RX32S632 参考手册

文档编号：RM00010

基于 32 位 RISC 内核的专业电机微控制器

版本：V2.1

目录

1 简介 24
2 文档约定 25
2.1 寄存器缩写列表 25
2.2 词汇表 25
3 系统和存储器概述 26
3.1 系统架构 26
3.2 存储器结构 26
3.2.1 介绍 26
3.2.2 存储器映像和寄存器边界地址 27
3.3 内置 SRAM 29
3.4 Flash 概述 29
3.5 Boot 配置 29
4 内置 Flash 存储器（FLASH） 30
4.1 简介 30
4.2 Flash 主要特征 30
4.3 Flash 功能描述 30
4.3.1 Flash 结构 30
4.3.2 读取等待周期 30
4.3.3 指令预取 31
4.3.4 Flash 编程 31
4.3.5 解锁 Flash 31
4.3.6 Flash 擦除流程 32
4.3.7 FLASH 写入流程 34
4.4 保护 35
4.4.1 读保护 35
4.4.2 写保护 35
4.5 选项字节 35
4.5.1 选项字节描述 35
4.5.2 选项字节擦除 36
4.5.3 选项字节写入 36
4.5.4 选项字节组织（Option Byte Organization） 36
4.5.5 用户和读保护选项字节 36
4.5.6 写保护 A 区地址选项字节 37
4.5.7 写保护 B 区地址选项字节 38
4.6 Flash 寄存器 38
4.6.1 Flash 访问控制寄存器（FLASH_ACR） 38
4.6.2 FPEC 键值寄存器（FLASH_KEYR） 39
4.6.3 FLASH OPTKER 寄存器（FLASH_OPTKEYR） 39
4.6.4 FLASH 状态寄存器（FLASH_SR） 39
4.6.5 Flash 控制寄存器（FLASH_CR） 40
4.6.6 Flash 地址寄存器（FLASH_OPTR） 41

4.6.1 写保护 A 区寄存器（FLASH_WRP1AR） 43
4.6.2 写保护 B 区寄存器（FLASH_WRP2BR） 43
5 CRC 计算单元 44
5.1 CRC 简介 44
5.2 CRC 主要特性 44
5.3 CRC 功能说明 44
5.4 CRC 寄存器 45
5.4.1 数据寄存器（CRC_DR） 45
5.4.2 独立数据寄存器（CRC_IDR） 45
5.4.3 控制寄存器（CRC_CR） 45
6 电源控制（PWR） 46
6.1 电源 46
6.2 电源管理器 46
6.2.1 上电复位（POR）和掉电复位（PDR） 46
6.2.2 可编程电压检测器（PVD） 47
6.3 低功耗模式 48
6.3.1 运行模式 49
6.3.2 睡眠模式 49
6.3.3 停止 0 模式 49
6.3.4 待机模式 49
6.4 低功耗模式下的自动唤醒（AWU） 49
6.5 电源控制寄存器 50
6.5.1 电源控制寄存器 1（PWR_CR1） 50
6.5.2 电源控制寄存器 2（PWR_CR2） 50
6.5.3 电源控制寄存器 3（PWR_CR3） 51
6.5.4 电源控制寄存器 4（PWR_CR4） 51
6.5.5 电源状态寄存器 1（PWR_SR1） 52
6.5.6 电源状态寄存器 2（PWR_SR2） 52
6.5.7 电源状态控制寄存器（PWR_SCR） 53
7 复位和时钟控制（RCC） 54
7.1 复位 54
7.1.1 系统复位 54
7.1.2 电源复位 54
7.2 时钟 55
7.2.1 HSI 时钟 56
7.2.2 PLL 56
7.2.3 LSI 时钟 56
7.2.4 系统时钟（SYSCLK）选择 57
7.2.5 RTC 时钟 57
7.2.6 看门狗时钟 57
7.2.7 时钟输出 57
7.3 RCC 寄存器 58
7.3.1 时钟控制寄存器（RCC_CR） 58

7.3.2 时钟配置寄存器（RCC_CFGR） 59
7.3.3 时钟中断寄存器（RCC_CIR） 61
7.3.4 APB2 外设复位寄存器（RCC_APB2RSTR） 62
7.3.5 APB1 外设复位寄存器（RCC_APB1RSTR） 63
7.3.6 AHB 外设时钟使能寄存器（RCC_AHBENR） 64
7.3.7 APB2 外设时钟使能寄存器（RCC_APB2ENR） 65
7.3.8 APB1 外设时钟使能寄存器（RCC_APB1ENR） 66
7.3.9 备份域控制寄存器（RCC_BDCR） 67
7.3.10 控制/状态寄存器（RCC_CSR） 67
7.3.11 AHB 外设复位寄存器（RCC_ AHBRSTR） 68
8 通用和复用功能 I/O（GPIO） 70
8.1 简介 70
8.2 GPIO 主要特征 70
8.3 GPIO 功能描述 70
8.3.1 通用 IO（GPIO） 72
8.3.2 IO 引脚复用功能 72
8.3.3 I/O 端口控制寄存器 73
8.3.4 I/O 端口数据寄存器 73
8.3.5 I/O 数据位处理 73
8.3.6 I/O 锁定机制 74
8.3.7 I/O 复用功能输入/输出 74
8.3.8 外部中断/唤醒线 74
8.3.9 输入模式配置 74
8.3.10 输出模式配置 75
8.3.11 复用功能配置 76
8.3.12 模拟功能配置 77
8.4 GPIO 寄存器 78
8.4.1 GPIO 端口模式寄存器（GPIOx_MODER）（x = A to D） 78
8.4.2 GPIO 端口输出类型寄存器（GPIOx_OTYPE）（x = A to D） 78
8.4.3 GPIO 端口输出速度寄存器（GPIOx_OSPEEDR）（x = A to D） 79
8.4.4 GPIO 端口上/下拉寄存器（GPIOx_PUPDR）（x = A to D） 79
8.4.5 GPIO 端口输入数据寄存器（GPIOx_IDR）（x = A to D） 80
8.4.6 GPIO 端口输出数据寄存器（GPIOx_ODR）（x = A to D） 80
8.4.7 GPIO 端口位置位/清零寄存器（GPIOx_BSRR）（x = A to D） 80
8.4.8 GPIO 端口配置锁寄存器（GPIOx_LCKR）（x = A to D） 81
8.4.9 GPIO 复用功能低位寄存器（GPIOx_AFRL）（x = A to D） 81
8.4.10 GPIO 端口位复位寄存器（GPIOx_BRR）（x = A to D） 82
9 系统配置控制器（SYSCFG） 83
9.1 SYSCFG 主要特征 83
9.2 SYSCFG 寄存器 83
9.2.1 外部中断配置寄存器 1（SYSCFG_EXTICR1） 83
9.2.2 外部中断配置寄存器 2（SYSCFG_EXTICR2） 83
9.2.3 CYSCFG 配置寄存器 2（SYSCFG_CFGR2） 84

10 中断和事件控制器（EXTI） 85
10.1 简介 85
10.2 框图 85
10.3 唤醒事件管理 85
10.4 功能说明 86
10.5 外部中断/事件线路映像 86
10.6 EXTI 寄存器 88
10.6.1 中断屏蔽寄存器 1（EXTI_IMR1） 88
10.6.2 事件屏蔽寄存器 1（EXTI_EMR1） 88
10.6.3 上升沿触发选择寄存器 1（EXTI_RTSR1） 89
10.6.4 下降沿触发选择寄存器 1（EXTI_FTSR1） 89
10.6.5 软件中断事件寄存器 1（EXTI_SWIER1） 90
10.6.6 挂起寄存器 1（EXTI_PR1） 90
11 内部参考电压缓冲器（VREFBUF） 92
11.1 简介 92
11.2 功能描述 92
11.2.1 切换 VREFBUF 电压档位步骤 93
11.2.2 内部电压切换到外部电压步骤 93
11.3 VREFBUF 校准 93
11.4 VREFBUF 寄存器 93
11.4.1 VREFBUF 状态寄存器（CSR） 93
11.4.2 VREFBUF 控制寄存器（CCR） 94
12 模拟数字转换器（ADC） 95
12.1 简介 95
12.2 ADC 主要特征 95
12.3 ADC 功能描述 95
12.3.1 ADC 引脚/通道 96
12.3.2 ADC 开关控制 97
12.3.3 ADC 时钟 97
12.3.4 通道选择 97
12.3.5 ADC 部分寄存器的配置步骤 98
12.3.6 单次转换模式 98
12.3.7 连续转换模式 98
12.3.8 时序图 99
12.3.9 模拟看门狗 99
12.3.10 扫描模式 101
12.3.11 注入通道管理 101
12.3.12 间断模式 102
12.4 校准 103
12.5 数据对齐 103
12.6 可编程的通道采样时间 104
12.7 外部触发转换 104
12.8 温度传感器（TPS） 105

12.9 ADC 中断 106
12.10 ADC 寄存器 106
12.10.1 ADC 状态寄存器（ADC_SR） 106
12.10.2 ADC 控制寄存器 1（ADC_CR1） 107
12.10.3 ADC 控制寄存器 2（ADC_CR2） 110
12.10.4 ADC 采样时间寄存器（ADC_SMPR1） 112
12.10.5 ADC 采样时间寄存器（ADC_SMPR2） 112
12.10.6 ADC 注入通道数据偏移寄存器 x（ADC_JOFRx）（x=1..4） 113
12.10.7 ADC 看门狗高阈值寄存器（ADC_HTR） 113
12.10.8 ADC 看门狗低阈值寄存器（ADC_LTR） 113
12.10.9 ADC 规则序列寄存器 1（ADC_SQR1） 114
12.10.10 ADC 规则序列寄存器 2（ADC_SQR2） 114
12.10.11 ADC 规则序列寄存器 3（ADC_SQR3） 115
12.10.12 ADC 注入序列寄存器（ADC_JSQR） 115
12.10.13 ADC 注入数据寄存器（ADC_JDRx（x=1..4）） 115
12.10.14 ADC DR 数据寄存器（ADC_DR） 116
12.10.15 ADC 控制寄存器（ADC_CxCR） 116
12.10.16 ADC 校准寄存器（ADC_CAL） 116
12.10.17 ADC EOC 控制寄存器（ADC_EOC） 117
12.10.18 ADC TPS 校准寄存器（TPS） 117
12.10.19 ADC 采样时间寄存器（ADC_SMPR0） 117
12.10.20 ADC 规则数据寄存器（ADC_DRx（x=1..8）） 118
12.10.21 ADC 控制寄存器 3（ADC_CR3） 118
13 高级控制定时器（TIM8） 119
13.1 TIM8 简介 119
13.2 TIM8 主要特征 119
13.3 TIM8 功能描述 120
13.3.1 框图 120
13.3.2 TIM8 引脚和内部信号 121
13.3.3 时基单元 123
13.3.4 计数模式 125
13.3.5 重复计数器 135
13.3.6 外部触发输入 136
13.3.7 时钟选择 137
13.3.8 捕获比较通道 139
13.3.9 强制输出模式 140
13.3.10 输出比较模式 140
13.3.11 PWM 模式 141
13.3.12 非对称 PWM 模式 149
13.3.13 组合 PWM 模式 152
13.3.14 三相组合 PWM 模式 153
13.3.15 互补输出和死区插入 154
13.3.16 用刹车功能 157

13.3.17 双向刹车输入 161
13.3.18 在外部事件时清除 OCxREF 信号 162
13.3.19 生成 6 步 PWM 164
13.3.20 单脉冲模式 165
13.3.21 可重触发单脉冲模式 166
13.3.22 在比较模式下的脉冲 167
13.3.23 方向位输出 169
13.3.24 UIF 位重映射 169
13.3.25 定时器同步 169
13.3.26 ADC 触发器 172
13.3.27 调试模式 173
13.4 TIM8 低功耗模式 173
13.5 TIM8 中断 173
13.6 TIM8 寄存器 174
13.6.1 TIM8 控制寄存器 1（TIM8_CR1） 174
13.6.2 TIM8 控制寄存器 2（TIM8_CR2） 175
13.6.3 TIM8 从模式控制寄存器（TIM8_SMCR） 178
13.6.4 TIM8/DMA/中断使能寄存器（TIM8_DIER） 180
13.6.5 TIM8 状态寄存器（TIM8_SR） 181
13.6.6 TIM8 事件产生寄存器（TIM8_EGR） 183
13.6.7 TIM8 捕获/比较模式寄存器 1（TIM8_CCMR1） 184
13.6.8 TIM8 捕获/比较模式寄存器 2（TIM8_CCMR2） 186
13.6.9 TIM8 捕获/比较使能寄存器（TIM8_CCER） 188
13.6.10 TIM8 计数器（TIM8_CNT） 190
13.6.11 TIM8 预分频器（TIM8_PSC） 191
13.6.12 TIM8 自动重装载寄存器（TIM8_ARR） 191
13.6.13 TIM8 重复计数寄存器（TIM8_RCR） 192
13.6.14 TIM8 捕获/比较寄存器 1（TIM8_CCR1） 192
13.6.15 TIM8 捕获/比较寄存器 2（TIM8_CCR2） 193
13.6.16 TIM8 捕获/比较寄存器 3（TIM8_CCR3） 193
13.6.17 TIM8 捕获/比较寄存器 4（TIM8_CCR4） 194
13.6.18 TIM8 刹车和死区寄存器（TIM8_BDTR） 194
13.6.19 TIM8 捕获/比较寄存器 5（TIM8_CCR5） 198
13.6.20 TIM8 捕获/比较寄存器 6（TIM8_CCR6） 199
13.6.21 TIM8 捕获/比较模式寄存器 3（TIM8_CCMR3） 199
13.6.22 TIM8 定时器死区寄存器 2（TIM8_DTR2） 200
13.6.23 TIM8 定时器编码器控制寄存器（TIM8_ECR） 201
13.6.24 TIM8 复用功能选择寄存器 1（TIM8_AF1） 201
13.6.25 TIM8 复用功能选择寄存器 2（TIM8_AF2） 203
13.6.26 TIM8 捕获/比较寄存器 1N（TIM8_CCR1N） 204
13.6.27 TIM8 捕获/比较寄存器 2N（TIM8_CCR2N） 205
13.6.28 TIM8 捕获/比较寄存器 3N（TIM8_CCR3N） 205
13.6.29 TIM8 刹车输入控制选择（TIM8_BICS） 206

13.6.30 TIM8 触发寄存器（TIM8_TRGI） 207
14 通用定时器（TIM2/TIM3） 208
14.1 TIM2/TIM3 简介 208
14.2 TIM2/TIM3 主要功能 208
14.3 TIM2/TIM3 实现 208
14.4 TIM2/TIM3 功能描述 209
14.4.1 框图 209
14.4.2 TIM2/TIM3 引脚和内部信号 211
14.4.3 时基单元 213
14.4.4 计数器模式 215
14.4.5 时钟选择 225
14.4.6 捕获/比较通道 228
14.4.7 输入捕获模式 230
14.4.8 PWM 输入模式 231
14.4.9 强制输出模式 232
14.4.10 输出比较模式 232
14.4.11 PWM 模式 233
14.4.12 非对称 PWM 模式 241
14.4.13 组合 PWM 模式 245
14.4.14 在外部事件时清除 tim_ocxref 信号 246
14.4.15 单脉冲模式 247
14.4.16 可重触发单脉冲模式 249
14.4.17 在比较模式下的脉冲 249
14.4.18 编码器接口模式 252
14.4.19 方向位输出 269
14.4.20 UIF 位重映射 270
14.4.21 定时器输入异或功能 270
14.4.22 定时器和外部触发的同步 270
14.4.23 定时器同步 273
14.4.24 ADC 触发 278
14.4.25 调试模式 278
14.5 TIM2/TIM3 低功耗模式 279
14.6 TIM2/TIM3 中断 279
14.7 TIMx 寄存器 279
14.7.1 TIMx 控制寄存器 1（TIMx_CR1）（x=2/3） 279
14.7.2 TIMx 控制寄存器 2（TIMx_CR2）（x=2/3） 281
14.7.3 TIMx 从模式控制寄存器（TIMx_SMCR）（x=2/3） 282
14.7.4 TIMx 中断使能寄存器（TIMx_DIER）（x=2/3） 285
14.7.5 TIMx 状态寄存器（TIMx_SR）（x=2/3） 286
14.7.6 TIMx 事件产生寄存器（TIMx_EGR）（x=2/3） 288
14.7.7 TIMx 捕获/比较模式寄存器 1 [复用]（TIMx_CCMR1）（x=2/3） 289
14.7.8 TIMx 捕获/比较模式寄存器 1 [复用]（TIMx_CCMR1）（x=2/3） 290
14.7.9 TIMx 捕获/比较模式寄存器 2 [复用]（TIMx_CCMR2）（x=2/3） 292

14.7.10 TIMx 捕获/比较模式寄存器 2 [复用]（TIMx_CCMR2）（x=2/3） 293
14.7.11 TIMx 捕获/比较使能寄存器（TIMx_CCER）（x=2/3） 294
14.7.12 TIM3 计数器（TIM3_CNT） 296
14.7.13 TIM2 计数器（TIM2_CNT） 296
14.7.14 TIMx 预分频器（TIMx_PSC）（x=2/3） 297
14.7.15 TIM3 自动重装载寄存器（TIM3_ARR） 297
14.7.16 TIM2 自动重装载寄存器（TIM2_ARR） 297
14.7.17 TIM3 捕获/比较寄存器 1（TIM3_CCR1） 298
14.7.18 TIM2 捕获/比较寄存器 1（TIM2_CCR1） 299
14.7.19 TIM3 捕获/比较寄存器 2（TIM3_CCR2） 299
14.7.20 TIM2 捕获/比较寄存器 2（TIM2_CCR2） 300
14.7.21 TIM3 捕获/比较寄存器 3（TIM3_CCR3） 301
14.7.22 TIM2 捕获/比较寄存器 3（TIM2_CCR3） 301
14.7.23 TIM3 捕获/比较寄存器 4（TIM3_CCR4） 302
14.7.24 TIM2 捕获/比较寄存器 4（TIM2_CCR4） 303
14.7.25 TIMx 编码器控制寄存器（TIMx_ECR）（x=2/3） 303
14.7.26 TIMx 定时器输入选择寄存器（TIMx_TISEL）（x=2/3） 304
14.7.27 TIMx 复用功能选择寄存器 1（TIMx_AF1）（x=2/3） 305
14.7.28 TIMx 复用功能选择寄存器 2（TIMx_AF2）（x=2/3） 306
14.7.29 TIMx 触发寄存器（TIMx_TRGI）（x=2/3） 306
15 通用定时器（TIM15） 307
15.1 TIM15 简介 307
15.2 TIM15 主要功能 307
15.3 TIM15 功能描述 308
15.3.1 框图 308
15.3.2 TIM15 引脚和内部信号 309
15.3.3 时基单元 311
15.3.4 计数器模式 313
15.3.5 重复计数器 316
15.3.6 时钟选择 317
15.3.7 捕获/比较通道 319
15.3.8 输入捕获模式 321
15.3.9 PWM 输入模式 322
15.3.10 强制输出模式 323
15.3.11 输出比较模式 323
15.3.12 PWM 模式 324
15.3.13 组合 PWM 模式 329
15.3.14 互补输出和死区插入 330
15.3.15 使用刹车功能 333
15.3.16 双向刹车输入 337
15.3.17 在外部事件时清除 tim_ocxref 信号 338
15.3.18 生成 6 步 PWM 339
15.3.19 单脉冲模式 339

15.3.20 可重触发单脉冲模式 341
15.3.21 UIF 位重映射 342
15.3.22 定时器输入异或功能 342
15.3.23 定时器和外部触发的同步 342
15.3.24 定时器同步 345
15.3.25 ADC 触发 345
15.3.26 调试模式 345
15.4 TIM15 低功耗模式 345
15.5 TIM15 中断 345
15.6 TIM15 寄存器 346
15.6.1 TIM15 控制寄存器 1（TIM15_CR1） 346
15.6.2 TIM15 控制寄存器 2（TIM15_CR2） 347
15.6.3 TIM15 从模式控制寄存器（TIM15_SMCR） 349
15.6.4 TIM15 中断使能寄存器（TIM15_DIER） 350
15.6.5 TIM15 状态寄存器（TIM15_SR） 351
15.6.6 TIM15 事件产生寄存器（TIM15_EGR） 352
15.6.7 TIM15 捕获/比较模式寄存器 1 [复用]（TIM15_CCMR1） 353
15.6.8 TIM15 捕获/比较模式寄存器 1 [复用]（TIM15_CCMR1） 355
15.6.9 TIM15 捕获/比较模式寄存器 2 [复用]（TIM15_CCMR2） 357
15.6.10 TIM15 捕获/比较模式寄存器 2 [复用]（TIM15_CCMR2） 358
15.6.11 TIM15 捕获/比较使能寄存器（TIM15_CCER） 358
15.6.12 TIM15 计数器（TIM15_CNT） 361
15.6.13 TIM15 预分频器（TIM15_PSC） 361
15.6.14 TIM15 自动重装载寄存器（TIM15_ARR） 362
15.6.15 TIM15 重复计数寄存器（TIM15_RCR） 362
15.6.16 TIM15 捕获/比较寄存器 1（TIM15_CCR1） 363
15.6.17 TIM15 捕获/比较寄存器 2（TIM15_CCR2） 364
15.6.18 TIM15 捕获/比较寄存器 3（TIM15_CCR3） 365
15.6.19 TIM15 刹车和死区寄存器（TIM15_BDTR） 365
15.6.20 TIM15 定时器死区寄存器 2（TIM15_DTR2） 368
15.6.21 TIM15 定时器输入选择寄存器（TIM15_TISEL） 369
15.6.22 TIM15 复用功能选择寄存器 1（TIM15_AF1） 369
15.6.23 TIM15 复用功能选择寄存器 2（TIM15_AF2） 370
15.6.24 TIM15 触发寄存器（TIM15_TRGI） 371
16 基本定时器（TIM6/TIM7） 372
16.1 TIM6/TIM7 简介 372
16.2 TIM6/TIM7 主要功能 372
16.3 TIM6/TIM7 功能描述 372
16.3.1 框图 372
16.3.2 TIM6/TIM7 内部信号 373
16.3.3 TIM6/TIM7 时钟 373
16.3.4 时基单元 373
16.3.5 计数器模式 375

16.3.6 UIF 位重映射 379
16.3.7 ADC 触发 379
16.3.8 调试模式 379
16.4 TIM6/TIM7 低功耗模式 380
16.5 TIM6/TIM7 中断 380
16.6 TIMx 寄存器 381
16.6.1 TIMx 控制寄存器 1（TIMx_CR1）（x=6/7） 381
16.6.2 TIMx 控制寄存器 2（TIMx_CR2）（x=6/7） 382
16.6.3 TIMx 中断使能寄存器（TIMx_DIER）（x=6/7） 382
16.6.4 TIMx 状态寄存器（TIMx_SR）（x=6/7） 382
16.6.5 TIMx 事件产生寄存器（TIMx_EGR）（x=6/7） 383
16.6.6 TIMx 计数器（TIMx_CNT）（x=6/7） 383
16.6.7 TIMx 预分频器（TIMx_PSC）（x=6/7） 384
16.6.8 TIMx 自动重装载寄存器（TIMx_ARR）（x=6/7） 384
17 运算放大器（OPAMP） 385
17.1 简介 385
17.2 运算放大器特征 385
17.3 OPAMP 模式 385
17.3.1 独立模式（仅 OPAMP3）： 385
17.3.2 电压跟随器模式 386
17.3.3 PGA 模式（P-IO，N-IO） 387
17.3.4 PGA 模式（P-IO，N-GND） 388
17.3.5 OPAMPx P 端偏置电压的选择 390
17.4 运算放大器寄存器 392
17.4.1 运算放大器 OPAMP1 控制寄存器（OPAMP1_CSR） 392
17.4.2 运算放大器 OPAMP2 控制寄存器（OPAMP2_CSR） 393
17.4.3 运算放大器 OPAMP3 控制寄存器（OPAMP3_CSR） 394
17.4.4 运算放大器 OPAMP1 偏置寄存器（OPAMP1_BIAS） 395
17.4.5 运算放大器 OPAMP2 偏置寄存器（OPAMP2_BIAS） 396
17.4.6 运算放大器 OPAMP3 偏置寄存器（OPAMP3_BIAS） 396
18 比较器（CMP） 398
18.1 简介 398
18.2 比较器特征 398
18.3 比较器开关控制 398
18.4 比较器器输入和输出 398
18.5 比较器锁定机制 398
18.6 比较器框图 399
18.7 比较器防误触发功能 400
18.8 比较器迟滞功能 401
18.9 比较器遮蔽功能 401
18.10 比较器寄存器 403
18.10.1 CMPx 控制寄存器（CMP_CxCR）（x=1/2） 403
18.10.2 CMPx 校正寄存器（CMP_RGxCAL）（x=1/2） 406

18.10.3 CMP 控制寄存器 1（CMP_CR1） 406
18.10.4 CMP 控制寄存器 2（CMP_CR2） 407
19 数学运算协同处理器（ME） 409
19.1 简介 409
19.2 功能描述 409
19.3 ME 寄存器 409
19.3.1 ME 控制寄存器（ME_CR） 409
19.3.2 ME ARG1/RES1 寄存器（ARG1/RES1） 410
19.3.3 ME ARG2/RES2 寄存器（ARG2/RES2） 410
20 实时时钟（RTC） 411
20.1 RTC 简介 411
20.2 主要特性 411
20.3 功能描述 411
20.3.1 概述 411
20.3.2 复位过程 412
20.3.3 读 RTC 寄存器 412
20.3.4 配置 RTC 寄存器 413
20.3.5 RTC 标志的设置 413
20.4 RTC 寄存器 414
20.4.1 RTC 控制寄存器高位（RTC_CRH） 414
20.4.2 RTC 控制寄存器低位（RTC_CRL） 414
20.4.3 RTC 预分频装载寄存器（RTC_PRLH/RTC_PRLL） 416
20.4.4 RTC 预分频器余数寄存器（RTC_DIVH / RTC_DIVL） 416
20.4.5 RTC 计数器寄存器（RTC_CNTH/RTC_CNTL） 417
20.4.6 RTC 闹钟寄存器（RTC_ALRH/RTC_ALRL） 418
21 独立看门狗（IWDG） 419
21.1 简介 419
21.2 IWDG 主要性能 419
21.3 IWDG 功能描述 419
21.3.1 硬件看门狗 419
21.3.2 寄存器访问保护 419
21.3.3 调试模式 419
21.4 IWDG 寄存器 420
21.4.1 键寄存器（IWDG_KR） 420
21.4.2 预分频寄存器（IWDG_PR） 421
21.4.3 重装载寄存器（IWDG_RLR） 421
21.4.4 状态寄存器（IWDG_SR） 422
22 串行外设接口（SPI） 423
22.1 SPI 简介 423
22.2 SPI 主要特征 423
22.2.1 SPI 特征 423
22.3 SPI 功能描述 423
22.3.1 概述 423

22.3.2 配置 SPI 为从模式 428
22.3.3 配置 SPI 为主模式 428
22.3.4 配置 SPI 通信方式 429
22.3.5 数据发送与接收过程 430
22.3.6 状态标志 435
22.3.7 关闭 SPI 436
22.3.8 错误标志 436
22.3.9 SPI 中断 437
22.4 SPI 寄存器 437
22.4.1 SPI 控制寄存器 1（SPI_CR1） 437
22.4.2 SPI 控制寄存器 2（SPI_CR2） 439
22.4.3 SPI 状态寄存器（SPI_SR） 439
22.4.4 SPI 数据寄存器（SPI_DR） 440
23 I2C 接口（I2C） 441
23.1 I²C 简介 441
23.2 I²C 主要特点 441
23.3 I²C 功能描述 441
23.3.1 模式选择 442
23.3.2 I²C 从模式 443
23.3.3 I²C 主模式 445
23.3.4 错误条件 448
23.3.5 SDA/SCL 线控制 449
23.4 I²C 中断请求 450
23.5 I²C 寄存器 451
23.5.1 控制寄存器 1（I2C_CR1） 451
23.5.2 控制寄存器 2（I2C_CR2） 452
23.5.3 自身地址寄存器 1（I2C_OAR1） 453
23.5.4 自身地址寄存器 2（I2C_OAR2） 454
23.5.5 数据寄存器（I2C_DR） 454
23.5.6 状态寄存器 1（I2C_SR1） 454
23.5.7 状态寄存器 2（I2C_SR2） 457
23.5.8 时钟控制寄存器（I2C_CCR） 458
23.5.9 TRISE 寄存器（I2C_TRISE） 459
24 通用异步收发器（UART） 460
24.1 UART 介绍 460
24.2 UART 主要特性 460
24.3 UART 功能概述 461
24.3.1 UART 特性描述 462
24.3.2 发送器 462
24.3.3 接收器 465
24.3.4 分数波特率的产生 468
24.3.5 UART 接收器容忍时钟的变化 469
24.3.6 校验控制 470

24.3.7 LIN（局域互联网）模式 470
24.3.8 单线半双工通信 472
24.4 UART 中断请求 473
24.5 UART 模式配置 474
24.1 UART 寄存器 474
24.1.1 状态寄存器（UART_SR） 474
24.1.2 数据寄存器（UART_DR） 476
24.1.3 波特比率寄存器（UART_BRR） 476
24.1.4 控制寄存器 1（UART_CR1） 477
24.1.5 控制寄存器 2（UART_CR2） 478
24.1.6 控制寄存器 3（UART_CR3） 479
25 栅极驱动器（GateDriver） 480
25.1 概述 480
25.2 功能描述 480
25.2.1 特性 480
25.3 HO/LO 输出真值表 481
26 器件电子签名 482
26.1 产品唯一身份标识寄存器（96 位） 482
27 调试支持（DBG） 484
27.1 DBGMCU 寄存器 484
27.1.1 微控制器设备 ID 编码寄存器（DBGMCU_IDCODE） 484
27.1.2 调试 MCU 配置寄存器（DBGMCU_CR） 484
28 版本历史 486

表目录

表 3.1 RX32S632 系列存储器映像和外设边界地址 28
表 3.2 Boot 模式 29
表 4.1 Flash 结构 30
表 4.2 等待周期和 CPU 时钟频率对应关系 31
表 4.3 选项字节格式 35
表 6.1 低功耗模式一览 48
表 8.1 端口位配置表 (1) 72
表 11.1 VREFBUF 模式 92
表 12.1 ADC 引脚 96
表 12.2 ADC1 通道 97
表 12.3 模拟看门狗通道选择 100
表 12.4 规则通道的外部触发信号 104
表 12.5 注入通道的外部触发信号 105
表 12.6 ADC 中断 106
表 13.1 TIM 输入/输出引脚 121
表 13.2 TIM 内部输入/输出信号 121
表 13.3 TIM8 内部触发连接 122
表 13.4 连接到 tim_etr 的输入多路复用器 122
表 13.5 定时器刹车互连 122
表 13.6 定时器刹车 2 互连 122
表 13.7 系统中断互连 122
表 13.8 连接到 ocref_clr 的输入多路复用器 123
表 13.9 CCR 和 ARR 寄存器的变化抖动模式 148
表 13.10 CCR 寄存器在中央对齐 PWM 模式下的变化抖动模式 149
表 13.11 刹车保护解除条件 162
表 13.12 低功耗模式对 TIM8 的影响 173
表 13.13 中断请求 173
表 13.14 带刹车功能的互补通道 tim_ocx 和 tim_ocxn 的输出控制位 190
表 14.1 RX32S632 系列通用定时器 208
表 14.2 TIM 输入/输出引脚 211
表 14.3 TIM 内部输入/输出信号 211
表 14.4 连接到 tim_ti1 的输入多路复用器 212
表 14.5 连接到 tim_ti2 的输入多路复用器 212
表 14.6 连接到 tim_ti3 的输入多路复用器 212
表 14.7 连接到 tim_ti4 的输入多路复用器 212
表 14.8 TIMx 内部触发连接 212
表 14.9 连接到 tim_etr 的输入多路复用器 213
表 14.10 连接到 ocref_clr 的输入多路复用器 213
表 14.11 CCR 和 ARR 寄存器的变化抖动模式 240
表 14.12 CCR 寄存器在中央对齐 PWM 模式下的变化抖动模式 241
表 14.13 计数方向与编码器信号的关系（CC1P = CC2P = 0） 253

表 14.14 计数方向与编码器信号及极性设置的关系 257
表 14.15 低功耗模式对 TIM2/TIM3 的影响 279
表 14.16 中断请求 279
表 14.17 标准 tim_ocx 通道的输出控制位 295
表 15.1 TIM15 输入/输出引脚 309
表 15.2 TIM15 内部输入/输出信号 309
表 15.3 连接到 tim_ti1 的输入多路复用器 309
表 15.4 连接到 tim_ti2 的输入多路复用器 310
表 15.5 连接到 tim_ti3 的输入多路复用器 310
表 15.6 TIM15 内部触发连接 310
表 15.7 TIM15 中断互连 310
表 15.8 系统中断互连 310
表 15.9 连接到 ocref_clr 的输入多路复用器 310
表 15.10 CCR 和 ARR 寄存器的变化抖动模式 329
表 15.11 刹车保护解除条件 337
表 15.12 低功耗模式对 TIM15 的影响 345
表 15.13 中断请求 346
表 15.14 带刹车功能的互补输出通道 tim_oc1 和 tim_oc1n 的控制位（TIM15） 360
表 16.1 TIM 内部输入/输出信号 373
表 16.2 低功耗模式对 TIM6/TIM7 的影响 380
表 16.3 中断请求 380
表 18.1 CMP/ OPAMP 和 ADC 的 IO 对应关系 400
表 18.2 遮蔽源 401
表 19.1 函数运算表 409
表 21.1 看门狗超时时间（32KHz 的输入时钟（LSI））(1) 420
表 22.1 SPI 中断请求 437
表 23.1 I²C 中断请求表： 450
表 24.1 检测噪声的数据采样 467
表 24.2 设置波特率时的误差计算 469
表 24.3 当 DIV_Fraction=0 时，UART 接收器的容忍度 469
表 24.4 当 DIV_Fraction!=0 时，UART 接收器的容忍度 469
表 24.5 帧格式 470
表 24.6 UART 中断请求 473
表 24.7 UART 模式设置 (1) 474
表 28.1 版本历史 486

图目录

图 3.1 存储器映像 27
图 4.1 页擦流程 32
图 4.2 全擦流程 33
图 4.3 FLASH 写入流程 34
图 5.1 CRC 计算单元框图 44
图 6.1 电源框图 46
图 6.2 上电复位和掉电复位波形图 47
图 6.3 PVD 门限 47
图 7.1 复位电路框图 55
图 7.2 时钟树 56
图 8.1 I/O 端口位基本结构 71
图 8.2 输入浮空/上拉/下拉结构 75
图 8.3 输出框图 76
图 8.4 复用功能框图 77
图 8.5 模拟引脚框图 77
图 10.1 外部中断/事件控制器框图 85
图 10.2 外部中断通用 I/O 映像 87
图 11.1 VREFBUF 框图 92
图 12.1 ADC1 框图 96
图 12.2 ADC 转换时序图 99
图 12.3 模拟看门狗警戒区 100
图 12.4 注入转换延迟 102
图 12.5 校准时序图 103
图 12.6 数据右对齐 104
图 12.7 数据左对齐 104
图 12.8 温度传感器和 VBG 通道框图 106
图 13.1 高级控制定时器框图 120
图 13.2 当预分频器的参数从 1 变成 2 时，计数器的时序图 124
图 13.3 当预分频器的参数从 1 变成 4 时，计数器的时序图 124
图 13.4 计数器时序图，内部时钟分频因子为 1 125
图 13.5 计数器时序图，内部时钟分频因子为 2 126
图 13.6 计数器时序图，内部时钟分频因子为 4 126
图 13.7 计数器时序图，内部时钟分频因子为 N 127
图 13.8 计数器时序图，当 ARPE=0 时的更新事件（TIM8_ARR 没有预装入） 127
图 13.9 计数器时序图，当 ARPE=1 时的更新事件（预装入了 TIM8_ARR） 128
图 13.10 计数器时序图，内部时钟分频因子为 1 129
图 13.11 计数器时序图，内部时钟分频因子为 2 129
图 13.12 计数器时序图，内部时钟分频因子为 4 130
图 13.13 计数器时序图，内部时钟分频因子为 N 130
图 13.14 计数器时序图，当没有使用重复计数器时的更新事件 131
图 13.15 计数器时序图，内部时钟分频因子为 1，TIM8_ARR=0x6 132

图 13.16 计数器时序图，内部时钟分频因子为 2 133
图 13.17 计数器时序图，内部时钟分频因子为 4，TIM8_ARR=0x36 133
图 13.18 计数器时序图，内部时钟分频因子为 N 134
图 13.19 计数器时序图，ARPE=1 时的更新事件（计数器下溢） 135
图 13.20 计数器时序图，ARPE=1 时的更新事件（计数器上溢） 135
图 13.21 不同模式下更新速率的例子，及 TIM8_RCR 的寄存器设置 136
图 13.22 外部触发输入框图 137
图 13.23 一般模式下的控制电路，内部时钟分频因子为 1 137
图 13.24 外部触发输入框图 138
图 13.25 外部时钟模式 2 下的控制电路 138
图 13.26 捕获/比较通道 1 的主电路 139
图 13.27 捕获/比较通道的输出阶段（通道 1，通道 2、3 和 4 同上） 139
图 13.28 捕获/比较通道的输出阶段（通道 5，通道 6 同上） 140
图 13.29 输出比较模式，翻转 OC1 141
图 13.30 边沿对齐的 PWM 波形（ARR=8） 142
图 13.31 中央对齐的 PWM 波形（APR=8） 143
图 13.32 抖动原理 144
图 13.33 抖动模式下的数据格式和寄存器编码 145
图 13.34 PWM 分辨率 vs 频率 146
图 13.35 PWM 抖动模式 147
图 13.36 中央对齐 PWM 模式下抖动对占空比的影响 148
图 13.37 生成两个具有 50%占空比的相位偏移 PWM 信号 150
图 13.38 边沿对齐向上计数的 PWM 波形（ARR=12） 150
图 13.39 边沿对齐向下计数的 PWM 波形（ARR=12） 151
图 13.40 中央对齐的 PWM 波形（ARR=8） 152
图 13.41 通道 1 和通道 3 的组合 PWM 模式 153
图 13.42 每周期具有多个触发脉冲的三相组合 PWM 信号 154
图 13.43 带死区插入的互补输出 155
图 13.44 非对称死区时间 156
图 13.45 死区波形延迟大于负脉冲 156
图 13.46 死区波形延迟大于正脉冲 157
图 13.47 刹车和刹车 2 电路概述 158
图 13.48 tim_brk 在发生刹车事件时的各种输出行为（OSSI=1） 160
图 13.49 当 tim_brk/tim_brk2 有效（OSSI=0）时 PWM 输出状态 161
图 13.50 输出重定向（未表示 tim_brk2 请求） 162
图 13.51 tim_ocref_clr 输入选择复用器 163
图 13.52 清除 TIM8 的 tim_ocxref 164
图 13.53 COM 事件生成 6 步 PWM 的示例（OSSR=1） 164
图 13.54 单脉冲模式的例子 165
图 13.55 可重触发单脉冲模式 166
图 13.56 脉冲发生器电路 167
图 13.57 在比较事件上的脉冲生成，适用于边沿对齐模式 168
图 13.58 在同时触发的情况下延长脉冲宽度 169

图 13.59 复位模式下的控制电路 170
图 13.60 门控模式下的控制电路 171
图 13.61 触发器模式下的控制电路 172
图 14.1 通用定时器 TIM2 框图 209
图 14.2 通用定时器 TIM3 框图 210
图 14.3 当预分频器的参数从 1 变到 2 时，计数器的时序图 214
图 14.4 当预分频器的参数从 1 变到 4 时，计数器的时序图 214
图 14.5 计数器时序图，内部时钟分频因子为 1 215
图 14.6 计数器时序图，内部时钟分频因子为 2 216
图 14.7 计数器时序图，内部时钟分频因子为 4 216
图 14.8 计数器时序图，内部时钟分频因子为 N 217
图 14.9 计数器时序图，当 ARPE=0 时的更新事件（TIMx_ARR 没有预装入） 217
图 14.10 计数器时序图，当 ARPE=1 时的更新事件（预装入了 TIMx_ARR） 218
图 14.11 计数器时序图，内部时钟分频因子为 1 219
图 14.12 计数器时序图，内部时钟分频因子为 2 219
图 14.13 计数器时序图，内部时钟分频因子为 4 220
图 14.14 计数器时序图，内部时钟分频因子为 N 220
图 14.15 计数器时序图，更新事件 221
图 14.16 计数器时序图，内部时钟分频因子为 1，TIMx_ARR=0x6 222
图 14.17 计数器时序图，内部时钟分频因子为 2 223
图 14.18 计数器时序图，内部时钟分频因子为 4，TIMx_ARR=0x36 223
图 14.19 计数器时序图，内部时钟分频因子为 N 224
图 14.20 计数器时序图，ARPE=1 时的更新事件（计数器下溢） 225
图 14.21 计数器时序图，ARPE=1 时的更新事件（计数器溢出） 225
图 14.22 一般模式下的控制电路，内部时钟分频因子为 1 226
图 14.23 tim_ti2 外部时钟连接例子 226
图 14.24 外部时钟模式 1 下的控制电路 227
图 14.25 外部触发输入框图 227
图 14.26 外部时钟模式 2 下的控制电路 228
图 14.27 捕获/比较通道（如：通道 1 输入部分） 229
图 14.28 捕获/比较通道 1 的主电路 229
图 14.29 捕获/比较通道的输出阶段（通道 1，通道 2、3 和 4 同上） 230
图 14.30 PWM 输入模式时序 232
图 14.31 输出比较模式，翻转 tim_oc1 233
图 14.32 边沿对齐的 PWM 波形（ARR=8） 234
图 14.33 中央对齐的 PWM 波形（APR=8） 235
图 14.34 抖动原理 236
图 14.35 抖动模式下的数据格式和寄存器编码 237
图 14.36 PWM 分辨率 vs 频率（16 位模式） 238
图 14.37 PWM 分辨率 vs 频率（28 位模式） 238
图 14.38 PWM 抖动模式 239
图 14.39 中央对齐 PWM 模式下抖动对占空比的影响 240
图 14.40 生成两个具有 50%占空比的相位偏移 PWM 信号 242

图 14.41 边沿对齐向上计数的 PWM 波形（ARR=12）..... 242
图 14.42 边沿对齐向下计数的 PWM 波形（ARR=12）..... 243
图 14.43 中央对齐的 PWM 波形（ARR=8）..... 244
图 14.44 通道 1 和通道 3 的组合 PWM 模式..... 245
图 14.45 tim_ocref_clr 输入选择复用器..... 246
图 14.46 清除 TIMx 的 tim_ocxref..... 247
图 14.47 单脉冲模式的例子..... 248
图 14.48 可重触发单脉冲模式..... 249
图 14.49 脉冲发生器电路..... 250
图 14.50 在比较事件上的脉冲生成，适用于边沿对齐模式和编码器模式..... 251
图 14.51 在同时触发的情况下延长脉冲宽度..... 252
图 14.52 编码器接口模式下的计数器工作示例..... 253
图 14.53 tim_ti1fp1 极性反相时的编码器接口模式示例..... 254
图 14.54 正交编码器的计数模式..... 254
图 14.55 方向加时钟编码器模式..... 255
图 14.56 方向时钟编码器模式（CC1P=CC2P=0）..... 256
图 14.57 方向时钟编码器模式（CC1P=CC2P=1）..... 256
图 14.58 索引门控选项..... 257
图 14.59 抖动的索引信号..... 258
图 14.60 IPOS[1:0] = 11 时的索引生成..... 259
图 14.61 通道 A 上带索引门控的计数器读取（IPOS[1:0] = 11）..... 259
图 14.62 非门控模式下索引的计数器读取（IPOS[1:0] = 00）..... 260
图 14.63 通道 A 和 B 上带索引门控的计数器读取..... 260
图 14.64 窄索引脉冲情况下的编码器模式行为（IPOS[1:0] = 11）..... 261
图 14.65 计数器重置窄索引脉冲（更近距离，ARR = 0x07）..... 262
图 14.66 x1 和 x2 模式下的索引行为（IPOS[1:0] = 01）..... 263
图 14.67 定向索引灵敏度..... 264
图 14.68 作为 FIDX 位设置的功能的计数器重置..... 264
图 14.69 时钟+方向模式下的索引行为，IPOS[0] = 1..... 265
图 14.70 方向时钟模式下的索引行为，IPOS[0] = 1..... 265
图 14.71 正交编码信号的状态图..... 266
图 14.72 递增计数编码器错误检测..... 267
图 14.73 递减计数编码器错误检测..... 268
图 14.74 更新时带有预加载转移的编码器模式更改（SMSPS = 0）..... 269
图 14.75 复位模式下的控制电路..... 271
图 14.76 门控模式下的控制电路..... 271
图 14.77 触发器模式下的控制电路..... 272
图 14.78 外部时钟模式 2＋触发模式下的控制电路..... 273
图 14.79 主/从定时器的例子..... 274
图 14.80 仅定时器通道 1 主/从连接示例..... 274
图 14.81 TIM_mstr 的 tim_oc1ref 控制 TIM_slv..... 275
图 14.82 通过使能 TIM_mstr 可以控制 TIM_slv..... 276
图 14.83 使用 TIM_mstr 的更新触发 TIM_slv..... 277

图 14.84 利用 TIM_mstr 的使能触发 TIM_slv 277
图 14.85 使用 TIM_mstr 的 tim_ti1 输入触发 TIM_mstr 和 TIM_slv 278
图 15.1 TIM15 框图 308
图 15.2 当预分频器的参数从 1 变到 2 时，计数器的时序图 312
图 15.3 当预分频器的参数从 1 变到 4 时，计数器的时序图 312
图 15.4 计数器时序图，内部时钟分频因子为 1 313
图 15.5 计数器时序图，内部时钟分频因子为 2 314
图 15.6 计数器时序图，内部时钟分频因子为 4 314
图 15.7 计数器时序图，内部时钟分频因子为 N 315
图 15.8 计数器时序图，当 ARPE=0 时的更新事件（TIM15_ARR 没有预装入） 315
图 15.9 计数器时序图，当 ARPE=1 时的更新事件（预装入了 TIM15_ARR） 316
图 15.10 取决于模式和 TIM15_RCR 寄存器设置的更新频率示例 317
图 15.11 一般模式下的控制电路，内部时钟分频因子为 1 318
图 15.12 tim_ti2 外部时钟连接例子 318
图 15.13 外部时钟模式 1 下的控制电路 319
图 15.14 捕获/比较通道（如：通道 1 输入部分） 320
图 15.15 捕获/比较通道 1 的主电路 320
图 15.16 捕获/比较通道的输出阶段（通道 1） 320
图 15.17 捕获比较通道的输出阶段（通道 2，通道 3 同上） 321
图 15.18 PWM 输入模式时序 323
图 15.19 输出比较模式，翻转 tim_oc1 324
图 15.20 边沿对齐的 PWM 波形（ARR=8） 325
图 15.21 抖动原理 326
图 15.22 抖动模式下的数据格式和寄存器编码 327
图 15.23 PWM 分辨率 vs 频率 328
图 15.24 PWM 抖动模式 328
图 15.25 通道 1 和通道 2 的组合 PWM 模式 330
图 15.26 带死区插入的互补输出 331
图 15.27 非对称死区时间 332
图 15.28 死区波形延迟大于负脉冲 332
图 15.29 死区波形延迟大于正脉冲 333
图 15.30 刹车电路概述 334
图 15.31 tim_brk 在发生刹车事件时的各种输出行为（OSSI=1） 336
图 15.32 输出重定向 338
图 15.33 tim_ocref_clr 输入选择复用器 338
图 15.34 COM 事件生成 6 步 PWM 的示例（OSSR=1） 339
图 15.35 单脉冲模式的例子 340
图 15.36 可重触发单脉冲模式 341
图 15.37 测量两个信号边沿之间的间隔时间 342
图 15.38 复位模式下的控制电路 343
图 15.39 门控模式下的控制电路 344
图 15.40 触发器模式下的控制电路 344
图 16.1 基本定时器框图 372

图 16.2 一般模式下的控制电路，内部时钟分频因子为 1 373
图 16.3 当预分频器的参数从 1 变到 2 时，计数器的时序图 374
图 16.4 当预分频器的参数从 1 变到 4 时，计数器的时序图 375
图 16.5 计数器时序图，内部时钟分频因子为 1 376
图 16.6 计数器时序图，内部时钟分频因子为 2 376
图 16.7 计数器时序图，内部时钟分频因子为 4 377
图 16.8 计数器时序图，内部时钟分频因子为 N 377
图 16.9 计数器时序图，当 ARPE=0 时的更新事件（TIMx_ARR 没有预装入） 378
图 16.10 计数器时序图，当 ARPE=1 时的更新事件（预装入了 TIMx_ARR） 379
图 17.1 OPAMP3 独立模式 385
图 17.2 OPAMP1/OPAMP2 电压跟随器模式 386
图 17.3 OPAMP3 电压跟随器模式 386
图 17.4 OPAMP1/OPAMP2 PGA 模式（P-IO，N-IO） 387
图 17.5 OPAMP3 PGA 模式（P-IO，N-IO） 388
图 17.6 OPAMP1/OPAMP2 PGA 模式（P-IO，N-GND） 389
图 17.7 OPAMP3 PGA 模式（P-IO，N-GND） 389
图 17.8 OPAMP1/OPAMP2 PGA 模式（P-偏置，N-GND） 391
图 17.9 OPAMP3 PGA 模式（P-偏置，N-GND） 391
图 18.1 比较器框图 399
图 18.2 比较器迟滞 401
图 18.3 比较器遮蔽功能 402
图 20.1 简化的 RTC 框图 412
图 20.2 RTC 秒和闹钟波形图示例，PR=0003，ALARM=00004 413
图 20.3 RTC 溢出波形图示例，PR=0003 414
图 21.1 独立看门狗框图 420
图 22.1 SPI 框图 424
图 22.2 单主和单从应用 425
图 22.3 硬件/软件的从选择管理 426
图 22.4 数据时钟时序图 427
图 22.5 主模式、全双工模式下（BIDIMODE=0 并且 RXONLY=0）连续传输时，TXE/RXNE/BSY 的变化示意图 432
图 22.6 从模式、全双工模式下（BIDIMODE=0 并且 RXONLY=0）连续传输时，TXE/RXNE/BSY 的变化示意图 432
图 22.7 主设备只发送模式（BIDIMODE=0 并且 RXONLY=0）下连续传输时，TXE/BSY 变化示意图 433
图 22.8 从设备只发送模式（BIDIMODE=0 并且 RXONLY=0）下连续传输时，TXE/BSY 变化示意图 433
图 22.9 只接收模式（BIDIMODE=0 并且 RXONLY=1）下连续传输时，RXNE 变化示意图 434
图 22.10 非连续传输发送（BIDIMODE=0 并且 RXONLY=0）时，TXE/BSY 变化示意图 435
图 23.1 I²C 总线协议 442
图 23.2 I²C 的功能框图 443
图 23.3 从发送器的传送序列图 444
图 23.4 从接收器的传送序列图 445

图 23.5 主发送器传送序列图 447
图 23.6 主接收器传送序列图 448
图 23.7 I²C 中断映射图 450
图 24.1 UART 框图 461
图 24.2 字长设置 462
图 24.3 配置停止位 463
图 24.4 发送时 TC/TXE 的变化情况 464
图 24.5 起始位侦测 465
图 24.6 检测噪声的数据采样 467
图 24.7 LIN 模式下的断开检测（11 位断开长度 – 设置了 LBDL 位） 471
图 24.8 LIN 模式下的断开检测与帧错误的检测 472
图 24.9 UART 中断映像图 473
图 25.1 GateDriver 内部结构 480
图 25.2 输入输出时序波形 481

1 简介

本参考手册用于帮助开发者理解并使用睿兴科技（南京）有限公司（后文简称“睿兴”或“RX”）的芯片。本文提供了 RX32S632 系列微控制器的存储和外设的完整信息。

相关封装和电气特性请参考相应的数据手册。

本参考手册和数据手册均可从睿兴官网 www.rxtek-icore.com 获得

2 文档约定

2.1 寄存器缩写列表

read/write（rw）	该位可读可写。
read-only（r）	该位只读。
write-only（w）	该位只写。
read/clear write0 (rc_w0)	软件可以读取这个位，并且可以通过写入 0 来清零这个位。写入 1 对这个位没有影响。
read/clear（rc_w1）	软件可以读此位，也可以通过写 1 清除此位，写 0 对此位无影响。
read/set (rs)	软件可以读取和设置这个位。写入 0 对这个位没有影响。

2.2 词汇表

Quad world：四字，128 位长度数据。
Double world：双字，64 位长度数据。
Word：字，32 位长度数据。
Half-word：半字，16 位长度数据。
Byte：字节，8 位长度数据

3 系统和存储器概述

3.1 系统架构

RX32S632 系列集成了 32 位 RISC 内核，是电机专用的高性能、低功耗微控制器。

- 一个驱动单元
  - 32 位 RISC 内核及先进高性能总线
- 三个被动单元
  - 内置 Flash
  - 内置 SRAM
  - AHB 外设包括 AHB 到 APB 的桥和 APB 外设（连接在 APB1 和 APB2）

3.2 存储器结构

3.2.1 介绍

程序存储器，数据存储器，寄存器和 I/O 口统一编址在 4G 的线性地址空间里。

字节在存储器中以小端格式编码。一个字中编号最低的字节被认为是该字的最低有效字节，最高地址字节是最高有效字节。

3.2.2 存储器映像和寄存器边界地址

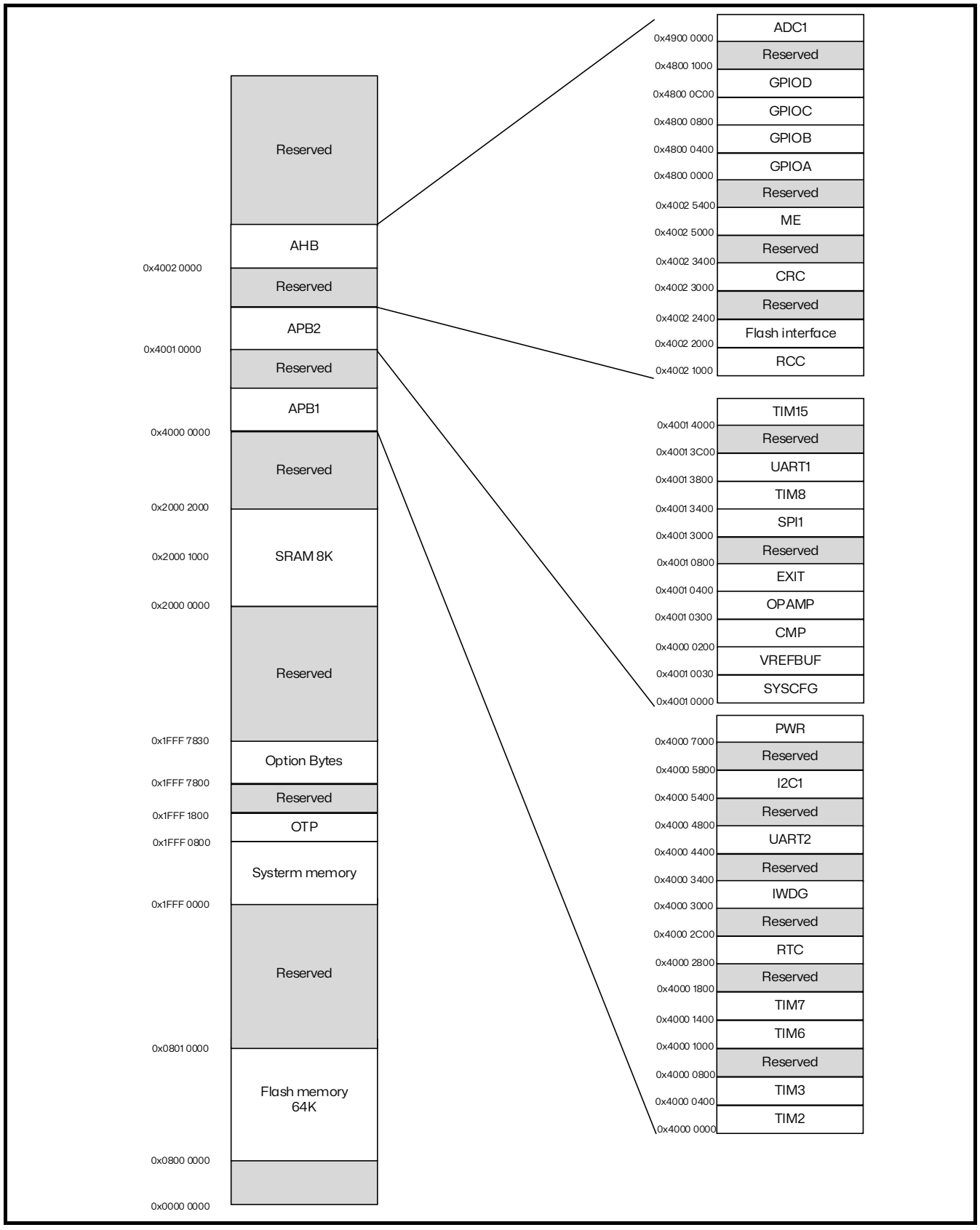


图 3.1 存储器映像

所有未分配给片上存储器和外围设备的内存映射区域都被配置为“Reserved”或者“保留”。有关可用内存和寄存器区域的详细映射，请参阅下表。

下表列出了设备中可用外设的边界地址。

表 3.1 RX32S632 系列存储器映像和外设边界地址

总线	边界地址	外设
AHB	0x4900 0000 - 0x4900 03FF	ADC1
	0x4800 1000 - 0x48FF FFFF	保留
	0x4800 0C00 - 0x4800 0FFF	GPIOD
	0x4800 0800 - 0x4800 0BFF	GPIOC
	0x4800 0400 - 0x4800 07FF	GPIOB
	0x4800 0000 - 0x4800 03FF	GPIOA
	0x4002 5400 - 0x47FF FFFF	保留
	0x4002 5000 - 0x4002 53FF	ME
	0x4002 3400 - 0x4002 4FFF	保留
	0x4002 3000 - 0x4002 33FF	CRC
	0x4002 2400 - 0x4002 2FFF	保留
	0x4002 2000 - 0x4002 23FF	Flash interface
	0x4002 1000 - 0x4002 13FF	RCC
APB2	0x4001 4400 - 0x4002 0FFF	保留
	0x4001 4000 - 0x4001 43FF	TIM15
	0x4001 3C00 - 0x4001 3FFF	保留
	0x4001 3800 - 0x4001 3BFF	UART1
	0x4001 3400 - 0x4001 37FF	TIM8
	0x4001 3000 - 0x4001 33FF	SPI1
	0x4001 0800 - 0x4001 2FFF	保留
	0x4001 0400 - 0x4001 07FF	EXTI
	0x4001 0300 - 0x4001 03FF	OPAMP
	0x4001 0200 - 0x4001 02FF	CMP
	0x4001 0030 - 0x4001 01FF	VREFBUF
	0x4001 0000 - 0x4001 0029	SYSCFG
APB1	0x4000 7400 - 0x4000 FFFF	保留
	0x4000 7000 - 0x4000 73FF	PWR
	0x4000 5800 - 0x4000 6FFF	保留
	0x4000 5400 - 0x4000 57FF	I2C1
	0x4000 4800 - 0x4000 53FF	保留
	0x4000 4400 - 0x4000 47FF	UART2
	0x4000 3400 - 0x4000 43FF	保留
	0x4000 3000 - 0x4000 33FF	IWDG
	0x4000 2C00 - 0x4000 2FFF	保留
	0x4000 2800 - 0x4000 2BFF	RTC
	0x4000 1800 - 0x4000 27FF	保留

总线	边界地址	外设
	0x4000 1400 - 0x4000 17FF	TIM7
	0x4000 1000 - 0x4000 13FF	TIM6
	0x4000 0800 - 0x4000 0FFF	保留
	0x4000 0400 - 0x4000 07FF	TIM3
	0x4000 0000 - 0x4000 03FF	TIM2

3.3 内置 SRAM

RX32S632 系列内置总计 8Kb 的 SRAM。

- 8Kb 的 SRAM（映射在 0x2000 0000）

这些 SRAM 可以以字节（8 位）、半字节（16 字节）、字（32 位）进行访问。这些存储器可以在没有等待周期的情况下由 CPU 寻址。

3.4 Flash 概述

Flash 有两个不同的物理区域组成：

- 主 Flash 块。用于存储应用代码和数据。
- 信息块，主要包含三个部分：
  - 选项字节：用于硬件和存储器保护。
  - 系统存储器：包含 RX 专用代码。

Flash 接口基于 AHB 协议进行指令和数据访问。Flash 寄存器具有 Flash 的操作（擦除/写入）功能。

3.5 Boot 配置

RX32S632 的 Boot 配置可以通过配置选项字节的方式实现。

表 3.2 Boot 模式

nBOOT1 FLASH_OPTR[23]	nBOOT0 FLASH_OPTR[27]	nSWBOOT0 FLASH_OPTR[26]	Boot 区域
x	x	1	主 Flash 块
x	1	0	主 Flash 块
0	0	0	SRAM
1	0	0	系统存储区

内置 Bootloader

内置 Bootloader 代码位于系统存储器（映射在 0x1FFF F000）中，用于以下串口接口重新编程 Flash：

- 引脚 PA0/PA1 上的 UART1。

4 内置 Flash 存储器（FLASH）

4.1 简介

Flash 接口管理 CPU 总线对 Flash 进行访问，具有 Flash 擦除和写入功能以及读写保护机制。
Flash 接口通过一个包含指令预取的系统来加速代码的执行。
注意：对 Flash 执行擦写操作时，需要进入临界区（遮蔽所有中断），然后检查待写入区域是否是全“F”，确认待写入区域是全“F”后才可以写入。

4.2 Flash 主要特征

- 总计 64 Kb 的 Flash，128 bit（1×128 bit）的位宽
- Flash 读取/写入支持 128 bit 操作
- 写保护
- 读保护
- 支持页（1Kb）擦除和全擦除
- 指令预取（128 bit）
- 选项字节加载器
- 4 Kb 的一次性可编程存储区（OTP）

4.3 Flash 功能描述

4.3.1 Flash 结构

Flash 包含 64 页（每页 1Kb）的主存储块和信息块，如下表所示。

表 4.1 Flash 结构

模块	名称	地址	大小（字节）
主存储块	页 0	0x0800 0000 - 0x0800 03FF	1K
	页 1	0x0800 0400 - 0x0800 07FF	1K
	页 2	0x0800 0800 - 0x0800 0BFF	1K
	…	…	…
	页 63	0x0800 FC00 - 0x0800 FFFF	1K
信息块	系统存储器	0x1FFF 0000 - 0x1FFF 07FF	2K
	OTP	0x1FFF 0800 – 0x1FFF 17FF	4K
	Option Bytes	0x1FFF 7800 - 0x1FFF 782F	48

4.3.2 读取等待周期

为了正确读取 Flash 的指令和数据，必须根据 CPU 的时钟（HCLK）在 Flash 访问控制寄存器（FLASH_ACR）中配置等待周期（Latency）。等待周期和 CPU 时钟频率的关系如下表所示。

表 4.2 等待周期和 CPU 时钟频率对应关系

等待周期（WS）（Latency）	HCLK（MHz）
1WS（2 CPU cycles）	≤76
2WS（3 CPU cycles）	≤114
3WS（4 CPU cycles）	≤152

复位后，CPU 时钟频率是 16MHz，FLASH_ACR 中默认为 1 等待周期。

修改 CPU 频率时，必须执行以下步骤以保证足够的 Flash 访问等待周期：

增加 CPU 频率：

1. 修改 Flash 访问控制寄存器（FLASH_ACR）中 Latency 的等待周期。
2. 读取并确认 FLASH_ACR 寄存器中的等待周期。
3. 修改 CPU 频率。
4. 确认 CPU 频率。

降低 CPU 频率：

1. 修改 CPU 频率。
2. 确认 CPU 频率。
3. 修改 Flash 访问控制寄存器（FLASH_ACR）中 Latency 的等待周期。
4. 读取并确认 FLASH_ACR 寄存器中的等待周期。

4.3.3 指令预取

预取缓冲区（128bit）：CPU 每次取指最多为 32 位的字，取一条指令时，下一条指令已经在缓冲区中等待，这样 CPU 可以工作在更高的主频。

复位后预取缓冲区默认开启。只有在 SYSCLK 小于 24MHz 且 AHB 没有预分频（即 SYSCLK 必须等于 HCLK）时才可以开启或关闭预取缓冲区。通常情况下，建议在芯片初始化阶开启或关闭缓冲区，此时芯片运行在内部 16MHz 晶振上。

4.3.4 Flash 编程

Flash 支持在线编程和应用编程。

在线编程（in-circuit programming，ICP）是指使用串行调试接口或者 bootloader 更新 Flash 的内容。

应用编程（in-application programming，IAP）是指使用芯片支持的通讯协议更新 Flash 的内容。IAP 允许用户在芯片运行过程中更新 Flash 的内容。然而要实现 IAP，需要预先使用 ICP 将部分应用代码编写进 Flash 中。

如果在 Flash 操作期间复位芯片将无法保证 Flash 的内容正确。

4.3.5 解锁 Flash

复位后，Flash 控制寄存器（Flash control register，FLASH_CR）不可写入，这是为了防止 Flash 由于干扰产生误操作。通过以下流程可以解锁 FLASH_CR：

1. 在 Flash 键值寄存器（Flash key register，FLASH_KEYR）写入 KEY1 = 0x45670123。
2. 在 Flash 键值寄存器（Flash key register，FLASH_KEYR）写入 KEY2 = 0xCDEF89AB。

任何错误的流程将会导致 FLASH_CR 上锁直到系统复位。一旦执行了错误的写入键值流程，将会产

生一个总线错误和一个硬件错误中断。

FLASH_CR 可以通过写入其中的 LOCK 位来上锁。

注意：FLASH_CR 寄存器在 Flash 状态寄存器（Flash status register，FLASH_SR）的 BSY 位置起时不可被写入。在 BSY 位置起时，任何写入动作将会导致 AHB 总线挂起直到 BSY 位被清除。

4.3.6 Flash 擦除流程

Flash 擦除可以分为页擦和全擦。全擦不会擦除信息区（系统存储器和选项字节）。

页擦

页擦流程如下：

1. 确认 FLASH_SR 寄存器的 BSY 位以保证没有正在执行的 Flash 操作。
2. 检查并清除之前的 Flash 操作造成的错误。
3. 将 FLASH_CR 的 PER 位置 1，配置 PNB 选择要擦除的页。
4. 将 FLASH_CR 的 STRT 位置 1。
5. 等 FLASH_SR 中 BSY 被清除。

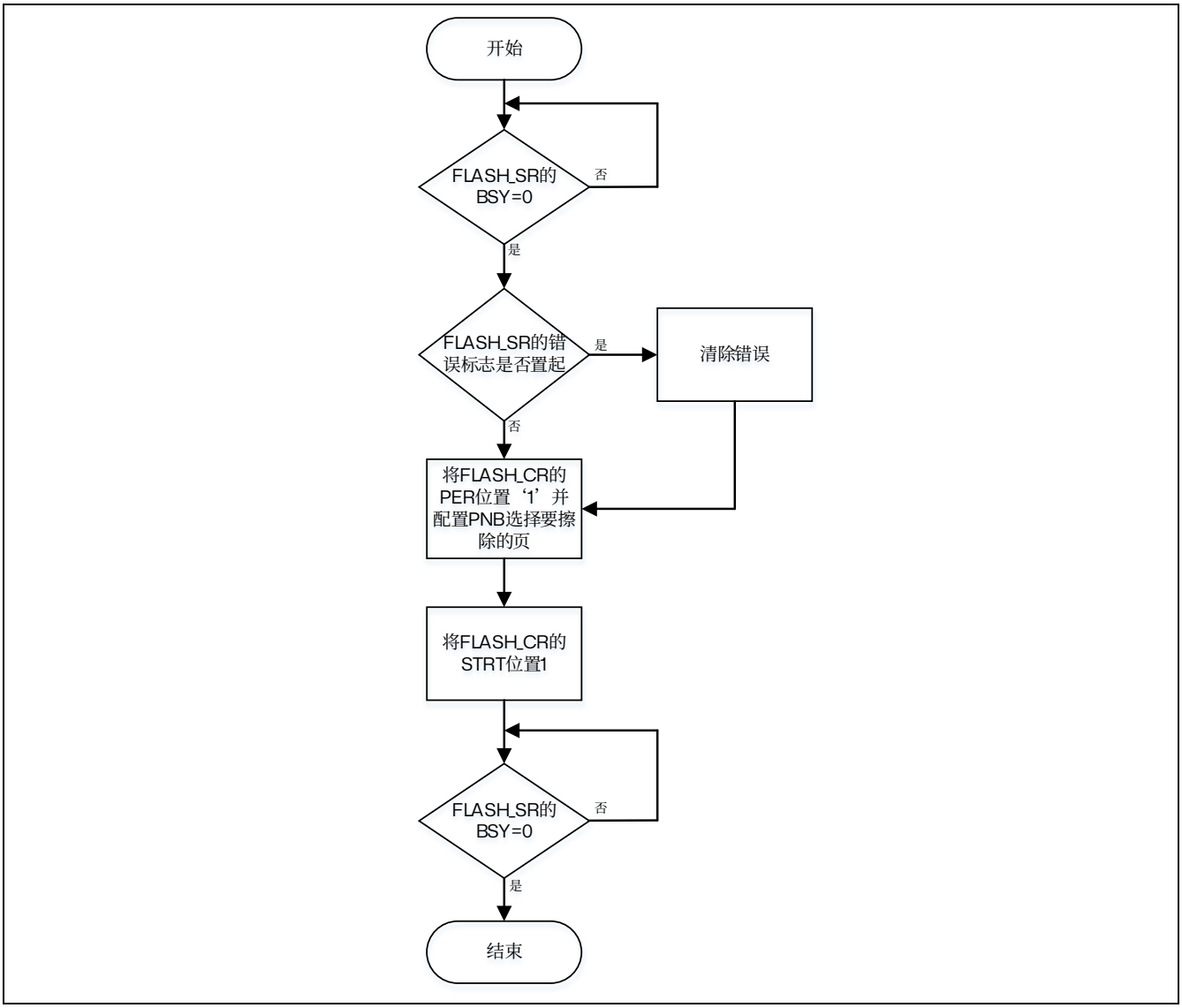


图 4.1 页擦流程

全擦

全擦流程如下：

1. 确认 FLASH_SR 寄存器的 BSY 位以保证没有正在执行的 Flash 操作。
2. 检查并清除之前的 Flash 操作造成的错误。
3. 将 FLASH_CR 的 MER 位置 1。
4. 将 FLASH_CR 的 STRT 位置 1。
5. 等 FLASH_SR 中 BSY 被清除。

Flash 超出 64K 不可擦除，也不可写入，不可读出，且不会进硬件错误中断。

注意：OTP 不可擦除。

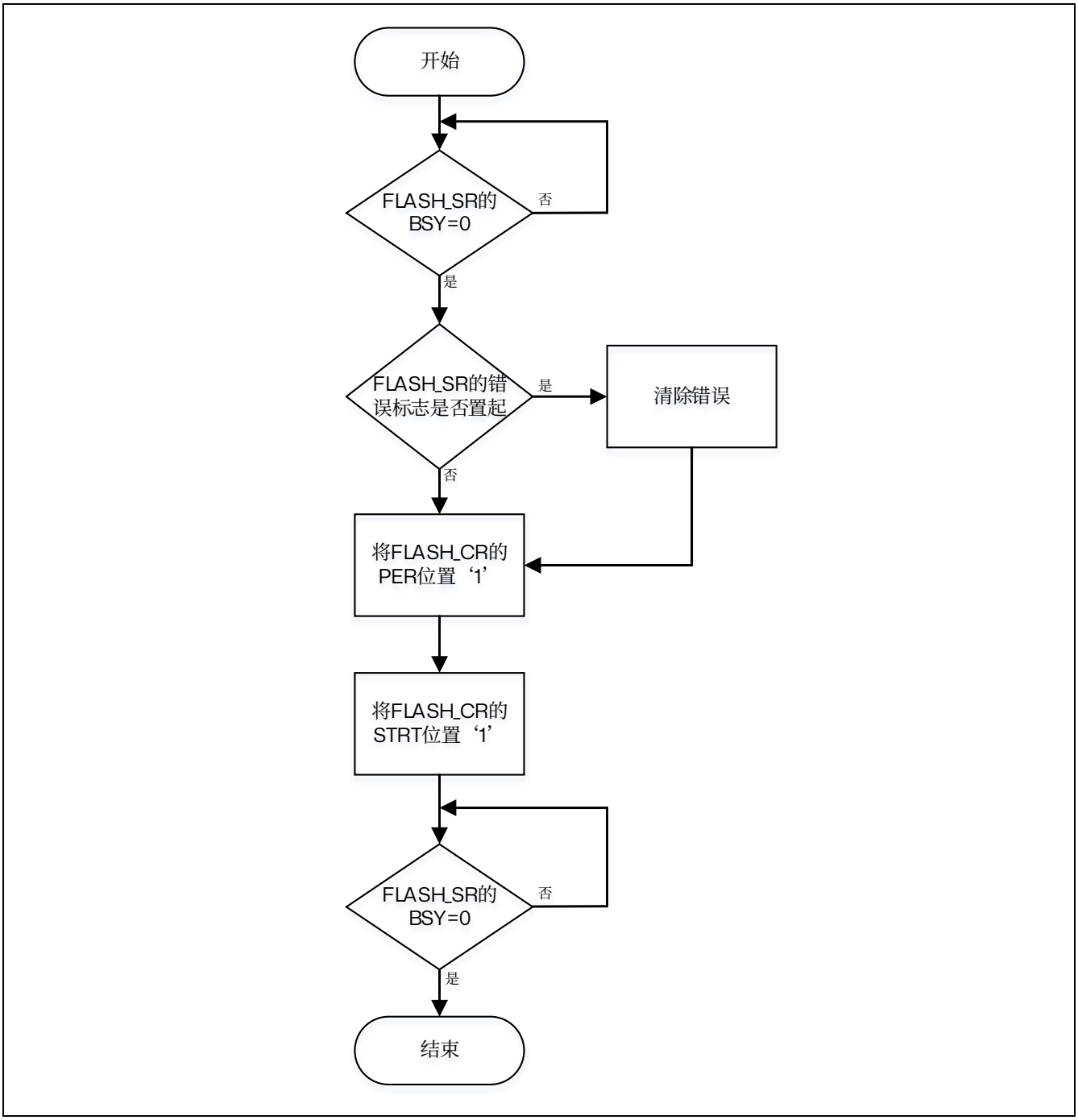


图 4.2 全擦流程

4.3.7 FLASH 写入流程

主 FLASH 一次写入一个 128 bit 的数据。写入流程如下：

1. 遮蔽所有中断并确认要写入的区域是否为全 F，若非全 F 需对要写入区域进行擦除操作。
2. 确认 FLASH_SR 的 BSY 位以保证没有正在执行的 Flash 操作。
3. 将 FLASH_CR 的 PG 位置 1。
4. 向目标地址写入数据（128bit）。
5. 等待 FLASH_SR 的 BSY 位清零。
6. 读取并验证写入的数据。

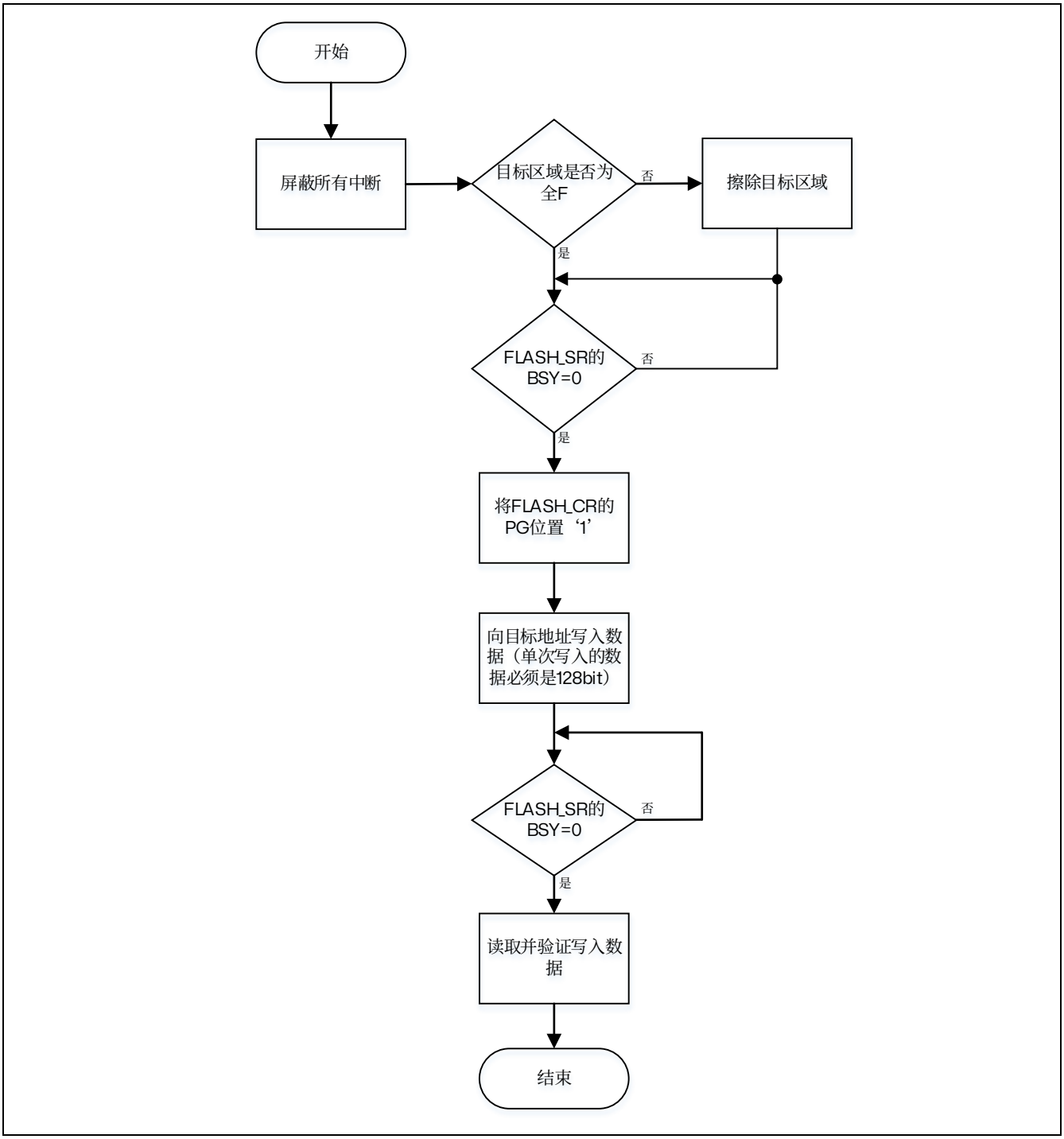


图 4.3 FLASH 写入流程

4.4 保护

闪存中的用户代码区可以防止非法的读出，同样可以对闪存区的页加以保护，防止在程序跑飞的情况下不被意外地改变，写保护的基本单位是 1Kb。

4.4.1 读保护

读保护是通过设置 RDP 选项字节，然后在系统重新复位加载了新的 RDP 选项字节后启动的。当保护字节被写入相应的值后，只允许 Flash 中的代码对 Flash 进行读操作。

解除读保护

1. 擦除整个选项字节区域，读保护码（RDP）将会变为 0xFF，此时读保护仍然有效。
2. 写入正确的 RDP 代码 0xAA 以解除 Flash 的保护，该操作会导致整块 Flash 被擦除。
3. 进行系统复位以重新加载选项字节，读保护被解除。

4.4.2 写保护

RX32S632 系列的写保护以 1Kb 为单位。如果在已经开启了写保护的区域进行写入或者擦除操作，FLASH_SR 中的 WRPERR 将会置起。

写保护是通过设置 FLASH_WRP1AR、FLASH_WRP1BR，然后在系统重新复位加载了新的 FLASH_WRP1AR、FLASH_WRP1BR 选项字节后启动的。

解除写保护

1. 擦除整个选项字节区域。
2. 写入正确的 FLASH_WRP1AR、FLASH_WRP1BR 代码。
3. 进行系统复位以重新加载选项字节，写保护被解除。

注意：全擦选项字节会将 RDP 擦除为 0xFF，系统重启后会启动读保护，因此需要将 RDP 进行修改。

4.5 选项字节

4.5.1 选项字节描述

选项字节一共有 48 字节，选项字节可以由终端用户根据应用情况进行配置。

选项字节中，一个 64 位的字被做如下划分。

表 4.3 选项字节格式

63-56	55-48	47-40	39-32	31-24	23-16	15-8	7-0
选项字节 3	选项字节 2	选项字节 1	选项字节 0	选项字节 3	选项字节 2	选项字节 1	选项字节 0

选项字节的内容可以通过直接访问选项字节所在地址获取，也可以通过访问选项字节寄存器（FLASH_OPTR）获取。

注意：选项字节的新配置在系统重启后才会被加载。

4.5.2 选项字节擦除

选项字节擦除由 FPEC 在选项字节写入过程中自动执行。

4.5.3 选项字节写入

选项字节写入流程如下：

1. 对 FPEC 解锁。
2. 对 FLASH_OPTKEYR 写入 KEY1 = 0x0819 2A3B。
3. 对 FLASH_OPTKEYR 写入 KEY2 = 0x4C5D 6E7F。
4. 检查 FLASH_SR 的 BSY 位，以确认没有其他正在进行的 Flash 操作。
5. 写选项值至 FLASH_OPTR/FLASH_WRP1AR/ FLASH_WRP1BR。
6. 将 FLASH_CR 寄存器的 OPTSTRT 位置 1，启动选项字节编程。
7. 等待 FLASH_SR 的 BSY 位清零。
8. 读取并验证写入的半字。

4.5.4 选项字节组织（Option Byte Organization）

Address	[63:56]	[55:48]	[47:40]	[39:32]	[31:24]	[23:16]	[15:8]	[7:0]
1FFF7800	用户选项			读保护	用户选项			读保护
1FFF7808	保留							
1FFF7810	保留							
1FFF7818	保留	写保护 A 区末端偏移	保留	写保护 A 区开始偏移	保留	写保护 A 区末端偏移	保留	写保护 A 区开始偏移
1FFF7820	保留	写保护 B 区末端偏移	保留	写保护 B 区开始偏移	保留	写保护 B 区末端偏移	保留	写保护 B 区开始偏移

4.5.5 用户和读保护选项字节

复位值：0x184F 30AA

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				nBOOT0	nSWBOOT0	Reserved		nBOOT1	Reserved				IWDG_STDBY	IWDG_STOP	IWDG_SW
				r	r			r					r	r	r

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

Bits 31:28 保留，必须保持为复位值。

Bit 27 **nBOOT0**：nBOOT0 选项位

0：nBOOT0=0。

1：nBOOT0=1。

Bit 26 **nSWBOOT0**：软件 BOOT0

0：BOOT0 由选项字节的 nBOOT0 配置。

1：FLASH Boot。

Bits 25:24 保留，必须保持为复位值。

Bit 23 **nBOOT1**：引导配置

0：nBOOT1=0。

1：nBOOT1=1。

Bits 22:19 保留，必须保持为复位值。

Bit 18 **IWDG_STDBY**：待机模式时独立看门狗计数冻结

0：进入待机模式时独立看门狗计数冻结。

1：进入待机模式时独立看门狗计数运行。

Bit 17 **IWDG_STOP**：停机模式时独立看门狗计数冻结

0：进入停机模式时独立看门狗计数冻结。

1：进入停机模式时独立看门狗计数运行。

Bit 16 **IWDG_SW**：独立看门狗选择

0：硬件独立看门狗。

1：软件独立看门狗。

Bits 15:14 保留，必须保持为复位值。

Bit 13 **nRST_STDBY**

0：进入待机模式时产生复位。

1：进入待机模式时不产生复位。

Bit 12 **nRST_STOP**

0：进入停止模式时产生复位。

1：进入停止模式时不产生复位。

Bits 11:8 保留，必须保持为复位值。

Bits 7:0 **RDP[7:0]**：读保护级

0xAA：级别为 0，读保护未激活。

其他：级别为 1，存储器读保护激活。

4.5.6 写保护 A 区地址选项字节

地址：0x1FFF 7818

复位值：0x0000 01FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							WRP1A_END[24:16]								
							rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							WRP1A_STRT[8:0]								
							rw								

Bits 31:24 保留，必须保持为复位值

Bits 23:16 **WRP1A_END[24:16]**：WRP 第一区域“A”区末端偏移(单位：1K 字节)

包含 WRP 第一个区域的最后一页。

Bits 15:8 保留，必须保持为复位值

Bits 7:0 **WRP1A_STRT[8:0]**：WRP 第一区域“A”区起始地址偏移(单位：1K 字节)

包含 WRP 第一个区域的第一个页面。

4.5.7 写保护 B 区地址选项字节

地址：0x1FFF 7820

复位值：0x0000 01FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							WRP1B_END[24:16]								
							rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							WRP1B_STRT[8:0]								
							rw								

Bits 31:24 保留，必须保持为复位值。

Bits 23:16 **WRP1B_END[24:16]**：WRP 第二区域“B”区末端偏移(单位：1K 字节)

包含 WRP 第二个区域的最后一页。

Bits 15:8 保留，必须保持为复位值。

Bits 7:0 **WRP1B_STRT[8:0]**：WRP 第二区域“B”区起始地址偏移(单位：1K 字节)

包含 WRP 的第二个区域的第一个页面。

4.6 Flash 寄存器

访问：无等待，支持字，半字和字节访问

4.6.1 Flash 访问控制寄存器（FLASH_ACR）

地址偏移：0x00

复位值：0x0000 0001

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

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

Bits 31:9 保留，必须保持为 0。

Bit 8 **PRFTEN**：预取缓冲区使能

0：关闭预取缓冲区

1：使能预取缓冲区

Bits 7:4 保留，必须保持为 0。

Bits 3:0 **LATENCY[3:0]**：等待周期

该位表示 SYSCLK（系统时钟）周期与 Flash 访问时间的比例。

000：零等待周期

001：一个等待周期

010：两个等待周期

011：三个等待周期

4.6.2 FPEC 键值寄存器（FLASH_KEYR）

地址偏移：0x08

复位值：0xXXXX XXXX

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

*注意：该寄存器是只写寄存器，读取为*0x0000 0000。

Bits 31:0 **KEYR[31:0]：**FPEC 键值

用于写入解锁 FPEC 的键值。

4.6.3 FLASH OPTKER 寄存器（FLASH_OPTKEYR）

地址偏移：0x0C

复位值：0xXXXX XXXX

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

*注意：该寄存器是只写寄存器，读取为*0x0000 0000。

Bits 31:0 **OPTKEYR[31:0]：**选项字节键值

用于写入解锁 OPTWRE 的键值。

4.6.4 FLASH 状态寄存器（FLASH_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
Reserved															BSY
															r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPTVERR	Reserved							PGSERR	SIZERR	PGAERR	WRPERR	Reserved		OPERR	EOP
rc_w1								rc_w1	rc_w1	rc_w1	rc_w1			rc_w1	rc_w1

Bits 31:17 保留，必须保持为 0。

Bit 16 **BSY：**忙碌

表示正在执行一个 Flash 操作。在 Flash 操作开始时该位由硬件置 1，当操作结束或者发生错误时该位由硬件清零。

Bit 15 **OPTVERR：**选项有效性错误

当选项字节没有被正确加载时，该位被硬件置位。

写 1 清零。

Bits 14:8 保留，必须保持为复位值。

Bit 7 **PGSERR：**编程顺序错误

写入闪存时发生顺序错误由硬件置位，或者当 SIZERR、PGAERR、WRPERR 由硬件

置位时该位由硬件置位。

写 1 清零。

Bit 6 **SIZERR**：字节大小错误

仅允许四字编程，否则该位由硬件置位。

写 1 清零。

Bit 5 **PGAERR**：写入对齐错误

待编程的数据不能被包含在同一个 128 位闪存行中，由硬件置位。

写 1 清零。

Bit 4 **WRPERR**：写保护错误

写入开启了写保护的 Flash 区域时由硬件置位。

写 1 清零。

Bits 3:2 保留，必须保持为 0。

Bit 1 **OPERR**：操作错误

Flash 区域操作（编程/擦除）未成功完成时，由硬件置位。

写 1 清零。

Bit 0 **EOP**：操作完成

一个或多个 Flash 操作（写入/擦除）成功完成时，由硬件置位。

写 1 清零。

注意：EOP 只有在 Flash 成功写入或者擦除结束时生效。

4.6.5 Flash 控制寄存器（FLASH_CR）

地址偏移：0x14

复位值：0xC000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	OPTLOCK	Reserved		OBL_LAUNCH	Res	ERRIE	EOPIE	Reserved						OPTSTRT	STRT
rs	rs			rcw1		rw	rw							rs	rs

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

Bit 31 **LOCK**：锁

软件写 1 有效。当该位置 1 时，FPEC 和 FLASH_CR 被锁住。检测到正确的解锁序列后，硬件清零此位。

在一次解锁不成功后，下次系统复位前，该位一直保持置位。

Bit 30 **OPTLOCK**：选项锁定

当该位置位时，所有关于用户选项的位寄存器和选项页面均被锁住。在检测到序列被解锁时，该位清零。

在一次解锁不成功后，下次系统复位前，该位一直保持置位。

Bits 29:28 保留，必须保持为复位值。

Bit 27 **OBL_LAUNCH**：强制选项字节加载

当该位置 1 时强制选项字节重加载。只有当字节选项加载完成时该位清零。如果 OPTLOCK 被置位，该位不可写。

0：选项字节加载完成

1：请求选项字节加载

Bit 26 保留，必须保持为复位值。

Bit 25 **ERRIE**：错误中断使能

该位允许在 FLASH_SR 中的 PGERR/WRPERR 被置位时产生中断。

0：关闭 ERR 中断

1：使能 ERR 中断

Bit 24 **EOPIE**：操作完成中断使能

该位允许在 FLASH_SR 中的 EOP 被置位时产生中断。

0：关闭 EOP 中断

1：使能 EOP 中断

Bits 23:18 保留，必须保持为复位值

Bit 17 **OPTSTRT**：选项修改开始

该位允许修改选项。该位仅由软件设置，在 BSY 置 1 时由硬件清零。

Bit 16 **STRT**：开始

软件写 1 后触发一次擦除操作。在 BSY 置 1 时由硬件清零。

Bits 15:9 保留，必须保持为复位值。

Bits 8:3 **PNB[5:0]：页码选择**

选择页面擦除。

000000：page 0

000001：page 1

…

111111：page 63

Bit 2 **MER**：全擦除

全擦除。

Bit 1 **PER**：页擦除

页擦除。

Bit 0 **PG**：写入 Flash

写入 Flash。

4.6.6 Flash 地址寄存器（FLASH_OPTR）

地址偏移：0x20

复位值：0x184F 30AA

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				n BOOT0	nSW BOOT0	Reserved		nBOOT1	Reserved				IWDG_ STDBY	IWDG_ STOP	IWDG_ SW
				rw	rw			rw					rw	rw	rw

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

Bits 31:28 保留，必须保持为复位值。

Bit 27 **nBOOT0**：nBOOT0 选项位

0：nBOOT0=0。

1：nBOOT0=1。

Bit 26 **nSWBOOT0**：软件 BOOT0

0：BOOT0 由选项字节的 nBOOT0 配置。

1：FLASH Boot。

Bits 25:24 保留，必须保持为复位值。

Bit 23 **nBOOT1：**引导配置

0：nBOOT1=0。

1：nBOOT1=1。

Bits 22:19 保留，必须保持为复位值。

Bit 18 **IWDG_STDBY：**待机模式时独立看门狗计数冻结

0：进入待机模式时独立看门狗计数冻结。

1：进入待机模式时独立看门狗计数运行。

Bit 17 **IWDG_STOP：**停机模式时独立看门狗计数冻结

0：进入停机模式时独立看门狗计数冻结。

1：进入停机模式时独立看门狗计数运行。

Bit 16 **IWDG_SW：**独立看门狗选择

0：硬件独立看门狗。

1：软件独立看门狗。

Bits 15:14 保留，必须保持为复位值。

Bit 13 **nRST_STDBY**

0：进入待机模式时产生复位。

1：进入待机模式时不产生复位。

Bit 12 **nRST_STOP**

0：进入停止模式时产生复位。

1：进入停止模式时不产生复位。

Bits 11:8 保留，必须保持为复位值。

Bits 7:0 **RDP[7:0]：读保护级**

0xAA：级别为 0，读保护未激活。

其他：级别为 1，存储器读保护激活。

4.6.1 写保护 A 区寄存器（FLASH_WRP1AR）

地址偏移：0x2C

复位值：0x0000 01FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							WRP1A_END[24:16]								
							rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							WRP1A_STRT[8:0]								
							rw								

Bits 31:25 保留，必须保持为复位值。

Bits 24:16 **WRP1A_END[24:16]：**WRP 第一区域“A”区末端偏移(单位：1K 字节)

包含 WRP 第一个区域的最后一个页面。

Bits 15:9 保留，必须保持为复位值。

Bits 8:0 **WRP1A_STRT[8:0]：**WRP 第一区域“A”区起始地址偏移(单位：1K 字节)

包含 WRP 第一个区域的第一个页面。

4.6.2 写保护 B 区寄存器（FLASH_WRP2BR）

地址偏移：0x30

复位值：0x0000 01FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							WRP1B_END[24:16]								
							rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							WRP1B_STRT[8:0]								
							rw								

Bits 31:25 保留，必须保持为复位值。

Bits 24:16 **WRP1B_END[24:16]：**WRP 第二区域“B”区末端偏移(单位：1K 字节)

包含 WRP 第二个区域的最后一个页面。

Bits 15:9 保留，必须保持为复位值。

Bits 8:0 **WRP1B_STRT[8:0]：**WRP 第二区域“B”区起始地址偏移(单位：1K 字节)

包含 WRP 第二个区域的第一个页面。

5 CRC 计算单元

5.1 CRC 简介

CRC（循环冗余校验）计算单元使用一个固定的多项式发生器从一个 32 位的数据字中产生 CRC 码。在众多的应用中，基于 CRC 的技术还常用来验证数据传输或存储的完整性。根据 EN/IEC 60335-1 标准的规定，这些技术提供了验证 Flash 完整性的方法。CRC 计算单元有助于在运行期间计算软件的签名，并将该签名与链接时生成并存储在指定存储单元的参考签名加以比较。

5.2 CRC 主要特性

- 使用 CRC-32（以太网）多项式：0x4C11DB7
  - $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- 单输入/输出 32 位数据寄存器
- CRC 计算在 4 个 AHB 时钟周期（HCLK）内完成
- 8 位通用寄存器（可用于临时存储）

下图为 CRC 计算单元框图

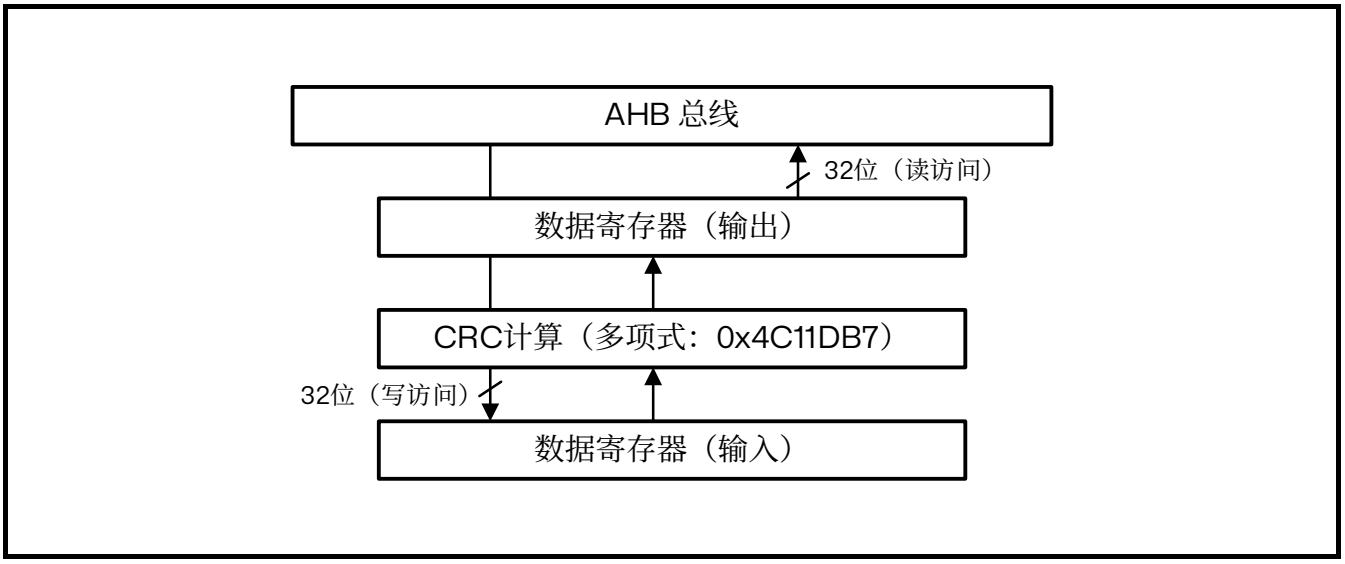


图 5.1 CRC 计算单元框图

5.3 CRC 功能说明

CRC 计算单元主要由单个 32 位数据寄存器组成，该寄存器：

- 用作输入寄存器，向 CRC 计算器中输入新数据（向寄存器写入数据时）
- 可保存之前的 CRC 计算结果（读取寄存器时）

对数据寄存器的每个写操作都会把当前新输入的数值和之前生成在数据寄存器中的 CRC 值再做一次 CRC 计算（CRC 计算针对整个 32 位数据字完成，而非逐字节进行）。

CRC 计算的时候，写操作被阻塞，因此允许执行背靠背写访问或连续的写读访问。

使用 CRC_CR 寄存器中的 RESET 控制位即可将 CRC 计算器复位为 0xFFFF FFFF。此操作不影响 CRC_IDR 寄存器的内容。

5.4 CRC 寄存器

CRC 计算单元包含两个数据寄存器和一个控制寄存器。CRC 寄存器必须按字（32 位）访问。

5.4.1 数据寄存器（CRC_DR）

偏移地址：0x00

复位值：0xFFFF FFFF

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

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

Bits 31:0 **DR[31:16]**：数据寄存器位

向 CRC 计算器写入新数据时用作输入寄存器。

读取寄存器时可读出之前的 CRC 计算结果。

5.4.2 独立数据寄存器（CRC_IDR）

地址偏移：0x04

复位值：0x0000 0000

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

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

Bits 31:8 保留，必须保持复位值。

Bits 7:0 **IDR[7:0]：**通用 8 位数据寄存器位

可用作一个字节的临时存储单元。

此寄存器不受 CRC_CR 寄存器中 RESET 位产生的 CRC 复位影响。

5.4.3 控制寄存器（CRC_CR）

地址偏移：0x08

复位值：0x0000 0000

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

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

Bits 31:1 保留，必须保持复位值。

Bit 0 **RESET：**

复位 CRC 计算单元并将数据寄存器设为 0xFFFF FFFF。

此位只能置 1，将由硬件自动进行清零。

6 电源控制（PWR）

6.1 电源

RX32S632 的 HVDD 电压范围为：5~28V，VDD 电压范围为：2.7~3.6V。

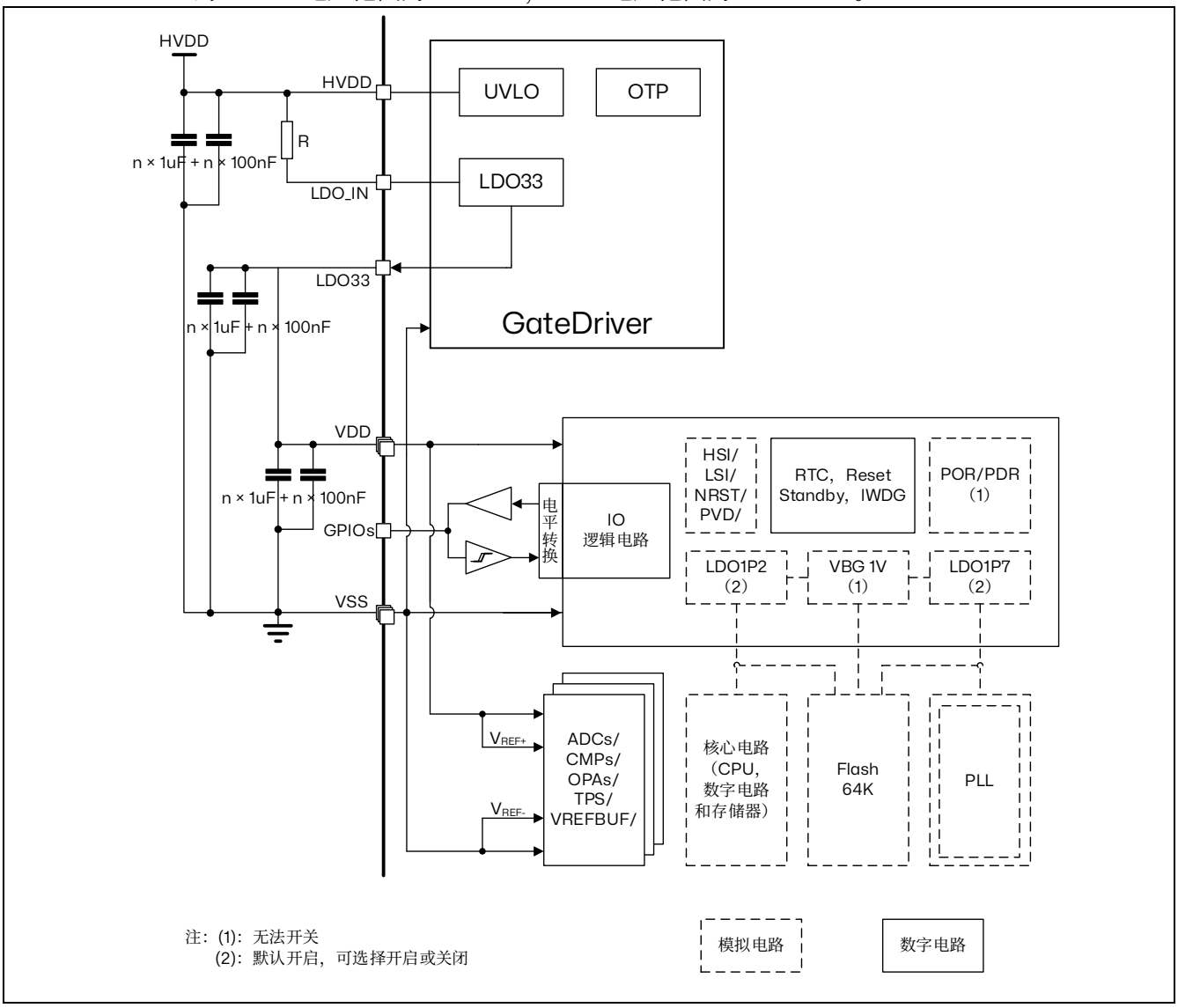


图 6.1 电源框图

6.2 电源管理器

6.2.1 上电复位（POR）和掉电复位（PDR）

RX32S632 内部有一个完整的上电复位（POR）和掉电复位（PDR）电路，当供电电压达到 2.7V 时，系统既能正常工作。

当 VDD/VDDA 低于指定的限位电压 V_{POR}/V_{PDR} 时，系统保持为复位状态，无需外部复位电路。关于上电复位和掉电复位的细节请参考数据手册的电气特性部分。

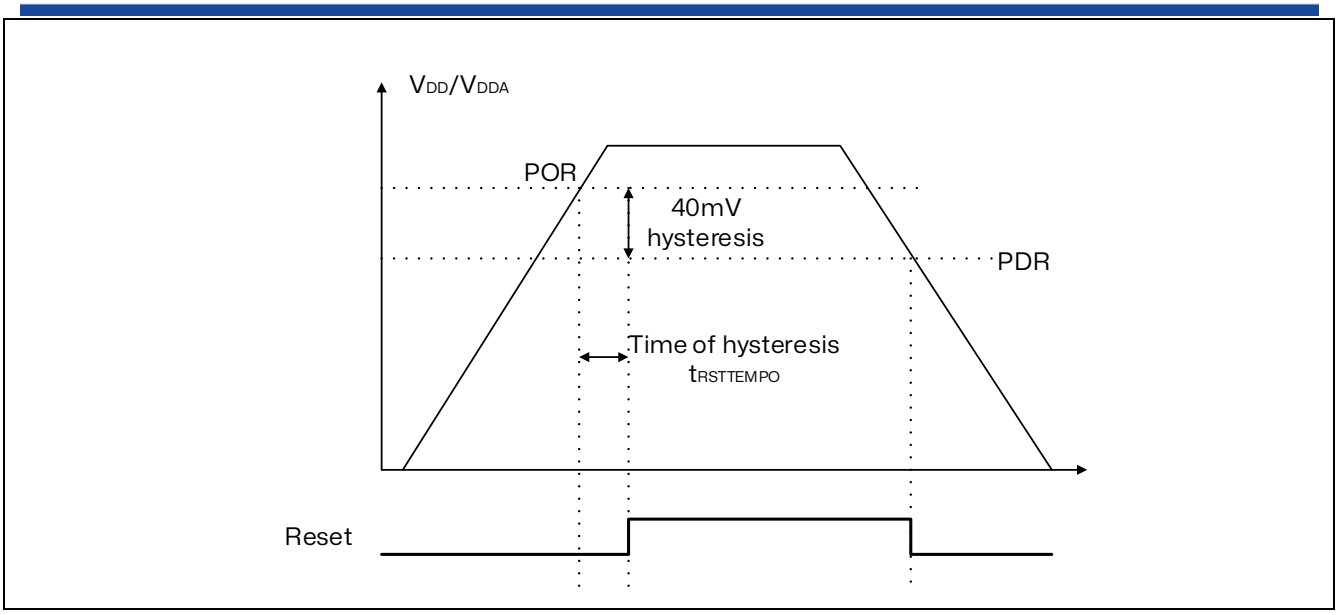


图 6.2 上电复位和掉电复位波形图

6.2.2 可编程电压检测器（PVD）

用户可以利用 PVD 对 VDD 电压与电源控制寄存器 2（PWR_CR2）中的 PLS[2:0]位进行比较来监控电源，这几位选择监控电压的阈值。通过设置 PVDE 位来使能 PVD。

电源状态寄存器 2（PWR_SR2）中的 PVDO 标志用来表明 VDD 是高于还是低于 PVD 的电压阈值。该事件在内部连接到外部中断的线 16，如果该中断在外部中断寄存器中是使能的，该事件就会产生中断。当 VDD 下降到 PVD 阈值以下和（或）当 VDD 上升到 PVD 阈值之上时，根据外部中断第 16 线的上升/下降边沿触发设置，就会产生 PVD 中断。

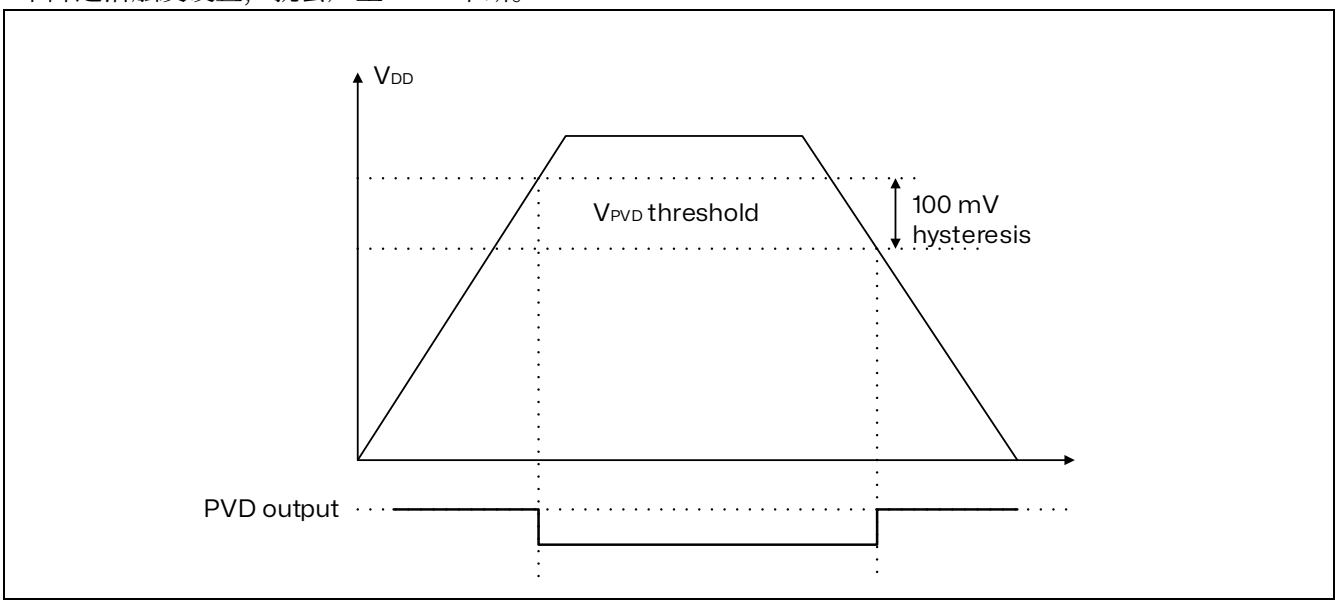


图 6.3 PVD 门限

6.3 低功耗模式

默认情况下，在系统或电源复位以后，微控制器处于运行状态。当 CPU 不需继续运行时（例如等待外部事件时），可以利用多种低功耗模式来节省功耗。用户需要根据最低电源消耗、最快速启动时间和可用的唤醒源等条件，选定一个最佳的低功耗模式。

RX32S632 支持三种低功耗模式，用户可以在以下模式中进行选择：

- 睡眠模式：CPU 时钟被关闭。所有外设继续运行，并且 CPU 可以被中断或者事件唤醒。
- 停止 0 和停止 1 模式：保留 SRAM 和寄存器内容。关闭除 LSI 以外所有时钟。该系列可以被任一外部中断线从停止模式中唤醒。
  在停止 0 模式下，内部 LDO1P7 和 LDO1P2 配置为正常模式，这允许最快的唤醒时间但功耗更高。活动的外设和唤醒源与停止 1 模式相同。
- 待机模式：待机模式用于实现最低的功耗，关闭 LDO1P7 和 LDO1P2，关闭除 LSI 以外所有时钟。该系列可以被外部复位（NRST 引脚）、IWDG 复位、RTC 唤醒以及 WKUP 唤醒引脚的上升沿和下降沿从待机模式中唤醒。

注：配置上升沿唤醒时，WKUP 引脚不要外部上拉。配置下降沿唤醒时，WKUP 引脚不要外部下拉。

此外，可以通过以下方式降低运行模式的功耗：

- 降低系统时钟速度
- 当 APB 和 AHB 外设未使用时对它们进行时钟门控。

表 6.1 低功耗模式一览

模式	唤醒	唤醒后时钟	对时钟影响	LDO	
				1P7	1P2
睡眠 (立即睡眠或退出时睡眠)	任一中断 唤醒事件	与进入睡眠模式前一致	CPU 时钟关闭	正常	正常
停止 0	任意外部中断（在 EXTI 寄存器中配置） 指定外设事件	HSI	关闭除 LSI 以外所有时钟	正常	正常
停止 1				低功耗	低功耗
待机	WKUP 引脚电平变化、RTC 唤醒事件、NRST 复位、IWDG 复位	HSI	关闭除 LSI 以外所有时钟	关闭	关闭

调试模式

默认情况下，如果在进行调试微处理器时，使微处理器进入停止 0、停止 1 或待机模式，将失去调试连接。这是因为内核失去了时钟。

通过设置 DBGMCU_CR 寄存器中的对应配置位，可以在低功耗模式下调试软件。有关更多详细信息，请参阅章节：调试支持（DBG）。

6.3.1 运行模式

降低系统时钟速度

在运行模式下，可通过对预分频寄存器编程来降低系统时钟（SYSCLK、HCLK、PCLK）速度。进入睡眠模式之前，也可以使用这些预分频器降低外设速度。

更多详细信息，请参阅第 7.3.2 节：时钟配置寄存器（RCC_CFGR）。

外设时钟门控

在运行模式下，可随时停止各外设和存储器的 HCLK 和 PCLK 以降低功耗。

要进一步降低睡眠模式的功耗，可在进入睡眠模式前禁止外设时钟。

外设时钟门控由 RCC_AHBENR 和 RCC_APBxENR 寄存器控制。

6.3.2 睡眠模式

睡眠模式下的 I/O 状态

睡眠模式下，所有 I/O 保持在进入睡眠模式前的状态。

6.3.3 停止 0 模式

停止模式基于 32 位 RISC 内核深度睡眠模式基础上结合了外设的时钟控制机制。该模式下，内部 LDO 配置为正常模式。关闭除 LSI 以外所有时钟。内部 SRAM 和寄存器内容都将保留。

注：停止 1 模式与停止 0 模式相同，但内部 LDO 进入低功耗模式。

6.3.4 待机模式

待机模式可以实现 MCU 的最低功耗。该模式是在 32 位 RISC 内核深度睡眠模式时关闭内部线性稳压器 LDO1P7 和 LDO1P2。PLL、HSI 和 HSE 也被关闭。

SRAM 和寄存器的内容会丢失，除了待机电路中的寄存器。

待机模式下的 I/O 状态

在待机模式下，I/O 可以保持在模拟状态。1 个唤醒引脚（WKUP，PD4）也是可用的。

6.4 低功耗模式下的自动唤醒（AWU）

RTC 可以在不需要依赖外部中断的情况下唤醒低功耗模式下的微控制器（自动唤醒模式）。RTC 提供一个可编程的时间基数，用于周期性从停止（0 或 1）或待机模式下唤醒。RTC 时钟源选择 LSI 可以实现此功能。

为了用 RTC 闹钟事件将系统从停止模式下唤醒，必须进行如下操作：

- 配置外部中断线 17 为上升沿触发。
- 配置 RTC 使其可产生 RTC 闹钟事件。

如果要从待机模式中唤醒，不必配置外部中断线 17。

6.5 电源控制寄存器

可以用半字（16 位）或字（32 位）的方式操作这些外设寄存器。

6.5.1 电源控制寄存器 1（PWR_CR1）

地址偏移：0x00
复位值：0x0000 0000
从待机模式唤醒后，此寄存器将被重置。

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

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

Bits 31:9 保留，必须保持复位值。

Bit 8 **DBP**：取消 RTC 的写保护

在复位后，RTC 处于被保护状态以防意外写入。

设置这位允许写入这些寄存器。

0：禁止写入 RTC

1：允许写入 RTC

Bits 7:2 保留，必须保持复位值。

Bits 1:0 **LPMS[1:0]**：低功耗模式选择

00：停止 0 模式

01：停止 1 模式

10：保留

11：待机模式

Bit 0 保留：必须保持为 0。

6.5.2 电源控制寄存器 2（PWR_CR2）

地址偏移：0x04
复位值：0x0000 0000
从待机模式唤醒后，此寄存器将被重置。

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

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

Bits 31:4 保留，必须保持复位值。

Bits 3:1 **PLS[2:0]**：PVD 电平选择

这些位用于选择电源电压监测器的电压阈值

000：2.2V　　100：2.6V

001：2.3V　　101：2.7V

010：2.4V　　110：2.8V

011：2.5V　　111：2.9V

Bit 0 **PVDE**：电源电压监测器(PVD)使能

0：禁止 PVD
1：开启 PVD

6.5.3 电源控制寄存器 3（PWR_CR3）

地址偏移：0x08
复位值：0x0000 0000
退出待机模式以及 RCC_APB1RSTR 寄存器中的 PWRRST 位被设置时，该寄存器不会被重置。
访问：访问此寄存器需要比标准 APB 访问更多的 APB 周期（写操作需要 3 个周期，读操作需要 2 个周期）。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EIWUL	Reserved														EWUP1
rw															rw

Bits 31:16 保留，必须保持复位值。

Bit 15 **EIWUL**：使能内部唤醒线（RTC/IWDG）

0：禁止内部唤醒线唤醒
1：使能内部唤醒线

Bits 14:1 保留，必须保持复位值。

Bit 0 **EWUP1**：使能 WKUP 引脚

0：WKUP 引脚为通用 I/O。WKUP 引脚 1 上的事件不能将 CPU 从待机模式唤醒
1：WKUP 引脚用于将 CPU 从待机模式唤醒，WKUP 引脚 1 上的上升沿/下降沿将系统从待机模式唤醒，有效边沿通过 PWR_CR4 的 WP1 设置。

6.5.4 电源控制寄存器 4（PWR_CR4）

地址偏移：0x0C
复位值：0x0000 0000
退出待机模式以及 RCC_APB1RSTR 寄存器中的 PWRRST 位被设置时，该寄存器不会被重置。
访问：访问此寄存器需要比标准 APB 访问更多的 APB 周期（写操作需要 3 个周期，读操作需要 2 个周期）。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															WP1
															rw

Bits 31:1 保留，必须保持复位值。

Bit 0 **WP1**：WKUP1 引脚唤醒极性

0：检测到高电平(上升沿)
1：检测到低电平(下降沿)

6.5.5 电源状态寄存器 1（PWR_SR1）

地址偏移：0x10

复位值：0x0000 0000

退出待机模式以及 RCC_APB1RSTR 寄存器中的 PWRRST 位被设置时，该寄存器不会被重置。

访问：与此标准 APB 读取相比，读取此寄存器需要额外的 2 个 APB 周期。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WUFI	Reserved						SBF	Reserved							WUF1
r							r								r

Bits 31:16 保留，必须保持复位值。

Bit 15 **WUFI**：内部唤醒标志(IWDG/RTC 唤醒)

当在内部唤醒线上检查到唤醒时，设置此位。当所有内部唤醒源被清除时该位被清除。

Bits 14:9 保留，必须保持复位值。

Bit 8 **SBF**：待机标志

该位由硬件设置，并只能由 POR/PDR(上电/掉电复位)或设置电源控制寄存器(PWR_CR)的 CSBF 位清除。

0：系统不在待机模式

1：系统进入待机模式

Bits 7:1 保留，必须保持复位值。

Bit 0 **WUF1**：WKUP1 引脚唤醒标志

当在唤醒引脚 WKUP1 上检测到唤醒事件时，设置此位。它被清除通过在 PWR_SCR 寄存器的 CWUF1 位写入“1”。

6.5.6 电源状态寄存器 2（PWR_SR2）

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
Reserved															

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

Bits 31:12 保留，必须保持复位值。

Bit 11 **PVDO**：PVD 输出

当 PVD 被 PVDE 位使能后该位才有效

0：VDD/VDDA 高于由 PLS[2:0]选定的 PVD 阈值

1：VDD/VDDA 低于由 PLS[2:0]选定的 PVD 阈值

*注：在待机模式下*PVD *被停止。因此，待机模式后或复位后，直到设置*PVDE *位之前，该位为*0。

Bits 10:0 保留，必须保持复位值。

6.5.7 电源状态控制寄存器（PWR_SCR）

地址偏移：0x18

复位值：0x0000 0000

访问：与此标准 APB 写入相比，写入此寄存器需要额外的 3 个 APB 周期。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CWUFI	Reserved						CSBF	Reserved							CWUF1
w							w								w

Bits 31:16 保留，必须保持复位值。

Bit 15 **CWUFI**：清除内部唤醒标志

写 1 清除内部唤醒标志 PWR_SR1 的 WUFI 位

Bits 14:9 保留，必须保持复位值。

Bit 8 **CSBF**：清除待机标志

写 1 清除待机标志 PWR_SR1 的 SBF 位

Bits 7:1 保留，必须保持复位值。

Bit 0 **CWUF1**：清除唤醒标志 1

写 1 清除 WKUP1 引脚唤醒标志 PWR_SR1 的 WUF1 位

7 复位和时钟控制（RCC）

7.1 复位

RX32S632 系列支持两种复位，分别是系统复位和上电复位。

7.1.1 系统复位

除了时钟控制器的 RCC_CSR 寄存器中的复位标志位以外，系统复位将复位所有寄存器至它们的复位状态。

当发生以下任一事件时，产生一个系统复位：

1. NRST 引脚上的低电平（外部复位）
2. 独立看门狗计数结束（IWDG 复位）
3. 软件复位（SW 复位）
4. 低功耗管理复位

可通过查看 RCC_CSR 控制状态寄存器中的复位状态标志位识别复位事件来源。

低功耗管理复位

在以下两种情况下可产生低功耗管理复位：

1. 在进入待机模式时产生低功耗管理复位：
   通过将用户选择字节中的 nRST_STDBY 位清零将使能该复位。这时，即使执行了进入待机模式的过程，系统将被复位而不是进入待机模式。
2. 在进入停止模式时产生低功耗管理复位：
   通过将用户选择字节中的 nRST_STOP 位清零将使能该复位。这时，即使执行了进入停止模式的过程，系统将被复位而不是进入停机模式。

7.1.2 电源复位

当以下事件中之一发生时，产生电源复位：

1. 上电/掉电复位（POR/PDR 复位）
2. 从待机模式中返回

电源复位将复位所有寄存器。

下图中复位源将最终作用于 NRST 引脚，并在复位过程中保持低电平。复位入口矢量被固定在地址 0x0000_0004。

芯片内部的复位信号会在 NRST 引脚上输出，脉冲发生器保证每一个（外部或内部）复位源都能有至少 20μs 的脉冲延时；当 NRST 引脚被拉低产生外部复位时，它将产生复位脉冲。

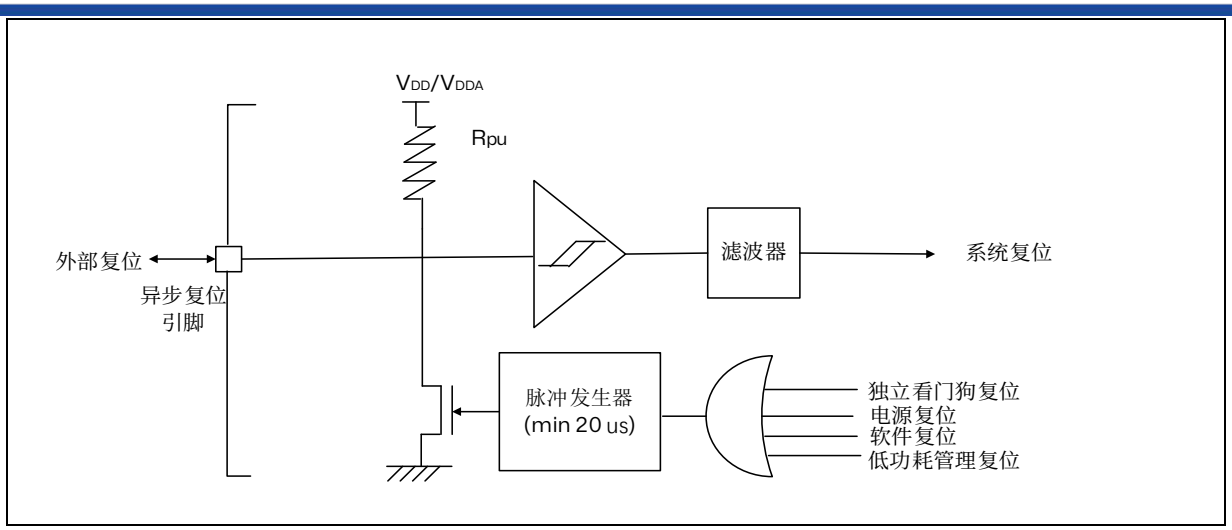


图 7.1 复位电路框图

7.2 时钟

3 种不同的时钟源可被用来驱动系统时钟（SYSCLK）：

- HSI
- LSI
- PLL

用户可通过多个预分频器配置 AHB、APB1 和 APB2 域的频率，AHB 最大允许频率是 152MHz，APB1 最大允许频率是 80MHz。

可选择 AHB 时钟（HCLK）8 分频后作为内核系统定时器时钟。ADC 时钟由 HCLK 或 PLL2CLK 时钟经 2、4、6 或 8 分频后获得，最大允许频率是 60.8MHz。

FCLK 是内核的自由运行时钟。

定时器时钟频率分配由硬件按以下情况自动设置：

如果相应的 APB 预分频系数是 1，定时器的时钟频率与所在 APB 总线频率一致。

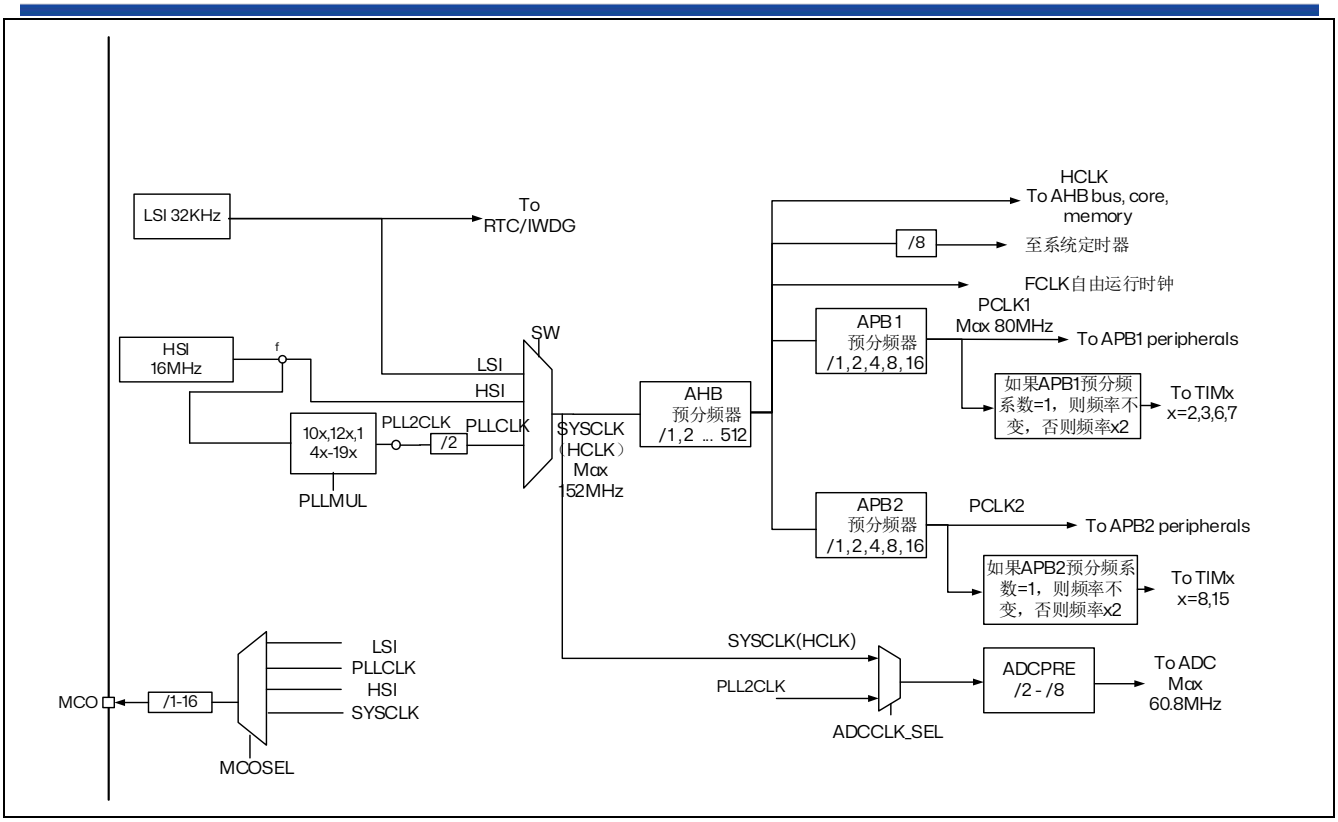


图 7.2 时钟树

7.2.1 HSI 时钟

HSI 时钟信号由 16MHz 的 RC 振荡器产生，可直接作为系统时钟或经过 PLL 倍频后 2 分频后作为 PLL 输入。

HSI RC 振荡器能够在不需要任何外部器件的条件下提供系统时钟。

校准

制造工艺决定了不同芯片的 RC 振荡器频率会不同，这就是为什么每个芯片的 HSI 时钟频率在出厂前已经被校准到 1.5%（25°C，具体参数参考相应的数据手册）的原因。系统复位时，工厂校准值被装载到时钟控制寄存器的 HSICAL[7:0]位。

如果用户的应用基于不同的电压或环境温度，这将会影响 RC 振荡器的精度。

时钟控制寄存器中的 HSIRDY 位用来指示 HSI RC 振荡器是否稳定。在时钟启动过程中，直到这一位被硬件置‘1’，HSI RC 输出时钟才被释放。HSI RC 可由时钟控制寄存器中的 HSION 位来启动和关闭。

7.2.2 PLL

内部 PLL 可以用来倍频 HSI 或者外部有源晶振的输出时钟。PLL 倍率范围为 10x，12x，14x-19x，PLL 输出 PLL2CLK，经过数字分频 2 后输出 PLLCLK。

如果 PLL 中断在时钟中断寄存器里被允许，当 PLL 准备就绪时，可产生中断申请。

7.2.3 LSI 时钟

LSI 担当一个低功耗时钟源的角色，它可以在停机和待机模式下保持运行，为独立看门狗和自动唤醒单元提供时钟。LSI 时钟频率大约 32KHz（在 30KHz 和 60KHz 之间）。进一步信息请参考数据手册中有关电气特性部分。

LSI 可以通过控制/状态寄存器（RCC_CSR）里的 LSION 位来启动或关闭。

在控制/状态寄存器（RCC_CSR）里的 LSIRDY 位指示低速内部振荡器是否稳定。在启动阶段，直到这个位被硬件设置为‘1’后，此时钟才被释放。如果在时钟中断寄存器（RCC_CIR）里被允许，将产生 LSI 中断申请。

7.2.4 系统时钟（SYSCLK）选择

系统复位后，HSI 振荡器被选为系统时钟。当时钟源被直接或通过 PLL 间接作为系统时钟时，不能被停止。

只有当目标时钟源准备就绪了（经过启动稳定阶段的延迟或 PLL 稳定），从一个时钟源到另一个时钟源的切换才会发生。在被选择时钟源没有就绪时，系统时钟的切换不会发生。直至目标时钟源就绪，才发生切换。

在时钟控制寄存器（RCC_CR）里的状态位指示哪个时钟已经准备好了，哪个时钟目前被用作系统时钟。

7.2.5 RTC 时钟

RTC 时钟固定为 LSI。

7.2.6 看门狗时钟

如果独立看门狗已经由硬件选项或软件启动，LSI 振荡器将被强制在打开状态，并且不能被关闭。在 LSI 振荡器稳定后，时钟供应给 IWDG。

7.2.7 时钟输出

微控制器允许输出时钟信号到外部 MCO 引脚。

相应的 GPIO 端口寄存器必须被配置为相应功能。以下四个时钟信号可被选作 MCO 时钟：

- SYSCLK
- HSI
- PLLCLK
- LSI

时钟的选择由时钟配置寄存器（RCC_CFGR）中的 MCO[2:0]位控制。

7.3 RCC 寄存器

7.3.1 时钟控制寄存器（RCC_CR）

地址偏移：0x00

复位值：0x0010 XX83

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						PLLRDY	PLLON	Reserved							
						r	rw								

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

Bits 31:26 保留，必须保持为复位值。

Bit 25 **PLLRDY**：PLL 时钟就绪标志

PLL 锁定后由硬件置“1”。

0：PLL 未锁定

1：PLL 已锁定

Bit 24 **PLLON**：PLL 使能

当进入待机和停止模式时，该位由硬件清零。当 PLL 时钟被用作或被选择将要作为系统时钟时，该位不能被清零。由软件置“1”或清零。

0：PLL 关闭

1：PLL 使能

Bits 23:16 保留，必须保持为复位值。

Bits 15:8 **HSICAL[7:0]**：内部高速时钟校准

在系统启动时，这些位被自动初始化。

Bits 7:2 保留，必须保持为复位值。

Bit 1 **HSIRDY**：内部高速时钟就绪标志

由硬件置“1”来指示内部 16MHz 振荡器已经稳定。在 HSION 位清零后，该位需要 6 个内部 16MHz 振荡器周期清零。

0：内部 16MHz 振荡器未就绪

1：内部 16MHz 振荡器就绪

Bit 0 **HSION**：内部高速时钟使能

用来启动内部 16MHz 的 RC 振荡器。当内部 16MHz 振荡器被直接或间接地用作或被选择将要作为系统时钟时，该位不能被清零。由软件置“1”或清零。

0：禁止 HSI

1：使能 HSI

7.3.2 时钟配置寄存器（RCC_CFGR）

地址偏移：0x04

复位值：0x8000 C000

访问：0 到 2 个等待周期，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADCPRE[2]	Res	ADCCLKSEL	MCOPRE[1:0]		MCO[2:0]			Reserved			PLLMUL[2:0]			Reserved	
rw		rw	rw	rw	rw	rw	rw				rw	rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCPRE[1:0]		PPRE2[2:0]			PPRE1[2:0]			HPRE[3:0]				SWS[1:0]		SW[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r	rw	rw

Bit 31 **ADCPRE[2]：**ADC 预分频

Bit 30 保留，必须保持为复位值。

Bit 29 **ADCCLK_SEL：**ADCCLK 时钟源选择位

由软件置“1”或清零来选择 ADC 时钟源。

0：SYSCLK 时钟分频后作为 ADC 时钟

1：PLL2CLK 时钟作为 ADC 时钟

Bits 28:27 **MCOPRE[1:0]：**

由软件置“1”或清零。

在 MCO 输出使能之前修改分频系数。

00：MCO 1 分频

01：MCO 2 分频

10：MCO 4 分频

11：MCO 8 分频

Bits 26:24 **MCOSEL[2:0]：**微控制器时钟输出

000：无时钟输出

001：SYSCLK

010：保留

011：HSI

101：PLLCLK

110：LSI

注意：该时钟输出在启动和切换 MCO *时钟源时可能会被截断。在系统时钟作为输出至* MCO *引脚时，请保证输出时钟频率不超过* 50 MHz。

Bits 23:21 保留，必须保持为复位值。

Bits 20:18 **PLLMUL[2:0]：**PLL 倍频系数

由软件设置来确定 PLL 倍频系数。只有在 PLL 关闭的情况下才可被写入。

000：10 倍频输出

001：12 倍频输出

010：14 倍频输出

011：15 倍频输出

100：16 倍频输出

101：17 倍频输出

110：18 倍频输出

111：19 倍频输出

Bit 17 保留，必须保持为复位值。

Bits 31,15:14 **ADCPRE[2:0]**：ADC 预分频

000：无

001：ADCCLK 2 分频后作为 ADC 时钟

010：ADCCLK 3 分频后作为 ADC 时钟

011：ADCCLK 4 分频后作为 ADC 时钟

100：ADCCLK 5 分频后作为 ADC 时钟

101：ADCCLK 6 分频后作为 ADC 时钟

110：ADCCLK 7 分频后作为 ADC 时钟

111：ADCCLK 8 分频后作为 ADC 时钟

Bits 13:11 **PPRE2[2:0]**：高速 APB（APB2）预分频

0xx：HCLK 不分频

100：HCLK 2 分频

101：HCLK 4 分频

110：HCLK 8 分频

111：HCLK 16 分频

Bits 10:8 **PPRE1[2:0]**：低速 APB（APB1）预分频

0xx：HCLK 不分频

100：HCLK 2 分频

101：HCLK 4 分频

110：HCLK 8 分频

111：HCLK 16 分频

Bits 7:4 **HPRE[3:0]**：AHB 预分频

0xxx：SYSCLK 不分频

1000：SYSCLK 2 分频

1001：SYSCLK 4 分频

1010：SYSCLK 8 分频

1011：SYSCLK 16 分频

1100：SYSCLK 64 分频

1101：SYSCLK 128 分频

1110：SYSCLK 256 分频

1111：SYSCLK 512 分频

注意：当 AHB 时钟的预分频系数大于 1 时，必须开启预取缓冲器。

Bits 3:2 **SWS[1:0]**：系统时钟切换状态

指示哪一个时钟源被作为系统时钟。

00：HSI 作为系统时钟

01：保留

10：PLL 作为系统时钟

11：LSI 作为系统时钟

Bits 1:0 **SW[1:0]**：系统时钟切换

在从停止或待机模式中返回时，由硬件强制选择 HSI 作为系统时钟

00：HSI 作为系统时钟

01：保留
10：PLL 作为系统时钟
11：LSI 作为系统时钟

7.3.3 时钟中断寄存器（RCC_CIR）

地址偏移：0x08
复位值：0x0000 0000
访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											PLL RDYC	Res.	HSI RDYC	Res.	LSI RDYC
											w		w		w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			PLL RDYIE	Res.	HSI RDYIE	Res.	LSI RDYIE	Reserved			PLL RDYF	Res.	HSI RDYF	Res.	LSI RDYF
			rw		rw		rw				r		r		r

Bits 31:21 保留，必须保持为复位值。

Bit 20 **PLLRDYC**：清除 PLL 就绪中断
0：无作用
1：清除 PLL 就绪中断标志位 PLLRDYF

Bit 19 保留，必须保持为复位值。

Bit 18 **HSIRDYC**：清除 HSI 就绪中断
0：无作用
1：清除 HSI 就绪中断标志位 HSIRDYF

Bit 17 保留，必须保持为复位值。

Bit 16 **LSIRDYC**：清除 LSI 就绪中断
0：无作用
1：清除 LSI 就绪中断标志位 LSIRDYF

Bits 15:13 保留，必须保持为复位值。

Bit 12 **PLLRDYIE**：PLL 就绪中断使能
0：PLL 就绪中断禁止
1：PLL 就绪中断使能

Bit 11 保留，必须保持为复位值。

Bit 10 **HSIRDYIE**：HSI 就绪中断使能
0：HSI 就绪中断禁止
1：HSI 就绪中断使能

Bit 9 保留，必须保持为复位值。

Bit 8 **LSIRDYIE**：LSI 就绪中断使能
0：LSI 就绪中断禁止
1：LSI 就绪中断使能

Bits 7:5 保留，必须保持为复位值。

Bit 4 **PLLRDYF**：PLL 就绪中断标志
在内部高速时钟就绪且 PLLRDYIE 位已被置“1”时，由硬件置“1”。由软件通过置“1” HSIRDYC 位来清除。
0：未产生 PLL 锁定就绪中断

1：已产生 PLL 锁定就绪中断

Bit 3 保留，必须保持为复位值。

Bit 2 **HSIRDYF**：HSI 就绪中断标志

在内部高速时钟就绪且 HSIRDYIE 位已被置“1”时，由硬件置“1”。由软件通过置“1” HSIRDYC 位来清除。

0：无内部 16MHz RC 振荡器产生的时钟就绪中断

1：内部 16MHz RC 振荡器导致时钟就绪中断

Bit 1 保留，必须保持为复位值。

Bit 0 **LSIRDYF**：LSI 就绪中断标志

在内部高速时钟就绪且 LSIRDYIE 位已被置“1”时，由硬件置“1”。由软件通过置“1” HSIRDYC 位来清除。

0：无内部 32KHz RC 振荡器产生的时钟就绪中断

1：内部 32KHz RC 振荡器导致时钟就绪中断

7.3.4 APB2 外设复位寄存器（RCC_APB2RSTR）

地址偏移：0x0C

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															TIM15RST
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	UART1RST	TIM8RST	SPI1RST	Reserved											SYSCFGRST
	rw	rw	rw												rw

Bits 31:17 保留，必须保持为复位值。

Bit 16 **TIM15RST**：TIM15 复位

0：无作用

1：复位 TIM15

Bit 15 保留，必须保持为复位值。

Bit 14 **UART1RST**：UART1 复位

0：无作用

1：复位 UART1

Bit 13 **TIM8RST**：TIM8 复位

0：无作用

1：复位 TIM8

Bit 12 **SPI1RST**：SPI1 复位

0：无作用

1：复位 SPI1

Bits 11:1 保留，必须保持为复位值。

Bit 0 **SYSCFGRST**：SYSCFG 复位

0：无作用

1：复位 SYSCFG + CMP + OPAMP+ VREFBUF 功能

7.3.5 APB1 外设复位寄存器（RCC_APB1RSTR）

地址偏移：0x10

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			PWR RST	Reserved						I2C1 RST	Reserved			UART2 RST	Res.
			rw							rw				rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										TIM7 RST	TIM6 RST	Reserved		TIM3 RST	TIM2 RST
										rw	rw			rw	rw

Bits 31:29 保留，必须保持为复位值。

Bit 28 **PWRRST**：电源接口复位

0：无作用

1：复位电源接口

Bits 27:22 保留，必须保持为复位值。

Bit 21 **I2C1RST**：I2C1 复位

0：无作用

1：复位 I2C1

Bits 20:18 保留，必须保持为复位值。

Bit 17 **UART2RST**：UART2 复位

0：无作用

1：复位 UART2

Bits 16:6 保留，必须保持为复位值。

Bit 5 **TIM7RST**：TIM7 复位

0：无作用

1：复位 TIM7

Bit 4 **TIM6RST**：TIM6 复位

0：无作用

1：复位 TIM6

Bits 3:2 保留，必须保持为复位值。

Bit 1 **TIM3RST**：TIM3 复位

0：无作用

1：复位 TIM3

Bit 0 **TIM2RST**：TIM2 复位

0：无作用

1：复位 TIM2

7.3.6 AHB 外设时钟使能寄存器（RCC_AHBENR）

地址偏移：0x14

复位值：0x0000 0014

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							ADC1EN	Reserved				IOPDEN	IOPCEN	IOPBEN	IOPAEN
							rw					rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				MEEN	Reserved				CRCEN	Res.	FLITFEN	Res.	SARMEN	Reserved	
				rw					rw		rw		rw		

Bits 31:25 保留，必须保持为复位值。

Bit 24 **ADC1EN**：ADC1 接口时钟使能

0：关闭 ADC1 接口时钟；

1：开启 ADC1 接口时钟。

Bits 23:20 保留，必须保持为复位值。

Bit 19 **IOPDEN**：IO 端口 D 时钟使能

0：关闭 IO 端口 D 时钟；

1：开启 IO 端口 D 时钟。

Bit 18 **IOPCEN**：IO 端口 C 时钟使能

0：关闭 IO 端口 C 时钟；

1：开启 IO 端口 C 时钟。

Bit 17 **IOPBEN**：IO 端口 B 时钟使能

0：关闭 IO 端口 B 时钟；

1：开启 IO 端口 B 时钟。

Bit 16 **IOPAEN**：IO 端口 A 时钟使能

0：关闭 IO 端口 A 时钟；

1：开启 IO 端口 A 时钟。

Bits 15:12 保留，必须保持为复位值。

Bit 11 **MEEN**：ME 时钟使能

0：关闭 ME 时钟；

1：开启 ME 时钟。

Bits 10:7 保留，必须保持为复位值。

Bit 6 **CRCEN**：CRC 时钟使能

0：关闭 CRC 时钟；

1：开启 CRC 时钟。

Bit 5 保留，必须保持为复位值。

Bit 4 **FLITFEN**：FLITF 时钟使能

0：睡眠模式时闪存接口电路时钟关闭；

1：睡眠模式时闪存接口电路时钟开启。

Bit 3 保留，必须保持为复位值。

Bit 2 **SRAMEN**：SRAM 时钟使能

0：睡眠模式时 SRAM 时钟关闭；

1：睡眠模式时 SRAM 时钟开启。

Bit 1:0 保留，必须保持为复位值。

7.3.7 APB2 外设时钟使能寄存器（RCC_APB2ENR）

地址偏移：0x18

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIM15 EN	UART1 EN	TIM8 EN	SPI1 EN	Reserved											SYSCFG EN
rw	rw	rw	rw												rw

Bits 31:16 保留，必须保持为复位值。

Bit 15 **TIM15EN**：TIM15 时钟使能

0：关闭 TIM15 时钟

1：开启 TIM15 时钟

Bit 14 **UART1EN**：UART1 时钟使能

0：关闭 UART1 时钟

1：开启 UART1 时钟

Bit 13 **TIM8EN**：TIM8 时钟使能

0：关闭 TIM8 时钟

1：开启 TIM8 时钟

Bit 12 **SPI1EN**：SPI1 时钟使能

0：关闭 SPI1 时钟

1：开启 SPI1 时钟

Bits 11:1 保留，必须保持为复位值。

Bit 0 **SYSCFGEN**：SYSCFG 时钟使能

0：关闭 SYSCFG + CMP + OPAMP + VREFBUF 时钟

1：开启 SYSCFG + CMP + OPAMP + VREFBUF 时钟

7.3.8 APB1 外设时钟使能寄存器（RCC_APB1ENR）

地址偏移：0x1C

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			PWREN	Reserved						I2C1EN	Reserved			UART2EN	Res.
			rw							rw				rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										TIM7EN	TIM6EN	Reserved		TIM3EN	TIM2EN
										rw	rw			rw	rw

Bits 31:29 保留，必须保持为复位值。

Bit 28 **PWREN**：电源接口时钟使能

0：关闭电源接口时钟

1：开启电源接口时钟

Bits 27:22 保留，必须保持为复位值。

Bit 21 **I2C1EN**：I2C1 时钟使能

0：关闭 I2C1 时钟

1：开启 I2C1 时钟

Bits 20:18 保留，必须保持为复位值。

Bit 17 **UART2EN**：UART2 时钟使能

0：关闭 UART2 时钟

1：开启 UART2 时钟

Bits 16:6 保留，必须保持为复位值。

Bit 5 **TIM7EN**：TIM7 时钟使能

0：关闭 TIM7 时钟

1：开启 TIM7 时钟

Bit 4 **TIM6EN**：TIM6 时钟使能

0：关闭 TIM6 时钟

1：开启 TIM6 时钟

Bits 3:2 保留，必须保持为复位值。

Bit 1 **TIM3EN**：TIM3 时钟使能

0：关闭 TIM3 时钟

1：开启 TIM3 时钟

Bit 0 **TIM2EN**：TIM2 时钟使能

0：关闭 TIM2 时钟

1：开启 TIM2 时钟

7.3.9 备份域控制寄存器（RCC_BDCR）

地址偏移：0x20
复位值：0x0000 0000，只能由备份域复位有效复位
访问：0 到 3 等待周期，支持字，半字和字节访问

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

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

Bit 31:17 保留，必须保持为复位值。

Bit 16 **BDRST**：RTC 软件复位

0：未激活复位

1：复位 RTC

Bit 15 **RTCEN**：RTC 时钟使能

0：关闭 RTC 时钟

1：开启 RTC 时钟

Bit 14:0 保留，必须保持为 0。

7.3.10 控制/状态寄存器（RCC_CSR）

地址偏移：0x24
复位值：0x0C00 0000，除复位标志外由系统复位清除，复位标志只能由电源复位清除
访问：0 到 3 等待周期，支持字，半字和字节访问
当连续对该寄存器进行访问时，将插入等待状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPWR RSTF	Res.	IWDG RSTF	SFT RSTF	POR RSTF	PIN RSTF	OBL RSTF	RMVF	Reserved							
rw		rw	rw	rw	rw	rw	rw								

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

Bit 31 **LPWRRSTF**：低功耗复位标志

在低功耗管理复位发生时由硬件置‘1’，由软件通过写 RMVF 位清除。

0：未发生低功耗管理复位

1：发生低功耗管理复位

Bit 30 保留，必须保持为复位值。

Bit 29 **IWDGRSTF**：独立看门狗复位标志

在独立看门狗复位发生在 V_{DD} 区域时由硬件置‘1’，由软件通过写 RMVF 位清除。

0：未发生独立看门狗复位

1：发生独立看门狗复位

Bit 28 **SFTRSTF**：软件复位标志

在软件复位发生时由硬件置‘1’，由软件通过写 RMVF 位清除。

0：未发生软件复位

1：发生软件复位

Bit 27 **PORRSTF**：上电/掉电复位标志

在上电/掉电复位发生时由硬件置‘1’，由软件通过写 RMVF 位清除。

0：未发生上电/掉电复位

1：发生上电/掉电复位

Bit 26 **PINRSTF**：NRST 引脚复位标志

在 NRST 引脚复位发生时由硬件置‘1’，由软件通过写 RMVF 位清除。

0：未发生 NRST 引脚复位

1：发生 NRST 引脚复位

Bit 25 **OBLRSTF**：选项字节加载器复位标志

0：选项字节加载器没有复位

1：选项字节加载器复位

Bit 24 **RMVF**：清除复位标志

0：无作用

1：清除复位标志

Bits 23:2 保留，必须保持为复位值。

Bit 1 **LSIRDY**：内部低速振荡器就绪

由硬件置 1 或清 0 来指示内部 32 KHz RC 振荡器是否就绪。在 LSION 清零后，3 个内部 32 KHz RC 振荡器的周期后 LSIRDY 被清零。

0：内部 32 KHz RC 振荡器时钟未就绪

1：内部 32 KHz RC 振荡器时钟就绪

Bit 0 **LSION**：内部低速振荡器使能

0：关闭内部 32 KHz RC 振荡器

1：使能内部 32 KHz RC 振荡器

7.3.11 AHB 外设复位寄存器（RCC_ AHBRSTR）

地址偏移：0x28

复位值：0x0000 0000，除复位标志外由系统复位清除，复位标志只能由电源复位清除

访问：0 到 3 等待周期，支持字，半字和字节访问

当连续对该寄存器进行访问时，将插入等待状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							ADC1 RST	Reserved				IOPD RST	IOPC RST	IOPB RST	IOPA RST
							rw					rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				ME RST	Reserved				CRC RST	Reserved					
				rw					rw						

Bits 31:25 保留，必须保持为复位值。

Bit 24 **ADC1RST**：ADC1 接口复位

0：无作用

1：复位 ADC1 接口

Bits 23:20 保留，必须保持为复位值。

Bit 19 **IOPDRST**：IO 端口 D 复位
0：无作用
1：复位 IO 端口 D
Bit 18 **IOPCRST**：IO 端口 C 复位
0：无作用
1：复位 IO 端口 C
Bit 17 **IOPBRST**：IO 端口 B 复位
0：无作用
1：复位 IO 端口 B
Bit 16 **IOPARST**：IO 端口 A 复位
0：无作用
1：复位 IO 端口 A
Bits 15:12 保留，必须保持为复位值。
Bit 11 **MERST**：ME 接口复位
0：无作用
1：复位 ME 接口
Bits 10:7 保留，必须保持为复位值。
Bit 6 **CRCRST**：CRC 复位
0：无作用
1：复位 CRC
Bits 5:0 保留，必须保持为复位值。

8 通用和复用功能 I/O（GPIO）

8.1 简介

每个通用 I/O 有 4 个 32 位配置寄存器（GPIOx_MODER，GPIOx_OTYPER，GPIOx_OSPEEDR 和 GPIOx_PUPDR），2 个 32 位数据寄存器（GPIOx_IDR 和 GPIOx_ODR），1 个 32 位置位/复位寄存器（GPIOx_BSRR），1 个 16 位复位寄存器（GPIOx_BRR），另外。所有 GPIO 有 1 个 32 位锁位寄存器（GPIOx_LCKR）和 1 个复用功能配置寄存器（GPIOx_AFRL）。

8.2 GPIO 主要特征

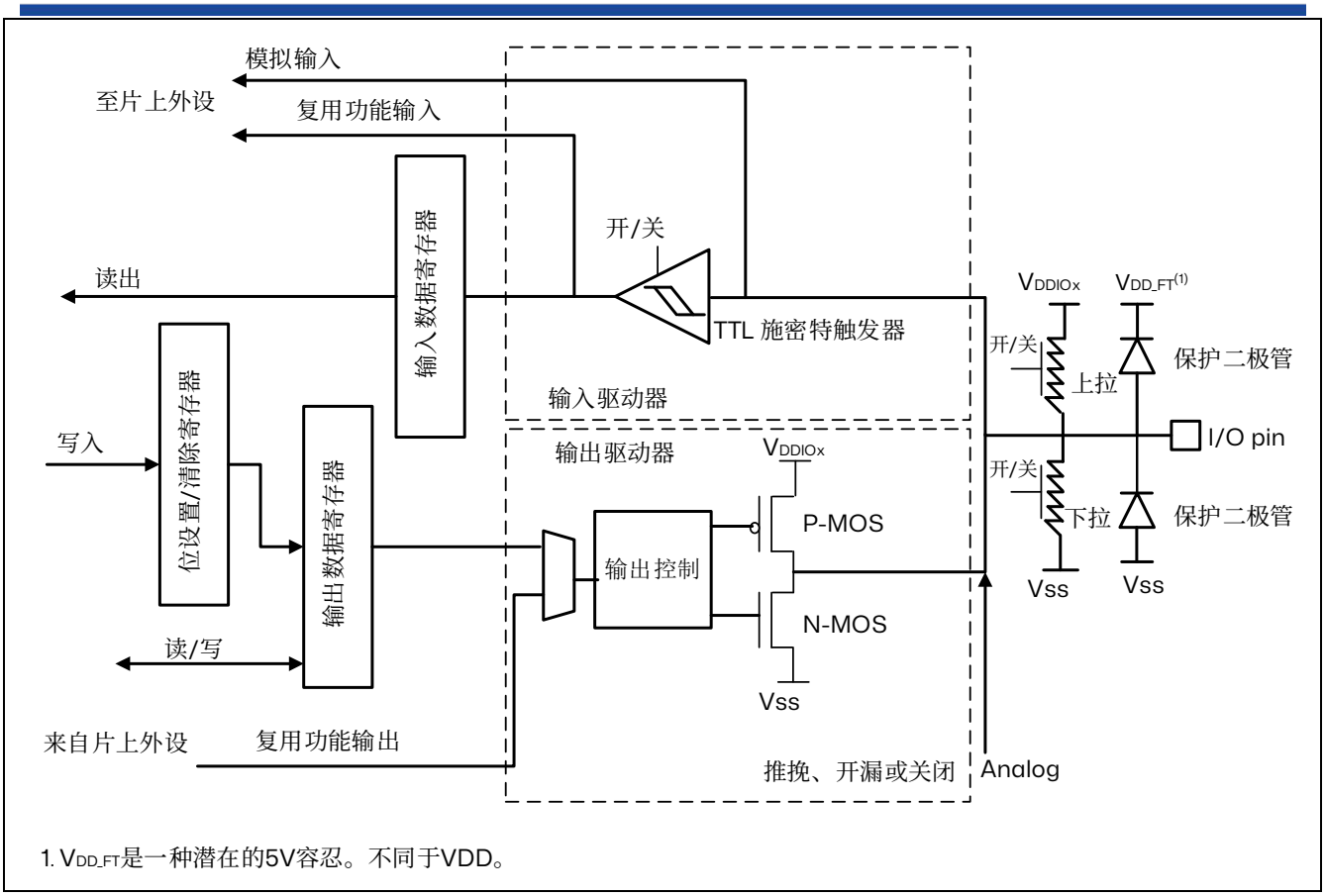
- 输出状态：推挽或开漏 + 上拉/下拉
- 从输出数据寄存器（GPIOx_ODR）或外设（复用功能输出）输出数据
- I/O 速度可选
- 输入状态：浮空，上拉/下拉，模拟
- 从输入数据寄存器（GPIOx_IDR）或外设（复用功能输入）输入数据
- 置位和清零寄存器（GPIOx_BSRR）对 GPIOx_ODR 进行访问
- 提供用来冻结 I/O 端口配置的锁定机制（GPIOx_LCKR）
- 模拟功能
- 复用功能选择寄存器
- 高度灵活的引脚复用允许使用 I/O 引脚作为 GPIO 或者几个外设复用功能之一

8.3 GPIO 功能描述

根据数据手册中列出的每个 I/O 端口的特定硬件特征，GPIO 端口的每个位可以由软件分别配置成多种模式：

- 输入浮空
- 输入上拉
- 输入下拉
- 模拟
- 上拉/下拉开漏输出
- 上拉/下拉推挽输出
- 上拉/下拉复用开漏输出
- 上拉/下拉复用推挽输出

每个 I/O 端口位可以自由编程，I/O 端口寄存器支持字、半字或字节访问。GPIOx_BSRR 寄存器旨在实现对 GPIO ODR 寄存器进行读取/修改访问。这样便可确保在读取和修改访问之间发生中断请求也不会有问题。



1. V_{DD_FT}是一种潜在的5V容忍。不同于VDD。

图 8.1 I/O 端口位基本结构

表 8.1 端口位配置表 (1)

MODE[1:0]	OTYPE	OSPEED[1:0]	PUPD[1:0]	I/O 配置	
01	0	SPEED[1:0]	00	GP 输出	PP
	0		01	GP 输出	PP + PU
	0		10	GP 输出	PP + PD
	0		11	保留	
	1		00	GP 输出	OD
	1		01	GP 输出	OD + PU
	1		10	GP 输出	OD + PD
	1		11	保留（GP 输出 OD）	
10	0	SPEED[1:0]	00	AF	PP
	0		01	AF	PP + PU
	0		10	AF	PP + PD
	0		11	保留	
	1		00	AF	OD
	1		01	AF	OD + PU
	1		10	AF	OD + PD
	1		11	保留	
00	x	x	00	输入	浮空
	x	x	01	输入	PU
	x	x	10	输入	PD
	x	x	11	保留（输入浮空）	
11	x	x	00	输入/输出	模拟
	x	x	01	保留	
	x	x	10		
	x	x	11		

1. GP = 通用功能，PP = 推挽，PU = 上拉，PD = 下拉，OD = 开漏，AF = 复用。

8.3.1 通用 IO（GPIO）

复位期间和刚复位后，复用功能未开启，I/O 端口默认是浮空模拟。

复位后，调试引脚被置于复用上拉或下拉模式：

- PA1：DBGCLK 下拉
- PA0：DBGIO 上拉

当作为输出配置时，写到输出数据寄存器上的值（GPIOx_ODR）输出到相应的 I/O 引脚。可以以推挽模式或开漏模式使用输出驱动器。

输入数据寄存器（GPIOx_IDR）在每个 AHB1 时钟周期捕捉 IO 引脚上的数据。

所有 GPIO 引脚有一个内部弱上拉和弱下拉，可以通过 GPIOx_PUPDR 寄存器配置。

8.3.2 IO 引脚复用功能

芯片的 I/O 引脚通过一个多路复用器连接到外设模块，同时只能将一个外设复用功能连接到一个 I/O。因此，同一个引脚上的复用功能不会出现冲突。

每一个 I/O 引脚有一个 8 路多路复用器（AF0 到 AF7），可以通过 GPIOx_AFRL（引脚 0 到 7）：

- 复位后，多路复用器选在复用功能 0（AF0）。可以通过 GPIOx_MODER 寄存器将 I/O 配置成复用功能模式。
- 引脚的复用功能请参考芯片数据手册。

除了这种灵活的 I/O 复用架构之外，各外设还可以将复用功能映射到不同 I/O 引脚，这可以优化小型封装中可用外设的数量。

要将 I/O 配制成所需功能，请按照以下步骤操作：

- **调试功能：**在芯片复位后，会将这些引脚指定为专用引脚，可供调试器立即使用。
- **GPIO：**使用 GPIOx_MODER 寄存器将所需的 I/O 配置为输出或输入模式。
- 外设复用功能：
  - 使用 GPIOx_AFRL 将 I/O 连接到指定的复用功能上。
  - 使用 GPIOx_OTYPER，GPIOx_PUPDR 和 GPIOx_OSPEEDER 寄存器分别配置类型、上拉/下拉和输出速度。
  - 使用 GPIOx_MODER 寄存器将 I/O 配置为复用模式。
- 模拟功能：对于 ADC、CMP、OPAMP 等模拟外设，在 GPIOx_MODER 寄存器中将所需 I/O 配置为模拟通道。

8.3.3 I/O 端口控制寄存器

每个 GPIO 端口有 4 个 32 位内存映射控制寄存器（GPIOx_MODER，GPIOx_OTYPER，GPIOx_OSPEEDR 和 GPIOx_PUPDR），可以配置最多 8 个 I/O。GPIOx_MODER 寄存器用于选择 I/O 模式（输入，输出，复用，模拟）。GPIOx_OTYPER 和 GPIOx_OSPEEDR 寄存器用于选择输出类型（推挽或开漏）和速度。GPIOx_PUPDR 寄存器用于选择任意 I/O 方向的上拉/下拉。

8.3.4 I/O 端口数据寄存器

每个 GPIO 有两个 32 位内存映射数据寄存器：输入和输出数据寄存器（GPIOx_IDR 和 GPIOx_ODR）。GPIOx_ODR 存储要输出的数据，可读可写。通过 I/O 输入的数据存储在输入数据寄存器（GPIOx_IDR）中，这是一个只读寄存器。

8.3.5 I/O 数据位处理

1. 置位和复位寄存器（GPIOx_BSRR）是一个 32 位寄存器，允许应用程序置位和复位输出数据寄存器（GPIOx_ODR）中的每个位。
   1) 对于 GPIOx_ODR 中的每一位，对应 GPIOx_BSRR 中的两个控制位：BSx 和 BRx。当 BSx 写入 1 时，置位相应的 ODRx 位。当 BRx 写入 1 时，复位 ODRx 对应的位。
   2) 将 GPIOx_BSRR 中的任何位写入 0 不会对 GPIOx_ODR 中的相应位产生任何影响。如果同时对 BSx 和 BRx 写 1，则置位操作具有优先级。
2. 复位寄存器（GPIOx_BRR）是一个 32 位寄存器，允许应用程序复位输出数据寄存器（GPIOx_ODR）中的每个位。
   1) 对于 GPIOx_ODR 中的每一位，对应 GPIOx_BRR 中的一个控制位：BRx。当 BRx 写入 1 时，复位 ODRx 对应的位。
   2) 将 GPIOx_BRR 中的任何位写入 0 不会对 GPIOx_ODR 中的相应位产生任何影响。

使用 GPIOx_BSRR/GPIOx_BRR 寄存器来更改 GPIOx_ODR 中单个位的值是“一次”效果，它不会锁定 GPIOx_ODR 位。GPIOx_ODR 位始终可以直接访问。GPIOx_BSRR/GPIOx_BRR 寄存器提供了一种执行位处理的方法。

在位级编程 GPIOx_ODR 时，软件不需要禁用中断：可以在单个 AHB1 写访问中修改一个或多个位。

8.3.6 I/O 锁定机制

通过对 GPIOx_LCKR 寄存器应用特定的写序列，可以冻结 GPIO 控制寄存器。冻结寄存器是 GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR、GPIOx_AFRL 和 GPIOx_AFRH。

要写入 GPIOx_LCKR 寄存器，必须应用特定的写/读序列。当正确的 LOCK 序列应用于该寄存器的第 16 位时，LCKR[7:0]的值用于锁定 I/O 的配置（在写入序列中 LCKR[7:0]的值必须相同）。当 LOCK 序列应用于端口位时，该端口位的值不能再被修改，直到下一次 MCU 复位或外设复位。每个 GPIOx_LCKR 位冻结控制寄存器中相应的位（GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR、GPIOx_AFRL）。

LOCK 序列只能通过对 GPIOx_LCKR 寄存器的一个字（32 位长）访问来执行，因为 GPIOx_LCKR 位 16 必须与[7:0]位同时设置。

8.3.7 I/O 复用功能输入/输出

提供了一个寄存器 AFAR 来为每个 I/O 选择一个可用的复用输入/输出。使用这些寄存器，用户可以根据应用程序的需要将复用功能连接到其他引脚。

使用 GPIOx_AFRL 复用寄存器在每个 GPIO 上多路复用外设功能。因此，应用程序可以为每个 I/O 选择任意一个功能。AF 选择信号对于复用输入和复用输出是共性的，因此对于给定 I/O 的复用功能输入/输出选择单个通道。

具体 GPIO 的复用功能情况请参考数据手册。

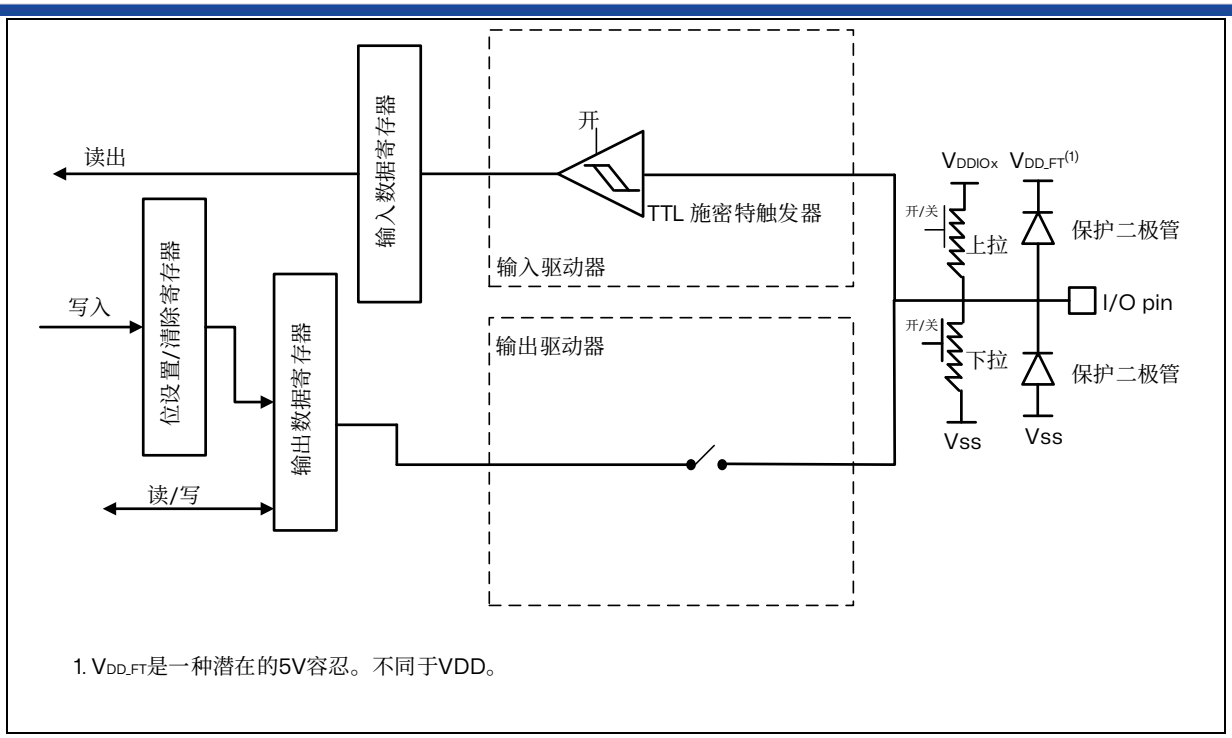
8.3.8 外部中断/唤醒线

所有端口都具有外部中断功能。要使用外部中断功能，需要将端口配置为输入模式。

8.3.9 输入模式配置

当 I/O 端口被配置为输入时：

- 输出缓冲区被禁用
- Schmitt 触发器输入激活
- 根据 GPIOx_PUPDR 寄存器中的值，上拉和下拉电阻被激活
- 每个 AHB 时钟周期，I/O 引脚上的数据被采样到的输入数据寄存器
- 输入数据寄存器提供 I/O 状态



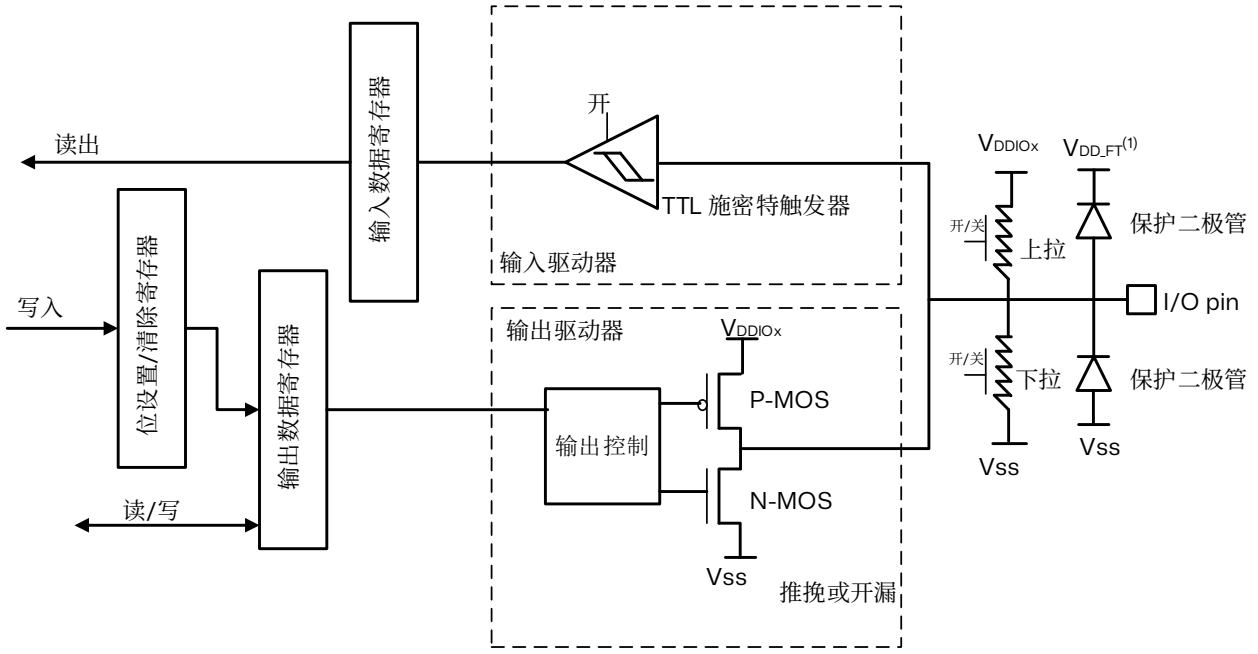
1. V_{DD_FT}是一种潜在的5V容忍。不同于VDD。

图 8.2 输入浮空/上拉/下拉结构

8.3.10 输出模式配置

当 I/O 端口被配置为输出时：

- 输出缓冲区已启用：
- 开漏模式：输出寄存器中的“0”开启 N-MOS，而输出寄存器中的“1”使端口处于 Hi-Z 状态（P-MOS 未被开启）
- 推挽模式：输出寄存器中的“0”开启 N-MOS，而输出寄存器中的“1”开启 P-MOS
- Schmitt 触发器输入激活
- 根据 GPIOx_PUPDR 寄存器中的值，上拉和下拉电阻被激活
- 每个 AHB1 时钟周期，I/O 引脚上的数据被采样到的输入数据寄存器
- 输入数据寄存器提供 I/O 状态
- 输出数据寄存器可读取最后写入的值



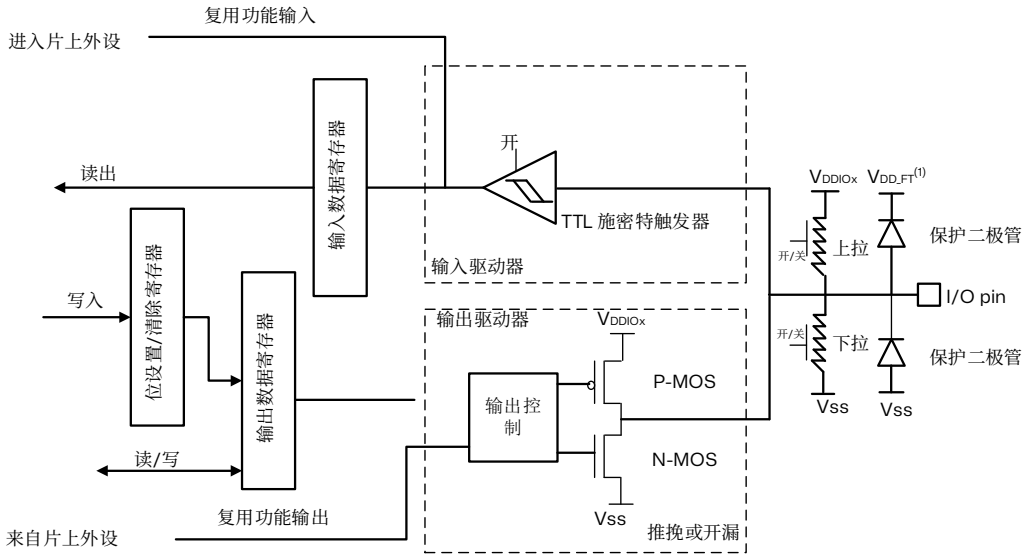
1. V_{DD_FT}是一种潜在的5V容忍。不同于VDD。

图 8.3 输出框图

8.3.11 复用功能配置

当 I/O 端口被配置为复用功能时：

- 输出缓冲区可以配置为开漏或推挽模式
- 输出缓冲区由来自外围设备的信号驱动（发射器使能和数据）
- Schmitt 触发器输入激活
- 根据 GPIOx_PUPDR 寄存器中的值，上拉和下拉电阻被激活
- 每个 AHB1 时钟周期，I/O 引脚上的数据被采样到输入数据寄存器
- 输入数据寄存器提供 I/O 状态



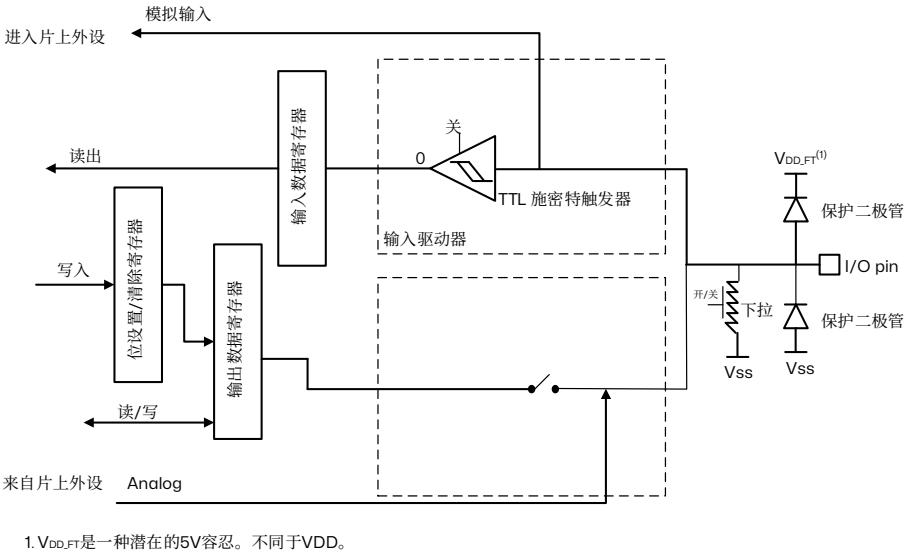
1. V_{DD_FT}是一种潜在的5V容忍。不同于VDD。

图 8.4 复用功能框图

8.3.12 模拟功能配置

当 I/O 端口配置为模拟配置时：

- 输出缓冲区被禁用
- Schmitt 触发输入是失活的，为 I/O 引脚的每个模拟值提供零消耗。Schmitt 触发器的输出被强制为恒定值（0）。
- 弱上拉被硬件禁用。弱下拉是可配置的。
- 输入数据寄存器读得到值固定为“0”



1. V_{DD_FT}是一种潜在的5V容忍。不同于VDD。

图 8.5 模拟引脚框图

8.4 GPIO 寄存器

访问：无等待，支持字，半字和字节访问

8.4.1 GPIO 端口模式寄存器（GPIOx_MODER）（x = A to D）

地址偏移：0x00

复位值：0xFFFF FFFA for port A

复位值：0xFFFF FFFF for other ports

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

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

Bits 31:16 保留，必须保持为复位值

Bits 2y:2y+1 **MODERy[1:0]：**端口 x 的 I/O 引脚 y 的模式配置位（y = 7 至 0）

这些位通过软件写入，用于配置 I/O 方向模式。

00：输入

01：通用输出模式

10：复用功能模式

11：模拟模式（复位状态）

8.4.2 GPIO 端口输出类型寄存器（GPIOx_OTYPE）（x = A to D）

地址偏移：0x04

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
								rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:8 保留，必须保持为复位值

Bits 7:0 **OTy：**端口 x 的 I/O 引脚 y 的输出类型配置位（y = 7 至 0）

这些位由软件写入，用于配置 I/O 输出类型。

0：输出推挽（复位状态）

1：输出开漏

8.4.3 GPIO 端口输出速度寄存器（GPIOx_OSPEEDR）（x = A to D）

地址偏移：0x08

复位值：0x0000 0003 for port A

复位值：0x0000 0000 for other ports

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSPEED7[1:0]		OSPEED6[1:0]		OSPEED5[1:0]		OSPEED4[1:0]		OSPEED3[1:0]		OSPEED2[1:0]		OSPEED1[1:0]		OSPEED0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 保留，必须保持为复位值

Bits 2y:2y+1 **OSPEEDRy[1:0]：**端口 x 的 I/O 引脚 y 的输出速度配置位（y = 7 至 0）

这些位通过软件写入，用于配置 I/O 输出速度。

00：低速 CL=50pF，输出频率 MAX：5MHz；

01：中速 CL=50pF，输出频率 MAX：20MHz；

10：高速 CL=50pF，输出频率 MAX：35MHz；

11：超高速 CL=30pF 输出频率 MAX：50MHz；

8.4.4 GPIO 端口上/下拉寄存器（GPIOx_PUPDR）（x = A to D）

地址偏移：0x0C

复位值：0x0000 0009 for port A

复位值：0x0000 0000 for other ports

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUPD7[1:0]		PUPD6[1:0]		PUPD5[1:0]		PUPD4[1:0]		PUPD3[1:0]		PUPD2[1:0]		PUPD1[1:0]		PUPD0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 保留，必须保持为复位值

Bits 2y:2y+1 **PUPDRy[1:0]：**端口 x 的 I/O 引脚 y 的上下拉配置位（y = 7 至 0）

这些位通过软件写入，用于配置 I/O 上拉或下拉。

00：无上拉或下拉

01：上拉

10：下拉

11：保留

8.4.5 GPIO 端口输入数据寄存器（GPIOx_IDR）（x = A to D）

地址偏移：0x10

复位值：0x0000 XXXX

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
								r	r	r	r	r	r	r	r

Bits 31:8 保留，必须保持为复位值。

Bits 7:0 **IDRy：**端口 x 的 I/O 引脚 y 的输入数据（y = 7 至 0）

这些位是只读的。它们包含相应 I/O 端口的输入值。

8.4.6 GPIO 端口输出数据寄存器（GPIOx_ODR）（x = A to D）

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
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
								rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:8 保留，必须保持为复位值。

Bits 7:0 **ODRy：**端口 x 的 I/O 引脚 y 的输出数据（y = 7 至 0）

这些位可以由软件读取和写入。

注意：对于原子置位 / 复位，通过写入 GPIOx_BSRR *寄存器，可分别对* ODR *位进行置位和复位（*x = A..D*）。*

8.4.7 GPIO 端口位置位/清零寄存器（GPIOx_BSRR）（x = A to D）

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
Reserved								BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
								w	w	w	w	w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
								w	w	w	w	w	w	w	w

Bits 31:24 保留，必须保持为复位值。

Bits 23:16 **BRy：**复位端口 x 的 I/O 引脚 y（y = 7 至 0）

这些位只能写。读取这些位将返回值 0x0000。

0：对应 ODRx 位不动

1：复位对应的 ODRx 位

注：如果同时配置 BSx *和* BRx，*则* BSx *优先。*

Bits 15:8 保留，必须保持为复位值。

Bits 7:0 **BSy：**置位端口 x 的 I/O 引脚 y（y = 7 至 0）

这些位只能写。读取这些位将返回值 0x0000。

0：对应 ODRx 位不动

1：置位对应的 ODRx 位

8.4.8 GPIO 端口配置锁寄存器（GPIOx_LCKR）（x = A to D）

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
Reserved															LCKK
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
								rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:17 保留，必须保持为复位值

Bit 16 **LCKK：**锁定键

这个位可以随时读取。只能使用锁键写入序列修改。

0：端口配置锁位键未激活

1：端口配置锁位键已激活。下次系统复位前 GPIOx_LCKR 寄存器被锁定。

LOCK 键写顺序：

WR LCKR[16] = ‘1’ + LCKR[7:0]

WR LCKR[16] = ‘0’ + LCKR[7:0]

WR LCKR[16] = ‘1’ + LCKR[7:0]

RD LCKR

RD LCKR[16] = ‘1’（此读取操作是可选的，但可以用来确认锁键已被激活）

注意：在 LOCK *键写顺序中，* LCK[7:0]*的值不能改变。*

锁序列中的任何错误都将中止锁。

在端口的任何位上的第一个锁定序列之后， LCKK *位上的任何读访问都返回* 1*，直到下一次* MCU *复位或外围设备复位。*

Bits 15:8 保留，必须保持为复位值

Bits 7:0 **LCKy：**端口 x 的 I/O 引脚 y 的锁定位（y = 7 至 0）

这些位是可读/可写的，但只有当 LCKK 位为‘0’时才能写入。

0：端口配置不锁定

1：端口配置锁定

8.4.9 GPIO 复用功能低位寄存器（GPIOx_AFRL）（x = A to D）

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
AFRL7[3:0]				AFRL6[3:0]				AFRL5[3:0]				AFRL4[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFRL3[3:0]				AFRL2[3:0]				AFRL1[3:0]				AFRL0[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:0 **AFSELy[3:0]：**端口 x 的 I/O 引脚 y 的复用功能选择（y = 7 至 0）

这些位由软件写入，以配置备用功能 I/O。

0000：AF0

0001：AF1
0010：AF2
0011：AF3
0100：AF4
0101：AF5
0110：AF6
0111：AF7
其他：无作用

8.4.10 GPIO 端口位复位寄存器（GPIOx_BRR）（x = A to D）

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
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
								w	w	w	w	w	w	w	w

Bits 31:8 保留，必须保持为复位值

Bits 7:0 **BRy：**复位端口 x 的 I/O 引脚 y（y = 7 至 0）

这些位只能写。读取这些位将返回值 0x0000。

0：对应 ODRx 位不动

1：复位对应的 ODRx 位

9 系统配置控制器（SYSCFG）

9.1 SYSCFG 主要特征

RX32S632 系列具备一组系统配置寄存器。系统配置控制器的主要用途如下：

- 配置外部中断线连接到 GPIO

9.2 SYSCFG 寄存器

访问：无等待，支持字，半字和字节访问

9.2.1 外部中断配置寄存器 1（SYSCFG_EXTICR1）

地址偏移：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI3[3:0]				EXTI2[3:0]				EXTI1[3:0]				EXTI0[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 保留，必须保持为复位值

Bits 15:0 **EXTIx[3:0]**：EXTIx 配置（x = 3 至 0）

这些位可由软件读写，用于选择 EXTIx 外部中断的输入源。

0000：PAx

0001：PBx

0010：PCx

0011：PDx

9.2.2 外部中断配置寄存器 2（SYSCFG_EXTICR2）

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
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI7[3:0]				EXTI6[3:0]				EXTI5[3:0]				EXTI4[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 保留，必须保持为复位值

Bits 15:0 **EXTIx[3:0]**：EXTIx 配置（x = 7 至 4）

这些位可由软件读写，用于选择 EXTIx 外部中断的输入源。

0000：PAx

0001：PBx

0010：PCx

0011：PDx

9.2.3 CYSCFG 配置寄存器 2（SYSCFG_CFGR2）

地址偏移：0x1C

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

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

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

Bits 31:3 保留，必须保持为复位值

Bit 2 **PVDL：**PVD 锁存使能位

该位可以由软件置位，仅在系统复位时清零。该位和 PWR_CR2 寄存器的 PVDE 和 PLS[2:0]位用于将 PVD 连接到 TIM8/15 的刹车输入。

0：PVD 中断与 TIM8/15 的刹车输入断开；

1：PVD 中断与 TIM8/15 的刹车输入连接。

注意：该位置位后，PVDE 和 PLS[2:0]只读。

Bit 1 保留，必须保持为复位值

Bit 0 **CLL：**内核锁定（硬件错误）输出使能位

该位可以由软件置位，仅在系统复位时清零。该位用于将内核内核锁定信号（硬件错误）输出连接到 TIM8/15 的刹车输入。

0：内核锁定输出与 TIM8/15 的刹车输入断开；

1：内核锁定输出与 TIM8/15 的刹车输入连接。

10 中断和事件控制器（EXTI）

10.1 简介

EXTI 主要特征如下：

- 每个中断/事件都有独立的触发和屏蔽
- 每个中断线都有专用的状态位
- 支持多达 12 个软件事件/中断请求
- 可配置上升沿和下降沿检测

10.2 框图

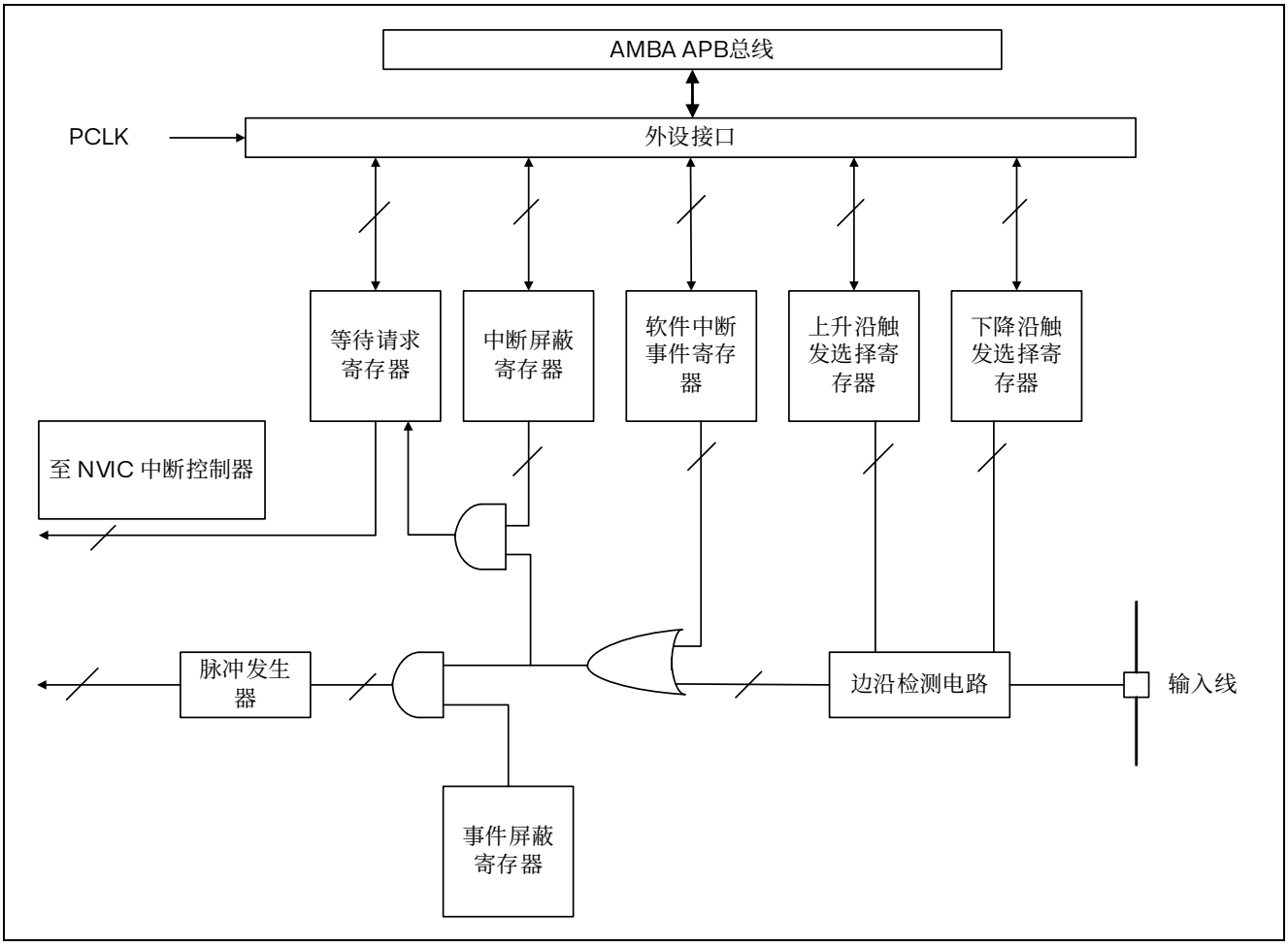


图 10.1 外部中断/事件控制器框图

10.3 唤醒事件管理

RX32S632 可以处理外部和内部事件来唤醒内核。唤醒事件可由下述配置产生：

- 在外设的控制寄存器使能一个中断，当内核唤醒后，需要清除相应外设的中断标志位
- 配置一个外部或内部 EXTI 线为事件模式，当内核唤醒后，因为对应事件线的挂起位没有被置位，不必清除相应中断挂起位。

10.4 功能说明

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。当外部中断线上出现选定信号沿时，便会产生中断请求，对应的挂起位也会置 1。在挂起寄存器的对应位写“1”，将清除该中断请求。

要产生事件，必须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。当事件线上出现选定信号沿时，便会产生事件脉冲，对应的挂起位不会置 1。

通过在软件中对软件中断/事件寄存器写“1”，也可以产生中断/事件请求。

硬件中断选择

要配置 12 根线作为中断源，请执行以下步骤：

- 配置 12 根中断线的屏蔽位（EXTI_IMR）
- 配置中断线的触发选择位（EXTI_RTSR 和 EXTI_FTSR）
- 配置对应到外部中断控制器（EXTI）的中断通道的使能和屏蔽位，使得 12 个中断线中的请求可以被正确地响应。

硬件事件选择

要配置 12 根线作为事件源，请执行以下步骤：

- 配置 12 根事件线的屏蔽位（EXTI_EMR）
- 配置事件线的触发选择位（EXTI_RTSR 和 EXTI_FTSR）

软件中断/事件选择

可将这 12 根线配置为软件中断/事件线。以下为产生软件中断的步骤。

- 配置 12 根中断/事件线的屏蔽位（EXTI_IMR、EXTI_EMR）
- 在软件中断寄存器设置相应的请求位（EXTI_SWIER）

10.5 外部中断/事件线路映像

通用 I/O 端口以下图的方式连接到 8 个外部中断/事件线上：

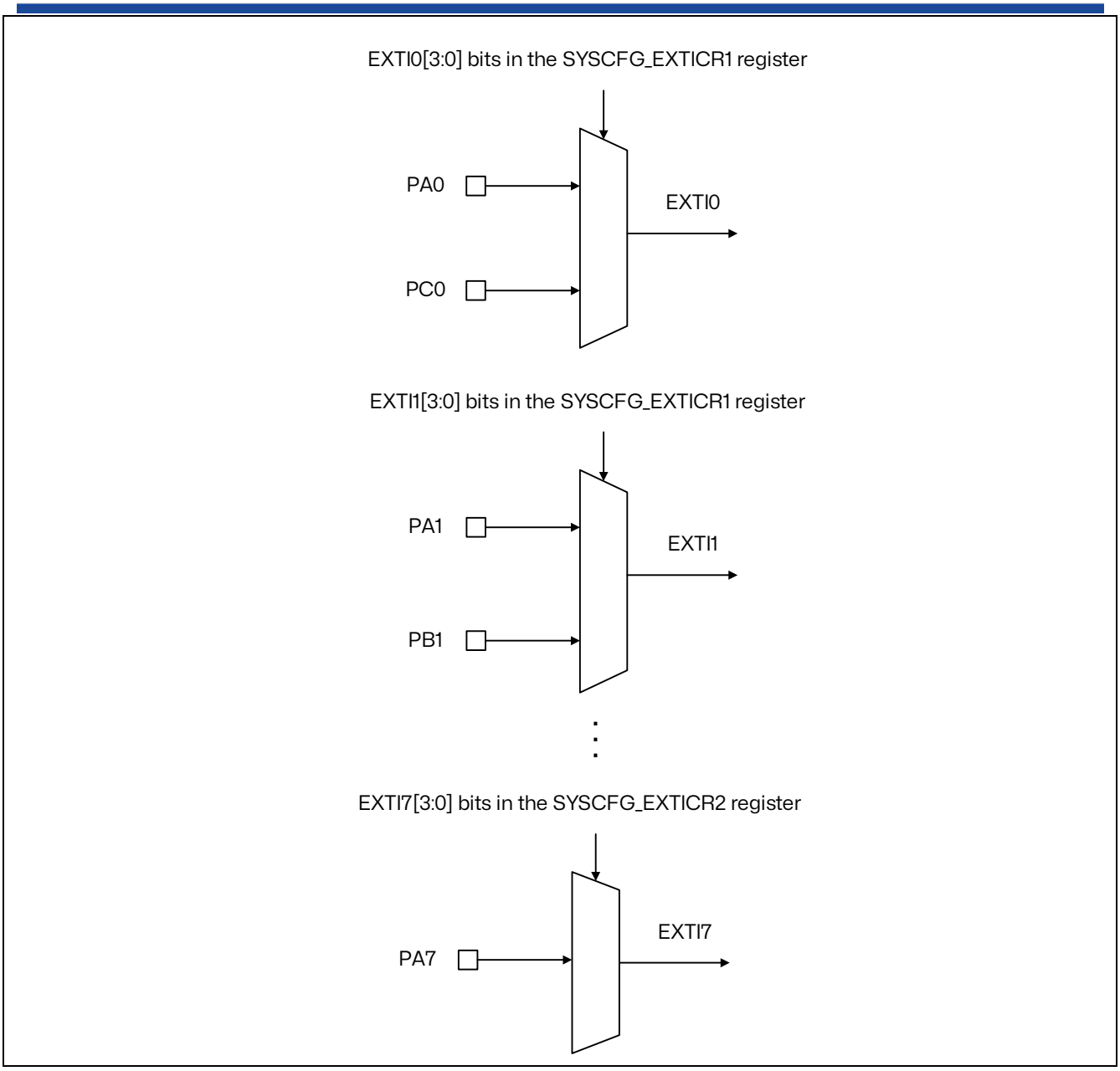


图 10.2 外部中断通用 I/O 映像

另外 4 个 EXTI 线的连接方式如下：

- EXTI 线 16 连接到 PVD 输出
- EXTI 线 17 连接到 RTC 闹钟事件
- EXTI 线 21 连接到 CMP1 比较器
- EXTI 线 22 连接到 CMP2 比较器

10.6 EXTI 寄存器

访问：无等待，支持字，半字和字节访问

10.6.1 中断屏蔽寄存器 1（EXTI_IMR1）

地址偏移：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									IM22	IM21	Reserved			IM17	IM16
									rw	rw				rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
								rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:23 保留，必须保持为复位值

Bits 22:21 **IMx：**线 x 上的中断屏蔽（x = 22 至 21）

0：屏蔽来自线 x 上的中断请求

1：开放来自线 x 上的中断请求

Bits 20:18 保留，必须保持为复位值

Bits 17:16 **IMx：**线 x 上的中断屏蔽（x = 17 至 16）

0：屏蔽来自线 x 上的中断请求

1：开放来自线 x 上的中断请求

Bits 15:8 保留，必须保持为复位值

Bits 7:0 **IMx：**线 x 上的中断屏蔽（x = 7 至 0）

0：屏蔽来自线 x 上的中断请求

1：开放来自线 x 上的中断请求

10.6.2 事件屏蔽寄存器 1（EXTI_EMR1）

地址偏移：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									EM22	EM21	Reserved			EM17	EM16
									rw	rw				rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								EM7	EM6	EM5	EM4	EM3	EM2	EM1	EM0
								rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:23 保留，必须保持为复位值

Bits 22:21 **EMx：**线 x 上的事件屏蔽（x = 22 至 21）

0：屏蔽来自线 x 上的事件请求

1：开放来自线 x 上的事件请求

Bits 20:18 保留，必须保持为复位值

Bits 17:16 **EMx：**线 x 上的事件屏蔽（x = 17 至 16）

0：屏蔽来自线 x 上的事件请求

1：开放来自线 x 上的事件请求

Bits 15:8 保留，必须保持为复位值

Bits 7:0 **EMx：**线 x 上的事件屏蔽（x = 7 至 0）

0：屏蔽来自线 x 上的事件请求

1：开放来自线 x 上的事件请求

10.6.3 上升沿触发选择寄存器 1（EXTI_RTSR1）

地址偏移：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									RT22	RT21	Reserved			RT17	RT16
									rw	rw				rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								RT7	RT6	RT5	RT4	RT3	RT2	RT1	RT0
								rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:23 保留，必须保持为复位值

Bits 22:21 **RTx：**线 x 上的上升沿触发事件配置（x = 22 至 21）

0：禁止输入线 x 上的上升沿触发（中断和事件）

1：允许输入线 x 上的上升沿触发（中断和事件）

Bits 20:18 保留，必须保持为复位值

Bits 17:16 **RTx：**线 x 上的上升沿触发事件配置（x = 17 至 16）

0：禁止输入线 x 上的上升沿触发（中断和事件）

1：允许输入线 x 上的上升沿触发（中断和事件）

Bits 15:8 保留，必须保持为复位值

Bits 7:0 **RTx：**线 x 上的上升沿触发事件配置（x = 7 至 0）

0：禁止输入线 x 上的上升沿触发（中断和事件）

1：允许输入线 x 上的上升沿触发（中断和事件）

注意：外部唤醒线是边沿触发的，这些线上不能出现毛刺信号。

在写 EXTI_RTSR 寄存器时，在外部中断线上的上升沿信号不能被识别，挂起位也不会被置位。在同一中断线上，可以同时设置上升沿和下降沿触发。即任一边沿都可触发中断。

10.6.4 下降沿触发选择寄存器 1（EXTI_FTSR1）

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
Reserved									FT22	FT21	Reserved			FT17	FT16
									rw	rw				rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								FT7	FT6	FT5	FT4	FT3	FT2	FT1	FT0
								rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:23 保留，必须保持为复位值

Bits 22:21 **FTx：**线 x 上的下降沿触发事件配置（x = 22 至 21）

0：禁止输入线 x 上的下降沿触发（中断和事件）

1：允许输入线 x 上的下降沿触发（中断和事件）

Bits 20:18 保留，必须保持为复位值

Bits 17:16 **FTx：**线 x 上的下降沿触发事件配置（x = 17 至 16）

0：禁止输入线 x 上的下降沿触发（中断和事件）

1：允许输入线 x 上的下降沿触发（中断和事件）

Bits 15:8 保留，必须保持为复位值

Bits 7:0 **FTx：**线 x 上的下降沿触发事件配置（x = 7 至 0）

0：禁止输入线 x 上的下降沿触发（中断和事件）

1：允许输入线 x 上的下降沿触发（中断和事件）

注意：外部唤醒线是边沿触发的，这些线上不能出现毛刺信号。

在写 EXTI_FTSR 寄存器时，在外部中断线上的下降沿信号不能被识别，挂起位也不会被置位。在同一中断线上，可以同时设置上升沿和下降沿触发。即任一边沿都可触发中断。

10.6.5 软件中断事件寄存器 1（EXTI_SWIER1）

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
Reserved									SWI22	SWI21	Reserved			SWI17	SWI16
									rw	rw				rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SWI7	SWI6	SWI5	SWI4	SWI3	SWI2	SWI1	SWI0
								rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:23 保留，必须保持为复位值

Bits 22:21 **SWIx：**线 x 上的软件中断（x = 22 至 21）

如果在此线上的 EXTI_IMR 上使能了中断，

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 中允许产生该中断，则此时将产生一个中断。

此位通过清除 EXTI_PR 的相应位（通过将位写入‘1’）而清除。

Bits 20:18 保留，必须保持为复位值

Bits 17:16 **SWIx：**线 x 上的软件中断（x = 17 至 16）

如果在此线上的 EXTI_IMR 上使能了中断，

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 中允许产生该中断，则此时将产生一个中断。

此位通过清除 EXTI_PR 的相应位（通过将位写入‘1’）而清除。

Bits 15:8 保留，必须保持为复位值

Bits 7:0 **SWIx：**线 x 上的软件中断（x = 7 至 0）

如果在此线上的 EXTI_IMR 上使能了中断，

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 中允许产生该中断，则此时将产生一个中断。

此位通过清除 EXTI_PR 的相应位（通过将位写入‘1’）而清除。

10.6.6 挂起寄存器 1（EXTI_PR1）

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
Reserved									PIF22	PIF21	Reserved			PIF17	PIF16
									rw	rw				rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								PIF7	PIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0
								rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:23 保留，必须保持为复位值

Bits 22:21 **PIFx：**线 x 上的挂起中断标志（x = 22 至 21）

0：没有发生触发请求

1：已发生触发请求

当在外部中断线上发生了选择的边沿事件，该位被置 1。在该位中写入 1 可以清除它。

Bits 20:18 保留，必须保持为复位值

Bits 17:16 **PIFx：**线 x 上的挂起中断标志（x = 17 至 16）

0：没有发生触发请求

1：已发生触发请求

当在外部中断线上发生了选择的边沿事件，该位被置 1。在该位中写入 1 可以清除它。

Bits 15:8 保留，必须保持为复位值

Bits 7:0 **PIFx：**线 x 上的挂起中断标志（x = 7 至 0）

0：没有发生触发请求

1：已发生触发请求

当在外部中断线上发生了选择的边沿事件，该位被置 1。在该位中写入 1 可以清除它。

11 内部参考电压缓冲器（VREFBUF）

11.1 简介

RX32S632 提供了一个内部参考电压 VREFBUF，可作为 ADC 和 OPAMP 的参考电压。

11.2 功能描述

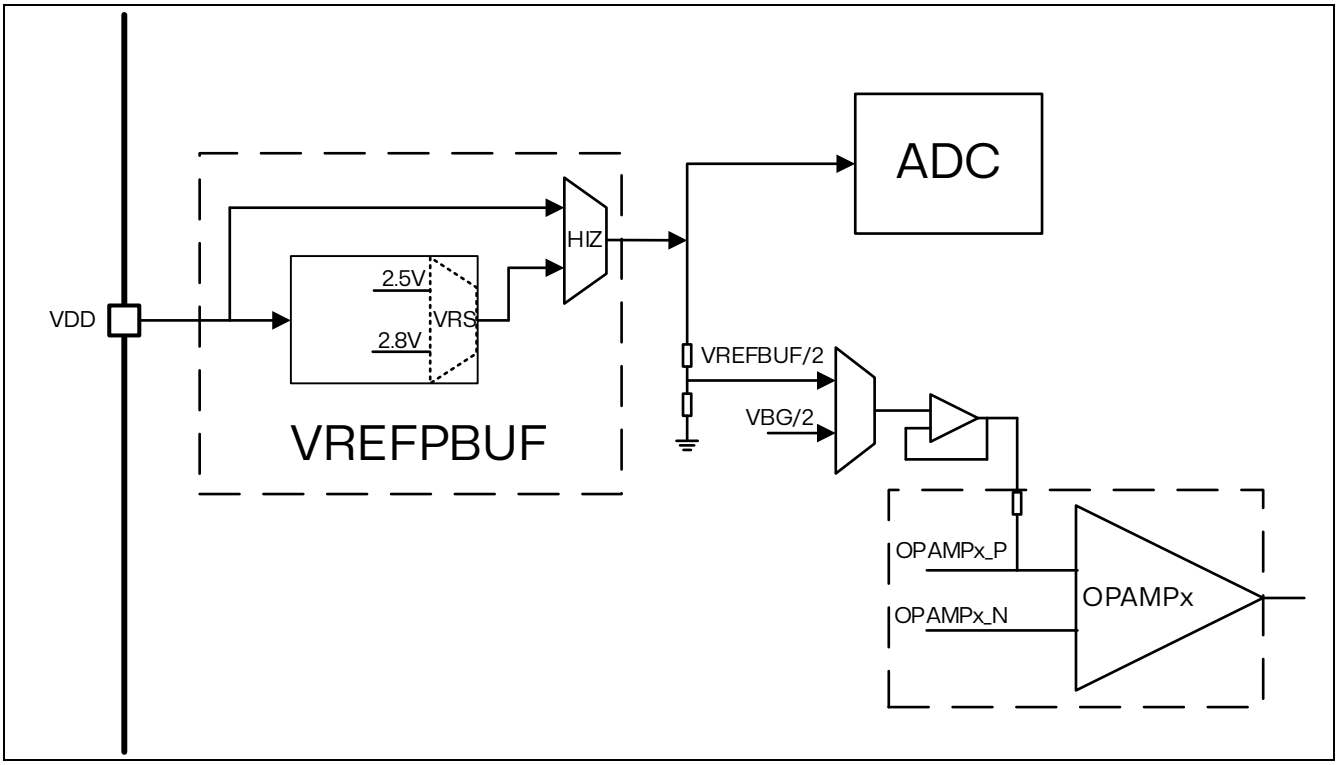


图 11.1 VREFBUF 框图

内部电压参考缓冲区支持 2 个电压
VREFBUF_CSR 寄存器中的 VRS 位（VRS 默认=0）：

- VRS = 0：约 2.5 V。
- VRS = 1：约 2.8 V。

内部电压参考可以根据四种不同的模式配置 ENVR 和 HIZ 位配置。以下表格提供了这些模式：

表 11.1 VREFBUF 模式

模式	ENVR	HIZ	描述
模式 1	0	0	参考电压缓冲区关闭，高阻态
模式 2	0	1	参考电压缓冲区关闭，内部参考电压值为 VDD
模式 3	1	0	参考电压缓冲区开启，内部参考电压值为参考电压缓冲区的输出
模式 4	1	1	同模式一

11.2.1 切换 VREFBUF 电压档位步骤

1. ENVR=0，HIZ=0
2. 更改 VRS 设定
3. ENVR=1，HIZ=0

切换 VREFBUF 档位需要 10us 稳定

11.2.2 内部电压切换到外部电压步骤

1. ENVR=0，HIZ=0
2. ENVR=0，HIZ=1

11.3 VREFBUF 校准

每个电压在出厂都需要校准，将校准值存在对应的自加载，当 VRS 改变就读取对应的校准值。

11.4 VREFBUF 寄存器

访问：无等待，支持字，半字和字节访问

11.4.1 VREFBUF 状态寄存器（CSR）

地址偏移：0x00

复位值：0x0000 0002

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											VRS	VRR	Res	HIZ	ENVR
											rw	r		rw	rw

Bits 31:5 保留，必须保持为复位值

Bit 4 **VRS**：电压基准刻度

这些位选择电压引用缓冲区生成的值。

0：参考电压缓冲区的输出为 2.5V

1：参考电压缓冲区的输出 2.8V

Bit 3 **VRR**：额定电压参考缓冲器

0：电压参考缓冲区输出还没有准备好。

1：电压参考缓冲区输出达到要求的水平。

Bit 2 保留，必须保持为复位值

Bit 1 **HIZ**：高阻抗模式

0：参考电压为参考电压缓冲区的输出。

1：参考电压为 VDD

Bit 0 **ENVR**：参考电压缓冲模式

此比特用于启用电压参考缓冲区模式。

0：内部参考电压模式禁用。

1：内部参考电压模式启用

11.4.2 VREFBUF 控制寄存器（CCR）

地址偏移：0x04

复位值：0x0000 0008

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

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

Bits 31:5 保留，必须保持为复位值

Bits 3:0 TRIM[3:0]：校准值

一个 4 位无符号数据（最小 0000，最大值 1111）

12 模拟数字转换器（ADC）

12.1 简介

12 位 ADC 是一种逐次逼近型模拟数字转换器。各通道的 A/D 转换可以单次、连续、扫描或间断模式执行。ADC 的结果可以左对齐或右对齐方式存储在 16 位数据寄存器中。

模拟看门狗特性允许应用程序检测输入电压是否超出用户定义的高/低阈值。

ADC 的输入时钟不得超过 60.8MHz，它是由 SYSCLK 或 PLL2CLK 产生。

12.2 ADC 主要特征

- 12 位分辨率
- 4Msps 的转换速率
- 快慢通道
- 转换结束、注入转换结束和发生模拟看门狗事件时产生中断
- 单次和连续转换模式
- 从信道 0 到信道 n 的自动扫描模式
- 自校准
- 数据对齐以保持内置数据一致性
- 采样间隔可以按通道分别编程
- 规则转换和注入转换均有外部触发选项
- 间断模式
- ADC 供电要求：2.5V 到 3.6V
- ADC 输入范围：$0 \le V_{IN} \le VREFBUF$

12.3 ADC 功能描述

下图为 ADC1 模块框图。

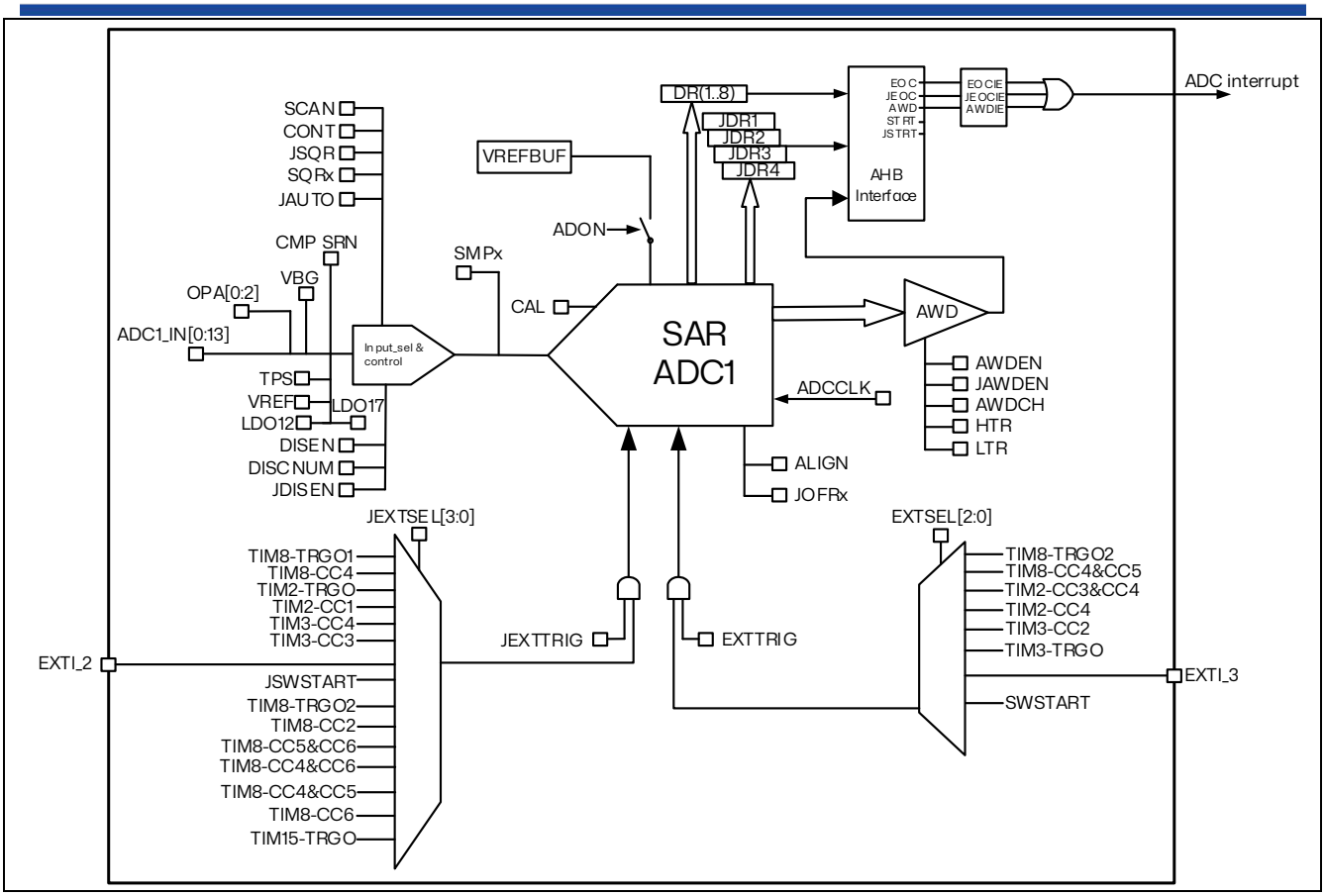


图 12.1 ADC1 框图

12.3.1 ADC 引脚/通道

表 12.1 ADC 引脚

引脚	信号类型	注解
VREFBUF	内部参考电压	ADC 的参考电压，2.5 V/2.8 V/VDD
VDDA	输入，模拟电源	等效于 VDD 的模拟电源，VDDA = VDD
ADCx_IN[m:0]	模拟输入信号	模拟输入通道

表 12.2 ADC1 通道

通道编号	模拟信号来源	IO Type
通道 0	VSSA	FAST_4M
通道 4	PC0	FAST_4M
通道 7	PB5	FAST_4M
通道 8	PB4	FAST_4M
通道 9	PB3	SLOW_2M
通道 10	PB1	SLOW_2M
通道 11	PA7	SLOW_2M
通道 12	PA6	SLOW_2M
通道 14	PD5	SLOW_2M
通道 15	OPAMP1 内部输出	FAST_4M
通道 16	OPAMP2 内部输出	FAST_4M
通道 17	OPAMP3 内部输出	FAST_4M
通道 18	LDO12	SLOW_1M
通道 19(3)	CMP SRN	SLOW_1M
通道 20(1)	VBG	SLOW_1M
通道 21	VREFBUF	SLOW_1M
通道 22(2)	TPS	SLOW_1M
通道 23	LDO17	SLOW_1M

注意：

1. *如果需要采样* VBG *通道，需要配置* ADC_CR2 *寄存器的* TSVREFE *位置*1
2. *如果需要采样* TPS *通道，采样时间要设定为* 640.5T *（选择最高档位）*
3. *如果需要采样* CMP_SRN *通道，需要配置* CMP_CR1 *寄存器的* SRN_EN *位置*1

12.3.2 ADC 开关控制

通过设置 ADC_CR2 寄存器的 ADON 位可给 ADC 上电。当第一次设置 ADON 位时，它将 ADC 从断电状态下唤醒。

ADC 上电延迟一段时间后（t_{STAB}），再次设置 ADON 位时开始进行转换。

通过清除 ADON 位可以停止转换，并将 ADC 置于断电模式。

12.3.3 ADC 时钟

由时钟控制器提供的 ADCCLK 时钟同步 SYSCLK 或者 PLL2CLK，RCC 控制器为 ADC 时钟提供一个可选时钟分频。

12.3.4 通道选择

RX32S632 的 ADC 有多路通道。可以把转换组织成两组：规则组和注入组。在任意多个通道上以任意顺序进行的一系列转换构成成组转换。

- **规则组：**由多达 8 个转换组成。规则通道和它们的转换顺序在 ADC_SQRx 寄存器中选择。规则组中转换的总数应写入 ADC_SQR1 寄存器的 L[2:0]位中。
- **注入组：**由多达 4 个转换组成。注入通道和它们的转换顺序在 ADC_JSQR 寄存器中选择。注入组里的转换总数目应写入 ADC_JSQR 寄存器的 JL[1:0]位中。

如果 ADC_SQRx 或 ADC_JSQR 寄存器在转换期间被更改，当前的转换被清除，一个新的启动脉冲将发送到 ADC 以转换新选择的组。

12.3.5 ADC 部分寄存器的配置步骤

对 ADC_SMPR1，ADC_SMPR2，ADC_SMPR0，ADC_SQR1，ADC_SQR2，ADC_SQR3，ADC_JSQR 寄存器的更改，必须通过置起 ADC_CR3 寄存器的 CFG_UPD 位置 1 使更改生效。

步骤：

1. ADC_CR2 寄存器的 ADON 位清 0；
2. 等待 ADC_ CR3 *寄存器的* PRG_RDY 位置 1；
3. 将新的配置更新到上述寄存器
4. 将 ADC_CR3 寄存器的 CFG_UPD 位置 1
5. 等待 ADC_ CR3 *寄存器的* PRG_RDY 位置 1；
6. ADC_CR2 寄存器的 ADON 位置 1；

12.3.6 单次转换模式

单次转换模式下，ADC 只执行一次转换。该模式既可通过设置 ADC_CR2 寄存器的 ADON 位（只适用于规则通道）启动也可通过外部触发启动（适用于规则通道或注入通道），这时 CONT 位为 0。一旦选择通道的转换完成：

一旦选择通道的转换完成：

- 如果一个规则通道被转换：
  - 转换数据被储存在 16 位 ADC_DR 寄存器中
  - EOC（转换结束）标志被置位
  - 如果设置了 EOCIE 则产生中断
- 如果一个注入通道被转换：
  - 转换数据被储存在 16 位的 ADC_JDR1 寄存器中
  - JEOC（注入转换结束）标志被设置
  - 如果设置了 JEOCIE 位则产生中断

然后 ADC 停止。

*注：*ADC_EOC *寄存器的* OPT *位可选择控制* EOC *（转换结束）标志*
OPT = 0*：规则或注入通道组转换结束时设置* ADC_SR *寄存器的* EOC
OPT = 1*：规则通道组转换结束时设置* ADC_SR *寄存器的* EOC

12.3.7 连续转换模式

在连续转换模式中，当前面 ADC 转换一结束马上就启动另一次转换。此模式可通过外部触发启动或通过设置 ADC_CR2 寄存器上的 ADON 位启动，此时 CONT 位是 1。

每个转换后：

- 如果一个规则通道被转换：
  - 转换数据被储存在 16 位 ADC_DR 寄存器中
  - EOC（转换结束）标志被置位
  - 如果设置了 EOCIE 则产生中断
- 如果一个注入通道被转换：
  - 转换数据被储存在 16 位的 ADC_JDR1 寄存器中
  - JEOC（注入转换结束）标志被设置

– 如果设置了 JEOCIE 位则产生中断

注：ADC_EOC 寄存器的OPT 位可选择控制EOC（转换结束）标志
OPT = 0*：规则或注入通道组转换结束时设置*ADC_SR *寄存器的*EOC
OPT = 1*：规则通道组转换结束时设置*ADC_SR *寄存器的*EOC

12.3.8 时序图

如下图所示，ADC 在开始精确转换前需要一个稳定时间 t_{STAB}。在开始 ADC 转换和 15 个时钟周期后，EOC 标志被设置，16 位 ADC 数据寄存器包含转换的结果。

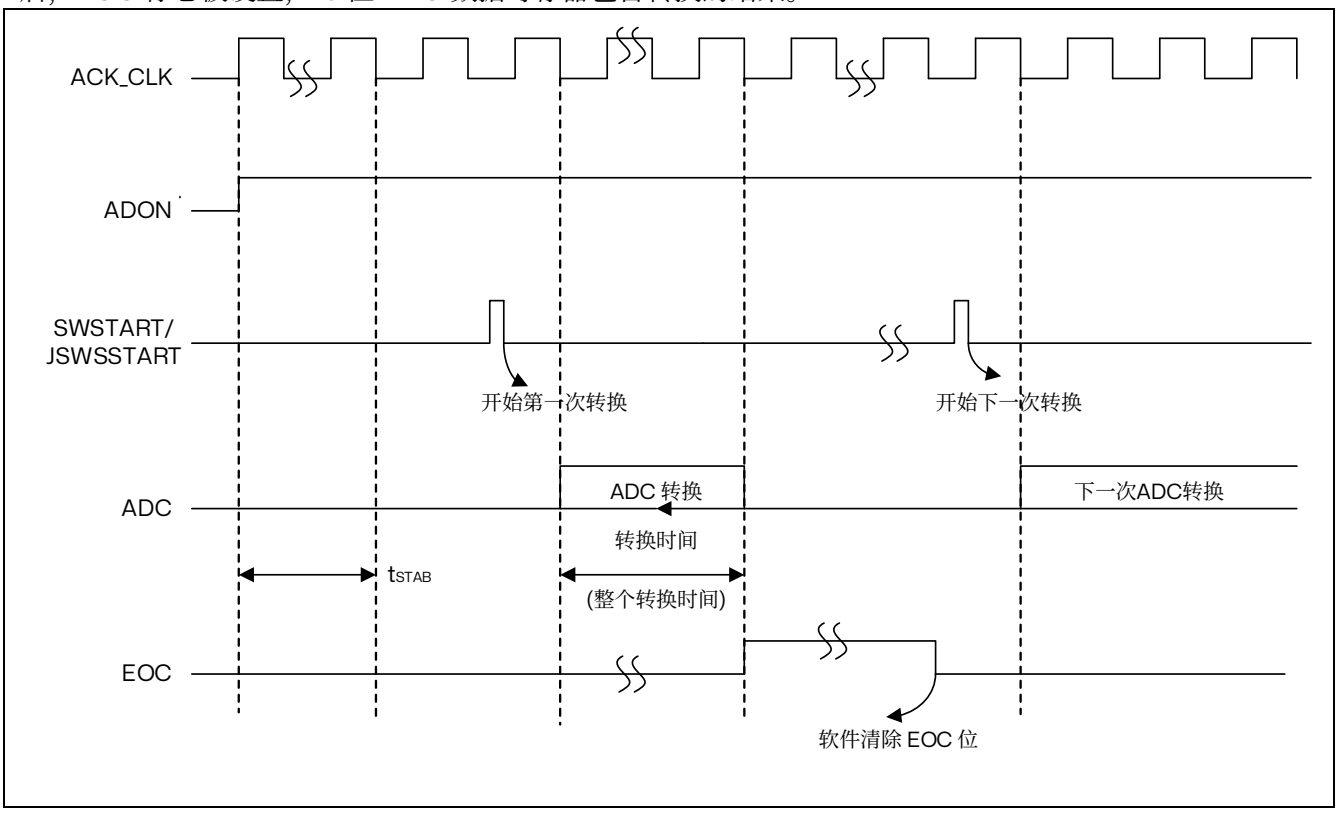


图 12.2 ADC 转换时序图

12.3.9 模拟看门狗

通过在 ADC_CR1 寄存器中设置 AWDEN 位使能 AWD 模拟看门狗。这个看门狗监视一个选定的通道或所有使能的通道是否保持在配置的电压范围内。

如果被 ADC 转换的模拟电压低于阈值下限或高于高阈值上限，AWD 模拟看门狗状态位被置 1。阈值位于 ADC_HTR 和 ADC_LTR 寄存器的 12 个最低有效位中。通过设置 ADC_CR1 寄存器的 AWDIE 位以允许产生相应中断。

阈值独立于由 ADC_CR2 寄存器上的 ALIGN 位选择的数据对齐模式。比较是在对齐之前完成的。

通过配置 ADC_CR1 寄存器的 AWDSGL 位，模拟看门狗可以作用于 1 个或多个通道。如果设置模拟看门狗作用于单一通道，可以通过 ADC_HTR 和 ADC_LTR 寄存器中的 HTRCH 和 LTRCH 来选择作用于哪一个通道。

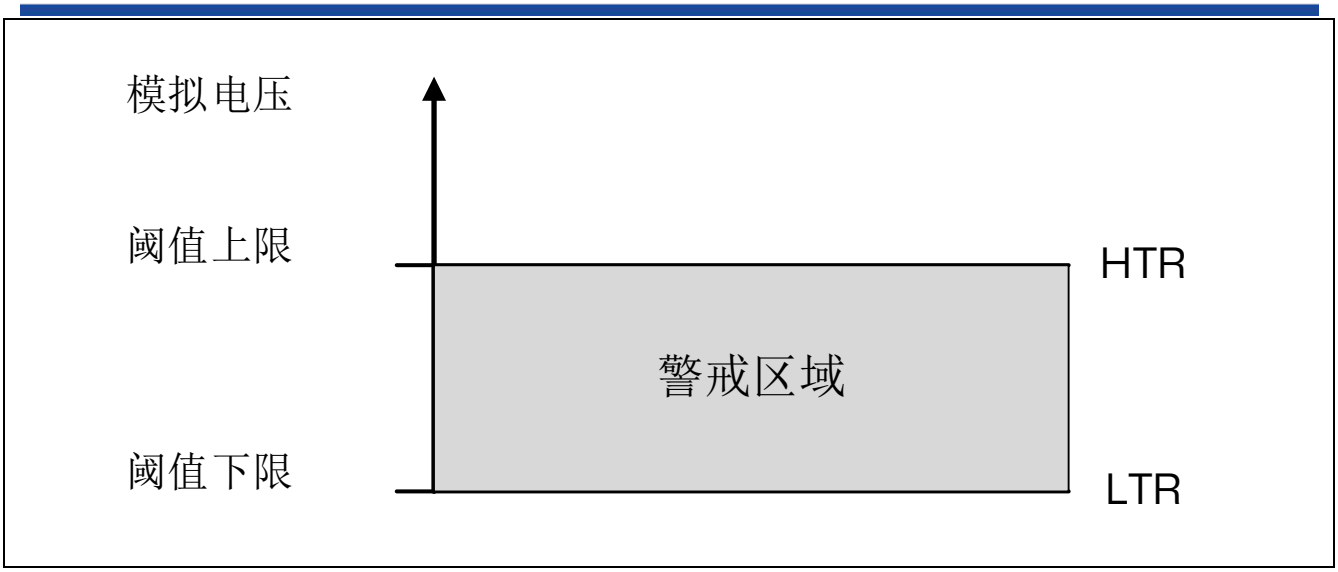


图 12.3 模拟看门狗警戒区

表 12.3 模拟看门狗通道选择

模拟看门狗警戒通道	ADC_CR1 寄存器控制位		
	AWDSGL 位	AWDEN 位	JAWDEN 位
无	任意值	0	0
所有注入通道	0	0	1
所有规则通道	0	1	0
所有注入和规则通道	0	1	1
单一的 (1) 注入通道	1	0	1
单一的 (1) 规则通道	1	1	0
单一的 (1) 注入或规则通道	1	1	1

1. 由 AWDCH[4:0]位选择

模拟看门狗滤波器

当 ADC 配置为只有一个输入通道时(不允许在扫描模式下选择多个通道)，可以通过 ADC_CR1 寄存器配置有效的 ADC 转换数据间隔:

如果数据超出范围数倍于 ADC_CR1 的 AWDFILT 位中指定的值，则设置 AWDx 标志并发出相应的中断。

模拟看门狗阈值控制 0

LTR[11:0]和 HTR[11:0]可以在模数转换进行时（即在 ADC 内部状态的转换开始和转换结束之间）改变。

如果在 ADC 保护通道的 ADC 转换过程中更新 LTR[11:0]和 HTR[11:0]，则看门狗功能将被屏蔽。这种屏蔽将在下一个转换开始时被移除，从而导致从下一个 ADC 转换应用模拟看门狗阈值。在转换的每一端执行模拟看门狗比较。如果当前 ADC 数据超出新间隔，则不发出中断和 AWDx 信号。中断和 AWDx 生成只发生在阈值更新后开始的转换结束时。如果已经发出了 AWDx 信号，编程新的阈值不会取消 AWDx 信号的发出。

12.3.10 扫描模式

此模式用来扫描一组模拟通道。

扫描模式可通过设置 ADC_CR1 寄存器的 SCAN 位来选择。一旦这个位被设置，ADC 扫描所有被 ADC_SQRx 寄存器（对规则通道）或 ADC_JSQR（对注入通道）选中的所有通道。在每个组的每个通道上执行单次转换。在每个转换结束时，同一组的下一个通道被自动转换。如果设置了 CONT 位，转换不会在选择组的最后一个通道上停止，而是再次从选择组的第一个通道继续转换。而注入通道转换的数据总是存储在 ADC_JDRx 寄存器中。

12.3.11 注入通道管理

触发注入

清除 ADC_CR1 寄存器的 JAUTO 位，并且设置 SCAN 位，即可使用触发注入功能。

1. 利用外部触发或通过设置 ADC_CR2 寄存器的 ADON 位，启动一组规则通道的转换。
2. 如果在规则通道转换期间产生一外部注入触发，当前转换被复位，注入通道序列被以单次扫描方式进行转换。
3. 然后，恢复上次被中断的规则组通道转换。如果在注入转换期间产生一规则事件，注入转换不会被中断，但是规则序列将在注入序列结束后被执行。

注：当使用触发的注入转换时，必须保证触发事件的间隔长于注入序列。例如：序列长度为 30 个 ADC 时钟周期（即 2 个具有 2.5 个时钟间隔采样时间的转换），触发之间最小的间隔必须是 31 个 ADC 时钟周期。

自动注入

如果设置了 JAUTO 位，在规则组通道之后，注入组通道被自动转换。这可以用来转换在 ADC_SQRx 和 ADC_JSQR 寄存器中设置的多至 12 个转换序列。

在此模式里，必须禁止注入通道的外部触发。

如果除 JAUTO 位外还设置了 CONT 位，规则通道至注入通道的转换序列被连续执行。

对于 ADC 时钟预分频系数为 4 至 8 时，当从规则转换切换到注入序列或从注入转换切换到规则序列时，会自动插入 1 个 ADC 时钟间隔；当 ADC 时钟预分频系数为 2 时，则有 2 个 ADC 时钟间隔的延迟。

注：不可能同时使用自动注入和间断模式。

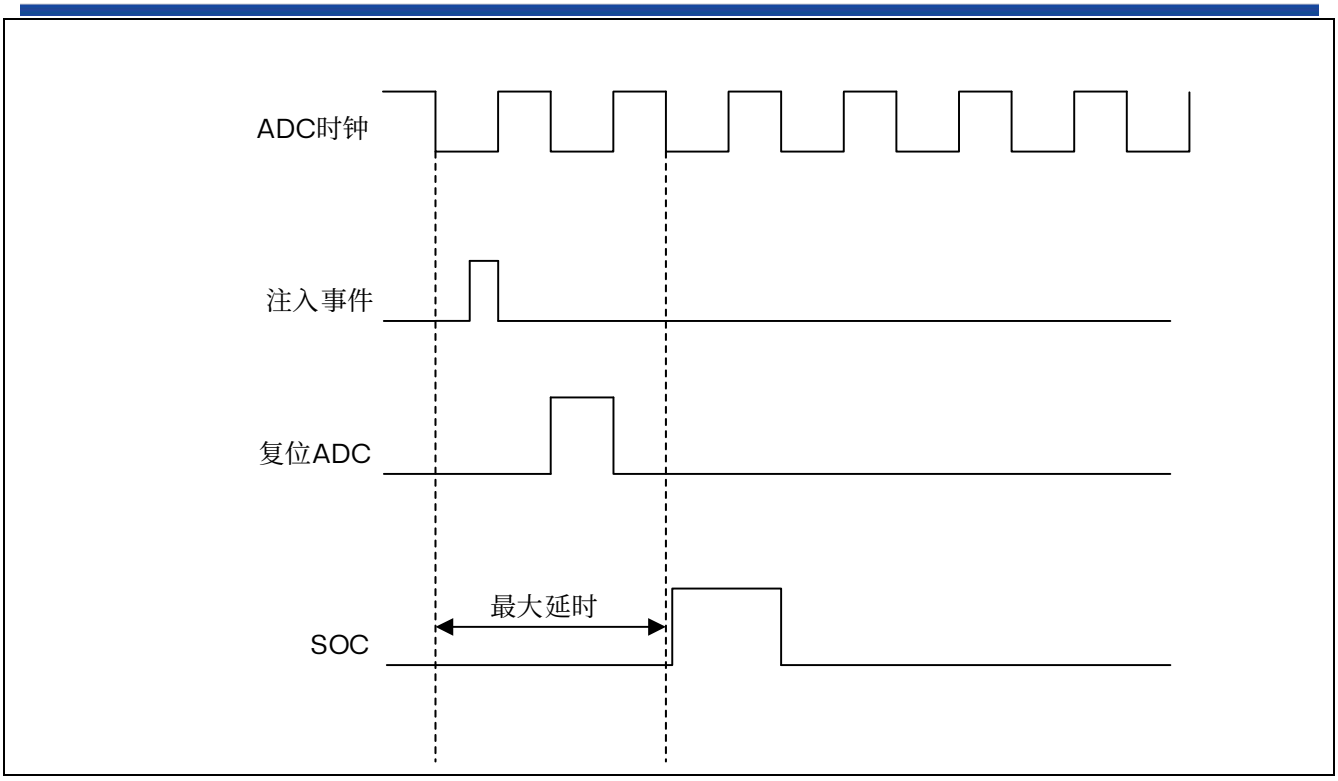


图 12.4 注入转换延迟

12.3.12 间断模式

规则组

此模式通过设置 ADC_CR1 寄存器上的 DISCEN 位激活。它可以用来执行一个短序列的 n 次转换（n<=8），此转换是 ADC_SQRx 寄存器所选择的转换序列的一部分。数值 n 由 ADC_CR1 寄存器的 DISCNUM[2:0]位给出。

一个外部触发信号可以启动 ADC_SQRx 寄存器中描述的下一轮 n 次转换，直到此序列所有的转换完成为止。总的序列长度由 ADC_SQR1 寄存器的 L[2:0]定义。

举例：

n = 3，被转换通道 = 0、1、2、3、6、7、9、10

第一次触发：转换的序列为 0、1、2

第二次触发：转换的序列为 3、6、7

第三次触发：转换的序列为 9、10，并产生 EOC 事件

第四次触发：转换的序列为 0、1、2

注意：当以间断模式转换一个规则组时，转换序列结束后不自动从头开始。

当所有子组被转换完成，下一次触发启动第一个子组的转换。在上面的例子中，第四次触发重新转换第一子组的通道 0、1 和 2。

注意：重启间断模式可以重置规则组间断计数

注入组

此模式通过设置 ADC_CR1 寄存器的 JDISCEN 位激活。在一个外部触发事件后，该模式按通道顺序逐个转换 ADC_JSQR 寄存器中选择的序列。

一个外部触发信号可以启动 ADC_JSQR 寄存器选择的下一个通道序列的转换，直到序列中所有的转换完成为止。总的序列长度由 ADC_JSQR 寄存器的 JL[1:0]位定义。

例子：

n=1，被转换的通道 = 1、2、3

第一次触发：通道 1 被转换

第二次触发：通道 2 被转换

第三次触发：通道 3 被转换，并且产生 EOC 和 JEOC 事件

第四次触发：通道 1 被转换

注意：当完成所有注入通道转换，下个触发启动第1 个注入通道的转换。在上述例子中，第四个触发重新转换第1 个注入通道1。

不能同时使用自动注入和间断模式。

注意：重启间断模式可以重置注入组间断计数

12.4 校准

ADC 有一个内置自校准模式。校准可大幅减小因内部电容器组的变化而造成的准精度误差。在校准期间，在每个电容器上都会计算出一个误差修正码（数字值），这个码用于消除在随后的转换中每个电容器上产生的误差。

通过设置 ADC_CR2 寄存器的 CAL 位启动校准。一旦校准结束，CAL 位被硬件复位，可以开始正常转换。建议在上电时执行一次 ADC 校准。校准阶段结束后，校准码储存在 ADC_CAL 中。

注意：建议在每次上电后执行一次校准。

启动校准前，ADC 必须处于关电状态（ADON='0'）超过至少两个 ADC 时钟周期。

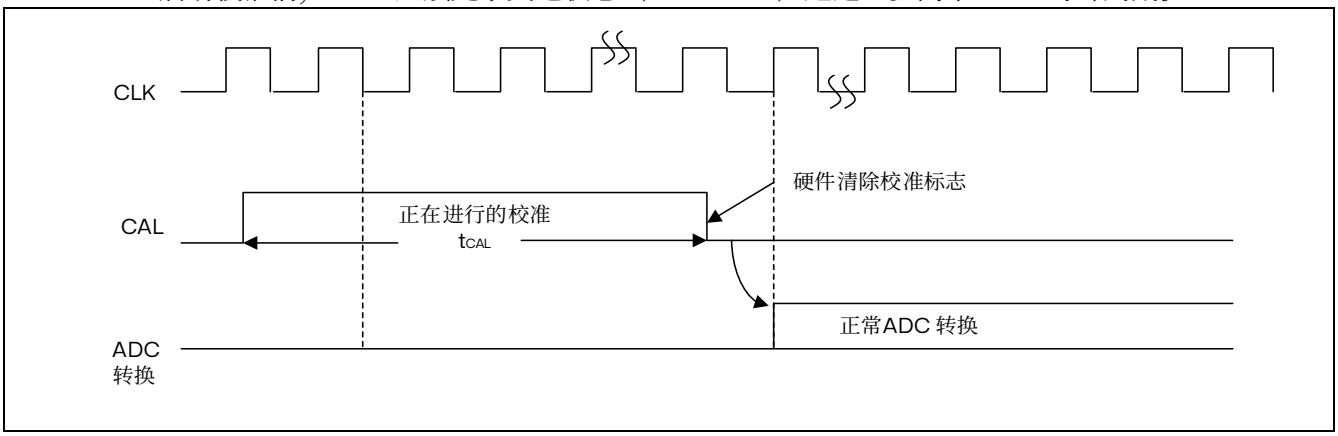


图 12.5 校准时序图

12.5 数据对齐

ADC_CR2 寄存器中的 ALIGN 位选择转换后数据储存的对齐方式。数据可以左对齐或右对齐。

注入组通道转换的数据值已经减去了在 ADC_JOFRx 寄存器中定义的偏移量，因此结果可以是一个负值。SEXT 位是扩展的符号值。

对于规则组通道，不需减去偏移值，因此只有 12 个位有效。

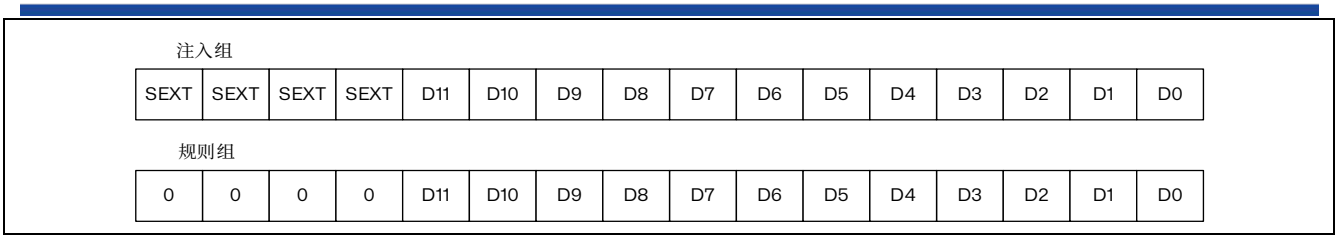


图 12.6 数据右对齐

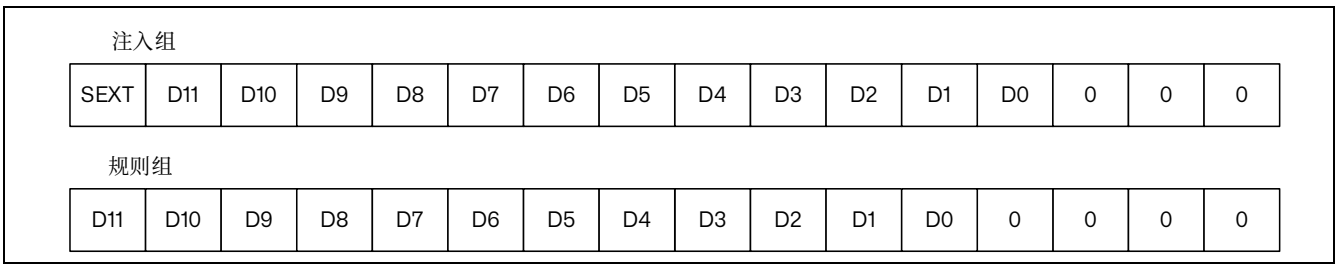


图 12.7 数据左对齐

12.6 可编程的通道采样时间

ADC 使用若干个 ADC_CLK 周期对输入电压采样，采样周期数目可以通过 ADC_SMPR 的 SMP[2:0]位更改。每个通道可以分别用不同的时间采样。

总转换时间如下计算：

T_{CONV} = 采样时间 + 12.5 个周期

例如：

当 ADCCLK = 60MHz，采样时间为 2.5 周期

T_{CONV} = 2.5 + 12.5 = 15 个周期 = 250ns

12.7 外部触发转换

转换可以由外部事件触发（例如定时器捕获，EXTI 线）。如果设置了 EXTTRIG 控制位，则外部事件就能够触发转换。EXTSEL[2:0]和 JEXTSEL[3:0]控制位允许应用程序选择所有可能的事件中的某一个，可以触发规则和注入组的采样。

注意：当外部触发信号被选为 ADC 规则或注入转换时，只有它的上升沿可以启动转换。

表 12.4 规则通道的外部触发信号

触发源	类型	EXTSEL[2:0]
定时器 8 的 TRGO2 事件	片上定时器内部信号	000
定时器 8 的 OC4&OC5 事件		001
定时器 2 的 OC3&OC4 事件		010
定时器 2 的 OC4 事件		011
定时器 3 的 OC2 事件		100
定时器 3 的 TRGO 事件		101
EXTI 线 3	外部引脚信号	110
SWSTART	软件控制位	111

表 12.5 注入通道的外部触发信号

触发源	类型	JEXTSEL[3:0]
定时器 8 的 TRGO1 事件	片上定时器内部信号	0000
定时器 8 的 OC4 事件		0001
定时器 2 的 TRGO 事件		0010
定时器 2 的 OC1 事件		0011
定时器 3 的 OC4 事件		0100
定时器 3 的 OC3 事件		0101
EXTI 线 2	外部引脚信号	0110
JSWSTART	软件控制位	0111
定时器 8 的 TRGO2 事件	片上定时器内部信号	1000
定时器 8 的 OC2 事件		1001
定时器 8 的 OC5&OC6 事件		1010
定时器 8 的 OC4&OC6 事件		1011
定时器 8 的 OC4&OC5 事件		1100
定时器 8 的 OC6 事件		1101
定时器 15 的 TRGO 事件		1110

软件触发事件可以通过对寄存器 ADC_CR2 的 SWSTART 或 JSWSTART 位置 1 产生。

规则组的转换可以被注入触发打断。

12.8 温度传感器（TPS）

温度传感器可以用来测量器件周围的温度。

温度传感器在内部和 ADC 的通道 22 相连，此通道把传感器输出的电压转换成数字值。

温度传感器输出电压随温度线性变化，由于生产过程的变化，温度变化曲线的偏移在不同芯片上会有不同（最多相差 45°C）。

注意：必须设置 TSVREFE 位激活内部通道：ADC1_IN22（温度传感器）和 ADC1_IN20（VBG）的转换。

内部温度传感器更适合于检测温度的变化，而不是测量绝对的温度。如果需要测量精确的温度，应该使用一个外置的温度传感器。

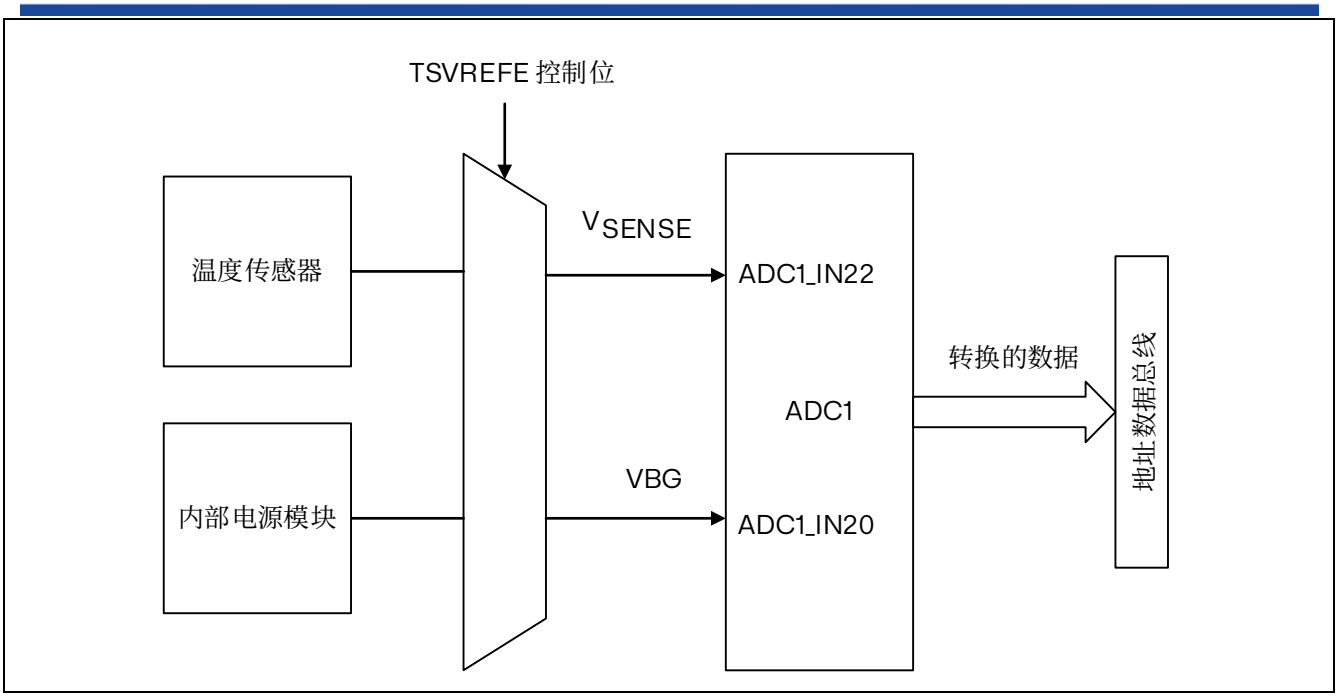


图 12.8 温度传感器和 VBG 通道框图

12.9 ADC 中断

规则和注入组转换结束时能产生中断，当模拟看门狗状态位被设置时也能产生中断。它们都有独立的中断使能位。

ADC_SR 寄存器中有 2 个其他标志，但是它们没有相关联的中断：

- JSTRT（注入组通道转换的启动）
- STRT（规则组通道转换的启动）

表 12.6 ADC 中断

中断事件	事件标志	使能控制位
规则组转换结束	EQC	EOCIE
注入组转换结束	JEOC	JEOCIE
设置了模拟看门狗状态位	AWD	AWDIE

12.10 ADC 寄存器

访问：无等待，支持字，半字和字节访问

12.10.1 ADC 状态寄存器（ADC_SR）

地址偏移：0x00

复位值：0x0020

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											STRT	JSTRT	JEOC	EOC	AWD
											rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

Bits 31:5 保留，必须保持为复位值

Bit 4 **STRT**：规则通道开始位

该位由硬件在规则通道转换开始时设置，由软件清除。

0：规则通道转换未开始

1：规则通道转换已开始

Bit 3 **JSTRT**：注入通道开始位

该位由硬件在注入通道转换开始时设置，由软件清除。

0：注入通道转换未开始

1：注入通道转换已开始

Bit 2 **JEOC**：注入通道转换结束位

该位由硬件在注入通道组转换结束时设置，由软件清除。

0：转换未完成

1：转换完成

Bit 1 **EOC**：转换结束位

该位由硬件在(规则或注入)通道组转换结束时设置，由软件清除。

0：转换未完成

1：转换完成

Bit 0 **AWD**：模拟看门狗标志位

该位由硬件在转换的电压值超出了 ADC_LTR 和 ADC_HTR 寄存器定义的范围时设置，由软件清除。

0：没有发生模拟看门狗事件

1：发生模拟看门狗事件

12.10.2 ADC 控制寄存器 1（ADC_CR1）

地址偏移：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					AWDFILT[2:0]			AWD EN	JAWD EN	Reserved					
					rc_w0	rc_w0	rc_w0	rw	rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISCNUM[2:0]			JDISC EN	DISC EN	JAUTO	AWD SGL	SCAN	JEOC IE	AWDIE	EOCIE	AWDCH[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:27 保留，必保持为复位值。

Bits 26:24 **AWDFILT[2:0]**：模拟监视滤波参数

当 AWDEN &JWADEN 为 0 时，或监测通道变化，AWDFILT counter clear

000：无作用

001：两个连续的检测产生一个 AWDx 信号或一个中断

...

111：连续 8 个检测产生一个 AWDx 信号或一个中断

注：AWDFILT 使用上只会转一个 channel AWDFILT 使用上不考虑 REG&INJ 交错

Bit 23 **AWDEN**：在规则通道上开启模拟看门狗

该位由软件设置和清除。

0：在规则通道上禁用模拟看门狗

1：在规则通道上使能模拟看门狗

Bit 22 **JAWDEN**：在注入通道上开启模拟看门狗

该位由软件设置和清除。

0：在注入通道上禁用模拟看门狗

1：在注入通道上使能模拟看门狗

Bits 21:16 保留，必须保持为复位值。

Bits 15:13 **DISCNUM[2:0]**：间断模式通道计数

软件通过这些位定义在间断模式下，收到外部触发后转换规则通道的数目。(间断长度只应用于规则组，注入组使能间断模式后，不需要设置间断长度，间断长度固定为 1 个通道)

000：1 个通道

001：2 个通道

.....

111：8 个通道

Bit 12 **JDISCEN**：注入通道上的间断模式

该位由软件设置和清除，用于开启或关闭注入通道组上的间断模式。

0：注入通道组上禁用间断模式

1：注入通道组上使能间断模式

Bit 11 **DISCEN**：规则通道上的间断模式

该位由软件设置和清除，用于开启或关闭规则通道组上的间断模式。

0：规则通道组上禁用间断模式

1：规则通道组上使能间断模式

Bit 10 **JAUTO**：自动的注入通道组转换

该位由软件设置和清除，用于开启或关闭规则通道组转换结束后自动的注入通道组转换。

0：关闭自动的注入通道组转换

1：使能自动的注入通道组转换

Bit 9 **AWDSGL**：扫描模式中在一个单一的通道上使用看门狗

该位由软件设置和清除，用于开启或关闭由 AWDCH[4:0]位指定的通道上的模拟看门狗功能

0：在所有的通道上使能模拟看门狗

1：在单一通道上使能模拟看门狗

Bit 8 **SCAN**：扫描模式

该位由软件设置和清除，用于开启或关闭扫描模式。在扫描模式中，转换由 ADC_SQRx 或 ADC_JSQRx 寄存器选中的通道。

0：禁止扫描模式

1：使能扫描模式

注：如果分别设置了 EOCIE 或 JEOCIE 位，只在最后一个通道转换完毕后才会产生 EOC 或 JEOC 中断。

Bit 7 **JEOCIE**：允许产生注入通道转换结束中断

该位由软件设置和清除，用于禁止或允许所有注入通道转换结束后产生中断。

0：禁止 JEOC 中断

1：允许 JEOC 中断。当硬件设置 JEOC 位时产生中断。

Bit 6 **AWDIE：**允许产生模拟看门狗中断

该位由软件设置和清除，用于禁止或允许模拟看门狗产生中断。在扫描模式下，如果看门狗检测到超范围的数值时，只有在设置了该位时扫描才会中止。

0：禁止模拟看门狗中断

1：允许模拟看门狗中断

Bit 5 **EOCIE：**允许产生 EOC 中断

该位由软件设置和清除，用于禁止或允许转换结束后产生中断。

0：禁止 EOC 中断

1：允许 EOC 中断。当硬件设置 EOC 位时产生中断。

Bits 4:0 **AWDCH[4:0]：**模拟看门狗通道选择位

这些位由软件设置和清除，用于选择模拟看门狗保护的输入通道。

00000：ADC 模拟输入通道 0

00001：ADC 模拟输入通道 1

.....

10111：ADC 模拟输入通道 22

12.10.3 ADC 控制寄存器 2（ADC_CR2）

地址偏移：0x08

复位值：0x0100 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								TS VREFE	SW START	JSW START	EXT TRIG	EXTSEL[2:0]			Res
								rw	rw	rw	rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JEXT TRIG	JEXTSEL[3:0]				ALIGN	Reserved						RST CAL	CAL	CONT	ADON
rw	rw	rw	rw	rw	rw							rw	rw	rw	rw

Bits 31:24 保留，必须保持为复位值。

Bit 23 **TSVREFE：**温度传感器和 VBG 使能

该位由软件设置和清除，用于使能或禁止温度传感器和 VBG 通道。

0：禁止温度传感器和 VBG

1：使能温度传感器和 VBG

Bit 22 **SWSTART：**开始转换规则通道

由软件设置该位以启动转换，转换开始后硬件马上清除此位。如果在 EXTSEL[2:0]位中选择了 SWSTART 为触发事件，该位用于启动一组规则通道的转换。

0：复位状态

1：开始转换规则通道

Bit 21 **JSWSTART：**开始转换注入通道

由软件设置该位以启动转换，转换开始后硬件马上清除此位。如果在 JEXTSEL[2:0]位中选择了 JSWSTART 为触发事件，该位用于启动一组注入通道的转换。

0：复位状态

1：开始转换注入通道

Bit 20 **EXTTRIG：**规则通道的外部触发转换模式

该位由软件设置和清除，用于开启或禁止可以启动规则通道组转换的外部触发事件。

0：禁止外部事件启动转换

1：使能外部事件启动转换

Bits 19:16 **EXTSEL[2:0]：**选择启动规则通道组转换的外部事件

这些位选择用于启动规则通道组转换的外部事件。

000：定时器 8 的 TRGO2 事件

001：定时器 8 的 OC4&OC5 事件

010：定时器 2 的 OC3&OC4 事件

011：定时器 2 的 OC4 事件

100：定时器 3 的 OC2 事件

101：定时器 3 的 TRGO 事件

110：EXTI 线 3

111：SWSTART

Bit 15 **JEXTTRIG：**注入通道的外部触发转换模式

该位由软件设置和清除，用于开启或禁止可以启动注入通道组转换的外部触发事件。

0：禁止外部事件启动转换

1：使能外部事件启动转换

Bits 14:11 **JEXTSEL[3:0]：**选择启动注入通道组转换的外部事件

这些位选择用于启动注入通道组转换的外部事件。

0000：定时器 8 的 TRGO1 事件

0001：定时器 8 的 OC4 事件

0010：定时器 2 的 TRGO 事件

0011：定时器 2 的 OC1 事件

0100：定时器 3 的 OC4 事件

0101：定时器 3 的 OC3 事件

0110：EXTI 线 2

0111：JSWSTART

1000：定时器 8 的 TRGO2 事件

1001：定时器 8 的 OC2 事件

1010：定时器 8 的 OC5&OC6 事件

1011：定时器 8 的 OC4&OC6 事件

1100：定时器 8 的 OC4&OC5 事件

1101：定时器 8 的 OC6 事件

1110：定时器 15 的 TRGO 事件

Bit 10 **ALIGN：**数据对齐

该位由软件设置和清除。

0：右对齐

1：左对齐

Bits 9:4 保留，必须保持为复位值。

Bit 3 **RSTCAL：**复位校准

该位由软件设置并由硬件清除。在校准寄存器被初始化后该位将被清除。

0：校准寄存器已初始化

1：初始化校准寄存器

注：如果正在进行转换时设置 RSTCAL，*清除校准寄存器需要额外的周期。*

Bit 2 **CAL：**A/D 校准

该位由软件设置以开始校准，并在校准结束时由硬件清除。

0：校准完成

1：开始校准

Bit 1 **CONT：**连续转换

该位由软件设置和清除。如果设置了此位，则规则组转换将连续进行直到该位被清除。

0：单次转换模式

1：连续转换模式

Bit 0 **ADON：**开/关 A/D 转换器

该位由软件设置和清除。当该位为 0 时，写入 1 将把 ADC 从断电模式下唤醒。当该位为 1 时，写入 1 将启动转换。

0：关闭 ADC 转换/校准，并进入断电模式

1：开启 ADC 并启动转换

注：如果在这个寄存器中与 ADON *一起还有其他位被改变，则转换不被触发。这是为了防止触发错误的转换。*

*注：*ADC *工作过程中，若需要重新配置* ADC *的某些寄存器（即* ADC register *会动态*

*变化），在配置之前需要先关闭*CR2_ADON，*配置完成后再开启*CR2_ADON。

12.10.4 ADC 采样时间寄存器（ADC_SMPR1）

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.		SMP19[2:0]			SMP18[2:0]			SMP17[2:0]			SMP16[2:0]			SMP15[2:1]	
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMP 15_0	SMP14[2:0]			SMP13[2:0]			SMP12[2:0]			SMP11[2:0]			SMP10[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit 31:30 保留，必须保持为复位值。

Bit 29:0 **SMPx[2:0]：**通道 CH10-19 的采样时间（x=19 至 10）

这些位用于独立地选择每个通道的采样时间。在采样周期中通道选择位必须保持不变。

000：2.5 周期

001：6.5 周期

010：11.5 周期

011：17.5 周期

100：25.5 周期

101：35.5 周期

110：48.5 周期

111：640.5 周期

12.10.5 ADC 采样时间寄存器（ADC_SMPR2）

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.		SMP9[2:0]			SMP8[2:0]			SMP7[2:0]			SMP6[2:0]			SMP5[2:1]	
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMP 5_0	SMP4[2:0]			SMP3[2:0]			SMP2[2:0]			SMP1[2:0]			SMP0[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:30 保留，必须保持为复位值。

Bits 29:0 **SMPx[2:0]：**通道 CH0-9 的采样时间（x=9 至 0）

这些位用于独立地选择每个通道的采样时间。在采样周期中通道选择位必须保持不变。

000：2.5 周期

001：6.5 周期

010：11.5 周期

011：17.5 周期

100：25.5 周期

101：35.5 周期

110：48.5 周期

111：640.5 周期

12.10.6 ADC 注入通道数据偏移寄存器 x（ADC_JOFRx）（x=1..4）

地址偏移：0x14-0x20

复位值：0x0000 0000

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

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

Bits 31:12 保留，必须保持为复位值。

Bits 11:0 **JOFFSETx[11:0]：**注入通道 x 的数据偏移

当转换注入通道时，这些位定义了用于从原始转换数据中减去的数值。转换的结果可以在 ADC_JDRx 寄存器中读出。

12.10.7 ADC 看门狗高阈值寄存器（ADC_HTR）

地址偏移：0x24

复位值：0x0000 0FFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HTREN	Reserved										HTRCH[5:0]				
rw											rw	rw	rw	rw	rw

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

Bit 31 **HTREN：**该位始能后，ADC 在转换 HTRCH 通道后，会将 ADC 转换值读取到 HTR 寄存器

0：禁止

1：使能

Bits 30:21 保留，必须保持为复位值。

Bits 20:16 **HTRCH[5:0]：**HTR 通道选择位

00000：CH0

...

10110：CH22

Bits 15:12 保留，必须保持为复位值。

Bits 11:0 **HTR[11:0]：**模拟看门狗高阈值

这些位定义了模拟看门狗的阈值高限。

12.10.8 ADC 看门狗低阈值寄存器（ADC_LTR）

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
LTREN	Reserved										LTRCH[5:0]				
rw											rw	rw	rw	rw	rw

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

Bit 31 **LTREN**：该位始能后，ADC 在转换 LTRCH 通道后，会将 ADC 转换值读取到 LTR 寄存器

0：禁止

1：使能

Bits 30:21 保留，必须保持为复位值。

Bits 20:16 **LTRCH[5:0]**：LTR 通道选择位

00000：CH0

...

10110：CH22

Bits 15:12 保留，必须保持为复位值。

Bits 11:0 **LTR[11:0]**：模拟看门狗低阈值

这些位定义了模拟看门狗的阈值低限。

12.10.9 ADC 规则序列寄存器 1（ADC_SQR1）

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
Reserved									L[2:0]			Reserved			
									rw	rw	rw				

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

Bits 31:23 保留，必须保持为复位值。

Bits 22:20 **L[2:0]**：规则通道序列长度

这些位由软件定义在规则通道转换序列中的通道数目。

000：1 个转换

001：2 个转换

.....

111：8 个转换

Bits 19:0 保留，必须保持为复位值。

12.10.10 ADC 规则序列寄存器 2（ADC_SQR2）

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
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						SQ8[4:0]					SQ7[4:0]				
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:10 保留，必须保持为复位值。

Bits 9:0 **SQx[4:0]**：规则序列中的第 x 个转换（x=8 至 7）

这些位由软件定义转换序列中的第 x 个转换通道的编号

12.10.11 ADC 规则序列寄存器 3（ADC_SQR3）

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
Reserved		SQ6[4:0]					SQ5[4:0]					SQ4[4:1]			
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ4_0	SQ3[4:0]					SQ2[4:0]					SQ1[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:30 保留，必须保持为复位值。

Bits 29:0 **SQx[4:0]：**规则序列中的第 x 个转换（x=6 至 1）

这些位由软件定义转换序列中的第 x 个转换通道的编号

12.10.12 ADC 注入序列寄存器（ADC_JSQR）

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
Reserved										JL[1:0]		JSQ4[4:1]			
										rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JSQ4_0	JSQ3[4:0]					JSQ2[4:0]					JSQ1[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:22 保留，必须保持为复位值。

Bits 21:20 **JL[1:0]：**注入通道序列长度（Injected sequence length）

这些位由软件定义在规则通道转换序列中的通道数目。

00：1 个转换

01：2 个转换

10：3 个转换

11：4 个转换

Bits 19:0 **JSQx[4:0]：**注入序列中的第 x 个转换（x=4 至 1）

这些位由软件定义转换序列中的第 x 个转换通道的编号

*注：不同于规则转换序列，如果 JL[1:0]的长度小于 4，则转换的序列顺序是从（4-JL）开始。例如：*ADC_JSQR[21:0] = 10 00011 00011 00111 00010*，意味着扫描转换将按下列通道顺序转换：*7、3、3*，而不是*2、7、3。

12.10.13 ADC 注入数据寄存器（ADC_JDRx（x=1...4））

地址偏移：0x3C-0x48

复位值：0x0000 0000

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

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

Bits 31:16 保留，必须保持为复位值。

Bits 15:0 **JDATAx[15:0]：**注入转换的数据

这些位为只读，包含了注入通道的转换结果。

12.10.14 ADC DR 数据寄存器（ADC_DR）

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
Reserved															

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

Bits 31:16 保留，必须保持为复位值。

Bits 15:0 **DATA[15:0]：**FIFO MODE 下的规则转换的数据（Regular data）

这些位为只读，包含了规则通道的转换结果。数据是左对齐或右对齐

12.10.15 ADC 控制寄存器（ADC_CxCR）

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
Reserved															BASW
															rw

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

Bits 31:17 保留，必须保持为复位值。

Bit 16 **BASW：**遮蔽 ADC 采样宽度

0：遮蔽宽度为 ADC 采样周期

1：遮蔽宽度为 ADC 采样周期+12.5T

Bits 15:0 保留，必须保持为复位值。

12.10.16 ADC 校准寄存器（ADC_CAL）

地址偏移：0x54

复位值：0x0040

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

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

Bits 31:7 保留，必须保持为复位值。

Bits 6:0 **DATA[6:0]：**ADC 自校准的值写入该寄存器

可以手动写入值，在 ADC 转换中生效

12.10.17 ADC EOC 控制寄存器（ADC_EOC）

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
Reserved															

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

Bits 31:6 保留，必须保持为复位值。

Bit 5 **OPT**：转换结束标志位 EOC 控制位

0：规则或注入通道组转换结束时设置 ADC_SR 寄存器的 EOC

1：规则通道组转换结束时设置 ADC_SR 寄存器的 EOC

Bits 4:0 保留，必须保持为复位值。

12.10.18 ADC TPS 校准寄存器（TPS）

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
Reserved															

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

Bits 31:8 保留，必须保持为复位值。

Bits 7:0 **OFFSET[7:0]**：TPS 的校准值写入该寄存器

可以手动写入值，也可以自加载

转换 TPS 时生效，转换值+OFFSET

Bit 7 为符号位

12.10.19 ADC 采样时间寄存器（ADC_SMPR0）

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
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				SMP23[2:0]			SMP22[2:0]			SMP21[2:0]			SMP20[2:0]		
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:12 保留，必须保持为复位值。

Bits 11:0 **SMPx[2:0]**：通道 CH20-23 的采样时间

这些位用于独立地选择每个通道的采样时间。在采样周期中通道选择位必须保持不变。

000：2.5 周期

001：6.5 周期
010：11.5 周期
011：17.5 周期
100：25.5 周期
101：35.5 周期
110：48.5 周期
111：640.5 周期

12.10.20 ADC 规则数据寄存器（ADC_DRx（x=1...8））

地址偏移：0x6C-0x88
复位值：0x0000 0000

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

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

Bits 31:16 保留，必须保持为复位值。

Bits 15:0 **DATAx[15:0]：**规则转换的数据

这些位为只读，包含了规则通道的转换结果。

12.10.21 ADC 控制寄存器 3（ADC_CR3）

地址偏移：0x94
复位值：0x0000 0001

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														CFG_UPD	PRG_RDY
														rw	r

Bits 31:2 保留，必须保持为复位值。

Bit 1 **CFG_UPD：**启动程序以地址值对寄存器执行写操作

设置 1 启动程序寄存器

当程序完成时，硬件清 0

0：配置寄存器完成

1：启动程序对寄存器进行配置

Bit 0 **PRG_RDY：**寄存器是否空闲

0：正在对寄存器进行配置

1：寄存器配置完成或空闲

注：详见 12.3.5 更新寄存器配置步骤

13 高级控制定时器（TIM8）

13.1 TIM8 简介

高级控制定时器（TIM8）由一个 16 位的自动装载计数器组成，它由一个可编程的预分频器驱动。

它适合多种用途，包含产生输出波形（输出比较、PWM、嵌入死区时间的互补 PWM 等）。

使用定时器预分频器和 RCC 时钟控制预分频器，可以实现脉冲宽度和波形周期从几个微秒到几个毫秒的调节。高级控制定时器（TIM8）和通用定时器（TIMx）是完全独立的，它们不共享任何资源。它们可以同步操作。

13.2 TIM8 主要特征

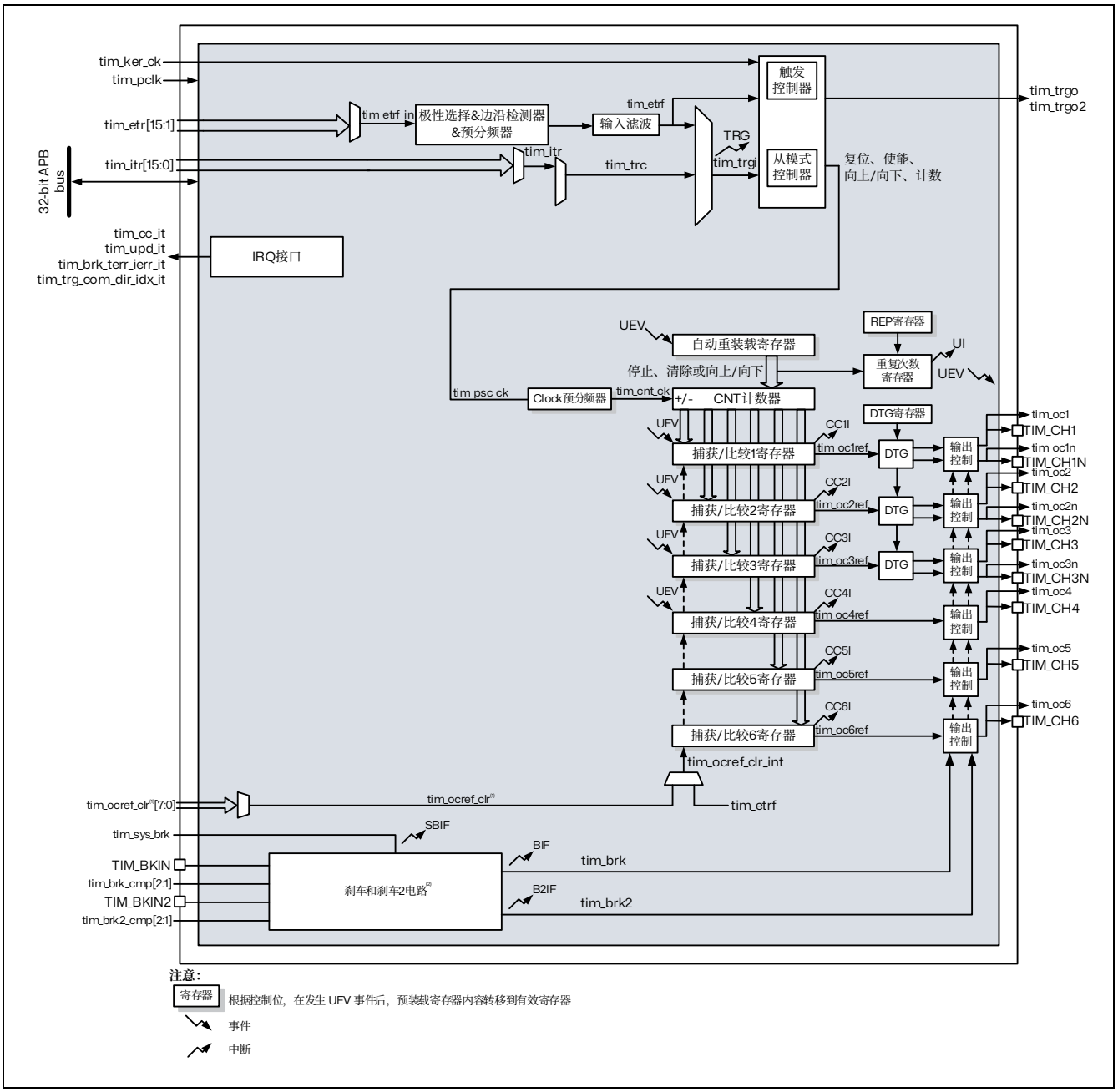
TIM8 定时器的功能包括：

- 16 位向上、向下、向上/下自动装载计数器
- 16 位可编程（可以实时修改）预分频器，计数器时钟频率的分频系数为 1～65536 之间的任意数值
- 多达 6 个独立通道：
  - 输出比较
  - PWM 生成（边缘或中间对齐模式）
  - 单脉冲模式输出
- 死区时间可编程的互补输出
- 使用外部信号控制定时器和定时器互联的同步电路
- 允许在指定数目的计数器周期之后更新定时器寄存器的重复计数器
- 用于将定时器输出信号置于安全的用户选择配置的两路刹车输入信号
- 如下事件发生时产生中断：
  - 更新：计数器向上溢出/向下溢出，计数器初始化（通过软件或者内部/外部触发）
  - 触发事件（计数器启动、停止、初始化或者由内部/外部触发计数）
  - 输出比较
  - 刹车信号输入
- 触发输入作为外部时钟或者按周期的电流管理
- 非对称模式

13.3 TIM8 功能描述

13.3.1 框图

图 13.1 高级控制定时器框图



1. 这一特性详见下一节：TIM8 引脚和内部信号
2. 参见刹车和刹车 2 电路以获得更多细节

13.3.2 TIM8 引脚和内部信号

本章节中的表格总结了 TIM 的输入和输出。

表 13.1 TIM 输入/输出引脚

引脚名称	信号类型	描述
TIM_CH1 TIM_CH2 TIM_CH3 TIM_CH5 TIM_CH6	输出	定时器多功能通道。 每个通道用于比较或 PWM。 外部触发器
TIM_CH1N TIM_CH2N TIM_CH3N	输出	由 TIM_CHx 输出派生的定时器的互补输出， 具有插入死区时间的能力
TIM_BKIN TIM_BKIN2	输入/输出	刹车和刹车 2 输入。这些输入也可以配置为双向模式。

表 13.2 TIM 内部输入/输出信号

内部信号名称	信号类型	描述
tim_etr[15:0]	输入	外部触发器内部输入总线。这些输入可被用作触发器、外部时钟或用于硬件的逐周期脉冲宽度控制。如果使用 tim_etr_in 预分频器，这些输入可以接收频率高于 tim_ker_ck 的时钟。
tim_itr[15:0]	输入	内部触发输入总线。这些输入可用于从模式控制器或作为输入时钟(tim_ker_ck 时钟的 1/4 以下)。
tim_trgo/tim_trgo2	输出	内部触发器输出。这些触发器被其他定时器或其他外设使用。
tim_ocref_clr[7:0]	输入	定时器 tim_ocref_clr 输入总线。这些输入可以用来清除 tim_ocxref 信号，通常用于硬件的逐周期脉冲宽度控制。
tim_brk_cmp[2:1]	输入	内部信号的刹车输入
tim_brk2_cmp[2:1]	输入	内部信号的刹车 2 输入
tim_sys_brk[n:0]	输入	系统中断输入。该输入收集 MCU 的系统级错误
tim_pclk	输入	定时器 APB 时钟
tim_ker_ck	输入	定时器内部时钟
tim_cc_it	输出	定时器比较中断
tim_upd_it	输出	定时器更新事件中断
tim_brk_terr_ierr_it	输出	定时器刹车、刹车 2、转换错误中断
tim_trg_com_dir_idx_it	输出	定时器触发、换向、方向

下表列出连接到 tim_itr 输入多路复用器的内部源

表 13.3 TIM8 内部触发连接

TIMx	TIM8
tim_itr0	TIM2_trgo
tim_itr1	TIM3_trgo
tim_itr2	TIM15_trgo

下表列出连接到 tim_etr 输入多路复用器的内部源

表 13.4 连接到 tim_etr 的输入多路复用器

定时器外部触发输入信号	定时器外部触发信号分配
	TIM8
tim_etr1	cmp1_out
tim_etr2	cmp2_out

下三张表列出了连接到 tim_brk 和 tim_brk2 输入的源

表 13.5 定时器刹车互连

tim_brk 输入	TIM8
TIM_BKIN	TIM8_BKIN pin
tim_brk_cmp1	cmp1_out
tim_brk_cmp2	cmp2_out

表 13.6 定时器刹车 2 互连

tim_brk2 输入	TIM8
TIM_BKIN2	TIM8_BKIN2 pin
tim_brk2_cmp1	cmp1_out
tim_brk2_cmp2	cmp2_out

表 13.7 系统中断互连

tim_sys_brk 输入	TIM8	SYSCFG_CFGR2 寄存器的使能位
tim_sys_brk0	内核锁定	CLL
tim_sys_brk1	Programmable Voltage Detector (PVD)	PVDL

下表列出连接到 tim_ocref_clr 输入多路复用器的内部源

表 13.8 连接到 ocref_clr 的输入多路复用器

定时器 tim_ocref_clr 信号	定时器 tim_ocref_clr 信号分配
	TIM8
tim_ocref_clr0	cmp1_out
tim_ocref_clr1	cmp2_out

13.3.3 时基单元

可编程高级控制定时器的主要模块是一个 16 位计数器及其相关的自动重载寄存器。计数器可递增计数、递减计数或同时进行递增和递减计数。计数器的时钟可通过预分频器进行分频。

计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。

时基单元包含：

- 计数器寄存器（TIM8_CNT）
- 预分频器寄存器（TIM8_PSC）
- 自动装载寄存器（TIM8_ARR）
- 重复次数寄存器（TIM8_RCR）

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以直接传送到影子寄存器，也可以在每次发生更新事件（UEV）时传送到影子寄存器，这取决于 TIM8_CR1 寄存器中的自动重载预装载使能位（ARPE）。当计数器达到上溢值（或者在递减计数时达到下溢值）并且 TIM8_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生 进行详细介绍。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIM8_CR1 寄存器中的计数器启动位（CEN）置 1 时，才会启动计数器（有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明）。

注意，计数器将在 TIM8_CR1 寄存器的 CEN 位置 1 时刻的一个时钟周期后开始计数。

预分频器描述

预分频器可对计数器时钟频率进行分频，分频系数介于 1 和 65536 之间。该预分频器基于 TIM8_PSC 寄存器中的 16 位寄存器所控制的 16 位计数器。由于该控制寄存器具有缓冲功能，因此可对预分频器进行实时更改。而新的预分频比将在下一更新事件发生时被采用。

下图以一些示例说明在预分频比实时变化时计数器的行为：

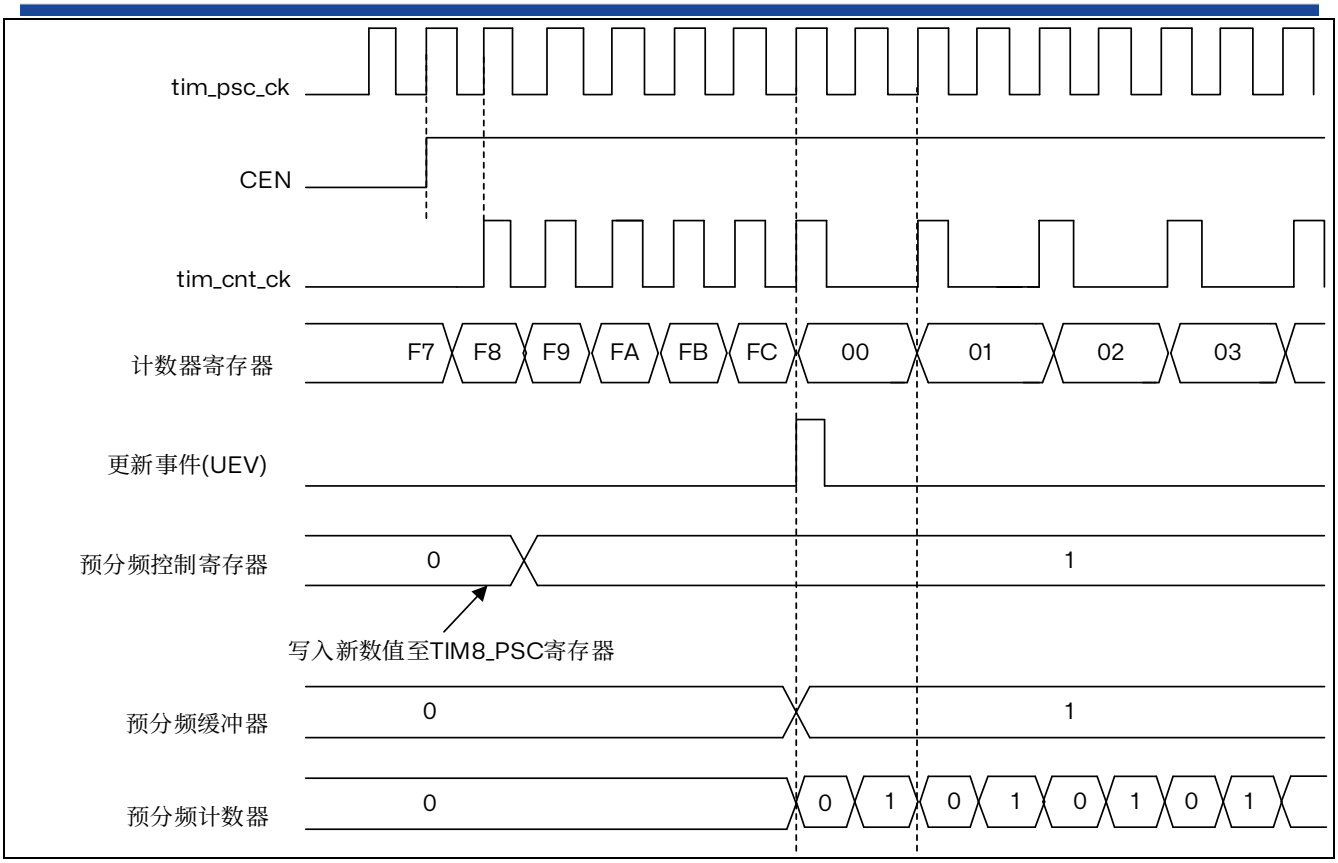


图 13.2 当预分频器的参数从 1 变成 2 时，计数器的时序图

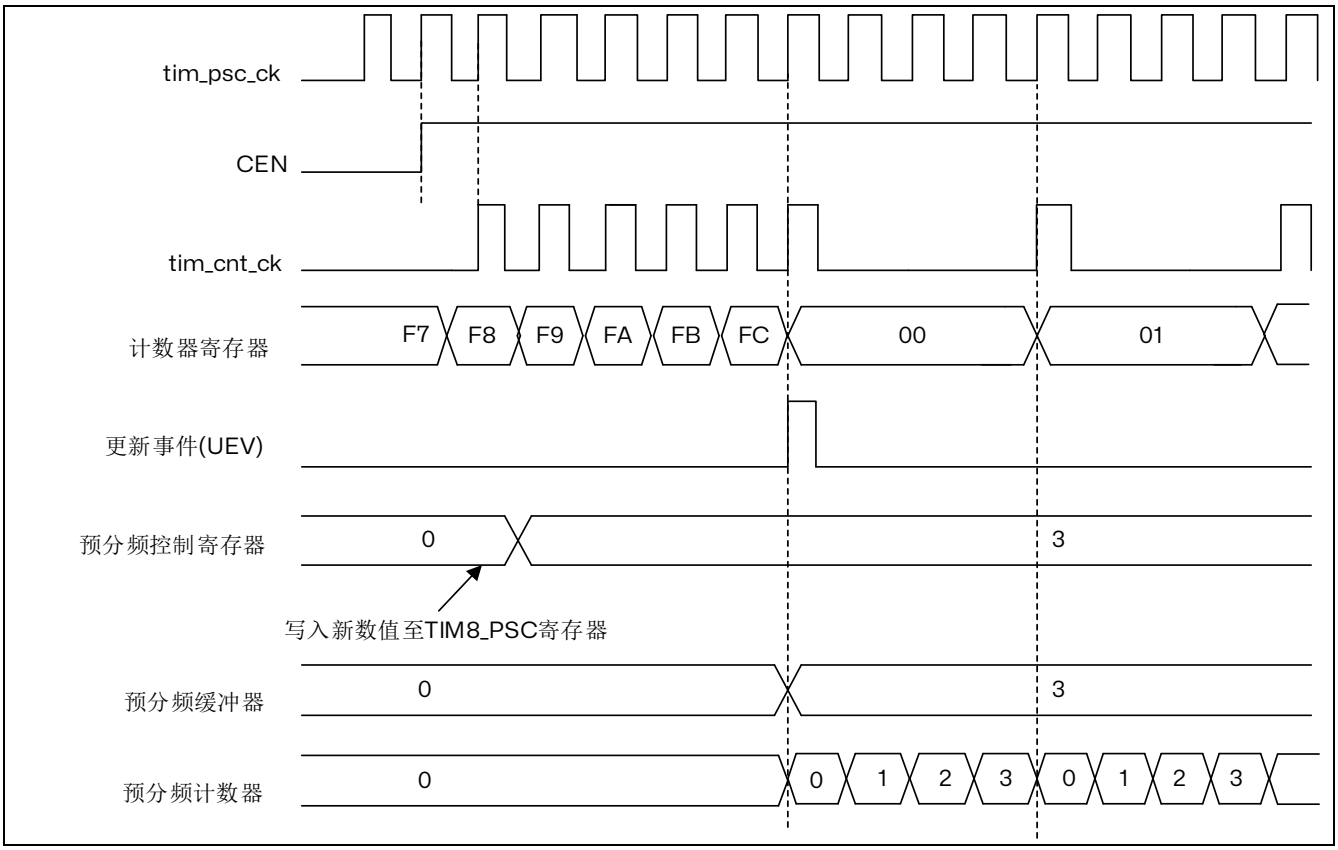


图 13.3 当预分频器的参数从 1 变成 4 时，计数器的时序图

13.3.4 计数模式

递增计数模式

在递增计数模式下，计数器从 0 计数到自动重载值（TIM8_ARR 寄存器的内容），然后重新从 0 开始计数并生成计数器上溢事件。

如果使用重复计数器，则当递增计数的重复次数达到重复计数器寄存器中编程的次数加一次（TIM8_RCR+1）后，将生成更新事件(UEV)。否则，将在每次计数器上溢时产生更新事件。

将 TIM8_EGR 寄存器的 UG 位置 1（通过软件或使用从模式控制器）时，也将产生更新事件。

通过软件将 TIM8_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIM8_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断）。

发生更新事件时，将更新所有寄存器且将更新标志（TIM8_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 重复计数器中将重新装载 TIM8_RCR 寄存器的内容
- 自动重载影子寄存器将以预装载值(TIM8_ARR)进行更新
- 预分频器的缓冲区中将重新装载预装载值（TIM8_PSC 寄存器的内容）

以下各图以一些示例说明当 TIM8_ARR=0x36 时不同时钟频率下计数器的行为。

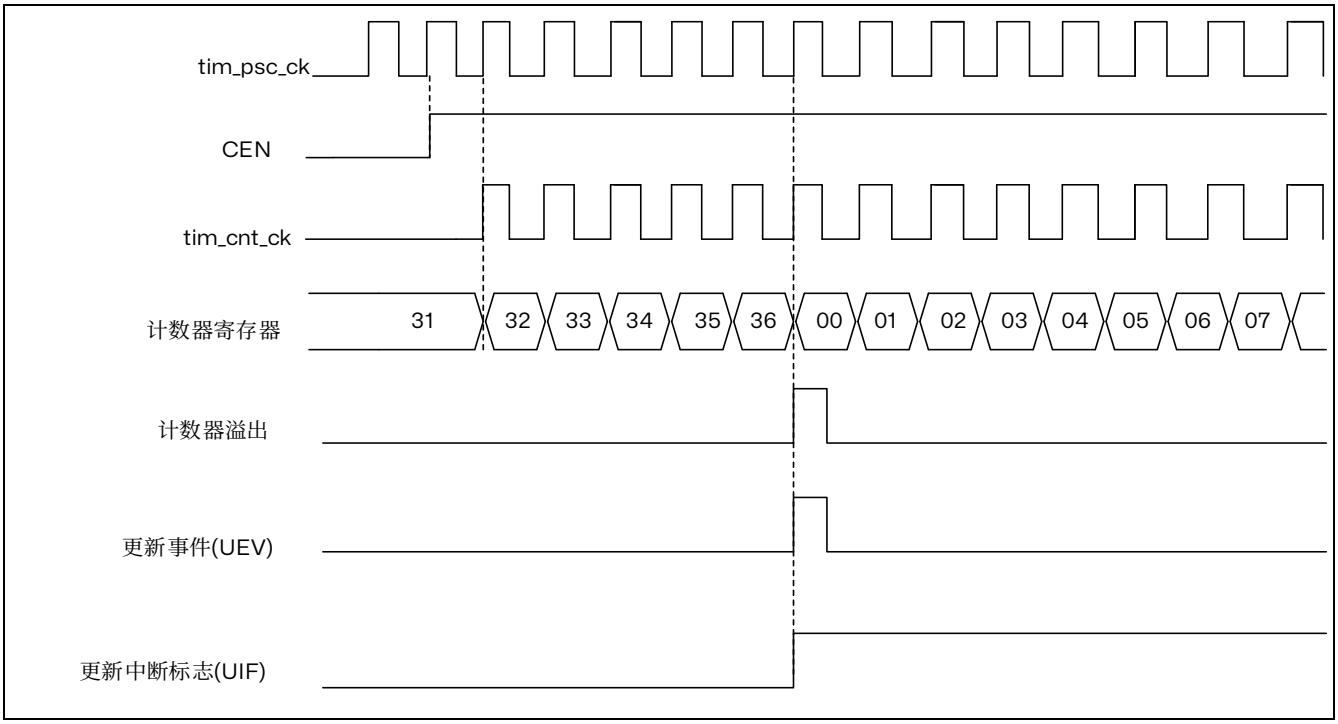


图 13.4 计数器时序图，内部时钟分频因子为 1

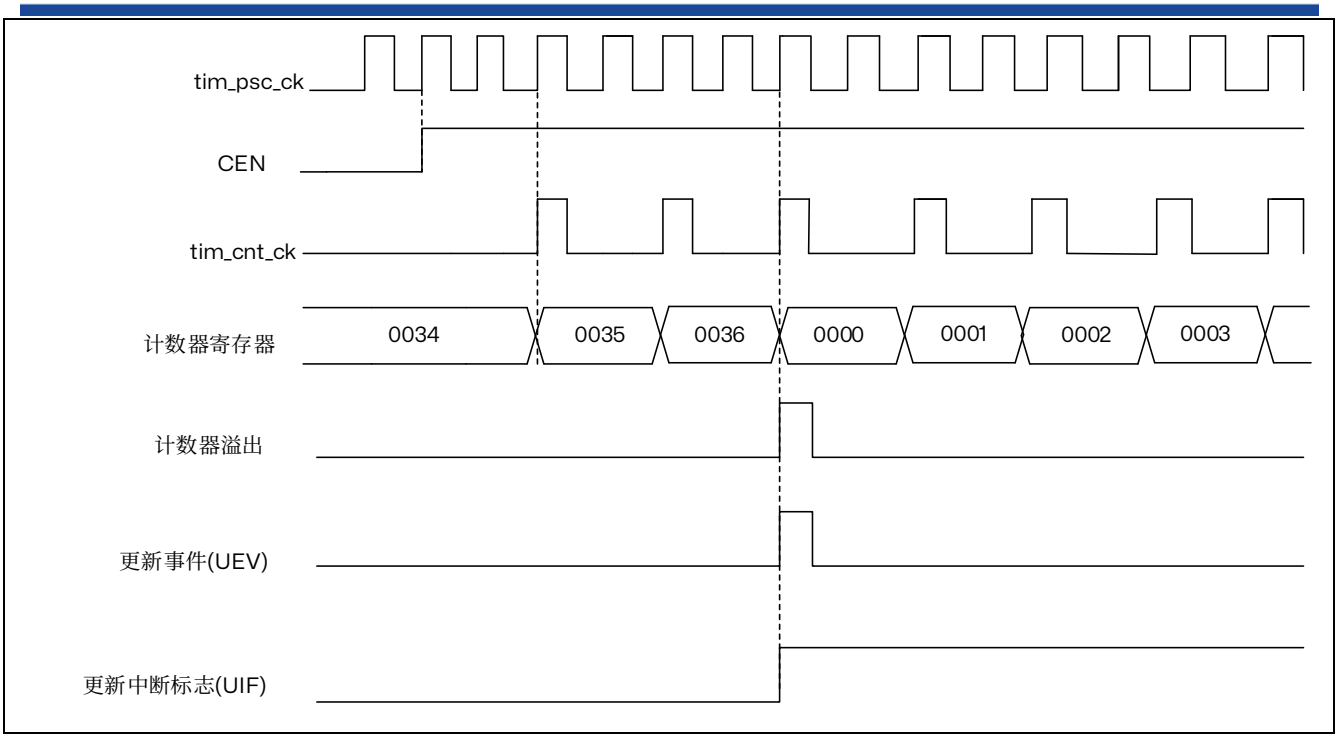


图 13.5 计数器时序图，内部时钟分频因子为 2

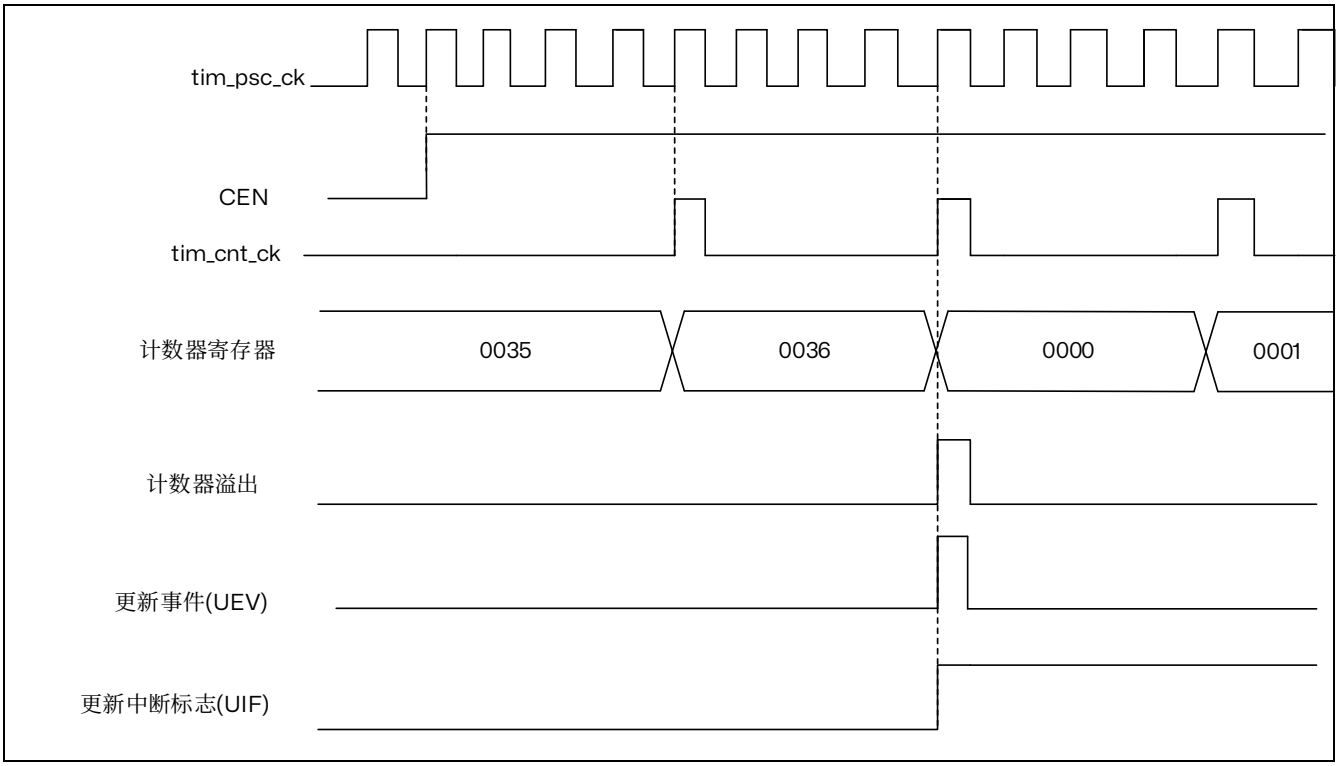


图 13.6 计数器时序图，内部时钟分频因子为 4

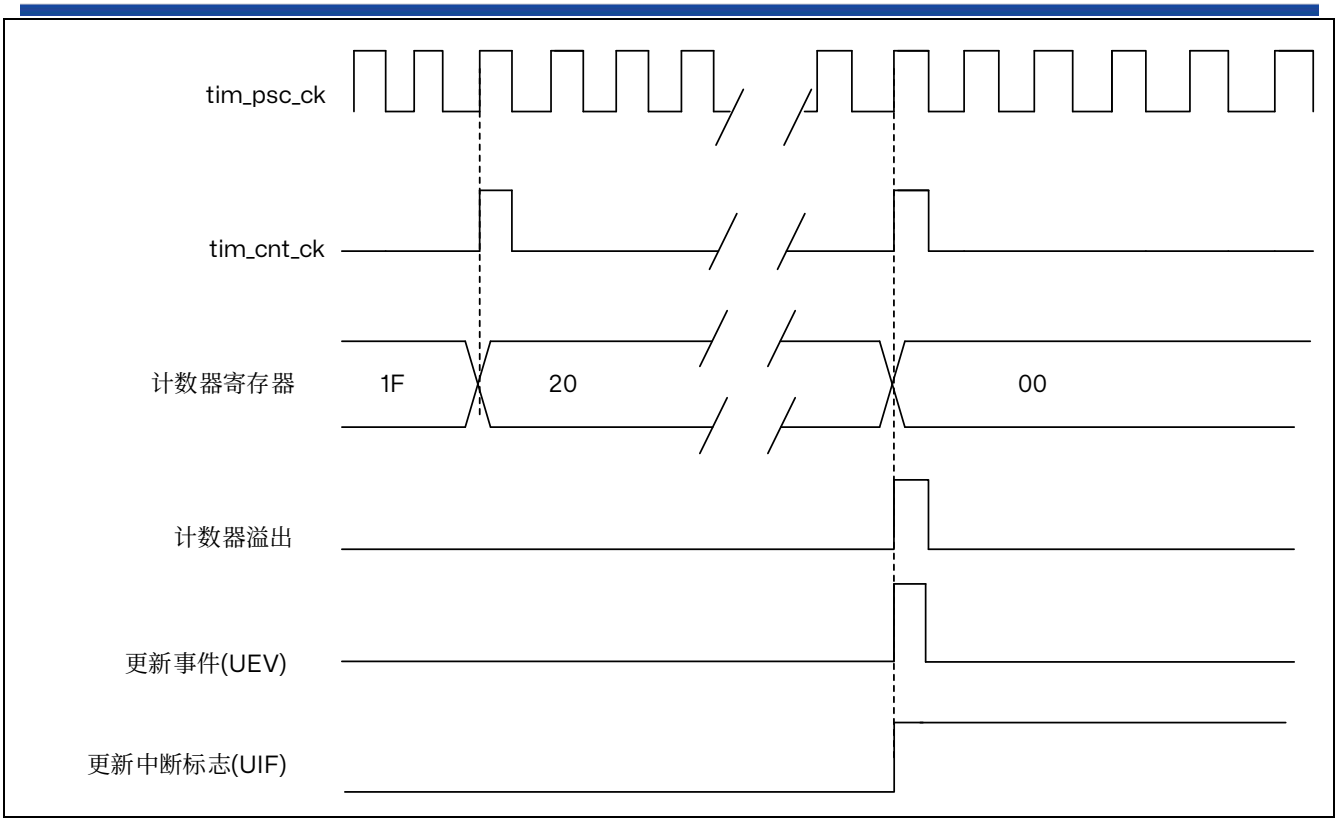


图 13.7 计数器时序图，内部时钟分频因子为 N

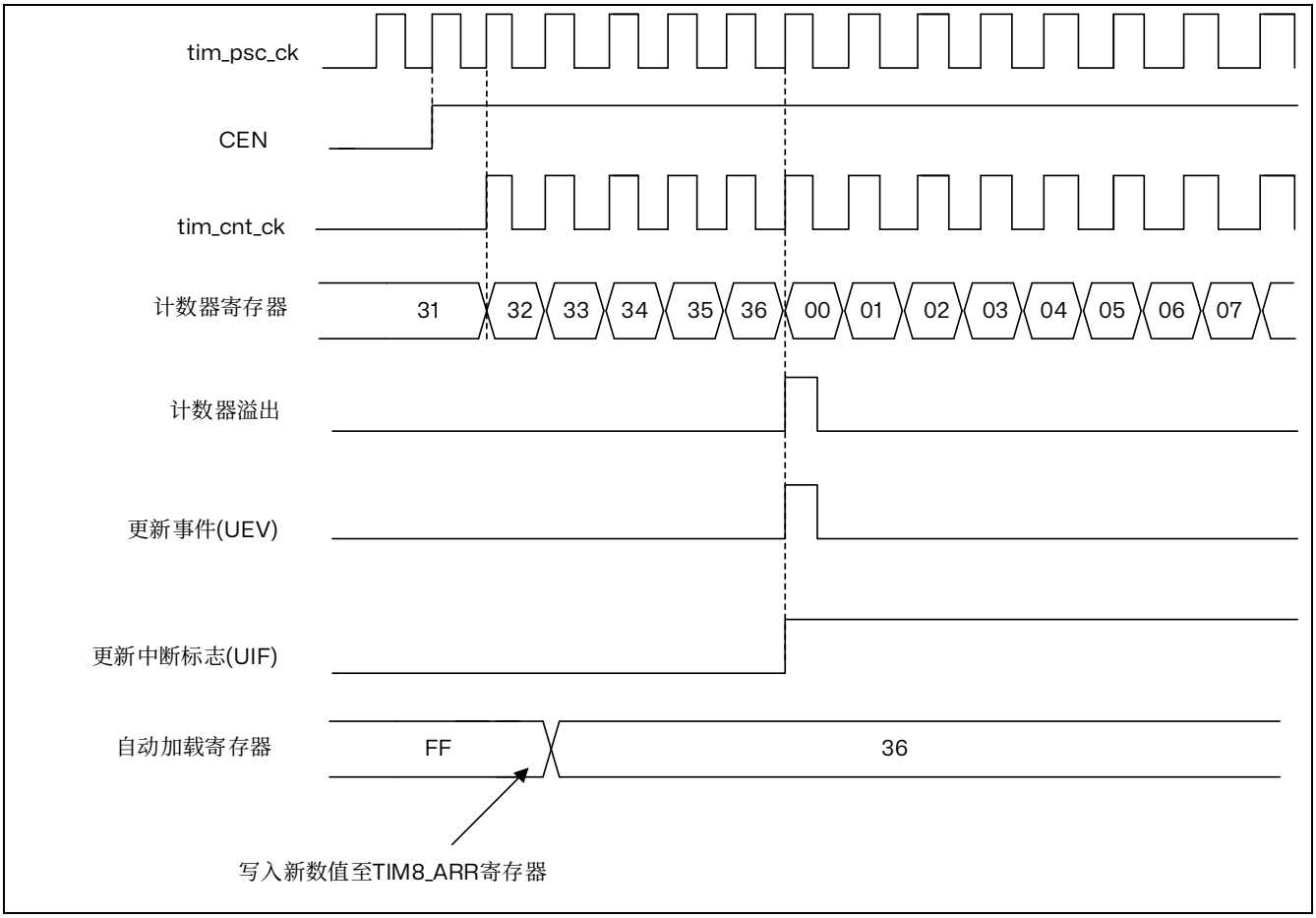


图 13.8 计数器时序图，当 ARPE=0 时的更新事件（TIM8_ARR 没有预装入）

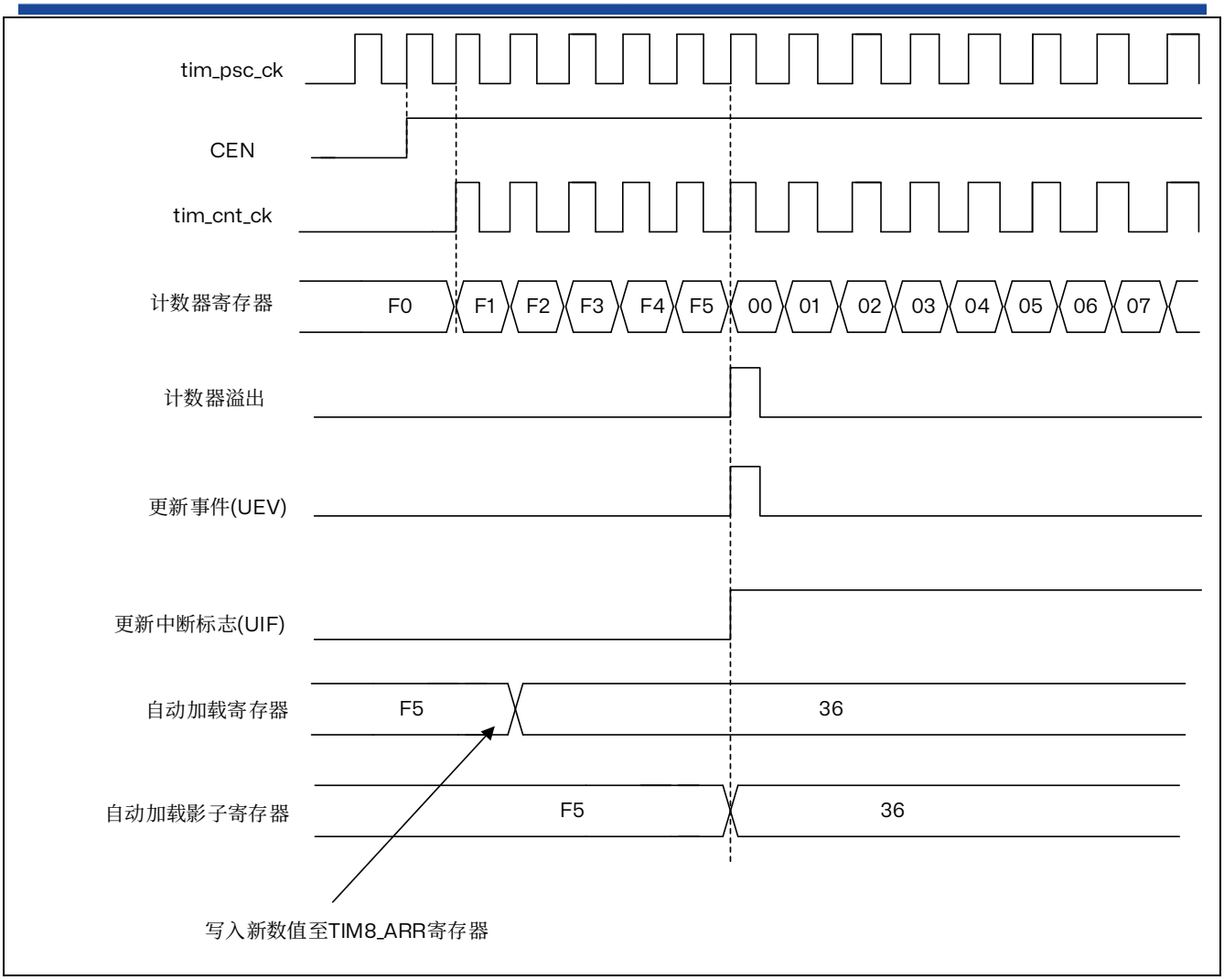


图 13.9 计数器时序图，当 ARPE=1 时的更新事件（预装入了 TIM8_ARR）

递减计数模式

在递减计数模式下，计数器从自动重载值（TIM8_ARR 寄存器的内容）开始递减计数到 0，然后重新从自动重载值开始计数并生成计数器下溢事件。

如果使用重复计数器，则当递减计数的重复次数达到重复计数器寄存器中编程的次数加一次(TIM8_RCR+1)后，将生成更新事件(UEV)。否则，将在每次计数器下溢时产生更新事件。

将 TIM8_EGR 寄存器的 UG 位置 1（通过软件或使用从模式控制器）时，也将产生更新事件。

通过软件将 TIM8_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器会重新从当前自动重载值开始计数，而预分频器计数器则重新从 0 开始计数（但预分频比保持不变）。

此外，如果 TIM8_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断请求）。

发生更新事件时，将更新所有寄存器且将更新标志（TIM8_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 重复计数器中将重新装载 TIM8_RCR 寄存器的内容
- 预分频器的缓冲区中将重新装载预装载值（TIM8_PSC 寄存器的内容）

- 自动重载活动寄存器将以预装载值（TIM8_ARR 寄存器的内容）进行更新。注意，自动重载寄存器会在计数器重载之前得到更新，因此，下一个计数周期就是我们所希望的新的周期长度

以下各图以一些示例说明当 TIM8_ARR=0x36 时不同时钟频率下计数器的行为。

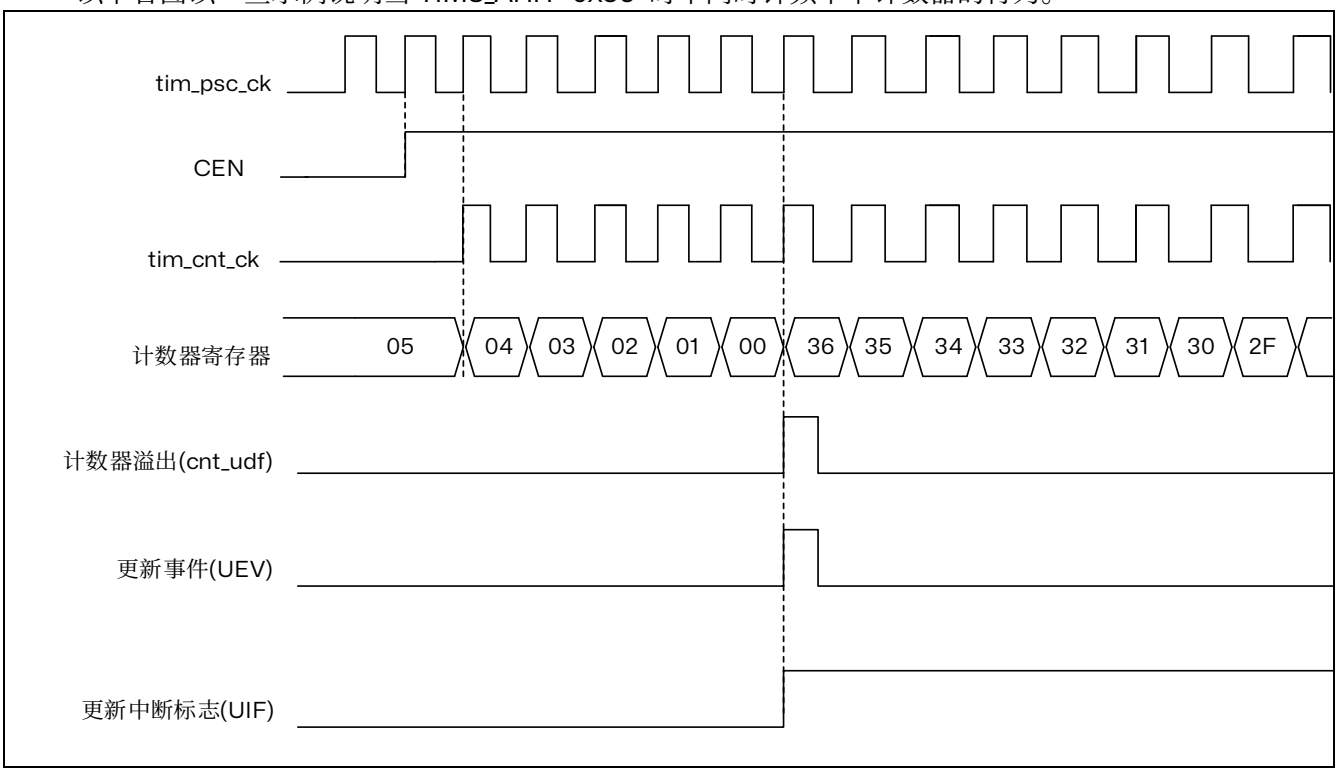


图 13.10 计数器时序图，内部时钟分频因子为 1

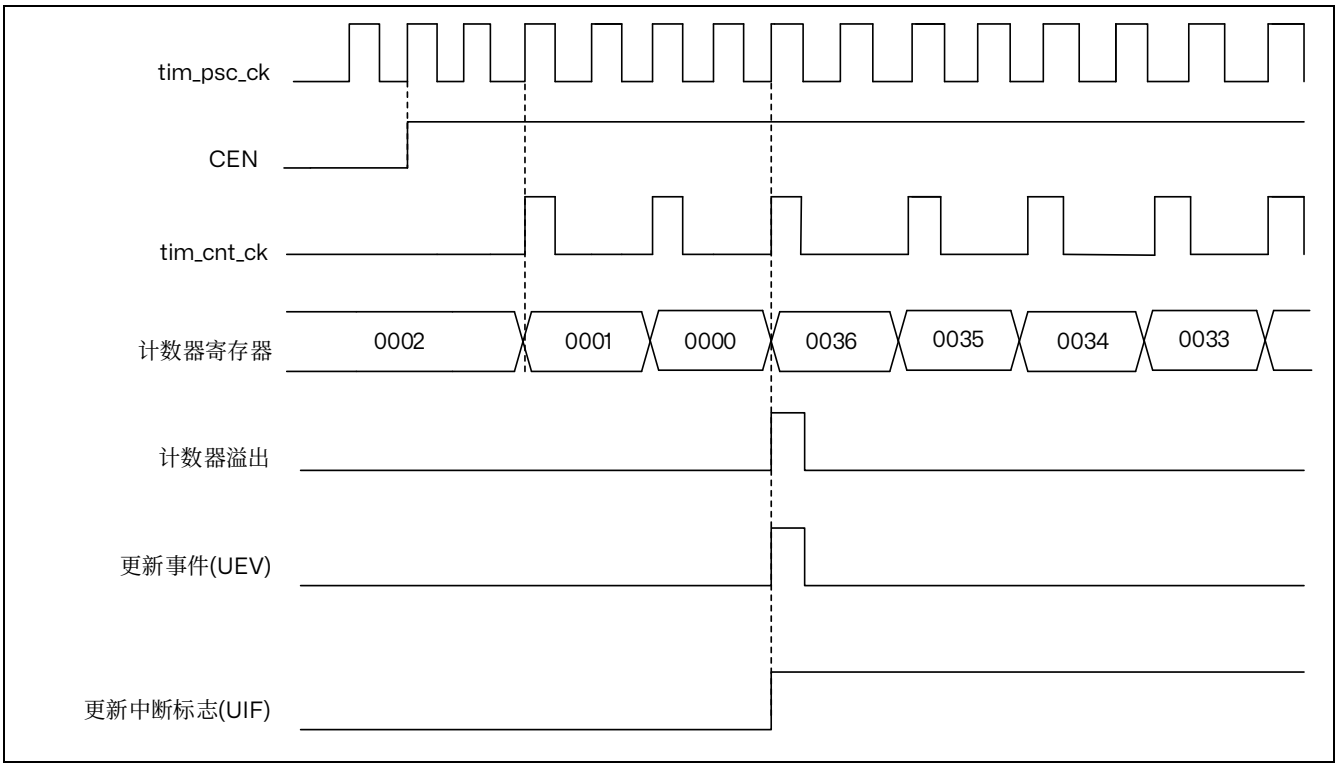


图 13.11 计数器时序图，内部时钟分频因子为 2

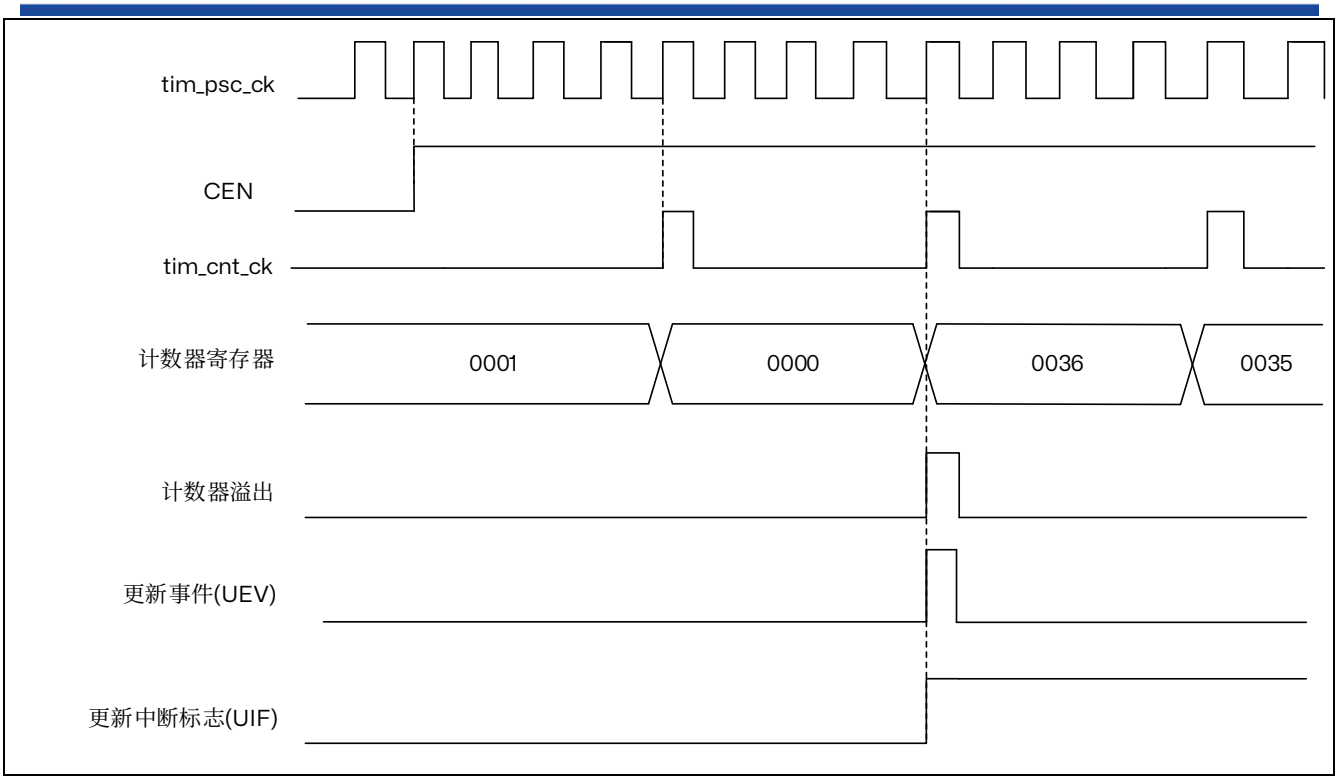


图 13.12 计数器时序图，内部时钟分频因子为 4

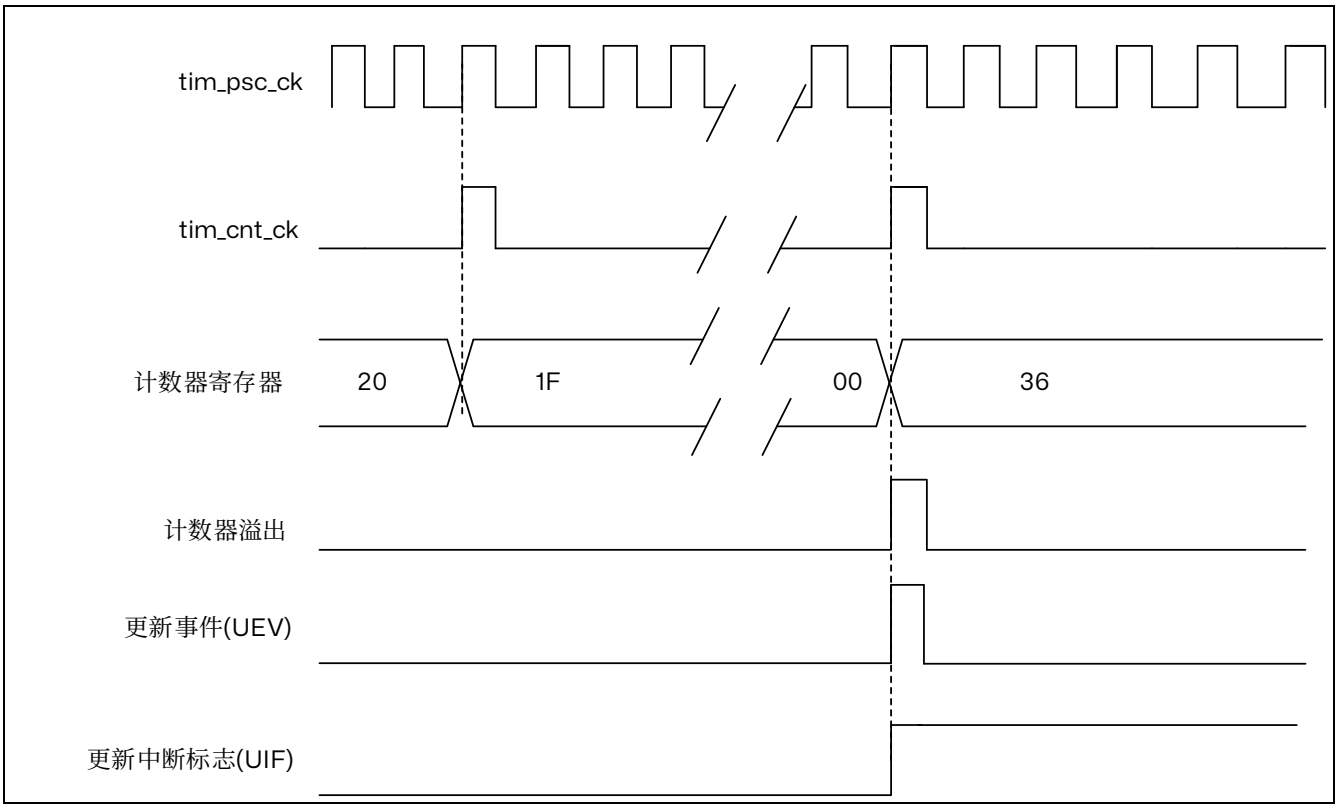


图 13.13 计数器时序图，内部时钟分频因子为 N

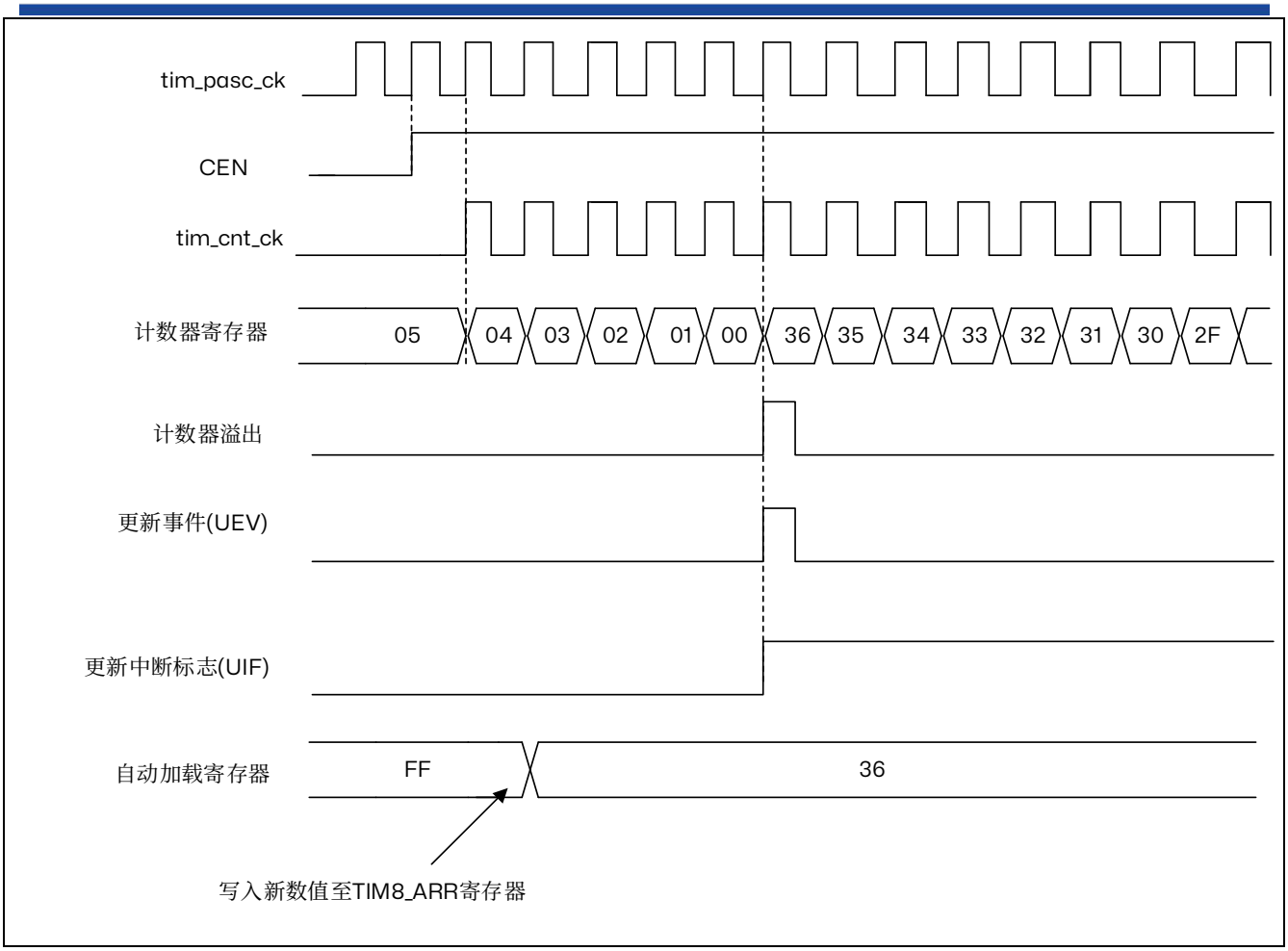


图 13.14 计数器时序图，当没有使用重复计数器时的更新事件

中央对齐模式（递增/递减计数）

在中央对齐模式下，计数器从 0 开始计数到自动重载值（TIM8_ARR 寄存器的内容）- 1，生成计数器上溢事件；然后从自动重载值开始向下计数到 1 并生成计数器下溢事件。之后从 0 开始重新计数。

当 TIM8_CR1 寄存器中的 CMS 位不为“00”时，中央对齐模式有效。将通道配置为输出模式时，其输出比较中断标志将在以下模式下置 1，即：计数器递减计数（中央对齐模式 1，CMS=“01”）、计数器递增计数（中央对齐模式 2，CMS =“10”）以及计数器递增/递减计数（中央对齐模式 3，CMS=“11”）。

在此模式下，TIM8_CR1 寄存器的 DIR 方向位不可写入值，而是由硬件更新并指示当前计数器方向。

每次发生计数器上溢和下溢时都会生成更新事件，或将 TIM8_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。这种情况下，计数器以及预分频器计数器将重新从 0 开始计数。

通过软件将 TIM8_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器仍会根据当前自动重载值进行递增和递减计数。

此外，如果 TIM8_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成 UEV 更新事件，但不会将 UIF 标志置 1（因此，不会发送任何中断）。

发生更新事件时，将更新所有寄存器且将更新标志（TIM8_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 重复计数器中将重新装载 TIM8_RCR 寄存器的内容
- 预分频器的缓冲区中将重新装载预装载值（TIM8_PSC 寄存器的内容）
- 自动重载活动寄存器将以预装载值（TIM8_ARR 寄存器的内容）进行更新。注意，如果更新操作是由计数器上溢触发的，则自动重载寄存器在重载计数器之前更新，因此，下一个计数周期就是我们所希望的新的周期长度（计数器被重载新的值）。

以下各图以一些示例说明不同时钟频率下计数器的行为。

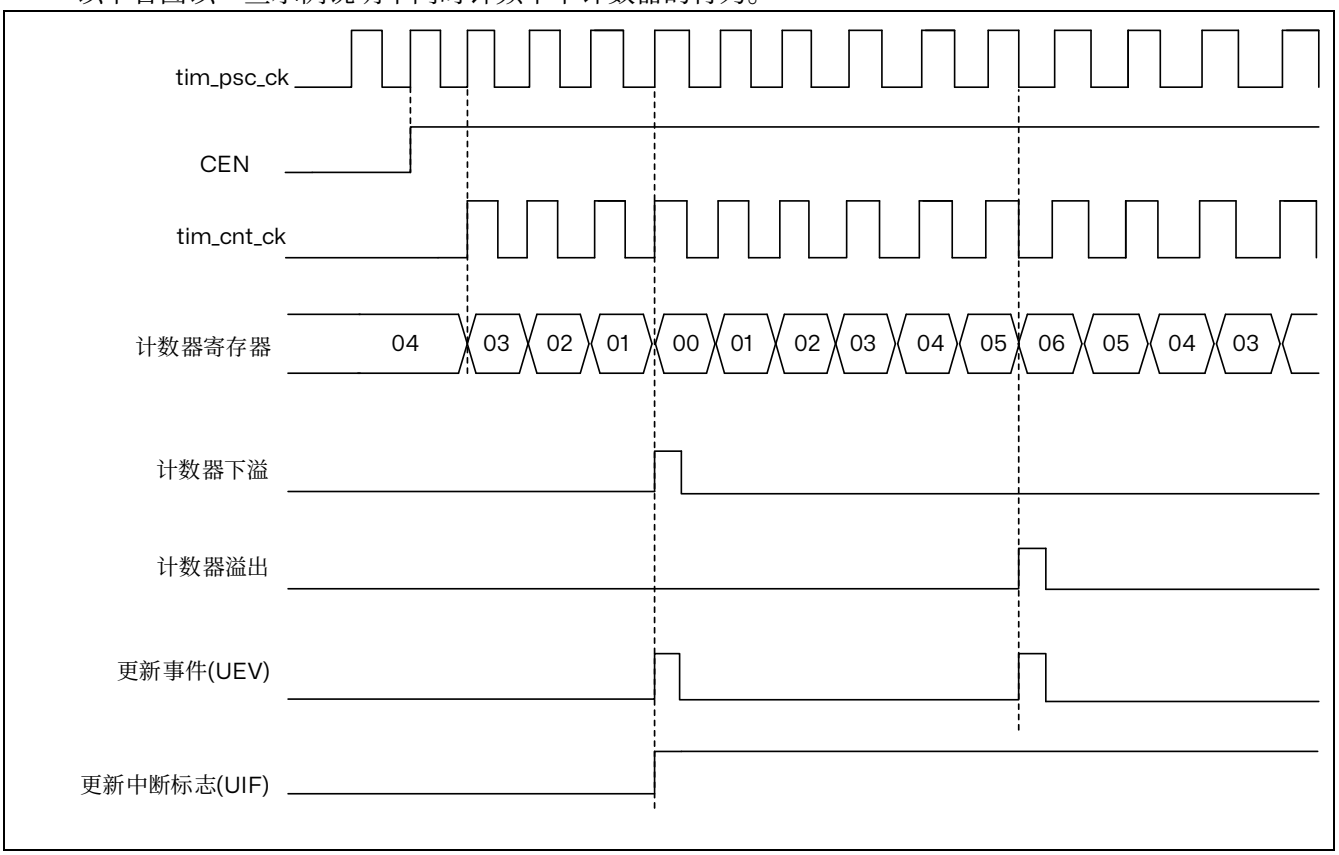


图 13.15 计数器时序图，内部时钟分频因子为 1，TIM8_ARR=0x6

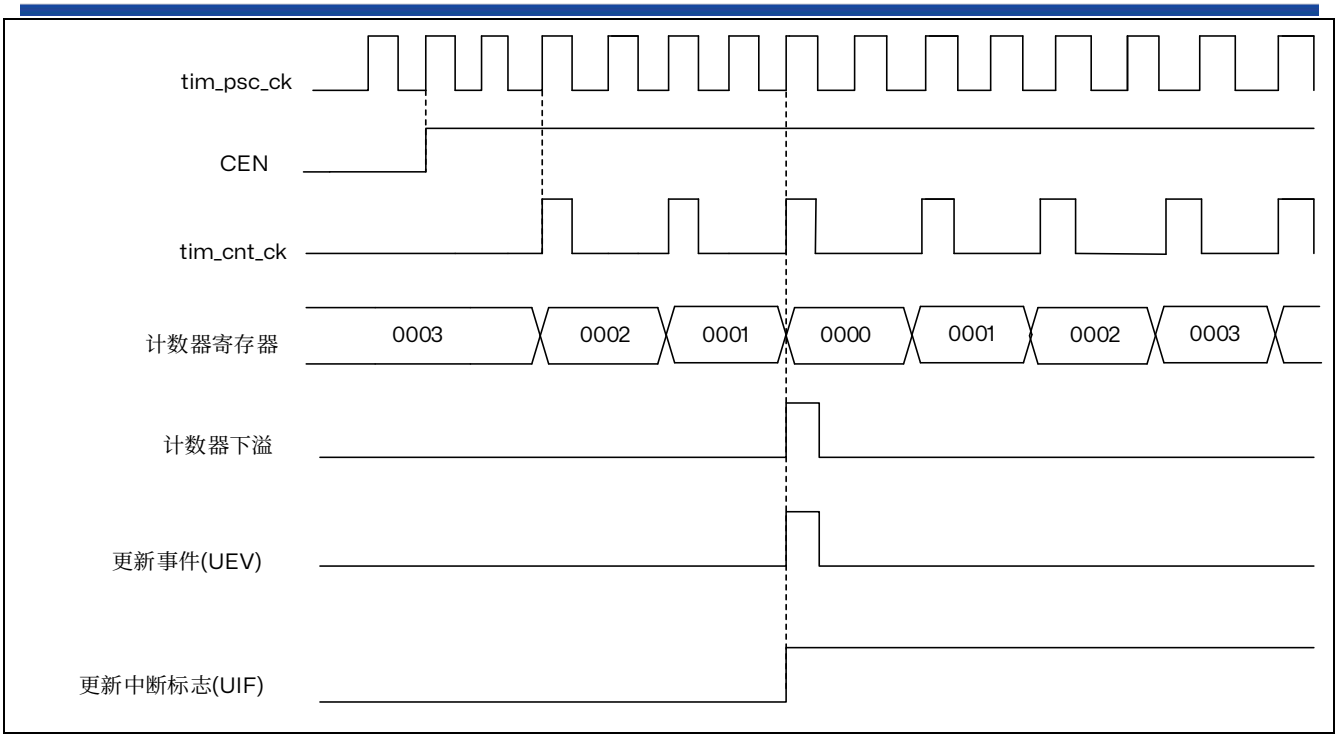


图 13.16 计数器时序图，内部时钟分频因子为 2

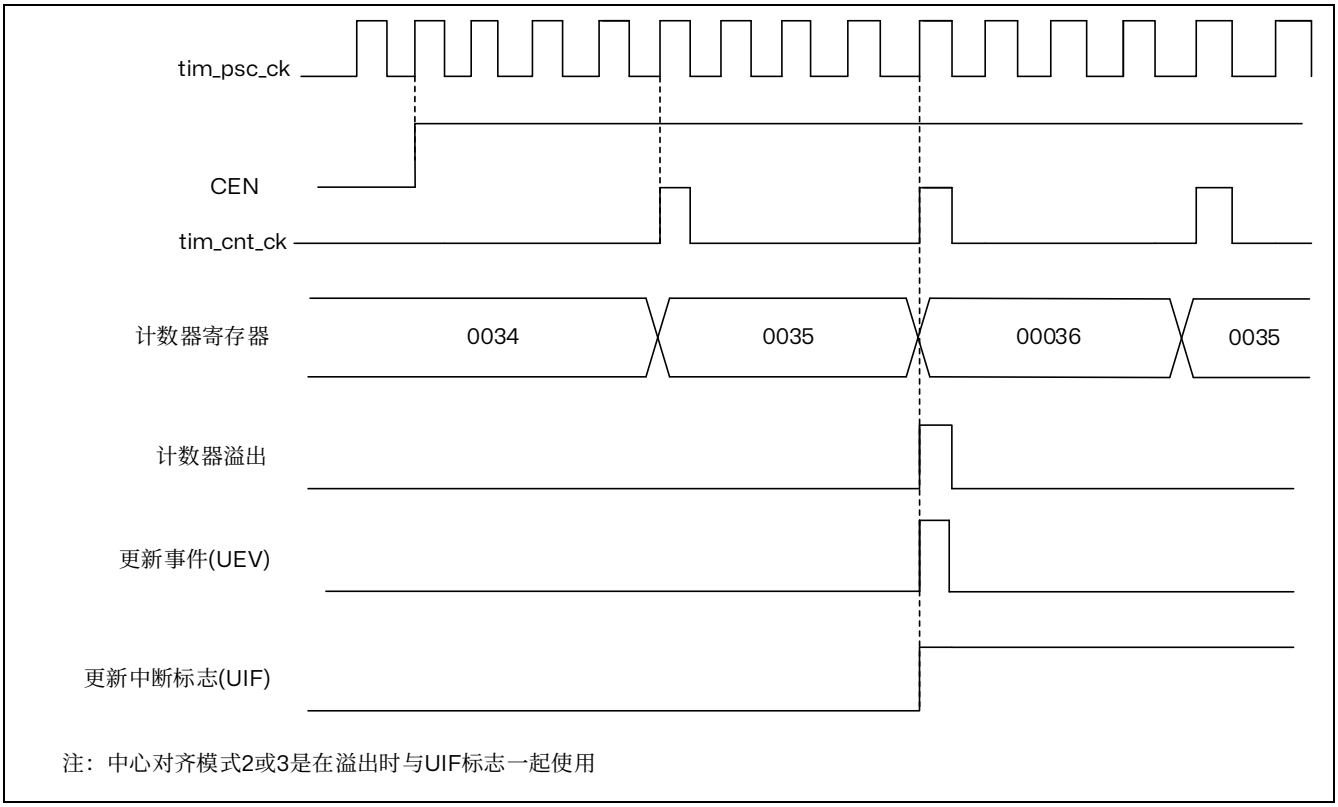


图 13.17 计数器时序图，内部时钟分频因子为 4，TIM8_ARR=0x36

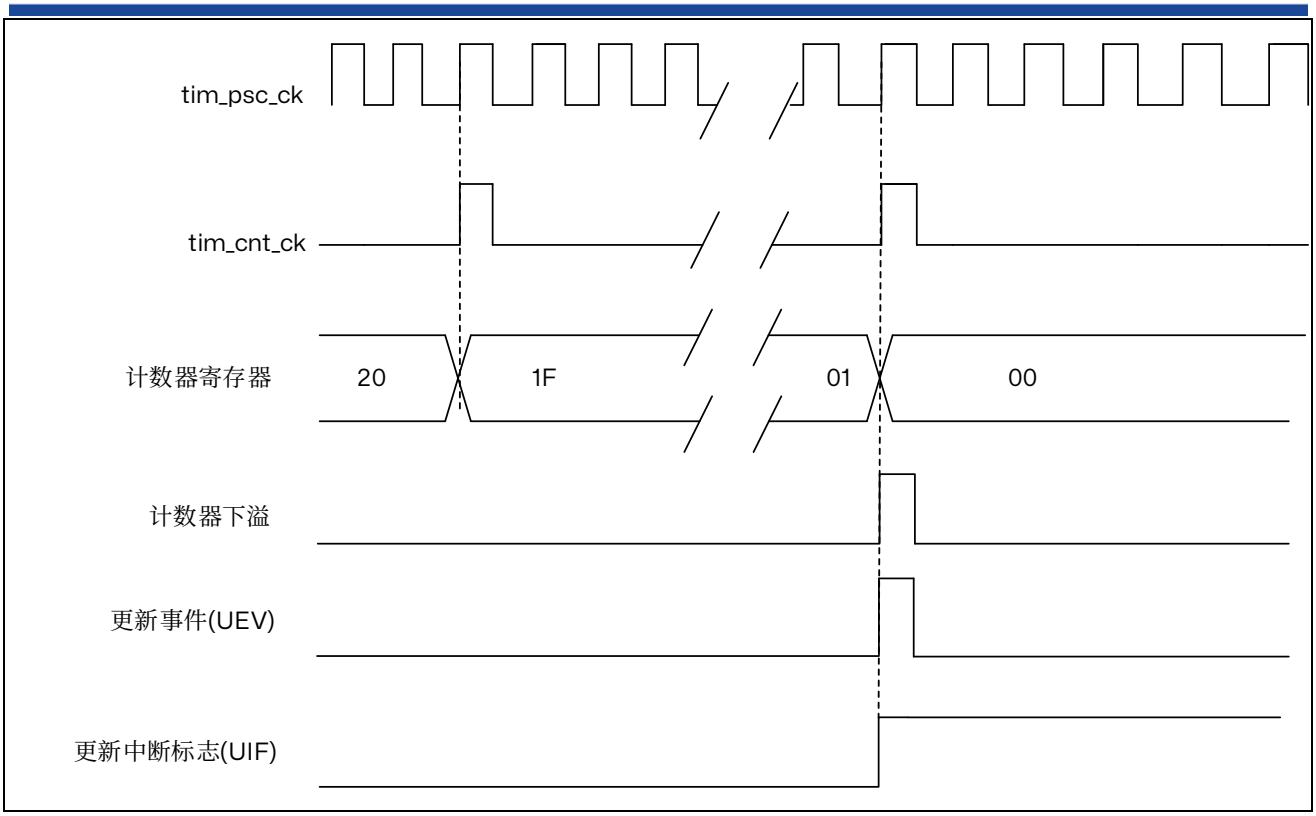


图 13.18 计数器时序图，内部时钟分频因子为 N

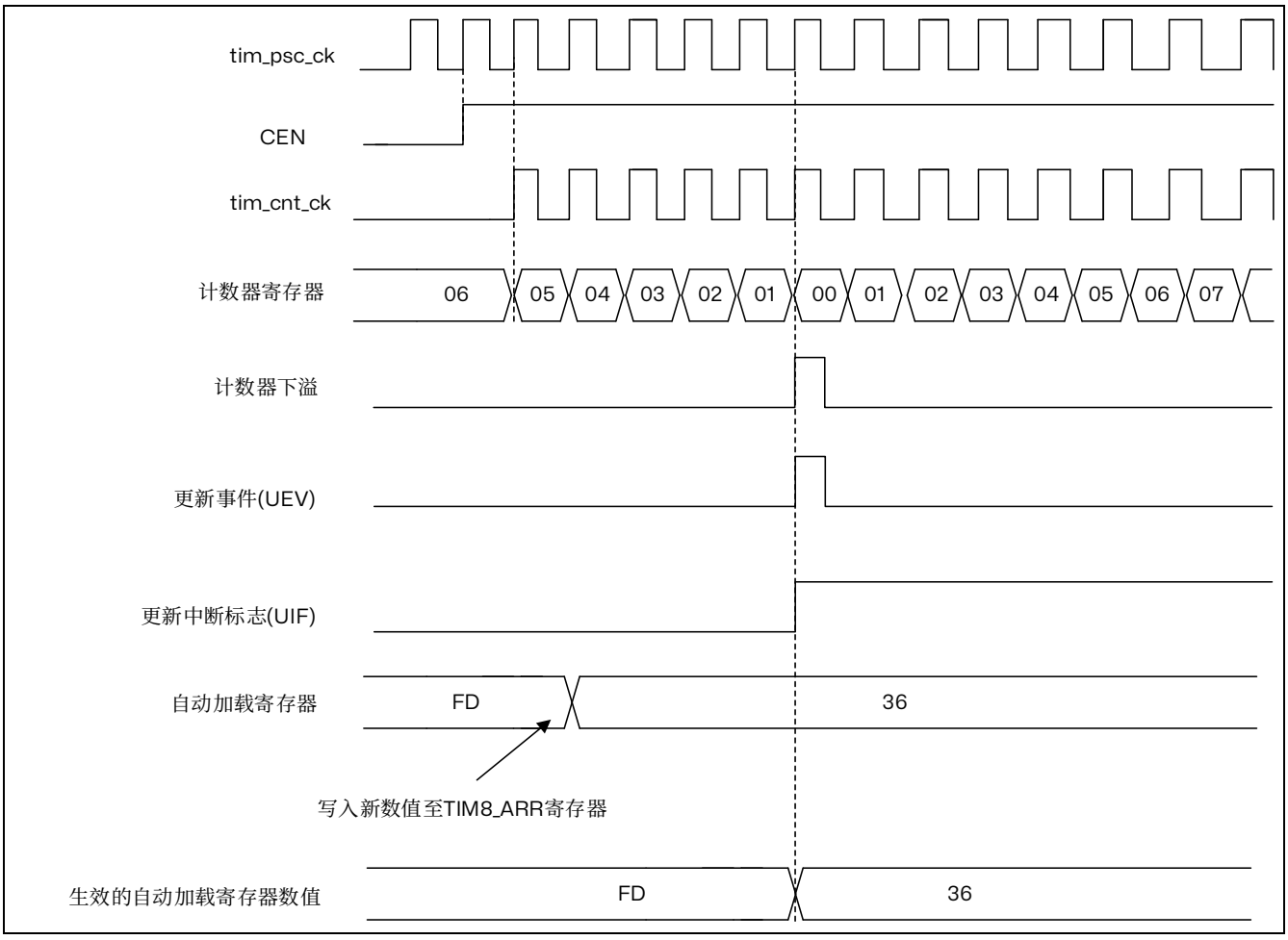


图 13.19 计数器时序图，ARPE=1 时的更新事件（计数器下溢）

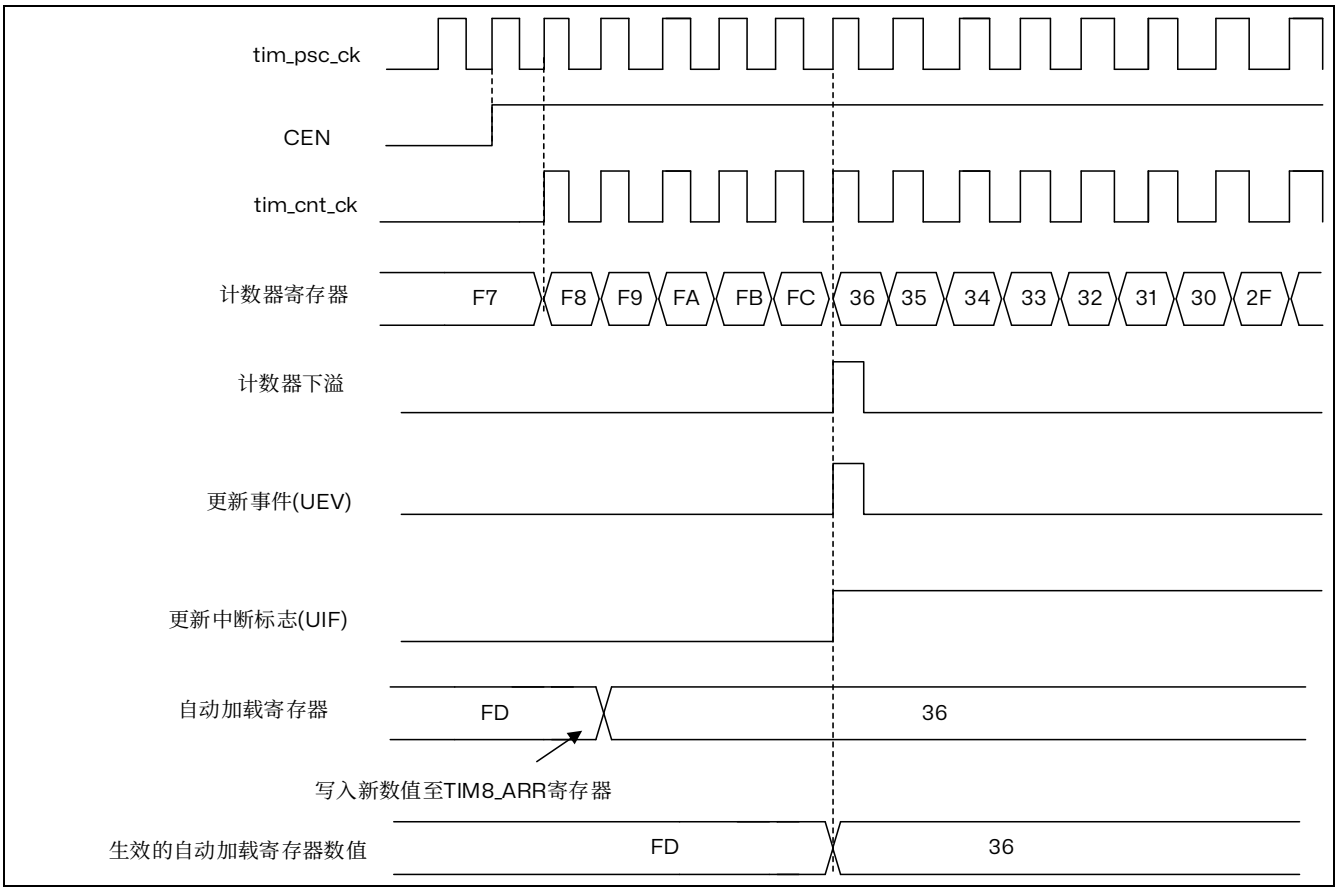


图 13.20 计数器时序图，ARPE=1 时的更新事件（计数器上溢）

13.3.5 重复计数器

时基单元介绍如何因计数器上溢/下溢而生成更新事件(UEV)。实际上，只有当重复计数器达到零时，才会生成更新事件。这在生成 PWM 信号时很有用。

这意味着，每当发生 N+1 个计数器上溢或下溢（其中，N 是 TIM8_RCR 重复计数器寄存器中的值），数据就将从预装载寄存器转移到影子寄存器（TIM8_ARR 自动重载寄存器、TIM8_PSC 预分频器寄存器以及比较模式下的 TIM8_CCRx 捕获/比较寄存器）。

重复计数器在下列情况下递减：

- 递增计数模式下的每个计数器上溢。
- 递减计数模式下的每个计数器下溢。
- 中央对齐模式下每个计数器上溢和计数器下溢。

  尽管这限制了最大重复次数为 32768 个 PWM 周期，但它使得每个 PWM 周期可以更新两次占空比。在中央对齐模式下，每个 PWM 周期仅刷新一次比较寄存器时，由于模式的对称性，最大分辨率为 $2xT_{ck}$。

重复计数器是自动重载类型；其重复率为 TIM8_RCR 寄存器所定义的值。当更新事件由软件（通过将 TIM8_EGR 寄存器的 UG 位置 1）或硬件（通过从模式控制器）生成时，无论重复计数器的值为多少，更新事件都将立即发生，并且在重复计数器中重新装载 TIM8_RCR 寄存器的内容。

在中央对齐模式下，如果 RCR 值为奇数，更新事件将在上溢或下溢时发生，这取决于何时写入 RCR

寄存器以及何时启动计数器。如果在启动计数器前写入 RCR，则 UEV 在上溢时发生。如果在启动计数器后写入 RCR，则 UEV 在下溢时发生。例如，如果 RCR = 3，UEV 将在每个周期的第四个上溢或下溢事件时生成（取决于何时写入 RCR）。

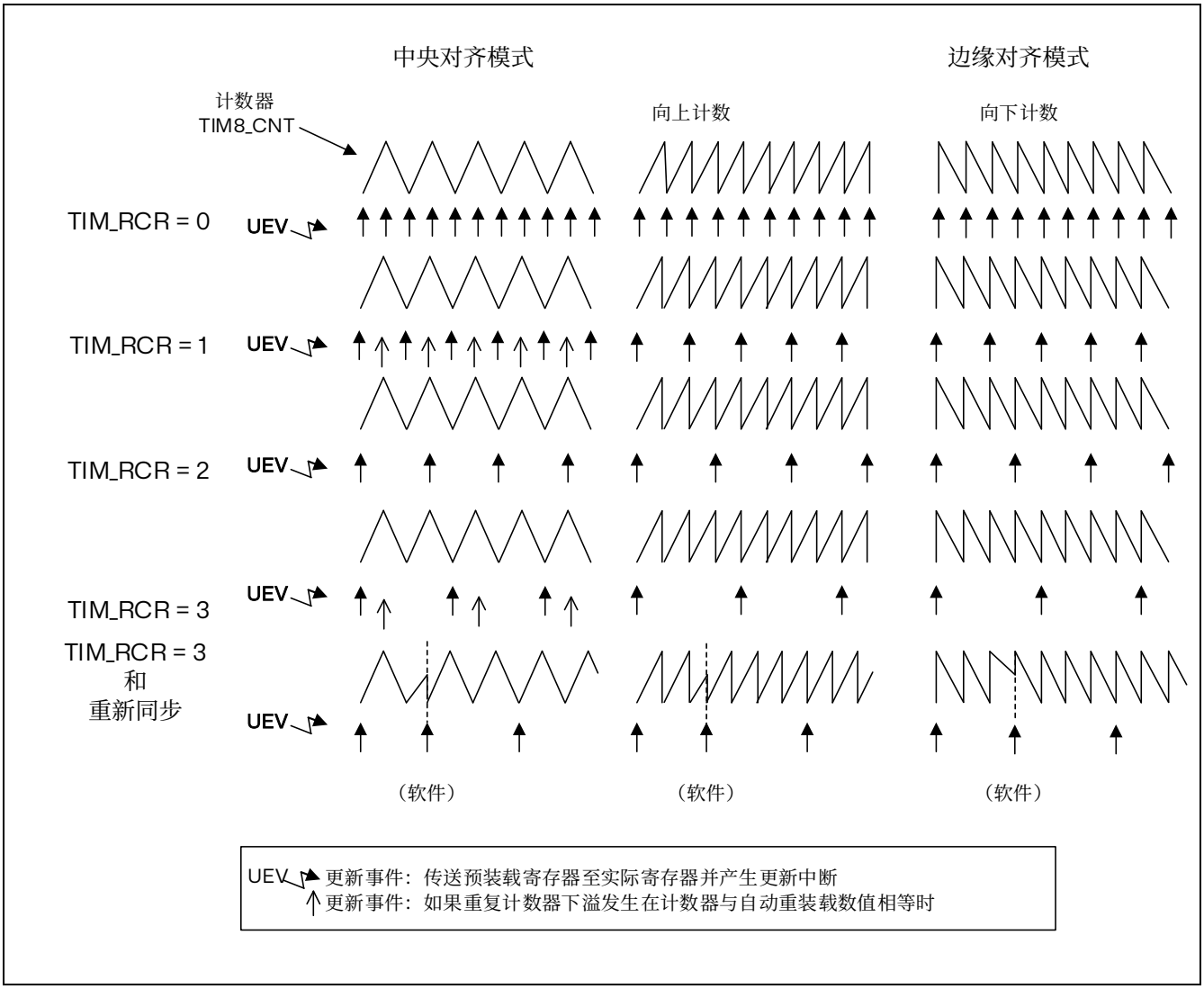


图 13.21 不同模式下更新速率的例子，及 TIM8_RCR 的寄存器设置

13.3.6 外部触发输入

定时器具有一个外部触发输入 tim_etr_in。它可以用于：

- 外部时钟（外部时钟模式 2）
- 从模式触发
- PWM 重置输入，用于逐周期电流调节

下图描述了 tim_etr_in 输入调节。输入极性由 TIM8_SMCR 寄存器的 ETP 位定义。触发器可以使用 ETPS[1:0]位域编程的分频器进行预分频，并使用 ETF[3:0]位域进行数字滤波。所得信号（tim_etrf）可用于三种目的：作为外部时钟、用于输出调节（通常用于重置 PWM 输出以进行电流限制）以及作为从模式控制器的触发器。

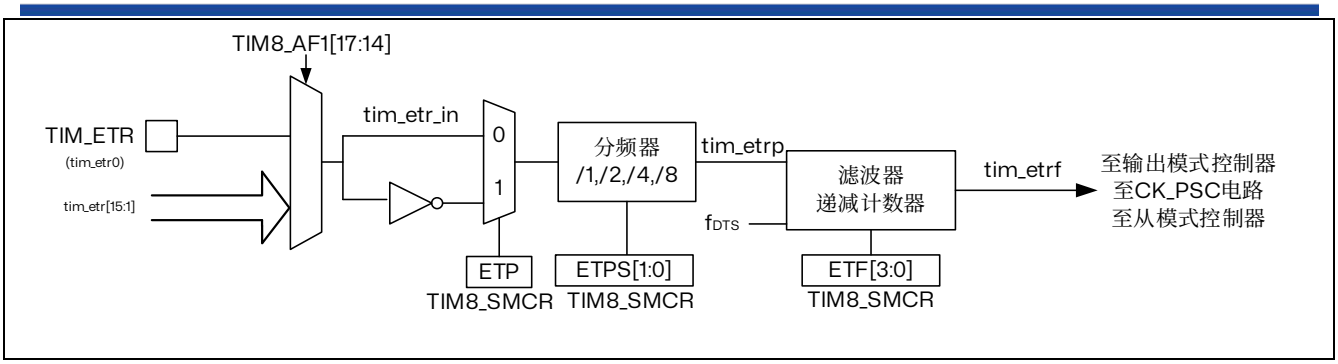


图 13.22 外部触发输入框图

tim_etr_in 输入来自多个来源：输入引脚（默认配置）或内部源。选择是通过 TIM8_AF1 寄存器中的 ETRSEL[3:0]位域完成的。

有关连接到 etr_in 输入的源列表，请参考之前的章节：TIM8 引脚和内部信号。

13.3.7 时钟选择

计数器时钟可由下列时钟源提供：

- 内部时钟（tim_ker_ck）
- 外部时钟模式 2：外部触发输入（tim_etr_in）

内部时钟源（tim_ker_ck）

如果禁止从模式控制器（SMS=000），则 CEN 位、DIR 位（TIM8_CR1 寄存器中）和 UG 位（TIM8_EGR 寄存器中）为实际控制位，并且只能通过软件进行更改（UG 除外，仍保持自动清零）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 tim_ker_ck 提供

下图显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）。

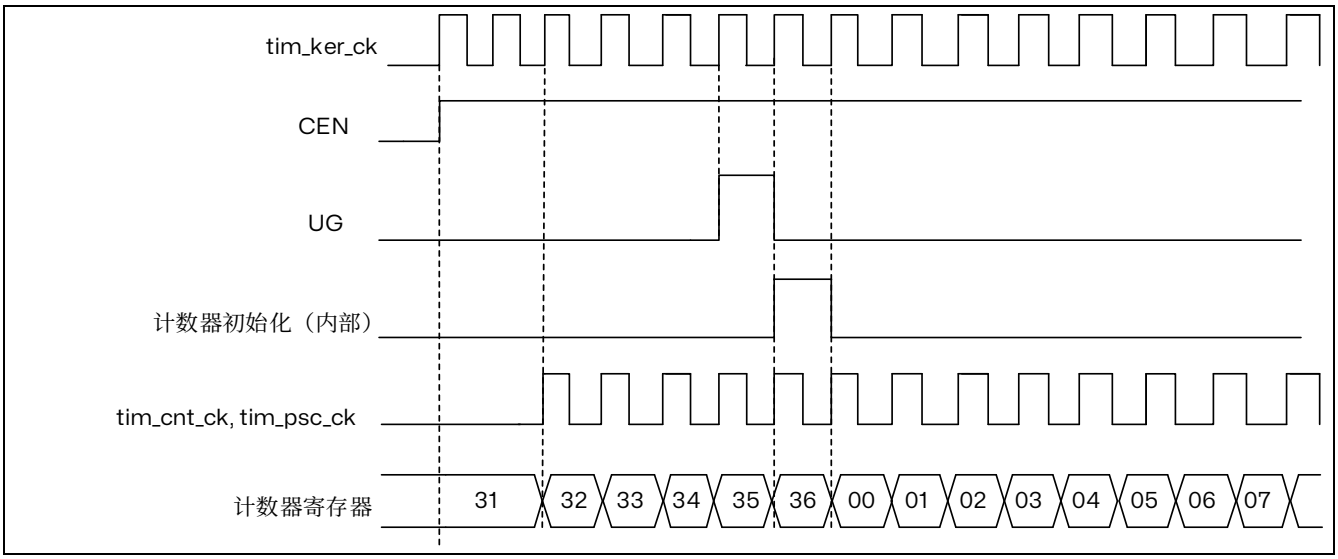


图 13.23 一般模式下的控制电路，内部时钟分频因子为 1

外部时钟模式 2

通过在 TIM8_SMCR 寄存器中写入 ECE=1 可选择此模式。

计数器可在外部触发输入 tim_etr_in 出现上升沿或下降沿时计数。

下图介绍了外部触发输入模块。

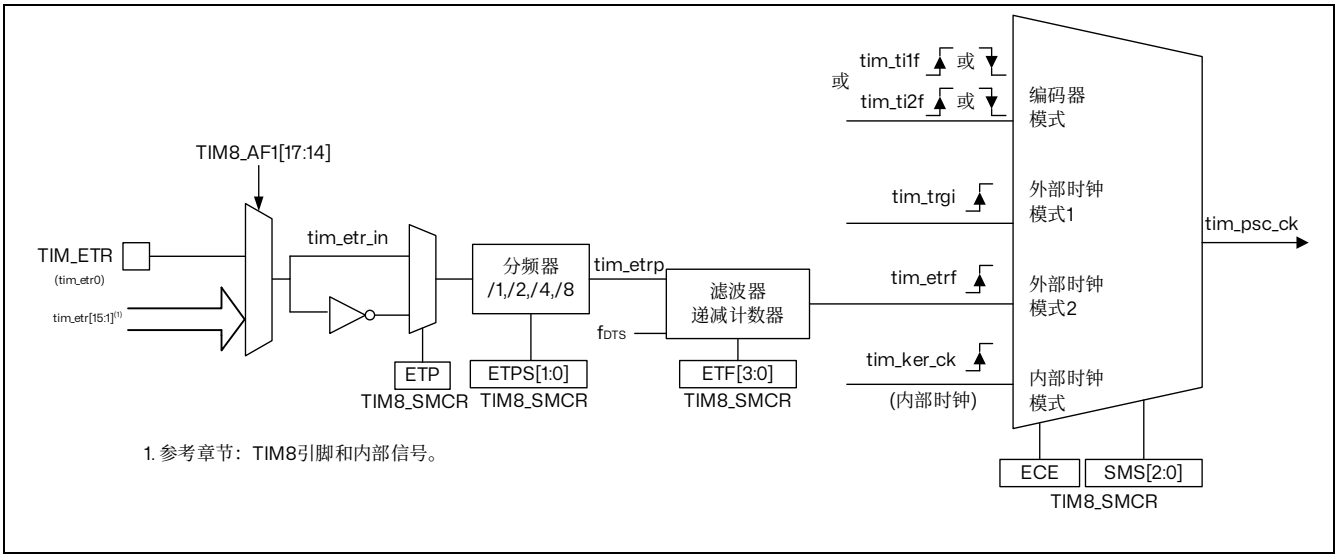


图 13.24 外部触发输入框图

例如，要使递增计数器在 tim_etr_in 每出现 2 个上升沿时计数，请执行以下步骤：

1. 由于此例中不需滤波器，因此在 TIM8_SMCR 寄存器中写入 ETF[3:0]=0000。
2. 通过在 TIM8_SMCR 寄存器中写入 ETPS[1:0]=01 来设置预分频器。
3. 通过在 TIM8_SMCR 寄存器中写入 ETP=0 来选择 ETR 引脚的上升沿检测。
4. 通过在 TIM8_SMCR 寄存器中写入 ECE=1 来使能外部时钟模式 2。
5. 通过在 TIM8_CR1 寄存器中写入 CEN=1 来使能计数器。

tim_etr_in 每出现 2 个上升沿，计数器计数一次。

ETR 的上升沿与实际计数器时钟之间的延迟是由于 ETRP 信号的重新同步电路引起的。

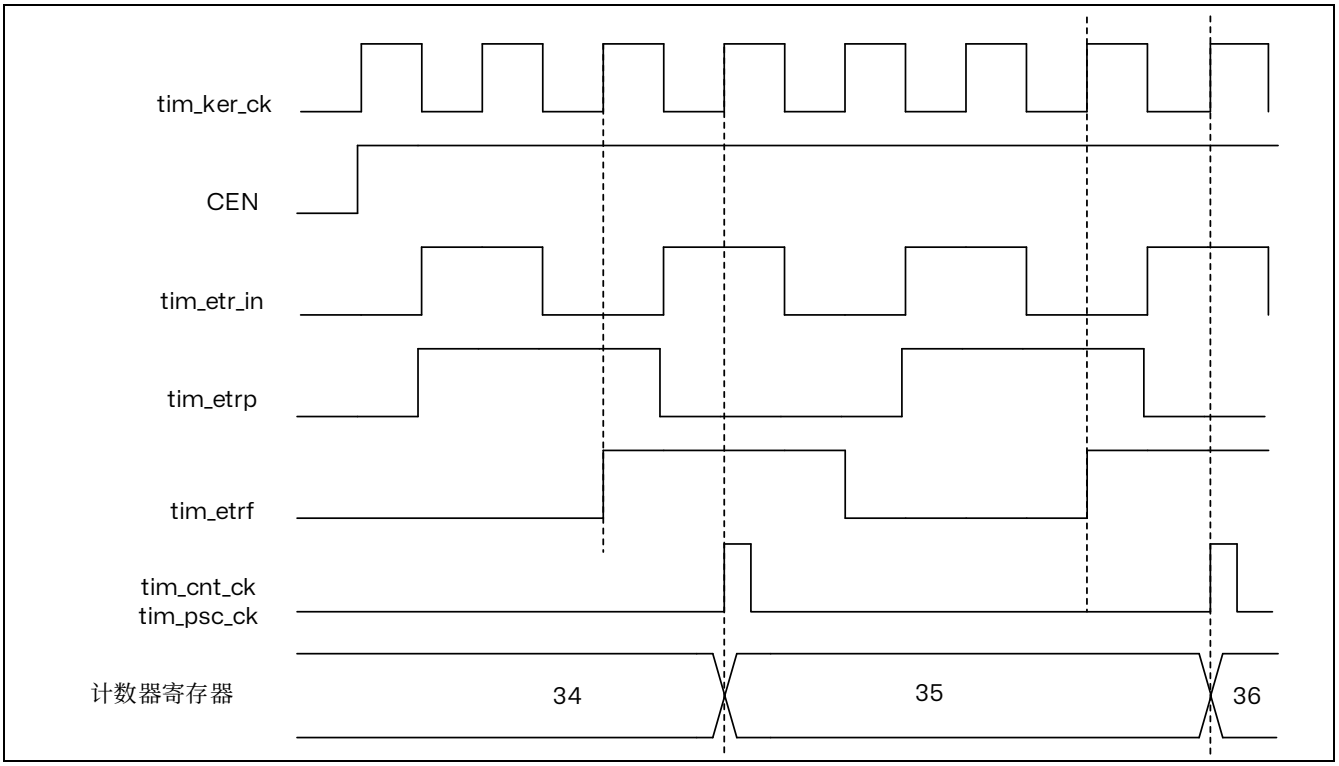


图 13.25 外部时钟模式 2 下的控制电路

13.3.8 捕获比较通道

每个捕获/比较通道均围绕一个捕获/比较寄存器（包括一个影子寄存器）和一个输出阶段（比较器和输出控制）构建而成。

下图概括介绍了一个捕获/比较通道。

输出级产生一个中间波形，其被用于参考：tim_ocx[x]（高电平有效）。极性在链路末尾起作用。

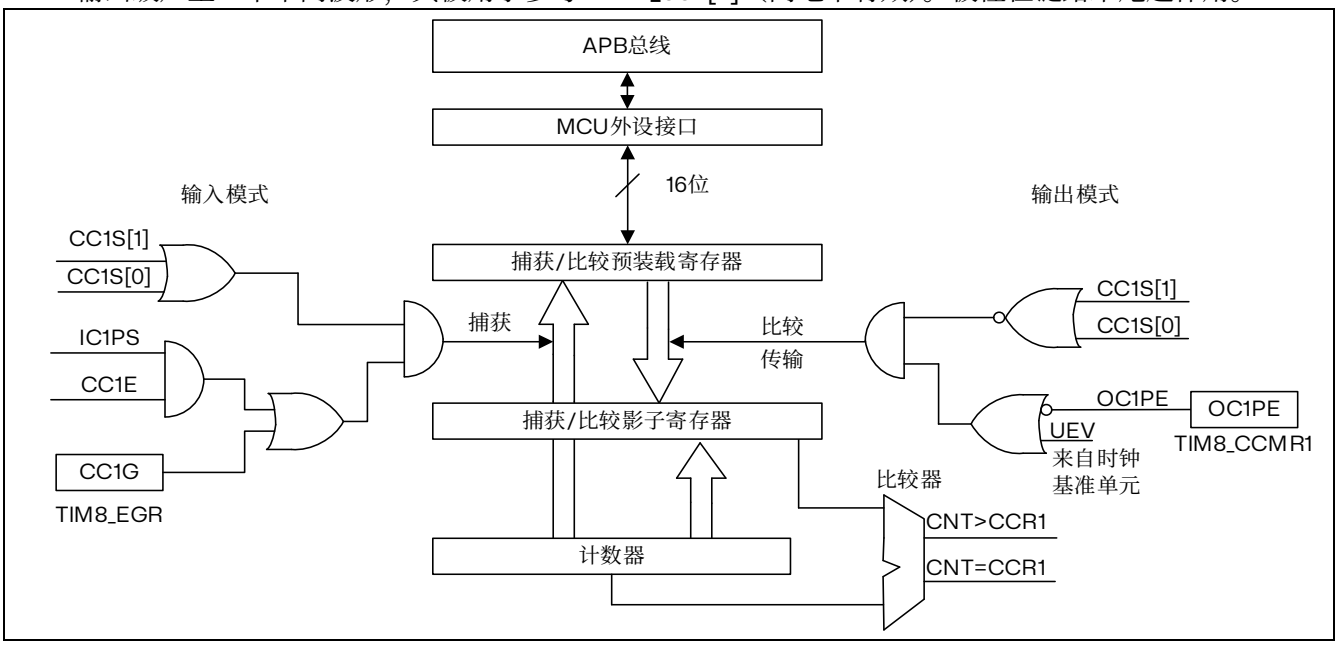


图 13.26 捕获/比较通道 1 的主电路

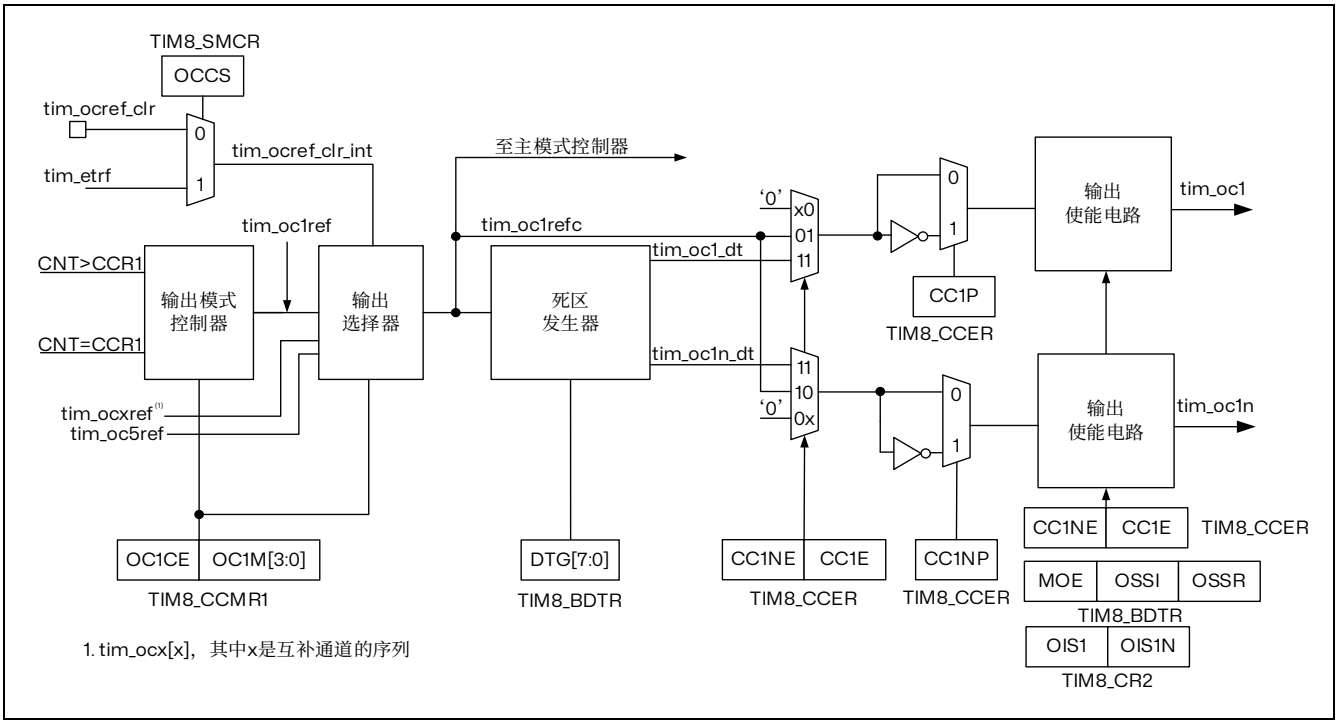


图 13.27 捕获/比较通道的输出阶段（通道 1，通道 2、3 和 4 同上）

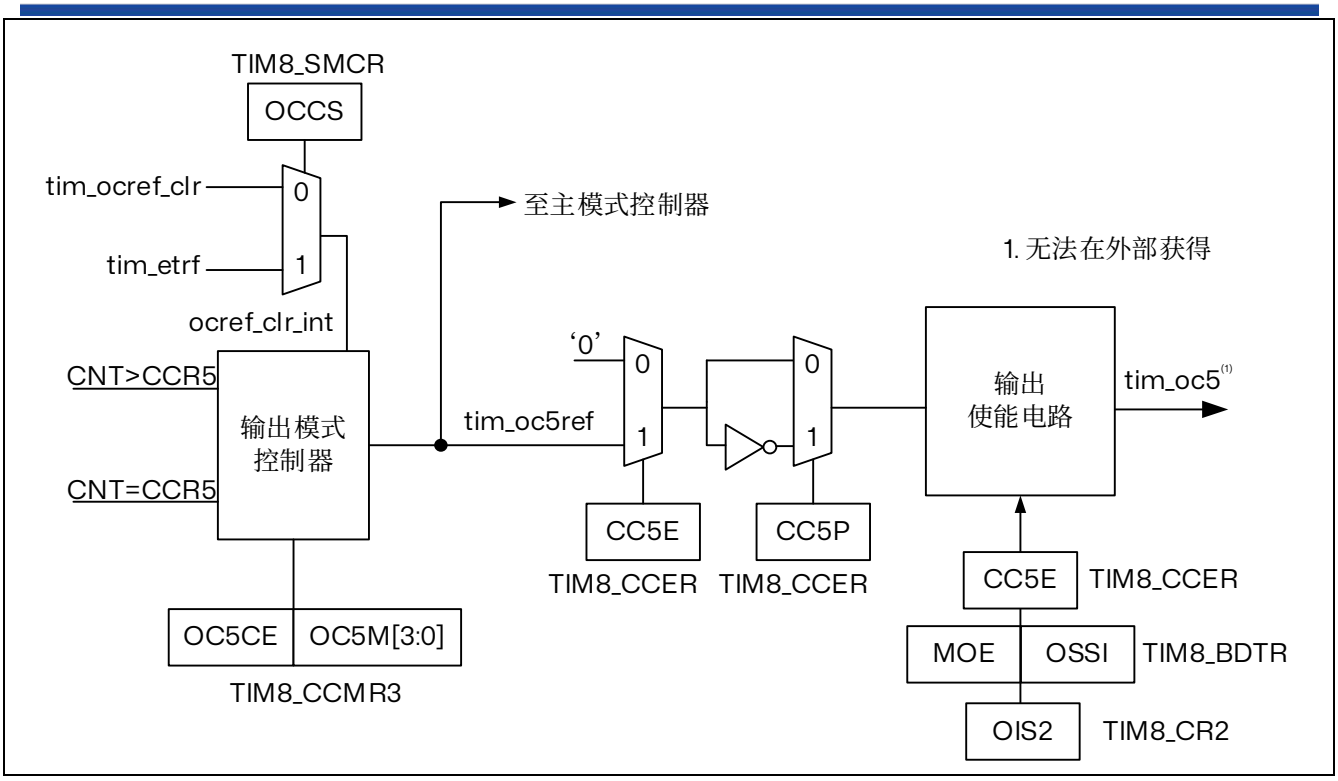


图 13.28 捕获/比较通道的输出阶段（通道 5，通道 6 同上）

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。始终可通过读写操作访问预装载寄存器。

在比较模式下，预装载寄存器的内容将被复制到影子寄存器中，然后将影子寄存器的内容与计数器进行比较。

13.3.9 强制输出模式

在输出模式（TIM8_CCMRx 寄存器中的 CCxS 位 = 00）下，可直接由软件将每个输出比较信号（tim_ocxref 和 tim_ocx/tim_ocxn）强制设置为有效电平或无效电平，而无需考虑输出比较寄存器和计数器之间的任何比较结果。

要将输出比较信号(tim_ocxref/tim_ocx)强制设置为有效电平，只需向相应 TIM8_CCMRx 寄存器中的 OCxM 位写入 0101。tim_ocxref 进而强制设置为高电平（tim_ocxref 始终为高电平有效），同时 tim_ocx 获取 CCxP 极性位的相反值。

例如：CCxP=0（tim_ocx 高电平有效）=> tim_ocx 强制设置为高电平。

通过向 TIM8_CCMRx 寄存器中的 OCxM 位写入 100，可将 tim_ocxref 信号强制设置为低电平。无论如何，TIM8_CCRx 影子寄存器与计数器之间的比较仍会执行，而且允许将标志置 1。因此可发送相应的中断请求。下面的输出比较模式一节对此进行了介绍。

13.3.10 输出比较模式

此功能用于控制输出波形，或指示已经过某一时间段。通道 1 至 4 可以输出，而通道 5 和 6 仅可在微控制器内部使用（例如，用于复合波形生成或 ADC 触发）。

当捕获/比较寄存器与计数器之间相匹配时，输出比较功能：

- 将为相应的输出引脚分配一个可编程值，该值由输出比较模式（TIM8_CCMRx 寄存器中的 OCxM 位）和输出极性（TIM8_CCER 寄存器中的 CCxP 位）定义。匹配时，输出引脚既可保持

其电平（OCXM=000），也可设置为有效电平(OCXM=001)、无效电平(OCXM=010)或进行翻转(OCxM=011)。

- 将中断状态寄存器中的标志置 1（TIM8_SR 寄存器中的 CCxIF 位）。
- 如果相应中断使能位（TIM8_DIER 寄存器中的 CCXIE 位）置 1，将生成中断。

使用 TIM8_CCMRx 寄存器中的 OCxPE 位，可将 TIM8_CCRx 寄存器配置为带或不带预装载寄存器。

在输出比较模式下，更新事件 UEV 对 tim_ocxref 和 tim_ocx 出毫无影响。同步的精度可以达到计数器的一个计数周期。输出比较模式也可用于输出单脉冲（在单脉冲模式下）。

步骤：

1. 选择计数器时钟（内部、外部、预分频器）。
2. 在 TIM8_ARR 和 TIM8_CCRx 寄存器中写入所需数据。
3. 如果要生成中断请求，则需将 CCxIE 位置 1。
4. 选择输出模式。例如：
   — 当 CNT 与 CCRx 匹配时，写入 OCxM =0 011 以翻转 tim_ocx 输出引脚
   — 写入 OCxPE = 0 以禁止预装载寄存器
   — 写入 CCxP = 0 以选择高电平有效极性
   — 写入 CCxE = 1 以使能输出
5. 通过将 TIM8_CR1 寄存器中的 CEN 位置 1 来使能计数器。

可通过软件随时更新 TIM8_CCRx 寄存器以控制输出波形，前提是未使能预加载寄存器（OCxPE="0"，否则仅当发生下一个更新事件 UEV 时，才会更新 TIM8_CCRx 影子寄存器）。下图给出了一个示例。

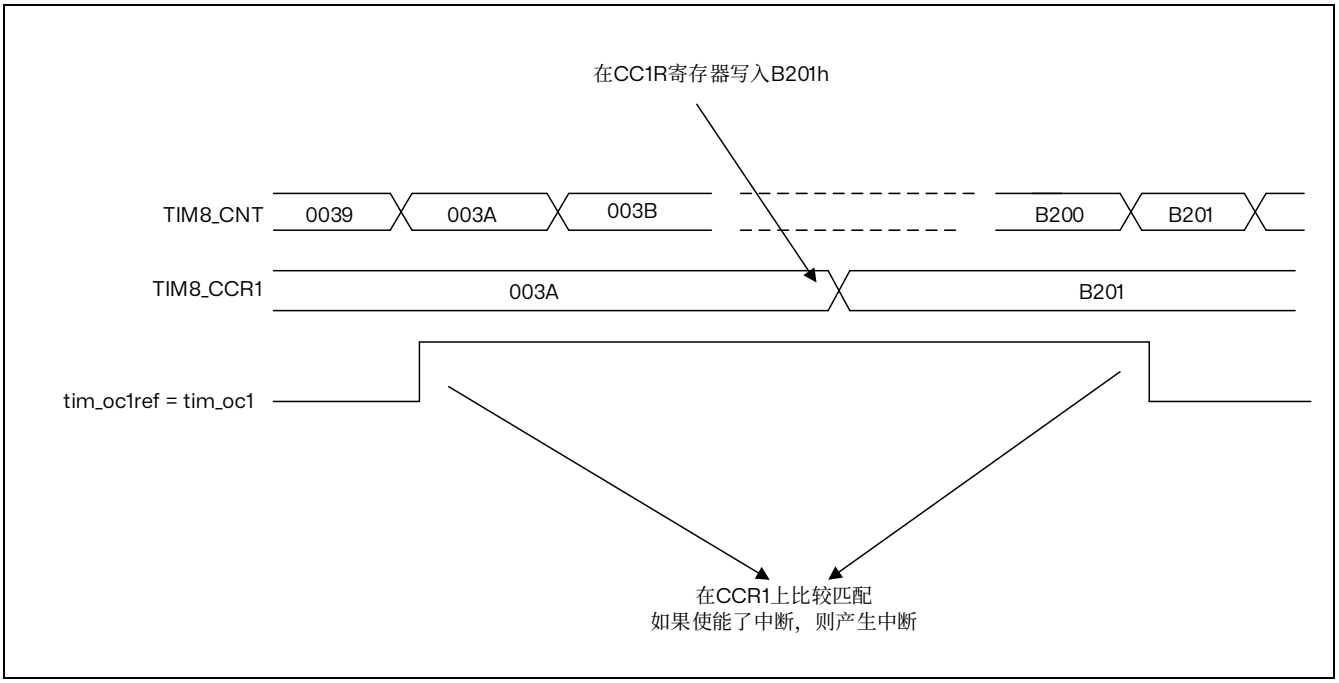


图 13.29 输出比较模式，翻转 OC1

13.3.11 PWM 模式

脉冲宽度调制模式可以生成一个信号，该信号频率由 TIM8_ARR 寄存器值决定，其占空比则由 TIM8_CCRx 寄存器值决定。

通过向 TIM8_CCMRx 寄存器中的 OCxM 位写入“0110”（PWM 模式 1）或“0111”（PWM 模式

2），可以独立选择各通道（每个 OCx 输出对应一个 PWM）的 PWM 模式。必须通过将 TIM8_CCMRx 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器，最后通过将 TIM8_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器（在递增计数或中央对齐模式下）。

由于只有在发生更新事件时预装载寄存器才会传送到影子寄存器，因此启动计数器之前，必须通过将 TIM8_EGR 寄存器中的 UG 位置 1 来初始化所有寄存器。

tim_ocx 极性可使用 TIM8_CCER 寄存器的 CCxP 位来编程。既可以设为高电平有效，也可以设为低电平有效。通过 CCxE、CCxNE、MOE、OSSI 和 OSSR 位（TIM8_CCER 和 TIM8_BDTR 寄存器）的组合使能 OCx 输出。有关详细信息，请参见 TIM8_CCER 寄存器说明。

在 PWM 模式（1 或 2）下，TIM8_CNT 总是与 TIM8_CCRx 进行比较，以确定是 TIM8_CCRx ≤ TIM8_CNT 还是 TIM8_CNT ≤ TIM8_CCRx（取决于计数器计数方向）。

根据 TIM8_CR1 寄存器中的 CMS 位状态，定时器能够产生边沿对齐模式或中央对齐模式的 PWM 信号。

PWM 边沿对齐模式

- 递增计数配置

当 TIM8_CR1 寄存器中的 DIR 位为低时执行递增计数。请参见递增计数模式一节。

以下以 PWM 模式 1 为例。只要 TIM8_CNT < TIM8_CCRx，PWM 参考信号 tim_ocxref 便为高电平，否则为低电平。如果 TIM8_CCRx 中的比较值大于自动重载值（TIM8_ARR 中），则 tim_ocxref 保持为“1”。如果比较值为 0，则 tim_ocxref 保持为“0”。下图举例介绍边沿对齐模式的一些 PWM 波形 (TIM8_ARR=8)。

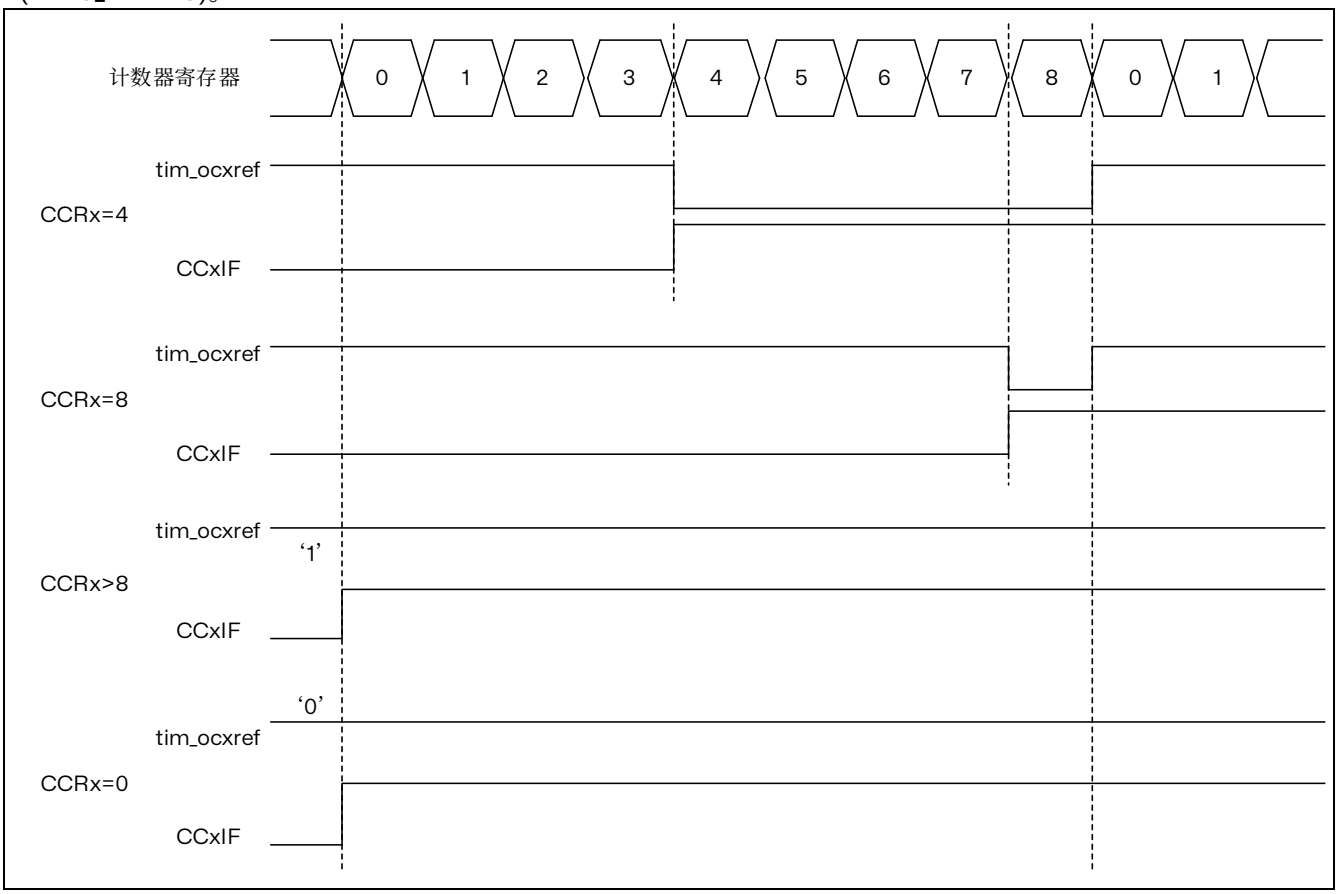


图 13.30 边沿对齐的 PWM 波形（ARR=8）

- 递减计数配置

当 TIM8_CR1 寄存器中的 DIR 位为高时执行递减计数。请参见递减计数模式一节。

在 PWM 模式 1 下，只要 TIM8_CNT > TIM8_CCRx，参考信号 tim_ocxref 即为低电平，否则其为高电平。如果 TIM8_CCRx 中的比较值大于 TIM8_ARR 中的自动重载值，tim_ocxref 保持为“1”。此模式下不可能产生 0%的 PWM 波形。

PWM 中央对齐模式

当 TIM8_CR1 寄存器中的 CMS 位不为“00”（其余所有配置对 tim_ocxref/tim_ocx 信号具有相同的作用），中央对齐模式生效。根据 CMS 位的配置，可以在计数器递增计数、递减计数或同时递增和递减计数时将比较标志置 1。TIM8_CR1 寄存器中的方向位(DIR)由硬件更新，不得通过软件更改。请参见中央对齐模式（递增/递减计数）。

下图显示了中央对齐模式的 PWM 波形，在此例中：

- TIM8_ARR=8
- PWM 模式为 PWM 模式 1,
- 在根据 TIM8_CR1 寄存器中 CMS=01 而选择的中央对齐模式 1 下，当计数器递减计数时，比较标志置 1。

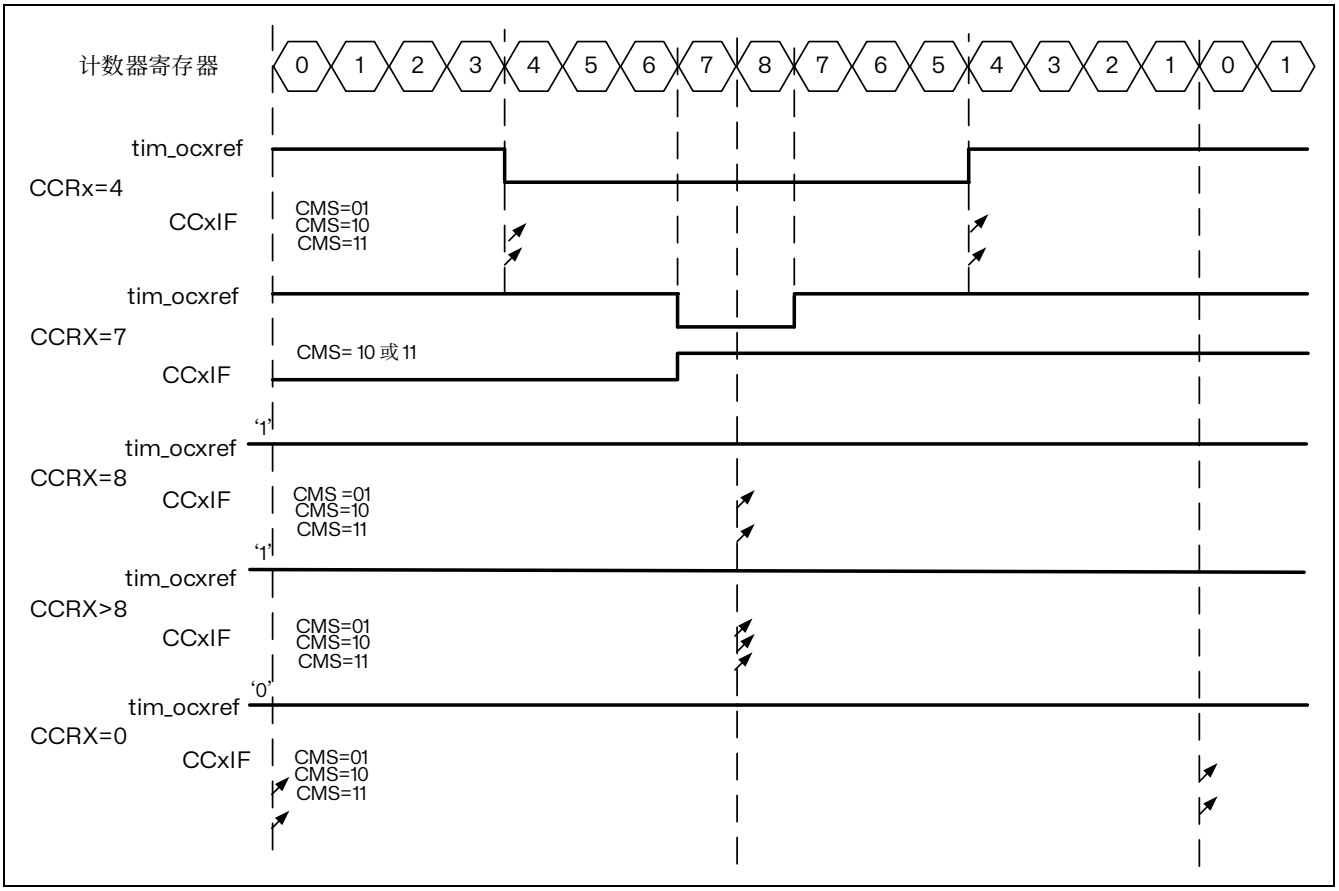


图 13.31 中央对齐的 PWM 波形（APR=8）

关于使用中央对齐模式的提示：

- 启动中央对齐模式时将使用当前的递增/递减计数配置。这意味着计数器将根据写入 TIM8_CR1 寄存器中 DIR 位的值进行递增或递减计数。此外，不得同时通过软件修改 DIR 和 CMS 位。

- 不建议在运行中央对齐模式时对计数器执行写操作，否则将发生意想不到的结果。尤其是：
  — 如果写入计数器中的值大于自动重载值(TIM8_CNT>TIM8_ARR)，计数方向不会更新。例如，如果计数器之前递增计数，则继续递增计数。
  — 如果向计数器写入 0 或 TIM8_ARR 的值，计数方向会更新，但不生成更新事件 UEV。
- 使用中央对齐模式最为保险的方法是：在启动计数器前通过软件生成更新（将 TIM8_EGR 寄存器中的 UG 位置 1），并且不要在计数器运行过程中对其执行写操作。

抖动模式

通过使能抖动模式，使用 TIM8_CR1 寄存器中的 DITHEN 位，可以提高 PWM 模式的有效分辨率。这既适用于 CCR（用于增加占空比分辨率），也适用于 ARR（用于增加 PWM 频率分辨率）。

工作原理是在 16 个连续的 PWM 周期中，以预定义的模式使实际的 CCR（或 ARR）值略有变化（增加或减少一个定时器时钟周期）。这使得占空比或 PWM 期间的平均值提高了 16 倍分辨率。下图展示了应用于 4 个连续 PWM 周期的抖动原理。

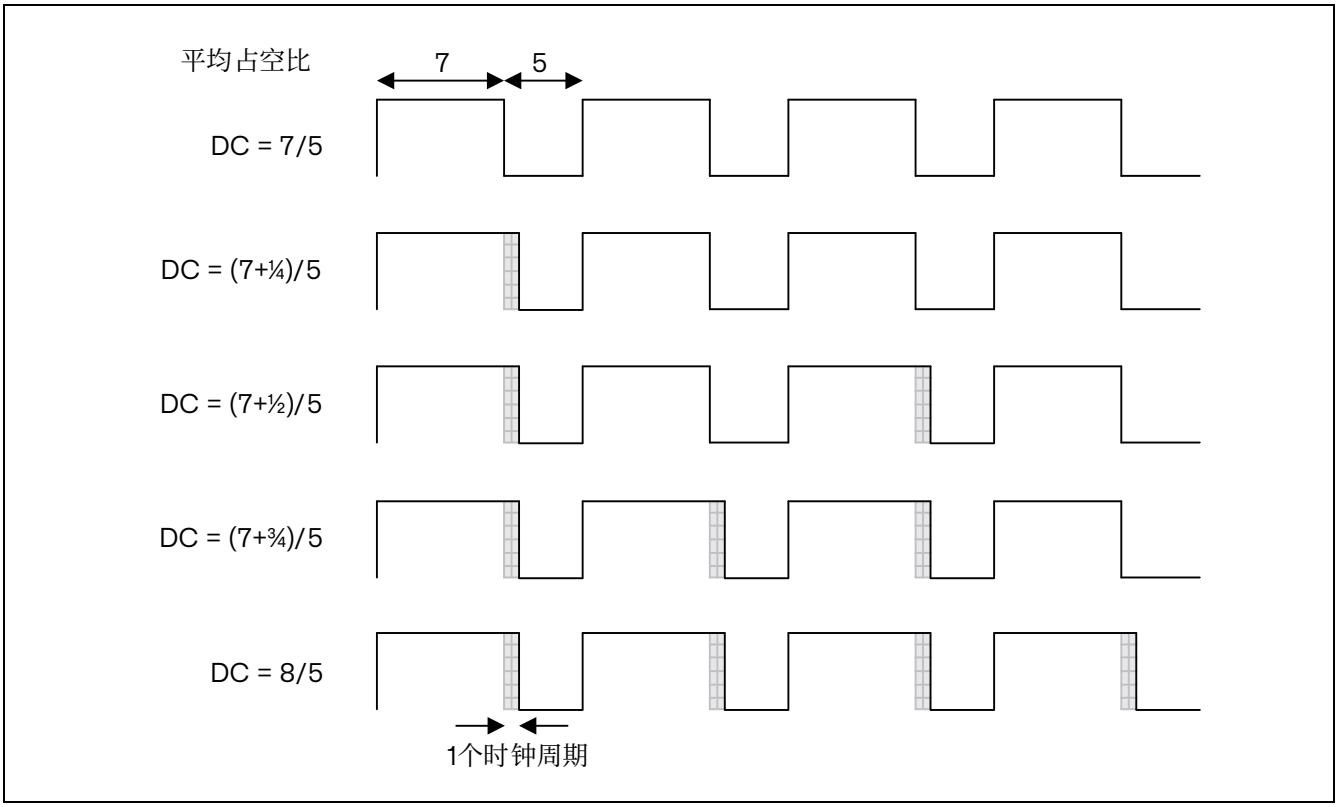


图 13.32 抖动原理

当抖动模式被使能时，寄存器编码将如下所示进行变化（参见下图示例）

- 4 个最低有效位（LSB）用于编码增强的分辨率部分（小数部分）
- 最高有效位（MSB）左移至 19:4 位，用于编码基本值

注意：如果设置或清除 DITHEN *位，* ARR *和* CCR *值将自动更新。（例如，如果* ARR=0x05 *且* DITHEN=0，*它将更新为* ARR=0x50 *且* DITHEN=1*）。*

1. 必须重置 CEN 和 ARPE 位
2. ARR[3:0]位必须重置
3. DITHEN 位必须重置
4. CCIF 标志必须清除

5. 可以设置 CEN 位（最终设置 ARPE=1）。

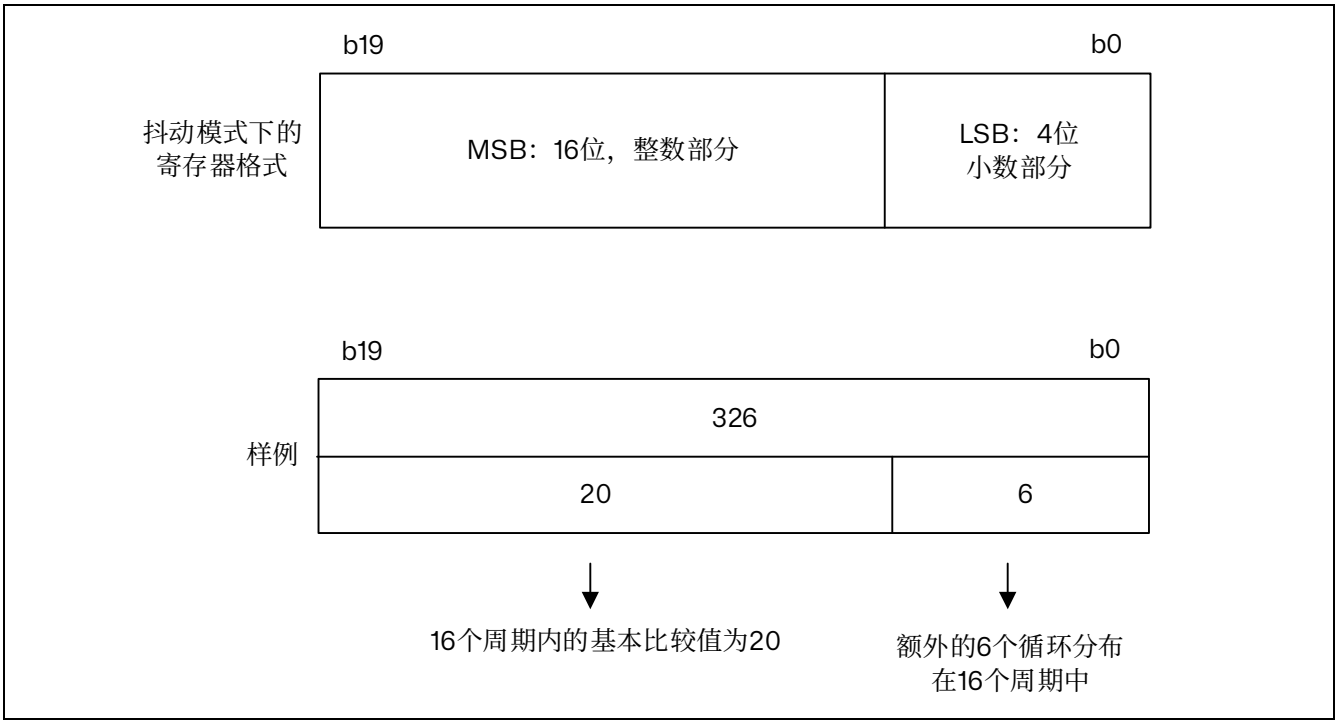


图 13.33 抖动模式下的数据格式和寄存器编码

最小频率由以下公式给出：

$$\text{分辨率} = \frac{\text{FTim}}{\text{Fpwm}} \Rightarrow \text{FpwmMin} = \frac{\text{FTim}}{\text{最大分辨率}}$$

$$\text{禁用抖动模式：FpwmMin} = \frac{\text{FTim}}{65536}$$

$$\text{使能抖动模式：FpwmMin} = \frac{\text{FTim}}{65535 + \frac{15}{16}}$$

*注意：在抖动模式下，*TIM8_ARR *和*TIM8CCRy *的最大值限制为*0xFFFEF*（对应于整数部分的*65534 *和抖动部分的*15*）。*

如下图所示，无论 PWM 频率如何，抖动模式都可以增加 PWM 分辨率。

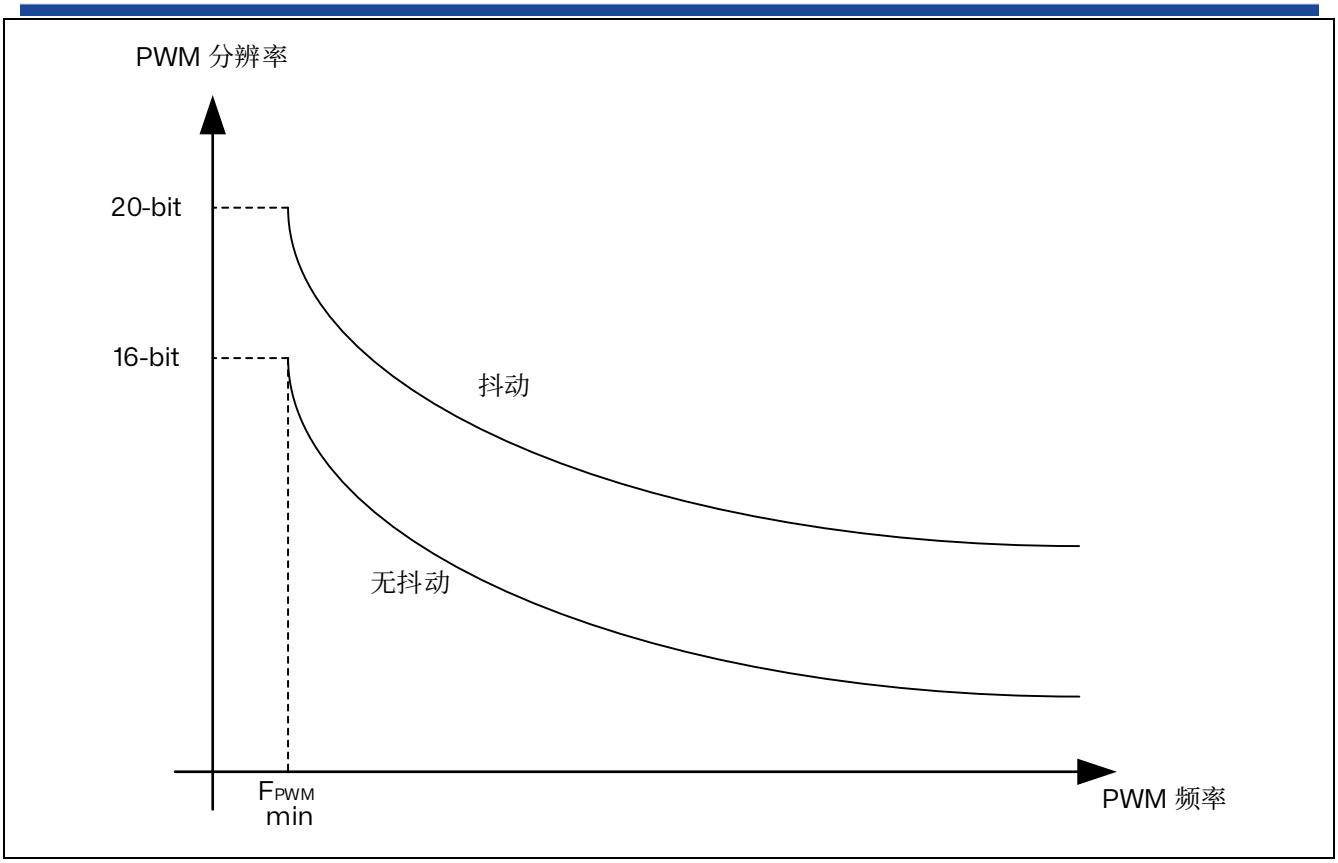


图 13.34 PWM 分辨率 vs 频率

如下图所示，占空比和/或周期的变化分散在 16 个连续周期。

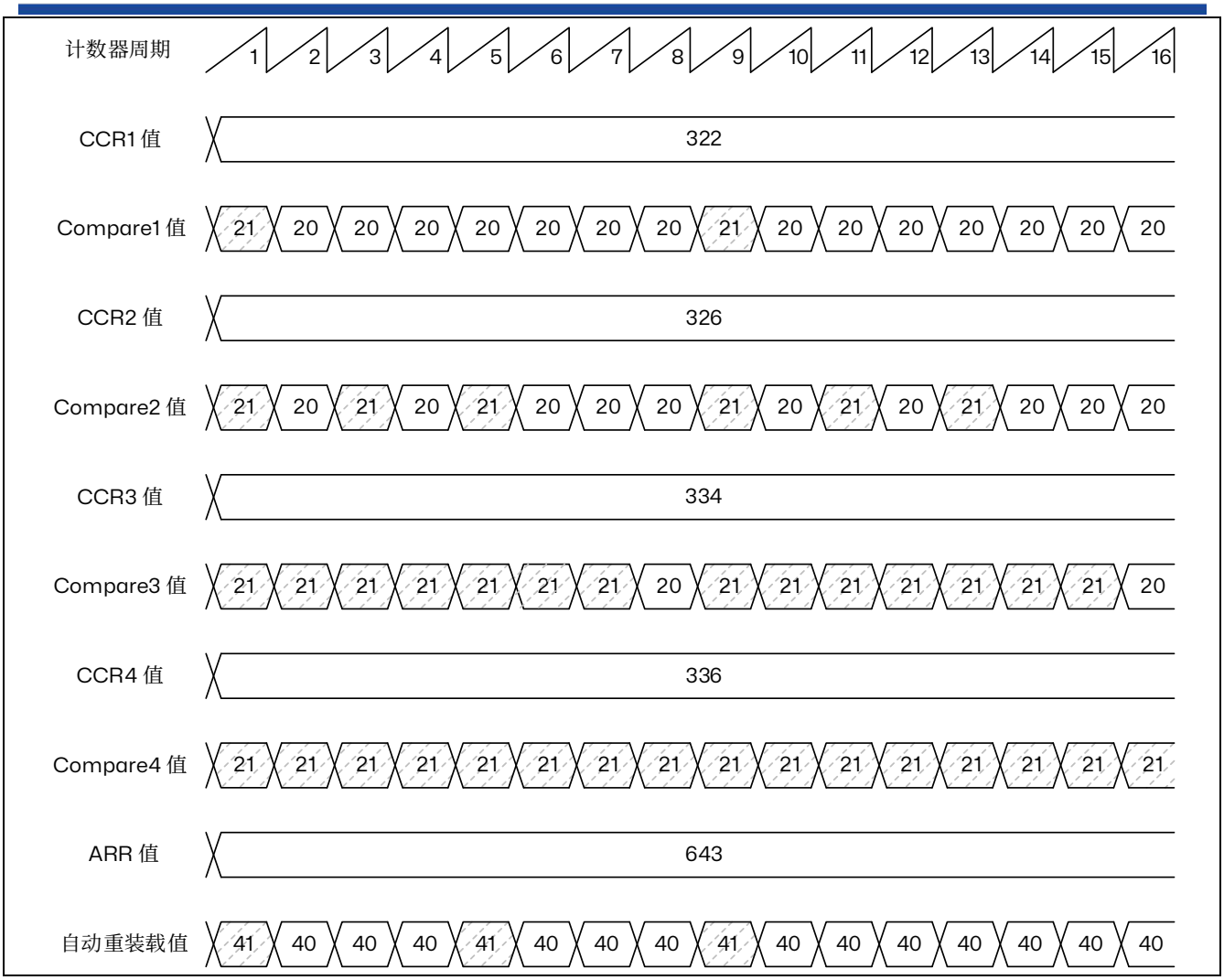


图 13.35 PWM 抖动模式

自动重新加载和比较值的增量按照下表中描述的特定模式分布。抖动序列是为了使增量分布尽可能均匀并最大限度地减少整体波动。

表 13.9 CCR 和 ARR 寄存器的变化抖动模式

LSB 值	PWM 周期															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0001	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0010	+1	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-
0011	+1	-	-	-	+1	-	-	-	+1	-	-	-	-	-	-	-
0100	+1	-	-	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0101	+1	-	+1	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0110	+1	-	+1	-	+1	-	-	-	+1	-	+1	-	+1	-	-	-
0111	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	-	-
1000	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1001	+1	+1	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1010	+1	+1	+1	-	+1	-	+1	-	+1	+1	+1	-	+1	-	+1	-
1011	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	-	+1	-
1100	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1101	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1110	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	+1	+1	+1	+1	-
1111	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	-

在中央对齐 PWM 模式下（TIM8_CR1 寄存器中的 CMS 位不为“00”），也提供了抖动模式。在这种模式下，抖动模式考虑到递增递减计数阶段，应用于 8 个连续的 PWM 周期，如下图所示。

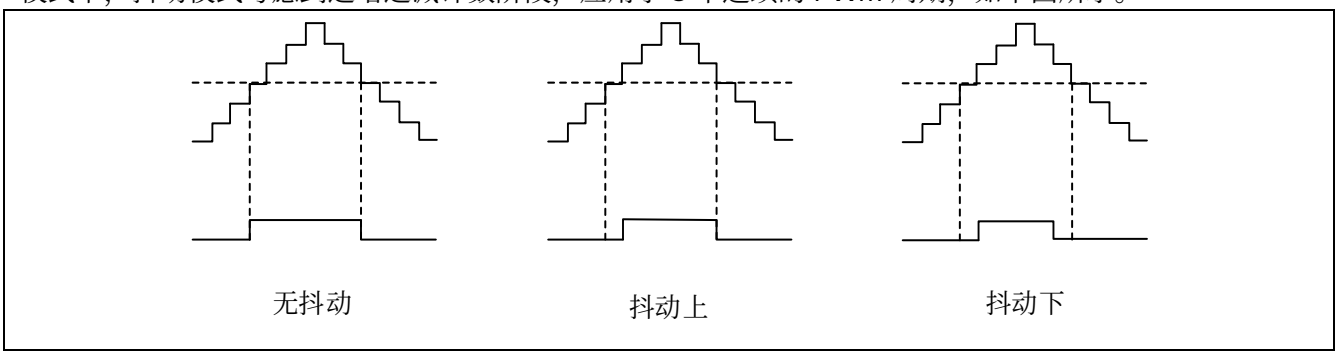


图 13.36 中央对齐 PWM 模式下抖动对占空比的影响

下表显示了如何在中央对齐 PWM 模式下添加抖动模式。

表 13.10 CCR 寄存器在中央对齐 PWM 模式下的变化抖动模式

LSB 值	PWM 周期															
	1		2		3		4		5		6		7		8	
	上	下	上	下	上	下	上	下	上	下	上	下	上	下	上	下
0000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0001	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0010	+1	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-
0011	+1	-	-	-	+1	-	-	-	+1	-	-	-	-	-	-	-
0100	+1	-	-	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0101	+1	-	+1	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0110	+1	-	+1	-	+1	-	-	-	+1	-	+1	-	+1	-	-	-
0111	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	-	-
1000	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1001	+1	+1	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1010	+1	+1	+1	-	+1	-	+1	-	+1	+1	+1	-	+1	-	+1	-
1011	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	-	+1	-
1100	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1101	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1110	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	+1	+1	+1	+1	-
1111	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	-

13.3.12 非对称 PWM 模式

非对称 PWM 模式 1/非对称 PWM 模式 2

非对称模式允许生成两个中央对齐的 PWM 信号，并具有可编程的相位偏移。虽然频率由 TIM8_ARR 寄存器的值决定，但占空比和相位偏移由一对 TIM8_CCRx 寄存器确定。一个寄存器控制 PWM 在计数上升期间，第二个寄存器在计数下降期间，这样 PWM 每半个 PWM 周期调整一次：

— tim_oc1refc（或 tim_oc2refc）由 TIM8_CCR1 和 TIM8_CCR2 控制

— tim_oc3refc（或 tim_oc4refc）由 TIM8_CCR3 和 TIM8_CCR4 控制

通过将“1110”（非对称 PWM 模式 1）或“1111”（非对称 PWM 模式 2）写入 TIM8_CCMRx 寄存器的 OCxM 位，可以在两个通道上独立选择非对称 PWM 模式（每个 CCR 寄存器对一个 tim_ocx 输出）。

*注意：为了兼容性原因，*OCxM[3:0]*位字段被分为两个部分，最高有效位与最低*3 *位不连续。*

当某个通道用作非对称 PWM 通道时，其互补通道也可以使用。例如，如果 tim_oc1refc 信号在通道 1 上生产（非对称 PWM 模式 1），可以在通道 2 上输出 tim_oc1refc 信号，或者产生非对称 PWM 模式 1 的结果 tim_oc2refc 信号。

下图标识使用非对称 PWM 模式生成信号的示例（通道 1 到 4 配置为非对称 PWM 模式 2）。与死区时间生成器一起使用，可以控制全桥移相 DC 到 DC 转换器。

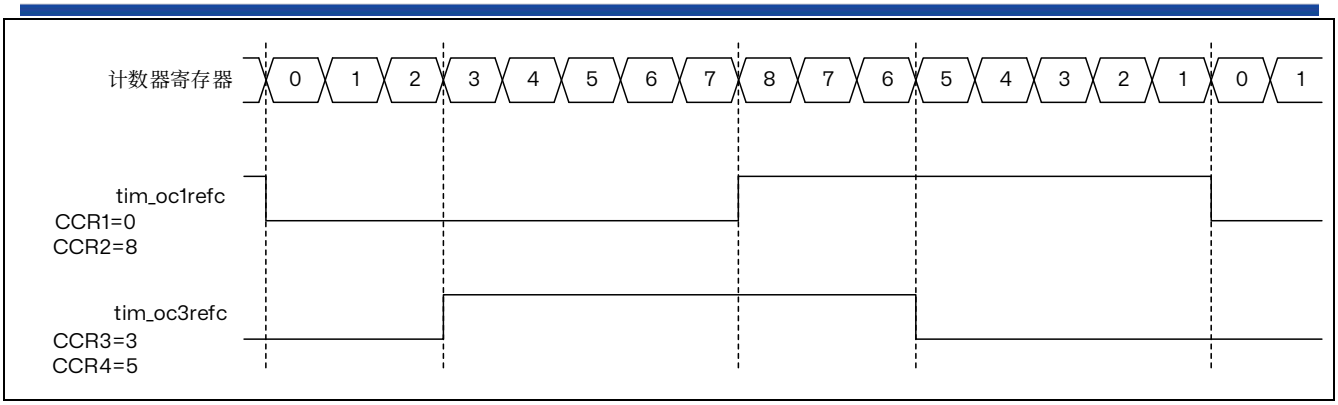


图 13.37 生成两个具有 50%占空比的相位偏移 PWM 信号

PWM 边沿对齐模式

- 向上计数配置

当 TIM8_CR1 寄存器中的 DIR 位为低的时候执行向上计数。当 TIM8_CR1 寄存器中的 CCRxN 位为‘1’时配置位非对称模式。下面是一个 PWM 模式 1 的例子。

当 TIM8_CCRx<=TIM8_CNT<TIM8_CCRxN 时，PWM 参考信号 OCxREF 为高，否则为低。如果 TIM8_CCRx 和 TIM8_CCRxN 中的比较值大于自动重装载值（TIM8_ARR），则 OC1REF 保持为‘1’。如果 TIM8_CCRx 中的比较值大于等于 TIM8_CCRxN 中的比较值或者比较值都为 0，则 OCxREF 保持为‘0’。下图为 TIM8_ARR=12 时边沿对齐的 PWM 波形实例。

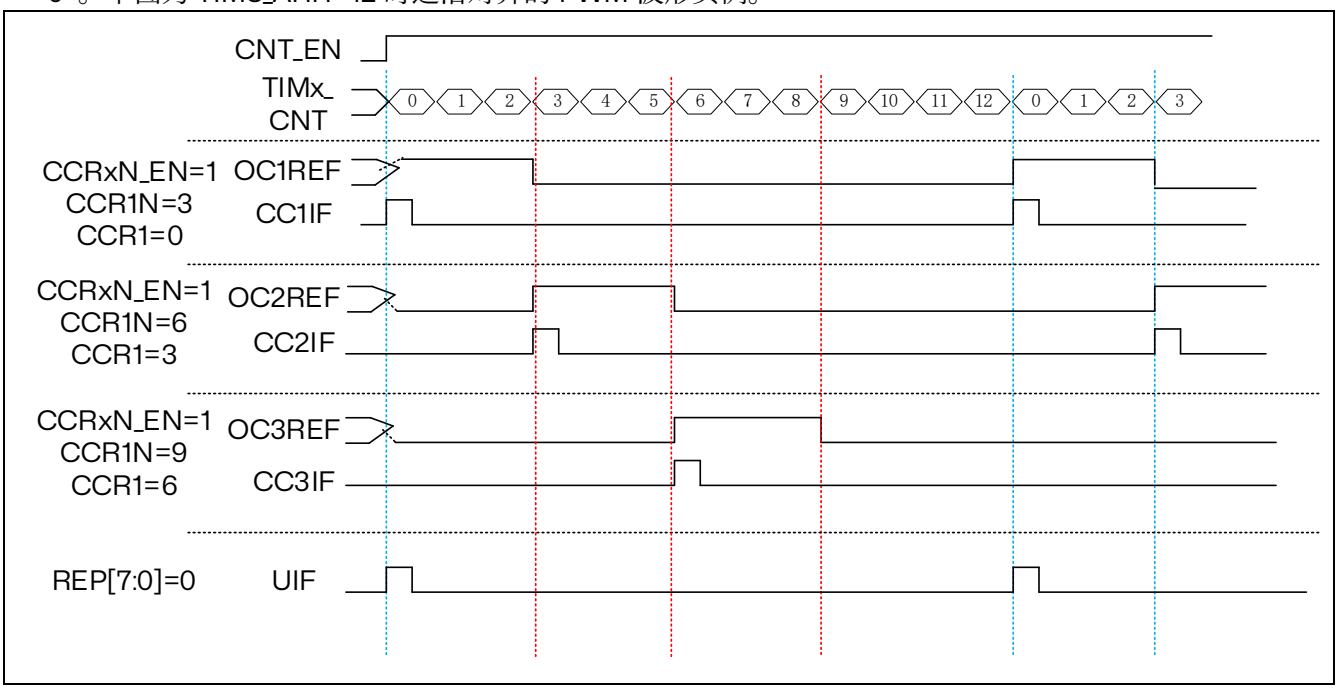


图 13.38 边沿对齐向上计数的 PWM 波形（ARR=12）

- 向下计数的配置

当 TIM8_CR1 寄存器中的 DIR 位为低的时候执行向上计数。当 TIM8_CR1 寄存器中的 CCRxN 位为‘1’时配置位非对称模式。下面是一个 PWM 模式 1 的例子。当 TIM8_CCRxN<= TIM8_CNT< TIM8_CCRx 时，PWM 参考信号 OCxREF 为高，否则为低。如果 TIM8_CCRx 和 TIM8_CCRxN 中的比较值大于自动重装载值（TIM8_ARR），则 OCxREF 保持为‘1’。如果 TIM8_CCRx 中的比较值小于等于

TIM8_CCRxN 中的比较值或者比较值都为 0，则 OCxREF 保持为‘0’。

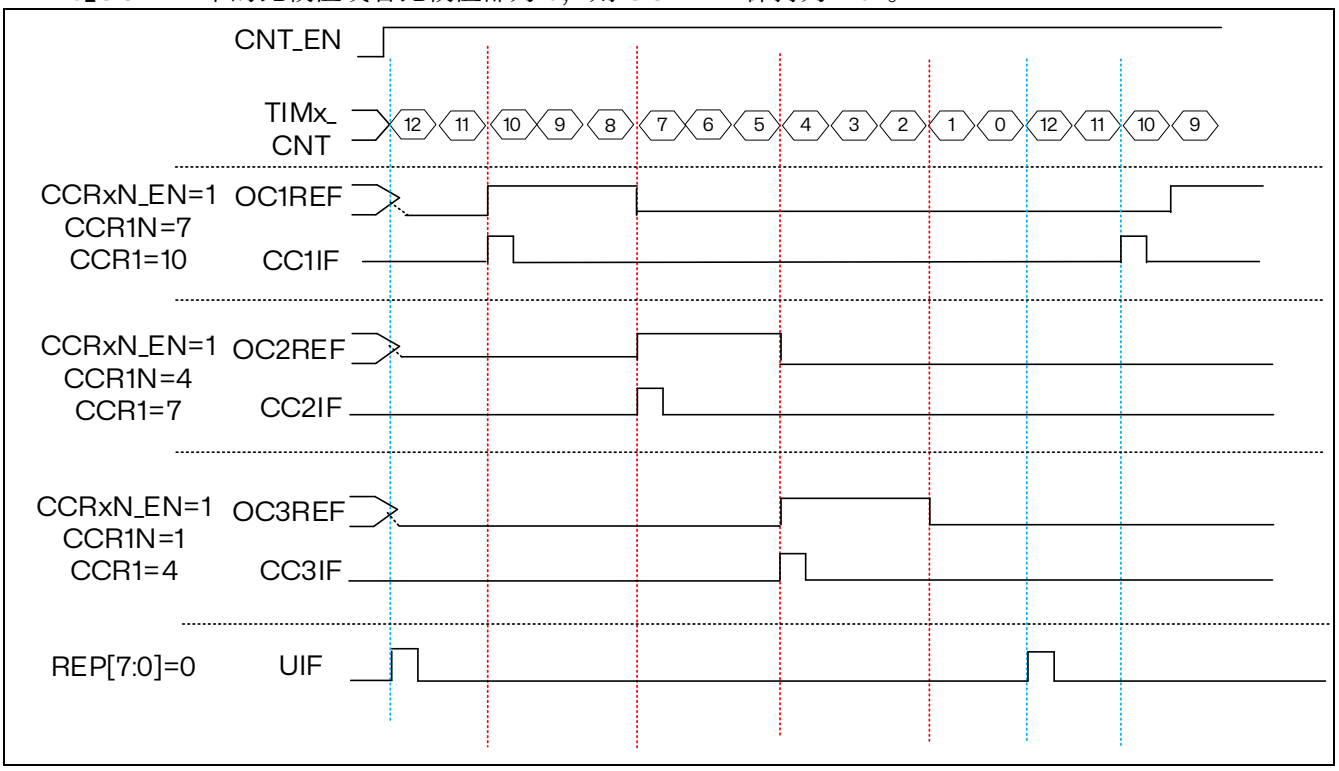


图 13.39 边沿对齐向下计数的 PWM 波形（ARR=12）

PWM 中央对齐模式

当 TIM8_CR1 寄存器中的 CMS 位不为‘00’时为中央对齐模式（所有其他的配置对 OCxREF/OCx 信号都有相同的作用）。根据不同的 CMS 位设置，比较标志可以在计数器向上计数时被置 1、在计数器向下计数时被置 1、或在计数器向上和向下计数时被置 1。TIM8_CR1 寄存器中的计数方向位（DIR）由硬件更新，不要用软件修改它。下图给出了一些中央对齐的 PWM 波形的例子

- TIM8_ARR=8
- PWM 模式 1
- TIM8_CR1 寄存器的 CMS=01，在中央对齐模式 1 下，当计数器向下计数时设置比较标志。

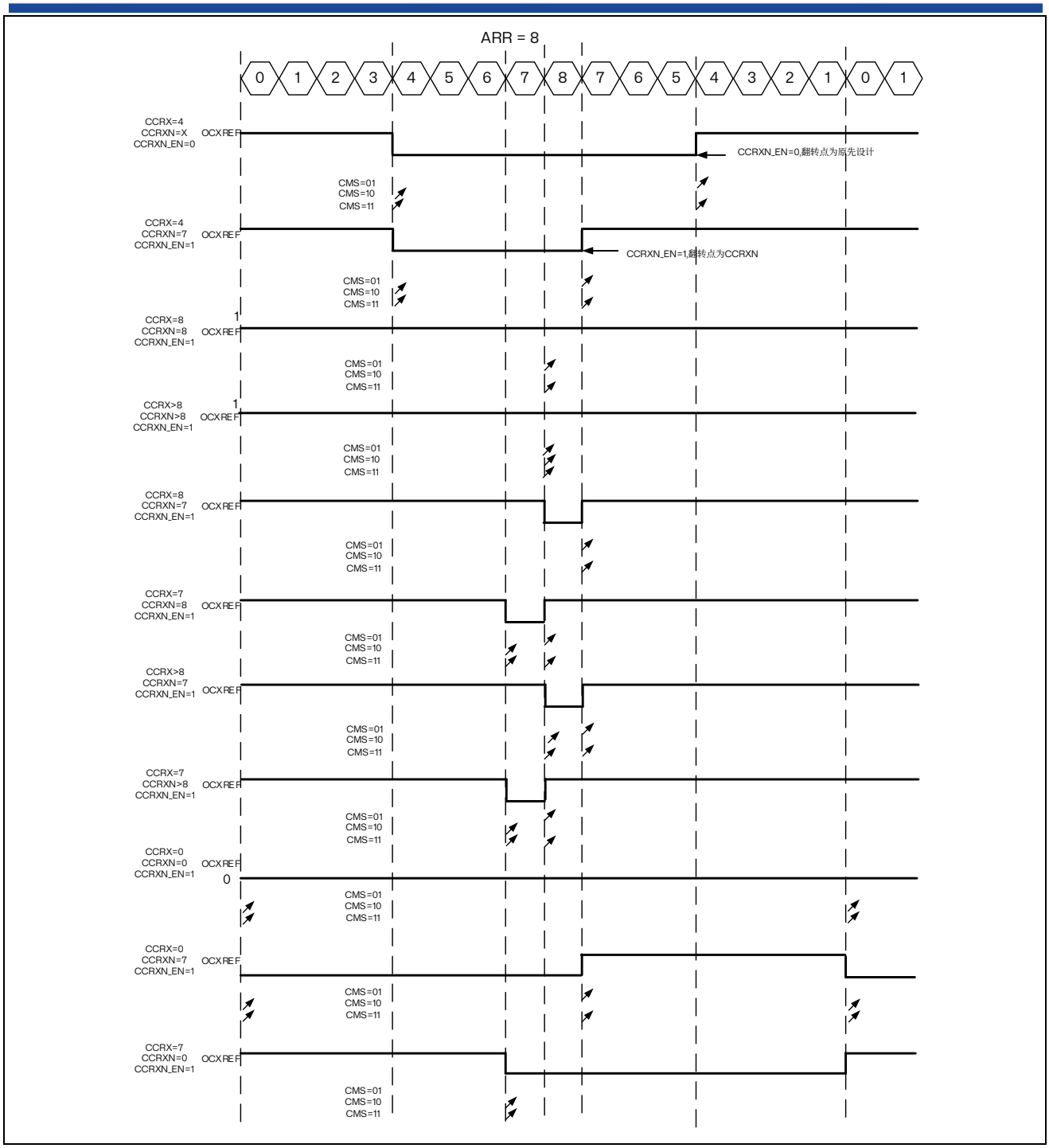


图 13.40 中央对齐的 PWM 波形（ARR=8）

13.3.13 组合 PWM 模式

组合 PWM 模式允许生成两个边缘对齐或中央对齐的 PWM 信号，各自脉冲之间的可编程延迟和相位偏移。虽然频率由 TIM8_ARR 寄存器的值决定，但占空比和延迟由两个 TIM8_CCRx 寄存器确定。结果信号 tim_ocxrefc 由两个参考 PWM 的 OR 或 AND 逻辑组合而成：

— tim_oc1refc（或 tim_oc2refc）由 TIM8_CCR1 和 TIM8_CCR2 控制

— tim_oc3refc（或 tim_oc4refc）由 TIM8_CCR3 和 TIM8_CCR4 控制

通过在 TIM8_CCMRx 寄存器的 OCxM 位中写入“1100”（组合 PWM 模式 1）或“1101”（组合

PWM 模式 2），可以在两个通道上独立选组合 PWM 模式（每个 CCR 寄存器对一个 tim_ocx 输出）。

当某个通道用作组合 PWM 通道时，其互补通道必须配置为相反的 PWM 模式（例如，一个通道为组合 PWM 模式 1，另一个通道为组合 PWM 模式 2）。

*注意：为了兼容性原因，*OCxM[3:0]*位字段被分为两个部分，最高有效位与最低*3 *位不连续。*

下图表示使用组合 PWM 模式生成信号的示例，该示例的配置如下：

— 通道 1 配置为组合 PWM 模式 2。
— 通道 2 配置为 PWM 模式 1。
— 通道 3 配置为组合 PWM 模式 2。
— 通道 4 配置为 PWM 模式 1。

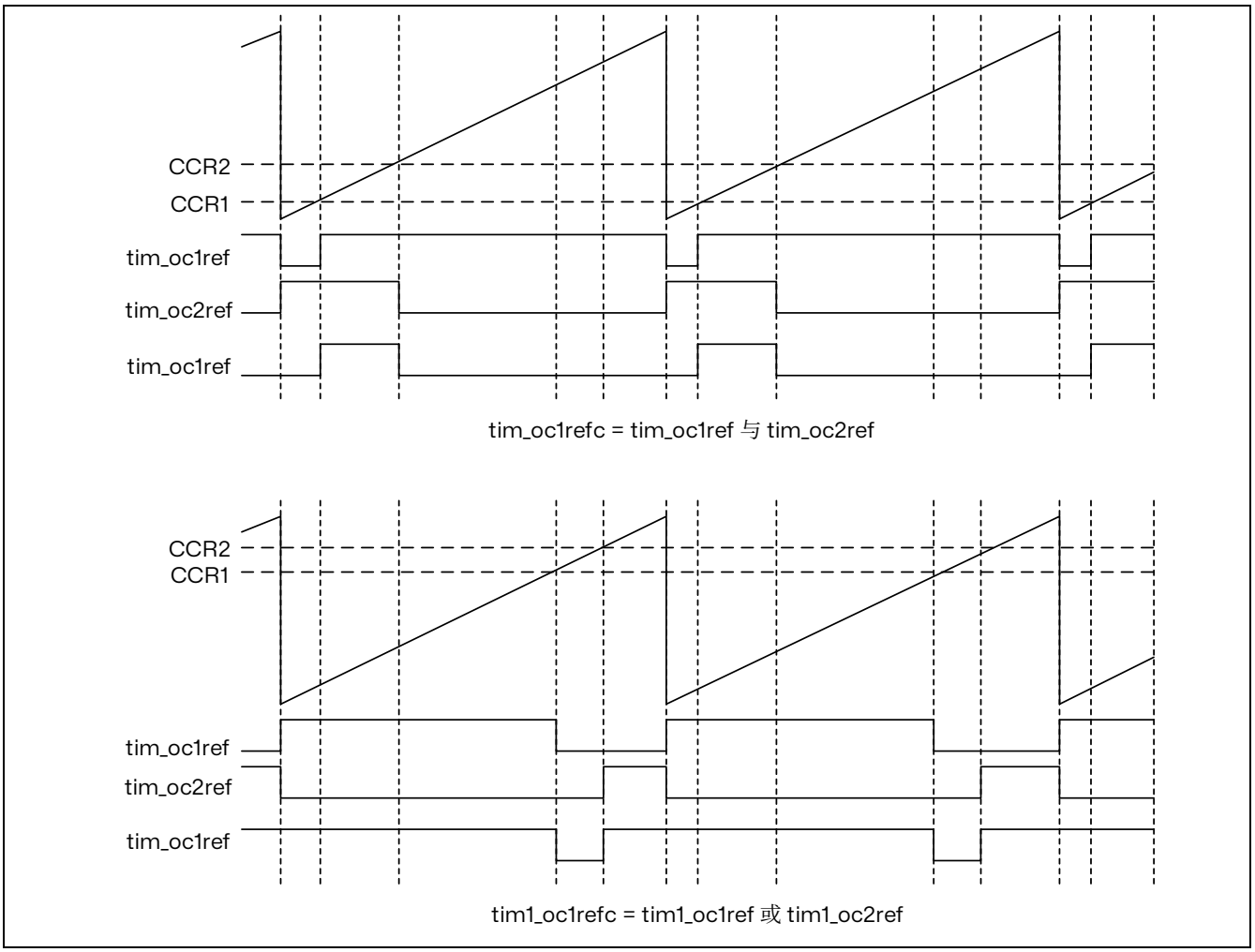


图 13.41 通道 1 和通道 3 的组合 PWM 模式

13.3.14 三相组合 PWM 模式

三相组合 PWM 模式允许生成一个到三个以脉冲中间的单个可编程信号 AND 为中央对齐的 PWM 信号。使用 tim_oc5ref 信号定义结果组合信号。TIM8_CCR5 中的 3 位 GC5C[3:1]允许选择 tim_oc5ref 与哪个参考信号组合。结果信号 tim_ocxrefc 由两个参考 PWM 的 AND 逻辑组合而成。

— 如果设置了 GC5C1，则 tim_oc1refc 由 TIM8_CCR1 和 TIM8_CCR5 控制
— 如果设置了 GC5C2，则 tim_oc2refc 由 TIM8_CCR2 和 TIM8_CCR5 控制

— 如果设置了 GC5C3，则 tim_oc3refc 由 TIM8_CCR3 和 TIM8_CCR5 控制

通过设置 3 位 GC5C[3:1]中的至少一位，可以在通道 1 到 3 上独立选择三相组合 PWM 模式。

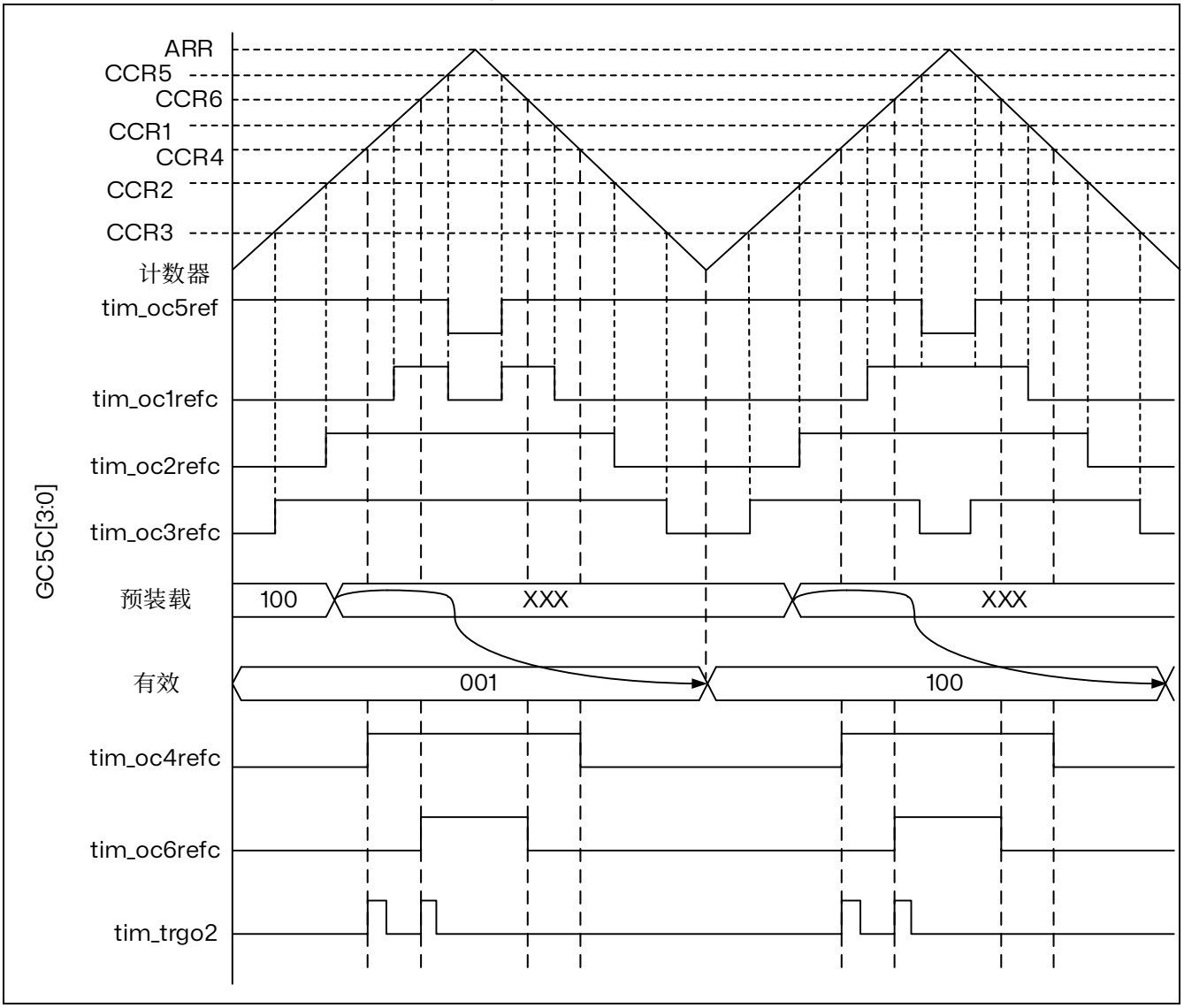


图 13.42 每周期具有多个触发脉冲的三相组合 PWM 信号

tim_trgo2 波形显示了 ADC 如何对给定的三相 PWM 信号进行同步。更多详细信息请参考本章后续小节：ADC 触发器。

13.3.15 互补输出和死区插入

高级控制定时器（TIM8）可以输出两路互补信号，并管理输出的关断与接通瞬间。

这段时间通常称为死区，用户必须根据与输出相连接的器件及其特性（电平转换器的固有延迟、开关器件产生的延迟...）来调整死区时间

每路输出可以独立选择输出极性（主输出 tim_ocx 或互补输出 tim_ocxn）。可通过对 TIM8_CCER 寄存器中的 CCxP 和 CCxNP 位执行写操作来完成极性选择。

互补信号 tim_ocx 和 tim_ocxn 通过以下多个控制位的组合进行激活：TIM8_CCER 寄存器中的 CCxE 和 CCxNE 位以及 TIM8_BDTR 和 TIM8_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位。更多详细信息，请参考表格：带刹车功能的互补输出通道 tim_ocx 和 tim_ocxn 的控制位。应当注

意，切换至 idle（MOE 下降到 0）的时刻，死区仍然有效。

CCxE 和 CCxNE 位同时置 1 并且 MOE 位置 1（如果存在刹车）时，将使能死区插入。TIM8_BDTR 寄存器中的 DTG[7:0]位用于控制所有通道的死区生成。将基于参考波形 tim_ocxref 生成 2 个输出 tim_ocx 和 tim_ocxn。如果 tim_ocx 和 tim_ocxn 为高电平有效：

- 输出信号 OCx 与参考信号相同，只是其上升沿相对参考上升沿存在延迟。
- 输出信号 OCxN 与参考信号相反，并且其上升沿相对参考下降沿存在延迟。

如果延迟时间大于有效输出（OCx 或 OCxN）的宽度，则不会产生相应的脉冲。

下图所示为死区发生器的输出信号与参考信号 OCxREF 之间的关系。（在这些示例中，假定 CCxP=0、CCxNP=0、MOE=1、CCxE=1 并且 CCxNE=1）

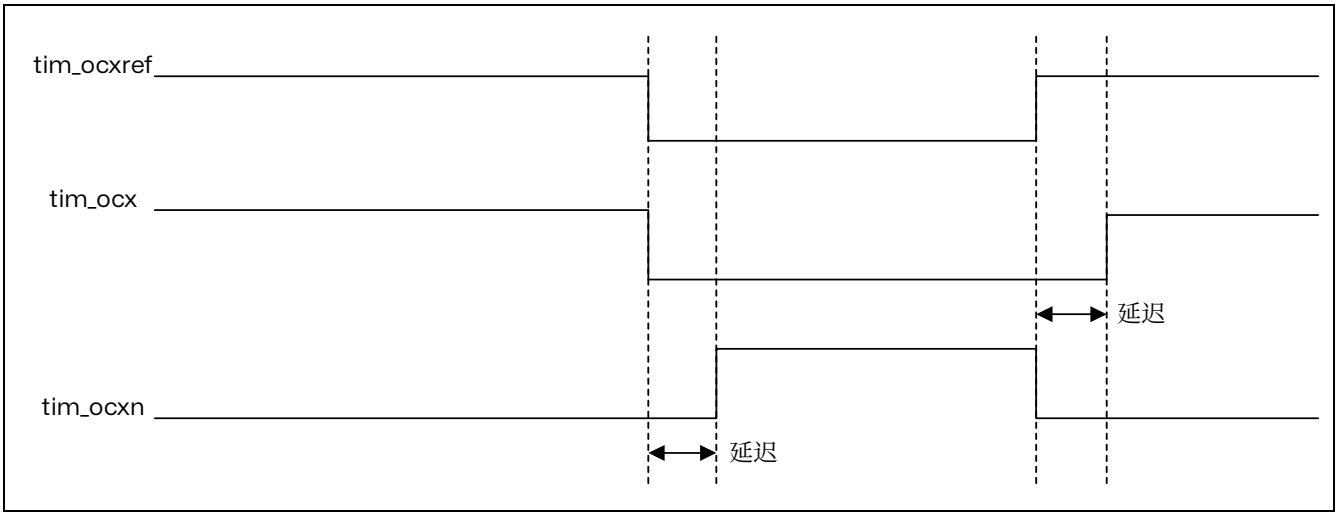


图 13.43 带死区插入的互补输出

TIM8_DTR2 中的 DTAE 位允许对参考信号上升沿和下降沿的死区值进行区分，如下图所示。

在非对称模式（DTAE=1）下，上升沿参考死区由 TIM8_BDTR 寄存器中的 DTG[7:0]位字段定义，而下降沿参考死区由 TIM8_DTR2 寄存器中的 DTGF[7:0]位字段定义。在使能计数器之前，必须写入 DTAE 位，并且在 CEN=1 时不得修改。

使用预加载机制可以在 PWM 操作期间动态更新死区值。当 TIM8_DTR2 寄存器中的 DTPE 位被设置时，死区位字段 DTG[7:0]和 DTGF[7:0]将被预加载。预加载值将在下一个更新事件上加载到活动寄存器中。

注意：如果 DTPE *位在计数器使能时被使能。自上次更新以来写入的新值将被丢弃，并使用先前的值。*

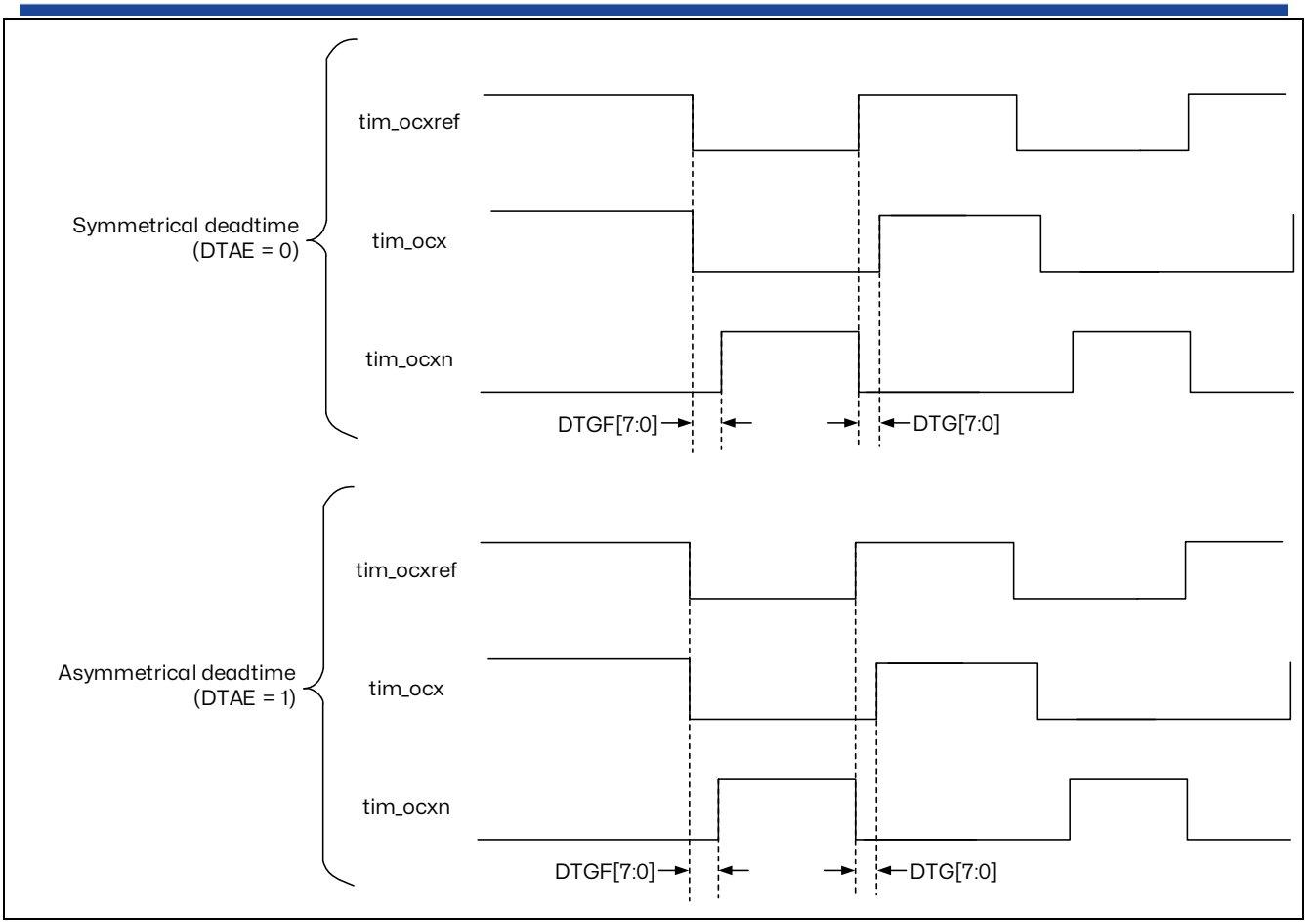


图 13.44 非对称死区时间

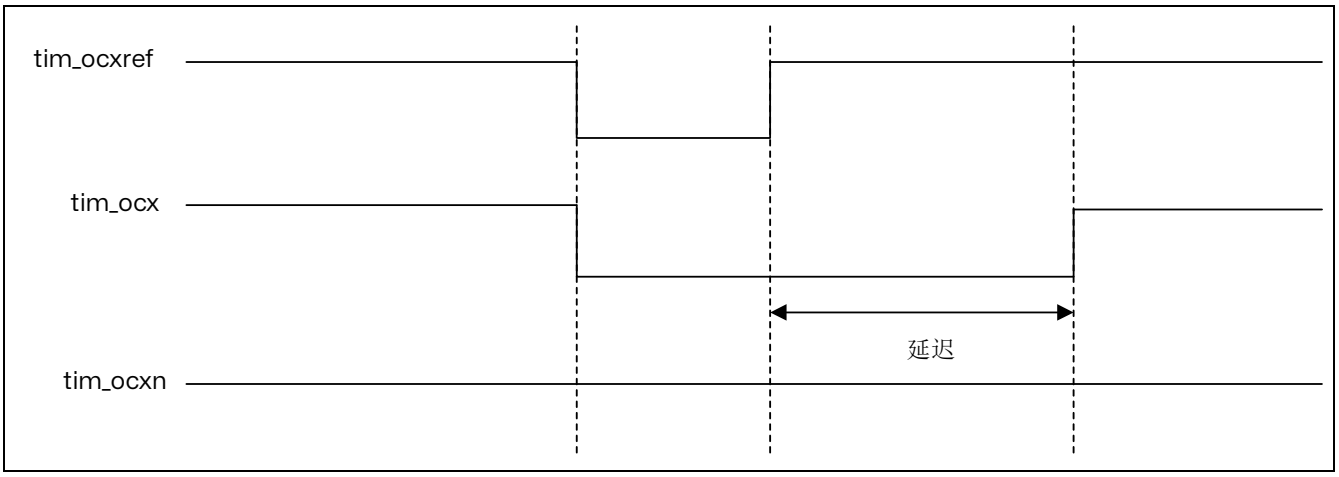


图 13.45 死区波形延迟大于负脉冲

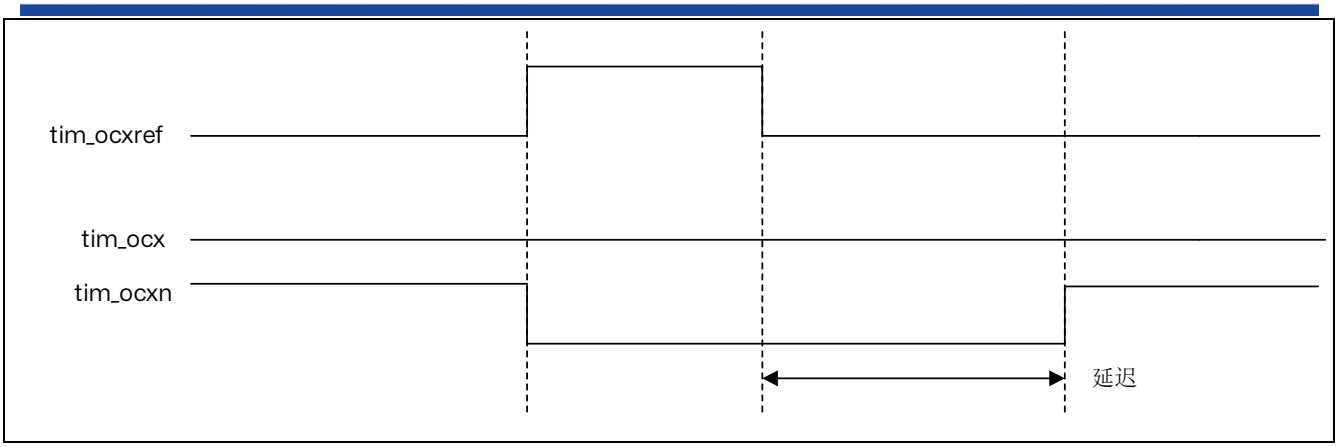


图 13.46 死区波形延迟大于正脉冲

死区延迟对于所有通道均相同，可通过 TIM8_BDTR 寄存器中的 DTG 位进行编程。有关延迟时间计算的信息，请参见后续小节：TIM8 刹车和死区寄存器(TIM8_BDTR)。

将 OCxREF 重定向到 OCx 或 OCxN

在输出模式（强制输出模式、输出比较模式或 PWM 模式）下，通过配置 TIM8_CCER 寄存器中的 CCxE 和 CCxNE 位，可将 tim_ocxref 重定向到 tim_ocx 输出或 tim_ocxn 输出。通过此功能，可以在一个输出上发送特定波形（如 PWM 或静态有效电平），而同时使互补输出保持其无效电平。或者，使两个输出同时保持无效电平，或者两个输出同时处于有效电平，两者互补并且带死区。

注：如果仅使能 tim_ocxn (CCxE=0，CCxNE=1)，*两者不互补，一旦* tim_ocxref *为高电平，* tim_ocxn *即变为有效。例如，如果* CCxNP=0，*则* tim_ocxn= tim_ocxref。*另一方面，如果同时使能* tim_ocxn *和* tim_ocxref（CCxE=CCxNE=1），tim_ocx *在* tim_ocxref *为高电平时变为有效，而* tim_ocxn *则与之互补，在* tim_ocxref *为低电平时变为有效。*

13.3.16 用刹车功能

刹车功能的目的是保护由定时器产生的 PWM 信号驱动的功率开关。两个刹车输入通常连接到功率级和三相逆变器的故障输出。当激活时，刹车电路会关闭 PWM 输出并强制它们进入预定义的安全状态。还可以选择许多内部 MCU 事件来触发输出关闭。

刹车具有两个通道。一个刹车通道可以收集系统级故障（时钟故障、ECC/奇偶校验错误，.....）和应用故障（来自输入引脚和内置比较器），并可以在死区时间后将输出强制到一个预定义的水平（无论是有效还是无效）。一个刹车 2 通道只包括应用故障，并能够将输出强制到非活动状态。

刹车期间输出使能信号和输出电平取决于几个控制位：

- TIM8_BDTR 寄存器中的 MOE 位允许通过软件使能/禁用输出，并在发生刹车或刹车 2 事件时重置。
- TIM8_BDTR 寄存器中的 OSSI 位定义了定时器是否在非活动状态下控制输出或将其控制权释放给 GPIO 控制器（通常是为了使其处于高阻模式）。
- TIM8_CR2 寄存器中的 OISx 和 OISxN 位设置输出关闭电平，可以是有效电平或无效电平。无论 OISx 和 OISxN 的值如何，tim_ocx 和 tim_ocxn 输出不能同时在给定时间内设置为有效电平。更多详细信息请参考后续表格：具有刹车功能的互补 tim_ocx 和 tim_ocxn 通道的输出控制位。

当退出重置状态时，刹车电路被禁用，并且 MOE 位为低电平。通过设置 TIM8_BDTR 寄存器中的 BKE 和 BK2E 位，可以使能刹车功能。通过配置相同的寄存器中的 BKP 和 BK2P 位，可以选择刹车输入

极性。BKEx 和 BKPx 可以同时修改。在写入 BKEx 和 BKPx 位后，需要等待 1 个 APB 时钟周期才能使写入生效。因此，在写入操作后，需要等待 1 个 APB 时钟周期才能正确地读取该位。

由于 MOE 下降沿可能是异步的，因此在实际信号（作用于输出）和同步控制位（在 TIM8_BDTR 寄存器中访问）之间插入了一个重新同步电路。这导致异步信号和同步信号之间存在一些延迟。特别是当 MOE 被设置为 1 而之前它是低电平时，在正确读取之前必须插入一个延迟（伪指令）。这是因为写入作用于异步信号，而读取反映的是同步信号。

刹车（tim_brk）通道的源为：

- 连接到 TIM8_BKIN 引脚的外部源（根据 GPIO 备用功能选择寄存器中的选择而定），具有极性选择和可选的数字滤波
- 内部源：
  - 来自 tim_brk_cmpx 输入（请参考章节：TIM8 引脚和内部信号）
  - 来自 tim_sys_brk 输入的系统中断请求（请参考章节：TIM8 引脚和内部信号）

刹车 2（tim_brk2）的源为：

- 连接到 TIM8_BKIN2 引脚的外部源（根据 GPIO 备用功能选择寄存器中的选择而定），具有极性选择和可选的数字滤波。
- 来自 tim_brk2_cmpx 输入的内部源（请参考章节：TIM8 引脚和内部信号）

也可以通过软件使用 TIM8_EGR 寄存器中的 BG 和 B2G 位生成刹车事件。

所有源在进入定时器的 tim_brk 或 tim_brk2 输入之前进行 OR 操作，如下图所示。

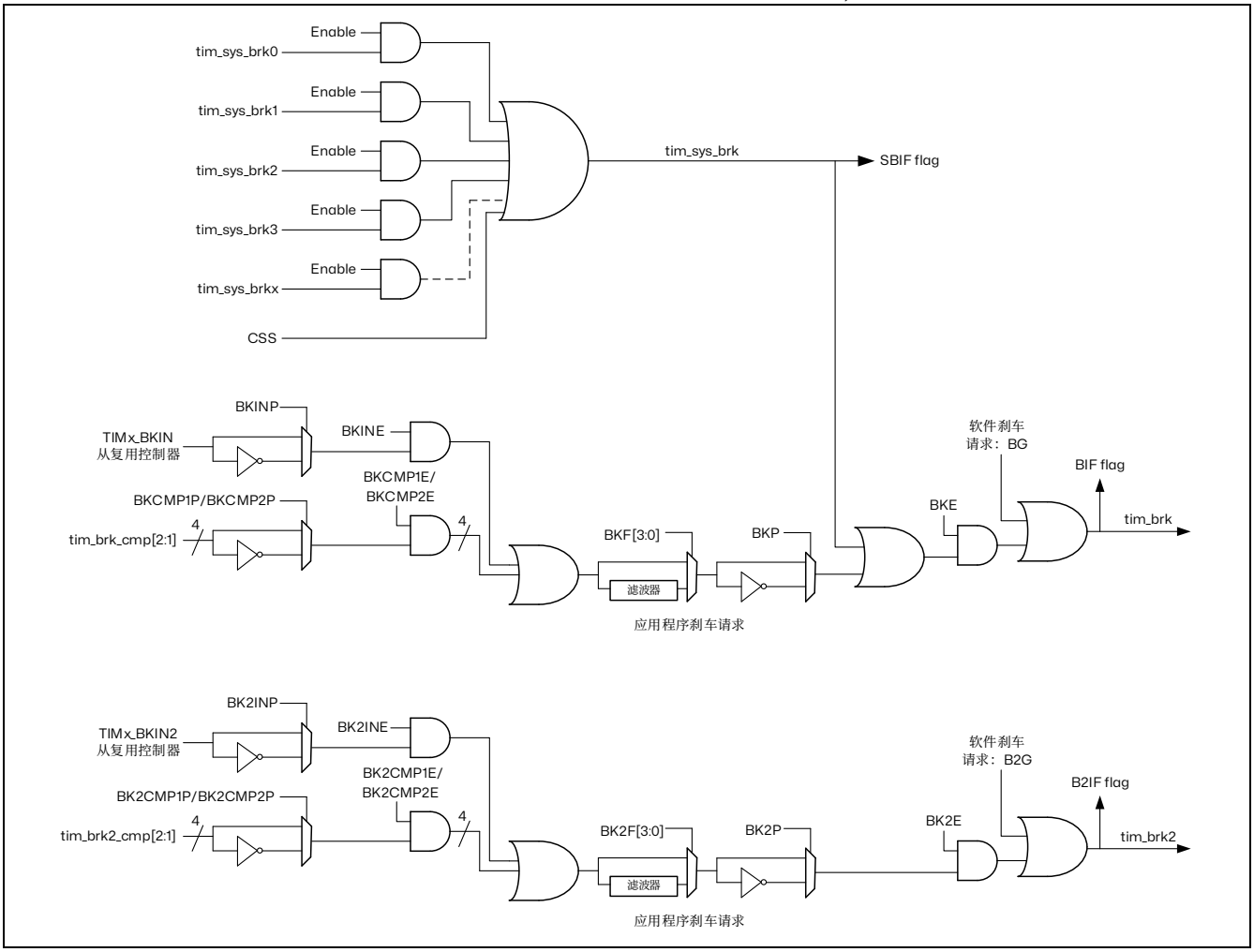


图 13.47 刹车和刹车 2 电路概述

*注意：只有当可编程滤波器禁用时，才保证异步（无时钟）操作。如果使能，必须使用故障安全时钟模式（例如使用内部*PLL *和/或*CSS）*来保证刹车事件得到处理。*

当其中一个刹车发生时（在所选的刹车输入线上检测到选定电平）：

- MOE 位会被异步清除，使输出处于非活动状态、空闲状态或甚至将控制权释放给 GPIO 控制器（由 OSSI 位选择）。即使 MCU 振荡器关闭，此功能也会使能。
- 一旦 MOE=0，每个输出通道将驱动 TIM8_CR2 寄存器中 OISx 位编程的电平。如果 OSSI=0，定时器将释放输出控制（由 GPIO 控制器接管），否则使能输出保持高电平。
- 当使用互补输出时：
  — 输出首先处于无效状态（取决于极性）。这是异步完成的，因此即使没有向定时器提供时钟也可以正常工作。
  — 如果定时器的时钟仍然存在，则重新激活死区生成器，以便在死区后驱动 OISx 和 OISxN 位中编程的电平的输出。即使在这种情况下，tim_ocx 和 tim_ocxn 也不能同时驱动到其活动电平。请注意，由于在 MOE 上的重新同步，死区持续时间比平常稍长（大约 2 个 tim_ker_ck 时钟周期）。
  — 如果 OSSI=0，定时器将释放输出控制（由 GPIO 控制器接管，该控制器强制高阻态），否则使能输出保持或变为高电平，一旦 CCxE 或 CCxNE 位为高电平。
- 设置刹车标志位（TIM8_SR 寄存器中的 SBIF、BIF 和 B2IF 位）。如果 TIM8_DIER 寄存器中的 BIE 位已设置，则生成刹车。
- 如果 TIM8_BDTR 寄存器中的 AOE 位已设置，则在下一个更新事件（UEV）时自动重新设置 MOE 位。例如，这可以用于执行调节。否则，MOE 保持低电平，直到应用程序再次将其设置为‘1’。在这种情况下，它可以用于安全，并将中断输入连接到来自功率驱动器、温度传感器或任何安全组件的警报。

*注意：如果*MOE *被*CPU *复位，而*AOE *位已设置，则输出处于空闲状态，并强制处于无效电平或高阻态，具体取决于*OSSI *值。如果*MOE *和*AOE *位都被*CPU *复位，则输出处于禁用状态，并驱动在*TIM8_CR2 *寄存器中*OISx *位编程的电平。*

*注意：刹车输入在电平上有效。因此，当刹车输入处于有效状态时（无论是自动还是通过软件设置），*MOE *都无法设置。同时，状态标志*BIF *和*B2IF *不能清除。*

除刹车输入和输出管理外，刹车电路内部还实施了写保护，用以保护应用的安全。通过该功能，用户可冻结多个参数配置（死区持续时间、tim_ocx/tim_ocxn 极性和禁止时的状态、OCxM 配置、刹车使能和极性）。可以通过 TIM8_BDTR 寄存器中的 LOCK 位，从 3 种保护级别中进行选择。请参考章节：TIM8 刹车和死区寄存器（TIM8_BDTR）。MCU 复位后只能对 LOCK 位执行一次写操作。

下图所示为输出对刹车响应行为的示例。

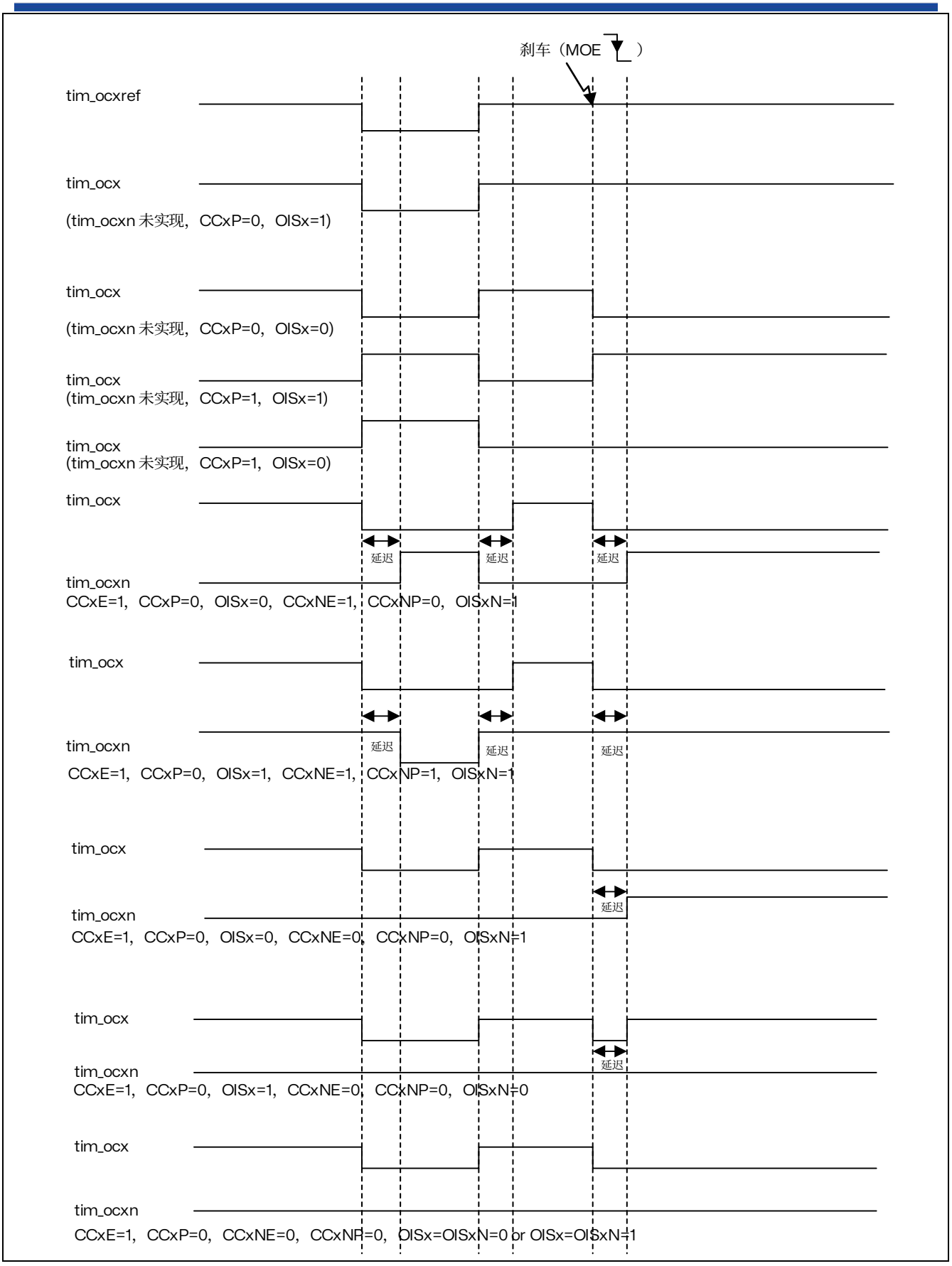


图 13.48 tim_brk 在发生刹车事件时的各种输出行为（OSSI=1）

两个中断输入在计时器输出上具有相同的行为：

— tim_brk/ tim_brk2 输入可禁止（无效状态）PWM 输出，也可将 PWM 输出强制为预定义的安全状态。

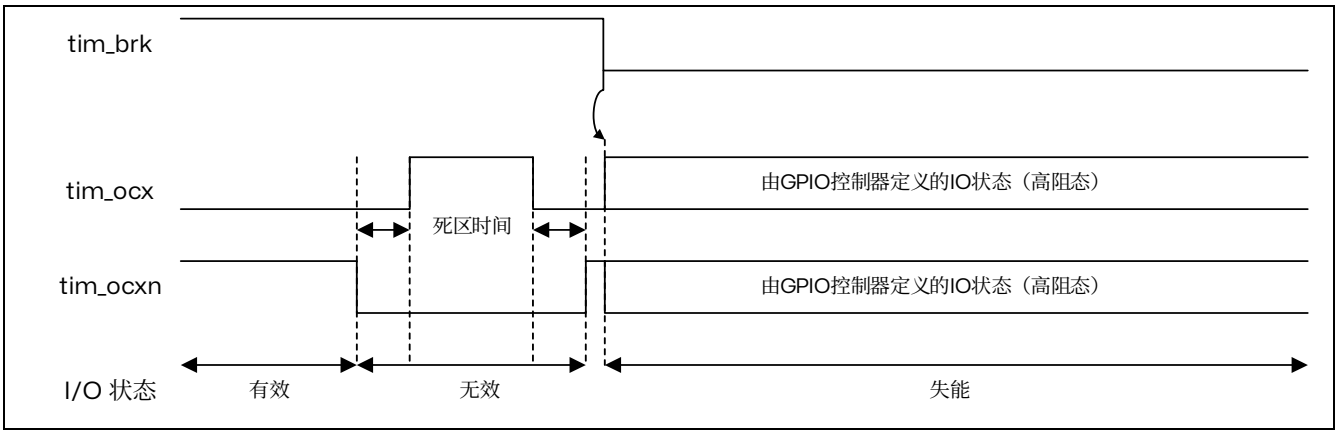


图 13.49 当 tim_brk/tim_brk2 有效（OSSI=0）时 PWM 输出状态

13.3.17 双向刹车输入

TIM8 具有双向刹车 I/O，如下图所示。

这为以下方面提供支持：

- 板级全局刹车信号，用于向外部 MCU 或门驱动器发送故障信号，使用唯一的引脚作为输入和输出状态引脚
- 内部刹车源和多个外部开漏源 OR 连接在一起，以触发唯一的刹车事件，当多个内部和外部刹车源必须合并时

使用 TIM8BDTR 寄存器中的 BKBID 和 BK2BID 位配置 tim_brk 和 tim_brk2 输入为双向模式。BKBID 编程位可以使用 TIM8BDTR 寄存器中的 LOCK 位锁定为只读模式（在 LOCK 级别 1 或更高）。

tim_brk 和 tim_brk2 输入都支持双向模式，并且要求 I/O 配置为开漏模式，使用低电平有效极性（使用 BKINP、BKP、BK2INP 和 BK2P 位）。来自系统（例如 CSS）、片上外设或刹车输入的任何刹车请求都会强制将刹车输入置于低电平，以指示故障事件。出于安全原因，如果极性位没有正确设置（高电平有效），则双向模式将被抑制。

刹车软件事件（BG 和 B2G）也会强制将刹车 I/O 置为“0”，以指示外部组件定时器已进入刹车状态。然而，只有使能了刹车（BKE 或 B2KE = 1）时，这才有效。当使用 BKE 或 B2KE = 0 生成软件刹车事件时，输出将处于安全状态，并且会设置刹车标志，但 TIM8_BKIN 和 TIM8_BKIN2 I/O 没有影响。

安全解发机制可防止系统被彻底锁定（刹车输入低电平触发刹车，强制该输入为低电平）。

当 BKDSRM（BK2DSRM）位设置为 1 时，这将释放刹车输出以清除故障信号并允许重新给系统装载。

表 13.11 刹车保护解除条件

MOE	BKBID (BK2BID)	BKDSRM (BK2DSRM)	刹车保护状态
0	0	X	装载
0	1	0	装载
0	1	1	解除
1	X	X	装载

装载和重装载刹车电路

默认情况下（外设复位配置），刹车电路（输入或双向模式）处于装载状态。
在发生刹车（刹车 2）事件后，必须按照以下步骤重新装载保护：

- BKDSRM（BK2DSRM）位必须设置为释放输出控制
- 软件必须等待系统刹车条件消失（如果有的话），并清除 SBIF 状态标志（或在重新装载之前系统地清除它）
- 软件必须轮询 BKDSRM（BK2DSRM）位，直到硬件清除该位（当应用程序刹车条件消失时）

从这一点开始，刹车电路处于装载和有效状态，并且可以设置 MOE 位以重新使能 PWM 输出。

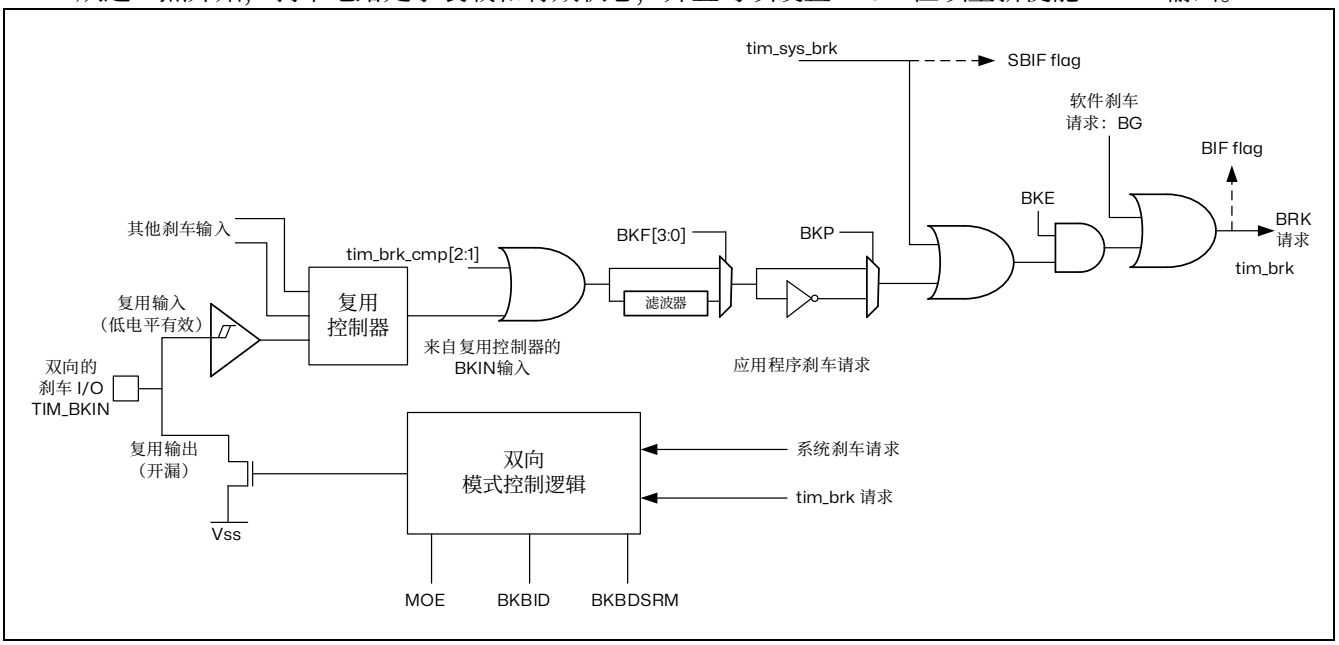


图 13.50 输出重定向（未表示 tim_brk2 请求）

13.3.18 在外部事件时清除 OCxREF 信号

对于给定通道，在 tim_ocref_clr_int 输入施加高电平（相应 TIM8_CCMRx 寄存器中的 OCxCE 使能位置“1”），可使 tim_ocxref 信号变为低电平。OCxREF 信号将保持低电平，直到发生下一更新事件(UEV)。（特别注意，当 TIMx_RCR>0 时，只有重复计数器达到零才会生成更新事件，波形才会恢复）。

此功能仅能用于输出比较模式和 PWM 模式，而不适用于强制输出模式。

可以通过配置 TIM8_SMCR 寄存器中的 OCCS 位，将 tim_ocref_clr_int 输入选择为 tim_ocref_clr 输入和 tim_etrf（滤波后的 tim_etr_in）之间。

tim_ocref_clr 输入可以在多个输入中进行选择，使用 TIM8_AF2 寄存器中的 OCRSEL[2:0]位域，如下图所示。请参考章节：TIM8 引脚和内部信号，以获取产品中可用源的列表。

当配置组合 PWM 模式和三相组合 PWM 模式，对于给定通道，在 tim_ocref_clr_int 输入施加高电平（相应 TIMx_CCMRx 寄存器中的 OCxCE 使能位置“1”），可使组合结果 tim_ocxrefc 信号变为低电平。tim_ocxrefc 信号将保持低电平，直到发生下一更新事件（UEV）（特别注意，当 TIMx_RCR>0 时，只有重复计数器达到零才会生成更新事件，波形才会恢复）。

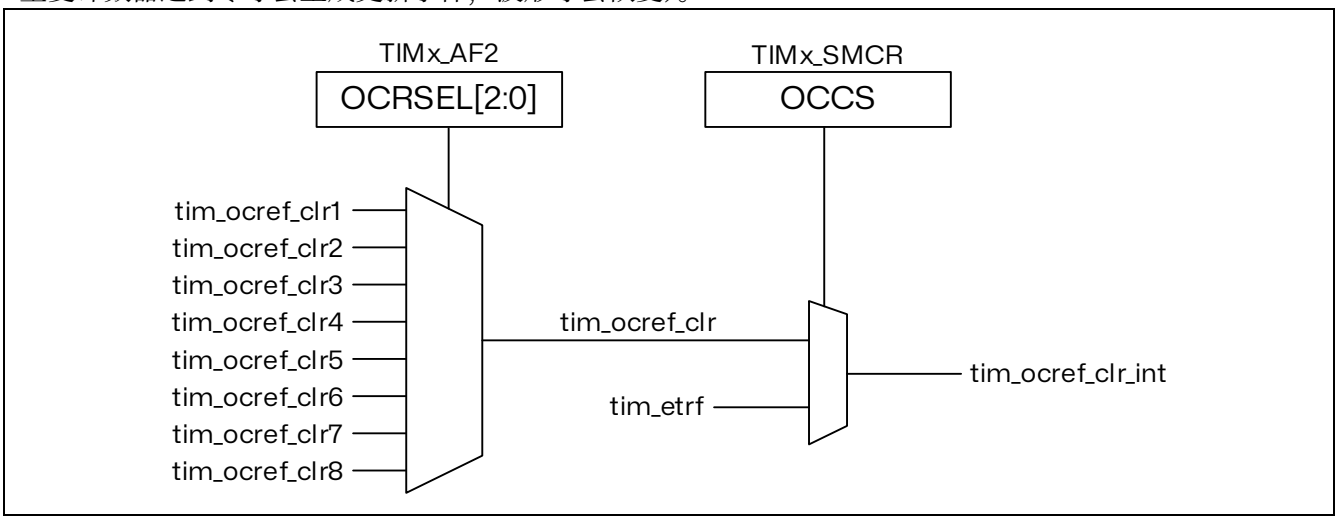


图 13.51 tim_ocref_clr 输入选择复用器

当 tim_etrf 被选择。tim_etr_in 必须如下配置：

1. 必须关闭外部触发预分频器：TIM8_SMCR 寄存器中的 ETPS[1:0]位置“00”。
2. 必须禁止外部时钟模式 2：TIM8_SMCR 寄存器中的 ECE 位置“0”。
3. 外部触发极性(ETP)和外部触发滤波器(ETF)可根据用户需要进行配置。

下图对比了使能位 OCxCE 在不同值下的情况，显示了当 tim_etrf 输入变为高电平时 tim_etrf 信号的行为。在本例中，定时器 TIM8 编程为 PWM 模式。

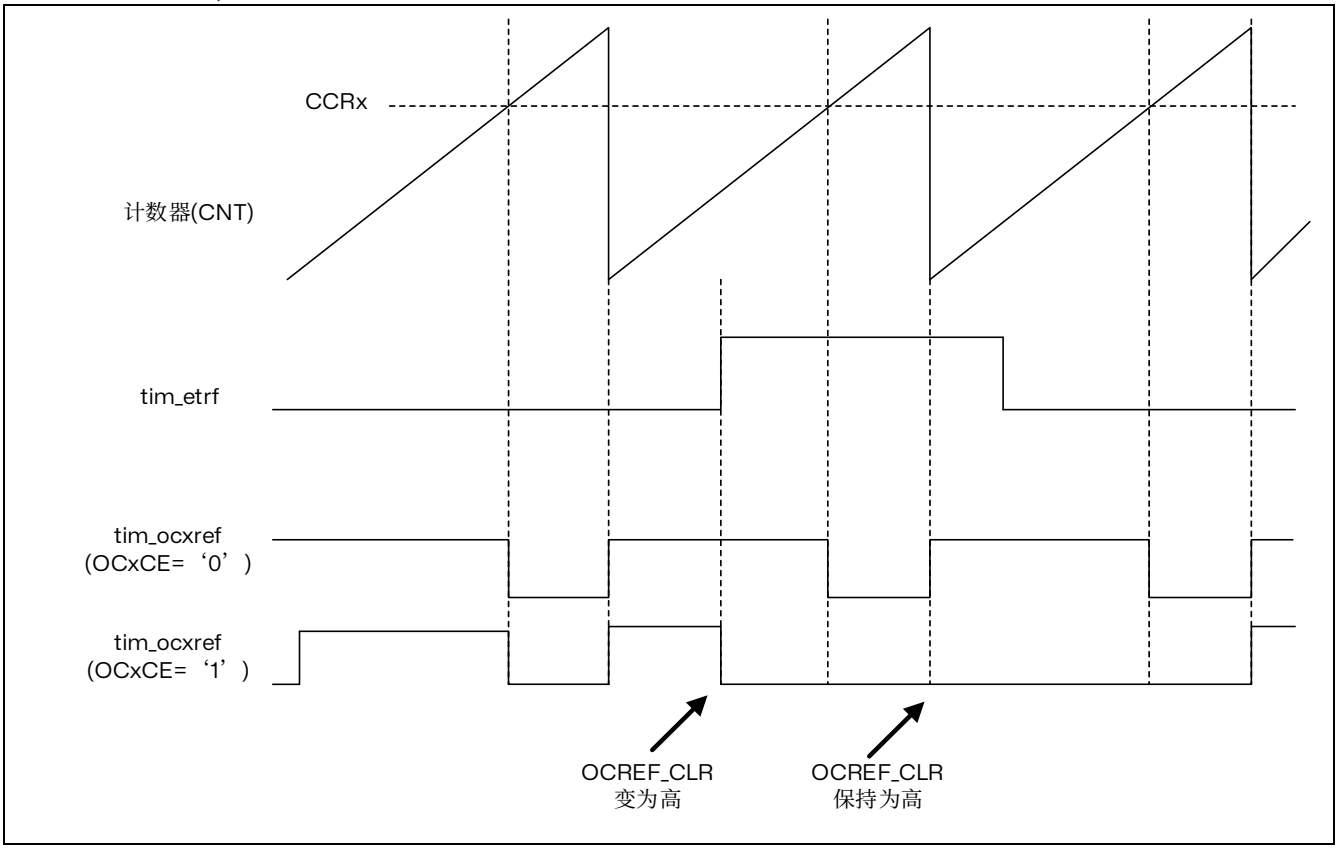


图 13.52 清除 TIM8 的 tim_ocxref

13.3.19 生成 6 步 PWM

当通道使用互补输出时，OCxM、CCxE 和 CCxNE 位上提供预装载位。发生 COM 换向事件时，这些预装载位将传输到影子位。因此，用户可以预先编程下一步骤的配置，并同时更改所有通道的配置。COM 可由软件通过将 TIM8_EGR 寄存器中的 COM 位置 1 而生成，也可以由硬件在 tim_trgi 上升沿生成。

发生 COM 事件时，TIM8_SR 寄存器中的 COMIF 位将会置 1。这时，如果 TIM8_DIER 寄存器中的 COMIE 位置 1，将产生中断。

下图以 3 种不同的编程配置为例，显示了发生 COM 事件时 tim_ocx 和 tim_ocxn 输出的行为。

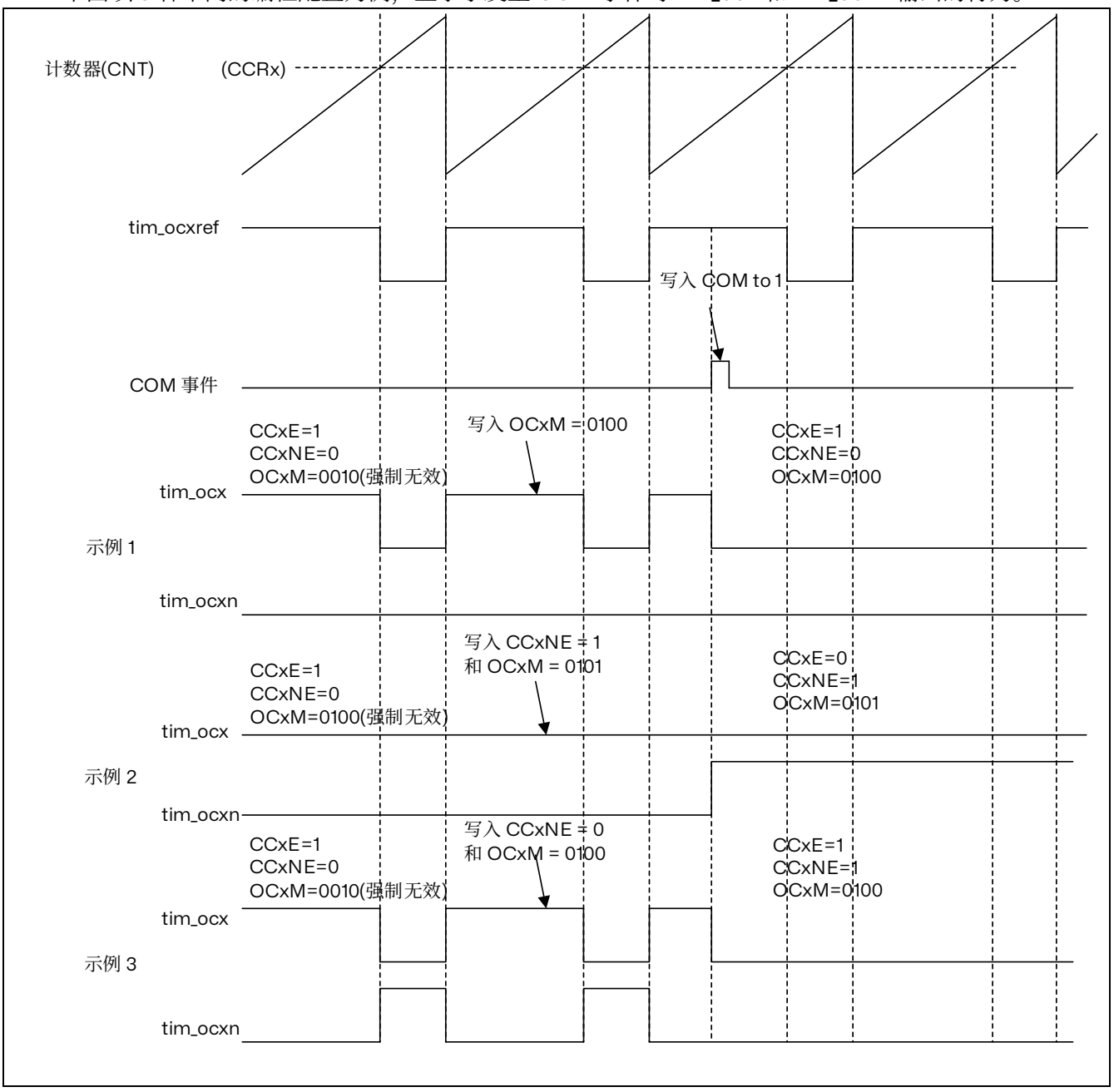


图 13.53 COM 事件生成 6 步 PWM 的示例（OSSR=1）

13.3.20 单脉冲模式

单脉冲模式（OPM）是上述模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。将 TIM8_CR1 寄存器中的 OPM 位置 1，即可选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

只有当比较值与计数器初始值不同时，才能正确产生一个脉冲。启动前（定时器等待触发时），必须进行如下配置：

- 递增计数模式下：CNT < CCRx ≤ ARR（特别注意，0<CCRx）
- 递减计数模式下：CNT > CCRx

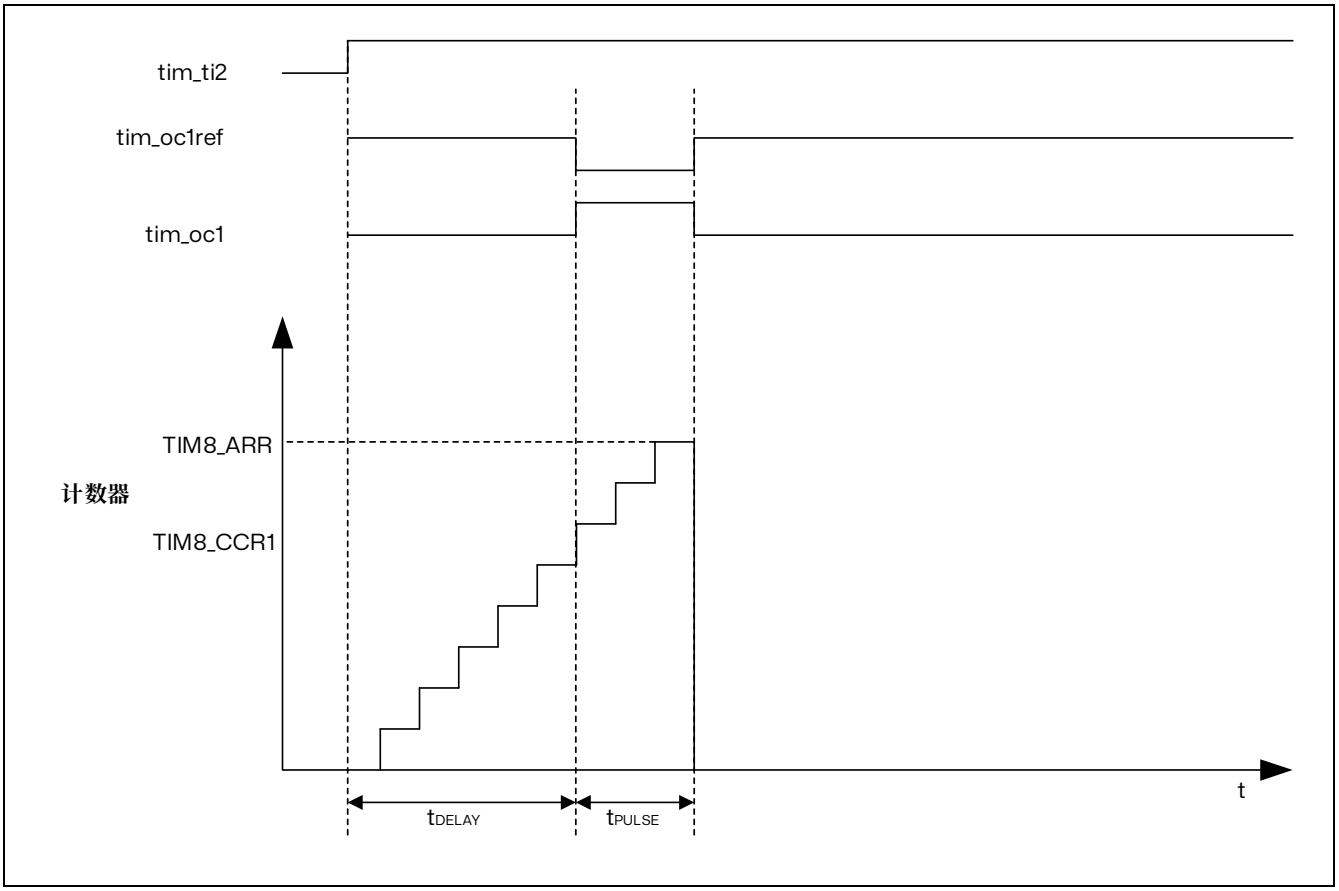


图 13.54 单脉冲模式的例子

例如，当 tim_trgi 信号到来时，经过 t_{DELAY} 的延迟，在 tim_oc1 上产生一个长度为 t_{PULSE} 的正脉冲。使用 tim_itr0 作为触发：

1. 在 TIMx_SMCR 寄存器中写入 TS=0000，将 tim_itr0 配置为从模式控制器的触发（tim_trgi）。
2. 在 TIMx_SMCR 寄存器中写入 SMS=“110”（触发模式），使用 tim_itr0 启动计数器。

OPM 波形通过对比较寄存器执行写操作来定义（考虑时钟频率和计数器预分频器）。

- t_{DELAY} 由写入 TIM8_CCR1 寄存器的值定义。
- t_{PULSE} 由自动重载值与比较值（TIM8_ARR - TIM8_CCR1）之差来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从“0”变为“1”，在计数器达到自动重载值时由“1”变为“0”。为此，应在 TIM8_CCMR1 寄存器中写入 OC1M=111，以使能 PWM 模式 2。如果需要，可选择在 TIM8_CCMR1 寄存器的 OC1PE 和 TIM8_CR1 寄存器的 ARPE 中写入

“1”，以使能预装载寄存器。这种情况下，必须在 TIM8_CCR1 寄存器中写入比较值并在 TIM8_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 tim_ti2 上的外部触发事件。本例中，CC1P 的值为“0”。

在本例中，TIM8_CR1 寄存器中的 DIR 和 CMS 位应为低。

由于仅需要 1 个脉冲（单脉冲模式），因此应向 TIM8_CR1 寄存器的 OPM 位写入“1”，以便在发生下一更新事件（计数器从自动重载值返回到 0）时使计数器停止计数。TIM8_CR1 寄存器中的 OPM 位置“0”时，即选择重复模式。

特例：tim_ocx 快速使能：

在单脉冲模式下，tim_tix 输入的边沿检测会将 CEN 位置 1，表示使能计数器。然后，在计数器值与比较值之间发生比较时，将切换输出。但是，完成这些操作需要多个时钟周期，这会限制可能的最小延迟（t_{DELAY} 最小值）。

如果要输出延迟时间最短的波形，可以将 TIM8_CCMRx 寄存器中的 OCxFE 位置 1。这样会强制 tim_ocxref（和 tim_ocx）对激励信号做出响应，而不再考虑比较的结果。其新电平与发生比较匹配时相同。仅当通道配置为 PWM1 或 PWM2 模式时，OCxFE 才会起作用

13.3.21 可重触发单脉冲模式

此模式允许计数器在接收到刺激后开始计数，并生成一个具有可编程长度的脉冲，但与上一节中描述的非重触发单脉冲模式存在以下差异：

— 只要触发器发生，脉冲就会立即开始（没有可编程的延迟）

— 如果在先前触发的脉冲完成之前发生新的触发，则脉冲会被延长

定时器必须处于从模式，TIM8_SMCR 寄存器的 SMS[3:0]位设置为‘1000’（组合重置+触发模式），OCxM[3:0]位设置为‘1000’或‘1001’，用于可重触发 OPM 模式 1 或 2。

如果定时器配置为递增计数模式，则相应的 CCRx 必须设置为 0（ARR 寄存器设置脉冲长度）。如果定时器配置为递减计数模式，CCRx 必须大于或等于 ARR。

*注意：*OCxM[[3:0]*和*SMS[3:0]*位字段由于兼容性的原因被分成两个部分，最高有效位与最低*3 *位不是连续的。*OCxM *这种模式不能与中央对齐*PWM *模式一起使用。*TIM8_CR1 *寄存器中的*CMS[1:0]*必须设置为*00。

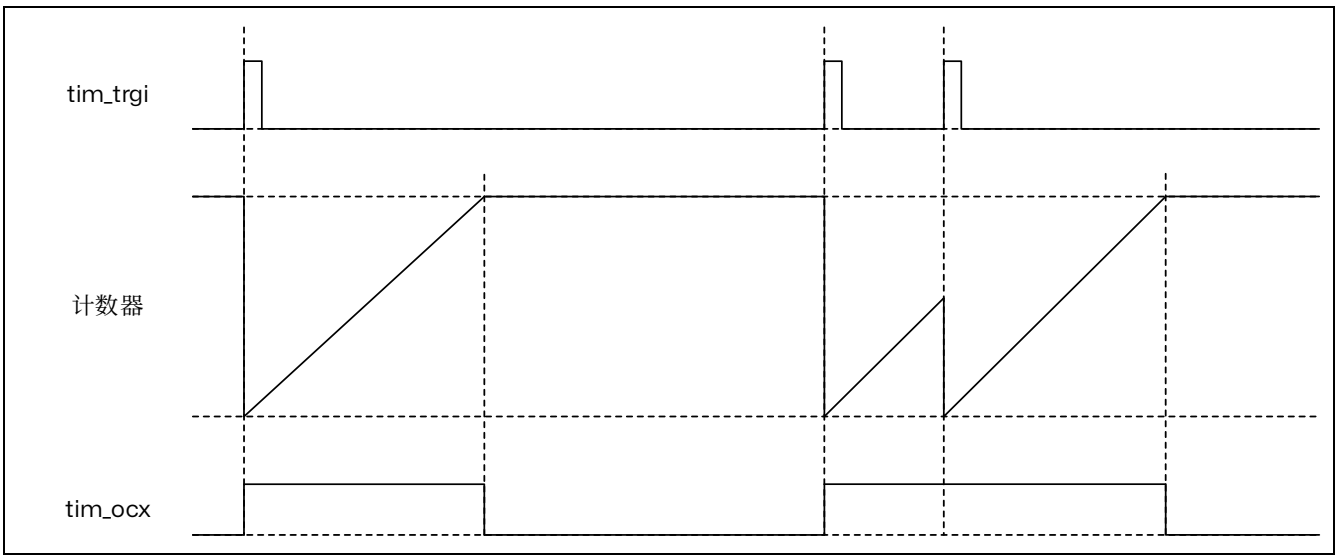


图 13.55 可重触发单脉冲模式

13.3.22 在比较模式下的脉冲

在比较匹配事件发生时，可以生成一个脉冲。当计数器的值等于给定的比较值时，可以生成一个具有可编程脉冲宽度的信号，用于调试或同步。

此模式适用于任何从机模式选择，包括边沿对齐和中央对齐计数模式。它仅适用于通道 3 和通道 4。脉冲发生器是唯一的，由两个通道共享，如下图所示。

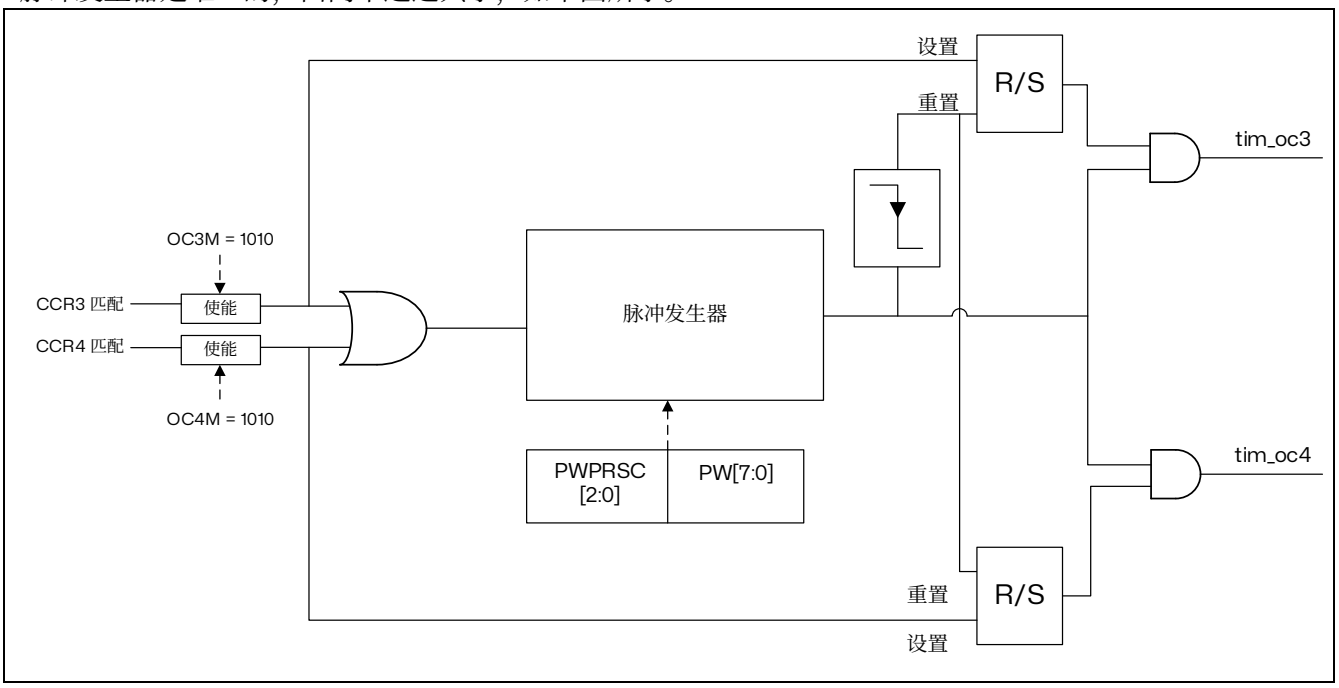


图 13.56 脉冲发生器电路

下图展示了如何在边沿对齐模式下生成脉冲。

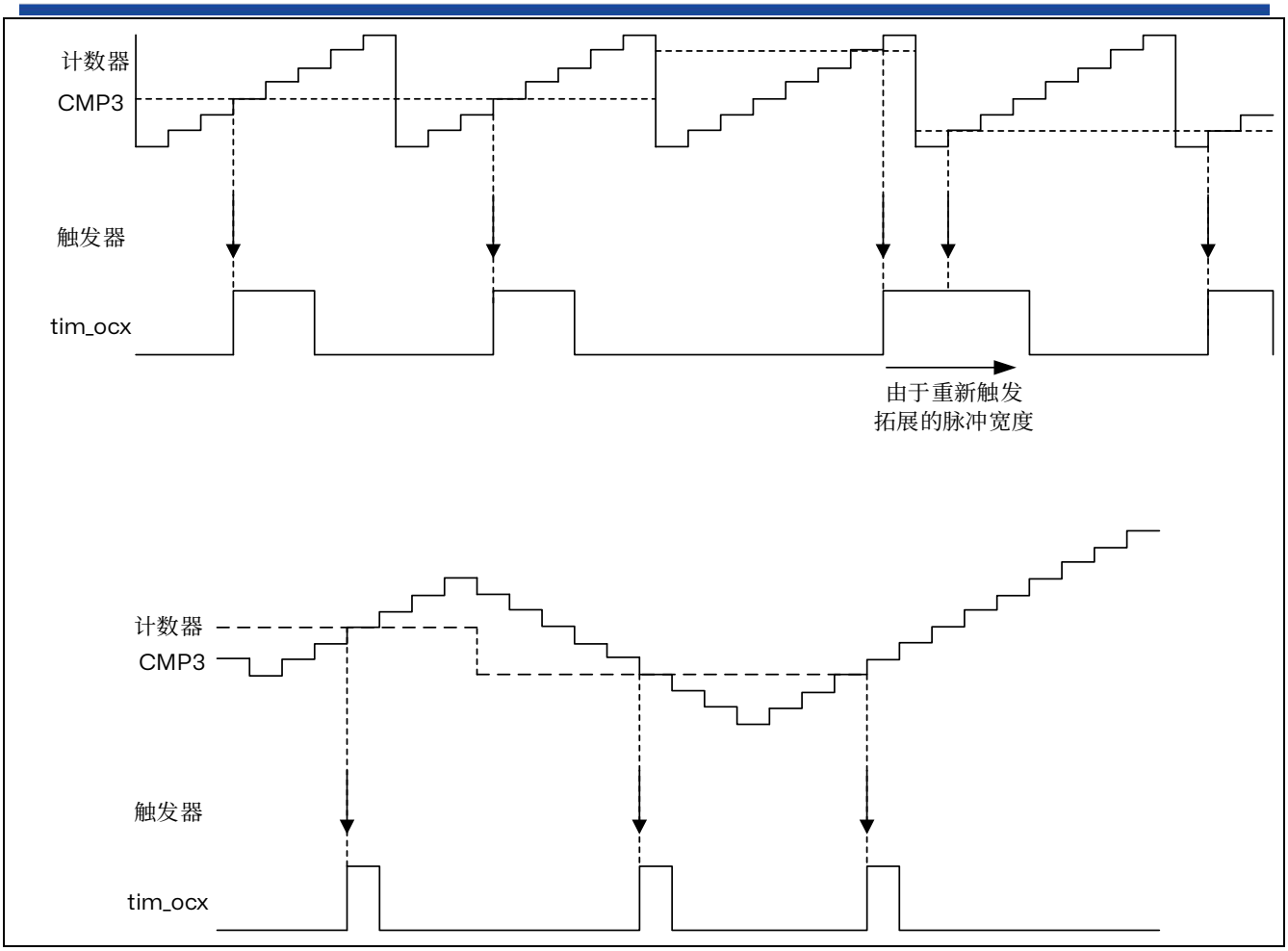


图 13.57 在比较事件上的脉冲生成，适用于边沿对齐模式

此输出比较模式是通过 TIM8_CCMR2 寄存器中的 OC3M[3:0]和 OC4M[3:0]位字段进行选择的。

脉冲宽度通过寄存器中的 PW[7:0]位域进行编程，使用特定的时钟，并根据 PWPRSC[2:0]位的预设进行预分频，具体如下：

$t_{PW} = PW[7:0] \times t_{PWG}$

其中 $t_{PWG} = (2^{(PWPRSC[2:0])}) \times t_{tim_ker_ck}$。

给出了分辨率和最大值，具体取决于预分频器的值。

脉冲可以重新触发：在脉冲正在进行时触发新的脉冲，会导致脉冲被延长。

注意：如果同时使能两个通道，只要一个通道上的触发器不与并发输出上生成的脉冲重叠，脉冲就会独立发出。相反，如果两个触发器重叠，与第一个到达的触发器相关的脉冲宽度会被延长（由于重新触发），而最后一个到达的触发器的脉冲宽度是正确的（如下图所示）。

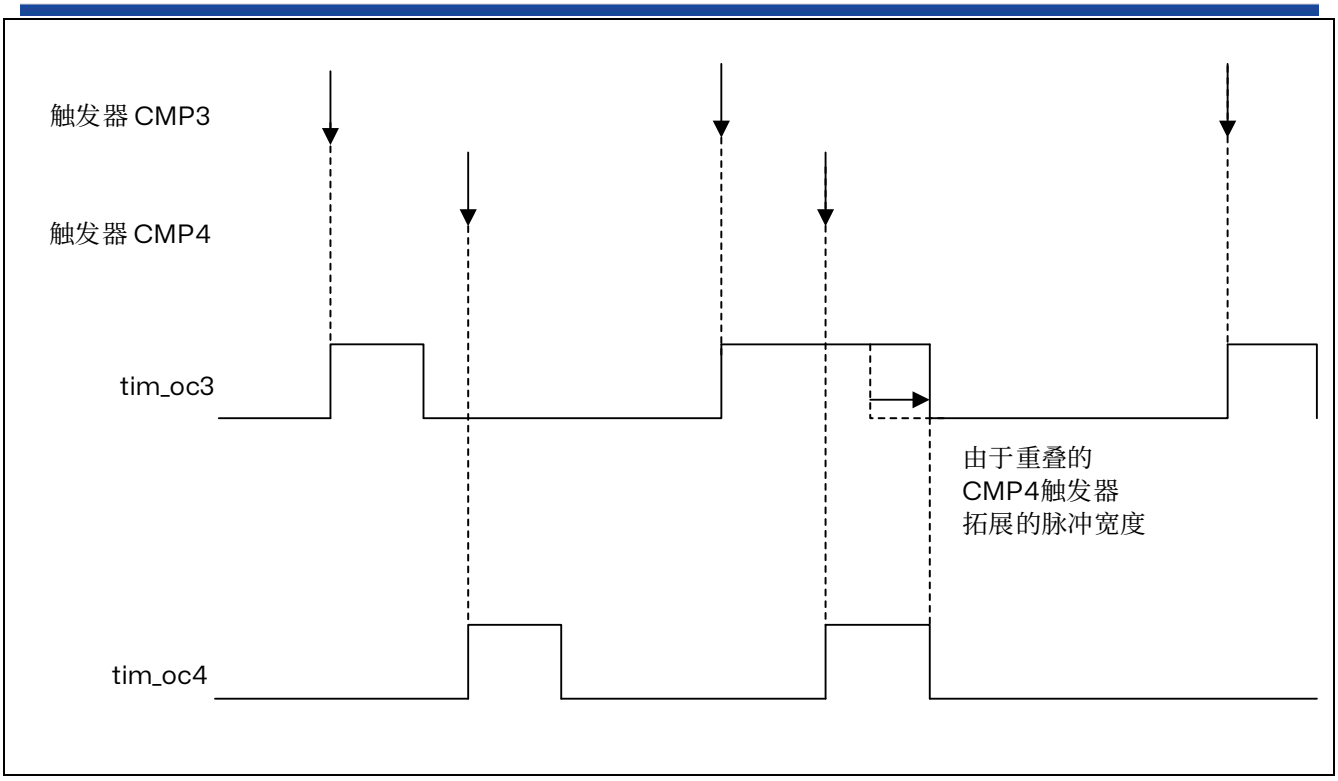


图 13.58 在同时触发的情况下延长脉冲宽度

13.3.23 方向位输出

可以将方向信号从定时器输出到 tim_oc3n 和 tim_oc4 输出信号（TIM8_CR1 寄存器中的 DIR 位的副本）。这是通过将 TIM8_CCMR2 寄存器中的 OC3M[3:0]或 OC4M[3:0]位字段设置为 1011 来实现的。

此功能在中央对齐的 PWM 模式下具有指示上/下相位的信号。

13.3.24 UIF 位重映射

TIM8_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志 UIF 连续复制到定时器计数器寄存器的第 31 位（TIM8CNT[31]）。这允许以原子方式读取计数器的值和一个潜在的由 UIFCPY 标志指示的回滚条件。在特定情况下，它可以避免由背景任务（读取计数器）和中断（更新中断）之间的处理引起的竞争条件，从而简化计算。

UIF 和 UIFCPY 标志之间没有延迟。

13.3.25 定时器同步

TIM1/8 定时器在内部连接在一起，用于定时器同步或链接。它们可以在几种模式下同步：复位模式，门控模式，触发模式，复位+触发和门控+复位模式。

从模式：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 TIM1/8_CR1 寄存器的 URS 位为低，还产生一个更新事件 UEV；然后所有的预装载寄存器（TIM1/8_ARR，TIM1/8_CCRx）都被更新了。

在以下的例子中，tim_itr1 内部触发输入端接收到 tim3_ trgo 触发信号导致向上计数器被清零：

- 置 TIM3_CR2 寄存器中 MMS=010，选择更新事件作为触发输出（tim3_ trgo）

- 置 TIM3_CR1 寄存器中 CEN=1，启动计数器。
- 置 TIM1/8_SMCR 寄存器中 SMS=100，配置定时器为复位模式；置 TIM1/8_SMCR 寄存器中 TS=0001，选择 tim_itr1 作为输入源。
- 置 TIM1/8_CR1 寄存器中 CEN=1，启动计数器。

计数器开始依据内部时钟计数，然后正常运转直到 tim_itr1 出现一个 tim3_ trgo 触发信号；此时，计数器被清零然后从 0 重新开始计数。同时，触发标志（TIM1/8_SR 寄存器中的 TIF 位）被设置，中断请求可以被发送（取决于 TIM1/8_DIER 寄存器中的 TIE 位）。

下图显示当自动重装载寄存器 TIM1/8_ARR=0x36 时的动作。

在 tim_itr1 触发信号和计数器的实际复位之间的延时取决于 tim_itr1 输入端的重同步电路。

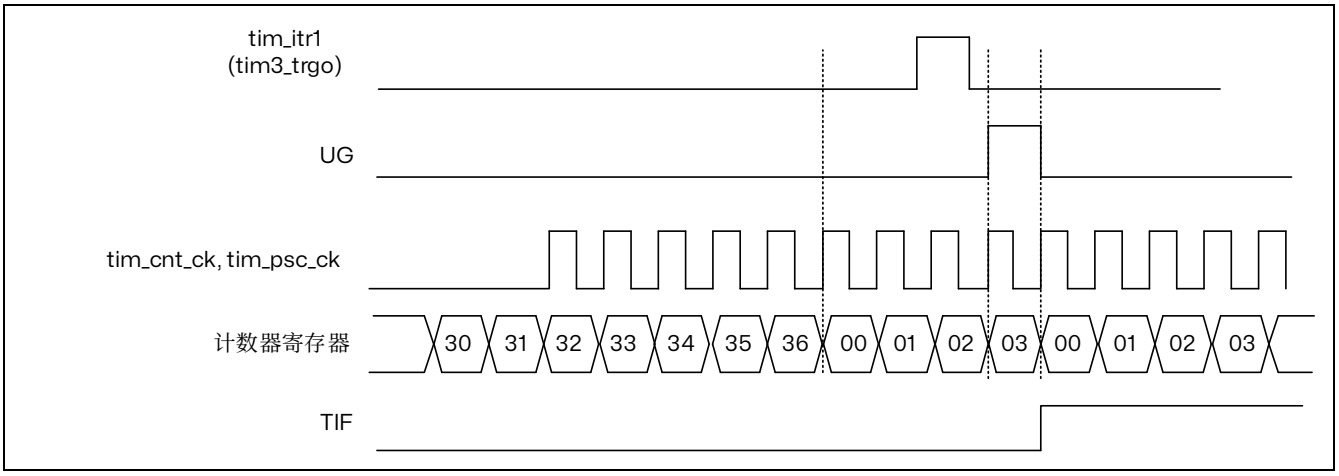


图 13.59 复位模式下的控制电路

从模式：门控模式

按照选中的**输入端有效电平**使能计数器。

在如下的例子中，计数器只在 tim_itr1（选 oc1refc 信号用作 tim3_trgo）电平有效时向上计数：

- 置 TIM3_CR2 寄存器中 MMS=100，选择 oc1refc 信号用作触发输出（tim_trgo）
- 置 TIM3_CCMR1 寄存器中 OC1M=0101，强制 tim_oc1ref 变为有效电平输出（tim_oc1refc）
- 置 TIM3_CCMR1 寄存器中 OC1M=0100，强制 tim_oc1ref 变为无效电平输出（tim_oc1refc）
- 置 TIM1/8_SMCR 寄存器中 SMS=101，配置定时器为门控模式；置 TIM1/8_SMCR 寄存器中 TS=0001，选择 tim_itr1 作为输入源。
- 置 TIM1/8_CR1 寄存器中 CEN=1，启动计数器（在门控模式下，如果 CEN=0，则计数器不能启动，不论触发输入电平如何）。

只要 tim_itr1 **电平有效**，计数器开始依据内部时钟计数，一旦 tim_itr1 **电平无效**则停止计数。当计数器开始或停止时都设置 TIM1/8_SR 中的 TIF 标置。

tim_itr1 触发信号和计数器实际停止之间的延时取决于 tim_itr1 输入端的重同步电路。

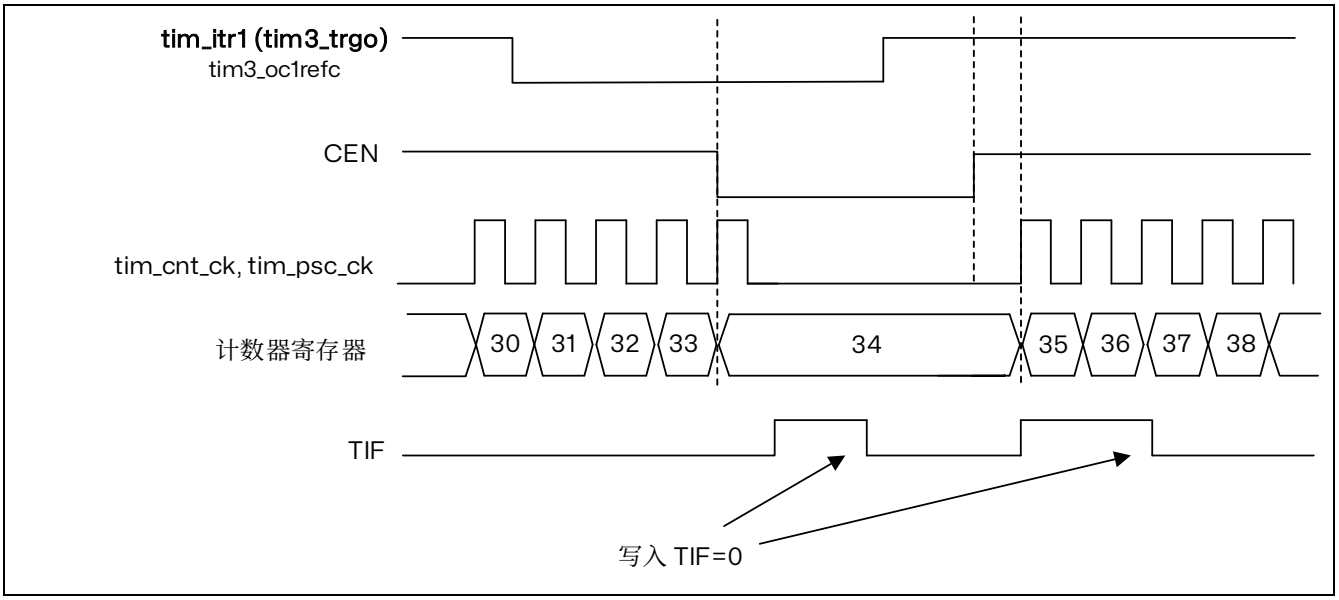


图 13.60 门控模式下的控制电路

从模式：触发模式

输入端上选中的事件使能计数器。

在以下的例子中，计数器在 tim_itr1 内部触发输入端接收到 tim3_ trgo 触发信号开始向上计数：

- 置 TIM3_CR2 寄存器中 MMS=001，选择计数器使能信号 CNT_EN 用作触发输出（tim3_trgo）
- 置 TIM1/8_SMCR 寄存器中 SMS=110，配置定时器为触发模式；置 TIM1/8_SMCR 寄存器中 TS=0001，选择 tim_itr1 作为输入源。
- 置 TIM3_CR1 寄存器中 CEN=1，启动计数器，同步发出 tim3_trgo。

当 tim_itr1 出现 tim3_ trgo 触发信号时，计数器开始在内部时钟驱动下计数，同时设置 TIF 标志。

tim_itr1 触发信号和计数器启动计数之间的延时，取决于 tim_itr1 输入端的重同步电路。

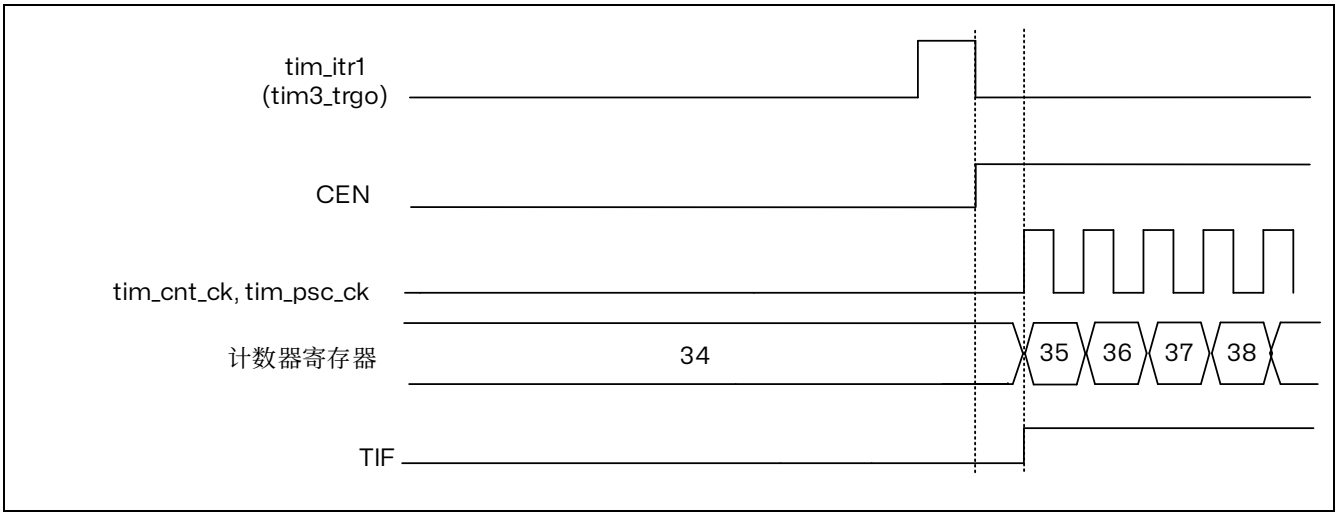


图 13.61 触发器模式下的控制电路

从模式：组合复位+触发模式

在这种情况下，当检测到选择的触发输入（tim_trgi）对应的触发信号就重新初始化计数器，生成寄存器的更新，并启动计数器。

从模式：组合门控+复位模式

触发输入（tim_trgi）电平有效时使能计数器。一旦触发电平无效，计数器停止并复位。计数器的开始和停止都是受控的。

13.3.26 ADC 触发器

定时器可以用各种内部信号（例如重置、使能或比较事件）生成 ADC 触发事件。也可以生成由内部边缘检测器发出的脉冲，例如：OC4ref 的上升沿和下降沿、OC5ref 的上升沿或 OC6ref 的下降沿。触发器在 tim_trgo2 内部线路发出，该线路被重定向到 ADC。共有 16 个可能的事件，可以使用 TIM8_CR2 寄存器中的 MMS2[3:0]位选择。前章节：三相组合 PWM 模式中给出了一个 3 相马达驱动的应用示例。

*注意：接收*tim_trgo *或*tim_trgo2 *信号的从属外设（定时器、*ADC *等）的时钟必须在从主定时器接收事件之前使能，并且时钟频率（预分频器）在从主定时器接收触发器时不能更改。*

*注意：*ADC *的时钟必须在从主定时器接收事件之前使能，并且在接收到定时器的触发器时不能更改*

13.3.27 调试模式

当微控制器进入调试模式时（内核停止），TIM8 计数器会继续正常工作或者停止。

在调试模式下，每个定时器的行为可以通过 Debug 支持（DBG）模块的专用配置位进行编程。

出于安全考虑，当计数器停止时，输出被禁用（就像将 MOE 位复位一样）。输出可以强制处于非活动状态（OSSI 位=1），或者由 GPIO 控制器接管控制（OSSI 位=0）以强制其处于高阻态。

有关更多详细信息，请参阅调试部分。

13.4 TIM8 低功耗模式

表 13.12 低功耗模式对 TIM8 的影响

模式	描述
睡眠	无影响，外设是运行的。中断会导致设备退出睡眠模式。
停机	定时器操作被停止并保留其寄存器内容。无法生成中断。
待机	定时器被断电并且退出待机模式后必须重新初始化。

13.5 TIM8 中断

TIM8 可以生成多个中断，如下表所示。

表 13.13 中断请求

中断缩写	中断事件	事件标志	使能控制位	中断清除方法	从睡眠模式退出	从停机和待机退出
TIM_UP	更新	UIF	UIE	UIF 写 0	是	否
TIM_CC	捕获/比较 1	CC1IF	CC1IE	CC1IF 写 0	是	否
	捕获/比较 2	CC2IF	CC2IE	CC2IF 写 0	是	否
	捕获/比较 3	CC3IF	CC3IE	CC3IF 写 0	是	否
	捕获/比较 4	CC4IF	CC4IE	CC4IF 写 0	是	否
TIM_TRG_COM	换向（COM）	COMIF	COMIE	COMIF 写 0	是	否
TIM_TRG	触发	TIF	TIE	TIF 写 0	是	否
TIM_DIR_IDX	方向	DIRF	DIRIE	DIRF 写 0	是	否
TIM_BRK	刹车	BIF	BIE	BIF 写 0	是	否
	刹车 2	B2IF		B2IF 写 0	是	否
	系统中断	SBIF		SBIF 写 0	是	否
TIM_TER	转换错误	TERRF	TERRIE	TERRF 写 0	是	否

13.6 TIM8 寄存器

13.6.1 TIM8 控制寄存器 1（TIM8_CR1）

偏移地址：0x00
复位值：0x0000 0000
访问：无等待，支持字，半字和字节访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			DITHEN	UIFREMAP	Res	CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
			rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:13 保留，必须保持为复位值。

Bit 12 **DITHEN**：使能抖动（Dithering enable）

0：抖动禁止

1：抖动使能

*注：*DITHEN *位只能在*CEN *位复位时修改*

Bit 11 **UIFREMAP**：UIF 状态位重映射（UIF status bit remapping）

0：无重映射。UIF 状态位未复制到 TIM8_CNT 寄存器位 31

1：重映射使能。UIF 状态位复制到 TIM8_CNT 寄存器位 31

Bit 10 保留，必须保持为复位值。

Bits 9:8 **CKD[1:0]**：时钟分频（Clock division）

此位域指示定时器时钟（tim_ker_ck）频率与死区发生器以及数字滤波器（tim_etr_in,tim_tix）所使用的死区及采样时钟（t_{DTS}）之间的分频比

00：$t_{DTS} = t_{tim_ker_ck}$

01：$t_{DTS} = 2 \times t_{tim_ker_ck}$

10：$t_{DTS} = 4 \times t_{tim_ker_ck}$

11：保留，不要设置成此值

Bit 7 **ARPE**：自动重载预装载使能（Auto-reload preload enable）

0：TIM8_ARR 寄存器不进行缓冲；

1：TIM8_ARR 寄存器进行缓冲。

Bits 6:5 **CMS[1:0]**：中央对齐模式选择（Center-aligned mode selection）

00：边沿对齐模式。计数器根据方向位（DIR）递增计数或递减计数。

01：中心对齐模式 1。计数器交替进行递增计数和递减计数。仅当计数器递减计数时，配置为输出的通道（TIM8_CCMRx 寄存器中的 CxS=00）的输出比较中断标志才置 1。

10：中心对齐模式 2。计数器交替进行递增计数和递减计数。仅当计数器递增计数时，配置为输出的通道（TIM8_CCMRx 寄存器中的 CxS=00）的输出比较中断标志才置 1。

11：中心对齐模式 3。计数器交替进行递增计数和递减计数。当计数器递增计数或递减计数时，配置为输出的通道（TIM8_CCMRx 寄存器中的 CxS=00）的输出比较中断标志都会置 1。

注：只要计数器处于使能状态 (CEN=1)，*就不得从边沿对齐模式切换为中心对齐模式。*

Bit 4 **DIR**：方向（Direction）

0：计数器递增计数；
1：计数器递减计数。
注：当定时器配置为中心对齐模式时，该位为只读状态。

Bit 3 **OPM**：单脉冲模式（One pulse mode）
0：计数器在发生更新事件时不会停止计数；
1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）

Bit 2 **URS**：更新请求源（Update request source）
此位由软件置 1 和清零，用以选择 UEV 事件源。
0：使能时，所有以下事件都会生成更新中断请求。此类事件包括：
– 计数器上溢/下溢
– 将 UG 位置 1
– 通过从模式控制器生成的更新事件
1：使能时，只有计数器上溢/下溢会生成更新中断。

Bit 1 **UDIS**：更新禁止（Update disable）
此位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。
0：使能 UEV。更新(UEV)事件可通过以下事件之一生成：
– 计数器上出/下溢
– 将 UG 位置 1
– 通过从模式控制器生成的更新事件
然后更新影子寄存器的值。
1：禁止 UEV。不会生成更新事件，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。但如果将 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。

Bit 0 **CEN**：计数器使能（Counter enable）
0：禁止计数器；
1：使能计数器。
*注：只有事先通过软件将*CEN *位置*1*，才可以使用外部时钟和门控模式。而触发模式可通过硬件自动将*CEN *位置*1*。*

13.6.2 TIM8 控制寄存器 2（TIM8_CR2）

偏移地址：0x04
复位值：0x0000 0000
访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						MMS[3]	Res	MMS2[3:0]				Res	OIS6	Res	OIS5
						rw		rw	rw	rw	rw		rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	Res	MMS[2:0]			Res	CCUS	Res	CCPC
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw		rw		rw

Bits 31:26 保留，必须保持为复位值。

Bit 25 **MMS [3]**：主模式选择

Bit 24 保留，必须保持为复位值。

Bits 23:20 **MMS2[3:0]**：主模式选择 2（Master mode selection 2）
这些位允许将信息发送到 ADC 进行同步（tim_trgo2）以进行选择。组合如下：

0000：**复位** - 来自 TIM8_EGR 寄存器的 UG 位用作触发器输出（tim_trgo2）。如果重置是由触发器输入（配置为重置模式的从机模式控制器）生成的，则 tim_trgo2 上的信号与实际重置相比会有所延迟。

0001：**使能** - 计数器使能信号 CNT_EN 用作触发器输出（tim_trgo2）。它对于同时启动多个定时器或在一段时间内控制从机定时器的使能窗口是有用的。当在门控模式下配置时，计数器使能信号是由 CEN 控制位和触发器输入之间的逻辑 AND 生成的。当计数器使能信号由触发器输入控制时，除选择了主/从模式的情况外（请参见 TIM8_SMCR 寄存器中的 MSM 位描述），tim_trgo2 上会有延迟。

0010：**更新** - 选择更新事件作为触发器输出（tim_trgo2）。例如，主定时器可以用作从机定时器的预分频器。

0011：**比较脉冲** - 当 CC1IF 标志被设置时（即使它已经是高电平），当发生捕获或比较匹配时，触发输出会发送一个正脉冲（tim_trgo2）。

0100：**比较** - tim_oc1refc 信号用作触发器输出（tim_trgo2）

0101：**比较** - tim_oc2refc 信号用作触发器输出（tim_trgo2）

0110：**比较** - tim_oc3refc 信号用作触发器输出（tim_trgo2）

0111：**比较** - tim_oc4refc 信号用作触发器输出（tim_trgo2）

1000：**比较** - tim_oc5refc 信号用作触发器输出（tim_trgo2）

1001：**比较** - tim_oc6refc 信号用作触发器输出（tim_trgo2）

1010：**比较脉冲** - tim_oc4refc 上升或下降沿在 tim_trgo2 上生成脉冲

1011：**比较脉冲** - tim_oc6refc 上升或下降沿在 tim_trgo2 上生成脉冲

1100：**比较脉冲** - tim_oc4refc 或 tim_oc6refc 上升沿在 tim_trgo2 上生成脉冲

1101：**比较脉冲** - tim_oc4refc 上升或 tim_oc6refc 下降沿在 tim_trgo2 上生成脉冲

1110：**比较脉冲** - tim_oc5refc 或 tim_oc6refc 上升沿在 tim_trgo2 上生成脉冲

1111：**比较脉冲** - tim_oc5refc 上升或 tim_oc6refc 下降沿在 tim_trgo2 上生成脉冲

注：在从主定时器接收事件之前，必须启用从机定时器或 ADC 的时钟，并且在从主定时器接收触发器时不能随意更改。

Bit 19 保留，必须保持为复位值。

Bit 18 **OIS6**：输出空闲状态 6（OC6 输出）

参见 OIS1 位

Bit 17 保留，必须保持为复位值。

Bit 16 **OIS5**：输出空闲状态 5（OC5 输出）

参见 OIS1 位

Bit 15 保留，必须保持为复位值。

Bit 14 **OIS4**：输出空闲状态 4（OC4 输出）

参见 OIS1 位

Bit 13 **OIS3N**：输出空闲状态 3（OC3N 输出）

参见 OIS1 位

Bit 12 **OIS3**：输出空闲状态 3（OC3 输出）

参见 OIS1 位

Bit 11 **OIS2N**：输出空闲状态 2（OC2N 输出）

参见 OIS1 位

Bit 10 **OIS2**：输出空闲状态 2（OC2 输出）

参见 OIS1 位

Bit 9 **OIS1N**：输出空闲状态 1（OC1N 输出）

0：当 MOE=0 时，经过死区时间后 tim_oc1n=0

1：当 MOE=0 时，经过死区时间后 tim_oc1n=1

*注：只要编程了*LOCK（TIM8_BDTR *寄存器中的*LOCK *位）级别*1、2 *或*3，*此位即无法修改。*

Bit 8 **OIS1**：输出空闲状态 1（OC1 输出）

0：当 MOE=0 时，（如果 OC1N 有效，则经过死区时间之后）tim_oc1=0

1：当 MOE=0 时，（如果 OC1N 有效，则经过死区时间之后）tim_oc1=1

*注：只要编程了*LOCK（TIM8_BDTR *寄存器中的*LOCK *位）级别*1、2 *或*3，*此位即无法修改。*

Bit 7 保留，必须保持为复位值。

Bits 25, 6:4 **MMS[3:0]：**主模式选择

这些位可选择主模式下将要发送到从定时器以实现同步的信息(tim_trgo)。这些位的组合如下：

0000：**复位** – TIM8_EGR 寄存器中的 UG 位用作触发输出（tim_trgo）。如果复位由触发输入生成（从模式控制器配置为复位模式），则 TRGO 上的信号相比实际复位会有延迟。

0001：**使能** – 计数器使能信号 CNT_EN 用作触发输出（tim_trgo）。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号可由 CEN 控制位产生。当配置为门控模式时，也可由触发输入产生。当计数器使能信号由触发输入控制时，tim_trgo 上会存在延迟，选择主/从模式时除外（请参见 TIM8_SMCR 寄存器中 MSM 位的说明）。

0010：**更新** – 选择更新事件作为触发输出（tim_trgo）。例如，主定时器可用作从定时器的预分频器。

0011：**比较脉冲** – 一旦发生比较匹配事件，当 CC1IF 被置 1 时（即使已为高电平），触发输出都会发送一个正脉冲。(tim_trgo)。

0100：**比较** – OC1REF 信号用作触发输出（tim_trgo）

0101：**比较** – OC2REF 信号用作触发输出（tim_trgo）

0110：**比较** – OC3REF 信号用作触发输出（tim_trgo）

0111：**比较** – OC4REF 信号用作触发输出（tim_trgo）

其他编码保留

*注：在从主定时器接收事件之前，必须启用从机定时器或*ADC *的时钟，并且在从主定时器接收触发器时不能随意更改。*

Bit 3 保留，必须保持为复位值。

Bit 2 **CCUS**：捕获/比较控制更新选择

0：如果捕获/比较控制位（CCPC=1）进行预装载，仅通过将 COMG 位置 1 来对这些位进行更新；

1：如果捕获/比较控制位（CCPC=1）进行预装载，可通过将 COMG 位置 1 或 TRGI 的上升沿对这些位进行更新。

注：此位仅对具有互补输出的通道有效。。

Bit 1 保留，必须保持为复位值。

Bit 0 **CCPC**：捕获/比较预装载控制（Capture/compare preloaded control）

0：CCxE，CCxNE 和 OCxM 位未进行预装载；

1：CCxE，CCxNE 和 OCxM 位进行了预装载，写入这些位后，仅当发生换向事件（COM）（COMG 位置 1 或在 TRGI 上检测到上升沿，取决于 CCUS 位）时才会对这些位进行更新。

注：此位仅对具有互补输出的通道有效。

13.6.3 TIM8 从模式控制寄存器（TIM8_SMCR）

偏移地址：0x08

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						SMSPS	SMSPE	Reserved							SMS[3]
						rw	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	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:26 保留，必须保持为复位值。

Bit 25 **SMSPS**：SMS 预加载源（SMS preload source）

此位选择触发 SMS[3:0]位域从预加载到活动的事件来源：

0：由定时器的更新事件触发

1：由索引事件触发

Bit 24 **SMSPE**：SMS 预加载使能（SMS preload enable）

此位选择是否使能 SMS[3:0]位域的预加载：

0：SMS[3:0]位域不进行预加载

1：使能 SMS[3:0]预加载

Bits 23:17 保留，必须保持为复位值。

Bit 16 **SMS[3]**：从模式选择（Slave mode selection）

Bit 15 **ETP**：外部触发极性（External trigger polarity）

此位可选择将 tim_etr_in 还是 tim_etr_in 的反相用于触发操作

0：tim_etr_in 未反相，高电平或上升沿有效。

1：tim_etr_in 反相，低电平或下降沿有效。

Bit 14 **ECE**：外部时钟使能位（External clock enable）

此位可使能外部时钟模式 2。

0：禁止外部时钟模式 2

1：使能外部时钟模式 2。计数器时钟由 tim_etrf 信号的任意有效边沿提供。

注：将 ECE 位置 1 与选择外部时钟模式 1 并将 tim_trgi 连接到 tim_etrf（SMS=111 且 TS=00111）具有相同效果。

外部时钟模式 2 可以和以下从模式同时使用：复位模式、门控模式和触发模式。不过此类情况下 tim_trgi 不得连接 tim_etrf（TS 位不得为 00111）。

如果同时使能外部时钟模式 1 和外部时钟模式 2，则外部时钟输入为 tim_etrf。

Bits 13:12 **ETPS[1:0]**：外部触发预分频器（External trigger prescaler）

外部触发信号 tim_etrp 频率不得超过 TIMxCLK 频率的 1/4。可通过使能预分频器来降低 tim_etrp 频率。这种方法在 tim_etr_in 输入快速外部时钟时非常有用。

00：预分频器关闭

01：2 分频 tim_etr_in 频率
10：4 分频 tim_etr_in 频率
11：8 分频 tim_etr_in 频率

Bits 11:8 **ETF[3:0]**：外部触发滤波器（External trigger filter）
此位域可定义 tim_etrp 信号的采样频率和适用于 tim_etrp 的数字滤波时间。数字滤波器由事件计数器组成，每 N 个事件才视为一个有效边沿：
0000：无滤波器，按 f_{DTS} 频率进行采样
0001：$f_{SAMPLING}=f_{CK_INT}$，N=2。
0010：$f_{SAMPLING}=f_{CK_INT}$，N=4。
0011：$f_{SAMPLING}=f_{CK_INT}$，N=8。
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6。
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8。
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6。
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8。
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6。
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8。
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5。
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6。
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8。
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5。
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6。
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8。

Bit 7 **MSM**：主/从模式（Master/slave mode）
0：不执行任何操作；
1：当前定时器的触发输入事件(tim_trgi)的动作被推迟，以使当前定时器与其从定时器实现完美同步（通过 tim_trgo）。此设置适用于由单个外部事件对多个定时器进行同步的情况。

Bits 6:4 **TS[2:0]**：触发选择（Trigger selection）
此位域可选择将要用于同步计数器的触发输入。
000：内部触发 0（tim_itr0）
001：内部触发 1（tim_itr1）
010：内部触发 2（tim_itr2）
111：外部触发输入（tim_etrf）
有关定时器 ITRx 含义的详细信息，请参见章节 TIM8 引脚和内部信号中的表格：TIM8 内部触发连接。

注：这些位只能在未使用的情况下（例如，SMS=000 时）进行更改，以避免转换时出现错误的边沿检测。

Bit 3 **OCCS**：OCREF 清除源选择（OCREF clear selection）
此位用于选择 OCREF 清除源。
0：tim_ocref_clr_int 信号连接到 tim_ocref_clr 输入
1：tim_ocref_clr_int 信号连接到 tim_etrf

Bits 16, 2:0 **SMS[3:0]**：从模式选择（Slave mode selection）
当选择了外部信号，触发信号（tim_trgi）的有效边沿与选中的外部输入极性相关（见

输入控制寄存器和控制寄存器的说明）

0000：**关闭从模式** – 如果 CEN=1，则预分频器直接由内部时钟驱动。

0100：**复位模式** – 选中的触发输入（tim_trgi）的上升沿重新初始化计数器，并且产生一个更新寄存器的信号。

0101：**门控模式** – 当触发输入（tim_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。

0110：**触发模式** – 计数器在触发输入 tim_trgi 的上升沿启动（但不复位），只有计数器的启动是受控的。

0111：**外部时钟模式 1** – 选中的触发输入（tim_trgi）的上升沿驱动计数器。

1000：**组合重置+触发器模式** - 选定触发器输入（tim_trgi）的上升沿会重新初始化计数器，生成寄存器的更新并启动计数器。

1001：**组合门控+复位模式**-当触发器输入（tim_trgi）为高电平时，计数器时钟使能。一旦触发器变为低电平，计数器将停止并重置。同时控制计数器的启动和停止。

注：如果选择 tim_ti1f_ed *作为触发输入（*TS=100*），则不得使用门控模式。实际上* tim_ti1f_ed *为* TI1F *上的每个过渡输出*1 *个脉冲，而门控模式检查触发信号的电平。*

注：接收 tim_trgo *或* tim_trgo2 *信号的从机外设（定时器、*ADC *等）的时钟必须在从主定时器接收事件之前使能，并且时钟频率（预分频器）在从主定时器接收触发器时不能随意更改。*

13.6.4 TIM8/DMA/中断使能寄存器（TIM8_DIER）

偏移地址：0x0C

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														CC6IE	Res
														rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC5IE	Reserved							BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
rw								rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:18 保留，必须保持为复位值。

Bit 17 **CC6IE**：捕获/比较 6 中断使能（Capture/Compare 6 interrupt enable）

0：禁止 CC6 中断；

1：使能 CC6 中断。

Bit 16 保留，必须保持为复位值。

Bit 15 **CC5IE**：捕获/比较 5 中断使能（Capture/Compare 5 interrupt enable）

0：禁止 CC5 中断；

1：使能 CC5 中断。

Bits 14:8 保留，必须保持为复位值。

Bit 7 **BIE**：刹车中断使能（Break interrupt enable）

0：禁止刹车中断；

1：使能刹车中断。

Bit 6 **TIE**：触发信号（TGRI）中断使能（Trigger interrupt enable）

0：禁止触发信号(TGRI) 中断；

1：使能触发信号(TGRI) 中断。

Bit 5 **COMIE**：COM 中断使能（COM interrupt enable）

0：禁止 COM 中断；

1：使能 COM 中断。

Bit 4 **CC4IE**：捕获/比较 4 中断使能（Capture/Compare 4 interrupt enable）

0：禁止 CC4 中断；

1：使能 CC4 中断。

Bit 3 **CC3IE**：捕获/比较 3 中断使能（Capture/Compare 3 interrupt enable）

0：禁止 CC3 中断；

1：使能 CC3 中断。

Bit 2 **CC2IE**：捕获/比较 2 中断使能（Capture/Compare 2 interrupt enable）

0：禁止 CC2 中断；

1：使能 CC2 中断。

Bit 1 **CC1IE**：捕获/比较 1 中断使能（Capture/Compare 1 interrupt enable）

0：禁止 CC1 中断；

1：使能 CC1 中断。

Bit 0 **UIE**：更新中断使能（Update interrupt enable）

0：禁止更新中断；

1：使能更新中断。

13.6.5 TIM8 状态寄存器（TIM8_SR）

偏移地址：0x10

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														CC6IF	CC5IF
														rc w0	rc w0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		SBIF	Reserved				B2IF	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
		rc w0					rc w0	rc w0	rc w0	rc w0	rc w0	rc w0	rc w0	rc w0	rc w0

Bits 31:18 保留，必须保持为复位值。

Bit 17 **CC6IF**：比较 6 中断标志

参考 CC1IF 描述

*注：*Channel6 *只能被配置为输出。*

Bit 16 **CC5IF**：比较 5 中断标志

参考 CC1IF 描述

*注：*Channel5 *只能被配置为输出。*

Bits 15:14 保留，必须保持为复位值。

Bit 13 **SBIF**：系统刹车中断标志（System break interrupt flag）

该标志由硬件在系统刹车输入激活时立即设置。如果系统刹车输入不处于有效状态，则可以通过软件清除该标志。

要重新启动 PWM 操作，必须将此标志重置为 0。

0：未发生刹车事件。

1：在系统刹车输入上检测到有效电平。如果 TIM8_DIER 寄存器中的 BIE=1，则生成中断。

Bits 12:9 保留，必须保持为复位值。

Bit 8 **BI2F**：刹车 2 中断标志（Break 2 interrupt flag）

只要刹车 2 输入变为有效状态，此标志便由硬件置 1。刹车 2 输入无效后可通过软件对其清零。

0：无刹车事件产生；

1：刹车 2 输入上检测到有效电平。如果 TIM8_DIER 寄存器上 BIE=1 则产生一个中断。

Bit 7 **BIF**：刹车中断标志（Break interrupt flag）

只要刹车输入变为有效状态，此标志便由硬件置 1。刹车输入无效后可通过软件对其清零。

0：未发生刹车事件。

1：刹车输入上检测到有效电平。

Bit 6 **TIF**：触发中断标志（Trigger interrupt flag）

在除门控模式以外的所有模式下，当使能从模式控制器后在 tim_trgi 输入上检测到有效边沿时，该标志将由硬件置 1。选择门控模式时，该标志将在计数器启动或停止时置 1。但需要通过软件清零。

0：未发生触发事件。

1：触发中断挂起。

Bit 5 **COMIF**：COM 中断标志（COM interrupt flag）

此标志在发生 COM 事件时（捕获/比较控制位 CCxE、CCxNE 和 OCxM 已更新时）由硬件置 1。但需要通过软件清零。

0：未发生 COM 事件。

1：COM 中断挂起。

Bit 4 **CC4IF**：捕获/比较 4 中断标志（Capture/Compare 4 interrupt flag）

参考 CC1IF 描述。

Bit 3 **CC3IF**：捕获/比较 3 中断标志（Capture/Compare 3 interrupt flag）

参考 CC1IF 描述。

Bit 2 **CC2IF**：捕获/比较 2 中断标志（Capture/Compare 2 interrupt flag）

参考 CC1IF 描述。

Bit 1 **CC1IF**：捕获/比较 1 中断标志（Capture/Compare 1 interrupt flag）

如果通道 CC1 配置为输出：

当计数器与比较值匹配时，此标志由硬件置 1，中央对齐模式下除外（请参见 TIM8_CR1 寄存器中的 CMS 位说明）。但需要通过软件清零。

0：不匹配；

1：TIM8_CNT 计数器的值与 TIM8_CCR1 寄存器的值匹配。当 TIM8_CCR1 的值大于 TIM8_ARR 的值时，CC1IF 位将在计数器发生上溢（递增计数模式和增减计数模式下）或下溢（递减计数模式下）时变为高电平。

Bit 0 **UIF**：更新中断标志（Update interrupt flag）

该位在发生更新事件时通过硬件置 1。但需要通过软件清零。

0：未发生更新。

1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：

- TIM8_CR1 寄存器中的 UDIS=0，并且重复计数器值上溢或下溢时（重复计数器= 0 时更新）。
- TIM8_CR1 寄存器中的 URS=0 且 UDIS=0，并且由软件使用 TIM8_EGR 寄

存器中的 UG 位重新初始化 CNT 时。

– TIM8_CR1 寄存器中的 URS=0 且 UDIS=0，并且 CNT 由触发事件重新初始化时（请参见章节：TIM8 从模式控制寄存器(TIM8_SMCR)）。

13.6.6 TIM8 事件产生寄存器（TIM8_EGR）

偏移地址：0x14

复位：0x0000 0000

访问：无等待，支持字，半字和字节访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							B2G	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
							w	w	w	w	w	w	w	w	w

Bits 15:9 保留，必须保持为复位值。

Bit 8 **B2G**：刹车 2 生成（Break generation）

此位由软件置 1 以生成事件，并由硬件自动清零。

0：不执行任何操作；

1：生成刹车 2 事件。MOE 位清零且 BIF 标志置 1。使能后可发生相关中断事件。

Bit 7 **BG**：刹车生成（Break generation）

此位由软件置 1 以生成事件，并由硬件自动清零。

0：不执行任何操作；

1：生成刹车事件。MOE 位清零且 BIF 标志置 1。使能后可发生相关中断事件。

Bit 6 **TG**：生成触发信号（Trigger generation）

此位由软件置 1 以生成事件，并由硬件自动清零。

0：不执行任何操作；

1：TIM8_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断事件。

Bit 5 **COMG**：捕获/比较控制更新生成（Capture/Compare control update generation）

该位可通过软件置 1，并由硬件自动清零

0：不执行任何操作；

1：CCPC 位置 1 时，可更新 CCxE、CCxNE 和 OCxM 位。

注：此位仅对具有互补输出的通道有效。

Bit 4 **CC4G**：捕获/比较 4 生成（Capture/Compare 4 generation）

参考 CC1G 描述。

Bit 3 **CC3G**：捕获/比较 3 生成（Capture/Compare 3 generation）

参考 CC1G 描述。

Bit 2 **CC2G**：捕获/比较 2 生成（Capture/Compare 2 generation）

参考 CC1G 描述。

Bit 1 **CC1G**：捕获/比较 1 生成（Capture/Compare 1 generation）

此位由软件置 1 以生成事件，并由硬件自动清零。

0：不执行任何操作；

1：通道 1 上生成捕获/比较事件：

若通道 CC1 配置为输出：

使能时，CC1IF 标志置 1 并发送相应的中断。

Bit 0 **UG**：更新生成（Update generation）

该位可通过软件置 1，并由硬件自动清零。

0：不执行任何操作；

1：重新初始化计数器并生成一个寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。如果选择中心对齐模式或 DIR=0（递增计数），计数器将清零；如果 DIR=1（递减计数），计数器将使用自动重载值（TIM8_ARR）。

13.6.7 TIM8 捕获/比较模式寄存器 1（TIM8_CCMR1）

偏移地址：0x18
复位值：0x0000 0000
访问：无等待，支持字，半字和字节访问
这些通道可用于输出（比较模式）模式。OCxx 用于说明通道配置为输出时该位对应的功能。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							OC2M[3]	Reserved							OC1M[3]
							rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]			OC2PE	OC2FE	Reserved		OC1CE	OC1M[2:0]			OC1PE	OC1FE	Reserved	
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw		

输出比较模式

Bits 31:25 保留，必须保持为复位值。
Bit 24 **OC2M[3]：**输出比较 2 模式（Output compare 2mode）
Bits 23:17 保留，必须保持为复位值。
Bit 16 **OC1M[3]：**输出比较 1 模式（Output compare 1 mode）
Bit 15 **OC2CE：**输出比较 2 清零使能（Output Compare 2 clear enable）
Bits 24, 14:12 **OC2M[3:0]：**输出比较 2 模式（Output Compare 2 mode）
参考 OC1M[3:0]位的描述
Bit 11 **OC2PE：**输出比较 2 预装载使能（Output Compare 2 preload enable）
Bit 10 **OC2FE：**输出比较 2 快速使能（Output Compare 2 fast enable）
Bits 9:8 保留，必须保持为复位值。
Bit 7 **OC1CE：**输出比较 1 清零使能（Output Compare 1 clear enable）
0：tim_oc1ref 不受 tim_ocref_clr_int 信号影响；
1：tim_oc1ref 在 tim_ocref_clr_int 信号中被检测到高电平，tim_oc1ref 立即清零。（tim_ocref_clr 输入或 tim_etrf 输入）
Bits 16, 6:4 **OC1M[3:0]：**输出比较 1 模式（Output Compare 1 mode）
这些位定义提供 tim_oc1 和 tim_oc1n 的输出参考信号 tim_oc1ref 的行为。tim_oc1ref 为高电平有效，而 tim_oc1 和 tim_oc1n 的有效电平则取决于 CC1P 位和 CC1NP 位。
0000：**冻结** - 输出比较寄存器 TIM8_CCR1 与计数器 TIM8_CNT 进行比较不会对输出造成任何影响。（该模式用于生成时基）。
0001：将通道 1 设置为匹配时输出有效电平。当计数器 TIM8_CNT 与捕获/比较寄存器 1（TIM8_CCR1）匹配时，tim_oc1ref 信号强制变为高电平。
0010：将通道 1 设置为匹配时输出无效电平。当计数器 TIM8_CNT 与捕获/比较寄存器 1（TIM8_CCR1）匹配时，tim_oc1ref 信号强制变为低电平。
0011：**翻转** – TIM8_CNT=TIM8_CCR1 时，tim_oc1ref 发生翻转。
0100：**强制变为无效电平** – tim_oc1ref 强制变为低电平。
0101：**强制变为有效电平** – tim_oc1ref 强制变为高电平。
0110：**PWM 模式 1**-在递增计数模式下，只要 TIM8_CNT<TIM8_CCR1，通道 1 便为有

效状态，否则为无效状态。在递减计数模式下，只要 TIM8_CNT>TIM8_CCR1，通道 1 便为无效状态（tim_oc1ref=0），否则为有效状态（tim_oc1ref=1）。

0111：**PWM 模式 2** - 在递增计数模式下，只要 TIM8_CNT<TIM8_CCR1，通道 1 便为无效状态，否则为有效状态。在递减计数模式下，只要 TIM8_CNT>TIM8_CCR1，通道 1 便为有效状态，否则为无效状态。

1000：**可重触发 OPM 模式 1** - 在向上计数模式下，通道在检测到触发事件（tim_trgi 信号）之前一直处于活动状态。然后，会像 PWM 模式 1 那样执行比较，并在下次更新时使通道再次处于活动状态。在向下计数模式下，通道在检测到触发事件（tim_trgi 信号）之前一直处于非活动状态。然后，会像 PWM 模式 1 那样执行比较，并在下次更新时使通道再次处于非活动状态。

1001：**可重触发 OPM 模式 2** - 在向上计数模式下，通道在检测到触发事件（tim_trgi 信号）之前一直处于非活动状态。然后，会像 PWM 模式 2 那样执行比较，并在下次更新时使通道再次处于非活动状态。在向下计数模式下，通道在检测到触发事件（tim_trgi 信号）之前一直处于活动状态。然后，会像 PWM 模式 1 那样执行比较，并在下次更新时使通道再次处于活动状态。

1010：保留。

1011：保留。

1100：**组合 PWM 模式 1** - tim_oc1ref 的行为与 PWM 模式 1 相同。tim_oc1refc 是 tim_oc1ref 和 tim_oc2ref 之间的逻辑 OR。

1101：**组合 PWM 模式 2** - tim_oc1ref 的行为与 PWM 模式 2 相同。tim_oc1refc 是 tim_oc1ref 和 tim_oc2ref 之间的逻辑 AND。

1110：**非对称 PWM 模式 1** - tim_oc1ref 的行为与 PWM 模式 1 相同。当计数器向上计数时，tim_oc1refc 输出 tim_oc1ref；当计数器向下计数时，tim_oc1refc 输出 tim_oc2ref。

1111：**非对称 PWM 模式 2** - tim_oc1ref 的行为与 PWM 模式 2 相同。当计数器向上计数时，tim_oc1refc 输出 tim_oc1ref；当计数器向下计数时，tim_oc1refc 输出 tim_oc2ref。

*注：只要编程了*LOCK *（*TIM8_BDTR *寄存器中的*LOCK *位）级别*3 *且*CC1S= *“*00*”（通道配置为输出），这些位即无法修改。*

*注：在*PWM *模式*1 *或*PWM *模式*2 *下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“*PWM*”模式时，*OCREF *电平才会发生更改。*

*注：此位域将在具有互补输出的通道上进行预装载。如果*TIM8_CR2 *寄存器中的*CCPC *位置*1*，则仅当生成*COM *事件时，*OC1M *有效位才会从预装载位获取新值。*

Bit 3 **OC1PE**：输出比较 1 预装载使能（Output Compare 1 preload enable）

0：禁止与 TIM8_CCR1 相关的预装载寄存器。可随时向 TIM8_CCR1 写入数据，写入后将立即使用新值。

1：使能与 TIM8_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIM8_CCR1 预装载值在每次生成更新事件时都会装载到活动寄存器中。

*注：只要编程了*LOCK *（*TIM8_BDTR *寄存器中的*LOCK *位）级别*3 *且*CC1S= *“*00*”（通道配置为输出），这些位即无法修改。*

Bit 2 **OC1FE**：输出比较 1 快速使能（Output Compare 1 fast enable）

此位减少了触发事件和定时器输出上的转换之间的延迟。它必须在单脉冲模式（在

TIM8_CR1 寄存器中设置 OPM 位）中使用，以便在启动触发后尽快开始输出脉冲。

0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的最短延迟时间为 5 个时钟周期。

1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OCFE 才会起作用。

Bits 1:0 保留，必须保持复位值。

13.6.8 TIM8 捕获/比较模式寄存器 2（TIM8_CCMR2）

偏移地址：0x1C

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

参看以上 CCMR1 寄存器的描述

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							OC4M[3]	Reserved							OC3M[3]
							rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	OC4FE	Reserved		OC3CE	OC3M[2:0]			OC3PE	OC3FE	Reserved	
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw		

输出比较模式：

Bits 31:25 保留，必须保持为复位值。

Bit 24 **OC4M[3]：**输出比较 4 模式（Output Compare 4 mode）

Bits 23:17 保留，必须保持为复位值。

Bit 16 **OC3M[3]：**输出比较 3 模式（Output compare 3 mode）

Bit 15 **OC4CE：**输出比较 4 清零使能（Output Compare 4 clear enable）

Bits 24, 14:12 **OC4M[3:0]：**输出比较 4 模式（Output Compare 4 mode）

参考 OC3M[3:0]描述

Bit 11 **OC4PE：**输出比较 4 预装载使能（Output Compare 4 preload enable）

Bit 10 **OC4FE：**输出比较 4 快速使能（Output Compare 4 fast enable）

Bits 9:8 保留，必须保持为复位值。

Bit 7 **OC3CE：**输出比较 3 清零使能（Output Compare 3 clear enable）

Bits 16, 6:4 **OC3M[3:0]：**输出比较 3 模式（Output Compare 3 mode）

这些位定义提供 tim_oc3 和 tim_oc3n 的输出参考信号 tim_oc3ref 的行为。tim_oc3ref 为高电平有效，而 tim_oc3 和 tim_oc3n 的有效电平则取决于 CC3P 位和 CC3NP 位。

0000：**冻结** - 输出比较寄存器 TIM8_CCR3 与计数器 TIM8_CNT 进行比较不会对输出造成任何影响。（该模式用于生成时基）。

0001：将通道 3 设置为匹配时输出有效电平。当计数器 TIM8_CNT 与捕获/比较寄存器 3（TIM8_CCR3）匹配时，tim_oc3ref 信号强制变为高电平。

0010：将通道 3 设置为匹配时输出无效电平。当计数器 TIM8_CNT 与捕获/比较寄存器 3（TIM8_CCR3）匹配时，tim_oc3ref 信号强制变为低电平。

0011：**翻转** - TIM8_CNT=TIM8_CCR3 时，tim_oc3ref 发生翻转。

0100：**强制变为无效电平** - tim_oc3ref 强制变为低电平。

0101：**强制变为有效电平** - tim_oc3ref 强制变为高电平。

0110：**PWM 模式 1**-在递增计数模式下，只要 TIM8_CNT<TIM8_CCR3，通道 3 便为有效状态，否则为无效状态。在递减计数模式下，只要 TIM8_CNT>TIM8_CCR3，通道 3 便为无效状态（tim_oc3ref=0），否则为有效状态（tim_oc3ref=1）。

0111：**PWM 模式 2**- 在递增计数模式下，只要 TIM8_CNT<TIM8_CCR3，通道 3 便为无效状态，否则为有效状态。在递减计数模式下，只要 TIM8_CNT>TIM8_CCR3，通道 3 便为有效状态，否则为无效状态。

1000：**可重触发 OPM 模式 1**- 在向上计数模式下，通道在检测到触发事件（tim_trgi 信号）之前一直处于活动状态。然后，会像 PWM 模式 1 那样执行比较，并在下次更新时使通道再次处于活动状态。在向下计数模式下，通道在检测到触发事件（tim_trgi 信号）之前一直处于非活动状态。然后，会像 PWM 模式 1 那样执行比较，并在下次更新时使通道再次处于非活动状态。

1001：**可重触发 OPM 模式 2**- 在向上计数模式下，通道在检测到触发事件（tim_trgi 信号）之前一直处于非活动状态。然后，会像 PWM 模式 2 那样执行比较，并在下次更新时使通道再次处于非活动状态。在向下计数模式下，通道在检测到触发事件（tim_trgi 信号）之前一直处于活动状态。然后，会像 PWM 模式 1 那样执行比较，并在下次更新时使通道再次处于活动状态。

1010：**比较脉冲输出**：当 CCR3 匹配事件发生时，根据 TIM8_ECR 中的 PWPRSC[2:0]和 PW[7:0]位字段编程，产生 tim_oc3ref 脉冲。

1011：**方向输出**。tim_oc3ref 信号被 DIR 位的副本覆盖。

1100：**组合 PWM 模式 1**- tim_oc3ref 的行为与 PWM 模式 1 相同。tim_oc3refc 是 tim_oc3ref 和 tim_oc4ref 之间的逻辑 OR。

1101：**组合 PWM 模式 2**- tim_oc3ref 的行为与 PWM 模式 2 相同。tim_oc3refc 是 tim_oc3ref 和 tim_oc4ref 之间的逻辑 AND。

1110：**非对称 PWM 模式 1**- tim_oc3ref 的行为与 PWM 模式 1 相同。当计数器向上计数时，tim_oc3refc 输出 tim_oc3ref；当计数器向下计数时，tim_oc3refc 输出 tim_oc4ref。

1111：**非对称 PWM 模式 2**- tim_oc3ref 的行为与 PWM 模式 2 相同。当计数器向上计数时，tim_oc3refc 输出 tim_oc3ref；当计数器向下计数时，tim_oc3refc 输出 tim_oc4ref。

*注：只要编程了*LOCK *（*TIM8_BDTR *寄存器中的*LOCK *位）级别*3 *且*CC1S= *“*00*”（通道配置为输出），这些位即无法修改。*

*注：在*PWM *模式*1 *或*PWM *模式*2 *下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“*PWM*”模式时，*OCREF *电平才会发生更改。*

*注：此位域将在具有互补输出的通道上进行预装载。如果*TIM8_CR2 *寄存器中的*CCPC *位置*1*，则仅当生成*COM *事件时，*OC1M *有效位才会从预装载位获取新值。*

Bit 3 **OC3PE**：输出比较 3 预装载使能（Output Compare 3 preload enable）

Bit 2 **OC3FE**：输出比较 3 快速使能（Output Compare 3 fast enable）

Bits 1:0 保留，必须保持为复位值。

13.6.9 TIM8 捕获/比较使能寄存器（TIM8_CCER）

偏移地址：0x20

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										CC6P	CC6E	Reserved		CC5P	CC5E
										rw	rw			rw	rw

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

Bits 31:22 保留，必须保持为复位值。

Bit 21 **CC6P**：输入/捕获 6 输出极性（Capture/Compare 6 output polarity）

参考 CC1P 的描述。

Bit 20 **CC6E**：输入/捕获 6 输出使能（Capture/Compare 6 output enable）

参考 CC1E 的描述。

Bits 19:18 保留，必须保持为复位值。

Bit 17 **CC5P**：输入/捕获 5 输出极性（Capture/Compare 5 output polarity）

参考 CC1P 的描述。

Bit 16 **CC5E**：输入/捕获 5 输出使能（Capture/Compare 5 output enable）

参考 CC1E 的描述。

Bits 15:14 保留，必须保持复位值。

Bit 13 **CC4P**：输入/捕获 4 输出极性（Capture/Compare 4 output polarity）

参考 CC1P 的描述。

Bit 12 **CC4E**：输入/捕获 4 输出使能（Capture/Compare 4 output enable）

参考 CC1E 的描述。

Bit 11 **CC3NP**：输入/捕获 3 互补输出极性（Capture/Compare 3 complementary output polarity）

参考 CC1NP 的描述。

Bit 10 **CC3NE**：输入/捕获 3 互补输出使能（Capture/Compare 3 complementary output enable）

参考 CC1NE 的描述。

Bit 9 **CC3P**：输入/捕获 3 输出极性（Capture/Compare 3 output polarity）

参考 CC1P 的描述。

Bit 8 **CC3E**：输入/捕获 3 输出使能（Capture/Compare 3 output enable）

参考 CC1E 的描述。

Bit 7 **CC2NP**：输入/捕获 2 互补输出极性（Capture/Compare 2 complementary output polarity）

参考 CC1NP 的描述。

Bit 6 **CC2NE**：输入/捕获 2 互补输出使能（Capture/Compare 2 complementary output enable）

参考 CC1NE 的描述。

Bit 5 **CC2P**：输入/捕获 2 输出极性（Capture/Compare 2 output polarity）

参考 CC1P 的描述。

Bit 4 **CC2E**：输入/捕获 2 输出使能（Capture/Compare 2 output enable）

参考 CC1E 的描述。

Bit 3 **CC1NP**：输入/捕获 1 互补输出极性（Capture/Compare 1 complementary output polarity）

CC1 通道配置为输出：

0：OC1N 高电平有效；
1：OC1N 低电平有效。

注：此位将在具有互补输出的通道上进行预装载。如果TIM8_CR2 寄存器中的CCPC 位置1，则仅当生成换向事件时，CC1NP 有效位才会从预装载位获取新值。

注：只要编程了LOCK（TIM8_BDTR 寄存器中的LOCK 位）级别2 或3 且CC1S=00（通道配置为输出），此位立即变为不可写状态。

Bit 2 **CC1NE**：输入/捕获 1 互补输出使能（Capture/Compare 1 complementary output enable）

0：关闭－OC1N 禁止输出，因此 OC1N 的电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的值。
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：此位将在具有互补输出的通道上进行预装载。如果TIM8_CR2 寄存器中的CCPC 位置1，则仅当生成换向事件时，CC1NE 有效位才会从预装载位获取新值。

Bit 1 **CC1P**：输入/捕获 1 输出极性（Capture/Compare 1 output polarity）

CC1 通道配置为输出：

0：OC1 高电平有效；
1：OC1 低电平有效。

注：只要编程了LOCK（TIM8_BDTR 寄存器中的LOCK 位）级别2 或3，此位立即变为不可写状态。

注：此位将在具有互补输出的通道上进行预装载。如果TIM8_CR2 寄存器中的CCPC 位置1，则仅当生成换向事件时，CC1P 有效位才会从预装载位获取新值。

Bit 0 **CC1E**：输入/捕获 1 输出使能（Capture/Compare 1 output enable）

CC1 通道配置为输出：

0：关闭 － OC1 禁止输出，因此 OC1 的输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。
1：开启 － OC1 信号输出到对应的输出引脚，其输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值。

注：此位将在具有互补输出的通道上进行预装载。如果TIM8_CR2 寄存器中的CCPC 位置1，则仅当生成换向事件时，CC1E 有效位才会从预装载位获取新值。

表 13.14 带刹车功能的互补通道 tim_ocx 和 tim_ocxn 的输出控制位

控制位					输出状态 (1)	
MOE 位	OSSI 位	OSSR 位	CCxE 位	CCxNE 位	tim_ocx 输出状态	tim_ocxn 输出状态
1	X	X	0	0	禁止输出（不由定时器驱动：高阻态） tim_ocx=0，tim_ocxn=0	
		0	0	1	禁止输出（不由定时器驱动：高阻态） tim_ocx=0	tim_ocxref + 极性 tim_ocxn = tim_ocxref 异或 CCxNP
		0	1	0	tim_ocxref + 极性 tim_ocx = tim_ocxref 异或 CCxP	禁止输出（不由定时器驱动：高阻态） tim_ocxn=0
		X	1	1	tim_ocxref +极性+死区	tim_ocxref 互补项(对 tim_ocxref 进行“非”运算）+极性+死区
		1	0	1	关闭状态（输出使能为无效状态） tim_ocx=CCxP	tim_ocxref + 极性 tim_ocxn = tim_ocxref 异或 CCxNP
		1	1	0	tim_ocxref + 极性 tim_ocx = tim_ocxref 异或 CCxP	关闭状态（输出使能为无效状态） tim_ocxn=CCxNP
0	0	X	X	X	禁止输出（不再由定时器驱动）。 输出状态由 GPIO 控制器定义，可以是高电平、低电平或高阻态。	
	1		0	0		
			0	1	关闭状态（输出使能为无效状态） 异步：tim_ocx=CCxP、tim_ocxn=CCxNP （如果触发 tim_brk 或 tim_brk2）。 随后，如果存在时钟：在死区后 tim_ocx=OISx 且 tim_ocxn=OISxN，假定 OISx 和 OISxN 并没有都设置成 tim_ocx 及 tim_ocxn 的有效电平（否则在半桥配置下驱动开关时可能导致短路）。	
			1	0		
			1	1		

1. 如果一个通道的两个输出均未使用（CCxE = CCxNE = 0，由 GPIO 接管控制控制），则 OISx、OISxN、CCxP 和 CCxNP 位必须保持清零状态。

注： *与互补通道* tim_ocx *和* tim_ocxn *相连的外部* I/O *引脚的状态取决于通道* tim_ocx *和* tim_ocxn *的状态以及* GPIO *寄存器。*

13.6.10 TIM8 计数器（TIM8_CNT）

偏移地址：0x24

复位值：0x0000

访问：无等待，支持字，半字和字节访问

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

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

Bit 31 **UIFCPY**：UIF 副本（UIF copy）

该位是 TIM8_SR 寄存器中 UIF 位的一个只读副本。如果 TIM8_CR1 上的 UIFREMAP 位被清零，则位 31 被保留并读为 0。

Bits 30:16 保留，必须保持为复位值。

Bits 15:0 **CNT[15:0]：**计数器的值（Counter value）

在非抖动模式下（DITHEN = 0）

该寄存器（CNT[15:0]）保存的是计数器的值。

在抖动模式下（DITHEN = 1）

该寄存器（CNT[15:0]）只保存计数器的不抖动部分。抖动部分不可用。

13.6.11 TIM8 预分频器（TIM8_PSC）

偏移地址：0x28

复位值：0x0000

访问：无等待，支持字，半字和字节访问

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

Bits 15:0 **PSC[15:0]：**预分频器的值（Prescaler value）

计数器时钟频率（$f_{tim_cnt_ck}$）等于 $f_{tim_psc_ck}$ /（PSC[15:0] + 1）。

PSC 包含每次发生更新事件（包括计数器通过 TIM8_EGR 寄存器中的 UG 位清零时，或在配置为“复位模式”时通过触发控制器清零时）时要装载到活动预分频器寄存器的值。

13.6.12 TIM8 自动重装载寄存器（TIM8_ARR）

偏移地址：0x2C

复位：0x0000 FFFF

访问：无等待，支持字，半字和字节访问

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

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

Bits 31:20 保留，必须保持为复位值。

Bits 19:0 **ARR[19:0]：**自动重装载的值（Auto-reload value）

ARR 为要装载到实际自动重载寄存器的值。

有关 ARR 更新和行为的详细信息，请参考章节 TIM8 功能描述：时基单元。

当自动重载值为空时，计数器不工作。

在非抖动模式下（DITHEN = 0）

该寄存器存储的是自动重新加载的值

在抖动模式下（DITHEN = 1）

该寄存器存储的是 ARR[19:4]中的整数部分，而 ARR[3:0]位域存储的是抖动的部分。

13.6.13 TIM8 重复计数寄存器（TIM8_RCR）

偏移地址：0x30

复位值：0x0000

访问：无等待，支持字，半字和字节访问

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

Bits 15:0 **REP[15:0]**：重复计数器的值（Repetition counter value）

使能预装载寄存器时，用户可通过这些位设置比较寄存器的更新频率（即从预装载寄存器向活动寄存器周期性传输数据）；使能更新中断时，也可设置更新中断的生成速率。

与 REP_CNT 相关的减计数器每次计数到 0 时，都将生成一个更新事件并且计数器从 REP 值重新开始计数。由于只有生成重复更新事件 U_RC 时，REP_CNT 才会重载 REP 值，因此在生成下一重复更新事件之前，无论向 TIM8_RCR 寄存器写入何值都无影响。

这意味着 PWM 模式下（REP+1）相当于：

– 边沿对齐模式下的 PWM 周期数；

– 中央对称模式下的 PWM 半周期数；

13.6.14 TIM8 捕获/比较寄存器 1（TIM8_CCR1）

偏移地址：0x34

复位值：0x0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												CCR1 [19:16]			
												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:20 保留，必须保持为复位值。

Bits 19:0 **CCR1[19:0]**：捕获/比较 1 值（Capture/Compare 1 value）

若 CC1 通道配置为输出：

CCR1 为要装载到实际捕获/比较 1 寄存器的值（预装载值）。

如果没有通过 TIM8_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较1寄存器中。实际捕获/比较寄存器中包含要与计数器 TIM8_CNT 进行比较并在 tim_oc1 输出上发出信号的值。

在非抖动模式下（DITHEN = 0）

该寄存器存储的是 CCR1[15:0]中的比较值。CCR1[19:16]位会被重置。

在抖动模式下（DITHEN = 1）

该寄存器存储的是 CCR1[19:4]中的整数部分。CCR1[3:0]位域存储的是抖动的部分。

13.6.15 TIM8 捕获/比较寄存器 2（TIM8_CCR2）

偏移地址：0x38

复位值：0x0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												CCR2 [19:16]			
												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:20 保留，必须保持为复位值。

Bits 19:0 **CCR2[19:0]：**捕获/比较 2 值（Capture/Compare 2 value）

若 CC2 通道配置为输出：

CCR2 为要装载到实际捕获/比较 1 寄存器的值（预装载值）。

如果没有通过 TIM8_CCMR1 寄存器中的 OC2PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 2 寄存器中。实际捕获/比较寄存器中包含要与计数器 TIM8_CNT 进行比较并在 tim_oc2 输出上发出信号的值。

在非抖动模式下（DITHEN = 0）

该寄存器存储的是 CCR2[15:0]中的比较值。CCR2[19:16]位会被重置。

在抖动模式下（DITHEN = 1）

该寄存器存储的是 CCR2[19:4]中的整数部分。CCR2[3:0]位域存储的是抖动的部分。

13.6.16 TIM8 捕获/比较寄存器 3（TIM8_CCR3）

偏移地址：0x3C

复位值：0x0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												CCR3 [19:16]			
												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:20 保留，必须保持为复位值。

Bits 19:0 **CCR3[19:0]：**捕获/比较 3 值（Capture/Compare 3 value）

若 CC3 通道配置为输出：

CCR3 为要装载到实际捕获/比较 3 寄存器的值（预装载值）。

如果没有通过 TIM8_CCMR2 寄存器中的 OC3PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 3 寄存器中。实际捕获/比较寄存器中包含要与计数器 TIM8_CNT 进行比较并在 tim_oc3 输出上发出信号的值。

在非抖动模式下（DITHEN = 0）

该寄存器存储的是 CCR3[15:0]中的比较值。CCR3[19:16]位会被重置。

在抖动模式下（DITHEN = 1）

该寄存器存储的是 CCR3[19:4]中的整数部分。CCR3[3:0]位域存储的是抖动的部分。

13.6.17 TIM8 捕获/比较寄存器 4（TIM8_CCR4）

偏移地址：0x40

复位值：0x0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												CCR4 [19:16]			
												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:20 保留，必须保持为复位值。

Bits 19:0 **CCR4[19:0]：**捕获/比较 4 值（Capture/Compare 4 value）

若 CC4 通道配置为输出：

CCR4 为要装载到实际捕获/比较 4 寄存器的值（预装载值）。

如果没有通过 TIM8_CCMR2 寄存器中的 OC4PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 4 寄存器中。实际捕获/比较寄存器中包含要与计数器 TIM8_CNT 进行比较并在 tim_oc4 输出上发出信号的值。

在非抖动模式下（DITHEN = 0）

该寄存器存储的是 CCR4[15:0]中的比较值。CCR4[19:16]位会被重置。

在抖动模式下（DITHEN = 1）

该寄存器存储的是 CCR4[19:4]中的整数部分。CCR4[3:0]位域存储的是抖动的部分。

13.6.18 TIM8 刹车和死区寄存器（TIM8_BDTR）

偏移地址：0x44

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		BK2BID	BKBID	BK2 DSRM	BK DSRM	BK2P	BK2E	BK2F[3:0]				BKF[3:0]			
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

注：由于可以根据 LOCK *配置锁定位* BKBID/BK2BID/BK2P，BK2E，BK2F[3:0]，BKF[3:0]，AOE，BKP，BKE，OSSI，OSSR *和* DTG[7:0]*的写操作，因此必须在第一次对* TIM8_BDTR *寄存器执行写访问时对这些位进行配置。*

Bits 31:30 保留，必须保持为复位值。

Bit 29 **BK2BID：**刹车 2 双向（Break2 bidirectional）

请参阅 BKBID 说明

Bit 28 **BKBID：**刹车双向（Break bidirectional）

0：在输入模式下刹车输入 tim_brk

1：在双向模式下刹车输入 tim_brk

在双向模式（BKBID 位设置为 1）中，刹车输入在输入和输出中都配置模式和开漏输出模式。任何活动中断事件都会在刹车输入，向外部设备指示内部中断事件。

*注：只要编程了*LOCK（TIM8_BDTR *寄存器中的*LOCK *位）级别*1，*此位即无法修改。*

*注：对此位的任何写入操作都需要延迟*1 *个*APB *时钟周期才能生效。*

Bit 27 **BK2DSRM：**刹车 2 解除（Break2 disarm）

请参阅 BKDSRM 说明

Bit 26 **BKDSRM：**刹车解除（Break disarm）

0：刹车输入 tim_brk 已装载

1：刹车输入 tim_brk 被解除

当没有刹车源处于有效状态时，此位由硬件清除。

BKDSRM 位必须由软件设置，以释放双向输出控制（高阻状态下的开漏输出），然后对其进行轮询，直到它被硬件重置，表明故障状态已消失。

*注：对此位的任何写入操作都需要延迟*1 *个*APB *时钟周期才能生效。*

Bit 25 **BK2P：**刹车 2 极性（Break 2 polarity）

0：输入时间_brk2 为低电平有效

1：输入时间中断_brk2 为高电平有效

*注：只要编程了*LOCK（TIM8_BDTR *寄存器中的*LOCK *位）级别*1，*此位即无法修改。*

*注：对此位的任何写入操作都需要延迟*1 *个*APB *时钟周期才能生效。*

Bit 24 **BK2E：**刹车 2 使能（Break 2 enable）

此位使能完全刹车 2 保护（包括所有连接到 bk_acth 和 BKIN 的源，参考图片：刹车和刹车 2 电路概述）。

0：刹车 2 功能禁用

1：使能刹车 2 功能

*注：只要编程了*LOCK（TIM8_BDTR *寄存器中的*LOCK *位）级别*1，*此位即无法修改。*

*注：对此位的任何写入操作都需要延迟*1 *个*APB *时钟周期才能生效。*

Bits 23:20 **BK2F[3:0]：**刹车 2 滤波器

这几位定义了 tim_brk2 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到 N 个事件后会产生一个输出的跳变

0000：无滤波器，tim_brk2 异步动作。

0001：$f_{SAMPLING}=f_{tim_ker_ck}$，N=2。

0010：$f_{SAMPLING}=f_{tim_ker_ck}$，N=4。

0011：$f_{SAMPLING}=f_{tim_ker_ck}$，N=8。

0100：$f_{SAMPLING}=f_{DTS}/2$，N=6。

0101：$f_{SAMPLING}=f_{DTS}/2$，N=8。

0110：$f_{SAMPLING}=f_{DTS}/4$，N=6。

0111：$f_{SAMPLING}=f_{DTS}/4$，N=8。

1000：$f_{SAMPLING}=f_{DTS}/8$，N=6。

1001：$f_{SAMPLING}=f_{DTS}/8$，N=8。

1010：$f_{SAMPLING}=f_{DTS}/16$，N=5。

1011：$f_{SAMPLING}=f_{DTS}/16$，N=6。

1100：$f_{SAMPLING}=f_{DTS}/16$，N=8。

1101：$f_{SAMPLING}=f_{DTS}/32$，N=5。

1110：$f_{SAMPLING}=f_{DTS}/32$，N=6。

1111：$f_{SAMPLING}=f_{DTS}/32$，N=8。

*注：只要编程了*LOCK（TIM8_BDTR *寄存器中的*LOCK *位）级别*1，*此位即无法修改。*

Bits 19:16 **BKF[3:0]：**刹车滤波器

这几位定义了 tim_brk 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到 N 个事件后会产生一个输出的跳变

0000：无滤波器，tim_brk2 异步动作。

0001：$f_{SAMPLING}=f_{tim_ker_ck}$，N=2。

0010：$f_{SAMPLING}=f_{tim_ker_ck}$，N=4。

0011：$f_{SAMPLING}=f_{tim_ker_ck}$，N=8。

0100：$f_{SAMPLING}=f_{DTS}/2$，N=6。

0101：$f_{SAMPLING}=f_{DTS}/2$，N=8。

0110：$f_{SAMPLING}=f_{DTS}/4$，N=6。

0111：$f_{SAMPLING}=f_{DTS}/4$，N=8。

1000：$f_{SAMPLING}=f_{DTS}/8$，N=6。

1001：$f_{SAMPLING}=f_{DTS}/8$，N=8。

1010：$f_{SAMPLING}=f_{DTS}/16$，N=5。

1011：$f_{SAMPLING}=f_{DTS}/16$，N=6。

1100：$f_{SAMPLING}=f_{DTS}/16$，N=8。

1101：$f_{SAMPLING}=f_{DTS}/32$，N=5。

1110：$f_{SAMPLING}=f_{DTS}/32$，N=6。

1111：$f_{SAMPLING}=f_{DTS}/32$，N=8。

*注：只要编程了*LOCK（TIM8_BDTR *寄存器中的*LOCK *位）级别*1，*此位即无法修改。*

Bit 15 **MOE：**主输出使能（Main output enable）

只要刹车输入变为有效状态，此位便由硬件异步清零。此位由软件置 1，也可根据 AOE 位状态自动置 1。此位仅对配置为输出的通道有效。

0：OC 和 OCN 输出禁止或被强制为空闲状态。

1：如果 OC 和 OCN 输出的相应使能位（TIM8_CCER 寄存器中的 CCxE 和 CCxNE 位）均置 1，则使能 OC 和 OCN 输出。

有关详细信息, 请参见OC/OCN 使能说明(TIM8 捕获/比较使能寄存器(TIM8_CCER))。

Bit 14 **AOE：**自动输出使能（Automatic output enable）

0：MOE 只能由软件置 1

1：MOE 可由软件置 1，也可在发生下一更新事件时自动置 1（如果刹车输入无效）

*注：只要编程了*LOCK（TIM8_BDTR *寄存器中的*LOCK *位）级别*1，*此位即无法修改。*

Bit 13 **BKP：**刹车极性（Break polarity）

0：刹车输入低电平有效；

1：刹车输入高电平有效。

*注：只要编程了*LOCK（TIM8_BDTR *寄存器中的*LOCK *位）级别*1，*此位即无法修改。*

*注：对该位执行任何写操作后，都需要经过*1 *个*APB *时钟周期的延迟才生效。*

Bit 12 **BKE：**刹车使能（Break enable）

0：禁止刹车输入（BRK 及 CCS 时钟失效事件）；

1：开启刹车输入（BRK 及 CCS 时钟失效事件）。

*注：只要编程了*LOCK（TIM8_BDTR *寄存器中的*LOCK *位）级别*1，*此位即无法修改。*

注：对该位执行任何写操作后，都需要经过1 个APB 时钟周期的延迟才生效。

Bit 11 **OSSR：**运行模式下"关闭状态"选择（Off-state selection for Run mode）

此位在 MOE=1 时作用于配置为输出模式且具有互补输出的通道。如果定时器中没有互补输出，则不存在 OSSR。

有关详细信息，请参见OC/OCN 使能说明(TIM8 捕获/比较使能寄存器(TIM8_CCER))。

0：处于无效状态时，禁止 OC/OCN 输出（OC/OCN 使能输出信号=0）。

1：处于无效状态时，一旦 CCxE=1 或 CCxNE=1，便使能 OC/OCN 输出并将其设为无效电平。然后设置 OC/OCN 使能输出信号=1。

*注：编程了*LOCK *（*TIM8_BDTR *寄存器中的*LOCK *位）级别*2 *后，此位即无法修改。*

Bit 10 **OSSI：**空闲模式下的关闭状态选择（Off-state selection for Idle mode）

此位在 MOE=0 时作用于配置为输出的通道。

有关详细信息，请参见OC/OCN 使能说明(TIM8 捕获/比较使能寄存器(TIM8_CCER))。

0：处于无效状态时，禁止 OC/OCN 输出（OC/OCN 使能输出信号=0）。

1：处于无效状态时，一旦 CCxE=1 或 CCxNE=1，便将 OC/OCN 输出首先强制为其空闲电平。然后设置 OC/OCN 使能输出信号=1。

*注：编程了*LOCK *（*TIM8_BDTR *寄存器中的*LOCK *位）级别*2 *后，此位即无法修改。*

Bits 9:8 **LOCK[1:0]：**锁定设置（Lock configuration）

这些位用于针对软件错误提供写保护。

00：关闭锁定--不对任何位提供写保护。

01：锁定级别 1，此时无法对 TIM8_BDTR 寄存器中的 DTG 位、TIM8_CR2 寄存器中的 OISx 和 OISxN 位以及 TIM8_BDTR 寄存器中的 BKE/BKP/AOE 位执行写操作。

10：锁定级别 2，此时无法对锁定级别 1 中适用的各位、CC 极性位（TIM8_CCER 寄存器中的 CCxP/CCxNP 位，只要通过 CCxS 位将相关通道配置为输出）以及 OSSR 和 OSSI 位执行写操作。

11：锁定级别 3，此时无法对锁定级别 2 中适用的各位、CC 控制位（TIM8_CCMRx 寄存器中的 OCxM 和 OCxPE 位，只要通过 CCxS 位将相关通道配置为输出）执行写操作。

*注：复位后只能对*LOCK *位执行一次写操作。对*TIM8_BDTR *寄存器执行写操作后其中的内容将冻结，直到下一次复位。*

Bits 7:0 **DTG[7:0]：**配置死区发生器（Dead-time generator setup）

此位域定义插入到互补输出之间的死区持续时间。DT 与该持续时间相对应。

DTG[7:5]=0xx => DT= DTG[7:0] × T_{dtg}，其中 T_{dtg} = T_{DTS}；

DTG[7:5]=10x => DT=（64+DTG[5:0]）× T_{dtg}，其中 T_{dtg} = 2 × T_{DTS}；

DTG[7:5]=110 => DT=（32+DTG[4:0]）× T_{dtg}，其中 T_{dtg} = 8 × T_{DTS}；

DTG[7:5]=111 => DT=（32+DTG[4:0]）× T_{dtg}，其中 T_{dtg} = 16 × T_{DTS}；

例：若 T_{DTS} = 125ns（8MHz），可能的死区时间为：

0 到 15875 ns（步长为 125 ns），

16 us 到 31750 ns（步长为 250 ns），

32 us 到 63us（步长为 1 us），

64 us 到 126 us（步长为 2 us）

*注：编程了*LOCK *（*TIM8_BDTR *寄存器中的*LOCK *位）级别*1、2 *或*3，*此位即无法修改。*

13.6.19 TIM8 捕获/比较寄存器 5（TIM8_CCR5）

偏移地址：0x48

复位值：0x0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GC5C3	GC5C2	GC5C1	Reserved									CCR5 [19:16]			
rw	rw	rw										rw	rw	rw	rw

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

Bit 31 **GC5C3：**通道 5 和通道 3 组（Group channel 5 and channel 3）

通道 3 输出失真：

0：tim_oc5ref 对 tim_oc3refc 无影响

1：tim_oc3refc 是 tim_oc3ref 和 tim_oc5ref 的逻辑 AND 操作

此位要么立即生效，要么被预加载并在更新事件后被考虑（如果 TIM8_CCMR2 中选择了预加载功能）。

注：也可以在组合 PWM *信号上应用这种失真。*

Bit 30 **GC5C2：**通道 5 和通道 2 组（Group channel 5 and channel 2）

通道 2 输出失真：

0：tim_oc5ref 对 tim_oc2refc 无影响

1：tim_oc2refc 是 tim_oc2ref 和 tim_oc5ref 的逻辑 AND 操作

此位要么立即生效，要么被预加载并在更新事件后被考虑（如果 TIM8_CCMR1 中选择了预加载功能）。

注：也可以在组合 PWM *信号上应用这种失真。*

Bit 29 **GC5C1：**通道 5 和通道 1 组（Group channel 5 and channel 1）

通道 1 输出失真：

0：tim_oc5ref 对 tim_oc1refc 无影响

1：tim_oc1refc 是 tim_oc1ref 和 tim_oc5ref 的逻辑 AND 操作

此位要么立即生效，要么被预加载并在更新事件后被考虑（如果 TIM8_CCMR1 中选择了预加载功能）。

注：也可以在组合 PWM *信号上应用这种失真。*

Bits 28:20 保留，必须保持为复位值。

Bits 19:0 **CCR5[15:0]：**捕获/比较 5 值（Capture/Compare 5 value）

若 CC5 通道配置为输出：

CCR5 为要装载到实际捕获/比较 5 寄存器的值（预装载值）。

如果没有通过 TIM8_CCMR3 寄存器中的 OC5PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 5 寄存器中。实际捕获/比较寄存器中包含要与计数器 TIM8_CNT 进行比较并在 tim_oc5 输出上发出信号的值。

在非抖动模式下（DITHEN = 0）

该寄存器存储的是 CCR5[15:0]中的比较值。CCR5[19:16]位会被重置。

在抖动模式下（DITHEN = 1）

该寄存器存储的是 CCR5[19:4]中的整数部分。CCR5[3:0]位域存储的是抖动的部分。

13.6.20 TIM8 捕获/比较寄存器 6（TIM8_CCR6）

偏移地址：0x4C
复位值：0x0000
访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												CCR6 [19:16]			
												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR6 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:20 保留，必须保持为复位值。

Bits 19:0 **CCR6[15:0]：**捕获/比较 6 值（Capture/Compare 6 value）

若 CC6 通道配置为输出：

CCR6 为要装载到实际捕获/比较 6 寄存器的值（预装载值）。

如果没有通过 TIM8_CCMR3 寄存器中的 OC6PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 6 寄存器中。实际捕获/比较寄存器中包含要与计数器 TIM8_CNT 进行比较并在 tim_oc6 输出上发出信号的值。

在非抖动模式下（DITHEN = 0）

该寄存器存储的是 CCR6[15:0]中的比较值。CCR6[19:16]位会被重置。

在抖动模式下（DITHEN = 1）

该寄存器存储的是 CCR6[19:4]中的整数部分。CCR6[3:0]位域存储的是抖动的部分。

13.6.21 TIM8 捕获/比较模式寄存器 3（TIM8_CCMR3）

偏移地址：0x50
复位值：0x0000 0000
访问：无等待，支持字，半字和字节访问
参看以上 CCMR1 寄存器的描述，通道 5 和 6 只能配置为输出。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							OC6M[3]	Reserved							OC5M[3]
							rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0C6CE	0C6M[2:0]			0C6PE	0C6FE	Reserved		OC5CE	OC5M[2:0]			OC5PE	OC5FE	Reserved	
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw		

Bits 31:25 保留，必须保持为复位值。

Bit 24 OC6M[3]：输出比较 6 模式（Output Compare 6 mode）

Bits 23:17 保留，必须保持为复位值。

Bits 16 **OC5M[3]：**输出比较 5 模式（Output Compare5 mode）

Bit 15 **OC6CE：**输出比较 6 清零使能（Output Compare 6 clear enable）

Bits 24,14:12 **OC6M[3:0]：**输出比较 6 模式（Output Compare 6 mode）

参考 OC1M 描述（TIMx_CCMR1 寄存器）

Bit 11 **OC6PE：**输出比较 6 预装载使能（Output Compare 6 preload enable）

Bit 10 **OC6FE：**输出比较 6 快速使能（Output Compare 6 fast enable）

Bits 9:8 保留，必须保持为复位值。

Bit 7 **OC5CE：**输出比较 5 清零使能（Output Compare 5 clear enable）

Bits 16, 6:4 **OC5M[3:0]：**输出比较 5 模式（Output Compare5 mode）

参考 OC1M 描述（TIMx_CCMR1 寄存器）

Bit 3 **OC5PE：**输出比较 5 预装载使能（Output Compare 5 preload enable）

Bit 2 **OC5FE：**输出比较 5 快速使能（Output Compare 5 fast enable）

Bit 1:0 保留，必须保持为复位值。

13.6.22 TIM8 定时器死区寄存器 2（TIM8_DTR2）

偏移地址：0x54

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

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

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

Bits 31:18 保留，必须保持为复位值。

Bit 17 **DTPE：**死区时间预加载使能（Deadtime preload enable）

0：死区时间值未预加载

1：使能死区时间值预加载

*注：编程了*LOCK（TIM8_BDTR *寄存器中的*LOCK *位）级别*1、2 *或*3，*此位即无法修改。*

Bit 16 **DTAE：**死区时间非对称使能（Deadtime asymmetric enable）

0：上升沿和下降沿的死区时间是相同的，并且使用 DTG[7:0]寄存器定义

1：上升沿的死区时间使用 DTG[7:0]寄存器定义，下降沿的死区时间使用 DTGF[7:0]位定义。

*注：编程了*LOCK（TIM8_BDTR *寄存器中的*LOCK *位）级别*1、2 *或*3，*此位即无法修改。*

Bits 15:8 保留，必须保持为复位值。

Bits 7:0 **DTGF[7:0]：**死区下降沿发生器（Dead-time falling edge generator setup）

这个位域定义了在下降沿时，互补输出之间插入死区时间的长度。

DTG[7:5]=0xx => DT= DTG[7:0] × T_{dtg}，其中 T_{dtg} = T_{DTS}；

DTG[7:5]=10x => DT=（64+DTG[5:0]）× T_{dtg}，其中 T_{dtg} = 2 × T_{DTS}；

DTG[7:5]=110 => DT=（32+DTG[4:0]）× T_{dtg}，其中 T_{dtg} = 8 × T_{DTS}；

DTG[7:5]=111 => DT=（32+DTG[4:0]）× T_{dtg}，其中 T_{dtg} = 16 × T_{DTS}；

例：若 T_{DTS} = 125ns（8MHz），可能的死区时间为：

0 到 15875 ns（步长为 125 ns），

16 us 到 31750 ns（步长为 250 ns），

32 us 到 63us（步长为 1 us），

64 us 到 126 us（步长为 2 us）

*注：编程了*LOCK（TIM8_BDTR *寄存器中的*LOCK *位）级别*1、2 *或*3，*此位即无法修改。*

13.6.23 TIM8 定时器编码器控制寄存器（TIM8_ECR）

偏移地址：0x58

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					PWPRSC[2:0]			PW[7:0]							
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

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

Bits 31:27 保留，必须保持为复位值。

Bits 26:24 **PWPRSC[2:0]：**脉冲宽度预分频器（Pulse width prescaler）

这个位域为脉冲发生器设置时钟预分频器，具体如下：

$t_{PWG} = (2^{(PWPRSC[2:0])}) \times t_{tim_ker_ck}$

Bits 23:16 **PW[7:0]：**脉冲宽度（Pulse width）

这个位域定义脉冲的持续时间，具体如下：

$t_{PW} = PW[7:0] \times t_{PWG}$

Bit 15:0 保留，必须保持为复位值。

13.6.24 TIM8 复用功能选择寄存器 1（TIM8_AF1）

偏移地址：0x60

复位值：0x0000 0001

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														ETRSEL[3:2]	
														rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETRSEL[1:0]		Reserved		BK CMP2P	BK CMP1P	BKINP	Reserved						BK CMP2E	BK CMP1E	BKINE
rw	rw			rw	rw	rw							rw	rw	rw

Bits 31:18 保留，必须保持为复位值。

Bits 17:14 **ETRSEL[3:0]：**etr_in 源选择（etr_in source selection）

这些位选择 etr_in 输入源。

0000：保留

0001：tim_etr1

0010：tim_etr2

0011：保留

请参考章节：TIM8 引脚和内部信号，了解特定于产品的实现。

*注：只要编程了*LOCK *（*TIM8_BDTR *寄存器中的*LOCK *位）级别*1*，此位即无法修改。*

Bits 13:12 保留，必须保持为复位值。

Bit 11 **BKCMP2P：**tim_brk_cmp2 输入极性（tim_brk_cmp2 input polarity）

此位选择 tim_brk_cmp2 输入敏感度。它必须与 BKP 极性位一起编程。

0：tim_brk_cmp2 输入极性未反转（如果 BKP=0，则低电平有效，如果 BKP=1，则高电平有效）

1：tim_brk_cmp2 输入极性反转（如果 BKP=0，则高电平有效，如果 BKP=1，则低电

平有效）

*注：只要编程了*LOCK（TIM8_BDTR *寄存器中的*LOCK *位）级别*1，*此位即无法修改。*

Bit 10 **BKCMP1P**：tim_brk_cmp1 输入极性（tim_brk_cmp1 input polarity）

此位选择 tim_brk_cmp1 输入敏感度。它必须与 BKP 极性位一起编程。

0：tim_brk_cmp1 输入极性未反转（如果 BKP=0，则低电平有效，如果 BKP=1，则高电平有效）

1：tim_brk_cmp1 输入极性反转（如果 BKP=0，则高电平有效，如果 BKP=1，则低电平有效）

*注：只要编程了*LOCK（TIM8_BDTR *寄存器中的*LOCK *位）级别*1，*此位即无法修改。*

Bit 9 **BKINP**：TIM8_BKIN 输入极性（TIM8_BKIN input polarity）

该位选择 TIM8_BKIN 备用功能输入敏感度。必须对其进行编程

0：TIM8_BKIN 输入极性未反转（如果 BKP=0，则低电平有效，如果 BKP=1，则高电平有效）

1：TIM8_BKIN 输入极性反转（如果 BKP=0，则高电平有效，如果 BKP=1，则低电平有效）

*注：只要编程了*LOCK（TIM8_BDTR *寄存器中的*LOCK *位）级别*1，*此位即无法修改。*

Bits 8:3 保留，必须保持为复位值。

Bit 2 **BKCMP2E**：tim_brk_cmp2 使能（tim_brk_cmp2 enable）

该位用于使能或禁用 tim_brk_cmp2 输入。如果该位设置为 1，则 tim_brk_cmp2 输入将被使能，否则将被禁用。

0：tim_brk_cmp2 输入禁用

1：tim_brk_cmp2 输入使能

*注：只要编程了*LOCK（TIM8_BDTR *寄存器中的*LOCK *位）级别*1，*此位即无法修改。*

Bit 1 **BKCMP1E**：tim_brk_cmp1 使能（tim_brk_cmp1 enable）

该位用于使能或禁用 tim_brk_cmp1 输入。如果该位设置为 1，则 tim_brk_cmp1 输入将被使能，否则将被禁用。

0：tim_brk_cmp1 输入禁用

1：tim_brk_cmp1 输入使能

*注：只要编程了*LOCK（TIM8_BDTR *寄存器中的*LOCK *位）级别*1，*此位即无法修改。*

Bit 0 **BKINE**：TIM8_BKIN 输入使能（TIM8_BKIN input enable）

该位用于使能 TIM8_BKIN 复用功能输入作为定时器的 tim_brk 输入。TIM8_BKIN 输入与其他 tim_brk 源进行“或”运算。

0：TIM8_BKIN 输入禁用

1：TIM8_BKIN 输入使能

*注：只要编程了*LOCK（TIM8_BDTR *寄存器中的*LOCK *位）级别*1，*此位即无法修改。*

*注意：有关产品特定实现的信息，请参考章节：*TIM8 *引脚和内部信号。*

13.6.25 TIM8 复用功能选择寄存器 2（TIM8_AF2）

偏移地址：0x64

复位值：0x0000 0001

访问：无等待，支持字，半字和字节访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				BK2 CMP2P	BK2 CMP1P	BK2INP	Reserved						BK2 CMP2E	BK2 CMP1E	BK2INE
				rw	rw	rw							rw	rw	rw

Bits 31:19 保留，必须保持为复位值。

Bits 18:16 **OCRSEL[2:0]：**ocref_clr 源选择（ocref_clr source selection）

这些位选择 ocref_clr 输入源。

000：tim_ocref_clr0

001：tim_ocref_clr1

参考章节：TIM8 引脚和内部信号获取更多特定产品信息。

*注：只要编程了*LOCK（TIM8_BDTR *寄存器中的*LOCK *位）级别*1，*此位即无法修改。*

Bits 15:12 保留，必须保持为复位值。

Bit 11 **BK2CMP2P：**tim_brk2_cmp2 输入极性（tim_brk2_cmp2 input polarity）

此位选择 tim_brk2_cmp2 输入敏感度。它必须与 BK2P 极性位一起编程。

0：tim_brk2_cmp2 输入极性未反转（如果 BK2P=0，则低电平有效，如果 BK2P=1，则高电平有效）

1：tim_brk2_cmp2 输入极性反转（如果 BK2P=0，则高电平有效，如果 BK2P=1，则低电平有效）

*注：只要编程了*LOCK（TIM8_BDTR *寄存器中的*LOCK *位）级别*1，*此位即无法修改。*

Bit 10 **BK2CMP1P：**tim_brk2_cmp1 输入极性（tim_brk2_cmp1 input polarity）

此位选择 tim_brk2_cmp1 输入敏感度。它必须与 BK2P 极性位一起编程。

0：tim_brk2_cmp1 输入极性未反转（如果 BK2P=0，则低电平有效，如果 BK2P=1，则高电平有效）

1：tim_brk2_cmp1 输入极性反转（如果 BK2P=0，则高电平有效，如果 BK2P=1，则低电平有效）

*注：只要编程了*LOCK（TIM8_BDTR *寄存器中的*LOCK *位）级别*1，*此位即无法修改。*

Bit 9 **BK2INP：**TIM8_BKIN2 输入极性（TIM8_BKIN2 input polarity）

该位选择 TIM8_BKIN2 复用功能输入敏感度。必须对其进行编程

0：TIM8_BKIN2 输入极性未反转（如果 BK2P=0，则低电平有效，如果 BK2P=1，则高电平有效）

1：TIM8_BKIN2 输入极性反转（如果 BK2P=0，则高电平有效，如果 BK2P=1，则低电平有效）

*注：只要编程了*LOCK（TIM8_BDTR *寄存器中的*LOCK *位）级别*1，*此位即无法修改。*

Bits 8:3 保留，必须保持为复位值。

Bit 2 **BK2CMP2E：**tim_brk2_cmp2 使能（tim_brk2_cmp2 enable）

该位用于使能或禁用 tim_brk2_cmp2 输入。如果该位设置为 1，则 tim_brk2_cmp2 输

入将被使能，否则将被禁用。

0：tim_brk2_cmp2 输入禁用

1：tim_brk2_cmp2 输入使能

*注：只要编程了*LOCK（TIM8_BDTR *寄存器中的*LOCK *位）级别*1，*此位即无法修改。*

Bit 1 **BK2CMP1E：**tim_brk2_cmp1 使能（tim_brk2_cmp1 enable）

该位用于使能或禁用 tim_brk2_cmp1 输入。如果该位设置为 1，则 tim_brk2_cmp1 输入将被使能，否则将被禁用。

0：tim_brk2_cmp1 输入禁用

1：tim_brk2_cmp1 输入使能

*注：只要编程了*LOCK（TIM8_BDTR *寄存器中的*LOCK *位）级别*1，*此位即无法修改。*

Bit 0 **BK2INE：**TIM8_BKIN2 输入使能（TIM8_BKIN2 input enable）

该位用于使能或禁用 TIM8_BKIN2 复用功能输入作为定时器的 tim_brk2 输入。

TIM8_BKIN2 输入与其他 tim_brk2 源进行“OR”操作。

0：TIM8_BKIN2 输入禁用

1：TIM8_BKIN2 输入使能

*注：只要编程了*LOCK（TIM8_BDTR *寄存器中的*LOCK *位）级别*1，*此位即无法修改。*

*注意：有关产品特定实现的信息，请参考章节：*TIM8 *引脚和内部信号。*

13.6.26 TIM8 捕获/比较寄存器 1N（TIM8_CCR1N）

偏移地址：0x68

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1N[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit 31 **EN：**若 CC1 通道配置为输出：

CCR1N_EN=1：CNT 向下数比较的寄存器为 CCR1N

CCR1N_EN=0：CNT 向下数比较的寄存器为 CCR1

Bits 30:16 保留，必须保持为复位值。

Bits 15:0 **CCR1N[15:0]：**捕获/比较通道 1N 的值（Capture/Compare 1N value）

若 CC1 通道配置为输出：

CCR1N 包含了装入当前捕获/比较 1 寄存器的值（预装载值）。

如果在 TIM8_CCMR1 寄存器（OC1PE 位）中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 1 寄存器中。

CCR1N_EN 必须为 1，寄存器才有效果。

13.6.27 TIM8 捕获/比较寄存器 2N（TIM8_CCR2N）

偏移地址：0x6C

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2N[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit 31 **EN**：若 CC2 通道配置为输出：

CCR2N_EN=1：CNT 向下数比较的寄存器为 CCR2N

CCR2N_EN=0：CNT 向下数比较的寄存器为 CCR2

Bits 30:16 保留，必须保持为复位值。

Bits 15:0 CCR2N[15:0]：捕获/比较通道 2N 的值（Capture/Compare 2N value）

若 CC2 通道配置为输出：

CCR2N 包含了装入当前捕获/比较 2 寄存器的值（预装载值）。

如果在 TIM8_CCMR1 寄存器（OC2PE 位）中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 2 寄存器中。

CCR2N_EN 必须为 1，寄存器才有效果。

13.6.28 TIM8 捕获/比较寄存器 3N（TIM8_CCR3N）

偏移地址：0x70

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3N[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit 31 **EN**：若 CC3 通道配置为输出：

CCR3N_EN=1：CNT 向下数比较的寄存器为 CCR3N

CCR3N_EN=0：CNT 向下数比较的寄存器为 CCR3

Bits 30:16 保留，必须保持为复位值。

Bits 15:0 **CCR3N[15:0]**：捕获/比较通道 3N 的值（Capture/Compare 3N value）

若 CC3 通道配置为输出：

CCR3N 包含了装入当前捕获/比较 3 寄存器的值（预装载值）。

如果在 TIM8_CCMR2 寄存器（OC3PE 位）中未选择预装载功能，写入的数值会立即传输至当前寄存器中。否则只有当更新事件发生时，此预装载值才传输至当前捕获/比较 3 寄存器中。

CCR3N_EN 必须为 1，寄存器才有效果。

13.6.29 TIM8 刹车输入控制选择（TIM8_BICS）

偏移地址：0x78

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									CH6	CH5	CH4	CH3	CH2	CH1	BICE
									rw	rw	rw	rw	rw	rw	rw

Bits 15:7 保留，必须保持为复位值。

Bit 6 **CH6：**刹车输入控制选择 CH6（Break lutput Control Selection CH6）

当 BICE=1 时：

0：CH6 刹车有效或设置 MOE=0 有效，通道根据极性配置输出对应电平

1：CH6 刹车无效，设置 MOE=0 无效，通道维持原有输出

Bit 5 **CH5：**刹车输入控制选择 CH5（Break lutput Control Selection CH5）

当 BICE=1 时：

0：CH5 刹车有效或设置 MOE=0 有效，通道根据极性配置输出对应电平

1：CH5 刹车无效，设置 MOE=0 无效，通道维持原有输出

Bit 4 **CH4：**刹车输入控制选择 CH4（Break lutput Control Selection CH4）

当 BICE=1 时：

0：CH4 刹车有效或设置 MOE=0 有效，通道根据极性配置输出对应电平

1：CH4 刹车无效，设置 MOE=0 无效，通道维持原有输出

Bit 3 **CH3：**刹车输入控制选择 CH3（Break lutput Control Selection CH3）

当 BICE=1 时：

0：CH3 刹车有效或设置 MOE=0 有效，通道根据极性配置输出对应电平

1：CH3 刹车无效，设置 MOE=0 无效，通道维持原有输出

Bit 2 **CH2：**刹车输入控制选择 CH2（Break lutput Control Selection CH2）

当 BICE=1 时：

0：CH2 刹车有效或设置 MOE=0 有效，通道根据极性配置输出对应电平

1：CH2 刹车无效，设置 MOE=0 无效，通道维持原有输出

Bit 1 **CH1：**刹车输入控制选择 CH1（Break lutput Control Selection CH1）

当 BICE=1 时：

0：CH1 刹车有效或设置 MOE=0 有效，通道根据极性配置输出对应电平

1：CH1 刹车无效，设置 MOE=0 无效，通道维持原有输出

Bit 0 **BICE：**刹车输入控制使能（Break lutput Control Enable）

当刹车输入或设置 MOE=0 时，此位控制刹车或设置 MOE=0 是否对 CHx/CHyN(x=1,2,3,4,5,6)(y=1,2,3,4)有效：

BICE=0：刹车有效或设置 MOE=0 有效，通道根据极性配置输出对应电平

BICE=1：根据 BICS_CHx 位可选某些通道忽略刹车输入或忽略设置 MOE=0 操作维持原有输出

13.6.30 TIM8 触发寄存器（TIM8_TRGI）

偏移地址：0x80

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

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

Bits 15:1 保留，必须保持为复位值。

Bit 0 **SLAVE_SYNC**：同步模式，从机同步配置。

当从模式选择为触发模式时：

0：从机同步模式禁止

1：从机同步模式开启

14 通用定时器（TIM2/TIM3）

14.1 TIM2/TIM3 简介

通用定时器包含一个 16 位或 28 位自动重载计数器，该计数器由可编程预分频器驱动。它们可用于多种用途，包括测量输入信号的脉冲宽度（输入捕获）或生成输出波形（输出比较和 PWM）。使用定时器预分频器和 RCC 时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。这些定时器彼此完全独立，不共享任何资源，它们可以同步操作。

14.2 TIM2/TIM3 主要功能

通用 TIMx（TIM2、TIM3）定时器功能包括：

- 16 位（TIM3）或 28 位（TIM2）向上、向下、向上/向下自动装载计数器
- 16 位可编程（可以实时修改）预分频器，计数器时钟频率的分频系数为 1～65536 之间的任意数值
- 4 个独立通道：
  - 输入捕获
  - 输出比较
  - PWM 生成（边沿或中间对齐模式）
  - 单脉冲模式输出
- 使用外部信号控制定时器和定时器互连的同步电路
- 如下事件发生时产生中断：
  - 更新：计数器向上溢出/向下溢出，计数器初始化（通过软件或者内部/外部触发）
  - 触发事件（计数器启动、停止、初始化或者由内部/外部触发计数）
  - 输入捕获
  - 输出比较
- 支持针对定位的增量（正交）编码器和霍尔传感器电路
- 触发输入作为外部时钟或者按周期的电流管理

14.3 TIM2/TIM3 实现

表 14.1 RX32S632 系列通用定时器

定时器实例	TIM2	TIM3
分辨率	28 位	16 位
OCREF 清除选择 源	是 tim_etrf tim_ocref_clr[7:0]	是 tim_etrf tim_ocref_clr[7:0]

14.4 TIM2/TIM3 功能描述

14.4.1 框图

图 14.1 通用定时器 TIM2 框图

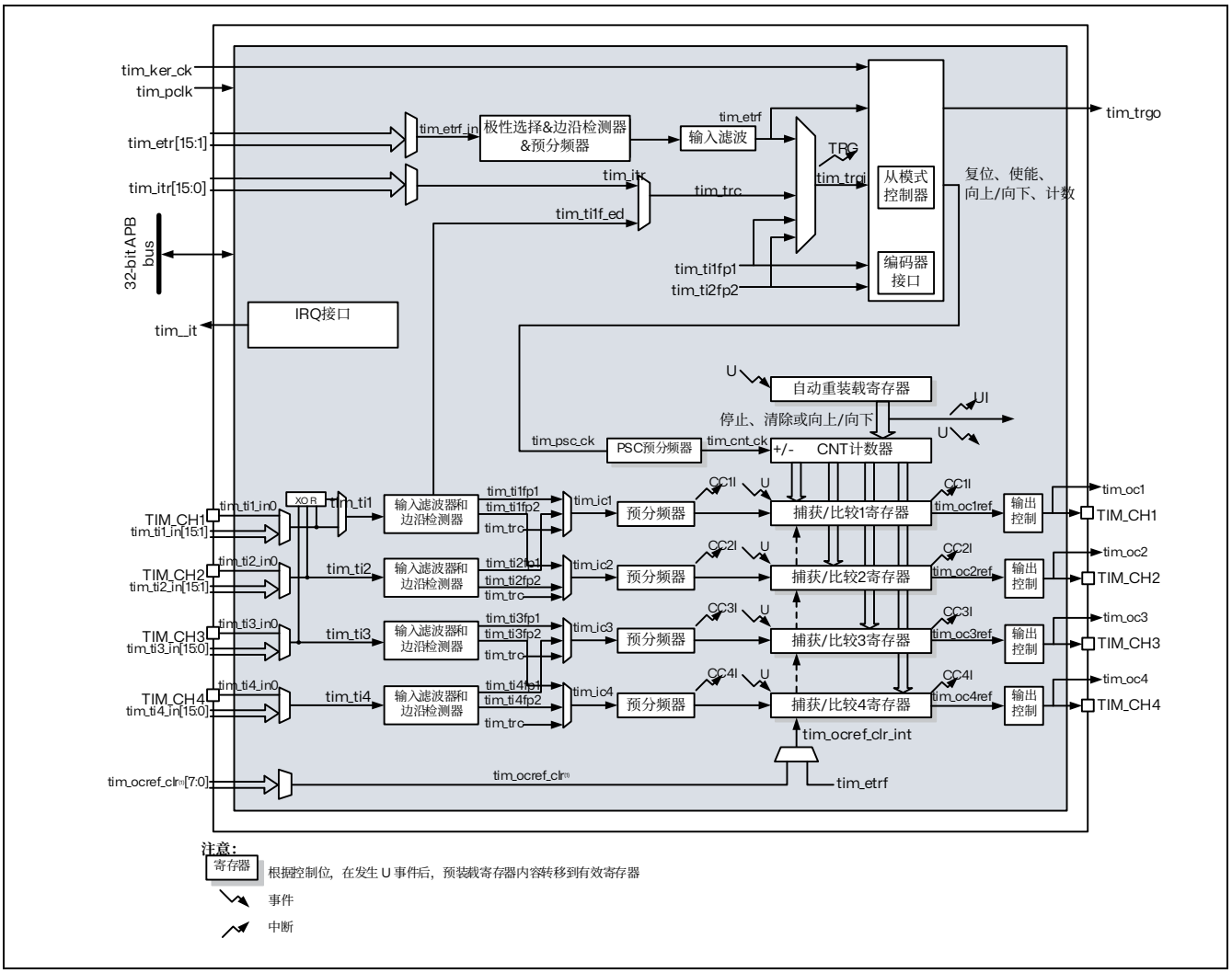
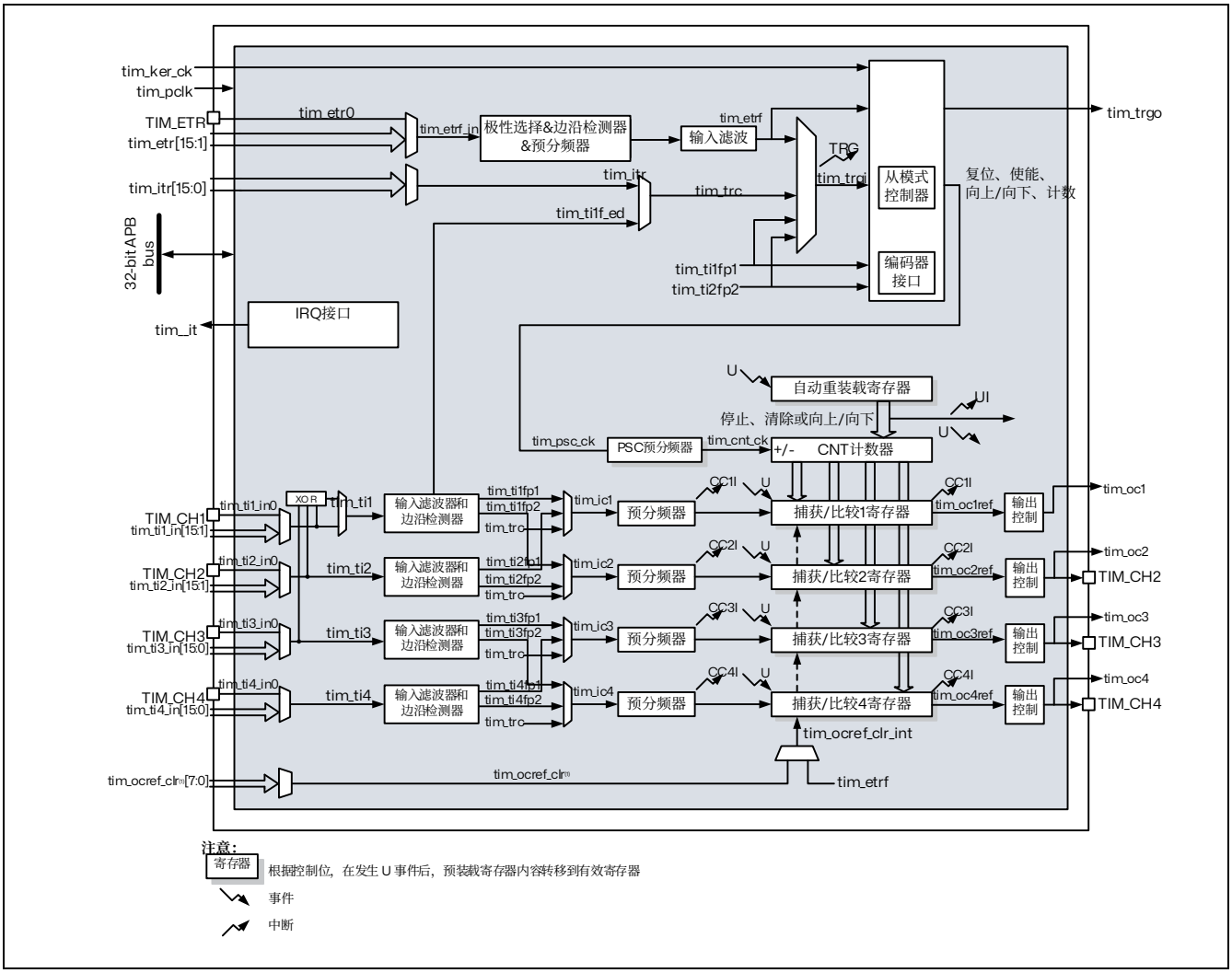


图 14.2 通用定时器 TIM3 框图



注 1. 此功能具体请参考章节：TIM2/TIM3 实现

14.4.2 TIM2/TIM3 引脚和内部信号

本节中的下两张表格总结了 TIM 的输入和输出。

表 14.2 TIM 输入/输出引脚

引脚名称	信号类型	描述
TIM_CH1 TIM_CH2 TIM_CH3 TIM_CH4	输入/输出	定时器多功能通道 每个通道用于捕获、比较或 PWM TIM_CH1 和 TIM_CH2 也可用作外部时钟（tim_ker_ck 时钟的 1/4 以下），外部触发器和正交编码器输入 TIM_CH1，TIM_CH2 和 TIM_CH3 可被用作数字霍尔效应传感器接口
TIM_ETR	输入	外部触发器输入。这个输入可被用作外部触发器或外部时钟源。如果使用 tim_etr_in 预分频器，则该输入可以接收频率高于 tim_ker_ck 的时钟。

表 14.3 TIM 内部输入/输出信号

内部信号名称	信号类型	描述
tim_ti1_in[15:0] tim_ti2_in[15:0] tim_ti3_in[15:0] tim_ti4_in[15:0]	输入	内部定时器输入总线 tim_ti1_in[15:0]和 tim_ti2_in[15:0]输入可被用作捕获或作为外部时钟(tim_ker_ck 时钟的 1/4 以下)和正交编码器信号
tim_etr[15:0]	输入	外部触发器内部输入总线。这些输入可被用作触发器、外部时钟或用于硬件的逐周期脉冲宽度控制。如果使用 tim_etr_in 预分频器，这些输入可以接收频率高于 tim_ker_ck 的时钟
tim_itr[15:0]	输入	内部触发输入总线。这些输入可用于从模式控制器或作为输入时钟(tim_ker_ck 时钟的 1/4 以下)
tim_trgo	输出	内部触发器输出。此触发器可以触发其他片上外设
tim_ocref_clr[7:0]	输入	定时器 tim_ocref_clr 输入总线。这些输入可以用来清除 tim_ocxref 信号，通常用于硬件的逐周期脉冲宽度控制
tim_pclk	输入	定时器 APB 时钟
tim_ker_ck	输入	定时器内部时钟
tim_it	输出	全局定时器中断，收集捕获/比较、更新和中断触发器请求

如下四张表列出了连接到 tim_ti[4:1]输入多路复用器的源。

表 14.4 连接到 tim_ti1 的输入多路复用器

tim_ti1 输入	源	
	TIM2	TIM3
tim_ti1_in0	TIM2_CH1	TIM3_CH1
tim_ti1_in1	cmp1_out	cmp1_out
tim_ti1_in2	cmp2_out	cmp2_out
tim_ti1_in3	LSI	LSI

表 14.5 连接到 tim_ti2 的输入多路复用器

tim_ti2 输入	源	
	TIM2	TIM3
tim_ti2_in0	TIM2_CH2	TIM3_CH2
tim_ti2_in1	cmp1_out	cmp1_out
tim_ti2_in2	cmp2_out	cmp2_out

表 14.6 连接到 tim_ti3 的输入多路复用器

tim_ti3 输入	源	
	TIM2	TIM3
tim_ti3_in0	TIM2_CH3	TIM3_CH3

表 14.7 连接到 tim_ti4 的输入多路复用器

tim_ti4 输入	源	
	TIM2	TIM3
tim_ti4_in0	TIM2_CH4	TIM3_CH4
tim_ti4_in1	cmp1_out	保留
tim_ti4_in2	cmp2_out	

下表列出连接到 tim_itr 输入多路复用器的内部源

表 14.8 TIMx 内部触发连接

TIMx	TIM2	TIM3
tim_itr0	tim3_trgo	Tim2trgo
tim_itr1	tim8_trgo	tim8_trgo
tim_itr2	tim15_trgo	tim15_trgo

下表列出连接到 tim_etr 输入多路复用器的内部源

表 14.9 连接到 tim_etr 的输入多路复用器

定时器外部触发输入信号	定时器外部触发信号分配	
	TIM2	TIM3
tim_etr0	TIM2_ETR	TIM3_ETR
tim_etr1	cmp1_out	cmp1_out
tim_etr2	cmp2_out	cmp2_out

下表列出连接到 tim_ocref_clr 输入多路复用器的内部源

表 14.10 连接到 ocref_clr 的输入多路复用器

定时器 tim_ocref_clr 信号	定时器 tim_ocref_clr 信号分配	
	TIM2	TIM3
tim_ocref_clr0	cmp1_out	cmp1_out
tim_ocref_clr1	cmp2_out	cmp2_out

14.4.3 时基单元

可编程定时器的主要模块由一个 16 位/28 位计数器及其相关的自动重装寄存器组成。此计数器可采用递增方式计数。计数器的时钟可通过预分频器进行分频。

计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。

时基单元包含：

- 计数器寄存器（TIMx_CNT）
- 预分频器寄存器（TIMx_PSC）
- 自动装载寄存器（TIMx_ARR）

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存 器。预装载寄存器的内容既可以直接传送到影子寄存器，也可以在每次发生更新事件（UEV）时传送到影子寄存器，这取决于 TIMx_CR1 寄存器中的自动重载预装载使能位（ARPE）。当计数器达到上溢值（或者在递减计数时达到下溢值）并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生 进行详细介绍。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位（CEN）置 1 时，才会启动计数器（有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明）。

请注意，真正的计数器使能信号 CNT_EN 在 CEN 置 1 的一个时钟周期后被置 1。

预分频器描述

预分频器可对计数器时钟频率进行分频，分频系数介于 1 到 65536 之间。该预分频器基于 16 位/28 位寄存器（TIMx_PSC 寄存器）所控制的 16 位计数器。由于该控制寄存器具有缓冲功能，因此预分频器可实现实时更改。而新的预分频比将在下一更新事件发生时被采用.

下图给出了在预分频器运行时，更改计数器参数的例子。

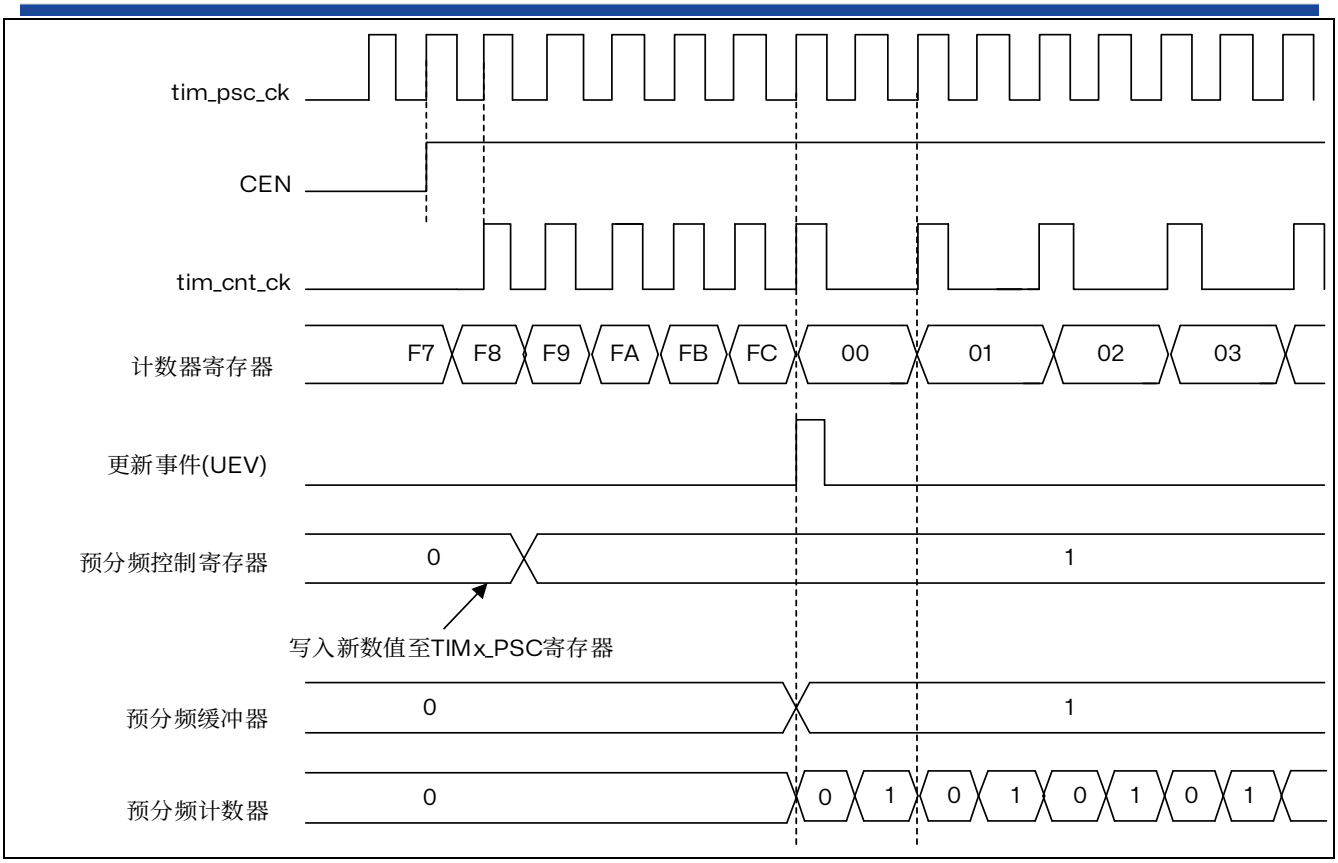


图 14.3 当预分频器的参数从 1 变到 2 时，计数器的时序图

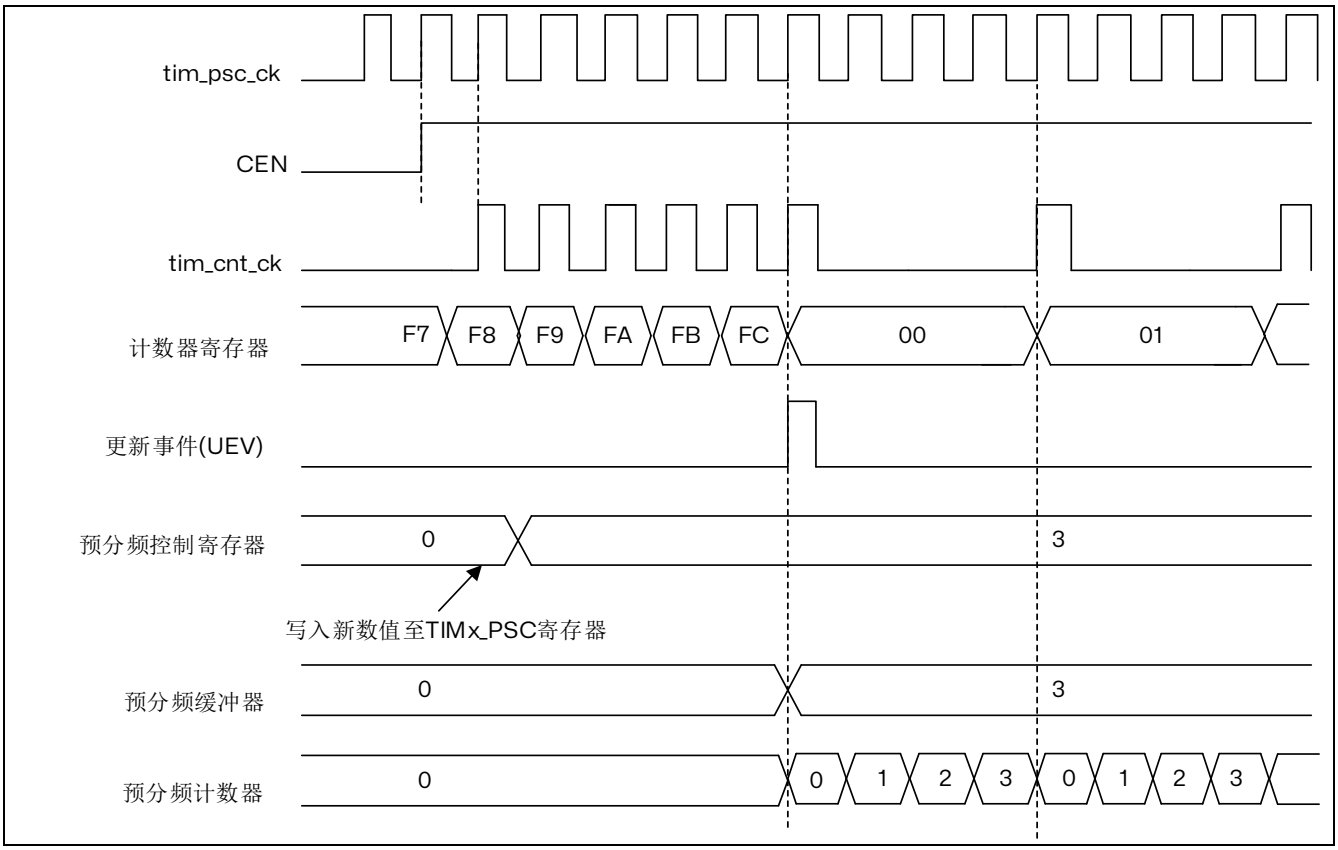


图 14.4 当预分频器的参数从 1 变到 4 时，计数器的时序图

14.4.4 计数器模式

递增计数模式

在递增计数模式下，计数器从 0 计数到自动重载值（TIMx_ARR 寄存器的内容），然后重新从 0 开始计数并生成计数器上溢事件。

每次发生计数器上溢时会生成更新事件，或将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）
- 自动重载影子寄存器将以预装载值进行更新

以下各图以一些示例说明当 TIMx_ARR=0x36 时不同时钟频率下计数器的行为。

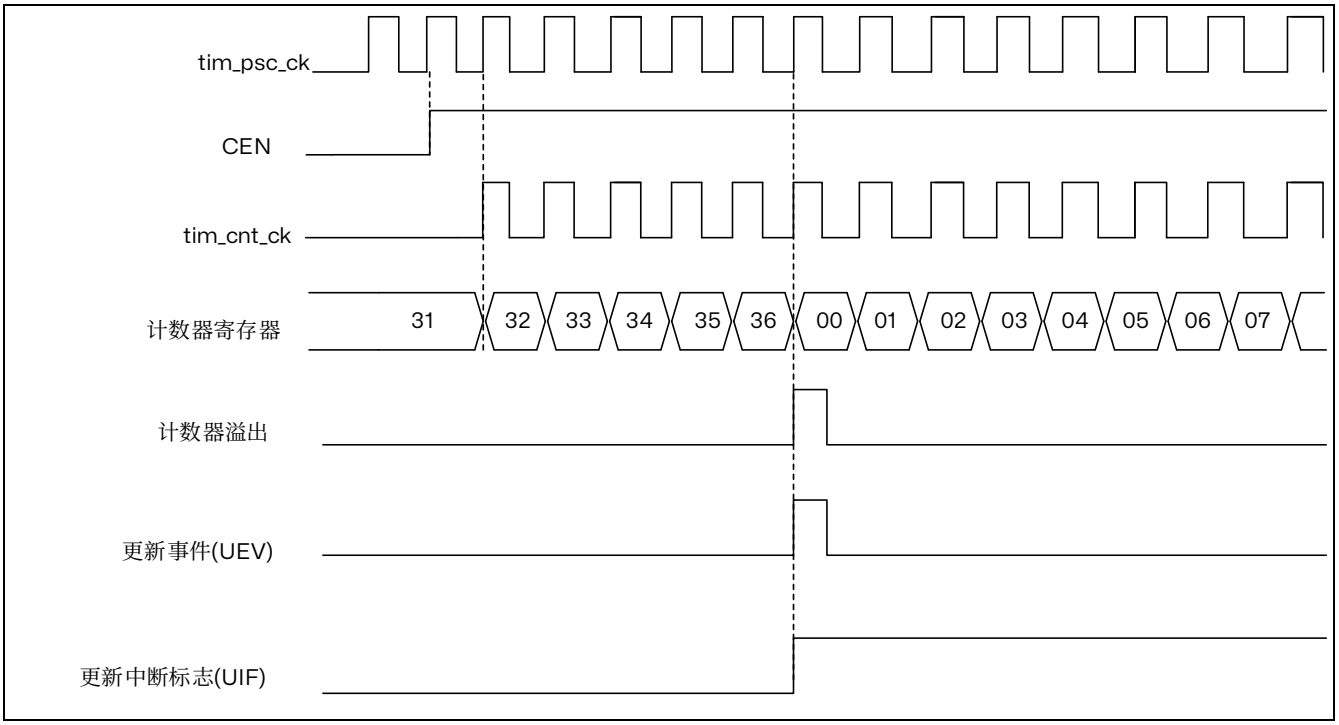


图 14.5 计数器时序图，内部时钟分频因子为 1

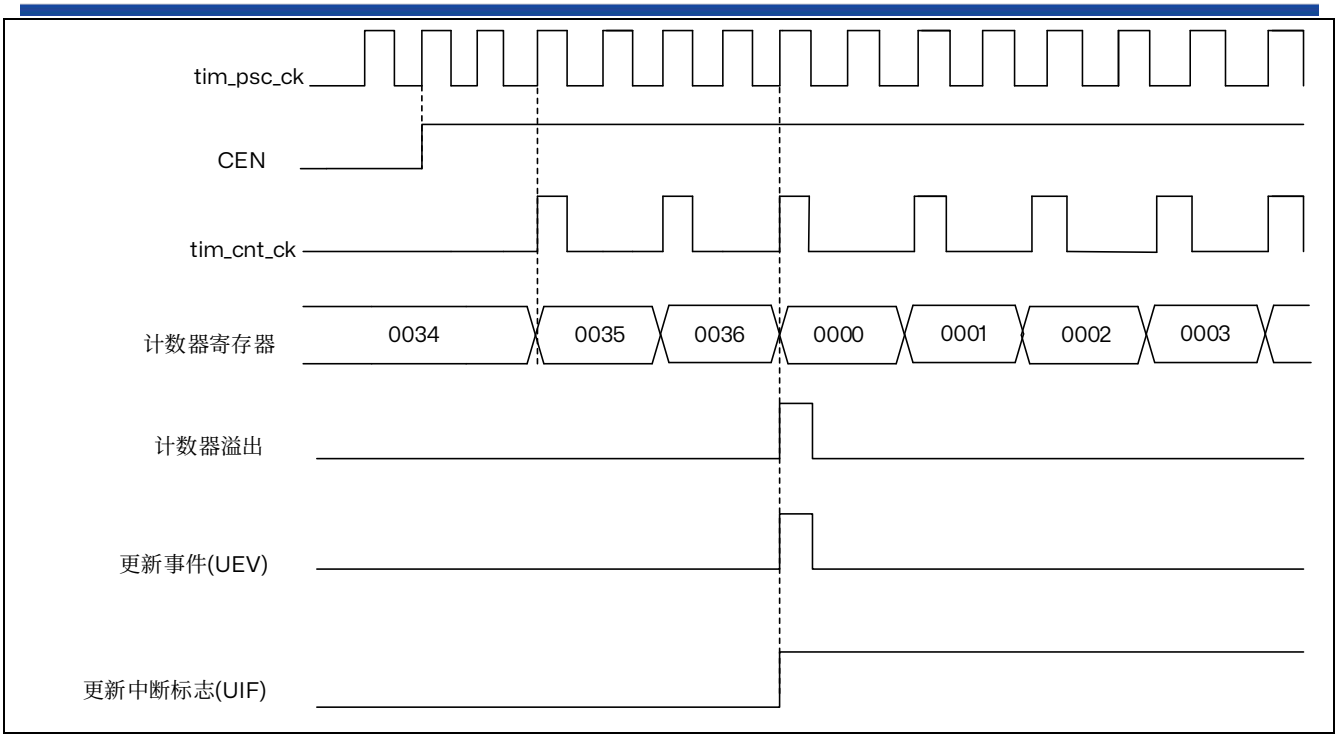


图 14.6 计数器时序图，内部时钟分频因子为 2

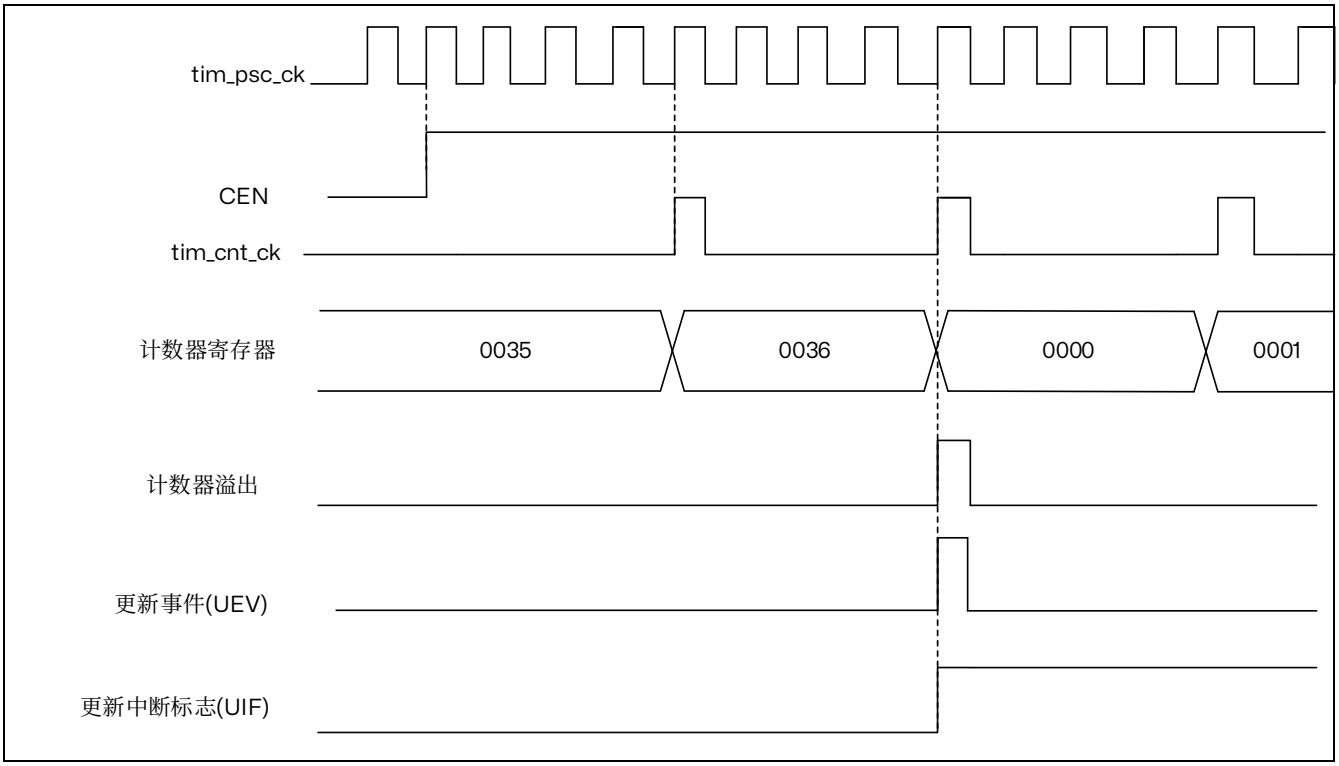


图 14.7 计数器时序图，内部时钟分频因子为 4

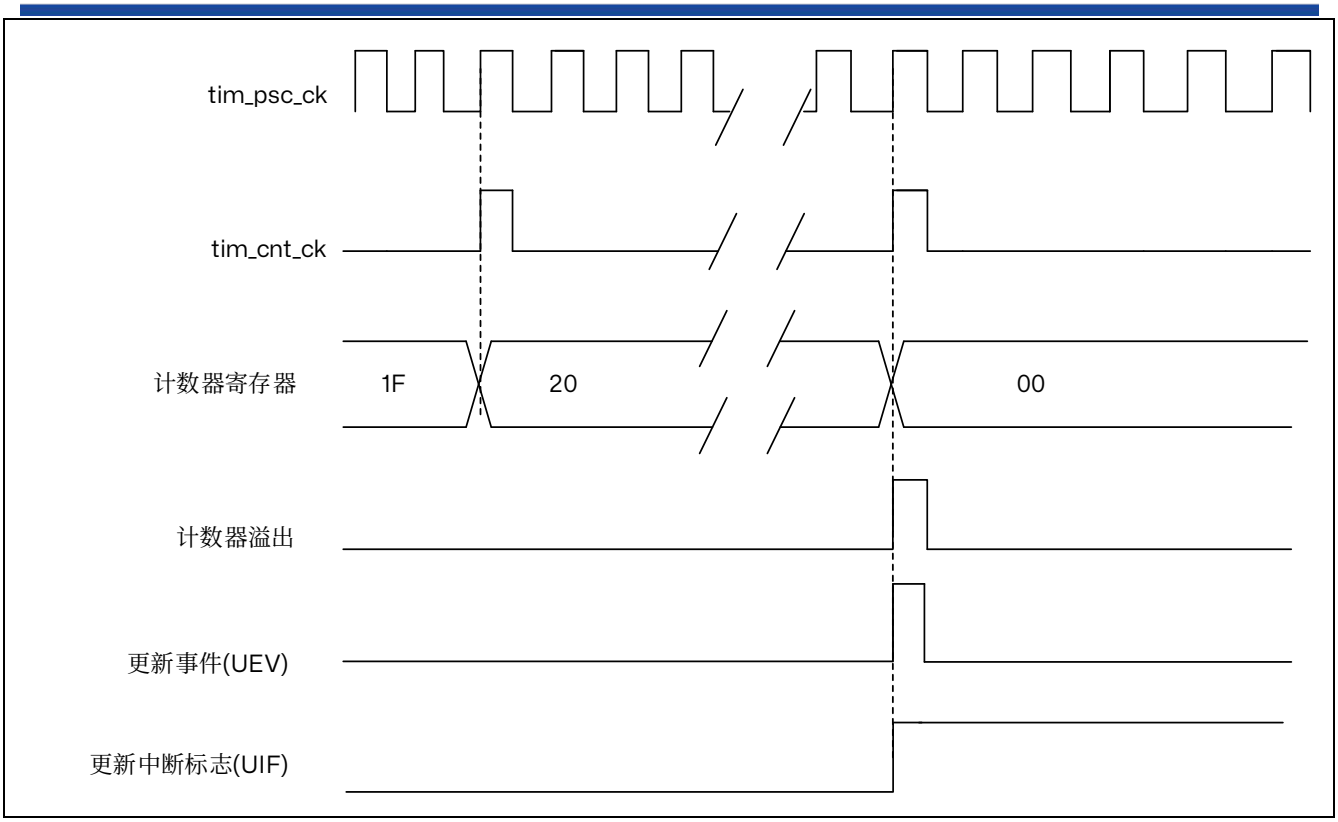


图 14.8 计数器时序图，内部时钟分频因子为 N

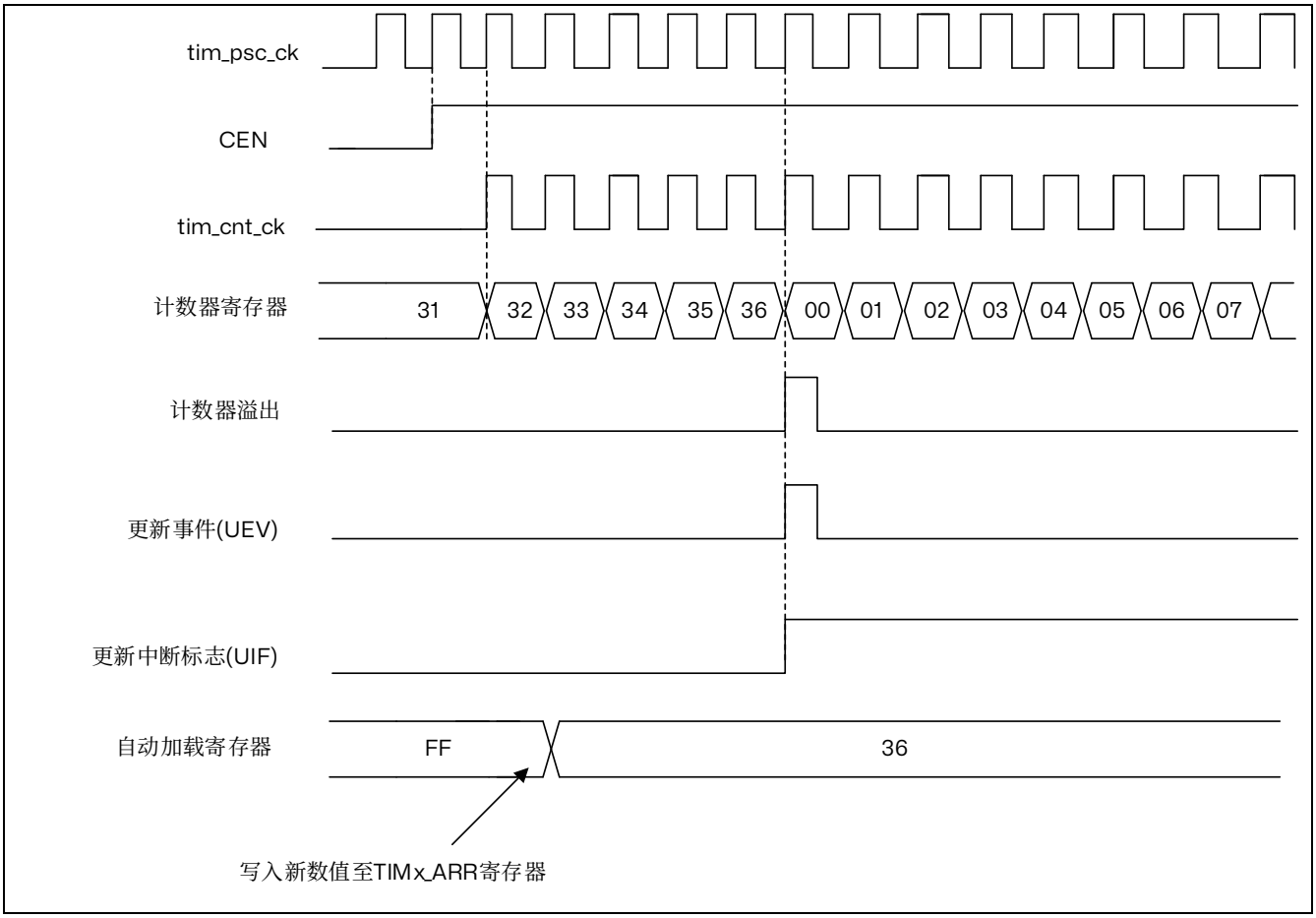


图 14.9 计数器时序图，当 ARPE=0 时的更新事件（TIMx_ARR 没有预装入）

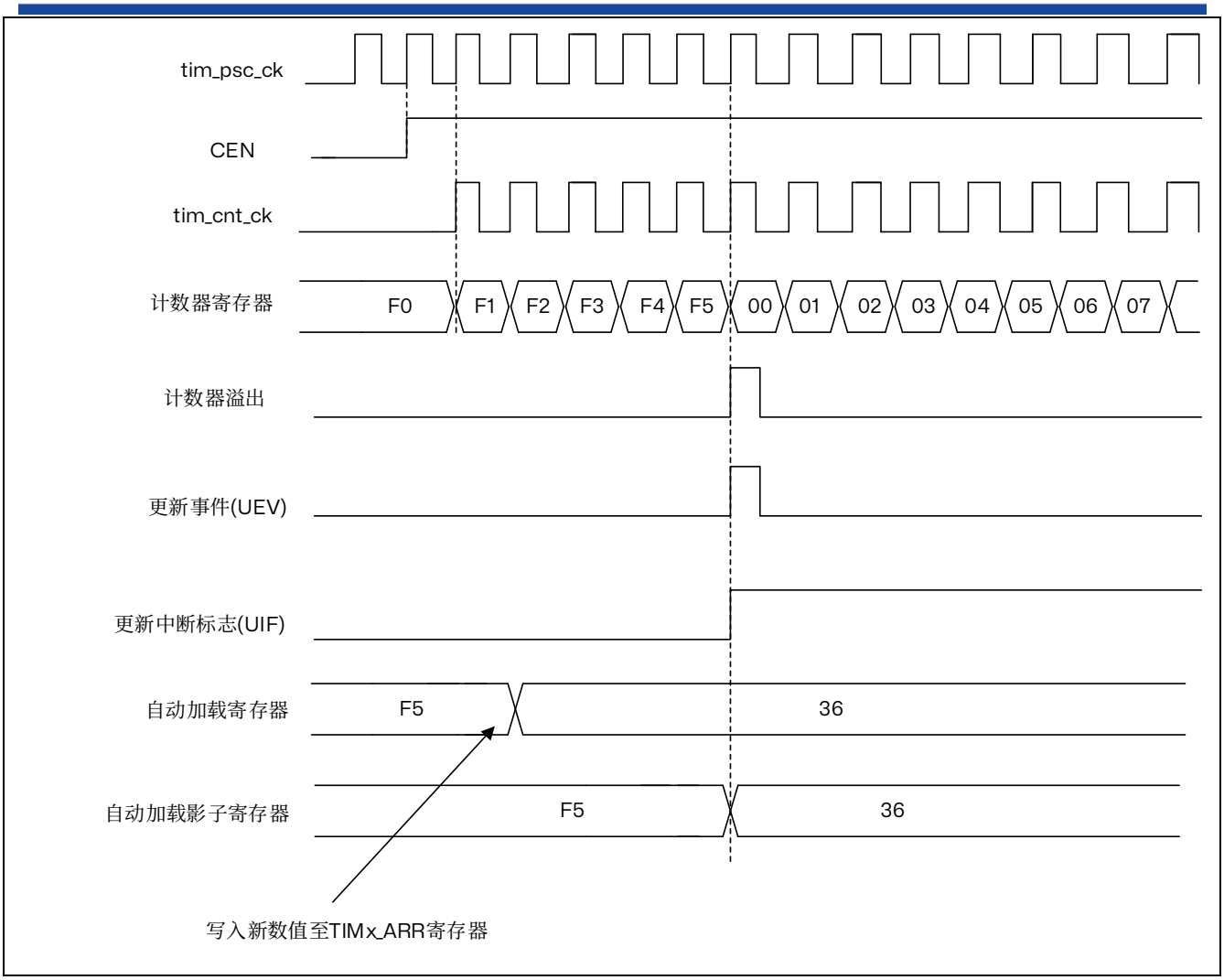


图 14.10 计数器时序图，当 ARPE=1 时的更新事件（预装入了 TIMx_ARR）

递减计数模式

在递减计数模式下，计数器从自动重载值（TIMx_ARR 寄存器的内容）开始递减计数到 0，然后重新从自动重载值开始计数并生成计数器下溢事件。

每次发生计数器下溢时会生成更新事件，或将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器会重新从当前自动重载值开始计数，而预分频器计数器则重新从 0 开始计数（但预分频比保持不变）。

此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。
- 自动重载活动寄存器将以预装载值（TIMx_ARR 寄存器的内容）进行更新。注意，自动重载寄存

器会在计数器重载之前得到更新，因此，下一个计数周期就是我们所希望的新的周期长度。

以下各图以一些示例说明当 TIMx_ARR=0x36 时不同时钟频率下计数器的行为。

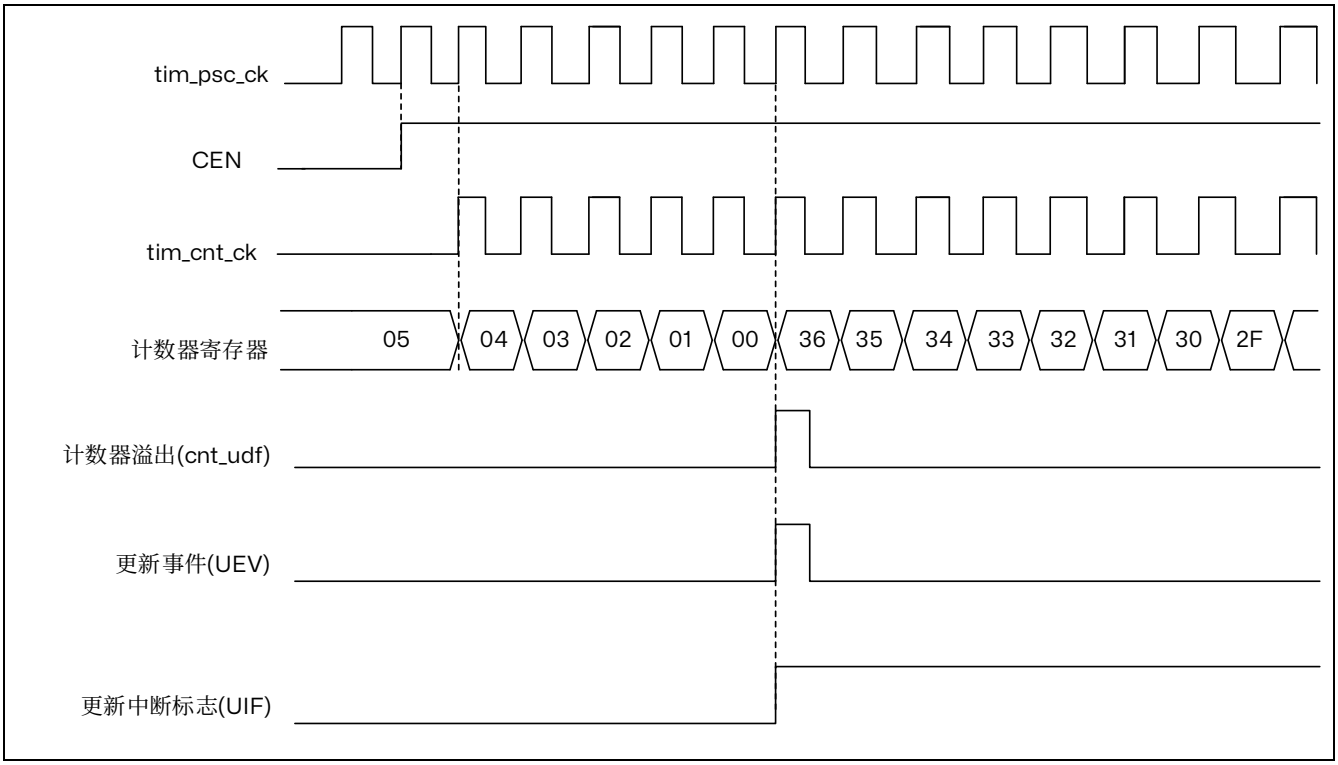


图 14.11 计数器时序图，内部时钟分频因子为 1

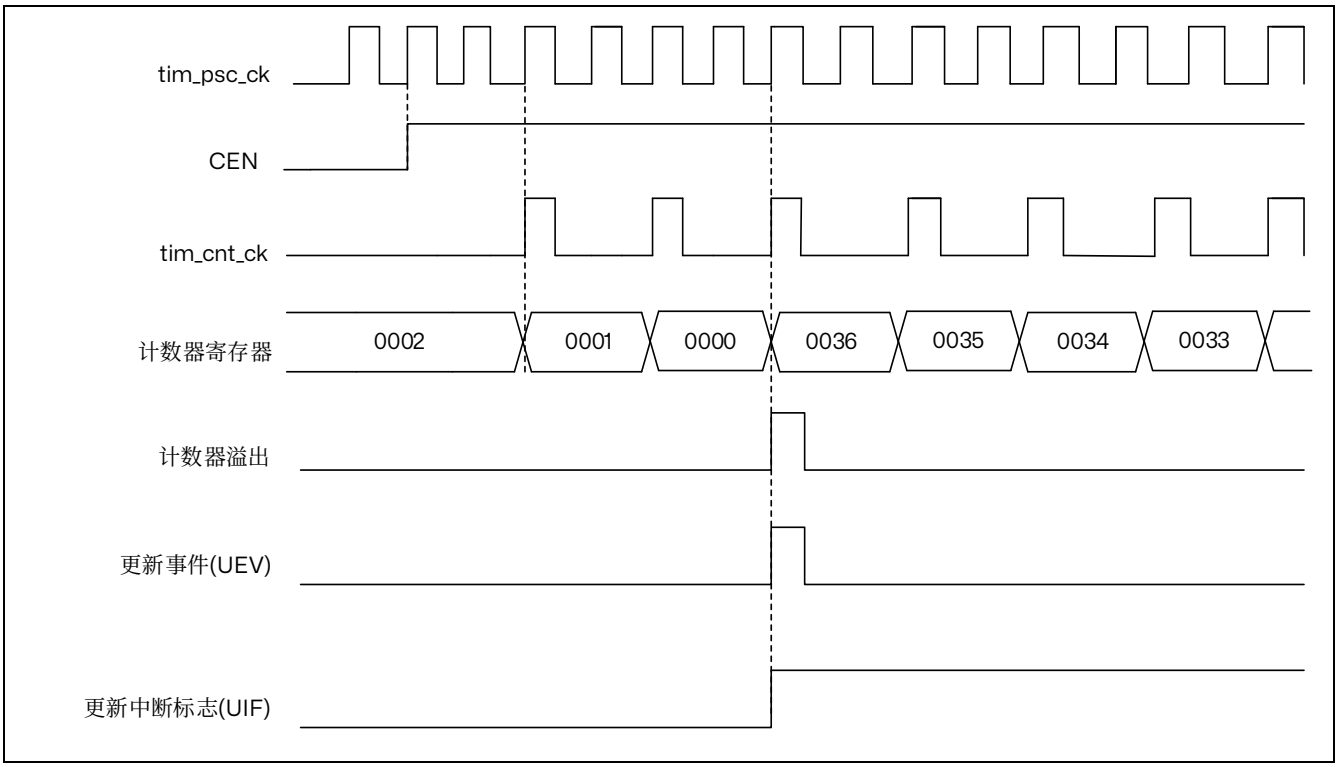


图 14.12 计数器时序图，内部时钟分频因子为 2

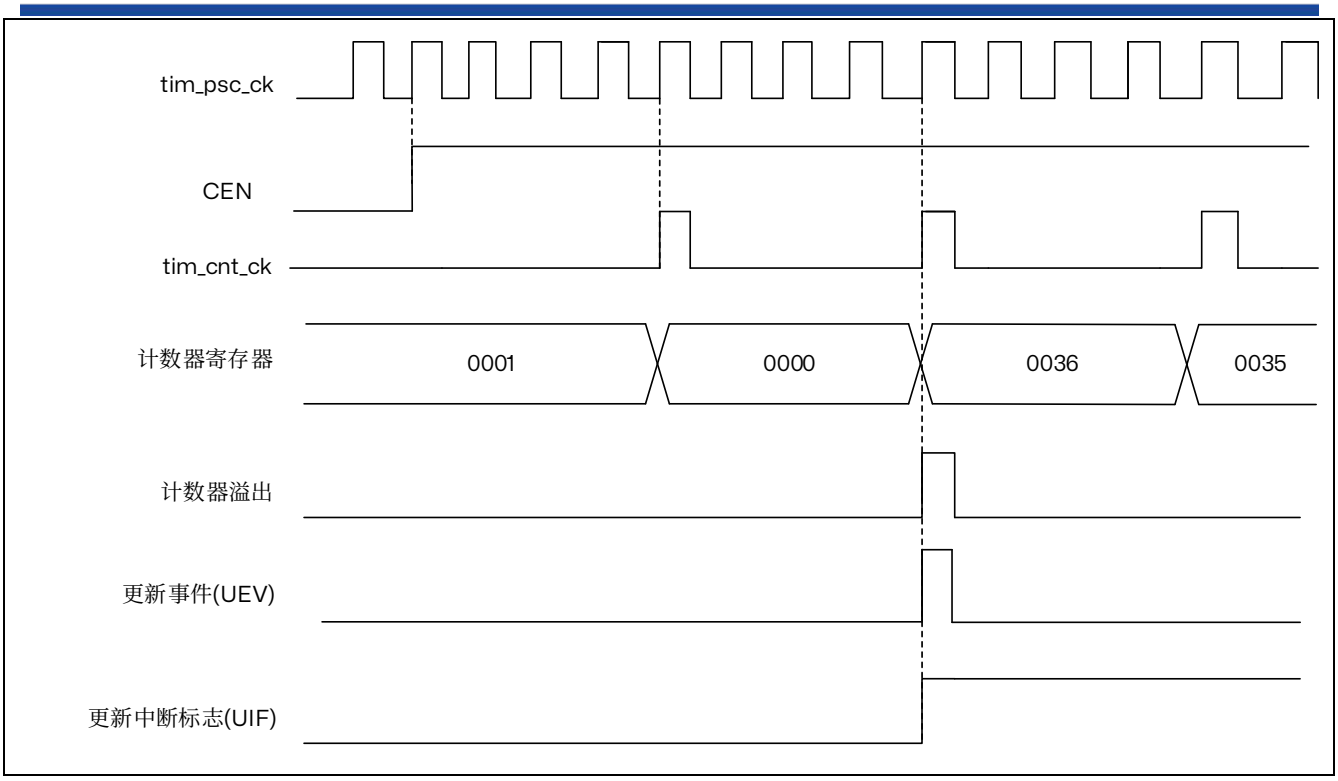


图 14.13 计数器时序图，内部时钟分频因子为 4

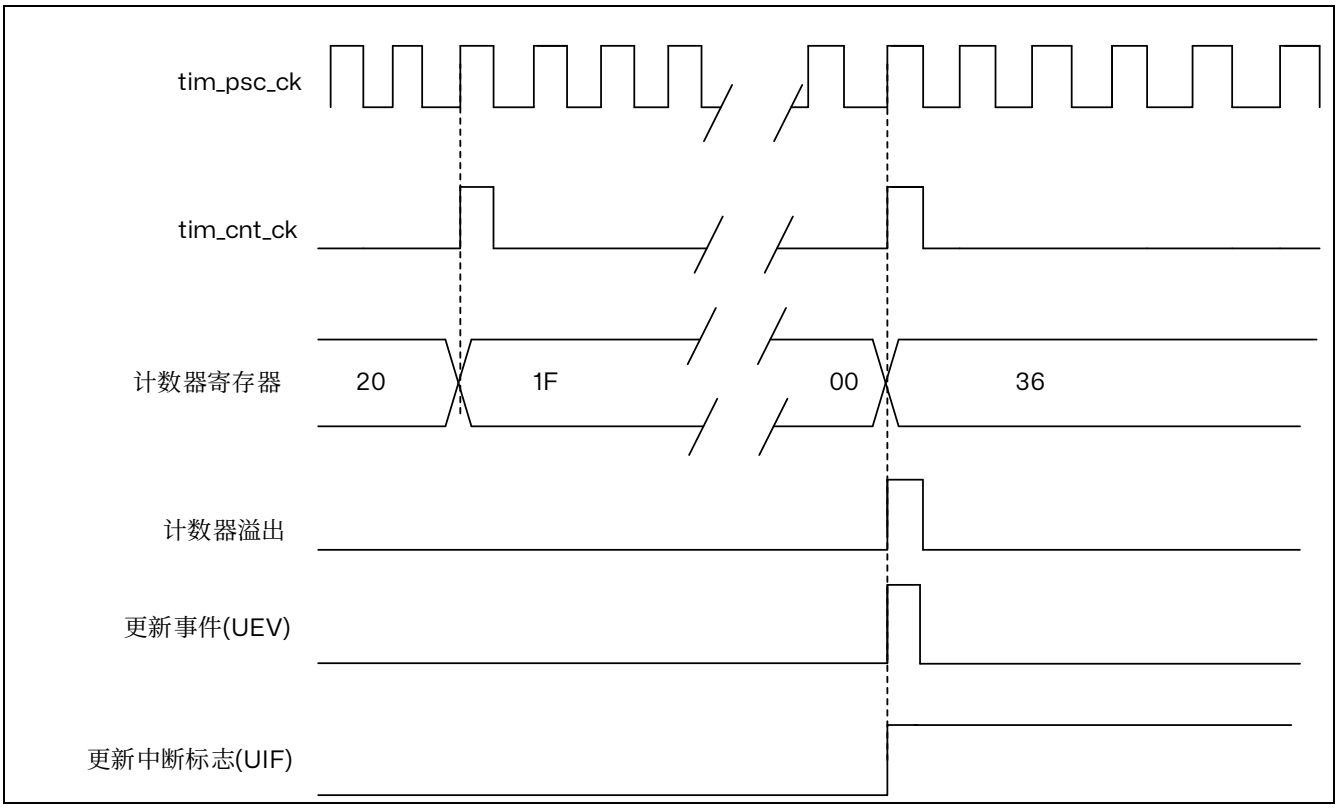


图 14.14 计数器时序图，内部时钟分频因子为 N

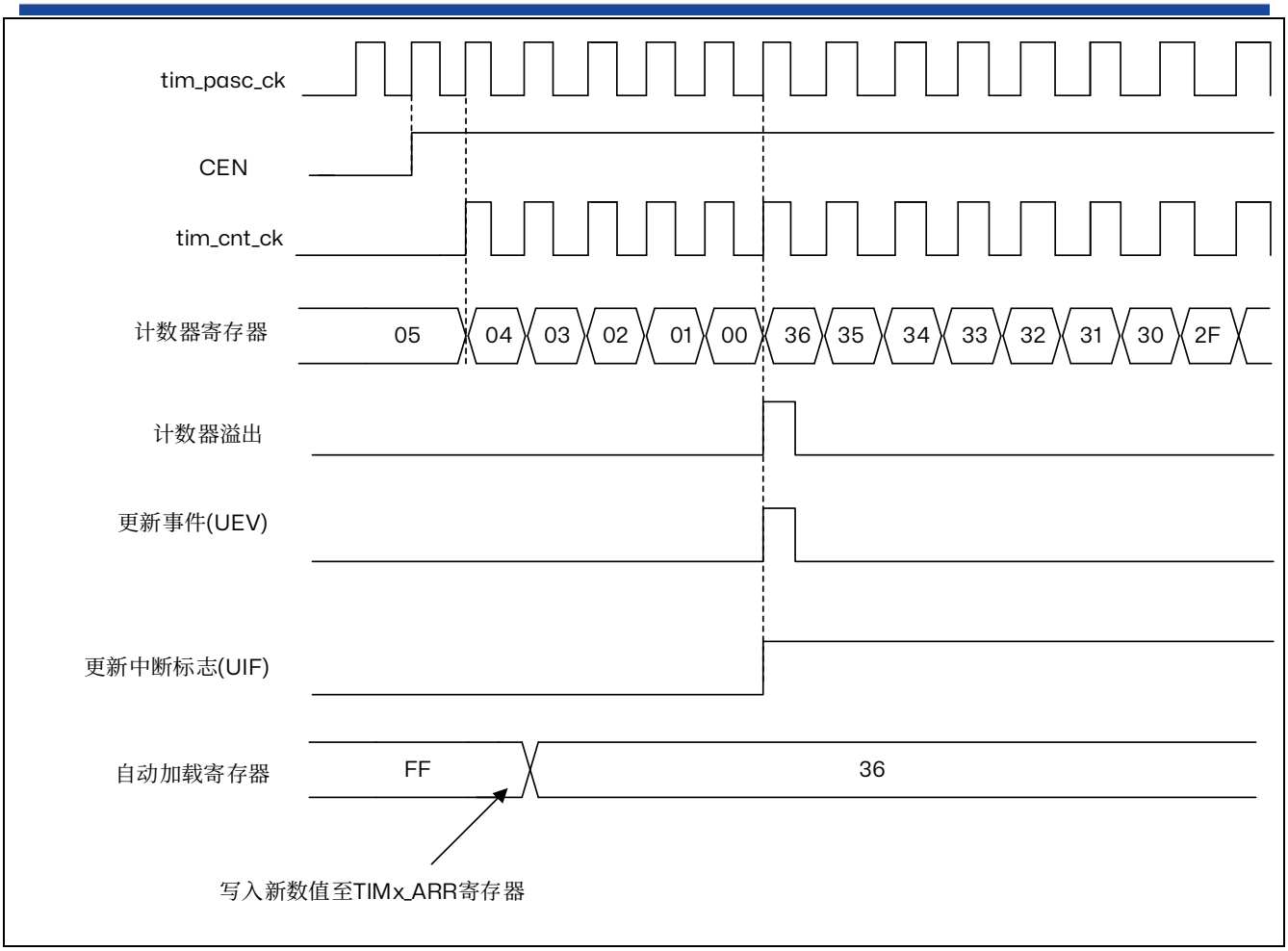


图 14.15 计数器时序图，更新事件

中央对齐模式（递增/递减计数）

在中央对齐模式下，计数器从 0 开始计数到自动重载值（TIMx_ARR 寄存器的内容）– 1，生成计数器上溢事件；然后从自动重载值开始向下计数到 1 并生成计数器下溢事件。之后从 0 开始重新计数。

当 TIMx_CR1 寄存器中的 CMS 位不为“00”时，中央对齐模式有效。将通道配置为输出模式时，其输出比较中断标志将在以下模式下置 1，即：计数器递减计数（中央对齐模式 1，CMS =“01”）、计数器递增计数（中央对齐模式 2，CMS =“10”）以及计数器递增/递减计数（中央对齐模式 3，CMS =“11”）。

此模式下无法写入方向位（TIMx_CR1 寄存器中的 DIR 位）。而是由硬件更新并指示当前计数器方向。

每次发生计数器上溢和下溢时都会生成更新事件，或将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。这种情况下，计数器以及预分频器计数器将重新从 0 开始计数。

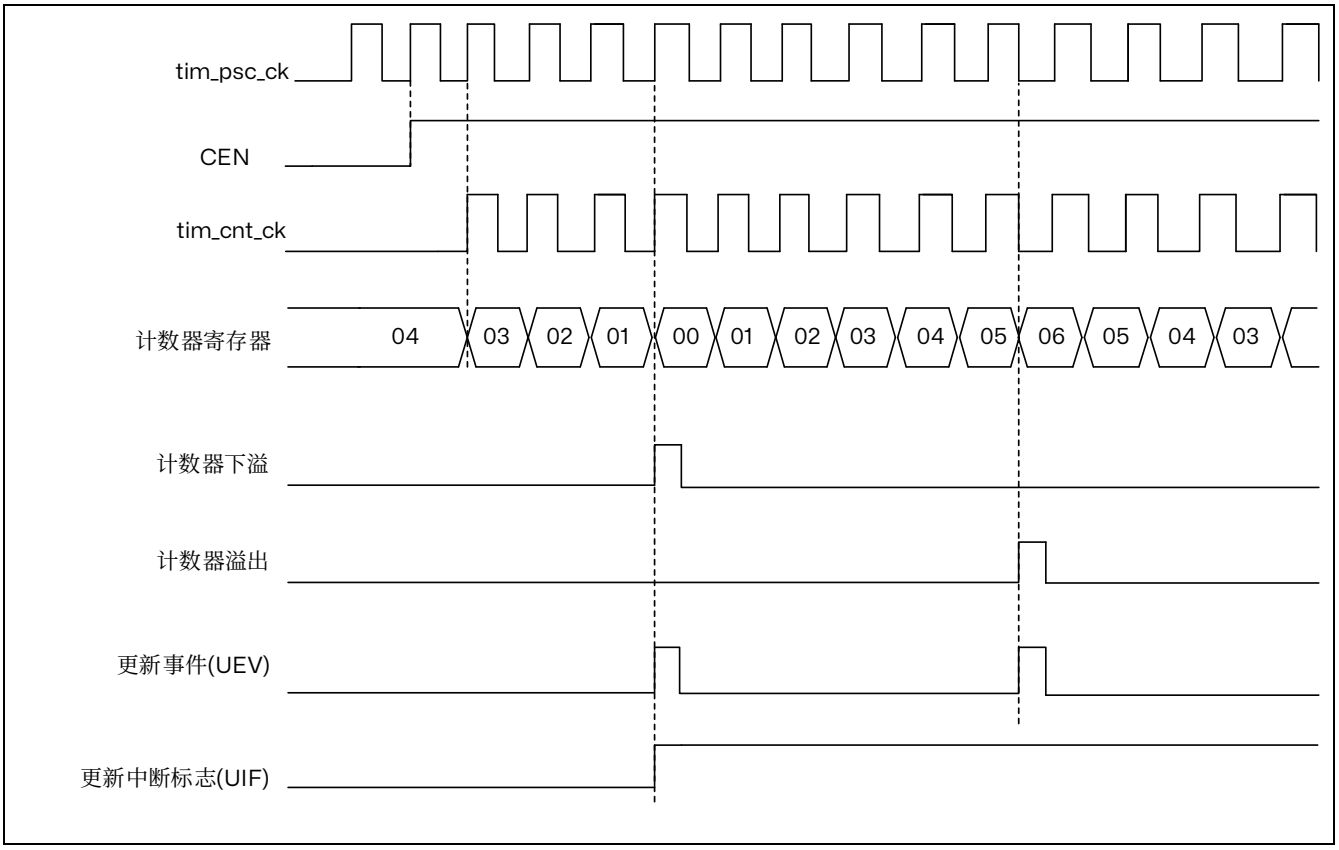
通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器仍会根据当前自动重载值进行递增和递减计数。

此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。
- 当自动重载活动寄存器将以预装载值（TIMx_ARR 寄存器的内容）进行更新。注意，如果更新操作是由计数器上溢触发的，则自动重载寄存器在重载计数器之前更新，因此，下一个计数周期就是我们所希望的新的周期长度（计数器被重载新的值）。

以下各图以一些示例说明不同时钟频率下计数器的行为。



1. 这里使用了中央对齐模式 1

图 14.16 计数器时序图，内部时钟分频因子为 1，TIMx_ARR=0x6

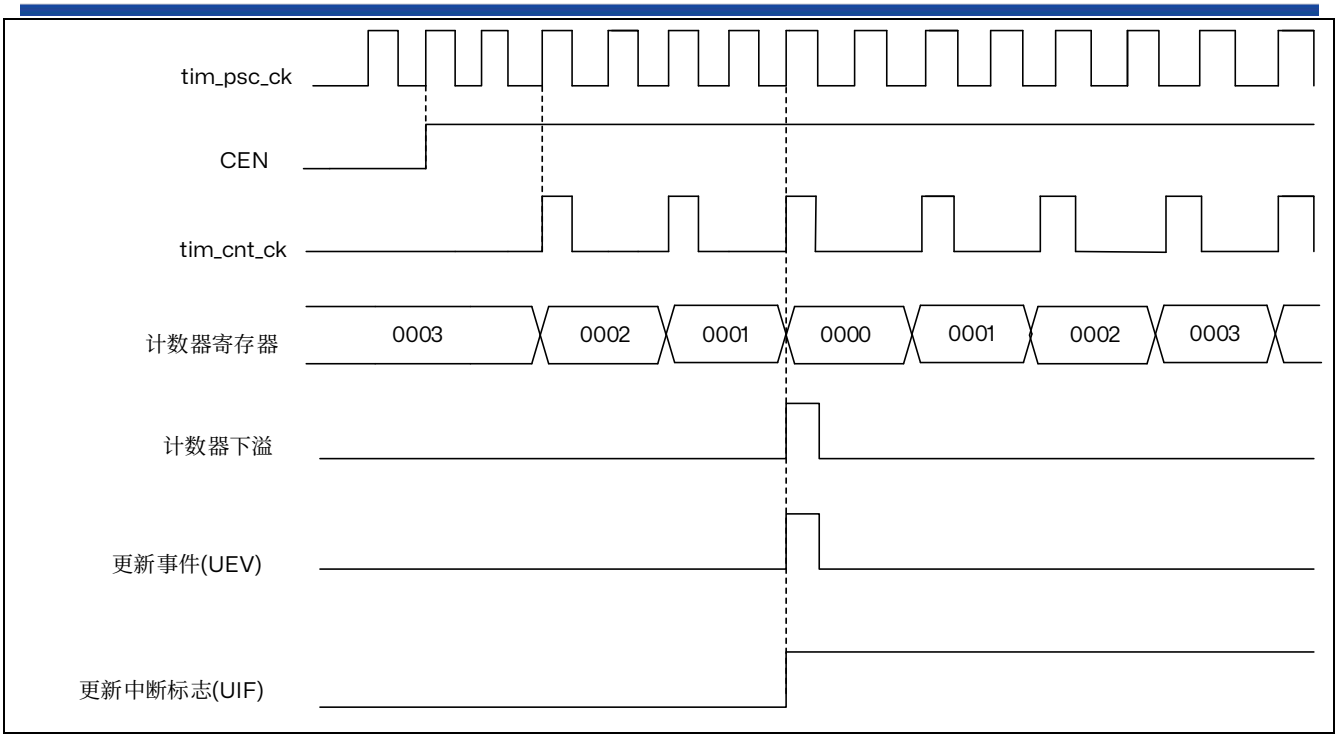


图 14.17 计数器时序图，内部时钟分频因子为 2

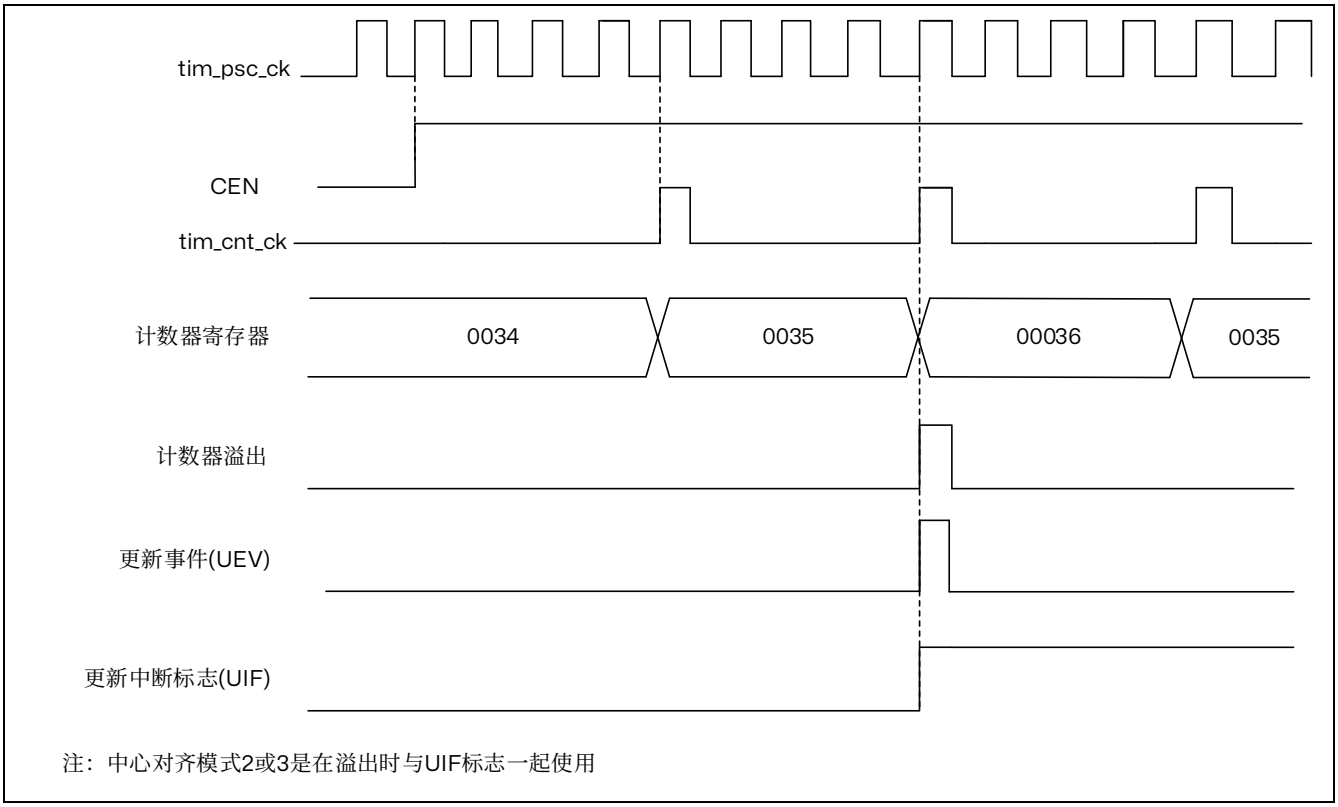


图 14.18 计数器时序图，内部时钟分频因子为 4，TIMx_ARR=0x36

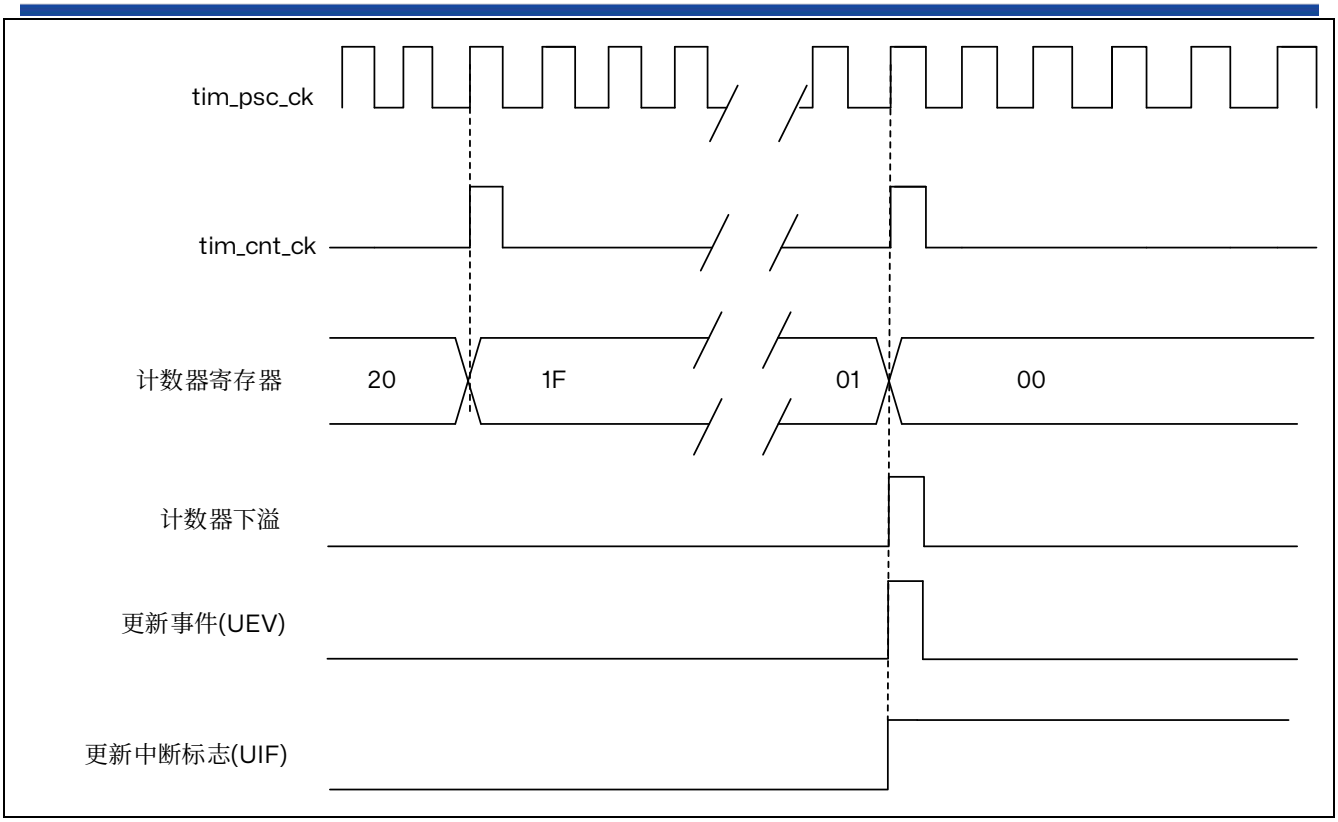


图 14.19 计数器时序图，内部时钟分频因子为 N

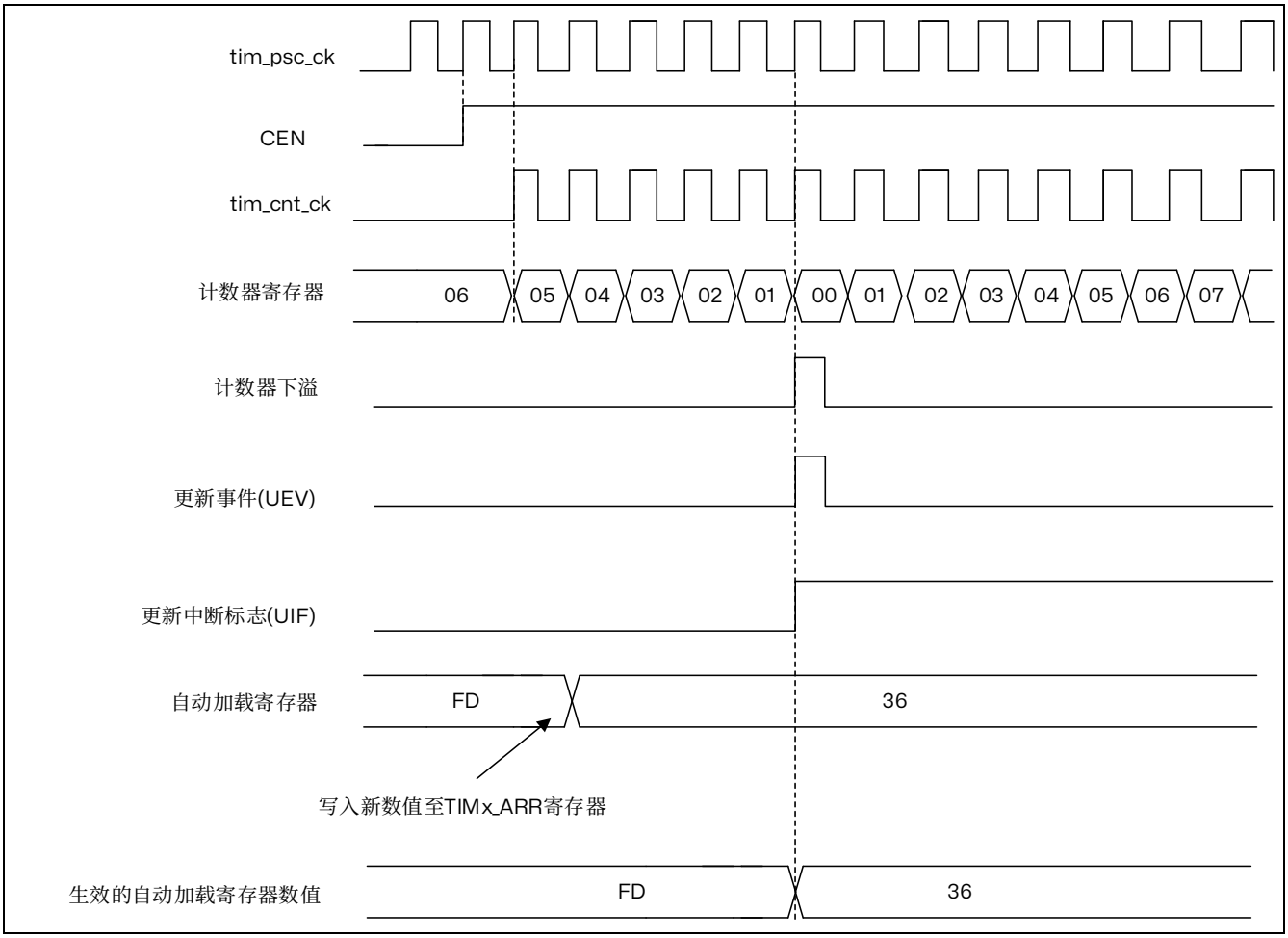


图 14.20 计数器时序图，ARPE=1 时的更新事件（计数器下溢）

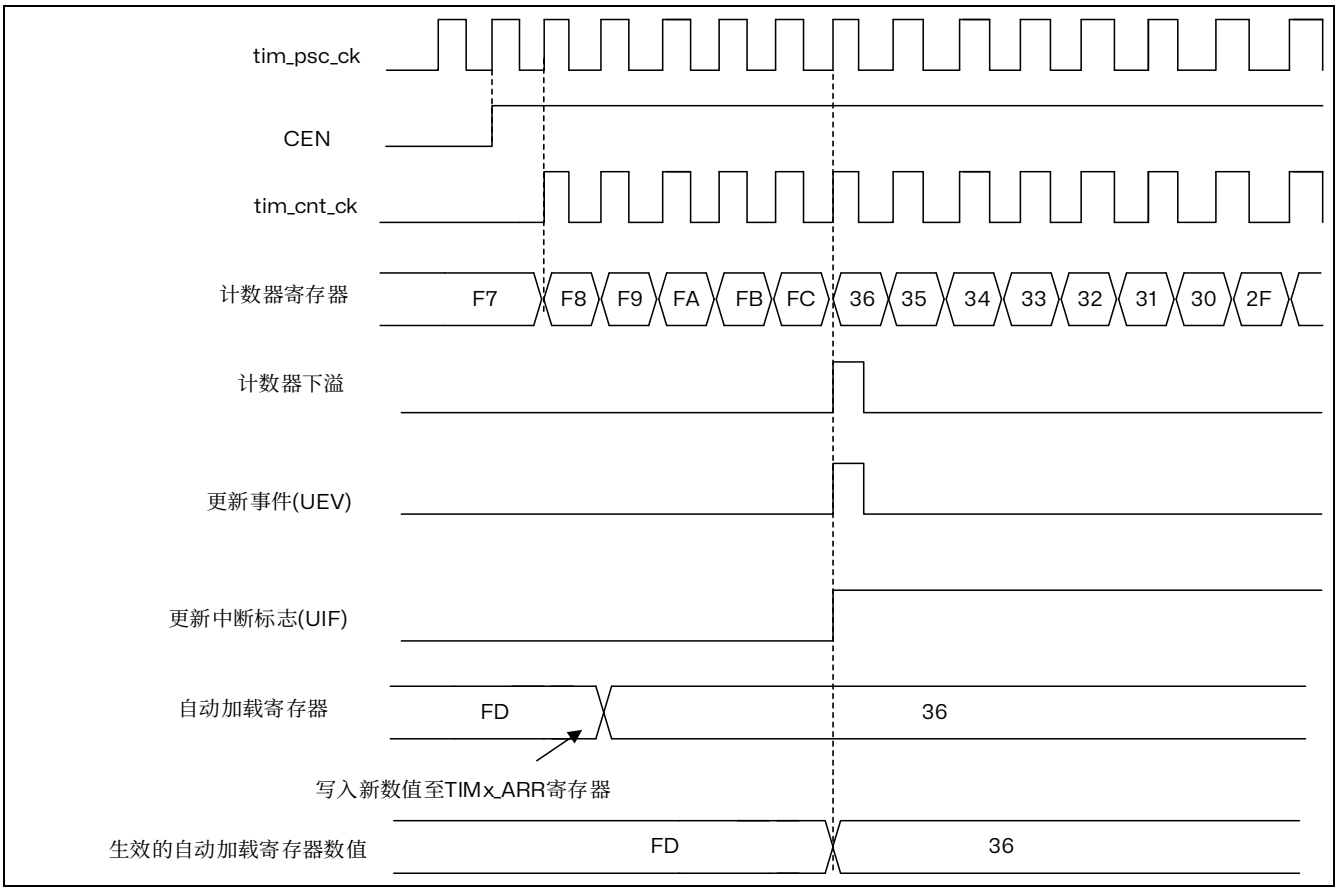


图 14.21 计数器时序图，ARPE=1 时的更新事件（计数器溢出）

14.4.5 时钟选择

计数器时钟可由下列时钟源提供：

- 内部时钟（tim_ker_ck）
- 外部时钟模式 1：外部输入引脚（tim_ti1 or tim_ti2）
- 外部时钟模式 2：外部触发输入（(tim_etr_in）
- 内部触发输入（tim_itr）：使用一个定时器作为另一个定时器的预分频器，例如可以将定时器 8 配置为定时器 2 的预分频器。

内部时钟源（tim_ker_ck）

如果禁止从模式控制器（TIMx_SMCR 寄存器中 SMS=000），则 CEN 位、DIR 位（TIMx_CR1 寄存器中）和 UG 位（TIMx_EGR 寄存器中）为实际控制位，并且只能通过软件进行更改（UG 除外，仍自动清零）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 tim_ker_ck 提供。

下图显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）。

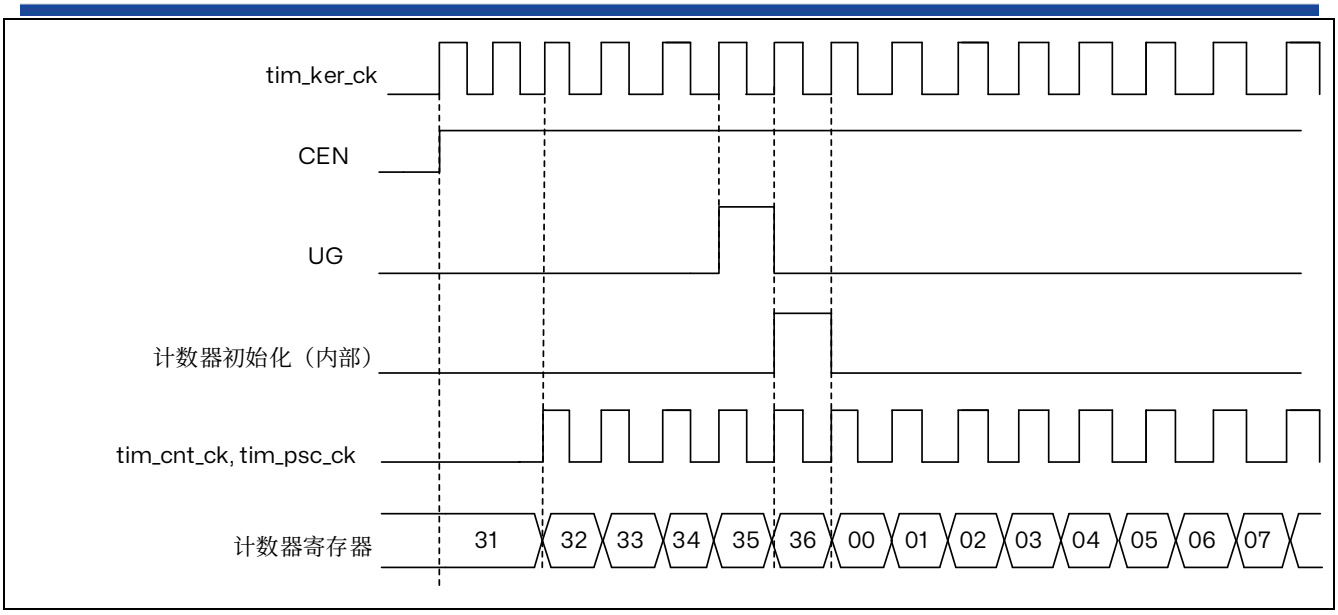


图 14.22 一般模式下的控制电路，内部时钟分频因子为 1

外部时钟源模式 1

当 TIMx_SMCR 寄存器中的 SMS=111 时，可选择此模式。计数器可在选定的输入信号上出现上升沿或下降沿时计数。

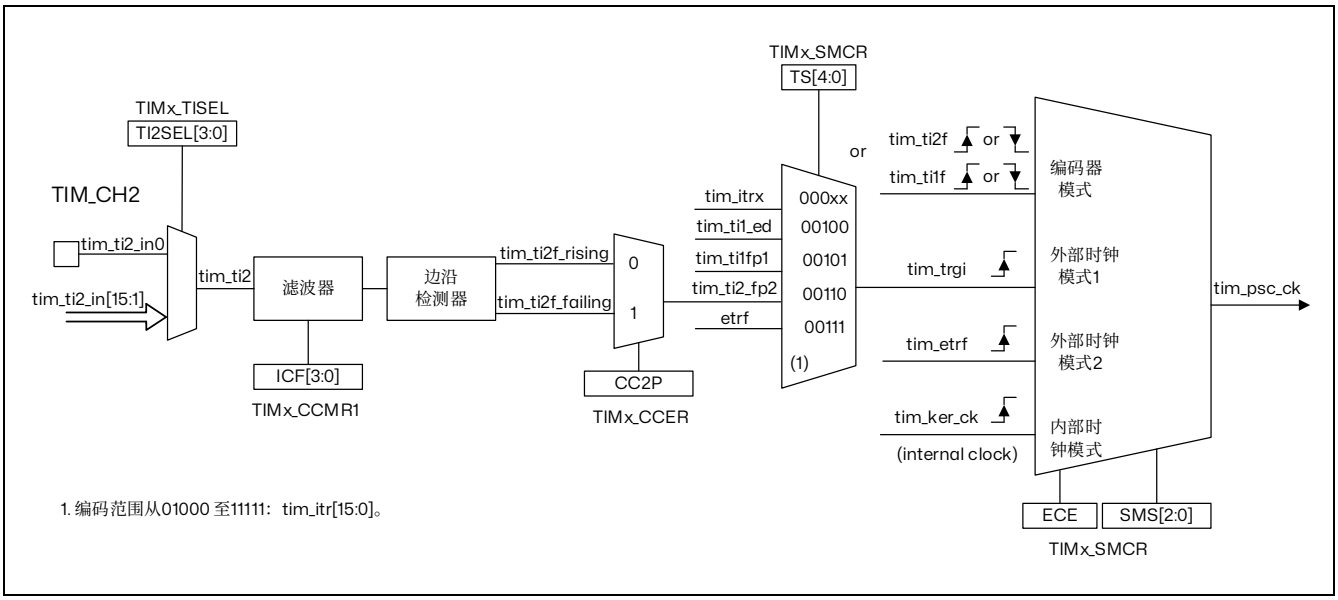


图 14.23 tim_ti2 外部时钟连接例子

例如，要使递增计数器在 tim_ti2 输入出现上升沿时计数，请执行以下步骤：

1. 通过在 TIMx_TISEL 寄存器中使用 TI2SEL[3:0]位选择合适的 tim_ti2_in[15:0]源（内部或外部）。
2. 通过在 TIMx_CCMR1 寄存器中写入 CC2S=“01”来配置通道 2，使其能够检测 tim_ti2 输入的上升沿。
3. 通过在 TIMx_CCMR1 寄存器中写入 IC2F[3:0]位来配置输入滤波时间（如果不需要任何滤波，请保持 IC2F=0000）。

注：由于捕获预分频器不用于触发操作，因此无需对其进行配置。

4. 通过在 TIMx_CCER 寄存器中写入 CC2P=0 和 CC2NP=0 来选择上升沿极性。
5. 通过在 TIMx_SMCR 寄存器中写入 SMS=111，使定时器在外部时钟模式 1 下工作。
6. 通过在 TIMx_SMCR 寄存器中写入 TS=110 来选择 tim_ti2 作为输入源。
7. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

当 tim_ti2 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。

tim_ti2 的上升沿与实际计数器时钟之间的延迟是由于 tim_ti2 输入的重新同步电路引起的。

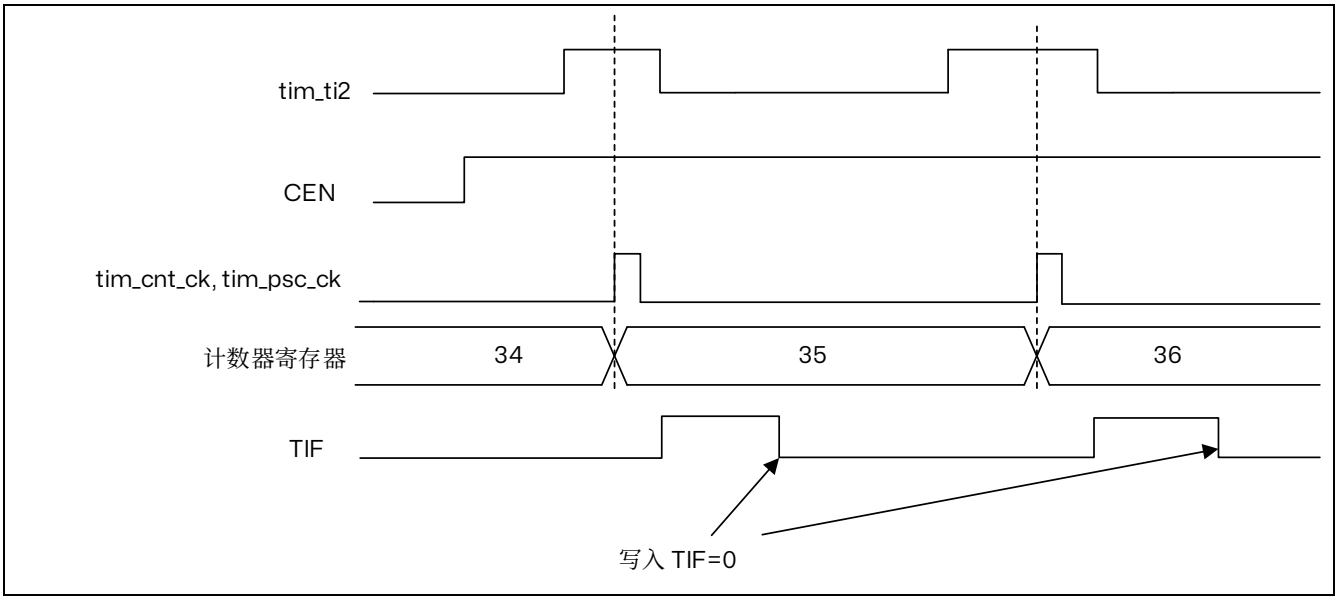


图 14.24 外部时钟模式 1 下的控制电路

外部时钟源模式 2

通过在 TIMx_SMCR 寄存器中写入 ECE=1 选择此模式。

计数器可在外部触发输入 ETR 出现上升沿或下降沿时计数。

下图是外部触发输入的框图

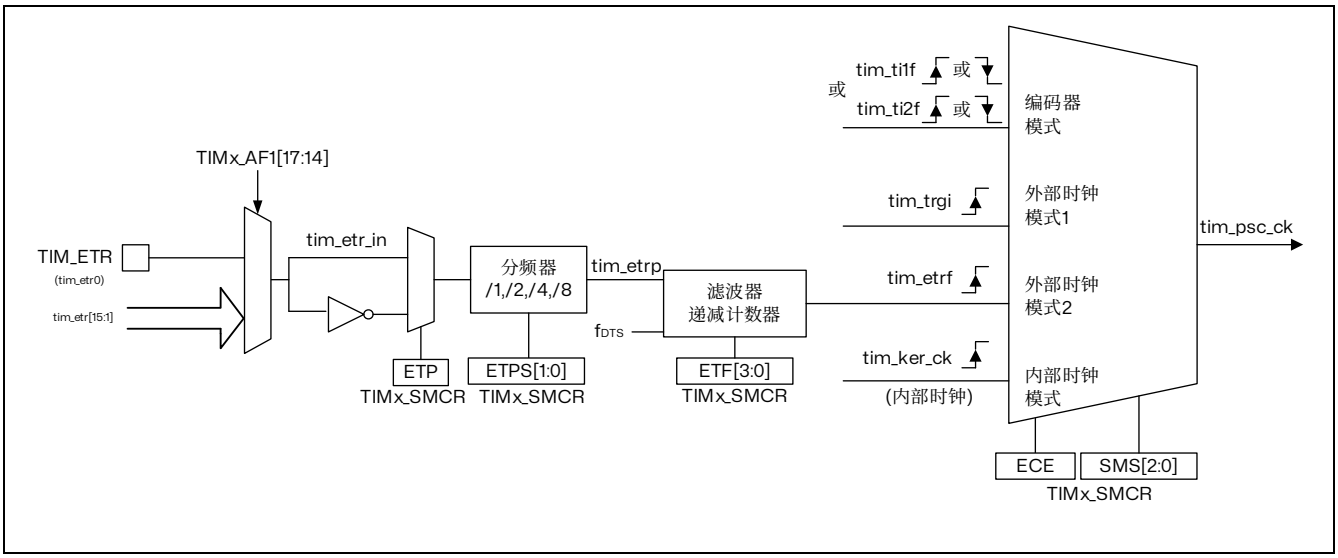


图 14.25 外部触发输入框图

例如，要使递增计数器在 ETR 每出现 2 个上升沿时计数，请执行以下步骤：

1. 通过在 TIMx_AF1 寄存器中使用 ETRSEL[3:0]位选择合适的 tim_etr_in 源（内部或外部）。
2. 由于此例中不需滤波器，因此在 TIMx_SMCR 寄存器中写入 ETF[3:0]=0000。
3. 通过在 TIMx_SMCR 寄存器中写入 ETPS[1:0]=01 来设置预分频器。
4. 通过在 TIMx_SMCR 寄存器中写入 ETP=0 来选择 tim_etr_in 引脚的上升沿检测。
5. 通过在 TIMx_SMCR 寄存器中写入 ECE=1 来使能外部时钟模式 2。
6. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

tim_etr_in 每出现 2 个上升沿，计数器计数一次。

在 tim_etr_in 的上升沿和计数器实际时钟之间的延时取决于在 tim_etrp 信号端的重新同步电路。因此，计数器可以正确捕获的最大频率最高为 TIMxCLK 频率的¼。当 ETRP 信号更快时，用户应通过适当的 ETPS 预分频器设置对外部信号进行分频。

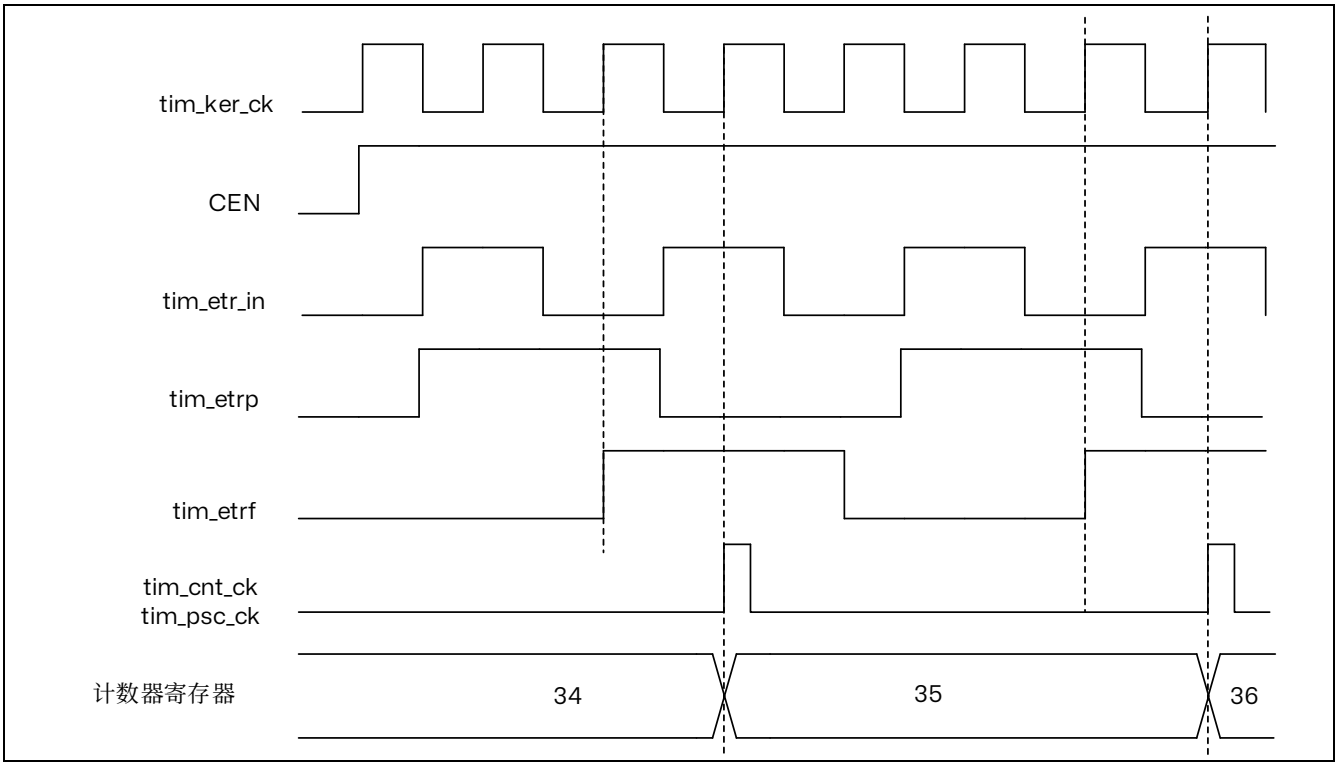


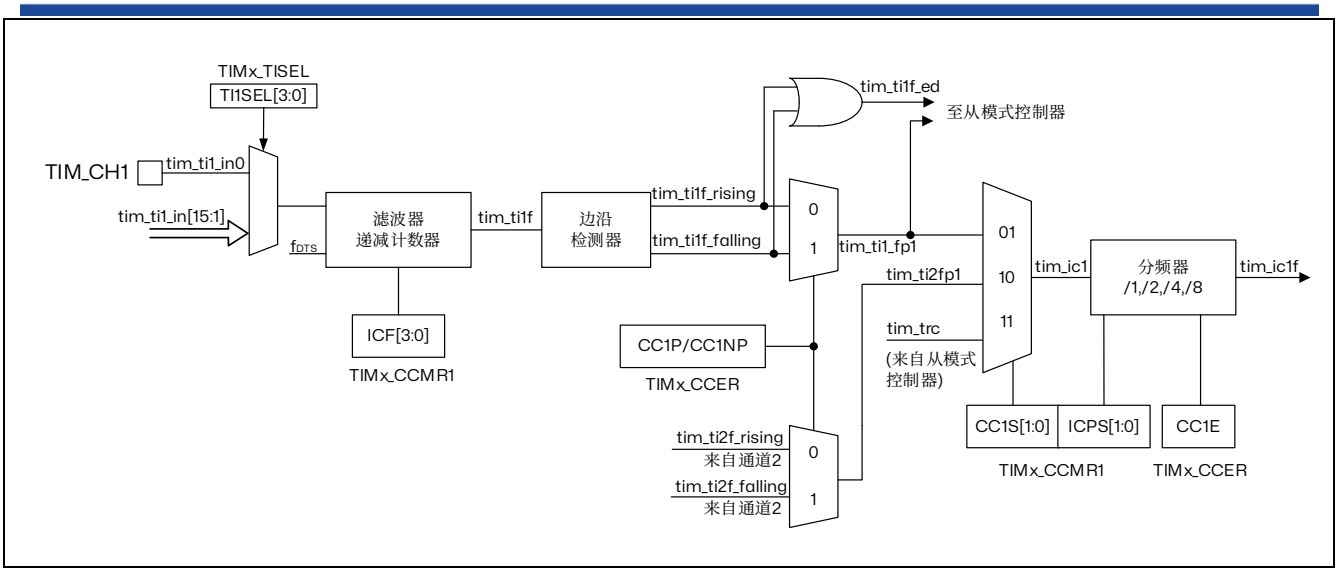
图 14.26 外部时钟模式 2 下的控制电路

14.4.6 捕获/比较通道

每个捕获/比较通道均围绕一个捕获/比较寄存器（包括一个影子寄存器）、一个捕获输入阶段（数字滤波、多路复用和预分频器）和一个输出阶段（比较器和输出控制）构建而成。

下图概括介绍了一个捕获/比较通道。

输入阶段对相应的 tim_tix 输入进行采样，生成一个滤波后的信号 tim_tixf。然后，带有极性选择功能的边沿检测器生成一个信号（tim_tixfpy），该信号可用作从模式控制器的触发输入，也可用作捕获命令。该信号先进行预分频（ICxPS），而后再进入捕获寄存器。



输出阶段生成一个中间波形作为基准：OCxRef（高电平有效）。链的末端决定最终输出信号的极性。

图 14.27 捕获/比较通道（如：通道 1 输入部分）

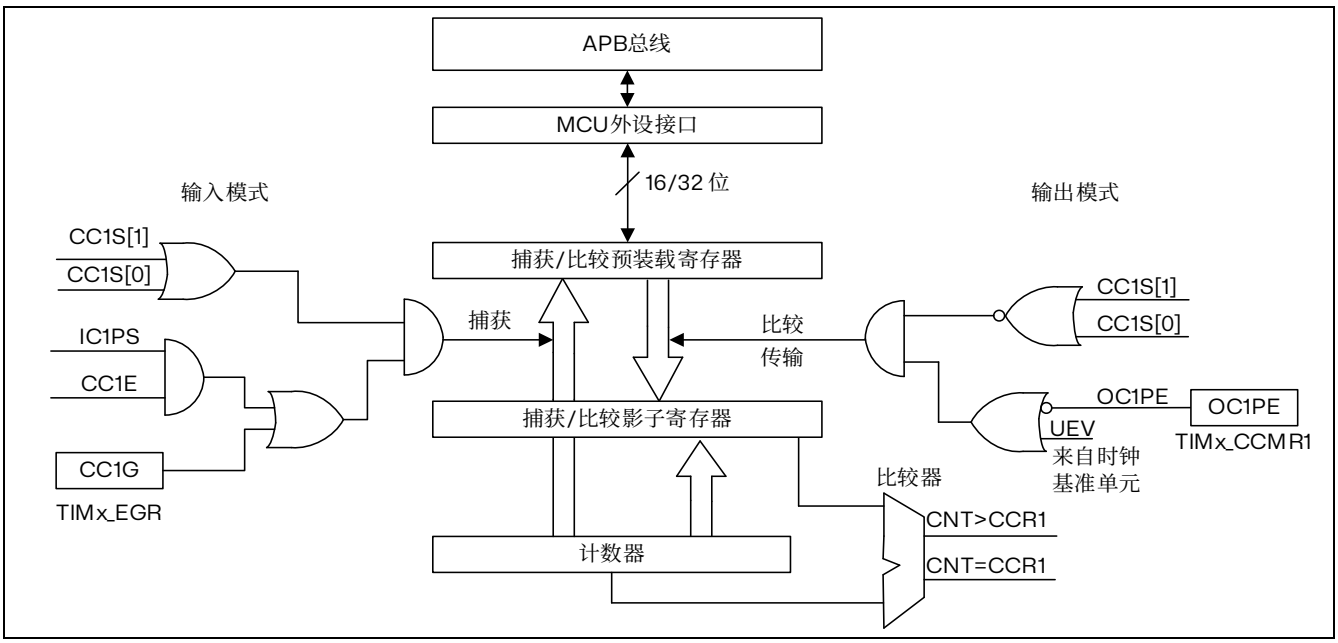


图 14.28 捕获/比较通道 1 的主电路

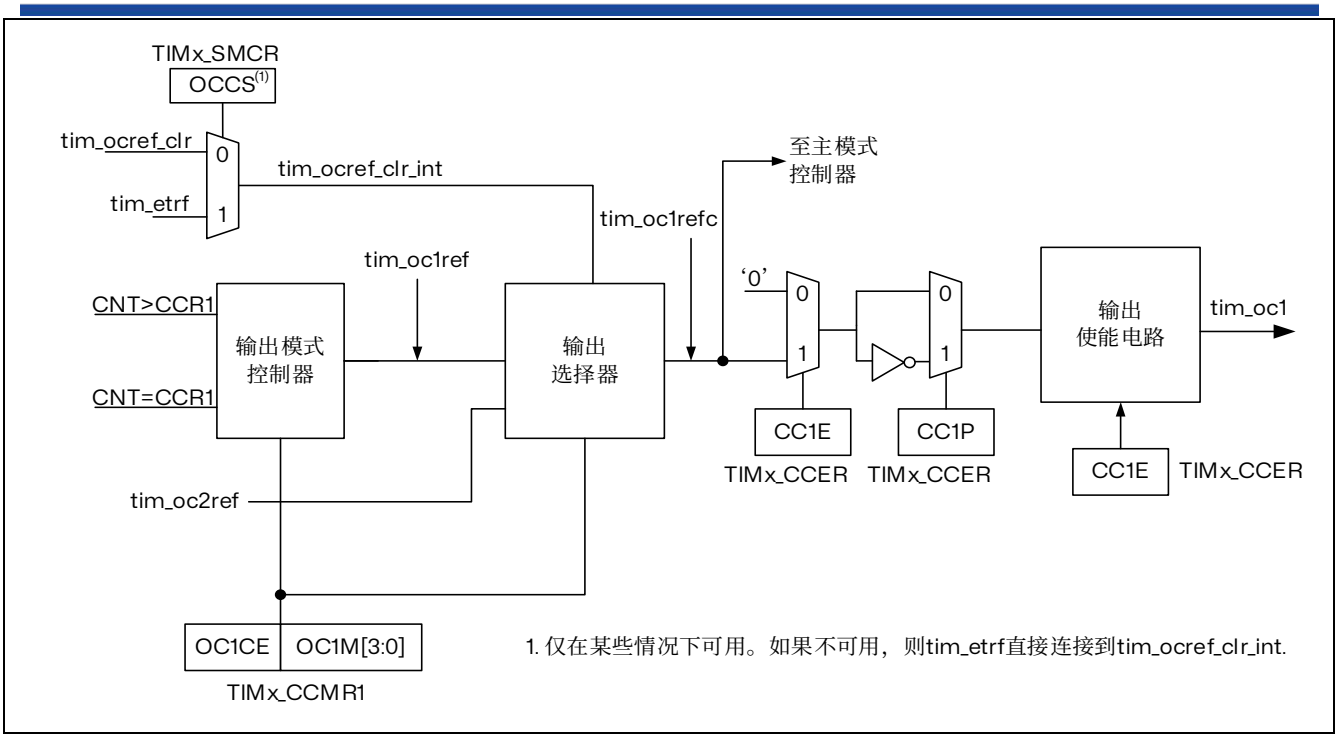


图 14.29 捕获/比较通道的输出阶段（通道 1，通道 2、3 和 4 同上）

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。始终可通过读写操作访问预装载寄存器。

在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容复制到预装载寄存器中。

在比较模式下，预装载寄存器的内容将复制到影子寄存器中，然后将影子寄存器的内容与计数器进行比较。

14.4.7 输入捕获模式

在输入捕获模式下，当相应的 ICx 信号检测到跳变沿后，将使用捕获/比较寄存器(TIMx_CCRx) 来锁存计数器的值。发生捕获事件时，会将相应的 CCXIF 标志（TIMx_SR 寄存器）置 1，并可发送中断请求（如果已使能）。如果发生捕获事件时 CCxIF 标志已处于高位，则会将重复捕获标志 CCxOF（TIMx_SR 寄存器）置 1。可通过软件向 CCxIF 写入 0 来给 CCxIF 清零，或读取存储在 TIMx_CCRx 寄存器中的已捕获数据。向 CCxOF 写入 0 后会将其清零。

以下示例说明了如何在 tim_ti1 输入出现上升沿时将计数器的值捕获到 TIMx_CCR1 中。具体操作步骤如下：

1. 通过在 TIMx_TISEL 寄存器中使用 TI1SEL[3:0]位选择合适的 tim_tix_in[15:0]源（内部或外部）。
2. 选择有效输入：TIMx_CCR1 必须连接到 tim_ti1 输入，因此向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01。只要 CC1S 不等于 00，就会将通道配置为输入模式，并且 TIMx_CCR1 寄存器将处于只读状态。
3. 根据连接到定时器的信号，对所需的输入滤波时间进行编程（如果输入为 tim_tix 输入之一，则对 TIMx_CCMRx 寄存器中的 ICxF 位进行编程）。假设信号变化时，输入信号最多在 5 个内部时钟周期内发生抖动。因此，我们必须将滤波时间设置为大于 5 个内部时钟周期。在检测到 8 个具有新电平的连续采样（以 f_{DTS} 频率采样）后，可以确认 tim_ti1 上的跳变沿。然后向 TIMx_CCMR1 寄存器中的 IC1F 位写入 0011。
4. 通过向 TIMx_CCER 寄存器中的 CC1P 位和 CC1NP 位写入 0，选择 tim_ti1 通道的有效转换边沿

（本例中为上升沿）。

5. 对输入预分频器进行编程。在本例中，我们希望每次有效转换时都执行捕获操作，因此需要禁止预分频器（向 TIMx_CCMR1 寄存器中的 IC1PS 位写入 00）。
6. 通过将 TIMx_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
7. 如果需要，可通过将 TIMx_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求。

发生输入捕获时：

- 发生有效跳变沿时，TIMx_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。
- 根据 CC1IE 位生成中断。

要处理重复捕获，建议在读出捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出标志之后与读取数据之前可能出现的重复捕获信息。

注：通过软件将 TIMx_EGR *寄存器中的相应* CCxG *位置*1 *可生成* IC *中断请求。*

14.4.8 PWM 输入模式

此模式允许测量连接到单个 tim_tix 输入的 PWM 信号的周期和占空比：

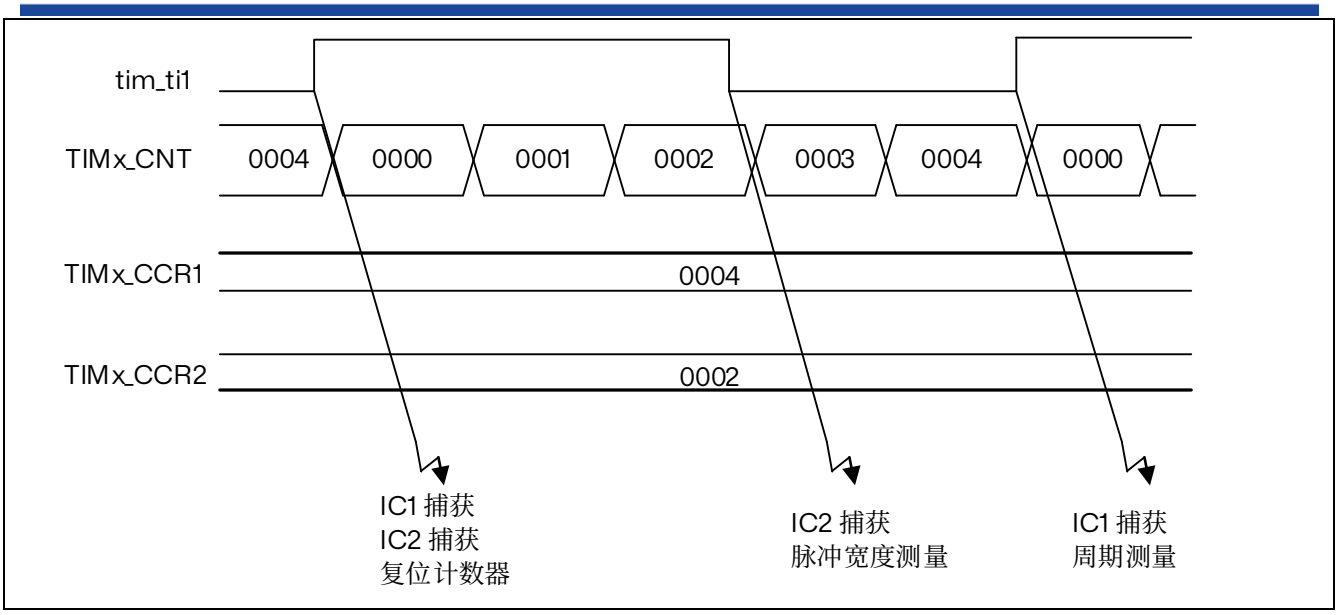
- TIMx_CCR1 寄存器保持周期值（两个连续上升沿之间的间隔）。
- TIMx_CCR2 寄存器保持脉冲宽度（两个连续上升沿和下降沿之间的间隔）。

此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

- 两个 ICx 信号被映射至同一个 tim_tix 输入。
- 这两个 ICx 信号在边沿处有效，但极性相反。
- 选择两个 TIxFP 信号之一作为触发输入，并将从模式控制器配置为复位模式。

可通过以下步骤测量施加在 tim_ti1 上的 PWM 信号的周期和脉冲宽度：

1. 通过在 TIMx_TISEL 寄存器中使用 TI1SEL[3:0]位选择合适的 tim_tix_in[15:0]源（内部或外部）。
2. 选择 TIMx_CCR1 的有效输入：向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01（选择 tim_ti1）。
3. 选择 tim_ti1fp1 的有效极性（用于 TIMx_CCR1 中的捕获和计数器清零）：向 CC1P 位和 CC1NP 位写入“0”（上升沿有效）。
4. 选择 TIMx_CCR2 的有效输入：向 TIMx_CCMR1 寄存器中的 CC2S 写入 10（选择 tim_ti1）。
5. 选择 tim_ti1fp2 的有效极性（用于 TIMx_CCR2 中的捕获）：向 CC2P 位写入 1 和 CC2NP 位写入 “0”（下降沿有效）。
6. 选择有效触发输入：向 TIMx_SMCR 寄存器中的 TS 位写入 101（选择 tim_ti1fp1）。
7. 将从模式控制器配置为复位模式：向 TIMx_SMCR 寄存器中的 SMS 位写入 100。
8. 使能捕获：向 TIMx_CCER 寄存器中的 CC1E 位和 CC2E 位写入“1”。



PWM 输入模式只能与 TIMx_CH1/TIMx_CH2 信号一起使用，因为只有 tim_ti1fp1 和 tim_ti2fp2 连接到从模式控制器。

图 14.30 PWM 输入模式时序

14.4.9 强制输出模式

在输出模式（TIMx_CCMRx 寄存器中的 CCxS 位 = 00）下，可直接由软件将每个输出比较信号（tim_ocxref 和 tim_ocx）强制设置为有效电平或无效电平，而无需考虑输出比较寄存器和计数器之间的任何比较结果。

要将输出比较信号（tim_ocxref/tim_ocx）强制设置为有效电平，只需向相应 TIMx_CCMRx 寄存器中的 OCxM 位写入 101。tim_ocxref 进而强制设置为高电平（tim_ocxref 始终为高电平有效），同时 tim_ocx 获取 CCxP 极性位的相反值。

例如：CCxP=0（tim_ocx 高电平有效）=> tim_ocx 强制设置为高电平。

通过向 TIMx_CCMRx 寄存器中的 OCxM 位写入 100，可将 tim_ocxref 信号强制设置为低电平。

无论如何，TIMx_CCRx 影子寄存器与计数器之间的比较仍会执行，而且允许将标志置 1。因此可发送相应的中断请求。输出比较模式一节对此进行了介绍。

14.4.10 输出比较模式

此功能用于控制输出波形，或指示已经过某一时间段。

当捕获/比较寄存器与计数器之间相匹配时，输出比较功能：

- 将为相应的输出引脚分配一个可编程值，该值由输出比较模式（TIMx_CCMRx 寄存器中的 OCxM 位）和输出极性（TIMx_CCER 寄存器中的 CCxP 位）定义。匹配时，输出引脚既可保持其电平（OCXM=000），也可设置为有效电平（OCXM=001）、无效电平（OCXM=010）或进行翻转（OCxM=011）。
- 将中断状态寄存器中的标志置 1（TIMx_SR 寄存器中的 CCxIF 位）。
- 如果相应中断使能位（TIMx_DIER 寄存器中的 CCXIE 位）置 1，将生成中断。

使用 TIMx_CCMRx 寄存器中的 OCxPE 位，可将 TIMx_CCRx 寄存器配置为带或不带预装载寄存器。

在输出比较模式下，更新事件 UEV 对 tim_ocxref 和 tim_ocx 输出毫无影响。同步的精度可以达到计

数器的一个计数周期。输出比较模式也可用于输出单脉冲（在单脉冲模式下）。

步骤：

1. 选择计数器时钟（内部、外部、预分频器）。
2. 在 TIMx_ARR 和 TIMx_CCRx 寄存器中写入所需数据。
3. 如果要生成中断请求，将 CCxIE 位和/或 CCxDE 位置 1。
4. 选择输出模式。例如，
   a) 写入 OCxM=0011 以在 CNT 匹配 CCRx 时翻转 tim_ocx 输出引脚
   b) 写入 OCxPE=0 以禁用预装载寄存器
   c) 写入 CCxP=0 以选择高电平有效
   d) 写入 CCxE=1 以使能输出
5. 通过将 TIMx_CR1 寄存器中的 CEN 位置 1 来使能计数器。

可随时通过软件更新 TIMx_CCRx 寄存器以控制输出波形，前提是未使能预装载寄存器（OCxPE=0，否则仅当发生下一个更新事件 UEV 时，才会更新 TIMx_CCRx 影子寄存器）。

下图列出了相关示例。

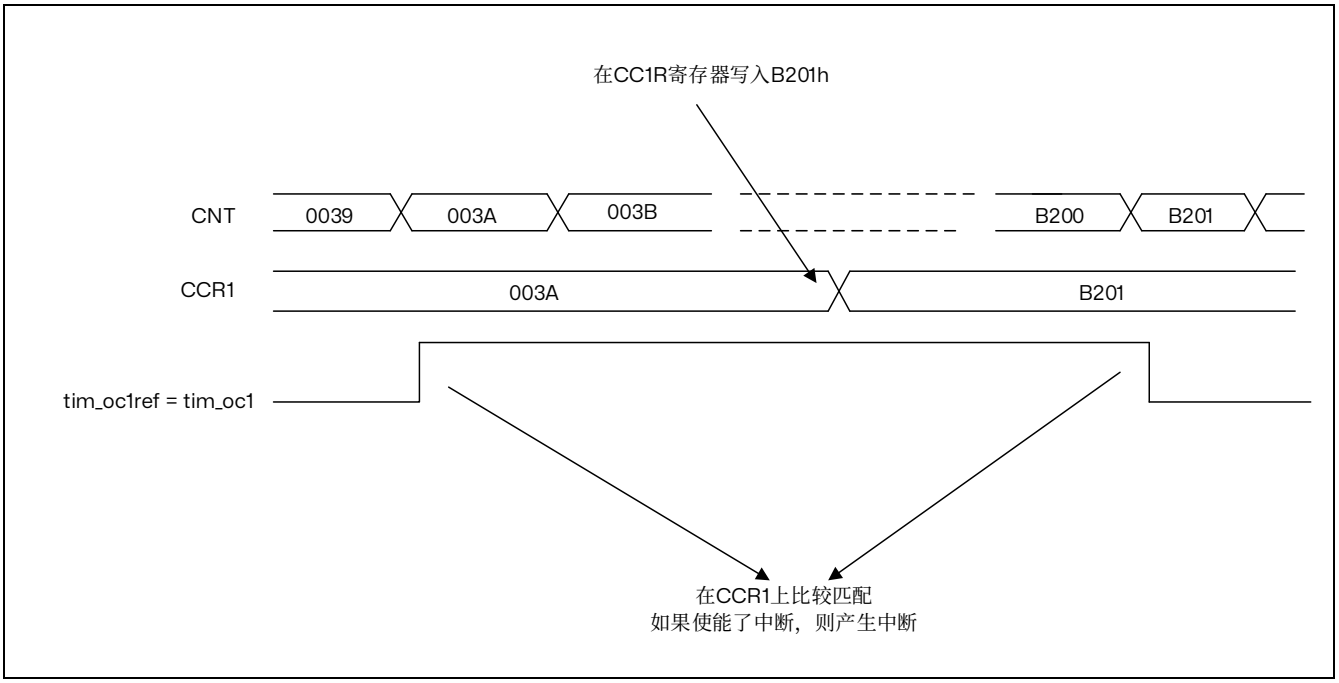


图 14.31 输出比较模式，翻转 tim_oc1

14.4.11 PWM 模式

脉冲宽度调制模式可以生成一个信号，该信号频率由 TIMx_ARR 寄存器值决定，其占空比则由 TIMx_CCRx 寄存器值决定。

通过向 TIMx_CCMRx 寄存器中的 OCxM 位写入 110（PWM 模式 1）或 111（PWM 模式 2），可以独立选择各通道（每个 tim_ocx 输出对应一个 PWM）的 PWM 模式。必须通过将 TIMx_CCMRx 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器，最后通过将 TIMx_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器。

由于只有在发生更新事件时预装载寄存器才会传送到影子寄存器，因此启动计数器之前，必须通过将 TIMx_EGR 寄存器中的 UG 位置 1 来初始化所有寄存器。

tim_ocx 极性可使用 TIMx_CCER 寄存器的 CCxP 位来编程。既可以设为高电平有效，也可以设为低电平有效。tim_ocx 输出通过将 TIMx_CCER 寄存器中的 CCxE 位置 1 来使能。有关详细信息，请参见 TIMx_CCERx 寄存器说明。

在 PWM 模式（1 或 2）下，TIMx_CNT 始终与 TIMx_CCRx 进行比较，以确定是 TIMx_CCRx ≤ TIMx_CNT 还是 TIMx_CNT ≤ TIMx_CCRx（取决于计数器计数方向）。外部事件可以通过 tim_etr_in 或 tim_oceref_clr 信号。在本例中仅声明了 tim_ocref_clr 信号

- 比较匹配事件之后
- 输出比较模式（TIMx_CCMRx 寄存器中的 OCxM 位）从“冻结”配置（不进行比较，OCxM=“000”）切换为任一 PWM 模式（OCxM=“110”或“111”）。定时器运行期间，可以通过软件强制 PWM 输出。

根据 TIMx_CR1 寄存器中的 CMS 位状态，定时器能够产生边沿对齐模式或中央对齐模式的 PWM 信号。

PWM 边沿对齐模式

递增计数配置

当 TIMx_CR1 寄存器中的 DIR 位为低时执行递增计数。

以下以 PWM 模式 1 为例。只要 TIMx_CNT < TIMx_CCRx，PWM 参考信号 tim_ocxref 便为高电平，否则为低电平。如果 TIMx_CCRx 中的比较值大于自动重载值（TIMx_ARR 中），则 tim_ocxref 保持为“1”。如果比较值为 0，则 tim_ocxref 保持为“0”。下图举例介绍边沿对齐模式的一些 PWM 波形 (TIMx_ARR=8)。

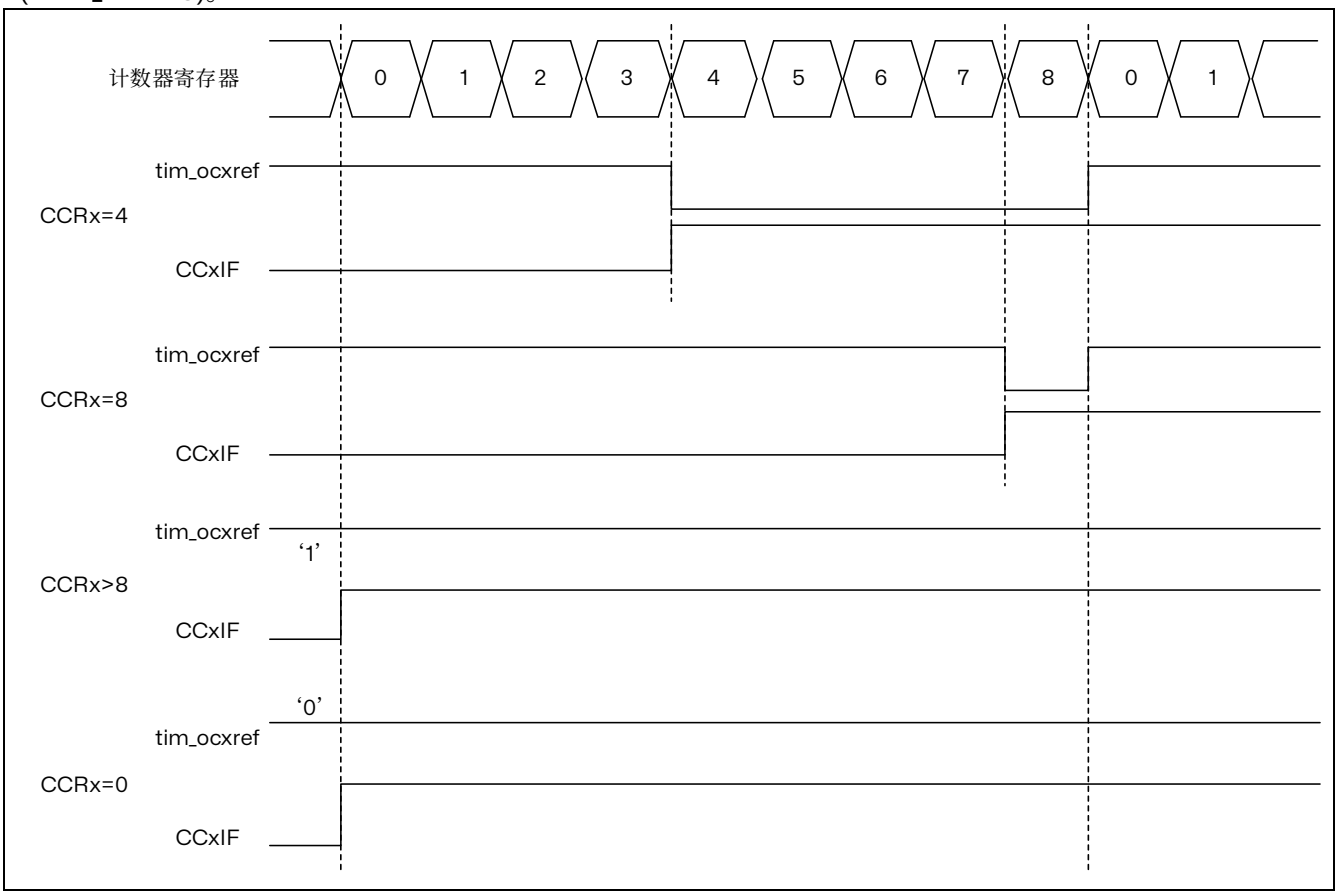


图 14.32 边沿对齐的 PWM 波形（ARR=8）

递减计数配置

当 TIMx_CR1 寄存器中的 DIR 位为高时执行递减计数。

在 PWM 模式 1 下，只要 TIMx_CNT>TIMx_CCRx，参考信号 tim_ocxref 便为低电平，否则为高电平。如果 TIMx_CCRx 中的比较值大于 TIMx_ARR 中的自动重载值，则 tim_ocxref 保持为“1”。此模式下不可能产生 0%的 PWM 波形。

PWM 中央对齐模式

当 TIMx_CR1 寄存器中的 CMS 位不为“00”时（其余所有配置对 tim_ocxref/tim_ocx 信号具有相同的作用），中央对齐模式生效。根据 CMS 位的配置，可以在计数器递增计数、递减计数或同时递增和递减计数时将比较标志置 1。TIMx_CR1 寄存器中的方向位（DIR）由硬件更新，不得通过软件更改。

下图显示了中央对齐模式的 PWM 波形，在此例中：

- TIMx_ARR=8
- PWM 模式为 PWM 模式 1
- 在根据 TIMx_CR1 寄存器中 CMS=01 而选择的中央对齐模式 1 下，当计数器递减计数时，比较标志置 1。

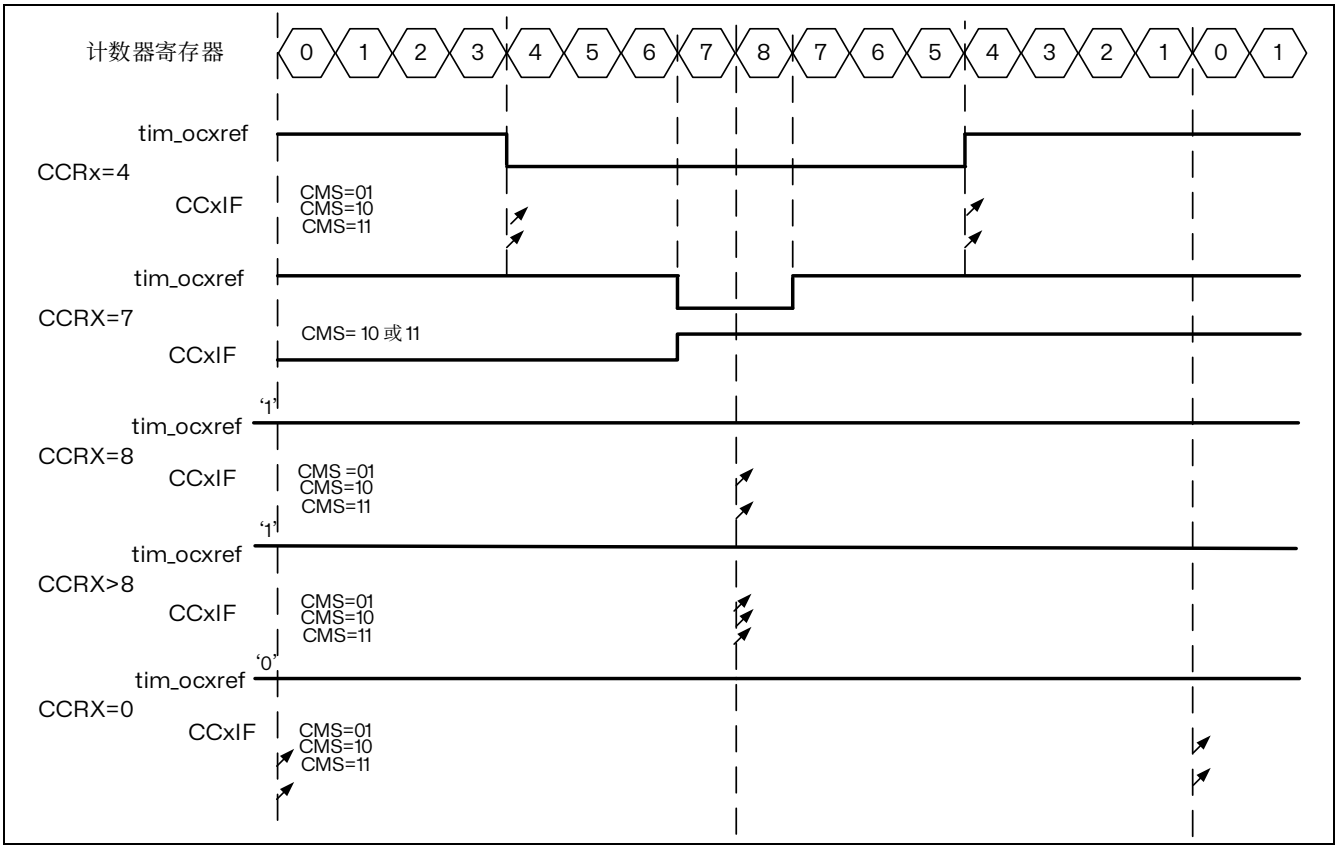


图 14.33 中央对齐的 PWM 波形（APR=8）

中央对齐模式使用建议：

- 启动中央对齐模式时将使用当前的递增/递减计数配置。这意味着计数器将根据写入 TIMx_CR1 寄存器中 DIR 位的值进行递增或递减计数。此外，不得同时通过软件修改 DIR 和 CMS 位。
- 不建议在运行中央对齐模式时对计数器执行写操作，否则将发生意想不到的结果。尤其是：

— 如果写入计数器中的值大于自动重载值（TIMx_CNT>TIMx_ARR），计数方向不会更新。例如，如果计数器之前递增计数，则继续递增计数。

— 如果向计数器写入 0 或 TIMx_ARR 的值，计数方向会更新，但不生成更新事件 UEV。

- 使用中央对齐模式最为保险的方法是：在启动计数器前通过软件生成更新（将 TIMx_EGR 寄存器中的 UG 位置 1），并且不要在计数器运行过程中对其执行写操作。

抖动模式

通过使能抖动模式，使用 TIMx_CR1 寄存器中的 DITHEN 位，可以提高 PWM 模式的有效分辨率。这既适用于 CCR（用于增加占空比分辨率），也适用于 ARR（用于增加 PWM 频率分辨率）。

工作原理是在 16 个连续的 PWM 周期中，以预定义的模式使实际的 CCR（或 ARR）值略有变化（增加或减少一个定时器时钟周期）。这使得占空比或 PWM 期间的平均值提高了 16 倍分辨率。下图展示了应用于 4 个连续 PWM 周期的抖动原理。

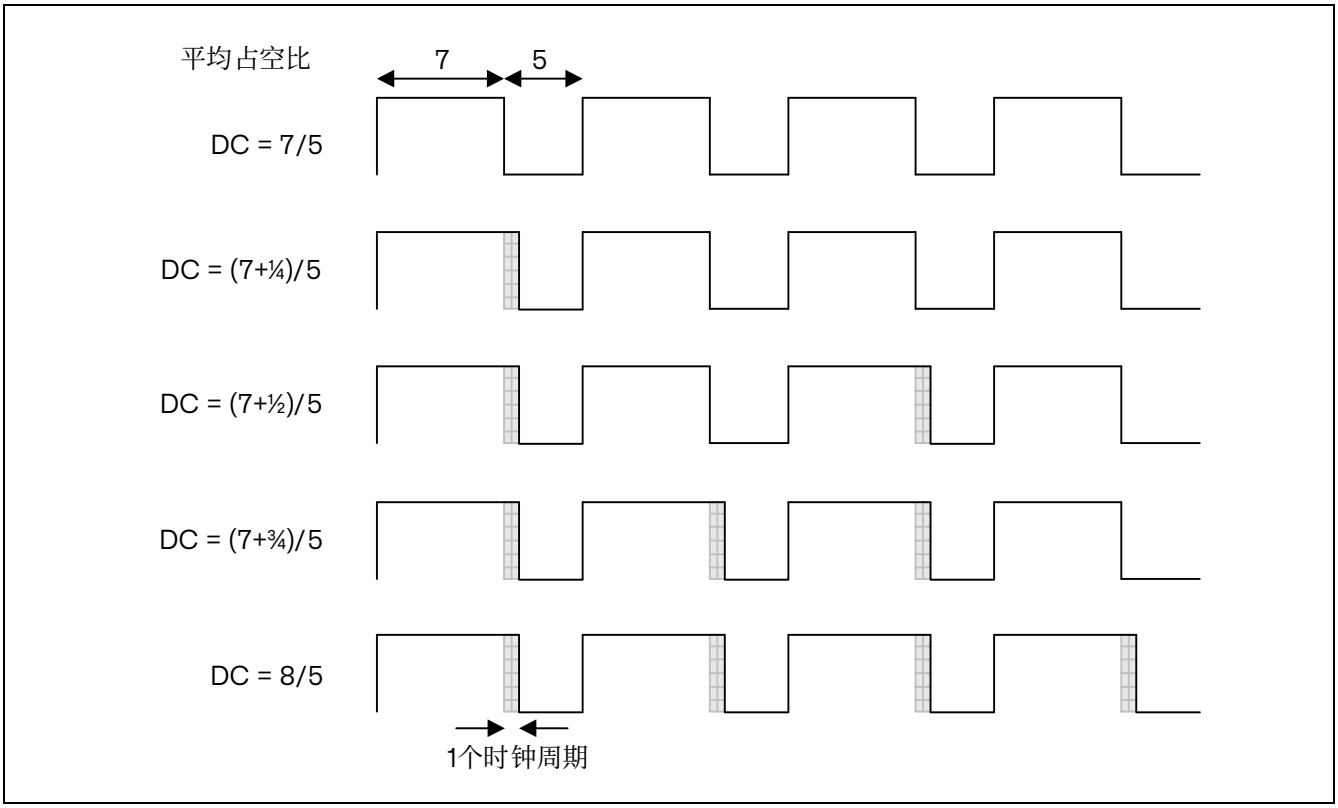


图 14.34 抖动原理

当抖动模式被使能时，寄存器编码将如下所示进行变化（参见下图示例）

- 4 个最低有效位（LSB）用于编码增强的分辨率部分（小数部分）
- 最高有效位（MSB）左移至 19:4 位，用于编码基本值

注意：如果设置或清除 DITHEN *位，* ARR *和* CCR *值将自动更新。（例如，如果* ARR=0x05 *且* DITHEN=0，*它将更新为* ARR=0x50 *且* DITHEN=1*）。*

1. 必须重置 CEN 和 ARPE 位
2. ARR[3:0]位必须重置
3. DITHEN 位必须重置
4. CCIF 标志必须清除
5. 可以设置 CEN 位（最终设置 ARPE=1）。

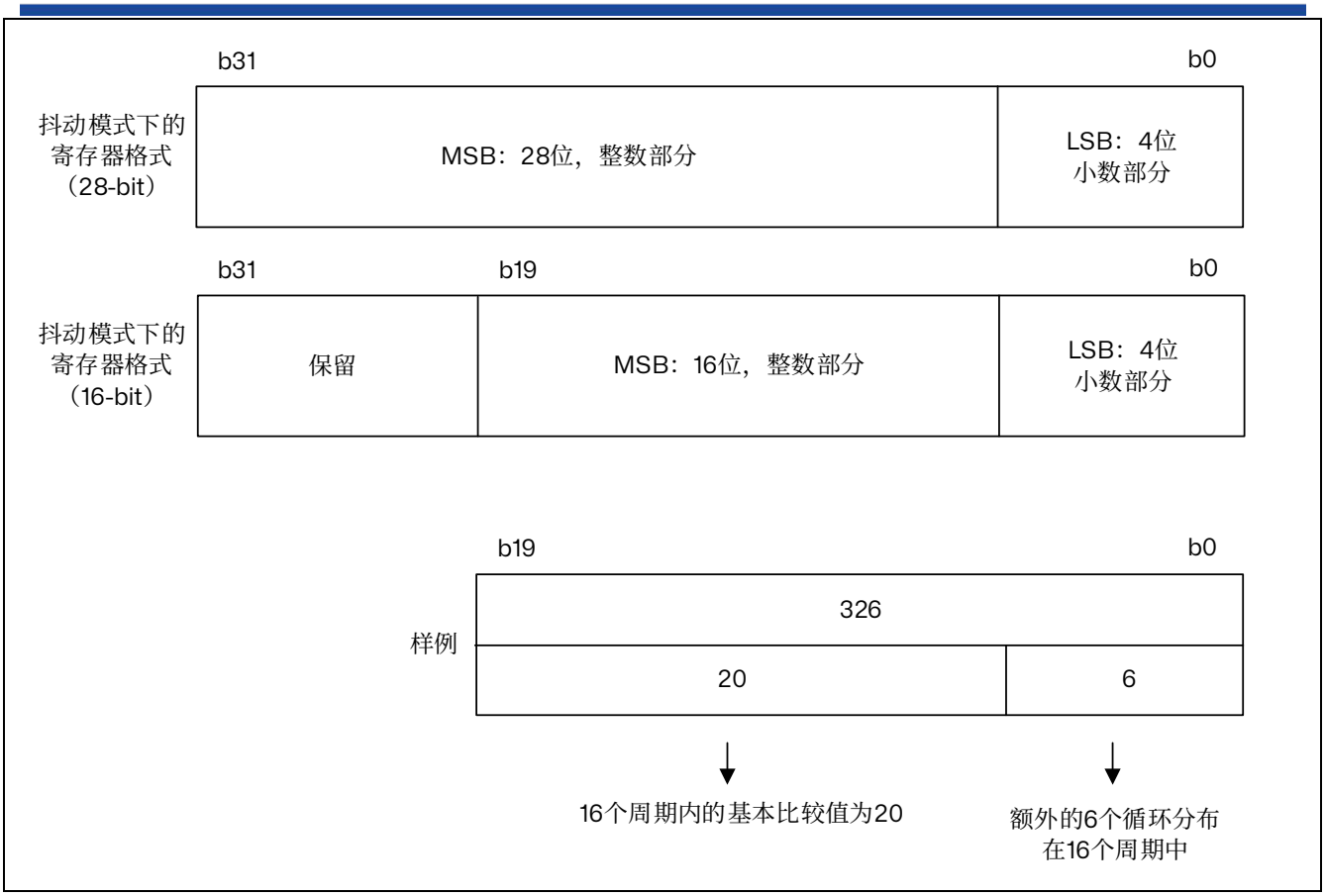


图 14.35 抖动模式下的数据格式和寄存器编码

最小频率由以下公式给出：

$$\text{分辨率} = \frac{\text{FTim}}{\text{Fpwm}} \Rightarrow \text{FpwmMin} = \frac{\text{FTim}}{\text{最大分辨率}}$$

$$\text{禁用抖动模式：FpwmMin} = \frac{\text{FTim}}{65536}$$

$$\text{使能抖动模式（16 位定时器）：FpwmMin} = \frac{\text{FTim}}{65535 + \frac{15}{16}}$$

$$\text{使能抖动模式（28 位定时器）：FpwmMin} = \frac{\text{FTim}}{268435454 + \frac{15}{16}}$$

*注意：对于16 位定时器，在抖动模式下，*TIMx_ARR *和*TIMx_CCRy *的最大值限制为*0xFFFEF *（对应于整数部分的*65534 *和抖动部分的*15*）。*

*对于28 位定时器，在抖动模式下，*TIMx_ARR *和*TIMx_CCRy *的最大值限制为*0xFFFFFFEF *（对应于整数部分的*268435454 *和抖动部分的*15*）。*

如下两张图所示，抖动模式可以增加 PWM 分辨率。

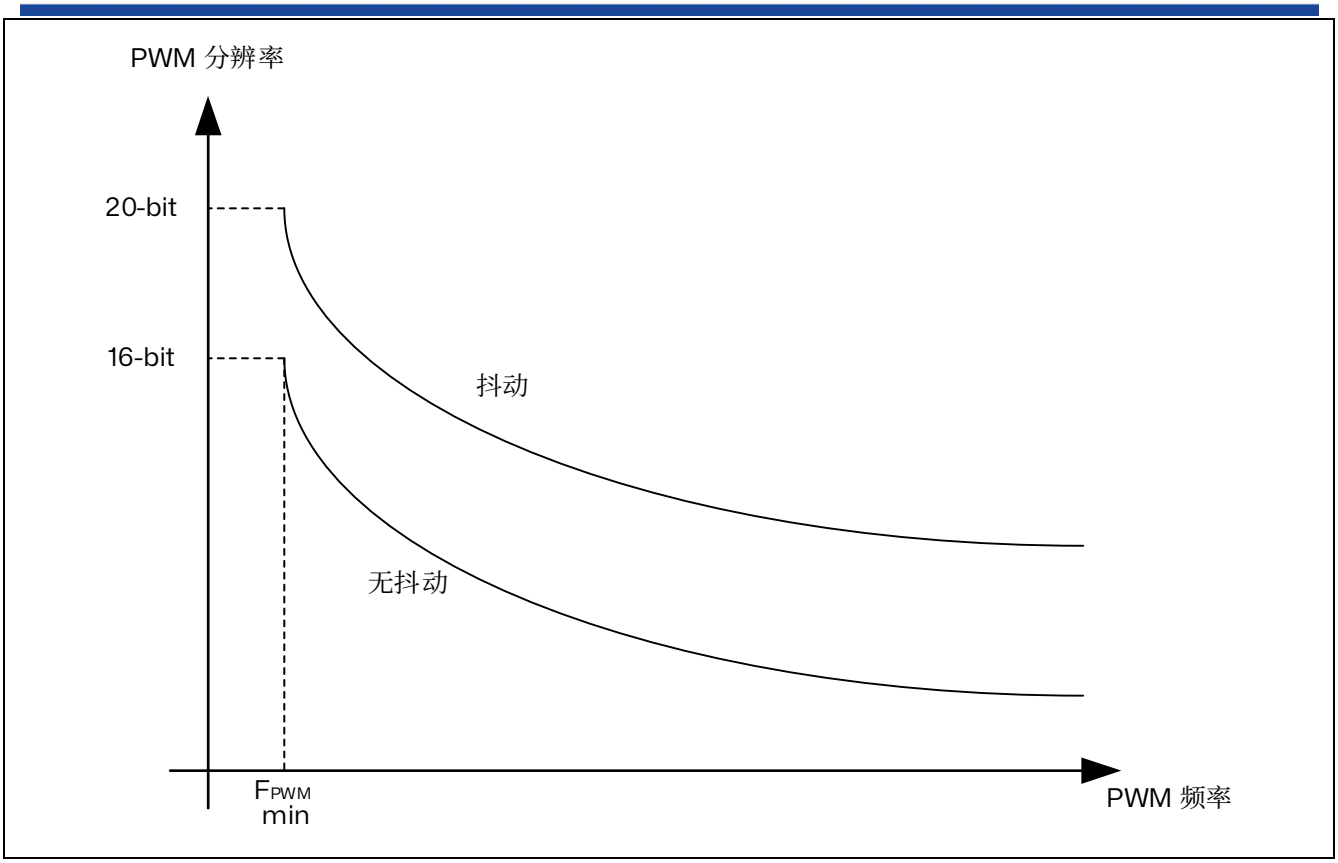


图 14.36 PWM 分辨率 vs 频率（16 位模式）

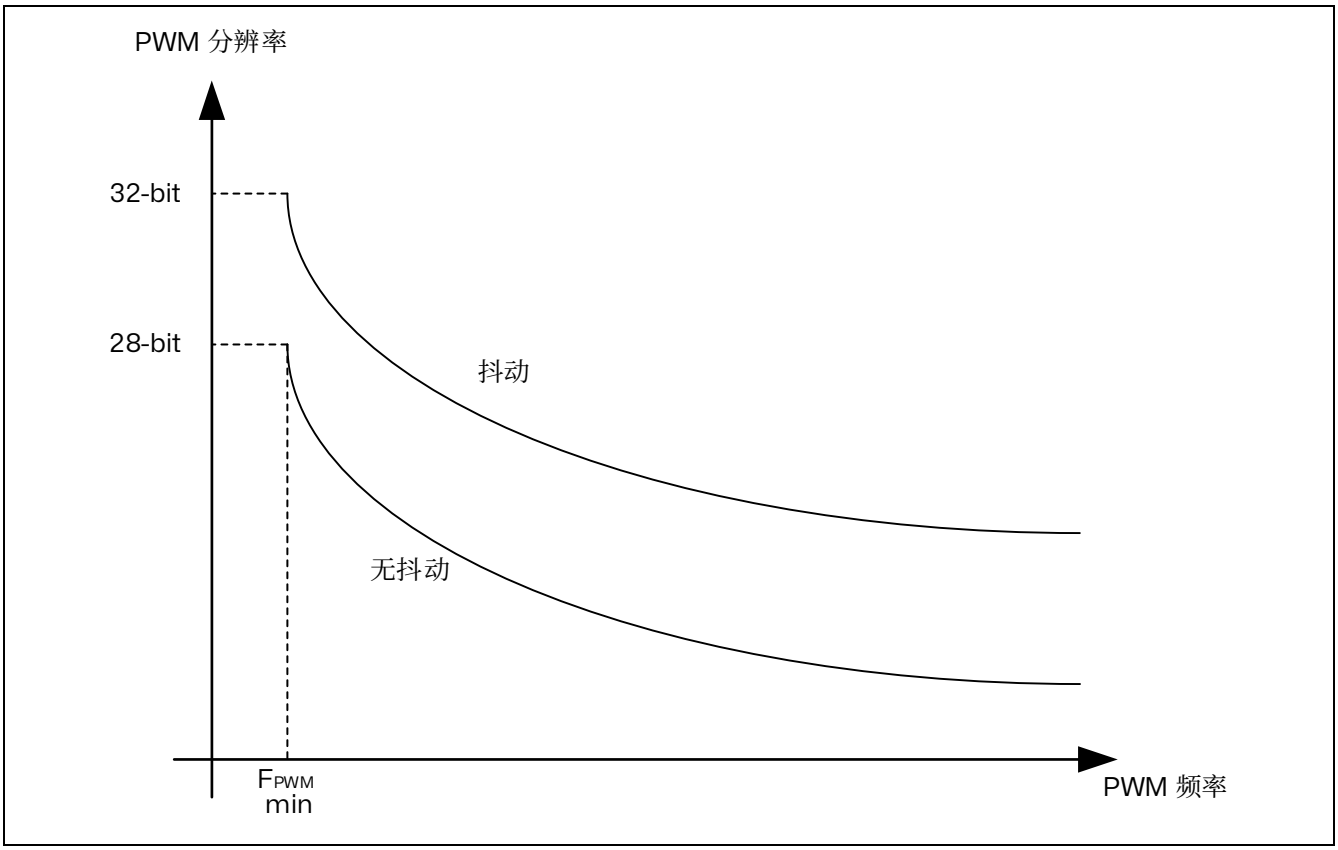


图 14.37 PWM 分辨率 vs 频率（28 位模式）

如下图所示，占空比和/或周期的变化分散在 16 个连续周期。

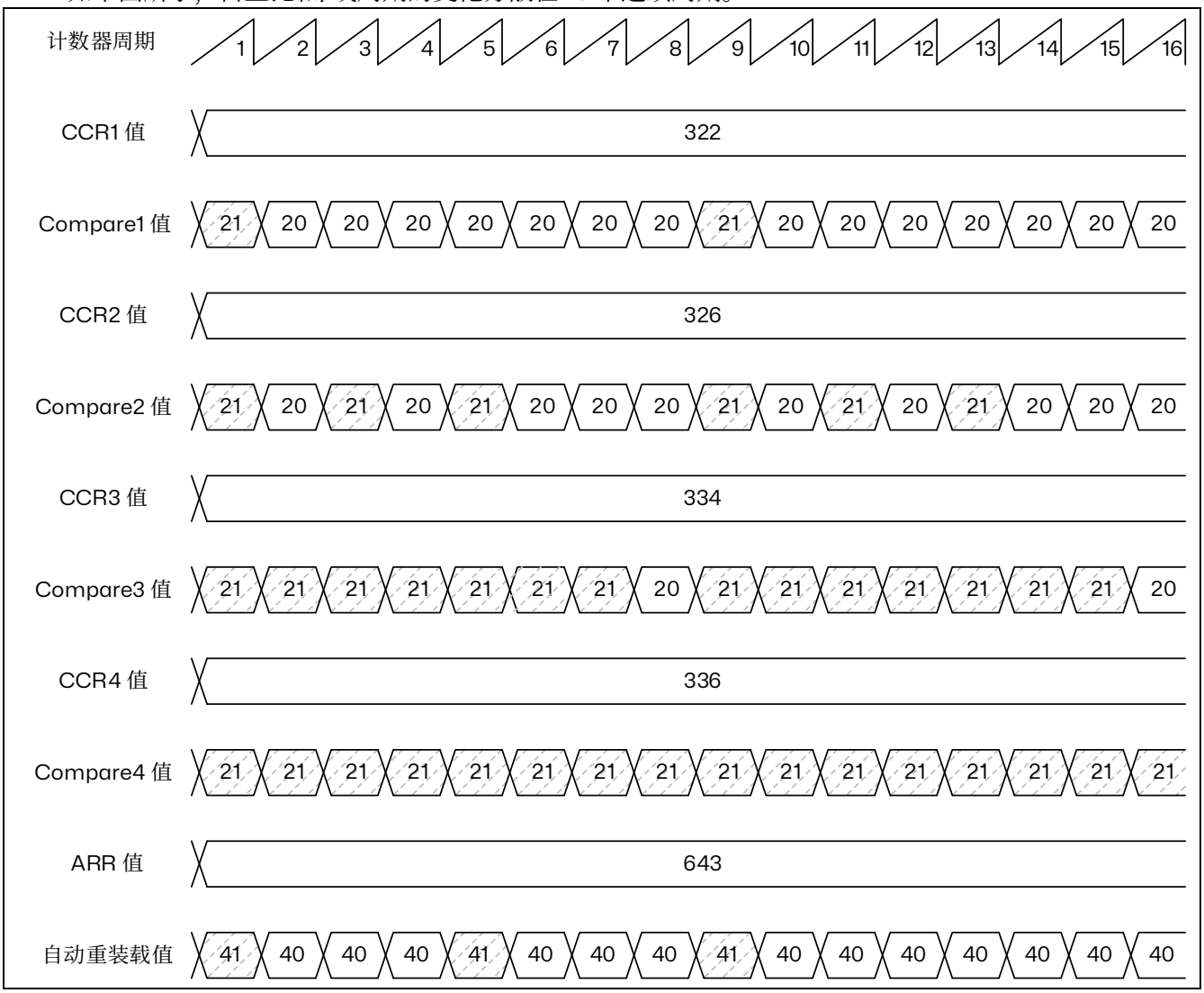


图 14.38 PWM 抖动模式

自动重新加载和比较值的增量按照下表中描述的特定模式分布。抖动序列是为了使增量分布尽可能均匀并最大限度地减少整体波动。

表 14.11 CCR 和 ARR 寄存器的变化抖动模式

LSB 值	PWM 周期															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0001	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0010	+1	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-
0011	+1	-	-	-	+1	-	-	-	+1	-	-	-	-	-	-	-
0100	+1	-	-	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0101	+1	-	+1	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0110	+1	-	+1	-	+1	-	-	-	+1	-	+1	-	+1	-	-	-
0111	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	-	-
1000	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1001	+1	+1	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1010	+1	+1	+1	-	+1	-	+1	-	+1	+1	+1	-	+1	-	+1	-
1011	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	-	+1	-
1100	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1101	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1110	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	+1	+1	+1	+1	-
1111	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	-

在中央对齐 PWM 模式下（TIMx_CR1 寄存器中的 CMS 位不为“00”），也提供了抖动模式。在这种模式下，抖动模式考虑到递增递减计数阶段，应用于 8 个连续的 PWM 周期，如下图所示。

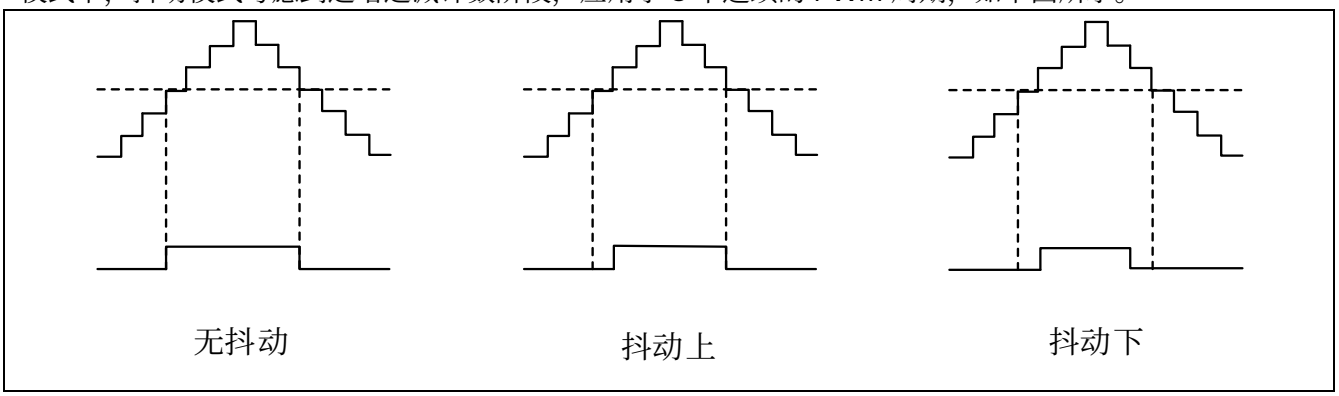


图 14.39 中央对齐 PWM 模式下抖动对占空比的影响

下表显示了如何在中央对齐 PWM 模式下添加抖动模式。

表 14.12 CCR 寄存器在中央对齐 PWM 模式下的变化抖动模式

LSB 值	PWM 周期															
	1		2		3		4		5		6		7		8	
	上	下	上	下	上	下	上	下	上	下	上	下	上	下	上	下
0000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0001	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0010	+1	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-
0011	+1	-	-	-	+1	-	-	-	+1	-	-	-	-	-	-	-
0100	+1	-	-	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0101	+1	-	+1	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0110	+1	-	+1	-	+1	-	-	-	+1	-	+1	-	+1	-	-	-
0111	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	-	-
1000	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1001	+1	+1	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1010	+1	+1	+1	-	+1	-	+1	-	+1	+1	+1	-	+1	-	+1	-
1011	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	-	+1	-
1100	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1101	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1110	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	+1	+1	+1	+1	-
1111	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	-

14.4.12 非对称 PWM 模式

非对称 PWM 模式 1/非对称 PWM 模式 2

非对称模式允许生成两个中央对齐的 PWM 信号，并具有可编程的相位偏移。虽然频率由 TIMx_ARR 寄存器的值决定，但占空比和相位偏移由一对 TIMx_CCRx 寄存器确定。一个寄存器控制 PWM 在计数上升期间，第二个寄存器在计数下降期间，这样 PWM 每半个 PWM 周期调整一次：

— tim_oc1refc（或 tim_oc2refc）由 TIMx_CCR1 和 TIMx_CCR2 控制

— tim_oc3refc（或 tim_oc4refc）由 TIMx_CCR3 和 TIMx_CCR4 控制

通过将“1110”（非对称 PWM 模式 1）或“1111”（非对称 PWM 模式 2）写入 TIMx_CCMRx 寄存器的 OCxM 位，可以在两个通道上独立选择非对称 PWM 模式（每个 CCR 寄存器对一个 tim_ocx 输出）。

*注意：为了兼容性原因，*OCxM[3:0]*位字段被分为两个部分，最高有效位与最低*3 *位不连续。*

当某个通道用作非对称 PWM 通道时，其互补通道也可以使用。例如，如果 tim_oc1refc 信号在通道 1 上生产（非对称 PWM 模式 1），可以在通道 2 上输出 tim_oc1refc 信号，或者产生非对称 PWM 模式 1 的结果 tim_oc2refc 信号。

下图标识使用非对称 PWM 模式生成信号的示例（通道 1 到 4 配置为非对称 PWM 模式 2）。

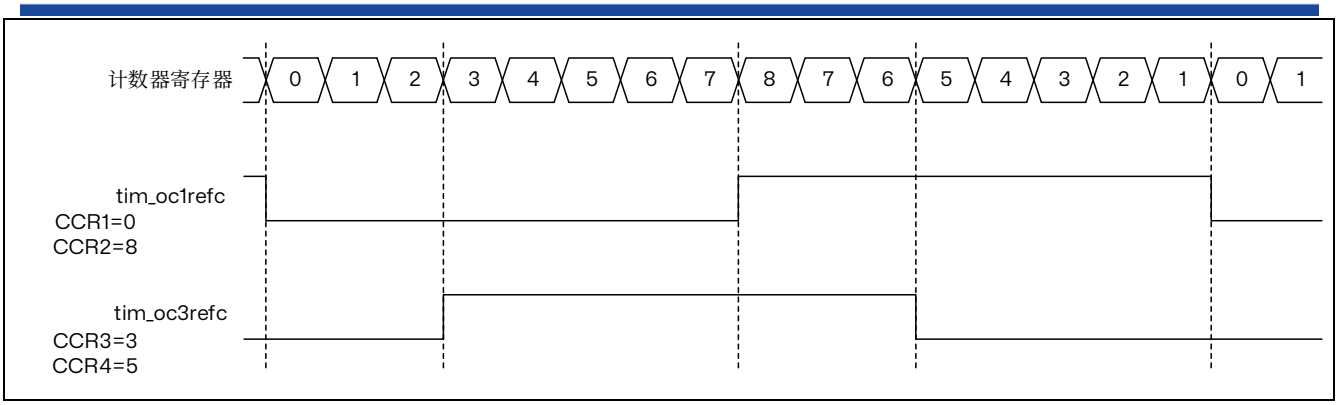


图 14.40 生成两个具有 50%占空比的相位偏移 PWM 信号

PWM 边沿对齐模式

- 向上计数配置

当 TIMx_CR1 寄存器中的 DIR 位为低的时候执行向上计数。当 TIMx_CR1 寄存器中的 CCRxN 位为‘1’时配置位非对称模式。下面是一个 PWM 模式 1 的例子。

当 TIMx_CCRx<=TIMx_CNT<TIMx_CCRxN 时，PWM 参考信号 OCxREF 为高，否则为低。如果 TIMx_CCRx 和 TIMx_CCRxN 中的比较值大于自动重装载值（TIMx_ARR），则 OCxREF 保持为‘1’。如果 TIMx_CCRx 中的比较值大于等于 TIMx_CCRxN 中的比较值或者比较值都为 0，则 OCxREF 保持为‘0’。下图为 TIMx_ARR=12 时边沿对齐的 PWM 波形实例。

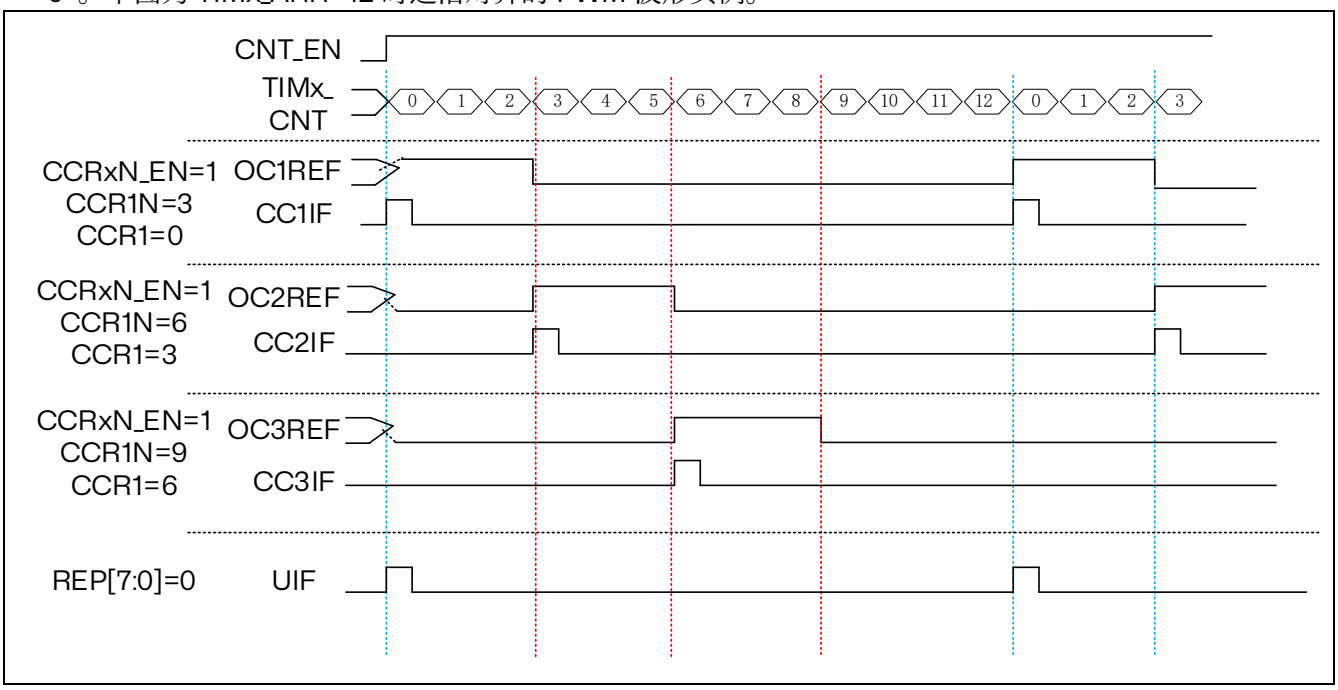


图 14.41 边沿对齐向上计数的 PWM 波形（ARR=12）

- 向下计数的配置

当 TIMx_CR1 寄存器中的 DIR 位为低的时候执行向上计数。当 TIMx_CR1 寄存器中的 CCRxN 位为‘1’时配置位非对称模式。下面是一个 PWM 模式 1 的例子。当 TIMx_CCRxN<=TIMx_CNT<TIMx_CCRx 时，PWM 参考信号 OCxREF 为高，否则为低。如果 TIMx_CCRx 和 TIMx_CCRxN 中的比较值大于自动重装载值（TIMx_ARR），则 OCxREF 保持为‘1’。如

果 TIMx_CCRx 中的比较值小于等于 TIMx_CCRxN 中的比较值或者比较值都为 0，则 OCxREF 保持为‘0’。

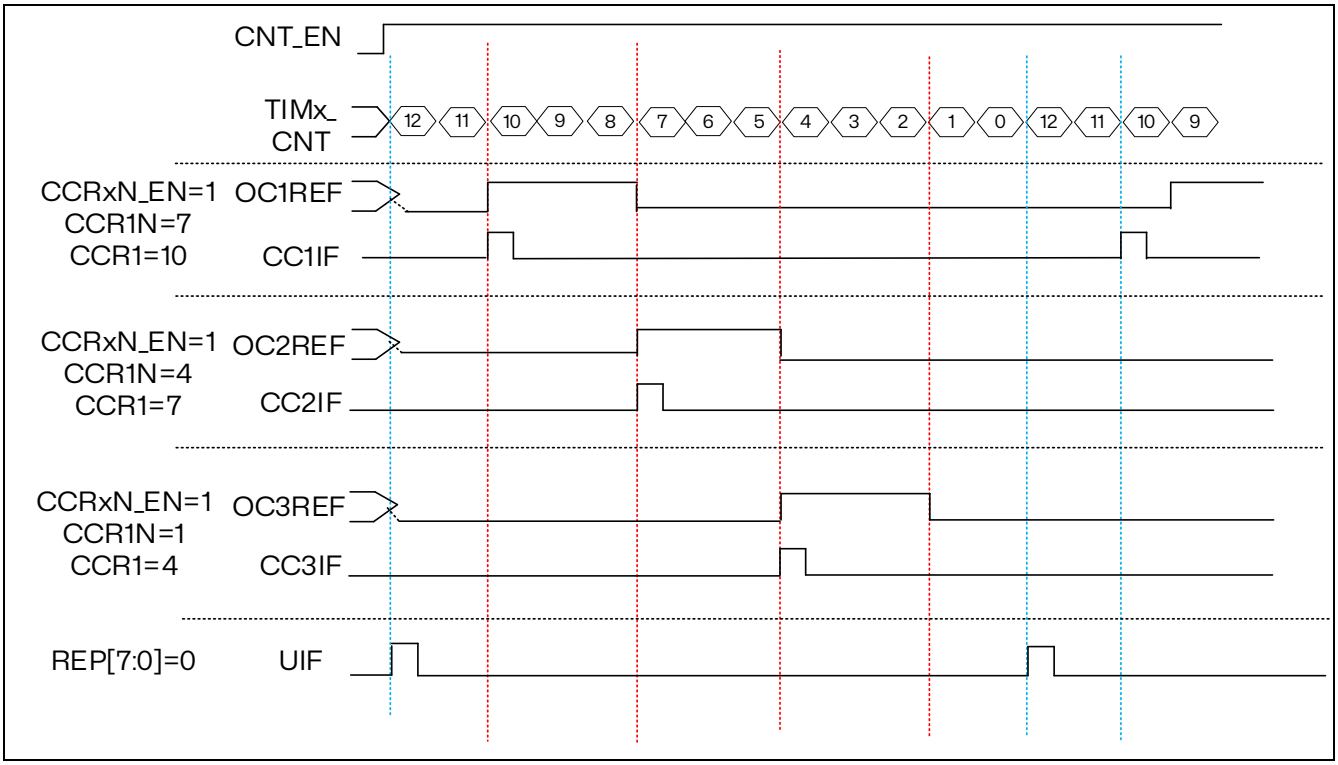


图 14.42 边沿对齐向下计数的 PWM 波形（ARR=12）

PWM 中央对齐模式

当 TIMx_CR1 寄存器中的 CMS 位不为‘00’时为中央对齐模式（所有其他的配置对 OCxREF/OCx 信号都有相同的作用）。根据不同的 CMS 位设置，比较标志可以在计数器向上计数时被置 1、在计数器向下计数时被置 1、或在计数器向上和向下计数时被置 1。TIMx_CR1 寄存器中的计数方向位（DIR）由硬件更新，不要用软件修改它。下图给出了一些中央对齐的 PWM 波形的例子

- TIMx_ARR=8
- PWM 模式 1
- TIMx_CR1 寄存器的 CMS=01，在中央对齐模式 1 下，当计数器向下计数时设置比较标志。

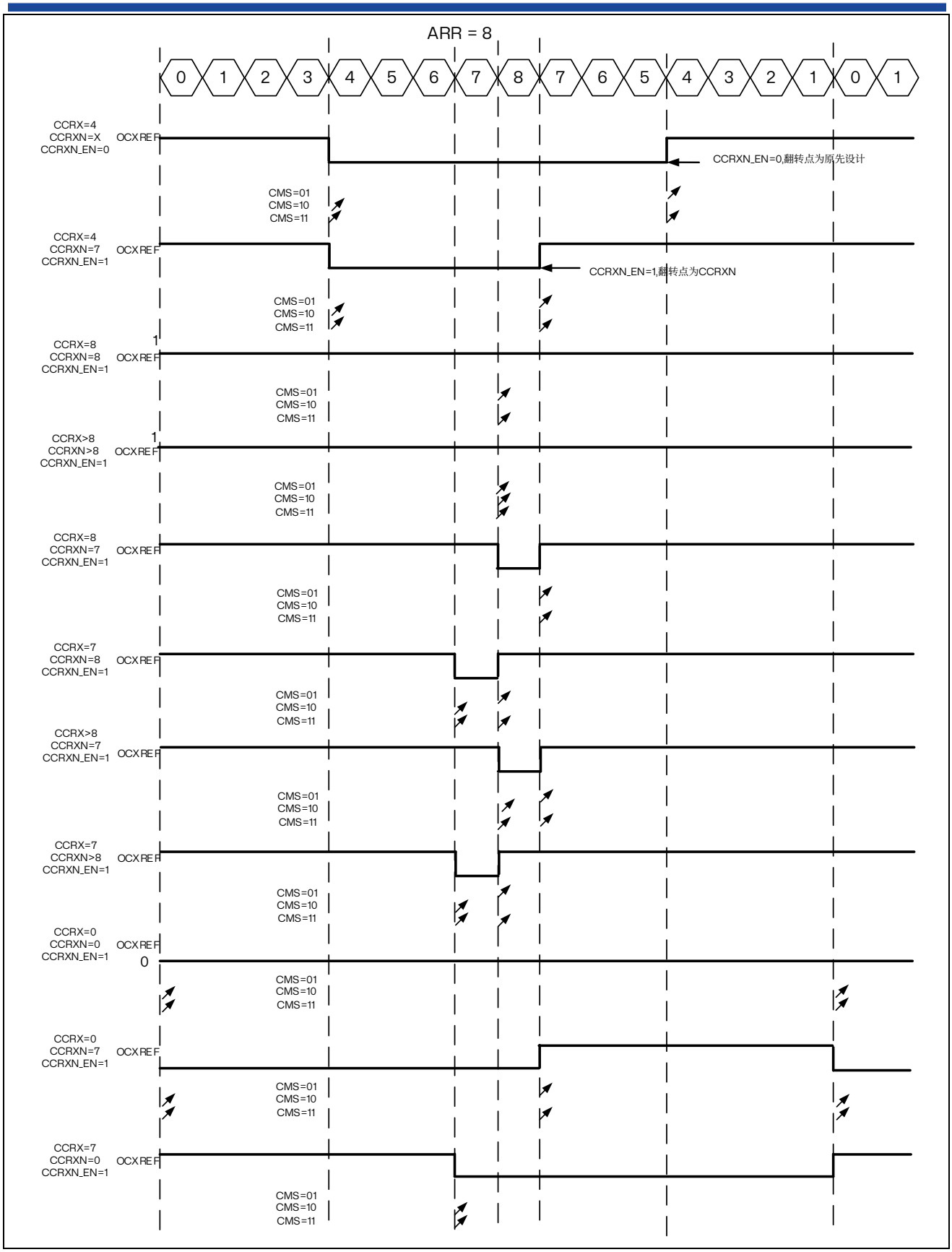


图 14.43 中央对齐的 PWM 波形（ARR=8）

14.4.13 组合 PWM 模式

组合 PWM 模式允许生成两个边缘对齐或中央对齐的 PWM 信号，各自脉冲之间的可编程延迟和相位偏移。虽然频率由 TIMx_ARR 寄存器的值决定，但占空比和延迟由两个 TIMx_CCRx 寄存器确定。结果信号 tim_ocxrefc 由两个参考 PWM 的 OR 或 AND 逻辑组合而成：

— tim_oc1refc（或 tim_oc2refc）由 TIMx_CCR1 和 TIMx_CCR2 控制

— tim_oc3refc（或 tim_oc4refc）由 TIMx_CCR3 和 TIMx_CCR4 控制

通过在 TIMx_CCMRx 寄存器的 OCxM 位中写入“1100”（组合 PWM 模式 1）或“1101”（组合 PWM 模式 2），可以在两个通道上独立选组合 PWM 模式（每个 CCR 寄存器对一个 tim_ocx 输出）。

当某个通道用作组合 PWM 通道时，其互补通道必须配置为相反的 PWM 模式（例如，一个通道为组合 PWM 模式 1，另一个通道为组合 PWM 模式 2）。

*注意：为了兼容性原因，*OCxM[3:0]*位字段被分为两个部分，最高有效位与最低*3 *位不连续。*

下图表示使用组合 PWM 模式生成信号的示例，该示例的配置如下：

— 通道 1 配置为组合 PWM 模式 2。

— 通道 2 配置为 PWM 模式 1。

— 通道 3 配置为组合 PWM 模式 2。

— 通道 4 配置为 PWM 模式 1。

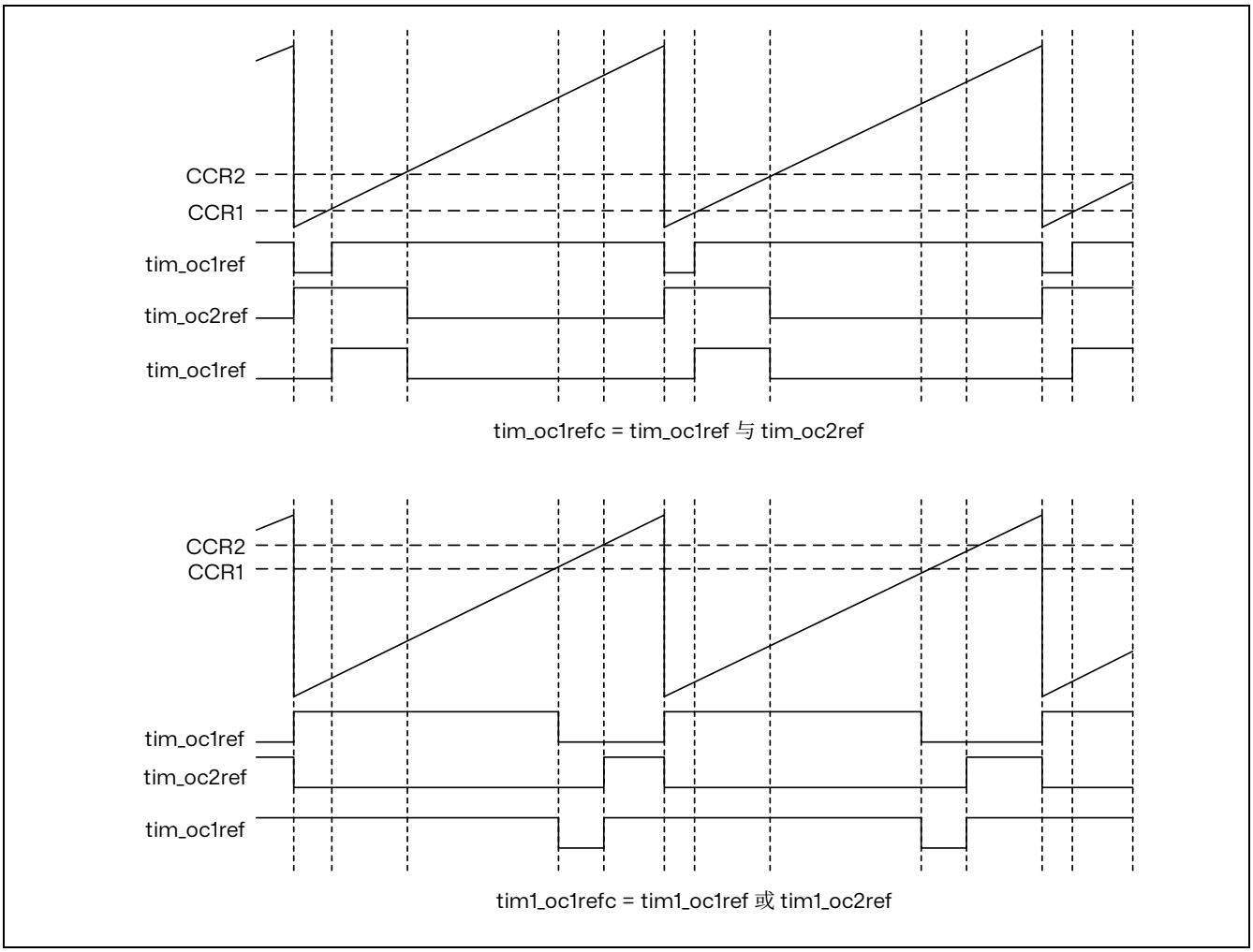


图 14.44 通道 1 和通道 3 的组合 PWM 模式

14.4.14 在外部事件时清除 tim_ocxref 信号

对于给定通道，在 tim_ocref_clr_int 输入施加高电平（相应 TIMx_CCMRx 寄存器中的 OCxCE 使能位置“1”），可使 tim_ocxref 信号变为低电平。tim_ocxref 信号将保持低电平，直到发生下一更新事件(UEV)。此功能仅能用于输出比较模式和 PWM 模式，而不适用于强制输出模式。

tim_ocref_clr_int 源取决于 OCREF 清除选择功能实施，请参考章节：TIM2/TIM3 实现。

如果实施了 OCREF 清除选择功能，则可以选择 tim_ocref_clr_int 在 tim_ocref_clr 输入和 tim_etrf 输入（滤波器后的 tim_etr_in）之间，通过以下方式进行在 TIMx_SMCR 寄存器中配置 OCCS 位。tim_ocref_clr 输入可以是使用中 OCRSEL[2:0]位字段，从多个 tim_ocref_clr[7:0]输入中选出 TIMx_AF2 寄存器，如下图所示。

当配置组合 PWM 模式，对于给定通道，在 tim_ocref_clr_int 输入施加高电平（相应 TIMx_CCMRx 寄存器中的 OCxCE 使能位置“1”），可使组合结果 tim_ocxrefc 信号变为低电平。tim_ocxrefc 信号将保持低电平，直到发生下一更新事件（UEV）。

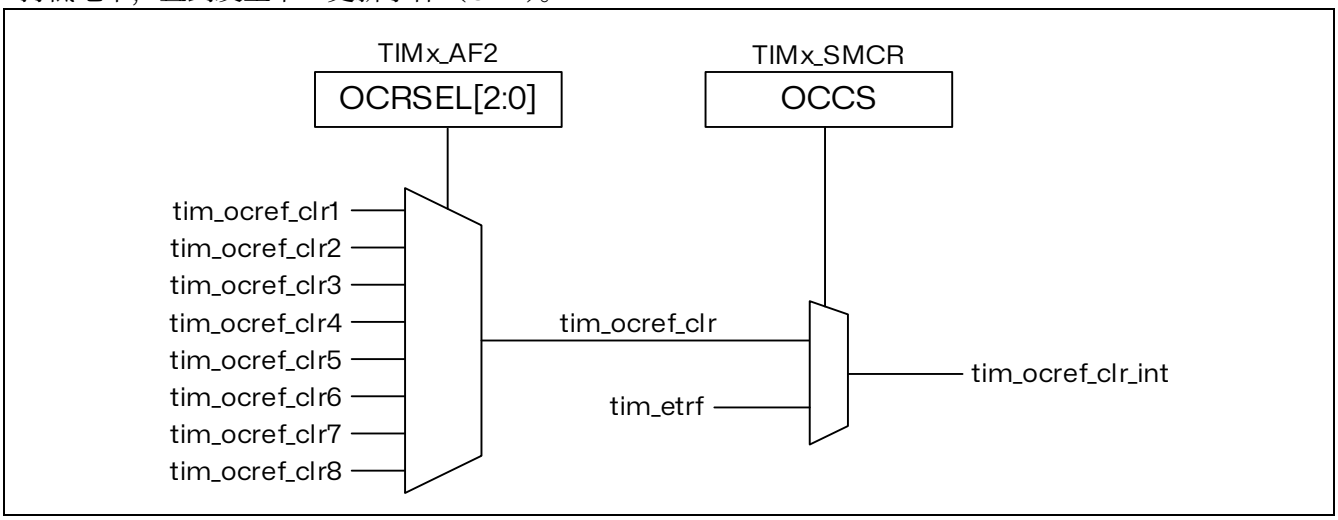


图 14.45 tim_ocref_clr 输入选择复用器

如果未实现 OCREF 清除选择功能，则 tim_ocref_clr_int 输入为直接连接到 tim_etrf 输入。

例如，tim_ocref_clr_int 信号可以连接到比较器的输出，用于控制电流。此时，tim_etr_in 必须如下配置：

1. 必须关闭外部触发预分频器：TIMx_SMCR 寄存器中的 ETPS[1:0]位置“00”。
2. 必须禁止外部时钟模式 2：TIMx_SMCR 寄存器中的 ECE 位置“0”。
3. 外部触发极性（ETP）和外部触发滤波器（ETF）可根据应用需要进行配置。

下图对比了使能位 OCxCE 在不同值下的情况，显示了当 tim_etrf 输入变为高电平时 tim_ocxref 信号的行为。在本例中，定时器 TIMx 编程为 PWM 模式。

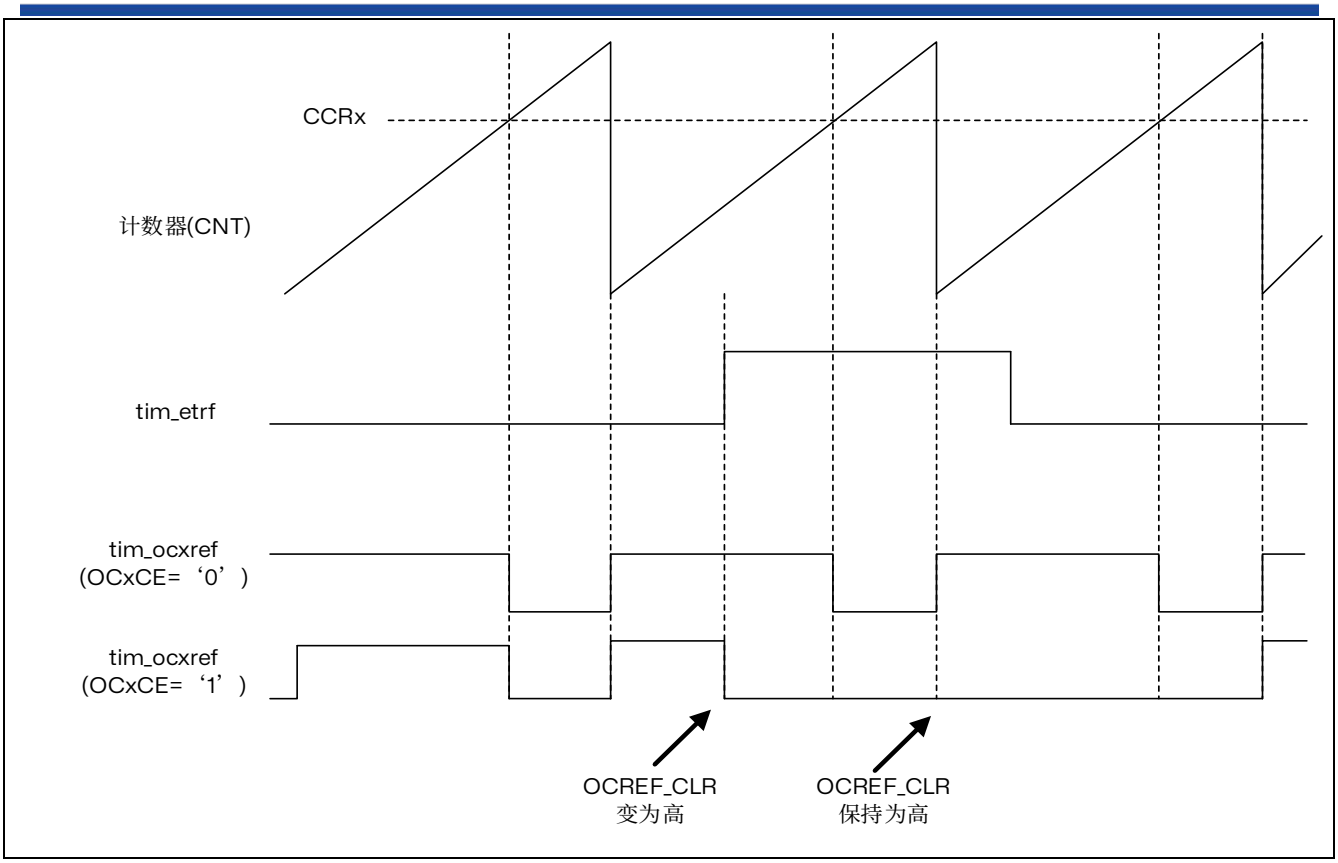


图 14.46 清除 TIMx 的 tim_ocxref

注：如果 PWM *的占空比为* 100% (CCRx>ARR)，*则下次计数器上溢时会再次使能* tim_ocxref。

14.4.15 单脉冲模式

单脉冲模式（OPM）是上述模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。将 TIMx_CR1 寄存器中的 OPM 位置 1，即可选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

只有当比较值与计数器初始值不同时，才能正确产生一个脉冲。启动前（定时器等待触发时），必须进行如下配置：

- CNT < CCRx ≤ ARR（特别注意，0<CCRx）

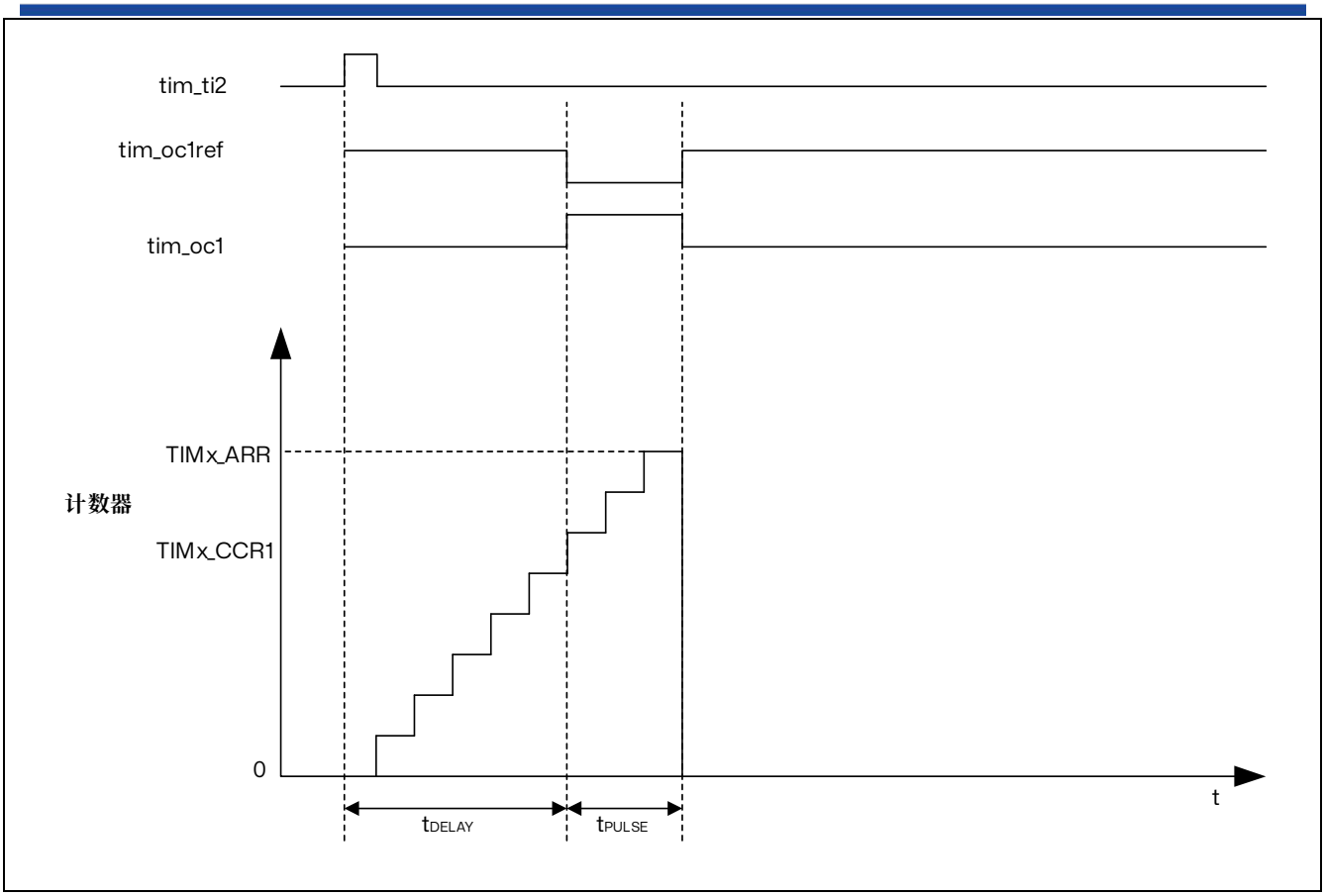


图 14.47 单脉冲模式的例子

例如，用户希望达到这样的效果：在 tim_ti2 输入引脚检测到正沿时，经过 t_{DELAY} 的延迟，在 tim_oc1 上产生一个长度为 t_{PULSE} 的正脉冲。

使用 tim_ti2fp2 作为触发 1:

3. 通过在 TIMx_TISEL 寄存器中使用 TI2SEL[3:0]位选择合适的 tim_ti2_in[15:0]源（内部或外部）。
4. 在 TIMx_CCMR1 寄存器中写入 CC2S=01，将 tim_ti2fp2 映射到 tim_ti2。
5. 在 TIMx_CCER 寄存器中写入 CC2P=0 和 CC2NP=“0”，使 tim_ti2fp2 能够检测上升沿。
6. 在 TIMx_SMCR 寄存器中写入 TS=00110，将 tim_ti2fp2 配置为从模式控制器的触发 (tim_trgi)。
7. 在 TIMx_SMCR 寄存器中写入 SMS=“110”（触发模式），使用 tim_ti2fp2 启动计数器。

OPM 波形通过对比较寄存器执行写操作来定义（考虑时钟频率和计数器预分频器）。

- t_{DELAY} 由写入 TIMx_CCR1 寄存器的值定义。
- t_{PULSE} 由自动重载值与比较值（TIMx_ARR - TIMx_CCR1）之差来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从“0”变为“1”，在计数器达到自动重载值时由“1”变为“0”。为此，应在 TIMx_CCMR1 寄存器中写入 OC1M=111，以使能 PWM 模式 2。如果需要，可选择在 TIMx_CCMR1 寄存器的 OC1PE 和 TIMx_CR1 寄存器的 ARPE 中写入“1”，以使能预装载寄存器。这种情况下，必须在 TIMx_CCR1 寄存器中写入比较值并在 TIMx_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 tim_ti2 上的外部触发事件。本例中，CC1P 的值为“0”。

在本例中，TIMx_CR1 寄存器中的 DIR 和 CMS 位应为低。

由于仅需要 1 个脉冲（单脉冲模式），因此应向 TIMx_CR1 寄存器的 OPM 位写入“1”，以便在发生下一

更新事件（计数器从自动重载值返回到 0）时使计数器停止计数。TIMx_CR1 寄存器中的 OPM 位置“0”时，即选择重复模式。

特例：tim_ocx 快速使能：

在单脉冲模式下，tim_tix 输入的边沿检测会将 CEN 位置 1，表示使能计数器。然后，在计数器值与比较值之间发生比较时，将切换输出。但是，完成这些操作需要多个时钟周期，这会限制可能的最小延迟（t_{DELAY} 最小值）。

如果要输出延迟时间最短的波形，可以将 TIMx_CCMRx 寄存器中的 OCxFE 位置 1。这样会强制 tim_ocxref（和 tim_ocx）对激励信号做出响应，而不再考虑比较的结果。其新电平与发生比较匹配时相同。仅当通道配置为 PWM1 或 PWM2 模式时，OCxFE 才会起作用

14.4.16 可重触发单脉冲模式

此模式允许计数器在接收到刺激后开始计数，并生成一个具有可编程长度的脉冲，但与上一节中描述的非重触发单脉冲模式存在以下差异：

- 只要触发器发生，脉冲就会立即开始（没有可编程的延迟）
- 如果在先前触发的脉冲完成之前发生新的触发，则脉冲会被延长

定时器必须处于从模式，TIMx_SMCR 寄存器的 SMS[3:0]位设置为‘1000’（组合重置+触发模式），OCxM[3:0]位设置为‘1000’或‘1001’，用于可重触发 OPM 模式 1 或 2。

如果定时器配置为递增计数模式，则相应的 CCRx 必须设置为 0（ARR 寄存器设置脉冲长度）。如果定时器配置为递减计数模式，CCRx 必须大于或等于 ARR。

*注意：在可重复触发单脉冲模式中，*CCxIF *标志不重要*

OCxM[[3:0]*和*SMS[3:0]*位字段由于兼容性的原因被分成两个部分，最高有效位与最低*3 *位不是连续的。*

OCxM *这种模式不能与中央对齐*PWM *模式一起使用。*TIMx_CR1 *寄存器中的*CMS[1:0]*必须设置为*00*。*

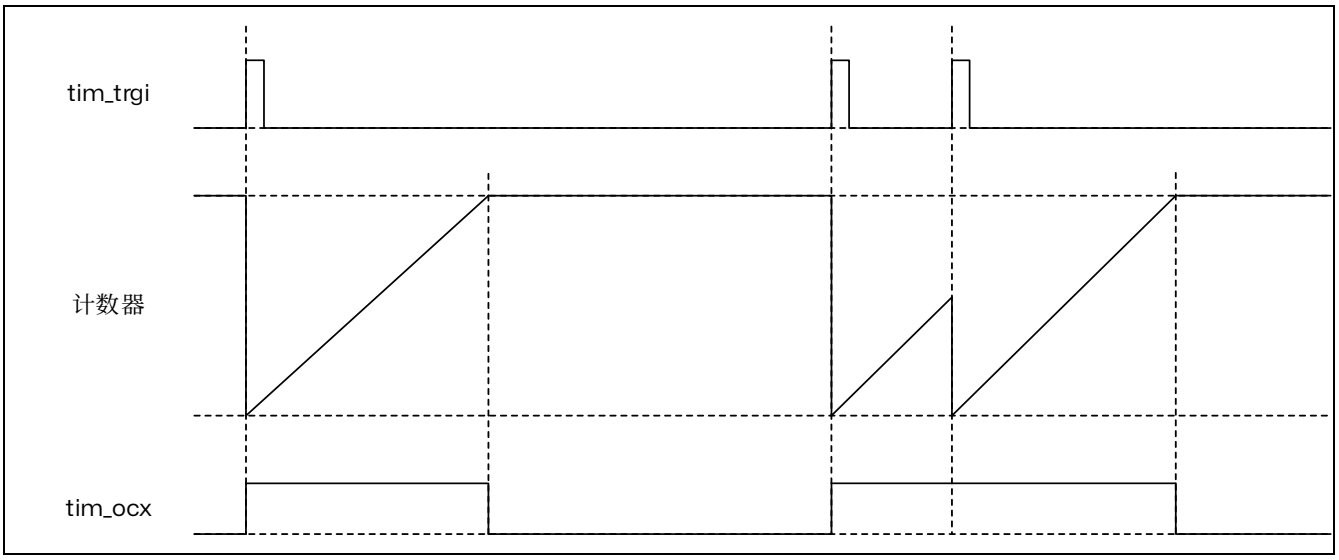


图 14.48 可重触发单脉冲模式

14.4.17 在比较模式下的脉冲

在比较匹配事件发生时，可以生成一个脉冲。当计数器的值等于给定的比较值时，可以生成一个具有

可编程脉冲宽度的信号，用于调试或同步。

此模式适用于任何从机模式选择，包括编码器模式、边沿对齐和中央对齐计数模式。它仅适用于通道 3 和通道 4。脉冲发生器是唯一的，由两个通道共享，如下图所示。

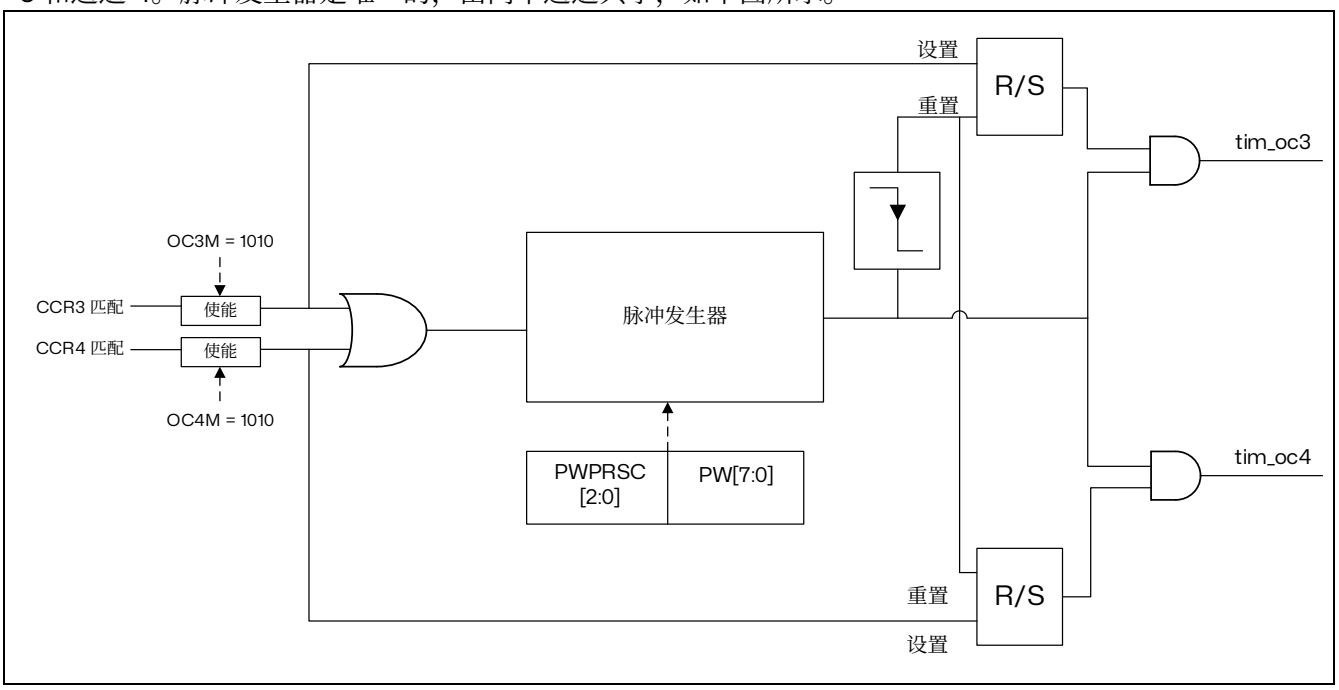


图 14.49 脉冲发生器电路

下图展示了如何在边沿对齐模式和编码器工作模式下生成脉冲。

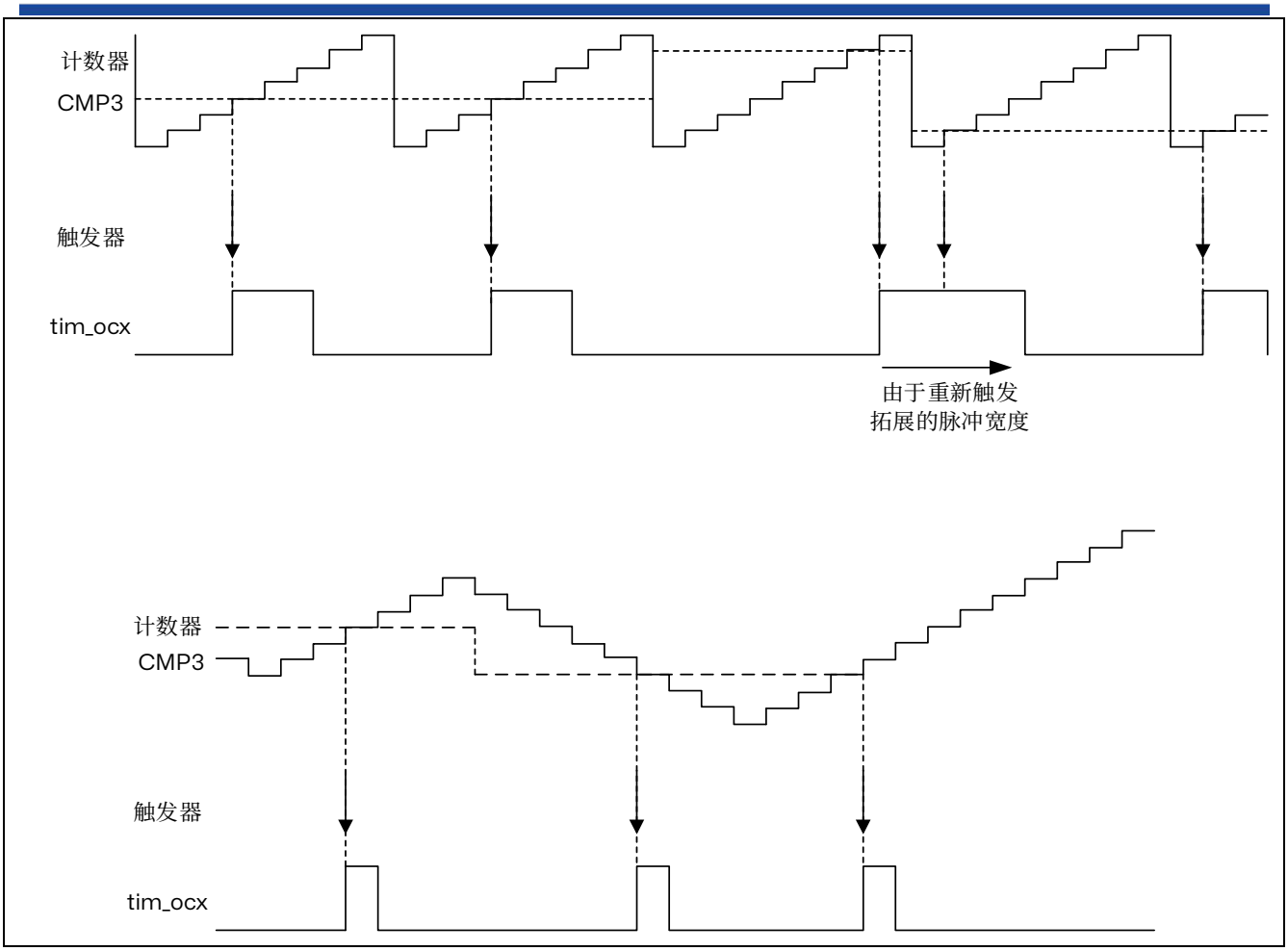


图 14.50 在比较事件上的脉冲生成，适用于边沿对齐模式和编码器模式

此输出比较模式是通过 TIMx_CCMR2 寄存器中的 OC3M[3:0]和 OC4M[3:0]位字段进行选择的。

脉冲宽度通过寄存器中的 PW[7:0]位域进行编程，使用特定的时钟，并根据 PWPRSC[2:0]位的预设进行预分频，具体如下：

$t_{PW} = PW[7:0] \times t_{PWG}$

其中 $t_{PWG} = (2^{(PWPRSC[2:0])}) \times t_{tim_ker_ck}$。

给出了分辨率和最大值，具体取决于预分频器的值。

脉冲可以重新触发：在脉冲正在进行时触发新的脉冲，会导致脉冲被延长。

注意：如果同时使能两个通道，只要一个通道上的触发器不与并发输出上生成的脉冲重叠，脉冲就会独立发出。相反，如果两个触发器重叠，与第一个到达的触发器相关的脉冲宽度会被延长（由于重新触发），而最后一个到达的触发器的脉冲宽度是正确的（如下图所示）。

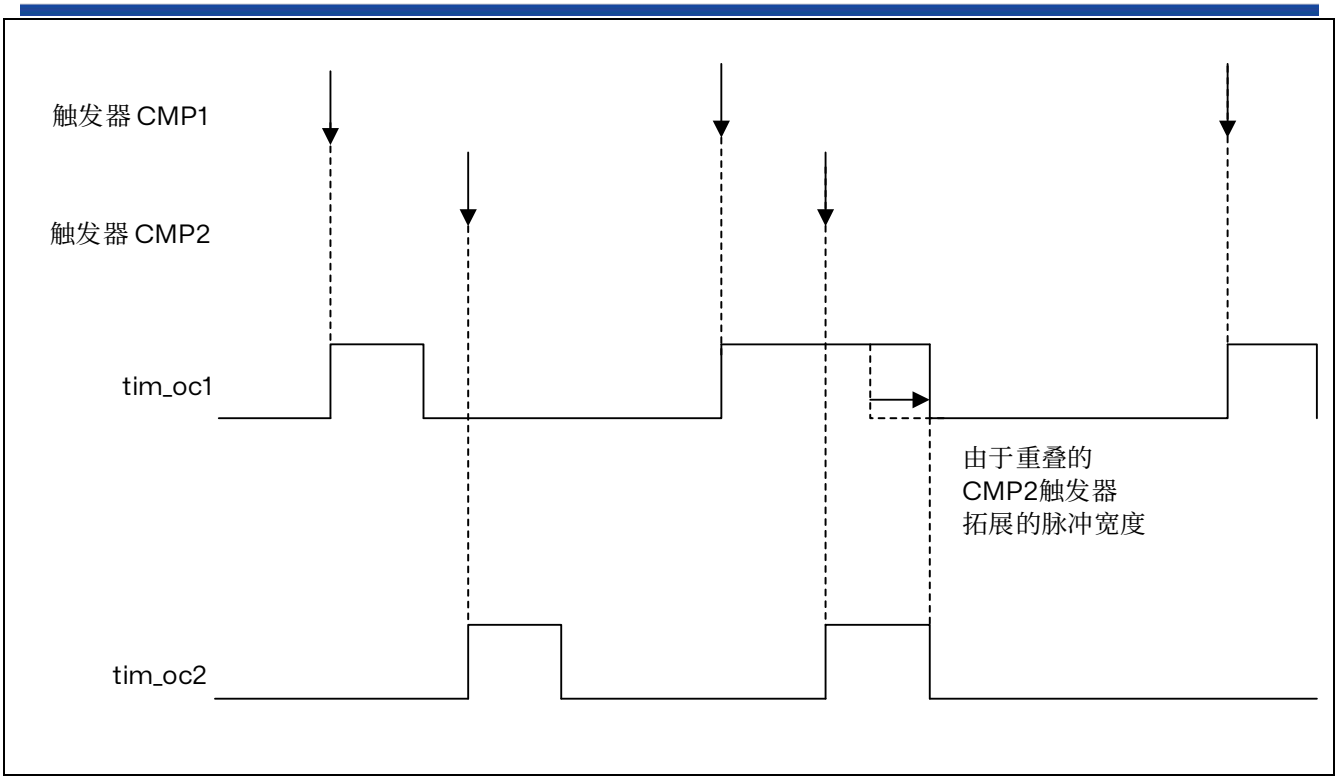


图 14.51 在同时触发的情况下延长脉冲宽度

14.4.18 编码器接口模式

正交编码器

选择编码器接口模式时，如果计数器仅在 tim_ti2 边沿处计数，在 TIMx_SMCR 寄存器中写入 SMS=0001；如果计数器仅在 tim_ti1 边沿处计数，写入 SMS=0010；如果计数器在 tim_ti1 和 tim_ti2 边沿处均计数，则写入 SMS=0011。

通过编程 TIMx_CCER 寄存器的 CC1P 和 CC2P 位，选择 tim_ti1 和 tim_ti2 极性。如果需要，还可对输入滤波器进行编程。

tim_ti1 和 tim_ti2 两个输入用于连接增量编码器。请参见下表。如果使能计数器（在 TIMx_CR1 寄存器的 CEN 位中写入“1”），则计数器的时钟由 tim_ti1fp1 或 tim_ti2fp2 上的每次有效信号转换提供。tim_ti1fp1 和 tim_ti2fp2 是进行输入滤波器和极性选择后 tim_ti1 和 tim_ti2 的信号，如果不进行滤波和反相，则 tim_ti1fp1=tim_ti1，tim_ti2fp2=tim_ti2。将根据两个输入的信号转换序列，产生计数脉冲和方向信号。根据该信号转换序列，计数器相应递增或递减计数，同时硬件对 TIMx_CR1 寄存器的 DIR 位进行相应修改。任何输入（tim_ti1 或 tim_ti2）发生信号转换时，都会计算 DIR 位，无论计数器是仅在 tim_ti1 或 tim_ti2 边沿处计数，还是同时在 tim_ti1 和 tim_ti2 处计数。

编码器接口模式就相当于带有方向选择的外部时钟。这意味着，计数器仅在 0 到 TIMx_ARR 寄存器中的自动重载值之间进行连续计数（根据具体方向，从 0 递增计数到 ARR，或从 ARR 递减计数到 0）。因此，在启动前必须先配置 TIMx_ARR。同样，捕获、比较、预分频器、触发输出功能继续正常工作。

在此模式下，计数器会根据增量编码器的速度和方向自动进行修改，因此，其内容始终表示编码器的位置。计数方向对应于所连传感器的旋转方向。下表汇总了可能的组合（假设 tim_ti1 和 tim_ti2 不同时切换）。

表 14.13 计数方向与编码器信号的关系（CC1P = CC2P = 0）

有效边沿	SMS[3:0]	相对信号的电平（tim_ti1fp1 对应 tim_ti2, tim_ti2fp2 对应 tim_ti1）	tim_ti1fp1 信号		tim_ti2fp2 信号	
			上升	下降	上升	下降
仅 x1 模式在 tim_ti1 计数	1110	高	递减	递增	不计数	不计数
		低	不计数	不计数	不计数	不计数
仅 x1 模式在 tim_ti2 计数	1111	高	不计数	不计数	递增	递减
		低	不计数	不计数	不计数	不计数
仅 x2 模式在 tim_ti1 计数	0001	高	递减	递增	不计数	不计数
		低	递增	递减	不计数	不计数
仅 x2 模式在 tim_ti2 计数	0010	高	不计数	不计数	递增	递减
		低	不计数	不计数	递减	递增
x4 模式在 tim_ti1 和 tim_ti2 计数	0011	高	递减	递增	递增	递减
		低	递增	递减	递减	递增

外部增量编码器可直接与 MCU 相连，无需外部接口逻辑。不过，通常使用比较器将编码器的差分输出转换为数字信号。这样大幅提高了抗噪声性能。用于指示机械零位的第三个编码器输出可与外部中断输入相连，用以触发计数器复位。

下图以计数器工作为例，说明了计数信号的生成和方向控制。同时也说明了选择双边沿时如何对输入抖动进行补偿。将传感器靠近其中一个切换点放置时可能出现这种情况。本例中假设配置如下：

- CC1S="01"（TIMx_CCMR1 寄存器，tim_ti1fp1 映射到 tim_ti1 上）。
- CC2S="01"（TIMx_CCMR2 寄存器，tim_ti1fp2 映射到 tim_ti2 上）。
- CC1P="0"，CC1NP="0"（TIMx_CCER 寄存器，tim_ti1fp1 未反相，tim_ti1fp1= tim_ti1）。
- CC2P="0"，CC2NP="0" TIMx_CCER 寄存器，tim_ti1fp2 未反相，tim_ti1fp2= tim_ti2）。
- SMS= "011"（TIMx_SMCR 寄存器，两个输入在上升沿和下降沿都处于有效状态）。
- CEN="1"（TIMx_CR1 寄存器，使能计数器）。

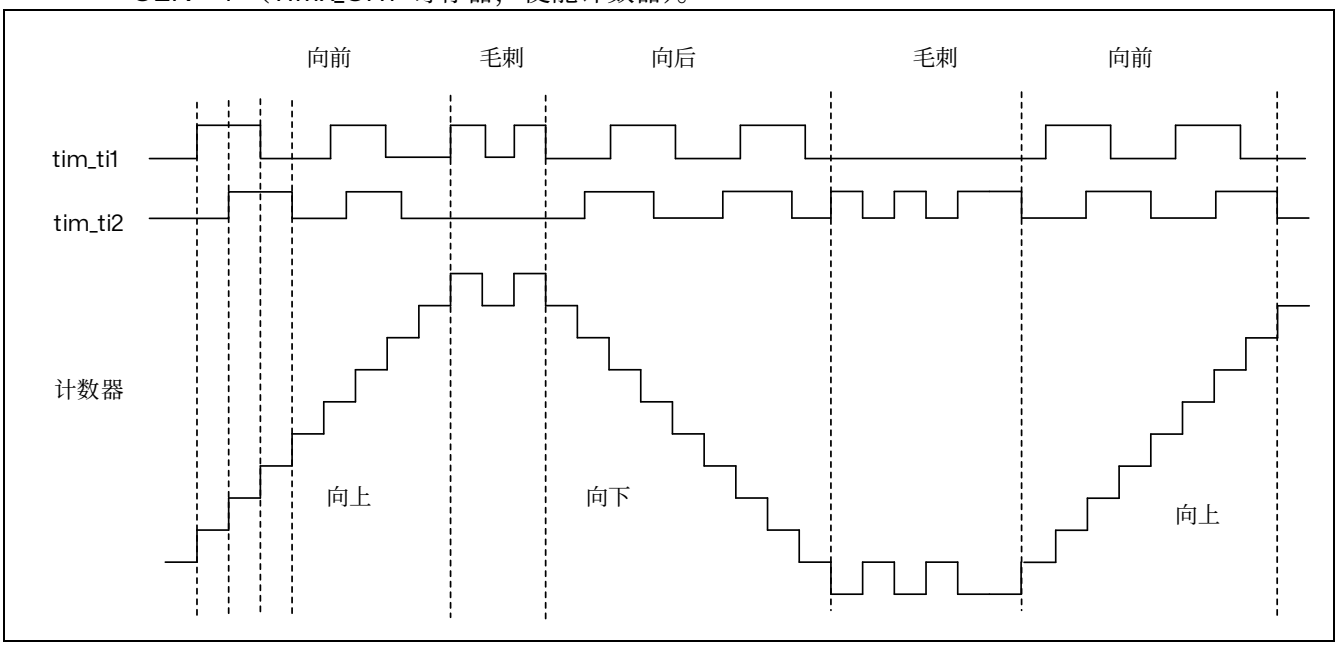


图 14.52 编码器接口模式下的计数器工作示例

下图举例说明 tim_ti1fp1 极性反相时计数器的行为（除 CC1P=“1”外，其它配置与上例相同）。

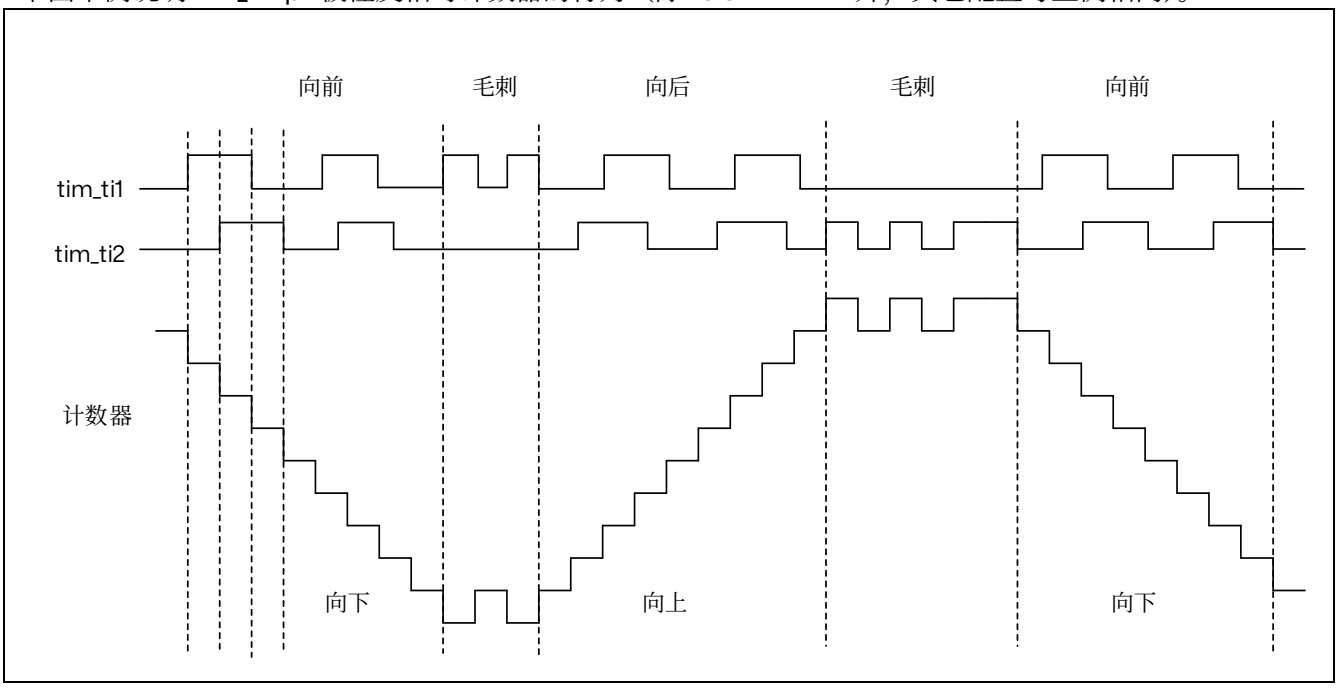


图 14.53 tim_ti1fp1 极性反相时的编码器接口模式示例

下图显示了各种计数模式下速度反转期间计时器计数器的值。

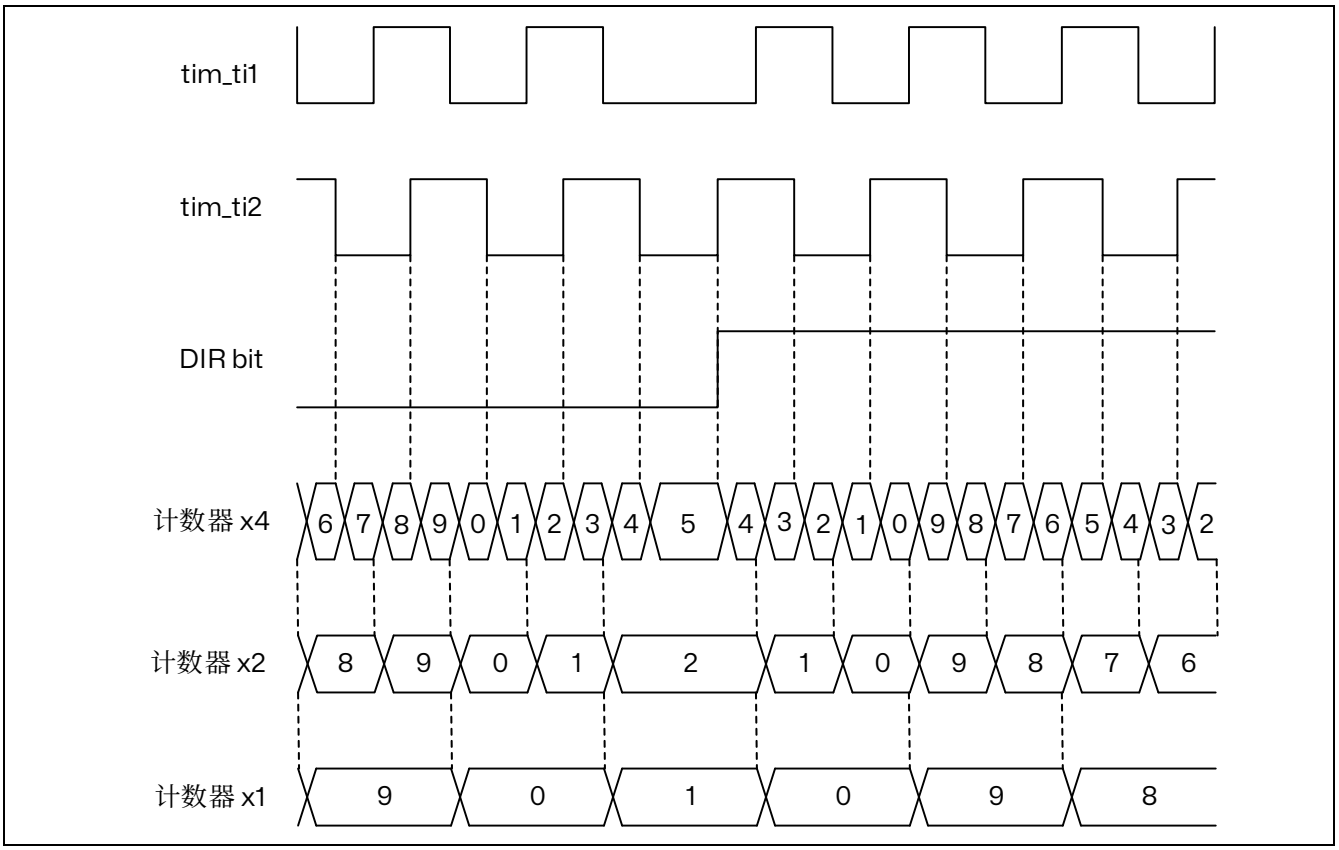


图 14.54 正交编码器的计数模式

当定时器配置为编码器接口模式时，可以提供传感器当前位置的信息。通过测量两个编码器事件之间

的周期，使用配置为捕获模式的第二个定时器，可以获得动态信息（速度、加速度、减速度）。指示机械零的编码器输出可用于此目的。根据两个事件之间的时间，还可以定期读取计数器的值。如果可用，可以通过将计数器的值锁存到第三个输入捕获寄存器来实现（此时捕获信号必须是周期性的，并且可以由另一个定时器生成）。

TIMx_CR1 寄存器中的 IUFREMAP 位将更新中断标志（UIF）的连续副本强制复制到定时器计数器寄存器的第 31 位（TIMxCNT[31]）。这允许原子地读取计数器的值和由 UIFCPY 标志触发的潜在的回滚条件。这可以通过避免在后台任务（计数器读取）和中断（更新中断）之间共享处理来简化角速度的计算。

UIF 和 UIFCPY 标志之间没有延迟。

时钟加方向编码器模式

除了正交编码器模式外，定时器还支持其他类型的编码器。

在下图中显示的“时钟加方向”模式下，时钟在 tim_ti2 上提供单线，而方向使用 tim_ti1 输入强制。

此模式通过 TIMx_SMCR 寄存器中的 SMS[3:0]位域使能，如下所示：

- 1010：x2 模式，计数器在时钟的上升沿和下降沿都更新
- 1011：x1 模式，计数器在单个时钟边沿更新，根据 CC2P 位的值而定：CC2P=0 对应上升沿敏感，CC2P=1 对应下降沿敏感。

tim_ti1 上的方向信号极性通过 CC1P 位设置：0 对应正极性（当 tim_ti1 为高电平时向上计数，当 tim_ti1 为低电平时向下计数），CC1P=1 对应负极性（当 tim_ti1 为低电平时向上计数）。

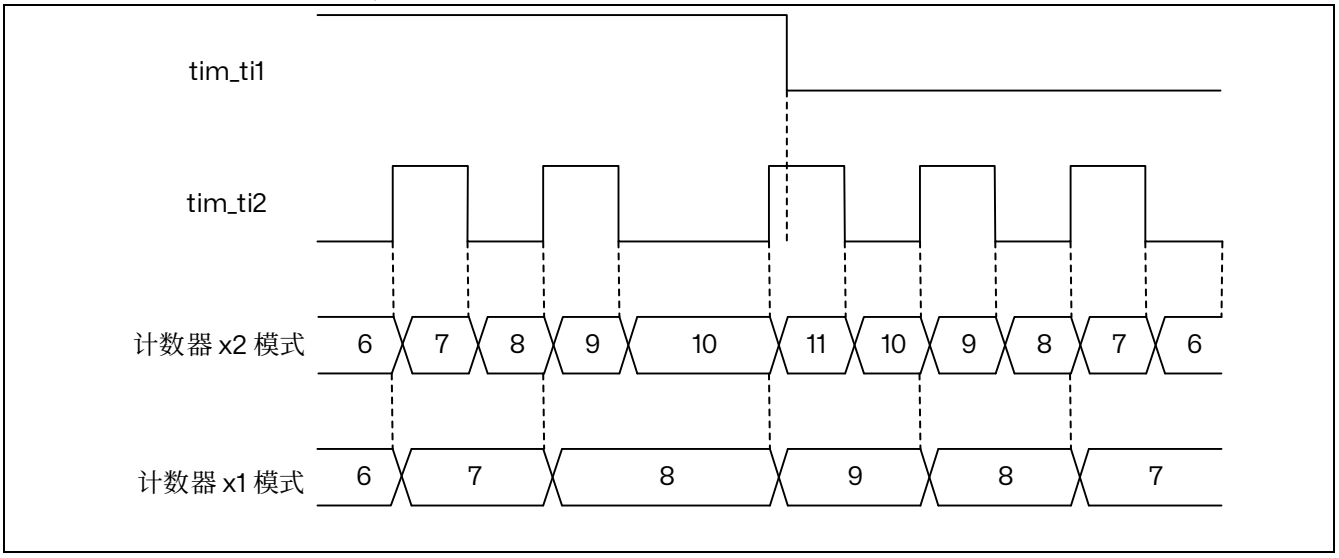


图 14.55 方向加时钟编码器模式

方向时钟编码器模式

在下图中的“方向时钟”模式下，时钟在两条线上提供，根据方向一次使用一条，以便有一条向上计数时钟线和一条向下计数时钟线。

此模式通过 TIMx_SMCR 寄存器中的 SMS[3:0]位域使能，如下所示：

- 1100：x2 模式，计数器在两条时钟线的上升沿和下降沿都更新。CC1P 和 CC2P 位编码时钟空闲状态。CCxP=0 对应高电平空闲状态（参考下图），CCxP=1 对应低电平空闲状态（参考下图）。
- 1101：x1 模式，计数器在单个时钟边沿更新，根据 CC1P 和 CC2P 位的值而定。CCxP=0 对应下降沿敏感性和高电平空闲状态（参考下图），CCxP=1 对应上升沿敏感性和低电平空闲状态（参考下图）。

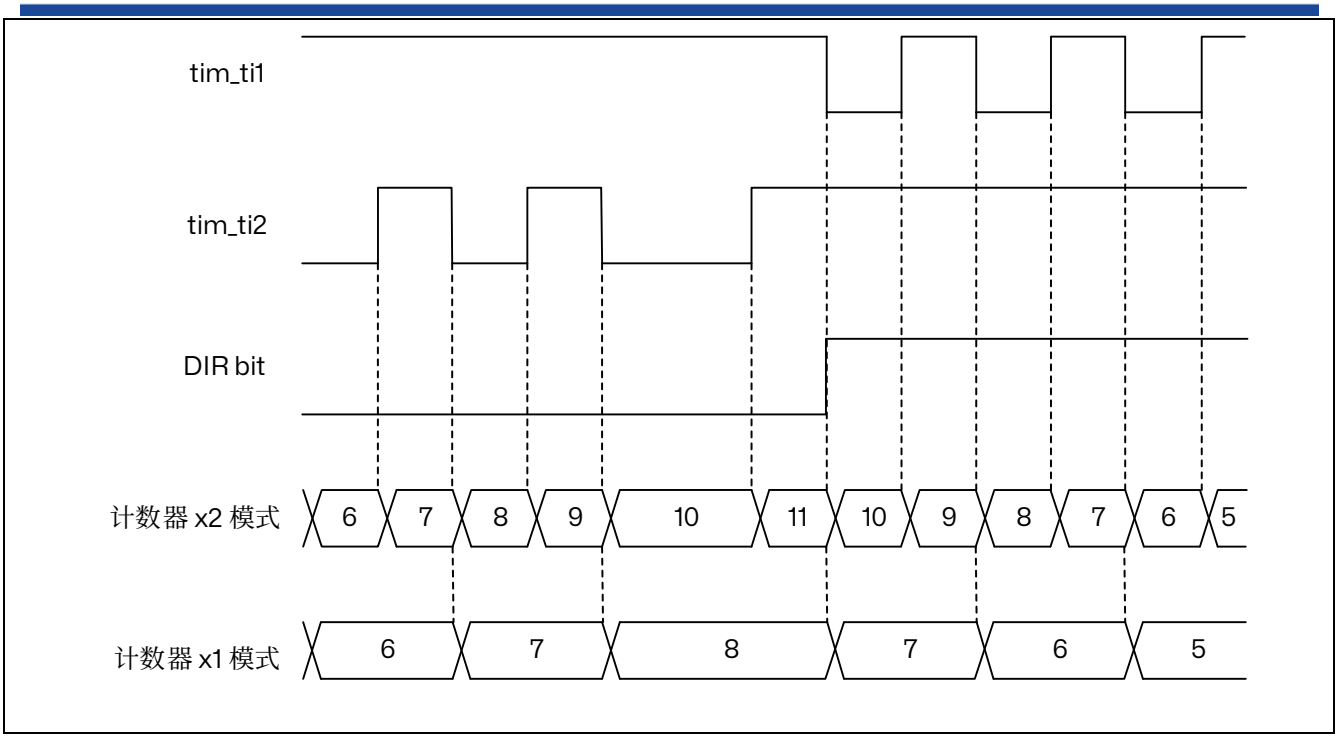


图 14.56 方向时钟编码器模式（CC1P=CC2P=0）

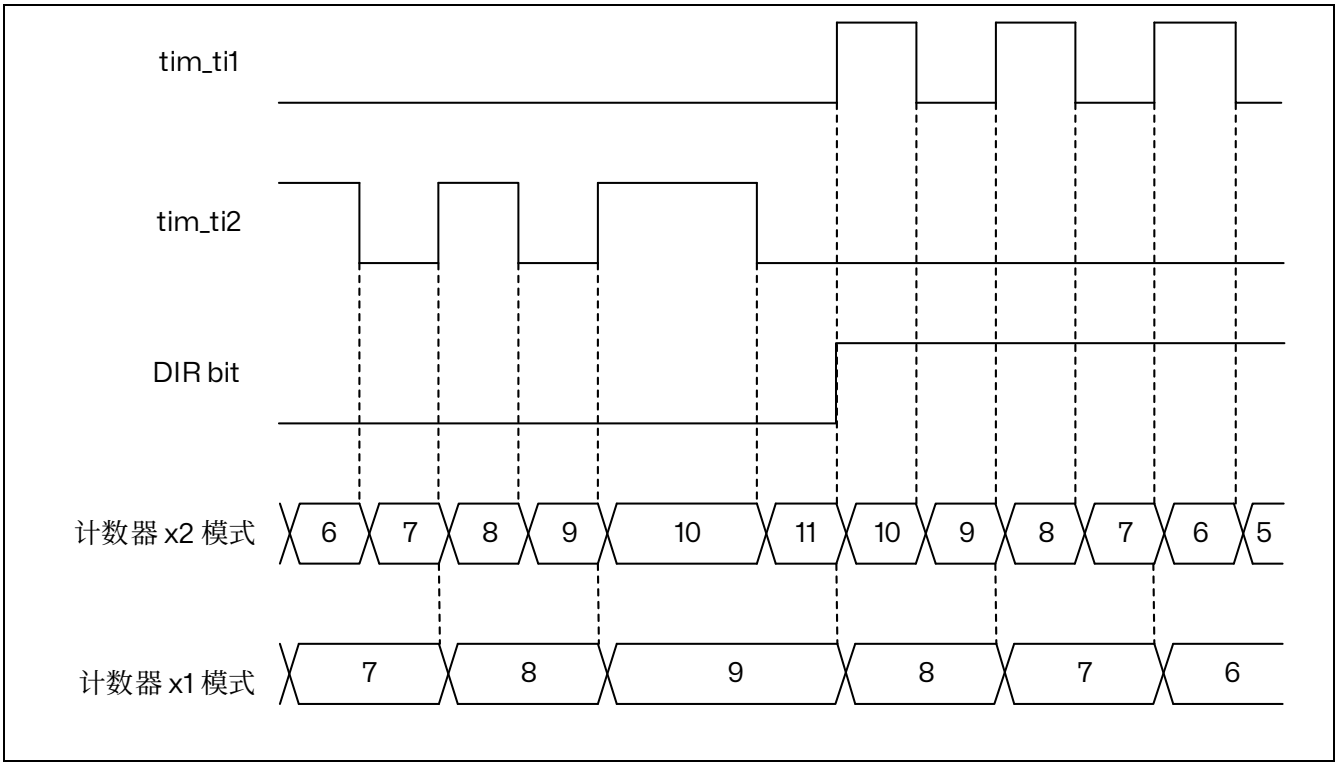


图 14.57 方向时钟编码器模式（CC1P=CC2P=1）

下表详细说明了方向时钟模式对任何输入转变的操作方式。

表 14.14 计数方向与编码器信号及极性设置的关系

方向时钟模式	SMS[3:0]	相对信号的电平（tim_ti1fp1 对应 tim_ti2, tim_ti2fp2 对应 tim_ti1）	tim_ti1fp1 信号		tim_ti2fp2 信号	
			上升	下降	上升	下降
x2 模式 CCxP=0	1100	高	递减	递减	递增	递增
		低	不计数	不计数	不计数	不计数
x2 模式 CCxP=1	1100	高	不计数	不计数	不计数	不计数
		低	递减	递减	递增	递增
x1 模式 CCxP=0	1101	高	不计数	递减	不计数	递增
		低	不计数	不计数	不计数	不计数
x1 模式 CCxP=1	1101	高	不计数	不计数	不计数	不计数
		低	递减	不计数	递增	不计数

索引输入

计数器可以通过来自编码器的索引信号进行复位，以指示绝对参考位置。索引信号必须连接到 tim_etr_in 输入。它可以使用数字输入滤波器进行滤波。

通过设置 TIMX_ECR 寄存器中的 IE 位，可以使能索引功能。只有在编码器模式下，当 SMS[3:0]位域具有以下值时，才可以设置 IE 位：0001、0010、011、1010、1011、1100、1101、1110、1111。

根据下图所示，有几种可设置的编码器选项，用于索引脉冲的调节：

- 与 A 和 B 门控：脉冲宽度为 1/4 的通道周期，与 A 和 B 边缘对齐
- 与 A（或与 B）门控：脉冲宽度为 1/2 的通道周期，与通道 A（或通道 B）的两个边缘对齐
- 非门控：脉冲宽度可达一个通道周期，不与任何边缘对齐

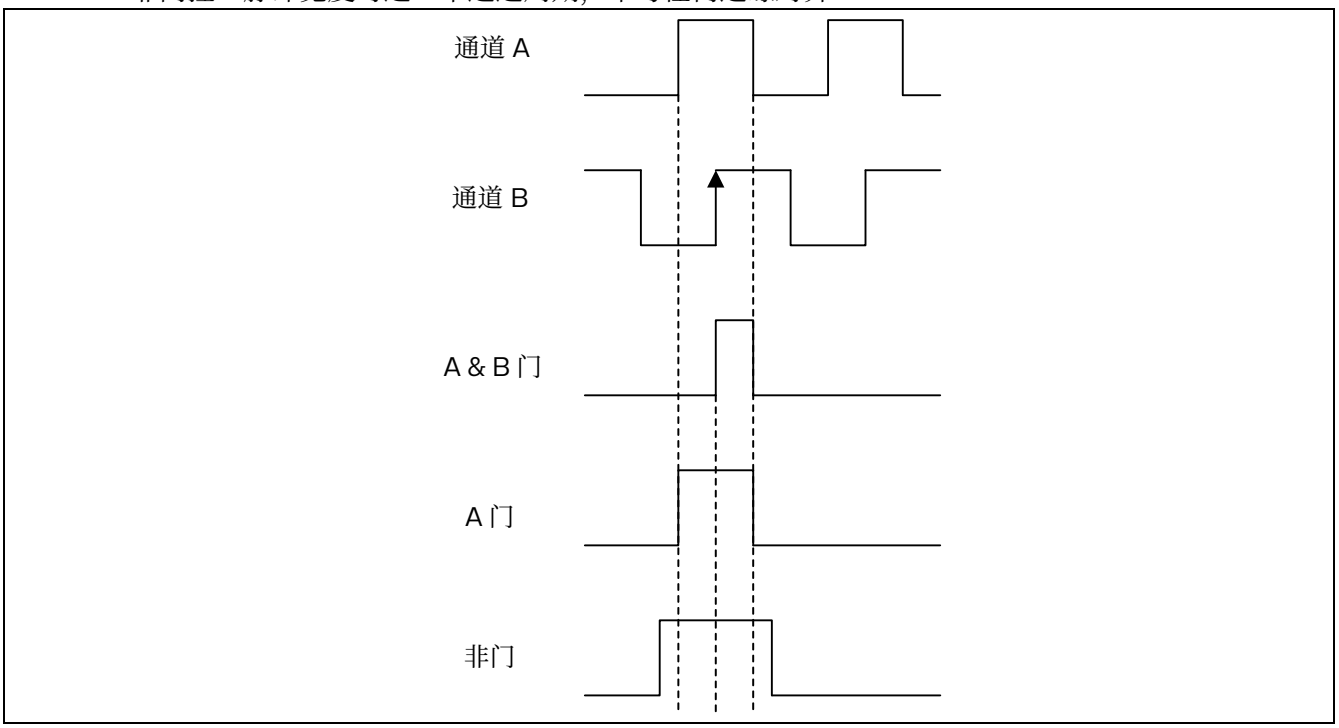


图 14.58 索引门控选项

电路可以容忍索引信号的抖动，无论门控模式如何，如下图所示。

在非门控模式下，信号必须严格低于 2 个编码器周期。如果脉冲宽度大于或等于 2 个编码器周期，则计数器会多次复位。

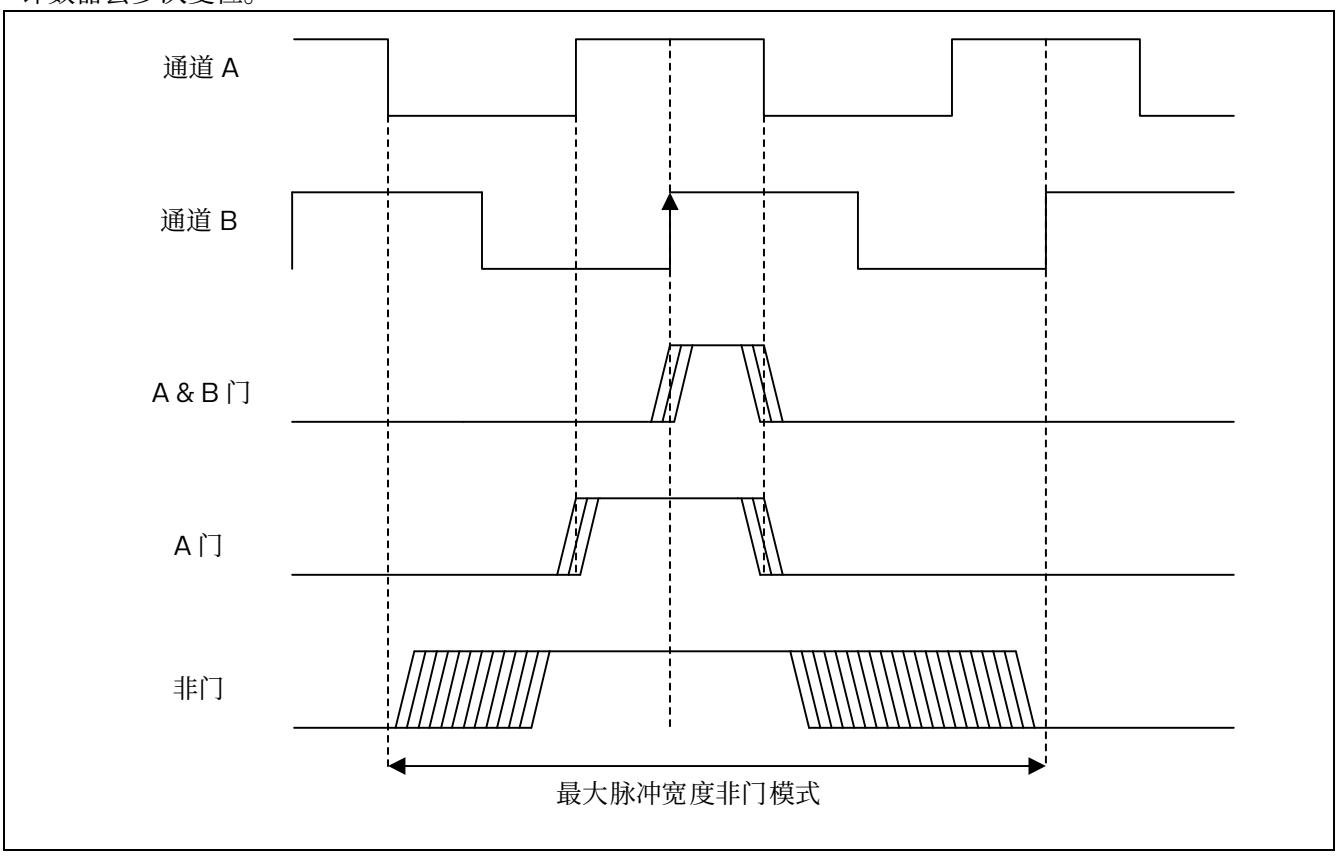


图 14.59 抖动的索引信号

定时器支持 3 种门控选项，不需要任何特定的编程。只需要定义在哪个编码器状态（即通道 A 和通道 B 状态组合）上，索引必须同步，使用 TIMx_ECR 寄存器中的 IPOS[1:0]位域。

索引检测事件根据计数方向的不同而有所不同，以确保速度反转时的对称操作：

- 递增计数时（DIR 位=0），计数器被复位。
- 递减计数时，计数器被设置为 TIMx_ARR。

这使得索引可以在相同的机械角度位置上生成，无论计数方向如何。下图显示了在一个简单的例子中（一个编码器每次机械旋转提供 4 个边缘），索引在哪个位置生成。

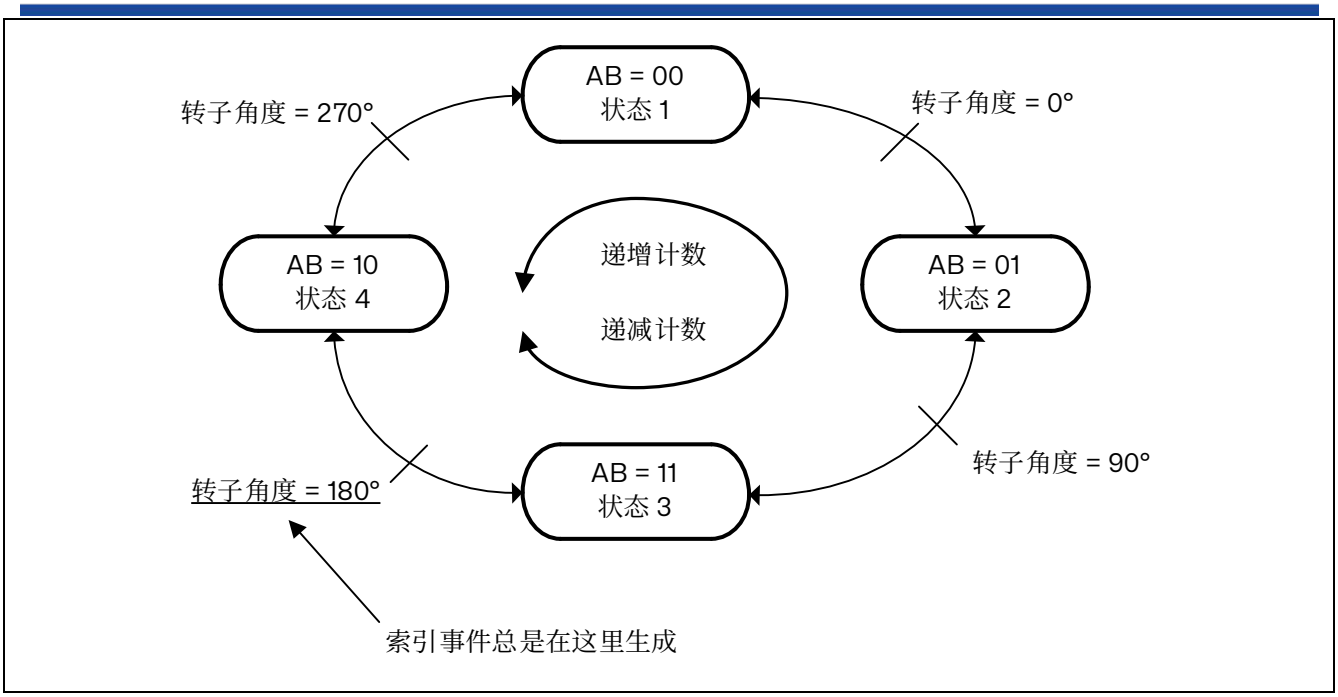


图 14.60 IPOS[1:0] = 11 时的索引生成

下图展示了 IPOS[1:0] = 11 时的波形和相应值。它表明，计数器值被强制的时刻会自动根据计数方向进行调整：

- 递增计数时（DIR 位=0），当编码器状态为“11”（ChA=1，ChB=1）时，计数器设为 0。
- 递减计数时（DIR 位=1），当退出“11”状态时，计数器设为 TIMx_ARR。

可以在索引检测事件发生时发出中断。

箭头指示在哪个转换上生成索引事件中断。

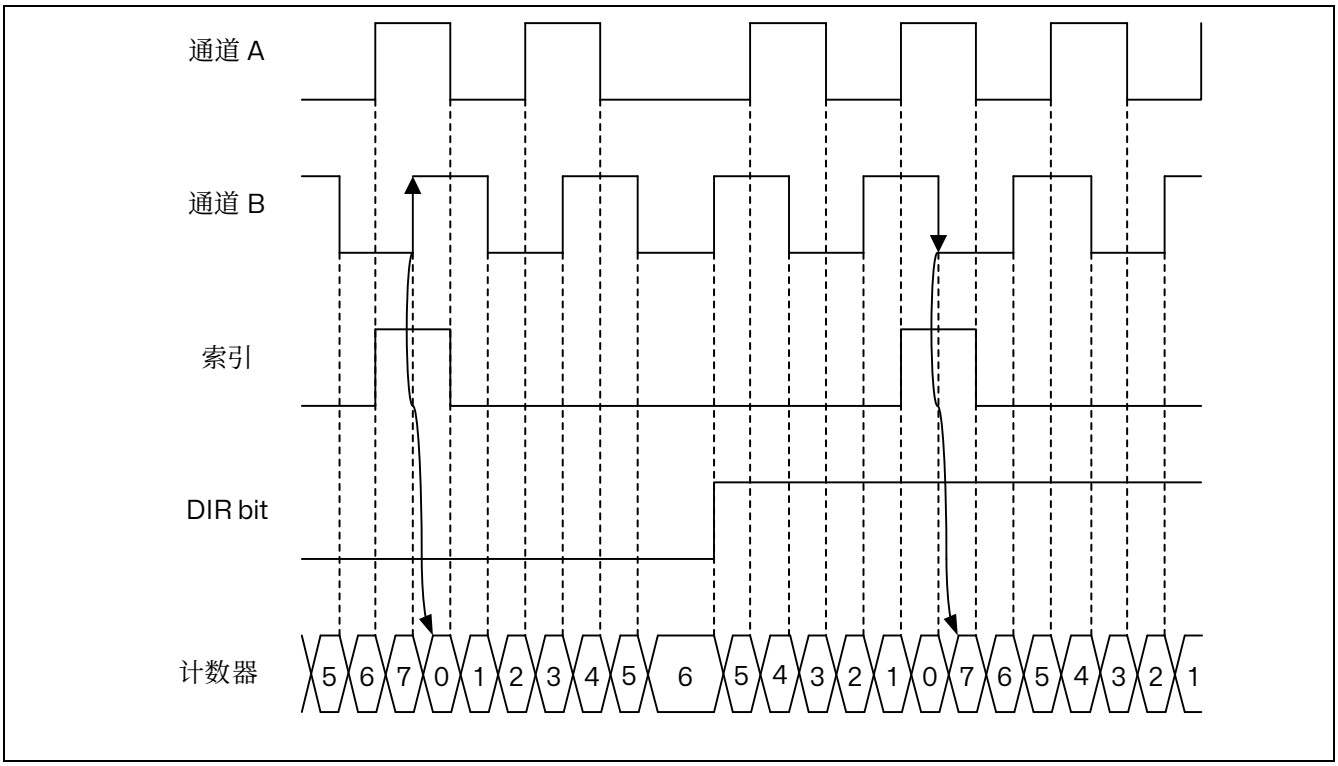


图 14.61 通道 A 上带索引门控的计数器读取（IPOS[1:0] = 11）

下图展示了非门控模式的波形和相应值。箭头指示在哪个转换上生成索引事件。

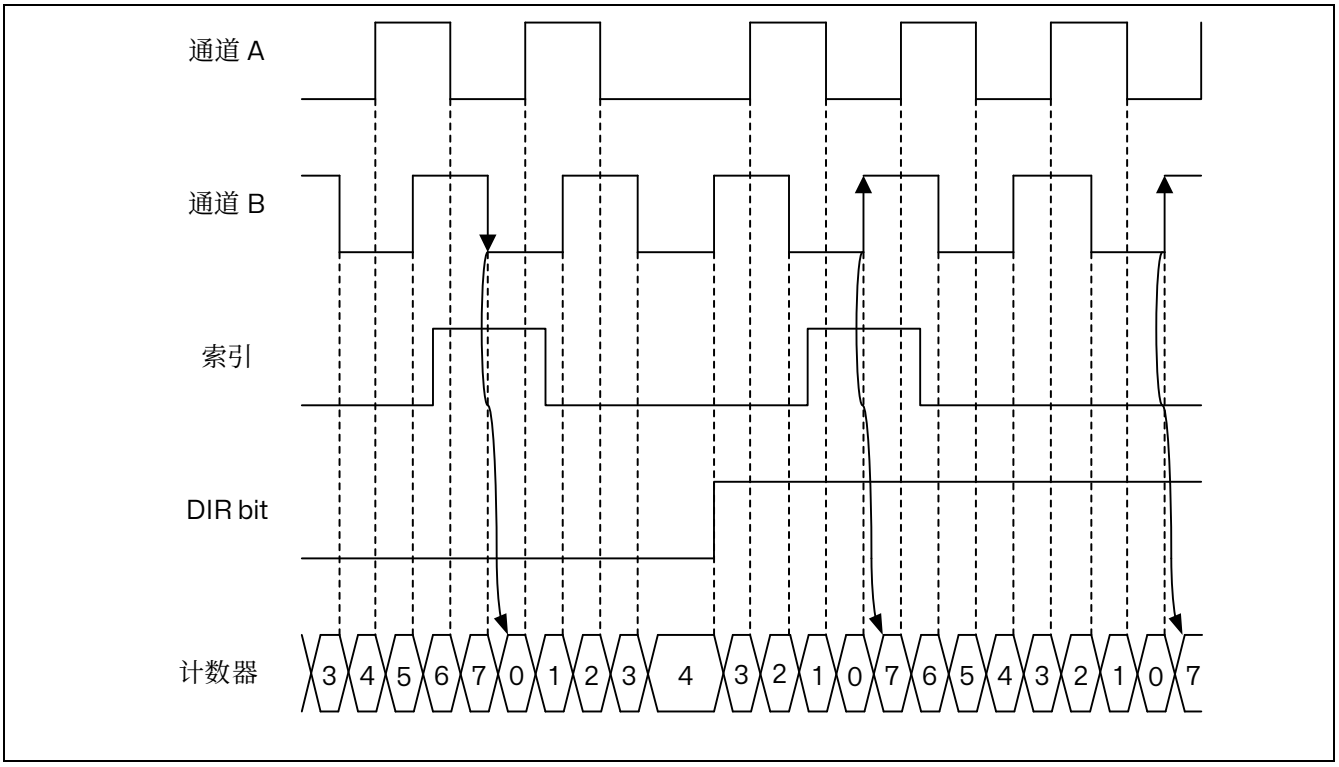


图 14.62 非门控模式下索引的计数器读取（IPOS[1:0] = 00）

下图展示了针对各种脉冲对齐情况下的“在 A 和 B 门控”模式处理方式。箭头指示在哪个转换上生成索引事件。

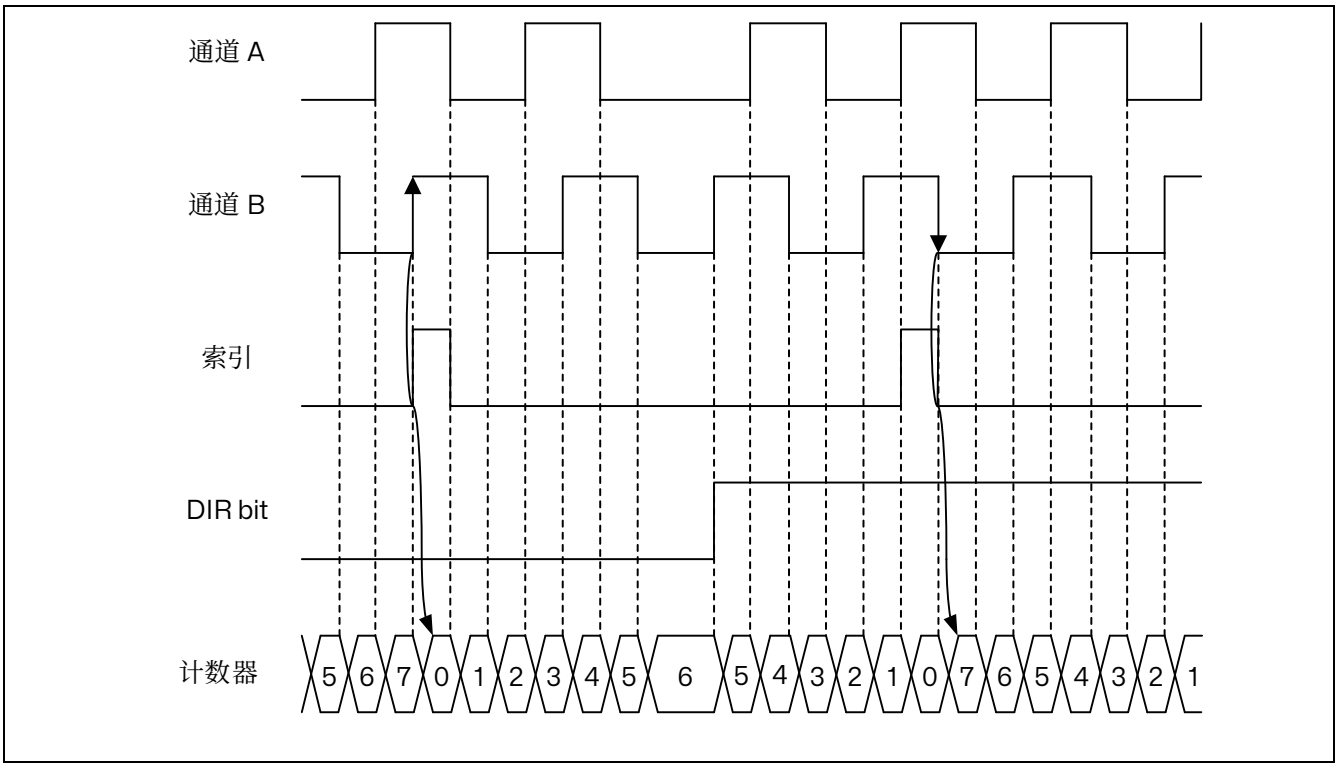
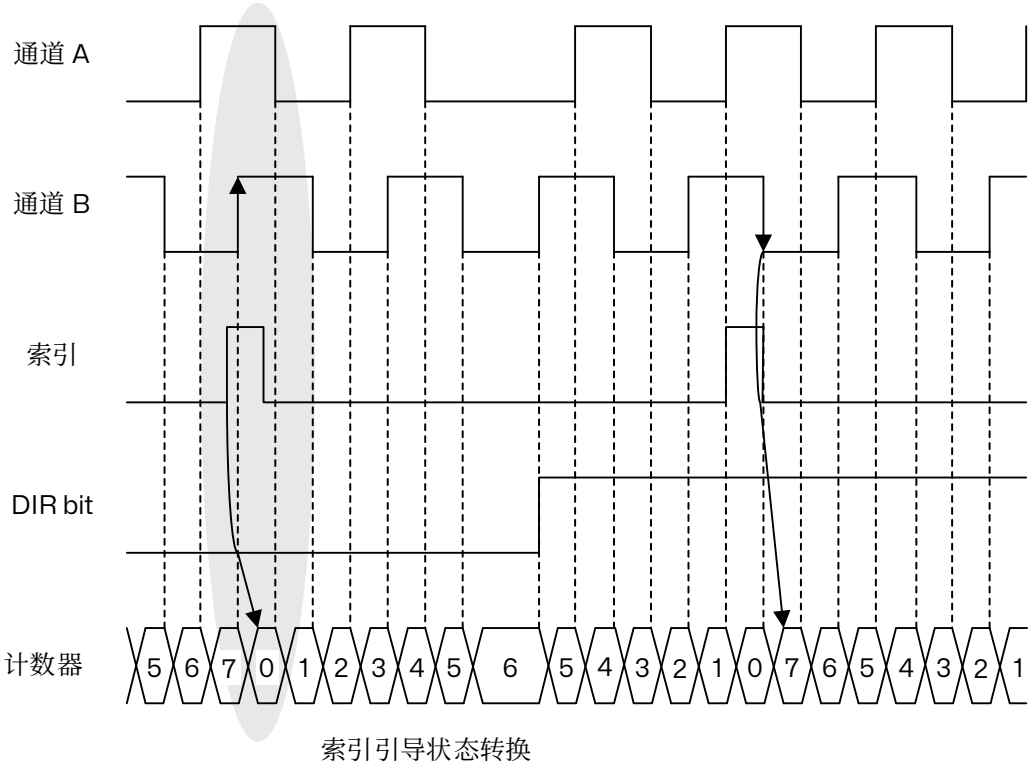
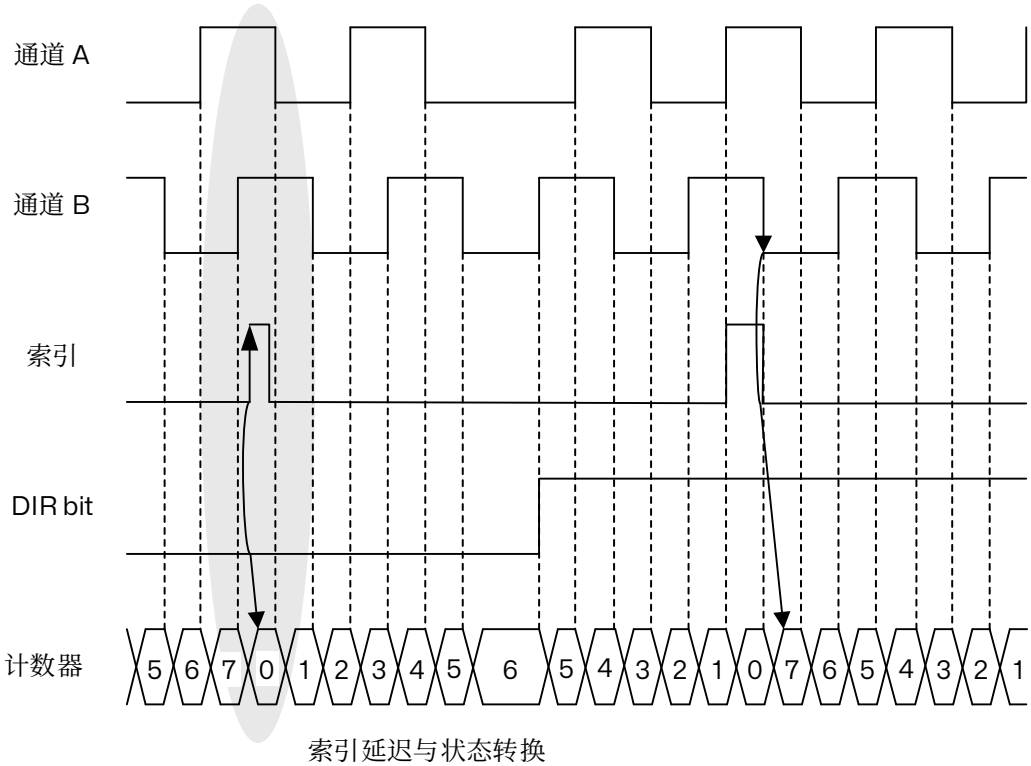


图 14.63 通道 A 和 B 上带索引门控的计数器读取

下两张图详细说明了后续索引脉冲可能比编码器时钟周期的四分之一窄的情况。



索引引导状态转换



索引延迟与状态转换

图 14.64 窄索引脉冲情况下的编码器模式行为（IPOS[1:0] = 11）

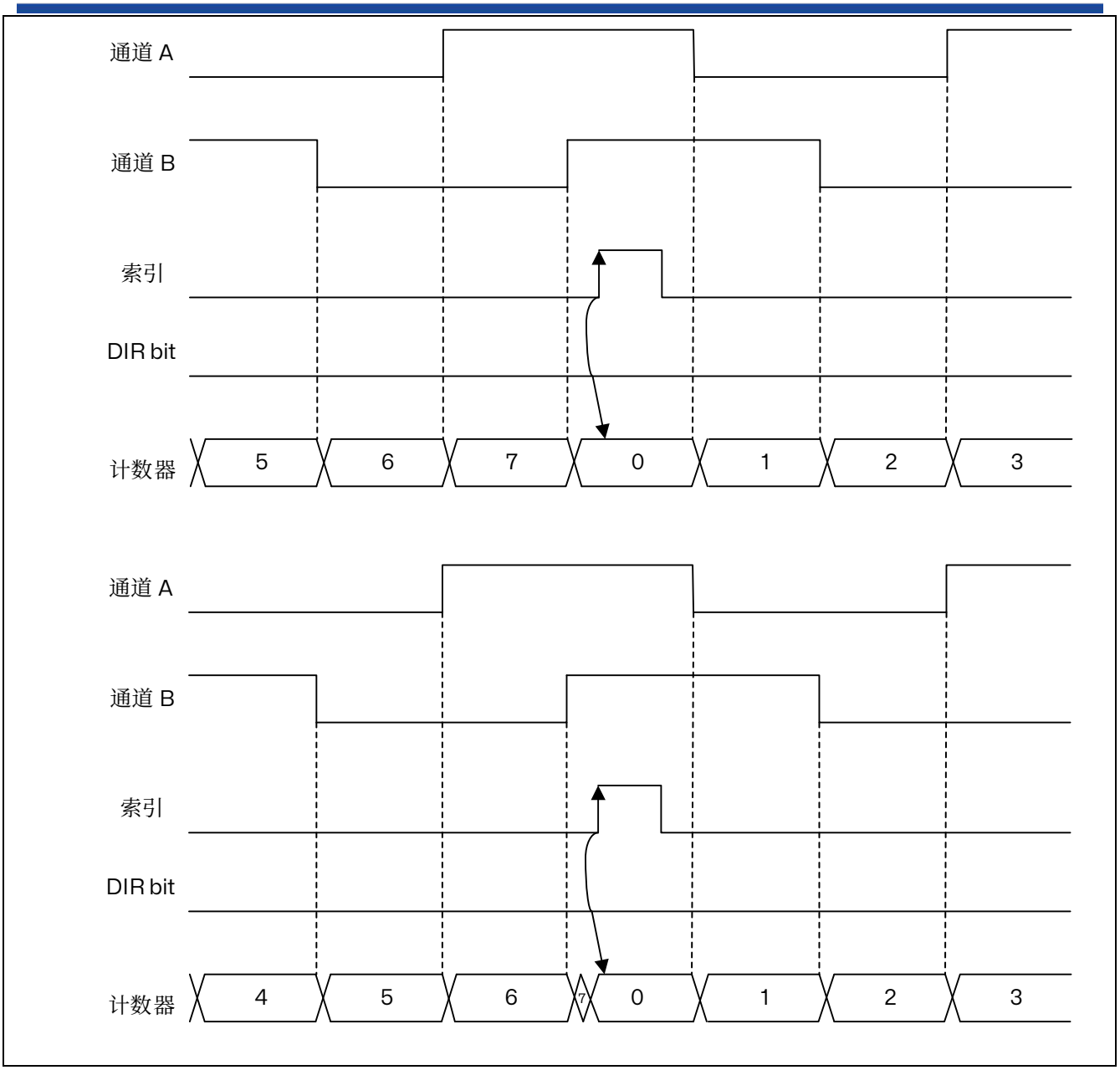


图 14.65 计数器重置窄索引脉冲（更近距离，ARR = 0x07）

下图说明了 x1 和 x2 模式下的索引管理方式。

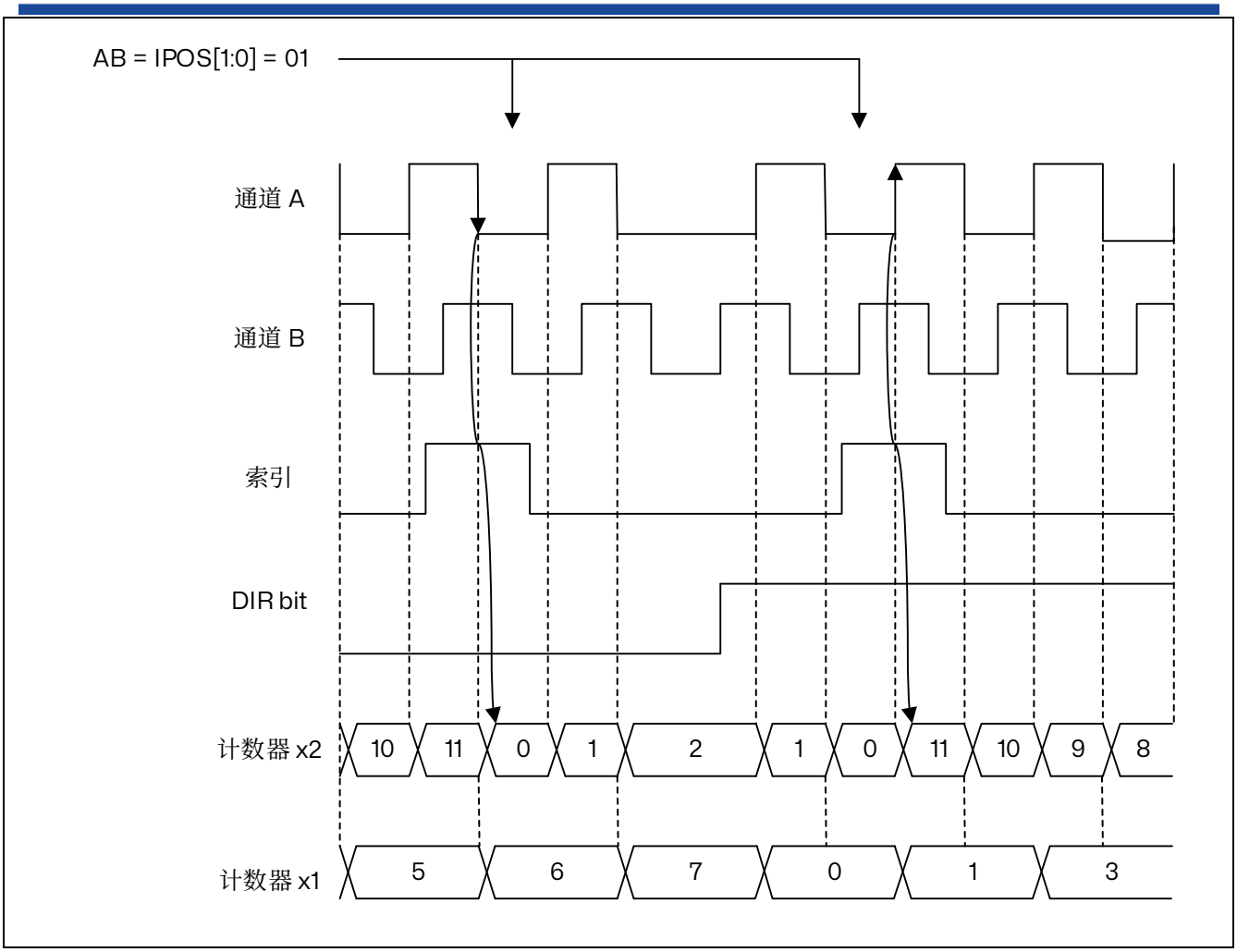


图 14.66 x1 和 x2 模式下的索引行为（IPOS[1:0] = 01）

方向索引灵敏度

TIMx_ECR 寄存器中的 IDIR[1:0]位域允许索引仅在选定计数方向上有效。

下图显示了根据 IDIR[1:0]值索引和计数器重置事件之间的关系。

注意：当 IE 位被复位时（禁用索引模式），必须写入 IDR[1:0]位域。

注意：在时钟+方向模式下不支持定向索引灵敏度。当 SMS[3:0] = 1010 或 1011 时，IDIR[1:0]必须设置为 00。

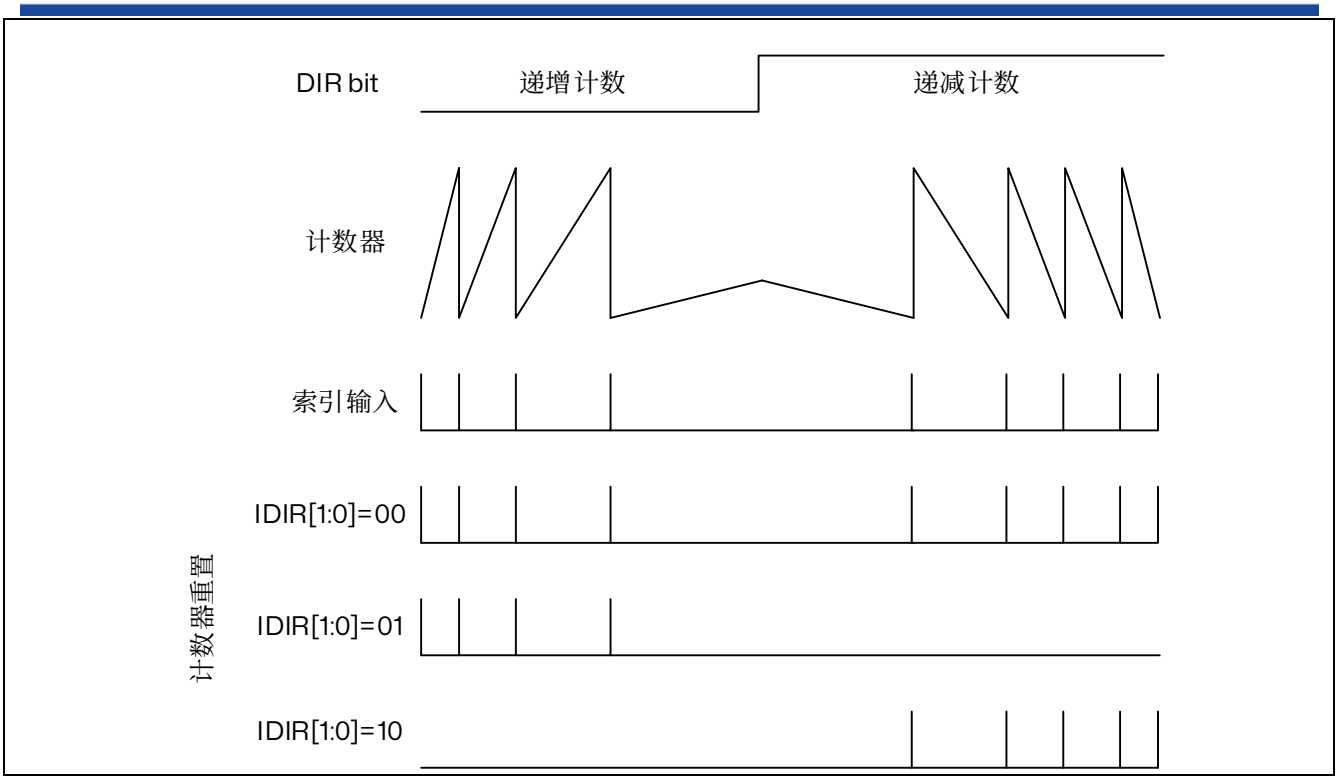


图 14.67 定向索引灵敏度

特殊的第一索引事件管理

TIMx_ECR 寄存器中的 FIDX 位允许仅捕获一次索引，如下图所示。一旦第一个索引到达，任何后续索引都会被忽略。如果需要，可以通过将 FIDX 位写入 0 并再次将其设置为 1 来重新启动电路。

注意：当 FIDX = 1 时，如果方向在位置 0 处改变（索引激活），索引可以发布两次（IDXF 标志设置）。

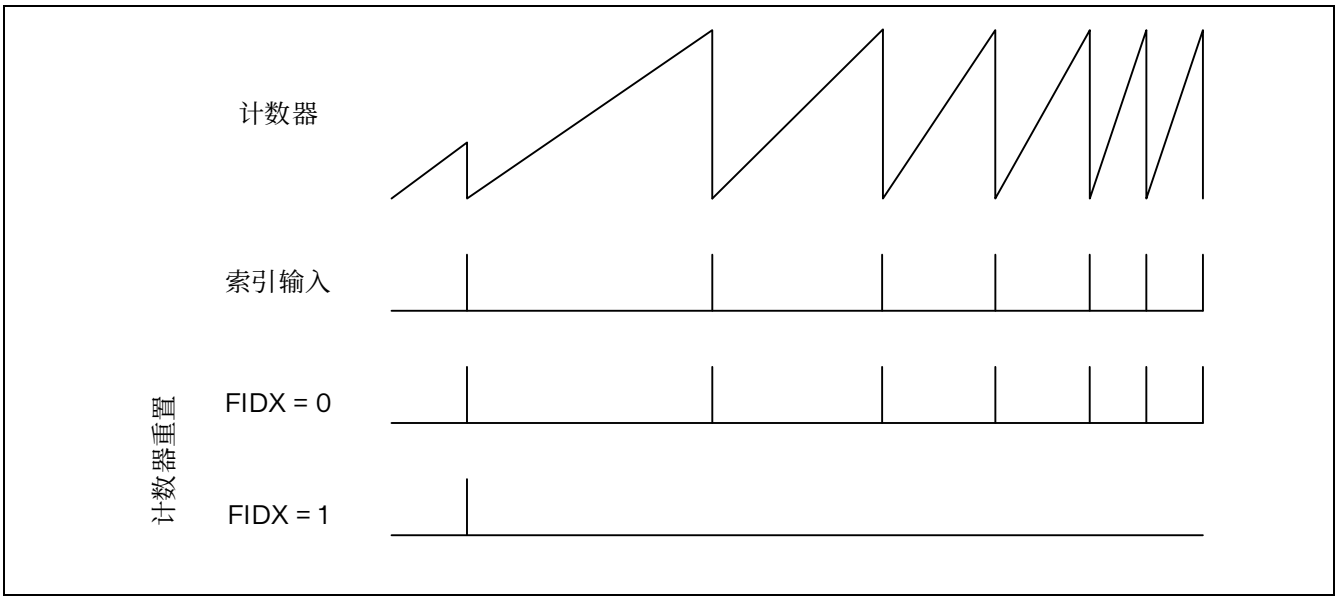


图 14.68 作为 FIDX 位设置的功能的计数器重置

非正交模式下的索引管理

下面两张图详细说明了当 SMS[3:0]位域等于 1010、1011、1100、1101 时，如何在方向时钟模式和时钟加方向模式下管理索引。

对于这两种模式，索引灵敏度通过 IPOS[0]位进行设置，具体如下：

- IPOS[0] = 0：在时钟低电平上检测索引
- IPOS[0] = 1：在时钟高电平上检测索引

IPOS[1]位不重要。

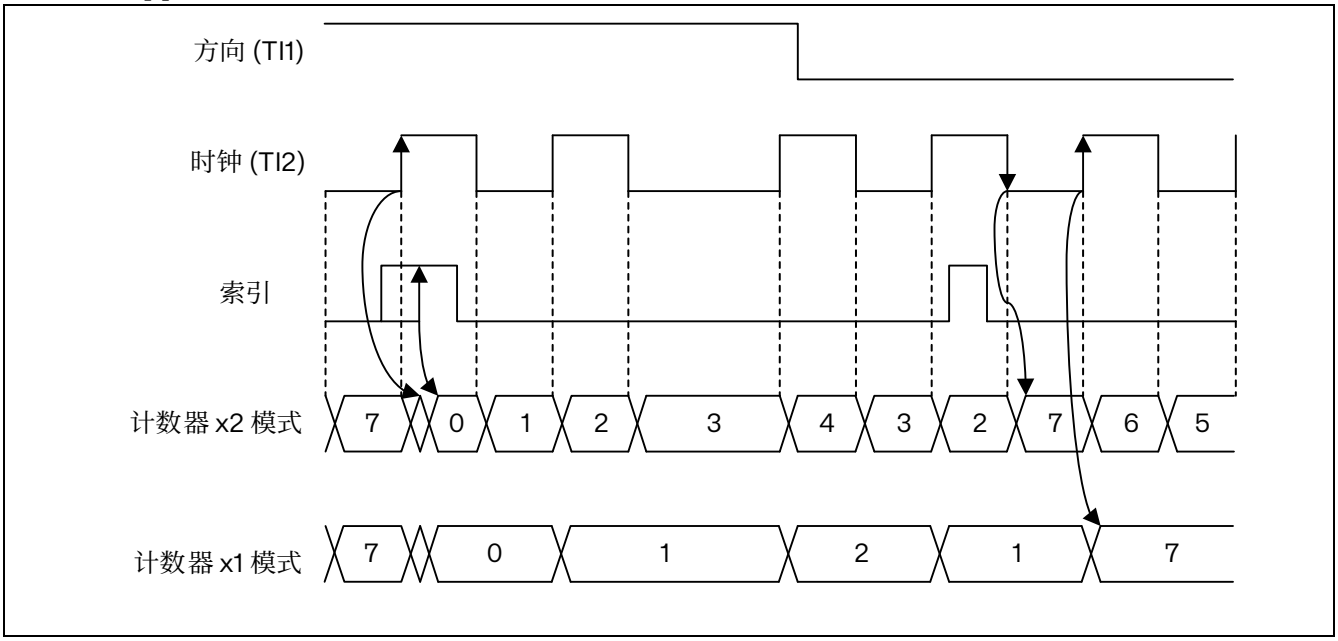


图 14.69 时钟+方向模式下的索引行为，IPOS[0] = 1

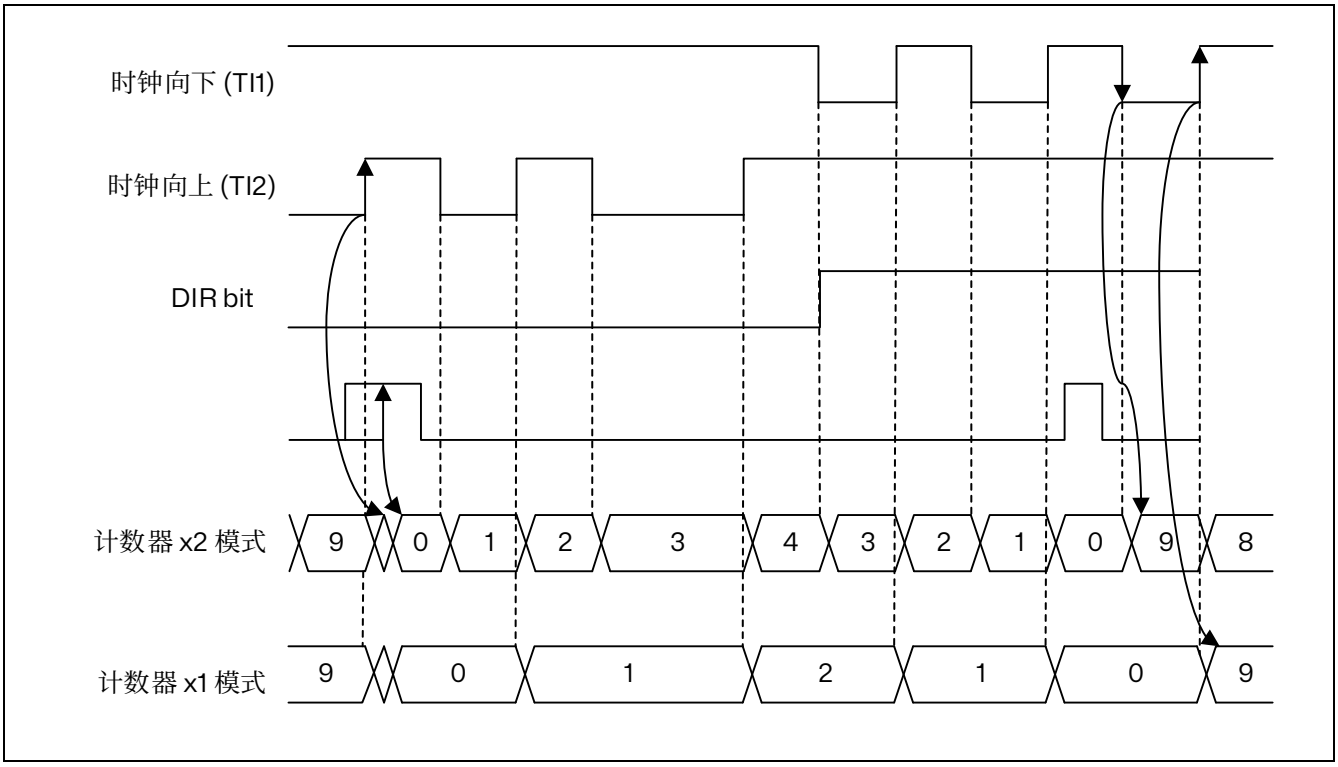


图 14.70 方向时钟模式下的索引行为，IPOS[0] = 1

编码器错误管理

对于具有 2 个正交信号的编码器配置，可以检测到转换错误。2 个输入的读数对应于一个 2 位灰度码，该灰度码可以表示为状态图，如以下图所示。期望单个比特立即更改。错误的转换会在 TIMx_SR 状态寄存器中设置 TERRF 中断标志。如果 TIMx_DIER 寄存器中的 TERRIE 位被设置，将产生转换错误中断。

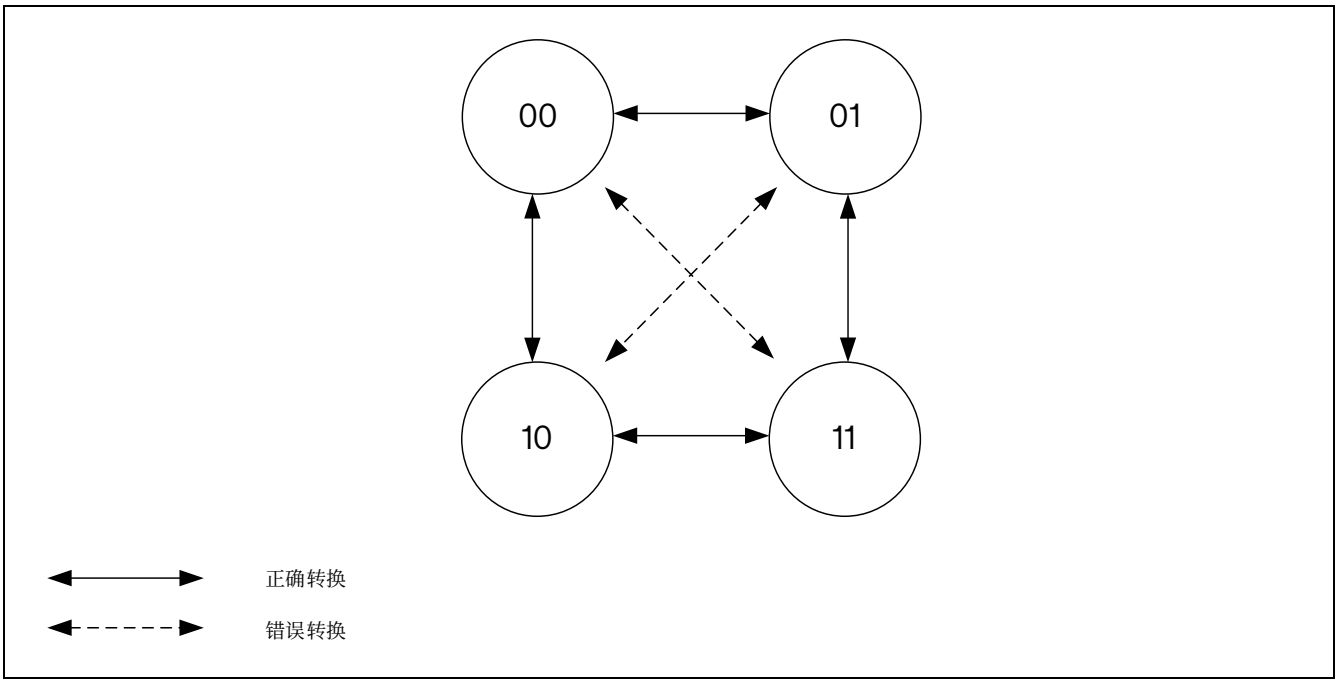


图 14.71 正交编码信号的状态图

对于具有索引信号的编码器，可以检测到导致每转脉冲过多的异常操作。一个编码器每转提供 N 个脉冲，则每转有 4xN 个计数。索引信号每 4xN 个时钟周期重置一次计数器。

如果计数器的值从 TIMx_ARR 增加到 0 或从 0 减少到 TIMxARR 的值而没有索引事件，则报告为索引位置错误。

可使用 TIMx_ARR 寄存器来编程溢出阈值。一个 1000 线的编码器导致计数器的值在 0 和 3999 之间（在 4 倍读取模式下）。必须通过设置 TIMx_ARR = 3999 + 1 = 4000 来编程溢出检测阈值。

错误断言被延迟到在递增计数时从 0 到 1 的转换。这是为了处理门控 A 和 B 模式中的窄索引脉冲，如下图所示。

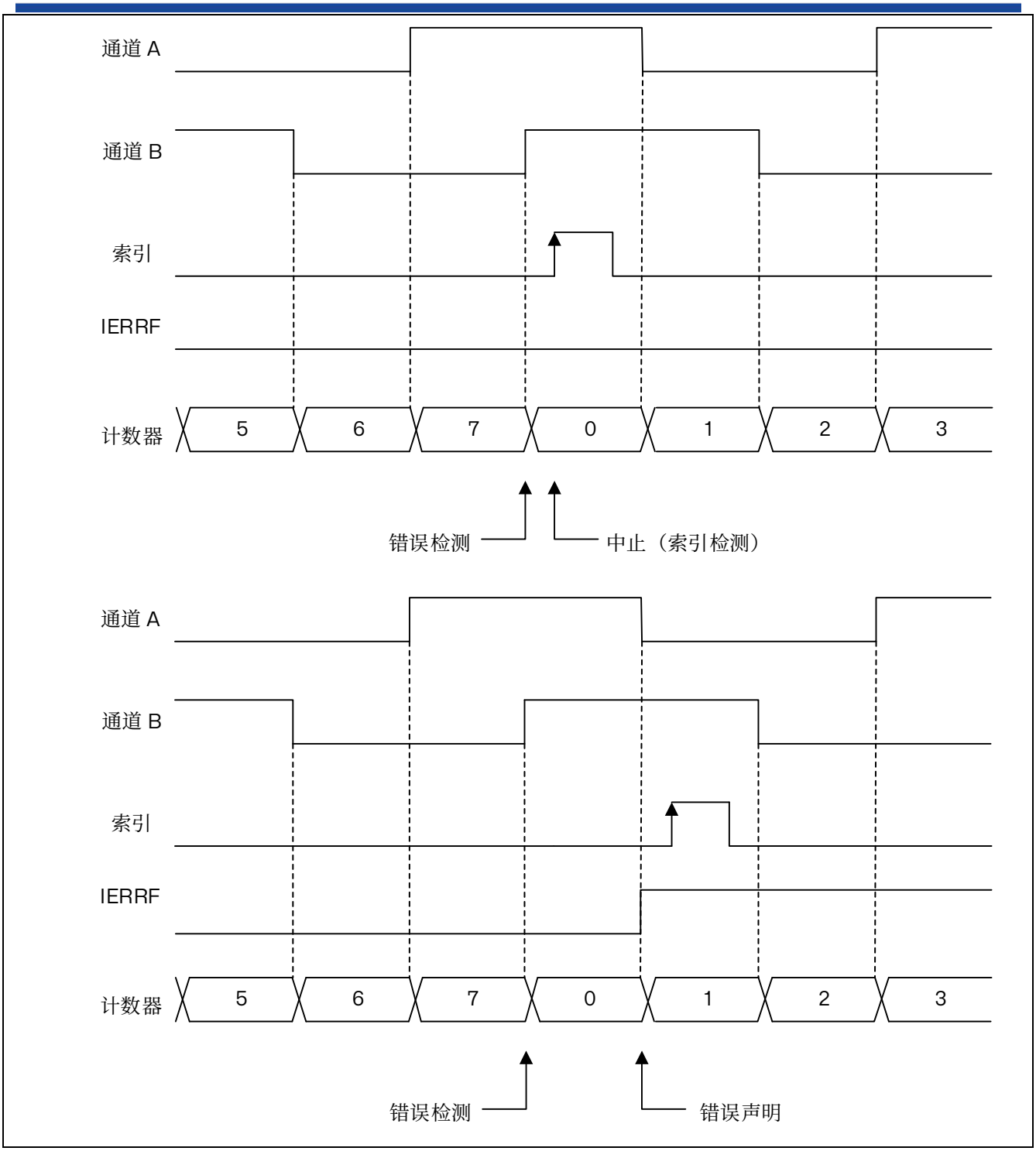


图 14.72 递增计数编码器错误检测

在递减计数模式下，检测是由从 1 到 0 的初步转换引起的。这是为了处理门控 A 和 B 模式中的窄索引脉冲，如下图所示，以避免编码器在索引检测后立即在 TIMx_ARR 和 0 之间抖动时出现任何错误检测。

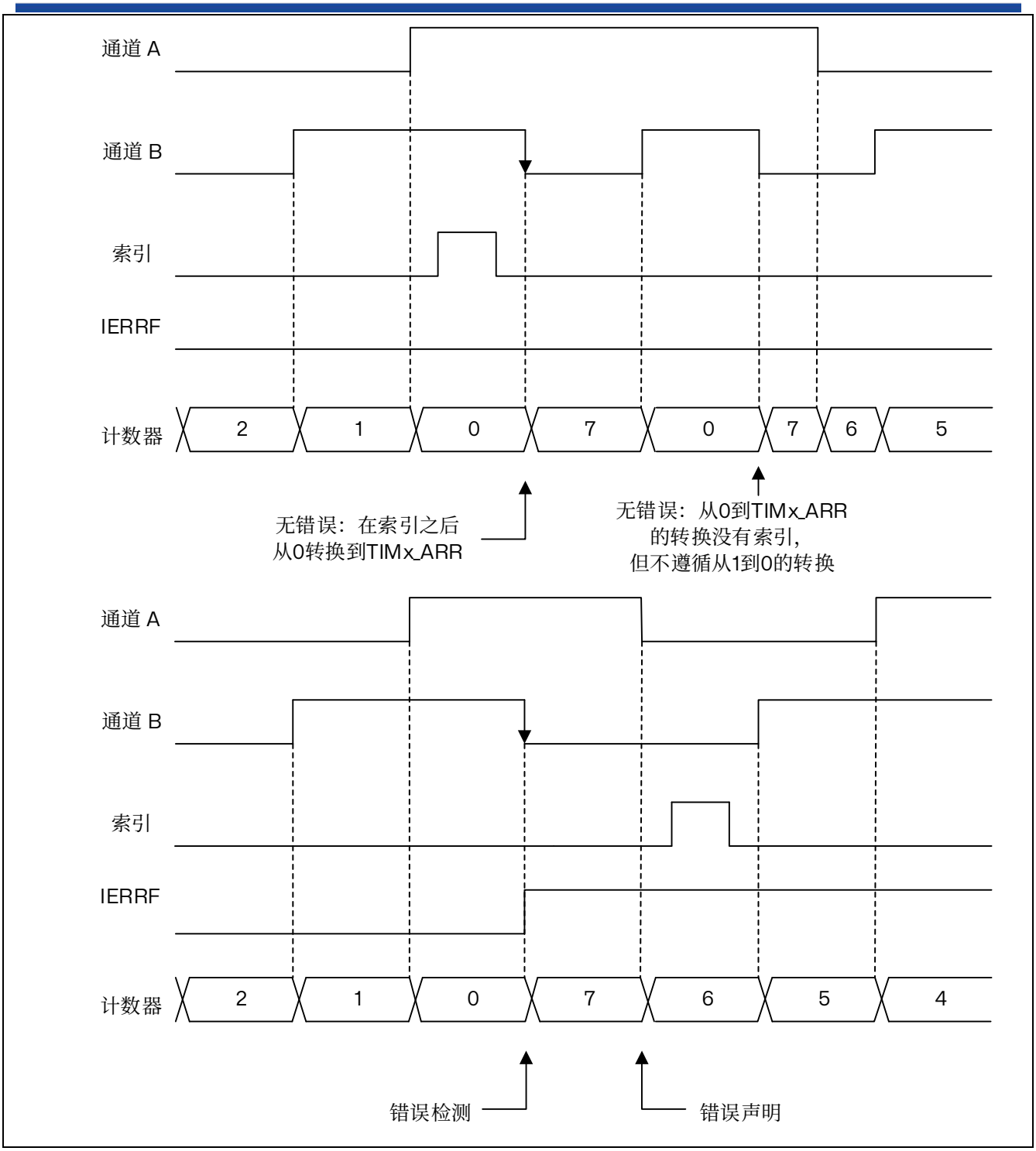


图 14.73 递减计数编码器错误检测

索引错误会在 TIMx_SR 状态寄存器中设置 IERRF 中断标志。如果 TIMx_DIER 寄存器中的 IERRIE 位被设置，将产生索引错误中断。

功能编码器中断

编码器模式下也可使用以下中断：

- 方向改变：在编码器模式下，任何计数方向的改变都会导致 TIMx_CR1 寄存器中的 DIR 位发生翻转。方向改变会设置 TIMx_SR 状态寄存器中的 DIRF 中断标志。如果 DIRIE 位在 TIMx_DIER 寄

存器中被设置，则产生方向改变中断。

- 索引事件：索引事件会设置 TIMx_SR 状态寄存器中的 IDXF 中断标志。如果 IDXIE 位在 TIMx_DIER 寄存器中被设置，则产生索引中断。

运行时更新编码器模式时从机模式选择预加载

在运行过程中，可能需要从一个编码器模式切换到另一个编码器模式。这通常在高速度下完成，以降低更新中断频率，从 x4 模式切换到 x2 模式再到 x1 模式，如下图所示。

为此，可以预加载 SMS[3:0]位。通过设置 TIMx_SMCR 寄存器中的 SMSPE 使能位可以使能此功能。从 SMS[3:0]预加载值转换到有效值可以由 TIMx_SMCR 寄存器中的 SMSPS 位选择触发。

- SMSPS = 0：当计数器递增时溢出并在计数器递减时下溢时，由更新事件(UEV)触发的转换。只有在索引被禁用时（位 IE = 0）才可以使用此模式。
- SMSPS = 1：由索引事件触发的转换。

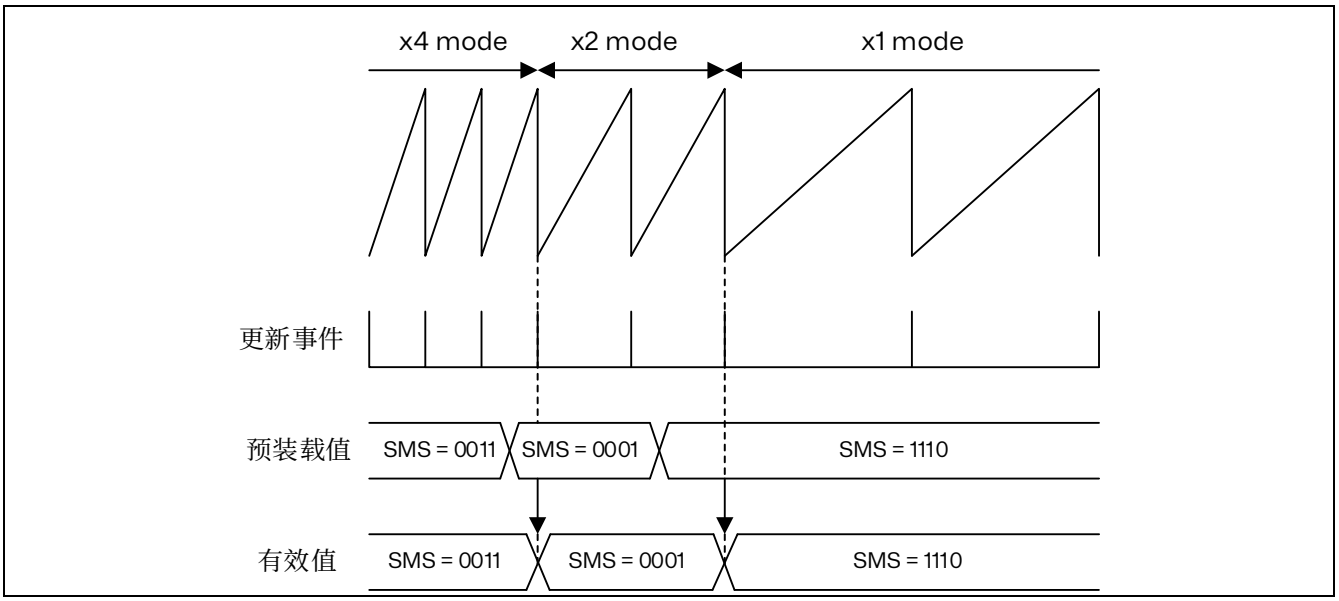


图 14.74 更新时带有预加载转移的编码器模式更改（SMSPS = 0）

编码器时钟输出

编码器模式的操作原理并不完全适合于高分辨率速度测量，在低速时，因为它需要相对较长的积分时间才能获得足够数量的时钟边缘和精确的测量。

在低速时，更好的解决方案是进行边到边的时钟周期测量。这可以通过使用从属定时器来实现。定时器可以在 tim_trgo 输出上输出编码器时钟信息。然后，从属定时器可以进行周期测量并为每个编码器时钟边缘提供速度信息。

通过将 TIMx_CR2 寄存器中的 MMS[3:0]位字段设置为 1000 可以使能此模式。对于以下 SMS[3:0]值有效：0001、0010、0011、1010、1011、1100、1101、1110、1111。任何可能导致意外行为的其他 SMS[3:0]代码是不被允许的。

14.4.19 方向位输出

可以将方向信号从定时器输出到 tim_oc3n 和 tim_oc4 输出信号（TIMx_CR1 寄存器中的 DIR 位的副本）。这是通过将 TIMx_CCMR2 寄存器中的 OC3M[3:0]或 OC4M[3:0]位字段设置为 1011 来实现的。

此功能可用于监测编码器模式下的计数方向（或旋转方向），或在中央对齐的 PWM 模式下有一个信号指示上升/下降相位。

14.4.20 UIF 位重映射

TIMx_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志 UIF 连续复制到定时器计数器寄存器的第 31 位（TIMxCNT[31]）。这允许以原子方式读取计数器的值和一个潜在的由 UIFCPY 标志指示的回滚条件。在特定情况下，它可以避免由背景任务（读取计数器）和中断（更新中断）之间的处理引起的竞争条件，从而简化计算。

UIF 和 UIFCPY 标志之间没有延迟。

14.4.21 定时器输入异或功能

TIMx_CR2 寄存器中的 TI1S 位，可将通道 1 的输入滤波器连接到异或门的输出，从而将 tim_ti1，tim_ti2，tim_ti3 这三个输入引脚组合在一起。

异或输出可与触发或输入捕获等所有定时器输入功能配合使用。

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。

14.4.22 定时器和外部触发的同步

TIMx 定时器以下列模式与外部触发实现同步：复位模式、门控模式、触发模式、复位+触发和门控+复位模式。

从模式：复位模式

当触发输入信号发生变化时，计数器及其预分频器可重新初始化。此外，如果 TIMx_CR1 寄存器中的 URS 位处于低电平，则会生成更新事件 UEV。然后，所有预装载寄存器（TIMx_ARR 和 TIMx_CCRx）都将更新。

在以下示例中，tim_ti1 输入上出现上升沿时，递增计数器清零：

- 将通道 1 配置为检测 tim_ti1 的上升沿。配置输入滤波时间（本例中不需要任何滤波，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S = 01。在 TIMx_CCER 寄存器中写入 CC1P=0 和 CC1NP=0，验证极性（仅检测上升沿）。
- 在 TIMx_SMCR 寄存器中写入 SMS=100，将定时器配置为复位模式。在 TIMx_SMCR 寄存器中写入 TS=101，选择 tim_ti1 作为输入源。
- 在 TIMx_CR1 寄存器中写入 CEN=1，启动计数器。

计数器使用内部时钟计数，然后正常运转，直到出现 tim_ti1 上升沿。当 tim_ti1 出现上升沿时，计数器清零，然后重新从 0 开始计数。同时，触发标志（TIMx_SR 寄存器中的 TIF 位）置 1，使能中断后，还可发送中断请求（取决于 TIMx_DIER 寄存器中的 TIE 和 TDE 位）。

下图显示了自动重载寄存器 TIMx_ARR=0x36 时的相关行为。tim_ti1 的上升沿与实际计数器复位之间的延迟是由于 tim_ti1 输入的重新同步电路引起的。

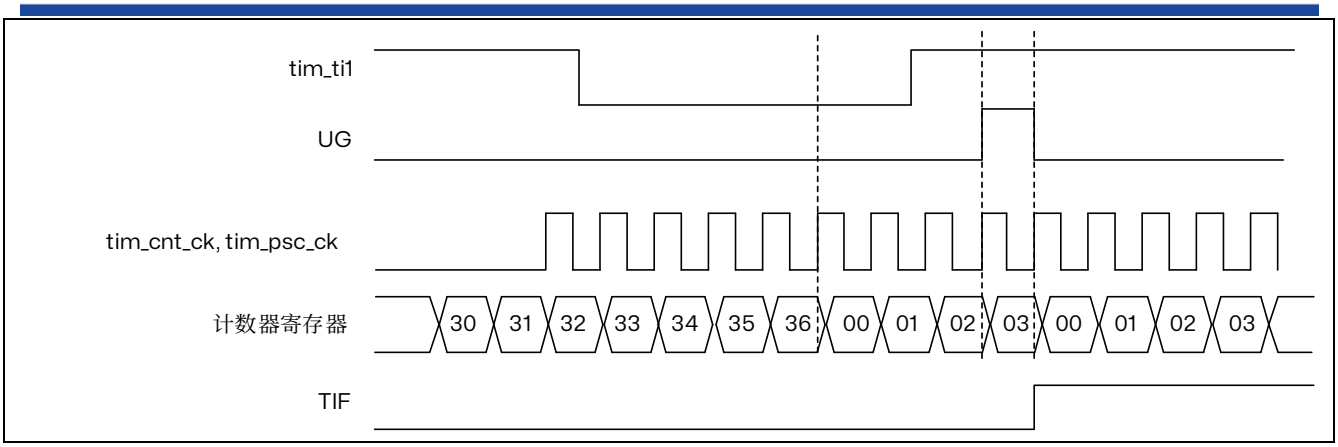


图 14.75 复位模式下的控制电路

从模式：门控模式

输入信号的电平可用来使能计数器。

在以下示例中，递增计数器仅在 tim_ti1 输入为低电平时计数：

- 将通道 1 配置为检测 tim_ti1 上的低电平。配置输入滤波时间（本例中不需要任何滤波，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S=01。在 TIMx_CCER 寄存器中写入 CC1P=1 和 CC1NP=0，以确定极性（仅检测低电平）。
- 在 TIMx_SMCR 寄存器中写入 SMS=101，将定时器配置为门控模式。在 TIMx_SMCR 寄存器中写入 TS=00101，选择 tim_ti1 作为输入源。
- 在 TIMx_CR1 寄存器中写入 CEN=1，使能计数器（在门控模式下，如果 CEN=0，则无论触发输入电平如何，计数器都不启动）。

只要 tim_ti1 为低电平，计数器就开始根据内部时钟计数，直到 tim_ti1 变为高电平时停止计数。计数器启动或停止时，TIMx_SR 寄存器中的 TIF 标志都会置 1。

tim_ti1 的上升沿与实际计数器停止之间的延迟是由于 tim_ti1 输入的重新同步电路引起的。

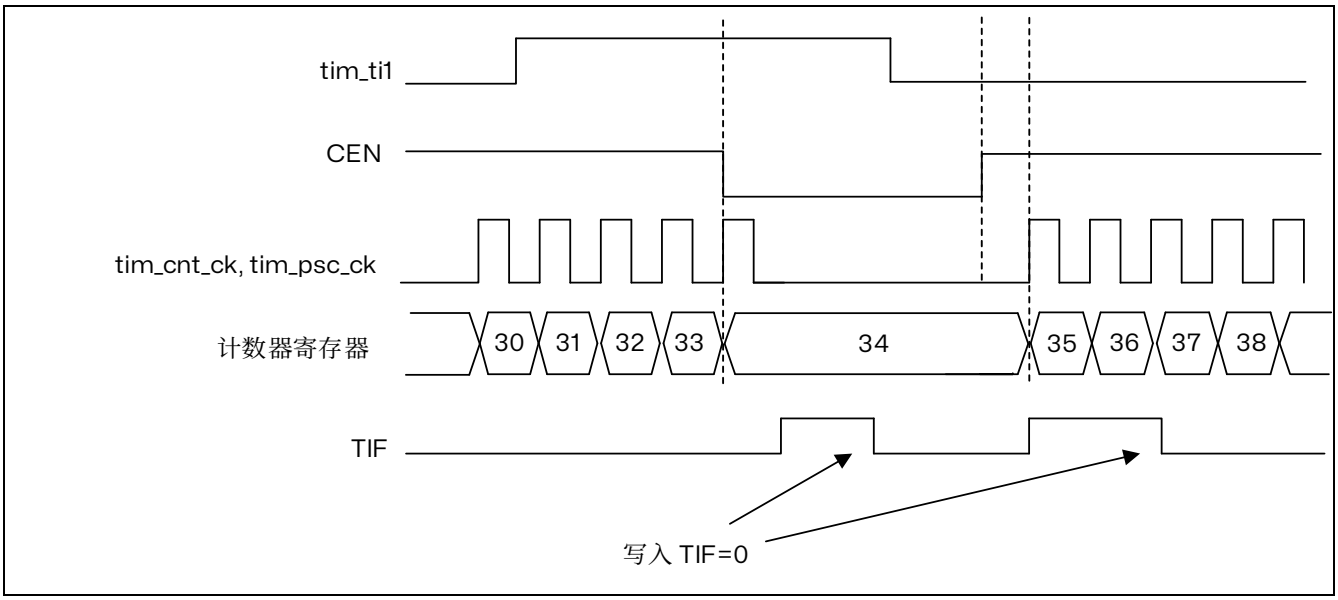


图 14.76 门控模式下的控制电路

注：由于门控模式作用于电平而非边沿，因此在门控模式下，“CCxP=CCxNP=1”（同时检测上升沿和下降沿）不发挥任何作用。

从模式：触发模式

所选输入上发生某一事件时可以启动计数器。
在以下示例中，tim_ti2 输入上出现上升沿时，递增计数器启动：

- 将通道 2 配置为检测 TI2 上的上升沿。配置输入滤波时间（本例中不需要任何滤波，因此保持 IC2F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC2S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC2S=01。在 TIMx_CCER 寄存器中写入 CC2P=1 和 CC2NP=0，以确定极性（仅检测低电平）。
- 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIMx_SMCR 寄存器中写入 TS=00110，选择 tim_ti2 作为输入源。

当 tim_ti2 出现上升沿时，计数器开始根据内部时钟计数，并且 TIF 标志置 1。
tim_ti2 的上升沿与实际计数器启动之间的延迟是由于 tim_ti2 输入的重新同步电路引起的。

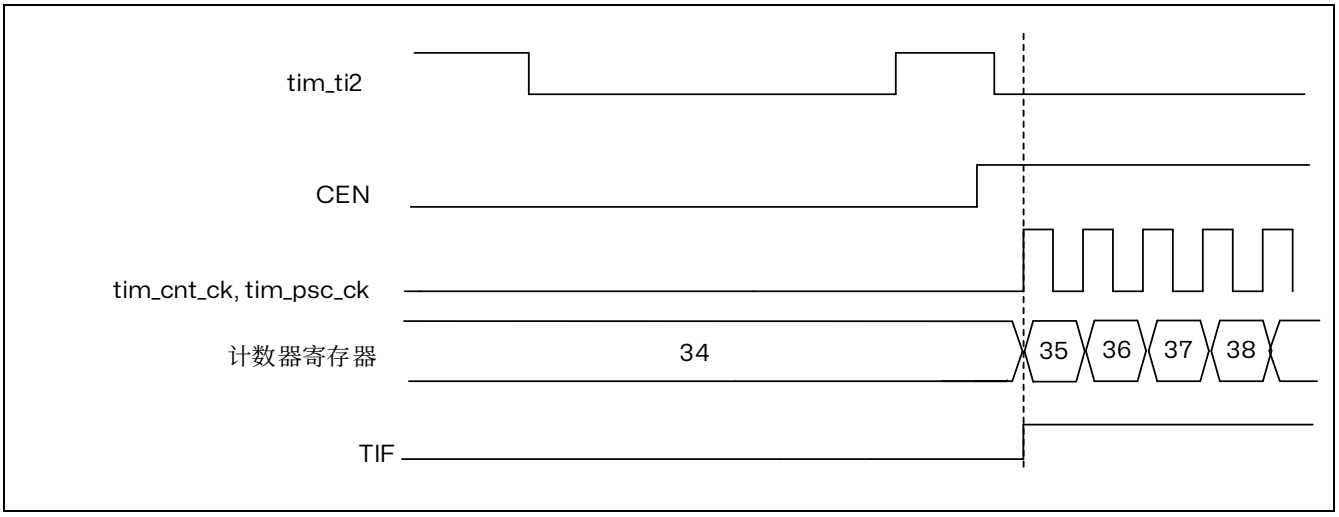


图 14.77 触发器模式下的控制电路

对实时更新编码器模式的从模式选择预装载

SMS[3:0]位可以预加载。这是通过在 TIMx_SMCR 寄存器中设置 SMSPE 使能位来实现的。从 SMS[3:0]预加载到有效值的触发器是计数器溢出时发生的更新事件（UEV）。

从模式 - 组合重置 + 触发模式

在这种情况下，所选触发输入（tim_trgi）的上升沿会重新初始化计数器，
此模式用于单脉冲模式。

从模式 - 组合重置 + 门控模式

当触发输入（tim_trgi）为高时，计数器使能。计数器停止并一旦触发器变为低，计数器就会重置。计数器的启动和停止都是受控的。
此模式允许检测超出范围的 PWM 信号（占空比超过最大期望值）。

从模式：外部时钟模式 2＋触发模式

外部时钟模式 2 可与另一种从模式（外部时钟模式 1 和编码器模式除外）结合使用。这种情况下，tim_etr_in 信号用作外部时钟输入，在复位模式、门控模式或触发模式下工作时，可选择另一个输入作为触发输入。不建议通过 TIMx_SMCR 寄存器中的 TS 位来选择 tim_etr_in 作为 tim_trgi。

在以下示例中，只要 tim_ti1 出现上升沿，递增计数器即会在 tim_etr_in 信号的每个上升沿处递增：

1. 通过对 TIMx_SMCR 寄存器进行如下编程，配置外部触发输入电路：
   - ETF = 0000：无滤波器
   - ETPS = 00：禁止预分频器
   - ETP = 0：检测 tim_etr_in 的上升沿，并写入 ECE=1，以使能外部时钟模式 2。
2. 如下配置通道 1，以检测 TI 的上升沿：
   - IC1F = 0000：无滤波器
   - 由于捕获预分频器不用于触发操作，因此无需对其进行配置。
   - TIMx_CCMR1 寄存器中的 CC1S = 01，只选择输入捕获源
   - TIMx_CCER 寄存器中的 CC1P = 0 和 CC1NP=0，以确定极性（仅检测上升沿）。
3. 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIMx_SMCR 寄存器中写入 TS=00101，选择 tim_ti1 作为输入源。

tim_ti1 出现上升沿时将使能计数器并且 TIF 标志置 1。然后计数器在 tim_etr_in 出现上升沿时计数。tim_etr_in 信号的上升沿与实际计数器复位之间的延迟是由于 tim_etrp 输入的重新同步电路引起的。

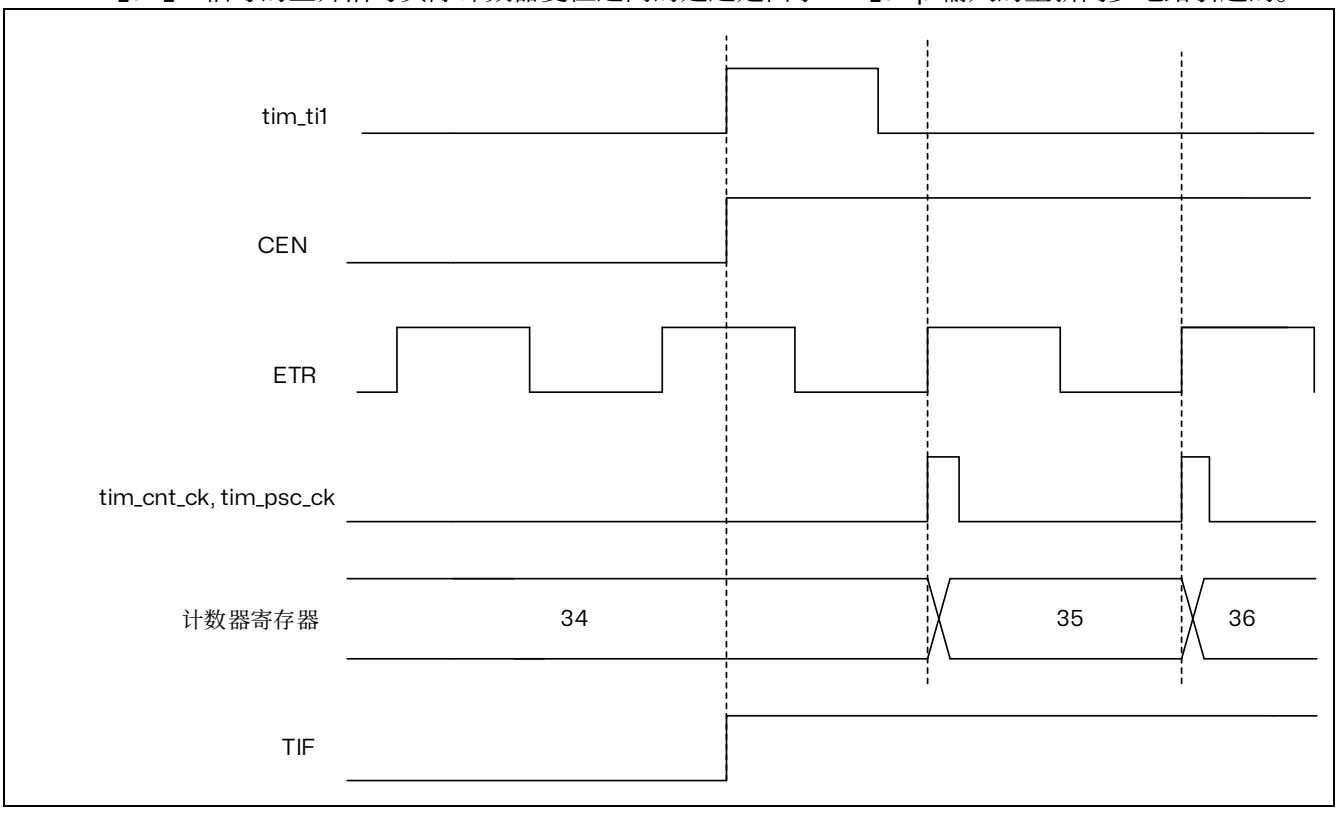


图 14.78 外部时钟模式 2＋触发模式下的控制电路

14.4.23 定时器同步

TIMx 定时器从内部连接在一起，以实现定时器同步或级联。当某个定时器配置为主模式时，可对另一个配置为从模式的定时器的计数器执行复位、启动、停止操作或为其提供时钟。下图显示了触发选择和主模式选择模块的概况。

下图简要介绍了触发选择和主模式选择框图。

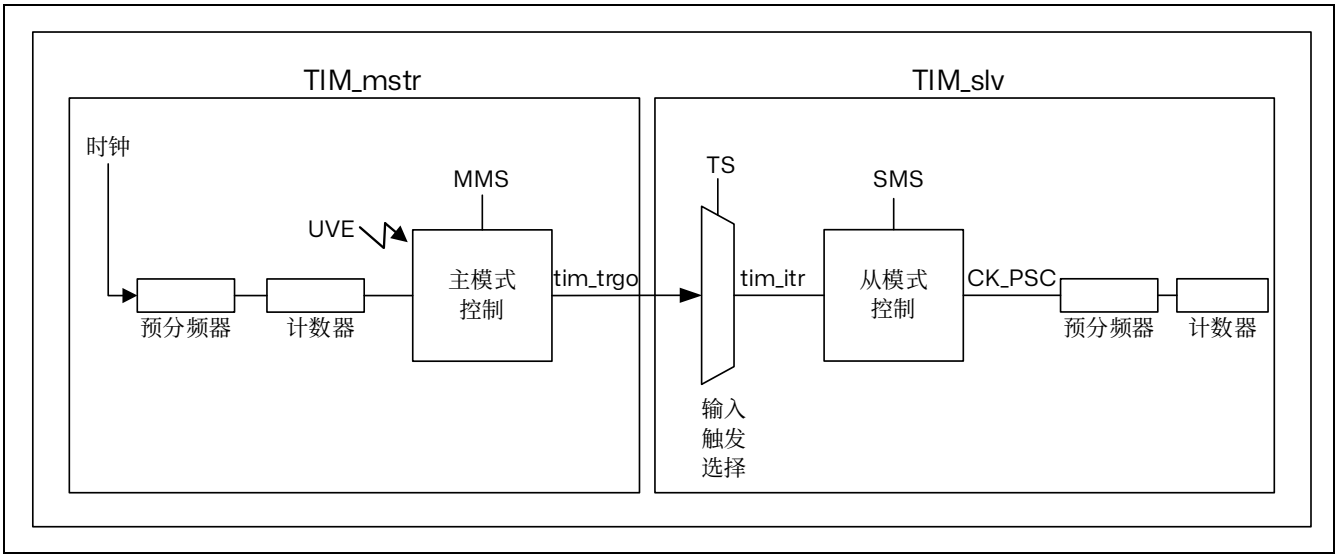


图 14.79 主/从定时器的例子

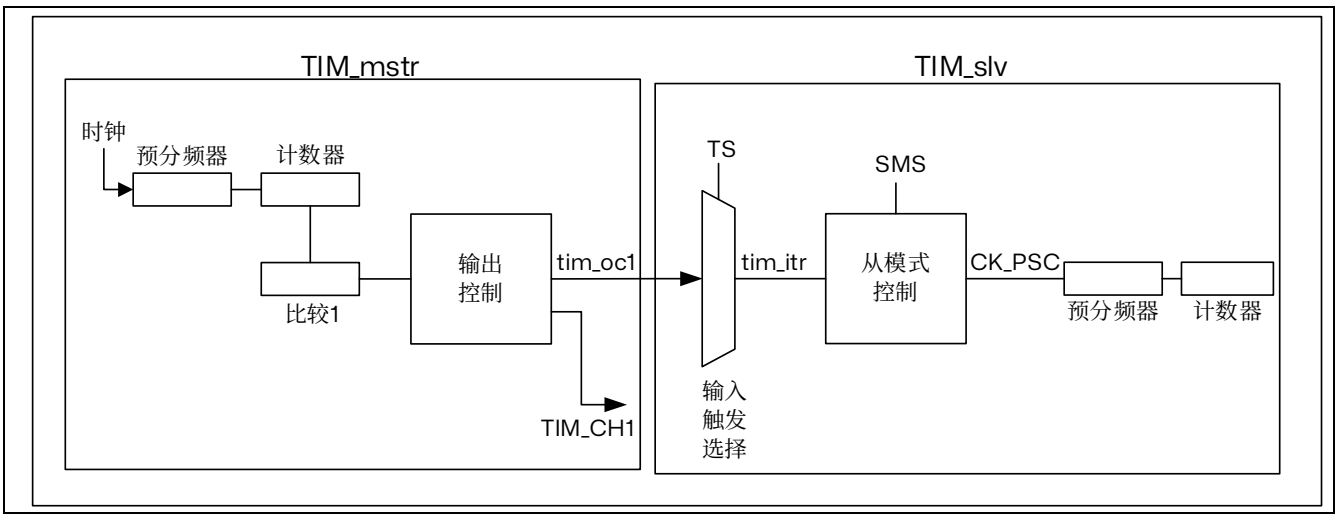


图 14.80 仅定时器通道 1 主/从连接示例

*注：只有一个通道的计时器（参见上图）不具有主模式。然而，*tim_oc1 *输出信号可以作为从属定时器的触发信号（参见*TIMx *内部触发连接表：*TIM2/TIM3 *引脚和内部信号）。必须将*tim_oc1 *信号脉冲宽度编程为至少为目的地定时器的*2 *个时钟周期，以确保从属定时器检测到触发信号。例如，如果目的地定时器*tim_ker_ck *时钟比源定时器慢*4 *倍，则*OC1 *脉冲宽度必须是*8 *个时钟周期。*

*注：主从同步功能时，需要将*TRGI *寄存器的*SLAVE_SYNC *位置*1。

使用一个定时器作为另一个定时器的预分频器

例如，可以将 TIM_mstr 配置为 TIM_slv 的预分频器。请参见上图。为此：

1. 将 TIM_mstr 配置为主模式，以便每次发生更新事件 UEV 时都输出一个周期性触发信号。如果在 TIM_mstr_CR2 寄存器中写入 MMS=010，则每次生成更新事件时，tim_trgo 都会输出一个上升沿。
2. 要将 TIM_mstr 的 tim_trgo 输出连接到 TIM_slv，必须将 TIM_slv 配置为从模式，使用 ITR0 作为内部触发。通过 TIM_slv_SMCR 寄存器中的 TS 位（写入 TS=000）可对此进行选择。

3. 然后将从模式控制器设为外部时钟模式 1（在 TIM_slv_SMCR 寄存器中写入 SMS=111）。这样一来，TIM_slv 的时钟将由 TIM_mstr 周期性触发信号的上升沿（与 TIM_mstr 的计数器上溢对应）提供。
4. 最后必须通过将这两个定时器的相应 CEN 位（TIMx_CR1 寄存器）置 1 同时使能二者。

注：如果选择 TIM_mstr 的 tim_ocx 信号作为触发输出（MMS=1xx），该信号的上升沿将用于驱动 TIM_slv 的计数器。

使用一个定时器使能另一个定时器

本例中通过 TIM_mstr 的输出比较 1 来使能 TIM_slv。相关连接图，参考下图。仅当 TIM_mstr 的 tim_oc1ref 为高电平时，TIM_slv 才根据分频后的内部时钟进行计数。两个计数器的时钟频率都基于 tim_ker_ck 通过预分频器执行 3 分频（$f_{tim_cnt_ck} = f_{tim_ker_ck}/3$）

1. 将 TIM_mstr 配置为主模式，发送其输出比较 1 参考信号（tim_oc1ref）作为触发输出（TIM_mstr_CR2 寄存器中的 MMS=100）。
2. 配置 TIM_mstr 的 tim_oc1ref 波形（TIM_mstr_CCMR1 寄存器）。
3. 配置 TIM_slv 以接收来自 TIM_mstr 的输入触发（TIM_slv_SMCR 寄存器中的 TS=00010）。
4. 将 TIM_slv 配置为门控模式（TIM_slv_SMCR 寄存器中的 SMS=101）。
5. 通过向 CEN 位（TIM_slv_CR1 寄存器）写入“1”使能 TIM_slv。
6. 通过向 CEN 位（TIM_mstr_CR1 寄存器）写入“1”启动 TIM_mstr。

注：计数器 2 的时钟与计数器 1 不同步，此模式仅影响 TIM_slv 的计数器使能信号。

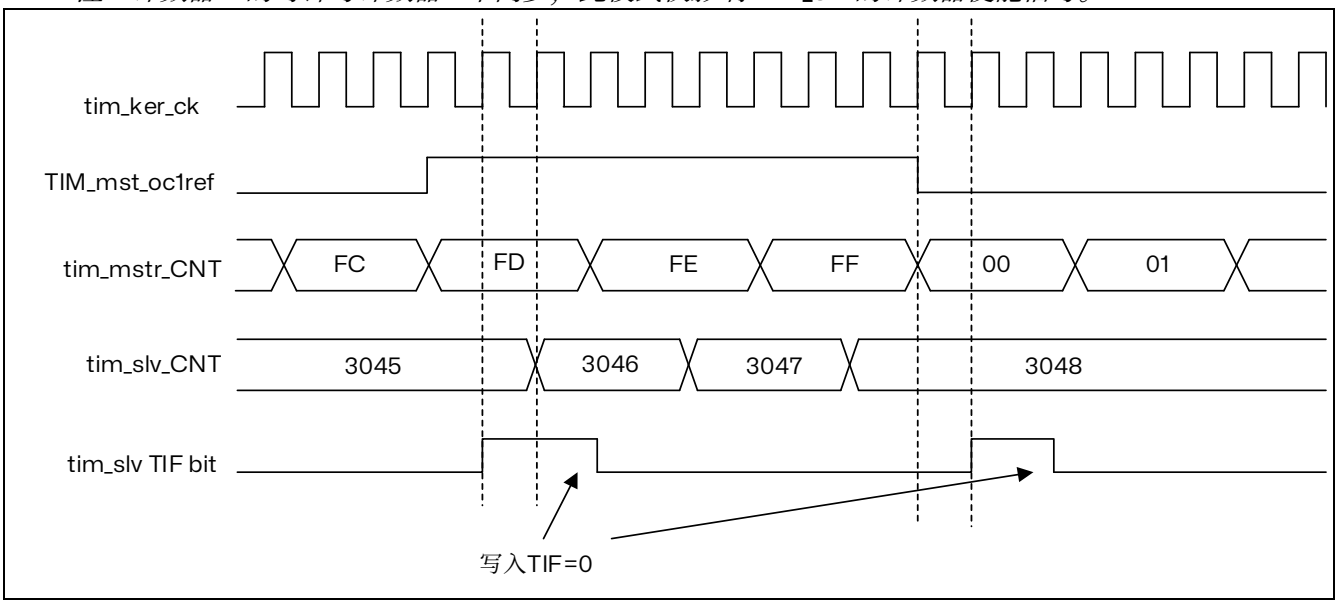


图 14.81 TIM_mstr 的 tim_oc1ref 控制 TIM_slv

在上图示例中，TIM_slv 的计数器和预分频器在启动前未进行初始化。因此从各自的当前值开始计数。启动 TIM_mstr 之前，通过复位这两个定时器可以从指定值开始计数。这样便可以在定时器计数器中写入所需的任意值。两个定时器都可通过软件使用 TIMx_EGR 寄存器中的 UG 位轻松复位。

在下一示例中，TIM_mstr 与 TIM_slv 同步。TIM_mstr 为主模式，从 0 开始计数。TIM_slv 为从模式，从 0xE7 开始计数。两个定时器的预分频比相同。在 TIM_mstr_CR1 寄存器中通过向 CEN 位写入“0”来禁止 TIM_mstr 时，TIM_slv 将停止：

1. 将 TIM_mstr 配置为主模式，发送其输出比较 1 参考信号（tim_oc1ref）作为触发输出（TIM_mstr_CR2 寄存器中的 MMS=100）。

2. 配置 TIM_mstr 的 tim_oc1ref（TIM_mstr_CCMR1 寄存器）。
3. 配置 TIM_slv 以接收来自 TIM_mstr 的输入触发（TIM_slv_SMCR 寄存器中的 TS=00010）。
4. 将 TIM_slv 配置为门控模式（TIM_slv_SMCR 寄存器中的 SMS=101）。
5. 通过向 UG 位（TIM_mstr_EGR 寄存器）写入“1”复位 TIM_mstr。
6. 通过向 UG 位（TIM_slv_EGR 寄存器）写入“1”复位 TIM_slv。
7. 通过在 TIM_slv 的计数器（TIM_slv_CNT）中写入“0xE7”使 TIM_slv 初始化为 0xE7。
8. 通过向 CEN 位（TIM_slv_CR1 寄存器）写入“1”使能 TIM_slv。
9. 通过向 CEN 位（TIM_mstr_CR1 寄存器）写入“1”启动 TIM_mstr。
10. 通过向 CEN 位（TIM_mstr_CR1 寄存器）写入“0”停止 TIM_mstr。

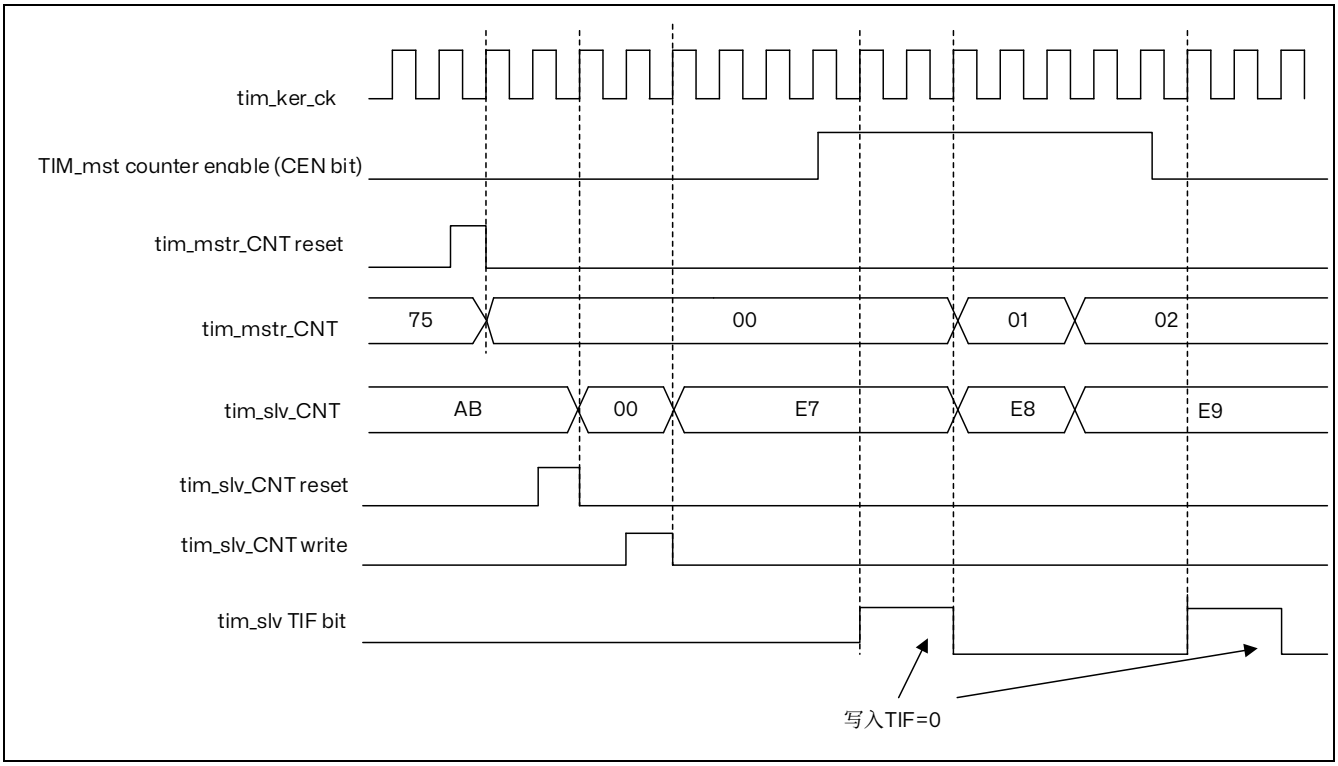


图 14.82 通过使能 TIM_mstr 可以控制 TIM_slv

使用一个定时器去启动另一个定时器

本例中使用 TIM_mstr 的更新事件使能 TIM_slv。相关连接图，请参见下图。只要 TIM_mstr 生成更新事件，TIM_slv 便根据分频后的内部时钟从当前值（可以不为 0）开始计数。TIM_slv 收到触发信号时，其 CEN 位自动置 1，并且计数器开始计数，直到向 TIM_slv_CR1 寄存器的 CEN 位写入“0”后停止计数。两个计数器的时钟频率都基于 CK_INT 通过预分频器执行 3 分频（$f_{tim_cnt_ck} = f_{tim_ker_ck}/3$）。

1. 将 TIM_mstr 配置为主模式，发送其更新事件（UEV）作为触发输出（TIM_mstr_CR2 寄存器中的 MMS=010）。
2. 配置 TIM_mstr 的周期（TIM_mstr_ARR 寄存器）。
3. 配置 TIM_slv 以接收来自 TIM_mstr 的输入触发（TIM_slv_SMCR 寄存器中的 TS=00010）。
4. 将 TIM_slv 配置为触发模式（TIM_slv_SMCR 寄存器中的 SMS=110）。
5. 通过向 CEN 位（TIM_mstr_CR1 寄存器）写入“1”启动 TIM_mstr。

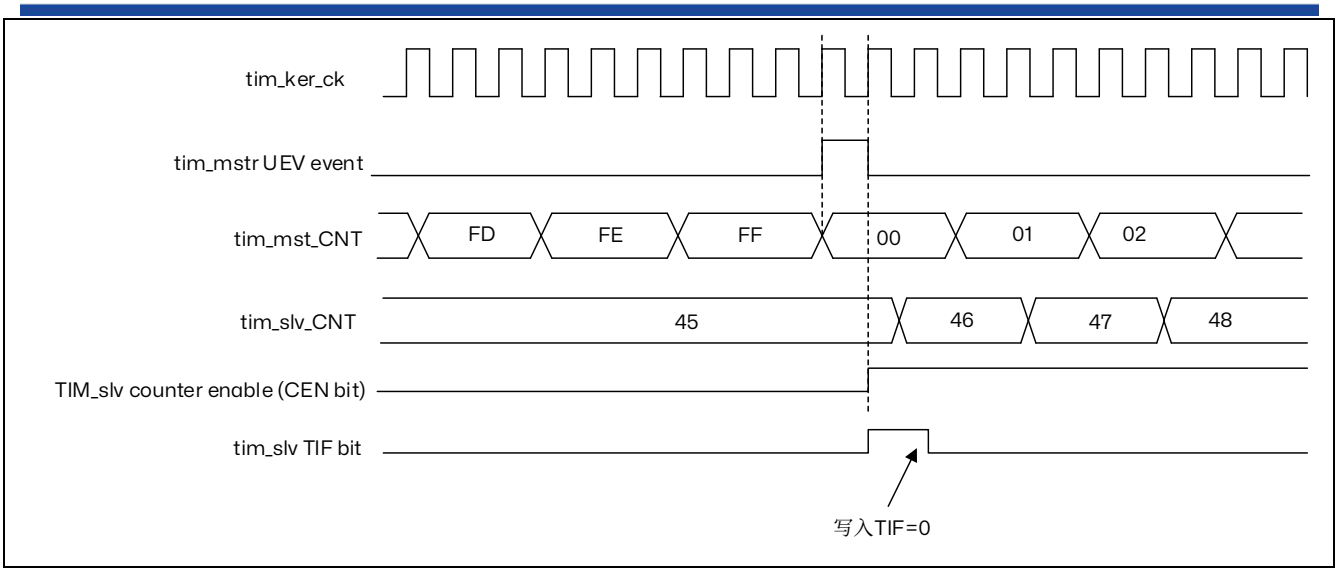


图 14.83 使用 TIM_mstr 的更新触发 TIM_slv

如上述示例所示，用户可以在开始计数之前初始化两个计数器。下图显示了与上图具有相同配置，只不过处于触发模式（TIM_slv_SMCR 寄存器中的 SMS=110）而非门控模式的计数行为。

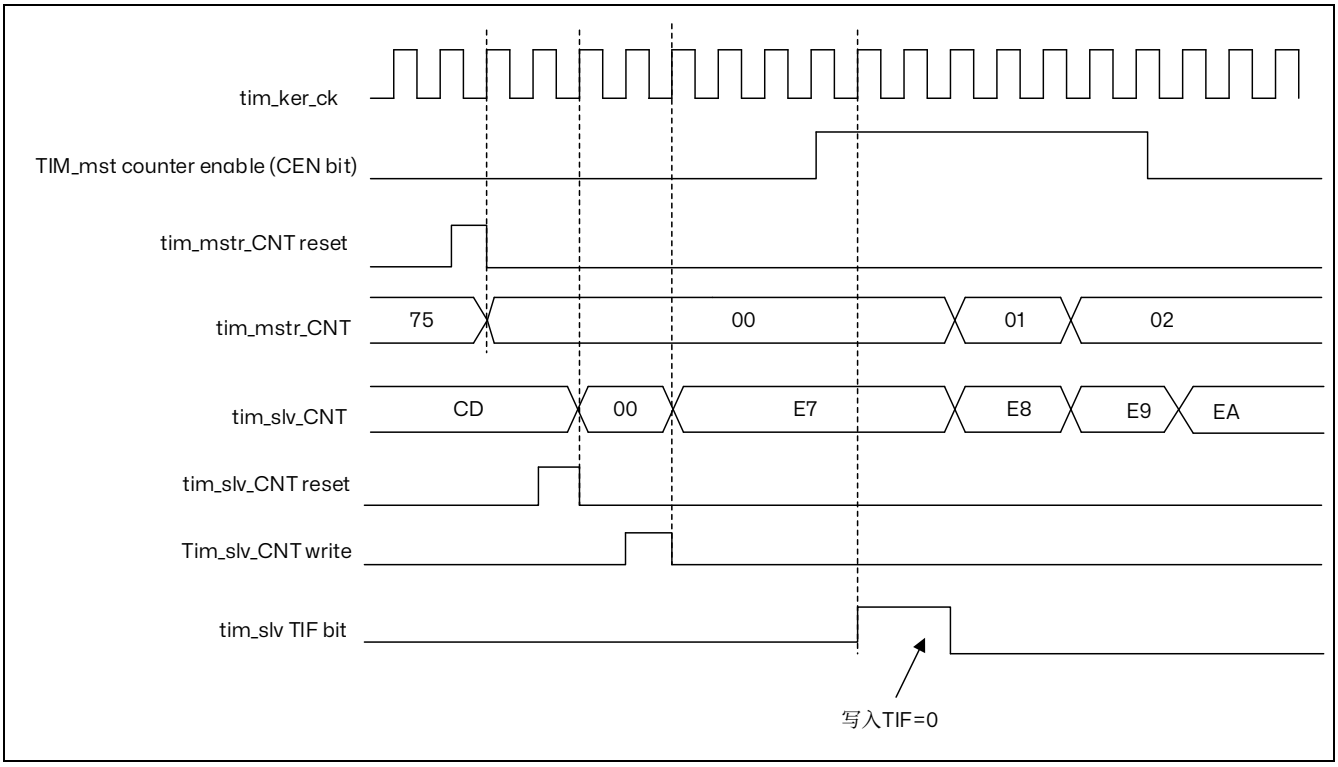


图 14.84 利用 TIM_mstr 的使能触发 TIM_slv

使用一个外部触发同步地启动 2 个定时器

本例中，TIM_mstr 的 tim_ti1 输入出现上升沿时使能 TIM_mstr，使能 TIM_mstr 的同时使能 TIM_slv。要确保两个计数器对齐，TIM_mstr 必须配置为主/从模式（对应的 tim_ti1 为从，对应 TIM_slv 为主）：

1. 将 TIM_mstr 配置为主模式，发送其使能信号作为触发输出（TIM_mstr_CR2 寄存器中

MMS=001）。

2. 将 TIM_mstr 配置为从模式以接收来自 tim_ti1 的输入触发（TIM_mstr_SMCR 寄存器中的 TS=00100）。
3. 将 TIM_mstr 配置为触发模式（TIM_mstrSMCR 寄存器中的 SMS=110）。
4. 通过写入 MSM=1（TIM_mstr_SMCR 寄存器）将 TIM_mstr 配置为主/从模式。
5. 配置 TIM_slv 以接收来自 TIM_mstr 的输入触发（TIM_slv_SMCR 寄存器中的 TS=00000）。
6. 将 TIM_slv 配置为触发模式（TIM_slv_SMCR 寄存器中的 SMS=110）。

当 tim_ti1（TIM_mstr）出现上升沿时，两个计数器开始根据内部时钟同步计数，并且两个 TIF 标志都置 1。

注：本例中，两个定时器都在启动之前进行了初始化（通过将各自的 UG 位置 1）。两个计数器都从 0 开始计数，但可以通过对任意一个计数器寄存器（TIMx_CNT）进行写操作，在二者之间轻松插入一个偏移量。可注意到主/从模式在 TIM_mstr 的 CNT_EN 与 CK_PSC 之间产生了延迟。

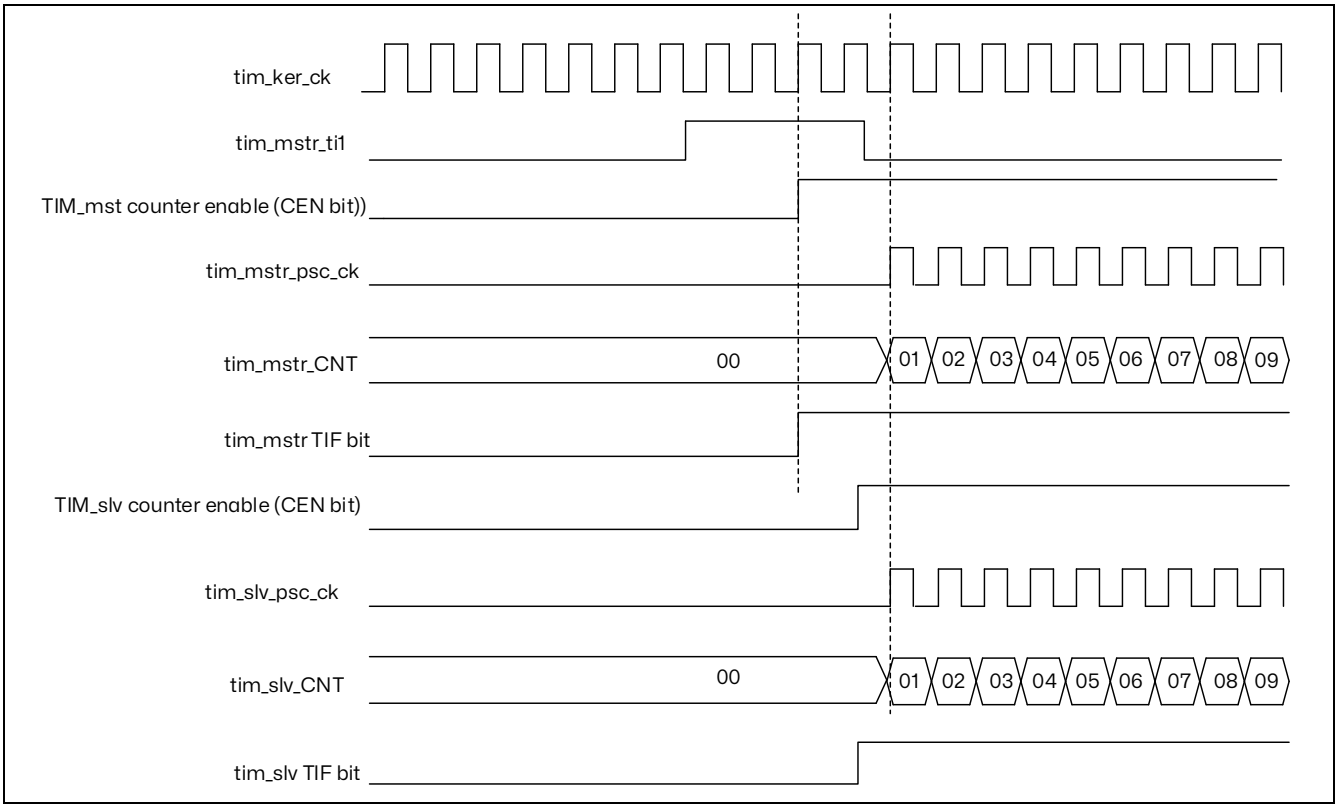


图 14.85 使用 TIM_mstr 的 tim_ti1 输入触发 TIM_mstr 和 TIM_slv

14.4.24 ADC 触发

定时器可以用各种内部信号（例如重置、使能或比较事件）生成 ADC 触发事件。

注意：接收 tim_trgo *或* tim_trgo2 *信号的从属外设（定时器、ADC 等）的时钟必须在从主定时器接收事件之前使能，并且时钟频率（预分频器）在从主定时器接收触发器时不能更改。*

14.4.25 调试模式

当微控制器进入调试模式时（内核停止），TIMx 计数器会根据 DBGMCU 模块中的 DBG_TIMx_STOP 配置位选择继续正常工作或者停止工作。

有关详细信息，请参阅 section Debug support（DBG）。

14.5 TIM2/TIM3 低功耗模式

表 14.15 低功耗模式对 TIM2/TIM3 的影响

模式	描述
睡眠	无影响，外设是运行的。中断会导致设备退出睡眠模式。
停机	定时器操作被停止并保留其寄存器内容。无法生成中断。
待机	定时器被断电并且退出待机模式后必须重新初始化。

14.6 TIM2/TIM3 中断

TIM2/TIM3 可以生成多个中断，如下表所示。

表 14.16 中断请求

中断缩写	中断事件	事件标志	使能控制位	中断清除方法	从睡眠模式退出	从停机和待机退出
TIM_UP	更新	UIF	UIE	UIF 写 0	是	否
TIM_CC	捕获/比较 1	CC1IF	CC1IE	CC1IF 写 0	是	否
	捕获/比较 2	CC2IF	CC2IE	CC2IF 写 0	是	否
	捕获/比较 3	CC3IF	CC3IE	CC3IF 写 0	是	否
	捕获/比较 4	CC4IF	CC4IE	CC4IF 写 0	是	否
TIM_TRG	触发	TIF	TIE	TIF 写 0	是	否
TIM_DIR_IDX	索引	IDXF	IDXIE	IDXF 写 0	是	否
	方向	DIRF	DIRIE	DIRF 写 0	是	否
TIM_IERR	索引错误	IERRF	IERRIE	IERRF 写 0	是	否
TIM_TER	转换错误	TERRF	TERRIE	TERRF 写 0	是	否

14.7 TIMx 寄存器

14.7.1 TIMx 控制寄存器 1（TIMx_CR1）（x=2/3）

偏移地址：0x00

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			DITHEN	UIFREMAP	Res	CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
			rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:13 保留，必须保持为复位值。

Bit 12 **DITHEN**：使能抖动（Dithering enable）

0：抖动禁止

1：抖动使能

*注：*DITHEN *位只能在*CEN *位复位时修改*

Bit 11 **UIFREMAP**：UIF 状态位重映射（UIF status bit remapping）

0：无重映射。UIF 状态位未复制到 TIMx_CNT 寄存器位 31
1：重映射使能。UIF 状态位复制到 TIMx_CNT 寄存器位 31

Bits 10 保留，必须保持为复位值。

Bits 9:8 **CKD[1:0]**：时钟分频（Clock division）

此位域指示定时器时钟（tim_ker_ck）频率以及数字滤波器（tim_etr_in，tim_tix）所使用的采样时钟（t_{DTS}）之间的分频比

00：$t_{DTS} = t_{tim_ker_ck}$

01：$t_{DTS} = 2 \times t_{tim_ker_ck}$

10：$t_{DTS} = 4 \times t_{tim_ker_ck}$

11：保留，不要设置成此值

Bit 7 **ARPE**：自动重载预装载使能（Auto-reload preload enable）

0：TIMx_ARR 寄存器不进行缓冲；
1：TIMx_ARR 寄存器进行缓冲。

Bits 6:5 **CMS[1:0]**：中央对齐模式选择（Center-aligned mode selection）

00：边沿对齐模式。计数器根据方向位（DIR）递增计数或递减计数。
01：中心对齐模式 1。计数器交替进行递增计数和递减计数。仅当计数器递减计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CxS=00）的输出比较中断标志才置 1。
10：中心对齐模式 2。计数器交替进行递增计数和递减计数。仅当计数器递增计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CxS=00）的输出比较中断标志才置 1。
11：中心对齐模式 3。计数器交替进行递增计数和递减计数。当计数器递增计数或递减计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CxS=00）的输出比较中断标志都会置 1。

注：只要计数器处于使能状态 (CEN=1)*，就不得从边沿对齐模式切换为中心对齐模式。*

Bit 4 **DIR**：方向（Direction）

0：计数器递增计数；
1：计数器递减计数。

注：当定时器配置为中心对齐模式或编码器模式时，该位为只读状态。

Bit 3 **OPM**：单脉冲模式（One pulse mode）

0：计数器在发生更新事件时不会停止计数；
1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）

Bit 2 **URS**：更新请求源（Update request source）

此位由软件置 1 和清零，用以选择 UEV 事件源。
0：使能时，所有以下事件都会生成更新中断。此类事件包括：
- 计数器上溢/下溢
- 将 UG 位置 1
- 通过从模式控制器生成的更新事件

1：使能时，只有计数器上溢/下溢会生成更新中断。

Bit 1 **UDIS**：更新禁止（Update disable）

此位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。
0：使能 UEV。更新（UEV）事件可通过以下事件之一生成：
- 计数器上出/下溢
- 将 UG 位置 1

– 通过从模式控制器生成的更新事件

然后更新影子寄存器的值。

1：禁止 UEV。不会生成更新事件，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。但如果将 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。

Bit 0 **CEN**：计数器使能（Counter enable）

0：禁止计数器；

1：使能计数器。

*注：只有事先通过软件将*CEN *位置*1*，才可以使用外部时钟、门控模式和编码器模式。而触发模式可通过硬件自动将*CEN *位置*1*。*

14.7.2 TIMx 控制寄存器 2（TIMx_CR2）（x=2/3）

偏移地址：0x04

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						MMS[3]	Reserved								
						rw									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TI1S	MMS[2:0]			Reserved			
								rw	rw	rw	rw				

Bits 31:26 保留，必须保持为复位值。

Bit 25 **MMS[3]**：主模式选择（Master mode selection）

Bits 24:8 保留，必须保持为复位值。

Bit 7 **TI1S**：tim_ti1 选择（tim_ti1 selection）

0：tim_ti1_in[15:0]引脚连接到 tim_ti1 输入

1：tim_ti1_in[15:0]，tim_ti2_in[15:0]和 tim_ti3_in[15:0]多路复用器输出被异或并连接至 tim_ti1 输入

Bits 25, 6:4 **MMS[3:0]：**主模式选择（Master mode selection）

这些位可选择主模式下将要发送到从定时器以实现同步的信息(tim_trgo)。这些位的组合如下：

0000：**复位** – TIMx_EGR 寄存器中的 UG 位用作触发输出（tim_trgo)。如果复位由触发输入生成（从模式控制器配置为复位模式），则 TRGO 上的信号相比实际复位会有延迟。

0001：**使能** – 计数器使能信号 CNT_EN 用作触发输出（tim_trgo)。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号可由 CEN 控制位产生。当配置为门控模式时，也可由触发输入产生。当计数器使能信号由触发输入控制时，tim_trgo 上会存在延迟，选择主/从模式时除外（请参见 TIMx_SMCR 寄存器中 MSM 位的说明)。

0010：**更新** – 选择更新事件作为触发输出（tim_trgo)。例如，主定时器可用作从定时器的预分频器。

0011：**比较脉冲** – 一旦发生输入捕获或比较匹配事件，当 CC1IF 被置 1 时（即使已为高电平），触发输出都会发送一个正脉冲。(tim_trgo)。

0100：**比较** – tim_oc1refc 信号用作触发输出（tim_trgo）

0101：**比较** – tim_oc2refc 信号用作触发输出（tim_trgo）
0110：**比较** – tim_oc3refc 信号用作触发输出（tim_trgo）
0111：**比较** – tim_oc4refc 信号用作触发输出（tim_trgo）
1000：**编码器时钟输出** – 编码器时钟信号用作出发输出（tim_trgo）。此编码对以下 SMS[3:0]值有效：0001，0010，0011，1010，1011，1100，1101，1110，1111。任何其他 SMS[3:0]编码都是不允许的，可能会导致意外行为。
其他编码保留
注：在从主定时器接收事件之前，必须启用从机定时器或 ADC *的时钟，并且在从主定时器接收触发器时不能随意更改。*

Bits 3:0 保留，必须保持复位值。

14.7.3 TIMx 从模式控制寄存器（TIMx_SMCR）（x=2/3）

偏移地址：0x08
复位值：0x0000 0000
访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						SMSPS	SMSPE	Reserved							SMS[3]
						rw	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	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:26 保留，必须保持复位值。

Bit 25 **SMSPS**：SMS 预加载源（SMS preload source）
此位选择触发 SMS[3:0]位域从预加载到活动的事件来源：
0：由定时器的更新事件触发
1：由索引事件触发

Bit 24 **SMSPE**：SMS 预加载使能（SMS preload enable）
此位选择是否使能 SMS[3:0]位域的预加载：
0：SMS[3:0]位域不进行预加载
1：使能 SMS[3:0]预加载

Bits 23:17 保留，必须保持复位值。

Bit 16 **SMS[3]**：从模式选择（Slave mode selection）

Bit 15 **ETP**：外部触发极性（External trigger polarity）
此位可选择将 tim_etr_in 还是 tim_etr_in 的反相用于触发操作
0：tim_etr_in 未反相，高电平或上升沿有效。
1：tim_etr_in 反相，低电平或下降沿有效。

Bit 14 **ECE**：外部时钟使能位（External clock enable）
此位可使能外部时钟模式 2。
0：禁止外部时钟模式 2
1：使能外部时钟模式 2。计数器时钟由 tim_etrf 信号的任意有效边沿提供。
注：将 ECE *位置*1 *与选择外部时钟模式*1 *并将* tim_trgi *连接到* tim_etrf *（*SMS=111 *且* TS=00111*）具有相同效果。*
*外部时钟模式*2 *可以和以下从模式同时使用：复位模式、门控模式和触发模式。不过此类情况下* tim_trgi *不得连接* tim_etrf *（*TS *位不得为*00111*）。*

*如果同时使能外部时钟模式1 和外部时钟模式2，则外部时钟输入为*tim_etrf。

Bits 13:12 **ETPS[1:0]**：外部触发预分频器（External trigger prescaler）

外部触发信号 tim_etrp 频率不得超过 TIMxCLK 频率的 1/4。可通过使能预分频器来降低 tim_etrp 频率。这种方法在 tim_etr_in 输入快速外部时钟时非常有用。

00：预分频器关闭

01：2 分频 tim_etrp 频率

10：4 分频 tim_etrp 频率

11：8 分频 tim_etrp 频率

Bits 11:8 **ETF[3:0]**：外部触发滤波器（External trigger filter）

此位域可定义 tim_etrp 信号的采样频率和适用于 tim_etrp 的数字滤波时间。数字滤波器由事件计数器组成，每 N 个事件才视为一个有效边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样

0001：$f_{SAMPLING}=f_{CK_INT}$，N=2。

0010：$f_{SAMPLING}=f_{CK_INT}$，N=4。

0011：$f_{SAMPLING}=f_{CK_INT}$，N=8。

0100：$f_{SAMPLING}=f_{DTS}/2$，N=6。

0101：$f_{SAMPLING}=f_{DTS}/2$，N=8。

0110：$f_{SAMPLING}=f_{DTS}/4$，N=6。

0111：$f_{SAMPLING}=f_{DTS}/4$，N=8。

1000：$f_{SAMPLING}=f_{DTS}/8$，N=6。

1001：$f_{SAMPLING}=f_{DTS}/8$，N=8。

1010：$f_{SAMPLING}=f_{DTS}/16$，N=5。

1011：$f_{SAMPLING}=f_{DTS}/16$，N=6。

1100：$f_{SAMPLING}=f_{DTS}/16$，N=8。

1101：$f_{SAMPLING}=f_{DTS}/32$，N=5。

1110：$f_{SAMPLING}=f_{DTS}/32$，N=6。

1111：$f_{SAMPLING}=f_{DTS}/32$，N=8。

Bit 7 **MSM：**主/从模式（Master/slave mode）

0：不执行任何操作；

1：当前定时器的触发输入事件(tim_trgi)的动作被推迟，以使当前定时器与其从定时器实现完美同步（通过 tim_trgo）。此设置适用于由单个外部事件对多个定时器进行同步的情况。

Bits 6:4 **TS[2:0]：**触发选择（Trigger selection）

此位域可选择将要用于同步计数器的触发输入。

00000：内部触发 0（tim_itr0）

00001：内部触发 1（tim_itr1）

00010：内部触发 2（tim_itr2）

00100：tim_ti1 边沿检测器（tim_ti1f_ed）

00101：滤波后的定时器输入 1（tim_ti1fp1）

00110：滤波后的定时器输入 1（tim_ti2fp2）

00111：外部触发输入（tim_etrf）

有关定时器 ITRx 含义的详细信息，请参见章节：TIM2/TIM3 引脚和内部信号

*注：这些位只能在未使用的情况下（例如，*SMS=000 *时）进行更改，以避免转换时出现错*

误的边沿检测。

Bit 3 **OCCS**：OCREF 清除源选择（OCREF clear selection）

此位用于选择 OCREF 清除源。

0：tim_ocref_clr_int 信号连接到 tim_ocref_clr 输入

1：tim_ocref_clr_int 信号连接到 tim_etrf

Bits 16, 2:0 **SMS[3:0]**：从模式选择（Slave mode selection）

当选择了外部信号，触发信号（tim_trgi）的有效边沿与选中的外部输入极性相关（见输入控制寄存器和控制寄存器的说明）

0000：**关闭从模式** – 如果 CEN=1，则预分频器直接由内部时钟驱动。

0001：**编码器模式 1** – 计数器根据 tim_ti1fp1 电平在 tim_ti2fp2 边沿递增/递减计数。

0010：**编码器模式 2** – 计数器根据 tim_ti2fp2 电平在 tim_ti1fp1 边沿递增/递减计数。

0011：**编码器模式 3** – 计数器在 tim_ti1fp1 和 tim_ti2fp2 的边沿计数，计数的方向取决于另外一个信号的电平。

0100：**复位模式** – 选中的触发输入（tim_trgi）的上升沿重新初始化计数器，并且产生一个更新寄存器的信号。

0101：**门控模式** – 当触发输入（tim_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。

0110：**触发模式** – 计数器在触发输入 tim_trgi 的上升沿启动（但不复位），只有计数器的启动是受控的。

0111：**外部时钟模式 1** – 选中的触发输入（tim_trgi）的上升沿驱动计数器。

1000：**组合重置+触发器模式** - 选定触发器输入（tim_trgi）的上升沿会重新初始化计数器，生成寄存器的更新并启动计数器。

1001：**组合门控+复位模式** - 当触发器输入（tim_trgi）为高电平时计数器时钟使能。一旦触发器变为低电平计数器将停止并重置。同时控制计数器的启动和停止。

1010：**编码器模式**：时钟加方向，x2 模式。

1011：**编码器模式**：时钟加方向，x1 模式，CC2P 设置 tim_ti2fp2 边沿敏感度。

1100：**编码器模式**：方向时钟，x2 模式。

1101：**编码器模式**：方向时钟，x1 模式，CC1P 和 CC2P 设置 tim_ti1fp1 和 tim_ti2fp2 边沿敏感度.

1110：**四象限编码器模式**：x1 模式，仅计算 tim_ti1fp1 边沿，CC1P 设置边沿敏感度。

1111：**四象限编码器模式**：x1 模式，仅计算 tim_ti2fp2 边沿，CC2P 设置边沿敏感度。

注：如果选择 tim_ti1f_ed *作为触发输入（*TS=00100*），则不得使用门控模式。实际上，*tim_ti1f_ed *为* TI1F *上的每个过渡输出* 1 *个脉冲，而门控模式检查触发信号的电平。*

注：接收 tim_trgo *或* tim_trgo2 *信号的从机外设（定时器、*ADC *等）的时钟必须在从主定时器接收事件之前使能，并且时钟频率（预分频器）在从主定时器接收触发器时不能随意更改。*

14.7.4 TIMx 中断使能寄存器（TIMx_DIER）（x=2/3）

偏移地址：0x0C

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								TERRIE	IERRIE	DIRIE	IDXIE	Reserved			
								rw	rw	rw	rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TIE	Res	CC4IE	CC3IE	CC2IE	CC1IE	UIE	
								rw		rw	rw	rw	rw	rw	

Bits 31:24 保留，必须保持为复位值。

Bit 23 **TERRIE**：转换错误中断使能（Transition error interrupt enable）

0：禁止转换错误中断；

1：使能转换错误中断。

Bit 22 **IERRIE**：索引错误中断使能（Index error interrupt enable）

0：禁止索引错误中断；

1：使能索引错误中断。

Bit 21 **DIRIE**：方向改变中断使能（Direction change interrupt enable）

0：禁止方向改变中断；

1：使能方向改变中断。

Bit 20 **IDXIE**：索引中断使能（Index interrupt enable）

0：禁止索引中断；

1：使能索引中断。

Bits 19:7 保留，必须保持为复位值。

Bit 6 **TIF**：触发信号（TGRI）中断使能（Trigger interrupt enable）

0：禁止触发信号（TGRI）中断；

1：使能触发信号（TGRI）中断。

Bit 5 保留，必须保持为复位值。

Bit 4 **CC4IE**：捕获/比较 4 中断使能（Capture/Compare 4 interrupt enable）

0：禁止 CC4 中断；

1：使能 CC4 中断。

Bit 3 **CC3IE**：捕获/比较 3 中断使能（Capture/Compare 3 interrupt enable）

0：禁止 CC3 中断；

1：使能 CC3 中断。

Bit 2 **CC2IE**：捕获/比较 3 中断使能（Capture/Compare 2 interrupt enable）

0：禁止 CC2 中断；

1：使能 CC2 中断。

Bit 1 **CC1IE**：捕获/比较 3 中断使能（Capture/Compare 1 interrupt enable）

0：禁止 CC1 中断；

1：使能 CC1 中断。

Bit 0 **UIE**：更新中断使能（Update interrupt enable）

0：禁止更新中断；

1：使能更新中断。

14.7.5 TIMx 状态寄存器（TIMx_SR）（x=2/3）

偏移地址：0x10

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								TERRF	IERRF	DIRF	IDXF	Reserved			
								rc w0	rc w0	rc w0	rc w0				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CC4OF	CC3OF	CC2OF	CC1OF	Reserved		TIF	Res	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			rc w0	rc w0	rc w0	rc w0			rc w0		rc w0	rc w0	rc w0	rc w0	rc w0

Bits 31:24 保留，必须保持为复位值。

Bit 23 **TERRF**：转换错误中断标志（Transition error interrupt flag）

当在编码器模式下检测到转换错误时，硬件会设置此标志当软件将其写入“0”时，该标志将被清除

0：未检测到编码器转换错误

1：已检测到编码器转换错误

Bit 22 **IERRF**：索引错误中断标志（Index error interrupt flag）

当检测到索引错误时，硬件会设置此标志当软件将其写入“0”时，该标志将被清除

0：未检测到索引错误

1：已检测到索引错误

Bit 21 **DIRF**：方向改变中断标志（Direction change interrupt flag）

当编码器模式中的方向发生变化时（TIMx_CR 中的 DIR 位值正在改变），硬件会设置此标志当软件将其写入“0”时，该标志将被清除

0：方向未改变

1：方向已改变

Bit 20 **IDXF**：索引中断标志（Index interrupt flag）

当检测到索引事件时，硬件会设置此标志当软件将其写入“0”时，该标志将被清除

0：未发生索引事件

1：已发生索引事件

Bits 19:13 保留，必须保持为复位值。

Bit 12 **CC4OF**：捕获/比较 4 重复捕获标志

参考 CC1OF 的描述

Bit 11 **CC3OF**：捕获/比较 3 重复捕获标志

参考 CC1OF 的描述

Bit 10 **CC2OF**：捕获/比较 2 重复捕获标志

参考 CC1OF 的描述

Bit 9 **CC1OF**：捕获/比较 1 重复捕获标志

仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将该位清零。

0：未检测到重复捕获

1：TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。

Bits 8:7 保留，必须保持为复位值。

Bit 6 TIF：触发中断标志（Trigger interrupt flag）

在除门控模式以外的所有模式下，当使能从模式控制器后在 tim_trgi 输入上检测到有效边沿时，该标志将由硬件置 1。选择门控模式时，该标志将在计数器启动或停止时置 1。但需要通过软件清零。

0：未发生触发事件。

1：触发中断挂起。

Bit 5 保留，必须保持为复位值。

Bit 4 **CC4IF**：捕获/比较 4 中断标志（Capture/Compare 4 interrupt flag）

参考 CC1IF 描述。

Bit 3 **CC3IF**：捕获/比较 3 中断标志（Capture/Compare 3 interrupt flag）

参考 CC1IF 描述。

Bit 2 **CC2IF**：捕获/比较 2 中断标志（Capture/Compare 2 interrupt flag）

参考 CC1IF 描述。

Bit 1 **CC1IF**：捕获/比较 1 中断标志（Capture/Compare 1 interrupt flag）

此标志由硬件设置，它可以通过软件（输入捕获或输出比较模式）或读取 TIMx_CCR1 寄存器（仅输入捕获模式）来清除。

0：无比较匹配/无输入捕获发生

1：发生比较匹配或输入捕获

如果通道 CC1 配置为输出：当计数器 TIMx_CNT 的内容与 TIMx_CCR1 寄存器内容匹配时，会设置此标志。当 TIMx_CCR1 的值大于 TIMx_ARR 的值时，CC1IF 位将在计数器发生上溢（递增计数模式和增减计数模式下）或下溢（递减计数模式下）时变为高电平。在中心对齐模式下有 3 种可能的标志设置选项，请参考 TIMx_CR1 寄存器中的 CMS 位以获取完整描述。

如果通道 CC1 配置为输入：当在 TIMx_CCR1 寄存器中捕获到计数器值时（在 TIMx_CCER 中定义的 CC1P 和 CC1NP 位设置所确定的 IC1 上检测到边沿时），会设置此位。

Bit 0 **UIF**：更新中断标志（Update interrupt flag）

该位在发生更新事件时通过硬件置 1。但需要通过软件清零。

0：未发生更新。

1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：

- TIMx_CR1 寄存器中的 UDIS=0，并且重复计数器值上溢或下溢时（重复计数器= 0 时更新）。
- TIMx_CR1 寄存器中的 URS=0 且 UDIS=0，并且由软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。
- TIMx_CR1 寄存器中的 URS=0 且 UDIS=0，并且 CNT 由触发事件重新初始化时（参考同步控制寄存器描述）。

14.7.6 TIMx 事件产生寄存器（TIMx_EGR）（x=2/3）

偏移地址：0x14

复位：0x0000 0000

访问：无等待，支持字，半字和字节访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									TG	Res	CC4G	CC3G	CC2G	CC1G	UG
									w		w	w	w	w	w

Bits 15:7 保留，必须保持为复位值。

Bit 6 **TG**：生成触发信号（Trigger generation）

此位由软件置 1 以生成事件，并由硬件自动清零。

0：不执行任何操作；

1：TIMx_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断事件。

Bit 5 保留，必须保持为复位值。

Bit 4 **CC4G**：捕获/比较 4 生成（Capture/Compare 4 generation）

参考 CC1G 描述。

Bit 3 **CC3G**：捕获/比较 3 生成（Capture/Compare 3 generation）

参考 CC1G 描述。

Bit 2 **CC2G**：捕获/比较 2 生成（Capture/Compare 2 generation）

参考 CC1G 描述。

Bit 1 **CC1G**：捕获/比较 1 生成（Capture/Compare 1 generation）

此位由软件置 1 以生成事件，并由硬件自动清零。

0：不执行任何操作；

1：通道 1 上生成捕获/比较事件：

若通道 CC1 配置为输出：

使能时，CC1IF 标志置 1 并发送相应的中断。

如果通道 CC1 配置为输入：

TIMx_CCR1 寄存器中将捕获到计数器当前值。使能时，CC1IF 标志置 1 并发送相应的中断请求。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。

Bit 0 **UG**：更新生成（Update generation）

该位可通过软件置 1，并由硬件自动清零。

0：不执行任何操作；

1：重新初始化计数器并生成一个寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。如果选择中心对齐模式或 DIR=0（递增计数），计数器将清零；如果 DIR=1（递减计数），计数器将使用自动重载值（TIMx_ARR）。

14.7.7 TIMx 捕获/比较模式寄存器 1 [复用]（TIMx_CCMR1）（x=2/3）

偏移地址：0x18

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

相同的寄存器可以用于输入捕获模式（本节）或输出比较模式（下节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其它位在输入模式和输出模式下的功能均不同。可以独立组合这两种模式（例如，通道 1 用作输入捕获模式而通道 2 用作输出比较模式）

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC2F[3:0]				IC2PSC[1:0]		CC2S[1:0]		IC1F[3:0]				IC1PSC[1:0]		CC1S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

输入捕获模式

Bits 31:16 保留，必须保持为复位值。

Bits 15:12 **IC2F[3:0]：**输入捕获 2 滤波器（Input capture 2 filter）

Bits 11:10 **IC2PSC[1:0]：**输入/捕获 2 预分频器（input capture 2 prescaler）

Bits 9:8 **CC2S[1:0]：**捕获/比较 2 选择（Capture/compare 2 selection）

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC2 通道被配置为输出；

01：CC2 通道被配置为输入，tim_ic2 映射在 tim_ti2 上；

10：CC2 通道被配置为输入，tim_ic2 映射在 tim_ti1 上；

11：CC2 通道被配置为输入，tim_ic2 映射在 tim_trc 上。此模此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注：仅当通道关闭时（TIMx_CCER 中的 CC2E = 0），才可向 CC2S 位写入数据。

Bits 7:4 **IC1F[3:0]：**输入捕获 1 滤波器（Input capture 1 filter）

此位域可定义 tim_ti1 输入的采样频率和适用于 tim_ti1 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个事件才视为一个有效边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样

0001：$f_{SAMPLING}=f_{tim_ker_ck}$，N=2。

0010：$f_{SAMPLING}=f_{tim_ker_ck}$，N=4。

0011：$f_{SAMPLING}=f_{tim_ker_ck}$，N=8。

0100：$f_{SAMPLING}=f_{DTS}/2$，N=6。

0101：$f_{SAMPLING}=f_{DTS}/2$，N=8。

0110：$f_{SAMPLING}=f_{DTS}/4$，N=6。

0111：$f_{SAMPLING}=f_{DTS}/4$，N=8。

1000：$f_{SAMPLING}=f_{DTS}/8$，N=6。

1001：$f_{SAMPLING}=f_{DTS}/8$，N=8。

1010：$f_{SAMPLING}=f_{DTS}/16$，N=5。

1011：$f_{SAMPLING}=f_{DTS}/16$，N=6。

1100：$f_{SAMPLING}=f_{DTS}/16$，N=8。

1101：$f_{SAMPLING}=f_{DTS}/32$，N=5。

1110：$f_{SAMPLING}=f_{DTS}/32$，N=6。

1111：$f_{SAMPLING}=f_{DTS}/32$，N=8。

Bits 3:2 **IC1PSC[1:0]**：输入/捕获 1 预分频器（Input capture 1 prescaler）

此位域定义 CC1 输入（tim_ic1）的预分频比。

只要 CC1E=0（TIMx_CCER 寄存器），预分频器便立即复位。

00：无预分频器，捕获输入上每检测到一个边沿便执行捕获

01：每发生 2 个事件便执行一次捕获；

10：每发生 4 个事件便执行一次捕获；

11：每发生 8 个事件便执行一次捕获。

Bits 1:0 **CC1S[1:0]**：捕获/比较 1 选择（Capture/Compare 1 selection）

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道被配置为输出；

01：CC1 通道被配置为输入，tim_ic1 映射在 tim_ti1 上；

10：CC1 通道被配置为输入，tim_ic1 映射在 tim_ti2 上；

11：CC1 通道被配置为输入，tim_ic1 映射在 tim_trc 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注：仅当通道关闭时（TIMx_CCER 中的 CC1E = 0），才可向 CC1S 位写入数据。

14.7.8 TIMx 捕获/比较模式寄存器 1 [复用]（TIMx_CCMR1）（x=2/3）

偏移地址：0x18

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

相同的寄存器可以用于输出比较模式（本节）或输入捕获模式（上节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其它位在输入模式和输出模式下的功能均不同。可以独立组合这两种模式（例如，通道 1 用作输入捕获模式而通道 2 用作输出比较模式）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							OC2M[3]	Reserved							OC1M[3]
							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	rw	rw	rw	rw	rw	rw

输出比较模式

Bits 31:25 保留，必须保持为复位值。

Bit 24 **OC2M[3]**：输出比较 2 模式（Output Compare 2 mode）

Bits 23:17 保留，必须保持为复位值。

Bit 16 **OC1M[3]**：输出比较 1 模式（Output compare 1 mode）

Bit 15 **OC2CE**：输出比较 2 清零使能（Output Compare 2 clear enable）

Bits 24, 14:12 **OC2M[3:0]**：输出比较 2 模式（Output Compare 2 mode）

参考 OC1M 描述

Bit 11 **OC2PE**：输出比较 2 预装载使能（Output Compare 2 preload enable）

Bit 10 **OC2FE**：输出比较 2 快速使能（Output Compare 2 fast enable）

Bits 9:8 **CC2S[1:0]**：捕获/比较 2 选择（Capture/Compare 2 selection）

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC2 通道配置为输出

01：CC2 通道配置为输入，tim_ic2 映射到 tim_ti2 上

10：CC2 通道配置为输入，tim_ic2 映射到 tim_ti1 上

11：CC2 通道配置为输入，tim_ic2 映射到 tim_trc 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（TIMx_CCER 中的 CC2E = 0），才可向 CC2S 位写入数据

Bit 7 **OC1CE：**输出比较 1 清零使能（Output Compare 1 clear enable）

0：tim_oc1ref 不受 tim_ocref_clr_int 信号影响；

1：tim_oc1ref 在 tim_ocref_clr_int 信号中被检测到高电平，OC1Ref 立即清零。(tim_ocref_clr 输入或 tim_etrf 输入)

Bits 16, 6:4 **OC1M[3:0]：**输出比较 1 模式（Output Compare 1 mode）

这些位定义提供 tim_oc1 和 tim_oc1n 的输出参考信号 tim_oc1ref 的行为。tim_oc1ref 为高电平有效，而 tim_oc1 和 tim_oc1n 的有效电平则取决于 CC1P 位和 CC1NP 位。

0000：**冻结** - 输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。（该模式用于生成时基）。

0001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1（TIMx_CCR1）匹配时，tim_oc1ref 信号强制变为高电平。

0010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1（TIMx_CCR1）匹配时，tim_oc1ref 信号强制变为低电平。

0011：**翻转** - TIMx_CNT=TIMx_CCR1 时，tim_oc1ref 发生翻转。

0100：**强制变为无效电平** – tim_oc1ref 强制变为低电平。

0101：**强制变为有效电平** – tim_oc1ref 强制变为高电平。

0110：**PWM 模式 1**-在递增计数模式下，只要 TIMx_CNT<TIMx_CCR1，通道 1 便为有效状态，否则为无效状态。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，通道 1 便为无效状态（tim_oc1ref=0），否则为有效状态（tim_oc1ref=1）。

0111：**PWM 模式 2** – 在递增计数模式下，只要 TIMx_CNT<TIMx_CCR1，通道 1 便为无效状态，否则为有效状态。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，通道 1 便为有效状态，否则为无效状态。

1000：**可重触发 OPM 模式 1** - 在向上计数模式下，通道在检测到触发事件（tim_trgi 信号）之前一直处于活动状态。然后，会像 PWM 模式 1 那样执行比较，并在下次更新时使通道再次处于活动状态。在向下计数模式下，通道在检测到触发事件（tim_trgi 信号）之前一直处于非活动状态。然后，会像 PWM 模式 1 那样执行比较，并在下次更新时使通道再次处于非活动状态。

1001：**可重触发 OPM 模式 2** - 在向上计数模式下，通道在检测到触发事件（tim_trgi 信号）之前一直处于非活动状态。然后，会像 PWM 模式 2 那样执行比较，并在下次更新时使通道再次处于非活动状态。在向下计数模式下，通道在检测到触发事件（tim_trgi 信号）之前一直处于活动状态。然后，会像 PWM 模式 1 那样执行比较，并在下次更新时使通道再次处于活动状态。

1010：保留。

1011：保留。

1100：**组合 PWM 模式 1** - tim_oc1ref 的行为与 PWM 模式 1 相同。tim_oc1refc 是 tim_oc1ref 和 tim_oc2ref 之间的逻辑 OR。

1101：**组合 PWM 模式 2** - tim_oc1ref 的行为与 PWM 模式 2 相同。tim_oc1refc 是

tim_oc1ref 和 tim_oc2ref 之间的逻辑 AND。

1110：**非对称 PWM 模式 1** - tim_oc1ref 的行为与 PWM 模式 1 相同。当计数器向上计数时，tim_oc1refc 输出 tim_oc1ref；当计数器向下计数时，tim_oc1refc 输出 tim_oc2ref。

1111：**非对称 PWM 模式 2** - tim_oc1ref 的行为与 PWM 模式 2 相同。当计数器向上计数时，tim_oc1refc 输出 tim_oc1ref；当计数器向下计数时，tim_oc1refc 输出 tim_oc2ref。

注：在 PWM 模式下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时，tim_ocref_clr 电平才会发生更改。

Bit 3 **OC1PE：**输出比较 1 预装载使能（Output Compare 1 preload enable）

0：禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立即使用新值。

1：使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到活动寄存器中。

Bit 2 **OC1FE：**输出比较 1 快速使能（Output Compare 1 fast enable）

此位用于加快触发输入事件对 CC 输出的影响。

0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的最短延迟时间为 5 个时钟周期。

1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OCFE 才会起作用。

Bits 1:0 **CC1S[1:0]：**捕获/比较 1 选择（Capture/Compare 1 selection）

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出

01：CC1 通道配置为输入，tim_ic1 映射到 tim_ti1 上

10：CC1 通道配置为输入，tim_ic1 映射到 tim_ti2 上

11：CC1 通道配置为输入，tim_ic1 映射到 tim_trc 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注：仅当通道关闭时（TIMx_CCER 中的 CC1E = 0），才可向 CC1S 位写入数据

14.7.9 TIMx 捕获/比较模式寄存器 2 [复用]（TIMx_CCMR2）（x=2/3）

偏移地址：0x1C

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

相同的寄存器可以用于输入捕获模式（本节）或输出比较模式（下节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其它位在输入模式和输出模式下的功能均不同。可以独立组合这两种模式（例如，通道 1 用作输入捕获模式而通道 2 用作输出比较模式）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC4F[3:0]				IC4PSC[1:0]		CC4S[1:0]		IC3F[3:0]				IC3PSC[1:0]		CC3S[1:0]	

输入捕获模式

Bits 31:16 保留，必须保持为复位值。

Bits 15:12 **IC4F[3:0]：**输入捕获 4 滤波器（Input capture 4 filter）

Bits 11:10 **IC4PSC[1:0]：**输入/捕获 4 预分频器（input capture 4 prescaler）

Bits 9:8 **CC4S[1:0]：**捕获/比较 4 选择 （Capture/compare 4 selection）

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC4 通道被配置为输出；

01：CC4 通道被配置为输入，tim_ic4 映射在 tim_ti4 上；

10：CC4 通道被配置为输入，tim_ic4 映射在 tim_ti3 上；

11：CC4 通道被配置为输入，tim_ic4 映射在 tim_trc 上。此模此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注：仅当通道关闭时（TIMx_CCER 中的 CC4E = 0），才可向CC4S 位写入数据。

Bits 7:4 **IC3F[3:0]：**输入捕获 3 滤波器（Input capture 3 filter）

Bits 3:2 **IC3PSC[1:0]：**输入/捕获 1 预分频器（Input capture 3 prescaler）

Bits 1:0 **CC3S[1:0]：**捕获/比较 3 选择（Capture/Compare 3 selection）

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC3 通道被配置为输出；

01：CC3 通道被配置为输入，tim_ic3 映射在 tim_ti3 上；

10：CC3 通道被配置为输入，tim_ic3 映射在 tim_ti4 上；

11：CC3 通道被配置为输入，tim_ic3 映射在 tim_trc 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注：仅当通道关闭时（TIMx_CCER 中的 CC3E = 0），才可向CC3S 位写入数据。

14.7.10 TIMx 捕获/比较模式寄存器 2 [复用]（TIMx_CCMR2）（x=2/3）

偏移地址：0x1C

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

相同的寄存器可以用于输出比较模式（本节）或输入捕获模式（上节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其它位在输入模式和输出模式下的功能均不同。可以独立组合这两种模式（例如，通道 1 用作输入捕获模式而通道 2 用作输出比较模式）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							OC4M[3]	Reserved							OC3M[3]
							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	rw	rw	rw	rw	rw	rw

输出比较模式

Bits 31:25 保留，必须保持为复位值。

Bits 24 **OC4M[3]：**输出比较 4 模式（Output Compare 4 mode）

Bits 23:17 保留，必须保持为复位值。

Bit 16 **OC3M[3]：**输出比较 3 模式（Output compare 3 mode）

Bit 15 **OC4CE：**输出比较 4 清零使能（Output Compare 4 clear enable）

Bits 24, 14:12 **OC4M[2:0]**：输出比较 4 模式（Output Compare 4 mode）

参考 OC1M 描述（TIMx_CCMR1 寄存器）

Bit 11 **OC4PE**：输出比较 4 预装载使能（Output Compare 4 preload enable）

Bit 10 **OC4FE**：输出比较 4 快速使能（Output Compare 4 fast enable）

Bits 9:8 **CC4S[1:0]**：捕获/比较 4 选择（Capture/Compare 4 selection）

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC4 通道配置为输出

01：CC4 通道配置为输入，tim_ic4 映射到 tim_ti4 上

10：CC4 通道配置为输入，tim_ic4 映射到 tim_ti3 上

11：CC4 通道配置为输入，tim_ic4 映射到 tim_trc 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注：仅当通道关闭时（TIMx_CCER 中的 CC4E = 0），才可向 CC4S 位写入数据

Bit 7 **OC3CE**：输出比较 3 清零使能（Output Compare 3 clear enable）

Bits 16, 6:4 **OC3M[3:0]**：输出比较 3 模式（Output Compare 1 mode）

参考 OC1M 描述（TIMx_CCMR1 寄存器）

Bit 3 **OC3PE**：输出比较 3 预装载使能（Output Compare 3 preload enable）

Bit 2 **OC3FE**：输出比较 3 快速使能（Output Compare 3 fast enable）

Bits 1:0 **CC3S[1:0]**：捕获/比较 3 选择（Capture/Compare 3 selection）

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC3 通道配置为输出

01：CC3 通道配置为输入，tim_ic3 映射到 tim_ti3 上

10：CC3 通道配置为输入，tim_ic3 映射到 tim_ti4 上

11：CC3 通道配置为输入，tim_ic3 映射到 tim_trc 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注：仅当通道关闭时（TIMx_CCER 中的 CC3E = 0），才可向 CC3S 位写入数据

14.7.11 TIMx 捕获/比较使能寄存器（TIMx_CCER）（x=2/3）

偏移地址：0x20

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

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

Bit 15 **CC4NP**：输入/捕获 4 输出极性（Capture/Compare 4 output polarity）

参考 CC1NP 的描述。

Bit 14 保留，必须保持为复位值。

Bit 13 **CC4P**：输入/捕获 4 输出极性（Capture/Compare 4 output polarity）

参考 CC1P 的描述。

Bit 12 **CC4E**：输入/捕获 4 输出使能（Capture/Compare 4 output enable）

参考 CC1E 的描述。

Bit 11 **CC3NP**：输入/捕获 3 输出极性（Capture/Compare 3 output polarity）

参考 CC1NP 的描述。

Bit 10 保留，必须保持为复位值。

Bit 9 **CC3P**：输入/捕获 3 输出极性（Capture/Compare 3 output polarity）

参考 CC1P 的描述。

Bit 8 **CC3E**：输入/捕获 3 输出使能（Capture/Compare 3 output enable）

参考 CC1E 的描述。

Bit 7 **CC2NP**：输入/捕获 2 输出极性（Capture/Compare 2 output polarity）

参考 CC1NP 的描述。

Bit 6 保留，必须保持为复位值。

Bit 5 **CC2P**：输入/捕获 2 输出极性（Capture/Compare 2 output polarity）

参考 CC1P 的描述。

Bit 4 **CC2E**：输入/捕获 2 输出使能（Capture/Compare 2 output enable）

参考 CC1E 的描述。

Bit 3 **CC1NP**：输入/捕获 1 输出极性（Capture/Compare 1 output polarity）

CC1 通道配置为输出：

在这种情况下，CC1NP 必须保持清零。

CC1 通道配置为输入：

此位与 CC1P 配合使用，用以定义 tim_ti1fp1/ tim_ti2fp1 的极性。请参见 CC1P 说明。

Bit 2 保留，必须保持为复位值。

Bit 1 **CC1P**：输入/捕获 1 输出极性（Capture/Compare 1 output polarity）

0：OC1 高电平有效（输出模式）/边沿灵敏度选择（输入模式，参见以下）

1：OC1 低电平有效（输出模式）/边沿灵敏度选择（输入模式，参见以下）

CC1 通道配置为输入，CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的极性。

CC1NP=0，CC1P=0：非反相/上升沿触发。电路对 TIxFP1 上升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式或编码器模式下执行触发操作）。

CC1NP=0，CC1P=1：反相/下降沿触发。电路对 TIxFP1 下降沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 反相（在门控模式或编码器模式下执行触发操作）。

CC1NP=1，CC1P=1：非反相/上升沿和下降沿均触发。电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。编码器模式下不得使用此配置。

CC1NP=1，CC1P=0：此配置是保留的，不应该使用此配置。

Bit 0 **CC1E**：输入/捕获 1 输出使能（Capture/Compare 1 output enable）

0：禁止捕获模式/OC1 未激活

1：使能捕获模式/ OC1 信号输出到对应的输出引脚

表 14.17 标准 tim_ocx 通道的输出控制位

CCxE 位	tim_ocx 输出状态
0	禁止输出（不由定时器驱动：Hi-Z）
1	输出使能（tim_ocx= tim_ocxref + 极性）

*注：当*CCxE=0 *时，连接到*tim_ocx *通道的外部*IO *引脚状态仅取决于*GPIO *寄存器*

14.7.12 TIM3 计数器（TIM3_CNT）

偏移地址：0x24

复位值：0x0000

访问：无等待，支持字，半字和字节访问

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

Bit 31 **UIFCPY：**值取决于 TIM3_CR1 的 UIFREMAP

如果 UIFREMAP = 0

保留

如果 UIFREMAP = 1

UIFCPY：复制 UIF

该位是 TIM3_SR 寄存器中 UIF 位的一个只读副本。

Bits 30:16 保留，必须保持为复位值。

Bits 15:0 **CNT[15:0]：**计数器的值（Counter value）

在非抖动模式下（DITHEN = 0）

该寄存器（CNT[15:0]）保存的是计数器的值。

在抖动模式下（DITHEN = 1）

该寄存器（CNT[15:0]）只保存计数器的不抖动部分。抖动部分不可用。

14.7.13 TIM2 计数器（TIM2_CNT）

偏移地址：0x24

复位值：0x0000

访问：无等待，支持字，半字和字节访问

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

Bit 31 **UIFCPY：**值取决于 TIM2_CR1 的 UIFREMAP

如果 UIFREMAP = 0

保留

如果 UIFREMAP = 1

UIFCPY：复制 UIF

该位是 TIM2_SR 寄存器中 UIF 位的一个只读副本。

Bits 30:28 保留，必须保持为复位值。

Bits 27:0 **CNT[27:0]：**计数器值的最小有效部分（Least significant part of counter value）

在非抖动模式下（DITHEN = 0）

该寄存器（CNT[27:0]）保存的是计数器的值。

在抖动模式下（DITHEN = 1）

该寄存器（CNT[27:0]）只保存计数器的不抖动部分。抖动部分不可用。

14.7.14 TIMx 预分频器（TIMx_PSC）（x=2/3）

偏移地址：0x28

复位值：0x0000

访问：无等待，支持字，半字和字节访问

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

Bits 15:0 **PSC[15:0]：**预分频器的值（Prescaler value）

计数器时钟频率（$f_{tim_cnt_ck}$）等于 $f_{tim_psc_ck}$ / (PSC[15:0] + 1)。

PSC 包含每次发生更新事件（包括计数器通过 TIMx_EGR 寄存器中的 UG 位清零时，或在配置为“复位模式”时通过触发控制器清零时）时要装载到活动预分频器寄存器的值。

14.7.15 TIM3 自动重装载寄存器（TIM3_ARR）

偏移地址：0x2C

复位：0x000F FFFF

访问：无等待，支持字，半字和字节访问

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

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

Bits 31:20 保留，必须保持为复位值。

Bits 19:0 **ARR[19:0]：**自动重装载的值（Auto-reload value）

ARR 为要装载到实际自动重载寄存器的值。

当自动重载值为空时，计数器不工作。

在非抖动模式下（DITHEN = 0）

该寄存器存储的是 ARR[15:0]中的自动重新加载的值

在抖动模式下（DITHEN = 1）

该寄存器存储的是 ARR[19:4]中的整数部分，而 ARR[3:0]位域存储的是抖动的部分

14.7.16 TIM2 自动重装载寄存器（TIM2_ARR）

偏移地址：0x2C

复位：0xFFFF FFFF

访问：无等待，支持字，半字和字节访问

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

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

Bits 31:0 **ARR[31:0]：**自动重装载的值（Auto-reload value）

ARR 为要装载到实际自动重载寄存器的值。
当自动重载值为空时，计数器不工作。
在非抖动模式下（DITHEN = 0）
该寄存器存储的是 ARR[27:0]中的自动重新加载的值
在抖动模式下（DITHEN = 1）
该寄存器存储的是 ARR[31:4]中的整数部分，而 ARR[3:0]位域存储的是抖动的部分

14.7.17 TIM3 捕获/比较寄存器 1（TIM3_CCR1）

偏移地址：0x34
复位值：0x0000
访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												CCR1 [19:16]			
												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:20 保留，必须保持为复位值。

Bits 19:0 **CCR1[19:0]：**捕获/比较 1 值（Capture/Compare 1 value）

若 CC1 通道配置为输出：

CCR1 为要装载到实际捕获/比较 1 寄存器的值（预装载值）。
如果没有通过 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 1 寄存器中。实际捕获/比较寄存器中包含要与计数器 TIM3_CNT 进行比较并在 tim_oc1 输出上发出信号的值。
在非抖动模式下（DITHEN = 0）
该寄存器存储的是 CCR1[15:0]中的比较值。CCR1[19:16]位会被重置。
在抖动模式下（DITHEN = 1）
该寄存器存储的是 CCR1[19:4]中的整数部分。CCR1[3:0]位域存储的是抖动的部分。

若 CC1 通道配置为输入：

CCR1 为上一个输入捕获 1 事件（tim_ic1）发生时的计数器值。TIM3_CCR1 是只读的并且不可编程。
在非抖动模式下（DITHEN = 0）
CCR1[15:0]位存储了捕获值。CCR1[19:16]位会被重置。
在抖动模式下（DITHEN = 1）
该寄存器存储捕获在 CCR1[19:0]中。CCR1[3:0]位会被重置。

14.7.18 TIM2 捕获/比较寄存器 1（TIM2_CCR1）

偏移地址：0x34

复位值：0x0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR1 [31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:0 **CCR1[31:0]：**捕获/比较 1 值（Capture/Compare 1 value）

若 CC1 通道配置为输出：

CCR1 为要装载到实际捕获/比较 1 寄存器的值（预装载值）。

如果没有通过 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 1 寄存器中。实际捕获/比较寄存器中包含要与计数器 TIM2_CNT 进行比较并在 tim_oc1 输出上发出信号的值。

在非抖动模式下（DITHEN = 0）

该寄存器存储的是 CCR1[27:0]中的比较值。CCR1[31:28]位会被重置。

在抖动模式下（DITHEN = 1）

该寄存器存储的是 CCR1[31:4]中的整数部分。CCR1[3:0]位域存储的是抖动的部分。

若 CC1 通道配置为输入：

CCR1 为上一个输入捕获 1 事件（tim_ic1）发生时的计数器值。TIM2_CCR1 是只读的并且不可编程。

在非抖动模式下（DITHEN = 0）

CCR1[27:0]位存储了捕获值。CCR1[31:28]位会被重置。

在抖动模式下（DITHEN = 1）

该寄存器存储捕获在 CCR1[31:0]中。CCR1[3:0]位会被重置。

14.7.19 TIM3 捕获/比较寄存器 2（TIM3_CCR2）

偏移地址：0x38

复位值：0x0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												CCR2 [19:16]			
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:20 保留，必须保持为复位值。

Bits 19:0 **CCR2[19:0]：**捕获/比较 2 值（Capture/Compare 2 value）

若 CC2 通道配置为输出：

CCR2 为要装载到实际捕获/比较 2 寄存器的值（预装载值）。

如果没有通过 TIMx_CCMR1 寄存器中的 OC2PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获

/比较 2 寄存器中。实际捕获/比较寄存器中包含要与计数器 TIM3_CNT 进行比较并在 tim_oc2 输出上发出信号的值。

在非抖动模式下（DITHEN = 0）

该寄存器存储的是 CCR2[15:0]中的比较值。CCR2[19:16]位会被重置。

在抖动模式下（DITHEN = 1）

该寄存器存储的是 CCR2[19:4]中的整数部分。CCR2[3:0]位域存储的是抖动的部分。

若 CC2 通道配置为输入：

CCR2 为上一个输入捕获 2 事件（tim_ic2）发生时的计数器值。TIM3_CCR2 是只读的并且不可编程。

在非抖动模式下（DITHEN = 0）

CCR2[15:0]位存储了捕获值。CCR2[19:16]位会被重置。

在抖动模式下（DITHEN = 1）

该寄存器存储捕获在 CCR2[19:0]中。CCR2[3:0]位会被重置。

14.7.20 TIM2 捕获/比较寄存器 2（TIM2_CCR2）

偏移地址：0x38

复位值：0x0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR2[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:0 **CCR2[31:0]：**捕获/比较 2 值（Capture/Compare 2 value）

若 CC2 通道配置为输出：

CCR2 为要装载到实际捕获/比较 2 寄存器的值（预装载值）。

如果没有通过 TIMx_CCMR1 寄存器中的 OC2PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 2 寄存器中。实际捕获/比较寄存器中包含要与计数器 TIM2_CNT 进行比较并在 tim_oc1 输出上发出信号的值。

在非抖动模式下（DITHEN = 0）

该寄存器存储的是 CCR2[27:0]中的比较值。CCR2[31:28]位会被重置。

在抖动模式下（DITHEN = 1）

该寄存器存储的是 CCR2[31:4]中的整数部分。CCR2[3:0]位域存储的是抖动的部分。

若 CC2 通道配置为输入：

CCR2 为上一个输入捕获 1 事件（tim_ic1）发生时的计数器值。TIM2_CCR2 是只读的并且不可编程。

在非抖动模式下（DITHEN = 0）

CCR2[27:0]位存储了捕获值。CCR2[31:28]位会被重置。

在抖动模式下（DITHEN = 1）

该寄存器存储捕获在 CCR2[31:0]中。CCR2[3:0]位会被重置。

14.7.21 TIM3 捕获/比较寄存器 3（TIM3_CCR3）

偏移地址：0x3C

复位值：0x0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												CCR3 [19:16]			
												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:20 保留，必须保持为复位值。

Bits 19:0 **CCR3[19:0]：**捕获/比较 3 值（Capture/Compare 3 value）

若 CC3 通道配置为输出：

CCR3 为要装载到实际捕获/比较 3 寄存器的值（预装载值）。

如果没有通过 TIMx_CCMR2 寄存器中的 OC3PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 3 寄存器中。实际捕获/比较寄存器中包含要与计数器 TIM3_CNT 进行比较并在 tim_oc3 输出上发出信号的值。

在非抖动模式下（DITHEN = 0）

该寄存器存储的是 CCR3[15:0]中的比较值。CCR3[19:16]位会被重置。

在抖动模式下（DITHEN = 1）

该寄存器存储的是 CCR3[19:4]中的整数部分。CCR3[3:0]位域存储的是抖动的部分。

若 CC3 通道配置为输入：

CCR3 为上一个输入捕获 3 事件（tim_ic3）发生时的计数器值。TIM3_CCR3 是只读的并且不可编程。

在非抖动模式下（DITHEN = 0）

CCR3[15:0]位存储了捕获值。CCR3[19:16]位会被重置。

在抖动模式下（DITHEN = 1）

该寄存器存储捕获在 CCR3[19:0]中。CCR3[3:0]位会被重置。

14.7.22 TIM2 捕获/比较寄存器 3（TIM2_CCR3）

偏移地址：0x3C

复位值：0x0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR3[31:16]															
rw	rw	rw	rw	rw	rw	rw	Rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:0 **CCR3[31:0]：**捕获/比较 3 值（Capture/Compare 3 value）

若 CC3 通道配置为输出：

CCR3 为要装载到实际捕获/比较 3 寄存器的值（预装载值）。

如果没有通过 TIMx_CCMR2 寄存器中的 OC3PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获

/比较 3 寄存器中。实际捕获/比较寄存器中包含要与计数器 TIM2_CNT 进行比较并在 tim_oc3 输出上发出信号的值。

在非抖动模式下（DITHEN = 0）

该寄存器存储的是 CCR3[27:0]中的比较值。CCR3[31:28]位会被重置。

在抖动模式下（DITHEN = 1）

该寄存器存储的是 CCR3[31:4]中的整数部分。CCR3[3:0]位域存储的是抖动的部分。

若 CC3 通道配置为输入：

CCR3 为上一个输入捕获 3 事件（tim_ic3）发生时的计数器值。TIM2_CCR3 是只读的并且不可编程。

在非抖动模式下（DITHEN = 0）

CCR3[27:0]位存储了捕获值。CCR3[31:28]位会被重置。

在抖动模式下（DITHEN = 1）

该寄存器存储捕获在 CCR3[31:0]中。CCR3[3:0]位会被重置。

14.7.23 TIM3 捕获/比较寄存器 4（TIM3_CCR4）

偏移地址：0x40

复位值：0x0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												CCR4[19:16]			
												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:20 保留，必须保持为复位值。

Bits 19:0 **CCR4[19:0]：**捕获/比较 4 值（Capture/Compare 4 value）

若 CC4 通道配置为输出：

CCR4 为要装载到实际捕获/比较 4 寄存器的值（预装载值）。

如果没有通过 TIMx_CCMR2 寄存器中的 OC4PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 4 寄存器中。实际捕获/比较寄存器中包含要与计数器 TIM3_CNT 进行比较并在 tim_oc4 输出上发出信号的值。

在非抖动模式下（DITHEN = 0）

该寄存器存储的是 CCR4[15:0]中的比较值。CCR4[19:16]位会被重置。

在抖动模式下（DITHEN = 1）

该寄存器存储的是 CCR4[19:4]中的整数部分。CCR4[3:0]位域存储的是抖动的部分。

若 CC4 通道配置为输入：

CCR4 为上一个输入捕获 4 事件（tim_ic4）发生时的计数器值。TIM3_CCR4 是只读的并且不可编程。

在非抖动模式下（DITHEN = 0）

CCR4[15:0]位存储了捕获值。CCR4[19:16]位会被重置。

在抖动模式下（DITHEN = 1）

该寄存器存储捕获在 CCR4[19:0]中。CCR4[3:0]位会被重置。

14.7.24 TIM2 捕获/比较寄存器 4（TIM2_CCR4）

偏移地址：0x40

复位值：0x0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR4[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:0 **CCR4[31:0]：**捕获/比较 4 值（Capture/Compare 4 value）

若 CC4 通道配置为输出：

CCR4 为要装载到实际捕获/比较 4 寄存器的值（预装载值）。

如果没有通过 TIMx_CCMR2 寄存器中的 OC4PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 4 寄存器中。实际捕获/比较寄存器中包含要与计数器 TIM2_CNT 进行比较并在 tim_oc4 输出上发出信号的值。

在非抖动模式下（DITHEN = 0）

该寄存器存储的是 CCR4[27:0]中的比较值。CCR4[31:28]位会被重置。

在抖动模式下（DITHEN = 1）

该寄存器存储的是 CCR4[31:4]中的整数部分。CCR4[3:0]位域存储的是抖动的部分。

若 CC4 通道配置为输入：

CCR4 为上一个输入捕获 4 事件（tim_ic4）发生时的计数器值。TIM2_CCR4 是只读的并且不可编程。

在非抖动模式下（DITHEN = 0）

CCR4[27:0]位存储了捕获值。CCR4[31:28]位会被重置。

在抖动模式下（DITHEN = 1）

该寄存器存储捕获在 CCR4[31:0]中。CCR4[3:0]位会被重置。

14.7.25 TIMx 编码器控制寄存器（TIMx_ECR）（x=2/3）

偏移地址：0x58

复位值：0x0000

访问：无等待，支持字，半字和字节访问

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

Bits 31:27 保留，必须保持为复位值。

Bits 26:24 **PWPRSC[2:0]：**脉冲宽度预分频器（Pulse width prescaler）

该位域设置脉冲发生器的时钟预分频，如下所示：

$t_{PWG} = (2^{(PWPRSC[2:0])}) \times t_{tim_ker_ck}$

Bits 23:16 **PW[7:0]：**脉冲宽度（Pulse width）

该位域定义脉冲持续时间，如下所示：

t_{PW} = PW[7:0] x t_{PWG}

Bits 15:8 保留，必须保持为复位值。

Bits 7:6 **IPOS[1:0]：索引位置**（Index positioning）

在正交编码器模式（SMS[3:0] = 0001，0010，0011，1110，1111）下，该位指示了索引事件在哪个 AB 输入配置下会重置计数器。

00：当 AB=00 时，索引重置计数器

01：当 AB=01 时，索引重置计数器

10：当 AB=10 时，索引重置计数器

11：当 AB=11 时，索引重置计数器

在方向时钟模式或时钟加方向模式（SMS[3:0] = 1010，1011，1100，1101）下，此位指示了索引事件在哪个时钟水平上重置计数器。在双向时钟模式下，这适用于两个时钟输入。具体为：

x0：当时钟=0 时，索引重置计数器

x1：当时钟=1 时，索引重置计数器

注意：IPOS[1]位不重要

Bit 5 **FIDX[1:0]：首位索引**（First index）

该位指示是否仅考虑首位索引

0：索引始终处于活跃状态

1：仅首个索引会重置计数器

Bits 4:3 保留，必须保持为复位值。

Bits 2:1 **IDIR[1:0]：索引方向**（Index direction）

该位指示索引事件在哪个方向上重置计数器。

00：无论方向如何，索引都会重置计数器

01：仅在递增计数时，索引会重置计数器

10：仅在递减计数时，索引会重置计数器

11：保留

注意：当 IE 位被复位（禁止索引）时，必须写入 IDR[1:0]位域

Bit 0 **IE：索引使能**（Index enable）

该位指示索引事件是否能重置计数器

0：禁止索引

1：使能索引

14.7.26 TIMx 定时器输入选择寄存器（TIMx_TISEL）（x=2/3）

偏移地址：0x5C

复位值：0x0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				TI4SEL[3:0]				Reserved							
				rw	rw	rw	rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				TI2SEL[3:0]				Reserved				TI1SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

Bits 31:28 保留，必须保持为复位值。

Bits 27:24 **TI4SEL[3:0]：选择 tim_ti4[0..2]输入**（Selects tim_ti4[0..2] input）

0000：tim_ti4_in0: TIMx_CH4
0001：tim_ti4_in1
0010：tim_ti4_in2
0011：保留

Bits 23:12 保留，必须保持为复位值。

Bits 11:8 **TI2SEL[3:0]：选择 tim_ti2[0..2]输入**（Selects tim_ti2[0..2] input）

0000：tim_ti2_in0: TIMx_CH2
0001：tim_ti2_in1
0010：tim_ti2_in2
0011：保留

Bits 7:4 保留，必须保持为复位值。

Bit 3: **TI1SEL[3:0]：选择 tim_ti1[0..3]输入**（Selects tim_ti1[0..3] input）

0000：tim_ti1_in0：TIMx_CH1
0001：tim_ti1_in1
0010：tim_ti1_in2
0011：tim_ti1_in3

14.7.27 TIMx 复用功能选择寄存器 1（TIMx_AF1）（x=2/3）

偏移地址：0x60

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														ETRSEL[3:2]	
														rw	rw

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

Bits 31:18 保留，必须保持为复位值。

Bits 17:14 **ETRSEL[3:0]：**etr_in 的源选择（etr_in source selection）

这些位选择 etr_in 的输入源
0000：tim_etr0: TIMx_ETR 输入
0001：tim_etr1
0010：tim_etr2
0011：reseved
请参考章节：TIM2/TIM3 引脚和内部信号，了解特定于产品的实现。

Bits 13:0 保留，必须保持为复位值。

14.7.28 TIMx 复用功能选择寄存器 2（TIMx_AF2）（x=2/3）

偏移地址：0x64

复位值：0x0000 0001

访问：无等待，支持字，半字和字节访问

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

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

Bits 31:19 保留，必须保持为复位值。

Bits 18:16 **OCRSEL[2:0]：**ocref_clr 的源选择（ocref_clr source selection）

这些位选择 ocref_clr 的输入源

000：tim_ocref_clr0

001：tim_ocref_clr1

请参考章节：TIM2/TIM3 引脚和内部信号，了解特定于产品的实现。

Bit 15:0 保留，必须保持为复位值。

14.7.29 TIMx 触发寄存器（TIMx_TRGI）（x=2/3）

偏移地址：0x80

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

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

Bits 15:1 保留，必须保持为复位值。

Bit 0 **SLAVE_SYNC：**同步模式，从机同步配置。

当从模式选择为触发模式时：

0：从机同步模式禁止

1：从机同步模式开启

15 通用定时器（TIM15）

15.1 TIM15 简介

TIM15 定时器包含一个 16 位自动重载计数器，该计数器由可编程预分频器驱动。

它可用于多种用途，包括测量输入信号的脉冲宽度（输入捕获）或生成输出波形（输出比较和 PWM 和带死区插入的互补 PWM）。使用定时器预分频器和 RCC 时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。TIM15 定时器是完全独立的，不与其他定时器共享任何资源。

15.2 TIM15 主要功能

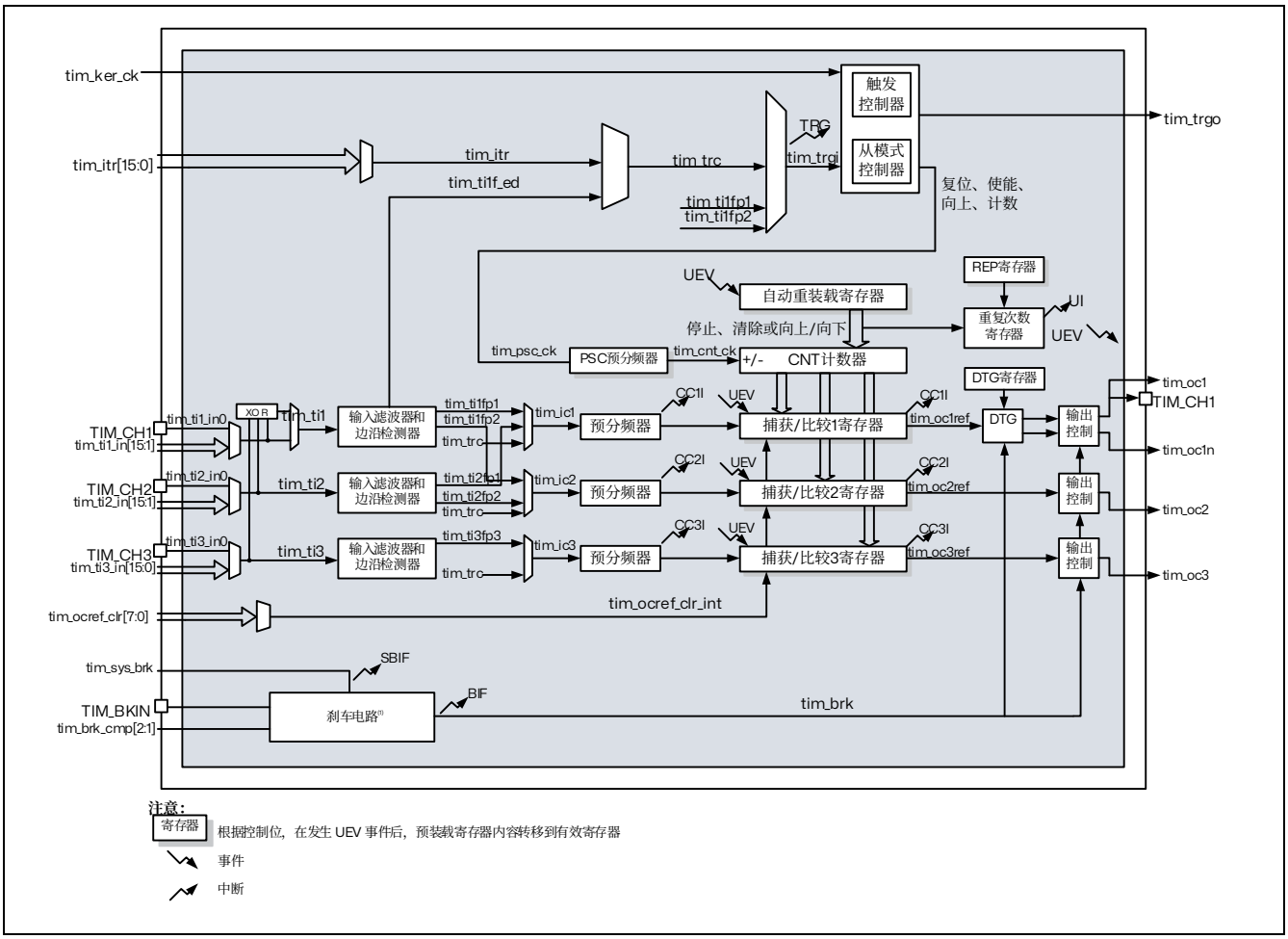
TIM15 定时器功能包括：

- 16 位自动装载计数器
- 16 位可编程（可以实时修改）预分频器，计数器时钟频率的分频系数为 1～65536 之间的任意数值
- 3 个独立通道：
  - 输入捕获
  - 输出比较
  - PWM 生成（边缘模式）
  - 单脉冲模式输出
- 带可编程死区的互补输出（仅限通道 1）
- 使用外部信号控制定时器和定时器互连的同步电路
- 重复计数器，用于仅在给定数目的计数器周期后更新定时器寄存器。
- 用于将定时器的输出信号置于复位状态或已知状态的刹车输入。
- 如下事件发生时产生中断：
  - 更新：计数器溢出，计数器初始化（通过软件或者内部/外部触发）
  - 触发事件（计数器启动、停止、初始化或者由内部/外部触发计数）
  - 输入捕获
  - 输出比较
  - 刹车输入（中断请求）

15.3 TIM15 功能描述

15.3.1 框图

图 15.1 TIM15 框图



注 1. 参考章节：使用刹车功能以获得细节

15.3.2 TIM15 引脚和内部信号

本节中的下两张表格总结了 TIM15 的输入和输出。

表 15.1 TIM15 输入/输出引脚

引脚名称	信号类型	描述
TIM_CH1 TIM_CH2 TIM_CH3	输入/输出	定时器多功能通道。 每个通道用于捕获、比较或 PWM。 TIM_CH1 和 TIM_CH2 也可用作外部时钟（tim_ker_ck 时钟的 1/4 以下）和外部触发器输入。
TIM_CH1N	输出	由 TIM_CH1 输出派生的定时器的互补输出， 具有插入死区时间的能力
TIM_BKIN	输入/输出	中断输入。此输入也可以配置为双向模式

表 15.2 TIM15 内部输入/输出信号

内部信号名称	信号类型	描述
tim_ti1_in[15:0] tim_ti2_in[15:0] tim_ti3_in[15:0]	输入	内部定时器输入总线。这些输入可被用作捕获或作为外部时钟 (tim_ker_ck 时钟的 1/4 以下)。
tim_itr[15:0]	输入	内部触发输入总线。这些输入可用于从模式控制器或作为输入时钟(tim_ker_ck 时钟的 1/4 以下)。
tim_trgo	输出	内部触发器输出。此触发器可以触发其他片上外设。
tim_ocref_clr[7:0]	输入	定时器 tim_ocref_clr 输入总线。这些输入可以用来清除 tim_ocxref 信号，通常用于硬件的逐周期脉冲宽度控制。
tim_brk_cmp[2:1]	输入	用于内部信号的刹车输入
tim_sys_brk[n:0]	输入	系统刹车输入。此输入收集 MCU 的系统级错误
tim_pclk	输入	定时器 APB 时钟
tim_ker_ck	输入	定时器内部时钟

如下四张表列出了连接到 tim_ti[3:1]输入多路复用器的源。

表 15.3 连接到 tim_ti1 的输入多路复用器

tim_ti1 输入	源
	TIM15
tim_ti1_in0	TIM15_CH1
tim_ti1_in1	cmp1_out
tim_ti1_in2	cmp2_out
tim_ti1_in3	LSI

表 15.4 连接到 tim_ti2 的输入多路复用器

tim_ti2 输入	源
	TIM2
tim_ti2_in0	TIM15_CH2

表 15.5 连接到 tim_ti3 的输入多路复用器

tim_ti3 输入	源
	TIM15
tim_ti3_in0	TIM15_CH3

下表列出连接到 tim_itr 输入多路复用器的内部源

表 15.6 TIM15 内部触发连接

tim_itrx inputs	TIM15
tim_itr0	Tim2_trgo
tim_itr1	Tim3_trgo
tim_itr2	Tim8_trgo

下表列出连接到 tim_brk 输入的源

表 15.7 TIM15 中断互连

tim_brk 输入	TIM15
TIM_BKIN	TIM15_BKIN 引脚
tim_brk_cmp1	cmp1_out
tim_brk_cmp2	comp2_out

表 15.8 系统中断互连

tim_sys_brk 输入	TIM15	SYSCFG_CFGR2 寄存器的使能位
tim_sys_brk0	内核锁定	CLL
tim_sys_brk1	Programmable Voltage Detector (PVD)	PVDL

下表列出连接到 tim_ocref_clr 输入多路复用器的内部源

表 15.9 连接到 ocref_clr 的输入多路复用器

定时器 tim_ocref_clr 信号	定时器 tim_ocref_clr 信号分配
	TIM15
tim_ocref_clr0	cmp1_out
tim_ocref_clr1	cmp2_out

15.3.3 时基单元

可编程定时器的主要模块由一个 16 位递增计数器及其相关的自动重装寄存器组成。计数器的时钟可通过预分频器进行分频。

计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。

时基单元包含：

- 计数器寄存器（TIM15_CNT）
- 预分频器寄存器（TIM15_PSC）
- 自动装载寄存器（TIM15_ARR）
- 重复计数器寄存器 (TIM15_RCR)

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以直接传送到影子寄存器，也可以在每次发生更新事件(UEV)时传送到影子寄存器，这取决于 TIM15_CR1 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到溢出并且 TIM15_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生 进行详细介绍。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIM15_CR1 寄存器中的计数器启动位（CEN）置 1 时，才会启动计数器（有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明）。

注意，计数器将在 TIM15_CR1 寄存器的 CEN 位置 1 时刻的一个时钟周期后开始计数。

预分频器描述

预分频器可对计数器时钟频率进行分频，分频系数介于 1 到 65536 之间。该预分频器基于 16 位寄存器（TIM15_PSC 寄存器）所控制的 16 位计数器。由于该控制寄存器具有缓冲功能，因此预分频器可实现实时更改。而新的预分频比将在下一更新事件发生时被采用.

下图给出了在预分频器运行时，更改计数器参数的例子。

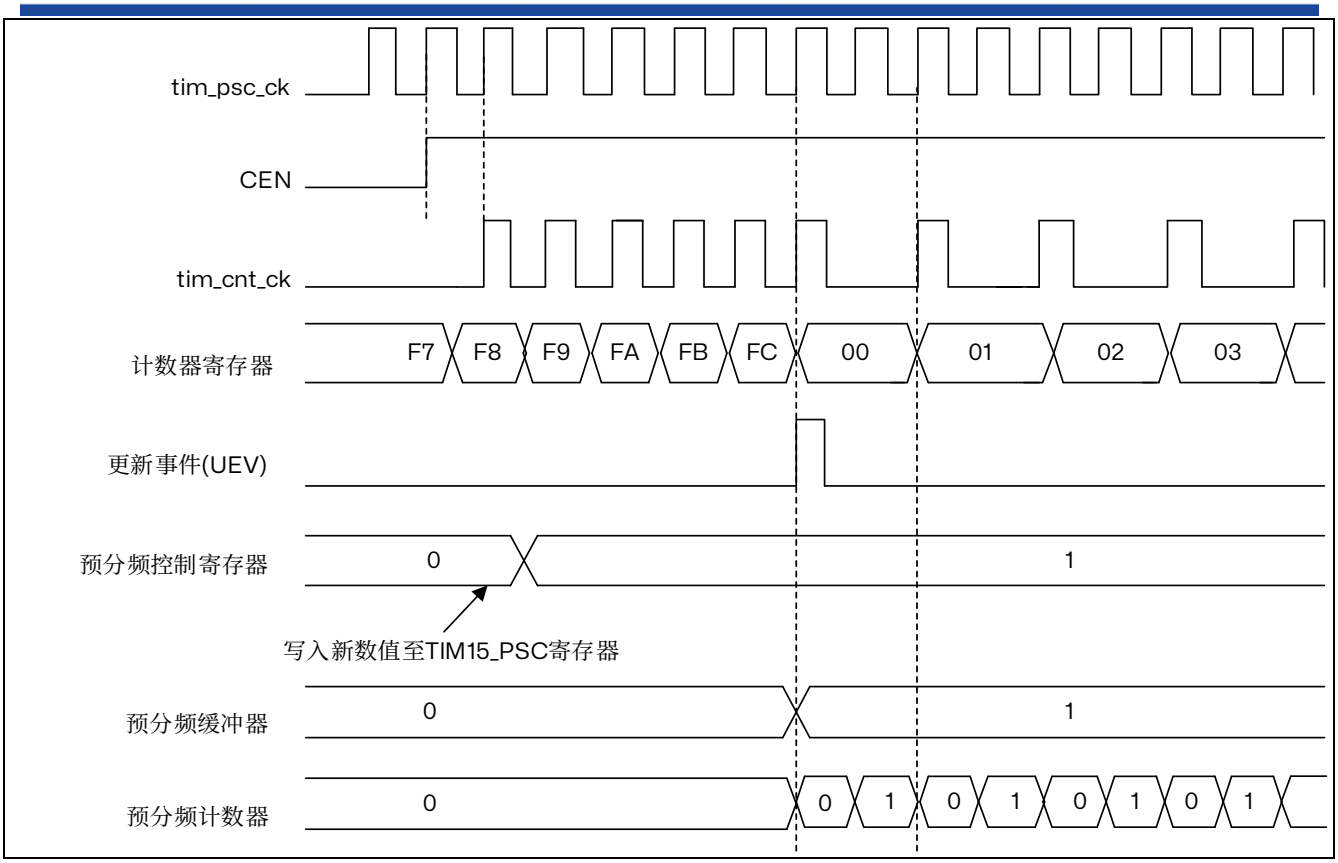


图 15.2 当预分频器的参数从 1 变到 2 时，计数器的时序图

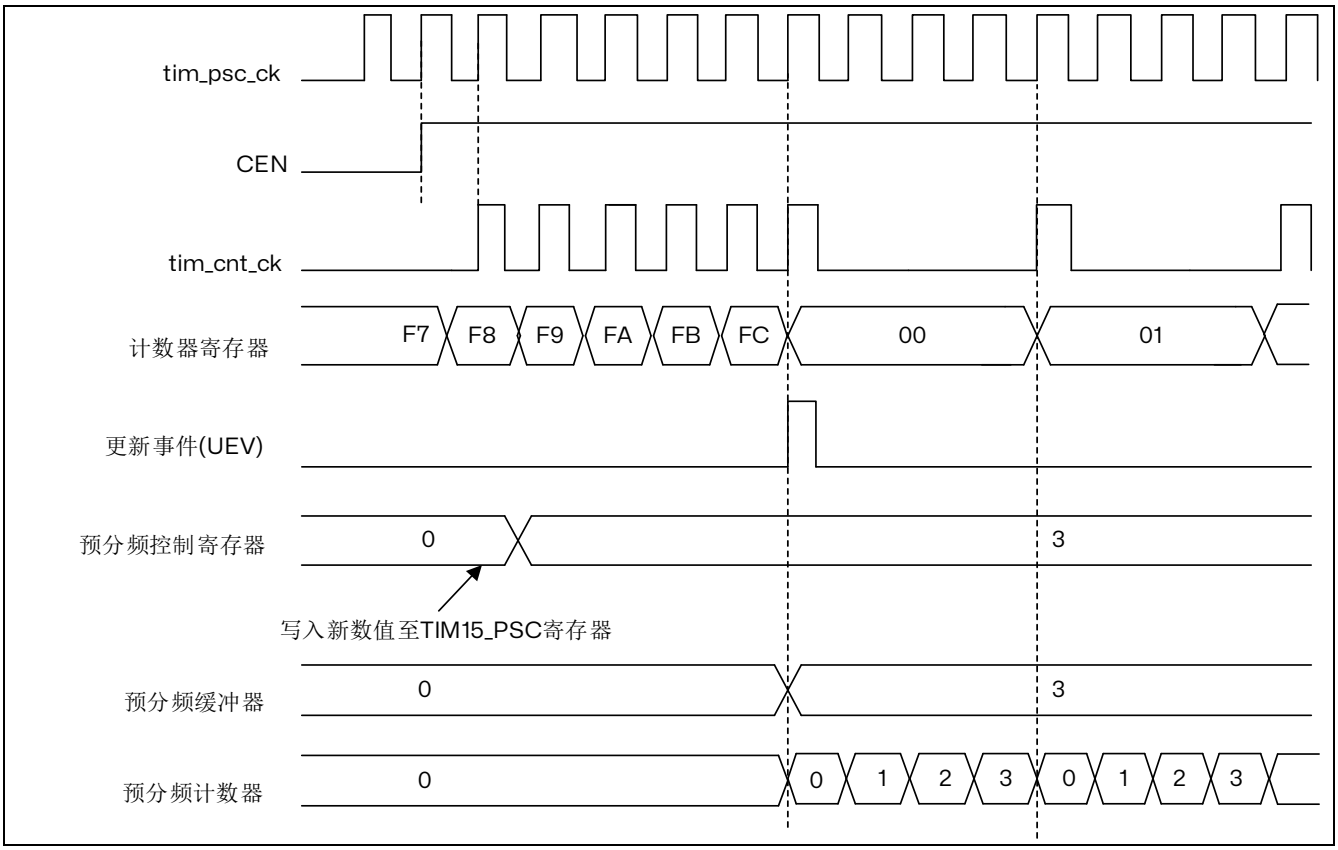


图 15.3 当预分频器的参数从 1 变到 4 时，计数器的时序图

15.3.4 计数器模式

递增计数模式

在递增计数模式下，计数器从 0 计数到自动重载值（TIM15_ARR 寄存器的内容），然后重新从 0 开始计数并生成计数器上溢事件。

如果使用重复计数器，则当递增计数的重复次数达到重复计数器寄存器中编程的次数加一次(TIM15_RCR+1)后，将生成更新事件（UEV)。否则，将在每次计数器上溢时产生更新事件。

每次发生计数器上溢时会生成更新事件，或将 TIM15_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。

通过软件将 TIM15_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变)。此外，如果 TIM15_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断)。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIM15_SR 寄存器中的 UIF 位）置 1（取决于 URS 位)：

- 重复计数器中将重新装载 TIM15_RCR 寄存器的内容
- 预分频器的缓冲区中将重新装载预装载值（TIM15_PSC 寄存器的内容）
- 自动重载影子寄存器将以预装载值进行更新

以下各图以一些示例说明当 TIM15_ARR=0x36 时不同时钟频率下计数器的行为。

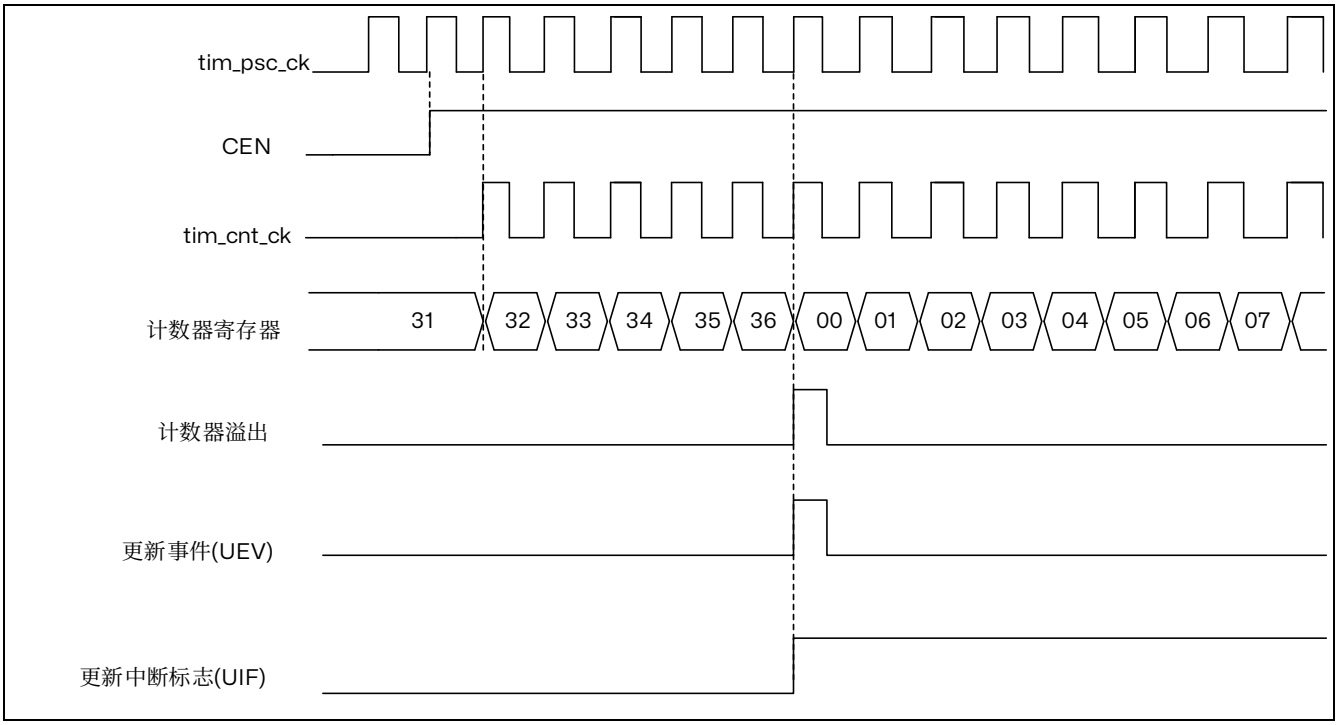


图 15.4 计数器时序图，内部时钟分频因子为 1

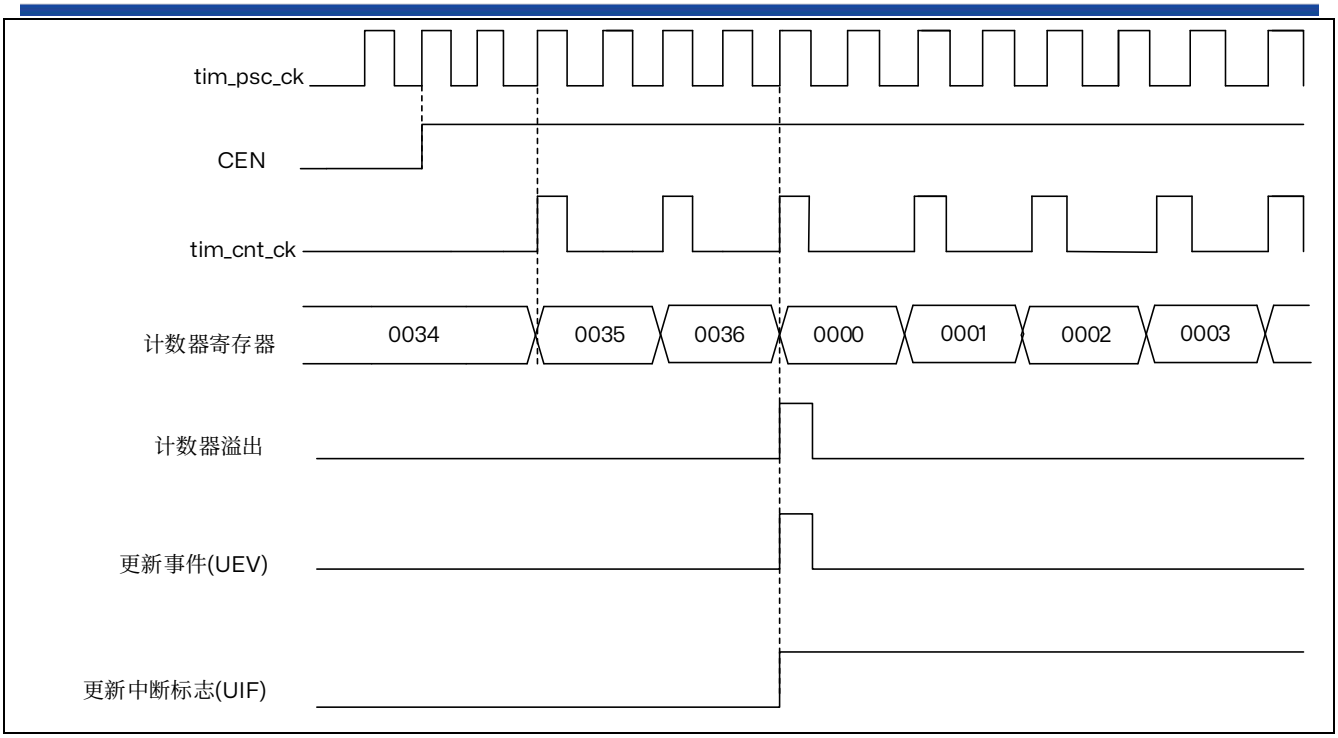


图 15.5 计数器时序图，内部时钟分频因子为 2

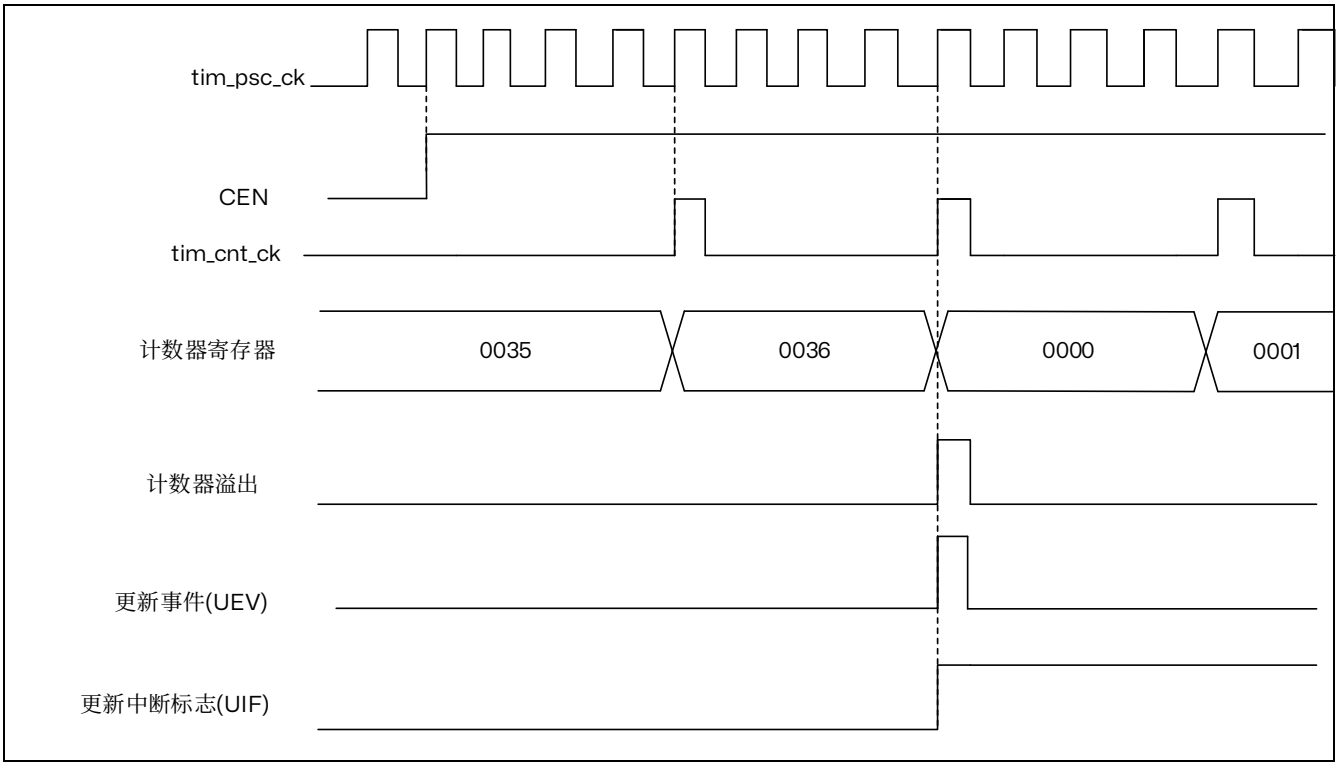


图 15.6 计数器时序图，内部时钟分频因子为 4

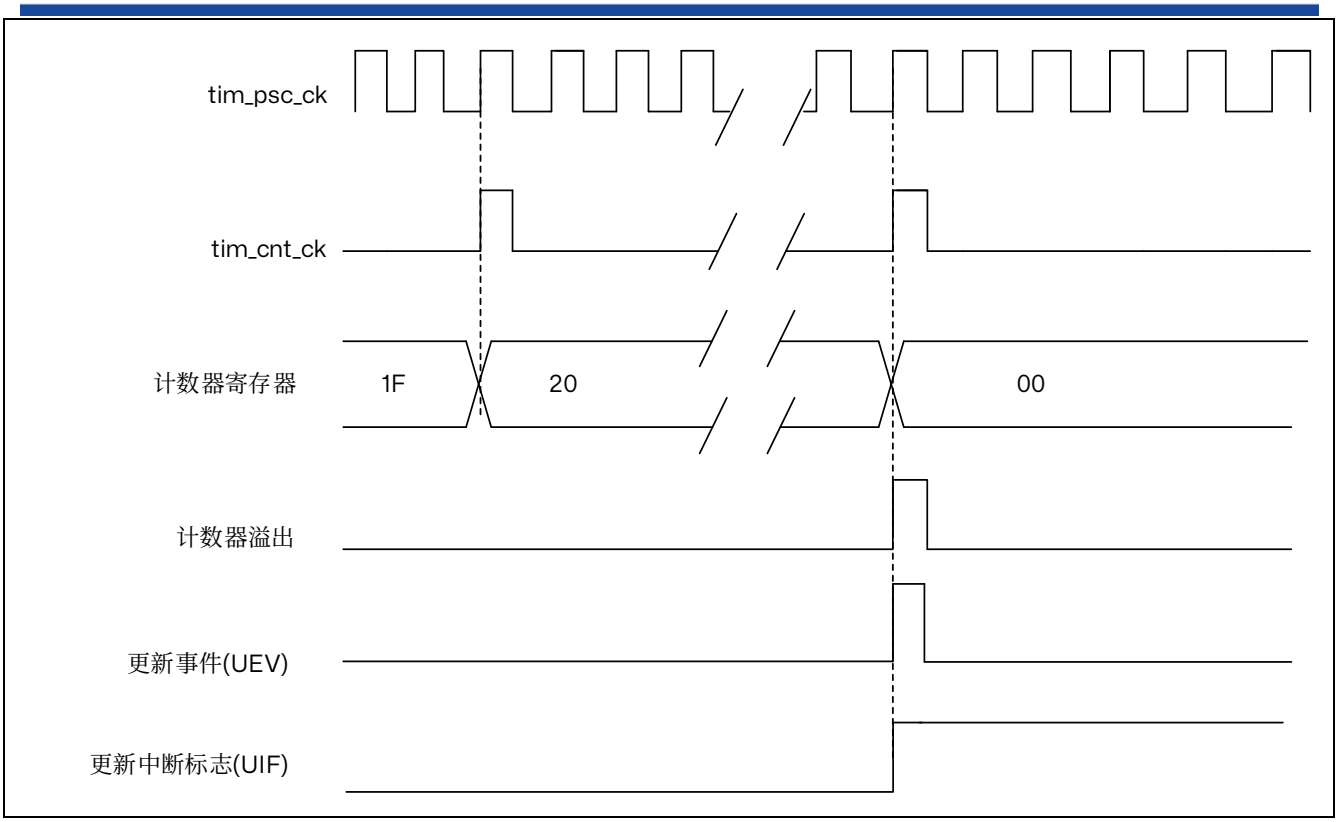


图 15.7 计数器时序图，内部时钟分频因子为 N

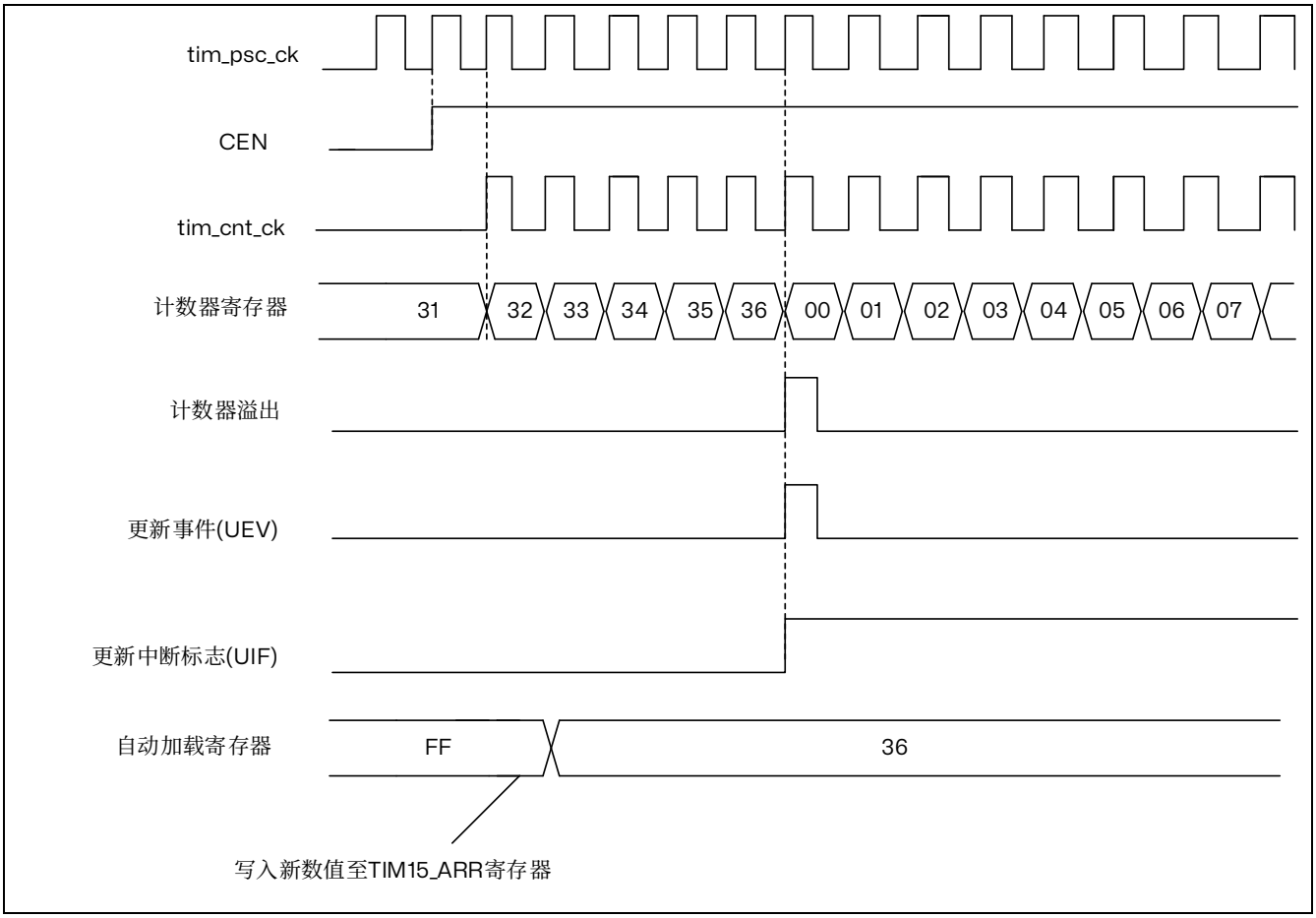


图 15.8 计数器时序图，当 ARPE=0 时的更新事件（TIM15_ARR 没有预装入）

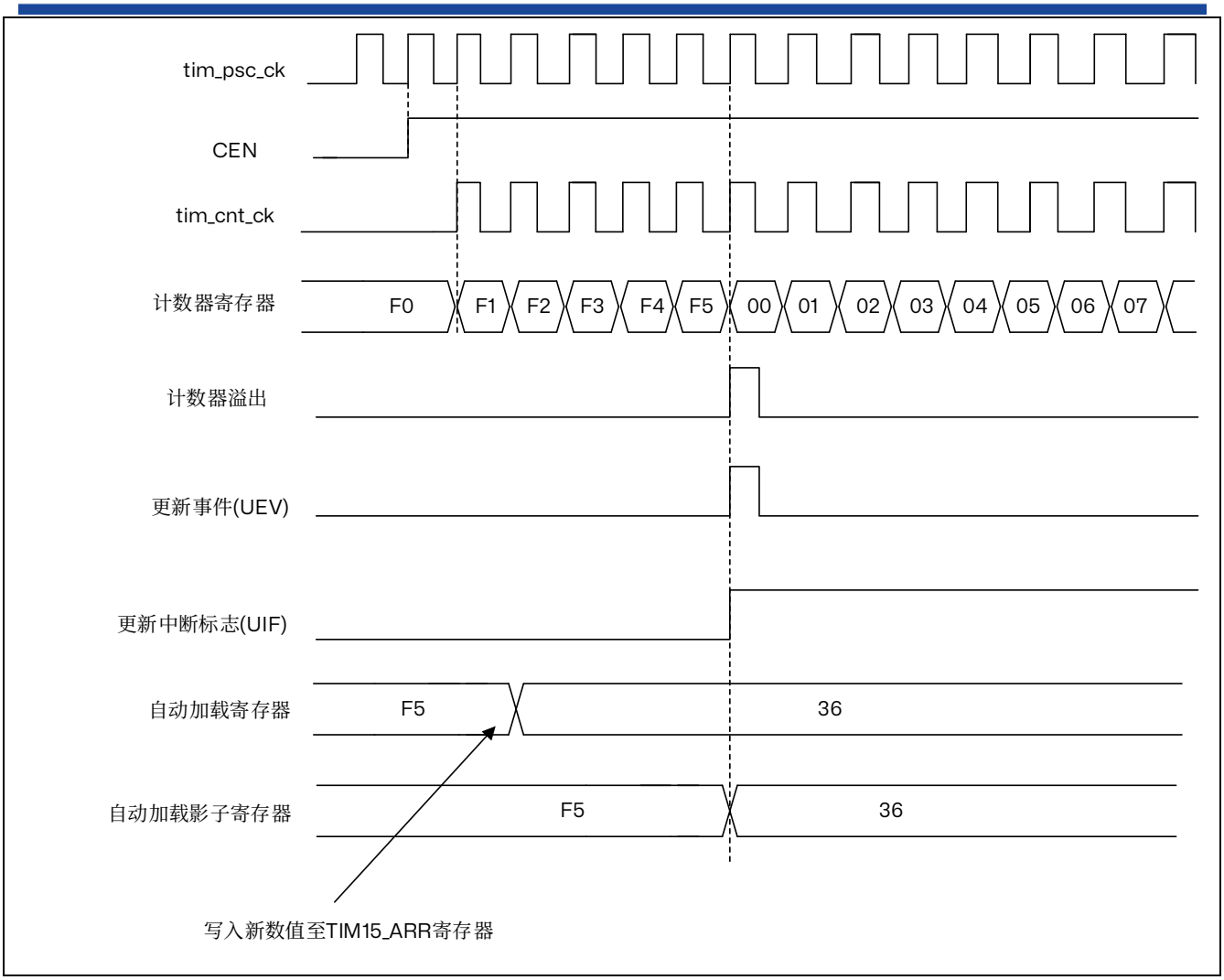


图 15.9 计数器时序图，当 ARPE=1 时的更新事件（预装入了 TIM15_ARR）

15.3.5 重复计数器

时基单元介绍如何因计数器溢出而生成更新事件(UEV)。实际上，只有当重复计数器达到零时，才会生成更新事件。这在生成 PWM 信号时很有用。

这意味着，每当发生 N 个计数器溢出（其中，N 是 TIM15_RCR 重复计数器寄存器中的值），数据就将从预装载寄存器转移到影子寄存器（TIM15_ARR 自动重载寄存器、TIM15_PSC 预分频器寄存器以及比较模式下的 TIM15_CCRx 捕获/比较寄存器）。

重复计数器在每次计数器溢出时递减：

重复计数器是自动重载类型；其重复率为 TIM15_RCR 寄存器所定义的值。当更新事件由软件（通过将 TIM15_EGR 寄存器的 UG 位置 1）或硬件（通过从模式控制器）生成时，无论重复计数器的值为多少，更新事件都将立即发生，并且在重复计数器中重新装载 TIM15_RCR 寄存器的内容。

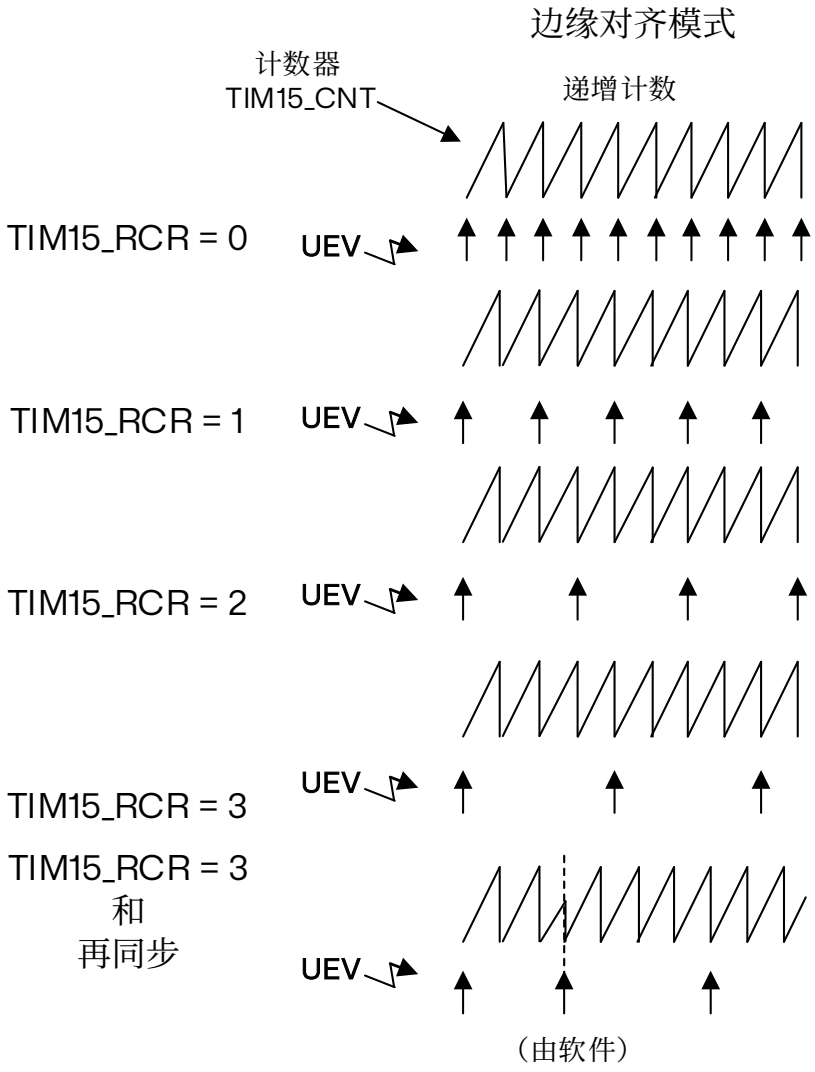


图 15.10 取决于模式和 TIM15_RCR 寄存器设置的更新频率示例

15.3.6 时钟选择

计数器时钟可由下列时钟源提供：

- 内部时钟（tim_ker_ck）
- 外部时钟模式 1：外部输入引脚（tim_ti1 or tim_ti2，如果可用）
- 内部触发输入(tim_itrx)：使用一个定时器作为另一个定时器的预分频器，例如可以将 TIM8 配置为 TIM15 的预分频器。

内部时钟源（tim_ker_ck）

如果禁止从模式控制器（TIM15_SMCR 寄存器中 SMS=000），则 CEN 位、DIR 位（TIM15_CR1 寄存器中）和 UG 位（TIM15_EGR 寄存器中）为实际控制位，并且只能通过软件进行更改（UG 除外，仍自动清零）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 tim_ker_ck 提供。

下图显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）。

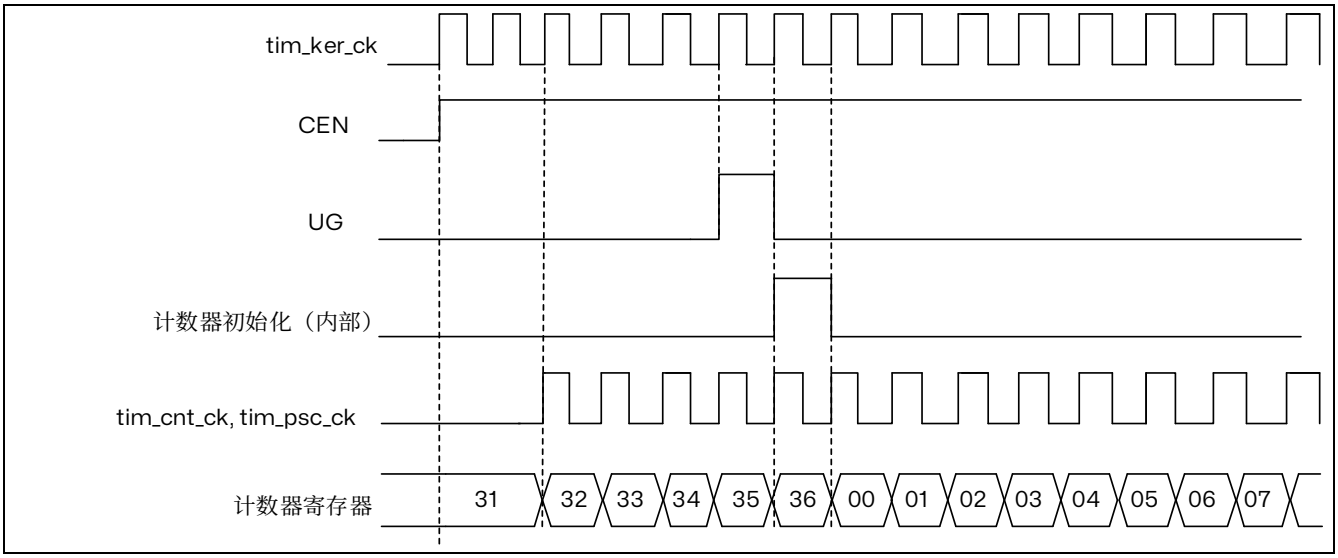


图 15.11 一般模式下的控制电路，内部时钟分频因子为 1

外部时钟源模式 1

当 TIM15_SMCR 寄存器中的 SMS=111 时，可选择此模式。计数器可在选定的输入信号上出现上升沿或下降沿时计数。

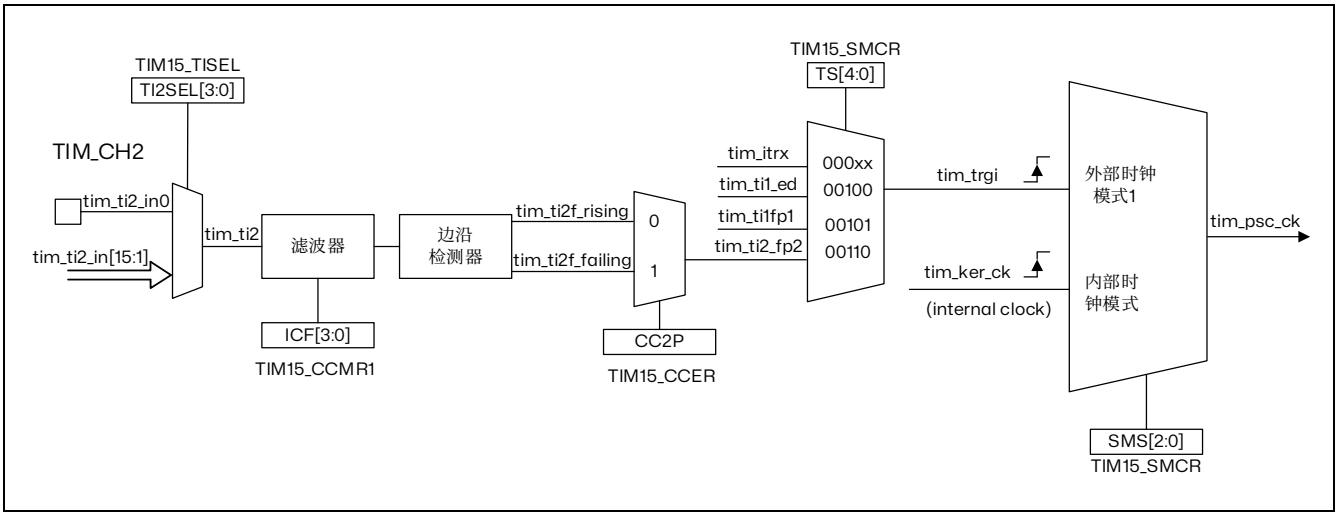


图 15.12 tim_ti2 外部时钟连接例子

例如，要使递增计数器在 tim_ti2 输入出现上升沿时计数，请执行以下步骤：

1. 通过在 TIM15_TISEL 寄存器中使用 TI2SEL[3:0]位选择合适的 tim_ti2_in[15:0]源（内部或外部）。
2. 通过在 TIM15_CCMR1 寄存器中写入 CC2S=“01”来配置通道 2，使其能够检测 tim_ti2 输入的上升沿。
3. 通过在 TIM15_CCMR1 寄存器中写入 IC2F[3:0]位来配置输入滤波时间（如果不需要任何滤波，请保持 IC2F=0000）。
4. 通过在 TIM15_CCER 寄存器中写入 CC2P=0 和 CC2NP=0 来选择上升沿极性。
5. 通过在 TIM15_SMCR 寄存器中写入 SMS=111，使定时器在外部时钟模式 1 下工作。

6. 通过在 TIM15_SMCR 寄存器中写入 TS=110 来选择 tim_ti2 作为输入源。
7. 通过在 TIM15_CR1 寄存器中写入 CEN=1 来使能计数器。

注：由于捕获预分频器不用于触发操作，因此无需对其进行配置。

当 tim_ti2 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。

tim_ti2 的上升沿与实际计数器时钟之间的延迟是由于 tim_ti2 输入的重新同步电路引起的。

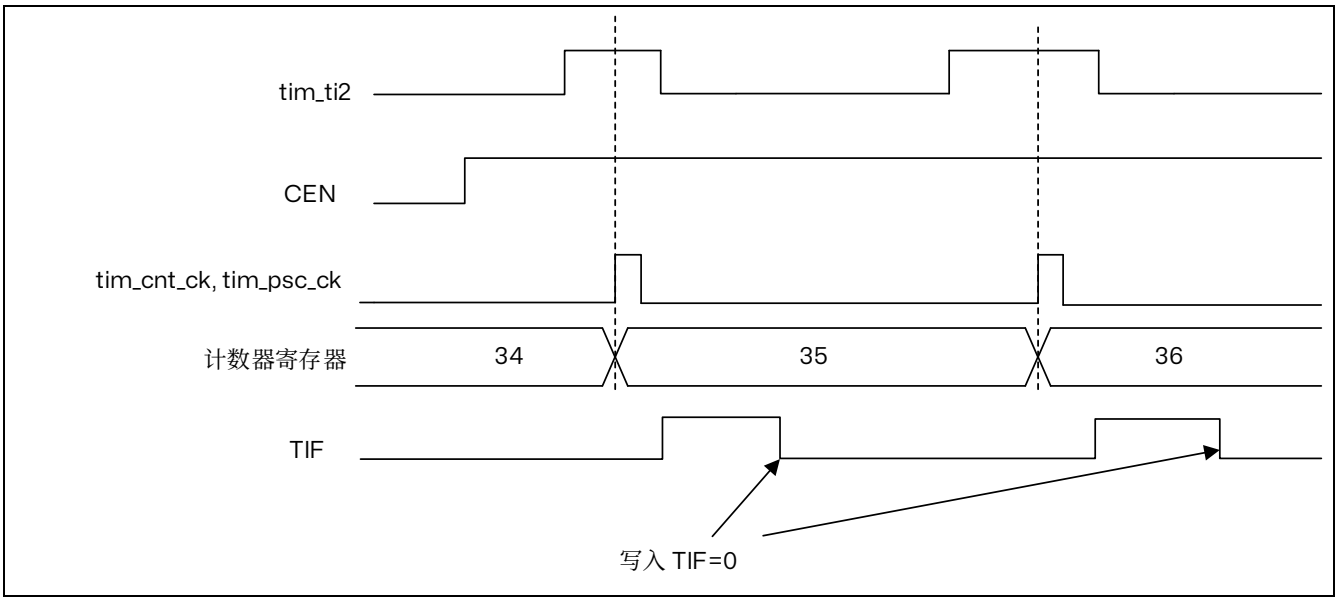


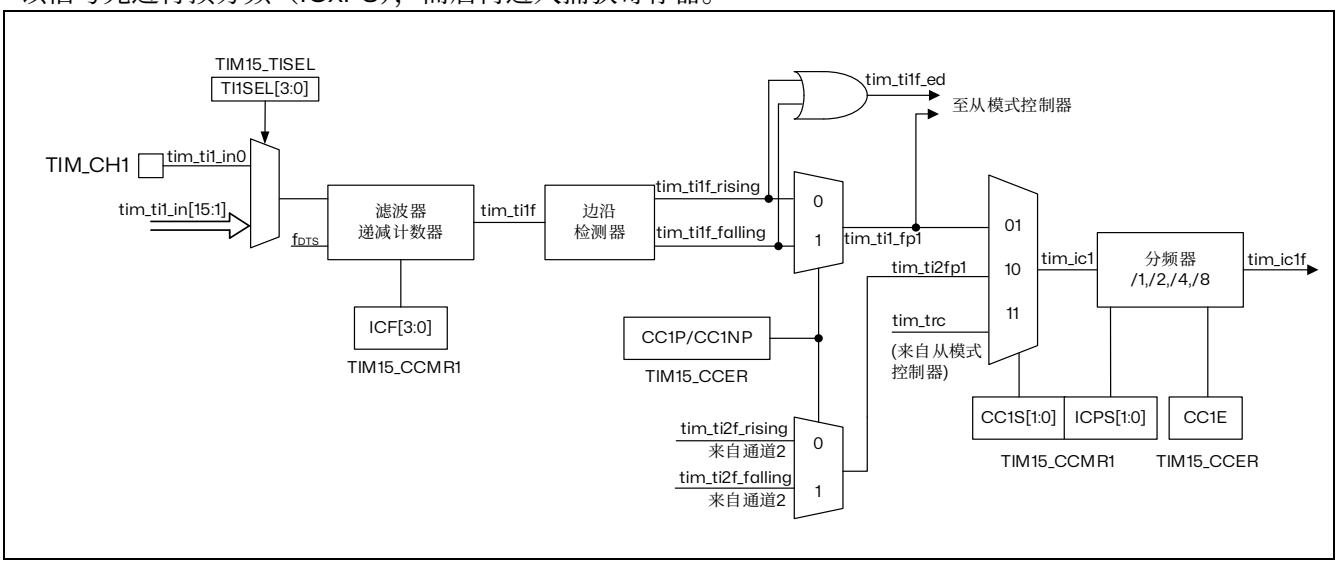
图 15.13 外部时钟模式 1 下的控制电路

15.3.7 捕获/比较通道

每个捕获/比较通道均围绕一个捕获/比较寄存器（包括一个影子寄存器）、一个捕获输入阶段（数字滤波、多路复用和预分频器）和一个输出阶段（比较器和输出控制）构建而成。

下图概括介绍了一个捕获/比较通道。

输入阶段对相应的 tim_tix 输入进行采样，生成一个滤波后的信号 tim_tixf。然后，带有极性选择功能的边沿检测器生成一个信号（tim_tixfpy），该信号可用作从模式控制器的触发输入，也可用作捕获命令。该信号先进行预分频（ICxPS），而后再进入捕获寄存器。



输出阶段生成一个中间波形作为基准：OCxRef（高电平有效）。链的末端决定最终输出信号的极性。

图 15.14 捕获/比较通道（如：通道 1 输入部分）

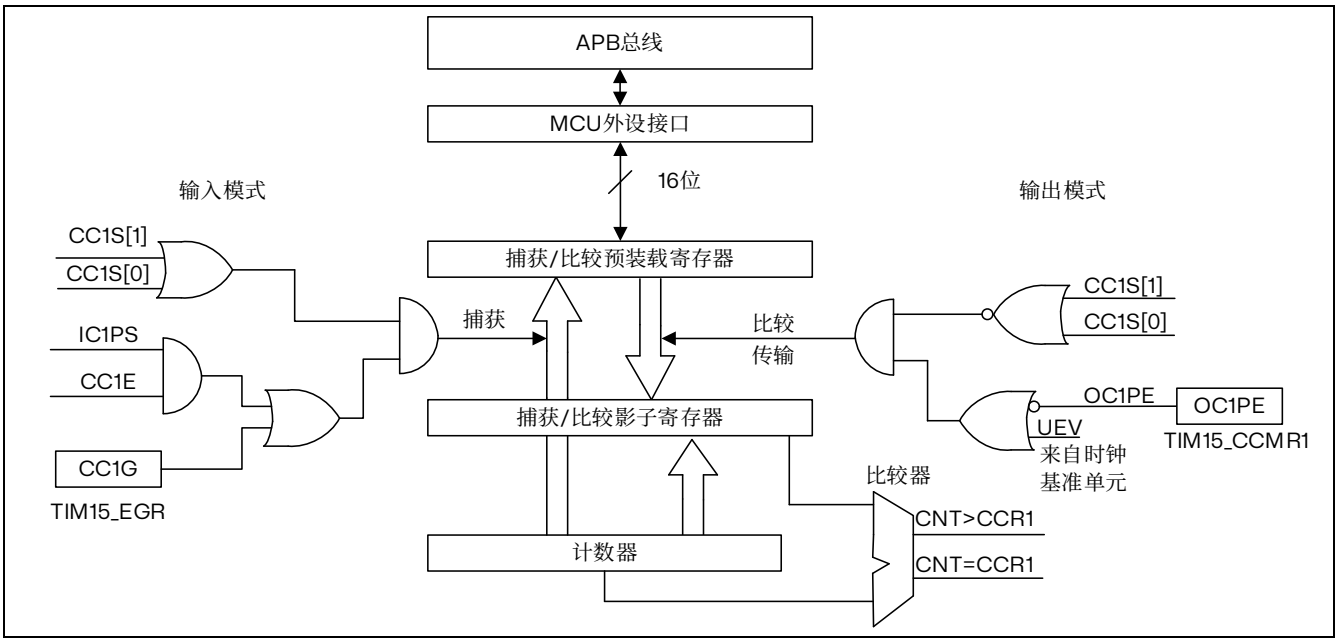


图 15.15 捕获/比较通道 1 的主电路

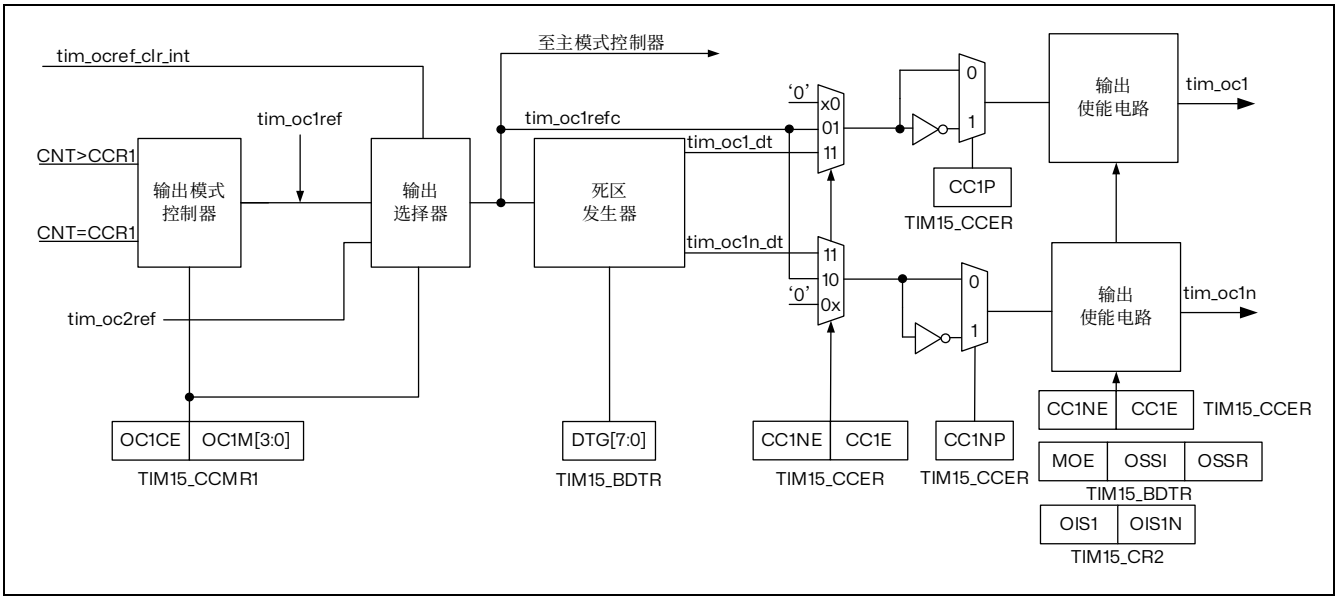


图 15.16 捕获/比较通道的输出阶段（通道 1）

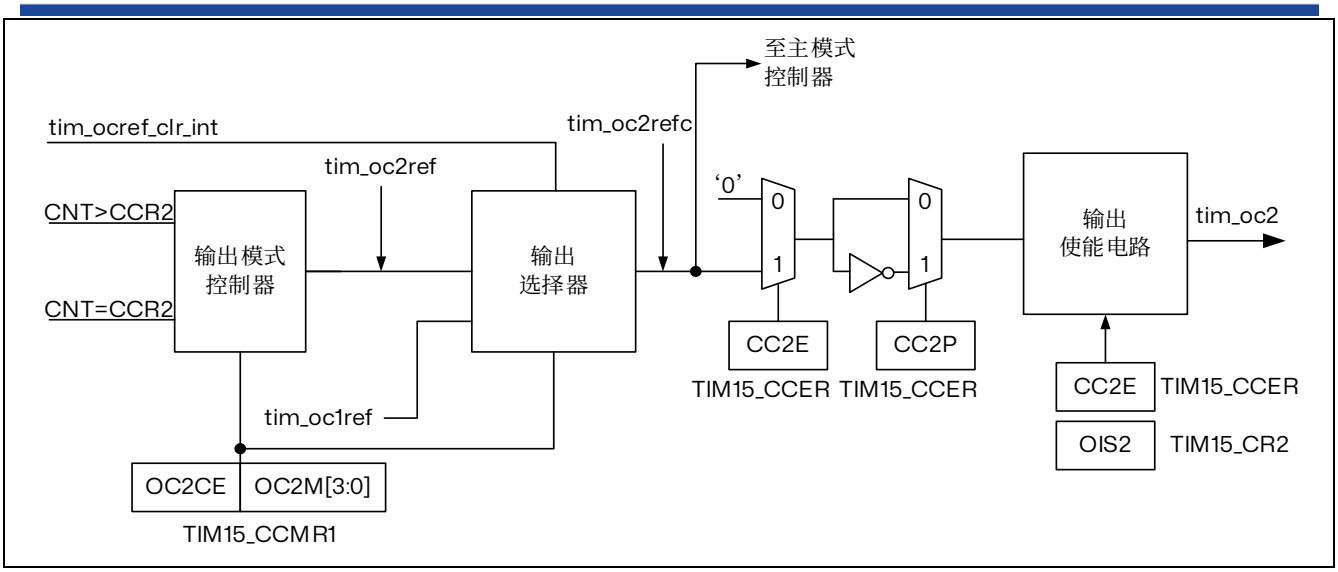


图 15.17 捕获比较通道的输出阶段（通道 2，通道 3 同上）

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。始终可通过读写操作访问预装载寄存器。

在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容复制到预装载寄存器中。

在比较模式下，预装载寄存器的内容将复制到影子寄存器中，然后将影子寄存器的内容与计数器进行比较。

15.3.8 输入捕获模式

在输入捕获模式下，当相应的 ICx 信号检测到跳变沿后，将使用捕获/比较寄存器(TIM15_CCRx)来锁存计数器的值。发生捕获事件时，会将相应的 CCXIF 标志（TIM15_SR 寄存器）置 1，并可发送中断请求（如果已使能）。如果发生捕获事件时 CCxIF 标志已处于高位，则会将重复捕获标志 CCxOF（TIM15_SR 寄存器）置 1。可通过软件向 CCxIF 写入 0 来给 CCxIF 清零，或读取存储在 TIM15_CCRx 寄存器中的已捕获数据。向 CCxOF 写入 0 后会将其清零。

以下示例说明了如何在 tim_ti1 输入出现上升沿时将计数器的值捕获到 TIM15_CCR1 中。具体操作步骤如下：

1. 通过在 TIM15_TISEL 寄存器中使用 TI1SEL[3:0]位选择合适的 tim_tix_in[15:1]源（内部或外部）。
2. 选择有效输入：TIM15_CCR1 必须连接到 tim_ti1 输入，因此向 TIM15_CCMR1 寄存器中的 CC1S 位写入 01。只要 CC1S 不等于 00，就会将通道配置为输入模式，并且 TIM15_CCR1 寄存器将处于只读状态。
3. 根据连接到定时器的信号，对所需的输入滤波时间进行编程（如果输入为 tim_tix 输入之一，则对 TIM15_CCMRx 寄存器中的 ICxF 位进行编程）。假设信号变化时，输入信号最多在 5 个内部时钟周期内发生抖动。因此，我们必须将滤波时间设置为大于 5 个内部时钟周期。在检测到 8 个具有新电平的连续采样（以 f_{DTS} 频率采样）后，可以确认 tim_ti1 上的跳变沿。然后向 TIM15_CCMR1 寄存器中的 IC1F 位写入 0011。
4. 通过向 TIM15_CCER 寄存器中的 CC1P 位和 CC1NP 位写入 0，选择 tim_ti1 通道的有效转换边沿（本例中为上升沿）。
5. 对输入预分频器进行编程。在本例中，我们希望每次有效转换时都执行捕获操作，因此需要禁止预分频器（向 TIM15_CCMR1 寄存器中的 IC1PS 位写入 00）。

6. 通过将 TIM15_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
7. 如果需要，可通过将 TIM15_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求。

发生输入捕获时：

- 发生有效跳变沿时，TIM15_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。
- 根据 CC1IE 位生成中断。

要处理重复捕获，建议在读出捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出标志之后与读取数据之前可能出现的重复捕获信息。

注：通过软件将 TIM15_EGR *寄存器中的相应* CCxG *位置*1 *可生成* IC *中断请求。*

15.3.9 PWM 输入模式

此模式允许测量连接到单个 tim_tix 输入的 PWM 信号的周期和占空比：

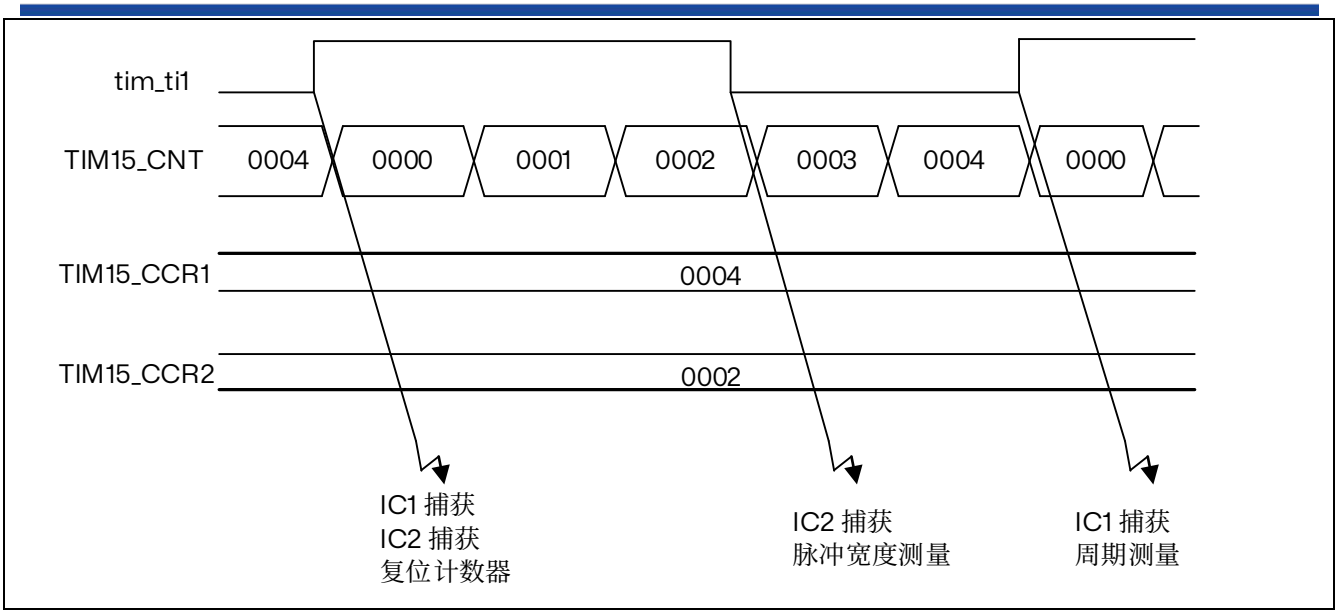
- TIM15_CCR1 寄存器保持周期值（两个连续上升沿之间的间隔）。
- TIM15_CCR2 寄存器保持脉冲宽度（两个连续上升沿和下降沿之间的间隔）。

此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

- 两个 tim_icx 信号被映射至同一个 tim_tix 输入。
- 这两个 tim_icx 信号在边沿处有效，但极性相反。
- 选择两个 tim_tixfpy 信号之一作为触发输入，并将从模式控制器配置为复位模式。

可通过以下步骤测量施加在 tim_ti1 上的 PWM 信号的周期和脉冲宽度：

1. 通过在 TIM15_TISEL 寄存器中使用 TI1SEL[3:0]位选择合适的 tim_tix_in[15:0]源（内部或外部）。
2. 选择 TIM15_CCR1 的有效输入：向 TIM15_CCMR1 寄存器中的 CC1S 位写入 01（选择 tim_ti1）。
3. 选择 tim_ti1fp1 的有效极性（用于 TIM15_CCR1 中的捕获和计数器清零）：向 CC1P 位和 CC1NP 位写入“0”（上升沿有效）。
4. 选择 TIM15_CCR2 的有效输入：向 TIM15_CCMR1 寄存器中的 CC2S 写入 10（选择 tim_ti1）。
5. 选择 tim_ti1fp2 的有效极性（用于 TIM15_CCR2 中的捕获）：向 CC2P 位写入 1 和 CC2NP 位写入“0”（下降沿有效）。
6. 选择有效触发输入：向 TIM15_SMCR 寄存器中的 TS 位写入 101（选择 tim_ti1fp1）。
7. 将从模式控制器配置为复位模式：向 TIM15_SMCR 寄存器中的 SMS 位写入 100。
8. 使能捕获：向 TIM15_CCER 寄存器中的 CC1E 位和 CC2E 位写入“1”。



PWM 输入模式只能与 TIM15_CH1/TIM15_CH2 信号一起使用，因为只有 tim_ti1fp1 和 tim_ti2fp2 连接到从模式控制器。

图 15.18 PWM 输入模式时序

15.3.10 强制输出模式

在输出模式（TIM15_CCMRx 寄存器中的 CCxS 位 = 00）下，可直接由软件将每个输出比较信号（tim_ocxref 和 tim_ocx）强制设置为有效电平或无效电平，而无需考虑输出比较寄存器和计数器之间的任何比较结果。

要将输出比较信号（tim_ocxref/tim_ocx）强制设置为有效电平，只需向相应 TIM15_CCMRx 寄存器中的 OCxM 位写入 101。tim_ocxref 进而强制设置为高电平（tim_ocxref 始终为高电平有效），同时 tim_ocx 获取 CCxP 极性位的相反值。

例如：CCxP=0（tim_ocx 高电平有效）=> tim_ocx 强制设置为高电平。

通过向 TIM15_CCMRx 寄存器中的 OCxM 位写入 100，可将 tim_ocxref 信号强制设置为低电平。

无论如何，TIM15_CCRx 影子寄存器与计数器之间的比较仍会执行，而且允许将标志置 1。因此可发送相应的中断请求。输出比较模式一节对此进行了介绍。

15.3.11 输出比较模式

此功能用于控制输出波形，或指示已经过某一时间段。

当捕获/比较寄存器与计数器之间相匹配时，输出比较功能：

- 将为相应的输出引脚分配一个可编程值，该值由输出比较模式（TIM15_CCMRx 寄存器中的 OCxM 位）和输出极性（TIM15_CCER 寄存器中的 CCxP 位）定义。匹配时，输出引脚既可保持其电平（OCXM=000），也可设置为有效电平（OCXM=001）、无效电平（OCXM=010）或进行翻转（OCxM=011）。
- 将中断状态寄存器中的标志置 1（TIM15_SR 寄存器中的 CCxIF 位）。
- 如果相应中断使能位（TIM15_DIER 寄存器中的 CCXIE 位）置 1，将生成中断。

使用 TIM15_CCMRx 寄存器中的 OCxPE 位，可将 TIM15_CCRx 寄存器配置为带或不带预装载寄存器。

在输出比较模式下，更新事件 UEV 对 tim_ocxref 和 tim_ocx 输出毫无影响。同步的精度可以达到计

数器的一个计数周期。输出比较模式也可用于输出单脉冲（在单脉冲模式下）。

步骤：

1. 选择计数器时钟（内部、外部、预分频器）。
2. 在 TIM15_ARR 和 TIM15_CCRx 寄存器中写入所需数据。
3. 如果要生成中断请求，将 CCxIE 位和/或 CCxDE 位置 1。
4. 选择输出模式。例如，
   a) 写入 OCxM=0011 以在 CNT 匹配 CCRx 时翻转 tim_ocx 输出引脚
   b) 写入 OCxPE=0 以禁用预装载寄存器
   c) 写入 CCxP=0 以选择高电平有效
   d) 写入 CCxE=1 以使能输出
5. 通过将 TIM15_CR1 寄存器中的 CEN 位置 1 来使能计数器。

可随时通过软件更新 TIM15_CCRx 寄存器以控制输出波形，前提是未使能预装载寄存器（OCxPE=0，否则仅当发生下一个更新事件 UEV 时，才会更新 TIM15_CCRx 影子寄存器）。

下图列出了相关示例。

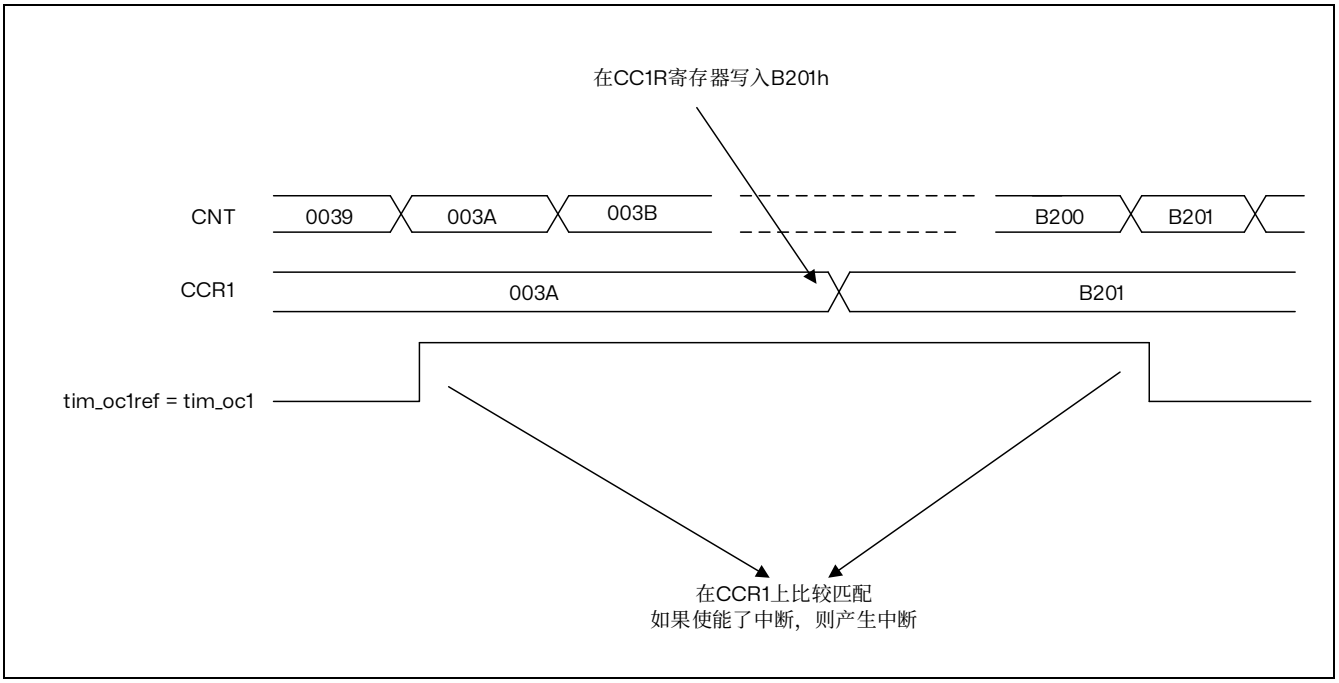


图 15.19 输出比较模式，翻转 tim_oc1

15.3.12 PWM 模式

脉冲宽度调制模式可以生成一个信号，该信号频率由 TIM15_ARR 寄存器值决定，其占空比则由 TIM15_CCRx 寄存器值决定。

通过向 TIM15_CCMRx 寄存器中的 OCxM 位写入 110（PWM 模式 1）或 111（PWM 模式 2），可以独立选择各通道（每个 tim_ocx 输出对应一个 PWM）的 PWM 模式。必须通过将 TIM15_CCMRx 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器，最后通过将 TIM15_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器。

由于只有在发生更新事件时预装载寄存器才会传送到影子寄存器，因此启动计数器之前，必须通过将 TIM15_EGR 寄存器中的 UG 位置 1 来初始化所有寄存器。

tim_ocx 极性可使用 TIM15_CCER 寄存器的 CCxP 位来编程。既可以设为高电平有效，也可以设为低电平有效。通过 CCxE、CCxNE、MOE、OSSI 和 OSSR 位（TIM15_CCER 和 TIM15_BDTR 寄存器）的组合使能 tim_ocx 输出。有关详细信息，请参见 TIM15_CCER 寄存器说明。

在 PWM 模式（1 或 2）下，TIM15_CNT 始终与 TIM15_CCRx 进行比较，以确定是 TIM15_CCRx ≤ TIM15_CNT 还是 TIM15_CNT ≤ TIM15_CCRx（取决于计数器计数方向）。

TIM15 只能递增计数，具体请参考递增计数模式。

以下以 PWM 模式 1 为例。只要 TIM15_CNT < TIM15_CCRx，PWM 参考信号 tim_ocxref 便为高电平，否则为低电平。如果 TIM15_CCRx 中的比较值大于自动重载值（TIM15_ARR 中），则 tim_ocxref 保持为“1”。如果比较值为 0，则 tim_ocxref 保持为“0”。下图举例介绍边沿对齐模式的一些 PWM 波形 (TIM15_ARR=8)。

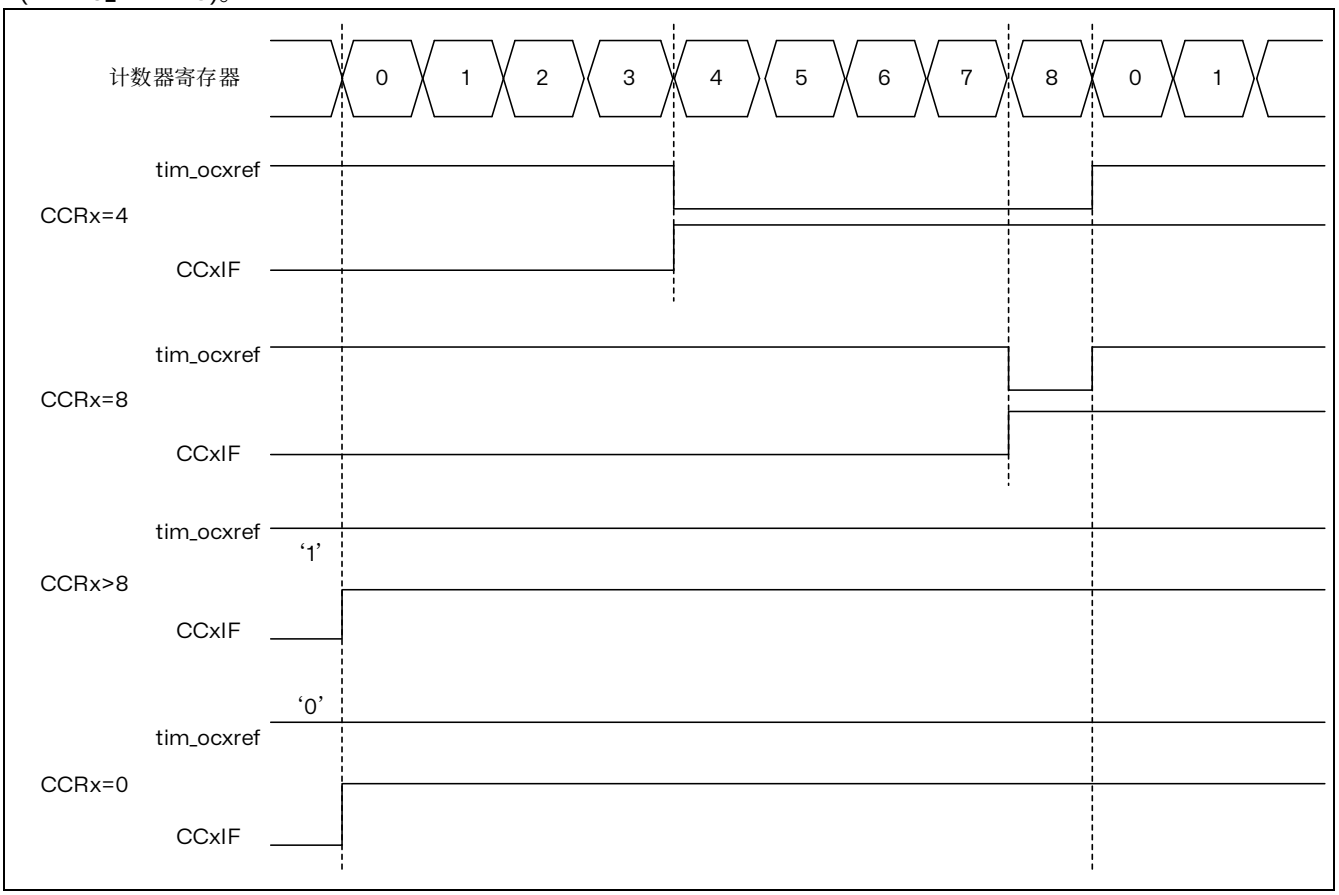


图 15.20 边沿对齐的 PWM 波形（ARR=8）

抖动模式

通过使能抖动模式，使用 TIM15_CR1 寄存器中的 DITHEN 位，可以提高 PWM 模式的有效分辨率。这既适用于 CCR（用于增加占空比分辨率），也适用于 ARR（用于增加 PWM 频率分辨率）。

工作原理是在 16 个连续的 PWM 周期中，以预定义的模式使实际的 CCR（或 ARR）值略有变化（增加或减少一个定时器时钟周期）。这使得占空比或 PWM 期间的平均值提高了 16 倍分辨率。下图展示了应用于 4 个连续 PWM 周期的抖动原理。

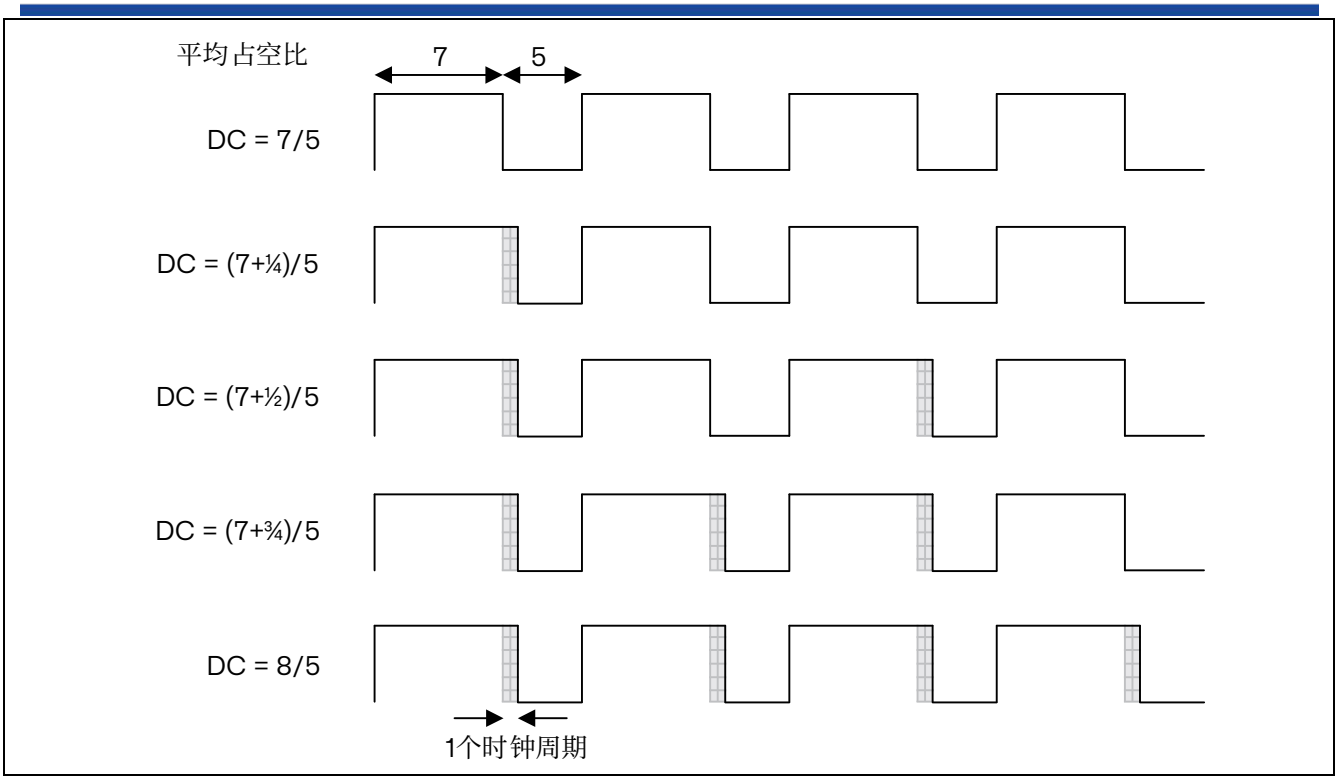


图 15.21 抖动原理

当抖动模式被使能时，寄存器编码将如下所示进行变化（参见下图示例）

- 4 个最低有效位（LSB）用于编码增强的分辨率部分（小数部分）
- 最高有效位（MSB）左移至 19:4 位，用于编码基本值

*注意：如果设置或清除*DITHEN *位，*ARR *和*CCR *值将自动更新。（例如，如果*ARR=0x05 *且*DITHEN=0，*它将更新为*ARR=0x50 *且*DITHEN=1*）。*

1. 必须重置 CEN 和 ARPE 位
2. ARR[3:0]位必须重置
3. DITHEN 位必须重置
4. CCIF 标志必须清除
5. 可以设置 CEN 位（最终设置 ARPE=1）。

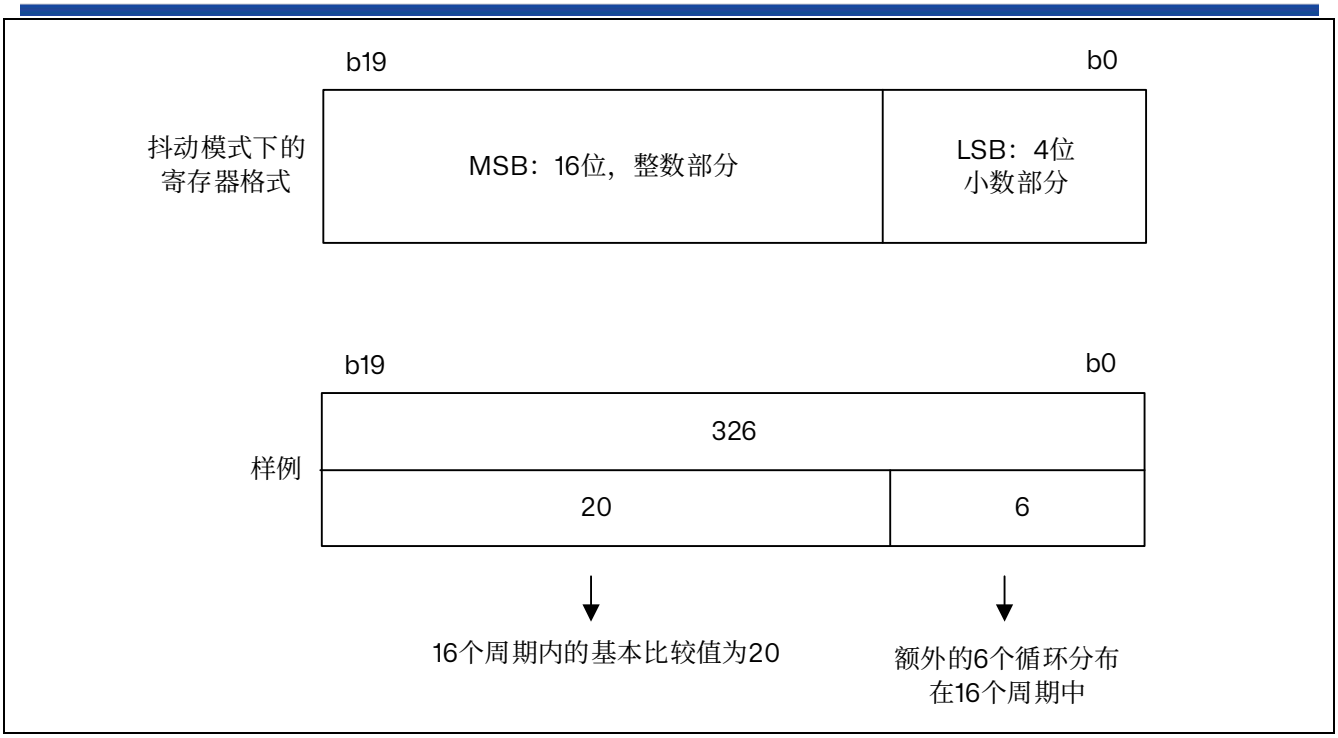


图 15.22 抖动模式下的数据格式和寄存器编码

最小频率由以下公式给出：

$$\text{分辨率} = \frac{\text{FTim}}{\text{Fpwm}} \Rightarrow \text{FpwmMin} = \frac{\text{FTim}}{\text{最大分辨率}}$$

$$\text{禁用抖动模式：FpwmMin} = \frac{\text{FTim}}{65536}$$

$$\text{使能抖动模式：FpwmMin} = \frac{\text{FTim}}{65535 + \frac{15}{16}}$$

*注意：在抖动模式下，*TIM15_ARR *和*TIM15CCRy *的最大值限制为*0xFFFEF*（对应于整数部分的*65534 *和抖动部分的*15*）。*

如下图所示，无论 PWM 频率如何，抖动模式都可以增加 PWM 分辨率。

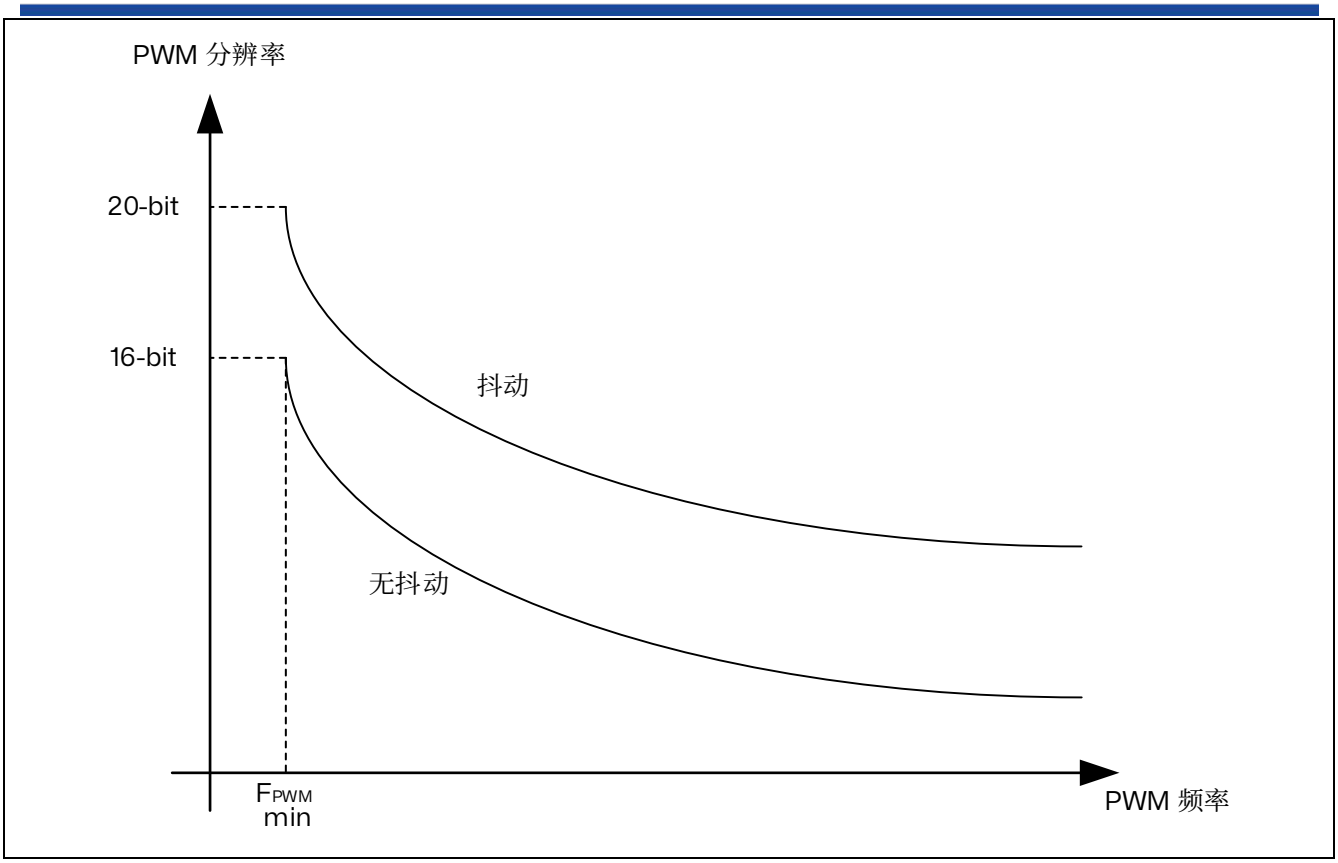


图 15.23 PWM 分辨率 vs 频率

如下图所示，占空比和/或周期的变化分散在 16 个连续周期。

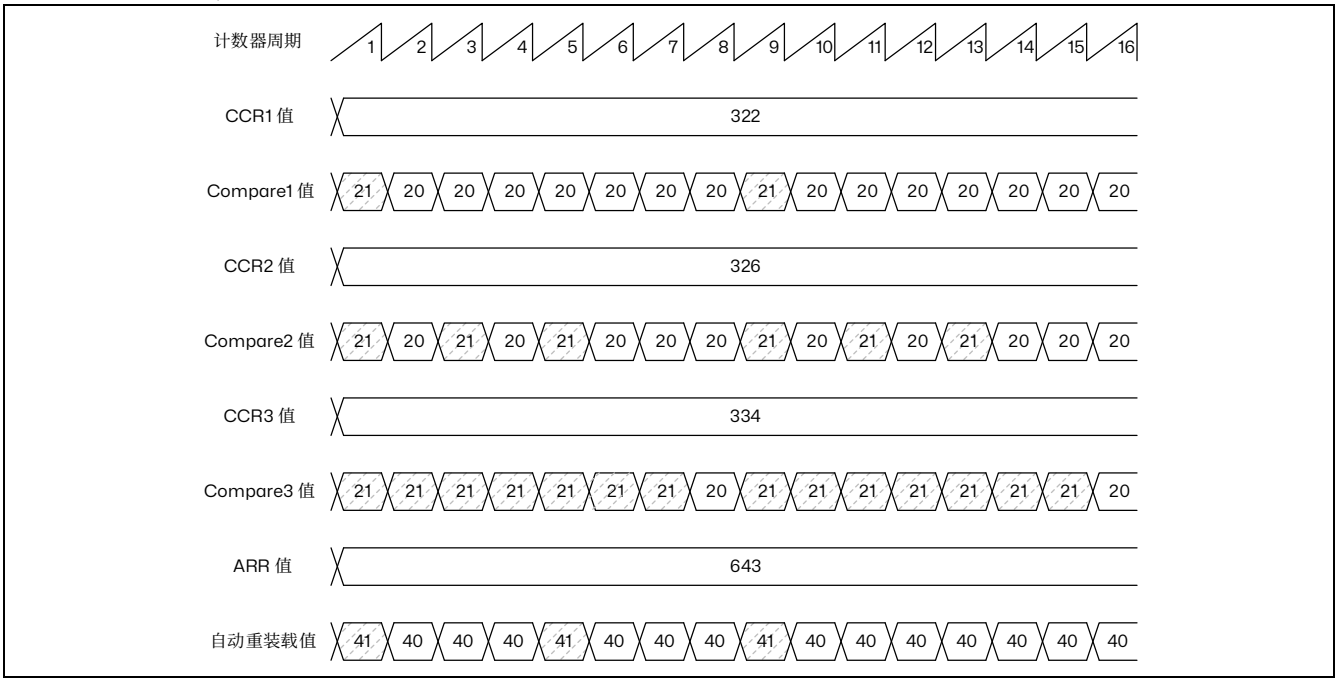


图 15.24 PWM 抖动模式

自动重新加载和比较值的增量按照下表中描述的特定模式分布。抖动序列是为了使增量分布尽可能均匀并最大限度地减少整体波动。

表 15.10 CCR 和 ARR 寄存器的变化抖动模式

LSB 值	PWM 周期															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0001	+1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0010	+1	-	-	-	-	-	-	-	+1	-	-	-	-	-	-	-
0011	+1	-	-	-	+1	-	-	-	+1	-	-	-	-	-	-	-
0100	+1	-	-	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0101	+1	-	+1	-	+1	-	-	-	+1	-	-	-	+1	-	-	-
0110	+1	-	+1	-	+1	-	-	-	+1	-	+1	-	+1	-	-	-
0111	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	-	-
1000	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1001	+1	+1	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-	+1	-
1010	+1	+1	+1	-	+1	-	+1	-	+1	+1	+1	-	+1	-	+1	-
1011	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	-	+1	-
1100	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1101	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	-	+1	+1	+1	-
1110	+1	+1	+1	+1	+1	+1	+1	-	+1	+1	+1	+1	+1	+1	+1	-
1111	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	-

15.3.13 组合 PWM 模式

组合 PWM 模式允许生成两个边缘对齐或中央对齐的 PWM 信号，各自脉冲之间的可编程延迟和相位偏移。虽然频率由 TIM15_ARR 寄存器的值决定，但占空比和延迟由两个 TIM15_CCRx 寄存器确定。结果信号 tim_ocxrefc 由两个参考 PWM 的 OR 或 AND 逻辑组合而成：

— tim_oc1refc（或 tim_oc2refc）由 TIM15_CCR1 和 TIM15_CCR2 控制

通过在 TIM15_CCMRx 寄存器的 OCxM 位中写入“1100”（组合 PWM 模式 1）或“1101”（组合 PWM 模式 2），可以在两个通道上独立选组合 PWM 模式（每个 CCR 寄存器对一个 tim_ocx 输出）。

当某个通道用作组合 PWM 通道时，其互补通道必须配置为相反的 PWM 模式（例如，一个通道为组合 PWM 模式 1，另一个通道为组合 PWM 模式 2）。

*注意：为了兼容性原因，*OCxM[3:0]*位字段被分为两个部分，最高有效位与最低*3 *位不连续。*

下图表示使用组合 PWM 模式生成信号的示例，该示例的配置如下：

— 通道 1 配置为组合 PWM 模式 2。

— 通道 2 配置为 PWM 模式 1。

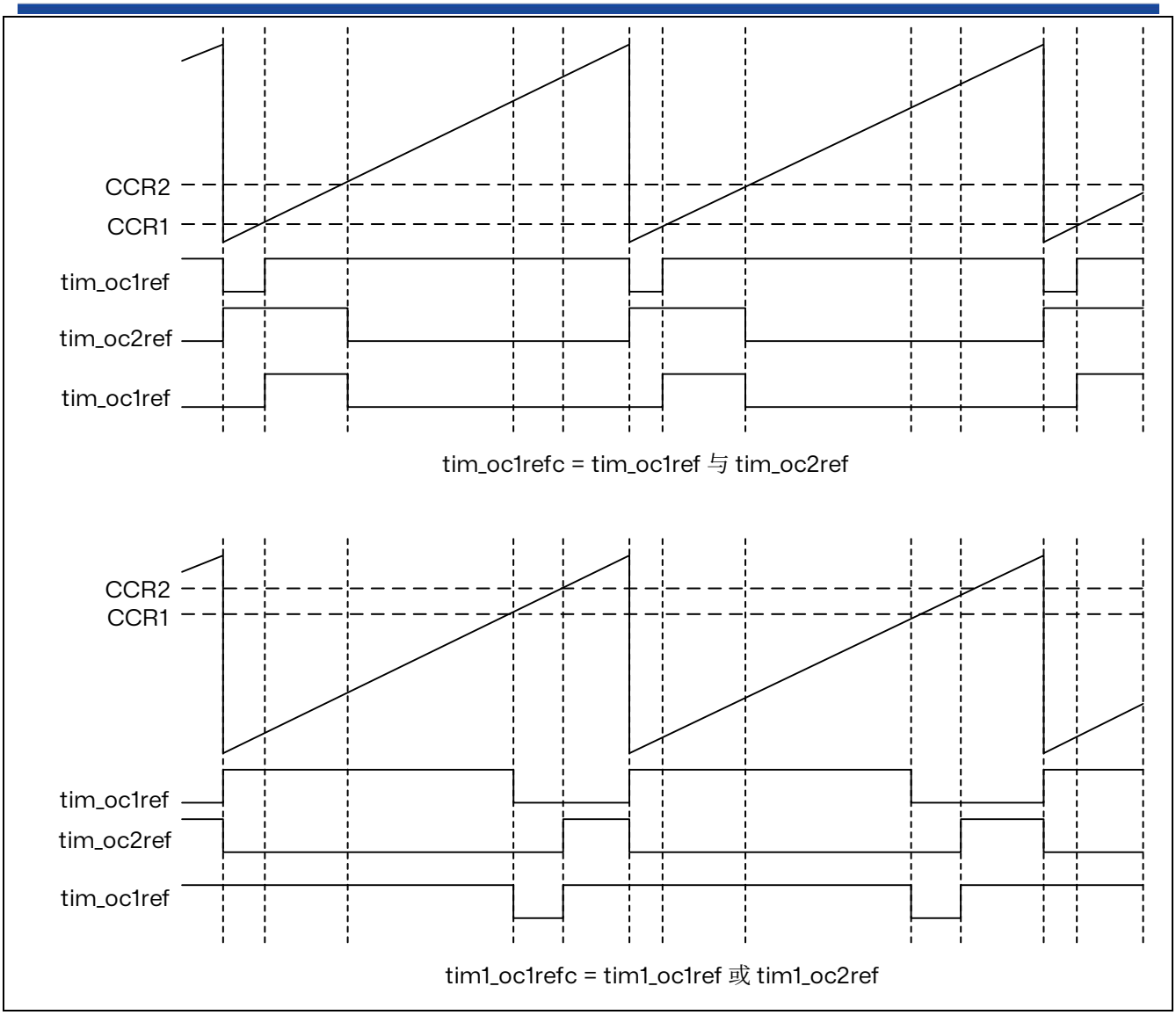


图 15.25 通道 1 和通道 2 的组合 PWM 模式

15.3.14 互补输出和死区插入

通用定时器（TIM15）可以输出一路互补信号，并管理输出的关断与接通瞬间。

这段时间通常称为死区，用户必须根据与输出相连接的器件及其特性（电平转换器的固有延迟、开关器件产生的延迟...）来调整死区时间

每路输出可以独立选择输出极性（主输出 tim_ocx 或互补输出 tim_ocxn）。可通过对 TIM15_CCER 寄存器中的 CCxP 和 CCxNP 位执行写操作来完成极性选择。

互补信号 tim_ocx 和 tim_ocxn 通过以下多个控制位的组合进行激活：TIM15_CCER 寄存器中的 CCxE 和 CCxNE 位以及 TIM15_BDTR 和 TIM15_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位。更多详细信息，请参考表格：带刹车功能的互补输出通道 tim_oc1 和 tim_oc1n 的控制位。应当注意，切换至 idle（MOE 下降到 0）的时刻，死区仍然有效。

CCxE 和 CCxNE 位同时置 1 并且 MOE 位置 1（如果存在刹车）时，将使能死区插入。TIM15_BDTR 寄存器中的 DTG[7:0]位用于控制所有通道的死区生成。将基于参考波形 tim_ocxref 生成 2 个输出 tim_ocx 和 tim_ocxn。如果 tim_ocx 和 tim_ocxn 为高电平有效：

- 输出信号 OCx 与参考信号相同，只是其上升沿相对参考上升沿存在延迟。
- 输出信号 OCxN 与参考信号相反，并且其上升沿相对参考下降沿存在延迟。

如果延迟时间大于有效输出（OCx 或 OCxN）的宽度，则不会产生相应的脉冲。

下图所示为死区发生器的输出信号与参考信号 OCxREF 之间的关系。（在这些示例中，假定 CCxP=0、CCxNP=0、MOE=1、CCxE=1 并且 CCxNE=1）

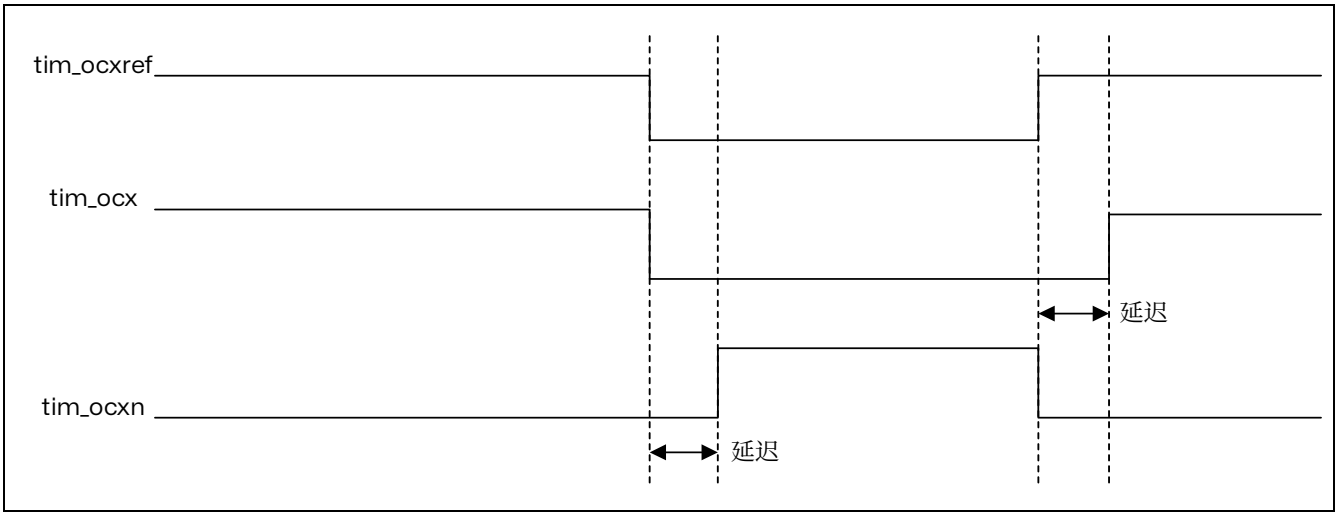


图 15.26 带死区插入的互补输出

TIM15_DTR2 中的 DTAE 位允许对参考信号上升沿和下降沿的死区值进行区分，如下图所示。

在非对称模式（DTAE=1）下，上升沿参考死区由 TIM15_BDTR 寄存器中的 DTG[7:0]位字段定义，而下降沿参考死区由 TIM15_DTR2 寄存器中的 DTGF[7:0]位字段定义。在使能计数器之前，必须写入 DTAE 位，并且在 CEN=1 时不得修改。

使用预加载机制可以在 PWM 操作期间动态更新死区值。当 TIM15_DTR2 寄存器中的 DTPE 位被设置时，死区位字段 DTG[7:0]和 DTGF[7:0]将被预加载。预加载值将在下一个更新事件上加载到活动寄存器中。

注意：如果 DTPE *位在计数器使能时被使能。自上次更新以来写入的新值将被丢弃，并使用先前的值。*

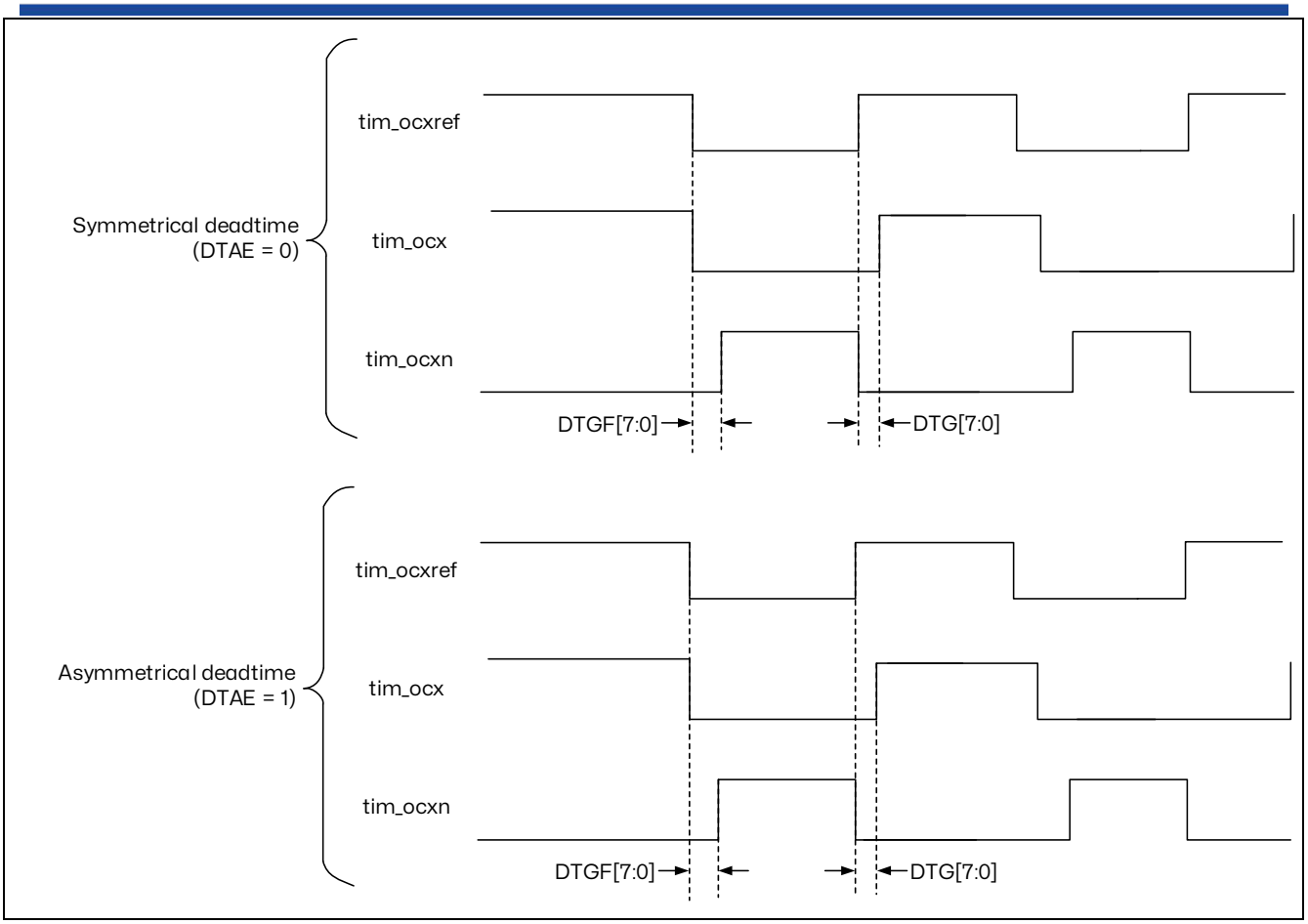


图 15.27 非对称死区时间

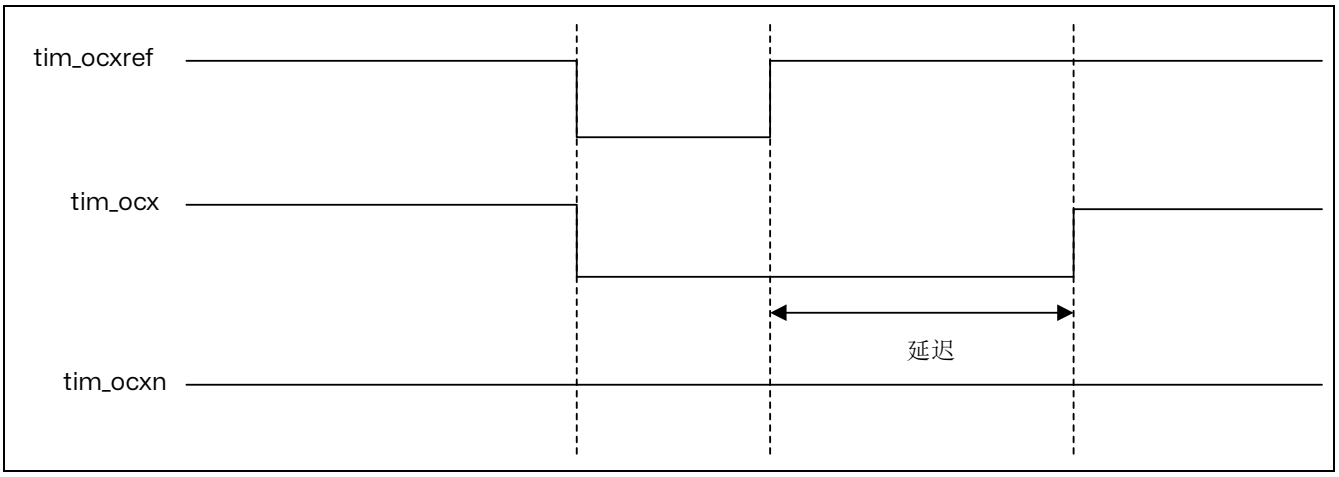


图 15.28 死区波形延迟大于负脉冲

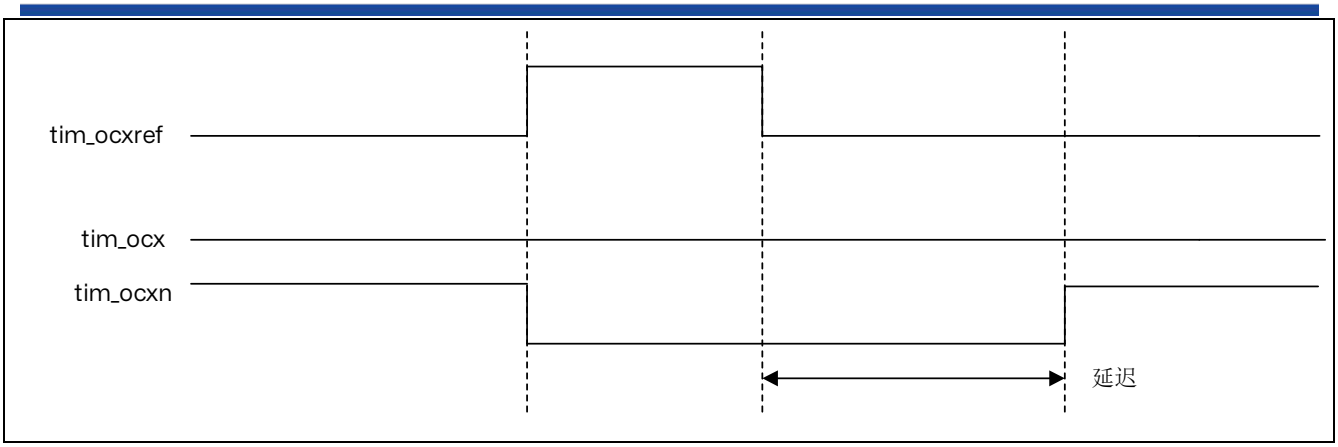


图 15.29 死区波形延迟大于正脉冲

死区延迟对于所有通道均相同，可通过 TIM15_BDTR 寄存器中的 DTG 位进行编程。有关延迟时间计算的信息，请参见后续小节：TIM15 刹车和死区寄存器(TIM15_BDTR)。

将 OCxREF 重定向到 OCx 或 OCxN

在输出模式（强制输出模式、输出比较模式或 PWM 模式）下，通过配置 TIM15_CCER 寄存器中的 CCxE 和 CCxNE 位，可将 tim_ocxref 重定向到 tim_ocx 输出或 tim_ocxn 输出。通过此功能，可以在一个输出上发送特定波形（如 PWM 或静态有效电平），而同时使互补输出保持其无效电平。或者，使两个输出同时保持无效电平，或者两个输出同时处于有效电平，两者互补并且带死区。

注意：如果仅使能 tim_ocxn *（*CCxE=0, CCxNE=1*），两者不互补，一旦* tim_ocxref *为高电平，*tim_ocxn *即变为有效。例如，如果* CCxNP=0，*则*tim_ocxn= tim_ocxref。*另一方面，如果同时使能*tim_ocxn *和* tim_ocxref *（*CCxE=CCxNE=1*），*tim_ocx *在*tim_ocxref *为高电平时变为有效，而*tim_ocxn *则与之互补，在*tim_ocxref *为低电平时变为有效。*

15.3.15 使用刹车功能

刹车功能的目的是保护由定时器产生的 PWM 信号驱动的功率开关。刹车输入通常连接到功率级和三相逆变器的故障输出。当激活时，刹车电路会关闭 PWM 输出并强制它们进入预定义的安全状态。

刹车通道可以收集系统级故障（时钟故障、ECC/奇偶校验错误，……）和应用故障（来自输入引脚和内置比较器），并可以在死区时间后将输出强制到一个预定义的水平（无论是有效还是无效）。

刹车期间输出使能信号和输出电平取决于几个控制位：

- TIM15_BDTR 寄存器中的 MOE 位允许通过软件使能/禁用输出，并在发生刹车事件时重置。
- TIM15_BDTR 寄存器中的 OSSI 位定义了定时器是否在非活动状态下控制输出或将其控制权释放给 GPIO 控制器（通常使其处于高阻模式）。
- TIM15_CR2 寄存器中的 OISx 和 OISxN 位用于设置输出关闭电平，可以是有效电平或无效电平。tim_oc1 和 tim_oc1n 输出不能同时设为有效电平。更多详细信息请参考后续表格：具有刹车功能的互补 tim_oc1 和 tim_oc1n 通道的输出控制位。

退出复位时，刹车电路被禁用，MOE 位为低电平。通过设置 TIM15_BDTR 寄存器中的 BKE 位，可以使能刹车功能。通过配置同一寄存器中的 BKP 位，可以选择刹车输入极性。BKE 和 BKP 可以同时修改。在写入 BKE 和 BKP 位后，需要等待 1 个 APB 时钟周期才能使写入生效。因此，在写入操作后，需要等待 1 个 APB 时钟周期才能正确地读取该位。

因为 MOE 下降沿可以是异步的，所以在实际的信号（作用于输出）和同步控制位（在 TIM15_BDTR 寄存器中访问）之间插入了一个重新同步电路。这导致异步信号和同步信号之间存在一些延迟。特别是当

MOE 被设置为 1 而之前它是低电平时，在正确读取之前必须插入一个延迟（伪指令）。这是因为写入作用于异步信号，而读取反映的是同步信号。

刹车是由 tim_brk 输入产生的，它具有：

- 可编程极性（TIM15_BDTR 寄存器中的 BKP 位）
- 可编程使能位（TIM15_BDTR 寄存器中的 BKE 位）
- 可编程滤波器（TIM15_BDTR 寄存器中的 BKF[3:0]位），以避免产生虚假事件。

可以使用 TIM15_AF1 寄存器，从多个源生成刹车，这些源可以单独使能，并具有可编程的边沿敏感度。

刹车（tim_brk）通道的源为：

- 连接到 TIM15_BKIN 引脚的外部源（根据 GPIO 备用功能选择寄存器中的选择而定），具有极性选择和可选的数字滤波
- 内部源：
  - 来自 tim_brk_cmpx 输入（请参考章节：TIM15 引脚和内部信号）
  - 来自 tim_sys_brk 输入的系统中断请求（请参考章节：TIM15 引脚和内部信号）

也可以通过软件使用 TIM15_EGR 寄存器中的 BG 位生成刹车事件。

所有源在进入定时器的 tim_brk 输入之前进行 OR 操作，如下图所示。

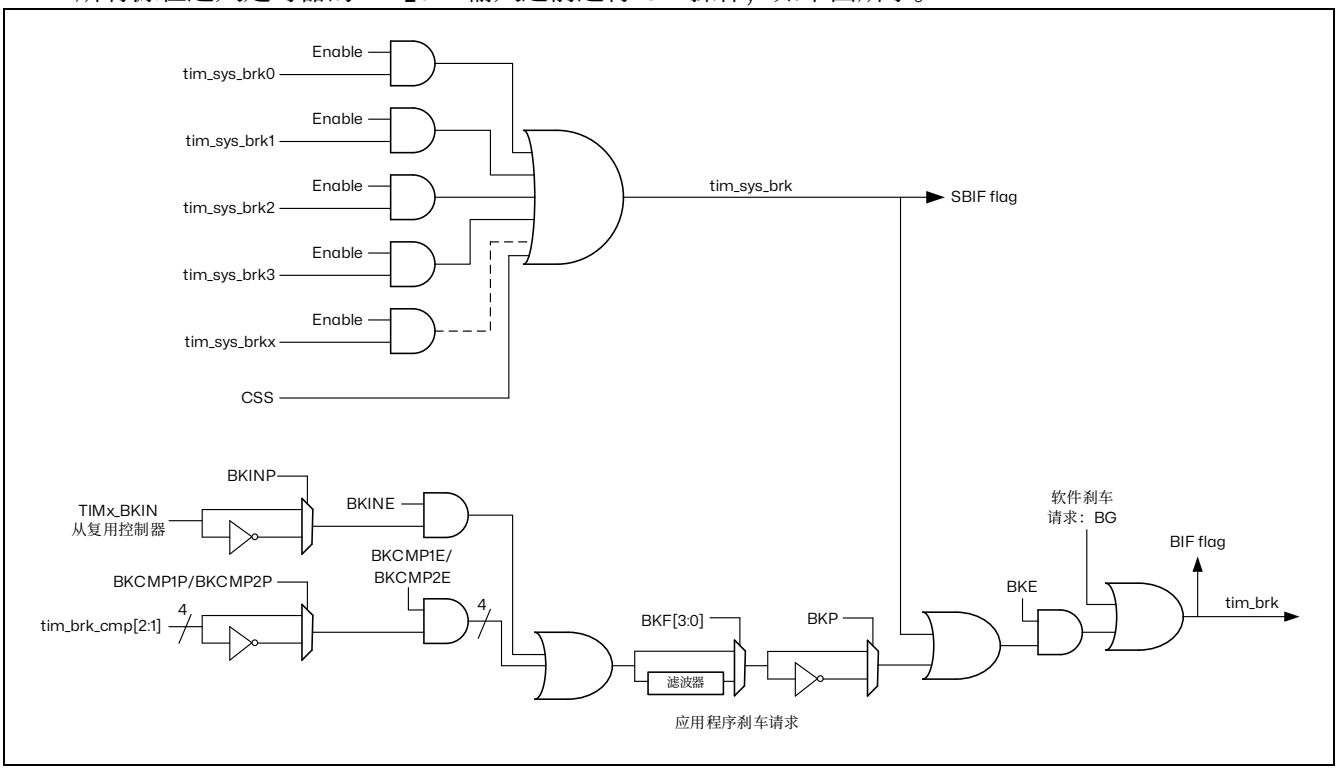


图 15.30 刹车电路概述

*注意：只有当可编程滤波器禁用时，才保证异步（无时钟）操作。如果使能，必须使用故障安全时钟模式（例如使用内部*PLL *和/或*CSS*）来保证刹车事件得到处理。*

当一个刹车发生时（在刹车输入上选择电平）：

- MOE 位被异步清除，使输出处于无效状态、空闲状态，甚至释放控制权给 GPIO（由 OSSI 位选择）。即使 MCU 振荡器关闭，此功能也可工作。
- 每当 MOE=0 时，每个输出通道都以 TIM15_CR2 寄存器中的 OISx 位中编程的电平驱动。如果 OSSI=0，定时器将释放输出控制（由 GPIO 接管），否则使能输出保持高电平。

- 当使用互补输出时：
  — 输出首先被置于复位状态（取决于极性），这是异步完成的，因此即使没有向定时器提供时钟也可以工作。
  — 如果定时器的时钟仍然存在，则重新激活死区时间发生器，以便在死区时间后驱动输出到编程在 OISx 和 OISxN 位中的电平。即使在这种情况下，tim_ocx 和 tim_ocxn 也不能同时驱动到有效电平。请注意，由于在 MOE 上进行了重新同步，死区时间持续时间比通常情况稍长（大约 2 个 tim_ker_ck 时钟周期）。
  — 如果 OSSI=0，则定时器释放使能输出（由 GPIO 接管，强制进入高阻态），否则使能输出在 CCxE 或 CCxNE 位中的任意一位变为高电平时立即变为高电平。
- 刹车状态标志（TIM15_SR 寄存器中的 BIF 位）被设置。如果 TIM15_DIER 寄存器中的 BIE 位被设置，则可以生成刹车。
- 如果 TIM15_BDTR 寄存器中的 AOE 位被设置，则 MOE 位将在下一个更新事件 UEV 处自动重新设置。这可以用于执行调节等操作。否则，MOE 保持低电平，直到再次写入 1。在这种情况下，它可以用于安全，刹车输入可以连接到来自功率驱动器、温度传感器或任何安全组件的警报。

注意：如果 MOE *被* CPU *复位，而* AOE *位已设置，则输出处于空闲状态，并强制处于非活动电平或高阻态，具体取决于* OSSI *值。如果* MOE *和* AOE *位都被* CPU *复位，则输出处于禁用状态，并驱动在* TIM15_CR2 *寄存器中* OISx *位编程的电平。*

注意：刹车输入在电平上有效。因此，当刹车输入处于活动状态时（无论是自动还是通过软件设置）， MOE *都无法设置。同时，状态标志* BIF *不能被清除。*

除刹车输入和输出管理外，刹车电路内部还实施了写保护，用以保护应用的安全。通过该功能，用户可冻结多个参数配置（死区持续时间、tim_ocx/tim_ocxn 极性和禁止时的状态、OCxM 配置、刹车使能和极性）。可以通过 TIM15_BDTR 寄存器中的 LOCK 位，从 3 种保护级别中进行选择。请参考章节：TIM15 刹车和死区寄存器（TIM15_BDTR）。MCU 复位后只能对 LOCK 位执行一次写操作。

下图所示为输出对刹车响应行为的示例。

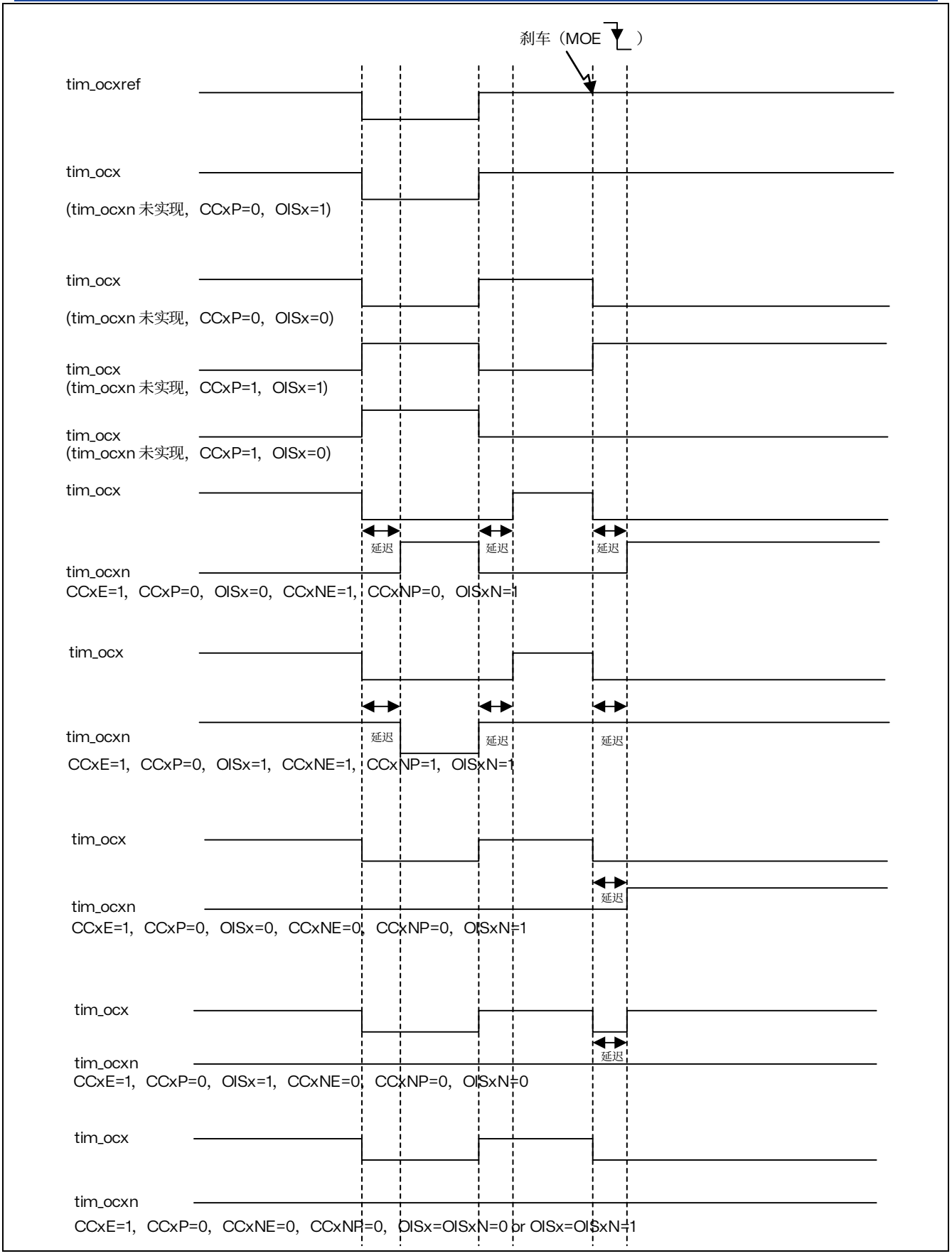


图 15.31 tim_brk 在发生刹车事件时的各种输出行为（OSSI=1）

15.3.16 双向刹车输入

TIM15 具有双向刹车 I/O，如下图所示。

这为以下方面提供支持：

- 板级全局刹车信号，用于向外部 MCU 或门驱动器发送故障信号，使用唯一的引脚作为输入和输出状态引脚
- 内部刹车源和多个外部开漏源 OR 连接在一起，以触发唯一的刹车事件，当多个内部和外部刹车源必须合并时

使用 TIM15BDTR 寄存器中的 BKBID 位配置 tim_brk 输入为双向模式。BKBID 编程位可以使用 TIM15BDTR 寄存器中的 LOCK 位锁定为只读模式（在 LOCK 级别 1 或更高）。

双向模式要求 I/O 配置为开漏模式，使用低电平有效极性（使用 BKINP、BKP 位）。来自系统（例如 CSS）、片上外设或刹车输入的任何刹车请求都会强制将刹车输入置于低电平，以指示故障事件。出于安全原因，如果极性位没有正确设置（高电平有效），则双向模式将被抑制。

刹车软件事件（通过设置 BG 位触发）也会强制将刹车 I/O 置为“0”，以指示外部组件定时器已进入刹车状态。然而，只有使能了刹车（BKE = 1）时，这才有效。当使用 BKE = 0 生成软件刹车事件时，输出将处于安全状态，并且会设置刹车标志，但 TIM15_BKIN I/O 没有影响。

安全解发机制可防止系统被彻底锁定（刹车输入低电平触发刹车，强制该输入为低电平）。

当 BKDSRM 位设置为 1 时，这将释放刹车输出以清除故障信号并允许重新给系统装载。

在任何时候都不能禁用刹车保护电路：

- 刹车输入路径始终处于活动状态：即使设置了 BKDSRM 位并且开放漏极控制被释放，刹车事件仍然处于活动状态。这可以防止 PWM 输出在刹车条件存在时重新启动。
- 只要输出使能（MOE 位被设置），BKDSRM 位就不能解除刹车保护（参见下表）

表 15.11 刹车保护解除条件

MOE	BKBID	BKDSRM	刹车保护状态
0	0	X	装载
0	1	0	装载
0	1	1	解除
1	X	X	装载

装载和重装载刹车电路

默认情况下（外设复位配置），刹车电路（输入或双向模式）处于装载状态。

在发生刹车事件后，必须按照以下步骤重新装载保护：

- BKDSRM 位必须设置为释放输出控制
- 软件必须等待系统刹车条件消失，并清除 SBIF 状态标志（或在重新装载之前系统地清除它）
- 软件必须轮询 BKDSRM 位，直到硬件清除该位（当应用程序刹车条件消失时）

从这一点开始，刹车电路处于装载和有效状态，并且可以设置 MOE 位以重新使能 PWM 输出。

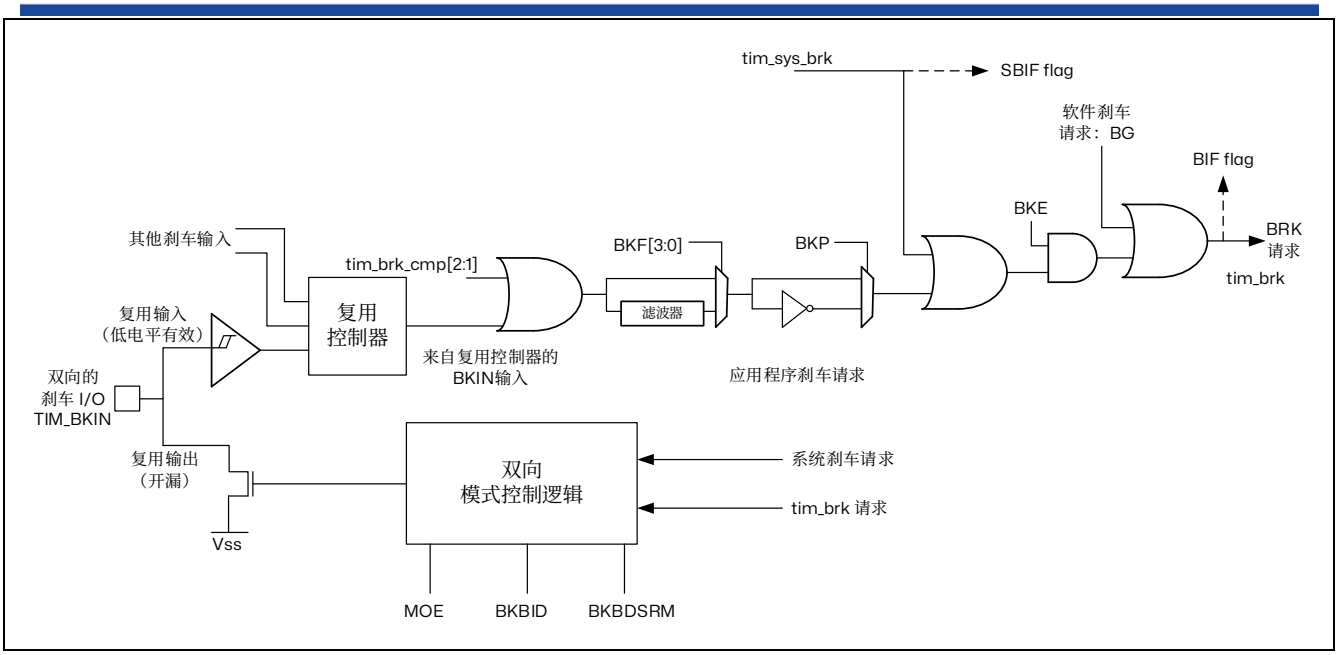


图 15.32 输出重定向

15.3.17 在外部事件时清除 tim_ocxref 信号

对于给定通道，在 tim_ocref_clr_int 输入施加高电平（相应 TIM15_CCMRx 寄存器中的 OCxCE 使能位置“1”），可使 tim_ocxref 信号变为低电平。tim_ocxref 信号将保持低电平，直到发生下一更新事件(UEV)。此功能仅能用于输出比较模式和 PWM 模式，而不适用于强制输出模式。

当配置组合 PWM 模式，对于 TIM15_CH1，在 tim_ocref_clr_int 输入施加高电平（相应 TIM15_CCMR1 寄存器中的 OC1CE 使能位置“1”），可使组合结果 tim_oc1refc 信号变为低电平。tim_oc1refc 信号将保持低电平，直到发生下一更新事件（UEV）（特别注意，当 TIM15_RCR>0 时，只有重复计数器达到零才会生成更新事件，波形才会恢复）。

tim_ocref_clr 输入可以在多个输入中进行选择，如下图所示。

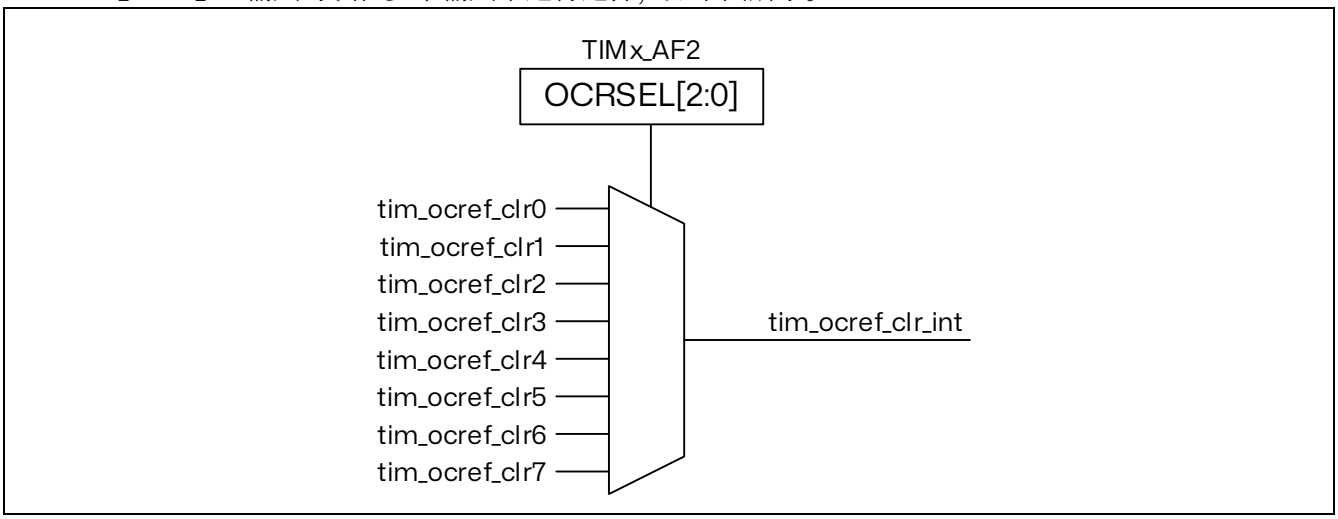


图 15.33 tim_ocref_clr 输入选择复用器

15.3.18 生成 6 步 PWM

当通道使用互补输出时，OCxM、CCxE 和 CCxNE 位上提供预装载位。发生 COM 换向事件时，这些预装载位将传输到影子位。因此，用户可以预先编程下一步骤的配置，并同时更改所有通道的配置。COM 可由软件通过将 TIM15_EGR 寄存器中的 COM 位置 1 而生成，也可以由硬件在 tim_trgi 上升沿生成。

发生 COM 事件时，某个标志位（TIM15_SR 寄存器中的 COMIF 位）将会置 1。这时，如果 TIM15_DIER 寄存器中的 COMIE 位置 1，将产生中断。

下图以 3 种不同的编程配置为例，显示了发生 COM 事件时 tim_ocx 和 tim_ocxn 输出的行为。

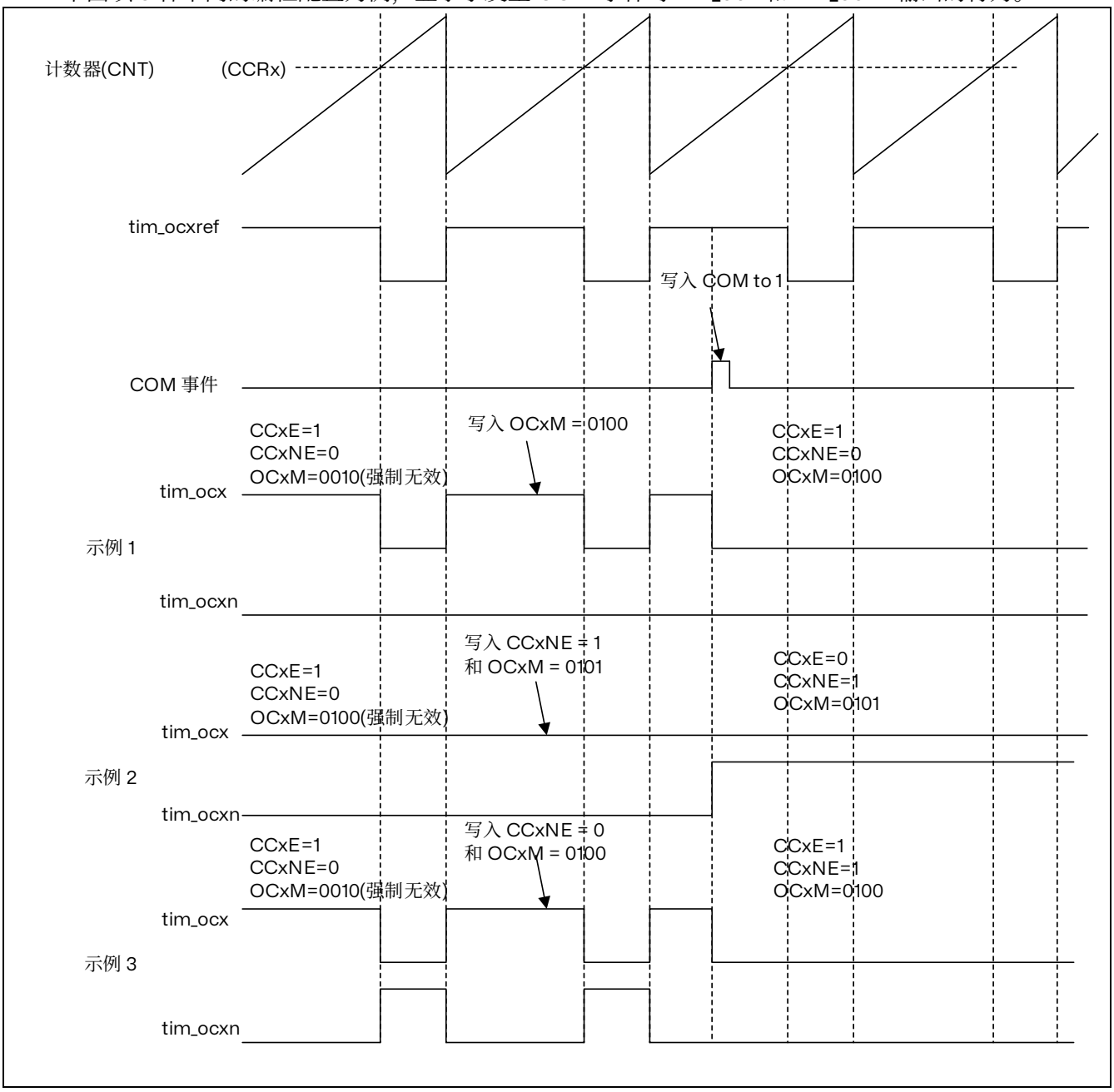


图 15.34 COM 事件生成 6 步 PWM 的示例（OSSR=1）

15.3.19 单脉冲模式

单脉冲模式（OPM）是上述模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启

动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。将 TIM15_CR1 寄存器中的 OPM 位置 1，即可选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

只有当比较值与计数器初始值不同时，才能正确产生一个脉冲。启动前（定时器等待触发时），必须进行如下配置：

- CNT < CCRx ≤ ARR（特别注意，0<CCRx）

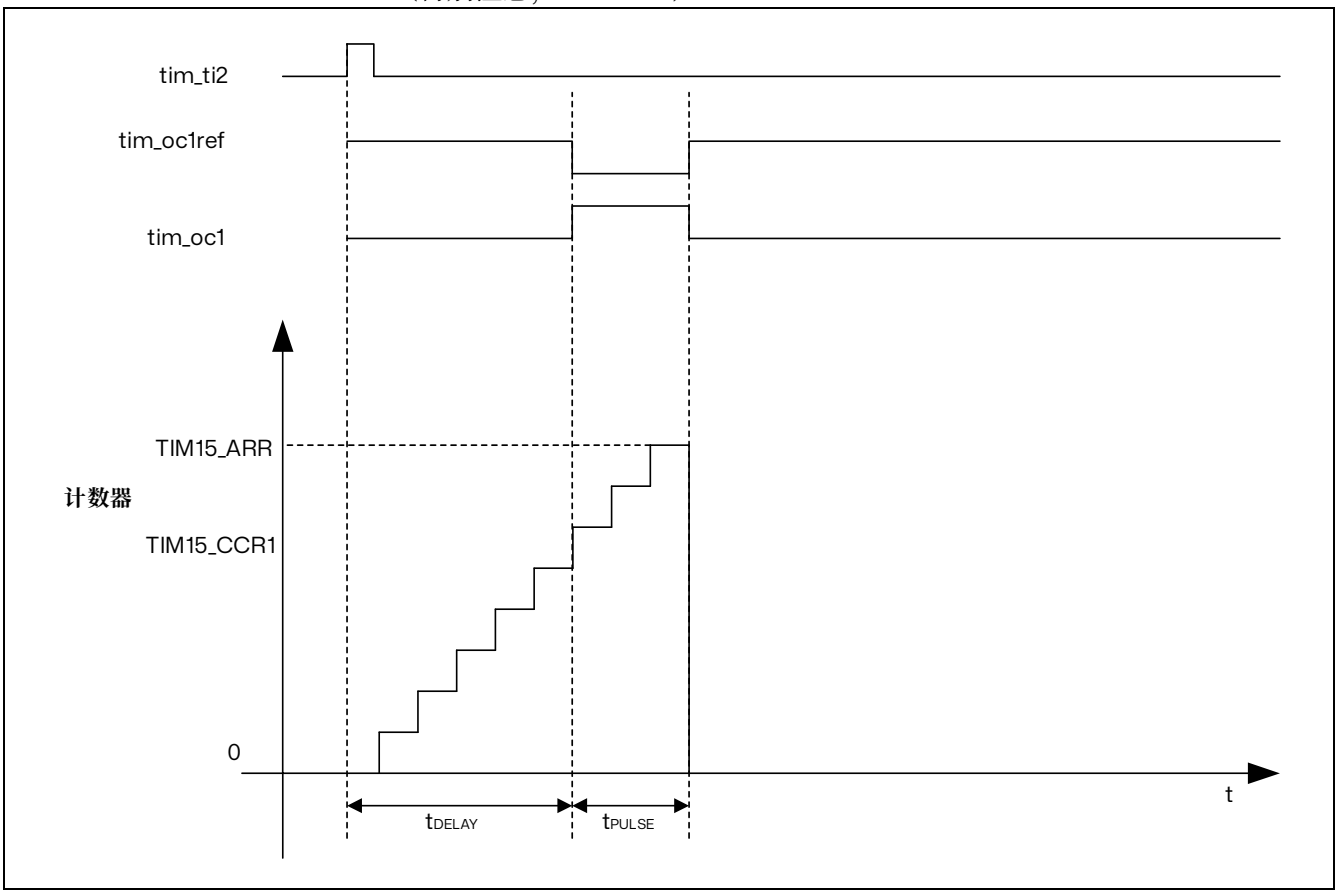


图 15.35 单脉冲模式的例子

例如，用户希望达到这样的效果：在 tim_ti2 输入引脚检测到正沿时，经过 t_{DELAY} 的延迟，在 tim_oc1 上产生一个长度为 t_{PULSE} 的正脉冲。

使用 tim_ti2fp2 作为触发 1:

1. 通过在 TIM15_TISEL 寄存器中使用 TI2SEL[3:0]位选择合适的 tim_ti2_in[15:0]源（内部或外部）。
2. 在 TIM15_CCMR1 寄存器中写入 CC2S=01，将 tim_ti2fp2 映射到 tim_ti2。
3. 在 TIM15_CCER 寄存器中写入 CC2P=0 和 CC2NP=“0”，使 tim_ti2fp2 能够检测上升沿。
4. 在 TIM15_SMCR 寄存器中写入 TS=00110，将 tim_ti2fp2 配置为从模式控制器的触发 (tim_trgi)。
5. 在 TIM15_SMCR 寄存器中写入 SMS=“110”（触发模式），使用 tim_ti2fp2 启动计数器。

OPM 波形通过对比较寄存器执行写操作来定义（考虑时钟频率和计数器预分频器）。

- t_{DELAY} 由写入 TIM15_CCR1 寄存器的值定义。
- t_{PULSE} 由自动重载值与比较值（TIM15_ARR - TIM15_CCR1）之差来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从“0”变为“1”，在计数器达到自动重载值时由“1”变为“0”。为此，应在 TIM15_CCMR1 寄存器中写入 OC1M=111，以使能 PWM 模式 2。如果

需要，可选择在 TIM15_CCMR1 寄存器的 OC1PE 和 TIM15_CR1 寄存器的 ARPE 中写入“1”，以使能预装载寄存器。这种情况下，必须在 TIM15_CCR1 寄存器中写入比较值并在 TIM15_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 tim_ti2 上的外部触发事件。本例中，CC1P 的值为“0”。

在本例中，TIM15_CR1 寄存器中的 DIR 和 CMS 位应为低。

由于仅需要 1 个脉冲（单脉冲模式），因此应向 TIM15_CR1 寄存器的 OPM 位写入“1”，以便在发生下一更新事件（计数器从自动重载值返回到 0）时使计数器停止计数。TIM15_CR1 寄存器中的 OPM 位置“0”时，即选择重复模式。

特例：OCx 快速使能：

在单脉冲模式下，TIx 输入的边沿检测会将 CEN 位置 1，表示使能计数器。然后，在计数器值与比较值之间发生比较时，将切换输出。但是，完成这些操作需要多个时钟周期，这会限制可能的最小延迟（t_{DELAY} 最小值）。

如果要输出延迟时间最短的波形，可以将 TIM15_CCMRx 寄存器中的 OCxFE 位置 1。这样会强制 tim_ocxref（和 tim_ocx）对激励信号做出响应，而不再考虑比较的结果。其新电平与发生比较匹配时相同。仅当通道配置为 PWM1 或 PWM2 模式时，OCxFE 才会起作用

15.3.20 可重触发单脉冲模式

此模式允许计数器在接收到刺激后开始计数，并生成一个具有可编程长度的脉冲，但与上一节中描述的非重触发单脉冲模式存在以下差异：

— 只要触发器发生，脉冲就会立即开始（没有可编程的延迟）

— 如果在先前触发的脉冲完成之前发生新的触发，则脉冲会被延长

定时器必须处于从模式，TIM15_SMCR 寄存器的 SMS[3:0]位设置为‘1000’（组合重置+触发模式），OCxM[3:0]位设置为‘1000’或‘1001’，用于可重触发 OPM 模式 1 或 2。

如果定时器配置为递增计数模式，则相应的 CCRx 必须设置为 0（ARR 寄存器设置脉冲长度）。如果定时器配置为递减计数模式，CCRx 必须大于或等于 ARR。

*注意：*OCxM[[3:0]*和*SMS[3:0]*位字段由于兼容性的原因被分成两个部分，最高有效位与最低*3 *位不是连续的。*OCxM *这种模式不能与中央对齐*PWM *模式一起使用。*TIM15_CR1 *寄存器中的*CMS[1:0]*必须设置为*00。

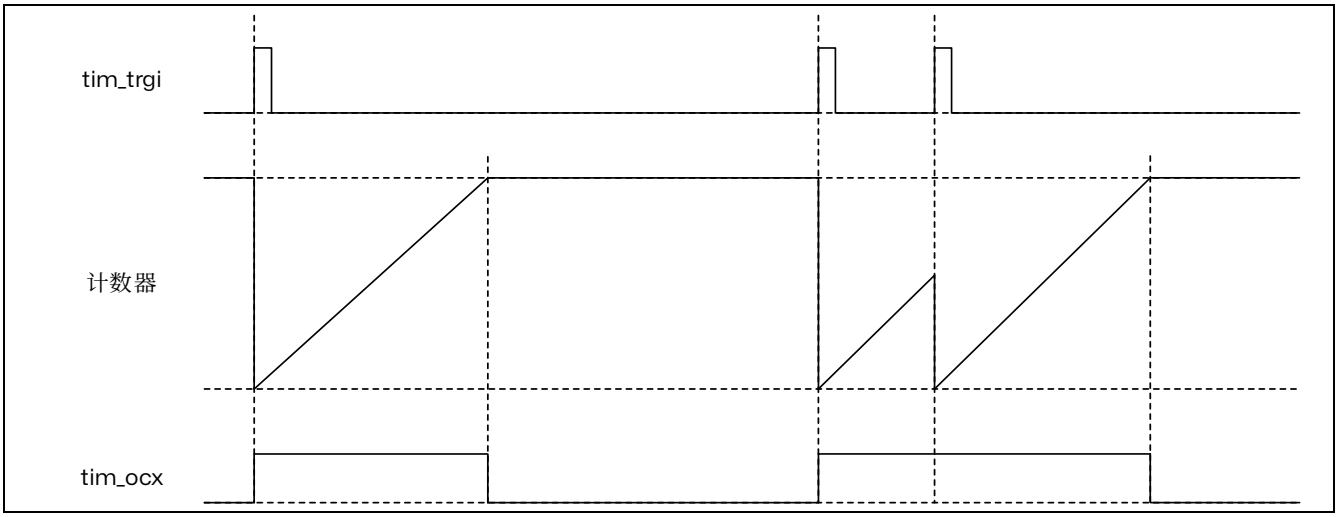


图 15.36 可重触发单脉冲模式

15.3.21 UIF 位重映射

TIM15_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志 UIF 连续复制到定时器计数器寄存器的第 31 位（TIM15CNT[31]）。这允许以原子方式读取计数器的值和一个潜在的由 UIFCPY 标志指示的回滚条件。在特定情况下，它可以避免由背景任务（读取计数器）和中断（更新中断）之间的处理引起的竞争条件，从而简化计算。

UIF 和 UIFCPY 标志之间没有延迟。

15.3.22 定时器输入异或功能

TIM15_CR2 寄存器中的 TI1S 位允许将通道 1 的输入滤波器连接到异或门的输出，将两个输入引脚 tim_ti1 和 tim_ti2 组合起来。

异或输出可以与所有定时器输入功能一起使用，例如触发器或输入捕获。它对于测量两个输入信号之间的边沿间隔非常有用，如下图所示。

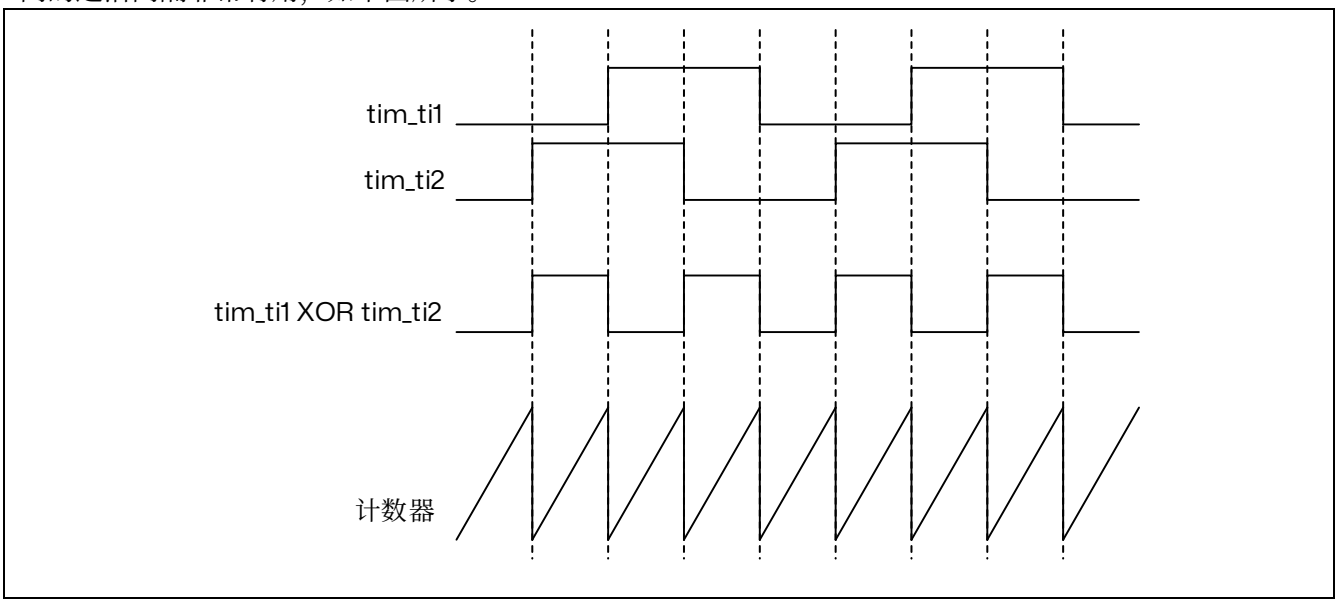


图 15.37 测量两个信号边沿之间的间隔时间

15.3.23 定时器和外部触发的同步

TIM 定时器在互相连接用于定时器的同步或链接。

TIM15 定时器以下列模式与外部触发实现同步：复位模式、门控模式、触发模式、复位+触发和门控+复位模式。

从模式：复位模式

当触发输入信号发生变化时，计数器及其预分频器可重新初始化。此外，如果 TIM15_CR1 寄存器中的 URS 位处于低电平，则会生成更新事件 UEV。然后，所有预装载寄存器（TIM15_ARR 和 TIM15_CCRx）都将更新。

在以下示例中，tim_ti1 输入上出现上升沿时，递增计数器清零：

1. 将通道 1 配置为检测 tim_ti1 的上升沿。配置输入滤波时间（本例中不需要任何滤波，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIM15_CCMR1 寄存器中的 CC1S = 01。在 TIM15_CCER 寄存器中写入 CC1P=0

和 CC1NP=0，验证极性（仅检测上升沿）。

2. 在 TIM15_SMCR 寄存器中写入 SMS=100，将定时器配置为复位模式。在 TIM15_SMCR 寄存器中写入 TS=101，选择 tim_ti1 作为输入源。
3. 在 TIM15_CR1 寄存器中写入 CEN=1，启动计数器。

计数器使用内部时钟计数，然后正常运转，直到出现 tim_ti1 上升沿。当 tim_ti1 出现上升沿时，计数器清零，然后重新从 0 开始计数。同时，触发标志（TIM15_SR 寄存器中的 TIF 位）置 1，使能中断后，还可发送中断请求（取决于 TIM15_DIER 寄存器中的 TIE）。

下图显示了自动重载寄存器 TIM15_ARR=0x36 时的相关行为。tim_ti1 的上升沿与实际计数器复位之间的延迟是由于 tim_ti1 输入的重新同步电路引起的。

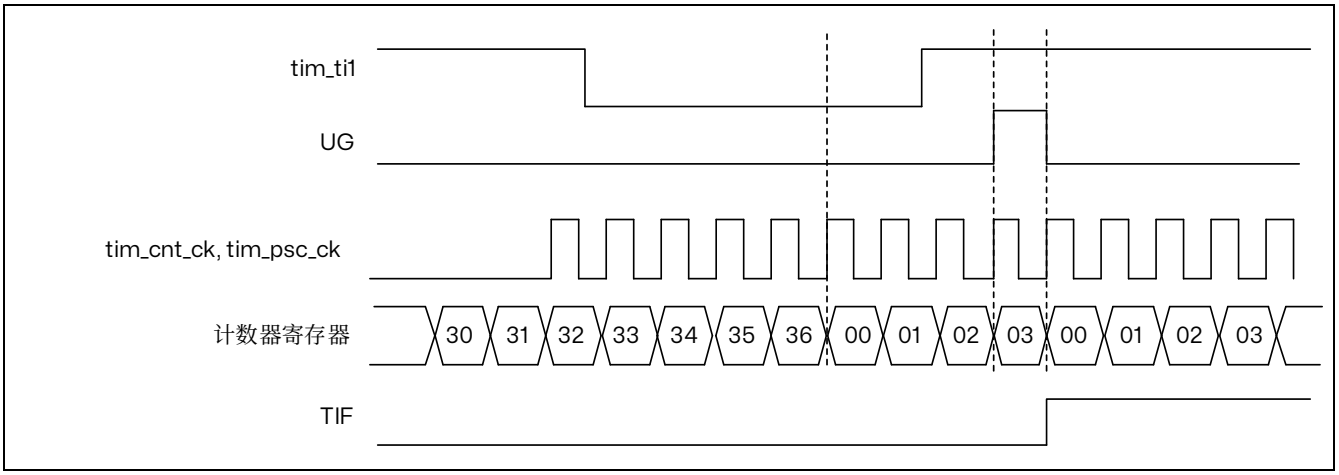


图 15.38 复位模式下的控制电路

从模式：门控模式

输入信号的电平可用来使能计数器。

在以下示例中，递增计数器仅在 tim_ti1 输入为低电平时计数：

1. 将通道 1 配置为检测 tim_ti1 上的低电平。配置输入滤波时间（本例中不需要任何滤波，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIM15_CCMR1 寄存器中的 CC1S=01。在 TIM15_CCER 寄存器中写入 CC1P=1 和 CC1NP=0，以确定极性（仅检测低电平）。
2. 在 TIM15_SMCR 寄存器中写入 SMS=101，将定时器配置为门控模式。在 TIM15_SMCR 寄存器中写入 TS=00101，选择 tim_ti1 作为输入源。
3. 在 TIM15_CR1 寄存器中写入 CEN=1，使能计数器（在门控模式下，如果 CEN=0，则无论触发输入电平如何，计数器都不启动）。

只要 tim_ti1 为低电平，计数器就开始根据内部时钟计数，直到 tim_ti1 变为高电平时停止计数。计数器启动或停止时，TIM15_SR 寄存器中的 TIF 标志都会置 1。

tim_ti1 的上升沿与实际计数器停止之间的延迟是由于 tim_ti1 输入的重新同步电路引起的。

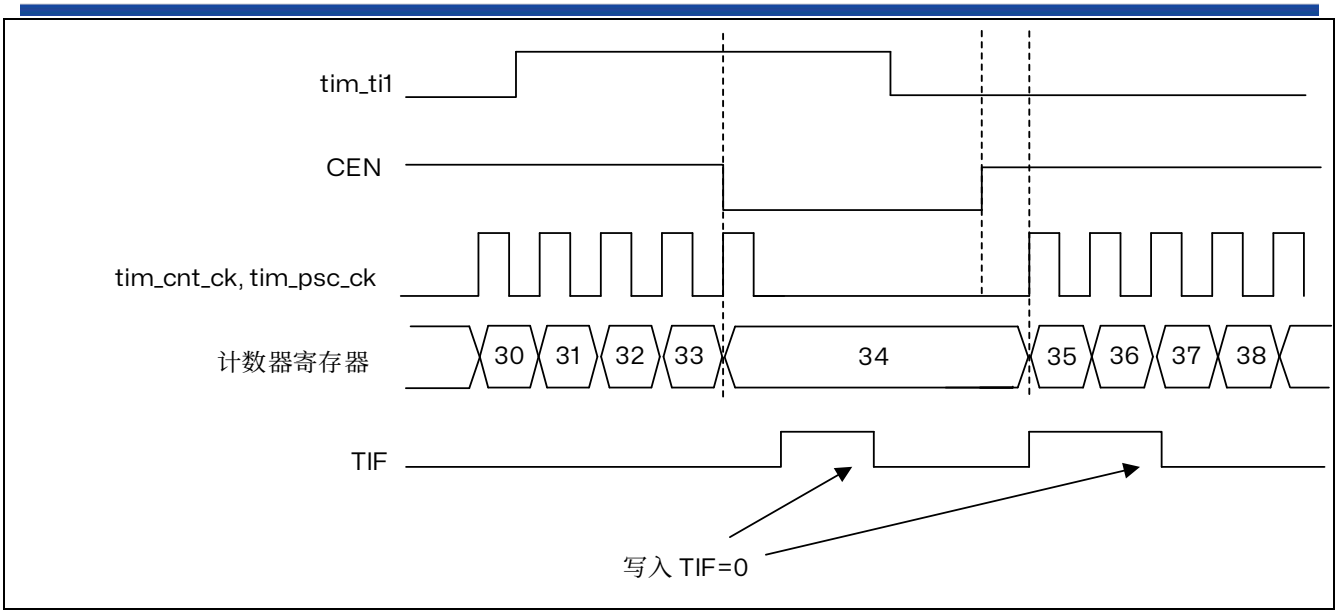


图 15.39 门控模式下的控制电路

从模式：触发模式

所选输入上发生某一事件时可以启动计数器。

在以下示例中，tim_ti2 输入上出现上升沿时，递增计数器启动：

1. 将通道 2 配置为检测 TI2 上的上升沿。配置输入滤波时间（本例中不需要任何滤波，因此保持 IC2F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC2S 位只选择输入捕获源，即 TIM15_CCMR1 寄存器中的 CC2S=01。在 TIM15_CCER 寄存器中写入 CC2P=1 和 CC2NP=0，以确定极性（仅检测低电平）。
2. 在 TIM15_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIM15_SMCR 寄存器中写入 TS=00110，选择 tim_ti2 作为输入源。

当 tim_ti2 出现上升沿时，计数器开始根据内部时钟计数，并且 TIF 标志置 1。

tim_ti2 的上升沿与实际计数器启动之间的延迟是由于 tim_ti2 输入的重新同步电路引起的。

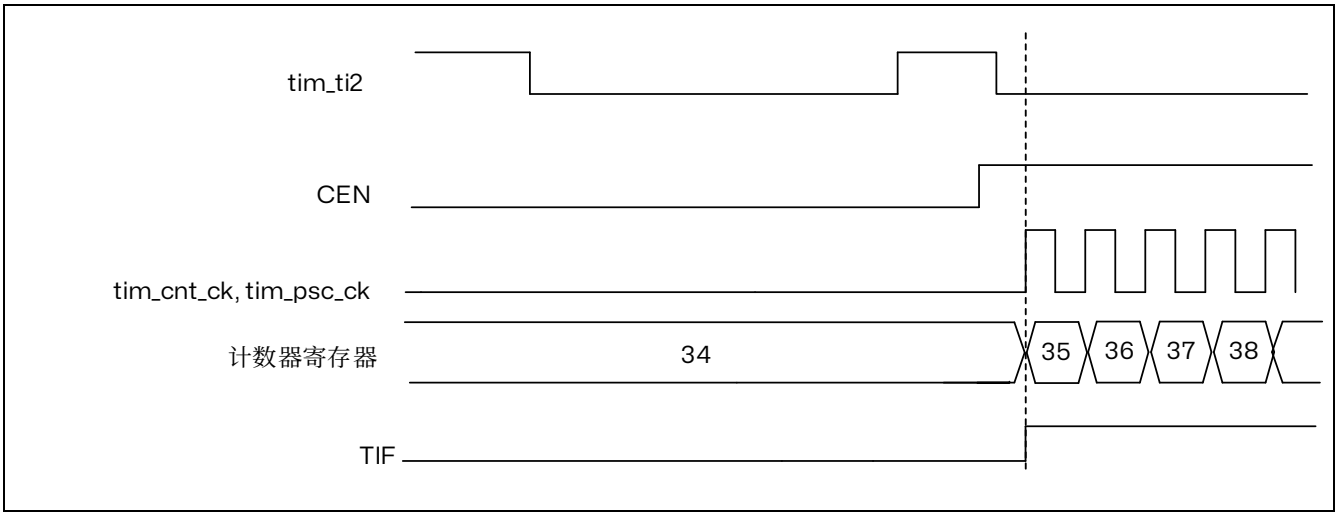


图 15.40 触发器模式下的控制电路

对实时更新编码器模式的从模式选择预装载

SMS[3:0]位可以预加载。这是通过在 TIM15_SMCR 寄存器中设置 SMSPE 使能位来实现的。从 SMS[3:0]预加载到有效值的触发器是计数器溢出时发生的更新事件（UEV）。

从模式 - 组合重置 + 触发模式

在这种情况下，所选触发输入（tim_trgi）的上升沿会重新初始化计数器，
此模式用于单脉冲模式。

从模式 - 组合重置 + 门控模式

当触发输入（tim_trgi）为高时，计数器使能。计数器停止并一旦触发器变为低，计数器就会重置。计数器的启动和停止都是受控的。

此模式允许检测超出范围的 PWM 信号（占空比超过最大期望值）。

15.3.24 定时器同步

TIM15 定时器从内部连接在一起，以实现定时器同步或级联。有关详细信息，请参见上一章节中的定时器同步。

注：接收 tim_trgo 信号的从外设（定时器、ADC...）的时钟必须在主定时器接收事件前使能，并且在主定时器接收触发时，时钟频率（预分频器）不得动态更改。

15.3.25 ADC 触发

定时器可以使用各种内部信号（例如重置、使能或比较事件）生成 ADC 触发事件。

注意：接收 tim_trgo 信号的从属外设（定时器、ADC 等）的时钟必须在从主定时器接收事件之前使能，并且时钟频率（预分频器）在从主定时器接收触发器时不能更改。

15.3.26 调试模式

当微控制器进入调试模式时（内核停止），TIM15 计数器会继续正常工作或者停止。

在调试模式下，每个定时器的行为可以通过 Debug 支持（DBG）模块的专用配置位进行编程。

出于安全考虑，当计数器停止时，输出被禁用（就像将 MOE 位复位一样）。输出可以强制处于非活动状态（OSSI 位=1），或者由 GPIO 控制器接管控制（OSSI 位=0）以强制其处于高阻态。

有关更多详细信息，请参阅调试部分。

15.4 TIM15 低功耗模式

表 15.12 低功耗模式对 TIM15 的影响

模式	描述
睡眠	无影响，外设是运行的。中断会导致设备退出睡眠模式。
停机	定时器操作被停止并保留其寄存器内容。无法生成中断。
待机	定时器被断电并且退出待机模式后必须重新初始化。

15.5 TIM15 中断

TIM15 可以生成多个中断，如下表所示。

表 15.13 中断请求

中断缩写	中断事件	事件标志	使能控制位	中断清除方法	从睡眠模式退出	从停机和待机退出
TIM	更新	UIF	UIE	UIF 写 0	是	否
	捕获/比较 1	CC1IF	CC1IE	CC1IF 写 0	是	否
	捕获/比较 2	CC2IF	CC2IE	CC2IF 写 0	是	否
	换向（COM）	COMIF	COMIE	COMIF 写 0	是	否
	触发	TIF	TIE	TIF 写 0	是	否
	刹车	BIF	BIE	BIF 写 0	是	否

15.6 TIM15 寄存器

15.6.1 TIM15 控制寄存器 1（TIM15_CR1）

偏移地址：0x00

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			DITHEN	UIFREMAP	Res	CKD[1:0]		ARPE	Reserved			OPM	URS	UDIS	CEN
			rw	rw		rw	rw	rw				rw	rw	rw	rw

Bits 15:13 保留，必须保持为复位值。

Bit 12 **DITHEN**：使能抖动（Dithering enable）

0：抖动禁止

1：抖动使能

*注：*DITHEN *位只能在* CEN *位复位时修改*

Bit 11 **UIFREMAP**：UIF 状态位重映射（UIF status bit remapping）

0：无重映射。UIF 状态位未复制到 TIM15_CNT 寄存器位 31

1：重映射使能。UIF 状态位复制到 TIM15_CNT 寄存器位 31

Bit 10 保留，必须保持为复位值。

Bits 9:8 **CKD[1:0]**：时钟分频（Clock division）

此位域指示定时器时钟（tim_ker_ck）频率以及数字滤波器（tim_etr_in, tim_tix）所使用的采样时钟（t_{DTS}）之间的分频比

00：$t_{DTS} = t_{tim_ker_ck}$

01：$t_{DTS} = 2 \times t_{tim_ker_ck}$

10：$t_{DTS} = 4 \times t_{tim_ker_ck}$

11：保留，不要设置成此值

Bit 7 **ARPE**：自动重载预装载使能（Auto-reload preload enable）

0：TIM15_ARR 寄存器不进行缓冲；

1：TIM15_ARR 寄存器进行缓冲。

Bits 6:4 保留，必须保持为复位值。

Bit 3 **OPM**：单脉冲模式（One pulse mode）

0：计数器在发生更新事件时不会停止计数；

1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）

Bit 2 **URS**：更新请求源（Update request source）

此位由软件置 1 和清零，用以选择 UEV 事件源。

0：使能时，所有以下事件都会生成更新中断。此类事件包括：

- 计数器上溢/下溢
- 将 UG 位置 1
- 通过从模式控制器生成的更新事件

1：使能时，只有计数器上溢/下溢会生成更新中断。

Bit 1 **UDIS**：更新禁止（Update disable）

此位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。

0：使能 UEV。更新（UEV）事件可通过以下事件之一生成：

- 计数器上出/下溢
- 将 UG 位置 1
- 通过从模式控制器生成的更新事件

然后更新影子寄存器的值。

1：禁止 UEV。不会生成更新事件，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。但如果将 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。

Bit 0 **CEN**：计数器使能（Counter enable）

0：禁止计数器；

1：使能计数器。

*注：只有事先通过软件将*CEN *位置*1，*才可以使用外部时钟、门控模式。而触发模式可通过硬件自动将*CEN *位置*1。

15.6.2 TIM15 控制寄存器 2（TIM15_CR2）

偏移地址：0x04

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				OIS3	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			Res	CCUS	Res	CCPC
				rw	rw	rw	rw	rw	rw	rw	rw		rw		rw

Bits 15:12 保留，必须保持为复位值。

Bit 11 **OIS3**：输出空闲状态 3（OC3 输出）(Output idle state 3 (tim_oc3 output)

参见 OIS1 位

Bit 10 **OIS2**：输出空闲状态 2（OC2 输出）(Output idle state 2 (tim_oc2 output)

参见 OIS1 位

Bit 9 **OIS1N**：输出空闲状态 1（OC1N 输出）(Output idle state 1 (tim_oc1n output))

0：当 MOE=0 时，经过死区时间后 tim_oc1n=0

1：当 MOE=0 时，经过死区时间后 tim_oc1n=1

*注：只要编程了*LOCK（TIM15_BDTR *寄存器中的*LOCK *位）级别*1、2 *或*3，*此位即无法修改。*

Bit 8 **OIS1**：输出空闲状态 1（OC1 输出）(Output idle state 1 (tim_oc1 output))

0：当 MOE=0 时，（如果 OC1N 有效，则经过死区时间之后）tim_oc1=0

1：当 MOE=0 时，（如果 OC1N 有效，则经过死区时间之后）tim_oc1=1

*注：只要编程了*LOCK（TIM15_BDTR *寄存器中的*LOCK *位）级别*1、2 *或*3，*此位即无法修改。*

Bit 7 **TI1S**：tim_ti1 选择（tim_ti1 selection）

0：tim_ti1_in[15:0]多路复用器输出连接到 tim_ti1_in 输入

1：tim_ti1_in[15:0]和 tim_ti2_in[15:0]多路复用器输出连接到 tim_ti1_in 输入（异或组合）

Bits 6:4 **MMS[2:0]：**主模式选择（Master mode selection）

这些位可选择主模式下将要发送到从定时器以实现同步的信息（tim_trgo）。这些位的组合如下：

000：**复位** – TIM15_EGR 寄存器中的 UG 位用作触发输出（tim_trgo）。如果复位由触发输入生成（从模式控制器配置为复位模式），则 tim_trgo 上的信号相比实际复位会有延迟。

001：**使能** – 计数器使能信号 CNT_EN 用作触发输出（tim_trgo）。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号可由 CEN 控制位产生。当配置为门控模式时，也可由触发输入产生。当计数器使能信号由触发输入控制时，tim_trgo 上会存在延迟，选择主/从模式时除外（请参见 TIM15_SMCR 寄存器中 MSM 位的说明）。

010：**更新** – 选择更新事件作为触发输出（tim_trgo）。例如，主定时器可用作从定时器的预分频器。

011：**比较脉冲** – 一旦发生输入捕获或比较匹配事件，当 CC1IF 被置 1 时（即使已为高电平），触发输出都会发送一个正脉冲。(tim_trgo)。

100：**比较** – tim_oc1refc 信号用作触发输出（tim_trgo）

101：**比较** – tim_oc1refc 信号用作触发输出（tim_trgo）

其他位：保留

Bit 3 保留，必须保持为复位值。

Bit 2 **CCUS**：捕获/比较控制更新选择（Capture/compare control update selection）

0：如果捕获/比较控制位（CCPC=1）进行预装载，仅通过将 COMG 位置 1 来对这些位进行更新；

1：如果捕获/比较控制位（CCPC=1）进行预装载，可通过将 COMG 位置 1 或 TRGI 的上升沿对这些位进行更新。

注：此位仅对具有互补输出的通道有效。

Bit 1 保留，必须保持为复位值。

Bit 0 **CCPC**：捕获/比较预装载控制（Capture/compare preloaded control）

0：CCxE，CCxNE 和 OCxM 位未进行预装载；

1：CCxE，CCxNE 和 OCxM 位进行了预装载，写入这些位后，仅当发生换向事件（COM）（COMG 位置 1 或在 TRGI 上检测到上升沿，取决于 CCUS 位）时才会对这些位进行更新。

注：此位仅对具有互补输出的通道有效。

15.6.3 TIM15 从模式控制寄存器（TIM15_SMCR）

偏移地址：0x08

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															SMS[3]
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								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（Slave mode selection）

Bits 15:8 保留，必须保持复位值。

Bit 7 **MSM**：主/从模式（Master/slave mode）

0：不执行任何操作；

1：当前定时器的触发输入事件（tim_trgi）的动作被推迟，以使当前定时器与其从定时器实现完美同步（通过 tim_trgo）。此设置适用于由单个外部事件对多个定时器进行同步的情况。

Bits 6:4 **TS[2:0]**：触发选择（Trigger selection）

此位域可选择将要用于同步计数器的触发输入。

000：内部触发 0（tim_itr0）

001：内部触发 1（tim_itr1）

010：内部触发 2（tim_itr2）

100：tim_ti1 边沿检测器(tim_ti1f_ed)

101：滤波后的定时器输入 1（tim_ti1fp1）

110：滤波后的定时器输入 1（tim_ti2fp2）

111：保留

有关定时器 ITRx 含义的详细信息，请参见章节：TIM15 引脚和内部信号。

注：这些位只能在未使用的情况下（例如，SMS=000 时）进行更改，以避免转换时出现错误的边沿检测。

Bit 3 保留，必须保持为复位值。

Bits 16, 2:0 **SMS[3:0]**：从模式选择（Slave mode selection）

当选择了外部信号，触发信号（tim_trgi）的有效边沿与选中的外部输入极性相关（见输入控制寄存器和控制寄存器的说明）

0000：关闭从模式 – 如果 CEN=1，则预分频器直接由内部时钟驱动。

0001：保留

0010：保留

0011：保留

0100：**复位模式** – 选中的触发输入（tim_trgi）的上升沿重新初始化计数器，并且产生一个更新寄存器的信号。

0101：**门控模式** – 当触发输入（tim_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。

0110：**触发模式** – 计数器在触发输入 tim_trgi 的上升沿启动（但不复位），只有计数器

的启动是受控的。

0111：**外部时钟模式 1** - 选中的触发输入（TRGI）的上升沿驱动计数器。

1000：**组合重置+触发模式** - 当选择的触发输入（tim_trgi）上升沿出现时，会重新初始化计数器，生成寄存器的更新并启动计数器。

1001：**组合门控+复位模式** - 当触发输入（tim_trgi）为高电平时，计数器时钟被使能。一旦触发变为低电平，计数器就会停止并重置。这样，计数器的启动和停止都得到了控制。

其他：保留

*注：如果选择了*tim_ti1f_ed *作为触发输入（*TS=00100*），则不能使用门控模式。事实上，*tim_ti1f_ed *为每个*tim_ti1f *上的转换输出一个脉冲，而门控模式检查触发信号的电平。*

*注：接收*tim_trgo *信号的从设备（定时器、*ADC *等）的时钟必须在从主定时器接收事件之前使能，并且时钟频率（预分频器）在从主定时器接收触发时不能改变。*

15.6.4 TIM15 中断使能寄存器（TIM15_DIER）

偏移地址：0x0C

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BIE	TIE	COMIE	Res	CC3IE	CC2IE	CC1IE	UIE
								rw	rw	rw		rw	rw	rw	rw

Bits 15:8 保留，必须保持为复位值。

Bit 7 **BIE**：刹车中断使能（Break interrupt enable）

0：禁止刹车中断；

1：使能刹车中断。

Bit 6 **TIE**：触发信号（TGRI）中断使能（Trigger interrupt enable）

0：禁止触发信号(TGRI)中断；

1：使能触发信号(TGRI)中断。

Bit 5 **COMIE**：COM 中断使能（COM interrupt enable）

0：禁止 COM 中断；

1：使能 COM 中断。

Bit 4 保留，必须保持为复位值。

Bit 3 **CC3IE**：捕获/比较 3 中断使能（Capture/Compare 3 interrupt enable）

0：禁止 CC3 中断；

1：使能 CC3 中断。

Bit 2 **CC2IE**：捕获/比较 2 中断使能（Capture/Compare 2 interrupt enable）

0：禁止 CC2 中断；

1：使能 CC2 中断。

Bit 1 **CC1IE**：捕获/比较 1 中断使能（Capture/Compare 1 interrupt enable）

0：禁止 CC1 中断；

1：使能 CC1 中断。

Bit 0 **UIE**：更新中断使能（Update interrupt enable）

0：禁止更新中断；

1：使能更新中断。

15.6.5 TIM15 状态寄存器（TIM15_SR）

偏移地址：0x10

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CC3OF	CC2OF	CC1OF	Res	BIF	TIF	COMIF	Res	CC3IF	CC2IF	CC1IF	UIF
				rc w0	rc w0	rc w0		rc w0	rc w0	rc w0		rc w0	rc w0	rc w0	rc w0

Bits 15:12 保留，必须保持为复位值。

Bit 11 **CC3OF**：捕获/比较 3 重复捕获标志

参考 CC1OF 的描述

Bit 10 **CC2OF**：捕获/比较 2 重复捕获标志

参考 CC1OF 的描述

Bit 9 **CC1OF**：捕获/比较 1 重复捕获标志

仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将该位清零。

0：未检测到重复捕获

1：TIM15_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。

Bit 8 保留，必须保持复位值。

Bit 7 **BIF**：刹车中断标志（Break interrupt flag）

只要刹车输入变为有效状态，此标志便由硬件置 1。刹车输入无效后可通过软件对其清零。

0：未发生刹车事件。

1：在刹车输入上检测到有效电平。

Bit 6 **TIF**：触发中断标志（Trigger interrupt flag）

在除门控模式以外的所有模式下，当使能从模式控制器后在 tim_trgi 输入上检测到有效边沿时，该标志将由硬件置 1。选择门控模式时，该标志将在计数器启动或停止时置 1。但需要通过软件清零。

0：未发生触发事件。

1：触发中断挂起。

Bit 5 **COMIF**：COM 中断标志（COM interrupt flag）

此标志在发生 COM 事件时（捕获/比较控制位 CCxE、CCxNE 和 OCxM 已更新时）由硬件置 1。但需要通过软件清零。

0：未发生 COM 事件。

1：COM 中断挂起。

Bit 4 保留，必须保持为复位值。

Bit 3 **CC3IF**：捕获/比较 3 中断标志（Capture/Compare 3 interrupt flag）

参考 CC1IF 描述。

Bit 2 **CC2IF**：捕获/比较 2 中断标志（Capture/Compare 2 interrupt flag）

参考 CC1IF 描述。

Bit 1 **CC1IF**：捕获/比较 1 中断标志（Capture/Compare 1 interrupt flag）

此标志由硬件设置，它可以通过软件（输入捕获或输出比较模式）或读取 TIM15_CCR1 寄存器（仅输入捕获模式）来清除。

0：未比较匹配/无输入捕获发生

1：发生比较匹配或输入捕获

如果通道 CC1 配置为输出：当计数器与比较值匹配时，此标志由硬件置 1，当 TIM15_CCR1 的值大于 TIM15_ARR 的值时，CC1IF 位将在计数器发生上溢（递增计数模式和增减计数模式下）或下溢（递减计数模式下）时变为高电平。在中心对齐模式下有 3 种可能的标志设置选项，请参考 TIM15_CR1 寄存器中的 CMS 位以获取完整描述。

如果通道 CC1 配置为输入：当在 TIM15_CCR1 寄存器中捕获到计数器值时（在 IC1 上检测到边沿，具体取决于在 TIM15_CCER 中通过 CC1P 和 CC1NP 位设置定义的边沿灵敏度），会设置此位。

Bit 0 **UIF**：更新中断标志（Update interrupt flag）

该位在发生更新事件时通过硬件置 1。但需要通过软件清零。

0：未发生更新。

1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：

- TIM15_CR1 寄存器中的 UDIS=0，并且重复计数器值上溢或下溢时（重复计数器=0 时更新）。
- TIM15_CR1 寄存器中的 URS=0 且 UDIS=0，并且由软件使用 TIM15_EGR 寄存器中的 UG 位重新初始化 CNT 时。
- TIM15_CR1 寄存器中的 URS=0 且 UDIS=0，并且 CNT 由触发事件重新初始化时（参考 TIM15 从模式控制寄存器（TIM15_SMCR））。

15.6.6 TIM15 事件产生寄存器（TIM15_EGR）

偏移地址：0x14

复位：0x0000 0000

访问：无等待，支持字，半字和字节访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BG	TG	COMG	Res	CC3G	CC2G	CC1G	UG
								w	w	rw		w	w	w	w

Bits 15:8 保留，必须保持为复位值。

Bit 7 **BG**：刹车生成（Break generation）

此位由软件置 1 以生成事件，并由硬件自动清零。

0：不执行任何操作；

1：生成刹车事件。MOE 位清零且 BIF 标志置 1。使能后可发生相关中断事件。

Bit 6 **TG**：生成触发信号（Trigger generation）

此位由软件置 1 以生成事件，并由硬件自动清零。

0：不执行任何操作；

1：TIM15_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断事件。

Bit 5 **COMG**：捕获/比较控制更新生成（Capture/Compare control update generation）

该位可通过软件置 1，并由硬件自动清零

0：不执行任何操作；

1：CCPC 位置 1 时，可更新 CCxE、CCxNE 和 OCxM 位。

注：此位仅对具有互补输出的通道有效。

Bit 4 保留，必须保持为复位值。

Bit 3 **CC3G**：捕获/比较 3 生成（Capture/Compare 3 generation）

参考 CC1G 描述。

Bit 2 **CC2G**：捕获/比较 2 生成（Capture/Compare 2 generation）

参考 CC1G 描述。

Bit 1 **CC1G**：捕获/比较 1 生成（Capture/Compare 1 generation）

此位由软件置 1 以生成事件，并由硬件自动清零。

0：不执行任何操作；

1：通道 1 上生成捕获/比较事件：

若通道 CC1 配置为输出：

使能时，CC1IF 标志置 1 并发送相应的中断。

如果通道 CC1 配置为输入：

TIM15_CCR1 寄存器中将捕获到计数器当前值。使能时，CC1IF 标志置 1 并发送相应的中断请求。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。

Bit 0 **UG**：更新生成（Update generation）

该位可通过软件置 1，并由硬件自动清零。

0：不执行任何操作；

1：重新初始化计数器并生成一个寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。如果选择 DIR=0（递增计数），计数器将清零；如果 DIR=1（递减计数），计数器将使用自动重载值（TIM15_ARR）。

15.6.7 TIM15 捕获/比较模式寄存器 1 [复用]（TIM15_CCMR1）

偏移地址：0x18

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

相同的寄存器可以用于输入捕获模式（本节）或输出比较模式（下节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其它位在输入模式和输出模式下的功能均不同。可以独立组合这两种模式（例如，通道 1 用作输入捕获模式而通道 2 用作输出比较模式）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IC2F[3:0]				IC2PSC[1:0]		CC2S[1:0]		IC1F[3:0]				IC1PSC[1:0]		CC1S[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

输入捕获模式

Bits 31:16 保留，必须保持为复位值。

Bits 15:12 **IC2F[3:0]：**输入捕获 2 滤波器（Input capture 2 filter）

Bits 11:10 **IC2PSC[1:0]：**输入/捕获 2 预分频器（input capture 2 prescaler）

Bits 9:8 **CC2S[1:0]：**捕获/比较 2 选择（Capture/compare 2 selection）

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC2 通道被配置为输出；

01：CC2 通道被配置为输入，tim_ic2 映射在 tim_ti2 上；

10：CC2 通道被配置为输入，tim_ic2 映射在 tim_ti1 上；

11：CC2 通道被配置为输入，tim_ic2 映射在 tim_trc 上。此模此模式仅在通过 TS 位（TIM15_SMCR 寄存器）选择内部触发输入时有效

注：仅当通道关闭时（TIM15_CCER 中的 CC2E = 0），才可向 CC2S 位写入数据。

Bits 7:4 **IC1F[3:0]：**输入捕获 1 滤波器（Input capture 1 filter）

此位域可定义 tim_ti1 输入的采样频率和适用于 tim_ti1 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个事件才视为一个有效边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样

0001：$f_{SAMPLING}=f_{tim_ker_ck}$，N=2。

0010：$f_{SAMPLING}=f_{tim_ker_ck}$，N=4。

0011：$f_{SAMPLING}=f_{tim_ker_ck}$，N=8。

0100：$f_{SAMPLING}=f_{DTS}/2$，N=6。

0101：$f_{SAMPLING}=f_{DTS}/2$，N=8。

0110：$f_{SAMPLING}=f_{DTS}/4$，N=6。

0111：$f_{SAMPLING}=f_{DTS}/4$，N=8。

1000：$f_{SAMPLING}=f_{DTS}/8$，N=6。

1001：$f_{SAMPLING}=f_{DTS}/8$，N=8。

1010：$f_{SAMPLING}=f_{DTS}/16$，N=5。

1011：$f_{SAMPLING}=f_{DTS}/16$，N=6。

1100：$f_{SAMPLING}=f_{DTS}/16$，N=8。

1101：$f_{SAMPLING}=f_{DTS}/32$，N=5。

1110：$f_{SAMPLING}=f_{DTS}/32$，N=6。

1111：$f_{SAMPLING}=f_{DTS}/32$，N=8。

Bits 3:2 **IC1PSC[1:0]：**输入/捕获 1 预分频器（Input capture 1 prescaler）

此位域定义 CC1 输入（tim_ic1）的预分频比。

只要 CC1E=0（TIM15_CCER 寄存器），预分频器便立即复位。

00：无预分频器，捕获输入上每检测到一个边沿便执行捕获

01：每发生 2 个事件便执行一次捕获；

10：每发生 4 个事件便执行一次捕获；

11：每发生 8 个事件便执行一次捕获。

Bits 1:0 **CC1S[1:0]：**捕获/比较 1 选择（Capture/Compare 1 selection）

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道被配置为输出；

01：CC1 通道被配置为输入，tim_ic1 映射在 tim_ti1 上；

10：CC1 通道被配置为输入，tim_ic1 映射在 tim_ti2 上；

11：CC1 通道被配置为输入，tim_ic1 映射在 tim_trc 上。此模式仅在通过 TS 位（TIM15_SMCR 寄存器）选择内部触发输入时有效

注：仅当通道关闭时（TIM15_CCER 中的 CC1E = 0），才可向 CC1S 位写入数据。

15.6.8 TIM15 捕获/比较模式寄存器 1 [复用]（TIM15_CCMR1）

偏移地址：0x18

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

相同的寄存器可以用于输出比较模式（本节）或输入捕获模式（上节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其它位在输入模式和输出模式下的功能均不同。可以独立组合这两种模式（例如，通道 1 用作输入捕获模式而通道 2 用作输出比较模式）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							OC2M[3]	Reserved							OC1M[3]
							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	rw	rw	rw	rw	rw	rw

输出比较模式

Bits 31:25 保留，必须保持复位值。

Bit 24 **OC2M[3]：**输出比较 2 模式（Output Compare 2 mode）

Bits 23:17 保留，必须保持复位值。

Bit 15 **OC2CE：**输出比较 2 清零使能（Output Compare 2 clear enable）

Bits 24, 14:12 **OC2M[2:0]：**输出比较 2 模式（Output Compare 2 mode）

参考 OC1M[3:0]描述

Bit 11 **OC2PE：**输出比较 2 预装载使能（Output Compare 2 preload enable）

Bit 10 **OC2FE：**输出比较 2 快速使能（Output Compare 2 fast enable）

Bits 9:8 **CC2S[1:0]：**捕获/比较 2 选择（Capture/Compare 2 selection）

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC2 通道配置为输出

01：CC2 通道配置为输入，tim_ic2 映射到 tim_ti2 上

10：CC2 通道配置为输入，tim_ic2 映射到 tim_ti1 上

11：CC2 通道配置为输入，tim_ic2 映射到 tim_trc 上。此模式仅在通过 TS 位（TIM15_SMCR 寄存器）选择内部触发输入时有效

注：仅当通道关闭时（TIM15_CCER 中的 CC2E = 0），才可向 CC2S 位写入数据

Bit 7 **OC1CE：**输出比较 1 清零使能（Output Compare 1 clear enable）

0：tim_oc1ref 不受 tim_ocref_clr_int 输入影响；

1：tim_oc1ref 在 tim_ocref_clr_int 输入中被检测到高电平，tim_oc1ref 立即清零。

Bits 16, 6:4 **OC1M[3:0]：**输出比较 1 模式（Output Compare 1 mode）

这些位定义提供 tim_oc1 和 tim_oc1n 的输出参考信号 tim_oc1ref 的行为。tim_oc1ref 为高电平有效，而 tim_oc1 和 tim_oc1n 的有效电平则取决于 CC1P 位和 CC1NP 位。

0000：**冻结** — 输出比较寄存器 TIM15_CCR1 与计数器 TIM15_CNT 进行比较不会对输出造成任何影响。（该模式用于生成时基）。

0001：将通道 1 设置为匹配时输出有效电平。当计数器 TIM15_CNT 与捕获/比较寄存器 1（TIM15_CCR1）匹配时，tim_oc1ref 信号强制变为高电平。

0010：将通道 1 设置为匹配时输出无效电平。当计数器 TIM15_CNT 与捕获/比较寄存器

1（TIM15_CCR1）匹配时，tim_oc1ref 信号强制变为低电平。

0011：**翻转** – TIM15_CNT=TIM15_CCR1 时，tim_oc1ref 发生翻转。

0100：**强制变为无效电平** – tim_oc1ref 强制变为低电平。

0101：**强制变为有效电平** – tim_oc1ref 强制变为高电平。

0110：**PWM 模式 1** – 在递增计数模式下，只要 TIM15_CNT<TIM15_CCR1，通道 1 便为有效状态，否则为无效状态。

0111：**PWM 模式 2** – 在递增计数模式下，只要 TIM15_CNT<TIM15_CCR1，通道 1 便为无效状态，否则为有效状态。

1000：**可重触发 OPM 模式 1** - 在向上计数模式下，通道在检测到触发事件（tim_trgi 信号）之前一直处于活动状态。然后，会像 PWM 模式 1 那样执行比较，并在下次更新时使通道再次处于活动状态。在向下计数模式下，通道在检测到触发事件（tim_trgi 信号）之前一直处于非活动状态。然后，会像 PWM 模式 1 那样执行比较，并在下次更新时使通道再次处于非活动状态。

1001：**可重触发 OPM 模式 2** - 在向上计数模式下，通道在检测到触发事件（tim_trgi 信号）之前一直处于非活动状态。然后，会像 PWM 模式 2 那样执行比较，并在下次更新时使通道再次处于非活动状态。在向下计数模式下，通道在检测到触发事件（tim_trgi 信号）之前一直处于活动状态。然后，会像 PWM 模式 1 那样执行比较，并在下次更新时使通道再次处于活动状态。

1010：保留

1011：保留

1100：**组合 PWM 模式 1** - tim_oc1ref 的行为与 PWM 模式 1 相同。tim_oc1refc 是 tim_oc1ref 和 tim_oc2ref 之间的逻辑或。

1101：**组合 PWM 模式 2** - tim_oc1ref 的行为与 PWM 模式 2 相同。tim_oc1refc 是 tim_oc1ref 和 tim_oc2ref 之间的逻辑与。

1110：保留

1111：保留

*注：只要编程了*LOCK *（*TIM15_BDTR *寄存器中的*LOCK *位）级别*3 *且*CC1S= *“*00*”（通道配置为输出），这些位即无法修改。*

在 PWM *模式下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“*PWM*”模式时，*tim_ocref_clr *电平才会发生更改。*

*对于具有互补输出的通道，此位字段是预加载的。如果*TIM15_CR2 *寄存器中*CCPC *位被设置，则*OC1M *有效位仅在生成*COM *事件时从预加载位获取新值。*

Bit 3 **OC1PE**：输出比较 1 预装载使能（Output Compare 1 preload enable）

0：禁止与 TIM15_CCR1 相关的预装载寄存器。可随时向 TIM15_CCR1 写入数据，写入后将立即使用新值。

1：使能与 TIM15_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIM15_CCR1 预装载值在每次生成更新事件时都会装载到有效寄存器中。

*注：只要编程了*LOCK *（*TIM15_BDTR *寄存器中的*LOCK *位）级别*3 *且*CC1S=00 *（通道配置为输出），这些位即无法修改。*

Bit 2 **OC1FE**：输出比较 1 快速使能（Output Compare 1 fast enable）

此位减少了触发事件和定时器输出上的转换之间的延迟。它必须在单脉冲模式（在 TIM15_CR1 寄存器中设置 OPM 位）中使用，以使输出脉冲在启动触发后尽快开始。

0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的最短延迟时间为 5 个时钟周期。

1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OCFE 才会起作用。

Bits 1:0 **CC1S[1:0]：**捕获/比较 1 选择（Capture/Compare 1 selection）

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出

01：CC1 通道配置为输入，tim_ic1 映射到 tim_ti1 上

10：CC1 通道配置为输入，tim_ic1 映射到 tim_ti2 上

11：CC1 通道配置为输入，tim_ic1 映射到 tim_trc 上。此模式仅在通过 TS 位（TIM15_SMCR 寄存器）选择内部触发输入时有效

*注：仅当通道关闭时（*TIM15_CCER *中的* CC1E = 0*），才可向* CC1S *位写入数据*

15.6.9 TIM15 捕获/比较模式寄存器 2 [复用]（TIM15_CCMR2）

偏移地址：0x1C

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

相同的寄存器可以用于输入捕获模式（本节）或输出比较模式（下节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其它位在输入模式和输出模式下的功能均不同。可以独立组合这两种模式（例如，通道 1 用作输入捕获模式而通道 2 用作输出比较模式）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								IC3F[3:0]				IC3PSC[1:0]		CC3S[1:0]	
								rw	rw	rw	rw	rw	rw	rw	rw

输入捕获模式

Bit 31:8 保留，必须保持为复位值。

Bit 7:4 **IC3F[3:0]：**输入捕获 3 滤波器（Input capture 3 filter）

参考 IC1F 描述

Bit 3:2 **IC3PSC[1:0]：**输入/捕获 1 预分频器（Input capture 3 prescaler）

参考 IC1PSC 描述

Bit 1:0 **CC3S[1:0]：**捕获/比较 3 选择（Capture/Compare 3 selection）

参考 CC1S 描述

15.6.10 TIM15 捕获/比较模式寄存器 2 [复用]（TIM15_CCMR2）

偏移地址：0x1C

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

相同的寄存器可以用于输出比较模式（本节）或输入捕获模式（上节）。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其它位在输入模式和输出模式下的功能均不同。可以独立组合这两种模式（例如，通道 1 用作输入捕获模式而通道 2 用作输出比较模式）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															OC3M[3]
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
								rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式

Bit 31:17 保留，必须保持为复位值。

Bit 16 **OC3M[3]**：输出比较 3 模式（Output compare 3 mode）

Bit 15:8 保留，必须保持为复位值。

Bit 7 **OC3CE**：输出比较 3 清零使能（Output Compare 3 clear enable）

参考 OC1CE 描述

Bit 16, 6:4 **OC3M[3:0]**：输出比较 3 模式（Output Compare 1 mode）

参考 OC1M 描述

Bit 3 **OC3PE**：输出比较 3 预装载使能（Output Compare 3 preload enable）

参考 OC1PE 描述

Bit 2 **OC3FE**：输出比较 3 快速使能（Output Compare 3 fast enable）

参考 OC1FE 描述

Bit 1:0 **CC3S[1:0]**：捕获/比较 3 选择（Capture/Compare 3 selection）

参考 CC1S 描述

15.6.11 TIM15 捕获/比较使能寄存器（TIM15_CCER）

偏移地址：0x20

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CC3P	Res	CC3E	CC3NP	CC2NP	Res	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
				rw		rw	rw	rw		rw	rw	rw	rw	rw	rw

Bits 15:12 保留，必须保持为复位值。

Bit 11 **CC3NP**：输入/捕获 3 输出极性（Capture/Compare 3 output polarity）

参考 CC1NP 的描述。

Bit 10 保留，必须保持为复位值。

Bit 9 **CC3P**：输入/捕获 3 输出极性（Capture/Compare 3 output polarity）

参考 CC1P 的描述。

Bit 8 **CC3E**：输入/捕获 3 输出使能（Capture/Compare 3 output enable）

Bit 7 **CC2NP**：输入/捕获 2 输出极性（Capture/Compare 2 output polarity）

参考 CC1NP 的描述。

Bit 6 保留，必须保持为复位值。

Bit 5 **CC2P**：输入/捕获 2 输出极性（Capture/Compare 2 output polarity）

参考 CC1P 的描述。

Bit 4 **CC2E**：输入/捕获 2 输出使能（Capture/Compare 2 output enable）

参考 CC1E 的描述。

Bit 3 **CC1NP**：输入/捕获 1 输出极性（Capture/Compare 1 output polarity）

CC1 通道配置为输出：

0：OC1N 高电平有效。

1：OC1N 低电平有效。

CC1 通道配置为输入：

此位与 CC1P 配合使用，用以定义 tim_ti1fp1/ tim_ti2fp1 的极性。请参见 CC1P 说明。

*注：只要编程了*LOCK *（*TIM15_BDTR *寄存器中的*LOCK *位）级别*2 *或*3 *且* CC1S=00 *（通道配置为输出），此位立即变为不可写状态*

注：此位将在具有互补输出的通道上进行预装载。如果 TIM15_CR2 *寄存器中的*CCPC *位置*1*，则仅当生成换向事件时，*CC1NP *有效位才会从预装载位获取新值。*

Bit 2 **CC1NE**：输入/捕获 1 互补输出使能（Capture/Compare 1 complementary output enable）

0：关闭 --OC1N 未激活。OC1N 电平是 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的函数。

1：开启 -- 在相应输出引脚上输出 OC1N 信号，具体取决于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位。

Bit 2 **CC1NE**：输入/捕获 1 互补输出使能（Capture/Compare 1 complementary output enable）

0：关闭 --OC1N 未激活。OC1N 电平是 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的函数。

1：开启 -- 在相应输出引脚上输出 OC1N 信号，具体取决于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位。

Bit 1 **CC1P**：输入/捕获 1 输出极性（Capture/Compare 1 output polarity）

CC1 通道配置为输出：

0：OC1 高电平有效（输出模式）/边沿灵敏度选择（输入模式，参见以下）

1：OC1 低电平有效（输出模式）/边沿灵敏度选择（输入模式，参见以下）

CC1 通道配置为输入，CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的极性。

CC1NP=0，CC1P=0：非反相/上升沿触发。电路对 TIxFP1 上升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式或编码器模式下执行触发操作）。

CC1NP=0，CC1P=1：反相/下降沿触发。电路对 TIxFP1 下降沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 反相（在门控模式或编码器模式下执行触发操作）。

CC1NP=1，CC1P=1：非反相/上升沿和下降沿均触发。电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。编码

器模式下不得使用此配置。

CC1NP=1，CC1P=0：此配置是保留的，不应该使用此配置。

*注：只要编程了*LOCK（TIM15_BDTR *寄存器中的* LOCK *位）级别*2 *或*3，*此位立即变为不可写状态。*

注：此位将在具有互补输出的通道上进行预装载。如果 TIM15_CR2 *寄存器中的*CCPC *位置*1，*则仅当生成换向事件时，*CC1P *有效位才会从预装载位获取新值。*

Bit 0 **CC1E**：输入/捕获 1 输出使能（Capture/Compare 1 output enable）

0：禁止捕获模式/OC1 未激活（参见以下）

1：使能捕获模式/ OC1 信号输出到对应的输出引脚

当 CC1 通道配置为输出，C1 的输出电平依赖于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的值，不论 CC1E 位状态。请参考下表获得更多细节

表 15.14 带刹车功能的互补输出通道 tim_oc1 和 tim_oc1n 的控制位（TIM15）

控制位					输出状态 (1)	
MOE 位	OSSI 位	OSSR 位	CCxE 位	CCxNE 位	tim_ocx 输出状态	tim_ocxn 输出状态
1	X	X	0	0	禁止输出（不由定时器驱动: Hi-Z） tim_ocx=0，tim_ocxn=0	
		0	0	1	禁止输出（不由定时器驱动: Hi-Z） tim_ocx=0	tim_ocxref + 极性 tim_ocxn =tim_ocxref 异或 CCxNP
		0	1	0	tim_ocxref + 极性 tim_ocx=tim_ocxref 异或 CCxP	禁止输出（不由定时器驱动：Hi-Z） tim_ocxn=0
		X	1	1	OCREF+极性+死区	OCREF 互补项（而非 OCREF） +极性+死区
		1	0	1	关闭状态（输出使能为无效状态） tim_ocx=CCxP	tim_ocxref + 极性 tim_ocxn =tim_ocxref 异或 CCxNP
		1	1	0	tim_ocxref + 极性 tim_ocx=tim_ocxref 异或 CCxP	关闭状态（输出使能为无效状态） tim_ocxn=CCxNP
0	0	X	X	X	禁止输出（不由定时器驱动: Hi-Z）	
	1		0	0		
			0	1	关闭状态（输出使能为无效状态）	
			1	0	异步：tim_ocx=CCxP，tim_ocxn=CCxNP	
			1	1	然后如果存在时钟：在死区后 tim_ocx=OISx 且 tim_ocxn=OISxN，从而假定 OISx 和 OISxN 在有效状态下与 OCX 和 OCxN 不对应	

1. 如果一个通道的两个输出均未使用（CCxE = CCxNE = 0），则 OISx、OISxN、CCxP 和 CCxNP 位必须保持清零状态。

*注：与互补通道*tim_ocx *和*tim_ocxn *相连的外部*I/O *引脚的状态取决于通道*tim_ocx *和*tim_ocxn *的状态以及*GPIO *寄存器。*

15.6.12 TIM15 计数器（TIM15_CNT）

偏移地址：0x24
复位值：0x0000
访问：无等待，支持字，半字和字节访问

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

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

Bit 31 **UIFCPY：**复制 UIF

该位是 TIM15_SR 寄存器中 UIF 位的一个只读副本。

Bits 30:16 保留，必须保持为复位值。

Bits 15:0 **CNT[15:0]：**计数器的值（Counter value）

在非抖动模式下（DITHEN = 0）

该寄存器（CNT[15:0]）保存的是计数器的值。

在抖动模式下（DITHEN = 1）

该寄存器（CNT[15:0]）只保存计数器的不抖动部分。抖动部分不可用。

15.6.13 TIM15 预分频器（TIM15_PSC）

偏移地址：0x28
复位值：0x0000
访问：无等待，支持字，半字和字节访问

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

Bits 15:0 **PSC[15:0]：**预分频器的值（Prescaler value）

计数器时钟频率（$f_{tim_cnt_ck}$）等于 $f_{tim_psc_ck}$ / (PSC[15:0] + 1)。

PSC 包含每次发生更新事件（包括计数器通过 TIM15_EGR 寄存器中的 UG 位清零时，或在配置为“复位模式”时通过触发控制器清零时）时要装载到活动预分频器寄存器的值。

15.6.14 TIM15 自动重装载寄存器（TIM15_ARR）

偏移地址：0x2C

复位：0x0000 FFFF

访问：无等待，支持字，半字和字节访问

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

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

Bits 31:20 保留，必须保持为复位值。

Bits 19:0 **ARR[19:0]：**自动重装载的值（Auto-reload value）

ARR 为要装载到实际自动重载寄存器的值。

当自动重载值为空时，计数器不工作。

在非抖动模式下（DITHEN = 0）

该寄存器存储的是自动重新加载的值

在抖动模式下（DITHEN = 1）

该寄存器存储的是 ARR[19:4]中的整数部分，而 ARR[3:0]位域存储的是抖动的部分

15.6.15 TIM15 重复计数寄存器（TIM15_RCR）

偏移地址：0x30

复位值：0x0000

访问：无等待，支持字，半字和字节访问

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

Bits 15:8 保留，必须保持复位值。

Bits 7:0 **REP[7:0]**：重复计数器的值（Repetition counter value）

使能预装载寄存器时，用户可通过这些位设置比较寄存器的更新频率（即从预装载寄存器向活动寄存器周期性传输数据）；使能更新中断时，也可设置更新中断的生成速率。与 REP_CNT 相关的减计数器每次计数到 0 时，都将生成一个更新事件并且计数器从 REP 值重新开始计数。由于只有生成重复更新事件 U_RC 时，REP_CNT 才会重载 REP 值，因此在生成下一重复更新事件之前，无论向 TIM15_RCR 寄存器写入何值都无影响。

这意味着 PWM 模式下（REP+1）相当于：

– 边沿对齐模式下的 PWM 周期数；

– 中央对称模式下的 PWM 半周期数；

15.6.16 TIM15 捕获/比较寄存器 1（TIM15_CCR1）

偏移地址：0x34

复位值：0x0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												CCR1 [19:16]			
												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:20 保留，必须保持为复位值。

Bits 19:0 **CCR1[19:0]：**捕获/比较 1 值（Capture/Compare 1 value）

若 CC1 通道配置为输出：

CCR1 为要装载到实际捕获/比较 1 寄存器的值（预装载值）。

如果没有通过 TIM15_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 1 寄存器中。实际捕获/比较寄存器中包含要与计数器 TIM15_CNT 进行比较并在 tim_oc1 输出上发出信号的值。

在非抖动模式下（DITHEN = 0）

该寄存器存储的是 CCR1[15:0]中的比较值。CCR1[19:16]位会被重置。

在抖动模式下（DITHEN = 1）

该寄存器存储的是 CCR1[19:4]中的整数部分。CCR1[3:0]位域存储的是抖动的部分。

若 CC1 通道配置为输入：

CCR1 为上一个输入捕获 1 事件（tim_ic1）发生时的计数器值。TIM15_CCR1 是只读的并且不可编程。

在非抖动模式下（DITHEN = 0）

CCR1[15:0]位存储了捕获值。CCR1[19:16]位会被重置。

在抖动模式下（DITHEN = 1）

该寄存器存储捕获在 CCR1[19:0]中。CCR1[3:0]位会被重置。

15.6.17 TIM15 捕获/比较寄存器 2（TIM15_CCR2）

偏移地址：0x38

复位值：0x0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												CCR2 [19:16]			
												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:20 保留，必须保持为复位值。

Bits 19:0 **CCR2[19:0]：**捕获/比较 2 值（Capture/Compare 2 value）

若 CC2 通道配置为输出：

CCR2 为要装载到实际捕获/比较 2 寄存器的值（预装载值）。

如果没有通过 TIM15_CCMR1 寄存器中的 OC2PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 2 寄存器中。实际捕获/比较寄存器中包含要与计数器 TIM15_CNT 进行比较并在 tim_oc2 输出上发出信号的值。

在非抖动模式下（DITHEN = 0）

该寄存器存储的是 CCR2[15:0]中的比较值。CCR2[19:16]位会被重置。

在抖动模式下（DITHEN = 1）

该寄存器存储的是 CCR2[19:4]中的整数部分。CCR2[3:0]位域存储的是抖动的部分。

若 CC2 通道配置为输入：

CCR2 为上一个输入捕获 2 事件（tim_ic2）发生时的计数器值。TIM15_CCR2 是只读的并且不可编程。

在非抖动模式下（DITHEN = 0）

CCR2[15:0]位存储了捕获值。CCR2[19:16]位会被重置。

在抖动模式下（DITHEN = 1）

该寄存器存储捕获在 CCR2[19:0]中。CCR2[3:0]位会被重置。

15.6.18 TIM15 捕获/比较寄存器 3（TIM15_CCR3）

偏移地址：0x3C

复位值：0x0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												CCR3 [19:16]			
												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3 [15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:20 保留，必须保持为复位值。

Bits 19:0 **CCR3[19:0]：**捕获/比较 3 值（Capture/Compare 3 value）

若 CC3 通道配置为输出：

CCR3 为要装载到实际捕获/比较 3 寄存器的值（预装载值）。

如果没有通过 TIM15_CCMR1 寄存器中的 OC3PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 3 寄存器中。实际捕获/比较寄存器中包含要与计数器 TIM15_CNT 进行比较并在 tim_oc3 输出上发出信号的值。

在非抖动模式下（DITHEN = 0）

该寄存器存储的是 CCR3[15:0]中的比较值。CCR3[19:16]位会被重置。

在抖动模式下（DITHEN = 1）

该寄存器存储的是 CCR3[19:4]中的整数部分。CCR3[3:0]位域存储的是抖动的部分。

若 CC3 通道配置为输入：

CCR3 为上一个输入捕获 3 事件（tim_ic3）发生时的计数器值。TIM15_CCR3 是只读的并且不可编程。

在非抖动模式下（DITHEN = 0）

CCR3[15:0]位存储了捕获值。CCR3[19:16]位会被重置。

在抖动模式下（DITHEN = 1）

该寄存器存储捕获在 CCR3[19:0]中。CCR3[3:0]位会被重置。

15.6.19 TIM15 刹车和死区寄存器（TIM15_BDTR）

偏移地址：0x44

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			BKBID	Res	BK DSRM	Reserved						BKF[3:0]			
			rw		rw							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	rw	rw	rw	rw	rw	rw	rw	rw

*注：由于可以根据*LOCK *配置锁定位*BKBID，BKDSRM，BKF[3:0]，AOE，BKP，BKE，OSSI，OSSR *和*DTG[7:0]*的写操作，因此必须在第一次对*TIM15_BDTR *寄存器执行写访问时对这些位进行配置。*

Bits 31:29 保留，必须保持为复位值。

Bit 28 **BKBID：**刹车双向（Break bidirectional）

0：在输入模式下刹车输入 tim_brk

1：在双向模式下刹车输入 tim_brk

在双向模式（BKBID 位设置为 1）中，刹车输入在输入和输出中都配置模式和开漏输出模式。任何活动中断事件都会在刹车输入，向外部设备指示内部中断事件。

*注：只要编程了*LOCK（TIM15_BDTR *寄存器中的*LOCK *位）级别*1，*此位即无法修改。*

*注：对此位的任何写入操作都需要延迟*1 *个*APB *时钟周期才能生效。*

Bit 27 保留，必须保持为复位值。

Bit 26 **BKDSRM：**刹车解除（Break disarm）

0：刹车输入 tim_brk 已装载

1：刹车输入 tim_brk 被解除

当没有刹车源处于有效状态时，此位由硬件清除。

BKDSRM 位必须由软件设置，以释放双向输出控制（高阻状态下的开漏输出），然后对其进行轮询，直到它被硬件重置，表明故障状态已消失。

注：对此位的任何写入操作都需要延迟 1 *个*APB *时钟周期才能生效。*

Bits 25:20 保留，必须保持为复位值。

Bits 19:16 **BKF[3:0]：**刹车滤波器

这几位定义了 tim_brk 输入的采样频率及数字滤波器长度。数字滤波器由一个事件计数器组成，它记录到 N 个事件后会产生一个输出的跳变

0000：No filter，BRK acts asynchronously

0001：$f_{SAMPLING}=f_{tim_ker_ck}$，N=2。

0010：$f_{SAMPLING}=f_{tim_ker_ck}$，N=4。

0011：$f_{SAMPLING}=f_{tim_ker_ck}$，N=8。

0100：$f_{SAMPLING}=f_{DTS}/2$，N=6。

0101：$f_{SAMPLING}=f_{DTS}/2$，N=8。

0110：$f_{SAMPLING}=f_{DTS}/4$，N=6。

0111：$f_{SAMPLING}=f_{DTS}/4$，N=8。

1000：$f_{SAMPLING}=f_{DTS}/8$，N=6。

1001：$f_{SAMPLING}=f_{DTS}/8$，N=8。

1010：$f_{SAMPLING}=f_{DTS}/16$，N=5。

1011：$f_{SAMPLING}=f_{DTS}/16$，N=6。

1100：$f_{SAMPLING}=f_{DTS}/16$，N=8。

1101：$f_{SAMPLING}=f_{DTS}/32$，N=5。

1110：$f_{SAMPLING}=f_{DTS}/32$，N=6。

1111：$f_{SAMPLING}=f_{DTS}/32$，N=8。

注：只要编程了 LOCK（TIM15_BDTR *寄存器中的* LOCK *位）级别* 1，*此位即无法修改。*

Bit 15 **MOE：**主输出使能（Main output enable）

只要刹车输入变为有效状态，此位便由硬件异步清零。此位由软件置 1，也可根据 AOE 位状态自动置 1。此位仅对配置为输出的通道有效。

0：tim_ocx 和 tim_ocxn 输出禁止或被强制为空闲状态。

1：如果 tim_ocx 和 tim_ocxn 输出的相应使能位（TIM15_CCER 寄存器中的 CCxE 和

CCxNE 位）均置 1，则使能 OC 和 OCN 输出。

有关详细信息，请参见 tim_ocx / tim_ocxn 使能说明（TIM15 捕获/比较使能寄存器(TIM15_CCER)）。

Bit 14 **AOE**：自动输出使能（Automatic output enable）

0：MOE 只能由软件置 1

1：MOE 可由软件置 1，也可在发生下一更新事件时自动置 1(如果刹车输入无效)

*注：只要编程了*LOCK（TIM15_BDTR *寄存器中的* LOCK *位）级别*1，*此位即无法修改。*

Bit 13 **BKP**：刹车极性（Break polarity）

0：刹车输入低电平有效；

1：刹车输入高电平有效。

*注：只要编程了*LOCK（TIM15_BDTR *寄存器中的*LOCK *位）级别*1，*此位即无法修改。*

*注：对此位的任何写入操作都需要延迟*1 *个*APB *时钟周期才能生效。*

Bit 12 **BKE**：刹车使能（Break enable）

0：禁止刹车输入（BRK 及 CCS 时钟失效事件）；

1：开启刹车输入（BRK 及 CCS 时钟失效事件）。

*注：只要编程了*LOCK（TIM15_BDTR *寄存器中的*LOCK *位）级别*1，*此位即无法修改。*

*注：对此位的任何写入操作都需要延迟*1 *个*APB *时钟周期才能生效。*

Bit 11 **OSSR**：运行模式下“关闭状态”选择（Off-state selection for Run mode）

此位在 MOE=1 时作用于配置为输出模式且具有互补输出的通道。如果定时器中没有互补输出，则不存在 OSSR。

有关详细信息，请参见 tim_ocx / tim_ocxn 使能说明（TIM15 捕获/比较使能寄存器(TIM15_CCER)）。

0：处于无效状态时，禁止 OC/OCN 输出（OC/OCN 使能输出信号=0）。

1：处于无效状态时，一旦 CCxE=1 或 CCxNE=1，便使能 OC/OCN 输出并将其设为无效电平。然后设置 OC/OCN 使能输出信号=1

*注：编程了*LOCK（TIM15_BDTR *寄存器中的*LOCK *位）级别*2 *后，此位即无法修改。*

Bit 10 **OSSI**：空闲模式下的关闭状态选择（Off-state selection for Idle mode）

此位在 MOE=0 时作用于配置为输出的通道。

有关详细信息，请参见 tim_ocx/ tim_ocxn 使能说明（TIM15 捕获/比较使能寄存器(TIM15_CCER)）。

0：处于无效状态时，禁止 OC/OCN 输出（OC/OCN 使能输出信号=0）。

1：处于无效状态时，一旦 CCxE=1 或 CCxNE=1，便将 OC/OCN 输出首先强制为其空闲电平。然后设置 OC/OCN 使能输出信号=1

*注：编程了*LOCK（TIM15_BDTR *寄存器中的*LOCK *位）级别*2 *后，此位即无法修改。*

Bits 9:8 **LOCK[1:0]**：锁定设置（Lock configuration）

这些位用于针对软件错误提供写保护。

00：关闭锁定--不对任何位提供写保护。

01：锁定级别 1，此时无法对 TIM15_BDTR 寄存器中的 DTG 位、TIM15_CR2 寄存器中的 OISx 和 OISxN 位以及 TIM15_BDTR 寄存器中的 BKE/BKP/AOE 位执行写操作。

10：锁定级别 2，此时无法对锁定级别 1 中适用的各位、CC 极性位（TIM15_CCER 寄存器中的 CCxP/CCxNP 位，只要通过 CCxS 位将相关通道配置为输出）以及 OSSR 和 OSSI 位执行写操作。

11：锁定级别 3，此时无法对锁定级别 2 中适用的各位、CC 控制位（TIM15_CCMRx 寄存器中的 OCxM 和 OCxPE 位，只要通过 CCxS 位将相关通道配置为输出）执行写操作。

注：复位后只能对 LOCK 位执行一次写操作。对 TIM15_BDTR 寄存器执行写操作后其中的内容将冻结，直到下一次复位。

Bits 7:0 **DTG[7:0]：**配置死区发生器（Dead-time generator setup）

此位域定义插入到互补输出之间的死区持续时间。DT 与该持续时间相对应。

DTG[7:5]=0xx => DT= DTG[7:0] × T_{dtg}，其中 T_{dtg} = T_{DTS}；

DTG[7:5]=10x => DT=（64+DTG[5:0]）× T_{dtg}，其中 T_{dtg} = 2 × T_{DTS}；

DTG[7:5]=110 => DT=（32+DTG[4:0]）× T_{dtg}，其中 T_{dtg} = 8 × T_{DTS}；

DTG[7:5]=111 => DT=（32+DTG[4:0]）× T_{dtg}，其中 T_{dtg} = 16 × T_{DTS}；

例：若 T_{DTS} = 125ns（8MHz），可能的死区时间为：

0 到 15875 ns（步长为 125 ns），

16 us 到 31750 ns（步长为 250 ns），

32 us 到 63us（步长为 1 us），

64 us 到 126 us（步长为 2 us）

*注：编程了*LOCK*（*TIM15_BDTR *寄存器中的*LOCK *位）级别*1、2 *或*3，*此位即无法修改。*

15.6.20 TIM15 定时器死区寄存器 2（TIM15_DTR2）

偏移地址：0x54

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

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

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

Bits 31:18 保留，必须保持为复位值。

Bit 17 **DTPE：**死区时间预加载使能（Deadtime preload enable）

0：死区时间值未预加载

1：使能死区时间值预加载

*注：编程了*LOCK*（*TIM15_BDTR *寄存器中的*LOCK *位）级别*1、2 *或*3，*此位即无法修改。*

Bit 16 **DTAE：**死区时间非对称使能（Deadtime asymmetric enable）

0：上升沿和下降沿的死区时间是相同的，并且使用 DTG[7:0]寄存器定义

1：上升沿的死区时间使用 DTG[7:0]寄存器定义，下降沿的死区时间使用 DTGF[7:0]位定义。

*注：编程了*LOCK*（*TIM15_BDTR *寄存器中的*LOCK *位）级别*1、2 *或*3，*此位即无法修改。*

Bits 15:8 保留，必须保持为复位值。

Bits 7:0 **DTGF[7:0]：**死区下降沿发生器（Dead-time falling edge generator setup）

这个位域定义了在下降沿时，互补输出之间插入死区时间的长度。

DTG[7:5]=0xx => DT= DTG[7:0] × T_{dtg}，其中 T_{dtg} = T_{DTS}；

DTG[7:5]=10x => DT=（64+DTG[5:0]）× T_{dtg}，其中 T_{dtg} = 2 × T_{DTS}；

DTG[7:5]=110 => DT=（32+DTG[4:0]）× T_{dtg}，其中 T_{dtg} = 8 × T_{DTS}；
DTG[7:5]=111 => DT=（32+DTG[4:0]）× T_{dtg}，其中 T_{dtg} = 16 × T_{DTS}；
例：若 T_{DTS} = 125ns（8MHz），可能的死区时间为：
0 到 15875 ns（步长为 125 ns），
16 us 到 31750 ns（步长为 250 ns），
32 us 到 63us（步长为 1 us），
64 us 到 126 us（步长为 2 us）

*注：编程了*LOCK（TIM15_BDTR *寄存器中的*LOCK *位）级别*1、2 *或*3，*此位即无法修改。*

15.6.21 TIM15 定时器输入选择寄存器（TIM15_TISEL）

偏移地址：0x5C
复位值：0x0000
访问：无等待，支持字，半字和字节访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														TI1SEL[3:0]	
														rw	rw

Bits 31:2 保留，必须保持为复位值。

Bits 1:0 **TI1SEL[1:0]：选择 tim_ti1_in[0..1]输入**（Selects tim_ti1_in[0..1] input）
00：TIM15_CH1 input (tim_ti1_in0)
01：tim_ti1_in1
10：tim_ti1_in2
11：tim_ti1_in3
参考 TIM15 引脚和内部信号互连列表。

15.6.22 TIM15 复用功能选择寄存器 1（TIM15_AF1）

偏移地址：0x60
复位值：0x0000 0001
访问：无等待，支持字，半字和字节访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				BK CMP2P	BK CMP1P	BKINP	Reserved						BK CMP2E	BK CMP1E	BKINE
				rw	rw	rw							rw	rw	rw

Bits 31:12 保留，必须保持为复位值。

Bit 11 **BKCMP2P：**tim_brk_cmp2 输入极性（tim_brk_cmp2 input polarity）
此位选择 tim_brk_cmp2 输入敏感度。它必须与 BKP 极性位一起编程。
0：tim_brk_cmp2 输入高电平有效
1：tim_brk_cmp2 输入低电平有效

*注：只要编程了*LOCK（TIM15_BDTR *寄存器中的*LOCK *位）级别*1，*此位即无法修改。*

Bit 10 **BKCMP1P：**tim_brk_cmp1 输入极性（tim_brk_cmp1 input polarity）

此位选择 tim_brk_cmp1 输入敏感度。它必须与 BKP 极性位一起编程。
0：tim_brk_cmp1 输入高电平有效
1：tim_brk_cmp1 输入低电平有效
*注：只要编程了*LOCK *（*TIM15_BDTR *寄存器中的*LOCK *位）级别*1*，此位即无法修改。*

Bit 9 **BKINP**：TIM15_BKIN 输入极性（TIM15_BKIN input polarity）
该位选择 TIM15_BKIN 备用功能输入敏感度。必须对其进行编程
0：TIM15_BKIN 输入电平有效
1：TIM15_BKIN 输入低电平有效
*注：只要编程了*LOCK *（*TIM15_BDTR *寄存器中的*LOCK *位）级别*1*，此位即无法修改。*

Bits 8:3 保留，必须保持为复位值。

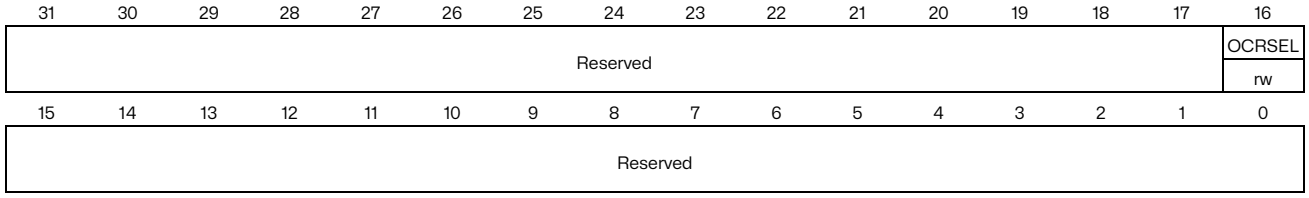
Bit 2 **BKCMP2E**：tim_brk_cmp2 使能（tim_brk_cmp2 enable）
该位用于使能或禁用 tim_brk_cmp2 输入。如果该位设置为 1，则 tim_brk_cmp2 输入将被使能，否则将被禁用。
0：tim_brk_cmp2 输入禁用
1：tim_brk_cmp2 输入使能
*注：只要编程了*LOCK *（*TIM15_BDTR *寄存器中的*LOCK *位）级别*1*，此位即无法修改。*

Bit 1 **BKCMP1E**：tim_brk_cmp1 使能（tim_brk_cmp1 enable）
该位用于使能或禁用 tim_brk_cmp1 输入。如果该位设置为 1，则 tim_brk_cmp1 输入将被使能，否则将被禁用。
0：tim_brk_cmp1 输入禁用
1：tim_brk_cmp1 输入使能
*注：只要编程了*LOCK *（*TIM15_BDTR *寄存器中的*LOCK *位）级别*1*，此位即无法修改。*

Bit 0 **BKINE**：TIM15_BKIN 输入使能（TIM15_BKIN input enable）
该位用于使能或禁用 TIM15_BKIN 复用功能输入作为定时器的 tim_brk 输入。
TIM15_BKIN 输入与其他 tim_brk 源进行“或”操作。
0：TIM15_BKIN 输入禁用
1：TIM15_BKIN 输入使能
*注：只要编程了*LOCK *（*TIM15_BDTR *寄存器中的*LOCK *位）级别*1*，此位即无法修改。*

15.6.23 TIM15 复用功能选择寄存器 2（TIM15_AF2）

偏移地址：0x64
复位值：0x0000 0001
访问：无等待，支持字，半字和字节访问



Bits 31:17 保留，必须保持为复位值。

Bit 16 **OCRSEL**：ocref_clr 的源选择（ocref_clr source selection）
这些位选择 ocref_clr 的输入源
0：tim_ocref_clr0

1：tim_ocref_clr1

参考 TIM15 引脚和内部信号用于特定功能实现。

*注：只要编程了*LOCK（TIM15_BDTR *寄存器中的*LOCK *位）级别*1，*此位即无法修改。*

Bits 15:0 保留，必须保持复位值。

15.6.24 TIM15 触发寄存器（TIM15_TRGI）

偏移地址：0x80

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

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

Bits 15:1 保留，必须保持为复位值。

Bit 0 **SLAVE_SYNC：**同步模式，从机同步配置。

当从模式选择为触发模式时：

0：从机同步模式禁止

1：从机同步模式开启

16 基本定时器（TIM6/TIM7）

16.1 TIM6/TIM7 简介

基本定时器 TIM6 和 TIM7 包含一个 16 位自动重载计数器，该计数器由可编程预分频器驱动。他们可以用作通用定时器以生成时基。这些定时器彼此完全独立，不共享任何资源。

16.2 TIM6/TIM7 主要功能

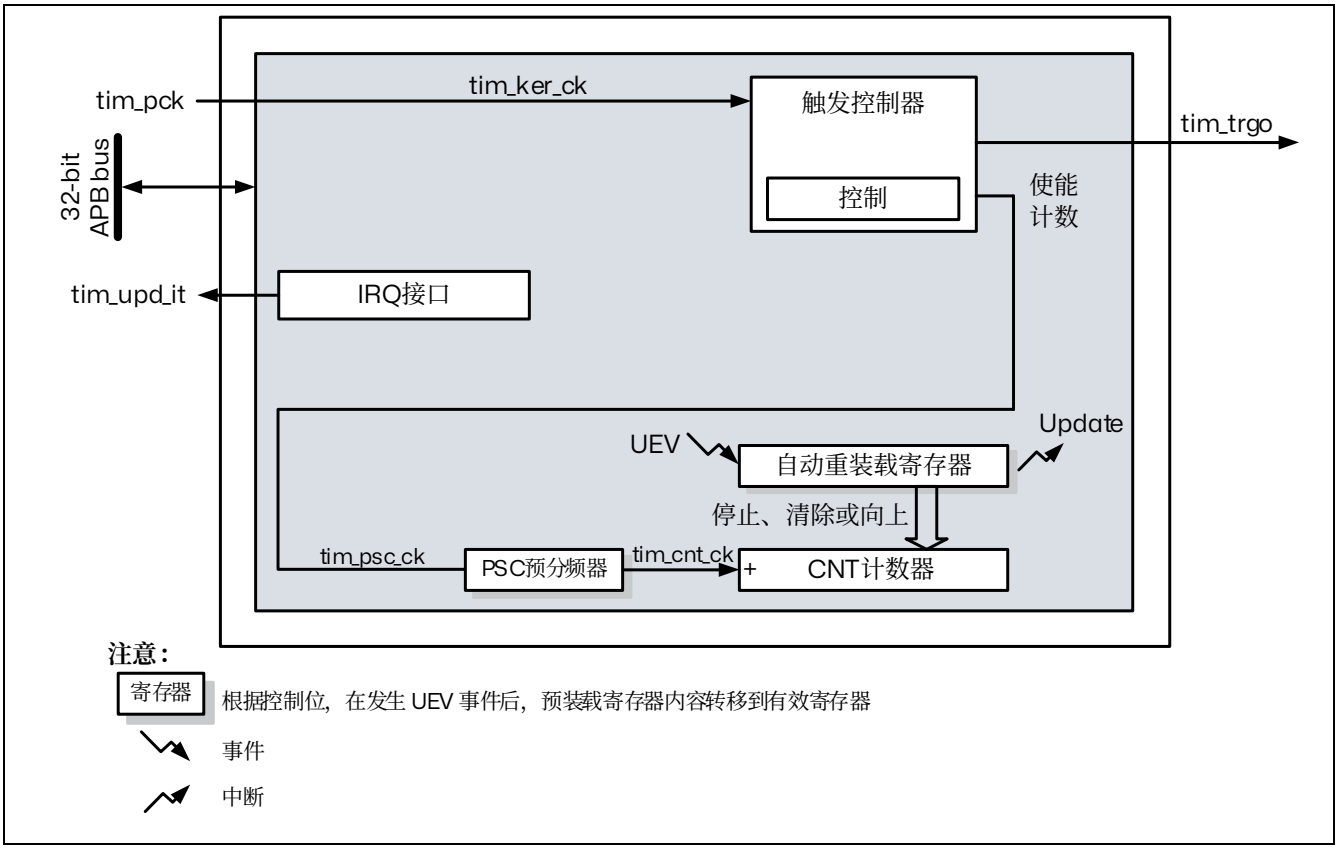
基本定时器（TIM6/TIM7）功能包括：

- 16 位自动重载递增计数器
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（即运行时修改），分频系数介于 1 和 65536 之间
- 发生如下更新事件时会生成中断：计数器上溢

16.3 TIM6/TIM7 功能描述

16.3.1 框图

图 16.1 基本定时器框图



16.3.2 TIM6/TIM7 内部信号

本节中的下表格总结了 TIM 的输入和输出。

表 16.1 TIM 内部输入/输出信号

内部信号名称	信号类型	描述
tim_pclk	输入	定时器 APB 时钟
tim_ker_ck	输入	定时器内核时钟。此时钟必须与 tim_plck 同步（来自同一源）。 时钟比率 tim_ker_ck/ tim_pclk 必须为整数： 1，2，3，...，16（最大值）
tim_trgo	输出	内部触发器输出。此触发器可以触发其他片上外设
tim_upd_it	输出	定时器更新事件中断

16.3.3 TIM6/TIM7 时钟

定时器总线接口由 tim_pclk APB 时钟驱动。

计数器时钟 tim_ker_ck 连接到 tim_pclk 输入。

CEN 位（TIMx_CR1 寄存器中）和 UG 位（TIMx_EGR 寄存器中）为实际控制位，并且只能通过软件进行更改（UG 除外，仍自动清零）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 tim_ker_ck 提供。

下图显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）。

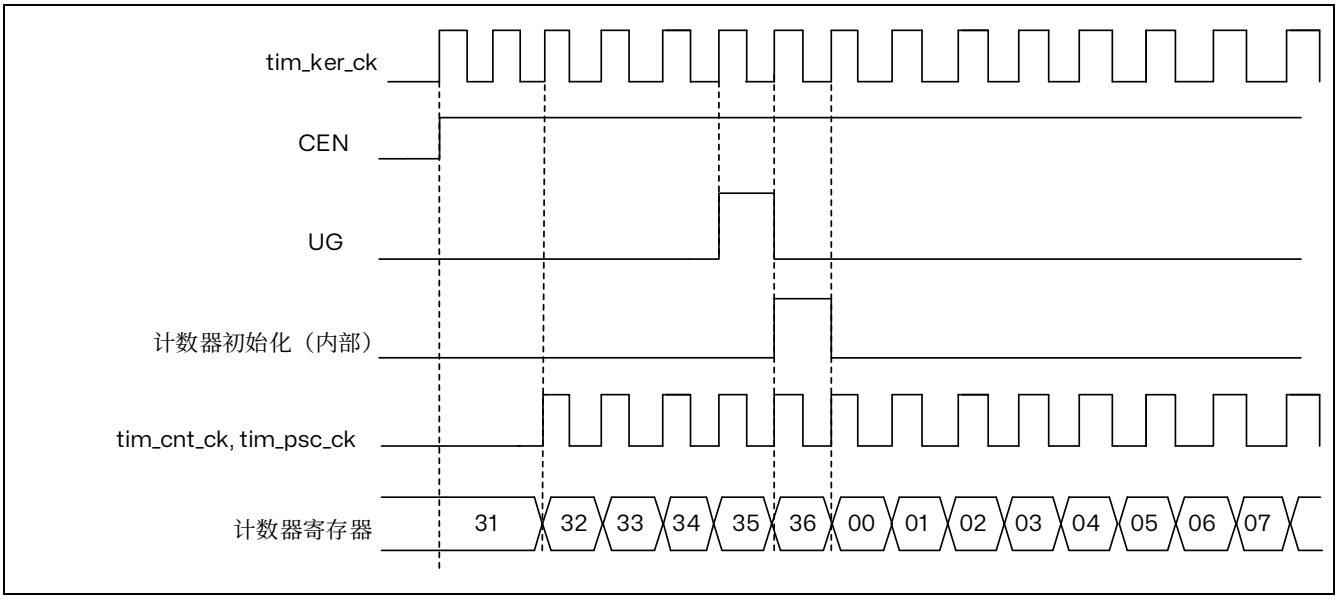


图 16.2 一般模式下的控制电路，内部时钟分频因子为 1

16.3.4 时基单元

可编程定时器的主要模块由一个 16 位递增计数器及其相关的自动重载寄存器组成。计数器的时钟可通过预分频器进行分频。

计数器、自动重载寄存器和预分频器寄存器可通过软件进行读写。即使在计数器运行时也可执行读写操作。

时基单元包含：

- 计数器寄存器（TIMx_CNT）
- 预分频器寄存器（TIMx_PSC）
- 自动装载寄存器（TIMx_ARR）

自动重载寄存器是预装载的。每次尝试对自动重载寄存器执行读写操作时，都会访问预装载寄存器。预装载寄存器的内容既可以直接传送到影子寄存器，也可以在每次发生更新事件 UEV 时传送到影子寄存器，这取决于 TIMx_CR1 寄存器中的自动重载预装载使能位(ARPE)。当计数器达到上溢值并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生进行详细介绍。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位（CEN）置 1 时，才会启动计数器。

请注意，实际的计数器使能信号 CNT_EN 在 CEN 置 1 的一个时钟周期后被置 1。

预分频器描述

预分频器可对计数器时钟频率进行分频，分频系数介于 1 和 65536 之间。该预分频器基于 TIMx_PSC 寄存器中的 16 位寄存器所控制的 16 位计数器。由于 TIMx_PSC 控制寄存器有缓冲，因此可对预分频器进行实时更改。而新的预分频比将在下一更新事件发生时被采用。

下图给出了在预分频比发生实时变化时一些计数器行为的示例。

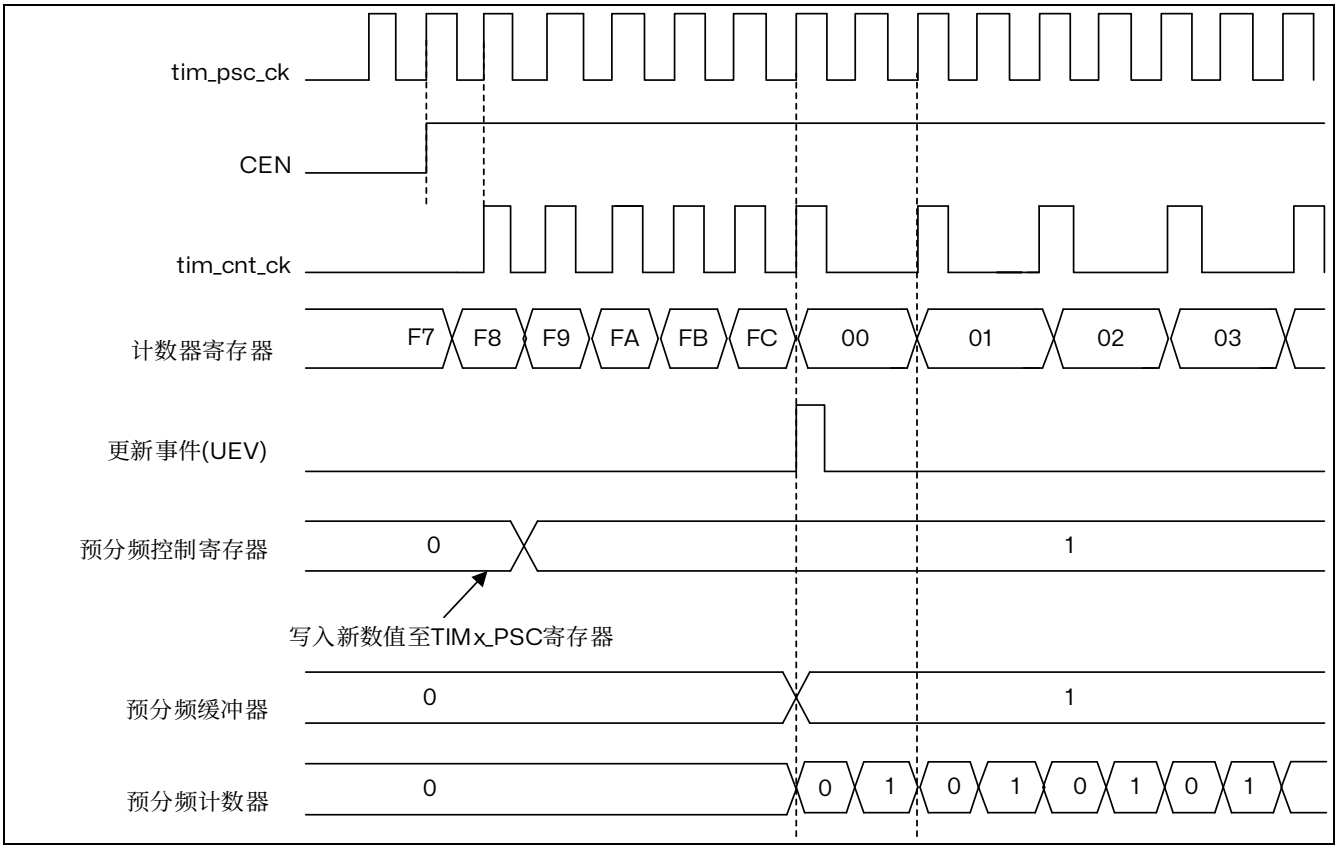


图 16.3 当预分频器的参数从 1 变到 2 时，计数器的时序图

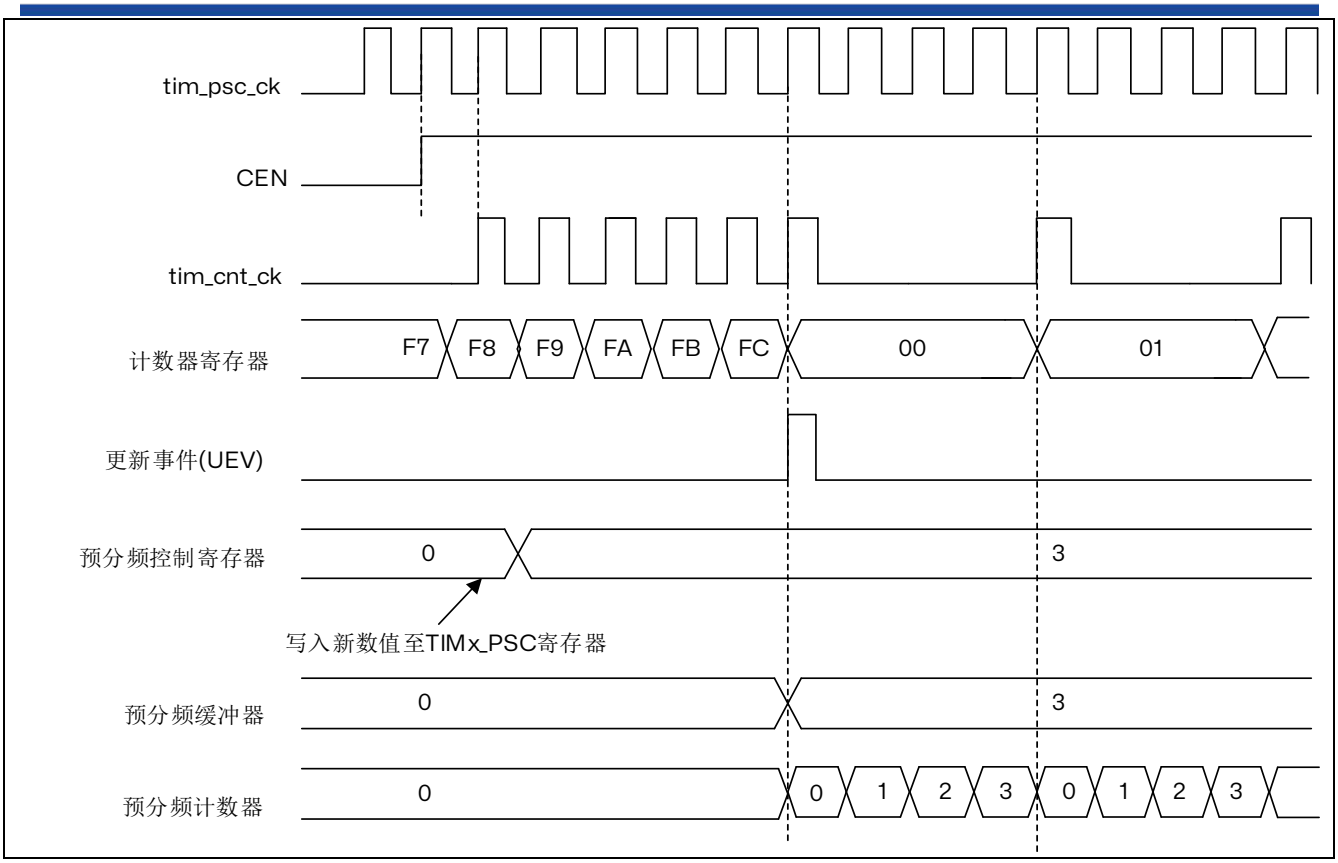


图 16.4 当预分频器的参数从 1 变到 4 时，计数器的时序图

16.3.5 计数器模式

计数器从 0 计数到自动重载值（TIMx_ARR 寄存器的内容），然后重新从 0 开始计数并生成计数器上溢事件。

每次发生计数器上溢时会生成更新事件，或将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。这样，直到 UDIS 位中写入 0 前便不会生成任何更新事件，但计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断请求）。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 使用预装载值（TIMx_PSC 寄存器的内容）重新装载预分频器的缓冲区
- 使用预装载值（TIMx_ARR）更新自动重载影子寄存器

以下各图显示了当 TIMx_ARR=0x36 时不同时钟频率下计数器行为的示例。

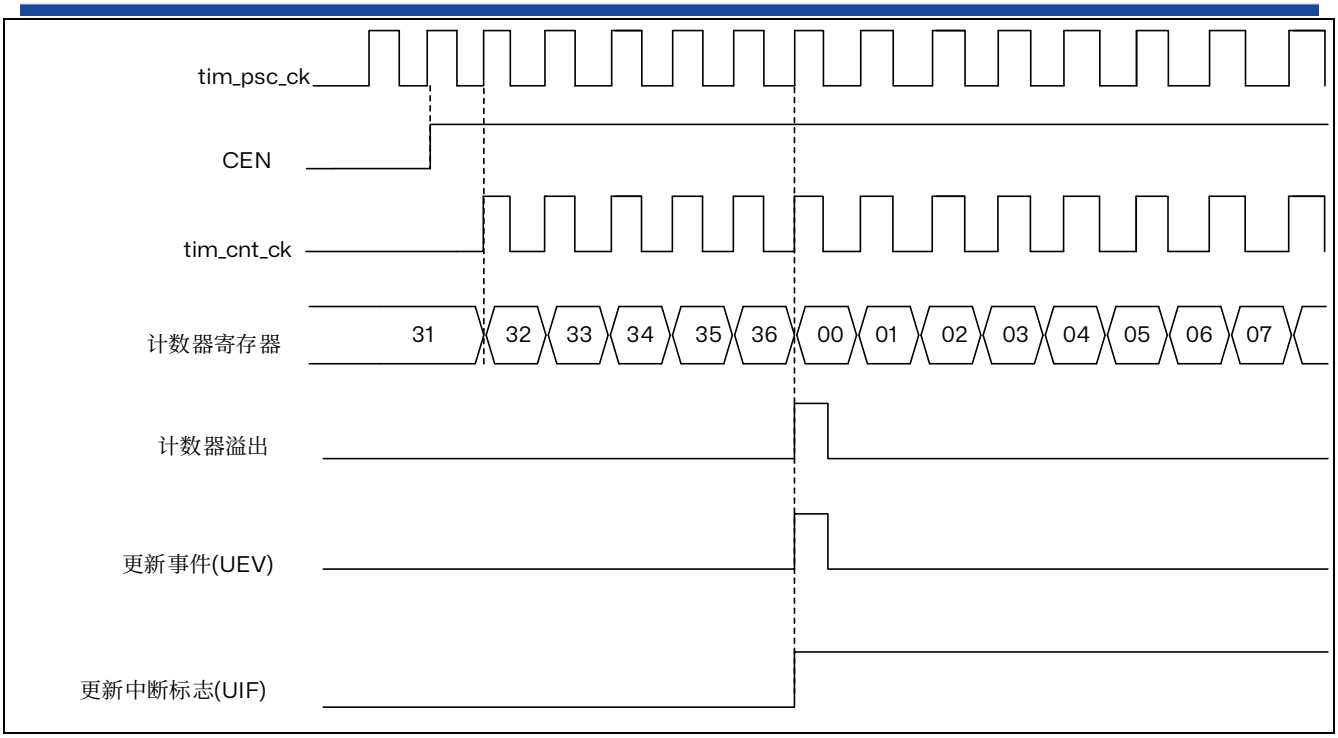


图 16.5 计数器时序图，内部时钟分频因子为 1

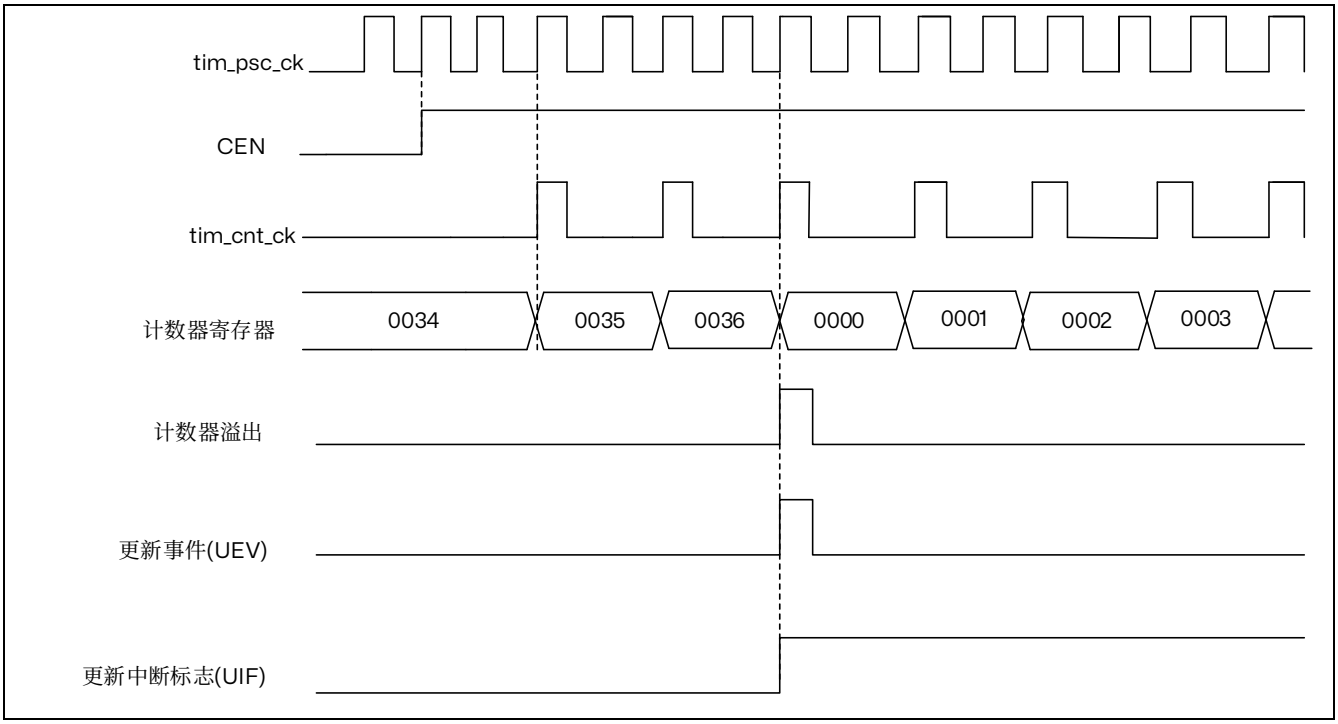


图 16.6 计数器时序图，内部时钟分频因子为 2

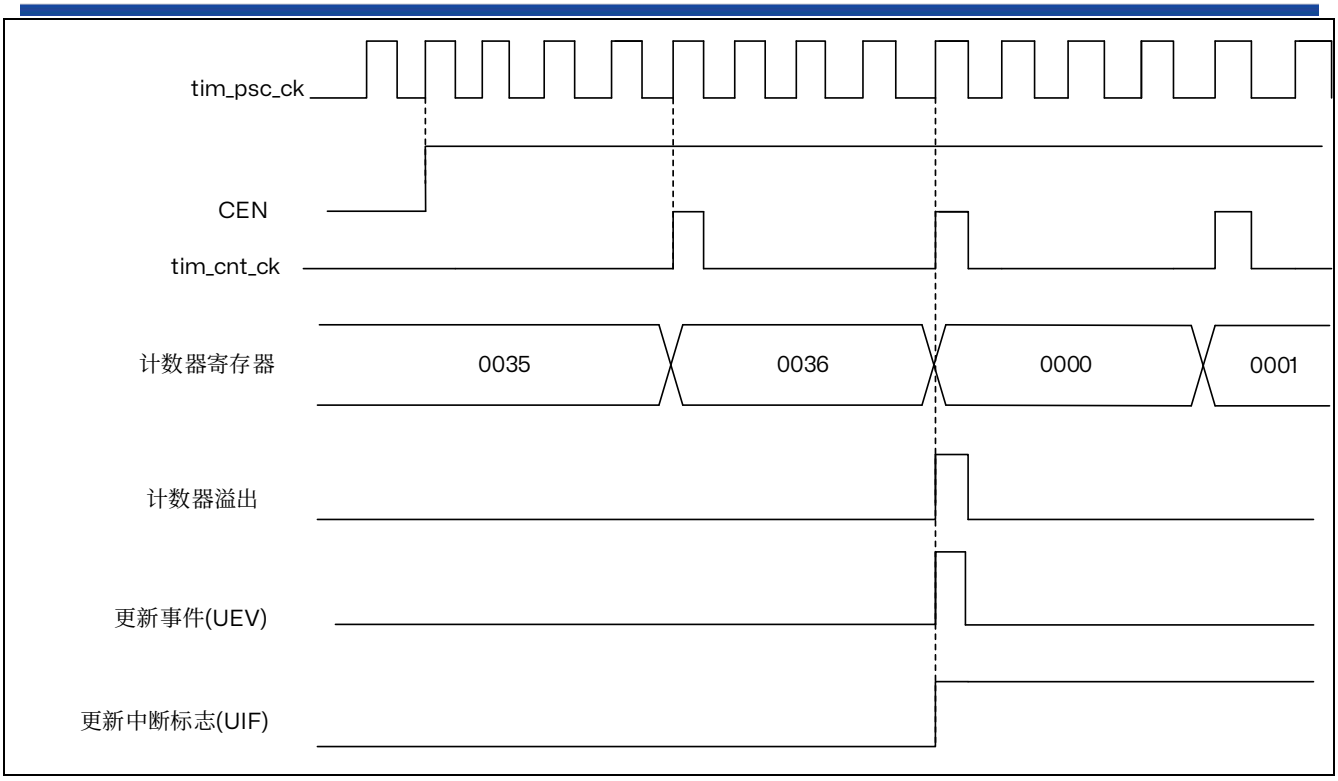


图 16.7 计数器时序图，内部时钟分频因子为 4

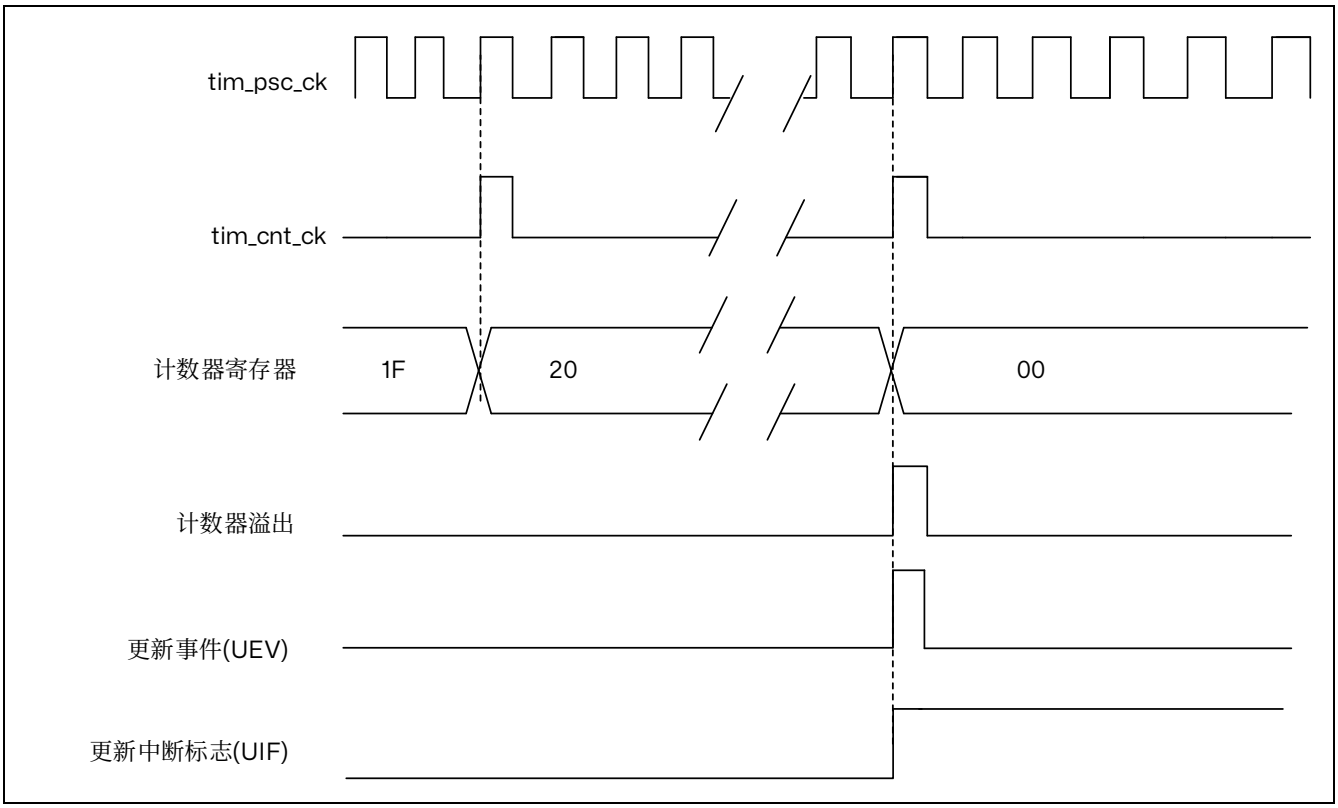


图 16.8 计数器时序图，内部时钟分频因子为 N

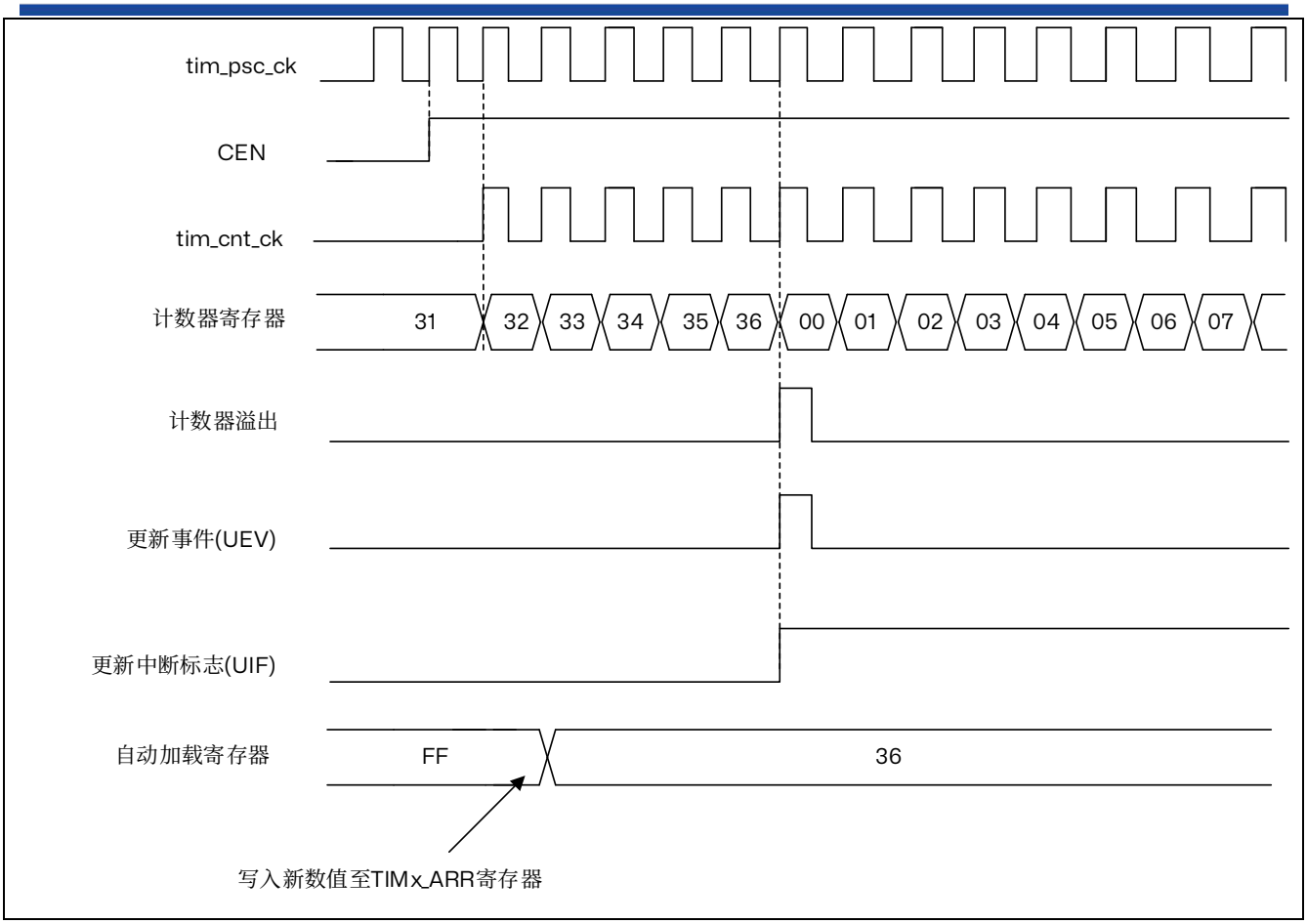


图 16.9 计数器时序图，当 ARPE=0 时的更新事件（TIMx_ARR 没有预装入）

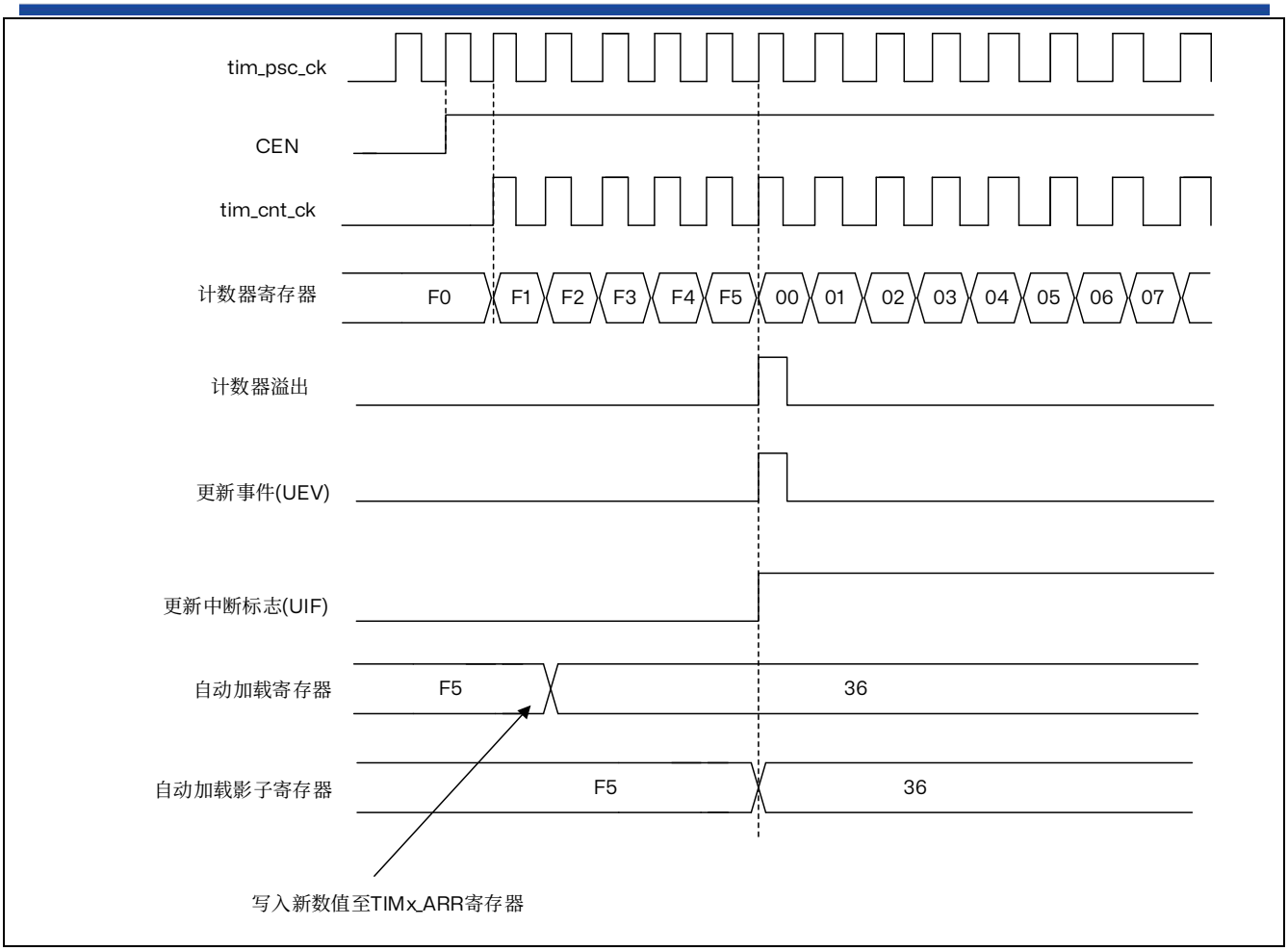


图 16.10 计数器时序图，当 ARPE=1 时的更新事件（预装入了 TIMx_ARR）

16.3.6 UIF 位重映射

TIMx_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志 UIF 连续复制到定时器计数器寄存器的第 31 位（TIMxCNT[31]）。这允许以原子方式读取计数器的值和一个潜在的由 UIFCPY 标志指示的回滚条件。在特定情况下，它可以避免由背景任务（读取计数器）和中断（更新中断）之间的处理引起的竞争条件，从而简化计算。

UIF 和 UIFCPY 标志之间没有延迟。

16.3.7 ADC 触发

定时器可以使用各种内部信号（例如重置、使能或比较事件）生成 ADC 触发事件。

注意：接收 tim_trgo *信号的从属外设（定时器、ADC 等）的时钟必须在从主定时器接收事件之前使能，并且时钟频率（预分频器）在从主定时器接收触发器时不能更改。*

16.3.8 调试模式

当微控制器进入调试模式时（内核停止），TIMx 计数器会根据 DBGMCU 模块中的 DBG_TIMx_STOP 配置位选择继续正常工作或者停止工作。

有关详细信息，请参阅 section Debug support（DBG）。

16.4 TIM6/TIM7 低功耗模式

表 16.2 低功耗模式对 TIM6/TIM7 的影响

模式	描述
睡眠	无影响，外设是运行的。中断会导致设备退出睡眠模式。
停机	定时器操作被停止并保留其寄存器内容。无法生成中断。
待机	定时器被断电并且退出待机模式后必须重新初始化。

16.5 TIM6/TIM7 中断

TIM6/TIM7 可以生成一个单一中断，如下表所示。

表 16.3 中断请求

中断缩写	中断事件	事件标志	使能控制位	中断清除方法	从睡眠模式退出	从停机和待机退出
TIM6 TIM7	更新	UIF	UIE	UIF 写 0	是	否

16.6 TIMx 寄存器

16.6.1 TIMx 控制寄存器 1（TIMx_CR1）（x=6/7）

偏移地址：0x00

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				UIFREMAP	Reserved			ARPE	Reserved			OPM	URS	UDIS	CEN
				rw				rw				rw	rw	rw	rw

Bits 15:12 保留，必须保持复位值。

Bit 11 **UIFREMAP**：UIF 状态位重映射（UIF status bit remapping）

0：无重映射。UIF 状态位未复制到 TIMx_CNT 寄存器位 31

1：重映射使能。UIF 状态位复制到 TIMx_CNT 寄存器位 31

Bits 10:8 保留，必须保持复位值。

Bit 7 **ARPE**：自动重载预装载使能（Auto-reload preload enable）

0：TIMx_ARR 寄存器不进行缓冲；

1：TIMx_ARR 寄存器进行缓冲。

Bits 6:4 保留，必须保持复位值。

Bit 3 **OPM**：单脉冲模式（One pulse mode）

0：计数器在发生更新事件时不会停止计数；

1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）

Bit 2 **URS**：更新请求源（Update request source）

此位由软件置 1 和清零，用以选择 UEV 事件源。

0：使能时，所有以下事件都会生成更新中断。此类事件包括：

- 计数器上溢/下溢
- 将 UG 位置 1
- 通过从模式控制器生成的更新事件

1：使能时，只有计数器上溢/下溢会生成更新中断。

Bit 1 **UDIS**：更新禁止（Update disable）

此位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。

0：使能 UEV。更新（UEV）事件可通过以下事件之一生成：

- 计数器上出/下溢
- 将 UG 位置 1
- 通过从模式控制器生成的更新事件

然后更新影子寄存器的值。

1：禁止 UEV。不会生成更新事件，各影子寄存器的值（ARR、PSC）保持不变。但如果将 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。

Bit 0 **CEN**：计数器使能（Counter enable）

0：禁止计数器；

1：使能计数器。

单脉冲模式下，当发生更新事件时会自动将 CEN 位清零。

16.6.2 TIMx 控制寄存器 2（TIMx_CR2）（x=6/7）

偏移地址：0x04

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

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

Bits 15:7 保留，必须保持复位值。

Bits 6:4 **MMS[2:0]：**主模式选择（Master mode selection）

这些位可选择主模式下将要发送到从定时器以实现同步的信息(tim_trgo)。这些位的组合如下：

000：**复位** – TIMx_EGR 寄存器中的 UG 位用作触发输出（tim_trgo）。

001：**使能** – 计数器使能信号 CNT_EN 用作触发输出（tim_trgo）。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号可由 CEN 控制位产生。

010：**更新** – 选择更新事件作为触发输出（tim_trgo）。例如，主定时器可用作从定时器的预分频器。

注：从定时器或接收 tim_trgo *的外设时钟必须在从主定时器接收事件之前使能，在主定时器接收触发期间不能实时更改。*

Bit 3:0 保留，必须保持复位值。

16.6.3 TIMx 中断使能寄存器（TIMx_DIER）（x=6/7）

偏移地址：0x0C

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

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

Bits 15:1 保留，必须保持为复位值。

Bit 0 **UIE**：更新中断使能（Update interrupt enable）

0：禁止更新中断；

1：使能更新中断。

16.6.4 TIMx 状态寄存器（TIMx_SR）（x=6/7）

偏移地址：0x10

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															UIF
															rc w0

Bits 15:1 保留，必须保持为复位值。

Bit 0 **UIF**：更新中断标志（Update interrupt flag）

该位在发生更新事件时通过硬件置 1。但需要通过软件清零。

0：未发生更新。

1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：

– TIMx_CR1 寄存器中的 UDIS=0，并且重复计数器值上溢或下溢时（重复计数器= 0 时更新）。

– TIMx_CR1 寄存器中的 URS=0 且 UDIS=0，并且由软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。

16.6.5 TIMx 事件产生寄存器（TIMx_EGR）（x=6/7）

偏移地址：0x14

复位：0x0000 0000

访问：无等待，支持字，半字和字节访问

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

Bits 15:1 保留，必须保持复位值。

Bit 0 **UG**：更新生成（Update generation）

该位可通过软件置 1，并由硬件自动清零。

0：不执行任何操作；

1：重新初始化计数器并生成一个寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。

16.6.6 TIMx 计数器（TIMx_CNT）（x=6/7）

偏移地址：0x24

复位值：0x0000

访问：无等待，支持字，半字和字节访问

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

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

Bit 31 **UIFCPY**：复制 UIF

该位是 TIMx_SR 寄存器中 UIF 位的一个只读副本。如果 TIMx_CR1 中的 UIFREMAP 位被复位，那位 31 被保留且读为 0

Bits 30:16 保留，必须保持为复位值。

Bits 15:0 **CNT[15:0]**：计数器的值（Counter value）

该寄存器（CNT[15:0]）保存的是计数器的值。

16.6.7 TIMx 预分频器（TIMx_PSC）（x=6/7）

偏移地址：0x28

复位值：0x0000

访问：无等待，支持字，半字和字节访问

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

Bits 15:0 **PSC[15:0]：**预分频器的值（Prescaler value）

计数器时钟频率（$f_{tim_cnt_ck}$）等于 $f_{tim_psc_ck}$ / (PSC[15:0] + 1)。

PSC 包含每次发生更新事件（包括计数器通过 TIMx_EGR 寄存器中的 UG 位清零）时要装载到活动预分频器寄存器的值。

16.6.8 TIMx 自动重装载寄存器（TIMx_ARR）（x=6/7）

偏移地址：0x2C

复位：0x0000 FFFF

访问：无等待，支持字，半字和字节访问

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

Bits 31:16 保留，必须保持为复位值。

Bits 19:0 **ARR[15:0]：**自动重装载的值（Auto-reload value）

ARR 为要装载到实际自动重载寄存器的值。

当自动重载值为空时，计数器不工作。

该寄存器存储的是自动重新加载的值

17 运算放大器（OPAMP）

17.1 简介

RX32S632 内嵌 3 个运算放大器（OPAMP1 和 OPAMP2 仅有 PGA 功能，OPAMP3 有 PGA 和 OPA 功能）。当 OPAMP 禁用时，输出为高阻抗。

17.2 运算放大器特征

运算放大器具有以下特征：

- 轨对轨输入/输出
- PGA 模式可内部接地
- PGA 模式（4~16 倍）
- 电压跟随器模式
- 独立模式（仅 OPAMP3）
- PGA 模式内部偏置电压可选（GND、VBG/2 或 VREFBUF/2）
- PGA 模式输出可内部连接 ADC 和 CMP

17.3 OPAMP 模式

17.3.1 独立模式（仅 OPAMP3）：

配置步骤：

1. 使能 OPAMP3 输出至外部 IO 引脚
   - 将 OPAMP3_CSR 寄存器的 TOEXT_EN 位置“1”
2. OPAMP3 P 端选择
   - 选 VIN P2：将 OPAMP3_CSR 寄存器的 VP_SEL 位置“1”
3. OPAMP3 N 端选择
   - 选择 VIN N：将 OPAMP3_CSR 寄存器的 VM_SEL[1:0]位设置为“11”
4. 使能 OPAMP3
   - 将 OPAMPx_CSR 寄存器的 OPAEN 位置“1”

注：1、放大倍数 = (R1+R2)/R2

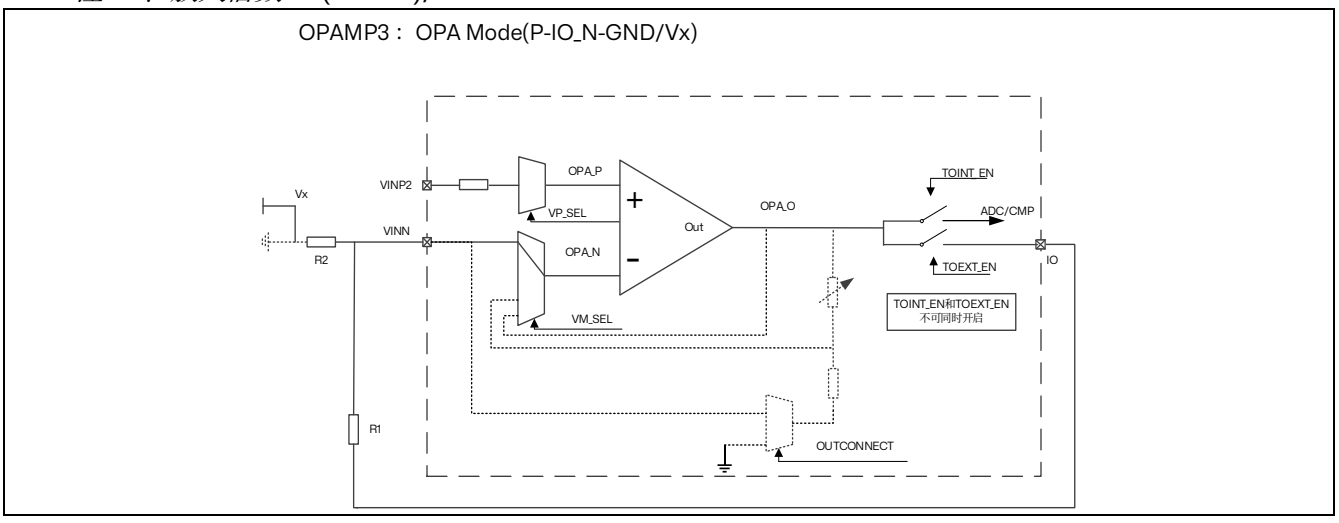


图 17.1 OPAMP3 独立模式

17.3.2 电压跟随器模式

配置步骤：

1. 将 OPAMPx 配置成电压跟随器模式
   - OPAMPx_CSR 寄存器的 VM_SEL[1:0] = 10
2. OPAMPx P 端选择（仅 OPAMP3）
   - 选 VIN P2：OPAMPx_CSR 寄存器的 VP_SEL 位置“1”
3. 使能 OPAMPx
   - OPAMPx_CSR 寄存器的 OPAEN 位置“1”

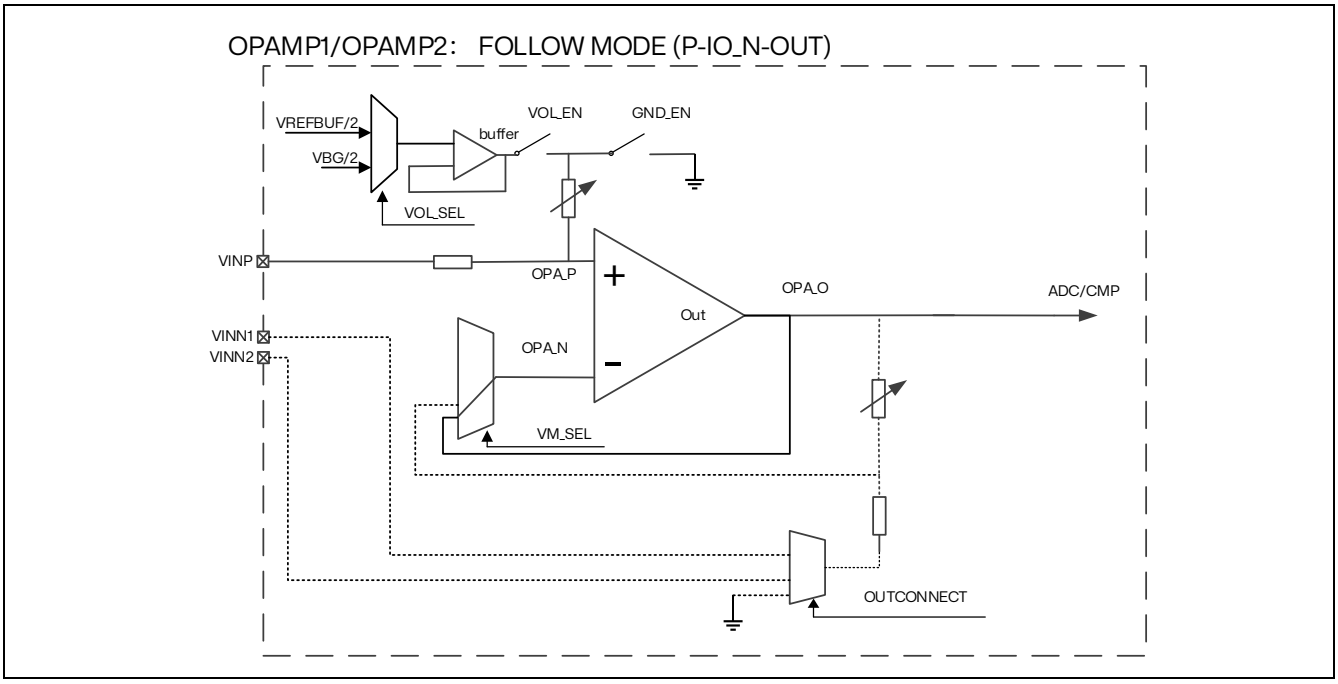


图 17.2 OPAMP1/OPAMP2 电压跟随器模式

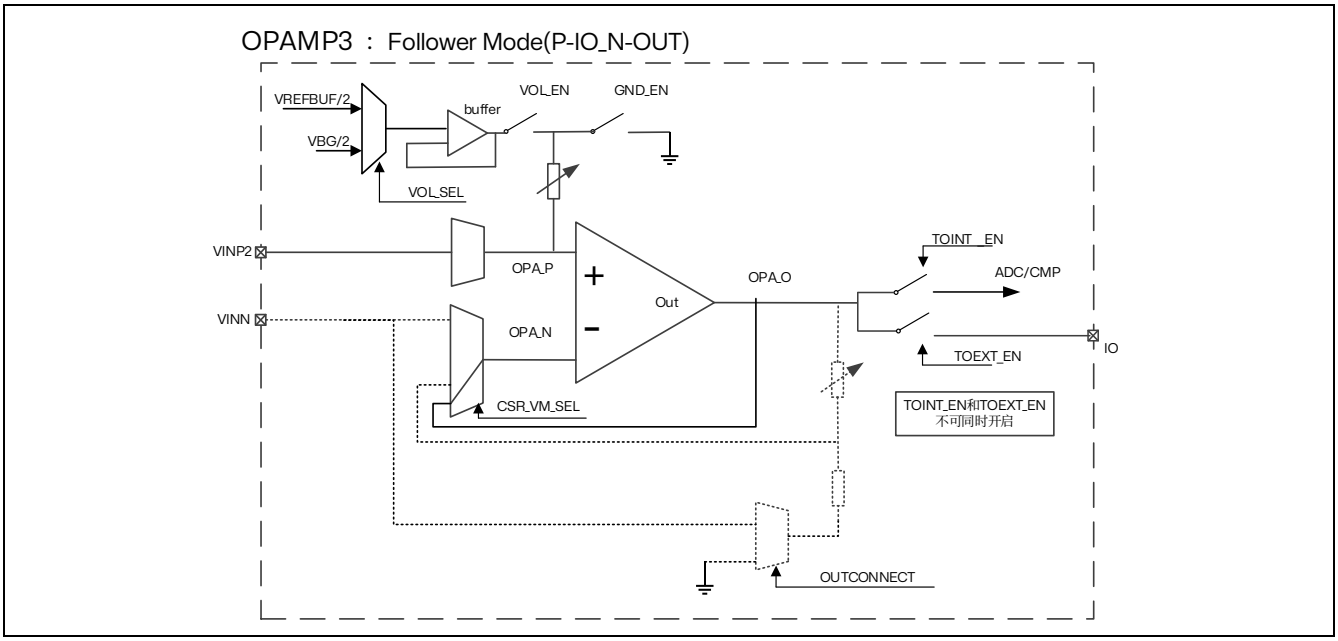


图 17.3 OPAMP3 电压跟随器模式

17.3.3 PGA 模式（P-IO，N-IO）

1. 将 OPAMPx 配置为 PGA 模式
   - OPAMPx_CSR 寄存器的 VM_SEL[1:0] = 01
2. OPAMPx P 端选择（仅 OPAMP3）
   - 选 VIN P2：将 OPAMPx_CSR 寄存器的 VP_SEL 位置“1”
3. OPAMPx N 端选择
   - 选择 VIN N1：OPAMPx_CSR 寄存器的 OUTCONNECT[1:0] = 10
   - 选择 VIN N2（仅 OPAMP1/2）：OPAMPx_CSR 寄存器的 OUTCONNECT[1:0] = 11
4. 选择 P 端输入外接偏置 GND
   - OPAMPx_BIAS 寄存器的 GND_EN 位置“1”
5. 放大倍数选择
   - 4X：OPAMPx_CSR 寄存器的 GAIN[1:0] = 00
   - 8X：OPAMPx_CSR 寄存器的 GAIN[1:0] = 01
   - 12X：OPAMPx_CSR 寄存器的 GAIN[1:0] = 10
   - 16X：OPAMPx_CSR 寄存器的 GAIN[1:0] = 11
6. 使能 OPAMPx
   - 将 OPAMPx_CSR 寄存器的 OPAEN 位置“1”

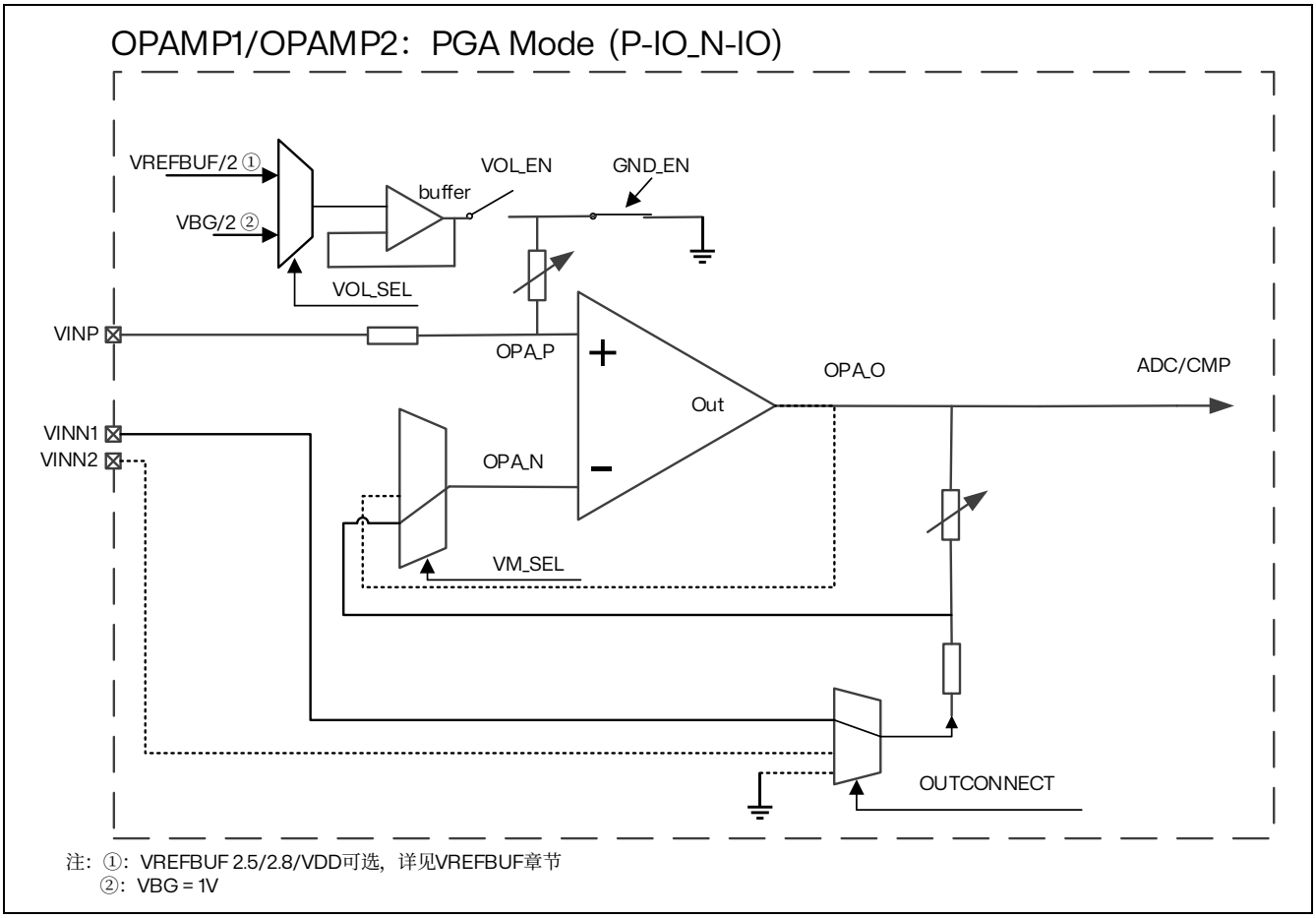


图 17.4 OPAMP1/OPAMP2 PGA 模式（P-IO，N-IO）

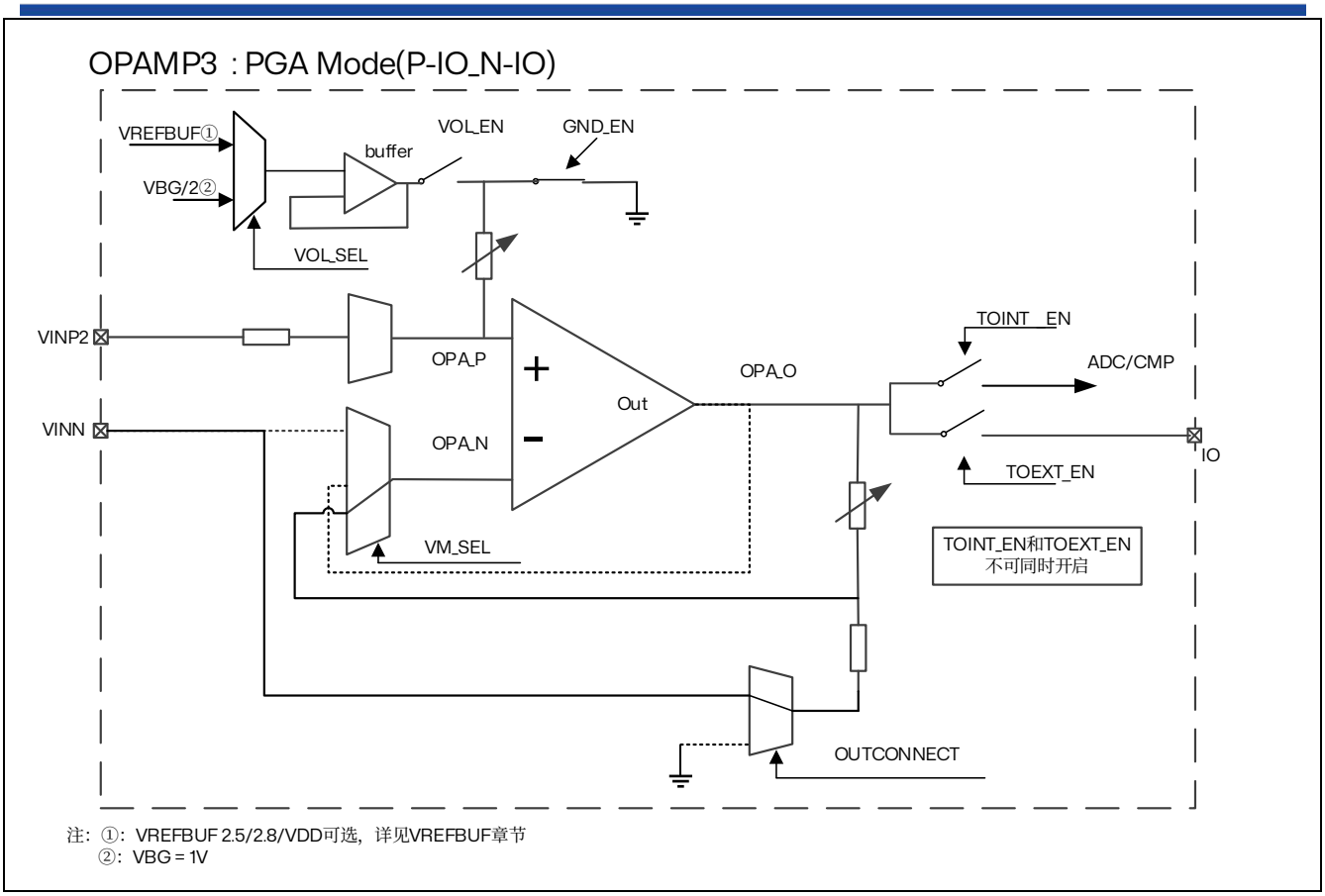


图 17.5 OPAMP3 PGA 模式（P-IO，N-IO）

17.3.4 PGA 模式（P-IO，N-GND）

1. 将 OPAMPx 配置为 PGA 模式
   - 将 OPAMPx_CSR 寄存器的 VM_SEL[1:0] = 01
2. OPAMPx P 端选择（仅 OPAMP3）
   - 选 VIN P2：将 OPAMPx_CSR 寄存器的 VP_SEL 位置“1”
3. OPAMPx N 端内部接地
   - 将 OPAMPx_CSR 寄存器的 OUTCONNECT[1:0] = 01
4. 选择 P 端输入外接偏置 GND
   - OPAMPx_BIAS 寄存器的 GND_EN 位置“1”
5. 放大倍数选择
   - 4X：OPAMPx_CSR 寄存器的 GAIN[1:0] = 00
   - 8X：OPAMPx_CSR 寄存器的 GAIN[1:0] = 01
   - 12X：OPAMPx_CSR 寄存器的 GAIN[1:0] = 10
   - 16X：OPAMPx_CSR 寄存器的 GAIN[1:0] = 11
6. 使能 OPAMPx
   - 将 OPAMPx_CSR 寄存器的 OPAEN 位置“1”

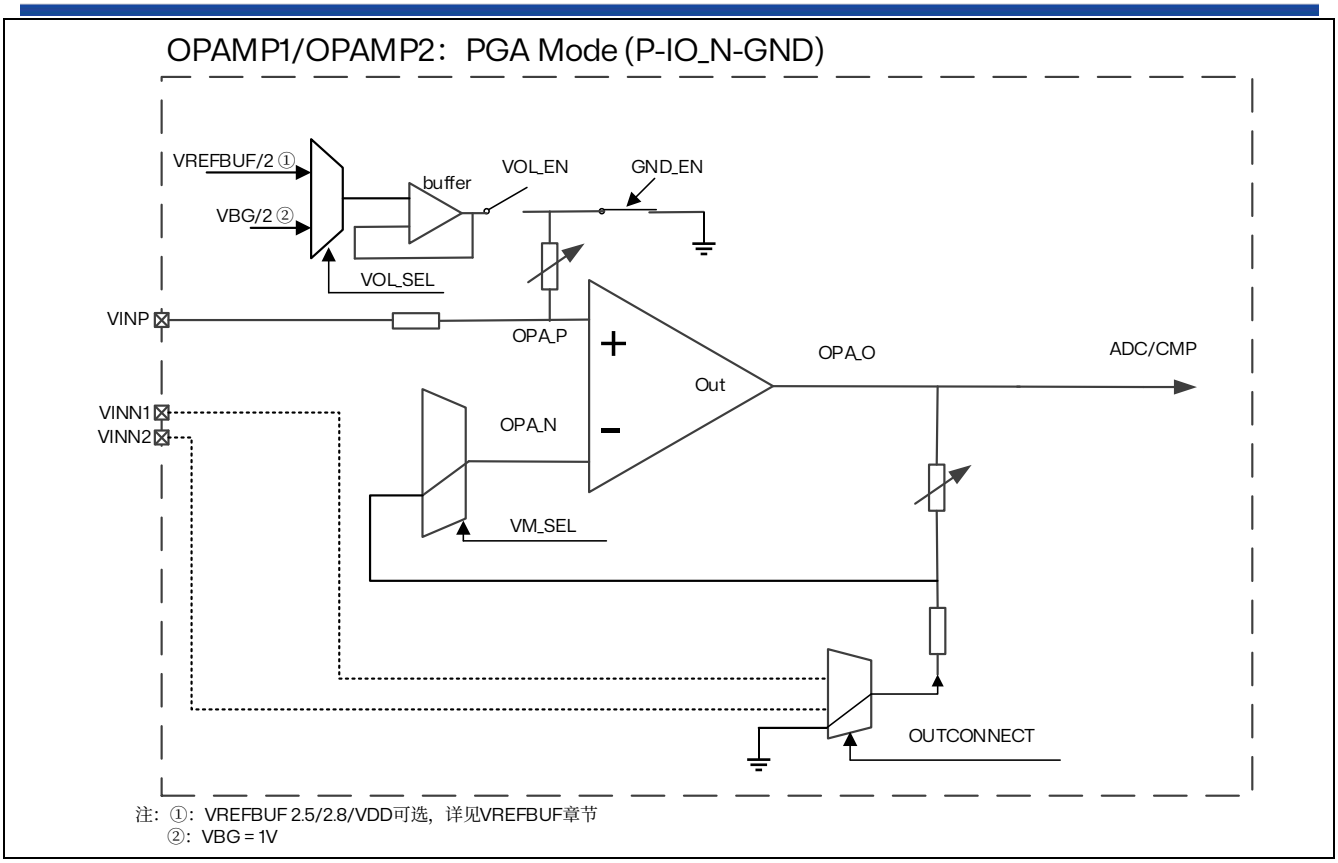


图 17.6 OPAMP1/OPAMP2 PGA 模式（P-IO，N-GND）

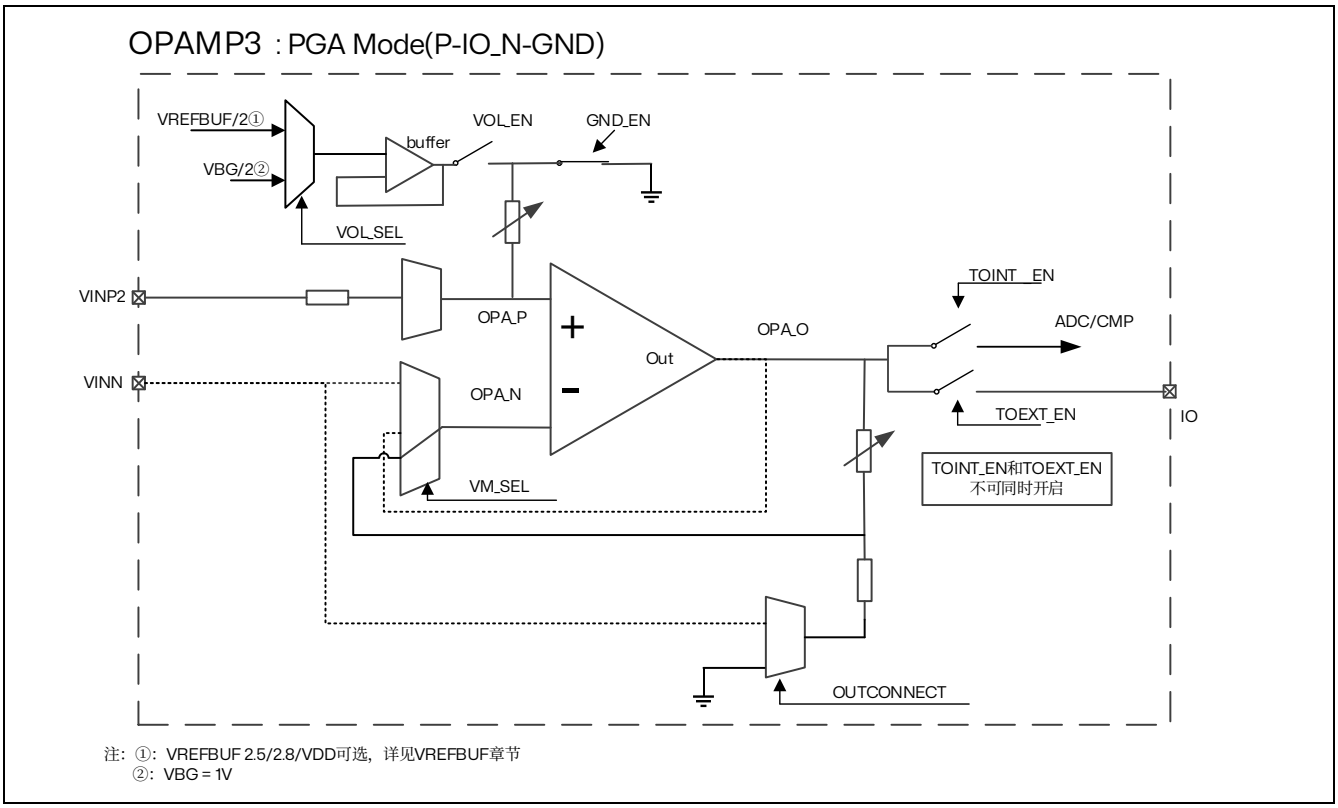


图 17.7 OPAMP3 PGA 模式（P-IO，N-GND）

17.3.5 OPAMPx P 端偏置电压的选择

注意：P 端输入外接偏置只能在 PGA 模式下使用

以 PGA 模式（P-偏置，N-GND）为例：

配置步骤：

1. 将 OPAMPx 配置为 PGA 模式
   - 将 OPAMPx_CSR 寄存器的 VM_SEL[1:0]位 = 01
2. OPAMPx P 端选择（仅 OPAMP3）
   - 选 VIN P2：将 OPAMPx_CSR 寄存器的 VP_SEL 位置“1”
3. OPAMPx N 端内部接地
   - 将 OPAMPx_CSR 寄存器的 OUTCONNECT[1:0] = 01
4. 根据需求选择 P 端输入外接偏置选择
   - 外接偏置 GND：OPAMPx_BIAS 寄存器的 GND_EN 位置“1”
   - 外接偏置电压：OPAMPx_BIAS 寄存器的 VOL_EN 位置“1”
5. 选择不同的偏置电压（P 端输入外接偏置 GND，则跳过此步）
   - 偏置电压为 VBG/2：OPAMPx_BIAS 寄存器的 VOL_SEL 位清“0”
   - 偏置配置 VREFBUF/2：OPAMPx_BIAS 寄存器的 VOL_SEL 位置“1”

     注：VBG 为内部稳定的 1V 电压输出

     VREFBUF 2.5/2.8/VDD *可选，详见* VREFBUF *章节*
6. 使能偏置电压缓冲器（P 端输入外接偏置 GND，则跳过此步）
   - OPAMPx_BIAS 寄存器的 VBGINT_EN 位置“1”
7. 放大倍数选择
   - 4X：OPAMPx_CSR 寄存器的 GAIN[1:0] = 00
   - 8X：OPAMPx_CSR 寄存器的 GAIN[1:0] = 01
   - 12X：OPAMPx_CSR 寄存器的 GAIN[1:0] = 10
   - 16X：OPAMPx_CSR 寄存器的 GAIN[1:0] = 11
8. 使能 OPAMPx
   - 将 OPAMPx_CSR 寄存器的 OPAEN 位置“1”

*注意：*1、OPAMP1 *和* OPAMP2 *共用一个* buffer，*即当* OPAMP1 *和* OPAMP2 *都需要用到偏置时，它们只能选择同一偏置挡位，通过* OPAMPx_BIAS *寄存器的* VOL_SEL *位来选择接* VBG/2 *还是* VREFBUF/2，OPAMP3 *单独一个* buffer。

2、VBGINT_EN *和* GND_EN *不可以同时开启*

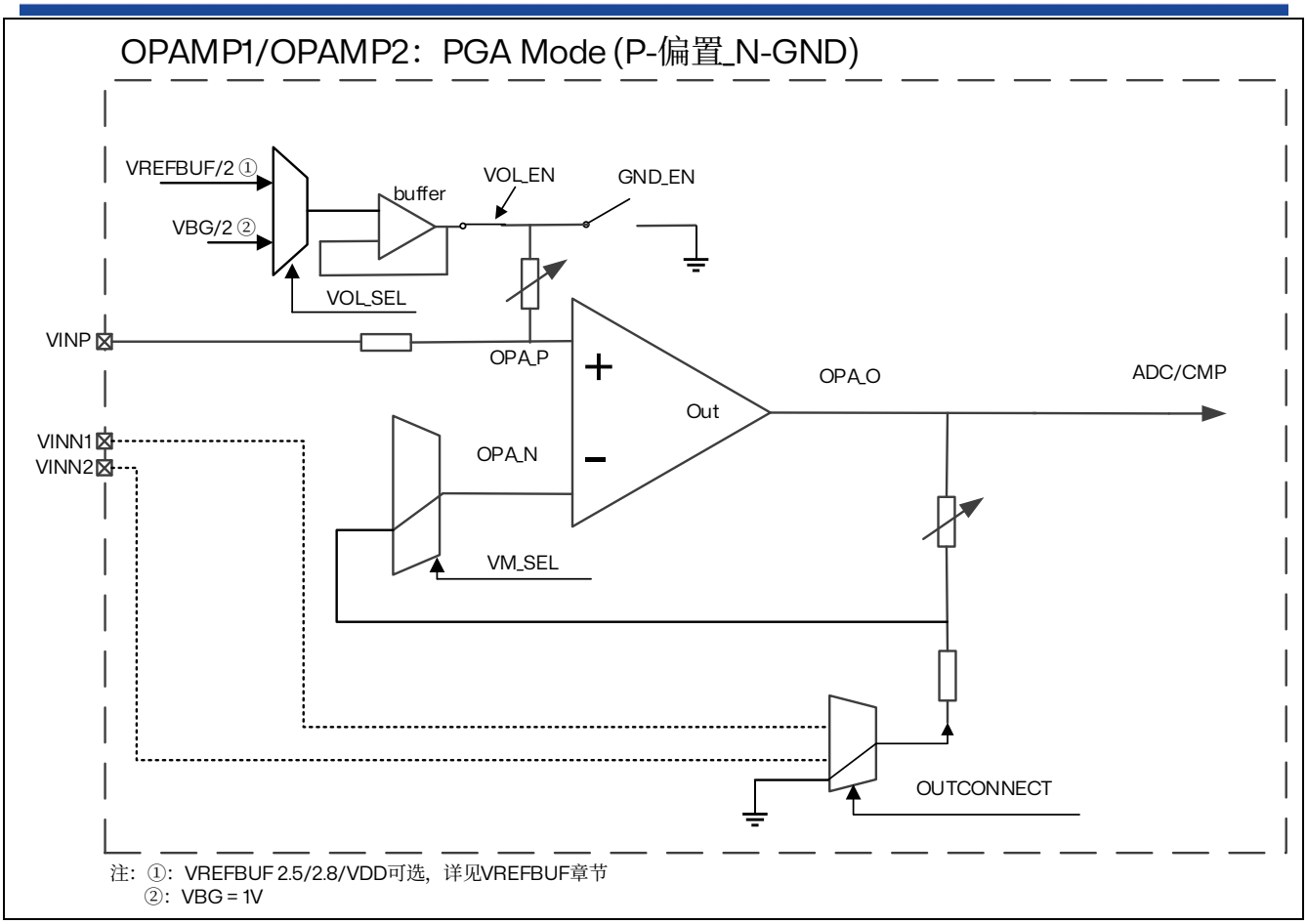


图 17.8 OPAMP1/OPAMP2 PGA 模式（P-偏置，N-GND）

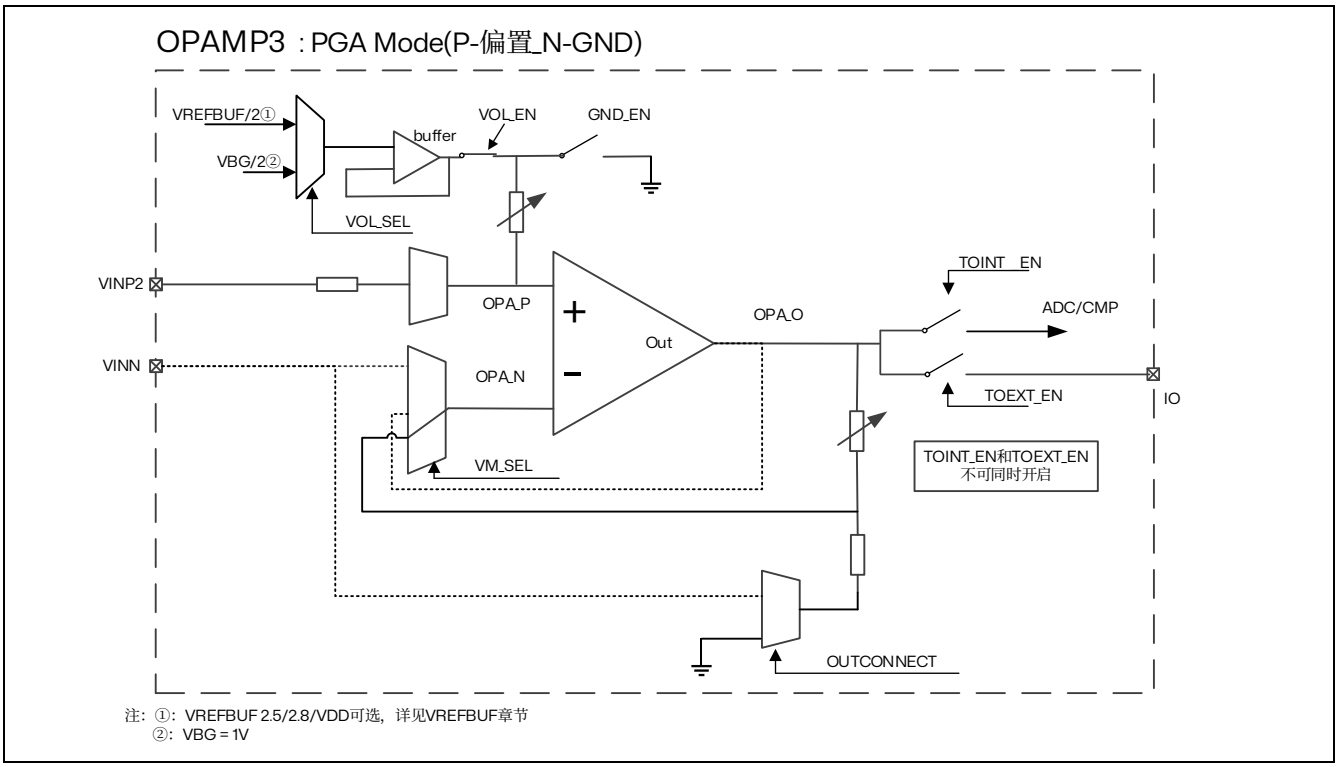


图 17.9 OPAMP3 PGA 模式（P-偏置，N-GND）

17.4 运算放大器寄存器

访问：无等待，支持字，半字和字节访问

17.4.1 运算放大器 OPAMP1 控制寄存器（OPAMP1_CSR）

偏移地址：0x00

复位值：0x0000 0080

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

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

Bit 31 **LOCK**：用于解锁寄存器 OPAMP1_CSR 和寄存器 OPAMP1_BIAS。

0：OPAMP1 寄存器解锁

1：OPAMP1 寄存器被锁住

*注：被锁住的位包括*CSR（all bits）*和*BIAS（all bits）*。通过*RCC_APB2RSTR *寄存器的*SYSCFGRST *位写入* 1 *解除* Lock *状态。*

Bit 30:19 保留，必须保持为复位值。

Bit 18:17 **OUTCONNECT[1:0]**：PGA 模式下 OPAMP1 输出连接

00：不连接

01：OPAMP1 的输出经过内部电阻接地

10：OPAMP1 的输出经过内部电阻接到 OPAMP1_VINN1

11：OPAMP1 的输出经过内部电阻接到 OPAMP1_VINN2

Bit 15:14 **GAIN[1:0]**：OPAMP1 PGA 模式放大倍率选择

00：4X

01：8X

10：12X

11：16X

Bit 13:4 保留，必须保持为复位值。

Bit 3:2 **VM_SEL[1:0]**：N 端输入选择

00：不连接

01：反馈电阻器与 OPAMP1 VINM 输入(PGA 模式)连接，PGA 模式放大倍率取决于 GAIN[1:0]的设置

10：OPAMP1_out 连接到 OPAMP1 VINM 输入(电压跟随器模式)

11：不连接

*注：*PGA *模式下，反向输入选择取决于*OUTCONNECT[1:0]

Bit 1 保留，必须保持为复位值。

Bit 0 **OPAEN**：OPAMP1 使能位

0：OPAMP1 禁止

1：OPAMP1 使能

17.4.2 运算放大器 OPAMP2 控制寄存器（OPAMP2_CSR）

偏移地址：0x04

复位值：0x0000 0080

访问：无等待，支持字，半字和字节访问

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

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

Bit 31 **LOCK**：用于解锁寄存器 OPAMP2_CSR 和寄存器 OPAMP2_BIAS。

0：OPAMP2 寄存器解锁

1：OPAMP2 寄存器被锁住

*注：被锁住的位包括*CSR（all bits）*和*BIAS（all bits）*。通过*RCC_APB2RSTR *寄存器的*SYSCFGRST *位写入* 1 *解除* Lock *状态。*

Bit 30:19 保留，必须保持为复位值。

Bit 18:17 **OUTCONNECT[1:0]**：PGA 模式下 OPAMP2 输出连接

00：不连接

01：OPAMP2 的输出经过内部电阻接地

10：OPAMP2 的输出经过内部电阻接到 OPAMP2_VINN1

11：OPAMP2 的输出经过内部电阻接到 OPAMP2_VINN2

Bit 15:14 **GAIN[1:0]**：OPAMP2 PGA 模式放大倍率选择

00：4X

01：8X

10：12X

11：16X

Bit 13:4 保留，必须保持为复位值。

Bit 3:2 **VM_SEL[1:0]**：负端输入选择

00：不连接

01：反馈电阻器与 OPAMP2 VINM 输入(PGA 模式)连接，PGA 模式放大倍率取决于 GAIN[1:0]的设置

10：OPAMP2_out 连接到 OPAMP2 VINM 输入(电压跟随器模式)

11：不连接

*注：*PGA *模式下，反向输入选择取决于*OUTCONNECT[1:0]

Bit 1 保留，必须保持为复位值。

Bit 0 **OPAEN**：OPAMP2 使能位

0：OPAMP2 禁止

1：OPAMP2 使能

17.4.3 运算放大器 OPAMP3 控制寄存器（OPAMP3_CSR）

偏移地址：0x08

复位值：0x0000 0080

访问：无等待，支持字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	Res	TOINT_EN	Reserved										OUTCONNECT[1:0]		Res
rw		rw											rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GAIN[1:0]		Reserved				VP_SEL	TOEXT_EN	Reserved				VM_SEL[1:0]		Res	OPAEN
rw	rw					rw	rw					rw	rw		rw

Bit 31 **LOCK**：用于解锁寄存器 OPAMP3_CSR 和寄存器 OPAMP3_BIAS。

0：OPAMP3 寄存器解锁

1：OPAMP3 寄存器被锁住

*注：被锁住的位包括*CSR（all bits）*和*BIAS（all bits）*。通过*RCC_APB2RSTR *寄存器的*SYSCFGRST *位写入* 1 *解除* Lock *状态。*

Bit 30 保留，必须保持为复位值。

Bit 29 **TOINT_EN**：

0：OPAMP3 输出至内部 ADC/CMP 禁止

1：OPAMP3 输出至内部 ADC/CMP 使能

*注：*TOINT_EN *和*TOEXT_EN *不能同时开启*

Bit 28:19 保留，必须保持为复位值。

Bit 18:17 **OUTCONNECT[1:0]**：PGA 模式下 OPAMP2 输出连接

00：不连接

01：OPAMP3 的输出经过内部电阻接地

10：OPAMP3 的输出经过内部电阻接到 OPAMP3_VINN

11：不连接

Bit 15:14 **GAIN[1:0]**：OPAMP3 PGA 模式放大倍率选择

00：4X

01：8X

10：12X

11：16X

Bit 13:10 保留，必须保持为复位值。

Bit 9 **VP_SEL**：P 端选择

0：保留

1：OPAMP3_P2

Bit 8 **TOEXT_EN**：OPAMP3 输出至外部 IO 引脚使能位

0：OPAMP3 输出至外部 IO 引脚禁止

1：OPAMP3 输出至外部 IO 引脚使能

*注：*TOINT_EN *和*TOEXT_EN *不能同时开启*

Bit 7:4 保留，必须保持为复位值。

Bit 3:2 **VM_SEL[1:0]**：负端输入选择

00：不连接

01：反馈电阻器与 OPAMP3 VINN 输入(PGA 模式)连接，PGA 模式放大倍率取决于 GAIN[1:0]的设置

10：OPAMP3_out 连接到 OPAMP3 VINN 输入(电压跟随器模式)

11：OPAMP3_N 连接到 OPAMP3 VINN 输入

*注：PGA 模式下，反向输入选择取决于*OUTCONNECT[1:0]

Bit 1 保留，必须保持为复位值。

Bit 0 **OPAEN：**OPAMP3 使能位

0：OPAMP3 禁止

1：OPAMP3 使能

17.4.4 运算放大器 OPAMP1 偏置寄存器（OPAMP1_BIAS）

偏移地址：0x0C

复位值：0x0000 0000

访问：无等待，支持字，半字和字节访问

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

Bit 31:4 保留，必须保持为复位值。

Bit 3 **VBGINT_EN：**偏置电压缓冲器使能位

0：禁止缓冲器

1：使能缓冲器

*注意：*VBGINT_EN *和*GND_EN *不可以同时开启*

Bit 2 **VOL_SEL：**P 端输入外接偏置电压选择

0：偏置电压选择为 VBG/2

1：偏置电压选择为 VREFBUF/2（VREFBUF 可选 2.5V/2.8V/VDD，详见 12 节 VREFBUF）

*注意：*OPAMP1 *和*OPAMP2 *的这个控制位共同控制偏置电压的选择，当*OPAMP1 *设定选择*VBG *或*VREFBUF *时，*OPAMP2 *也会被设定为*VBG *或*VREFBUF*，反之同样。*

Bit 1 **GND_EN：**P 端输入外接偏置 GND 使能

0：禁止

1：使能

*注意：*GND_EN *和*VBGINT_EN *不可以同时开启*

Bit 0 **VOL_EN：**P 端输入外接偏置电压使能

0：禁止

1：使能

17.4.5 运算放大器 OPAMP2 偏置寄存器（OPAMP2_BIAS）

偏移地址：0x10
复位值：0x0000 0000
访问：无等待，支持字，半字和字节访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												VBGINT_EN	VOL_SEL	GND_EN	VOL_EN
												rw	rw	rw	rw

Bit 31:4 保留，必须保持为复位值。

Bit 3 **VBGINT_EN**：偏置电压缓冲器使能位

0：禁止缓冲器

1：使能缓冲器

*注意：*VBGINT_EN *和*GND_EN *不可以同时开启*

Bit 2 **VOL_SEL**：P 端输入外接偏置电压选择

0：偏置电压选择为 VBG/2

1：偏置电压选择为 VREFBUF/2（VREFBUF 可选 2.5V/2.8V/VDD，详见 12 节 VREFBUF）

*注意：*OPAMP1 *和*OPAMP2 *的这个控制位共同控制偏置电压的选择，当*OPAMP1 *设定选择*VBG *或*VREFBUF *时，*OPAMP2 *也会被设定为*VBG *或*VREFBUF*，反之同样。*

Bit 1 **GND_EN**：P 端输入外接偏置 GND 使能

0：禁止

1：使能

*注意：*GND_EN *和*VBGINT_EN *不可以同时开启*

Bit 0 **VOL_EN**：P 端输入外接偏置电压使能

0：禁止

1：使能

17.4.6 运算放大器 OPAMP3 偏置寄存器（OPAMP3_BIAS）

偏移地址：0x14
复位值：0x0000 0000
访问：无等待，支持字，半字和字节访问

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												VBGINT_EN	VOL_SEL	GND_EN	VOL_EN
												rw	rw	rw	rw

Bit 31:4 保留，必须保持为复位值。

Bit 3 **VBGINT_EN**：偏置电压缓冲器使能位

0：禁止缓冲器

1：使能缓冲器

注意：VBGINT_EN 和 GND_EN 不可以同时开启

Bit 2 **VOL_SEL：**P 端输入外接偏置电压选择

0：偏置电压选择为 VBG/2

1：偏置电压选择为 VREFBUF/2（VREFBUF 可选 2.5V/2.8V/VDD，详见 12 节 VREFBUF）

Bit 1 **GND_EN：**P 端输入外接偏置 GND 使能

0：禁止

1：使能

注意：GND_EN 和 VBGINT_EN 不可以同时开启

Bit 0 **VOL_EN：**P 端输入外接偏置电压使能

0：禁止

1：使能

18 比较器（CMP）

18.1 简介

RX32S632 内嵌 2 个轨对轨比较器，可独立使用，也可与高级定时器或 OPAMP 配合使用。

18.2 比较器特征

比较器具有以下特征：

- 轨对轨输入/输出
- 具有迟滞功能
- 可选遮蔽源

18.3 比较器开关控制

通过设置 CMP_CxCR 寄存器的 EN 位可给 CMP 上电。设置使能位时，它将 CMP 从断电状态唤醒，清除使能位可停止比较器工作。

18.4 比较器器输入和输出

CMP1-2 有 3 个正向输入通道可选择。CMP1-2 的正向输入包含外部引脚通道和 OPAMP 的输出。CMP1 的正向输入有 2 个 IO 可选，CMP2 的正向输入有 1 个 IO 可选。当选择 CMPx_INP 时，可通过 CxCR_INPSEL 位选择哪一个 IO 进行输入。

CMP1 有 3 个反向输入通道可选择，CMP2 有 2 个反向输入通道可选择。CMP1-2 的反向输入包含外部引脚通道和 CRV 电压分压值。CMP1 的反向输入有 1 个 IO 可选，CMP2 的反向输入有 1 个 IO 可选。当选择 CMPx_INM 时，可通过 CMP_CxCR 寄存器的 INMSEL 位选择哪一个 IO 进行输入。

CMP1-2 的输出可以配置为 TIM8 的刹车输入。

18.5 比较器锁定机制

比较器能用于安全的用途，比如过流或者过热保护。在某些特定的安全需求的应用中，有必要保证比较器设置不能被无效寄存器访问或者程序计数器破坏所改变。

为此，比较器控制寄存器可以设为写保护（只读）。在 CMPx 控制寄存器 CxCR 寄存器中，当 LOCK 位设为 1 后，除了 CH_BALNK_EN、INPSEL、POL、CH_DIS、CH_DIS_BLANKING、RIF、FIF、HIF、LIF、RIE、FIE、HIE、LIE 以外，CxCR[31:0]的其他位只读（对应的 CAL 寄存器也会锁上，只有 RCC_APB2RSTR_SYSCFGRST 置 1 可解除 lock），具体请参考相关寄存器描述部分。

18.6 比较器框图

比较器框图如下图所示：

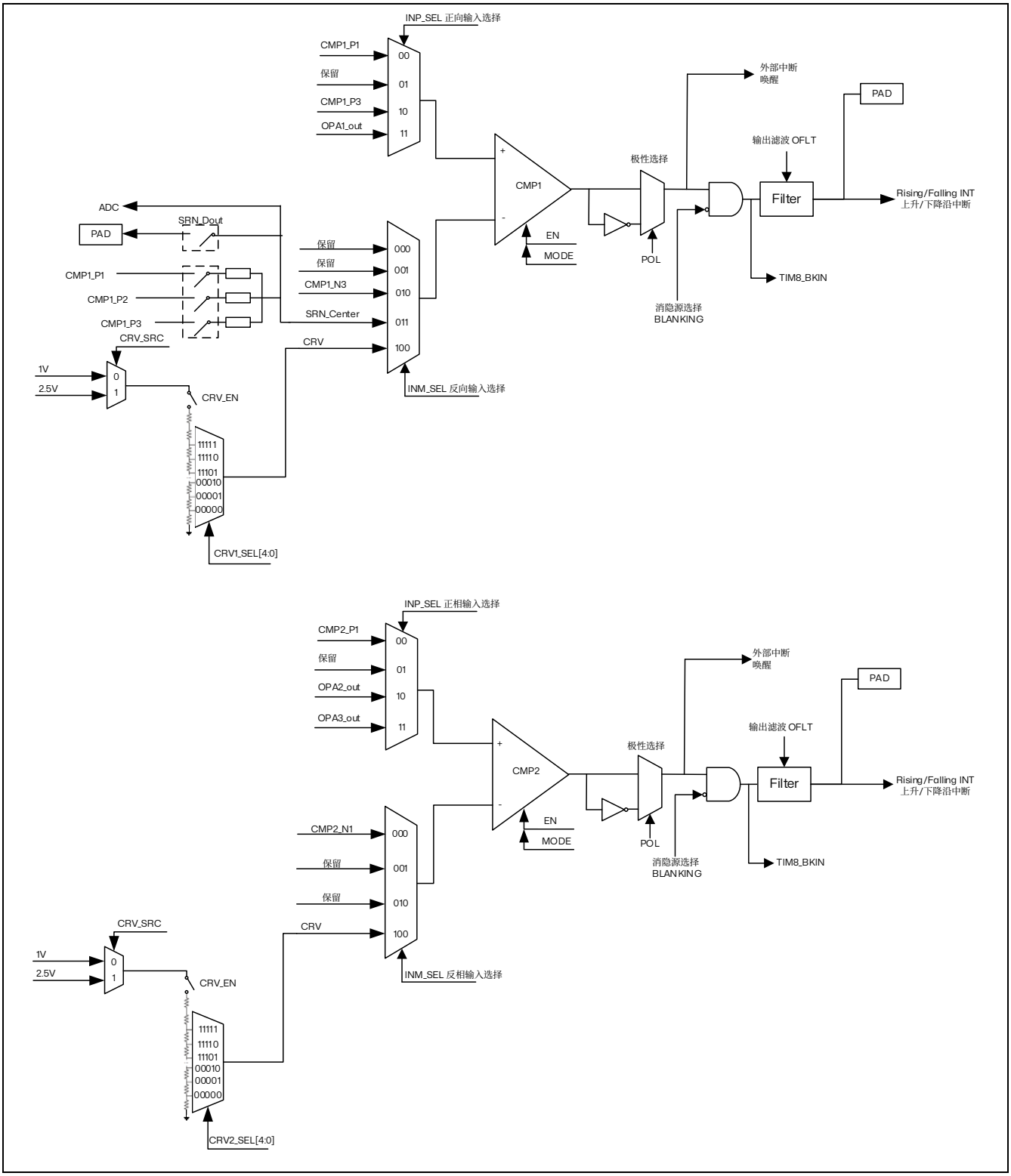


图 18.1 比较器框图

18.7 比较器防误触发功能

（1）当 CMP 和 ADC 同时使用时

当 CMP 的 P 或 N 端与 ADC 采样通道是同一个 IO 时，当外部接 RC 电路或 ADC 在通道转换时造成压差，此时 CMP 可能被 ADC 在通道转换时产生的压差影响而导致误触发。

当 CMP_CxCR 寄存器的 BAS_EN=1 时，根据 CMP_CxCR 寄存器的 INPSEL 和 CMP_CxCR 寄存器的 INMSEL（x=1/2）的配置，在对应的 ADC 采样通道采样时对 CMP 输出进行遮蔽，这样可以防止由 ADC 的转换压差和 RC 电路造成的 CMP 误触发。

注意：配置 ADC_CxCR *寄存器中的* BASW *位可以选择* ADC *采样的遮蔽宽度*

（2）当 OPAMP 和 ADC 同时使用时

由于 OPAMP 的输出内部连接至 CMP，即当 OPAMP 的输入或输出端口与 ADC 采样通道是同一个 IO 时，需要使能 CMP_CxCR 寄存器的 CMP_BAS_EN_OPA=1 进行遮蔽。

下表为 CMP 或 OPAMP 和 ADC 的 IO 对应关系。

表 18.1 CMP/ OPAMP 和 ADC 的 IO 对应关系

CMP/OPAMP	ADC	IO	备注
CMP1_P1	ADC1_IN4	PC0	-
CMP1_N3	ADC1_IN7	PB5	-
CMP1_P3	ADC1_IN8	PB4	-
CMP2_P1	ADC1_IN9	PB3	-
OPAMP 2_P1	ADC1_IN12	PA6	OPAMP 2_O 输出内部连接至 CMP2 的正向输入
OPAMP 2_N1	ADC1_IN11	PA7	OPAMP 2_O 输出内部连接至 CMP2 的正向输入
OPAMP 1_N2	ADC1_IN11	PA7	OPAMP 1_O 输出内部连接至 CMP1 的正向输入
OPAMP 3_O	ADC1_IN10	PB1	OPAMP 3_O 输出内部连接至 CMP2 的正向输入
OPAMP 3_P2	ADC1_IN9	PB3	OPAMP 3_O 输出内部连接至 CMP2 的正向输入

*注：*P *端选择* OPAMP，N *端使用不会选择* SRN*：当* N *端选择* SRN *时，*P *端只会选择* P1/2/3，*且* ADC *采样也只会采对应的* P *端（*OPAMP *和* SRN *不会同时使用）。*

（3）当 CMP 切换引脚时

当 CMP 的 P 端或 N 端切换引脚会产生电压变化，此时 CMP 会被影响而导致误触发。

CMP 的 P 端切换默认被硬件遮蔽。即 CxCR_CH_BLANK_EN 默认为 1 开启遮蔽功能。而 N 端的切换需软件开启遮蔽。同时，P 端也提供了软件开启遮蔽的办法，具体参考如下：

1. 使能遮蔽，将 CMP_CxCR 寄存器的 CH_DIS_BLANKING 位置“1”
2. 关闭 CMP 通道，将 CMP_CxCR 寄存器的 CH_DIS 位置“1” (N 端是将 CMPx_EN 关闭)
3. 进行 P（N）端的引脚切换
4. 开启 CMP 通道，将 CMP_CxCR 寄存器的 CH_DIS 位清“0” （N 端开启 CMPx_EN）
5. 关闭遮蔽将 CMP_CxCR 寄存器的 CH_DIS_BLANKING 位清“0”

18.8 比较器迟滞功能

该比较器包括一个可编程迟滞，避免带有噪声输入信号的杂散输出转换。

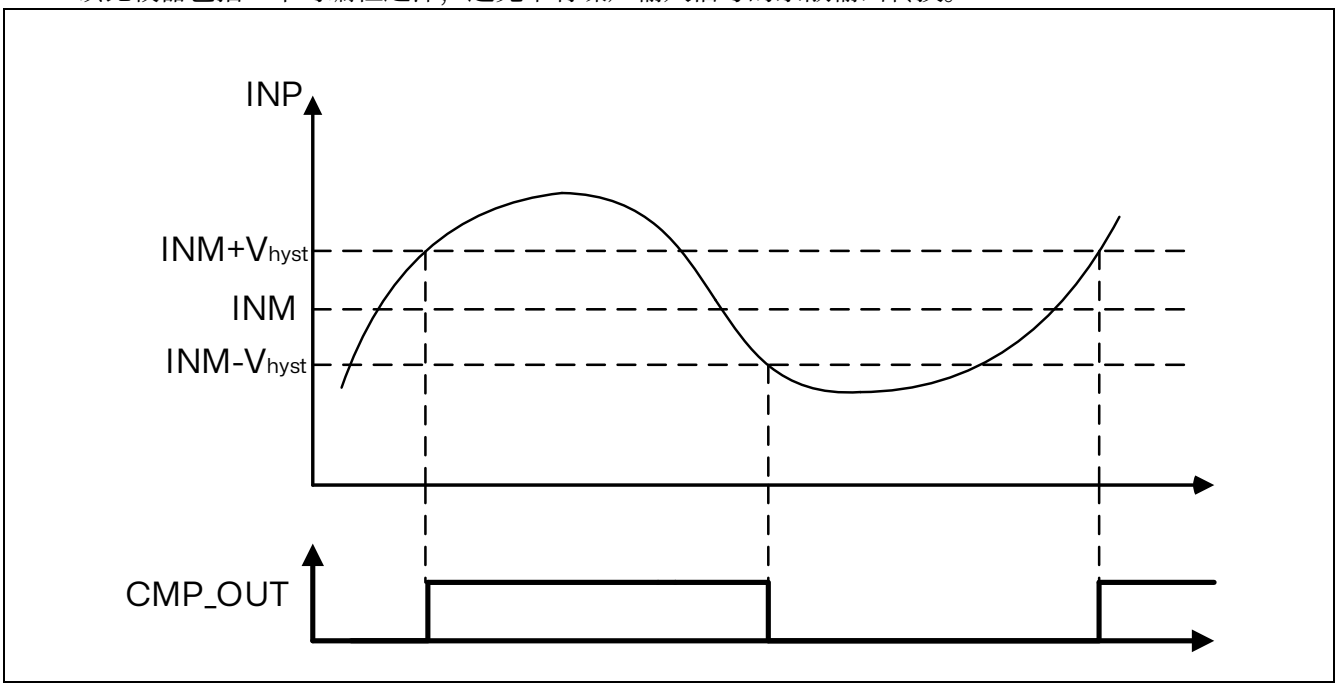


图 18.2 比较器迟滞

18.9 比较器遮蔽功能

遮蔽功能的目是防止在 PWM 周期开始时的短电流尖峰时电流调节跳闸。这里通过设置一个带有定时器输出比较信号的遮蔽窗来实现的。每个比较器通道有软件通过对应的 CMP_CxCR 寄存器的 BLANKING[2:0]位域单独选择遮蔽源，如下表：遮蔽源所示。

表 18.2 遮蔽源

BLANKING[2:0]	CMP1	CMP2
001	TIM2_OC4	TIM2_OC4
010	TIM2_OC3	TIM2_OC3
011	TIM3_OC3	TIM3_OC3
100	TIM8_OC5	TIM8_OC5
101	TIM8_OC4	TIM8_OC4
110	TIM15_OC1	TIM15_OC1
111	TIM15_OC2	TIM15_OC2

对遮蔽信号进行取反后与比较器级联输出进行逻辑与操作，以产生比较器通道 x 的输出。请参考下图中提供的示例。

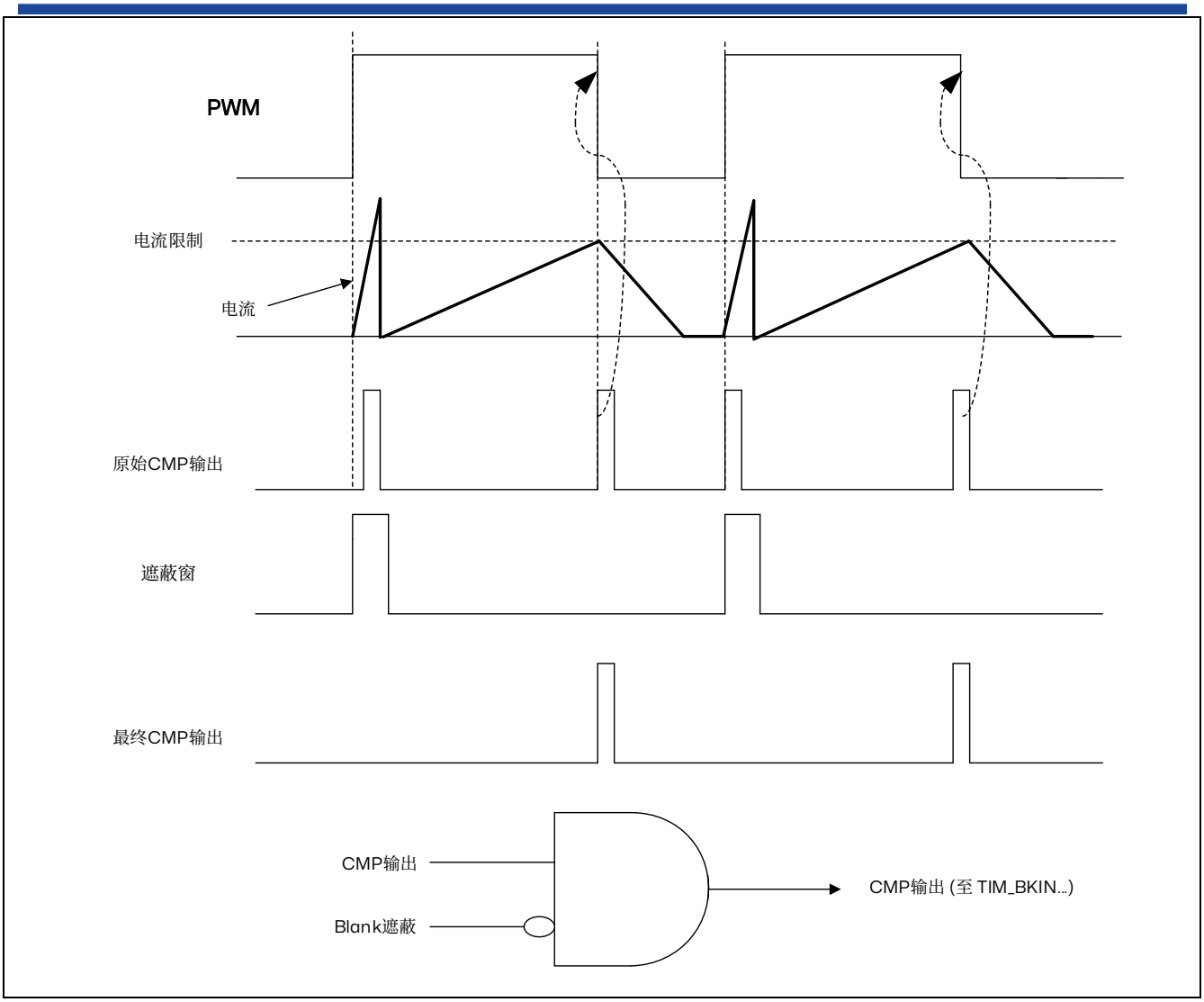


图 18.3 比较器遮蔽功能

18.10 比较器寄存器

访问：无等待，支持字，半字和字节访问

18.10.1 CMPx 控制寄存器（CMP_CxCR）（x=1/2）

偏移地址：0x000+0x04*(x-1)

复位值：0x0000 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	OUT	LIF	FIF	RIF	HIF	FHYST[1:0]		LIE	HIE	BLANKING[2:0]			CMP_BAS_EN_OPA	RHYST[1:0]	
rw	r	rc w0	rc w0	rc w0	rc w0	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POL	OFLT[2:0]			CH_DIS_BLANKING	CH_DIS	BAS_EN	INPSEL[1:0]		INMSEL[2:0]			FIE	RIE	CH_BLANK_EN	EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit 31 **LOCK**：CMP_CxCR 寄存器锁定

该位由软件设置和清除

0：除了 OUT 只读以外，CxCR[31:0]的其他位能读/写

1：除了 CH_BALNK_EN、INPSEL、POL、CH_DIS、CH_DIS_BLANKING、RIF、FIF、HIF、LIF、RIE、FIE、HIE、LIE 以外，CxCR[31:0]的其他位只读（对应的 CAL 寄存器也会锁上，只有 APB2RSTR_SYSCFGRST 置 1 可解除 lock）

Bit 30 **OUT**：比较器的输出

0：低输出

1：高输出

注：该位只读

Bit 29 **LIF**：低电平输出中断事件标志位

0：无低电平输出中断事件

1：低电平输出中断事件发生

Bit 28 **FIF**：下降沿中断事件标志位

0：无下降沿中断事件

1：下降沿中断事件发生

Bit 27 **RIF**：上升沿中断事件标志位

0：无上升沿中断事件

1：上升沿中断事件发生

Bit 26 **HIF**：高电平输出中断事件标志位

0：无高电平输出中断事件

1：高电平输出中断事件发生

Bit 25:24 **FHYST[1:0]**：下降迟滞电压

00：0mv

01：20mv

10：40mv

11：60mv

Bit 23 **LIE**：低电平输出中断使能位

0：低电平输出中断禁止
1：低电平输出中断使能

Bit 22 **HIE**：高电平输出中断使能位
0：高电平输出中断禁止
1：高电平输出中断使能

Bit 21:19 **BLANKING[2:0]**：遮蔽源选择
000：无遮蔽
001：TIM2 OC4 被选为遮蔽源
010：TIM2 OC3 被选为遮蔽源
011：TIM3 OC3 被选为遮蔽源
100：TIM8 OC5 被选为遮蔽源
101：TIM8 OC4 被选为遮蔽源
110：TIM15 OC1 被选为遮蔽源
111：TIM15 OC2 被选为遮蔽源

注：遮蔽需要屏蔽掉比较器输出 OUT *和中断事件* RIF、FIF

Bit 18 **CMP_BAS_EN_OPA**：OPAMP 通道 ADC 采样的遮蔽
0：OPAMP 通道遮蔽 ADC 采样禁止
1：OPAMP 通道遮蔽 ADC 采样使能（参考表 19.1 CMP/ OPAMP 和 ADC 的 IO 对应关系）

Bit 17:16 **RHYST[1:0]**：上升迟滞电压
00：0mv
01：20mv
10：40mv
11：60mv

Bit 15 **POL**：比较器输出极性
0：同向输出
1：反向输出

Bit 14:12 **OFLT[2:0]**：比较器的输出滤波，连续的 PCLK 时钟比较输出不变则认为有效，否则保持不变
000：无滤波
001：32 个时钟周期
010：64 个时钟周期
011：128 个时钟周期
100：256 个时钟周期
101：512 个时钟周期
110：1024 个时钟周期
111：2048 个时钟周期

Bit 11 **CH_DIS_BLANKING**：切换通道前进行 BLANKING 遮蔽
0：通道切换前不开启遮蔽
1：通道切换前开启遮蔽

Bit 10 **CH_DIS**：P 端通道切换时软件主动关闭通道（N 端由 CxCR_EN 控制）
0：通道切换时不主动关闭通道
1：通道切换时软件主动关闭通道，避免切换引脚造成 CMP 误触发

Bit 9 **BAS_EN**：ADC 采样的遮蔽使能

0：遮蔽 ADC 采样禁止

1：遮蔽 ADC 采样使能（参考表 19.1 CMP/ OPAMP 和 ADC 的 IO 对应关系）

注意：配置 ADC_CxCR *寄存器中的* BASW *位可以选择* ADC *采样的遮蔽宽度*

Bit 8:7 **INPSEL[1:0]**：比较器 x 正向输入信号选择

00：CMPx_P1

01：保留

10：CMPx_P3（CMP1），OPAMP 2_out（CMP2）

11：OPAMP 1_out（CMP1），OPAMP 3_out（CMP2）

Bit 6:4 **INMSEL[2:0]**：比较器 x 反向输入信号选择

000：CMPx_N1（仅 CMP2）

001：保留

010：CMPx_N3（仅 CMP1）

011：CMPx_SRN_Center（仅 CMP1）

100：CMPx_CRV

Bit 3 **FIE**：下降沿中断使能位

0：下降沿中断禁止

1：下降沿中断使能

Bit 2 **RIE**：上升沿中断使能位

0：上升沿中断禁止

1：上升沿中断使能

Bit 1 **CH_BLANK_EN**：硬件切换 P 端遮蔽（N 端需软件开启遮蔽）

由硬件默认为 1 开启 P 端通道切换遮蔽，可软件关闭。

具体参考章节 19.7 比较器防误触发功能。

0：硬件遮蔽关闭

1：硬件遮蔽使能

Bit 0 **EN**：比较器使能位

0：比较器禁止

1：比较器使能

注意：比较器的时钟由 RCC_APB2ENR *的* SYSCFGEN *位控制。*

18.10.2 CMPx 校正寄存器（CMP_RGxCAL）（x=1/2）

偏移地址：0x040+0x04*(x-1)

复位值：0x0000 0708

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

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

Bit 31:12 保留，必须保持为复位值。

Bit 11:8 **NADJ[3:0]：**CMP NMOS offset 偏置调节

Bit 7:6 保留，必须保持为复位值。

Bit 5 **NEN：**CMP NMOS 校正使能位

0：CMP NMOS 校正禁止

1：CMP NMOS 校正使能

Bit 4 **PEN：**CMP PMOS 校正使能位

0：CMP PMOS 校正禁止

1：CMP PMOS 校正使能

Bit 3:0 **PADJ[3:0]：**CMP PMOS offset 偏置调节

18.10.3 CMP 控制寄存器 1（CMP_CR1）

偏移地址：0x080

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												SRN_BAS_EN	SRN_LOCK	SRN_Dout	SRN_EN
												rw	rw	rw	rw

Bit 15:4 保留，必须保持为复位值。

Bit 3 **SRN_BAS_EN：**ADC 采样的遮蔽使能

0：遮蔽 ADC 采样禁止

1：遮蔽 ADC 采样使能（需同时开启 CxCR_BAS_EN）

Bit 2 **SRN_LOCK：**

该位由软件设置和清除。

0：CR1 的所有位能读/写

1：CR1 的所有位只读

注：只有 RCC_APB2RSTR_SYSCFGRST *置*1 *可解除锁定*

Bit 1 **SRN_Dout：**星型电阻网络输出使能位

该位由软件设置和清除。

0：星型电阻网络不可通过 PAD 测量

1：星型电阻网络可通过 PAD 测量（前提：CR1_SRN_EN=1）

Bit 0 **SRN_EN：**星型电阻网络使能位

该位由软件设置和清除。

0：CMP_P1/P2/P3 不接入星型电阻网络（仅 CMP1）

1：CMP_ P1/P2/P3 接入星型电阻网络（仅 CMP1）

18.10.4 CMP 控制寄存器 2（CMP_CR2）

偏移地址：0x084

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		CRV_BAS_EN	CRV2_SEL[4:0]					CRV1_SEL[4:0]					CRV_LOCK	CRV_EN	CRV_SRC
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit 15:14 保留，必须保持为复位值。

Bit 13 **CRV_BAS_EN**：ADC 采样的遮蔽使能位

0：遮蔽 ADC 采样禁止

1：遮蔽 ADC 采样使能

Bit 12:8 **CRV2_SEL[4:0]**：比较器 2 外部参考电压选择：

00000b：1/32 源
00001b：2/32 源
00010b：3/32 源
00011b：4/32 源
00100b：5/32 源
00101b：6/32 源
00110b：7/32 源
00111b：8/32 源
01000b：9/32 源
01001b：10/32 源
01010b：11/32 源
01011b：12/32 源
01100b：13/32 源
01101b：14/32 源
01110b：15/32 源
01111b：16/32 源
10000b：17/32 源
10001b：18/32 源
10010b：19/32 源
10011b：20/32 源
10100b：21/32 源
10101b：22/32 源
10110b：23/32 源
10111b：24/32 源
11000b：25/32 源
11001b：26/32 源
11010b：27/32 源
11011b：28/32 源
11100b：29/32 源
11101b：30/32 源
11110b：31/32 源
11111b：32/32 源

Bit 7:3 **CRV1_SEL[4:0]**：比较器 1 外部参考电压选择：

00000b：1/32 源
00001b：2/32 源
00010b：3/32 源
00011b：4/32 源
00100b：5/32 源
00101b：6/32 源
00110b：7/32 源
00111b：8/32 源
01000b：9/32 源
01001b：10/32 源
01010b：11/32 源
01011b：12/32 源
01100b：13/32 源
01101b：14/32 源
10000b：17/32 源
10001b：18/32 源
10010b：19/32 源
10011b：20/32 源
10100b：21/32 源
10101b：22/32 源
10110b：23/32 源
10111b：24/32 源
11000b：25/32 源
11001b：26/32 源
11010b：27/32 源
11011b：28/32 源
11100b：29/32 源
11101b：30/32 源

01110b：15/32 源　　11110b：31/32 源
01111b：16/32 源　　11111b：32/32 源

Bit 2 **CRV_LOCK：**

该位由软件设置和清除。

0：CR2 的所有位能读/写

1：CR2 的所有位只读

*注：只有*RCC_APB2RSTR_SYSCFGRST *置*1 *可解除锁定*

Bit 1 **CRV_EN：**比较器外部参考电压使能位

0：比较器外部参考电压禁止

1：比较器外部参考电压使能

Bit 0 **CRV_SRC：**比较器外部电压源选择

0：1V

1：2.5V

19 数学运算协同处理器（ME）

19.1 简介

RX32S632 提供一个数学运算协同处理器（Math Engine），简称 ME。包含除法运算和一个开平方运算。

19.2 功能描述

ARGx 和 RESx 属于同一个寄存器，写入时，是写入到 ARGx，运算完成后读取 RESx 值获取运算结果。

表 19.1 函数运算表

Function	ARG1（写入）	ARG2（写入）	RES1（读取）	RES2（读取）
DIV	x	y	x/y	x%y
SQRT	x	none	$\sqrt{x}$	none

运算步骤：

1. 选择除法运算还是开平方运算（ME_CR 寄存器的 FUN=0 是除法运算，ME_CR 寄存器的 FUN=1 是开平方运算）；
2. ARG1 填入启动运算，有 1 个以上参数需要先填入其他参数，再填入 ARG1；
3. 运算启动后，BSY=1；
4. 运算完成，BSY=0。

注意：

1. *当* BSY=1 *时，再次填入* ARG1 *会被忽略；*
2. *除法运算时* ARG2 *不能为* 0，*填* 0 *会触发* CALF=1；RES1 *为* ARG1 *除* ARG2 *的整数部分，* RES2 *为余数。*

19.3 ME 寄存器

访问：无等待，支持字，半字和字节访问

19.3.1 ME 控制寄存器（ME_CR）

偏移地址：0x00

复位值：0x0000 0000

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

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

Bit 31 **BSY：**

0：空闲或运算完成

1：运算中

Bit 30 **CALF：**运算错误标志位

0：运算正常

1：运算错误，除数为 0

Bits 29:5 保留，必须保持为复位值。

Bit 4 **SN：**

0：无符号

1：有符号

Bits 3:1 保留，必须保持为复位值。

Bit 0 **FUN：**函数运算选择

0：DIV

1：SQRT

19.3.2 ME ARG1/RES1 寄存器（ARG1/RES1）

偏移地址：0x04

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARG1/RES1[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARG1/RES1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:0 **ARG1/RES1[31:0]：**写入 ARG1，填入启动运算

读出 RES1，BSY=0 时才有效

19.3.3 ME ARG2/RES2 寄存器（ARG2/RES2）

偏移地址：0x08

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARG2/RES2[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARG2/RES2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:0 **ARG2/RES2[31:0]：**

读出 RES2，BSY=0 时才有效

20 实时时钟（RTC）

20.1 RTC 简介

实时时钟是一个独立的定时器。RTC 模块拥有一组连续计数的计数器，在相应软件配置下，可提供时钟日历的功能。修改计数器的值可以重新设置系统当前的时间和日期。

在系统复位或从待机模式唤醒后，RTC 的设置和时间维持不变。

系统复位后，对 RTC 的访问被禁止。执行以下操作将使能对 RTC 的访问：

- 设置寄存器 RCC_APB1ENR 的 PWREN 位，使能电源。
- 设置寄存器 PWR_CR1 的 DBP 位，使能对 RTC 的访问。

20.2 主要特性

- 可编程的预分频系数：分频系数最高为 2^{20}。
- 32 位的可编程计数器，可用于较长时间段的测量。
- 1 个时钟源： RTC 时钟。
- 1 独立的复位类型：
  - APB1 接口由系统复位；
- 3 个专门的可屏蔽中断：
  - 闹钟中断，用来产生一个软件可编程的闹钟中断。
  - 秒中断，用来产生一个可编程的周期性中断信号（最长可达 1 秒）。
  - 溢出中断，指示内部可编程计数器溢出并回转为 0 的状态。

20.3 功能描述

20.3.1 概述

RTC 由两个主要部分组成（参见下图）。第一部分（APB1 接口）用来和 APB1 总线相连。此单元还包含一组 16 位寄存器，可通过 APB1 总线对其进行读写操作。APB1 接口由 APB1 总线时钟驱动，用来与 APB1 总线接口。

另一部分（RTC 核心）由一组可编程计数器组成，分成两个主要模块。第一个模块是 RTC 的预分频模块，它可编程产生最长为 1 秒的 RTC 时间基准 TR_CLK。RTC 的预分频模块包含了一个 20 位的可编程分频器（RTC 预分频器）。如果在 RTC_CR 寄存器中设置了相应的允许位，则在每个 TR_CLK 周期中 RTC 产生一个中断（秒中断）。第二个模块是一个 32 位的可编程计数器，可被初始化为当前的系统时间。系统时间按 TR_CLK 周期累加并与存储在 RTC_ALR 寄存器中的可编程时间相比较，如果 RTC_CR 控制寄存器中设置了相应允许位，比较匹配时将产生一个闹钟中断。

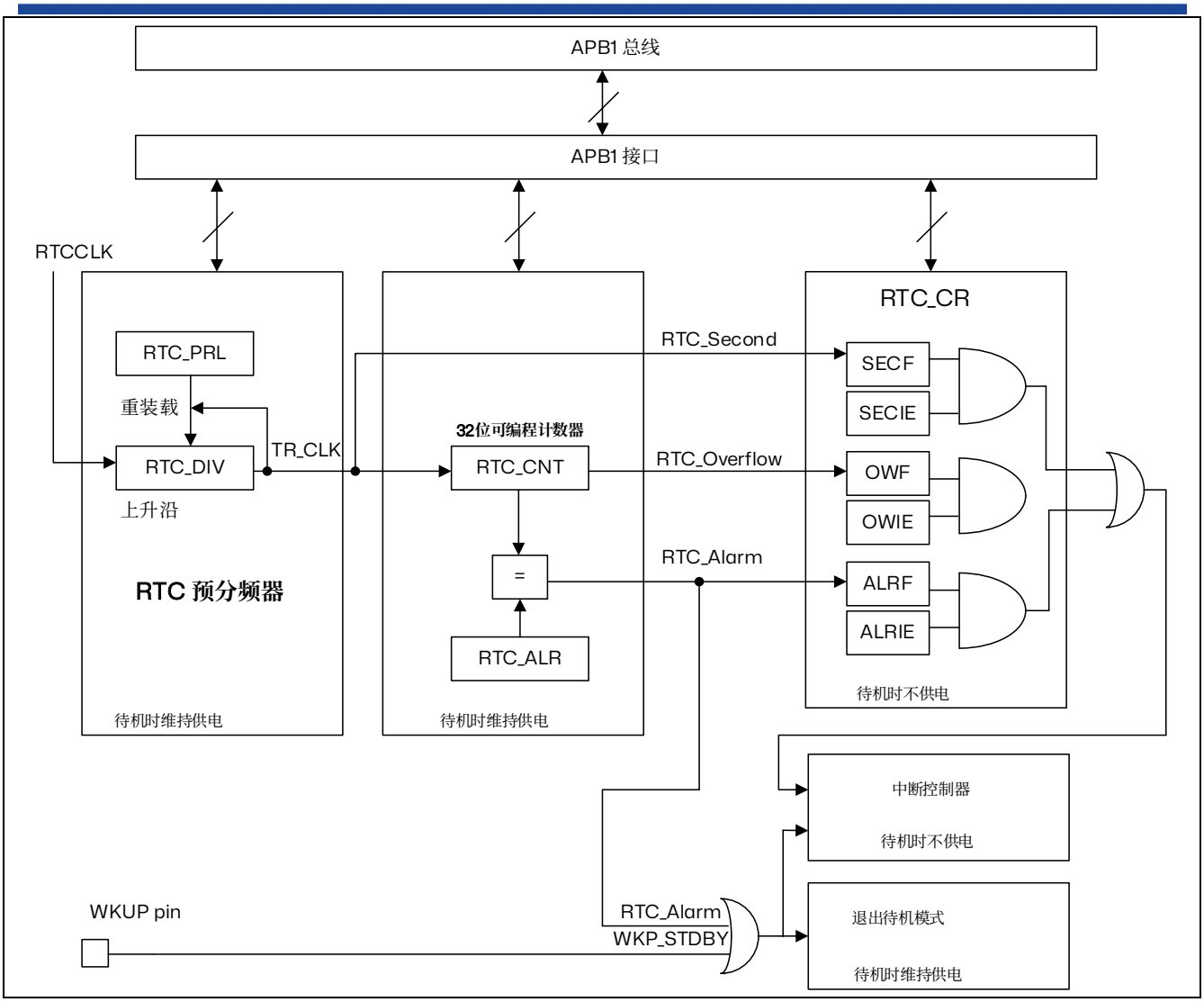


图 20.1 简化的 RTC 框图

20.3.2 复位过程

除了 RTC_PRL、RTC_ALR、RTC_CNT 和 RTC_DIV 寄存器外，所有的系统寄存器都由系统复位或电源复位进行异步复位。

20.3.3 读 RTC 寄存器

RTC 核完全独立于 RTC APB1 接口。

软件通过 APB1 接口访问 RTC 的预分频值、计数器值和闹钟值。但是，相关的可读寄存器只在与 RTC APB1 时钟进行重新同步的 RTC 时钟的上升沿被更新。RTC 标志也是如此的。

这意味着，如果 APB1 接口曾经被关闭，而读操作又是在刚刚重新开启 APB1 之后，则在第一次的内部寄存器更新之前，从 APB1 上读出的 RTC 寄存器数值可能被破坏了（通常读到 0）。下述几种情况下能够发生这种情形：

- 发生系统复位或电源复位
- 系统刚从待机模式唤醒。
- 系统刚从停机模式唤醒。

所有以上情况中，APB1 接口被禁止时（复位、无时钟或断电）RTC 核仍保持运行状态。

因此，若在读取 RTC 寄存器时，RTC 的 APB1 接口曾经处于禁止状态，则软件首先必须等待 RTC_CRL 寄存器中的 RSF 位（寄存器同步标志）被硬件置‘1’。

注：RTC 的APB1 接口不受低功耗模式的影响。

20.3.4 配置 RTC 寄存器

必须设置 RTC_CRL 寄存器中的 CNF 位，使 RTC 进入配置模式后，才能写入 RTC_PRL、RTC_CNT、RTC_ALR 寄存器。

另外，对 RTC 任何寄存器的写操作，都必须在前一次写操作结束后进行。可以通过查询 RTC_CR 寄存器中的 RTOFF 状态位，判断 RTC 寄存器是否处于更新中。仅当 RTOFF 状态位是‘1’时，才可以写入 RTC 寄存器。

配置过程：

1. 查询 RTOFF 位，直到 RTOFF 的值变为’1’
2. 置 CNF 值为 1，进入配置模式
3. 对一个或多个 RTC 寄存器进行写操作
4. 清除 CNF 标志位，退出配置模式
5. 查询 RTOFF，直至 RTOFF 位变为‘1’以确认写操作已经完成。

仅当 CNF 标志位被清除时，写操作才能进行，这个过程至少需要 3 个 RTCCLK 周期。

20.3.5 RTC 标志的设置

在每一个 RTC 核心的时钟周期中，更改 RTC 计数器之前设置 RTC 秒标志（SECF）。

在计数器到达 0x0000 之前的最后一个 RTC 时钟周期中，设置 RTC 溢出标志（OWF）。

在计数器的值到达闹钟寄存器的值加 1（RTC_ALR+1）之前的 RTC 时钟周期中，设置 RTC_Alarm 和 RTC 闹钟标志（ALRF）。对 RTC 闹钟的写操作必须使用下述过程之一与 RTC 秒标志同步：

- 使用 RTC 闹钟中断，并在中断处理程序中修改 RTC 闹钟和/或 RTC 计数器。
- 等待 RTC 控制寄存器中的 SECF 位被设置，再更改 RTC 闹钟和/或 RTC 计数器。

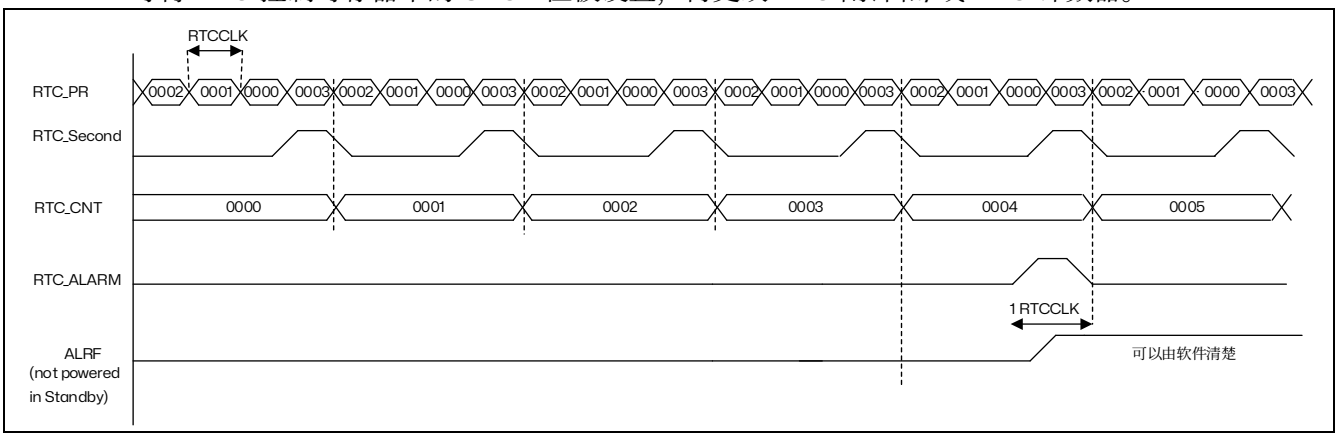


图 20.2 RTC 秒和闹钟波形图示例，PR=0003，ALARM=00004

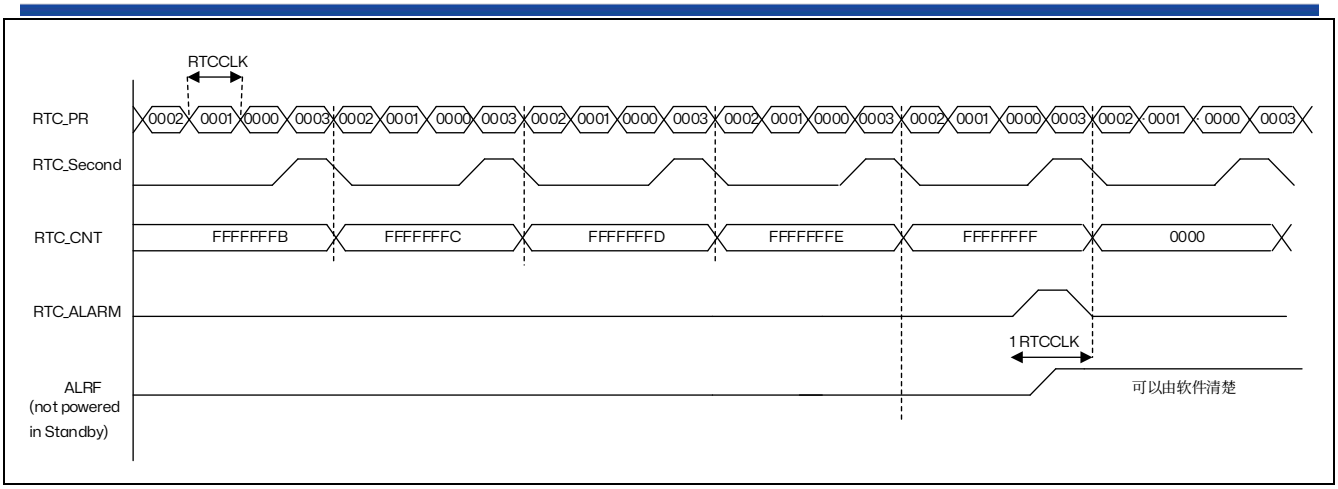


图 20.3 RTC 溢出波形图示例，PR=0003

20.4 RTC 寄存器

访问：无等待，支持字，半字和字节访问

20.4.1 RTC 控制寄存器高位（RTC_CRH）

地址偏移量：0x00

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													OWIE	ALRIE	SECIE
													rw	rw	rw

Bits 15:3 保留，必须保持为复位值。

Bit 2 **OWIE**：允许溢出中断位（Overflow interrupt enable）

0：屏蔽（不允许）溢出中断

1：允许溢出中断

Bit 1 **ALRIE**：允许闹钟中断（Alarm interrupt enable）

0：屏蔽（不允许）闹钟中断

1：允许闹钟中断

Bit 0 **SECIE**：允许秒中断（Second interrupt enable）

0：屏蔽（不允许）秒中断

1：允许秒中断

这些位用来屏蔽中断请求。注意：系统复位后所有的中断被屏蔽，因此可通过写 RTC 寄存器来确保在初始化后没有挂起的中断请求。当外设正在完成前一次写操作时（标志位 RTOFF=0），不能对 RTC_CRH 寄存器进行写操作。

RTC 功能由这个控制寄存器控制。一些位的写操作必须经过一个特殊的配置过程来完成

20.4.2 RTC 控制寄存器低位（RTC_CRL）

偏移地址：0x04

复位值：0x0020

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										RTOFF	CNF	RSF	OWF	ALRF	SECF
										r	rw	rc w0	rc w0	rc w0	rc w0

Bits 15:6 保留，必须保持为复位值。

Bit 5 **RTOFF**：RTC 操作关闭（RTC operation OFF）

RTC 模块利用这位来指示对其寄存器进行的最后一次操作的状态，指示操作是否完成。若此位为‘0’，则表示无法对任何的 RTC 寄存器进行写操作。此位为只读位。

0：上一次对 RTC 寄存器的写操作仍在进行；

1：上一次对 RTC 寄存器的写操作已经完成。

Bit 4 **CNF**：配置标志（Configuration flag）

此位必须由软件置‘1’以进入配置模式，从而允许向 RTC_CNT、RTC_ALR 或 RTC_PRL 寄存器写入数据。只有当此位在被置'1'并重新由软件清‘0’后，才会执行写操作。

0：退出配置模式（开始更新 RTC 寄存器）；

1：进入配置模式。

Bit 3 **RSF**：寄存器同步标志（Registers synchronized flag）

每当 RTC_CNT 寄存器和 RTC_DIV 寄存器由软件更新或清‘0’时，此位由硬件置‘1’。在 APB1 复位后，或 APB1 时钟停止后，此位必须由软件清‘0’。要进行任何的读操作之前，用户程序必须等待这位被硬件置‘1’，以确保 RTC_CNT、RTC_ALR 或 RTC_PRL 已经被同步。

0：寄存器尚未被同步；

1：寄存器已经被同步。

Bit 2 **OWF**：溢出标志（Overflow flag）

当 32 位可编程计数器溢出时，此位由硬件置'1'。如果 RTC_CRH 寄存器中 OWIE=1，则产生中断。此位只能由软件清‘0’。对此位写‘1’是无效的。

0：无溢出；

1：32 位可编程计数器溢出。

Bit 1 **ALRF**：闹钟标志（Alarm flag）

当 32 位可编程计数器达到 RTC_ALR 寄存器所设置的预定值，此位由硬件置‘1’。如果 RTC_CRH 寄存器中 ALRIE=1，则产生中断。此位只能由软件清‘0’。对此位写‘1’是无效的。

0：无闹钟；

1：有闹钟。

Bit 0 **SECF**：秒标志（Second flag）

当 32 位可编程预分频器溢出时，此位由硬件置‘1’同时 RTC 计数器加 1。因此，此标志为分辨率可编程的 RTC 计数器提供一个周期性的信号（通常为 1 秒）。如果 RTC_CRH 寄存器中 SECIE=1，则产生中断。此位只能由软件清除。对此位写‘1’是无效的。

0：秒标志条件不成立；

1：秒标志条件成立。

RTC 的功能由这个控制寄存器控制。当前一个写操作还未完成时，不能写 RTC_CR 寄存器。

*注：1. 任何标志位都将保持挂起状态，直到适当的*RTC_CR *请求位被软件复位，表示所请求的中断已经被接受。*

*2. 在复位时禁止所有中断，无挂起的中断请求，可以对*RTC *寄存器进行写操作。*

*3. 当*APB1 *时钟不运行时，*OWF、ALRF、SECF *和*RSF *位不被更新。*

4. OWF、ALRF、SECF *和*RSF *位只能由硬件置位，由软件来清零。*

5 *若* ALRF=1 *且* ALRIE=1，*则允许产生* RTC *全局中断。如果在* EXTI *控制器中允许产生* EXTI *线* 17 *中断，则允许产生* RTC *全局中断和* RTC *闹钟中断。*

6 *若* ALRF=1，*如果在* EXTI *控制器中设置了* EXTI *线* 17 *的中断模式，则允许产生* RTC *闹钟中断；如果在* EXTI *控制器中设置了* EXTI *线* 17 *的事件模式，则这条线上会产生一个脉冲（不会产生* RTC *闹钟中断）。*

20.4.3 RTC 预分频装载寄存器（RTC_PRLH/RTC_PRLL）

预分频装载寄存器用来保存 RTC 预分频器的周期计数值。它们受 RTC_CR 寄存器的 RTOFF 位保护，仅当 RTOFF 值为‘1’时允许进行写操作。

RTC 预分频装载寄存器高位（RTC_PRLH）

偏移地址：0x08

只写

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												PRL[19:16]			
												w	w	w	w

Bits 15:4 保留，必须保持为复位值。

Bits 3:0 **PRL[19:16]：** RTC 预分频装载值高位（RTC prescaler reload value high）

根据以下公式，这些位用来定义计数器的时钟频率：

$f_{TR_CLK} = f_{RTCCLK}/（PRL[19:0]+1）$

注：不推荐使用 0 值，否则无法正确的产生 RTC 中断和标志位。

RTC 预分频装载寄存器低位（RTC_PRLL）

偏移地址：0x0C

只写

复位值：0x8000

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

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

Bits 31:16 保留，必须保持为复位值。

Bits 15:0 **PRL[15:0]：** RTC 预分频装载值低位

根据以下公式，这些位用来定义计数器的时钟频率：

$f_{TR_CLK} = f_{RTCCLK}/（PRL[19:0]+1）$

注：如果输入时钟频率是 32.768KHz（f_{RTCCLK}），*这个寄存器中写入* 7FFFh *可获得周期为* 1 *秒钟的信号。*

20.4.4 RTC 预分频器余数寄存器（RTC_DIVH / RTC_DIVL）

在 TR_CLK 的每个周期里，RTC 预分频器中计数器的值都会被重新设置为 RTC_PRL 寄存器的值。用户可通过读取 RTC_DIV 寄存器，以获得预分频计数器的当前值，而不停止分频计数器的工作，从而获得精确的时间测量。此寄存器是只读寄存器，其值在 RTC_PRL 或 RTC_CNT 寄存器中的值发生改变后，由硬件重新装载。

RTC 预分频器余数寄存器高位（RTC_DIVH）

偏移地址：0x10

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												RTC_DIV[19:16]			
												r	r	r	r

Bits 15:4 保留，必须保持为复位值。

Bits 3:0 **RTC_DIV[19:16]：**RTC 时钟分频器余数高位（RTC clock divider high）

RTC 预分频器余数寄存器低位（RTC_DIVL）

偏移地址：0x14

复位值：0x8000

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

Bits 15:0 **RTC_DIV[15:0]：**RTC 时钟器余数低位（RTC clock divider low）

20.4.5 RTC 计数器寄存器（RTC_CNTH/RTC_CNTL）

RTC 核有一个 32 位可编程的计数器，可通过两个 16 位的寄存器访问。计数器以预分频器产生的 TR_CLK 时间基准为参考进行计数。RTC_CNT 寄存器用来存放计数器的计数值。他们受 RTC_CR 的位 RTOFF 写保护，仅当 RTOFF 值为‘1’时，允许写操作。在高或低寄存器（RTC_CNTH 或 RTC_CNTL）上的写操作，能够直接装载到相应的可编程计数器，并且重新装载 RTC 预分频器。当进行读操作时，直接返回计数器内的计数值（系统时间）。

RTC 计数器寄存器高位（RTC_CNTH）

偏移地址：0x18

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC_CNT[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:0 **RTC_CNT[31:16]：**RTC 计数器高位（RTC counter high）

可通过读 RTC_CNTH 寄存器来获得 RTC 计数器当前值的高位部分。要对此寄存器进行写操作前，必须先进入配置模式。

RTC 计数器寄存器低位（RTC_CNTL）

偏移地址：0x1C

复位值：0x0000

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

Bits 15:0 **RTC_CNT[15:0]：**RTC 计数器低位。

可通过读 RTC_CNTL 寄存器来获得 RTC 计数器当前值的低位部分。要对此寄存器进行写操作，必须先进入配置模式。

20.4.6 RTC 闹钟寄存器（RTC_ALRH/RTC_ALRL）

当可编程计数器的值与 RTC_ALR 中的 32 位值相等时，即触发一个闹钟事件，并且产生 RTC 闹钟中断。此寄存器受 RTC_CR 寄存器里的 RTOFF 位写保护，仅当 RTOFF 值为‘1’时，允许写操作。

RTC 闹钟寄存器高位（RTC_ALRH）

偏移地址：0x20
只写
复位值：0xFFFF

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC_ALR[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits 15:0 **RTC_ALR[31:16]**：RTC 闹钟值高位（RTC alarm high）
此寄存器用来保存由软件写入的闹钟时间的高位部分。要对此寄存器进行写操作，必须先进入配置模式。

RTC 闹钟寄存器低位（RTC_ALRL）

偏移地址：0x24
只写
复位值：0xFFFF

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

Bits 15:0 **RTC_ALR[15:0]**：RTC 闹钟值低位（RTC alarm low）
此寄存器用来保存由软件写入的闹钟时间的低位部分。要对此寄存器进行写操作，必须先进入配置模式。

21 独立看门狗（IWDG）

21.1 简介

RX32S632 内置一个独立看门狗，提供了更高的安全性、时间的精确性和使用的灵活性。独立看门狗设备可用来检测和解决由软件错误引起的故障；当计数器达到给定的超时值时，产生系统复位。

独立看门狗（IWDG）由专用的低速时钟（LSI）驱动，即使主时钟发生故障它也仍然有效。

IWDG 最适合应用于那些需要看门狗作为一个在主程序之外，能够完全独立工作，并且对时间精度要求较低的场合。

21.2 IWDG 主要性能

- 自由运行的递减计数器
- 时钟由独立的 RC 振荡器提供（可在停止和待机模式下工作）
- 看门狗被激活后，则在计数器计数至 0x000 时产生复位

21.3 IWDG 功能描述

下图为独立看门狗模块的功能框图。

在键寄存器（IWDG_KR）中写入 0xCCCC，开始启用独立看门狗；此时计数器开始从其复位值 0xFFF 递减计数。当计数器计数到末尾 0x000 时，会产生一个复位信号（IWDG_RESET）。

无论何时，只要在键寄存器 IWDG_KR 中写入 0xAAAA，IWDG_RLR 中的值就会被重新加载到计数器，从而避免产生看门狗复位。

21.3.1 硬件看门狗

如果用户在选择字节中启用了“硬件看门狗”功能，在系统上电复位后，看门狗会自动开始运行；如果在计数器计数结束前，若软件没有向键寄存器写入相应的值，则系统会产生复位。

21.3.2 寄存器访问保护

IWDG_PR 和 IWDG_RLR 寄存器具有写保护功能。要修改这两个寄存器的值，必须先向 IWDG_KR 寄存器中写入 0x5555。以不同的值写入这个寄存器将会打乱操作顺序，寄存器将重新被保护。重装载操作（即写入 0xAAAA）也会启动写保护功能。

状态寄存器指示预分频值和递减计数器是否正在被更新。

21.3.3 调试模式

当微控制器进入调试模式时（内核停止），根据调试模块中的 DBG_IWDG_STOP 配置位的状态，IWDG 的计数器能够继续工作或停止。

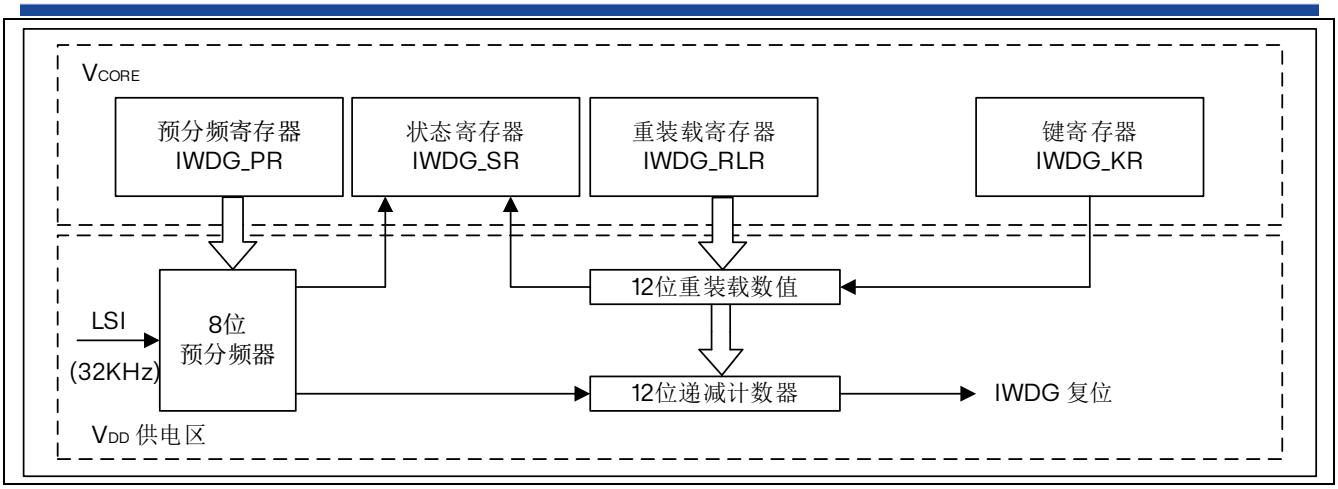


图 21.1 独立看门狗框图

注：看门狗功能处于VDD 供电区，即在停机和待机模式时仍能正常工作。

表 21.1 看门狗超时时间（32KHz 的输入时钟（LSI））(1)

预分频系数	PR[2:0]位	最短时间（ms） RL[11:0] = 0x000	最长时间（ms） RL[11:0] = 0xFFF
/4	0	0.125	512
/8	1	0.25	1024
/16	2	0.5	2048
/32	3	1	4096
/64	4	2	8192
/128	5	4	16384
/256	（6 或 7）	8	32768

注：这些时间是按照32KHz 时钟给出。实际上，MCU 内部的RC 频率会在30KHz 到60KHz 之间变化。此外，即使RC 振荡器的频率是精确的，确切的时序仍然依赖于APB 接口时钟与RC 振荡器时钟之间的相位差，因此总会有一个完整的RC 周期是不确定的。

21.4 IWDG 寄存器

访问：无等待，支持字，半字访问

21.4.1 键寄存器（IWDG_KR）

地址偏移：0x00

复位值：0x0000 0000（在待机模式复位）

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

Bit 31:16 保留，必须保持为复位值。

Bit 15:0 **KEY[15:0]：**键值（只写寄存器，读出值为 0x0000）（Key value）

软件必须以一定的间隔写入 0xAAAA，否则，当计数器为 0 时，看门狗会产生复位。

写入 0x5555 表示允许访问 IWDG_PR 和 IWDG_RLR 寄存器。
写入 0xCCCC，启动看门狗工作（若选择了硬件看门狗则不受此命令字限制）。

21.4.2 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
Reserved															

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

Bit 31:3 保留，必须保持为复位值。

Bit 2:0 PR[2:0]：预分频因子（Prescaler divider）

这些位具有写保护设置。通过设置这些位来选择计数器时钟的预分频因子。要改变预分频因子，IWDG_SR 寄存器的 PVU 位必须为 0。

000：预分频因子=4　　100：预分频因子=64
001：预分频因子=8　　101：预分频因子=128
010：预分频因子=16　　110：预分频因子=256
011：预分频因子=32　　111：预分频因子=256

注：对此寄存器进行读操作，将从 VDD 电压域返回预分频值。如果写操作正在进行，则读回的值可能是无效的。因此，只有当 IWDG_SR 寄存器的 PVU 位为 0 时，读出的值才有效。

21.4.3 重装载寄存器（IWDG_RLR）

地址偏移：0x08
复位值：0x0000 0FFF（待机模式时复位）

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

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

Bit 31:12 保留，必须保持为复位值。

Bit 11:0 RL[11:0]：看门狗计数器重装载值（Watchdog counter reload value）

这些位具有写保护功能。用于定义看门狗计数器的重装载值，每当向 IWDG_KR 寄存器写入 0xAAAA 时，重装载值会被传送到计数器中。随后计数器从这个值开始递减计数。看门狗超时周期可通过此重装载值和时钟预分频值来计算。

只有当 IWDG_SR 寄存器中的 RVU 位为 0 时，才能对此寄存器进行修改。

注：对此寄存器进行读操作，将从 VDD 电压域返回预分频值。如果写操作正在进行，则读回的值可能是无效的。因此，只有当 IWDG_SR 寄存器的 RVU 位为 0 时，读出的值才有效。

21.4.4 状态寄存器（IWDG_SR）

地址偏移：0x0C

复位值：0x0000 0000（待机模式时不复位）

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

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

Bit 31:2 保留，必须保持为复位值。

Bit 1 RVU：看门狗计数器重装载值更新（Watchdog counter reload value update）

此位由硬件置“1”用来指示重装载值的更新正在进行中。当在 VDD 域中的重装载更新结束后，此位由硬件清“0”（最多需 5 个 32KHz 的 RC 周期）。重装载值只有在 RVU 位被清“0”后才可更新。

Bit 0 PVU：看门狗预分频值更新（Watchdog prescaler value update）

此位由硬件置“1”用来指示预分频值的更新正在进行中。当在 VDD 域中的预分频值更新结束后，此位由硬件清“0”（最多需 5 个 32KHz 的 RC 周期）。预分频值只有在 PVU 位被清“0”后才可更新。

注：如果在应用程序中使用了多个重装载值或预分频值，则必须在 RVU *位被清除后才能重新改变预装载值，在* PVU *位被清除后才能重新改变预分频值。然而，在预分频和/或重装值更新后，不必等待* RVU *或* PVU *复位，可继续执行下面的代码。（即是在低功耗模式下，此写操作仍会被继续执行完成。）*

22 串行外设接口（SPI）

22.1 SPI 简介

串行外设接口（SPI）允许芯片与外部设备以半/全双工、同步、串行方式通信。此接口可以被配置成主模式，并为外部从设备提供通信时钟（SCK）。接口还能以多主配置方式工作。

它可用于多种用途，包括使用一条双向数据线的双线单工同步传输。

22.2 SPI 主要特征

22.2.1 SPI 特征

- 3 线全双工同步传输
- 带或不带第三根双向数据线的双线单工同步传输
- 8 或 16 位传输帧格式选择
- 主或从操作
- 支持多主模式
- 8 个主模式波特率预分频系数（最大为 $f_{PCLK}/2$）
- 从模式频率（最大为 $f_{PCLK}/2$）
- 主模式和从模式的快速通信
- 主模式和从模式下均可以由软件或硬件进行 NSS 管理：主/从操作模式的动态改变
- 可编程的时钟极性和相位
- 可编程的数据顺序，MSB 在前或 LSB 在前
- 可触发中断的专用发送和接收标志
- SPI 总线忙状态标志

22.3 SPI 功能描述

22.3.1 概述

SPI 的方框图见下图。

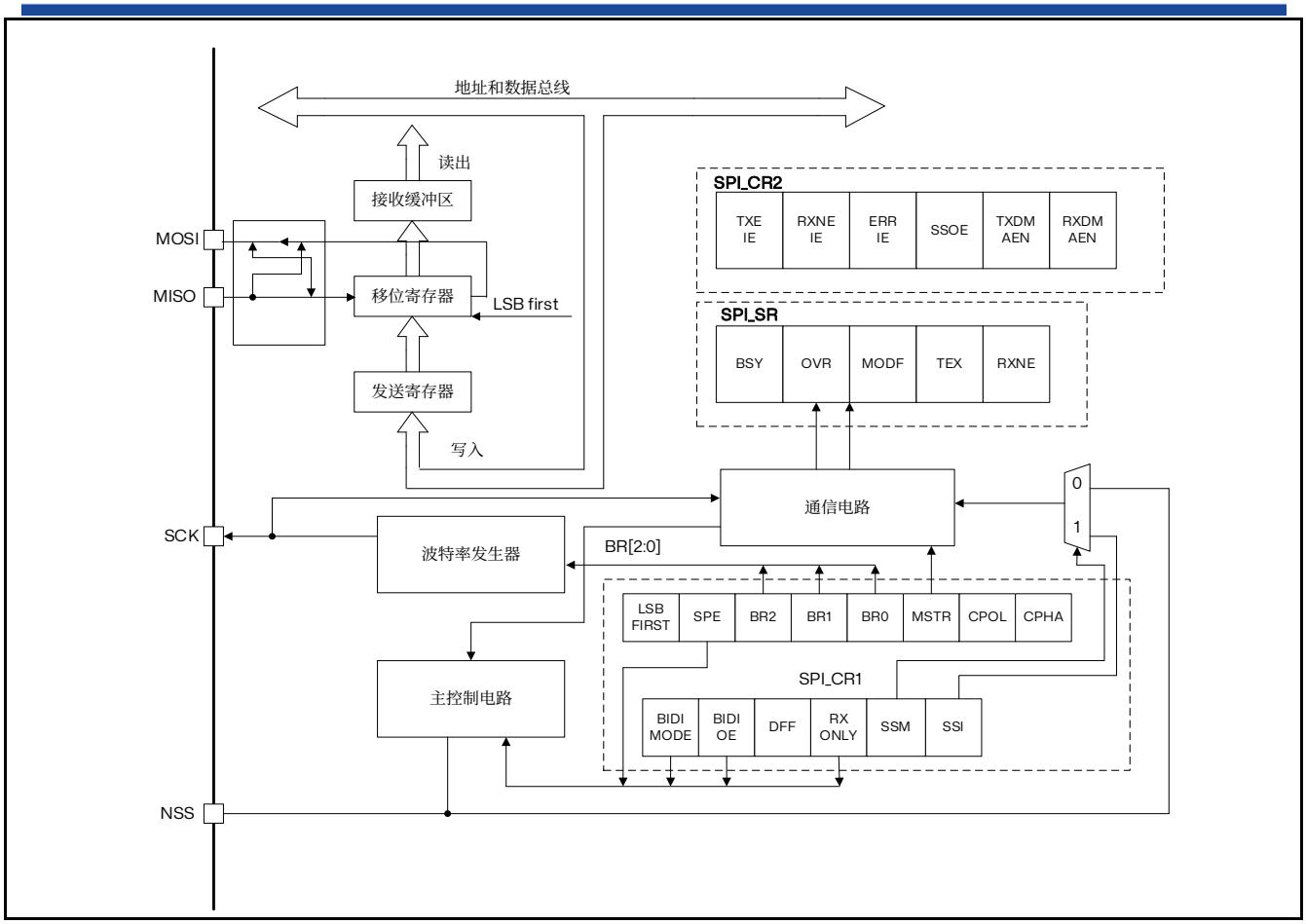
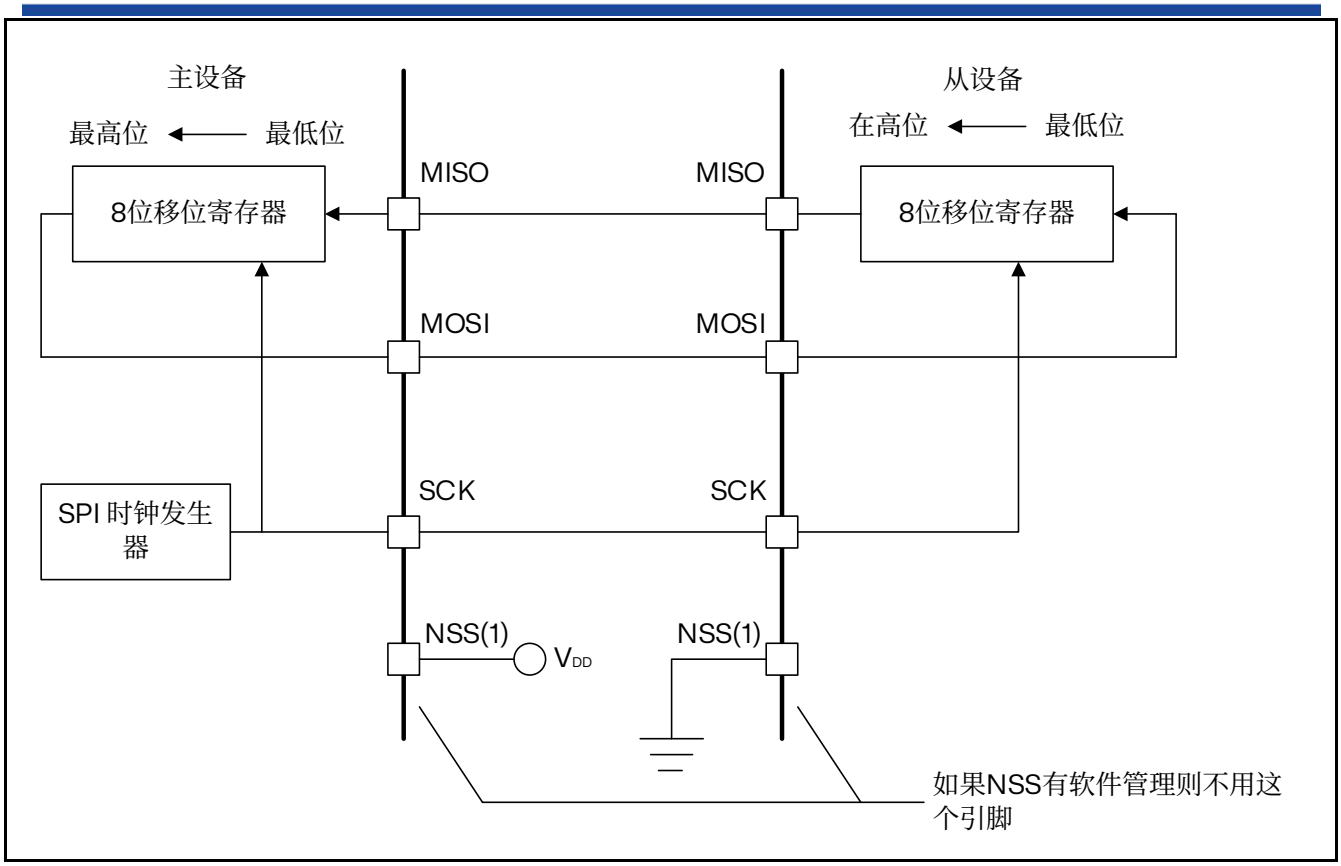


图 22.1 SPI 框图

通常 SPI 通过 4 个引脚与外部器件相连：

- MISO：主设备输入/从设备输出引脚。该引脚在从模式下发送数据，在主模式下接收数据。
- MOSI：主设备输出/从设备输入引脚。该引脚在主模式下发送数据，在从模式下接收数据。
- SCK：串口时钟，作为主设备的输出，从设备的输入
- NSS：从设备选择。这是一个可选的引脚，用来选择主/从设备。它的功能是用来作为“片选引脚”，让主设备可以单独地与特定从设备通讯，避免数据线上的冲突。从设备的 NSS 引脚可以由主设备的一个标准 I/O 引脚来驱动。一旦被使能（SSOE 位），NSS 引脚也可以作为输出引脚，并在 SPI 处于主模式时拉低；此时，所有的 SPI 设备，如果它们的 NSS 引脚连接到主设备的 NSS 引脚，则会检测到低电平，如果它们被设置为 NSS 硬件模式，就会自动进入从设备状态。当配置为主设备、NSS 配置为输入引脚（MSTR=1，SSOE=0）时，如果 NSS 被拉低，则这个 SPI 设备进入主模式失败状态：即 MSTR 位被自动清除，此设备进入从模式。

下图是一个单主和单从设备互连的例子。



1. 这里 NSS 引脚设置为输入

图 22.2 单主和单从应用

MOSI 脚相互连接，MISO 脚相互连接。这样，数据在主和从之间串行地传输（MSB 位在前）。

通信总是由主设备发起。主设备通过 MOSI 脚把数据发送给从设备，从设备通过 MISO 引脚回传数据。这意味全双工通信的数据输出和数据输入是用同一个时钟信号同步的；时钟信号由主设备通过 SCK 脚提供。

从选择（NSS）脚管理

有 2 种 NSS 模式：

- 软件 NSS 模式：可以通过设置 SPI_CR1 寄存器的 SSM 位来使能这种模式。在这种模式下 NSS 引脚可以用作它用，而内部 NSS 信号电平可以通过写 SPI_CR1 的 SSI 位来驱动
- 硬件 NSS 模式，分两种情况：
  - NSS 输出被使能：当 RX32S632 工作为主 SPI，并且 NSS 输出已经通过 SPI_CR2 寄存器的 SSOE 位使能，这时 NSS 引脚被拉低，所有 NSS 引脚与这个主 SPI 的 NSS 引脚相连并配置为硬件 NSS 的 SPI 设备，将自动变成从 SPI 设备。当一个 SPI 设备需要发送广播数据，它必须拉低 NSS 信号，以通知所有其它的设备它是主设备；如果它不能拉低 NSS，这意味着总线上有另外一个主设备在通信，这时将产生一个硬件错误。
  - NSS 输出被关闭：允许操作于多主环境。

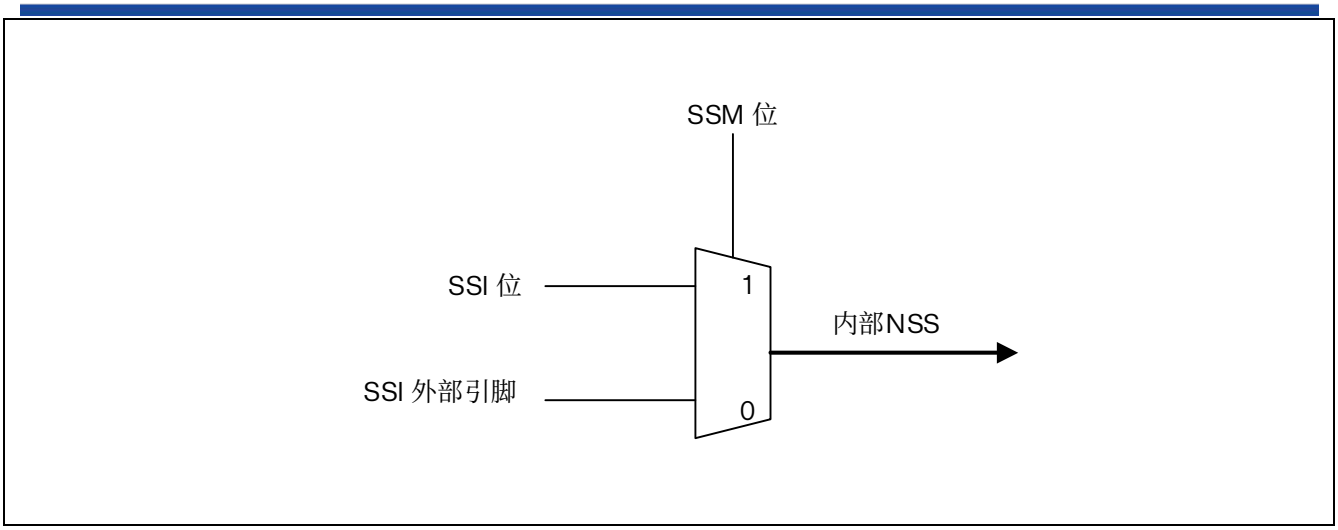


图 22.3 硬件/软件的从选择管理

时钟信号的相位和极性

SPI_CR 寄存器的 CPOL 和 CPHA 位，能够组合成四种可能的时序关系。CPOL（时钟极性）位控制在没有数据传输时时钟的空闲状态电平，此位对主模式和从模式下的设备都有效。如果 CPOL 被清 0，SCK 引脚在空闲状态保持低电平；如果 CPOL 被置‘1’，SCK 引脚在空闲状态保持高电平。

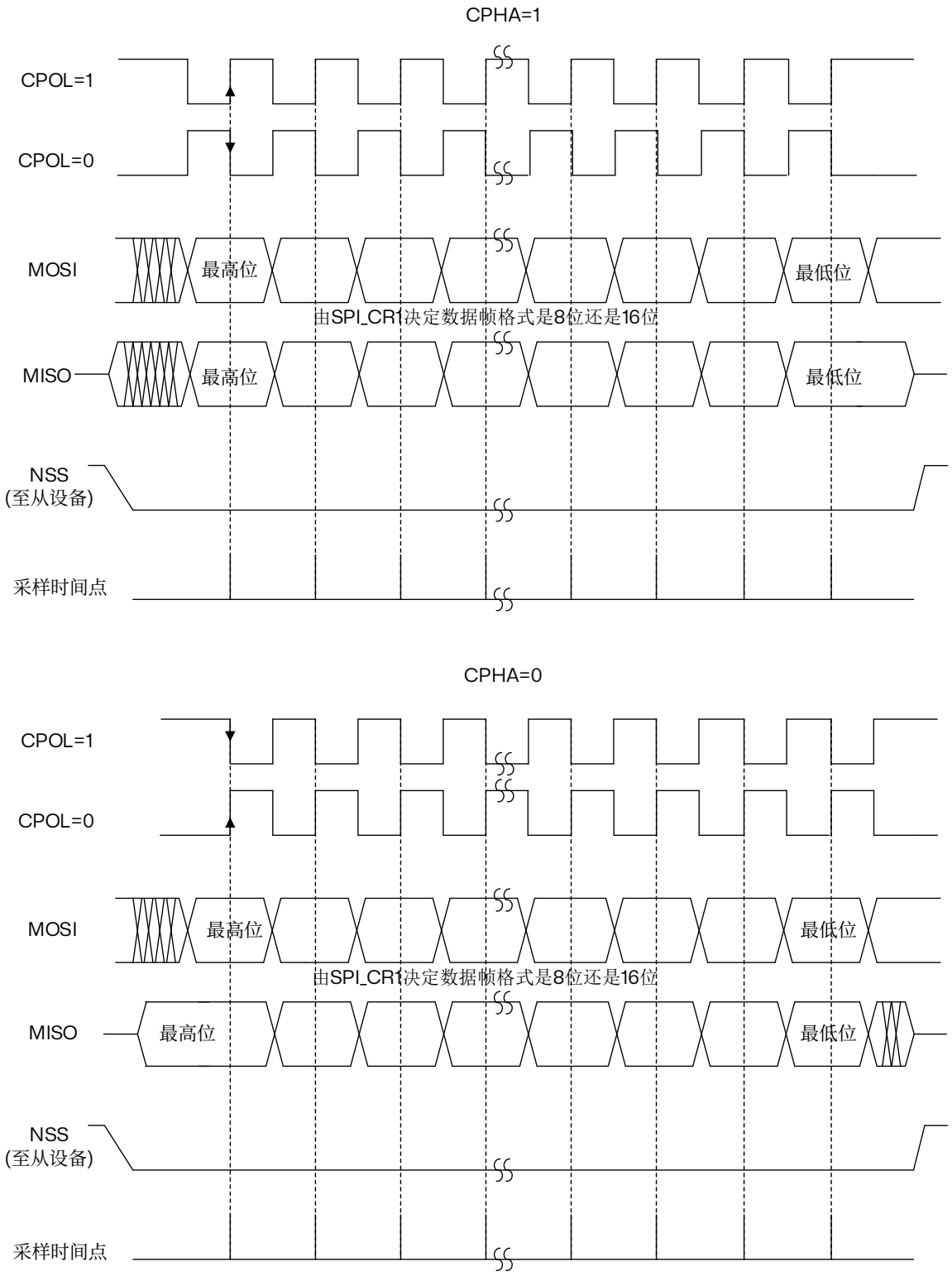
如果 CPHA（时钟相位）位被置‘1’，SCK 时钟的第二个边沿（CPOL 位为 0 时就是下降沿，CPOL 位为‘1’时就是上升沿）进行数据位的采样，数据在第二个时钟边沿被锁存。如果 CPHA 位被清‘0’，SCK 时钟的第一边沿（CPOL 位为‘0’时就是上升沿，CPOL 位为‘1’时就是下降沿）进行数据位采样，数据在第一个时钟边沿被锁存。

CPOL 时钟极性和 CPHA 时钟相位的组合选择数据捕捉的时钟边沿。

下图显示了 SPI 传输的 4 种 CPHA 和 CPOL 位组合。此图可以解释为主设备和从设备的 SCK 脚、MISO 脚、MOSI 脚直接连接的主或从时序图。

注意：

1. *在改变*CPOL/CPHA *位之前，必须清除*SPE *位将*SPI *禁止。*
2. *主和从必须配置成相同的时序模式。*
3. SCK *引脚空闲状态必须和*SPI_CR1 *寄存器指定的极性一致（即当*CPOL *为‘1’时，*SCK *引脚需要配置为上拉；*CPOL *为‘0’时，*SCK *引脚需要配置为下拉）。*
4. *数据帧格式（*8 *位或*16 *位）由*SPI_CR1 *寄存器的*DFF *位选择，并且决定发送/接收的数据长度。*



注意：这里显示的是SPI_CR1寄存器的LSBFIRST=0时的时序

图 22.4 数据时钟时序图

数据帧格式

根据 SPI_CR1 寄存器中的 LSBFIRST 位，输出数据位时可以 MSB 在先也可以 LSB 在先。

根据 SPI_CR1 寄存器的 DFF 位，每个数据帧可以是 8 位或是 16 位。所选择的数据帧格式对发送和/或接收都有效。

22.3.2 配置 SPI 为从模式

在从模式下，SCK 引脚用于接收从主设备来的串行时钟。SPI_CR1 寄存器中 BR[2:0]的设置不影响数据传输速率。

注：建议在主设备发送时钟之前使能 SPI 从设备，否则可能会发生意外的数据传输。在通信时钟的第一个边沿到来之前或正在进行的通信结束之前，从设备的数据寄存器必须就绪。在使能从设备和主设备之前，通信时钟的极性必须处于稳定的数值。 请按照以下步骤配置 SPI 为从模式：

配置步骤

1. 设置 DFF 位以定义数据帧格式为 8 位或 16 位。
2. 选择 CPOL 和 CPHA 位来定义数据传输和串行时钟之间的相位关系。为保证正确的数据传输，从设备和主设备的 CPOL 和 CPHA 位必须配置成相同的方式。
3. 帧格式（SPI_CR1 寄存器中的 LSBFIRST 位定义的”MSB 在前”还是”LSB 在前”）必须与主设备相同。
4. 硬件模式下（参考从选择（NSS）脚管理部分），在完整的数据帧（8 位或 16 位）传输过程中，NSS 引脚必须为低电平。在 NSS 软件模式下，设置 SPI_CR1 寄存器中的 SSM 位并清除 SSI 位。
5. 清除 MSTR 位、设置 SPE 位（SPI_CR1 寄存器），使相应引脚工作于 SPI 模式下。在这个配置中，MOSI 引脚是数据输入，MISO 引脚是数据输出。

数据发送过程

在写操作中，数据字被并行地写入发送缓冲器。

当从设备收到时钟信号，并且在 MOSI 引脚上出现第一个数据位时，发送过程开始（译注：此时第一个位被发送出去）。余下的位（对于 8 位数据帧格式，还有 7 位；对于 16 位数据帧格式，还有 15 位）被装进移位寄存器。当发送缓冲器中的数据传输到移位寄存器时，SPI_SP 寄存器的 TXE 标志被设置，如果设置了 SPI_CR2 寄存器的 TXEIE 位，将会产生中断。

数据接收过程

对于接收器，当数据接收完成时：

- 移位寄存器中的数据传送到接收缓冲器，SPI_SR 寄存器中的 RXNE 标志被设置。
- 如果设置了 SPI_CR2 寄存器中的 RXNEIE 位，则产生中断。

在最后一个采样时钟边沿后，RXNE 位被置‘1’，移位寄存器中接收到的数据字节被传送到接收缓冲器。当读 SPI_DR 寄存器时，SPI 设备返回这个接收缓冲器的数值。

读 SPI_DR 寄存器时，RXNE 位被清除。

22.3.3 配置 SPI 为主模式

在主配置时，在 SCK 脚产生串行时钟。

配置步骤

1. 通过 SPI_CR1 寄存器的 BR[2:0]位定义串行时钟波特率。
2. 选择 CPOL 和 CPHA 位，定义数据传输和串行时钟间的相位关系。
3. 设置 DFF 位来定义 8 位或 16 位数据帧格式。
4. 配置 SPI_CR1 寄存器的 LSBFIRST 位定义帧格式。
5. 如果需要 NSS 引脚工作在输入模式，硬件模式下，在整个数据帧传输期间应把 NSS 脚连接到高电平；在软件模式下，需设置 SPI_CR1 寄存器的 SSM 位和 SSI 位。如果 NSS 引脚工作在输出模式，则只需设置 SSOE 位。
6. 必须设置 MSTR 位和 SPE 位（只当 NSS 脚被连到高电平，这些位才能保持置位）。在这个配置中，MOSI 引脚是数据输出，而 MISO 引脚是数据输入。

数据发送过程

当写入数据至发送缓冲器时，发送过程开始。

在发送第一个数据位时，数据字被并行地（通过内部总线）传入移位寄存器，而后串行地移出到 MOSI 脚上；MSB 在先还是 LSB 在先，取决于 SPI_CR1 寄存器中的 LSBFIRST 位的设置。数据从发送缓冲器传输到移位寄存器时 TXE 标志将被置位，如果设置了 SPI_CR1 寄存器中的 TXEIE 位，将产生中断。

数据接收过程

对于接收器来说，当数据传输完成时：

- 传送移位寄存器里的数据到接收缓冲器，并且 RXNE 标志被置位。
- 如果设置了 SPI_CR2 寄存器中的 RXNEIE 位，则产生中断。

在最后采样时钟沿，RXNE 位被设置，在移位寄存器中接收到的数据字被传送到接收缓冲器。读 SPI_DR 寄存器时，SPI 设备返回接收缓冲器中的数据。

读 SPI_DR 寄存器将清除 RXNE 位。

一旦传输开始，如果下一个将发送的数据被放进了发送缓冲器，就可以维持一个连续的传输流。在试图写发送缓冲器之前，需确认 TXE 标志应该为‘1’。

注：在 NSS 硬件模式下，从设备的 NSS 输入由 NSS 引脚控制或另一个由软件驱动的 GPIO 引脚控制。

22.3.4 配置 SPI 通信方式

SPI 模块能够以两种方式通信：

- 半双工方式：1 条时钟线和 1 条双向数据线；

1 条时钟线和 1 条双向数据线（BIDIMODE=1）

设置 SPI_CR1 寄存器中的 BIDIMODE 位而启用此模式。在这个模式下，SCK 引脚作为时钟，主设备使用 MOSI 引脚而从设备使用 MISO 引脚作为数据通信。传输的方向由 SPI_CR1 寄存器里的 BIDIOE 控制，当这个位是‘1’的时候，数据线是输出，否则是输入。

- 单工方式：1 条时钟线和 1 条数据线（只接收或只发送）；

1 条时钟和 1 条单向数据线（BIDIMODE=0）

在这个模式下，SPI 模块可以或者作为只发送，或者作为只接收。

– 只发送模式类似于全双工模式（BIDIMODE=0，RXONLY=0）：数据在发送引脚（主模式时是 MOSI、从模式时是 MISO）上传输，而接收引脚（主模式时是 MISO、从模式时是 MOSI）可以作为通用的 I/O 使用。此时，软件不必理会接收缓冲器中的数据（如果读出数据

寄存器，它不 包含任何接收数据）。

– 在只接收模式，可以通过设置 SPI_CR2 寄存器的 RXONLY 位而关闭 SPI 的输出功能；此时，发送引脚（主模式时是 MOSI、从模式时是 MISO）被释放，可以作为其它功能使用。配置并使能 SPI 模块为只接收模式的方式是：
– 在主模式时，一旦使能 SPI，通信立即启动，当清除 SPE 位时立即停止当前的接收。在此模式下，不必读取 BSY 标志，在 SPI 通信期间这个标志始终为‘1’。
– 在从模式时，只要 NSS 被拉低（或在 NSS 软件模式时，SSI 位为‘0’）同时 SCK 有时钟脉冲，SPI 就一直在接收。

22.3.5 数据发送与接收过程

接收与发送缓冲器

在接收时，接收到的数据被存放在一个内部的接收缓冲器中；在发送时，在被发送之前，数据将首先被存放在一个内部的发送缓冲器中。

对 SPI_DR 寄存器的读操作，将返回接收缓冲器的内容；写入 SPI_DR 寄存器的数据将被写入发送缓冲器中。

主模式下开始传输

- 全双工模式（BIDIMODE=0 并且 RXONLY=0）
  – 当写入数据到 SPI_DR 寄存器（发送缓冲器）后，传输开始；
  – 在传送第一位数据的同时，数据被并行地从发送缓冲器传送到 8 位的移位寄存器中，然后按顺序被串行地移位送到 MOSI 引脚上；
  – 与此同时，在 MISO 引脚上接收到的数据，按顺序被串行地移位进入 8 位的移位寄存器中，然后被并行地传送到 SPI_DR 寄存器（接收缓冲器）中。
- 单向的只接收模式（BIDIMODE=0 并且 RXONLY=1）
  – SPE=1 时，传输开始；
  – 只有接收器被激活，在 MISO 引脚上接收到的数据，按顺序被串行地移位进入 8 位的移位寄存器中，然后被并行地传送到 SPI_DR 寄存器（接收缓冲器）中。
- 双向模式，发送时（BIDIMODE=1 并且 BIDIOE=1）
  – 当写入数据到 SPI_DR 寄存器（发送缓冲器）后，传输开始；
  – 在传送第一位数据的同时，数据被并行地从发送缓冲器传送到 8 位的移位寄存器中，然后按顺序被串行地移位送到 MOSI 引脚上；
  – 不接收数据。
- 双向模式，接收时（BIDIMODE=1 并且 BIDIOE=0）
  – SPE=1 并且 BIDIOE=0 时，传输开始；
  – 在 MOSI 引脚上接收到的数据，按顺序被串行地移位进入 8 位的移位寄存器中，然后被并行地传送到 SPI_DR 寄存器（接收缓冲器）中。
  – 不激活发送器，没有数据被串行地送到 MOSI 引脚上。

从模式下开始传输

- 全双工模式（BIDIMODE=0 并且 RXONLY=0）
  – 当从设备接收到时钟信号并且第一个数据位出现在它的 MOSI 时，数据传输开始，随后的数据位依次移动进入移位寄存器；

  - 与此同时，在传输第一个数据位时，发送缓冲器中的数据被并行地传送到 8 位的移位寄存器，随后被串行地发送到 MISO 引脚上。软件必须保证在 SPI 主设备开始数据传输之前在发送寄存器中写入要发送的数据。
- 单向的只接收模式（BIDIMODE=0 并且 RXONLY=1）
  - 当从设备接收到时钟信号并且第一个数据位出现在它的 MOSI 时，数据传输开始，随后数据位依次移动进入移位寄存器；
  - 不启动发送器，没有数据被串行地传送到 MISO 引脚上。
- 双向模式，发送时（BIDIMODE=1 并且 BIDIOE=1）
  - 当从设备接收到时钟信号并且发送缓冲器中的第一个数据位被传送到 MISO 引脚上的时候，数据传输开始；
  - 在第一个数据位被传送到 MISO 引脚上的同时，发送缓冲器中要发送的数据被平行地 传送到 8 位的移位寄存器中，随后被串行地发送到 MISO 引脚上。软件必须保证在 SPI 主设备开始数据传输之前在发送寄存器中写入要发送的数据；
  - 不接收数据。
- 双向模式，接收时（BIDIMODE=1 并且 BIDIOE=0）
  - 当从设备接收到时钟信号并且第一个数据位出现在它的 MOSI 时，数据传输开始；
  - 从 MISO 引脚上接收到的数据被串行地传送到 8 位的移位寄存器中，然后被平行地传送到 SPI_DR 寄存器（接收缓冲器）；
  - 不启动发送器，没有数据被串行地传送到 MISO 引脚上。

处理数据的发送与接收

当数据从发送缓冲器传送到移位寄存器时，设置 TXE 标志（发送缓冲器空），它表示内部的发送缓冲器可以接收下一个数据；如果在 SPI_CR2 寄存器中设置了 TXEIE 位，则此时会产生一个中断；写入 SPI_DR 寄存器即可清除 TXE 位。

注：在写入发送缓冲器之前，软件必须确认 TXE 标志为‘1’，否则新的数据会覆盖已经在发送缓冲器中的数据。

在采样时钟的最后一个边沿，当数据被从移位寄存器传送到接收缓冲器时，设置 RXNE 标志（接收缓冲器非空）；它表示数据已经就绪，可以从 SPI_DR 寄存器读出；如果在 SPI_CR2 寄存器中设置了 RXNEIE 位，则此时会产生一个中断；读出 SPI_DR 寄存器即可清除 RXNIE 标志位。在一些配置中，传输最后一个数据时，可以使用 BSY 标志等待数据传输的结束。

主或从模式下（BIDIMODE=0 并且 RXONLY=0）全双工发送和接收过程模式

软件必须遵循下述过程，发送和接收数据：

1. 设置 SPE 位为‘1’，使能 SPI 模块；
2. 在 SPI_DR 寄存器中写入第一个要发送的数据，这个操作会清除 TXE 标志；
3. 等待 TXE=1，然后写入第二个要发送的数据。等待 RXNE=1，然后读出 SPI_DR 寄存器并获得第一个接收到的数据，读 SPI_DR 的同时清除了 RXNE 位。重复这些操作，发送后续的数据同时接收 n-1 个数据；
4. 等待 RXNE=1，然后接收最后一个数据；
5. 等待 TXE=1，在 BSY=0 之后关闭 SPI 模块。也可以在响应 RXNE 或 TXE 标志的上升沿产生的中断的处理程序中实现这个过程。

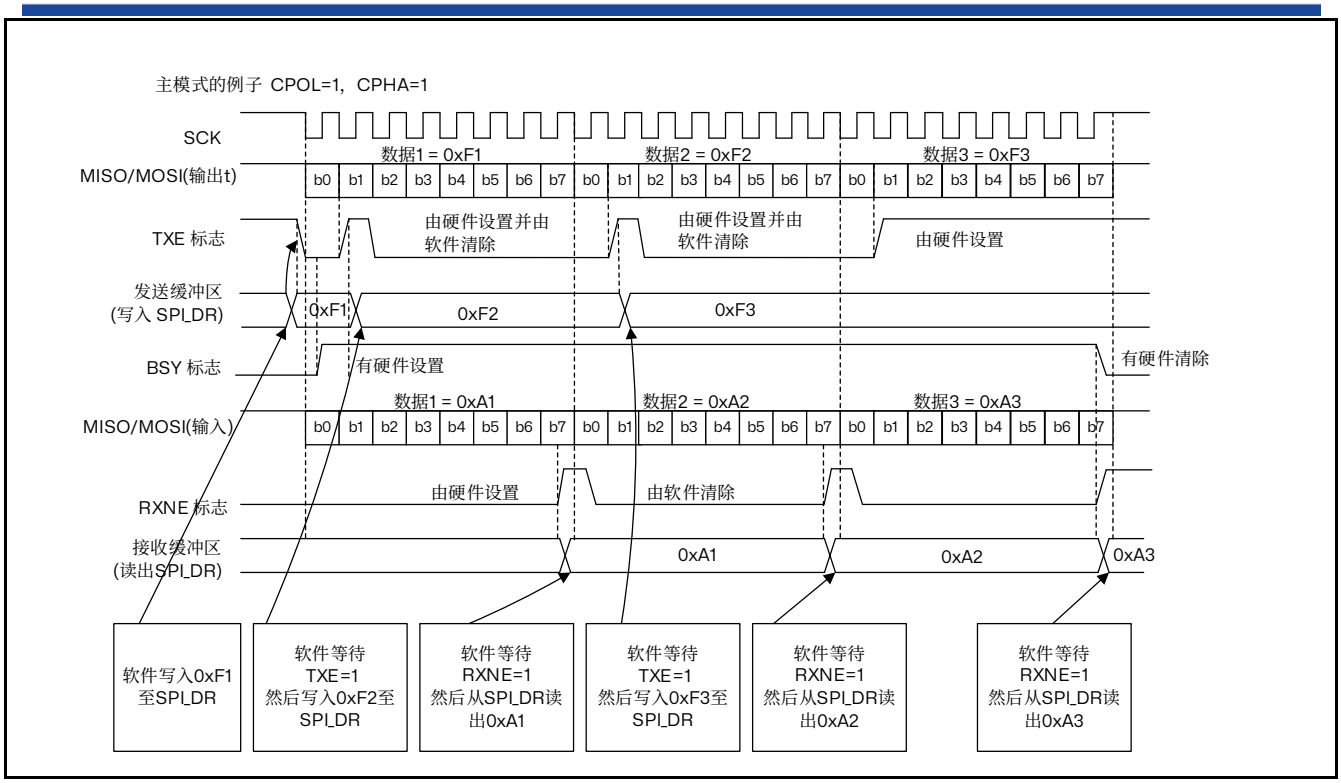


图 22.5 主模式、全双工模式下（BIDIMODE=0 并且 RXONLY=0）连续传输时，TXE/RXNE/BSY 的变化示意图

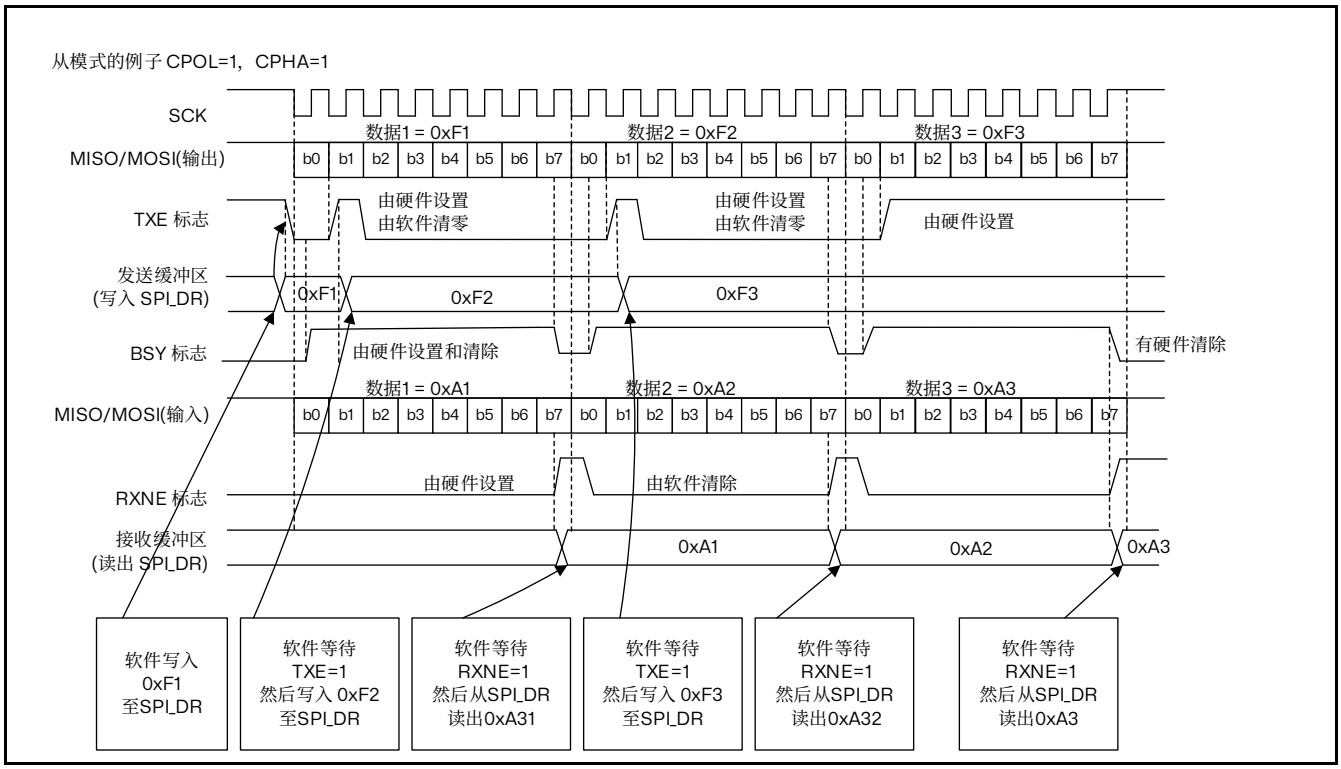


图 22.6 从模式、全双工模式下（BIDIMODE=0 并且 RXONLY=0）连续传输时，TXE/RXNE/BSY 的变化示意图

只发送过程（BIDIMODE=0 并且 RXONLY=0）

在此模式下，传输过程可以简要说明如下，使用 BSY 位等待传输的结束：

1. 设置 SPE 位为‘1’，使能 SPI 模块；
2. 在 SPI_DR 寄存器中写入第一个要发送的数据，这个操作会清除 TXE 标志；
3. 等待 TXE=1，然后写入第二个要发送的数据。重复这个操作，发送后续的数据；
4. 写入最后一个数据到 SPI_DR 寄存器之后，等待 TXE=1；然后等待 BSY=0，这表示最后一个数据的传输已经完成。也可以在响应 TXE 标志的上升沿产生的中断的处理程序中实现这个过程。

注：1. 对于不连续的传输，在写入 SPI_DR 寄存器的操作与设置 BSY 位之间有 2 个 APB 时钟周期的延迟，因此在只发送模式下，写入最后一个数据后，最好先等待 TXE=1，然后再等待 BSY=0。

2. 只发送模式下，在传输 2 个数据之后，由于不会读出接收到的数据，SPI_SR 寄存器中的 OVR 位‘1’。（译注：软件不必理会这个 OVR 标志位）

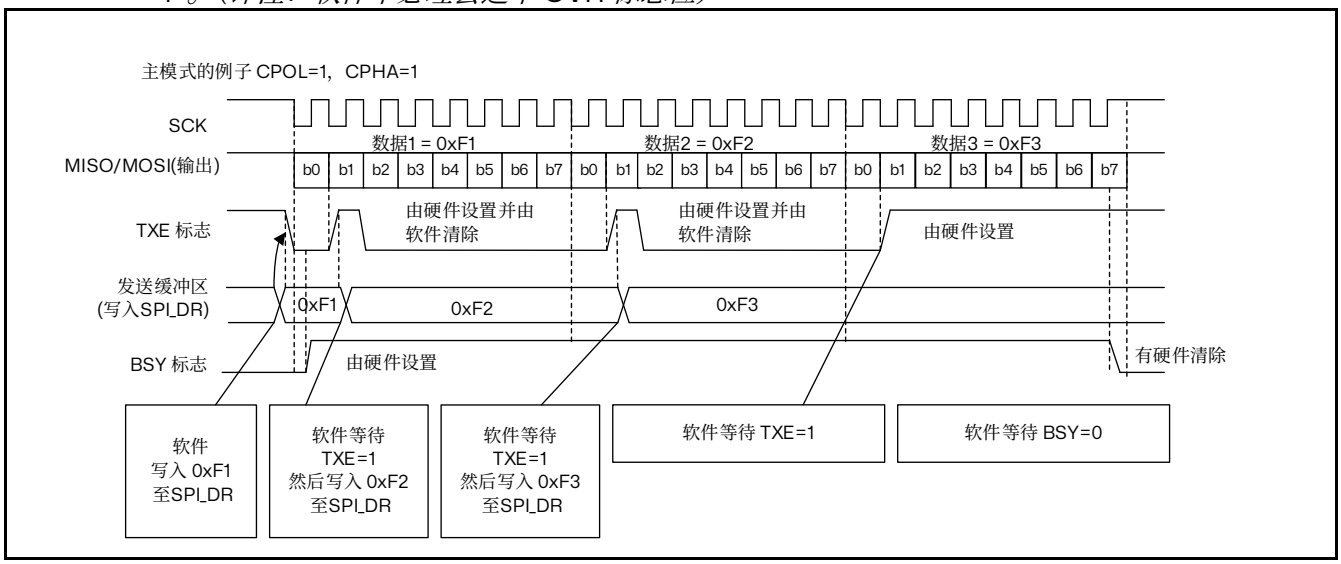


图 22.7 主设备只发送模式（BIDIMODE=0 并且 RXONLY=0）下连续传输时，TXE/BSY 变化示意图

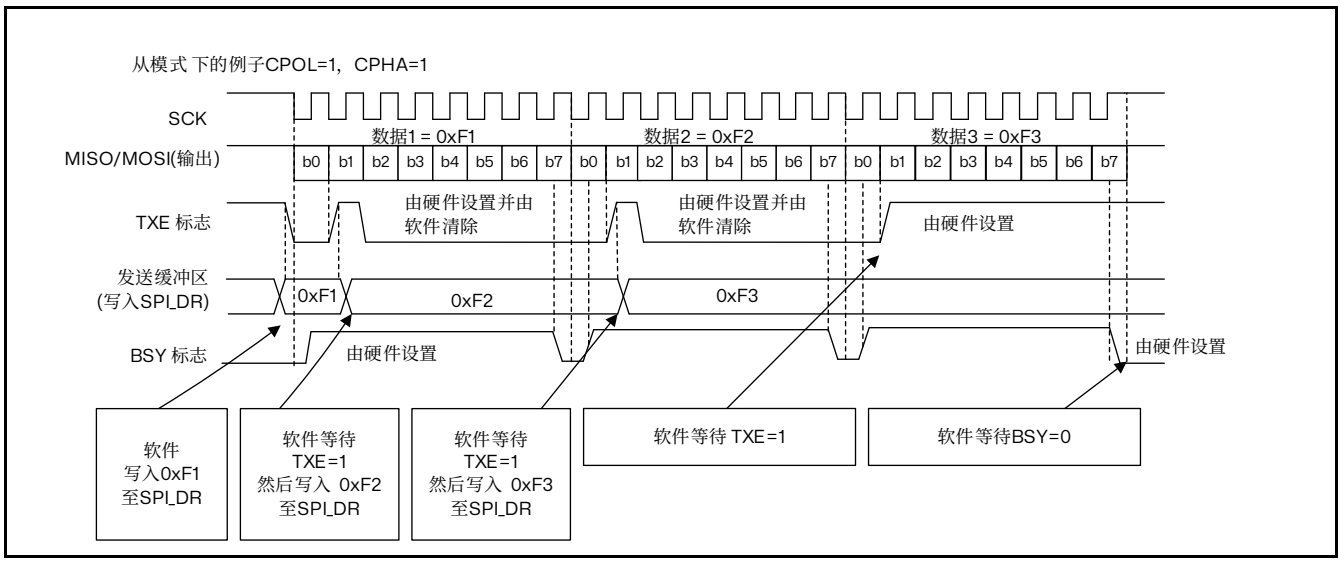


图 22.8 从设备只发送模式（BIDIMODE=0 并且 RXONLY=0）下连续传输时，TXE/BSY 变化示意图

双向发送过程（BIDIMODE=1 并且 BIDIOE=1）

在此模式下，操作过程类似于只发送模式，不同的是：在使能 SPI 模块之前，需要在 SPI_CR2 寄存器中同时设置 BIDIMODE 和 BIDIOE 位为‘1’。

单向只接收模式（BIDIMODE=0 并且 RXONLY=1）

在此模式下，传输过程可以简要说明如下：

1. 在 SPI_CR2 寄存器中，设置 RXONLY=1;
2. 设置 SPE=1，使能 SPI 模块：
   a）主模式下，立刻产生 SCK 时钟信号，在关闭 SPI（SPE=0）之前，不断地接收串行数据；
   b）从模式下，当 SPI 主设备拉低 NSS 信号并产生 SCK 时钟时，接收串行数据。
3. 等待 RXNE=1，然后读出 SPI_DR 寄存器以获得收到的数据（同时会清除 RXNE 位）。重复这个操作接收所有数据。

也可以在响应 RXNE 标志的上升沿产生的中断的处理程序中实现这个过程。

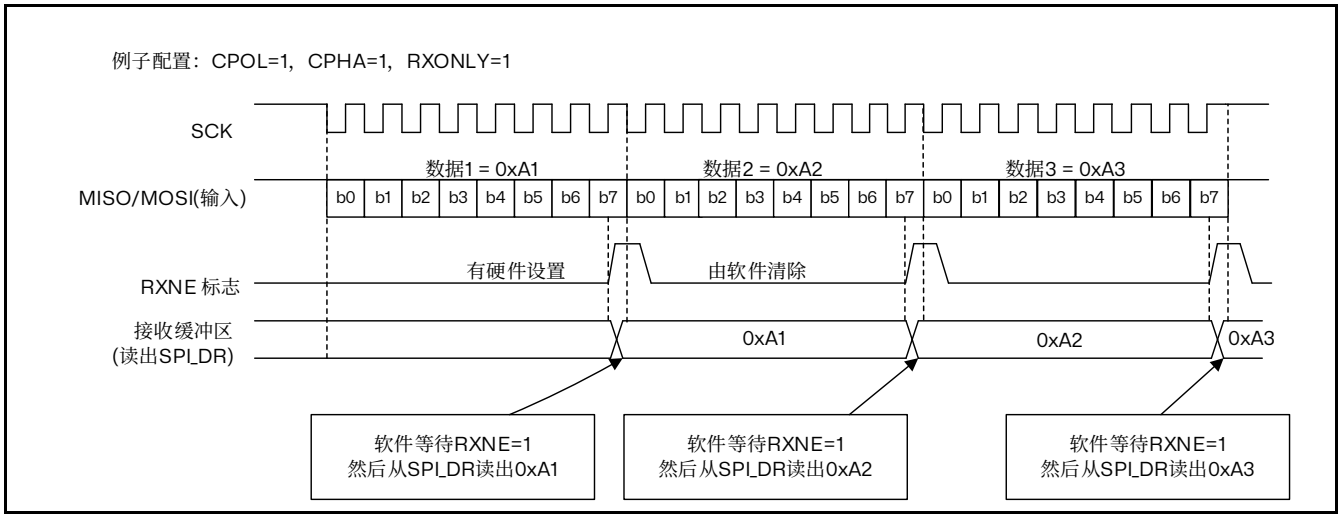


图 22.9 只接收模式（BIDIMODE=0 并且 RXONLY=1）下连续传输时，RXNE 变化示意图

单向接收过程（BIDIMODE=1 并且 BIDIOE=0）

在此模式下，操作过程类似于只接收模式，不同的是：在使能 SPI 模块之前，需要在 SPI_CR2 寄存器中设置 BIDIMODE 为‘1’并清除 BIDIOE 位为‘0’。

连续和非连续传输

当在主模式下发送数据时，如果软件足够快，能够在检测到每次 TXE 的上升沿（或 TXE 中断），并立即在正在进行的传输结束之前写入 SPI_DR 寄存器，则能够实现连续的通信；此时，在每个数据项的传输之间的 SPI 时钟保持连续，同时 BSY 位不会被清除。

如果软件不够快，则会导致不连续的通信；这时，在每个数据传输之间会被清除（见下图）。

在主模式的只接收模式下（RXONLY=1），通信总是连续的，而且 BSY 标志始终为‘1’。

在从模式下，通信的连续性由 SPI 主设备决定。不管怎样，即使通信是连续的，BSY 标志会在每个数据项之间至少有一个 SPI 时钟周期为低。

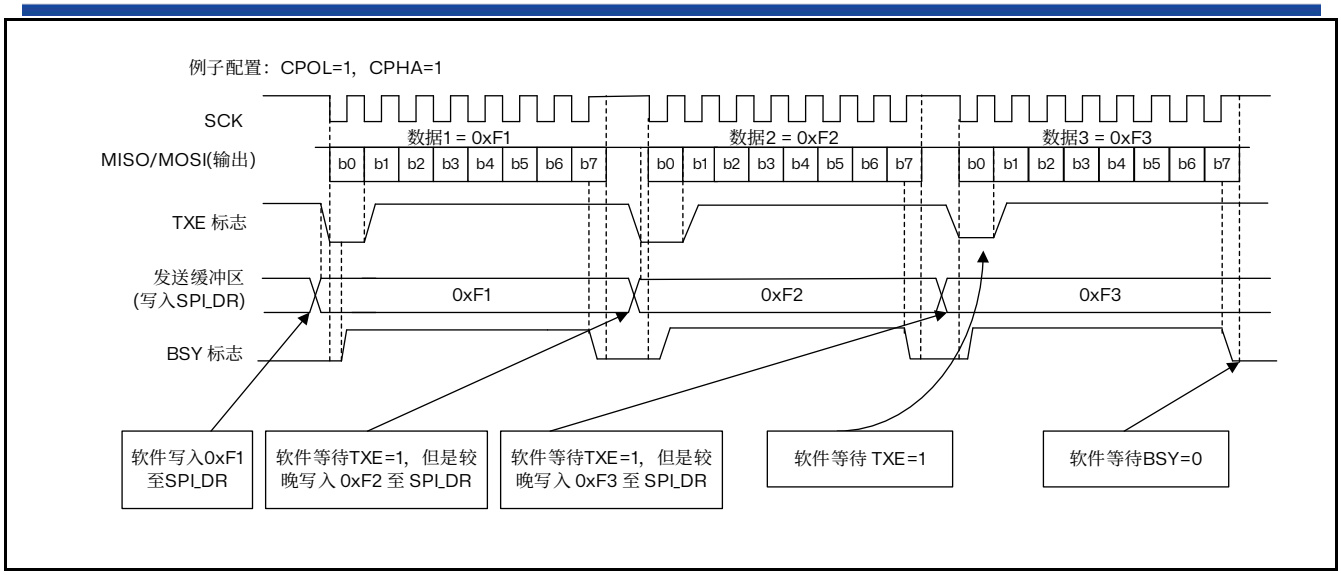


图 22.10 非连续传输发送（BIDIMODE=0 并且 RXONLY=0）时，TXE/BSY 变化示意图

22.3.6 状态标志

应用程序通过 3 个状态标志可以完全监控 SPI 总线的状态。

发送缓冲器空闲标志（TXE）

此标志为‘1’时表明发送缓冲器为空，可以写下一个待发送的数据进入缓冲器中。当写入 SPI_DR 时，TXE 标志被清除。

接收缓冲器非空（RXNE）

此标志为‘1’时表明在接收缓冲器中包含有效的接收数据。读 SPI 数据寄存器可以清除此标志。

忙（Busy）标志

BSY 标志由硬件设置与清除（写入此位无效果），此标志表明 SPI 通信层的状态。

当它被设置为‘1’时，表明 SPI 正忙于通信，但有一个例外：在主模式的双向接收模式下（MSTR=1、BDM=1 并且 BDOE=0），在接收期间 BSY 标志保持为低。

在软件要关闭 SPI 模块并进入停机模式（或关闭设备时钟）之前，可以使用 BSY 标志检测传输是否结束，这样可以避免破坏最后一次传输，因此需要严格按照下述过程执行。

BSY 标志还可以用于在多主系统中避免写冲突。

除了主模式的双向接收模式（MSTR=1、BDM=1 并且 BDOE=0），当传输开始时，BSY 标志被置‘1’。

以下情况时此标志将被清除为‘0’：

- 当传输结束（主模式下，如果是连续通信的情况例外）；
- 当关闭 SPI 模块；
- 当产生主模式失效（MODF=1）。 如果通信不是连续的，则在每个数据项的传输之间，BSY 标志为低。

当通信是连续时：

- 主模式下：在整个传输过程中，BSY 标志保持为高；
- 从模式下：在每个数据项的传输之间，BSY 标志在一个 SPI 时钟周期中为低。

*注：不要使用*BSY *标志处理每一个数据项的发送和接收，最好使用*TXE *和*RXNE *标志。*

22.3.7 关闭 SPI

当通讯结束，可以通过关闭 SPI 模块来终止通讯。清除 SPE 位即可关闭 SPI。

在某些配置下，如果再传输还未完成时，就关闭 SPI 模块并进入停机模式，则可能导致当前的传输被破坏，而且 BSY 标志也变得不可信。

为了避免发生这种情况，关闭 SPI 模块时，建议按照下述步骤操作：

在主或从模式下的全双工模式（BIDIMODE=0，RXONLY=0）

1. 等待 RXNE=1 并接收最后一个数据；
2. 等待 TXE=1；
3. 等待 BSY=0；
4. 关闭 SPI（SPE=0），最后进入停机模式（或关闭该模块的时钟）。

在主或从模式下的单向只发送模式（BIDIMODE=0，RXONLY=0）或双向的发送模式（BIDIMODE=1，BIDIOE=1）

在 SPI_DR 寄存器中写入最后一个数据后：

1. 等待 TXE=1；
2. 等待 BSY=0；
3. 关闭 SPI（SPE=0），最后进入停机模式（或关闭该模块的时钟）。

在主或从模式下的单向只接收模式（MSTR=1，BIDIMODE=0，RXONLY=1）或双向的接收模式（MSTR=1，BIDIMODE=1，BIDIOE=0）

这种情况需要特别地处理，以保证 SPI 不会开始一次新的传输：

1. 等待倒数第二个（第 n-1 个）RXNE=1；
2. 在关闭 SPI（SPE=0）之前等待一个 SPI 时钟周期（使用软件延迟）；
3. 在进入停机模式（或关闭该模块的时钟）之前等待最后一个 RXNE=1。

注：在主模式下的单向只发送模式（MSTR=1，BDM=1，BDOE=0）时，传输过程中 BSY 标志始终为低。

在从模式下的只接收模式（MSTR=0，BIDIMODE=0，RXONLY=1）或双向的接收模式（MSTR=0，BIDIMODE=1，BIDIOE=0）

1. 可以在任何时候关闭 SPI（SPE=0），SPI 会在当前的传输结束后被关闭；
2. 如果希望进入停机模式，在进入停机模式（或关闭该模块的时钟）之前必须首先等待 BSY=0。

22.3.8 错误标志

主模式失效错误（MODF）

主模式失效仅发生在：NSS 引脚硬件模式管理下，主设备的 NSS 脚被拉低；或者在 NSS 引脚软件模式管理下，SSI 位被置为‘0’时；MODF 位被自动置位。主模式失效对 SPI 设备有以下影响：

- MODF 位被置为‘1’，如果设置了 ERRIE 位，则产生 SPI 中断；
- SPE 位被清为‘0’。这将停止一切输出，并且关闭 SPI 接口；
- MSTR 位被清为‘0’，因此强迫此设备进入从模式

下面的步骤用于清除 MODF 位：

1. 当 MODF 位被置为‘1’时，执行一次对 SPI_SR 寄存器的读或写操作；
2. 然后写 SPI_CR1 寄存器。

在有多个 MCU 的系统中，为了避免出现多个从设备的冲突，必须先拉高该主设备的 NSS 脚，再对 MODF 位进行清零。在完成清零之后，SPE 和 MSTR 位可以恢复到它们的原始状态。

出于安全的考虑，当 MODF 位为‘1’时，硬件不允许设置 SPE 和 MSTR 位。

通常配置下，从设备的 MODF 位不能被置为‘1’。然而，在多主配置里，一个设备可以在设置了 MODF 位的情况下，处于从设备模式；此时，MODF 位表示可能出现了多主冲突。中断程序可以执行一个复位或返回到默认状态来从错误状态中恢复。

溢出错误

当主设备已经发送了数据字节，而从设备还没有清除前一个数据字节产生的 RXNE 时，即为溢出错误。当产生溢出错误时：

- OVR 位被置为‘1’；当设置了 ERRIE 位时，则产生中断。

此时，接收器缓冲器的数据不是主设备发送的新数据，读 SPI_DR 寄存器返回的是之前未读的数据，所有随后传送的数据都被丢弃。

依次读出 SPI_DR 寄存器和 SPI_SR 寄存器可将 OVR 清除。

22.3.9 SPI 中断

表 22.1 SPI 中断请求

中断事件	事件标志	使能控制位
发送缓冲器空标志	TXE	TXEIE
接收缓冲器非空标志	RXNE	RXNEIE
主模式失效事件	MODF	ERRIE
溢出错误	OVR	

22.4 SPI 寄存器

访问：无等待，支持字，半字和字节访问

22.4.1 SPI 控制寄存器 1（SPI_CR1）

地址偏移：0x00

复位值：0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIDI MODE	BIDIOE	Reserved		DFF	RX ONLY	SSM	SSI	LSB FIRST	SPE	BR[2:0]			MSTR	CPOL	CPHA
rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit 15 **BIDIMODE**：双向数据模式使能（Bidirectional data mode enable）

0：选择“双线双向”模式；

1：选择“单线双向”模式。

Bit 14 **BIDIOE**：双向模式下的输出使能（Output enable in bidirectional mode）

和 BIDIMODE 位一起决定在“单线双向”模式下数据的输出方向

0：输出禁止（只收模式）；

1：输出使能（只发模式）。

这个“单线”数据线在主设备端为 MOSI 引脚，在从设备端为 MISO 引脚。

Bits 13:12 保留，必须保持为复位值。

Bit 11 **DFF**：数据帧格式（Data frame format）

0：使用 8 位数据帧格式进行发送/接收；
1：使用 16 位数据帧格式进行发送/接收。
注：只有当 SPI 禁止（SPE=0）时，才能写该位，否则出错。

Bit 10 **RXONLY**：只接收（Receive only）
该位和 BIDIMODE 位一起决定在“双线双向”模式下的传输方向。在多个从设备的配置中，在未被访问的从设备上该位被置 1，使得只有被访问的从设备有输出，从而不会造成数据线上数据冲突。
0：全双工（发送和接收）；
1：禁止输出（只接收模式）。

Bit 9 **SSM**：软件从设备管理 （Software slave management）
当 SSM 被置位时，NSS 引脚上的电平由 SSI 位的值决定。
0：禁止软件从设备管理；
1：使能软件从设备管理。

Bit 8 **SSI**：内部从设备选择（Internal slave select）
该位只在 SSM 位为‘1’时有意义。它决定了 NSS 上的电平，在 NSS 引脚上的 I/O 操作无效。

Bit 7 **LSBFIRST**：帧格式（Frame format）
0：先发送 MSB；
1：先发送 LSB。
注：当通信在进行时不能改变该位的值。

Bit 6 **SPE**：SPI 使能（SPI enable）
0：禁止 SPI 设备；
1：使能 SPI 设备。

Bits 5:3 **BR[2:0]**：波特率控制（Baud rate control）
000：$f_{PCLK}/2$
001：$f_{PCLK}/4$
010：$f_{PCLK}/8$
011：$f_{PCLK}/16$
100：$f_{PCLK}/32$
101：$f_{PCLK}/64$
110：$f_{PCLK}/128$
111：$f_{PCLK}/256$
当通信正在进行的时候，不能修改这些位。

Bit 2 **MSTR**：主设备选择（Master selection）
0：配置为从设备；
1：配置为主设备。
注：当通信正在进行的时候，不能修改该位。

Bit 1 **CPOL**：时钟极性（Clock polarity）
0：空闲状态时，SCK 保持低电平；
1：空闲状态时，SCK 保持高电平。
注：当通信正在进行的时候，不能修改该位。

Bit 0 **CPHA**：时钟相位（Clock phase）
0：数据采样从第一个时钟边沿开始；

1：数据采样从第二个时钟边沿开始。

注：当通信正在进行的时候，不能修改该位。

22.4.2 SPI 控制寄存器 2（SPI_CR2）

地址偏移：0x04

复位值：0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TXEIE	RXNEIE	ERRIE	Res		SSOE	Res	OSCSEL
								rw	rw	rw			rw		rw

Bits 15:8 保留，必须保持为复位值。

Bit 7 **TXEIE**：发送缓冲区空中断使能（Tx buffer empty interrupt enable）

0：禁止 TXE 中断；

1：使能 TXE 中断，当 TXE 标志置位为‘1’时产生中断请求。

Bit 6 **RXNEIE**：接收缓冲区非空中断使能（RX buffer not empty interrupt enable）

0：禁止 RXNE 中断；

1：使能 RXNE 中断，当 RXNE 标志置位时产生中断请求。

Bit 5 **ERRIE**：错误中断使能（Error interrupt enable）

当错误（OVR、MODF）产生时，该位控制是否产生中断

0：禁止错误中断；

1：使能错误中断。

Bits 4:3 保留，必须保持为复位值。

Bit 2 **SSOE**：SS 输出使能（SS output enable）

0：禁止在主模式下 SS 输出，该设备可以工作在多主设备模式；

1：设备开启时，开启主模式下 SS 输出，该设备不能工作在多主设备模式。

Bit 1 保留，必须保持为复位值。

Bit 0 **OSCSEL**：有源晶振选择（Oscillator select）

0：基本时钟源选择 HSI

1：基本时钟源切换至有源晶振

22.4.3 SPI 状态寄存器（SPI_SR）

地址偏移：0x08

复位值：0x0000 0002

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BSY	OVR	MODF	Res.	UDR	CHSIDE	TXE	RXNE
								r	r	r		r	r	r	r

Bits 15:8 保留，必须保持为复位值。

Bit 7 **BSY**：忙标志（Busy flag）

0：SPI 不忙；

1：SPI 正忙于通信，或者发送缓冲非空。

该位由硬件置位或者复位。

Bit 6 **OVR**：溢出标志（Overrun flag）

0：没有出现溢出错误；

1：出现溢出错误。

该位由硬件置位；通过先读 SPI_SR 再读 SPI_DR，来清除该标志位。

Bit 5 **MODF：**模式错误（Mode fault）

0：没有出现模式错误；

1：出现模式错误。

该位由硬件置位；通过先读或写 SPI_SR 再写 SPI_CR1，来清除该标志位。

Bit 4 保留，必须保持为复位值。

Bit 3 **UDR：**下溢标志位（Underrun flag）

0：未发生下溢；

1：发生下溢。

该标志位由硬件置‘1’，由一个软件序列清‘0’。

注：在 SPI 模式下不使用。

Bit 2 **CHSIDE：**声道（Channel side）

0：需要传输或者接收左声道；

1：需要传输或者接收右声道。

注：在 SPI 模式下不使用。在 PCM 模式下无意义。

Bit 1 **TXE：**发送缓冲为空（Transmit buffer empty）

0：发送缓冲非空；

1：发送缓冲为空。

Bit 0 **RXNE：**接收缓冲非空（Receive buffer not empty）

0：接收缓冲为空；

1：接收缓冲非空。

22.4.4 SPI 数据寄存器（SPI_DR）

地址偏移：0x0C

复位值：0x0000

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

Bits 15:0 **DR[15:0]：**数据寄存器（Data register）

待发送或者已经收到的数据

数据寄存器对应两个缓冲区：一个用于写（发送缓冲）；另外一个用于读（接收缓冲）。写操作将数据写到发送缓冲区；读操作将返回接收缓冲区里的数据。

对 SPI 模式的注释：根据 SPI_CR1 的 DFF 位对数据帧格式的选择，数据的发送和接收可以是 8 位或者 16 位的。为保证正确的操作，需要在启用 SPI 之前就确定好数据帧格式。

对于 8 位的数据，缓冲器是 8 位的，发送和接收时只会用到 SPI_DR[7:0]。在接收时，SPI_DR[15:8]被强制为 0。

对于 16 位的数据，缓冲器是 16 位的，发送和接收时会用到整个数据寄存器，即 SPI_DR[15:0]。

23 I2C 接口（I2C）

23.1 I²C 简介

I²C（Inter-Integrated Circuit，内置集成电路）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。支持标准和快速两种模式。

I²C 模块有多种用途，包括 CRC 码的生成和校验和 PMBus（电源管理总线–Power Management Bus）。

23.2 I²C 主要特点

- 并行总线/I²C 总线协议转换器
- 多主机功能：该模块既可做主设备也可做从设备
- I²C 主设备功能
  - 产生时钟
  - 产生起始和停止信号
- I²C 从设备功能
  - 可编程的 I²C 地址检测
  - 可响应 2 个从地址的双地址能力
  - 停止位检测
- 产生和检测 7 位/10 位地址和广播呼叫
- 支持不同的通讯速度
  - 标准速度（高达 100 KHz）
  - 快速（高达 400 KHz）
- 状态标志：
  - 发送器/接收器模式标志
  - 字节发送结束标志
  - I²C 总线忙标志
- 错误标志
  - 主模式时的仲裁丢失
  - 地址/数据传输后的应答（ACK）错误
  - 检测到错位的起始或停止条件
  - 禁止拉长时钟功能时的上溢或下溢
- 2 个中断向量
  - 1 个中断用于地址/数据通讯成功
  - 1 个中断用于错误
- 可选的拉长时钟功能

注：不是所有产品中都包含上述所有特性。请参考相关的数据手册，确认该产品支持的 I²C 功能。

23.3 I²C 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，或并行转换成串行。可以开启或禁止中断。接口通过数据引脚（SDA）和时钟引脚（SCL）连接到 I²C 总线。允许连接到标准（高达 100KHz）或快速（高达 400KHz）的 I²C 总线。

23.3.1 模式选择

接口可以下述 4 种模式中的一种运行:

- 从发送器模式
- 从接收器模式
- 主发送器模式
- 主接收器模式

该模块默认地工作于从模式。接口在生成起始条件后自动地从从模式切换到主模式；当仲裁丢失或产生停止信号时，则从主模式切换到从模式。允许多主机功能。

通信流

主模式时，I²C 接口启动数据传输并产生时钟信号。串行数据传输总是以起始条件开始并以停止条件结束。起始条件和停止条件都是在主模式下由软件控制产生。

从模式时，I²C 接口能识别它自己的地址（7 位或 10 位）和广播呼叫地址。软件能够控制开启或禁止广播呼叫地址的识别。

数据和地址按 8 位/字节进行传输，高位在前。跟在起始条件后的 1 或 2 个字节是地址（7 位模式为 1 个字节，10 位模式为 2 个字节）。地址只在主模式发送。

在一个字节传输的 8 个时钟后的第 9 个时钟期间，接收器必须回送一个应答位（ACK）给发送器。参考下图。

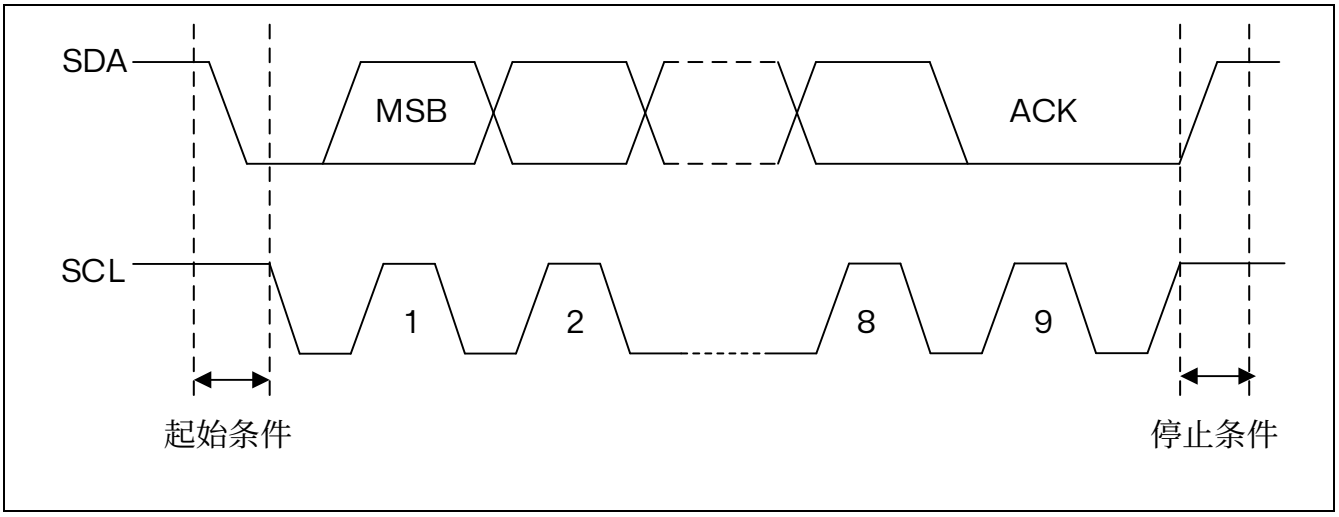


图 23.1 I²C 总线协议

软件可以开启或禁止应答（ACK），并可以设置 I²C 接口的地址（7 位、10 位地址或广播呼叫地址）。

I²C 接口的功能框图示于下图。

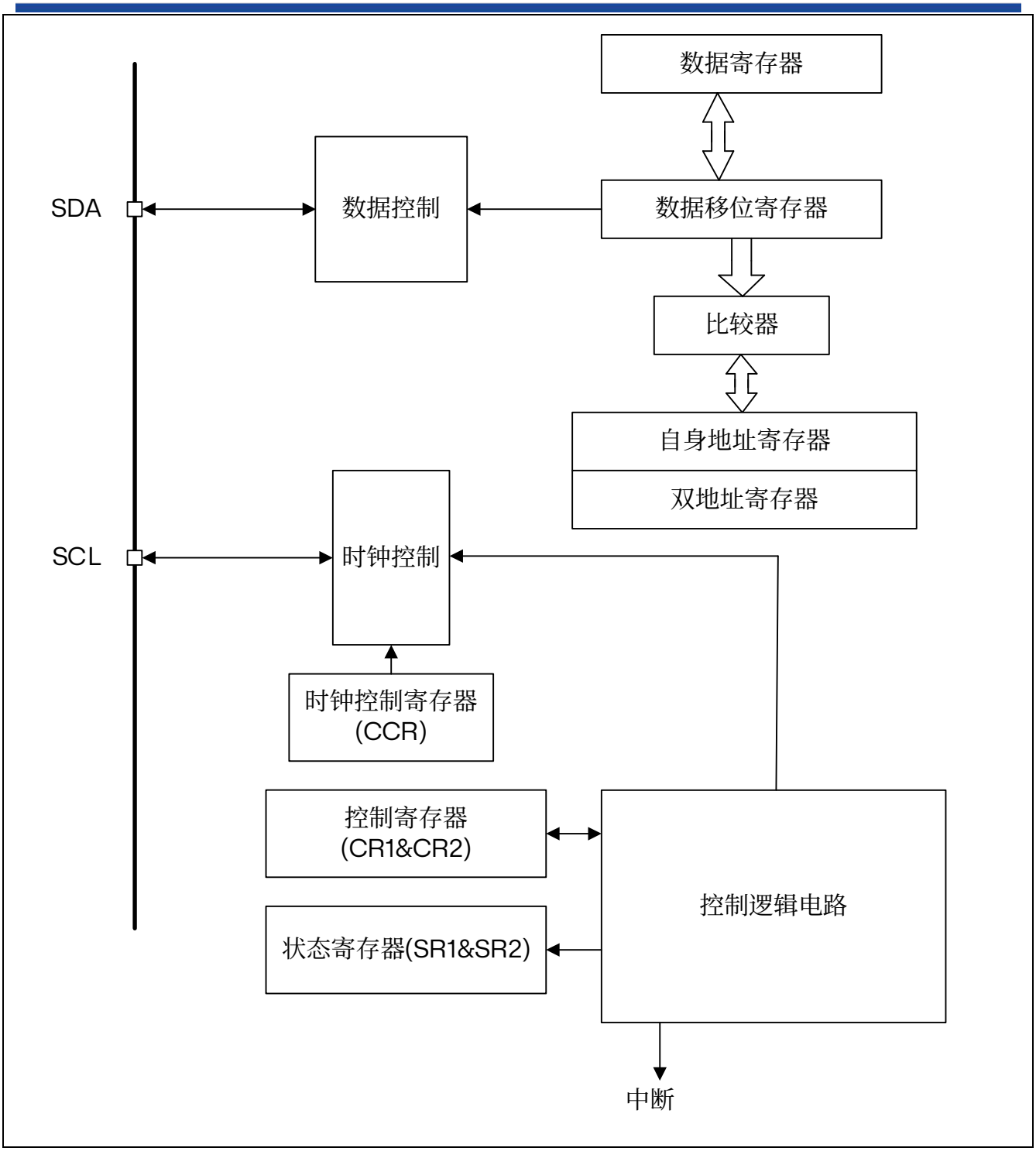


图 23.2 I²C 的功能框图

23.3.2 I²C 从模式

默认情况下，I²C 接口总是工作在从模式。从从模式切换到主模式，需要产生一个起始条件。为了产生正确的时序，必须在 I2C_CR2 寄存器中设定该模块的输入时钟。输入时钟的频率必须至少是：

- 标准模式下为：2MHz
- 快速模式下为：4MHz

一旦检测到起始条件，在 SDA 线上接收到的地址被送到移位寄存器。然后与芯片自己的地址 OAR1 和 OAR2（当 ENDUAL=1）或者广播呼叫地址（如果 ENGC=1）相比较。

注：在 10 位地址模式时，比较包括头段序列（11110xx0），其中的 xx 是地址的两个最高有效位。

头段或地址不匹配：I²C 接口将其忽略并等待另一个起始条件。

头段匹配（仅 10 位模式）：如果 ACK 位被置‘1’，I²C 接口产生一个应答脉冲并等待 8 位从地址。

地址匹配：I²C 接口产生以下时序：

- 如果 ACK 被置‘1’，则产生一个应答脉冲
- 硬件设置 ADDR 位；如果设置了 ITEVFEN 位，则产生一个中断
- 如果 ENDUAL=1，软件必须读 DUALF 位，以确认响应了哪个从地址。

在 10 位模式，接收到地址序列后，从设备总是处于接收器模式。在收到与地址匹配的头序列并且最低位为‘1’（即 11110xx1）后，当接收到重复的起始条件时，将进入发送器模式。

在从模式下 TRA 位指示当前是处于接收器模式还是发送器模式。

从发送器

在接收到地址和清除 ADDR 位后，从发送器将字节从 DR 寄存器经由内部移位寄存器发送到 SDA 线上。

从设备保持 SCL 为低电平，直到 ADDR 位被清除并且待发送数据已写入 DR 寄存器。（见下图中的 EV1 和 EV3）。

当收到应答脉冲时：

- TxE 位被硬件置位，如果设置了 ITEVFEN 和 ITBUFEN 位，则产生一个中断。

如果 TxE 位被置位，但在下一个数据发送结束之前没有新数据写入到 I2C_DR 寄存器，则 BTF 位被置位，在清除 BTF 之前 I²C 接口将保持 SCL 为低电平；读出 I2C_SR1 之后再写入 I2C_DR 寄存器将清除 BTF 位。

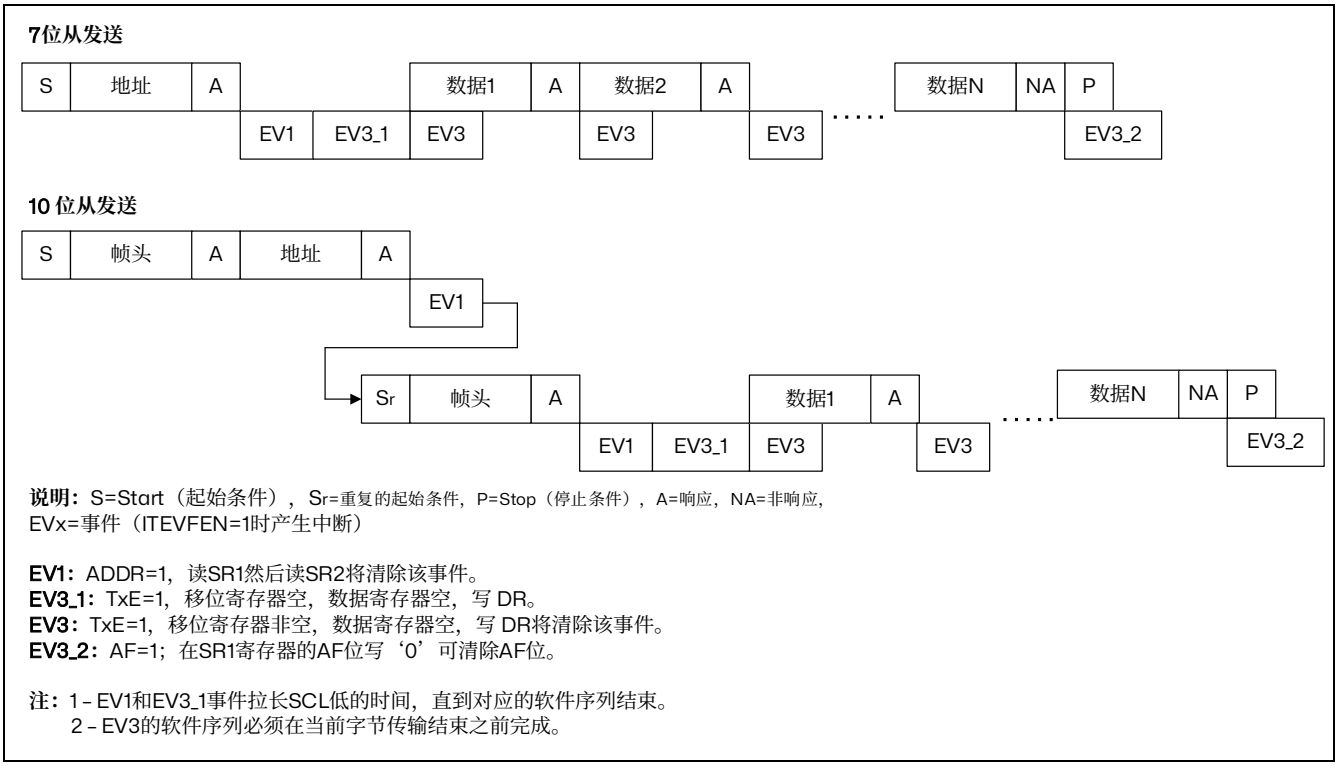


图 23.3 从发送器的传送序列图

从接收器

在接收到地址并清除 ADDR 后，从接收器将通过内部移位寄存器从 SDA 线接收到的字节存进 DR 寄存器。I²C 接口在接收到每个字节后都执行下列操作：

- 如果设置了 ACK 位，则产生一个应答脉冲
- 硬件设置 RxNE=1。如果设置了 ITEVFEN 和 ITBUFEN 位，则产生一个中断。

如果 RxNE 被置位，并且在接收新的数据结束之前 DR 寄存器未被读出，BTF 位被置位，在清除 BTF 之前 I²C 接口将保持 SCL 为低电平；读出 I2C_SR1 之后再写入 I2C_DR 寄存器将清除 BTF 位。（见下图）。

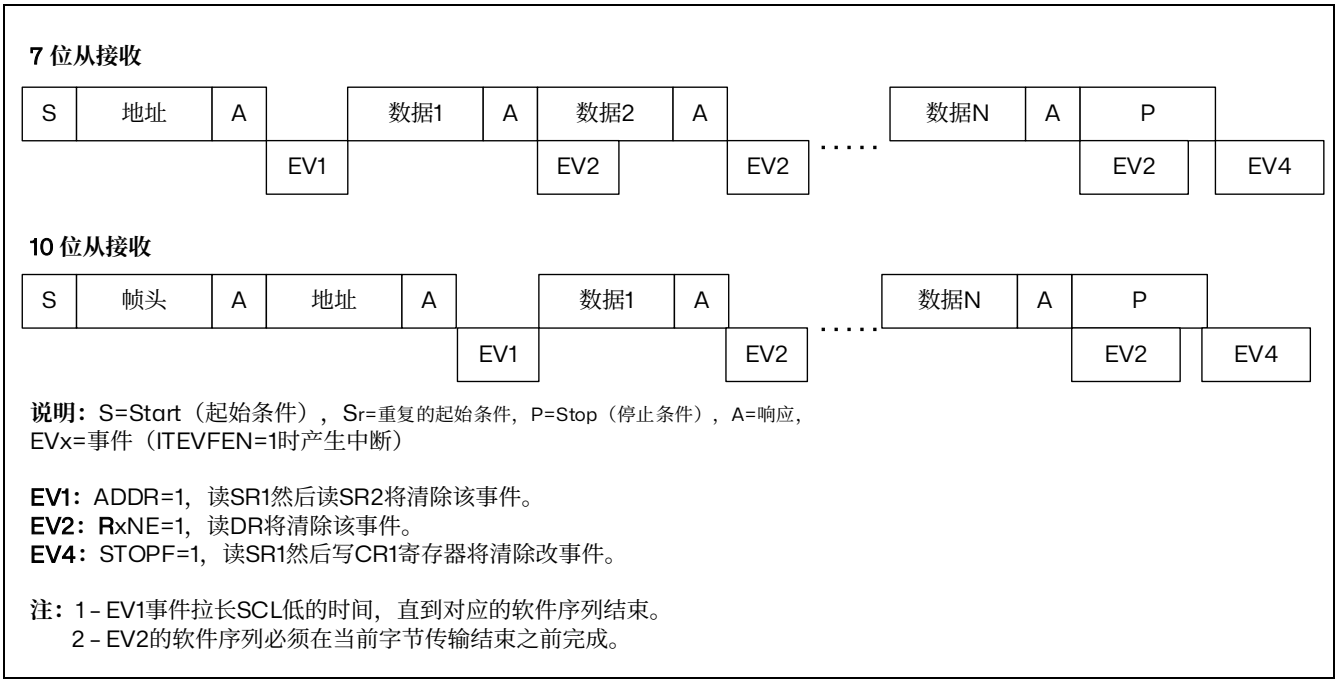


图 23.4 从接收器的传送序列图

关闭从通信

在传输完最后一个数据字节后，主设备产生一个停止条件，I²C 接口检测到这一条件时：

- 设置 STOPF=1，如果设置了 ITEVFEN 位，则产生一个中断。

然后 I²C 接口等待读 SR1 寄存器，再写 CR1 寄存器。（见上图的 EV4）。

23.3.3 I²C 主模式

在主模式时，I²C 接口启动数据传输并产生时钟信号。串行数据传输总是以起始条件开始并以停止条件结束。当通过 START 位在总线上产生了起始条件，设备就进入了主模式。

以下是主模式所要求的操作顺序：

- 在 I2C_CR2 寄存器中设定该模块的输入时钟以产生正确的时序
- 配置时钟控制寄存器
- 配置上升时间寄存器
- 编程 I2C_CR1 寄存器启动外设
- 置 I2C_CR1 寄存器中的 START 位为 1，产生起始条件

I²C 模块的输入时钟频率必须至少是：

- 标准模式下为：2MHz
- 快速模式下为：4MHz

起始条件

当 BUSY=0 时，设置 START=1，I²C 接口将产生一个开始条件并切换至主模式（M/SL 位置位）。

注：在主模式下，设置 START 位将在当前字节传输完后由硬件产生一个重开始条件。

一旦发出开始条件：

- SB 位被硬件置位，如果设置了 ITEVFEN 位，则会产生一个中断。

然后主设备等待读 SR1 寄存器，紧跟着将从地址写入 DR 寄存器。

从地址的发送

从地址通过内部移位寄存器被送到 SDA 线上。

- 在 10 位地址模式时，发送一个头段序列产生以下事件：
  - ADD10 位被硬件置位，如果设置了 ITEVFEN 位，则产生一个中断。
    然后主设备等待读 SR1 寄存器，再将第二个地址字节写入 DR 寄存器
  - ADDR 位被硬件置位，如果设置了 ITEVFEN 位，则产生一个中断。
    随后主设备等待一次读 SR1 寄存器，跟着读 SR2 寄存器。
- 在 7 位地址模式时，只需送出一个地址字节。一旦该地址字节被送出，
  - ADDR 位被硬件置位，如果设置了 ITEVFEN 位，则产生一个中断。
    随后主设备等待一次读 SR1 寄存器，跟着读 SR2 寄存器。

根据送出从地址的最低位，主设备决定进入发送器模式还是进入接收器模式。

- 在 7 位地址模式时，
  - 要进入发送器模式，主设备发送从地址时置最低位为‘0’。
  - 要进入接收器模式，主设备发送从地址时置最低位为‘1’。
- 在 10 位地址模式时
  - 要进入发送器模式，主设备先送头字节（11110xx0），然后送最低位为‘0’的从地址。（这里 xx 代表 10 位地址中的最高 2 位。）
  - 要进入接收器模式，主设备先送头字节（11110xx0），然后送最低位为‘1’的从地址。然后再重新发送一个开始条件，后面跟着头字节（11110xx1）（这里 xx 代表 10 位地址中的最高 2 位。）

TRA 位指示主设备是在接收器模式还是发送器模式。

主发送器

在发送了地址和清除了 ADDR 位后，主设备通过内部移位寄存器将字节从 DR 寄存器发送到 SDA 线上。

主设备等待，直到 TxE 被清除。

当收到应答脉冲时:

- TxE 位被硬件置位，如果设置了 INEVFEN 和 ITBUFEN 位，则产生一个中断。

如果 TxE 被置位并且在上一次数据发送结束之前没有写新的数据字节到 DR 寄存器，则 BTF 被硬件置位，在清除 BTF 之前 I²C 接口将保持 SCL 为低电平；读出 I2C_SR1 之后再写入 I2C_DR 寄存器将清除 BTF 位。

关闭通信

在 DR 寄存器中写入最后一个字节后，通过设置 STOP 位产生一个停止条件，然后 I²C 接口将自动回到从模式（M/S 位清除）。

注：当 TxE 或 BTF 位置位时，停止条件应安排在出现 EV8_2 事件时。

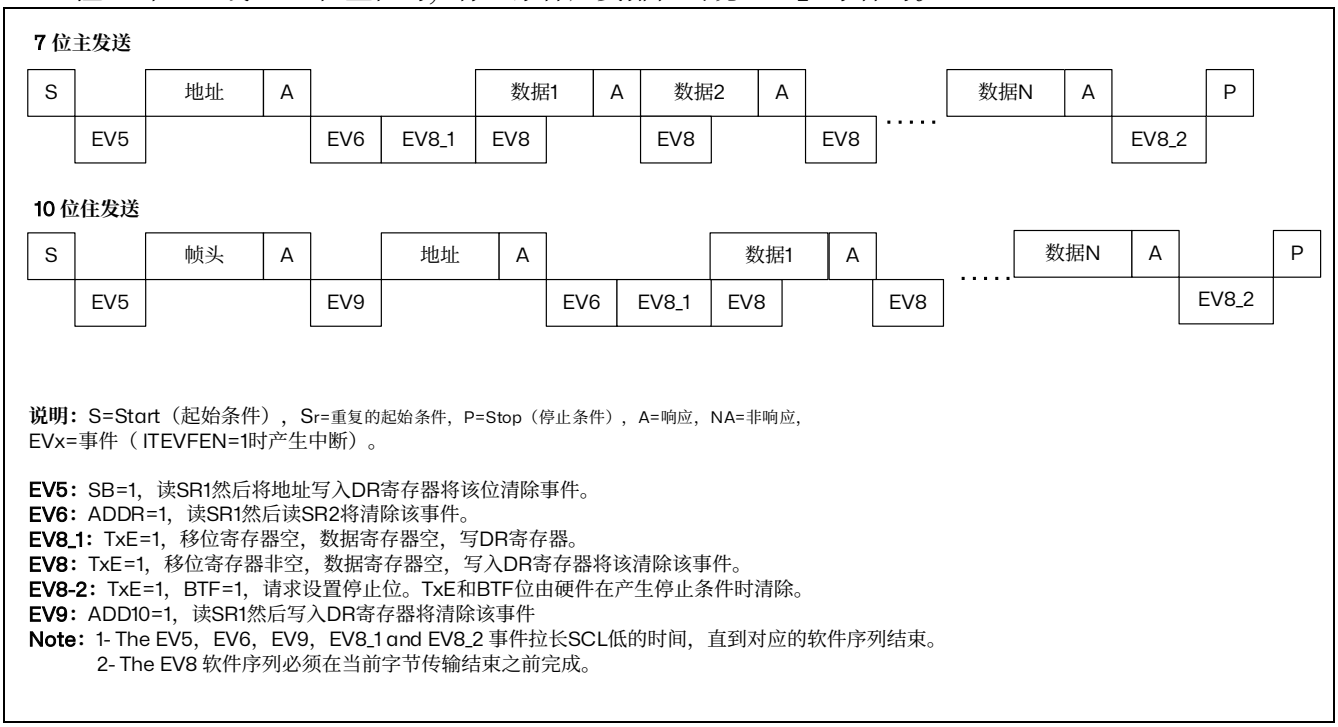


图 23.5 主发送器传送序列图

主接收器

在发送地址和清除 ADDR 之后，I²C 接口进入主接收器模式。在此模式下 I²C 接口从 SDA 线接收数据字节，并通过内部移位寄存器送至 DR 寄存器。在每个字节后，I²C 接口依次执行以下操作：

- 如果 ACK 位被置位，发出一个应答脉冲。
- 硬件设置 RxNE=1，如果设置了 INEVFEN 和 ITBUFEN 位，则会产生一个中断。

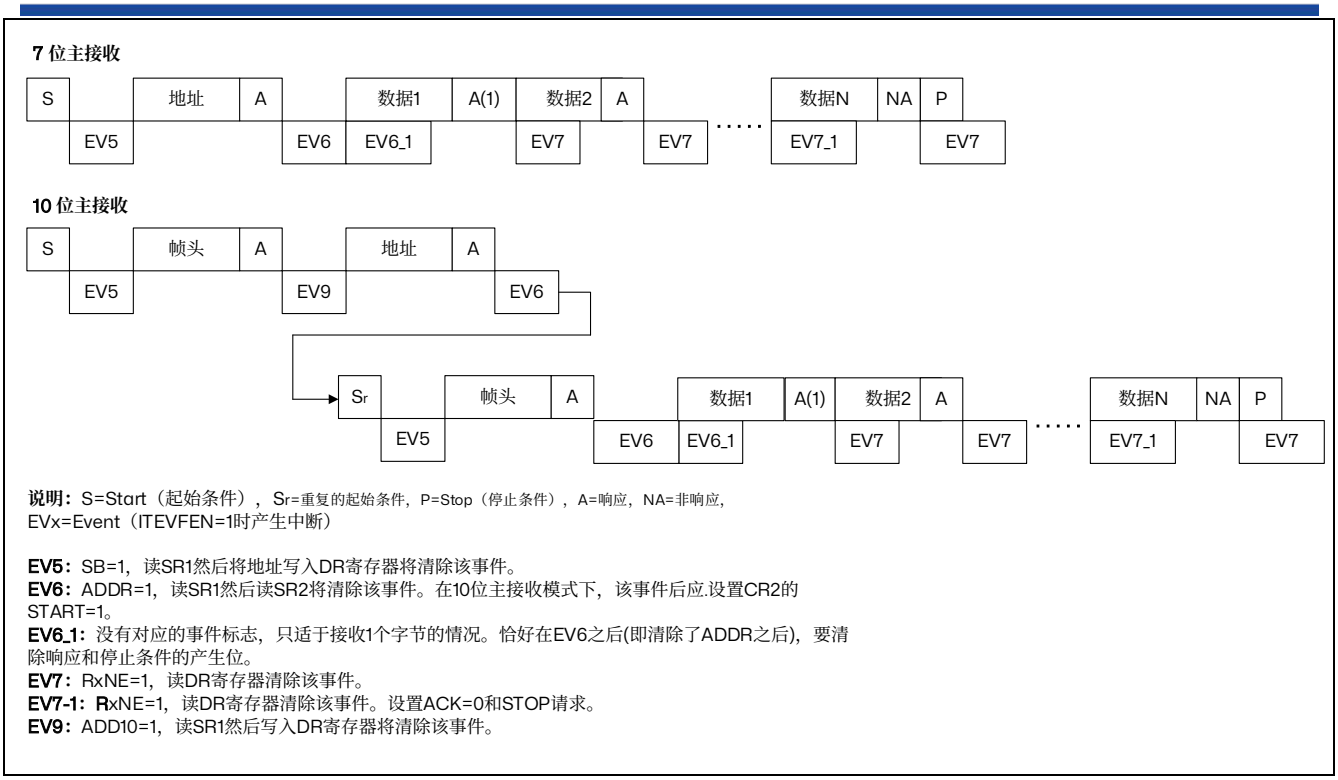
如果 RxNE 位被置位，并且在接收新数据结束前，DR 寄存器中的数据没有被读走，硬件将设置 BTF=1，在清除 BTF 之前 I²C 接口将保持 SCL 为低电平；读出 I2C_SR1 之后再读出 I2C_DR 寄存器将清除 BTF 位。

关闭通信

主设备在从从设备接收到最后一个字节后发送一个 NACK。接收到 NACK 后，从设备释放对 SCL 和 SDA 线的控制；主设备就可以发送一个停止/重起始条件。

- 为了在收到最后一个字节后产生一个 NACK 脉冲，在读倒数第二个数据字节之后（在倒数第二个 RxNE 事件之后）必须清除 ACK 位。
- 为了产生一个停止/重起始条件，软件必须在读倒数第二个数据字节之后（在倒数第二个 RxNE 事件之后）设置 STOP/START 位。
- 只接收一个字节时，刚好在 EV6 之后（EV6_1 时，清除 ADDR 之后）要关闭应答和停止条件的产生位。

在产生了停止条件后，I²C 接口自动回到从模式（M/SL 位被清除）。



说明： S=Start（起始条件），Sr=重复的起始条件，P=Stop（停止条件），A=响应，NA=非响应，EVx=Event（ITEVFEN=1时产生中断）

EV5： SB=1，读SR1然后将地址写入DR寄存器将清除该事件。
EV6： ADDR=1，读SR1然后读SR2将清除该事件。在10位主接收模式下，该事件后应.设置CR2的START=1。
EV6_1： 没有对应的事件标志，只适于接收1个字节的情况。恰好在EV6之后(即清除了ADDR之后)，要清除响应和停止条件的产生位。
EV7： RxNE=1，读DR寄存器清除该事件。
EV7-1： RxNE=1，读DR寄存器清除该事件。设置ACK=0和STOP请求。
EV9： ADD10=1，读SR1然后写入DR寄存器将清除该事件。

图 23.6 主接收器传送序列图

1. 如果收到一个单独的字节，则是 NA。
2. EV5、EV6 和 EV9 事件拉长 SCL 低电平，直到对应的软件序列结束。
3. EV7 的软件序列必须在当前字节传输结束前完成。
4. EV6_1 或 EV7_1 的软件序列 必须在当前传输字节的 ACK 脉冲之前完成。

23.3.4 错误条件

以下条件可能造成通讯失败。

总线错误（BERR）

在一个地址或数据字节传输期间，当 I²C 接口检测到一个外部的停止或起始条件则产生总线错误。此时：

- BERR 位被置位为‘1’；如果设置了 ITERREN 位，则产生一个中断；
- 在从模式情况下，数据被丢弃，硬件释放总线：
  - 如果是错误的开始条件，从设备认为是一个重启动，并等待地址或停止条件。
  - 如果是错误的停止条件，从设备按正常的停止条件操作，同时硬件释放总线。
- 在主模式情况下，硬件不释放总线，同时不影响当前的传输状态。此时由软件决定是否要中止当前的传输。

应答错误（AF）

当接口检测到一个无应答位时，产生应答错误。此时：

- AF 位被置位，如果设置了 ITERREN 位，则产生一个中断；
- 当发送器接收到一个 NACK 时，必须复位通讯：
  - 如果是处于从模式，硬件释放总线。

– 如果是处于主模式，软件必须生成一个停止条件。

仲裁丢失（ARLO）

当 I²C 接口检测到仲裁丢失时产生仲裁丢失错误，此时：

- ARLO 位被硬件置位，如果设置了 ITERREN 位，则产生一个中断；
- I²C 接口自动回到从模式（M/SL 位被清除）。当 I²C 接口丢失了仲裁，则它无法在同一个传输中响应它的从地址，但它可以在赢得总线的主设备发送重起始条件之后响应；
- 硬件释放总线。

过载/欠载错误（OVR）

在从模式下，如果禁止时钟延长，I²C 接口正在接收数据时，当它已经接收到一个字节（RxNE=1），但在 DR 寄存器中前一个字节数据还没有被读出，则发生过载错误。此时：

- 最后接收的数据被丢弃；
- 在过载错误时，软件应清除 RxNE 位，发送器应该重新发送最后一次发送的字节。

在从模式下，如果禁止时钟延长，I²C 接口正在发送数据时，在下一个字节的时钟到达之前，新的数据还未写入 DR 寄存器（TxE=1），则发生欠载错误。此时：

- 在 DR 寄存器中的前一个字节将被重复发出；
- 用户应该确定在发生欠载错时，接收端应丢弃重复接收到的数据。发送端应按 I²C 总线标准在规定的时间更新 DR 寄存器。

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

23.3.5 SDA/SCL 线控制

- 如果允许时钟延长：
  – 发送器模式：如果 TxE=1 且 BTF=1：I²C 接口在传输前保持时钟线为低，以等待软件读取 SR1，然后把数据写进数据寄存器（缓冲器和移位寄存器都是空的）。
  – 接收器模式：如果 RxNE=1 且 BTF=1：I²C 接口在接收到数据字节后保持时钟线为低，以等待软件读 SR1，然后读数据寄存器 DR（缓冲器和移位寄存器都是满的）。
- 如果在从模式中禁止时钟延长：
  – 如果 RxNE=1，在接收到下个字节前 DR 还没有被读出，则发生过载错。接收到的最后一个字节丢失。
  – 如果 TxE=1，在必须发送下个字节之前却没有新数据写进 DR，则发生欠载错。相同的字节将被重复发出。
  – 不控制重复写冲突。

23.4 I²C 中断请求

下表列出了所有的 I²C 中断请求

表 23.1 I²C 中断请求表:

中断事件	事件标志	开启控制位
起始位已发送（主）	SB	ITEVFEN
地址已发送（主）或地址匹配（从）	ADDR	
10 位头段已发送（主）	ADD10	
已收到停止（从）	STOPF	
数据字节传输完成	BTF	
接收缓冲区非空	RxNE	ITEVFEN 和 ITBUFEN
发送缓冲区空	TxE	
总线错误	BERR	ITERREN
仲裁丢失（主）	ARLO	
响应失败	AF	
过载/欠载	OVR	
超时/Tlow 错误	TIMEOUT	

注： 1. SB、ADDR、ADD10、STOPF、BTF、RxNE *和* TxE *通过逻辑或汇到同一个中断通道中。*
2. BERR、ARLO、AF、OVR、TIMEOUT *通过逻辑或汇到同一个中断通道中。*

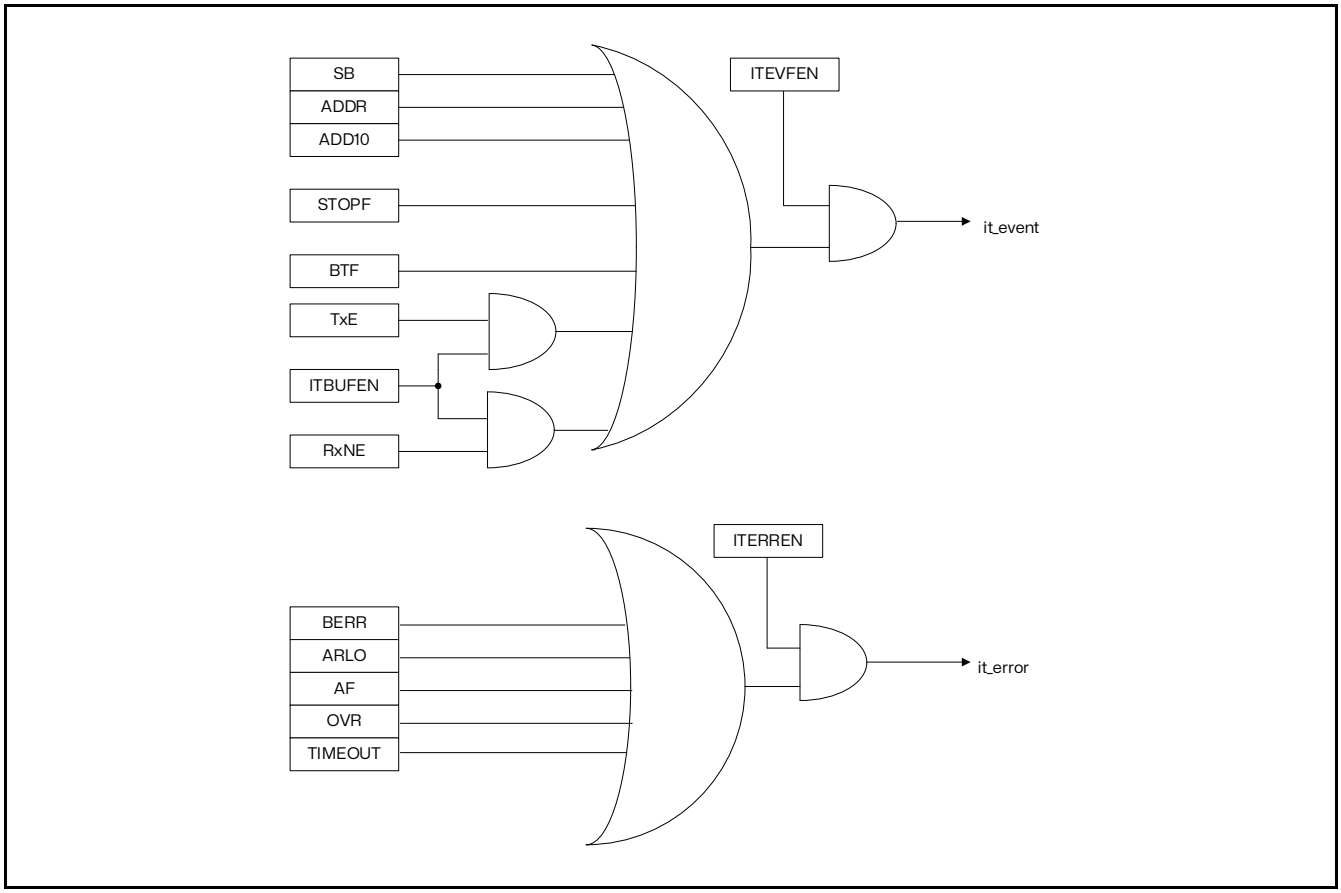


图 23.7 I²C 中断映射图

23.5 I²C 寄存器

访问：无等待，支持字，半字和字节访问

23.5.1 控制寄存器 1（I2C_CR1）

地址偏移：0x00

复位值：0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWRST	Reserved			POS	ACK	STOP	START	NOSTRETCH	ENGC	Reserved					PE
rw				rw	rw	rw	rw	rw	rw						rw

Bit 15 **SWRST**：软件复位（Software reset）

当被置位时，I²C 处于复位状态。在复位该位前确信 I²C 的引脚被释放，总线是空的。

0：I²C 模块不处于复位状态；

1：I²C 模块处于复位状态。

注：该位可以用于 BUSY 位为‘1’，在总线上又没有检测到停止条件时。

Bits 14:12 保留，必须保持为复位值。

Bit 11 **POS**：应答（Acknowledge）

软件可以设置或清除该位，或当 PE=0 时，由硬件清除。

0：ACK 位控制当前移位寄存器内正在接收的字节的（N）ACK。

1：ACK 位控制在移位寄存器里接收的下一个字节的（N）ACK。

注：POS 位只能用在 2 字节的接收配置中，必须在接收数据之前配置。

为了 NACK 第 2 个字节，必须在清除 ADDR 为之后清除 ACK 位。

Bit 10 **ACK**：应答使能（Acknowledge enable）

软件可以设置或清除该位，或当 PE=0 时，由硬件清除。

0：无应答返回；

1：在接收到一个字节后返回一个应答（匹配的地址或数据）。

Bit 9 **STOP**：停止条件产生（Stop generation）

软件可以设置或清除该位；或当检测到停止条件时，由硬件清除；当检测到超时错误时，硬件将其置位。

在主模式下：

0：无停止条件产生；

1：在当前字节传输或在当前起始条件发出后产生停止条件。

在从模式下：

0：无停止条件产生；

1：在当前字节传输或释放 SCL 和 SDA 线。

注：当设置了 STOP、START 位，在硬件清除这个位之前，软件不要执行任何对 I2C_CR1 的写操作；否则有可能会第 2 次设置 STOP、START 位。

Bit 8 **START**：起始条件产生（Start generation）

软件可以设置或清除该位，或当起始条件发出后或 PE=0 时，由硬件清除。

在主模式下：

0：无起始条件产生；

1：重复产生起始条件。

在从模式下：

0：无起始条件产生；

1：当总线空闲时，产生起始条件。

Bit 7 **NOSTRETCH**：禁止时钟延长（从模式）（Clock stretching disable （Slave mode））

该位用于当 ADDR 或 BTF 标志被置位，在从模式下禁止时钟延长，直到它被软件复位。

0：允许时钟延长；

1：禁止时钟延长。

Bit 6 ENGC：广播呼叫使能（General call enable）

0：禁止广播呼叫。以非应答响应地址 00h；

1：允许广播呼叫。以应答响应地址 00h。

Bits 5:1 保留，必须保持为复位值。

Bit 0 **PE**：I²C 模块使能 （Peripheral enable）

0：禁用 I²C 模块；

1：使能 I²C 模块。

注：如果清除该位时通讯正在进行，在当前通讯结束后，I²C 模块被禁用并返回空闲状态。

由于在通讯结束后发生 PE＝0，所有的位被清除。

在主模式下，通讯结束之前，绝不能清除该位。

23.5.2 控制寄存器 2（I2C_CR2）

地址偏移：0x04

复位值：0x0000 0000

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

Bits 15:11 保留，必须保持为复位值。

Bit 10 **ITBUFEN**：缓冲器中断使能（Buffer interrupt enable）

0：当 TxE=1 或 RxNE=1 时，不产生任何中断；

1：当 TxE=1 或 RxNE=1 时，产生事件中断

Bit 9 **ITEVTEN**：事件中断使能（Event interrupt enable）

0：禁止事件中断；

1：允许事件中断。

在下列条件下，将产生该中断：

- SB = 1 （主模式）；
- ADDR = 1 （主/从模式）；
- ADD10= 1 （主模式）；
- STOPF = 1 （从模式）；
- BTF = 1，但是没有 TxE 或 RxNE 事件；
- 如果 ITBUFEN = 1，TxE 事件为 1；
- 如果 ITBUFEN = 1，RxNE 事件为 1。

Bit 8 **ITERREN**：出错中断使能（Error interrupt enable）

0：禁止出错中断；

1：允许出错中断。

在下列条件下，将产生该中断：

– BERR = 1;
– ARLO = 1;
– AF = 1;
– OVR = 1;
– TIMEOUT = 1;

Bits 7:6 保留，必须保持为复位值。

Bits 5:0 **FREQ[5:0]：**外设时钟频率（Peripheral clock frequency）

外设利用 FREQ 位域生成符合 I2C 规范的数据建立时间和保持时间。最小运行频率为 2MHz，最大频率受 APB 频率限制，且不得超过 36MHz（外设固有最大限制）。

0b000000：禁用
0b000001：禁用
0b000010：2MHz
…
0b100100：36MHz
大于 100100：禁用。

23.5.3 自身地址寄存器 1（I2C_OAR1）

复位地址偏移：0x08

复位值：0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADD MODE	Res	Reserved				ADD[9:8]		ADD[7:1]							ADD0
rw						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit 15 ADDMODE：寻址模式（从模式）（Addressing mode （slave mode））

0：7 位从地址（不响应 10 位地址）；
1：10 位从地址（不响应 7 位地址）。

Bit 14 必须始终由软件保持为‘1’。

Bits 13:12 保留，必须保持为复位值。

Bits 9:8 ADD[9:8]：接口地址（Interface address）

7 位地址模式时不用关心。
10 位地址模式时为地址的 9~8 位。

Bits 7:1 ADD[7:1]：接口地址（Interface address）

地址的 7~1 位。

Bit 0 ADD0：接口地址（Interface address）

7 位地址模式时不用关心。
10 位地址模式时为地址第 0 位。

23.5.4 自身地址寄存器 2（I2C_OAR2）

地址偏移：0x0C

复位值：0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ADD2[7:1]							ENDUAL
								rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:8 保留，必须保持为复位值。

Bits 7:1 ADD2[7:1]：接口地址（Interface address）

在双地址模式下地址的 7~1 位。

Bit 0 ENDUAL：双地址模式使能位（Dual addressing mode enable）

0：在 7 位地址模式下，只有 OAR1 被识别；

1：在 7 位地址模式下，OAR1 和 OAR2 都被识别。

23.5.5 数据寄存器（I2C_DR）

地址偏移：0x10

复位值：0x0000

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

Bits 15:8 保留，必须保持为复位值。

Bits 7:0 DR[7:0]：8 位数据寄存器（8-bit data register）

用于存放接收到的数据或放置用于发送到总线的数据

发送器模式：当写一个字节至 DR 寄存器时，自动启动数据传输。一旦传输开始（TxE=1），如果能及时把下一个需传输的数据写入 DR 寄存器，I²C 模块将保持连续的数据流。

接收器模式：接收到的字节被拷贝到 DR 寄存器（RxNE=1）。在接收到下一个字节（RxNE=1）之前读出数据寄存器，即可实现连续的数据传送。

注：在从模式下，地址不会被拷贝进数据寄存器 DR；

注：硬件不管理写冲突（如果 TxE=0，仍能写入数据寄存器）；

注：如果在处理 ACK 脉冲时发生 ARLO 事件，接收到的字节不会被拷贝到数据寄存器里，因此不能读到它。

23.5.6 状态寄存器 1（I2C_SR1）

地址偏移：0x14

复位值：0x0000 00000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TIME OUT	Reserved		OVR	AF	ARLO	BERR	TxE	RxNE	Res	STOPF	ADD10	BTF	ADDR	SB
	rc w0			rc w0	rc w0	rc w0	rc w0	r	r		r	r	r	r	r

Bit 15 保留，必须保持为复位值。

Bit 14 **TIMEOUT**：超时或 Tlow 错误（Timeout or Tlow error）

0：无超时错误；

1：SCL 处于低已达到 25ms（超时）；或者主机低电平累积时钟扩展时间超过 10ms（Tlow:mext）；或从设备低电平累积时钟扩展时间超过 25ms（Tlow:sext）。

– 当在从模式下设置该位：从设备复位通讯，硬件释放总线。
– 当在主模式下设置该位：硬件发出停止条件。
– 该位由软件写‘0’清除，或在 PE=0 时由硬件清除。

Bits 13:12 保留，必须保持为复位值。

Bit 11 **OVR**：过载/欠载（Overrun/Underrun）

0：无过载/欠载；
1：出现过载/欠载。
– 当 NOSTRETCH=1 时，在从模式下该位被硬件置位，同时：
– 在接收模式中当收到一个新的字节时（包括 ACK 应答脉冲），数据寄存器里的内容还未被读出，则新接收的字节将丢失。
– 在发送模式中当要发送一个新的字节时，却没有新的数据写入数据寄存器，同样的字节将被发送两次。
– 该位由软件写‘0’清除，或在 PE=0 时由硬件清除。
注：如果数据寄存器的写操作发生时间非常接近 SCL 的上升沿，发送的数据是不确定的，并发生保持时间错误。

Bit 10 **AF**：应答失败（Acknowledge failure）

0：没有应答失败；
1：应答失败。
– 当没有返回应答时，硬件将置该位为‘1’。
– 该位由软件写‘0’清除，或在 PE=0 时由硬件清除。

Bit 9 **ARLO**：仲裁丢失（主模式）（Arbitration lost（master mode））

0：没有检测到仲裁丢失；
1：检测到仲裁丢失。
当接口失去对总线的控制给另一个主机时，硬件将将置该位为‘1’。
– 该位由软件写‘0’清除，或在 PE=0 时由硬件清除。
在 ARLO 事件之后，I²C 接口自动切换回从模式（M/SL=0）。

Bit 8 **BERR**：总线出错（Bus error）

0：无起始或停止条件出错；
1：起始或停止条件出错。
– 当接口检测到错误的起始或停止条件，硬件将该位置‘1’。
– 该位由软件写‘0’清除，或在 PE=0 时由硬件清除。

Bit 7 **TxE**：数据寄存器为空（发送时）（Data register empty （transmitters））

0：数据寄存器非空；
1：数据寄存器空。
– 在发送数据时，数据寄存器为空时该位被置‘1’，在发送地址阶段不设置该位。
– 软件写数据到 DR 寄存器可清除该位；或在发生一个起始或停止条件后，或当 PE=0 时由硬件自动清除。
如果收到一个 NACK，该位不被置位。
注：在写入第 1 个要发送的数据后，或设置了 BTF 时写入数据，都不能清除 TxE 位，这是因为数据寄存器仍然为空。

Bit 6 **RxNE**：数据寄存器非空（接收时）（Data register not empty （receivers））

0：数据寄存器为空；
1：数据寄存器非空。

– 在接收时，当数据寄存器不为空，该位被置‘1’。在接收地址阶段，该位不被置位。
– 软件对数据寄存器的读写操作清除该位，或当 PE=0 时由硬件清除。

在发生 ARLO 事件时，RxNE 不被置位。

注：当设置了 BTF 时，读取数据不能清除 RxNE 位，因为数据寄存器仍然为满。

Bit 5 保留，必须保持为复位值。

Bit 4 **STOPF：**停止条件检测位（从模式）（Stop detection （slave mode））

0：没有检测到停止条件；

1：检测到停止条件。

– 在一个应答之后（如果 ACK=1），当从设备在总线上检测到停止条件时，硬件将该位置‘1’。
– 软件读取 SR1 寄存器后，对 CR1 寄存器的写操作将清除该位，或当 PE=0 时，硬件清除该位。

注：在收到 NACK 后，STOPF 位不被置位。

Bit 3 **ADD10：**10 位头序列已发送（主模式）（10-bit header sent （Master mode））

0：没有 ADD10 事件发生；

1：主设备已经将第一个地址字节发送出去。

– 在 10 位地址模式下，当主设备已经将第一个字节发送出去时，硬件将该位置‘1’。
– 软件读取 SR1 寄存器后，对 CR1 寄存器的写操作将清除该位，或当 PE=0 时，硬件清除该位。

注：收到一个 NACK 后，ADD10 位不被置位。

Bit 2 **BTF：**字节发送结束（Byte transfer finished）

0：字节发送未完成；

1：字节发送结束。

当 NOSTRETCH=0 时，在下列情况下硬件将该位置‘1’：

– 在接收时，当收到一个新字节（包括 ACK 脉冲）且数据寄存器还未被读取（RxNE=1）。
– 在发送时，当一个新数据将被发送且数据寄存器还未被写入新的数据（TxE=1）。
– 在软件读取 SR1 寄存器后，对数据寄存器的读或写操作将清除该位；或在传输中发送一个起始或停止条件后，或当 PE=0 时，由硬件清除该位。

*注：在收到一个*NACK *后，*BTF *位不会被置位。*

Bit 1 **ADDR：**地址已被发送（主模式）/地址匹配（从模式）（Address sent（master mode）/matched（slave mode））

在软件读取 SR1 寄存器后，对 SR2 寄存器的读操作将清除该位，或当 PE=0 时，由硬件清除该位。

地址匹配（从模式）

0：地址不匹配或没有收到地址；

1：收到的地址匹配。

– 当收到的从地址与 OAR 寄存器中的内容相匹配、或发生广播呼叫时，硬件就将该位置‘1’（当对应的设置被使能时）。

地址已被发送（主模式）

0：地址发送没有结束；

1：地址发送结束。

– 10 位地址模式时，当收到地址的第二个字节的 ACK 后该位被置‘1’。
– 7 位地址模式时，当收到地址的 ACK 后该位被置‘1’。

注：在收到 NACK 后，ADDR 位不会被置位。

Bit 0 **SB：**起始位（主模式）（Start bit（Master mode））

0：未发送起始条件；
1：起始条件已发送。
– 当发送出起始条件时该位被置‘1’。
– 软件读取 SR1 寄存器后，写数据寄存器的操作将清除该位，或当 PE=0 时，硬件清除该位。

23.5.7 状态寄存器 2（I2C_SR2）

地址偏移：0x18

复位值：0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								DUALF	Reserved		GENCALL	Res	TRA	BUSY	MSL
								r			r		r	r	r

Bits 15:8 保留，必须保持为复位值。

Bit 7 **DUALF：**双标志（从模式）（Dual flag（Slave mode））

0：接收到的地址与 OAR1 内的内容相匹配；
1：接收到的地址与 OAR2 内的内容相匹配。
– 在产生一个停止条件或一个重复的起始条件时，或 PE=0 时，硬件将该位清除。

Bits 6:5 保留，必须保持为复位值。

Bit 4 **GENCALL：**广播呼叫地址（从模式）（General call address（Slave mode））

0：未收到广播呼叫地址；
1：当 ENGC=1 时，收到广播呼叫的地址
– 在产生一个停止条件或一个重复的起始条件时，或 PE=0 时，硬件将该位清除。

Bit 3 保留，必须保持为复位值。

Bit 2 **TRA：**发送/接收（Transmitter/receiver）

0：接收到数据；
1：数据已发送；
在整个地址传输阶段的结尾，该位根据地址字节的 R/W 位来设定。
在检测到停止条件（STOPF=1）、重复的起始条件或总线仲裁丢失（ARLO=1）后，或当 PE=0 时，硬件将其清除。

Bit 1 **BUSY：**总线忙（Bus busy）

0：在总线上无数据通讯；
1：在总线上正在进行数据通讯。
– 在检测到 SDA 或 SCl 为低电平时，硬件将该位置‘1’；
– 当检测到一个停止条件时，硬件将该位清除。
该位指示当前正在进行的总线通讯，当接口被禁用（PE=0）时该信息仍然被更新。

Bit 0 **MSL：**主从模式（Master/slave）

0：从模式；
1：主模式。
– 当接口处于主模式（SB=1）时，硬件将该位置位；

– 当总线上检测到一个停止条件、仲裁丢失（ARLO=1 时）、或当 PE=0 时，硬件清除该位。

23.5.8 时钟控制寄存器（I2C_CCR）

地址偏移：0x1C

复位值：0x0000 0000

注：1.要求 F_{PCLK1} 应当是 10MHz 的整数倍，这样可以正确地产生 400KHz 的快速时钟。

2.CCR 寄存器只有在关闭 I²C 时（PE=0）才能设置

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F/S	DUTY	Reserved		CCR[11:0]											
rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit 15 **F/S**：I²C 主模式选项（I2C master mode selection）

0：标准模式的 I²C；

1：快速模式的 I²C。

Bit 14 **DUTY**：快速模式时的占空比（Fast mode duty cycle）

0：快速模式下：$T_{low}/T_{high} = 2$；

1：快速模式下：$T_{low}/T_{high} = 16/9$。

Bits 13:12 保留，必须保持为复位值。

Bits 11:0 **CCR[11:0]**：快速/标准模式下的时钟控制分频系数（主模式）（Clock control register in Fast/Standard mode (Master mode)）

该分频系数用于设置主模式下的 SCL 时钟。

在 I²C 标准模式或 SMBus 模式下：

$T_{high} = CCR \times T_{PCLK1}$

$T_{low} = CCR \times T_{PCLK1}$

在 I²C 快速模式下：

如果 DUTY = 0：

$T_{high} = CCR \times T_{PCLK1}$

$T_{low} = 2 \times CCR \times T_{PCLK1}$

如果 DUTY = 1：

$T_{high} = 9 \times CCR \times T_{PCLK1}$

$T_{low} = 16 \times CCR \times T_{PCLK1}$

例如：在标准模式下，产生 100kHz 的 SCL 的频率：

如果 FREQ=08，T_{PCLK1} = 125ns，则 CCR 必须写入 0x28

（0x28 <=> 40d × 125ns = 5000 ns）。

注：*允许设定的最小值为 0x04，在快速 DUTY 模式下允许的最小值为 0x01；*

$t_{high} = t_{r(SCL)}+t_{w(SCLH)}$，*详见数据手册中对这些参数的定义；*

$t_{low} = t_{f(SCL)}+t_{w(SCLL)}$，*详见数据手册中对这些参数的定义；*

I2C *通信速率* $f_{SCL} \approx 1/(t_{high} + t_{low})$。*实际频率可能因模拟噪声滤波器输入延迟存在差异；*

只有在关闭 I²C 时（PE = 0）才能设置 CCR 寄存器；

23.5.9 TRISE 寄存器（I2C_TRISE）

地址偏移：0x20

复位值：0x0000 0002

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

Bits 15:6 保留，必须保持为复位值。

Bits 5:0 **TRISE[5:0]：**在快速/标准模式下的最大上升时间（主模式）（Maximum rise time in Fast/Standard mode (Master mode)）

这些位用于设置主模式中 SCL 反馈环路的最大持续时间，其目的是无论 SCL 上升沿持续时间如何变化，均能维持 SCL 频率的稳定性。

编程时需将 I2C 总线规范中规定的最大 SCL 上升时间值加 1 后写入该位域。

例如：标准模式中最大允许 SCL 上升时间为 1000ns。若 T_{PCLK1}=125ns，故 TRISE[5:0]中必须写入 09h（1000ns/125 ns = 8+1）。

滤波器的值也可以加到 TRISE[5:0]内。

如果结果不是一个整数，则将整数部分写入 TRISE[5:0]以确保 t_{HIGH} 参数。

注：只有当 I^2C *被禁用（*PE=0*）时，才能设置* TRISE[5:0]。

24 通用异步收发器（UART）

24.1 UART 介绍

通用异步收发器（UART）提供了一种灵活的方法与使用工业标准 NRZ 异步串行数据格式的外部设备之间进行全双工数据交换。UART 利用分数波特率发生器提供宽范围的波特率选择。它支持单向通信和半双工单线通信，也支持 LIN（局部互连网）。

24.2 UART 主要特性

- 全双工的，异步通信
- NRZ 标准格式
- 分数波特率发生器系统
  - 发送和接收共用的可编程波特率，最高达 4.5Mbits/s
- 可编程数据字长度（8 位或 9 位）
- 可配置的停止位-支持 1 或 2 个停止位
- LIN 主发送同步断开符的能力以及 LIN 从检测断开符的能力
  - 当 UART 硬件配置成 LIN 时，生成 13 位断开符；检测 10/11 位断开符
- 单线半双工通信
- 单独的发送器和接收器使能位
- 检测标志
  - 接收缓冲器满
  - 发送缓冲器空
  - 传输结束标志
- 校验控制
  - 发送校验位
  - 对接收数据进行校验
- 四个错误检测标志
  - 溢出错误
  - 噪音错误
  - 帧错误
  - 校验错误
- 9 个带标志的中断源
  - LIN 断开符检测
  - 发送数据寄存器空
  - 发送完成
  - 接收数据寄存器满
  - 溢出错误
  - 帧错误
  - 噪音错误
  - 校验错误

24.3 UART 功能概述

接口通过三个引脚与其他设备连接在一起。任何 UART 双向通信至少需要两个脚：接收数据输入（RX）和发送数据输出（TX）。

RX：接收数据串行输。通过过采样技术来区别数据和噪音，从而恢复数据。

TX：发送数据输出。当发送器被禁止时，输出引脚恢复到它的 I/O 端口配置。当发送器被激活，并且未发送数据时，TX 引脚处于高电平；在发送完数据后，TX 引脚处于高阻态，需要配置内部上拉来保持 TX 引脚处于高电平。在单线模式里，此 I/O 口被同时用于数据的发送和接收。

- 总线在发送或接收前应处于空闲状态
- 一个起始位
- 一个数据字（8 或 9 位），最低有效位在前
- 2 个的停止位，由此表明数据帧的结束
- 使用分数波特率发生器 –– 12 位整数和 4 位小数的表示方法。
- 一个状态寄存器（UART_SR）
- 数据寄存器（UART_DR）
- 一个波特率寄存器（UART_BRR），12 位的整数和 4 位小数

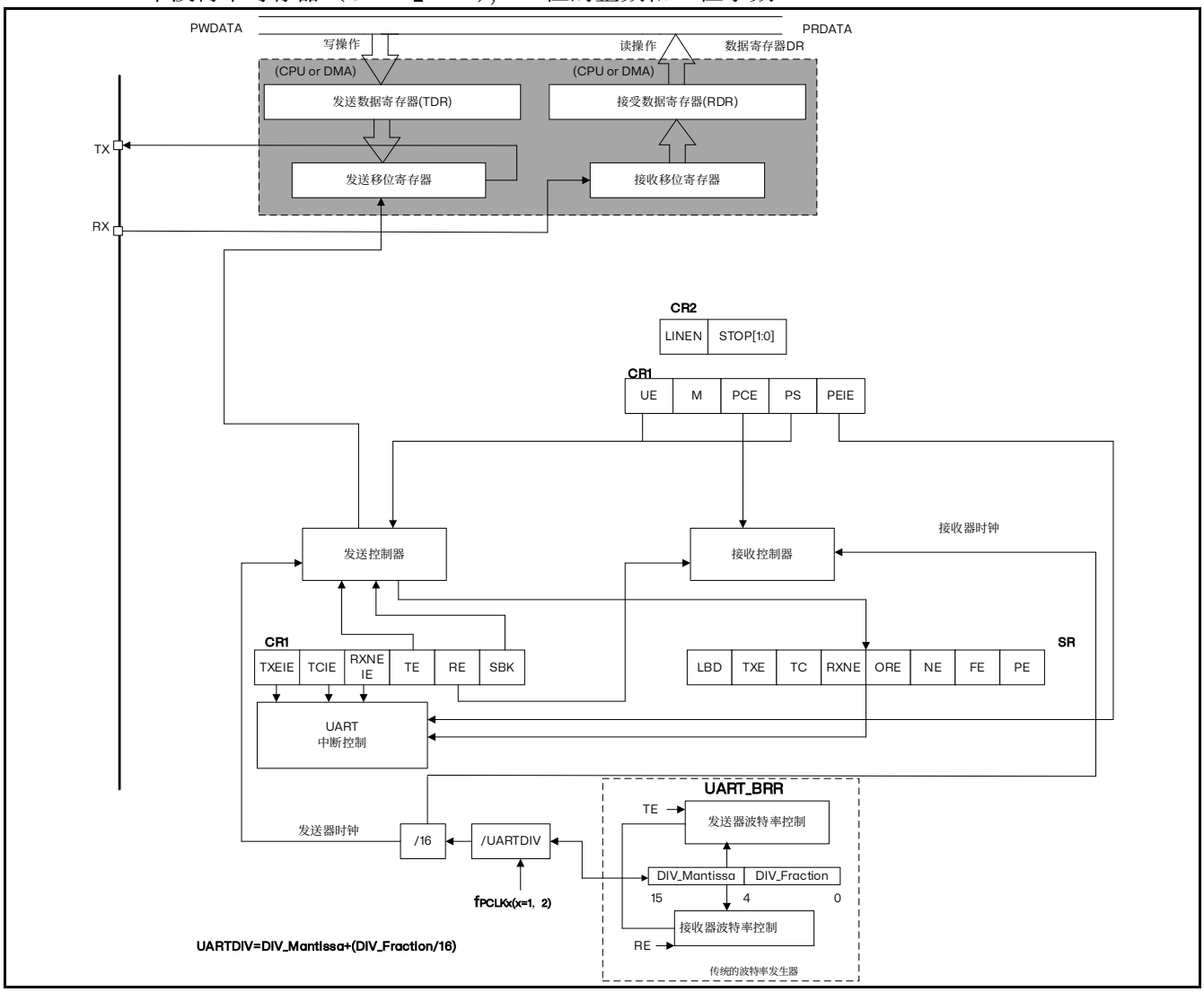


图 24.1 UART 框图

24.3.1 UART 特性描述

字长可以通过编程 UART_CR1 寄存器中的 M 位，选择成 8 或 9 位。在起始位期间，TX 脚处于低电平，在停止位期间处于高电平。

断开符号被视为在一个帧周期内全部收到‘0’（包括停止位期间，也是‘0’）。在断开帧结束时，发送器再插入 1 或 2 个停止位（‘1’）来应答起始位。

发送和接收由一共用的波特率发生器驱动，当发送器和接收器的使能位分别置位时，分别为其产生时钟。

随后将有每个功能块的详细说明。

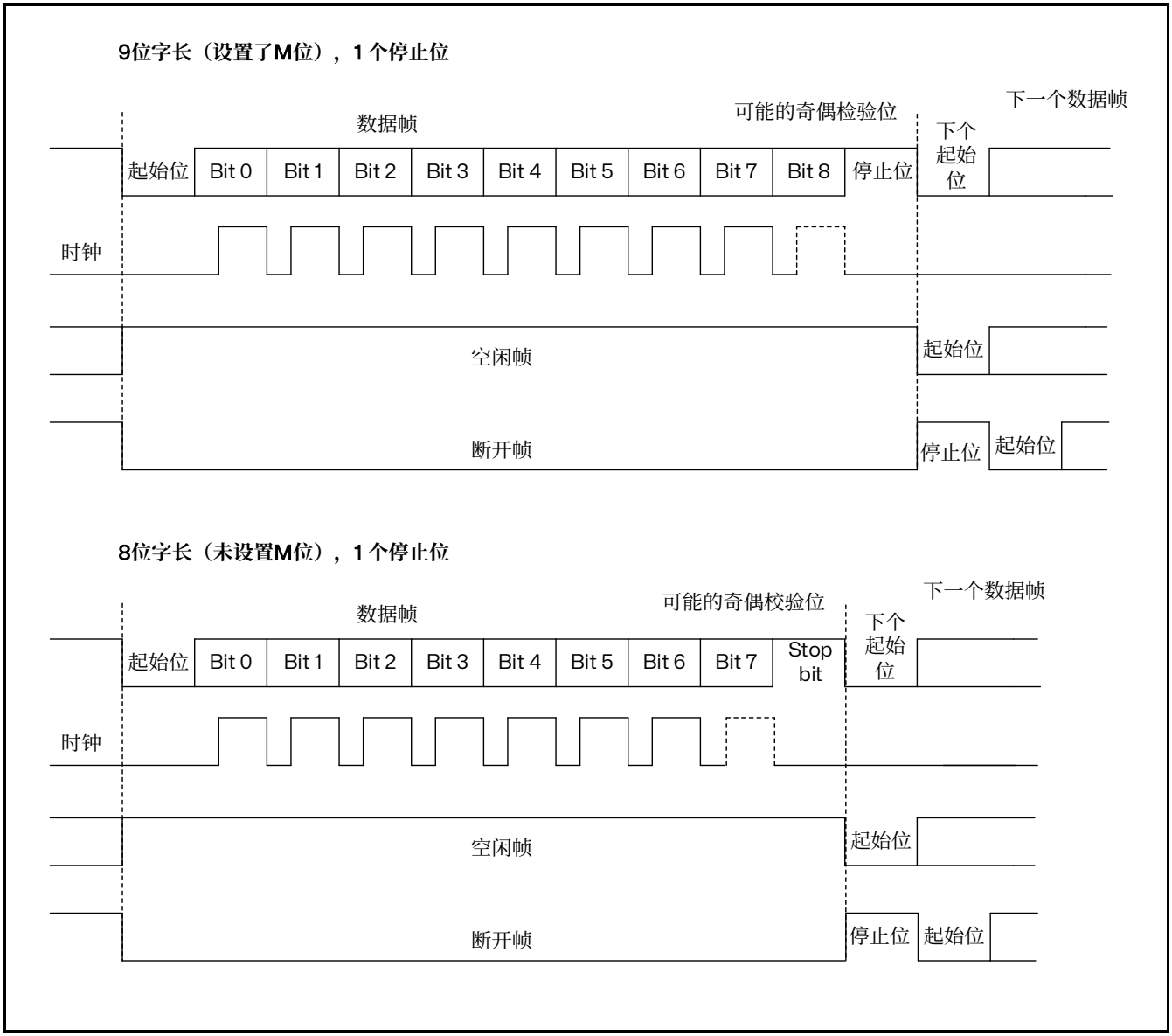


图 24.2 字长设置

24.3.2 发送器

发送器根据 M 位的状态发送 8 位或 9 位的数据字。当发送使能位（TE）被设置时，发送移位寄存器中的数据在 TX 脚上输出。

字符发送

在 UART 发送期间，在 TX 引脚上首先移出数据的最低有效位。在此模式里，UART_DR 寄存器包含了一个内部总线和发送移位寄存器之间的缓冲器。

每个字符之前都有一个低电平的起始位；之后跟着的停止位，其数目可配置。

UART 支持多种停止位的配置：1 和 2 个停止位。

注：1. 在数据传输期间不能复位 TE *位，否则将破坏* TX *脚上的数据，因为波特率计数器停止计数。正在传输的当前数据将丢失。*

2. TE *位被激活后将发送一个空闲帧。*

可配置的停止位

随每个字符发送的停止位的位数可以通过控制寄存器 2 的位 13、12 进行编程。

1. 1 个停止位：停止位位数的默认值。
2. 2 个停止位：可用于常规 UART 模式、单线模式。

空闲帧包括了停止位。

断开帧是 10 位低电平，后跟停止位（当 m=0 时）；或者 11 位低电平，后跟停止位（m=1 时）。不可能传输更长的断开帧（长度大于 10 或者 11 位）。

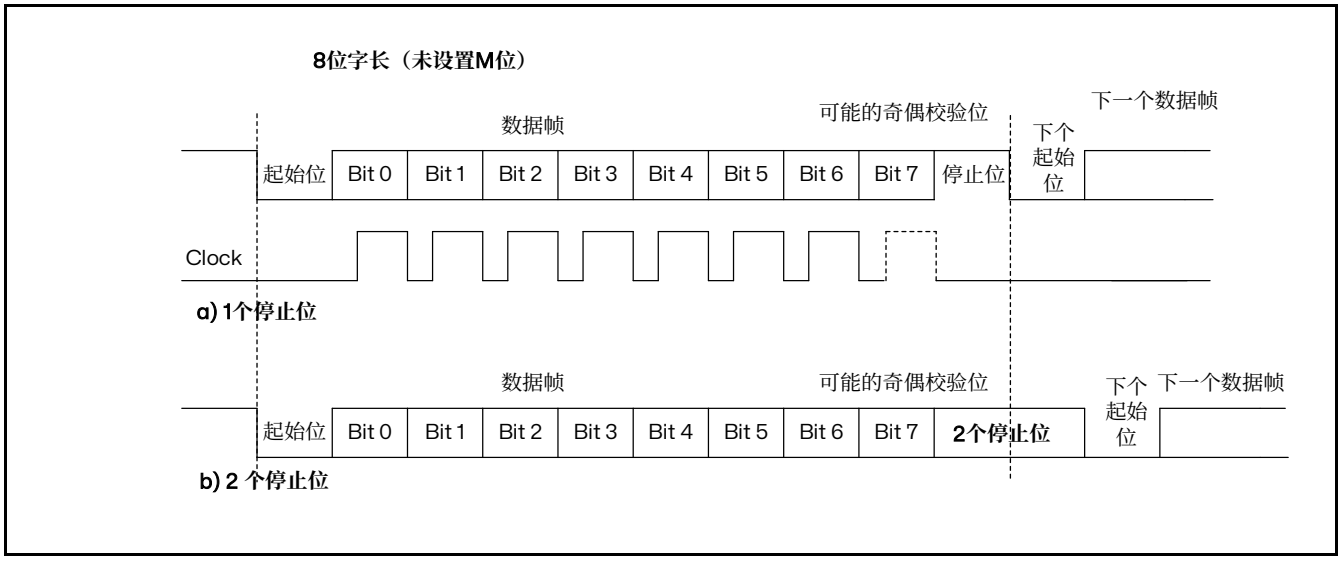


图 24.3 配置停止位

配置步骤：

1. 通过在 UART_CR1 寄存器上置位 UE 位来激活 UART。
2. 编程 UART_CR1 的 M 位来定义字长。
3. 在 UART_CR2 中编程停止位的位数。
4. 利用 UART_BRR 寄存器选择要求的波特率。
5. 设置 UART_CR1 中的 TE 位，发送一个空闲帧作为第一次数据发送。
6. 把要发送的数据写进 UART_DR 寄存器（此动作清除 TXE 位）。在只有一个缓冲器的情况下，对每个待发送的数据重复步骤 6。
7. 在 UART_DR 寄存器中写入最后一个数据字后，要等待 TC=1，它表示最后一个数据帧的传输结束。当需要关闭 UART 或需要进入停机模式之前，需要确认传输结束，避免破坏最后一次传输。

单字节通信

清零 TXE 位总是通过对数据寄存器的写操作来完成的。TXE 位由硬件来设置，它表明：

- 数据已经从 TDR 移送到移位寄存器，数据发送已经开始
- TDR 寄存器被清空
- 下一个数据可以被写进 UART_DR 寄存器而不会覆盖先前的数据

如果 TXEIE 位被设置，此标志将产生一个中断。

如果此时 UART 正在发送数据，对 UART_DR 寄存器的写操作把数据存进 TDR 寄存器，并在当前传输结束时把该数据复制进移位寄存器。

如果此时 UART 没有在发送数据，处于空闲状态，对 UART_DR 寄存器的写操作直接把数据放进移位寄存器，数据传输开始，TXE 位立即被置起。

当一帧发送完成时（停止位发送后）并且设置了 TXE 位，TC 位被置起，如果 UART_CR1 寄存器中的 TCIE 位被置起时，则会产生中断。

在 UART_DR 寄存器中写入了最后一个数据字后，在关闭 UART 模块之前或设置微控制器进入低功耗模式之前，必须先等待 TC=1。

使用下列软件过程清除 TC 位：

1. 读一次 UART_SR 寄存器；
2. 写一次 UART_DR 寄存器。

注：TC 位也可以通过软件对它写‘0’来清除。此清零方式只推荐在多缓冲器通信模式下使用。

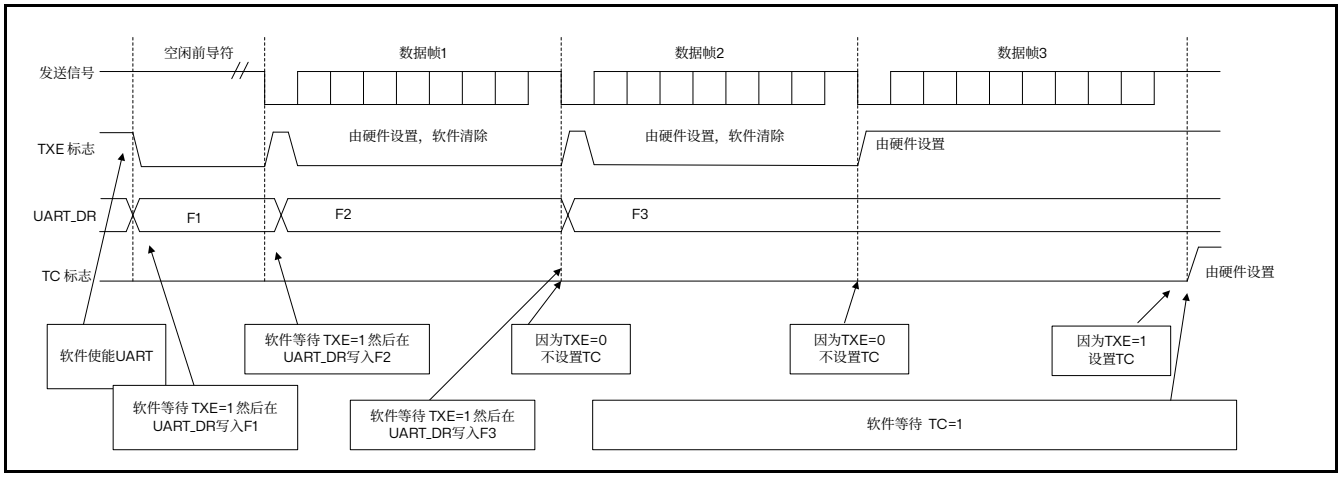


图 24.4 发送时 TC/TXE 的变化情况

断开符号

设置 SBK 可发送一个断开符号。断开帧长度取决 M 位。如果设置 SBK=1，在完成当前数据发送后，将在 TX 线上发送一个断开符号。断开字符发送完成时（在断开符号的停止位时）SBK 被硬件复位。UART 在最后一个断开帧的结束处插入一逻辑‘1’，以保证能识别下一帧的起始位。

注意：如果在开始发送断开帧之前，软件又复位了 SBK 位，断开符号将不被发送。如果要发送两个连续的断开帧，SBK 位应该在前一个断开符号的停止位之后置起。

空闲符号

置位 TE 将使得 UART 在第一个数据帧前发送一空闲帧。

24.3.3 接收器

UART 可以根据 UART_CR1 的 M 位接收 8 位或 9 位的数据字。

起始位侦测

在 UART 中，如果辨认出一个特殊的采样序列，那么就认为侦测到一个起始位。
该序列为：1110X0X0X0000

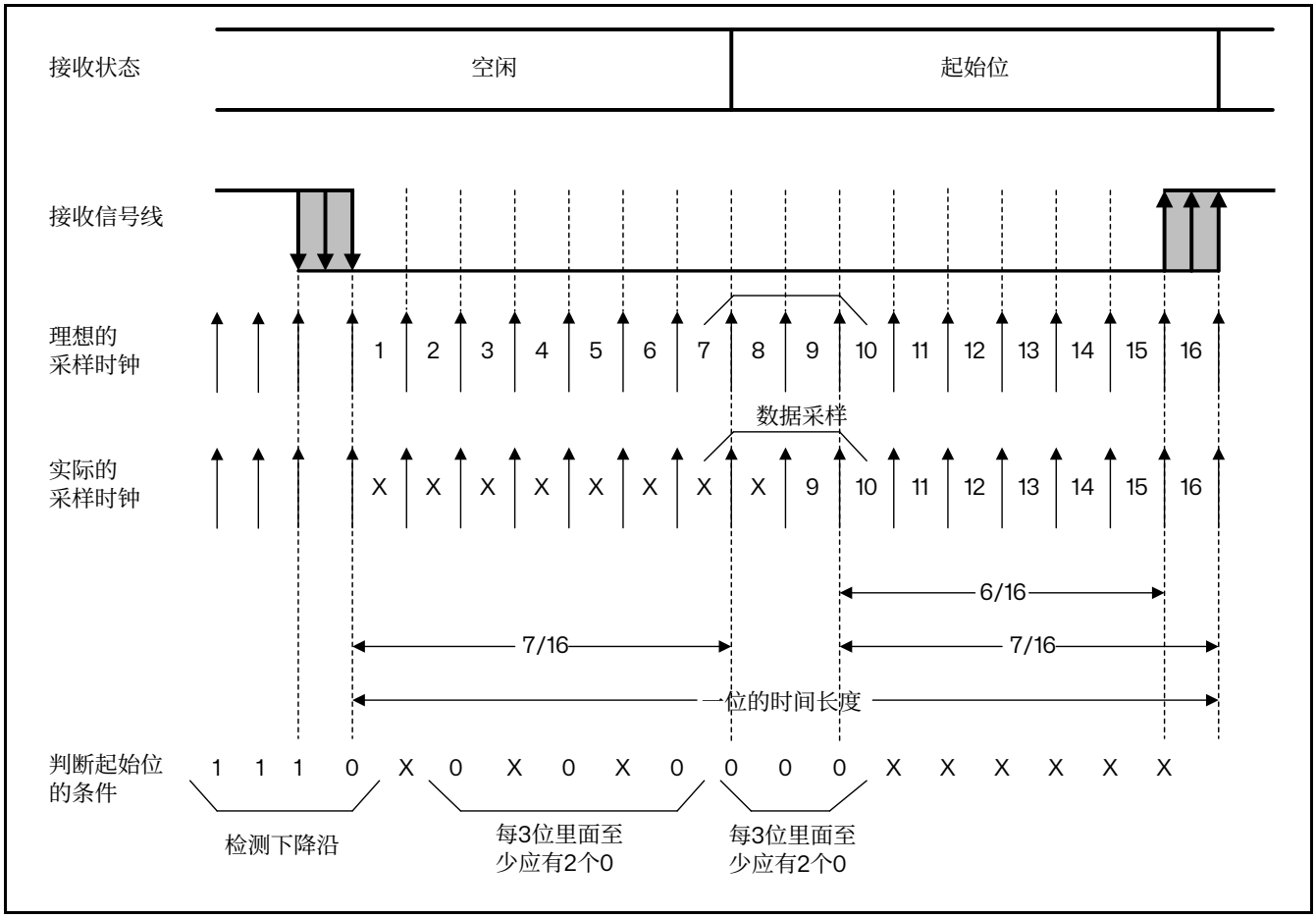


图 24.5 起始位侦测

注：如果该序列不完整，那么接收端将退出起始位侦测并回到空闲状态（不设置标志位）等待下降沿。

如果3 个采样点都为 ‘0’（在第3、5、7 位的第一次采样，和在第8、9、10 的第二次采样都为0’），则确认收到起始位，这时设置RXNE 标志位，如果RXNEIE=1，则产生中断。

如果两次3 个采样点上仅有2 个是 ‘0’（第3、5、7 位的采样点和第8、9、10 位的采样点），那么起始位仍然是有效的，但是会设置NE 噪声标志位。如果不能满足这个条件，则中止起始位的侦测过程，接收器会回到空闲状态（不设置标志位）。

如果有一次3 个采样点上仅有2 个是 ‘0’（第3、5、7 位的采样点或第8、9、10 位的采样点），那么起始位仍然是有效的，但是会设置NE 噪声标志位。

字符接收

在 UART 接收期间，数据的最低有效位首先从 RX 脚移进。在此模式里，UART_DR 寄存器包含的缓

冲器位于内部总线和接收移位寄存器之间。

配置步骤：

1. 将 UART_CR1 寄存器的 UE 置 1 来激活 UART。
2. 编程 UART_CR1 的 M 位定义字长
3. 在 UART_CR2 中编写停止位的个数
4. 利用波特率寄存器 UART_BRR 选择希望的波特率。
5. 设置 UART_CR1 的 RE 位。激活接收器，使它开始寻找起始位。当一字符被接收到时，

- RXNE 位被置位。它表明移位寄存器的内容被转移到 RDR。换句话说，数据已经被接收并且可以被读出（包括与之有关的错误标志）。
- 如果 RXNEIE 位被设置，产生中断。
- 在接收期间如果检测到帧错误，噪音或溢出错误，错误标志将被置起，
- 在单缓冲器模式里，由软件读 UART_DR 寄存器完成对 RXNE 位清除。RXNE 标志也可以通过对它写 0 来清除。RXNE 位必须在下一字符接收结束前被清零，以避免溢出错误。

注意：在接收数据时，RE 位不应该被复位。如果 RE 位在接收时被清零，当前字节的接收被丢失

断开符号

当接收到一个断开帧时，UART 像处理帧错误一样处理它。

溢出错误

如果 RXNE 还没有被复位，又接收到一个字符，则发生溢出错误。数据只有当 RXNE 位被清零后才能从移位寄存器转移到 RDR 寄存器。RXNE 标记是接收到每个字节后被置位的。

当溢出错误产生时：

- ORE 位被置位。
- RDR 内容将不会丢失。读 UART_DR 寄存器仍能得到先前的数据。
- 移位寄存器中以前的内容将被覆盖。随后接收到的数据都将丢失。
- 顺序执行对 UART_SR 和 UART_DR 寄存器的读操作，可复位 ORE 位

注意：当 ORE 位置位时，表明至少有 1 个数据已经丢失。有两种可能性：

- 如果 RXNE=1，上一个有效数据还在接收寄存器 RDR 上，可以被读出。
- 如果 RXNE=0，这意味着上一个有效数据已经被读走，RDR 已经没有东西可读。当上一个有效数据在 RDR 中被读取的同时又接收到新的（也就是丢失的）数据时，此种情况可能发生。在读序列期间（在 UART_SR 寄存器读访问和 UART_DR 读访问之间）接收到新的数据，此种情况也可能发生。

噪音错误

使用过采样技术（同步模式除外），通过区别有效输入数据和噪音来进行数据恢复。

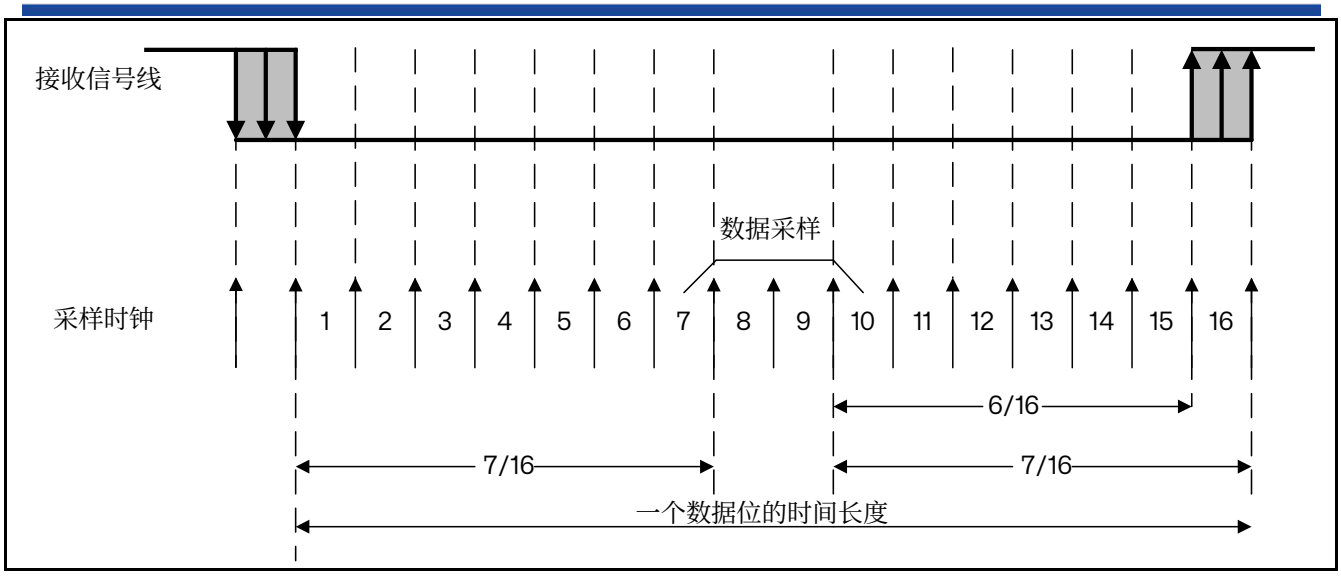


图 24.6 检测噪声的数据采样

表 24.1 检测噪声的数据采样

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

当在接收帧中检测到噪音时：

- 在 RXNE 位的上升沿设置 NE 标志。
- 无效数据从移位寄存器传送到 UART_DR 寄存器。
- 在单个字节通信情况下，没有中断产生。然而，因为 NE 标志位和 RXNE 标志位是同时被设置，RXNE 将产生中断。在多缓冲器通信情况下，如果已经设置了 UART_CR3 寄存器中 EIE 位，将产生一个中断。

先读出 UART_SR，再读出 UART_DR 寄存器，将清除 NE 标志位

帧错误

当以下情况发生时检测到帧错误：
由于没有同步上或大量噪音的原因，停止位没有在预期的时间上接和收识别出来。
当帧错误被检测到时：

- FE 位被硬件置起
- 无效数据从移位寄存器传送到 UART_DR 寄存器。
- 在单字节通信时，没有中断产生。然而，这个位和 RXNE 位同时置起，后者将产生中断。在多缓冲器通信情况下，如果 UART_CR3 寄存器中 EIE 位被置位的话，将产生中断。

顺序执行对 UART_SR 和 UART_DR 寄存器的读操作，可复位 FE 位。

接收期间的可配置的停止位

被接收的停止位的个数可以通过控制寄存器 2 的控制位来配置，在正常模式时，可以是 1 或 2 个。

1. 1 个停止位：对 1 个停止位的采样在第 8，第 9 和第 10 采样点上进行。
2. 2 个停止位：对 2 个停止位的采样是在第一停止位的第 8，第 9 和第 10 个采样点完成的。如果第一个停止位期间检测到一个帧错误，帧错误标志将被设置。第二个停止位不再检查帧错误。在第一个停止位结束时 RXNE 标志将被设置。

24.3.4 分数波特率的产生

接收器和发送器的波特率在 UARTDIV 的整数和小数寄存器中的值应设置成相同。

$$\text{Tx / Rx 波特率} = \frac{f_{ck}}{(16*UARTDIV)}$$

这里的f_{ck}是给外设的时钟（PCLK1 用于 UART2、3、4、5，PCLK2 用于 UART1）

UARTDIV 是一个无符号的定点数。这 12 位的值设置在 UART_BRR 寄存器。

注：在写入 UART_BRR 之后，波特率计数器会被波特率寄存器的新值替换。因此，不要在通信进行中改变波特率寄存器的数值。

如何从 UART_BRR 寄存器值得到 UARTDIV

示例1：

如果 DIV_Mantissa = 0d27，DIV_Fraction = 0d12（UART_BRR=0x1BC），于是
尾数（UARTDIV）= 0d27
小数（UARTDIV）= 12/16 = 0d0.75
所以 UARTDIV = 0d27.75

示例2：

要求 UARTDIV = 0d25.62,
就有：
DIV_Fraction = 16*0d0.62 = 0d9.92
最接近的整数是：0d10 = 0x0A
DIV_Mantissa = 尾数（0d25.620）= 0d25 = 0x19
于是，UART_BRR = 0x19A，因此 USARTDIV = 0d25.625

示例3：

要求 UARTDIV = 0d50.99
就有：
DIV_Fraction = 16*0d0.99 = 0d15.84
最接近的整数是：0d16 = 0x10 => DIV_frac[3:0]溢出 => 尾数必须添加进位
DIV_Mantissa = 尾数（0d50.990 + 进位）= 0d51 = 0x33
于是：UART_BRR = 0x330，UARTDIV = 0d51.000

表 24.2 设置波特率时的误差计算

波特率		f_{PCLK} = 76MHz			f_{PCLK} = 152MHz		
序号	Kbps	实际	置于波特率寄存器中的值	误差%	实际	置于波特率寄存器中的值	误差%
1	2.4	2.3999	1979.1875	0.0011%	2.4000	3958.3125	0.0005%
2	9.6	9.5995	494.8125	0.0042%	9.6002	989.5625	0.0021%
3	19.2	19.2016	247.375	0.0084%	19.1991	494.8125	0.0042%
4	38.4	38.4032	123.6875	0.0084%	38.4032	247.375	0.0084%
5	115.2	115.1515	41.25	0.0421%	115.2388	82.4375	0.0337%
6	230.4	230.3030	20.625	0.0421%	230.3030	41.25	0.0421%
7	460.8	460.6060	10.3125	0.0421%	460.6060	20.625	0.0421%
8	921.6	926.8292	5.125	0.5674%	921.2121	10.3125	0.0421%
9	4750	4750	1	0%	4750	2	0%
10	9500	不可能	不可能	不可能	9500	1	0%

注：

1. CPU *时钟越低，特定波特率的精度就越低。可达到的波特率的上限可使用这些数据进行确定。*
2. *只有*UART1 *使用*PCLK2*（最高*152MHz*）。其它*UART2 *使用*PCLK1*（最高*80MHz*）。*

24.3.5 UART 接收器容忍时钟的变化

只有当整体的时钟系统地变化小于 UART 异步接收器能够容忍的范围，UART 异步接收器才能正常地工作。影响这些变化的因素有：

- DTRA：由于发送器误差而产生的变化（包括发送器端振荡器的变化）
- DQUANT：接收器端波特率取整所产生的误差
- DREC：接收器端振荡器的变化
- DTCL：由于传输线路产生的变化（通常是由于收发器在由低变高的转换时序，与由高变低转 换时序之间的不一致性所造成）。

需要满足：DTRA + DQUANT + DREC + DTCL < UART 接收器的容忍度

对于正常接收数据，UART 接收器的容忍度等于最大能容忍的变化，它依赖于下述选择：

- 由 UART_CR1 寄存器的 M 位定义的 10 或 11 位字符长度
- 是否使用分数波特率产生

表 24.3 当 DIV_Fraction=0 时，UART 接收器的容忍度

M 位	认为 NF 是错误	不认为 NF 是错误
0	3.75%	4.375%
1	3.41%	3.97%

表 24.4 当 DIV_Fraction!=0 时，UART 接收器的容忍度

M 位	认为 NF 是错误	不认为 NF 是错误
0	3.33%	3.88%
1	3.03%	3.53%

*注：在特殊的情况下，即当收到的帧包含一些在*M=0 *时，正好是*10 *位（*M=1 *时是*11 *位）的空闲*

帧，上面2 个表格中的数据可能会有少许出入。

24.3.6 校验控制

设置 UART_CR1 寄存器上的 PCE 位，可以使能奇偶控制（发送时生成一个奇偶位，接收时进行奇偶校验）。根据 M 位定义的帧长度，可能的 UART 帧格式列在下表中。

表 24.5 帧格式

M 位	PCE 位	UART 帧
0	0	起始位 | 8 位数据 | 停止位 |
0	1	起始位 | 7 位数据 | 奇偶检验位 | 停止位 |
1	0	起始位 | 9 位数据 | 停止位 |
1	1	起始位 | 8 位数据 | 奇偶检验位 | 停止位 |

*注意：在用地址标记唤醒设备时，地址的匹配只考虑到数据的*MSB *位，而不用关心校验位。(*MSB *是数 据位中最后发出的，后面紧跟校验位或者停止位)*

偶校验：校验位使得一帧中的 7 或 8 个 LSB 数据以及校验位中‘1’的个数为偶数。例如：数据=00110101，有 4 个‘1’，如果选择偶校验（在 UART_CR1 中的 PS＝0），校验位将是‘0’。

奇校验：此校验位使得一帧中的 7 或 8 个 LSB 数据以及校验位中‘1’的个数为奇数。例如：数据=00110101，有 4 个‘1’，如果选择奇校验（在 UART_CR1 中的 PS＝1），校验位将是‘1’。

传输模式：如果 UART_CR1 的 PCE 位被置位，写进数据寄存器的数据的 MSB 位被校验位替换后发送出去（如果选择偶校验偶数个‘1’，如果选择奇校验奇数个‘1’）。如果奇偶校验失败，UART_SR 寄存器中的 PE 标志被置‘1’，并且如果 UART_CR1 寄存器的 PEIE 在被预先设置的话，中断产生。

24.3.7 LIN（局域互联网）模式

LIN 模式是通过设置 UART_CR2 寄存器的 LINEN 位选择。在 LIN 模式下，下列位必须保持为 0:

- UART_CR3 寄存器的 STOP[1:0]
- UART_CR3 寄存器的 HDSEL。

LIN 发送

同样步骤适用于 LIN 主发送，但和正常 UART 发送有以下区别:

- 清零 M 位以配置 8 位字长
- 置位 LINEN 位以进入 LIN 模式。这时，置位 SBK 将发送 13 位‘0’作为断开符号。然后发一位‘1’，以允许对下一个开始位的检测。

LIN 接收

当 LIN 模式被使能时，断开符号检测电路被激活。该检测完全独立于 UART 接收器。断开符号只要一出现就能检测到，不管是在总线空闲时还是在发送某数据帧其间，数据帧还未完成，又插入了断开符号的发送。

当接收器被激活时（UART_CR1 的 RE=1），电路监测 RX 上的起始信号。监测起始位的方法同检测断开符号或数据是一样的。当起始位被检测到后，电路对每个接下来的位，在每个位的第 8，9，10 个过采样时钟点上进行采样。如果 10 个（当 UART_CR2 的 LBDL = 0）或 11 个（当 UART_CR2 的 LBDL = 1）连续位都是‘0’，并且又跟着一个定界符，UART_SR 的 LBD 标志被设置。如果 LBDIE 位=1，中断产生。在确认断开符号前，要检查定界符，因为它意味 RX 线已经回到高电平。

如果在第 10 或 11 个采样点之前采样到了‘1’，检测电路取消当前检测并重新寻找起始位。如果 LIN

模式被禁止，接收器继续如正常 UART 那样工作，不需要考虑检测断开符号。

如果 LIN 模式没有被激活（LINEN=0），接收器仍然正常工作于 UART 模式，不会进行断开检测。

如果 LIN 模式被激活（LINEN=1），只要一发生帧错误（也就是停止位检测到‘0’，这种情况出现在断开帧），接收器就停止，直到断开符号检测电路接收到一个‘1’（这种情况发生于断开符号没有完整的发出来），或一个定界符（这种情况发生于已经检测到一个完整的断开符号）。

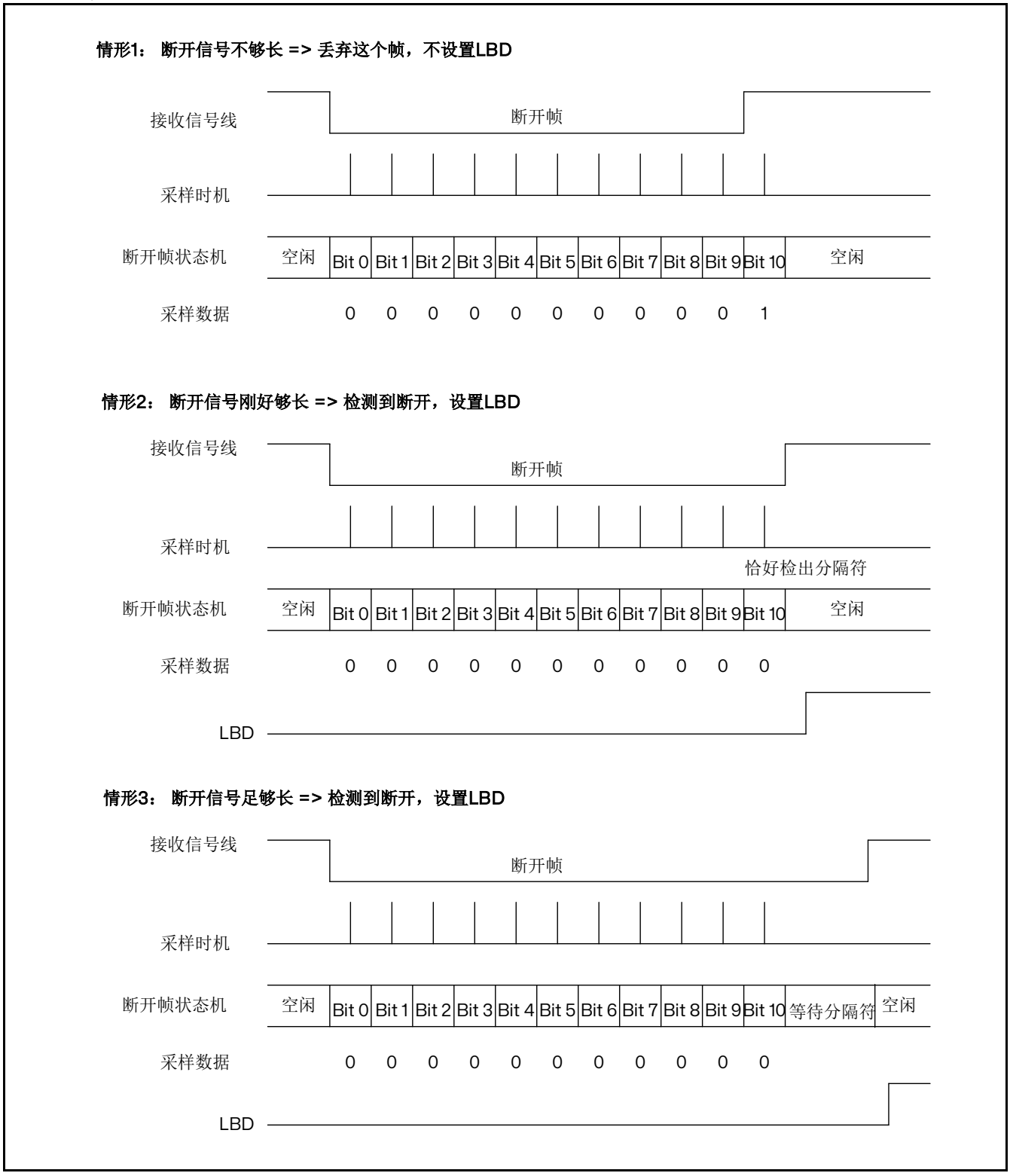
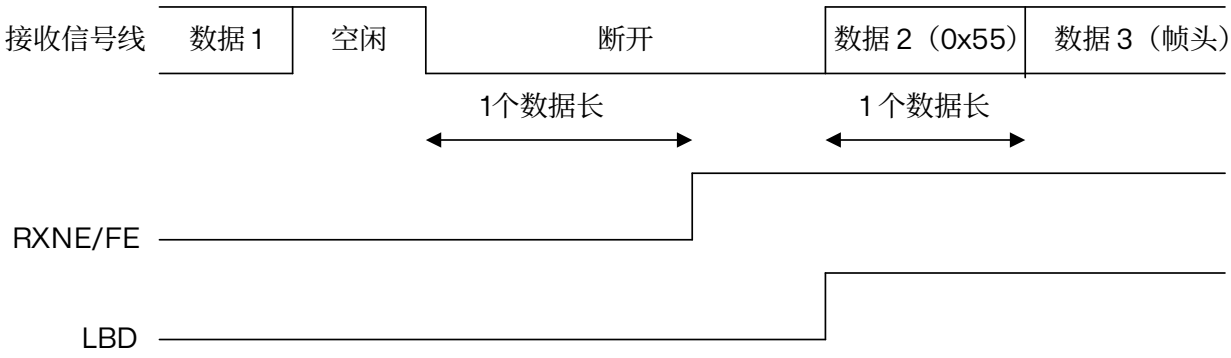


图 24.7 LIN 模式下的断开检测（11 位断开长度 – 设置了 LBDL 位）

下面的例子中，假定 LBDL=1（断开帧长度为11位）,M=0（8位数据）

情形 1：断开发生在空闲之后



情形 2：断开发生在正在接收数据时

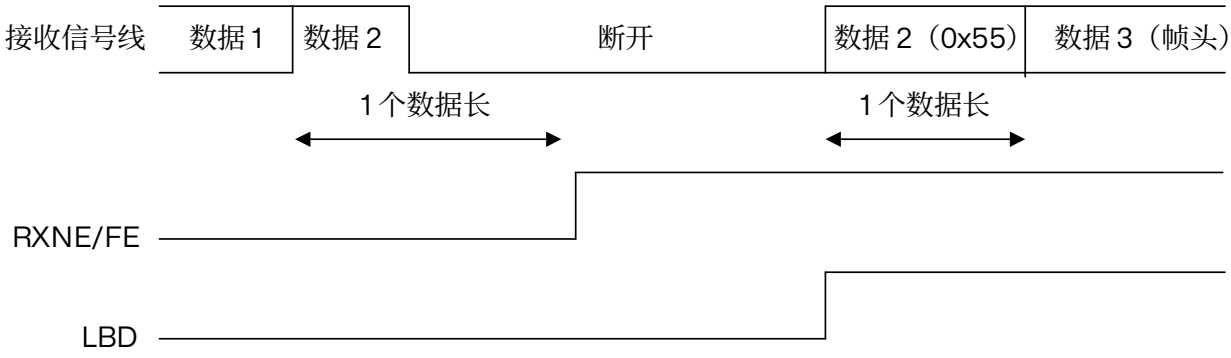


图 24.8 LIN 模式下的断开检测与帧错误的检测

24.3.8 单线半双工通信

单线半双工模式通过设置 UART_CR3 寄存器的 HDSEL 位选择。在这个模式里，下面的位必须保持清零状态:

- UART_CR2 寄存器的 LINEN 位

UART 可以配置成遵循单线半双工协议。在单线半双工模式下，TX 和 RX 引脚在芯片内部互连。使用控制位”HALF DUPLEX SEL”（UART_CR3 中的 HDSEL 位）选择半双工和全双工通信。当 HDSEL 为‘1’时

- RX 不再被使用
- 当没有数据传输时，TX 总是被释放。因此，它在空闲状态的或接收状态时表现为一个标准 I/O 口。这就意味该 I/O 在不被 UART 驱动时，必须配置成悬空输入（或开漏的输出高）。

除此以外，通信与正常 UART 模式类似。由软件来管理线上的冲突（例如通过使用一个中央仲裁器）。特别的是，发送从不会被硬件所阻碍。当 TE 位被设置时，只要数据一写到数据寄存器上，发送就继续。

24.4 UART 中断请求

表 24.6 UART 中断请求

中断事件	事件标志	使能位
发送数据寄存器空	TXE	TXEIE
发送完成	TC	TCIE
接收数据就绪可读	RXNE	RXNEIE
检测到数据溢出	ORE	
奇偶检验错	PE	PEIE
断开标志	LBD	LBDIE
噪声标志，多缓冲通信中的溢出错误和帧错误	NE 或 FE	EIE

UART 的各种中断事件被连接到同一个中断向量（见下图），有以下各种中断事件：

- 发送期间：发送完成、清除发送。
- 接收期间：溢出错误、接收数据寄存器非空、校验错误、LIN 断开符号检测、噪音标志（仅在多缓冲器通信）和帧错误（仅在多缓冲器通信）。

如果设置了对应的使能控制位，这些事件就可以产生各自的中断。

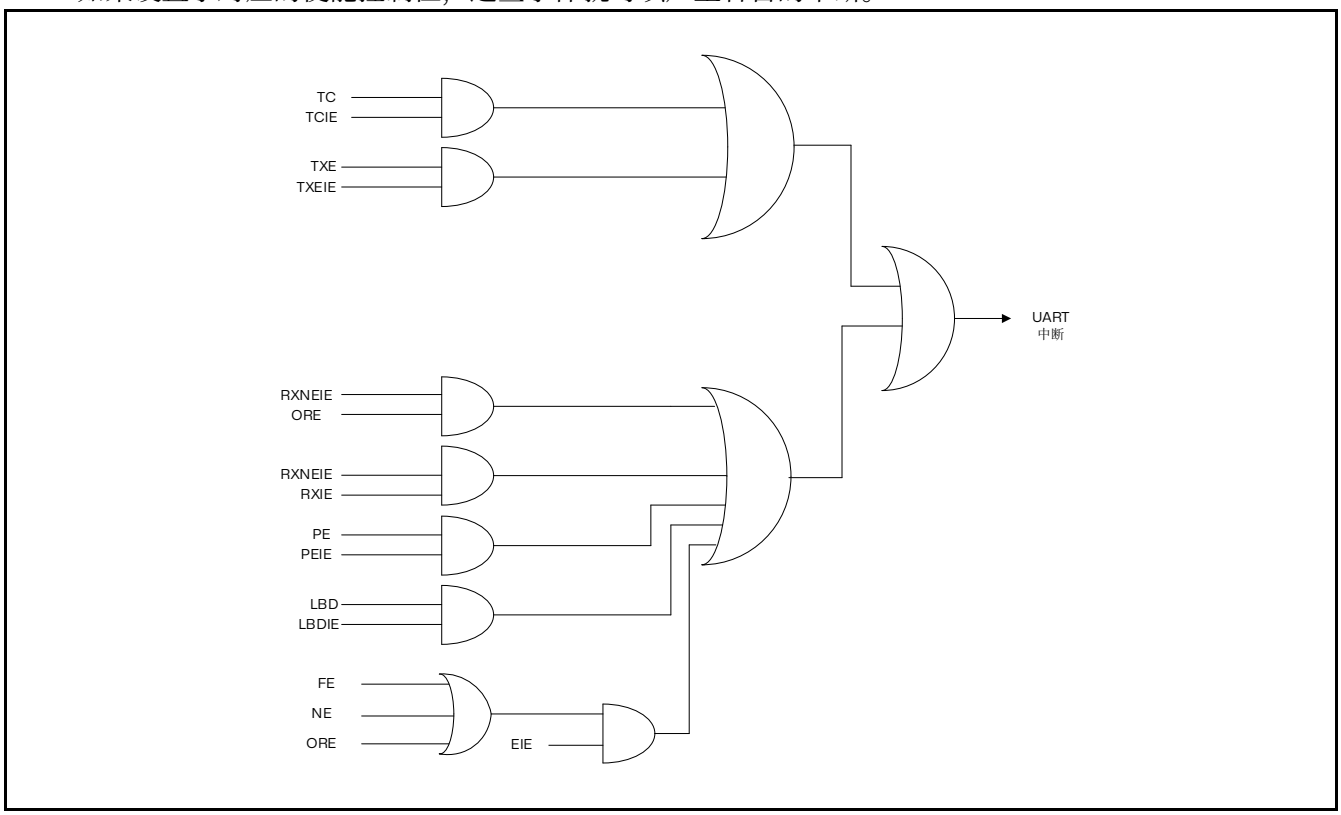


图 24.9 UART 中断映像图

24.5 UART 模式配置

表 24.7 UART 模式设置 (1)

USART 模式	UART1
异步模式	√
硬件流控制	×
多缓存通讯（DMA）	×
同步	×
半双工（单线模式）	√
LIN	√

（1）√ = 支持，× = 不支持该应用

24.1 UART 寄存器

访问：无等待，支持字，半字和字节访问

24.1.1 状态寄存器（UART_SR）

地址偏移：0x00

复位值：0x0000 00C0

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							LBD	TXE	TC	RXNE	Res	ORE	NE	FE	PE
							rc w0	r	rc w0	rc w0		r	r	r	r

Bits 31:9 保留，必须保持为复位值。

Bit 8 **LBD**：LIN 断开检测标志（LIN break detection flag）

当探测到 LIN 断开时，该位由硬件置‘1’，由软件清‘0’（向该位写 0）。如果 UART_CR3 中的 LBDIE = 1，则产生中断。

0：没有检测到 LIN 断开；

1：检测到 LIN 断开。

Bit 7 **TXE**：发送数据寄存器空（Transmit data register empty）

当 TDR 寄存器中的数据被硬件转移到移位寄存器的时候，该位被硬件置位。如果 UART_CR1 寄存器中的 TXEIE 为 1，则产生中断。对 UART_DR 的写操作，将该位清零。

0：数据还没有被转移到移位寄存器；

1：数据已经被转移到移位寄存器。

Bit 6 **TC**：发送完成（Transmission complete）

当包含有数据的一帧发送完成后，并且 TXE=1 时，由硬件将该位置‘1’。如果 UART_CR1 中的 TCIE 为‘1’，则产生中断。由软件序列清除该位（先读 UART_SR，然后写入 UART_DR）。TC 位也可以通过写入‘0’来清除，只有在多缓存通讯中才推荐这种清除程序。

0：发送还未完成；

1：发送完成。

Bit 5 **RXNE**：读数据寄存器非空（Read data register not empty）

当 RDR 移位寄存器中的数据被转移到 UART_DR 寄存器中，该位被硬件置位。如果 UART_CR1 寄存器中的 RXNEIE 为 1，则产生中断。对 UART_DR 的读操作可以将该位清零。RXNE 位也可以通过写入 0 来清除，只有在多缓存通讯中才推荐这种清除程序。

0：数据没有收到；

1：收到数据，可以读出。

Bit 4 保留，必须保持为复位值。

Bit 3 **ORE**：过载错误（Overrun error）

当 RXNE 仍然是‘1’的时候，当前被接收在移位寄存器中的数据，需要传送至 RDR 寄存器时，硬件将该位置位。如果 UART_CR1 中的 RXNEIE 为‘1’的话，则产生中断。由软件序列将其清零（先读 UART_SR，然后读 UART_CR）。

0：没有过载错误；

1：检测到过载错误。

Bit 2 **NE**：噪声错误标志（Noise error flag）

在接收到的帧检测到噪音时，由硬件对该位置位。由软件序列对其清零（先读 UART_SR，再读 UART_DR）。

0：没有检测到噪声；

1：检测到噪声。

Bit 1 **FE**：帧错误（Framing error）

当检测到同步错位，过多的噪声或者检测到断开符，该位被硬件置位。由软件序列将其清零（先读 UART_SR，再读 UART_DR）。

0：没有检测到帧错误；

1：检测到帧错误或者 break 符。

Bit 0 **PE**：校验错误（Parity error）

在接收模式下，如果出现奇偶校验错误，硬件对该位置位。

由软件序列对其清零（依次读 UART_SR 和 UART_DR）。在清除 PE 位前，软件必须等待 RXNE 标志位被置‘1’。如果 UART_CR1 中的 PEIE 为‘1’，则产生中断。

0：没有奇偶校验错误；

1：奇偶校验错误。

24.1.2 数据寄存器（UART_DR）

地址偏移：0x04

复位值：0x0000

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

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

Bits 31:9 保留，必须保持为复位值。

Bits 8:0 **DR[8:0]：**数据值（Data value）

包含了发送或接收的数据。由于它是由两个寄存器组成的，一个给发送用（TDR），一个给接收用（RDR），该寄存器兼具读和写的功能。TDR 寄存器提供了内部总线和输出移位寄存器之间的并行接口。RDR 寄存器提供了输入移位寄存器和内部总线之间的并行接口。

当使能校验位（UART_CR1 中 PCE 位被置位）进行发送时，写到 MSB 的值（根据数据的长度不同，MSB 是第 7 位或者第 8 位）会被后来的校验位该取代。

当使能校验位进行接收时，读到的 MSB 位是接收到的校验位。

24.1.3 波特比率寄存器（UART_BRR）

注意：如果 TE *或* RE *被分别禁止，波特计数器停止计数*

地址偏移：0x08

复位值：0x0000 0000

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

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

Bits 31:16 保留，必须保持为复位值。

Bits 15:4 **DIV_Mantissa[11:0]：**UARTDIV 的整数部分

这 12 位定义了 UART 分频器除法因子（UARTDIV）的整数部分。

Bits 3:0 **DIV_Fraction[3:0]：**UARTDIV 的小数部分

这 4 位定义了 UART 分频器除法因子（UARTDIV）的小数部分。

24.1.4 控制寄存器 1（UART_CR1）

地址偏移：0x0C

复位值：0x0000 0000

31	31	30	29	28	27	26	25	24	23	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		UE	M	Res.	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	Res	TE	RE	Res.	SBK
		rw	rw		rw	rw	rw	rw	rw	rw		rw	rw		rw

Bits 31:14 保留，必须保持为复位值。

Bit 13 **UE**：UART 使能（UART enable）

当该位被清零，在当前字节传输完成后 UART 的分频器和输出停止工作，以减少功耗。该位由软件设置和清零。

0：UART 分频器和输出被禁止；

1：UART 模块使能。

Bit 12 **M**：字长（Word length）

该位定义了数据字的长度，由软件对其设置和清零

0：一个起始位，8 个数据位，n 个停止位；

1：一个起始位，9 个数据位，n 个停止位。

Bit 11 保留，必须保持为复位值。

Bit 10 **PCE**：检验控制使能（Parity control enable）

用该位选择是否进行硬件校验控制（对于发送来说就是校验位的产生；对于接收来说就是校验位的检测）。当使能了该位，在发送数据的最高位（如果 M=1，最高位就是第 9 位；如果 M=0，最高位就是第 8 位）插入校验位；对接收到的数据检查其校验位。软件对它置‘1’或清‘0’。一旦设置了该位，当前字节传输完成后，校验控制才生效。

0：禁止校验控制；

1：使能校验控制。

Bit 9 **PS**：校验选择（Parity selection）

当校验控制使能后，该位用来选择是采用偶校验还是奇校验。软件对它置‘1’或清‘0’。当前字节传输完成后，该选择生效。

0：偶校验；

1：奇校验。

Bit 8 **PEIE**：PE 中断使能（PE interrupt enable）

该位由软件设置或清除。

0：禁止产生中断；

1：当 UART_SR 中的 PE 为‘1’时，产生 UART 中断。

Bit 7 **TXEIE**：发送缓冲区空中断使能（TXE interrupt enable）

该位由软件设置或清除。

0：禁止产生中断；

1：当 UART_SR 中的 TXE 为‘1’时，产生 UART 中断。

Bit 6 **TCIE**：发送完成中断使能（Transmission complete interrupt enable）

该位由软件设置或清除。

0：禁止产生中断；

1：当 UART_SR 中的 TC 为‘1’时，产生 UART 中断。

Bit 5 **RXNEIE**：接收缓冲区非空中断使能（RXNE interrupt enable）

该位由软件设置或清除。

0：禁止产生中断；

1：当 UART_SR 中的 ORE 或者 RXNE 为‘1’时，产生 UART 中断。

Bit 4 保留，必须保持为复位值。

Bit 3 **TE**：发送使能（Transmitter enable）

该位使能发送器。该位由软件设置或清除。

0：禁止发送；

1：使能发送。

注意：1. 在数据传输过程中，如果 TE 位上有个 0 脉冲（即设置为‘0’之后再设置为‘1’），会在当前数据字传输完成后，发送一个“前导符”（空闲总线）。

2. 当 TE 被设置后，在真正发送开始之前，有一个比特时间的延迟。

Bit 2 **RE**：接收使能（Receiver enable）

该位由软件设置或清除。

0：禁止接收；

1：使能接收，并开始搜寻 RX 引脚上的起始位。

Bit 1 保留，必须保持为复位值。

Bit 0 **SBK**：发送断开帧（Send break）

使用该位来发送断开字符。该位可以由软件设置或清除。操作过程应该是软件设置该位，然后在断开帧的停止位时，由硬件将该位复位。

0：没有发送断开字符；

1：将要发送断开字符。

24.1.5 控制寄存器 2（UART_CR2）

地址偏移：0x10

复位值：0x0000 0000

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

Bits 31:15 保留，必须保持为复位值。

Bit 14 LINEN：LIN 模式使能（LIN mode enable）

该位由软件设置或清除。

0：禁止 LIN 模式；

1：使能 LIN 模式。

Bits 13:12 STOP：停止位（STOP bits）

这 2 位用来设置停止位的位数

00：1 个停止位；

10：2 个停止位；

其他：保留。

Bits 11:7 保留，必须保持为复位值。

Bit 6 LBDIE：LIN 断开符检测中断使能（LIN break detection interrupt enable）

0：禁止中断；

1：只要 UART_SR 寄存器中的 LBD 为‘1’就产生中断。

Bit 5 LBDL：LIN 断开符检测长度（LIN break detection length）

0：10 位的断开符检测；

1：11 位的断开符检测。

Bit 4 保留，必须保持为复位值。

Bits 3:0 ADDR[3:0]：本设备的 UART 节点地址

24.1.6 控制寄存器 3（UART_CR3）

地址偏移：0x14

复位值：0x0000 0000

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

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

Bit 31:4 保留，必须保持为复位值。

Bit 3 **HDSEL：**半双工选择（Half-duplex selection）

0：不选择半双工模式；

1：选择半双工模式。

Bits 2:1 保留，必须保持为复位值。

Bit 0 **EIE：**错误中断使能（Error interrupt enable）

0：禁止中断；

1：只要 ORE=1，或者 NE=1，则产生中断。

25 栅极驱动器（GateDriver）

25.1 概述

RX32S632 内部合封 PMOS & NMOS 的三相半桥栅极驱动芯片。HVDD 电压范围：5V 至 28V，VDD 电压范围：2.7V 至 3.6V。

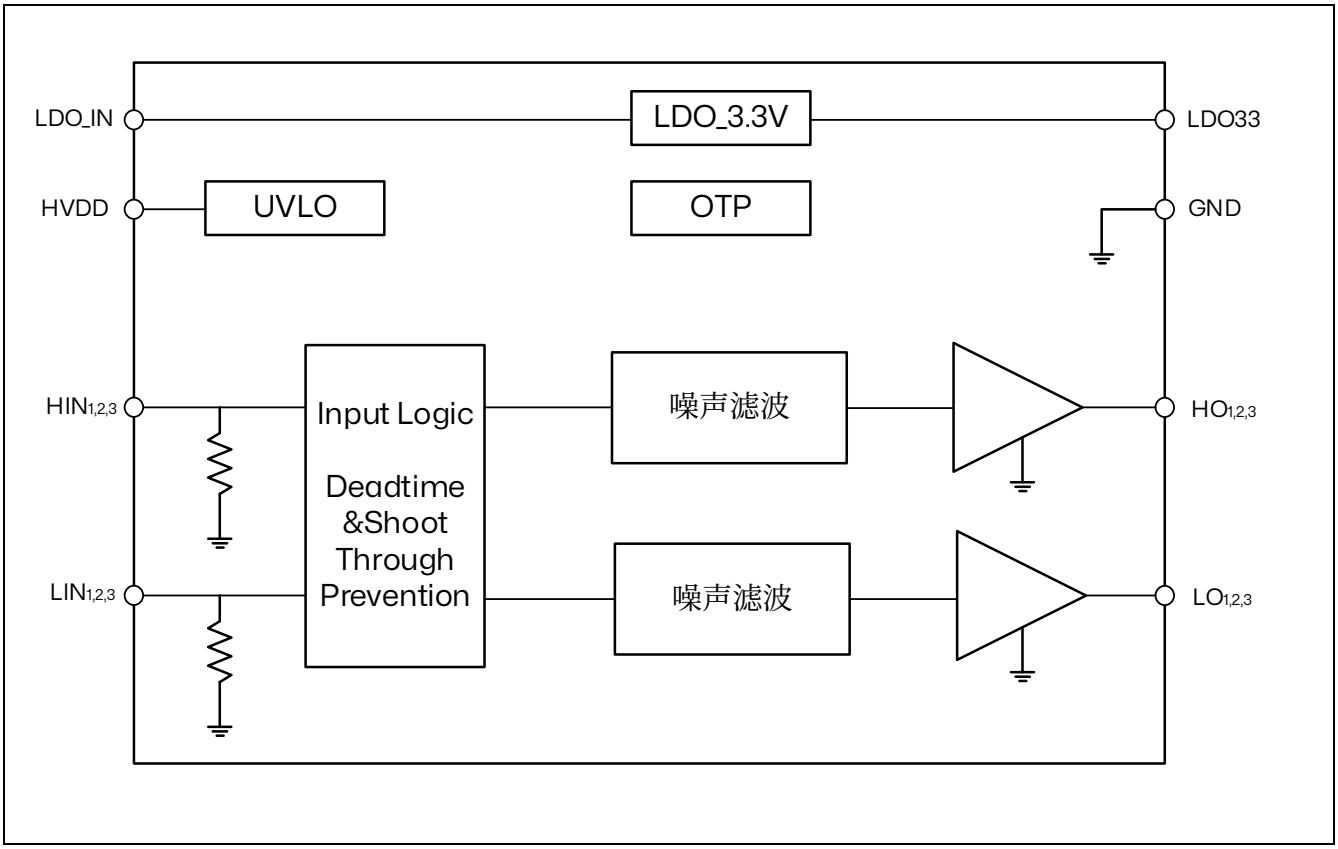


图 25.1 GateDriver 内部结构

25.2 功能描述

25.2.1 特性

- P/N MOS 三相半桥输出
- 最高工作电压可达 40 V
- 内置 LDO 3.3V
- 输出电流能力 IO+/-= +50mA/ -300mA
- 内置过温保护
- 内置欠压锁定
- 高低侧延时匹配
- 内置防直通逻辑保护功能
- 内置死区：DT=130ns

25.3 HO/LO 输出真值表

- HIN 输入低电平有效，控制上桥 HO 输出
- LIN 输入高电平有效，控制下桥 LO 输出

表 25.1 栅极驱动器真值表

HIN	LIN	HO	LO
H	L	L	L
H	H	H	L
L	H	H	H
L	L	H	L
H	L	L	L

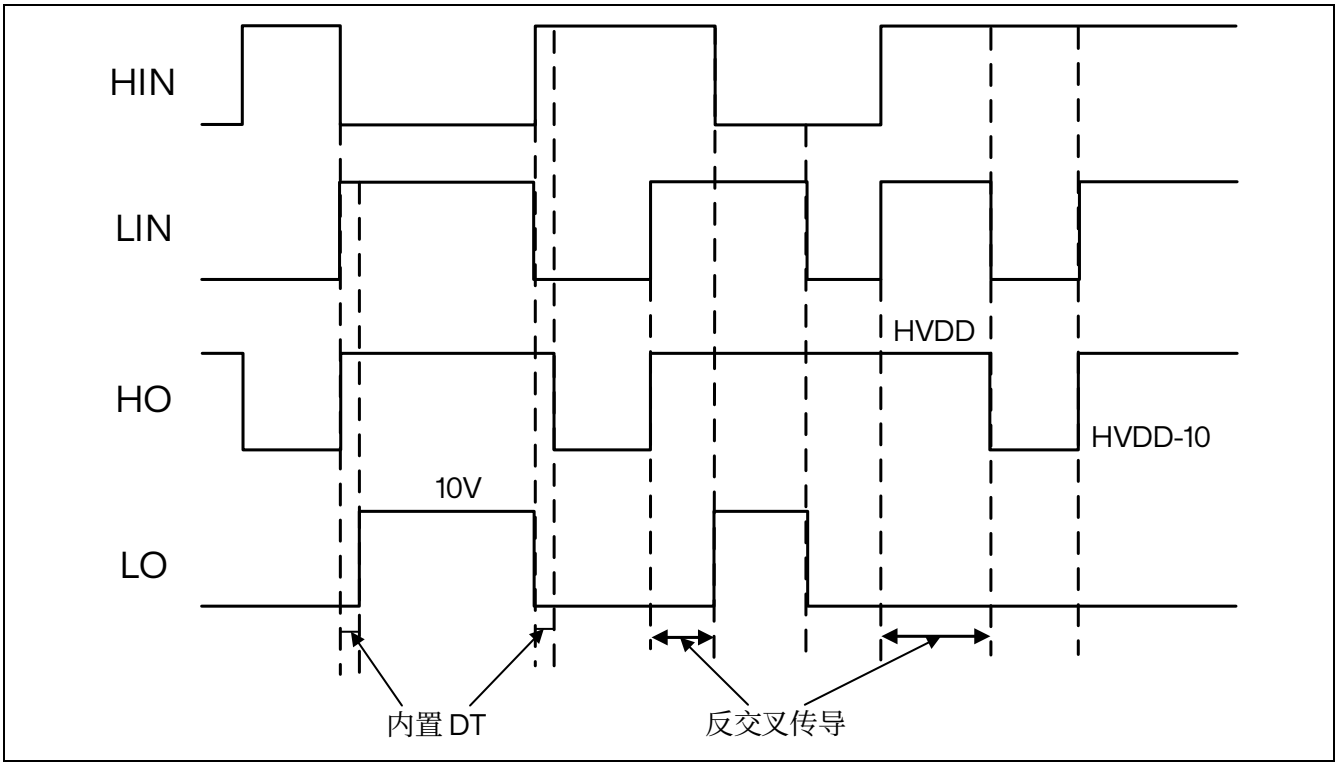


图 25.2 输入输出时序波形

26 器件电子签名

26.1 产品唯一身份标识寄存器（96 位）

产品唯一的身份标识非常适合：

- 用来作为序列号
- 用来作为密码，在编写闪存时，将此唯一标识与软件加解密算法结合使用，提高代码在闪存存储器内的安全性。
- 用来激活带安全机制的自举过程

96 位的产品唯一身份标识所提供的参考号码对任意一个微控制器，在任何情况下都是唯一的。用户在何种情况下，都不能修改这个身份标识。

这个 96 位的产品唯一身份标识，按照用户不同的用法，可以以字节（8 位）为单位读取，也可以以半字（16 位）或者全字（32 位）读取。

基地址：0x1FFF 07E4

地址偏移：0x00

只读，其值在出厂时编写

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

Bits 31:0 **U_ID[31:0]：**唯一身份标志 31:0 位

地址偏移：0x04

只读，其值在出厂时编写

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID[63:48]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID[47:32]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits 31:0 **U_ID[63:32]**：唯一身份标志 63:32 位

地址偏移：0x08

只读，其值在出厂时编写

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID[95:80]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID[79:64]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits 31:0 **U_ID[95:64]**：唯一身份标志 95:64 位

27 调试支持（DBG）

27.1 DBGMCU 寄存器

27.1.1 微控制器设备 ID 编码寄存器（DBGMCU_IDCODE）

地址：0xE004 2000

复位值：0x20036010

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

Bits 31:0 **IDCODE[31:0]**：微控制器设备 ID 编码

这些位只读

27.1.2 调试 MCU 配置寄存器（DBGMCU_CR）

地址：0xE004 2004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							DBG_TIM15_STOP	Reserved			DBG_TIM7_STOP	DBG_TIM6_STOP	Res.	DBG_TIM8_STOP	Res.
							rw				rw	rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			DBG_TIM3_STOP	DBG_TIM2_STOP	Res.		DBG_IWDG_STOP	Reserved					DBG_STANDBY	DBG_STOP	DBG_SLEEP
			rw	rw			rw						rw	rw	rw

Bits 31:25 保留，必须保持为复位值。

Bit 24 **DBG_TIM15_STOP**：当内核停止时停止 TIM15 计数器

0：当内核停止时，仍然向 TIM15 的计数器提供时钟；

1：当内核停止时，停止向 TIM15 的计数器提供时钟。

Bits 23:21 保留，必须保持为复位值。

Bit 20 **DBG_TIM7_STOP**：当内核停止时停止 TIM7 计数器

0：当内核停止时，仍然向 TIM7 的计数器提供时钟；

1：当内核停止时，停止向 TIM7 的计数器提供时钟。

Bit 19 **DBG_TIM6_STOP**：当内核停止时停止 TIM6 计数器

0：当内核停止时，仍然向 TIM6 的计数器提供时钟；

1：当内核停止时，停止向 TIM6 的计数器提供时钟。

Bit 18 保留，必须保持为复位值。

Bit 17 **DBG_TIM8_STOP**：当内核停止时停止 TIM8 计数器

0：当内核停止时，仍然向 TIM8 的计数器提供时钟；

1：当内核停止时，停止向 TIM8 的计数器提供时钟。

Bits 16:13 保留，必须保持为复位值。

Bit 12 **DBG_TIM3_STOP**：当内核停止时停止 TIM3 计数器

0：当内核停止时，仍然向 TIM3 的计数器提供时钟；
1：当内核停止时，停止向 TIM3 的计数器提供时钟。

Bit 11 **DBG_TIM2_STOP**：当内核停止时停止 TIM2 计数器
0：当内核停止时，仍然向 TIM2 的计数器提供时钟；
1：当内核停止时，停止向 TIM2 的计数器提供时钟。

Bits 10:9 保留，必须保持为复位值。

Bit 8 **DBG_IWDG_STOP**：当内核停止时停止 IWDG 计数器
0：当内核停止时，IWDG 计数器时钟继续运行；
1：当内核停止时，IWDG 计数器时钟停止运行。

Bits 7:3 保留，必须保持为复位值。

Bit 2 **DBG_STANDBY**：调试待机模式
0：（FCLK 关，HCLK 关）整个数字电路部分都断电。从软件的观点看，退出 STANDBY 模式与复位是一样的（除了一些状态位指示了微控制器刚从 STANDBY 状态退出）。
1：（FCLK 开，HCLK 开）数字电路部分不下电，FCLK 和 HCLK 时钟由内部 RL 振荡器提供时钟。另外，微控制器通过产生系统复位来退出 STANDBY 模式和复位是一样的；

Bit 1 **DBG_STOP**：调试停止模式
0：（FCLK 关，HCLK 关）整个数字电路部分都断电。从软件的观点看，退出 STANDBY 模式与复位是一样的（除了一些状态位指示了微控制器刚从 STANDBY 状态退出）。
1：（FCLK 开，HCLK 开)在停止模式时，FCLK 和 HCLK 时钟由内部 RC 振荡器提供。当退出停止模式时，软件必需重新配置时钟系统启动 PLL，晶振等（与配置此比特位为 0 时的操作一样);

Bit 0 **DBG_SLEEP**：调试睡眠模式
0：（FCLK 关，HCLK 关）整个数字电路部分都断电。从软件的观点看，退出 STANDBY 模式与复位是一样的（除了一些状态位指示了微控制器刚从 STANDBY 状态退出）。
1：（FCLK 开，HCLK 开）在睡眠模式时，FCLK 和 HCLK 时钟都由原先配置好的系统时钟提供；

28 版本历史

表 28.1 版本历史

日期	版本	更改内容
2026.6.11	V2.0	新版
2026.6.30	V2.1	修正 TIM2 计数器为 28 位 删除 UART IDLE 中断