件触发寄存器	R/W	0x0000_0000	0x4001_7CCC
ACMP0_CHNLSEL	ACMP0 轮询通道选择寄存器	R/W	0x0000_0000	0x4001_7CD8
ACMP1_CHNLSEL	ACMP1 轮询通道选择寄存器	R/W	0x0000_0000	0x4001_7CDC
HALL_CR	HALL_MID 控制寄存器	R/W	0x0000_0000	0x4001_7CF4
ANA_CTRL	模拟控制及状态寄存器	R/W	0x0000_0000	0x4001_7CF8
SysCtrl_EDU_CFG15	ACMP 互联配置寄存器 15	R/W	0x0000_0000	0x4800_705C
SysCtrl_EDU_CFG16	ACMP 互联配置寄存器 16	R/W	0x0000_0000	0x4800_7060
SysCtrl_EDU_CFG17	DAC 互联配置寄存器 17	R/W	0x0000_0000	0x4800_7064

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：0x0008_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.			CP_OUT_SYNC	FILTER_SEL[4:0]						Res.	MISS_TRIG	CP_PF[4:0]			
			r	rw							rw	rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CP_LPS	Res.			CP_FS	CP_DLY[1:0]		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 : ACMP0 比较结果同步输出（受 CP_IS 影响）
Bits	27:23	FILTER_SEL[4:0] : ACMP0 数字滤波挡位选择，这里按照 PCLK0 配置为 48MHz 来计算滤波时间，如果配置为其他频率则按比例换算
		00000: 0us（默认）
		00001: 2us
		00010: 4us
		...
		11111: 62us
Bit	22	保留，必须保持复位值
Bit	21	MISS_TRIG : 外部触发遗漏标志位，软件写 1 清零
		0: 在单次序列模式或断序序列模式工作过程中没有未响应的外部触发（默认）
		1: 在单次序列模式或断序序列模式工作过程中有未响应的外部触发
Bits	20:16	CP_PF[4:0] : ACMP0 偏移校准值，必须先软件写入 CP_PF 进行校准后才能启动比较器

	00000: 最大
	...
	10000: 0mV (默认)
	...
	11111: 最小
Bit 15	CP_LPS: ACMP0 低功耗选择
	0: ACMP0 非低功耗模式 (默认)
	1: ACMP0 低功耗模式
Bit 11	CP_FS: ACMP0 输出滤波使能
	0: 禁止 (默认)
	1: 使能
Bits 14:12	保留, 必须保持复位值
Bits 10:9	CP_DLY[1:0]: ACMP0 输出延迟
	00: 20ns (默认)
	01: 100ns
	10: 1us
	11: 2us
Bits 8:4	保留, 必须保持复位值
Bit 3	CP_IS: ACMP0 输出极性选择
	0: ACMP0 正相输出 (默认)
	1: ACMP0 反相输出
Bit 2	CP_HYSEN: ACMP0 迟滞使能
	0: 禁用迟滞 (默认)
	1: 使能迟滞, 迟滞电压约为 20mv
Bit 1	CP_FREN: ACMP0 快速响应使能
	0: 禁止快速响应 (默认)
	1: 使能快速响应
Bit 0	CP_EN: ACMP0 模块使能 (写入 CP_PF 后才允许使能)
	0: 禁用 (默认)
	1: 使能

11.3.2 ACMP0 多路选择控制寄存器 (ACMP0_MUXCSR)

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
TRIG_MODE	CMP_INT_FLG	CMP_INTEN	CMP_MODE	ACMP_N_SEL[2:0]						ACMP_P_SEL1[1:0]		ACMP_P_SEL0[2:0]			
rw	rw	rw	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[6:0]						CMP_SEQ_START	
rw								rw						rw	

Bit	31	TRIG_MODE: ACMP0 触发模式选择
		0: 软件触发模式（默认）
		1: 外部硬件触发模式
Bit	30	CMP_INT_FLG: ACMP0 单次序列模式或断续序列模式下，序列比较结束的中断标志位，在 CMP_INTEN 使能的前提下，产生 ACMP0 中断，写 0 清零
		0: 未发生中断（默认）
		1: 发生中断
Bit	29	CMP_INTEN: ACMP0 单次序列模式或断续序列模式下，序列比较结束中断使能位
		0: 禁止（默认）
		1: 使能
Bits	28:27	CMP_MODE: ACMP0 工作模式选择
		00: 单次比较（默认）
		01: 单次序列
		10: 断续序列
		11: 连续序列
Bits	26:24	ACMP_N_SEL[2:0]: ACMP0 负端通道选择
		000: ACMP0_N0（默认）
		001: ACMP0_N1
		010: ACMP0_N2
		011: ACMP0_N3
		100: ACMP0_N4
		101: HALL_MID 输出
		110: DACBUF_OUT 输出
		111: CVREF
Bits	23:21	保留，必须保持复位值
Bits	20:19	ACMP_P_SEL1[1:0]: ACMP0 正端通道内部选择
		00: ACMP0 外部端口输入（默认）
		01: HALL_MID 输出
		10: DACBUF_OUT 输出
		11: CVREF
Bits	18:16	ACMP_P_SEL0[2:0]: ACMP0 正端 IO 通道选择
		000: ACMP0_P0（默认）
		001: ACMP0_P1
		010: ACMP0_P2
		011: ACMP0_P3
		100: ACMP0_P4
		101: ACMP0_P5
		110: ACMP0_P6
		111: ACMP0_P7
Bits	15:8	CPOUT_SEQ[7:0]: 三种序列模式下不同通道的比较值，对应的轮询通道的比较结果会放入 CPOUT_SEQ 的对应位。默认值为 0。
Bits	7:1	CHNL_CNT_CFG[6:0]: 三种序列模式下通道间切换时钟周期数选择
		0000000/0000001: 通道间切换间隔 4 个 PCLK0 时钟周期（默认）

	0000010: 通道间切换间隔 8 个 PCLK0 时钟周期
	0000011: 通道间切换间隔 12 个 PCLK0 时钟周期
	...
	1111111: 通道间切换间隔 512 个 PCLK0 时钟周期
Bit 0	CMP_SEQ_START: 三种序列比较模式触发位
	软件模式下 (TRIG_MODE=0), 写 1 触发比较, 比较结束后硬件清 0, 若需重复比较, 需重新置 1; 硬件模式下 (TRIG_MODE=1), 该位由硬件自动设置, 比较结束后硬件清 0。
	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

11.3.3 ACMP0 硬件触发寄存器 (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.										
			rw												

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

Bits 31:29	保留, 必须保持复位值
Bits 28:27	EXTSE[1:0]: ACMP0 两组触发源使能控制, 具体的触发信息请参考 ACMP 互联配置寄存器 16 (SysCtrl_EDU_CFG16)
	00: 禁止相应触发源触发 ACMP0 (默认)
	x1: 使能 EXTSELG0
	1x: 使能 EXTSELG1
Bits 26:12	保留, 必须保持复位值
Bits 11:4	EXTDTG[7:0]: ACMP0 触发时延
	延迟模式选择 EXTDTG[7:5]: T_d (延迟时间) 计算公式
	0xx: $T_d = \text{EXTDTG}[6:0] * \text{PCLK0}$ (默认)
	10x: $T_d = (\text{EXTDTG}[5:0] + 64) * 2 * \text{PCLK0}$
	110: $T_d = (\text{EXTDTG}[4:0] + 32) * 8 * \text{PCLK0}$
	111: $T_d = (\text{EXTDTG}[4:0] + 32) * 16 * \text{PCLK0}$
Bits 3:0	保留, 必须保持复位值

11.3.4 ACMP0 轮询通道选择寄存器 (ACMP0_CHNLSEL)

地址偏移: 0xD8

复位值: 0x0000_0000

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



Res.			POLL_PS	Res.		CHNL_NUM[1:0]	OUT_TYPE	CPOUT_CHNL[2:0]	Res.			
			rw			rw	rw	rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		CHNLSEL3[2:0]			Res.	CHNLSEL2[2:0]			Res.	CHNLSEL1[2:0]			Res.	CHNLSEL0[2:0]	
		rw				rw				rw				rw	

Bits	31:29	保留，必须保持复位值
Bit	28	POLL_PS : ACMP0 三种序列模式下通道轮询的正负端选择
		0: 负端为固定通道，正端轮询比较（默认）
		1: 正端为固定通道，负端轮询比较
Bits	27:26	保留，必须保持复位值
Bits	25:24	CHNL_NUM[1:0] : ACMP0 轮询比较通道数选择，表示在三种序列模式下依次比较的通道个数
		00: 1 个通道轮询（默认）
		01: 2 个通道轮询
		10: 3 个通道轮询
		11: 4 个通道轮询
Bit	23	OUT_TYPE : ACMP0 对外输出结果选择，与 CPOUT_CHNL 结合使用
		0: 输出正常的比较结果（默认）
		1: 输出 CPOUT_CHNL 选定的通道比较结果
Bits	22:20	CPOUT_CHNL[2:0] : 选择输出至外部的通道结果，与 CPOUT_SEQ[7:0]结合使用
		000: 输出 CPOUT_SEQ[0]结果
		001: 输出 CPOUT_SEQ[1]结果
		...
		111: 输出 CPOUT_SEQ[7]结果
Bits	19:15	保留，必须保持复位值
Bits	14:12	CHNLSEL3[2:0] : ACMP0 轮询比较通道位置 3，配置在三种序列模式下依次比较的第 4 个比较的通道，与 POLL_PS 结合使用（默认为 0）
Bit	11	保留，必须保持复位值
Bits	10:8	CHNLSEL2[2:0] : ACMP0 轮询比较通道位置 2，配置在三种序列模式下依次比较的第 3 个比较的通道，与 POLL_PS 结合使用（默认为 0）
Bit	17	保留，必须保持复位值
Bits	6:4	CHNLSEL1[2:0] : ACMP0 轮询比较通道位置 1，配置在三种序列模式下依次比较的第 2 个比较的通道，与 POLL_PS 结合使用（默认为 0）
Bit	13	保留，必须保持复位值
Bits	2:0	CHNLSEL0[2:0] : ACMP0 轮询比较通道位置 0，配置在三种序列模式下依次比较的第 1 个比较的通道，与 POLL_PS 结合使用（默认为 0）

11.3.5 ACMP1 控制及状态寄存器（ACMP1_CSR）

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.			CP_OUT_SYNC	FILTER_SEL[4:0]					Res.	MISS_TRIG	CP_PF[4:0]				
			r	rw						rw	rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CP_LPS	Res.			CP_FS	CP_DLY[1:0]	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 : ACMP1 比较结果同步输出（受 CP_IS 影响）
Bits	27:23	FILTER_SEL[4:0] : ACMP1 数字滤波挡位选择，这里按照 PCLK0 配置为 48MHz 来计算滤波时间，如果配置为其他频率则按比例换算
		00000: 0us（默认）
		00001: 2us
		00010: 4us
		...
		11111: 62us
Bit	22	保留，必须保持复位值
Bit	21	MISS_TRIG : 外部触发遗漏标志位，软件写 1 清零
		0: 在单次序列模式或断序序列模式工作过程中没有未响应的外部触发（默认）
		1: 在单次序列模式或断序序列模式工作过程中有未响应的外部触发
Bits	20:16	CP_PF[4:0] : ACMP1 偏移校准值，必须先软件写入 CP_PF 进行校准后才能启动比较器
		00000: 最大
		...
		10000: 0mV（默认）
		...
		11111: 最小
Bit	15	CP_LPS : ACMP1 低功耗选择
		0: ACMP1 非低功耗模式（默认）
		1: ACMP1 低功耗模式
Bit	11	CP_FS : ACMP1 输出滤波使能
		0: 禁止（默认）
		1: 使能
Bits	14:12	保留，必须保持复位值
Bits	10:9	CP_DLY[1:0] : ACMP1 输出延迟
		00: 20ns（默认）
		01: 100ns
		10: 1us
		11: 2us
Bits	8:4	保留，必须保持复位值
Bit	3	CP_IS : ACMP1 输出极性选择
		0: ACMP1 正相输出（默认）

	1: ACMP1 反相输出
Bit 2	CP_HYSEN: ACMP1 迟滞使能
	0: 禁用迟滞 (默认)
	1: 使能迟滞, 迟滞电压约为 20mv
Bit 1	CP_FREN: ACMP1 快速响应使能
	0: 禁止快速响应 (默认)
	1: 使能快速响应
Bit 0	CP_EN: ACMP1 模块使能 (写入 CP_PF 后才允许使能)
	0: 禁用 (默认)
	1: 使能

11.3.6 ACMP1 多路选择控制寄存器 (ACMP1_MUXCSR)

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
TRIG_MODE	CMP_INT_FLG	CMP_INTEN	CMP_MODE	ACMP_N_SEL[2:0]							ACMP_P_SEL1[1:0]		ACMP_P_SEL0[2:0]		
rw	rw	rw	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[6:0]						CMP_SEQ_START	
rw								rw						rw	

Bit 31	TRIG_MODE: ACMP1 触发模式选择
	0: 软件触发模式 (默认)
	1: 外部硬件触发模式
Bit 30	CMP_INT_FLG: ACMP1 单次序列模式或断续序列模式下, 序列比较结束的中断标志位, 在 CMP_INTEN 使能的前提下, 产生 ACMP1 中断, 写 0 清零
	0: 未发生中断 (默认)
	1: 发生中断
Bit 29	CMP_INTEN: ACMP1 单次序列模式或断续序列模式下, 序列比较结束中断使能位
	0: 禁止 (默认)
	1: 使能
Bits 28:27	CMP_MODE: ACMP1 工作模式选择
	00: 单次比较 (默认)
	01: 单次序列
	10: 断续序列
	11: 连续序列
Bits 26:24	ACMP_N_SEL[2:0]: ACMP1 负端通道选择
	000: ACMP1_N0 (默认)
	001: ACMP1_N1
	010: ACMP1_N2
	011: ACMP1_N3

	100: ACMP1_N4
	101: HALL_MID 输出
	110: DACBUF_OUT 输出
	111: CVREF
Bits 23:21	保留, 必须保持复位值
Bits 20:19	ACMP_P_SEL1[1:0]: ACMP1 正端通道内部选择
	00: ACMP1 外部端口输入 (默认)
	01: HALL_MID 输出
	10: DACBUF_OUT 输出
	11: CVREF
Bits 18:16	ACMP_P_SEL0[2:0]: ACMP1 正端 IO 通道选择
	000: ACMP1_P0 (默认)
	001: ACMP1_P1
	010: ACMP1_P2
	011: ACMP1_P3
	100: ACMP1_P4
	101: ACMP1_P5
	110: ACMP1_P6
	111: ACMP1_P7
Bits 15:8	CPOUT_SEQ[7:0]: 三种序列模式下不同通道的比较值, 对应的轮询通道的比较结果会放入 CPOUT_SEQ 的对应位。默认值为 0。
Bits 7:1	CHNL_CNT_CFG[6:0]: 连续采样通道间切换时钟周期数选择
	0000000/0000001: 通道间切换间隔 4 个 PCLK0 时钟周期 (默认)
	0000010: 通道间切换间隔 8 个 PCLK0 时钟周期
	0000011: 通道间切换间隔 12 个 PCLK0 时钟周期
	...
	1111111: 通道间切换间隔 512 个 PCLK0 时钟周期
Bit 0	CMP_SEQ_START: 三种序列比较模式触发位
	软件模式下 (TRIG_MODE=0), 写 1 触发比较, 比较结束后硬件清 0, 若需重复比较, 需重新置 1; 硬件模式下 (TRIG_MODE=1), 该位由硬件自动设置, 比较结束后硬件清 0。
	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

11.3.7 ACMP1 硬件触发寄存器 (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.									
			rw												

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

Res.				EXTDTG[7:0]				Res.			
				rw							

Bits	31:29	保留，必须保持复位值
Bits	28:27	EXTSE[1:0]: ACMP1 两组触发源使能控制，具体的触发信息请参考 ACMP 互联配置寄存器 16（SysCtrl_EDU_CFG16）
		00: 禁止相应触发源触发 ACMP1（默认）
		x1: 使能 EXTSELG0
		1x: 使能 EXTSELG1
Bits	26:12	保留，必须保持复位值
Bits	11:4	EXTDTG[7:0]: ACMP1 触发时延
		延迟模式选择 EXTDTG[7:5]: T_d （延迟时间）计算公式
		0xx: $T_d = \text{EXTDTG}[6:0] * \text{PCLK0}$ （默认）
		10x: $T_d = (\text{EXTDTG}[5:0] + 64) * 2 * \text{PCLK0}$
		110: $T_d = (\text{EXTDTG}[4:0] + 32) * 8 * \text{PCLK0}$
		111: $T_d = (\text{EXTDTG}[4:0] + 32) * 16 * \text{PCLK0}$
Bits	3:0	保留，必须保持复位值

11.3.8 ACMP1 轮询通道选择寄存器（ACMP1_CHNLSEL）

地址偏移：0xDC

复位值：0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.			POLL_PS	Res.		CHNL_NUM[1:0]		OUT_TYPE	CPOUT_CHNL[2:0]			Res.			
			rw			rw		rw	rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	CHNLSEL3[2:0]			Res.	CHNLSEL2[2:0]			Res.	CHNLSEL1[2:0]			Res.	CHNLSEL0[2:0]		
	rw				rw				rw				rw		

Bits	31:29	保留，必须保持复位值
Bit	28	POLL_PS: ACMP1 三种序列模式下通道轮询的正负端选择
		0: 正端轮询（默认）
		1: 负端轮询
Bits	27:26	保留，必须保持复位值
Bits	25:24	CHNL_NUM[1:0]: ACMP1 轮询比较通道数选择，表示在三种序列模式下依次比较的通道个数
		00: 1 个通道轮询（默认）
		01: 2 个通道轮询
		10: 3 个通道轮询
		11: 4 个通道轮询
Bit	23	OUT_TYPE: ACMP1 对外输出结果选择，与 CPOUT_CHNL 结合使用

	0: 输出正常的比较结果（默认）
	1: 输出 CPOUT_CHNL 选定的通道比较结果
Bits 22:20	CPOUT_CHNL[2:0]: 选择输出至外部的通道结果，与 CPOUT_SEQ[7:0]结合使用
	000: 输出 CPOUT_SEQ[0]结果
	001: 输出 CPOUT_SEQ[1]结果
	...
	111: 输出 CPOUT_SEQ[7]结果
Bits 19:15	保留，必须保持复位值
Bits 14:12	CHNLSEL3[2:0]: ACMP1 轮询比较通道位置 3，配置在三种序列模式下依次比较的第 4 个比较的通道，与 POLL_PS 结合使用（默认为 0）
Bit 11	保留，必须保持复位值
Bits 10:8	CHNLSEL2[2:0]: ACMP1 轮询比较通道位置 2，配置在三种序列模式下依次比较的第 3 个比较的通道，与 POLL_PS 结合使用（默认为 0）
Bit 7	保留，必须保持复位值
Bits 6:4	CHNLSEL1[2:0]: ACMP1 轮询比较通道位置 1，配置在三种序列模式下依次比较的第 2 个比较的通道，与 POLL_PS 结合使用（默认为 0）
Bit 3	保留，必须保持复位值
Bits 2:0	CHNLSEL0[2:0]: ACMP1 轮询比较通道位置 0，配置在三种序列模式下依次比较的第 1 个比较的通道，与 POLL_PS 结合使用（默认为 0）

11.3.9 DAC 控制寄存器（DAC_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]		Res.					EN	
rw	rw	rw	rw	rw			rw								rw

Bits 31:16	保留，必须保持复位值
Bit 15	ALIGN: DAC 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: DAC 参考电压选择
	0: ADC 参考电压（默认）
	1: VDDA
Bit 13	DMAUDRIE: DAC DMA 欠载中断使能位
	0: DAC DMA 欠载中断禁止（默认）
	1: DAC DMA 欠载中断使能
Bit 12	DMAEN: DAC DMA 使能位

	0: DAC DMA 模式禁止（默认）
	1: DAC DMA 模式使能
Bits 11:8	MAMP[3:0]: DAC 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
	1000: 放开序列移位寄存器 LFSR 的[8:0]/三角波形自增幅值为 511
	≥ 1001: 放开序列移位寄存器 LFSR 的[9:0]/三角波形自增幅值为 1023
Bits 7:6	WAVE[1:0]: DAC 噪声或者三角波形发生器使能控制
	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
Bits 5:1	保留，必须保持复位值
Bit 0	EN: DAC 使能位
	0: DAC 禁止（默认）
	1: DAC 使能

11.3.10 DAC 状态寄存器（DAC_SR）

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								rw

Bits 31:9	保留，必须保持复位值
Bit 8	DMAUDR: DAC DMA 欠载标志
	由硬件置位，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: DAC 软件触发位

	软件配置此位，触发一次 DAC 转换（软件触发的条件为 DAC_TSEL 为 0）
	0：无操作（默认）
	1：软件触发
	注： 当触发产生，寄存器 DAC_DHR 被载入到寄存器 DAC_DOR 后，此位被硬件自动清除

11.3.11 DAC 数据保持寄存器（DAC_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.															

DAC_CR.ALIGN 为 0，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									

DAC_CR.ALIGN 为 1，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
DAC_CR.ALIGN 为 0，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]： DAC 数据右对齐
DAC_CR.ALIGN 为 1，左对齐模式	
Bits 15:6	DHR[9:0]： DAC 数据左对齐
Bits 5:0	保留，必须保持复位值

11.3.12 DAC 数据转换输出寄存器（DAC_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.															

DAC_CR.ALIGN 为 0，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									

DAC_CR.ALIGN 为 1，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
DAC_CR.ALIGN 为 0，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]: DAC 数据右对齐
DAC_CR.ALIGN 为 1，左对齐模式	
Bits 15:6	DOR[9:0]: DAC 数据左对齐
Bits 5:0	保留，必须保持复位值

11.3.13 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 ACMP0_P6（默认）
	1: HALL_IP3 通道选择 ACMP0_P2
Bit 2	HALL_IP2_SEL: HALL_IP2 通道选择
	0: HALL_IP2 通道选择 ACMP0_P5（默认）
	1: HALL_IP2 通道选择 ACMP0_P1

Bit 1	HALL_IP1_SEL: HALL_IP1 通道选择
	0: HALL_IP1 通道选择 ACMP0_P4 (默认)
	1: HALL_IP1 通道选择 ACMP0_P0
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 信号允许输出

11.3.14 模拟控制及状态寄存器 (ANA_CTRL)

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
VRHH	VRHL	VRHVS	RDS[3:0]			CVERFEN		Res.							
rw	rw	rw	rw			rw									

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

Bit	31	VRHH: 分压电阻范围选择位, 默认为 0									
Bit	30	VRHL: 分压电阻范围选择位, 默认为 0									
Bit	29	VRHVS: 分压电阻串源电压 V_{in} 选择位									
		0: VDDA (默认)									
		1: ADC 参考电压									
Bits	28:25	RDS[3:0]: CVREF 配置位, 与 VRHH 和 VRHL 配合, 可以实现各种电压, 详见下表									
		VRHH		VRHL		CVREF					
		0		0		$\frac{8 + RDS[3:0]}{32} \times V_{in}$					
		0		1		$\frac{RDS[3:0]}{24} \times V_{in}$					
		1		0		$\frac{8 + RDS[3:0]}{24} \times V_{in}$					
		1		1		$\frac{RDS[3:0]}{16} \times V_{in}$					
Bit	24	CVREFEN: 比较器内部电压 CVREF 使能									
		0: 关闭内部电压 CVREF (默认)									
		1: 开启内部电压 CVREF									
Bits	23:5	保留, 必须保持复位值									
Bit	4	DAC_BUFEN: DACOUT_BUF 输出 BUFF 使能位									
		0: DAC 输出 BUFF 不使能, 输出端 DACOUT_BUF 为悬空状态 (默认)									
		1: DAC 输出 BUFF 使能									
		注: 选取 VDDA 作为 DAC 参考电压时, 该输出电压在接近 DAC 满量程时会不准确。									
Bits	3:0	保留, 必须保持复位值									

11.3.15 ACMP 互联配置寄存器 15 (SysCtrl_EDU_CFG15)

SysCtrl_EDU_CFG15 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 0x05FA659A 到 SysCtrl_KEY 密钥寄存器，且在随后的 64 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。

地址：0x4800_705C

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
CP1_BLKING[7:0]								CP0_BLKING[7:0]							
rw								rw							

Bits 31:16	保留，必须保持复位值
Bits 15:8	CP1_BLANKING[7:0]: ACMP1 消隐源选择
	ACMP1 消隐源可选择为软件/ TIM2_TRIG_OC4/ TIM15_TRIG_OC2/ TIM15_TRIG_OC1 / TIM1_TRIG_OC5 / TIM1_TRIG_OC3 / TIM1_TRIG_OC2/ TIM1_TRIG_OC1 输出（从高位到低位）
	所有消隐源均可被配置为使能或不使能，通过控制位 CP1_BLANKING[x]配置；当 CP1_BLANKING[7]配置为 1 时为软件消隐，会直接触发消隐
	所有使能的触发源通过逻辑或的方式组合
	CP1_BLANKING[x] = 0: 相应消隐源不选择（默认）
	CP1_BLANKING[x] = 1: 相应消隐源选择
Bits 7:0	CP0_BLANKING[7:0]: ACMP0 消隐源选择
	ACMP0 消隐源可选择为软件/ TIM2_TRIG_OC4/ TIM15_TRIG_OC2/ TIM15_TRIG_OC1 / TIM1_TRIG_OC5 / TIM1_TRIG_OC3 / TIM1_TRIG_OC2/ TIM1_TRIG_OC1 输出（从高位到低位）
	所有消隐源均可被配置为使能或不使能，通过控制位 CP0_BLANKING[x]配置；当 CP0_BLANKING[7]配置为 1 时为软件消隐，会直接触发消隐
	所有使能的触发源通过逻辑或的方式组合
	CP0_BLANKING[x] = 0: 相应消隐源不选择（默认）
	CP0_BLANKING[x] = 1: 相应消隐源选择
注： TIMx_TRIG_OCI 和 TIMx_TRGO 不同，详见 TIM 章节说明。	

11.3.16 ACMP 互联配置寄存器 16 (SysCtrl_EDU_CFG16)

SysCtrl_EDU_CFG16 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 0x05FA659A 到 SysCtrl_KEY 密钥寄存器，且在随后的 64 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。

地址：0x4800_7060

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
CP1_EXTSP1	CP1_EXTSP0	CP0_EXTSP1	CP0_EXTSP0	CP1_EXTSEL1 [1:0]	CP1_EXTSEL0 [1:0]	CP0_EXTSEL1 [1:0]	CP0_EXTSEL0 [1:0]	CP1_EXTEN [1:0]	CP0_EXTEN [1:0]						
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	CP1_EXTSP1 : ACMP1 第 2 组触发源极性选择位
		0: 同相输入（默认）
		1: 反相输入
Bit	14	CP1_EXTSP0 : ACMP1 第 1 组触发源极性选择位
		0: 同相输入（默认）
		1: 反相输入
Bit	13	CP0_EXTSP1 : ACMP0 第 2 组触发源极性选择位
		0: 同相输入（默认）
		1: 反相输入
Bit	12	CP0_EXTSP0 : ACMP0 第 1 组触发源极性选择位
		0: 同相输入（默认）
		1: 反相输入
Bits	11:10	CP1_EXTSEL1[1:0] : ACMP1 第 2 组触发源选择
		00: TIM1_TRIG_OC3（默认）
		01: TIM1_TRIG_OC5
		10: TIM16_TRIG_OC1
		11: TIM17_TRIG_OC1
Bits	9:8	CP1_EXTSEL0[1:0] : ACMP1 第 1 组触发源选择
		00: TIM1_TRIG_OC1（默认）
		01: TIM1_TRIG_OC2
		10: TIM2_TRGO
		11: TIM15_TRGO
Bits	7:6	CP0_EXTSEL1[1:0] : ACMP0 第 2 组触发源选择
		00: TIM1_TRIG_OC3（默认）
		01: TIM1_TRIG_OC5
		10: TIM16_TRIG_OC1
		11: TIM17_TRIG_OC1
Bits	5:4	CP0_EXTSEL0[1:0] : ACMP0 第 1 组触发源选择
		00: TIM1_TRIG_OC1（默认）
		01: TIM1_TRIG_OC2
		10: TIM2_TRGO
		11: TIM15_TRGO
Bits	3:2	CP1_EXTEN[1:0] : ACMP1 边沿触发方式
		00: 外部上升沿触发（默认）

	01: 外部下降沿触发
	10: 外部双沿触发
	11: 禁止外部触发, 仅支持软件触发
Bits 1:0	CP0_EXTEN[1:0]: ACMP0 边沿触发方式
	00: 外部上升沿触发 (默认)
	01: 外部下降沿触发
	10: 外部双沿触发
	11: 禁止外部触发, 仅支持软件触发

11.3.17 DAC 互联配置寄存器 17 (SysCtrl_EDU_CFG17)

SysCtrl_EDU_CFG17 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 0x05FA659A 到 SysCtrl_KEY 密钥寄存器, 且在随后的 64 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。

地址: 0x4800_7064

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.								DAC_TSEL[2:0]			Res.		DAC_TEN[1:0]		
									rw					rw	

Bits 31:7	保留, 必须保持复位值
Bits 6:4	DAC_TSEL[2:0]: DAC 触发源选择
	000: 软件触发 (默认)
	001: TIM1_TRGO
	010: TIM2_TRGO
	011: TIM15_TRGO
	100: TIM16_TRIG_OC1
	101: TIM17_TRIG_OC1
	110: EXTI1
	111: EXTI9
Bits 3:2	保留, 必须保持复位值
Bits 1:0	DAC_TEN[1:0]: DAC 触发源使能控制
	00: 外部上升沿触发, 触发事件发生时, DAC_DHR 寄存器的值导入到转换寄存器 DAC_DOR (默认)
	01: 外部下降沿触发, 触发事件发生时, DAC_DHR 寄存器的值导入到转换寄存器 DAC_DOR
	10: 外部双沿触发, 触发事件发生时, DAC_DHR 寄存器的值导入到转换寄存器 DAC_DOR
	11: 禁止外部触发, 写入寄存器 DAC_DHR 的值直接传入转换寄存器 DAC_DOR

注: `TIMx_TRIG_OC1` 和 `TIMx_TRGO` 不同, 详见 `TIM` 章节说明。

12. 运算放大器（OPA）

LCM32F06X 内置 3 个运算放大器，放大倍数为 1/2/4/6/8/10/16/32，同时输出偏置为 ADC 参考电压的一半。

12.1 OPA 主要特性

- 输入共模电压：0V ~ VDDA
- 输出偏置可选为 ADC 参考电压的一半或 PMU_AOUT
- 可调放大倍数：1/2/4/6/8/10/16/32
- 正负端分别有两个输入可选，输入通道切换可与 ADC 进行联动
- 正负端输入可配置下拉到地
- 正负端输入电阻阻值可配

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	I/O 序号	GPIO 模拟通道	
				YA1	YA2
OPA0_P0	I	OPA0 正端输入 0	PA10	OPA0_P0 ADCIN[15]	DACOUT_BUF
OPA0_P1	I	OPA0 正端输入 1	PA3	ADCIN[3] ACMP0_N2 ACMP0_P6 ACMP1_N0	OPA0_P1
OPA0_N0	I	OPA0 负端输入 0	PA1	ADCIN[1] ACMP0_P5 ACMP0_N1 OPA1_OX	OPA0_N0 OPA2_N0 ACMP1_N3
			PA4	ADCIN[4] ACMP0_P0 ACMP1_P1	OPA0_N0 OPA2_N0 ACMP1_N3
			PA9	OPA0_N0 OPA2_N0 ACMP1_N3	ADCIN[14]
OPA0_N1	I	OPA0 负端输入 1	PA6	ADCIN[6] ACMP0_P1 ACMP1_N4	OPA0_N1 OPA1_N1 OPA2_N1 ACMP1_P4
			PA12	OPA0_N1 OPA1_N1 OPA2_N1 ACMP1_P4	ADCIN[13] OSCL_IN
OPA0_O	O	OPA0 输出	PA8	ADCIN[10]	ADCIN[11]

管脚名称	管脚类型	管脚描述	I/O 序号	GPIO 模拟通道	
				YA1	YA2
				ACMP0_P3 OPA0_OX ELVI	OPA0_O ACMP1_P7
OPA0_OX	O	OPA0 带电阻输出	PA8	ADCIN[10] ACMP0_P3 OPA0_OX ELVI	ADCIN[11] OPA0_O ACMP1_P7
			PA14	ADCIN[10] ACMP0_P3 OPA0_OX ELVI	PMU_AOUT
OPA1_P0	I	OPA1 正端输入 0	PA7	ADCIN[7] OPA1_P0 ACMP0_P2	ADCIN[12] OPA1_O ACMP1_P6
			PB6	ADCIN[7] OPA1_P0 ACMP0_P2	ADCIN[9] ACMP0_N3 ACMP1_P0 OPA2_OX
OPA1_P1	I	OPA1 正端输入 1	PA11	OPA1_P1 ACMP0_P7 ACMP1_N2	ADCIN[14] OSCL_OUT
OPA1_N0	I	OPA1 负端输入 0	PB0	ADCIN[4] ACMP0_P0 ACMP1_P1	ADCIN[8] OPA1_N0 ACMP1_P3 ACMP1_N1 ACMP0_N4
			PB7	ADCIN[8] OPA1_N0 ACMP1_P3 ACMP1_N1 ACMP0_N4	-
OPA1_N1	I	OPA1 负端输入 1	PA6	ADCIN[6] ACMP0_P1 ACMP1_N4	OPA0_N1 OPA1_N1 OPA2_N1 ACMP1_P4
			PA12	OPA0_N1 OPA1_N1 OPA2_N1 ACMP1_P4	ADCIN[13] OSCL_IN
OPA1_O	O	OPA1 输出	PA7	ADCIN[7] OPA1_P0	ADCIN[12] OPA1_O

管脚名称	管脚类型	管脚描述	I/O 序号	GPIO 模拟通道	
				YA1	YA2
				ACMP0_P2	ACMP1_P6
OPA1_OX	O	OPA1 带电阻输出	PA1	ADCIN[1] ACMP0_P5 ACMP0_N1 OPA1_OX	OPA0_N0 OPA2_N0 ACMP1_N3
			PA15	ADCIN[1] ACMP0_P5 ACMP0_N1 OPA1_OX	PMU_AMUX
OPA2_P0	I	OPA2 正端输入 0	PA5	ADCIN[4] ACMP0_P0 ACMP1_P1	ADCIN[5] OPA2_P0 ACMP1_P2
			PB3	ADCIN[5] OPA2_P0 ACMP1_P2	-
OPA2_P1	I	OPA2 正端输入 1	PA0	ADCIN[0] ACMP0_P4 ACMP0_N0	OPA2_P1
OPA2_N0	I	OPA2 负端输入 0	PA1	ADCIN[1] ACMP0_P5 ACMP0_N1 OPA1_OX	OPA0_N0 OPA2_N0 ACMP1_N3
			PA4	ADCIN[4] ACMP0_P0 ACMP1_P1	OPA0_N0 OPA2_N0 ACMP1_N3
			PA9	OPA0_N0 OPA2_N0 ACMP1_N3	ADCIN[14]
OPA2_N1	I	OPA2 负端输入 1	PA6	ADCIN[6] ACMP0_P1 ACMP1_N4	OPA0_N1 OPA1_N1 OPA2_N1 ACMP1_P4
			PA12	OPA0_N1 OPA1_N1 OPA2_N1 ACMP1_P4	ADCIN[13] OSCL_IN
OPA2_O	O	OPA2 输出	PA2	ADCIN[13]	ADCIN[2] OPA2_O ACMP1_P5
			PB1	ADCIN[9] ACMP0_N3	ADCIN[2] OPA2_O

管脚名称	管脚类型	管脚描述	I/O 序号	GPIO 模拟通道	
				YA1	YA2
				ACMP1_P0 OPA2_OX	ACMP1_P5
OPA2_OX	O	OPA2 带电阻输出	PB1	ADCIN[9] ACMP0_N3 ACMP1_P0 OPA2_OX	ADCIN[2] OPA2_O ACMP1_P5
			PB6	ADCIN[7] OPA1_P0 ACMP0_P2	ADCIN[9] ACMP0_N3 ACMP1_P0 OPA2_OX

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 ADC 参考电压的一半或 PMU_AOUT。正负端分别有两个输入可选，可配置下拉到地，输入电阻阻值可配。



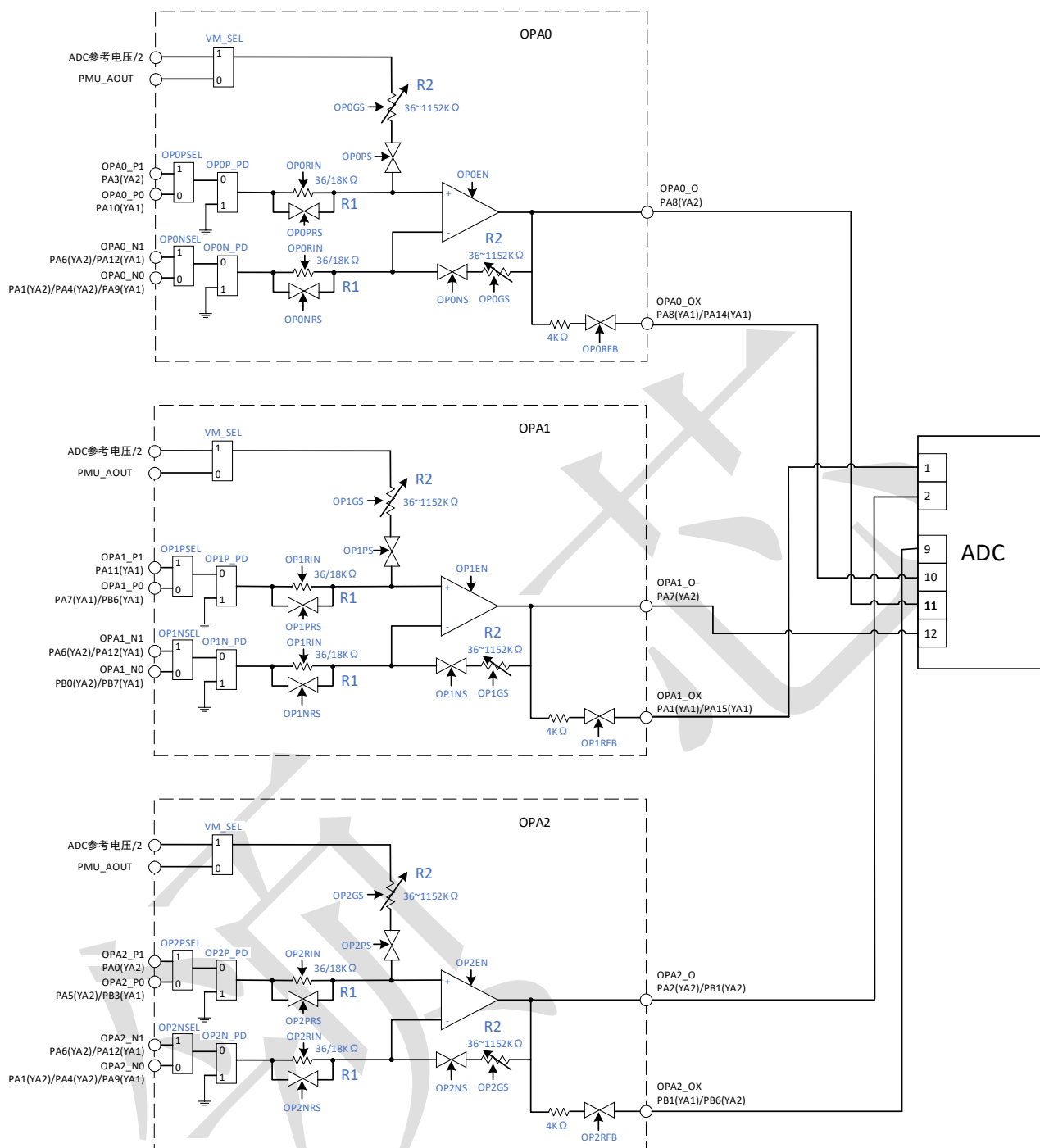


图 12-1 OPA 内部结构和外部互联

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 OPx_P 和负端输入 OPx_N 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 的校准字节中，用户需要软件加载 TRIM 值。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 $OPAx_CSR.OPxNS=0$ ，断开反馈电阻， $OPAx_CSR.OPxPS=0$ ，使运放处于开环无限增益状态。
2. 设置寄存器 $OPAx_CSR.OPxP_PD=1$ ， $OPAx_CSR.OPxN_PD=1$ ，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，

OPAx_CSR.OPxPRS 和 OPAx_CSR.OPxNRS 的值保持一致。

3. 设置寄存器 OPAx_CSR.OPxTRIMEN=1，使能校准过程。

4. 设置寄存器 OPAx_CSR.OPxEN=1，使能运放。

5. 调准 OPAx_CSR.OPxOS[4:0]的值：

- 将 OPxOS 的值逐步从 11111 减小到 00000，判断 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 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。

12.3.2 运放 OPA 与 ADC 联动

运放 OPA0/1/2 支持与 ADC 的联动。如果 ADC 采样 OPA0/1/2 的输出所在的 IO 通道后，可以配置其切换对应 OPA 的输入通道 OPAx_P 和 OPAx_N。OPA0/1/2 与 ADC 的联动基于固定的 ADCIN 通道，OPA0 对应 ADCIN[11]，OPA1 对应 ADCIN[12]，OPA2 对应 ADCIN[2]。

12.3.3 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，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。

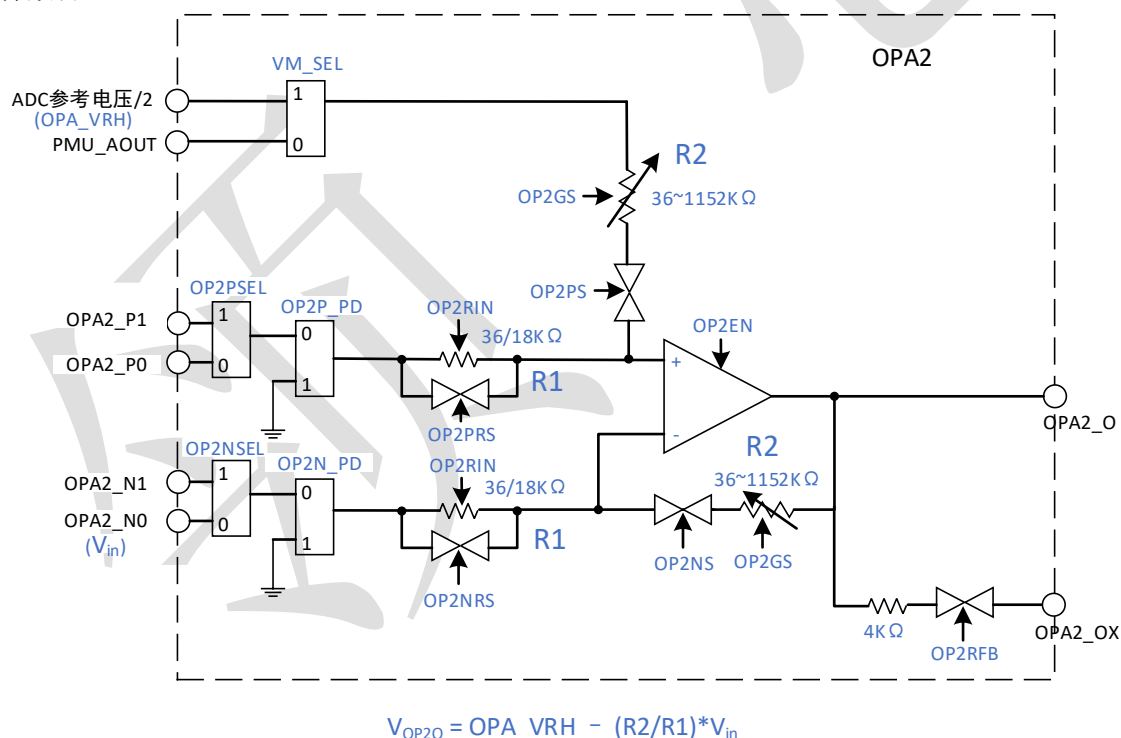


图 12-2 OPA2 典型应用配置

图 12-3 给出了 OPA2 的典型应用方式，内部带 ADC 参考电压一半的偏置和反馈，配置如下：

- OPA2_CSR.VM_SEL = 1
- OPA2_CSR.OP2P_PD = 1，OPA2_CSR.OP2N_PD = 0
- OPA2_CSR.OP2NSEL = 0
- OPA2_CSR.OP2PRS = 0，OPA2_CSR.OP2NRS = 0

- OPA2_CSR.OP2PS = 1, OPA2_CSR.OP2NS = 1

其它配置根据实际的使用情况配置，放大倍数由 OPA2_CSR.OP2GS 所设置的 R2 决定。

12.4 OPA 寄存器描述

表 12-2 OPA 相关寄存器表

名称	说明	读写权限	复位值	字节地址
OPA0_CSR	OPA0 控制状态寄存器	R/W	0x0000_0000	0x4001_7C00
OPA1_CSR	OPA1 控制状态寄存器	R/W	0x0000_0000	0x4001_7C04
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（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
OP0TRIMO	OP0TRIMEN	OP0P_PD	OP0N_PD	Res.	OP0P_SEL	Res.	OP0N_SEL	Res.			OP0OS[4:0]				
r	rw	rw	rw		rw		rw				rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OP0RFB	OP0RIN	ADC_P_EN	ADC_N_EN	OP0GS[2:0]			Res.	VM_SEL	OP0PS	OP0NS	Res.	OP0PRS	OP0NRS	OP0EN
	rw	rw	rw	rw	rw				rw	rw	rw		rw	rw	rw

Bit 31	OP0TRIMO: 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	OP0TRIMEN: OPA0 修调使能
	0: OPA0 修调禁止（默认）
	1: OPA0 修调使能
Bit 29	OP0P_PD: OPA0 正端输入下拉到地
	0: OPA0 正端不下拉
	1: OPA0 正端下拉
Bit 28	OP0N_PD: OPA0 负端输入下拉到地
	0: OPA0 负端不下拉
	1: OPA0 负端下拉
Bit 27	保留，必须保持复位值
Bit 26	OP0PSEL: OPA0 正端输入（OPA0_P）选择
	0: OPA0_P0 作为正端输入（默认）
	1: OPA0_P1 作为正端输入
Bit 25	保留，必须保持复位值

Bit	24	OPONSEL: OPA0 负端输入 (OPA0_N) 选择
		0: OPA0_N0 作为负端输入 (默认)
		1: OPA0_N1 作为负端输入
Bits	23:21	保留, 必须保持复位值
Bits	20:16	OP0OS[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
		00000: OPA0_N-OPA0_P 电压偏差最大 (默认)
		...
		11111: OPA0_P-OPA0_N 电压偏差最大
		在修调过程中, OP0OS[4:0]的值由 00000->11111 变化, OP0TRIMO 变为 1 即修调结束
Bit	15	保留, 必须保持复位值
Bit	14	OPORFB: OPA0 带电阻输出使能
		0: OPA0 带 4K Ω 电阻输出断开 (默认)
		1: OPA0 带 4K Ω 电阻输出使能
Bit	13	OPORIN: OPA0 输入电阻选择
		0: OPA0 输入电阻为 36K Ω (默认)
		1: OPA0 输入电阻为 18K Ω
Bit	12	ADC_P_EN: ADCIN[11]采样后触发 OPA0 正端输入切换使能
		0: 禁止 (默认)
		1: 使能
Bit	11	ADC_N_EN: ADCIN[11]采样后触发 OPA0 负端输入切换使能
		0: 禁止 (默认)
		1: 使能
Bits	10:8	OP0GS[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: x4
		011: x6
		100: x8
		101: x10
		110: x16
		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: 运放输出偏置来自 PMU_AOUT (默认)
		1: 运放输出偏置为 ADC 参考电压的一半
Bit	5	OP0PS: 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	OP0NS: 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	保留, 必须保持复位值
Bit 2	OP0PRS: OPA0 正端输入电阻短路
	0: 不短路 (默认)
	1: 短路
Bit 1	OP0NRS: OPA0 负端输入电阻短路
	0: 不短路 (默认)
	1: 短路
Bit 0	OP0EN: 运放 OPA0 使能
	0: 禁止 (默认)
	1: 使能

12.4.2 OPA1 控制状态寄存器 (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	OP1P_PD	OP1N_PD	Res.	OP1P_SEL	Res.	OP1N_SEL	Res.			OP1OS[4:0]				
r	rw	rw	rw		rw		rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OP1RFB	OP1RIN	ADC_P_EN	ADC_N_EN	OP1GS[2:0]			Res.	VM_SEL	OP1PS	OP1NS	Res.	OP1PRS	OP1NRS	OP1EN
	rw	rw	rw	rw	rw				rw	rw	rw		rw	rw	rw

Bit 31	OP1TRIMO: OPA1 修调状态 (以 OPxOS 从 00000 到 11111 为例)
	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
Bit 29	OP1P_PD: OPA1 正端输入下拉到地
	0: OPA1 正端不下拉
	1: OPA1 正端下拉
Bit 28	OP1N_PD: OPA1 负端输入下拉到地
	0: OPA1 负端不下拉
	1: OPA1 负端下拉
Bit 27	保留, 必须保持复位值
Bit 26	OP1PSEL: OPA1 正端输入 (OPA1_P) 选择
	0: OPA1_P0 作为正端输入 (默认)
	1: OPA1_P1 作为正端输入
Bit 25	保留, 必须保持复位值

Bit	24	OP1NSEL: OPA1 负端输入 (OPA1_N) 选择
		0: OPA1_N0 作为负端输入 (默认)
		1: OPA1_N1 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
		00000: OPA1_N-OPA1_P 电压偏差最大 (默认)
		...
		11111: OPA1_P-OPA1_N 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
Bit	15	保留, 必须保持复位值
Bit	14	OP1RFB: OPA1 带电阻输出使能
		0: OPA1 带 4K Ω 电阻输出断开 (默认)
		1: OPA1 带 4K Ω 电阻输出使能
Bit	13	OP1RIN: OPA1 输入电阻选择
		0: OPA1 输入电阻为 36K Ω (默认)
		1: OPA1 输入电阻为 18K Ω
Bit	12	ADC_P_EN: ADCIN[12]采样后触发 OPA1 正端输入切换使能
		0: 禁止 (默认)
		1: 使能
Bit	11	ADC_N_EN: ADCIN[12]采样后触发 OPA1 负端输入切换使能
		0: 禁止 (默认)
		1: 使能
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: x4
		011: x6
		100: x8
		101: x10
		110: x16
		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: 运放输出偏置来自 PMU_AOUT (默认)
		1: 运放输出偏置为 ADC 参考电压的一半
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	保留, 必须保持复位值
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 (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	OP2P_PD	OP2N_PD	Res.	OP2P_SEL	Res.	OP2N_SEL	Res.			OP2OS[4:0]				
r	rw	rw	rw		rw		rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OP2RFB	OP2RIN	ADC_P_EN	ADC_N_EN	OP2GS[2:0]			Res.	VM_SEL	OP2PS	OP2NS	Res.	OP2PRS	OP2NRS	OP2EN
	rw	rw	rw	rw	rw				rw	rw	rw		rw	rw	rw

Bit 31	OP2TRIMO: OPA2 修调状态 (以 OPxOS 从 00000 到 11111 为例)
	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
Bit 29	OP2P_PD: OPA2 正端输入下拉到地
	0: OPA2 正端不下拉
	1: OPA2 正端下拉
Bit 28	OP2N_PD: OPA2 负端输入下拉到地
	0: OPA2 负端不下拉
	1: OPA2 负端下拉
Bit 27	保留, 必须保持复位值
Bit 26	OP2PSEL: OPA2 正端输入 (OPA2_P) 选择
	0: OPA2_P0 作为正端输入 (默认)
	1: OPA2_P1 作为正端输入
Bit 25	保留, 必须保持复位值

Bit	24	OP2NSEL: OPA2 负端输入 (OPA2_N) 选择
		0: OPA2_N0 作为负端输入 (默认)
		1: OPA2_N1 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
		00000: OPA2_N-OPA2_P 电压偏差最大 (默认)
		...
		11111: OPA2_P-OPA2_N 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
Bit	15	保留, 必须保持复位值
Bit	14	OP2RFB: OPA2 带电阻输出使能
		0: OPA2 带 4K Ω 电阻输出断开 (默认)
		1: OPA2 带 4K Ω 电阻输出使能
Bit	13	OP2RIN: OPA2 输入电阻选择
		0: OPA2 输入电阻为 36K Ω (默认)
		1: OPA2 输入电阻为 18K Ω
Bit	12	ADC_P_EN: ADCIN[2]采样后触发 OPA2 正端输入切换使能
		0: 禁止 (默认)
		1: 使能
Bit	11	ADC_N_EN: ADCIN[2]采样后触发 OPA2 负端输入切换使能
		0: 禁止 (默认)
		1: 使能
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: x4
		011: x6
		100: x8
		101: x10
		110: x16
		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: 运放输出偏置来自 PMU_AOUT (默认)
		1: 运放输出偏置为 ADC 参考电压的一半
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	保留, 必须保持复位值
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. PWM 专用定时器 TIM1

LCM32F06X 包括 1 个 PWM 专用定时器、4 个通用定时器、1 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 比较了定时器的相关特性。

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
PWM 专用定时器	TIM1	16-bit	Up, Down, Up/down	1~65536	Yes	5	4
通用定时器	TIM2	16-bit	Up, Down, Up/down	1~65536	Yes	4	0
	TIM15	16-bit	Up, Down, Up/down	1~65536	Yes	2	2
	TIM16	16-bit	Up, Down, Up/down	1~65536	Yes	2	1
	TIM17	16-bit	Up, Down, Up/down	1~65536	Yes	1	1

13.1 TIM1 主要特性

PWM 专用定时器 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 5 路不同的比较通道，其中通道 1~4 支持双路互补输出，通道 5 为单路输出。主要用于基本定时和产生输出波形（输出比较，PWM 和单脉冲模式），对应于不同事件（比较、溢出、刹车、触发）产生相应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）进行同步。

TIM1 可以看作是一个复用 8 个通道的 4 相 PWM，支持互补的 PWM 输出，能插入可编程的死区时间。4 个独立的通道可以被分别用于：

- 输出比较
- PWM 生成（边沿对齐、中央对齐、中央偏移模式）
- 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。

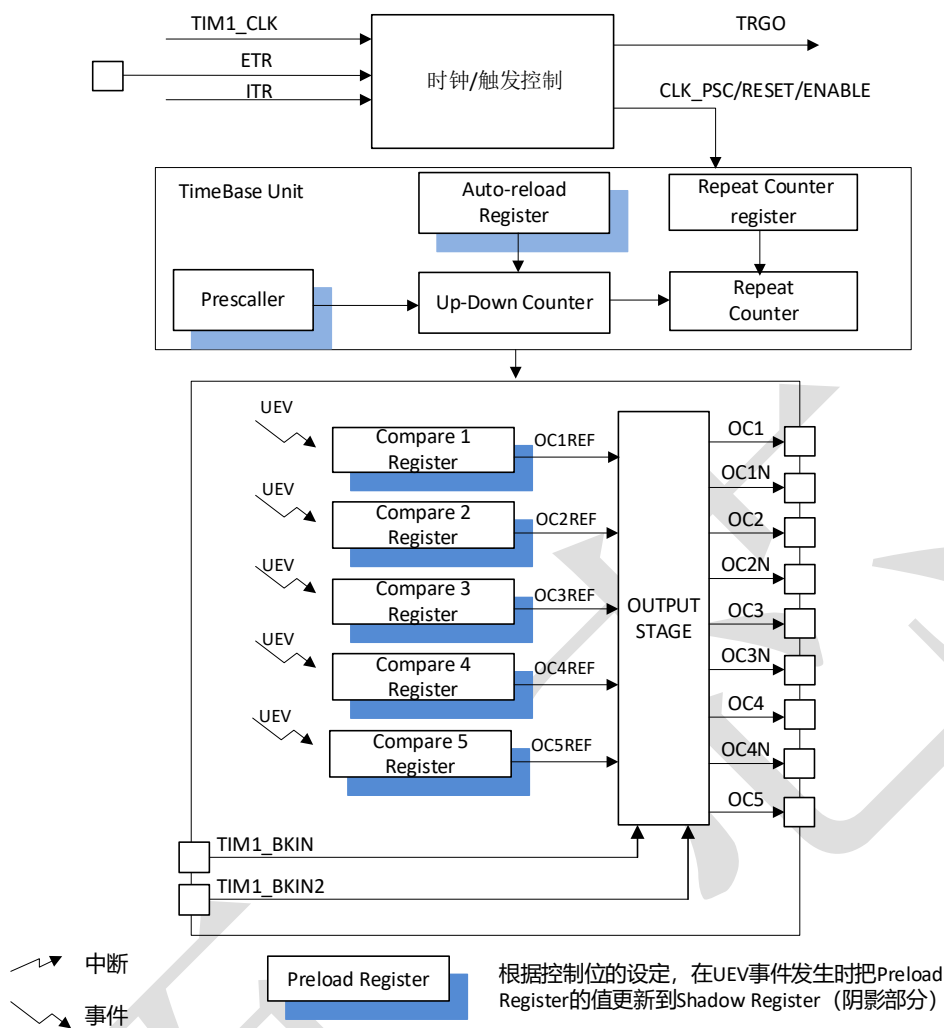


图 13-1 PWM 专用定时器 TIM1 框图

13.2 管脚配置

表 13-2 TIM1 管脚配置

管脚名称	管脚类型	管脚描述	复用 I/O 口	备注
TIM1_PWM1	O	TIM1 的 PWM 输出 1	PB2/PA8/PA15/PB3/PB6	可通过 TIM1_OMR.PWM1_SEL 配置, 可选择 OC1~OC4, OC1N~OC4N
TIM1_PWM2	O	TIM1 的 PWM 输出 2	PB7/PB8/PA12/PA9	可通过 TIM1_OMR.PWM2_SEL 配置, 可选择 OC1~OC4, OC1N~OC4N
TIM1_PWM3	O	TIM1 的 PWM 输出 3	PB7/PB8/PA11/PA7/PA10/PB5	可通过 TIM1_OMR.PWM3_SEL 配置, 可选择 OC1~OC4, OC1N~OC4N
TIM1_PWM4	O	TIM1 的 PWM 输出 4	PA2/PA13/PA15/PB5	可通过 TIM1_OMR.PWM4_SEL 配置, 可选择 OC1~OC4, OC1N~OC4N
TIM1_PWM5	O	TIM1 的 PWM 输出 5	PB8/PA2/PA7/PA6/PA13/PB3/PB6	可通过 TIM1_OMR.PWM5_SEL 配置, 可选择 OC1~OC4, OC1N~OC4N
TIM1_PWM6	O	TIM1 的 PWM 输出 6	PB7/PA5/PB0/PB4/PB5	可通过 TIM1_OMR.PWM6_SEL 配置, 可选择 OC1~OC4, OC1N~OC4N

管脚名称	管脚类型	管脚描述	复用 I/O 口	备注
TIM1_PWM7	O	TIM1 的 PWM 输出 7	PB1/PB3/PB4	可通过 TIM1_OMR.PWM7_SEL 配置, 可选择 OC1~OC4, OC1N~OC4N
TIM1_PWM8	O	TIM1 的 PWM 输出 8	PA10/PA14/PB3/PB5	可通过 TIM1_OMR.PWM8_SEL 配置, 可选择 OC1~OC4, OC1N~OC4N
TIM1_CH5	O	TIM1 的 OC5 通道	PA14	单独的 OC5 通道, 没有互补
TIM1_GPIO_BKIN	I	TIM1 的外部 I/O 刹车	PA6/PA10/PA14/PA15/PB2	-
TIM1_ETR	I	TIM1 的外部 I/O 触发	PA0	-

13.3 时基单元

时基单元包含:

- 16位向上/向下计数寄存器 (TIM1_CNT)
- 16位预分频寄存器 (TIM1_PSC)
- 16位自动重载寄存器 (TIM1_ARR)
- 8位重复计数寄存器 (TIM1_RCR)

16 位计数寄存器、预分频寄存器、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.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。

写计数寄存器的操作没有缓存, 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。

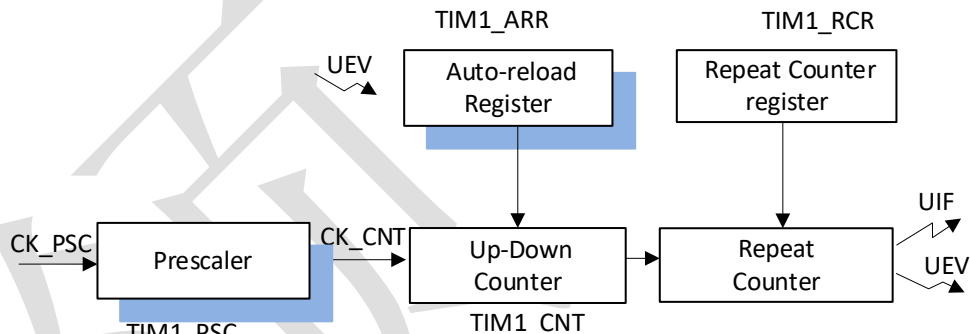


图 13-2 定时器时基单元

13.3.1 自动重载寄存器

自动重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 TIM1_CR1.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.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.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.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.UG 位
- 时钟/触发控制器产生硬件复位

在预装载使能时（TIM1_CR1.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.UDIS 位将禁止产生更新事件（UEV）。

13.3.2 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影子寄存器，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 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。

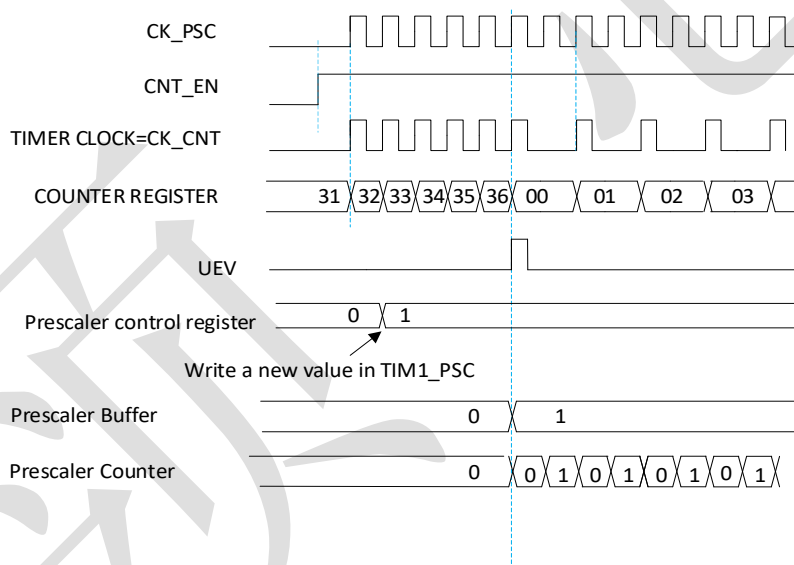


图 13-3 预分频系数从 1 变 2 的计数器时序

13.4 计数器模式

13.4.1 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（TIM1_ARR），然后重新从 0 开始计数并产生一个计数器上溢出事件，同时，如果 TIM1_CR1.UDIS 位是 0，将会产生一个更新事件（UEV）。

使用软件置位 TIM1_CR1.UDIS 位，可以禁止产生更新事件，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，如果配置 TIM1_CR1.URS=0（选择更新请求

源），置位 TIM1_EGR.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，但硬件不设置 UIF 标志（即不产生中断请求）。

如果定时器带有重复计数器，由于重复寄存器没有影子寄存器，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（TIM1_SR.UIF 位），自动重装载影子寄存器会更新为新的预装载寄存器的值（TIM1_ARR）。预分频寄存器的影子寄存器会更新为新的预装载寄存器的值（TIM1_PSC）。

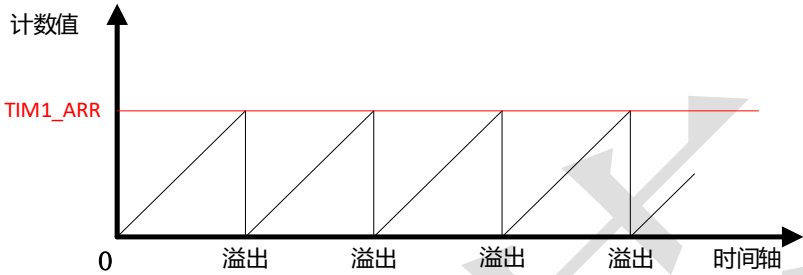


图 13-4 向上计数溢出

下图给出例子，当 TIM1_ARR=0x36 时计数器在二分频时钟频率下的动作。

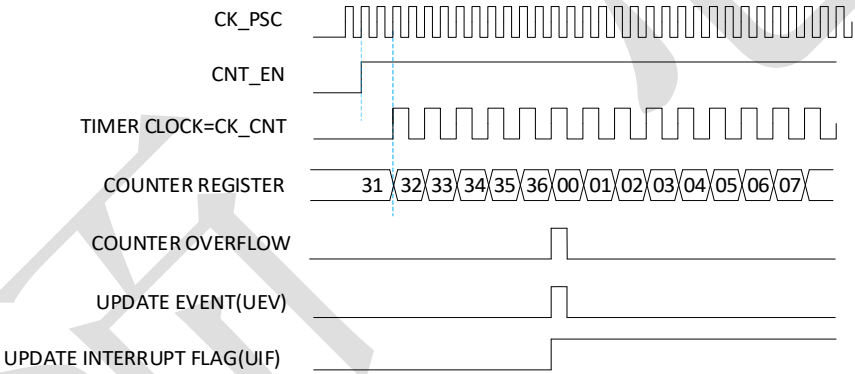


图 13-5 计数器在二分频时钟频率下的时序图

13.4.2 向下计数模式

在向下计数模式中，计数器从自动重装载寄存器的值（TIM1_ARR）开始向下计数到 0，然后再从 TIM1_ARR 重新开始计数，并产生一个计数器向下溢出事件。如果 TIM1_CR1.UDIS 位是 0，将会产生一个更新事件（UEV）。

置位 TIM1_CR1.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，如果配置 TIM1_CR1.URS=0（选择更新请求源），置位 TIM1_EGR.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，但硬件不设置 UIF 标志（即不产生中断请求）。

如果定时器带有重复计数器，由于重复寄存器没有影子寄存器，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（TIM1_SR.UIF

位），自动重装载影子寄存器会更新为新的预装载寄存器的值（TIM1_ARR）。预分频寄存器的影子寄存器会更新为新的预装载寄存器的值（TIM1_PSC）。

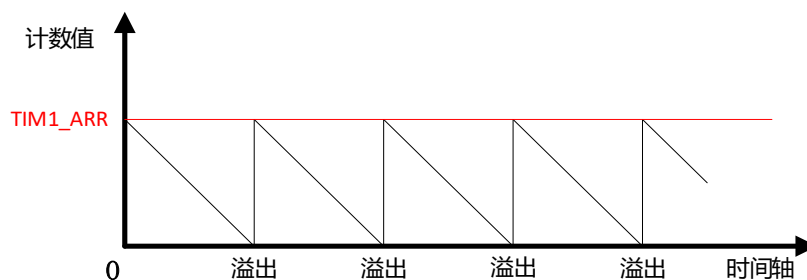


图 13-6 向下计数溢出

13.4.3 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（TIM1_ARR-1），产生一个计数器上溢出事件，然后向下计数到 0 并且产生一个计数器下溢出事件；然后再从 0 开始重新计数。如果 TIM1_CR1.UDIS 位是 0，计数器向上/向下溢出都会将会产生一个更新事件（UEV）。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。

在此模式下，不能写入 TIM1_CR1 中的 DIR 方向位。它由硬件更新并指示当前的计数方向。

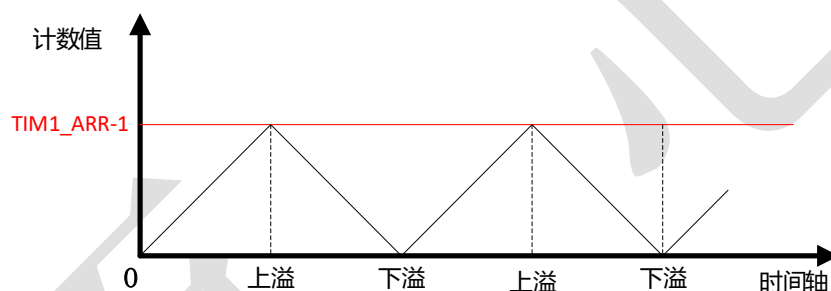


图 13-7 中央对齐溢出

如果定时器带有重复计数器，在重复了指定次数（TIM1_RCR）的向上和向下溢出之后才会产生更新事件（UEV）。

设置 TIM1_CR1.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，如果配置 TIM1_CR1.URS=0（选择更新请求源），置位 TIM1_EGR.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，但硬件不设置 UIF 标志（即不产生中断请求）。

如果定时器带有重复计数器，由于重复寄存器没有影子寄存器，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（TIM1_SR.UIF 位），自动重装载影子寄存器会更新为新的预装载寄存器的值（TIM1_ARR）。预分频寄存器的影子寄存器的会更新为新的预装载寄存器的值（TIM1_PSC）。

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 TIM1_CR1.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。具体来说：

- 向计数器写入了比自动装载值更大的数值时（TIM1_CNT > TIM1_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 TIM1_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 TIM1_EGR.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。

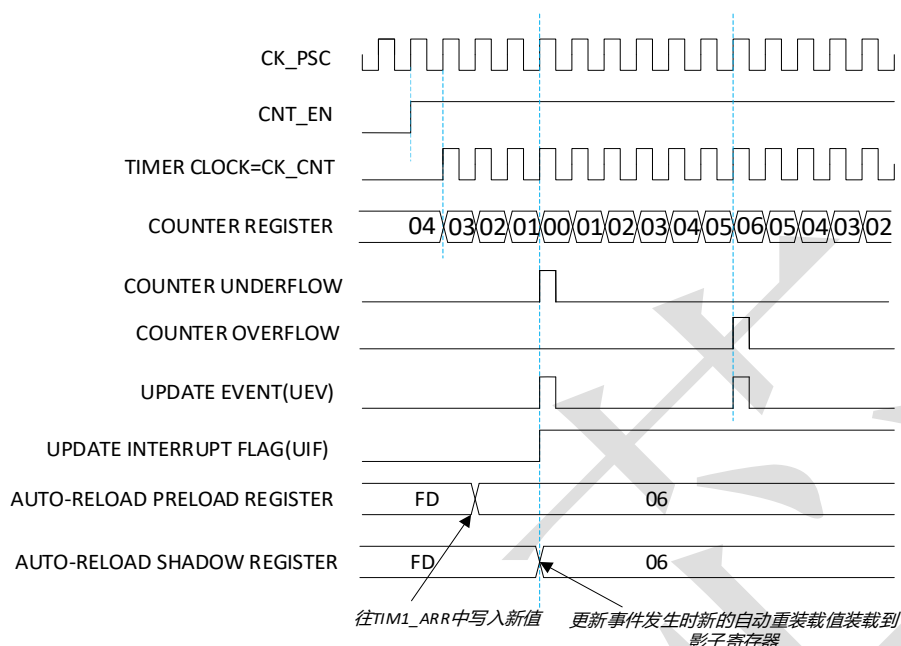


图 13-8 计数器时序图，内部时钟分频为 1，TIM1_ARR=0x6

13.5 重复计数器

13.4 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（TIM1_ARR、TIM1_PSC、在比较模式下的比较寄存器 TIM1_CCRi 和 TIM1_CCORi 都带有影子寄存器），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出和每次下溢出时

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 $2 \times t_{CK_PSC}$ 。

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 TIM1_EGR.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。

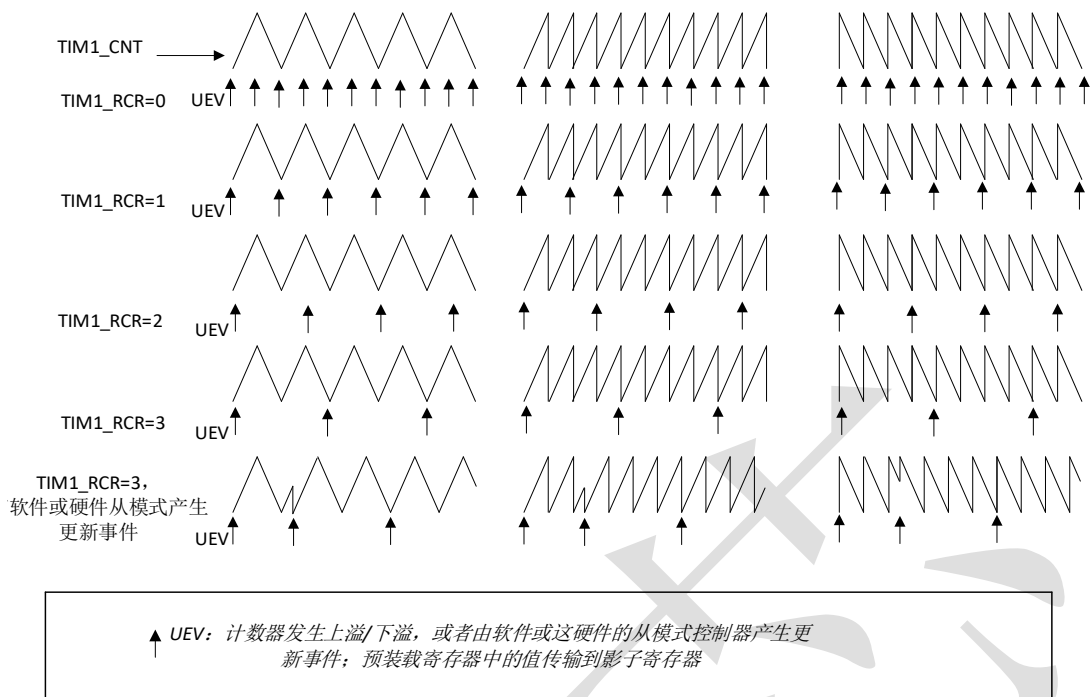


图 13-9 不同模式下更新速率的例子, 以及寄存器 TIM1_RCR 的设置

13.6 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。

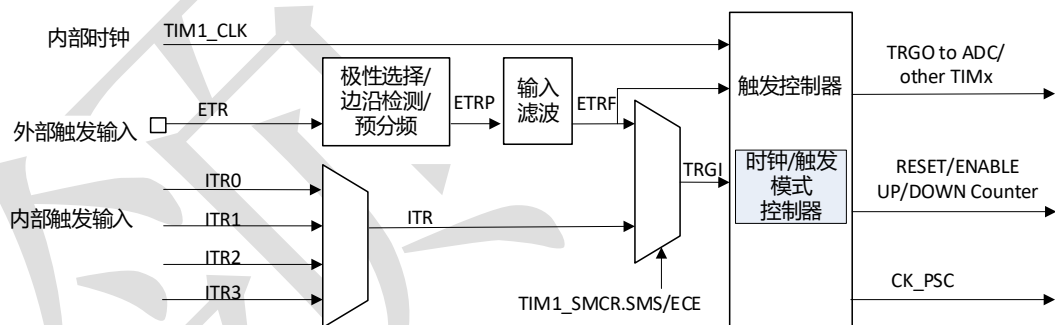


图 13-10 时钟/触发控制器框图

时基单元的预分频时钟 (CK_PSC) 可以有以下来源:

- 内部时钟 (TIM1_CLK): 来自于 SYS_CLK 或 PCLK0
- 外部时钟模式 1: 内部触发输入 (ITRx)
- 外部时钟模式 2: 外部触发输入 (ETR)

13.6.1 内部时钟源 (TIM1_CLK)

如果 TIM1_SMCR.SMS=3'b000, TIM1_SMCR.ECE=0, 则 TIM1_CR1.CEN、TIM1_CR1.DIR 和 TIM1_EGR.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。

13.6.2 外部时钟模式 1

当 TIM1_SMCR.SMS=3'b111 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。

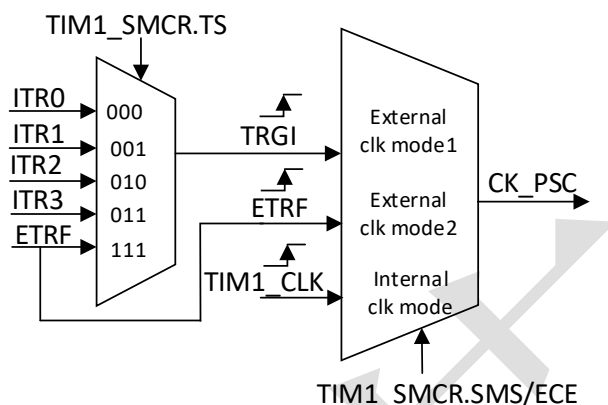


图 13-11 TIM1 外部时钟框图

例如，要配置向上计数器在内部触发 ITR1 上升沿计数，使用下列步骤：

1. 配置 TIM1_SMCR.SMS=3'b111，配置计数器使用外部时钟模式 1
2. 配置 TIM1_SMCR.TS=3'b001，选定 ITR1 作为输入源
3. 设置 TIM1_CR1.CEN=1，启动计数器

当上升沿出现在 ITR1，计数器计数一次，且触发标识位（TIM1_SR.TIF 位）被置 1，如果使能了中断（在 TIM1_DIER 寄存器中配置）则会产生中断请求。

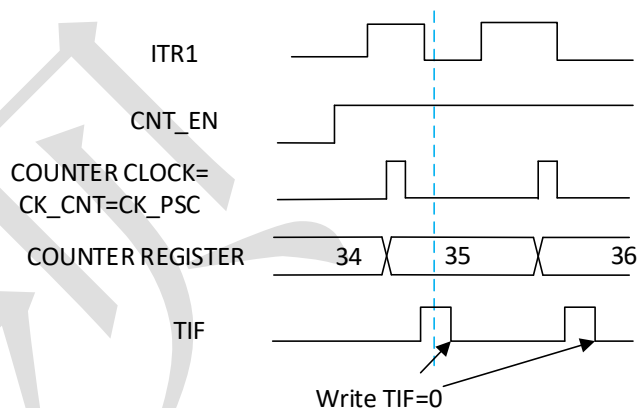


图 13-12 外部时钟模式 1 下的控制电路

13.6.3 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 TIM1_SMCR.ECE 位写 1，即可选定此模式。

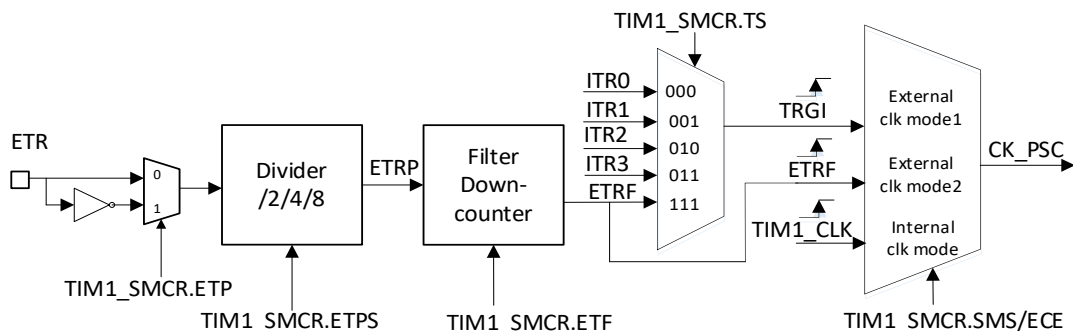


图 13-13 外部触发时钟框图

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 TIM1_SMCR.ETF=4'b0000
2. 设置预分频器，配置 TIM1_SMCR.ETPS=2'b01
3. 选择 ETR 的上升沿检测，配置 TIM1_SMCR.ETP=0
4. 开启外部时钟模式 2，配置 TIM1_SMCR.ECE=1
5. 启动计数器，写 TIM1_CR1.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。

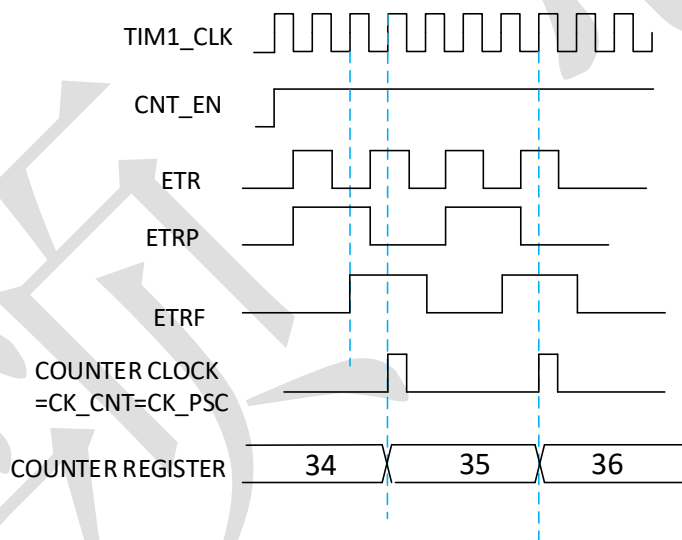


图 13-14 外部时钟模式 2 下的控制电路

13.7 比较通道

定时器的 I/O 引脚（TIM1_CCI）可以用作输出比较，这个功能可以通过配置比较通道模式寄存器 TIM1_CCMRi.CCIiS 通道选择位来实现，此处的 i 代表通道数。

每一个比较通道都是围绕着一个比较寄存器 CCRi 和一个比较偏移寄存器 CCORi（包含影子寄存器）来构建的，包括比较器和输出控制。通过配置比较偏移控制寄存器 TIM1_OCR.OCi_MODE 可以使比较通道在不同计数方向上选择与 CCRi 或者 CCORi 进行比较。

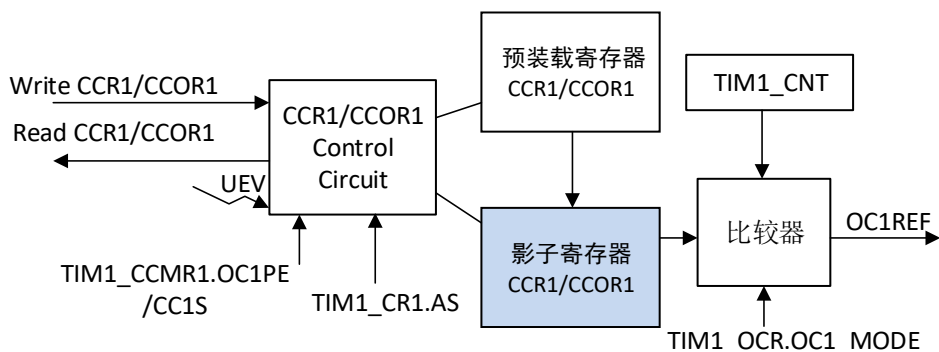


图 13-15 比较通道 1 的主电路

比较模块 CCRi 和 CCORi 都是由一个预装载寄存器和一个影子寄存器组成，比较模式下，读过程可根据 TIM1_CR1.AS 位选择读取影子寄存器或预装载寄存器，写过程中如果 TIM1_CCMi.OCiPE=0，则可直接写入到影子寄存器。在比较模式下，TIM1_CCMi.OCiPE=1 且在 UEV 事件到来时预装载寄存器的内容被复制到影子寄存器中，然后使用影子寄存器的值和计数器进行比较。

当通道被配置成输出模式时（TIM1_CCMRi.CCIS=0），可以随时访问 TIM1_CCRi 寄存器和 TIM1_CCORi 寄存器（此处的 i 指通道数）。

13.7.1 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）。TIM1 输出模块如图 13-16 所示，TIM1 通道经死区控制、极性选择、刹车控制和滤波后产生 OCi 和 OCiN 信号（通道 5 不存在互补通道，因此不受死区时间控制）。TIM1 可以通过 OMR 输出模式寄存器 TIM1_OMR.PWMx_SEL 来选择 TIM1_PWMx 输出是 OC1~OC4 或 OC1N~OC4N，TIM1_CH5 单独输出。另外，可通过 TIM1_BDTR.BKTRIG 选择连接到 ADC/ACMP 的触发信号 TIM1_trig_oci 是否经过刹车控制。

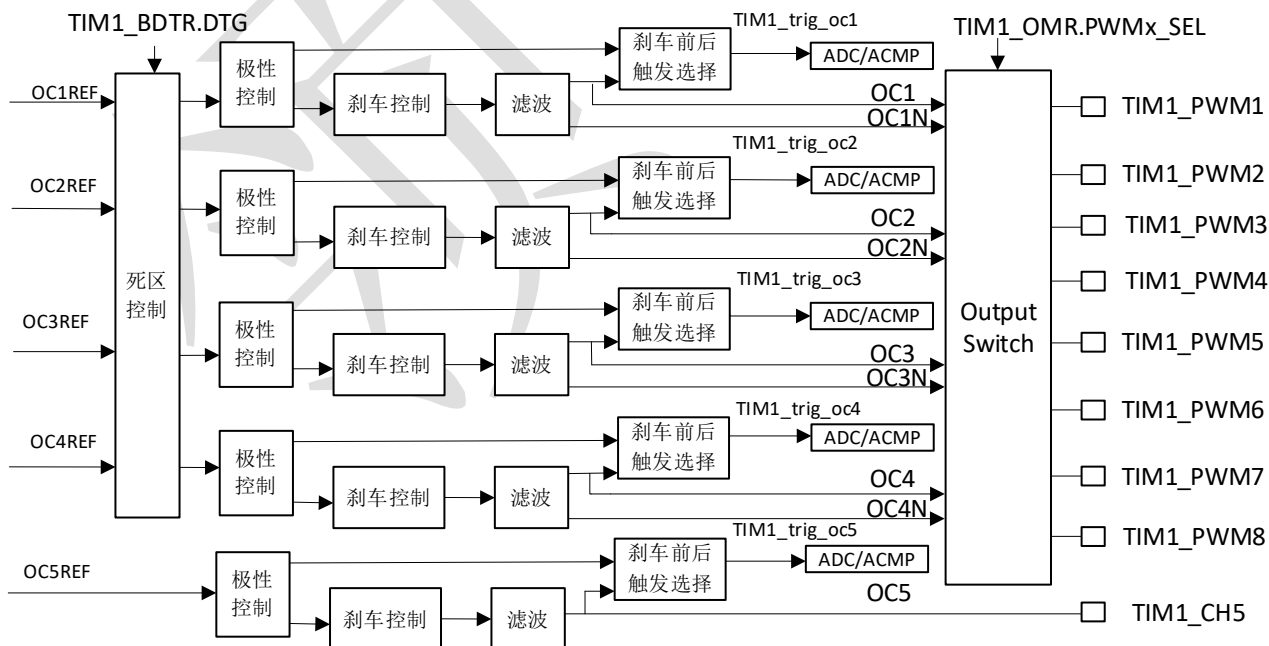


图 13-16 输出模块框图

13.7.2 强制输出模式

在输出模式（TIM1_CCMRi.CCIS=2'b00）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。

配置 TIM1_CCMRi.OCiM=3'b101，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。

配置 TIM1_CCMRi.OCiM=3'b100，可强置 OCiREF 信号为低。

该模式下，在 TIM1_CCRi/TIM1_CCORi 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。

13.7.3 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。当计数器与比较寄存器的内容相等时，有如下操作：

- 根据不同的输出比较模式（由 TIM1_CCMRi.OCiM 设定），相应的 OCiREF 输出信号为：
 - 保持不变（OCiM=3'b000）
 - 设置为有效电平（OCiM=3'b001）
 - 设置为无效电平（OCiM=3'b010）
 - 翻转（OCiM=3'b011）
- 置位中断状态寄存器中的标志位（TIM1_SR.CCiIF 位）
- 若设置了相应的中断使能位（TIM1_DIER.CCiIE 位），则产生一个中断

TIM1_CCER.CCiP 位用于选择有效和无效的电平极性。TIM1_CCMRi.OCiPE 位用于选择 TIM1_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 TIM1_ARR 和 TIM1_CCRi 寄存器中，并配置 TIM1_OCR.OCi_MODE=4'b0011（默认值，使用 CCRi 作为比较寄存器）
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=3'b011
 - 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 TIM1_CR1.CEN 位来启动计数器

在未使用预装载寄存器（OCiPE=0，否则 TIM1_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，TIM1_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。

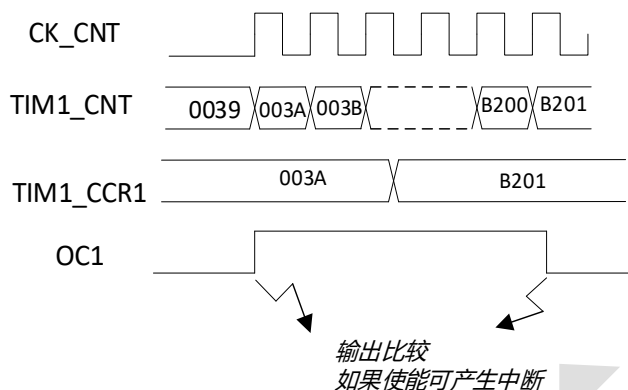


图 13-17 输出比较模式，翻转 OC1

如果在输出比较模式的翻转输出模式下，当配置 OCi_MODE 的值同时选择了 $CCRi$ 和 $CCORi$ 进行比较（如 $OCi_MODE=4'b1111$ ），则每次匹配都会翻转 $OCiREF$ 。

13.7.4 PWM 模式

脉冲宽度调制（PWM）模式可以产生一个由 $TIM1_ARR$ 寄存器确定频率，由 $TIM1_CCRi$ 寄存器确定占空比的信号。

在 $TIM1_CCMRi.OCiM$ 位写入 $3'b110$ （PWM 模式 1）或 $3'b111$ （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 $TIM1_CCMRi.OCiPE$ 位使能相应的预装载寄存器，若设置 $TIM1_CR1.ARPE=1$ ，则使能 $TIM1_ARR$ 寄存器的预装载功能，即软件向 $TIM1_ARR$ 写值的时候，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。因此在计数器开始计数之前，必须通过设置 $TIM1_EGR.UG$ 位来初始化所有的寄存器。

OCi 的极性可以通过软件在 $TIM1_CCER.CCiP$ 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 $TIM1_CCER$ 和 $TIM1_BDTR$ 寄存器中 $CCiE$ 、 MOE 、 $OSSR$ 和 $OSSI$ 位的组合来控制。详见 $TIM1_CCER$ 寄存器的描述。

在 PWM 模式（模式 1 或模式 2）下， $TIM1_CNT$ 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 $TIM1_CCRi \leq TIM1_CNT$ 或者 $TIM1_CNT \leq TIM1_CCRi$ 。如果使用比较偏移功能， $TIM1_CNT$ 会和 $TIM1_CCRi/TIM1_CCORi$ 进行比较。

根据 $TIM1_CR1$ 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。

13.7.4.1 PWM 边沿对齐模式

向上计数配置

当 $TIM1_CR1.DIR$ 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 $TIM1_CNT < TIM1_CCRi$ 时，PWM 参考信号 $OCiREF$ 为高，否则为低。如果 $TIM1_CCRi$ 中的比较值大于自动重装载值（ $TIM1_ARR$ ），则 $OCiREF$ 保持为 1。如果比较值为 0，则 $OCiREF$ 保持为 0。下图为 $TIM1_ARR=8$ 时边沿对齐的 PWM 波形实例。

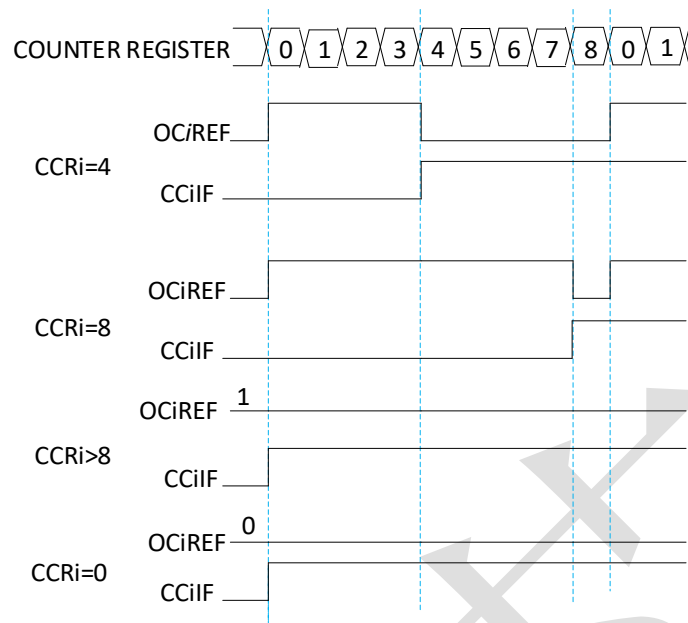


图 13-18 TIM1_ARR=8 时边沿对齐的 PWM 波形实例

向下计数的配置

当 TIM1_CR1.DIR 位为高时执行向下计数。在 PWM 模式 1 时，当 TIM1_CNT>TIM1_CCRi 时参考信号 OCiREF 为低，否则为高。如果 TIM1_CCRi 中的比较值大于 TIM1_ARR 中的自动重装载值，则 OCiREF 保持为 1。该模式下不能产生 0% 的 PWM 波形。

13.7.4.2 PWM 中央对齐模式

当 TIM1_CR1.CMS 位!=0 时为中央对齐模式。可配置 TIM1_OCR.OCi_MODE，选择在计数器向上/向下计数过程中采用 CCRi 或 CCORi 的值进行比较。可以输出 0% 的 PWM 波形。

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。TIM1_CR1.DIR 由硬件更新，不要用软件修改它。

下图为 PWM 中央对齐模式波形示例：

- TIM1_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 13-19 中标出）
 - 只有在计数器向下计数时（中央对齐模式 1，CMS=2'b01）
 - 只有在计数器向上计数时（中央对齐模式 2，CMS=2'b10）
 - 在计数器向上和向下计数时（中央对齐模式 3，CMS=2'b11）

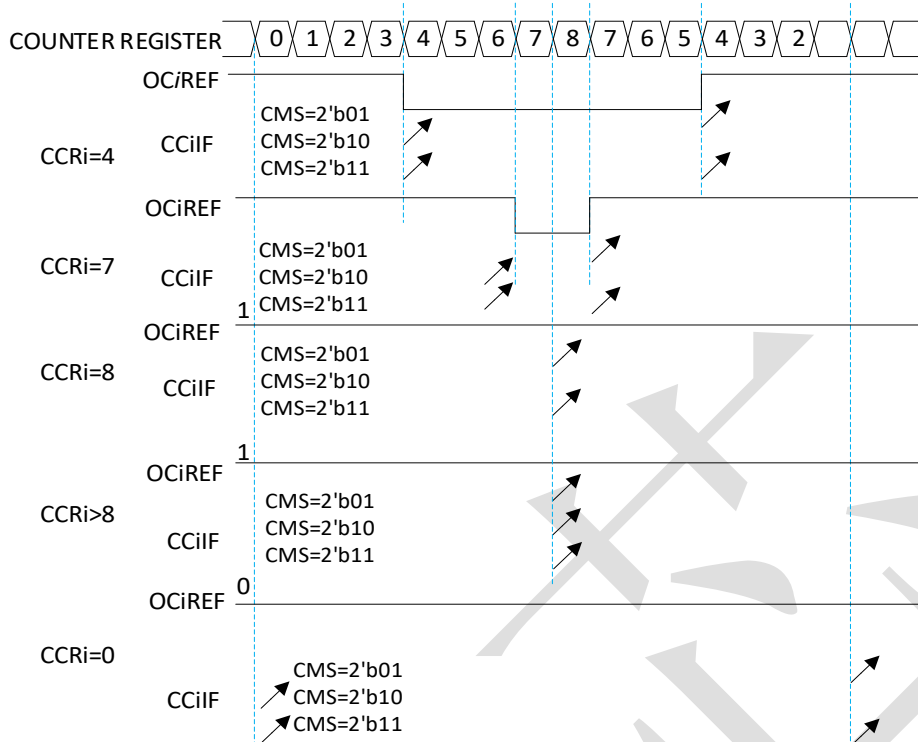


图 13-19 中央对齐模式 PWM (TIM1_ARR=8, OCi_MODE=3)

13.7.4.3 PWM 中央偏移模式

PWM 中央偏移模式是在中央对齐模式的基础上，设置 TIM1_OCR.OCi_MODE，选择在计数器向上/向下计数过程中分别采用 CCRi 和 CCORi 的值进行比较。

例如 OCi_MODE=4'b1001 时：当 TIM1_CR1.DIR 位为低的时候执行向上计数，此时 $TIM1_CNT < TIM1_CCRi$ 时，PWM 参考信号 OCiREF 为高，否则为低。如果 TIM1_CCRi 中的比较值大于自动重装载值 (TIM1_ARR)，则 OCiREF 保持为 1。如果比较值为 0，则 OCiREF 保持为 0。当 TIM1_CR1.DIR 位为高时执行向下计数，此时 $TIM1_CNT > TIM1_CCORi$ 时参考信号 OCiREF 为低，否则为高。如果 TIM1_CCRi 中的比较值大于 TIM1_ARR 中的自动重装载值，则 OCiREF 保持为 1。

特别地，在 PWM 模式下，当向上计数过程中同时选择了 CCRi 和 CCORi 进行比较，TIM1_CNT 将和两个比较值进行比较，此时如果 TIM1_CNT 小于更小的比较值或 TIM1_CNT 大于更大的比较值时，PWM 参考信号 OCiREF 为高，否则为低。当向下计数过程中同时选择了 CCRi 和 CCORi 进行比较，TIM1_CNT 将和两个比较值进行比较，此时 TIM1_CNT 大于更大的比较值或 TIM1_CNT 小于或等于更小的比较值时，PWM 参考信号 OCiREF 为高，否则为低。

中央偏移模式下可以输出 0% 的 PWM 波形。

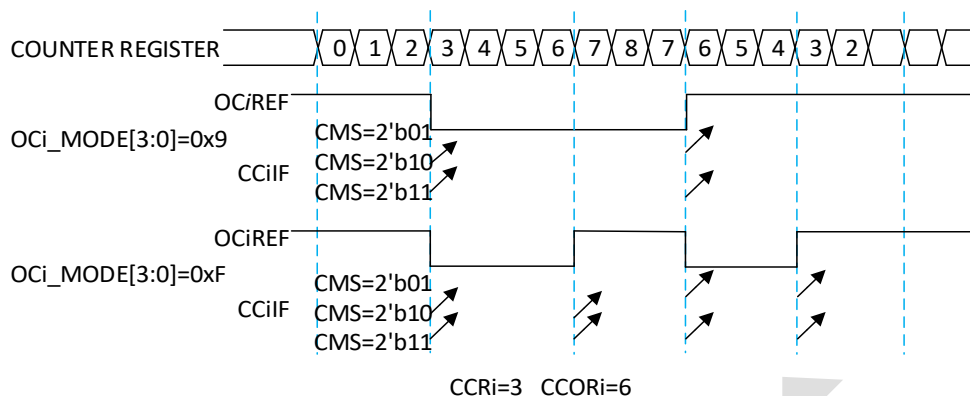


图 13-20 中央偏移模式 PWM (TIM1_ARR=8, CCRi=3, CCORi=6)

13.7.5 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 TIM1_CR1.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 产生时停止。

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 13-21 所示，如果想在 OC1 生成一个高电平脉冲，它通过 ETR 的输入触发，启动延迟为 t_{DELAY} ，维持时间为 t_{PULSE} 。首先要将 ETR 设定为触发信号，此外单脉冲信号还要通过比较寄存器来定义 t_{DELAY} 和 t_{PULSE} （需要考虑时钟频率和预分频器）：

1. 设置外部触发预分频以及外部触发滤波器均为 0，(TIM1_SMCR.ETPS/ETF)
2. 设置 ETR 为从模式下的触发信号 (TRGI)，配置 TIM1_SMCR.TS=3'b111
3. 设置 TI2FP2 为计数器的启动信号，配置 TIM1_SMCR.SMS=3'b110 (触发模式)
4. 设置触发延迟窗口时间为 0 (TIM1_DLG.DLG)
5. t_{DELAY} 通过 TIM1_CCR1 寄存器配置得到
6. t_{PULSE} 通过 TIM1_ARR - TIM1_CCR1 得到
7. 如果希望高电平脉冲是在计数器到达 TIM1_CCR1 时信号由 0 变为 1，到达 TIM1_ARR 时信号由 1 变为 0，则设置 PWM 模式 2，配置 TIM1_CCMR1.OC1M=3'b111
8. 可选启用预装载功能，配置 TIM1_CCMR1.OC1PE=1 和 TIM1_CR1.ARPE=1。如果使用预装载功能，在 TIM1_ARR 和 TIM1_CCR1 配置后要配置 TIM1_EGR.UG 位进行更新，并等待定时器的启动。在这个例子里，CC1P=0, DIR=0, CMS=2'b00。

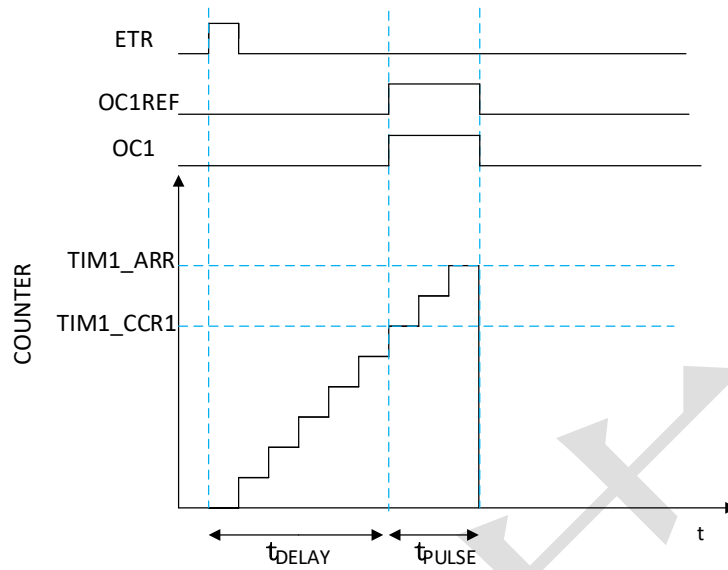


图 13-21 单脉冲模式

13.7.6 特殊情况：OCi 快速使能

在单脉冲模式下，外部触发经预分频和滤波处理后会设置 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.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。

13.7.7 互补输出和死区插入

TIM1 能够输出两路互补信号，并且能够对输出的瞬时关断和接通进行延时管理。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 TIM1_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：TIM1_CCER 寄存器的 CCiE 和 CCiNE 位，TIM1_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 13-22 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，如图 13-23 和图 13-24 所示。

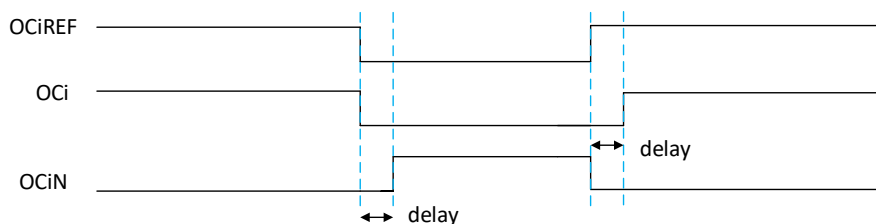


图 13-22 带死区插入的互补输出

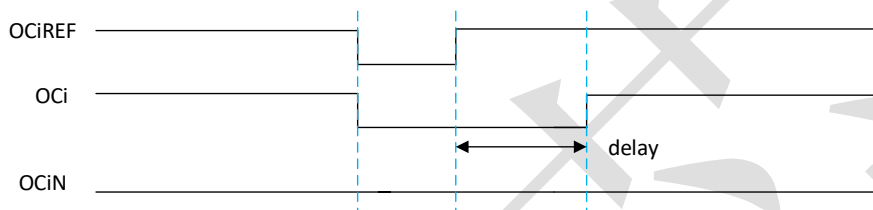


图 13-23 死区波形延迟大于负脉冲

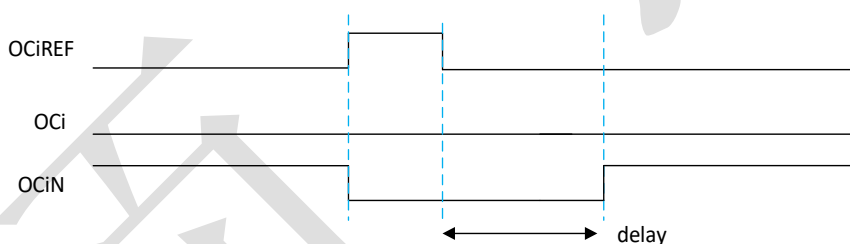


图 13-24 死区波形延迟大于正脉冲

13.7.8 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 TIM1_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），OCi 和 OCiN 受到死区时间控制，且 OCi 和 OCiN 输出相位相反

13.7.9 刹车功能

刹车功能常用于马达控制中。当刹车功能开启后，刹车源输入有效时，输出使能信号和输出电平将会依据刹车控制位的配置（TIM1_BDTR 寄存器中的 MOE、OSSI 和 OSSR 位，TIM1_CR2 寄存器中的 OISi 和

OISiN 位) 立即生效。TIM1 有 2 路刹车通道 TIM1_BKIN 和 TIM1_BKIN2, 各个通道输入刹车源不同, 且分别具有各自的刹车控制位, 互联配置寄存器 4 (SysCtrl_EDU_CFG4) 中给出了具体各个通道的刹车源。对于 TIM1_BKIN, 刹车源还包括系统产生的 2 个特殊刹车源, 分别为发生 CPU 内核锁死和电压过低触发 LVD 产生的低压检测标志, 这两个刹车源不受使能和极性控制, 具体描述请见芯片控制寄存器 ChipCtrl_CTRL.LOCKUPEN 和 ChipCtrl_CTRL.LVD_LOCK 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, TIM1_BDTR.MOE 位为低。配置 TIM1_BDTR.BKE 位可以使能 TIM1_BKIN 功能, 其刹车输入信号的极性可以通过配置 TIM1_BDTR.BKP 位选择, BKE 和 BKP 可以被同时修改; 同理可配置 TIM1_BDTR.BK2E 和 TIM1_BDTR.BK2P 分别进行 TIM1_BKIN2 使能和其刹车输入信号极性控制。

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 TIM1_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。

当需要使用 TIMER 和 ADC/ACMP 联动时, 可设置 TIM1_BDTR.BKTRIG 位选择连接 ADC/ACMP 的触发信号是否经过刹车控制。若设置 TIM1_BDTR.BKTRIG=1'b1, 则触发 ADC/ACMP 的 TIM1_trig_oci 为刹车处理前的信号, 此时若刹车输入信号有效, TIM1 会进入刹车状态, 但 ADC/ACMP 仍可被触发。反之, 设置 TIM1_BDTR.BKTRIG=1'b0 (默认), 则触发 ADC/ACMP 的 TIM1_trig_oci 为刹车处理后的信号, 此时若刹车输入信号有效, TIM1 会进入刹车状态, ADC/ACMP 不会被触发。

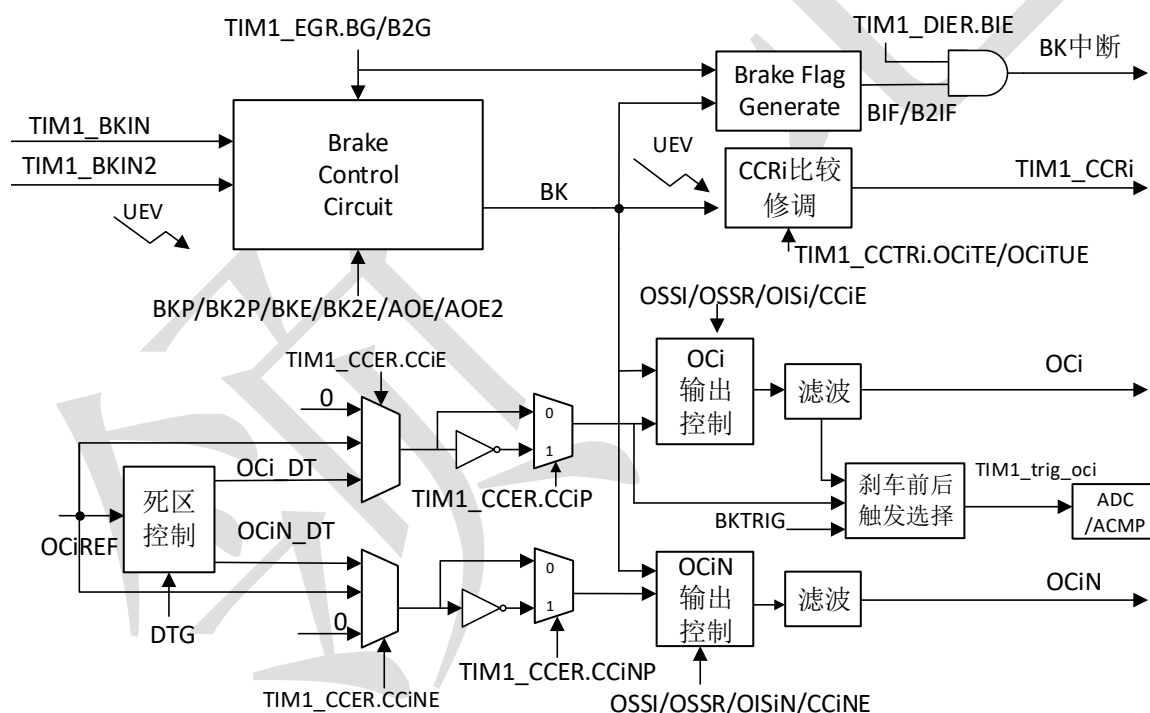


图 13-25 TIM1 刹车通道框图

例如当刹车通道 1 中有刹车源触发时:

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, 如果 OSSI=0, 定时器输出不使能; 如果 OSSI=1, 则定时器输出使能依 TIM1_CCER.CCiE/CCiNE 决定。但不管 OSSI=0/1, 输出都可以由 OISi/OISiN 位驱动, 具体情况请参考表 13-5。
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。
4. 刹车状态标志（TIM1_SR.BIF 位）被置起，且如果 TIM1_DIER.BIE 位使能，则会产生中断。
 5. 如果 TIM1_BDTR.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 TIM1_CCRi 寄存器的值可以通过相应的 TIM1_CCTRI 修调寄存器来更新：如果 TIM1_CCMRi.OCiTE=1，当更新事件 UEV 发生时，TIM1_CCRi 寄存器的影子寄存器被 TIM1_CCTRI 更新；如果 TIM1_CCMRi 寄存器中的 OCiTE=1 且 OCiTUE=1，当更新事件 UEV 发生时，TIM1_CCRi 寄存器的影子寄存器和预装载寄存器都被 TIM1_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，刹车标志 BIF/B2IF 不能被清除。

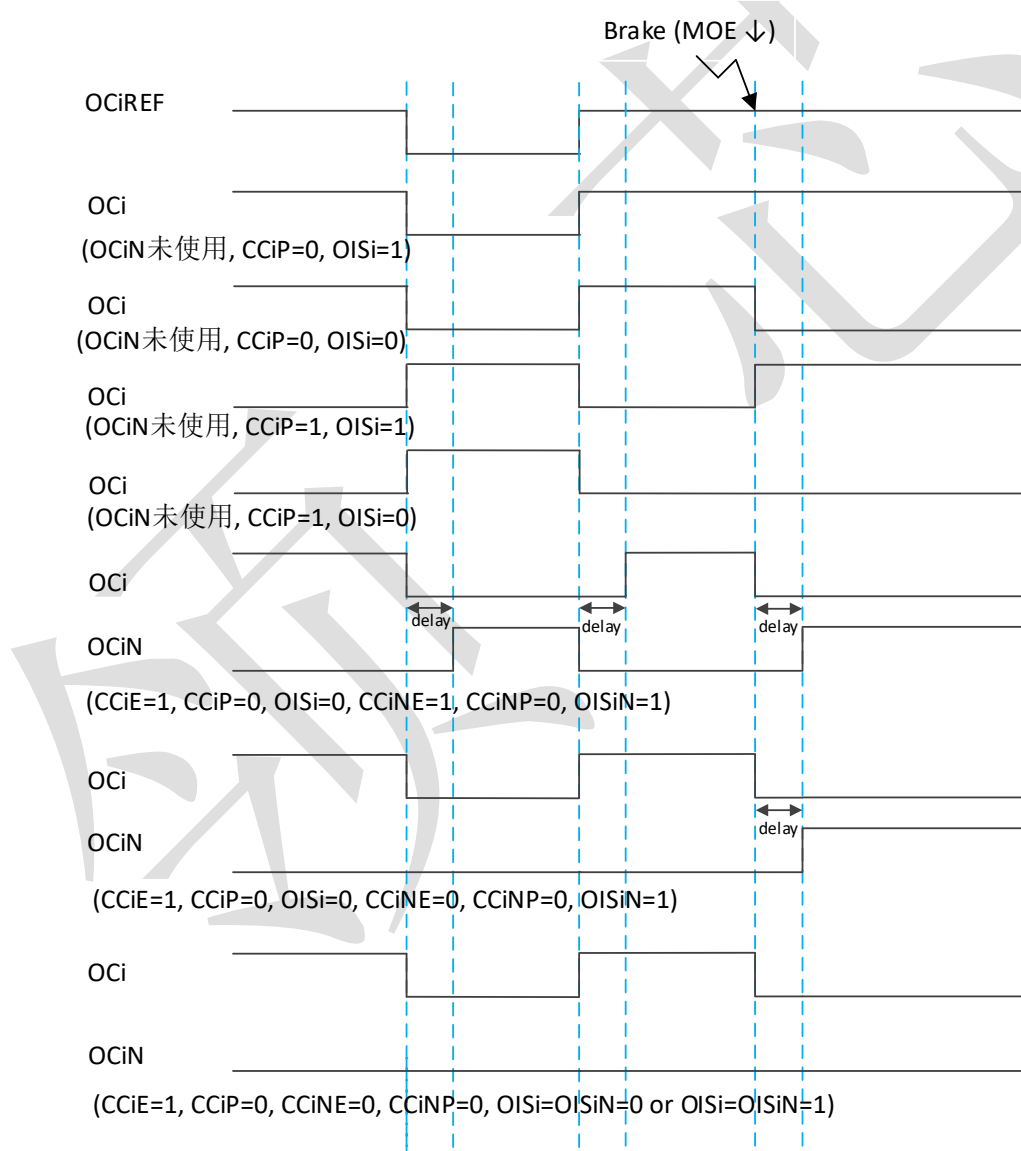


图 13-26 刹车对应的输出

有两种方式来产生刹车：

- 通过刹车输入并配合 TIM1_BDTR 寄存器中的 BKE/BK2E 位和 BKP/BK2P 位

- 通过软件配置 TIM1_EGR.BG/B2G 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 TIM1_BDTR.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。

13.7.10 通过外部事件清除 OCiREF

设定相应 TIM1_CCMRi.OCiCE 置为 1，则给定通道的 OCiREF 信号可以通过外部输入信号或内部其他信号来清除，并保持低电平，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 TIM1_SMCR.OCCS 位，可以在内部其他信号 OCREF_CLR 输入和外部输入信号 ETRF（滤波器后的 ETR）之间选择 OCiREF 清除信号的来源。如果选择了 ETRF，ETR 需要如下配置：

- 外部触发预分频器要保持关闭：TIM1_SMCR.ETPS=0
- 外部时钟源模式 2 关闭：TIM1_SMCR.ECE=0
- 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，可通过 TIM1 互联配置寄存器 4（SysCtrl_EDU_CFG4）配置。OCREF_CLR 的极性可以通过 TIM1_SMCR.OCCP 位配置。

13.7.11 PWM 生成

当使用互补输出时，可以通过 TIM1_CR2.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，使得用户可以提前为下一步的配置进行编程，并在 COM 通信事件发生时同时改变所有通道上的配置。COM 通信事件可以通过软件配置 TIM1_EGR.COMG 位产生，或当 TIM1_CR2.CCUS=1 时通过硬件在 TRGI 的上升沿产生。

当 COM 发生时，TIM1_SR.COMIF 位置起，它可以产生中断（如果 TIM1_DIER.COMIE 位被置位），或产生 DMA 请求（如果 TIM1_DIER.COMDE 位被置位）。

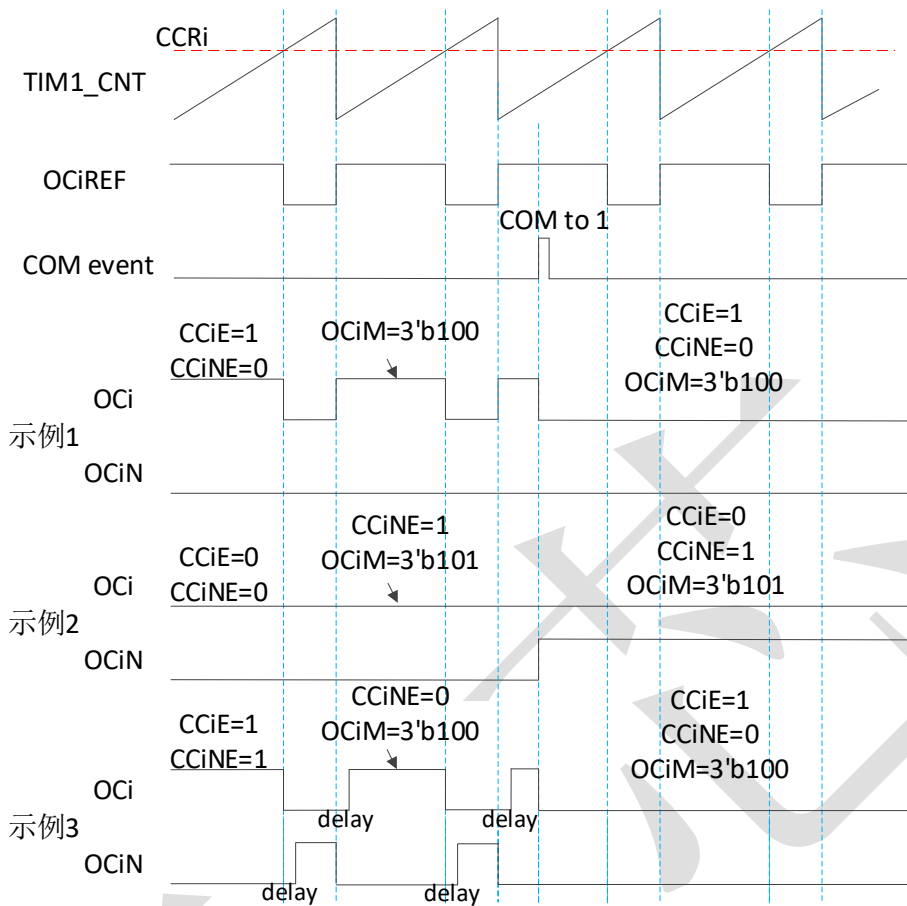


图 13-27 COM 触发的 PWM 生成示例

13.7.12 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 TIM2 定时器作为“接口定时器”来连接霍尔传感器，通过设置 TIM2_CR2.TI1S 位来选择 3 个“接口定时器”输入脚（CC1、CC2、CC3）相异或之后再连接到 TI1 输入通道，“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。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，并设置捕获信号为 TRC（TIM2_CCMR1.CC1S=2'b11）。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 通信事件）用于改变 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 TIM1。

举例：霍尔输入连接到 TIM2 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 TIM1 的 PWM 配置。

- 配置 TIM2_CR2.TI1S 位为 1，配置三个定时器输入逻辑或到 TI1 输入。
- 时基编程：配置 TIM2_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）：配置 TIM2_CCMR1.CC1S=2'b11，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：配置 TIM2_CCMR1 寄存器中的 OC2M=3'b111 和

CC2S=2'b00。

- 选择 OC2REF 作为 TRGO 上的触发输出：配置 TIM2_CR2.MMS=3'b101。

在 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，比较控制信号为预装载的（TIM1_CR2.CCPC=1），同时设置 COM 通信事件来源可以为硬件 TRGI 的上升沿（TIM1_CR2.CCUS=1）。在一次 COM 通信事件后，写入下一次的 TIM1 输出 PWM 的控制位（OCiM、CCiP、CCiNP、CCiE 和 CCiNE），这可以在处理 CMO 事件中断子程序里实现（如果 TIM1_DIER.COMIE 位被置位）。

13.8 TIM1 定时器与外部触发的同步

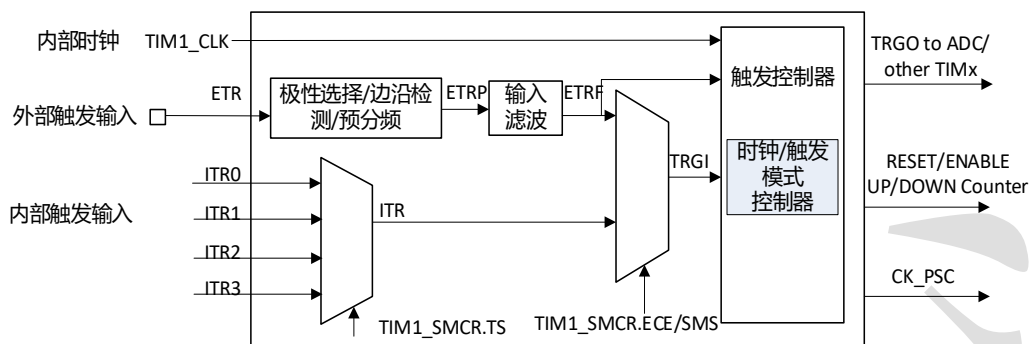


图 13-28 TIM1 触发输入功能框图

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。计数器允许两种触发输入：ETR（外部触发）；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。

13.8.1 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 计数器在输入 ETRF 的上升沿开始向上计数：

- 本例中不需要滤波器，配置 TIM1_SMCR.ETF=4'b0000
- 设置预分频器，配置 TIM1_SMCR.ETPS=2'b01
- 选择 ETR 的上升沿检测，配置 TIM1_SMCR.ETP=0
- 配置 TIM1_SMCR.SMS=3'b110，选择定时器为触发模式
- 配置 TIM1_SMCR.TS=3'b111，选择 ETRF 作为输入源

当 ETRF 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 TIM1_SR.TIF 位被置起。

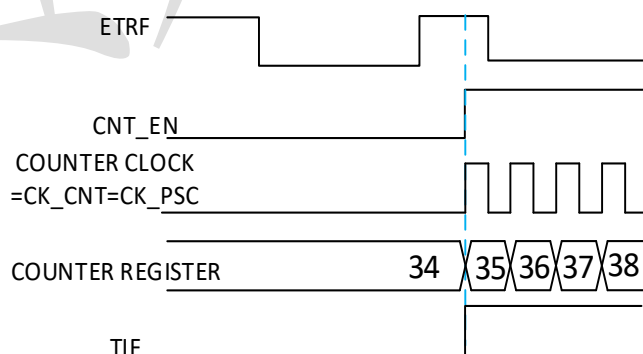


图 13-29 标准触发模式下的时序图

13.8.2 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 TIM1_CR1.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，ITR1 输入端的上升沿导致向上计数器被清零：

- 配置 TIM1_SMCR.SMS=3'b100，选择定时器为复位模式
- 配置 TIM1_SMCR.TS=3'b001，选择 ITR1 作为输入源
- 配置 TIM1_CR1.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 ITR1 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

下图显示当自动重装载寄存器 TIM1_ARR=0x36 时的动作。

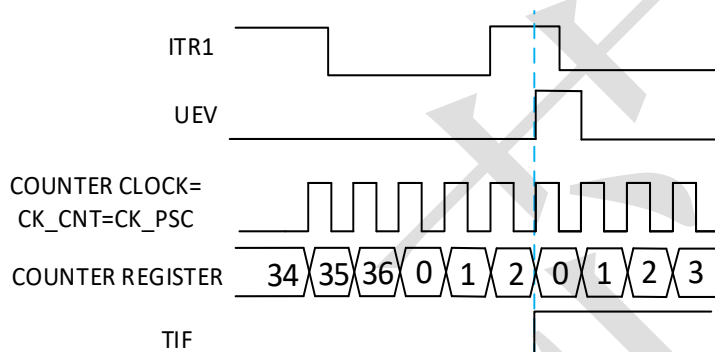


图 13-30 复位模式下的时序图

13.8.3 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 ITR1 为高时向上计数：

- 配置 TIM1_SMCR.SMS=3'b101，选择定时器为门控模式
- 配置 TIM1_SMCR.TS=3'b001，选择 ITR1 作为输入源
- 配置 TIM1_CR1.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）

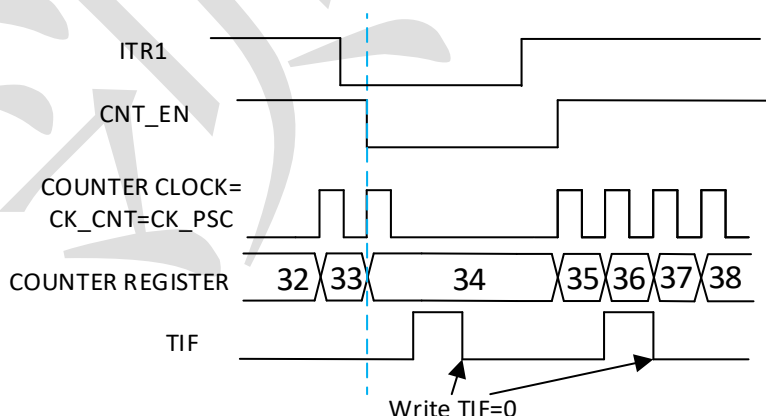


图 13-31 门控模式下的时序图

只要 ITR1 为高，计数器开始依据内部时钟计数，一旦 ITR1 变低则停止计数。当计数器开始或停止时，TIM1_SR.TIF 位都会被置起。

13.8.4 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 TIM1_CR1.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 TIM1_SMCR.SMS=4'b1000。

13.8.5 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（TIM1_SMCR.TS != 3'b111）。

在下面的例子中，一旦在 ITR1 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 TIM1_SMCR 寄存器配置外部触发输入电路。

- 首先配置 ETR：TIM1_SMCR 寄存器中配置 ETF=4'b0000 禁止滤波器，配置 ETPS=2'b00 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 TIM1_SMCR.SMS=3'b110，选择定时器为触发模式
- 配置 TIM1_SMCR.TS=3'b001，选择 ITR1 作为输入源

当 ITR1 上出现一个上升沿时，同时 TIM1_SR.TIF 位被置起，计数器开始在 ETR 的上升沿计数。ETR 信号的上升沿和计数器实际时钟之间的延时取决于 ETRP 输入端的同步电路。

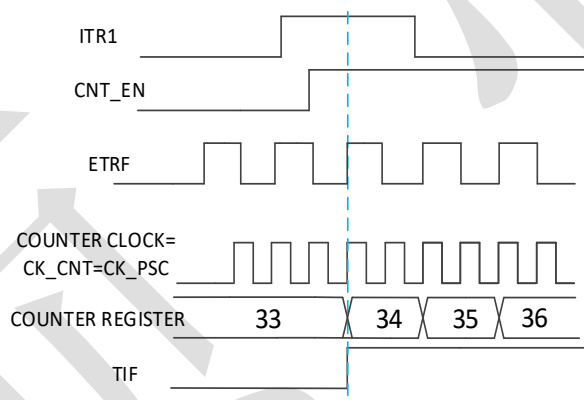


图 13-32 外部时钟模式 2+触发模式下的时序图

13.8.6 触发延迟和防误触发功能

在触发输入中，用户可根据需求设置触发延迟和防误触发功能。配置 TIM1_DLAMT.DTG 可设定触发输入的延迟时间，配置 TIM1_DLAMT.AMTG 可设定防误触发的窗口时间，并且配置 TIM1_DLAMT.AMTDL 位可选择防误触发窗口的开始时间。如果在防误触发窗口时间内出现新的触发输入，则会被舍弃。如果在触发延迟过程中出现新的触发输入，且不在防误触发窗口时间内，则新的触发输入会被保留，触发延迟过程将重新开始。如图 13-33 所示，在以下的例子中，设定选择 ITR1 的上升沿作为触发输入。

- 配置 TIM1_SMCR.SMS=3'b100，选择定时器为复位模式
- 配置 TIM1_SMCR.TS=3'b001，选择 ITR1 作为输入源
- 配置 TIM1_DLAMT.AMTDL=0，选择防误触发窗口时间从延迟的触发输入开始
- 配置 TIM1_DLAMT.DTG=0x7F，TIM1_DLAMT.AMTG=0x5F。

当 ITR1 出现一个上升沿时，内部触发延迟计数器开始计数，在触发延迟过程中，出现新的触发输入，将导致内部触发延迟计数器（delay_cnt）重新开始计数，延迟窗口加长，最终经触发延迟后，计数器被清

零然后从 0 重新开始计数，同时 TIM1_SR.TIF 位被置起，在防误触发窗口时间内出现新触发输入将会被舍弃。

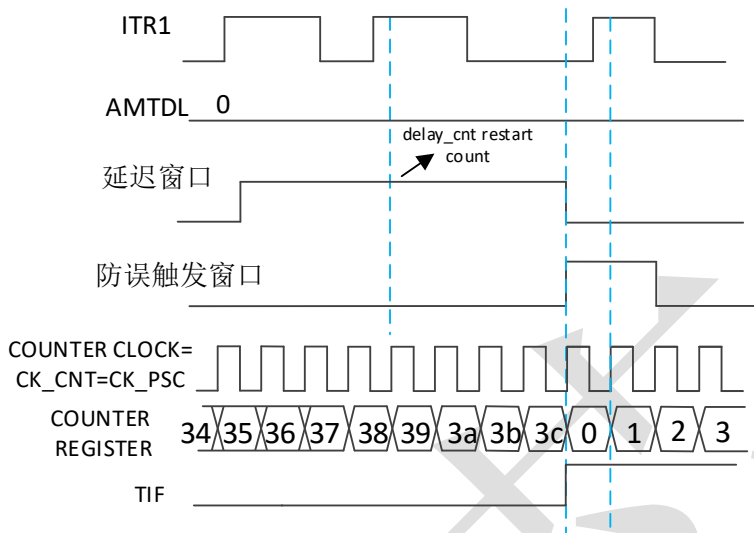


图 13-33 防误触发窗口从延迟的触发输入开始

当选择防误触发窗口从输入触发开始时（配置 TIM1_DLAMT.AMTDL=1），如图 13-34 所示。在防误触发窗口中新的触发输入将会被舍弃，内部延迟计数器继续计数，若在防误触发窗口之外，且仍处于触发延迟过程中再次出现新的触发输入，则内部延迟计数器将重新开始计数，且防误触发功能将会被激活。直到触发延迟结束后，计数器被清零然后从 0 重新开始计数，同时 TIM1_SR.TIF 位被置起。

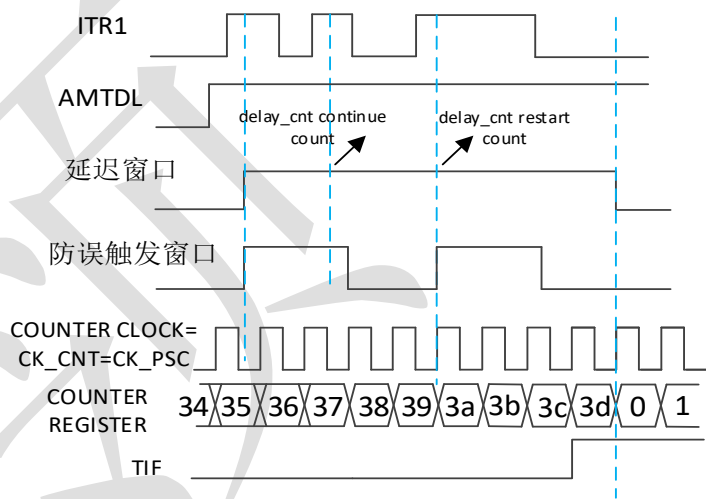


图 13-34 防误触发窗口从触发输入开始

另外，可配置 TIM1_DLAMT.DLS=1，可在触发延时过程中清除 OCiREF，注意在输出比较模式中若选择强制输出模式，则此功能无效。

13.9 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 13-28。

TIM1 的输入触发可以来自芯片内部的其他定时器和比较器 ACMP0/1 输出；TIM1 的输出 TRGO 可触发



芯片内部的 ADC、DAC 和其他定时器；TIM1 的特殊输出 TIM1_trig_oci 可触发芯片内部的 ADC 和其他定时器，它可以选择刹车控制前或刹车控制后的 OCi 信号，具体详见 TIM1_BDTR.BKTRIG 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	tim2_trgo	-
ITR1	内部触发 1	tim15_trgo	-
ITR2	内部触发 2	tim16_trig_oc1	-
ITR3	内部触发 3	tim17_trig_oc1	-
外部触发输入 ETR			
ETR	外部触发	tim1_gpio_etr/tim2_trgo/ ACMP0_out/ACMP1_out/ tim15_trig_oc1 / tim16_trig_oc1	SysCtrl_EDU_CFG1

13.10 TIM1 中断

TIM1 有 9 个中断请求源，前四个映射到 TIM1_NON_CC 中断，后五个映射到 TIM1_CC 中断：

- 刹车中断
- 更新事件中断
- 触发中断
- COM 事件中断
- 输出比较 1 中断
- 输出比较 2 中断
- 输出比较 3 中断
- 输出比较 4 中断
- 输出比较 5 中断

为了使用中断特性，对每个被使用的中断通道，设置 TIM1_DIER 寄存器中相应的中断使能位：BIE、TIE、COMIE、CCiE 和 UIE 位。

通过设置 TIM1_EGR 寄存器中的相应位，也可以用软件产生上述各个中断源。

13.11 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	比较模式寄存器 1	R/W	0x0000_0000	0x4001_0018
CCMR2	比较模式寄存器 2	R/W	0x0000_0000	0x4001_001C
CCER	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	比较寄存器 1	R/W	0x0000_0000	0x4001_0034
CCR2	比较寄存器 2	R/W	0x0000_0000	0x4001_0038
CCR3	比较寄存器 3	R/W	0x0000_0000	0x4001_003C
CCR4	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
CCMR3	比较模式寄存器 3	R/W	0x0000_0000	0x4001_0054
CCR5	比较寄存器 5	R/W	0x0000_0000	0x4001_0058
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
CCTR5	比较修调寄存器 5	R/W	0x0000_0000	0x4001_0074
DLAMT	触发延迟和防误触发寄存器	R/W	0x0001_0000	0x4001_007C
CCOR1	比较偏移寄存器 1	R/W	0x0000_0000	0x4001_0080
CCOR2	比较偏移寄存器 2	R/W	0x0000_0000	0x4001_0084
CCOR3	比较偏移寄存器 3	R/W	0x0000_0000	0x4001_0088
CCOR4	比较偏移寄存器 4	R/W	0x0000_0000	0x4001_008C
CCOR5	比较偏移寄存器 5	R/W	0x0000_0000	0x4001_0090
OCR	比较偏移控制寄存器	R/W	0x0003_3333	0x4001_0098
OMR	输出模式选择寄存器	R/W	0x7654_3210	0x4001_009C
SysCtrl_EDU_CFG1	TIM1 互联配置寄存器 1	R/W	0x0000_0000	0x4800_7024
SysCtrl_EDU_CFG4	TIM1 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）。

13.11.1 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	Res.				ETRE	FTE
							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 21:18	保留，必须保持复位值
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: ETR 数字滤波器的 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 ETR 数字滤波器使用
	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。

13.11.2 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.															OIS5
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OIS4N	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	Res.	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	OIS5： 输出空闲状态 5（OC5 输出），参考 OIS1
Bit 15	OIS4N： 输出空闲状态 4（OC4N 输出），参考 OIS1N
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	保留, 必须保持复位值
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 : 比较 DMA 选择
		0: 当比较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 : 比较控制位的更新控制选择
		0: 当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 : 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
	注： 该位只对拥有互补输出的通道有效。

13.11.3 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 决不能与 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[1:0]： 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 TIM2_TRGO (默认)
	001: 内部触发 ITR1, 连接 TIM15_TRGO
	010: 内部触发 ITR2, 连接 TIM16_trig_oc1
	011: 内部触发 ITR3, 连接 TIM17_trig_oc1
	100: 保留
	101: 保留
	110: 保留
	111: 外部触发输入 (ETRF)
	注: 这些位只能在未用到 (如 $\text{SMS}=000$) 时被改变, 以避免在改变时产生错误的边沿检测。
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 $\text{CEN}=1$, 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。

13.11.4 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.													CC5DE	Res.	CC5IE
													rw		rw

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	31:19	保留, 必须保持复位值
Bit	18	CC5DE: 比较通道 5 的 DMA 请求使能
		0: 比较通道 5 的 DMA 请求禁止 (默认)
		1: 比较通道 5 的 DMA 请求使能
Bit	17	保留, 必须保持复位值
Bit	16	CC5IE: 比较通道 5 中断使能
		0: 比较通道 5 中断禁止 (默认)
		1: 比较通道 5 中断使能
Bit	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: 比较通道 4 的 DMA 请求使能
		0: 比较通道 4 的 DMA 请求禁止 (默认)
		1: 比较通道 4 的 DMA 请求使能
Bit	11	CC3DE: 比较通道 3 的 DMA 请求使能
		0: 比较通道 3 的 DMA 请求禁止 (默认)
		1: 比较通道 3 的 DMA 请求使能
Bit	10	CC2DE: 比较通道 2 的 DMA 请求使能
		0: 比较通道 2 的 DMA 请求禁止 (默认)

	1: 比较通道 2 的 DMA 请求使能
Bit 9	CC1DE : 比较通道 1 的 DMA 请求使能
	0: 比较通道 1 的 DMA 请求禁止 (默认)
	1: 比较通道 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 : 比较通道 4 中断使能
	0: 比较通道 4 中断禁止 (默认)
	1: 比较通道 4 中断使能
Bit 3	CC3IE : 比较通道 3 中断使能
	0: 比较通道 3 中断禁止 (默认)
	1: 比较通道 3 中断使能
Bit 2	CC2IE : 比较通道 2 中断使能
	0: 比较通道 2 中断禁止 (默认)
	1: 比较通道 2 中断使能
Bit 1	CC1IE : 比较通道 1 中断使能
	0: 比较通道 1 中断禁止 (默认)
	1: 比较通道 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

13.11.5 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.															CC5IF
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		CC5OF	CC4OF	CC3OF	CC2OF	CC1OF	B2IF	B1F	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
		r	r	r	r	r	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:17	保留，必须保持复位值
Bit 16	CC5IF : 比较通道 5 中断标志，参考 CC1IF 位
Bits 15:14	保留，必须保持复位值
Bit 13	CC5OF : 比较通道 5 过比较标志，参考 CC1OF 位
Bit 12	CC4OF : 比较通道 4 过比较标志，参考 CC1OF 位
Bit 11	CC3OF : 比较通道 3 过比较标志，参考 CC1OF 位
Bit 10	CC2OF : 比较通道 2 过比较标志，参考 CC1OF 位
Bit 9	CC1OF : 比较通道 1 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: 无过比较产生（默认）
	1: 计数器的值与 TIM1_CCR1 寄存器匹配时 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	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 : 比较通道 4 中断标志，参考 CC1IF 位
Bit 3	CC3IF : 比较通道 3 中断标志，参考 CC1IF 位
Bit 2	CC2IF : 比较通道 2 中断标志，参考 CC1IF 位
Bit 1	CC1IF : 比较通道 1 中断标志
	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。
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

13.11.6 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.						CC5G	B2G	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
						w	w	w	w	w	w	w	w	w	w

Bits 31:10	保留，必须保持复位值
Bit 9	CC5G ：产生比较 5 事件，参考 CC1G 位
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 ：产生比较控制更新事件
	该位由软件置 1，用于产生一个比较控制更新事件，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
	注： 该位只对拥有互补输出的通道有效。
Bit 4	CC4G ：产生比较 4 事件，参考 CC1G 位
Bit 3	CC3G ：产生比较 3 事件，参考 CC1G 位

Bit 2	CC2G: 产生比较 2 事件, 参考 CC1G 位
Bit 1	CC1G: 产生比较 1 事件
	该位由软件置 1, 用于产生一个比较事件, 由硬件自动清 0
	0: 无动作 (默认)
	1: 在通道 CC1 上产生一个比较事件
	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

13.11.7 CCMR1 比较模式寄存器 1 (TIM1_CCMR1)

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
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]	
rw	rw			rw	rw	rw		rw	rw			rw	rw	rw	

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 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
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]: 比较通道 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)

	其他：保留
	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。
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/TIM1_CCOR1 寄存器的预装载功能，可随时写入 TIM1_CCR1/TIM1_CCOR1 寄存器，并且新写入的数值立即起作用（默认）
	1：使能 TIM1_CCR1/TIM1_CCOR1 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/TIM1_CCOR1 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/CCOR1 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]：比较通道 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）
	其他：保留
	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。

13.11.8 CCMR2 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]	
rw	rw			rw	rw	rw		rw	rw			rw	rw	rw	

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]：比较通道 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）
	其他：保留
	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。
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] : 比较通道 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)
		其他: 保留
		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。

13.11.9 CCMR3 比较模式寄存器 3 (TIM1_CCMR3)

地址偏移: 0x54

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.								OC5TE	OC5TUE	Res.					
								rw	rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								OC5CE	OC5M[2:0]			OC5PE	OC5FE	CC5S[1:0]	
								rw	rw			rw	rw	rw	

Bits	31:24	保留, 必须保持复位值
Bit	23	OC5TE : 输出比较 5 修调使能, 参考 OC1TE
Bit	22	OC5TUE : 输出比较 5 修调更新使能, 参考 OC1TUE
Bits	21:8	保留, 必须保持复位值
Bit	7	OC5CE : 输出比较 5 清除使能, 参考 OC1CE
Bits	6:4	OC5M[2:0] : 输出比较 5 模式, 参考 OC1M
Bit	3	OC5PE : 输出比较 5 预装载使能, 参考 OC1PE
Bit	2	OC5FE : 输出比较 5 快速使能, 参考 OC1FE
Bits	1:0	CC5S[1:0] : 比较通道 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: CC5 通道被配置为输出 (默认)
		其他: 保留
		注 : CC5S 仅在通道关闭时 (TIM1_CCER 寄存器的 CC5E=0、CC5NE=0 且已被更新) 才是可写的。

13.11.10 CCER 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.												CC5NP	CC5NE	CC5P	CC5E
												rw	rw	rw	rw

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	31:20	保留，必须保持复位值
Bit	19	CC5NP : 比较通道 5 互补输出极性，参考 CC1NP
Bit	18	CC5NE : 比较通道 5 互补输出使能，参考 CC1NE
Bit	17	CC5P : 比较通道 5 极性，参考 CC1P
Bit	16	CC5E : 比较通道 5 使能，参考 CC1E
Bit	15	CC4NP : 比较通道 4 互补输出极性，参考 CC1NP
Bit	14	CC4NE : 比较通道 4 互补输出使能，参考 CC1NE
Bit	13	CC4P : 比较通道 4 极性，参考 CC1P
Bit	12	CC4E : 比较通道 4 使能，参考 CC1E
Bit	11	CC3NP : 比较通道 3 互补输出极性，参考 CC1NP
Bit	10	CC3NE : 比较通道 3 互补输出使能，参考 CC1NE
Bit	9	CC3P : 比较通道 3 极性，参考 CC1P
Bit	8	CC3E : 比较通道 3 使能，参考 CC1E
Bit	7	CC2NP : 比较通道 2 互补输出极性，参考 CC1NP
Bit	6	CC2NE : 比较通道 2 互补输出使能，参考 CC1NE
Bit	5	CC2P : 比较通道 2 极性，参考 CC1P
Bit	4	CC2E : 比较通道 2 使能，参考 CC1E
Bit	3	CC1NP : 比较通道 1 互补输出极性
		0: OC1N 高电平有效（默认）
		1: OC1N 低电平有效
		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 : 比较通道 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 : 比较通道 1 极性
		0: OC1 高电平有效（默认）
		1: OC1 低电平有效
		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 : 比较通道 1 使能

	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。
	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	IO 输出禁止 (TIM1 管脚 IO 输出为高阻态) OCi=0, OCiN=0	
		0	0	1	IO 输出禁止 (TIM1 管脚 IO 输出为高阻态), OCi=0	OCiREF + 极性, OCiN=OCiREF xor CCiNP
		0	1	0	OCiREF + 极性, OCi=OCiREF xor CCiP	IO 输出禁止 (TIM1 管脚 IO 输出为高阻态), OCiN=0
		0	1	1	OCiREF + 极性 + 死区	OCiREF 反相 + 极性 + 死区
		1	0	0	IO 输出禁止 (TIM1 管脚 IO 输出为高阻态) OCi=CCiP, OCiN=CCiNP	
		1	0	1	关闭状态 (TIM1 输出无效电平), OCi=CCiP	OCiREF + 极性, OCiN=OCiREF xor CCiNP
		1	1	0	OCiREF + 极性, OCi=OCiREF xor CCiP	关闭状态 (TIM1 输出无效电平), OCiN=CCiNP
		1	1	1	OCiREF + 极性 + 死区	OCiREF 反相 + 极性 + 死区
0	0	X	0	0	IO 输出禁止 (TIM1 管脚 IO 输出为高阻态) OCi=CCiP, OCiN=CCiNP	
	0		0	1	首先处于 IO 输出禁止状态 (TIM1 通道 IO 输出为高阻态), OCi=CCiP, OCiN=CCiNP, 死区时钟存在且经过死区时间后, 如果 CCiP≠OISi 且 CCiNP≠OISiN, 则继续输出无效电平, 即 OCi=CCiP, OCiN=CCiNP; 否则输出空闲电平, 即 OCi=OISi, OCiN=OISiN	
	0		1	0		
	0		1	1		
	1		0	0	IO 输出禁止 (TIM1 管脚 IO 输出为高阻态) OCi=CCiP, OCiN=CCiNP	
	1		1	0	首先处于关闭状态 (TIM1 输出无效电平), OCi=CCiP, OCiN=CCiNP, 死区时钟存在且经过死区时间后, 如果 CCiP≠OISi 且 CCiNP≠OISiN, 则继续输出无效电平, 即 OCi=CCiP, OCiN=CCiNP; 否则输出空闲电平, 即 OCi=OISi, OCiN=OISiN	
	1		0	1		
	1		1	1		

注: 连接 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 寄存器配置。

13.11.11 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
-----------	------------------

13.11.12 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.															

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。

13.11.13 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.															

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。

13.11.14 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：
	– 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

13.11.15 CCR1 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] : 比较通道 1 的值
	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_CCR1 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_CCR1 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_CCR1 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。

13.11.16 CCR2 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]: 比较通道 2 的值
	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_CCR2 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_CCR2 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_CCR2 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。

13.11.17 CCR3 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]: 比较通道 3 的值
	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。

13.11.18 CCR4 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] ：比较通道 4 的值
	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。

13.11.19 CCR5 比较寄存器 5（TIM1_CCR5）

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.															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR5[15:0]															
rw															

Bits 31:16	保留，必须保持复位值
Bits 15:0	CCR5[15:0]：比较通道 5 的值
	CCR5 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 OC5 端口上产生输出信号。
	如果在 OC5PE 位（TIM1_CCMR3 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，CCR5 的值可以通过相应的 TIM1_CCR5 修调寄存器来更新：如果 TIM1_CCMR3 寄存器中的 OC5TE=1，当更新事件 UEV 发生时，CCR5 的影子寄存器被 TIM1_CCR5 的值更新；如果 TIM1_CCMR3 寄存器中的 OC5TE=1 且 OC5TUE=1，当更新事件 UEV 发生时，CCR5 的影子寄存器和预装载寄存器都被 TIM1_CCR5 的值更新。
	可以通过 AS 位（TIM1_CR1 寄存器）来选择读取 CCR5 寄存器的值来自影子寄存器或预装载寄存器。

13.11.20 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.				BKTRIG	AOE2	BK2P	BK2E	Res.							
				rw	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 31:28	保留，必须保持复位值
Bit 27	BKTRIG ：TIM1_trig_oci 的触发源选择
	TIM1_BDTR.BKTRIG 位选择连接 ADC/ACMP 的触发信号是否经过刹车控制。若设置 TIM1_BDTR.BKTRIG=1'b1，则触发 ADC/ACMP 的 TIM1_trig_oci 为刹车处理前的信号，此时若刹车输入信号有效，TIM1 会进入刹车状态，但 ADC/ACMP 仍可被触发。反之，设置 TIM1_BDTR.BKTRIG=1'b0，则触发 ADC/ACMP 的 TIM1_trig_oci 为刹车处理后的信号，此时若刹车输入信号有效，TIM1 会进入刹车状态，ADC/ACMP 不会被触发。
	0：刹车控制处理后的 OC _i （默认）
	1：刹车控制处理前的 OC _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。
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）
	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。
Bit 25	BK2P: TIM1_BKIN2 输入极性
	0: TIM1_BKIN2 输入低电平有效（默认）
	1: TIM1_BKIN2 输入高电平有效
	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。
Bit 24	BK2E: TIM1_BKIN2 功能使能
	0: TIM1_BKIN2 功能禁止（默认）
	1: TIM1_BKIN2 功能使能
	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。
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）
	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。
Bit 13	BKP: TIM1_BKIN 输入极性
	0: TIM1_BKIN 输入低电平有效（默认）
	1: TIM1_BKIN 输入高电平有效
	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。
Bit 12	BKE: TIM1_BKIN 功能使能
	0: TIM1_BKIN 功能禁止（默认）
	1: TIM1_BKIN 功能使能
	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。
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 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
	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。
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 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
	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。
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、BK2E、BKP、BK2P、

	AOE、AOE2、DTG 位，TIM1_CR2 寄存器的 OISi/OISiN 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 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，TIM1_CCMRi 寄存器的 OCiM/OCiPE 位）
	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。
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。
	$DTG[7:5]=0xx \Rightarrow DT=DTG[7:0] \times t_{dtg}$ ，其中 $t_{dtg} = t_{DTS}$
	$DTG[7:5]=10x \Rightarrow DT=(64+DTG[5:0]) \times t_{dtg}$ ，其中 $t_{dtg} = 2 \times t_{DTS}$
	$DTG[7:5]=110 \Rightarrow DT=(32+DTG[4:0]) \times t_{dtg}$ ，其中 $t_{dtg} = 8 \times t_{DTS}$
	$DTG[7:5]=111 \Rightarrow DT=(32+DTG[4:0]) \times t_{dtg}$ ，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 15.875 us，步长时间为 125 ns
	DTG[7:0] = 80H 到 BFH，16 us 到 31.75 us，步长时间为 250 ns
	DTG[7:0] = C0H 到 DFH，32 us 到 63 us，步长时间为 1 us
	DTG[7:0] = E0H 到 FFH，64 us 到 126 us，步长时间为 2 us
	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。

13.11.21 DLAMT 触发延迟和防误触发寄存器（TIM1_DLAMT）

地址偏移：0x7C

复位值：0x0001_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
AMTG[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: 防误触发窗口开始时间
	0: 从延迟的触发输入开始
	1: 从触发输入开始（默认）
	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	AMTG[7:0]: 防误触发窗口时间设置。在防误触发窗口的时间内如果出现新的触发输入, 则会舍弃当前触发输入, 实现防误触发功能。
	该位定义了防误触发窗口的时间, 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 $t_{amtg}=2\times t_{DTS}$
	AMTG[7:5]=110 => AMT=(32+AMTG[4:0]) $\times t_{amtg}$, 其中 $t_{amtg}=8\times t_{DTS}$
	AMTG[7:5]=111 => AMT=(32+AMTG[4:0]) $\times t_{amtg}$, 其中 $t_{amtg}=16\times t_{DTS}$
	举例, 如果 $t_{DTS}=125\text{ ns}$ (8 MHz), 可能的防误触发窗口时间为:
	AMTG[7:0] = 0 到 7FH, 0 到 15.875 μs , 步长时间为 125 ns
	AMTG[7:0] = 80H 到 BFH, 16 μs 到 31.75 μs , 步长时间为 250 ns
	AMTG[7:0] = C0H 到 DFH, 32 μs 到 63 μs , 步长时间为 1 μs
	AMTG[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。
Bits 7:0	DLG[7:0]: 触发延迟窗口时间设置, 如果在触发延迟过程中有新的触发输入, 且不在防误触发窗口时间内, 则会把新的触发保留住, 触发延迟过程将重新进行。
	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]) $\times t_{dlg}$, 其中 $t_{dlg}=2\times t_{DTS}$
	DLG[7:5]=110 => DLT=(32+DLG[4:0]) $\times t_{dlg}$, 其中 $t_{dlg}=8\times t_{DTS}$
	DLG[7:5]=111 => DLT=(32+DLG[4:0]) $\times t_{dlg}$, 其中 $t_{dlg}=16\times t_{DTS}$
	举例, 如果 $t_{DTS}=125\text{ ns}$ (8 MHz), 可能的触发延迟窗口时间为:
	DLG[7:0] = 0 到 7FH, 0 到 15.875 μs , 步长时间为 125 ns
	DLG[7:0] = 80H 到 BFH, 16 μs 到 31.75 μs , 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。

13.11.22 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 连续传输长度
	定义了 DMA 在连续模式下的传输长度 (当对 TIM1_DMAR 寄存器的地址进行读或写时, TIM1 进行一次连续传输), 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					
	...					
	11111: 32 字节					
Bits 7:5	DSEL[2:0]: DMA 连续传输请求选择，默认为 000					
	定义了 TIM1 DMA 请求 1/2/3/4 的具体内容，以及哪个 DMA 请求使用连续传输，其它的未选中的则使用单次传输					
		DMA 请求 1	DMA 请求 2	DMA 请求 3	DMA 请求 4	连续传输
	000:	TIM1_CH1	TIM1_CH2	TIM1_CH3 TIM1_UP	TIM1_CH4 TIM1_TRIG TIM1_COM	使用单次传输
	001:					DMA 请求 1
	010:					DMA 请求 2
	011:					DMA 请求 3
	100:					DMA 请求 4
	101:	TIM1_CH5				DMA 请求 2
	110:	TIM1_CH1	TIM1_CH5			DMA 请求 2
	111:		TIM1_CH2		TIM1_CH5	DMA 请求 2
Bits 4:0	DBA[4:0]: DMA 基地址，默认为 00000					
	该位定义了 DMA 在连续模式下的基地址（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。					

13.11.23 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[31:0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	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) x4，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。
--

13.11.24 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。

13.11.25 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。

13.11.26 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。

13.11.27 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。

13.11.28 CCTR5 比较修调寄存器 5 (TIM1_CCTR5)

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
CCTR5[15:0]															
rw															

Bits 31:16	保留，必须保持复位值
Bits 15:0	CCTR5[15:0]：比较 5 修调值
	通道 CC5 作为输出时，如果刹车触发且对应的 AOE 位使能，TIM1_CCMR3 寄存器中的 OC5TE=1，当更新事件 UEV 发生时，TIM1_CCR5 的影子寄存器被 TIM1_CCTR5 的值更新；如果 TIM1_CCMR3 寄存器中的 OC5TE=1 且 OC5TUE=1，当更新事件 UEV 发生时，TIM1_CCR5 的影子寄存器和预装载寄存器都被 TIM1_CCTR5 的值更新。

13.11.29 CCOR1 比较偏移寄存器 1（TIM1_CCOR1）

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
Res.															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCOR1[15:0]															
rw															

Bits 31:16	保留，必须保持复位值
Bits 15:0	CCOR1[15:0]：比较 1 偏移值
	当通道 CC1 配置为输出且输出模式配置为 PWM1/2 模式时：CC1 通道的比较值由 CCR1 和 CCOR1 共同决定。
	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。
	可以通过 AS 位（TIM1_CR1 寄存器）来选择读取 CCOR1 寄存器的值来自影子寄存器或预装载寄存器。

13.11.30 CCOR2 比较偏移寄存器 2（TIM1_CCOR2）

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
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

CCOR2[15:0]
rw

Bits 31:16	保留，必须保持复位值
Bits 15:0	CCOR2[15:0]: 比较 2 偏移值
	当通道 CC2 配置为输出且输出模式配置为 PWM1/2 模式时：CC2 通道的比较值由 CCR2 和 CCOR2 共同决定。
	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。
	可以通过 AS 位（TIM1_CR1 寄存器）来选择读取 CCOR2 寄存器的值来自影子寄存器或预装载寄存器。

13.11.31 CCOR3 比较偏移寄存器 3（TIM1_CCOR3）

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
CCOR3[15:0]															
rw															

Bits 31:16	保留，必须保持复位值
Bits 15:0	CCOR3[15:0]: 比较 3 偏移值
	当通道 CC3 配置为输出且输出模式配置为 PWM1/2 模式时：CC3 通道的比较值由 CCR3 和 CCOR3 共同决定。
	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。
	可以通过 AS 位（TIM1_CR1 寄存器）来选择读取 CCOR3 寄存器的值来自影子寄存器或预装载寄存器。

13.11.32 CCOR4 比较偏移寄存器 4（TIM1_CCOR4）

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.															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCOR4[15:0]															
rw															

Bits 31:16	保留，必须保持复位值
Bits 15:0	CCOR4[15:0]: 比较 4 偏移值
	当通道 CC4 配置为输出且输出模式配置为 PWM1/2 模式时：CC4 通道的比较值由 CCR4 和 CCOR4 共同决定。
	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。
	可以通过 AS 位（TIM1_CR1 寄存器）来选择读取 CCOR4 寄存器的值来自影子寄存器或预装载寄存器。

13.11.33 CCOR5 比较偏移寄存器 5（TIM1_CCOR5）

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.															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCOR5[15:0]															
rw															

Bits 31:16	保留，必须保持复位值
Bits 15:0	CCOR5[15:0]: 比较 5 偏移值
	当通道 CC5 配置为输出且输出模式配置为 PWM1/2 模式时：CC5 通道的比较值由 CCR5 和 CCOR5 共同决定。
	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。
	可以通过 AS 位（TIM1_CR1 寄存器）来选择读取 CCOR5 寄存器的值来自影子寄存器或预装载寄存器。

13.11.34 OCR 比较偏移控制寄存器（TIM1_OCR）

地址偏移：0x98

复位值：0x0003_3333

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.												OC5_MODE[3:0]			
												rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4_MODE[3:0]				OC3_MODE[3:0]				OC2_MODE[3:0]				OC1_MODE[3:0]			
rw				rw				rw				rw			

Bits 31:20	保留，必须保持复位值
Bits 19:16	OC5_MODE[3:0] : 通道 5 比较偏移模式选择，参考 OC1_MODE
Bits 15:12	OC4_MODE[3:0] : 通道 4 比较偏移模式选择，参考 OC1_MODE
Bits 11:8	OC3_MODE[3:0] : 通道 3 比较偏移模式选择，参考 OC1_MODE
Bits 7:4	OC2_MODE[3:0] : 通道 2 比较偏移模式选择，参考 OC1_MODE
Bits 3:0	OC1_MODE[3:0] : 通道 1 比较偏移模式选择
	当通道 CC1 配置为输出且计数模式为中央对齐模式（CMS!=0）时，通过 OC1_MODE 选择 CCR1 和 CCOR1 是否作为 CC1 通道的比较值。
	Bit3: 在计数器向下计数过程中，使用 CCOR1 作为 CC1 通道的比较值
	Bit2: 在计数器向上计数过程中，使用 CCOR1 作为 CC1 通道的比较值
	Bit1: 在计数器向下计数过程中，使用 CCR1 作为 CC1 通道的比较值
	Bit0: 在计数器向上计数过程中，使用 CCR1 作为 CC1 通道的比较值

13.11.35 OMR 输出模式寄存器（TIM1_OMR）

地址偏移：0x9C

复位值：0x7654_3210

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PWM8_SEL[3:0]				PWM7_SEL[3:0]				PWM6_SEL[3:0]				PWM5_SEL[3:0]			
rw				rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWM4_SEL[3:0]				PWM3_SEL[3:0]				PWM2_SEL[3:0]				PWM1_SEL[3:0]			
rw				rw				rw				rw			

Bits 31:28	PWM8_SEL[3:0] : PWM8 输出映射，参考 PWM1_SEL
Bits 27:24	PWM7_SEL[3:0] : PWM7 输出映射，参考 PWM1_SEL
Bits 23:20	PWM6_SEL[3:0] : PWM6 输出映射，参考 PWM1_SEL
Bits 19:16	PWM5_SEL[3:0] : PWM5 输出映射，参考 PWM1_SEL
Bits 15:12	PWM4_SEL[3:0] : PWM4 输出映射，参考 PWM1_SEL
Bits 11:8	PWM3_SEL[3:0] : PWM3 输出映射，参考 PWM1_SEL
Bits 7:4	PWM2_SEL[3:0] : PWM2 输出映射，参考 PWM1_SEL
Bits 3:0	PWM1_SEL[3:0] : PWM1 输出映射
	通过 PWM1_SEL[3:0]选择从 TIM1_PWM1 端口输出的 OC 信号。
	PWM1_SEL[3:0] = 0~3: 对应 OC1~OC4
	PWM1_SEL[3:0] = 4~7: 对应 OC1N~OC4N
	其他值：保留

13.11.36 TIM1 互联配置寄存器 1 (SysCtrl_EDU_CFG1)

SysCtrl_EDU_CFG1 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 0x05FA659A 到 SysCtrl_KEY 密钥寄存器，且在随后的 64 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。

地址：0x4800_7024

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.															

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: TIM2_TRGO
	010: ACMP0 输出
	011: ACMP1 输出
	100: 保留
	101: TIM15_TRIG_OC1
	110: TIM16_TRIG_OC1
	111: 保留
Bits 15:0	保留，必须保持复位值
注： TIMx_TRIG_OC1 和 TIMx_TRGO 不同，详见 TIM 章节说明。	

13.11.37 TIM1 互联配置寄存器 4 (SysCtrl_EDU_CFG4)

SysCtrl_EDU_CFG4 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 0x05FA659A 到 SysCtrl_KEY 密钥寄存器，且在随后的 64 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。

地址：0x4800_7030

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 窗口看门狗 1/ADC 窗口看门狗 0/ TIM15_TRGO / TIM2_TRGO / 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_BKIN2 输入触发源使能控制，默认值 000000
	TIM1_BKIN2 输入触发源为 ADC 窗口看门狗 1/ADC 窗口看门狗 0/ TIM16_TRIG_OC1 /TIM17_TRIG_OC1/ 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
	TIM1_BKIN2 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_BKIN 输入触发源使能控制，默认值 0000000
	TIM1_BKIN 输入触发源为外部 GPIO 输入/ADC 窗口看门狗 1/ADC 窗口看门狗 0/ TIM15_TRGO / TIM2_TRGO / 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
	TIM1_BKIN 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_BKIN/TIM1_BKIN2/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
注: TIMx_TRIG_OCI 和 TIMx_TRGO 不同，详见 TIM 章节说明。	

14. 通用定时器 TIM2

LCM32F06X 包括 1 个 PWM 专用定时器、4 个通用定时器、1 个看门狗定时器和 1 个系统时基定时器。具体差异请参考第 13 章 PWM 专用定时器 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 TIM2 主要特性

通用定时器 TIM2 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，TIM2 没有重复计数器。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）产生相应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）进行同步。

TIM2 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（边沿对齐或中央对齐模式）
- 单脉冲模式输出

TIM2 没有互补输出，如果配置为 PWM 生成器，TIM2 具有全调制能力（0 到 100%）。在调试模式下，计数器可以被冻结。TIM2 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。

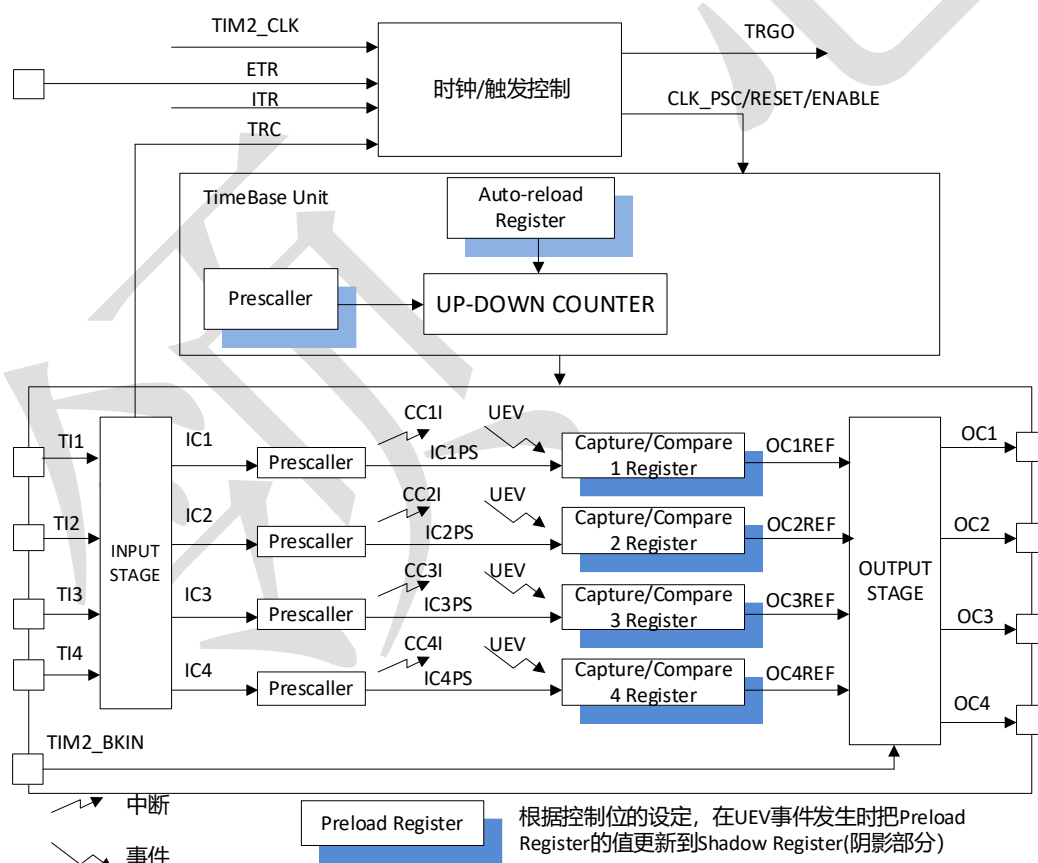


图 14-1 通用定时器 TIM2 框图

14.2 管脚配置

表 14-1 TIM2 管脚配置

管脚名称	管脚类型	管脚描述	复用 I/O 口	备注
TIM2_CH1	I/O	TIM2 捕获/输出通道 1	PA0/PA6/PB4	TIM2_CH1 没有互补通道
TIM2_CH2	I/O	TIM2 捕获/输出通道 2	PA1/PA7/PB5	TIM2_CH2 没有互补通道
TIM2_CH3	I/O	TIM2 捕获/输出通道 3	PB2/PA3/PB0/PB6	TIM2_CH3 没有互补通道
TIM2_CH4	I/O	TIM2 捕获/输出通道 4	PB1	TIM2_CH4 没有互补通道
TIM2_GPIO_BKIN	I	TIM2 的外部 I/O 刹车	PB2	-
TIM2_ETR	I	TIM2 的外部 I/O 触发	PA5	-

14.3 时基单元

时基单元包含：

- 16位向上/向下计数寄存器（TIM2_CNT）
- 16位预分频寄存器（TIM2_PSC）
- 16位自动重装载寄存器（TIM2_ARR）

16 位计数寄存器、预分频寄存器和自动重装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 TIM2_CR1.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。

写计数寄存器的操作没有缓存，可以在任何时候写 TIM2_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。

14.3.1 自动重装载寄存器

自动重装载寄存器是预先装载的。写或读自动重装载寄存器将访问预装载寄存器。根据 TIM2_CR1.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 TIM2_CR1.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）。

写自动重装载寄存器的两种模式：

- 自动预装载已使能（TIM2_CR1.ARPE 位置位）。在此模式下，写入自动重装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（TIM2_CR1.ARPE 位清除）。在此模式下，写入自动重装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 TIM2_EGR.UG 位
- 时钟/触发控制器产生硬件复位

在预装载使能时（TIM2_ARPE=1），如果发生了更新事件，预装载寄存器中的数值（TIM2_ARR）将写入影子寄存器中。置位 TIM2_CR1.UDIS 位将禁止产生更新事件（UEV）。

14.3.2 预分频器

TIM2 的预分频器基于一个由 16 位寄存器 (TIM2_PSC) 控制的 16 位计数器。由于这个控制寄存器带有影子寄存器, 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 TIM2_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。

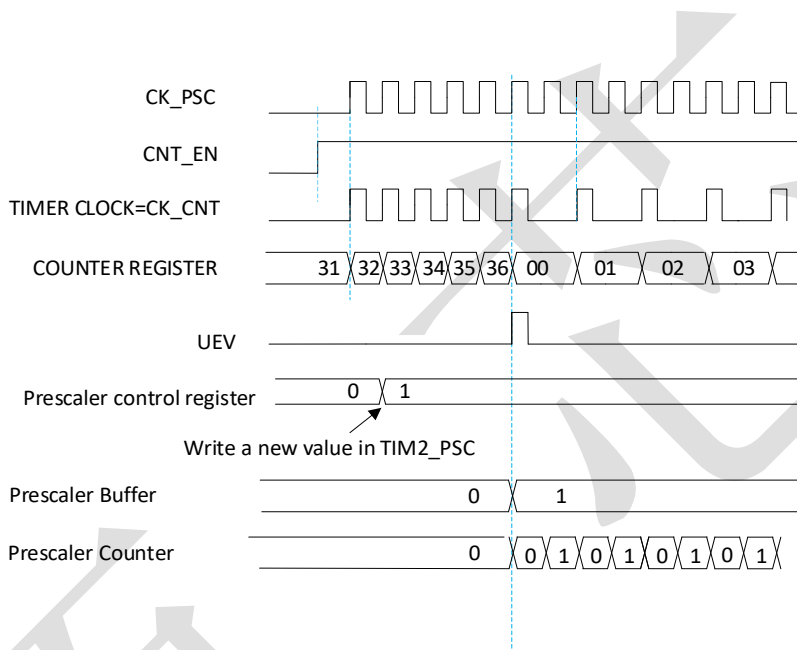


图 14-2 预分频系数从 1 变 2 的计数器时序

14.4 计数器模式

14.4.1 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 (TIM2_ARR 寄存器的值), 然后重新从 0 开始计数并产生一个计数器上溢出事件, 同时, 如果 TIM2_CR1.UDIS 位是 0, 将会产生一个更新事件 (UEV)。置位 TIM2_EGR.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 TIM2_CR1.UDIS 位, 可以禁止产生更新事件, 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, 如果配置 TIM2_CR1.URS=0 (选择更新请求源), 置位 TIM2_EGR.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, 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 (TIM2_SR.UIF 位): 自动重装载影子寄存器会更新为新的预装载寄存器的值 (TIM2_ARR)。预分频寄存器的影子寄存器

会更新为新的预装载寄存器的值（TIM2_PSC）。

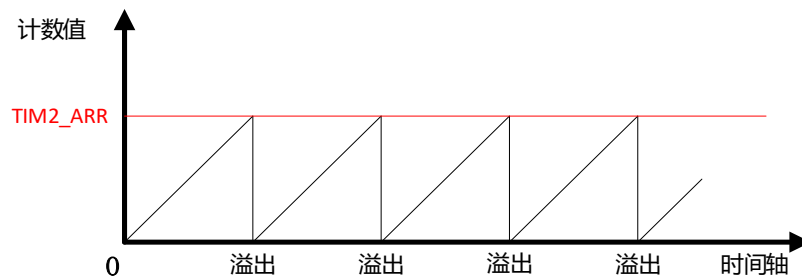


图 14-3 向上计数溢出

下图给出例子，当 TIM2_ARR=0x36 时计数器在二分频时钟频率下的动作。

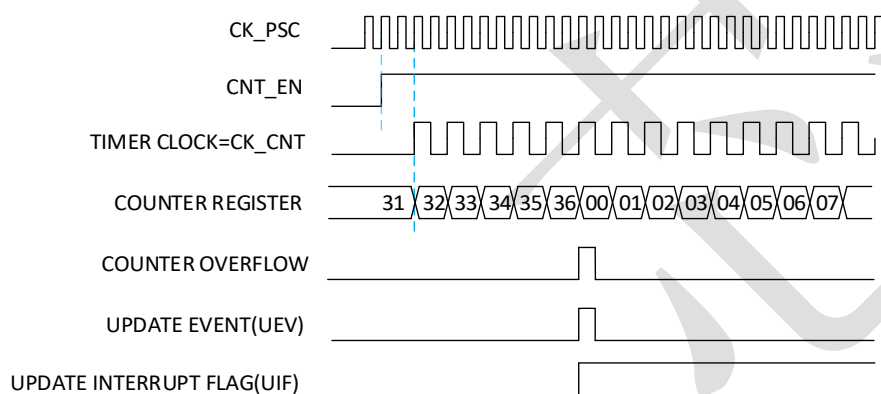


图 14-4 计数器在二分频时钟频率下的时序图

14.4.2 向下计数模式

在向下计数模式中，计数器从自动重装载寄存器的值（TIM2_ARR）开始向下计数到 0，然后再从 TIM2_ARR 重新开始计数，并产生一个计数器向下溢出事件。如果 TIM2_CR1.UDIS 位是 0，将会产生一个更新事件（UEV）。

置位 TIM2_CR1.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，如果配置 TIM2_CR1.URS=0 位（选择更新请求源），置位 TIM2_EGR.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。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（TIM2_SR.UIF 位），自动重装载影子寄存器会更新为新的预装载寄存器的值（TIM2_ARR）。预分频寄存器的影子寄存器会更新为新的预装载寄存器的值（TIM2_PSC）。

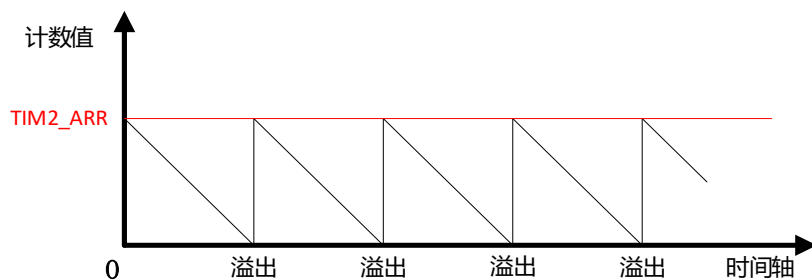


图 14-5 向下计数溢出

14.4.3 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（TIM2_ARR-1），产生一个计数器上溢出事件，然后向下计数到 0 并且产生一个计数器下溢出事件；然后再从 0 开始重新计数。如果 TIM2_CR1.UDIS 位是 0，计数器向上向下溢出都会将会产生一个更新事件（UEV）。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。

在此模式下，不能写入 TIM2_CR1.DIR 方向位。它由硬件更新并指示当前的计数方向。

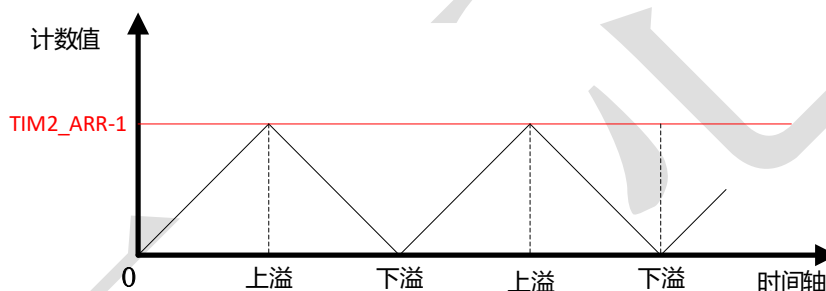


图 14-6 中央对齐溢出

设置 TIM2_CR1.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，如果配置 TIM2_CR1.URS=0（选择更新请求源），设置 TIM2_EGR.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，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（TIM2_SR.UIF 位），自动重装载影子寄存器会更新为新的预装载寄存器的值（TIM2_ARR）。预分频寄存器的影子寄存器的会更新为新的预装载寄存器的值（TIM2_PSC）。

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 TIM2_CR1.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（TIM2_CNT > TIM2_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 TIM2_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 TIM2_EGR.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。

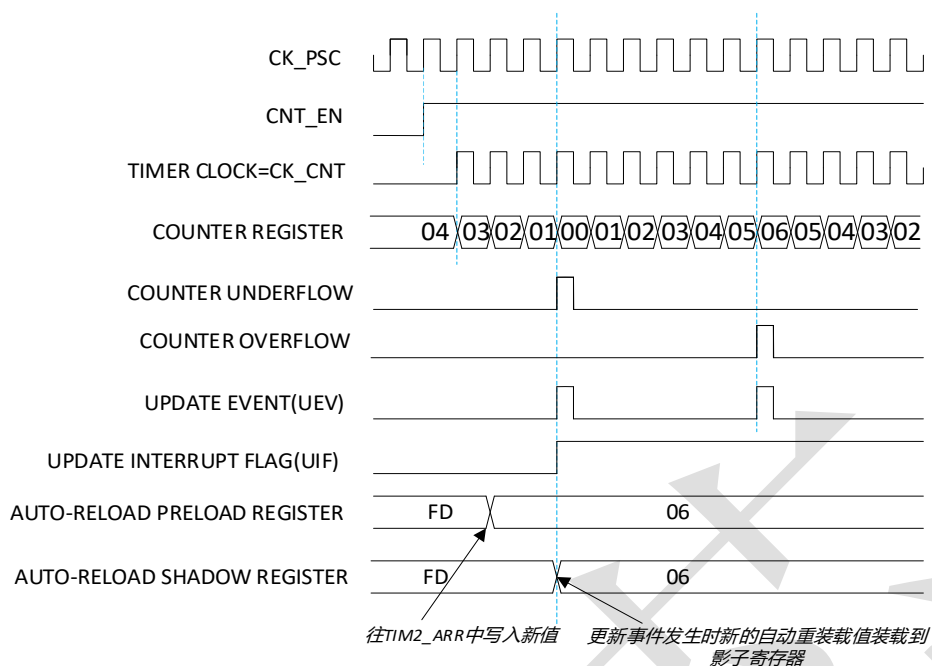


图 14-7 计数器时序图，内部时钟分频为 1，TIM2_ARR=0x6

14.5 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），如图 14-8 所示。

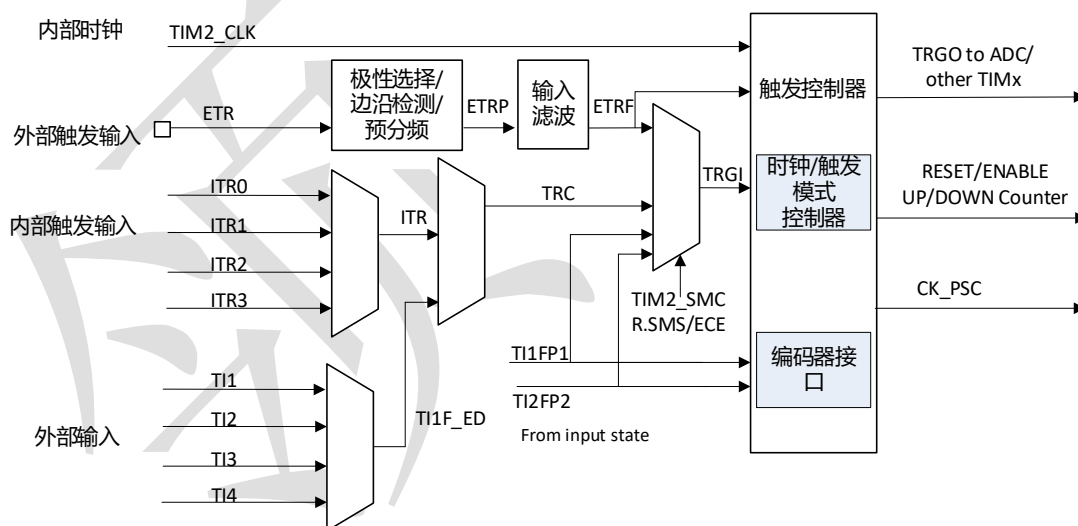


图 14-8 时钟/触发控制器框图

时基单元的预分频时钟（CK_PSC）可以有以下来源：

- 内部时钟（TIM2_CLK）：来自于 SYS_CLK 或 PCLK0
- 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

14.5.1 内部时钟源 (TIM2_CLK)

如果 TIM2_SMCR.SMS=3'b000, TIM2_SMCR.ECE=0, 则 TIM2_CR1.CEN、TIM2_CR1.DIR 和 TIM2_EGR.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。

14.5.2 外部时钟模式 1

当 TIM2_SMCR.SMS=3'b111 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。

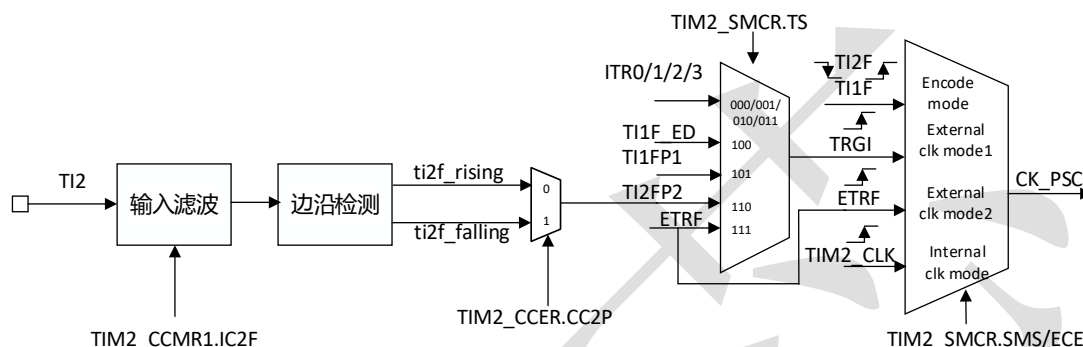


图 14-9 TI2 外部时钟框图

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 TIM2_CCMR1.CC2S=2'b01, 使用通道 2 检测 TI2 输入的上升沿
2. 配置 TIM2_CCMR1.IC2F 位, 选择输入滤波器带宽 (如果不需要滤波器, 保持 IC2F=4'b0000)
3. 配置 TIM2_CCER.CC2P=0, 选定上升沿极性
4. 配置 TIM2_SMCR.SMS=3'b111, 配置计数器使用外部时钟模式 1
5. 配置 TIM2_SMCR.TS=3'b110, 选定 TI2 作为输入源
6. 设置 TIM2_CR1.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 (TIM2_SR.TIF 位) 被置 1, 如果使能了中断 (在 TIM2_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)。

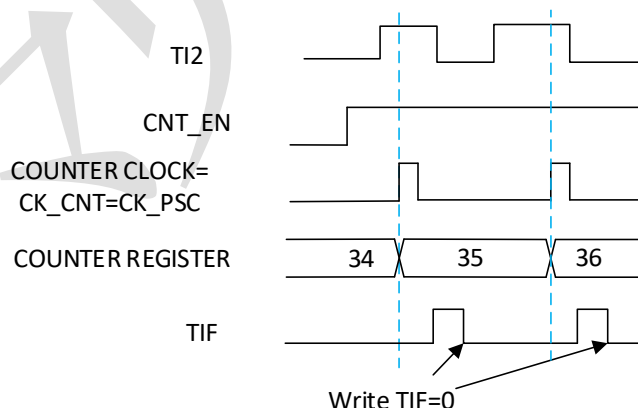


图 14-10 外部时钟模式 1 下的控制电路

14.5.3 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 TIM2_SMCR.ECE 位写 1，即可选定此模式。

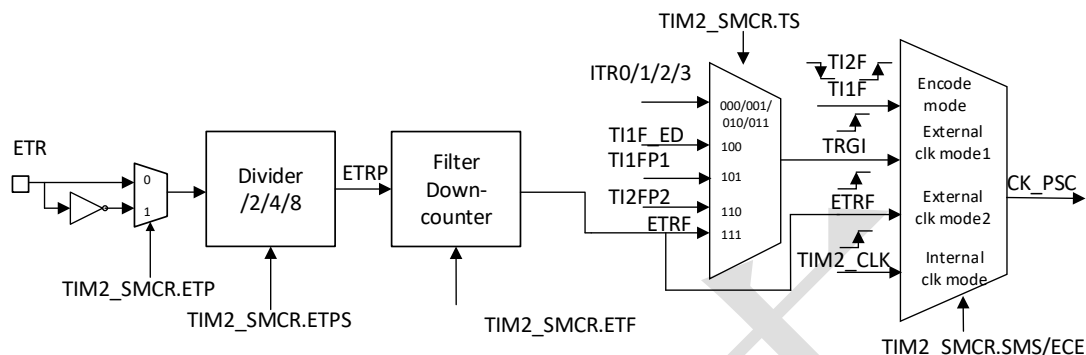


图 14-11 外部触发时钟框图

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 TIM2_SMCR.ETF=4'b0000
2. 设置预分频器，配置 TIM2_SMCR.ETPS=2'b01
3. 选择 ETR 的上升沿检测，配置 TIM2_SMCR.ETP=0
4. 开启外部时钟模式 2，配置 TIM2_SMCR.ECE=1
5. 启动计数器，写 TIM2_CR1.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。

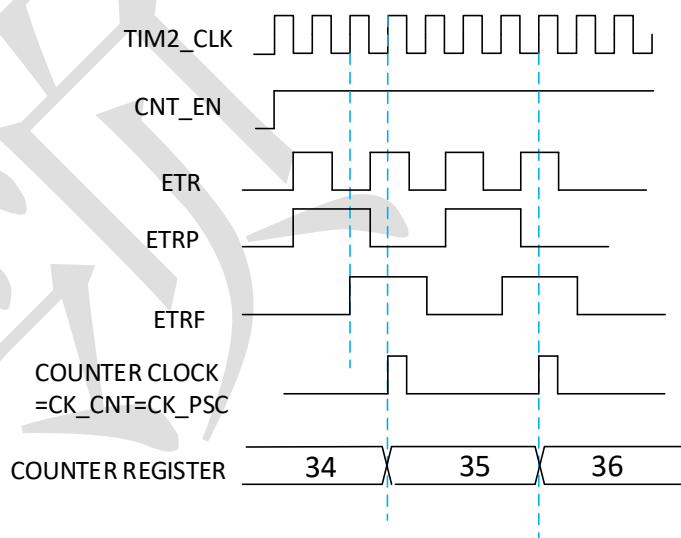


图 14-12 外部时钟模式 2 下的控制电路

14.6 捕获/比较通道

定时器的 I/O 引脚（TIM2_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 TIM2_CCMRi.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）。

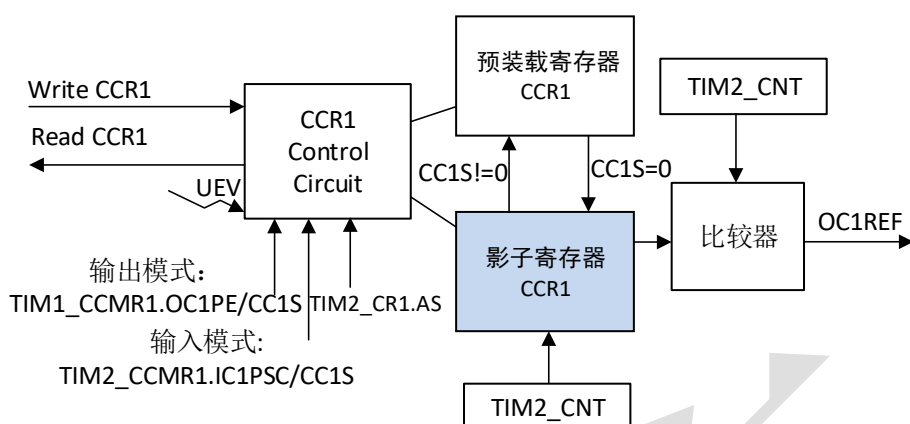


图 14-13 捕获/比较通道 1 的主电路

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，比较模式下，读过程可根据 TIM2_CR1.AS 位选择读取影子寄存器或预装载寄存器，写过程中如果 TIM2_CCMi.OCiPE=0，则可直接写入到影子寄存器。在比较模式下，TIM2_CCMi.OCiPE=1 且在 UEV 事件到来时预装载寄存器的内容被复制到影子寄存器中，然后使用影子寄存器的值和计数器进行比较。

当通道被配置成输出模式时（TIM2_CCMRi.CCiS=0），可以随时访问 TIM2_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 TIM2_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 TIM2_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）。

14.6.1 输入模块

输入部分对相应的 Tii 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缘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）。

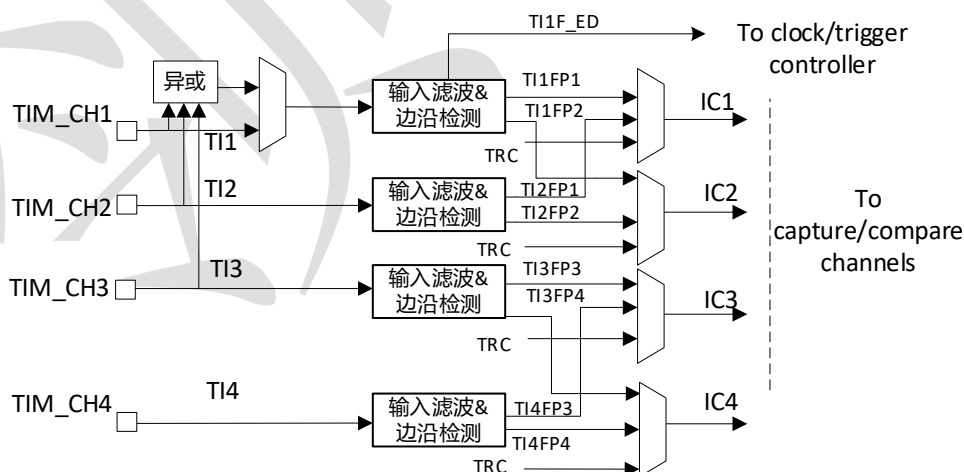


图 14-14 输入模块框图

14.6.2 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

(TIM2_CCRi) 中，当读取 TIM2_CCRi 时，会将捕获通道的最近一次边沿锁存到 TIM2_SR.CAPiP 标志。当发生捕获事件时，相应的 TIM2_SR.CC1IF 标志被置 1。

如果 TIM2_DIER.CC1IE 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 CC1IF 标志已经为高，那么重复捕获标志 TIM2_SR.CC1OF 被置 1，写 CC1IF=0 可清除 CC1OF。写 CC1IF=0 或读取存储在 TIM2_CCRi 寄存器中的捕获数据都可清除 CC1IF。

TIM2 通道 1 的输入如下图所示：

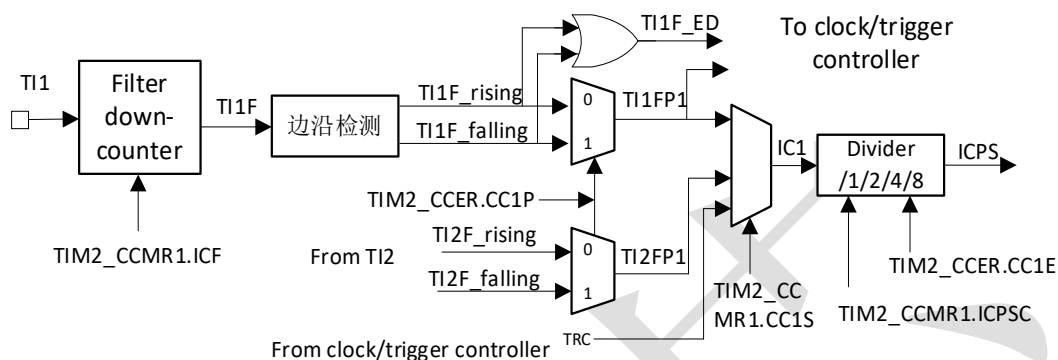


图 14-15 TIM2 通道 1 的输入

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 TIM2_CCR1 寄存器中，步骤如下：

1. 选择有效输入端：例如 TIM2_CCR1 连接到 TI1 输入，配置 TIM2_CCMR1.CC1S=2'b01，此时通道被配置为输入，并且 TIM2_CCR1 寄存器变为只读。
2. 根据输入信号 Tii 的特点，可通过配置 TIM2_CCMRi.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 TIM2_CCMR1.IC1F=4'b0011，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，配置 TIM2_CCER.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（配置 TIM2_CCMR1.IC1PSC=2'b00）。
5. 设置 TIM2_CCER.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 TIM2_DIER.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 TIM2_CCR1 寄存器。
- CC1IF 标志被设置（中断标志）。当发生至少 2 个连续的捕获时，而 CC1IF 未曾被清除时，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。

14.6.3 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

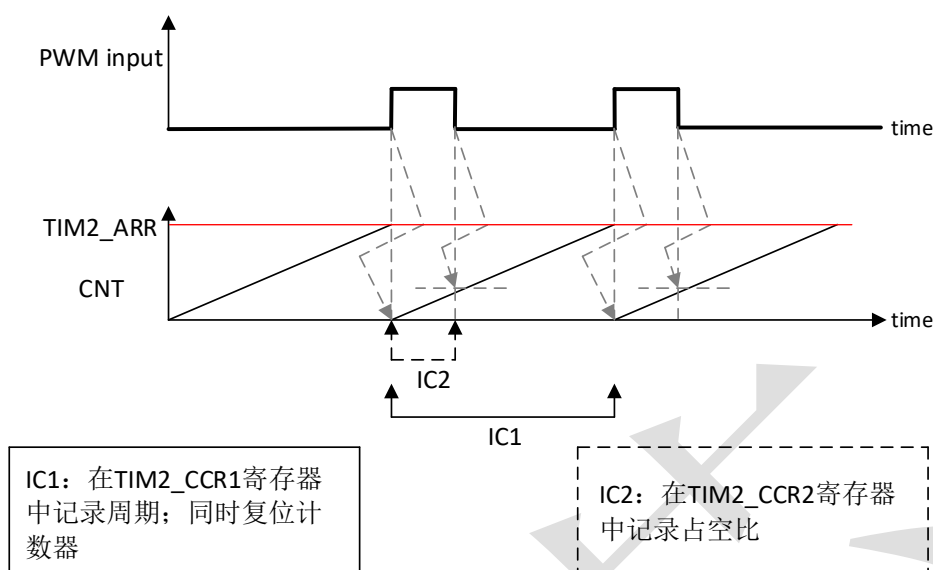


图 14-16 PWM 输入信号测量

例如，可以用以下方式测量 TI1 上输入的 PWM 信号的周期（锁存于 TIM2_CCR1 寄存器）和占空比（锁存于 TIM2_CCR2 寄存器）。（具体取决于 TIM2_CLK 的频率 f_{TIM2_CLK} 和预分频器的值）

1. 选择 TIM2_CCR1 的有效输入：配置 TIM2_CCMR1.CC1S=2'b01（选中 TI1）。
2. 选择 TI1FP1 的有效极性（用来捕获数据到 TIM2_CCR1 中和清除计数器）：配置 TIM2_CCER.CC1P=0（上升沿有效）。
3. 选择 TIM2_CCR2 的有效输入：配置 TIM2_CCMR1.CC2S=2'b10（选中 TI1FP2）。
4. 选择 TI1FP2 的有效极性（捕获数据到 TIM2_CCR2）：配置 TIM2_CCER.CC2P=1（下降沿有效）。
5. 选择有效的触发输入信号：配置 TIM2_SMCR.TS=3'b101（选择 TI1FP1）。
6. 配置触发模式控制器为复位触发模式：配置 TIM2_SMCR.SMS=3'b100。
7. 使能捕获：配置 TIM2_CCER.CC1E=1，TIM2_CCER.CC2E=1。

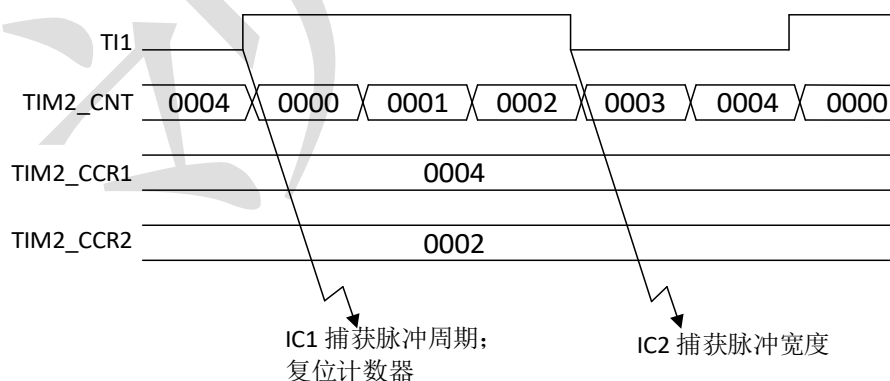


图 14-17 PWM 输入信号测量实例

14.6.4 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）。TIM2 输出模块如图 14-18

所示，TIM2 没有互补输出，经极性选择、刹车控制和滤波后产生 OC_i 信号。另外，可通过 TIM2_BDTR.BKTRIG 选择连接到 ADC 的触发信号 TIM2_trig_oc_i 是否经过刹车控制。

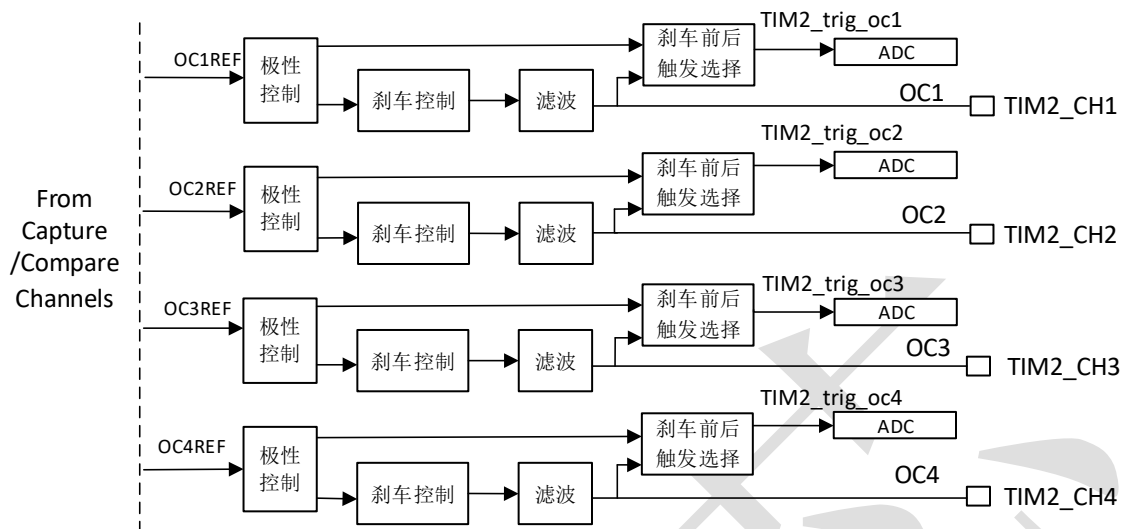


图 14-18 输出模块框图

14.6.5 强制输出模式

在输出模式（TIM2_CCMRi.CCiS=2'b00）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。

配置 TIM2_CCMRi.OCiM=3'b101，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 OC_i 的输出是高还是低则取决于 CCiP 极性标志位。例如 CCiP=0（OC_i 高电平有效），则 OC_i 被强置为高电平。

配置 TIM2_CCMRi.OCiM=3'b100，可强置 OCiREF 信号为低。

该模式下，在 TIM2_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。

14.6.6 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等时，有如下操作：

- 根据不同的输出比较模式（由 TIM2_CCMRi.OCiM 设定），相应的 OCiREF 输出信号：
 - 保持不变（OCiM=3'b000）
 - 设置为有效电平（OCiM=3'b001）
 - 设置为无效电平（OCiM=3'b010）
 - 翻转（OCiM=3'b011）
- 置位中断状态寄存器中的标志位（TIM2_SR.CCiIF 位）
- 若设置了相应的中断使能位（TIM2_DIER.CCiIE 位），则产生一个中断

TIM2_CCER.CCiP 位用于选择有效和无效的电平极性。TIM2_CCMRi.OCiPE 位用于选择 TIM2_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 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 TIM2_ARR 和 TIM2_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=3'b011
 - 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 TIM2_CR1.CEN 位来启动计数器

在未使用预装载寄存器（OCiPE=0，否则 TIM2_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，TIM2_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。

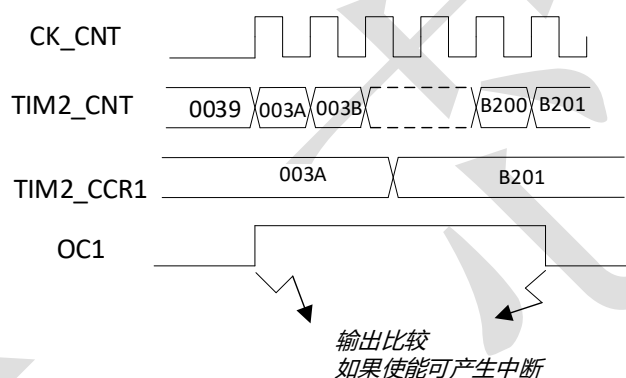


图 14-19 输出比较模式，翻转 OC1

14.6.7 PWM 模式

脉冲宽度调制（PWM）模式可以产生一个由 TIM2_ARR 寄存器确定频率，由 TIM2_CCRi 寄存器确定占空比的信号。

在 TIM2_CCMRi.OCiM 位写入 3'b110（PWM 模式 1）或 3'b111（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 TIM2_CCMR.OCiPE 位使能相应的预装载寄存器，若设置 TIM2_CR1.ARPE=1，则使能 TIM2_ARR 寄存器的预装载功能，即软件向 TIM1_ARR 写值的时候，只有当发生更新事件时，此预装载值才传输至影子寄存器中，因此在计数器开始计数之前，必须通过设置 TIM2_EGR.UG 位来初始化所有的寄存器。

OCi 的极性可以通过软件在 TIM2_CCER.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 TIM2_CCER.CCIe 位控制，详见 TIM2_CCER 寄存器的描述。

在 PWM 模式（模式 1 或模式 2）下，TIM2_CNT 和 TIM2_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 $TIM2_CCRi \leq TIM2_CNT$ 或者 $TIM2_CNT \leq TIM2_CCRi$ 。

根据 TIM2_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。

14.6.7.1 PWM 边沿对齐模式

向上计数配置

当 TIM2_CR1.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 $TIM2_CNT <$

TIM2_CCRi 时，PWM 参考信号 OCiREF 为高，否则为低。如果 TIM2_CCRi 中的比较值大于自动重装载值（TIM2_ARR），则 OCiREF 保持为 1。如果比较值为 0，则 OCiREF 保持为 0。下图为 TIM2_ARR=8 时边沿对齐的 PWM 波形实例。

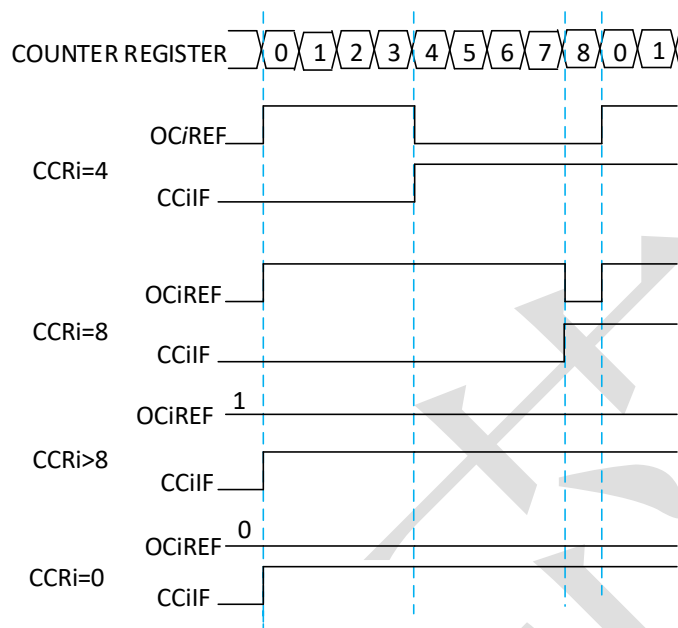


图 14-20 TIM2_ARR=8 时边沿对齐的 PWM 波形实例

向下计数的配置

当 TIM2_CR1.DIR 位为高时执行向下计数。在 PWM 模式 1 时，当 TIM2_CNT>TIM2_CCRi 时参考信号 OCiREF 为低，否则为高。如果 TIM2_CCRi 中的比较值大于 TIM2_ARR 中的自动重装载值，则 OCiREF 保持为 1。该模式下不能产生 0% 的 PWM 波形。

14.6.7.2 PWM 中央对齐模式

当 TIM2_CR1.CMS 位!=00 时为中央对齐模式。

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。

TIM2_CR1.DIR 由硬件更新，不要用软件修改它。

下图为 PWM 中央对齐模式波形示例：

- TIM2_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（中央对齐模式 1，CMS=2'b01）
 - 只有在计数器向上计数时（中央对齐模式 2，CMS=2'b10）
 - 在计数器向上和向下计数时（中央对齐模式 3，CMS=2'b11）

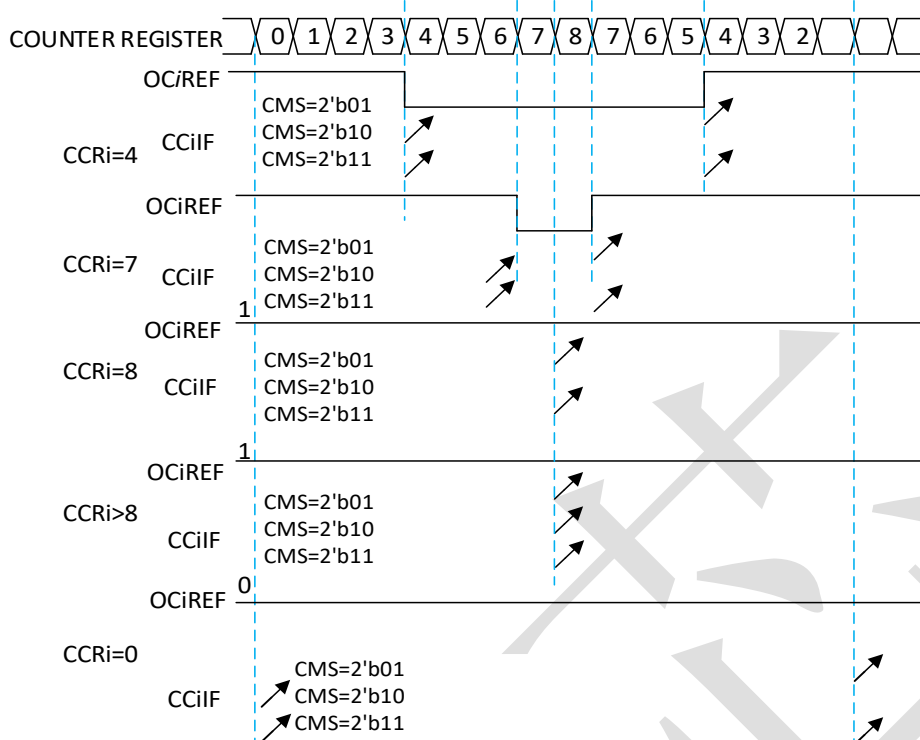


图 14-21 中央对齐模式 PWM (TIM2_ARR=8)

14.6.8 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 TIM2_CR1.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 产生时停止。

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来定义（需要考虑时钟频率和预分频器）：

1. TI2FP2 对应到 TI2，配置 TIM2_CCMR1.CC2S=2'b01
2. TI2FP2 上升沿检测，配置 TIM2_CCER.CC2P=0
3. 设置 TI2FP2 为从模式下的触发信号 (TRGI)，配置 TIM2_SMCR.TS=3'b110
4. 设置 TI2FP2 为计数器的启动信号，配置 TIM2_SMCR.SMS=3'b110（触发模式）
5. t_{DELAY} 通过 TIM2_CCR1 寄存器配置得到
6. t_{PULSE} 通过 TIM2_ARR - TIM2_CCR1 得到
7. 如果希望高电平脉冲是在计数器到达 TIM2_CCR1 时信号由 0 变为 1，到达 TIM2_ARR 时信号由 1

变为 0，则设置 PWM 模式 2，配置 TIM2_CCMR1.OC1M=3'b111

8. 可选启用预装载功能，配置 TIM2_CCMR1.OC1PE=1 和 TIM2_CR1.ARPE=1。如果使用预装载功能，在 TIM2_ARR 和 TIM2_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，DIR=0，CMS=2'b00。

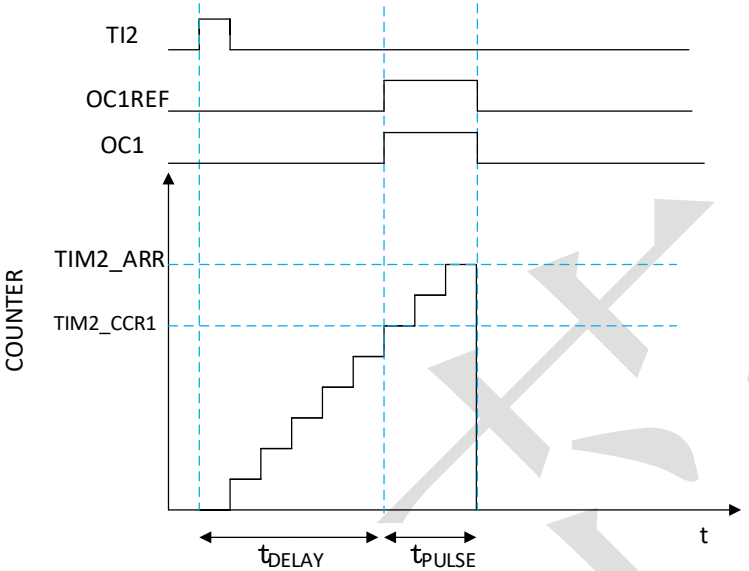


图 14-22 单脉冲模式

14.6.9 特殊情况：OCi 快速使能

在单脉冲模式下，对 Ti 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 TIM2_CCMR1.OCiFE 位；此时强制 OCiREF（和 OCi）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。

14.6.10 刹车功能

刹车功能常用于马达控制中。当刹车功能开启后，刹车源输入有效时，输出使能信号和输出电平将会依据刹车控制位的配置（TIM2_BDTR 寄存器中的 MOE、OSS1 和 OSSR 位，TIM1_CR2 寄存器中的 OISi 位）立即生效。TIM2 有 1 路刹车通道 TIM2_BKIN。互联配置寄存器 9（SysCtrl_EDU_CFG9）中给出了具体的刹车源，此外系统中会产生的 2 个特殊的刹车源，分别为 CPU 内核锁死和电压过低触发 LVD 产生的低压检测标志，这两个刹车源不受使能和极性控制，具体描述请见芯片控制寄存器 ChipCtrl_CTRL.LOCKUPEN 和 ChipCtrl_CTRL.LVD_LOCK 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，TIM2_BDTR.MOE 位为低。配置 TIM2_BDTR.BKE 位可以使能刹车功能，其刹车输入信号的极性可以通过配置其中的 TIM2_BDTR.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 TIM2_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。



当需要使用 TIMER 和 ADC/ACMP 联动时，可设置 TIM2_BDTR.BKTRIG 位选择连接 ADC 的触发信号是否经过刹车控制。若设置 TIM2_BDTR.BKTRIG=1'b1，则触发 ADC 的 TIM2_trig_oci 为刹车处理前的信号，此时若刹车输入信号有效，TIM2 会进入刹车状态，但 ADC 仍可被触发。反之，设置 TIM2_BDTR.BKTRIG=1'b0（默认），则触发 ADC 的 TIM2_trig_oci 为刹车处理后的信号，此时若刹车输入信号有效，TIM2 会进入刹车状态，ADC 不会被触发。

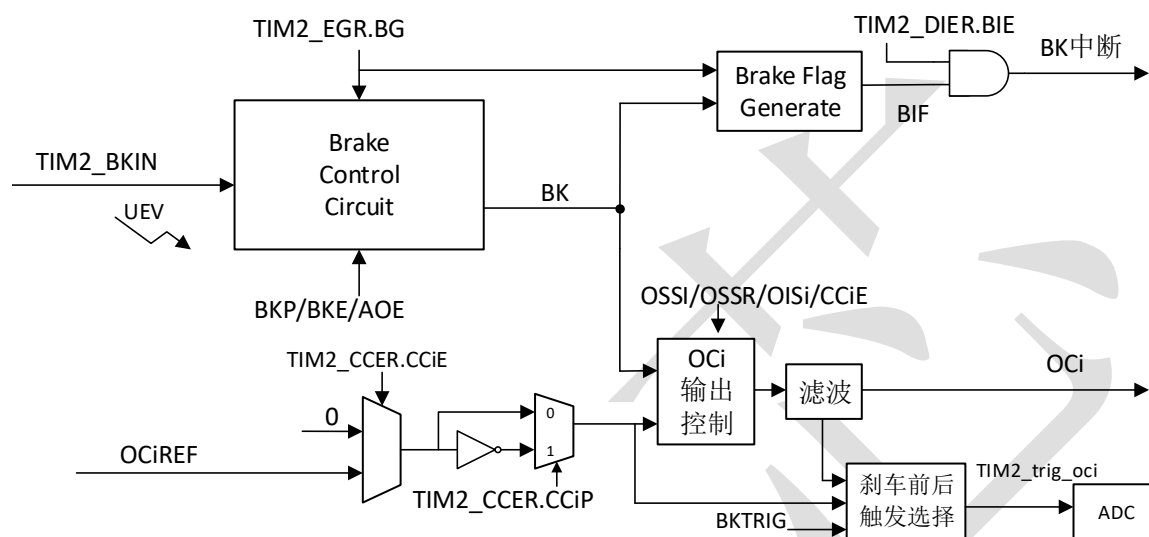


图 14-23 TIM2 刹车通道框图

例如，当刹车通道中有刹车源触发时：

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，如果 OSSI=0，定时器输出不使能；如果 OSSI=1，则定时器输出使能依 TIM1_CCER.CCIE 决定。但不管 OSSI=0/1，输出都可以由 OISi 位驱动，具体情况描述可参考表 13-5。
3. 刹车状态标志 TIM2_SR.BIF 被置起。且如果 TIM2_DIER.BIE 位使能，则会产生中断。
4. 如果 TIM2_BDTR.AOE 位被置位，MOE 位会自动在下次更新事件 UEV 到来时被置位。否则 MOE 位会一直为低直到被手动写 1。

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。

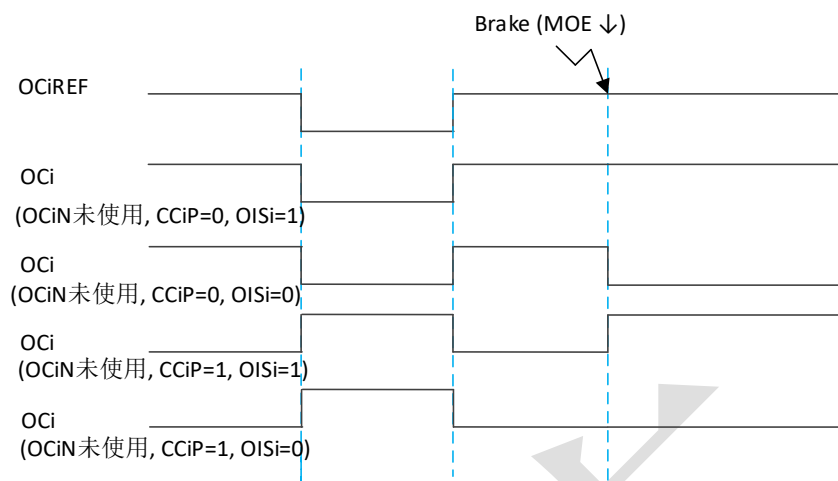


图 14-24 刹车对应的输出波形

有两种方式来产生刹车：

- 通过刹车输入并配合 TIM2_BDTR 寄存器中的 BKE 位和 BKP 位
- 通过软件配置 TIM2_EGR.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（OCi 极性和状态、OCiM 配置、刹车使能和极性等）。用户可以在 TIM2_BDTR.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。

14.6.11 通过外部事件清除 OCiREF

设定相应 TIM2_CCMRi.OCiCE 置为 1，则给定通道的 OCiREF 信号可以通过外部输入信号或内部其他信号来清除，并保持低电平，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 TIM2_SMCR.OCiCS 位，可以在内部其他信号 OCREF_CLR 输入和外部输入信号 ETRF（滤波器后的 ETR）之间选择 OCiREF 清除信号的来源。如果选择了 ETRF，ETR 需要如下配置：

- 外部触发预分频器要保持关闭：TIM2_SMCR.ETPS=2'b00
- 外部时钟源模式 2 关闭：TIM2_SMCR.ECE=0
- 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，可通过互联配置寄存器 9（SysCtrl_EDU_CFG9）配置。OCREF_CLR 的极性可以通过 TIM2_SMCR.OCCP 配置。

14.6.12 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，则配置 TIM2_SMCR.SMS=3'b001；如果只在 TI1 边沿计数，则配置 SMS=3'b010；如果计数器同时在 TI1 和 TI2 边沿计数，则配置 SMS=3'b011。通过设置 TIM2_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 14-2，假定计数器已经启动（TIM2_CR1.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 TIM2_ARR 寄存器的自动重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 TIM2_ARR 寄存器；同样，捕获器、预分频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 14-2 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 14-2 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 14-25 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；抖动可能会在当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：

- TIM2_CCMR1.CC1S=2'b01（TI1FP1 映射到 TI1）
- TIM2_CCMR1.CC2S=2'b01（TI2FP2 映射到 TI2）
- TIM2_CCER.CC1P=0，TIM2_CCMR1.IC1F=4'b0000（TI1FP1 不反相，TI1FP1=TI1）
- TIM2_CCER.CC2P=0，TIM2_CCMR1.IC2F=4'b0000（TI2FP2 不反相，TI2FP2=TI2）
- TIM2_SMCR.SMS=3'b011（TI1 和 TI2 输入均在上升沿和下降沿有效）
- TIM2_CR1.CEN=1（计数器使能）

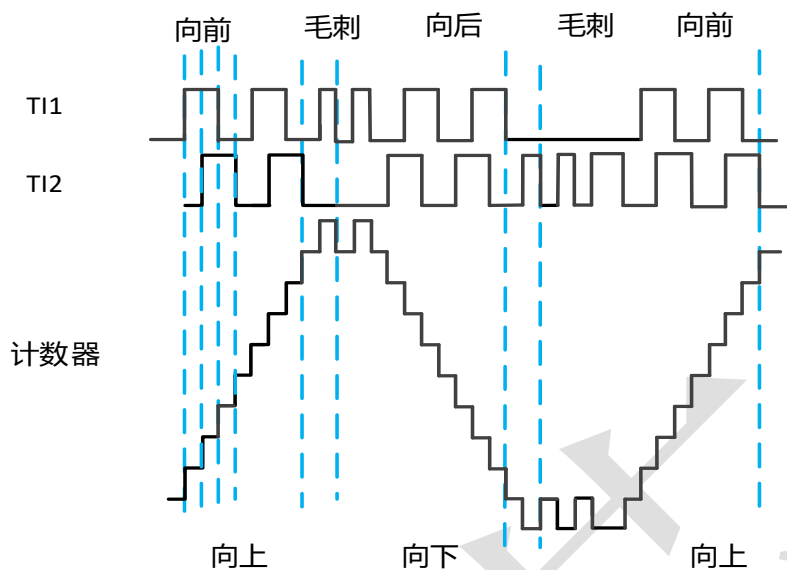


图 14-25 编码器接口模式示例

当其他配置保持一致，TI1FP1 极性反相时的示例：

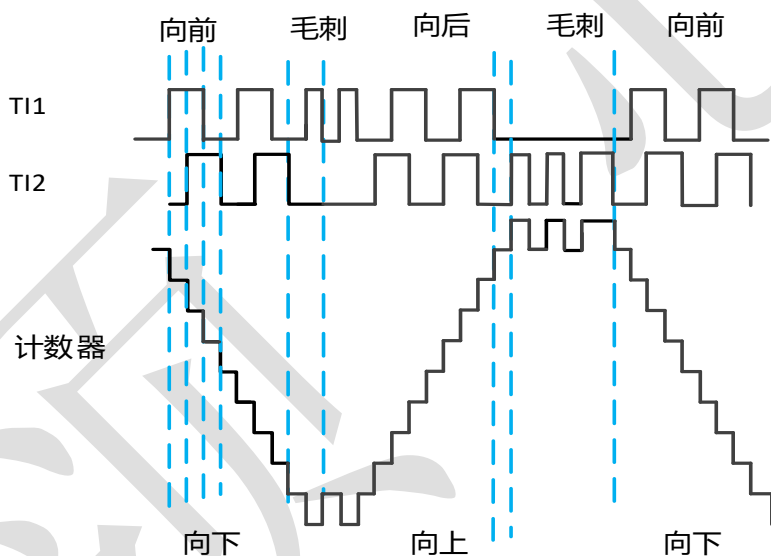


图 14-26 TI1FP1 反相的编码器接口模式示例

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。

14.6.13 定时器输入异或功能

通过设置 TIM2_CR2.TI1S 位允许 TI1 输入通道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 TIM2_CH1、TIM2_CH2 和 TIM2_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。

14.7 TIM2 定时器与外部触发的同步

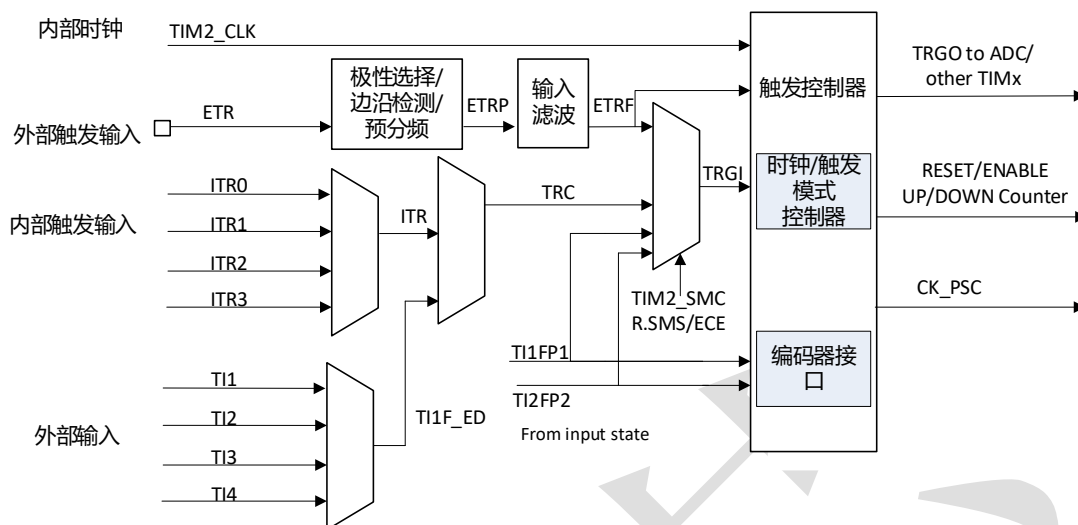


图 14-27 TIM2 触发输入功能框图

TIM2 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。TIM2 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。

14.7.1 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，TIM2 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 TIM2_CCMR1.IC2F=4'b0000）
- 触发操作中不使用捕获预分频器，不需要配置；配置 TIM2_CCMR1.CC2S=2'b01，用于选择输入捕获源
- 配置 TIM2_CCER.CC2P=0，选择上升沿做为触发条件
- 配置 TIM2_SMCR.SMS=3'b110，选择定时器为触发模式
- 配置 TIM2_SMCR.TS=3'b110，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 TIM2_SR.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 输入端的同步电路。

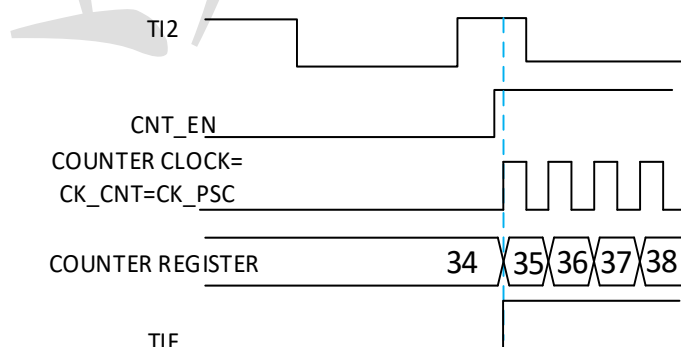


图 14-28 标准触发模式下的时序图

14.7.2 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 TIM2_CR1.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 TIM2_CCMR1.IC1F=4'b0000）
- 触发操作中不使用捕获预分频器，所以不需要配置；配置 TIM2_CCMR1.CC1S=2'b01，用于选择输入捕获源
- 配置 TIM2_CCER.CC1P=0 来选择极性（只检测上升沿）
- 配置 TIM2_SMCR.SMS=3'b100，选择定时器为复位模式
- 配置 TIM2_SMCR.TS=3'b101，选择 TI1 作为输入源
- 配置 TIM2_CR1.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，TIM2_SR.TIF 位被置起，如果使能了中断（TIM2_DIER.TIE 位置 1），则产生一个中断请求

图 14-29 显示当自动重装载寄存器 TIM2_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 输入端的同步电路。

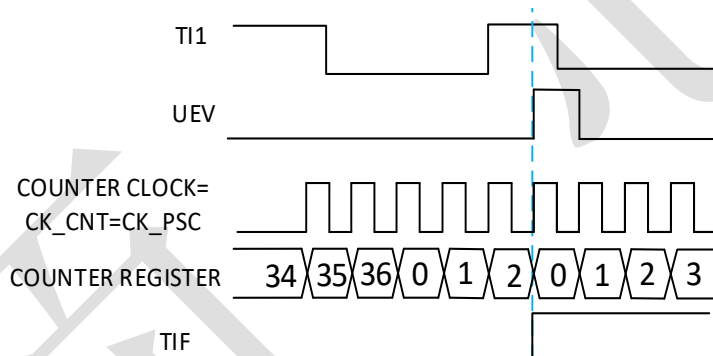


图 14-29 复位模式下的时序图

14.7.3 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 TIM2_CCMR1.IC1F=4'b0000）
- 触发操作中不使用捕获预分频器，所以不需要配置；配置 TIM2_CCMR1.CC1S=2'b01，用于选择输入捕获源
- 配置 TIM2_CCER.CC1P=1 来确定极性（只检测低电平）
- 配置 TIM2_SMCR.SMS=3'b101，选择定时器为门控模式
- 配置 TIM2_SMCR.TS=3'b101，选择 TI1 作为输入源
- 配置 TIM2_CR1.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，TIM2_SR.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 输入端的同步电路。

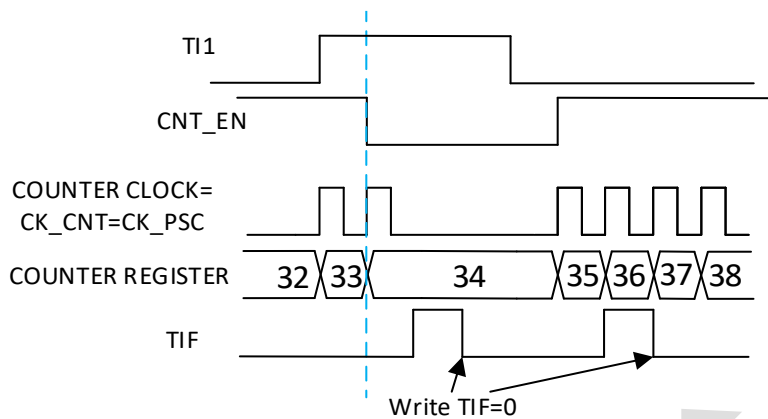


图 14-29 门控模式下的时序图

14.7.4 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 TIM2_CR1.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 TIM2_SMCR.SMS=4'b1000。

14.7.5 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 TIM2_SMCR.TS != 3'b111）。

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 TIM2_SMCR 寄存器配置外部触发输入电路。

- 首先配置 ETR：TIM2_SMCR.ETF=4'b0000 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 TIM2_CCMR1.IC1F=4'b0000）
- 触发操作中不使用捕获预分频器，所以不需要配置；配置 TIM2_CCMR1.CC1S=2'b01，用于选择输入捕获源
- 配置 TIM2_CCER.CC1P=0，选择上升沿做为触发条件
- 配置 TIM2_SMCR.SMS=3'b110，选择定时器为触发模式
- 配置 TIM2_SMCR.TS=3'b101，选择 TI1 作为输入源

当 TI1 上出现一个上升沿时，同时 TIM2_SR.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 输入端的同步电路。

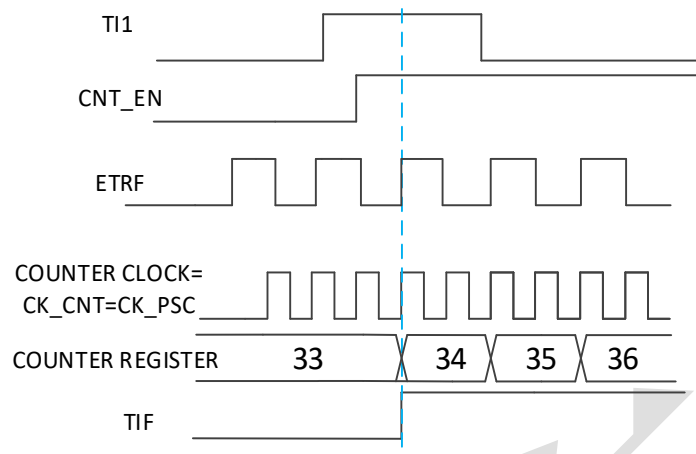


图 14-30 外部时钟模式 2+触发模式下的时序图

14.8 TIM2 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 14-27。

TIM2 的输入触发可以来自芯片内部的其他定时器和比较器 ACMP0/1 输出；TIM2 的输出 TRGO 可触发芯片内部的 ADC、DAC 和其他定时器；TIM2 的特殊输出 TIM2_trig_oci 可触发芯片内部的 ADC 和其他定时器，可以来自比较通道的 OCi 信号或捕获通道的 ICiPS 信号；当 TIM2_trig_oci 来自 OCi 信号时，可选择为刹车控制前或刹车控制后的 OCi 信号，具体详见 TIM2_BDTR.BKTRIG 位说明。

表 14-3 TIM2 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	tim15_trgo	-
ITR2	内部触发 2	tim16_trig_oc1	-
ITR3	内部触发 3	tim17_trig_oc1	-
外部 Timer 输入 TI			
TI1	外部 Timer 输入 1	tim2_gpio_ti1/ adc_awd[0]/ ACMP0_out/ACMP1_out/tim1_trig_oc1/ tim1_trig_oc5/tim15_trig_oc1/tim16_trig_oc1	SysCtrl_EDU_CFG8
TI2	外部 Timer 输入 2	tim2_gpio_ti2/adc_awd[1]/ ACMP0_out/ACMP1_out/ tim1_trig_oc1/ tim1_trig_oc5/tim15_trig_oc1/tim16_trig_oc1	
TI3	外部 Timer 输入 3	tim2_gpio_ti3/adc_awd[0]/ ACMP0_out/ACMP1_out/ tim1_trig_oc1/ tim1_trig_oc5/tim15_trig_oc1/tim16_trig_oc1	
TI4	外部 Timer 输入 4	tim2_gpio_ti4/adc_awd[1]/ ACMP0_out/ACMP1_out/tim1_trig_oc1/ tim1_trig_oc5/tim15_trig_oc1/tim16_trig_oc1	

外部触发输入 ETR			
ETR	外部触发	tim2_gpio_etr/ ACMP0_out/ACMP1_out/ tim1_trig_oc1/ tim1_trig_oc5/tim15_trig_oc1/tim16_trig_oc1	SysCtrl_EDU_CFG8

14.9 TIM2 中断

TIM2 有 7 个中断请求源，共同映射在 TIM2 全局中断：

- 刹车中断
- 更新事件中断
- 触发中断
- 输入捕获 1 中断
- 输入捕获 2 中断
- 输入捕获 3 中断
- 输入捕获 4 中断

为了使用中断特性，对每个被使用的中断通道，设置 TIM2_DIER 寄存器中相应的中断使能位：TIE、CCiE 和 UIE 位。

通过设置 TIM2_EGR 寄存器中的相应位，也可以用软件产生上述各个中断源。

14.10 TIM2 寄存器描述

表 14-4 TIM2 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_F000
CR2	控制寄存器 2	R/W	0x0000_0000	0x4001_F004
SMCR	从模式控制寄存器	R/W	0x0000_0000	0x4001_F008
DIER	DMA 和中断控制寄存器	R/W	0x0000_0000	0x4001_F00C
SR	状态寄存器	R/W	0x0000_0000	0x4001_F010
EGR	事件产生寄存器	R/W	0x0000_0000	0x4001_F014
CCMR1	捕获模式寄存器 1	R/W	0x0000_0000	0x4001_F018
CCMR2	捕获模式寄存器 2	R/W	0x0000_0000	0x4001_F01C
CCER	捕获使能寄存器	R/W	0x0000_0000	0x4001_F020
CNT	计数寄存器	R/W	0x0000_0000	0x4001_F024
PSC	预分频寄存器	R/W	0x0000_0000	0x4001_F028
ARR	自动重装载寄存器	R/W	0x0000_0000	0x4001_F02C
CCR1	捕获寄存器 1	R/W	0x0000_0000	0x4001_F034
CCR2	捕获寄存器 2	R/W	0x0000_0000	0x4001_F038
CCR3	捕获寄存器 3	R/W	0x0000_0000	0x4001_F03C
CCR4	捕获寄存器 4	R/W	0x0000_0000	0x4001_F040
BDTR	刹车死区控制寄存器	R/W	0x0000_0000	0x4001_F044
DCR	DMA 控制寄存器	R/W	0x0000_0000	0x4001_F048
DMAR	DMA 传输寄存器	R/W	0x0000_0000	0x4001_F04C
SysCtrl_EDU_CFG8	TIM2 互联配置寄存器 8	R/W	0x0000_0000	0x4800_7040
SysCtrl_EDU_CFG9	TIM2 互联配置寄存器 9	R/W	0x0000_0000	0x4800_7044



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）。

14.10.1 CR1 控制寄存器 1 (TIM2_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 TIM2_PSC/TIM2_ARR/TIM2_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	23:22	保留，必须保持复位值
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: TIM2_ARR 寄存器可以被直接写入 (默认)
	1: TIM2_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 (TIM2_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 (TIM2_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 (TIM2_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 (TIM2_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: TIM2 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。

14.10.2 CR2 控制寄存器 2 (TIM2_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.															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OIS4	Res.	OIS3	Res.	OIS2	Res.	OIS1	TI1S	MMS[2:0]		CCDS	Res.			
	rw		rw		rw		rw	rw	rw		rw				

Bits	31:15	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	保留，必须保持复位值
Bit	12	OIS3 ：输出空闲状态 3（OC3 输出），参考 OIS1
Bit	11	保留，必须保持复位值
Bit	10	OIS2 ：输出空闲状态 2（OC2 输出），参考 OIS1
Bit	9	保留，必须保持复位值
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
		注： 已经设置了 LOCK（TIM2_BDTR 寄存器）级别 1、2 或 3 后，该位不能被修改。
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 TIM2_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

14.10.3 SMCR 从模式控制寄存器 (TIM2_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 (TIM2_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 (TIM2_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
	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 TIM2 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 TIM15_TRGO
	010: 内部触发 ITR2，连接 TIM15_trig_oc1
	011: 内部触发 ITR3，连接 TIM16_trig_oc1
	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）
	注： 这些位只能在未用到（如 $\text{SMS}=000$ ）时被改变，以避免在改变时产生错误的边沿检测。
Bit 3	OCES: 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 $\text{CEN}=1$ ，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。
	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。

14.10.4 DIER DMA 和中断控制寄存器 (TIM2_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/比较通道 4 的 DMA 请求使能
	0:	捕获/比较通道 4 的 DMA 请求禁止 (默认)
	1:	捕获/比较通道 4 的 DMA 请求使能
Bit	11	CC3DE: 捕获/比较通道 3 的 DMA 请求使能
	0:	捕获/比较通道 3 的 DMA 请求禁止 (默认)
	1:	捕获/比较通道 3 的 DMA 请求使能
Bit	10	CC2DE: 捕获/比较通道 2 的 DMA 请求使能
	0:	捕获/比较通道 2 的 DMA 请求禁止 (默认)
	1:	捕获/比较通道 2 的 DMA 请求使能

Bit 9	CC1DE: 捕获/比较通道 1 的 DMA 请求使能
	0: 捕获/比较通道 1 的 DMA 请求禁止 (默认)
	1: 捕获/比较通道 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/比较通道 4 中断使能
	0: 捕获/比较通道 4 中断禁止 (默认)
	1: 捕获/比较通道 4 中断使能
Bit 3	CC3IE: 捕获/比较通道 3 中断使能
	0: 捕获/比较通道 3 中断禁止 (默认)
	1: 捕获/比较通道 3 中断使能
Bit 2	CC2IE: 捕获/比较通道 2 中断使能
	0: 捕获/比较通道 2 中断禁止 (默认)
	1: 捕获/比较通道 2 中断使能
Bit 1	CC1IE: 捕获/比较通道 1 中断使能
	0: 捕获/比较通道 1 中断禁止 (默认)
	1: 捕获/比较通道 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

14.10.5 SR 状态寄存器 (TIM2_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.		CAP4P	CAP3P	CAP2P	CAP1P										
		r	r	r	r										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.			CC4OF	CC3OF	CC2OF	CC1OF	Res.	BIF	TIF	Res.	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			r	r	r	r		rw	rw		rw	rw	rw	rw	rw

Bits 31:30	保留, 必须保持复位值
Bit 29	CAP4P: 通道 4 捕获边沿标志, 参考 CAP1P 位
Bit 28	CAP3P: 通道 3 捕获边沿标志, 参考 CAP1P 位
Bit 27	CAP2P: 通道 2 捕获边沿标志, 参考 CAP1P 位

Bit	26	CAP1P: 通道 1 捕获边沿标志
		0: 读取 TIM2_CCR1 时, 捕获通道 1 的最近一次边沿为上升沿
		1: 读取 TIM2_CCR1 时, 捕获通道 1 的最近一次边沿为下降沿
Bits	25:13	保留, 必须保持复位值
Bit	12	CC4OF: 捕获/比较通道 4 过捕获/过比较标志, 参考 CC1OF 位
Bit	11	CC3OF: 捕获/比较通道 3 过捕获/过比较标志, 参考 CC1OF 位
Bit	10	CC2OF: 捕获/比较通道 2 过捕获/过比较标志, 参考 CC1OF 位
Bit	9	CC1OF: 捕获/比较通道 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: 无过捕获产生 (默认)
		1: 计数器的值被捕获或匹配到 TIM2_CCR1 寄存器时 CC1IF 已经置 1
Bit	8	保留, 必须保持复位值
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	保留, 必须保持复位值
Bit	4	CC4IF: 捕获/比较通道 4 中断标志, 参考 CC1IF 位
Bit	3	CC3IF: 捕获/比较通道 3 中断标志, 参考 CC1IF 位
Bit	2	CC2IF: 捕获/比较通道 2 中断标志, 参考 CC1IF 位
Bit	1	CC1IF: 捕获/比较通道 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 TIM2_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: TIM2_CNT 的值与 TIM2_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 TIM2_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 TIM2_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 TIM2_CR1 寄存器的 UDIS=0, 当计数器上溢或下溢时
	– 若 TIM2_CR1 寄存器的 UDIS=0、URS=0, 当软件设置 TIM2_EGR 寄存器的 UG 位对计数器 CNT 重新初始化时
	– 若 TIM2_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

14.10.6 EGR 事件产生寄存器 (TIM2_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.								BG	TG	Res.	CC4G	CC3G	CC2G	CC1G	UG
								w	w		w	w	w	w	w

Bits 31:8	保留, 必须保持复位值
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	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 TIM2_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 TIM2_ARR 的值

14.10.7 CCMR1 捕获模式寄存器 1（TIM2_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		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 TIM2_SMCR 寄存器的 TS 位选择）
	注 ：CC2S 仅在通道关闭时（TIM2_CCER 寄存器的 CC2E=0 且已被更新）才是可写的。
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 TIM2_CCR1 与计数器 TIM2_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 TIM2_CNT 的值与捕获/比较寄存器 1 (TIM2_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 TIM2_CNT 的值与捕获/比较寄存器 1 (TIM2_CCR1) 相同时, 强制 OC1REF 为低。
	011: 翻转。当 TIM2_CCR1=TIM2_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 TIM2_CNT<TIM2_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 TIM2_CNT>TIM2_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 TIM2_CNT<TIM2_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 TIM2_CNT>TIM2_CCR1 时通道 1 为有效电平, 否则为无效电平。
	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 13.5.7 PWM 模式)。
Bit 3	OC1PE: 输出比较 1 预装载使能
	0: 禁止 TIM2_CCR1 寄存器的预装载功能, 可随时写入 TIM2_CCR1 寄存器, 并且新写入的数值立即起作用 (默认)
	1: 使能 TIM2_CCR1 寄存器的预装载功能, 读写操作仅对预装载寄存器操作, TIM2_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: 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 (TIM2_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 TIM2_SMCR 寄存器的 TS 位选择)
	注: CC1S 仅在通道关闭时 (TIM2_CCER 寄存器的 CC1E=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 TIM2_SMCR 寄存器的 TS 位选择)
	注: CC2S 仅在通道关闭时 (TIM2_CCER 寄存器的 CC2E=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: 请注意采样频率中 $f_{\text{CK_INT}}$ 和 f_{DTS} 的差别。
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 (TIM2_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 TIM2_SMCR 寄存器的 TS 位选择)
	注: CC1S 仅在通道关闭时 (TIM2_CCER 寄存器的 CC1E=0 且已被更新) 才是可写的。

14.10.8 CCMR2 捕获模式寄存器 2 (TIM2_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]				IC3F[3:0]			IC3PSC[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 TIM2_SMCR 寄存器的 TS 位选择)
	注: CC4S 仅在通道关闭时 (TIM2_CCER 寄存器的 CC4E=0 且已被更新) 才是可写的。
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 TIM2_SMCR 寄存器的 TS 位选择）
	注：CC3S 仅在通道关闭时（TIM2_CCER 寄存器的 CC3E=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	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 TIM2_SMCR 寄存器的 TS 位选择）
	注：CC4S 仅在通道关闭时（TIM2_CCER 寄存器的 CC4E=0 且已被更新）才是可写的。
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 TIM2_SMCR 寄存器的 TS 位选择）
	注：CC3S 仅在通道关闭时（TIM2_CCER 寄存器的 CC3E=0 且已被更新）才是可写的。

14.10.9 CCER 捕获使能寄存器（TIM2_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
	注: 一旦 LOCK 级别设为 3 或 2 (TIM2_BDTR 寄存器中的 LOCK 位) 并且 CC1S=2'b00 (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 和 TI2FP1 作为触发或捕获的极性
	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; 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 TIM2_CCR1 寄存器
	0: 捕获禁止 (默认)
	1: 捕获使能

14.10.10 CNT 计数寄存器 (TIM2_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

14.10.11 PSC 预分频寄存器（TIM2_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（TIM2_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。

14.10.12 ARR 自动重装载寄存器（TIM2_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（TIM2_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。

14.10.13 CCR1 捕获寄存器 1 (TIM2_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]															
r															

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 TIM2_CNT 的值相比较, 并在 OC1 端口上产生输出信号
	如果在 OC1PE 位 (TIM2_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 (TIM2_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

14.10.14 CCR2 捕获寄存器 2 (TIM2_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]															
r															

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 TIM2_CNT 的值相比较, 并在 OC2 端口上产生输出信号
	如果在 OC2PE 位 (TIM2_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（TIM2_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

14.10.15 CCR3 捕获寄存器 3（TIM2_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]															
r															

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 TIM2_CNT 的值相比较，并在 OC3 端口上产生输出信号
	如果在 OC3PE 位（TIM2_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（TIM2_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

14.10.16 CCR4 捕获寄存器 4（TIM2_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]															

r

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 TIM2_CNT 的值相比较，并在 OC4 端口上产生输出信号
	如果在 OC4PE 位（TIM2_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（TIM2_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

14.10.17 BDTR 刹车和死区控制寄存器（TIM2_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.				BKTRIG	Res.										
				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]		Res.							
rw	rw	rw	rw	rw	rw	rw									

Bits 31:28	保留，必须保持复位值
Bit 27	BKTRIG ：TIM2_trig_oci 的触发源选择
	TIM2_BDTR.BKTRIG 位选择连接 ADC 的触发信号是否经过刹车控制。若设置 TIM2_BDTR.BKTRIG=1'b1，则触发 ADC 的 TIM2_trig_oci 为刹车处理前的信号，此时若刹车输入信号有效，TIM2 会进入刹车状态，但 ADC 仍可被触发。反之，设置 TIM2_BDTR.BKTRIG=1'b0，则触发 ADC 的 TIM2_trig_oci 为刹车处理后的信号，此时若刹车输入信号有效，TIM2 会进入刹车状态，ADC 不会被触发。
	0：刹车控制处理后的 OC _i （默认）
	1：刹车控制处理前的 OC _i
	注： 一旦 LOCK 级别设为 1（TIM2_BDTR 寄存器中的 LOCK 位）则该位不能被修改。
Bits 26:16	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 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 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（TIM2_CCER 寄存器的 CC _i E 位），则使

	能 OC _i 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)
	注: 一旦 LOCK 级别设为 1 (TIM2_BDTR 寄存器中的 LOCK 位) 则该位不能被修改。
Bit 13	BKP: TIM2_BKIN 输入极性
	0: TIM2_BKIN 输入低电平有效 (默认)
	1: TIM2_BKIN 输入高电平有效
	注: 一旦 LOCK 级别设为 1 (TIM2_BDTR 寄存器中的 LOCK 位) 则该位不能被修改。
Bit 12	BKE: TIM2_BKIN 功能使能
	0: TIM2_BKIN 功能禁止 (默认)
	1: TIM2_BKIN 功能使能
	注: 一旦 LOCK 级别设为 1 (TIM2_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 且通道输出时
	0: 当 OC _i 未使能时, OC _i 输出禁止 (默认)
	1: 当 OC _i 未使能时, 如果 CCiE=1, 则 OC _i 输出无效电平
	注: 一旦 LOCK 级别设为 2 (TIM2_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
	0: 当 OC _i 未使能时, OC _i 输出禁止 (默认)
	1: 当 OC _i 未使能时, 如果 CCiE=1, 则 OC _i 输出空闲电平
	注: 一旦 LOCK 级别设为 2 (TIM2_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 TIM2_BDTR 寄存器的 BKE、BKP、AOE 位, TIM2_CR2 寄存器的 OIS _i 位和 TIM2_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 极位 (当通道配置为输出时, TIM2_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, TIM2_CCMR _i 寄存器的 OCiM/OCiPE 位)
	注: 在系统复位后, 只能写一次 LOCK 位, 一旦写入 TIM2_BDTR 寄存器, 则其内容保持不变直至复位。
Bits 7:0	保留, 必须保持复位值

14.10.18 DCR DMA 控制寄存器 (TIM2_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 连续传输长度
	定义了 DMA 在连续模式下的传输长度（当对 TIM2_DMAR 寄存器的地址进行读或写时，TIM2 进行一次连续传输），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
	...
	11111: 32 字节
Bits 7:5	DSEL[2:0]: DMA 连续传输请求选择，默认为 000
	定义了哪个 TIM2 DMA 请求使用连续传输，其它的未选中的则使用单次传输
	000: 所有 DMA 请求都使用单次传输
	001: TIM2_CH1/TIM2_TRIG 使用连续传输
	010: TIM2_CH2/TIM2_CH3 使用连续传输
	011: TIM2_CH4/TIM2_UP/TIM2_COM 使用连续传输
	100 至 111: 保留
Bits 4:0	DBA[4:0]: DMA 基地址
	该位定义了 DMA 在连续模式下的基地址（当对 TIM2_DMAR 寄存器的地址进行读或写时），DBA 定义为从 TIM2_CR1 寄存器所在地址开始的偏移量
	00000: TIM2_CR1
	00001: TIM2_CR2
	00010: TIM2_SMCR
	...
	举例：当 DBL=7 且 DBA=00000 时，表示从 TIM2_CR1 开始向下读取或发送 7 个寄存器的数据。

14.10.19 DMAR DMA 传输寄存器（TIM2_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 TIM2_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： TIM2_CR1 地址 + (DBA + DMA 指针) x4，其中“TIM2_CR1 地址”是 TIM2 寄存器的起始地址；DBA 是 TIM2_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 TIM2_DCR 寄存器中定义的 DBL。
--	--

14.10.20 TIM2 互联配置寄存器 8（SysCtrl_EDU_CFG8）

SysCtrl_EDU_CFG8 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 0x05FA659A 到 SysCtrl_KEY 密钥寄存器，且在随后的 64 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。

地址：0x4800_7040

复位值：0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.												TIM2_ETRS[2:0]			
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TIM2_TI4S[2:0]			Res.	TIM2_TI3S[2:0]			Res.	TIM2_TI2S[2:0]			Res.	TIM2_TI1S[2:0]		
	rw				rw				rw				rw		

Bits 31:19	保留，必须保持复位值
Bits 18:16	TIM2_ETRS[2:0] : TIM2 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: 保留
	010: ACMP0 输出
	011: ACMP1 输出
	100: TIM1_TRIG_OC1
	101: TIM1_TRIG_OC5
	110: TIM15_TRIG_OC1
	111: TIM16_TRIG_OC1
Bit 15	保留，必须保持复位值
Bits 14:12	TIM2_TI4S[2:0] : TIM2 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 窗口看门狗
	010: ACMP0 输出
	011: ACMP1 输出
	100: TIM1_TRIG_OC1
	101: TIM1_TRIG_OC5
	110: TIM15_TRIG_OC1
	111: TIM16_TRIG_OC1
Bit 11	保留，必须保持复位值
Bits 10:8	TIM2_TI3S[2:0] : TIM2 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 窗口看门狗
	010: ACMP0 输出
	011: ACMP1 输出
	100: TIM1_TRIG_OC1
	101: TIM1_TRIG_OC5
	110: TIM15_TRIG_OC1
	111: TIM16_TRIG_OC1
Bit 7	保留, 必须保持复位值
Bits 6:4	TIM2_TI2S[2:0]: TIM2 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 窗口看门狗
	010: ACMP0 输出
	011: ACMP1 输出
	100: TIM1_TRIG_OC1
	101: TIM1_TRIG_OC5
	110: TIM15_TRIG_OC1
	111: TIM16_TRIG_OC1
Bit 3	保留, 必须保持复位值
Bits 2:0	TIM2_TI1S[2:0]: TIM2 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 窗口看门狗
	010: ACMP0 输出
	011: ACMP1 输出
	100: TIM1_TRIG_OC1
	101: TIM1_TRIG_OC5
	110: TIM15_TRIG_OC1
	111: TIM16_TRIG_OC1
注: TIMx_TRIG_OC1 和 TIMx_TRGO 不同, 详见 TIM 章节说明。	

14.10.21 TIM2 互联配置寄存器 9 (SysCtrl_EDU_CFG9)

SysCtrl_EDU_CFG9 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 0x05FA659A 到 SysCtrl_KEY 密钥寄存器, 且在随后的 64 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。

地址: 0x4800_7044

复位值: 0x0000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.		TIM2_OCCSE[5:0]							Res.						
		rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		TIM2_BKSE[6:0]							Res.	TIM2_SP[6:0]					
		rw								rw					

Bits 31:30	保留，必须保持复位值
Bits 29:24	TIM2_OCCSE[5:0]: TIM2 OCREF_CLR 输入触发源使能控制，默认值 000000
	TIM2 OCREF_CLR 输入源为 ADC 窗口看门狗 1/ADC 窗口看门狗 0/ TIM15_TRGO / TIM1_TRGO / 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 TIM2_OCCSE[x]配置
	所有输入源的有效触发电平，通过控制位 TIM2_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
	TIM2_OCCSE[x] = 0: 相应触发源不使能
	TIM2_OCCSE[x] = 1: 相应触发源使能
Bits 23:15	保留，必须保持复位值
Bits 14:8	TIM2_BKSE[6:0]: TIM2_BKIN 输入触发源使能控制，默认值 0000000
	TIM2_BKIN 输入触发源为外部 GPIO 输入/ADC 窗口看门狗 1/ADC 窗口看门狗 0/ TIM1_TRGO / TIM15_TRGO / 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 TIM2_BKSE[x]配置
	所有输入源的有效触发电平，通过控制位 TIM2_SP[x]配置
	TIM2_BKIN 使能的输入源的有效电平触发通过逻辑或的方式组合
	TIM2_BKSE[x] = 0: 相应触发源不使能
	TIM2_BKSE[x] = 1: 相应触发源使能
Bit 7	保留，必须保持复位值
Bits 6:0	TIM2_SP[6:0]: TIM2_BKIN/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 TIM2_SP[x]配置（低位对齐）
	TIM2_SP[x] = 0: 相应触发源的有效电平是高
	TIM2_SP[x] = 1: 相应触发源的有效电平是低
注: TIMx_TRIG_OCI 和 TIMx_TRGO 不同，详见 TIM 章节说明。	

15. 通用定时器 TIM15/16/17

LCM32F06X 包括 1 个 PWM 专用定时器、4 个通用定时器、1 个看门狗定时器和 1 个系统时基定时器。具体差异请参考第 13 章 PWM 专用定时器 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 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。

15.1.1 TIM15 主要特性

TIM15 拥有 2 个独立的通道，支持 2 路互补的 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。

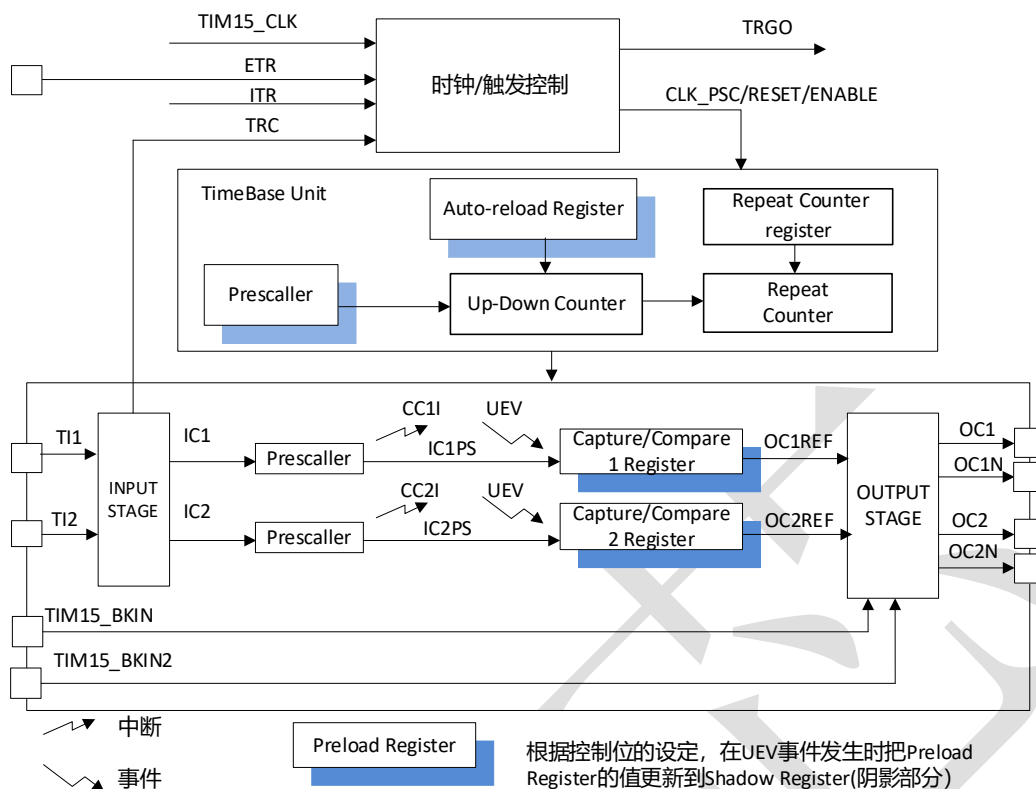


图 15-1 通用定时器 TIM15 框图

15.1.2 TIM16 主要特性

TIM16 拥有 2 个独立的通道，支持 1 路互补的 PWM 输出，能插入可编程的死区时间。TIM16 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 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 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 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。

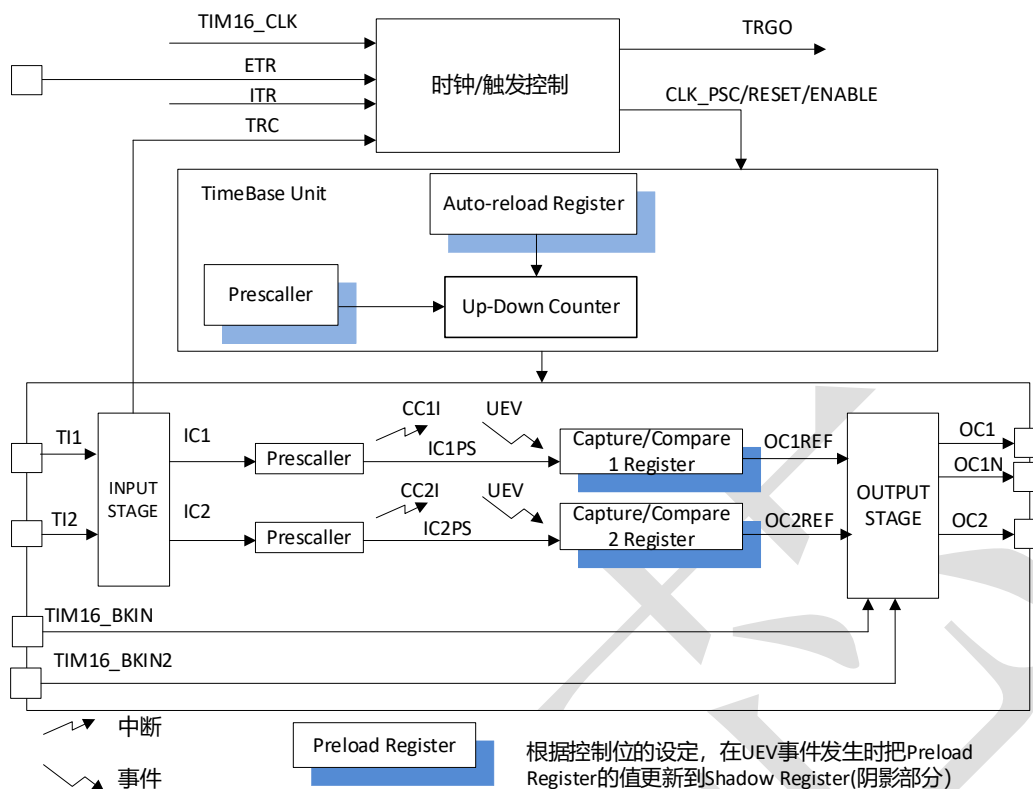


图 15-2 通用定时器 TIM16 框图

15.1.3 TIM17 主要特性

TIM17 拥有 1 个通道，支持 1 路互补的 PWM 输出，能插入可编程的死区时间。TIM17 的 1 个通道可以被用于：

- 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，TIM17 具有全调制能力（0 到 100%）。在调试模式下，计数器可以被冻结。TIM17 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。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。

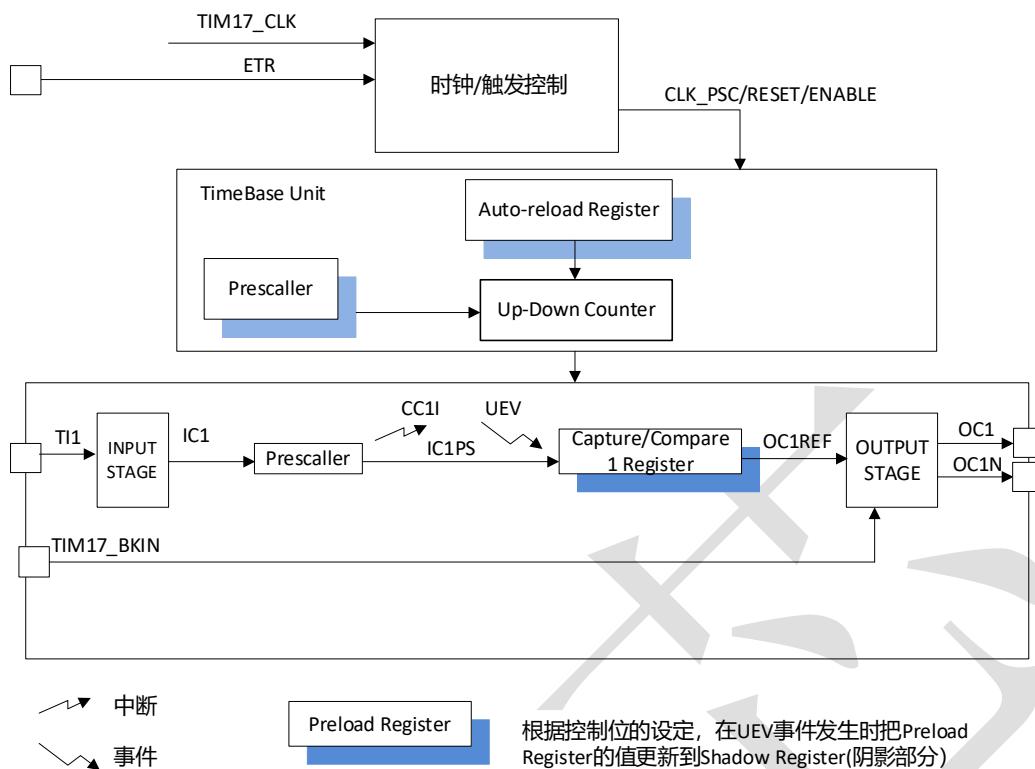


图 15-3 通用定时器 TIM17 框图

15.2 管脚配置

表 15-1 TIM15/16/17 管脚配置

管脚名称	管脚类型	管脚描述	复用 I/O 口	备注
TIM15_CH1	I/O	TIM15 捕获/输出通道 1	PA11/PA3/PA5/PB1/PA9	-
TIM15_CH1N	O	TIM15 互补通道 1	PA12/PA1	-
TIM15_CH2	I/O	TIM15 捕获/输出通道 2	PB8/PB2/PA2/PA4/PA8/PA10/PB6	-
TIM15_CH2N	O	TIM15 互补通道 2	PA9	-
TIM15_GPIO_BKIN	I	TIM15 的外部 I/O 刹车	PA9/PA15/PB2	-
TIM15_ETR	I	TIM15 的外部 I/O 触发	PA0/PA5	-
TIM16_CH1	I/O	TIM16 捕获/输出通道 1	PB8/PA0/PA6/PA13/PB3	-
TIM16_CH1N	O	TIM16 互补通道 1	PB2/PA1/PA15/PB6	-
TIM16_CH2	I/O	TIM16 捕获/输出通道 2	PA1/PA5/PB4	TIM16_CH2 没有互补通道
TIM16_GPIO_BKIN	I	TIM16 的外部 I/O 刹车	PA8/PA9/PB2	-
TIM16_ETR	I	TIM16 的外部 I/O 触发	PA1/PB3	-

管脚名称	管脚类型	管脚描述	复用 I/O 口	备注
TIM17_CH1	I/O	TIM17 捕获/输出通道 1	PB7/PA4/PA7/PA14/PA15	-
TIM17_CH1N	O	TIM17 互补通道 1	PB8/PB7	-
TIM17_GPIO_BKIN	I	TIM17 的外部 I/O 刹车	PA10/PB4/PB2	-
TIM17_ETR	I	TIM17 的外部 I/O 触发	PB8/PA8	-

15.3 时基单元

时基单元包含：

- 16位向上/向下计数寄存器（TIMx_CNT），以下x均表示15/16/17
- 16位预分频寄存器（TIMx_PSC）
- 16位自动重载寄存器（TIMx_ARR）
- 8位重复计数寄存器（TIM15_RCR）（仅TIM15具有）

16 位计数寄存器、预分频寄存器、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 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。

写计数寄存器的操作没有缓存，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。

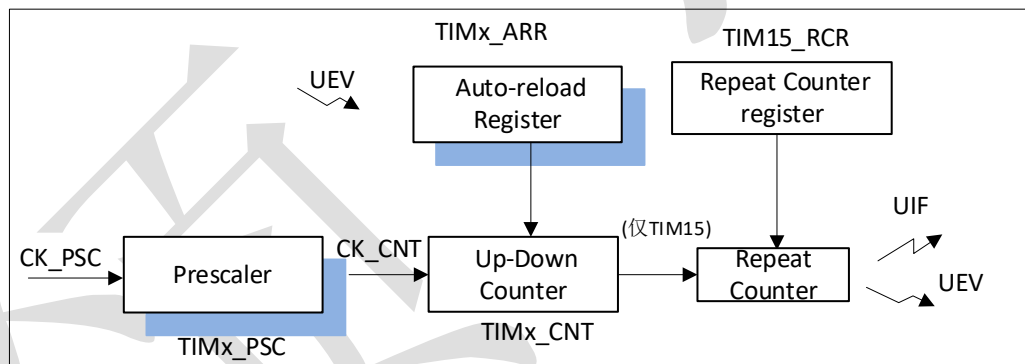


图 15-4 定时器时基单元

15.3.1 自动重载寄存器

自动重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 TIMx_CR1.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.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.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.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 TIMx_EGR.UG 位
- 时钟/触发控制器产生硬件复位

在预装载使能时（TIMx_CR1.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.UDIS 位将禁止产生更新事件（UEV）。

15.3.2 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影子寄存器，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。

图 15-5 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。

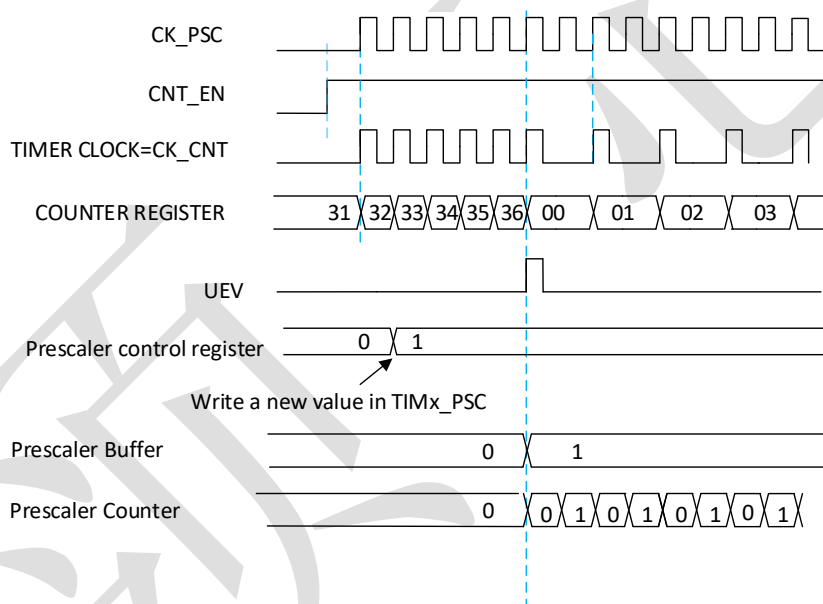


图 15-5 预分频系数从 1 变 2 的计数器时序

15.4 计数器模式

15.4.1 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），然后重新从 0 开始计数并产生一个计数器上溢出事件，同时，如果 TIMx_CR1.UDIS 位是 0，将会产生一个更新事件（UEV）。

使用软件置位 TIMx_CR1.UDIS 位，可以禁止产生更新事件，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，如配置置 TIMx_CR1.URS=0 位（选择更新请求源），置位 TIMx_EGR.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，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影子寄存器，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.UIF 位），自动重装载影子寄存器会更新为新的预装载寄存器的值（TIMx_ARR）。预分频寄存器的影子寄存器会被更新为新的预装载寄存器的值（TIMx_PSC）。

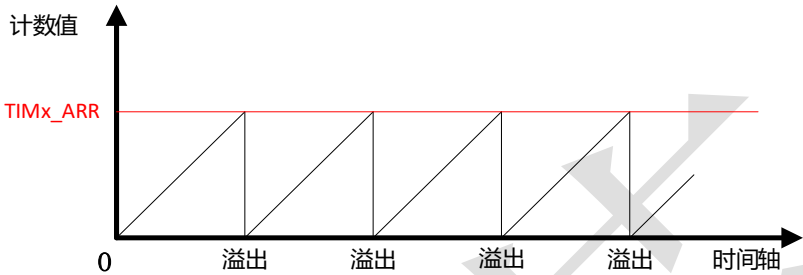


图 15-6 向上计数溢出

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。

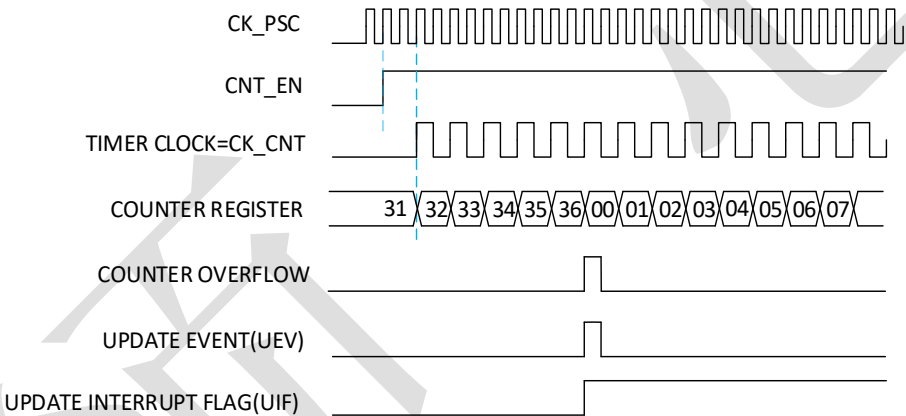


图 15-7 计数器在二分频时钟频率下的时序图

15.4.2 向下计数模式

在向下计数模式中，计数器从自动重装载寄存器的值（TIMx_ARR）开始向下计数到 0，然后再从 TIMx_ARR 重新开始计数，并产生一个计数器向下溢出事件。如果 TIMx_CR1.UDIS 位是 0，将会产生一个更新事件（UEV）。

置位 TIMx_CR1.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，如果配置 TIMx_CR1.URS=0（选择更新请求源），置位 TIMx_EGR.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，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影子寄存器，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.UIF 位），自动重装载影子寄存器会被更新为新的预装载寄存器的值（TIMx_ARR）。预分频寄存器的影子寄存器会更新为新的预装载寄存器的值（TIMx_PSC）。

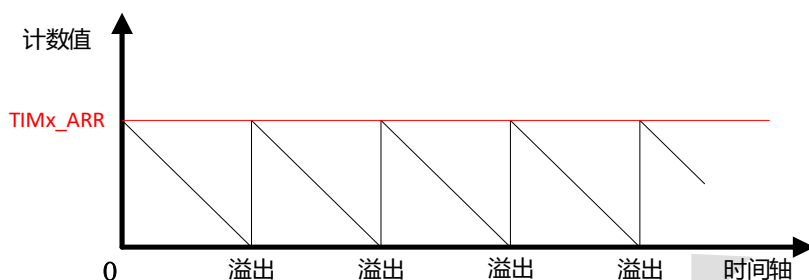


图 15-8 向下计数溢出

15.4.3 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 0 并且产生一个计数器下溢事件；然后再从 0 开始重新计数。如果 TIMx_CR1.UDIS 位是 0，计数器向上向下溢出都会将会产生一个更新事件（UEV）。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。

在此模式下，不能写入 TIMx_CR1.DIR 方向位。它由硬件更新并指示当前的计数方向。

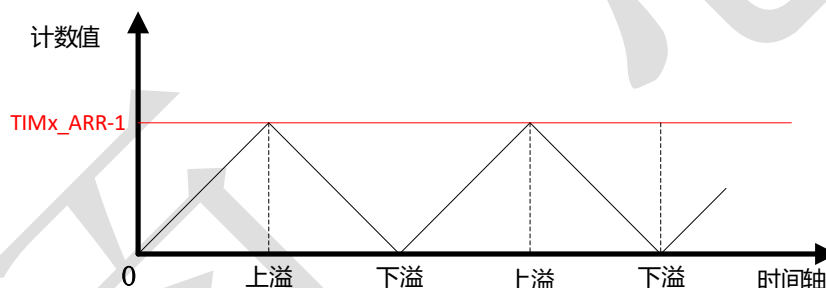


图 15-9 中央对齐溢出

如果定时器带有重复计数器，在重复了指定次数的向上和向下溢出之后才会产生更新事件（UEV）。否则每一次的向上向下溢出都会产生更新事件。

设置 TIMx_CR1.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，如果配置 TIMx_CR1.URS=0（选择更新请求源），置位 TIMx_EGR.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，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影子寄存器，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.UIF 位）：自动重装载影子寄存器会更新为新的预装载寄存器的值（TIMx_ARR）。预分频寄存器的影子寄存器的会更新为新的预装载寄存器的值（TIMx_PSC）。

使用中央对齐模式：

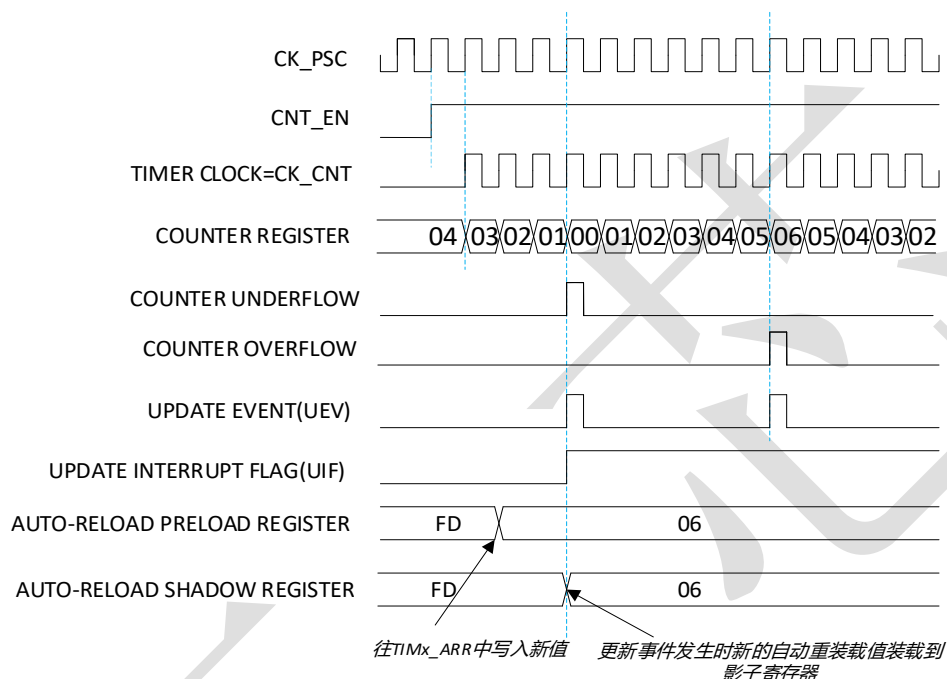
启动中央对齐模式时，计数器将按照原有的向上/向下的配置计数。也就是说 TIMx_CR1.DIR 位将决定

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：

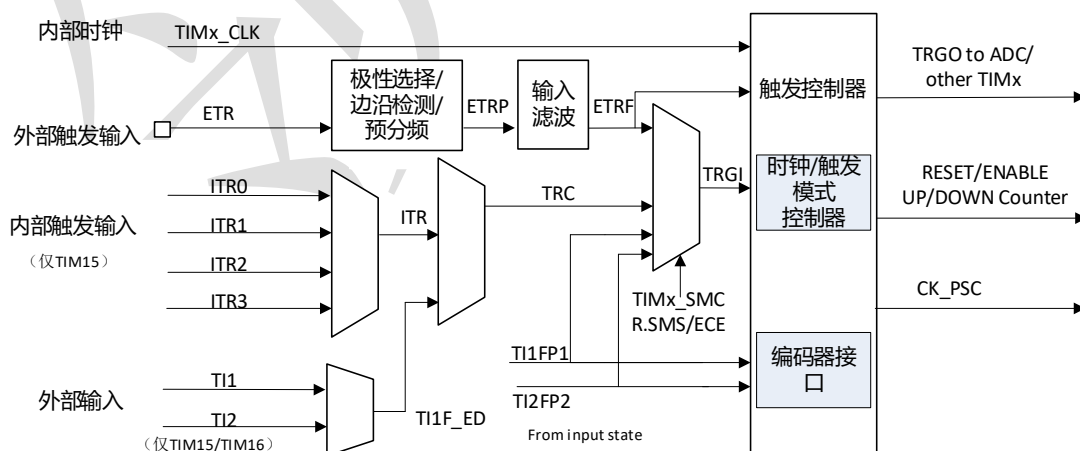
- 向计数器写入了比自动装载值更大的数值时（TIMx_CNT > TIMx_ARR），但计数器的计数方向不发生改变。例如计数器已经向上溢出，但计数器仍然向上计数。
- 向计数器写入了 0 或者 TIMx_ARR 的值，但更新事件不发生。

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.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。



15.5 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），如图 15-11 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（TIMx_CLK）：来自于 SYS_CLK 或 PCLK0
- 外部时钟模式 1：外部触发输入（Tli）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。

15.5.1 内部时钟源（TIMx_CLK）

如果 TIMx_SMCR.SMS=000，TIMx_SMC.ECE=0，则 TIMx_CR1.CEN、TIMx_CR1.DIR 和 TIMx_EGR.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。

15.5.2 外部时钟模式 1

当 TIMx_SMCR.SMS=3'b111 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，下图中以 TIM15/16 为例。

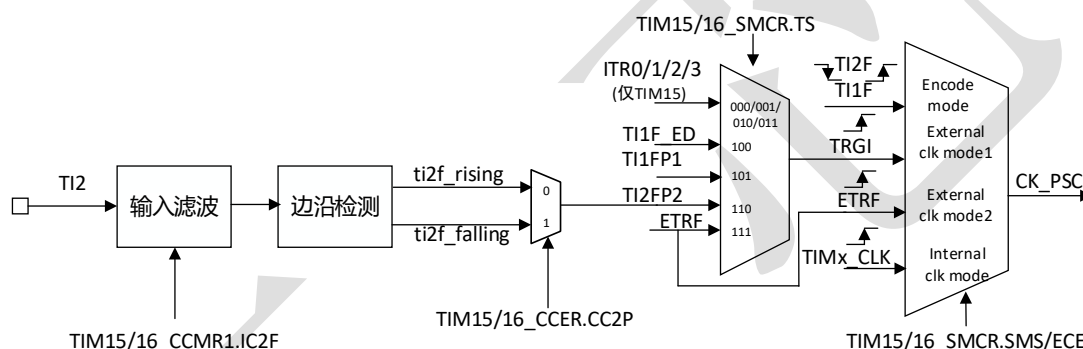


图 15-12 TI2 外部时钟框图

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.CC2S=2'b01，使用通道 2 检测 TI2 输入的上升沿
2. 配置 TIM15_CCMR1.IC2F 位，选择输入滤波器带宽（如果不需要滤波器，保持 IC2F=4'b0000）
3. 配置 TIM15_CCER.CC2P=0，选定上升沿极性
4. 配置 TIM15_SMCR.SMS=3'b111，配置计数器使用外部时钟模式 1
5. 配置 TIM15_SMCR.TS=3'b110，选定 TI2 作为输入源
6. 设置 TIM15_CR1.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.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 输入端的同步电路（图 15-13 预分频系数为 1）。

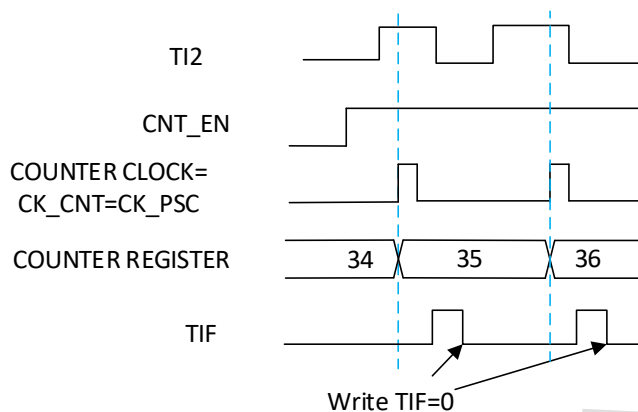


图 15-13 外部时钟模式 1 下的控制电路

15.5.3 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.ECE 位写 1，即可选定此模式。

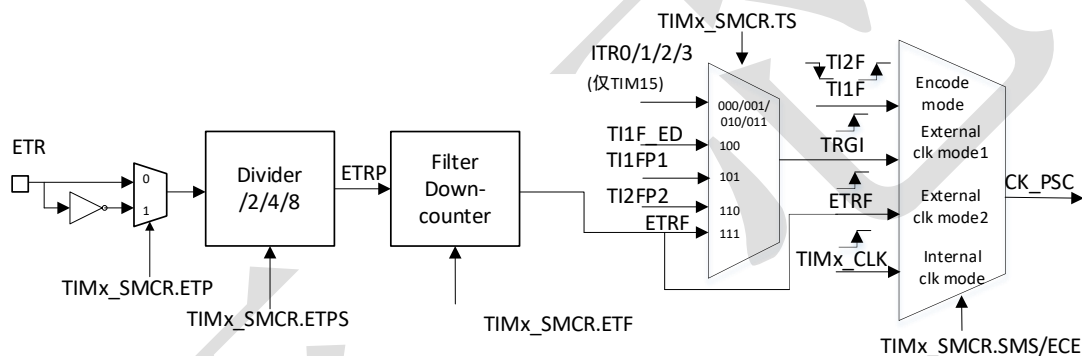


图 15-14 外部触发时钟框图

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.ETF=4'b0000
2. 设置预分频器，配置 TIM15_SMCR.ETPS=2'b01
3. 选择 ETR 的上升沿检测，配置 TIM15_SMCR.ETP=0
4. 开启外部时钟模式 2，配置 TIM15_SMCR.ECE=1
5. 启动计数器，配置 TIM15_CR1.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 信号端的同步电路。

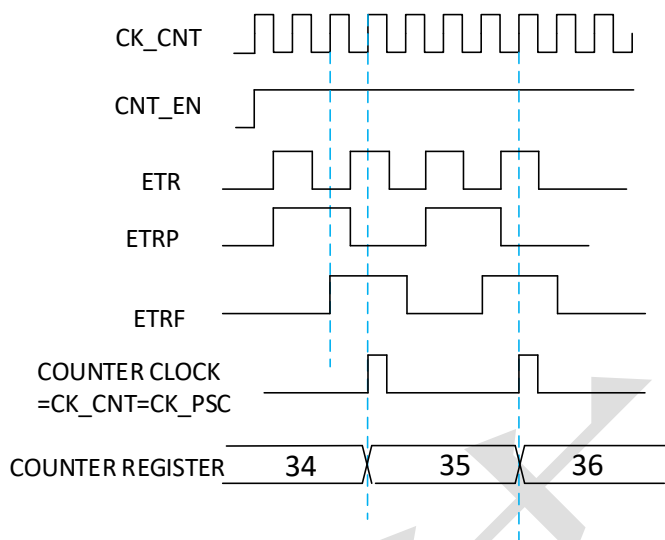


图 15-15 外部时钟模式 2 下的控制电路

15.6 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 TIMx_CCMRi.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)。

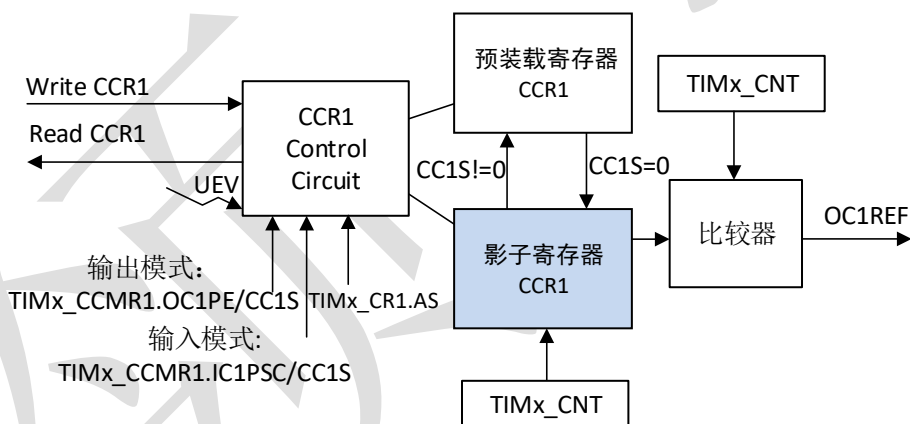


图 15-16 捕获/比较通道 1 的主电路

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。比较模式下, 读过程可根据 TIMx_CR1.AS 位选择读取影子寄存器或预装载寄存器, 写过程中如果 TIMx_CCMi.OC1PE=0, 则可直接写入到影子寄存器。在比较模式下, TIMx_CCMi.OC1PE=1 且在 UEV 事件到来时预装载寄存器的内容被复制到影子寄存器中, 然后使用影子寄存器的值和计数器进行比较。

当通道被配置成输出模式时 (TIMx_CCMRi.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, 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)。

15.6.1 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 TIMx_SR.CCiIF 标志被置 1。

如果 TIMx_DIER.CCiIE 位被置 1，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 TIMx_SR.CCiOF 被置 1，写 CCIIF=0 可清除 CCIIF。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：

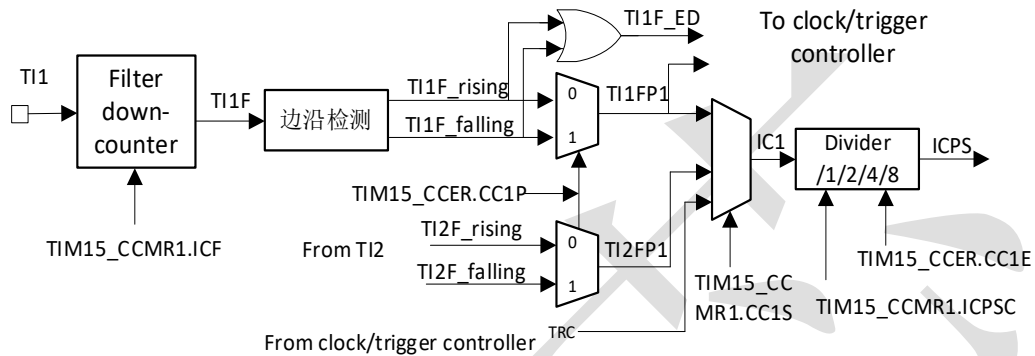


图 15-17 TIM15 通道 1 的输入

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.CC1S=2'b01，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 Tli 的特点，可通过配置 TIMx_CCMRi.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 TIMx_CCMR1.IC1F=4'b0011，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，配置 TIMx_CCER.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（TIMx_CCMR1.IC1PSC=2'b00）。
5. 设置 TIMx_CCER.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.CCiIE 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。
- CCI1F 标志被设置（中断标志）。当发生至少 2 个连续的捕获时，而 CCI1F 未曾被清除时，CCI1OF 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（CCI1OF 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。

15.6.2 PWM 输入

这里 PWM 输入模式只有 TIM15 和 TIM16 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

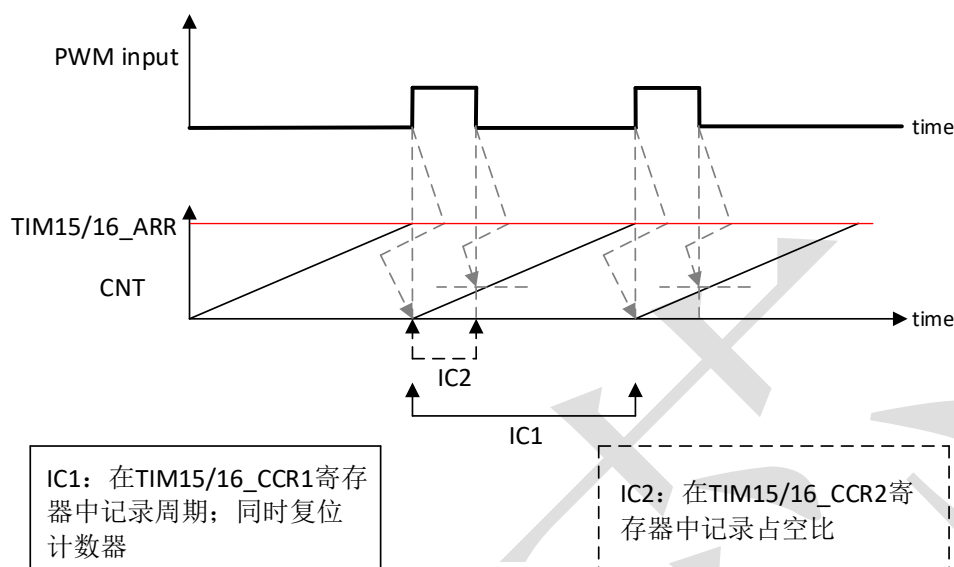


图 15-18 PWM 输入信号测量

例如，可以用以下方式测量 TI1 上输入的 PWM 信号的周期（锁存于 TIM15/16_CCR1 寄存器）和占空比（锁存于 TIM15/16_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 TIM15/16_CCR1 的有效输入：配置 TIM15/16_CCMR1.CC1S=2'b01（选中 TI1）。
2. 选择 TI1FP1 的有效极性（用来捕获数据到 TIM15/16_CCR1 中和清除计数器）：配置 TIM15/16_CCER.CC1P=0（上升沿有效）。
3. 选择 TIM15/16_CCR2 的有效输入：配置 TIM15/16_CCMR1.CC2S=2'b10（选中 TI1FP2）。
4. 选择 TI1FP2 的有效极性（捕获数据到 TIM15/16_CCR2）：配置 TIM15/16_CCER.CC2P=1（下降沿有效）。
5. 选择有效的触发输入信号：配置 TIM15/16_SMCR.TS=3'b101（选择 TI1FP1）。
6. 配置触发模式控制器为复位触发模式：配置 TIM15/16_SMCR.SMS=3'b100。
7. 使能捕获：配置 TIM15/16_CCER 寄存器中 CC1E=1，CC2E=1。

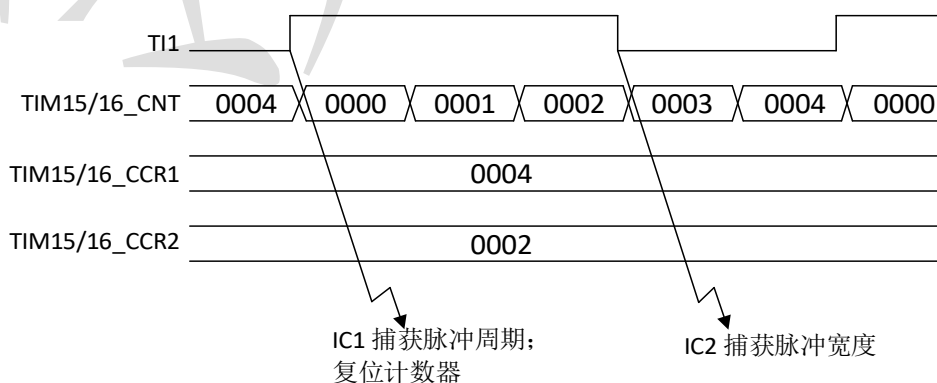


图 15-19 PWM 输入信号测量实例

15.6.3 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.CCiS=2'b00）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。

配置 TIMx_CCMRi.OCiM=3'b101，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。

配置 TIMx_CCMRi.OCiM=3'b100，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。这将会在下节的输出比较模式一节中介绍。

15.6.4 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等时，有如下操作：

- 根据不同的输出比较模式（由 TIMx_CCMRi.OCiM 设定），相应的 OCiREF 输出信号为：
 - 保持不变（OCiM=3'b000）
 - 设置为有效电平（OCiM=3'b001）
 - 设置为无效电平（OCiM=3'b010）
 - 翻转（OCiM=3'b011）
- 置位中断状态寄存器中的标志位（TIMx_SR.CCiIF 位）
- 若设置了相应的中断使能位（TIMx_DIER.CCiIE 位），则产生一个中断

TIMx_CCER.CCiP 位用于选择有效和无效的电平极性。TIMx_CCMRi.OCiPE 位用于选择 TIMx_CCRi 寄存器是否需要使用预装载寄存器。

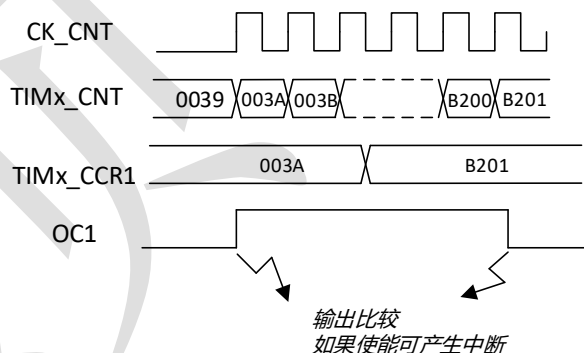


图 15-20 输出比较模式，翻转 OC1

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=3'b011
 - 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 $TIMx_CR1.CEN$ 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。

15.6.5 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 $TIMx_CCMRi.OCiM$ 位写入 3'b110 (PWM 模式 1) 或 3'b111 (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 $TIMx_CCMRi.OCiPE$ 位使能相应的预装载寄存器，若设置 $TIMx_CR1.ARPE=1$ ，则使能 $TIMx_ARR$ 寄存器的预装载功能（在向上计数模式或中央对称模式中）。只有当发生一个更新事件的时候，此预装载值才传输至影子寄存器中，因此在计数器开始计数之前，必须通过设置 $TIMx_EGR.UG$ 位来初始化所有的寄存器。

OCi 的极性可以通过软件在 $TIMx_CCER.CCiP$ 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。

15.6.5.1 PWM 边沿对齐模式

向上计数配置

当 $TIMx_CR1.DIR$ 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装载值 ($TIMx_ARR$)，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。

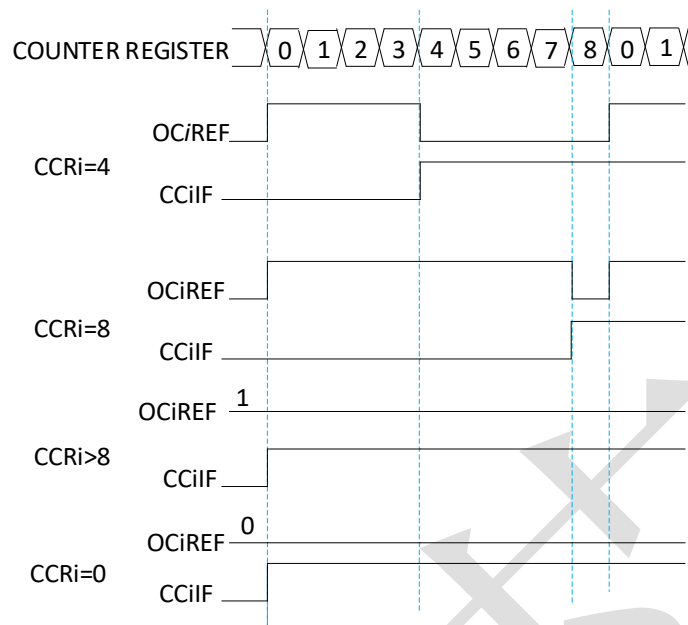


图 15-21 TIMx_ARR=8 时边沿对齐的 PWM 波形实例

向下计数的配置

当 TIMx_CR1.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 中的自动重装载值，则 OCiREF 保持为 1。该模式下不能产生 0% 的 PWM 波形。

15.6.5.2 PWM 中央对齐模式

当 TIMx_CR1.CMS != 0 时为中央对齐模式。

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.DIR 方向位由硬件更新，不要的软件修改它。

下图为 PWM 中央对齐模式波形示例：

- 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 15-22 中标出）
 - 只有在计数器向下计数时（中央对齐模式 1，CMS=2'b01）
 - 只有在计数器向上计数时（中央对齐模式 2，CMS=2'b10）
 - 在计数器向上和向下计数时（中央对齐模式 3，CMS=2'b11）

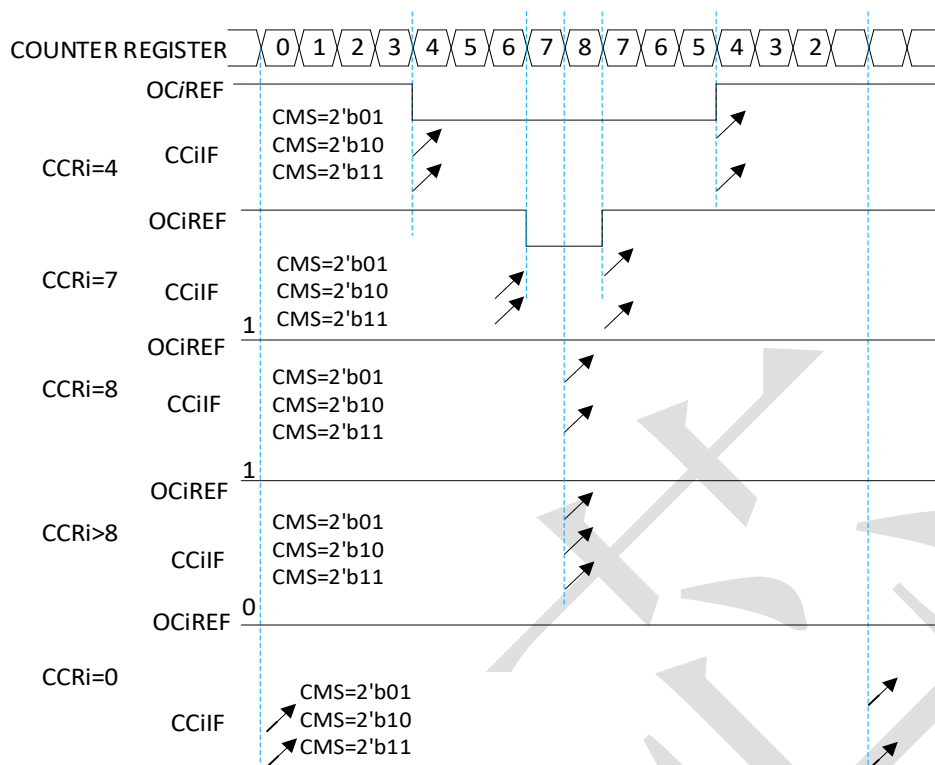


图 15-22 中央对齐模式 PWM (TIMx_ARR=8)

15.6.6 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.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 产生时停止。

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 15-23 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（仅在 TIM15/16 中才有），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来定义（需要考虑到时钟频率和预分频器）：

1. TI2FP2 对应到 TI2，配置 TIM15/16_CCMR1.CC2S=2'b01
2. TI2FP2 上升沿检测，配置 TIM15/16_CCER.CC2P=0
3. 设置 TI2FP2 为从模式下的触发信号 (TRGI)，配置 TIM15/16_SMCR.TS=3'b110
4. 设置 TI2FP2 为计数器的启动信号，配置 TIM15/16_SMCR.SMS=3'b110（触发模式）
5. t_{DELAY} 通过 TIM15/16_CCR1 寄存器配置得到
6. t_{PULSE} 通过 TIM15/16_ARR - TIM15/16_CCR1 得到
7. 如果希望高电平脉冲是在计数器到达 TIM15/16_CCR1 时信号由 0 变为 1，到达 TIM15/16_ARR 时

信号由 1 变为 0，则设置 PWM 模式 2，配置 TIM15/16_CCMR1.OC1M=3'b111

8. 可选启用预装载功能，配置 TIM15/16_CCMR1.OC1PE=1 和 TIM15/16_CR1.ARPE=1。如果使用预装载功能，在 TIM15/16_ARR 和 TIM15/16_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，DIR=0，CMS=2'b00。

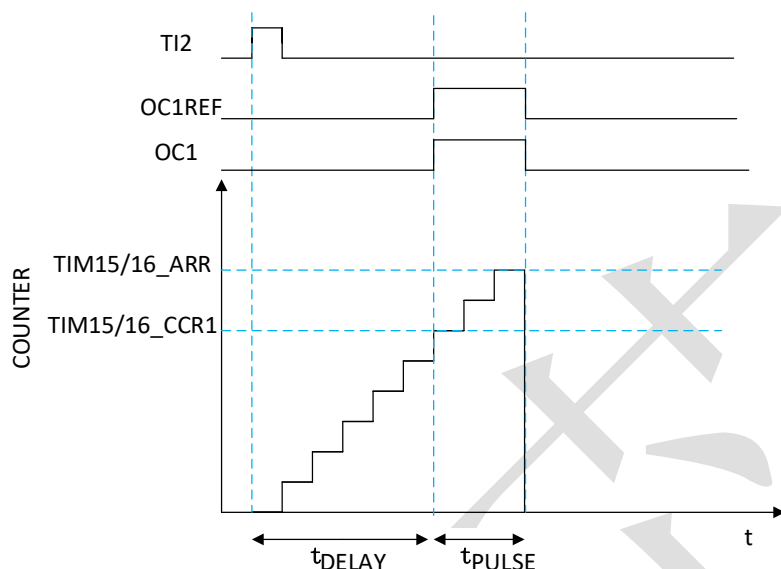


图 15-23 单脉冲模式

15.6.7 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.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。

15.6.8 编码器接口模式

编码器接口仅 TIM15 和 TIM16 支持，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，则配置 TIM15/16_SMCR.SMS=3'b001；如果只在 TI1 边沿计数，则配置 SMS=3'b010；如果计数器同时在 TI1 和 TI2 边沿计数，则配置 SMS=3'b011。通过设置 TIM15/16_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 15-2，假定计数器已经启动（TIM15/16_CR1.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 TIM15/16_ARR 寄存器的自动重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 TIM15/16_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 15-2 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 15-2 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 15-24 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：

- TIM15/16_CCMR1.CC1S=2'b01（TI1FP1 映射到 TI1）
- TIM15/16_CCMR1.CC2S=2'b01（TI2FP2 映射到 TI2）
- TIM15/16_CCER.CC1P=0，TIMx_CCMR1.IC1F=4'b0000（TI1FP1 不反相，TI1FP1=TI1）
- TIM15/16_CCER.CC2P=0，TIMx_CCMR1.IC2F=3'b0000（TI2FP2 不反相，TI2FP2=TI2）
- TIM15/16_SMCR.SMS=3'b011（TI1 和 TI2 输入均在上升沿和下降沿有效）
- TIM15/16_CR1.CEN=1（计数器使能）

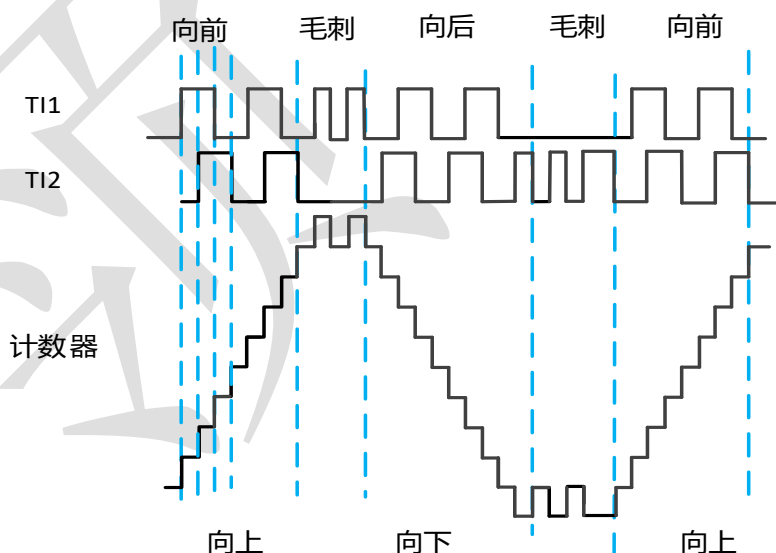


图 15-24 编码器接口模式示例

当其他配置保持一致，TI1FP1 极性反相时的示例：

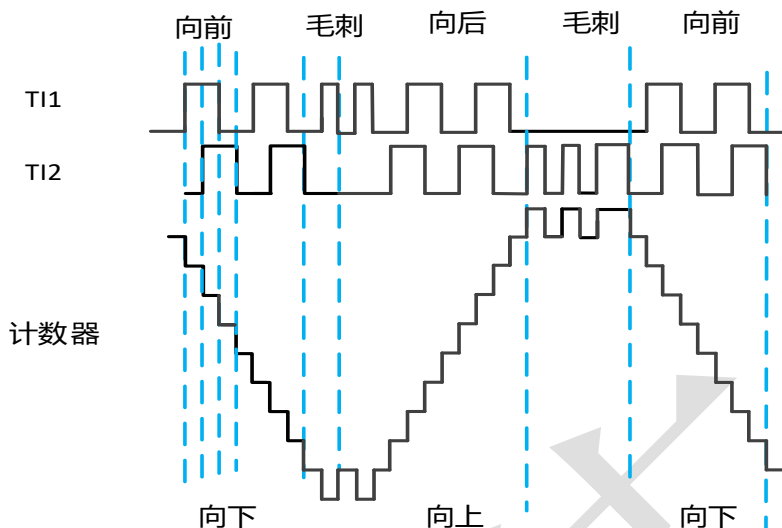


图 15-25 TI1FP1 反相的编码器接口模式示例

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。

15.6.9 互补输出和死区插入

TIM15 能够输出两路互补信号，TIM16/TI17 能够输出 1 路互补信号，并且能够对输出的瞬时关断和接通进行延时管理。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 15-26 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，如图 15-27 和图 15-28 所示。

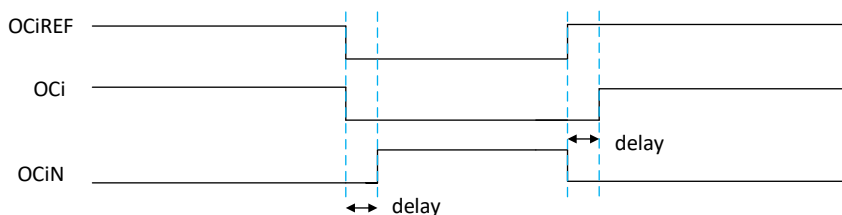


图 15-26 带死区插入的互补输出

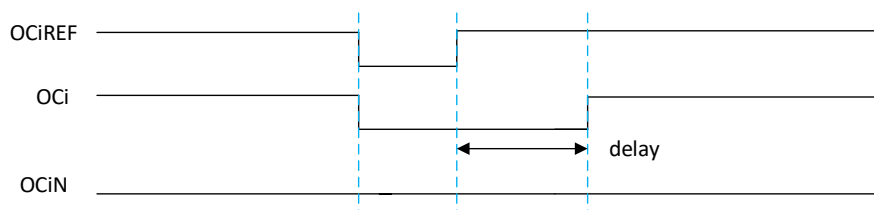


图 15-27 死区波形延迟大于负脉冲

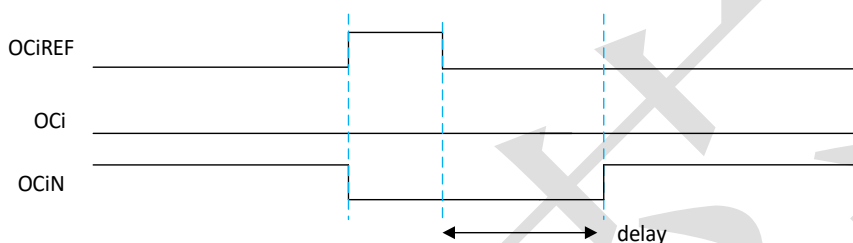


图 15-28 死区波形延迟大于正脉冲

15.6.10 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），OCi 和 OCiN 受到死区时间控制，且 OCi 和 OCiN 输出相位相反。

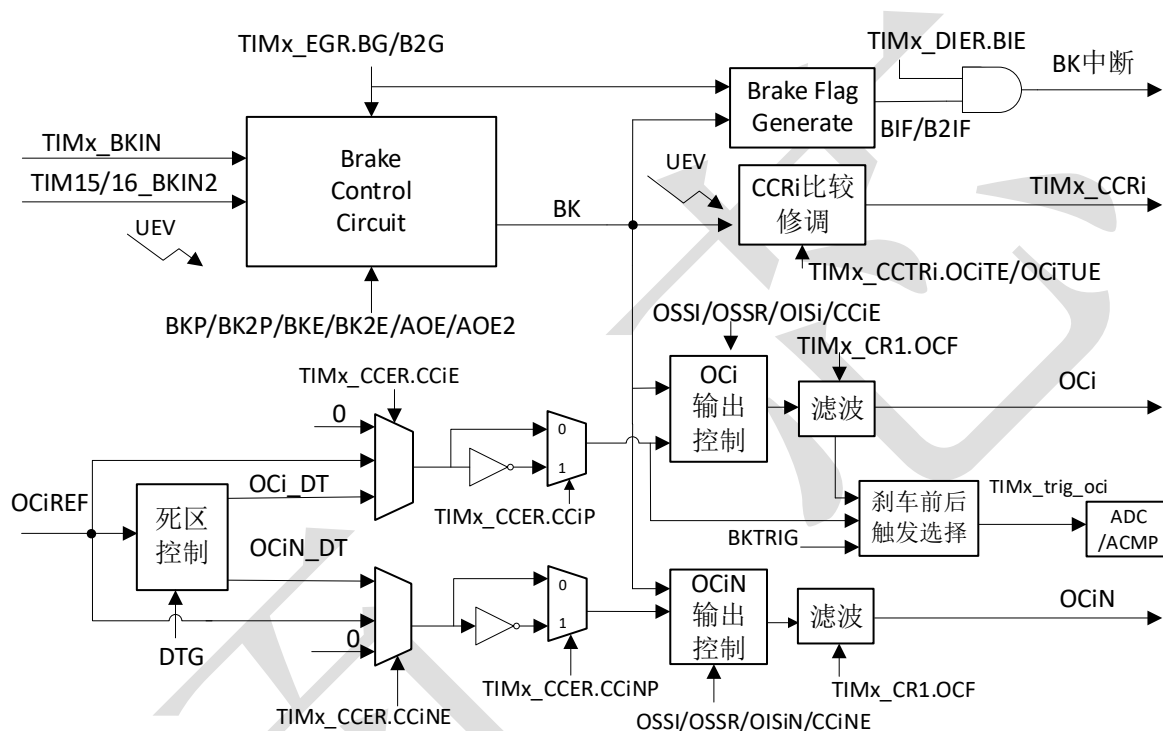
15.6.11 刹车功能

刹车功能常用于马达控制中。当刹车功能开启后，刹车源输入有效时，输出使能信号和电平将会依据刹车控制位的配置（TIMx_BDTR 寄存器中的 MOE、OSS1 和 OSSR 位，TIMx_CR2 寄存器中的 OISi 和 OISiN 位）立即生效。TIM15 和 TIM16 有 2 路刹车通道，TIM15_BKIN 和 TIM15_BKIN2，TIM16_BKIN 和 TIM16_BKIN2。TIM17 有 1 路刹车通道，TIM17_BKIN。各个通道输入刹车源不同，且分别具有各自的刹车控制位，互联配置寄存器 5/6/7（SysCtrl_EDU_CFG5/6/7）中给出具体各个通道的刹车源。对于 TIM15/16/17_BKIN，刹车源还包括系统产生的 2 个特殊的刹车源，分别为发生 CPU 内核锁死和电压过低触发 LVD 产生的低压检测标志，这两个刹车源不受使能和极性控制，具体描述请见芯片控制寄存器 ChipCtrl_CTRL.LOCKUPEN 和 ChipCtrl_CTRL.LVD_LOCK 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，TIMx_BDTR.MOE 位为低。配置 TIMx_BDTR.BKE 位可以使能 TIMx_BKIN 功能，其刹车输入信号的极性可以通过配置 TIMx_BDTR.BKP 位选择。BKE 和 BKP 可以被同时修改。同理可配置 TIM15/16_BDTR.BK2E 和 TIM15/16_BDTR.BK2P 分别进行 TIM15/16_BKIN2 使能和其刹车输入信号极性控制。

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。



例如当刹车通道 1 中有刹车源触发时:

更新：如果 $TIMx_CCMRi.OCiTE=1$ ，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，刹车标志 BIF/B2IF 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/BK2E 位和 BKP/BK2P 位
- 通过软件配置 $TIMx_EGR.BG/B2G$ 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 $TIMx_BDTR.LOCK$ 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。

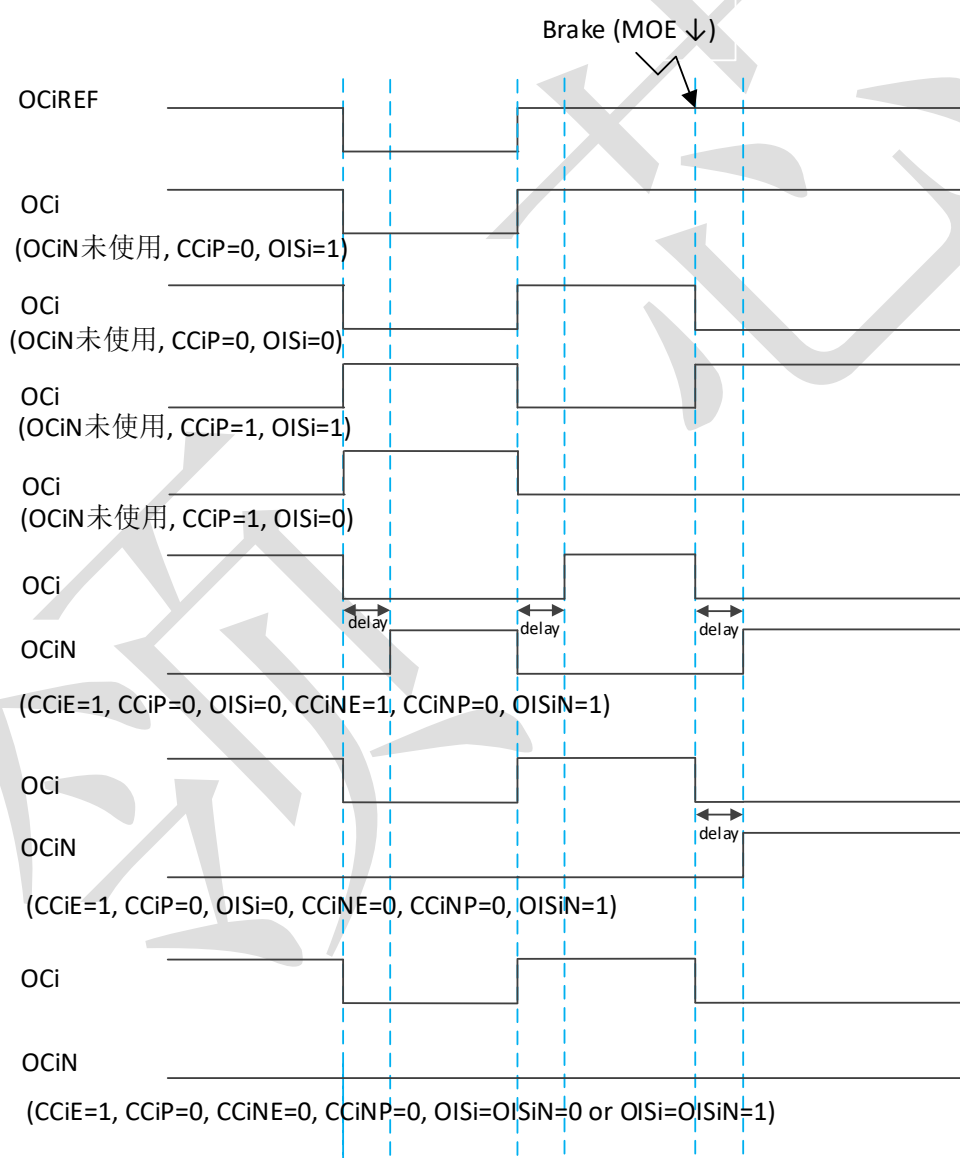


图 15-30 刹车对应的输出

15.7 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。

15.7.1 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，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，TIM15_CCMR1.IC2F=4'b0000）
- 触发操作中不使用捕获预分频器，不需要配置；TIM15_CCMR1.CC2S=2'b01，用于选择输入捕获源
- 配置 TIM15_CCER.CC2P=0，选择上升沿做为触发条件
- 配置 TIM15_SMCR.SMS=3'b110，选择定时器为触发模式
- 配置 TIM15_SMCR.TS=3'b110，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 TIM15_SR.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 输入端的同步电路。

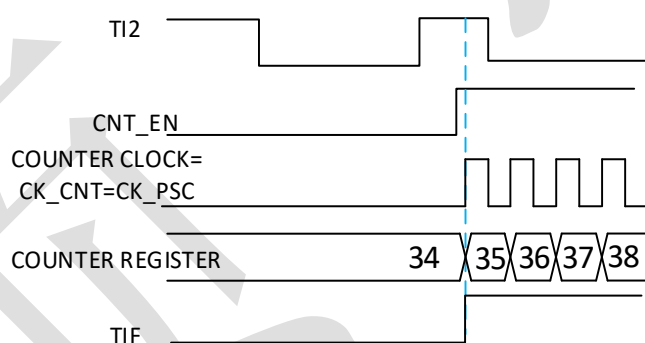


图 15-31 标准触发模式下的时序图

15.7.2 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.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 TIMx_CCMR1.IC1F=4'b0000）
- 触发操作中不使用捕获预分频器，所以不需要配置；TIMx_CCMR1.CC1S=2'b01，用于选择输入捕获源
- 配置 TIMx_CCER.CC1P=0 来选择极性（只检测上升沿）
- 配置 TIMx_SMCR.SMS=3'b100，选择定时器为复位模式
- 配置 TIMx_SMCR.TS=3'b101，选择 TI1 作为输入源

- 配置 TIMx_CR1.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.TIF 位被置起，如果使能了中断（TIMx_DIER.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 输入端的同步电路。

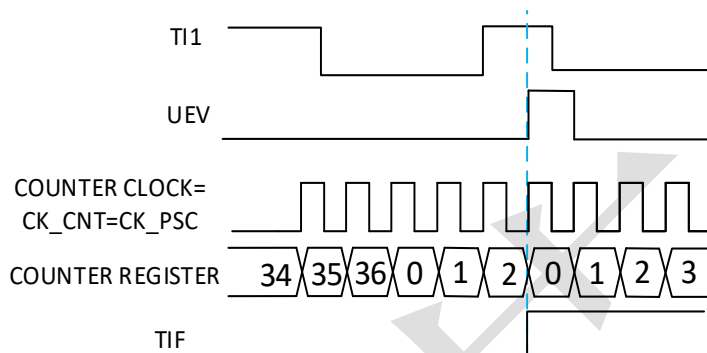


图 15-32 复位模式下的时序图

15.7.3 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 TIMx_CCMR1.IC1F=4'b0000）
- 触发操作中不使用捕获预分频器，所以不需要配置；TIMx_CCMR1.CC1S=2'b01，用于选择输入捕获源
- 配置 TIMx_CCER.CC1P=1 来确定极性（只检测低电平）
- 配置 TIMx_SMCR.SMS=3'b101，选择定时器为门控模式
- 配置 TIMx_SMCR.TS=3'b101，选择 TI1 作为输入源
- 配置 TIMx_CR1.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.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 输入端的同步电路。

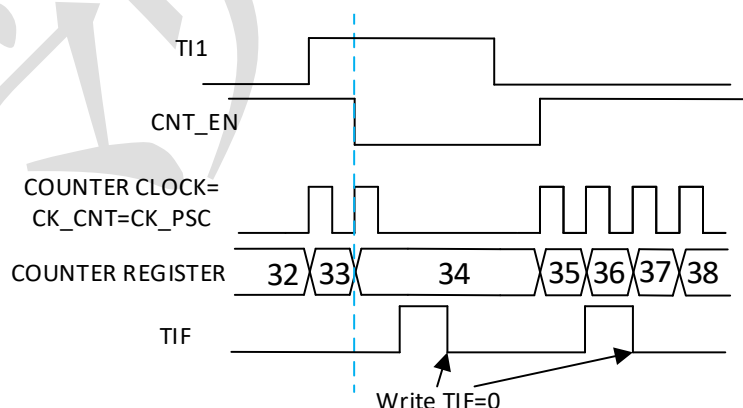


图 15-33 门控模式下的时序图

15.7.4 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.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_SMCR.SMS=4'b1000`。

15.7.5 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.TS != 3'b111`）。

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`：配置 `TIMx_SMCR.ETF=4'b0000` 禁止滤波器，配置 `ETPS=2'b00` 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 `TIMx_CCMR1.IC1F=4'b0000`）
- 触发操作中不使用捕获预分频器，所以不需要配置；`TIMx_CCMR1.CC1S=2'b01`，用于选择输入捕获源
- 配置 `TIMx_CCER.CC1P=0`，选择上升沿做为触发条件
- 配置 `TIMx_SMCR.SMS=3'b110`，选择定时器为触发模式
- 配置 `TIMx_SMCR.TS=3'b101`，选择 `TI1` 作为输入源

当 `TI1` 上出现一个上升沿时，同时 `TIMx_SR.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` 输入端的同步电路。

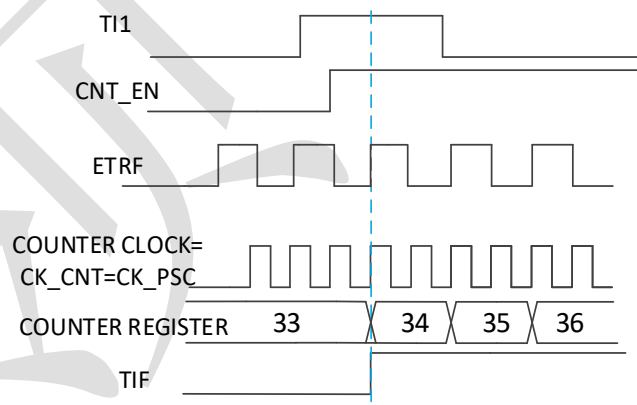


图 15-34 外部时钟模式 2 且触发模式下的时序图

15.7.6 触发延迟和防误触发功能

在触发输入中，用户可根据需求设置触发延迟和防误触发功能。配置 `TIMx_DLAMT.DTG` 可设定触发输入的延迟时间，配置 `TIMx_DLAMT.AMTG` 可设定防误触发的窗口时间，并且配置 `TIMx_DLAMT.AMTDL` 位可选择防误触发窗口的开始时间。如果在防误触发窗口时间内出现新的触发输入，则会被舍弃。如果在触发延迟过程中出现新的触发输入，且不在防误触发窗口时间内，则新的触发输入会被保留，触发延迟过程将重新开始。如图 15-35 所示，在以下的例子中，设定选择 `ITR1`（仅 `TIM15`）的上升沿作为触发输入。

- 配置 `TIM15_SMCR.SMS=3'b100`，选择定时器为复位模式

- 配置 TIM15_SMCR.TS=3'b001，选择 ITR1 作为输入源
- 配置 TIM15_DLAMT.AMTDL=0，选择防误触发窗口时间从延迟的触发输入开始
- 配置 TIM15_DLAMT.DTG=0x7F，TIM1_DLAMT.AMTG=0x5F。

当 ITR1 出现一个上升沿时，内部触发延时计数器开始计数，在触发延迟过程中，出现新的触发输入，将导致内部触发延迟计数器（delay_cnt）重新开始计数，延迟窗口加长，最终经触发延迟后，计数器被清零然后从 0 重新开始计数，同时 TIM1_SR.TIF 位被置起，在防误触发窗口时间内出现新触发输入将会被舍弃。

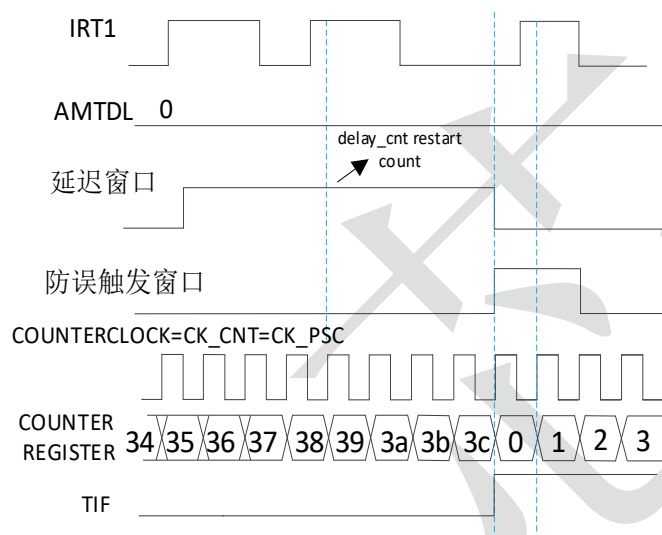


图 15-35 防误触发窗口从延迟的触发输入开始

当选择防误触发窗口从输入触发开始时（配置 TIM15_DLAMT.AMTDL=1），如图 13-36 所示。在防误触发窗口中新的触发输入将会被舍弃，内部延迟计数器继续计数，若在防误触发窗口之外，且仍处于触发延迟过程中再次出现新的触发输入，则内部延迟计数器将重新开始计数，且防误触发功能将会被激活。直到触发延迟结束后，计数器被清零然后从 0 重新开始计数，同时 TIM15_SR.TIF 位被置起。

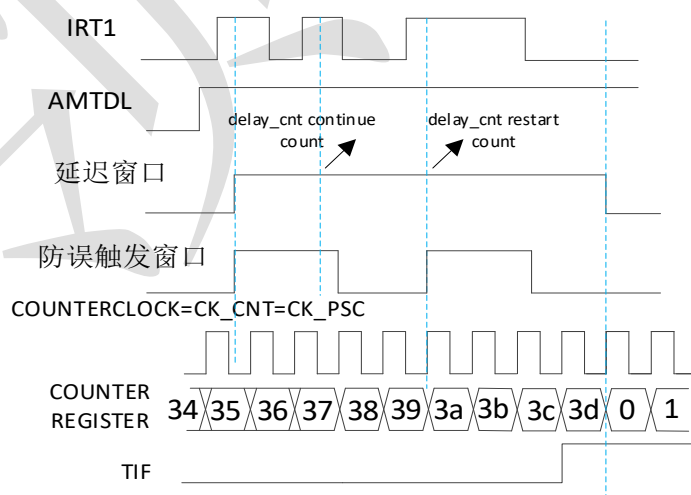


图 15-36 防误触发窗口从触发输入开始

另外，可配置 TIM15_DLAMT.DLS=1，可在触发延时过程中清除 OCiREF，注意在输出比较模式中若选择强制输出模式，则此功能无效。

15.8 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。

TIM15/16/17 输入触发可以来自芯片内部的其他定时器、比较器 ACMP0/1 输出；TIM15/16/17 的输出 TRGO 可触发芯片内部的 ADC、DAC 和其他定时器；TIM15/16/17 的特殊输出 TIMx_trig_oci 可触发芯片内部的 ADC 和其他定时器，它来自相应捕获通道上的 ICiPS 信号或相应比较通道上的 OCi 信号，当 TIMx_trig_oci 来自 OCi 信号时，它可以选择刹车控制前或刹车控制后的 OCi 信号，具体详见 TIMx_BDTR 寄存器 BKTRIG 位说明。

表 15-3 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	tim2_trgo	-
ITR2	内部触发 2	tim16_trig_oc1	-
ITR3	内部触发 3	tim17_trig_oc1	-
外部 Timer 输入 TI			
TI1	外部 Timer 输入 1	tim15_gpio_ti1/adc_awd[0]/adc_awd[1]/ ACMP0_out/ACMP1_out/tim1_trig_oc1/ tim1_trig_oc5/tim2_trig_oc1	SysCtrl_EDU_CFG3
TI2	外部 Timer 输入 2	tim15_gpio_ti1/adc_awd[0]/adc_awd[1]/ ACMP0_out/ACMP1_out/tim1_trig_oc1/ tim1_trig_oc5/tim2_trig_oc1	
外部触发输入 ETR			
ETR	外部触发	tim15_gpio_etr/tim2_trgo/ ACMP0_out/ACMP1_out/tim1_trig_oc1/ tim1_trig_oc5/tim16_trig_oc1	SysCtrl_EDU_CFG3

表 15-4 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[0]/ ACMP0_out/ACMP1_out/tim2_trgo/tim1 _trgo/ tim1_trig_oc5/tim15_trgo	SysCtrl_EDU_CFG3
TI2	外部 Timer 输入 2	tim16_gpio_ti2/adc_awd[1]/ ACMP0_out/ACMP1_out/tim2_trgo/tim1 _trgo/ tim1_trig_oc5/tim15_trgo	SysCtrl_EDU_CFG3

外部触发输入 ETR			
ETR	外部触发	tim16_gpio_etr/tim2_trgo/ ACMP0_out/ACMP1_out/ tim1_trgo/tim1_trig_oc5/tim15_trgo	SysCtrl_EDU_CFG3

表 15-5 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[1]/ ACMP0_out/ACMP1_out/tim2_trgo/tim1_trgo/ tim1_trig_oc5/tim15_trgo	SysCtrl_EDU_CFG3
外部触发输入 ETR			
ETR	外部触发	tim17_gpio_etr/tim2_trgo/ ACMP0_out/ACMP1_out/ tim1_trgo/tim1_trig_oc5/tim15_trgo	SysCtrl_EDU_CFG3

15.9 TIM15/16/17 中断

TIM15 有 6 个中断请求源，共同映射在 TIM15 全局中断：

- 刹车中断
- 更新事件中断
- 触发中断
- COM 事件中断
- 输入捕捉/输出比较 1 中断
- 输入捕捉/输出比较 2 中断

TIM16 有 6 个中断请求源，共同映射在 TIM16 全局中断：

- 刹车中断
- 更新事件中断
- 触发中断
- COM 事件中断
- 输入捕捉/输出比较 1 中断
- 输入捕捉/输出比较 2 中断

TIM17 有 5 个中断请求源，共同映射在 TIM17 全局中断：

- 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

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、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。

15.10 TIM15/16/17 寄存器描述

表 15-6 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	-	-
DLAMT	触发延迟和防误触发寄存器	R/W	0x0001_0000	0x4001_B07C	0x4001_C07C	0x4001_E07C
SysCtrl_EDU_CFG3	TIM15/16/17 互联配置寄存器 3	R/W	0x0000_0000	0x4800_702C	0x4800_702C	0x4800_702C
SysCtrl_EDU_CFG5	TIM17 互联配置寄存器 5	R/W	0x0000_0000	-	-	0x4800_7034
SysCtrl_EDU_CFG6	TIM15 互联配置寄存器 6	R/W	0x0000_0000	0x4800_7038	-	-
SysCtrl_EDU_CFG7	TIM16 互联配置寄存器 7	R/W	0x0000_0000	-	0x4800_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）。

15.10.1 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 TIM15/16）
		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)
	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。
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。

15.10.2 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.															
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 31:12	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 TIM15/16)
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: CCI _E 、CCI _{NE} 、CCI _P 、CCI _{NP} 和 OC _{IM} 位是预装载的；设置该位后，它们只在设置了 COM 事件（COM _G 位置 1 或 TRGI 上升沿，由 CCUS 位控制）发生时后被更新
	注： 该位只对拥有互补输出的通道有效。

15.10.3 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（TIM _x _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（TIM _x _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 _{SAMPLING} =f _{DTS} ，无滤波器（默认）
	0001: 采样频率 f _{SAMPLING} =f _{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 TIM15/16)
	111: 外部触发输入 (ETRF)
	<i>注: 这些位只能在未用到 (如 $\text{SMS}=000$) 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 $\text{CEN}=1$, 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。(仅用于 TIM15/16)
	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。(仅用于 TIM15/16)
	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。(仅用于 TIM15/16)
	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。
	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。

15.10.4 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 TIM15/16)
	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 (仅用于 TIM15)
	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 TIM15/16)
	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

15.10.5 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.				CAP2P	CAP1P	Res.									
				r	r										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.~					CC2OF	CC1OF	B2IF	B1F	TIF	COMIF	Res.		CC2IF	CC1IF	UIF
					r	r	rw	rw	rw	rw			rw	rw	rw

Bits 31:28	保留, 必须保持复位值
Bit 27	CAP2P : 通道 2 捕获边沿标志, 参考 CAP1P 位 (仅 TIM15/16)
Bit 26	CAP1P : 通道 1 捕获边沿标志
	0: 读取 TIMx_CCR1 时, 捕获通道 1 的最近一次边沿为上升沿
	1: 读取 TIMx_CCR1 时, 捕获通道 1 的最近一次边沿为下降沿
Bits 25:11	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 TIM15/16)
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	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
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 TIM15/16）
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
	<i>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。</i>
	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

15.10.6 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
	注 : 该位只对拥有互补输出的通道有效。
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 TIM15/16）
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

15.10.7 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 TIM15/16)
Bit	30	OC2TUE: 输出比较 2 修调更新使能, 参考 OC1TUE (仅用于 TIM15/16)
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 TIM15/16)
Bit	11	OC2PE: 输出比较 2 预装载使能, 参考 OC1PE (仅用于 TIM15/16)
Bit	10	OC2FE: 输出比较 2 快速使能, 参考 OC1FE (仅用于 TIM15/16)
Bits	9:8	CC2S[1:0]: 捕获/比较 2 选择 (仅用于 TIM15/16)
		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) (仅用于 TIM15)
		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 15.5.5 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 TIM15/16)
	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 TIM15/16)
Bits 11:10	IC2PSC[1:0]: 输入捕获 2 预分频器, 参考 IC1PSC (仅用于 TIM15/16)
Bits 9:8	CC2S[1:0]: 捕获/比较 2 选择 (仅用于 TIM15/16)
	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) (仅用于 TIM15)
	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 TIM15/16）
	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。

15.10.8 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 TIM15/16）
Bit 6	CC2NE: 捕获/比较 2 互补输出使能，参考 CC1NE（仅用于 TIM15/16）
Bit 5	CC2P: 捕获/比较 2 极性，参考 CC1P（仅用于 TIM15/16）
Bit 4	CC2E: 捕获/比较 2 使能，参考 CC1E（仅用于 TIM15/16）
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 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。
	<i>注:</i> 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, 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
	<i>注 1:</i> 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。
	<i>注 2:</i> 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
	<i>注:</i> 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 15-7 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	IO 输出禁止 (TIMx 管脚 IO 输出为高阻态) OCi=0, OCiN=0	
		0	0	1	IO 输出禁止 (TIMx 管脚 IO 输出为高阻态), OCi=0	OCiREF + 极性, OCiN=OCiREF xor CCiNP
		0	1	0	OCiREF + 极性, OCi=OCiREF xor CCiP	IO 输出禁止 (TIMx 管脚 IO 输出为高阻态), OCiN=0
		0	1	1	OCiREF + 极性 + 死区	OCiREF 反相 + 极性 + 死区
		1	0	0	IO 输出禁止 (TIMx 管脚 IO 输出为高阻态) OCi=CCiP, OCiN=CCiNP	
		1	0	1	关闭状态 (TIMx 输出无效电平), OCi=CCiP	OCiREF + 极性, OCiN=OCiREF xor CCiNP
		1	1	0	OCiREF + 极性, OCi=OCiREF xor CCiP	关闭状态 (TIMx 输出无效电平), OCiN=CCiNP
		1	1	1	OCiREF + 极性 + 死区	OCiREF 反相 + 极性 + 死区
0	0	X	0	0	IO 输出禁止 (TIMx 管脚 IO 输出为高阻态) OCi=CCiP, OCiN=CCiNP	
	0		0	1	首先处于 IO 输出禁止状态 (TIMx 管脚 IO 输出为高阻态), OCi=CCiP, OCiN=CCiNP, 死区时钟存在且经过死区时间后, 如果 CCiP≠OISi 且 CCiNP≠OISiN, 则继续输出无效电平, 即 OCi=CCiP, OCiN=CCiNP; 否则输出空闲电平, 即 OCi=OISi, OCiN=OISiN	
	0		1	0		
	0		1	1		
	1		0	0	IO 输出禁止 (TIMx 管脚 IO 输出为高阻态) OCi=CCiP, OCiN=CCiNP	
	1		1	0	首先处于关闭状态 (TIMx 输出无效电平), OCi=CCiP, OCiN=CCiNP, 死区时钟存在且经过死区时间后, 如果 CCiP≠OISi 且 CCiNP≠OISiN, 则继续输出无效电平, 即 OCi=CCiP, OCiN=CCiNP; 否则输出空闲电平, 即 OCi=OISi, OCiN=OISiN	
	1		0	1		
	1		1	1		

注: 连接 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 寄存器配置。

15.10.9 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
-----------	------------------

15.10.10 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。

15.10.11 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 (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。

15.10.12 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

15.10.13 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

15.10.14 CCR2 捕获/比较寄存器 2 (TIM15/16_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 TIM15/16)
	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_CCR2 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_CCR2 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_CCR2 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

15.10.15 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.				BKTRIG	AOE2	BK2P	BK2E	Res.							
				rw	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	31:28	保留，必须保持复位值
Bit	27	BKTRIG: TIMx_trig_oci 的触发源选择
		TIMx_BDTR.BKTRIG 位选择连接 ADC/ACMP 的触发信号是否经过刹车控制。若设置 TIMx_BDTR.BKTRIG=1'b1，则触发 ADC/ACMP 的 TIMx_trig_oci 为刹车处理前的信号，此时若刹车输入信号有效，TIMx 会进入刹车状态，但 ADC/ACMP 仍可被触发。反之，设置 TIMx_BDTR.BKTRIG=1'b0，则触发 ADC/ACMP 的 TIMx_trig_oci 为刹车处理后的信号，此时若刹车输入信号有效，TIMx 会进入刹车状态，ADC/ACMP 不会被触发。
		0: 刹车控制处理后的 OCi (默认)
		1: 刹车控制处理前的 OCi
		注: 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	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 TIM15/16_BKIN2 输入无效)
		注: 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: TIM15/16_BKIN2 输入极性
		0: TIM15/16_BKIN2 输入低电平有效 (默认)
		1: TIM15/16_BKIN2 输入高电平有效
		注: 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: TIM15/16_BKIN2 功能使能
		0: TIM15/16_BKIN2 功能禁止 (默认)
		1: TIM15/16_BKIN2 功能使能
		注: 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 OCi 和 OCiN 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 CCiE 位)，则使能 OCi 和 OCiN 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 TIMx_BKIN 输入无效)
		注: 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: TIMx_BKIN 输入极性
		0: TIMx_BKIN 输入低电平有效 (默认)
		1: TIMx_BKIN 输入高电平有效

	注： 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： TIMx_BKIN 功能使能
	0： TIMx_BKIN 功能禁止（默认）
	1： TIMx_BKIN 功能使能
	注： 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 15-7
	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
	注： 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 15-7
	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
	注： 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、BK2E、BKP、BK2P、AOE、AOE2、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）
	注： 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 15.875 μs ，步长时间为 125 ns
	DTG[7:0] = 80H 到 BFH，16 μs 到 31.75 μs ，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
	注： 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。

15.10.16 DLAMT 触发延迟和防误触发寄存器（TIM15/16/17_DLAMT）

地址偏移：0x7C

复位值：0x0001_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
AMTG[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
	<i>注：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。</i>
Bit 16	AMTDL: 防误触发窗口开始时间
	0: 从延迟的触发输入开始
	1: 从触发输入开始（默认）
	<i>注：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。</i>
Bits 15:8	AMTG[7:0]: 防误触发窗口时间设置。在防误触发窗口的时间内如果出现新的触发输入，则会舍弃当前触发输入，实现防误触发功能。
	该位定义了防误触发窗口的时间，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 $t_{amtg} = 2 \times t_{DTS}$
	AMTG[7:5]=110 => AMT=(32+AMTG[4:0]) $\times t_{amtg}$ ，其中 $t_{amtg} = 8 \times t_{DTS}$
	AMTG[7:5]=111 => AMT=(32+AMTG[4:0]) $\times t_{amtg}$ ，其中 $t_{amtg} = 16 \times t_{DTS}$
	举例，如果 $t_{DTS} = 125 \text{ ns}$ （8 MHz），可能的防误触发窗口时间为：
	AMTG[7:0] = 0 到 7FH，0 到 15.875 μs ，步长时间为 125 ns
	AMTG[7:0] = 80H 到 BFH，16 μs 到 31.75 μs ，步长时间为 250 ns
	AMTG[7:0] = C0H 到 DFH，32 μs 到 63 μs ，步长时间为 1 μs
	AMTG[7:0] = E0H 到 FFH，64 μs 到 126 μs ，步长时间为 2 μs
	<i>注：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。</i>
Bits 7:0	DLG[7:0]: 触发延迟窗口时间设置，如果在触发延迟过程中有新的触发输入，且不在防误触发窗口时间内，则会把新的触发保留住，触发延迟过程将重新进行。
	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]) $\times t_{dlg}$ ，其中 $t_{dlg} = 2 \times t_{DTS}$
	DLG[7:5]=110 => DLT=(32+DLG[4:0]) $\times t_{dlg}$ ，其中 $t_{dlg} = 8 \times t_{DTS}$
	DLG[7:5]=111 => DLT=(32+DLG[4:0]) $\times t_{dlg}$ ，其中 $t_{dlg} = 16 \times t_{DTS}$
	举例，如果 $t_{DTS} = 125 \text{ ns}$ （8 MHz），可能的触发延迟窗口时间为：
	DLG[7:0] = 0 到 7FH，0 到 15.875 μs ，步长时间为 125 ns
	DLG[7:0] = 80H 到 BFH，16 μs 到 31.75 μs ，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
	<i>注：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。</i>

15.10.17 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 连续传输长度
	定义了 DMA 在连续模式下的传输长度（当对 TIMx_DMAR 寄存器的地址进行读或写时，TIMx 进行一次连续传输），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
	...
	11111: 32 字节
Bits 7:5	DSEL[2:0]: DMA 连续传输请求选择，默认为 000
	定义了哪个 TIMx DMA 请求使用连续传输，其它的未选中的则使用单次传输
	000: 所有 DMA 请求都使用单次传输
	001: TIMx_CH1/TIMx_CH2/TIMx_UP/TIMx_TRIG/TIMx_COM 使用连续传输
	110 至 111: 保留
Bits 4:0	DBA[4:0]: DMA 基地址
	该位定义了 DMA 在连续模式下的基地址（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。

15.10.18 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。

15.10.19 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。

15.10.20 CCTR2 比较修调寄存器 2 (TIM15/16_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 TIM15/16)

	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。
--	--

15.10.21 TIM15/16/17 互联配置寄存器 3（SysCtrl_EDU_CFG3）

SysCtrl_EDU_CFG3 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 0x05FA659A 到 SysCtrl_KEY 密钥寄存器，且在随后的 64 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。

地址：0x4800_702C

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_TI2S[2:0]		
	rw				rw				rw				rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TIM16_TI1S[2:0]			Res.	TIM15_ETRS[2:0]			Res.	TIM15_TI2S[2:0]			Res.	TIM15_TI1S[2:0]		
	rw				rw				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: 外部 GPIO 输入，参考相关 I/O 复用说明（默认）
		001: TIM2_TRGO
		010: ACMP0 输出
		011: ACMP1 输出
		100: 保留
		101: TIM1_TRGO
		110: TIM1_TRIG_OC5
		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 窗口看门狗 1
		010: ACMP0 输出
		011: ACMP1 输出
		100: TIM2_TRGO
		101: TIM1_TRGO
		110: TIM1_TRIG_OC5
		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: 外部 GPIO 输入，参考相关 I/O 复用说明（默认）

	001: TIM2_TRGO
	010: ACMP0 输出
	011: ACMP1 输出
	100: 保留
	101: TIM1_TRGO
	110: TIM1_TRIG_OC5
	111: TIM15_TRGO
Bit 19	保留, 必须保持复位值
Bits 18:16	TIM16_TI2S[2:0]: TIM16 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 窗口看门狗 1
	010: ACMP0 输出
	011: ACMP1 输出
	100: TIM2_TRGO
	101: TIM1_TRGO
	110: TIM1_TRIG_OC5
	111: TIM15_TRGO
Bit 15	保留, 必须保持复位值
Bits 14:12	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 窗口看门狗 0
	010: ACMP0 输出
	011: ACMP1 输出
	100: TIM2_TRGO
	101: TIM1_TRGO
	110: TIM1_TRIG_OC5
	111: TIM15_TRGO
Bit 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: 外部 GPIO 输入, 参考相关 I/O 复用说明 (默认)
	001: TIM2_TRGO
	010: ACMP0 输出
	011: ACMP1 输出
	100: 保留
	101: TIM1_TRIG_OC1
	110: TIM1_TRIG_OC5
	111: TIM16_TRIG_OC1
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 窗口看门狗 0
	010: MCO
	011: ACMP0 输出

	100: ACMP1 输出
	101: TIM1_TRIG_OC1
	110: TIM1_TRIG_OC5
	111: TIM2_TRIG_OC1
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 窗口看门狗 0
	010: ADC 窗口看门狗 1
	011: ACMP0 输出
	100: ACMP1 输出
	101: TIM1_TRIG_OC1
	110: TIM1_TRIG_OC5
	111: TIM2_TRIG_OC1
注: TIMx_TRIG_OCi 和 TIMx_TRGO 不同, 详见 TIM 章节说明。	

15.10.22 TIM17 互联配置寄存器 5 (SysCtrl_EDU_CFG5)

SysCtrl_EDU_CFG5 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 0x05FA659A 到 SysCtrl_KEY 密钥寄存器, 且在随后的 64 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。

地址: 0x4800_7034

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]							Res.	TIM17_SP[6:0]						
	rw								rw						

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

Bit 31	保留, 必须保持复位值
Bits 30:24	TIM17_BKSE[6:0]: TIM17_BKIN 输入触发源使能控制, 默认值 0000000
	TIM17_BKIN 输入触发源为外部 GPIO 输入/ADC 窗口看门狗 1/ADC 窗口看门狗 0/TIM1_TRGO / TIM2_TRGO / 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
	TIM17_BKIN 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_BKIN 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 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 15:0	保留, 必须保持复位值
注: TIMx_TRIG_OCI 和 TIMx_TRGO 不同, 详见 TIM 章节说明。	

15.10.23 TIM15 互联配置寄存器 6 (SysCtrl_EDU_CFG6)

SysCtrl_EDU_CFG6 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 0x05FA659A 到 SysCtrl_KEY 密钥寄存器, 且在随后的 64 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。

地址: 0x4800_7038

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_BKIN2 输入触发源使能控制, 默认值 000000
	TIM15_BKIN2 输入触发源为 ADC 窗口看门狗 1/ADC 窗口看门狗 0/ TIM16_TRIG_OC1 / TIM17_TRIG_OC1 /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
	TIM15_BKIN2 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_BKIN 输入触发源使能控制, 默认值 0000000
	TIM15_BKIN 输入触发源为外部 GPIO 输入/ADC 窗口看门狗 1/ADC 窗口看门狗 0/ TIM1_TRGO / TIM2_TRGO / 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
	TIM15_BKIN 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_BKIN/TIM15_BKIN2 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
注: TIMx_TRIG_OCI 和 TIMx_TRGO 不同, 详见 TIM 章节说明。	

15.10.24 TIM16 互联配置寄存器 7 (SysCtrl_EDU_CFG7)

SysCtrl_EDU_CFG7 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 0x05FA659A 到 SysCtrl_KEY 密钥寄存器, 且在随后的 64 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。

地址: 0x4800_703C

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_BKIN2 输入触发源使能控制, 默认值 000000
		TIM16_BKIN2 输入触发源为 ADC 窗口看门狗 1/ADC 窗口看门狗 0/ TIM15_TRGO /TIM17_TRIG_OCI/ 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
		TIM16_BKIN2 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_BKIN 输入触发源使能控制, 默认值 0000000
		TIM16_BKIN 输入触发源为外部 GPIO 输入/ADC 窗口看门狗 1/ADC 窗口看门狗 0/ TIM1_TRGO / TIM2_TRGO / 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
		TIM16_BKIN 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_BKIN/TIM16_BKIN2 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
注: TIMx_TRIG_OCI 和 TIMx_TRGO 不同, 详见 TIM 章节说明。		

16. 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 16-1 所示。

表 16-1 定时器间的互联

	TRGI ¹		TI	TRGO	BKIN	BKIN2	OCREF_CLR
	ITR(0/1/2/3)	ETR					
TIM1	tim2_trgo tim15_trgo tim16_trig_oc1 tim17_trig_oc1	tim1_gpio_etr tim2_trgo ACMP0_out ACMP1_out tim15_trig_oc1 tim16_trig_oc1	无	UG 软件更新位 CEN 计数器使能位 更新事件 CC1 比较脉冲 OC1REF 信号 OC2REF 信号 OC3REF 信号 OC4REF 信号	tim1_gpio_bkin adc_awd[1] adc_awd[0] tim15_trgo tim2_trgo ACMP1_out ACMP0_out	adc_awd[1] adc_awd[0] tim16_trig_oc1 tim17_trig_oc1 ACMP1_out ACMP0_out	adc_awd[1] adc_awd[0] tim15_trgo tim2_trgo ACMP1_out ACMP0_out
TIM2	tim1_trgo tim15_trigo tim16_trig_oc1 tim17_trig_oc1	tim2_gpio_etr ACMP0_out ACMP1_out tim1_trig_oc1 tim1_trig_oc5 tim15_trig_oc1 tim16_trig_oc1	tim2_gpio_ti adc_awd[0] adc_awd[1] ACMP0_out ACMP1_out tim1_trig_oc1 tim1_trig_oc5 tim15_trig_oc1 tim16_trig_oc1	UG 软件更新位 CEN 计数器使能位 更新事件 CC1 比较脉冲 OC1REF 信号 OC2REF 信号 OC3REF 信号 OC4REF 信号	tim2_gpio_bkin adc_awd[1] adc_awd[0] tim1_trgo tim15_trigo ACMP1_out ACMP0_out	无	adc_awd[1] adc_awd[0] tim15_trgo tim1_trgo ACMP1_out ACMP0_out
TIM15	tim1_trgo tim2_trgo tim16_trig_oc1 tim17_trig_oc1	tim15_gpio_etr tim2_trgo ACMP0_out ACMP1_out tim1_trig_oc1 tim1_trig_oc5 tim16_trig_oc1	tim15_gpio_ti adc_awd[0] adc_awd[1] ACMP0_out ACMP1_out tim1_trig_oc1 tim1_trig_oc5 tim2_trig_oc1	UG 软件更新位 CEN 计数器使能位 更新事件 CC1 比较脉冲 OC1REF 信号 OC2REF 信号	tim15_gpio_bkin adc_awd[1] adc_awd[0] tim1_trgo tim2_trgo ACMP1_out ACMP0_out	adc_awd[1] adc_awd[0] tim16_trig_oc1 tim17_trig_oc1 ACMP1_out ACMP0_out	无
TIM16	无	tim16_gpio_etr tim2_trgo ACMP0_out ACMP1_out tim1_trgo tim1_trig_oc5 tim15_trgo	tim16_gpio_ti adc_awd[0] adc_awd[1] ACMP0_out ACMP1_out tim2_trgo tim1_trgo tim1_trig_oc5 tim15_trgo	OC1REF 信号	tim16_gpio_bkin adc_awd[1] adc_awd[0] tim1_trgo tim2_trgo ACMP1_out ACMP0_out	adc_awd[1] adc_awd[0] tim15_trig_oc1 tim17_trig_oc1 ACMP1_out ACMP0_out	无

TIM17	无	tim17_gpio_etr tim2_trgo ACMP0_out ACMP1_out tim1_trgo tim1_trig_oc5 tim15_trgo	tim17_gpio_ti adc_awd[1] ACMP0_out ACMP1_out tim2_trgo tim1_trgo tim1_trig_oc5 tim15_trgo	OC1REF 信号	tim17_gpio_bkin adc_awd[1] adc_awd[0] tim1_trgo tim2_trgo ACMP1_out ACMP0_out	无	无
-------	---	---	--	-----------	---	---	---

注 1: TIMx_trig_oci 信号不是直接输出的通道信号，具体详见 TIMx 章节中的说明。

注 2: TRGO 的具体配置详见 TIMx 章节 CR2 寄存器 MMS 位说明。

注 3: ITR 的具体配置详见 TIMx 章节 SMCR 寄存器 TS 位说明。

注 4: 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。



17. 独立看门狗（IWDG）

17.1 IWDG 主要特性

LCM32F06X 内置一个独立看门狗 IWDG，用于解决处理器由于硬件或软件故障所引起的错误。IWDG 由内部独立时钟 RCL 提供时钟，即使系统时钟发生故障也能保持活动状态。它可以被当成看门狗，在发生问题时复位整个系统，或者作为一个自由运行的定时器为应用程序提供超时管理。通过选项字节可以配置软件或硬件启动 IWDG，一旦硬件启动后就不能停止。在调试模式下，IWDG 计数器可以被冻结。

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，且对时间精度要求较低的场合。IWDG 还可以实现计时窗口的功能。IWDG 具有以下特性：

- 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

17.2 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 0xFFF 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。

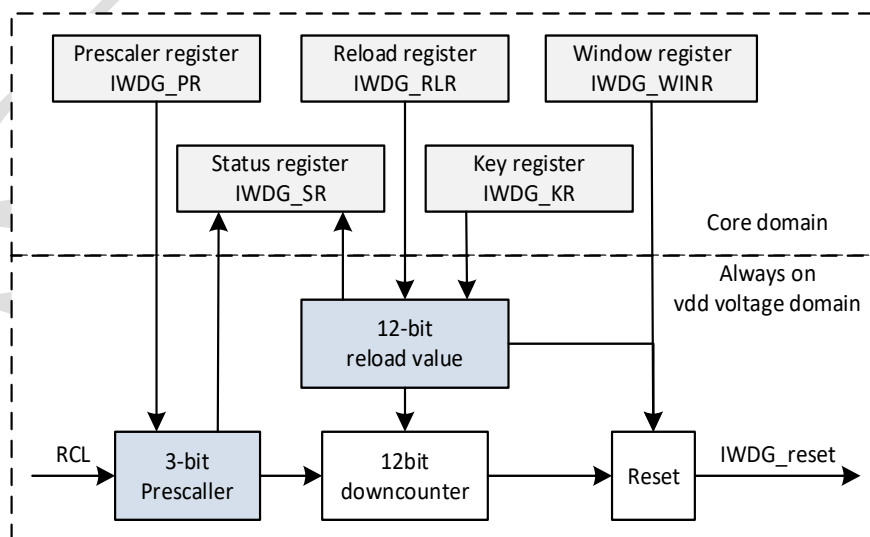


图 17-1 独立看门狗 IWDG 框图

17.2.1 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；

- 通过往 IWDG_RLR 寄存器写入重装载值，设置重装载寄存器；
 - 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；
 - 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；
 - 等待寄存器被更新（查询 IWDG_SR = 0x0000_0000）；
 - 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。

17.2.2 当窗口功能不使能时配置 IWDG

- 通过往 IWDG_KR 寄存器写入 0x0000_CCCC，使能 IWDG；
- 通过往 IWDG_KR 寄存器写入 0x0000_5555，允许寄存器访问；
- 通过往 IWDG_PR 寄存器写入 0 到 7 的值，编程 IWDG 预分频器；
- 通过往 IWDG_RLR 寄存器写入重装载值，设置重装载寄存器；
- 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；
- 等待寄存器被更新（IWDG_SR = 0x0000_0000）；
- 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。

17.2.3 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。

17.2.4 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。

17.2.5 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。

17.3 IWDG 寄存器描述

表 17-1 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_0FFF	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）。

17.3.1 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；如果通过软件启动，系统复位时会停止 IWDG，需要对 KEY 重新写入 0xCCCC 启动 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。

17.3.2 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

17.3.3 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。

17.3.4 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 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。	

17.3.5 IWDG 窗口寄存器 (IWDG_WINR)

地址偏移: 0x10

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.				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。

18. 系统时基定时器（SysTick）

18.1 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）

18.2 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）。

如果系统时钟为 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。

18.3 SysTick 寄存器描述

表 18-1 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）。

18.3.1 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

18.3.2 SysTick 重装载值寄存器（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.								RELOAD[23:16]							
								rw							

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

Bits	31:24	保留，必须保持复位值
Bits	23:0	RELOAD[23:0] : SysTick 定时器的重装载值，计数到 0 时该值将被重装载

18.3.3 SysTick 当前值寄存器（SysTick_VAL）

地址偏移: 0x08

复位值: 0xFFFF_FFFF



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

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

18.3.4 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.						TENMS[23:16]							
r	r							r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TENMS[15:0]															
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，此处是按照 6MHz（HCLK 为 48MHz）作为 SysTick 计数器时钟运行的 10ms 的校准值；如果 HCLK 为其他频率需要软件换算。

19. 钟表定时器（WT）

19.1 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、RCH 的 488 分频
- 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）
- 内部时钟 RCH 经 488 分频

19.2 管脚配置

表 19-1 WT 管脚配置

管脚名称	管脚类型	管脚描述	复用 I/O 口	备注
WT_Buz	O	WT 蜂鸣器输出	PA11/PA4	-
WT_nBuz	O	WT 蜂鸣器反相输出	PA12/PA5	-

19.3 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。

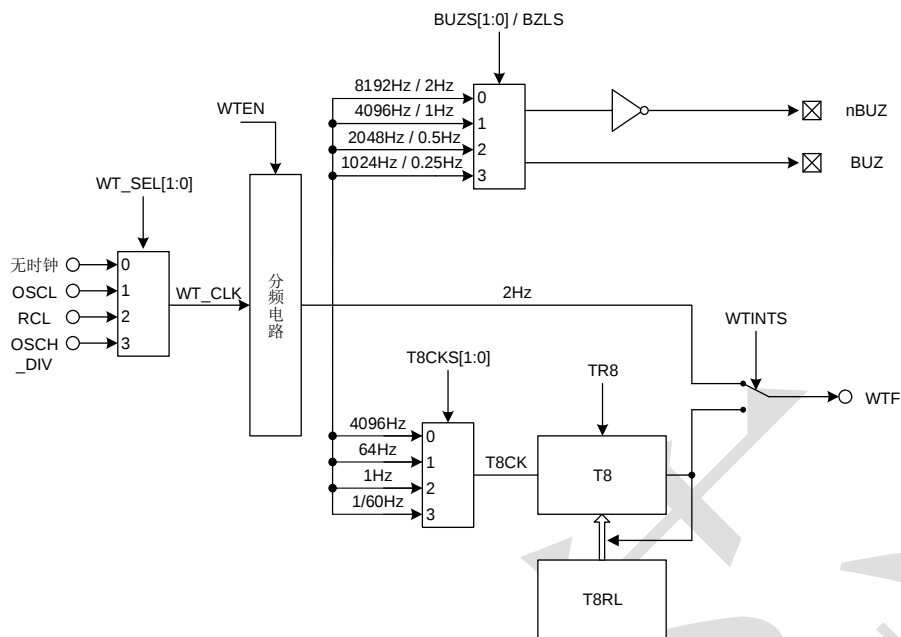


图 19-1 WT 框图

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

19.4 WT 寄存器描述

表 19-2 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）。

19.4.1 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

19.4.2 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)。

20. I2C 接口

LCM32F06X 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。

20.1 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，支持双从地址寻址
- 支持批量传输模式
- 支持 SMBus 功能
- 独立的工作时钟，高达 48MHz
- 带有 8Bytes 的发送和接收 FIFO，支持 DMA 操作
- 可编程的 SDA 保持时间

20.2 管脚配置

表 20-1 I2C 管脚配置

管脚名称	管脚类型	管脚描述	复用 I/O 口	备注
I2C0_SDA	I/O	I2C 数据管脚	PA1/PA10/PA11/PA13/PB7	-
I2C0_SCL	I/O	I2C 时钟管脚	PA3/PA5/PA9/PA12/PA14/PB6	-
I2C0_ALERT	I/O	I2C SMBus Alert 管脚	PA0/PB1/PB8	-

20.3 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：

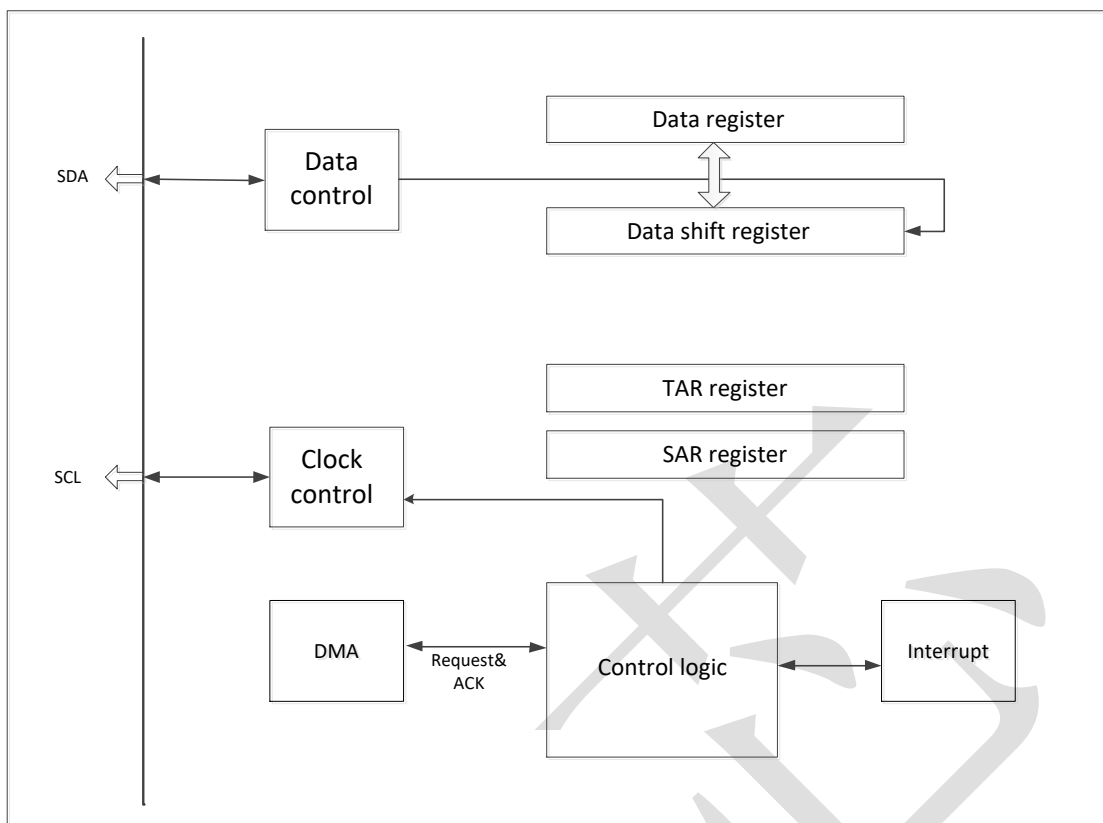


图 20-1 I2C 框图

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。

每个从机可以有两个地址，由用户定义。当主机想跟从机通讯时，主机发送 **START/RESTART** 信号，紧接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**，传输会持续到主机收完最后一字节数据并主机返回 **NACK**，随即主机发送 **STOP** 或发出 **RESTART** 来寻址其他从机。

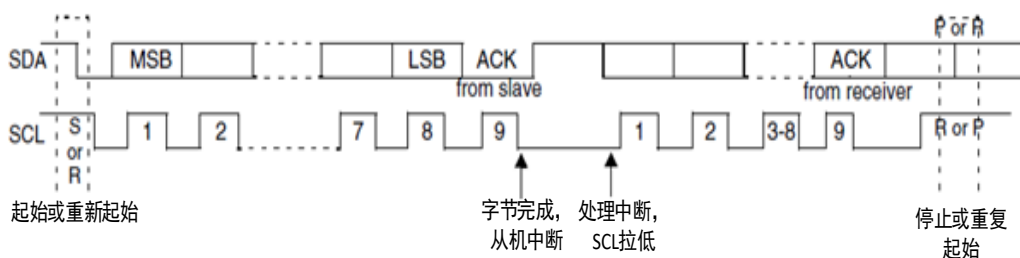


图 20-2 I2C 基本传输协议

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。

20.3.1 操作模式

20.3.1.1 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. 关闭 I2C 使能，配置 `IC_ENABLE.ENABLE = 0`
2. 写 `IC_SAR[9:0]` 设置从机地址；如果需要设置双地址，写 `IC_CON[11]` 为 1，之后写 `IC_SAR[25:16]` 设置从机的第二个地址
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.ENABLE`，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
5. 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
8. 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 FIFO 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 `IC_STATUS.RFNE=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。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

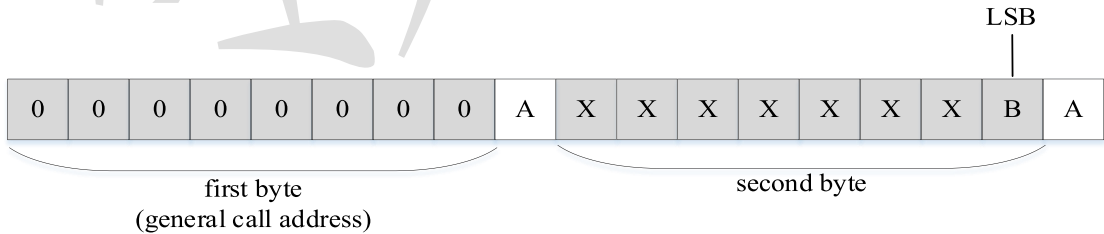
广呼模式用于同时寻址连接到 i2c 总线的每个设备，如果设备不需要广播提供的任何数据，它可以 NACK 来忽略广播地址；如果设备需要来自广播的数据，它就会对广播地址进行 ACK，并作为一个从接收设备。

如果多个设备都发出了响应，主设备实际上并不知道有多少设备回应，每个能够处理此数据的从接收设备都会对第二个和后面的字节 ACK，一般广播地址的含义总是在第二个字节中指定。

- 主机初始化配置：

寄存器配置：IC_TAR 寄存器 bit11 SPECIAL=1，bit10 GC_STR=0；

发送数据格式：



第二个字节的含义有两种情况需要考虑：最低有效位为 0 时和最低有效位为 1 时。

LSB=0:

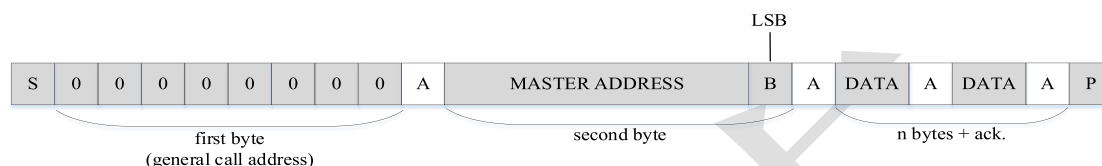
1. 00000110(06h): 复位并通过硬件写从机地址的可编程部分，在接收到这样的 2 个字节时，所有可以响应这个广播地址的设备将复位并进入地址的可编程部分，要采取预防措施来确保设备不会在加上电源电压后将 SDA 或 SCL 拉低，因为这些低电平会阻塞总线。



2. 00000100(04h): 通过硬件编写从地址的可编程部分, 行为如上述, 但设备不复位。
3. 00000010(02h): 软件复位, 这个特性是可选的, 不是所有的设备都会响应这个命令。
4. 00000000(00h): 此代码不允许作为第二个字节使用。

LSB=1:

当 LSB=1 时, 2 个字节的地址序列是一个硬件通用调用。这意味着地址序列是由硬件主设备传输的, 例如键盘扫描器。第二个字节剩下的 7 位包含硬件主设备地址, 这个地址会被一个连接到总线的智能设备 (例如微控制器) 识别, 然后总线接受来自硬件设备的信息。由于硬件主设备实现不知道消息必须要传输到哪个设备, 因此它只能生成这个硬件通用调用和他自己的地址, 然后标识给系统。



- 从机初始化配置:

从机在初始化时要将 IC_ACK_GENERAL_CALL 寄存器的 bit 0 置 1 来保证广呼的应答。

20.3.1.2 主机模式

初始化配置

1. 写入 IC_ENABLE.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. 写入 IC_ENABLE.ENABLE=1 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.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.STOP 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

写入 IC_ENABLE.ENABLE=0, 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.ENABLE=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.ENABLE = 0
6. 读 IC_ENABLE_STATUS.IC_EN 位。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.IC_EN=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.ABORT 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. 配置 IC_ENABLE.ABORT= 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。

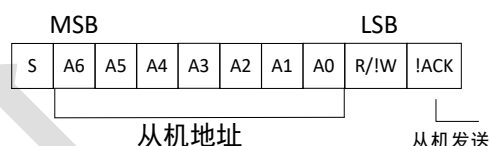


图 20-3 I2C 7 位地址格式

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 20-4)，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]。

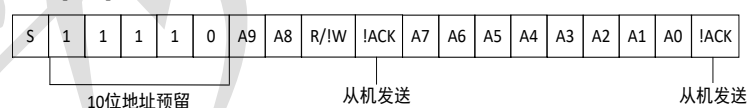


图 20-4 I2C 10 位地址格式

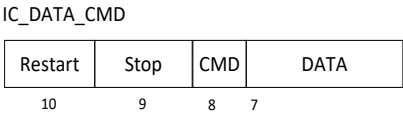
表 20-2 10 位地址的前五位含义表

首字节前五位和 A9、A8	R/W 位	说明
0000 000	0	广播地址，I2C 把数据放入接收 FIFO，并产生广播中断
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

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.STOP 位特意请求时，STOP 条件产生。



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

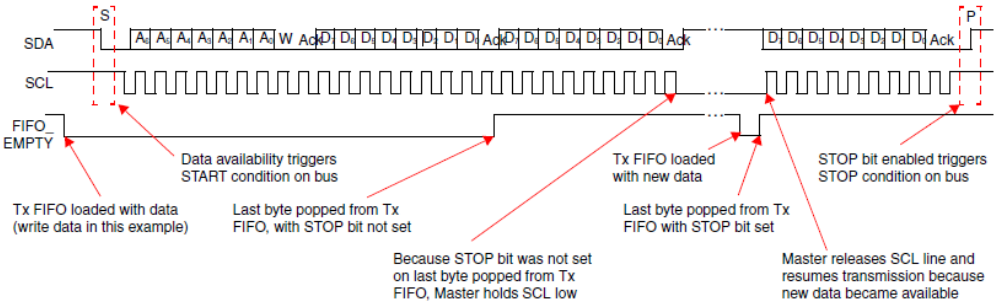


图 20-5 主机发送：发送 FIFO 空/STOP 产生时序图

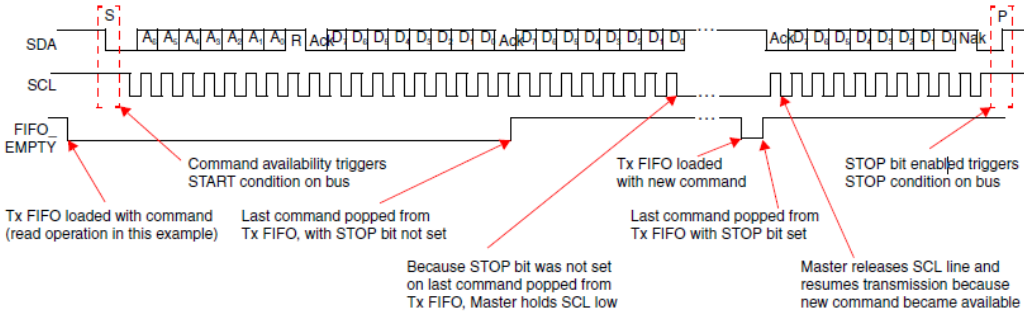


图 20-6 主机接收：发送 FIFO 空/STOP 产生时序图

此外，用户可控制总线上 RESTART 条件的产生。如果 IC_DATA_CMD.RESTART=1，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 20-7 所示。

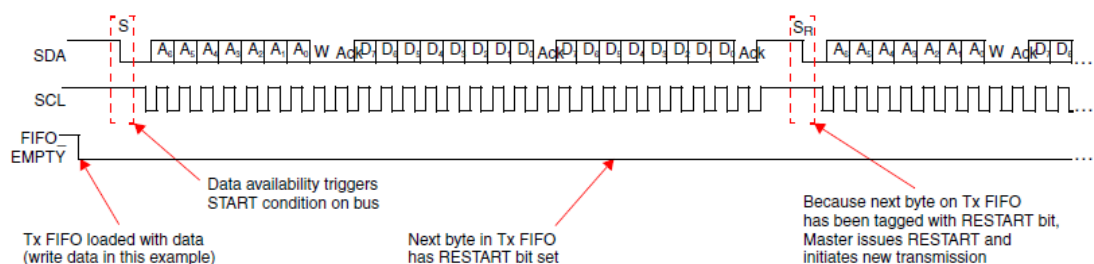


图 20-7 主机发送：IC_DATA_CMD 的 Restart 置位

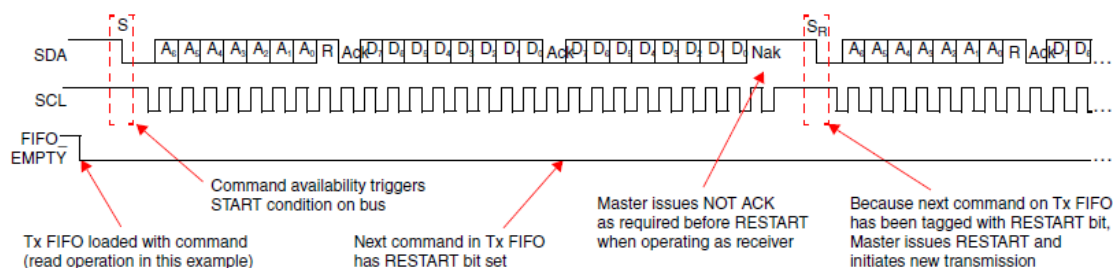


图 20-8 主机接收：IC_DATA_CMD 的 Restart 置位

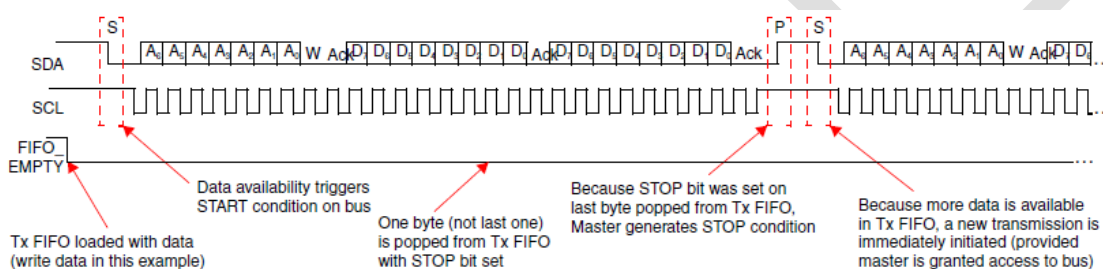


图 20-9 主机发送：IC_DATA_CMD 的 Stop 置位/发送 FIFO 非空

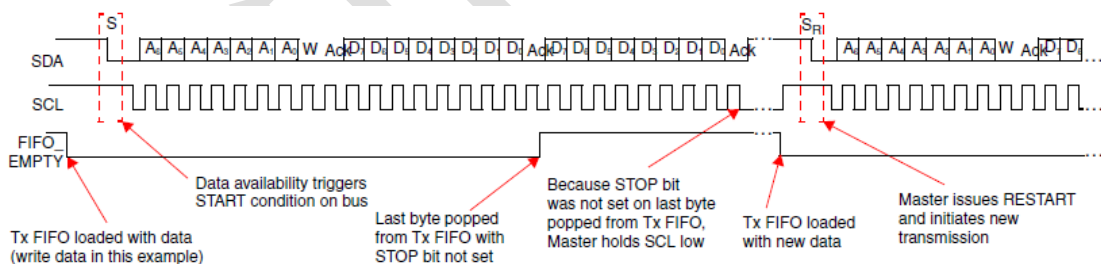


图 20-10 主机发送：FIFO 为空后载入的首字节，Restart 置位

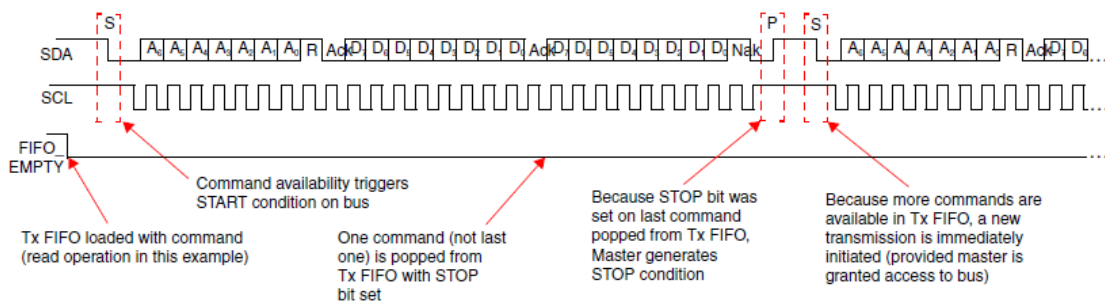


图 20-11 主机接收：IC_DATA_CMD 的 Stop 置位/发送 FIFO 非空

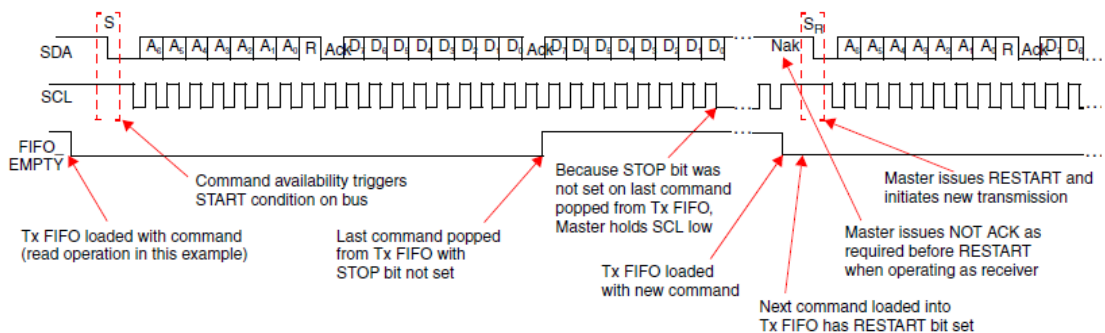


图 20-12 主机接收：允许 TX FIFO 为空/Restart 置位后，加载第一个命令

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一个从机设备，仲裁可在数据阶段进行。

I2C 一旦侦测到仲裁被其他主机获得，I2C 停止产生 SCL。

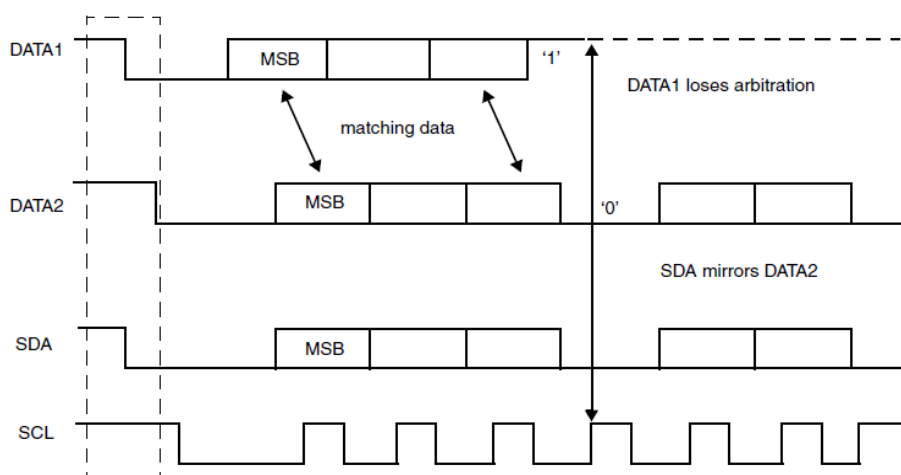


图 20-13 两个主机仲裁的时序

20.3.2 尖峰抑制

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。

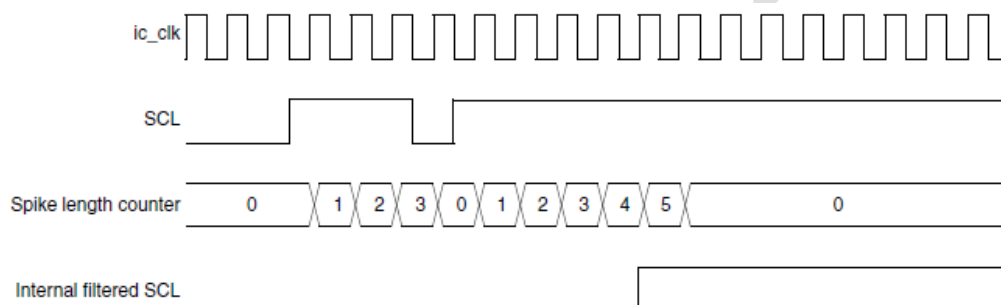


图 20-14 尖峰抑制示例图

图 20-14 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 I2C_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 I2C_CLK 的低频，可以超过峰值长度规范。

20.3.3 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 I2C_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

20.3.4 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。

20.3.4.1 最小高低计数值

当 I2C 作为主机模式时，在发送和接收的过程中：

- IC_SS_SCL_LCNT 和 IC_FS_SCL_LCNT 寄存器值必须大于 IC_FS_SPKLEN+7
- IC_SS_SCL_HCNT 和 IC_FS_SCL_HCNT 寄存器值必须大于 IC_FS_SPKLEN+5

有关 I2C 高计数和低计数的详细信息如下：

- *_LCNT 寄存器的 IC_*_SPKLEN+7 的最小值是由于在 SCL 的下降沿之后 I2C 驱动 SDA 所需的时间。
- *_HCNT 寄存器的 IC_*_SPKLEN+5 的最小值是由于 DW_apb_i2c 在 SCL 的高时段期间采样 SDA 所需的时间。
- I2C 将一个周期添加到编程的 *_LCNT 值以产生 SCL 时钟的低周期。
- I2C 将 IC_*_SPKLEN+7 个周期添加到编程的 *_HCNT 值以产生 SCL 时钟的高周期。

20.3.4.2 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。

以下示例适用于 IC_FS_SPKLEN 编程为 1 的情况。

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 IC_FS_SCL_HCNT = IC_FS_SPKLEN+5 = 1+5 = 6
- 最小 IC_HCNT_FS = IC_FS_SCL_HCNT + IC_FS_SPKLEN + 7 = 6+1+7 = 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：

$$\frac{\text{SCL_PERIOD_FS}}{\text{IC_HCNT_FS} + \text{IC_LCNT_FS}} = \text{IC_CLK_PERIOD}$$

$$\text{IC_LCNT_FS} \times \text{IC_CLK_PERIOD} = \text{MIN_SCL_LOWtime_FS}$$

上述等式产生以下：

$$\text{IC_LCNT_FS} \times \frac{\text{SCL_PERIOD_FS}}{\text{IC_LCNT_FS} + \text{IC_HCNT_FS}} = \text{MIN_SCL_LOWtime_FS}$$

求解 IC_LCNT_FS：

$$\text{IC_LCNT_FS} \times \frac{2.5\mu\text{s}}{\text{IC_LCNT_FS} + 14} = 1.3\mu\text{s}$$

由以上等式得出：

$$\text{IC_LCNT_FS} = \text{roundup}(15.166) = 16$$

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 20-3 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

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 20-2 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

20.3.5 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。

下图显示了一次块传输，其中 DMA 控制器中的块大小为 12，一次连续传输长度设置为 4。在这种情况下，块大小是连续传输长度的倍数。因此，DMA 块传输一系列连续传输。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。

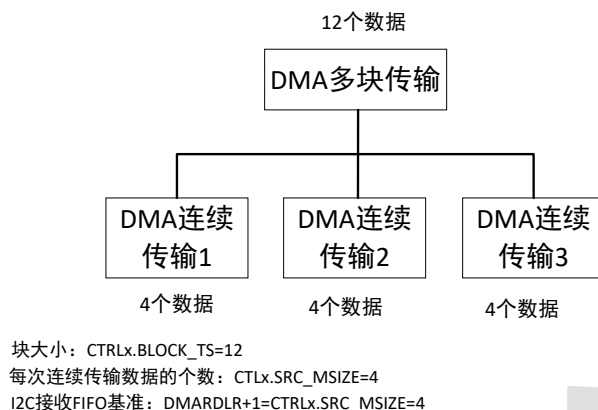


图 20-15 DMA 连续传输分解

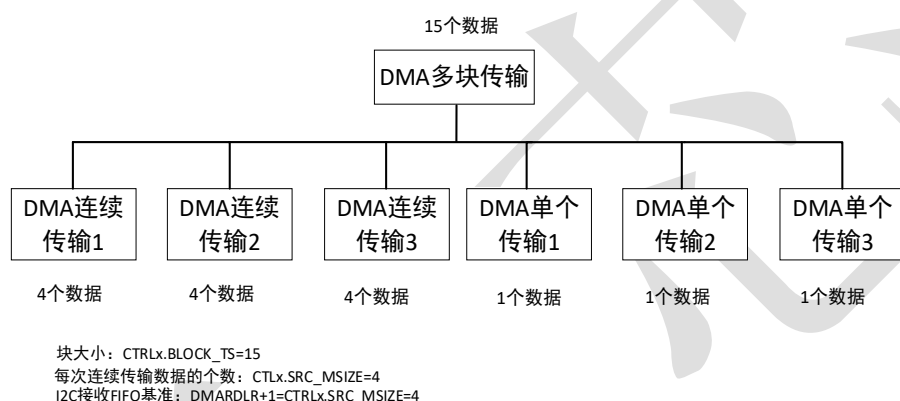


图 20-16 DMA 单个和连续传输分解

如果块大小不是连续传输长度的倍数，在多次连续传输以后，当剩余的数据小于一次连续传输数据大小时，需要多次单次数据传输来完成整个块的传输。

20.3.6 SMBus 模式

系统管理总线（SMBus）是一个双线接口，通过它，各种系统组件芯片和设备可以相互通信，也可以与系统的其他部分通信。它基于 I2C 总线的操作原理。I2C 可以通过 IC_CON 寄存器的 bit[12]选择工作在 SMBus 模式下，它可以连接到 SMBus 总线上与其他设备进行通信。

20.3.6.1 数据传输

总线上传输的每个字节后面都必须有一个确认位。

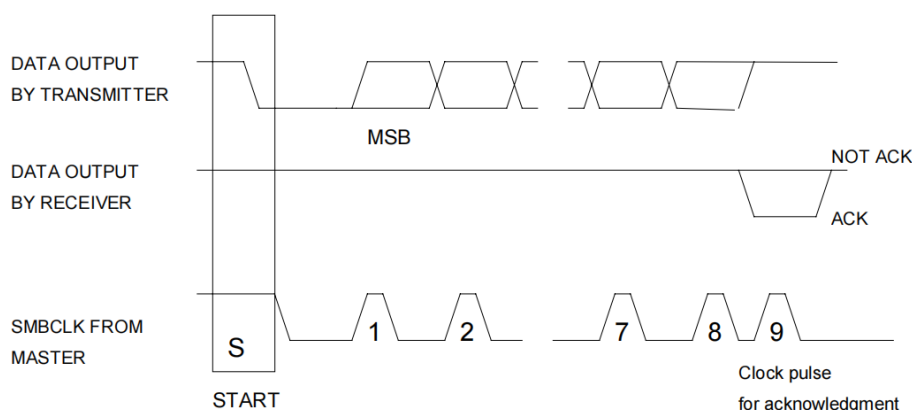


图 20-17 SMBus 数据传输示例

20.3.6.2 超时

设备超时定义和条件

SMBus 协议规定任何设备不得无限期的将 SCL 时钟线保持在低电平，在检测到任何单个时钟保持在低电平的时间超过 $t_{\text{TIMEOUT,MIN}}$ 时将触发超时错误中断，并释放总线。总线上的设备必须重置其通信接口，并且能够在不晚于 $t_{\text{TIMEOUT,MAX}}$ 的时间内接收新的 START 条件。I2C 设备使用该功能时首先要配置 IC_CON 寄存器的 bit[15]和 bit[12]为 1，使能 SMBUS MODE 和 TIMEOUT 超时检测功能，然后根据 I2C 模块的工作时钟 IC_CLK 计算得到 $t_{\text{TIMEOUT,MIN}}$ 的配置值并写入 IC_TIMEOUT_DATA 寄存器。

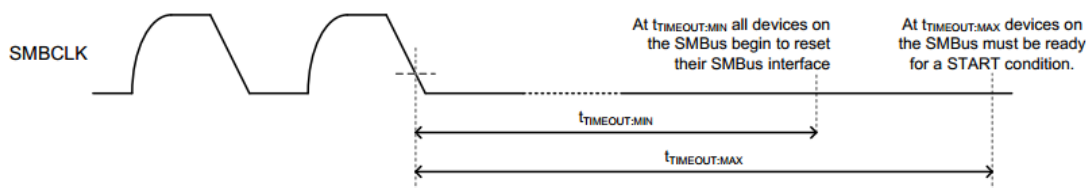


图 20-18 超时时间定义

设备时钟扩展定义和条件

图 20-19 所示时钟扩展间隔 $t_{\text{LOW:MEXT}}$ 和 $t_{\text{LOW:SEXT}}$ 的定义。 $t_{\text{LOW:MEXT}}$ 是允许主设备在从 START 到 ACK、从 ACK 到 ACK 或从 ACK 到 STOP 的消息的每个字节内延长其时钟周期的累计时间。从设备或其他主设备也可能延长时钟，导致组合时钟低时间在给定字节上大于 $t_{\text{LOW:MEXT}}$ 。

$t_{\text{LOW:SEXT}}$ 是允许一个给定的从设备在一个消息中延长时钟周期从初始开始到停止的累积时间。有可能，另一个从设备或主设备也会延长时钟，导致组合时钟低延长时间大于 $t_{\text{LOW:SEXT}}$ 。

I2C 设备使用该功能时首先要配置 IC_CON 寄存器的 bit[14]和 bit[12]为 1，使能 SMBUS MODE 和 TEXT 超时检测功能，然后根据 I2C 模块的工作时钟 IC_CLK 计算得到 $t_{\text{LOW:MEXT}}$ 或 $t_{\text{LOW:SEXT}}$ 的配置值并写入 IC_TIMEOUT_DATA 寄存器。

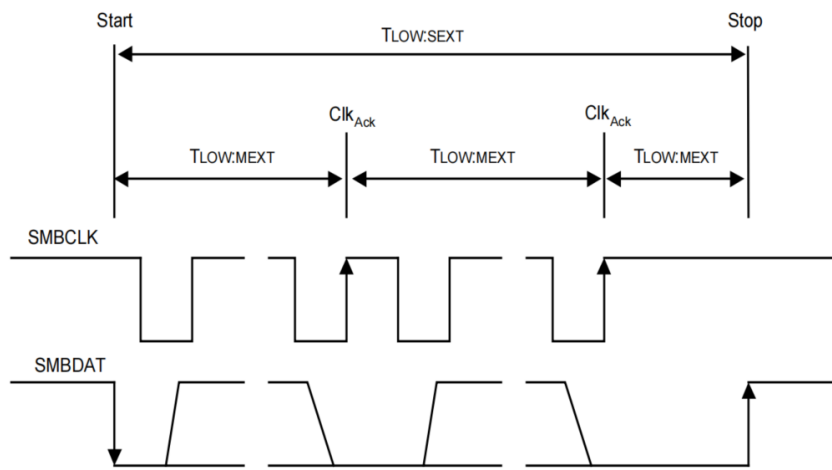


图 20-19 时钟扩展时间定义

表 20-4 SMBus 超时规范

时间	描述	限制		单位
		Min	Max	
t_{TIMEOUT}	检测时钟低超时时间	25	35	ms
$t_{\text{LOW:SEXT}}$	累计时钟低扩展时间（从设备）	-	25	ms
$t_{\text{LOW:MEXT}}$	累计时钟低扩展时间（主设备）	-	10	ms

IC_TIMEOUT_DATA寄存器配置示例

IC_TIMEOUT_DATA 的配置值由以下公式计算得到，CNT 为三种超时条件不同的计数值，由硬件决定， t_{TIMEOUT} 与 $t_{\text{LOW:SEXT}}$ 的 CNT 为 25， $t_{\text{LOW:MEXT}}$ 的 CNT 为 10，三种超时时间由表 20-3 中 SMBus 超时规范决定。

$$IC_TIMEOUT_DATA \times CNT \times \frac{1}{f_{IC_CLK}} = t_{\text{TIMEOUT}} / t_{\text{LOW:SEXT}} / t_{\text{LOW:MEXT}}$$

- 配置 $t_{\text{TIMEOUT,MIN}}=25\text{ms}$

表 20-5 各种 IC_CLK 频率的 IC_TIMEOUT_DATA 配置示例 1

f_{IC_CLK}	IC_TIMEOUT_DATA bits	TIMEOUT_EN bit	$t_{\text{TIMEOUT,MIN}}$
32KHz	0X20	1	$32 \times 25 \times 31.25\mu\text{s} = 25\text{ms}$
8MHz	0X1F40	1	$8000 \times 25 \times 125\text{ns} = 25\text{ms}$
16MHz	0X3E80	1	$16000 \times 25 \times 62.5\text{ns} = 25\text{ms}$
48MHz	0XBB80	1	$48000 \times 25 \times 20.08\text{ns} = 25\text{ms}$

- 配置 $t_{\text{LOW:SEXT}}=25\text{ms}$

表 20-6 各种 IC_CLK 频率的 IC_TIMEOUT_DATA 配置示例 2

f_{IC_CLK}	IC_TIMEOUT_DATA bits	TIMEOUT_EN bit	$t_{\text{TIMEOUT,MIN}}$
32KHz	0X20	1	$32 \times 25 \times 31.25\mu\text{s} = 25\text{ms}$
8MHz	0X1F40	1	$8000 \times 25 \times 125\text{ns} = 25\text{ms}$
16MHz	0X3E80	1	$16000 \times 25 \times 62.5\text{ns} = 25\text{ms}$
48MHz	0XBB80	1	$48000 \times 25 \times 20.08\text{ns} = 25\text{ms}$

- 配置 $t_{\text{LOW:MEXT}}=10\text{ms}$

表 20-7 各种 IC_CLK 频率的 IC_TIMEOUT_DATA 配置示例 3

$f_{\text{IC_CLK}}$	IC_TIMEOUT_DATA bits	TIMEOUT_EN bit	$t_{\text{TIMEOUT,MIN}}$
32KHz	0X20	1	$32*10*31.25\mu\text{s}=10\text{ms}$
8MHz	0X1F40	1	$8000*10*125\text{ns}=10\text{ms}$
16MHz	0X3E80	1	$16000*10*62.5\text{ns}=10\text{ms}$
48MHz	0XBB80	1	$48000*10*20.08\text{ns}=10\text{ms}$

20.3.6.3 错误检测

数据包错误检查机制提高了可靠性和通信稳定性。SMBus 设备对数据包错误检查的实现对于 SMBus 设备是可选的，但对于参与 ARP 过程且仅在 ARP 过程中的设备是必需的。实现数据包错误检查的 SMBus 设备必须能够与其他不实现数据包故障检查机制的设备通信。

数据包错误检查是通过在每次消息传输结束时附加数据包错误代码（PEC）来实现的。PEC 是 CRC-8 错误检查字节，根据所有消息字节（包括地址和读/写位）计算。PEC 由提供最后一个数据字节的设备附加到消息中。

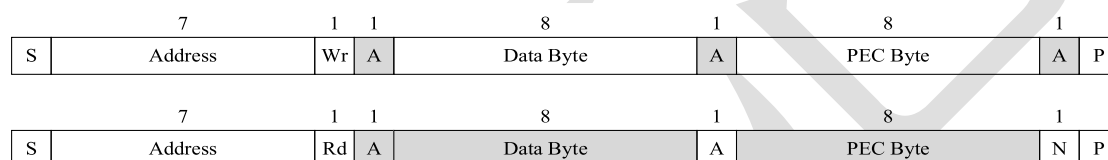


图 20-20 带 PEC 的数据读/写传输示例

I2C 设备使用错误检测功能时首先要配置 IC_CON 寄存器的 bit[13]与 bit[12]为 1，使能 SMBUS MODE 和 PEC 错误检测功能。发送机将要发送的地址信息、读写操作位以及所有数据字节放入内部 CRC 模块计算得到 PEC，在发送过程的最后一个字节将 PEC 发送出去。

接收机在接收带有 PEC 的数据时首先将地址读写信息以及所有的数据字节放入内部 CRC 模块进行计算得到 PEC 计算值，然后由软件将 PEC 计算值写入 IC_PEC_DATA 寄存器。I2C 设备在接收完最后一组 PEC 字节后由硬件将 IC_PEC_DATA 寄存器内的 PEC 计算值与接收到的 PEC 字节进行对比，若对比错误则产生 PEC_ERROR 中断。

20.3.6.4 SMBus Alert

SMBALERT 是一个与 SCL 和 SDA 信号线相同的有线 AND 信号。从属设备可以通过 SMBALERT 向主机发出它想要对话的信号。主机处理中断，并同时通过警报响应地址（ARA）访问所有拉低 SMBALERT 设备。只有拉低了 SMBALERT 的设备才会确认警报响应地址。I2C 从机设备通过将 IC_ENABLE.SMBUS_ALERT 写 1 来驱动 SMBus Alert 线为低电平。主设备执行修改后的接收字节操作。由从属发送设备提供的 7 位设备地址被放置在字节的 7 个最高有效位中。第八位可以是零或一。

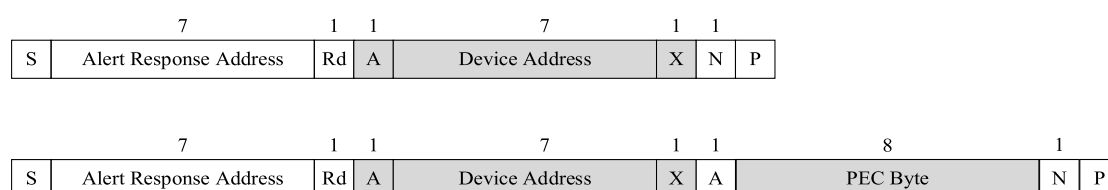


图 20-21 7 位可寻址设备响应 ARA

如果多个设备拉低 SMBALERT，最高优先级（最低地址）设备将在从地址传输期间通过标准仲裁获得

通信权利。确认从地址后，设备必须放开 SMBALERT。如果主机在消息传输完成时仍然看到 SMBALERT 是低电平，那么它知道要再次读取 ARA。

20.3.6.5 地址解析协议（ARP）

SMBus 从属地址冲突可以通过动态地为每个从属设备分配一个新的唯一地址来解决。地址解析协议（ARP）具有以下属性：地址分配使用标准 SMBus 物理层仲裁机制分配设备电源时，分配地址保持不变；通过设备电源损失保留地址；地址分配后不会产生额外的 SMBus 包开销。使用地址分配功能必须使能 SMBUS MODE 和 PEC 错误检测功能。

唯一设备标识符（UDID）

为了提供一种机制来隔离每个设备，以分配地址，每个设备必须实现一个唯一的设备标识符（UDID）。这个 128 位的数字由以下字段组成。

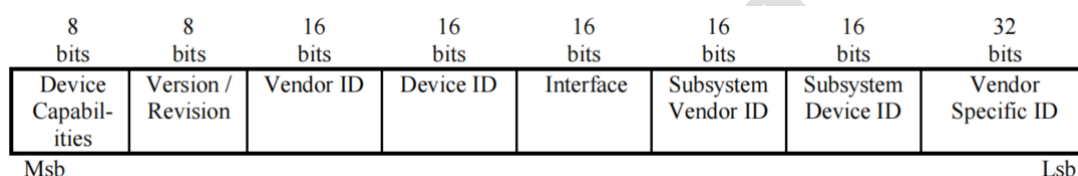


图 20-22 UDID 组成

ARP命令

ARP 控制器可以发出通用命令和定向命令。通用命令针对所有设备，是地址解析过程所必需的。定向命令针对单个设备。ARP 控制器产生的所有数据包都使用数据包错误检查，并以基本格式开始。ARP 命令应由用户通过软件执行。

有关 SMBus 地址解析协议的更多详细信息，请参阅 SMBus 规范（<http://smbus.org>）。

20.4 I2C 中断

I2C 有 19 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_UNDER: 当接收 FIFO 为空时，通过 IC_DATA_CMD 寄存器读 FIFO，该位置 1。
- RX_OVER: 当接收 FIFO 的数据溢出时，该位置 1。
- RX_FULL: 当接收 FIFO 的数据超过 IC_RX_TL 所设定的阈值时，该位置 1。
- TX_OVER: 当发送 FIFO 写满的情况下，软件再次写入数据，该位置 1。
- TX_EMPTY: 当发送 FIFO 的数据不高于 IC_TX_TL 所设定的阈值时，该位置 1。
- RD_REQ: 当 I2C 作为从机，另一主机试图从该从机读取数据时，该位置 1。
- TX_ABRT: 当 I2C 作为发送器，发送过程意外中止时，该位置 1。
- RX_DONE: 当 I2C 作为从机发送时，如果主机不反馈已传送数据，该位置 1。这发生数据传输的最后一个字节，指示传输完毕。
- ACTIVITY: 当 I2C 处于工作状态时，该位置 1。
- STOP_DET: 当检测到 STOP 条件时，该位置 1。
- START_DET: 当检测到 START 或 RESTART 条件时，该位置 1。
- GEN_CALL: 当 I2C 收到且识别 General Call 地址，该位置 1。
- RESTART_DET: 当 I2C 工作在从机模式且已被寻址的情况下，检测到一个 RESTART 条件时，该位置 1。

- MST_ON_HOLD: 当 I2C 主机占有总线时, 该位置 1。
- SMBUS_ALERT: 当总线上某个设备向 SMBALERT 线上产发送低电平信号时, 该位置 1。
- TIMEOUT: 当 I2C 工作在 SMBUS 协议下发生超时错误时, 该位置 1。
- PEC_ERROR: I2C 工作在 SMBUS 协议下开启 PEC 校验功能后, 若产生 PEC 校验错误, 该位置 1。
- ADDR0_MATCH: 当 I2C 工作在从机模式下, IC_SAR0 匹配另一主机的寻址时, 该位置 1。
- ADDR1_MATCH: 当 I2C 工作在从机模式下, 并使能双地址功能后, IC_SAR1 匹配另一主机的寻址时, 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。

20.5 I2C 寄存器描述

表 20-8 I2C 相关寄存器表

名称	说明	读写权限	复位值	字节地址
IC_CON	控制寄存器	R/W	0x0000_007D	0x4001_3000
IC_TAR	I2C 目标地址寄存器	R/W	0x0000_1055	0x4001_3004
IC_SAR	I2C 从机地址寄存器	R/W	0x0000_0055	0x4001_3008
IC_DATA_CMD	I2C 数据命令寄存器	R/W	0x0000_0000	0x4001_3010
IC_SS_SLC_HCNT	I2C 标准模式时钟高字节计数器	R/W	0x0000_011E	0x4001_3014
IC_SS_SLC_LCNT	I2C 标准模式时钟低字节计数器	R/W	0x0000_0150	0x4001_3018
IC_FS_SLC_HCNT	I2C 快速模式时钟高字节计数器	R/W	0x0000_002B	0x4001_301C
IC_FS_SLC_LCNT	I2C 快速模式时钟低字节计数器	R/W	0x0000_005D	0x4001_3020
IC_INTR_STAT	中断状态寄存器	R/W	0x0000_0000	0x4001_302C
IC_INTR_MASK	中断屏蔽寄存器	R/W	0x0000_08FF	0x4001_3030
IC_RAW_INTR_STAT	中断原始状态寄存器	R/W	0x0000_0000	0x4001_3034
IC_RX_TL	接收 FIFO 阈值寄存器	R/W	0x0000_0000	0x4001_3038
IC_TX_TL	发送 FIFO 阈值寄存器	R/W	0x0000_0000	0x4001_303C
IC_CLR_INTR	中断清除寄存器	R/W	0x0000_0000	0x4001_3040
IC_CLR_RX_UNDER	清除 RX_UNDER 中断寄存器	R	0x0000_0000	0x4001_3044
IC_CLR_RX_OVER	清除 RX_OVER 中断寄存器	R	0x0000_0000	0x4001_3048
IC_CLR_TX_OVER	清除 TX_OVER 中断寄存器	R	0x0000_0000	0x4001_304C
IC_CLR_RD_REQ	清除 RD_REQ 中断寄存器	R	0x0000_0000	0x4001_3050
IC_CLR_TX_ABRT	清除 TX_ABRT 中断寄存器	R	0x0000_0000	0x4001_3054
IC_CLR_RX_DONE	清除 RX_DONE 中断寄存器	R	0x0000_0000	0x4001_3058
IC_CLR_ACTIVITY	清除 ACTIVITY 中断寄存器	R	0x0000_0000	0x4001_305C
IC_CLR_STOP_DET	清除 STOP_DET 中断寄存器	R	0x0000_0000	0x4001_3060
IC_CLR_START_DET	清除 START_DET 中断寄存器	R	0x0000_0000	0x4001_3064
IC_CLR_GEN_CALL	清除 GEN_CALL 中断寄存器	R	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	0x0000_0000	0x4001_3070
IC_TXFLR	I2C 发送 FIFO 数据量寄存器	R	0x0000_0000	0x4001_3074
IC_RXFLR	I2C 接收 FIFO 数据量寄存器	R	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	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	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	0x0000_0000	0x4001_30A8
IC_CLR_TIEMOUT	清除 TIMEOUT 寄存器	R	0x0000_0000	0x4001_30AC
IC_CLR_SMBUS_ALERT	清除 SMBUS_ALERT 寄存器	R	0x0000_0000	0x4001_30B0
IC_CLR_PEC_ERROR	清除 PEC_ERROR 寄存器	R	0x0000_0000	0x4001_30B4
IC_TIMEOUT_DATA	TIMEOUT 计时寄存器	R/W	0x0000_5554	0x4001_30B8
IC_PEC_DATA	PEC 数据寄存器	R/W	0x0000_0000	0x4001_30BC
IC_CLR_ADDR0_MATCH	清除 ADDR0_MATCH 寄存器	R	0x0000_0000	0x4001_30C0
IC_CLR_ADDR1_MATCH	清除 ADDR1_MATCH 寄存器	R	0x0000_0000	0x4001_30C4

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）。

20.5.1 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
TIEMO UT_EN	TEXT _EN	PEC_EN	SMB_ MODE	DUAL_A DDR	SDIMA	RFFHC	TXEC	SDIAD	SLVDIS	RESTEN	10ADDR _MAS	10ADDR _SLV	SPEED [1:0]	MST_ MODE	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	rw	rw	rw	rw

Bits	31:16	保留，必须保持复位值
Bit	15	TIMEOUT_EN : SCL 低电平保持时间超时错误检测功能使能（仅在 SMBUS 模式使能下有效）
		0: 超时错误检测功能禁止（默认）
		1: 超时错误检测功能使能
Bit	14	TEXT_EN : 控制器设备在传输每个字节内扩展时钟累计时间超时检测功能使能（仅在 SMBUS 模式使能下有效）
		0: 超时错误检测功能禁止（默认）
		1: 超时错误检测功能使能
Bit	13	PEC_EN : PEC 错误检测功能使能（仅在 SMBUS 模式使能下有效）

	0: PEC 错误检测功能禁止（默认）
	1: PEC 错误检测功能使能
Bit 12	SMB_MODE: SMBUS 模式使能
	0: SMBUS 模式禁止（默认）
	1: SMBUS 模式使能
Bit 11	DUAL_ADDR: 从机的双地址使能
	0: 不启用双地址，作从机时地址只能用 IC_SAR0（默认）
	1: 启用双地址，作从机时 IC_SAR0 和 IC_SAR1 均有效
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 达到满时是否占有总线
	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“重新起始（Restart）”信号
	0: 禁止
	1: 使能（默认）
	当 RESTEN=0，主机不支持以下功能： ● 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_SAR0 和 IC_SAR1 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

20.5.2 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。		

20.5.3 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.						IC_SAR1[9:0]									
						rw									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.						IC_SAR0[9:0]									



	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。当读该寄存器时，DAT 返回 I2C 从总线接收到的数据

20.5.5 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.ENABLE = 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

20.5.6 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.ENABLE = 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

20.5.7 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.ENABLE = 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

20.5.8 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.ENABLE = 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

20.5.9 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.													R_ADDR1_MATCH	R_ADDR0_MATCH	R_PEC_ERROR
													r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R_TIMOUT	R_SMBUS_ALERT	R_MASTER_ON_HOLD	R_RESTART_DETECTED	R_GENERATE_CALL	R_STARDET	R_STOP_DET	R_ACTIVITY	R_RX_DONE	R_TX_ABORT	R_RD_REQ	R_TX_EMPTY	R_TX_OVER	R_RX_FULL	R_RX_OVER	R_RX_UNDER
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits 31:19	保留，必须保持复位值
------------	------------

Bit 18	R_ADDR1_MATCH: 详细描述参见 IC_RAW_INTR_STAT
Bit 17	R_ADDR0_MATCH: 详细描述参见 IC_RAW_INTR_STAT
Bit 16	R_PEC_ERROR: 详细描述参见 IC_RAW_INTR_STAT
Bit 15	R_TIMEOUT: 详细描述参见 IC_RAW_INTR_STAT
Bit 14	R_SMBUS_ALERT: 详细描述参见 IC_RAW_INTR_STAT
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。	

20.5.10 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.													M_ADDR1_MATCH	M_ADDR0_MATCH	M_PEC_ERROR
													r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M_TIMEOUT	M_SMBUS_ALERT	M_MST_ON_HOLD	M_RESTART_DET	M_GEN_CALL	M_START_DET	M_STOP_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	rw	rw

Bits 31:19	保留，必须保持复位值
Bit 18	M_ADDR1_MATCH: 详细描述参见 IC_RAW_INTR_STAT
Bit 17	M_ADDR0_MATCH: 详细描述参见 IC_RAW_INTR_STAT
Bit 16	M_PEC_ERROR: 详细描述参见 IC_RAW_INTR_STAT
Bit 15	M_TIMEOUT: 详细描述参见 IC_RAW_INTR_STAT
Bit 14	M_SMBUS_ALERT: 详细描述参见 IC_RAW_INTR_STAT
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

20.5.11 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.													ADDR1_MATCH	ADDR0_MATCH	PEC_ERROR
													r	r	r

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

Bits	31:19	保留, 必须保持复位值
Bit	18	ADDR1_MATCH: I2C 工作在从机模式下, 并使能双地址功能后, IC_SAR1 匹配另一主机的寻址时, 该位置 1
Bit	17	ADDR0_MATCH: 当 I2C 工作在从机模式下, 地址 IC_SAR0 匹配另一主机的寻址时, 该位置 1
Bit	16	PEC_ERROR: I2C 工作在 SMBUS 模式下开启 PEC 错误检测功能后, 若产生 PEC 校验错误, 该位置 1
Bit	15	TIMEOUT: 当 I2C 工作在 SMBUS 模式下发生超时错误时该位置 1
Bit	14	SMBUS_ALERT: 当总线上某个设备向 SMBALERT 线上产发送低电平信号时, 该位置 1 (仅在 SMBUS 模式下有效)
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 禁止或读 CLR_GEN_CALL 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.SDIAD=1，STOP_DET 中断只发生在从机地址匹配后。注意：一个广播地址，当 IC_CON.SDIAD=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.SDIAD = 0，无论寻址匹配与否，产生 STOP_DET 中断。
	主机模式：
	如果 IC_CON.SDIMA=1，产生 STOP_DET 中断仅在主机模式工作状态下有效。 如果 IC_CON.SDIMA=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 TX_ABRT=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.TXEC
	当 IC_CON.TXEC=0，当发送 FIFO 不高于 IC_TX_TL 所设定的阈值时，该位置 1。
	当 IC_CON.TXEC=1，当发送 FIFO 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 FIFO 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 FIFO 超过 IC_RX_TL 所设定的阈值时，该位置为 1。当 FIFO 低于阈值时，该位自动清 0。一旦 IC_ENABLE.ENABLE 置 0，该位清 0。
Bit 1	RX_OVER ：I2C 的接收 FIFO 满且从外部 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.ENABLE 置 0，该位清 0。
Bit 0	RX_UNDER ：当接收 FIFO 为空时，CPU 通过 IC_DATA_CMD 寄存器读 FIFO，该位置 1。一旦 IC_ENABLE.ENABLE 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>	

20.5.12 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。

20.5.13 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。

20.5.14 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。		

20.5.15 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

20.5.16 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

20.5.17 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



20.5.18 I2C 清除 RD_REQ 中断寄存器 (IC_CLR_RD_REQ)

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.															CLR_RD_REQ
															r

Bits	31:1	保留, 必须保持复位值
Bit	0	CLR_RD_REQ: 读该寄存器清除 RD_REQ 中断

20.5.19 I2C 清除 TX_ABRT 中断寄存器 (IC_CLR_TX_ABRT)

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.															CLR_TX_ABRT
															r

Bits	31:1	保留, 必须保持复位值
Bit	0	CLR_TX_ABRT: 读该寄存器清除 TX_ABRT 中断和 IC_TX_ABRT_SOURCE 寄存器, 允许把更多数据写入 TX FIFO 中, 具体请参考 IC_TX_ABRT_SOURCE 寄存器。

20.5.20 I2C 清除 RX_DONE 中断寄存器 (IC_CLR_RX_DONE)

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.															

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

Bits	31:1	保留, 必须保持复位值
Bit	0	CLR_RX_DONE: 读该寄存器清除 RX_DONE 中断

20.5.21 I2C 清除 ACTIVITY 中断寄存器 (IC_CLR_ACTIVITY)

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.															

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

Bits	31:1	保留, 必须保持复位值
Bit	0	CLR_ACTIVITY:
		如果 I2C 不再活跃, 读该寄存器清除 ACTIVITY 中断
		如果 I2C 仍活跃在总线上, ACTIVITY 中断置位
		如果 I2C 不在总线上活跃且 I2C 禁止, 该位硬件自动清 0
		读 IC_RAW_INTR_STAT.ACTIVITY, 可知 ACTIVITY 中断的状态信息

20.5.22 I2C 清除 STOP_DET 中断寄存器 (IC_CLR_STOP_DET)

地址偏移: 0x60

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_STOP_DET
															r

Bits	31:1	保留, 必须保持复位值
Bit	0	CLR_STOP_DET: 读该寄存器清除 STOP_DET 中断

20.5.23 I2C 清除 START_DET 中断寄存器 (IC_CLR_START_DET)

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
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

20.5.24 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

20.5.25 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.											SMBUS_ALERT	ACK_MODE	TX_CMD_BLOCK	ABORT	ENABLE
											rw	rw	rw	rw	rw

Bits 31:5	保留，必须保持复位值
Bit 4	SMBUS_ALERT : 驱动 SMBus Alert 线操作
	0: 释放 SMBus Alert 线 (默认)
	1: 驱动 SMBus Alert 线为低电平
Bit 3	ACK_MODE : 数据 ACK 选择
	0 : 数据接收结束后返回 ACK (匹配地址或接收数据)
	1 : 数据接收结束后不返回 ACK
	注 1 : 从机: 硬件保证, 在 SMB_MODE 模式下, 不管地址匹配是否成功都回复 ACK。
	注 2 : 主机: 硬件保证, 在地址发送后检查 ACK 回复, 如果检测到 NACK, 发送 STOP。
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
	注 : 为阻止主机命令的执行, 只能在 TX FIFO 为空 (IC_STATUS.TFE=1) 且主机处于 IDLE 状态

	(IC_STATUS.MST_ACTIVITY = 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 FIFO 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。

20.5.26 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_HOLD_RX_FIFO_FULL	SLV_HOLD_TX_FIFO_EMPTY	MST_HOLD_RX_FIFO_FULL	MST_HOLD_TX_FIFO_EMPTY	SLV_ACTIVITY	MST_ACTIVITY	RFF	RFNE	TFE	TFNF	ACTIVITY
					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

20.5.27 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： ● 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

20.5.28 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： ● I2C 禁止 ● 发生一个发送丢弃，由 IC_TX_ABRT_SOURCE 的任何事件引起

20.5.29 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.ENABLE 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。	

20.5.30 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 _ABRT
r															r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRT_ SLVRD _INTX	ABRT_ _SLV_ ARBLOST	ABRT_ SLVFLUSH _TXFIFO	ARB_ LOST	ABRT_ MASTER _DIS	ABRT_ 10B_RD_ NORSTRT	ABRT_ SBYTE_ NORSTRT	Re s.	ABRT_ SBYTE_ ACKDET	Re s.	ABRT_ GCALL _READ	ABRT_ GCALL _NOACK	ABRT_ TXDATA _NOACK	ABRT_ 10ADDR2 _NOACK	ABRT_ 10ADDR1 _NOACK	ABRT_ 7ADDR _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_ABRT 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_ABRT: 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: 在 IC_DATA_CMD.CMD 位写入 1，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_ABRT_SOURCE.ARB_LOST 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_ABRT 中断来清除 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_ABRT_SOURCE.ABRT_SLV_ARBLOST 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_CON.RESTEN= 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_CON.RESTEN=0）并且用户正在尝试发送 START Byte
	清除 ABRT_SBYTE_NORSTRT，首先必须固定 ABRT_SBYTE_NORSTRT 的来源；重启必须使能（IC_CON.RESTEN =1），必须清除 IC_TAR.SPECIAL 位或必须清零 IC_TAR.GC_OR_START 位。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.STOP 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.STOP=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

20.5.31 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 FIFO 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: ● I2C 禁止 (IC_ENABLE.ENABLE = 0) ● 从机部分不活跃 (IC_STATUS.SLV_ACTIVITY = 0)

20.5.32 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

20.5.33 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。

20.5.34 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。

20.5.35 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.															

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 1000ns 的时延，在 10MHz 的 I2C_CLK 下，IC_SDA_SETUP 应该写入 11。
注1 ：只有当 IC_ENABLE.ENABLE = 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。	

20.5.36 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.															

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）

20.5.37 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.ENABLE 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.ENABLE 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
		注： 外围 I2C 主机在从机发送 NACK 之前，发出 STOP 条件，IC_ENABLE.ENABLE 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.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： ● 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）
		注： 外围 I2C 主机在从机发送 NACK 之前，发出 STOP 条件，IC_ENABLE.ENABLE 被设置为 0，该位也被设置为 1。
Bit	0	IC_EN: 该位反映 I2C 的状态信息
		0: I2C 不活跃状态（默认）
		1: I2C 活跃状态

	注：可在任何时候读该位。当该位读到的值为 0 时，可安全读取 SLV_RX_DATA_LOST 和 SLV_DISABLED_WHILE_BUSY。
	注：当 IC_ENABLE.ENABLE=1，SLV_RX_DATA_LOST 和 SLV_DISABLED_WHILE_BUSY 固定为 0，IC_EN 固定为 1；当 IC_ENABLE.ENABLE=0，SLV_RX_DATA_LOST 和 SLV_DISABLED_WHILE_BUSY 仅在 IC_EN 读到 0 时有效。

20.5.38 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.ENABLE 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。

20.5.39 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）

20.5.40 I2C 清除 TIMEOUT 寄存器（IC_CLR_TIMEOUT）

地址偏移：0xAC

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_TIMEOUT
															r

Bits	31:1	保留，必须保持复位值
Bit	0	CLR_TIMEOUT:
		读该寄存器来清除 TIMEOUT 中断（IC_RAW_INTR_STAT 的 bit 15）

20.5.41 I2C 清除 SMBUS_ALERT 寄存器（IC_CLR_SMBUS_ALERT）

地址偏移：0xB0

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_SMBUS_ALERT
															r

Bits	31:1	保留，必须保持复位值
Bit	0	CLR_SMBUS_ALERT:
		读该寄存器来清除 SMBUS_ALERT 中断（IC_RAW_INTR_STAT 的 bit 14）

20.5.42 I2C 清除 PEC_ERROR 寄存器（IC_CLR_PEC_ERROR）

地址偏移：0xB4

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_PEC_ERROR
															r

Bits	31:1	保留，必须保持复位值
Bit	0	CLR_PEC_ERROR:
		读该寄存器来清除 PEC_ERROR 中断（IC_RAW_INTR_STAT 的 bit 16）

20.5.43 I2CTIMEOUT 计时寄存器（IC_TIMEOUT_DATA）

地址偏移：0xB8



复位值: 0x0000_5554

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

16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC_TIMEOUT_DATA[7:0]																
rw																

Bits 31:17	保留, 必须保持复位值
Bits 16:0	IC_TIMEOUT_DATA[15:0]: 超时错误检测计数阈值, 当内部超时错误检测计数器计数达到该阈值时, 产生 TIMEOUT 中断, 即 IC_RAW_INTR_STAT 的 bit[15]。 该寄存器的配置值需要根据 IC_CLK 来计算得到, 如 SMBUS 协议中要求 $t_{LOW:MEXT}$ 最大为 10ms, 若 IC_CLK 为 32kHz, 那么 $timeout_data \times 10 \times \frac{1}{32KHz} = 10ms$, 最终计算得到 $timeout_data=32$, 即 0x20。

20.5.44 I2C PEC 数据寄存器 (IC_PEC_DATA)

地址偏移: 0xBC

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.								IC_PEC_DATA[7:0]							
								rw							

Bits 31:8	保留, 必须保持复位值
Bits 7:0	IC_PEC_DATA[7:0]: CRC 模块计算值, 由软件写入 I2C 设备在接收带有 PEC 的数据时, 首先由软件将接收的所有数据放入 CRC 模块中进行计算, 然后将计算结果放入内部 PEC_DATA 寄存器中。接收完最后一组数据后将接收到的 PEC 与 PEC_DATA 内部的计算结果进行对比, 若对比错误则产生 PEC_ERROR 中断, 即 IC_RAW_INTR_STAT 的 bit[16]。

20.5.45 I2C 清除 ADDR0_MATCH 寄存器 (IC_CLR_ADDR0_MATCH)

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.															

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

Res.														CLR_ADDR0_MATCH
														r

Bits	31:1	保留，必须保持复位值
Bit	0	CLR_ADDR0_MATCH:
		读该寄存器来清除 ADDR0_MATCH 中断（IC_RAW_INTR_STAT 的 bit 17）

20.5.46 I2C 清除 ADDR1_MATCH 寄存器（IC_CLR_ADDR1_MATCH）

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.															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.														CLR_ADDR0_MATCH	
														r	

Bits	31:1	保留，必须保持复位值
Bit	0	CLR_ADDR1_MATCH:
		读该寄存器来清除 ADDR1_MATCH 中断（IC_RAW_INTR_STAT 的 bit 18）

21. USART/LIN 接口

LCM32F06X 内置了 2 个 USART 接口（USART0、USART1），USART 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 48MHz。

USART0 和 USART1 支持 modem 流控操作（CTS/RTS），所有 USART 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，USART0/1 的功能如下表所示：

表 21-1 USART0/1 支持的功能

	DMA 访问	CTS/RTS 硬件流控	发送/接收 FIFO
USART0	支持	支持	支持
USART1	支持	支持	支持

21.1 USART 主要特性

- 支持全双工/单线半双工，支持异步或同步发送模式
- 精准的波特率生成，最高可达 3.686Mbps
- 带有 16Bytes 的发送和接收 FIFO，支持 DMA 操作
- 发送和接收 FIFO 中断触发阈值可选
- USART0 和 USART1 实现 CTS/RTS 硬件流控
- 2 个 USART 均硬件支持 LIN 功能，在半双工模式或 LIN 模式下支持冲突检测
- 支持控制输入、输出的极性反转
- 独立的工作时钟
- 2 个 USART 均支持低功耗模式，能够把系统从低功耗模式唤醒

21.2 管脚配置

表 21-2 USART 管脚配置

管脚名称	管脚类型	管脚描述	复用 I/O 口	备注
USART0_RX	I	USART0 接收管脚	PA7/PA9/PA10/PA12/PB0/PB1/PB4/PB6/PB7	-
USART0_TX	O	USART0 发送管脚	PA8/PA9/PA10/PA11/PB0/PB1/PB2/PB5/PB6/PB7	-
USART0_CK	O	USART0 同步时钟管脚	PA6	-
USART0_CTS	I	USART0 硬件控制 CTS 管脚	PA8/PA11	-
USART0_RTS	O	USART0 硬件控制 RTS 管脚	PA4/PA12	-
USART1_RX	I	USART1 接收管脚	PA0/PA2/PA3/PA13/PA14/PA15	-
USART1_TX	O	USART1 发送管脚	PA1/PA2/PA3/PA4/PA14/PB2/PB3	-
USART1_CK	O	USART1 同步时钟管脚	PA4	-
USART1_CTS	I	USART1 硬件控制 CTS 管脚	PA0	-
USART1_RTS	O	USART1 硬件控制 RTS 管脚	PA1/PB1	-

21.3 功能描述

21.3.1 时钟

USART 模块的时钟 USART_CLK 必须适应波特率的设置

- $F_{\text{USART_CLK}}(\text{min}) \geq 16 * \text{baud_rate}(\text{max})$
- $F_{\text{USART_CLK}}(\text{max}) \leq 16 * 65535 * \text{baud_rate}(\text{min})$

例如，波特率的范围为 110bps ~ 460800bps，那么工作时钟 USART_CLK 频率必须在 7.3728MHz ~ 115.34MHz 间。USART_CLK 和配置时钟 PCLK0 存在一个约束关系：USARTCLK 频率不超过 PCLK0 频率的 5/3 倍，即 $F_{\text{USART_CLK}} \leq 5/3 * F_{\text{PCLK0}}$ 。在 USART_CLK 为 14.7456MHz 的条件下产生 921600 的波特率，PCLK0 需设置高于 8.85276MHz，以此保证 USART 有足够的时间来将接收到的数据写入接收 FIFO。时钟 USART_CLK 相关配置为：

- 配置系统 SysCtrl_ClkEnR1 寄存器中的 USARTx_CLKSEL[1:0] 和 USARTx_CLKDIV 位，选择 USART_CLK 时钟源和分频（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 USARTx_CLKEN 位，使能 USART 时钟

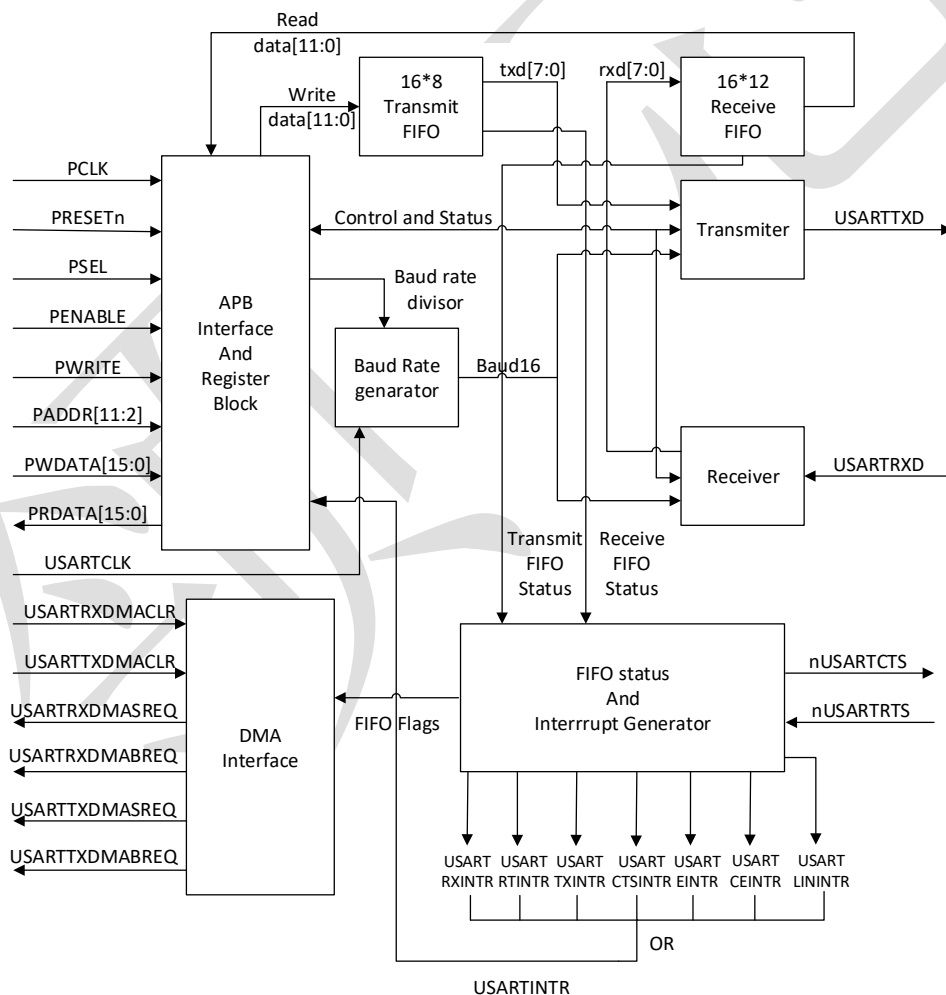


图 21-1 USART 框图

21.3.2 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。

$$\text{Baud Rate Divisor} = \text{USART_CLK} / (16 \times \text{Baud Rate}) = \text{BRDI} + \text{BRDF}$$

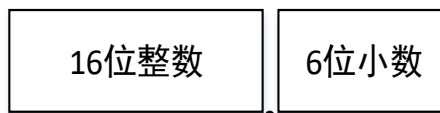


图 21-2 波特率分配器

波特率计算公式如下：

$$\text{BAUDDIV} = (F_{\text{USART_CLK}} / (16 \times \text{Baudrate}))$$

$F_{\text{USART_CLK}}$ 是 USART 的外设时钟。BAUDDIV 由整数值 (BAUD DIVINT) 和小数值 (BAUD DIVFRAC) 组成。USARTIBRD 和 USARTFBRD 仅能在发送或接收完成后才能更新配置。BAUDDIV 的范围是 $1 \sim 65535$ ($2^{16} - 1$)，USARTIBRD 最小值为 1，且当 USARTIBRD = 65535 时 USARTFBRD 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 $\text{BRDI} = 1$ ， $\text{BRDF} = 0.085$ 。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 USARTFBRD 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 USART_CLK 为 16MHz，对应的 USARTIBRD，USARTFBRD 以及产生的波特率。

表 21-3 波特率表 (USART_CLK 为 16MHz)

USARTIBRD	USARTFBRD	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 USART_CLK 为 24MHz，对应的 USARTIBRD，USARTFBRD 以及产生的波特率。

表 21-4 波特率表 (USART_CLK 为 24MHz)

USARTIBRD	USARTFBRD	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

USARTLCR_H、USARTIBRD 和 USARTFBRD 构成了位宽为 30 的寄存器 USARTLCR，这个寄存器通过写 USARTLCR_H 操作更新，所以 USARTLCR_H 的写操作必须放在写 USARTIBRD/USARTFBRD 操作之后进行。同时更新 3 个寄存器的顺序：

- 写 USARTIBRD，写 USARTFBRD，写 USARTLCR_H
- 写 USARTFBRD，写 USARTIBRD，写 USARTLCR_H

仅更新寄存器 USARTIBRD 或 USARTFBRD 的顺序：

- 写 USARTIBRD，写 USARTLCR_H
- 写 USARTFBRD，写 USARTLCR_H

21.3.3 USART 帧格式

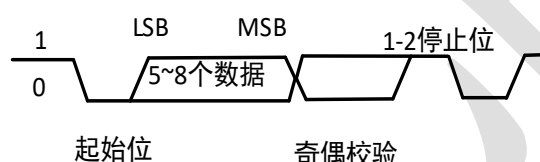


图 21-3 USART 字节帧

通过配置 USARTLCR_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 USARTLCR_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

21.3.4 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 16 的 FIFO 中。对于发送，数据写入到发送 FIFO 中。一旦 USART 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）才清除 BUSY 状态。

USART 的发送过程如下：

1. 时钟配置
2. 通过 USARTIBRD 和 USARTFBRD 选择合适的波特率
3. 通过 USARTLCR_H 寄存器选择发送数据位宽、停止位、奇偶校验
4. 通过 USARTCR 寄存器选择是否使能硬件控制流和 LBE 回环，并使能 USART 和发送
5. 通过向 USARTDR 寄存器写入数据激活 USART 发送

21.3.5 数据的接收

接收存储在 12 位宽，深度为 16 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（USARTRXD 持续为 1）且数据线上检测到低电平，计数器开始工作。在 USART 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 USARTRXD 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 USARTRXD 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 21-5 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 USARTSR 前，必须先读取 USARTDR，次序不能颠倒。USARTDR 内的 OE/BE/PE/FE 等信息不能长时间维持。通过读 USARTDR 操作，将接收状态保存到 USARTSR 中，通过读取 USARTDR 也可以获取当前状态信息。

USART 的接收过程如下：

1. 时钟配置
2. 通过 USARTIBRD 和 USARTFBRD 选择合适的波特率
3. 通过 USARTLCR_H 寄存器选择接收数据位宽、停止位、奇偶校验
4. 通过 USARTCR 寄存器选择是否使能硬件控制流，并使能 USART 和接收
5. 从数据寄存器 USARTDR 中读走接收到的数据

21.3.6 USART 硬件流控制

USART 支持硬件流控通过 nRTS 输出信号和 nCTS 输入信号控制两个 USART 之间的数据。

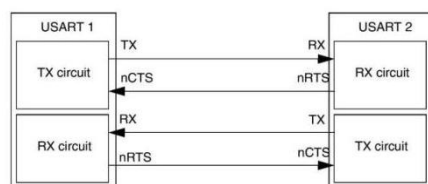


图 21-4 两个相似设备间的硬件流控

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 USARTCR 的 RTSEn 和 CTSEn 控制。表 21-5 描述了这两位在不同配置下的控制流情况。

表 21-6 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

21.3.6.1 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，nUSARTRTS 为低电平。一旦达到基准 nUSARTRTS 置 1，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，nUSARTRTS 置成低电平。如果 RTS 流控使能关闭（USART 使能），数据根据当前 nUSARTRTS 的状态，要么一直接收至 FIFO 满，要么不接收数据。

21.3.6.2 CTS 流控制

当 CTSEn 使能时，发送器在传送数据前检查 nUSARTCTS 的状态。如果 nUSARTCTS 为低电平，发送数据。反之，不发送。当发送 FIFO 非空且 nUSARTCTS 为低电平，发送器将持续发送数据。当 CTSEn 使能时，nUSARTCTS 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（USART 使能），发送器持续发送数据直至 FIFO 空。

21.3.7 单线半双工模式

开启单线半双工通讯前需要内部或外部接入上拉电阻，维持空闲时线上为无效电平。HDSEL 置位开启半双工模式后，所有 RX 端口开启输出功能（TX 端口功能保留）。

RX 端口在半双工模式下默认保持开漏输出，HDOD 置位后切换到推挽输出模式（如果此时 GPIO 配置为开漏模式，则最后端口仍为开漏输出）。

未通讯时 USART 一直处于接收模式，接收到线上数据时接收状态机开始工作；如果需要向线上发送数据，则在写数据寄存器发起通讯前必须软件主动查询 RXBUSY 位，保证 USART 在非接收状态下软件配置进行发送数据，防止线上出现信号错误。若通讯双方同时发起通讯，则检测 RXBUSY 可能会失效。

21.3.8 同步发送模式

USART 寄存器 USARTCR2.CLKEN 置位开启 USART 模式（同步模式）后，USART_CK 端口开启输出功能，接收端也切换为同步模式。USART 模式仅支持主机模式，软件配置发送后，在同一帧内完成发送和接收（可以关闭接收使能来阻止接收数据）。发送端保持不变；接收端改为只同步接收数据，不检查 start 位、stop 位和奇偶校验位；同步时钟 CK 在发送帧内的数据位时有效，电平极性和相位由寄存器 USARTCR2 中的 CPOL 和 CPHA 位控制。

21.3.9 低功耗唤醒模式（LPUSART）

当前 2 个 USART 支持在低功耗模式下唤醒系统。进入低功耗前，用户需把 USART 时钟切换到 OSCL，

同时配置 EXTI 线 22 或 23 有效。置位寄存器 USARTCR3.UELP 开启 LPUSART 模式（低功耗唤醒模式）后，USART 低功耗功能有效。可通过寄存器 USARTCR3.WUS 选择唤醒事件，在发生选择的唤醒事件后，低功耗唤醒中断标志位由硬件置位，经过屏蔽位后生成中断信号，并通过 EXTI 唤醒系统。

唤醒事件包括：

- 地址匹配激活
- 起始位检测时激活，即检测到一位电平后唤醒
- RX 时激活，完成一帧数据接收后唤醒

USART 进入低功耗唤醒模式后，在接收端检测到低电平后打开 LPUSART 工作时钟。在完成一帧数据的接收后，若没有通过唤醒事件唤醒 MCU，则关闭 LPUSART 工作时钟。

21.3.10 LIN 模式

LIN 总线是基于 SCI（USART）数据格式，采用单主控制器/多从设备的模式，是 USART 中的一种特殊情况。USART 开启 LIN 模式后，支持硬件检测 13bit 长低电平的同步间隔段并产生相对应的 LIN 中断。配合软件可以实现所有 LIN 协议的通讯。使用 LIN 模式时，USART 必须将帧格式调整为 8 位数据，无奇偶校验位。

21.3.10.1 LIN 帧格式

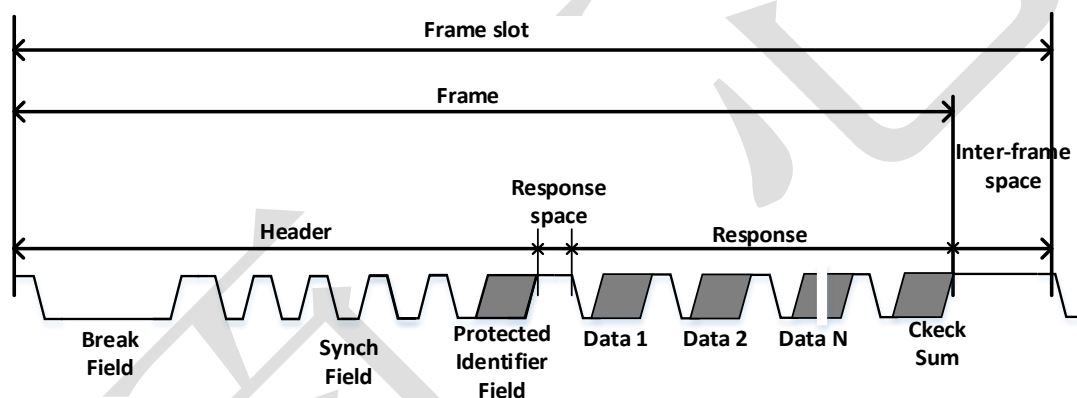


图 21-5 LIN 的帧结构

帧包含帧头和应答两部分。主机负责发送帧头。从机接收帧头并对帧头所包含信息进行解析，然后决定发送应答，或接收应答，或不作任何反应。帧头包括同步间隔段、同步段以及 PID 段，应答包括数据段和校验和段。

LIN 主机：

LIN 同步间隔段推荐使用以下三种操作产生，其他通讯阶段通过软件配合实现。

1. 调整波特率：通过放慢波特率将 9bit 低电平变相实现成快波特率下的 13bit 长低电平信号
2. 端口模拟：通过软件控制 GPIO 端口输出低电平，模拟 13bit 长低电平信号后调整复用给 USART
3. Break 信号：支持发送 Break 信号，直接发送两帧以上的低电平信号代替同步间隔段

LIN 从机：

USART 配置寄存器 USARTCR.LINEN 位，启动 USART 的 LIN 模式。此模式下，USART 可以硬件实时监测长低电平信号，一旦满足 13bit 低电平，立即产生对应的 LIN 中断，配合软件实现从机全功能的通讯。

21.3.10.2 LIN 校验方式

通过配置数据配置寄存器 DATCON 中 CSM 位的值，可选择经典校验和增强校验。

经典校验

经典型校验算法涵盖的范围只有 LIN 的数据段，即一帧 LIN 信号的数据段包含几个字节的数据就计算几个字节，计算的时候是累加计算。 $\text{byte } 0 + \dots + \text{byte } n + \text{checksum} = 0\text{xFF}$ 的整数倍

增强校验

增强型校验相比于经典型校验，计算的范围除了数据段还加上了 ID 的数值。 $\text{ID} + \text{byte } 0 + \dots + \text{byte } n + \text{checksum} = 0\text{xFF}$ 的整数倍

21.3.10.3 LIN 数据的发送与接收

LIN 帧头传输

只由主机进行，在中断检测有效后，依次传输帧头的同步间隔段、同步段、PID 段。发送的数据存储在 16 位宽，深度为 8 的 FIFO 中。LIN 同步间隔的产生通过放慢波特率将 9bit 低电平变相实现成快波特率下的 13bit 长低电平信号。在整个过程中中断检测功能始终保持开启，帧头发送结束时产生中断。

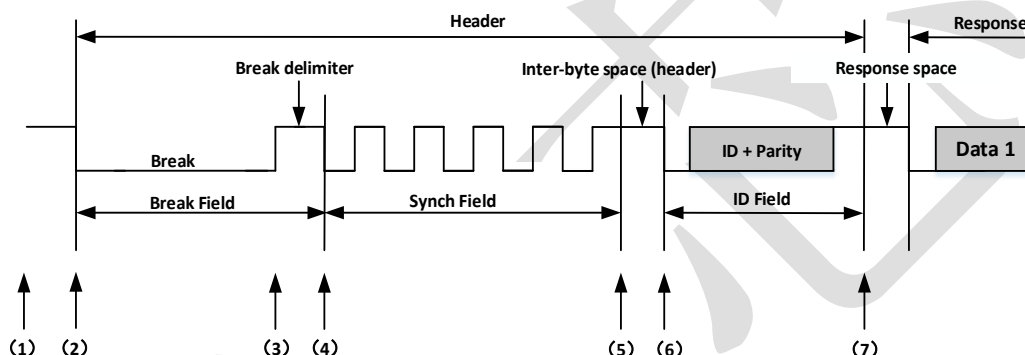


图 21-6 帧头发送中 LIN 主机操作

LIN 应答传输/接收

主机与从机都可进行应答传输/接收，依次进行数据段与校验和的传输/接收，一个字段中最多有八个数据段，在传输/接收结束后产生中断。校验方式可选经典校验和增强校验。可通过配置 DATCON.HO 选择 LIN 帧是由报头和应答组成还是仅由报头组成。

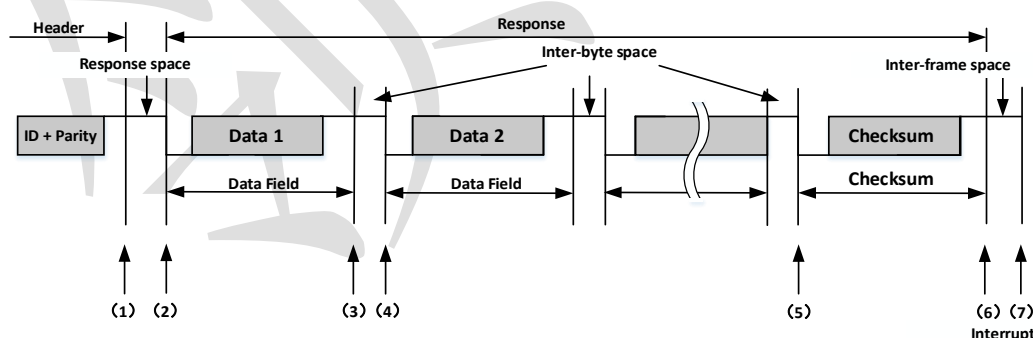


图 21-7 响应传输操作

LIN 帧头接收

仅从机支持帧头应答，自动进行波特率检测，并自动同步波特率。

波特率同步原理：在已知主机波特率情况下，从机在同步字节 0x55 期间，统计第 1 个下降沿到第 5 个下降沿（A 点→B 点）主频时钟所需要的个数，即 BRDMSU。根据公式推导，BRDMSU[15:7]为新波特率的整数部分，BRDMSU[6:1]为新波特率的小数部分。在同步字节期间完成波特率同步，作用域 PID 和应答数据，

以保证主从机正常通讯。使用 ABD 波特率自动同步时，应检查同步字节错误 SFRIS /帧错误 FERIS /波特率溢出错误 LARIS。在有些情况下（如 0x55 出错），即使波特率能校正，接收数据准确性也不能保证（出现位置偏移）。

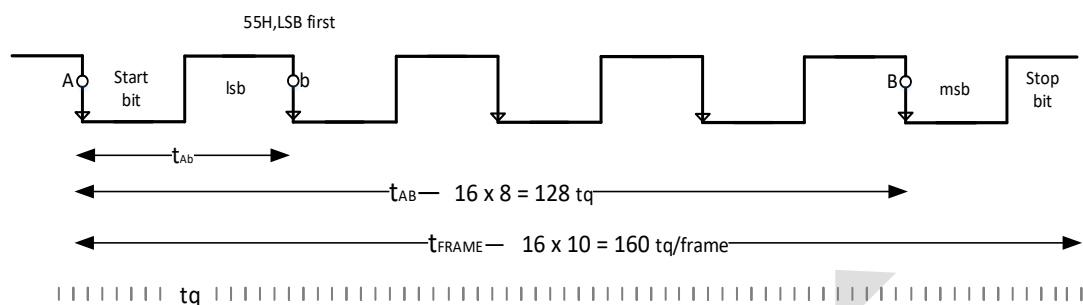


图 21-8 波特率同步示例

21.3.10.4 睡眠模式

LIN 有两种工作模式：正常模式与睡眠模式。将 LINSLP 位的值置为 1，用于指示 LIN 网络进入睡眠模式，芯片功耗并无减少。如需降低功耗，用户需另行配置。通过睡眠命令或检测总线空闲的方式进入睡眠模式。睡眠模式下，通过接收一字节数据并根据连续低电平位数计算长低电平时间，决定是否退出睡眠模式。在一个处于睡眠状态的 LIN 网络中，任何一个节点都可以发送唤醒信号。

唤醒信号是一个 250us（在 20Kbit/s 波特率）到 5ms（在 1Kbit/s 波特率）的显性电平（5 个位）。每一个节点都需要检测唤醒信号，当检测到唤醒信号之后，在 100ms 内完成初始化工作。当从节点发出唤醒信号之后 150ms，主节点仍未帧头发送时，从节点可以再次发送唤醒信号。当连续发送了 3 次唤醒信号之后如果主节点仍未帧头发送，从节点至少要等待 1.5 秒以后才可以发送第四次唤醒信号。

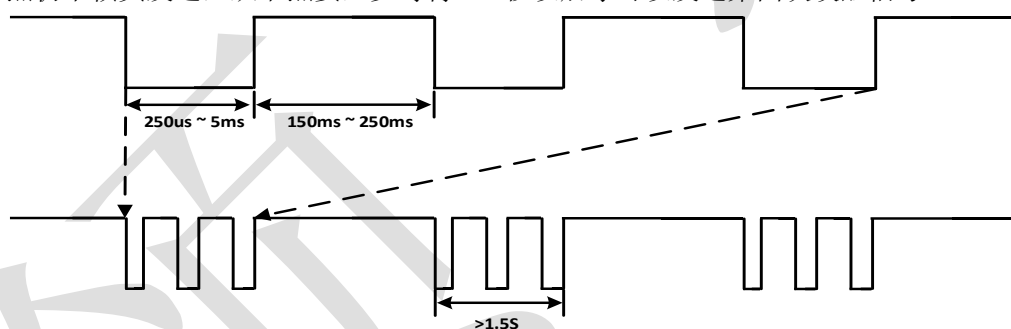


图 21-9 网络唤醒示意图

网络唤醒：主机可以通过传输唤醒周期来唤醒从机（高电平 250us-5ms）。其中三个唤醒请求之间间隔最小为 150 ms，最大为 250 ms。该周期在第六次唤醒传播之后结束。每三次唤醒中间的间隔时间最小是 1.5s，最大是 3s，一个完整唤醒周期的最小运行时间是 2.1 秒。

进入睡眠模式：

方式 1：接收到主机发送的睡眠命令，软件置位 LINSLP 进入睡眠模式。睡眠命令为：Data0 ~ Data8 : 0x00 + 0xFF + 0xFF + 0xFF + 0xFF + 0xFF + 0xFF + 0xFF

方式 2：检测总线空闲 4-10s，软件置位 LINSLP 进入睡眠模式。总线状态检测可通过定时器或 WT，结合总线的边沿中断 FED/RED 实现。

退出睡眠模式：

在睡眠模式下，接收数据功能正常工作。收到一个数据起中断，用户读取该数据，根据波特率计算该数据低电平（连续 0 的位数）维持的时间是否在 150us-5ms 范围内，再决定是否清除 LINSLP 退出睡眠模式。

发送唤醒信号:

用移位寄存器和 1 个字节数据实现唤醒信号，结合波特率，产生 250us~5ms 的低电平脉冲。用户写入数据，通常是数据 0xF0（连续 4 个 0 的低电平）或 0xE0（连续 5 个 0 的低电平），0 的位数由波特率决定。往 USARTDR 写入 0xF0 或 0xE0，并置位 WUTR 发送该字节，产生 250us~5ms 的低电平脉冲。

21.3.10.5 测试模式

LINCORE 共有两种测试模式：环回模式和自测模式。

环回模式

在该模式下，LINCORE 执行一个从其 TX 输出到其 RX 输入的内部反馈。RX 输入引脚和 LINRX 引脚断开。在主模式下，LINCORE 可以接收其自身发送的消息，总线可以接收消息。

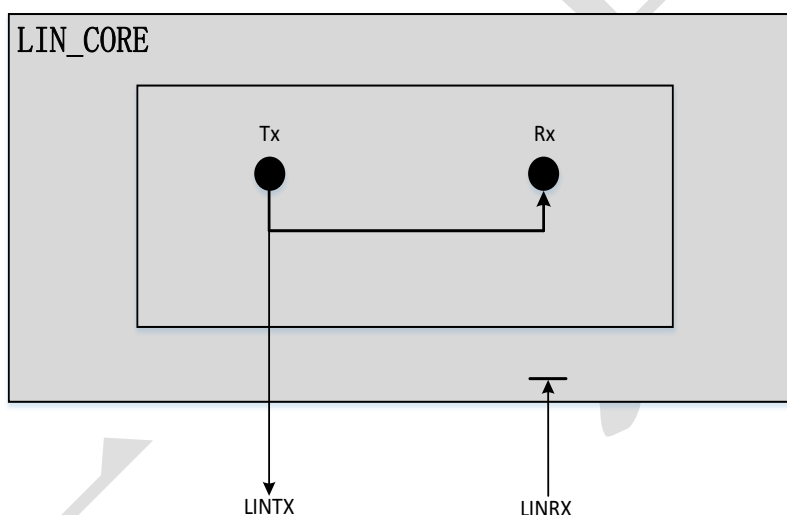


图 21-10 环回模式

自测模式

在该模式下，LINCORE 执行一个从其 TX 输出到其 RX 输入的内部反馈。RX 输入引脚和 LINRX 引脚断开。TX 输出引脚和 LINRTX 引脚断开。在主模式下，LINCORE 可以接收其自身发送的消息，但总线不能接收消息。

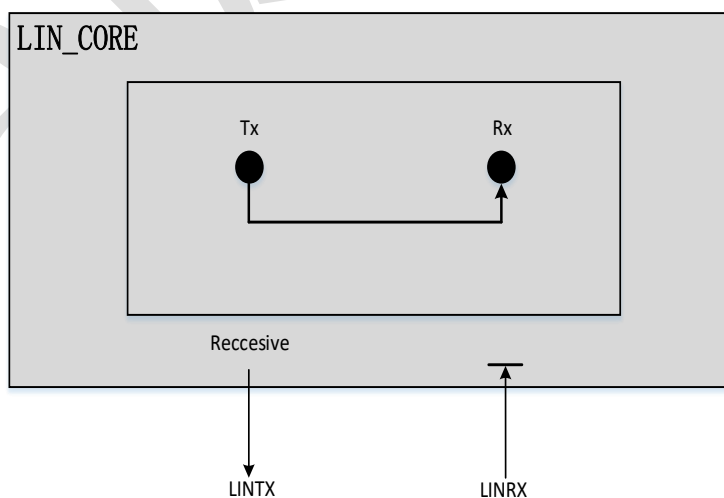


图 21-11 自测模式

21.3.10.6 错误检测

LIN 模式中会检测以下几种错误

- **位错误 (Bit error)**：在发送过程中，从总线读回的值和发送值不同时发生位错误，该错误在帧头或应答发送期间发生。Bit error 检测使能存在 FRAMECON 寄存器中，Bit error 中断屏蔽存在 USARTIMSC 寄存器中。
- **帧错误 (Framing error)**：已经对当前接收字符的停止位进行采样的显性状态 (sync field, identifier field 或 data field)。Framing error 中断状态存在 USARTRIS 寄存器中。
- **校验和错误 (Checksum error)**：计算的校验和和收到的校验和不匹配。Checksum error 中断状态位存在 USARTRIS1 寄存器中。
- **超时错误 (Time out error)**：帧头超时，应答超时或帧超时。

LINTIMER 寄存器定义了 LIN 帧头的时间窗口。在主模式下，定时器在 break 脉冲的下降沿开始计数，并在帧头的最后一位传输完成后停止计数。如果超出了 HEADER 字段中的预定义时间，将生成错误中断（如果中断使能）。

在从模式下，当检测到 break 脉冲时（满足 break 侦测阈值），定时器才开始计时。从机收到帧头的最后一个位，定时器停止工作。如果超出了 HEADER 位字段中的预定义时间，则会产生错误中断（如果中断使能）。

- **同步错误 (Sync error)**：从机模式下，帧头收到的同步字节不是 0x55。Sync error 中断状态存在 USARTRIS1 寄存器中。
- **PID 错误 (Pid error)**：从机模式下，帧头收到的 pid 高两位奇偶校验错误。Pid error 中断状态存在 USARTRIS1 寄存器中。
- **波特率溢出错误 (Baud Overflow error)**：从机模式下，自动波特率校准时发生超范围错误。Baud Overflow error 中断状态存在 USARTRIS1 寄存器中。

表 21-7 USARTRIS1 寄存器功能位

USARTRIS1 bit	功能
12	SFRIS: LIN 同步段错误中断状态位
11	LZ RIS: LIN 长零中断状态位
10	RED RIS: LINRX 上升沿侦测中断状态位
9	FED RIS: LINRX 下降沿侦测中断状态位
8	LPRIS: LIN 标识符奇偶校验中断状态位（仅用于 LIN 模式，PID 字节校验）
7	LCRIS: LIN 校验和错误 (checksum) 中断状态位
6	LARIS: LIN 自动波特率侦测错误中断状态位
5	RTRIS: LIN 应答超时中断状态位
4	HTRIS: LIN 帧头超时中断状态位
3	TRRIS: LIN 应答发送中断状态位
2	THRIS: LIN 帧头发送中断状态位
1	RRRIS: LIN 应答接收中断状态位
0	RHRIS: LIN 帧头接收中断状态位

21.3.11 冲突检测

冲突检测功能在半双工模式和 LIN 模式下有效，支持在 TX 发送阶段到与 RX 接收端电平的一致性检查。开启冲突检测（配置寄存器 FRAMECON.CEN=1）后，相关功能有效。当发送的数据和接收的数据不一致时，冲突检测标志位硬件置位，经过屏蔽位后生成中断信号。

21.3.12 DMA 接口

USART0/1 支持 DMA 访问，提高数据通讯的效率。USART 的 DMA 接口有以下信号：

- USARTRXDMASREQ：DMA 单次读取请求，由 USART 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。
- USARTRXDMABREQ：DMA 连续读取请求，由 USART 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 USARTIFLS 配置 FIFO 的水平基准。
- USARTRXDMACLR：DMA 读取请求清除信号，由 DMA 控制器置位，用于清除 USART 读取请求信号。如果 USART 向 DMA 控制器发出连续读数据请求（USARTRXDMABREQ），该清除信号将维持高电平到连续传输的最后一个数据。
- USARTTXDMASREQ：DMA 单次写入请求，由 USART 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。
- USARTTXDMABREQ：DMA 连续写入请求，由 USART 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 USARTIFLS 配置 FIFO 的水平基准。
- USARTTXDMACLR：写入请求清除信号，由 DMA 控制器置位，用于清除 USART 写入请求信号。如果 USART 向 DMA 控制器发出连续写数据请求（USARTTXDMABREQ），该清除信号将维持高电平到连续传输的最后一个数据。

DMA 单次请求和连续请求不会相互干扰，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，USART 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，USART 的请求信号又可被再次置位。

所有 DMA 请求信号在 USART 禁止或相关 DMA 使能（TXDMAE、RXDMAE）关闭，USART 请求信号自动清除。如果 FIFO 禁止，仅支持 DMA 次单请求。

此外，当接收错误发生时，DMAONERR 置位（USARTDMACR），产生接收错误中断（USARTEINTR），USARTRXDMASREQ 和 USARTRXDMABREQ 将被屏蔽，直至 USARTEINTR 清 0。DMA 的发送请求不受接收错误影响。

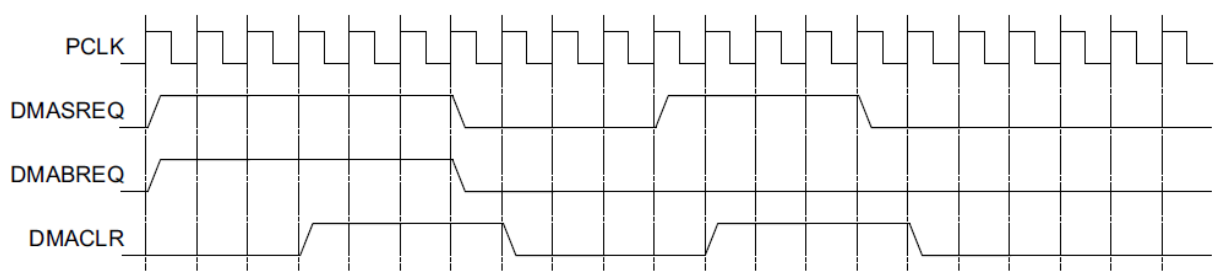


图 21-12 DMA 单请求和 DMA 连续请求与 DMACLR 间的时序关系

21.4 USART 中断

USART 有 7 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 USART 全局中断向量。

USARTINTR 包含以下 7 个中断源：

- USARTRXINTR
- USARTTXINTR
- USARTRTINTR
- USARTEINTR，由以下条件触发：
 - USARTBEINTR，接收过程中出现打断
 - USARTPEINTR，接收的字节出现奇偶校验错误
 - USARTFEINTR，接收的字节出现帧错误
- USARTEINTR
- USARTCEINTR
- USARTLININTR

最终 USARTINTR 输出由上述 7 个中断源经过或逻辑输出，用户可通过 USARTIMSC 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 USARTRIS 或 USARTMIS 中读取。

USARTRXINTR

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种方式清除中断：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。

USARTTXINTR

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种方式清除中断：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。

USARTRTINTR

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 种方式清除中断：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。

USARTEINTR

USART 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 USARTMIS 或 USARTRIS 来判断错误源，写中断清除寄存器相关位清除中断。

USARTCTSINTR

USART 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。

USARTCEINTR

半双工模式或 LIN 模式下，发送数据和接收数据不一致时，发生冲突中断，写中断清除寄存器相关位清除中断。

USARTLININTR

USART 开启 LIN 模式后，所有的 LIN 相关中断都会经过或逻辑产生 USARTLININTR 中断，写中断清除寄存器相关位清除中断。



表 21-8 USART 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
冲突检测错误	CEMIS	CEIM
LIN 接收报头	LINMIS	LINIM

USART 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。

21.5 USART 寄存器描述

USART0 基地址为 0x4001_4000，USART1 基地址为 0x4001_5000，USART0/USART1 的地址偏移分别对应各自的基地址。

表 21-9 USART 相关寄存器表

名称	说明	读写权限	复位值	地址偏移
USARTDR	数据寄存器	R/W	0x0000_0000	0x00
USARTSR	接收状态寄存器	R/W	0x0000_0000	0x04
USARTCR2	控制寄存器 2	R/W	0x0000_0000	0x0C
USARTCR3	控制寄存器 3	R/W	0x0000_0000	0x10
USARTFR	标志寄存器	R	0x0000_0090	0x18
USARTIBRD	整数波特率寄存器	R/W	0x0000_0000	0x24
USARTFBRD	小数波特率寄存器	R/W	0x0000_0000	0x28
USARTLCR_H	线控制寄存器	R/W	0x0000_0000	0x2C
USARTCR	控制寄存器	R/W	0x0000_0300	0x30
USARTIFLS	中断 FIFO 触发水平选择寄存器	R/W	0x0000_0088	0x34
USARTIMSC	中断屏蔽寄存器	R/W	0x0000_0000	0x38
USARTRIS	原始中断状态寄存器	R	0x0000_0000	0x3C
USARTMIS	屏蔽后中断状态寄存器	R	0x0000_0000	0x40
USARTICR	中断清除寄存器	W	0x0000_0000	0x44
USARTDMACR	DMA 控制寄存器	R/W	0x0000_0000	0x48
LINCON1	LIN 控制寄存器 1	R/W	0x0000_0000	0x74
LINCON2	LIN 控制寄存器 2	R/W	0x0000_0000	0x78
FRAMECON	帧控制寄存器	R/W	0x0000_0000	0x7C
LINSTS	LIN 状态寄存器	R/W	0x0000_0000	0x94
DATCON	数据配置寄存器	R/W	0x0000_FF00	0x98
BRDMSU	波特率侦测寄存器	R/W	0x0000_0000	0x9C

BRDUPPER	波特率侦测上限寄存器	R/W	0x0000_0000	0xA0
BRDLOWER	波特率侦测下限寄存器	R/W	0x0000_0000	0xA4
LINBTIMER	LIN BREAK 定时寄存器	R/W	0x0000_000D	0xA8
LINHTIMER	LIN 帧头定时寄存器	R/W	0x0000_00FF	0xAC
USARTIMSC1	中断屏蔽寄存器 1	R	0x0000_0000	0xB0
USARTRIS1	原始中断状态寄存器 1	R	0x0000_0000	0xB4
USARTMIS1	屏蔽中断状态寄存器 1	R	0x0000_0000	0xB8
USARTICR1	中断清除寄存器 1	W	0x0000_0000	0xBC

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）。

21.5.1 USART0/1 数据寄存器（USARTDR）

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。	

21.5.2 USART0/1 接收状态寄存器 (USARTRSR)

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 USARTDR 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 USARTDR 操作, 使 FIFO 处非满状态, 该位清 0。
	该位和 USARTDR 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

21.5.3 USART0/1 控制寄存器 2 (USARTCR2)

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.								CPHA	CPOL	CLKEN	Res.			HDOD	HDSEL
								rw	rw	rw				rw	rw

Bits 31:8	保留，必须保持复位值
Bit 7	CPHA : 同步时钟相位，与 CPOL 位共同控制 CK 信号
	0: 数据在同步时钟脉冲第一个边沿有效（默认）
	1: 数据在同步时钟脉冲第二个边沿有效
Bit 6	CPOL : 同步时钟极性，与 CPHA 位共同控制 CK 信号
	0: CK 时钟空闲时为低电平（默认）
	1: CK 时钟空闲时为高电平
Bit 5	CLKEN : USART 同步时钟使能信号
	0: 关闭 USART 同步模式和 CK 引脚（默认）
	1: 启动 USART 同步模式和 CK 引脚
Bits 4:2	保留，必须保持复位值
Bit 1	HDOD : 半双工模式输出模式选择
	0: 半双工模式下 RXD 端口为开漏输出（默认）
	1: 半双工模式下 RXD 端口为推挽输出
Bit 0	HDSEL : 半双工模式选择
	0: 关闭单线半双工模式（默认）
	1: 开启单线半双工模式
注 1: 半双工模式下 RXD 端口的 GPIO 配置为开漏输出，则强制为开漏输出。	
注 2: LIN 模式、半双工模式和 USART 同步模式仅可以开启一个。	

21.5.4 USART0/1 控制寄存器 3（USARTCR3）

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
ADDR[7:0]								ADDMM	Res.			WUS[1:0]	Res.		UELP
rw								rw				rw			rw

Bits 31:16	保留，必须保持复位值
Bits 15:8	ADDR[7:0] : USART 低功耗唤醒模式的唤醒地址
Bit 7	ADDMM : 唤醒地址模式选择
	0: 4 位地址检测（默认）
	1: 7 位地址检测（LPUSART 要求数据格式在 7/8 位数据模式下）

Bits 6:5	保留，必须保持复位值
Bits 4:3	WUS[1:0]: USART 低功耗唤醒事件选择
	00: 在地址匹配时发生唤醒事件（按 ADDR[7:0]和 ADDM 决定）（默认）
	01: 保留
	10: 在起始位检测时发生唤醒事件
	11: 保留
Bits 2:1	保留，必须保持复位值
Bit 0	UELP: 低功耗模式 LPUSART 使能
	0: 低功耗模式 LPUSART 禁止（默认）
	1: 低功耗模式 LPUSART 使能

21.5.5 USART0/1 标志寄存器（USARTFR）

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.						RXBUSY	Res.	TXFE	RXFF	TXFF	RXFE	BUSY	Res.	nCTS	
						r		r	r	r	r	r			r

Bits 31:10	保留，必须保持复位值
Bit 9	RXBUSY: 接收忙标志位
	0: 处于非接收状态（默认）
	1: 处于接收状态
Bit 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（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（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: USART 忙标志位
		当 USART 在发送数据时该位置 1。不管 USART 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: USART 处于空闲状态 (默认)
		1: USART 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 nUSARTCTS 的互补位
		0: 表示 nUSARTCTS 为 1 (默认)
		1: 表示 nUSARTCTS 为 0

21.5.6 USART0/1 整数波特率寄存器 (USARTIBRD)

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

21.5.7 USART0/1 小数波特率寄存器 (USARTFBRD)

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

21.5.8 USART0/1 线控制寄存器 (USARTLCR_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: 5 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, USART 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, USART 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, USARTTXD 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：USARTLCR_H、USARTIBRD 和 USARTFBRD 构成了位宽为 30 的寄存器 USARTLCR，这个寄存器通过写 USARTLCR_H 操作更新，所以 USARTLCR_H 的写操作必须放在写 USARTIBRD/USARTFBRD 操作之后进行。 同时更新 3 个寄存器的顺序： ● 写 USARTIBR，写 USARTFBRD，写 USARTLCR_H ● 写 USARTFBRD，写 USARTIBRD，写 USARTLCR_H 仅更新寄存器 USARTIBRD 或 USARTFBRD 的顺序： ● 写 USARTIBRD，写 USARTLCR_H ● 写 USARTFBRD，写 USARTLCR_H	

表 21-10 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

21.5.9 USART0/1 控制寄存器（USARTCR）

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	RXTSP	TXTSP	LINEN	Res.		USARTEN	
rw	rw			rw		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 nUSARTCTS 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
Bit	13	保留，必须保持复位值
Bit	11	RTS: RTS 流控非使能时，USART 发送请求的互补位 0: 控制 nUSARTRTS 为 1（默认） 1: 控制 nUSARTRTS 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 USART 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 USART 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: USARTTXD 与 USARTRXD 分离 (默认)
	1: USARTTXD 路径插入到 USARTRXD
	如果 LBE =1, SIREN=1, SIRTEST=1, nISR0UT 路径插入到 SIRIN
	如果 LBE =1, SIREN=0, SIRTEST=0, USARTTXD 路径插入到 USARTRXD
	在 SIR 和 USART 模式下, 调制输出插入到调制输入
Bits 6	RXTSP : 接收端极性反转
	0: 空闲电平为高电平 (默认)
	1: 空闲电平为低电平
Bits 5	TXTSP : 发送端极性反转
	0: 空闲电平为高电平 (默认)
	1: 空闲电平为低电平
Bits 4	LINEN : LIN 模式使能位
	0: 禁用 LIN 相关功能 (默认)
	1: 开启 LIN 接收报头检测 (检查 13bit 低电平的报头)
Bits 3:1	保留, 必须保持复位值
Bit 0	USARTEN : USART 使能
	0: USART 禁止 (默认)
	1: USART 使能
注: 发送时 TXE 和 USARTEN 必须使能; 接收时 RXE 和 USARTEN 必须使能。配置 USARTLCR 流程: 1. 清 USART 使能 2. 等待当前字节发送或接收完毕 3. 通过把 USARTLCR_H 的 FEN 位清 0, 来清空发送 FIFO 4. 重新配置 USARTCR 5. 使能 USART	

21.5.10 USART0/1 中断 FIFO 触发水平选择寄存器 (USARTIFLS)

地址偏移: 0x34

复位值: 0x0000_0088

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[3:0]				TXIFLSEL[3:0]			
								rw				rw			

Bits 31:8	保留，必须保持复位值
Bits 7:4	RXIFLSEL[3:0] : 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，默认为 1000
Bits 3:0	TXIFLSEL[3:0] : 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，默认为 1000

21.5.11 USART0/1 中断屏蔽寄存器（USARTIMSC）

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.	CEIM	WUFIM	LINIM	Res.	OEIM	BEIM	PEIM	FEIM	RTIM	TXIM	RXIM	Res.		CTSIM	Res.
	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	CEIM : 检测错误中断屏蔽位
	0: 屏蔽（默认）
	1: 不屏蔽
Bit 13	WUFIM : 低功耗模式下唤醒事件屏蔽位
	0: 屏蔽（默认）
	1: 不屏蔽
Bit 12	LINIM : LIN 模式下接收报头中断屏蔽位
	0: 屏蔽（默认）
	1: 不屏蔽
Bit 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。	

21.5.11 USART0/1 原始中断状态寄存器（USARTRIS）

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.	CERIS	WUFRIS	LINRIS	Res.	OERIS	BERIS	PERIS	FERIS	RTRIS	TXRIS	RXRIS	Res.		CTSRIS	Res.
	r	r	r		r	r	r	r	r	r	r			r	

Bits 31:15	保留，必须保持复位值
Bit 14	CERIS: 冲突检测中断状态位
	0: 冲突检测中断未发生（默认）
	1: 冲突检测中断发生
Bit 13	WUFRIS: 低功耗模式下唤醒事件状态位
	0: 低功耗唤醒事件未发生（默认）
	1: 低功耗唤醒事件发生
Bit 12	LINRIS: LIN 模式下接收报头中断状态位
	0: LIN 模式下接收报头中断未发生（默认）
	1: LIN 模式下接收报头中断发生
Bit 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。	

21.5.12 USART0/1 屏蔽后中断状态寄存器 (USARTMIS)

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.	CEMIS	WUFMIS	LINMIS	Res.	OEMIS	BEMIS	PEMIS	FEMIS	RTMIS	TXMIS	RXMIS	Res.		CTSMIS	Res.
	r	r	r		r	r	r	r	r	r	r			r	

Bits 31:15	保留, 必须保持复位值
Bit 14	CEMIS: 冲突检测屏蔽中断状态位
	0: 冲突检测中断未发生或被屏蔽 (默认)
	1: 冲突检测中断未被屏蔽
Bit 13	WUFMIS: 低功耗模式下唤醒屏蔽事件状态位
	0: 低功耗模式下唤醒事件未发生或被屏蔽 (默认)
	1: 低功耗模式下唤醒事件未被屏蔽
Bit 12	LINMIS: LIN 模式下接收报头屏蔽中断状态位
	0: LIN 模式下接收报头中断未发生或被屏蔽 (默认)

	1: LIN 模式下接收报头中断未被屏蔽
Bit 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。	

21.5.13 USART0/1 中断清除寄存器 (USARTICR)

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.	CEIC	WUFIC	LINIC	Res.	OEIC	BEIC	PEIC	FEIC	RTIC	TXIC	RXIC	Res.		CTSIC	Res.
	w	w	w		w	w	w	w	w	w	w			w	

Bits 31:15	保留, 必须保持复位值
------------	-------------

Bit	14	CEIC: 冲突检测中断清除位
Bit	13	WUFIC: 低功耗模式下唤醒事件清除位
Bit	12	LINIC: LIN 模式下接收报头中断清除位
Bit	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。		

21.5.14 USART0/1 DMA 控制寄存器 (USARTDMACR)

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: 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, USARTRXDMASREQ 和 USARTRXDMABREQ 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, USARTRXDMASREQ 和 USARTRXDMABREQ 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

21.5.15 USART0/1 LIN 控制寄存器 1 (LINCON1)

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.							LIN SLP	LZEN	ABD	CSI	CSEN	SCM	Res.		MS
							rw	rw	rw	rw	rw	rw			rw

Bits 31:9	保留, 必须保持复位值
Bit 8	LINSLP: 配置 LIN 网络当前工作状态
	0: 活跃 (默认)
	1: 睡眠
Bit 7	LZEN: 总线长零检测使能
	检测总线 256 bit 时间 (16*Baud16) 的长低电平, 若溢出且中断使能, 发生中断。
	0: 禁用 (默认)
	1: 使能
Bit 6	ABD: 自动波特率同步使能
	该位在 LIN 从机模式下使能自动波特率同步, 主机模式下无效。如果自动波特率侦测关闭, 波特率的测量仍有效。当时钟计数超过上/下限阈值时, USARTRIS1.LARIS 置位。此外, 如果收到同步字节不是 0x55, 报同步错误。如果同步字节的停止位出错, 报帧错误。
	0: 禁用 (默认)
	1: 使能
Bit 5	CSI: checksum 存储选择
	该位决定是否将 LIN 帧中接收到的 checksum 字节写入 RX FIFO。
	0: 不写入 RXFIFO (默认)
	1: 写入 RXFIFO
	<i>注: 当 CSEN 不使能时, 强制写入 RXFIFO, 操作该位无效。</i>
Bit 4	CSEN: 硬件 checksum 使能
	使能该位, 硬件将生成 checksum 或检查收到的 checksum
	0: 关闭 (通过 FIFO 发送和接收实现) (默认)
	1: 开启
	CSEN=0 时, 通过 FIFO 实现发送和接收, checksum 的生成和校验由软件负责。
Bit 3	SCM: 软件控制模式
	与 LINEN 配合使用: LINEN=1, SCM=1: 软件实现 LIN 功能 LINEN=1, SCM=0: 硬件实现 LIN 功能

	0: 关闭（默认）
	1: 开启
Bits 2:1	保留，必须保持复位值
Bit 0	MS: LIN 主从模式选择
	0: 从机模式（默认）
	1: 主机模式
注： 长零检测与总线边沿检测 RED 的灵活使用。总线处于长零状态时，长零计数器计满 256 bit 时间，起长零中断，读取长零状态位并清除相应中断使能位。等待总线上升沿 RED 中断，可知总线长零状态退出。	

21.5.16 USART0/1 LIN 控制寄存器 2（LINCON2）

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.												WUTR	RDR	RTR	HTR
												rw	rw	rw	rw

Bits 31:4	保留，必须保持复位值
Bit 3	WUTR: LIN 唤醒发送请求（主/从）
	唤醒发送请求（仅写 1，写 0 无效），产生一个唤醒脉冲。当唤醒字符发送后，由硬件清 0。发送的数据写入 USARTDR，置位 WUTR 即可发送一定时间（250us~5ms）的低电平。写入的内容（连续 0 的位数）由波特率决定，通常是 0xF0/0xE0。
Bit 2	RDR: 应答中止请求（主/从）
	应答中止请求（仅写 1，写 0 无效），用于退出 LIN 应答状态，停止数据发送或接收。一旦 LIN 进入空闲状态，该位由硬件清 0。再次发起操作时，需注意清空 TX FIFO/RX FIFO。
Bit 1	RTR: 应答发送请求（主/从）
	数据发送请求（仅写 1，写 0 无效），请求发送存储在 TXFIFO 中的 LIN 数据字段。 主机模式下，在帧头发送请求之前，需提前把 LIN 数据字段写入 TX FIFO，然后置位 HTR 和 RTR。 从机模式下，仅当 USARTRIS1 中的 RHRIS 位置 1 时，才能设置该位。当应答发送完成或 RDR 置位时，硬件清零。 RTR 和 HO 不能同时写 1，HO 优先级高于 RTR。
Bit 0	HTR: 帧头发送请求（仅主机）
	帧头发送请求（仅写 1，写 0 无效）由软件设置来请求 LIN 帧头的发送。 当发送开始后，硬件清零。 HTR 置位前，需将数据提前写入 TX FIFO。写入内容如下。
	主机模式： 发帧头+发应答，PID + RESPONSE 提前写入 TXFIFO，HTR=1，RTR=1，HO = 0；

发帧头+收应答，PID 提前写入 TXFIFO，HTR=1，RTR=0，HO=0； 仅发帧头，PID 提前写入 TXFIFO，HTR=1，RTR=0，HO=1； RTR 和 HO 不能同时写 1，HO 优先级高于 RTR； 从机模式： 收到帧头后，根据 PID 信息发数据：将 RESPONSE 写入 TXFIFO，RTR=1，HO=0； 收到帧头后，根据 PID 信息收数据：RTR=0，HO=0； 收到帧头后，根据 PID 信息不应答：RTR=0，HO=1；LIN 状态机返回 idle，等待下一次 break 发起。 RTR 和 HO 不能同时写 1，HO 优先级高于 RTR。

21.5.17 USART0/1 帧控制寄存器（FRAMECON）

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

Bits 31:7	保留，必须保持复位值
Bit 6	CEN ：位错误检查使能
	0：关闭（默认）
	1：开启
Bits 5:3	LEAD[2:0] ：break 结束和同步字节开始之间的间隔长度
	000：1（默认）
	001：2
	...
	111：8
Bits 2:0	IDLE[2:0] ：传输字节的间隔长度，也用于帧头和应答的间隔长度
	000：0（默认）
	001：1
	...
	111：7
注： 使用超时检测时，需根据 LEAD，IDLE，DATLEN 计算定时时间。	

21.5.18 USART0/1 LIN 状态寄存器（LINSTS）

地址偏移：0x94

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.											STS[3:0]			PS	
											r			r	

Bits 31:5	保留，必须保持复位值
Bits 4:1	STS[3:0]: LIN 模式状态
	0000: Idle 当发生如下几个事件时，进入该状态 – USARTEN 关闭或 LINEN 关闭或 SCM 置位（软件实现 LIN） – 位错误发生，即发送的数据和接收的数据不一致 – 帧头发送或接收完毕，RDR 置位或 HO 置位 – 软件置位 LINSLP，进入睡眠模式 – 唤醒命令 WUTR 发送完成 – 前面的帧接收/传输完成
	0001: Break 在从机模式下，检测到一个中断长度配置（BREAK 长度低电平）的有效打断,等待一个上升沿进入 Break Delimiter。 在主机模式下，不断进行 Break 传输。
	0010: Break Delimiter 在从机模式下，已识别到 Break，等待一个 LEAD 位时间进入 Synch Field。 在主机模式下，已完成中断传输，正在进行 Break Delimiter 传输。 Break 检测有效时返回 Break 状态。
	0011: Synch Field 在从机模式下，接收 Synch Field，完成波特率检测。 在主机模式下，正在进行 Synch Field 传输。 Break 检测有效时返回 Break 状态。
	0100: Parity Identifier Field 在从机模式下，已经完成有效 Synch Field 的接收，正在接收 Identifier Field。 在主机模式下，正在进行标识符传输。 Break 检测有效时返回 Break 状态。
	0101: Header reception/transmission completed 在从机模式下，已经接收一个有效的帧头，标识符字段可用。 在主机模式下，完成帧头发送。 Break 检测有效时返回 Break 状态。
	0110: Data reception/transmission Field 正在进行应答接收/传输。 Break 检测有效时返回 Break 状态。
	0111: Checksum 已完成数据接收/传输，正在进行校验和接收/传输 Break 检测有效时返回 Break 状态。

	1111: Wake up transmission field 正在进行唤醒信号传输。
	其他值保留
Bit 0	PS: LIN 接收引脚电平状态
	0: 低电平（默认）
	1: 高电平

21.5.19 USART0/1 数据配置寄存器（DATCON）

地址偏移: 0x98

复位值: 0x0000_FF00

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
RESPONSE[7:0]								Res.		CSM	RM	HO	DATLEN[2:0]		
rw										rw	rw	rw	rw		

Bits 31:16	保留，必须保持复位值
Bits 15:11	RESPONSE[15:8]: 应答/帧超时阈值（默认 0xFF）
	帧或应答超时阈值取决于应答数据的长度（1 到 8 字节）。作为从机时，根据收到的 PID 进行设置。
Bits 7:6	保留，必须保持复位值
Bit 5	CSM: 校验和类型
	0: 经典校验，仅包括数据字段（默认）
	1: 增强校验，包括标识符和数据字段
Bit 4	RM: 定义 RESPONSE 的用途
	0: Frame 超时阈值（默认）
	1: Response 超时阈值
Bit 3	HO: Header Only，定义 LIN 帧是由报头和应答组成，还是仅由报头组成
	0: 报头 + 应答（默认）
	1: 只有报头，应答被忽略
	在应答状态下，HO 使 LIN 状态机回到空闲状态，优先级高于 RTR，且不能和 RTR 同时置位。
Bits 2:0	DATLEN[2:0]: 应答数据长度
	该字段定义帧的应答部分中的数据字节数。取值范围 0-7 表示 1-8 个字节长度，主/从机都需要配置该寄存器。从机在收到帧头后根据 PID 设置。
	000: 1（默认）
	001: 2
	...
	111: 8

注：HO 和 RTR 同时置 1 时无效（软件保证不同时置位），HO 的优先级比 RTR 高。做主机时，HO/RTR 必须在初始化配置好，帧传输过程不得修改。从机时，从机根据收到的 PID 决定是否置位 HO 和 RTR。

21.5.20 USART0/1 波特率侦测寄存器（BRDMSU）

地址偏移：0x9C

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

复位值：0x0000_0000

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

Bits	31:16	保留，必须保持复位值
Bits	15:0	MEASURED[15:0] ：波特率测量时钟数，复位后寄存器清 0
		该位包含 LIN 帧中同步字节的时钟测量值，在 LIN 从机波特率自校准（ABD=1）的从机模式，该寄存器自动载入整数波特率寄存器 USARTIBRD 和小数波特率寄存器 USARTFBRD，产生新的波特率，以相同的速率接收主机发送的数据，实现与主机的同步。
注：从机开启自动波特率同步时，LIN 工作频率选择 1~16Mhz，受限于该寄存器的计数位数。		

21.5.21 USART0/1 波特率侦测上限寄存器（BRDUPPER）

地址偏移：0xA0

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
UPPERLIMIT[15:0]															
rw															

Bits	31:16	保留，必须保持复位值
Bits	15:0	UPPERLIMIT[15:0] ：波特率测量时钟数上限
注：从机开启自动波特率同步时，LIN 工作频率选择 1~16Mhz，受限于该寄存器的计数位数。		

21.5.22 USART0/1 波特率侦测下限寄存器（BRDLOWER）

地址偏移：0xA4

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
LOWERLIMIT[15:0]															
rw															

Bits	31:16	保留，必须保持复位值
Bits	15:0	LOWERLIMIT[15:0] ：波特率测量时钟数下限
注： 从机开启自动波特率同步时，LIN 工作频率选择 1~16Mhz，受限于此寄存器的计数位数。		

21.5.23 USART0/1 LIN BREAK 定时寄存器（LINBTIMER）

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.										BREAK[5:0]					
										rw					

Bits	31:6	保留，必须保持复位值
Bits	5:0	BREAK[5:0] ：生成和侦测 break 脉冲（默认值 0xD）
		LIN 从机模式，该位域定义 break 脉冲侦测阈值。 LIN 主机模式，该位域定义 break 脉冲的持续时间。 以 bit 为时间单位（16*Baud），做主机时，典型值为 13；做从机时，典型值为 11。

21.5.24 USART0/1 LIN 帧头定时寄存器（LINHTIMER）

地址偏移：0xAC

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.								HEADER[7:0]							
								rw							

Bits	31:8	保留，必须保持复位值
Bits	5:0	HEADER: 帧头超时阈值（默认 0xFF）
		定义 1 到 256 位时间范围内的计时限制。

21.5.25 USART0/1 中断屏蔽寄存器 1（USARTIMSC1）

地址偏移：0xB0

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.			SFIM	LZIM	REDIM	FEDIM	LPIM	LCIM	LAIM	RTIM	HTIM	TRIM	THIM	RRIM	RHIM
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	31:13	保留，必须保持复位值
Bit	12	SFIM: LIN 同步段错误中断屏蔽位
Bit	11	LZIM: LIN 长零中断屏蔽位
Bit	10	REDIM: LINRX 上升沿侦测中断状态位
Bit	9	FEDIM: LINRX 下降沿侦测中断状态位
Bit	8	LPIM: LIN 标识符奇偶校验中断屏蔽位（仅用于 LIN 模式，PID 字节验）
Bit	7	LCIM: LIN 校验和错误（checksum）中断屏蔽位
Bit	6	LAIM: LIN 自动波特率侦测错误中断屏蔽位
Bit	5	RTIM: LIN 应答超时中断屏蔽位
Bit	4	HTIM: LIN 帧头超时中断屏蔽位
Bit	3	TRIM: LIN 应答发送中断屏蔽位
Bit	2	THIM: LIN 帧头发送中断屏蔽位
Bit	1	RRIM: LIN 应答接收中断屏蔽位
Bit	0	RHIM: LIN 帧头接收中断屏蔽位
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。寄存器写 0 清除，写 1 置位。		

21.5.26 USART0/1 原始中断状态寄存器 1（USARTRIS1）

地址偏移：0xB4

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.			SFRIS	LZ RIS	RED RIS	FED RIS	LPRIS	LCRIS	LARIS	RTRIS	HTRIS	TRRIS	THRIS	RRRIS	RHRIS
			r	r	r	r	r	r	r	r	r	r	r	r	r

Bits	31:13	保留，必须保持复位值
Bit	12	SFRIS: LIN 同步段错误中断状态位
Bit	11	LZ RIS: LIN 长零中断状态位
Bit	10	RED RIS: LINRX 上升沿侦测中断状态位
Bit	9	FED RIS: LINRX 下降沿侦测中断状态位
Bit	8	LPRIS: LIN 标识符奇偶校验中断状态位（仅用于 LIN 模式，PID 字节验）
Bit	7	LCRIS: LIN 校验和错误（checksum）中断状态位
Bit	6	LARIS: LIN 自动波特率侦测错误中断状态位
Bit	5	RTRIS: LIN 应答超时中断状态位
Bit	4	HTRIS: LIN 帧头超时中断状态位
Bit	3	TRRIS: LIN 应答发送中断状态位
Bit	2	THRIS: LIN 帧头发送中断状态位
Bit	1	RRRIS: LIN 应答接收中断状态位
Bit	0	RHRIS: LIN 帧头接收中断状态位
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。		

21.5.27 USART0/1 屏蔽中断状态寄存器 1（USARTMIS1）

地址偏移：0xB8

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.			SFIMIS	LZMIS	REDMIS	FEDMIS	RRMIS	RHMIS	TRMIS	THMIS	RRMIS	TRMIS	THMIS	RRMIS	RHMIS
			r	r	r	r	r	r	r	r	r	r	r	r	r

Bits	31:13	保留，必须保持复位值
Bit	12	SFIMIS: LIN 同步段错误中断屏蔽位
Bit	11	LZMIS: LIN 长零屏蔽中断状态位
Bit	10	REDMIS: LINRX 上升沿侦测中断状态位
Bit	9	FEDMIS: LINRX 下降沿侦测中断状态位
Bit	8	LPMIS: LIN 标识符奇偶校验屏蔽中断状态位（仅用于 LIN 模式，PID 字节校验）
Bit	7	LCMIS: LIN 校验和错误（checksum）屏蔽中断状态位
Bit	6	LAMIS: LIN 自动波特率侦测错误屏蔽中断状态位
Bit	5	RTMIS: LIN 应答超时屏蔽中断状态位

Bit 4	HTMIS: LIN 帧头超时屏蔽中断状态位
Bit 3	TRMIS: LIN 应答发送屏蔽中断状态位
Bit 2	THMIS: LIN 帧头发送屏蔽中断状态位
Bit 1	RRMIS: LIN 应答接收屏蔽中断状态位
Bit 0	RHMIS: LIN 帧头接收屏蔽中断状态位
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。	

21.5.28 USART0/1 中断清除寄存器 1 (USARTICR1)

地址偏移 : 0xBC

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.			SFIC	LZIC	REDIC	FEDIC	LPIC	LCIC	LAIC	RTIC	HTIC	TRIC	THIC	RRIC	RHIC
			w	w	w	w	w	w	w	w	w	w	w	w	w

Bits 31:13	保留, 必须保持复位值
Bit 12	SFIC: LIN 同步段错误中断清除位
Bit 11	LZIC: LIN 长零中断清除位
Bit 10	REDIC: LINRX 上升沿侦测中断清除位
Bit 9	FEDIC: LINRX 下降沿侦测中断清除位
Bit 8	LPIC: LIN 标识符奇偶校验中断清除位 (仅用于 LIN 模式, PID 字节校验)
Bit 7	LCIC: LIN 校验和错误 (checksum) 中断清除位
Bit 6	LAIC: LIN 自动波特率侦测错误中断清除位
Bit 5	RTIC: LIN 应答超时中断清除位
Bit 4	HTIC: LIN 帧头超时中断清除位
Bit 3	TRIC: LIN 应答发送中断清除位
Bit 2	THIC: LIN 帧头发送中断清除位
Bit 1	RRIC: LIN 应答接收中断清除位
Bit 0	RHIC: LIN 帧头接收中断清除位
注: 该寄存器只能进行写操作, 写1清中断标志位, 写0无效。	

22. SPI 接口

22.1 SPI 主要特性

- 支持主机和从机模式
- 支持普通模式和高速模式，主机模式下通讯速率最高可达 1/2 工作时钟，从机模式下最高速率可达 24Mbps
- 支持 Motorola SPI 接口协议
- 帧长度 4 到 16 位可配
- 独立的工作时钟，高达 48MHz
- 可编程的预分频系数和传输比特率
- 支持内部回环测试模式
- 独立的 4x16bit 发送和接收 FIFO
- 支持 DMA 操作

22.2 管脚配置

表 22-1 SPI 管脚配置

管脚名称	管脚类型	管脚描述	复用 I/O 口	备注
SPIO_FSS	I/O	SPI 片选管脚	PA4/PA15	-
SPIO_SCK	I/O	SPI 时钟管脚	PA5/PB3	-
SPIO_RXD	I	SPI 接收管脚	PA2/PA3/PA6/PA7/PB4/PB5	-
SPIO_TXD	O	SPI 发送管脚	PA2/PA3/PA6/PA7/PB4/PB5	-

22.3 功能描述

SPI (Serial Peripheral Interface) 是同步串行接口，可作为主设备或从设备，允许与 Motorola SPI 协议的主从设备进行同步串行通信。图 22-1 展示了 SPI 的主要功能组成。

SPI 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，SPI 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。

SPI 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 4 个 16 位值。

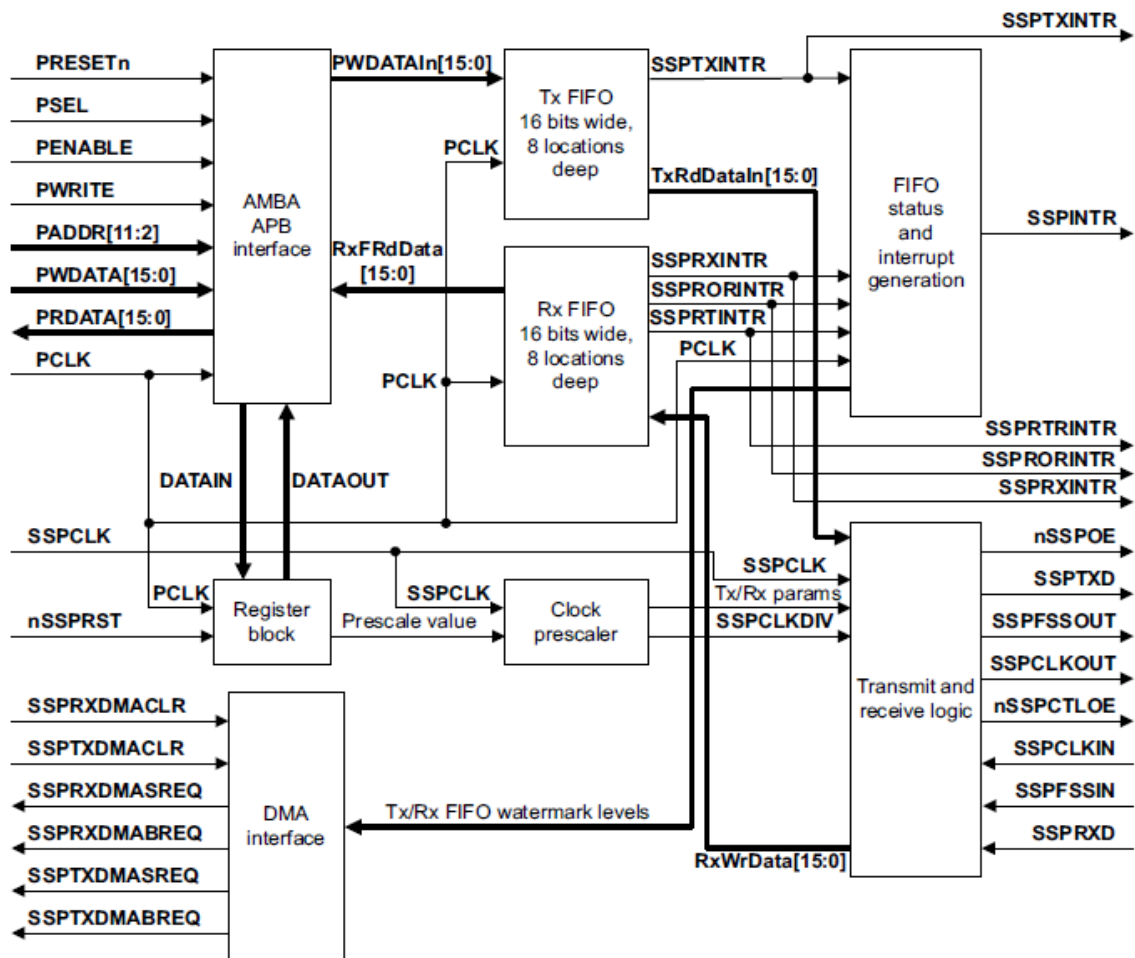


图 22-1 SPI 结构框图

SPI 包含一个可编程比特率时钟分频器和预分频器，输入时钟 $SSPCLK$ 通过分频器产生串行输出时钟 $SSPCLKOUT$ ，支持比特率 24MHz 或更高频率，实际中最大传输比特率会受到外围设备的影响。

用户可用控制寄存器 $SPICR0$ 和 $SPICR1$ 改变操作模式，帧格式和帧长度。

22.4 SPI 操作

22.4.1 串行时钟的生成

串行时钟比特率是输入时钟 $SSPCLK$ 经过分频而来。该时钟先经过偶数分频值 $CPSDVSR$ （范围 2~254）分频后的时钟再次经过 $1+SCR$ （范围 0~255）分频， SCR 在 $SPICR0$ 寄存器中配置。

下面等式定义了输出信号位时钟 $SSPCLKOUT$ 频率：

$$F_{SSPCLKOUT} = \frac{F_{SSPCLK}}{CPSDVSR \times (1 + SCR)}$$

例如， F_{SSPCLK} 为 3.6864Mhz， $CPSDVSR=2$ ， $F_{SSPCLKOUT}$ 的频率范围为 7.2KHZ ~1.8432MHZ。

22.4.2 时钟频率与模式选择

22.4.2.1 普通模式

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，检测 SSPCLKIN 边沿至少需要 3 个 SSPCLK。为确保设备的正确操作，从机的 SSPCLK 频率需要比 SSPCLKIN 的最高期望频率快 12 倍。如果 SSPCLK 为 48MHz，普通模式下串行时钟速率 SSPCLKIN 最高为 4MHz，即从机最高速率为 4MHz。如果需要从机达到更高的速率，则要配置 SPICR2.HR=1 切换到高速模式进行工作。

22.4.2.2 高速模式

在主机模式时，时钟限制依旧是 SSPCLKOUT 的最高频率不高于 SSPCLK 的 1/2。此时由于从机没有时钟限制，因此在 SSPCLK 为 48MHz 的情况下，主机可以配置两个与 SSPCLKOUT 频率有关的寄存器：SPICR0.SCR=0x0，SPICPSR.CPSDVSR=0x2，通过配置这两个寄存器可以实现 $SSPCLKOUT = 1/2 SSPCLK$ ，理论上串行时钟频率最快可以达到 24MHz。在实际传输中，由于时钟是主机自己产生的，而数据是通过线上接收来自从机发送的，该过程中线上会存在一定的时间延迟，而串行速率越快，时间延迟对主机采样造成的影响就越大，因此需要配置 SPICR2.MRXDDS 寄存器来选择主机延迟 0~30ns 采样，以此来补偿时间延迟对主机采样带来的影响。

在从机模式时，时钟来自片外主机的 SSPCLKIN，理论上 SSPCLKIN 的速率与从机 SSPCLK 无关。高速模式下串行时钟速率 SSPCLKIN 最高为 24MHz，即从机最高速率为 24MHz。

22.4.3 配置 SPICR0 控制寄存器

SPICR0 寄存器用于：

- 配置串行时钟速率
- 配置 SSPCLKOUT 相位和极性
- 选择帧数据长度

串行时钟速率（SCR）值，与时钟预分频寄存器 SPICPSR 一起使用，用于从 SPI 工作时钟 SSPCLK 分频得到 SPI 发送和接收比特率。

SSPCLKOUT 相位和极性通过 SPH 位和 SPO 位选择，数据长度通过 DSS 位配置。

22.4.4 配置 SPICR1 控制寄存器

SPICR1 寄存器用于：

- 主机从机选择
- 使能回环测试
- 使能 SPI

要将 SPI 配置为主机模式，设置 SPICR1 寄存器主从机选择位 MS 为 0，即复位值。

要将 SPI 配置为从机模式，设置 SPICR1 寄存器主从机选择位 MS 为 1。当 SPI 配置为从机时，SSPTXD 信号使能与否取决于 SPICR1 的 SOD 位。这可用在并行广播的多从机环境。

要使能 SPI，设置 SPICR1 寄存器 SSE 为 1。

22.4.5 配置 SPICR2 控制寄存器

SPICR2 寄存器用于：



- 采样延迟时间配置
- 配置为高速模式或普通模式

在高速通信下，由于传输存在延迟，主机通常需要推迟采样时刻，以便采样到正确的数据。为了正确通讯，配置 SPICR2 寄存器的主机模式采样延迟时间选择位 MRXDDS 位为适当值。

为了匹配高速通信下，数据、片选、时钟线间可能存在的延迟时间差，配置 SPICR2 寄存器的从机模式延迟使能位 SCLKD、SFSSD、SRXDD 位。

22.4.6 SPI 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。支持摩托罗拉 SPI 帧格式。SPI 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。对于摩托罗拉 SPI 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。

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 SPICR0 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。

22.4.6.1 SPO = 0, SPH = 0

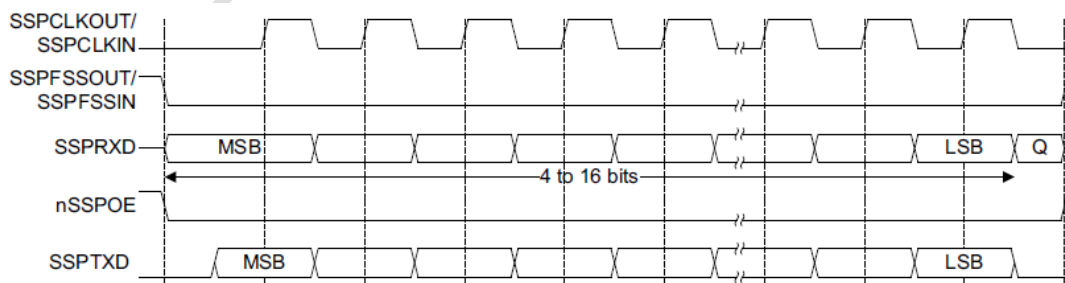


图 22-2 SPO=0, SPH=0, 单字节传输时序

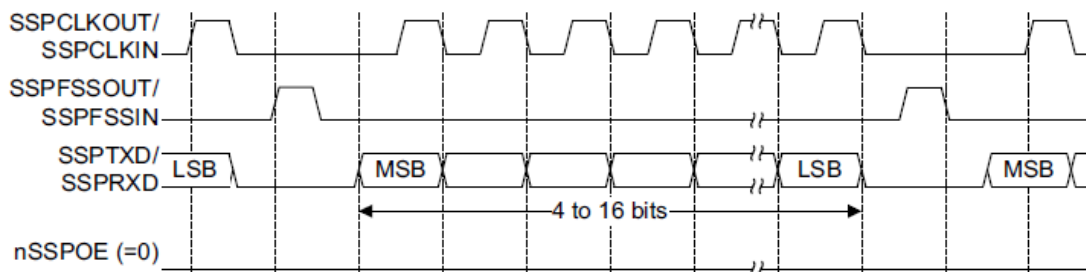


图 22-3 SPO=0, SPH=0, 多字节传输时序

在此配置下：

- 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 SPI 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 SPI 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 SPI 使能，且发送 FIFO 内含有效数据，SSPFSSOUT 主信号被驱动为低电平表示传输开始。这使能从机的 SSPRXD 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。

22.4.6.2 SPO = 0, SPH = 1

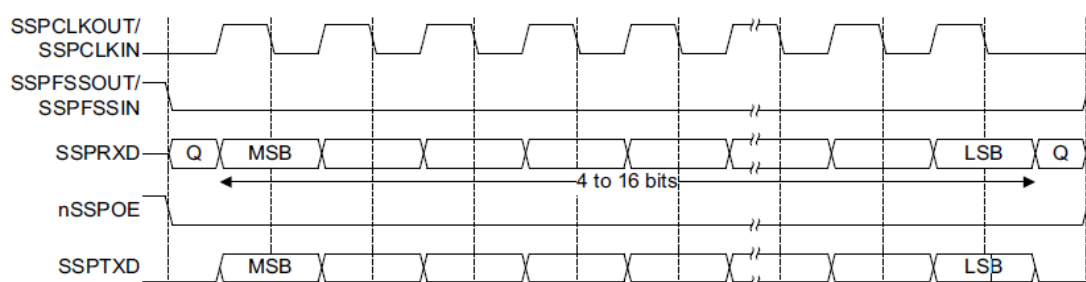


图 22-4 SPO=0, SPH=1, 包括单次和连续传输

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 SPI 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 SPI 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

如果使能了 SPI，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 引脚在连续之间保持低电平，终止与单字传输的相同。

22.4.6.3 SPO =1, SPH =0

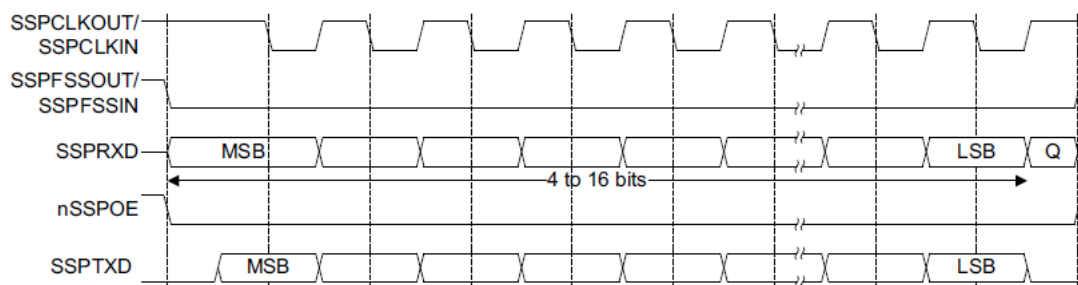


图 22-5 SPO=1, SPH=0 单字节传输

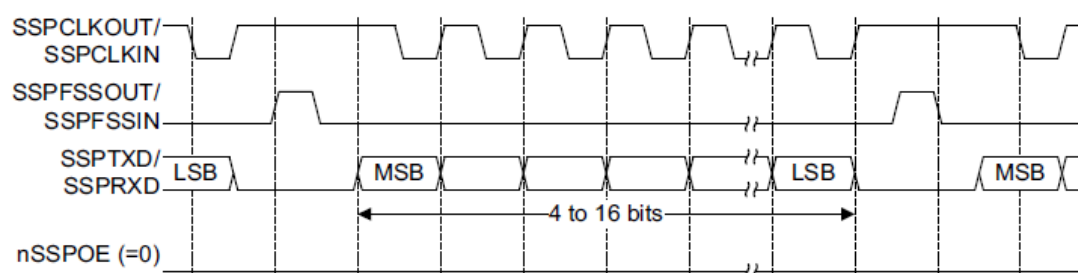


图 22-6 SPO=1, SPH=0 多字节传输

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 SPI 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 SPI 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 SPI，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。如果 SPH 位为 0，则串行外设寄存器中的数据不允许更改。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。

22.4.6.4 SPO =1, SPH =1

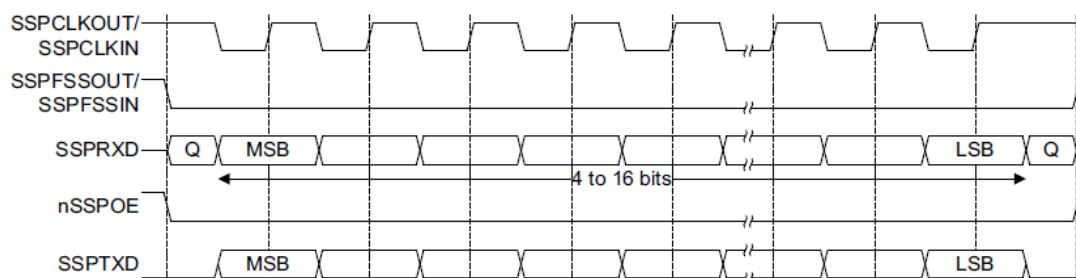


图 22-7 SPO=1, SPH=1, 包括单次和连续传输

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 SPI，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，然后返回其空闲状态为上一节描述。对于连续的对背传输，SSPFSSOUT 引脚在连续之间保持低电平，数据字和终止与单字传输的相同。

22.4.7 主从配置实例

图 22-8 显示 SPI 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 SSPTXD。

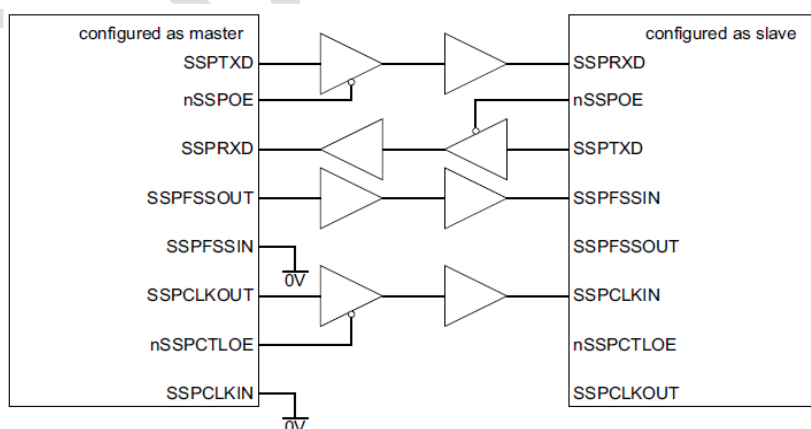


图 22-8 SPI 主机与 SPI 从机

图 22-9 显示 SPI 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 SSPTXD 访问从机，从机驱动 SPI MISO 端口到主机的 SSPTXD 线。

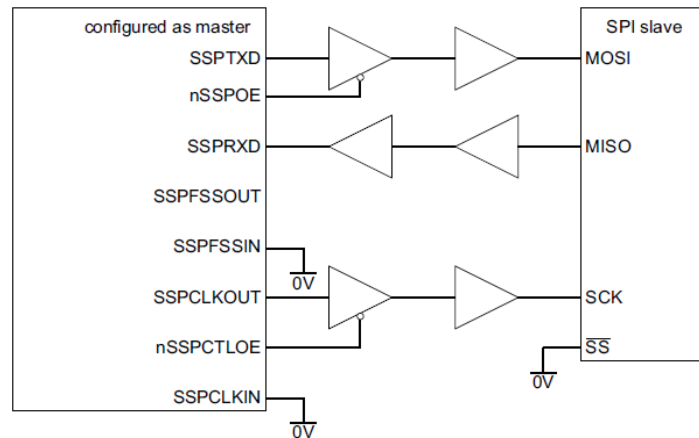


图 22-9 SPI 主机和外部 SPI 从机

图 22-10 显示外部 SPI 配置成主机，与配成从机的 SPI 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

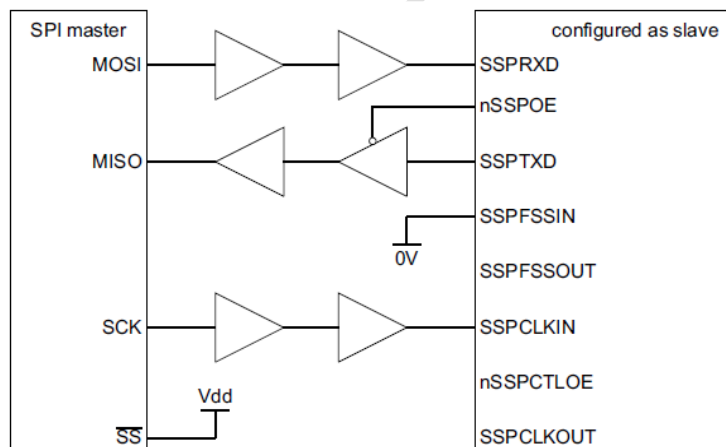


图 22-10 外部 SPI 主机与 SPI 从机

22.4.8 DMA 接口

SPI 提供一个接口，连接到 DMA 控制器。DMA 控制寄存器控制 SPI 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）：

- SPIRXDMASREQ（REQ 信号）：DMA 单次传输请求，由 SPI 置位。当接收 FIFO 包含至少 1 个数据该信号置位。
- SPIRXDMABREQ（REQ 信号）：DMA 连续传输请求，由 SPI 置位。当接收 FIFO 包含等于 4 个数据时该信号置位。
- SPIRXDMACLR（ACK 信号）：DMA 请求清除，DMA 控制器置位用于清除 SPI 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：

- SPITXDMASREQ（REQ 信号）：DMA 单次传输请求，由 SPI 置位。在发送 FIFO 至少含 1 个空位置时该

信号置位。

- SPITXDMABREQ (REQ 信号)：DMA 连续传输请求，由 SPI 置位。当发送 FIFO 包含 0 个数据时该信号置位。
- SPITXDMACLR (ACK 信号)：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 22-2 SSP DMABREQ 的触发点，用于发送和接收 FIFO

发送 (FIFO 空闲数据数)	接收 (FIFO 包含数据数)
4	4

连续传输请求和单次传输请求信号并不互斥，可同时置位。例如，当接收 FIFO 中数据的个数为 4 时，这两个请求信号均置位。当接收 FIFO 内包含的数据少于 4 个时，只置位单次请求信号。

例如须要收 19 个字节，DMA 控制器发起 4 次 4 字节长度的连续请求，3 次单次请求，完成整个任务。对于最后 3 个字节，SPI 不会置位 DMA 连续请求信号。

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。如果 SPI 使能禁止或 DMA 使能禁止，所有请求信号都无效。

图 22-11 显示了单次传输请求和连续传输请求的时序，和相应的 DMA 清除信号，所有信号都与 PCLK0 同步。

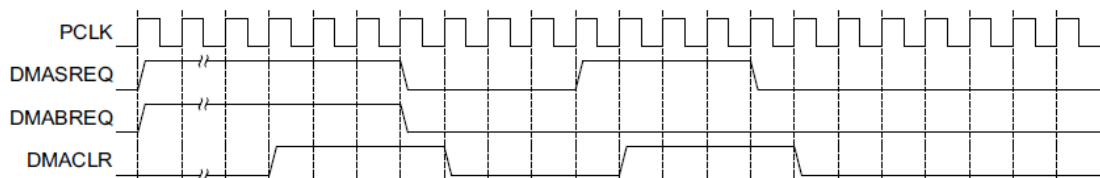


图 22-11 DMA 传输波形

22.5 SPI 中断

SPI 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，对应着系统中的 SPI 全局中断向量。

- SPIRXINTR SPI 接收中断
- SPITXINTR SPI 发送中断
- SPIRORINTR SPI 接收 FIFO 溢出中断
- SPIRTINTR SPI 接收 FIFO 超时中断

用户可以通过设置 SPIIMSC 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 SPITXINTR 和 SPIRXINTR 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 SPIRIS 和 SPIMIS 寄存器中读取各个中断源的状态。

- SPIRXINTR

当接收 FIFO 中有 2 个或更多有效数据时，接收中断被置位。

- SPITXINTR

当发送 FIFO 中有 2 个或更少的有效数据时，发送中断被置位。发送中断 SPITXINTR 因 SPI 使能信号无效时，可以通过以下任一方式进行操作：

- 在使能 SPI 和中断前，可以将数据写入发送 FIFO 中。
- 可以启用使能 SPI 和中断，以便可以在中断服务程序将数据写入发送 FIFO 中。

- SPIRORINTR

当 FIFO 已满，且又收到额外的一帧数据，产生接收溢出中断 SPIRORINTR，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 SPIICR 寄存器中的 RORIC 位可以清除该中断。

● SPIRTINTR

当接收 FIFO 非空且 SPI 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 SPIRXD 上接收到新数据，该中断清除。此外，通过写入 SPIICR 寄存器中的 RTIC 位也可以清除该中断。

● SPIINTR

以上四个独立中断组合成单个输出 SPIINTR，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 SPI 中断被置位。

22.6 SPI 寄存器描述

SPIO 的基地址为 0x4001_8000，地址偏移对应其基地址。

表 22-3 SPI 相关寄存器表

名称	说明	读写权限	复位值	地址偏移
SPICR0	SPI 控制寄存器 0	R/W	0x0000_0000	0x00
SPICR1	SPI 控制寄存器 1	R/W	0x0000_0000	0x04
SPICR2	SPI 控制寄存器 2	R/W	0x0000_0001	0x28
SPIDR	SPI 数据寄存器	R/W	0x0000_0000	0x08
SPISR	SPI 状态寄存器	R	0x0000_0003	0x0C
SPICPSR	SPI 时钟分频寄存器	R/W	0x0000_0000	0x10
SPIIMSC	SPI 中断屏蔽寄存器	R/W	0x0000_0000	0x14
SPIRIS	SPI 原始中断状态寄存器	R	0x0000_0000	0x18
SPIMIS	SPI 屏蔽后中断状态寄存器	R	0x0000_0000	0x1C
SPIICR	SPI 中断清除寄存器	W	0x0000_0000	0x20
SPIDMACR	SPIDMA 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）。

22.6.1 SPI 控制寄存器 0（SPICR0）

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	Res.		DSS[3:0]			
rw								rw	rw			rw			

Bits 31:16	保留，必须保持复位值
Bits 15:8	SCR: 串行时钟速率（默认为 0）
	SCR 用于生成传输并接收 SPI 的比特率，比特率是：
	$\frac{F_{ssplk}}{CPSDVR \times (1 + SCR)}$



	其中 CPSDVSr 是 2-254 的偶数值，通过 SPICPSR 寄存器编程，SCR 是 0-255 的值
Bit 7	SPH: SSPCLKOUT 相位
	0: SPI 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: SPI 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
	0: SPI 控制器使总线时钟在每帧传输之间保持低电平（默认）
	1: SPI 控制器使总线时钟在每帧传输之间保持高电平
Bits 5:4	保留，必须保持复位值
Bits 3:0	DSS: 数据长度选择，该字段控制着每帧传输的位数目
	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

22.6.2 SPI 控制寄存器 1（SPICR1）

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 SPI 控制器驱动发送数据线 TXD（默认）
	1: 禁止 SPI 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 SPI 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，SPI 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: SPI 操作禁止 (默认)
	1: SPI 操作使能
	SPI 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 SPI 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)

22.6.3 SPI 控制寄存器 2 (SPICR2)

地址偏移: 0x28

复位值: 0x0000_0001

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.								LSB	SRXDD	SFSSD	SCLKD	MRXDDS	Res.	HR	
								rw	rw	rw	rw	rw			rw

Bits 31:8	保留, 必须保持复位值
Bit 7	LSB: 发送与接收的数据 MSB 优先还是 LSB 优先
	0: MSB 优先 (默认)
	1: LSB 优先
Bit 6	SRXDD: 从机模式延迟接收数据使能, 该位只与从机模式有关 (MS=1), 只在高速模式下生效
	0: SPI 控制器直接接收数据 (默认)
	1: SPI 控制器延迟 5ns 后接收数据
Bit 5	SFSSD: 从机模式延迟接收片选使能, 该位只与从机模式有关 (MS=1), 只在高速模式下生效
	0: SPI 控制器直接接收片选 (默认)
	1: SPI 控制器延迟 5ns 后接收片选
Bit 4	SCLKD: 从机模式延迟接收时钟使能, 该位只与从机模式有关 (MS=1), 只在高速模式下生效
	0: SPI 控制器直接接收时钟 (默认)
	1: SPI 控制器延迟 5ns 后接收时钟
	SPI 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 SPI 寄存器和中断寄存器
Bit 3:2	MRXDDS: 主机模式采样延迟时间选择, 该位只与主机模式有关 (MS=0), 只在高速模式下生效
	00: SPI 控制器直接接收数据 (默认)
	01: SPI 控制器延迟 10ns 后接收数据

	10: SPI 控制器延迟 20ns 后接收数据
	11: SPI 控制器延迟 30ns 后接收数据
Bit 1	保留, 必须保持复位值
Bit 0	HR: SPI 高速模式配置
	0: 普通模式
	1: 高速模式 (默认)

22.6.4 SPI 数据寄存器 (SPIDR)

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 SPI 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, SPI 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

22.6.5 SPI 状态寄存器 (SPISR)

地址偏移: 0x0C

复位值: 0x0000_0003

SPISR 是只读状态寄存器, 包含指示 FIFO 填充的位状态和 SPI 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: SPI 忙标志



	0: SPI 为空闲状态（默认）
	1: SPI 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）

22.6.6 SPI 时钟分频寄存器（SPICPSR）

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 : 时钟分频寄存器（默认为 0）
	这是 2~254 中的一个偶数值。它是 SSPCLK 的分频因子，SSPCLK 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 SSICPSR 值，否则 SPI 控制器将无法正确发送数据。	
- 在从机模式下，当 SPICR2.HR=0 普通模式时，主机提供的 SPI 时钟速率不能大于 SPI 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）。	

22.6.7 SPI 中断屏蔽寄存器（SPIIMSC）

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

22.6.8 SPI 原始中断状态寄存器（SPIRIS）

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

22.6.9 SPI 屏蔽后中断状态寄存器（SPIMIS）

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

22.6.10 SPI 中断清除寄存器（SPIICR）

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。	

22.6.11 SPI DMA 控制寄存器（SPIDMACR）

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

23. CRC 模块

23.1 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 0x04C1_1DB7 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

23.2 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。

23.3 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 23-1 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）。

23.3.1 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]}
		11: 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

23.3.2 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[31:0]															
rw															

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

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

23.3.3 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

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

24. DIV 除法器

24.1 DIV 主要特性

- 支持 32/16 的有符号操作，8 个周期完成
- 向下支持 16/16 的有符号除法
- 支持除 0 中断

24.2 功能描述

将数据写入除数寄存器 DIV_B 后，DIV 开始计算，在 8 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 第 0 位），完成除法器逻辑复位
2. 配置 DIV32 位（DIV_CTRL 第 8 位），选择被除数的位宽 32/16
3. 将被除数按字写入 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 8 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

24.3 DIV 中断

DIV 支持可屏蔽的除 0 中断，当 DIV_CTRL.DIVO_INTREN 使能且发生除 0 操作时会产生 DIV 中断，对应着系统中的 DIV/CORDIC 中断向量。

24.4 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_5000。

表 24-1 DIV 相关寄存器表

名称	说明	读写权限	复位值	字节地址
DIV_CTRL	DIV 控制寄存器	R/W	0x0000_0000	0x4801_5000
DIV_ALO	DIV 被除数寄存器	R/W	0x0000_0000	0x4801_5004
DIV_B	DIV 除数寄存器	R/W	0x0000_0001	0x4801_500C
DIV_QUOT	DIV 商寄存器	R	0x0000_0000	0x4801_5010
DIV_REM	DIV 余数寄存器	R	0x0000_0000	0x4801_5018

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 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.						DIVO_INTREN	DIVO	Res.							
						rw	r								

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



Res.							DIV16	Res.			DIV_B _WRITED	Res.	DIV_ALO_WRITED	DIV_IN_ PROCEED	DIV_ RST
							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 时不进除 0 中断（默认）
		1: 除数为 0 时允许进除 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	DIV16 : 被除数长度配置
		0: 被除数为 32 位（默认）
		1: 被除数为 16 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	保留，必须保持复位值
Bit	2	DIV_ALO_WRITED : 被除数写入标志位，写 1 清除
		0: 被除数寄存器（DIV_ALO）未写入（默认）
		1: 被除数寄存器（DIV_ALO）已写入
Bit	1	DIV_IN_PROCEED : 除法器被使用标志位，写 1 清除
		0: 除法器未被使用过（默认）
		1: 除法器被使用过（B/ALO 任一寄存器写入）
		一般用作软件标识位，防止除法器工作过程中有中断插入，导致无法识别除法器状态
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]

24.4.2 DIV 被除数寄存器（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[31:0]															
rw															

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



Bits	31:0	DIV_ALO[31:0]: 被除数寄存器
------	------	-----------------------

24.4.3 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[31:0]															
rw															

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

Bits	31:0	DIV_B[31:0]: 除数寄存器
------	------	--------------------

24.4.4 DIV 商寄存器 (DIV_QUOT)

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_QUOT[31:0]															
r															

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

Bits	31:0	DIV_QUOT[31:0]: 商寄存器
------	------	----------------------

24.4.5 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[31:0]															
r															

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

Bits	31:0	DIV_REM[31:0]: 余数寄存器
------	------	----------------------



25. CORDIC 模块

CORDIC (Coordinate Rotation Digital Computer) 是坐标旋转数字算法的简称, 主要用于解决导航系统中三角函数、反三角函数和开方等运算的实时计算问题。CORDIC 算法应用广泛, 如离散傅里叶变换、离散余弦变换、离散 Hartley 变换、Chirp-Z 变换、各种滤波以及矩阵的奇异值分解中都可应用 CORDIC 算法。从广义上讲, CORDIC 算法提供了一种数学计算的逼近方法。

25.1 CORDIC 主要特性

- 支持圆坐标系和双曲线坐标系
- 支持旋转模式和向量模式
- 运算迭代次数可配, 在精度和运算时间之间做权衡
- 支持运算结果直接作为输入循环计算

25.2 功能描述

CORDIC 算法的本质是将三角函数、反三角函数和开方等运算分解为一些列的加减和移位运算, 具体的算法不在此处赘述, 有兴趣了解的用户请自行查找。这里直接给出 CORDIC 算法中圆坐标系和双曲线坐标系的统一形式:

$$\begin{cases} x_{i+1} = x_i - \mu d_i(2^{-i}y_i) \\ y_{i+1} = y_i + d_i(2^{-i}x_i) \\ z_{i+1} = z_i - d_i\theta_i \end{cases}$$

旋转系统	参数	补偿因子 A_n ($n \rightarrow \infty$)	迭代顺序
圆周系统	$\mu = 1, \theta_i = \tan^{-1}2^{-i}$	1.64676	0, 1, 2, ..., n-1
双曲系统	$\mu = -1, \theta_i = \tanh^{-1}2^{-i}$	0.82815936	1, 2, 3, ..., n

CORDIC 算法对应到具体函数的计算中, 如下表所示。

表 25-1 CORDIC 旋转系统详细结果

旋转系统	工作模式	最终旋转结果	初始化	可直接求取	可间接求取	备注
圆周系统	旋转	$x_n = A_n(x_0 \cos z_0 - y_0 \sin z_0)$ $y_n = A_n(y_0 \cos z_0 + x_0 \sin z_0)$ $z_n = 0$	$x_0 = 1/A_n$ $y_0 = 0$ $z_0 = \theta$	$\cos \theta$ $\sin \theta$	$\tan \theta = \sin \theta / \cos \theta$	$\theta \in [-\pi, \pi]$ (预处理)
	向量	$x_n = A_n \sqrt{x_0^2 + y_0^2}$ $y_n = 0$ $z_n = z_0 + \tan^{-1}(y_0/x_0)$	$x_0 = I$ $y_0 = Q$ $z_0 = 0$	$\sqrt{I^2 + Q^2}$ $\tan^{-1}(Q/I)$	-	-
双曲系统	旋转	$x_n = A_n(x_1 \cosh z_1 - y_1 \sinh z_1)$ $y_n = A_n(y_1 \cosh z_1 - x_1 \sinh z_1)$ $z_n = 0$	$x_1 = 1/A_n$ $y_1 = 0$ $z_1 = w$	$\cosh w$ $\sinh w$	$\tanh w = \sinh w / \cosh w$ $e^w = \sinh w + \cosh w$	$w \in [-1.1181, 1.1181]$
	向量	$x_n = A_n \sqrt{x_1^2 - y_1^2}$ $y_n = 0$ $z_n = z_1 + \tanh^{-1}(y_1/x_1)$	$w = y_1/x_1$ $z_1 = 0$	$\tanh^{-1} w$	$\sqrt{a}$ $\ln a$ $\begin{cases} x_1 = a + 1 \\ y_1 = a - 1 \end{cases}$	$a \in [0.1069, 9.3573]$ $w \in [-0.8069, 0.8069]$

在 CORDIC 模块中，表 25-1 中的初始化一列即对应寄存器 CORDIC_Xi/Yi/Zi，但寄存器只能配置为整数，因此需要在计算前对输入的小数向左移动适当的位数，将其变成整数配置到寄存器 CORDIC_Xi/Yi/Zi 中，在得到结果后再向右移位将其变为小数形式。值得注意的是，在 CORDIC 迭代计算的过程中会检查计算是否溢出，如果溢出则会得到错误的结果，因此对输入的小数向左移位时不能过度放大。

CORDIC 模块可以配置工作周期数 CORDIC_CFG.CYCLES，这里工作周期就代表着 CORDIC 算法的迭代次数，最多可以配置迭代 32 次。通过配置 CORDIC_CYCLES 寄存器可以实现精度和计算时间的权衡。对于大多数算法来说，CORDIC 模块可以达到小数点后 3 位以上的精度要求。

对于圆周系统和双曲系统，如果 z 输入时代表弧度，则需要先将 z 除以 π ；如果 z 输出时需要表示为弧度，则需要将 z 乘以 π 。具体操作请看以下示例。

示例 1：计算 $\sin 30^\circ$

- 配置旋转系统为圆周系统，CORDIC_CFG.FUNC=2'b01，配置模式为旋转模式，CORDIC_CFG.MODE=1'b0
- 配置 CORDIC 迭代次数 CORDIC_CFG.CYCLES=5'b11111
- 配置 CORDIC_Xi，把 $x_0 = 1/A_n$ 左移放大， $x_0 = 1/1.64676 \times 2^{31} = 1304065952$ ，配置 CORDIC_Yi 为 0
- 配置 CORDIC_Zi，此处要把 z 由弧度转换为数值并左移放大， $z_0 = 30^\circ/\pi \times 2^{31} = 357913941$
- 配置 CORDIC_CTRL.START=1'b1，启动 CORDIC 计算
- 可通过 CORDIC 中断或轮询 CORDIC_CTRL.DONE==1'b1 来判断运算是否完成；当运算完成时可读出 CORDIC_Yo 结果，为 1073742001
- 通过软件计算 $\sin 30^\circ = 1073742001/2^{31} = 0.5000000824...$

示例 2：计算 $\sqrt{3}$

- 配置旋转系统为双曲系统，CORDIC_CFG.FUNC=2'b11，配置模式为向量模式，CORDIC_CFG.MODE=1'b1
- 配置 CORDIC 迭代次数 CORDIC_CFG.CYCLES=5'b11111
- 配置 CORDIC_Xi，把 $x_1 = 3 + 1$ 左移放大， $x_1 = (3 + 1) \times 2^{23} = 33554432$
- 配置 CORDIC_Yi，把 $y_1 = 3 - 1$ 左移放大， $y_1 = (3 - 1) \times 2^{23} = 16777216$
- 配置 CORDIC_Zi 为 0
- 配置 CORDIC_CTRL.START=1'b1，启动 CORDIC 计算
- 可通过 CORDIC 中断或轮询 CORDIC_CTRL.DONE==1'b1 来判断运算是否完成；当运算完成时可读出 CORDIC_Xo 结果，为 24064723
- 此处还要除以 A_n ，通过软件计算 $\sqrt{3} = 24064723/2^{23}/0.82815936/2 = 1.731996...$

示例 3：计算 $e^{0.5}$

- 配置旋转系统为双曲系统，CORDIC_CFG.FUNC=2'b11，配置模式为旋转模式，CORDIC_CFG.MODE=1'b0
- 配置 CORDIC 迭代次数 CORDIC_CFG.CYCLES=5'b11111
- 配置 CORDIC_Xi，把 $x_1 = 1/A_n$ 左移放大， $x_1 = 1/0.82815936 \times 2^{23} = 10129223$ ，配置 CORDIC_Yi 为 0
- 配置 CORDIC_Zi，把 $z_1 = 0.5/\pi$ 左移放大， $z_1 = 0.5/\pi \times 2^{31} = 341782637$
- 配置 CORDIC_CTRL.START=1'b1，启动 CORDIC 计算
- 可通过 CORDIC 中断或轮询 CORDIC_CTRL.DONE==1'b1 来判断运算是否完成；当运算完成时可读出

- CORDIC_Xo 和 CORDIC_Yo 结果并相加，为 13830056
- 此处还要除以 2^{23} ，通过软件计算 $e^{0.5}=13830056/2^{23}=1.648671...$

25.3 CORDIC 中断

CORDIC 支持可屏蔽的溢出中断和运算结束中断，它们经过或逻辑产生 1 个中断，对应着系统中的 DIV/CORDIC 中断向量。

25.4 CORDIC 寄存器描述

CORDIC 模块归属于总线 AHB1 管理，此空间的基地址为 0x4801_0000。

表 25-2 CORDIC 相关寄存器表

名称	说明	读写权限	复位值	字节地址
CORDIC_CTRL	CORDIC 控制寄存器	R/W	0x0000_0000	0x4801_0000
CORDIC_CFG	CORDIC 配置寄存器	R/W	0x0000_0000	0x4801_0004
CORDIC_Xi	CORDIC X 输入寄存器	R/W	0x0000_0000	0x4801_000C
CORDIC_Yi	CORDIC Y 输入寄存器	R/W	0x0000_0000	0x4801_0010
CORDIC_Zi	CORDIC Z 输入寄存器	R/W	0x0000_0000	0x4801_0014
CORDIC_Xo	CORDIC X 输出寄存器	R	0x0000_0000	0x4801_0018
CORDIC_Yo	CORDIC Y 输出寄存器	R	0x0000_0000	0x4801_001C
CORDIC_Zo	CORDIC Z 输出寄存器	R	0x0000_0000	0x4801_0020

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）。

25.4.1 CORDIC 控制寄存器（CORDIC_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.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.							START	Res.		OVF_INTEN	DONE_INTEN	Res.		OVF	DONE
							rw			rw	rw			rw	rw

Bits	31:9	保留，必须保持复位值
Bit	8	START: CORDIC 启动配置，写 1 后 CORDIC 开始运算，运算结束后该位硬件自动清零
		0: CORDIC 未运行（默认）
		1: CORDIC 正在运行中
Bits	7:6	保留，必须保持复位值
Bit	5	OVF_INTEN: 溢出中断使能位
		0: 溢出中断禁止（默认）

	1: 溢出中断使能
Bit 4	DONE_INTEN: 运算结束中断使能位
	0: 运算结束中断禁止（默认）
	1: 运算结束中断使能
Bits 3:2	保留，必须保持复位值
Bit 1	OVF: 运算溢出标志位，写 1 清除，或在启动 CORDIC 时自动清除
	0: 运算过程中未发生溢出（默认）
	1: 运算过程中发生溢出
Bit 0	DONE: 运算结束标志位，写 1 清除，或在启动 CORDIC 时自动清除
	0: 当前无运算或运算未结束（默认）
	1: 运算已结束

25.4.2 CORDIC 配置寄存器（CORDIC_CFG）

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.											CYCLES[4:0]				
											rw				

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

Bits 31:21	保留，必须保持复位值
Bits 20:16	CYCLES[4:0]: CORDIC 运算的迭代次数，即计算消耗的周期数
	00000: 做 1 次迭代
	00001: 做 2 次迭代
	...
	11111: 做 32 次迭代
Bits 15:14	保留，必须保持复位值
Bits 13:12	FUNC[1:0]: 旋转系统选择
	00: 保留（默认）
	01: 圆周系统
	10: 保留
	11: 双曲系统
Bits 11:9	保留，必须保持复位值
Bit 8	MODE: CORDIC 模式选择
	0: 旋转模式（默认）
	1: 向量模式
Bits 7:6	保留，必须保持复位值
Bit 5	BYPASS_POST: 忽略 CORDIC 后处理，无特殊情况不要修改
	0: 进行后处理（默认）
	1: 忽略后处理

Bit 4	BYPASS_PRE: 忽略 CORDIC 预处理，无特殊情况不要修改
	0: 进行预处理（默认）
	1: 忽略预处理
Bits 3:0	保留，必须保持复位值

25.4.3 CORDIC X 输入寄存器（CORDIC_Xi）

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
CORDIC_Xi[31:0]															
rw															

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

Bits 31:0	CORDIC_Xi[31:0]: CORDIC X 输入（有符号数）
-----------	------------------------------------

25.4.4 CORDIC Y 输入寄存器（CORDIC_Yi）

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
CORDIC_Yi[31:0]															
rw															

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

Bits 31:0	CORDIC_Yi[31:0]: CORDIC Y 输入（有符号数）
-----------	------------------------------------

25.4.5 CORDIC Z 输入寄存器（CORDIC_Zi）

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
CORDIC_Zi[31:0]															
rw															

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

Bits 31:0	CORDIC_Zi[31:0]: CORDIC Z 输入（有符号数）
-----------	------------------------------------

25.4.6 CORDIC X 输出寄存器 (CORDIC_Xo)

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
CORDIC_Xo[31:0]															
r															

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

Bits	31:0	CORDIC_Xo[31:0]: CORDIC X 输出
------	------	------------------------------

25.4.7 CORDIC Y 输出寄存器 (CORDIC_Yo)

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
CORDIC_Yo[31:0]															
r															

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

Bits	31:0	CORDIC_Yo[31:0]: CORDIC Y 输出
------	------	------------------------------

25.4.8 CORDIC Z 输出寄存器 (CORDIC_Zo)

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
CORDIC_Zo[31:0]															
r															

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

Bits	31:0	CORDIC_Zo[31:0]: CORDIC Z 输出
------	------	------------------------------

26. 调试模块

26.1 主要特性

LCM32F06X 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。LCM32F06X 使用 SW-DP 调试口，仅使用 2 个 I/O（SWDIO 和 SWCLK）执行调试，其中 SWCLK 有 2 组 I/O（PA14 或 PB2）功能复用，SWDIO 有一组 I/O（PA13）功能复用。

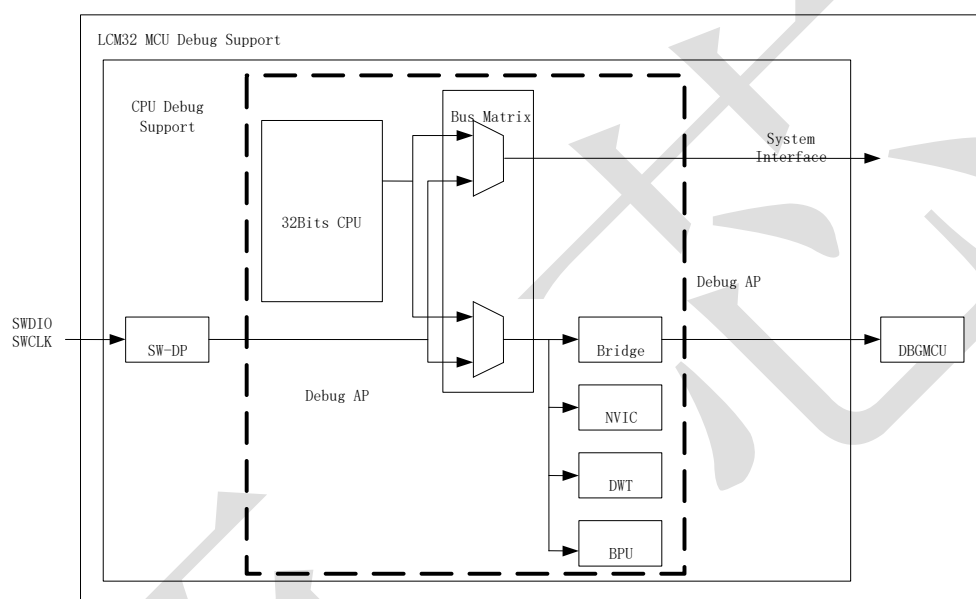


图 26-1 调试模块结构图

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，LCM32F06X 还包括了一些特性：

- 可以调整的调试复用接口
- 可以在外部复位条件下在 I/O 上打入特定序列改变 SWD 复用的 I/O
- MCU 调试模块（支持低功耗模式，外设时钟等）

26.2 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。

26.2.1 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，LCM32F06X 有 2 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 26-1 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/PB2

芯片正常上电时，PB2 引脚初始为外部复位功能，如果校准字节 13 中配置 PB2 作为 SWCLK，则 PB2 会在芯片启动完成后自动关闭复位功能，之后可通过软件配置 ChipCtrl_BDCR.RST_CFG 寄存器启动 PB2 复位功能。

26.2.2 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。

LCM32F06X 支持在外部复位情况下通过 PB2 和 PA13 打入一串特定的序列，可以从外部直接调整 SWD 接口的复用，防止芯片因软件把相应配置改掉后芯片无法进入调试状态。具体操作为上电时按住外部复位一段时间，随后向 PB2 输入 SWCLK，向 PA13 打入序列 0101101110+ DebugMode (5 bits) + 结束位 (1)，如果配置 PA14 为 SWCLK 则 DebugMode=5'b00001，如果配置 PB2 为 SWCLK 则 DebugMode=5'b00010。配置完成后可退出打入序列过程，并放开外部复位。再次发生外部复位时该功能会被复位。

26.2.3 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。PB2在作为外部复位时会带有内部输入上拉，在作为 SWCLK时会自动调整为内部输入下拉。

26.3 SWD 通讯

26.3.1 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 I/O 口外部必须上拉（推荐 100KΩ，LCM32F06X 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。

26.3.2 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 26-2 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 26-3 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 26-4 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。

26.3.3 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）。

26.3.4 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。

26.3.5 SW-DP 寄存器

当 APnDP = 0 时，访问如下的寄存器。

表 26-5 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： ● 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

26.3.6 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 26-6 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：

		● 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 ● 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）

26.4 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 26-7 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）

26.5 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。

26.6 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。

26.6.1 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。

26.6.2 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。

26.7 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

26.7.1 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，系统时钟必须保持开启的状态，不同低功耗模式下：

- 睡眠模式下，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。

26.7.2 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

26.7.3 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_ TIM2 _STOP	DBG_ IWDG_ STOP	Res.	DBG_ WT_ STOP	DBG_ TIM1 7 _STOP	DBG_ TIM16 _STOP	DBG_ TIM15 _STOP	Res.			DBG_ TIM1 _STOP
					rw	rw		rw	rw	rw	rw				rw

Bits	31:10	保留，必须保持复位值
------	-------	------------



Bit 10	DBG_TIM2_STOP: 调试停止状态下 TIM2 的控制
	0: 调试状态下 TIM2 继续保持计数 (默认)
	1: 调试状态下 TIM2 停止计数
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	保留, 必须保持复位值
Bit 7	DBG_WT_STOP: 调试停止状态下 WT 的控制
	0: 调试状态下 WT 继续保持计数 (默认)
	1: 调试状态下 WT 停止计数
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
Bits 3:1	保留, 必须保持复位值
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

27. 预驱概述

27.1 预驱 LND31A01

LND31A01 是一款高性价比三相 PMOS、NMOS 管栅极驱动专用模块，内部集成了 LDO、死区控制电路、欠压关断电路、闭锁电路、输出驱动电路，用于电机控制器、电源的驱动电路。电源电压范围 6V~36V，静态功耗小于 1mA。内部集成 5V 输出 LDO，可为外部 MCU 等器件供电。当输入电压超过 12V 时，为了更好的匹配 P/N MOS 管，LO 输出最高电压为 11V，HO 输出最低电压为 VCC 减去 11V。LO 输出电流能力 IO+/- 0.045/0.28A，HO 输出电流能力 IO+/- 0.26/0.04A。

特性：

- 三相 P/N MOS 半桥式三相输出
- 电源电压输入范围：6V-36V，最高电压可达 40V
- 适应 3V-30V 输入电压
- 内置 5V/50mA 输出 LDO
- 驱动电流能力：LO +/- 0.045/0.28A，HO +/- 0.26/0.04A
- 内置 VCC 欠压保护
- 内置 60ns 死区时间
- 自带闭锁功能
- 温度范围-45~125℃
- LIN1/2/3 输入通道高电平有效，控制 LO 输出
- HIN1/2/3 输入通道高电平有效，控制 HO 输出

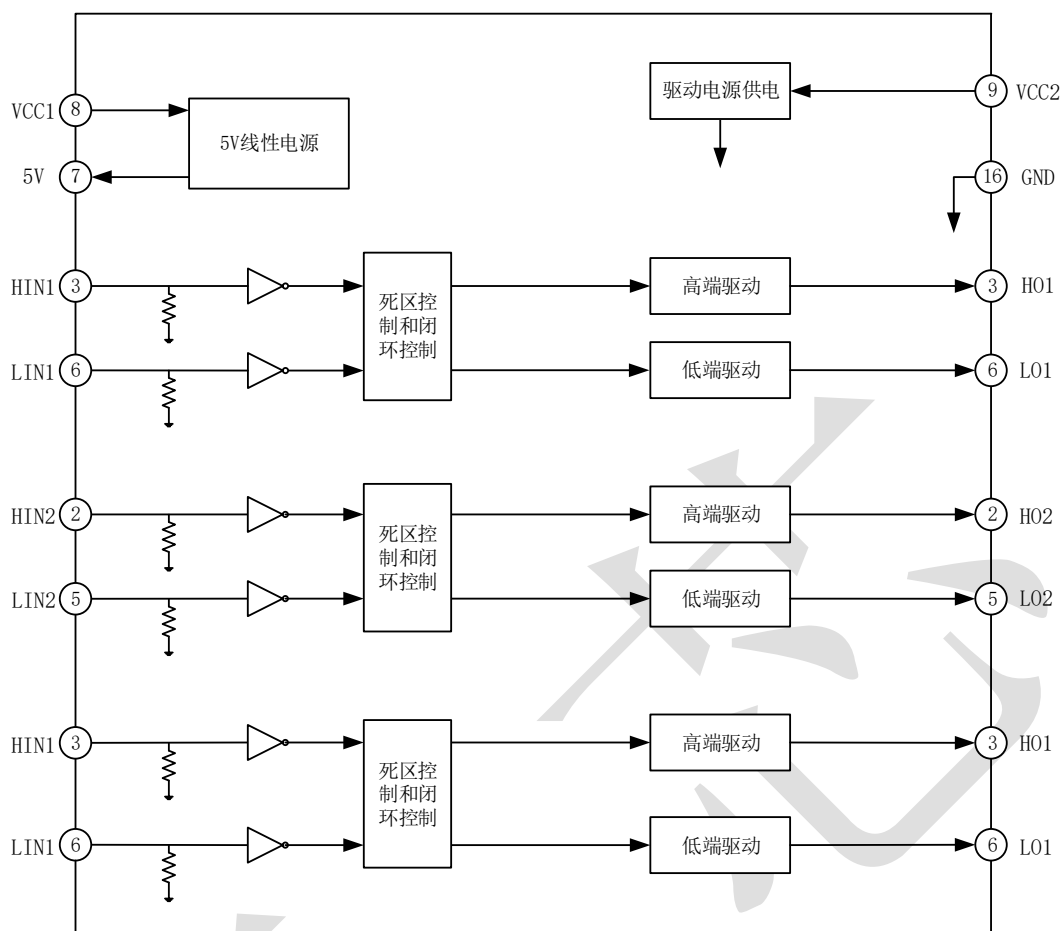


图 27-1 预驱 LND31A01 结构框图

27.1.1 LND31A01 开关特性和死区时间

如下图所示， T_{on} 为开关开延时， T_{off} 为开关关延时， T_r 为上升时间， T_f 为下降时间， DT 为死区时间，具体数值请参考说明书中的相应电气特性。

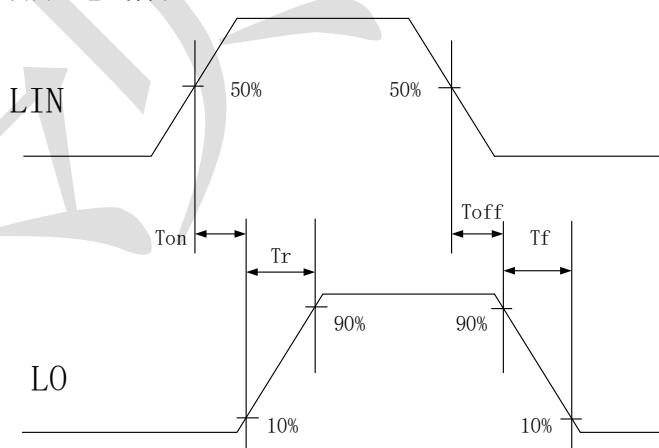


图 27-2 LND31A01 输出 LO 开关时间

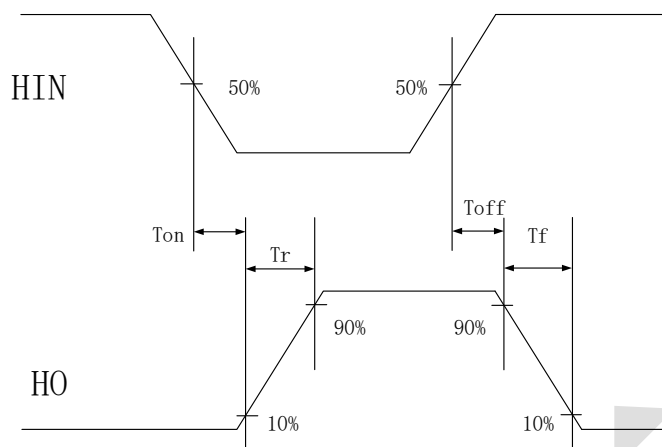


图 27-3 LND31A01 输出 HO 开关时间

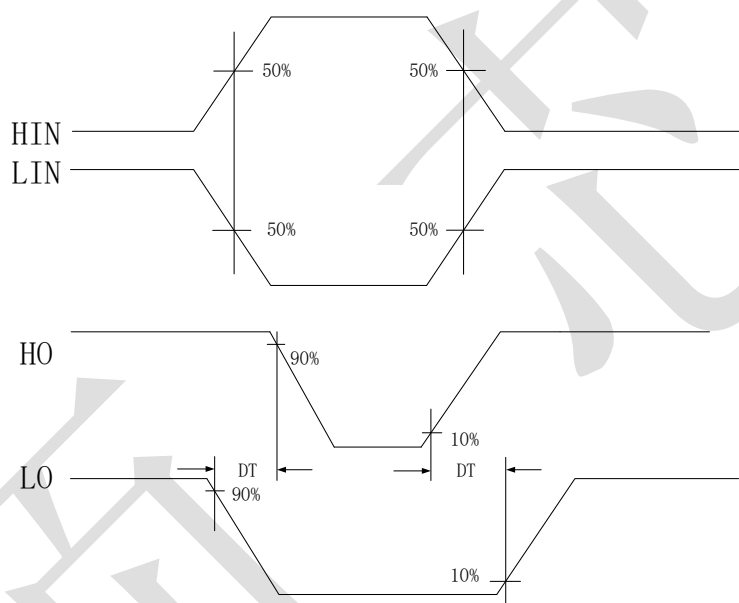


图 27-4 LND31A01 死区时间

27.1.2 LND31A01 VCC 端电源电压

VCC1 给 5V 线性电源供电，输入高压场合，外部输入电源可以串联一个功率电阻到 VCC1，降低芯片功耗。5V 输出根据外部供电要求，外接 1 μ F-10 μ F 贴片瓷片电容。VCC2 给驱动部分供电，外部电容尽量靠近芯片端口。

27.1.3 LND31A01 逻辑信号要求和特性

LND31A01 主要功能有逻辑信号输入处理、死区时间控制和图腾柱式输出。逻辑信号输入端高电平阈值为 2V 以上，低电平阈值为 0.8V 以下，要求逻辑信号的输出电流小，可以使 MCU 输出逻辑信号直接连接到预驱的输入通道上。LO 输出驱动器的最大灌入为 0.28A，最大输出电流为 45mA；HO 输出驱动器的最大灌入为 40mA，最大输出电流为 0.26A。输入逻辑信号与输出控制信号之间的传导延时小，LO 输出开通传导延时为 90ns，关断传导延时为 30ns；HO 输出开通传导延时为 90ns，关断传导延时为 30ns。LO 输出开通的上升时间为 280ns，关断的下降时间为 60ns；HO 输出开通的下降时间为 300ns，关断的上升时间为 80ns。输入信号和输出信号逻辑功能图如下：

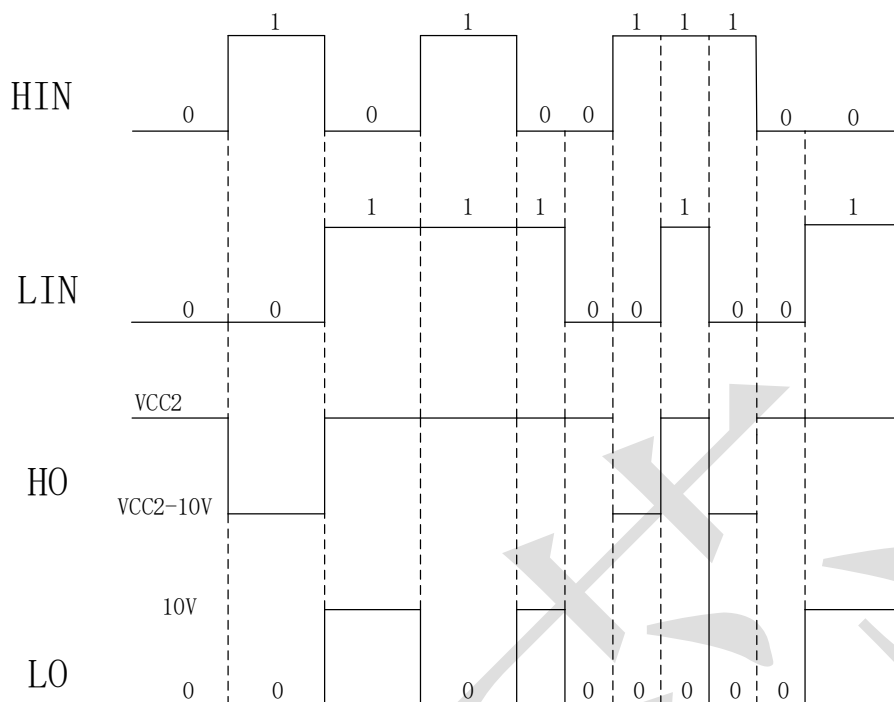


图 27-5 LND31A01 信号的逻辑功能

表 27-1 LND31A01 信号逻辑真值表

输入		输出	
输入、输出逻辑			
LIN	HIN	LO	HO
0	0	OFF	OFF
0	1	OFF	ON
1	0	ON	OFF
1	1	OFF	OFF

注：OFF：LO=0V，HO=VCC2；ON：LO=10V，HO=VCC2-10V

从真值表可知，当输入逻辑信号 LIN 为“0”和 HIN 为“1”时，驱动器控制输出 LO 下 NMOS 管关闭，HO 上 PMOS 管开启；当输入逻辑信号 LIN 为“1”和 HIN 为“0”时，驱动器控制输出 LO 下 NMOS 管开启，HO 上 PMOS 管关闭；在输入逻辑信号 LIN 和 HIN 同时为“0”和同时为“1”情况下，驱动器控制输出 LO、HO 同时关闭；内部逻辑处理器杜绝 LO、HO 同时开启，具有相互闭锁功能。

27.2 预驱 LND31A02

LND31A02 是三相高速功率 MOSFET 驱动器，具有 6 个通道，最高工作电压可达 40V。采用高低压兼容工艺使得高、低侧栅驱动电路可以单芯片集成。逻辑兼容 CMOS 或 LSTTL 输出，低至 3.3V 逻辑输入。

此外，采用内置死区功能来避免高压侧交叉导通。预驱模块为 PMOS 和 NMOS 输出 10V 门电源电压，为了简化 PCB 设计，预驱模块中集成了一个 5V/40mA 的 LDO，用于为 MCU 供电。此外出于安全考虑，预驱模块还集成了热关闭保护。

特性：

- P/N MOS 半桥式三相输出
- 电源电压输入范围：5V-28V，最高电压可达 40V

- 适应 3.3V-20V 输入电压
- 内置 5V/40mA LDO
- 驱动电流能力: IO +/- 0.05/0.3A
- 内置 VCC 欠压保护
- 内置 130ns 死区时间
- 自带闭锁功能
- 温度范围-40~125℃
- 内置过温保护
- LIN1/2/3 输入通道高电平有效, 控制 LO 输出
- HIN1/2/3 输入通道高电平有效, 控制 HO 输出

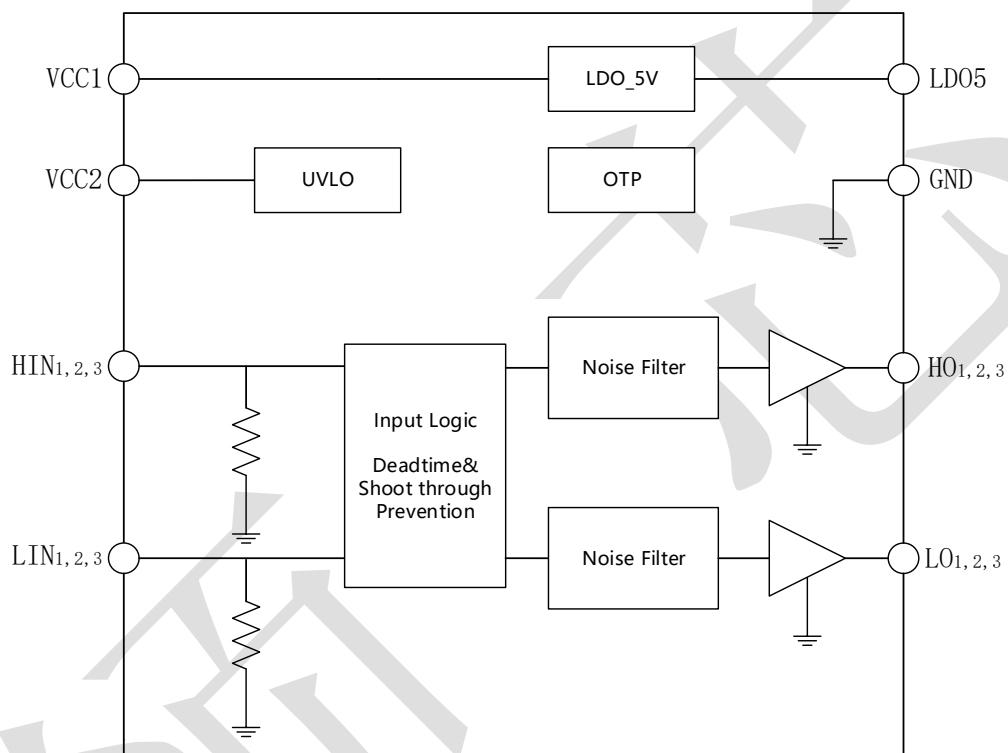


图 27-6 预驱 LND31A02 结构框图

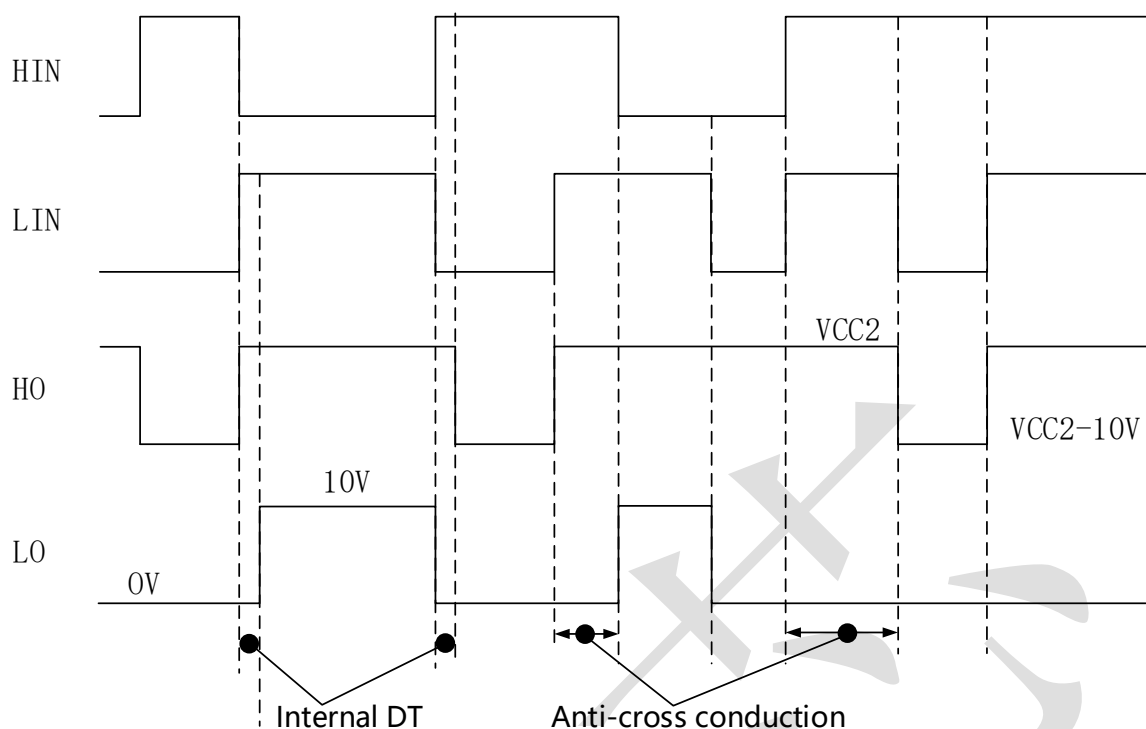


图 27-7 LND31A02 输入输出时序

27.3 预驱 LND32B01

LND32B01 是一款高性价比的大功率 MOS 管、IGBT 管栅极驱动专用模块，内部集成了逻辑信号输入处理电路、死区时控制电路、欠压保护电路、闭锁电路、电平位移电路、脉冲滤波电路及输出驱动电路。模块高端的工作电压可达 260V，低端 VCC 的电源电压范围宽 4.5V~20V。该模块具有闭锁功能防止输出功率管同时导通，输入通道 HIN 和 LIN 内建了下拉电阻，在输入悬空时使上、下功率 MOS 管处于关闭状态，输出电流能力 IO +0.8A/-1.2A。

特性：

- 高端悬浮自举电源设计，耐压高达 260V
- 集成自举二极管
- 电源电压输入范围：5V-20V，最高电压可达 25V
- 适应 3.3V-25V 输入电压
- 驱动电流能力：IO +/- 0.8/1.2A
- 内置 VCC 和 VB 欠压保护
- 内置 200ns 死区时间
- 自带闭锁功能
- 温度范围-40~125℃
- HIN 输入通道高电平有效，控制高端 HO 输出
- LIN 输入通道高电平有效，控制低端 LO 输出

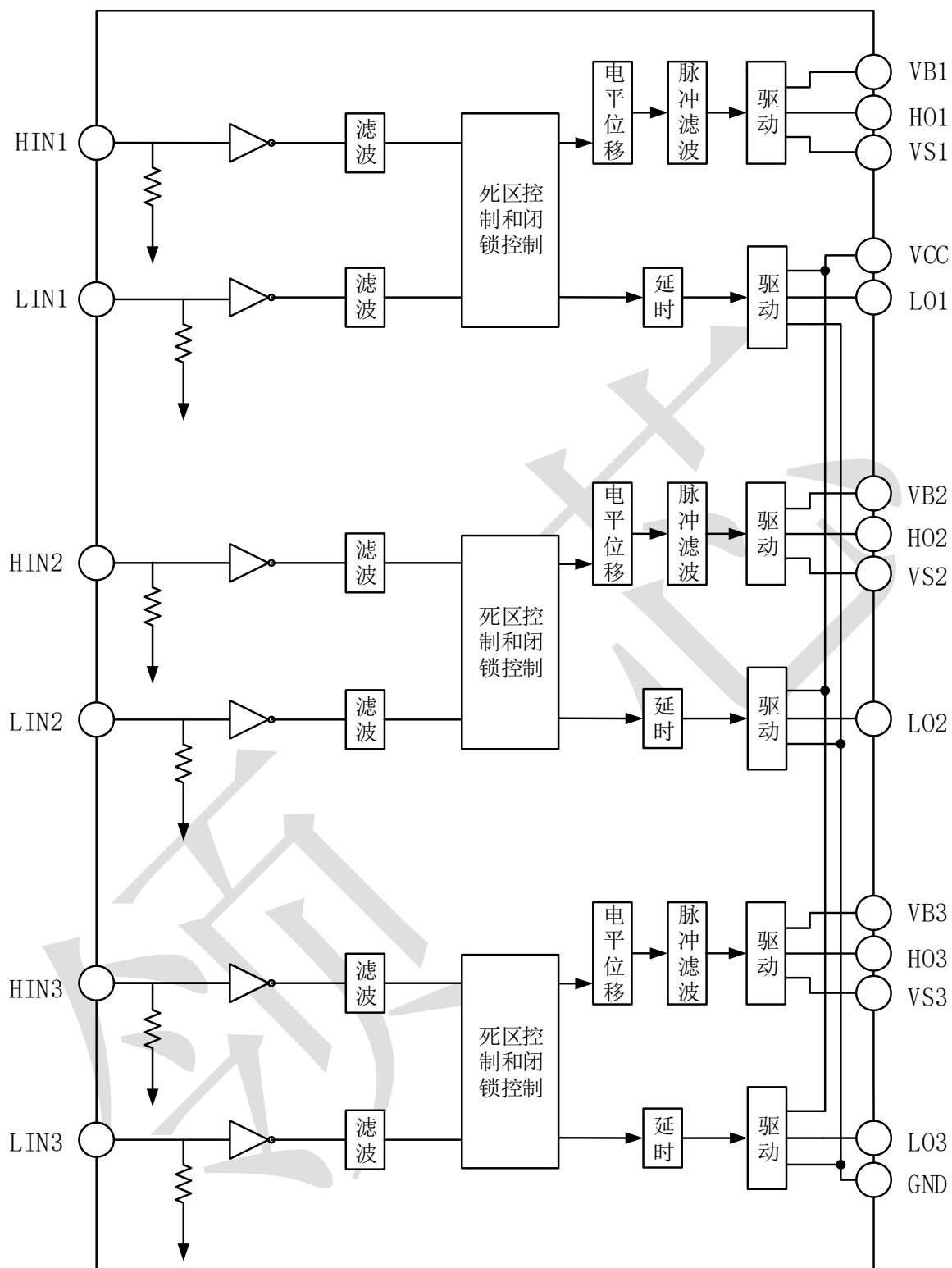


图 27-8 预驱 LND32B01 结构框图

27.3.1 LND32B01 开关特性和死区时间

如下图所示， T_{on} 为开关开延时， T_{off} 为开关关延时， T_r 为上升时间， T_f 为下降时间， DT 为死区时间，具体数值请参考说明书中的相应电气特性。

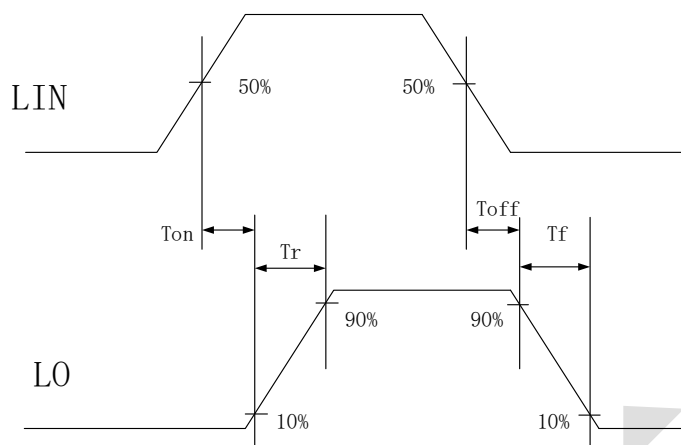


图 27-9 LND32B01 输出 LO 开关时间

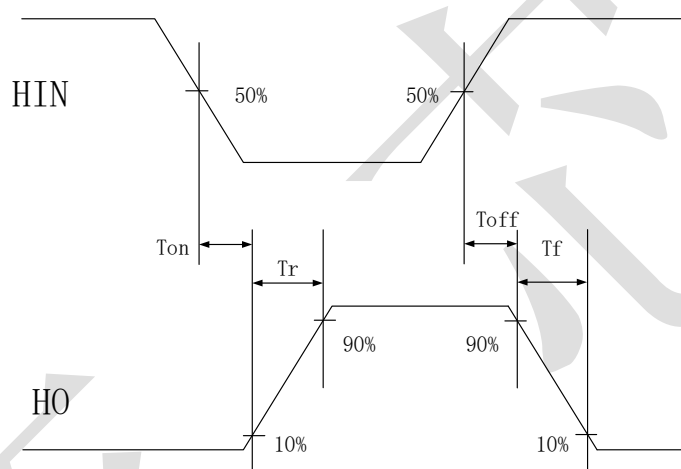


图 27-10 LND32B01 输出 HO 开关时间

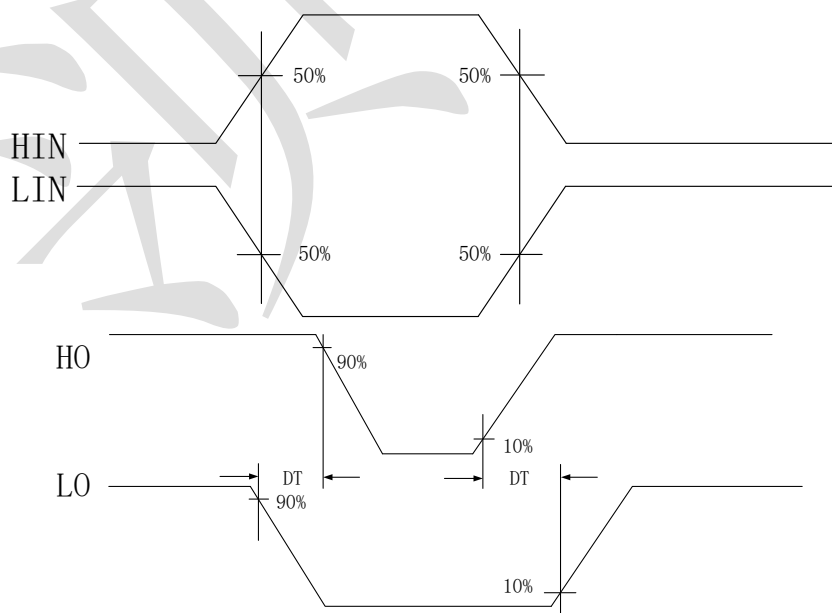


图 27-11 LND32B01 死区时间

27.3.2 LND32B01 VCC 端电源电压

针对不同的 MOS 管，选择不同的驱动电压，开启 MOS 管推荐电源 VCC 工作电压典型值为 5V-15V。

LND32B01 采用自举悬浮驱动电源结构，只用一路电源电压 VCC 即可完成高端 N 沟道 MOS 管和低端 N 沟道 MOS 管两个功率开关器件的驱动，给实际应用带来极大的方便。LND32B01 可以使用外接一个自举二极管和一个自举电容自动完成自举升压功能，如图 27-13 所示。假定在下管开通、上管关断期间 VC 自举电容已充到足够的电压（VC=VCC），当 HO 输出高电平时上管开通、下管关断时，VC 自举电容上的电压将等效一个电压源作为内部驱动器 VB 和 VS 的电源，完成高端 N 沟道 MOS 管的驱动。自举电容 VC 的 PCB 布局尽量靠近芯片的 VB 和 VS，自举电容可以选择瓷片电容或者电解电容，最小容值可按以下公式计算：

$$VC \geq 15 * \frac{2 * [2 * Qg + Q_{period} + \frac{I_{bso}}{F} + \frac{I_{bsc}}{F}]}{VCC - VF - V_{ds}}$$

其中：Qg 为高压侧 MOS 管的栅极电荷；

Qperiod 为每个周期中电平转换电路的电荷要求，约为 5nC；

Ibso 为高压侧驱动电路打开时的静态电流；

Ibsc 为自举电容的漏电流；

F 为电路工作频率；

VCC 为低端电源电压；

VF 为自举二极管的正向压降；

Vds 为 MOS 管需要彻底打开的 GS 电压。

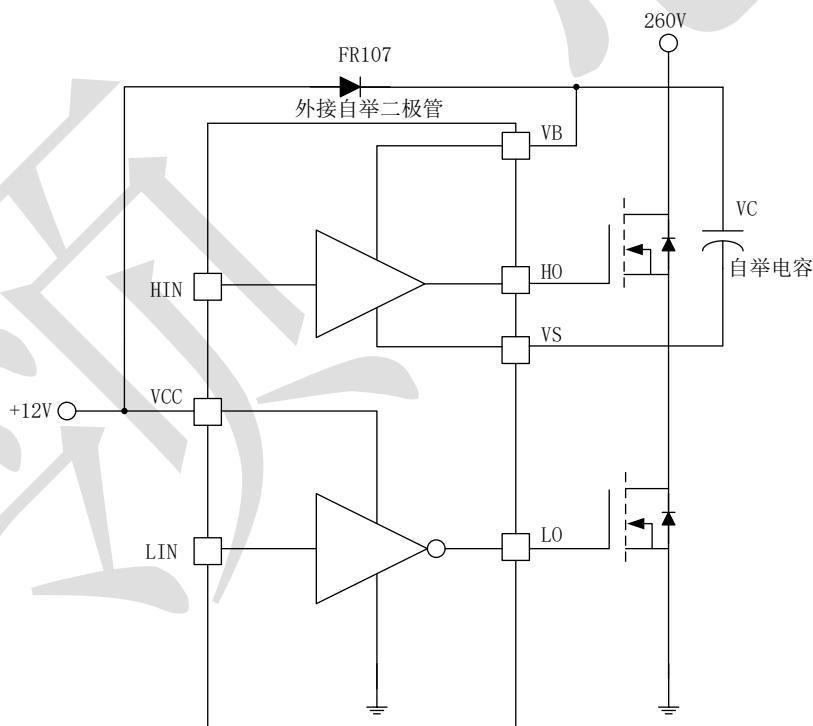


图 27-12 LND32B01 自举电路结构

27.3.3 LND32B01 逻辑信号要求和特性

LND32B01 主要功能有逻辑信号输入处理、死区时间控制、电平转换功能、悬浮自举电源结构和上下桥图腾柱式输出。逻辑信号输入端高电平阈值为 2.5V 以上，低电平阈值为 1.0V 以下，要求逻辑信号的输出电流小，可以使 MCU 输出逻辑信号直接连接到预驱的输入通道上。高端上桥臂和低端下桥臂输出驱动

器的最大灌入可达 0.8A，最大输出电流可达 1.2A，高端上桥臂通道可以承受 260V 的电压。输入逻辑信号与输出控制信号之间的传导延时小：低端输出开通传导延时为 320ns，关断传导延时为 120ns；高端输出开通传导延时为 320ns，关断传导延时为 120ns。低端输出开通的上升时间为 35ns，关断的下降时间为 25ns；高端输出开通的上升时间为 35ns，关断的下降时间为 25ns。

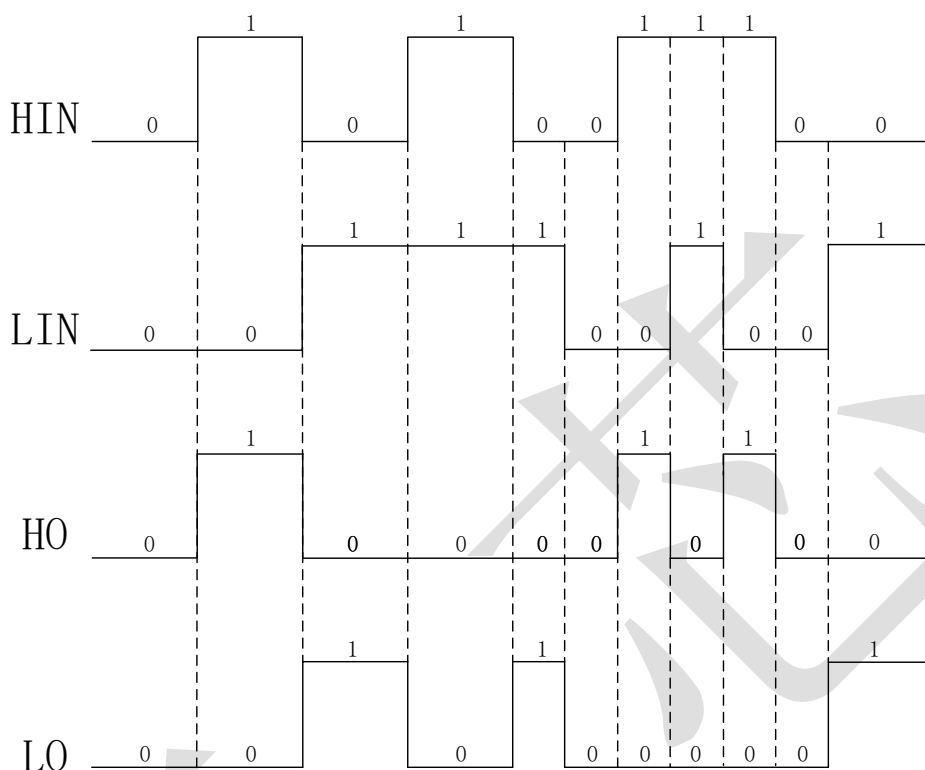


图 27-13 LND32B01 信号的逻辑功能

表 27-2 LND32B01 信号逻辑真值表

输入		输出	
输入、输出逻辑			
LIN	HIN	LO	HO
0	0	0	0
0	1	0	1
1	0	1	0
1	1	0	0

从真值表可知，当输入逻辑信号 HIN 为“1”和 LIN 为“0”时，驱动器控制输出 HO 为“1”上管打开，LO 为“0”下管关断；当输入逻辑信号 HIN 为“0”和 LIN 为“1”时，驱动器控制输出 HO 为“0”上管关断，LO 为“1”下管打开；在输入逻辑信号 HIN 为“1”和 LIN 为“1”或者 HIN 为“0”和 LIN 为“0”时，驱动器控制输出 HO、LO 为“0”将上、下功率管同时关断，内部逻辑处理器杜绝控制器输出上、下功率管同时导通，具有相互闭锁功能。

27.4 预驱 LND33A04

LND33A04 是一款集成了 70V 耐压的三个独立半桥栅极驱动、自举二极管、5V LDO 和 12V LDO 控制电路的控制芯片，适合于 12V、24V 和多节锂电池供电应用中三相电机应用中高速功率 MOSFET 和 IGBT 的栅极驱动。内置低侧电源和高侧电源欠压 (UVLO) 保护功能，防止功率管在过低的电压下工作，提高效率。

输入脚兼容 3.3/5.0V 输入逻辑，集成防穿通死区时间为 250ns，驱动能力为+1.5A/-1.8A。集成共模噪声消除技术使得高边驱动器在高 dv/dt 噪声环境能稳定工作，并且具有宽泛的负瞬态电压忍受能力。

特性：

- 高端悬浮自举电源设计，耐压高达 70V
- 集成自举二极管
- 电源电压 VM 输入范围：5.5V-55V
- 适应 3.3V-6.5V 输入电压
- 内置 5V/50mA LDO
- 驱动电流能力：IO +/- 1.5/1.8A
- 内置 VCC 和 VBS 欠压保护
- 内置 250ns 死区时间
- 自带闭锁功能
- 温度范围-40~125℃
- HIN 输入通道高电平有效，控制高端 HO 输出
- LIN 输入通道高电平有效，控制低端 LO 输出
- 集成 12V LDO 控制电路
- 集成 5V LDO 输出短路保护

27.4.1 LND33A04 低边电源 VCC 和欠压锁定

VCC 为低边电路电源供应端，能为输入逻辑电路和低边输出功率级工作提供所需的驱动能量。内置的欠压锁定电路能保证芯片工作在足够高的电源电压范围，进而防止由于低驱动电压所产生的热耗散对 MOSFET/IGBT 造成损害。如图 27-14 所示，当 VCC 上升并超过阈值电压 $V_{CC_R} = 3.6V$ 后，低边控制电路锁开始工作，LO 开始输出；反之，VCC 下降并低于阈值电压 $V_{CC_F} = 3.3V$ 后，低边电路锁定，芯片停止工作，LO 停止输出。VCC 工作电压范围建议为 4.5V-15.0V。

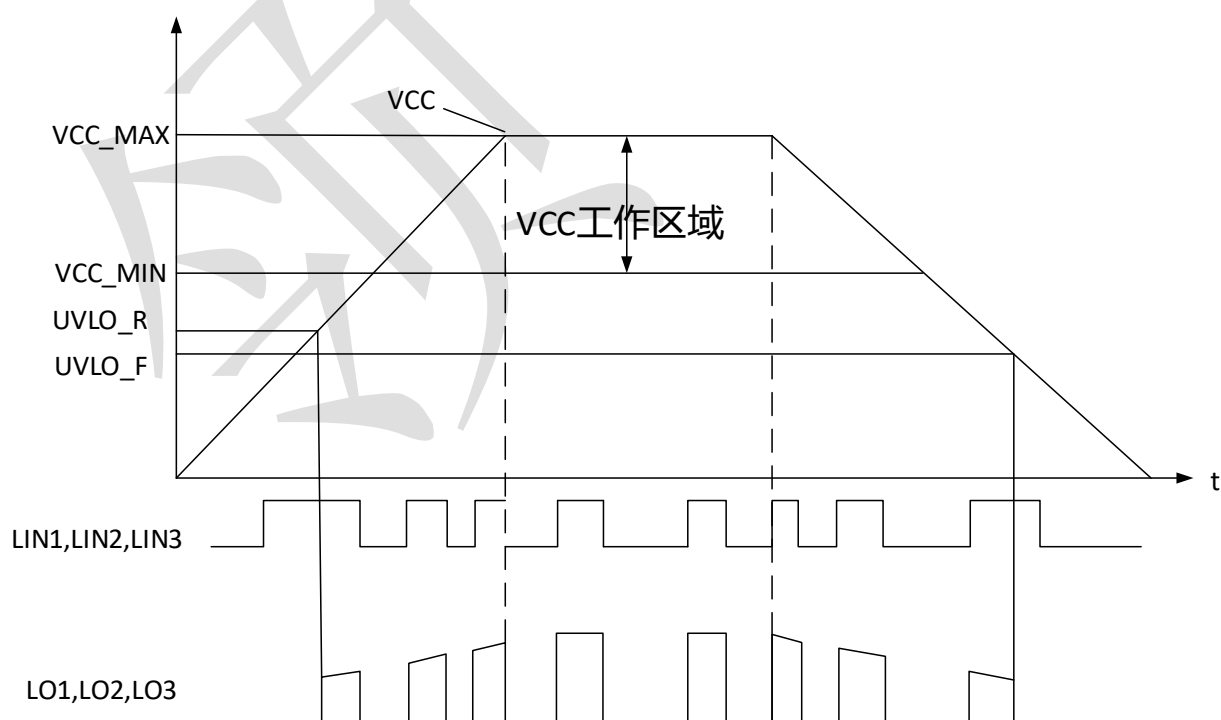


图 27-14 LND33A04 VCC 电源 UVLO 工作区域

27.4.2 LND33A04 高边电源 VBS 和欠压锁定

VBS 电源为高边电路供电电源，其中 VBS1 (VB1-VS1)、VBS2 (VB2-VS2) 和 VBS3 (VB3-VS3) 分别对应相 1、相 2 和相 3 高边驱动电源。由浮动电源 VBS 供电的整体高边电路以地 GND 为参考点，并跟随外部功率管 MOSFET/IGBT 的源/发射极电压，在地线和母线电压之间摆动。由于高边电路具有低静态电流消耗，因此整个高边电路可以由与 VCC 连接的自举电路供电，并且只需一个较小的电容就能维持驱动功率管所需电压。如图 27-15 所示，高边电源 VBS 的欠压锁定类似于低边 VCC 电源，VBS 工作电压范围建议在 4.5V-15.0V。

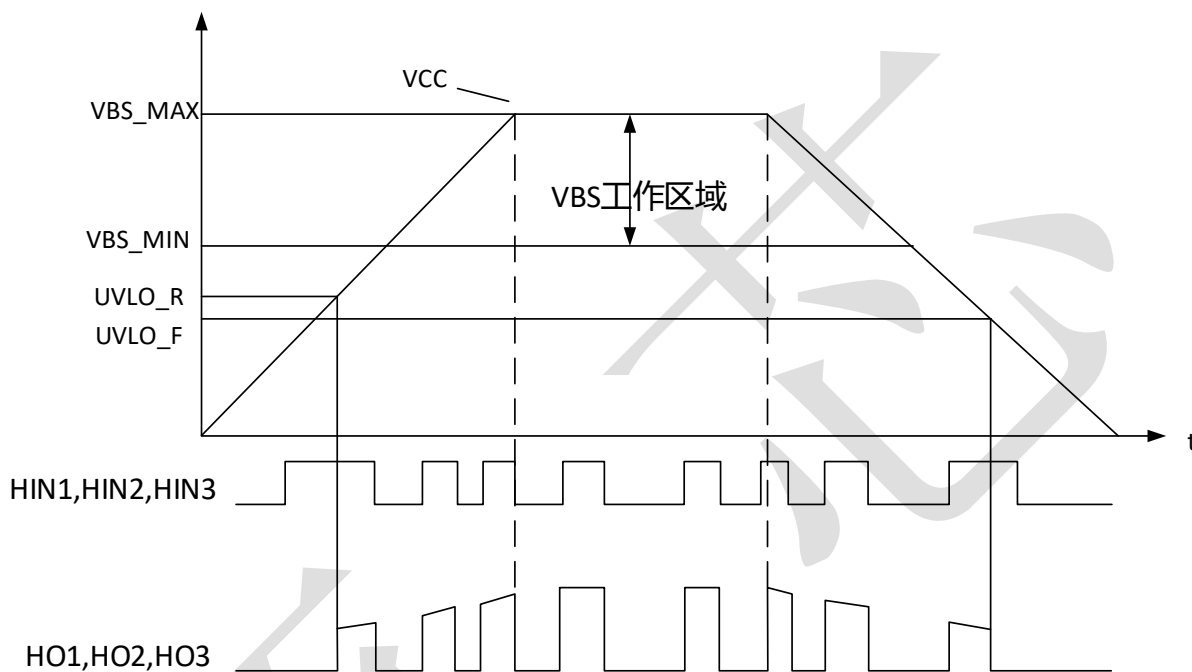


图 27-15 LND33A04 VBS 电源 UVLO 工作区域

27.4.3 LND33A04 低边和高边逻辑输入控制: HIN 和 LIN

为了更好地兼容各类控制器，特别将 6 个输入施密特反相器的阈值调整到最低可以兼容 3.3V 的 LSTTL 和 CMOS 逻辑电平。内置施密特反相器和先进脉冲滤波器更加有效地屏蔽非正常的输入短脉冲信号，大幅提升系统的对于干扰免疫力和可靠性。每个逻辑输入端在芯片内部都预置 140kΩ 的下拉电阻，保证在焊接（虚焊）和输入非有效连接等异常情况下能提供关断功率管控制信号。输入脉冲宽度尽量不低于 300ns，以保证正确的输入和输出关系。

27.4.4 LND33A04 功率管直通保护

LND33A04 内部配备了专门用于防止功率管直通的保护电路，能有效地防止高边和低边输入信号受到共模干扰时造成的功率管损害。图 27-16 展示了直通保护电路保护功率管的过程。功率管直通意味着同一个半桥中的高边栅极驱动器输出 HO 和低边栅极驱动器输出 LO 同时为“高”，这时会有非常大的有害电流同时流过上下边功率管，并伴有较大的功率损耗产生，严重时甚至会直接损坏功率管。如图 27-16 所示，当同一相的低边输入 LIN 和高边输入 HIN 同时为“高”时，内部保护电路将驱动器输出 HO 和 LO 拉至“低”，有效关断功率管。当其中一个输入信号变为“低”时，驱动器输出需要经过一个死区时间的延时才能输出“高”。该措施避免了有害短输入脉冲造成的功率管开关过度状态，有效地减小损耗，降低功率管损坏的风险。

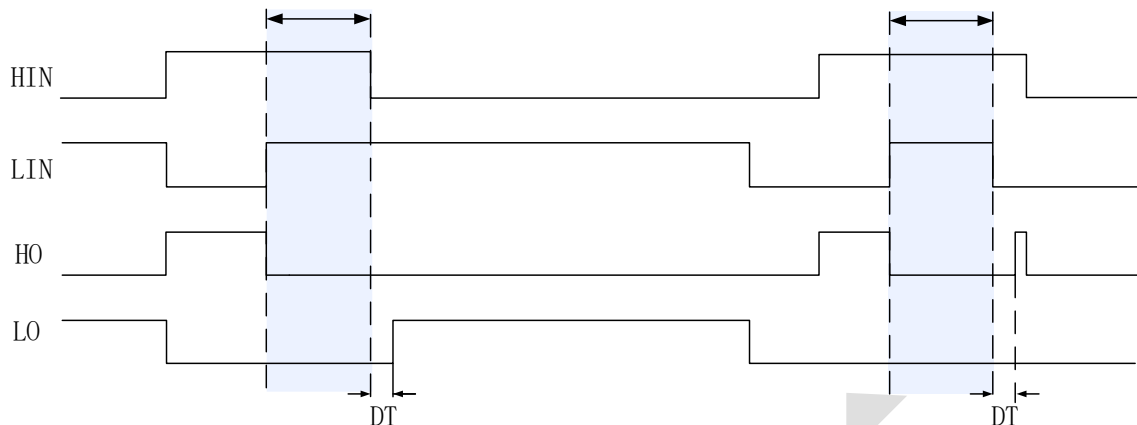


图 27-16 LND33A04 功率管直通保护

27.4.5 LND33A04 死区时间

LND33A04 内部设置了固定的死区时间保护电路。在死区时间内，高边和低边驱动器输出均被设定为“低”。所设定的死区时间必须在确保一个功率管有效关断之后，再开启另外一个功率管，这样防止产生上下管直通现象。如果由逻辑输入设定的外部死区时间小于内部最小死区时间，则驱动器输出的死区时间为芯片内部设定的死区时间；一旦由逻辑输入设定的外部死区时间大于芯片内部设定死区时间，则以逻辑输入设定的外部死区时间为驱动器输出死区时间。图 27-17 描述了死区时间、输入信号和驱动器输出信号的时序关系。

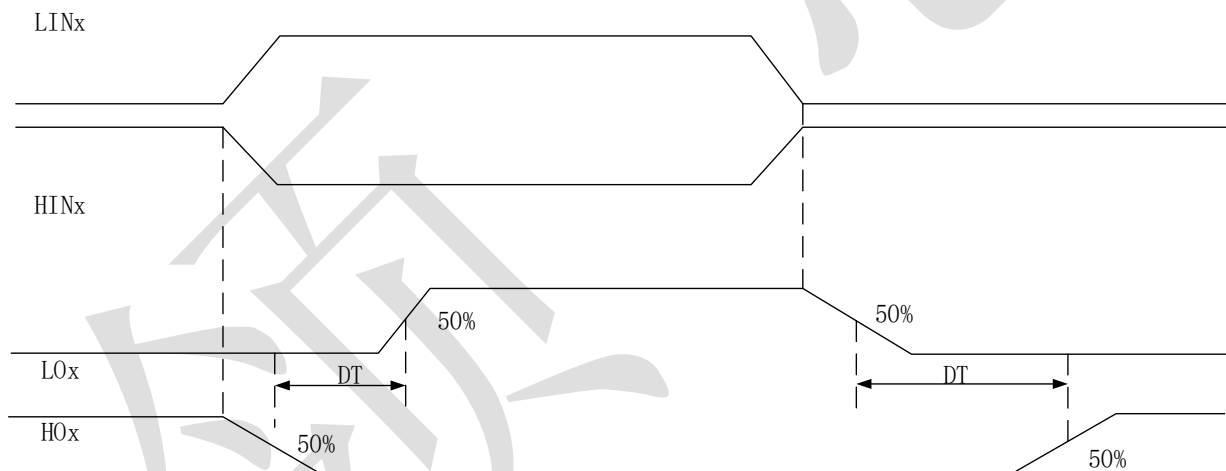


图 27-17 LND33A04 死区保护

27.5 预驱 LND36C01

LND36C01 是一款高性价比的大功率 MOS 管、IGBT 管栅极驱动专用预驱模块，内部集成了 LDO、逻辑信号输入处理电路、死区时控制电路、欠压保护电路、闭锁电路、电平位移电路、脉冲滤波电路及输出驱动电路。LND36C01 高端的工作电压可达 600V，低端 VCC 的电源电压范围宽 8V~20V。该模块具有闭锁功能防止输出功率管同时导通，输入通道 HIN 和 LIN 内建了下拉电阻，在输入悬空时使上、下功率 MOS 管处于关闭状态，输出电流能力 IO +0.6A/-1.2A。

特性：

- 高端悬浮自举电源设计，耐压高达 600V
- 集成自举二极管

- 电源电压输入范围：8V-20V，最高电压可达 25V
- 适应 3.3V-25V 输入电压
- 内置 5V/50mA LDO
- 驱动电流能力：IO +/- 0.6/1.2A
- 内置 VCC 和 VBS 欠压保护
- 内置 175ns 死区时间
- 自带闭锁功能
- 温度范围-40~125°C
- HIN 输入通道高电平有效，控制高端 HO 输出
- LIN 输入通道高电平有效，控制低端 LO 输出

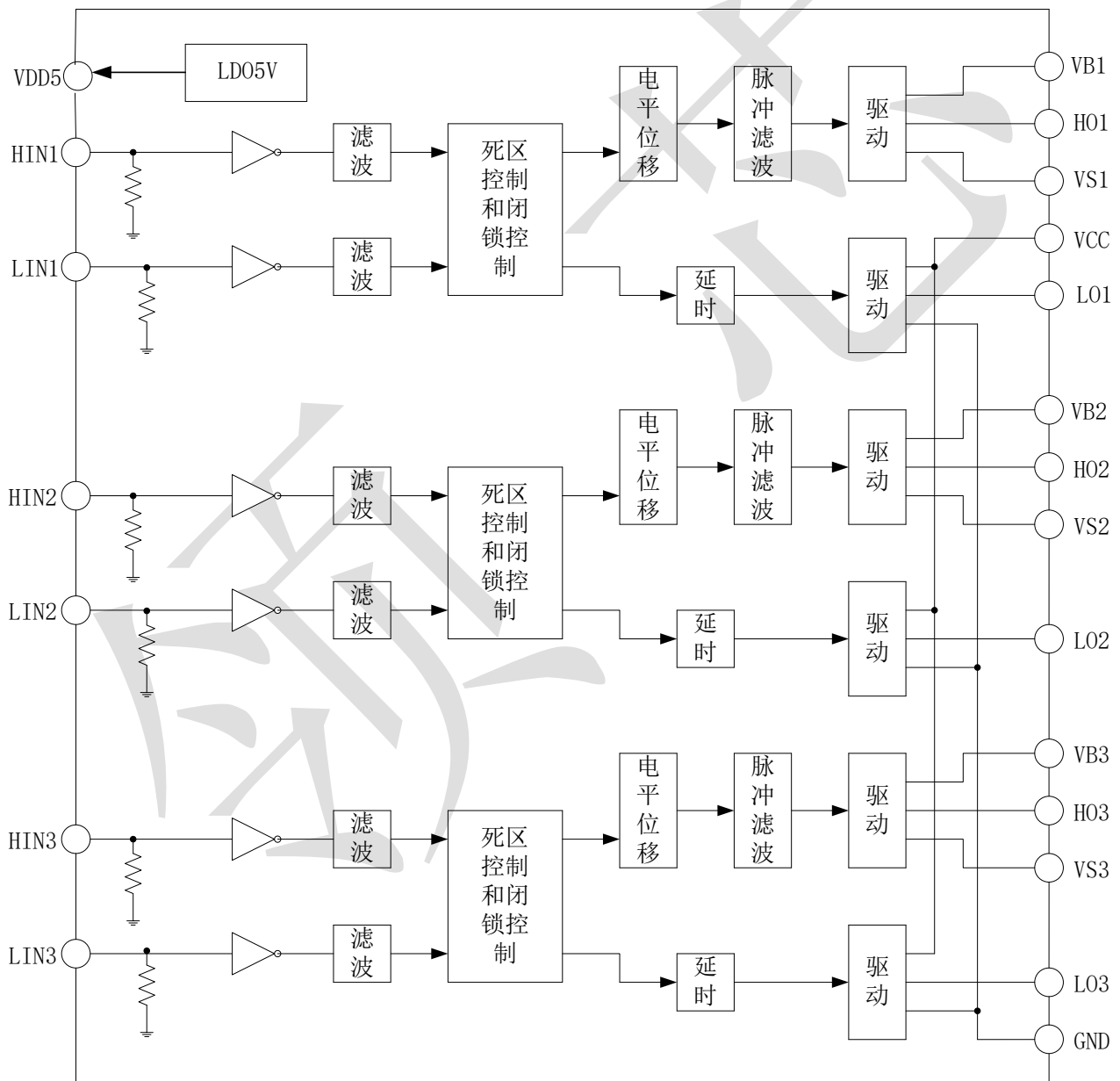


图 27-18 预驱 LND36C01 结构框图

27.5.1 LND36C01 开关特性和死区时间

如下图所示， T_{on} 为开关开延时， T_{off} 为开关关延时， T_r 为上升时间， T_f 为下降时间，具体数值请参考说明书中的相应电气特性。

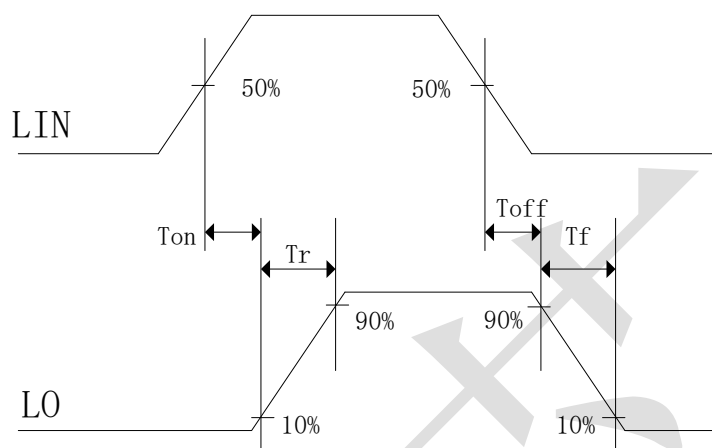


图 27-19 LND36C01 输出 LO 开关时间

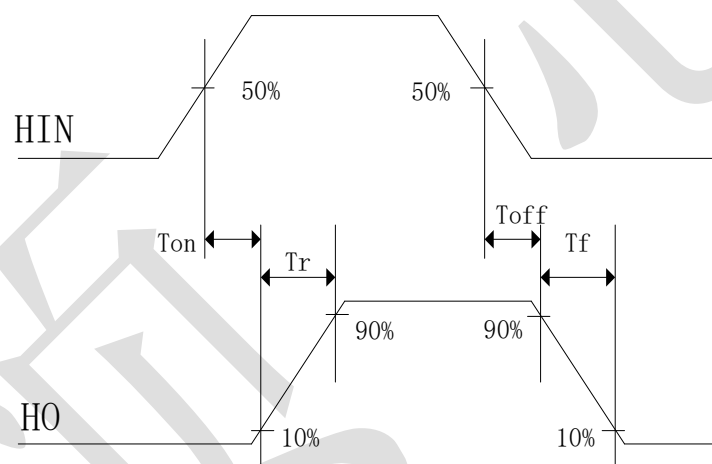


图 27-20 LND36C01 输出 HO 开关时间

27.5.2 LND36C01 VCC 端电源电压

针对不同的 MOS 管，选择不同的驱动电压，开启 MOS 管推荐电源 VCC 工作电压典型值为 8V-15V。

27.5.3 LND36C01 逻辑信号要求和特性

LND36C01 主要功能有逻辑信号输入处理、死区时间控制、电平转换功能、悬浮自举电源结构和上下桥图腾柱式输出。逻辑信号输入端高电平阈值为 2.5V 以上，低电平阈值为 1.0V 以下，要求逻辑信号的输出电流小，可以使 MCU 输出逻辑信号直接连接到 LND36C01 的输入通道上。

高端上桥臂和低端下桥臂输出驱动器的最大灌入可达 0.6A 和最大输出电流可达 1.2A，高端上桥臂通道可以承受 600V 的电压，输入逻辑信号与输出控制信号之间的传导延时小，低端输出开通传导延时为 140ns，关断传导延时为 150ns，高端输出开通传导延时为 140ns，关断传导延时为 150ns。低端输出开通的上升时间为 35ns，关断的下降时间为 25ns，高端输出开通的上升时间为 35ns，关断的下降时间为 25ns。

输入信号和输出信号逻辑功能如图 27-21 所示。

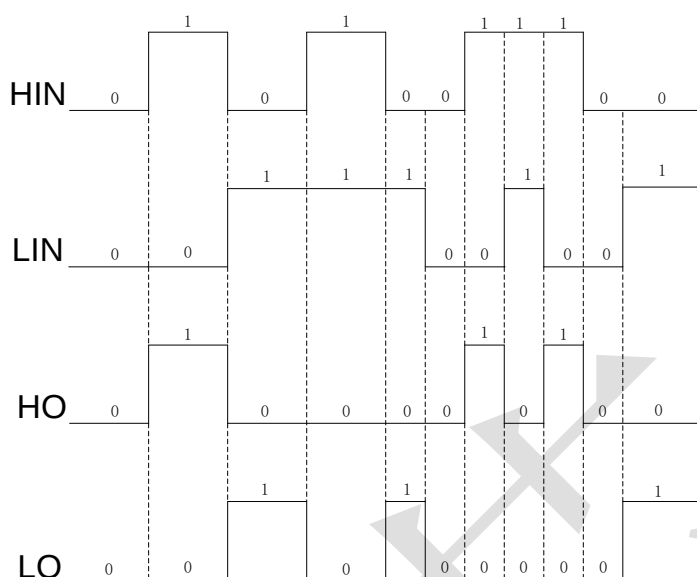


图 27-21 LND36C01 信号的逻辑功能

表 27-3 LND36C01 信号逻辑真值表

输入		输出	
输入、输出逻辑			
HIN	LIN	HO	LO
0	0	0	0
0	1	0	1
1	0	1	0
1	1	0	0

从真值表可知，当输入逻辑信号 HIN 为“1”和 LIN 为“0”时，驱动器控制输出 HO 为“1”上管打开，LO 为“0”下管关断；当输入逻辑信号 HIN 为“0”和 LIN 为“1”时，驱动器控制输出 HO 为“0”上管关断，LO 为“1”下管打开；在输入逻辑信号 HIN 为“1”和 LIN 为“1”或者 HIN 为“0”和 LIN 为“0”时，驱动器控制输出 HO、LO 为“0”将上、下功率管同时关断；内部逻辑处理器杜绝控制器输出上、下功率管同时导通，具有相互闭锁功能。

27.6 预驱 LND33A01

LND33A01 是一款高性价比的大功率 MOS 管的三相独立栅极驱动专用芯片，内部集成了逻辑信号输入处理电路、死区时控制电路、欠压保护电路、闭锁电路、电平位移电路、脉冲滤波电路、自举二极管、5V LDO 和 12V LDO 电路及输出驱动电路。非常适合于 12V 和 24V 的三相电机应用中。

LND33A01 高端的工作电压可达 70V，低端 VCC 的电源电压范围宽 5V~20V。该芯片具有闭锁功能防止输出功率管同时导通，输入通道 HIN 和 LIN 内建了下拉电阻，在输入悬空时使上、下功率 MOS 管处于关闭状态，输出电流能力 IO +1.5A/-1.8A。

特性：

-
- 高端悬浮自举电源，耐压可达 70V
 - 集成自举二极管
 - 电源电压 VM 输入范围：5.5V-65V
 - 适应 3.3V-5.5V 输入电压
 - 内置 5V/20mA LDO
 - 集成 12V LDO 控制电路
 - 驱动电流能力 IO +/- 1.5A/1.8A
 - 内置 VCC 和 VBS 欠压保护
 - 内置 170ns 死区时间
 - 自带闭锁功能
 - 温度范围-40~125℃
 - HIN 输入通道高电平有效，控制高端 HO 输出
 - LIN 输入通道高电平有效，控制低端 LO 输出



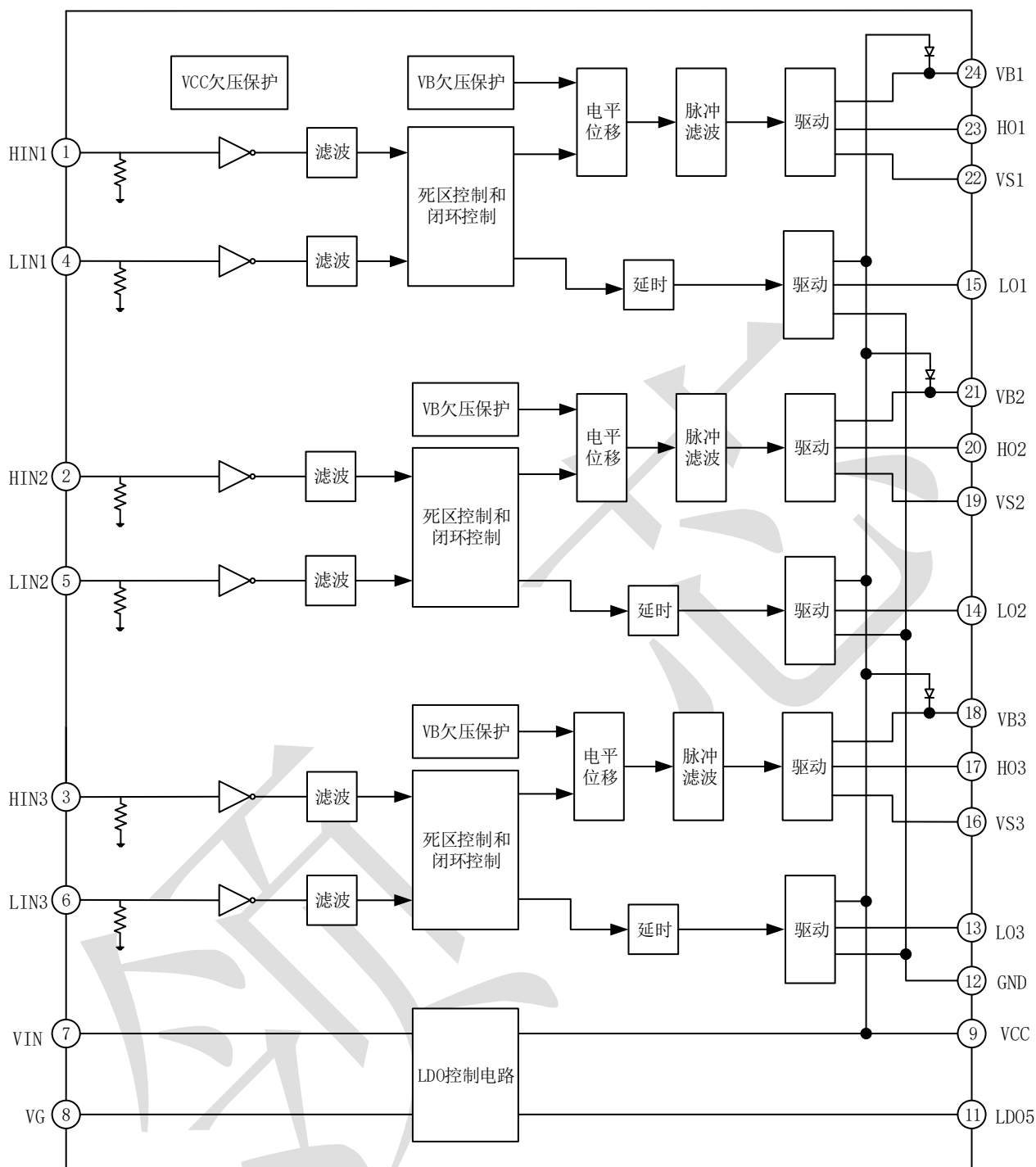


图 27-22 预驱 LND33A01 结构框图

27.6.1 LND33A01 开关特性和死区时间

如下图所示， T_{on} 为开关开延时， T_{off} 为开关关延时， T_r 为上升时间， T_f 为下降时间， DT 为死区时间，具体数值请参考说明书中的相应电气特性。

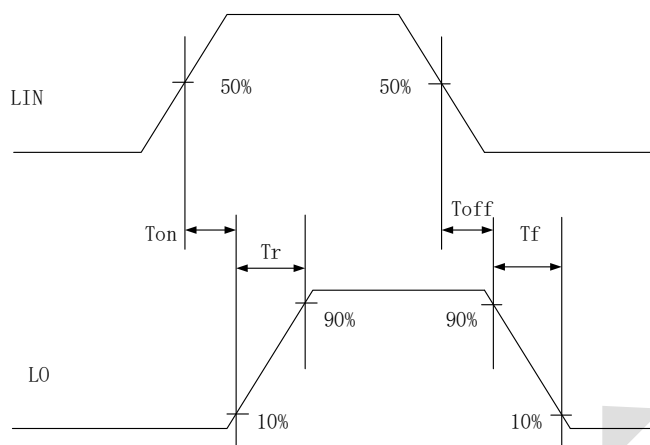


图 27-23 LND33A01 输出 LO 开关时间

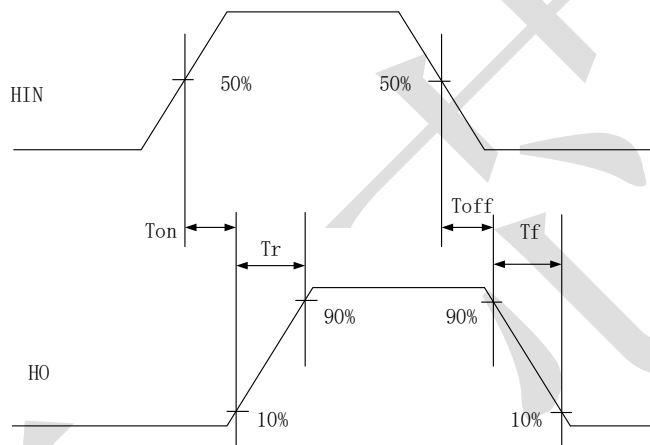


图 27-24 LND33A01 输出 HO 开关时间

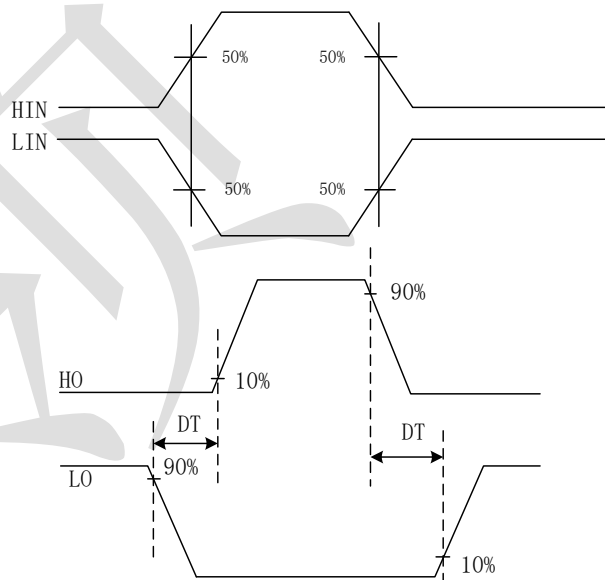


图 27-25 LND33A01 死区时间

27.6.2 LND33A01 VCC 端电源电压

VCC 为低边电路电源供应端，能为输入逻辑电路和低边输出功率级工作提供所需的驱动能量。内置的欠压锁定电路能保证芯片工作在足够高的电源电压范围，进而防止由于低驱动电压所产生的热耗散对 MOSFET/IGBT 造成损害。如图 27-26 所示，当 VCC 上升并超过阈值电压 $VCC_R = 6.7V$ 后，低边控制电路解

锁并开始工作，LO 开始输出；反之，VCC 下降并低于阈值电压 $VCC_F = 6.4V$ 后，低边电路锁定，芯片停止工作，LO 停止输出。VCC 工作电压范围建议为 10.0V-15.0V。

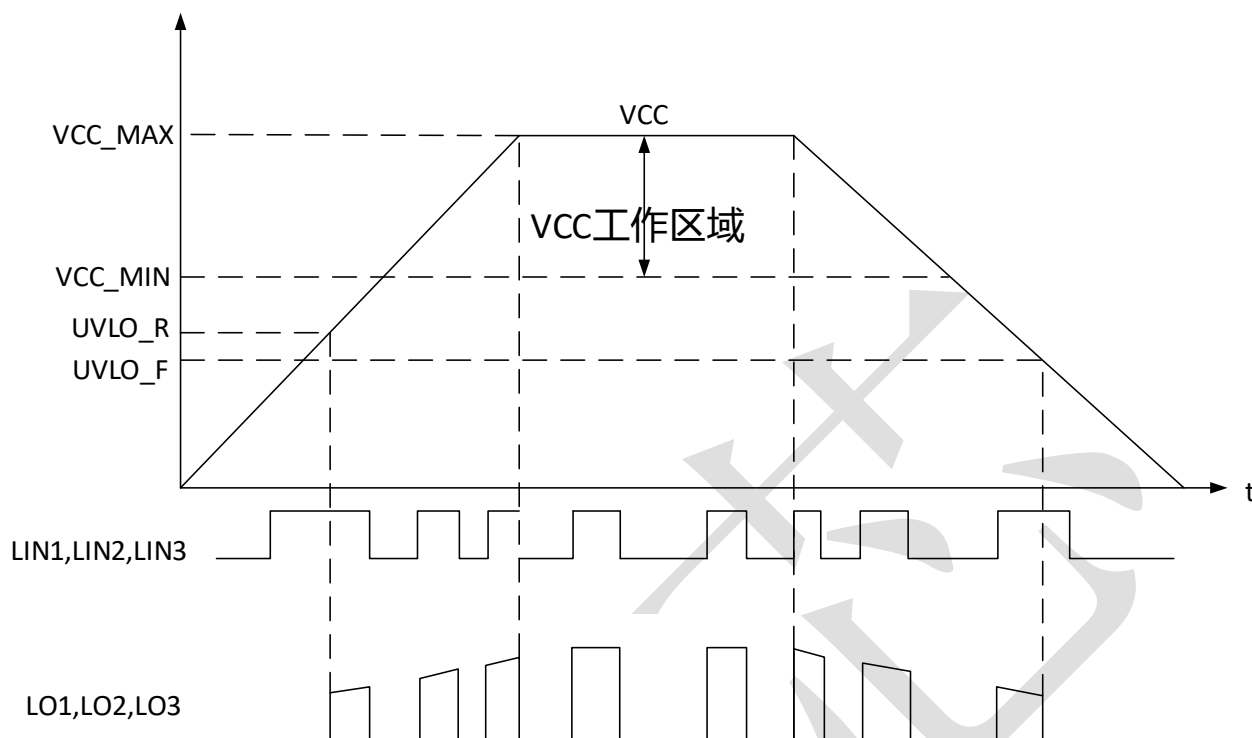


图 27-26 LND33A01 VCC 电源 UVLO 工作区域

27.6.3 LND33A01 VB 端电源电压

VB 电源为高边电路供电电源，其中 VBS1 (VB1-VS1)、VBS2 (VB2-VS2) 和 VBS3 (VB3-VS3) 分别对应相 1、相 2 和相 3 高边驱动电源。由浮动电源 VBS 供电的整体高边电路以地 GND 为参考点，并跟随外部功率管 MOSFET/IGBT 的源/发射极电压，在地线和母线电压之间摆动。由于高边电路具有低静态电流消耗，因此整个高边电路可以由与 VCC 连接的自举电路技术供电，并且只需一个较小的电容就能维持驱动功率管所需电压。如图 27-27 所示，高边电源 VBS 的欠压锁定类似于低边 VCC 电源，VBS 工作电压范围建议在 10.0V-15V。

高端上桥臂和低端下桥臂输出驱动器的最大灌入可达 1.8A 和最大输出电流可达 1.5A，高端上桥臂通道可以承受 65V 的电压，输入逻辑信号与输出控制信号之间的传导延时小，低端输出开通传导延时为 200ns，关断传导延时为 60ns，高端输出开通传导延时为 200ns，关断传导延时为 60ns。低端输出开通的上升时间为 30ns，关断的下降时间为 20ns，高端输出开通的上升时间为 30ns，关断的下降时间为 20ns。

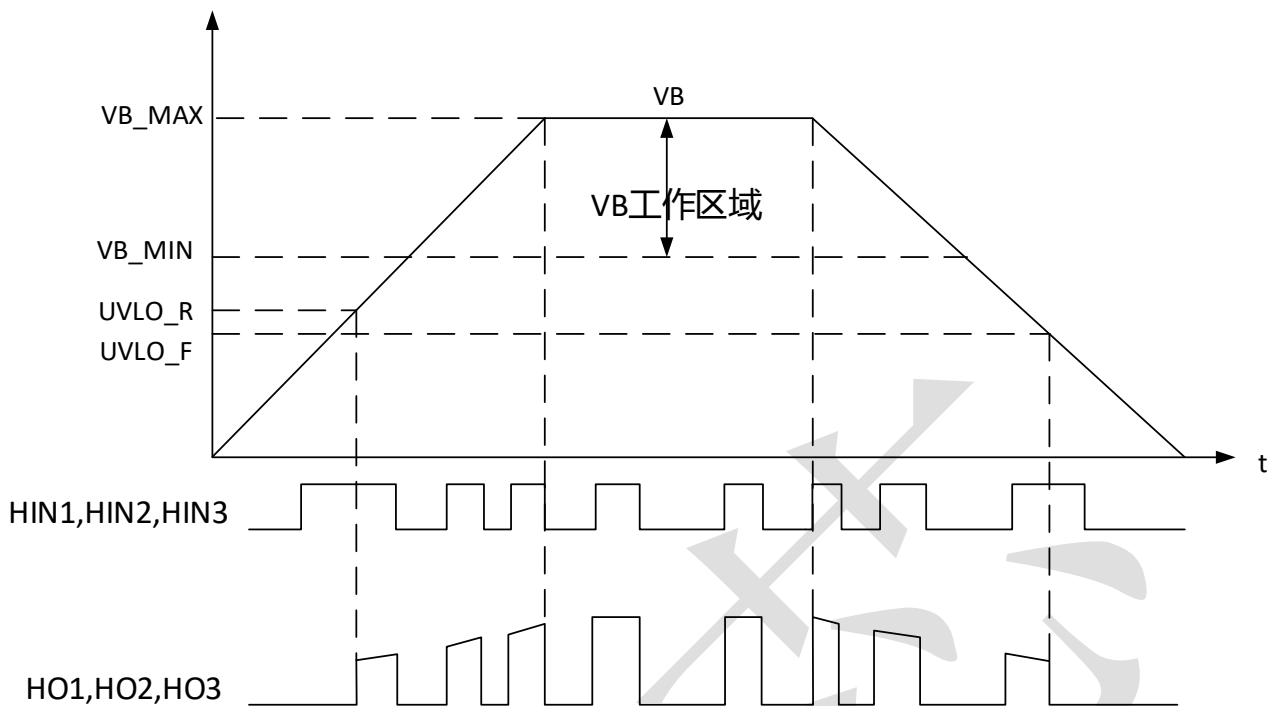


图 27-27 LND33A01 VBS 电源 UVLO 工作区域

27.6.4 LND33A01 逻辑信号要求和特性

LND33A01 主要功能有逻辑信号输入处理、死区时间控制、电平转换功能、悬浮自举电源结构和上下桥图腾柱式输出。逻辑信号输入端高电平阈值为 2.0V 以上，低电平阈值为 0.8V 以下，要求逻辑信的输出电流小，可以使 MCU 输出逻辑信号直接连接到 LND33A01 的输入通道上。

输入信号和输出信号逻辑功能图如下：

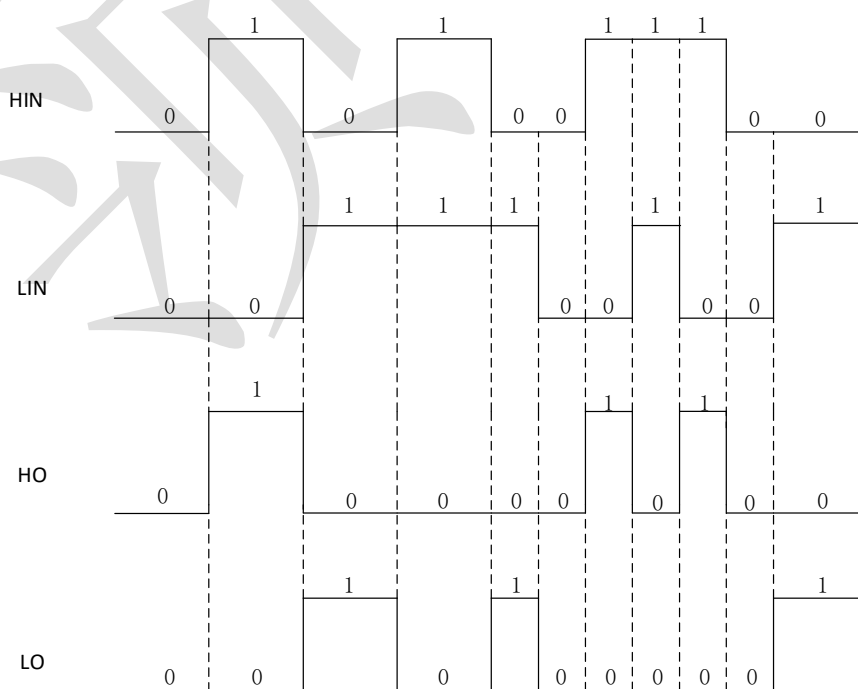


图 27-28 LND33A01 信号的逻辑功能

表 27-4 LND33A01 信号逻辑真值表

输入		输出	
HIN	LIN	HO	LO
0	0	0	0
0	1	0	1
1	0	1	0
1	1	0	0

从真值表可知，当输入逻辑信号 HIN 为“1”和 LIN 为“0”时，驱动器控制输出 HO 为“1”上管打开，LO 为“0”下管关断；当输入逻辑信号 HIN 为“0”和 LIN 为“1”时，驱动器控制输出 HO 为“0”上管关断，LO 为“1”下管打开；在输入逻辑信号 HIN 为“1”和 LIN 为“1”或者 HIN 为“0”和 LIN 为“0”时，驱动器控制输出 HO、LO 为“0”将上、下功率管同时关断；内部逻辑处理器杜绝控制器输出上、下功率管同时导通，具有相互闭锁功能。

28. 修订历史

版本	修订日期	变更内容	修订人
2.0	2024.09.07	初始版本	Li
2.1	2024.11.06	产品型号调整	Liu
2.2	2024.12.09	TIM 章节调整	Liu
2.3	2025.02.14	全文调整	Liu

