# CIU32F042x8

## 32 位 ARM Cortex-M0+微控制器

# 参考手册

## RM1014

北京中电华大电子设计有限责任公司
CEC Huada Electronic Design Co.,Ltd

# 声　明

本手册的版权属北京中电华大电子设计有限责任公司所有。任何未经授权对本手册进行复印、印刷、出版发行的行为，都将被视为是对北京中电华大电子设计有限责任公司版权的侵害。北京中电华大电子设计有限责任公司保留对此行为诉诸法律的权利。

北京中电华大电子设计有限责任公司保留未经通知用户对本手册内容进行修改的权利。虽然我们已经核对本手册的内容，但是差错有时候难以完全避免，所以，我们会对手册的内容进行定期的审查，并在下一版的文件中作必要的修改。建议您在最终设计前从华大电子获取本文档的最新版本。

# 目录

1 文档约定 ..... 27
1.1 寄存器列表符号定义 ..... 27
1.2 术语及定义 ..... 27
2 简介 ..... 29
3 系统与存储器架构 ..... 30
3.1 系统架构 ..... 30
3.1.1 System Bus ..... 31
3.1.2 DMA Bus ..... 32
3.1.3 Bus Matrix ..... 32
3.1.4 AHB to APB 总线桥 ..... 32
3.2 存储器 ..... 32
3.2.1 SRAM ..... 32
3.2.2 Flash ..... 32
3.2.3 存储器映射 ..... 32
3.3 启动配置 ..... 36
3.3.1 Bootloader ..... 36
4 Flash 存储器（Flash） ..... 37
4.1 简介 ..... 37
4.2 Flash 主要特性 ..... 37
4.3 Flash 功能描述 ..... 37
4.3.1 Flash 存储器组成 ..... 37
4.3.2 Flash 读取 ..... 38
4.3.3 Flash 解锁 ..... 39
4.3.4 User flash 区擦除操作 ..... 40
4.3.5 User flash 区编程操作 ..... 41
4.4 Option bytes 区操作说明 ..... 42
4.4.1 选项字节组成 ..... 42
4.4.2 选项字节加载 ..... 43
4.4.3 Option bytes 区擦除与编程 ..... 43
4.5 Engineer 区 ..... 44
4.6 Flash 安全保护机制 ..... 45
4.6.1 Flash 读出保护 ..... 45
4.6.2 Flash 写入保护 ..... 49
4.7 Flash 操作错误 ..... 49

4.8 Flash 中断 ..... 50
4.9 Flash 寄存器 ..... 51
4.9.1 Flash 访问控制寄存器（FLASH_ACR） ..... 51
4.9.2 Flash 控制解锁密钥寄存器（FLASH_CRKEY） ..... 51
4.9.3 Flash 选项字节解锁密钥寄存器（FLASH_OPTKEY） ..... 52
4.9.4 Flash 状态寄存器（FLASH_SR） ..... 52
4.9.5 Flash 控制寄存器（FLASH_CR） ..... 54
4.9.6 Flash 选项字节寄存器 1（FLASH_OPTR1） ..... 55
4.9.7 Flash 写入保护寄存器（FLASH_WRP） ..... 55
5 电源管理单元（PMU） ..... 57
5.1 电源 ..... 57
5.2 上电复位和掉电复位 ..... 58
5.3 可编程电压检测（PVD） ..... 58
5.3.1 主要特性 ..... 58
5.3.2 低电压检测 ..... 59
5.3.3 操作示例 ..... 60
5.4 内部带隙基准电压 ..... 61
5.5 低功耗模式 ..... 61
5.5.1 Run ..... 63
5.5.2 Sleep ..... 63
5.5.3 Stop ..... 64
5.6 PMU 寄存器 ..... 66
5.6.1 电源管理控制寄存器 1（PMU_CR1） ..... 66
5.6.2 电源管理控制寄存器 2（PMU_CR2） ..... 66
5.6.3 电源管理状态寄存器（PMU_SR） ..... 69
5.6.4 电源管理状态清除寄存器（PMU_CLR） ..... 69
6 复位和时钟控制器（RCC） ..... 71
6.1 复位 ..... 71
6.1.1 POR/PDR 复位 ..... 71
6.1.2 NRST 复位 ..... 71
6.1.3 系统复位 ..... 72
6.1.4 AWON 域复位 ..... 72
6.2 时钟 ..... 72
6.2.1 RCH 时钟 ..... 73
6.2.2 RCL 时钟 ..... 74
6.2.3 EXTCLK 时钟 ..... 75

6.2.4 LXTAL 时钟 ..... 75
6.2.5 系统时钟 ..... 76
6.2.6 LXTAL CSS ..... 76
6.2.7 外设异步时钟选择 ..... 77
6.2.8 IWDG 时钟 ..... 77
6.2.9 RTC 时钟 ..... 78
6.2.10 时钟输出功能 ..... 78
6.2.11 基于 TIM4 的内/外部时钟测量 ..... 78
6.3 低功耗模式 ..... 79
6.4 RCC 寄存器 ..... 80
6.4.1 时钟控制/状态寄存器 1（RCC_CSR1） ..... 80
6.4.2 时钟配置寄存器（RCC_CFG） ..... 81
6.4.3 时钟中断使能寄存器（RCC_IER） ..... 84
6.4.4 时钟中断标志寄存器（RCC_ISR） ..... 84
6.4.5 时钟中断清除寄存器（RCC_ICR） ..... 85
6.4.6 GPIO 端口复位寄存器（RCC_IOPRST） ..... 85
6.4.7 AHB 外设复位寄存器（RCC_AHBRST） ..... 86
6.4.8 APB1 外设复位寄存器（RCC_APB1RST） ..... 87
6.4.9 APB2 外设复位寄存器（RCC_APB2RST） ..... 89
6.4.10 I/O 端口时钟使能寄存器（RCC_IOPEN） ..... 90
6.4.11 AHB 外设时钟使能寄存器（RCC_AHBEN） ..... 91
6.4.12 APB1 外设时钟使能寄存器（RCC_APB1EN） ..... 92
6.4.13 APB2 外设时钟使能寄存器（RCC_APB2EN） ..... 94
6.4.14 外设异步时钟配置寄存器（RCC_CLKSEL） ..... 95
6.4.15 AWON 域控制寄存器（RCC_AWCR） ..... 96
6.4.16 时钟控制/状态寄存器 2（RCC_CSR2） ..... 98
6.4.17 RCL 校准寄存器（RCC_RCLCAL） ..... 100
6.4.18 RCH 校准寄存器（RCC_RCHCAL） ..... 101
7 通用输入/输出接口（GPIO） ..... 102
7.1 简介 ..... 102
7.2 GPIO 主要特性 ..... 102
7.3 GPIO 功能描述 ..... 102
7.3.1 通用 I/O (GPIO) ..... 104
7.3.2 I/O 引脚功能复用器和映射 ..... 105
7.3.3 I/O 端口配置寄存器 ..... 106
7.3.4 I/O 端口数据寄存器 ..... 106

7.3.5 I/O 数据位操作 ..... 106
7.3.6 外部中断/唤醒 ..... 106
7.3.7 输入配置 ..... 107
7.3.8 输出配置 ..... 108
7.3.9 复用功能配置 ..... 109
7.3.10 复用功能选择 ..... 109
7.3.11 模拟模式配置 ..... 113
7.3.12 外部振荡器引脚用作 GPIO ..... 113
7.3.13 外部时钟输入引脚用作 GPIO ..... 114
7.4 GPIO 寄存器 ..... 115
7.4.1 GPIO 端口模式寄存器（GPIOx_MODE）（x＝A、B、C、F） ..... 115
7.4.2 GPIO 端口输出类型寄存器（GPIOx_OTYPE）（x＝A、B、C、F） ..... 115
7.4.3 GPIO 端口上拉/下拉寄存器（GPIOx_PUPD）（x＝A、B、C、F） ..... 116
7.4.4 GPIO 端口输入寄存器（GPIOx_IDR）（x＝A、B、C、F） ..... 116
7.4.5 GPIO 端口输出数据寄存器（GPIOx_ODR）（x＝A、B、C、F） ..... 117
7.4.6 GPIO 端口置位/复位寄存器（GPIOx_BSR）（x＝A、B、C、F） ..... 117
7.4.7 GPIO 复用功能低位寄存器（GPIOx_AFL）（x＝A、B、F） ..... 118
7.4.8 GPIO 复用功能高位寄存器（GPIOx_AFH）（x＝A、B、C） ..... 118
7.4.9 GPIO 端口位复位寄存器（GPIOx_BR）（x＝A、B、C、F） ..... 119
8 直接存储器访问控制器（DMA） ..... 120
8.1 简介 ..... 120
8.2 DMA 主要特性 ..... 120
8.3 DMA 功能说明 ..... 121
8.3.1 DMA 框图 ..... 121
8.3.2 DMA 通道请求 ..... 121
8.3.3 DMA 传输 ..... 122
8.3.4 优先级 ..... 124
8.3.5 数据位宽 ..... 124
8.3.6 寻址范围 ..... 124
8.3.7 地址递增 ..... 125
8.3.8 循环模式 ..... 125
8.3.9 配置流程 ..... 126
8.3.10 暂停和停止通道 ..... 126
8.3.11 错误标志 ..... 127
8.4 DMA 中断 ..... 127
8.5 DMA 寄存器 ..... 128

8.5.1 DMA 中断状态寄存器（DMA_ISR） 128
8.5.2 DMA 中断标志清除寄存器（DMA_ICR） 130
8.5.3 DMA 通道 x 控制寄存器（DMA_CCx） 131
8.5.4 DMA 通道 x 待传输次数寄存器（DMA_CNDTRx） 133
8.5.5 DMA 通道 x 源地址寄存器（DMA_CSARx） 134
8.5.6 DMA 通道 x 目的地址寄存器（DMA_CDARx） 134
9 嵌套向量中断控制器（NVIC） 136
9.1 简介 136
9.2 主要特征 136
9.3 中断向量表 136
10 扩展中断和事件控制器（EXTI） 139
10.1 简介 139
10.2 EXTI 主要功能 139
10.3 EXTI 模块示意图 139
10.4 EXTI 的主要信号连接及功能 140
10.5 EXTI 通道及信号对应表 141
10.6 EXTI 功能描述 141
10.6.1 可配置通道 142
10.6.2 直接通道 143
10.6.3 I/O 多路选择器 143
10.7 EXTI 寄存器 144
10.7.1 EXTI 上升沿触发选择寄存器（EXTI_RTSR） 144
10.7.2 EXTI 下降沿触发选择寄存器（EXTI_FTSR） 144
10.7.3 EXTI 中断挂起寄存器（EXTI_PIR） 145
10.7.4 EXTI I/O 选择寄存器 1（EXTI_EXTICR1） 145
10.7.5 EXTI I/O 选择寄存器 2（EXTI_EXTICR2） 147
10.7.6 EXTI 中断唤醒屏蔽寄存器（EXTI_IMR） 149
10.7.7 EXTI 事件唤醒屏蔽寄存器（EXTI_EMR） 150
11 循环冗余校验（CRC） 152
11.1 简介 152
11.2 CRC 主要特性 152
11.3 CRC 功能描述 152
11.3.1 CRC 框图 152
11.3.2 CRC 操作说明 152
11.4 CRC 寄存器 154
11.4.1 CRC 控制状态寄存器（CRC_CSR） 154

11.4.2 CRC 结果寄存器（CRC_RDR） ..... 154
11.4.3 CRC 数据寄存器（CRC_DR） ..... 155
12 硬件除法器（HWDIV） ..... 156
12.1 简介 ..... 156
12.2 HWDIV 主要特性 ..... 156
12.3 HWDIV 功能描述 ..... 156
12.4 HWDIV 寄存器 ..... 157
12.4.1 HWDIV 控制寄存器（HWDIV_CR） ..... 157
12.4.2 HWDIV 状态寄存器（HWDIV_SR） ..... 157
12.4.3 HWDIV 被除数寄存器（HWDIV_DVDR） ..... 158
12.4.4 HWDIV 除数寄存器（HWDIV_DVSR） ..... 158
12.4.5 HWDIV 商寄存器（HWDIV_QUOTR） ..... 159
12.4.6 HWDIV 余数寄存器（HWDIV_RMDR） ..... 159
13 硬件开方器（SQRT） ..... 160
13.1 简介 ..... 160
13.2 SQRT 主要特性 ..... 160
13.3 SQRT 功能描述 ..... 160
13.4 SQRT 寄存器 ..... 161
13.4.1 SQRT 状态寄存器（SQRT_SR） ..... 161
13.4.2 SQRT 被开方数寄存器（SQRT_SQR） ..... 161
13.4.3 SQRT 开方根寄存器（SQRT_RDR） ..... 162
14 模数转换器（ADC） ..... 163
14.1 简介 ..... 163
14.2 ADC 主要特性 ..... 163
14.3 ADC 功能描述 ..... 164
14.3.1 ADC 引脚和内部信号 ..... 165
14.3.2 时钟源 ..... 165
14.3.3 使能/禁止控制 ..... 165
14.3.4 转换通道 ..... 166
14.3.5 单次扫描转换模式 ..... 168
14.3.6 循环扫描转换模式 ..... 171
14.3.7 循环间断转换模式 ..... 172
14.3.8 启动/停止转换 ..... 177
14.3.9 转换时序示例 ..... 178
14.3.10 采样时间 ..... 182
14.3.11 数据溢出 ..... 182

14.3.12 DMA 转换数据处理 ..... 184
14.3.13 数据对齐 ..... 184
14.3.14 模拟看门狗 ..... 184
14.3.15 温度传感器 ..... 185
14.3.16 内部带隙基准电压检测 ..... 187
14.3.17 $V_{DDA}$/3 电压检测 ..... 188
14.3.18 OPAMP 通道切换 ..... 189
14.4 ADC 中断 ..... 190
14.5 ADC 寄存器 ..... 191
14.5.1 ADC 控制寄存器（ADC_CR） ..... 191
14.5.2 ADC 配置寄存器 1（ADC_CFG1） ..... 192
14.5.3 ADC 配置寄存器 2（ADC_CFG2） ..... 195
14.5.4 ADC 中断和状态寄存器（ADC_ISR） ..... 196
14.5.5 ADC 中断使能寄存器（ADC_IER） ..... 198
14.5.6 ADC 采样时间寄存器（ADC_SAMPT） ..... 200
14.5.7 ADC 模拟看门狗配置寄存器（ADC_AWDGCR） ..... 201
14.5.8 ADC 模拟看门狗监控电压阈值寄存器（ADC_AWDGTR） ..... 202
14.5.9 ADC 规则序列寄存器 1（ADC_RSEQ1） ..... 202
14.5.10 ADC 规则序列寄存器 2（ADC_RSEQ2） ..... 203
14.5.11 ADC 规则数据寄存器（ADC_DR） ..... 204
14.5.12 ADC 配置寄存器 3（ADC_CFG3） ..... 204
14.5.13 ADC 注入序列段触发选择寄存器（ADC_JTRIGSEL） ..... 205
14.5.14 ADC 注入序列段通道选择寄存器（ADC_JSETSEL） ..... 207
14.5.15 ADC 注入序列寄存器（ADC_JSEQ） ..... 208
14.5.16 ADC 注入数据寄存器 x（ADC_JDRx）（x = 0 ~ 3） ..... 209
14.5.17 ADC OPA 控制寄存器（ADC_OPACR） ..... 209
15 参考电压源（VREFBUF） ..... 212
15.1 简介 ..... 212
15.2 VREFBUF 主要特性 ..... 212
15.3 VREFBUF 功能描述 ..... 212
15.4 VREFBUF 寄存器 ..... 213
15.4.1 VREFBUF 控制和状态寄存器（VREFBUF_CSR） ..... 213
15.4.2 VREFBUF 校准寄存器（VREFBUF_CAL） ..... 214
16 运算放大器（OPAMP） ..... 215
16.1 简介 ..... 215
16.2 OPAMP 主要特性 ..... 215

16.3 OPAMP 功能描述 ..... 216
16.3.1 OPAMP 结构框图 ..... 216
16.3.2 OPAMP 引脚和内部信号 ..... 216
16.3.3 独立运放模式 ..... 217
16.3.4 电压跟随模式 ..... 218
16.3.5 伪差分 PGA 模式 ..... 219
16.3.6 单端 PGA 模式 ..... 220
16.3.7 OPAMP 校准 ..... 222
16.3.8 TIM 控制 OPAMP 输入切换 ..... 223
16.3.9 ADC 控制 OPAMP 输入切换 ..... 224
16.4 OPAMP 寄存器 ..... 225
16.4.1 OPAMP 控制寄存器（OPAMP_CR） ..... 225
16.4.2 OPAMP 写使能寄存器（OPAMP_LOCK） ..... 225
16.4.3 OPAMP1 控制寄存器（OPAMP1_CR） ..... 226
16.4.4 OPAMP1 校准寄存器（OPAMP1_CAL） ..... 228
16.4.5 OPAMP1 状态寄存器（OPAMP1_SR） ..... 229
16.4.6 OPAMP2 控制寄存器（OPAMP2_CR） ..... 229
16.4.7 OPAMP2 校准寄存器（OPAMP2_CAL） ..... 232
16.4.8 OPAMP2 状态寄存器（OPAMP2_SR） ..... 232
16.4.9 OPAMP2 切换配置寄存器（OPAMP2_CFG） ..... 233
17 比较器（COMP） ..... 235
17.1 简介 ..... 235
17.2 COMP 主要特性 ..... 235
17.3 COMP 功能描述 ..... 236
17.3.1 COMP 结构框图 ..... 236
17.3.2 COMP 引脚和内部信号 ..... 236
17.3.3 使能/禁止控制 ..... 237
17.3.4 窗口比较功能 ..... 237
17.3.5 开窗功能 ..... 238
17.3.6 输出极性 ..... 239
17.3.7 数字滤波 ..... 239
17.3.8 迟滞 ..... 240
17.3.9 速度和功耗 ..... 240
17.4 COMP 低功耗模式 ..... 241
17.5 COMP 中断 ..... 241
17.6 COMP 寄存器 ..... 242

17.6.1 COMP 控制寄存器（COMP_CR） 242
17.6.2 COMPx 控制和状态寄存器 1（COMPx_CSR1） 243
17.6.3 COMPx 控制和状态寄存器 2（COMPx_CSR2） 245
17.6.4 COMPx 状态寄存器（COMPx_SR） 246
18 液晶显示控制器（LCD） 248
18.1 简介 248
18.2 LCD 主要特性 248
18.3 LCD 功能描述 249
18.3.1 LCD 结构框图 249
18.3.2 片内 buffer 驱动模式 249
18.3.3 对比度设置 250
18.3.4 频率发生器 251
18.3.5 帧间死区时间 252
18.3.6 闪烁显示 252
18.3.7 驱动波形 253
18.3.8 LCD 显示存储器映射 258
18.4 LCD 中断 259
18.5 低功耗显示 259
18.6 LCD 寄存器 260
18.6.1 LCD 控制寄存器（LCD_CR） 260
18.6.2 LCD 帧控制寄存器（LCD_FCR） 261
18.6.3 LCD 状态寄存器（LCD_SR） 263
18.6.4 LCD 状态清除寄存器（LCD_CLR） 264
18.6.5 LCD 显示存储器 x（LCD_RAMx）（x = 0 ~ 3） 264
19 高级控制定时器（TIM1） 266
19.1 简介 266
19.2 TIM1 主要特性 266
19.3 TIM1 功能描述 267
19.3.1 TIM1 框图 267
19.3.2 TIM1 引脚 267
19.3.3 PSC 预分频器 268
19.3.4 计数器 268
19.3.5 重复计数器 277
19.3.6 计数时钟选择 278
19.3.7 比较通道 278
19.3.8 比较输出模式 279

19.3.9 强制输出模式 281
19.3.10 PWM 模式 281
19.3.11 清除 OCxREF 信号 284
19.3.12 单脉冲模式 285
19.3.13 互补模式、PWM 移相及死区插入 286
19.3.14 刹车及恢复功能 288
19.3.15 输出通道重映射 290
19.3.16 生成 6 步 PWM 290
19.3.17 霍尔传感器接口 291
19.3.18 定时器同步主模式 292
19.3.19 DMA Burst 传输模式 292
19.3.20 调试模式 293
19.4 TIM1 中断 293
19.5 TIM1 寄存器 295
19.5.1 TIM1 控制寄存器 1（TIM1_CR1） 295
19.5.2 TIM1 控制寄存器 2（TIM1_CR2） 296
19.5.3 TIM1 从模式控制寄存器（TIM1_SMC） 299
19.5.4 TIM1 中断/DMA 使能寄存器（TIM1_DIER） 299
19.5.5 TIM1 状态寄存器（TIM1_SR） 302
19.5.6 TIM1 事件产生寄存器（TIM1_EVTG） 304
19.5.7 TIM1 计数值寄存器（TIM1_CNT） 305
19.5.8 TIM1 重复计数器寄存器（TIM1_RCR） 306
19.5.9 TIM1 比较使能寄存器（TIM1_CCEN） 306
19.5.10 TIM1 比较模式寄存器 1（TIM1_CCM1） 308
19.5.11 TIM1 比较模式寄存器 2（TIM1_CCM2） 310
19.5.12 TIM1 预分频寄存器（TIM1_PSC） 312
19.5.13 TIM1 自动重载值寄存器（TIM1_ARR） 312
19.5.14 TIM1 比较寄存器 x（TIM1_CCx） 313
19.5.15 TIM1 刹车和死区寄存器 1（TIM1_BDT1） 313
19.5.16 TIM1 刹车和死区寄存器 2（TIM1_BDT2） 315
19.5.17 TIM1 配置寄存器（TIM1_CFG） 315
19.5.18 TIM1 轮换功能寄存器 1（TIM1_AF1） 316
19.5.19 TIM1 输出选择寄存器（TIM1_OSEL） 318
19.5.20 TIM1 写使能寄存器（TIM1_LOCK） 319
20 通用定时器（TIM3/4） 321
20.1 简介 321

20.2 TIM3/4 主要特性 ..... 321
20.3 TIM3/4 功能描述 ..... 322
20.3.1 TIM3/4 框图 ..... 322
20.3.2 TIM3/4 引脚 ..... 323
20.3.3 PSC 预分频器 ..... 323
20.3.4 计数器 ..... 324
20.3.5 外部触发输入 ..... 332
20.3.6 计数时钟选择 ..... 332
20.3.7 捕获/比较通道 ..... 336
20.3.8 输入捕获模式 ..... 338
20.3.9 PWM 输入模式 ..... 339
20.3.10 比较输出模式 ..... 340
20.3.11 强制输出模式 ..... 341
20.3.12 PWM 模式 ..... 341
20.3.13 清除 OCxREF 信号 ..... 345
20.3.14 单脉冲模式 ..... 346
20.3.15 TIMx 输入异或功能 ..... 348
20.3.16 编码器接口模式 ..... 349
20.3.17 定时器同步从模式 ..... 352
20.3.18 定时器同步主模式 ..... 356
20.3.19 DMA Burst 传输模式 ..... 359
20.3.20 调试模式 ..... 360
20.4 TIM3/4 中断 ..... 360
20.5 TIM3/4 寄存器 ..... 361
20.5.1 TIM 控制寄存器 1（TIMx_CR1） ..... 361
20.5.2 TIM 控制寄存器 2（TIMx_CR2） ..... 363
20.5.3 TIM 从模式控制寄存器（TIMx_SMC） ..... 364
20.5.4 TIM 中断/DMA 使能寄存器（TIMx_DIER） ..... 366
20.5.5 TIM 状态寄存器（TIMx_SR） ..... 368
20.5.6 TIM 事件产生寄存器（TIMx_EVTG） ..... 370
20.5.7 TIM 计数值寄存器（TIMx_CNT） ..... 371
20.5.8 TIM 捕获/比较使能寄存器（TIMx_CCEN） ..... 371
20.5.9 TIM 捕获/比较模式寄存器 1（TIMx_CCM1）--（输出比较模式） ..... 373
20.5.10 TIM 捕获/比较模式寄存器 1（TIMx_CCM1）--（输入捕获模式） ..... 375
20.5.11 TIM 预分频值寄存器（TIMx_PSC） ..... 376
20.5.12 TIM 自动重载值寄存器（TIMx_ARR） ..... 377

20.5.13 TIM 捕获/比较寄存器 x（TIMx_CCx） ..... 377
20.5.14 TIM 配置寄存器（TIMx_CFG） ..... 378
20.5.15 TIM3 轮换功能寄存器 1（TIM3_AF1） ..... 378
20.5.16 TIM4 轮换功能寄存器 1（TIM4_AF1） ..... 379
20.5.17 TIM3 定时器输入选择寄存器（TIM3_TISEL） ..... 380
20.5.18 TIM4 定时器输入选择寄存器（TIM4_TISEL） ..... 380
21 霍尔定时器（HALLTIM） ..... 382
21.1 简介 ..... 382
21.2 HALLTIM 主要特性 ..... 382
21.3 HALLTIM 功能描述 ..... 383
21.3.1 HALLTIM 框图 ..... 383
21.3.2 HALLTIM 引脚 ..... 383
21.3.3 PSC 预分频器 ..... 383
21.3.4 计数器 ..... 383
21.3.5 滤波器 ..... 384
21.3.6 捕获 ..... 384
21.3.7 DMA 访问 ..... 385
21.3.8 调试模式 ..... 385
21.4 HALLTIM 中断 ..... 385
21.5 HALLTIM 寄存器 ..... 386
21.5.1 HALLTIM 控制寄存器（HALLTIM_CR） ..... 386
21.5.2 HALLTIM 中断/DMA 使能寄存器（HALLTIM_DIER） ..... 387
21.5.3 HALLTIM 状态寄存器（HALLTIM_SR） ..... 388
21.5.4 HALLTIM 宽度计数值寄存器（HALLTIM_WIDTH） ..... 389
21.5.5 HALLTIM 自动重载值寄存器（HALLTIM_ARR） ..... 389
21.5.6 HALLTIM 计数值寄存器（HALLTIM_CNT） ..... 390
22 低功耗定时器（LPTIM） ..... 391
22.1 简介 ..... 391
22.2 LPTIM 主要特性 ..... 391
22.3 LPTIMx 功能描述 ..... 391
22.3.1 LPTIMx 框图 ..... 391
22.3.2 LPTIM 引脚 ..... 392
22.3.3 时钟源 ..... 392
22.3.4 计数器模式 ..... 393
22.3.5 定时器使能 ..... 394
22.3.6 调试模式 ..... 394

22.4 LPTIM 低功耗模式 ..... 394
22.5 LPTIM 中断 ..... 394
22.6 LPTIM 寄存器 ..... 395
22.6.1 LPTIM 中断和状态寄存器（LPTIMx_ISR） ..... 395
22.6.2 LPTIM 中断清除寄存器（LPTIMx_ICR） ..... 395
22.6.3 LPTIM 中断使能寄存器（LPTIMx_IER） ..... 396
22.6.4 LPTIM 配置寄存器（LPTIMx_CFG） ..... 397
22.6.5 LPTIM 控制寄存器（LPTIMx_CR） ..... 398
22.6.6 LPTIM 自动重载寄存器（LPTIMx_ARR） ..... 398
22.6.7 LPTIM 计数器寄存器（LPTIMx_CNT） ..... 399
23 红外控制模块（IRTIM） ..... 400
23.1 IRTIM 寄存器 ..... 401
23.1.1 IRTIM 控制寄存器（IRTIM_CR） ..... 401
24 独立看门狗（IWDG） ..... 402
24.1 简介 ..... 402
24.2 IWDG 主要特性 ..... 402
24.3 IWDG 功能描述 ..... 402
24.3.1 IWDG 框图 ..... 402
24.3.2 寄存器访问保护 ..... 403
24.3.3 溢出时间设置 ..... 403
24.3.4 调试模式 ..... 403
24.4 IWDG 低功耗模式 ..... 404
24.5 IWDG 寄存器 ..... 405
24.5.1 IWDG 控制寄存器（IWDG_CR） ..... 405
24.5.2 IWDG 配置寄存器（IWDG_CFG） ..... 405
24.5.3 IWDG 计数值寄存器（IWDG_CNT） ..... 406
25 SysTick 定时器（SysTick） ..... 408
25.1 简介 ..... 408
25.2 SysTick 主要特性 ..... 408
25.3 SysTick 功能描述 ..... 408
25.4 SysTick 寄存器 ..... 409
25.4.1 SysTick 控制和状态寄存器（SysTick_CTRL） ..... 409
25.4.2 SysTick 重载值寄存器（SysTick_LOAD） ..... 410
25.4.3 SysTick 当前计数值寄存器（SysTick_VAL） ..... 410
25.4.4 SysTick 校准值寄存器（SysTick_CALIB） ..... 410
26 实时时钟（RTC） ..... 412

26.1 简介 ..... 412
26.2 RTC 主要特性 ..... 412
26.3 RTC 功能描述 ..... 413
26.3.1 RTC 框图 ..... 413
26.3.2 RTC 引脚 ..... 413
26.3.3 RTC 寄存器写保护 ..... 414
26.3.4 RTC 时钟和分频 ..... 414
26.3.5 RTC 日历的读取 ..... 415
26.3.6 RTC 日历的配置 ..... 415
26.3.7 RTC 闹钟 ..... 416
26.3.8 RTC 周期定时 ..... 416
26.3.9 RTC 时钟数字平滑校准 ..... 417
26.3.10 调试模式 ..... 419
26.4 RTC 低功耗模式 ..... 419
26.5 RTC 中断 ..... 419
26.6 RTC 寄存器 ..... 420
26.6.1 RTC 时间寄存器（RTC_TR） ..... 420
26.6.2 RTC 日期寄存器（RTC_DR） ..... 421
26.6.3 RTC 亚秒寄存器（RTC_SSR） ..... 422
26.6.4 RTC 控制与状态寄存器（RTC_CSR） ..... 422
26.6.5 RTC 控制寄存器（RTC_CR） ..... 423
26.6.6 RTC 周期定时寄存器 1（RTC_WUTIM1） ..... 425
26.6.7 RTC 周期定时寄存器 2（RTC_WUTIM2） ..... 426
26.6.8 RTC 写保护寄存器（RTC_WPR） ..... 427
26.6.9 RTC 时钟数字平滑校准寄存器（RTC_CALR） ..... 428
26.6.10 RTC 闹钟寄存器（RTC_ALARMR） ..... 428
26.6.11 RTC 状态寄存器（RTC_SR） ..... 429
26.6.12 RTC 中断状态寄存器（RTC_ISR） ..... 430
26.6.13 RTC 状态清除寄存器（RTC_CLR） ..... 431
27 I2C 接口（I2C） ..... 433
27.1 简介 ..... 433
27.2 I2C 主要特性 ..... 433
27.3 I2C 功能描述 ..... 433
27.3.1 I2C 框图 ..... 434
27.3.2 I2C 引脚 ..... 434
27.3.3 I2C 时钟要求 ..... 434

27.3.4 I2C 初始化 ..... 434
27.3.5 软件复位 ..... 437
27.3.6 数据传输 ..... 437
27.3.7 通信流程 ..... 439
27.3.8 DMA 传输 ..... 444
27.3.9 错误条件 ..... 444
27.4 I2C 中断 ..... 445
27.5 I2C 寄存器 ..... 446
27.5.1 I2C 控制寄存器 1（I2Cx_CR1） ..... 446
27.5.2 I2C 控制寄存器 2（I2Cx_CR2） ..... 448
27.5.3 I2C 地址寄存器 1（I2Cx_ADDR1） ..... 449
27.5.4 I2C 地址寄存器 2（I2Cx_ADDR2） ..... 449
27.5.5 I2C 中断和状态寄存器（I2Cx_ISR） ..... 450
27.5.6 I2C 中断清除寄存器（I2Cx_ICR） ..... 452
27.5.7 I2C 接收数据寄存器（I2Cx_RDR） ..... 453
27.5.8 I2C 发送数据寄存器（I2Cx_TDR） ..... 454
28 通用异步收发器（UART） ..... 455
28.1 简介 ..... 455
28.2 UART 主要特性 ..... 455
28.3 UART 功能描述 ..... 456
28.3.1 UART 框图 ..... 456
28.3.2 UART 引脚 ..... 456
28.3.3 帧格式 ..... 456
28.3.4 校验控制 ..... 457
28.3.5 发送器 ..... 458
28.3.6 接收器 ..... 459
28.3.7 波特率生成 ..... 463
28.3.8 接收容差 ..... 464
28.3.9 单线半双工通信 ..... 465
28.3.10 使用 DMA 通信 ..... 465
28.4 UART 中断 ..... 467
28.5 UART 寄存器 ..... 468
28.5.1 UART 控制寄存器 1（UARTx_CR1） ..... 468
28.5.2 UART 控制寄存器 2（UARTx_CR2） ..... 470
28.5.3 UART 控制寄存器 3（UARTx_CR3） ..... 471
28.5.4 UART 波特率分频寄存器（UARTx_BRR） ..... 473

28.5.5 UART 中断和状态寄存器（UARTx_ISR） 473
28.5.6 UART 中断标志清零寄存器（UARTx_ICR） 475
28.5.7 UART 接收数据寄存器（UARTx_RDR） 476
28.5.8 UART 发送数据寄存器（UARTx_TDR） 476
29 低功耗通用异步收发器（LPUART） 478
29.1 简介 478
29.2 主要特性 478
29.3 LPUART 功能描述 479
29.3.1 LPUART 框图 479
29.3.2 LPUART 引脚 479
29.3.3 帧格式 480
29.3.4 校验控制 480
29.3.5 发送器 481
29.3.6 接收器 483
29.3.7 波特率生成 485
29.3.8 接收容差 486
29.3.9 单线半双工通信 486
29.3.10 使用 DMA 通信 487
29.3.11 RS232 硬件流控 488
29.4 LPUART 低功耗模式 490
29.4.1 LPUART 对低功耗模式的支持 490
29.4.2 LPUART 内核时钟（LPUARTx_KCLK）停止时从 Stop 模式唤醒 491
29.5 LPUART 中断 491
29.6 LPUART 寄存器 493
29.6.1 LPUART 控制寄存器 1（LPUARTx_CR1） 493
29.6.2 LPUART 控制寄存器 2（LPUARTx_CR2） 495
29.6.3 LPUART 控制寄存器 3（LPUARTx_CR3） 496
29.6.4 LPUART 波特率分频寄存器（LPUARTx_BRR） 498
29.6.5 LPUART 中断和状态寄存器（LPUARTx_ISR） 499
29.6.6 LPUART 中断标志清零寄存器（LPUARTx_ICR） 501
29.6.7 LPUART 接收数据寄存器（LPUARTx_RDR） 503
29.6.8 LPUART 发送数据寄存器（LPUARTx_TDR） 503
29.6.9 LPUART 预分频器寄存器（LPUARTx_PRESC） 503
30 串行外设接口（SPI） 505
30.1 简介 505
30.2 SPI 主要特性 505

30.3 SPI 功能说明 ..... 506
30.3.1 SPI 框图 ..... 506
30.3.2 SPI 引脚 ..... 506
30.3.3 单主机通信 ..... 506
30.3.4 多主机通信 ..... 507
30.3.5 从机通信 ..... 508
30.3.6 NSS 管理 ..... 509
30.3.7 通信模式 ..... 510
30.3.8 状态标志 ..... 511
30.3.9 错误标志 ..... 511
30.3.10 初始化 SPI ..... 512
30.3.11 数据发送和接收 ..... 513
30.3.12 禁止 SPI ..... 515
30.4 SPI 中断 ..... 515
30.5 SPI 寄存器 ..... 516
30.5.1 SPI 控制寄存器 1（SPIx_CR1） ..... 516
30.5.2 SPI 控制寄存器 2（SPIx_CR2） ..... 518
30.5.3 SPI 中断和状态寄存器（SPIx_ISR） ..... 519
30.5.4 SPI 中断标志清零寄存器（SPIx_ICR） ..... 520
30.5.5 SPI 数据寄存器（SPIx_DR） ..... 521
31 调试接口（DBG） ..... 522
31.1 简介 ..... 522
31.2 DBG 功能描述 ..... 522
31.3 调试组件（DBG） ..... 522
31.3.1 对低功耗模式的调试支持 ..... 522
31.3.2 对定时器和看门狗的调试支持 ..... 523
31.4 DBG 寄存器 ..... 524
31.4.1 DBG 控制寄存器（DBG_CR） ..... 524
31.4.2 APB1 冻结寄存器（DBG_APB1_FZ） ..... 524
31.4.3 APB2 冻结寄存器（DBG_APB2_FZ） ..... 526
32 器件电子签名 ..... 527
32.1 产品唯一身份识别（UID）寄存器（96 位） ..... 527
32.2 Flash 空间参数寄存器 ..... 528
32.3 SRAM 空间参数寄存器 ..... 528
33 版本历史 ..... 529
34 联系方式 ..... 530

# 表目录

表 3-1 存储器地址范围 33
表 3-2 外设寄存器地址范围 34
表 3-3 启动配置 36
表 4-1 Flash 存储器组成 37
表 4-2 HCLK 时钟频率与 Flash 读取访问等待周期的对应关系 38
表 4-3 选项字节组成 42
表 4-4 产品信息 44
表 4-5 出厂校准参数组成 45
表 4-6 RDP 保护等级配置 48
表 4-7 WRP 保护区域配置 49
表 4-8 中断请求 50
表 4-9 Flash 基地址 51
表 5-1 低功耗模式概述 61
表 5-2 低功耗模式下可用外设 (1) 62
表 5-3 Sleep 说明 63
表 5-4 Stop 说明 65
表 5-5 PMU 基地址 66
表 6-1 时钟关系表 73
表 6-2 LXTAL 时钟源 75
表 6-3 LXTAL 时钟配置 76
表 6-4 外设异步时钟来源 77
表 6-5 RCC 基地址 80
表 7-1 GPIO 端口配置表 (1) 103
表 7-2 端口复用功能映射 110
表 7-3 外部振荡器引脚 114
表 7-4 GPIO 基地址 115
表 8-1 DMA 通道请求信号 121
表 8-2 DMA 传输类型概述 123
表 8-3 DMA 地址对齐 124
表 8-4 DMA 寻址范围 124
表 8-5 DMA 地址递增 125
表 8-6 中断请求 127
表 8-7 DMA 基地址 128
表 9-1 中断向量表 (1) 136

表 10-1 EXTI 内部信号 ..... 140
表 10-2 EXTI 通道及信号对应表 ..... 141
表 10-3 EXTI 屏蔽功能 ..... 141
表 10-4 EXTI 基地址 ..... 144
表 11-1 CRC 基地址 ..... 154
表 12-1 HWDIV 基地址 ..... 157
表 13-1 SQRT 基地址 ..... 161
表 14-1 ADC 输入引脚 ..... 165
表 14-2 ADC 内部信号 ..... 165
表 14-3 ADC 自动注入序列配置 ..... 168
表 14-4 温度传感器校准参数 ..... 186
表 14-5 内部带隙基准电压校准参数 ..... 187
表 14-6 中断请求 ..... 190
表 14-7 ADC 基地址 ..... 191
表 16-1 OPAMP1 正相输入信号 ..... 217
表 16-2 OPAMP1 反相输入信号 ..... 217
表 16-3 OPAMP2 正相输入信号 ..... 217
表 16-4 OPAMP2 反相输入信号 ..... 217
表 16-5 OPAMP 基地址 ..... 225
表 17-1 COMP1 正相输入信号 ..... 236
表 17-2 COMP1 反相输入信号 ..... 237
表 17-3 COMP2 正相输入信号 ..... 237
表 17-4 COMP2 反相输入信号 ..... 237
表 17-5 低功耗模式对 COMP 的影响 ..... 241
表 17-6 中断请求 ..... 241
表 17-7 COMP 基地址 ..... 242
表 18-1 $V_{LCD}$ 输出电压配置 ..... 250
表 18-2 帧速率典型值计算示例 ..... 252
表 18-3 LCD 控制器显示映射 ..... 258
表 18-4 中断请求 ..... 259
表 18-5 低功耗模式下 LCD 控制器工作状态 ..... 259
表 18-6 LCD 基地址 ..... 260
表 19-1 TIM1 引脚 ..... 267
表 19-2 中断请求 ..... 293
表 19-3 TIM1 基地址 ..... 295
表 20-1 TIM3 引脚 ..... 323

表 20-2 TIM4 引脚 ..... 323
表 20-3 正交编码计数方向与编码器信号的关系 ..... 350
表 20-4 脉冲电平编码模式计数方向与编码器信号的关系 ..... 351
表 20-5 TIMx 内部级联 ..... 352
表 20-6 中断请求 ..... 360
表 20-7 TIMx 基地址 ..... 361
表 21-1 HALLTIM 引脚 ..... 383
表 21-2 中断请求 ..... 385
表 21-3 HALLTIM 基地址 ..... 386
表 22-1 LPTIMx 引脚 ..... 392
表 22-2 低功耗模式对 LPTIMx 的影响 ..... 394
表 22-3 中断请求 ..... 394
表 22-4 LPTIMx 基地址 ..... 395
表 23-1 IRTIM 基地址 ..... 401
表 24-1 IWDG 溢出周期表 ..... 403
表 24-2 低功耗模式对 IWDG 的影响 ..... 404
表 24-3 IWDG 基地址 ..... 405
表 26-1 RTC 输出引脚 ..... 413
表 26-2 定时闹钟举例 ..... 416
表 26-3 中断请求 ..... 419
表 26-4 RTC 基地址 ..... 420
表 27-1 I2Cx 引脚 ..... 434
表 27-2 中断请求 ..... 445
表 27-3 I2Cx 基地址 ..... 446
表 28-1 UART 校验控制 ..... 457
表 28-2 中断请求 ..... 467
表 28-3 UARTx 基地址 ..... 468
表 29-1 LPUART 校验控制 ..... 480
表 29-2 LPUARTx_KCLK_PRES=32.768KHz 时的波特率误差 ..... 485
表 29-3 低功耗模式说明 ..... 491
表 29-4 中断请求 (1) ..... 492
表 29-5 LPUARTx 基地址 ..... 493
表 30-1 SPI 引脚 ..... 506
表 30-2 NSS 配置 ..... 509
表 30-3 通信模式配置 ..... 510
表 30-4 中断请求 ..... 515

表 30-5 SPIx 基地址 ..... 516
表 31-1 DBG 基地址 ..... 524
表 32-1 UID 规则 ..... 527
表 33-1 版本更改履历 ..... 529

# 图目录

图 3-1 系统架构示意图 31
图 3-2 存储器映射 33
图 4-1 RDP 保护等级变更 48
图 5-1 电源结构框图 57
图 5-2 POR/PDR 阈值 58
图 5-3 PVD 阈值 59
图 5-4 PVD 滤波 60
图 5-5 低功耗模式转换框图 61
图 6-1 复位电路框图 71
图 6-2 时钟树 73
图 6-3 TIM4 在捕获模式下的频率测量 79
图 7-1 I/O 基本结构 103
图 7-2 I/O 引脚输入配置（浮空/上拉/下拉） 107
图 7-3 I/O 引脚输出配置 108
图 7-4 I/O 引脚复用功能配置 109
图 7-5 I/O 引脚高阻态模拟配置 113
图 8-1 DMA 框图 121
图 10-1 EXTI 框图 140
图 10-2 可配置通道&直接通道的事件输出框图 142
图 10-3 可配置通道的中断输出框图 142
图 10-4 EXTI 多路复用器 143
图 11-1 CRC 框图 152
图 11-2 反转框图 153
图 11-3 CRC 计算单元框图 153
图 14-1 ADC 结构框图 164
图 14-2 ADC 输入通道 167
图 14-3 单次扫描模式下的规则序列和注入序列转换示意图 169
图 14-4 循环扫描模式下的规则序列和注入序列转换示意图 171
图 14-5 循环间断模式下的注入序列转换示意图 174
图 14-6 停止正在进行的转换 178
图 14-7 单次扫描转换模式，规则软件触发，注入外部事件触发 179
图 14-8 循环扫描转换模式，软件触发，使能自动注入 180
图 14-9 循环间断转换模式，规则硬件触发，注入硬件触发，仅规则间断 181
图 14-10 循环间断转换模式，规则软件触发，注入硬件触发，仅注入间断 182

图 14-11 数据溢出处理示例 (1) ..... 183
图 14-12 数据对齐 ..... 184
图 14-13 模拟看门狗阈值更新 (1) ..... 185
图 14-14 内部输入通道结构 ..... 186
图 14-15 ADC 控制 OPAMP2 输入切换结构图 ..... 189
图 16-1 OPAMP 框图 ..... 216
图 16-2 独立运放模式 ..... 218
图 16-3 电压跟随模式 ..... 219
图 16-4 伪差分 PGA 模式 ..... 220
图 16-5 同相 PGA 模式 ..... 221
图 16-6 反相 PGA 模式 ..... 222
图 17-1 COMP 框图 ..... 236
图 17-2 窗口比较功能示意图 ..... 238
图 17-3 开窗功能控制示意图 ..... 239
图 17-4 COMP 输出滤波 ..... 240
图 17-5 COMP 迟滞 ..... 240
图 18-1 LCD 控制器框图 ..... 249
图 18-2 片内 buffer 驱动模式示意图 ..... 250
图 18-3 死区时间 ..... 252
图 18-4 静态占空比驱动波形示例 ..... 253
图 18-5 1/2duty，1/2bias 驱动波形示例 ..... 254
图 18-6 1/3duty，1/3bias 驱动波形示例 ..... 255
图 18-7 1/4duty，1/3bias 驱动波形示例 ..... 256
图 18-8 1/4duty，1/4bias 驱动波形示例 ..... 257
图 19-1 TIM1 框图 ..... 267
图 19-2 预分频器分频系数由 1 变为 4 时的计数器时序图 ..... 268
图 19-3 递增计数模式下配置预分频器为 1 分频上溢事件时序图 ..... 270
图 19-4 递增计数模式下配置预分频器为 4 分频上溢事件时序图 ..... 270
图 19-5 递增计数模式下计数器时序图，ARPE=0 时更新事件 ..... 271
图 19-6 递增计数模式下计数器时序图，ARPE=1 时更新事件 ..... 271
图 19-7 递减计数模式下配置预分频器为 1 分频下溢事件时序图 ..... 272
图 19-8 递减计数模式下配置预分频器为 4 分频下溢事件时序图 ..... 273
图 19-9 递减计数模式下计数器时序图，ARPE=0 时更新事件 ..... 273
图 19-10 中心对齐模式下配置预分频器为 1 分频上/下溢事件时序图 ..... 275
图 19-11 中心对齐模式下配置预分频器为 2 分频下溢事件时序图 ..... 275
图 19-12 中心对齐模式下，产生下溢事件，ARPE=1 时的时序图 ..... 276

图 19-13 中心对齐模式下，产生上溢事件，ARPE=1 时的时序图 ..... 276
图 19-14 中心对齐模式下重复计数加载时序图 ..... 277
图 19-15 不同模式下设置不同 TIM1_RCR 寄存器值时的更新频率示例 ..... 278
图 19-16 比较通道 ..... 279
图 19-17 比较通道的输出连接图（通道 1） ..... 279
图 19-18 比较输出模式，翻转 OC1（OCxPE 位为 0） ..... 280
图 19-19 递增计数下的边沿对齐模式 PWM 波形（TIM1_ARR 寄存器为 8） ..... 282
图 19-20 递减计数下的边沿对齐模式 PWM 波形（TIM1_ARR 寄存器为 8） ..... 283
图 19-21 中心对齐模式 PWM 波形（TIM1_ARR 寄存器为 8） ..... 284
图 19-22 清除 TIM1 的 OCxREF ..... 285
图 19-23 单脉冲输出波形 ..... 285
图 19-24 单脉冲模式示例 ..... 286
图 19-25 互补模式下比较输出示意图 ..... 287
图 19-26 带死区插入的互补输出 ..... 288
图 19-27 右死区时间大于负脉冲宽度的死区波形 ..... 288
图 19-28 左死区时间大于正脉冲宽度的死区波形 ..... 288
图 19-29 刹车电路概述 ..... 289
图 19-30 输出重映射示意图 ..... 290
图 19-31 霍尔传感器接口的示例 ..... 292
图 20-1 TIM3 框图 ..... 322
图 20-2 TIM4 框图 ..... 322
图 20-3 预分频器分频系数由 1 变为 4 时的计数器时序图 ..... 323
图 20-4 递增计数模式下配置预分频器为 1 分频上溢事件时序图 ..... 325
图 20-5 递增计数模式下配置预分频器为 4 分频上溢事件时序图 ..... 325
图 20-6 递增计数模式下计数器时序图，ARPE=0 时更新事件 ..... 326
图 20-7 递增计数模式下计数器时序图，ARPE=1 时更新事件 ..... 326
图 20-8 递减计数模式下配置预分频器为 1 分频下溢事件时序图 ..... 327
图 20-9 递减计数模式下配置预分频器为 4 分频下溢事件时序图 ..... 328
图 20-10 递减计数模式下计数器时序图，ARPE=0 时更新事件 ..... 328
图 20-11 中心对齐模式下配置预分频器为 1 分频上/下溢事件时序图 ..... 330
图 20-12 中心对齐模式下配置预分频器为 2 分频下溢事件时序图 ..... 330
图 20-13 中心对齐模式下，产生下溢事件，ARPE=1 时的时序图 ..... 331
图 20-14 中心对齐模式下，产生上溢事件，ARPE=1 时的时序图 ..... 331
图 20-15 TIM3 ETR 输入源选择 ..... 332
图 20-16 TIM4 ETR 输入源选择 ..... 332
图 20-17 外部触发输入模块 ..... 332

图 20-18 选择 TI2 作为时钟模式 1 触发输入的连接示例（上升沿计数） ..... 334
图 20-19 选择时钟模式 1 时的时序图 ..... 335
图 20-20 时钟模式 2 触发输入模块 ..... 335
图 20-21 选择时钟模式 2 下的时序图 ..... 336
图 20-22 捕获/比较通道 ..... 336
图 20-23 捕获/比较通道的输入连接图（通道 1） ..... 337
图 20-24 捕获/比较通道的输出连接图（通道 1） ..... 338
图 20-25 PWM 输入模式的时序 ..... 340
图 20-26 比较输出模式，翻转 OC1（OCxPE 位为 0） ..... 341
图 20-27 递增计数下的边沿对齐模式 PWM 波形（TIMx_ARR 寄存器为 8） ..... 343
图 20-28 递减计数下的边沿对齐模式 PWM 波形（TIMx_ARR 寄存器为 8） ..... 344
图 20-29 中心对齐模式 PWM 波形（TIMx_ARR 寄存器为 8） ..... 345
图 20-30 清除 TIMx 的 OCxREF ..... 346
图 20-31 单脉冲输出波形 ..... 347
图 20-32 单脉冲模式示例 ..... 347
图 20-33 测量 3 个信号边沿之间的时间间隔 ..... 349
图 20-34 编码器接口模式下的计数器工作示例 ..... 351
图 20-35 脉冲电平编码模式示例 ..... 352
图 20-36 复位模式下的控制时序 ..... 353
图 20-37 门控模式下的控制时序 ..... 354
图 20-38 触发模式下的控制时序 ..... 355
图 20-39 外部时钟模式 2+触发模式下的控制时序 ..... 356
图 20-40 主从定时器连接示例图 ..... 356
图 20-41 使用 TIMx 的 OC1REF 信号对 TIMy 实施门控控制 ..... 357
图 20-42 使用 TIMx 的门控信号对 TIMy 实施门控控制 ..... 358
图 20-43 使用 TIMx 的 TI1 输入触发 TIMx 和 TIMy ..... 359
图 21-1 HALLTIM 框图 ..... 383
图 21-2 第一级滤波示意图 ..... 384
图 22-1 LPTIMx 框图（x = 1） ..... 391
图 22-2 单次计数模式使能说明 ..... 394
图 23-1 IRTIM 内部硬件连接 ..... 400
图 24-1 IWDG 框图 ..... 402
图 26-1 RTC 框图 ..... 413
图 26-2 校准参数计算流程 ..... 418
图 27-1 I2Cx 框图（x = 1） ..... 434
图 27-2 I2C 初始化流程 ..... 435

图 27-3 单字节控制数据接收 438
图 27-4 多字节控制数据接收 438
图 27-5 单字节控制数据发送 439
图 27-6 多字节控制数据发送 439
图 27-7 使能时钟低电平延长单字节控制发送数据 440
图 27-8 使能时钟低电平延长多字节控制发送数据 441
图 27-9 禁止时钟低电平延长发送数据 441
图 27-10 使能时钟低电平延长单字节控制接收数据 442
图 27-11 使能时钟低电平延长多字节控制接收数据 443
图 27-12 禁止时钟低电平延长接收数据 443
图 28-1 UARTx 框图（x = 1，2） 456
图 28-2 帧格式示意图 457
图 28-3 可编程停止位示例 458
图 28-4 发送过程 TC 和 TXE 的动作 459
图 28-5 起始位检测 460
图 28-6 数据采样（OBS=0） 463
图 29-1 LPUARTx 框图（x = 1） 479
图 29-2 帧格式示意图 480
图 29-3 可编程停止位示例 482
图 29-4 发送过程 TC 和 TXE 的动作 483
图 29-5 两个 LPUART 间流控信号连接示例 489
图 29-6 RTS 动作示例 489
图 29-7 CTS 动作示例 490
图 29-8 检测到有效唤醒事件 491
图 30-1 SPIx 框图（x = 1，2） 506
图 30-2 单主单从应用 507
图 30-3 单主多从应用 507
图 30-4 多主多从应用 508
图 30-5 通信模式时序 510
图 30-6 主从模式 BUSY 状态 511
图 30-7 主模式通信时序 514
图 30-8 从模式通信时序 515

# 1 文档约定

## 1.1 寄存器列表符号定义

| | |
|---|---|
| 读/写（rw） | 软件可以读写这些位 |
| 只读（r） | 软件只能读取这些位 |
| 只写（w） | 软件只能写入该位。读取该位时将返回复位值 |
| 读取/清零（rc_w1） | 软件可以读取该位，也可以通过写入“1”将该位清零。写入“0”对该位的值无影响 |
| 读取/清零（rc_w0） | 软件可以读取该位，也可以通过写入“0”将该位清零。写入“1”对该位的值无影响 |
| 读取/读取清零（rc_r） | 软件可以读取该位。读取该位时，将自动清零。写入“0”对该位的值无影响 |
| 读取/读取置位（rs_r） | 软件可以读取该位。读取该位时，将自动置“1”。写入该位对其值无影响。 |
| 读取/置位（rs） | 软件可以读取该位，也可将其置“1”。写入“0”对该位的值无影响 |
| 读/仅可写入一次（rwo） | 软件仅可写入一次该位，但可随时读取该位。只能通过复位将该位返回到复位值。 |
| 只读，写触发（rt_w1） | 软件可以读取该位。写入“1”时，将触发事件，但不会影响该位的值。 |
| 保留（Res.） | 保留位，读为“0”，写入无效 |

## 1.2 术语及定义

本节简要介绍本文档中所用首字母缩略词和缩写词的定义：

| | |
|---|---|
| 页 | 512 字节大小的 Flash 空间为一页 |
| 字 | 32 位数据 |
| 半字 | 16 位数据 |
| 字节 | 8 位数据 |
| 选项字节（Option Byte） | 存储于 Flash 中的产品配置位 |

| | |
|---|---|
| AHB | 高级高性能总线 |
| APB | 高级外设总线 |
| Run | 运行模式 |
| Sleep | 睡眠模式 |
| Stop | 停止模式 |

# 2 简介

CIU32F042 系列安全 MCU 基于 ARM Cortex-M0+内核，最高频率可达 72MHz，具有多达 64K bytes Flash、8K bytes SRAM，支持 LQFP48、QFN32 等多种封装。内部集成 1.125Msps ADC、多档内部参考电压源 VREFBUF、3 个支持 PGA 模式的运算放大器、2 个快速或超低功耗比较器、LCD、3 通道 DMA、2 个最高速率可达 24Mbps 的 SPI、2 个 UART、LPUART、I2C 等多种丰富的外设资源；同时提供多种定时器，包括 8 通道或 4 通道互补高级定时器、具备多种编码器功能的通用定时器、32bit 通用定时器和 24bit 霍尔定时器等。

# 3 系统与存储器架构

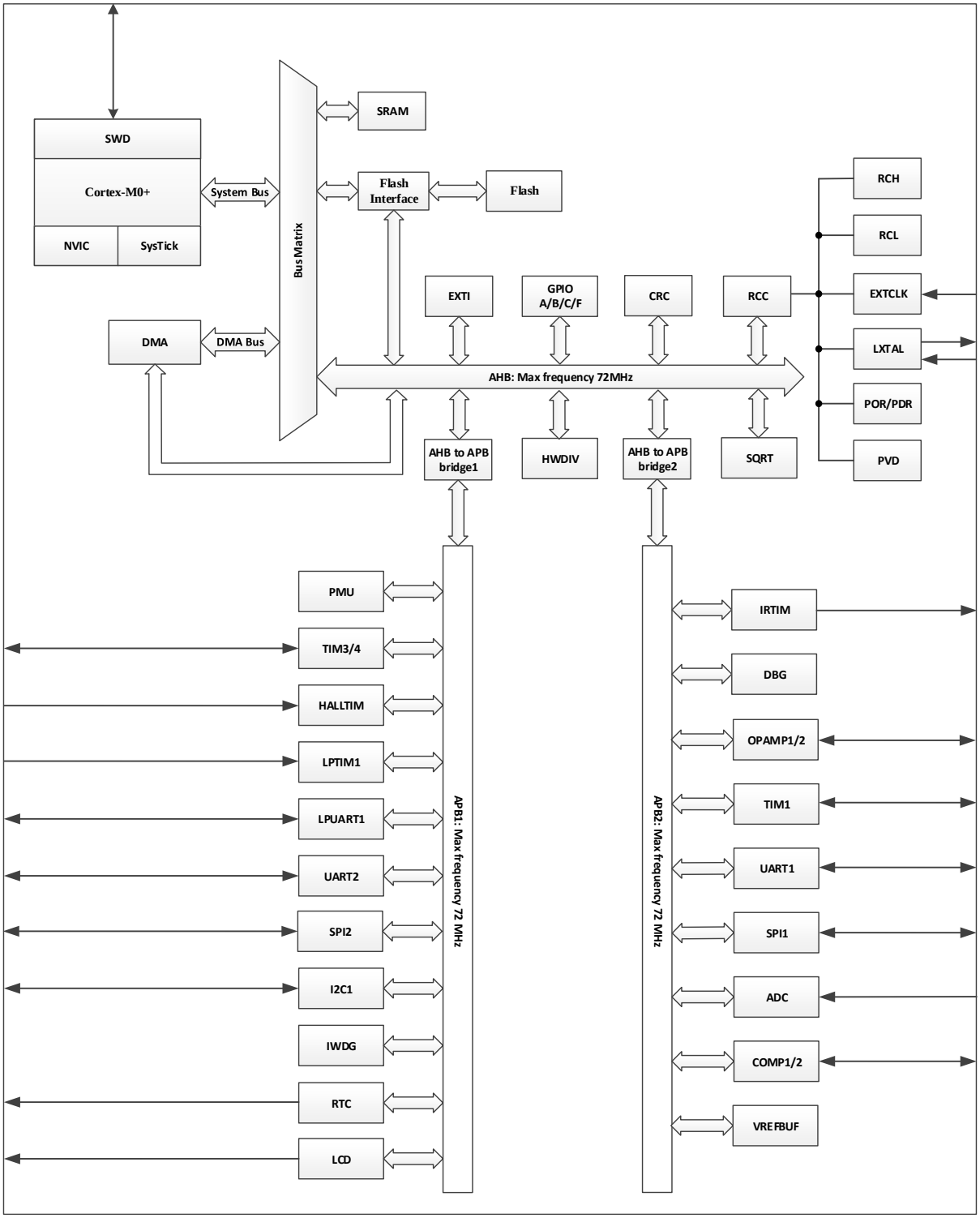
## 3.1 系统架构

主系统包括：

- 两个主机：
  - Cortex-M0+内核
  - DMA
- 三个从机：
  - SRAM
  - Flash
  - AHB：带有两条 AHB-to-APB 的总线桥，连接 AHB 和所有 APB 外设

所有外设通过多层 AHB 总线架构连接，具体如下图：

图 3-1 系统架构示意图




### 3.1.1 System Bus

该总线用于将 Cortex-M0+内核连接到 Bus Matrix，内核通过此总线进行取指、数据操作以及访问 AHB/APB 外设。

### 3.1.2 DMA Bus

该总线用于将 DMA 的 AHB 接口连接到 Bus Matrix，DMA 通过此总线访问 Flash、SRAM 以及 APB 外设。

### 3.1.3 Bus Matrix

总线矩阵包含两个主机（CPU 和 DMA）和三个从机（Flash、SRAM、带有 AHB-to-APB 总线桥的 AHB），并管理 Cortex-M0+内核的 System Bus 和 DMA Bus 的访问仲裁。

### 3.1.4 AHB to APB 总线桥

AHB to APB 总线桥用于连接 AHB 和 APB 总线，可以实现 AHB 和 APB 总线的同步；APB 的最高工作频率可达 72MHz。

## 3.2 存储器

### 3.2.1 SRAM

SRAM 总容量 8KB，支持按字节、半字、字访问。支持以最高系统时钟频率（72MHz）进行寻址，无需插入等待周期。

### 3.2.2 Flash

Flash 存储器区域划分如下：

- User flash 区：高达 64KB，包含应用程序和用户数据，由用户管理；
- System memory 区：用于存储 Bootloader；
- Option bytes 区：用于存储存储器保护配置的选项字节；
- Engineer 区：用于存储产品信息及出厂校准参数。

Flash 接口根据 AHB 协议实现指令访问和数据访问，可通过 Flash 寄存器控制 Flash 操作（编程/擦除）。

### 3.2.3 存储器映射

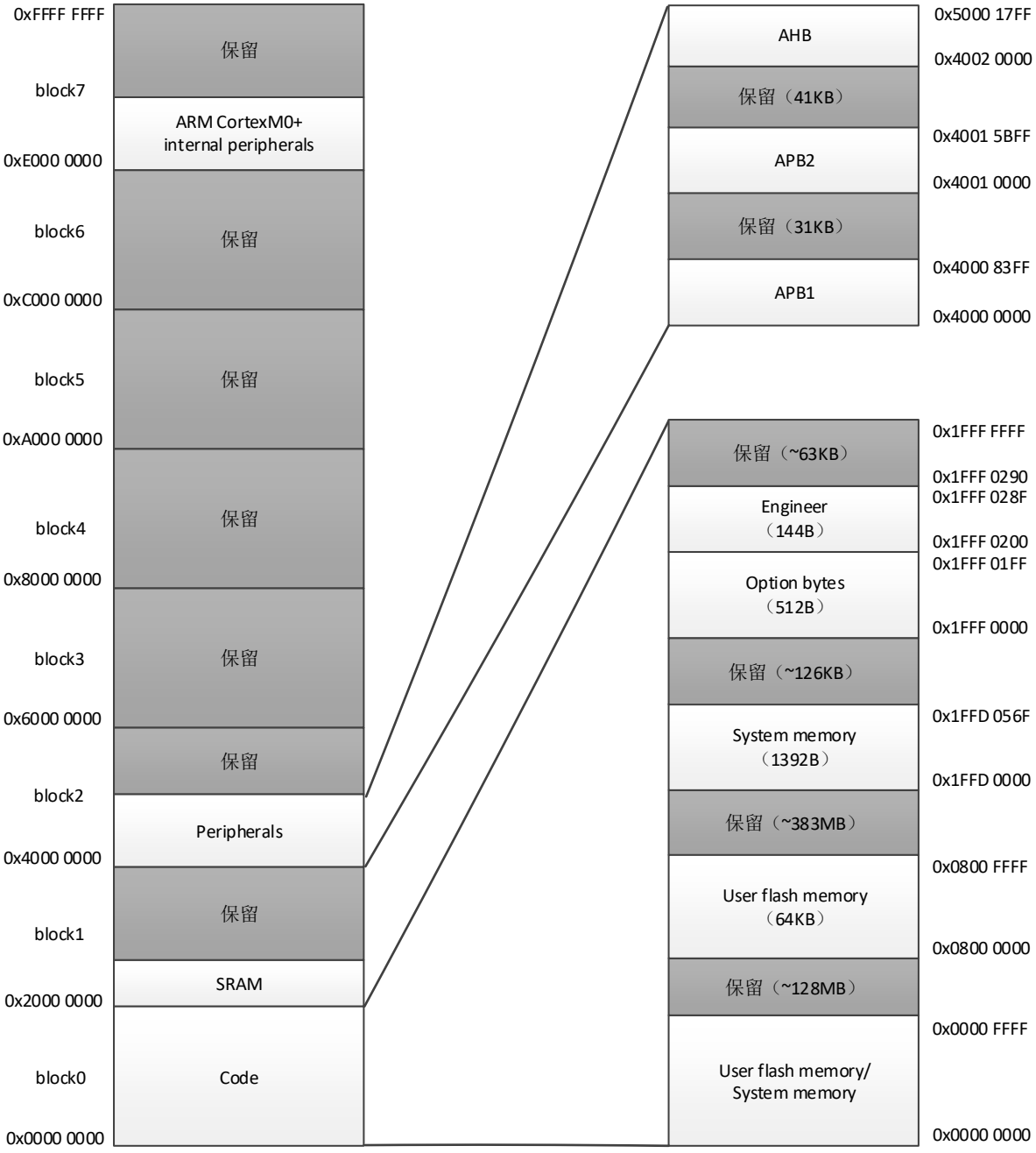
程序存储器、数据存储器、寄存器排列在一个地址连续的 4GB 地址空间内。

数据按小端格式保存在存储器中，地址最小的存储单元保存该字的最低位数据，

而地址最大的存储单元保存该字的最高位数据。

可寻址的存储空间分为 8 个块，每个块为 512MB。

图 3-2 存储器映射




未分配给片上存储器和外设的所有存储映射区域均视为“保留区”，对“保留区”的访问会产生总线错误，具体存储映射参见下表。

表 3-1 存储器地址范围

| 存储类型 | 地址范围 | 大小 | 存储内容 |
|---|---|---|---|
| SRAM | 0x2000 2000 - 0x3FFF FFFF | ~512MB | Reserved |

| 存储类型 | 地址范围 | 大小 | 存储内容 |
|---|---|---|---|
| | 0x2000 0000 - 0x2000 1FFF | 8KB | SRAM |
| FLASH | 0x1FFF 0290 - 0x1FFF FFFF | ~63KB | Reserved |
| | 0x1FFF 0200 - 0x1FFF 028F | 144B | Engineer |
| | 0x1FFF 0000 - 0x1FFF 01FF | 512B | Option bytes |
| | 0x1FFD 0570 - 0x1FFE FFFF | ~126KB | Reserved |
| | 0x1FFD 0000 - 0x1FFD 056F | 1392B | System memory |
| | 0x0801 0000 - 0x1FFC FFFF | ~383MB | Reserved |
| | 0x0800 0000 - 0x0800 FFFF | 64KB | User flash memory |

表 3-2 外设寄存器地址范围

| 总线 | 地址范围 | 大小 | 外设 |
|---|---|---|---|
| CPU | 0xE000 0000 - 0xE00F FFFF | 1MB | Cortex-M0+ internal peripherals |
| AHB | 0x5000 1800 - 0x5FFF FFFF | ~256MB | Reserved |
| | 0x5000 1400 - 0x5000 17FF | 1KB | GPIOF |
| | 0x5000 0C00 - 0x5000 13FF | 2KB | Reserved |
| | 0x5000 0800 - 0x5000 0BFF | 1KB | GPIOC |
| | 0x5000 0400 - 0x5000 07FF | 1KB | GPIOB |
| | 0x5000 0000 - 0x5000 03FF | 1KB | GPIOA |
| | 0x4002 4400 - 0x4FFF FFFF | ~256MB | Reserved |
| | 0x4002 4000 - 0x4002 43FF | 1KB | SQRT |
| | 0x4002 3C00 - 0x4002 3FFF | 1KB | Reserved |
| | 0x4002 3800 - 0x4002 3BFF | 1KB | HWDIV |
| | 0x4002 3400 - 0x4002 37FF | 1KB | Reserved |
| | 0x4002 3000 - 0x4002 33FF | 1KB | CRC |
| | 0x4002 2400 - 0x4002 2FFF | 3KB | Reserved |
| | 0x4002 2000 - 0x4002 23FF | 1KB | FLASH |
| | 0x4002 1C00 - 0x4002 1FFF | 1KB | Reserved |
| | 0x4002 1800 - 0x4002 1BFF | 1KB | EXTI |
| | 0x4002 1400 - 0x4002 17FF | 1KB | Reserved |
| | 0x4002 1000 - 0x4002 13FF | 1KB | RCC |
| | 0x4002 0400 - 0x4002 0FFF | 3KB | Reserved |
| | 0x4002 0000 - 0x4002 03FF | 1KB | DMA |
| APB2 | 0x4001 5C00 - 0x4001 FFFF | 41KB | Reserved |
| | 0x4001 5800 - 0x4001 5BFF | 1KB | DBG |
| | 0x4001 3C00 - 0x4001 57FF | 7KB | Reserved |

| 总线 | 地址范围 | 大小 | 外设 |
|---|---|---|---|
| | 0x4001 3800 - 0x4001 3BFF | 1KB | UART1 |
| | 0x4001 3400 - 0x4001 37FF | 1KB | Reserved |
| | 0x4001 3000 - 0x4001 33FF | 1KB | SPI1 |
| | 0x4001 2C00 - 0x4001 2FFF | 1KB | TIM1 |
| | 0x4001 2800 - 0x4001 2BFF | 1KB | Reserved |
| | 0x4001 2400 - 0x4001 27FF | 1KB | ADC |
| | 0x4001 0400 - 0x4001 23FF | 8KB | Reserved |
| | 0x4001 0300 - 0x4001 03FF | 1KB | OPAMP1/2 |
| | 0x4001 0200 - 0x4001 02FF | | COMP1/2 |
| | 0x4001 01B0 - 0x4001 01FF | | VREFBUF |
| | 0x4001 0000 - 0x4001 01AF | | IRTIM |
| APB1 | 0x4000 8400- 0x4000 FFFF | 31KB | Reserved |
| | 0x4000 8000 - 0x4000 83FF | 1KB | LPUART1 |
| | 0x4000 7C00 - 0x4000 7FFF | 1KB | LPTIM1 |
| | 0x4000 7400 - 0x4000 7BFF | 2KB | Reserved |
| | 0x4000 7000 - 0x4000 73FF | 1KB | PMU |
| | 0x4000 6400 - 0x4000 6FFF | 3KB | Reserved |
| | 0x4000 6000 - 0x4000 63FF | 1KB | HALLTIM |
| | 0x4000 5800 - 0x4000 5FFF | 2KB | Reserved |
| | 0x4000 5400 - 0x4000 57FF | 1KB | I2C1 |
| | 0x4000 4800 - 0x4000 53FF | 3KB | Reserved |
| | 0x4000 4400 - 0x4000 47FF | 1KB | UART2 |
| | 0x4000 3C00 - 0x4000 43FF | 2KB | Reserved |
| | 0x4000 3800 - 0x4000 3BFF | 1KB | SPI2 |
| | 0x4000 3400 - 0x4000 37FF | 1KB | Reserved |
| | 0x4000 3000 - 0x4000 33FF | 1KB | IWDG |
| | 0x4000 2C00 - 0x4000 2FFF | 1KB | Reserved |
| | 0x4000 2800 - 0x4000 2BFF | 1KB | RTC |
| | 0x4000 2400 - 0x4000 27FF | 1KB | LCD |
| | 0x4000 0C00 - 0x4000 23FF | 6KB | Reserved |
| | 0x4000 0800 - 0x4000 0BFF | 1KB | TIM4 |
| | 0x4000 0400 - 0x4000 07FF | 1KB | TIM3 |
| | 0x4000 0000 - 0x4000 03FF | 1KB | Reserved |

## 3.3 启动配置

通过 BOOT0 引脚选择启动模式，如下表：

表 3-3 启动配置

| RDP | BOOT0 pin | 启动模式 |
|---|---|---|
| 0/1/2 | 0 | User flash |
| | 1 | Bootloader |
| 3 | x | User flash |

复位后，根据选择的启动模式，将对应的存储空间重映射到 0x0000 0000 地址；CPU 先从地址 0x0000 0000 获取栈顶值，然后从地址 0x0000 0004 获取代码的启动地址，开始执行程序。

*注意：仅在 POR/PDR 或 NRST 复位时，BOOT0 pin 才会被检测用于启动配置，其他系统复位发生时，忽略 BOOT0 pin，维持上一次启动模式。*

根据启动模式的选择，User flash 或 Bootloader 可以通过如下地址进行访问：

- 从 User flash 启动：User flash（地址开始于 0x0800 0000）的存储空间会同时被映射到地址 0x0000 0000，原始地址 0x0800 0000 的存储空间也可以进行访问；即 User flash 的存储空间通过 0x0000 0000 或 0x0800 0000 均可以访问；
- 从 Bootloader 启动：Bootloader（地址开始于 0x1FFD 0000）的存储空间会同时被映射到地址 0x0000 0000，原始地址 0x1FFD 0000 的存储空间也可以进行访问。

当 RDP 为等级 3 时，固定从 User flash 启动。

### 3.3.1 Bootloader

Bootloader 位于 System memory，用于对 User flash 进行编程。Bootloader 支持以下通信接口：

- UART2：PA13/PA14

更详细的信息参见 Bootloader 用户手册（UM1000）。

# 4 Flash 存储器（Flash）

## 4.1 简介

Flash 存储器连接在 AHB 总线上，由 Flash 控制器统一管理，可对存储器执行取指、读取、编程和擦除操作，并具有安全访问机制。

## 4.2 Flash 主要特性

- 高达 64KB 的用户存储空间，页大小：512 字节
- 32bits 位宽写入，8/16/32bits 位宽读取
- 支持页擦除、批量擦除
- 支持预取功能，可提高 CPU 的指令执行效率
- 可配置 4 种读出保护等级（RDP）
- 可配置的写入保护区域（WRP）

## 4.3 Flash 功能描述

### 4.3.1 Flash 存储器组成

Flash 存储器支持 32 位编程，支持 8/16/32 位读取访问，可存储指令和数据。

Flash 存储器的组成如下：

- User flash 区：用于存储用户程序和数据，存储空间最大为 64KB，包含 128 个页（Page），每页 512 字节；
- System memory 区：用于存储 Bootloader，存储空间为 1392Bytes；
- Option bytes 区：用于存储存储器保护配置的选项字节；
- Engineer 区：用于存储产品信息及出厂校准参数。

表 4-1 Flash 存储器组成

| 区域 | 地址 | 空间大小（字节） | 说明 |
|---|---|---|---|
| User flash | 0x0800 0000 – 0x0800 01FF | 512 | Page 0 |
| | … | … | … |

| 区域 | 地址 | 空间大小（字节） | 说明 |
|---|---|---|---|
| | 0x0800 7E00 – 0x0800 7FFF | 512 | Page 63 |
| | … | … | … |
| | 0x0800 FE00 – 0x0800 FFFF | 512 | Page 127 |
| System memory | 0x1FFD 0000 – 0x1FFD 056F | 1392 | Bootloader |
| Option bytes | 0x1FFF 0000 – 0x1FFF 01FF | 512 | 选项字节存储区 |
| Engineer | 0x1FFF 0200 – 0x1FFF 028F | 144 | 产品信息、出厂校准参数 |

### 4.3.2 Flash 读取

#### 访问等待周期

Flash 访问控制寄存器（FLASH_ACR）中的 LATENCY[1:0]位域，用于配置 Flash 读取访问的等待周期，HCLK 时钟频率与 Flash 读取访问等待周期的对应关系见下表。

表 4-2 HCLK 时钟频率与 Flash 读取访问等待周期的对应关系

| 频率范围 | 等待周期 | LATENCY 配置 |
|---|---|---|
| HCLK ≤ 24MHz | 0 HCLK | LATENCY = 0 |
| 24 < HCLK ≤ 48MHz | 1 HCLK | LATENCY = 1 |
| 48 < HCLK ≤ 72MHz | 2 HCLK | LATENCY = 2 |

改变 HCLK 频率与修改访问等待周期的流程如下：

- 提高 HCLK 频率的配置步骤：

  1） 配置 FLASH_ACR 寄存器中的 LATENCY[1:0]位域，增大 Flash 读取的等待周期；
  2） 读取 LATENCY[1:0]位域，检查等待周期已配置成功；
  3） 提高 HCLK 频率，可通过配置 *RCC 时钟配置寄存器（RCC_CFG）*中的 SYSW[1:0]位域，切换更高频率的时钟源，或通过配置 HPRE[2:0]位域，减小系统时钟的分频值；
  4） 确认系统时钟已切换完成。

- 降低 HCLK 频率的配置步骤：

  1） 降低 HCLK 频率，可通过配置 RCC 时钟配置寄存器（RCC_CFG）中的 SYSW[1:0]位域，切换更低频率的时钟源，或通过配置 HPRE[2:0]位域，增大系统时钟的分频值；

2） 确认系统时钟已切换完成；

3） 配置 FLASH_ACR 寄存器中的 LATENCY[1:0]位域，减小 Flash 读取的等待周期；

4） 读取 LATENCY[1:0]位域，检查等待周期已配置成功。

**预取功能**

Flash 访问控制寄存器（FLASH_ACR）中的 PF_EN 位，用于配置 Flash 预取功能。当 Flash 访问等待周期≥1 时，使能预取功能后，可提高 CPU 的指令执行效率。复位后，预取功能默认处于禁止状态。

### 4.3.3 Flash 解锁

为防止 Flash 被意外修改，增加了保护措施，必须向特定寄存器写入密钥，才能解锁相关功能的配置权限。

**Flash 控制寄存器解锁**

复位后，Flash 控制寄存器（FLASH_CR）将处于写保护锁定状态。要配置 FLASH_CR 寄存器，就要先进行解锁操作。

FLASH_CR 寄存器的解锁操作，必须严格按照以下步骤顺序执行：

1） 向 FLASH_CRKEY 寄存器写入密钥 1：0xE57A 1A85；

2） 向 FLASH_CRKEY 寄存器写入密钥 2：0x7C6E 8391；

3） 检查 FLASH_CR 寄存器中的 LOCK 位，当该位清 0 时，表明 FLASH_CR 寄存器已解锁。

解锁完成后，才能对 FLASH_CR 寄存器进行配置。

密钥必须严格按照顺序写入，否则将产生总线错误。出错后，直到再次复位，才能重新对 FLASH_CR 寄存器进行解锁。

将 FLASH_CR 寄存器中的 LOCK 位重新置 1，能恢复 FLASH_CR 寄存器的写保护锁定状态。通过复位，也能使 FLASH_CR 寄存器恢复成写保护锁定状态。

**Flash 选项字节解锁**

复位后，Option bytes 区处于写保护锁定状态。要对选项字节进行更新，需首先进行解锁操作。

Option bytes 区的解锁操作，必须严格按照以下步骤顺序执行：

1） 先解锁 Flash 控制寄存器 FLASH_CR（详见：*Flash 控制寄存器解锁*）；
2） 向 FLASH_OPTKEY 寄存器写入密钥 1：0x6A89 4D7B；
3） 向 FLASH_OPTKEY 寄存器写入密钥 2：0x7C31 1F5A；
4） 检查 FLASH_CR 寄存器中的 OPTLOCK 位，当该位清 0 时，表明 Option bytes 区已解锁。

解锁完成后，才能对 Option bytes 区执行编程与擦除操作。

密钥须按照顺序正确写入，否则将产生总线错误。出错后，直到再次复位后，才能重新对 Option bytes 区进行解锁。

将 FLASH_CR 寄存器中的 OPTLOCK 位重新置 1，能恢复 Option bytes 区的写保护锁定状态。通过复位，也能使 Option bytes 区恢复成写保护锁定状态。

当 FLASH_CR 寄存器恢复成写保护锁定状态时(LOCK 位置 1)，Option bytes 区也会被恢复成写保护锁定状态，OPTLOCK 位将自动置 1。

### 4.3.4 User flash 区擦除操作

User flash 区支持以下擦除方式：

- 页擦除（512 字节）
- 批量擦除

无论采用哪种擦除模式，都会受到 Flash 安全保护机制的影响。

**User flash 区页擦除步骤**

对 User flash 区进行页擦除操作，可遵循以下步骤：

1） 检查 FLASH_SR 寄存器中的 BSY 标志，确认当前没有正在执行的 Flash 操作；
2） 检查 FLASH_SR 寄存器中的 WRPERR 标志，确认已清除；
3） 解锁 FLASH_CR 寄存器，使 FLASH_CR 寄存器中的 LOCK 位清 0；
4） 配置 FLASH_CR 寄存器中的 OP_MODE[1:0]位域为 10，进入页擦除模式；
5） 配置 FLASH_WRP 寄存器中的待擦除页对应位为 1，使待擦除页写入保护失效；
6） 向待擦除页的任意地址写任意数据（必须 32 位字对齐），启动 Flash 页擦除，同时 BSY 标志将自动置 1；
7） 查询并等待 BSY 标志清 0，表明擦除操作已完成；

8） 如果要对多个页进行擦除操作，可重复执行步骤 5 到 7；
9） 配置 FLASH_CR 寄存器中的 OP_MODE[1:0]位域为 00，退出页擦除模式；
10） 配置 FLASH_WRP 寄存器中的擦除页对应位为 0，恢复擦除页的写入保护状态；
11） 配置 FLASH_CR 寄存器中的 LOCK 位为 1，恢复 FLASH_CR 寄存器的写保护锁定状态。

**User flash 区批量擦除步骤**

批量擦除用于擦除整个 User flash 区域，可遵循以下步骤：

1） 检查 FLASH_SR 寄存器中的 BSY 标志，确认当前没有正在执行的 Flash 操作；
2） 检查 FLASH_SR 寄存器中的 WRPERR 标志，确认已清除；
3） 解锁 FLASH_CR 寄存器，使 FLASH_CR 寄存器中的 LOCK 位清 0；
4） 配置 FLASH_CR 寄存器中的 OP_MODE[1:0]位域为 11，进入批量擦除模式；
5） 配置 FLASH_WRP 寄存器使整个 User flash 写入保护失效；
6） 向 User flash 区写任意数据（必须 32 位字对齐），启动 Flash 批量擦除，同时 BSY 标志将自动置 1；
7） 查询并等待 BSY 标志清 0，表明擦除操作已完成；
8） 配置 FLASH_CR 寄存器中的 OP_MODE[1:0]位域为 00，退出批量擦除模式；
9） 配置 FLASH_WRP 寄存器中的擦除页对应位为 0，恢复擦除页的写入保护状态；
10） 配置 FLASH_CR 寄存器中的 LOCK 位为 1，恢复 FLASH_CR 寄存器的写保护锁定状态。

### 4.3.5 User flash 区编程操作

Flash 存储器支持按照 32bits 位宽执行编程操作。编程目标地址必须按照字对齐，否则写入数据无效，产生总线错误。

与 Flash 擦除操作类似，编程操作也会受到安全保护机制的影响。

**User flash 区编程操作步骤**

对 User flash 区进行编程操作，可遵循以下步骤：

1） 检查 FLASH_SR 寄存器中的 BSY 标志，确认当前没有正在执行的 Flash

操作；

2） 检查 FLASH_SR 寄存器中的 WRPERR 标志，确认均已清除；

3） 解锁 FLASH_CR 寄存器，使 FLASH_CR 中的 LOCK 位清 0；

4） 配置 FLASH_CR 寄存器中的 OP_MODE[1:0]位域为 01，进入编程模式；

5） 配置 FLASH_WRP 寄存器中的待编程页对应位为 1，使待编程页写入保护失效；

6） 向目标地址写入 32 位待写数据，同时 BSY 标志将自动置 1；

7） 查询并等待 BSY 标志清 0，表明编程操作已完成；

8） 如果要对多个地址进行编程操作，可重复步骤 5 到 7；

9） 配置 FLASH_CR 寄存器中的 OP_MODE[1:0]位域为 00，退出编程模式；

10） 配置 FLASH_WRP 寄存器中的编程页对应位为 0，恢复编程页的写入保护状态；

11） 配置 FLASH_CR 寄存器中的 LOCK 位为 1，恢复 FLASH_CR 寄存器的写保护锁定状态。

## 4.4 Option bytes 区操作说明

### 4.4.1 选项字节组成

选项字节存放于 Flash 存储器的 Option bytes 区，用于保存与芯片硬件功能相关的配置项，用户可根据需求对选项字节进行配置。

选项字节仅占用 Option bytes 区第 1 个字，其余存储空间由用户管理，可存放用户数据；为了校验选项字节的正确性，第 1 个字（32bits）被拆分成两部分，低 16bits 存放选项字节，高 16bits 存放选项字节的反码。

选项字节的详细组成如下表所示：

表 4-3 选项字节组成

| 存储地址 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x1FFF 0000 | Reserved | | | | | | | | RDP[7:0] | | | | | | | |

复位后，硬件会自动将 Option bytes 区中的 RDP[7:0]加载到 *Flash 选项字节寄存器 1（FLASH_OPTR1）* 的 RDPRP[1:0]位域中。

### 4.4.2 选项字节加载

要使 Option bytes 区中选项字节生效，可通过 POR/PDR 复位、NRST 复位或系统复位加载生效。

在选项字节加载过程中，将自动对选项字节及其反码进行校验。如果出现错误，错误标志 OPTVERR 置 1。

当选项字节加载生效后，Option bytes 区中的选项字节被自动复制到对应的选项字节加载寄存器。读取选项字节加载寄存器，获得的总是加载生效后的选项字节值。

### 4.4.3 Option bytes 区擦除与编程

#### Option bytes 区擦除步骤

对 Option bytes 区的擦除遵循以下步骤：

1） 检查 FLASH_SR 寄存器中的 BSY 标志位，确认当前没有正在执行的 Flash 操作；
2） 检查 FLASH_SR 寄存器中的 WRPERR 标志，确认已清除；
3） 解锁 FLASH_CR 寄存器，使 FLASH_CR 寄存器中的 LOCK 位清 0；
4） 解锁 Flash 选项字节的写保护，使 FLASH_CR 寄存器中的 OPTLOCK 位清 0；
5） 配置 FLASH_CR 寄存器中的 OP_MODE[1:0]位域为 10，进入页擦除模式；
6） 向 Option bytes 区任意地址写任意数据（必须 32 位字对齐），启动选项字节页擦除，同时 BSY 标志将自动置 1；
7） 查询并等待 BSY 标志清 0，表明擦除操作已完成；
8） 配置 FLASH_CR 寄存器中的 OP_MODE[1:0]位域为 00，退出页擦除模式；
9） 配置 FLASH_CR 寄存器中的 OPTLOCK 位为 1，恢复选项字节的写保护锁定状态；
10） 配置 FLASH_CR 寄存器中的 LOCK 位为 1，恢复 FLASH_CR 寄存器的写保护锁定状态。

#### Option bytes 区编程步骤

Option bytes 区仅支持 32bits 编程，其中第 1 个字低 16bits 存放选项字节，高 16bits 存放选项字节的反码，反码由用户计算并写入；其余存储空间无正反码要求。

对 Option bytes 区的编程遵循以下步骤：

1） 检查 FLASH_SR 寄存器中的 BSY 标志位，确认当前没有正在执行的 Flash 操作；
2） 检查 FLASH_SR 寄存器中的 WRPERR 标志，确认已清除；
3） 解锁 FLASH_CR 寄存器，使 FLASH_CR 寄存器中的 LOCK 位清 0；
4） 解锁 Flash 选项字节的写保护，使 FLASH_CR 寄存器中的 OPTLOCK 位清 0；
5） 配置 FLASH_CR 寄存器中的 OP_MODE[1:0]位域为 01，进入 Flash 编程模式；
6） 向 Option bytes 区的目标地址写入待编程数据（用户需保证数据高 16bit 与低 16bit 符合正反码校验），同时 BSY 标志将自动置 1；
7） 查询并等待 BSY 标志清 0，表明编程操作已完成；
8） 配置 FLASH_CR 寄存器中的 OP_MODE[1:0]位域为 00，退出编程模式；
9） 配置 FLASH_CR 寄存器中的 OPTLOCK 位为 1，恢复选项字节的写保护锁定状态；
10） 配置 FLASH_CR 寄存器中的 LOCK 位为 1，恢复 FLASH_CR 寄存器的写保护锁定状态。

在完成对选项字节的编程后，直接读取 Option bytes 区，可获取更新后的选项字节值。但是更新后的选项字节此时并未生效，如果读取选项字节加载寄存器，获得的仍将是最近一次已加载生效的选项字节值。

## 4.5 Engineer 区

Engineer 区存储产品信息和出厂校准参数，出厂写入，软件可读不可写。

表 4-4 产品信息

| 存储地址 | 长度（字） | 参数描述 |
|---|---|---|
| 0x1FFF 0234 | 3 | 96 位的产品唯一身份识别码 |
| 0x1FFF 0274 | 1 | Flash 空间大小，单位：Kbytes<br>0x00000040：64 |
| 0x1FFF 0278 | 1 | SRAM 空间大小，单位：Kbytes<br>0x00000008：8 |

表 4-5 出厂校准参数组成

| 存储地址 | 位域[15:0] | 定义 | 参数描述 |
|---|---|---|---|
| 0x1FFF 0208 | [5:0] | RCH48_CAL_FINE[5:0] | RCH48M 时钟细校准值 |
| | [9:8] | RCH48_CAL_COARSE[1:0] | RCH48M 时钟粗校准值 |
| 0x1FFF 020C | [5:0] | RCH64_CAL_FINE[5:0] | RCH64M 时钟细校准值 |
| | [9:8] | RCH64_CAL_COARSE[1:0] | RCH64M 时钟粗校准值 |
| 0x1FFF 0210 | [5:0] | RCH72_CAL_FINE[5:0] | RCH72M 时钟细校准值 |
| | [9:8] | RCH72_CAL_COARSE[1:0] | RCH72M 时钟粗校准值 |
| 0x1FFF 0214 | [4:0] | RCL_CAL[4:0] | RCL 时钟校准值 |
| 0x1FFF 0218 | [5:0] | VREFBUF_CAL_20[5:0] | VREFBUF 2.048V 的校准值 |
| | [13:8] | VREFBUF_CAL_25[5:0] | VREFBUF 2.5V 的校准值 |
| 0x1FFF 021C | [5:0] | VREFBUF_CAL_36[5:0] | VREFBUF 3.6V 的校准值 |
| 0x1FFF 0220 | [5:0] | VREFBUF_CAL_42[5:0] | VREFBUF 4.2V 的校准值 |
| 0x1FFF 027C | [11:0] | BGR_CAL[11:0] | 内部带隙基准电压校准参数 |
| 0x1FFF 0280 | [11:0] | TS_CAL_25[11:0] | 25℃下温度传感器校准值 |

*注意： 出厂校准参数，低 16bits 为有效数据，高 16bits 为反码。*

## 4.6 Flash 安全保护机制

Flash 存储器支持以下安全保护机制：

- Flash 读出保护（RDP）：通过配置不同 RDP 保护等级，对存储器的操作权限进行限制；
- Flash 写入保护（WRP）：禁止对受保护的区域执行编程或擦除操作，但允许取指和读取操作。

### 4.6.1 Flash 读出保护

**RDP 保护权限**

Flash 读出保护（RDP），共有四种保护等级：

**RDP 0：**

- 从 User flash 启动

- User flash 区：允许读取、编程、擦除
- Option bytes 区：允许读取、擦除、编程
- System memory 区：允许读取
- Engineer 区：允许读取

- 从 Bootloader 启动
  - User flash 区：允许读取、编程、擦除
  - Option bytes 区：允许读取、擦除、编程
  - System memory 区：允许读取
  - Engineer 区：允许读取
- 调试接口
  - User flash 区：允许读取、编程、擦除
  - Option bytes 区：允许读取、擦除、编程
  - System memory 区：允许读取
  - Engineer 区：允许读取

**RDP 1：**

- 从 User flash 启动
  - User flash 区：允许读取、编程、擦除
  - Option bytes 区：允许读取、擦除、编程
  - System memory 区：允许读取
  - Engineer 区：允许读取
- 从 Bootloader 启动
  - User flash 区：禁止访问
  - Option bytes 区：允许读取、擦除、编程
  - System memory 区：允许读取
  - Engineer 区：允许读取

- 调试接口
  - User flash 区：禁止访问
  - Option bytes 区：允许读取、擦除、编程
  - System memory 区：允许读取
  - Engineer 区：允许读取

**RDP 2：**

- 从 User flash 启动
  - User flash 区：允许读取、编程、擦除
  - Option bytes 区：允许读取、擦除、编程
  - System memory 区：允许读取
  - Engineer 区：允许读取
- 从 Bootloader 启动
  - User flash 区：禁止访问
  - Option bytes 区：允许读取、擦除、编程
  - System memory 区：允许读取
  - Engineer 区：允许读取
- 禁用调试接口

**RDP 3：**

- 从 User flash 启动
  - User flash 区：允许读取、编程、擦除
  - Option bytes 区：允许读取
  - System memory 区：允许读取
  - Engineer 区：允许读取
- 禁止从 Bootloader 启动
- 禁用调试接口

对于 Flash 存储器，当 Flash 控制器检测到无权限的非法操作时，非法操作会被立刻终止，并产生总线访问错误。

### RDP 保护等级配置

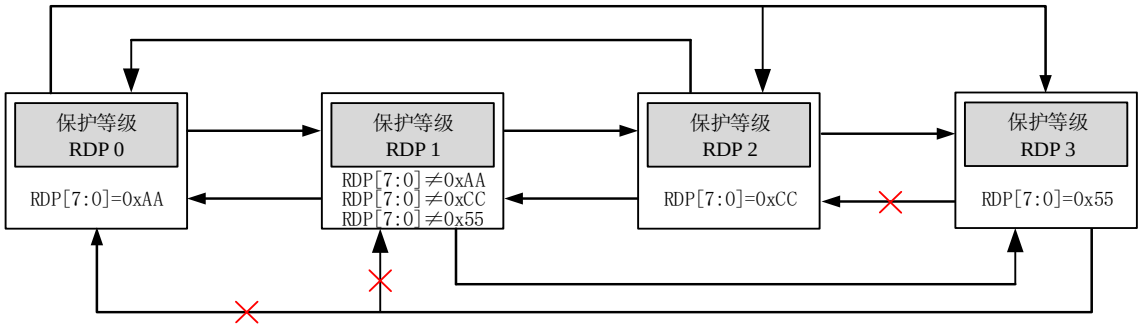
选项字节中 RDP[7:0]位域用于配置 RDP 保护等级，当选项字节加载生效后，FLASH_OPTR1 寄存器中 RDPRP[1:0]位域指示当前保护等级，选项字节的相关使用方法详见：*Option bytes 区操作说明*。

表 4-6 RDP 保护等级配置

| RDP 保护等级 | RDP[7:0]配置值 |
|---|---|
| RDP 0 | 0xAA（默认值） |
| RDP 1 | 除 0xAA、0xCC 和 0x55 以外其它值 |
| RDP 2 | 0xCC |
| RDP 3 | 0x55 |

RDP 保护等级变更可通过修改选项字节中 RDP[7:0]位域实现。当保护等级为 RDP3 时，保护等级锁定无法降级。RDP 保护等级的变更规则如下图所示：

图 4-1 RDP 保护等级变更




当 RDP 升级时，除选项字节 RDP[7:0]外，存储器中的内容保持不变。

当 RDP 降级时，为防止存储器中的内容被读取或修改，硬件将自动执行以下操作：

- 擦除整个 User flash 区中的内容；
- 选项字节中 RDP[7:0]更新为降级编程值。

### RDP 保护等级降级操作步骤

对 RDP 保护等级进行降级操作，可遵循以下步骤：

1） 执行 *Option bytes 区擦除步骤*的 1-7 步；

2） 配置 FLASH_CR 寄存器中的 OP_MODE[1:0]位域为 01，进入 Flash 编程模式；

3） 向 Option bytes 区的 0x1FFF 0000 地址写入待编程数据（用户需保证数据高 16bit 与低 16bit 符合正反码校验），将启动 Flash 降级，同时 BSY 标志将自动置 1；

4） 查询并等待 BSY 标志清 0，表明降级操作完成。

### 4.6.2 Flash 写入保护

写入保护寄存器（FLASH_WRP）仅作用于 User flash 区，只允许对受保护区域执行取指或读取操作，禁止执行擦除或编程操作。对受保护的区域执行非法操作，写保护错误标志 WRPERR 置 1。

在 User flash 区，WRP[7:0]每个控制位保护区域为 8K，各保护区域相互独立。WRP 保护区域配置见下表：

表 4-7 WRP 保护区域配置

| 寄存器配置 | WRP 保护区域 |
|---|---|
| WRP[0] | 0x0800 0000 - 0x0800 1FFF |
| WRP[1] | 0x0800 2000 - 0x0800 3FFF |
| WRP[2] | 0x0800 4000 - 0x0800 5FFF |
| WRP[3] | 0x0800 6000 - 0x0800 7FFF |
| WRP[4] | 0x0800 8000 - 0x0800 9FFF |
| WRP[5] | 0x0800 A000 - 0x0800 BFFF |
| WRP[6] | 0x0800 C000 - 0x0800 DFFF |
| WRP[7] | 0x0800 E000 - 0x0800 FFFF |

芯片发生系统复位后，User flash 区处于写入保护状态，对 User flash 区的擦除或编程操作需先配置 FLASH_WRP 寄存器中的待擦除或待编程页对应位为 1，使写入保护失效。在配置了有效的 WRP 保护区域后，保护区域范围立即生效。

## 4.7 Flash 操作错误

#### 写保护错误标志 WRPERR

对 User flash 启动擦除或编程操作时，若 WRPERR 未被清除，则擦除或编程操作不会启动。

发生以下情况时，相应操作不会启动，WRPERR 置 1：

- 擦除受 WRP 保护的区域时；
- 编程受 WRP 保护的区域时。

**选项字节校验错误标志 OPTVERR**

当选项字节加载值与其反码不匹配时，OPTVERR 置 1。

## 4.8 Flash 中断

- Flash 操作完成中断

  配置 FLASH_CR 寄存器中的 EOPIE 位为 1，使能该中断，当 Flash 完成编程操作、擦除操作时，FLASH_SR 寄存器中 EOP 标志将置 1 并触发中断。

- Flash 操作异常中断

  配置 FLASH_CR 寄存器中 OPERRIE 位为 1，使能该中断，当 WRPERR 位置 1 时，触发中断。

表 4-8 中断请求

| 中断事件 | 事件标志 | 中断使能位 | 事件标志/中断清除方法 |
|---|---|---|---|
| Flash 操作完成 | EOP | EOPIE | 向 EOP 标志位写 1 |
| Flash 操作异常 | WRPERR | OPERRIE | 向 WRPERR 标志位写 1 |

## 4.9 Flash 寄存器

Flash 寄存器支持 32 位访问。

表 4-9 Flash 基地址

| 外设 | 基地址 |
|---|---|
| Flash | 0x4002 2000 |

### 4.9.1 Flash 访问控制寄存器（FLASH_ACR）

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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:5 | 保留 | 写入无效 |
| 4 | PF_EN | 预取使能<br>0：禁止<br>1：使能 |
| 3:2 | 保留 | 写入无效 |
| 1:0 | LATENCY[1:0] | Flash 读取访问等待周期<br>00：0 等待周期<br>01：1 等待周期<br>10：2 等待周期<br>11：保留 |

### 4.9.2 Flash 控制解锁密钥寄存器（FLASH_CRKEY）

偏移地址：0x08

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | CRKEY[31:0] | 用于解锁Flash控制寄存器（FLASH_CR）<br>解锁时按顺序写入以下密钥：<br>密钥1：0xE57A 1A85<br>密钥2：0x7C6E 8391<br>解锁步骤及注意事项详见：*Flash控制寄存器解锁*。 |

### 4.9.3 Flash 选项字节解锁密钥寄存器（FLASH_OPTKEY）

偏移地址：0x0C

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | OPTKEY[31:0] | 用于解锁选项字节区<br>解锁时按顺序写入以下密钥：<br>密钥1：0x6A89 4D7B<br>密钥2：0x7C31 1F5A<br>解锁步骤及注意事项详见：*Flash选项字节解锁*。 |

### 4.9.4 Flash 状态寄存器（FLASH_SR）

偏移地址：0x10

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | EOP | Res. | | | | | | | BSY |
| | | | | | | | rc_w1 | | | | | | | | r |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OPTVERR | Res. | | | | | | | | | | WRPERR | Res. | | | |
| r | | | | | | | | | | | rc_w1 | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:25 | 保留 | 写入无效 |
| 24 | EOP | Flash操作完成标志<br>当Flash完成编程操作、擦除操作时，此位将置1。<br>该标志写1清0，写0无效。<br>0：Flash操作未完成<br>1：Flash操作已完成 |
| 23:17 | 保留 | 写入无效 |
| 16 | BSY | Flash操作状态标志<br>当对Flash执行操作时，此位将自动置1，在操作完成后，此位将自动清0。<br>0：Flash处于空闲状态<br>1：Flash处于忙状态，操作正在执行 |
| 15 | OPTVERR | 选项字节校验错误标志<br>当选项字节与其反码不匹配时，该标志置1。<br>0：正常<br>1：发生校验错误 |
| 14:5 | 保留 | 写入无效 |
| 4 | WRPERR | 写保护错误标志<br>当擦除或编程受WRP保护的区域时，该标志置1。<br>该标志写1清0，写0无效。<br>0：正常<br>1：发生写保护错误 |

| | | |
|---|---|---|
| 3:0 | 保留 | 写入无效 |

### 4.9.5 Flash 控制寄存器（FLASH_CR）

偏移地址：0x14

复位值：0xC000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LOCK | OPTLOCK | Res. | | | | OPERRIE | EOPIE | Res. | | | | | | | |
| rs | rs | | | | | rw | rw | | | | | | | | |

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31 | LOCK | FLASH_CR锁定控制<br>此位置1后，FLASH_CR将被写保护锁定。<br>当检测到正确的解锁序列时，此位自动清0。<br>解锁步骤及注意事项详见：*Flash控制寄存器解锁*。 |
| 30 | OPTLOCK | 选项字节锁定控制<br>此位置1后，选项字节区将被写保护锁定。<br>当检测到正确的解锁序列时，此位自动清0。<br>解锁步骤及注意事项详见：*Flash选项字节解锁*。 |
| 29:26 | 保留 | 写入无效 |
| 25 | OPERRIE | 编程/擦除操作异常中断使能控制<br>此位置1后，当FLASH_SR寄存器中的WRPERR置1时，将触发中断。<br>0：禁止<br>1：使能 |
| 24 | EOPIE | 编程/擦除操作完成中断使能控制<br>此位置1后，当FLASH_SR寄存器中的EOP标志置1时，将触发中断。<br>0：禁止<br>1：使能 |

| | | |
|---|---|---|
| 23:3 | 保留 | 写入无效 |
| 2:1 | OP_MODE[1:0] | Flash操作模式选择<br>00：退出编程/擦除模式<br>01：编程模式<br>10：页擦除模式<br>11：批量擦除模式 |
| 0 | 保留 | 写入无效 |

### 4.9.6 Flash 选项字节寄存器 1（FLASH_OPTR1）

偏移地址：0x20

复位值：bit：00000000 00000000 00000000 000000XX

出厂默认值：0x0000 0001

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:2 | 保留 | 写入无效 |
| 1:0 | RDPRP[1:0] | RDP保护等级选择<br>0x0：等级0（RDP0）<br>0x1：等级1（RDP1）<br>0x2：等级2（RDP2）<br>0x3：等级3（RDP3） |

### 4.9.7 Flash 写入保护寄存器（FLASH_WRP）

偏移地址：0x38

复位值：0x0000 0000

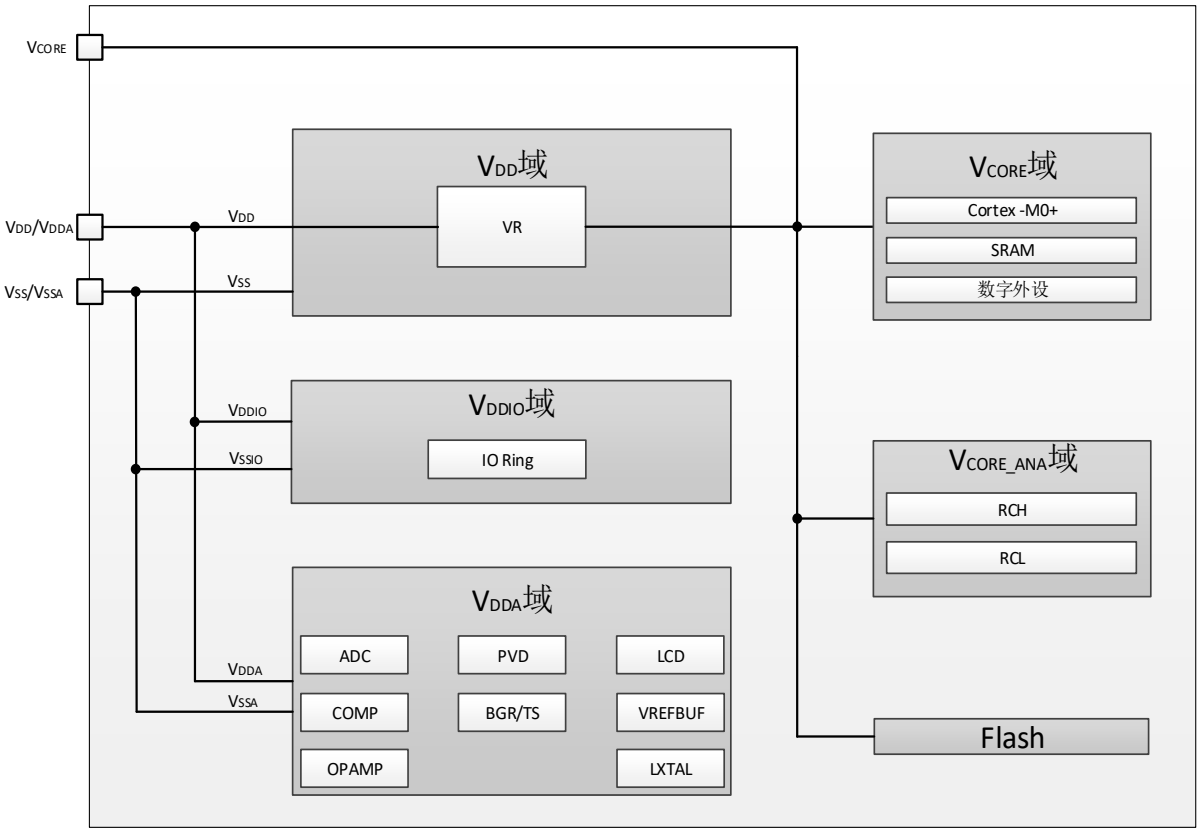
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. |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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 写入无效 |
| 7:0 | WRP [7:0] | WRP保护区域<br>WRP每bit对应8K保护区域，详见*表：WRP保护区域配置*<br>0：写保护生效<br>1：写保护失效 |

# 5 电源管理单元（PMU）

## 5.1 电源

图 5-1 电源结构框图




芯片电源分为两个电源域，具体如下：

- $V_{DD}/V_{DDA}$ 引脚输入电压为 2.0V~5.5V，支持 POR/PDR 和 PVD，PVD 可触发复位或中断，详见：*可编程电压检测（PVD）*；
- $V_{DD}$ 域包含调压器（VR），支持 MR 和 LPR 两种模式，$V_{CORE}$ 域、$V_{CORE_ANA}$ 域和 Flash 均由 VR 供电；
- $V_{CORE}$ 引脚需外接电容，推荐外接 1μF+0.1μF 的电容。

复位后 VR 默认 MR 模式，根据不同的低功耗模式，VR 以两种方式工作：

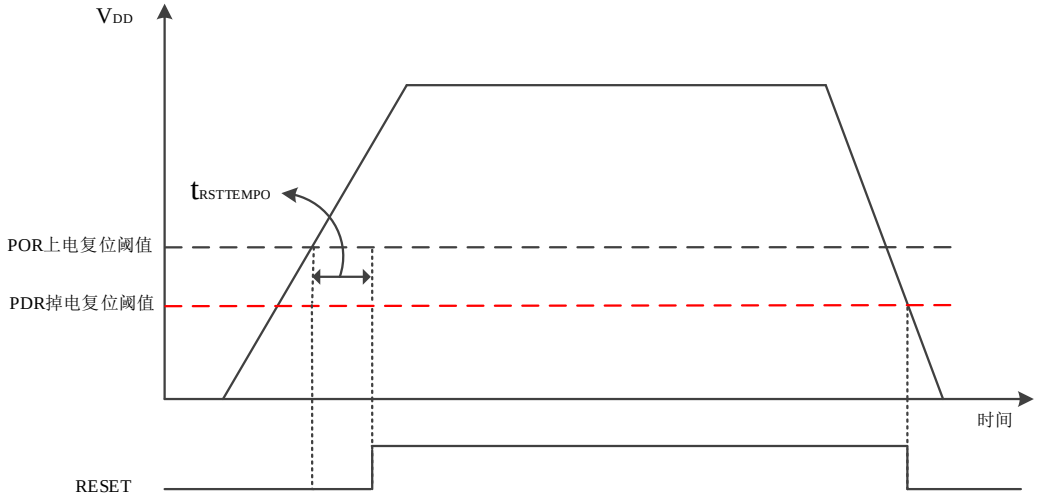
- Run 或 Sleep 模式下，VR 处于 MR 模式；
- Stop 模式下，VR 处于 LPR 模式。

## 5.2 上电复位和掉电复位

芯片内置上电复位（POR）和掉电复位模块（PDR），该模块可工作在所有功耗模式。

当 $V_{DD}$ 电压达到 POR 上电复位阈值时，等待 $t_{RSTTEMPO}$ 后释放复位，当 $V_{DD}$ 电压低于 PDR 掉电复位阈值时，触发复位。

图 5-2 POR/PDR 阈值




## 5.3 可编程电压检测（PVD）

PVD 用于监测 $V_{DD}$ 引脚电压，实现上电复位释放后的低电压检测。当监测电压与阈值比较结果满足设定条件时，可触发复位或中断。

### 5.3.1 主要特性

- 8 阶阈值电压，范围 2.2V ~ 4.6V
- 3 种报警触发条件，可组合配置：
  - 电平触发：监测电压低于阈值时触发报警
  - 下降沿触发：监测电压从高于阈值到低于阈值变化时触发报警
  - 上升沿触发：监测电压从低于阈值到高于阈值变化时触发报警
- 2 种触发动作，复位或中断
- 具备迟滞功能，抗干扰

- 可编程滤波时间，防止误触发
- 支持低功耗模式唤醒

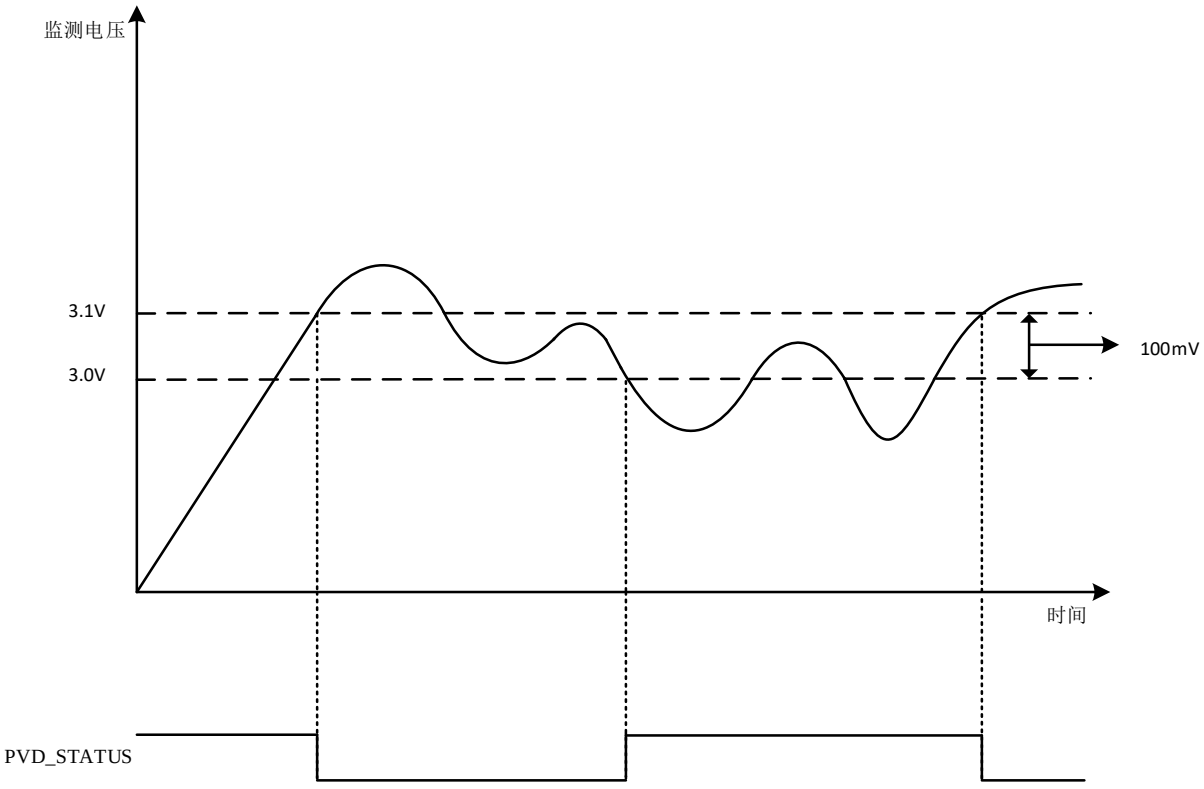
### 5.3.2 低电压检测

PVD 可用于低电压检测，通过配置阈值、滤波、触发条件和触发动作实现对 $V_{DD}$ 电压监测和报警处理。

#### 阈值选择

PVD 阈值有多档可选，由*电源管理控制寄存器 2（PMU_CR2）*中 PVD_LEVEL[2:0]决定；PVD 具有迟滞功能，当 PVD 阈值等级选定后，阈值下限电压会比阈值上限电压低 100mV，参见下图（PVD_LEVEL[2:0]为 010）。

图 5-3 PVD 阈值




#### 数字滤波

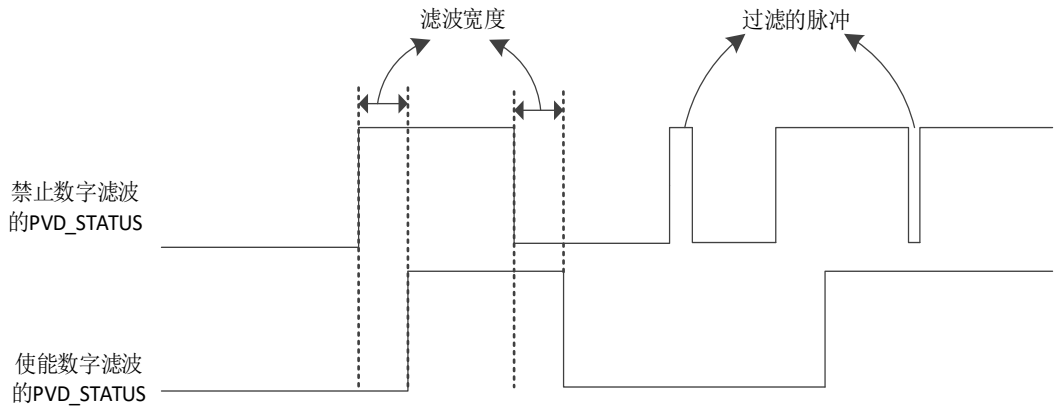
PVD 支持数字滤波，通过将 PMU_CR2 寄存器的 FILTER_EN 位置 1 来使能数字滤波；当滤波使能时，PVD_STATUS 为滤波后 PVD 状态：

- 滤波时钟可选择 SYSCLK、RCL 或 LXTAL，通过 PMU_CR2 寄存器的 FILTER_CLK[1:0]位域选择滤波时钟源；滤波时钟选择 RCL 或 LXTAL 时，

可正常工作在 Stop 模式；

- 滤波时间通过 PMU_CR2 寄存器的 FILTER_TIME[2:0]位域配置。

图 5-4　PVD 滤波




**中断/系统复位选择**

PVD 报警触发动作可系统复位或中断二选一，通过 PMU_CR2 寄存器的 PVD_ACT 位选择：

- PVD_ACT 为 0 时，触发 PVD 报警会将 PVD_INTF 位会置 1，如果 PVD 报警中断使能（PVD_IE 位为 1），则产生 PVD 报警中断；
- PVD_ACT 为 1 时，触发 PVD 报警会产生 PVD 电压检测系统复位。

### 5.3.3　操作示例

**PVD 配置为低电压复位**

1) 配置 PMU_CR2 寄存器中 PVD_LEVEL[2:0]位域，选择报警阈值；
2) 将 PMU_CR2 寄存器中 LT_EN 置 1，选择监测电压低于阈值报警；
3) 配置 PMU_CR2 寄存器中 FILTER_CLK[1:0]位域和 FILTER_TIME[2:0]，选择滤波时钟和滤波时间；
4) 将 PMU_CR2 寄存器中 FILTER_EN 置 1，使能滤波功能；
5) 将 PMU_CR2 寄存器中 PVD_ACT 置 1，选择报警时触发复位；
6) 将 PMU_CR2 寄存器中 PVD_EN 置 1，使能 PVD。

**PVD 配置为电压变化触发中断**

1) 配置 PMU_CR2 寄存器中 PVD_LEVEL[2:0]位域，选择报警阈值；
2) 配置 PMU_CR2 寄存器中 FT_EN 和 RT_EN，选择报警触发边沿；
3) 配置 PMU_CR2 寄存器中 FILTER_CLK[1:0]位域和 FILTER_TIME[2:0]，

选择滤波时钟和滤波时间；

4) 将 PMU_CR2 寄存器中 FILTER_EN 置 1，使能滤波功能；
5) 将 PMU_CR2 寄存器中 PVD_ACT 置 0，选择报警时触发中断；
6) 将 PMU_CR2 寄存器中 PVD_IE 置 1，使能 PVD 中断；
7) 将 PMU_CR2 寄存器中 PVD_EN 置 1，使能 PVD。

## 5.4 内部带隙基准电压

内部带隙基准电压（BGR，典型电压 0.8V）有多种用途，具体如下：

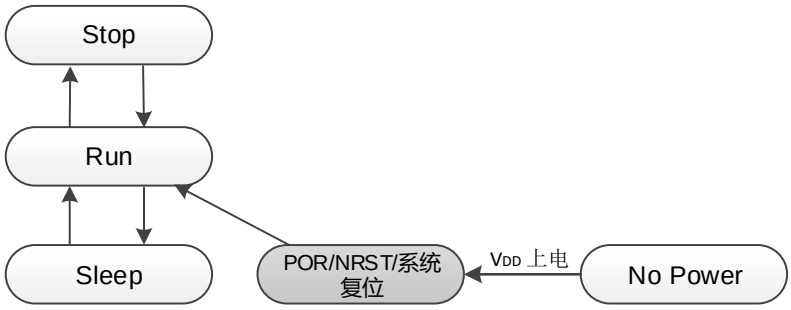
- 可为 TS（温度传感器）和 VREFBUF 提供基准电压；
- 可作为 ADC 的内部通道输入。

通过将 BGR_EN 置 1，来使能 BGR；使能 BGR 后，需等待 BGR 稳定（$t_{START}$ 启动稳定时间参见数据手册），上述关联功能才可使用。

## 5.5 低功耗模式

复位后，芯片处于 Run 模式，系统时钟源为 RCHSYS（4MHz）。芯片提供多种低功耗模式以降低系统功耗；另外，还可以通过降低系统时钟频率和关闭未使用外设的时钟来降低 Run 模式下的系统功耗。

图 5-5 低功耗模式转换框图




芯片提供两种低功耗模式，不同功耗模式差异以及可用外设参见下表：

表 5-1 低功耗模式概述

| 功耗模式 | 模式描述 | 进入方式 | 唤醒源 | 唤醒后时钟 |
|---|---|---|---|---|
| Sleep | 仅关闭 CPU 时钟 | SLEEPDEEP 清 0<br>WFI 或从中断服务函数中返回 | 所有 NVIC 中断 | 与进入前的时钟保持一致 |

| 功耗模式 | 模式描述 | 进入方式 | 唤醒源 | 唤醒后时钟 |
|---|---|---|---|---|
| | | SLEEPDEEP 清 0<br>WFE | 唤醒事件 | |
| Stop | $V_{CORE}$ 电源域（包括 CPU、数字外设）时钟关闭；RCH 和 EXTCLK 关闭，RCL 和 LXTAL 可正常工作；Flash 处于 PowerDown 状态；SRAM 数据保持。 | SLEEPDEEP 置 1<br>WFI 或从中断服务函数中返回 | 与 EXTI 信号线连接的外设生成的中断 | RCHSYS（频率由 RCHFREQ[1:0] 和 RCHDIV[2:0]位域决定） |
| | | SLEEPDEEP 置 1<br>WFE | 与 EXTI 信号线连接的外设生成的唤醒事件 | |

表 5-2 低功耗模式下可用外设 (1)

| 外设 | Run | Sleep | Stop | |
|---|---|---|---|---|
| | | | 功能 | 唤醒源 |
| CPU | √ | × | × | × |
| Flash | √ | √ | × | × |
| SRAM | √ | √ | √ | × |
| POR/PDR | √ | √ | √ | √ |
| PVD | ○ | ○ | ○ | ○ |
| NRST | √ | √ | √ | √ |
| RCH | ○ | ○ | ○ (2) | × |
| RCL | ○ | ○ | ○ | × |
| EXTCLK | ○ | ○ | × | × |
| LXTAL | ○ | ○ | ○ | × |
| LXTAL_CSS | ○ | ○ | ○ | ○ |
| DMA | ○ | ○ | × | × |
| I2C1 | ○ | ○ | × | × |
| SPI1/2 | ○ | ○ | × | × |
| LPUART1 | ○ | ○ | ○ | ○ |
| UART1/2 | ○ | ○ | × | × |
| TIM1/3/4 | ○ | ○ | × | × |
| HALLTIM | ○ | ○ | × | × |
| LPTIM1 | ○ | ○ | ○ | ○ |
| IRTIM | ○ | ○ | × | × |
| IWDG | ○ | ○ | ○ | ○ |
| RTC | ○ | ○ | ○ | ○ |

| 外设 | Run | Sleep | Stop | |
|---|---|---|---|---|
| | | | 功能 | 唤醒源 |
| LCD | ○ | ○ | ○ | × |
| SysTick | ○ | ○ | × | × |
| ADC | ○ | ○ | × | × |
| BGR | ○ | ○ | ○ | × |
| VREFBUF | ○ | ○ | ○ | × |
| COMP1/2 | ○ | ○ | ○ | ○ |
| OPAMP1/2 | ○ | ○ | ○ | × |
| TS | ○ | ○ | × | × |
| GPIO | ○ | ○ | ○ | ○ |
| CRC | ○ | ○ | × | × |
| HWDIV | ○ | ○ | × | × |
| SQRT | ○ | ○ | × | × |
| DBG | ○ | ○ | × | × |

1. “√”表示常开，“×”表示不可用，“○”表示可配置；
2. 在 Stop 下，可通过置位*时钟控制/控制寄存器（RCC_CSR1）*中 RCH_AWON 位保持 RCH 使能。

### 5.5.1 Run

芯片复位后进入 Run 模式，默认系统时钟源为 RCHSYS，时钟频率为 4MHz，所有外设均可使用。

该模式下，为降低功耗可进行如下配置：

- 配置相关预分频寄存器，降低 HCLK 和 PCLK 的时钟频率；
- 禁止不使用的外设时钟；
- 为了进一步降低功耗，可以选择进入 Sleep 或 Stop 模式。

### 5.5.2 Sleep

进入 Sleep 模式仅关闭 CPU 的时钟，所有外设均可正常工作；退出 Sleep 模式后，芯片进入 Run 模式，系统时钟源维持不变。

表 5-3 Sleep 说明

| 行为类型 | 描述说明 |
|---|---|
| 进入 Sleep | 1. SLEEPDEEP 清 0；<br>2. 执行 WFI； |

| 行为类型 | 描述说明 |
| --- | --- |
| | 1. SLEEPDEEP 清 0；<br>2. 无唤醒事件处于挂起状态（无待处理的唤醒事件）；<br>3. 执行 WFE； |
| | 1. SLEEPDEEP 清 0；<br>2. SLEEPONEXIT 置 1；<br>3. 退出最低优先级中断服务函数； |
| 退出 Sleep | 通过 WFI 或退出中断服务函数方式进入低功耗模式：所有的中断服务请求（中断使能）； |
| | 通过 WFE（SEVONPEND 为 0）方式进入低功耗模式：<br>– 所有的中断服务请求（NVIC 中断使能）；<br>– EXTI 生成的事件； |
| | 通过 WFE（SEVONPEND 为 1）方式进入低功耗模式：<br>– 所有的中断服务请求，不论是否 NVIC 中断使能；<br>– EXTI 生成的事件； |
| 退出延迟 | 无 |

### 5.5.3 Stop

Stop 基于 Cortex-M0+的深度休眠功能实现，$V_{CORE}$ 电源域（包括 CPU、数字外设）时钟关闭，RCH 和 EXTCLK 处于 PowerDown 状态，RCL 和 LXTAL 根据配置可保持正常运行。

LPUART1 可开启并使用 RCH 来检测唤醒事件，当唤醒事件匹配时唤醒系统，如果不匹配，则关闭 RCH。

在 Stop 模式下，SRAM 和寄存器的数据保持，Flash 处于 PowerDown 状态。如果 Flash 处于编程或擦除过程中，会等待 Flash 操作完成后进入 Stop。

Stop 下可工作的外设参见*表：低功耗模式下可用外设*。

为了进一步降低 Stop 的功耗，可进行如下配置：

- 在 Stop 下如果未使用到 BGR，可在进入低功耗模式前禁止 BGR 来降低系统功耗；退出 Stop 后，使用 BGR 需重新使能 BGR，并等待 BGR 稳定（$t_{START}$ 启动稳定时间参见数据手册）；
- ADC 和 TS 在 Stop 模式下虽然无法工作，但仍会产生功耗；进入 Stop 前，关闭这些外设可进一步降低功耗。

退出 Stop 模式后，芯片进入 Run 模式，此时系统时钟源为 RCHSYS，频率由*时钟控制/状态寄存器1（RCC_CSR1）*的 RCHFREQ[1:0]和 RCHDIV[2:0]位域决定。

表 5-4　Stop 说明

| 行为类型 | 描述说明 |
|---|---|
| 进入 Stop | 1. 配置 RCHFREQ[1:0]和 RCHDIV[2:0]位域，选择退出 Stop 后的时钟频率；<br>2. SLEEPDEEP 置 1；<br>3. 执行 WFI； |
| | 1. 配置 RCHFREQ[1:0]和 RCHDIV[2:0]位域，选择退出 Stop 后的时钟频率；<br>2. SLEEPDEEP 置 1；<br>3. 无唤醒事件处于挂起状态（无待处理的唤醒事件）；<br>4. 执行 WFE； |
| | 1. 配置 RCHFREQ[1:0]和 RCHDIV[2:0]位域，选择退出 Stop 后的时钟频率；<br>2. SLEEPDEEP 置 1；<br>3. SLEEPONEXIT 置 1；<br>4. 退出最低优先级中断服务函数； |
| 退出 Stop | 通过 WFI 或退出中断服务函数方式进入低功耗模式：与 EXTI 信号线连接的外设的中断服务请求（中断使能），参见*表：EXTI 通道及信号对应表*； |
| | 通过 WFE（SEVONPEND 为 0）方式进入低功耗模式：<br>– 与 EXTI 信号线连接的外设中断服务请求（NVIC 中断使能），参见*表：EXTI 通道及信号对应表*；<br>– EXTI 生成的事件； |
| | 通过 WFE（SEVONPEND 为 1）方式进入低功耗模式：<br>– 与 EXTI 信号线连接的外设中断服务请求，不论是否 NVIC 中断使能，参见*表：EXTI 通道及信号对应表*；<br>– EXTI 生成的事件； |
| 退出延迟 | RCH 唤醒时间+Flash 唤醒时间 |

## 5.6 PMU 寄存器

PMU 寄存器支持 32 位访问。

表 5-5 PMU 基地址

| 外设 | 基地址 |
|---|---|
| PMU | 0x4000 7000 |

### 5.6.1 电源管理控制寄存器 1（PMU_CR1）

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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BGR_EN | Res. | | | | | | VAON_WEN | Res. | | | | | | | |
| rw | | | | | | | rw | | | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15 | BGR_EN | BGR使能<br>0：禁止<br>1：使能 |
| 14:9 | 保留 | 写入无效 |
| 8 | VAON_WEN | RCC_AWCR寄存器写使能<br>0：禁止写RCC_AWCR寄存器<br>1：允许写RCC_AWCR寄存器 |
| 7:0 | 保留 | 写入无效 |

### 5.6.2 电源管理控制寄存器 2（PMU_CR2）

偏移地址：0x04

复位值：0x0000 0000

*注意：* *仅 POR/PDR 和 NRST 复位时，复位该寄存器。*

*PVD 使能后（PVD_EN 为 1），无法修改 PVD 阈值等级、报警触发条件、滤波使能、滤波时间和滤波时钟源的配置。*

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PVD_IE | Res. | LT_EN | FT_EN | RT_EN | FILTER_EN | FILTER_TIME[2:0] | | | Res. | | | PVD_LEVEL[2:0] | | | PVD_EN |
| rw | | rw | rw | rw | rw | rw | rw | rw | | | | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:22 | 保留 | 写入无效 |
| 21:20 | FILTER_CLK[1:0] | PVD滤波时钟源选择<br>00：SYSCLK<br>01：RCL<br>10：LXTAL<br>11：保留（默认为SYSCLK） |
| 19:17 | 保留 | 写入无效 |
| 16 | PVD_ACT | PVD触发动作选择<br>0：中断<br>1：系统复位 |
| 15 | PVD_IE | PVD报警中断使能<br>0：禁止<br>1：使能 |
| 14 | 保留 | 写入无效 |
| 13 | LT_EN | 监测电压低于阈值报警使能<br>0：禁止<br>1：使能 |

| | | |
|---|---|---|
| 12 | FT_EN | 监测电压从高于阈值到低于阈值变化时报警使能<br>0：禁止<br>1：使能 |
| 11 | RT_EN | 监测电压从低于阈值到高于阈值变化时报警使能<br>0：禁止<br>1：使能 |
| 10 | FILTER_EN | PVD监测信号数字滤波使能<br>0：禁止<br>1：使能 |
| 9:7 | FILTER_TIME[2:0] | PVD监测信号数字滤波时间<br>000：2个时钟周期<br>001：4个时钟周期<br>010：8个时钟周期<br>011：16个时钟周期<br>100：32个时钟周期<br>101：64个时钟周期<br>110：256个时钟周期<br>111：1024个时钟周期 |
| 6:4 | 保留 | 写入无效 |
| 3:1 | PVD_LEVEL[2:0] | PVD阈值等级<br>000：level0上升/下降（2.3/2.2V）<br>001：level1上升/下降（2.7/2.6V）<br>010：level2上升/下降（3.1/3.0V）<br>011：level3上升/下降（3.5/3.4V）<br>100：level4上升/下降（3.9/3.8V）<br>101：level5上升/下降（4.2/4.1V）<br>110：level6上升/下降（4.6/4.5V）<br>111：level7上升/下降（4.7/4.6V） |
| 0 | PVD_EN | PVD监测使能<br>0：禁止<br>1：使能 |

### 5.6.3 电源管理状态寄存器（PMU_SR）

偏移地址：0x14

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | PVD_INTF | PVD_STATUS | Res. | | | | | | | | | | |
| | | | r | r | | | | | | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:13 | 保留 | 写入无效 |
| 12 | PVD_INTF | PVD报警中断状态标志<br>当PVD_ACT为0且发生PVD报警事件时，该位置1；向PVD_INTF_CLR（PMU_CLR寄存器）写1可清0该位。 |
| 11 | PVD_STATUS | PVD当前的监测状态（如果滤波使能，该状态表示滤波后的结果）<br>0：被监测的信号电压高于PVD的阈值<br>1：被监测的信号电压低于PVD的阈值 |
| 10:0 | 保留 | 写入无效 |

### 5.6.4 电源管理状态清除寄存器（PMU_CLR）

偏移地址：0x18

复位值：0x0000 0000

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

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

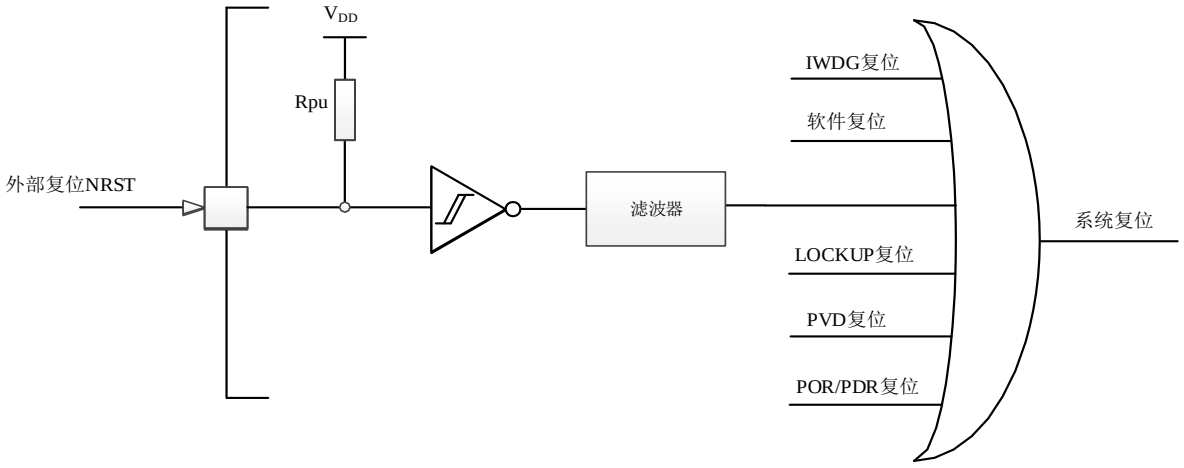
| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:13 | 保留 | 写入无效 |
| 12 | PVD_INTF_CLR | 清除PVD报警中断状态标志<br>写1将PMU_SR寄存器中的PVD_INTF标志清0。 |
| 11:0 | 保留 | 写入无效 |

# 6 复位和时钟控制器（RCC）

## 6.1 复位

芯片支持多种类型的复位，分别为 POR/PDR 复位、NRST 复位、系统复位和 AWON 域复位。

图 6-1 复位电路框图




### 6.1.1 POR/PDR 复位

POR/PDR 会复位所有寄存器。

### 6.1.2 NRST 复位

NRST 复位将复位 $V_{CORE}$ 域寄存器，不包括 RCC_AWCR 寄存器、RCC_CSR2 部分寄存器位、RTC 外设部分寄存器、DBG 相关寄存器、OPAMP1 校准寄存器和 OPAMP2 校准寄存器。

NRST 复位会重新加载出厂校准参数和 Flash 选项字节寄存器，复位期间 RTC、LXTAL 和 RCL 可以正常工作。

PF2 上电默认为 NRST，通过*时钟控制/状态寄存器（RCC_CSR2）*中的 NRST_MODE 位可以将 PF2 引脚配置为 NRST 或 GPIO 功能。

当 PF2 配置为 NRST 功能时，检测到低电平会产生一个 NRST 复位。该复位引脚内置上拉电阻，并集成了毛刺过滤电路；为确保可靠的复位，NRST 引脚低电平保持时间应大于 500μs。详见*图：复位电路框图*。

### 6.1.3 系统复位

系统复位将复位 $V_{CORE}$ 域寄存器，不包括 RCC_RCLCAL 寄存器、RCC_RCHCAL 寄存器、RCC_AWCR 寄存器、RCC_CSR2 部分寄存器位、RTC 外设部分寄存器、PMU_CR2 寄存器、DBG 相关寄存器、OPAMP1 校准寄存器和 OPAMP2 校准寄存器。

系统复位会重新加载 Flash 选项字节寄存器，复位期间 RTC、LXTAL、RCL 和 PVD 可以正常工作。

只要发生以下事件之一，就会产生系统复位：

- 独立看门狗事件（IWDG 复位）；
- 软件复位；
- LOCKUP 复位；
- PVD 电压检测复位。

可通过查询寄存器 RCC_CSR2 中的复位标志确定复位源。

**软件复位**

将 Cortex-M0+内核 SCB_AIRCR 寄存器中的 SYSRESETREQ 位置 1，可产生软件复位。

### 6.1.4 AWON 域复位

有两种方式可产生 AWON 域复位：

- 软件复位，通过将 RCC_AWCR 寄存器中的 AW_RST 位置 1 触发。此方式将复位 RCC_AWCR 部分寄存器位和 RTC 外设；
- POR/PDR 复位会复位 AWON 域。

## 6.2 时钟

时钟单元提供了一系列频率的时钟源，包括：

- RCH：内部高速 48/64/72MHz RC 振荡器时钟；
- RCL：内部低速 32KHz RC 振荡器时钟；
- EXTCLK：外部输入时钟，支持最高 72MHz 时钟输入；

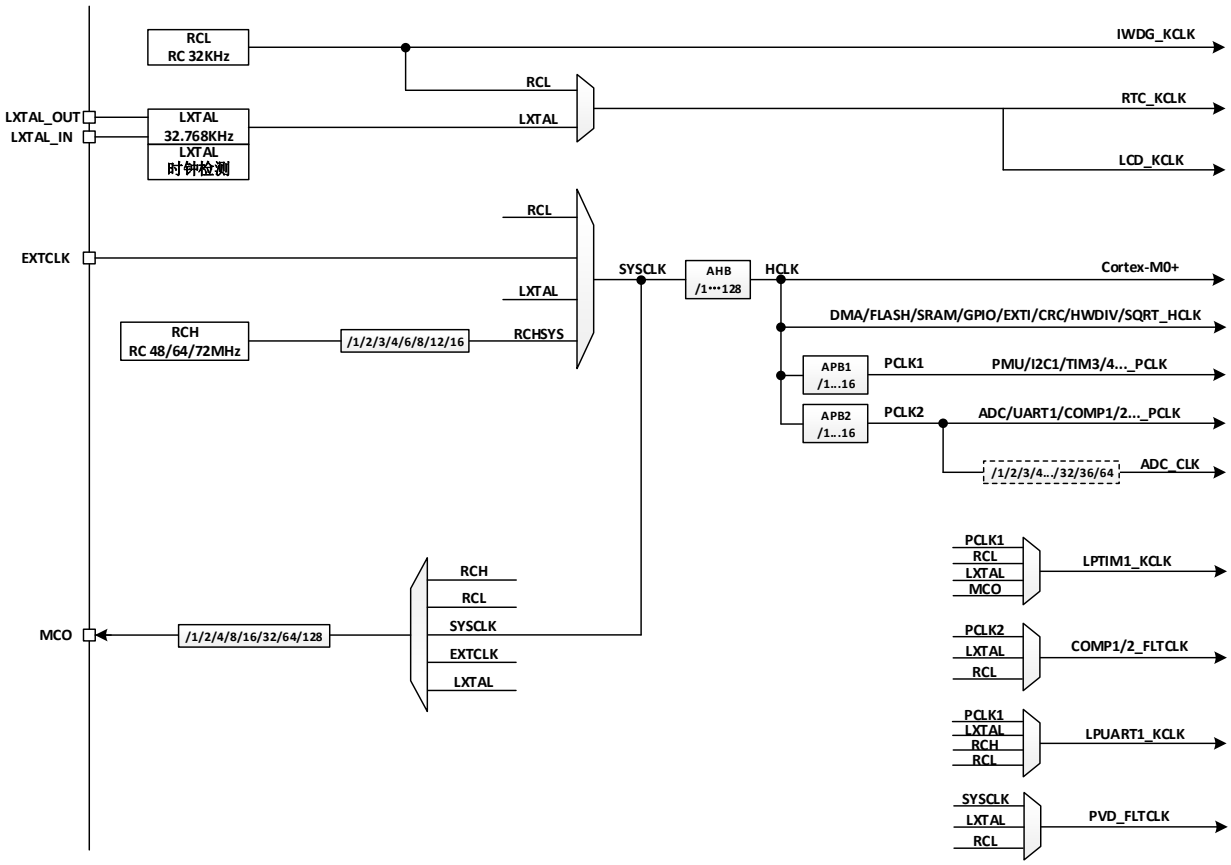
- LXTAL：外部低速时钟，32.768KHz。

时钟输入/输出关系表：

表 6-1 时钟关系表

| 输出时钟 | 输入时钟 | 说明 |
|---|---|---|
| RCHSYS | RCH | RCH 的 1/2/3/4/6/8/12/16 分频 |
| SYSCLK | RCHSYS、RCL、EXTCLK、LXTAL | 系统时钟，通过寄存器选择时钟源 |
| HCLK | SYSCLK | AHB 时钟，SYSCLK 的 1/2/4/8/16/32/64/128 分频 |
| PCLK1 | HCLK | APB1 时钟，HCLK 的 1/2/4/8/16 分频 |
| PCLK2 | HCLK | APB2 时钟，HCLK 的 1/2/4/8/16 分频 |

图 6-2 时钟树




### 6.2.1 RCH 时钟

RCH 时钟是内部高速时钟振荡器，其时钟频率支持 48MHz、64MHz 和 72MHz，可通过*时钟控制/状态寄存器(RCC_CSR1)*的 RCHFREQ[1:0]位域进行频率选择。上电或复位后默认 RCH 频率为 64MHz，系统时钟选择 RCHSYS 4MHz。

出厂时 RCH 已校准，校准值保存在 0x1FFF_0208（48MHz）、0x1FFF_020C

（64MHz）和 0x1FFF_0210（72MHz）的位置，具体格式参见*表：出厂校准参数组成*。如果根据实际需求进行更高精度的校准，可通过 RCC_RCHCAL 寄存器对其进行频率调整；校准可通过 TIM4 的捕获功能实现，具体参见*基于 TIM4 的内/外部时钟测量*。

当 RCHSYS 作为系统时钟或 RCH 作为外设工作时钟时，如果 RCH 时钟频率由 48MHz 切换为 72MHz，需要先切换为 64MHz 并等待 700ns 的稳定时间后，再切换至 72MHz，其他频率之间可直接进行切换。

当 RCHSYS 作为系统时钟时，如果时钟需要校准，建议遵循如下流程：

- 配置*时钟控制/状态寄存器（RCC_CSR1）*的 RCHDIV[2:0]位域为 0x7；
- 配置 *RCH 校准寄存器（RCC_RCHCAL）*进行时钟频率校准；
- 配置 RCC_CSR1 寄存器的 RCHDIV[2:0]位域选取合适的 RCHSYS 分频值。

RCH 时钟通过 RCC_CSR1 寄存器中的 RCHON 位来使能或禁止，RCHRDY 标志指示 RCH 是否稳定。

### 6.2.2 RCL 时钟

RCL 时钟频率为 32KHz，可在 Stop 模式下保持运行，为 IWDG、RTC、LCD、LPTIM1 和 LPUART1 提供工作时钟，还可以为 COMP1/2 和 PVD 提供滤波时钟。

RCL 使能的方式有多种：

- 通过 RCC_CSR2 寄存器中的 RCLON 位来使能；
- 使能 IWDG 时，RCL 强制使能；
- RCL 作为 RTC 时钟源且使能 RTC 时，RCL 强制使能。

只有在下面所有条件满足时，才会禁止 RCL：

- RCC_CSR2 寄存器中的 RCLON 为 0；
- 禁止 IWDG；
- RTC 未使用 RCL（RCL 未作为 RTC 的时钟源或者禁止 RTC）。

RCC_CSR2 寄存器中的 RCLRDY 标志指示 RCL 是否稳定。

出厂时 RCL 已校准，校准值保存在 0x1FFF_0214 的位置，具体格式参见*表：出*

*厂校准参数组成*。如果根据实际需求进行更高精度的校准，可通过 RCC_RCLCAL 寄存器对其进行频率调整；校准可通过 TIM4 的捕获功能实现，具体参见*基于 TIM4 的内/外部时钟测量*。

### 6.2.3 EXTCLK 时钟

EXTCLK 时钟仅支持外部输入，时钟频率最大不超过 72MHz。

EXTCLK 可通过 RCC_CSR1 寄存器中的 EXTCLKON 位来使能或禁止。

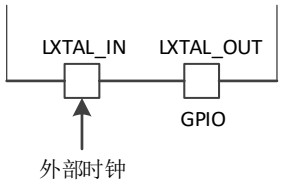
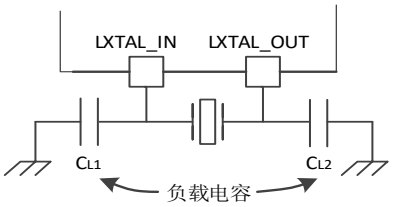
### 6.2.4 LXTAL 时钟

外部低速时钟信号（LXTAL）有 2 种时钟源：

- 外部晶体谐振器/外部陶瓷谐振器（外部晶体模式）；
- 外部时钟源（旁路模式）。

晶体谐振器和负载电容必须尽可能地靠近芯片引脚，以尽量减少输出失真和起振稳定时间。负载电容值必须根据所选晶体谐振器的不同做适当调整。

表 6-2 LXTAL 时钟源

| 时钟源 | 硬件配置 |
|---|---|
| 外部时钟<br>（旁路模式） |  |
| 外部晶体谐振器/<br>外部陶瓷谐振器<br>（外部晶体模式） |  |

**外部晶体（LXTAL 外部晶体模式）**

32.768 KHz 低速外部晶体，可作为实时时钟（RTC）的时钟源来提供时钟/日历或其它定时功能，具有功耗低的优点。

LXTAL 通过 RCC_AWCR 寄存器中的 LXTALON 位来使能或禁止。

LXTAL 时钟的稳定时间可通过 RCC_AWCR 寄存器中的 LXTAL_STAB_SEL[1:0] 位域来配置，仅在 LXTALON 位为 0 时，该位域才可修改。

RCC_AWCR 寄存器的 LXTALRDY 标志指示 LXTAL 时钟是否稳定。

使用 RCC_AWCR 寄存器中的 LXTAL_DRV[2:0]位域，可更改起振电路的驱动能力，以实现启动时间和低功耗之间的最佳平衡，增强对不同晶体的兼容性。

下表为 LXTAL 时钟配置：

表 6-3 LXTAL 时钟配置

| LXTALON | LXTALBYP | 状态 |
|---|---|---|
| 0 | x | 禁止 LXTAL |
| 1 | 0 | 使能 LXTAL 外部晶体模式 |
| 1 | 1 | 使能 LXTAL 旁路模式 |

**外部时钟源（LXTAL 旁路模式）**

在此模式下，必须提供外部时钟源。最高频率不超过 1MHz。

RCC_AWCR 寄存器的 LXTALON 和 LXTALBYP 位置 1 可选择此模式。必须使用占空比约为 40%~60%的外部时钟信号（方波、正弦波或三角波）来驱动 LXTAL_IN 引脚，LXTAL_OUT 引脚可用作 GPIO。

### 6.2.5 系统时钟

可以使用下列不同的时钟源来驱动系统时钟（SYSCLK）：

- RCHSYS
- RCL
- EXTCLK
- LXTAL

系统时钟最高频率为 72MHz。系统复位后，选择 RCHSYS（4MHz）作为系统时钟。

当时钟切换时，如果选择尚未开启或稳定的时钟源，则切换在该时钟源开启及稳定后才会进行。RCC_CFG 寄存器中的 SYSWS[1:0]位域用于指示当前的系统时钟源，作为系统时钟的时钟源无法被关闭。

### 6.2.6 LXTAL CSS

将 RCC_AWCR 寄存器中的 LXTAL_CSSON 位置 1 来使能 LXTAL CSS。

LXTAL CSS 使能后，在 LXTALRDY 和 RCLRDY 均置 1 后，该功能才能正常检测。

LXTAL CSS 可工作在所有工作模式，NRST 复位和系统复位过程中 LXTAL CSS 功能不受影响，AWON 域复位可以将 LXTAL CSS 功能关闭。

如果 LXTAL 时钟发生故障：

- RCC_AWCR 寄存器中的 LXTAL_CSSD 置 1，将 LXTAL_CSSON 清 0 或通过 AWON 域复位可清 0 LXTAL_CSSD 标志；
- RCC_ISR 寄存器中的 LXTAL_CSSF 置 1，如果 RCC_IER 寄存器中的 LXTAL_CSSIE 位置 1，则会触发 LXTAL CSS 中断，将 RCC_ICR 寄存器中的 LXTAL_CSSC 位置 1 可清除 LXTAL_CSSF；
- 当 LXTAL 作为系统时钟时，系统时钟自动切换为 RCL；
- 当 LXTAL 作为 RTC 时钟源时，建议软件更改 RTC 时钟源，或者采取必要措施来确保应用的安全。

当使能 LXTAL CSS 时，LXTAL 的频率必须高于 30KHz，以免发生 CSS 误报。

LXTAL CSS 使能后，可作为 Stop 模式的唤醒源。

### 6.2.7 外设异步时钟选择

所有外设时钟均由其总线时钟（HCLK、PCLK1 和 PCLK2）提供，但有些外设的内核时钟可单独配置。

表 6-4 外设异步时钟来源

| 模块 | 异步时钟来源 |
|---|---|
| IWDG | RCL |
| LPUART1 | PCLK、LXTAL、RCH、RCL |
| LPTIM1 | PCLK、RCL、LXTAL、MCO |
| RTC | RCL、LXTAL |
| LCD | RCL、LXTAL |
| COMP1/2（滤波时钟） | PCLK、RCL、LXTAL |
| PVD（滤波时钟） | SYSCLK、RCL、LXTAL |

### 6.2.8 IWDG 时钟

IWDG 可通过软件使能，使能 IWDG 时 RCL 将被强制开启且不能被关闭。当

RCL 使能并稳定后，将为 IWDG 提供工作时钟。

### 6.2.9 RTC 时钟

RTC 时钟源通过 RCC_AWCR 寄存器中的 RTCSEL[1:0]位域进行选择：

- LXTAL 外部低速时钟；
- RCL 内部低速 RC 振荡器时钟；

RTC 时钟配置流程：

1) 配置 RTCSEL[1:0]位域（RCC_AWCR 寄存器），选择 RTC_KCLK 时钟源；
2) 将 RTC_EN 位（RCC_AWCR 寄存器）置 1 使能 RTC_KCLK；
3) 将 RTCAPBEN 位（RCC_APB1EN 寄存器）置 1 使能 RTC 的 PCLK。

RTC 可在 Stop 模式下工作，并可用作 Stop 模式的唤醒源，RTC 在系统复位时可正常工作。

### 6.2.10 时钟输出功能

可选择以下时钟之一通过 MCO 引脚输出到外部：

- RCH
- RCL
- SYSCLK
- EXTCLK
- LXTAL

由 RCC_CFG 寄存器中的 MCO_SEL[2:0]位域选择输出的时钟，所选时钟可以通过 MCO_PRE[2:0]位域进行分频。

Stop 模式下，软件开启 RCL 或 LXTAL 时可输出 RCL 或 LXTAL。

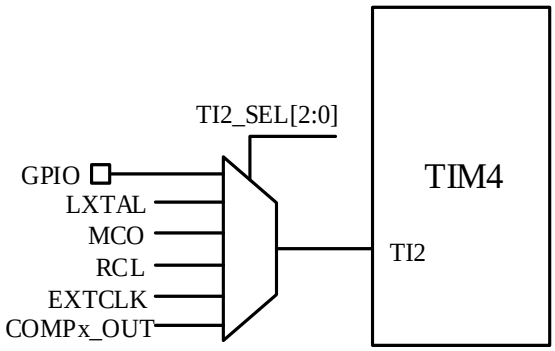
### 6.2.11 基于 TIM4 的内/外部时钟测量

所有时钟源的频率都可通过 TIM4 通道 2 的输入捕获功能进行测量，配置 TIM4 的捕获预分频器可提高测量精度，预分频器值越大，测量精度越高。

通过配置 *TIM4 定时器输入选择寄存器（TIM4_TISEL）* 的 TI2_SEL[2:0]位域来选择不同的输入捕获源：

- GPIO（TIM4_CH2）
- LXTAL
- MCO
- RCL
- EXTCLK
- COMPx_OUT

图 6-3　TIM4 在捕获模式下的频率测量




## 6.3　低功耗模式

- 可通过软件禁止 AHB 和 APB 未使用的外设时钟；
- Sleep 模式下，仅关闭 CPU 时钟，其他时钟与 Run 模式一致；
- Stop 模式下，将关闭 RCH 和 EXTCLK 时钟，RCL 和 LXTAL 不受影响；若 RCH_AWON 配置为 1，Stop 模式 RCH 时钟仍处于使能状态；
- 当 LPUART1 选择 RCH 或 LXTAL 或 RCL（需使能 LXTAL 或 RCL）作为时钟源时，在 Stop 模式下可正常工作；
- 如果进入 Stop 模式前将 *DBG 控制寄存器（DBG_CR）*中的 DBG_STOP 位置 1，那么在 Stop 模式下 RCH 将被开启并为系统提供时钟，以实现 Stop 模式的调试功能。

## 6.4 RCC 寄存器

RCC 寄存器支持 32 位访问。

表 6-5 RCC 基地址

| 外设 | 基地址 |
|---|---|
| RCC | 0x4002 1000 |

### 6.4.1 时钟控制/状态寄存器 1（RCC_CSR1）

偏移地址：0x00

复位值：0x0000 7D00

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RCHFREQ[1:0] | | RCHDIV[2:0] | | | RCH RDY | RCH_ AWON | RCH ON | Res. | | | | | | | |
| rw | rw | rw | rw | rw | r | rw | rw | | | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:17 | 保留 | 写入无效 |
| 16 | EXTCLKON | EXTCLK时钟使能<br>在进入Stop模式时该位硬件清0。<br>0：禁止<br>1：使能 |
| 15:14 | RCHFREQ[1:0] | RCH时钟输出频率选择<br>当切换RCH频率时，RCC_RCHCAL寄存器会自动切换到目标频率的校准值。<br>00：RCH输出频率48MHz<br>01：RCH输出频率64MHz<br>10：保留（写入无效）<br>11：RCH输出频率72MHz |
| 13:11 | RCHDIV[2:0] | RCH时钟分频因子<br>000：RCH时钟1分频作为RCHSYS时钟 |

001：RCH时钟2分频作为RCHSYS时钟
010：RCH时钟3分频作为RCHSYS时钟
011：RCH时钟4分频作为RCHSYS时钟
100：RCH时钟6分频作为RCHSYS时钟
101：RCH时钟8分频作为RCHSYS时钟
110：RCH时钟12分频作为RCHSYS时钟
111：RCH时钟16分频作为RCHSYS时钟

| | | |
|---|---|---|
| 10 | RCHRDY | RCH时钟稳定标志<br>当配置RCHON为0时，RCH_RDY将自动清0。<br>0：RCH时钟未稳定<br>1：RCH时钟已稳定 |
| 9 | RCH_AWON | 始终为外设使能RCH<br>在Stop模式下强制开启RCH，为LPUART1提供时钟，避免由于RCH启动消耗时间而降低通信速率。该位不受RCHON位影响。<br>0：对RCH时钟无影响<br>1：在Stop模式下保持RCH使能 |
| 8 | RCHON | RCH时钟使能<br>进入Stop模式时该位硬件自动清0。<br>0：禁止<br>1：使能 |
| 7:0 | 保留 | 写入无效 |

### 6.4.2 时钟配置寄存器（RCC_CFG）

偏移地址：0x08

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | MCOPRE[2:0] | | | Res. | MCOSEL[2:0] | | | Res. | | | | | P2PRE[2:0] | | |
| | rw | rw | rw | | rw | rw | rw | | | | | | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | P1PRE[2:0] | | | Res. | HPRE[2:0] | | | Res. | | | SYSWS[1:0] | | Res. | SYSW[1:0] | |
| | rw | rw | rw | | rw | rw | rw | | | | r | r | | rw | rw |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 31 | 保留 | 写入无效 |
| 30:28 | MCOPRE[2:0] | 时钟输出预分频<br>000：1分频<br>001：2分频<br>010：4分频<br>011：8分频<br>100：16分频<br>101：32分频<br>110：64分频<br>111：128分频 |
| 27 | 保留 | 写入无效 |
| 26:24 | MCOSEL[2:0] | 时钟输出源选择<br>000：禁止MCO输出<br>001：选择SYSCLK<br>011：选择RCH<br>100：选择EXTCLK<br>110：选择RCL<br>111：选择LXTAL<br>其他：保留（禁止MCO输出）<br>*注意：使能或切换MCO时钟时，时钟输出会存在不完整的周期。* |
| 23:19 | 保留 | 写入无效 |
| 18:16 | P2PRE[2:0] | APB2预分频器<br>用于选择APB2时钟（PCLK2）<br>0xx：HCLK 1分频<br>100：HCLK 2分频<br>101：HCLK 4分频<br>110：HCLK 8分频<br>111：HCLK 16分频 |
| 15 | 保留 | 写入无效 |

| | | |
|---|---|---|
| 14:12 | P1PRE[2:0] | APB1预分频器<br>用于选择APB1时钟（PCLK1）<br>0xx：HCLK 1分频<br>100：HCLK 2分频<br>101：HCLK 4分频<br>110：HCLK 8分频<br>111：HCLK 16分频 |
| 11 | 保留 | 写入无效 |
| 10:8 | HPRE[2:0] | AHB预分频器<br>用于选择AHB时钟<br>000：SYSCLK 1分频<br>001：SYSCLK 2分频<br>010：SYSCLK 4分频<br>011：SYSCLK 8分频<br>100：SYSCLK 16分频<br>101：SYSCLK 32分频<br>110：SYSCLK 64分频<br>111：SYSCLK 128分频 |
| 7:5 | 保留 | 写入无效 |
| 4:3 | SYSWS[1:0] | 系统时钟源指示<br>00：RCHSYS<br>01：EXTCLK<br>10：LXTAL<br>11：RCL |
| 2 | 保留 | 写入无效 |
| 1:0 | SYSW[1:0] | 系统时钟源选择<br>00：RCHSYS用作系统时钟<br>01：EXTCLK用作系统时钟<br>10：LXTAL用作系统时钟<br>11：RCL用作系统时钟 |

### 6.4.3 时钟中断使能寄存器（RCC_IER）

偏移地址：0x10

复位值：0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:10 | 保留 | 写入无效 |
| 9 | LXTAL_CSSIE | LXTAL CSS中断使能<br>0：禁止<br>1：使能 |
| 8:0 | 保留 | 写入无效 |

### 6.4.4 时钟中断标志寄存器（RCC_ISR）

偏移地址：0x14

复位值：0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:10 | 保留 | 写入无效 |
| 9 | LXTAL_CSSF | LXTAL CSS中断标志<br>当LXTAL CSS功能生效后，若检测到故障时，该位被置1。将 |

LXTAL_CSSC位置1，可清除该位。

0：未触发LXTAL CSS中断

1：触发LXTAL CSS中断

| | | |
|---|---|---|
| 8:0 | 保留 | 写入无效 |

### 6.4.5 时钟中断清除寄存器（RCC_ICR）

偏移地址：0x18

复位值：0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:10 | 保留 | 写入无效 |
| 9 | LXTAL_CSSC | LXTAL CSS中断标志清除<br>0：无影响<br>1：将LXTAL_CSSF标志清零 |
| 8:0 | 保留 | 写入无效 |

### 6.4.6 GPIO 端口复位寄存器（RCC_IOPRST）

偏移地址：0x1C

复位值：0x0000 0000

*注意：* *GPIO 端口复位软件流程为：相应 bit 位置 1，复位 GPIO 端口；清除该位，使 GPIO 端口处于正常工作模式。*

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

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

| Res. | | | | | | | | | | GPIOF_RST | Res. | | GPIOC_RST | GPIOB_RST | GPIOA_RST |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | rw | | | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:6 | 保留 | 写入无效 |
| 5 | GPIOF_RST | GPIOF复位<br>0：正常工作模式<br>1：复位GPIOF |
| 4:3 | 保留 | 写入无效 |
| 2 | GPIOC_RST | GPIOC复位<br>0：正常工作模式<br>1：复位GPIOC |
| 1 | GPIOB_RST | GPIOB复位<br>0：正常工作模式<br>1：复位GPIOB |
| 0 | GPIOA_RST | GPIOA复位<br>0：正常工作模式<br>1：复位GPIOA |

### 6.4.7 AHB 外设复位寄存器（RCC_AHBRST）

偏移地址：0x20

复位值：0x0000 0000

*注意：* *所有外设复位软件流程为：相应 bit 位置 1，复位外设；清除该位，使外设处于正常工作模式。*

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | SQRT_RST | HWDIV_RST | Res. | | | | | | | |
| | | | | | | rw | rw | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | CRC_RST | Res. | | | | | | | | | | | DMA_RST |
| | | | rw | | | | | | | | | | | | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:26 | 保留 | 写入无效 |
| 25 | SQRT_RST | SQRT复位<br>0：正常工作模式<br>1：复位SQRT |
| 24 | HWDIV_RST | HWDIV复位<br>0：正常工作模式<br>1：复位HWDIV |
| 23:13 | 保留 | 写入无效 |
| 12 | CRC_RST | CRC复位<br>0：正常工作模式<br>1：复位CRC |
| 11:1 | 保留 | 写入无效 |
| 0 | DMA_RST | DMA复位<br>0：正常工作模式<br>1：复位DMA |

### 6.4.8 APB1 外设复位寄存器（RCC_APB1RST）

偏移地址：0x24

复位值：0x0000 0000

*注意：* *所有外设复位软件流程为：相应 bit 位置 1，复位外设；清除该位，使其外设处于正常工作模式。*

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LPTIM1_RST | Res. | | | | | | | | | I2C1_RST | LPUART1_RST | Res. | | UART2_RST | Res. |
| rw | | | | | | | | | | rw | rw | | | rw | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | SPI2_RST | Res. | | | | | LCD_RST | Res. | | | | | TIM4_RST | TIM3_RST | HALLTIM_RST |
| | rw | | | | | | rw | | | | | | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 31 | LPTIM1_RST | LPTIM1复位<br>0：正常工作模式<br>1：复位LPTIM1 |
| 30:22 | 保留 | 写入无效 |
| 21 | I2C1_RST | I2C1复位<br>0：正常工作模式<br>1：复位I2C1 |
| 20 | LPUART1_RST | LPUART1复位<br>0：正常工作模式<br>1：复位LPUART1 |
| 19:18 | 保留 | 写入无效 |
| 17 | UART2_RST | UART2复位<br>0：正常工作模式<br>1：复位UART2 |
| 16:15 | 保留 | 写入无效 |
| 14 | SPI2_RST | SPI2复位<br>0：正常工作模式<br>1：复位SPI2 |
| 13:9 | 保留 | 写入无效 |
| 8 | LCD_RST | LCD复位<br>0：正常工作模式<br>1：复位LCD |
| 7:3 | 保留 | 写入无效 |
| 2 | TIM4_RST | TIM4复位<br>0：正常工作模式 |

| | | |
|---|---|---|
| | | 1：复位TIM4 |
| 1 | TIM3_RST | TIM3复位<br>0：正常工作模式<br>1：复位TIM3 |
| 0 | HALLTIM_RST | HALLTIM复位<br>0：正常工作模式<br>1：复位HALLTIM |

### 6.4.9 APB2 外设复位寄存器（RCC_APB2RST）

偏移地址：0x28

复位值：0x0000 0000

*注意：* *所有外设复位软件流程为：相应 bit 位置 1，复位外设；清除该位，使其外设处于正常工作模式。*

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | DBG_RST | Res. | | | | | | ADC_RST | Res. | | | |
| | | | | rw | | | | | | | rw | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | UART1_RST | Res. | SPI1_RST | TIM1_RST | Res. | | | | | | | | | | COMP_RST |
| | rw | | rw | rw | | | | | | | | | | | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:28 | 保留 | 写入无效 |
| 27 | DBG_RST | DBG寄存器复位<br>0：正常工作模式<br>1：复位DBG寄存器 |
| 26:21 | 保留 | 写入无效 |
| 20 | ADC_RST | ADC复位<br>0：正常工作模式<br>1：复位ADC |

| | | |
|---|---|---|
| 19:15 | 保留 | 写入无效 |
| 14 | UART1_RST | UART1复位<br>0：正常工作模式<br>1：复位UART1 |
| 13 | 保留 | 写入无效 |
| 12 | SPI1_RST | SPI1复位<br>0：正常工作模式<br>1：复位SPI1 |
| 11 | TIM1_RST | TIM1复位<br>0：正常工作模式<br>1：复位TIM1 |
| 10:1 | 保留 | 写入无效 |
| 0 | COMP_RST | COMP1和COMP2复位<br>0：正常工作模式<br>1：复位 COMP1 和 COMP2 |

### 6.4.10 I/O 端口时钟使能寄存器（RCC_IOPEN）

偏移地址：0x2C

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | GPIOFEN | Res. | | GPIOCEN | GPIOBEN | GPIOAEN |
| | | | | | | | | | | rw | | | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:6 | 保留 | 写入无效 |
| 5 | GPIOFEN | GPIOF时钟使能 |

0：禁止

1：使能

| | | |
|---|---|---|
| 4:3 | 保留 | 写入无效 |
| 2 | GPIOCEN | GPIOC时钟使能<br>0：禁止<br>1：使能 |
| 1 | GPIOBEN | GPIOB时钟使能<br>0：禁止<br>1：使能 |
| 0 | GPIOAEN | GPIOA时钟使能<br>0：禁止<br>1：使能 |

### 6.4.11 AHB 外设时钟使能寄存器（RCC_AHBEN）

偏移地址：0x30

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | SQRTEN | HWDIVEN | Res. | | | | | | | |
| | | | | | | rw | rw | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | CRCEN | Res. | | | | | | | | | | | DMAEN |
| | | | rw | | | | | | | | | | | | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:26 | 保留 | 写入无效 |
| 25 | SQRTEN | SQRT时钟使能<br>0：禁止<br>1：使能 |
| 24 | HWDIVEN | HWDIV时钟使能<br>0：禁止 |

| | | |
|---|---|---|
| | | 1：使能 |
| 23:13 | 保留 | 写入无效 |
| 12 | CRCEN | CRC时钟使能<br>0：禁止<br>1：使能 |
| 11:1 | 保留 | 写入无效 |
| 0 | DMAEN | DMA时钟使能<br>0：禁止<br>1：使能 |

### 6.4.12 APB1 外设时钟使能寄存器（RCC_APB1EN）

偏移地址：0x34

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LPTIM1 EN | Res. | | PMU EN | Res. | | | | | | I2C1 EN | LPUART1 EN | Res. | | UART2 EN | Res. |
| rw | | | rw | | | | | | | rw | rw | | | rw | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | SPI2 EN | Res. | | | RTCAPB EN | Res. | LCD EN | Res. | | | | | TIM4 EN | TIM3 EN | HALLTIM EN |
| | rw | | | | rw | | rw | | | | | | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31 | LPTIM1EN | LPTIM1时钟使能<br>0：禁止<br>1：使能 |
| 30:29 | 保留 | 写入无效 |
| 28 | PMUEN | PMU寄存器时钟使能<br>0：禁止<br>1：使能<br>*注意：PMUEN仅作用于PMU的寄存器时钟，PVD的滤波时钟不受该位影响。* |

| | | |
|---|---|---|
| 27:22 | 保留 | 写入无效 |
| 21 | I2C1EN | I2C1时钟使能<br>0：禁止<br>1：使能 |
| 20 | LPUART1EN | LPUART1时钟使能<br>0：禁止<br>1：使能 |
| 19:18 | 保留 | 写入无效 |
| 17 | UART2EN | UART2时钟使能<br>0：禁止<br>1：使能 |
| 16:15 | 保留 | 写入无效 |
| 14 | SPI2EN | SPI2时钟使能<br>0：禁止<br>1：使能 |
| 13:11 | 保留 | 写入无效 |
| 10 | RTCAPBEN | RTC APB时钟使能<br>0：禁止<br>1：使能 |
| 9 | 保留 | 写入无效 |
| 8 | LCDEN | LCD时钟使能<br>0：禁止<br>1：使能 |
| 7:3 | 保留 | 写入无效 |

| | | |
|---|---|---|
| 2 | TIM4EN | TIM4时钟使能<br>0：禁止<br>1：使能 |
| 1 | TIM3EN | TIM3时钟使能<br>0：禁止<br>1：使能 |
| 0 | HALLTIMEN | HALLTIM时钟使能<br>0：禁止<br>1：使能 |

### 6.4.13 APB2 外设时钟使能寄存器（RCC_APB2EN）

偏移地址：0x38

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | DBG EN | Res. | | | | | | ADC EN | Res. | | | |
| | | | | rw | | | | | | | rw | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | UART1 EN | Res. | SPI1 EN | TIM1 EN | Res. | | | | | | | | | | COMP EN |
| | rw | | rw | rw | | | | | | | | | | | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:28 | 保留 | 写入无效 |
| 27 | DBGEN | DBG寄存器时钟使能<br>0：禁止<br>1：使能 |
| 26:21 | 保留 | 写入无效 |
| 20 | ADCEN | ADC时钟使能<br>0：禁止<br>1：使能 |
| 19:15 | 保留 | 写入无效 |

| | | |
|---|---|---|
| 14 | UART1EN | UART1时钟使能<br>0：禁止<br>1：使能 |
| 13 | 保留 | 写入无效 |
| 12 | SPI1EN | SPI1时钟使能<br>0：禁止<br>1：使能 |
| 11 | TIM1EN | TIM1时钟使能<br>0：禁止<br>1：使能 |
| 10:1 | 保留 | 写入无效 |
| 0 | COMPEN | COMP1和COMP2时钟使能<br>0：禁止<br>1：使能 |

### 6.4.14 外设异步时钟配置寄存器（RCC_CLKSEL）

偏移地址：0x3C

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | LCD_SEL | Res. | | | | LPTIM1_SEL [1:0] | | Res. | |
| | | | | | | | rw | | | | | rw | rw | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | LPUART1_SEL [1:0] | | Res. | | | | | | | | | |
| | | | | rw | rw | | | | | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:25 | 保留 | 写入无效 |
| 24 | LCD_SEL | LCD时钟源选择<br>0：RCL |

| | | |
|---|---|---|
| | | 1：LXTAL |
| 23:20 | 保留 | 写入无效 |
| 19:18 | LPTIM1_SEL[1:0] | LPTIM1时钟源选择<br>00：PCLK<br>01：RCL<br>10：MCO<br>11：LXTAL |
| 17:12 | 保留 | 写入无效 |
| 11:10 | LPUART1_SEL[1:0] | LPUART1时钟源选择<br>00：PCLK<br>01：RCL<br>10：RCH<br>11：LXTAL |
| 9:0 | 保留 | 写入无效 |

### 6.4.15 AWON 域控制寄存器（RCC_AWCR）

偏移地址：0x40

复位值：0x0020 0008

*注意：* *POR/PDR 复位可复位该寄存器，AWON 域复位可复位除 bit16 外的所有寄存器。复位后，这些位都被写保护，因此必须将电源管理控制寄存器 1（PMU_CR1）的 VAON_WEN 位置 1 才能被修改。*

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RTC EN | Res. | | | | | RTCSEL[1:0] | | LXTAL BYP | LXTAL_CSSD | LXTAL_CSSON | LXTAL_DRV[2:0] | | | LXTAL RDY | LXTAL ON |
| rw | | | | | | rw | rw | rw | r | rw | rw | rw | rw | r | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:22 | 保留 | 写入无效 |

| | | |
|---|---|---|
| 21:20 | LXTAL_STAB_SEL[1:0] | 等待 LXTAL 时钟稳定的周期数<br>00：256 个时钟周期<br>01：1024 个时钟周期<br>10：4096 个时钟周期<br>11：16384 个时钟周期<br>*注意：当LXTALON位为0时，该位域才可修改。* |
| 19:17 | 保留 | 写入无效 |
| 16 | AW_RST | 软件复位AWON域<br>0：正常工作模式<br>1：复位AWON域 |
| 15 | RTCEN | RTC KCLK时钟使能<br>0：禁止<br>1：使能 |
| 14:10 | 保留 | 写入无效 |
| 9:8 | RTCSEL[1:0] | RTC时钟源选择<br>00：无时钟<br>01：LXTAL时钟<br>1x：RCL时钟 |
| 7 | LXTALBYP | LXTAL旁路<br>该位仅在LXTALON和LXTALRDY均为0时，才可更改。<br>0：LXTAL外部晶体模式<br>1：LXTAL旁路模式 |
| 6 | LXTAL_CSSD | LXTAL CSS故障检测状态指示<br>0：未检测到故障<br>1：检测到故障<br>*注意：当LXTAL_CSSON位清0时，该位也被清0。* |
| 5 | LXTAL_CSSON | LXTAL CSS使能<br>0：禁止 |

| | | |
|---|---|---|
| | | 1：使能<br>*注意：该位一旦使能则无法更改，只有检测到LXTAL发生故障时，才可清0该位。* |
| 4:2 | LXTAL_DRV[2:0] | LXTAL驱动能力档位<br>000：最弱<br>⋮<br>111：最强 |
| 1 | LXTALRDY | LXTAL时钟稳定标志<br>当配置LXTALON为0时，LXTALRDY将自动清0。<br>0：LXTAL未稳定<br>1：LXTAL已稳定 |
| 0 | LXTALON | LXTAL使能<br>0：禁止<br>1：使能 |

### 6.4.16 时钟控制/状态寄存器 2（RCC_CSR2）

偏移地址：0x44

复位值：0x0X00 0000

*注意：　POR/PDR 复位可复位该寄存器；NRST 复位和系统复位可复位 bit8 和 bit16。*

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | IWDG_RSTF | SW_RSTF | POR_RSTF | NRST_RSTF | Res. | LOCKUP_RSTF | PVD_RSTF | Res. | | | | | | RMVF |
| | | r | r | r | r | | r | r | | | | | | | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | NRST_MODE | Res. | | | LOCKUP_RSTEN | Res. | | | | | | RCLRDY | RCLON |
| | | | rw | | | | rw | | | | | | | r | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:30 | 保留 | 写入无效 |
| 29 | IWDG_RSTF | 独立看门狗复位标志<br>独立看门狗复位时，该位被置1。将RMVF位置1可清除该位。<br>0：未产生独立看门狗复位<br>1：产生独立看门狗复位 |

| | | |
|---|---|---|
| 28 | SW_RSTF | 软件复位标志<br>软件复位时，该位被置1。将RMVF位置1可清除该位。<br>0：未产生软件复位<br>1：产生软件复位 |
| 27 | POR_RSTF | POR/PDR复位标志<br>POR/PDR复位时，该位被置1。将RMVF位置1可清除该位。<br>0：未产生POR/PDR复位<br>1：产生POR/PDR复位 |
| 26 | NRST_RSTF | NRST引脚复位标志<br>NRST引脚复位时，该位被置1。将RMVF位置1可清除该位。<br>0：未产生来自NRST引脚的复位<br>1：产生来自NRST引脚的复位 |
| 25 | 保留 | 写入无效 |
| 24 | LOCKUP_RSTF | LOCKUP复位标志<br>当LOCKUP功能使能（LOCKUP_RSTEN置1），且产生LOCKUP故障时，该位被置1。将RMVF位置1可清除该位。<br>0：未产生LOCKUP的复位<br>1：产生LOCKUP的复位 |
| 23 | PVD_RSTF | PVD复位标志<br>PVD复位时，该位被置1。将RMVF位置1可清除该位。<br>0：未产生PVD复位<br>1：产生PVD复位 |
| 22:17 | 保留 | 写入无效 |
| 16 | RMVF | 清除复位标志<br>由软件置1，用于将复位标志清0，同时该位也被清0。<br>0：无影响<br>1：清除复位标志 |
| 15:13 | 保留 | 写入无效 |

| | | |
|---|---|---|
| 12 | NRST_MODE | NRST端口功能配置<br>0：PF2引脚作为NRST端口<br>1：PF2引脚作为GPIO端口 |
| 11:9 | 保留 | 写入无效 |
| 8 | LOCKUP_RSTEN | LOCKUP复位使能<br>0：禁止<br>1：使能 |
| 7:2 | 保留 | 写入无效 |
| 1 | RCLRDY | RCL稳定标志<br>当RCLON位被清0后，RCLRDY将自动清0。<br>0：RCL未稳定<br>1：RCL已稳定 |
| 0 | RCLON | RCL使能<br>0：禁止<br>1：使能 |

### 6.4.17 RCL 校准寄存器（RCC_RCLCAL）

偏移地址：0x50

复位值：0x0000 00XX

*注意：　仅 POR/PDR 和 NRST 复位时，复位该寄存器。*

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:5 | 保留 | 写入无效 |

| | | |
|---|---|---|
| 4:0 | RCL_CAL[4:0] | RCL 时钟校准值 |

### 6.4.18 RCH 校准寄存器（RCC_RCHCAL）

偏移地址：0x54

复位值：0x0000 0XXX

*注意：　仅 POR/PDR 和 NRST 复位时，复位该寄存器。*

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | RCH_CAL_ COARSE[1:0] | | Res. | | RCH_CAL_FINE[5:0] | | | | | |
| | | | | | | rw | rw | | | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:10 | 保留 | 写入无效 |
| 9:8 | RCH_CAL_COARSE[1:0] | RCH 时钟粗校准值<br>每增减 1 档，RCH 48MHz、64MHz 和 72MHz 档位的输出频率分别同步增减约 14MHz、19MHz 和 21MHz。 |
| 7:6 | 保留 | 写入无效 |
| 5:0 | RCH_CAL_FINE[5:0] | RCH 时钟细校准值<br>每增减 1 档，RCH 48MHz、64MHz 和 72MHz 档位的输出频率分别同步增减约 0.24MHz、0.3MHz 和 0.33MHz。 |

# 7 通用输入/输出接口（GPIO）

## 7.1 简介

每组通用 I/O 端口包括 3 个 32 位配置寄存器（GPIOx_MODE、GPIOx_OTYPE、GPIOx_PUPD），2 个 32 位数据寄存器（GPIOx_IDR、GPIOx_ODR），1 个 32 位置位/复位寄存器（GPIOx_BSR）、1 个 32 位复位寄存器（GPIOx_BR）；每个 GPIO 包括 2 个 32 位复用功能选择寄存器（GPIOx_AFH、GPIOx_AFL）。

## 7.2 GPIO 主要特性

- GPIO 模式
  - 输入
  - 输出
  - 复用功能
  - 模拟模式
- 上拉/下拉单独可配
- 推挽/开漏单独可配
- 引脚复用灵活，允许将 I/O 引脚用作 GPIO 或外设复用功能
- 置位和复位寄存器（GPIOx_BSR），对 GPIOx_ODR 具有按位写权限

## 7.3 GPIO 功能描述

软件可在配置寄存器中对每个 GPIO 的模式进行独立配置，多种可选的模式组合如下：

- 输入浮空
- 输入上拉
- 输入下拉
- 具有上拉或下拉功能的开漏输出
- 具有上拉或下拉功能的推挽输出

- 具有上拉或下拉功能的复用功能推挽
- 具有上拉或下拉功能的复用功能开漏
- 模拟功能

下图描述了 I/O 电路的基本结构，GPIO 端口配置表给出了可能的 I/O 配置方案。

图 7-1 I/O 基本结构

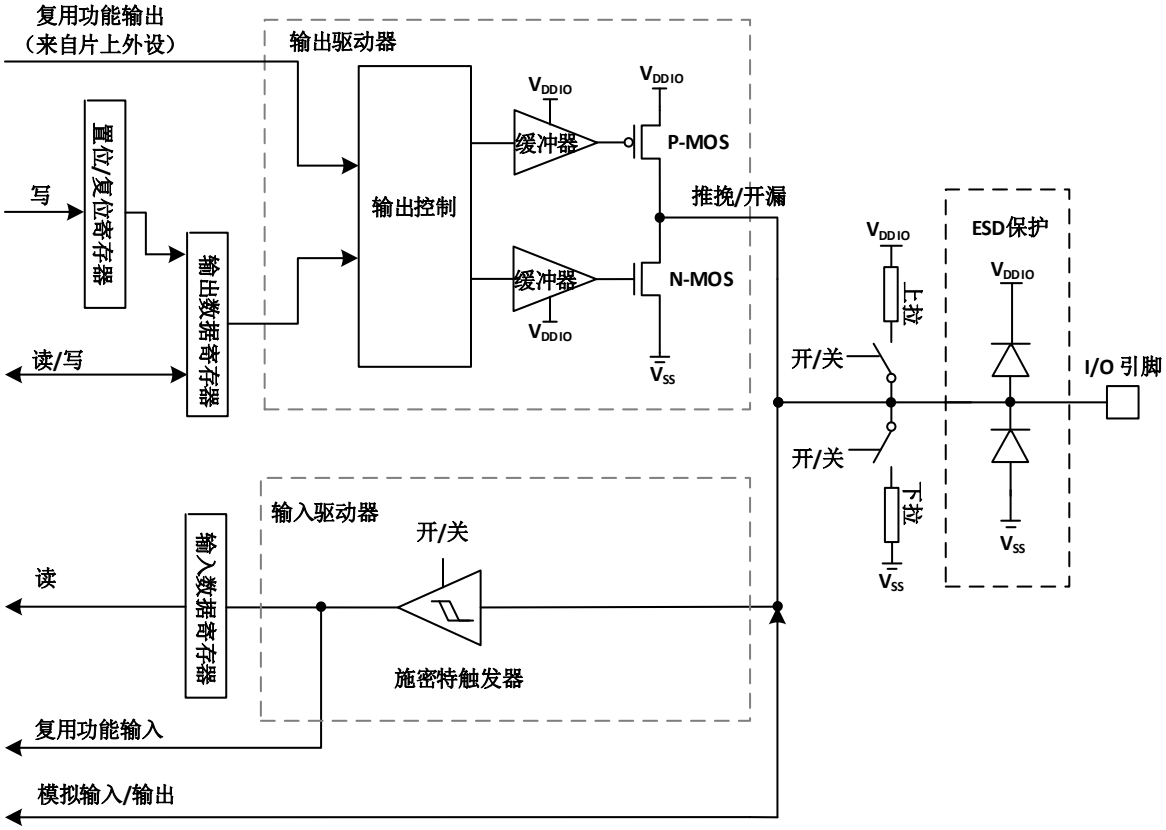



表 7-1 GPIO 端口配置表 (1)

| I/O 端模式组合 | | I/O 端口配置 | | | |
|---|---|---|---|---|---|
| | | GPIO 模式<br>MODE[1:0] | 输出类型<br>OTYPE | 上拉/下拉<br>PUPD[1:0] | |
| 输入 | 浮空 | 00 | - | 0 | 0 |
| | 上拉 | | - | 0 | 1 |
| | 下拉 | | - | 1 | 0 |
| 输出 | 推挽 | 01 | 0 | 0 | 0 |
| | 推挽+上拉 | | 0 | 0 | 1 |
| | 推挽+下拉 | | 0 | 1 | 0 |
| | 开漏 | | 1 | 0 | 0 |
| | 开漏+上拉 | | 1 | 0 | 1 |

| I/O 端模式组合 | | I/O 端口配置 | | | |
|---|---|---|---|---|---|
| | | GPIO 模式<br>MODE[1:0] | 输出类型<br>OTYPE | 上拉/下拉<br>PUPD[1:0] | |
| | 开漏+下拉 | | 1 | 1 | 0 |
| 复用 | 推挽 | 10 | 0 | 0 | 0 |
| | 推挽+上拉 | | 0 | 0 | 1 |
| | 推挽+下拉 | | 0 | 1 | 0 |
| | 开漏 | | 1 | 0 | 0 |
| | 开漏+上拉 | | 1 | 0 | 1 |
| | 开漏+下拉 | | 1 | 1 | 0 |
| 模拟 | 输入/输出 | 11 | - | 上拉/下拉禁用 | |
| | | | - | | |
| | | | - | | |

1. “–”配置无效。

### 7.3.1 通用 I/O (GPIO)

复位释放前/后，除 PF2、PF5、PA14、PA13 引脚外，其他引脚均处于模拟模式。

- PF2 引脚
  - POR/PDR 复位释放前，默认作为 NRST 引脚、输入上拉
  - POR/PDR 复位释放后，由 RCC 的*时钟控制/状态寄存器 2（RCC_CSR2）*的 NRST_MODE 位的配置决定

    0：PF2 引脚作为 NRST 引脚、输入上拉

    1：PF2 引脚作为 GPIO
- PF5 引脚
  - POR/PDR 和 NRST 复位释放前，作为 BOOT0 引脚、输入下拉，复位释放后，处于模拟模式
  - 系统复位释放前/后，均为模拟模式
- PA14 引脚
  - 复位释放前/后，均复用为 SWCLK、输入下拉
- PA13 引脚

- 复位释放前/后，均复用为 SWDIO、输入上拉

### 7.3.2 I/O 引脚功能复用器和映射

I/O 引脚功能包括 GPIO、外设复用功能和附加功能。

**GPIO**

在 I/O 模式寄存器 GPIOx_MODE 中，将 I/O 配置为输入或输出。

**复用功能**

每个 I/O 电路结构中包含一个引脚复用器，将各外设模块的复用功能映射到不同的 I/O 引脚上，实现芯片与板载外设或模块通信。这种 I/O 复用架构下，每个 I/O 引脚一次只允许映射一种复用功能，确保了不同外设的复用功能在同一个 I/O 上不会发生应用冲突。

- 芯片复位释放后，复用功能寄存器 GPIOx_AFH/GPIOx_AFL 的值默认选择复用功能 0；
- 复用功能配置。
  - 在 GPIOx_AFH/GPIOx_AFL 寄存器中，将 I/O 映射到所需的复用功能；
  - 通过 GPIOx_OTYPE、GPIOx_PUPD 寄存器，分别选择输出类型、上拉/下拉；
  - 在 GPIOx_MODE 寄存器中将所需 I/O 配置为复用功能。

**附加功能**

- 模拟功能
  - 对于 ADC、COMP、OPAMP，在 GPIOx_MODE 寄存器中将所需 I/O 配置为模拟模式，并在 ADC、COMP、OPAMP 寄存器中使能。
- 其他功能
  - 对于振荡器（LXTAL）和外部时钟（EXTCLK）等其它功能，在 RCC 相关寄存器中配置所需功能，且这些配置优先级高于 GPIO 寄存器的配置，详见*外部振荡器引脚用作 GPIO* 和*外部时钟输入引脚用作 GPIO* 章节。

### 7.3.3 I/O 端口配置寄存器

每个 I/O 端口包括 3 个 32 位配置寄存器（GPIOx_MODE、GPIOx_OTYPE、GPIOx_PUPD）用于配置端口的每个 GPIO。

- GPIO 模式寄存器 GPIOx_MODE，用于配置 I/O 模式；
  - 输入、输出、复用、模拟模式。
- 输出类型寄存器 GPIOx_OTYPE，用于配置输出类型推挽/开漏；
- 上拉/下拉寄存器 GPIOx_PUPD，用于配置上拉/下拉。

### 7.3.4 I/O 端口数据寄存器

每个 I/O 端口有 2 个 32 位数据寄存器：

输入和输出数据寄存器（GPIOx_IDR 和 GPIOx_ODR）。

- GPIOx_ODR 用于存储待输出数据，可进行读/写访问；
- GPIOx_IDR 为只读寄存器，I/O 引脚输入的数据存储到 GPIOx_IDR。

### 7.3.5 I/O 数据位操作

置位/复位寄存器 GPIOx_BSR 允许应用程序对输出数据寄存器 GPIOx_ODR 按位操作，即对输出数据寄存器中的每个位单独执行置位和复位。

- 置位/复位寄存器中 BS[15:0]和 BR[15:0]的 bit(i)与 GPIOx_ODR 寄存器中的 bit(i)对应，分别执行置位操作和复位操作；
- 置位/复位寄存器中的任何位写 0 无效，不会改变输出数据寄存器中对应位的状态；
- 同时尝试对某个位执行置位和复位操作，则置位操作优先。

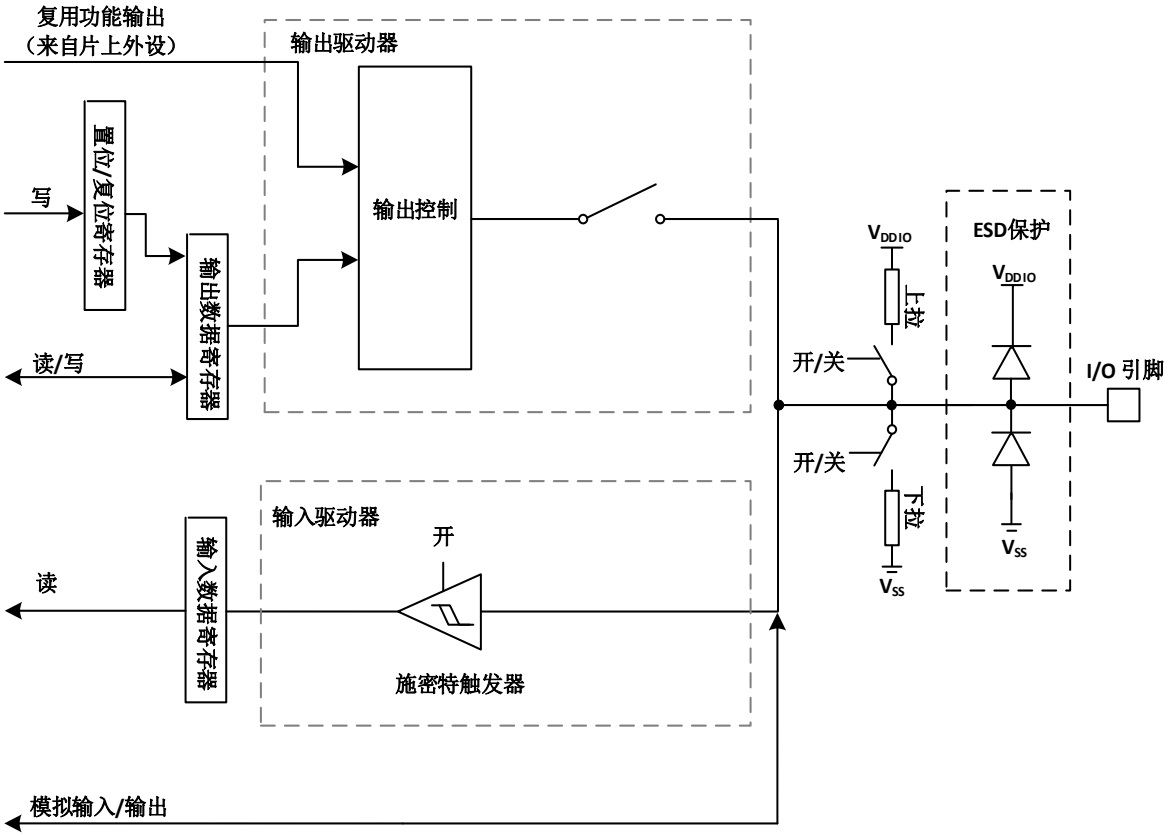
通过写访问 GPIOx_BSR 寄存器修改 1 个或多个 GPIOx_ODR 寄存器中的对应位，无需读 GPIOx_ODR 寄存器后，再次写入，因此修改寄存器值的过程，不会被读取和写入之间产生的中断打断，无需软件关闭中断。

### 7.3.6 外部中断/唤醒

全部 GPIO 端口均具有外部中断功能，且由 EXTI 进行配置和管理，参见*扩展中断和事件控制器（EXTI）*。系统应用存在多个 I/O 中断的情况下，应选择在不同 EXTI 通道上的 I/O，参见 *EXTI I/O 选择寄存器 1* 和 *EXTI I/O 选择寄存器 2*。

### 7.3.7 输入配置

图 7-2 I/O 引脚输入配置（浮空/上拉/下拉）

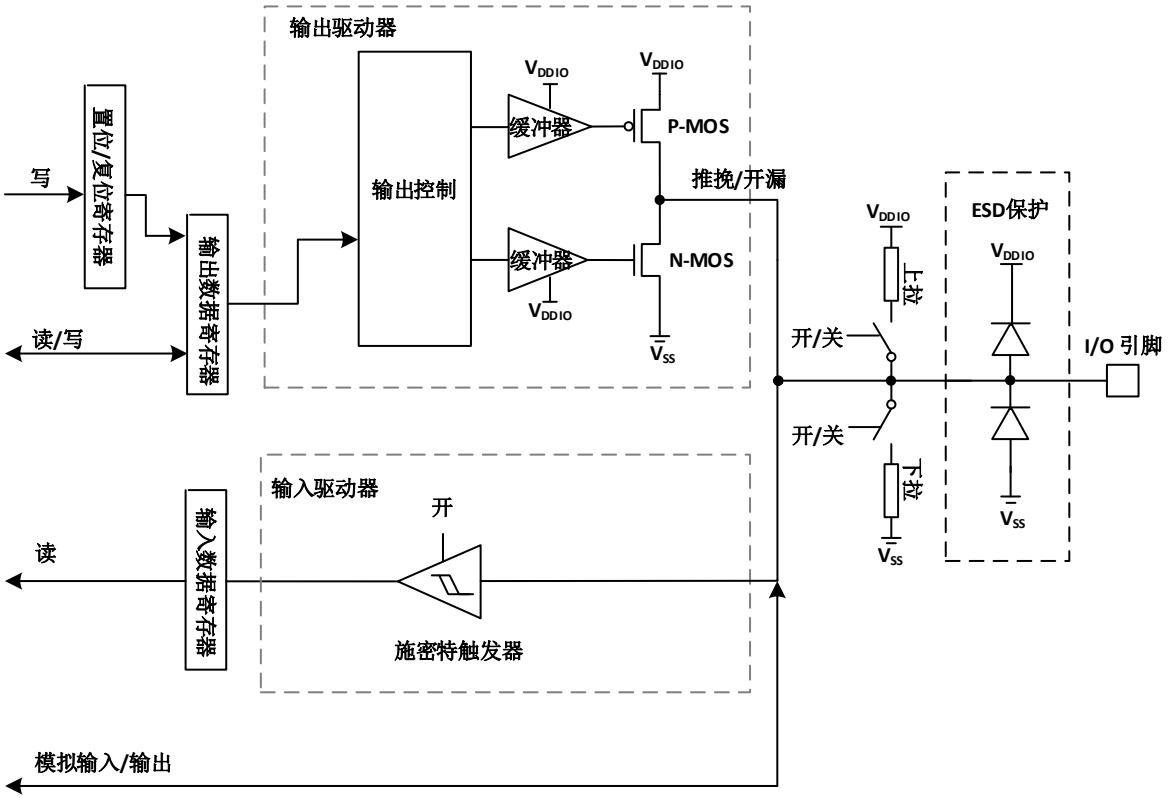



配置 I/O 为输入时，I/O 电路结构示意如上图所示：

- 施密特触发器输入使能；
- 上拉、下拉电阻的选择由 GPIOx_PUPD 寄存器中的值决定；
- 输出缓冲器禁用；
- 读访问输入数据寄存器获取 I/O 状态。

### 7.3.8 输出配置

图 7-3 I/O 引脚输出配置

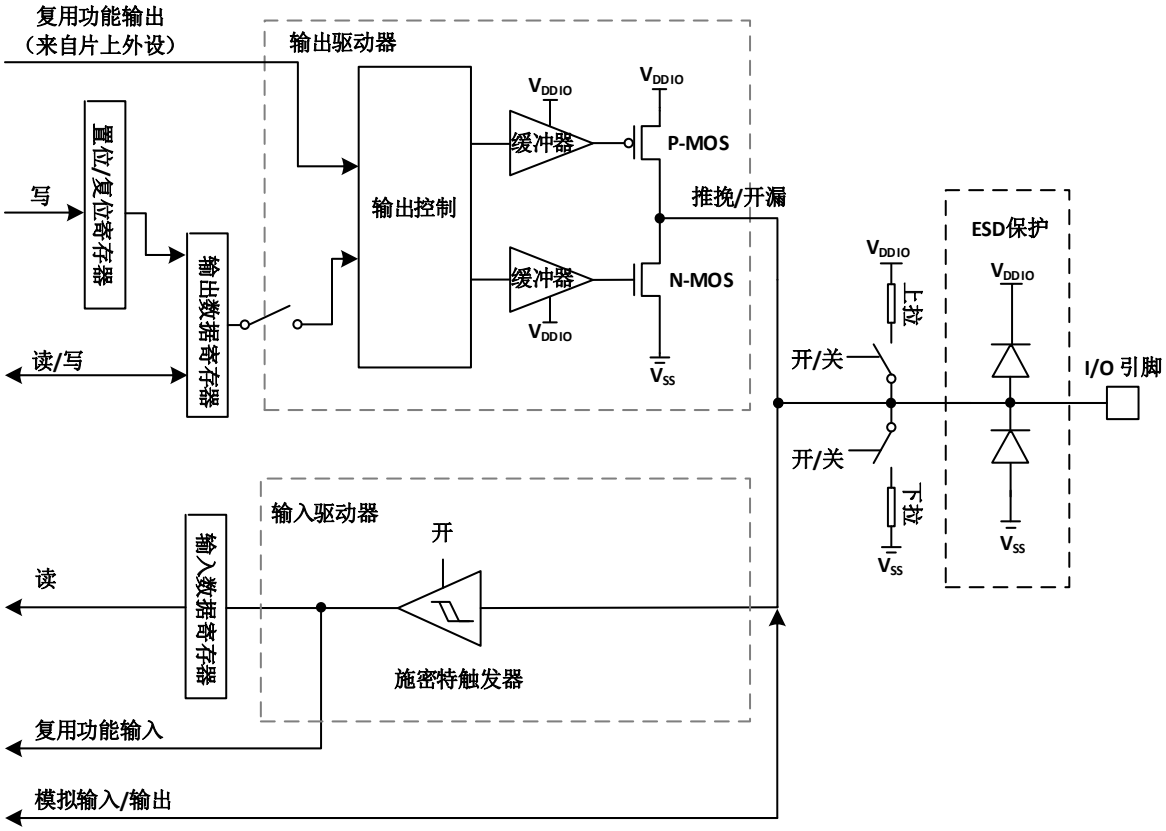



配置 I/O 为输出时，I/O 电路结构示意如上图所示：

- 输出缓冲器使能；
  - 开漏模式：输出数据寄存器写 0，激活 N-MOS 输出低；输出数据寄存器写 1，输出高阻态。
  - 推挽模式：输出数据寄存器写 0，激活 N-MOS 输出低；输出数据寄存器写 1，激活 P-MOS 输出高。
- 软件读访问输出数据寄存器获取最后写入值；
- 施密特触发器输入使能；
- 上拉、下拉电阻的选择由 GPIOx_PUPD 寄存器的值决定；
- 读访问输入数据寄存器获取 I/O 状态。

### 7.3.9 复用功能配置

图 7-4 I/O 引脚复用功能配置




配置 I/O 为复用功能时，I/O 电路结构如上图所示：

- 施密特触发器输入使能；
- 上拉、下拉电阻的选择由 GPIOx_PUPD 寄存器中的值决定；
- 读访问输入数据寄存器获取 I/O 状态；
- 输出缓冲器
  - 可配置为开漏或推挽模式；
  - 由外设信号驱动。

### 7.3.10 复用功能选择

用户可通过复用功能寄存器 GPIOx_AFH 和 GPIOx_AFL 为每个 GPIO 分配可用的复用输入/输出功能。每个 GPIO 引脚包含多个外设复用功能，即应用程序可根据外设需求，将外设复用功能连接到包含该复用功能的引脚上。外设复用功

能参见下表：

表 7-2　端口复用功能映射

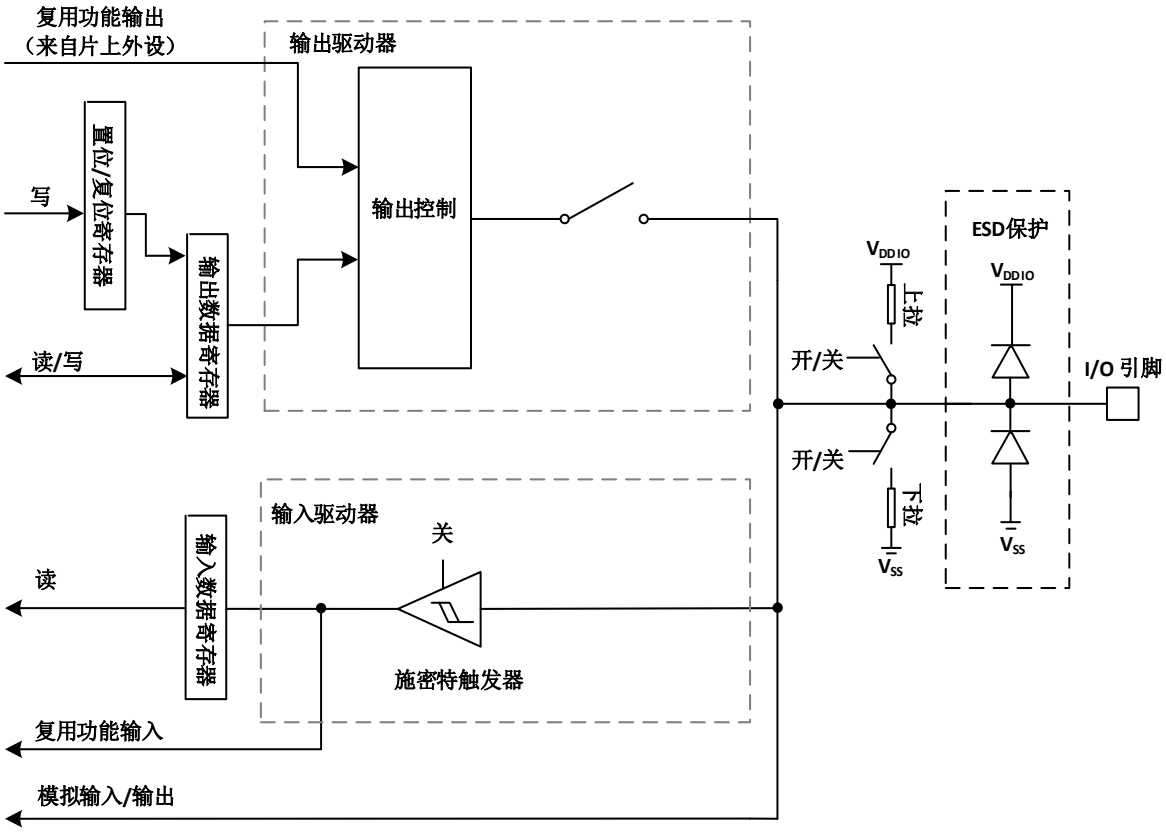
| PORT | AF0 | AF1 | AF2 | AF3 | AF4 | AF5 | AF6 | AF7 |
|---|---|---|---|---|---|---|---|---|
| PA0 | SPI1_SCK | UART2_RX | LPUART1_RX | HALL_IN1 | SEG17 | LPTIM1_IN | TIM4_ETR | COMP1_OUT |
| PA1 | SPI1_NSS | SPI2_MOSI | LPUART1_TX | SPI1_MISO | SEG16 | TIM3_ETR | HALL_IN3 | - |
| PA2 | SPI1_MOSI | SPI2_MISO | COMP2_OUT | UART2_TX | SEG15 | TIM1_BKIN | HALL_IN2 | UART1_TX |
| PA3 | SPI1_MISO | SPI2_NSS | LPUART1_RX | UART2_RX | SEG14 | TIM1_CH5 | HALL_IN1 | UART1_RX |
| PA4 | SPI1_NSS | SPI1_MISO | LPUART1_TX | TIM4_CH1 | SEG13 | TIM1_CH6 | TIM3_CH1 | I2C1_SCL |
| PA5 | SPI1_SCK | SPI1_MOSI | - | TIM4_CH2 | SEG12 | TIM1_CH7 | TIM3_CH2 | - |
| PA6 | SPI1_MISO | COMP1_OUT | LPUART1_TX | TIM1_BKIN | SEG11 | TIM1_CH8 | TIM3_CH1 | I2C1_SCL |
| PA7 | SPI1_MOSI | SPI2_NSS | LPUART1_RX | COMP2_OUT | SEG10 | TIM1_CH1 | TIM3_CH2 | I2C1_SDA |
| PA8 | - | SPI2_NSS | UART1_TX | TIM4_CH1 | SEG0 | TIM1_CH4 | TIM1_CH8 | MCO |
| PA9 | MCO | SPI2_SCK | UART1_TX | UART2_TX | COM0 | TIM1_CH5 | TIM4_CH2 | I2C1_SCL |
| PA10 | SPI1_SCK | SPI2_MISO | UART1_RX | UART2_RX | COM1 | TIM1_CH6 | TIM3_CH3 | I2C1_SDA |
| PA11 | SPI1_MISO | SPI2_MOSI | COMP1_OUT | UART1_RX | COM2 | TIM1_CH7 | TIM3_CH2 | I2C1_SCL |
| PA12 | SPI1_MOSI | SPI2_MISO | COMP2_OUT | UART1_TX | COM3 | TIM1_BKIN | TIM3_CH1 | I2C1_SDA |
| PA13 | SWDIO | SPI1_SCK | UART1_RX | UART2_RX | - | TIM1_CH8 | IR_OUT | I2C1_SDA |

| PORT | AF0 | AF1 | AF2 | AF3 | AF4 | AF5 | AF6 | AF7 |
|---|---|---|---|---|---|---|---|---|
| PA14 | SWCLK | SPI1_MISO | UART1_TX | UART2_TX | - | TIM1_CH7 | MCO | I2C1_SCL |
| PA15 | SPI1_NSS | LPUART1_RTS | UART1_RX | UART2_RX | - | TIM1_CH1 | TIM4_CH1 | TIM3_CH1 |
| PB0 | - | SPI2_SCK | UART1_RX | LPUART1_TX | SEG9 | TIM1_CH2 | TIM3_CH3 | - |
| PB1 | SPI1_MOSI | SPI2_MOSI | UART1_TX | LPUART1_RTS | SEG8 | TIM1_CH3 | HALL_IN1 | TIM3_CH2 |
| PB2 | TIM4_CH1 | SPI2_MISO | - | LPUART1_RX | SEG7 | TIM1_CH4 | HALL_IN2 | TIM3_CH1 |
| PB3 | SPI1_SCK | - | LPUART1_TX | TIM3_ETR | - | TIM1_CH2 | TIM4_CH2 | TIM3_CH2 |
| PB4 | SPI1_MISO | - | LPUART1_RX | TIM3_CH1 | - | TIM1_CH3 | TIM4_ETR | - |
| PB5 | SPI1_MOSI | - | LPUART1_TX | TIM3_CH2 | - | TIM1_CH4 | - | LPTIM1_IN |
| PB6 | SPI2_MOSI | UART1_TX | - | TIM3_CH3 | - | TIM1_CH5 | - | I2C1_SCL |
| PB7 | SPI2_MISO | UART1_RX | LPUART1_RX | IR_OUT | SEG20 | TIM1_CH6 | LPUART1_CTS | I2C1_SDA |
| PB8 | SPI2_SCK | HALL_IN3 | UART1_RX | UART2_TX | SEG19 | TIM1_CH7 | TIM4_CH1 | I2C1_SCL |
| PB9 | SPI2_NSS | IR_OUT | UART1_TX | UART2_RX | SEG18 | TIM1_CH8 | TIM4_CH2 | I2C1_SDA |
| PB10 | SPI1_MOSI | SPI2_SCK | SPI1_MISO | LPUART1_TX | SEG6 | TIM4_CH2 | HALL_IN3 | I2C1_SCL |
| PB11 | - | - | UART1_RX | LPUART1_RX | SEG5 | TIM1_CH8 | TIM3_ETR | I2C1_SDA |
| PB12 | SPI1_NSS | SPI2_NSS | UART1_TX | LPUART1_TX | SEG4 | TIM1_BKIN | TIM3_CH1 | COMP1_OUT |
| PB13 | SPI1_SCK | SPI2_SCK | LPUART1_CTS | COMP2_OUT | SEG3 | TIM1_CH1 | TIM3_CH2 | I2C1_SCL |

| PORT | AF0 | AF1 | AF2 | AF3 | AF4 | AF5 | AF6 | AF7 |
|---|---|---|---|---|---|---|---|---|
| PB14 | SPI1_MISO | SPI2_MISO | LPUART1_RTS | RTC_OUT | SEG2 | TIM1_CH2 | TIM3_CH3 | I2C1_SDA |
| PB15 | SPI1_MOSI | SPI2_MOSI | UART1_RX | TIM4_CH2 | SEG1 | TIM1_CH3 | - | LPTIM1_IN |
| PC13 | HALL_IN2 | SPI2_MISO | LPUART1_CTS | SPI2_MOSI | - | - | - | RTC_OUT |
| PC14 | HALL_IN1 | - | - | - | - | TIM1_BKIN | TIM3_ETR | - |
| PC15 | HALL_IN2 | SPI2_MOSI | LPUART1_RTS | - | - | TIM1_CH1 | TIM3_CH3 | - |
| PF0 | HALL_IN3 | SPI2_NSS | UART1_RX | UART2_RX | - | TIM1_CH2 | TIM3_CH2 | I2C1_SDA |
| PF1 | HALL_IN1 | SPI2_SCK | UART1_TX | UART2_TX | - | TIM1_CH3 | TIM3_CH1 | I2C1_SCL |
| PF2 | - | - | - | - | - | - | - | - |
| PF3 | HALL_IN3 | TIM1_BKIN | LPUART1_RTS | UART2_TX | - | TIM1_CH6 | TIM4_CH1 | - |
| PF4 | HALL_IN2 | RTC_OUT | LPUART1_CTS | UART2_RX | - | TIM1_CH7 | TIM4_CH2 | IR_OUT |
| PF5 | - | - | - | - | - | - | - | - |
| PF6 | TIM4_CH1 | SPI2_SCK | UART1_TX | TIM3_CH2 | - | TIM1_BKIN | TIM1_CH8 | I2C1_SCL |
| PF7 | TIM4_CH2 | SPI2_NSS | UART1_RX | TIM3_CH1 | - | - | TIM1_CH7 | I2C1_SDA |

### 7.3.11 模拟模式配置

图 7-5 I/O 引脚高阻态模拟配置




配置 I/O 为模拟模式时，I/O 电路结构如上图所示：

- 输出缓冲器禁用；
- 施密特触发器输入禁用，强制施密特触发器输出为 0，此时 I/O 引脚的模拟输入功耗为 0；
- 上拉、下拉电阻禁用；
- 读访问输入数据寄存器的值为 0。

### 7.3.12 外部振荡器引脚用作 GPIO

当 LXTAL 振荡器关闭（复位后的默认状态）时，可将相关的振荡器引脚用作通用 GPIO。

当 LXTAL 振荡器开启（设置 *AWON 域控制寄存器(RCC_AWCR)*中的 LXTALON 位）时，振荡器会控制与其相关联的引脚，这些引脚的 GPIO 配置无效。

将振荡器配置为外部时钟源旁路模式（设置 *AWON 域控制寄存器(RCC_AWCR)*

中的 LXTALBYP 位）时，振荡器引脚仅保留 LXTAL_IN，LXTAL_OUT 可用作通用 GPIO。

表 7-3　外部振荡器引脚

| 外部晶体振荡器 | 振荡器引脚功能 | 引脚编号 |
|---|---|---|
| LXTAL | LXTAL_IN | PC14 |
| | LXTAL_OUT | PC15 |

### 7.3.13　外部时钟输入引脚用作 GPIO

当关闭外部时钟输入（复位后的默认状态）时，可将相关引脚用作通用 GPIO。

使能外部时钟输入通过*时钟控制/状态寄存器 1（RCC_CSR1）*中的 EXTCLKON 位控制 PF0 引脚输入外部时钟，且该引脚的 GPIO 配置无效。

## 7.4 GPIO 寄存器

GPIO 寄存器支持 32 位访问。

表 7-4 GPIO 基地址

| 外设 | 基地址 |
|---|---|
| GPIOA | 0x5000 0000 |
| GPIOB | 0x5000 0400 |
| GPIOC | 0x5000 0800 |
| GPIOF | 0x5000 1400 |

### 7.4.1 GPIO 端口模式寄存器（GPIOx_MODE）（x = A、B、C、F）

偏移地址：0x00

复位值：端口 A 0xEBFF FFFF

端口 B 0xFFFF FFFF

端口 C 0xFC00 0000

端口 F 0x0000 FFFF

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MODE15[1:0] | | MODE14[1:0] | | MODE13[1:0] | | MODE12[1:0] | | MODE11[1:0] | | MODE10[1:0] | | MODE9[1:0] | | MODE8[1:0] | |
| 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 |
| MODE7[1:0] | | MODE6[1:0] | | MODE5[1:0] | | MODE4[1:0] | | MODE3[1:0] | | MODE2[1:0] | | MODE1[1:0] | | MODE0[1:0] | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | MODEy[1:0] | Portx端口I/O引脚y的I/O模式配置（x = A、B、C、F；y = 0～15）<br>00：输入模式<br>01：输出模式<br>10：复用功能模式<br>11：模拟模式 |

### 7.4.2 GPIO 端口输出类型寄存器（GPIOx_OTYPE）（x = A、B、C、F）

偏移地址：0x04

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OT15 | OT14 | OT13 | OT12 | OT11 | OT10 | OT9 | OT8 | OT7 | OT6 | OT5 | OT4 | OT3 | OT2 | OT1 | OT0 |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15:0 | OTy | Portx端口I/O引脚y的输出类型配置（x = A、B、C、F；y = 0 ~ 15）<br>0：推挽输出<br>1：开漏输出 |

### 7.4.3 GPIO 端口上拉/下拉寄存器（GPIOx_PUPD）（x = A、B、C、F）

偏移地址：0x0C

复位值：端口 A　　0x2400 0000

其他端口　　0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PUPD15[1:0] | | PUPD14[1:0] | | PUPD13[1:0] | | PUPD12[1:0] | | PUPD11[1:0] | | PUPD10[1:0] | | PUPD9[1:0] | | PUPD8[1:0] | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PUPD7[1:0] | | PUPD6[1:0] | | PUPD5[1:0] | | PUPD4[1:0] | | PUPD3[1:0] | | PUPD2[1:0] | | PUPD1[1:0] | | PUPD0[1:0] | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | PUPDy[1:0] | Portx端口I/O引脚y的上/下拉配置（x=A、B、C、F；y=0~15）<br>00：无上拉或下拉<br>01：上拉<br>10：下拉<br>11：保留（写入无效） |

### 7.4.4 GPIO 端口输入寄存器（GPIOx_IDR）（x = A、B、C、F）

偏移地址：0x10

复位值： 0x0000 XXXX

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ID15 | ID14 | ID13 | ID12 | ID11 | ID10 | ID9 | ID8 | ID7 | ID6 | ID5 | ID4 | ID3 | ID2 | ID1 | ID0 |
| r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15:0 | IDy | Portx端口I/O引脚y的输入数据（x＝A、B、C、F；y＝0～15） |

### 7.4.5 GPIO 端口输出数据寄存器（GPIOx_ODR）（x＝A、B、C、F）

偏移地址：0x14

复位值： 0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OD15 | OD14 | OD13 | OD12 | OD11 | OD10 | OD9 | OD8 | OD7 | OD6 | OD5 | OD4 | OD3 | OD2 | OD1 | OD0 |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15:0 | ODy | Portx端口I/O引脚y的输出数据（x＝A、B、C、F；y＝0～15） |

### 7.4.6 GPIO 端口置位/复位寄存器（GPIOx_BSR）（x＝A、B、C、F）

偏移地址：0x18

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BR15 | BR14 | BR13 | BR12 | BR11 | BR10 | BR9 | BR8 | BR7 | BR6 | BR5 | BR4 | BR3 | BR2 | BR1 | BR0 |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |

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

| BS15 | BS14 | BS13 | BS12 | BS11 | BS10 | BS9 | BS8 | BS7 | BS6 | BS5 | BS4 | BS3 | BS2 | BS1 | BS0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | BRy | Portx端口I/O引脚y的复位配置（x = A、B、C、F；y = 0 ~ 15）<br>0：写入无效<br>1：复位<br>*注意：如果同时对BSy和BRy置位，则BSy的优先级更高。* |
| 15:0 | BSy | Portx端口I/O引脚y的置位配置（x = A、B、C、F；y = 0 ~ 15）<br>0：写入无效<br>1：置位 |

### 7.4.7 GPIO 复用功能低位寄存器（GPIOx_AFL）（x = A、B、F）

偏移地址：0x20

复位值： 0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | AFSELy[2:0] | Portx端口I/O引脚y的复用功能选择（x = A、B、F；y = 0 ~ 7）<br>000：AF0<br>001：AF1<br>010：AF2<br>011：AF3<br>100：AF4<br>101：AF5<br>110：AF6<br>111：AF7 |

### 7.4.8 GPIO 复用功能高位寄存器（GPIOx_AFH）（x = A、B、C）

偏移地址：0x24

复位值： 0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | AFSEL15[2:0] | | | Res. | AFSEL14[2:0] | | | Res. | AFSEL13[2:0] | | | Res. | AFSEL12[2:0] | | |
| | rw | rw | rw | | rw | rw | rw | | rw | rw | rw | | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | AFSEL11[2:0] | | | Res. | AFSEL10[2:0] | | | Res. | AFSEL9[2:0] | | | Res. | AFSEL8[2:0] | | |
| | rw | rw | rw | | rw | rw | rw | | rw | rw | rw | | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | AFSELy[2:0] | Portx端口I/O引脚y的复用功能选择（x＝A、B、C；y＝8～15）<br>000：AF0<br>001：AF1<br>010：AF2<br>011：AF3<br>100：AF4<br>101：AF5<br>110：AF6<br>111：AF7 |

### 7.4.9 GPIO 端口位复位寄存器（GPIOx_BR）（x＝A、B、C、F）

偏移地址：0x28

复位值： 0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BR15 | BR14 | BR13 | BR12 | BR11 | BR10 | BR9 | BR8 | BR7 | BR6 | BR5 | BR4 | BR3 | BR2 | BR1 | BR0 |
| w | w | w | w | w | w | w | w | w | w | w | w | w | w | w | w |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15:0 | BRy | Portx端口I/O引脚y位复位（x＝A、B、C、F；y＝0～15）<br>0：写入无效<br>1：复位 |

# 8 直接存储器访问控制器（DMA）

## 8.1 简介

直接访问控制器（DMA）是总线矩阵的一个主机，用于存储器和存储器、存储器和外设、外设和外设之间的数据传输，传输过程中无需 CPU 参与。

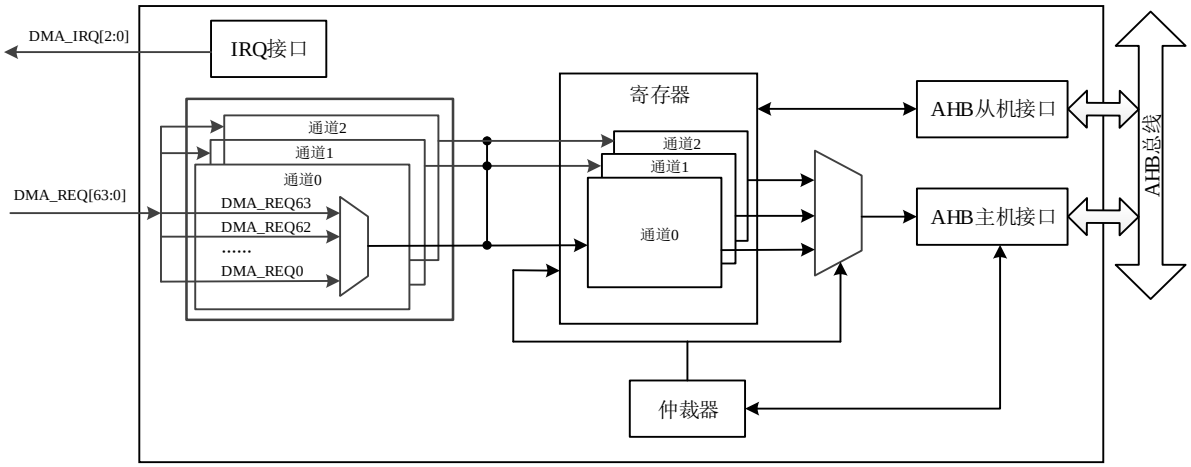
## 8.2 DMA 主要特性

- 3 条独立 DMA 通道
- 2 种通道请求触发类型：外设触发和软件触发
- 4 种 DMA 传输模式：存储器到存储器、存储器到外设、外设到存储器及外设到外设数据传输
- 2 种 DMA 传输类型：Block 传输和 Burst 传输
- 可配置 4 种软件优先级（超高、高、中或低），软件优先级相同时由硬件优先级决定（通道编号小的优先级高）
- 可配置的地址变化模式：固定、递增
- 可配置的传输数据位宽：字节、半字或字
- 可配置的传输次数：0 到 65535
- 支持循环模式
- 可触发中断的事件：传输完成、传输一半和传输错误

## 8.3 DMA 功能说明

### 8.3.1 DMA 框图

图 8-1 DMA 框图




### 8.3.2 DMA 通道请求

DMA 通道支持的所有通道请求信号参见下表。

表 8-1 DMA 通道请求信号

| 请求序号 | 请求外设 | 请求信号 |
|---|---|---|
| | | 通道 0/1/2 |
| 0 | 软件触发 | 软件触发通道请求 |
| 1 | ADC | ADC_DMA |
| 2 | SPI1 | SPI1_RX |
| 3 | | SPI1_TX |
| 4 | SPI2 | SPI2_RX |
| 5 | | SPI2_TX |
| 6 | I2C1 | I2C1_RX |
| 7 | | I2C1_TX |
| 8 | UART1 | UART1_RX |
| 9 | | UART1_TX |
| 10 | UART2 | UART2_RX |
| 11 | | UART2_TX |
| 12 | LPUART1 | LPUART1_RX |
| 13 | | LPUART1_TX |
| 14 | TIM1 | TIM1_CC1 |

| 请求序号 | 请求外设 | 请求信号 |
|---|---|---|
| | | 通道 0/1/2 |
| 15 | | TIM1_CC2 |
| 16 | | TIM1_CC3 |
| 17 | | TIM1_CC4 |
| 18 | | TIM1_CC5 |
| 19 | | TIM1_CC6 |
| 20 | | TIM1_CC7 |
| 21 | | TIM1_CC8 |
| 22 | | TIM1_COM |
| 23 | | TIM1_UP |
| 24 | TIM3 | TIM3_CC1 |
| 25 | | TIM3_CC2 |
| 26 | | TIM3_TRIG |
| 27 | | TIM3_UP |
| 28 | 保留 | 保留 |
| 29 | LPTIM1 | LPTIM1_ARRM |
| 30 | TIM4 | TIM4_CC1 |
| 31 | | TIM4_CC2 |
| 32 | | TIM4_TRIG |
| 33 | | TIM4_UP |
| 34 | HALLTIM | HALLTIM_CHG |
| 35 | | HALLTIM_OVR |
| 36-63 | 保留 | 保留 |

DMA 的通道请求有两种触发方式：

- 外设触发通道请求：配置 DMA_CCx 寄存器中的 REQ_ID[5:0]位域为 1~63。通道使能后，由外设的通道请求启动 DMA 传输；
- 软件触发通道请求：配置 DMA_CCx 寄存器中的 REQ_ID[5:0]位域为 0。通道使能后，立即启动 DMA 传输。

### 8.3.3 DMA 传输

DMA 通道启动传输后，首先从源地址读取单个数据（字节、半字或字），再向目的地址写入单个数据(字节、半字或字)。DMA_CNDTRx 寄存器中的 NDT[15:0]位域在传输后递减，该寄存器指示待传输的次数。若 DMA 通道无传输错误产

生，则会重复上述传输过程直至 DMA_CNDTRx 寄存器的值递减到 0。

### 传输模式

DMA 按照源地址和目的地址的地址类型分为四种传输模式：

- 存储器到存储器

  源和目的地址均为存储器。在此模式下，配置为循环模式时：若软件触发通道请求，则 DMA 通道在传输完一轮之后（DMA_CNDTRx 寄存器递减至 0 时）停止传输，不会进行循环传输；若外设触发通道请求，则支持循环模式。

- 存储器到外设

  源和目的地址为存储器和外设。

- 外设到存储器

  源和目的地址为外设和存储器。

- 外设到外设

  源和目的地址均为外设。当通道请求由一个外设触发时，另一外设作为此通道的被动访问外设。

### 传输类型

DMA 包含两种传输类型：Block 传输和 Burst 传输，可以通过 DMA_CCx 寄存器中的 TYPE 位选择 DMA 通道的传输类型。两种类型传输的通道请求可由外设请求触发，也可由软件请求触发。

表 8-2　DMA 传输类型概述

| 功能对比 | 软件触发<br>Block 传输 | 软件触发<br>Burst 传输 | 外设触发<br>Block 传输 | 外设触发<br>Burst 传输 |
|---|---|---|---|---|
| 触发传输条件 | 通道使能 | 通道使能 | 外设触发通道请求 | 外设触发通道请求 |
| CNDTR 指示 | 待传输数据 | 待传输数据 | 待处理请求 | 待传输数据 |
| 单次请求传输数量 | NDT(1) | NDT | 1 | NDT |
| 配置后的传输总数(2) | NDT | NDT | NDT | NDT |
| 通道暂停与停止 | 支持 | 不支持 | 支持 | 不支持 |

1. NDT 指写入 CNDTR 中 NDT[15:0]位域的值；
2. 配置后的传输总数指在写入 CNDTR 的值为 NDT 时，传输完成后传输数据的总数。

### 8.3.4 优先级

通道优先级分为软件优先级和硬件优先级：

- 软件优先级：软件优先级由 DMA_CCx 寄存器中的 PL[1:0]位域决定，可配置为超高、高、中和低四个等级；
- 硬件优先级：当多个通道配置的软件优先级相同时，则通道号小的优先级高。例如，通道 1 和通道 2 配置相同的软件优先级，则通道 1 的优先级高于通道 2。

当多个通道同时接收到请求时，则最高优先级的通道优先启动传输。在 Burst 传输过程中，不会被其它通道接收到的高优先级请求打断，当 Burst 传输完成后启动其它高优先级通道的传输。

### 8.3.5 数据位宽

单次传输数据位宽（字节、半字或字）由 DMA_CCx 寄存器中的 SIZE[1:0]位域决定。

*注意：　传输地址与数据位宽保持对齐，否则硬件会自动对齐地址，导致传输数据错误。*

表 8-3　DMA 地址对齐

| SIZE[1:0] | 数据位宽 | 源和目的地址 |
|---|---|---|
| 00 | 字节 | 无限制 |
| 01 | 半字 | 地址的第 0 位为 0 |
| 10 | 字 | 地址的第 0 位和第 1 位均为 0 |

### 8.3.6 寻址范围

DMA_CSARx 寄存器和 DMA_CDARx 寄存器，决定 DMA 传输的源地址和目的地址，可配置地址范围见下表，详见*图：储器映射*。

表 8-4　DMA 寻址范围

| 地址段 | 地址范围 |
|---|---|
| 重映射地址段 | 0x0000 0000 ~ 0x0000 FFFF |
| User flash 地址段 | 0x0800 0000 ~ 0x0800 FFFF |
| SRAM 地址段 | 0x2000 0000 ~ 0x2000 1FFF |
| APB 地址段 | 0x4000 0000 ~ 0x4001 FFFF |

### 8.3.7 地址递增

DMA_CCx 寄存器中的 SINC 位和 DINC 位，决定源和目的地址指针在每次传输完成后是否递增，详情见下表。

表 8-5 DMA 地址递增

| SINC | DINC | 源地址 | 目的地址 |
|---|---|---|---|
| 0 | 0 | 地址固定 | 地址固定 |
| 0 | 1 | 地址固定 | 地址递增 |
| 1 | 0 | 地址递增 | 地址固定 |
| 1 | 1 | 地址递增 | 地址递增 |

SINC 位或 DINC 位置 1 时，则源地址指针或者目的地址指针在本次传输的地址加上 1、2 或 4，取决于 SIZE[1:0]位域配置的数据位宽。

DMA_CSARx 寄存器和 DMA_CDARx 寄存器指示当前传输地址。

### 8.3.8 循环模式

#### 非循环模式

当通道配置为非循环模式（DMA_CCx 寄存器 CIRC 位清 0），最后一次数据传输完成后（即待传输的数据数量 DMA_CNDTRx 寄存器减小到 0 后），DMA_CNDTRx 寄存器将保持为 0，DMA_CSARx 寄存器和 DMA_CDARx 寄存器将自动重新加载起始地址，通道继续保持使能状态，但不响应任何 DMA 通道请求。重新使用通道进行相同配置传输时，只需先禁止 DMA 通道，然后修改 DMA_CNDTRx 寄存器的值，并再次使能通道即可。

#### 循环模式

当通道配置为循环模式（DMA_CCx 寄存器 CIRC 位置 1），最后一次数据传输完成后，DMA_CNDTRx 寄存器将自动重新加载初始编程值，DMA_CSARx 寄存器和 DMA_CDARx 寄存器将自动重新加载起始地址，通道继续保持使能状态，并响应后续 DMA 通道请求。

循环模式可用于处理循环缓冲区，例如 ADC 10 个通道循环扫描的数据保存在 SRAM 一段固定地址的缓冲区内；ADC 10 个通道的数据采集完成后再次回到第一个通道时，DMA 的起始地址也自动切回到初始地址，ADC 再次请求传输时，相同通道的数据将覆盖上一次数据。

### 8.3.9 配置流程

配置 DMA 通道 x 时需按照以下步骤操作：

1) 配置 DMA_CCx 寄存器中的下列参数：
   - 传输数据位宽 SIZE[1:0]
   - 源地址递增 SINC
   - 目的地址递增 DINC
   - 优先级 PL[1:0]
   - 传输类型 TYPE
   - 通道请求 REQ_ID[5:0]
   - 循环模式 CIRC
   - 传输完成中断使能 TF、传输一半中断使能 TH 和传输错误中断使能 TE
2) 配置 DMA_CNDTRx 寄存器中的 NDT[15:0]位域，设置传输的数据次数；
3) 配置 DMA_CSARx 寄存器设置读取数据的源地址；
4) 配置 DMA_CDARx 寄存器设置写入数据的目的地址；
5) 配置 DMA_CCx 寄存器中的 EN 位置 1 以使能通道；
6) 如果设置外设触发通道请求，等待外设触发请求以启动传输。如果设置软件触发通道请求，则 DMA 通道立即启动传输。

### 8.3.10 暂停和停止通道

DMA 通道设置为 Burst 传输类型时，无法暂停和停止正在进行的传输，通道设置为 Block 传输类型时可以暂停和停止。

- 暂停和恢复通道
  - DMA 待传输的数据量不为 0 之前，将 DMA_CCx 寄存器中的 EN 位清 0，通道其它配置不更改，可暂停此通道。
  - 将 DMA_CCx 寄存器中的 EN 位置 1 可继续通道传输。
- 停止和重启通道
  - DMA 待传输的数据量不为 0 之前，将 DMA_CCx 寄存器中的 EN 位清 0（如果通道 x 由外设触发 DMA 通道请求，需先将外设请求禁止），则通道 x 会在当次数据传输完成后停止，等待 DMA_CCx 寄存器中的 EN 位清 0 确保通道停止。
  - 重启此通道进行其它传输，需重新配置 DMA_CCx、DMA_CSARx、

DMA_CDARx 或 DMA_CNDTRx 寄存器中任意参数后，再次使能通道启动传输。

### 8.3.11 错误标志

当通道 x 对保留的地址空间执行读写操作时，传输错误标志 TEx 置 1 且通道使能位 EN 清 0，从而自动禁止出错的通道 x。将 DMA_ICR 寄存器中的 TECFx 位置 1，TEx 标志将同步清 0。

如果通道 x 为外设触发通道请求时，首先要停止外设的 DMA 触发使能，以禁止任何挂起或后续新生成的 DMA 请求。清除 TEx 后再正常地将 DMA 和外设重新配置，以便进行新的传输。

## 8.4 DMA 中断

DMA 中断详情参见下表。

表 8-6 中断请求

| 中断事件 | 事件标志 | 中断使能位 | 事件标志/中断清除方法 |
|---|---|---|---|
| 通道 x 传输完成 | TFx | TFIE | 向 TFCFx 位或 GCFx 位写 1 |
| 通道 x 传输一半 | THx | THIE | 向 THCFx 位或 GCFx 位写 1 |
| 通道 x 传输错误 | TEx | TEIE | 向 TECFx 位或 GCFx 位写 1 |

## 8.5 DMA 寄存器

DMA 寄存器支持 32 位访问。

表 8-7 DMA 基地址

| 外设 | 基地址 |
|---|---|
| DMA | 0x4002 0000 |

### 8.5.1 DMA 中断状态寄存器（DMA_ISR）

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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | TE2 | TH2 | TF2 | G2 | TE1 | TH1 | TF1 | G1 | TE0 | TH0 | TF0 | G0 |
| | | | | r | r | r | r | r | r | r | r | r | r | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:12 | 保留 | 写入无效 |
| 11 | TE2 | 通道 2 的传输错误标志<br>传输错误时置 1。将 DMA_ICR 寄存器中的 TECF2 位置 1，此位清 0。<br>0：无 TE 事件<br>1：发生 TE 事件 |
| 10 | TH2 | 通道 2 的传输一半标志<br>传输一半时置 1。将 DMA_ICR 寄存器中的 THCF2 位置 1，此位清 0。<br>0：无 TH 事件<br>1：发生 TH 事件 |
| 9 | TF2 | 通道 2 的传输完成标志<br>传输完成时置 1。将 DMA_ICR 寄存器中的 TFCF2 位置 1，此位清 0。 |

| | | |
|---|---|---|
| | | 0：无 TF 事件<br>1：发生 TF 事件 |
| 8 | G2 | 通道 2 的全局标志<br>传输错误、传输一半或传输完成时置 1。将 DMA_ICR 寄存器中的 GCF2 位置 1，TE2、TH2、TF2 和 G2 位都清 0。<br>0：无 TE、TH 或 TF 事件<br>1：发生 TE、TH 或 TF 事件 |
| 7 | TE1 | 通道 1 的传输错误标志<br>传输错误时置 1。将 DMA_ICR 寄存器中的 TECF1 位置 1，此位清 0。<br>0：无 TE 事件<br>1：发生 TE 事件 |
| 6 | TH1 | 通道 1 的传输一半标志<br>传输一半时置 1。将 DMA_ICR 寄存器中的 THCF1 位置 1，此位清 0。<br>0：无 TH 事件<br>1：发生 TH 事件 |
| 5 | TF1 | 通道 1 的传输完成标志<br>传输完成时置 1。将 DMA_ICR 寄存器中的 TFCF1 位置 1，此位清 0。<br>0：无 TF 事件<br>1：发生 TF 事件 |
| 4 | G1 | 通道 1 的全局标志<br>传输错误、传输一半或传输完成时置 1。将 DMA_ICR 寄存器中的 GCF1 位置 1，TE1、TH1、TF1 和 G1 位都清 0。<br>0：无 TE、TH 或 TF 事件<br>1：发生 TE、TH 或 TF 事件 |
| 3 | TE0 | 通道 0 的传输错误标志<br>传输错误时置 1。将 DMA_ICR 寄存器中的 TECF0 位置 1，此位清 0。<br>0：无 TE 事件 |

| | | |
|---|---|---|
| | | 1：发生 TE 事件 |
| 2 | TH0 | 通道 0 的传输一半标志<br>传输一半时置 1。将 DMA_ICR 寄存器中的 THCF0 位置 1，此位清 0。<br>0：无 TH 事件<br>1：发生 TH 事件 |
| 1 | TF0 | 通道 0 的传输完成标志<br>传输完成时置 1。将 DMA_ICR 寄存器中的 TFCF0 位置 1，此位清 0。<br>0：无 TF 事件<br>1：发生 TF 事件 |
| 0 | G0 | 通道 0 的全局标志<br>传输错误、传输一半或传输完成时置 1。将 DMA_ICR 寄存器中的 GCF0 位置 1，TE0、TH0、TF0 和 G0 位都清 0。<br>0：无 TE、TH 或 TF 事件<br>1：发生 TE、TH 或 TF 事件 |

### 8.5.2 DMA 中断标志清除寄存器（DMA_ICR）

偏移地址：0x04

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | | | | TECF2 | THCF2 | TFCF2 | GCF2 | TECF1 | THCF1 | TFCF1 | GCF1 | TECF0 | THCF0 | TFCF0 | GCF0 |
| | | | | w | w | w | w | w | w | w | w | w | w | w | w |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:12 | 保留 | 写入无效 |
| 11 | TECF2 | 通道 2 的传输错误清除<br>写 1 清除 DMA_ISR 寄存器中的 TE2 标志。 |

| | | |
|---|---|---|
| 10 | THCF2 | 通道 2 的传输一半标志清除<br>写 1 清除 DMA_ISR 寄存器中的 TH2 标志。 |
| 9 | TFCF2 | 通道 2 的传输完成标志<br>写 1 清除 DMA_ISR 寄存器中的 TF2 标志。 |
| 8 | GCF2 | 通道 2 的全局标志<br>写 1 清除 DMA_ISR 寄存器中的 TE2、TH2、TF2 和 G2 标志。 |
| 7 | TECF1 | 通道 1 的传输错误清除<br>写 1 清除 DMA_ISR 寄存器中的 TE1 标志。 |
| 6 | THCF1 | 通道 1 的传输一半标志清除<br>写 1 清除 DMA_ISR 寄存器中的 TH1 标志。 |
| 5 | TFCF1 | 通道 1 的传输完成标志<br>写 1 清除 DMA_ISR 寄存器中的 TF1 标志。 |
| 4 | GCF1 | 通道 1 的全局标志<br>写 1 清除 DMA_ISR 寄存器中的 TE1、TH1、TF1 和 G1 标志。 |
| 3 | TECF0 | 通道 0 的传输错误清除<br>写 1 清除 DMA_ISR 寄存器中的 TE0 标志。 |
| 2 | THCF0 | 通道 0 的传输一半标志清除<br>写 1 清除 DMA_ISR 寄存器中的 TH0 标志。 |
| 1 | TFCF0 | 通道 0 的传输完成标志<br>写 1 清除 DMA_ISR 寄存器中的 TF0 标志。 |
| 0 | GCF0 | 通道 0 的全局标志<br>写 1 清除 DMA_ISR 寄存器中的 TE0、TH0、TF0 和 G0 标志。 |

### 8.5.3 DMA 通道 x 控制寄存器（DMA_CCx）

偏移地址：0x08+0x14 ×x，（x = 0 ~ 2）

复位值：0x0000 0000

*注意：　使能通道后（EN=1），此寄存器除 TEIE、THIE、TFIE 位外其它位域只读，禁止写入。*

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | TYPE | CIRC | Res. | PL[1:0] | | SIZE[1:0] | | SINC | DINC | TEIE | THIE | TFIE | EN |
| | | | rw | rw | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:22 | 保留 | 写入无效 |
| 21:16 | REQ_ID[5:0] | 通道请求<br>参见*表：DMA 通道请求*。 |
| 15:13 | 保留 | 写入无效 |
| 12 | TYPE | 传输类型<br>0：Block 传输<br>1：Burst 传输 |
| 11 | CIRC | 循环模式<br>0：禁止<br>1：使能 |
| 10 | 保留 | 写入无效 |
| 9:8 | PL[1:0] | 软件优先级<br>00：低<br>01：中<br>10：高<br>11：超高 |
| 7:6 | SIZE[1:0] | 传输数据位宽<br>00：源和目的数据均为 8 位<br>01：源和目的数据均为 16 位<br>10：源和目的数据均为 32 位 |

11：保留（写入无效）

| | | |
|---|---|---|
| 5 | SINC | 源地址递增<br>0：禁止<br>1：使能 |
| 4 | DINC | 目的地址递增<br>0：禁止<br>1：使能 |
| 3 | TEIE | 传输错误中断使能<br>0：禁止<br>1：使能 |
| 2 | THIE | 传输一半中断使能<br>0：禁止<br>1：使能 |
| 1 | TFIE | 传输完成中断使能<br>0：禁止<br>1：使能 |
| 0 | EN | 通道使能<br>发生通道传输错误后，此位清 0。DMA_ISR 中的 TEx 位清 0 后，此位才能再次置 1。<br>0：禁止<br>1：使能<br>*注意：若停止正在传输的通道，则该通道会在当次数据传输完成后停止，即EN位会在传输完成后清0。* |

### 8.5.4 DMA 通道 x 待传输次数寄存器（DMA_CNDTRx）

偏移地址：0x0C+0x14 ×x，（x = 0 ~ 2）

复位值：0x0000 0000

*注意：　使能通道后为只读，禁止写入。*

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15:0 | NDT[15:0] | 待传输次数<br>使能通道后，每次 DMA 传输后，该位域都会减 1，指示剩余的待传输次数。如果该位域为 0，则不会处理任何传输请求。 |

### 8.5.5 DMA 通道 x 源地址寄存器（DMA_CSARx）

偏移地址：0x10+0x14 ×x，（x = 0 ~ 2）

复位值：0x0000 0000

*注意：　使能通道后为只读，禁止写入。*

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SA[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 |
| SA[15:0] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | SA[31:0] | 源地址<br>SIZE[1:0] = 01（16 位）时，忽略 SA[31:0]的位 0，访问将自动对齐到半字地址。<br>SIZE[1:0] = 10（32 位）时，忽略 SA[31:0]的位 1 和位 0，访问将自动对齐到字地址。<br>寻址范围详见*表：DMA 寻址范围*。 |

### 8.5.6 DMA 通道 x 目的地址寄存器（DMA_CDARx）

偏移地址：0x14+0x14 ×x，（x = 0 ~ 2）

复位值：0x0000 0000

*注意：　使能通道后为只读，禁止写入。*

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DA[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 |
| DA[15:0] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | DA[31:0] | 存储器地址<br>此寄存器由软件置 1 和清 0。<br>SIZE[1:0]＝01（16 位）时，忽略 DA[31:0]的位 0，访问将自动对齐到半字地址。<br>SIZE[1:0]＝10（32 位）时，忽略 DA[31:0]的位 1 和位 0，访问将自动对齐到字地址。<br>寻址范围详见*表：DMA 寻址范围*。 |

# 9 嵌套向量中断控制器（NVIC）

## 9.1 简介

嵌套向量中断控制器（NVIC）实现高效的异常处理及中断处理。所有的中断均由 NVIC 管理。

## 9.2 主要特征

- 支持 32 个可屏蔽中断，以及 Cortex-M0+的 16 个中断
- 低延时的异常及中断处理
- 4 个可编程优先级，“0”代表最高优先级
- 中断向量表中的编号代表硬件优先级，当两个软件优先级相同的中断同时发生，编号小的中断优先

NVIC 和处理器内核紧密配合，实现了低延迟的中断处理，以及对晚到高优先级中断的高效处理。当异常发生时，NVIC 会自动将处理状态压栈保存，中断子程序执行完毕会自动将状态出栈。

## 9.3 中断向量表

表 9-1 中断向量表 (1)

| 位置 | 优先级 | 优先级类型 | 缩略语 | 说明 | 地址 |
|---|---|---|---|---|---|
| - | - | - | - | 保留 | 0x0000_0000 |
| - | -3 | 固定 | 复位 | 复位 | 0x0000_0004 |
| - | -2 | 固定 | NMI_Handler | 不可屏蔽中断 | 0x0000_0008 |
| - | -1 | 固定 | HardFault_Handler | 所有类型的错误 | 0x0000_000C |
| - | - | - | 保留 | 保留 | 0x0000_0010~<br>0x0000_0028 |
| - | 3 | 可设置 | SVC_Handler | 通过 SWI 指令调用的系统服务 | 0x0000_002C |
| - | - | - | 保留 | 保留 | 0x0000_0030<br>0x0000_0034 |
| - | 5 | 可设置 | PendSV_Handler | 可挂起的系统服务请求 | 0x0000_0038 |
| - | 6 | 可设置 | SysTick_Handler | 系统节拍定时器 | 0x0000_003C |

| 位置 | 优先级 | 优先级类型 | 缩略语 | 说明 | 地址 |
|---|---|---|---|---|---|
| 0 | 7 | - | 保留 | 保留 | 0x0000_0040 |
| 1 | 8 | 可设置 | PVD | 可编程电压检测中断 | 0x0000_0044 |
| 2 | 9 | 可设置 | RTC | RTC 全局中断 | 0x0000_0048 |
| 3 | 10 | 可设置 | FLASH | Flash 全局中断 | 0x0000_004C |
| 4 | 11 | 可设置 | RCC | RCC 全局中断 | 0x0000_0050 |
| 5 | 12 | 可设置 | EXTI[1:0] | EXTI 通道 0 和 1 中断 | 0x0000_0054 |
| 6 | 13 | 可设置 | EXTI[3:2] | EXTI 通道 2 和 3 中断 | 0x0000_0058 |
| 7 | 14 | 可设置 | EXTI[15:4] | EXTI 通道 4 到 15 中断 | 0x0000_005C |
| 8 | 15 | 可设置 | LCD | LCD 全局中断 | 0x0000_0060 |
| 9 | 16 | 可设置 | DMA_Channel0 | DMA 通道 0 中断 | 0x0000_0064 |
| 10 | 17 | 可设置 | DMA_Channel1 | DMA 通道 1 中断 | 0x0000_0068 |
| 11 | 18 | 可设置 | DMA_Channel2 | DMA 通道 2 中断 | 0x0000_006C |
| 12 | 19 | 可设置 | ADC | ADC 全局中断 | 0x0000_0070 |
| 13 | 20 | 可设置 | TIM1_BRK_UP_COM | TIM1 刹车、更新和换相事件中断 | 0x0000_0074 |
| 14 | 21 | 可设置 | TIM1_CC | TIM1 比较中断 | 0x0000_0078 |
| 15 | 22 | 可设置 | TIM3 | TIM3 全局中断 | 0x0000_007C |
| 16 | 23 | 可设置 | TIM4 | TIM4 全局中断 | 0x0000_0080 |
| 17 | 24 | - | 保留 | 保留 | 0x0000_0084 |
| 18 | 25 | - | 保留 | 保留 | 0x0000_0088 |
| 19 | 26 | 可设置 | COMP | COMP1/2 全局中断 | 0x0000_008C |
| 20 | 27 | 可设置 | LPTIM1 | LPTIM1 全局中断 | 0x0000_0090 |
| 21 | 28 | 可设置 | I2C1 | I2C1 全局中断 | 0x0000_0094 |
| 22 | 29 | - | 保留 | 保留 | 0x0000_0098 |
| 23 | 30 | 可设置 | SPI1 | SPI1 全局中断 | 0x0000_009C |
| 24 | 31 | 可设置 | SPI2 | SPI2 全局中断 | 0x0000_00A0 |
| 25 | 32 | 可设置 | UART1 | UART1 全局中断 | 0x0000_00A4 |

| 位置 | 优先级 | 优先级类型 | 缩略语 | 说明 | 地址 |
|---|---|---|---|---|---|
| 26 | 33 | 可设置 | UART2 | UART2 全局中断 | 0x0000_00A8 |
| 27 | 34 | - | 保留 | 保留 | 0x0000_00AC |
| 28 | 35 | 可设置 | LPUART1 | LPUART1 全局中断 | 0x0000_00B0 |
| 29 | 36 | 可设置 | HALLTIM | HALLTIM 全局中断 | 0x0000_00B4 |
| 30 | 37 | - | 保留 | 保留 | 0x0000_00B8 |
| 31 | 38 | - | 保留 | 保留 | 0x0000_00BC |

1. 灰色部分对应 Cortex-M0+中断。

# 10 扩展中断和事件控制器（EXTI）

## 10.1 简介

扩展中断和事件控制器 EXTI 用于管理 CPU 唤醒和系统的唤醒。EXTI 的输入有可配置通道和直接通道两种。

EXTI 内部集成了 I/O 端口多路选择器，可将任意 I/O 配置为唤醒源。

## 10.2 EXTI 主要功能

- 由输入的任意事件唤醒系统；
- 可配置通道，来自不含中断功能的 GPIO，功能如下：
  - 触发沿可选；
  - 为事件和中断分别提供独立的屏蔽功能，灵活管理 CPU 的唤醒、中断处理以及事件响应。
- 直接通道，来自含中断功能的外设，功能如下：
  - 为事件和中断分别提供独立的屏蔽功能，灵活管理 CPU 的唤醒和事件响应。
- 集成 I/O 端口多路选择器，可将任意 I/O 配置为唤醒源。

## 10.3 EXTI 模块示意图

EXTI 的输入分为可配置通道输入和直接通道输入，输出分为事件输出和中断输出。

EXTI 信号处理模块内部实现的功能参见 *EXTI 功能描述*。

图 10-1 EXTI 框图

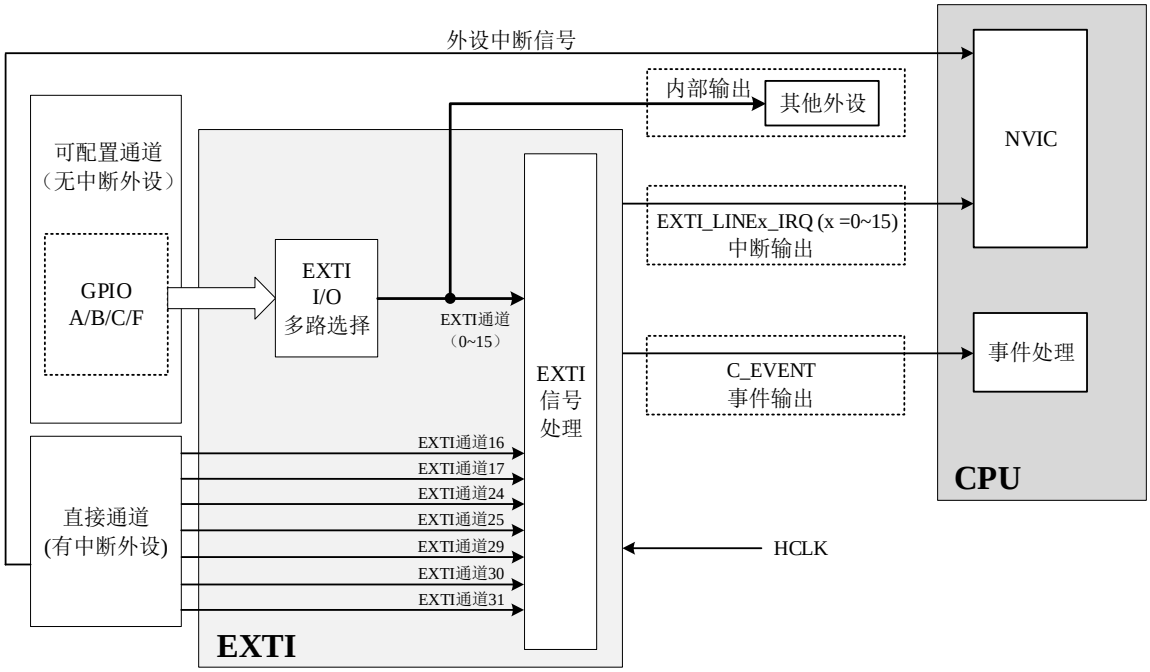



表 10-1 EXTI 内部信号

| 信号名称 | 信号类型 | 说明 |
|---|---|---|
| EXTI_LINEx_IRQ | 输出 | 由可配置通道产生的中断信号（x = 0 ~ 15） |
| C_EVENT | 输出 | 输出到 CPU 的事件输出信号，与 CPU 时钟同步 |
| HCLK | 输入 | EXTI 工作时钟 |

## 10.4 EXTI 的主要信号连接及功能

EXTI 为可配置通道提供了中断挂起标志，挂起标志置 1 代表有中断待处理，需要软件写“1”将标志位清零。EXTI 的中断挂起标志可触发 CPU 中断。

EXTI 中实现了 I/O 多路选择器。所有的 GPIO 端口都接入到 EXTI I/O 多路选择器，可选择将任意 GPIO 配置为 EXTI 的可配置通道，选中的 I/O 可将 CPU 从低功耗模式唤醒。

本身已实现唤醒事件及中断的外设作为 EXTI 的直接通道。

EXTI 管理来自各个通道的事件，输入到 CPU 的事件处理模块。

EXTI 的事件输出和中断输出可以将 CPU 从低功耗模式下唤醒，唤醒方式主要有以下两种：

- 使用 WFE 进入低功耗模式时，输入到 CPU 事件处理模块的事件信号可将 CPU 从低功耗模式唤醒，中断信号也可以将 CPU 从低功耗模式唤醒；

- 使用 WFI 进入低功耗模式时，中断信号可以将 CPU 从低功耗模式唤醒。

详情参见*低功耗模式*。

## 10.5 EXTI 通道及信号对应表

表 10-2 EXTI 通道及信号对应表

| EXTI 通道 | 事件源 | 类型 | 外设关联的信号 |
|---|---|---|---|
| 0~15 | GPIO | 可配置 | GPIO 输入信号 |
| 16 | COMP1 | 直接 | COMP1 比较器输出结果 |
| 17 | COMP2 | 直接 | COMP2 比较器输出结果 |
| 24 | PVD | 直接 | PVD报警事件 |
| 25 | RTC | 直接 | 中断使能时，RTC的中断请求 |
| 29 | LPUART1 | 直接 | 低功耗唤醒使能（UEWK=1）且中断使能时，如下中断请求支持唤醒：字符匹配中断、接收非空中断和检测到唤醒事件中断 |
| 30 | LPTIM1 | 直接 | 中断使能时，自动重载匹配 |
| 31 | LXTAL CSS | 直接 | 中断使能时，LXTAL时钟故障中断信号 |

## 10.6 EXTI 功能描述

可配置通道的使能由 EXTI 控制，通过配置触发沿寄存器使能相应通道。直接通道的使能在外设中。已使能的通道信号是否可以唤醒系统，在中断唤醒屏蔽寄存器 EXTI_IMR 或事件唤醒屏蔽寄存器 EXTI_EMR 中设置。详见下表：

表 10-3 EXTI 屏蔽功能

| 唤醒屏蔽寄存器的配置 | | 唤醒屏蔽寄存器的配置效果 | | |
|---|---|---|---|---|
| 中断唤醒屏蔽寄存器 IMR.IMn | 事件唤醒屏蔽寄存器 EMR.EMn | 可配置通道挂起寄存器 PIR.PIFn | 中断输出 | 事件输出 |
| 0 | 0 | 无效 | 屏蔽 | 屏蔽 |
| | 1 | 无效 | 屏蔽 | 有效 |
| 1 | 0 | 有效 | 有效 | 屏蔽 |
| | 1 | 有效 | 有效 | 有效 |

由可配置通道及直接通道产生的未屏蔽事件（EXTI_EMR. EMn=1）可以产生事件请求，也可以作为唤醒源唤醒 CPU。

### 10.6.1 可配置通道

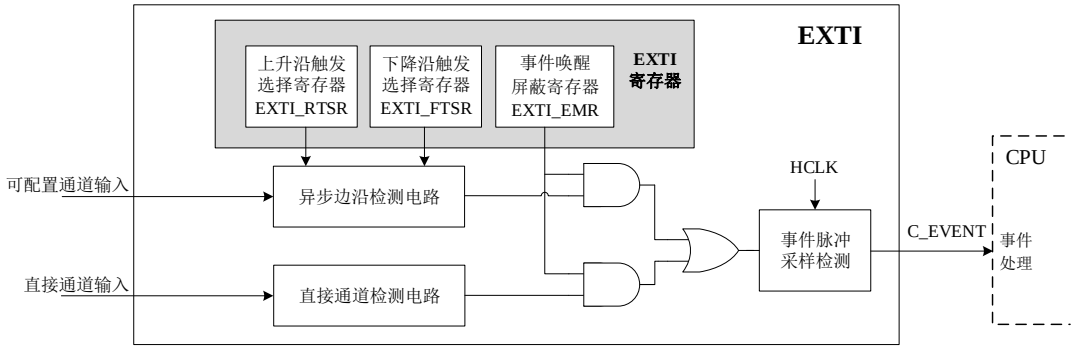
可配置事件的触发沿通过寄存器 EXTI_RTSR/FTSR 可选为上升沿、下降沿或双沿。

**可配置通道的事件输出**

可配置通道的事件输出由 EXTI_EMR 寄存器管理屏蔽，其结果输出到 CPU 事件处理模块，可触发 CPU 的事件响应。

参见下图：

图 10-2 可配置通道&直接通道的事件输出框图



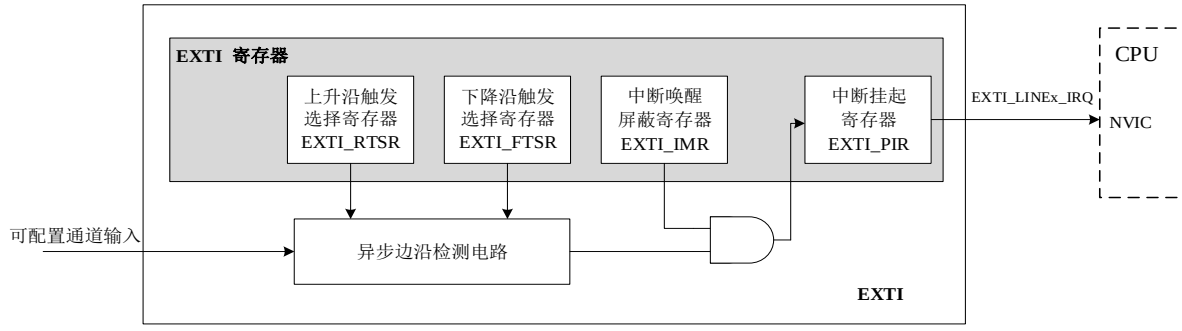

**可配置通道的中断输出**

EXTI 为可配置通道实现中断管理，提供中断挂起寄存器（EXTI_PIR）。产生的中断信号 EXTI_LINEx_IRQ 输入到 NVIC 管理，可触发 CPU 中断。

可配置通道检测到触发沿时，如果中断未屏蔽（EXTI_IMR.IMn=1），则中断挂起寄存器 EXTI_PIR 中的相应 PIFn 位被置位，产生中断请求输入到 NVIC，唤醒 CPU 并触发 CPU 中断。后续需将 PIFn 位软件写“1”以清除中断请求。

中断挂起寄存器仅响应来自可配置通道且未被屏蔽的（IMn=1）中断请求。与 EXTI_EMR 寄存器的配置无关。参见下图：

图 10-3 可配置通道的中断输出框图



### 10.6.2 直接通道

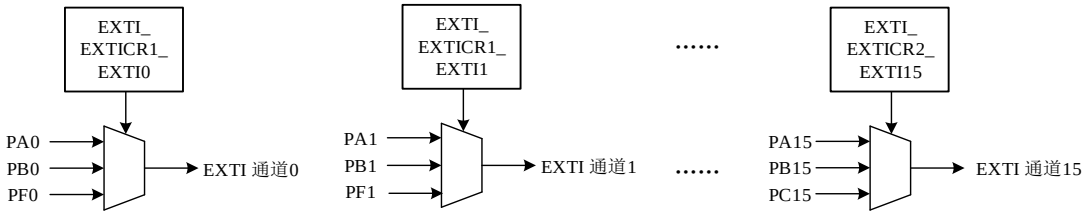
来自直接通道的事件或中断都可以唤醒系统。EXTI_EMR 寄存器和 EXTI_IMR 寄存器用来设置该事件或中断是否能将系统唤醒。

直接通道的中断在外设模块中实现，与 EXTI 无关。直接通道的事件输出参见*图：可配置通道&直接通道的事件输出框图*。

### 10.6.3 I/O 多路选择器

EXTI 的 I/O 多路选择器可选任意 GPIO 作为 EXTI 的信号源。所有的 GPIO 作为可配置通道分组接到 16 个多路选择器上，对应 EXTI 通道 0~15。在 EXTI_EXTICR1 和 EXTI_EXTICR2 寄存器中选择 GPIO 信号源。

图 10-4 EXTI 多路复用器




EXTI 的 I/O 多路选择器的内部输出信号直接连接到其他外设，可作为相关外设的触发源。I/O 多路选择器内部输出到外设的信号与 EXTI_IMR 和 EXTI_EMR 寄存器的配置无关。详情参见 *ADC 配置寄存器 1（ADC_CFG1）*和 *ADC 注入序列段触发选择寄存器（ADC_JTRIGSEL）*。

## 10.7 EXTI 寄存器

EXTI 寄存器支持 32 位访问。

表 10-4 EXTI 基地址

| 外设 | 基地址 |
|---|---|
| EXTI | 0x4002 1800 |

### 10.7.1 EXTI 上升沿触发选择寄存器（EXTI_RTSR）

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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RT15 | RT14 | RT13 | RT12 | RT11 | RT10 | RT9 | RT8 | RT7 | RT6 | RT5 | RT4 | RT3 | RT2 | RT1 | RT0 |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15:0 | RTx | 设置EXTI可配置通道x（x＝0～15）为上升沿触发事件或中断。<br>0：禁止<br>1：使能 |

### 10.7.2 EXTI 下降沿触发选择寄存器（EXTI_FTSR）

偏移地址：0x04

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FT15 | FT14 | FT13 | FT12 | FT11 | FT10 | FT9 | FT8 | FT7 | FT6 | FT5 | FT4 | FT3 | FT2 | FT1 | FT0 |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15:0 | FTx | 设置EXTI可配置通道x（x＝0～15）为下降沿触发事件或中断。<br>0：禁止<br>1：使能 |

### 10.7.3 EXTI 中断挂起寄存器（EXTI_PIR）

偏移地址：0x08

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PIF15 | PIF14 | PIF13 | PIF12 | PIF11 | PIF10 | PIF9 | PIF8 | PIF7 | PIF6 | PIF5 | PIF4 | PIF3 | PIF2 | PIF1 | PIF0 |
| rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 | rc_w1 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15:0 | PIFx | 在EXTI可配置通道x（x＝0～15）检测到触发中断并挂起，表示中断待处理。<br>各个bit在相应通道检测到上升/下降沿中断时置1。<br>软件写1清零。<br>0：未检测到中断挂起请求。<br>1：中断请求已挂起。 |

### 10.7.4 EXTI I/O 选择寄存器 1（EXTI_EXTICR1）

偏移地址：0x50

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | EXTI7[1:0] | | Res. | | EXTI6[1:0] | | Res. | | EXTI5[1:0] | | Res. | | EXTI4[1:0] | |
| | | rw | rw | | | rw | rw | | | rw | rw | | | rw | rw |

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

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | rw | rw | | | rw | rw | | | rw | rw | | | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:30 | 保留 | 写入无效 |
| 29:28 | EXTI7[1:0] | EXTI7 GPIO端口选择<br>00：PA7 引脚<br>01：PB7 引脚<br>11：PF7 引脚<br>其他：保留（写入保留值，按00处理） |
| 27:26 | 保留 | 写入无效 |
| 25:24 | EXTI6[1:0] | EXTI6 GPIO端口选择<br>00：PA6 引脚<br>01：PB6 引脚<br>11：PF6 引脚<br>其他：保留（写入保留值，按00处理） |
| 23:22 | 保留 | 写入无效 |
| 21:20 | EXTI5[1:0] | EXTI5 GPIO端口选择<br>00：PA5 引脚<br>01：PB5 引脚<br>11：PF5 引脚<br>其他：保留（写入保留值，按00处理） |
| 19:18 | 保留 | 写入无效 |
| 17:16 | EXTI4[1:0] | EXTI4 GPIO端口选择<br>00：PA4 引脚<br>01：PB4 引脚<br>11：PF4 引脚<br>其他：保留（写入保留值，按00处理） |
| 15:14 | 保留 | 写入无效 |

| | | |
|---|---|---|
| 13:12 | EXTI3[1:0] | EXTI3 GPIO端口选择<br>00：PA3 引脚<br>01：PB3 引脚<br>11：PF3 引脚<br>其他：保留（写入保留值，按00处理） |
| 11:10 | 保留 | 写入无效 |
| 9:8 | EXTI2[1:0] | EXTI2 GPIO端口选择<br>00：PA2 引脚<br>01：PB2 引脚<br>11：PF2 引脚<br>其他：保留（写入保留值，按00处理） |
| 7:6 | 保留 | 写入无效 |
| 5:4 | EXTI1[1:0] | EXTI1 GPIO端口选择<br>00：PA1 引脚<br>01：PB1 引脚<br>11：PF1 引脚<br>其他：保留（写入保留值，按00处理） |
| 3:2 | 保留 | 写入无效 |
| 1:0 | EXTI0[1:0] | EXTI0 GPIO端口选择<br>00：PA0 引脚<br>01：PB0 引脚<br>11：PF0 引脚<br>其他：保留（写入保留值，按00处理） |

### 10.7.5 EXTI I/O 选择寄存器 2（EXTI_EXTICR2）

偏移地址：0x54

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | EXTI15[1:0] | | Res. | | EXTI14[1:0] | | Res. | | EXTI13[1:0] | | Res. | | EXTI12[1:0] | |
| | | rw | rw | | | rw | rw | | | rw | rw | | | rw | rw |

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:30 | 保留 | 写入无效 |
| 29:28 | EXTI15[1:0] | EXTI15 GPIO端口选择<br>00：PA15 引脚<br>01：PB15 引脚<br>10：PC15 引脚<br>其他：保留（写入保留值，按00处理） |
| 27:26 | 保留 | 写入无效 |
| 25:24 | EXTI14[1:0] | EXTI14 GPIO端口选择<br>00：PA14 引脚<br>01：PB14 引脚<br>10：PC14 引脚<br>其他：保留（写入保留值，按00处理） |
| 23:22 | 保留 | 写入无效 |
| 21:20 | EXTI13[1:0] | EXTI13 GPIO端口选择<br>00：PA13 引脚<br>01：PB13 引脚<br>10：PC13 引脚<br>其他：保留（写入保留值，按00处理） |
| 19:18 | 保留 | 写入无效 |
| 17:16 | EXTI12[1:0] | EXTI12 GPIO端口选择<br>00：PA12 引脚<br>01：PB12 引脚<br>其他：保留（写入保留值，按00处理） |
| 15:14 | 保留 | 写入无效 |

| | | |
|---|---|---|
| 13:12 | EXTI11[1:0] | EXTI11 GPIO端口选择<br>00：PA11 引脚<br>01：PB11 引脚<br>其他：保留（写入保留值，按00处理） |
| 11:10 | 保留 | 写入无效 |
| 9:8 | EXTI10[1:0] | EXTI10 GPIO端口选择<br>00：PA10 引脚<br>01：PB10 引脚<br>其他：保留（写入保留值，按00处理） |
| 7:6 | 保留 | 写入无效 |
| 5:4 | EXTI9[1:0] | EXTI9 GPIO端口选择<br>00：PA9 引脚<br>01：PB9 引脚<br>其他：保留（写入保留值，按00处理） |
| 3:2 | 保留 | 写入无效 |
| 1:0 | EXTI8[1:0] | EXTI8 GPIO端口选择<br>00：PA8 引脚<br>01：PB8 引脚<br>其他：保留（写入保留值，按00处理） |

### 10.7.6 EXTI 中断唤醒屏蔽寄存器（EXTI_IMR）

偏移地址：0x70

复位值：0xE303 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IM31 | IM30 | IM29 | Res. | | | IM25 | IM24 | Res. | | | | | | IM17 | IM16 |
| rw | rw | rw | | | | rw | rw | | | | | | | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IM15 | IM14 | IM13 | IM12 | IM11 | IM10 | IM9 | IM8 | IM7 | IM6 | IM5 | IM4 | IM3 | IM2 | IM1 | IM0 |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:29 | IMx | 允许直接通道x的中断唤醒CPU（x = 29 ~ 31）<br>0：禁止<br>1：使能 |
| 28:26 | 保留 | 写入无效 |
| 25:24 | IMx | 允许直接通道x的中断唤醒CPU（x = 24 ~ 25）<br>0：禁止<br>1：使能 |
| 23:18 | 保留 | 写入无效 |
| 17:16 | IMx | 允许直接通道x的中断唤醒CPU（x = 16 ~ 17）<br>0：禁止<br>1：使能 |
| 15:0 | IMx | 允许可配置通道x的中断唤醒CPU（x = 0 ~ 15）<br>0：禁止<br>1：使能 |

### 10.7.7 EXTI 事件唤醒屏蔽寄存器（EXTI_EMR）

偏移地址：0x74

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EM31 | EM30 | EM29 | Res. | | | EM25 | EM24 | Res. | | | | | | EM17 | EM16 |
| rw | rw | rw | | | | rw | rw | | | | | | | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| EM15 | EM14 | EM13 | EM12 | EM11 | EM10 | EM9 | EM8 | EM7 | EM6 | EM5 | EM4 | EM3 | EM2 | EM1 | EM0 |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:29 | EMx | 允许直接通道x的事件唤醒CPU（x = 29 ~ 31）<br>0：禁止<br>1：使能 |

| | | |
|---|---|---|
| 28:26 | 保留 | 写入无效 |
| 25:24 | EMx | 允许直接通道x的事件唤醒CPU（x = 24 ~ 25）<br>0：禁止<br>1：使能 |
| 23:18 | 保留 | 写入无效 |
| 17:16 | EMx | 允许直接通道x的事件唤醒CPU（x = 16 ~ 17）<br>0：禁止<br>1：使能 |
| 15:0 | EMx | 允许可配置通道x的事件唤醒CPU（x = 0 ~ 15）<br>0：禁止<br>1：使能 |

# 11 循环冗余校验（CRC）

## 11.1 简介

CRC 根据初始值和多项式计算输入数据的 CRC 值。CRC 运算通常用于验证数据在传输及存储时的完整性。

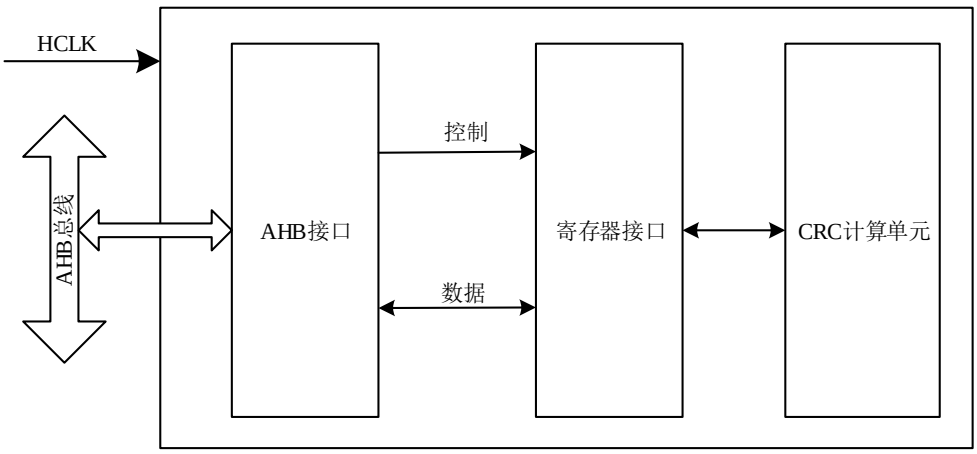
## 11.2 CRC 主要特性

- 支持 CRC-16 多项式 0x1021（$X^{16}+X^{12}+X^5+1$）
- 可配置初始值、输入/输出反转选择和输出结果异或选择，包含 CRC16-CCITT，CRC16-CCITT_false，CRC16-X25，CRC16-XMODEM 等算法
- 1 个 AHB 时钟周期完成 8 位 CRC 计算

## 11.3 CRC 功能描述

### 11.3.1 CRC 框图

图 11-1 CRC 框图

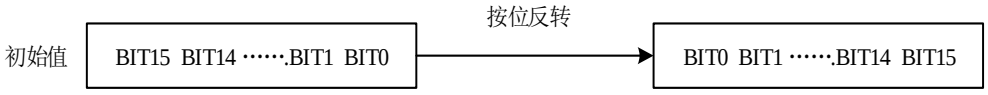



### 11.3.2 CRC 操作说明

CRC 的数据寄存器 CRC_DR 用于输入原始数据。CRC 的结果寄存器 CRC_RDR 用于输出计算生成的 CRC 值和输入初始值。CRC 计算流程见下图。

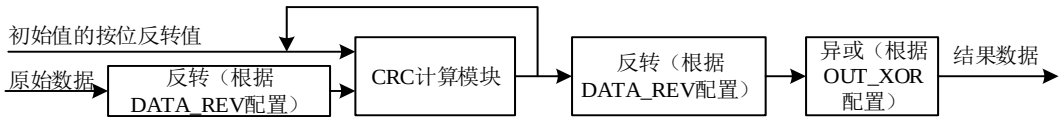
- 向 CRC_RDR 输入初始值时，需按位反转后写入；

图 11-2 反转框图




- 输入到 CRC_DR 的原始数据，以及计算生成的数据需要读出时，根据 CRC_CSR 中 DATA_REV 位进行反转选择；
- 计算生成的数据需要读出时，需根据 CRC_CSR 中 OUT_XOR 位与对应的数值异或。

图 11-3 CRC 计算单元框图




CRC 计算模式是指向 CRC 输入原始数据，然后获取 CRC 计算结果。操作步骤如下：

1) 根据算法需要，配置输入/输出数据反转选择和输出结果异或选择，写入控制状态寄存器 CRC_CSR 中 OUT_XOR 和 DATA_REV 位；
2) 将初始值的按位反转值写入结果寄存器 CRC_RDR；
3) 将待计算的原始数据按 8 位位宽方式，依次写入数据寄存器 CRC_DR；
4) 读取结果寄存器 CRC_RDR 中的 CRC 值。

## 11.4 CRC 寄存器

CRC 寄存器支持 32 位访问。

表 11-1 CRC 基地址

| 外设 | 基地址 |
|---|---|
| CRC | 0x4002 3000 |

### 11.4.1 CRC 控制状态寄存器（CRC_CSR）

偏移地址：0x00

复位值：0x0000 000C

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | DATA_REV | OUT_XOR | Res. | |
| | | | | | | | | | | | | rw | rw | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:4 | 保留 | 写入无效 |
| 3 | DATA_REV | 输入输出数据反转选择<br>0：输入数据和输出数据不反转<br>1：输入数据和输出数据反转 |
| 2 | OUT_XOR | 输出结果异或选择<br>0：输出结果与 0x0000 异或<br>1：输出结果与 0xFFFF 异或 |
| 1:0 | 保留 | 写入无效 |

### 11.4.2 CRC 结果寄存器（CRC_RDR）

偏移地址：0x04

复位值：0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15:0 | RESULT[15:0] | CRC 计算结果<br>读取 RESULT[15:0]以获取 CRC-16 的计算结果；<br>向 RESULT[15:0]写入初始值以初始化 CRC-16 计算。 |

### 11.4.3 CRC 数据寄存器（CRC_DR）

偏移地址：0x80

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 写入无效 |
| 7:0 | DATA[7:0] | 输入数据 |

# 12 硬件除法器（HWDIV）

## 12.1 简介

硬件除法器支持有符号或者无符号的 32 位整数除法运算。

## 12.2 HWDIV 主要特性

- 16 个 AHB 时钟周期完成一次有符号或无符号除法运算
- 32 位被除数，32 位除数，输出 32 位商和 32 位余数
- 除数为零警告标志位
- 写除数寄存器启动除法运算

## 12.3 HWDIV 功能描述

硬件除法器支持有符号或者无符号的 32 位除法运算，可通过 HWDIV 控制寄存器（HWDIV_CR）的 SIGN 位配置。有符号运算时，商和余数寄存器保存运算结果的补码。

硬件除法器包含 4 个 32 位数据寄存器，分别为被除数，除数，商和余数。每次写 HWDIV 除数寄存器（HWDIV_DVSR）会启动除法运算，通过查询 HWDIV 状态寄存器（HWDIV_SR）的 BUSY 标志判断是否运算结束。运算结束后，结果会写到 HWDIV 商寄存器（HWDIV_QUOTR）和 HWDIV 余数寄存器（HWDIV_RMDR）。

如果除数为零会产生除零警告标志位，HWDIV 状态寄存器（HWDIV_SR）的 DIVBY0 位置 1。

*注意：* *在运算期间写 HWDIV_CR、HWDIV_DVDR 或 HWDIV_DVSR 寄存器，会导致结果不可预知。*

## 12.4 HWDIV 寄存器

HWDIV 寄存器支持 32 位访问。

表 12-1 HWDIV 基地址

| 外设 | 基地址 |
|---|---|
| HWDIV | 0x4002 3800 |

### 12.4.1 HWDIV 控制寄存器（HWDIV_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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:1 | 保留 | 写入无效 |
| 0 | SIGN | 除法器符号选择位<br>0：无符号除法<br>1：有符号除法 |

### 12.4.2 HWDIV 状态寄存器（HWDIV_SR）

偏移地址：0x04

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | BUSY | DIVBY0 |
| | | | | | | | | | | | | | | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:2 | 保留 | 写入无效 |
| 1 | BUSY | 运算状态标志位<br>0：运算完成或空闲状态<br>1：运算正在进行 |
| 0 | DIVBY0 | 除数为零警告标志位<br>0：除数不为 0<br>1：除数为 0<br>*注意：写除数寄存器后自动更新该位。* |

### 12.4.3 HWDIV 被除数寄存器（HWDIV_DVDR）

偏移地址：0x08

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIVIDEND[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 |
| DIVIDEND[15:0] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | DIVIDEND[31:0] | 被除数 |

### 12.4.4 HWDIV 除数寄存器（HWDIV_DVSR）

偏移地址：0x0C

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIVISOR[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 |
| DIVISOR[15:0] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | DIVISOR[31:0] | 除数<br>写入该寄存器后启动除法运算。 |

### 12.4.5 HWDIV 商寄存器（HWDIV_QUOTR）

偏移地址：0x10

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| QUOTIENT[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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| QUOTIENT[15:0] | | | | | | | | | | | | | | | |
| r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | QUOTIENT[31:0] | 商 |

### 12.4.6 HWDIV 余数寄存器（HWDIV_RMDR）

偏移地址：0x14

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| REMAINDER[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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| REMAINDER[15:0] | | | | | | | | | | | | | | | |
| r | r | r | r | r | r | r | r | r | r | r | r | r | r | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | REMAINDER[31:0] | 余数 |

# 13 硬件开方器（SQRT）

## 13.1 简介

硬件开方器支持 32 位无符号数开根号运算。

## 13.2 SQRT 主要特性

- 16 个 AHB 时钟周期完成一次无符号整数开方运算
- 32 位无符号被开方整数，16 位开方根输出
- 写入被开方数寄存器启动开方运算

## 13.3 SQRT 功能描述

硬件开方器支持 32 位无符号数开根号运算，包括 1 个 32 位的被开方数寄存器和 1 个 16 位的开方根寄存器。每次写入 SQRT 被开方数寄存器（SQRT_SQR）会启动开方运算，通过查询 SQRT 状态寄存器（SQRT_SR）的 BUSY 标志判断是否运算结束。运算结束后，结果会写到 SQRT 开方根寄存器（SQRT_RDR）。

*注意：　在运算期间写 SQRT_SQR 寄存器，会导致结果不可预知。*

## 13.4 SQRT 寄存器

SQRT 寄存器支持 32 位访问。

表 13-1 SQRT 基地址

| 外设 | 基地址 |
|---|---|
| SQRT | 0x4002 4000 |

### 13.4.1 SQRT 状态寄存器（SQRT_SR）

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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:1 | 保留 | 写入无效 |
| 0 | BUSY | 运算状态标志位<br>0：运算完成或空闲状态<br>1：运算正在进行 |

### 13.4.2 SQRT 被开方数寄存器（SQRT_SQR）

偏移地址：0x04

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SQUARE[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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SQUARE[15:0] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | SQUARE[31:0] | 被开方数<br>写入该寄存器后启动开方运算。 |

### 13.4.3 SQRT 开方根寄存器（SQRT_RDR）

偏移地址：0x08

复位值：0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15:0 | ROOT[15:0] | 开方根 |

# 14 模数转换器（ADC）

## 14.1 简介

芯片内置 12 位逐次逼近型模数转换器（SAR ADC），可对 15 路外部通道和 1 路内部通道的电压信号进行模数转换，支持三种转换模式：单次扫描、循环扫描或循环间断转换模式，每种转换模式下有两种序列转换方式：规则序列转换、注入序列转换。ADC 转换结果存储（左对齐/右对齐）在 16 位数据寄存器中。

ADC 具有模拟看门狗功能，用于监控转换通道的输入电压是否超出用户设定阈值的上限或下限。

## 14.2 ADC 主要特性

- 工作电压 2.0 V~ 5.5V
- 参考电压 $V_{REF_ADC}$：
  - $V_{DDA}$；
  - 内置参考电压源 VREFBUF：2.048V、2.5V、3.6V、4.2V。
- 输入信号幅度 0 ~ $V_{REF_ADC}$
- 12 位分辨率最高采样率 1.125Msps
- 模拟输入通道
  - 15 路外部输入通道，可测量高输出阻抗信号；
  - 1 路内部输入通道：$V_{BGR}$ 或 $V_{DDA}$/3 或 $V_{TS}$。
- 转换模式
  - 单次扫描模式
  - 循环扫描模式
  - 循环间断模式
- 可配置通道序列转换方式：规则序列转换、注入序列转换
- 注入序列支持单段、双段、三段、四段自定义采样序列
- 启动方式，注入序列的四段触发和规则序列触发单独可配

- 软件触发；
- 硬件触发：

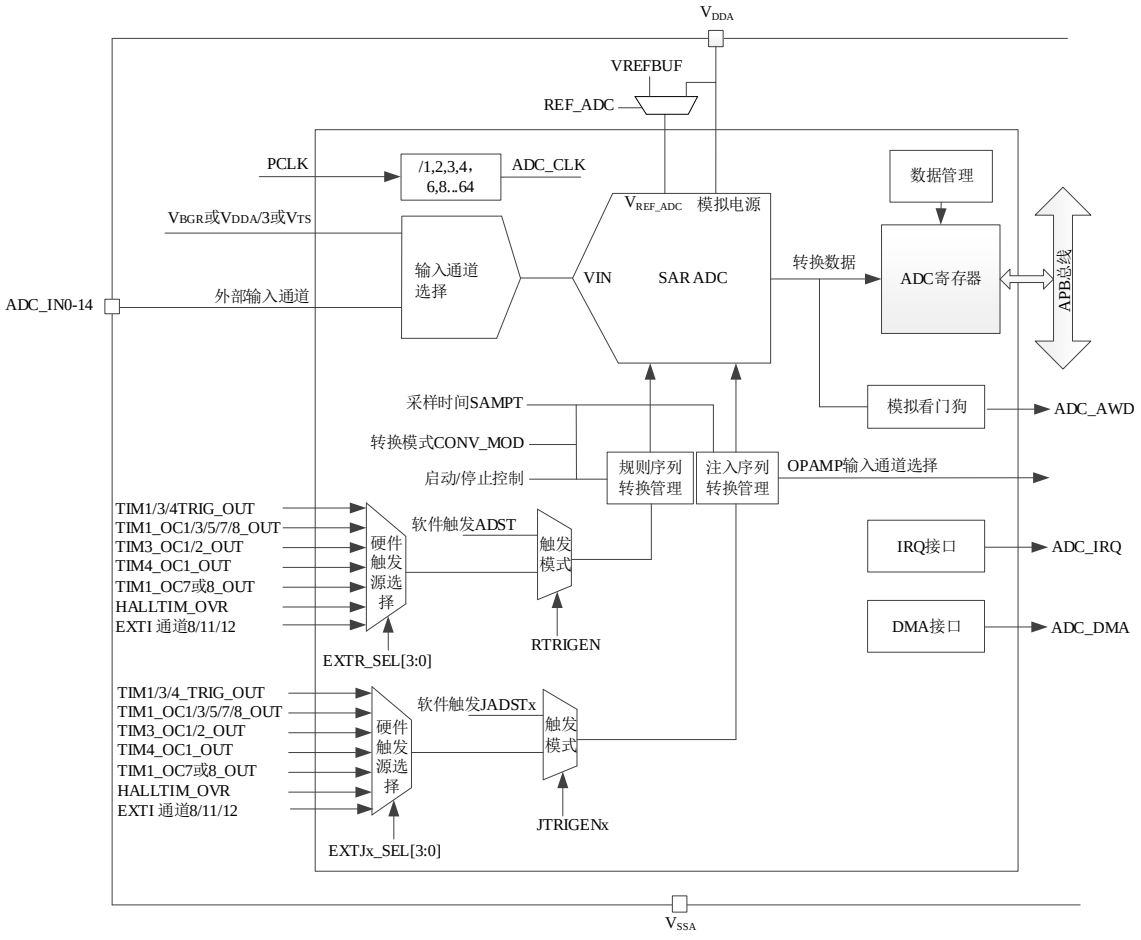
  触发源为内部定时器事件或 GPIO 输入事件；

  触发极性可配置（上升沿、下降沿、双沿）。

- 规则序列和注入序列可独立配置采样时间
- 可配置数据对齐方式：左对齐、右对齐
- 模拟看门狗
- ADC 注入通道支持 OPAMP2 输入通道切换
- 支持 DMA

## 14.3 ADC 功能描述

图 14-1 ADC 结构框图



### 14.3.1 ADC 引脚和内部信号

表 14-1 ADC 输入引脚

| 引脚名称 | 引脚类型 | 说明 |
|---|---|---|
| $V_{DDA}$ | 输入，模拟电源 | ADC 模拟电源 |
| | 输入，参考电压源 | ADC 参考电压源 |
| $V_{SSA}$ | 输入，模拟地 | ADC 模拟地 |
| ADC_INx (x = 0~14) | 模拟输入 | 15 路外部输入通道 |

表 14-2 ADC 内部信号

| 信号名称 | 信号类型 | 说明 |
|---|---|---|
| $V_{TS}$ | 输入 | 片内温度传感器输出电压 |
| $V_{BGR}$ | 输入 | 内部带隙基准电压 |
| $V_{DDA}/3$ | 输入 | 电源电压分压 |
| VREFBUF | 输入 | ADC 参考电压源 |

### 14.3.2 时钟源

ADC 工作时钟 ADC_CLK 源于 APB 总线时钟 PCLK，并可以通过 ADC_CFG2 寄存器中 PRESC[3:0]位域进行 1、2、3、4、6、8、12、16、18、24、32、36、64 分频。ADC 工作时钟频率最大支持 18MHz。

ADC 采样速率 $f_S$ 与工作电压之间的关系如下：

- 2.4V ＜ $V_{DDA}$ ≤ 5.5V 时，$f_S$ ≤ 1.125Msps
- 2.0 V ≤ $V_{DDA}$ ≤ 2.4V 时，$f_S$ ≤ 500Ksps

*注意：* *ADC 时钟 ADC_CLK 要求占空比 50%（典型值，占空比范围为 40%~60%）。为此当 PRESC[3:0]选择 1 分频作为 ADC 时钟源时，需要选择占空比为 50%的系统时钟并禁止 AHB 和 APB 预分频器对系统时钟进行分频。*

### 14.3.3 使能/禁止控制

使用 ADC 之前需使能 BGR，并等待其稳定（$t_{START}$ 启动稳定时间参见数据手册），详见*内部带隙基准电压*。

ADC 使能、禁止由 ADC_CR 寄存器的 ADEN 和 ADDIS 位控制：

使能 ADC 步骤：

1) 将 ADC_CR 寄存器中 ADEN 位置 1；

2) 等待 ADC 启动稳定，启动稳定时间 1μs。

禁止 ADC 步骤：

1) 查询 ADC_CR 寄存器中 ADST 和 JADSTx 位均为 0，以确认当前未执行任何转换；若 ADST 或 JADSTx 位不为 0，可向 ADC_CR 寄存器 STOP 位写 1，并等待 ADST 和 JADSTx 位清 0，以停止正在进行的转换；
2) 将 ADC_CR 寄存器中 ADDIS 位写 1；
3) 查询 ADC_CR 寄存器中 ADEN 位为 0，确认 ADC 已禁止，此时所有标志清 0。

### 14.3.4 转换通道

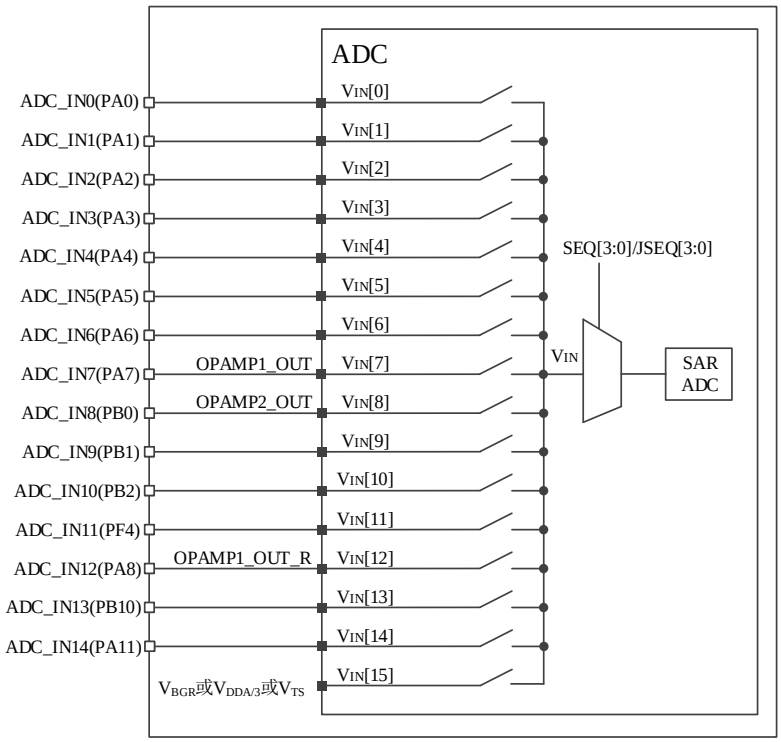
ADC 模块具有 16 路输入通道：

- 15 路外部输入通道，通过 GPIO 引脚输入（ADC_INx），其中 ADC_IN7/12 连接 OPAMP1 的输出信号，ADC_IN8 连接 OPAMP2 的输出信号，ADC_IN14 连接 OPAMP1/2 输出；
- 1 路内部输入通道（$V_{BGR}$ 或 $V_{DDA}/3$ 或 $V_{TS}$），内部带隙基准电压 $V_{BGR}$ 、$V_{DDA}/3$ 和温度传感器输出电压 $V_{TS}$ 连接至通道 ADC_VIN[15]。

*注意：* *$V_{BGR}$，$V_{DDA}/3$ 和 $V_{TS}$ 共用同一内部通道，应避免同时使能，操作流程具体参见对应小节介绍。*

ADC 输入通道见下图：

图 14-2 ADC 输入通道




**转换通道选择**

每个通道可配置为规则序列和注入序列。

规则序列由多次转换组成，最多可转换 12 个通道。ADC_RSEQx 寄存器指定规则通道和规则通道的转换顺序。ADC_CFG1 寄存器中 LEN[3:0]位域指定规则通道序列长度，仅当 ADST=0 时才可更改 ADC_RSEQx 寄存器。

注入序列可配置为单段、双段、三段、四段注入序列，ADC_CFG3 寄存器的 EXTJx_EN 位置 1 可分别使能注入序列第 x 段的转换。每段注入序列由多次转换组成，最多可转换 4 个通道，由 ADC_JSETSEL 寄存器的 JSETx_SEL[3:0]位域配置。由 ADC_JSEQ 寄存器指定注入通道。

规则序列和注入序列转换优先级：

- 注入序列可打断规则序列优先进行转换；
- 注入序列不同段之间触发事件优先级一致，按照触发事件顺序先后启动转换，如果在一段转换过程中，产生的新的触发事件被忽略（其他段的触发事件、当前段的触发事件）；当不同段触发事件同时产生时，段号较小的优先处理，段号较大的触发事件被忽略。

### 14.3.5 单次扫描转换模式

ADC_CFG1 寄存器中 CONV_MOD[1:0]位域设置为“00”，选择单次扫描转换模式。该模式下转换行为如下：

- 规则序列转换时（JEN=0），当软件触发或外部硬件触发事件产生时，ADC 对所配置规则序列通道只进行一轮转换。规则序列转换结束后，ADC 停止工作，等待下一次触发事件产生。

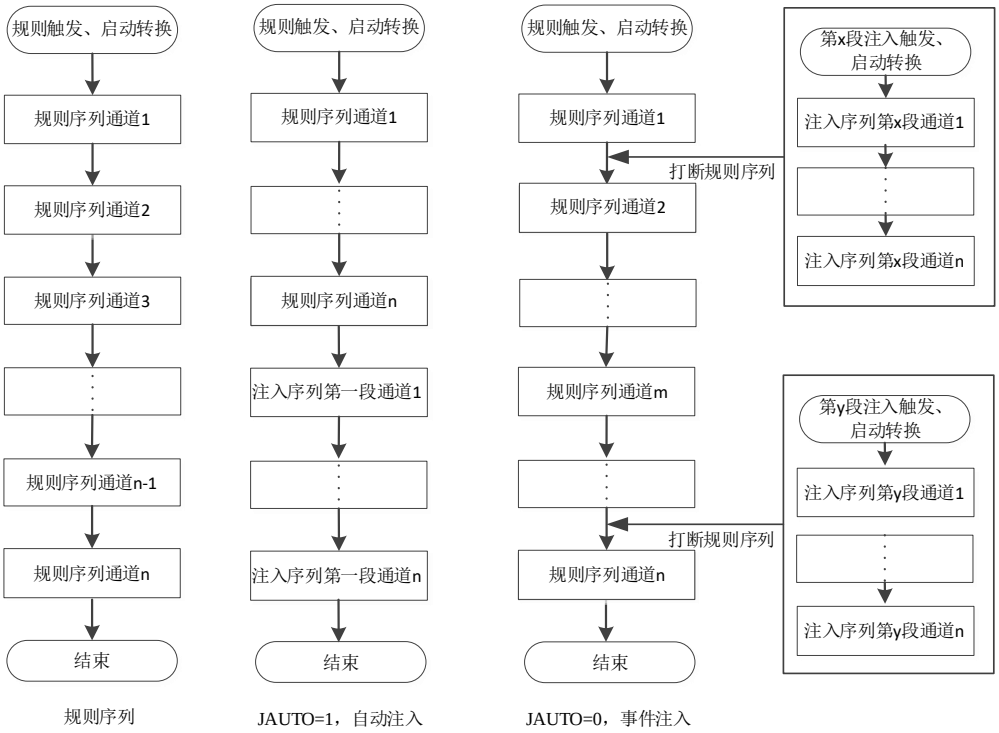
- 注入序列选择自动注入模式时（JEN=1 且 JAUTO=1），注入序列仅第一段有效，其他段被忽略，配置参见*表：ADC 自动注入序列配置*；使用规则序列触发事件，无需配置 ADC_CFG3 寄存器的 EXTJx_EN 位。当规则触发事件产生时，会启动规则序列转换，配置的注入序列第一段的通道将在规则序列转换结束之后自动转换，最多可以转换多达 12+4 个通道。配置的规则序列和注入序列的通道全部转换完成之后 ADC 停止工作，等待下一次规则序列触发事件产生。

表 14-3 ADC 自动注入序列配置

| 注入阶段需要转换序列数 | JSET1_SEL[3:0] |
|---|---|
| 1 | 0001 |
| 2 | 0011 |
| 3 | 0111 |
| 4 | 1111 |

- 注入序列选择事件注入模式时（JEN=1 且 JAUTO=0），当第 x 段软件触发或外部硬件触发事件产生时，ADC 对所配置注入序列第 x 段的通道进行一轮转换，可打断规则序列优先进行。注入序列转换结束后，ADC 停止工作或回到规则序列转换，下一次注入触发事件产生时可进行新一轮的注入序列转换。

图 14-3 单次扫描模式下的规则序列和注入序列转换示意图




## 规则序列转换

规则序列启动转换方式如下：

- 软件触发，将 ADC_CR 寄存器中 ADST 位置 1，立即启动转换；
- 硬件触发，ADC_CR 寄存器中 ADST 位置 1 后，当外部硬件触发事件产生，启动转换。

每个通道转换完成后，ADC 执行以下操作：

- 转换数据存储在 ADC_DR 寄存器；
- 规则序列通道转换结束标志（EOC）置 1；
- 如果规则序列通道转换结束中断使能（EOCIE 置 1），则产生中断。

在规则序列的所有通道转换完成后，ADC 执行以下操作：

- 规则序列转换结束标志（EOS）置 1；
- 如果规则序列转换结束中断使能（EOSIE 置 1），则产生中断；
- 如果选择软件触发启动方式（RTRIGEN[1:0]=00），ADST 硬件清 0，等待软件触发 ADST 位再次置 1。
- 如果选择外部硬件触发启动方式（RTRIGEN[1:0]≠00），ADST 不清 0，等

待下一次外部硬件触发事件产生。

**注入序列转换**

ADC_CFG3 寄存器的 JEN 位置 1 可使能注入序列转换。JAUTO=0 时为事件注入模式，JAUTO=1 时为自动注入模式。

事件注入模式时，注入序列启动转换方式如下：

- 软件触发，将 ADC_CR 寄存器中 JADSTx 位置 1，立即启动第 x 段转换，如果此时有规则序列通道转换，可打断规则序列；
- 硬件触发，ADC_CR 寄存器中 JADSTx 位置 1 后，当第 x 段的外部硬件触发事件产生，启动第 x 段的转换，如果此时有规则序列通道转换，可打断规则序列。

每个通道转换完成后，ADC 执行以下操作：

- 转换数据存储在每个通道独立的 ADC_JDRx 寄存器；
- 注入序列通道转换结束标志（JEOC）和各通道独立的 JEOC_SEQx 置 1；
- 如果注入通道转换结束中断使能（JEOCIE 或 JEOC_SEQxIE 置 1），则产生相应中断。

在注入序列第 x 段的所有通道转换完成后，ADC 执行以下操作：

- 注入序列第 x 段序列转换结束标志（JEOSx）置 1；
- 如果注入序列第 x 段序列转换结束中断使能（JEOSxIE 置 1），则产生中断。
- 如果选择软件触发启动方式（JTRIGENx[1:0]=00），JADSTx 硬件清 0，等待下一次软件触发 JADSTx。
- 如果选择外部硬件触发启动方式（JTRIGENx[1:0]≠00），JADSTx 不清 0，等待下一次外部硬件触发事件产生。
- 如果注入启动转换时打断了规则序列转换，则继续恢复规则序列转换。

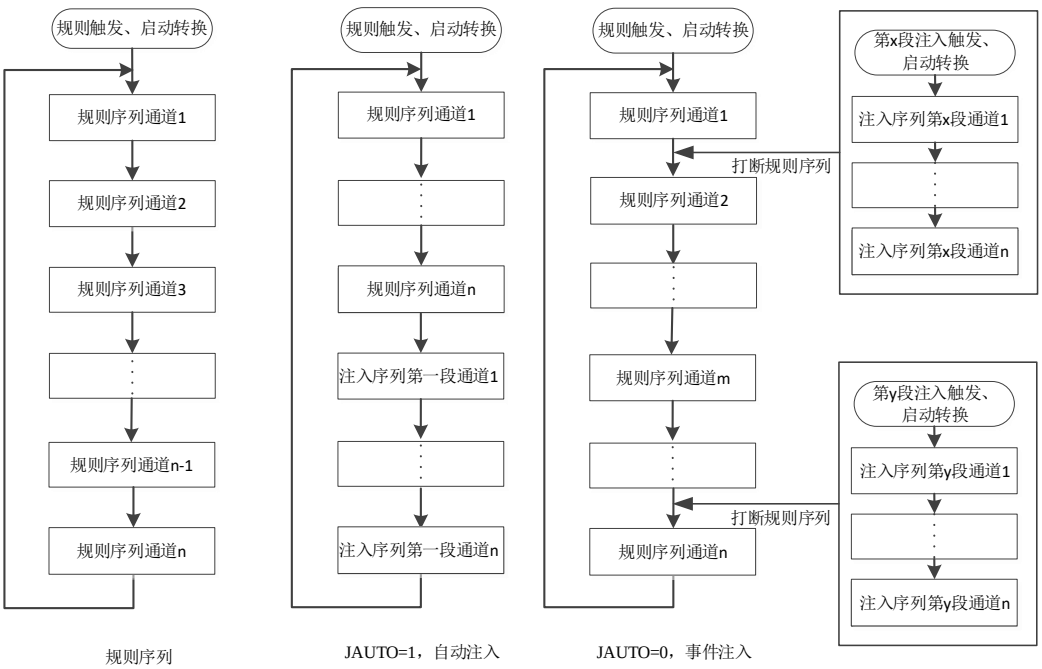
自动注入模式时，此时注入序列的触发事件将不被识别。此时启动转换方式、通道转换完成后 ADC 行为和序列转换结束后 ADC 行为，参见*规则序列转换*流程，自动注入下配置的注入序列转换完成时，置位注入序列相对应的标志，数据存放在 ADC_JDRx 寄存器。

### 14.3.6 循环扫描转换模式

ADC_CFG1 寄存器中 CONV_MOD[1:0]位域设置为“01”，选择循环扫描转换模式。该模式下转换行为如下：

- 规则序列转换时（JEN=0），当软件触发或外部硬件触发事件产生时，ADC 对所配置规则序列通道进行循环转换。规则通道序列转换一轮结束后，ADC 自动循环执行相同的转换序列，直至软件向 STOP 写 1。
- 注入序列选择自动注入模式时（JEN=1 且 JAUTO=1），注入序列仅第一段有效，其他段被忽略，配置参见*表：ADC 自动注入序列配置*。当规则触发事件产生时，会启动规则序列转换，配置的注入序列第一段的通道将在规则序列转换结束之后自动转换，最多可以转换多达 12+4 个通道。配置的规则序列和注入序列的通道全部转换完成之后 ADC 又会从头开启规则序列的转换，循环往复，直至软件向 STOP 写 1。
- 注入序列选择事件注入模式时（JEN=1 且 JAUTO=0），当第 x 段软件触发或外部硬件触发事件产生时，ADC 对所配置第 x 段的注入序列通道进行一轮转换，可打断规则序列优先进行。注入序列转换结束后，ADC 停止工作或回到规则序列转换，下一次注入触发事件产生时可进行新一轮的注入序列转换。

图 14-4 循环扫描模式下的规则序列和注入序列转换示意图




#### 规则序列转换

规则序列启动转换方式如下：

- 软件触发，将 ADC_CR 寄存器中 ADST 位置 1，立即启动转换；
- 硬件触发，ADC_CR 寄存器中 ADST 位置 1 后，当外部硬件触发事件产生，启动转换。

每个规则通道转换完成后，ADC 执行以下操作：

- 转换数据存储在 ADC_DR 寄存器；
- 规则序列通道转换结束标志（EOC）置 1；
- 如果规则序列通道转换结束中断使能（EOCIE 置 1），则产生中断。

在每次规则序列中所有通道转换完成后，ADC 执行以下操作：

- 规则序列转换结束标志（EOS）置 1；
- 如果规则序列转换结束中断使能（EOSIE 置 1），则产生中断。
- 无论选择软件触发启动方式（RTRIGEN[1:0]=00）还是外部硬件触发启动方式（RTRIGEN[1:0]≠00），ADST 均不清 0，ADC 继续对相同的序列通道进行转换。

**注入序列转换**

ADC_CFG3 寄存器的 JEN 位置 1 可使能注入序列转换。JAUTO=0 时为事件注入模式，JAUTO=1 时为自动注入模式。

事件注入模式的转换流程和相关标志与单次扫描转换模式下事件注入模式一致，详见单次扫描转换模式对应章节描述。

自动注入模式时，此时注入序列的触发事件将不被识别。此时启动转换方式、通道转换完成后 ADC 行为和序列转换结束后 ADC 行为，参见*规则序列转换*流程，自动注入下配置的注入序列转换完成时，注入序列相应标志会置位，数据存放在 ADC_JDRx 寄存器。

### 14.3.7 循环间断转换模式

ADC_CFG1 寄存器中 CONV_MOD[1:0]位域设置为“10”，选择循环间断转换模式。该模式下转换行为如下：

- 仅规则序列转换时（JEN=0），可通过 ADC_CFG2 寄存器的 DIS_SEL[1:0]

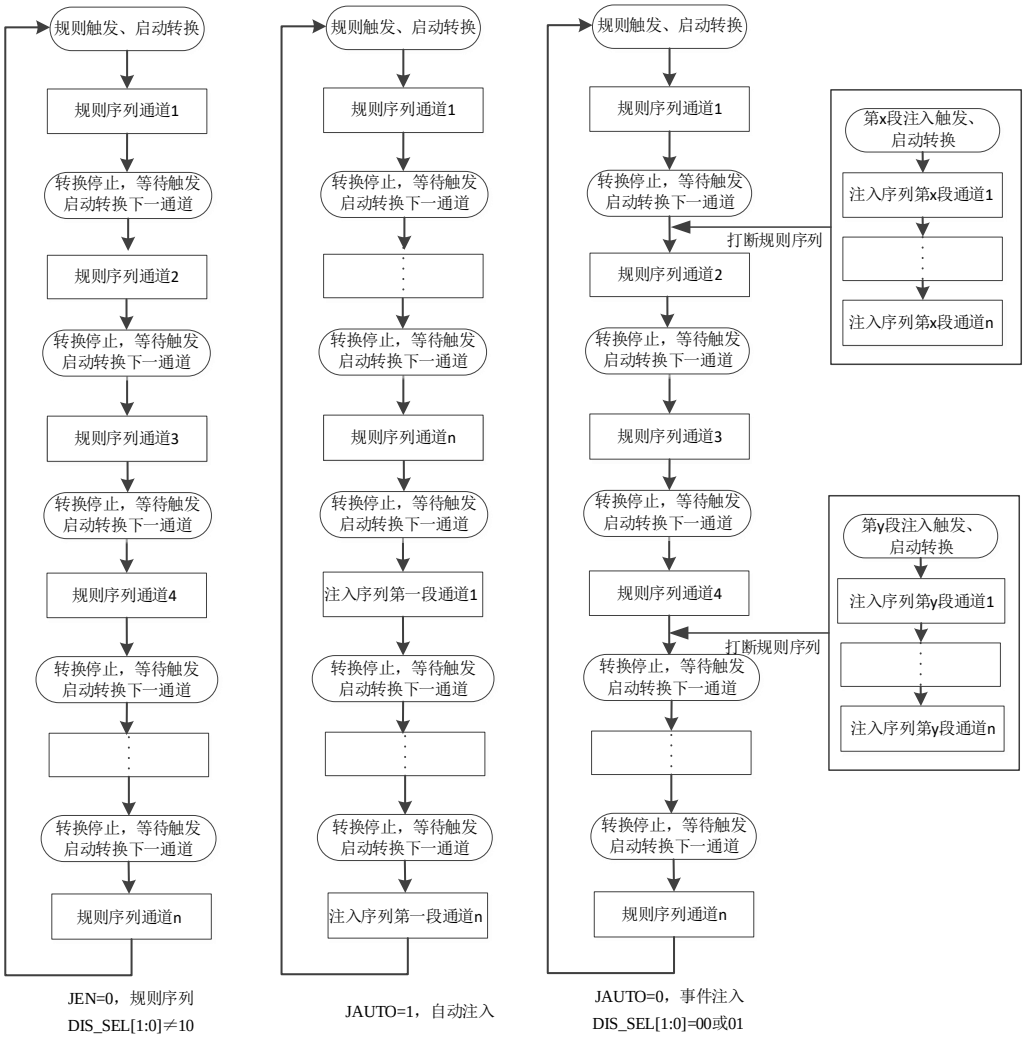
位域选择间断配置：

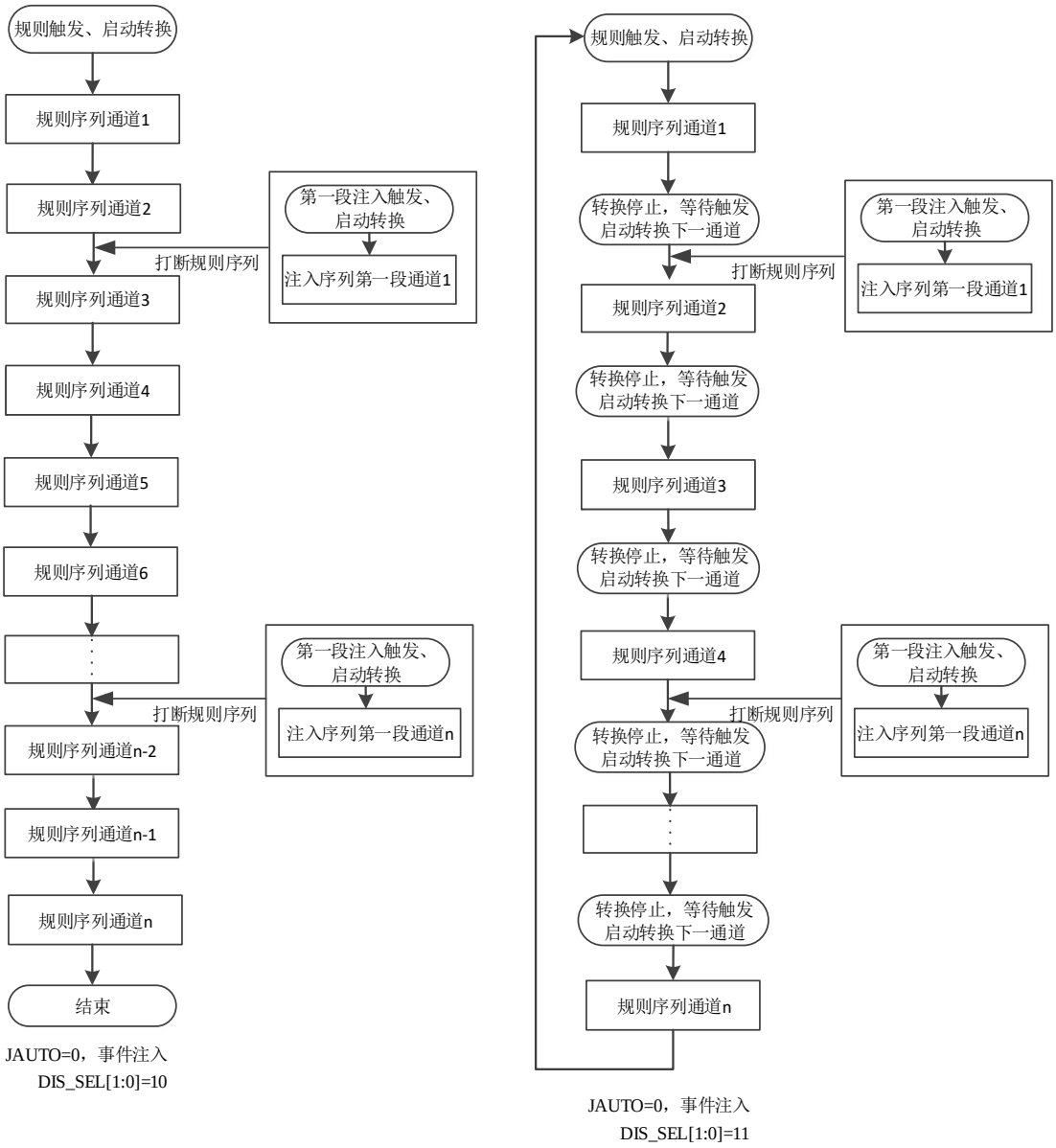
- DIS_SEL[1:0]为非 10 时，规则序列转换有间断效果。当软件触发或外部硬件触发事件产生时，ADC 对所配置规则序列通道只进行一个通道转换。当该通道转换完成，ADC 停止转换，等待下一次触发事件，之后进行下一通道的转换。规则序列转换完成后，ADC 执行相同的转换序列；
- DIS_SEL[1:0]为 10 时，规则序列转换没有间断效果，将以单次扫描的方式转换。

- 注入序列选择自动注入模式时（JEN=1 且 JAUTO=1），注入序列仅第一段有效，其他段被忽略，配置参见*表：ADC 自动注入序列配置*。当规则触发事件产生时，会启动规则序列转换，配置的注入序列第一段的通道将在规则序列转换结束之后自动转换，最多可以转换多达 12+4 个通道。无论规则序列还是注入序列，每次规则触发仅转换一个通道后 ADC 就停止工作，等待下一次触发。注入序列所有的通道一轮转换完成之后 ADC 继续按照相同序列进行间断转换，循环往复。

- 注入序列选择事件注入模式时（JEN=1 且 JAUTO=0），可通过 ADC_CFG2 寄存器的 DIS_SEL[1:0]位域选择间断配置。
  - DIS_SEL[1:0]为 00 或者 01 时，仅规则序列转换时有间断效果。每次规则触发仅转换一个规则序列通道，当注入序列第 x 段触发事件产生时，注入序列第 x 段将打断规则序列并完成一轮注入序列转换后再继续规则序列。规则序列转换完成后，ADC 继续按照规则序列进行间断转换；
  - DIS_SEL[1:0]为 10 时，仅注入序列转换时有间断效果，此时注入序列仅第一段有效，其他段被忽略，规则序列将以单次扫描的方式转换。当注入触发事件产生时，注入序列打断规则序列转换，且每次注入触发事件按照注入序列次序转换一个注入序列通道；
  - DIS_SEL[1:0]为 11 时，规则序列和注入序列转换时均有间断效果，此时注入序列仅第一段有效，其他段被忽略。每次规则触发事件产生时转换一个规则序列通道，每次注入触发事件产生时转换一个注入序列通道，循环往复。

图 14-5 循环间断模式下的注入序列转换示意图






## 规则序列转换

规则序列启动转换方式如下：

- 软件触发，将 ADC_CR 寄存器中 ADST 位置 1，立即启动转换；
- 硬件触发，ADC_CR 寄存器中 ADST 位置 1 后，当外部硬件触发事件产生，启动转换。

每个规则通道转换完成后，ADC 执行以下操作：

- 转换数据存储在 ADC_DR 寄存器；
- 规则序列通道转换结束标志（EOC）置 1；
- 如果规则序列通道转换结束中断使能（EOCIE 置 1），则产生中断。

- 如果选择软件触发启动方式（RTRIGEN[1:0]=00），ADST 硬件清 0，等待下一次软件触发 ADST。
- 如果选择外部硬件触发启动方式（RTRIGEN[1:0]≠00），ADST 不清 0，等待下一次外部硬件触发事件产生。

在每次规则序列中所有通道转换完成后，ADC 执行以下操作：

- 规则序列转换结束标志（EOS）置 1；
- 如果规则序列转换结束中断使能（EOSIE 置 1），则产生中断。
- 如果选择软件触发启动方式（RTRIGEN[1:0]=00），ADST 硬件清 0，等待下一次软件触发 ADST 后开启相同序列转换。
- 如果选择外部硬件触发启动方式（RTRIGEN[1:0]≠00），ADST 不清 0，等待下一次外部硬件触发事件产生后开启相同序列转换。

**注入序列转换**

ADC_CFG3 寄存器的 JEN 位置 1 可使能注入序列转换。JAUTO=0 时为事件注入模式，JAUTO=1 时为自动注入模式。

事件注入模式时，注入序列启动转换方式如下：

- 软件触发，将 ADC_CR 寄存器中 JADSTx 位置 1，立即启动第 x 段转换，如果此时有规则序列通道转换，打断规则序列；
- 硬件触发，ADC_CR 寄存器中 JADSTx 位置 1 后，当第 x 段外部硬件触发事件产生，启动第 x 段转换，如果此时有规则序列通道转换，打断规则序列。

每个通道转换完成后，ADC 执行以下操作：

- 转换数据存储在每个序列通道独立的 ADC_JDRx 寄存器；
- 注入序列通道转换结束标志（JEOC）和各序列通道独立的 JEOC_SEQx 置 1；
- 如果转换通道结束中断使能（JEOCIE 或 JEOC_SEQxIE 置 1），则产生相应中断；
- 如果选择软件触发启动方式（JTRIGENx[1:0]=00）且注入序列上有间断效果（DIS_SEL[1:0]=10 或 11），JADSTx 硬件清 0，回到规则序列转换并等

待下一次软件触发事件产生；

- 如果选择外部硬件触发启动方式（JTRIGENx[1:0]≠00）且注入序列上有间断效果（DIS_SEL[1:0]=10 或 11），JADSTx 不清 0，回到规则序列转换并等待下一次外部硬件触发事件产生；
- 如果注入序列上没有间断效果（DIS_SEL[1:0]=00 或 01），继续下一个注入序列通道转换。

在注入序列的第 x 段的所有通道转换完成后，ADC 执行以下操作：

- 注入序列第 x 段序列转换结束标志（JEOSx）置 1；
- 如果注入序列第 x 段序列转换结束中断使能（JEOSxIE 置 1），则产生中断。
- 如果选择软件触发启动方式（JTRIGENx[1:0]=00），JADSTx 硬件清 0，回到规则序列转换并等待下一次软件触发事件产生；
- 如果选择外部硬件触发启动方式（JTRIGENx[1:0]≠00），JADSTx 不清 0，回到规则序列转换并等待下一次外部硬件触发事件产生。

自动注入模式时，此时注入序列的触发事件将不被识别。此时启动转换方式、通道转换完成后 ADC 行为和序列转换结束后 ADC 行为，参见*规则序列转换*流程，自动注入下配置的注入序列转换完成时，置位注入序列相对应的标志，数据存放在 ADC_JDRx 寄存器。

### 14.3.8 启动/停止转换

**启动转换**

ADC 规则序列和注入序列启动方式独立可配置，支持软件触发方式和外部硬件触发方式，具体配置如下：

- 配置为软件触发方式（RTRIGEN[1:0]=00 或 JTRIGENx[1:0]=00），ADC_CR 寄存器中 ADST 或 JADSTx 位置 1 后立即启动转换；
- 配置为外部硬件触发方式（RTRIGEN[1:0]≠00 或 JTRIGENx[1:0]≠00），ADC_CR 寄存器中 ADST 或 JADSTx 位置 1 后，根据所配置触发极性，在下一个触发事件产生时，启动转换。触发事件通过 ADC_CFG1 寄存器中 EXTR_SEL[3:0]位域和 ADC_CFG3 寄存器 EXTJx_SEL[3:0]位域选择。

当 ADC 正处于转换的过程中，除注入序列打断规则序列以外的情况，任何触发

事件都会被忽略。如果寄存器位 ADST＝0，则会忽略任何规则序列外部硬件触发事件。如果寄存器位 JADSTx＝0，则会忽略任何注入序列外部硬件触发事件。

ADC_CR 寄存器中 ADST 和 JADSTx 位也用于指示当前 ADC 是否正在处于工作状态。ADST 或 JADSTx 位置 1 指示当前 ADC 处于工作状态；当 ADST 和 JADSTx 位清 0 时，指示 ADC 处于空闲状态，此时可以重新配置 ADC。

**停止转换**

可通过将 ADC_CR 寄存器中 STOP 位写入 1，停止任何正在进行的转换。这样可以复位 ADC 操作，使 ADC 处于空闲状态，从而准备好进行新的操作。

STOP 位写入 1 后，正在进行的规则序列转换或注入序列转换会被停止，当前转换结果会被丢弃（ADC_DR 或 ADC_JDRx 寄存器不会更新为当前转换结果）。通道扫描序列也会终止并复位，这意味着重启 ADC 将会重新开始新的转换序列。

一旦停止转换过程完成，ADC_CR 寄存器中 ADST 和 JADSTx 位会被硬件清 0，规则序列和注入序列当前转换通道所有标志不再置起。软件必须等待 ADST 和 JADSTx 清 0，然后才能开始进行新的转换。

图 14-6 停止正在进行的转换




### 14.3.9 转换时序示例

以下以规则序列通道 0、3、8，注入序列选取第一段触发且 JSET1_SEL[3:0]配置为“0111”，通道 1、2、6 为例，展示 ADC 转换时序。

图 14-7　单次扫描转换模式，规则软件触发，注入外部事件触发

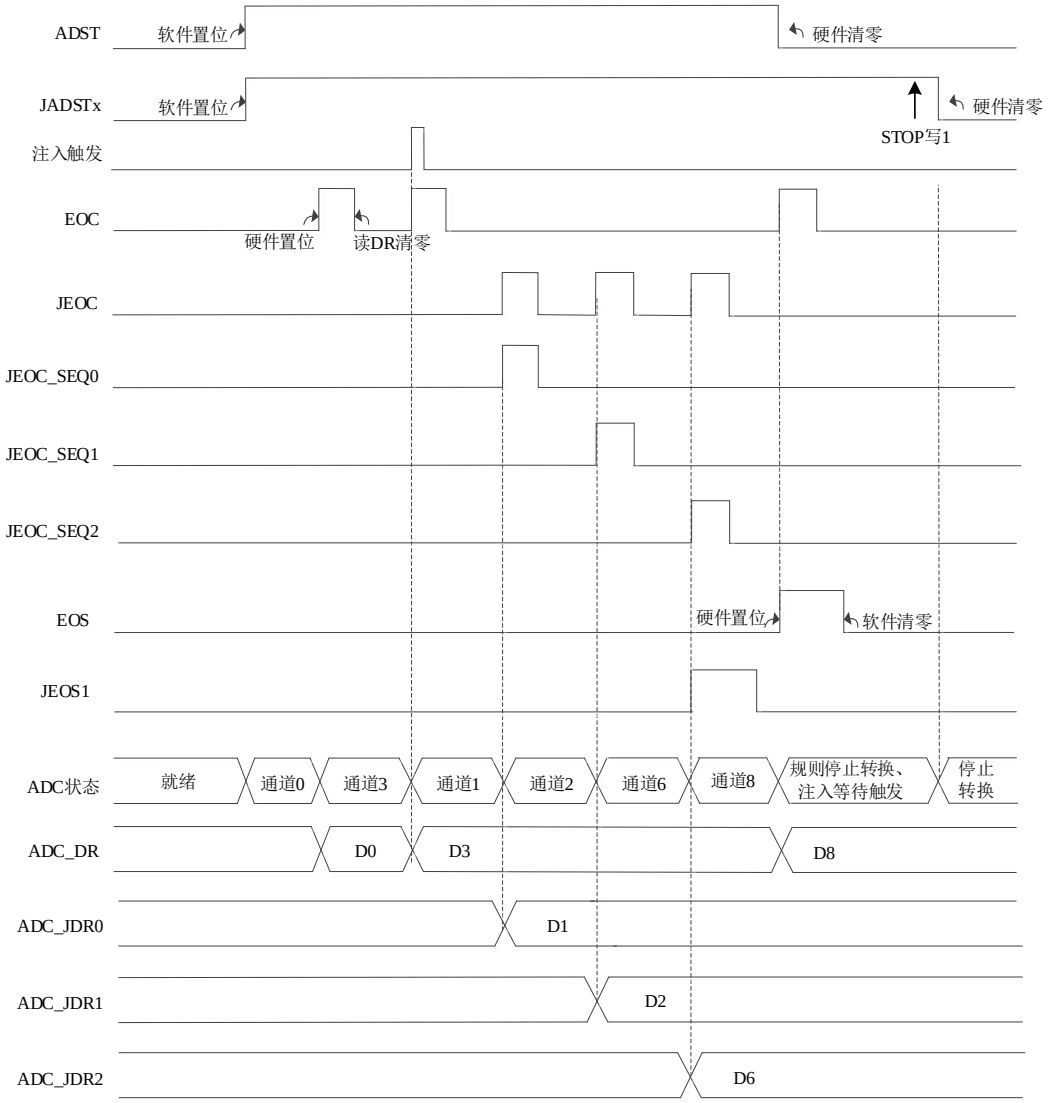


图 14-8 循环扫描转换模式，软件触发，使能自动注入

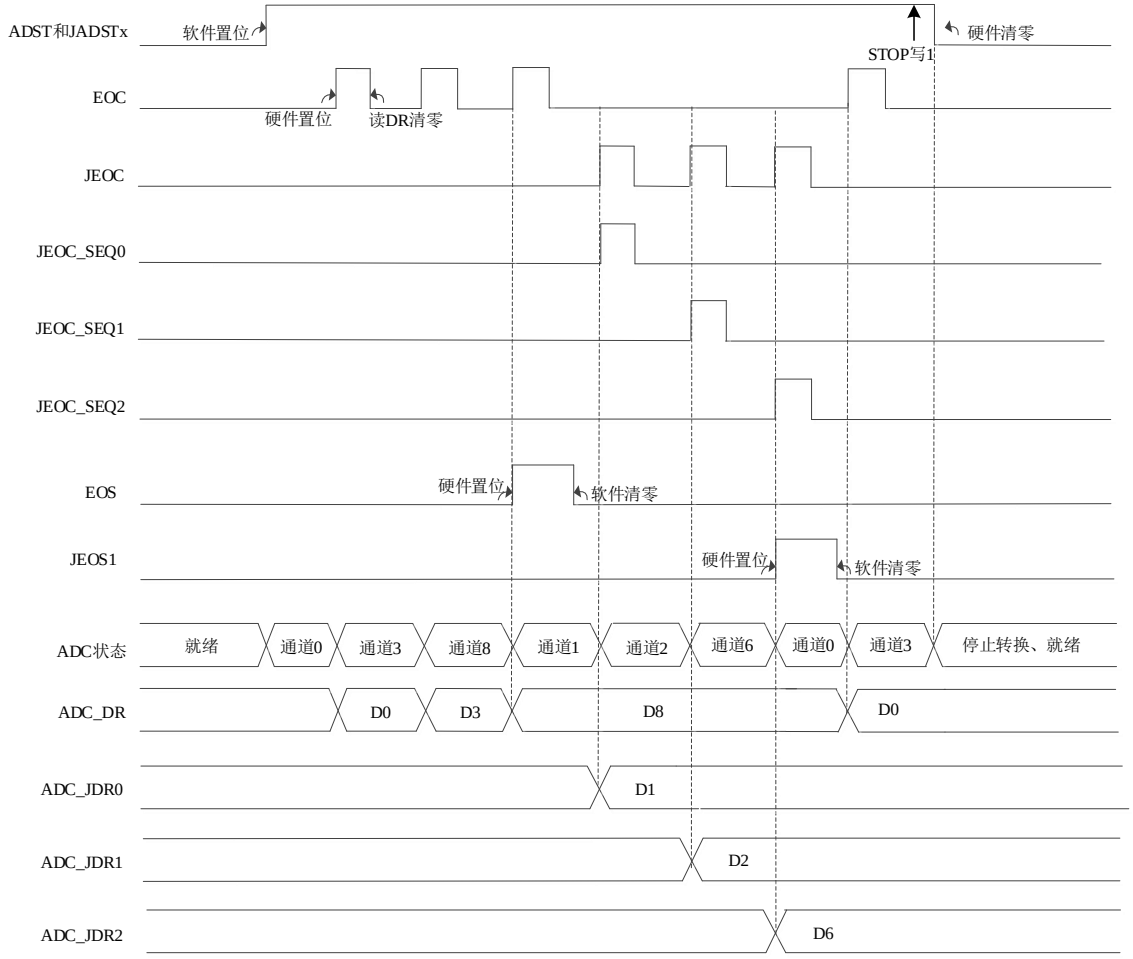


图 14-9 循环间断转换模式，规则硬件触发，注入硬件触发，仅规则间断

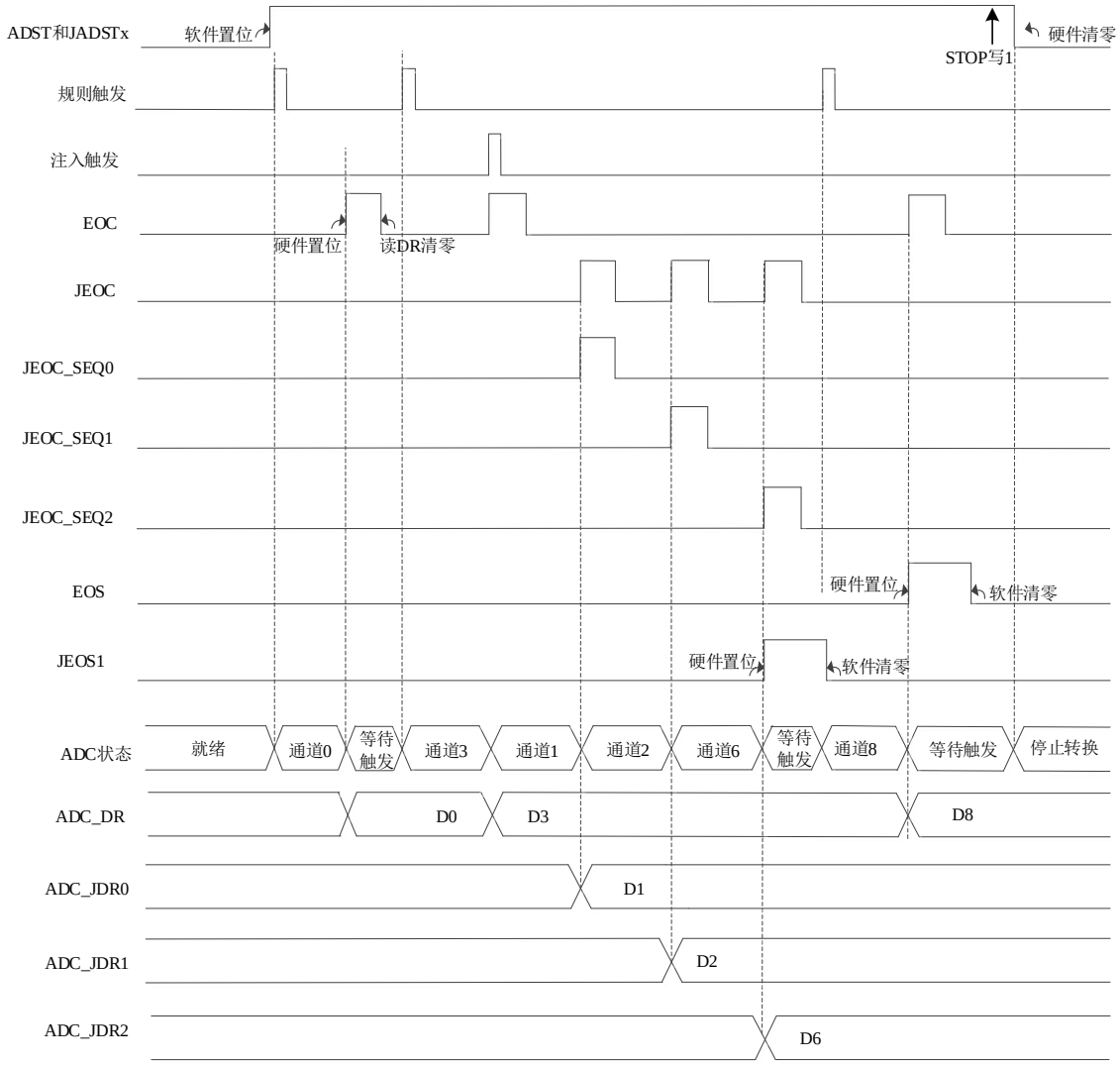
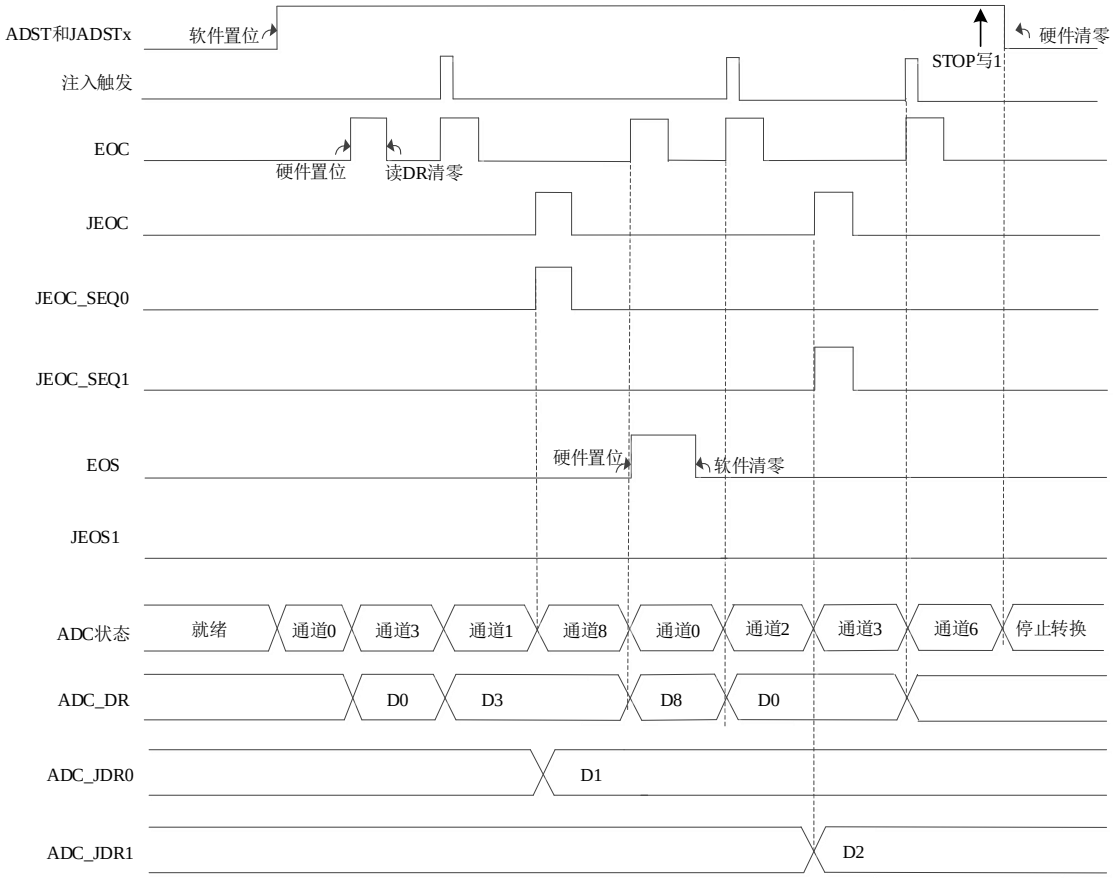


图 14-10 循环间断转换模式，规则软件触发，注入硬件触发，仅注入间断




### 14.3.10 采样时间

ADC 进行电压模数转换的总转换时间计算公式如下：

$$t_{CONV} = t_{SAMP} + t_{SAR} = t_{SAMP} + 13 \times t_{ADC_CLK}$$

其中：$t_{SAR}$ 为固定值，即 13 个 ADC_CLK 时钟周期，$t_{SAMP}$ 为采样时间，由用户根据输入信号在 ADC_SAMPT 寄存器中配置。

示例：

如果 ADC_CLK 频率为 18MHz，采样时间为 3 个 ADC 时钟周期，则转换时间计算如下：

$$t_{CONV} = （3 + 13）\times t_{ADC_CLK} = 16 \times t_{ADC_CLK} = 0.89\ \mu s$$

### 14.3.11 数据溢出

所有规则序列的转换结果都会输出到同一个数据寄存器 ADC_DR，若之前的转换结果未被 CPU 或 DMA 及时处理（读取转换数据或向 EOC 标志写 1 清 0），当新的通道转换完成，向数据寄存器 ADC_DR 输出转换结果时，会产生数据溢

出事件，ADC_ISR 寄存器中数据溢出标志 OVRN 会置 1。

如果 ADC_IER 寄存器中数据溢出中断使能位 OVRNIE 置 1，可产生数据溢出中断。OVRN 标志位通过软件写 1 清 0。

当产生数据溢出事件，ADC 会继续保持工作状态并可以继续进行电压转换，除非将 ADC_CR 寄存器的 STOP 位写 1，从而停止转换并复位规则序列和注入序列转换通道。

ADC_CFG1 寄存器中 OVRN_MOD 位用于配置发生数据溢出事件时，规则数据寄存器 ADC_DR 继续保留之前的转换数据，还是被新的转换数据覆盖。

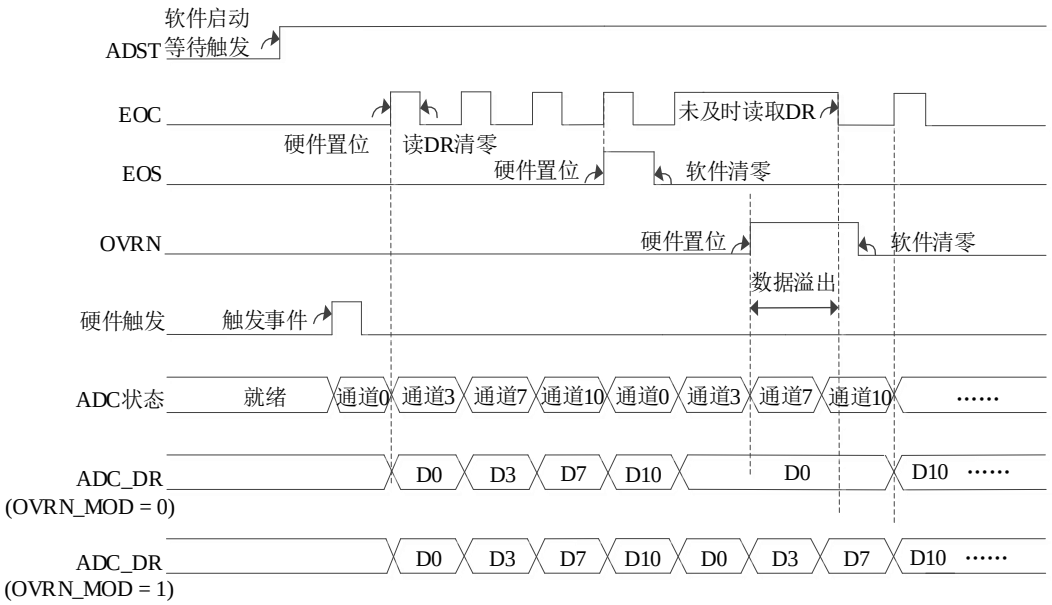
- OVRN_MOD=0

  当产生数据溢出事件，ADC_DR 数据寄存器中之前的转换结果会被保留，并丢弃新的转换结果。如果 OVRN 位保持为 1，ADC 可以继续进行转换，但会丢弃后续转换结果。

- OVRN_MOD=1

  当产生数据溢出事件，ADC_DR 数据寄存器会被新的转换结果所覆盖，从而丢弃之前未读取的结果。如果 OVRN 位保持为 1，ADC 可以继续进行转换，ADC_DR 寄存器始终存储最新的转换结果。

图 14-11 数据溢出处理示例 (1)




1. 规则转换序列选择 0、3、7、10，循环扫描转换模式，硬件触发。

### 14.3.12 DMA 转换数据处理

由于 ADC 规则序列中所有通道的转换结果都会存储在同一个数据寄存器 ADC_DR，因此在多通道转换序列情况下或者软件来不及读取转换数据时，采用 DMA 处理转换数据可以提高效率。这样可以避免存储在 ADC_DR 寄存器的转换结果丢失。

当 DMA 模式已使能（ADC_CFG1 寄存器中 DMAEN 位置 1），在每个规则序列通道转换完成后，ADC 都会生成 DMA 请求。此时，DMA 控制器会将 ADC_DR 寄存器的转换结果传输到软件配置的目标地址。

尽管如此，如果因 DMA 控制器无法及时处理 DMA 传输请求而导致数据溢出事件产生（OVRN=1），ADC 会停止生成 DMA 请求，新转换得到的数据不会通过 DMA 进行传输，直至软件将数据溢出标志位 OVRN 清零。这意味着传输到目标地址（如 SRAM）的所有数据都视为有效数据。

### 14.3.13 数据对齐

转换后的数据有两种对齐方式：左对齐和右对齐，可通过 ADC_CFG1 寄存器的 ALIG 位设置。ALIG = 0 表示数据右对齐，ALIG = 1 表示数据左对齐，如下图所示。

图 14-12 数据对齐

数据左对齐

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 0 | 0 | 0 | 0 |

数据右对齐

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

### 14.3.14 模拟看门狗

#### 模拟看门狗简介

模拟看门狗用于监控通道输入电压是否在所配置的阈值范围之内。

通过设置 ADC_AWDGCR 寄存器中 CHNy（y＝0～15）位，模拟看门狗对所选定转换通道的输入电压进行监控。当 CHNy 位置 1 时，可使能相应通道的模拟看门狗。

模拟看门狗监控通道输入电压的方式是将完整的 12 位原始转换数据与所设置阈值进行比较。由于模拟看门狗的比较发生在转换数据进入数据寄存器之前，

数据对齐的方式不影响模拟看门狗的比较输出。监控电压阈值上下限分别在 ADC_AWDGTR 寄存器中的 AWDG_HT[11:0]和 AWDG_LT[11:0]位域进行设置。

如果转换电压低于阈值下限或高于阈值上限，ADC_ISR 寄存器中模拟看门狗监控电压状态标志位 AWDG 置 1。此时如果 ADC_IER 寄存器中断使能位 AWDGIE 置 1，会产生模拟看门狗监控电压中断。通过向 AWDG 状态标志位写 1，可清 0 此标志。

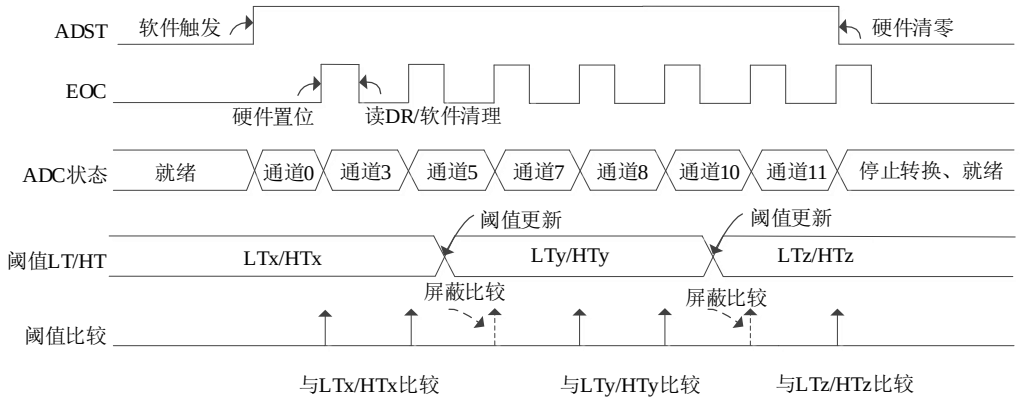
**模拟看门狗监控电压阈值控制**

模拟看门狗监控电压阈值上下限 AWDG_HT[11:0]、AWDG_LT[11:0]可以在 ADC 电压转换过程中（在转换开始和转换结束之间）进行改变。

如果在 ADC 电压转换过程中改变了监控电压阈值，则模拟看门狗会屏蔽本次监控电压比较。开始新转换时，会清除此屏蔽功能，新的监控电压阈值将会应用于下一次 ADC 转换。

ADC 会在每一次转换结束时进行模拟看门狗监控电压比较。在阈值更新过程中，如果当前 ADC 转换电压超出了新设置的阈值范围，AWDG 标志状态不变，不会产生中断。

图 14-13 模拟看门狗阈值更新 (1)




1. 规则转换序列通道选择 0、3、5、7、8、10、11，单次扫描转换模式，软件触发。ADC 模拟看门狗监控通道 0、3、5、7、8、10、11。

### 14.3.15 温度传感器

芯片内置温度传感器，用于测量芯片的结温（$T_J$）。可将温度传感器输出电压 $V_{TS}$ 转换为数字值。

温度传感器的输出电压随温度线性变化。由于制造工艺的差异，该线性的偏移量取决于各个芯片。为提高温度传感器测量的准确性，在生产过程中对每个芯

片的温度传感器进行了校准，校准温度、电压条件及校准值数据存储区地址如下表：

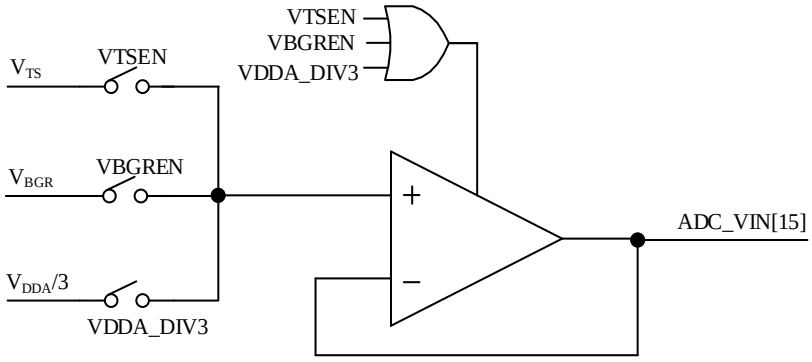
表 14-4 温度传感器校准参数

| 校准参数名称 | 校准温度、电压等条件 | 存储区地址 |
|---|---|---|
| TS_CAL_25 | 温度：25 °C （± 2 °C），<br>$V_{DDA} = V_{REF_ADC} = 3.3$ V | 0x1FFF 0280 |

TS_CAL_25 是在温度 25°C，参考电压为 3.3V 下，温度传感器的 ADC 转换值。

温度传感器与 ADC 之间连接结构如下图所示。使用时需通过将 ADC_CFG2 寄存器中 VTSEN 位置 1 使能 ADC_VIN[15]的转换。由于温度传感器会产生一定功耗，因此不使用时可将 VTSEN 位清零，使温度传感器下电。

图 14-14 内部输入通道结构




使用温度传感器前需要使能 BGR，并等待其稳定（$t_{START}$ 启动稳定时间参见数据手册），详见*内部带隙基准电压*。

ADC 读取温度传感器温度操作步骤如下：

1) 选择 ADC_VIN[15]输入通道；
2) 选择合适的采样时间 $t_{SAMP}$（$t_{SAMP}$ 参见数据手册温度传感器特性）；
3) 将 ADC_CFG2 寄存器中 VTSEN，VDDA_DIV3 和 VBGREN 位清零；
4) 将 ADC_CFG2 寄存器中 VTSEN 位置 1，并等待 BUFF 稳定；
5) 将 ADC_CR 寄存器中 ADST 位置 1（通过软件或外部硬件触发），启动 ADC 转换；
6) 等待转换完成，读取 ADC_DR 寄存器中生成的 $V_{TS}$ 转换数据；
7) 使用以下公式计算温度：

$$\text{Temperature}(°C) = \frac{(V_{TS} - V_{25})}{\text{Avg_Slope}} + 25\ °C$$

其中：

- $V_{TS}$是 ADC 实际转换值换算得到的电压值（单位 mV）；
- $V_{25}$是由 TS_CAL_25（温度传感器出厂 ADC 转换值），在参考电压 3.3V 下换算得到的电压值（单位 mV）；
- Avg_Slope 是斜率（Avg_Slope 参见数据手册温度传感器特性）；
- Temperature 是有符号数据，计算过程需要保留符号进行。

### 14.3.16 内部带隙基准电压检测

内部带隙基准电压 $V_{BGR}$ 为 ADC 提供了一个稳定的电压输出。由于制造工艺的差异，每颗芯片的 $V_{BGR}$ 输出电压不同。在生产过程中对每颗芯片的 $V_{BGR}$ 电压单独进行了校准，校准温度、电压条件及校准值数据存储区地址如下表。用户可以使用此精确电压值来计算实际 $V_{REF_ADC}$ 电压。

表 14-5 内部带隙基准电压校准参数

| 校准参数名称 | 校准温度、电压等条件 | 存储区地址 |
|---|---|---|
| BGR_CAL | 温度：25 °C（± 2 °C）<br>$V_{DDA}$ = $V_{REF_ADC}$ = 3.3 V | 0x1FFF 027C |

其中 BGR_CAL 是在 $V_{REF_ADC}$ = $V_{DDA}$ = 3.3V 条件下，由 ADC 转换得到的内部参考电压输出值 $V_{BGR}$。

内部带隙基准电压与 ADC 之间连接结构如*图：内部输入通道结构*所示。

ADC 转换内部带隙基准电压操作步骤如下：

1) 选择 ADC_VIN[15]输入通道；
2) 选择合适的采样时间 $t_{SAMP}$（$t_{SAMP}$ 参见数据手册内置参考电压特性）；
3) 将 ADC_CFG2 寄存器中 VTSEN，VDDA_DIV3 和 VBGREN 位清零；
4) 将 ADC_CFG2 寄存器中 VBGREN 位置 1，并等待 BUFF 稳定；
5) 将 ADC_CR 寄存器中 ADST 位置 1（通过软件或外部硬件触发），启动 ADC 转换；
6) 等待转换完成，读取 ADC_DR 寄存器中生成的 $V_{BGR}$ 转换数据。

**使用内部带隙基准电压 $V_{BGR}$ 计算实际的 ADC 参考电压 $V_{REF_ADC}$**

ADC 的参考电压可能会有变化，或无法获得准确值。制造过程中在 $V_{REF_ADC}$ = 3.3V 的条件下测试得到的内部参考电压（$V_{BGR}$）及其校准数据可用于计算实际

的 $V_{REF_ADC}$ 电压。

由以下公式可得出为器件供电的实际参考电压 $V_{REF_ADC}$ ：

$$V_{REF_ADC} = 3.3V \times V_{BGR_CAL} / V_{BGR_DATA}$$

其中：

- $V_{BGR_CAL}$ 是 $V_{BGR}$ 的校准值；
- $V_{BGR_DATA}$ 是在当前 $V_{REF_ADC}$ 电压下由 ADC 转换得到的实际 $V_{BGR}$ 输出值。

**将 ADC 采集值转换为绝对电压值**

根据采集到的通道输入电压与参考电压 $V_{REF_ADC}$ 的比例关系，ADC 将通道输入电压转换为数字值。

对于 $V_{REF_ADC}$ 电压值已知的应用，可使用以下公式计算得到该通道输入电压的绝对值：

$$V_{CHANNELX} = \frac{V_{REF_ADC}}{FULL_SCALE} \times ADC_DATAx$$

对于 $V_{REF_ADC}$ 电压值未知的应用，可以使用内部带隙基准电压 $V_{BGR}$ 计算实际的 $V_{REF_ADC}$ 电压，因此 $V_{REF_ADC}$ 可替换为 $V_{BGR}$ 的表达式，从而得出以下通道输入电压公式：

$$V_{CHANNELX} = \frac{3.3V \times V_{BGR_CAL} \times ADC_DATAx}{V_{BGR_DATA} \times FULL_SCALE}$$

其中：

- $V_{BGR_CAL}$ 是 $V_{BGR}$ 的校准值；
- ADC_DATAx 是在当前 $V_{REF_ADC}$ 电压下由 ADC 在通道 x 上测得的值；
- $V_{BGR_DATA}$ 是在当前 $V_{REF_ADC}$ 电压下由 ADC 转换得到的实际 $V_{BGR}$ 输出值；
- FULL_SCALE 是 ADC 输出的最大数字值。例如，由于分辨率为 12 位，该值为 $2^{12} - 1 = 4095$。

### 14.3.17 $V_{DDA}$/3 电压检测

ADC 可用于 $V_{DDA}$ 电压采样，采样前经过分压，得到 $V_{DDA}$/3，连接到 ADC_VIN[15] 内部输入通道，与 ADC 之间连接结构如*图：内部输入通道结构*所示。

通过 ADC_CFG2 寄存器中 VDDA_DIV3 位置 1，可使能 $V_{DDA}$ 输入。为降低分压产生的功耗，不使用时可将 VDDA_DIV3 位清零，禁止 $V_{DDA}$ 输入。

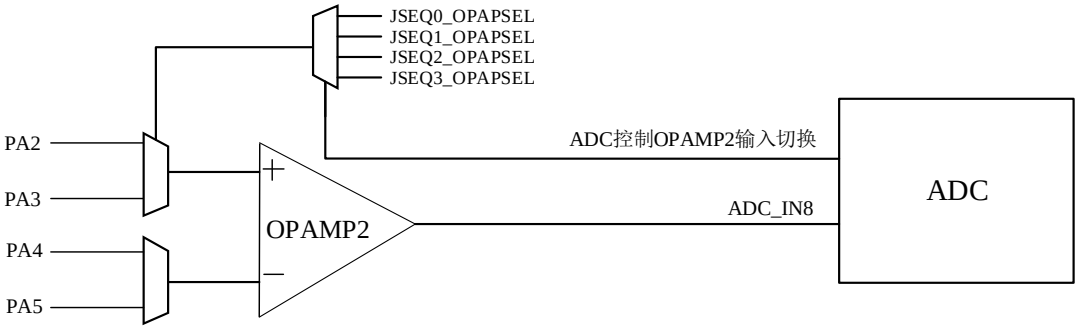
ADC 转换 $V_{DDA}/3$ 操作步骤如下：

1) 选择 ADC_VIN[15]输入通道；
2) 选择合适的采样时间 $t_{SAMP}$（$t_{SAMP}$ 参见数据手册 $V_{DDA}/3$ 检测特性）；
3) 将 ADC_CFG2 寄存器中 VTSEN，VDDA_DIV3 和 VBGREN 位清零；
4) 将 ADC_CFG2 寄存器中 VDDA_DIV3 位置 1，并等待 BUFF 稳定；
5) 将 ADC_CR 寄存器中 ADST 位置 1（通过软件或外部硬件触发），启动 ADC 转换；
6) 等待转换完成，读取 ADC_DR 寄存器中生成的 $V_{TS}$ 转换数据。

### 14.3.18 OPAMP 通道切换

在注入序列使能且选择 ADC_IN8 时，ADC 可控制切换 OPAMP2 的输入通道，如下图所示。注入通道采样之前，ADC 会控制切换 OPAMP2 输入选择，等待建立时间后再开始采样，在前一个注入序列 JESQx 转换完成时，会将 OPAMP2 的正相输入切换到下一个序列 JSEQx+1 对应的输入配置。

图 14-15 ADC 控制 OPAMP2 输入切换结构图




OPAMP 通道切换配置操作步骤如下：

- 通过配置 ADC_OPACR 寄存器中 OPA_SETTIME[9:0]位域，设置 OPAMP2 的切换建立时间；
- 通过配置 ADC_OPACR 寄存器中 JSEQx_OPAINP 位选择对应的 OPAMP2 正相输入；
- 将 ADC_OPACR 寄存器中 JSEQx_OPAEN 位置 1，使能 OPAMP2 输入切换。

## 14.4 ADC 中断

ADC_IRQ 为 ADC 中断请求信号，中断请求信号源详见下表：

表 14-6 中断请求

| 中断事件 | 事件标志 | 中断使能位 | 事件标志/中断清除方法 |
|---|---|---|---|
| 规则序列通道转换结束 | EOC | EOCIE | 向 EOC 位写 1 或读取 ADC_DR 寄存器 |
| 规则序列转换结束 | EOS | EOSIE | 向 EOS 位写 1 |
| 模拟看门狗监控电压事件 | AWDG | AWDGIE | 向 AWDG 位写 1 |
| 注入序列第 x 个转换结束 | JEOC_SEQx | JEOC_SEQxIE | 向 JEOC_SEQx 位写 1 |
| 注入序列通道转换结束 | JEOC | JEOCIE | 向 JEOC 位写 1 |
| 注入序列第 x 段序列转换结束 | JEOSx | JEOSxIE | 向 JEOSx 位写 1 |
| 数据溢出 | OVRN | OVRNIE | 向 OVRN 位写 1 |

## 14.5 ADC 寄存器

ADC 寄存器支持 32 位访问。

表 14-7 ADC 基地址

| 外设 | 基地址 |
|---|---|
| ADC | 0x4001 2400 |

### 14.5.1 ADC 控制寄存器（ADC_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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | JADST4 | JADST3 | JADST2 | JADST1 | Res. | | | STOP | Res. | ADST | ADDIS | ADEN |
| | | | | rs | rs | rs | rs | | | | w | | rs | w | rs |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:12 | 保留 | 写入无效 |
| 11:8 | JADSTx | ADC注入序列第x段启动转换（x = 1 ~ 4）<br>根据JTRIGENx[1:0]位域的配置，ADC立即开始转换（软件触发方式），或者在发生外部硬件触发事件后开始转换（外部硬件触发方式）。<br>根据转换模式、转换启动触发方式配置，此位通过硬件清0，详见*启动/停止转换*。<br>0：当前未进行ADC注入序列第x段转换<br>1：写入1可启动ADC注入序列第x段。读取值为1表示ADC正处于转换状态<br>*注意：仅当ADEN=1且JEN=1时，可将JADSTx置1。* |
| 7:5 | 保留 | 写入无效 |
| 4 | STOP | ADC停止转换<br>此位写入1可以停止正在进行的转换，当前转换结果会丢弃。当 |

| | | |
|---|---|---|
| | | 转换已停止、并且ADC已准备好接收新的启动转换命令时，硬件将ADST和JADSTx位清0。<br>0：写0无效<br>1：写入1可停止ADC转换 |
| 3 | 保留 | 写入无效 |
| 2 | ADST | ADC规则序列启动转换<br>根据RTRIGEN[1:0]位域的配置，ADC立即开始转换（软件触发方式），或者在发生外部硬件触发事件后开始转换（外部硬件触发方式）。<br>根据转换模式、转换启动触发方式配置，此位通过硬件清0，详见*启动/停止转换*。<br>0：当前未进行ADC规则序列转换<br>1：写入1可启动ADC规则序列。读取值为1表示ADC正处于转换状态<br>*注意：仅当ADEN=1时，可将ADST置1。* |
| 1 | ADDIS | 禁止ADC<br>仅当ADST和JADSTx均为0时，此位写入1可以禁止ADC并将ADEN清0。<br>0：写0无效<br>1：写入1可禁止ADC |
| 0 | ADEN | 使能ADC<br>ADEN位由软件置1，等待启动稳定时间后，ADC处于就绪状态，可进行电压转换。<br>0：未使能<br>1：写入 1 使能 |

### 14.5.2 ADC 配置寄存器 1（ADC_CFG1）

偏移地址：0x04

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | CONV_MOD[1:0] | | OVRN_MOD | Res. | | RTRIGEN[1:0] | | Res. | ALIG | Res. | | | | DMAEN |
| | | rw | rw | rw | | | rw | rw | | rw | | | | | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:28 | LEN[3:0] | 规则通道序列长度<br>这些位定义为规则通道转换序列中的通道数。<br>0000：1个转换，RSEQ0<br>0001：2个转换，RSEQ0~RSEQ1<br>⋮<br>1010：11个转换，RSEQ0~RSEQ10<br>1011：12个转换，RSEQ0~RSEQ11<br>其他：保留（默认1个转换）<br>*注意：仅当ADST为0时，可对此位域执行写操作。* |
| 27:20 | 保留 | 写入无效 |
| 19:16 | EXTR_SEL[3:0] | 规则序列外部硬件触发源选择<br>0000：TIM1_TRIG_OUT<br>0001：TIM1_OC1_OUT<br>0010：TIM1_OC3_OUT<br>0011：TIM1_OC5_OUT<br>0100：TIM1_OC7_OUT<br>0101：TIM1_OC8_OUT<br>0110：TIM1_OC7_OUT或TIM1_OC8_OUT<br>0111：TIM3_TRIG_OUT<br>1000：TIM3_OC1_OUT<br>1001：TIM3_OC2_OUT<br>1010：TIM4_TRIG_OUT<br>1011：TIM4_OC1_OUT<br>1100：HALLTIM_OVR<br>1101：EXTI通道8<br>1110：EXTI通道11<br>1111：EXTI通道12<br>*注意：仅当ADST为0时，可对此位域执行写操作。* |
| 15:14 | 保留 | 写入无效 |

| | | |
|---|---|---|
| 13:12 | CONV_MOD[1:0] | 转换模式选择<br>00：单次扫描转换模式<br>01：循环扫描转换模式<br>10：循环间断转换模式<br>11：保留（默认选择单次扫描转换模式）<br>*注意：仅当ADST和JADSTx均为0时，可对此位域执行写操作。* |
| 11 | OVRN_MOD | 数据溢出管理方式<br>0：数据溢出时，ADC_DR保留上次数据，丢弃本次转换数据<br>1：数据溢出时，ADC_DR覆盖上次数据<br>*注意：仅当ADST为0时，可对此位执行写操作。* |
| 10:9 | 保留 | 写入无效 |
| 8:7 | RTRIGEN[1:0] | 规则序列触发方式和极性选择<br>用于选择规则序列触发方式及外部硬件触发事件极性，并使能触发。<br>00：软件触发方式<br>01：外部硬件事件上升沿触发<br>10：外部硬件事件下降沿触发<br>11：外部硬件事件上升沿和下降沿均触发<br>*注意：仅当ADST为0时，可对此位域执行写操作。* |
| 6 | 保留 | 写入无效 |
| 5 | ALIG | 数据对齐方式<br>0：右对齐<br>1：左对齐<br>*注意：仅当ADST和JADSTx均为0时，可对此位执行写操作。* |
| 4:1 | 保留 | 写入无效 |
| 0 | DMAEN | DMA使能<br>0：禁止<br>1：使能<br>*注意：仅当ADST为0时，可对此位执行写操作。* |

### 14.5.3 ADC 配置寄存器 2（ADC_CFG2）

偏移地址：0x08

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | PRESC[3:0] | | | | Res. | | | | VDDA_DIV3 | Res. | VTSEN | VBGREN |
| | | | | rw | rw | rw | rw | | | | | rw | | rw | rw |

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:28 | 保留 | 写入无效 |
| 27:24 | PRESC[3:0] | ADC时钟源PCLK预分频系数<br>0000：PCLK 1分频<br>0001：PCLK 2分频<br>0010：PCLK 3分频<br>0011：PCLK 4分频<br>0100：PCLK 6分频<br>0101：PCLK 8分频<br>0110：PCLK 12分频<br>0111：PCLK 16分频<br>1000：PCLK 18分频<br>1001：PCLK 24分频<br>1010：PCLK 32分频<br>1011：PCLK 36分频<br>1100：PCLK 64分频<br>其他：保留（默认PCLK 1分频）<br>*注意：仅当ADEN=0时，可对此位域执行写操作。* |
| 23:20 | 保留 | 写入无效 |
| 19 | VDDA_DIV3 | $V_{DDA}$输入使能<br>0：禁止<br>1：使能<br>*注意：$V_{BGR}$，$V_{DDA}$/3和$V_{TS}$共用一个ADC通道，不允许同时使能；* |

*仅当ADST=0且JADSTx均为0时，可对此位域执行写操作。*

| | | |
|---|---|---|
| 18 | 保留 | 写入无效 |
| 17 | VTSEN | 温度传感器通道输入使能<br>0：禁止<br>1：使能<br>*注意：$V_{BGR}$，$V_{DDA}$/3和$V_{TS}$共用一个ADC通道，不允许同时使能；仅当ADST=0且JADSTx均为0时，可对此位域执行写操作。* |
| 16 | VBGREN | $V_{BGR}$通道输入使能<br>0：禁止<br>1：使能<br>*注意：$V_{BGR}$，$V_{DDA}$/3和$V_{TS}$共用一个ADC通道，不允许同时使能；仅当ADST=0且JADSTx均为0时，可对此位域执行写操作。* |
| 15:14 | 保留 | 写入无效 |
| 13:12 | DIS_SEL[1:0] | 循环间断模式间断配置选择<br>0x：规则序列使能间断效果，每次触发转换1个通道<br>10：注入序列使能间断效果，每次触发转换1个通道<br>11：规则和注入序列均使能间断效果，每次触发转换1个通道<br>*注意：仅当ADST和JADSTx均为0时，允许对此寄存器执行写操作。* |
| 11:1 | 保留 | 写入无效 |
| 0 | REF_ADC | ADC参考电压选择<br>0：$V_{DDA}$电压<br>1：VREFBUF<br>*注意： 仅当ADEN=0时，才允许对此位域执行写操作。* |

### 14.5.4 ADC 中断和状态寄存器（ADC_ISR）

偏移地址：0x10

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | JEOS4 | JEOS3 | JEOS2 | JEOS1 | Res. | | | | JEOC_SEQ3 | JEOC_SEQ2 | JEOC_SEQ1 | JEOC_SEQ0 |
| | | | | rc_w1 | rc_w1 | rc_w1 | rc_w1 | | | | | rc_w1 | rc_w1 | rc_w1 | rc_w1 |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | AWDG | Res. | JEOC | OVRN | EOS | EOC | Res. | |
| | | | | | | | | rc_w1 | | rc_w1 | rc_w1 | rc_w1 | rc_w1 | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:28 | 保留 | 写入无效 |
| 27:24 | JEOSx | 注入序列第x段序列转换结束标志（x = 1 ~ 4）<br>注入序列转换结束时，由硬件将此位置1。通过软件写入1将此位清0。<br>0：注入序列第x段的通道序列转换未完成<br>1：注入序列第x段的通道序列转换已完成 |
| 23:20 | 保留 | 写入无效 |
| 19:16 | JEOC_SEQx | 注入序列第x个转换结束标志（x = 0 ~ 3）<br>当注入序列中每个通道转换结束，由硬件将对应的位置1。通过软件写入1，或读取ADC_JDRx寄存器将对应的位清0。<br>0：注入通道JSEQx转换未完成<br>1：注入通道JSEQx转换已完成 |
| 15:8 | 保留 | 写入无效 |
| 7 | AWDG | ADC模拟看门狗监控电压事件标志<br>当转换电压超出ADC_AWDGTR寄存器中设置的阈值范围时，硬件将该位置1，通过软件写入1将此位清0。<br>0：未发生监控电压事件<br>1：已发生监控电压事件 |
| 6 | 保留 | 写入无效 |
| 5 | JEOC | 注入序列通道转换结束标志<br>当注入序列中每个通道转换结束，由硬件将此位置1。通过软件 |

写入1将此位清0。
0：注入通道转换未完成
1：注入通道转换已完成

| | | |
|---|---|---|
| 4 | OVRN | 数据溢出标志<br>在EOC标志已置1时，再次发生新的转换，硬件将此位置1。通过软件写入1将此位清0。<br>0：未发生数据溢出事件<br>1：已发生数据溢出事件 |
| 3 | EOS | 规则序列转换结束标志<br>规则序列转换结束时，由硬件将此位置1。通过软件写入1将此位清0。<br>0：规则通道序列转换未完成<br>1：规则通道序列转换已完成 |
| 2 | EOC | 规则序列通道转换结束标志<br>当规则序列中每个通道转换结束，由硬件将此位置1。通过软件写入1，或读取ADC_DR寄存器都将此位清0。<br>0：规则通道转换未完成<br>1：规则通道转换已完成 |
| 1:0 | 保留 | 写入无效 |

### 14.5.5 ADC 中断使能寄存器（ADC_IER）

偏移地址：0x14

复位值：0x0000 0000

*注意：　仅当 ADST=0 且 JADSTx 均为 0 时，允许对此寄存器执行写操作。*

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | JEOS4IE | JEOS3IE | JEOS2IE | JEOS1IE | Res. | | | | JEOC_SEQ3IE | JEOC_SEQ2IE | JEOC_SEQ1IE | JEOC_SEQ0IE |
| | | | | rw | rw | rw | rw | | | | | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | AWDGIE | Res. | JEOCIE | OVRNIE | EOSIE | EOCIE | Res. | |
| | | | | | | | | rw | | rw | rw | rw | rw | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|

| | | |
|---|---|---|
| 31:28 | 保留 | 写入无效 |
| 27:24 | JEOSxIE | 注入序列第x段序列转换结束中断使能（x = 1 ~ 4）<br>0：禁止<br>1：使能，JEOSx位置1时产生中断 |
| 23:20 | 保留 | 写入无效 |
| 19:16 | JEOC_SEQxIE | 注入通道第x个转换结束中断使能（x = 0 ~ 3）<br>0：禁止<br>1：使能，JEOC_SEQx位置1时产生中断 |
| 15:8 | 保留 | 写入无效 |
| 7 | AWDGIE | 模拟看门狗监控电压事件中断使能<br>0：禁止<br>1：使能，AWDG位置1时产生中断 |
| 6 | 保留 | 写入无效 |
| 5 | JEOCIE | 注入序列通道转换结束中断使能<br>0：禁止<br>1：使能，JEOC位置1时产生中断 |
| 4 | OVRNIE | 数据溢出中断使能<br>0：禁止<br>1：使能，OVRN位置1时产生中断 |
| 3 | EOSIE | 规则序列转换结束中断使能<br>0：禁止<br>1：使能，EOS位置1时产生中断 |
| 2 | EOCIE | 规则通道转换结束中断使能<br>0：禁止<br>1：使能，EOC位置1时产生中断 |
| 1:0 | 保留 | 写入无效 |

### 14.5.6 ADC 采样时间寄存器（ADC_SAMPT）

偏移地址：0x18

复位值：0x0000 0000

*注意：　仅当 ADST=0 且 JADSTx 均为 0 时，允许对此寄存器执行写操作。*

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 写入无效 |
| 7:4 | JSAMPT[3:0] | 注入序列采样时间参数<br>0000：3个ADC_CLK时钟周期<br>0001：4个ADC_CLK时钟周期<br>0010：5个ADC_CLK时钟周期<br>0011：7个ADC_CLK时钟周期<br>0100：9个ADC_CLK时钟周期<br>0101：12个ADC_CLK时钟周期<br>0110：19个ADC_CLK时钟周期<br>0111：39个ADC_CLK时钟周期<br>1000：59个ADC_CLK时钟周期<br>1001：79个ADC_CLK时钟周期<br>1010：119个ADC_CLK时钟周期<br>1011：159个ADC_CLK时钟周期<br>1100：239个ADC_CLK时钟周期<br>1101：319个ADC_CLK时钟周期<br>1110：479个ADC_CLK时钟周期<br>1111：639个ADC_CLK时钟周期 |
| 3:0 | RSAMPT[3:0] | 规则序列采样时间参数<br>0000：3个ADC_CLK时钟周期<br>0001：5个ADC_CLK时钟周期 |

0010：7个ADC_CLK时钟周期

0011：12个ADC_CLK时钟周期

0100：19个ADC_CLK时钟周期

0101：39个ADC_CLK时钟周期

0110：79个ADC_CLK时钟周期

0111：119个ADC_CLK时钟周期

1000：159个ADC_CLK时钟周期

1001：239个ADC_CLK时钟周期

1010：319个ADC_CLK时钟周期

1011：479个ADC_CLK时钟周期

1100：639个ADC_CLK时钟周期

1101：959个ADC_CLK时钟周期

1110：1279个ADC_CLK时钟周期

1111：1919个ADC_CLK时钟周期

### 14.5.7 ADC 模拟看门狗配置寄存器（ADC_AWDGCR）

偏移地址：0x20

复位值：0x0000 0000

*注意：　仅当 ADST=0 且 JADSTx 均为 0 时，允许对此寄存器执行写操作。*

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CHN15 | CHN14 | CHN13 | CHN12 | CHN11 | CHN10 | CHN9 | CHN8 | CHN7 | CHN6 | CHN5 | CHN4 | CHN3 | CHN2 | CHN1 | CHN0 |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15:0 | CHNy | 模拟看门狗监控通道选择（y = 0 ~ 15）<br>每一个寄存器位对应相应通道号。<br>0：通道CHNy不被AWDG监控<br>1：通道CHNy被AWDG监控 |

### 14.5.8 ADC 模拟看门狗监控电压阈值寄存器（ADC_AWDGTR）

偏移地址：0x28

复位值：0x0FFF 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | AWDG_HT[11:0] | | | | | | | | | | | |
| | | | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:28 | 保留 | 写入无效 |
| 27:16 | AWDG_HT[11:0] | ADC模拟看门狗监控电压阈值上限 |
| 15:12 | 保留 | 写入无效 |
| 11:0 | AWDG_LT[11:0] | ADC模拟看门狗监控电压阈值下限 |

### 14.5.9 ADC 规则序列寄存器 1（ADC_RSEQ1）

偏移地址：0x34

复位值：0x0000 0000

*注意：　仅当 ADST=0 时，允许对此寄存器执行写操作。*

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SEQ7[3:0] | | | | SEQ6[3:0] | | | | SEQ5[3:0] | | | | SEQ4[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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SEQ3[3:0] | | | | SEQ2[3:0] | | | | SEQ1[3:0] | | | | SEQ0[3:0] | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:28 | SEQ7[3:0] | 规则序列中第7个转换的通道配置<br>请参见SEQ0[3:0]说明。 |

| | | |
|---|---|---|
| 27:24 | SEQ6[3:0] | 规则序列中第6个转换的通道配置<br>请参见SEQ0[3:0]说明。 |
| 23:20 | SEQ5[3:0] | 规则序列中第5个转换的通道配置<br>请参见SEQ0[3:0]说明。 |
| 19:16 | SEQ4[3:0] | 规则序列中第4个转换的通道配置<br>请参见SEQ0[3:0]说明。 |
| 15:12 | SEQ3[3:0] | 规则序列中第3个转换的通道配置<br>请参见SEQ0[3:0]说明。 |
| 11:8 | SEQ2[3:0] | 规则序列中第2个转换的通道配置<br>请参见SEQ0[3:0]说明。 |
| 7:4 | SEQ1[3:0] | 规则序列中第1个转换的通道配置<br>请参见SEQ0[3:0]说明。 |
| 3:0 | SEQ0[3:0] | 规则序列中第0个转换的通道配置<br>配置为“0000”时，转换ADC_IN0，“0001”时，转换ADC_IN1，以此类推。 |

### 14.5.10 ADC 规则序列寄存器 2（ADC_RSEQ2）

偏移地址：0x38

复位值：0x0000 0000

*注意：　仅当 ADST=0 时，允许对此寄存器执行写操作。*

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SEQ11[3:0] | | | | SEQ10[3:0] | | | | SEQ9[3:0] | | | | SEQ8[3:0] | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |

| | | |
|---|---|---|
| 15:12 | SEQ11[3:0] | 规则序列中第11个转换的通道配置<br>请参见SEQ0[3:0]说明。 |
| 11:8 | SEQ10[3:0] | 规则序列中第10个转换的通道配置<br>请参见SEQ0[3:0]说明。 |
| 7:4 | SEQ9[3:0] | 规则序列中第9个转换的通道配置<br>请参见SEQ0[3:0]说明。 |
| 3:0 | SEQ8[3:0] | 规则序列中第8个转换的通道配置<br>请参见SEQ0[3:0]说明。 |

### 14.5.11 ADC 规则数据寄存器（ADC_DR）

偏移地址：0x40

复位值：0x0000 0000

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

| 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 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15:0 | DATA[15:0] | 规则转换数据<br>包含规则通道的转换结果，数据左对齐或右对齐。 |

### 14.5.12 ADC 配置寄存器 3（ADC_CFG3）

偏移地址：0x60

复位值：0x0000 0000

*注意：　仅当 JADSTx 均为 0 时，允许对此寄存器执行写操作。*

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

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | | | | EXTJ4_EN | EXTJ3_EN | EXTJ2_EN | EXTJ1_EN | Res. | | | | | | JAUTO | JEN |
| | | | | rw | rw | rw | rw | | | | | | | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:12 | 保留 | 写入无效 |
| 11:8 | EXTJx_EN | 注入序列第x段使能（x = 1 ~ 4）<br>0：禁止<br>1：使能 |
| 7:2 | 保留 | 写入无效 |
| 1 | JAUTO | 自动注入模式使能<br>0：禁止<br>1：使能 |
| 0 | JEN | 注入序列转换使能<br>0：禁止<br>1：使能 |

### 14.5.13 ADC 注入序列段触发选择寄存器（ADC_JTRIGSEL）

偏移地址：0x64

复位值：0x0000 0000

*注意：　仅当 JADSTx 均为 0 时，允许对此寄存器执行写操作。*

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EXTJ4_SEL[3:0] | | | | Res. | | JTRIGEN4[1:0] | | EXTJ3_SEL[3:0] | | | | Res. | | JTRIGEN3[1:0] | |
| 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 |
| EXTJ2_SEL[3:0] | | | | Res. | | JTRIGEN2[1:0] | | EXTJ1_SEL[3:0] | | | | Res. | | JTRIGEN1[1:0] | |
| rw | rw | rw | rw | | | rw | rw | rw | rw | rw | rw | | | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:28 | EXTJ4_SEL[3:0] | 注入序列第四段外部硬件触发源选择<br>请参见EXTJ1_SEL[3:0]说明。 |

| | | |
|---|---|---|
| 27:26 | 保留 | 写入无效 |
| 25:24 | JTRIGEN4[1:0] | 注入序列第四段触发方式和极性选择<br>请参见JTRIGEN1 [1:0]说明。 |
| 23:20 | EXTJ3_SEL[3:0] | 注入序列第三段外部硬件触发源选择<br>请参见EXTJ1_SEL[3:0]说明。 |
| 19:18 | 保留 | 写入无效 |
| 17:16 | JTRIGEN3[1:0] | 注入序列第三段触发方式和极性选择<br>请参见JTRIGEN1 [1:0]说明。 |
| 15:12 | EXTJ2_SEL[3:0] | 注入序列第二段外部硬件触发源选择<br>请参见EXTJ1_SEL[3:0]说明。 |
| 11:10 | 保留 | 写入无效 |
| 9:8 | JTRIGEN2[1:0] | 注入序列第二段触发方式和极性选择<br>请参见JTRIGEN2 [1:0]说明。 |
| 7:4 | EXTJ1_SEL[3:0] | 注入序列第一段外部硬件触发源选择<br>0000：TIM1_TRIG_OUT<br>0001：TIM1_OC1_OUT<br>0010：TIM1_OC3_OUT<br>0011：TIM1_OC5_OUT<br>0100：TIM1_OC7_OUT<br>0101：TIM1_OC8_OUT<br>0110：TIM1_OC7_OUT或TIM1_OC8_OUT<br>0111：TIM3_TRIG_OUT<br>1000：TIM3_OC1_OUT<br>1001：TIM3_OC2_OUT<br>1010：TIM4_TRIG_OUT<br>1011：TIM4_OC1_OUT<br>1100：HALLTIM_OVR<br>1101：EXTI通道8 |

1110：EXTI通道11

1111：EXTI通道12

| | | |
|---|---|---|
| 3:2 | 保留 | 写入无效 |
| 1:0 | JTRIGEN1[1:0] | 注入序列第一段触发方式和极性选择<br>用于选择注入序列触发方式及外部硬件触发事件极性，并使能触发。<br>00：软件触发方式<br>01：外部硬件事件上升沿触发<br>10：外部硬件事件下降沿触发<br>11：外部硬件事件上升沿和下降沿均触发 |

### 14.5.14 ADC 注入序列段通道选择寄存器（ADC_JSETSEL）

偏移地址：0x68

复位值：0x0000 0000

*注意：　仅当JADSTx 均为0 时，允许对此寄存器执行写操作。*

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| JSET4_SEL[3:0] | | | | JSET3_SEL[3:0] | | | | JSET2_SEL[3:0] | | | | JSET1_SEL[3:0] | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15:12 | JSET4_SEL[3:0] | 注入序列第四段转换通道选择<br>请参见JSET1_SEL[3:0]说明。 |
| 11:8 | JSET3_SEL[3:0] | 注入序列第三段转换通道选择<br>请参见JSET1_SEL[3:0]说明。 |
| 7:4 | JSET2_SEL[3:0] | 注入序列第二段转换通道选择<br>请参见JSET1_SEL[3:0]说明。 |

| | | |
|---|---|---|
| 3 | JSET1_SEL[3] | 注入序列第一段转换通道选择JSEQ3<br>0：不选择<br>1：选择 |
| 2 | JSET1_SEL[2] | 注入序列第一段转换通道选择JSEQ2<br>0：不选择<br>1：选择 |
| 1 | JSET1_SEL[1] | 注入序列第一段转换通道选择JSEQ1<br>0：不选择<br>1：选择 |
| 0 | JSET1_SEL[0] | 注入序列第一段转换通道选择JSEQ0<br>0：不选择<br>1：选择 |

### 14.5.15 ADC 注入序列寄存器（ADC_JSEQ）

偏移地址：0x6C

复位值：0x0000 0000

*注意：　仅当 JADSTx 均为 0 时，允许对此寄存器执行写操作。*

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| JSEQ3[3:0] | | | | JSEQ2[3:0] | | | | JSEQ1[3:0] | | | | JSEQ0[3:0] | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15:12 | JSEQ3[3:0] | 注入序列中第3个转换的通道配置<br>请参见JSEQ0[3:0]说明。 |
| 11:8 | JSEQ2[3:0] | 注入序列中第2个转换的通道配置<br>请参见JSEQ0[3:0]说明。 |

| | | |
|---|---|---|
| 7:4 | JSEQ1[3:0] | 注入序列中第1个转换的通道配置<br>请参见JSEQ0[3:0]说明。 |
| 3:0 | JSEQ0[3:0] | 注入序列中第0个转换的通道配置<br>配置为“0000”时，转换ADC_IN0，“0001”时，转换ADC_IN1，以此类推。 |

### 14.5.16 ADC 注入数据寄存器 x（ADC_JDRx）（x = 0 ~ 3）

偏移地址：0x70 ~ 0x7C

复位值：0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15:0 | JDATA[15:0] | 注入转换数据<br>包含注入通道的转换结果，数据左对齐或右对齐。 |

### 14.5.17 ADC OPA 控制寄存器（ADC_OPACR）

偏移地址：0x90

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | OPA_SETTIME[9:0] | | | | | | | | | | Res. | JSEQ3_ OPAINP | Res. | |
| | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | | rw | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | JSEQ2_ OPAINP | Res. | | | JSEQ1_ OPAINP | Res. | | | JSEQ0_ OPAINP | Res. | | JSEQ3_ OPAEN | JSEQ2_ OPAEN | JSEQ1_ OPAEN | JSEQ0_ OPAEN |
| | rw | | | | rw | | | | rw | | | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:30 | 保留 | 写入无效 |

| | | |
|---|---|---|
| 29:20 | OPA_SETTIME[9:0] | OPAMP切换建立时间<br>0：0个ADC_CLK时钟周期<br>1：1个ADC_CLK时钟周期<br>⋮<br>1023：1023个ADC_CLK时钟周期 |
| 19 | 保留 | 写入无效 |
| 18 | JSEQ3_OPAINP | 注入序列第3个转换OPAMP的正相选择<br>0：PA2<br>1：PA3 |
| 17:15 | 保留 | 写入无效 |
| 14 | JSEQ2_OPAINP | 注入序列第2个转换OPAMP的正相选择<br>0：PA2<br>1：PA3 |
| 13:11 | 保留 | 写入无效 |
| 10 | JSEQ1_OPAINP | 注入序列第1个转换OPAMP的正相选择<br>0：PA2<br>1：PA3 |
| 9:7 | 保留 | 写入无效 |
| 6 | JSEQ0_OPAINP | 注入序列第0个转换OPAMP的正相选择<br>0：PA2<br>1：PA3 |
| 5:4 | 保留 | 写入无效 |
| 3 | JSEQ3_OPAEN | 注入序列第3个转换OPAMP切换使能<br>0：不切换<br>1：切换为JSEQ3_OPAINP位表示的通道 |

| | | |
|---|---|---|
| 2 | JSEQ2_OPAEN | 注入序列第2个转换OPAMP切换使能<br>0：不切换<br>1：切换为JSEQ2_OPAINP位表示的通道 |
| 1 | JSEQ1_OPAEN | 注入序列第1个转换OPAMP切换使能<br>0：不切换<br>1：切换为JSEQ1_OPAINP位表示的通道 |
| 0 | JSEQ0_OPAEN | 注入序列第0个转换OPAMP切换使能<br>0：不切换<br>1：切换为JSEQ0_OPAINP位表示的通道 |

# 15 参考电压源（VREFBUF）

## 15.1 简介

芯片内置 VREFBUF，作为高精度基准电压源，可在全温度范围内稳定工作。VREFBUF 已出厂校准，上电自动加载出厂校准值到校准寄存器。VREFBUF 输出电压不随 $V_{DDA}$ 变化，可为 ADC 提供高精度参考电压，可为 6 bit RDAC（64 级分压参考源）提供输入电压，可作为 OPAMP 的 1/2 分压参考电压源。

## 15.2 VREFBUF 主要特性

- 支持 4 种参考电压输出
  - 2.048V（2.4V ≤ $V_{DDA}$ ≤5.5V）；
  - 2.5V（2.8V ≤ $V_{DDA}$ ≤5.5V）；
  - 3.6V（3.9V ≤ $V_{DDA}$ ≤5.5V）；
  - 4.2V（4.5V ≤ $V_{DDA}$ ≤5.5V）；
- 支持 ADC 采样率最大 1.125Msps

## 15.3 VREFBUF 功能描述

使用 VREFBUF 之前需使能 BGR，并等待其稳定（$t_{START}$ 启动稳定时间参见数据手册），详见*内部带隙基准电压*。

VREFBUF 输出的使能和禁止，可通过 VREFBUF_CSR 寄存器 EN 位配置，使能后，需等待启动稳定时间 $t_{STAB}$（$t_{STAB}$ 参见数据手册），以保证参考电压输出达到预期值。

VREFBUF 对 2.048V、2.5V、3.6V 和 4.2V 四种输出电压都进行了出厂校准，使能前需从校准参数存储区中读取相应校准值并写入 *VREFBUF 校准寄存器（VREFBUF_CAL）*，POR/PDR 和 NRST 复位自动加载 2.048V 对应的校准值，校准值存储在校准参数存储区中，详见*表：出厂校准参数组成*。

## 15.4 VREFBUF 寄存器

VREFBUF 寄存器支持 32 位访问。

表 15-2 VREFBUF 基地址

| 外设 | 基地址 |
|---|---|
| VREFBUF | 0x4001 01B0 |

### 15.4.1 VREFBUF 控制和状态寄存器（VREFBUF_CSR）

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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | DRV | Res. | | | VRS[1:0] | | Res. | EN |
| | | | | | | | | rw | | | | rw | rw | | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 写入无效 |
| 7 | DRV | VREFBUF驱动控制<br>0：高驱<br>1：低驱 |
| 6:4 | 保留 | 写入无效 |
| 3:2 | VRS[1:0] | VREFBUF输出电压<br>00：2.048V<br>01：2.5V<br>10：3.6V<br>11：4.2V |
| 1 | 保留 | 写入无效 |

| | | |
|---|---|---|
| 0 | EN | VREFBUF使能<br>0：禁止<br>1：使能 |

### 15.4.2 VREFBUF 校准寄存器（VREFBUF_CAL）

偏移地址：0x04

复位值：0x0000 00XX

*注意：　仅 POR/PDR 和 NRST 复位时，复位该寄存器。*

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:6 | 保留 | 写入无效 |
| 5:0 | VREFBUF_CAL[5:0] | VREFBUF 输出电压的校准值<br>POR/PDR 和 NRST 复位后，将 2.048V 的出厂校准值自动加载到该寄存器。配置 VREFBUF_CSR 寄存器中的 VRS[1:0]位域切换 VREFBUF 输出电压后，需要从 Flash 存储地址中读取对应档位的校准值，并加载到当前寄存器中完成校准，可通过修改该位域进一步细调 VREFBUF。 |

# 16 运算放大器（OPAMP）

## 16.1 简介

芯片内置 3 路可编程增益运算放大器，可用于放大微弱输入信号或弱驱动信号阻抗匹配。运算放大器支持多种工作模式：独立运算放大器、电压跟随、伪差分 PGA 和单端 PGA 模式。

OPAMP2 支持由 ADC 或 TIMx 控制的输入通道分时切换，可以实现两个 OPAMP 功能。

## 16.2 OPAMP 主要特性

- 支持多种工作模式：
  - 独立运放模式
  - 电压跟随模式
  - 伪差分 PGA 模式
  - 单端 PGA 模式，包含同相 PGA 模式、反相 PGA 模式
- 正相偏置电压可选：$V_{DDA}$/2、VREFBUF/2、$V_{BGR}$
- 可编程增益设置
  - 同相 PGA 增益：x2、x4、x8、x12、x16
  - 反相 PGA 和伪差分 PGA 增益：x1、x3、x7、x11、x15
- 支持输入失调电压校准
- 四档功耗可选：高档位、中高档位、中低档位、低档位
- OPAMP1 输出可配置通过片内电阻输出到 I/O，外接电容实现 RC 滤波功能
- OPAMP2 与ADC/TIMx 互联，支持输入通道分时切换
- 输出信号互联：
  - OPAMP1 通过片内电阻输出连接到 ADC 输入通道
  - OPAMP1/2 输出连接到 ADC 输入通道和 COMP1/2 的正相输入

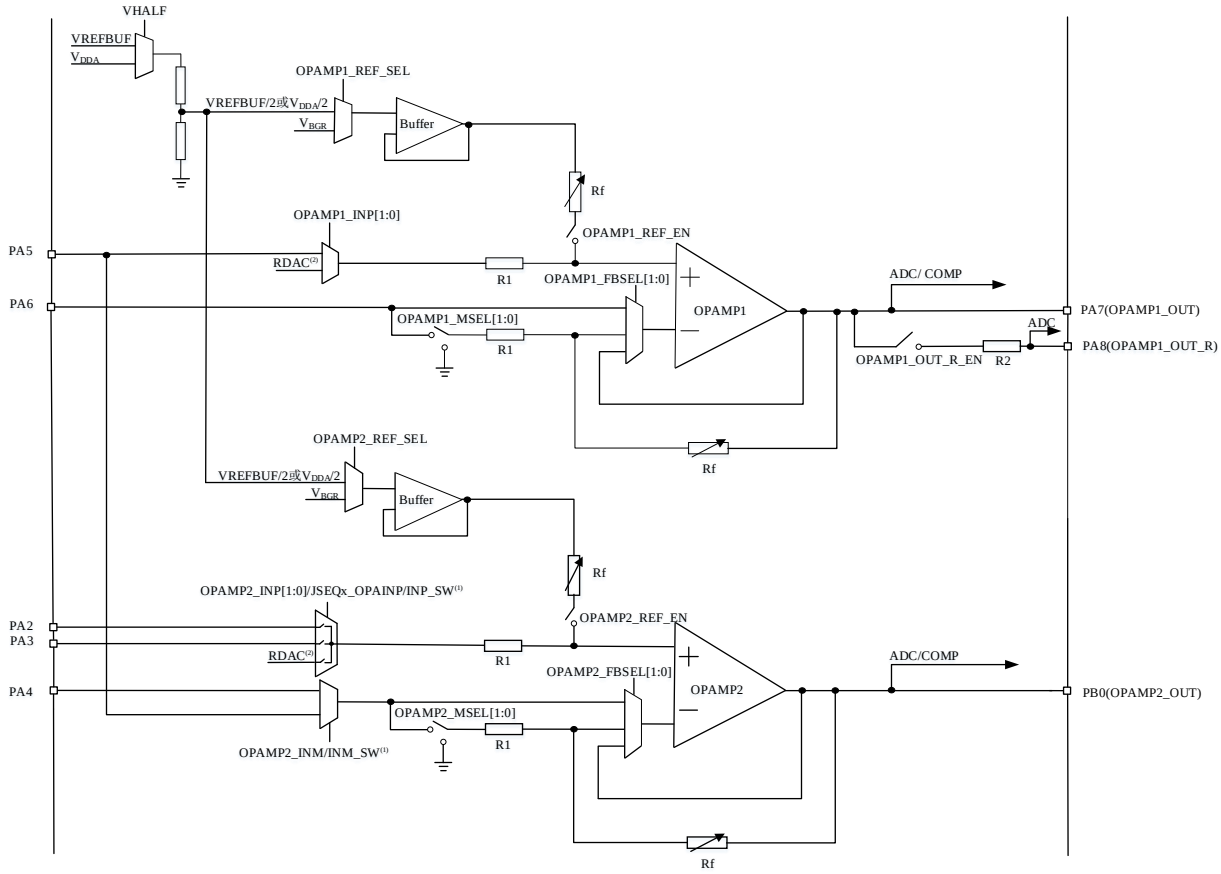
- 支持独立写保护

## 16.3 OPAMP 功能描述

使用 OPAMP 之前需使能 BGR，并等待其稳定（$t_{START}$ 启动稳定时间参见数据手册），详见*内部带隙基准电压。*

### 16.3.1 OPAMP 结构框图

图 16-1 OPAMP 框图




1. OPAMP2 支持由 ADC 或 TIMx 控制的输入通道分时切换，从而实现两路 OPAMP 效果。
2. RDAC 配置请参见：*COMP 控制寄存器（COMP_CR）。*

### 16.3.2 OPAMP 引脚和内部信号

OPAMP 输入可选择 RDAC 输出电压、GPIO，使用 GPIO 时必须将其配置为模拟模式。

OPAMP 输出可连接到 GPIO，也可以作为 COMP 和 ADC 输入。

表 16-1 OPAMP1 正相输入信号

| OPAMP1 正相输入 | INP[1:0] |
|---|---|
| 输入浮空 | 00 |
| RDAC | 01 |
| PA5 | 10 |

表 16-2 OPAMP1 反相输入信号

| OPAMP1 反相输入 | - |
|---|---|
| PA6 | - |

表 16-3 OPAMP2 正相输入信号

| OPAMP2 正相输入 | INP[1:0] |
|---|---|
| 输入浮空 | 00 |
| RDAC | 01 |
| PA2 | 10 |
| PA3 | 11 |

表 16-4 OPAMP2 反相输入信号

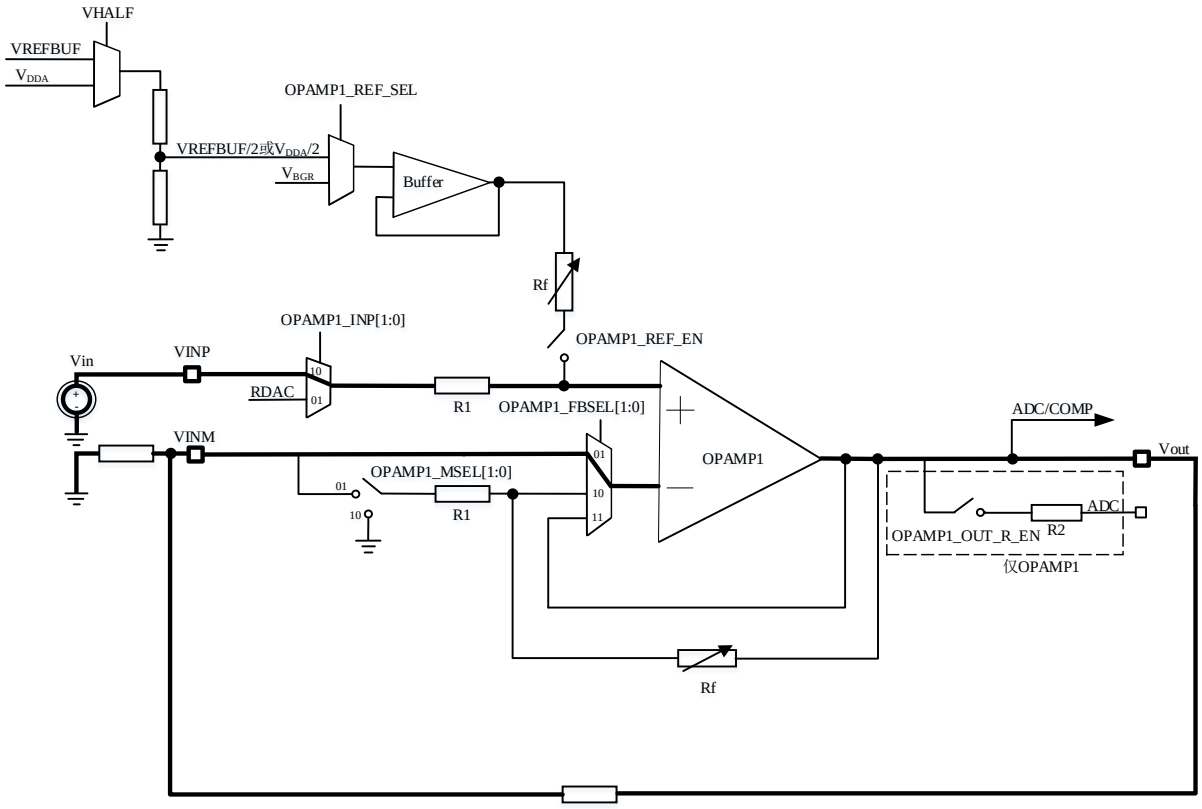
| OPAMP2 反相输入 | INM |
|---|---|
| PA4 | 0 |
| PA5 | 1 |

### 16.3.3 独立运放模式

独立运放模式下 OPAMP 的输入、输出都直接连接到 GPIO，通过在片外连接反馈电阻，用户可以灵活调节运放的负反馈增益。

独立运放模式应用示例见下图，示例为正相放大模式，正相输入选择 VINP，反相输入选择 VINM，输出 VOUT 通过片外反馈电阻连接到 VINM。

图 16-2　独立运放模式



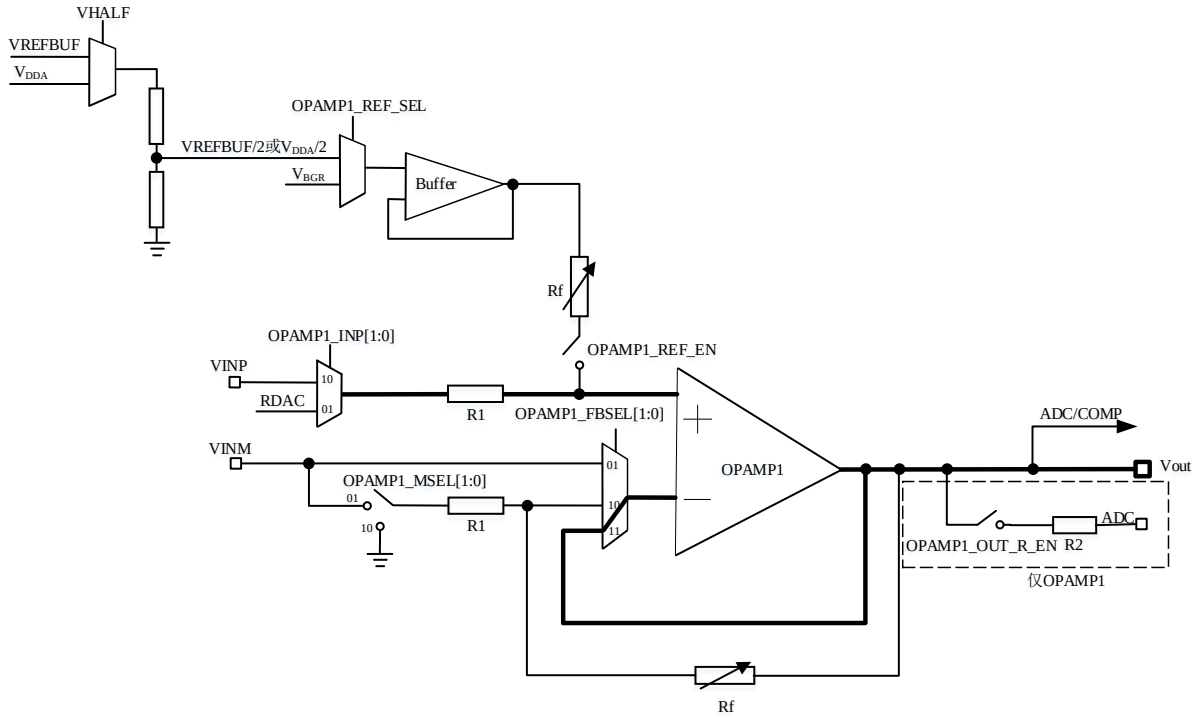

独立运放模式配置步骤如下：

1) 选择正反相输入：配置 OPAMPx_CR 寄存器的 INP[1:0]及 INM 位域；
2) 独立运放模式：配置 OPAMPx_CR 寄存器的 FBSEL[1:0]为 01；
3) 断开 PGA 模式反馈电阻开关：配置 OPAMPx_CR 寄存器的 MSEL[1:0]为 00；
4) 禁止正相偏置：配置 OPAMPx_CR 寄存器的 REF_EN 为 0；
5) 使能 OPAMP：配置 OPAMPx_CR 寄存器中的 EN 位为 1。

### 16.3.4 电压跟随模式

电压跟随模式应用示例见下图。正相输入可选择 VINP 或 RDAC，无需连接反相输入引脚，输出 VOUT 直接连接到反相输入端。

图 16-3 电压跟随模式



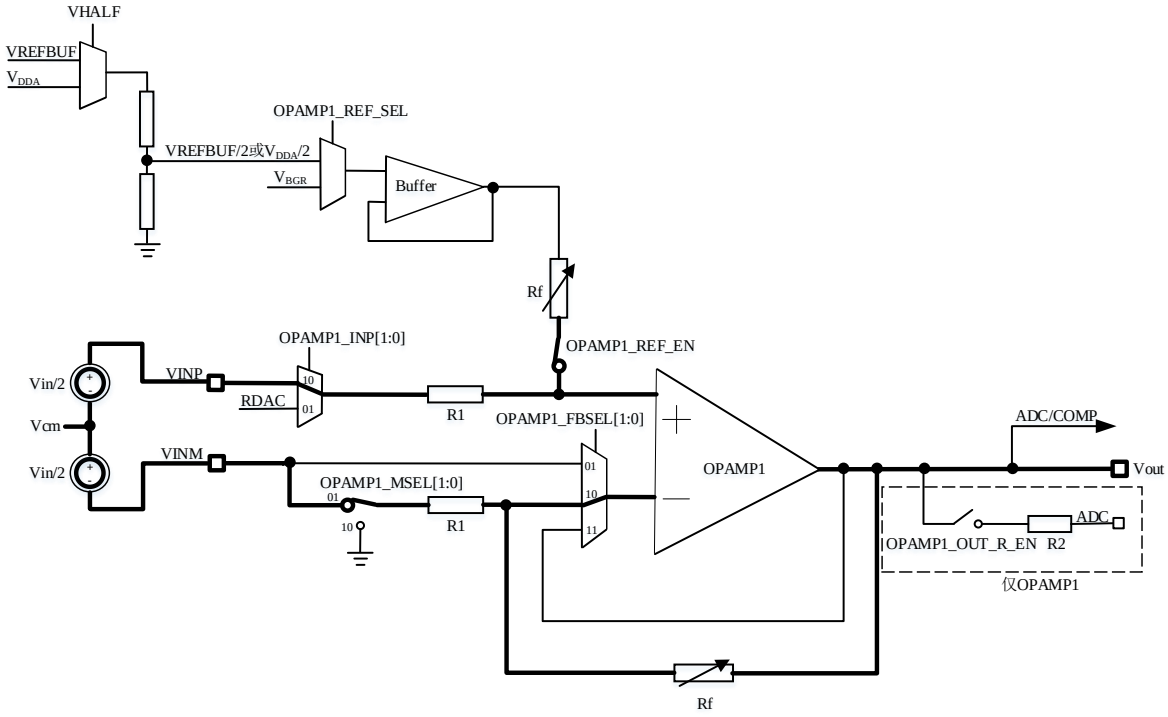

电压跟随模式配置步骤如下：

1) 选择正相输入：配置 OPAMPx_CR 寄存器的 INP[1:0]（选择 RDAC 可实现 RDAC 电压输出）；
2) 电压跟随模式：配置 OPAMPx_CR 寄存器的 FBSEL[1:0]为 11；
3) 断开 PGA 模式反馈电阻开关：配置 OPAMPx_CR 寄存器的 MSEL[1:0]为 00；
4) 禁止正相偏置：配置 OPAMPx_CR 寄存器的 REF_EN 为 0；
5) 使能 OPAMP：配置 OPAMPx_CR 寄存器中的 EN 位为 1。

### 16.3.5 伪差分 PGA 模式

伪差分 PGA 模式应用示例见下图，图中正相输入选择 VINP，反相输入选择 VINM，输出 VOUT 通过内部反馈电阻连接到反相输入端，正相偏置控制开关闭合。伪差分 PGA 增益可配置为 x1、x3、x7、x11、x15。

图 16-4 伪差分 PGA 模式




使用伪差分 PGA 模式时，需要考虑通道阻抗 $R_{CH}$（内部开关和 GPIO 通道），该阻抗会影响 PGA 增益，具体计算方式参见如下示例：

选择伪差分 PGA 增益 GAIN=15，则 $R_f$=450KΩ，$R_1$=30KΩ（$R_f$/$R_1$ 参见数据手册 OPAMP 特性），$R_{CH}$=300Ω（$R_{CH}$ 参见数据手册 OPAMP 特性），实际增益 GAIN：

$$\mathrm{GAIN} = \frac{R_f}{R_1 + R_{CH}} \approx 14.85$$

伪差分 PGA 模式配置步骤如下：

1) 选择 GPIO 作为正反相输入：配置 OPAMPx_CR 寄存器的 INP[1:0]及 INM 位域；
2) PGA 模式：配置 OPAMPx_CR 寄存器的 FBSEL[1:0]为 10，
3) PGA 模式反馈电阻连接 GPIO：配置 OPAMPx_CR 寄存器的 MSEL[1:0]为 01；
4) 使能正相偏置：配置 OPAMPx_CR 寄存器的 REF_EN 为 1；
5) 配置增益：配置 OPAMPx_CR 寄存器的 PGA_GAIN[2:0]；
6) 使能 OPAMP：配置 OPAMPx_CR 寄存器中的 EN 位为 1。

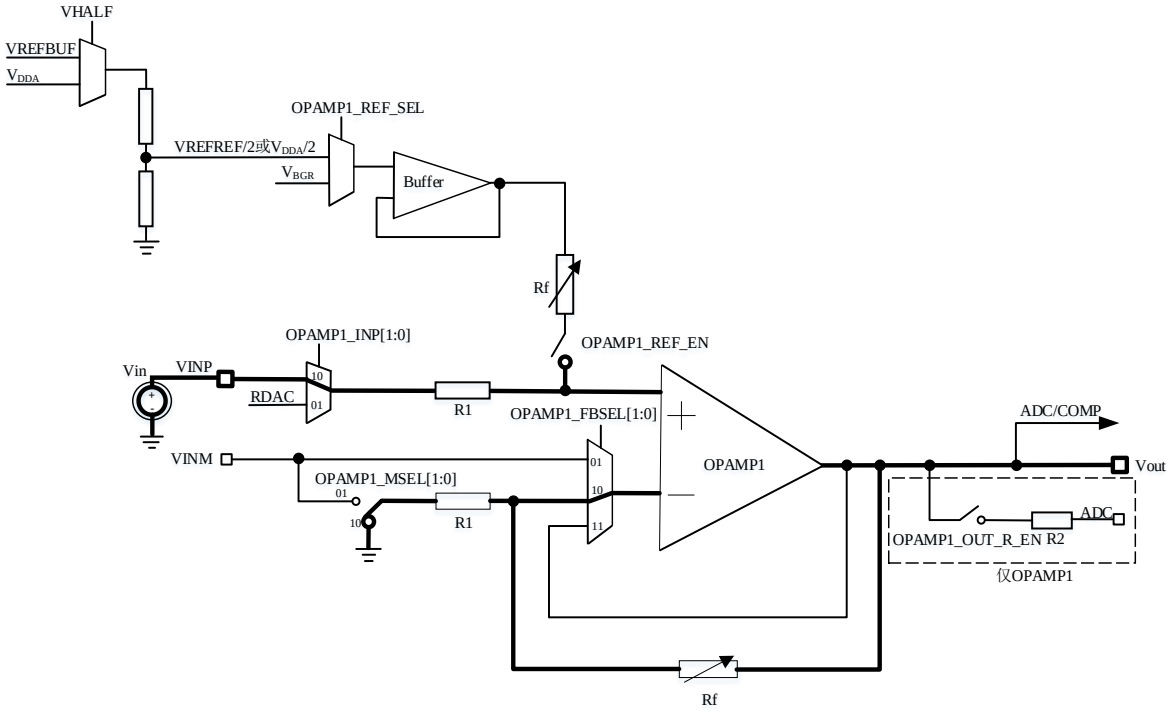
### 16.3.6 单端 PGA 模式

**同相 PGA 模式**

PGA 模式下，通过调整片内电阻阻值，可以实现固定增益的放大效果，无需连接片外反馈电阻。同相 PGA 增益可配置为 x2、x4、x8、x12、x16。

同相 PGA 模式应用示例见下图，正相输入选择 VINP，无需连接反相输入引脚，输出 VOUT 通过内部反馈电阻连接到反相输入端，正相偏置控制开关断开。

图 16-5 同相 PGA 模式




同相 PGA 模式配置步骤如下：

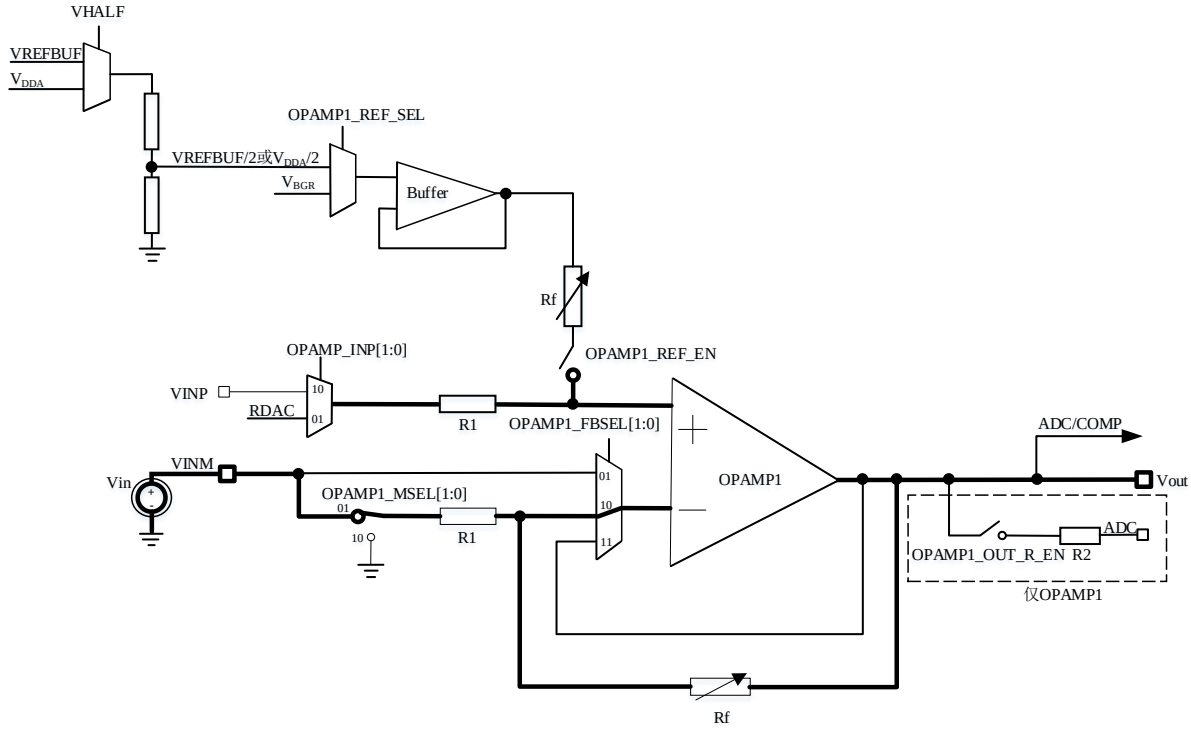
1) 选择正相输入：配置 OPAMPx_CR 寄存器的 INP[1:0]；
2) PGA 模式：配置 OPAMPx_CR 寄存器的 FBSEL[1:0]为 10；
3) PGA 模式反馈电阻接地：配置 OPAMPx_CR 寄存器的 MSEL[1:0]为 10；
4) 根据需要选择正相偏置：如果选择片内偏置，配置 OPAMPx_CR 寄存器的 REF_EN 为 1；
5) 配置增益：配置 OPAMPx_CR 寄存器的 PGA_GAIN[2:0]；
6) 使能 OPAMP：配置 OPAMPx_CR 寄存器中的 EN 位为 1。

**反相 PGA 模式**

反相 PGA 模式可以通过 OPAx_REF、RDAC 或外部电源在 OPAMP 正端施加一个直流偏置。反相 PGA 增益可配置为 x1、x3、x7、x15。

反相 PGA 模式应用示例见下图，反相输入通过 VIM 引脚连接内部反馈电阻，输出 VOUT 通过内部反馈电阻连接到反相输入端，根据需要选择正相偏置。

图 16-6 反相 PGA 模式




使用反相 PGA 模式时，需要考虑通道阻抗 $R_{CH}$（内部开关和 GPIO 通道），$R_{CH}$ 参见数据手册 OPAMP 特性。PGA 增益计算参见*伪差分 PGA 模式*。

反相 PGA 模式配置步骤如下：

1) 选择反相输入：配置 OPAMPx_CR 寄存器的 INM 位域；
2) PGA 模式：配置 OPAMPx_CR 寄存器的 FBSEL[1:0]为 10；
3) PGA 模式反馈电阻连接 GPIO：配置 OPAMPx_CR 寄存器的 MSEL[1:0]为 01；
4) 配置正相偏置：根据需要选择 RDAC、外部电压输入或 OPAx_REF_SEL 作为正相偏置；
5) 配置增益：配置 OPAMPx_CR 寄存器的 PGA_GAIN[2:0]；
6) 使能 OPAMP：配置 OPAMPx_CR 寄存器中的 EN 位为 1。

### 16.3.7 OPAMP 校准

OPAMP 支持用户失调电压校准，以减小 OPAMP 输入失调电压。OPAMP 功耗模式的高档位、中高档位和中低档位支持校准，且不同功耗模式的校准参数不同，用户根据精度需求可选择校准。

校准步骤如下：

1) OPAMP 正相输入 GPIO 断开：配置 OPAMPx_CR 寄存器的 INP[1:0]为 00；
2) OPAMP 反相输入 GPIO 断开：配置 OPAMPx_CR 寄存器的 FBSEL[1:0]为 00；
3) 断开 PGA 模式反馈电阻开关：配置 OPAMPx_CR 寄存器的 MSEL[1:0]为 00；
4) 选择正相偏置电压：配置 OPAMP_CR 寄存器 VHALF 和 OPAMPx_CR 寄存器 REF_SEL，根据输入共模电压范围，选择正向偏置电压（$V_{BGR}$、$V_{DDA}/2$ 或 VREFBUF/2）；
5) 使能正相偏置：配置 OPAMPx_CR 寄存器的 REF_EN 为 1；
6) 配置功耗模式：配置 OPAMPx_CR 寄存器的 IDD_SEL[1:0]；
7) 使能校准功能：配置 OPAMPx_CR 寄存器中 CAL_EN 位为 1；
8) 使能 OPAMP：配置 OPAMPx_CR 寄存器中的 EN 位为 1；
9) 设置 OPAMPx_CAL 寄存器中 OPA_CAL[4:0]位域初始值为 10000B，延时校准时间 $t_{OFFCAL}$（$t_{OFFCAL}$ 参见数据手册 OPAMP 特性），判断 OPAMPx_SR 寄存器中 CAL_OUT 状态位：

   - 如果 CAL_OUT= 1，校准寄存器 OPA_CAL[4:0]逐步递增，延时 $t_{OFFCAL}$ 后连续读取 CAL_OUT 状态位 10 次；
     - 若 10 次 CAL_OUT 均为 1，继续递增 OPA_CAL[4:0]，重复读取与判断流程；
     - 若 10 次 CAL_OUT 均为 0，OPA_CAL[4:0]上一次值作为最终校准值；
     - 若 10 次 CAL_OUT 有 0 有 1，当前 OPA_CAL[4:0]作为最终校准值；
   - 如果 CAL_OUT = 0，校准寄存器 OPA_CAL[4:0]逐步递减，延时 tOFFCAL 后连续读取 CAL_OUT 状态位 10 次；
     - 若 10 次 CAL_OUT 均为 0，继续递减 OPA_CAL[4:0]，重复读取与判断流程；
     - 若 10 次 CAL_OUT 均为 1，OPA_CAL[4:0]上一次值作为最终校准值；
     - 若 10 次 CAL_OUT 有 0 有 1，当前 OPA_CAL[4:0]作为最终校准值；

### 16.3.8 TIM 控制 OPAMP 输入切换

支持使用 TIM1_OC7_OUT、TIM1_OC8_OUT、TIM3_TRIG_OUT 和

TIM4_TRIG_OUT 对 OPAMP2 进行输入切换。将 OPAMP2_CFG 寄存器中的 TCMEN 位置 1 使能该功能，通过 OPAMP2_CFG 寄存器中的 TIM_SEL[1:0]选择用于切换的定时器输入源。

使能后，选择的定时器输入源为高电平时，OPAMP2 选择 INP_SW 和 INM_SW 作为正反相输入；选择的定时器输入源为低电平时，OPAMP2 选择 INP 和 INM 作为正反相输入。禁止时，OPAMP2 选择 INP 和 INM 作为正反相输入。

### 16.3.9 ADC 控制 OPAMP 输入切换

OPAMP2 支持通过 ADC 控制正相输入切换，具体参见 *OPAMP 通道切换*。

- 当 ADC 切换控制使能且转换到相应通道时，OPAMP2 选择 *ADC OPA 控制寄存器（ADC_OPACR）*中配置的正相输入；
- ADC 切换控制不使能时，若 TCMEN 使能，则 OPAMP2 通过 TIM 的对应输出信号进行输入切换，若 TCMEN 不使能，选择 INP[1:0]和 INM 作为正反相输入。

## 16.4 OPAMP 寄存器

OPAMP 寄存器支持 32 位访问。

表 16-5 OPAMP 基地址

| 外设 | 基地址 |
|---|---|
| OPAMP | 0x4001 0300 |

### 16.4.1 OPAMP 控制寄存器（OPAMP_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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:1 | 保留 | 写入无效 |
| 0 | VHALF | OPAMP的1/2分压参考电压源选择<br>0：$V_{DDA}$<br>1：VREFBUF |

### 16.4.2 OPAMP 写使能寄存器（OPAMP_LOCK）

偏移地址：0x04

复位值：0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 写入无效 |
| 0 | LOCK | OPAMP写使能控制<br>对OPAMP_LOCK寄存器低8位写入0x55时，该位清0，使能OPAMP寄存器写操作，写入非0x55的值时，该位置1，锁定OPAMP寄存器写操作。<br>0：OPAMP寄存器可写<br>1：OPAMP寄存器不可写 |

### 16.4.3 OPAMP1 控制寄存器（OPAMP1_CR）

偏移地址：0x10

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CAL_EN | Res. | | OUT_R_EN | Res. | REF_SEL | Res. | REF_EN | Res. | | MSEL[1:0] | | Res. | | FBSEL[1:0] | |
| rw | | | rw | | rw | | rw | | | rw | rw | | | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | PGA_GAIN[2:0] | | | Res. | | INP[1:0] | | Res. | | | | IDD_SEL[1:0] | | Res. | EN |
| | rw | rw | rw | | | rw | rw | | | | | rw | rw | | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31 | CAL_EN | OPAMP1校准使能<br>0：禁止<br>1：使能 |
| 30:29 | 保留 | 写入无效 |
| 28 | OUT_R_EN | OPAMP1输出到端口串联电阻控制<br>0：输出不串联电阻<br>1：输出串联电阻 |
| 27 | 保留 | 写入无效 |
| 26 | REF_SEL | OPAMP1正相偏置电压选择<br>0：$V_{BGR}$<br>1：VREFBUF/2或$V_{DDA}$/2 |

| | | |
|---|---|---|
| 25 | 保留 | 写入无效 |
| 24 | REF_EN | OPAMP1正相偏置使能<br>0：禁止<br>1：使能 |
| 23:22 | 保留 | 写入无效 |
| 21:20 | MSEL[1:0] | OPAMP1的PGA模式选择<br>00：反馈电阻开关断开<br>01：反馈电阻连接GPIO<br>10：反馈电阻连接GND<br>11：保留（写入无效保留上一次值） |
| 19:18 | 保留 | 写入无效 |
| 17:16 | FBSEL[1:0] | OPAMP1工作模式选择<br>00：反相输入端口浮空<br>01：独立运放模式<br>10：PGA模式<br>11：电压跟随模式 |
| 15 | 保留 | 写入无效 |
| 14:12 | PGA_GAIN[2:0] | PGA增益选择<br>同相PGA增益：<br>000：PGA增益GAIN = 2<br>001：PGA增益GAIN = 4<br>010：PGA增益GAIN = 8<br>011：PGA增益GAIN = 12<br>100：PGA增益GAIN = 16<br>其他：保留<br><br>伪差分/反相PGA增益：<br>000：PGA增益GAIN = 1<br>001：PGA增益GAIN = 3 |

| | | |
|---|---|---|
| | | 010：PGA增益GAIN = 7<br>011：PGA增益GAIN = 11<br>100：PGA增益GAIN = 15<br>其他：保留 |
| 11:10 | 保留 | 写入无效 |
| 9:8 | INP[1:0] | OPAMP1正相输入选择<br>00：端口浮空<br>01：RDAC<br>10：PA5<br>11：保留（写入无效保留上一次值） |
| 7:4 | 保留 | 写入无效 |
| 3:2 | IDD_SEL [1:0] | OPAMP1功耗模式选择<br>00：高档位<br>01：中高档位<br>10：中低档位<br>11：低档位 |
| 1 | 保留 | 写入无效 |
| 0 | EN | OPAMP1使能<br>0：禁止<br>1：使能 |

### 16.4.4 OPAMP1 校准寄存器（OPAMP1_CAL）

偏移地址：0x14

复位值：0x0000 0010

*注意：　POR/PDR 复位时，复位该寄存器。*

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

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:6 | 保留 | 写入无效 |
| 4:0 | OPA_CAL[4:0] | OPAMP1 的校准值 |

### 16.4.5 OPAMP1 状态寄存器（OPAMP1_SR）

偏移地址：0x18

复位值：0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:1 | 保留 | 写入无效 |
| 0 | CAL_OUT | OPAMP1校准模式下的输出 |

### 16.4.6 OPAMP2 控制寄存器（OPAMP2_CR）

偏移地址：0x30

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CAL_EN | Res. | | | | REF_SEL | Res. | REF_EN | Res. | | MSEL[1:0] | | Res. | | FBSEL[1:0] | |
| rw | | | | | rw | | rw | | | rw | rw | | | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | PGA_GAIN[2:0] | | | Res. | | INP[1:0] | | Res. | | | INM | IDD_SEL[1:0] | | Res. | EN |
| | rw | rw | rw | | | rw | rw | | | | rw | rw | rw | | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31 | CAL_EN | OPAMP2校准使能<br>0：禁止 |

| | | |
|---|---|---|
| | | 1：使能 |
| 30:27 | 保留 | 写入无效 |
| 26 | REF_SEL | OPAMP2正端偏置电压选择<br>0：$V_{BGR}$<br>1：VREFBUF/2或$V_{DDA}$/2 |
| 25 | 保留 | 写入无效 |
| 24 | REF_EN | OPAMP2正相偏置使能<br>0：禁止<br>1：使能 |
| 23:22 | 保留 | 写入无效 |
| 21:20 | MSEL[1:0] | OPAMP2的PGA模式选择<br>00：反馈电阻开关断开<br>01：反馈电阻连接GPIO<br>10：反馈电阻连接GND<br>11：保留（写入无效保留上一次值） |
| 19:18 | 保留 | 写入无效 |
| 17:16 | FBSEL[1:0] | OPAMP2工作模式选择<br>00：反相输入端口浮空<br>01：独立运放模式<br>10：PGA模式<br>11：电压跟随模式 |
| 15 | 保留 | 写入无效 |
| 14:12 | PGA_GAIN[2:0] | PGA增益选择<br>同相PGA增益：<br>000：PGA增益GAIN = 2<br>001：PGA增益GAIN = 4<br>010：PGA增益GAIN = 8 |

| | | |
|---|---|---|
| | | 011：PGA增益GAIN = 12<br>100：PGA增益GAIN = 16<br>其他：保留<br><br>伪差分/反相PGA增益：<br>000：PGA增益GAIN = 1<br>001：PGA增益GAIN = 3<br>010：PGA增益GAIN = 7<br>011：PGA增益GAIN = 11<br>100：PGA增益GAIN = 15<br>其他：保留 |
| 11:10 | 保留 | 写入无效 |
| 9:8 | INP[1:0] | OPAMP2正相选择<br>使用定时器切换OPAMP2输入且定时器输出信号无效，或不使能定时器切换输入时，选择此位域配置的正相输入。<br>00：端口浮空<br>01：RDAC<br>10：PA2<br>11：PA3 |
| 7:5 | 保留 | 写入无效 |
| 4 | INM | OPAMP2反相选择<br>使用定时器切换OPAMP2输入且定时器输出信号无效，或不使能定时器切换输入时，选择此位域配置的反相输入。<br>0：PA4<br>1：PA5 |
| 3:2 | IDD_SEL [1:0] | OPAMP2功耗模式选择<br>00：高档位<br>01：中高档位<br>10：中低档位<br>11：低档位 |
| 1 | 保留 | 写入无效 |

| | | |
|---|---|---|
| 0 | EN | OPAMP2使能<br>0：禁止<br>1：使能 |

### 16.4.7 OPAMP2 校准寄存器（OPAMP2_CAL）

偏移地址：0x34

复位值：0x0000 0010

*注意：　POR/PDR 复位时，复位该寄存器。*

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:6 | 保留 | 写入无效 |
| 4:0 | OPA_CAL[4:0] | OPAMP2 的校准值 |

### 16.4.8 OPAMP2 状态寄存器（OPAMP2_SR）

偏移地址：0x38

复位值：0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:1 | 保留 | 写入无效 |
| 0 | CAL_OUT | OPAMP2校准模式下的输出 |

### 16.4.9 OPAMP2 切换配置寄存器（OPAMP2_CFG）

偏移地址：0x3C

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | INP_SW | Res. | | | INM_SW | Res. | TIM_SEL[1:0] | | TCMEN |
| | | | | | | | rw | | | | rw | | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:9 | 保留 | 写入无效 |
| 8 | INP_SW | 正相输入信号定时器切换选择<br>使用定时器切换OPAMP2输入时，定时器输出信号有效时选择此位域配置的正相输入。<br>0：PA2<br>1：PA3 |
| 7:5 | 保留 | 写入无效 |
| 4 | INM_SW | 反相输入信号定时器切换选择<br>使用定时器切换OPAMP2输入时，定时器输出信号有效时选择此位域配置的反相输入。<br>0：PA4<br>1：PA5 |
| 3 | 保留 | 写入无效 |
| 2:1 | TIM_SEL[1:0] | OPAMP2正反相切换定时器输入源选择<br>00：TIM1_OC7_OUT<br>01：TIM1_OC8_OUT<br>10：TIM3_TRIG_OUT<br>11：TIM4_TRIG_OUT |
| 0 | TCMEN | 定时器切换OPAMP2输入使能 |

0：禁止

1：使能

# 17 比较器（COMP）

## 17.1 简介

芯片内置 2 个模拟电压比较器 COMP1 和 COMP2，可独立使用，可组合实现窗口比较功能。比较器可用于多种功能，包括以下典型应用：

- 模拟信号触发从低功耗模式唤醒
- 与定时器结合使用，作为刹车输入、OCREF_CLR 事件、捕获事件输入

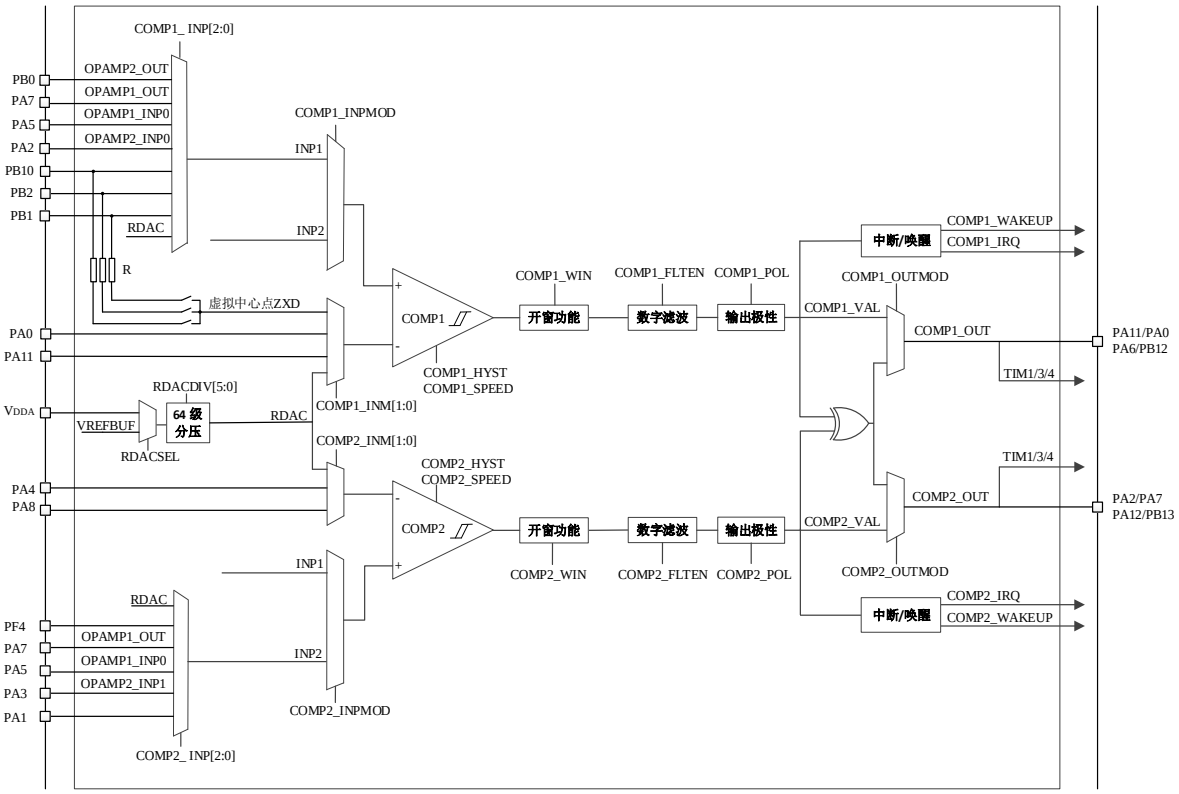
## 17.2 COMP 主要特性

- 输入电压范围 0 ~ $V_{DDA}$-0.6V
- 速度、功耗可配置，最小功耗为 120nA
- 输入信号可灵活配置：
  - 具备 COMP 输入功能的 I/O
  - RDAC（$V_{DDA}$ 或 VREFBUF 的 64 级分压）
  - OPAMP1/2 的输入、输出
- 支持虚拟中心点，用于反电势过零点检测
- 可配置的开窗控制功能
- 输出信号可连接到 GPIO、TIM1、TIM3、TIM4，与 TIM1/3/4 的互联，用于刹车输入、OCREF_CLR 事件、捕获事件输入
- 支持窗口比较功能
- 具备迟滞功能
- 可配置的滤波时钟和滤波时间，防止误触发
- 可选输出极性
- 3 种中断触发方式：上升沿、下降沿或双沿触发中断
- 支持低功耗模式唤醒

## 17.3 COMP 功能描述

### 17.3.1 COMP 结构框图

图 17-1 COMP 框图




### 17.3.2 COMP 引脚和内部信号

比较器输入可以选择 RDAC 输出电压、OPAMP1/2 的正相输入/输出，也可以选择 GPIO 信号，但必须将其配置为模拟模式；COMP1/2 正相输入模式位 INPMOD 同时置 1 时，可交换 COMP1/2 的正相输入。

比较器输出可以连接到 GPIO，也可以在片内连接到 TIM1/3/4 作为输入，用于输入捕获、刹车输入、OCREF_CLR 事件等典型应用；可将比较器输出同时连接到 GPIO 和片内外设。

表 17-1 COMP1 正相输入信号

| COMP1 正相输入 | COMP1_INP[2:0] |
|---|---|
| RDAC | 000 |
| PB1 | 001 |
| PB2 | 010 |
| PB10 | 011 |

| COMP1 正相输入 | COMP1_INP[2:0] |
|---|---|
| OPAMP1_INP0(PA5) | 100 |
| OPAMP1_OUT(PA7) | 101 |
| OPAMP2_OUT(PB0) | 110 |
| OPAMP2_INP0(PA2) | 111 |

表 17-2 COMP1 反相输入信号

| COMP1 反相输入 | COMP1_INM[1:0] |
|---|---|
| RDAC | 00 |
| 虚拟中心点 | 01 |
| PA11 | 10 |
| PA0 | 11 |

表 17-3 COMP2 正相输入信号

| COMP2 正相输入 | COMP2_INP[2:0] |
|---|---|
| RDAC | 000 |
| PF4 | 001 |
| PA1 | 010 |
| OPAMP2_INP1(PA3) | 011 |
| OPAMP1_INP0(PA5) | 100 |
| OPAMP1_OUT(PA7) | 101 |

表 17-4 COMP2 反相输入信号

| COMP2 反相输入 | COMP2_INM[1:0] |
|---|---|
| RDAC | 00 |
| PA4 | 01 |
| PA8 | 10 |

### 17.3.3 使能/禁止控制

比较器使能/禁止通过 COMPx_CSR1 寄存器中 EN 位控制，EN 位置 1 使能 COMPx，EN 位清零禁止 COMPx。

### 17.3.4 窗口比较功能

比较器支持窗口比较功能，用于监控输入电压是否处于所设置的上/下阈值范围之内。此时这两个比较器的正相输入连接到一起，用于输入待监控的模拟电压；而两个比较器的反相分别输入上/下阈值电压。

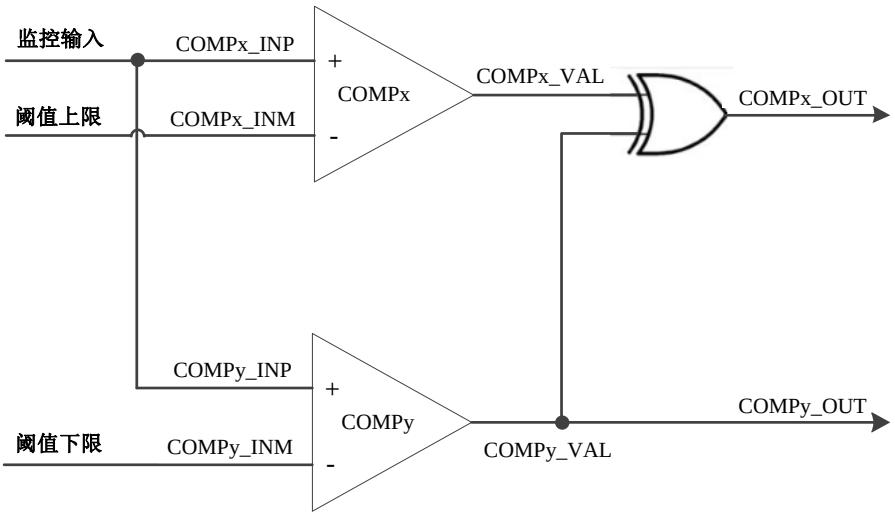
使用窗口比较功能时，将其中一个 COMPy 的寄存器 INPMOD 位设置为 1，使

其正相输入信号选择 COMPx_INP 对应的配置，从而两个比较器正相输入选择同一输入信号。此外，配置 COMPx 的寄存器 OUTMOD 位为 1，可将两个比较器的输出状态 VAL（滤波和极性控制结果）进行异或计算后输出：

- COMPx_OUT 为 COMPx_VAL 和 COMPy_VAL 的异或值。
- COMPy_OUT 与 COMPy_VAL 相同。

比较器窗口比较功能如下图所示：

图 17-2 窗口比较功能示意图




比较器输出信号模式（OUTMOD 位）与窗口比较功能不强制关联：即在窗口比较功能时两个比较器的输出信号也可以相互独立，此时可由软件对两个比较器的输出信号进行处理；同时非窗口比较功能时，也可以设置两个比较器结果异或输出，由软件进行处理。

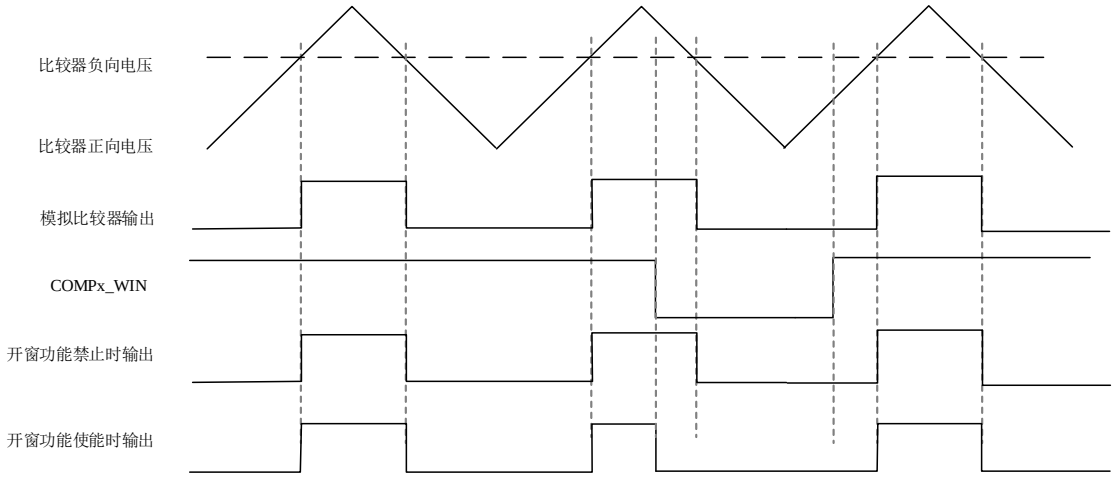
### 17.3.5 开窗功能

通过 COMPx_CSR2 寄存器的 WINSEL[2:0]位域选择比较器开窗信号 COMPx_WIN 的输入来源，并控制开窗功能的使能或禁止。

使能开窗功能后，当 COMPx_WIN 信号为高时，开窗控制模块的输出为模拟比较器输出结果，当 COMPx_WIN 信号为低时，开窗控制的输出保持为 0。

禁止开窗功能时，模拟比较器的输出信号直接输出。

图 17-3 开窗功能控制示意图




### 17.3.6 输出极性

比较器输出极性可以通过寄存器的 POL 位选择，比较器输出状态 VAL 如下：

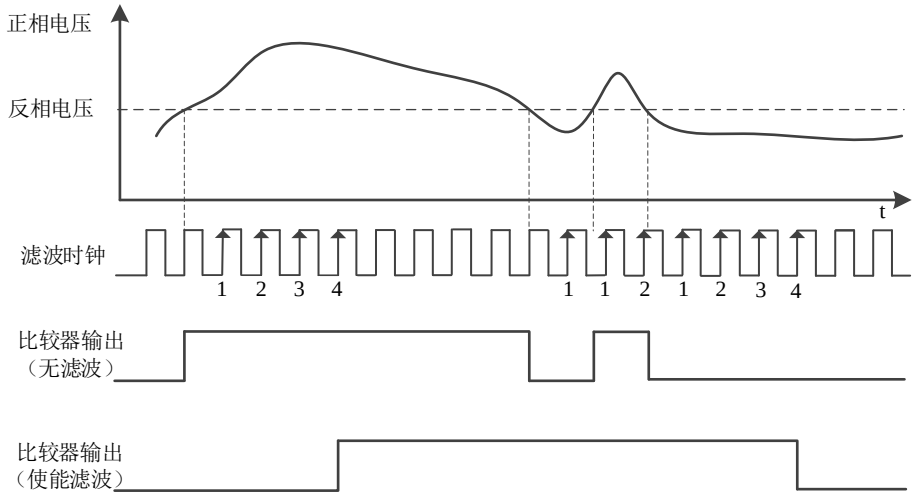
- 寄存器 POL 位置 0 时，比较器输出状态不反相。
  - 正相输入电压大于反相输入电压，输出状态 VAL 为 1；反之为 0。
- 寄存器 POL 位置 1 时，比较器输出状态反相。
  - 正相输入电压大于反相输入电压，输出状态 VAL 为 0；反之为 1。

### 17.3.7 数字滤波

比较器具有数字滤波功能，用于滤除系统噪声。将寄存器 COMPx_CSR1 中 FLTEN 位置 1 使能数字滤波功能，滤波时间由 FLTIME[2:0]位域配置，滤波时钟源由 FLTCLK[1:0]位域选择，可以选择 PCLK、RCL 或 LXTAL 对比较器原始输出状态进行滤波；选择 RCL 或 LXTAL 作为滤波时钟源时，可在 Stop 低功耗模式下正常工作。

比较器输出滤波时间设置为 4 个时钟周期的示意图如下：

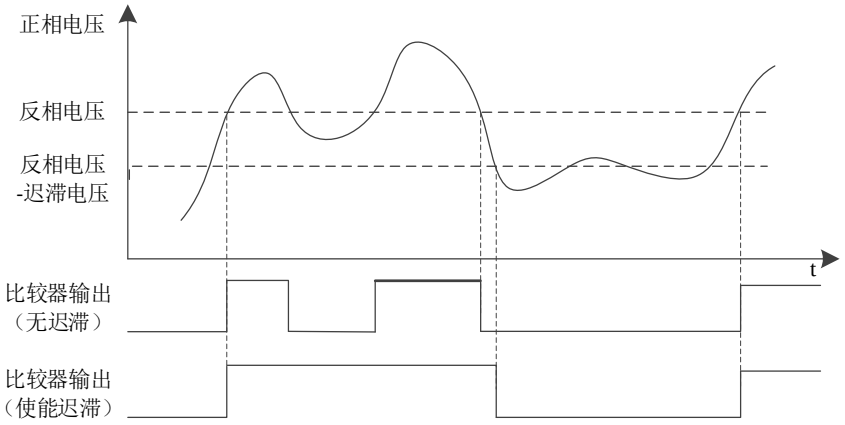
图 17-4 COMP 输出滤波




### 17.3.8 迟滞

比较器具有迟滞功能，可避免在输入信号有噪声时输出产生非预期变化。通过 HYST 位设置 30mV 迟滞电压值。比较器迟滞功能可在不需要时禁止，以便使用外部电阻搭建迟滞电路，设置迟滞值。比较器迟滞功能示意图如下：

图 17-5 COMP 迟滞




### 17.3.9 速度和功耗

比较器响应速度是从正反相两端输入电压变化到输出正确比较结果的时间。比较器响应速度与功耗相关，响应越快，功耗越大。通过寄存器 SPEED 位进行比较器响应速度 $t_D$ 与功耗 $I_{COMP}$ 配置（$t_D$ 和 $I_{COMP}$ 参见数据手册 COMP 特性）。

## 17.4 COMP 低功耗模式

比较器支持在低功耗模式下工作，支持将芯片从低功耗模式（Sleep 和 Stop）唤醒。

表 17-5 低功耗模式对 COMP 的影响

| 模式 | 说明 |
|---|---|
| Sleep | 比较器输出状态 COMPx_VAL 匹配中断触发边沿时，中断标志 INTF 置 1，同时可使芯片退出 Sleep 模式。 |
| Stop | 当比较器禁止滤波或滤波时钟选择 RCL、LXTAL 时，可正常工作。<br>比较器输出状态 COMPx_VAL 匹配中断触发边沿时，中断标志 INTF 置 1，同时可使芯片退出 Stop 模式。 |

## 17.5 COMP 中断

通过配置 COMPx_CSR2 寄存器中 RISEEN 位和 FALLEN 位，选择比较器输出状态 COMPx_VAL 的上升沿、下降沿或双沿触发中断/唤醒请求：

表 17-6 中断请求

| 中断事件 | 事件标志 | 中断使能位 | 事件标志/中断清除方法 |
|---|---|---|---|
| COMPx 输出状态上升沿 | INTF | IE | 向 INTF 位写 1 |
| COMPx 输出状态下降沿 | | | |
| COMPx 输出状态的双沿 | | | |

## 17.6 COMP 寄存器

COMP 寄存器支持 32 位访问。

表 17-7 COMP 基地址

| 外设 | 基地址 |
|---|---|
| COMP | 0x4001 0200 |

### 17.6.1 COMP 控制寄存器（COMP_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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RDACEN | Res. | | RDACSEL | Res. | | | | | | RDACDIV[5:0] | | | | | |
| rw | | | rw | | | | | | | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15 | RDACEN | RDAC使能<br>0：禁止<br>1：使能 |
| 14:13 | 保留 | 写入无效 |
| 12 | RDACSEL | RDAC输入电压源选择<br>0：$V_{DDA}$<br>1：VREFBUF |
| 11:7 | 保留 | 写入无效 |
| 5:0 | RDACDIV[5:0] | RDAC输出分压选择<br>000000：1/64<br>000001：2/64 |

000010：3/64

000011：4/64

⋮

111111：64/64

### 17.6.2 COMPx 控制和状态寄存器 1（COMPx_CSR1）

偏移地址：0x10+0x10 ×x，（x = 0 ~ 1）

复位值：0x0000 0000

*注意：　仅当 COMPx 禁止时，允许对此寄存器执行写操作（EN 位除外）。*

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | VAL | Res. | FLTEN | FLTIME[2:0] | | | Res. | FLTCLK[1:0] | | Res. | | | SPEED | Res. | HYST |
| | r | | 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 | OUTMOD | INPMOD | Res. | | INP[2:0] | | | Res. | | INM[1:0] | | Res. | | | EN |
| rw | rw | rw | | | rw | rw | rw | | | rw | rw | | | | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31 | 保留 | 写入无效 |
| 30 | VAL | 比较器输出状态<br>表示比较器数字滤波及极性控制后的输出结果 |
| 29 | 保留 | 写入无效 |
| 28 | FLTEN | 比较器数字滤波使能<br>0：禁止<br>1：使能 |
| 27:25 | FLTIME[2:0] | 比较器数字滤波时间选择<br>000：2个时钟周期<br>001：4个时钟周期<br>010：8个时钟周期<br>011：16个时钟周期<br>100：32个时钟周期<br>101：64个时钟周期<br>110：256个时钟周期 |

| | | |
|---|---|---|
| | | 111：1024个时钟周期 |
| 24 | 保留 | 写入无效 |
| 23:22 | FLTCLK[1:0] | 比较器数字滤波时钟源选择<br>00：PCLK<br>01：RCL<br>10：LXTAL<br>11：保留（默认为PCLK） |
| 21:19 | 保留 | 写入无效 |
| 18 | SPEED | 比较器响应速度选择<br>0：高速、高功耗<br>1：低速、低功耗 |
| 17 | 保留 | 写入无效 |
| 16 | HYST | 比较器迟滞功能控制<br>0：禁止<br>1：使能 |
| 15 | POL | 比较器输出极性选择<br>0：输出不反相<br>1：输出反相 |
| 14 | OUTMOD | 比较器输出信号（COMPx_OUT）选择<br>0：比较器输出状态（COMPx_VAL）<br>1：比较器输出状态（COMPx_VAL）和另一个比较器输出状态（COMPy_VAL）的异或结果 |
| 13 | INPMOD | 比较器正相输入信号选择，用于配置窗口比较功能<br>0：比较器x正相输入信号由本比较器的INP[2:0]位决定<br>1：比较器x正相输入信号由另一个比较器的INP[2:0]位决定 |
| 12:9 | 保留 | 写入无效 |

| | | |
|---|---|---|
| 10:8 | INP[2:0] | 比较器正相输入信号选择<br>详见*COMP 引脚和内部信号*章节。 |
| 7:6 | 保留 | 写入无效 |
| 5:4 | INM[1:0] | 比较器反相输入信号选择<br>详见*COMP 引脚和内部信号*章节。 |
| 3:1 | 保留 | 写入无效 |
| 0 | EN | 比较器使能<br>0：禁止<br>1：使能 |

### 17.6.3 COMPx 控制和状态寄存器 2（COMPx_CSR2）

偏移地址：0x14+0x10 ×x，（x = 0 ~ 1）

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IE | Res. | | | | | RISE EN | FALL EN | Res. | | | | | ZXDEN | Res. | |
| rw | | | | | | rw | rw | | | | | | rw | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:27 | 保留 | 写入无效 |
| 26:24 | WINSEL[2:0] | 比较器开窗输入COMPx_WIN来源选择<br>000：禁止开窗功能<br>001：TIM1_TRIG_OUT<br>010：TIM1_OC1_OUT<br>011：TIM1_OC7_OUT<br>100：TIM1_OC8_OUT<br>101：TIM3_TRIG_OUT<br>110：TIM4_TRIG_OUT<br>111：保留（默认禁止开窗功能） |

| | | |
|---|---|---|
| 23:16 | 保留 | 写入无效 |
| 15 | IE | 比较器中断使能<br>0：禁止<br>1：使能 |
| 14:10 | 保留 | 写入无效 |
| 9 | RISEEN | 比较器输出状态上升沿触发中断控制<br>0：禁止<br>1：使能 |
| 8 | FALLEN | 比较器输出状态下降沿触发中断控制<br>0：禁止<br>1：使能 |
| 7:3 | 保留 | 写入无效 |
| 2 | ZXDEN | 虚拟中心点使能<br>0：禁止<br>1：使能<br>*注意：此位域仅COMP1写入有效。* |
| 1:0 | 保留 | 写入无效 |

### 17.6.4 COMPx 状态寄存器（COMPx_SR）

偏移地址：0x18+0x10 ×x，（x = 0 ~ 1）

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | INTF |
| | | | | | | | | | | | | | | | rc_w1 |

| 位/位域 | 名称 | 描述 |
| --- | --- | --- |
| 31:1 | 保留 | 写入无效 |
| 0 | INTF | 比较器中断标志<br>当比较器输出状态COMPx_VAL匹配中断触发边沿时，该位硬件置1。如果COMPx_CSR2寄存器中IE位置1，则触发中断。该位写1清零，写0无效。<br>0：比较器未发生中断请求<br>1：比较器已发生中断请求 |

# 18 液晶显示控制器（LCD）

## 18.1 简介

LCD 控制器用于驱动段码式液晶屏，可驱动最大段数为 84（4COMx21SEG）。区段端子的确切数量取决于芯片封装及引脚规模。

## 18.2 LCD 主要特性

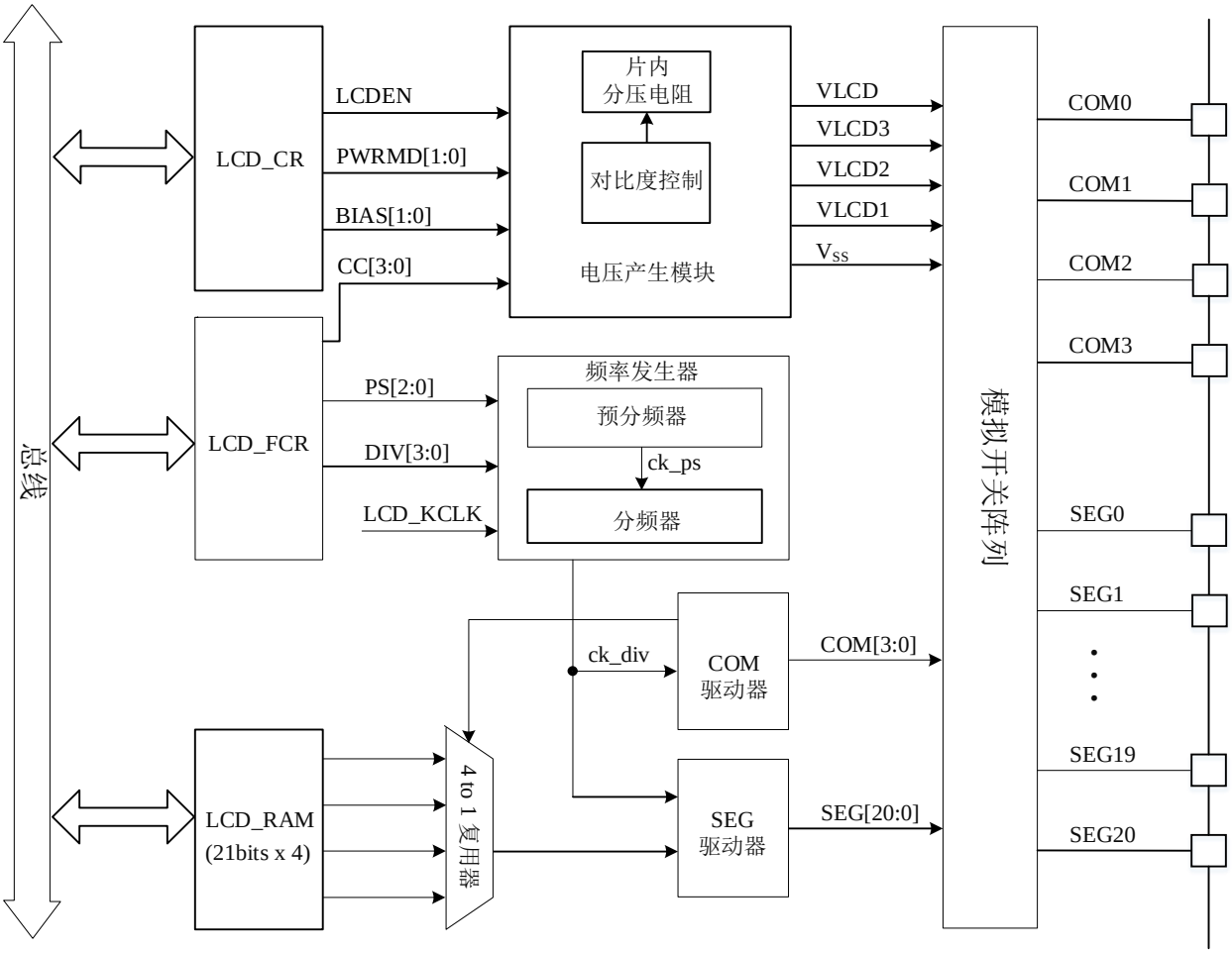
- 静态、1/2、1/3 和 1/4 duty
- 1/2、1/3 和 1/4 bias
- 对比度 16 级可调
- 支持片内 buffer 驱动模式，内置低功耗 buffer，免外部电容，提高 IO 利用率
- 支持闪烁显示，且闪烁模式和闪烁频率可配置
- 可编程的帧间死区时间
- 帧速率范围 30~100Hz
- 支持 LCD 防极化功能
- 支持 TypeB 驱动波形
- 支持低功耗模式显示

## 18.3 LCD 功能描述

### 18.3.1 LCD 结构框图

LCD 控制器主要由频率发生器、电压产生模块、公共和区段驱动器（COM/SEG）四个模块构成，如下图所示。

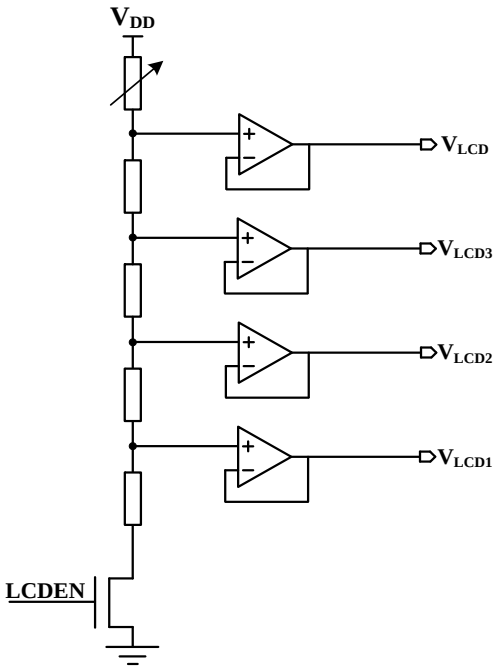
图 18-1 LCD 控制器框图




### 18.3.2 片内 buffer 驱动模式

LCD 通过片内电阻分压网络生成介于 $V_{LCD}$ 和 $V_{SS}$ 之间的中间电压，然后输入到低功耗 buffer 以增强驱动能力。可通过控制寄存器 LCD_CR 的 PWRMD[1:0]位域选择合适的 buffer 功耗档位，以平衡 LCD 功耗和显示效果。此模式无需连接外部电容器件，具有最大 IO 利用率。其结构示意图如下：

图 18-2 片内 buffer 驱动模式示意图




### 18.3.3 对比度设置

可通过帧控制寄存器 LCD_FCR 的 CC[3:0]位域，选择 $V_{LCD}$ 电压档位，实现对比度调整，如下表所示。

表 18-1 $V_{LCD}$ 输出电压配置

| CC[3:0] | 不同 $V_{DD}$ 下 $V_{LCD}$ 电压(V) | | | | |
|---|---|---|---|---|---|
| | **5** | **4.5** | **3.6** | **3.3** | **3.0** |
| 0000 | 2.74 | 2.47 | 1.97 | 1.81 | 1.64 |
| 0001 | 2.83 | 2.55 | 2.04 | 1.87 | 1.70 |
| 0010 | 2.92 | 2.63 | 2.10 | 1.93 | 1.75 |
| 0011 | 3.01 | 2.71 | 2.17 | 1.99 | 1.81 |
| 0100 | 3.12 | 2.81 | 2.25 | 2.06 | 1.87 |
| 0101 | 3.23 | 2.91 | 2.33 | 2.13 | 1.94 |
| 0110 | 3.35 | 3.02 | 2.41 | 2.21 | 2.01 |
| 0111 | 3.47 | 3.12 | 2.50 | 2.29 | 2.08 |
| 1000 | 3.61 | 3.25 | 2.60 | 2.38 | 2.17 |
| 1001 | 3.76 | 3.38 | 2.71 | 2.48 | 2.26 |
| 1010 | 3.93 | 3.54 | 2.83 | 2.59 | 2.35 |
| 1011 | 4.10 | 3.69 | 2.95 | 2.71 | 2.46 |
| 1100 | 4.30 | 3.87 | 3.10 | 2.83 | 2.58 |

| CC[3:0] | 不同 $V_{DD}$ 下 $V_{LCD}$ 电压(V) | | | | |
|---|---|---|---|---|---|
| | 5 | 4.5 | 3.6 | 3.3 | 3.0 |
| 1101 | 4.51 | 4.06 | 3.25 | 2.98 | 2.71 |
| 1110 | 4.75 | 4.28 | 3.42 | 3.14 | 2.85 |
| 1111 | 5.00 | 4.50 | 3.60 | 3.30 | 3.00 |

### 18.3.4 频率发生器

频率发生器由预分频器和分频器两部分组成，参见*图：LCD 控制器框图*。

**频率发生器时钟源**

LCD 控制器输入时钟源选择参见*外设异步时钟配置寄存器（RCC_CLKSEL）*的 LCD_SEL 位：

- LXTAL 时钟
- RCL 时钟

**频率发生器输出**

频率发生器模块可根据输入时钟 LCD_KCLK 输出预分频时钟 ck_ps 和分频时钟 ck_div，从而获得 LCD 控制器所需的各项时钟频率。时钟分频计算公式如下：

$$f_{ck_ps} = \frac{f_{LCDCLK}}{2^{PS[2:0]}}$$

$$f_{ck_div} = \frac{f_{ck_ps}}{(16 + DIV)} = \frac{f_{LCDCLK}}{2^{PS[2:0]}(16 + DIV[3:0])}$$

- 预分频时钟 ck_ps
  - 预分频时钟，用于脉冲持续时间。
- 分频时钟 ck_div
  - 作为 LCD 控制器显示驱动波形的相位时钟，参见*驱动波形*；
  - 用于产生帧速率；
  - 用于产生 LCD 控制器的闪烁显示频率，参见*闪烁显示*。

**帧速率**

LCD 帧速率与频率发生器的输出 $f_{ck_div}$ 的关系为 $Frame\ rate = f_{ck_div} \times duty$。下表举例说明典型帧速率的配置方式。

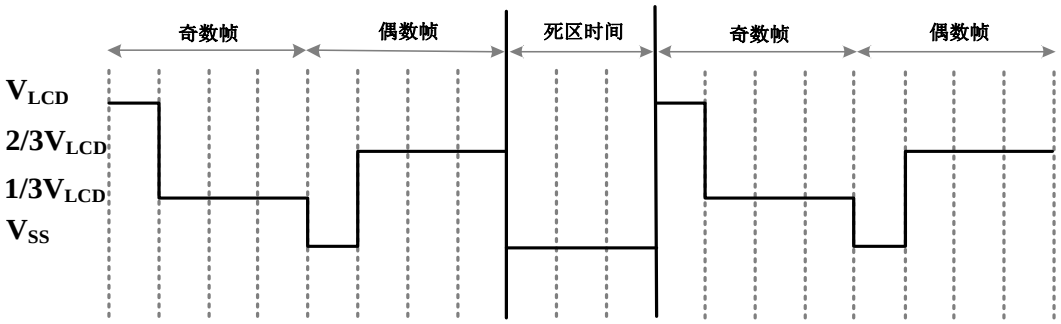
表 18-2 帧速率典型值计算示例

| Frame rate | 4COM | | 3COM | | 2COM | | 静态 | |
|---|---|---|---|---|---|---|---|---|
| | PS[2:0] | DIV[3:0] | PS[2:0] | DIV[3:0] | PS[2:0] | DIV[3:0] | PS[2:0] | DIV[3:0] |
| 30Hz | 4 | 1 | 4 | 7 | 5 | 1 | 6 | 1 |
| 51Hz | 3 | 3 | 3 | 9 | 4 | 4 | 5 | 4 |
| 57Hz | 3 | 2 | 3 | 8 | 4 | 3 | 5 | 2 |
| 64Hz | 3 | 0 | 3 | 5 | 4 | 0 | 5 | 0 |
| 68Hz | 2 | 14 | 3 | 4 | 3 | 14 | 4 | 14 |
| 76Hz | 2 | 11 | 3 | 2 | 3 | 11 | 4 | 11 |

### 18.3.5 帧间死区时间

通过帧控制寄存器 LCD_FCR 的帧间死区时间位域 DEAD[2:0]，可设置死区时间，死区时间内输出驱动电压保持为 $V_{SS}$，参见 *LCD 帧控制寄存器(LCD_FCR)*。

图 18-3 死区时间




### 18.3.6 闪烁显示

LCD 控制器在硬件上实现了闪烁显示特性，即允许一些像素以特定频率间断显示。可通过帧控制寄存器 LCD_FCR 的闪烁显示模式位域 BLINK[1:0]和闪烁显示频率位域 BLINKF[2:0]进行设置。

- 闪烁模式
  - 闪烁模式 1：BLINK[1:0]=01，COM[0]和 SEG[0]驱动的段码闪烁
  - 闪烁模式 2：BLINK[1:0]=10，COMx 和 SEG[0]驱动的段码闪烁
  - 闪烁模式 3：BLINK[1:0]=11，所有段码闪烁
- 闪烁频率
  - 典型闪烁频率为 0.25 Hz、0.5 Hz、1 Hz、2 Hz 或 4 Hz。

– 闪烁显示频率 $f_{blink}$ 由分频时钟频率 $f_{ck_div}$ 分频实现，不受帧速率影响。

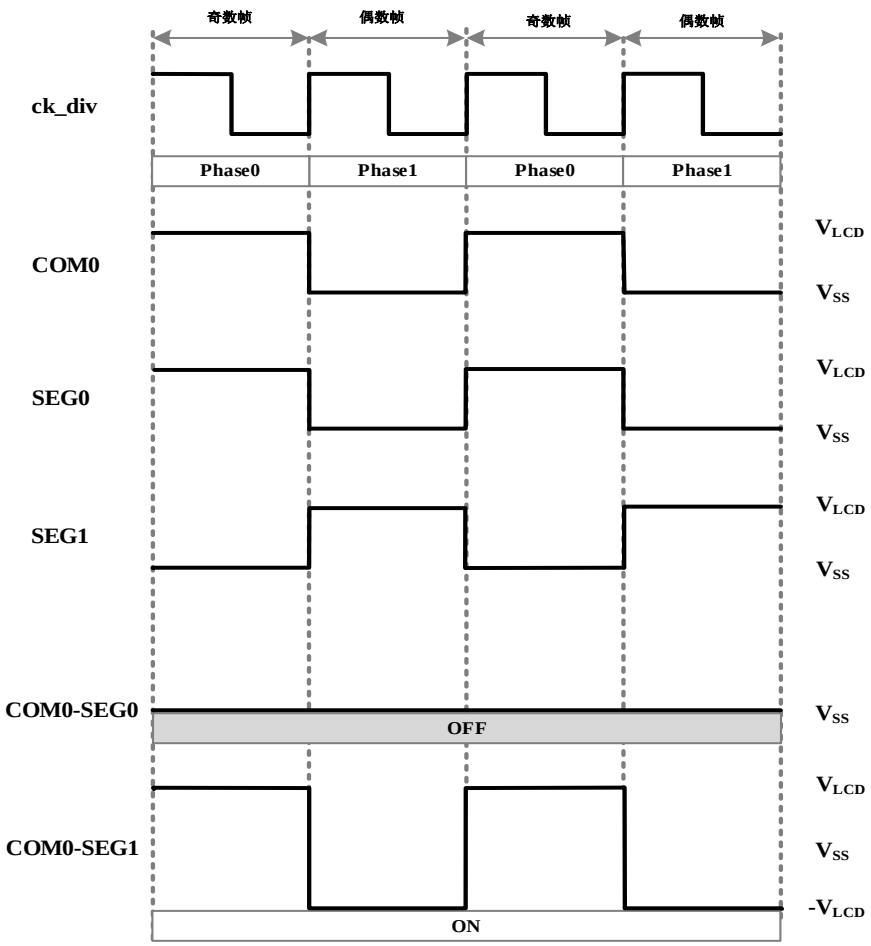
$$f_{blink} = \frac{f_{ck_div}}{2^{[BLINKF+3]}}$$

### 18.3.7 驱动波形

LCD 控制器支持 Type B 类驱动波形，本章给出部分常用驱动波形。

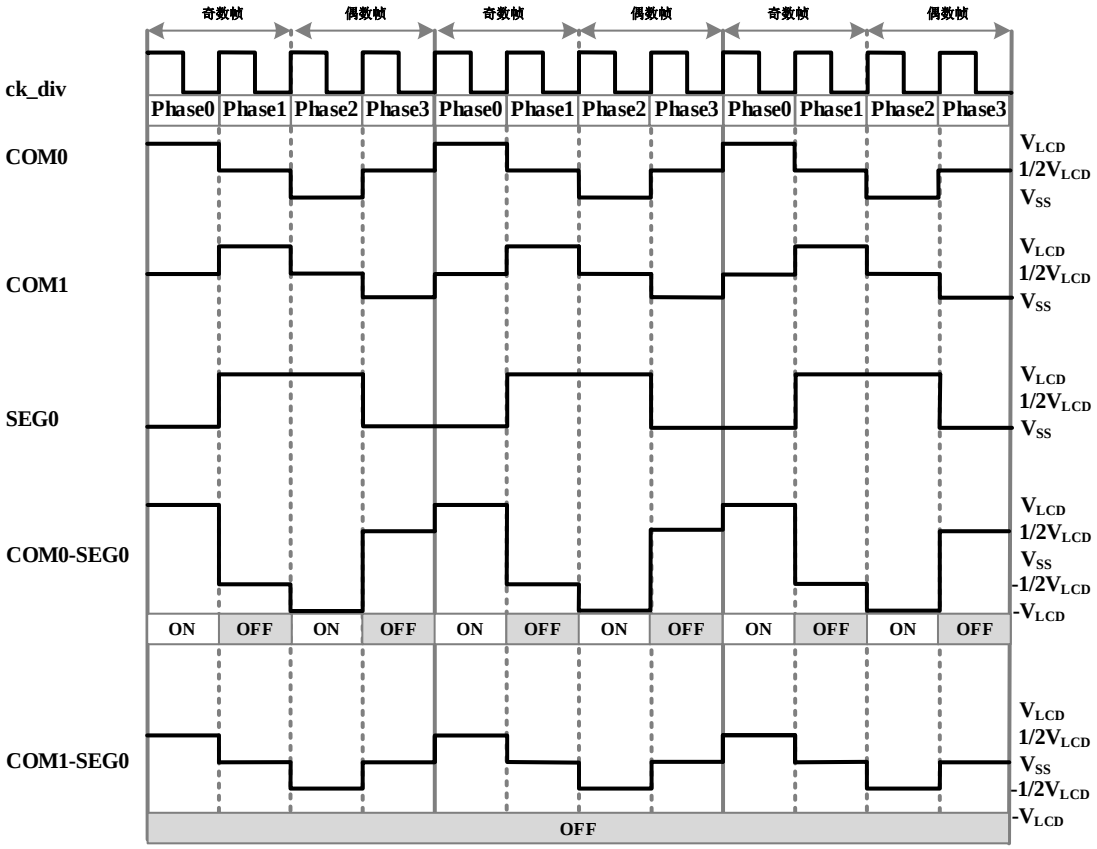
**静态占空比驱动波形**

图 18-4 静态占空比驱动波形示例



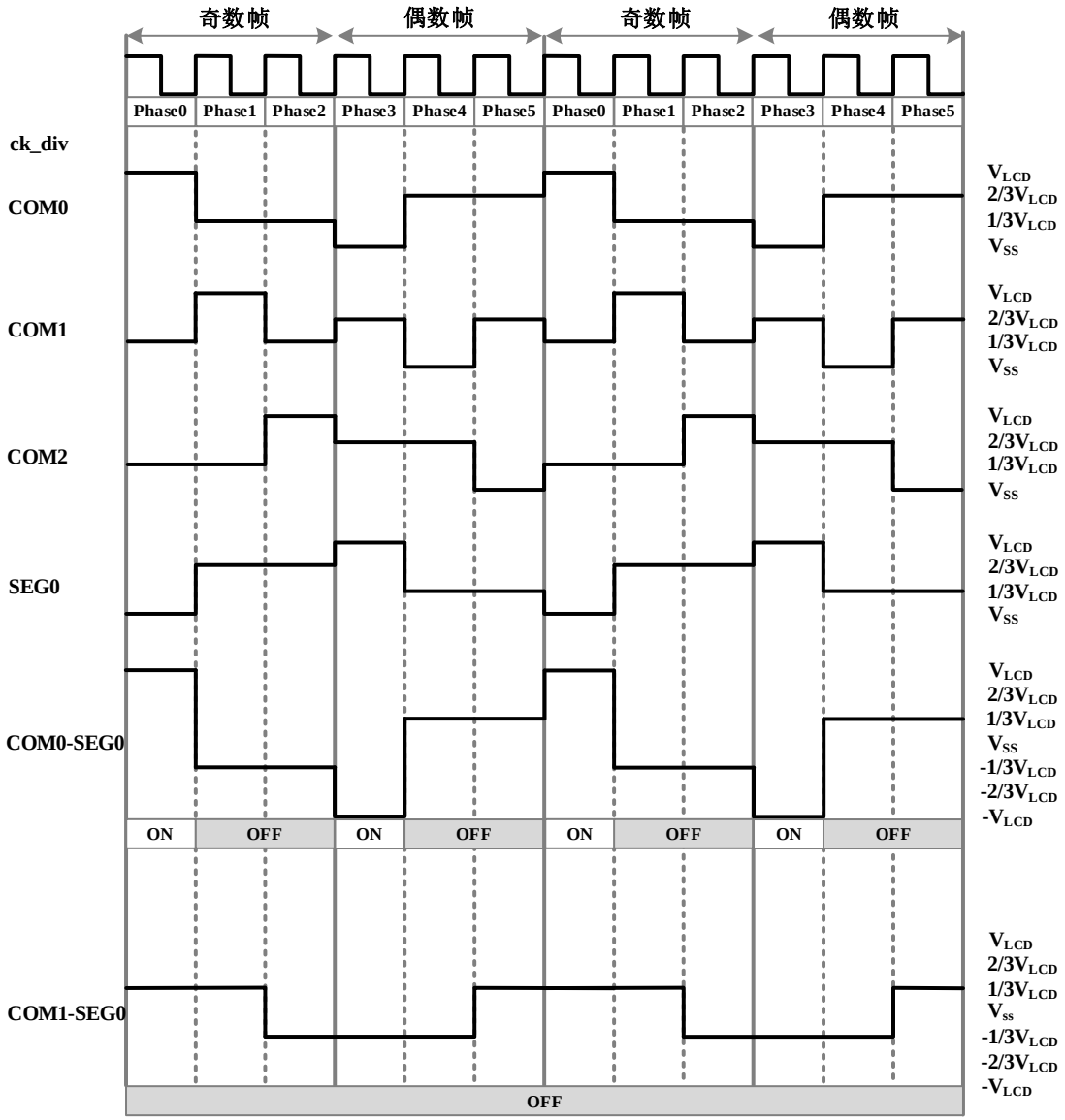
### 1/2duty，1/2bias 驱动波形

图 18-5　1/2duty，1/2bias 驱动波形示例



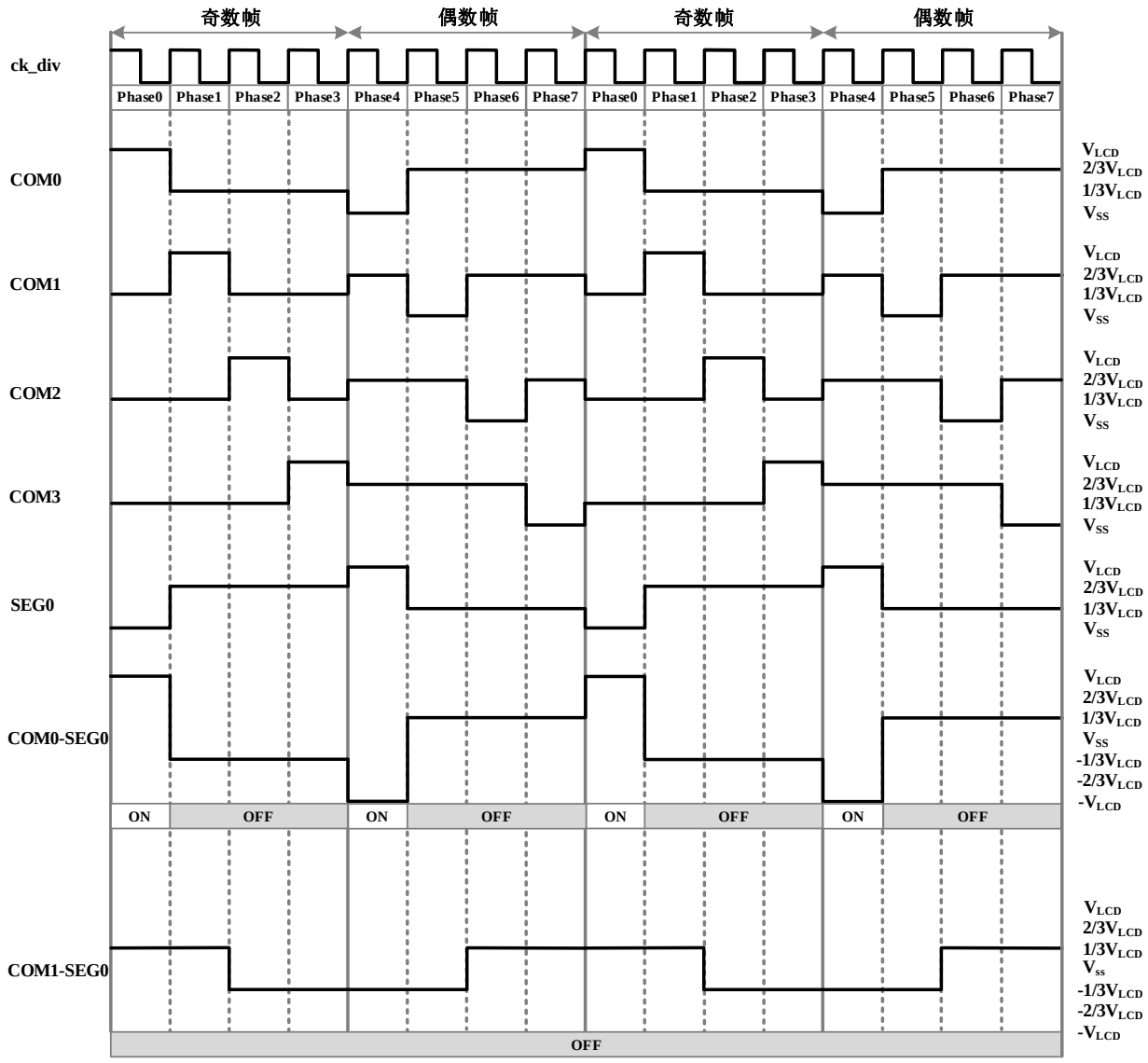
### 1/3duty，1/3bias 驱动波形

图 18-6　1/3duty，1/3bias 驱动波形示例



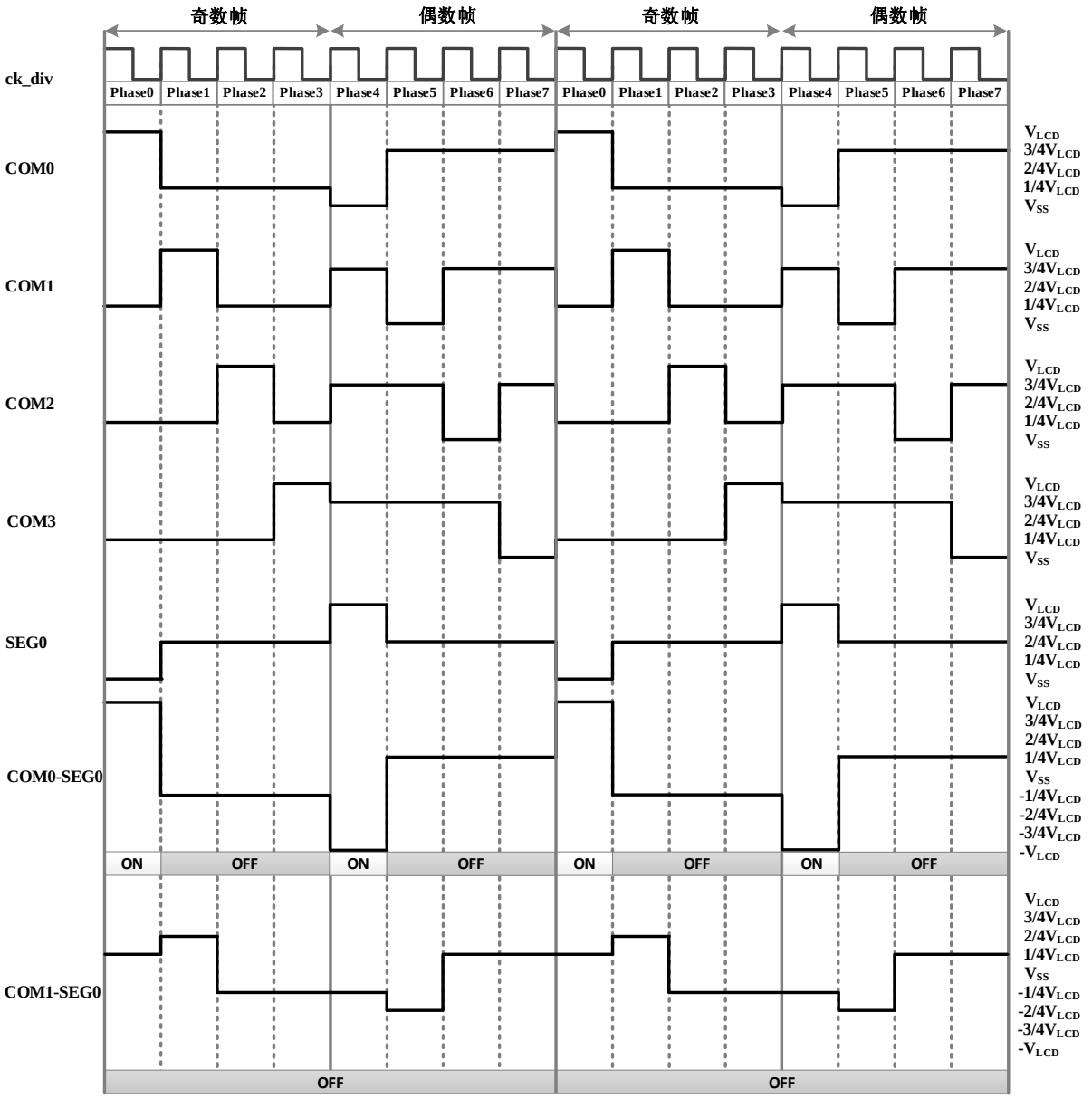
### 1/4duty，1/3bias 驱动波形

图 18-7　1/4duty，1/3bias 驱动波形示例



### 1/4duty，1/4bias 驱动波形

图 18-8　1/4duty，1/4bias 驱动波形示例



### 18.3.8 LCD 显示存储器映射

LCD 控制器驱动显示时，以 SEG 端子为显示单元，表 LCD 控制器显示映射给出了公共端子 COM 和区段端子 SEG 以及 LCD_RAM 对应关系。

表 18-3 LCD 控制器显示映射

| COM | LCD_RAM | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| COM0 | LCD_RAM0 | Res. | | | | | | | | | | | SEG20 | SEG19 | SEG18 | SEG17 | SEG16 | SEG15 | SEG14 | SEG13 | SEG12 | SEG11 | SEG10 | SEG9 | SEG8 | SEG7 | SEG6 | SEG5 | SEG4 | SEG3 | SEG2 | SEG1 | SEG0 |
| COM1 | LCD_RAM1 | Res. | | | | | | | | | | | SEG20 | SEG19 | SEG18 | SEG17 | SEG16 | SEG15 | SEG14 | SEG13 | SEG12 | SEG11 | SEG10 | SEG9 | SEG8 | SEG7 | SEG6 | SEG5 | SEG4 | SEG3 | SEG2 | SEG1 | SEG0 |
| COM2 | LCD_RAM2 | Res. | | | | | | | | | | | SEG20 | SEG19 | SEG18 | SEG17 | SEG16 | SEG15 | SEG14 | SEG13 | SEG12 | SEG11 | SEG10 | SEG9 | SEG8 | SEG7 | SEG6 | SEG5 | SEG4 | SEG3 | SEG2 | SEG1 | SEG0 |
| COM3 | LCD_RAM3 | Res. | | | | | | | | | | | SEG20 | SEG19 | SEG18 | SEG17 | SEG16 | SEG15 | SEG14 | SEG13 | SEG12 | SEG11 | SEG10 | SEG9 | SEG8 | SEG7 | SEG6 | SEG5 | SEG4 | SEG3 | SEG2 | SEG1 | SEG0 |

## 18.4 LCD 中断

帧起始标志在 LCD 控制器输出驱动波形的每个扫描周期的起始位硬件置 1，表示新扫描周期的开始。帧起始标志置 1 时，若帧起始中断已使能（SOFIE＝1），则执行帧起始中断，SOFC 位写 1 可清除帧起始标志。

表 18-4 中断请求

| 中断事件 | 事件标志 | 中断使能位 | 事件标志/中断清除方法 |
|---|---|---|---|
| 帧起始中断 | SOF | SOFIE | 向 SOFC 位写 1 |

## 18.5 低功耗显示

在低功耗模式下，LCD 控制器仍可驱动 LCD 段码屏进行显示。不需要显示时可禁用以降低功耗。

表 18-5 低功耗模式下 LCD 控制器工作状态

| 低功耗模式 | LCD 时钟源 | LCD 控制器工作状态 |
|---|---|---|
| Sleep | LXTAL/RCL | LCD 控制器已激活，则保持激活状态 |
| Stop | | |

## 18.6 LCD 寄存器

LCD 寄存器支持 32 位访问。

表 18-6 LCD 基地址

| 外设 | 基地址 |
|---|---|
| LCD | 0x4000 2400 |

### 18.6.1 LCD 控制寄存器（LCD_CR）

偏移地址：0x00

复位值：0x0000 4000

*注意：仅当 LCD 禁止时，允许对此寄存器执行写操作（LCDEN 位除外）。*

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PWRMD[1:0] | | Res. | | | | | | | BIAS[1:0] | | Res. | DUTY[1:0] | | Res. | LCDEN |
| rw | rw | | | | | | | | rw | rw | | rw | rw | | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15:14 | PWRMD[1:0] | 内置buffer功耗档位选择<br>00：超低功耗<br>01：低功耗<br>10：中功耗<br>11：高功耗 |
| 13:7 | 保留 | 写入无效 |
| 6:5 | BIAS[1:0] | 偏置选择<br>00：1/2偏置<br>01：1/3偏置<br>10：1/4偏置<br>11：保留（默认1/2偏置） |

| | | |
|---|---|---|
| 4 | 保留 | 写入无效 |
| 3:2 | DUTY[1:0] | 占空比选择<br>00：静态<br>01：1/2占空比<br>10：1/3占空比<br>11：1/4占空比 |
| 1 | 保留 | 写入无效 |
| 0 | LCDEN | LCD控制器使能<br>0：禁止<br>1：使能 |

### 18.6.2 LCD 帧控制寄存器（LCD_FCR）

偏移地址：0x04

复位值：0x0000 0000

*注意：　仅当LCD 禁止时，允许对此寄存器执行写操作（SCOC 和 ANTI_POLAR 位除外）。*

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | CC[3:0] | | | | ANTI_POLAR | SCOC | PS[2:0] | | | DIV[3:0] | | | | BLINK[1:0] | |
| | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BLINKF[2:0] | | | Res. | | | | DEAD[2:0] | | | Res. | | | | SOFIE | Res. |
| rw | rw | rw | | | | | rw | rw | rw | | | | | rw | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31 | 保留 | 写入无效 |
| 30:27 | CC[3:0] | $V_{LCD}$输出电压控制<br>参见*表：$V_{LCD}$输出电压配置。* |
| 26 | ANTI_POLAR | 防极化使能<br>仅当LCD禁止时该位生效。<br>0：COM和SEG浮空<br>1：COM和SEG连接$V_{SS}$ |

| | | |
|---|---|---|
| 25 | SCOC | COM和SEG输出控制<br>仅当LCD使能时，该控制位生效。<br>0：COM和SEG连接$V_{SS}$<br>1：COM和SEG根据显示配置输出 |
| 24:22 | PS[2:0] | 时钟预分频器<br>ck_ps = LCD_KCLK/$2^{PS[2:0]}$，参见*频率发生器*。<br>000：ck_ps=LCD_KCLK<br>001：ck_ps=LCD_KCLK/2<br>010：ck_ps=LCD_KCLK/4<br>⋮<br>111：ck_ps=LCD_KCLK/128 |
| 21:18 | DIV[3:0] | 时钟分频器<br>0000：ck_div=ck_ps/16<br>0001：ck_div=ck_ps/17<br>0010：ck_div=ck_ps/18<br>⋮<br>1111：ck_div=ck_ps/31 |
| 17:16 | BLINK[1:0] | 闪烁显示模式选择<br>00：禁止闪烁<br>01：在SEG[0]、COM[0]上启用闪烁（1个像素）<br>10：在SEG[0]、所有COM上启用闪烁（最多4个像素，取决于占空比）<br>11：在所有SEG和所有COM上启用闪烁（所有像素） |
| 15:13 | BLINKF[2:0] | 闪烁频率选择<br>000：$f_{ck_div}$/8<br>001：$f_{ck_div}$/16<br>010：$f_{ck_div}$/32<br>011：$f_{ck_div}$/64<br>100：$f_{ck_div}$/128<br>101：$f_{ck_div}$/256<br>110：$f_{ck_div}$/512<br>111：$f_{ck_div}$/1024 |
| 12:9 | 保留 | 写入无效 |

| | | |
|---|---|---|
| 8:6 | DEAD[2:0] | 帧间死区时间<br>000：无死区<br>001：1个ck_div时钟周期<br>010：2个ck_div时钟周期<br>⋮<br>111：7个ck_div时钟周期 |
| 5:2 | 保留 | 写入无效 |
| 1 | SOFIE | 帧起始中断使能<br>0：禁止<br>1：使能 |
| 0 | 保留 | 写入无效 |

### 18.6.3 LCD 状态寄存器（LCD_SR）

偏移地址：0x08

复位值： 0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:2 | 保留 | 写入无效 |
| 1 | SOF | 帧起始标志<br>新帧开始时SOF位置1，通过向LCD_CLR寄存器中的SOFC写1，该位清零。<br>0：无事件<br>1：发生帧起始事件。若SOFIE位置1，则产生LCD帧起始中断 |
| 0 | 保留 | 写入无效 |

### 18.6.4 LCD 状态清除寄存器（LCD_CLR）

偏移地址：0x0C

复位值：0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:2 | 保留 | 写入无效 |
| 1 | SOFC | 帧起始标志清零位<br>向此位写1时，LCD_SR寄存器中SOF标志清零。 |
| 0 | 保留 | 写入无效 |

### 18.6.5 LCD 显示存储器 x（LCD_RAMx）（x = 0 ~ 3）

偏移地址：0x14~0x20

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | SEGMENT_DATA[20:16] | | | | |
| | | | | | | | | | | | rw | rw | rw | rw | rw |

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:21 | 保留 | 写入无效 |
| 20:0 | SEGMENT_DATA[20:0] | LCD显示寄存器<br>每个bit位对应一个LCD屏的像素点<br>0：不显示<br>1：显示 |

*注意：LCD_RAM与COM/SEG端子对应关系参见：LCD显示存储器映射。*

# 19 高级控制定时器（TIM1）

## 19.1 简介

高级控制定时器 TIM1 由一个 16 位自动重载计数器构成。可用于生成输出波形（比较输出、单脉冲输出、PWM 和带左右死区插入的互补 PWM）等多种用途。

高级控制定时器 TIM1 和通用定时器结合使用，可实现定时器互联功能。

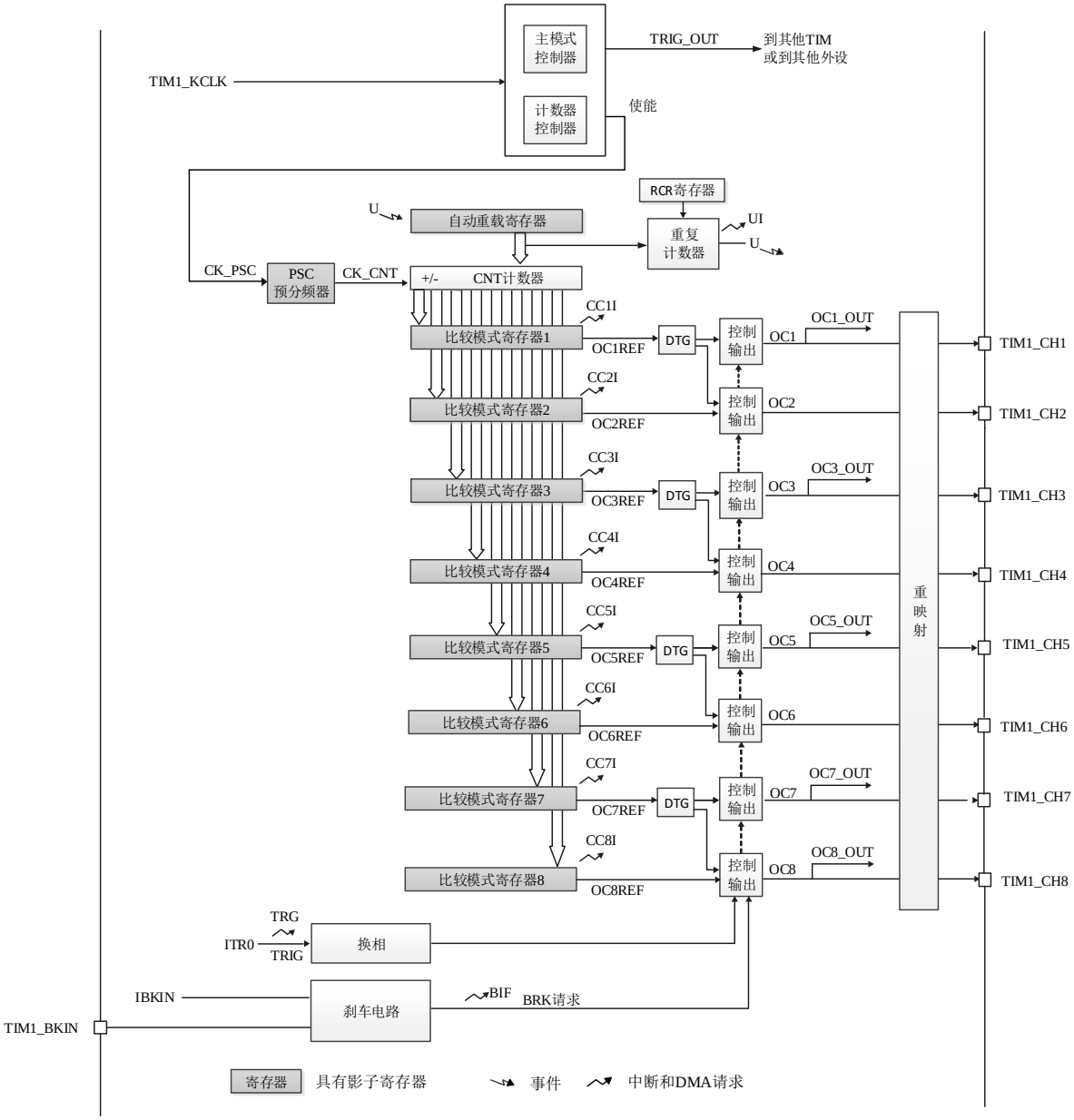
## 19.2 TIM1 主要特性

- 16 位递增、递减、递增/递减自动重载计数器
- 可编程预分频器
- 8 位重复计数器
- 8 路独立输出或 4 路可编程左右死区的互补输出，可用于：
  - 输出比较
  - PWM 输出，互补输出支持硬件移相
  - 单脉冲模式输出
- 1 个刹车输入，可选择停止模式、暂停模式和恢复模式：
  - 软件触发
  - 具备复用功能的 GPIO
  - COMP1 输出、COMP2 输出
  - 系统刹车（CPU LOCKUP 事件输出、PVD 报警事件输出、ADC 模拟看门狗监控电压事件）
- PWM 边沿或周期点可触发启动 ADC 转换
- 支持与多个定时器互联，以实现定时器换相、同步或级联
- 与比较器互联实现开窗控制
- 与 OPAMP2 互联实现输入开关切换
- 支持 DMA

## 19.3 TIM1 功能描述

### 19.3.1 TIM1 框图

图 19-1 TIM1 框图




### 19.3.2 TIM1 引脚

表 19-1 TIM1 引脚

| 引脚名称 | 信号类型 | 说明 |
|---|---|---|
| TIM1_CHx | 输出 | 通道输出引脚 |
| TIM1_BKIN | 输入 | 刹车输入信号 |

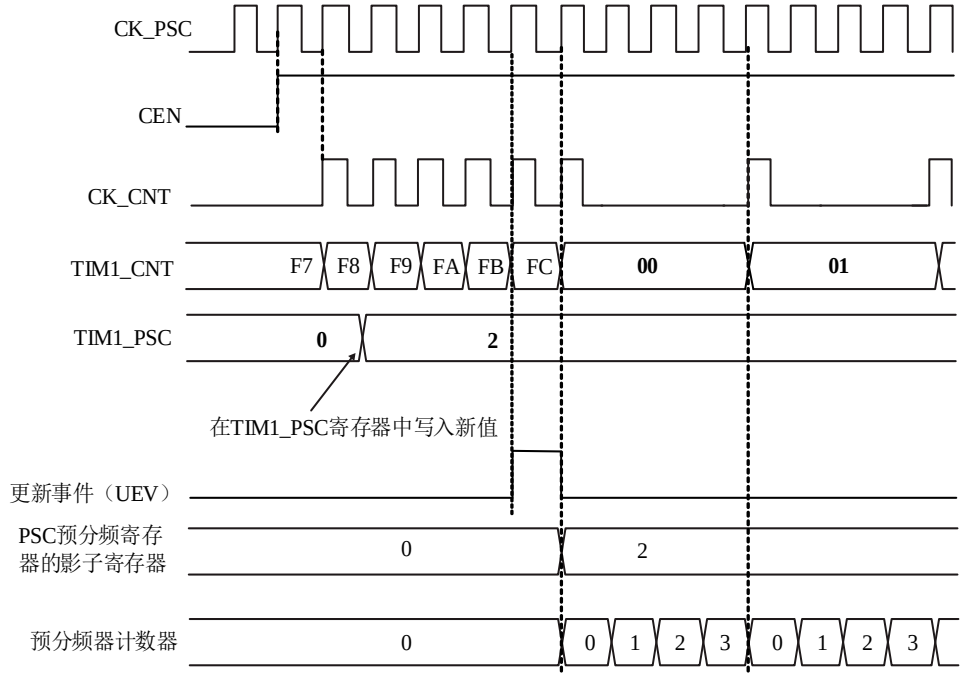
### 19.3.3 PSC 预分频器

PSC 预分频器对 CNT 计数器的输入时钟进行分频，分频系数介于 1 和 32768 之间。TIM1_PSC 寄存器中的 PSC[3:0]位域用于配置分频值，实际分频系数为 $2^{PSC[3:0]}$。

TIM1_PSC 寄存器具有缓冲功能（影子寄存器），每次写入的预分频值，会在下一次更新事件发生时生效。

下图举例显示了预分频器更新生效的时序。

图 19-2 预分频器分频系数由 1 变为 4 时的计数器时序图




### 19.3.4 计数器

TIM1 内置一个 16 位的计数器，可选择递增计数模式、递减计数模式或中心对齐计数模式。

计数器配置支持动态修改，在计数器运行时，TIM1_CNT 寄存器、TIM1_ARR 寄存器和 TIM1_PSC 寄存器均可进行读写操作。

自动重载值寄存器（TIM1_ARR）中的 ARR[15:0]位域可用于配置计数器的重载值，该寄存器具有缓冲功能（影子寄存器）。

如果 TIM1_CR1 寄存器中的 ARPE 位为 0 时，影子寄存器的更新不受更新事件的影响，TIM1_ARR 寄存器的值会立即更新。如果 ARPE 位为 1 时，当发生更新（UEV）事件，则 TIM1_ARR 寄存器中的值将更新到影子寄存器中。

如果将 TIM1_CR1 寄存器中的 UDIS 位置 1，可禁止更新事件，从而避免对影子寄存器的更新，但不影响计数器及预分频计数器均被初始化。当 UDIS 位为 0 时，允许更新事件生成，允许加载使能预装载功能的影子寄存器。

更新事件标志的请求源：

- 计数器上溢/下溢（RCR 设置为 0）；
- 重复计数器递减到 0，下一个上溢或下溢（RCR 设置为非 0）。

将 UG 位置 1，会加载 PSC 及使能预装载功能的影子寄存器（UDIS 位为 0 时），计数器及预分频计数器均被初始化，但不会将 TIM1_SR 寄存器中的 UIF_UDR 或 UIF_OVR 置位。

**递增计数模式**

在递增计数模式下，计数器从 0 开始递增计数，每个 CK_CNT 周期自动加 1，当计数值等于 TIM1_ARR 寄存器的值时，将产生计数器上溢事件，TIM1_SR 寄存器的 UIF_OVR 标志将自动置 1，同时计数器将被清 0 并重新开始计数。

如果使用重复计数器，在定时器发生上溢事件时，仅递减重复计数器的值，只有当递增计数的重复次数达到其设定次数加 1 次（TIM1_RCR+1）后，才会生成更新事件。

发生更新事件时，TIM1_SR 寄存器中的 UIF_OVR 位将自动置 1，并对以下寄存器进行更新：

- TIM1_RCR 寄存器的值更新到重复计数器中；
- 预装载值（TIM1_ARR 寄存器）更新到自动重载影子寄存器中；
- 预装载值（TIM1_PSC 寄存器）重新加载到 PSC 预分频器的影子寄存器中。

以下各图，显示了递增计数模式下，计数器发生上溢更新事件时的时序，TIM1_ARR 寄存器配置为 0x36：

图 19-3 递增计数模式下配置预分频器为 1 分频上溢事件时序图

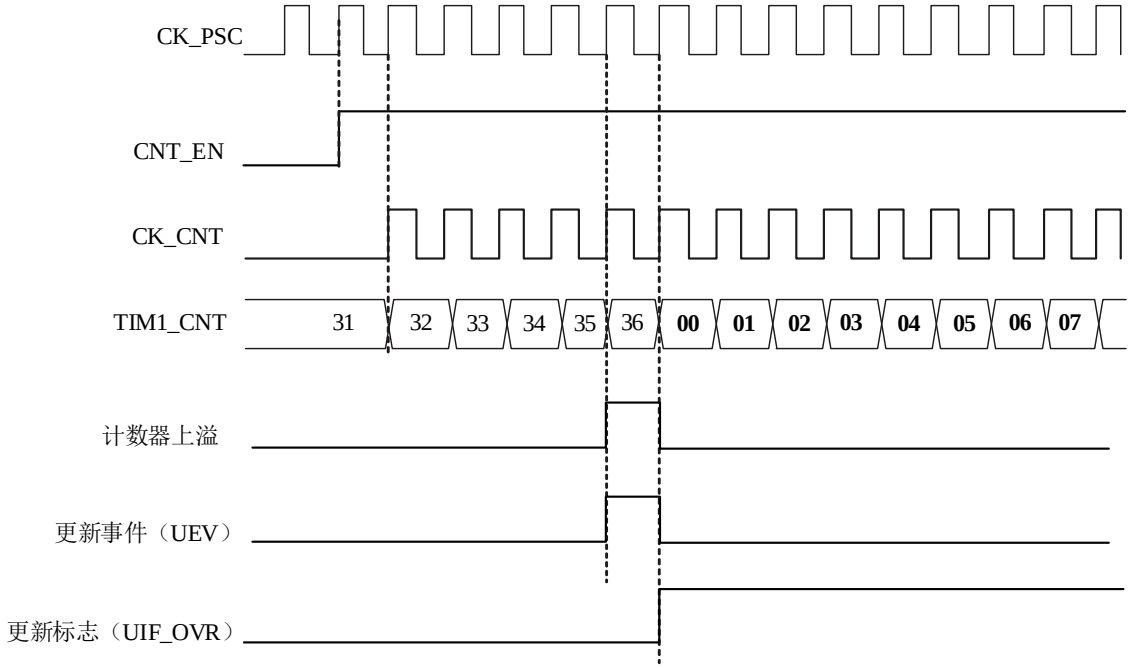



图 19-4 递增计数模式下配置预分频器为 4 分频上溢事件时序图

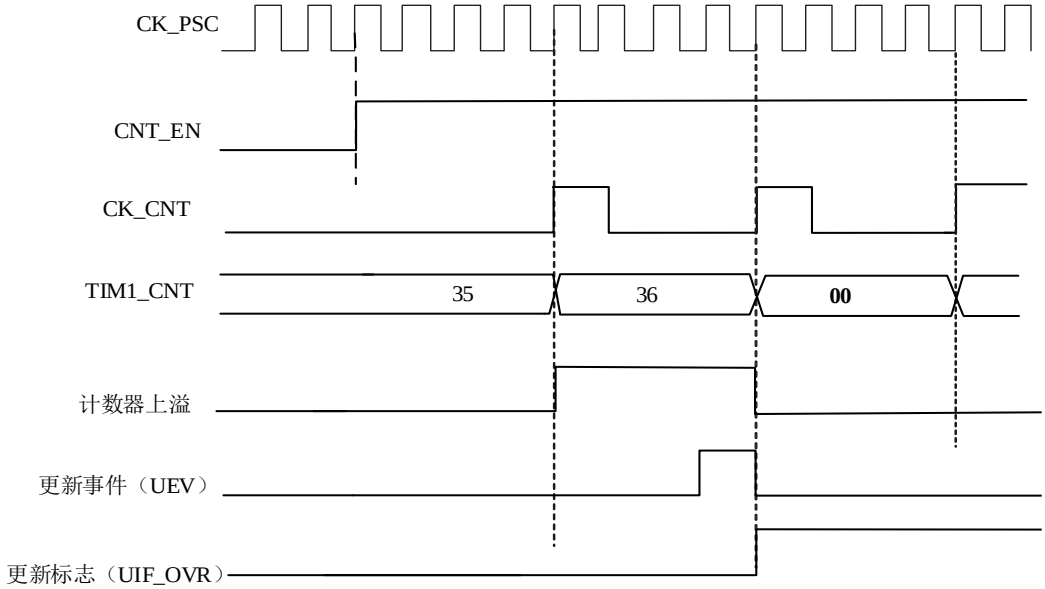


图 19-5 递增计数模式下计数器时序图，ARPE=0 时更新事件

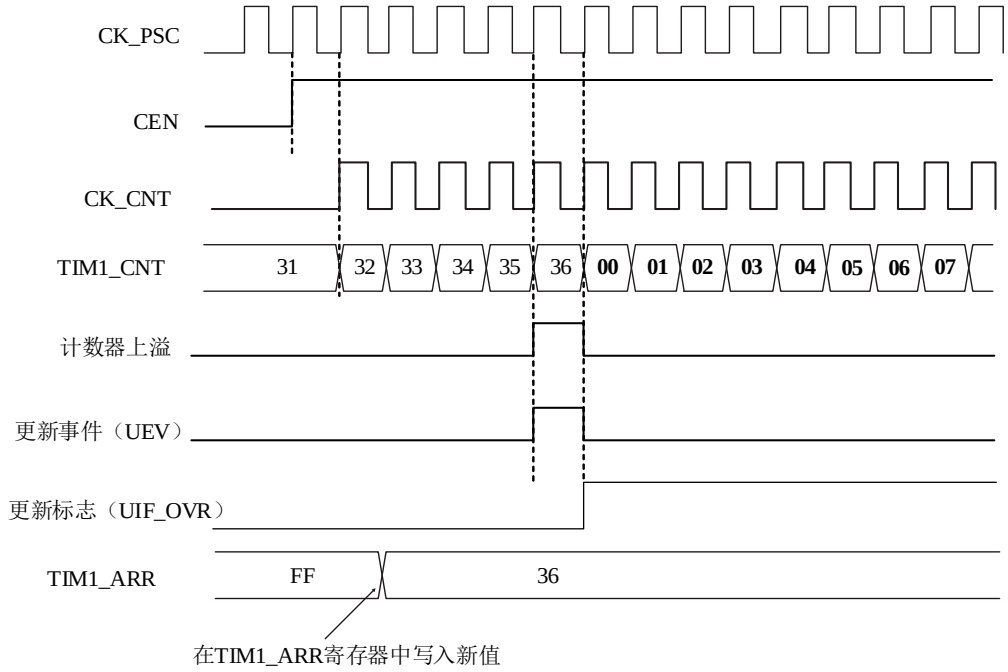
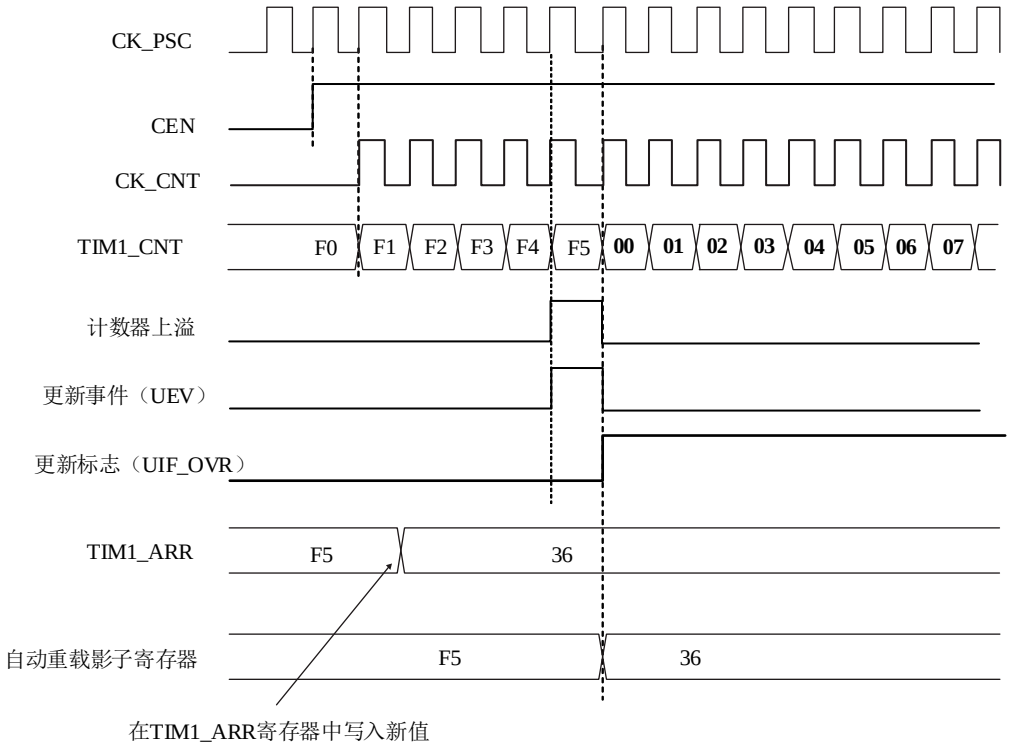



图 19-6 递增计数模式下计数器时序图，ARPE=1 时更新事件




## 递减计数模式

在递减计数模式下，计数器从 TIM1_ARR 寄存器的值开始递减计数，每个 CK_CNT 周期自动减 1，当计数器等于 0 时，将产生计数器下溢事件，TIM1_SR 寄存器的 UIF_UDR 标志将自动置 1，同时计数器加载自动重载值，并重新开始

递减计数。

如果使用重复计数器，在定时器发生下溢事件时，仅递减重复计数器的值，只有递减计数的重复次数达到其次数加 1 次（TIM1_RCR+1）后，才会产生更新事件。

发生更新事件时，TIM1_SR 寄存器中的 UIF_UDR 位将自动置 1，并对以下寄存器进行更新：

- TIM1_RCR 寄存器的值更新到重复计数器中；
- 预装载值（TIM1_PSC 寄存器）重新加载到 PSC 预分频器的影子寄存器中；
- 预装载值（TIM1_ARR 寄存器）更新到自动重载影子寄存器中。TIM1_ARR 寄存器的值在计数器重载之前被更新，因此下一个周期才是预期的值。

以下各图，显示了计数器发生下溢更新事件时的时序，TIM1_ARR 寄存器配置为 0x36：

图 19-7　递减计数模式下配置预分频器为 1 分频下溢事件时序图

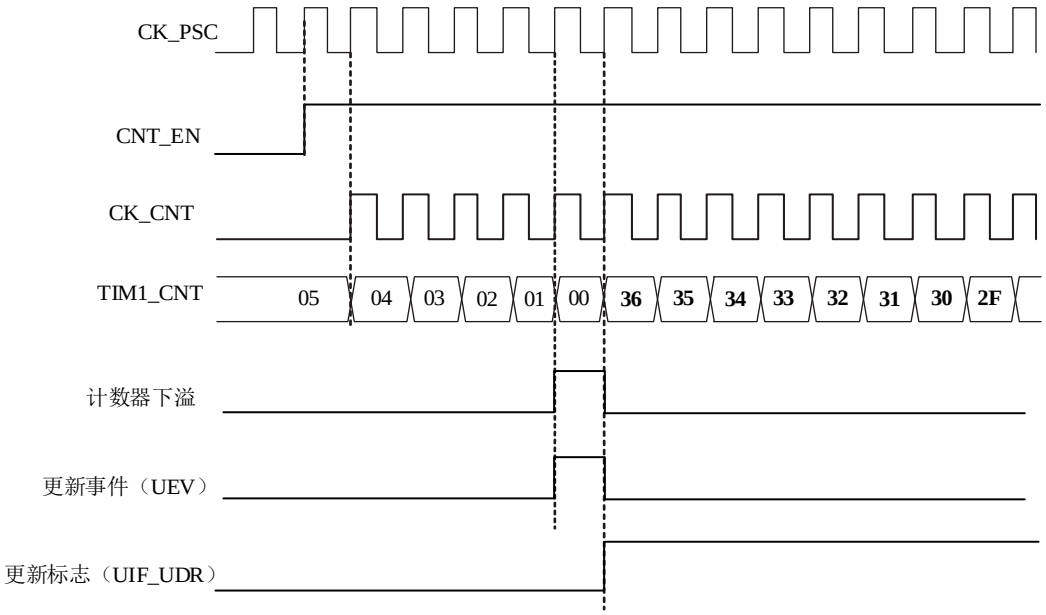


图 19-8 递减计数模式下配置预分频器为 4 分频下溢事件时序图

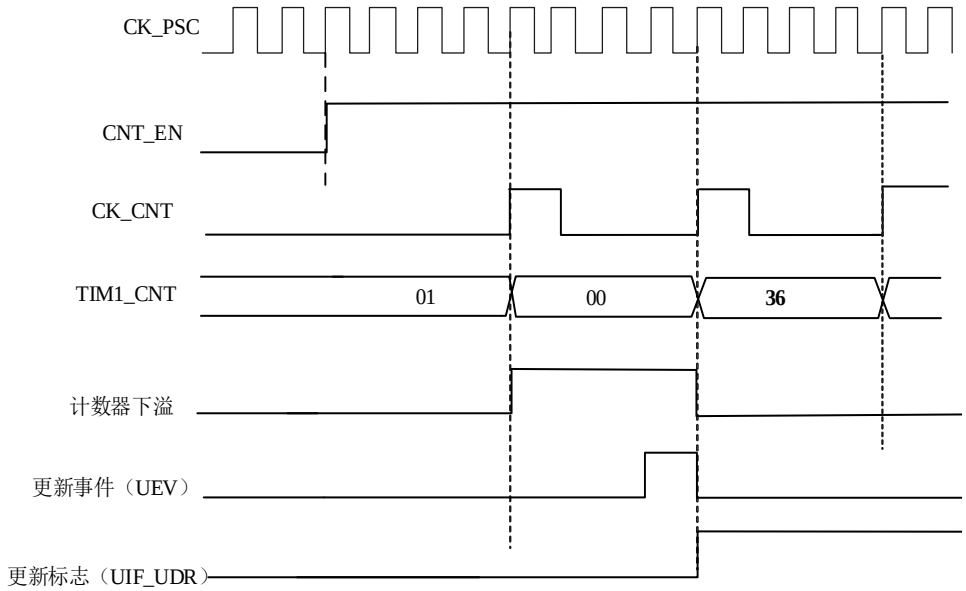
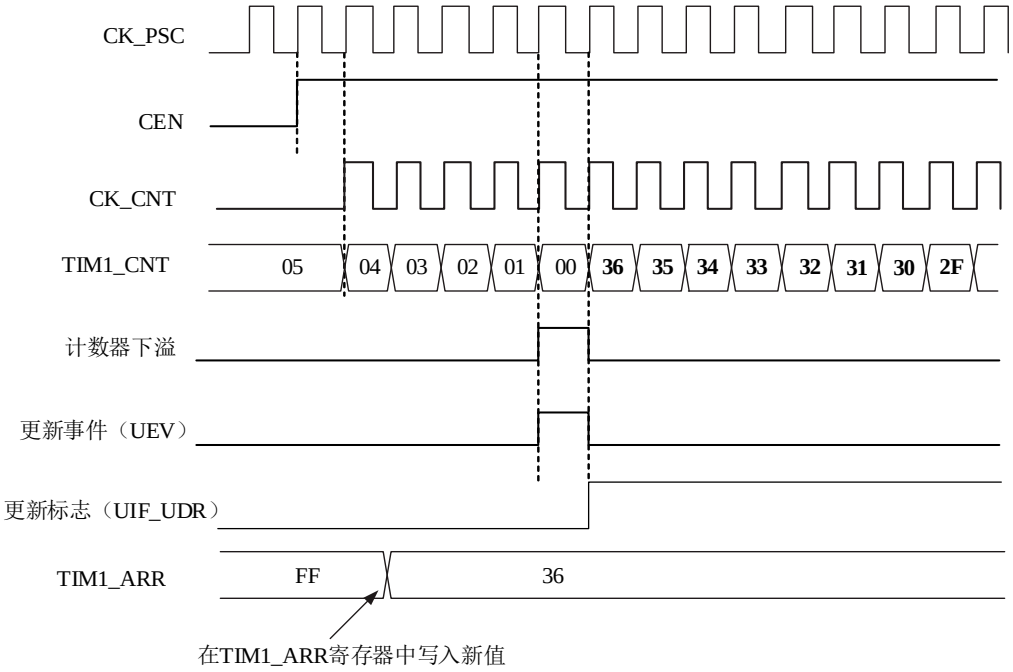



图 19-9 递减计数模式下计数器时序图，ARPE=0 时更新事件




### 中心对齐模式（递增/递减计数）

在中心对齐模式下，计数过程由以下阶段组成：

1） 计数器从 TIM1_CNT 的值开始递增计数到 TIM1_ARR–1，生成计数器上溢事件；

2） 从 TIM1_ARR 的值开始递减计数到 1，生成计数器下溢事件；

3） 从 0 开始递增计数。

当 TIM1_CR1 寄存器中的 DIR 位为 1 时，启动计数后，计数器从当前 TIM1_CNT

的值开始递减计数。

启动计数后，无论当前是递增还是递减计数，当 UG 位置 1 时，TIM1_CNT 寄存器的值和 TIM1_CR1 寄存器中的 DIR 均变为 0。

当 TIM1_CR1 寄存器中的 CMS[1:0]位域不为 00 时，计数器将采用中心对齐模式，在此模式下，TIM1_CR1 寄存器中的 DIR 位不能写入，读取 DIR 位可获取当前计数器的计数方向。中心对齐模式共有以下几种计数模式：

- 中心对齐模式 1（CMS[1:0]位域为 01）：计数器交替进行递增计数和递减计数。上溢或下溢均触发更新事件；仅在递减计数过程中，当发生比较匹配时，TIM1_SR 寄存器中的 CCxIF 位会置 1；
- 中心对齐模式 2（CMS[1:0]位域为 10）：计数器交替进行递增计数和递减计数。上溢或下溢均触发更新事件；仅在递增计数过程中，当发生比较匹配时，TIM1_SR 寄存器中的 CCxIF 位会置 1；
- 中心对齐模式 3（CMS[1:0]位域为 11）：计数器交替进行递增计数和递减计数。上溢或下溢均触发更新事件；在递增计数或递减计数过程中，当发生比较匹配时，TIM1_SR 寄存器中的 CCxIF 位会置 1。

发生更新事件时，TIM1_SR 寄存器中的 UIF_OVR/UIF_UDR 位将自动置 1，并对以下寄存器进行更新：

- TIM1_RCR 寄存器的值更新到重复计数器中；
- 预装载值（TIM1_PSC 寄存器）重新加载到 PSC 预分频器的影子寄存器中；
- 预装载值（TIM1_ARR 寄存器）更新到自动重载影子寄存器中。

以下各图，显示了中心对齐模式下，计数器发生上/下溢更新事件时的时序，TIM1_ARR 寄存器配置为 0x06：

图 19-10 中心对齐模式下配置预分频器为 1 分频上/下溢事件时序图

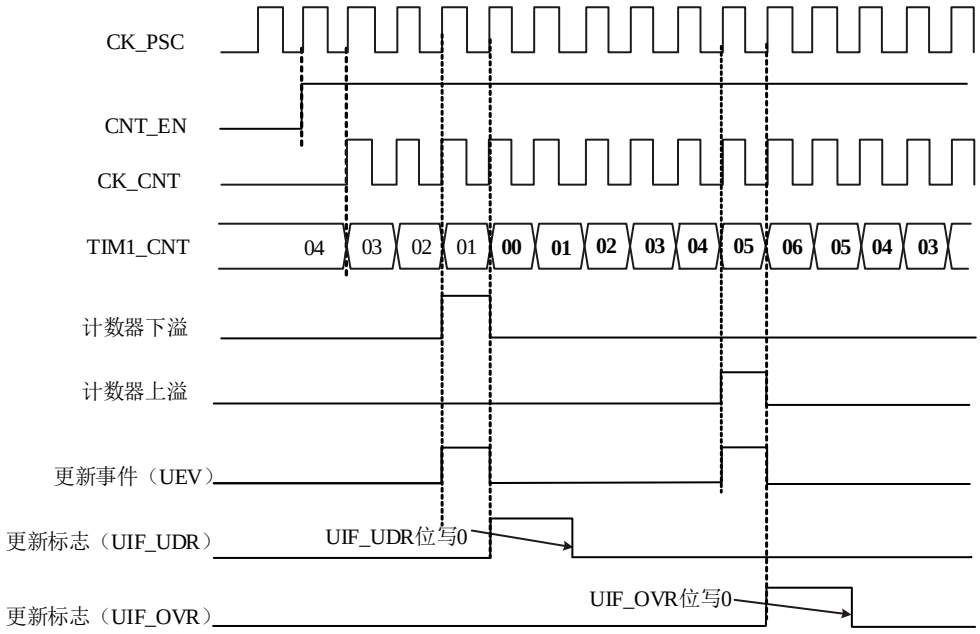



图 19-11 中心对齐模式下配置预分频器为 2 分频下溢事件时序图

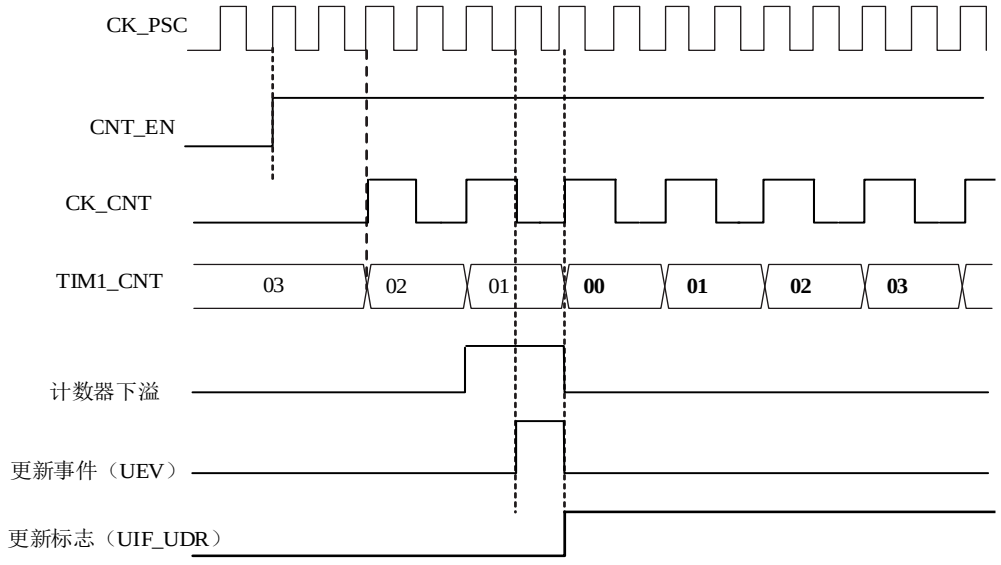


图 19-12 中心对齐模式下，产生下溢事件，ARPE=1 时的时序图

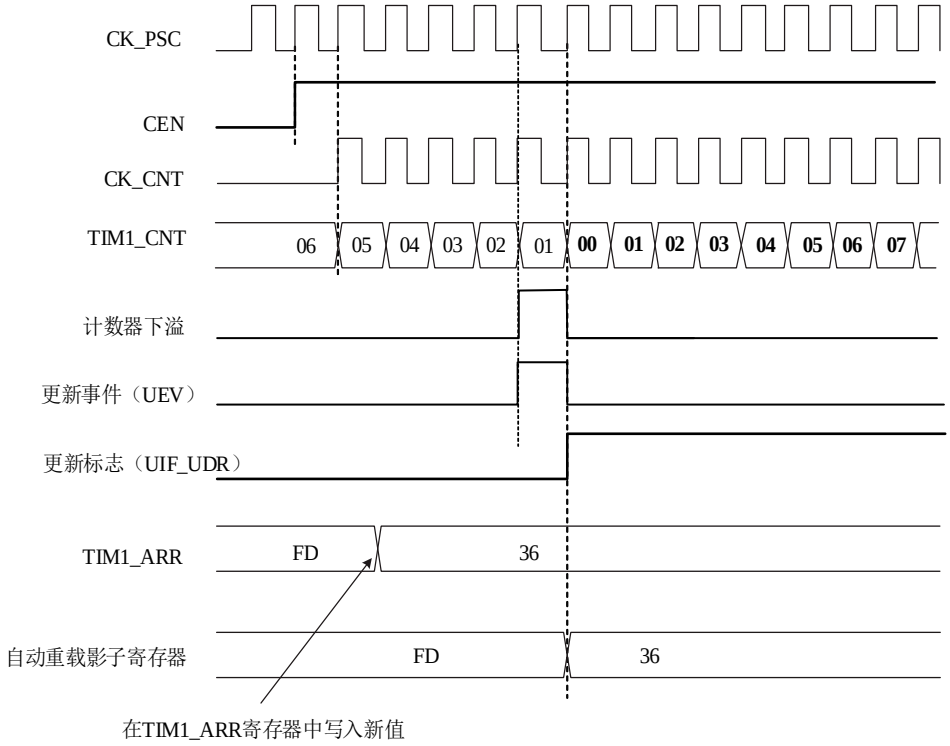
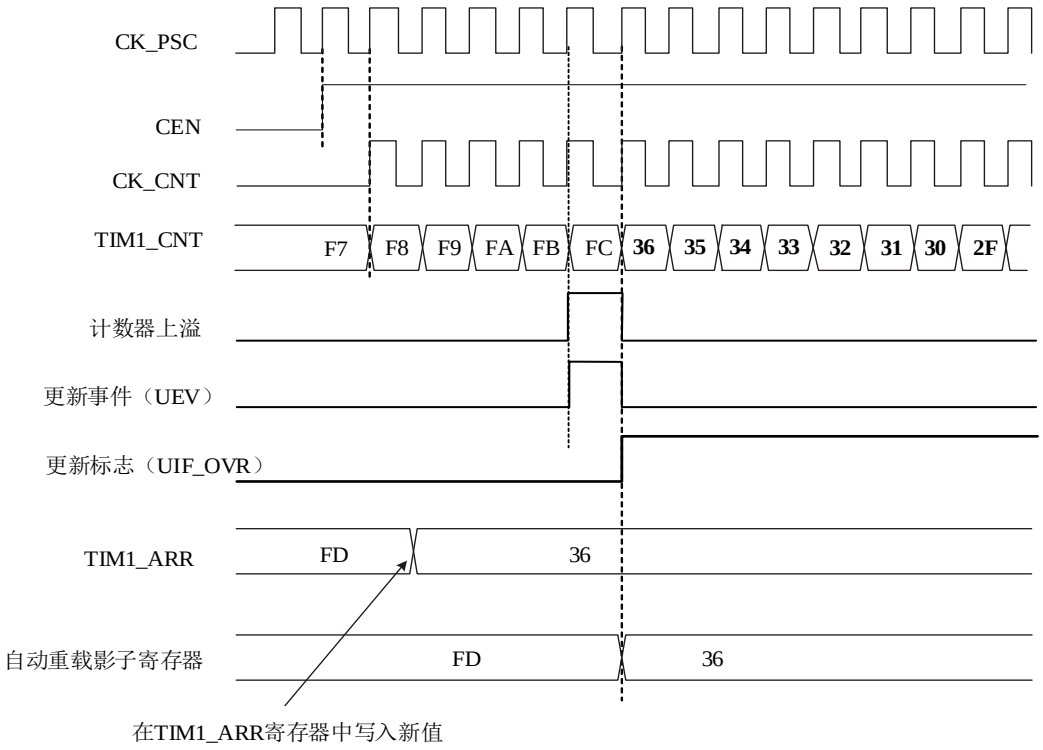



图 19-13 中心对齐模式下，产生上溢事件，ARPE=1 时的时序图




建议使用中心对齐模式时，在启动计数器前置位 TIM1_EVTG 寄存器中的 UG 位，产生一个更新事件效果，并且不要在计数器计数期间修改计数器的值。

### 19.3.5 重复计数器

重复计数器由8位递减计数器构成，计数值可通过TIM1_RCR寄存器的REP[7:0]位域进行配置。

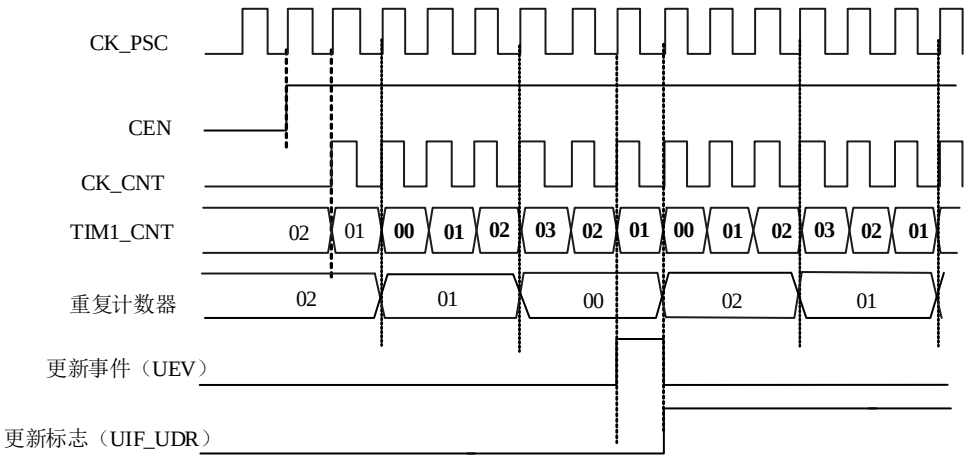
- REP 等于 0：更新事件在计数器上/下溢时产生；
- REP 不等于 0：当重复计数器递减到 0，下一次上溢或下溢产生时，触发更新事件。

重复计数器在下列情况下自动递减：

- 递增计数模式下的每次计数器上溢；
- 递减计数模式下的每次计数器下溢；
- 中心对齐模式下每次计数器上溢和计数器下溢。

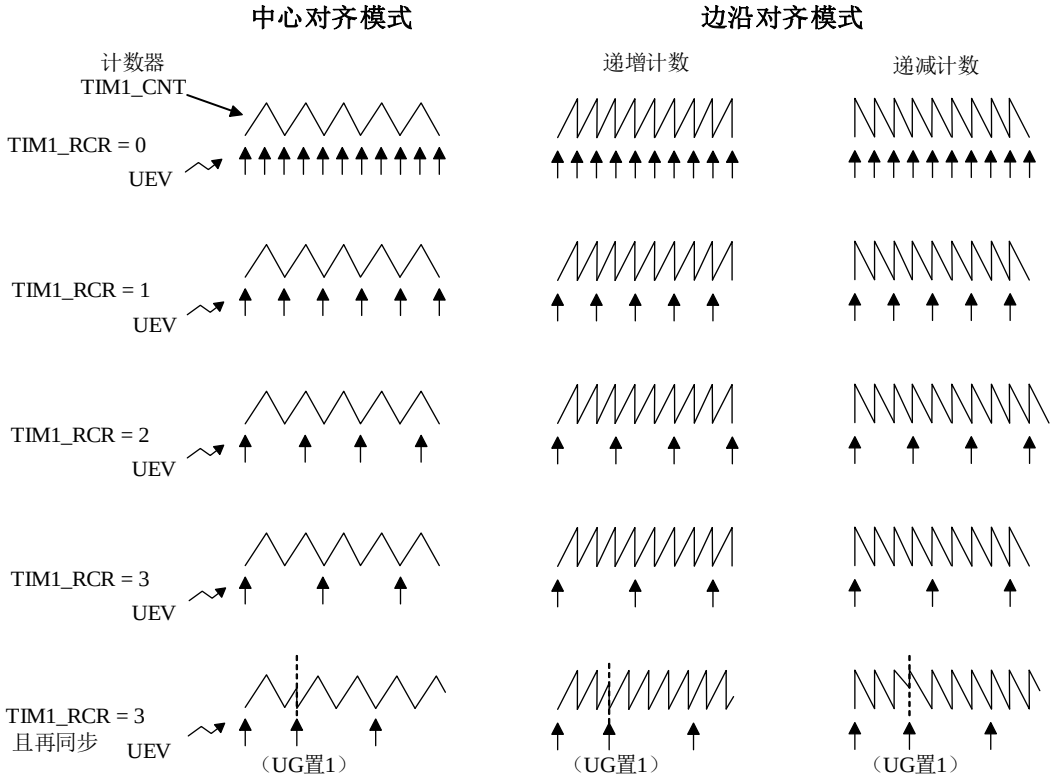
当 TIM1_EVTG 寄存器中的 UG 置 1、计数上溢或下溢时，重复计数器自动加载 TIM1_RCR 寄存器的值。

图 19-14 中心对齐模式下重复计数加载时序图




下图显示不同模式下，设置不同 TIM1_RCR 寄存器值时更新事件的频率示例。

图 19-15 不同模式下设置不同 TIM1_RCR 寄存器值时的更新频率示例




### 19.3.6 计数时钟选择

计数器的计数时钟由 APB 总线时钟提供，作为 TIM1 时钟（TIM1_KCLK）。

可通过配置 TIM1_PSC 寄存器的值进行预分频后，作为计数时钟（CK_CNT）。

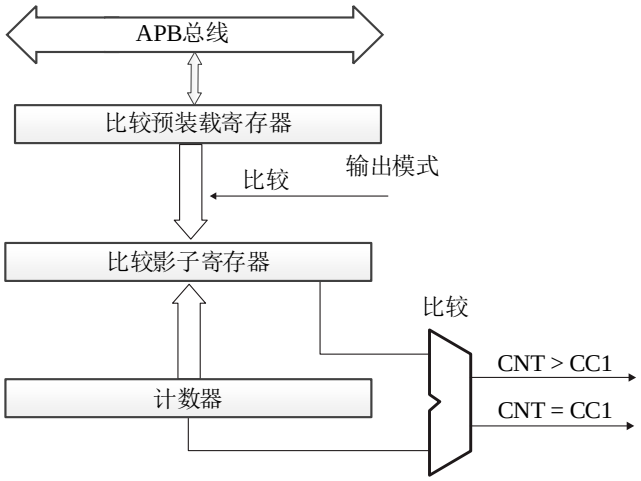
### 19.3.7 比较通道

比较通道可通过 TIM1_CFG 寄存器的 MODE_SETx 位配置成两种工作模式：独立模式和互补模式（OC1 和 OC2 配对，OC3 和 OC4 配对，OC5 和 OC6 配对，OC7 和 OC8 配对）。互补模式请参见：*互补模式、PWM 移相及死区插入*。

比较通道包含：

- 比较寄存器（包括一个影子寄存器）
- 输出比较通道（比较器和输出控制）

图 19-16 比较通道



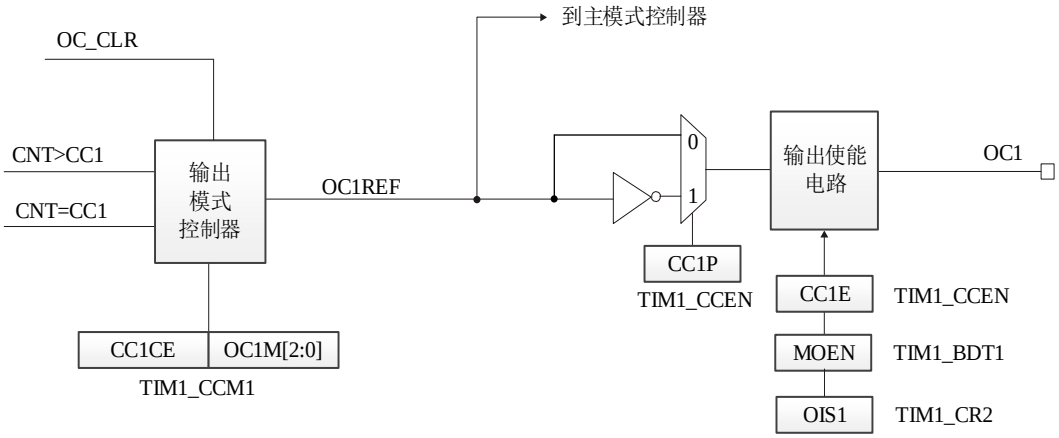

### 输出比较模式

输出通道未使能（MOEN 或 CCxE 为 0）时，固定输出空闲电平，空闲电平可通过 TIM1_CR2 寄存器的 OISx 位控制。

输出比较模式分为：

- 比较输出模式，详见*比较输出模式*；
- 强制输出模式，详见*强制输出模式*；
- PWM 模式，详见 *PWM 模式*；
- 单脉冲模式，详见*单脉冲模式*。

图 19-17 比较通道的输出连接图（通道 1）




## 19.3.8 比较输出模式

该模式包含：匹配输出有效电平、匹配输出无效电平和翻转模式。通过

TIM1_CCMx 寄存器中的 OCxPE 位，可使能或禁止 TIM1_CCx 寄存器的预装载功能。

当 TIM1_CCx 的影子寄存器与 TIM1_CNT 寄存器的值匹配时：

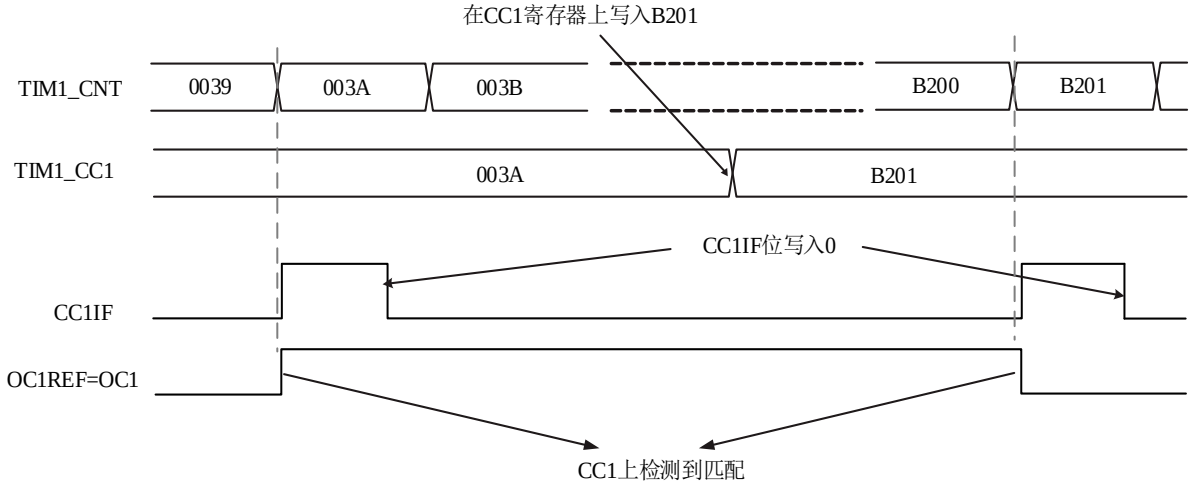
- TIM1_SR 寄存器中的 CCxIF 位置 1；
- 输出极性由 TIM1_CCEN 寄存器中的 CCxP 位决定；
- 如果此时 TIM1_DIER 寄存器中的 CCxIE 位为 1，则触发中断；
- 如果此时 TIM1_DIER 寄存器中的 CCx_DMAEN 位为 1，则产生 DMA 请求。

配置为比较输出模式时，其步骤如下：

1) 配置 TIM1_ARR 寄存器和 TIM1_CCx 寄存器；
2) 选择输出模式。例如：
   - 配置 TIM1_CCMx 寄存器中的 OCxM[2:0]位域为 011（当 TIM1_CNT 寄存器的值与 TIM1_CCx 寄存器的值匹配时，OCx 输出发生翻转）；
   - 禁止预装载寄存器：配置 TIM1_CCMx 寄存器中的 OCxPE 位为 0；
   - 选择高电平有效：配置 TIM1_CCEN 寄存器中的 CCxP 为 0；
   - 使能输出：配置 TIM1_CCEN 寄存器中的 CCxE 为 1；
3) 使能计数器：配置 TIM1_CR1 寄存器中的 CEN 位为 1。

当未使能预装载寄存器（TIM1_CCMx 寄存器中的 OCxPE 位为 0）时，可实时更改 TIM1_CCx 寄存器的值，以控制输出波形。

**图 19-18 比较输出模式，翻转 OC1（OCxPE 位为 0）**



### 19.3.9 强制输出模式

在强制输出模式下，可配置输出信号为有效或无效电平，此时输出信号不受TIM1_CCx 影子寄存器和计数器之间的比较结果影响。

- 配置 TIM1_CCMx 寄存器中的 OCxM[2:0]位域为 101，可将输出信号（OCxREF）强置为高电平（OCxREF 始终为高电平有效）；
- 配置 TIM1_CCMx 寄存器中的 OCxM[2:0]位域为 100，可将输出信号（OCxREF）强置为低电平。

OCx 输出信号的极性由 TIM1_CCEN 寄存器中的 CCxP 位决定。

该模式下，TIM1_CCx 影子寄存器与计数器之间仍在进行比较，当产生匹配时，相应标志会被置位。

### 19.3.10 PWM 模式

PWM 模式生成的信号，其频率由 TIM1_ARR 寄存器的值决定，其占空比由TIM1_CCx 寄存器的值决定。

每个输出通道都可以独立选择 PWM 模式输出：

- 通过 TIM1_CCMx 寄存器中的 OCxM[2:0]位域来配置。
  - PWM 模式 1（OCxM[2:0]位域为 110）：在递增计数模式下，当TIM1_CNT＜TIM1_CCx，通道输出为有效电平，否则为无效电平。在递减计数模式下，当 TIM1_CNT＞TIM1_CCx，通道输出为无效电平，否则为有效电平；
  - PWM 模式 2（OCxM[2:0]位域为 111）：在递增计数模式下，当TIM1_CNT＜TIM1_CCx，通道输出为无效电平，否则为有效电平。在递减计数模式下，当 TIM1_CNT＞TIM1_CCx，通道输出为有效电平，否则为无效电平；
- 配置 TIM1_CCMx 寄存器中的 OCxPE 位置 1 使能相应比较寄存器的预装载功能（TIM1_CCx）；
- 配置 TIM1_CR1 寄存器中的 ARPE 位置 1 使能自动重载寄存器的预装载功能（TIM1_ARR）。

当使能预装载功能时，在发生更新事件时，预装载寄存器的值才会加载到影子寄存器中，所以启动计数器前，建议将 TIM1_EVTG 寄存器中的 UG 位置 1 来

初始化所有寄存器。

OCx 的极性可通过 TIM1_CCEN 寄存器的 CCxP 位来配置。通过 TIM1_CCEN 寄存器中的 CCxE 和 TIM1_BDT1 寄存器中的 MOEN 来控制输出。

根据 TIM1_CR1 寄存器中 CMS[1:0]位域的值，选择 PWM 信号的对齐模式：

**PWM 边沿对齐模式**

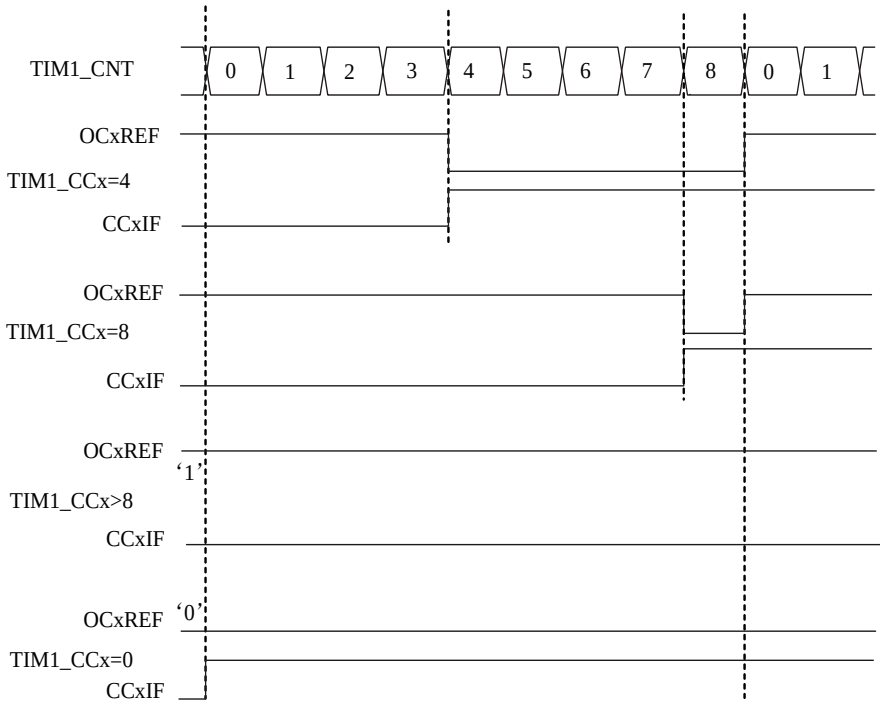
- 递增计数

  当 TIM1_CR1 寄存器中的 DIR 位为 0 时执行递增计数。请参考*递增计数模式*。

  下面是一个 PWM 模式 1 的例子。当 TIM1_CNT < TIM1_CCx，OCxREF 信号输出高电平，否则为低电平。如果 TIM1_CCx 寄存器的值大于 TIM1_ARR 寄存器的值，则 OCxREF 保持为 1。如果 TIM1_CCx 寄存器的值为 0，则 OCxREF 保持为 0。

  例如：TIM1_ARR 寄存器的值为 8 时，递增计数下的边沿对齐模式 PWM 波形。

图 19-19 递增计数下的边沿对齐模式 PWM 波形（TIM1_ARR 寄存器为 8）




- 递减计数

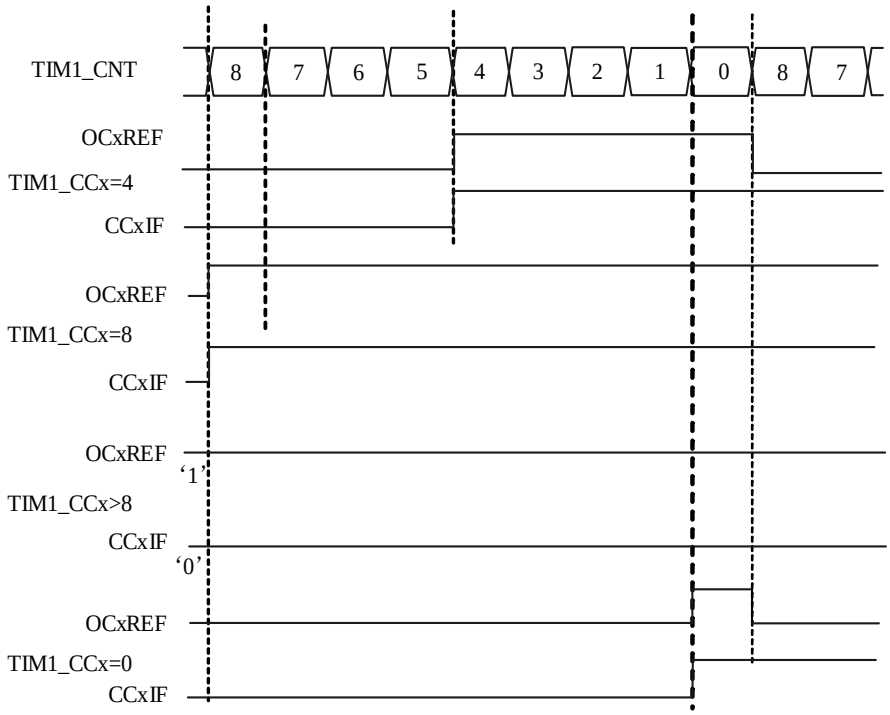
  当 TIM1_CR1 寄存器中的 DIR 位为 1 时执行递减计数。请参考*递减计数*

*模式*。

在 PWM 模式 1 下，当 TIM1_CNT > TIM1_CCx，OCxREF 信号输出低电平，否则为高电平。如果 TIM1_CCx 中的值大于自动重载值（TIM1_ARR 中的值），则 OCxREF 保持为 1。此模式下不可能产生占空比为 0%的 PWM 波形。

例如：TIM1_ARR 寄存器的值为 8 时，递减计数下的边沿对齐模式 PWM 波形。

图 19-20 递减计数下的边沿对齐模式 PWM 波形（TIM1_ARR 寄存器为 8）



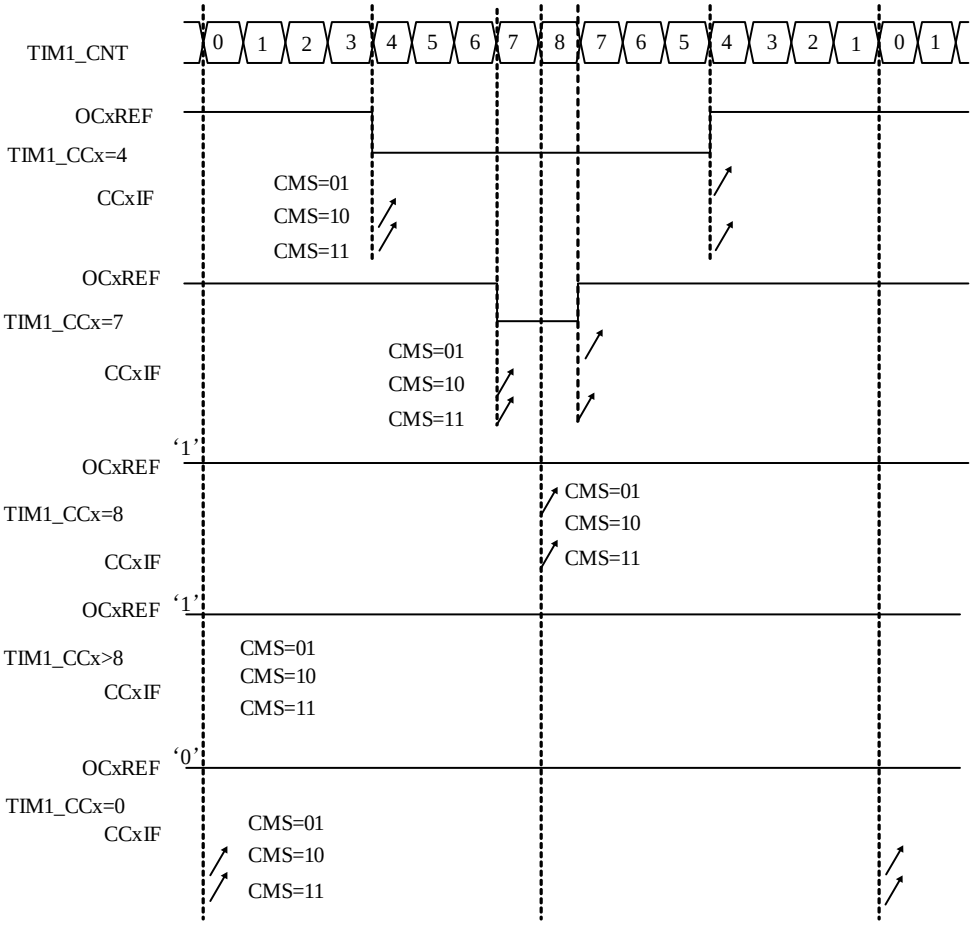

**PWM 中心对齐模式**

当 TIM1_CR1 寄存器中的 CMS[1:0]位域不为 00 时，为中心对齐模式。

如果配置 CMS[1:0]位域为中心对齐模式 3 时，在每个 PWM 周期内，计数器将产生上溢和下溢各一次，所以可更新 2 次 PWM 占空比。请参见*中心对齐模式（递增/递减计数）*。

例如：当 PWM 模式为 PWM 模式 1，TIM1_ARR 寄存器为 8 时，中心对齐模式的 PWM 波形。

图 19-21 中心对齐模式 PWM 波形（TIM1_ARR 寄存器为 8）




### 19.3.11 清除 OCxREF 信号

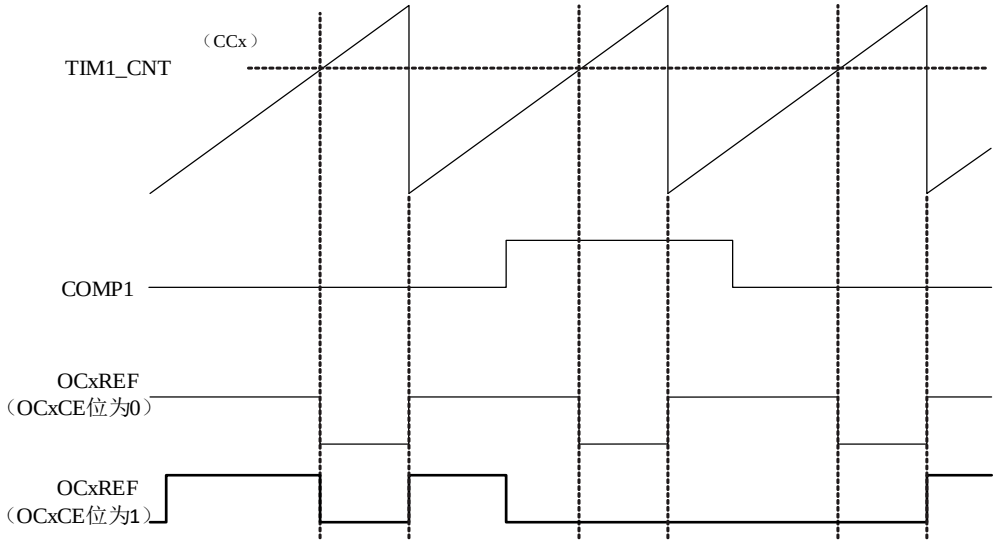
对于给定的通道，OC_CLR 信号上的高电平可将 OCxREF 信号复位，OCxREF 信号将保持低电平，直到发生更新事件。

TIM1_CFG 寄存器中的 OCREF_CLR 位为 0 时，COMP1 输出连接到 OC_CLR 输入；OCREF_CLR 位为 1 时，COMP2 输出连接到 OC_CLR 输入。

该功能只能用于比较输出模式和 PWM 模式（包含 PWM 中心对齐下移相使能时），在强制模式下不起作用。例如，OCxREF 信号可以连接到比较器的输出，用于控制电流。

下图为 OCxCE 不同配置下，当 COMP1 输入变为高电平时 OCxREF 的信号波形。在本例中，TIM1 配置为 PWM 模式。

图 19-22 清除 TIM1 的 OCxREF




### 19.3.12 单脉冲模式

单脉冲模式（OPM）是比较输出模式的一个特例。将 TIM1_CR1 寄存器中的 OPM 位置 1，即可选择单脉冲模式。在此模式下，将 CEN 位置 1，在一段可编程的延时后产生一个脉宽可编程的脉冲。

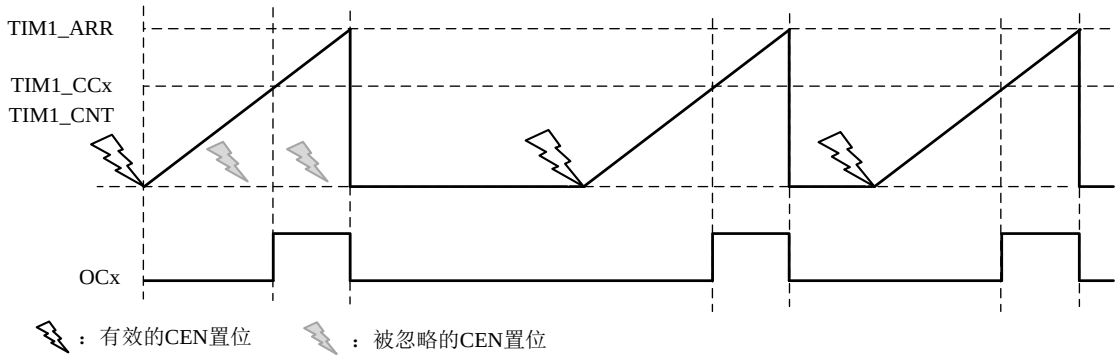
在比较输出模式或 PWM 模式下生成波形。当发生下一更新事件时或 UG 置 1 时，计数器将自动停止。

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。启动计数前，必须进行如下配置：

- 递增计数时：CNT < CCx ≤ ARR（特别注意，0 < CCx）
- 递减计数时：CNT > CCx

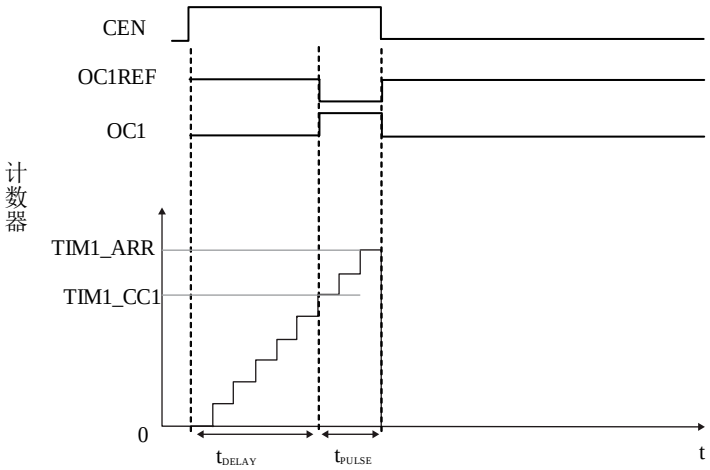
启动计数后，在一个计数周期再次置位 CEN 将被忽略。如下图所示：

图 19-23 单脉冲输出波形



例如，将 TIM1_CR1 寄存器的 CEN 置 1，经过 $t_{DELAY}$ 的延迟，在 OC1 上产生一个宽度为 $t_{PULSE}$ 的正脉冲。

图 19-24 单脉冲模式示例




单脉冲模式的脉冲宽度由写入的 TIM1_CC1 决定（考虑时钟频率和计数器预分频器）。

- $t_{DELAY}$ 由写入 TIM1_CC1 寄存器的值定义。
- $t_{PULSE}$ 由自动重载值与比较值之差（TIM1_ARR–TIM1_CC1）来定义。
- 若产生这样的波形：信号在发生比较匹配时从 0 变 1，在计数器达到自动重载值时由 1 变为 0：
  - OC1 输出极性：CC1P 位为 1
  - 使能 PWM 模式 1（TIM1_CCM1 寄存器中的 OC1M 写入 110）
  - 如果需要，可使能预装载功能（TIM1_CCM1 寄存器的 OC1PE 和 TIM1_CR1 寄存器的 ARPE 位置 1）
  - 在 TIM1_CC1 寄存器中写入比较值
  - 在 TIM1_ARR 寄存器中写入自动重载值
  - 将 UG 位置 1，之后等待再次启动计数

此例中，TIM1_CR1 寄存器中的 DIR 位应为 0，CMS[1:0]位域应为 00。

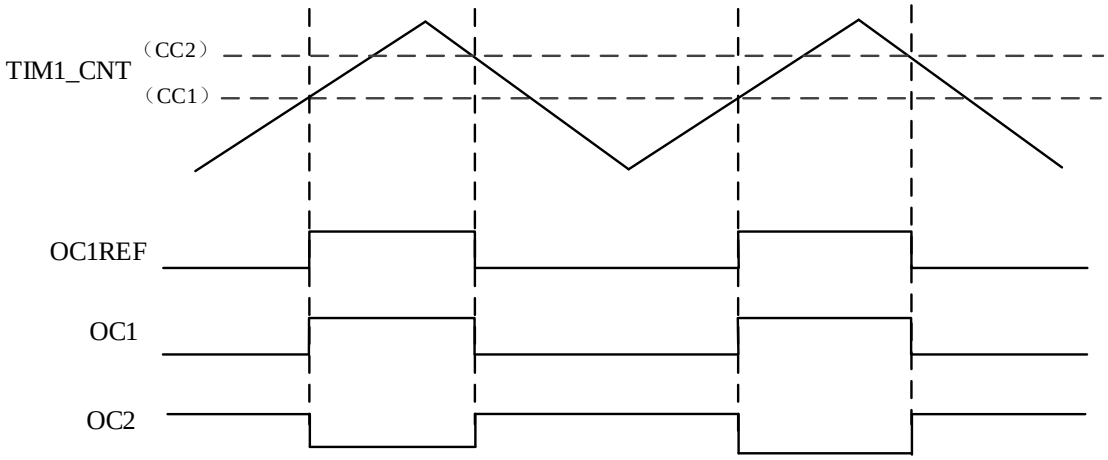
### 19.3.13 互补模式、PWM 移相及死区插入

通过配置 TIM1_CFG 寄存器的 MODE_SETx 位使能通道互补模式时，TIM1 的

互补输出信号 OCx（x＝1，3，5，7）和 OCy（y＝2，4，6，8）互补配对，其中 OC1 和 OC2 配对，OC3 和 OC4 配对，OC5 和 OC6 配对，OC7 和 OC8 配对，当通道 x 和 y 使用互补模式时：

- 输出比较模式由 OCxM[2:0]决定；
- 比较值由 CCx 和 CCy 决定，OCxREF 在计数器递增计数时与 CCx 比较，在递减计数时与 CCy 比较；
- 输出由通道 x 的 OCxREF 决定，通道 x 和通道 y 输出相反；
- 支持死区插入，左右死区时间可配；
- 两个通道可独立配置使能（TIM1_CCEN 寄存器中的 CCxE 和 CCyE 位）、极性（TIM1_CCEN 寄存器中的 CCxP 和 CCyP 位）、空闲电平（TIM1_CR2 寄存器中的 OISx 和 OISy 位）。

图 19-25 互补模式下比较输出示意图




互补通道有一个 8 位的死区发生器，每个通道的死区时间共用一组配置，分为左死区时间和右死区时间，可通过 TIM1_BDT2 寄存器中的 DTG_L[7:0]位域和 TIM1_BDT1 寄存器中的 DTG_R[7:0]位域进行配置：

- OCx 相对 OCxREF 的上升沿之间存在死区时间为左死区时间
- OCy 相对 OCxREF 的下降沿之间存在死区时间为右死区时间

如果死区时间大于有效输出（OCx 或 OCy）的宽度，则不会产生相应的脉冲。

下图所示为带死区插入的 OCx/OCy 与 OCxREF 之间的关系。（假定 CCxP 位为 0、CCyP 位为 0、MOEN 位为 1、CCxE 和 CCyE 位为 1）。

图 19-26 带死区插入的互补输出

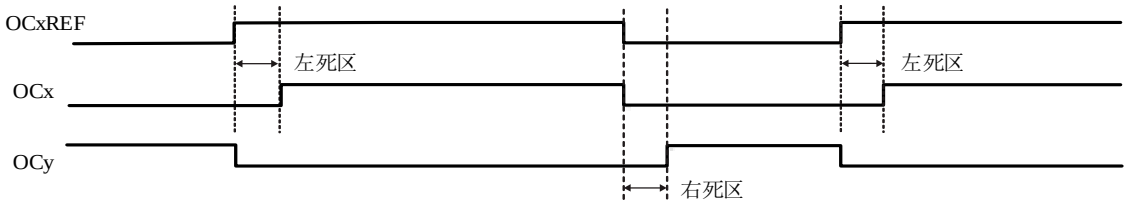



图 19-27 右死区时间大于负脉冲宽度的死区波形

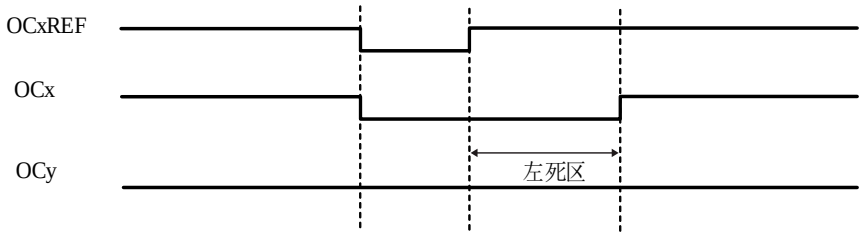
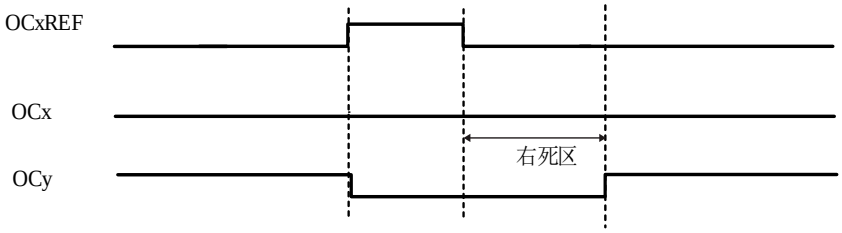



图 19-28 左死区时间大于正脉冲宽度的死区波形




**PWM 移相功能**

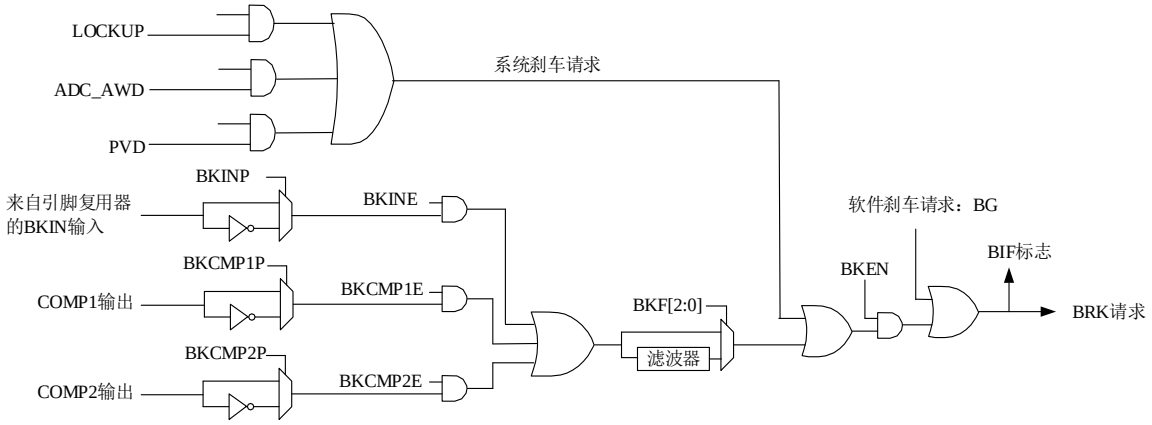
在通道使能互补模式时，PWM 中心对齐模式下，通过配置不同的 CCx 和 CCy 可产生一组具有可编程相位差的 PWM 信号。

以通道 1 和通道 2 使能互补模式为例，根据需要移动的相位，配置 CC1 和 CC2 的值即可实现 PWM 输出波形的可编程移相，可以左移或者右移，参见*图：互补模式下比较输出示意图*。

### 19.3.14 刹车及恢复功能

TIM1 支持 1 个刹车输入，通过将 TIM1_BDT1 寄存器中的 BKEN 位置 1 来使能刹车，通过 TIM1_BDT1 寄存器中的 BKF[2:0]位域来配置数字滤波。

图 19-29 刹车电路概述




### 刹车源

刹车通道源有多路可选，可通过配置 TIM1_AF1 寄存器的相应位来使能或禁止刹车事件源：

- 连接到 BKIN 引脚的外部源（由 GPIO 复用功能设定）
- 内部源：
  - CPU LOCKUP 事件输出；
  - PVD 报警事件输出；
  - ADC 模拟看门狗监控电压事件；
  - 比较器的输出，具有可配置的数字滤波和有效电平选择；
  - 软件触发刹车事件，通过配置 TIM1_EVTG 寄存器中的 BG 位。

### 刹车模式

通过配置 TIM_BDT1 寄存器的 BKMODE[1:0]位域可选择刹车模式。

当发生刹车时：

- MOEN 位清 0；
- 无论通道 CCxE 是否使能，都将输出 TIM1_CR2 寄存器预定义的 OISx 空闲电平；
- TIM1_SR 寄存器中的 BIF 位置 1，如果此时 TIM1_DIER 寄存器中的 BIE 位置 1，则会产生中断；
- 选择停止模式时，自动将 CEN 清 0，停止计数。

当刹车输入为有效电平时，MOEN 位无法置 1，同时 TIM1_SR 寄存器中的 BIF 位也无法清 0。

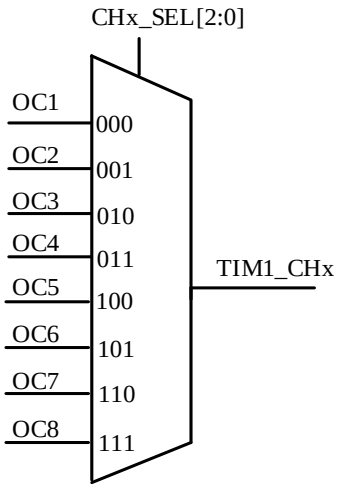
不同模式下刹车恢复行为不同，具体如下：

- 停止模式：需刹车信号撤销，清除 BIF 标志，使能 MOEN 和 CEN 后恢复输出。在刹车生效期间 TIM1 的计数停止。
- 暂停模式：需刹车信号撤销，BIF 标志清除后，自动在最近更新事件点将 MOEN 置 1 恢复输出，在刹车生效期间 TIM1 的计数不停止。
- 恢复模式：仅需要刹车撤销，标志无需清除，自动在最近更新事件点将 MOEN 置 1 恢复输出，同时清除 BIF 标志，在刹车生效期间 TIM1 的计数不停止。

### 19.3.15 输出通道重映射

通过配置 TIM1_OSEL 寄存器中的 CHx_SEL[2:0]位域，可将各通道的输出信号重映射至任意通道的输出 IO。

图 19-30 输出重映射示意图




### 19.3.16 生成 6 步 PWM

OCxM[2:0]位域支持预装载，发生换相（COM）事件时，将装载到影子寄存器位。

换相事件可通过配置 TIM1_EVTG 寄存器中的 COMG 位置 1 来触发，也可由 TRIG（TIM1 的 ITR0 来自 TIM3 的 TRIG_OUT）上升沿自动触发。

发生换相事件时，TIM1_SR 寄存器中的 COMIF 位会自动置 1。如果 TIM1_DIER

寄存器中的 COMIE 位置 1，将产生中断。

生成 6 步 PWM 示意图请参加*图：霍尔传感器接口的示例*。

### 19.3.17 霍尔传感器接口

使用高级控制定时器（TIM1）产生 PWM 信号驱动马达时，可采用通用 TIM3 定时器作为“接口定时器”来连接霍尔传感器。

示例：霍尔输入连接到 TIM3，每当霍尔输入发生变化后的一个指定时间后，改变 TIM1 的 PWM 配置（OCxM）。

对 TIM3 的配置，步骤如下：

- TIM3_CR2 寄存器的 TI1_XOR_SEL 位置 1，使 3 个定时器输入经过异或运算后到 TI1 输入通道；
- 配置 TIM3_ARR 寄存器的值（TIM3_CNT 的值必须通过 TI1 的变化清 0）。设置预分频值，以得到最大计数器周期，该周期大于传感器上两次变化的间隔时间；
- 将通道 1 配置为捕获模式（选择 TRC）：向 TIM3_CCM1 寄存器的 CC1S[1:0]位域写入 11。如果需要，还可配置数字滤波器；
- 配置为复位模式：TIM3_SMC 寄存器中的 SM_SEL[2:0]位域写入 100；
- 选择 TI1_FD 作为输入源：TIM3_SMC 寄存器中的 TS[2:0]位域写入 100；
- 将通道 2 配置为 PWM2 模式，并向 TIM3_CCM1 寄存器的 OC2M[2:0]位域写入 111，CC2S[1:0]位域写入 00；
- 选择 OC2REF 作为 TRIG_OUT 上的触发输出：向 TIM3_CR2 寄存器的 MM_SEL[2:0]位域写入 111。

对 TIM1 的配置，步骤如下：

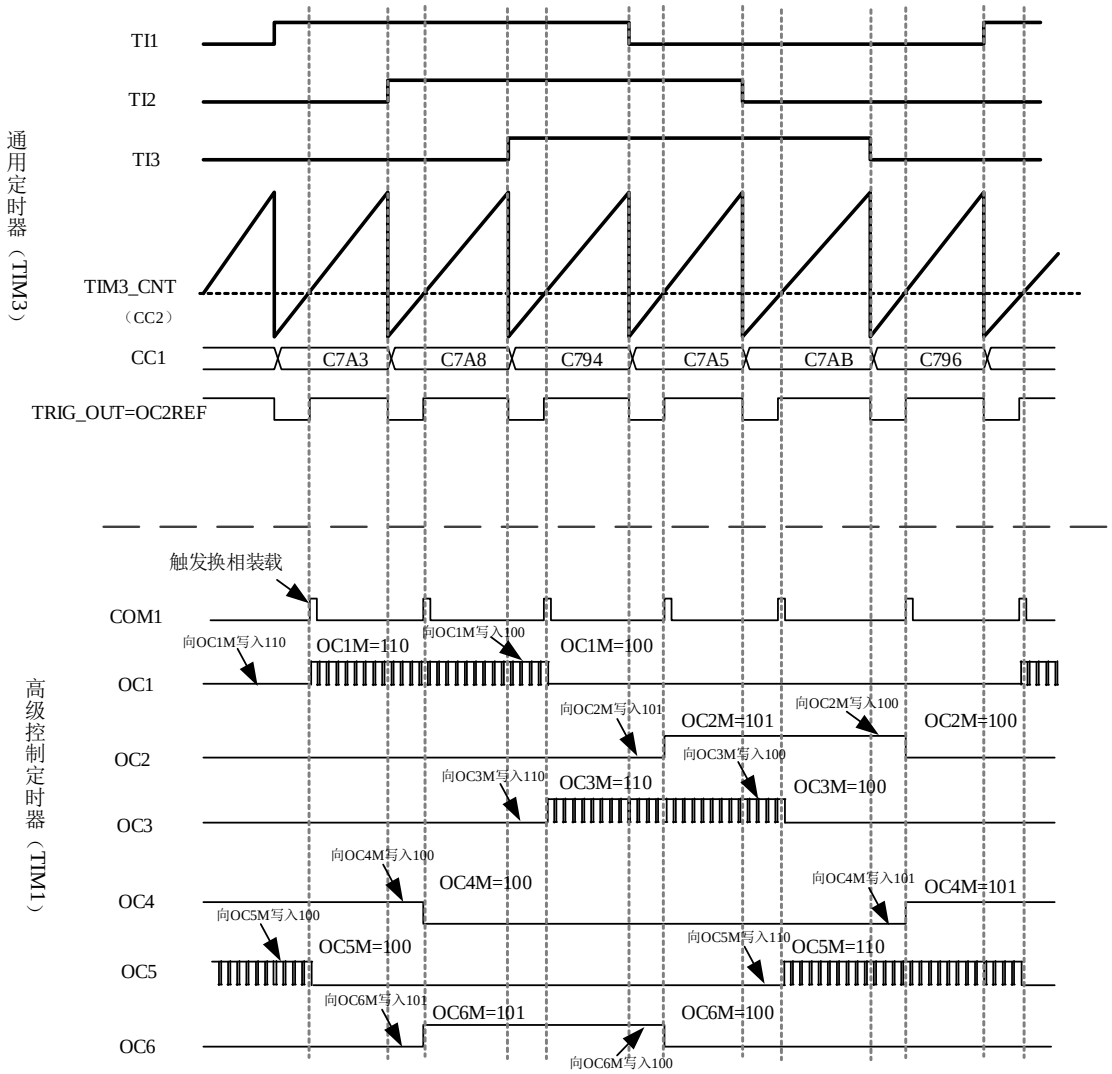
- 触发输入的上升沿生成一个换相事件：TIM1_CR2 寄存器中的 CCU_SEL 位置 1；
- 使能比较换相预装载：TIM1_CR2 寄存器的 CC_PRECR 位置 1；
- 如果此时 TIM1_DIER 寄存器中的 COMIE 位置 1，将产生中断。

在 TIM1 中，定时器配置为 PWM 模式，发生换相事件后，预装载使能位有效时，PWM 控制位（OCxM）中写入下一步的配置，此操作可在换相中断子程序

中完成。

下图为本示例的示意图。

图 19-31 霍尔传感器接口的示例




## 19.3.18 定时器同步主模式

TIM1 和内部其他 TIM 连在一起，可实现定时器同步或级联。

详细内容及相关配置请参见：*定时器同步主模式（TIM3）*。

## 19.3.19 DMA Burst 传输模式

TIM1 能够根据一个事件生成相应的 DMA 请求，将数据写入定时器的多个寄存器，也可读取定时器多个寄存器的值。关于 DMA Burst 功能，详见 *DMA 传输*。

例如，当发生更新事件，将数据依次更新到 TIM1_CCx 寄存器（x=1~8）中。

具体操作步骤如下（DMA 时钟已使能）：

1) 配置 DMA 通道（通道号请参见*表：DMA 通道请求信号*）：

   - DMA 通道的目标地址：TIM1_CC1 寄存器；
   - DMA 通道的源地址：要通过 DMA 传输到 TIM1_CCx 寄存器数据的 SRAM 缓冲区地址；
   - 根据配置目标地址为递增，源地址按需配置；
   - 要传输的数量：8；
   - 配置 DMA 的传输类型：*DMA 通道 x 控制寄存器（DMA_CCx）*中的 TYPE 位置 1 选择 Burst 传输；
   - 配置 DMA 的数据位宽：字（32bit）。

2) 使能 TIM1 的更新 DMA 请求：TIM1_DIER 寄存器中的 UDMA_EN 位置 1；

3) 使能 TIM1：TIM1_CR1 寄存器的 CEN 位置 1；

4) 使能 DMA 通道：*DMA 通道 x 控制寄存器（DMA_CCx）*中的 EN 位置 1。

### 19.3.20 调试模式

当 CPU 进入调试模式（Cortex-M0+内核停止）时，TIM1 计数器会根据 *APB2 冻结寄存器（DBG_APB2_FZ）*寄存器中的 TIM1_HOLD 配置位选择继续计数或者暂停计数：

- TIM1_HOLD 配置为 0：内核停止时，正常输出；
- TIM1_HOLD 配置为 1：内核停止时，固定输出空闲电平，计数暂停；当内核恢复运行时，计数器继续计数，立即恢复输出。

## 19.4 TIM1 中断

置位 TIM1_DIER 寄存器中的相关使能位，则以下事件会触发中断：

表 19-2 中断请求

| 中断事件 | 事件标志 | 中断使能位 | 事件标志/中断清除方法 |
|---|---|---|---|
| 更新事件 | UIF_OVR/UIF_UDR | UIE | 向 UIF_OVR/UIF_UDR 位写 0 |
| 比较 1~8 事件 | CCxIF | CCxIE | 向 CCxIF 位写 0 |

| 中断事件 | 事件标志 | 中断使能位 | 事件标志/中断清除方法 |
|---|---|---|---|
| 换相事件 | COMIF | COMIE | 向 COMIF 位写 0 |
| 刹车事件 | BIF | BIE | 向 BIF 位写 0 |

## 19.5 TIM1 寄存器

TIM1 寄存器支持 32 位访问。

表 19-3 TIM1 基地址

| 外设 | 基地址 |
|---|---|
| TIM1 | 0x4001 2C00 |

### 19.5.1 TIM1 控制寄存器 1（TIM1_CR1）

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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | CLK_DIV[1:0] | | ARPE | CMS[1:0] | | DIR | OPM | Res. | UDIS | CEN |
| | | | | | | rw | rw | rw | rw | rw | rw | rw | | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:10 | 保留 | 写入无效 |
| 9:8 | CLK_DIV[1:0] | 时钟分频<br>定时器时钟频率（$f_{TIM1_KCLK}$）与死区发生器以及数字滤波所使用的采样时钟（$f_{DTS}$）之间的分频比。<br>00：$f_{DTS} = f_{TIM1_KCLK}$<br>01：$f_{DTS} = f_{TIM1_KCLK}/2$<br>10：$f_{DTS} = f_{TIM1_KCLK}/4$<br>11：保留（默认为：$f_{DTS} = f_{TIM1_KCLK}$） |
| 7 | ARPE | TIM1_ARR寄存器预装载使能控制<br>0：禁止<br>1：使能 |
| 6:5 | CMS[1:0] | 计数模式选择<br>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。上溢或下溢均触发更新事件；仅在递减计数过程中，当发生比较匹配时，TIM1_SR寄存器中的CCxIF位会置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。上溢或下溢均触发更新事件；仅在递增计数过程中，当发生比较匹配时，TIM1_SR寄存器中的CCxIF位会置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。上溢或下溢均触发更新事件；在递增计数或递减计数过程中，当发生比较匹配时，TIM1_SR寄存器中的CCxIF位会置1

| | | |
|---|---|---|
| 4 | DIR | 计数方向<br>当定时器配置为中心对齐模式时，该位为只读状态。<br>0：递增计数<br>1：递减计数 |
| 3 | OPM | 单脉冲模式<br>0：禁止：计数器在发生更新事件时不会停止计数<br>1：使能：计数器在发生更新事件时停止计数（CEN位自动清0） |
| 2 | 保留 | 写入无效 |
| 1 | UDIS | 更新事件禁止控制位<br>0：使能：允许更新事件产生<br>1：禁止：禁止更新事件产生 |
| 0 | CEN | 计数器使能<br>0：禁止<br>1：使能<br>*注意：在单脉冲模式下，当发生更新事件时会自动将CEN位清0。刹车恢复选用停止模式时，发生刹车后，自动将CEN位清0。* |

### 19.5.2 TIM1 控制寄存器 2（TIM1_CR2）

偏移地址：0x04

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OIS8 | OIS7 | OIS6 | OIS5 | OIS4 | OIS3 | OIS2 | OIS1 | Res. | MM_SEL[3:0] | | | | CCU_SEL | Res. | CC_ PRECR |
| rw | rw | rw | rw | rw | rw | rw | rw | | rw | rw | rw | rw | rw | | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15 | OIS8 | OC8的空闲电平<br>请参见OIS1位 |
| 14 | OIS7 | OC7的空闲电平<br>请参见OIS1位 |
| 13 | OIS6 | OC6的空闲电平<br>请参见OIS1位 |
| 12 | OIS5 | OC5的空闲电平<br>请参见OIS1位 |
| 11 | OIS4 | OC4的空闲电平<br>请参见OIS1位 |
| 10 | OIS3 | OC3的空闲电平<br>请参见OIS1位 |
| 9 | OIS2 | OC2的空闲电平<br>请参见OIS1位 |
| 8 | OIS1 | OC1的空闲电平<br>0：当TIM1_BDT1寄存器的MOEN位为0，OC1输出低电平<br>1：当TIM1_BDT1寄存器的MOEN位为0，OC1输出高电平<br>*注意：当TIM1_LOCK寄存器中的LOCK位域定义为级别1、2时，此位即无法修改。* |
| 7 | 保留 | 写入无效 |

| | | |
|---|---|---|
| 6:3 | MM_SEL[3:0] | 主模式选择<br>触发输出（TRIG_OUT）信号连接到其他外设的控制方式。<br>0000：复位 — TIM1_EVTG寄存器中的UG位可以用作触发输出（TRIG_OUT）<br>0001：使能 — 计数器使能信号用作触发输出（TRIG_OUT）当TIM1_CR1寄存器的CEN位置位产生。为保证当前定时器与从外设实现完美同步，需选择主/从模式（TIM1_SMC寄存器中MS_MOD位置1）<br>0010：更新 — 选择更新事件作为触发输出（TRIG_OUT）<br>0011：上溢更新 — 仅计数上溢产生的更新事件作为触发输出（TRIG_OUT）<br>0100：下溢更新 — 仅计数下溢产生的更新事件作为触发输出（TRIG_OUT）<br>0101：比较脉冲 — 当通道1发生比较匹配时，输出（TRIG_OUT）会发送一个正脉冲<br>0110：比较 — OC1REF信号用作触发输出（TRIG_OUT）<br>0111：比较 — OC2REF信号用作触发输出（TRIG_OUT）<br>1000：比较 — OC3REF信号用作触发输出（TRIG_OUT）<br>1001：比较 — OC4REF信号用作触发输出（TRIG_OUT）<br>1010：比较 — OC5REF信号用作触发输出（TRIG_OUT）<br>1011：比较 — OC6REF信号用作触发输出（TRIG_OUT）<br>1100：比较 — OC7REF信号用作触发输出（TRIG_OUT）<br>1101：比较 — OC8REF信号用作触发输出（TRIG_OUT）<br>其他：保留（默认选择复位） |
| 2 | CCU_SEL | 换相控制更新选择<br>0：当CC_PRECR置1时，仅通过TIM1_EVTG寄存器中的COMG位置1，产生换相事件，此时对TIM1_CCM1寄存器的OCxM位进行更新<br>1：当CC_PRECR置1时，可通过将TIM1_EVTG寄存器中的COMG位置1或TRIG（ITR0）换相事件，此时对TIM1_CCM1寄存器的OCxM位进行更新 |
| 1 | 保留 | 写入无效 |
| 0 | CC_PRECR | 比较换相预装载控制<br>0：禁止：TIM1_CCM1寄存器的OCxM位未进行预装载 |

1：使能：TIM1_CCM1寄存器的OCxM位进行预装载，写入后，仅当发生换相事件（TIM1_EVTG寄存器中的COMG位置1或在TRIG（ITR0）上检测到上升沿，取决于CCU_SEL位）时才会对这些位进行更新

### 19.5.3 TIM1 从模式控制寄存器（TIM1_SMC）

偏移地址：0x08

复位值：0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 写入无效 |
| 7 | MS_MOD | 主/从模式<br>0：不执行任何操作<br>1：当前定时器的触发输入事件（TRIG）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TRIG_OUT）。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 |
| 6:0 | 保留 | 写入无效 |

### 19.5.4 TIM1 中断/DMA 使能寄存器（TIM1_DIER）

偏移地址：0x0C

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | CC8_DMAEN | CC7_DMAEN | CC6_DMAEN | CC5_DMAEN | Res. | | | | CC8IE | CC7IE | CC6IE | CC5IE |
| | | | | rw | rw | rw | rw | | | | | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | COM_DMAEN | CC4_DMAEN | CC3_DMAEN | CC2_DMAEN | CC1_DMAEN | UDMA_EN | BIE | Res. | COMIE | CC4IE | CC3IE | CC2IE | CC1IE | UIE |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | rw | rw | rw | rw | rw | rw | rw | | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:28 | 保留 | 写入无效 |
| 27 | CC8_DMAEN | 通道8比较事件的DMA请求使能<br>0：禁止<br>1：使能 |
| 26 | CC7_DMAEN | 通道7比较事件的DMA请求使能<br>0：禁止<br>1：使能 |
| 25 | CC6_DMAEN | 通道6比较事件的DMA请求使能<br>0：禁止<br>1：使能 |
| 24 | CC5_DMAEN | 通道5比较事件的DMA请求使能<br>0：禁止<br>1：使能 |
| 23:20 | 保留 | 写入无效 |
| 19 | CC8IE | 通道8比较事件的中断使能<br>0：禁止<br>1：使能 |
| 18 | CC7IE | 通道7比较事件的中断使能<br>0：禁止<br>1：使能 |
| 17 | CC6IE | 通道6比较事件的中断使能<br>0：禁止<br>1：使能 |
| 16 | CC5IE | 通道5比较事件的中断使能<br>0：禁止<br>1：使能 |

| | | |
|---|---|---|
| 15:14 | 保留 | 写入无效 |
| 13 | COM_DMAEN | 换相事件的DMA请求使能<br>0：禁止<br>1：使能 |
| 12 | CC4_DMAEN | 通道4比较事件的DMA请求使能<br>0：禁止<br>1：使能 |
| 11 | CC3_DMAEN | 通道3比较事件的DMA请求使能<br>0：禁止<br>1：使能 |
| 10 | CC2_DMAEN | 通道2比较事件的DMA请求使能<br>0：禁止<br>1：使能 |
| 9 | CC1_DMAEN | 通道1比较事件的DMA请求使能<br>0：禁止<br>1：使能 |
| 8 | UDMA_EN | 更新事件的DMA请求使能<br>0：禁止<br>1：使能 |
| 7 | BIE | 刹车事件的中断使能<br>0：禁止<br>1：使能 |
| 6 | 保留 | 写入无效 |
| 5 | COMIE | 换相事件的中断使能<br>0：禁止<br>1：使能 |
| 4 | CC4IE | 通道4比较事件的中断使能 |

0：禁止
1：使能

| | | |
|---|---|---|
| 3 | CC3IE | 通道3比较事件的中断使能<br>0：禁止<br>1：使能 |
| 2 | CC2IE | 通道2比较事件的中断使能<br>0：禁止<br>1：使能 |
| 1 | CC1IE | 通道1比较事件的中断使能<br>0：禁止<br>1：使能 |
| 0 | UIE | 更新事件的中断使能<br>0：禁止<br>1：使能 |

### 19.5.5 TIM1 状态寄存器（TIM1_SR）

偏移地址：0x10

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | UIF_UDR | UIF_OVR | Res. | | | | CC8IF | CC7IF | CC6IF | CC5IF |
| | | | | | | rc_w0 | rc_w0 | | | | | rc_w0 | rc_w0 | rc_w0 | rc_w0 |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | BIF | Res. | COMIF | CC4IF | CC3IF | CC2IF | CC1IF | Res. |
| | | | | | | | | rc_w0 | | rc_w0 | rc_w0 | rc_w0 | rc_w0 | rc_w0 | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:26 | 保留 | 写入无效 |
| 25 | UIF_UDR | 计数下溢更新事件标志<br>发生计数下溢更新事件时自动置1。通过写0可清除该位。如果TIM1_DIER寄存器中UIE位为1，触发中断。<br>0：未发生下溢更新 |

| | | |
|---|---|---|
| | | 1：发生下溢更新 |
| 24 | UIF_OVR | 计数上溢更新事件标志<br>发生计数上溢更新事件时自动置1。通过写0可清除该位。如果TIM1_DIER寄存器中UIE位为1，触发中断。<br>0：未发生上溢更新<br>1：发生上溢更新 |
| 23:20 | 保留 | 写入无效 |
| 19 | CC8IF | 通道8的比较事件标志<br>请参见CC1IF说明 |
| 18 | CC7IF | 通道7的比较事件标志<br>请参见CC1IF说明 |
| 17 | CC6IF | 通道6的比较事件标志<br>请参见CC1IF说明 |
| 16 | CC5IF | 通道5的比较事件标志<br>请参见CC1IF说明 |
| 15:8 | 保留 | 写入无效 |
| 7 | BIF | 刹车事件标志<br>当发生断刹车事件时，该位自动置1，如果TIM1_DIER寄存器中BIE位为1时，生成中断。刹车事件消失后，通过写0可清除该位。<br>0：未发生刹车事件<br>1：发生刹车事件 |
| 6 | 保留 | 写入无效 |
| 5 | COMIF | 换相事件标志<br>当发生换相事件时，该位自动置1，同时TIM1_CCMx寄存器的OCxM位被更新。通过写0可清除该位。如果TIM1_DIER寄存器中COMIE位为1，触发中断。<br>0：未发生换相事件<br>1：发生换相事件 |

| | | |
|---|---|---|
| 4 | CC4IF | 通道4的比较事件标志<br>请参见CC1IF说明 |
| 3 | CC3IF | 通道3的比较事件标志<br>请参见CC1IF说明 |
| 2 | CC2IF | 通道2的比较事件标志<br>请参见CC1IF说明 |
| 1 | CC1IF | 通道1的比较事件标志<br>发生比较事件时，如果TIM1_DIER寄存器中CCxIE位为1，触发中断。通过写0可清除该位。<br>该标志置位条件如下：<br>– 边沿对齐模式下，当TIM1_CNT计数器的值与TIM1_CC1寄存器的值匹配时，此标志自动置1；<br>– 中心对齐模式下，请参见TIM1_CR1寄存器中的CMS位说明。其中，中心对齐模式3下，如果通道是互补模式，则CC1IF仅在递增计数时置位，CC2IF仅在递减计数时置位。<br>0：不匹配<br>1：匹配 |
| 0 | 保留 | 写入无效 |

### 19.5.6 TIM1 事件产生寄存器（TIM1_EVTG）

偏移地址：0x14

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | BG | Res. | COMG | Res. | | | | UG |
| | | | | | | | | w | | w | | | | | w |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 写入无效 |

| | | |
|---|---|---|
| 7 | BG | 生成刹车事件<br>软件置1可生成刹车事件，此时MOEN位清0，且BIF标志置1。<br>如果TIM1_DIER寄存器中BIE位为1，触发中断。 |
| 6 | 保留 | 写入无效 |
| 5 | COMG | 生成换相事件<br>软件置1可生成换相事件。如果TIM1_DIER寄存器中COMIE位为1，触发中断。如果CC_PRECR位置1时，可更新TIM1_CCMx寄存器的OCxM位。 |
| 4:1 | 保留 | 写入无效 |
| 0 | UG | 初始化生成<br>此位由软件置1可生成相应加载和初始化计数器效果。<br>0：不执行任何操作<br>1：计数器，预分频计数器，重复计数器均被初始化，并加载使能预装载功能的影子寄存器和PSC寄存器（UDIS位为0时） |

### 19.5.7 TIM1 计数值寄存器（TIM1_CNT）

偏移地址：0x18

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15:0 | CNT[15:0] | 计数器值 |

### 19.5.8 TIM1 重复计数器寄存器（TIM1_RCR）

偏移地址：0x1C

复位值：0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 写入无效 |
| 7:0 | REP[7:0] | 重复计数器值<br>可配置更新事件产生的频率。<br>每次重复计数器递减计数到0时，将生成一个更新事件，此时重复计数器会加载REP[7:0]的值，并重新开始计数。<br>在生成更新事件之前，无论向该寄存器写入何值都不生效。 |

### 19.5.9 TIM1 比较使能寄存器（TIM1_CCEN）

偏移地址：0x20

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | CC8P | CC8E | Res. | | CC7P | CC7E | Res. | | CC6P | CC6E | Res. | | CC5P | CC5E |
| | | rw | rw | | | rw | rw | | | rw | rw | | | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | CC4P | CC4E | Res. | | CC3P | CC3E | Res. | | CC2P | CC2E | Res. | | CC1P | CC1E |
| | | rw | rw | | | rw | rw | | | rw | rw | | | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:30 | 保留 | 写入无效 |
| 29 | CC8P | 通道8比较输出极性<br>请参见CC1P说明 |

| | | |
|---|---|---|
| 28 | CC8E | 通道8比较输出使能<br>请参见CC1E说明 |
| 27:26 | 保留 | 写入无效 |
| 25 | CC7P | 通道7比较输出极性<br>请参见CC1P说明 |
| 24 | CC7E | 通道7比较输出使能<br>请参见CC1E说明 |
| 23:22 | 保留 | 写入无效 |
| 21 | CC6P | 通道6比较输出极性<br>请参见CC1P说明 |
| 20 | CC6E | 通道6比较输出使能<br>请参见CC1E说明 |
| 19:18 | 保留 | 写入无效 |
| 17 | CC5P | 通道5比较输出极性<br>请参见CC1P说明 |
| 16 | CC5E | 通道5比较输出使能<br>请参见CC1E说明 |
| 15:14 | 保留 | 写入无效 |
| 13 | CC4P | 通道4比较输出极性<br>请参见CC1P说明 |
| 12 | CC4E | 通道4比较输出使能<br>请参见CC1E说明 |
| 11:10 | 保留 | 写入无效 |
| 9 | CC3P | 通道3比较输出极性 |

| | | |
|---|---|---|
| | | 请参见CC1P说明 |
| 8 | CC3E | 通道3比较输出使能<br>请参见CC1E说明 |
| 7:6 | 保留 | 写入无效 |
| 5 | CC2P | 通道2比较输出极性<br>请参见CC1P说明 |
| 4 | CC2E | 通道2比较输出使能<br>请参见CC1E说明 |
| 3:2 | 保留 | 写入无效 |
| 1 | CC1P | 通道1比较输出极性<br>0：OC1高电平有效<br>1：OC1低电平有效 |
| 0 | CC1E | 通道1比较输出使能<br>0：禁止<br>1：使能（OC1输出具体取决于该位和MOEN、CC1P、OIS1位） |

### 19.5.10 TIM1 比较模式寄存器 1（TIM1_CCM1）

偏移地址：0x28

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OC4CE | OC4M[2:0] | | | OC4PE | Res. | | | OC3CE | OC3M[2:0] | | | OC3PE | Res. | | |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OC2CE | OC2M[2:0] | | | OC2PE | Res. | | | OC1CE | OC1M[2:0] | | | OC1PE | Res. | | |
| rw | rw | rw | rw | rw | | | | rw | rw | rw | rw | rw | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31 | OC4CE | 通道4输出比较清零使能<br>请参见OC1CE说明 |

| | | |
|---|---|---|
| 30:28 | OC4M[2:0] | 通道4输出比较模式选择<br>请参见OC1M[2:0]说明 |
| 27 | OC4PE | 通道4输出比较预装载使能<br>请参见OC1PE说明 |
| 26:24 | 保留 | 写入无效 |
| 23 | OC3CE | 通道3输出比较清零使能<br>请参见OC1CE说明 |
| 22:20 | OC3M[2:0] | 通道3输出比较模式选择<br>请参见OC1M[2:0]说明 |
| 19 | OC3PE | 通道3输出比较预装载使能<br>请参见OC1PE说明 |
| 18:16 | 保留 | 写入无效 |
| 15 | OC2CE | 通道2输出比较清零使能<br>请参见OC1CE说明 |
| 14:12 | OC2M[2:0] | 通道2输出比较模式选择<br>请参见OC1M[2:0]说明 |
| 11 | OC2PE | 通道2输出比较预装载使能<br>请参见OC1PE说明 |
| 10:8 | 保留 | 写入无效 |
| 7 | OC1CE | 通道1输出比较清零使能<br>0：OC1REF不受OC_CLR信号影响<br>1：OC_CLR信号上检测到高电平时，OC1REF立即清0 |
| 6:4 | OC1M[2:0] | 通道1输出比较模式选择<br>定义信号OC1REF的行为，OC1REF信号决定OC1的值。<br>OC1REF为高电平有效。OC1的有效电平则取决于TIM1_CCEN |

寄存器的CC1P位。

000：冻结 — TIM1_CC1与TIM1_CNT间的比较对输出无影响

001：匹配输出有效电平 — 当TIM1_CNT与TIM1_CC1匹配时，OC1REF信号强制变为有效电平（匹配前OC1REF信号为无效电平）

010：匹配输出无效电平 — 当TIM1_CNT与TIM1_CC1匹配时，OC1REF信号强制变为无效电平（匹配前OC1REF信号为无效电平）

011：翻转 — 当TIM1_CNT与TIM1_CC1匹配时，OC1REF发生翻转

100：强制变为无效电平 — OC1REF强制变为无效电平

101：强制变为有效电平 — OC1REF强制变为有效电平

110：PWM模式1：

- 在递增计数模式下，当TIM1_CNT < TIM1_CC1，通道1输出为有效电平，否则为无效电平
- 在递减计数模式下，当TIM1_CNT > TIM1_CC1，通道1输出为无效电平，否则为有效电平

111：PWM模式2：

- 在递增计数模式下，当TIM1_CNT < TIM1_CC1，通道1输出为无效电平，否则为有效电平
- 在递减计数模式下，当TIM1_CNT > TIM1_CC1，通道1输出为有效电平，否则为无效电平

*注意：如果TIM1_CR2寄存器中的CC_PRECR位置1，则仅当生成换相事件时，OC1M有效位才会从预装载位获取新值。*

| | | |
|---|---|---|
| 3 | OC1PE | 通道 1 输出比较预装载使能<br>0：禁止。可随时向 TIM1_CC1 写入数据，写入后将立即生效<br>1：使能。可读/写访问预装载寄存器，TIM1_CC1 预装载值在更新事件时才会生效 |
| 2:0 | 保留 | 写入无效 |

### 19.5.11 TIM1 比较模式寄存器 2（TIM1_CCM2）

偏移地址：0x2C

复位值：0x0000 0000

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

| OC8CE | OC8M[2:0] | | | OC8PE | Res. | | | OC7CE | OC7M[2:0] | | | OC7PE | Res. | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |
| OC6CE | OC6M[2:0] | | | OC6PE | Res. | | | OC5CE | OC5M[2:0] | | | OC5PE | Res. | | |
| rw | rw | rw | rw | rw | | | | rw | rw | rw | rw | rw | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31 | OC8CE | 通道8输出比较清零使能<br>请参见TIM1_CCM1寄存器的OC1CE说明 |
| 30:28 | OC8M[2:0] | 通道8输出比较模式<br>请参见TIM1_CCM1寄存器的OC1M[2:0]说明 |
| 27 | OC8PE | 通道8输出比较预装载使能<br>请参见TIM1_CCM1寄存器的OC1PE说明 |
| 26:24 | 保留 | 写入无效 |
| 23 | OC7CE | 通道7输出比较清零使能<br>请参见TIM1_CCM1寄存器的OC1CE说明 |
| 22:20 | OC7M[2:0] | 通道7输出比较模式选择<br>请参见TIM1_CCM1寄存器的OC1M[2:0]说明 |
| 19 | OC7PE | 通道7输出比较预装载使能<br>请参见TIM1_CCM1寄存器的OC1PE说明 |
| 18:16 | 保留 | 写入无效 |
| 15 | OC6CE | 通道6输出比较清零使能<br>请参见TIM1_CCM1寄存器的OC1CE说明 |
| 14:12 | OC6M[2:0] | 通道6输出比较模式<br>请参见TIM1_CCM1寄存器的OC1M[2:0]说明 |
| 11 | OC6PE | 通道6输出比较预装载使能<br>请参见TIM1_CCM1寄存器的OC1PE说明 |

| | | |
|---|---|---|
| 10:8 | 保留 | 写入无效 |
| 7 | OC5CE | 通道5输出比较清零使能<br>请参见TIM1_CCM1寄存器的OC1CE说明 |
| 6:4 | OC5M[2:0] | 通道5输出比较模式选择<br>请参见TIM1_CCM1寄存器的OC1M[2:0]说明 |
| 3 | OC5PE | 通道5输出比较预装载使能<br>请参见 TIM1_CCM1 寄存器的 OC1PE 说明 |
| 2:0 | 保留 | 写入无效 |

### 19.5.12 TIM1 预分频寄存器（TIM1_PSC）

偏移地址：0x34

复位值：0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:4 | 保留 | 写入无效 |
| 3:0 | PSC[3:0] | 预分频值<br>计数器时钟频率（CK_CNT）等于 $f_{CK_PSC}/（2^{PSC[3:0]}）$。<br>当发生更新事件时，该位域的值加载到预分频影子寄存器中。 |

### 19.5.13 TIM1 自动重载值寄存器（TIM1_ARR）

偏移地址：0x38

复位值：0x0000 FFFF

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15:0 | ARR[15:0] | 自动重载值<br>当自动重载值为零时，计数器暂停计数。<br>如果 TIM1_CR1 寄存器的 ARPE 位为 0，TIM1_ARR 寄存器的值会立即更新。如果 ARPE 位为 1 时，只有当发生更新事件，TIM1_ARR 寄存器中的值才会更新到影子寄存器中。 |

### 19.5.14 TIM1 比较寄存器 x（TIM1_CCx）

偏移地址：0x3C+0x04 × x，（x = 0 ~ 7）

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15:0 | CCx[15:0] | 通道x的比较值（x = 1 ~ 8）<br>– 当TIM1_CCM1/2寄存器中的OCxPE位为1时，使能预装载功能，该位域的新值将在更新事件后生效<br>– 当TIM1_CCM1/2寄存器中的OCxPE位为0时，该位域的新值立即生效 |

### 19.5.15 TIM1 刹车和死区寄存器 1（TIM1_BDT1）

偏移地址：0x6C

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BKMODE[1:0] | | Res. | | | | | | | | | | | BKF[2:0] | | |
| rw | rw | | | | | | | | | | | | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MOEN | Res. | | BKEN | Res. | | | | DTG_R[7:0] | | | | | | | |
| rw | | | rw | | | | | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:30 | BKMODE[1:0] | 刹车模式选择<br>00：停止模式<br>01：暂停模式<br>10：恢复模式<br>11：保留（默认停止模式） |
| 29:19 | 保留 | 写入无效 |
| 18:16 | BKF[2:0] | 刹车滤波器控制<br>刹车信号的采样频率和数字滤波宽度。连续 4 个采样周期均采样到有效电平时，才视为一个有效电平：<br>000：无滤波器<br>001：$f_{SAMPLING} = f_{TIM1_KCLK}$<br>010：$f_{SAMPLING} = f_{DTS}$<br>011：$f_{SAMPLING} = f_{DTS}/2$<br>100：$f_{SAMPLING} = f_{DTS}/4$<br>101：$f_{SAMPLING} = f_{DTS}/8$<br>110：$f_{SAMPLING} = f_{DTS}/16$<br>111：$f_{SAMPLING} = f_{DTS}/32$ |
| 15 | MOEN | 主输出使能<br>当刹车发生时，此位自动异步清0。此位由软件置1或根据BKMODE[1:0]位域的配置自动置1。<br>0：OCx被强制为空闲电平状态<br>1：若OCx输出使能位CCxE置1，则使能OCx输出 |
| 14:13 | 保留 | 写入无效 |
| 12 | BKEN | 刹车功能使能位<br>0：禁止 |

| | | |
|---|---|---|
| | | 1：使能 |
| 11:8 | 保留 | 写入无效 |
| 7:0 | DTG_R[7:0] | 配置右死区发生器<br>定义互补输出之间的右侧死区时间。其中DT为死区时间。<br>DTG_R [7:5] = 0xx：DT=DTG_R [7:0] ×$t_{DTG}$，其中$t_{DTG}$=$t_{DTS}$。<br>DTG_R [7:5] = 10x：DT=（64+DTG_R [5:0]）×$t_{DTG}$，其中$t_{DTG}$=2×$t_{DTS}$。<br>DTG_R [7:5] = 110：DT=（32+DTG_R [4:0]）×$t_{DTG}$，其中$t_{DTG}$=8×$t_{DTS}$。<br>DTG_R [7:5] = 111：DT=（32+DTG_R [4:0]）×$t_{DTG}$，其中$t_{DTG}$=16×$t_{DTS}$。 |

### 19.5.16 TIM1 刹车和死区寄存器 2（TIM1_BDT2）

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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 写入无效 |
| 7:0 | DTG_L[7:0] | 配置左死区发生器<br>定义互补输出之间的左侧死区时间，请参见DTG_R[7:0]说明。 |

### 19.5.17 TIM1 配置寄存器（TIM1_CFG）

偏移地址：0x74

复位值：0x0000 0000

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

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | MODE_SET4 | Res. | MODE_SET3 | Res. | MODE_SET2 | Res. | MODE_SET1 | Res. | | | | | | | OCREF_CLR |
| | rw | | rw | | rw | | rw | | | | | | | | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:15 | 保留 | 写入无效 |
| 14 | MODE_SET4 | 通道7/8互补模式选择<br>请参见MODE_SET1说明 |
| 13 | 保留 | 写入无效 |
| 12 | MODE_SET3 | 通道5/6互补模式选择<br>请参见MODE_SET1说明 |
| 11 | 保留 | 写入无效 |
| 10 | MODE_SET2 | 通道3/4互补模式选择<br>请参见MODE_SET1说明 |
| 9 | 保留 | 写入无效 |
| 8 | MODE_SET1 | 通道1/2互补模式选择<br>0：独立模式<br>1：互补模式工作，硬件插入死区 |
| 7:1 | 保留 | 写入无效 |
| 0 | OCREF_CLR | OC_CLR源选择<br>0：COMP1输出连接到OC_CLR输入<br>1：COMP2输出连接到OC_CLR输入 |

### 19.5.18 TIM1 轮换功能寄存器 1（TIM1_AF1）

偏移地址：0x78

复位值：0x0000 0000

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

| Res. | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | | | | BK CMP2P | BK CMP1P | BKINP | PVD_ LOCK | LOCKUP_ LOCK | AWD_ LOCK | Res. | | | BK CMP2E | BK CMP1E | BKINE |
| | | | | rw | rw | rw | rw | rw | rw | | | | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:12 | 保留 | 写入无效 |
| 11 | BKCMP2P | COMP2输入极性选择<br>0：COMP2输入低电平有效<br>1：COMP2输入高电平有效 |
| 10 | BKCMP1P | COMP1输入极性选择<br>0：COMP1输入低电平有效<br>1：COMP1输入高电平有效 |
| 9 | BKINP | BKIN输入极性选择<br>0：BKIN输入低电平有效<br>1：BKIN输入高电平有效 |
| 8 | PVD_LOCK | 用于控制PVD与TIM1刹车输入的连接。<br>软件置1，写0无效，直到下次复位时清0。<br>0：PVD报警与TIM1刹车输入断开<br>1：PVD报警与TIM1刹车输入连接 |
| 7 | LOCKUP_LOCK | 用于控制LOCKUP与TIM1刹车输入的连接。<br>软件置1，写0无效，直到下次复位时清0<br>0：LOCKUP输出与TIM1刹车输入断开<br>1：LOCKUP输出与TIM1刹车输入连接 |
| 6 | AWD_LOCK | 用于控制ADC_AWD与TIM1刹车输入的连接。<br>软件置1，写0无效，直到下次复位时清0。<br>0：ADC_AWD报警与TIM1刹车输入断开<br>1：ADC_AWD报警与TIM1刹车输入连接 |
| 5:3 | 保留 | 写入无效 |

| | | |
|---|---|---|
| 2 | BKCMP2E | COMP2刹车输入使能<br>0：禁止<br>1：使能 |
| 1 | BKCMP1E | COMP1刹车输入使能<br>0：禁止<br>1：使能 |
| 0 | BKINE | BKIN刹车输入使能<br>0：禁止<br>1：使能 |

### 19.5.19 TIM1 输出选择寄存器（TIM1_OSEL）

偏移地址：0x84

复位值：0x7654 3210

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | CH8_SEL[2:0] | | | Res. | CH7_SEL[2:0] | | | Res. | CH6_SEL[2:0] | | | Res. | CH5_SEL[2:0] | | |
| | rw | rw | rw | | rw | rw | rw | | rw | rw | rw | | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | CH4_SEL[2:0] | | | Res. | CH3_SEL[2:0] | | | Res. | CH2_SEL[2:0] | | | Res. | CH1_SEL[2:0] | | |
| | rw | rw | rw | | rw | rw | rw | | rw | rw | rw | | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31 | 保留 | 写入无效 |
| 30:28 | CH8_SEL[2:0] | 通道8输出信号源选择<br>请参见CH1_SEL[2:0]位域说明。 |
| 27 | 保留 | 写入无效 |
| 26:24 | CH7_SEL[2:0] | 通道7输出信号源选择<br>请参见CH1_SEL[2:0]位域说明。 |
| 23 | 保留 | 写入无效 |
| 22:20 | CH6_SEL[2:0] | 通道6输出信号源选择 |

| | | |
|---|---|---|
| | | 请参见CH1_SEL[2:0]位域说明。 |
| 19 | 保留 | 写入无效 |
| 18:16 | CH5_SEL[2:0] | 通道5输出信号源选择<br>请参见CH1_SEL[2:0]位域说明。 |
| 15 | 保留 | 写入无效 |
| 14:12 | CH4_SEL[2:0] | 通道4输出信号源选择<br>请参见CH1_SEL[2:0]位域说明。 |
| 11 | 保留 | 写入无效 |
| 10:8 | CH3_SEL[2:0] | 通道3输出信号源选择<br>请参见CH1_SEL[2:0]位域说明。 |
| 7 | 保留 | 写入无效 |
| 6:4 | CH2_SEL[2:0] | 通道2输出信号源选择<br>请参见CH1_SEL[2:0]位域说明。 |
| 3 | 保留 | 写入无效 |
| 2:0 | CH1_SEL[2:0] | 通道1输出信号源选择<br>000：OC1<br>001：OC2<br>010：OC3<br>011：OC4<br>100：OC5<br>101：OC6<br>110：OC7<br>111：OC8 |

### 19.5.20 TIM1 写使能寄存器（TIM1_LOCK）

偏移地址：0x90

复位值：0x0000 0002

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:2 | 保留 | 写入无效 |
| 1:0 | LOCK[1:0] | TIM1寄存器锁定控制<br>00：关闭锁定（不对任何位提供写保护），对TIM1_LOCK寄存器低8位写入0x55时，该位域变为00<br>01：锁定级别1，对TIM1_LOCK寄存器低8位写任意非0x55和0xAA的值时，该位域变为01，此时无法对以下寄存器相应位执行写操作：<br>– TIM1_CR2寄存器中的OISx位<br>– TIM1_BDT1寄存器中的BKMODE[1:0]、BKF[2:0]、BKEN和DTG_R[7:0]位域<br>– TIM1_BDT2寄存器中的DTG_L[7:0]位域<br>– TIM1_AF1寄存器<br>10：锁定级别2，对TIM1_LOCK寄存器低8位写入0xAA时，该位域变为10，此时无法对以下寄存器相应位执行写操作：<br>– 锁定级别1中适用的各位域<br>– TIM1_CCEN寄存器中的CCxP位<br>– TIM1_CCMx寄存器中的OCxPE和OCxM[2:0]位域 |

# 20 通用定时器（TIM3/4）

## 20.1 简介

通用定时器 TIM3 由一个 16 位自动重载计数器构成。通用定时器 TIM4 由一个 32 位自动重载计数器构成。支持测量输入信号的脉冲宽度（输入捕获），或者生成输出波形（比较输出、单脉冲输出和 PWM）等多种用途。

通用定时器 TIM3/4 和其他定时器结合使用，可实现定时器互联功能。

## 20.2 TIM3/4 主要特性

- 16 位（TIM3）/32 位（TIM4）递增、递减、递增/递减自动重载计数器
- 可编程预分频器
- 多个独立通道，各通道可用于：
  - 输入捕获，支持捕获预分频
  - 输出比较
  - PWM 输出
  - 单脉冲模式输出
- 多种编码器：
  - 正交编码
  - 脉冲电平编码
- 可实现定时器的启动、停止、初始化
- PWM 边沿或周期点可触发启动 ADC 转换
- 支持与多个定时器互联，以实现定时器换相、同步或级联
- 支持与比较器互联实现开窗
- TIM3 与 TIM4 互联，实现编码器信号捕获
- 与 OPAMP2 互联实现输入开关切换
- 支持 DMA

## 20.3 TIM3/4 功能描述

### 20.3.1 TIM3/4 框图

图 20-1 TIM3 框图

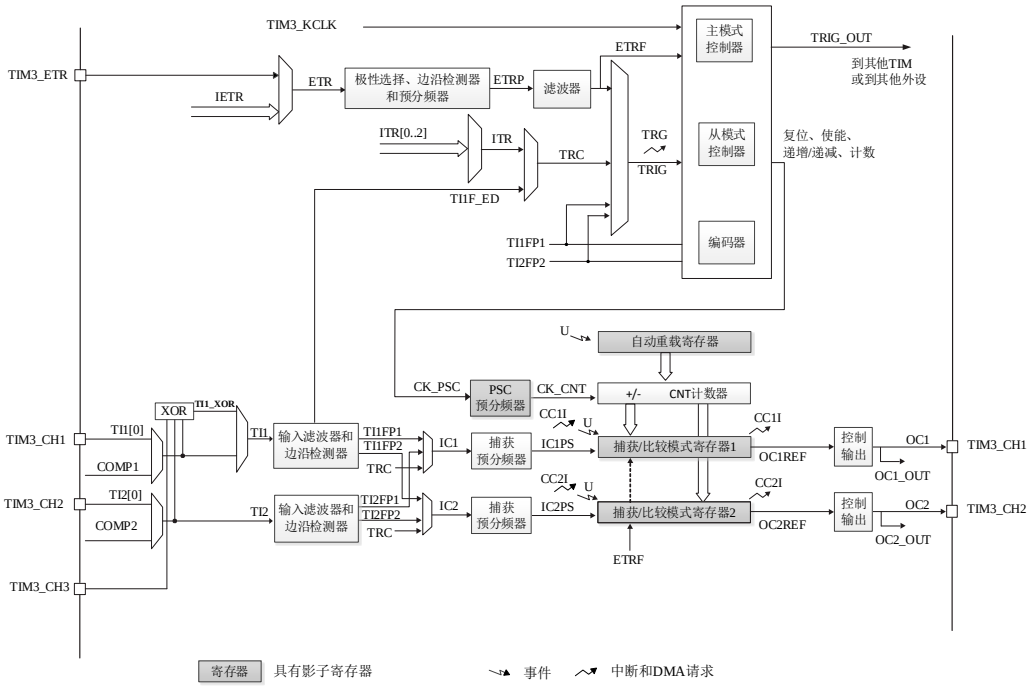
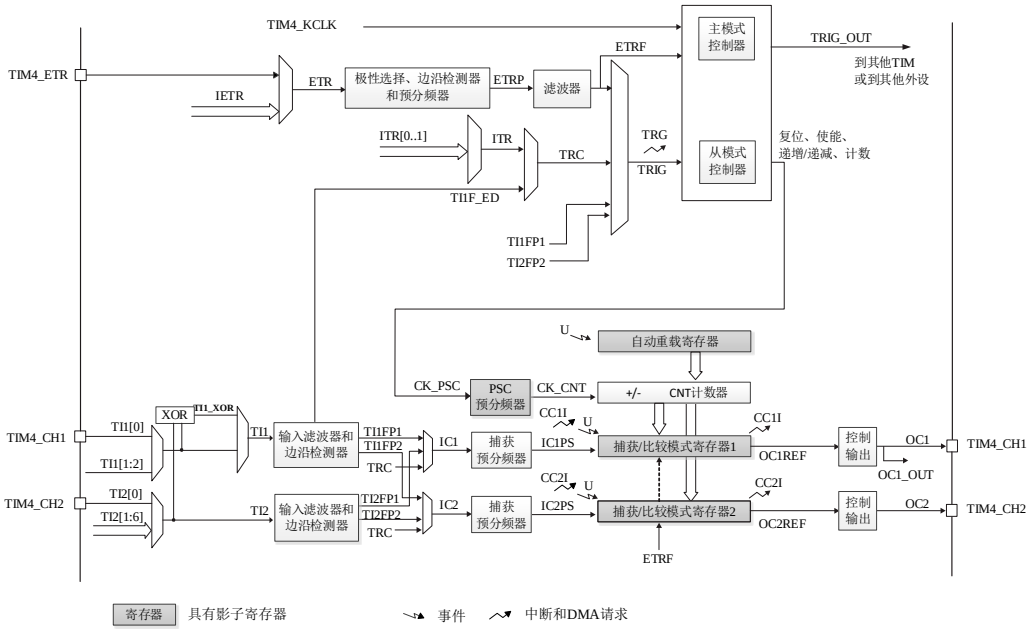



图 20-2 TIM4 框图



### 20.3.2 TIM3/4 引脚

表 20-1 TIM3 引脚

| 引脚名称 | 信号类型 | 说明 |
|---|---|---|
| TIM3_ETR | 输入 | 外部触发 ETR 引脚 |
| TIM3_CHx | 输入/输出 | 通道输入输出引脚 |

表 20-2 TIM4 引脚

| 引脚名称 | 信号类型 | 说明 |
|---|---|---|
| TIM4_ETR | 输入 | 外部触发 ETR 引脚 |
| TIM4_CHx | 输入/输出 | 通道输入输出引脚 |

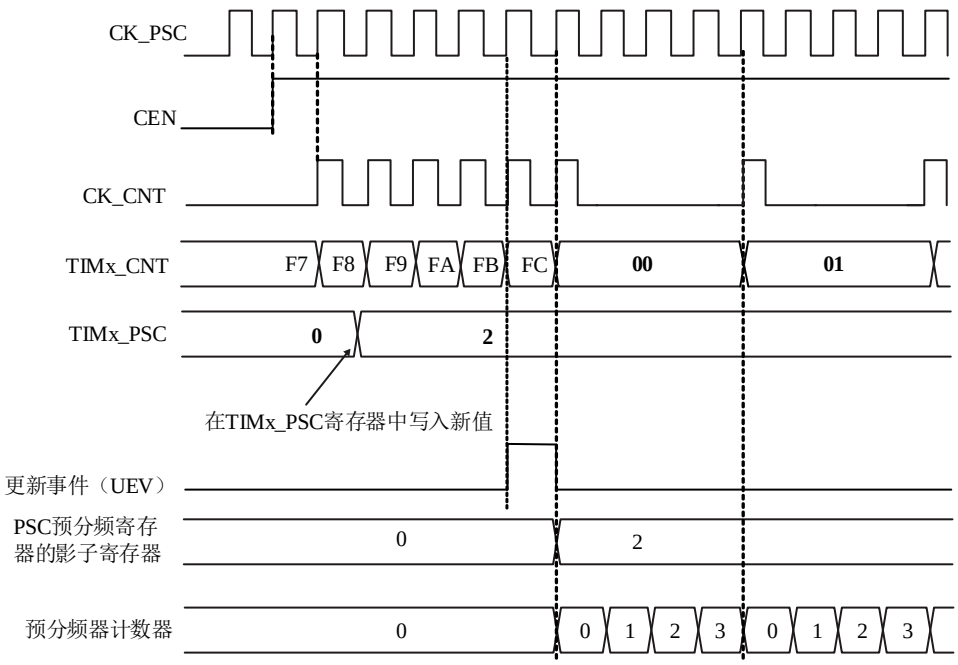
### 20.3.3 PSC 预分频器

PSC 预分频器对 CNT 计数器的输入时钟进行分频，分频系数介于 1 和 32768 之间。TIMx_PSC 寄存器中的 PSC[3:0]位域用于配置分频值，实际分频系数为 $2^{PSC[3:0]}$。

TIMx_PSC 寄存器具有缓冲功能（影子寄存器），每次写入的预分频值，会在下一次更新事件发生时生效。

下图举例显示了预分频器更新生效的时序。

图 20-3 预分频器分频系数由 1 变为 4 时的计数器时序图



### 20.3.4 计数器

TIM3 内置一个 16 位的计数器，TIM4 内置一个 32 位的计数器，可选择递增计数模式、递减计数模式或中心对齐计数模式。

计数器配置支持动态修改，在计数器运行时，TIMx_CNT 寄存器、TIMx_ARR 寄存器和 TIMx_PSC 寄存器均可进行读写操作。

自动重载值寄存器 TIM3_ARR 中的 ARR[15:0]位域和 TIM4_ARR 中的 ARR[31:0]位域可用于配置计数器的重载值，该寄存器具有缓冲功能（影子寄存器）。

如果 TIMx_CR1 寄存器中的 ARPE 位为 0 时，影子寄存器的更新不受更新事件的影响，TIMx_ARR 寄存器的值会立即更新。如果 ARPE 位为 1 时，当发生更新（UEV）事件，则 TIMx_ARR 寄存器中的值将更新到影子寄存器中。

如果将 TIMx_CR1 寄存器中的 UDIS 位置 1，可禁止更新事件，从而避免对影子寄存器的更新，但不影响计数器及预分频计数器均被初始化。当 UDIS 位为 0 时，允许更新事件生成，允许加载使能预装载功能的影子寄存器。

更新事件标志的请求源：

- 计数器上溢/下溢；
- 通过从模式控制器生成的更新事件（复位模式）。

将 UG 位置 1，会加载 PSC 及使能预装载功能的影子寄存器（UDIS 位为 0 时），计数器及预分频计数器均被初始化，但不会将 TIMx_SR 寄存器中的 UIF_UDR 或 UIF_OVR 置位。

**递增计数模式**

在递增计数模式下，计数器从 0 开始递增计数，每个 CK_CNT 周期自动加 1，当计数值等于 TIMx_ARR 寄存器的值时，将产生计数器上溢事件，TIMx_SR 寄存器的 UIF_OVR 标志将自动置 1，同时计数器将被清 0 并重新开始计数。

发生更新事件时，TIMx_SR 寄存器中的 UIF_OVR 位将自动置 1，并对以下寄存器进行更新：

- 预装载值（TIMx_ARR 寄存器）更新到自动重载影子寄存器中
- 预装载值（TIMx_PSC 寄存器）重新加载到 PSC 预分频器的影子寄存器中

以下各图，显示了递增计数模式下，计数器发生上溢更新事件时的时序，

TIMx_ARR 寄存器配置为 0x36：

图 20-4　递增计数模式下配置预分频器为 1 分频上溢事件时序图

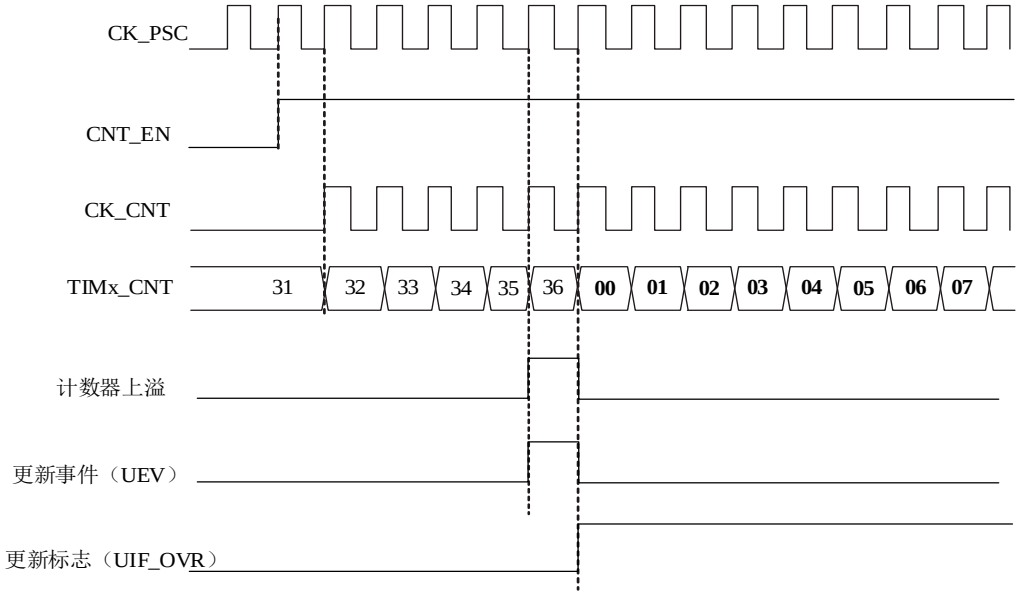



图 20-5　递增计数模式下配置预分频器为 4 分频上溢事件时序图

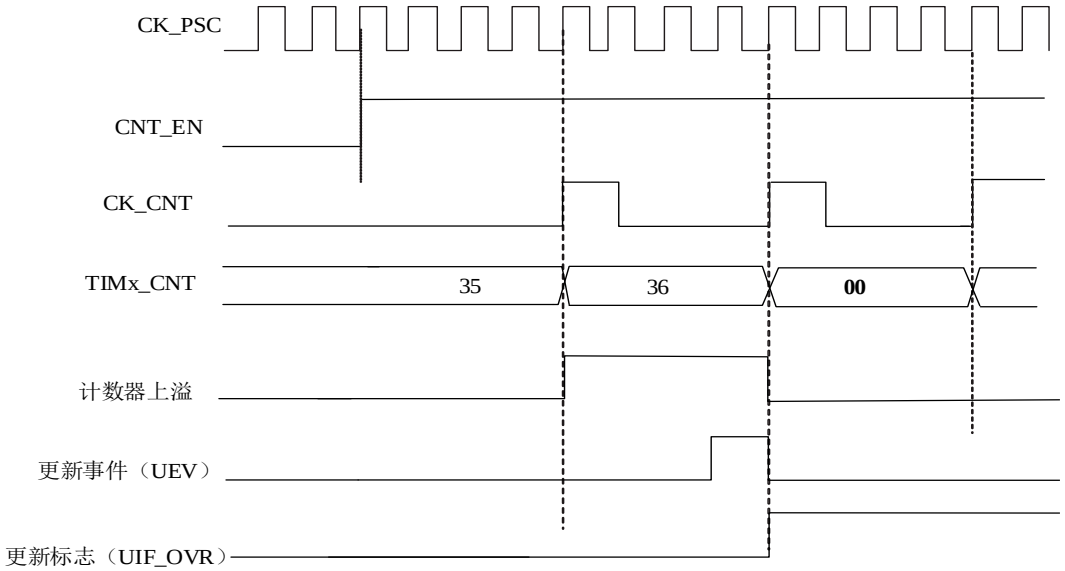


图 20-6 递增计数模式下计数器时序图，ARPE=0 时更新事件

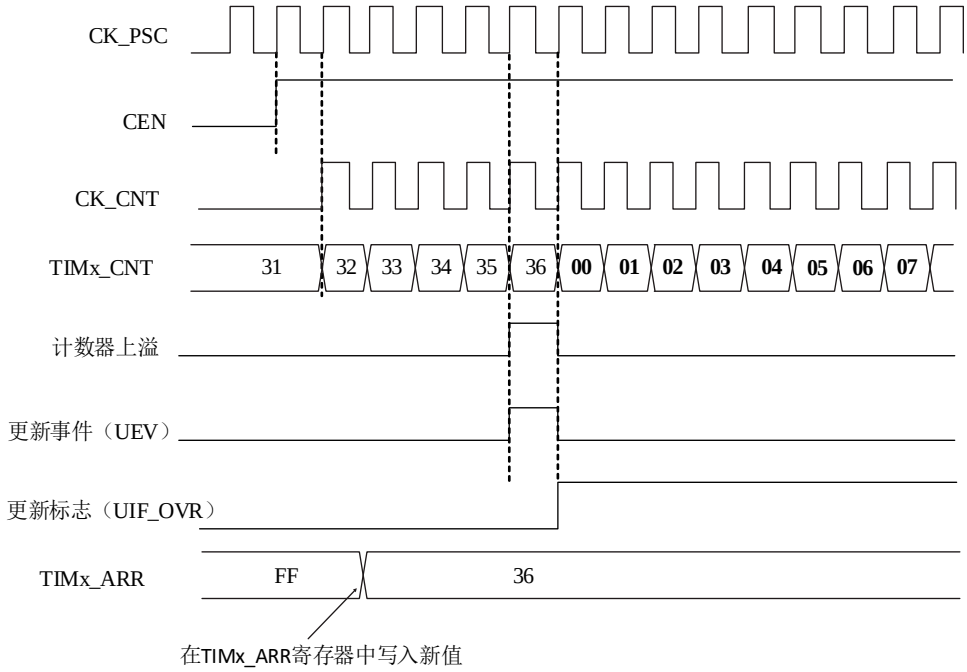
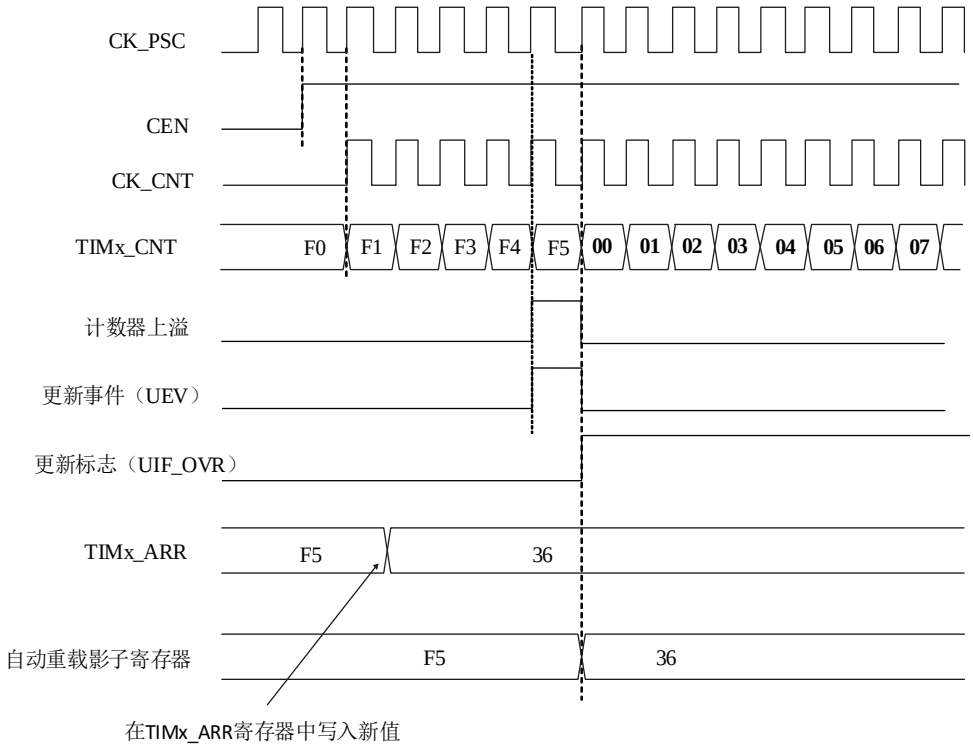



图 20-7 递增计数模式下计数器时序图，ARPE=1 时更新事件




## 递减计数模式

在递减计数模式下，计数器从 TIMx_ARR 寄存器的值开始递减计数，每个 CK_CNT 周期自动减 1，当计数器等于 0 时，将产生计数器下溢事件，TIMx_SR 寄存器的 UIF_UDR 标志将自动置 1，同时计数器加载自动重载值，并重新开始

递减计数。

发生更新事件时，TIMx_SR 寄存器中的 UIF_UDR 位将自动置 1，并对以下寄存器进行更新：

- 预装载值（TIMx_PSC 寄存器）重新加载到 PSC 预分频器的影子寄存器中
- 预装载值（TIMx_ARR 寄存器）更新到自动重载影子寄存器中。TIMx_ARR 寄存器的值在计数器重载之前被更新，因此下一个周期才是预期的值

以下各图，显示了计数器发生下溢更新事件时的时序，TIMx_ARR 寄存器配置为 0x36：

图 20-8 递减计数模式下配置预分频器为 1 分频下溢事件时序图

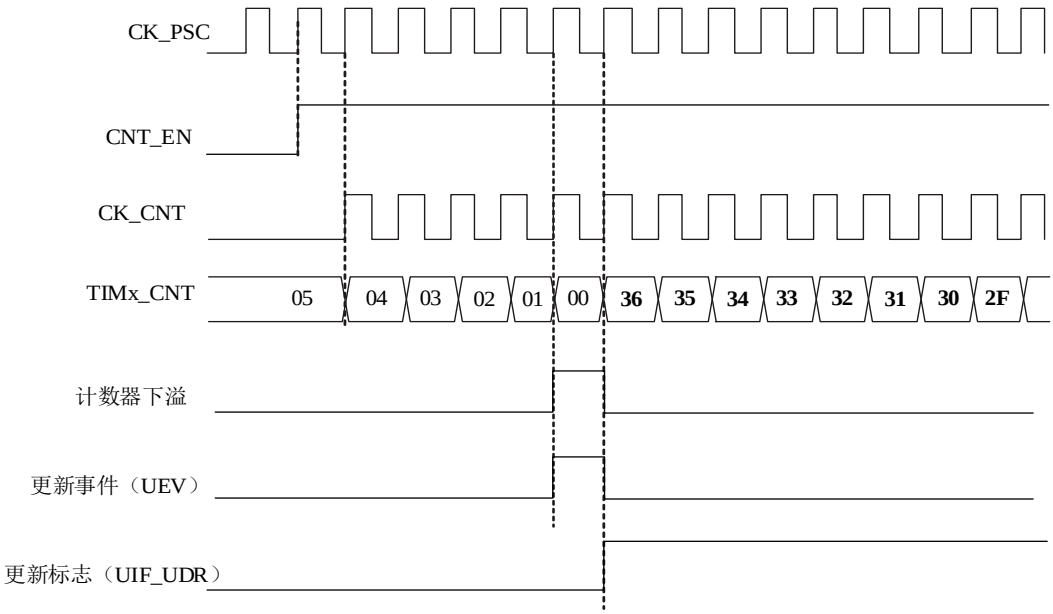


图 20-9 递减计数模式下配置预分频器为 4 分频下溢事件时序图

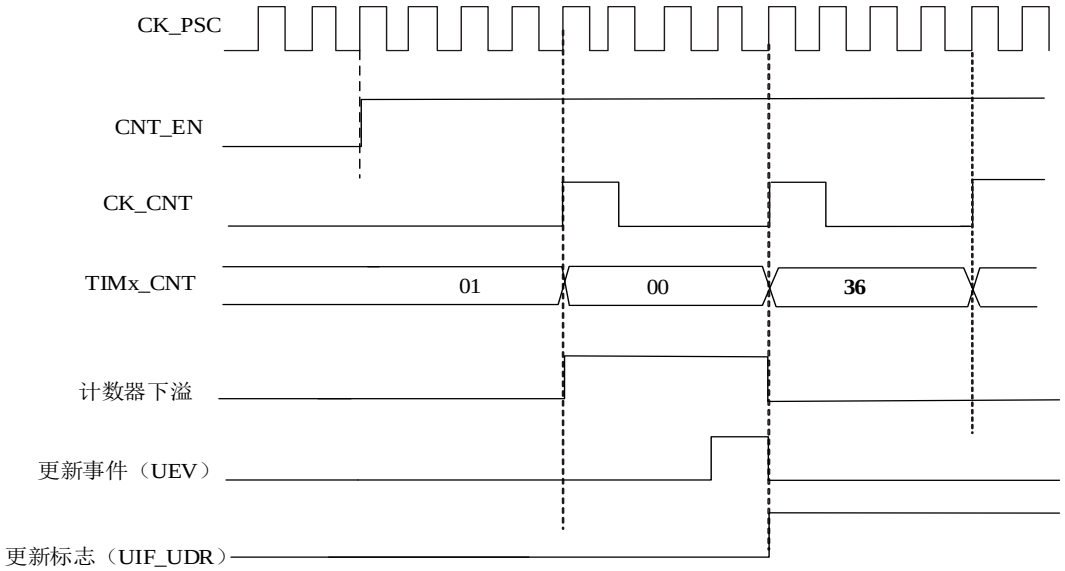
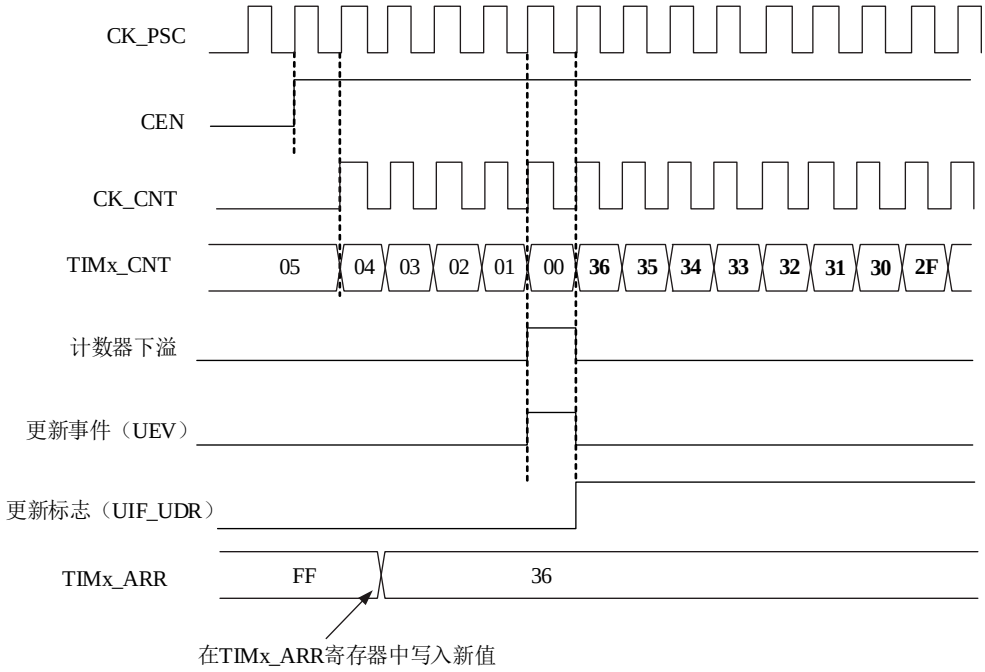



图 20-10 递减计数模式下计数器时序图，ARPE=0 时更新事件




### 中心对齐模式（递增/递减计数）

在中心对齐模式下，计数过程由以下阶段组成：

1） 计数器从 TIMx_CNT 的值开始递增计数到 TIMx_ARR–1，生成计数器上溢事件；

2） 从 TIMx_ARR 的值开始递减计数到 1，生成计数器下溢事件；

3） 从 0 开始递增计数。

当 TIMx_CR1 寄存器中的 DIR 位为 1 时，启动计数后，计数器从当前 TIMx_CNT 的值开始递减计数。

启动计数后，无论当前是递增还是递减计数，当 UG 位置 1 时，TIMx_CNT 寄存器的值和 TIMx_CR1 寄存器中的 DIR 均变为 0。

当 TIMx_CR1 寄存器中的 CMS[1:0]位域不为 00 时，计数器将采用中心对齐模式，在此模式下，TIMx_CR1 寄存器中的 DIR 位不能写入，读取 DIR 位可获取当前计数器的计数方向。中心对齐模式共有以下几种计数模式：

- 中心对齐模式 1（CMS[1:0]位域为 01）：计数器交替进行递增计数和递减计数。上溢或下溢均触发更新事件；仅在递减计数过程中，当发生比较匹配时，TIMx_SR 寄存器中的 CCxIF 位会置 1；
- 中心对齐模式 2（CMS[1:0]位域为 10）：计数器交替进行递增计数和递减计数。上溢或下溢均触发更新事件；仅在递增计数过程中，当发生比较匹配时，TIMx_SR 寄存器中的 CCxIF 位会置 1；
- 中心对齐模式 3（CMS[1:0]位域为 11）：计数器交替进行递增计数和递减计数。上溢或下溢均触发更新事件；在递增计数或递减计数过程中，当发生比较匹配时，TIMx_SR 寄存器中的 CCxIF 位会置 1。

发生更新事件时，TIMx_SR 寄存器中的 UIF_UDR/UIF_OVR 位将自动置 1，并对以下寄存器进行更新：

- 预装载值（TIMx_PSC 寄存器）重新加载到 PSC 预分频器的影子寄存器中；
- 预装载值（TIMx_ARR 寄存器）更新到自动重载影子寄存器中。

以下各图，显示了中心对齐模式下，计数器发生上/下溢更新事件时的时序，TIMx_ARR 寄存器配置为 0x06：

图 20-11 中心对齐模式下配置预分频器为 1 分频上/下溢事件时序图

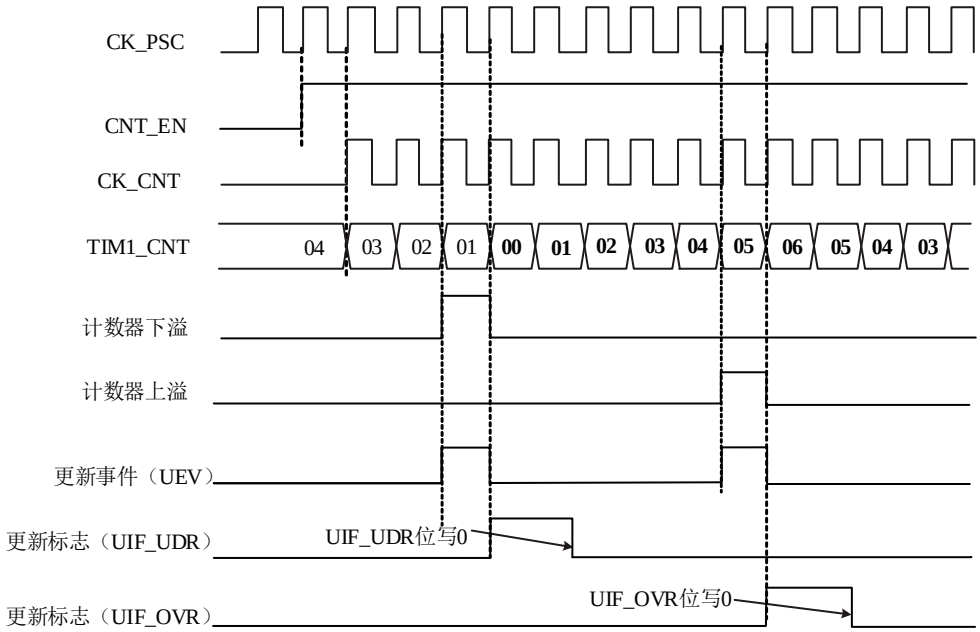



图 20-12 中心对齐模式下配置预分频器为 2 分频下溢事件时序图

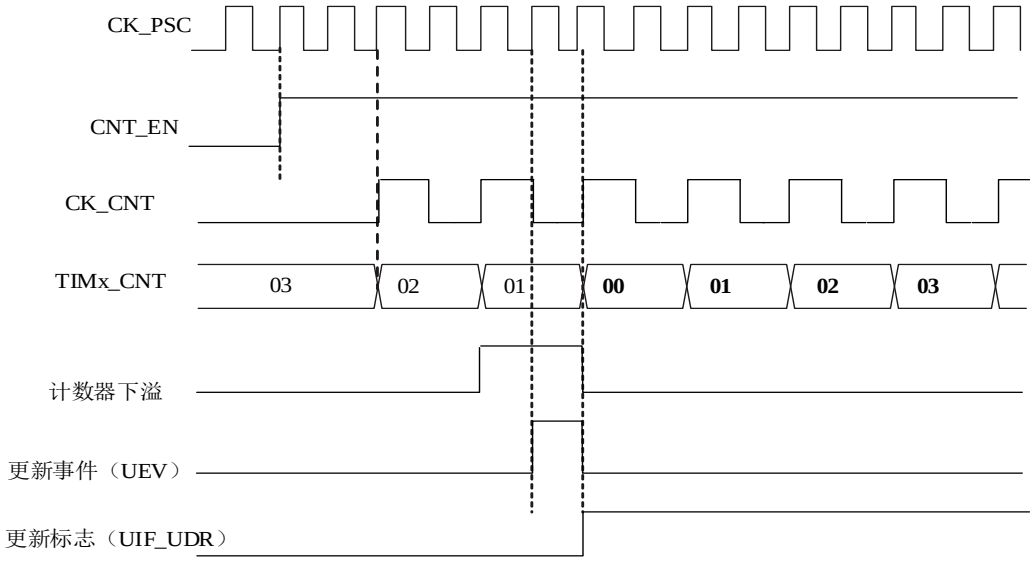


图 20-13 中心对齐模式下，产生下溢事件，ARPE=1 时的时序图

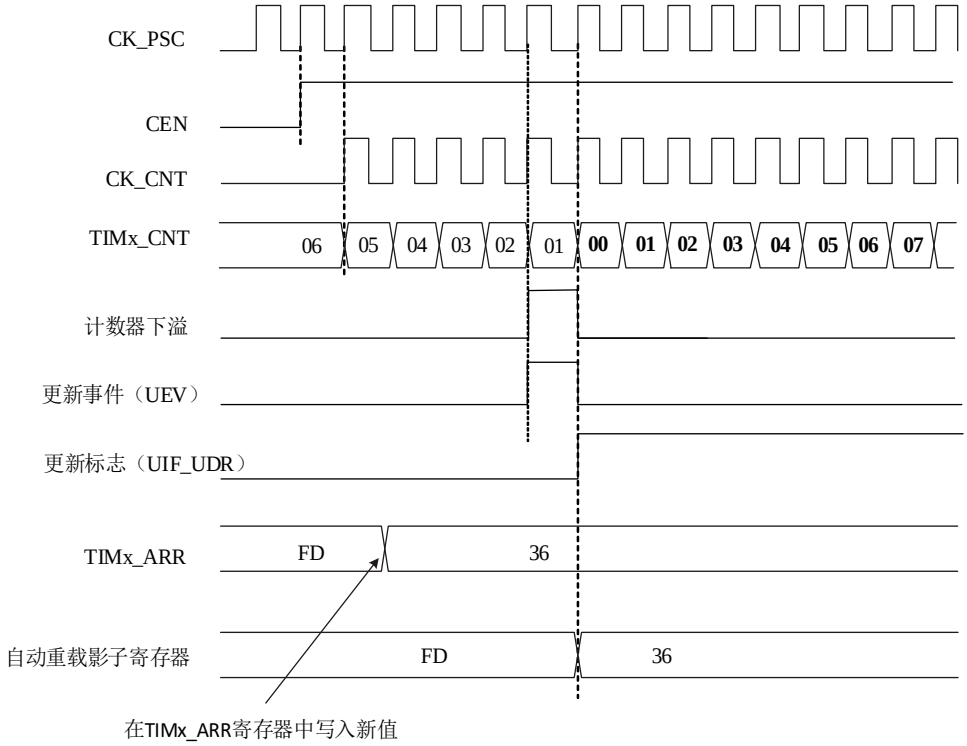
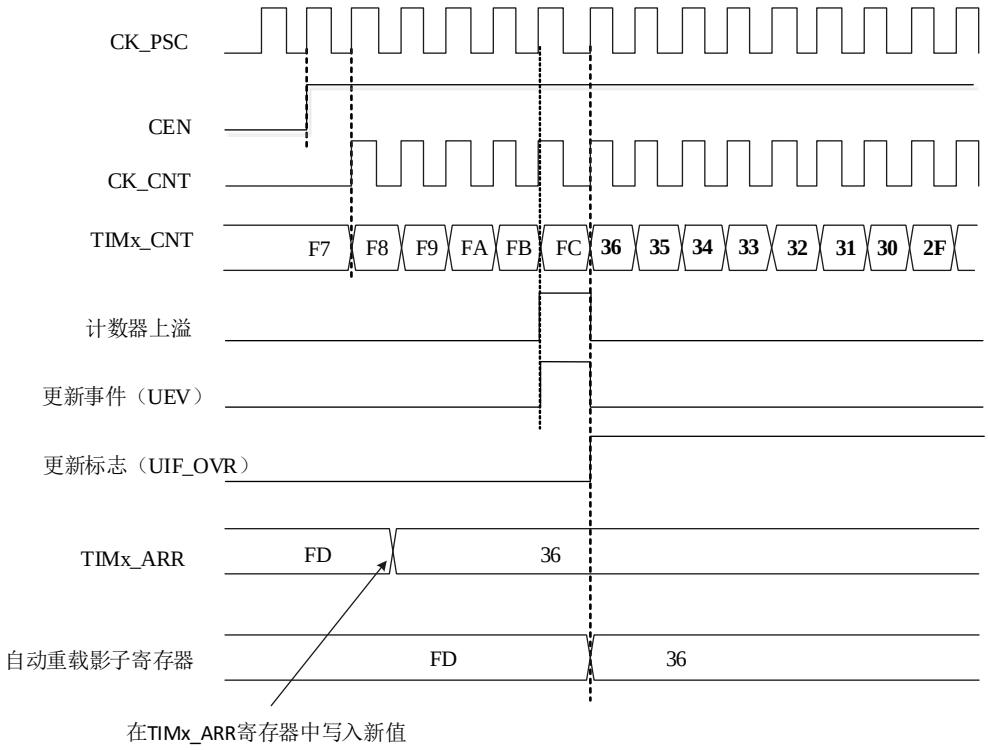



图 20-14 中心对齐模式下，产生上溢事件，ARPE=1 时的时序图




建议使用中心对齐模式时，在启动计数器前置位 TIMx_EVTG 寄存器中的 UG 位，产生一个更新事件，并且不要在计数器计数期间修改计数器的值。

### 20.3.5 外部触发输入

TIMx 具有外部触发功能，其触发源可通过 TIMx_AF1 寄存器中的 ETR_SEL[2:0] 位域进行配置，如下图所示。

图 20-15 TIM3 ETR 输入源选择

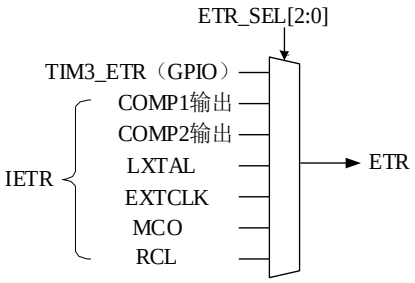
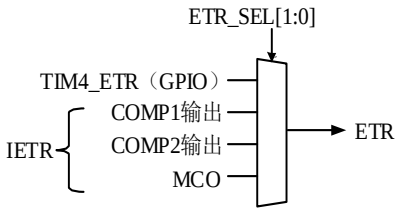


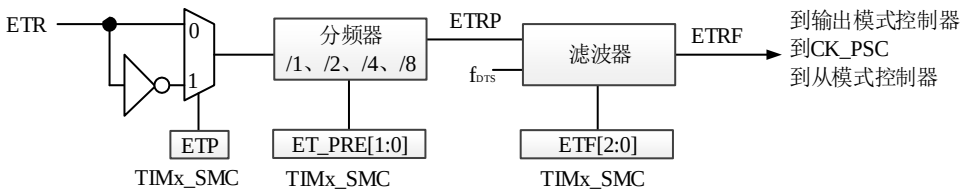

图 20-16 TIM4 ETR 输入源选择




如下图所示，ETR 的输入极性可通过 TIMx_SMC 寄存器中的 ETP 位进行配置，经过预分频器和数字滤波器，产生 ETRF 信号，该信号可用作：

- 时钟模式 2（请参见*计数时钟选择*）；
- 从模式的触发输入（请参见*定时器同步从模式*）。
- 发生外部事件时用于清除 OCxREF 信号（请参见*清除 OCxREF 信号*）

图 20-17 外部触发输入模块




### 20.3.6 计数时钟选择

计数器的计数时钟可由下列信号源提供：

- 内部时钟源：TIMx 外设时钟（TIMx_KCLK）；
- 时钟模式 1：TRIG 触发输入作为计数时钟；

- 时钟模式 2：ETR 触发输入作为计数时钟。

以上时钟源可通过配置 TIMx_PSC 寄存器的值进行预分频后，作为计数时钟（CK_CNT）。

**内部时钟源**

当配置 TIMx_SMC 寄存器中的 SM_SEL[2:0]位域为 0xx 时，计数器的计数时钟源为 TIMx_KCLK，来自 RCC，即 TIMx_PCLK。

**时钟模式 1**

当配置 TIMx_SMC 寄存器中的 SM_SEL[2:0]位域为 111 时，计数器的计数时钟源为 TRIG 触发输入信号，可通过配置 TIMx_CCEN 寄存器中的 CCxP 和 CCxNP 位来选择上升沿或下降沿时计数。

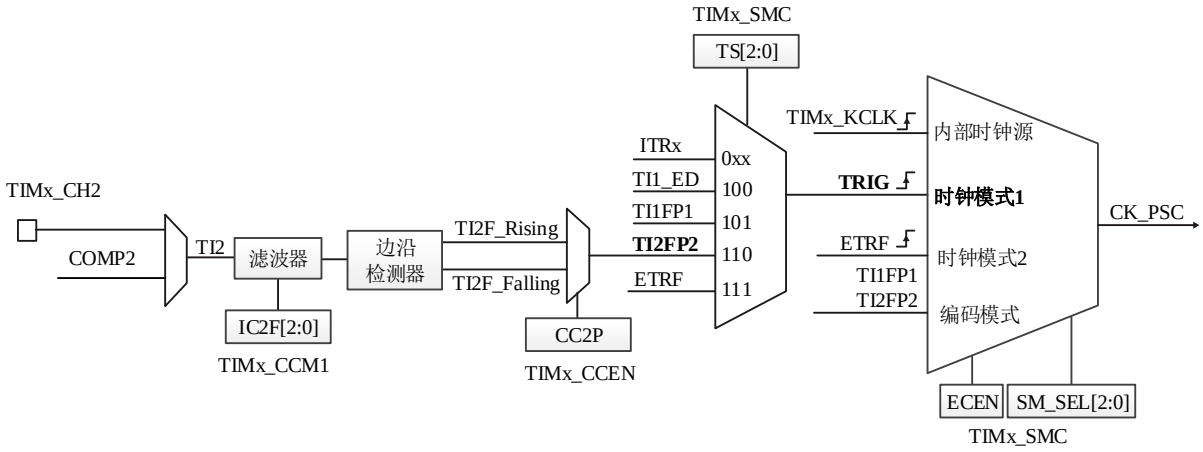
通过配置 TIMx_SMC 寄存器中的 TS[2:0]位域可选择不同的 TRIG 信号：

- ITR0
- ITR1
- ITR2（仅 TIM3）
- TI1F_ED
- TI1FP1
- TI2FP2
- ETRF

ETRF 和 TIx 信号与 TIMx_KCLK 为异步信号，故 ETRF 和 TIx 的最大频率为 TIMx_KCLK 频率的 1/2。

当选择 TI1F_ED 作为 TRIG 信号时，其信号最大频率为 TIMx_KCLK 频率的 1/4。

图 20-18 选择 TI2 作为时钟模式 1 触发输入的连接示例（上升沿计数）

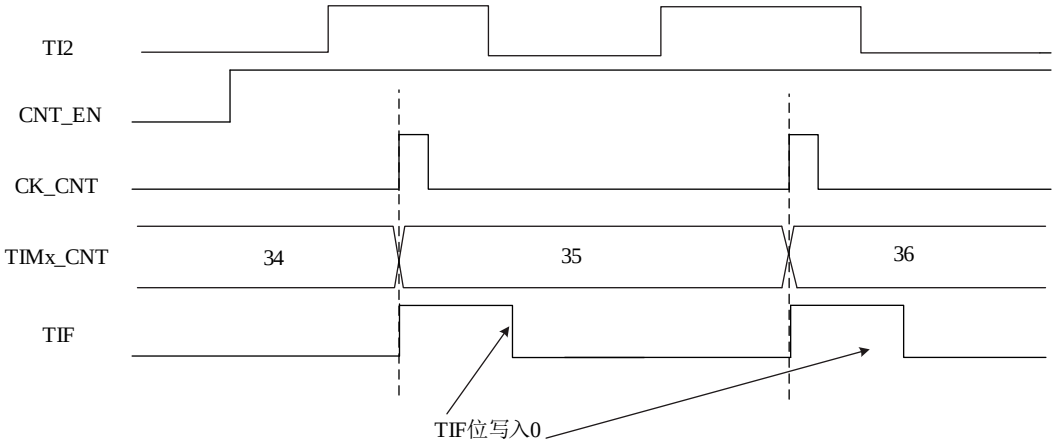



例如，要使计数器在 TI2 输入的上升沿时计数，其步骤如下：

1) 选择 TI2x 触发源：配置 TIMx_TISEL 寄存器中的 TI2SEL 位域为 0；
2) 配置通道 2 为输入模式：配置 TIMx_CCM1 寄存器中的 CC2S[1:0]位域为 01；
3) 设置输入滤波器：配置 TIMx_CCM1 寄存器中 IC2F[2:0]位域（如果不需要滤波器，请保持 IC2F[2:0]为 000）；
4) 极性选择（上升沿有效）：配置 TIMx_CCEN 寄存器中的 CC2NP/CC2P 位为 00；
5) 选择时钟模式 1：配置 TIMx_SMC 寄存器中的 SM_SEL[2:0]位域为 111；
6) 选择 TI2FP2 为触发输入源：配置 TIMx_SMC 寄存器中的 TS[2:0]位域为 110；
7) 使能计数器：配置 TIMx_CR1 寄存器中的 CEN 位为 1；

当 TI2 出现上升沿时，计数器便会计数一次，TIMx_SR 寄存器中的 TIF 被自动置 1。由于受信号同步和滤波器的影响，TI2 的上升沿与计数器时钟之间存在延迟。

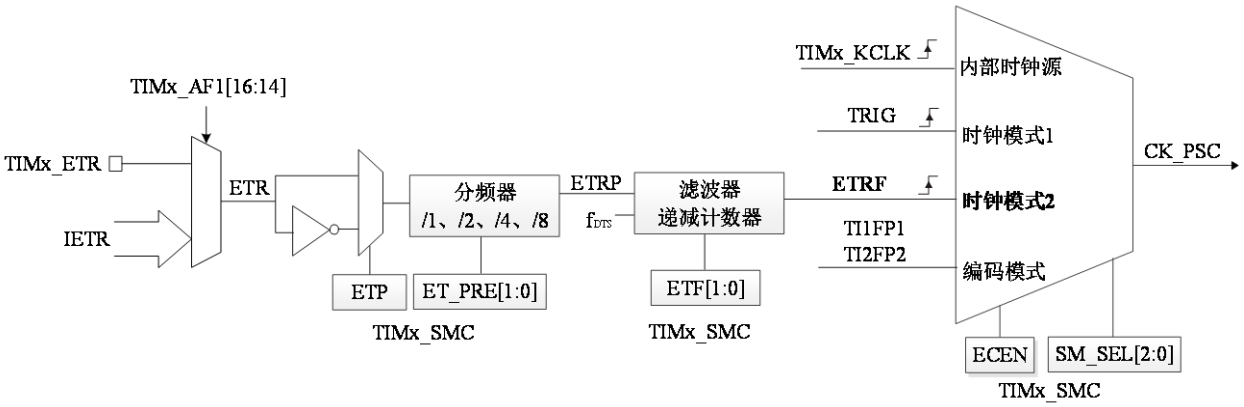
图 20-19 选择时钟模式 1 时的时序图




### 时钟模式 2

当配置 TIMx_SMC 寄存器中的 ECEN 位置 1，计数器可在 ETR 触发输入的上升沿或下降沿时计数。

图 20-20 时钟模式 2 触发输入模块

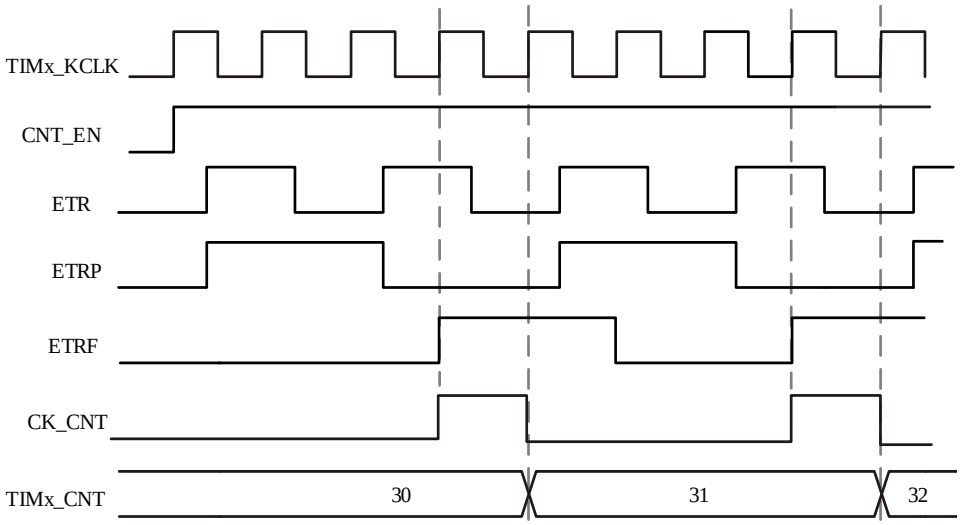



经 ETR 分频后的信号为 ETRP，ETRP 与 TIMx_KCLK 为异步信号，其最大频率为 TIMx_KCLK 频率的 1/2。

例如，递增计数器在 ETR 的每 2 个上升沿时计数一次，其步骤如下：

1) 选择 ETR 触发源：配置 TIMx_AF1 寄存器中的 ETR_SEL[2:0]位域为 000；
2) 设置输入滤波器：配置 TIMx_SMC 寄存器中 ETF[2:0]位域（如果不需要滤波器，请保持 ETF[2:0]为 000）；
3) 设置预分频器：配置 TIMx_SMC 寄存器中的 ET_PRE[1:0]位域为 01；
4) 选择触发信号上升沿有效：配置 TIMx_SMC 寄存器中的 ETP 位为 0；
5) 使能时钟模式 2：TIMx_SMC 寄存器中的 ECEN 位为 1；
6) 使能计数器：配置 TIMx_CR1 寄存器中的 CEN 位为 1。

图 20-21 选择时钟模式 2 下的时序图



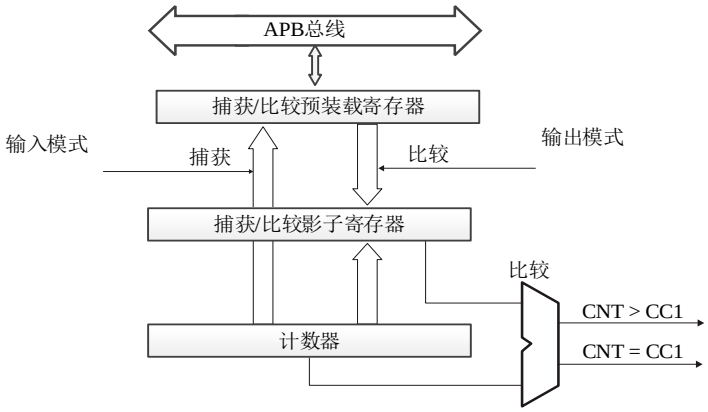

当时钟模式 1 和时钟模式 2 的信号源均为 ETRF，且时钟模式 1 和时钟模式 2 同时使能时，时钟模式 2 的优先级高。

### 20.3.7 捕获/比较通道

捕获/比较通道包含：

- 捕获/比较寄存器（包括一个影子寄存器）
- 输入捕获通道（数字滤波、多路复用和预分频器）
- 输出比较通道（比较器和输出控制）

图 20-22 捕获/比较通道



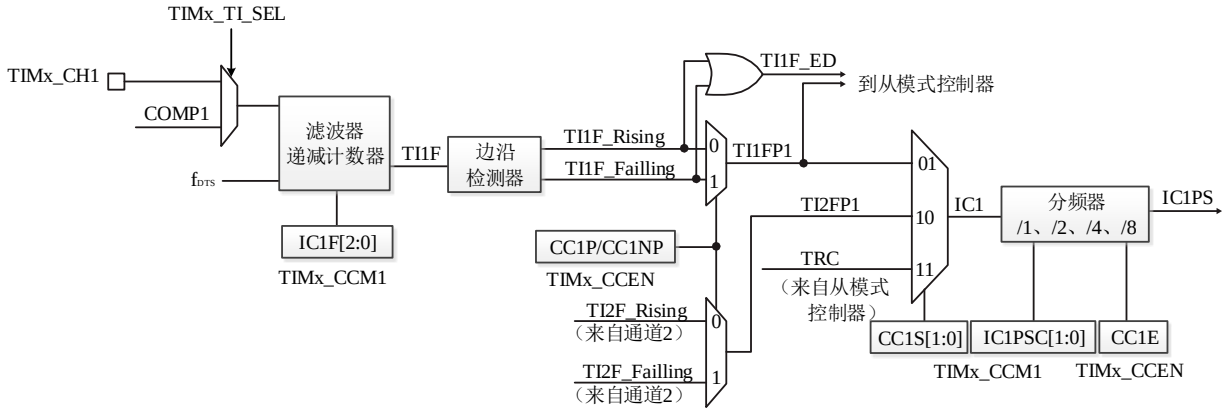

#### 输入捕获模式

TIx 的输入源可通过 TIMx_TISEL 寄存器的相关位域来配置。输入捕获源选择可通过 TIMx_CCMx 寄存器中的 CCxS[1:0]位域来配置。以通道 1 为例，有 TRC、TI1FP1、TI2FP1。

对 TIx 输入经采样滤波后产生 TIxF 信号，随后通过一个带极性选择的边沿检测器后产生 ICx 信号。该信号通过捕获预分频器进入捕获/比较寄存器。

各通道的具体组成，详见下列各图：

图 20-23 捕获/比较通道的输入连接图（通道 1）



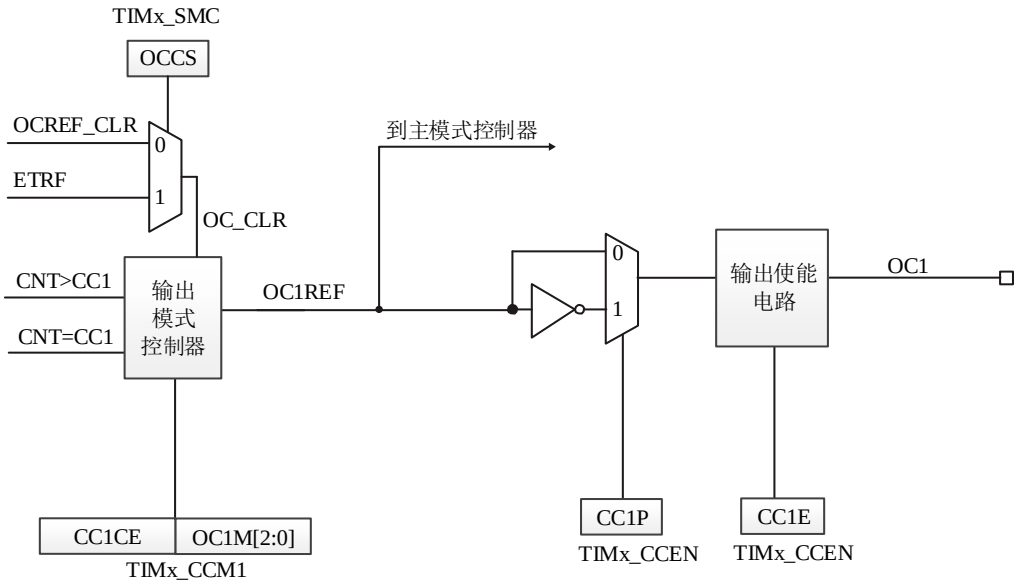

### 输出比较模式

输出通道 1 到通道 2 可输出到芯片外部。

输出比较模式分为：

- 比较输出模式，详见*比较输出模式*；
- 强制输出模式，详见*强制输出模式*；
- PWM 模式，详见 *PWM 模式*；
- 单脉冲模式，详见*单脉冲模式*。

图 20-24 捕获/比较通道的输出连接图（通道 1）




### 20.3.8 输入捕获模式

在输入捕获模式下，有如下方式生成捕获事件：

- 检测到 ICx 信号的有效触发沿；
- 将 TIMx_EVTG 寄存器中相应的 CCxG 位置 1。

当产生输入捕获事件，此时会有如下行为：

- TIMx_CCx 寄存器会捕获当前计数器的值；
- TIMx_SR 寄存器中的 CCxIF 位置 1；
- 如果此时 TIMx_DIER 寄存器中的 CCxIE 位为 1，则触发中断；
- 如果此时 TIMx_DIER 寄存器中的 CCx_DMAEN 位置 1，则生成 DMA 请求。

CCxIF 位为 1 时，再次检测到捕获事件，将产生捕获溢出事件，寄存器 CCxOF 位会自动置 1，同时 TIMx_CCx 寄存器会重新捕获当前计数器的值。

当 CCxS[1:0]位域的值不为 00（输入捕获模式）时，TIMx_CCx 寄存器将处于只读状态。

CCxIF 清 0 的方式有：

- 向 CCxIF 写入 0；
- 读取 TIMx_CCx 寄存器中的已捕获的数据。

例如：在 TI1 输入信号上检测到上升沿时将计数器的值捕获到 TIMx_CC1 寄存器中。具体配置步骤如下：

1) 将 TI1 映射到 IC1：配置 TIMx_CCM1 寄存器中的 CC1S[1:0]位域为 01；
2) 设置输入滤波器：配置 TIMx_CCM1 寄存器中的 IC1F[2:0]位域为 011；
3) 选择 TI1 为上升沿有效：配置 TIMx_CCEN 寄存器中的 CC1P 位为 0；
4) 使能捕获：配置 TIMx_CCEN 寄存器中的 CC1E 位为 1。

### 20.3.9 PWM 输入模式

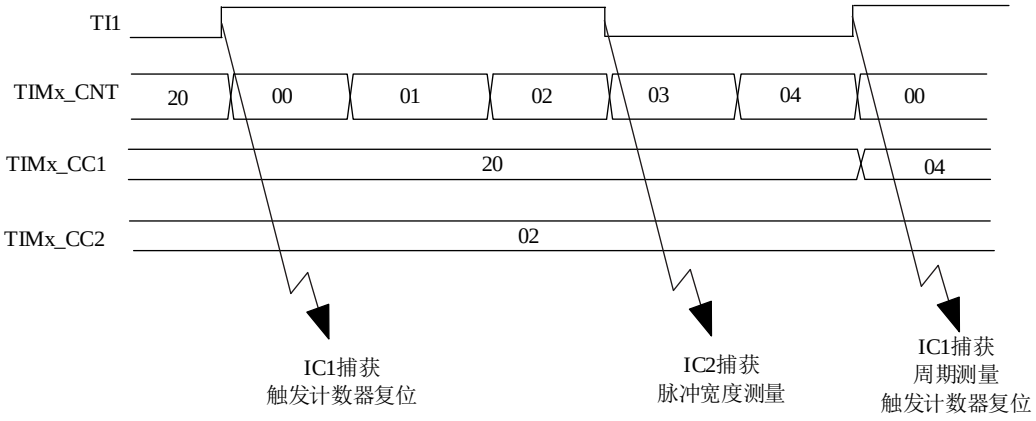
此模式是输入捕获模式的一种特例，但和输入捕获相比，有以下几点区别：

- 一个 TIx 输入信号被映射到两个 ICx 信号上；
- 这两个 ICx 信号为边沿有效，但极性相反；
- 其中一个 TIxFP 信号用作触发输入信号，而从模式控制器配置为复位模式，复位模式详见：*定时器同步从模式*。

例如，测量 TI1 上的 PWM 周期和占空比，其配置步骤如下：

1) 选择 TI1FP1 的有效极性：配置 TIMx_CCEN 寄存器中的 CC1NP/CC1P 位为 00（上升沿有效）；
2) 将 TI1FP1 映射到 IC1 上：配置 TIMx_CCM1 寄存器中的 CC1S[1:0]位域为 01；
3) 选择 TI1FP2 的有效极性：配置 TIMx_CCEN 寄存器中的 CC2NP/CC2P 位为 01（下降沿有效）；
4) 将 TI1FP2 映射到 IC2 上：配置 TIMx_CCM1 寄存器中的 CC2S[1:0]位域为 10；
5) 选择从模式控制器的触发输入：配置 TIMx_SMC 寄存器中的 TS[2:0]位域为 101（选择 TI1FP1）；
6) 选择从模式控制器为复位模式：配置 TIMx_SMC 寄存器中的 SM_SEL[2:0]位域为 100；
7) 使能捕获：配置 TIMx_CCEN 寄存器中的 CC1E 位和 CC2E 位为 1。

图 20-25 PWM 输入模式的时序




### 20.3.10 比较输出模式

该模式包含：匹配输出有效电平、匹配输出无效电平和翻转模式。通过 TIMx_CCMx 寄存器中的 OCxPE 位，可使能或禁止 TIMx_CCx 寄存器的预装载功能。

当 TIMx_CCx 的影子寄存器与 TIMx_CNT 寄存器的值匹配时：

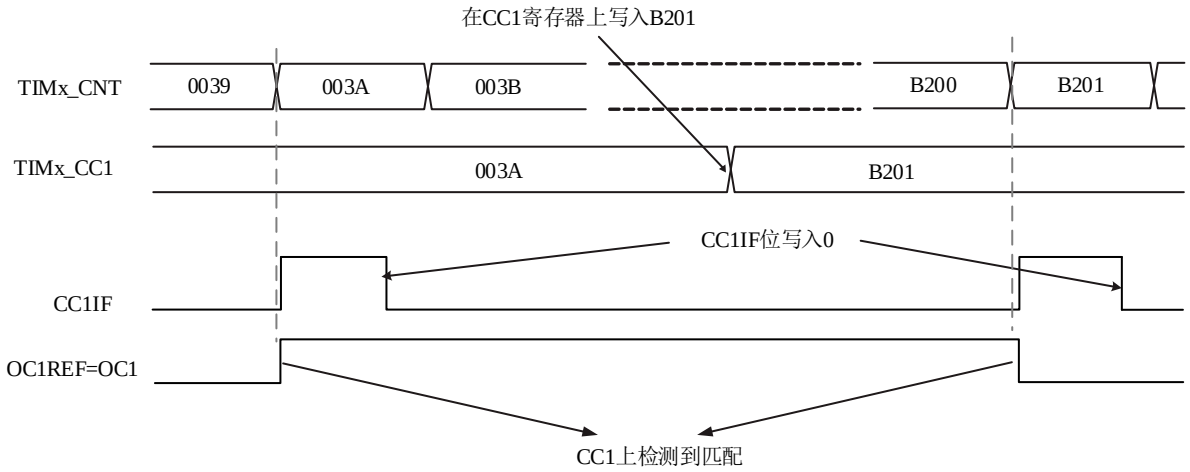
- TIMx_SR 寄存器中的 CCxIF 位置 1；
- 输出极性由 TIMx_CCEN 寄存器中的 CCxP 位决定；
- 如果此时 TIMx_DIER 寄存器中的 CCxIE 位为 1，则触发中断；
- 如果此时 TIMx_DIER 寄存器中的 CCx_DMAEN 位为 1，则产生 DMA 请求。

配置为比较输出模式时，其步骤如下：

1) 配置 TIMx_ARR 寄存器和 TIMx_CCx 寄存器；
2) 选择输出模式。例如：
   - 配置 TIMx_CCMx 寄存器中的 OCxM[2:0]位域为 011（当 TIMx_CNT 寄存器的值与 TIMx_CCx 寄存器的值匹配时，OCx 输出发生翻转）；
   - 禁止预装载寄存器：配置 TIMx_CCMx 寄存器中的 OCxPE 位为 0；
   - 选择高电平有效：配置 TIMx_CCEN 寄存器中的 CCxP 为 0；
   - 使能输出：配置 TIMx_CCEN 寄存器中的 CCxE 为 1；
3) 使能计数器：配置 TIMx_CR1 寄存器中的 CEN 位为 1。

当未使能预装载寄存器（TIMx_CCMx 寄存器中的 OCxPE 位为 0）时，可实时更改 TIMx_CCx 寄存器的值，以控制输出波形。

图 20-26 比较输出模式，翻转 OC1（OCxPE 位为 0）




### 20.3.11 强制输出模式

在强制输出模式下，可配置输出信号为有效或无效电平，此时输出信号不受 TIMx_CCx 影子寄存器和计数器之间的比较结果影响。

- 配置 TIMx_CCMx 寄存器中的 OCxM[2:0]位域为 101，可将输出信号（OCxREF）强置为高电平（OCxREF 始终为高电平有效）；
- 配置 TIMx_CCMx 寄存器中的 OCxM[2:0]位域为 100，可将输出信号（OCxREF）强置为低电平。

OCx 输出信号的极性由 TIMx_CCEN 寄存器中的 CCxP 位决定。

该模式下，TIMx_CCx 影子寄存器与计数器之间仍在进行比较，当产生匹配时，相应标志会被置位。

### 20.3.12 PWM 模式

PWM 模式生成的信号，其频率由 TIMx_ARR 寄存器的值决定，其占空比由 TIMx_CCx 寄存器的值决定。

每个输出通道都可以独立选择 PWM 模式输出：

- 通过 TIMx_CCMx 寄存器中的 OCxM[2:0]位域来配置。
  - PWM 模式 1（OCxM[2:0]位域为 110）：在递增计数模式下，当 TIMx_CNT＜TIMx_CCx，通道输出为有效电平，否则为无效电平。在递减计数模式下，当 TIMx_CNT＞TIMx_CCx，通道输出为无效电平，否则为有效电平；

- PWM 模式 2（OCxM[2:0]位域为 111）：在递增计数模式下，当 TIMx_CNT＜TIMx_CCx，通道输出为无效电平，否则为有效电平。在递减计数模式下，当 TIMx_CNT＞TIMx_CCx，通道输出为有效电平，否则为无效电平；

- 配置 TIMx_CCMx 寄存器中的 OCxPE 位置 1 使能相应比较寄存器的预装载功能（TIMx_CCx）；
- 配置 TIMx_CR1 寄存器中的 ARPE 位置 1 使能自动重载寄存器的预装载功能（TIMx_ARR）。

当使能预装载功能时，在发生更新事件时，预装载寄存器的值才会加载到影子寄存器中，所以启动计数器前，建议将 TIMx_EVTG 寄存器中的 UG 位置 1 来初始化所有寄存器。

OCx 输出信号的极性由 TIMx_CCEN 寄存器中的 CCxP 位决定。

根据 TIMx_CR1 寄存器中 CMS[1:0]位域的值，选择 PWM 信号的对齐模式：

**PWM 边沿对齐模式**

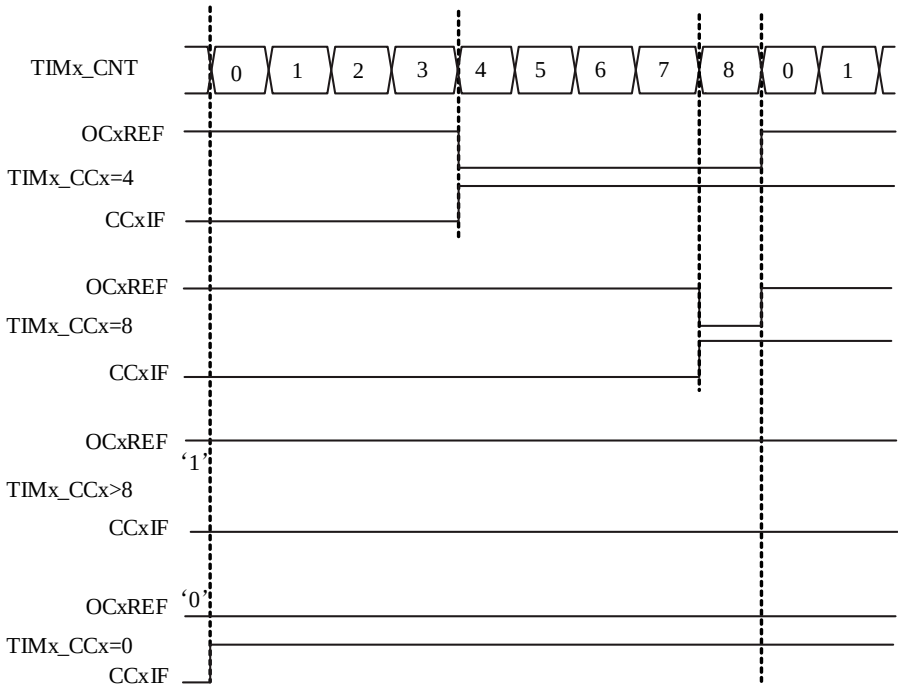
- 递增计数

  当 TIMx_CR1 寄存器中的 DIR 位为 0 时执行递增计数。请参见*递增计数模式*。

  下面是一个 PWM 模式 1 的例子。当 TIMx_CNT < TIMx_CCx，OCxREF 信号输出高电平，否则为低电平。如果 TIMx_CCx 寄存器的值大于 TIMx_ARR 寄存器的值，则 OCxREF 保持为 1。如果 TIMx_CCx 寄存器的值为 0，则 OCxREF 保持为 0。

  例如：TIMx_ARR 寄存器的值为 8 时 PWM 波形。

图 20-27 递增计数下的边沿对齐模式 PWM 波形（TIMx_ARR 寄存器为 8）



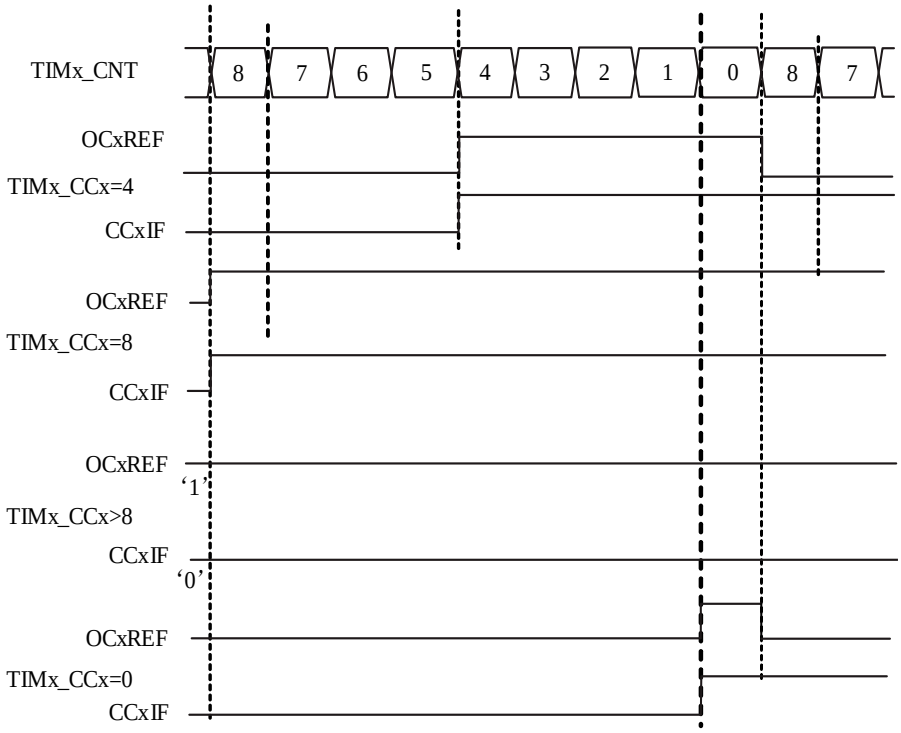

- 递减计数

  当 TIMx_CR1 寄存器中的 DIR 位为 1 时执行递减计数。请参考*递减计数模式*。

  在 PWM 模式 1 下，当 TIMx_CNT > TIMx_CCx，OCxREF 信号输出低电平，否则为高电平。如果 TIMx_CCx 中的值大于自动重载值（TIMx_ARR 中的值），则 OCxREF 保持为 1。此模式下不可能产生占空比为 0%的 PWM 波形。

  例如：TIMx_ARR 寄存器的值为 8 时，递减计数下的边沿对齐模式 PWM 波形。

图 20-28 递减计数下的边沿对齐模式 PWM 波形（TIMx_ARR 寄存器为 8）



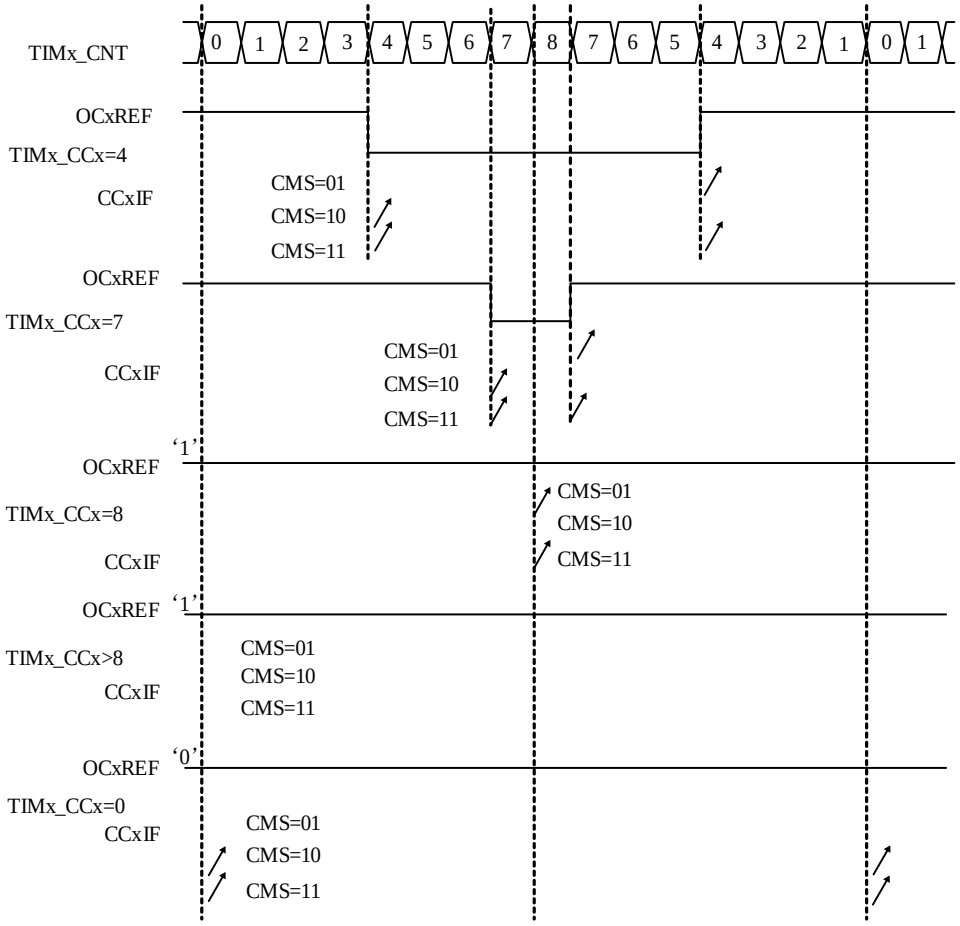

**PWM 中心对齐模式**

当 TIMx_CR1 寄存器中的 CMS[1:0]位域不为 00 时，为中心对齐模式。

如果配置 CMS[1:0]位域为中心对齐模式 3 时，在每个 PWM 周期内，计数器将产生上溢和下溢各一次，所以可更新 2 次 PWM 占空比。请参见：*中心对齐模式（递增/递减计数）*。

例如：当 PWM 模式为 PWM 模式 1，TIMx_ARR 寄存器为 8 时，中心对齐模式的 PWM 波形。

图 20-29 中心对齐模式 PWM 波形（TIMx_ARR 寄存器为 8）




### 20.3.13 清除 OCxREF 信号

对于给定的通道，OC_CLR 信号上的高电平可将 OCxREF 信号复位，OCxREF 信号将保持低电平，直到发生更新事件。

通过 TIMx_SMC 寄存器中的 OCCS 位来选择清除 OCxREF 的信号源：

- OCCS 位为 0：COMP1/COMP2 输出；
  - TIMx_CFG 寄存器中的 OCREF_CLR 位为 0：COMP1 输出连接到 OCREF_CLR 输入；
  - TIMx_CFG 寄存器中的 OCREF_CLR 位为 1：COMP2 输出连接到 OCREF_CLR 输入；
- OCCS 位为 1：ETRF 信号。

  TIMx_CCMx 寄存器中的 OCxCE 位为 1，ETRF 输入上检测到高电平时，OCxREF 立即清 0。

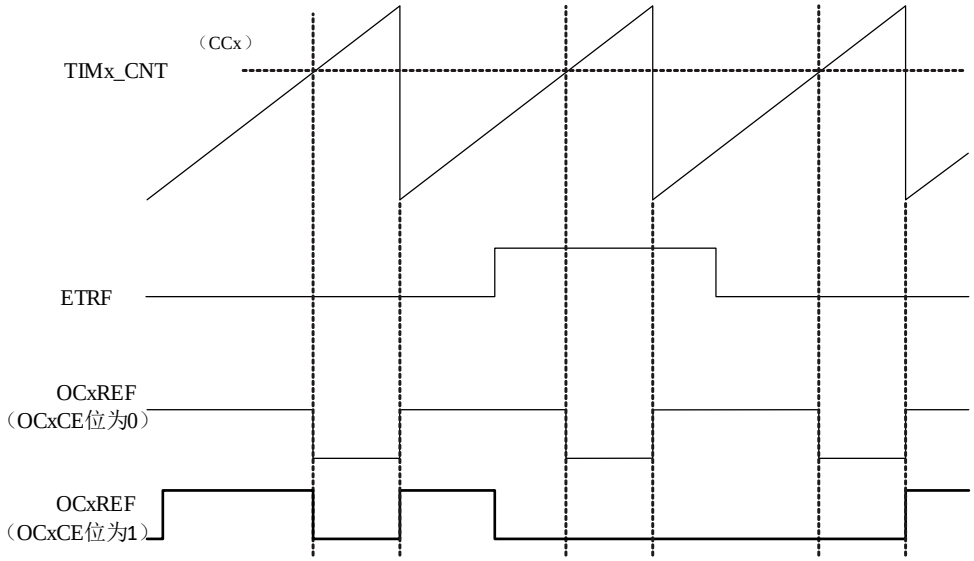
该功能只能用于比较输出模式和 PWM 模式，在强制模式下不起作用。例如，OCxREF 信号可以连接到比较器的输出，用于控制电流。

如果选择 ETRF 信号作为清除 OCxREF 信号时，ETR 必须配置如下：

1) 关闭外部触发预分频器：TIMx_SMC 寄存器中的 ET_PRE[1:0]位域写入 00；
2) 禁止外部时钟模式 2：TIMx_SMC 寄存器中的 ECEN 位清 0；
3) 外部触发极性和外部触发数字滤波可根据用户需要进行配置。

下图为 OCxCE 不同配置下，当 ETRF 输入变为高电平时 OCxREF 的信号波形。在本例中，TIMx 配置为 PWM 模式。

图 20-30 清除 TIMx 的 OCxREF




### 20.3.14 单脉冲模式

单脉冲模式（OPM）是比较输出模式的一个特例。将 TIMx_CR1 寄存器中的 OPM 位置 1，即可选择单脉冲模式。在此模式下，计数器接收到触发信号，在一段可编程的延时后产生一个脉宽可编程的脉冲。

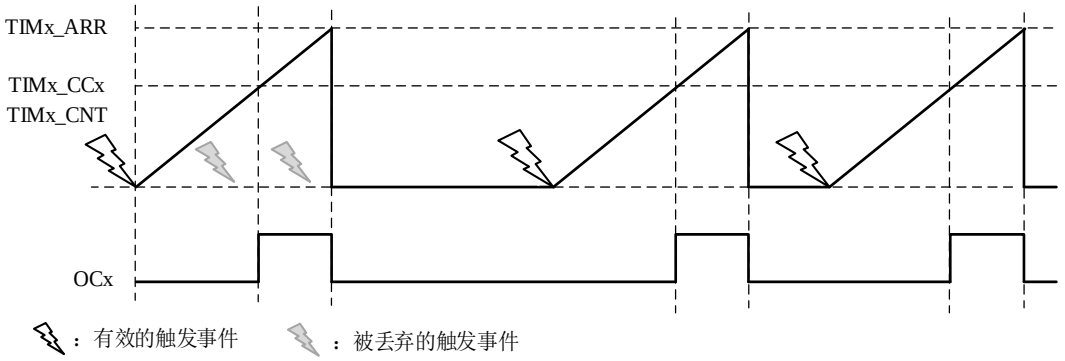
可以通过从模式控制器启动计数器。在比较输出模式或 PWM 模式下生成波形。当发生下一更新事件时，计数器将自动停止。

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：

- 递增计数时：CNT < CCx ≤ ARR（特别注意，0 < CCx）
- 递减计数时：CNT > CCx

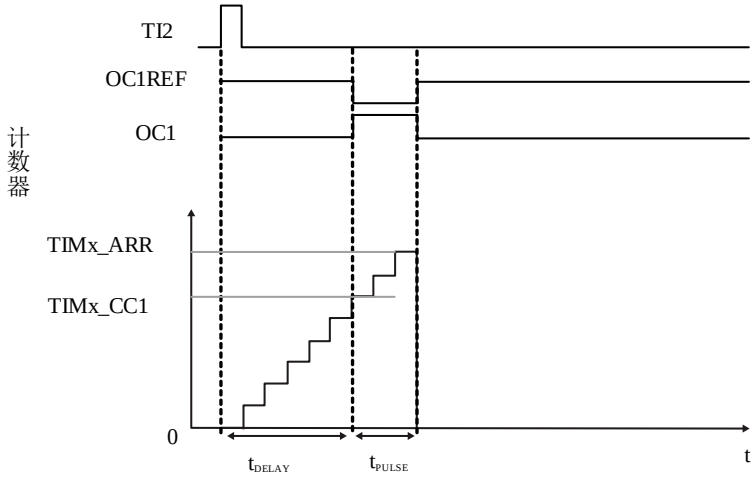
启动计数后，在一个计数周期内发生的任何触发事件均将被丢弃。如下图所示：

图 20-31 单脉冲输出波形




例如，在 TI2 输入引脚检测到上升沿时，经过 $t_{DELAY}$ 的延迟，在 OC1 上产生一个宽度为 $t_{PULSE}$ 的正脉冲。

图 20-32 单脉冲模式示例




使用 TI2FP2 作为触发信号：

1) 选择 TI2x 源：配置 TIMx_TISEL 寄存器中的 TI2_SEL 位域；

2) TI2FP2 映射到 TI2：TIMx_CCM1 寄存器中的 CC2S[2:0]位域写入 01；

3) 检测 TI2FP2 上升沿：TIMx_CCEN 寄存器中的 CC2P 位和 CC2NP 位清 0；

4) TI2FP2 配置为从模式控制器的触发模式（TRIG）：TIMx_SMC 寄存器中的 TS[2:0]写入 110；

5) TI2FP2 启动计数器：TIMx_SMC 寄存器中的 SM_SEL[2:0]位域写入 110（触发模式）。

单脉冲模式的脉冲宽度由写入的 TIMx_CC1 决定（考虑时钟频率和计数器预分频器）。

- $t_{DELAY}$ 由写入 TIMx_CC1 寄存器的值定义。
- $t_{PULSE}$ 由自动重载值与比较值之差（TIMx_ARR–TIMx_CC1）来定义。
- 若产生这样的波形：信号在发生比较匹配时从 0 变 1，在计数器达到自动重载值时由 1 变为 0：
  - OC1 输出极性：CC1P 位为 1
  - 使能 PWM 模式 1（TIMx_CCM1 寄存器中的 OC1M[2:0]写入 110）
  - 如果需要，可使能预装载功能（TIMx_CCM1 寄存器的 OC1PE 和 TIMx_CR1 寄存器的 ARPE 位置 1）
  - 在 TIMx_CC1 寄存器中写入比较值
  - 在 TIMx_ARR 寄存器中写入自动重载值
  - 将 UG 位置 1，产生一个更新事件，之后等待 TI2 上的外部触发事件启动计数器计数。

此例中，TIMx_CR1 寄存器中的 DIR 和 CMS 位应为 0。

**OCx 快速使能：**

在单脉冲模式下，输入的边沿检测会使能计数器（CEN 位自动置 1），之后在计数器值与比较值之间发生比较操作产生输出的转换。但此操作需要一定的时钟周期，因此它限制了可得到的最小延时 $t_{DELAY}$。

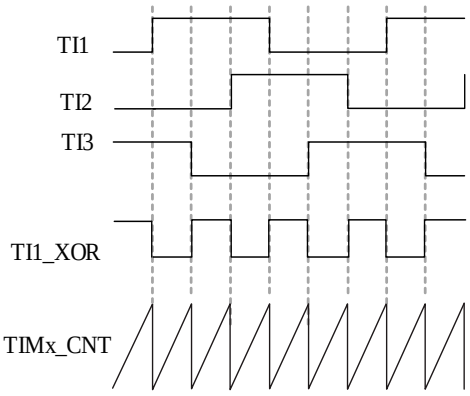
如果要输出延迟时间最短的波形，可以将 TIMx_CCMx 寄存器中的 OCxFE 位置 1。这样会强制 OCxREF（和 OCx）对触发信号做出响应，而不依赖于比较的结果。其输出的波形与比较匹配时的波形相同。仅在通道配置为 PWM1 或 PWM2 模式时，OCxFE 才会起作用。

### 20.3.15 TIMx 输入异或功能

将 TIMx_CR2 寄存器中的 TI1_XOR_SEL 位置 1，可将通道 1 的输入滤波器连接到异或门的输出端，TIM3 异或门的 3 个输入端为 TI1、TI2 和 TI3（TIM4 为 TI1、TI2 异或）。

异或输出用于定时器的输入功能，如触发或输入捕获。这样便于测量两个输入信号上边沿之间的间隔（如下图所示）。

图 20-33 测量 3 个信号边沿之间的时间间隔




TIM3 用于编码器的信号可以互联到 TIM4 作为通道输入，由 TIM4 进行捕获，从而获得编码器信号变化沿之间的时长信息，检测速度。

### 20.3.16 编码器接口模式

编码器接口模式仅 TIM3 支持。

编码器接口模式下，计数器仅在 0 到 TIMx_ARR 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、比较和触发输出功能继续正常工作。

编码器模式和时钟模式 1，2 不兼容，因此不能同时选择。当两者均有效时，编码器模式的优先级高。使能编码器模式时，PSC 预分频器配置无效，建议设置为零。

#### 正交编码模式

正交编码模式，有 5 种计数方式，通过 TIMx_SMC 寄存器中的 ENCMD[2:0]位域配置。

TI1 和 TI2 两个输入用于连接正交编码器。配置 TIMx_CCEN 寄存器的 CC1P 和 CC2P 位，选择 TI1 和 TI2 极性。如果需要，还可对输入滤波器进行配置（通过 TIMx_CCM1 寄存器中的 IC1F[2:0]位域和 IC2F[2:0]位域来配置）。如果使能计数器（TIMx_CR1 寄存器中的 CEN 位置 1），则计数器的时钟由 TI1FP1 或 TI2FP2 上升沿或下降沿提供。

根据两个输入的信号跳变顺序，产生计数脉冲和方向信号。根据该信号跳变顺序，计数器做相应递增或递减计数，同时自动修改计数方向（TIMx_CR1 寄存器中的 DIR 位）。无论计数器是仅在 TI1 或 TI2 边沿处计数，还是同时在 TI1 和 TI2 处计数，在任一输入端（TI1 或 TI2）信号的跳变，都会更新 DIR。

下表为计数方向与编码信号的关系。

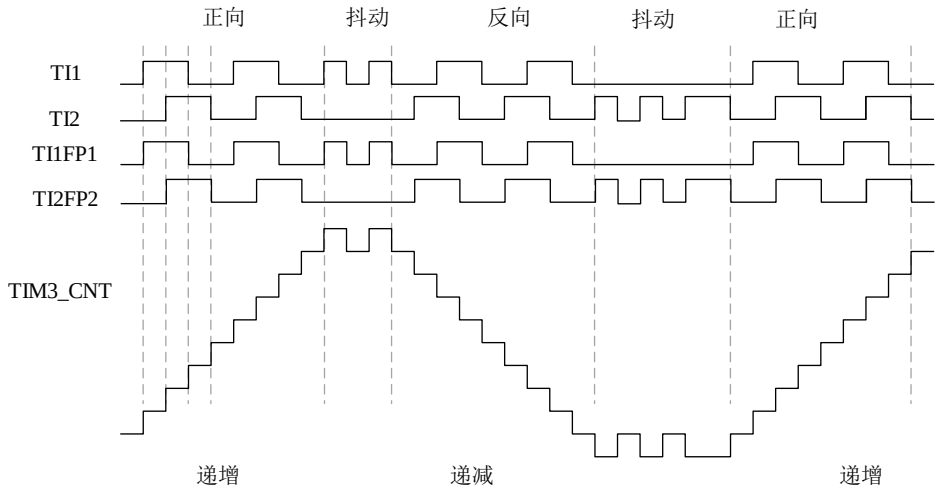
表 20-3　正交编码计数方向与编码器信号的关系

| 模式 | 计数方式 | TI1FP1对应TI2，TI2FP2对应TI1 | TI1FP1 信号 | | TI2FP2 信号 | |
|---|---|---|---|---|---|---|
| | | | 上升 | 下降 | 上升 | 下降 |
| 正交编码模式1（ENCMD[2:0]=001） | 仅在TI1处计数 | 高 | 递减 | 递增 | 不计数 | 不计数 |
| | | 低 | 递增 | 递减 | 不计数 | 不计数 |
| 正交编码模式2（ENCMD[2:0]=010） | 仅在TI2处计数 | 高 | 不计数 | 不计数 | 递增 | 递减 |
| | | 低 | 不计数 | 不计数 | 递减 | 递增 |
| 正交编码模式3（ENCMD[2:0]=011） | 在TI1和TI2处均计数 | 高 | 递减 | 递增 | 递增 | 递减 |
| | | 低 | 递增 | 递减 | 递减 | 递增 |
| 正交编码模式4（ENCMD[2:0]=100） | 仅在 TI2 高电平时的 TI1 处计数 | 高 | 递减 | 递增 | 不计数 | 不计数 |
| | | 低 | 不计数 | 不计数 | 不计数 | 不计数 |
| 正交编码模式5（ENCMD[2:0]=101） | 仅在 TI1 高电平时的 TI2 处计数 | 高 | 不计数 | 不计数 | 递增 | 递减 |
| | | 低 | 不计数 | 不计数 | 不计数 | 不计数 |

下图为计数器的工作原理，显示了计数信号的产生和方向控制。同时也显示了选择双边沿时，如何抑制输入抖动。抖动可能会在传感器靠近其中一个切换点放置时产生。本例中假设配置如下：

- TI1 映射到 TI1FP1 上：TIMx_CCM1 寄存器的 CC1S[1:0]位域写入 01；
- TI2 映射到 TI2FP2 上：TIMx_CCM1 寄存器中的 CC2S[1:0]位域写入 01；
- TI1FP1 未反相，TI1FP1 等于 TI1：TIMx_CCEN 寄存器中的 CC1P 位和 CC1NP 位清 0；
- TI2FP2 未反相，TI2FP2 等于 TI2：TIMx_CCEN 寄存器中的 CC2P 位和 CC2NP 位清 0；
- 上升沿和下降沿均有效：TIMx_SMC 寄存器中的 ENCMD[2:0]位域写入 011；
- 使能计数器：TIMx_CR1 寄存器中的 CEN 位置 1。

图 20-34 编码器接口模式下的计数器工作示例




### 脉冲电平编码模式

脉冲电平编码模式中，时钟是在 TI2 上单线提供的，而计数方向是 TI1 输入提供的。

TI1 的方向信号的极性通过 CC1P 位设置。CC1P=0 时当 TI1 为高电平时向上计数，当 TI1 为低电平时向下计数；CC1P＝1 时当 TI1 为低电平时向上计数，TI1 为高电平时向下计数。

脉冲电平编码模式，有 2 种计数方式：

- 脉冲电平编码模式 1：TIMx_SMC 寄存器中的 ENCMD[2:0]位域为 110 时，根据 CC2P 值，计数器在单个时钟沿上更新。CC2P＝0 对应于上升沿计数，CC2P＝1 对应于下降沿计数。
- 脉冲电平编码模式 2：ENCMD[2:0]位域为 111 时，计数器在时钟的上升沿和下降沿都被更新；

下表为计数方向与编码信号的关系。

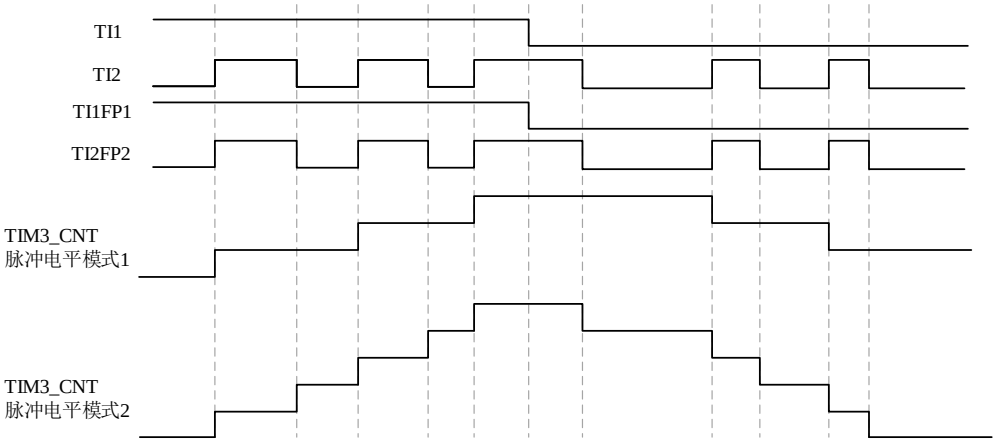
表 20-4 脉冲电平编码模式计数方向与编码器信号的关系

| 模式 | 计数方式 | TI1 | CC2P | TI2FP2 信号 | |
|---|---|---|---|---|---|
| | | | | 上升 | 下降 |
| 脉冲电平编码模式1（CC1P=0）（ENCMD[2:0]=110） | 根据CC2P值，单个时钟沿计数 | 高 | 0 | 递增 | 不计数 |
| | | | 1 | 不计数 | 递增 |
| | | 低 | 0 | 递减 | 不计数 |
| | | | 1 | 不计数 | 递减 |

| 模式 | 计数方式 | TI1 | CC2P | TI2FP2 信号 | |
|---|---|---|---|---|---|
| | | | | 上升 | 下降 |
| 脉冲电平编码模式1（CC1P=1）（ENCMD[2:0]=110） | 根据CC2P值，单个时钟沿计数 | 高 | 0 | 递减 | 不计数 |
| | | | 1 | 不计数 | 递减 |
| | | 低 | 0 | 递增 | 不计数 |
| | | | 1 | 不计数 | 递增 |
| 脉冲电平编码模式2（CC1P=0）（ENCMD[2:0]=111） | 时钟的上升沿和下降沿均计数 | 高 | - | 递增 | 递增 |
| | | 低 | - | 递减 | 递减 |
| 脉冲电平编码模式2（CC1P=1）（ENCMD[2:0]=111） | 时钟的上升沿和下降沿均计数 | 高 | - | 递减 | 递减 |
| | | 低 | - | 递增 | 递增 |

CC1P=CC2P=0 时，脉冲电平编码示例如下：

图 20-35 脉冲电平编码模式示例




### 20.3.17 定时器同步从模式

定时器从内部连接在一起，以实现 TIM 同步或级联。

表 20-5 TIMx 内部级联

| 从 TIM | ITR0(TS=000) | ITR1(TS=001) | ITR2(TS=010) |
|---|---|---|---|
| TIM3 | TIM1 | TIM4 | HALLTIM_CHG |
| TIM4 | TIM1 | TIM3 | - |

**复位模式**

当触发信号有效时，计数器及其预分频器被重新初始化。如果 TIMx_CR1 寄存 UDIS 位为 0，则会生成更新事件，且 TIMx_ARR 和输出比较模式下的 TIMx_CCx

寄存器的影子寄存器被更新。

在以下示例中，TI1 输入信号出现上升沿时，递增计数器清 0：

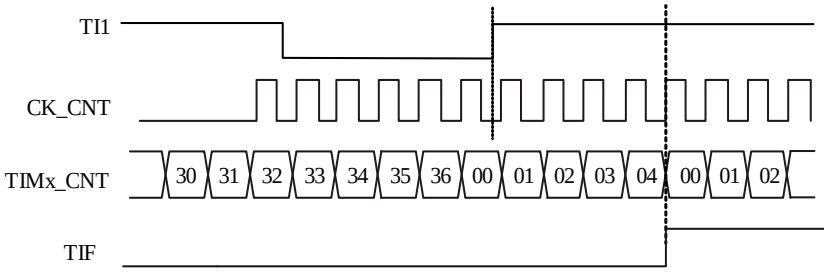
1) TI1 映射到 IC1 信号上：TIMx_CCM1 寄存器中的 CC1S[1:0]位域写入 01；
2) 配置 TI1 上升沿有效：TIMx_CCEN 寄存器中的 CC1P 位 CC1NP 位为 0；
3) 配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC1F[2:0]位域为 000）；
4) 配置为复位模式：TIMx_SMC 寄存器中的 SM_SEL[2:0]位域写入 100；
5) 选择 TI1FP1 作为触发源：TIMx_SMC 寄存器中的 TS[2:0]位域写入 101；
6) 使能计数器：TIMx_CR1 寄存器中的 CEN 位置 1。

当 TI1 出现上升沿时，计数器清 0，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；如果使能 TIMx_DIER 寄存器中的 TIE 或 TDMA_EN 位，则可发送中断或 DMA 请求。

TI1 的上升沿与实际计数器复位之间的延迟是由于触发信号 TI1 输入的同步电路引起的。

下图为自动重载值寄存器 TIMx_ARR=0x36 时的时序。

图 20-36 复位模式下的控制时序




### 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 TI1 输入为低电平时计数：

1) TI1 映射到 IC1 信号上：TIMx_CCM1 寄存器中的 CC1S[1:0]位域写入 01；
2) 配置 TI1 低电平有效：TIMx_CCEN 寄存器中的 CC1P 位为 1，CC1NP 位为 0；
3) 配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC1F[2:0]位域为 000）；
4) 定时器配置为门控模式：TIMx_SMC 寄存器中的 SM_SEL[2:0]位域写入

101；

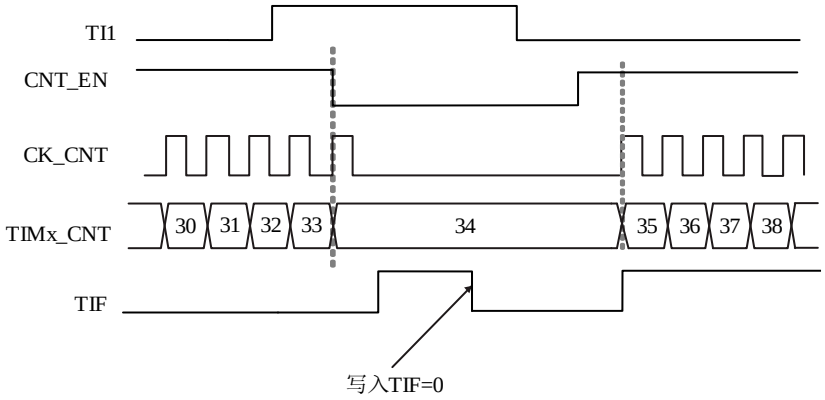
5) 选择 TI1FP1 作为触发源：TIMx_SMC 寄存器中的 TS[2:0]位域写入 101；

6) 使能计数器：TIMx_CR1 寄存器中的 CEN 位置 1。

当 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 标志置 1。

TI1 的上升沿与实际计数器停止之间的延迟是由于触发信号 TI1 输入的同步电路引起的。

图 20-37 门控模式下的控制时序




**触发模式**

所选输入端触发信号有效时可以使能计数器。

在以下示例中，当 TI2 输入上出现上升沿时，使能计数器：

1) TI2 映射到 IC2 信号上：TIMx_CCM1 寄存器中的 CC2S[1:0]位域写入 01；

2) 配置 TI2 上升沿有效：TIMx_CCEN 寄存器中的 CC2P 位和 CC2NP 位为 0，上升沿有效；

3) 配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC2F[2:0]位域为 000）；

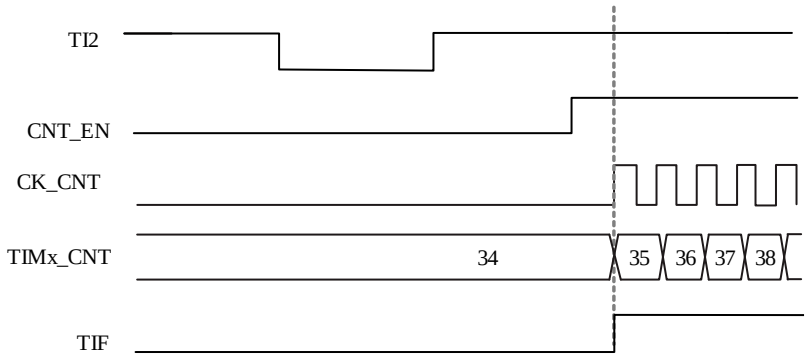
4) 定时器配置为触发模式：TIMx_SMC 寄存器中 SM_SEL[2:0]位域写入 110；

5) 选择 TI2FP2 作为触发源：TIMx_SMC 寄存器中的 TS[2:0]位域 110。

当 TI2 出现上升沿时，计数器开始根据内部时钟计数，TIMx_SR 寄存器中的 TIF 标志置 1。

TI2 的上升沿与实际计数器启动之间的延迟是由于触发信号 TI2 输入的同步电路引起的。

图 20-38 触发模式下的控制时序




**时钟模式 2+触发模式**

时钟模式 2 可与另一种从模式（时钟模式 1 除外）一起使用。此时，ETR 信号（时钟模式 2)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。

*注意：* *此模式下，不能通过 TIMx_SMC 寄存器中的 TS[2:0] 位域来选择 ETR 作为 TRIG 信号。*

在以下示例中，只要 TI1 出现上升沿，递增计数器即会在 ETR 信号的每个上升沿处计数一次：

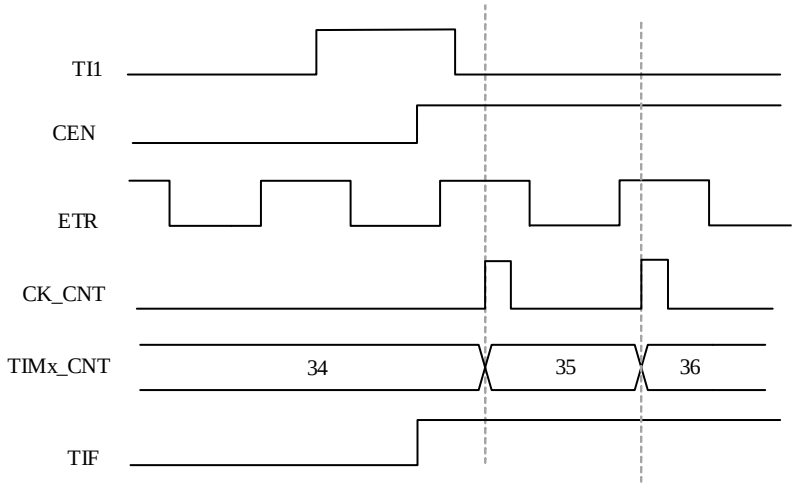
1) 配置 ETR（配置 TIMx_SMC 寄存器相关位）：
   - 配置输入滤波带宽（本例中不需要任何滤波器，因此保持 ETF[2:0]位域为 000）；
   - 配置预分频值为 1：ET_PRE[1:0]位域写入 00；
   - ETR 的上升沿有效：ETP 位清 0；
   - 使能时钟模式 2：ECEN 位置 1。
2) 配置 TI1（配置 TIMx_CCM1 寄存器相关位）：
   - 配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC1F[2:0]位域为 000）；
   - TI1 映射到 IC1 上：CC1S[1:0]位域写入 01；
3) 配置上升沿有效：TIMx_CCEN 寄存器中 CC1P 位和 CC1NP 位清 0；
4) 配置为触发模式：TIMx_SMC 寄存器中的 SM_SEL[2:0]写入 110；

5) TI1FP1 为触发源：TIMx_SMC 寄存器中的 TS[2:0]位域写入 101。

TI1 出现上升沿时将使能计数器，且 TIMx_SR 寄存器中的 TIF 标志置 1。在 ETR 出现上升沿时计数器计数。

ETR 信号的上升沿与实际计数器复位之间的延迟是由于触发信号 ETRP 输入的同步电路引起的。

图 20-39 外部时钟模式 2+触发模式下的控制时序

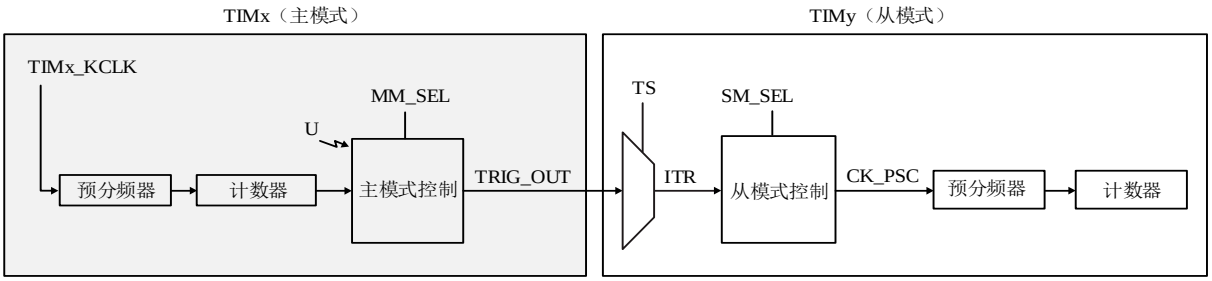



## 20.3.18 定时器同步主模式

TIMx 和内部其他 TIM 连在一起，可实现定时器同步或级联。

下图显示了配置为主/从定时器的连接框图。

图 20-40 主从定时器连接示例图




**TIMx 配置为 TIMy 的预分频器**

例如，将 TIMx 配置为 TIMy 的预分频器。其具体步骤如下：

1) 设置 TIMx 为主模式：配置 TIMx_CR2 寄存器中的 MM_SEL[2:0]为 010，则每次生成更新事件时，TRIG_OUT 都会输出一个上升沿；
2) 设置 TIMx 的 TRIG_OUT 输出到 TIMy：配置 TIMy 为从模式，并配置

ITRx 作为内部触发源。可通过 TIMy_SMC 寄存器中的 TS[2:0]位域配置；

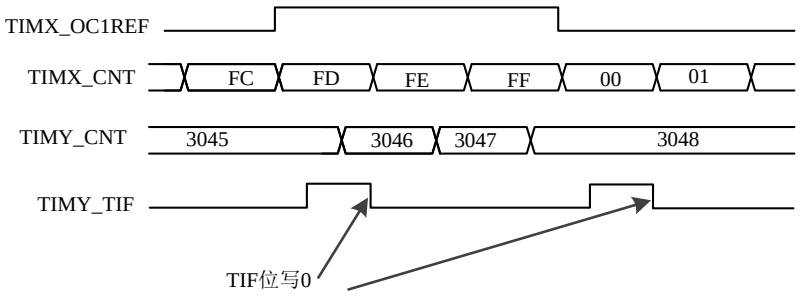
3) 设置 TIMy 的从模式控制器设为时钟模式 1：配置 TIMy_SMC 寄存器中的 SM_SEL[2:0]位域为 111，此时 TIMy 的时钟由 TIMx 周期性触发信号的上升沿提供；
4) 使能 TIMy 计数器：配置 TIMy_CR1 寄存器中的 CEN 位为 1；
5) 使能 TIMx 计数器：配置 TIMx_CR1 寄存器中的 CEN 位为 1。

**TIMx 使能 TIMy**

例如：TIMx 的输出比较 1 使能 TIMy，连接图如下所示。当 TIMx 的 OC1REF 为高电平时，TIMy 开始计数。

1) 设置 TIMx 为主模式，且输出比较 1 的参考信号（OC1REF）为触发输出：配置 TIMx_CR2 寄存器中的 MM_SEL[2:0]位域为 110（TIM1 则 MM_SEL[3:0]为 0110）；
2) 设置 TIMx 的 OC1REF 波形：配置 TIMx_CCM1 寄存器；
3) 设置 TIMy 用于接收 TIMx 的触发输入：配置 TIMy_SMC 寄存器中的 TS[2:0]位域，选择内部触发 ITRx；
4) 设置 TIMy 为门控模式：配置 TIMy_SMC 寄存器中的 SM_SEL[2:0]位域为 101；
5) 使能 TIMy 计数器：配置 TIMy_CR1 寄存器中的 CEN 位为 1；
6) 使能 TIMx 计数器：配置 TIMx_CR1 寄存器中的 CEN 位为 1；

图 20-41 使用 TIMx 的 OC1REF 信号对 TIMy 实施门控控制

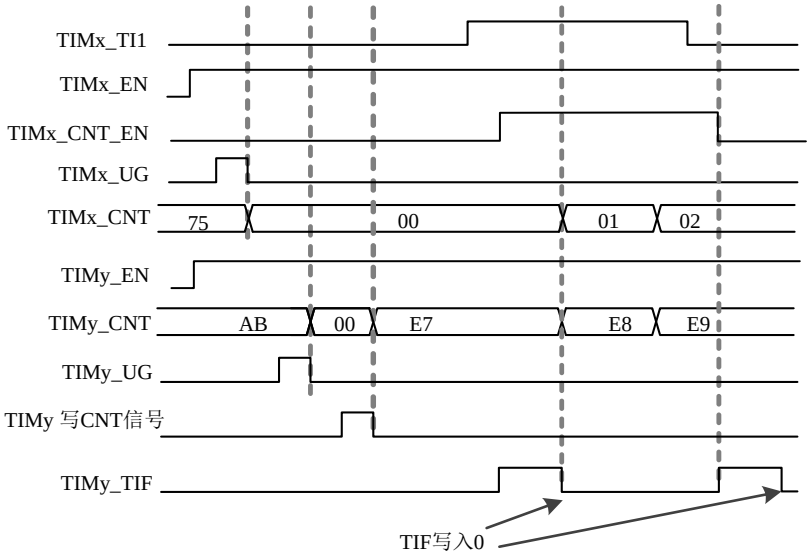



上图中，TIMy 的计数器和 PSC 预分频器在启动前未进行初始化，因此从各自的当前值开始计数。启动 TIMx 前，通过 UG 置 1 的方式复位 TIM，再从指定值开始计数。

如果配置 TIMx 从模式模式为门控模式，则当 TI1 为高电平，两个计数器同时开始计数；当 TI1 为低电平，两个计数器同时停止计数。下例中 TIMx 为主模式，从 0 开始计数。TIMy 为从模式，从 0xE7 开始计数。两个 TIM 的预分频比

相同。

图 20-42 使用 TIMx 的门控信号对 TIMy 实施门控控制




**使用一个外部触发同步启动 2 个定时器**

例如：TIMx 的 TI1 输入信号的上升沿使能 TIMx，同时也使能 TIMy。此时 TIMx 必须为主/从模式（对于 TI1 而言，TIMx 为从模式；对于 TIMy 而言，TIMx 为主模式）：

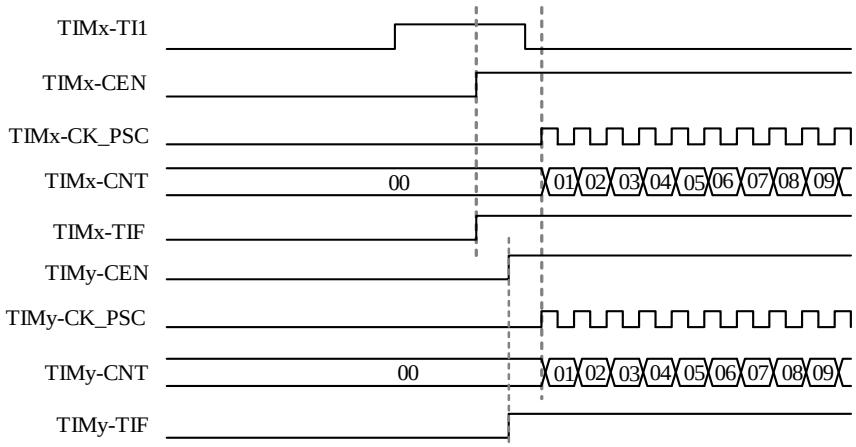
1) 设置 TIMx 的触发输出 TRIG_OUT：配置 TIMx_CR2 寄存器中的 MM_SEL[2:0]位域为 001；
2) 设置 TIMx 用于接收 TI1 的输入触发 TRIG：配置 TIMx_SMC 寄存器中的 TS[2:0]位域为 101；
3) 设置 TIMx 为触发模式：配置 TIMx_SMC 寄存器中的 SM_SEL[2:0]位域为 110；
4) 设置 TIMx 为主模式：配置 TIMx_SMC 寄存器中的 MS_MOD 位为 1；
5) 设置 TIMy 用于接收 TIMx 的输入触发：配置 TIMy_SMC 寄存器中的 TS[2:0]位域为 000；
6) 设置 TIMy 为触发模式：配置 TIMy_SMC 寄存器中的 SM_SEL[2:0]位域为 110。

当 TIMx 的 TI1 出现上升沿时，两个计数器开始同步计数，并且两个 TIF 标志都会置 1。

两个定时器在启动计数前进行了初始化（通过将各自的 UG 位置 1），两个计数器从 0 开始计数。如果计数器之前需要一个计数偏移量，可实时对其中一个计

数器的 TIM_CNT 寄存器进行写操作。

图 20-43 使用 TIMx 的 TI1 输入触发 TIMx 和 TIMy




### 20.3.19 DMA Burst 传输模式

TIMx 能够根据一个事件生成相应的 DMA 请求，将数据写入定时器的多个寄存器，也可读取定时器多个寄存器的值。关于 DMA Burst 功能，详见 *DMA 传输*。

例如，当发生更新事件，将数据依次更新到 TIM3_CCx 寄存器（x=1、2）中。

具体操作步骤如下（DMA 时钟已使能）：

1) 配置 DMA 通道（通道号请参见*表：DMA 通道请求信号*）：
   - DMA 通道的目标地址：TIM3_CC1 寄存器；
   - DMA 通道的源地址：要通过 DMA 传输到 TIM3_CCx 寄存器数据的 SRAM 缓冲区地址；
   - 根据配置目标地址为递增，源地址按需配置；
   - 要传输的数量：2；
   - 配置 DMA 的传输类型：*DMA 通道 x 控制寄存器（DMA_CCx）*中的 TYPE 位置 1 选择 Burst 传输；
   - 配置 DMA 的数据位宽：字（32bit）。
2) 使能 TIM3 的更新 DMA 请求：TIM3_DIER 寄存器中的 UDMA_EN 位置 1；

3) 使能 TIM3：TIM3_CR1 寄存器的 CEN 位置 1；

4) 使能 DMA 通道：*DMA 通道 x 控制寄存器（DMA_CCx）*中的 EN 位置 1。

### 20.3.20 调试模式

当 CPU 进入调试模式时（Cortex-M0+内核停止），TIMx 计数器会根据 *APB1 冻结寄存器（DBG_APB1_FZ）*寄存器中的 TIMx_HOLD 配置位选择继续计数或者停止计数。

## 20.4 TIM3/4 中断

置位 TIMx_DIER 寄存器中的相关使能位，则以下事件会触发中断：

表 20-6 中断请求

| 中断事件 | 事件标志 | 中断使能位 | 事件标志/中断清除方法 |
|---|---|---|---|
| 更新事件 | UIF_OVR/UIF_UDR | UIE | 向 UIF_OVR/UIF_UDR 位写 0 |
| 捕获/比较 1~2 事件 | CCxIF | CCxIE | 向 CCxIF 位写 0 或读取 TIMx_CCx 寄存器 |
| 触发事件 | TIF | TIE | 向 TIF 位写 0 |

## 20.5 TIM3/4 寄存器

TIMx 寄存器支持 32 位访问。

表 20-7 TIMx 基地址

| 外设 | 基地址 |
|---|---|
| TIM3 | 0x4000 0400 |
| TIM4 | 0x4000 0800 |

### 20.5.1 TIM 控制寄存器 1（TIMx_CR1）

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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | CLK_DIV[1:0] | | ARPE | CMS[1:0] | | DIR | OPM | Res. | UDIS | CEN |
| | | | | | | rw | rw | rw | rw | rw | rw | rw | | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:10 | 保留 | 写入无效 |
| 9:8 | CLK_DIV[1:0] | 时钟分频<br>定时器时钟频率（$f_{TIMx_KCLK}$）与数字滤波器所使用的采样时钟（$f_{DTS}$）之间的分频比。<br>00：$f_{DTS} = f_{TIMx_KCLK}$<br>01：$f_{DTS} = f_{TIMx_KCLK}/2$<br>10：$f_{DTS} = f_{TIMx_KCLK}/4$<br>11：保留（默认$f_{DTS} = f_{TIMx_KCLK}$） |
| 7 | ARPE | TIMx_ARR寄存器预装载使能控制<br>0：禁止<br>1：使能 |
| 6:5 | CMS[1:0] | 计数模式选择<br>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。上溢或下溢均触发更新事件；仅在递减计数过程中，当发生比较匹配时，TIMx_SR寄存器中的CCxIF位会置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。上溢或下溢均触发更新事件；仅在递增计数过程中，当发生比较匹配时，TIMx_SR寄存器中的CCxIF位会置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。上溢或下溢均触发更新事件；在递增计数或递减计数过程中，当发生比较匹配时，TIMx_SR寄存器中的CCxIF位会置1

| | | |
|---|---|---|
| 4 | DIR | 计数方向<br>当定时器配置为中心对齐模式时，该位为只读状态。<br>0：递增计数<br>1：递减计数 |
| 3 | OPM | 单脉冲模式<br>0：禁止：计数器在发生更新事件时不会停止计数<br>1：使能：计数器在发生更新事件时停止计数（CEN位自动清0） |
| 2 | 保留 | 写入无效 |
| 1 | UDIS | 更新事件禁止控制位<br>0：使能：允许更新事件产生<br>1：禁止：禁止更新事件产生<br>*注意：当UDIS位为1，从模式控制器接收到一个硬件复位（复位模式），计数器及PSC预分频计数器均被初始化，但不会将UIF_OVR/UIF_UDR置位。* |
| 0 | CEN | 计数器使能<br>0：禁止<br>1：使能<br>*注意：在使用时钟模式1/2或门控模式时，需软件将CEN置1。而触发模式可自动将CEN位置1。在单脉冲模式下，当发生更新事件时会自动将CEN位清0。* |

### 20.5.2 TIM 控制寄存器 2（TIMx_CR2）

偏移地址：0x04

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | TI1_XOR_SEL | Res. | MM_SEL[2:0] | | | Res. | | |
| | | | | | | | | rw | | rw | rw | rw | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 写入无效 |
| 7 | TI1_XOR_SEL | TI1选择<br>0：TIMx_CH1引脚连接到TI1输入<br>1：TIMx_CHx引脚异或组合后连接到TI1输入 |
| 6 | 保留 | 写入无效 |
| 5:3 | MM_SEL[2:0] | 主模式选择<br>触发输出（TRIG_OUT）信号连接到其他外设的控制方式<br>000：复位 — 以下信号都可以用作触发输出（TRIG_OUT）：<br>– TIMx_EVTG寄存器中的UG位<br>– 触发输入生成（从模式控制器配置为复位模式），但TRIG_OUT上的信号相比实际复位会有一定延时<br>001：使能 — 计数器使能信号用作触发输出（TRIG_OUT）。计数器使能信号由TIMx_CR1寄存器的CEN控制位与门控模式下的触发输入的逻辑“与”运算组合而成。为保证当前定时器与从外设实现完美同步，需选择主/从模式（TIMx_SMC寄存器中MS_MOD位置1）<br>010：更新 — 选择更新事件作为触发输出（TRIG_OUT）<br>011：上溢更新 — 仅计数上溢产生的更新事件作为触发输出（TRIG_OUT）<br>100：下溢更新 — 仅计数下溢产生的更新事件作为触发输出（TRIG_OUT）<br>101：比较脉冲 — 当发生捕获或比较匹配时，CC1IF标志置1 |

（即使已为高），此时触发输出（TRIG_OUT）都会发送一个正脉冲

110：比较 — OC1REF信号用作触发输出（TRIG_OUT）

111：比较 — OC2REF信号用作触发输出（TRIG_OUT）

| | | |
|---|---|---|
| 2:0 | 保留 | 写入无效 |

### 20.5.3 TIM 从模式控制寄存器（TIMx_SMC）

偏移地址：0x08

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ETP | ECEN | ET_PRE[1:0] | | Res. | ETF[2:0] | | | MS_MOD | TS[2:0] | | | OCCS | SM_SEL[2:0] | | |
| rw | rw | rw | rw | | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:27 | 保留 | 写入无效 |
| 26:24 | ENCMD[2:0] | 编码器模式选择<br>000：禁止编码器模式<br>001：正交编码模式 1<br>010：正交编码模式 2<br>011：正交编码模式 3<br>100：正交编码模式 4<br>101：正交编码模式 5<br>110：脉冲电平编码模式 1<br>111：脉冲电平编码模式 2<br>*注意：此位域仅 TIM3 写入有效。* |
| 23:16 | 保留 | 写入无效 |
| 15 | ETP | ETR极性选择<br>0：高电平或上升沿有效<br>1：低电平或下降沿有效 |

| | | |
|---|---|---|
| 14 | ECEN | 时钟模式2使能<br>该模式和以下从模式同时使用：<br>– 复位模式<br>– 门控模式<br>– 触发模式<br>如果同时使能时钟模式1和时钟模式2，则时钟模式2的优先级高。<br>0：禁止<br>1：使能 |
| 13:12 | ET_PRE[1:0] | ETR 预分频器<br>ETRP 频率不得超过 TIMx_KCLK 频率的 1/2。<br>00：1 分频<br>01：2 分频<br>10：4 分频<br>11：8 分频 |
| 11 | 保留 | 写入无效 |
| 10:8 | ETF[2:0] | ETR 滤波器控制<br>ETRP 信号的采样频率和 ETRP 的数字滤波宽度。连续 4 个采样周期均采样到有效电平时，才视为一个有效电平：<br>000：无滤波器，按 $f_{DTS}$ 频率进行采样<br>001：$f_{SAMPLING} = f_{TIMx_KCLK}$<br>010：$f_{SAMPLING} = f_{DTS}/2$<br>011：$f_{SAMPLING} = f_{DTS}/4$<br>100：$f_{SAMPLING} = f_{DTS}/8$<br>101：$f_{SAMPLING} = f_{DTS}/16$<br>110：$f_{SAMPLING} = f_{DTS}/32$<br>111：$f_{SAMPLING} = f_{DTS}/32$ |
| 7 | MS_MOD | 主/从模式<br>0：不执行任何操作<br>1：当前定时器的触发输入事件（TRIG）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 TRIG_OUT）。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 |

| | | |
|---|---|---|
| 6:4 | TS[2:0] | 触发源选择<br>000：内部触发 0（ITR0）<br>001：内部触发 1（ITR1）<br>010：内部触发 2（ITR2）<br>100：TI1 边沿检测器 TI1F_ED<br>101：定时器输入 1（TI1FP1）<br>110：定时器输入 2（TI2FP2）<br>111：ETR 输入（ETRF）<br>其他：保留（默认内部触发 0）<br>*注意：有关 ITRx 含义的详细信息，请参见表：TIMx 内部级联；建议在 SM_SEL[2:0] 位域配置为“000”时进行更改，以避免在改变时产生错误的边沿检测。“010”配置仅对 TIM3 生效。* |
| 3 | OCCS | OCREF 清除信号选择<br>0：清除信号连接到 COMP1/2 输出（由 TIMx_CFG 寄存器的 OCREF_CLR 位决定）<br>1：清除信号连接到 ETRF |
| 2:0 | SM_SEL[2:0] | 从模式选择<br>触发信号（TRIG）的有效边沿由外部输入端所选的极性决定。<br>0xx：禁止从模式 — PSC 预分频器时钟由内部时钟提供，TIMx_CR1 寄存器中的 CEN 位为 1 时，启动计数<br>100：复位模式 — 触发输入 TRIG 信号上升沿时，重新初始化计数器并生成更新事件<br>101：门控模式 — 触发输入 TRIG 信号为高电平时使能计数器时钟。只要 TRIG 信号变为低电平，计数器立即停止计数（但不复位）。计数器的启动和停止都被控制<br>110：触发模式 — 触发输入 TRIG 信号出现上升沿时启动计数器（但不复位）。只控制计数器的启动<br>111：时钟模式 1 — 触发输入 TRIG 信号的上升沿提供计数器时钟 |

### 20.5.4 TIM 中断/DMA 使能寄存器（TIMx_DIER）

偏移地址：0x0C

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | TDMA_EN | Res. | | | CC2_DMAEN | CC1_DMAEN | UDMA_EN | Res. | TIE | Res. | | | CC2IE | CC1IE | UIE |
| | rw | | | | rw | rw | rw | | rw | | | | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:15 | 保留 | 写入无效 |
| 14 | TDMA_EN | 触发事件的DMA请求使能<br>0：禁止<br>1：使能 |
| 13:11 | 保留 | 写入无效 |
| 10 | CC2_DMAEN | 通道2捕获/比较事件的DMA请求使能<br>0：禁止<br>1：使能 |
| 9 | CC1_DMAEN | 通道1捕获/比较事件的DMA请求使能<br>0：禁止<br>1：使能 |
| 8 | UDMA_EN | 更新事件的DMA请求使能<br>0：禁止<br>1：使能 |
| 7 | 保留 | 写入无效 |
| 6 | TIE | 触发事件的中断使能<br>0：禁止<br>1：使能 |
| 5:3 | 保留 | 写入无效 |
| 2 | CC2IE | 通道2捕获/比较事件的中断使能<br>0：禁止 |

| | | |
|---|---|---|
| | | 1：使能 |
| 1 | CC1IE | 通道1捕获/比较事件的中断使能<br>0：禁止<br>1：使能 |
| 0 | UIE | 更新事件的中断使能<br>0：禁止<br>1：使能 |

### 20.5.5 TIM 状态寄存器（TIMx_SR）

偏移地址：0x10

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | UIF_UDR | UIF_OVR | Res. | | | | | | | |
| | | | | | | rc_w0 | rc_w0 | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | CC2OF | CC1OF | Res. | | TIF | Res. | | | CC2IF | CC1IF | Res. |
| | | | | | rc_w0 | rc_w0 | | | rc_w0 | | | | rc_w0 | rc_w0 | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:26 | 保留 | 写入无效 |
| 25 | UIF_UDR | 计数下溢更新事件标志<br>发生计数下溢更新事件时自动置1。通过写0可清除该位。如果TIMx_DIER寄存器中UIE位为1，触发中断。<br>0：未发生下溢更新<br>1：发生下溢更新 |
| 24 | UIF_OVR | 计数上溢更新事件标志<br>发生计数上溢更新事件时自动置1。通过写0可清除该位。如果TIMx_DIER寄存器中UIE位为1，触发中断。<br>0：未发生上溢更新<br>1：发生上溢更新 |
| 23:11 | 保留 | 写入无效 |

| | | |
|---|---|---|
| 10 | CC2OF | 通道2的重复捕获标志<br>请参见CC1OF位说明 |
| 9 | CC1OF | 通道1的重复捕获标志<br>通过写入0可将该位清0。<br>0：未检测到重复捕获<br>1：CC1IF标志已置1时，再次检测到捕获；TIMx_CC1寄存器捕获到新的计数器值 |
| 8:7 | 保留 | 写入无效 |
| 6 | TIF | 触发事件件标志<br>– 门控模式：该标志将在计数器启动或停止时置1。<br>– 非门控模式：当使能从模式控制器后，在TRIG信号上检测到有效边沿时，该标志自动置1。<br>通过写0可清除该位。如果TIMx_DIER寄存器中TIE位为1，触发中断。<br>0：未发生触发事件<br>1：发生触发事件 |
| 5:3 | 保留 | 写入无效 |
| 2 | CC2IF | 通道2的捕获/比较事件件标志<br>请参见CC1IF位说明 |
| 1 | CC1IF | 通道1的捕获/比较事件件标志<br>发生捕获/比较事件件时，如果TIMx_DIER寄存器中CCxIE位为1，触发中断。<br>如果通道1配置为输出：<br>当TIMx_CNT计数器的值与TIMx_CC1寄存器的值匹配时，此标志自动置1。中心对齐模式下除外（请参见TIMx_CR1寄存器中的CMS位说明）。通过写0可清除该位。<br>0：不匹配<br>1：匹配<br>如果通道1配置为输入：<br>发生捕获事件件时自动置1，通过写0或读取TIMx_CC1寄存器将清除该位。<br>0：未发生输入捕获事件 |

1：发生输入捕获事件，TIMx_CC1寄存器中已捕获到计数器值

| | | |
|---|---|---|
| 0 | 保留 | 写入无效 |

### 20.5.6 TIM 事件产生寄存器（TIMx_EVTG）

偏移地址：0x14

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | CC2G | CC1G | UG |
| | | | | | | | | | | | | | w | w | w |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:3 | 保留 | 写入无效 |
| 2 | CC2G | 生成通道2捕获/比较事件<br>请参见CC1G位说明 |
| 1 | CC1G | 生成通道1捕获/比较事件<br>软件置1可生成捕获/比较事件。<br>如果通道配置为输出：<br>– TIMx_SR寄存器的CC1IF标志位置1<br>– 如果TIMx_DIER寄存器中CC1IE位为1，生成中断<br>如果通道配置为输入：<br>– TIMx_SR寄存器的CC1IF标志位置1；如果CC1IF标志已为高电平，CC1OF标志将置1<br>– TIMx_CC1寄存器中将捕获到计数器当前值如果TIMx_DIER寄存器中CC1IE位为1，触发中断 |
| 0 | UG | 初始化生成<br>此位由软件置1可生成相应加载和初始化计数器效果。<br>0：不执行任何操作<br>1：计数器，预分频计数器均被初始化，并加载使能预装载功能的影子寄存器和PSC寄存器（UDIS位为0时） |

### 20.5.7 TIM 计数值寄存器（TIMx_CNT）

偏移地址：0x18

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CNT[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 |
| CNT[15:0] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | CNT[31:0] | 计数器值<br>*注意：TIM3仅CNT[15:0]有效。* |

### 20.5.8 TIM 捕获/比较使能寄存器（TIMx_CCEN）

偏移地址：0x20

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | | | | | | | | CC2NP | Res. | CC2P | CC2E | CC1NP | Res. | CC1P | CC1E |
| | | | | | | | | rw | | rw | rw | rw | | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 写入无效 |
| 7 | CC2NP | 通道2捕获极性<br>请参见CC1NP位说明，仅在配置为输入有效 |
| 6 | 保留 | 写入无效 |
| 5 | CC2P | 通道2捕获/比较输出极性<br>请参见CC1P位说明 |

| | | |
|---|---|---|
| 4 | CC2E | 通道2捕获/比较输出使能<br>请参见CC1E位说明 |
| 3 | CC1NP | 通道1捕获极性<br>此位与CC1P配合使用，用以定义TI1FP1/TI2FP1的极性。请参见CC1P说明。 |
| 2 | 保留 | 写入无效 |
| 1 | CC1P | 通道1捕获/比较输出极性<br>通道1配置为输出：<br>0：OC1高电平有效<br>1：OC1低电平有效<br>通道1配置为输入：<br>CC1NP/CC1P位选择TI1FP1和TI2FP1的有效极性：<br>– CC1NP=0，CC1P=0：<br>TIxFP1上升沿有效（在复位模式、时钟模式1、触发模式或捕获模式下生效）<br>TIxFP1未反相（在门控模式或编码器模式下生效）<br>– CC1NP=0，CC1P=1：<br>TIxFP1下降沿有效（在复位模式、时钟模式1、触发模式或捕获模式下生效）<br>TIxFP1反相（在门控模式或编码器模式下生效）<br>– CC1NP=1，CC1P=1：<br>TIxFP1上升沿和下降沿都有效（在复位模式、时钟模式1、触发模式或捕获模式下生效）<br>TIxFP1未反相（在门控模式下生效，编码器模式下不得使用此配置）。<br>– CC1NP=1，CC1P=0：保留（默认未反相/上升沿有效） |
| 0 | CC1E | 通道1捕获/比较输出使能<br>通道1配置为输出：<br>0：禁止<br>1：使能<br>通道1配置为输入：<br>输入捕获使能位<br>0：禁止<br>1：使能 |

### 20.5.9 TIM 捕获/比较模式寄存器 1（TIMx_CCM1）--（输出比较模式）

偏移地址：0x28

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OC2CE | OC2M[2:0] | | | OC2PE | OC2FE | CC2S[1:0] | | OC1CE | OC1M[2:0] | | | OC1PE | OC1FE | CC1S[1:0] | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15 | OC2CE | 通道2输出比较清零使能<br>请参见OC1CE说明 |
| 14:12 | OC2M[2:0] | 通道2输出比较模式选择<br>请参见OC1M[2:0]位域说明 |
| 11 | OC2PE | 通道2输出比较预装载使能<br>请参见OC1PE位说明 |
| 10 | OC2FE | 通道2输出比较快速使能<br>请参见OC1FE位说明 |
| 9:8 | CC2S[1:0] | 通道2捕获/比较选择<br>仅当通道关闭时（TIMx_CCEN中的CC2E位为0），CC2S位才可写入。<br>00：通道2配置为输出<br>01：通道2配置为输入，TI2映射到IC2上<br>10：通道2配置为输入，TI1映射到IC2上<br>11：通道2配置为输入，TRC映射到IC2上 |
| 7 | OC1CE | 通道1输出比较清零使能<br>0：OC1REF不受OC_CLR信号影响<br>1：OC_CLR信号上检测到高电平时，OC1REF立即清0 |

| | | |
|---|---|---|
| 6:4 | OC1M[2:0] | 通道1输出比较模式选择<br>定义信号OC1REF的行为，OC1REF信号决定OC1的值。OC1REF为高电平有效。OC1的有效电平则取决于TIMx_CCEN寄存器的CC1P位。<br>000：冻结 — TIMx_CC1与TIMx_CNT间的比较对输出无影响<br>001：匹配输出有效电平 — 当TIMx_CNT与TIMx_CC1匹配时，OC1REF信号强制变为有效电平（匹配前OC1REF信号为无效电平）<br>010：匹配输出无效电平 — 当TIMx_CNT与TIMx_CC1匹配时，OC1REF信号强制变为无效电平（匹配前OC1REF信号为无效电平）<br>011：翻转 — 当TIMx_CNT与TIMx_CC1匹配时，OC1REF发生翻转<br>100：强制变为无效电平 — OC1REF强制变为无效电平<br>101：强制变为有效电平 — OC1REF强制变为有效电平<br>110：PWM模式1：<br>– 在递增计数模式下，当TIMx_CNT < TIMx_CC1，通道1输出为有效电平，否则为无效电平<br>– 在递减计数模式下，当TIMx_CNT > TIMx_CC1，通道1输出为无效电平，否则为有效电平<br>111：PWM模式2：<br>– 在递增计数模式下，当TIMx_CNT < TIMx_CC1，通道1输出为无效电平，否则为有效电平<br>– 在递减计数模式下，当TIMx_CNT > TIMx_CC1，通道1输出为有效电平，否则为无效电平 |
| 3 | OC1PE | 通道1输出比较预装载使能<br>0：禁止：可随时向TIMx_CC1写入数据，写入后将立即生效<br>1：使能：可读/写访问预装载寄存器，TIMx_CC1预装载值在更新事件时才会生效 |
| 2 | OC1FE | 通道1输出比较快速使能<br>此位用于加快触发输入事件对OC输出的影响<br>0：禁止：触发输入事件发生时，OC1也将根据计数器和CC1值匹配结果正常输出<br>1：使能：触发输入有效边沿相当于OC1输出上的比较匹配。随后，无论比较结果如何，OC1都设置为比较后电平。仅当通 |

道配置为PWM1或PWM2模式时，OC1FE才会起作用

| | | |
|---|---|---|
| 1:0 | CC1S[1:0] | 通道1捕获/比较选择<br>仅当通道关闭时（TIMx_CCEN中的CC1E位为0），才可向CC1S位写入数据。<br>00：通道1配置为输出<br>01：通道1配置为输入，TI1映射到IC1上<br>10：通道1配置为输入，TI2映射到IC1上<br>11：通道1配置为输入，TRC映射到IC1上 |

### 20.5.10 TIM 捕获/比较模式寄存器 1（TIMx_CCM1）--（输入捕获模式）

偏移地址：0x28

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | IC2F[2:0] | | | IC2PSC[1:0] | | CC2S[1:0] | | Res. | IC1F[2:0] | | | IC1PSC[1:0] | | CC1S[1:0] | |
| | rw | rw | rw | rw | rw | rw | rw | | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:15 | 保留 | 写入无效 |
| 14:12 | IC2F[2:0] | 通道2输入捕获滤波器配置<br>请参见IC1F[2:0]位域说明 |
| 11:10 | IC2PSC[1:0] | 通道2输入捕获预分频器<br>请参见IC1PSC[1:0]位域说明 |
| 9:8 | CC2S[1:0] | 通道2捕获/比较选择<br>请参见*TIM捕获/比较模式寄存器1（TIMx_CCM1）--（输出比较模式）*CC2S[1:0]说明 |
| 7 | 保留 | 写入无效 |
| 6:4 | IC1F[2:0] | 通道1输入捕获滤波器配置<br>配置 TI1 输入的采样频率和数字滤波宽度。连续 4 个采样周期 |

| | | |
|---|---|---|
| | | 均采样到有效电平时，才视为一个有效电平：<br>000：无滤波器，按 $f_{DTS}$ 频率进行采样<br>001：$f_{SAMPLING} = f_{TIMx_KCLK}$<br>010：$f_{SAMPLING} = f_{DTS}/2$<br>011：$f_{SAMPLING} = f_{DTS}/4$<br>100：$f_{SAMPLING} = f_{DTS}/8$<br>101：$f_{SAMPLING} = f_{DTS}/16$<br>110：$f_{SAMPLING} = f_{DTS}/32$<br>111：$f_{SAMPLING} = f_{DTS}/32$ |
| 3:2 | IC1PSC[1:0] | 通道1输入捕获预分频器<br>只要TIMx_CCEN寄存器的CC1E位为0，预分频器便立即复位。<br>00：无预分频器，捕获输入上信号上每检测到一个有效边沿便执行捕获<br>01：每发生2个事件执行一次捕获<br>10：每发生4个事件执行一次捕获<br>11：每发生8个事件执行一次捕获 |
| 1:0 | CC1S [1:0] | 通道1捕获/比较选择<br>请参见*TIM捕获/比较模式寄存器1（TIMx_CCM1）--（输出比较模式）*CC1S[1:0]说明 |

### 20.5.11 TIM 预分频值寄存器（TIMx_PSC）

偏移地址：0x34

复位值：0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:4 | 保留 | 写入无效 |
| 3:0 | PSC[3:0] | 预分频值 |

计数器时钟频率（CK_CNT）等于 $f_{CK_PSC}/（2^{psc[3:0]}）$。
当发生更新事件时，该位域的值加载到预分频影子寄存器中。

### 20.5.12 TIM 自动重载值寄存器（TIMx_ARR）

偏移地址：0x38

复位值：TIM3 0x0000 FFFF

TIM4 0xFFFF FFFF

| 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 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | ARR[31:0] | 自动重载值<br>当自动重载值为零时，计数器暂停计数。<br>如果TIMx_CR1寄存器的ARPE位为0，TIMx_ARR寄存器的值会立即更新。如果ARPE位为1时，只有当发生更新事件，TIMx_ARR寄存器中的值才会更新到影子寄存器中。<br>*注意：TIM3仅ARR[15:0]有效。* |

### 20.5.13 TIM 捕获/比较寄存器 x（TIMx_CCx）

偏移地址：0x3C+0x04 ×x，（x = 0 ~ 1）

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CCx[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 |
| CCx[15:0] | | | | | | | | | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:0 | CCx[31:0] | 通道x的捕获/比较值（x = 1 ~ 2）<br>通道x配置为输出比较模式： |

- 当TIMx_CCM1寄存器中的OCxPE位为1时，使能预装载功能，该位域的新值将在更新事件后生效
- 当TIMx_CCM1寄存器中的OCxPE位为0时，该位域的新值立即生效

通道x配置为输入捕获模式：

该寄存器为只读状态。当发生输入捕获事件时，将捕获当前计数器的值。

*注意：TIM3仅CCx[15:0]有效。*

### 20.5.14 TIM 配置寄存器（TIMx_CFG）

偏移地址：0x74

复位值：0x00000000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:1 | 保留 | 写入无效 |
| 0 | OCREF_CLR | OCxREF清除源选择<br>0：COMP1的输出连接到OCREF_CLR输入<br>1：COMP2的输出连接到OCREF_CLR输入 |

### 20.5.15 TIM3 轮换功能寄存器 1（TIM3_AF1）

偏移地址：0x78

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | ETR_SEL[2] |
| | | | | | | | | | | | | | | | rw |

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:17 | 保留 | 写入无效 |
| 16:14 | ETR_SEL[2:0] | ETR输入源选择<br>000：GPIO<br>001：COMP1输出<br>010：COMP2输出<br>011：MCO<br>100：LXTAL<br>101：EXTCLK<br>110：RCL<br>111：保留（默认GPIO） |
| 13:0 | 保留 | 写入无效 |

### 20.5.16 TIM4 轮换功能寄存器 1（TIM4_AF1）

偏移地址：0x78

复位值：0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15:14 | ETR_SEL[1:0] | ETR输入源选择<br>00：GPIO<br>01：COMP1输出<br>10：COMP2输出<br>11：MCO |
| 13:0 | 保留 | 写入无效 |

### 20.5.17 TIM3 定时器输入选择寄存器（TIM3_TISEL）

偏移地址：0x7C

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | TI2_SEL | Res. | | | | | | | TI1_SEL |
| | | | | | | | rw | | | | | | | | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:9 | 保留 | 写入无效 |
| 8 | TI2_SEL | TI2 输入源选择<br>0：TIM3_CH2<br>1：COMP2 输出 |
| 7:1 | 保留 | 写入无效 |
| 0 | TI1_SEL | TI1 输入源选择<br>0：TIM3_CH1<br>1：COMP1 输出 |

### 20.5.18 TIM4 定时器输入选择寄存器（TIM4_TISEL）

偏移地址：0x7C

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | TI2_SEL[2:0] | | | Res. | | | | | | TI1_SEL | |
| | | | | | rw | rw | rw | | | | | | | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:11 | 保留 | 写入无效 |

| | | |
|---|---|---|
| 10:8 | TI2_SEL[2:0] | TI2 输入源选择<br>000：TIM4_CH2<br>001：COMP2 输出<br>010：LXTAL<br>011：EXTCLK<br>100：MCO<br>101：RCL<br>110：TIM3_IC2 输出<br>111：保留（默认 TIM4_CH2） |
| 7:2 | 保留 | 写入无效 |
| 1:0 | TI1_SEL[1:0] | TI1 输入源选择<br>00：TIM4_CH1<br>01：COMP1 输出<br>10：TIM3_IC1 输出<br>11：保留（默认 TIM4_CH1） |

# 21 霍尔定时器（HALLTIM）

## 21.1 简介

霍尔传感器专用定时器（HALLTIM）用于处理霍尔传感器的输出信号。霍尔传感器是一种集成有霍尔效应传感器的电路，能够感知磁场强度的变化，并通过内部电路将这个变化转化为电信号，常用于电机控制、位置检测等应用中。

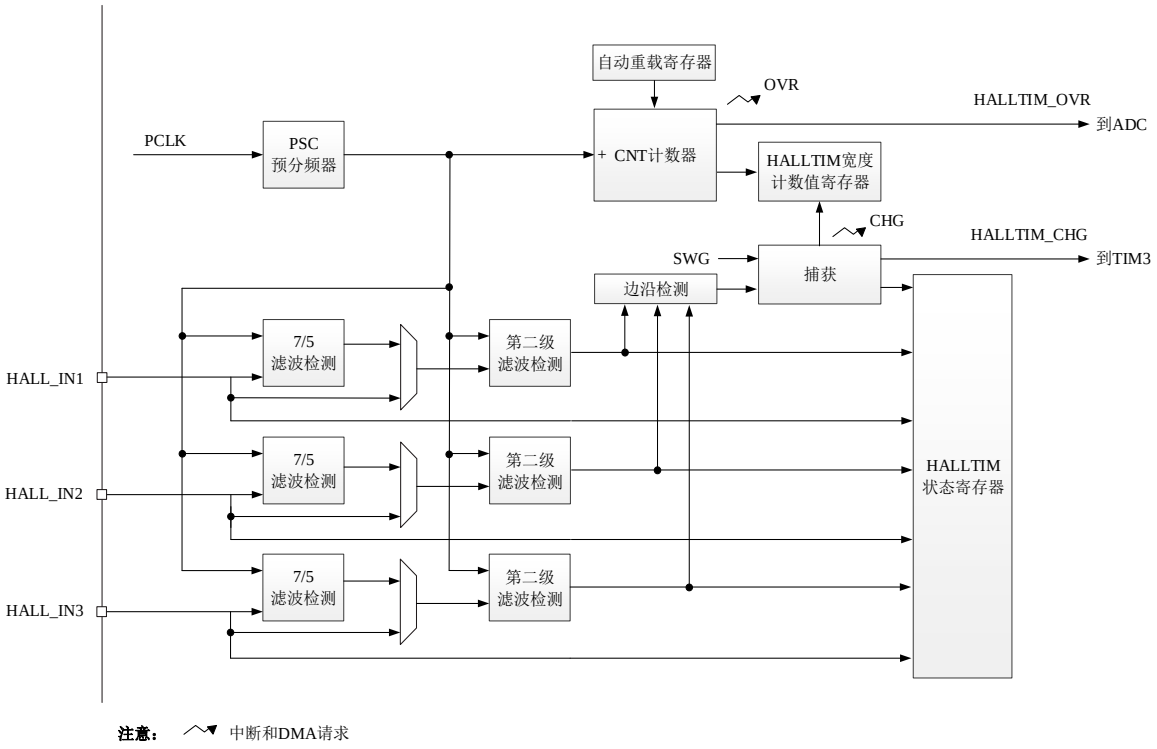
## 21.2 HALLTIM 主要特性

- 24 位递增自动重载计数器
- 可编程预分频器
- 3 路独立输入捕获通道
- 每路通道均包含两级滤波器
- 支持 HALL 信号捕获：
  - 记录滤波后 HALL 信号
  - 记录信号变化时间并输出中断
- 支持计数溢出并输出中断
- 可软件触发 HALL 信号捕获
- 可定时触发 ADC
- 支持 DMA

## 21.3 HALLTIM 功能描述

### 21.3.1 HALLTIM 框图

图 21-1 HALLTIM 框图




### 21.3.2 HALLTIM 引脚

表 21-1 HALLTIM 引脚

| 引脚名称 | 信号类型 | 说明 |
|---|---|---|
| HALL_IN1 | 输入 | HALLTIM 通道 1 输入引脚 |
| HALL_IN2 | 输入 | HALLTIM 通道 2 输入引脚 |
| HALL_IN3 | 输入 | HALLTIM 通道 3 输入引脚 |

### 21.3.3 PSC 预分频器

PSC 预分频器对 PCLK 进行分频后作为 HALLTIM 工作时钟使用，用于滤波器和 CNT 计数器，通过 HALLTIM_CR 寄存器中的 CLK_DIV[1:0]位域配置分频值。

### 21.3.4 计数器

HALLTIM 内置 24 位的递增计数器，计数器具有自动重载功能，重载值可通过 HALLTIM_ARR 寄存器配置，当计数器的计数值达到设置的自动重载值时，

HALLTIM_SR 寄存器的 OVR 位置 1，同时计数器重新从 0 开始计数，如果 HALLTIM_DIER 寄存器的 OVR_IE 位置 1，则会触发计数溢出中断。

当 HALLTIM 计数溢出后，可产生 HALLTIM_OVR 信号到 ADC。

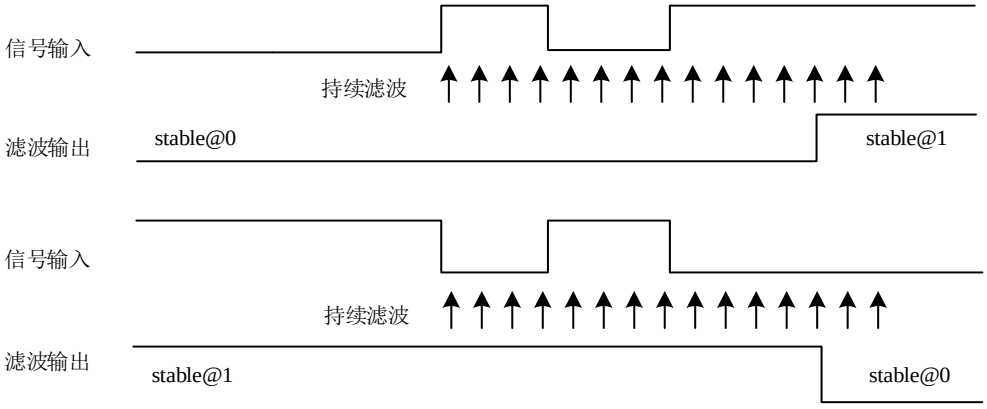
### 21.3.5 滤波器

霍尔传感器的输出信号通过 GPIO 输入到 HALLTIM 模块。HALLTIM 每路通道均包含两级滤波器，主要用于去除 HALL 信号上的毛刺；滤波前 HALL 信号实时值可通过 HALLTIM_SR 寄存器中的 RAW_DATA[2:0]位域获取。

#### 第一级滤波

第一级滤波为 7 判 5 滤波，即连续 7 个采样点中，如果达到或超过 5 个 1 则输出 1，如果达到或超过 5 个 0 则输出 0，否则输出保持上一次的滤波结果。具体如下图所示，第一级滤波使能可通过 HALLTIM_CR 寄存器的 FLT75_EN 位配置。

图 21-2 第一级滤波示意图




#### 第二级滤波

第二级滤波为连续滤波，连续 N 个采样点中，如果全为 0 则输出 0，如果全为 1 则输出 1，否则输出保持上一次的滤波结果，滤波宽度可通过 HALLTIM_CR 寄存器的 FILT_LEN[14:0]位域设置。

### 21.3.6 捕获

独立的 24 位计数器可用于测量连续两次 HALL 信号变化之间的时间宽度。24 位计数器从 0 开始计数，当发生 HALL 信号变化时:

- HALLTIM_CNT 寄存器中的值保存到 HALLTIM_WIDTH 寄存器中的 CAP_CNT[23:0]位域；

- 滤波后的 HALL 信号保存到 HALLTIM_SR 寄存器中的 FILT_DATA[2:0]位域；
- HALLTIM_SR 寄存器中的 CHG 位置 1，如果 HALLTIM_DIER 寄存器的 CHG_IE 位置 1，则会触发 HALL 信号变化中断；
- HALLTIM_CNT 寄存器中的 CNT[23:0]位域清 0，计数器重新从 0 开始计数。

向 HALLTIM_CNT 寄存器写入任意值，可清零计数值，计数器重新从 0 开始计数。

当 HALLTIM 检测到 HALL 信号变化后，可产生 HALLTIM_CHG 信号到 TIM3。

### 21.3.7 DMA 访问

通过 HALLTIM_DIER 寄存器的 OVR_DMAEN 位使能计数溢出 DMA 请求。

通过 HALLTIM_DIER 寄存器的 CHG_DMAEN 位使能 HALL 信号变化 DMA 请求。

### 21.3.8 调试模式

当芯片进入调试模式调试暂停时，HALLTIM 计数器会根据 DBG 模块中的 *APB1 冻结寄存器（DBG_APB1_FZ）*中的 HALLTIM_HOLD 选择继续正常工作或者停止工作。

## 21.4 HALLTIM 中断

置位 HALLTIM_DIER 寄存器中的相关使能位，则以下事件会触发中断：

表 21-2 中断请求

| 中断事件 | 事件标志 | 中断使能位 | 事件标志/中断清除方法 |
|---|---|---|---|
| 计数溢出 | OVR | OVR_IE | 向 OVR 位写 1 |
| HALL 信号变化 | CHG | CHG_IE | 向 CHG 位写 1 |

## 21.5 HALLTIM 寄存器

HALLTIM 寄存器支持 32 位访问。

表 21-3 HALLTIM 基地址

| 外设 | 基地址 |
|---|---|
| HALLTIM | 0x4000 6000 |

### 21.5.1 HALLTIM 控制寄存器（HALLTIM_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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | FILT_LEN[14:0] | | | | | | | | | | | | | | |
| | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | CLK_DIV[1:0] | | Res. | | | | | | FLT75_EN | EN |
| | | | | | | rw | rw | | | | | | | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31 | 保留 | 写入无效 |
| 30:16 | FILT_LEN[14:0] | 滤波宽度<br>低于对应脉冲宽度的信号将被硬件自动过滤掉。滤波宽度的计算公式为 FILT_LEN[14:0] + 1。 |
| 15:10 | 保留 | 写入无效 |
| 9:8 | CLK_DIV[1:0] | HALLTIM 时钟分频系数<br>00：1 分频<br>01：2 分频<br>10：4 分频<br>11：8 分频 |
| 7:2 | 保留 | 写入无效 |
| 1 | FIL75_EN | 7/5 滤波器使能 |

0：禁止

1：使能

| | | |
|---|---|---|
| 0 | EN | HALLTIM 模块使能<br>0：禁止<br>1：使能 |

### 21.5.2 HALLTIM 中断/DMA 使能寄存器（HALLTIM_DIER）

偏移地址：0x04

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | OVR_DMAEN | CHG_DMAEN |
| | | | | | | | | | | | | | | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | SW_IE | OVR_IE | CHG_IE |
| | | | | | | | | | | | | | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:18 | 保留 | 写入无效 |
| 17 | OVR_DMAEN | 计数溢出 DMA 请求使能<br>0：禁止<br>1：使能 |
| 16 | CHG_DMAEN | HALL 信号变化 DMA 请求使能<br>0：禁止<br>1：使能 |
| 15:3 | 保留 | 写入无效 |
| 2 | SW_IE | 软件触发HALL信号捕获使能<br>0：禁止<br>1：使能<br>*注意：该位置1后，对HALLTIM_SR寄存器的SWG位执行写操作，方可触发软件捕获事件。* |
| 1 | OVR_IE | 计数溢出中断使能 |

0：禁止

1：使能

| | | |
|---|---|---|
| 0 | CHG_IE | HALL 信号变化中断使能<br>0：禁止<br>1：使能 |

### 21.5.3 HALLTIM 状态寄存器（HALLTIM_SR）

偏移地址：0x08

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | RAW_DATA[2:0] | | |
| | | | | | | | | | | | | | r | r | r |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | FILT_DATA[2:0] | | | Res. | | | | | SWG | OVR | CHG |
| | | | | | r | r | r | | | | | | w | rc_w1 | rc_w1 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:19 | 保留 | 写入无效 |
| 18:16 | RAW_DATA[2:0] | HALL信号实时值 |
| 15:11 | 保留 | 写入无效 |
| 10:8 | FILT_DATA[2:0] | 滤波后HALL信号捕获值 |
| 7:3 | 保留 | 写入无效 |
| 2 | SWG | 软件触发 HALL 信号变化事件<br>0：不执行任何操作<br>1：触发 HALL 信号变化事件 |
| 1 | OVR | 计数溢出标志<br>写入1可将该位清0。<br>0：未发生计数溢出<br>1：发生计数溢出 |

| | | |
|---|---|---|
| 0 | CHG | HALL 信号变化标志<br>写入1可将该位清0。<br>0：未发生HALL信号变化<br>1：发生 HALL 信号变化 |

### 21.5.4 HALLTIM 宽度计数值寄存器（HALLTIM_WIDTH）

偏移地址：0x0C

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | CAP_CNT[23:16] | | | | | | | |
| | | | | | | | | r | r | r | r | r | r | r | r |

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:24 | 保留 | 写入无效 |
| 23:0 | CAP_CNT[23:0] | HALLTIM宽度计数值 |

### 21.5.5 HALLTIM 自动重载值寄存器（HALLTIM_ARR）

偏移地址：0x10

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | ARR[23:16] | | | | | | | |
| | | | | | | | | 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 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:24 | 保留 | 写入无效 |
| 23:0 | ARR[23:0] | HALLTIM自动重载值 |

### 21.5.6 HALLTIM 计数值寄存器（HALLTIM_CNT）

偏移地址：0x14

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | CNT[23:16] | | | | | | | |
| | | | | | | | | 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 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:24 | 保留 | 写入无效 |
| 23:0 | CNT[23:0] | 计数器值，写入任意值可清0。 |

# 22 低功耗定时器（LPTIM）

## 22.1 简介

LPTIM 是一个 16 位定时器，可选择 LXTAL、RCL、PCLK、MCO 或 LPTIM 输入引脚作为计数源。可在 Stop 模式下工作，具有定时唤醒功能。

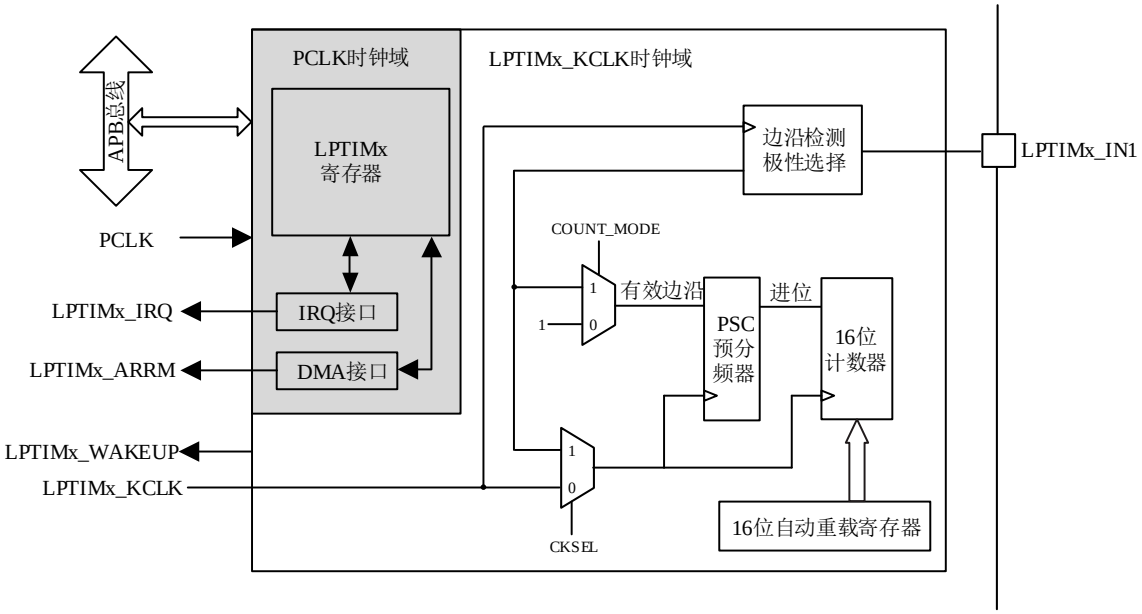
## 22.2 LPTIM 主要特性

- 16 位递增计数器
- 可编程预分频器
- 支持单次计数模式和连续计数模式
- 可选计数源：LXTAL、RCL、PCLK、MCO 或 LPTIM 输入引脚
- 支持低功耗模式唤醒
- 支持 DMA

## 22.3 LPTIMx 功能描述

### 22.3.1 LPTIMx 框图

图 22-1 LPTIMx 框图（x = 1）



### 22.3.2 LPTIM 引脚

表 22-1 LPTIMx 引脚

| 信号名称 | 信号类型 | 描述 |
|---|---|---|
| LPTIMx_IN1 | 输入 | LPTIMx 输入通道 1，GPIO 引脚 |

### 22.3.3 时钟源

LPTIMx 的工作时钟可由下列信号源提供：

- 内部时钟源：LPTIMx_KCLK 可通过*外设异步时钟配置寄存器（RCC_CLKSEL）*中的 LPTIM1_SEL[1:0]位域配置，可选择 LXTAL、RCL、PCLK 或 MCO 作为 LPTIMx 时钟源；
- 外部时钟源：LPTIMx_IN1 输入引脚提供的时钟源。

计数器可对输入到 LPTIMx_IN1 引脚上的外部时钟进行计数，也可对内部时钟进行计数。

**内部时钟源**

由内部时钟源提供工作时钟：LPTIMx_CFG 寄存器中的 CKSEL 位为 0。

- COUNT_MODE 位为 0

  对内部时钟源进行计数，计数器根据每个内部时钟脉冲进行计数。

- COUNT_MODE 位为 1

  对 LPTIMx_IN1 引脚上的外部时钟信号进行计数，计数器可以在外部时钟信号的上升沿、下降沿、上升沿和下降沿均计数。由 LPTIMx_CFG 寄存器中的 CKPOL[1:0]位域来选择外部时钟信号的有效边沿。

  当在外部时钟信号的上升沿或下降沿计数时，外部时钟信号变化频率必须小于内部时钟频率；当在外部时钟信号的上升沿和下降沿均计数时，内部时钟频率应至少为外部时钟信号频率的 4 倍。

  为不丢失任何事件，不得对提供给 LPTIMx 的内部时钟进行预分频（LPTIMx_CFG 寄存器中的 PRESC[2:0]=000）。

**外部时钟源**

由外部时钟源提供工作时钟：LPTIMx_CFG 寄存器中的 CKSEL 位为 1。该配置可在进入低功耗模式后，实现脉冲计数。

对 LPTIMx_IN1 引脚上的外部时钟信号进行计数时，计数器可以在外部时钟信号的上升沿或下降沿进行计数，但不可在上升沿和下降沿均计数。由 LPTIMx_CFG 寄存器中的 CKPOL 位来选择外部时钟信号的有效边沿。

由于外部时钟信号同时用作工作时钟的时钟源，计数器递增计数前存在一些延时，使能定时器后，由于同步问题，外部时钟的前 5 个有效边沿将不会计数。

### 22.3.4 计数器模式

使能 LPTIMx 后，LPTIMx_CR 寄存器中的 SNGSTRT 位或 CNTSTRT 位置位才能生效。

- 连续计数模式：启动计数后，一直计数直到 LPTIMx 被关闭才会停止；
- 单次计数模式：启动计数后，计数计完 ARR 时停止。

LPTIMx 可以在连续计数模式和单次计数模式间切换。如果之前是连续计数模式，将 SNGSTRT 置位会切换为单次计数模式，计数器（激活时）将在计数完 LPTIMx_ARR 寄存器值后立即停止。如果之前是单次计数模式，将 CNTSTRT 置位会切换为连续计数模式。计数器（激活时）将在计数完 LPTIMx_ARR 寄存器值后立即重新启动计数。

若 CNTSTRT、SNGSTRT 位被同时置位（连续计数模式和单次计数模式均有效），此时工作在连续计数模式。

**连续计数模式**

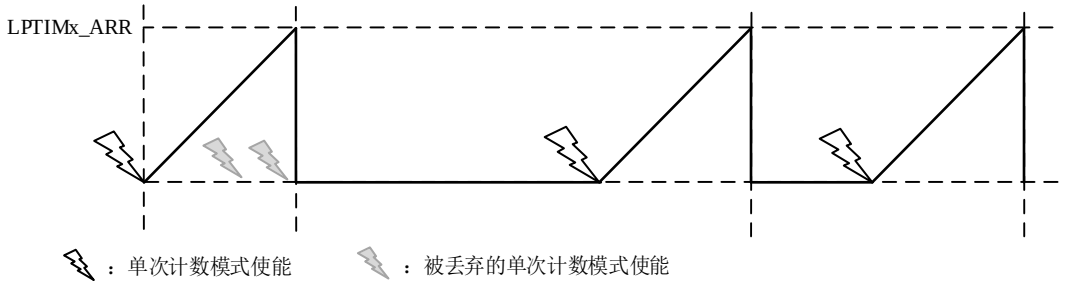
使能连续计数模式计数，需要将 LPTIMx_CR 寄存器中的 CNTSTRT 位置位。CNTSTRT 置位后由硬件立即自动清零。连续计数期间再次将 CNTSTRT 置位，将被丢弃。

若 LPTIMx_CNT 寄存器值与 LPTIMx_ARR 寄存器值匹配，则 LPTIMx_ISR 寄存器中的 ARRM 位置位。ARRM 位由 LPTIMx_ICR 寄存器中的 ARRM_CF 置位清零。

**单次计数模式**

使能单次计数模式计数，需要将 LPTIMx_CR 寄存器中的 SNGSTRT 位置位。SNGSTRT 置位后，由硬件立即自动清零。在 LPTIMx_CNT 计数完 LPTIMx_ARR 寄存器值之前，SNGSTRT 位多次置位将被丢弃，如下图所示。

图 22-2 单次计数模式使能说明




### 22.3.5 定时器使能

配置 LPTIMx_CR 寄存器中的 ENABLE 位，使能或者禁止 LPTIMx。将 ENABLE 位置位后，在 2 个工作时钟周期后使能生效。

LPTIMx_CFG 寄存器必须在禁止 LPTIMx 后才能修改。

当禁止 LPTIMx 时，LPTIMx_CNT 和 LPTIMx_ISR 寄存器被清零。

### 22.3.6 调试模式

当 CPU 进入调试模式（Cortex-M0+内核停止）时，计数器会根据 *APB1 冻结寄存器（DBG_APB1_FZ）*寄存器中的 LPTIMx_HOLD 配置位选择继续正常工作或者停止工作。

## 22.4 LPTIM 低功耗模式

表 22-2 低功耗模式对 LPTIMx 的影响

| 模式 | 说明 |
|---|---|
| Sleep | 无影响，LPTIMx 中断可使芯片退出 Sleep 模式。 |
| Stop | 当 LPTIMx 由 RCL、LXTAL 提供内部时钟源或使用外部时钟源时无影响，LPTIMx 的 ARRM 中断可使芯片退出 Stop 模式。 |

## 22.5 LPTIM 中断

置位 LPTIMx_IER 寄存器中的相关使能位，则以下事件会触发中断/唤醒请求：

表 22-3 中断请求

| 中断事件 | 事件标志 | 中断使能位 | 事件标志/中断清除方法 |
|---|---|---|---|
| 自动重载匹配 | ARRM | ARRM_IE | 向 ARRM_CF 位写 1 |

## 22.6 LPTIM 寄存器

LPTIM 寄存器支持 32 位访问。

表 22-4 LPTIMx 基地址

| 外设 | 基地址 |
|---|---|
| LPTIM1 | 0x4000 7C00 |

### 22.6.1 LPTIM 中断和状态寄存器（LPTIMx_ISR）

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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:2 | 保留 | 写入无效 |
| 1 | ARRM | 自动重载匹配标志<br>当 LPTIMx_CNT 寄存器的值计完 LPTIMx_ARR 寄存器的值时，该位自动置位。如果 LPTIMx_IER 寄存器中的 ARRM_IE 位置位，则触发中断。LPTIMx_ICR 寄存器中 ARRM_CF 位置位，该位清零。<br>0：未发生自动重载匹配事件<br>1：发生自动重载匹配事件<br>*注意：当 ENABLE 为 0，该位自动清 0。* |
| 0 | 保留 | 写入无效 |

### 22.6.2 LPTIM 中断清除寄存器（LPTIMx_ICR）

偏移地址：0x04

复位值：0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:2 | 保留 | 写入无效 |
| 1 | ARRM_CF | 自动重载匹配标志清零<br>该位置位，LPTIMx_ISR寄存器中的ARRM位清零。 |
| 0 | 保留 | 写入无效 |

### 22.6.3 LPTIM 中断使能寄存器（LPTIMx_IER）

偏移地址：0x08

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | ARR_DMAEN | Res. | | | | | | | | | | | ARRM_IE | Res. |
| | | rw | | | | | | | | | | | | rw | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:14 | 保留 | 写入无效 |
| 13 | ARR_DMAEN | 自动重载匹配DMA请求使能<br>0：禁止<br>1：使能 |
| 12:2 | 保留 | 写入无效 |
| 1 | ARRM_IE | 自动重载匹配中断使能位<br>0：禁止<br>1：使能 |

| | | |
|---|---|---|
| 0 | 保留 | 写入无效 |

### 22.6.4 LPTIM 配置寄存器（LPTIMx_CFG）

偏移地址：0x0C

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | PRESC[2:0] | | | Res. | | | | | | CKPOL | | CKSEL |
| | | | | rw | rw | rw | | | | | | | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:24 | 保留 | 写入无效 |
| 23 | COUNT_MODE | 计数源选择<br>0：计数器在工作时钟每个有效边沿递增<br>1：计数器在输入到LPTIMx_IN1上的外部时钟每个有效边沿递增 |
| 22:12 | 保留 | 写入无效 |
| 11:9 | PRESC[2:0] | PSC预分频器的分频系数选择<br>000：1<br>001：2<br>010：4<br>011：8<br>100：16<br>101：32<br>110：64<br>111：128 |
| 8:3 | 保留 | 写入无效 |
| 2:1 | CKPOL[1:0] | 外部时钟源极性选择 |

| | | |
|---|---|---|
| | | 00：上升沿有效<br>01：下降沿有效<br>10：上升沿和下降沿均有效，LPTIMx工作时钟必须选择内部时钟源，且内部时钟源频率大于等于外部时钟频率的4倍<br>11：保留（默认为上升沿有效） |
| 0 | CKSEL | 工作时钟选择<br>0：由内部时钟源提供工作时钟<br>1：由外部时钟源输入到LPTIMx_IN1引脚提供工作时钟 |

### 22.6.5 LPTIM 控制寄存器（LPTIMx_CR）

偏移地址：0x10

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | CNTSTRT | SNGSTRT | ENABLE |
| | | | | | | | | | | | | | w | w | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:3 | 保留 | 写入无效 |
| 2 | CNTSTRT | 连续计数模式使能控制<br>软件置位，硬件自动清零。 |
| 1 | SNGSTRT | 单次计数模式使能控制<br>软件置位，硬件自动清零。 |
| 0 | ENABLE | LPTIMx使能控制<br>0：禁止<br>1：使能 |

### 22.6.6 LPTIM 自动重载寄存器（LPTIMx_ARR）

偏移地址：0x18

复位值：0x0000 0001

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15:0 | ARR[15:0] | 自动重载值<br>仅在 LPTIMx_CR 寄存器中的 ENABLE 位为 1 时，才可修改该位域。 |

### 22.6.7 LPTIM 计数器寄存器（LPTIMx_CNT）

偏移地址：0x1C

复位值：0x0000 0000

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

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

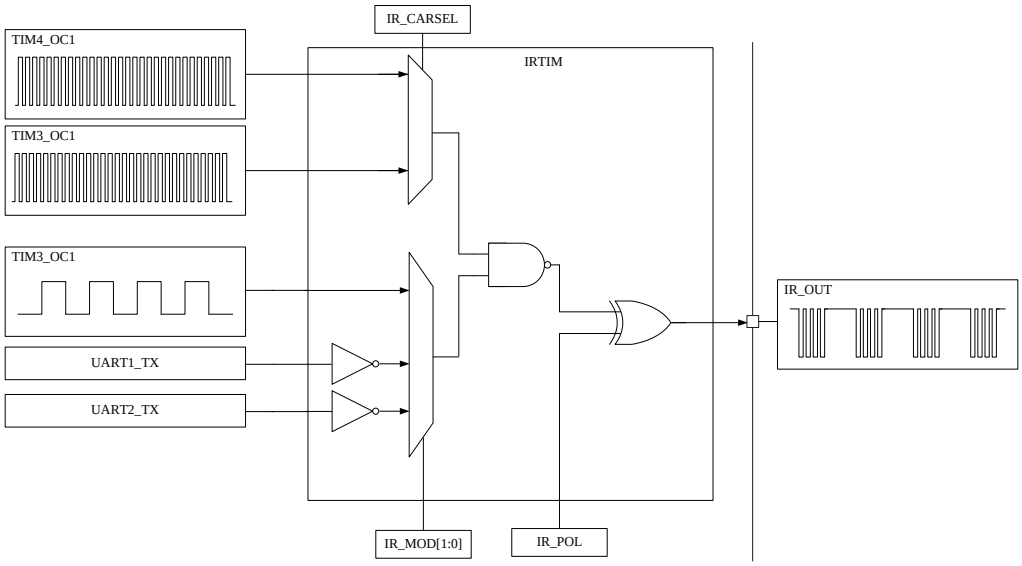
| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15:0 | CNT[15:0] | 计数器值<br>仅当连续两次读取的计数值一致时，读取值才是正确结果。当选用PCLK作为LPTIMx_KCLK时，连续读取两次可能也不能得到正确结果。 |

# 23 红外控制模块（IRTIM）

红外控制模块（IRTIM）能够产生红外控制信号，可以配合红外 LED 发射器一同使用，以实现红外远程控制功能。

IRTIM 在芯片内部与 UART1、UART2、TIM3、TIM4 相连，详见下图。

图 23-1 IRTIM 内部硬件连接




通过配置不同的载波和调制信号，可以产生符合任意红外脉冲调制标准的红外控制信号。

TIM3 定时器的通道 1（TIM3_OC1）和 TIM4 定时器的通道 1（TIM4_OC1）用于产生高频载波信号，可根据 IRTIM 控制寄存器（IRTIM_CR）中的 IR_CARSEL 位选择。

通过配置 IRTIM 控制寄存器（IRTIM_CR）中的 IR_MODE[1:0]位域，可以选择 TIM3 定时器的通道 1（TIM3_OC1）、UART1 或 UART2 中的一路信号作为调制信号。

IRTIM 输出信号的极性，由 IRTIM 控制寄存器（IRTIM_CR）中的 IR_POL 控制位进行选择，可以根据需要将输出极性反转。

IRTIM 产生的红外控制信号由 IR_OUT 引脚输出，通过 GPIO 引脚的复用功能进行配置。

## 23.1 IRTIM 寄存器

IRTIM 寄存器支持 32 位访问。

表 23-1 IRTIM 基地址

| 外设 | 基地址 |
|---|---|
| IRTIM | 0x4001 0000 |

### 23.1.1 IRTIM 控制寄存器（IRTIM_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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | IR_CARSEL | IR_MODE[1:0] | | IR_POL | Res. | |
| | | | | | | | | | | rw | rw | rw | rw | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:6 | 保留 | 写入无效 |
| 5 | IR_CARSEL | IRTIM载波信号选择<br>0：TIM4_OC1<br>1：TIM3_OC1 |
| 4:3 | IR_MODE[1:0] | IRTIM调制信号选择<br>00：TIM3_OC1<br>01：UART1_TX<br>10：UART2_TX<br>11：保留 |
| 2 | IR_POL | IR_OUT输出信号极性选择<br>0：IRTIM（IR_OUT）输出信号未反相<br>1：IRTIM（IR_OUT）输出信号反相 |
| 1:0 | 保留 | 写入无效 |

# 24 独立看门狗（IWDG）

## 24.1 简介

芯片内置独立看门狗，可用于检测并解决芯片发生的异常，并在计数器由给定的超时值递减到 0 时触发系统复位。

独立看门狗由内部低速时钟 RCL 驱动，因此在系统时钟发生故障时仍然保持工作状态，提高系统可靠性。

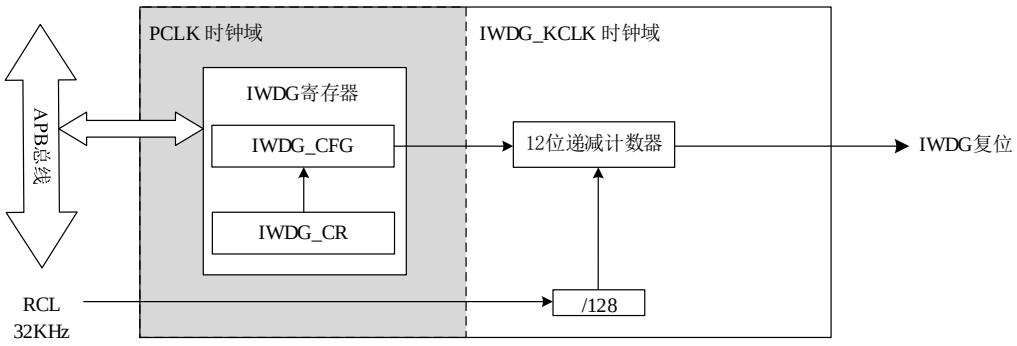
## 24.2 IWDG 主要特性

- 12 位递减计数器
- 工作时钟由 RCL 提供，可在 Stop 模式下工作
- 当达到溢出时间时复位

## 24.3 IWDG 功能描述

### 24.3.1 IWDG 框图

图 24-1 IWDG 框图




IWDG 功能在 Stop 模式下仍能工作。

通过将 0x0000 CCCC 写到控制寄存器（IWDG_CR）来使能独立看门狗时，计数器从复位值（0xFFF）开始递减计数。当计数达到 0x000 时会产生 IWDG 复位。

IWDG 使能后任何时候将 0x0000 AAAA 写到控制寄存器（IWDG_CR）中，配置寄存器（IWDG_CFG）中设置的计数溢出时间值将被重载到计数器中，从而避免产生看门狗复位。

IWDG 一旦运行，其时钟源（RCL）无法停止；IWDG 使能后，RCL 自动打开。

### 24.3.2 寄存器访问保护

配置寄存器（IWDG_CFG）具有写访问保护功能。若要修改这些寄存器，用户必须首先对控制寄存器（IWDG_CR）写入 0x0000 5555，而写入其他值则会破坏该序列，从而使寄存器访问保护再次生效。

### 24.3.3 溢出时间设置

IWDG 内部预分频为 128，分频后的计数溢出时间共 8 个可用档位，溢出时间与计数周期之间的关系如下：

$t_{IWDG} = t_{RCL} * 128 * N$

表 24-1 IWDG 溢出周期表

| RCL | 计数周期（N） | 溢出时间（ms） |
|---|---|---|
| 32KHz | 32 | 128 |
| | 64 | 256 |
| | 128 | 512 |
| | 256 | 1024 |
| | 512 | 2048 |
| | 1024 | 4096 |
| | 2048 | 8192 |
| | 4096 | 16384 |

实时修改溢出时间，可按以下步骤配置：

1) 确保 IWDG 正在运行；
2) 将 0x0000 5555 写入控制寄存器（IWDG_CR），解除写保护；
3) 修改配置寄存器（IWDG_CFG）中的计数溢出时间；
4) 将 0x0000 AAAA 写入控制寄存器（IWDG_CR），进行喂狗操作。

### 24.3.4 调试模式

当芯片进入调试模式时，IWDG 计数器会根据 DBG 模块中的 *APB1 冻结寄存器（DBG_APB1_FZ）*中的 IWDG_HOLD 选择继续计数或者停止计数。

## 24.4 IWDG 低功耗模式

表 24-2 低功耗模式对 IWDG 的影响

| 模式 | 说明 |
| --- | --- |
| Sleep | 无影响 |
| Stop | 无影响，IWDG 由 RCL 提供时钟源，依据 IWDG 配置寄存器（IWDG_CFG）的 LP_STOP 位进行计数或停止计数。 |

## 24.5 IWDG 寄存器

IWDG 寄存器支持 32 位访问。

表 24-3 IWDG 基地址

| 外设 | 基地址 |
|---|---|
| IWDG | 0x4000 3000 |

### 24.5.1 IWDG 控制寄存器（IWDG_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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

| 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 |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15:0 | KEY[15:0] | 控制值<br>必须定期对这些位写入 0x0000 AAAA，使递减计数器重新载入超时值；否则递减到 0 时，看门狗会产生复位。<br>写入0x0000 5555可使能对配置寄存器（IWDG_CFG）的访问。<br>写入 0x0000 CCCC 可使能看门狗。 |

### 24.5.2 IWDG 配置寄存器（IWDG_CFG）

偏移地址：0x04

复位值：0x0000 0007

*注意：* *该寄存器默认处于写保护状态，相关内容详见：寄存器访问保护。*

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

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

| LP_STOP | Res. | | | | | | | | | | | OVP[2:0] | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| rw | | | | | | | | | | | | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15 | LP_STOP | 独立看门狗在Stop模式下计数器停止控制<br>0：计数器正常计数<br>1：计数器停止运行 |
| 14:3 | 保留 | 写入无效 |
| 2:0 | OVP[2:0] | 计数溢出时间选择<br>000：128ms<br>001：256ms<br>010：512ms<br>011：1.024s<br>100：2.048s<br>101：4.096s<br>110：8.192s<br>111：16.384s |

### 24.5.3 IWDG 计数值寄存器（IWDG_CNT）

偏移地址：0x14

复位值：0x0000 0FFF

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:12 | 保留 | 写入无效 |
| 11:0 | CNT[11:0] | 当前计数值 |

仅当连续两次读取的计数值一致时，读取值才是正确结果。

# 25 SysTick 定时器（SysTick）

## 25.1 简介

ARM Cortex-M0+内核提供了一个定时器 SysTick。

## 25.2 SysTick 主要特性

- 24 位计数器
- 递减计数
- 提供异常处理中断

## 25.3 SysTick 功能描述

SysTick 是一个 24 位定时器。通过将 SysTick_CTRL 寄存器的 ENABLE 位置 1 使能时，计数器从当前计数值寄存器 SysTick_VAL 的值开始递减计数，每当减到 0 时，会在下一个时钟沿重新装载 SysTick_LOAD 寄存器的值到 SysTick_VAL 中，并再次进行递减计数。

每次计数到 0 时，SysTick_CTRL 的 COUNTFLAG 标志位将置位，读此位时标志清零。

向 SysTick_VAL 寄存器写入任意值时，SysTick_VAL 寄存器和 COUNTFLAG 标志位都将清零，在下一个时钟沿 SysTick_VAL 寄存器会装载 SysTick_LOAD 中的计数值，重新开始计数，此过程不会触发中断。读 SysTick_VAL 寄存器时返回值为当前计数器的值。

向 SysTick_LOAD 寄存器中写“0”时，计数器在当前计数完成后停止计数，停止后计数器的值保持为“0”。

在首次使能 SysTick 前首先写入 SysTick_LOAD 寄存器，再写入 SysTick_VAL。这样的操作顺序保证了正确的重载值被写入，并同时清空 SysTick_VAL。之后 SysTick 使能时直接从 SysTick_LOAD 开始计数。

当芯片处于调试模式内核停止时，SysTick 也会停止计数。

## 25.4 SysTick 寄存器

### 25.4.1 SysTick 控制和状态寄存器（SysTick_CTRL）

地址：0xE000 E010

复位值：0x0000 0004

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | COUNT FLAG |
| | | | | | | | | | | | | | | | r |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | CLK SOURCE | TICK INT | ENABLE |
| | | | | | | | | | | | | | r | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:17 | 保留 | 写入无效 |
| 16 | COUNTFLAG | 溢出标志位<br>上一次读SysTick_CTRL寄存器后，SysTick是否出现过溢出（计数到0）。<br>此位在计数值从1减为0时置位，读取该位或者SysTick_VAL寄存器写入任意值后，该标志位自动清零。<br>0：未发生过溢出<br>1：发生过溢出 |
| 15:3 | 保留 | 写入无效 |
| 2 | CLKSOURCE | SysTick时钟源选择<br>无外部参考时钟，计数时钟源始终为内核时钟HCLK。<br>读为1，写无效。 |
| 1 | TICKINT | SysTick中断使能<br>0：禁止<br>1：使能 |
| 0 | ENABLE | SysTick使能计数<br>0：禁止<br>1：使能 |

### 25.4.2 SysTick 重载值寄存器（SysTick_LOAD）

地址：0xE000 E014

复位值：0x00XX XXXX

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | RELOAD[23:16] | | | | | | | |
| | | | | | | | | rw | rw | rw | rw | rw | rw | rw | rw |

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:24 | 保留 | 写入无效 |
| 23:0 | RELOAD[23:0] | SysTick的重载计数值<br>当计数值递减到0时，该值装载到SysTick_VAL寄存器。 |

### 25.4.3 SysTick 当前计数值寄存器（SysTick_VAL）

地址：0xE000 E018

复位值：0x00XX XXXX

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | CURRENT[23:16] | | | | | | | |
| | | | | | | | | rw | rw | rw | rw | rw | rw | rw | rw |

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:24 | 保留 | 写入无效 |
| 23:0 | CURRENT[23:0] | 计数器当前值<br>写任意值到该寄存器，该寄存器和SysTick_CTRL寄存器的COUNTFLAG标志都清零。 |

### 25.4.4 SysTick 校准值寄存器（SysTick_CALIB）

地址：0xE000 E01C

复位值：0x8001 193F

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NOREF | SKEW | Res. | | | | | | TENMS[23:16] | | | | | | | |
| r | r | | | | | | | r | r | r | r | r | r | r | r |

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31 | NOREF | 计数时钟标志<br>1：SysTick总是使用内核时钟 |
| 30 | SKEW | TENMS精度指示<br>0：TENMS校准值代表精确的1ms |
| 29:24 | 保留 | 写入无效 |
| 23:0 | TENMS[23:0] | 1ms校准计数值0x1193F<br>时钟源HCLK配置为72MHz时，SysTick计数间隔为1ms的计数值。 |

# 26 实时时钟（RTC）

## 26.1 简介

实时时钟 RTC，提供日历计时功能，采用 BCD 码格式记录年、月、日、时、分、秒、星期等信息，并包含有独立可配置的闹钟和周期定时中断功能，支持在所有低功耗模式下的运行和唤醒。内置时钟数字平滑校准功能，能显著提高 RTC 日历的长期计时精度。

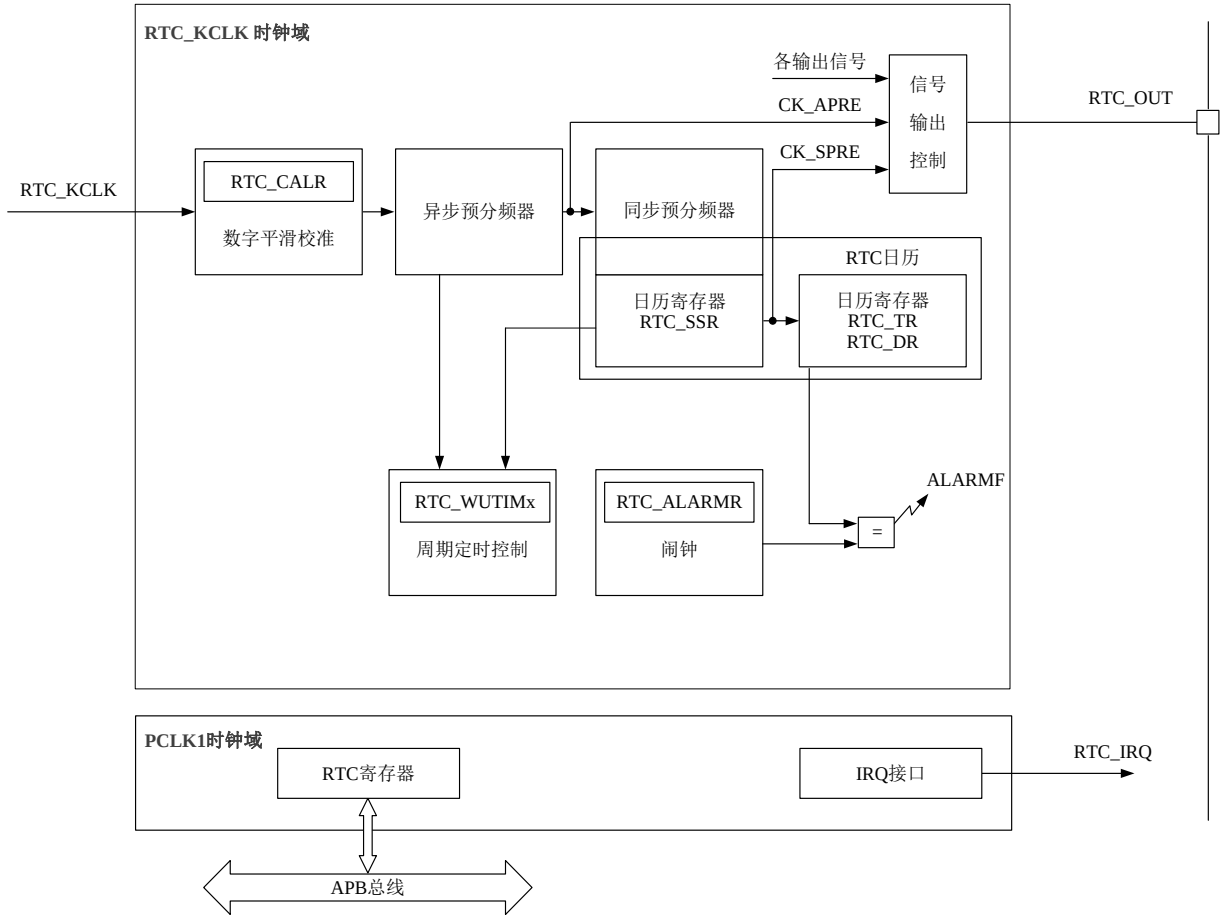
## 26.2 RTC 主要特性

- 时钟数字平滑校准，精度±0.477ppm
- 日历计时功能（BCD 码）
- 日历能根据闰年、月份自动调整日期
- 1 个可配置的闹钟
- 两个从毫秒到月的可配置周期定时
- 内部时钟信号可通过引脚输出（可用于温度校准误差的测量）

## 26.3 RTC 功能描述

### 26.3.1 RTC 框图

图 26-1 RTC 框图




### 26.3.2 RTC 引脚

表 26-1 RTC 输出引脚

| 引脚名称 | 类型 | 说明 |
|---|---|---|
| RTC_OUT | 输出 | 输出内部时钟和标志信号 |

通过配置 RTC 控制寄存器（RTC_CR）中的 OUT_SEL[2:0]位域，选择以下信号作为 RTC_OUT 功能的输出源：

- CK_APRE 时钟信号：异步预分频器的输出时钟，频率为 256Hz 信号；
- CK_SPRE 时钟信号：同步预分频器的输出时钟，频率为 1Hz 信号；
- ALARMF 标志信号：闹钟匹配事件的标志状态；

- 日历计时进位信息：日历天、小时、分钟的进位信号；
- 校准周期信号：数字平滑校准周期信号。

在配置输出信号后，通过将 RTC_CR 寄存器中的 OUTEN 位置 1，并将 GPIO 复用为 RTC_OUT 功能可以从 GPIO 引脚输出 RTC_OUT 信号。

通过设置 RTC_CR 寄存器中的 POL 位，配置 RTC_OUT 输出信号的极性，如果 POL 位被置 1，则输出信号的极性与信号的实际状态相反。

### 26.3.3 RTC 寄存器写保护

为了防止误操作，RTC 模块内部具有写保护功能。当发生 AWON 域复位后，RTC 将处于写保护状态，除 RTC 写保护寄存器（RTC_WPR）与 RTC 状态清除寄存器（RTC_CLR）外，对其它 RTC 寄存器的写入操作都将无效。

为解除 RTC 寄存器的写保护状态，必须连续顺序执行以下步骤：

1) 向 RTC_WPR 寄存器写入密钥 1：0xCA。
2) 向 RTC_WPR 寄存器写入密钥 2：0x53。

写保护解除后，如果再向 RTC_WPR 寄存器写入任意值（包括密钥 1：0xCA），RTC 寄存器将重新进入写保护状态。

### 26.3.4 RTC 时钟和分频

RTC 时钟（RTC_KCLK）可以在以下时钟源中进行选择：

- LXTAL 时钟 32.768KHz；
- RCL 时钟 32KHz。

通过配置*AWON 域控制寄存器（RCC_AWCR）*中的 RTCSEL[1:0]位域，对 RTC_KCLK 时钟进行选择，推荐采用频率为 32.768KHz 的 LXTAL 时钟，以获得高精度的 RTC 日历计时，相关内容参见：*RTC 时钟*。

RTC 采用内部预分频器对 RTC_KCLK 时钟分频，产生 1Hz 的时钟信号作为 RTC 日历的计数时钟，RTC 内部有 2 个预分频器：

- 异步预分频器：对 RTC_KCLK 时钟进行分频，分频后输出 CK_APRE 时钟信号，频率为 256Hz；
- 同步预分频器：对 CK_APRE 时钟进行分频，分频后输出 CK_SPRE 时钟信号，频率为 1Hz。

其中同步预分频器由递增计数器构成，计数值用以表示日历的亚秒。通过 RTC 亚秒寄存器（RTC_SSR）中的 SS[7:0]位域，可以读取同步预分频器的当前计数值，单位为 3.91ms（1/256 s）。

### 26.3.5 RTC 日历的读取

RTC 日历采用 BCD 码格式，可提供年、月、日、时、分、秒、星期等信息，并根据年、月自动调整日历中的日期，例如月末的 30 日或 31 日、闰年的 2 月 29 日等。

通过以下 RTC 日历寄存器，可以获取 RTC 日历信息：

- RTC_SSR 亚秒寄存器；
- RTC_TR 时间寄存器；
- RTC_DR 日期寄存器。

更新日历的时钟 RTC_KCLK 和读取日历寄存器的时钟 PCLK1 是异步的，推荐时间读取方式如下：

1) 连续读取相关日历寄存器；
2) 再次连续读取相关日历寄存器；
3) 比较两次读取结果，如果内容一致则结果正确，否则重复以上步骤。

RTC 日历能在 2000-2099 年范围内自动判断闰年，并调整相关日期。当 RTC 日历计数到达 99 年 12 月 31 日 23:59:59 时，将发生日历溢出。

### 26.3.6 RTC 日历的配置

进入 RTC 初始化模式后，才能对 RTC 日历进行配置，配置流程如下：

1) 配置 RTC_CSR 寄存器中的 MODSEL 位置 1，进入 RTC 初始化模式；
2) 查询 RTC_CSR 寄存器中的 MODSTA 位。当 MODSTA 位置 1 时，表明 RTC 已经进入初始化模式，此时 RTC 内部的各计数器都将停止工作；
3) 向 RTC 日历寄存器写入要配置的时间和日期；
4) 配置 RTC_CSR 寄存器中的 MODSEL 位清 0，退出 RTC 初始化模式，RTC_CSR 寄存器中的 MODSTA 标志将自动清 0。日历将加载寄存器中的值，并在 1 个 RTC_KCLK 时钟周期后恢复计时。

读取 RTC 亚秒寄存器（RTC_SSR）可以获取同步预分频器的当前计数值。进入初始化模式时，RTC 异步预分频器中的计数值将被清除，以实现日历亚秒级对

时。

### 26.3.7 RTC 闹钟

通过将 RTC_CR 寄存器中的 ALARME 位置 1，使能 RTC 闹钟。

闹钟使能后，如果 RTC 日历寄存器中的相关参数（秒、分钟、小时），与 RTC_ALARMR 寄存器中的对应参数相等，则触发闹钟匹配事件，RTC_SR 寄存器中的 ALARMF 标志将置 1。如果此时 RTC_CR 寄存器中的 ALARMIE 位已经置 1，则 RTC_ISR 寄存器中的 ALARMIF 中断标志也将置 1，并触发中断。

通过 RTC_ALARMR 寄存器中的 MSKx 位可对闹钟参数进行屏蔽，以选择哪些参数不参与闹钟的匹配比较，示例如下表：

表 26-2 定时闹钟举例

| 定时要求 | RTC_ALARMR 值 | 闹钟时刻 |
|---|---|---|
| 早上 7:30 闹钟 | 0x0007 3000 | 07:30:00 |
| 每小时的 20 分发生一次 | 0x0080 2000 | xx:20:00 |
| 上午 9 点每分钟的 10 秒发生一次 | 0x0009 8010 | 09:xx:10 |
| 中午 12 点 30 分每秒钟发生一次 | 0x0012 3080 | 12:30:xx |
| 每分钟的 10 秒发生一次 | 0x0080 8010 | xx:xx:10 |
| 下午 16 点每秒钟发生一次 | 0x0016 8080 | 16:xx:xx |

RTC 闹钟配置步骤如下：

1) 将 RTC_CR 寄存器中的 ALARME 位清 0，禁止 RTC 闹钟运行；
2) 配置闹钟寄存器 RTC_ALARMR；
3) 将 RTC_CR 寄存器中的 ALARME 位置 1，使能 RTC 闹钟。

### 26.3.8 RTC 周期定时

RTC 支持两个独立的周期定时，通过将 RTC_CR 寄存器中的 WUTIMxE 位置 1，使能 RTC 周期定时功能。

通过配置 RTC 周期定时寄存器（RTC_WUTIMx）中的 PERIOD 位，可以选择 1024Hz、256Hz、64Hz、16Hz、8Hz、4Hz、2Hz、1 秒钟、1 分钟、1 小时、1 天、1 个月作为定时周期。每当到达计时周期，RTC 状态寄存器（RTC_SR）中的 WUTIMxF 位将自动置 1。若 RTC_CR 中对应的 WUTIMxIE 位已经置 1，则 RTC 中断状态寄存器（RTC_ISR）中的 WUTIMxIF 位也将置 1，并触发中断。

RTC 周期定时功能配置步骤如下：

1) 将 RTC_CR 寄存器中的 WUTIMxE 位清 0，禁止 RTC 周期定时功能运行；
2) 配置 RTC_WUTIMx 中的 PERIOD 位，选择需要的定时周期；
3) 将 RTC_CR 寄存器中的 WUTIMxE 位置 1，使能 RTC 周期定时。

### 26.3.9 RTC 时钟数字平滑校准

为了提高 RTC 日历的走时精度，可以对 RTC_KCLK 频率进行数字校准，校准范围可达-487.1ppm 到+488.5ppm，校准精度可达±0.477ppm。

通过 ADC 温度传感器采集温度信息，可利用数字平滑校准功能实现对 RTC 的温度补偿，以提高不同温度下的走时精度。

**RTC 校准周期**

RTC 时钟使能后，数字平滑校准功能将持续运行，校准周期为 $2^{20}$ 个 RTC_KCLK 时钟周期，当 RTC 时钟频率为 32.768KHz 时，校准周期为 32 秒（$2^{20}$/32768），校准分辨率可达 0.954ppm（$1/2^{20}$）。

校准周期由一个 20 位的计数器 CAL_CNT 进行控制，每个校准周期结束时，RTC_SR 寄存器中的 CALF 标志将置 1。若 RTC_CR 寄存器中的 CALIE 位已经置 1，则 RTC_ISR 寄存器中的 CALIF 中断标志也将置 1，并触发中断。

**RTC 校准参数**

通过配置 RTC_CALR 寄存器中的 CALM[8:0]位域，最多可减少 487.1ppm（$511/2^{20}$）。

通过配置 RTC_CALR 寄存器中的 CALP 位，最多可增加 488.5ppm（$512/2^{20}$）。

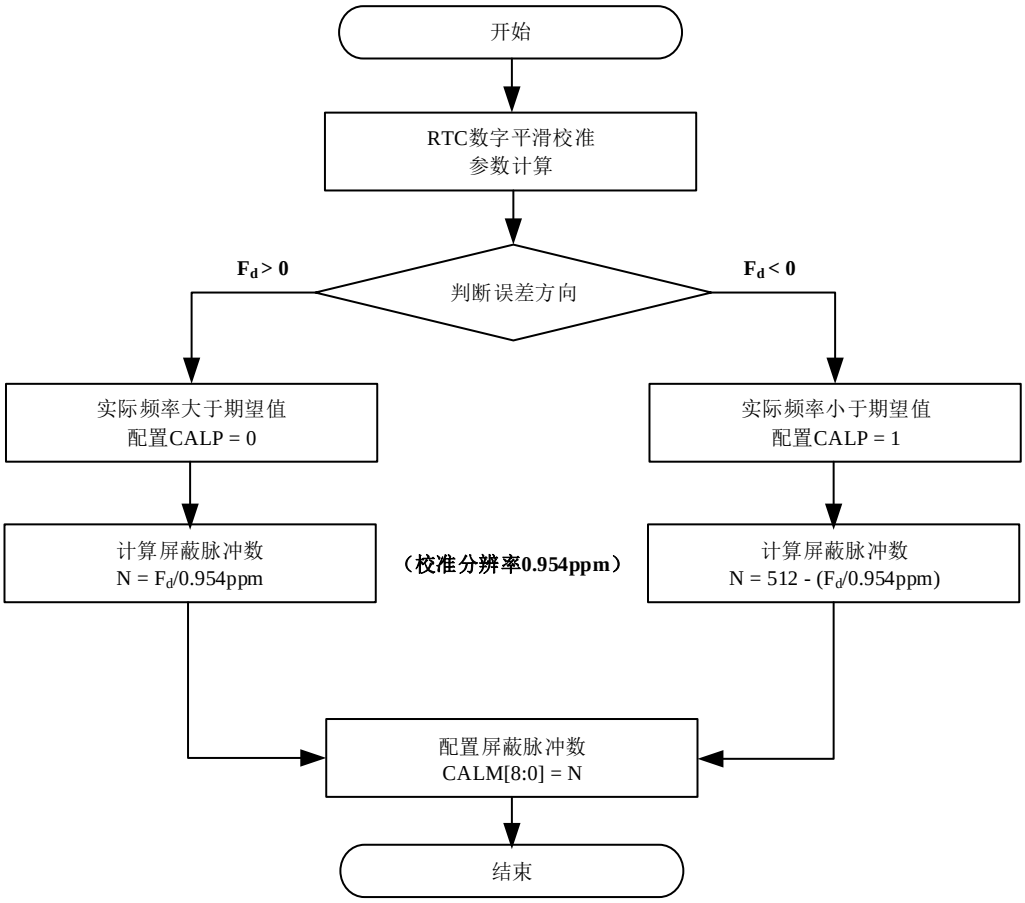
同时配置 CALM[8:0]位域和 CALP 位，则在每个校准周期内，RTC 时钟频率的校准范围为：-487.1ppm 到+488.5ppm。

**RTC 校准参数的计算**

使用时，先测量 RTC 时钟（RTC_KCLK）的频率误差。测量 RTC 时钟频率误差时，请先将 RTC_CALR 寄存器中的校准参数配置为 0，并通过配置 RTC_OUT 引脚，输出异步或同步预分频器的时钟信号（CK_APRE 或 CK_SPRE）。

测量 RTC_OUT 输出时钟信号的频率，计算 RTC 时钟的频率误差$\boldsymbol{F_d}$，将频率误差的单位换算成 ppm 的形式，并按照如下流程计算校准参数：

图 26-2 校准参数计算流程




以下给出几种不同 RTC 时钟频率误差$F_d$下的校准参数计算示例：

- 示例 1：频率误差 $F_d$=20ppm：

  1) 误差大于 0，时钟频率快，需要屏蔽脉冲；
  2) 配置 CALP=0；（不插入脉冲）
  3) 配置 CALM[8:0]= 20/0.954 = 20.964 ≈ 21；

- 示例 2：频率误差 $F_d$=-20ppm：

  1) 误差小于 0，时钟频率慢，需要插入脉冲；
  2) 配置 CALP=1；（先插入 512 脉冲）
  3) 配置 CALM[8:0] = 512 -（20/0.954）≈ 491。

**RTC 动态校准**

RTC 时钟数字平滑校准的参数，可以在 RTC 运行时动态配置，配置后的校准参数将持续有效，具体步骤如下：

1) 查询 RTC 控制与状态寄存器（RTC_CSR）中的 CALSTA 标志；
2) 当 CALSTA 标志为 0 时，可以向 RTC_CALR 寄存器写入校准参数；

3) 对 RTC_CALR 寄存器执行写入操作后，CALSTA 标志将自动置 1，生效后 CALSTA 标志将自动清 0。

### 26.3.10 调试模式

当芯片处于调试模式，内核停止工作时，如果要暂停 RTC 的计数，可将 *APB1 冻结寄存器（DBG_APB1_FZ）* 中的 RTC_HOLD 位配置为 1。

## 26.4 RTC 低功耗模式

在所有功耗模式下，RTC 均可正常工作，中断可唤醒 Sleep、Stop 模式。

RTC 各功能不受 NRST 复位和系统复位的影响，发生 NRST 复位和系统复位时，RTC 仍能正常运行。

## 26.5 RTC 中断

如果 RTC 中断状态寄存器（RTC_ISR）中的标志已置 1，则表明对应的中断请求已产生。

表 26-3 中断请求

| 中断事件 | 中断标志 | 中断使能位 | 事件标志/中断清除方法 |
|---|---|---|---|
| 闹钟匹配 | ALARMIF | ALARMIE | 向 CALARMF 位写 1 |
| 校准周期 | CALIF | CALIE | 向 CCALF 位写 1 |
| 周期定时 | WUTIM1IF | WUTIM1E | 向 CWUTIM1F 位写 1 |
| | WUTIM2IF | WUTIM2E | 向 CWUTIM2F 位写 1 |

## 26.6 RTC 寄存器

RTC 寄存器支持 32 位访问。

表 26-4 RTC 基地址

| 外设 | 基地址 |
|---|---|
| RTC | 0x4000 2800 |

### 26.6.1 RTC 时间寄存器（RTC_TR）

偏移地址：0x00

复位值：0x0000 0000

*注意：* *该寄存器仅可被 AWON 域复位，并默认处于写保护状态，相关内容详见：RTC 寄存器写保护。*

*仅在 RTC 初始化模式下，对该寄存器的写入操作有效。*

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | HT[1:0] | | HU[3:0] | | | |
| | | | | | | | | | | rw | rw | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | MNT[2:0] | | | MNU[3:0] | | | | Res. | ST[2:0] | | | SU[3:0] | | | |
| | rw | rw | rw | rw | rw | rw | rw | | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:22 | 保留 | 写入无效 |
| 21:20 | HT[1:0] | 日历小时的十位（BCD格式） |
| 19:16 | HU[3:0] | 日历小时的个位（BCD 格式） |
| 15 | 保留 | 写入无效 |
| 14:12 | MNT[2:0] | 日历分钟的十位（BCD格式） |
| 11:8 | MNU[3:0] | 日历分钟的个位（BCD格式） |
| 7 | 保留 | 写入无效 |

| | | |
|---|---|---|
| 6:4 | ST[2:0] | 日历秒钟的十位（BCD格式） |
| 3:0 | SU[3:0] | 日历秒钟的个位（BCD格式） |

### 26.6.2 RTC 日期寄存器（RTC_DR）

偏移地址：0x04

复位值：0x0000 2101

*注意：* *该寄存器仅可被 AWON 域复位，并默认处于写保护状态，相关内容详见：RTC 寄存器写保护。*

*仅在 RTC 初始化模式下，对该寄存器的写入操作有效。*

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | YT[3:0] | | | | YU[3:0] | | | |
| | | | | | | | | rw | rw | rw | rw | rw | rw | rw | rw |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| WDU[2:0] | | | MT | MU[3:0] | | | | Res. | | DT[1:0] | | DU[3:0] | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | | | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:24 | 保留 | 写入无效 |
| 23:20 | YT[3:0] | 日历年份的十位（BCD格式） |
| 19:16 | YU[3:0] | 日历年份的个位（BCD格式） |
| 15:13 | WDU[2:0] | 日历的星期<br>000：禁止<br>001：星期一<br>⋮<br>111：星期日 |
| 12 | MT | 日历月份的十位（BCD格式） |
| 11:8 | MU[3:0] | 日历月份的个位（BCD格式） |

| | | |
|---|---|---|
| 7:6 | 保留 | 写入无效 |
| 5:4 | DT[1:0] | 日历日期的十位（BCD格式） |
| 3:0 | DU[3:0] | 日历日期的个位（BCD格式） |

### 26.6.3 RTC 亚秒寄存器（RTC_SSR）

偏移地址：0x08

复位值：0x0000 0000

*注意：* *该寄存器仅可被 AWON 域复位，并默认处于写保护状态，相关内容详见：RTC 寄存器写保护。*

*仅在 RTC 初始化模式下，对该寄存器的写入操作有效。*

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 写入无效 |
| 7:0 | SS[7:0] | 日历亚秒值<br>亚秒是秒的小数部分<br>单位为：3.91ms（1/256 s）。 |

### 26.6.4 RTC 控制与状态寄存器（RTC_CSR）

偏移地址：0x0C

复位值：0x0000 0000

*注意：* *AWON 域复位可复位该寄存器。NRST 复位和系统复位仅可复位 MODSEL 和 MODSTA 位。*

*该寄存器默认处于写保护状态，相关内容详见：RTC 寄存器写保护。*

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

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

| Res. | | | | | | | | MODSEL | MODSTA | Res. | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | rw | r | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:17 | 保留 | 写入无效 |
| 16 | CALSTA | RTC 时钟数字平滑校准状态<br>当对 RTC 校准寄存器（RTC_CALR）执行写操作时，该标志将自动置 1，以指示新的校准参数已写入。<br>当写入的校准参数生效后，该标志将自动清 0。<br>0：校准参数已生效，可以写入新的校准参数<br>1：校准参数已写入，但还未生效 |
| 15:8 | 保留 | 写入无效 |
| 7 | MODSEL | RTC 模式选择<br>0：退出 RTC 初始化模式<br>1：进入 RTC 初始化模式<br>进入 RTC 初始化模式后，RTC 内部各计数器都将停止计数。 |
| 6 | MODSTA | RTC 模式状态<br>当该位置 1 时，表明 RTC 已处于初始化模式，此时可对 RTC 日历相关寄存器进行配置。<br>0：RTC 正常运行<br>1：RTC 处于初始化模式 |
| 5:0 | 保留 | 写入无效 |

### 26.6.5 RTC 控制寄存器（RTC_CR）

偏移地址：0x18

复位值：0x0000 0000

*注意：* *该寄存器仅可被 AWON 域复位，并默认处于写保护状态，相关内容详见 RTC 寄存器写保护。*

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | WUTIM2IE | WUTIM1IE | Res. | | | OUT_SEL[2:0] | | | POL | OUTEN | Res. | | |
| | | | rw | rw | | | | rw | rw | rw | rw | rw | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | CALIE | ALARMIE | Res. | WUTIM2E | WUTIM1E | ALARME | Res. | | | | | | | |
| | | rw | rw | | rw | rw | rw | | | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:29 | 保留 | 写入无效 |
| 28 | WUTIM2IE | 周期定时2中断使能控制<br>0：禁止<br>1：使能 |
| 27 | WUTIM1IE | 周期定时1中断使能控制<br>0：禁止<br>1：使能 |
| 26:24 | 保留 | 写入无效 |
| 23:21 | OUT_SEL[2:0] | RTC_OUT输出信号选择<br>000：输出异步预分频器的输出时钟信号（CK_APRE：256Hz）<br>001：输出同步预分频器的输出时钟信号（CK_SPRE：1Hz）<br>010：输出闹钟匹配状态（ALARMF）<br>011：保留（无信号输出）<br>100：输出日历分钟进位信号<br>101：输出日历小时进位信号<br>110：输出日历天进位信号<br>111：输出校准周期信号 |
| 20 | POL | RTC_OUT输出极性选择<br>0：输出极性与信号状态相同<br>1：输出极性与信号状态相反 |
| 19 | OUTEN | RTC_OUT输出使能控制<br>0：禁止<br>1：使能 |

| | | |
|---|---|---|
| 18:14 | 保留 | 写入无效 |
| 13 | CALIE | 校准周期中断使能控制<br>0：禁止<br>1：使能 |
| 12 | ALARMIE | 闹钟匹配中断使能控制<br>0：禁止<br>1：使能 |
| 11 | 保留 | 写入无效 |
| 10 | WUTIM2E | 周期定时2使能控制<br>0：禁止<br>1：使能 |
| 9 | WUTIM1E | 周期定时1使能控制<br>0：禁止<br>1：使能 |
| 8 | ALARME | 闹钟使能控制<br>0：禁止<br>1：使能 |
| 7:0 | 保留 | 写入无效 |

### 26.6.6 RTC 周期定时寄存器 1（RTC_WUTIM1）

偏移地址：0x1C

复位值：0x0000 0000

*注意：* *该寄存器仅可被 AWON 域复位，并默认处于写保护状态，相关内容详见：RTC 寄存器写保护。*

*仅在 RTC_CR 寄存器中的 WUTIM1E 位为 0 时，对该寄存器的写操作有效。*

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:4 | 保留 | 写入无效 |
| 3:0 | PERIOD[3:0] | 定时周期设置<br>0000：1024Hz<br>0001：256Hz<br>0010：64Hz<br>0011：16Hz<br>0100：8Hz<br>0101：4Hz<br>0110：2Hz<br>0111：1秒<br>1000：1分钟<br>1001：1小时<br>1010：1天<br>1011：1个月<br>其他：保留（默认为1个月） |

### 26.6.7 RTC 周期定时寄存器 2（RTC_WUTIM2）

偏移地址：0x20

复位值：0x0000 0000

*注意：* *该寄存器仅可被 AWON 域复位，并默认处于写保护状态，相关内容详见：RTC 寄存器写保护。*

*仅在 RTC_CR 寄存器中的 WUTIM2E 位为 0 时，对该寄存器的写操作有效。*

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:4 | 保留 | 写入无效 |
| 3:0 | PERIOD[3:0] | 定时周期设置<br>0000：1024Hz<br>0001：256Hz<br>0010：64Hz<br>0011：16Hz<br>0100：8Hz<br>0101：4Hz<br>0110：2Hz<br>0111：1秒<br>1000：1分钟<br>1001：1小时<br>1010：1天<br>1011：1个月<br>其他：保留（默认为1个月） |

### 26.6.8 RTC 写保护寄存器（RTC_WPR）

偏移地址：0x24

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 写入无效 |
| 7:0 | KEY[7:0] | 用于解锁RTC寄存器的写保护<br>解锁时只需连续顺序写入以下密钥：<br>密钥1：0xCA<br>密钥2：0x53<br>RTC寄存器写保护相关内容详见：*RTC寄存器写保护*。 |

### 26.6.9 RTC 时钟数字平滑校准寄存器（RTC_CALR）

偏移地址：0x28

复位值：0x0000 0000

*注意：* *该寄存器仅可被 AWON 域复位，并默认处于写保护状态，相关内容详见：RTC 寄存器写保护。*

*仅在 RTC_CSR 寄存器中的 CALSTA 位为 0 时，对该寄存器的写操作有效。*

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15 | CALP | RTC时钟脉冲插入控制<br>0：RTC_KCLK不插入有效脉冲。<br>1：每$2^{11}$个RTC_KCLK时钟周期，等效插入一个有效脉冲<br>详细内容可参见：*RTC时钟数字平滑校准*。 |
| 14:9 | 保留 | 写入无效 |
| 8:0 | CALM[8:0] | RTC时钟脉冲屏蔽控制<br>用以配置在每个校准周期内，等效屏蔽的RTC时钟脉冲个数。<br>详细内容可参见：*RTC时钟数字平滑校准*。 |

### 26.6.10 RTC 闹钟寄存器（RTC_ALARMR）

偏移地址：0x40

复位值：0x0000 0000

*注意：* *该寄存器仅可被 AWON 域复位，并默认处于写保护状态，相关内容详见：RTC 寄存器写保护。*

*仅在 RTC_CR 寄存器中的 ALARME 位为 0 时，对该寄存器的写操作有效。*

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | MSK3 | Res. | HT[1:0] | | HU[3:0] | | | |
| | | | | | | | | rw | | rw | rw | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MSK2 | MNT[2:0] | | | MNU[3:0] | | | | MSK1 | ST[2:0] | | | SU[3:0] | | | |
| rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:24 | 保留 | 写入无效 |
| 23 | MSK3 | 小时屏蔽位<br>0：闹钟的小时有效，参与日历比较<br>1：屏蔽闹钟的小时，不参与日历比较 |
| 22 | 保留 | 写入无效 |
| 21:20 | HT[1:0] | 闹钟小时的十位（BCD格式） |
| 19:16 | HU[3:0] | 闹钟小时的个位（BCD格式） |
| 15 | MSK2 | 分钟屏蔽位<br>0：闹钟的分钟有效，参与日历比较<br>1：屏蔽闹钟的分钟，不参与日历比较 |
| 14:12 | MNT[2:0] | 闹钟分钟的十位（BCD格式） |
| 11:8 | MNU[3:0] | 闹钟分钟的个位（BCD格式） |
| 7 | MSK1 | 秒钟屏蔽位<br>0：闹钟的秒钟有效，参与日历比较<br>1：屏蔽闹钟的秒钟，不参与日历比较 |
| 6:4 | ST[2:0] | 闹钟秒钟的十位（BCD格式） |
| 3:0 | SU[3:0] | 闹钟秒钟的个位（BCD格式） |

### 26.6.11 RTC 状态寄存器（RTC_SR）

偏移地址：0x50

复位值：0x0000 0000

*注意：　该寄存器仅可被 AWON 域复位。*

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | WUTIM2F | WUTIM1F | Res. | | | | CALF | ALARMF |
| | | | | | | | | r | r | | | | | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 写入无效 |
| 7 | WUTIM2F | 周期定时2标志<br>周期定时2使能后，每当到达RTC_WUTIM2寄存器中的定时周期，该标志将自动置1。 |
| 6 | WUTIM1F | 周期定时1标志<br>周期定时1使能后，每当到达RTC_WUTIM1寄存器中的定时周期，该标志将自动置1。 |
| 5:2 | 保留 | 写入无效 |
| 1 | CALF | 校准周期标志<br>当每个校准周期结束时，该标志将自动置1。 |
| 0 | ALARMF | 闹钟匹配标志<br>当RTC日历寄存器（RTC_TR）与闹钟寄存器（RTC_ALARMR）中对应位域的值相等时，该标志将自动置1。 |

### 26.6.12　RTC 中断状态寄存器（RTC_ISR）

偏移地址：0x54

复位值：0x0000 0000

*注意：　该寄存器仅可被 AWON 域复位。*

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

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | | | | | | | | WUTIM2IF | WUTIM1IF | Res. | | | | CALIF | ALARMIF |
| | | | | | | | | r | r | | | | | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 写入无效 |
| 7 | WUTIM2IF | 周期定时2中断标志<br>RTC_CR寄存器中的WUTIM2IE位置1后，RTC_SR寄存器中的WUTIM2F标志置1时，该标志将自动置1，并触发中断。 |
| 6 | WUTIM1IF | 周期定时1中断标志<br>RTC_CR寄存器中的WUTIM1IE位置1后，RTC_SR寄存器中的WUTIM1F标志置1时，该标志将自动置1，并触发中断。 |
| 5:2 | 保留 | 写入无效 |
| 1 | CALIF | 校准周期中断标志<br>在RTC_CR寄存器中的CALIE位置1后，当每个校准周期结束时，该标志将自动置1，并触发校准周期中断。 |
| 0 | ALARMIF | 闹钟匹配中断标志<br>在RTC_CR寄存器中的ALARMIE置1后，当RTC日历寄存器（RTC_TR）与闹钟寄存器（RTC_ALARMR）中对应位域的值相等时，该标志将自动置1，并触发闹钟匹配中断。 |

### 26.6.13 RTC 状态清除寄存器（RTC_CLR）

偏移地址：0x5C

复位值：0x0000 0000

*注意：　该寄存器仅可被 AWON 域复位。*

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | | | | | | | | CWUTIM2F | CWUTIM1F | Res. | | | | CCALF | CALARMF |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | w | w | | | | | w | w |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 写入无效 |
| 7 | CWUTIM2F | 清除周期定时2标志<br>对该位写1，将同时清除WUTIM2F标志和WUTIM2IF标志。 |
| 6 | CWUTIM1F | 清除周期定时1标志<br>对该位写1，将同时清除WUTIM1F标志和WUTIM1IF标志。 |
| 5:2 | 保留 | 写入无效 |
| 1 | CCALF | 清除校准周期标志<br>对该位写1，将同时清除CALF标志和CALIF标志。 |
| 0 | CALARMF | 清除闹钟匹配标志<br>对该位写1，将同时清除ALARMF标志和ALARMIF标志。 |

# 27 I2C 接口（I2C）

## 27.1 简介

I2C 接口实现芯片和外部 I2C 设备的同步通信，支持标准模式（Standard-mode）、快速模式（Fast-mode）和超快速模式（Fast-mode Plus）。

## 27.2 I2C 主要特性

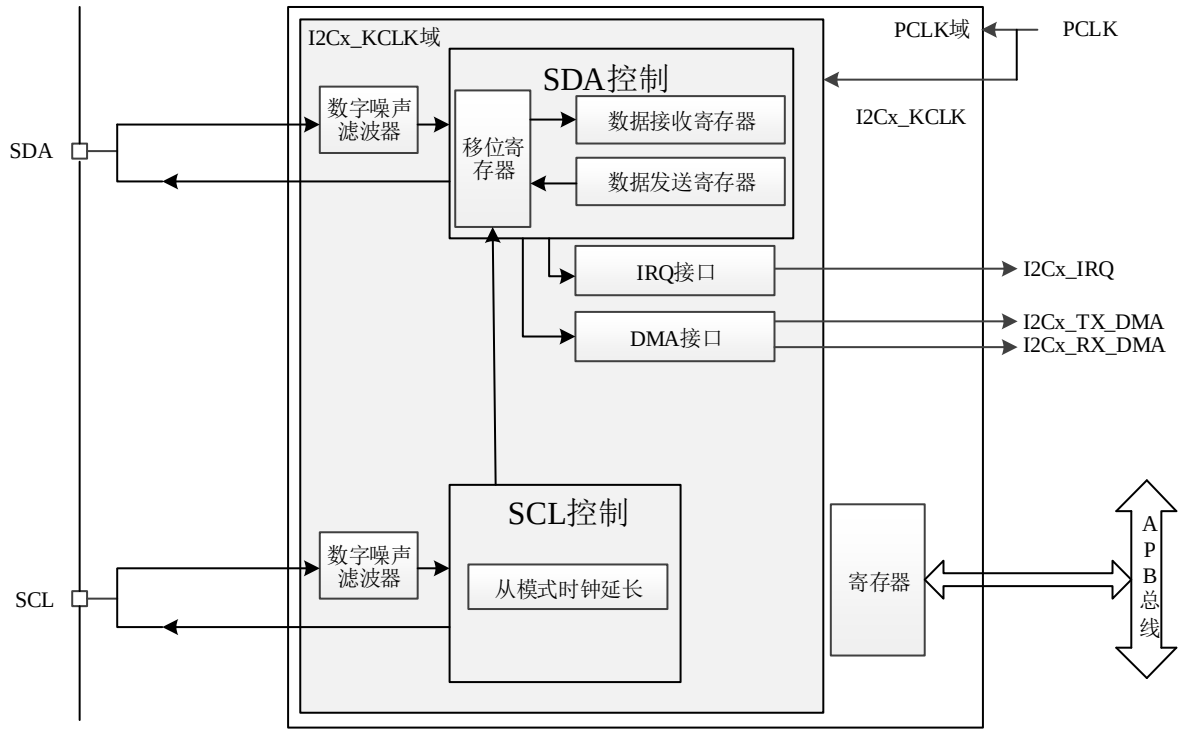
- 支持从模式
- 支持标准模式（最高 100KHz）、快速模式（最高 400KHz）、超快速模式（最高 1MHz）
- 2 个从地址
- 广播地址
- 从模式可配置时钟延长
- 支持单字节控制
- 可配置的数字噪声滤波器
- 支持 DMA 传输

## 27.3 I2C 功能描述

该接口通过数据引脚（SDA）和时钟引脚（SCL）连接到 I2C 总线。

### 27.3.1 I2C 框图

图 27-1 I2Cx 框图（x = 1）




### 27.3.2 I2C 引脚

表 27-1 I2Cx 引脚

| 引脚名称 | 引脚类型 | 说明 |
|---|---|---|
| SCL | 输入/输出 | I2Cx 同步时钟引脚 |
| SDA | 输入/输出 | I2Cx 数据引脚 |

### 27.3.3 I2C 时钟要求

I2C 内核的时钟由 PCLK 提供，I2Cx_KCLK 周期 $t_{I2Cx_KCLK}$ 必须遵循以下条件：

$t_{I2Cx_KCLK} < (t_{LOW} - t_{FILTERS})/6$ 且 $t_{I2Cx_KCLK} < t_{HIGH}$

其中 $t_{LOW}$：SCL 低电平时间；

$t_{HIGH}$：SCL 高电平时间；

$t_{FILTERS}$：噪声滤波器使能时，该值为数字噪声滤波器引入的延时总和（数字滤波器延时为 DNF × $t_{I2Cx_KCLK}$）。

### 27.3.4 I2C 初始化

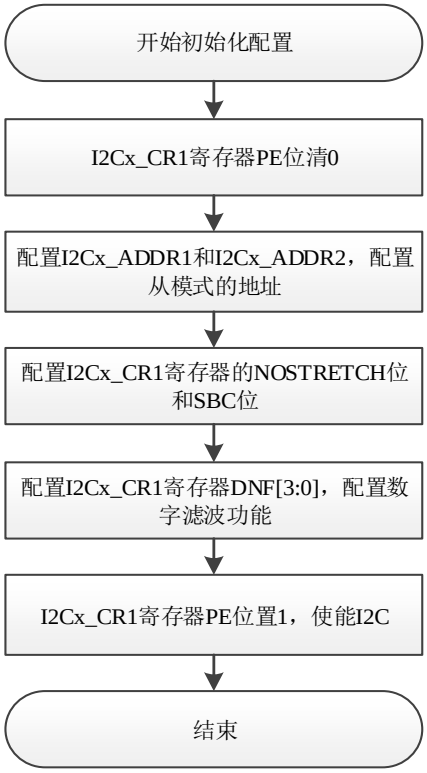
**使能和关闭外设**

在 RCC 中配置使能 I2C 时钟，然后可通过将 I2Cx_CR1 寄存器中的 PE 位置 1 以使能 I2C。

当禁止 I2C（PE 清 0）时，I2C 将执行软件复位。详细信息，请参见*软件复位*。

I2C 初始化流程如下：

图 27-2 I2C 初始化流程




### 噪声滤波器

提供可配置的数字噪声滤波器，用于实现输入信号滤波。默认情况下，SDA 和 SCL 数字噪声滤波器是关闭的，通过配置 I2Cx_CR1 寄存器中的 DNF[3:0]位来使能数字滤波器。使能数字滤波器后，SCL 或 SDA 的电平只有在电平稳定时间超过 DNF 个 I2Cx_KCLK 时钟周期后才会认为电平有效，从而抑制在 1~15 个 I2Cx_KCLK 时钟周期的噪声。

### 地址模式

I2C 模块通过地址寄存器 I2Cx_ADDR1 和地址寄存器 I2Cx_ADDR2 支持 7 位地址模式。

如果使能多个地址，从模式地址匹配后可以读取 I2Cx_ISR 寄存器中的 DUADDR 位，以确定是哪个地址匹配，同时检查 DIR 标志获取传输方向。

通过将 I2Cx_CR1 寄存器中的 GCEN 位置 1 来使能广播呼叫地址，从模式地址

匹配后可以读取 I2Cx_ISR 寄存器中的 GCALLADDR 位确认广播地址匹配。

**时钟低电平延长**

- 使能时钟延长的从模式（NOSTRETCH 为 0）
  - 单字节控制模式（SBC 为 1），I2C 从设备会在以下情况下延长 SCL 时钟线：

    当接收到的地址和从地址匹配时，ADDR 标志置 1，在发送完 ACK 后，SCL 时钟线被延长。向 I2Cx_ICR 寄存器的 ADDRCF 和 SICF 位写 1 清除 ADDR 和 SI 标志后，将释放 SCL 时钟线。

    在发送过程中，当发送数据寄存器 I2Cx_TDR 和移位寄存器中均无有效数据时，接收完 ACK 后，SCL 时钟线被延长。向 I2Cx_TDR 寄存器中写入数据，且向 I2Cx_ICR 寄存器的 SICF 位写 1 清除 SI 标志后，将释放 SCL 时钟线。

    在接收过程中，当接收数据寄存器 I2Cx_RDR 中还有未被读取的数据，发送完 ACK/NACK 后，SCL 时钟线被延长。读取 I2Cx_RDR 寄存器，且向 I2Cx_ICR 寄存器的 SICF 位写 1 清除 SI 标志后，将释放 SCL 时钟线。
  - 多字节控制模式（SBC 为 0），I2C 从设备会在以下情况下延长 SCL 时钟线：

    当接收到的地址和从地址匹配时，ADDR 标志置 1，在发送完 ACK 后，SCL 时钟线被延长。向 I2Cx_ICR 寄存器的 ADDRCF 位写 1 清除 ADDR 标志，将释放 SCL 时钟线。

    在发送过程中，当发送数据寄存器 I2Cx_TDR 和移位寄存器中均无有效数据时，接收完 ACK 后，SCL 时钟线被延长。向 I2Cx_TDR 寄存器中写入数据后，将释放 SCL 时钟线。

    在接收过程中，当接收数据寄存器 I2Cx_RDR 中还有未被读取的数据，且移位寄存器又接收完成新数据时，发送完 ACK/NACK 后，SCL 时钟线被延长。读取 I2Cx_RDR 寄存器后，将释放 SCL 时钟线。
- 禁止时钟延长的从模式（NOSTRETCH 为 1）

  当 I2Cx_CR1 寄存器中的 NOSTRETCH 为 1 时，I2C 从设备不会延长 SCL 时钟线。

ADDR 标志置 1 时，不会延长 SCL 时钟线。

### 27.3.5 软件复位

可通过将 I2Cx_CR1 寄存器中的 PE 位清 0 来执行软件复位。在这种情况下，SCL 和 SDA 被释放为高电平，内部状态机复位，通信控制位和状态位恢复为复位值。下面列出了受影响的寄存器位：

- I2Cx_CR2 寄存器：NACK。
- I2Cx_ISR 寄存器：SI、DUADDR、GCALLADDR、DIR、BUSY、TXE、TXIS、RXNE、ADDR、NACKF、STOPF、BERR 和 OVR。

### 27.3.6 数据传输

I2C 支持单字节控制（SBC=1）和多字节控制（SBC=0）两种数据传输，以方便管理字节传输和结束通信。

**接收**

接收数据时，SDA 接收的数据输入到移位寄存器，接收到完整的数据字节在第 8 个 SCL 脉冲后，如果 I2Cx_RDR 寄存器为空（RXNE 为 0），则移位寄存器的内容会复制到其中。针对两种数据传输方式：

单字节控制数据传输时，在第 9 个和第 1 个 SCL 脉冲之间（应答脉冲之后），SI 标志置 1，延长 SCL 的低电平时间，直到写 SICF 清除 SI 标志为止。注意在清除 SI 标志前，必须读走 I2Cx_RDR 中的数据。

多字节控制数据传输时，移位寄存器中的数据往 I2Cx_RDR 放置时，如果尚未读 I2Cx_RDR 中接收的数据字节（RXNE 为 1），则将在第 9 个和第 1 个 SCL 脉冲之间（应答脉冲之后）延长 SCL 的低电平时间，直到读取了 I2Cx_RDR 的数据为止。

图 27-3 单字节控制数据接收

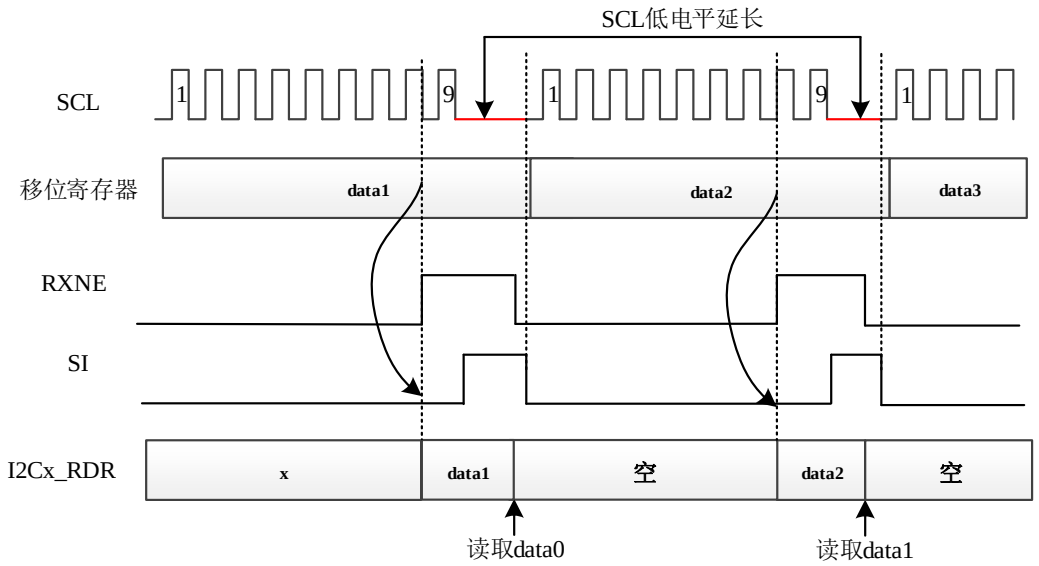
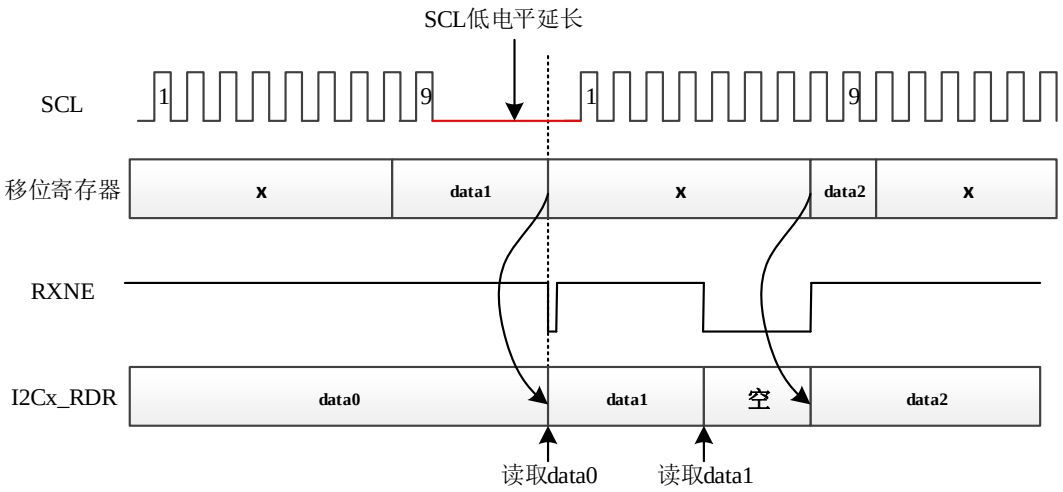



图 27-4 多字节控制数据接收




## 发送

发送数据时，如果 I2Cx_TDR 寄存器非空（TXE 为 0），则其内容会在第 9 个 SCL 脉冲（应答脉冲）后复制到移位寄存器中。然后移位寄存器的内容会依次发送到 SDA 线上。针对两种数据传输方式：

单字节控制数据传输时，在第 9 个和第 1 个 SCL 脉冲之间（应答脉冲之后），SI 标志置 1，延长 SCL 的低电平时间，直到写 SICF 清除 SI 标志为止。注意在清除 SI 标志前，必须往 I2Cx_TDR 内写入要发送的数据。

多字节控制数据传输时，如果 I2Cx_TDR 内尚未写入任何数据（TXE 为 1），则将在第 9 个 SCL 脉冲后延长 SCL 线的低电平时间，直到待发送数据写入了 I2Cx_TDR 为止。

图 27-5 单字节控制数据发送

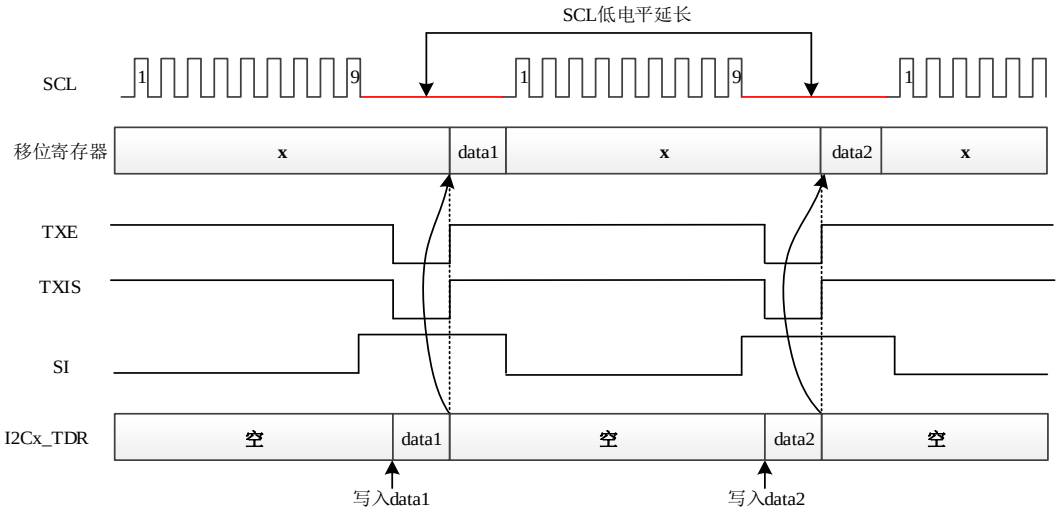
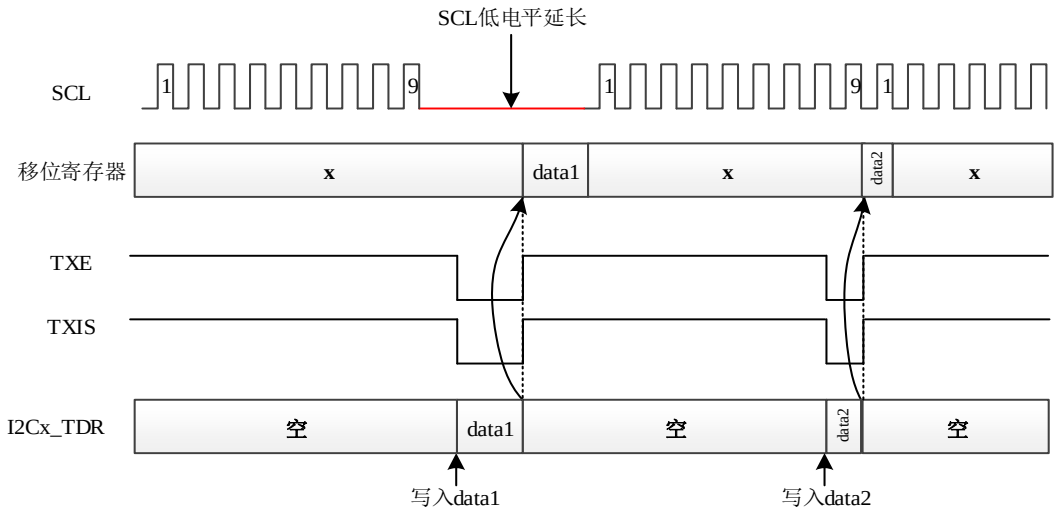



图 27-6 多字节控制数据发送




### 27.3.7 通信流程

I2C 从模式支持 7 位地址，用户通过地址寄存器 I2Cx_ADDR1 和 I2Cx_ADDR2 配置自身从地址。I2C 从设备接收到的地址和自身地址一致时，硬件发送应答位 ACK，ADDR 标志置 1，DIR 传输方向更新，如果使能多个地址，从模式地址匹配后可以读取 I2Cx_ISR 寄存器中的 DUADDR 位，以确定是哪个地址匹配，如果 I2Cx_CR1 寄存器中的 EVTIE 位置 1，同时会生成中断。

从地址匹配后，接收到停止位，STOPF 标志置 1，如果 I2Cx_CR1 寄存器中的 EVTIE 位置 1，同时会生成中断。

**发送数据时序**

当 I2Cx_TDR 寄存器为空时，TXE 置 1。在发送模式下，当从地址匹配且 I2Cx_TDR 寄存器为空时，TXIS 将置 1，如果 I2Cx_CR1 寄存器中的 BUFIE 位

置 1，同时会生成中断。I2Cx_TDR 寄存器中写入待发送的数据时，TXIS 和 TXE 位将被清 0。

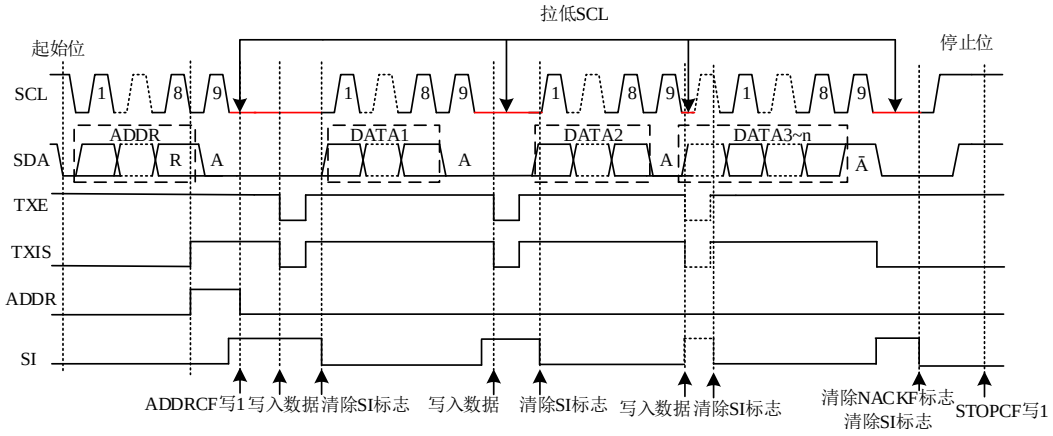
在数据传输前，如果 I2Cx_TDR 寄存器已有数据存在，可以通过将 I2Cx_ISR 寄存器中的 TXE 置 1，使原数据失效重新写入新的数据。

当 NOSTRETCH 置 1 时，ADDR 标志置 1 时不会延长 SCL，为避免无法及时将待发送数据写入 I2Cx_TDR 寄存器，产生下溢错误，必须提前在 I2Cx_TDR 寄存器中写入待发送的第一个数据。

数据传输过程中，接收到 NACK 时，I2Cx_ISR 寄存器中的 NACKF 位将置 1，如果 I2Cx_CR1 寄存器中的 EVTIE 位置 1，同时会生成中断，从设备自动释放 SCL 和 SDA。收到 NACK 时，TXIS 位不会置 1。

使能低电平延长，单字节控制（NOSTRETCH 清 0，SBC 置 1）发送数据时序图：

图 27-7　使能时钟低电平延长单字节控制发送数据



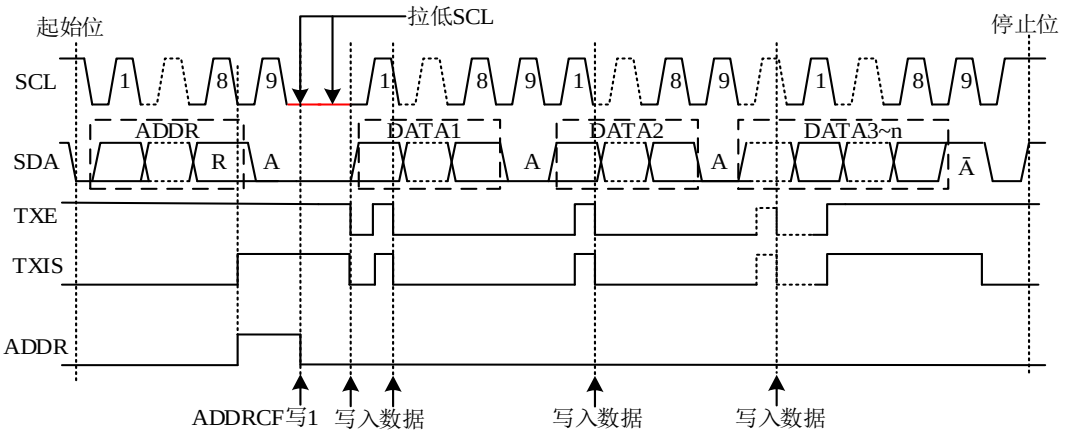

操作流程如下：

1) 等待 ADDR 置 1，查询 DUADDR、GCALLADDR 和 DIR 值，确认匹配的地址和数据传输方向，ADDRCF 置 1 清除 ADDR 标志；
2) 等待 SI 置 1，查询 TXIS 置 1，DATA1 写入 I2Cx_TDR 寄存器，SICF 置 1 清除 SI 标志；
3) 重复步骤 2，依次将 DATA2~DATAn 写入 I2Cx_TDR 寄存器；
4) 等待 SI 置 1，查询 NACKF 置 1，NACKCF 置 1 清除 NACKF 标志，SICF 置 1 清除 SI 标志；
5) 等待 STOPF 置 1，清除 STOPF 标志，结束本次数据传输。

使能低电平延长，多字节控制（NOSTRETCH 清 0，SBC 清 0）发送数据时序

图：

图 27-8　使能时钟低电平延长多字节控制发送数据

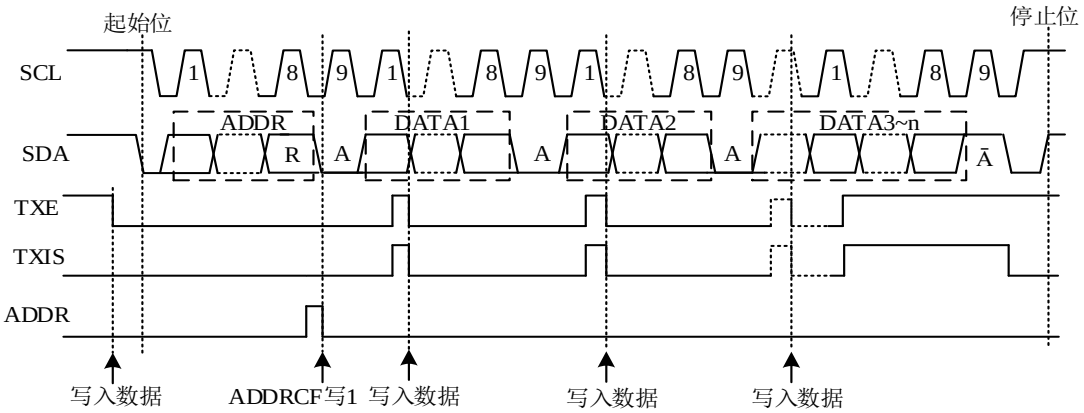



操作流程如下：

1) 等待 ADDR 置 1，查询 DUADDR、GCALLADDR 和 DIR 值，确认匹配的地址和数据传输方向，ADDRCF 置 1 清除 ADDR 标志；
2) 等待 TXIS 置 1，DATA1 写入 I2Cx_TDR 寄存器；
3) 重复步骤 2，依次将 DATA2~DATAn 写入 I2Cx_TDR 寄存器；
4) 等待 NACKF 置 1，NACKCF 置 1 清除 NACKF 标志；
5) 等待 STOPF 置 1，清除 STOPF 标志，结束本次数据传输。

禁止低电平延长功能（NOSTRETCH 置 1）发送数据时需要在地址匹配事件前将待发送的第一个字节数据写入 I2Cx_TDR 寄存器，如下图：

图 27-9　禁止时钟低电平延长发送数据




操作流程如下：

1) 等待 TXE 置 1，DATA1 写入 I2Cx_TDR 寄存器；

2) 等待 ADDR 置 1，查询 DUADDR、GCALLADDR 和 DIR 值，确认匹配的地址和数据传输方向，ADDRCF 置 1 清除 ADDR 标志；
3) 等待 TXIS 置 1，DATA2 写入 I2Cx_TDR 寄存器；
4) 重复步骤 3，依次将 DATA3~DATAn 写入 I2Cx_TDR 寄存器；
5) 等待 NACKF 置 1，NACKCF 置 1 清除 NACKF 标志；
6) 等待 STOPF 置 1，清除 STOPF 标志，结束本次数据传输。

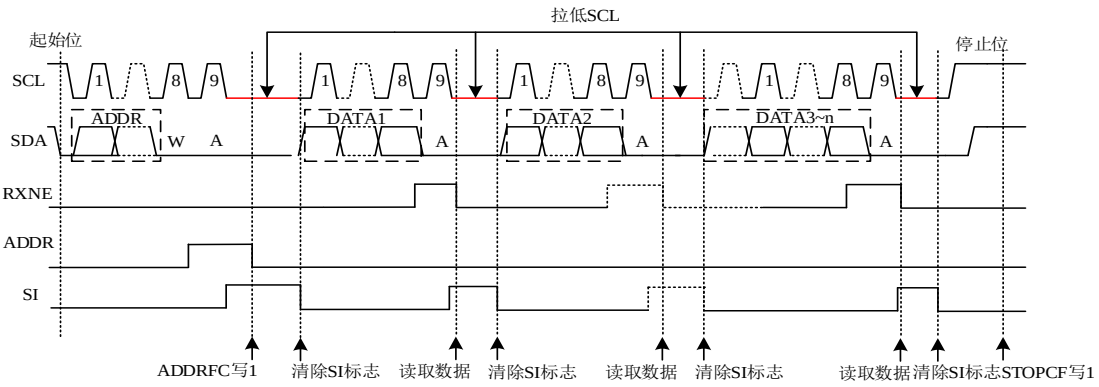
**接收数据时序**

I2Cx_RDR 寄存器接收到数据，I2Cx_ISR 中的 RXNE 将置 1，如果 I2Cx_CR1 中的 BUFIE 置 1，同时会生成中断。读取 I2Cx_RDR 时，RXNE 标志将被清 0。

通常情况下应由主设备控制数据的传输，如果从设备想主动结束数据的接收，可以将 I2Cx_CR2 控制寄存器中的 NACK 位置 1，从设备在完成当前接收的字节后发送 NACK 响应位，自动释放 SCL 和 SDA，主设备在接收到 NACK 响应位后，将自动发送停止位结束本次数据传输。

使能低电平延长功能，单字节控制（NOSTRETCH 为 0，SBC 置 1）接收数据时序图：

图 27-10 使能时钟低电平延长单字节控制接收数据

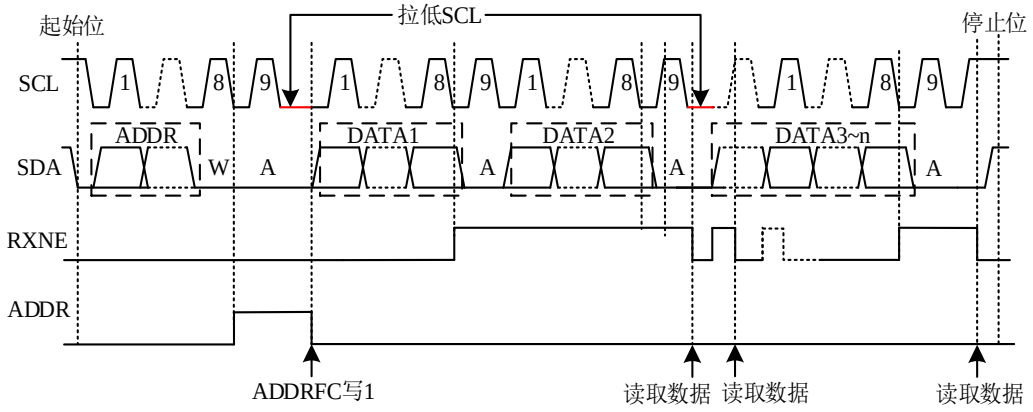



操作流程如下：

1) 等待 SI 置 1，查询 ADDR、DUADDR、GCALLADDR 和 DIR 值，确认匹配的地址和数据传输方向，ADDRCF 置 1 清除 ADDR 标志，SICF 置 1 清除 SI 标志；
2) 等待 SI 置 1，查询 RXNE 置 1，从 I2Cx_RDR 寄存器读取 DATA1，SICF 置 1 清除 SI 标志；
3) 重复步骤 2，依次从 I2Cx_RDR 寄存器读取 DATA2~DATAn；
4) 等待 STOPF 置 1，清除 STOPF 标志，结束本次数据传输。

使能低电平延长功能，多字节控制（NOSTRETCH 为 0，SBC 清 0）接收数据时序图：

图 27-11 使能时钟低电平延长多字节控制接收数据

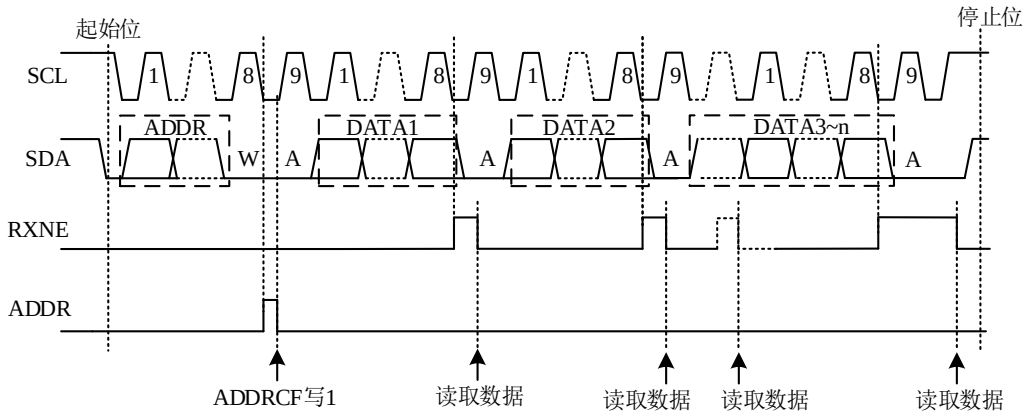



操作流程如下：

1) 等待 ADDR 置 1，查询 DUADDR、GCALLADDR 和 DIR 值，确认匹配的地址和数据传输方向，ADDRCF 置 1 清除 ADDR 标志；
2) 等待 RXNE 置 1，从 I2Cx_RDR 寄存器读取 DATA1；
3) 重复步骤 2，依次从 I2Cx_RDR 寄存器读取 DATA2~DATAn；
4) 等待 STOPF 置 1，清除 STOPF 标志，结束本次数据传输。

禁止低电平延长功能（NOSTRETCH 置 1）接收数据时序图：

图 27-12 禁止时钟低电平延长接收数据




操作流程如下：

1) 等待 ADDR 置 1，查询 DUADDR、GCALLADDR 和 DIR 值，确认匹配的地址和数据传输方向，ADDRCF 置 1 清除 ADDR 标志；

2) 等待 RXNE 置 1，从 I2Cx_RDR 寄存器读取 DATA1；
3) 重复步骤 2，依次从 I2Cx_RDR 寄存器读取 DATA2~DATAn；
4) 等待 STOPF 置 1，清除 STOPF 标志，结束本次数据传输。

### 27.3.8 DMA 传输

**DMA 发送**

将 I2Cx_CR1 寄存器中的 TXDMAEN 位置 1 来使能 DMA（请参见：*直接存储器访问控制器（DMA）*）发送。当 TXIS 位置 1 时，DMA 将源地址的数据写入 I2Cx_TDR 寄存器。

DMA 只能传输数据，不能传输应答位、起始位和停止位：

**DMA 接收**

将 I2Cx_CR1 寄存器中的 RXDMAEN 位置 1 来使能 DMA（请参见：*直接存储器访问控制器（DMA）*）接收。当 RXNE 位置 1 时，数据将从 I2Cx_RDR 寄存器写入 DMA 目的地址。

### 27.3.9 错误条件

以下错误条件会导致通信失败。

- 总线错误

  地址匹配后的从设备进入数据传输阶段，在每个数据的传输过程检测到起始位或停止位，生成总线错误。I2Cx_ISR 寄存器中的 BERR 标志将置 1。

  在地址接收阶段，检测到起始位或重复起始位时，将重新接收地址，BERR 标志不会置 1，如果接收到的是停止位，则直接结束本次地址接收，BERR 标志也不会置 1。

- 上溢、下溢错误

  仅在 I2Cx_CR1 寄存器 NOSTRETCH 为 1 时会检测到上溢或下溢错误。当检测到溢出错误时，I2Cx_ISR 寄存器中的 OVR 标志将置 1。

  在接收过程中，当接收到一个新的字节，但 I2Cx_RDR 寄存器的值还未被读取时，将产生上溢错误，接收的新字节将被丢弃，自动发送 NACK 来响应丢弃的字节。

  在发送过程中，当响应位的 SCL 下降沿出现时，发送数据寄存器 I2Cx_TDR

和移位寄存器中均无有效数据，将产生下溢错误，并自动发送数据 0xFF。

## 27.4 I2C 中断

I2C 中断请求如下：

表 27-2 中断请求

| 中断事件 | 事件标志 | 中断使能位 | 事件标志/中断清除方法 |
|---|---|---|---|
| 接收数据寄存器非空 | RXNE | BUFIE | 读取 I2Cx_RDR 寄存器 |
| 发送数据寄存器中断状态 | TXIS | | 向 I2Cx_TDR 寄存器写入数据 |
| 地址匹配 | ADDR | EVTIE | 向 ADDRCF 位写 1 |
| 接收到 NACK | NACKF | | 向 NACKCF 位写 1 |
| 接收到停止位 | STOPF | | 向 STOPCF 位写 1 |
| 总线错误 | BERR | ERRIE | 向 BERRCF 位写 1 |
| 上溢、下溢错误 | OVR | | 向 OVRCF 位写 1 |
| I2C 单字节状态 | SI | SIIE | 向 SICF 位写 1 |

## 27.5 I2C 寄存器

I2C 寄存器支持 32 位访问。

表 27-3 I2Cx 基地址

| 外设 | 基地址 |
|---|---|
| I2C1 | 0x4000 5400 |

### 27.5.1 I2C 控制寄存器 1（I2Cx_CR1）

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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | GCEN | Res. | NO STRETCH | SBC |
| | | | | | | | | | | | | rw | | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RXDMA EN | TXDMA EN | SIIE | Res. | DNF[3:0] | | | | ERRIE | Res. | | | EVTIE | Res. | BUFIE | PE |
| rw | rw | rw | | rw | rw | rw | rw | rw | | | | rw | | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:20 | 保留 | 写入无效 |
| 19 | GCEN | 广播地址使能<br>0：禁止<br>1：使能 |
| 18 | 保留 | 写入无效 |
| 17 | NOSTRETCH | 时钟低电平延长控制<br>该位用于在从模式下禁止时钟延长。<br>0：使能<br>1：禁止<br>*注意：该位只能在I2C禁止状态（PE为0）时写入。* |
| 16 | SBC | 单字节控制使能<br>该位用于控制数据传输形式为单字节传输还是多字节传输。<br>0：禁止<br>1：使能 |

*注意：该位仅当NOSTRETCH为0时使用。*

| | | |
|---|---|---|
| 15 | RXDMAEN | DMA接收请求使能<br>0：禁止<br>1：使能 |
| 14 | TXDMAEN | DMA发送请求使能<br>0：禁止<br>1：使能 |
| 13 | SIIE | 单字节状态中断使能<br>0：禁止<br>1：使能 |
| 12 | 保留 | 写入无效 |
| 11:8 | DNF[3:0] | 数字噪声滤波器<br>配置SCL和SDA输入端的数字噪声滤波器功能，可滤除的脉宽为小于DNF[3:0]×I2Cx_KCLK以下的脉冲。<br>0000：禁止数字噪声滤波器<br>0001：开启数字噪声滤波器，滤除脉宽小于1×I2Cx_KCLK的噪声<br>⋮<br>1110：开启数字噪声滤波器，滤除脉宽小于14×I2Cx_KCLK的噪声<br>1111：开启数字噪声滤波器，滤除脉宽小于15×I2Cx_KCLK的噪声<br>*注意：该位域只能在I2C禁止状态（PE为0）时写入。* |
| 7 | ERRIE | 错误中断检测使能<br>0：禁止<br>1：使能 |
| 6:4 | 保留 | 写入无效 |
| 3 | EVTIE | 事件中断使能<br>0：禁止<br>1：使能 |

| | | |
|---|---|---|
| 2 | 保留 | 写入无效 |
| 1 | BUFIE | 缓冲器中断使能<br>0：禁止<br>1：使能 |
| 0 | PE | I2C使能<br>0：禁止<br>1：使能<br>*注意：将PE清0，会执行软件复位：将释放SCL和SDA，内部状态机和状态位均恢复为复位值。* |

### 27.5.2 I2C 控制寄存器 2（I2Cx_CR2）

偏移地址：0x04

复位值：0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15 | NACK | NACK发送控制<br>该位由软件置1，并可在发送NACK、接收到停止位、地址匹配、或当PE为0时，该位自动清0。<br>0：在当前接收的字节后发送ACK<br>1：在当前接收的字节后发送NACK<br>*注意：当NOSTRETCH为1，从模式接收数据发生上溢错误时，将自动生成NACK，不受NACK位的控制。* |
| 14:0 | 保留 | 写入无效 |

### 27.5.3 I2C 地址寄存器 1（I2Cx_ADDR1）

偏移地址：0x08

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDR1EN | Res. | | | | | | | ADDR1[7:1] | | | | | | | Res. |
| rw | | | | | | | | rw | rw | rw | rw | rw | rw | rw | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15 | ADDR1EN | 从模式地址1使能<br>0：禁止：接收到地址1不应答ACK<br>1：使能：接收到地址1应答ACK |
| 14:8 | 保留 | 写入无效 |
| 7:1 | ADDR1[7:1] | 从模式地址1<br>ADDR1[7:1]：7位从模式地址<br>*注意：在ADDR1EN为0时才可以写入该位域。* |
| 0 | 保留 | 写入无效 |

### 27.5.4 I2C 地址寄存器 2（I2Cx_ADDR2）

偏移地址：0x0C

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDR2EN | Res. | | | | | | | ADDR2[7:1] | | | | | | | Res. |
| rw | | | | | | | | rw | rw | rw | rw | rw | rw | rw | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15 | ADDR2EN | 从模式地址2使能<br>0：禁止：接收到地址2不应答ACK<br>1：使能：接收到地址2应答ACK |
| 14:8 | 保留 | 写入无效 |
| 7:1 | ADDR2[7:1] | 从模式地址2<br>ADDR2[7:1]：7位从模式地址<br>*注意：在ADDR2EN为0时才可以写入该位域。* |
| 0 | 保留 | 写入无效 |

### 27.5.5 I2C 中断和状态寄存器（I2Cx_ISR）

偏移地址：0x18

复位值：0x0000 0001

*注意：　当 PE 为 0 时，该寄存器保持复位值。*

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SI | Res. | | | | | DU ADDR | GCALL ADDR | Res. | | | | | | | DIR |
| r | | | | | | r | r | | | | | | | | r |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BUSY | Res. | | | | OVR | Res. | BERR | Res. | | STOPF | NACKF | ADDR | RXNE | TXIS | TXE |
| r | | | | | r | | r | | | r | r | r | r | rs | rs |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31 | SI | I2C单字节状态<br>将SICF位置1，该位自动清0。<br>0：未接收或发送ACK/NACK<br>1：接收或发送ACK/NACK<br>*注意：在SI被清0之前，SCL低电平周期延长传输暂停。*<br>*在清除SI前，软件应该准备好合适的寄存器设置，参见数据传输章节。* |
| 30:26 | 保留 | 写入无效 |

| | | |
|---|---|---|
| 25 | DUADDR | 指示哪个地址匹配<br>在发生地址匹配事件ADDR置1时更新。<br>当检测到起始位或停止位时，该位自动清0。<br>0：接收到的地址与I2Cx_ADDR1的地址匹配<br>1：接收到的地址与I2Cx_ADDR2的地址匹配 |
| 24 | GCALLADDR | 广播呼叫地址匹配（从模式）<br>在发生地址匹配事件ADDR置1时更新。<br>当检测到起始位或停止位时，该位自动清0。<br>0：未收到广播呼叫地址<br>1：当GCEN=1时，收到广播呼叫地址 |
| 23:17 | 保留 | 写入无效 |
| 16 | DIR | 数据传输方向<br>在发生地址匹配事件ADDR置1时更新。<br>当检测到起始位或停止位时，该位自动清0。<br>0：接收数据<br>1：发送数据 |
| 15 | BUSY | 总线忙<br>该标志用于指示总线上正在进行通信。当检测到起始位时，该位由硬件置1。当检测到停止位时，该位自动清0。 |
| 14:11 | 保留 | 写入无效 |
| 10 | OVR | 从模式溢出错误标志<br>在从模式下且NOSTRETCH为1时，如果发生上溢/下溢错误，该标志由硬件置1，将OVRCF位置1，该位自动清0。 |
| 9 | 保留 | 写入无效 |
| 8 | BERR | 总线错误标志<br>数据传输中检测到总线错误时，该标志置1，将BERRCF位置1，该位自动清0。在从模式地址接收阶段，该标志不会置1。 |
| 7:6 | 保留 | 写入无效 |

| | | |
|---|---|---|
| 5 | STOPF | 停止位标志<br>从模式地址匹配后，在总线上检测到停止位，该标志由硬件置1。将STOPCF位置1，该位自动清0。 |
| 4 | NACKF | 接收到NACK标志<br>当前字节发送完后接收到NACK，该标志由硬件置1，将NACKCF位置1，该位自动清0。 |
| 3 | ADDR | 地址匹配<br>从模式接收的地址与使能的从设备地址寄存器匹配时，该标志由硬件置1，将ADDRCF位置1，该位自动清0。 |
| 2 | RXNE | 接收数据寄存器非空<br>I2Cx_RDR寄存器非空，该标志由硬件置1；读取I2Cx_RDR数据后，该位自动清0。 |
| 1 | TXIS | 发送中断状态<br>当处于发送模式时，当I2Cx_TDR为空，该标志置1；将数据写入I2Cx_TDR后，该位自动清0。 |
| 0 | TXE | 发送数据寄存器为空<br>当I2Cx_TDR寄存器为空时，该标志由硬件置1，将数据写入I2Cx_TDR后，该位自动清0。<br>该位可软件写1，使I2Cx_TDR寄存器中的数据失效。 |

### 27.5.6 I2C 中断清除寄存器（I2Cx_ICR）

偏移地址：0x1C

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | OVRCF | Res. | BERRCF | Res. | | STOPCF | NACKCF | ADDRCF | Res. | | |
| | | | | | w | | w | | | w | w | w | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31 | SICF | 状态改变标志清除<br>该位写入1，I2Cx_ISR寄存器中的SI标志清0。 |
| 30:11 | 保留 | 写入无效 |
| 10 | OVRCF | 溢出标志清除<br>该位写入1，I2Cx_ISR寄存器中的OVR标志清0。 |
| 9 | 保留 | 写入无效 |
| 8 | BERRCF | 总线错误标志清除<br>该位写入1，I2Cx_ISR寄存器中的BERR标志清0。 |
| 7:6 | 保留 | 写入无效 |
| 5 | STOPCF | 停止位标志清除<br>该位写入1，I2Cx_ISR寄存器中的STOPF标志清0。 |
| 4 | NACKCF | NACK位标志清除<br>该位写入1，I2Cx_ISR寄存器中的NACKF标志清0。 |
| 3 | ADDRCF | 从模式地址匹配标志清除<br>该位写入1，I2Cx_ISR寄存器中的ADDR标志清0。 |
| 2:0 | 保留 | 写入无效 |

### 27.5.7 I2C 接收数据寄存器（I2Cx_RDR）

偏移地址：0x24

复位值：0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 写入无效 |
| 7:0 | RXDATA[7:0] | 接收数据寄存器 |

### 27.5.8 I2C 发送数据寄存器（I2Cx_TDR）

偏移地址：0x28

复位值：0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 写入无效 |
| 7:0 | TXDATA[7:0] | 发送数据寄存器<br>*注意：仅可在I2Cx_ISR寄存器的TXE为1时，该位域才可以写入数据。* |

# 28 通用异步收发器（UART）

## 28.1 简介

异步收发器 UART 支持全双工或半双工，异步的通信方式，符合通用工业标准 NRZ 异步串行编码格式，内部集成小数级波特率发生器，波特率范围可配置。

UART 支持单线半双工通信，支持 DMA 通信。

提供了 2 个异步收发器 UART1/2。

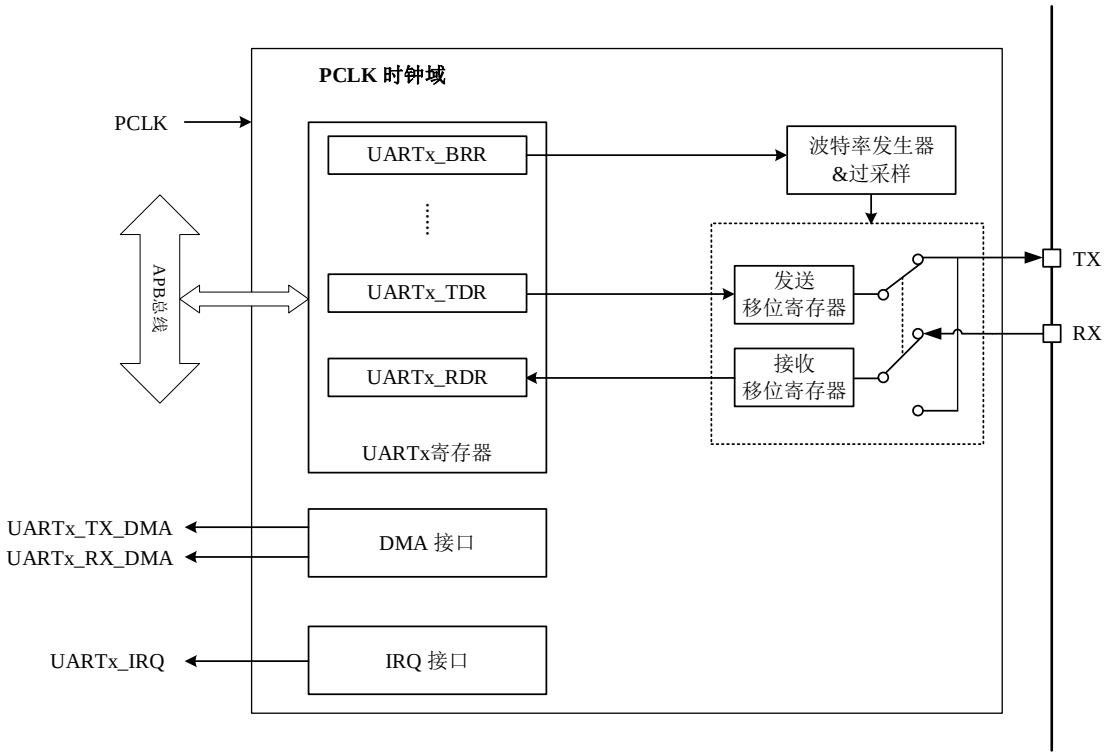
## 28.2 UART 主要特性

- 可编程波特率发生器
  - 系统时钟 72MHz，16 倍过采样，最高速度可达 4.5Mbps
  - 小数级波特率发生器
- 帧格式
  - 数据长度：8 位、9 位
  - 停止位位数：1 位或 2 位
  - 奇校验、偶校验、无校验
  - MSB 或 LSB 优先
- 单线半双工通信
- TX/RX 引脚可交换
- 发送/接收的信号极性可独立控制
- 支持 DMA 通信

## 28.3 UART 功能描述

### 28.3.1 UART 框图

图 28-1 UARTx 框图（x = 1，2）




### 28.3.2 UART 引脚

UART 使用 2 个引脚与外部器件通信：数据接收（RX）和数据发送（TX）。

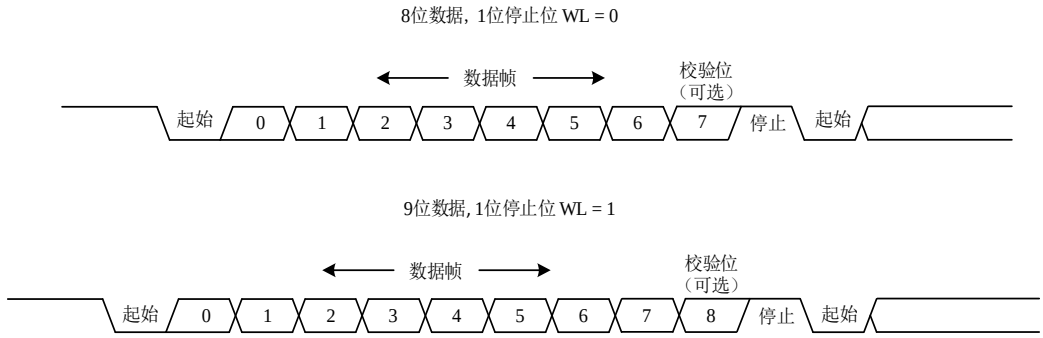
支持 TX 引脚 RX 引脚互换功能，在 UARTx_CR2 的 SWAP 位配置。

### 28.3.3 帧格式

可通过 UARTx_CR1 寄存器中的 WL 位配置字符长度，参见帧格式示意图。

- 8 位字符长度：WL = 0
- 9 位字符长度：WL = 1

图 28-2 帧格式示意图




### 28.3.4 校验控制

表 28-1 UART 校验控制

| 字符长度 | | 校验控制 PEN | UART 帧格式 |
|---|---|---|---|
| 8 位 | WL=0 | 0 | \| 起始位 \| 8 位数据 \| 停止位 \| |
| | | 1 | \| 起始位 \| 7 位数据 \| 校验位 \| 停止位 \| |
| 9 位 | WL=1 | 0 | \| 起始位 \| 9 位数据 \| 停止位 \| |
| | | 1 | \| 起始位 \| 8 位数据 \| 校验位 \| 停止位 \| |

**偶校验**

使能偶校验时，一个字符中（含校验位）“1”的总数为偶数。（UARTx_CR1 寄存器中 PTS 位=0）。

例如，数据“0101 1010”中已包含偶数个“1”，则使能偶校验时，校验位为“0”。

**奇校验**

使能奇校验时，一个字符中（含校验位）“1”的总数为奇数。（UARTx_CR1 寄存器中 PTS 位=1）。

例如，数据“0101 1010”中包含偶数个“1”，则使能奇校验时，校验位为“1”。

**接收检测奇偶校验**

UARTx_CR1 寄存器中的 PEN 位置位使能奇偶校验，使能后，如果奇偶校验失败，则 UARTx_ISR 寄存器中的奇偶校验错误标志 PE 位置位，如果 UARTx_CR1 寄存器中 PEIE 位置位，则触发中断。向 UARTx_ICR 寄存器中的 PECF 位写 1，可清零 PE 标志。

**发送生成奇偶校验**

UARTx_CR1 寄存器中的 PEN 位置位使能奇偶校验，使能后，数据寄存器中数据的 MSB 会替换为奇偶校验位。如果选择偶校验（PTS=0），则一个字符中“1”的总数为偶数；如果选择奇校验（PTS=1），则一个字符中“1”的总数为奇数。

### 28.3.5 发送器

根据 WL 的配置，发送的字符长度可以是 8 位或 9 位。发送使能位（TE）置位即可使能发送器。发送移位寄存器的数据由 TX 引脚输出。

#### 字符发送

通过 UARTx_CR1 寄存器的 TE 位置位使能发送器。

UART 发送期间，TX 引脚首先移出数据的最低有效位（默认 LSB 优先）。发送数据寄存器 UARTx_TDR 为内部总线和发送移位寄存器之间提供缓冲区。

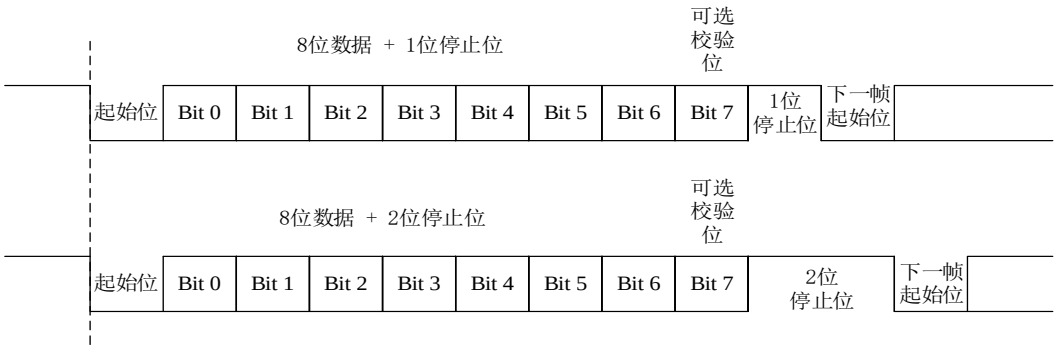
每个字符的发送都以起始位开始，以停止位结束，停止位的位数在 UARTx_CR2 寄存器的 STOPBIT 位域可配置为 1 或 2 位。

#### 停止位

可以在 UARTx_CR2 的 STOPBIT 中编程停止位的位数，可配置为 1 位、2 位。

下图为不同停止位配置示例：

图 28-3 可编程停止位示例




#### 字符发送配置步骤

1) 配置 UARTx_BRR 寄存器选择通信波特率；
2) 配置 UARTx_CR1 中的 WL 位定义字符长度；
3) 配置 UARTx_CR2 中的 STOPBIT 设置停止位数；
4) 通过向 UARTx_CR1 寄存器中的 UE 位写入“1”使能 UART；
5) 使用 DMA 通信时，将 UARTx_CR3 中的 DMA 发送使能位 DMAT 写“1”；参见 *UART 使用 DMA 通信*中的说明配置 DMA 寄存器；

6) 将 UARTx_CR1 中的 TE 位置位，使能发送器；

7) 在 UARTx_TDR 寄存器中写入准备发送的数据；向 UARTx_TDR 写入数据会将 TXE 标志清零，表示 UARTx_TDR 发送寄存器中数据已满；

8) 当最后一个待发送的数据写入 UARTx_TDR 寄存器后，需等待发送完成直到 UARTx_ISR 中的发送完成标志位 TC 置位。TC 置位时表示最后一个字符已发送完成。

**字符发送过程**

UARTx_TDR 寄存器写入 1 个字符后，发送数据空标志 TXE 位清零。

发送数据寄存器 UARTx_TDR 为空时，TXE 标志位自动置位，此时表示：

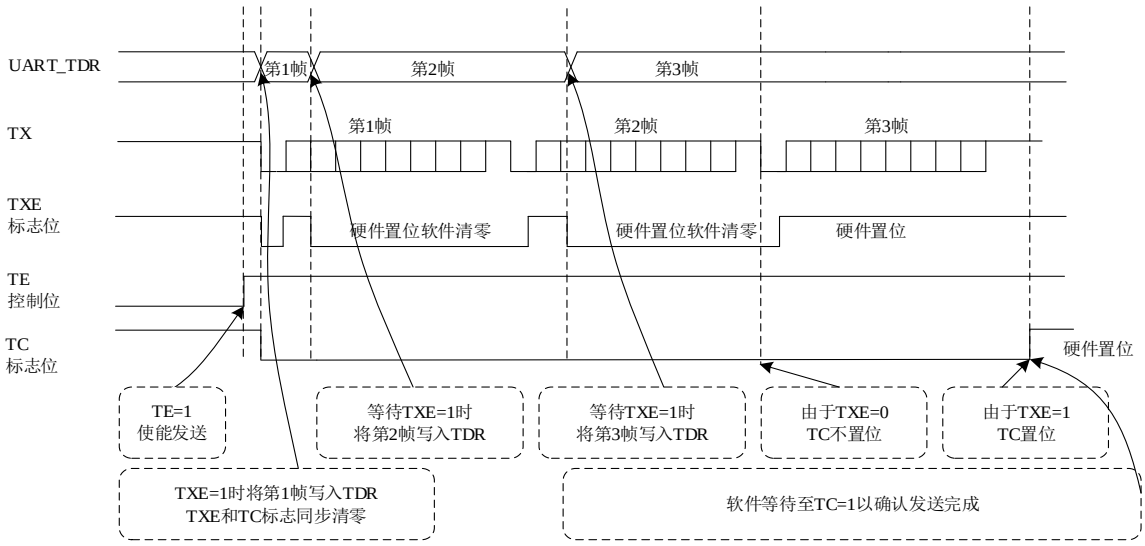
- 数据已由 UARTx_TDR 寄存器移入移位寄存器，开始发送数据；
- 可以在 UARTx_TDR 寄存器中写入数据，前一数据不会被覆盖。

TXEIE 位置位使能中断时，TXE 标志置位会触发中断。

在发送未进行时（TE=0），写入 UARTx_TDR 寄存器后发送数据非空标志 TXE 位清零，数据发送开始后，TXE 位立即置位，指示可以写入下一数据。

发送时的 TC 和 TXE 的动作请参见下图：

图 28-4 发送过程 TC 和 TXE 的动作




## 28.3.6 接收器

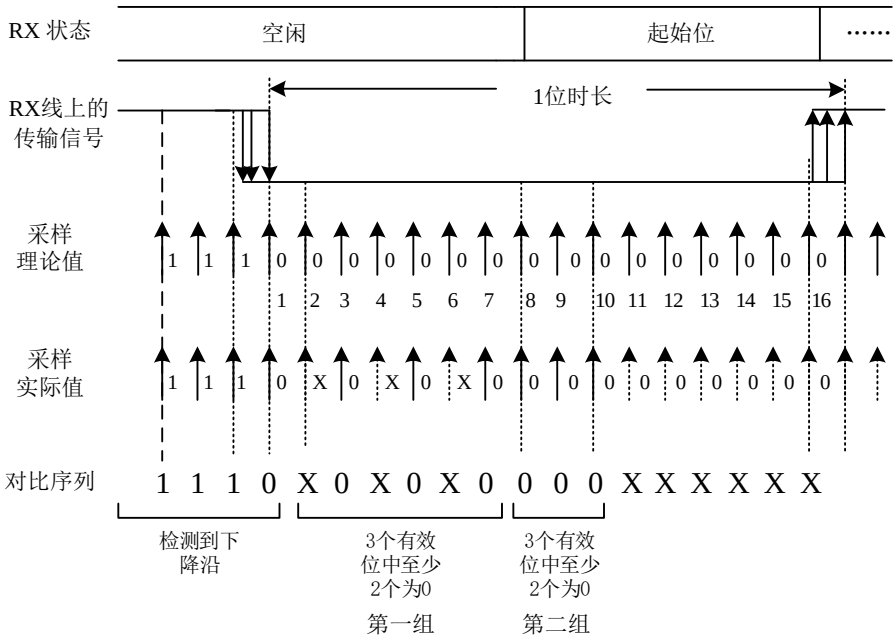
UART 可接收 8 位或 9 位长度的字符，可在 UARTx_CR1 寄存器中 WL 位配置。

**起始位检测**

在 UART 中识别出特定序列的采样结果时识别为起始位。

此序列为：111 0 X 0 X 0 X 0 0 0 0 X X X X X X

图 28-5 起始位检测




如果第一组 3 个采样位均为“0”，即第 3 、5 和第 7 位首次采样时检测到的全为“0”，第二组 3 个采样位，第 8、9 和第 10 位仍检测到全为“0”，可确认接收到起始位。

满足以下条件时，起始位有效，噪声标志位 NOISE 置位：

- 两组采样的 3 个采样位中只有两位为 0 时。 例如，第 3 、5 、7 位采样结果为“010”，第 8、9、10 位采样结果为“001”。
- 两组采样中任意一组的 3 个采样位中有两位为 0 时。例如，对第 3、5、7 位进行采样为“000”，但对第 8、9、10 位采样结果为“001”。

上述描述包含了检测到起始位无噪声和检测到起始位含噪声两种情况，如果两种情况均不满足，则起始位检测失败，检测终止，接收器返回空闲状态，无标志位置位。

**字符接收**

接收配置如下：

1) 配置 UARTx_BRR 寄存器选择通信波特率；
2) 配置 UARTx_CR1 中的 WL 位定义字符长度；

3) 配置 UARTx_CR2 中的 STOPBIT 位域设置停止位数；
4) 向 UARTx_CR1 寄存器中的 UE 位写入“1”使能 UART；
5) 使用 DMA 通信时，将 UARTx_CR3 中的 DMA 接收使能位 DMAR 写“1”；参见 *UART 使用 DMA 通信*中的说明配置 DMA 寄存器；
6) UARTx_CR1 寄存器 RE 位置位，使能接收。

接收字符时：

- 不使用 DMA 通信：

  接收非空标志 RXNE 置位指示 UARTx_RDR 寄存器已接收到数据。

  由于 UARTx_RDR 寄存器中仅有 1 字符空间，读 UARTx_RDR 寄存器时 RXNE 标志位清零。

- 使用 DMA 通信：

  由于 UARTx_RDR 寄存器中仅有 1 字节空间，每收到 1 个字符，接收非空标志位 RXNE 都会置位，DMA 读取 UARTx_RDR 后 RXNE 位清零。

- 接收期间如果检测到帧错误、噪声错误、上溢错误或奇偶校验错误，相应错误标志位置位。

**上溢错误**

上溢错误检测默认开启，当检测到上溢错误时，上溢错误标志 ORE 位置位，如果 UARTx_CR3 中的 EIE 位置位，则触发中断。

通过向 UARTx_ICR 寄存器中的 ORECF 位写 1 来清零 ORE 位。

可以通过将 UARTx_CR3 中的 ORED 位置位禁止上溢错误检测，禁止后 ORE 标志保持为 0。

每接收到一个字符后，接收非空标志位 RXNE 置位。当 RXNE 标志为 1 时，或上一个 DMA 读请求未处理时，又接收到新的字符，就会产生上溢错误。上溢错误标志 ORE 位清零前，数据无法从移位寄存器传到接收数据寄存器 UARTx_RDR 中。

发生上溢错误时有以下动作：

- 上溢错误标志 ORE 位置位；
- 移位寄存器被新数据覆盖，ORE 位清零前接收到的数据都将丢失；

- 上溢错误发生后，在 ORE 位清零前，UARTx_RDR 寄存器中的内容不会被覆盖。读取 UARTx_RDR 寄存器时，得到溢出前接收到的最后一个数据；
- UARTx_CR3 中的 EIE 位置位时，会触发中断。

ORE 位置位时表示至少已有一个数据丢失。

**过采样模式**

接收器配备了 16 倍过采样功能，可在噪声环境中提取有效数据。

采样时钟是波特率时钟的 16 倍，参见图：数据采样（OBS=0）。对每个 bit 采样 16 次，此情况下最大通信速率为 UARTx_PCLK/16。

可通过 UARTx_CR3 寄存器中的 OBS 位选择采样逻辑电平的方式，有以下两种选项：

- 在接收位的中心进行 3 次采样，使用多数表决的方式确定逻辑值。此情况下，如果参与多数表决的 3 次采样结果不同，则 NOISE 标志位置位。
- 在接收位的中心进行 1 次采样，无 NOISE 标志输出。

根据应用场景可选：

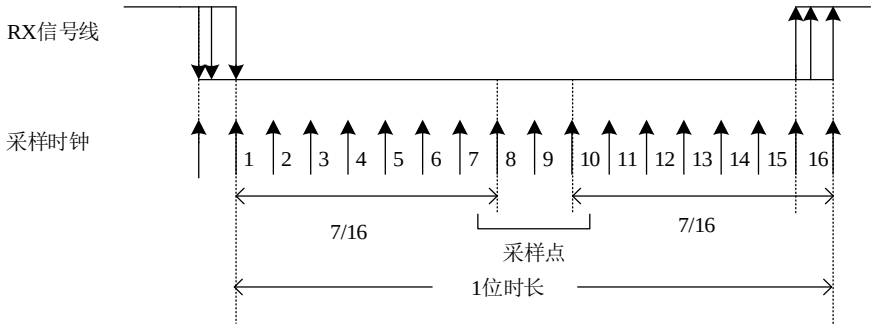
- 噪声较大的环境下，建议选择三次采样多数表决法（OBS=0）；如果检测到了噪声，则表示采样过程中有噪声干扰，为保证数据准确，推荐丢弃检测到噪声的数据。
- 在线路无噪声影响时，可选择单次采样法(OBS=1)，可以提高接收容差(参见 *UART 接收容差*章节)。

接收数据检测到噪声时：

- 在接收非空标志 RXNE 位置位的上升沿时 NOISE 位置位。
- 无效数据从移位寄存器传送到 UARTx_RDR 寄存器。
- NOISE 位置位时，如果 UARTx_CR3 寄存器中的 EIE 位置位则触发中断。NOISE 位置位时，RXNE 位同时置位，如果 RXNEIE 位置位，则触发中断。

使用 UARTx_ICR 寄存器中的 NOISECF 位写“1”清零 NOISE 位。

图 28-6 数据采样（OBS=0）




**帧错误**

接收数据时如果在预期时间内未能识别出停止位，则检测为帧错误，同步失败或噪声过大的情况下可能引发此错误。

检测到帧错误时有以下动作：

- 帧错误标志 FE 位置位。
- 无效数据从移位寄存器传送到 UARTx_RDR 寄存器。
- FE 位置位时，如果 UARTx_CR3 寄存器中的 EIE 位置位，则触发错误中断。FE 置位时，RXNE 位也被置位，如果 RXNEIE 位置位会触发接收非空中断。

向 UARTx_ICR 寄存器中的 FECF 位写 1，清零 FE 位。

**配置接收停止位**

在 UARTx_CR2 中的 STOPBIT 位域中，配置预计接收的停止位的位数。

采样逻辑电平的方式选择 3 次采样即 OBS=0 时，有以下几个选项：

- 1 位停止位：在第 8、第 9 和第 10 个采样沿采样 1 位停止位。
- 2 位停止位：在第一个停止位的第 8、第 9 和第 10 个采样沿采样。

采样逻辑电平的方式选择 1 次采样，即 OBS=1 时，在第一个停止位的中间位置采样一次。

### 28.3.7 波特率生成

接收和发送的波特率通过编程 UARTx_BRR 寄存器得到。

计算公式如下：

标准 UART 的波特率，计算公式如下：

$$\text{Tx或Rx的波特率} = \frac{UART_PCLK}{BRR}$$

BRR 是一个无符号小数，存储在 UARTx_BRR 寄存器中。

*注意：　BRR 必须大于等于 16。*

例：在 UARTx_PCLK = 72MHz 时得到 115.2K 的波特率

BRR = 72 000 000/115 200

BRR = 0d625 = 0x271

### 28.3.8 接收容差

仅当时钟系统总偏差小于 UART 接收容差时，接收器才可以正常工作。影响系统总时钟偏差的因素有以下几点：

- DTR_ERR：由发送器误差引起的偏差，其中包含发送器本地振荡器的偏差。
- DQU_ERR：接收器的波特率量化误差。
- DRX_OSERR：接收器本地振荡器的偏差。
- DL_ERR：传输线路引入的偏差（通常是由于收发器引入了上升沿下降沿的时序不对称）

  需以下条件:

  DTR_ERR+DQU_ERR+DRX_OSERR+DL_ERR<UART 接收容差。

影响接收容差的条件有以下几点：

- 8 或 9 位字符长度，使用 8 位字符长度时接收容差更大。
- BRR[3:0]位是否等于“0000”。由于需要支持过采样时钟，所以约束 BRR 不小于 16，反映到 UARTx_BRR 寄存器中为“0x10h”。BRR[3:0]如果等于“0000”则 BRR 为 16 的整数倍，BRR[3:0]不等于“0000”时引入了小数部分，所以 BRR[3:0]等于“0000”时接收容差更大。
- 单次采样（OBS=1）或 3 次采样多数表决（OBS=0）的方式，使用单次采样方式时接收容差更大。

### 28.3.9 单线半双工通信

UART 支持单线半双工通信协议，通过将 UARTx_CR3 寄存器中的 HDEN 位置位可开启单线半双工通信模式。

向 HDEN 位写 1 使能单线半双工模式，有以下动作：

- TX 和 RX 信号在内部相连接。

  默认使用 TX 引脚进行数据收发，RX 引脚不使用，可做 GPIO 使用。

  当 UARTx_CR2 中的 SWAP 为置位时，TX/RX 引脚交换使能，使用 RX 引脚进行数据收发。

- 无数据传输时 TX 处于接收状态。

  需要将 TX 引脚配置为复用开漏模式，并配置内置上拉电阻，或连接外部上拉电阻。

### 28.3.10 使用 DMA 通信

UART 支持通过 DMA 进行通信。接收和发送有独立的 DMA 请求。

**DMA 发送**

将 UARTx_CR3 寄存器中的 DMAT 位置位，使能 DMA 发送。当 TXE 标志置位时，DMA 可将数据由预设存储器加载到 UARTx_TDR 寄存器。参考以下步骤建立从 UART 到 DMA 的映射（x 表示通道编号）：

1) 在 DMA 通道 x 控制寄存器 DMA_CCx 中配置通道的相关信息，参见 *DMA 配置流程*；
2) 在 DMA 通道 x 待传输次数寄存器 DMA_CNDTRx 中，配置需要发送的总字节数；
3) 在 DMA 源地址寄存器 DMA_CSARx 中，配置存储区地址，作为传输源地址；在 DMA 目的地址寄存器 DMA_CDARx 中，写入 UARTx_TDR 寄存器地址，作传输目的地址；每次 TXE 位置位时，数据都从存储区加载到 UARTx_TDR 寄存器中；
4) 根据具体应用配置中断触发的条件；
5) 向 UARTx_ICR 寄存器中的 TCCF 位写 1，清零 UARTx_ISR 寄存器中的 TC 标志；
6) 将 DMA_CCx 寄存器中的 EN 位置位以使能通道。

当达到 DMA 控制器中设置传输数据次数时，在 DMA 相应通道上可以产生中断。

在发送模式下，DMA 完成了对所有需要发送的数据的写操作时，DMA_ISR 寄存器中的 TFx 标志置位，此外，还可以通过检测 UARTx_ISR 寄存器中的 TC 标志位确认 UART 通信是否完成。TC 标志在数据发送期间保持为 0，在最后一个字符发送结束后自动置位。

**DMA 接收**

将 UARTx_CR3 寄存器中的 DMAR 位置位，使能 DMA 接收。

接收字符时，数据从 UARTx_RDR 寄存器加载到 DMA 配置的存储区域中。参考以下步骤建立从 UART 到 DMA 的映射（x 表示通道编号）：

1) 在 DMA 通道 x 控制寄存器 DMA_CCx 中配置通道的相关信息，参见 *DMA 配置流程*；
2) 在 DMA 通道 x 待传输次数寄存器 DMA_CNDTRx 中，配置需要接收的总字节数；
3) 在 DMA 源地址寄存器 DMA_CSARx 中，写入 UARTx_RDR 寄存器地址，作为传输源地址；在 DMA 目的地址寄存器 DMA_CDARx 中，配置存储区地址，作传输目的地址；每次发生接收非空 RXNE 置位时，数据都从源地址 UARTx_RDR 复制到 DMA 配置的接收存储区中；
4) 根据具体应用配置触发中断的条件；
5) 将 DMA_CCx 寄存器中的 EN 位置位以使能通道。

**DMA 通信模式下的错误标志及中断**

在 UARTx_CR3 寄存器的 DDRE 位可配置在接收出错时禁止 DMA。

默认 DDRE=0，接收出错时不禁止 DMA，相应的错误标志置位，但 RXNE 保持为 0，不产生 DMA 请求，错误数据不会被接收到 UARTx_RDR 寄存器，但下一个正确数据可以被接收。

DDRE=1 时，接收出错后相应的错误标志和 RXNE 均置位，错误数据会被接收到 UARTx_RDR 寄存器。但 DMA 请求会被屏蔽。此种情况下接收出错时，软件需将 RXNE 清零，然后再将错误标志清除后才可以继续进行 DMA 接收。

当发生上溢错误时，上溢错误标志 ORE 位置位期间，接收到的数据只会刷新移位寄存器，不会覆盖 RDR 中的最后一个数据，所以无论 DDRE 位是否置位，都需要将 RXNE 清零，再清除错误标志后才可以继续接收数据。

## 28.4 UART 中断

UART_IRQ 为 UART 中断请求信号，中断请求信号源详见下表：

表 28-2 中断请求

| 中断事件 | 事件标志 | 中断使能位 | 事件标志/中断清除方法 |
|---|---|---|---|
| 发送数据寄存器空 | TXE | TXEIE | 向 UARTx_TDR 寄存器写入数据 |
| 接收数据寄存器非空 | RXNE | RXNEIE | 读取 UARTx_RDR 寄存器 |
| 发送数据寄存器空且移位寄存器空 | TC | TCIE | 向 UARTx_TDR 写入数据或向 TCCF 位写 1 |
| 奇偶校验错误 | PE | PEIE | 向 PECF 位写 1 |
| 上溢错误 | ORE | EIE | 向 ORECF 位写 1 |
| 检测到噪声 | NOISE | | 向 NOISECF 位写 1 |
| 检测到帧错误 | FE | | 向 FECF 位写 1 |

## 28.5 UART 寄存器

UART 寄存器支持 32 位访问。

表 28-3 UARTx 基地址

| 外设 | 基地址 |
|---|---|
| UART1 | 0x4001 3800 |
| UART2 | 0x4000 4400 |

### 28.5.1 UART 控制寄存器 1（UARTx_CR1）

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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | WL | Res. | PEN | PTS | PEIE | TXE IE | TCIE | RXNE IE | Res. | TE | RE | Res. | UE |
| | | | rw | | rw | rw | rw | rw | rw | rw | | rw | rw | | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:13 | 保留 | 写入无效 |
| 12 | WL | 字符长度<br>0：8位<br>1：9位<br>*注意：在禁止UART（UE=0）时才能写入此位。* |
| 11 | 保留 | 写入无效 |
| 10 | PEN | 奇偶校验控制使能<br>此位使能硬件奇偶校验控制，包括发送时生成校验位和接收时对校验位的检测。使能奇偶校验时，校验位位于数据寄存器的MSB（参见*UART帧格式*）。<br>此位置位后，奇偶校验从下一字节生效。<br>0：禁止<br>1：使能 |

| | | |
|---|---|---|
| | | *注意：使能奇偶校验，占用1个数据位。*<br>*在禁止UART（UE=0）时才能写入此位。* |
| 9 | PTS | 奇偶校验选择<br>此位用于在奇偶校验使能时（PEN位置位）时，选择奇校验或偶校验。此位选择后从下一字节开始生效。<br>0：偶校验<br>1：奇校验<br>*注意：在禁止UART（UE=0）时才能写入此位。* |
| 8 | PEIE | 奇偶校验错误中断使能<br>0：禁止<br>1：使能 |
| 7 | TXEIE | 发送数据寄存器空中断使能<br>0：禁止<br>1：使能 |
| 6 | TCIE | 传输完成中断使能<br>0：禁止<br>1：使能 |
| 5 | RXNEIE | 接收数据非空中断使能<br>0：禁止<br>1：使能 |
| 4 | 保留 | 写入无效 |
| 3 | TE | 发送器使能<br>0：禁止<br>1：使能 |
| 2 | RE | 接收器使能<br>0：禁止<br>1：使能 |
| 1 | 保留 | 写入无效 |

| | | |
|---|---|---|
| 0 | UE | UART 使能<br>此位清零后，UART 配置保留；UARTx_TDR、UARTx_RDR 寄存器清零，UARTx_ISR 的所有状态标志均被复位。<br>0：禁止<br>1：使能 |

### 28.5.2 UART 控制寄存器 2（UARTx_CR2）

偏移地址：0x04

复位值：0x0000 0000

*注意：　在禁止 UART（UE=0）时才能修改此寄存器。*

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | MSB FIRST | DATAIVC | TXIVC | RXIVC |
| | | | | | | | | | | | | rw | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SWAP | Res. | STOPBIT | Res. | | | | | | | | | | | | |
| rw | | rw | | | | | | | | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:20 | 保留 | 写入无效 |
| 19 | MSBFIRST | MSB优先<br>0：发送\接收时LSB优先<br>1：发送\接收MSB优先 |
| 18 | DATAIVC | 数据极性反相<br>0：按正向逻辑处理发送/接收的数据（1=H，0=L）<br>1：按反向逻辑处理发送/接收数的数据（1=L，0=H）（奇偶校验位也取反。） |
| 17 | TXIVC | TX引脚有效电平反相<br>0：TX引脚使用标准逻辑电平（$V_{DD}$ = 1，GND = 0）<br>1：TX引脚信号反向（$V_{DD}$ = 0，GND=1） |
| 16 | RXIVC | RX引脚有效电平反相<br>0：RX引脚信号使用标准逻辑电平（$V_{DD}$ = 1，GND = 0）<br>1：对RX引脚信号反相（$V_{DD}$ = 0，GND =1） |

| | | |
|---|---|---|
| 15 | SWAP | 交换TX/RX引脚<br>0：按标准引脚排列使用TX/RX引脚<br>1：TX和RX引脚交换 |
| 14 | 保留 | 写入无效 |
| 13 | STOPBIT | 停止位位数<br>0：1位<br>1：2位 |
| 12:0 | 保留 | 写入无效 |

### 28.5.3 UART 控制寄存器 3（UARTx_CR3）

偏移地址：0x08

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | DDRE | ORED | OBS | Res. | | | DMAT | DMAR | Res. | | HDEN | Res. | | EIE |
| | | rw | rw | rw | | | | rw | rw | | | rw | | | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:14 | 保留 | 写入无效 |
| 13 | DDRE | 接收出错时禁止DMA<br>0：不禁止DMA<br>相应错误标志置位，但RXNE保持为0，所以不产生DMA请求。<br>1：禁止DMA<br>相应的错误标志以及RXNE均置位。不产生DMA请求，直到错误标志清零。<br>*注意：接收错误包括：奇偶校验错误、帧错误或噪声错误。*<br>*在禁止UART（UE=0）时才能写入此位。* |

| | | |
|---|---|---|
| 12 | ORED | 禁止接收上溢<br>0：使能上溢检测<br>接收数据未读取（RXNE=1）时，又接收到新数据时，上溢错误标志位ORE置位<br>1：禁止上溢功能，ORE标志不会置位<br>*注意：在禁止UART（UE=0）时才能写入此位。* |
| 11 | OBS | 采样方式选择<br>0：对每个bit进行三次采样，使用多数表决的方法确认采样值<br>1：对每个bit进行单次采样，无噪声检测功能<br>*注意：OBS功能仅适用于数据位采样和停止位的采样，不适用于起始位的采样。*<br>*在禁止UART（UE=0）时才能写入此位。* |
| 10:8 | 保留 | 写入无效 |
| 7 | DMAT | 使能DMA发送<br>0：禁止<br>1：使能 |
| 6 | DMAR | 使能DMA接收<br>0：禁止<br>1：使能 |
| 5:4 | 保留 | 写入无效 |
| 3 | HDEN | 使能单线半双工<br>0：禁止<br>1：使能<br>*注意：在禁止UART（UE=0）时才能写入此位。* |
| 2:1 | 保留 | 写入无效 |
| 0 | EIE | 错误中断使能<br>如果发生帧错误、上溢错误或噪声错误时（UARTx_ISR寄存器中FE = 1、ORE = 1或NOISE = 1）时，此位置位会触发中断。<br>0：禁止<br>1：使能 |

### 28.5.4 UART 波特率分频寄存器（UARTx_BRR）

偏移地址：0x0C

复位值：0x0000 0000

*注意：　在禁止 UART（UE=0）时才能修改此寄存器。*

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:16 | 保留 | 写入无效 |
| 15:0 | BRR[15:0] | BRR[15:0] 位域为波特率分频值。<br>详情参见 *UART 波特率生成。* |

### 28.5.5 UART 中断和状态寄存器（UARTx_ISR）

偏移地址：0x1C

复位值：0x0000 00C0

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | TXE | TC | RXNE | Res. | ORE | NOISE | FE | PE |
| | | | | | | | | r | r | r | | r | r | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:17 | 保留 | 写入无效 |
| 16 | BUSY | 忙标志<br>当 RX 线路有接收的数据时，此位置位。<br>在接收结束时，无论是否成功，此位均清零。<br>0：UART 处于空闲状态 |

| | | |
|---|---|---|
| | | 1：正在接收数据 |
| 15:8 | 保留 | 写入无效 |
| 7 | TXE | 发送数据寄存器为空<br>当 UARTx_TDR 寄存器的内容已发送时，TXE 置位。<br>写入 UARTx_TDR 寄存器时，此位清零。<br>0：发送数据寄存器 UARTx_TDR 非空<br>1：发送数据寄存器 UARTx_TDR 为空 |
| 6 | TC | 发送完成标志<br>此标志位置位，表示写入到UARTx_TDR中的最后一个数据已从移位寄存器中发出。如果已完成所有数据的发送，且TXE置位，则此位置位。<br>向UARTx_ICR寄存器中的TCCF位写1，或写入UARTx_TDR寄存器时，此位清零。<br>0：发送未完成<br>1：发送已完成 |
| 5 | RXNE | 接收数据寄存器非空<br>当接收到数据时，RXNE 位置位。<br>读取 UARTx_RDR 寄存器时，此位清零。<br>0：未接收到数据<br>1：已接收到数据 |
| 4 | 保留 | 写入无效 |
| 3 | ORE | 上溢错误标志<br>当 RXNE 标志为 1 时，或上一个 DMA 读请求未处理时，又接收到新的字符，就会产生上溢错误，则此位置位。<br>向 UARTx_ICR 寄存器中的 ORECF 位写 1，此位清零。<br>0：无上溢错误<br>1：检测到上溢错误<br>*注意：UARTx_CR3 寄存器中的ORED 位置位时，上溢错误检测被禁止，则此位清零（无上溢检测）。* |
| 2 | NOISE | 噪声检测标志<br>当接收中检测到噪声时，此位自动置位。 |

向 UARTx_ICR 寄存器中的 NOISECF 位写 1，此位清零。

0：未检测到噪声

1：检测到噪声

*注意：当线路无噪声时，可以使用单次采样的方式（OBS=1）来提高UART的接收容差，此时无噪声检测，无NOISE标志输出。*

| | | |
|---|---|---|
| 1 | FE | 帧错误标志<br>接收数据时，如果在预期时间内未能识别出停止位，则检测为帧错误，此位置位。<br>向 UARTx_ICR 寄存器中的 FECF 位写 1，此位清零。<br>0：未检测到帧错误<br>1：检测到帧错误 |
| 0 | PE | 奇偶校验错误标志<br>当在接收时发生奇偶校验错误时，此位置位。<br>向 UARTx_ICR 寄存器中的 PECF 位写 1，此位清零。<br>0：无奇偶校验错误<br>1：奇偶校验错误 |

### 28.5.6 UART 中断标志清零寄存器（UARTx_ICR）

偏移地址：0x20

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | TCCF | Res. | | ORECF | NOISECF | FECF | PECF |
| | | | | | | | | | w | | | w | w | w | w |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:7 | 保留 | 写入无效 |
| 6 | TCCF | 发送完成标志清零<br>向此位写1，UARTx_ISR寄存器中的TC标志将清零。 |
| 5:4 | 保留 | 写入无效 |

| | | |
|---|---|---|
| 3 | ORECF | 上溢错误标志清零<br>向此位写1，UARTx_ISR寄存器中的ORE标志将清零。 |
| 2 | NOISECF | 噪声检测标志清零<br>向此位写1，UARTx_ISR寄存器中的NOISE标志将清零。 |
| 1 | FECF | 帧错误标志清零<br>向此位写1，UARTx_ISR寄存器中的FE标志将清零。 |
| 0 | PECF | 奇偶校验错误标志清零<br>向此位写1，UARTx_ISR寄存器中的PE标志将清零。 |

### 28.5.7 UART 接收数据寄存器（UARTx_RDR）

偏移地址：0x24

复位值：0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:9 | 保留 | 写入无效 |
| 8:0 | RDR[8:0] | 接收数据寄存器<br>在使能奇偶校验的时，接收数据的 MSB 为奇偶校验位。 |

### 28.5.8 UART 发送数据寄存器（UARTx_TDR）

偏移地址：0x28

复位值：0x0000 0000

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

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

| Res. | | | | | | | TDR[8:0] | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | rw | rw | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:9 | 保留 | 写入无效 |
| 8:0 | TDR[8:0] | 发送数据寄存器<br>在使能奇偶校验时，发送数据中写入的 MSB 自动填充为奇偶校验位，此时写入 MSB 的值无效。 |

# 29 低功耗通用异步收发器（LPUART）

## 29.1 简介

低功耗异步收发器 LPUART 支持全双工或半双工的通信方式，内部集成小数级波特率发生器，波特率范围可配置。

LPUART 仅用 32.768 KHz 的 LXTAL 低速时钟源就能实现 9600bps 的通信波特率。使用速率更高的时钟源时，可支持更高的通信波特率。

当芯片处于 Stop 模式时，LPUART 可以收发数据，支持 Stop 唤醒。

LPUART 支持 DMA 通信、单线半双工通信和硬件流控。

提供 1 个低功耗异步收发器 LPUART。

## 29.2 主要特性

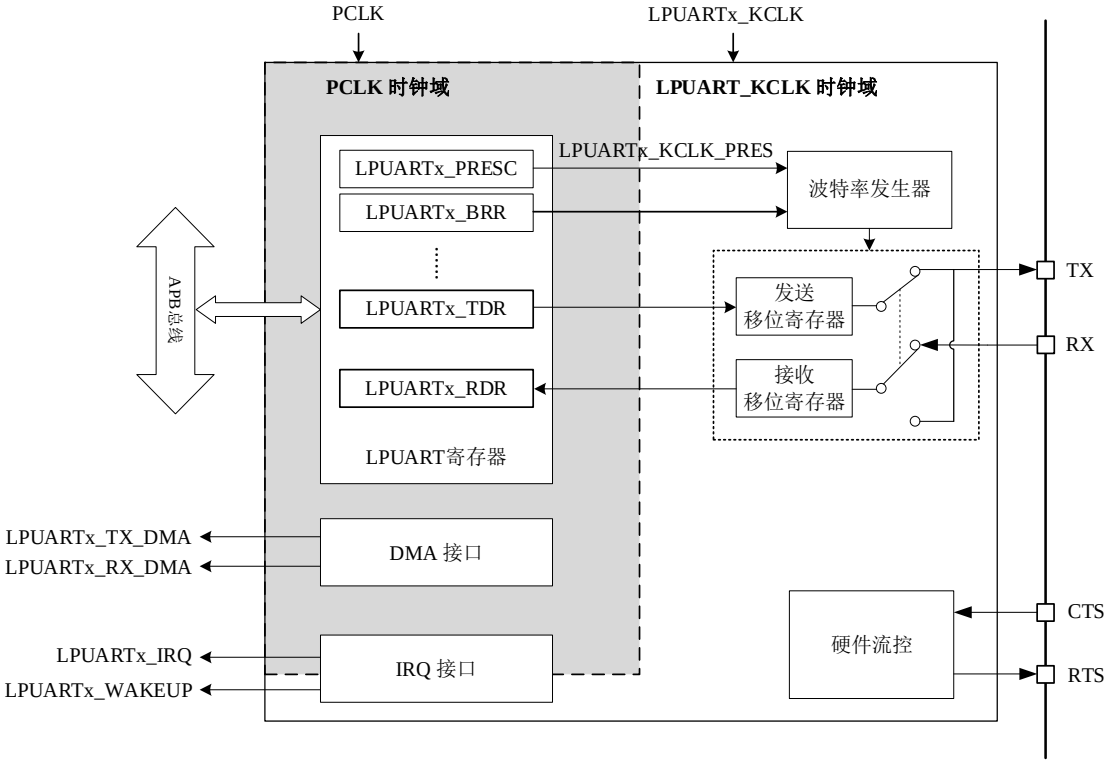
- 可编程波特率发生器
  - 32.768 KHz 时钟源下支持波特率范围为 300bps~9600bps
  - 在频率更高的时钟源下可实现更高的波特率，最高可达 24Mbps
  - 小数级波特率发生器
- 时钟源可配置为 PCLK、LXTAL、RCH 或 RCL
- 支持低功耗模式唤醒，唤醒源可配置
- 帧格式：
  - 数据长度：8 位或 9 位
  - 奇校验、偶校验和无校验
  - 1 位或 2 位停止位
  - MSB 或 LSB 优先
- 单线半双工通信
- 硬件流控
- Tx/Rx 引脚配置可交换

- 发送/接收的信号极性可独立控制
- 支持 DMA 通信

## 29.3 LPUART 功能描述

### 29.3.1 LPUART 框图

图 29-1 LPUARTx 框图（x = 1）




如上图所示，LPUART 工作在两个时钟域：

- PCLK 时钟域的时钟源与 APB 总线时钟相同，修改寄存器时需提供 PCLK。
- LPUARTx_KCLK 时钟源由 RCC 提供，可通过*外设异步时钟配置寄存器（RCC_CLKSEL）*中 LPUARTx_SEL[1:0]位域配置，可选择 PCLK、LXTAL、RCH 或 RCL。当时钟源配置为 RCH 或 LXTAL 或 RCL 时，支持从 Stop 模式唤醒。

### 29.3.2 LPUART 引脚

LPUART 使用 2 个引脚与外部器件通信：数据接收（RX）和数据发送（TX）。

支持 TX 引脚 RX 引脚互换功能，在 LPUARTx_CR2 的 SWAP 位中可配置。

RS232 流控中需要用到以下引脚：

- CTS（Clear To Send）：

  此引脚为输入，接收到高电平时，当前传输完成后数据发送被阻止。

- RTS（Request To Send）：

  此引脚为输出，LPUART 准备好接收数据时，将此引脚拉低。

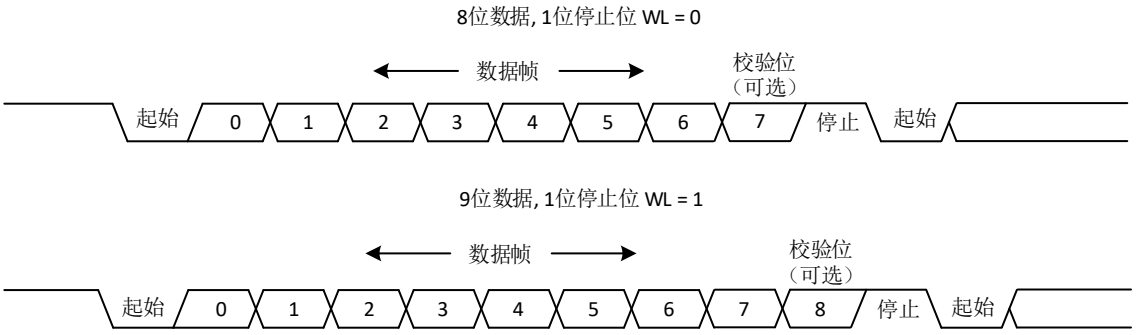
### 29.3.3 帧格式

可通过 LPUARTx_CR1 寄存器中的 WL 位配置字符长度，参见帧格式示意图。

- 8 位字符长度：WL=“0”
- 9 位字符长度：WL=“1”

默认情况下，TX 和 RX 引脚在起始位期间都是低电平，停止位期间都是高电平。发送和接收信号的极性可独立配置是否翻转，参见 *寄存器 LPUARTx_CR2* 中 TXIVC，RXIVC 位的说明。详见下图：

图 29-2 帧格式示意图




### 29.3.4 校验控制

表 29-1 LPUART 校验控制

| 字符长度 | | 校验控制 PEN | LPUART 帧格式 |
|---|---|---|---|
| 8 位 | WL=0 | 0 | \| 起始位 \|8 位数据 \| 停止位 \| |
| | | 1 | \| 起始位 \|7 位数据 \| 校验位 \| 停止位 \| |
| 9 位 | WL=1 | 0 | \| 起始位 \|9 位数据 \| 停止位 \| |
| | | 1 | \| 起始位 \|8 位数据 \| 校验位 \| 停止位 \| |

**偶校验**

使能偶校验时，一个字符中（含校验位）“1”的总数为偶数。

**奇校验**

使能奇校验时，一个字符中（含校验位）“1”的总数为奇数。

**接收检测奇偶校验**

LPUARTx_CR1 寄存器中的 PEN 位置位使能奇偶校验，使能后，如果奇偶校验失败，则 LPUARTx_ISR 寄存器中的奇偶校验错误标志 PE 位置位，如果 LPUARTx_CR1 寄存器中 PEIE 位置位，则触发中断。向 LPUARTx_ICR 寄存器中的 PECF 位写 1，可清零 PE 标志。

**发送生成奇偶校验**

LPUARTx_CR1 寄存器中的 PEN 位置位使能奇偶校验，使能后，数据寄存器中数据的 MSB 会替换为奇偶校验位。如果选择偶校验(PTS=0)，则一个字符中“1”的总数为偶数；如果选择奇校验(PTS=1)，则一个字符中“1”的总数为奇数。

### 29.3.5 发送器

根据 WL 位的配置，发送的字符长度可以是 8 位或 9 位。发送使能位 TE 置位即可使能发送器。发送移位寄存器的数据由 TX 引脚输出。

**字符发送**

通过 LPUARTx_CR1 寄存器的 TE 位置位使能发送器。

LPUART 发送期间，TX 引脚首先移出数据的最低有效位（默认 LSB 优先）。发送数据寄存器 LPUARTx_TDR 为内部总线和发送移位寄存器之间提供缓冲区。

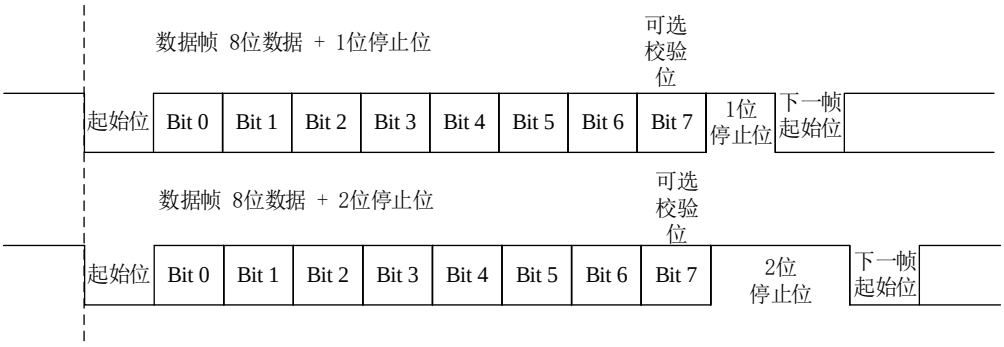
每个字符的发送都以起始位开始，以停止位结束。

**停止位**

可以在 LPUARTx_CR2 的 STOPBIT 位编程停止位的位数，可配置为 1 位或 2 位，停止位将在每一个字符后发送。

下图为不同停止位配置示例：

图 29-3 可编程停止位示例




### 字符发送配置步骤

1) 配置 LPUARTx_BRR 寄存器选择通信波特率；
2) 配置 LPUARTx_CR1 中的 WL 位定义字符长度；
3) 配置 LPUARTx_CR2 中的 STOPBIT 设置停止位数；
4) 通过向 LPUARTx_CR1 寄存器中的 UE 位写入“1”使能 LPUART；
5) 使用 DMA 通信时，将 LPUARTx_CR3 中的 DMA 发送使能位 (DMAT)写“1”；参见 *LPUART 使用 DMA 通信*中的说明配置 DMA 寄存器；
6) 将 LPUARTx_CR1 中的 TE 位置位，使能发送器；等待 TEACK 位置位，表示发送器已准备好发送；
7) 在 LPUARTx_TDR 寄存器中写入准备发送的数据。在 LPUARTx_TDR 写入数据会将 TXE 标志清零，表示 LPUARTx_TDR 发送寄存器中数据已满；
8) 当最后一个待发送的数据写入 LPUARTx_TDR 寄存器后，需等待发送完成直到 LPUARTx_ISR 中的发送完成标志位 TC 置位。TC 置位时表示最后一个字符已发送完成。

### 字符发送过程

由于 LPUARTx_TDR 寄存器只有 1 字符空间，所以写入 1 个字符后，发送数据非空标志 TXE 位清零。

发送数据寄存器 LPUARTx_TDR 为空时，TXE 标志位自动置位，此时表示：

- 数据已由 LPUARTx_TDR 寄存器移入移位寄存器，开始发送数据；
- 可以在 LPUARTx_TDR 寄存器中写入数据，前一数据不会被覆盖。

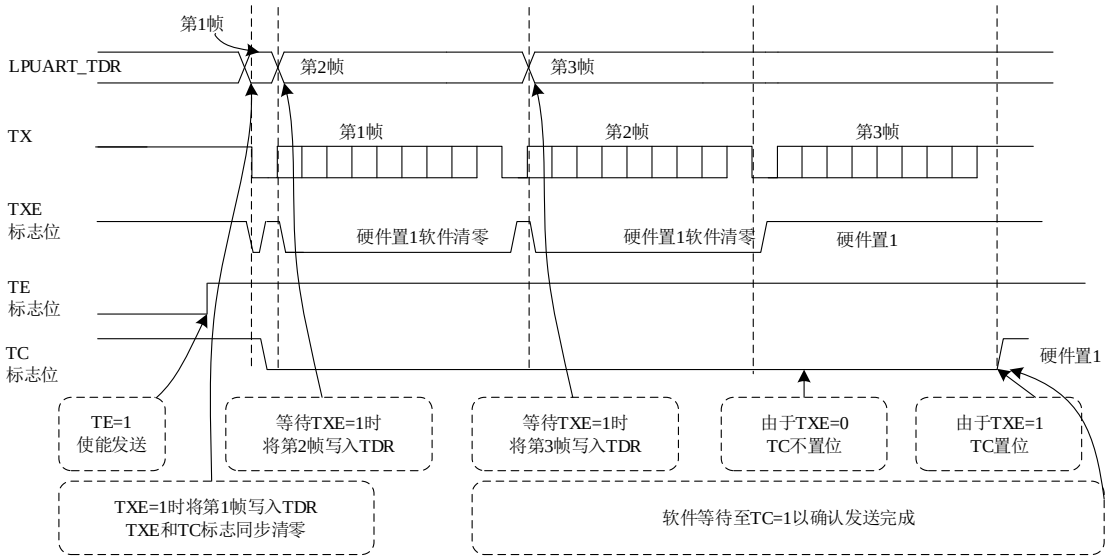
TXEIE 位置位使能中断时，TXE 标志置位时会触发中断。

在发送未进行（TE=0）时，写入 LPUARTx_TDR 寄存器后，发送数据非空标志 TXE 位清零，发送开始后，TXE 位立即置位，指示可以写入下一数据。

随着停止位发送完毕，一帧数据发送完成，如果此时发送数据非空标志 TXE 置位，表示后续没有等待发送的数据，发送过程完成，此时发送完成标志位 TC 置位。如果 LPUARTx_CR1 寄存器中 TCIE 位置位使能中断，则触发中断。

发送时的 TC 和 TXE 的动作请参见下图：

图 29-4 发送过程 TC 和 TXE 的动作




### 29.3.6 接收器

LPUART 可接收 8 位或 9 位长度的字符，可在 LPUARTx_CR1 寄存器中 WL 位配置。

#### 起始位检测

接收线上出现下降沿时，启动起始位检测，在起始位中间进行 1 次采样，以确认该位是否为起始位。如果正常检测到起始位，则继续对后续数据进行采样，否则结束起始位检测，不采样后续数据，重新等待下降沿。

#### 字符接收过程

接收配置如下：

1) 配置 LPUARTx_BRR 寄存器选择通信波特率；
2) 配置 LPUARTx_CR1 中的 WL 位定义字符长度；
3) 配置 LPUARTx_CR2 中的 STOPBIT 位设置停止位数；
4) 向 LPUARTx_CR1 寄存器中的 UE 位写 1 使能 LPUART；
5) 使用 DMA 通信时，将 LPUARTx_CR3 中的 DMA 接收使能位（DMAR）置位。参见 *LPUART 使用 DMA 通信*中的说明配置 DMA 寄存器；

6) LPUARTx_CR1 寄存器 RE 位置位，使能接收；
7) 等待 REACK 标志置位以确认接收器准备完毕。

接收字符时：

- 不使用 DMA 通信：

  接收非空标志 RXNE 置位指示 LPUARTx_RDR 寄存器已接收到数据。

  由于 LPUARTx_RDR 寄存器中仅有 1 字符空间，读 LPUARTx_RDR 寄存器时 RXNE 标志位清零。

- 使用 DMA 通信：

  由于 LPUARTx_RDR 寄存器中仅有 1 字符空间，每收到 1 个字符，接收非空标志位 RXNE 都会置位，DMA 读取 LPUARTx_RDR 后 RXNE 位清零。

- 接收期间如果检测到帧错误、上溢错误或奇偶校验错误，相应错误标志位置位。

**上溢错误**

上溢错误检测默认开启，当检测到上溢错误时，上溢错误标志 ORE 位置位，如果中断使能，即 LPUARTx_CR3 中的 EIE 位置位，则触发中断。

通过向 LPUARTx_ICR 寄存器中的 ORECF 位写 1 来清零 ORE 位。

每接收到一个字符后，接收非空标志位 RXNE 置位。当 RXNE 标志为 1 时，或上一个 DMA 读请求未处理时，又接收到新的字符，就会产生上溢错误。RXNE 位清零前，数据无法从移位寄存器传到接收数据寄存器 LPUARTx_RDR 中。

发生上溢错误时有以下动作：

- 上溢错误标志 ORE 位置位；
- 移位寄存器被新数据覆盖，RDR 中上溢前接收到的数据不会被覆盖。ORE 位清零前，接收到的数据都将丢失。
- 上溢错误发生后，在 RXNE 位清零前，读取 LPUARTx_RDR 寄存器时，得到溢出前接收到的最后一个数据。
- LPUARTx_CR3 中的 EIE 位置位时，会触发中断。

ORE 位置位时表示至少已有 1 个数据丢失。

**帧错误**

接收数据时，如果在预期时间内未能识别出停止位，则检测为帧错误。

检测到帧错误时有以下动作：

- 帧错误标志 FE 位置位。
- 无效数据从移位寄存器传送到 LPUARTx_RDR 寄存器。
- FE 位置位时，如果 LPUARTx_CR3 寄存器中的 EIE 位置位，则触发中断。

向 LPUARTx_ICR 寄存器中的 FECF 位写 1，清零 FE 位。

**配置接收停止位**

在 LPUARTx_CR2 中的 STOPBIT 位中，配置预计接收的停止位的位数。有以下几个选项：

- 1 个停止位：在停止位的中间采样一次。
- 2 个停止位：在第二个停止位的中间位置采样一次。不对第一个停止位进行采样。

如果采样时未检测到停止位，则产生帧错误，帧错误标志位 FE 和 RXNE 标志置位。

### 29.3.7 波特率生成

通过 LPUARTx_BRR 设置发送和接收的时钟频率：

$$\text{Tx 或 Rx 的波特率} = \frac{256 * LPUART_KCLK_PRES}{BRR}$$

其中 LPUARTx_KCLK_PRES 由 LPUARTx_KCLK 时钟通过 LPUARTx_PRESC 寄存器中定义的系数进行预分频后得到。

BRR 的值在 LPUARTx_BRR 寄存器中设置。

BRR 取值范围：

$$0x100000 > BRR \geq 0x300$$

表 29-2 LPUARTx_KCLK_PRES=32.768KHz 时的波特率误差

| 波特率目标值 | 波特率实际值 | BRR 寄存器的值 | 误差（%） |
|---|---|---|---|
| 300bps | 300bps | 0x6D3A | 0 |

| 波特率目标值 | 波特率实际值 | BRR 寄存器的值 | 误差（%） |
|---|---|---|---|
| 600bps | 600bps | 0x369D | 0 |
| 1200bps | 1200.087bps | 0x1B4E | 0.007 |
| 2400bps | 2400.17 bps | 0xDA7 | 0.007 |
| 4800bps | 4801.72 bps | 0x6D3 | 0.035 |
| 9600bps | 9597.95 bps | 0x36A | 0.021 |

### 29.3.8 接收容差

仅当时钟系统总偏差小于 LPUART 接收容差时，接收器才可以正常工作。影响系统总时钟偏差的因素有以下几点：

- DTR_ERR：由发送器误差引起的偏差，其中包含发送器本地振荡器的偏差。
- DQU_ERR：接收器的波特率量化误差。
- DRX_OSERR：接收器本地振荡器的偏差。
- DL_ERR：传输线路引入的偏差（通常是由于收发器引入了上升沿下降沿的时序不对称）

需以下条件：

DTR_ERR + DQU_ERR + DRX_OSERR + DL_ERR + DWKUP < LPUART 接收容差。

其中 DWKUP 是从低功耗唤醒时，采样点偏差产生的误差。

影响接收容差的条件有以下几点：

- 8 或 9 位字符长度，使用 8 位字符长度时接收容差更大。
- 1 位或 2 位停止位，1 位停止位时接收容差更大。
- LPUARTx_BRR 寄存器的配置，BRR 寄存器赋值越大接收容差越大。

### 29.3.9 单线半双工通信

LPUART 支持单线半双工通信协议，通过将 LPUARTx_CR3 寄存器中的 HDEN 位置位可开启单线半双工通信模式，此时 TX 和 RX 信号在内部相连接。

单线半双工模式下，有以下动作：

- TX 和 RX 信号在内部相连接。

默认使用 TX 引脚进行数据收发，RX 引脚不使用，可做 GPIO 使用。

当 LPUARTx_CR2 中的 SWAP 为置位时，TX/RX 引脚交换使能，使用 RX 引脚进行数据收发。

- 无数据传输时 TX 处于接收状态。

  需要将 TX 引脚配置为复用开漏模式，并配置内置上拉电阻，或连接外部上拉电阻。

### 29.3.10 使用 DMA 通信

LPUART 支持通过 DMA 进行通信。接收和发送有独立的 DMA 请求。

**DMA 发送**

将 LPUARTx_CR3 寄存器中的 DMAT 位置位，使能 DMA 发送。当 TXE 标志置位时，DMA 可将数据由预设存储器加载到 LPUARTx_TDR 寄存器。参考以下步骤建立从 LPUART 到 DMA 的映射（x 表示通道编号）：

1) 在 DMA 通道 x 控制寄存器 DMA_CCx 中配置通道的相关信息，参见 *DMA 配置流程*。
2) 在 DMA 通道 x 待传输次数寄存器 DMA_CNDTRx 中，配置需要发送的总字节数。
3) 在 DMA 源地址寄存器 DMA_CSARx 中，配置存储区地址，作为传输源地址；在 DMA 目的地址寄存器 DMA_CDARx 中，写入 LPUARTx_TDR 寄存器地址，作传输目的地址；每次 TXE 置位时，数据都从存储区加载到 LPUARTx_TDR 寄存器中。
4) 根据具体应用配置中断触发的条件。
5) 向 LPUARTx_ICR 寄存器中的 TCCF 位写 1，清零 LPUARTx_ISR 寄存器中的 TC 标志。
6) 将 DMA_CCx 寄存器中的 EN 位置位以使能通道。

当达到 DMA 控制器中设置传输数据次数时，会在 DMA 相应通道上产生中断。

在发送模式下，DMA 完成了对所有需要发送的数据的写操作时，DMA_ISR 寄存器中的 TFx 标志置位，此外，还可以通过检测 LPUARTx_ISR 寄存器中的 TC 标志位确认 LPUART 通信是否完成。TC 标志在数据发送期间保持清零，在最后一个字符发送结束后自动置位。

**DMA 接收**

将 LPUARTx_CR3 寄存器中的 DMAR 位置位，使能 DMA 接收。

接收字符时，数据从 LPUARTx_RDR 寄存器加载到 DMA 配置的存储区域中。参考以下步骤建立从 LPUART 到 DMA 的映射（x 表示通道编号）：

1) 在 DMA 通道 x 控制寄存器 DMA_CCx 中配置通道的相关信息，参见 *DMA 配置流程*。
2) 在 DMA 通道 x 待传输次数寄存器 DMA_CNDTRx 中，配置需要接收的总字节数。
3) 在 DMA 源地址寄存器 DMA_CSARx 中，写入 LPUARTx_RDR 寄存器地址，作为传输源地址；在 DMA 目的地址寄存器 DMA_CDARx 中，配置存储区地址，作传输目的地址；每次发生接收非空 RXNE 置位时，数据都从源地址 LPUARTx_RDR 复制到 DMA 配置的接收存储区中。
4) 根据具体应用配置触发中断的条件。
5) 将 DMA_CCx 寄存器中的 EN 位置位以使能通道。

**接收出错时 DMA 的动作**

在 LPUARTx_CR3 寄存器的 DDRE 位可配置在接收出错时禁止 DMA，包括帧错误和校验错误。

默认 DDRE=0，接收出错时不禁止 DMA，相应的错误标志置位，但 RXNE 保持为 0，不产生 DMA 请求，错误数据不会被接收到 LPUARTx_RDR 寄存器，但下一个正确数据可以被接收。

DDRE=1 时，接收出错后相应的错误标志置位，错误数据会被接收到 LPUARTx_RDR 寄存器。但 DMA 请求会被屏蔽。此种情况下接收出错时，软件需将 RXNE 清零，再将错误标志清除后才可以继续进行 DMA 接收。

当发生上溢错误时，上溢错误标志 ORE 位置位期间，接收到的数据只会刷新移位寄存器，不会覆盖 RDR 中的最后一个数据，所以无论 DDRE 位是否置位，都需要将 LPUARTx_RDR 中的数据读走或做清空，再清除错误标志后才可以继续接收数据。

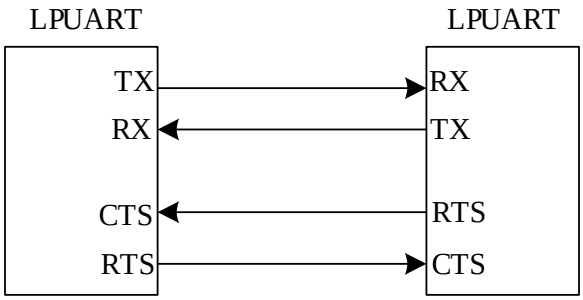
### 29.3.11 RS232 硬件流控

RS232 硬件流控 RTS/CTS 用于通信流量控制，RTS 为输出引脚，CTS 为输入引脚，低电平有效，RTS 和 CTS 引脚连接关系参见下图。

CTS 和 RTS 硬件流控可通过 LPUARTx_CR3 寄存器中的 RTSE 位和 CTSE 独立

配置，置位使能，清零禁止。

图 29-5 两个 LPUART 间流控信号连接示例

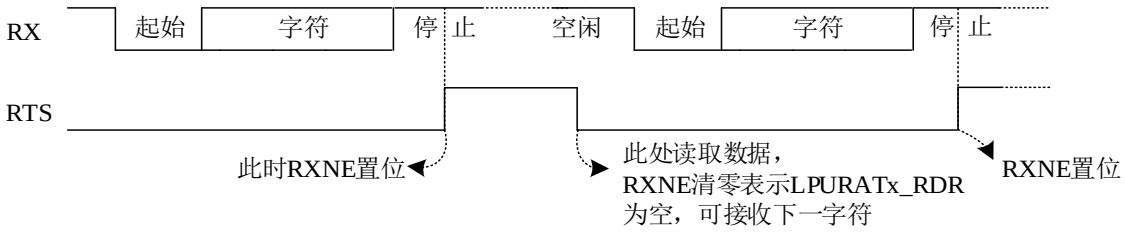



**RS232 RTS 流控**

使能 RTS 流控（RTSE=1）后，接收数据寄存器 LPUARTx_RDR 空（RXNE＝0）时，RTS 引脚输出低电平，表示可以接收数据；接收数据寄存器非空（RXNE＝1）时，RTS 引脚输出高电平，通知对端设备暂停发送数据。

RTS 流控示例参见下图：

图 29-6 RTS 动作示例



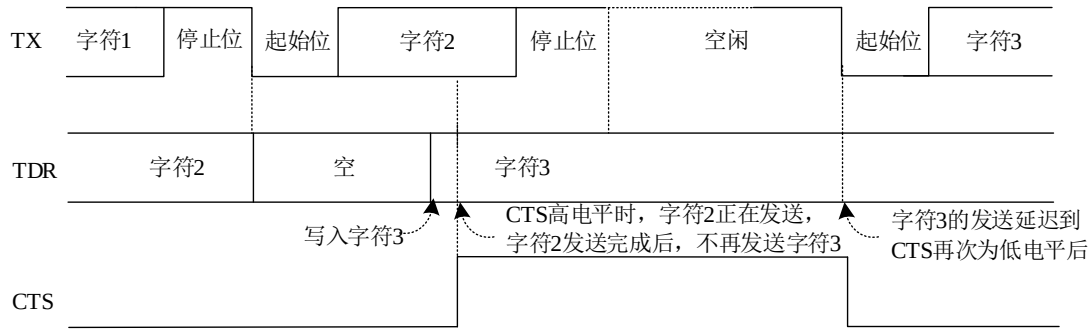

**RS232 CTS 流控**

使能 CTS 流控（CTSE=1）时，发送器会在发下一字符前检查 CTS。只要 CTS 电平发生变化（由低到高或由高到低），CTSIF 位均置位，LPUARTx_CR3 寄存器中的 CTSIE 位置位时会触发中断。

- 发送端 LPUART 发送器检测到 CTS 引脚低电平时，可发送数据。
- 发送端 LPUART 发送器检测到 CTS 引脚高电平时，当前正在发送的数据发送完成后停止发送。
- 发送端 LPUART 发送器检测到 CTS 引脚高电平时，正在写入发送数据寄存器中的数据不向外发送，当发送器再次检测到 CTS 低电平时，恢复发送。

CTS 流控通信示例参见下图：

图 29-7 CTS 动作示例




*注意：* *为保证正常工作，CTS 信号上升沿需在当前字符停止位结束前，至少提前 5 个 LPUARTx_KCLK 时钟周期。*

## 29.4 LPUART 低功耗模式

### 29.4.1 LPUART 对低功耗模式的支持

LPUART 支持在低功耗模式下工作，LPUARTx_CR1 寄存器中的 UEWK 位置 1 时，支持将芯片从 Stop 模式唤醒。

选用 Stop 模式下可以工作的时钟（RCH 或 LXTAL 或 RCL）作为 LPUART 的时钟源时，即使停用 PCLK 时钟，也可以保证数据正常传输，并支持将芯片从 Stop 模式唤醒。

LPUART_WAKEUP 信号为 LPUART 发出的唤醒信号，支持将芯片从 Stop 模式唤醒。可以通过使能相关中断来配置 LPUART_WAKEUP 信号的触发事件源，可以使用以下唤醒事件作为 LPUART_WAKEUP 信号的触发事件：

- 接收到一帧数据时唤醒，通过 RXNEIE=1 使能。
- 接收到与 LPUARTx_CR2 寄存器中 CMFD[7:0]位域匹配的字符时唤醒，通过则 CMIE=1 使能。
- LPWK 位（低功耗模式事件唤醒标志）置位也可以作为 LPUART_WAKEUP 信号的触发事件，LPWK 置位的条件为检测到起始位。

  只有将 LPWKIE 位置位时，LPWK 标志置位时才可以将芯片从低功耗模式唤醒。

### 29.4.2 LPUART 内核时钟（LPUARTx_KCLK）停止时从 Stop 模式唤醒

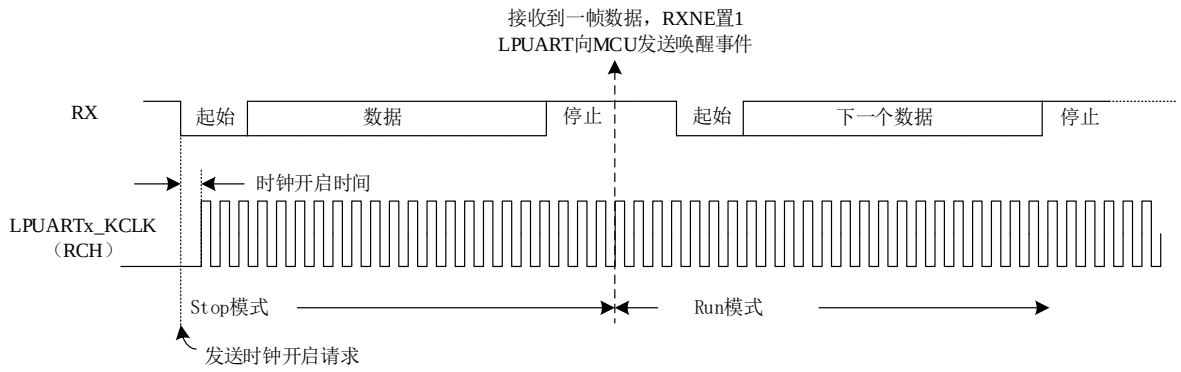
如果在 Stop 模式下 LPUARTx_KCLK 时钟停止，在 LPUART 接收线上检测到下降沿时开启 LPUARTx_KCLK 时钟。

此后如果检测到了有效的唤醒事件，芯片从 Stop 模式唤醒并正常接收数据。

如果唤醒事件无效，则 LPUARTx_KCLK 再次关闭，芯片不被唤醒，保持在 Stop 模式下。

以下示例展示了唤醒事件为接收到一帧数据的情况。

图 29-8 检测到有效唤醒事件




*注意：* *唤醒事件为起始位检测，则 LPUART 在起始位检测成功时发送唤醒事件。*

不同低功耗模式下 LPUART 外设的状态如下表所示：

表 29-3 低功耗模式说明

| 模式 | 说明 |
|---|---|
| Sleep | 对 LPUART 通信无影响，LPUART 的中断请求信号可使芯片退出 Sleep。 |
| Stop | LPUART 输出的唤醒请求信号或中断请求信号可在此模式下唤醒芯片。需将时钟源配置为 RCH 或 LXTAL 或 RCL，且将 UEWK 位置位。 |

## 29.5 LPUART 中断

LPUARTx_IRQ 为 LPUART 中断请求信号；LPUARTx_WAKEUP 为 LPUART 唤醒请求信号。

以上信号的信号源详见下表：

表 29-4 中断请求 (1)

| 中断事件 | 事件标志 | 中断使能位 | 事件标志/中断清除方法 | 输出信号 | |
|---|---|---|---|---|---|
| | | | | LPUART_IRQ | LPUART_WAKEUP |
| 发送数据寄存器空 | TXE | TXEIE | 向 LPUARTx_TDR 寄存器写入数据 | √ | × |
| 接收数据寄存器非空 | RXNE | RXNEIE | 读取 LPUARTx_RDR 寄存器 | √ | √ |
| 发送数据寄存器空且移位寄存器空 | TC | TCIE | 向 LPUARTx_TDR 寄存器写入数据或 TCCF 位写 1 | √ | × |
| CTS 电平发生变化 | CTSIF | CTSIE | 向 CTSCF 位写 1 | √ | × |
| 奇偶校验错误 | PE | PEIE | 向 PECF 位写 1 | √ | × |
| 检测到帧错误 | FE | EIE | 向 FECF 位写 1 | √ | × |
| 接收上溢错误 | ORE | | 向 ORECF 位写 1 | √ | × |
| 检测到字符匹配 | CMF | CMIE | 向 CMCF 位写 1 | √ | √ |
| 检测到唤醒事件 | LPWK | LPWKIE | 向 LPWKCF 位写 1 | √ | √ |

1. “√”表示支持此功能，“×”表示不支持此功能。

## 29.6 LPUART 寄存器

LPUART 寄存器支持 32 位访问。

表 29-5 LPUARTx 基地址

| 外设 | 基地址 |
|---|---|
| LPUART1 | 0x4000 8000 |

### 29.6.1 LPUART 控制寄存器 1（LPUARTx_CR1）

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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Res. | CMIE | Res. | WL | Res. | PEN | PTS | PEIE | TXEIE | TCIE | RXNEIE | Res. | TE | RE | UEWK | UE |
| | rw | | rw | | rw | rw | rw | rw | rw | rw | | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:15 | 保留 | 写入无效 |
| 14 | CMIE | 字符匹配中断使能<br>0：禁止<br>1：使能 |
| 13 | 保留 | 写入无效 |
| 12 | WL | 字符长度<br>0：8位<br>1：9位<br>*注意：在禁止LPUART（UE=0）时才能写入此位。* |
| 11 | 保留 | 写入无效 |
| 10 | PEN | 奇偶校验控制使能<br>0：禁止 |

| | | |
|---|---|---|
| | | 1：使能<br>*注意： 在禁止LPUART（UE=0）时才能写入此位。* |
| 9 | PTS | 奇偶校验选择<br>0：偶校验<br>1：奇校验<br>*注意：在禁止LPUART（UE=0）时才能写入此位。* |
| 8 | PEIE | 奇偶校验错误中断使能<br>0：禁止<br>1：使能 |
| 7 | TXEIE | 发送寄存器空中断使能<br>0：禁止<br>1：使能 |
| 6 | TCIE | 传输完成中断使能<br>0：禁止<br>1：使能 |
| 5 | RXNEIE | 接收数据非空中断使能<br>0：禁止<br>1：使能 |
| 4 | 保留 | 写入无效 |
| 3 | TE | 发送器使能<br>0：禁止<br>1：使能 |
| 2 | RE | 接收器使能<br>0：禁止<br>1：使能 |
| 1 | UEWK | LPUART 在低功耗模式下的唤醒功能使能控制<br>0：禁止<br>1：使能 |

| | | |
|---|---|---|
| 0 | UE | LPUART 使能<br>此位清零后，LPUART 预分频器立即停止，当前工作丢弃；LPUART 配置保留；LPUARTx_TDR、LPUARTx_RDR 寄存器被清零；LPUARTx_ISR 寄存器的所有状态标志均被复位。<br>0：禁止<br>1：使能 |

### 29.6.2 LPUART 控制寄存器 2（LPUARTx_CR2）

偏移地址：0x04

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CMFD[7:0] | | | | | | | | Res. | | | | MSB FIRST | DATAIVC | TXIVC | RXIVC |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SWAP | Res. | STOPBIT | Res. | | | | | | | | | | | | |
| rw | | rw | | | | | | | | | | | | | |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:24 | CMFD[7:0] | 此位域用于字符匹配检测，接收到的整个字符（8bits）与CMFD[7:0]的值进行比较，如果一致则CMF置位。如果CMIE=1则触发中断。<br>*注意：仅在禁止接收（RE=0）或禁止LPUART（UE=0）时才能写入此位域。* |
| 23:20 | 保留 | 写入无效 |
| 19 | MSBFIRST | MSB优先<br>0：发送/接收时LSB优先<br>1：发送/接收MSB优先<br>*注意：在禁止LPUART（UE=0）时才能写入此位。* |
| 18 | DATAIVC | 数据极性反相<br>0：按正向逻辑处理发送/接收的数据（1=H，0=L）<br>1：按反向逻辑处理发送/接收数的数据（1=L，0=H）（奇偶校验位也取反。）<br>*注意：在禁止LPUART（UE=0）时才能写入此位。* |

| | | |
|---|---|---|
| 17 | TXIVC | TX引脚电平反相<br>0：TX引脚使用标准逻辑电平（$V_{DD}$ = 1，GND = 0）<br>1：TX引脚信号反向（$V_{DD}$ = 0，GND=1）<br>*注意：在禁止LPUART（UE=0）时才能写入此位。* |
| 16 | RXIVC | RX引脚电平反相<br>0：RX引脚信号使用标准逻辑电平（$V_{DD}$ = 1，GND = 0）<br>1：对RX引脚信号反相（$V_{DD}$ = 0，GND =1）<br>*注意：在禁止LPUART（UE=0）时才能写入此位。* |
| 15 | SWAP | 交换TX/RX引脚<br>0：按标准引脚排列使用TX/RX引脚<br>1：TX和RX引脚交换<br>*注意：在禁止LPUART（UE=0）时才能写入此位。* |
| 14 | 保留 | 写入无效 |
| 13 | STOPBIT | 停止位位数<br>0：1位<br>1：2位<br>*注意：在禁止LPUART（UE=0）时才能写入此位。* |
| 12:0 | 保留 | 写入无效 |

### 29.6.3 LPUART 控制寄存器 3（LPUARTx_CR3）

偏移地址：0x08

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | DDRE | Res. | | CTSIE | CTSE | RTSE | DMAT | DMAR | Res. | | HDEN | Res. | | EIE |
| | | rw | | | rw | rw | rw | rw | rw | | | rw | | | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|

| | | |
|---|---|---|
| 31:23 | 保留 | 写入无效 |
| 22 | LPWKIE | 低功耗模式事件唤醒信号中断使能<br>0：禁止<br>1：使能<br>*注意：LPWKIE需在进入低功耗模式前置位。* |
| 21:14 | 保留 | 写入无效 |
| 13 | DDRE | 接收出错时禁止DMA<br>0：不禁止DMA<br>相应错误标志置位，但RXNE保持为0，所以不产生DMA请求。<br>1：禁止DMA<br>相应的错误标志以及RXNE均置位。不产生DMA请求，直到错误标志清零。<br>*注意：接收错误包括：奇偶校验错误和帧错误。*<br>*在禁止LPUART（UE=0）时才能写入此位。* |
| 12:11 | 保留 | 写入无效 |
| 10 | CTSIE | CTS中断使能<br>0：禁止<br>1：使能 |
| 9 | CTSE | CTS使能<br>0：禁止<br>1：使能<br>*注意：在禁止LPUART（UE=0）时才能写入此位。* |
| 8 | RTSE | RTS使能<br>0：禁止<br>1：使能<br>*注意：在禁止LPUART（UE=0）时才能写入此位。* |
| 7 | DMAT | 使能DMA发送请求<br>0：禁止<br>1：使能 |

| | | |
|---|---|---|
| 6 | DMAR | 使能DMA接收请求<br>0：禁止<br>1：使能 |
| 5:4 | 保留 | 写入无效 |
| 3 | HDEN | 使能单线半双工<br>0：禁止<br>1：使能<br>*注意：在禁止LPUART（UE=0）时才能写入此位。* |
| 2:1 | 保留 | 写入无效 |
| 0 | EIE | 错误中断使能<br>如果发生帧错误或上溢错误时（LPUARTx_ISR寄存器中FE＝1或ORE=1）时，此位置位会触发中断。<br>0：禁止<br>1：使能 |

### 29.6.4 LPUART 波特率分频寄存器（LPUARTx_BRR）

偏移地址：0x0C

复位值：0x0000 0000

*注意：　在禁止 LPUART（UE=0）时才能修改此寄存器。*

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:20 | 保留 | 写入无效 |
| 19:0 | BRR[19:0] | 波特率分频值<br>详情参见*波特率生成*。 |

### 29.6.5 LPUART 中断和状态寄存器（LPUARTx_ISR）

偏移地址：0x1C

复位值：0x0000 00C0

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | REACK. | TEACK | LPWK | Res. | | CMF | BUSY |
| | | | | | | | | | r | r | r | | | r | r |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | CTS | CTSIF | Res. | TXE | TC | RXNE | Res. | ORE | Res. | FE | PE |
| | | | | | r | r | | r | r | r | | r | | r | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:23 | 保留 | 写入无效 |
| 22 | REACK | 接收使能确认标志<br>0：接收器未准备好<br>1：接收器准备完成，可以接收数据 |
| 21 | TEACK | 发送使能确认标志:<br>TE 清零时，发送器禁止完成后此位清零；当发送器已准备好发送时，此位置位。<br>0：发送器未准备好<br>1：发送器准备完成 |
| 20 | LPWK | 低功耗模式事件唤醒标志<br>当检测到起始位时，此位置位。<br>向 LPUARTx_ICR 寄存器中的 LPWKCF 位写 1，此位清零。<br>0：未检测到事件<br>1：检测到事件 |
| 19:18 | 保留 | 写入无效 |
| 17 | CMF | 字符匹配标志<br>接收到由CMFD[7:0]定义的字符时此位由硬件置位。<br>向LPUARTx_ICR寄存器中的CMCF位写1，此位清零。<br>0：未检测到字符匹配<br>1：检测到字符匹配 |

| | | |
|---|---|---|
| 16 | BUSY | 忙标志<br>当 RX 线路有接收的数据时，从检测到起始位开始，此位置位。<br>在接收结束时，无论是否成功，此位均清零。<br>0：LPUART 处于空闲状态<br>1：正在接收数据 |
| 15:11 | 保留 | 写入无效 |
| 10 | CTS | CTS 标志<br>0：CTS 引脚为高电平<br>1：CTS 引脚为低电平 |
| 9 | CTSIF | CTS 中断标志<br>CTSE 位置位且 CTS 输入出现电平翻转时，此位置位。<br>向 LPUARTx_ICR 寄存器中的 CTSCF 位写 1，此位清零。<br>0：CTS 引脚电平未发生变化<br>1：CTS引脚电平发生变化 |
| 8 | 保留 | 写入无效 |
| 7 | TXE | 发送数据寄存器为空<br>当 LPUARTx_TDR 寄存器的内容已发送时，TXE 置位。<br>写入 LPUARTx_TDR 寄存器时，此位清零。<br>0：发送数据寄存器 LPUARTx_TDR 非空<br>1：发送数据寄存器 LPUARTx_TDR 为空 |
| 6 | TC | 发送完成标志<br>此标志位置位，表示写入到LPUARTx_TDR中的最后一个数据已完成发送，且TXE置位，则此位置位。<br>向 LPUARTx_ICR 寄存器中的 TCCF 位写 1，或写入 LPUARTx_TDR寄存器时，此位清零。<br>0：发送未完成<br>1：发送已完成 |
| 5 | RXNE | 接收数据寄存器非空<br>当接收到数据时，RXNE 位置位。<br>读取 LPUARTx_RDR 寄存器时，此位清零。 |

0：未接收到数据
1：已接收到数据

| | | |
|---|---|---|
| 4 | 保留 | 写入无效 |
| 3 | ORE | 上溢错误标志<br>当 RXNE 标志为 1 时，或上一个 DMA 读请求未处理时，又接收到新的字符，就会产生上溢错误，则此位置位。<br>向 LPUARTx_ICR 寄存器中的 ORECF 位写 1，此位清零。<br>0：无上溢错误<br>1：检测到上溢错误<br>*注意：当此位置位时，RDR 寄存器的内容不会丢失，但移位寄存器会被覆盖。* |
| 2 | 保留 | 写入无效 |
| 1 | FE | 帧错误标志<br>接收数据时，如果在预期时间内未能识别出停止位，则检测为帧错误，此位置位。<br>向 LPUARTx_ICR 寄存器中的 FECF 位写 1，此位清零。<br>0：未检测到帧错误<br>1：检测到帧错误 |
| 0 | PE | 奇偶校验错误标志<br>当在接收时发生奇偶校验错误时，此位置位。<br>向 LPUARTx_ICR 寄存器中的 PECF 位写 1，此位清零。<br>0：无奇偶校验错误<br>1：奇偶校验错误 |

### 29.6.6 LPUART 中断标志清零寄存器（LPUARTx_ICR）

偏移地址：0x20

复位值：0x0000 0000

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | LPWKCF | Res. | | CMCF | Res. |
| | | | | | | | | | | | w | | | w | |

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

| Res. | | | | | | CTSCF | Res. | | TCCF | Res. | | ORECF | Res. | FECF | PECF |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | w | | | w | | | w | | w | w |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:21 | 保留 | 写入无效 |
| 20 | LPWKCF | 低功耗模式事件唤醒标志清零<br>向此位写1，LPUARTx_ISR寄存器中的LPWK标志将清零。 |
| 19:18 | 保留 | 写入无效 |
| 17 | CMCF | 字符匹配标志清零<br>向此位写1，LPUARTx_ISR寄存器中的CMF标志将清零。 |
| 16:10 | 保留 | 写入无效 |
| 9 | CTSCF | CTS标志清零<br>向此位写1，LPUARTx_ISR寄存器中的CTSIF标志将清零。 |
| 8:7 | 保留 | 写入无效 |
| 6 | TCCF | 发送完成标志清零<br>向此位写1，LPUARTx_ISR寄存器中的TC标志将清零。 |
| 5:4 | 保留 | 写入无效 |
| 3 | ORECF | 上溢错误标志清零<br>向此位写1，LPUARTx_ISR寄存器中的ORE标志将清零。 |
| 2 | 保留 | 写入无效 |
| 1 | FECF | 帧错误标志清零<br>向此位写1，LPUARTx_ISR寄存器中的FE标志将清零。 |
| 0 | PECF | 奇偶校验错误标志清零<br>向此位写1，LPUARTx_ISR寄存器中的PE标志将清零。 |

### 29.6.7 LPUART 接收数据寄存器（LPUARTx_RDR）

偏移地址：0x24

复位值：0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:9 | 保留 | 写入无效 |
| 8:0 | RDR[8:0] | 接收数据寄存器<br>在使能奇偶校验的时，接收数据的MSB为奇偶校验位。 |

### 29.6.8 LPUART 发送数据寄存器（LPUARTx_TDR）

偏移地址：0x28

复位值：0x0000 0000

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:9 | 保留 | 写入无效 |
| 8:0 | TDR[8:0] | 发送数据寄存器<br>在使能奇偶校验时，发送数据中写入的 MSB 自动填充为奇偶校验位，此时写入 MSB 的值无效。 |

### 29.6.9 LPUART 预分频器寄存器（LPUARTx_PRESC）

偏移地址：0x2C

复位值：0x0000 0000

*注意：　在禁止 LPUART（UE=0）时才能修改此寄存器。*

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:4 | 保留 | 写入无效 |
| 3:0 | PRESCALER[3:0] | 时钟预分频器<br>LPUARTx_KCLK 时钟经过此寄存器的分频系数分频后得到 LPUARTx_KCLK_PRES。<br>0000：1 分频<br>0001：2 分频<br>0010：4 分频<br>0011：6 分频<br>0100：8 分频<br>0101：10 分频<br>0110：12 分频<br>0111：16 分频<br>1000：32 分频<br>1001：64 分频<br>1010：128 分频<br>1011：256 分频<br>其他：保留（写入保留值时，256 分频） |

# 30 串行外设接口（SPI）

## 30.1 简介

串行外设接口协议支持与外部设备进行同步全双工串行通信。

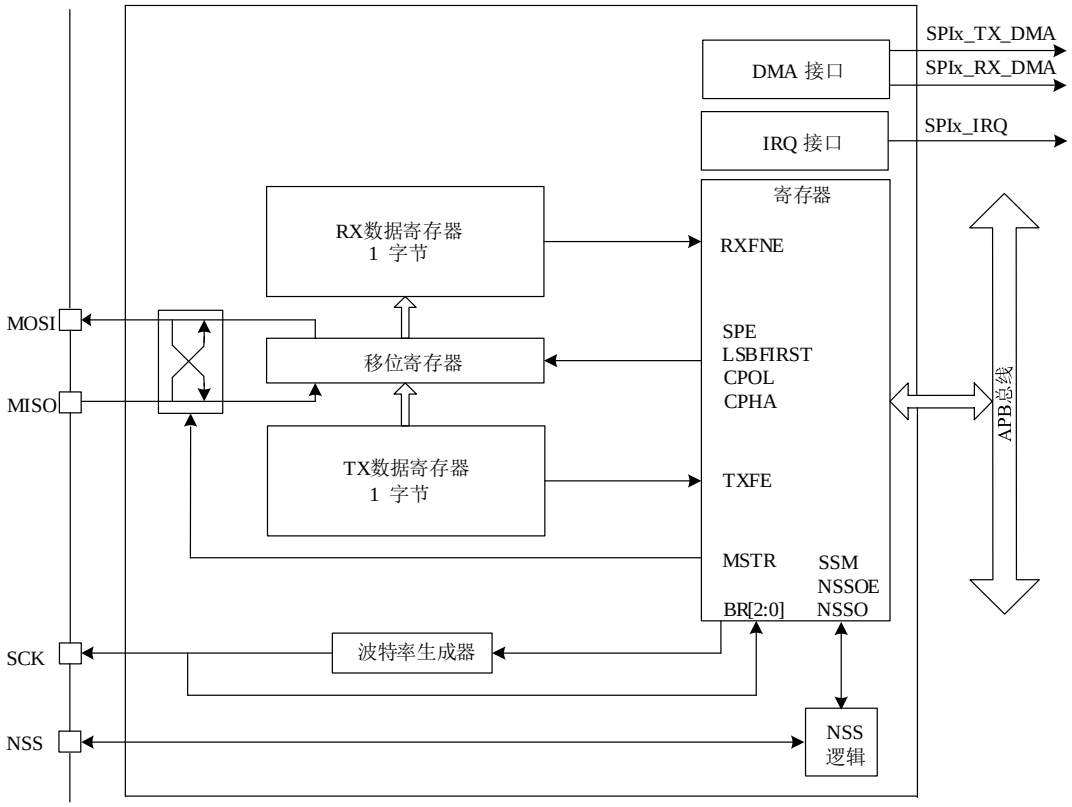
## 30.2 SPI 主要特性

- 支持 NSS、SCK、MISO 和 MOSI 标准四线的同步全双工通信
- 可配置为主机模式或从机模式
- 多主机模式功能
- 从模式支持软件片选
- 主机模式支持 8 种通信速率，可达 PCLK/2
- 可编程时钟极性和相位
- 传输数据大小固定为 8bit
- 可编程数据传输顺序：MSB 或 LSB
- 支持 SPI Motorola 协议
- 可触发中断的错误事件：主机模式冲突、上溢
- 支持 DMA 通信

## 30.3 SPI 功能说明

### 30.3.1 SPI 框图

图 30-1 SPIx 框图（x = 1，2）




### 30.3.2 SPI 引脚

表 30-1 SPI 引脚

| 引脚名称 | 信号类型 | 说明 |
|---|---|---|
| MISO | 输入/输出 | 主入/从出数据 |
| MOSI | 输入/输出 | 主出/从入数据 |
| SCK | 输入/输出 | 串行时钟 |
| NSS | 输入/输出 | 从机选择 |

### 30.3.3 单主机通信

SPI 是同步全双工通信接口，主从设备通过 MOSI 和 MISO 两条数据线进行数据传输。单主单从和单主多从连接关系见下图。

图 30-2 单主单从应用

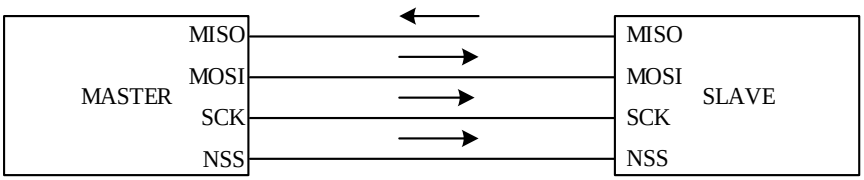
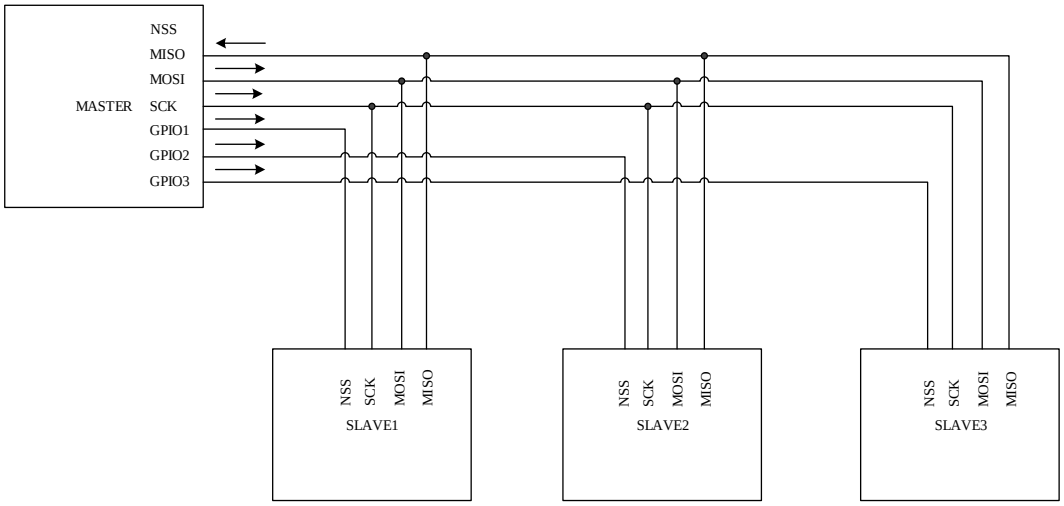



图 30-3 单主多从应用




当 SPIx_CR1 寄存器的 MSTR 置位时，SPI 接口工作于主机模式。

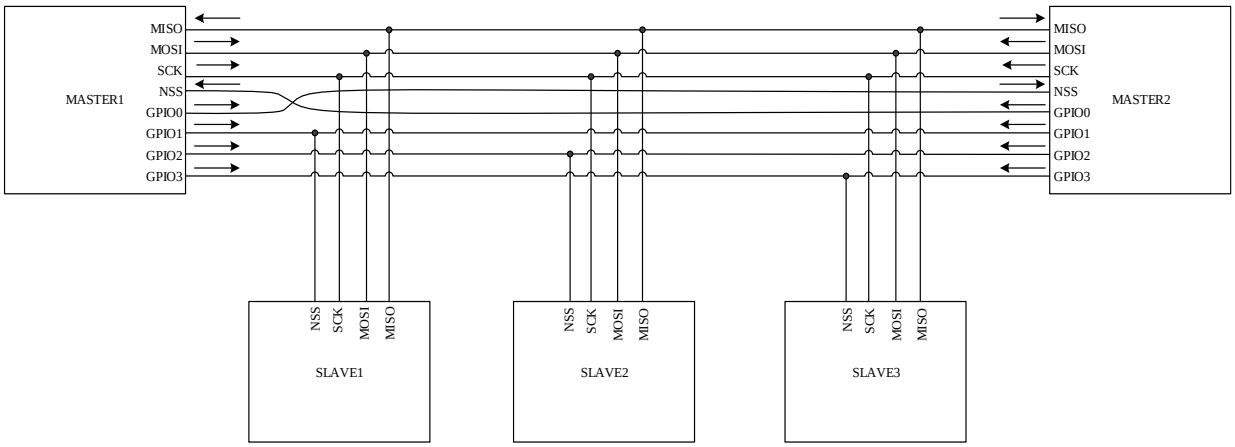
SCK 由主机提供，其波特率由 SPIx_CR1 寄存器的 BR[2:0]控制，范围为 $f_{PCLK}/2$~$f_{PCLK}/128$。

NSS 使用参见：*NSS 管理*。

### 30.3.4 多主机通信

SPI 总线包含两个主机时，用户可使用主机冲突功能来检测主机间是否存在冲突，从而实现多主机通信。多主机连接关系见下图。

图 30-4　多主多从应用




当总线上的主机设备无数据传输时，均保持为禁止状态（SPE 为 0），GPIO1~3 为输入状态。主从设备进行数据传输时，按照如下步骤操作：

1) 当 MASTER1 要接管对总线的控制，将自身切换到主机模式。如果此时其 NSS 输入为低，则总线已被 MASTER2 占用，MASTER1 将会生成主机模式冲突；
2) MASTER1 通过 GPIO0 向 MASTER2 的 NSS 施加低电平，通知其总线已占用；
3) MASTER1 通过 GPIO1 输出低电平选择与 SLAVE1 进行通信；
4) 传输完成后，MASTER1 由主机模式切换至禁止状态；
5) MASTER1 通过 GPIO0 向 MASTER2 的 NSS 施加高电平，通知其总线已空闲；
6) MASTER1 的 GPIO1 切换至输入状态。

如果两个主机同时发出总线占用的控制请求，则会出现总线冲突（参见：*主机模式冲突*）。发生冲突后，用户需通过应用程序进行冲突仲裁处理，例如，通过延迟尝试请求占用总线。

除 NSS 配置不同外，多主模式配置与单主机配置相同。NSS 配置差异参见：*NSS 管理*。

### 30.3.5 从机通信

当 SPIx_CR1 寄存器的 MSTR 清零时，SPI 接口工作于从机模式。从机模式下，数据收发流程受控于主机时钟，从机应在主机时钟的第 1 个时钟沿到来之前将发送数据写入 SPIx_DR 寄存器，以保证数据正常传输。从机应用连接关系参见*图：单主单从应用*。

### 30.3.6 NSS 管理

在从机模式下（MSTR=0），NSS 引脚作为输入，NSSOE 被硬件清零，NSSO 被硬件置位。可通过 SSM 位配置硬件片选或软件片选：

- 硬件片选（SSM=0）：NSS 引脚被片选后才能与主机进行通信。
- 软件片选（SSM=1）：忽略 NSS 引脚的输入状态，从机始终保持被片选的状态。

在主机模式下（MSTR=1），NSS 引脚可以通过 NSSOE 选择用作输入或者输出，SSM 被硬件清零：

- 用作输出时（NSSOE=1），应用于单主机模式，可以通过 NSS 驱动单个从机的片选信号，也可用若干 GPIO 驱动多个从机的片选信号。
  - 若选择 NSS 作为片选引脚，通过 NSSO 位控制片选信号的电平状态。
  - 若选择 GPIO 作为片选引脚，NSSO 位清零后通过 GPIO 控制片选信号的电平状态。
- 用作输入时（NSSOE=0），应用于多主机通信模式以检测 SPI 总线是否被占用，NSSO 位保持清零。
  - NSS 引脚输入为高，总线未被其他主机占用，通过 GPIO 控制片选信号的电平状态。
  - NSS 引脚输入为低，总线被其他主机占用，产生主模式冲突错误。

NSS 的配置应用见下表：

表 30-2 NSS 配置

| 主从模式 | NSS 引脚 | NSSOE | NSSO | SSM | 说明 |
|---|---|---|---|---|---|
| 从机 | 输入 | 0 | 1 | 0 | 从机 NSS 为输入，用于确定从机的片选是否有效。 |
| | | | | 1 | 从机始终处于被选中状态，忽略 NSS 引脚输入。 |
| 主机 | 输出 | 1 | 0/1 | 0 | NSS 为输出信号，NSS 输出电平由 NSSO 控制。 |
| | 输入 | 0 | 0 | 0 | NSS 输入高电平时，NSSO 保持清零，通过 GPIO 控制片选。 |

### 30.3.7 通信模式

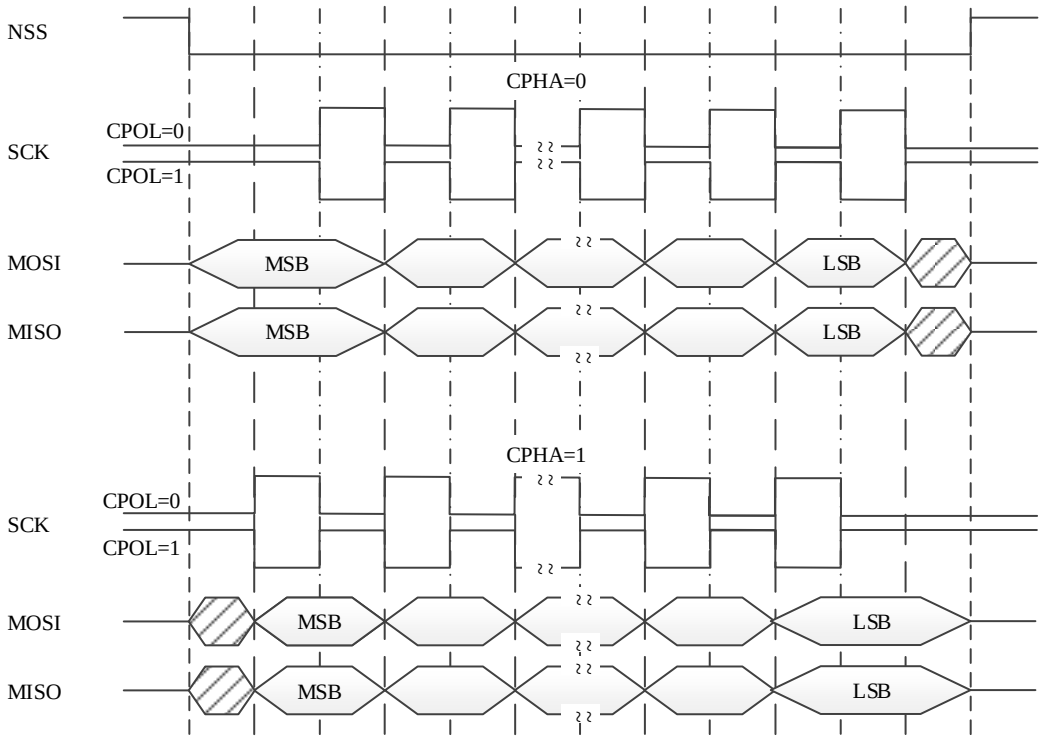
SPI 通信过程中，将同时执行接收和发送操作。数据线上的信息的移位和采样与串行时钟同步。四种通信格式取决于时钟相位和时钟极性，具体见下表。

表 30-3 通信模式配置

| SPI 模式 | CPOL | CPHA | 说明 |
|---|---|---|---|
| 模式 0 | 0 | 0 | 时钟空闲为低电平，在时钟第一个边沿进行数据采样，即数据在时钟的上升沿采样，下降沿变化。 |
| 模式 1 | 0 | 1 | 时钟空闲为低电平，在时钟第二个边沿进行数据采样，即数据在时钟的下降沿采样，上升沿变化。 |
| 模式 2 | 1 | 0 | 时钟空闲为高电平，在时钟第一个边沿进行数据采样，即数据在时钟的下降沿采样，上升沿变化。 |
| 模式 3 | 1 | 1 | 时钟空闲为高电平，在时钟第二个边沿进行数据采样，即数据在时钟的上升沿采样，下降沿变化。 |

四种通信模式通过 SPIx_CR1 寄存器的时钟极性 CPOL 位和时钟相位 CPHA 位配置。

图 30-5 通信模式时序




SPI 内部移位寄存器支持可配置的数据传输顺序；通过 SPIx_CR1 寄存器的 LSBFIRST 位选择，LSBFIRST 为 0，MSB 优先；LSBFIRST 为 1，LSB 优先。

### 30.3.8 状态标志

提供发送数据寄存器空状态、接收数据寄存器非空状态和总线忙状态，通过这些状态管理数据的发送和接收。

- 发送数据寄存器空标志（TXFE）

  当发送数据寄存器中没有数据时，TXFE 标志置位。如果发送数据寄存器非空时，TXFE 标志由硬件自动清零。

- 接收数据寄存器非空标志（RXFNE）

  当接收数据寄存器非空时，RXFNE 标志置位。只有将接收数据寄存器中的数据读出，RXFNE 标志由硬件自动清零。

- 忙标志（BUSY）

  BUSY 标志由硬件置位和清零。

  当 BUSY 置位时，表示 SPI 总线上正在进行数据传输，BUSY 标志检测一帧数据是否传输完成，用以正确禁止 SPI，避免破坏数据的完整性。

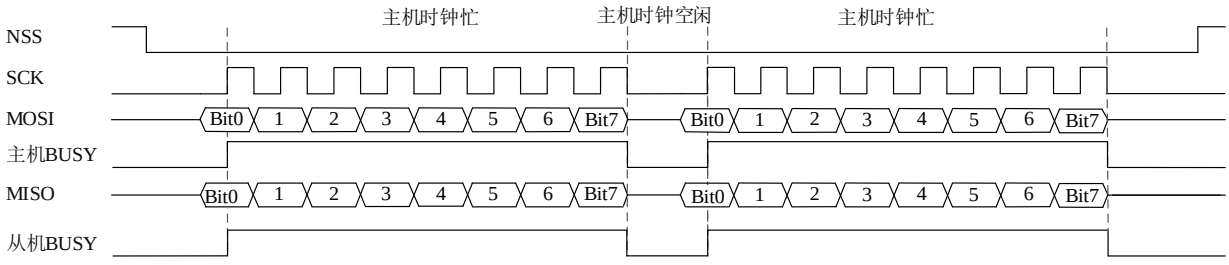
  在以下任意一种条件下，BUSY 标志将清零：

  – 在主机/从机应用下，上一个数据帧最后的翻转时钟边沿与下一个数据帧起始的翻转时钟边沿之间；

  – 在主机应用下，发送数据寄存器已为空且最后一个数据发送完成时；

  – 在多主机应用下，检测到主机模式冲突时。

图 30-6 主从模式 BUSY 状态




### 30.3.9 错误标志

以下错误标志中，如有一个置位且错误中断使能 ERRIE 位置位，则将生成 SPI 错误中断。

- 上溢标志

  当移位寄存器已接收到数据，但接收数据寄存器没有空间保存此数据，此时 OVR 标志置位。当出现上溢的情况时，新接收的数据不会覆盖接收数据寄存器中之前的值，新接收的数据将被丢弃。将 SPIx_ICR 寄存器中 OVRCF 位置位，OVR 标志将同步清零。

- 主机模式冲突标志

  当 SPI 主机模式且 NSS 配置为输入时，NSS 信号被拉低，将发生主机模式冲突，此时 MMF 标志置位。多应用于多主机通信系统。

  发生主机模式冲突时，硬件将自动执行如下动作：

  – SPE 位清零，禁止 SPI 接口；

  – MSTR 位清零，进入从机模式。

  将 SPIx_ICR 寄存器中 MMFCF 位置位，MMF 标志将同步清零。

  为避免包含多个芯片的系统中发生主机冲突，必须在 MMF 位清零期间将输出给其他从机的 NSS 引脚拉高。在 MMF 清零后，可以将 SPE 和 MSTR 位重新置位。在从机模式中，MMF 位不会置位，但由主机模式冲突引起的自动进入从机模式的情况除外。

### 30.3.10 初始化 SPI

除 BR[2:0]和 NSSOE 外，主机模式和从机模式的配置步骤相同。对于具体的配置，参看相应功能介绍的内容。SPI 初始化步骤如下：

1) 将对应的 GPIO 配置为 NSS、SCK、MOSI 和 MISO 功能，参考：*通用输入/输出接口（GPIO）*。主机模式下，不使用 SPI 控制 NSS 情况下，可将 NSS 配置为 GPIO 功能。
2) 将 SPIx_CR1 寄存器的 SPE 位清零。
3) 写 SPIx_CR1 寄存器：

   – 通过 BR[2:0]位配置串行时钟波特率。从机模式无需配置此项。

   – 配置 CPOL 位和 CPHA 位，选择 SPI 模式。

   – 配置 LSBFIRST 位，选择数据位传输顺序。

   – 配置 MSTR 位，选择主从模式。

- 配置 NSSOE 位，选择 NSS 状态。从机模式无需配置此项。

4) 将 SPIx_CR1 寄存器的 SPE 位置位。

### 30.3.11 数据发送和接收

**查询和中断方式**

SPIx_DR 的写访问可通过 TXFE 状态管理。当 TXFE 置位时，可以对 SPIx_DR 执行写访问。

SPIx_DR 的读访问可通过 RXFNE 状态管理。当 RXFNE 置位时，可以对 SPIx_DR 执行读访问。

TXFE 和 RXFNE 事件可以通过查询或者中断的方式进行判断和处理。当使用中断方式时，将 SPx_CR1 的相关事件中断使能置位，发生事件时产生中断，参见：*SPI 中断*。

为确保 SPI 最后一个数据通信的完整性，正确步骤如下：

1) 等待发送数据寄存器为空 TXFE 为 1。
2) 等待 BUSY 为 0，最后一帧数据已传输完成。
3) 读出接收数据直至 RXFNE 为 0。

**DMA 方式**

SPIx_CR1 寄存器中 DMA 方式发送使能（DMAT 为 1）时，每次 TXFE 为 1，SPI 会发出 DMA 请求。

SPIx_CR1 寄存器中 DMA 方式接收使能（DMAR 为 1）时，每次 RXFNE 为 1，SPI 会发出 DMA 请求。

当 SPI 仅用于发送数据时，可以只使能 SPI 发送的 DMA 通道。在这种情况下，SPI 的 OVR 标志会置位。当 SPI 仅用于接收数据时，可以只使能 SPI 接收的 DMA 通道。

在发送模式下，DMA 传输完成标志置位后，可以对 TXFE 和 BUSY 标志进行查询，以确保 SPI 通信已完成。在关闭 SPI 前必须执行此步骤，以避免损坏最后一次发送。

通过 DMA 发送和接收数据时，必须按顺序执行以下步骤：

1) 配置 DMA 的数据发送通道和数据接收通道，参见：*直接存储器访问控制*

*器（DMA）*；

2) 使能 SPI 接收的 DMA 通道；
3) SPI_CR1 寄存器中 DMAR 位置位，启动 SPI DMA 方式接收；
4) 使能 SPI 发送的 DMA 通道；
5) SPI_CR1 寄存器中 DMAT 位置位，启动 SPI DMA 方式发送。

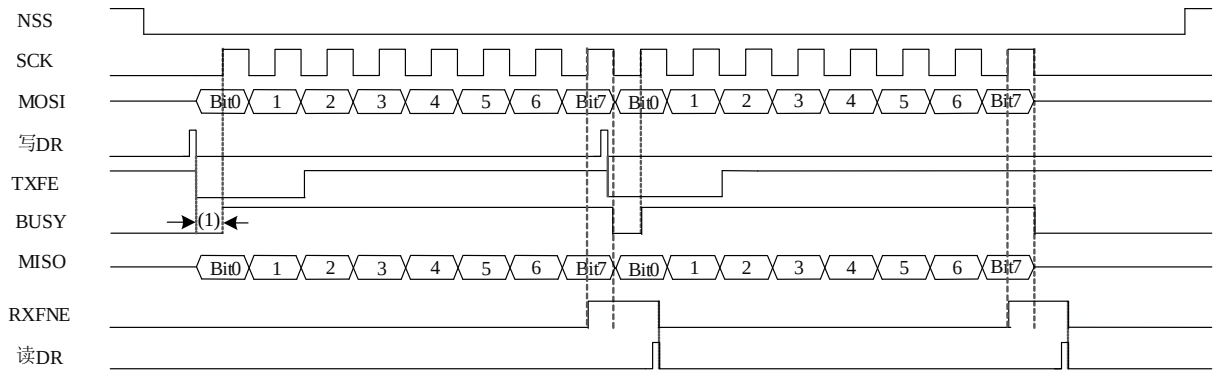
要关闭 DMA 发送和接收数据，必须按顺序执行以下步骤：

1) 等待 DMA 发送和接收通道的传输完成标志置位；
2) 等待发送数据寄存器为空 TXFE 标志置位；
3) 等待 BUSY 为 0，最后一帧数据已传输完成；
4) 等待接收寄存器为空 RXFNE 标志清零；
5) 将 SPI_CR1 寄存器中 DMAR 和 DMAT 位清零；
6) 禁止 DMA 的数据发送通道和数据接收通道，参见：*直接存储器访问控制器（DMA）*。

时钟信号由主机设备提供，直至发送数据寄存器和移位寄存器为空，之后时钟停止，直至主机再次发送数据。通信过程参见下图。

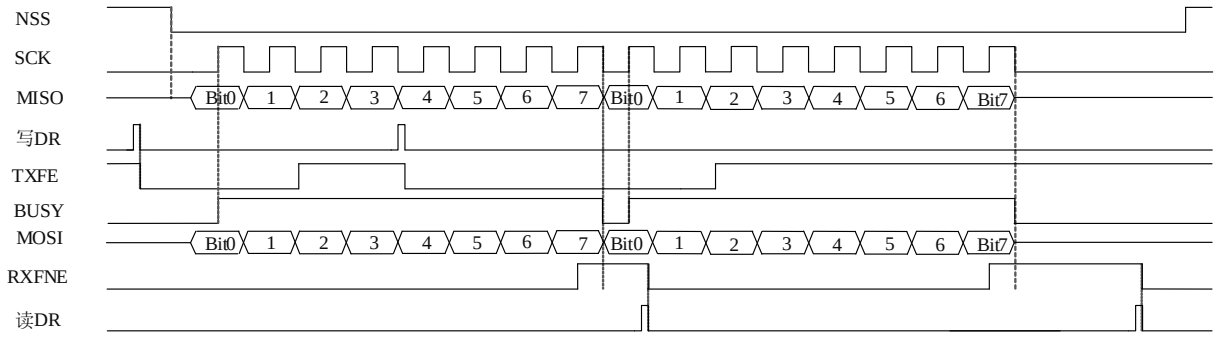
下图为 SPI 主机模式，传输 2 字节数据的通信时序，通信参数配置为：时钟空闲低电平，在第一个边沿采样。

图 30-7 主模式通信时序




下图为 SPI 从机模式，传输 2 字节数据的通信时序，通信参数配置为：时钟空闲低电平，在第一个边沿采样。

图 30-8 从模式通信时序




### 30.3.12 禁止 SPI

当禁止 SPI 时，需确保 SPI 最后一次通信数据收发完成，参见：*数据发送和接收*，否则会损坏正在进行的交互。将 SPIx_CR1 寄存器的 SPE 位清零后禁止 SPI。

## 30.4 SPI 中断

在 SPI 通信过程中，SPI 中断可由不同事件生成。不同事件可配置使能或禁止生成中断。SPI 中断详情参见下表。

表 30-4 中断请求

| 中断事件 | 事件标志 | 中断使能位 | 事件标志/中断清除方法 |
|---|---|---|---|
| 发送数据寄存器空 | TXFE | TXFEIE | 写 SPIx_DR 寄存器，使发送数据寄存器不为空 |
| 接收数据寄存器非空 | RXFNE | RXFNEIE | 读 SPIx_DR 寄存器，使接收数据寄存器为空 |
| 上溢 | OVR | ERRIE | 向OVRCF位写1 |
| 主机模式冲突 | MMF | | 向 MMFCF 位写 1 |

## 30.5 SPI 寄存器

SPI 寄存器支持 32 位访问。

表 30-5 SPIx 基地址

| 外设 | 基地址 |
|---|---|
| SPI1 | 0x4001 3000 |
| SPI2 | 0x4000 3800 |

### 30.5.1 SPI 控制寄存器 1（SPIx_CR1）

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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | TXFEIE | Res. | | | | RXFNEIE | Res. | ERRIE | DMAT | DMAR |
| | | | | | | rw | | | | | rw | | rw | rw | rw |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | SSM | NSSOE | Res. | LSB FIRST | Res. | MSTR | CPOL | CPHA | BR[2:0] | | | SPE |
| | | | | rw | rw | | rw | | rw | rw | rw | rw | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:26 | 保留 | 写入无效 |
| 25 | TXFEIE | 发送数据寄存器为空中断使能<br>0：禁止<br>1：使能 |
| 24:21 | 保留 | 写入无效 |
| 20 | RXFNEIE | 接收数据寄存器非空中断使能<br>0：禁止<br>1：使能 |
| 19 | 保留 | 写入无效 |
| 18 | ERRIE | 错误中断使能<br>如果使能错误中断，MMF、OVR 任一事件将产生中断。 |

| | | |
|---|---|---|
| | | 0：禁止<br>1：使能 |
| 17 | DMAT | DMA 方式发送使能<br>0：禁止<br>1：使能 |
| 16 | DMAR | DMA 方式接收使能<br>0：禁止<br>1：使能 |
| 15:12 | 保留 | 写入无效 |
| 11 | SSM | 从机模式 NSS 管理<br>0：从机选择由 NSS 引脚输入决定是否被片选<br>1：从机选择始终处于被选中状态，并忽略 NSS 引脚输入<br>*注意：此位仅在从机模式下设置有效，主机模式下硬件清零。* |
| 10 | NSSOE | NSS 输出使能<br>0：NSS 为输入模式<br>1：NSS 为输出模式<br>*注意：此位仅在主机模式下设置有效，从机模式下硬件清零。* |
| 9 | 保留 | 写入无效 |
| 8 | LSBFIRST | 数据传输顺序<br>0：发送/接收数据时 MSB 在前<br>1：发送/接收数据时 LSB 在前<br>*注意：仅在禁止SPI（SPE为0）时才可对此位执行写操作。* |
| 7 | 保留 | 写入无效 |
| 6 | MSTR | SPI 主从模式<br>0：从机模式<br>1：主机模式<br>*注意：仅在禁止SPI（SPE为0）时才可对此位执行写操作。* |
| 5 | CPOL | 时钟极性 |

| | | |
|---|---|---|
| | | 0：SCK 空闲状态为低电平<br>1：SCK 空闲状态为高电平<br>*注意：仅在禁止SPI（SPE为0）时才可对此位执行写操作。* |
| 4 | CPHA | 时钟相位配置<br>0：第一边沿采样数据<br>1：第二边沿采样数据<br>*注意：仅在禁止SPI（SPE为0）时才可对此位执行写操作。* |
| 3:1 | BR[2:0] | 波特率控制<br>000：$f_{PCLK}/2$<br>001：$f_{PCLK}/3$<br>010：$f_{PCLK}/4$<br>011：$f_{PCLK}/8$<br>100：$f_{PCLK}/16$<br>101：$f_{PCLK}/32$<br>110：$f_{PCLK}/64$<br>111：$f_{PCLK}/128$<br>*注意：仅在禁止SPI（SPE为0）时才可对此位执行写操作。* |
| 0 | SPE | SPI 使能控制<br>0：禁止<br>1：使能 |

### 30.5.2 SPI 控制寄存器 2（SPIx_CR2）

偏移地址：0x04

复位值：0x0000 0001

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

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:1 | 保留 | 写入无效 |

| | | |
|---|---|---|
| 0 | NSSO | NSS 输出电平<br>该位结合 NSSOE 使用，NSSOE 为 1 时，NSS 输出有效。<br>0：NSS 输出低电平<br>1：NSS 输出高电平<br>*注意：此位仅在主机模式下设置有效，从机模式下硬件置位。* |

### 30.5.3 SPI 中断和状态寄存器（SPIx_ISR）

偏移地址：0x0C

复位值：0x0000 0100

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | TXFE | Res. | RXFNE | Res. | BUSY | Res. | OVR | Res. | MMF |
| | | | | | | | r | | r | | r | | r | | r |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:9 | 保留 | 写入无效 |
| 8 | TXFE | 发送数据寄存器空标志<br>发送数据寄存器为空时置位，否则由硬件清零。<br>0：发送数据寄存器非空<br>1：发送数据寄存器为空<br>*注意：当 SPE 为 0，该位自动置位。* |
| 7 | 保留 | 写入无效 |
| 6 | RXFNE | 接收数据寄存器非空标志<br>接收数据寄存器非空时置位，否则硬件清零。<br>0：接收数据寄存器为空，未接收到数据<br>1：接收数据寄存器非空，已接收到数据<br>*注意：当 SPE 为 0，该位自动清零。* |
| 5 | 保留 | 写入无效 |

| | | |
|---|---|---|
| 4 | BUSY | SPI 总线传输状态标志<br>由硬件置位和清零。<br>0：SPI 总线空闲<br>1：SPI 总线正在传输数据<br>*注意：当 SPE 为 0，该位自动清零。* |
| 3 | 保留 | 写入无效 |
| 2 | OVR | 上溢标志<br>接收数据寄存器已满而移位寄存器又接收到数据时，此位置位。将 SPIx_ICR 寄存器中的 OVRCF 位置位，此位清零。<br>0：未发生上溢<br>1：发生上溢<br>*注意：当 SPE 为 0，该位自动清零。* |
| 1 | 保留 | 写入无效 |
| 0 | MMF | 主机模式冲突标志<br>在主机模式下，NSS 输入电平为低，此位置位。将 SPIx_ICR 寄存器中的 MMFCF 位置位，此位清零。<br>0：未发生主机模式冲突<br>1：发生主机模式冲突<br>*注意：发生主机模式冲突时，MSTR 和 SPE 同时由硬件清零。此标志不清除，无法置位 MSTR 和 SPE。清除此标志时软件应检查 NSS 的输入为高电平。* |

### 30.5.4 SPI 中断标志清零寄存器（SPIx_ICR）

偏移地址：0x10

复位值：0x0000 0000

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | OVRCF | Res. | MMFCF |
| | | | | | | | | | | | | | w | | w |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:3 | 保留 | 写入无效 |
| 2 | OVRCF | 上溢标志清除<br>写 1 清除 SPIx_ISR 寄存器中的 OVR 标志。 |
| 1 | 保留 | 写入无效 |
| 0 | MMFCF | 主机模式冲突标志清除<br>写 1 清除 SPIx_ISR 寄存器中的 MMF 标志。 |

### 30.5.5 SPI 数据寄存器（SPIx_DR）

偏移地址：0x14

复位值：0x0000 0000

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

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:8 | 保留 | 写入无效 |
| 7:0 | DR[7:0] | 数据寄存器<br>对数据寄存器执行写操作时，将要发送数据写入发送数据寄存器；对数据寄存器执行读操作时，将返回接收数据寄存器中的已接收的数据。 |

# 31 调试接口（DBG）

## 31.1 简介

芯片采用 Cortex-M0+内核，具有硬件调试功能，支持复杂的调试操作。可以通过 SWD 调试接口为程序设置断点，使内核在获取特定指令（指令断点）或访问特定数据（数据断点）时停止。内核停止时，内核的内部状态和外设的状态都可以通过调试接口查询。当芯片连接到调试器后，可通过调试接口进行调试操作。

## 31.2 DBG 功能描述

Cortex-M0+内置的调试功能是 ARM CoreSight 设计套件的一部分，集成了 SW-DP（串行调试接口）、DWT（数据观察点触发）和 BPU（断点单元），具体如下：

- 调试支持串行线（SW）协议；
- BPU 提供 3 个断点寄存器；
- DWT 提供 1 个观察点寄存器组。

## 31.3 调试组件（DBG）

芯片调试组件为以下调试功能提供支持：

- 低功耗模式；
- 断点期间定时器和看门狗的时钟控制。

### 31.3.1 对低功耗模式的调试支持

支持 Sleep 模式和 Stop 模式的调试功能。

内核不允许在调试期间关闭 HCLK。由于调试期间需要使用它们进行调试连接，因此其必须保持激活状态。

必须设置一些调试寄存器，以便在低功耗模式进行调试：

- 在 Sleep 模式下，HCLK 仍有效，对调试功能没有任何限制。
- 调试 Stop 模式时，DBG_CR 寄存器的 DBG_STOP 位置位，系统时钟根据配置自动切换为 RCHSYS 给 HCLK 提供时钟。

### 31.3.2 对定时器和看门狗的调试支持

当芯片处于调试模式内核停止时，可以选择定时器或看门狗计数器是否继续计数，参见：*APB1 冻结寄存器（DBG_APB1_FZ）*、*APB2 冻结寄存器（DBG_APB2_FZ）*。

## 31.4 DBG 寄存器

DBG 寄存器支持 32 位访问。

表 31-1 DBG 基地址

| 外设 | 基地址 |
|---|---|
| DBG | 0x4001 5800 |

### 31.4.1 DBG 控制寄存器（DBG_CR）

偏移地址：0x00

复位值：0x0000 0000

*注意： 仅 POR/PDR 复位时，复位该寄存器。*

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | DBG_ STOP |
| | | | | | | | | | | | | | | | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:1 | 保留 | 写入无效 |
| 0 | DBG_STOP | Stop模式调试使能<br>0：禁止<br>1：使能 |

### 31.4.2 APB1 冻结寄存器（DBG_APB1_FZ）

偏移地址：0x04

复位值：0x0000 0000

*注意： 仅 POR/PDR 复位时，复位该寄存器。*

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | LPTIM1 _HOLD | Res. | | | | | | | | | | IWDG _HOLD | Res. | RTC_ HOLD |
| | | rw | | | | | | | | | | | rw | | rw |

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

| Res. | | | | | | | | | | | | | TIM4_HOLD | TIM3_HOLD | HALLTIM_HOLD |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | rw | rw | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:30 | 保留 | 写入无效 |
| 29 | LPTIM1_HOLD | 内核停止时，LPTIM1计数控制<br>0：正常计数<br>1：停止计数 |
| 28:19 | 保留 | 写入无效 |
| 18 | IWDG_HOLD | 内核停止时，IWDG计数控制<br>0：正常计数<br>1：停止计数 |
| 17 | 保留 | 写入无效 |
| 16 | RTC_HOLD | 内核停止时，RTC计数控制<br>0：正常计数<br>1：停止计数 |
| 15:3 | 保留 | 写入无效 |
| 2 | TIM4_HOLD | 内核停止时，TIM4计数控制<br>0：正常计数<br>1：停止计数 |
| 1 | TIM3_HOLD | 内核停止时，TIM3计数控制<br>0：正常计数<br>1：停止计数 |
| 0 | HALLTIM_HOLD | 内核停止时，HALLTIM计数控制<br>0：正常计数<br>1：停止计数 |

### 31.4.3 APB2 冻结寄存器（DBG_APB2_FZ）

偏移地址：0x08

复位值：0x0000 0000

*注意：　仅 POR/PDR 复位时，复位该寄存器。*

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Res. | | | | | | | | | | | | | | | TIM1_HOLD |
| | | | | | | | | | | | | | | | rw |

| 位/位域 | 名称 | 描述 |
|---|---|---|
| 31:1 | 保留 | 写入无效 |
| 0 | TIM1_HOLD | 内核停止时，TIM1计数控制<br>0：正常计数<br>1：停止计数 |

# 32 器件电子签名

器件电子签名可通过 SWD 或者 CPU 读取。它包含的芯片识别信息在出厂时编写，用户程序或者外部设备可以读取电子签名，用以自动匹配微处理器的特性。

## 32.1 产品唯一身份识别（UID）寄存器（96 位）

产品唯一身份识别码存放在 Flash 模块的 Engineer 区，其典型应用场景为：

- 用作序列号
- 对 Flash 编程前将 UID 和软件加密原语及协议结合使用，作为安全密钥的一部分以提高 Flash 代码的安全性
- 激活安全启动过程等

96 位的唯一设备标识符提供了一个对于任何器件和任何上下文都唯一的参考号码。用户不能改变这些位。

产品唯一身份识别码包含 Lot、Wafer 信息，以及芯片坐标信息等。

存储地址：0x1FFF 0234。

表 32-1 UID 规则

| 偏移地址 | 描述 | UID Bits | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | Lot Number | Lot Number ASCII 码 | | | | | | | |
| 1 | | | | | | | | | |
| 2 | | | | | | | | | |
| 3 | | | | | | | | | |
| 4 | | | | | | | | | |
| 5 | | | | | | | | | |
| 6 | Rev ID | 保留 ID | | | | | | | |
| 7 | Wafer Number | Wafer 编号，十进制编码 | | | | | | | |
| 8 | X,Y 坐标高位 | Y 坐标高位 | | | | X 坐标高位 | | | |
| 9 | X 坐标 | X 坐标，十六进制编码 | | | | | | | |
| 10 | Y 坐标 | Y 坐标，十六进制编码 | | | | | | | |
| 11 | Rev ID | 保留 ID | | | | | | | |

## 32.2 Flash 空间参数寄存器

存储地址：0x1FFF 0274，参见*表：产品信息*。

## 32.3 SRAM 空间参数寄存器

存储地址：0x1FFF 0278，参见*表：产品信息*。

# 33 版本历史

表 33-1 版本更改履历

| 日期 | 版本号 | 修改范围 |
|---|---|---|
| 2026-5-18 | V1.0 | 初版 |

# 34 联系方式

网址：www.hed.com.cn

地址：北京市昌平区北七家未来科技城南区中国电子网络安全和信息化产业基地 C 栋

邮编：102209

如果您在购买与使用过程中有任何意见或建议，请随时与我们联系。