STC micro™
宏 晶 科 技

深圳国芯人工智能有限公司

STC33F 系列单片机
讨论稿

STC MCU

技术支持网站：www.STCMCUDATA.com
资料更新日期：**2022/9/1**

目录

1 特性 14
1.1 主要特色 14
1.2 模拟模块 16
1.3 封装 17
2 DSP 与浮点运算单元 20
2.1 DSP 库 20
2.2 单精度浮点单元 23
3 开发环境的建立与 ISP 下载 24
3.1 添加型号和头文件到 Keil 27
3.2 使用 KEIl 进行烧录 29
3.3 使用 keil 进行仿真调试 35
4 嵌入式 Flash 接口 38
4.1 前言 38
4.2 主要特性 38
4.3 嵌入式 Flash 38
4.4 读接口 38
4.4.1 CPU 时钟频率与 Flash 读取时间之间的关系 38
4.4.2 缓存存储器加速器 41
4.5 擦除和编程操作 41
4.5.1 Flash 控制寄存器解锁 41
4.5.2 编程/ 擦除并行位数 41
4.5.3 擦除 42
4.5.4 编程 42
4.5.5 中断 43
4.6 选项字节 45
4.6.1 关于用户选项字节的说明 45
4.6.2 用户选项字节编程 46
4.6.3 读保护 (RDP) 46
4.6.4 写保护 47
4.7 Flash 接口寄存器 48
4.7.1 Flash 访问控制寄存器 (FLASH_ACR) 48
4.7.2 Flash 密钥寄存器 (FLASH_KEYR) 49
4.7.3 Flash 选项密钥寄存器 (FLASH_OPTKEYR) 50
4.7.4 Flash 状态寄存器 (FLASH_SR) 50
4.7.5 Flash 控制寄存器(FLASH_CR) 51
4.7.6 Flash 选项控制寄存器 (FLASH_OPTCR) 52
4.7.7 Flash 接口寄存器映射 54
5 Quad-SPI 接口 （QUADSPI） 55
5.1 前言 55
5.2 QUADSPI 主要特性 55

STC MCU

5.3 QUADSPI 功能说明 56
5.3.1 QUADSPI 框图 56
5.3.2 QUADSPI 命令序列 56
5.3.3 QUADSPI 信号接口协议模式 58
5.3.4 QUADSPI 间接模式 59
5.3.5 QUADSPI 状态标志轮询模式 60
5.3.6 QUADSPI 内存映射模式 60
5.3.7 QUADSPI Flash 配置 61
5.3.8 QUADSPI 延迟数据采样 61
5.3.9 QUADSPI 配置 61
5.3.10 QUADSPI 的用法 62
5.3.11 指令仅发送一次 63
5.3.12 QUADSPI 的繁忙位和中止功能 64
5.3.13 nCS 行为 64
5.4 QUADSPI 中断 65
5.5 QUADSPI 寄存器 66
5.5.1 QUADSPI 控制寄存器(QUADSPLCR) 66
5.5.2 QUADSPI 器件配置寄存器(QUADSPI_CCR) 68
5.5.3 QUADSPI 状态寄存器(QUADSPI_SR) 69
5.5.4 QUADSPI 标志清零寄存器(QUADSPI_FCR) 70
5.5.5 QUADSPI 数据长度寄存器(QUADSPI_DLR) 70
5.5.6 QUADSPI 通信配置寄存器(QUADSPI_CCR) 71
5.5.7 QUADSPI 地址寄存器(QUADSPI_AR) 73
5.5.8 QUADSPI 交替字节寄存器(QUADSPI_ABR) 73
5.5.9 QUADSPI 数据寄存器(QUADSPI_DR) 74
5.5.10 QUADSPI 轮询状态屏蔽寄存器(QUADSPI _PSMKR) 74
5.5.11 QUADSPI 轮询状态匹配寄存器(QUADSPI _PSMAR) 75
5.5.12 QUADSPI 轮询间隔寄存器(QUADSPI _PIR) 75
5.5.13 QUADSPI 低功耗超时寄存器(QUADSPIJ~PTR) 75
5.5.14 QUADSPI 寄存器映射 76
6 灵活的静态存储器控制器(FSMC) 78
6.1 FSMC 功能描述 78
6.2 框图 78
6.3 AHB 接口 79
6.3.1 支持的存储器和操作 79
6.4 外部设备地址映像 80
6.4.1 NOR 和 PSRAM 地址映像 81
6.5 NOR 闪存和 PSRAM 控制器 82
6.5.1 外部存储器接口信号 82
6.5.2 支持的存储器及其操作 83
6.5.3 时序规则 84
6.5.4 NOR 闪存和 PSRAM 控制器时序图 84
6.5.5 NOR 闪存和 PSRAM 控制器存储器 97
6.5.6 FSMC 寄存器地址映象 100

6.6 NAND Flash 控制器 101
6.6.1 外部存储器接口信号 101
6.6.2 NAND Flash 支持的存储器和事务 102
6.6.3 NAND 的时序图 103
6.6.4 每个 NAND Flash 存储区域均通过以下一组寄存器进行管理： 103
6.6.5 NAND Flash 预等待功能 105
6.6.6 错误修正代码计算 ECC(NAND Flash) 106
6.6.7 NAND Flash 控制寄存器 106
7 电源控制器 (PWR) 112
7.1 电源 112
7.1.1 独立 A/D 转换器电源和参考电压 112
7.1.2 调压器 112
7.2 电源监控器 113
7.2.1 上电复位 (POR)/ 掉电复位 (PDR) 113
7.2.2 欠压复位 (BOR) 113
7.2.3 可编程电压检测器 (PVD) 114
7.3 低功耗模式 115
7.3.1 降低系统时钟速度 116
7.3.2 外设时钟门控 116
7.3.3 睡眠模式 116
7.3.4 停止模式 117
7.3.5 待机模式 118
7.4 电源控制寄存器 120
7.4.1 PWR 电源控制寄存器(PWR_CR) 120
7.4.2 PWR 电源控制/ 状态寄存器 (PWR_CSR) 121
7.5 PWR 寄存器映射 122
8 复位和时钟控制 (RCC) 124
8.1 复位 124
8.1.1 系统复位 124
8.1.2 电源复位 124
8.1.3 备份域复位 125
8.2 时钟 125
8.2.1 HSE 时钟 126
8.2.2 HSI 时钟 127
8.2.3 LSE 时钟 127
8.2.4 LSI 时钟 127
8.2.5 系统时钟 (SYSCLK) 选择 128
8.2.6 时钟 128
8.2.7 看门狗时钟 128
8.2.8 时钟输出功能 128
8.2.9 基于 PWM5/PWM11 的内部/ 外部时钟测量 128
8.3 RCC 寄存器 130
8.3.1 RCC 时钟控制寄存器 (RCC_CR) 130
8.3.2 RCC PLL 配置寄存器 (RCC_PLLCFGR) 131

8.3.3 RCC 时钟配置寄存器 (RCC_CFGR) 131
8.3.4 RCC 钟中断寄存器 (RCC_CIR) 133
8.3.5 RCC AHB1 外设复位寄存器 (RCC_AHB1RSTR) 135
8.3.6 RCC AHB2 外设复位寄存器 (RCC_AHB2RSTR) 136
8.3.7 RCC AHB3 外设复位寄存器 (RCC_AHB3RSTR) 137
8.3.8 RCC APB1 外设复位寄存器 (RCC_APB1RSTR) 137
8.3.9 RCC APB2 外设复位寄存器外设复位寄存器 (RCC_APB2RSTR) 139
8.3.10 RCC AHB1 外设时钟使能寄存器 (RCC_AHB1ENR) 140
8.3.11 RCC AHB3 外设时钟使能寄存器 (RCC_AHB3ENR) 141
8.3.12 RCC APB1 外设时钟使能寄存器外设时钟使能寄存器 (RCC_APB1ENR) 142
8.3.13 RCC APB2 外设时钟使能寄存器外设时钟使能寄存器 (RCC_APB2ENR) 144
8.3.14 RCC 时钟控制和状态寄存器 (RCC_CSR) 145
8.3.15 RCC 寄存器映射 147
9 通用 I/O (GPIO) 149
9.1 GPIO 简介 149
9.2 GPIO 主要特性 149
9.3 GPIO 功能描述 149
9.3.1 通用 I/O (GPIO) 151
9.3.2 I/O 引脚复用器和映射 151
9.3.3 I/O 端口控制寄存器 154
9.3.4 I/O 端口数据寄存器 155
9.3.5 I/O 数据位操作 155
9.3.6 GPIO 锁定机制 155
9.3.7 I/O 复用功能输入/ 输出 155
9.3.8 外部中断线/ 唤醒线 156
9.3.9 输入配置 156
9.3.10 输出配置 156
9.3.11 复用功能配置 157
9.3.12 模拟配置 157
9.3.13 将 XOSCI/ XOSCO 引脚用作 GPIO P0F/P0E 端口引脚 158
9.3.14 选择复用功能 158
9.4 GPIO 寄存器 158
9.4.1 GPIO 端口模式寄存器 (GPIOx_MODER) (x = A..I) 158
9.4.2 GPIO 端口输出类型寄存器 (GPIOx_OTYPER) (x = A..I) 159
9.4.3 GPIO 端口输出速度寄存器 (GPIOx_OSPEEDR) (x = A..I/) 159
9.4.4 GPIO 端口上拉/ 下拉寄存器 (GPIOx_PUPDR) (x = A..I/) 160
9.4.5 GPIO 端口输入数据寄存器 (GPIOx_IDR) (x = A..I) 160
9.4.6 GPIO 端口输出数据寄存器 (GPIOx_ODR) (x = A..I) 160
9.4.7 GPIO 端口置位/ 复位寄存器 (GPIOx_BSRR) (x = A..I) 161
9.4.8 GPIO 端口配置锁定寄存器 (GPIOx_LCKR) (x = A..I) 161
9.4.9 GPIO 复用功能低位寄存器 (GPIOx_AFRL) (x = A..I) 162
9.4.10 GPIO 复用功能高位寄存器 (GPIOx_AFRH) (x = A..I) 163
9.4.11 GPIO 寄存器映射 163
10 系统配置控制器 (SYSCFG) 165

10.1.1 SYSCFG 外部中断配置寄存器 1 (SYSCFG_EXTICR1) 165
10.1.2 SYSCFG 外部中断配置寄存器 2 (SYSCFG_EXTICR2) 165
10.1.3 SYSCFG 外部中断配置寄存器 3 (SYSCFG_EXTICR3) 165
10.1.4 SYSCFG 外部中断配置寄存器 4 (SYSCFG_EXTICR4) 166
10.1.5 SYSCFG 寄存器映射 166
11 中断和事件 168
11.1 嵌套向量中断控制器 (NVIC) 168
11.1.1 NVIC 特性 168
11.1.2 SysTick 校准值寄存器 168
11.1.3 中断和异常向量 168
11.2 外部中断/ 事件控制器 (EXTI) 171
11.2.1 EXTI 主要特性 171
11.2.2 EXTI 框图 172
11.2.3 唤醒事件管理 172
11.2.4 功能说明 172
11.2.5 外部中断/ 事件线映射 173
11.2.6 EXTI 寄存器 173
11.2.7 中断屏蔽寄存器 (EXTI_IMR) 174
11.2.8 事件屏蔽寄存器 (EXTI_EMR) 174
11.2.9 上升沿触发选择寄存器 (EXTI_RTSR) 174
11.2.10 下降沿触发选择寄存器 (EXTI_FTSR) 175
11.2.11 软件中断事件寄存器 (EXTI_SWIER) 175
11.2.12 挂起寄存器 (EXTI_PR) 175
11.2.13 EXTI 寄存器映射 176
12 模数转换器 (ADC) 177
12.1 ADC 简介 177
12.2 ADC 主要特性 177
12.3 ADC 功能说明 177
12.3.1 ADC 开关控制 178
12.3.2 ADC 时钟 178
12.3.3 通道选择 178
12.3.4 单次转换模式 178
12.3.5 连续转换模式 179
12.3.6 时序图 179
12.3.7 扫描模式 179
12.3.8 注入通道管理 180
12.3.9 不连续采样模式 181
12.4 数据对齐 182
12.5 可独立设置各通道采样时间 183
12.6 外部触发转换和触发极性 183
12.7 快速转换模式 184
12.8 数据管理 185
12.8.1 使用 DMA 185
12.8.2 在不使用 DMA 的情况下管理转换序列 185

12.8.3 在不使用 DMA 和溢出检测的情况下进行转换 185
12.9 ADC 中断 186
12.10 ADC 寄存器 186
12.10.1 ADC 状态寄存器 (ADC_SR) 186
12.10.2 ADC 控制寄存器 1 (ADC_CR1) 187
12.10.3 ADC 控制寄存器 2 (ADC_CR2) 188
12.10.4 ADC 采样时间寄存器 1 (ADC_SMPR1) 190
12.10.5 ADC 采样时间寄存器 2 (ADC_SMPR2) 190
12.10.6 ADC 注入通道数据偏移寄存器 X (ADC_JOFRx)(x=1..4) 191
12.10.7 ADC 规则序列寄存器 1 (ADC_SQR1) 191
12.10.8 ADC 规则序列寄存器 2 (ADC_SQR2) 192
12.10.9 ADC 规则序列寄存器 3 (ADC_SQR3) 192
12.10.10 ADC 注入序列寄存器 (ADC_JSQR) 193
12.10.11 ADC 注入数据寄存器 x (ADC_JDRx) (x= 1..4) 193
12.10.12 ADC 规则数据寄存器 (ADC_DR) 194
12.10.13 ADC 通用状态寄存器 (ADC_CSR) 194
12.10.14 ADC 通用控制寄存器 (ADC_CCR) 195
12.10.15 ADC 寄存器映射 196
13 SPI 串行外设接口及 I2S 198
13.1 SPI 简介 198
13.2 SPI 和 I2S 主要特性 198
13.2.1 SPI 特性 198
13.2.2 I2S 198
13.3 SPI 199
13.3.1 一般说明 199
13.3.2 把 SPI 配置成从器件 201
13.3.3 把 SPI 配置成主器件 203
13.3.4 配置 SPI 进行半双工通信 205
13.3.5 数据发送和接收过程 205
13.3.6 CRC 计算 210
13.3.7 状态标志 211
13.3.8 关闭 SPI 211
13.3.9 使用 DMA （直接存储器寻址）进行 SPI 通信 212
13.3.10 错误标志 214
13.3.11 SPI 中断 215
13.4 I2S 215
13.4.1 I2S 一般说明 215
13.4.2 I2S 全双工 216
13.4.3 支持的音频协议 217
13.4.4 时钟发生器 223
13.4.5 I2S 主模式 225
13.4.6 I2S 从模式 227
13.4.7 状态标志 228
13.4.8 错误标志 228

13.4.9 I 2 S 中断 229
13.4.10 DMA....229
13.5 SPI 和 I 2 S....229
13.5.1 SPI 控制寄存器 1 (SPI_CR1) （不用于 I 2 S 模式） 229
13.5.2 SPI 控制寄存器 2 (SPI_CR2)....231
13.5.3 SPI 状态寄存器 (SPI_SR)....232
13.5.4 SPI 数据寄存器 (SPI_DR) 233
13.5.5 SPI CRC 多项式寄存器 (SPI_CRCPR) （不用于 I 2 S 模式）....233
13.5.6 SPI RX CRC 寄存器 (SPI_RXCRCR) （不用于 I 2 S 模式）....233
13.5.7 SPI TX CRC 寄存器 (SPI_TXCRCR) （不用于 I 2 S 模式） 234
13.5.8 SPI_I 2 S 配置寄存器 (SPI_I2SCFGR) 234
13.5.9 SPI_I 2 S 预分频器寄存器 (SPI_I2SPR)....235
13.5.10 SPI....236
14 I 2 C 串行外设接口 238
14.1 I 2 C 简介....238
14.2 I 2 C 主要特性....238
14.3 I 2 C 功能说明....239
14.3.1 模式选择....239
14.3.2 I 2 C 从模式....240
14.3.3 I 2 C 主模式....242
14.3.4 错误条件....245
14.3.5 SDA/SCL 线控制....246
14.3.6 DMA 请求....246
14.3.7 数据包错误校验....247
14.4 I 2 C 中断....248
14.5 I 2 C 寄存器....249
14.5.1 I 2 C 控制寄存器 1 (I2C_CR1)....249
14.5.2 I 2 C 控制寄存器 2 (I2C_CR2)....251
14.5.3 I 2 C 自有地址寄存器 1 (I2C_OAR1)....252
14.5.4 I 2 C 自有地址寄存器 2 (I2C_OAR2)....252
14.5.5 I 2 C 数据寄存器 (I2C_DR)....252
14.5.6 I 2 C 状态寄存器 1 (I2C_SR1)....253
14.5.7 I 2 C 状态寄存器 2 (I2C_SR2)....255
14.5.8 I 2 C 时钟控制寄存器 (I2C_CCR) 256
14.5.9 I 2 C TRISE 寄存器 (I2C_TRISE) 257
14.5.10 I 2 C 寄存器映射....257
15 高级控制定时器（PWM1 和 PWM8）....259
15.1 PWM1 和 PWM8 简介....259
15.2 PWM1 和 PWM8 主要特性....259
15.3 PWM1 和 PWM8 功能说明....260
15.3.1 时基单元....260
15.3.2 计数器模式....261
15.3.3 重复计数器....268
15.3.4 时钟选择....269

15.3.5 捕获/比较通道 271
15.3.6 输入捕获模式 273
15.3.7 PWM 输入模式 274
15.3.8 强制输出模式 275
15.3.9 输出比较模式 275
15.3.10 PWM 模式 276
15.3.11 互补输出和死区插入 278
15.3.12 使用断路功能 279
15.3.13 发生外部事件时清除 OCxREF 信号 281
15.3.14 生成 6 步 PWM 282
15.3.15 单脉冲模式 283
15.3.16 编码器接口模式 284
15.3.17 定时器输入异或功能 286
15.3.18 连接霍尔传感器 286
15.3.19 PWMx 与外部触发同步 287
15.3.20 定时器同步 289
15.3.21 调试模式 290
15.4 PWM1 和 PWM8 寄存器 290
15.4.1 PWM1 和 PWM8 控制寄存器 1 (PWMx_CR1) 290
15.4.2 PWM1 和 PWM8 控制寄存器 2 (PWMx_CR2) 291
15.4.3 PWM1 和 PWM8 从模式控制寄存器 (PWMx_SMCR) 292
15.4.4 PWM1 和 PWM8 DMA/中断使能寄存器 294
15.4.5 PWM1 和 PWM8 状态寄存器 (PWMx_SR) 295
15.4.6 PWM1 和 PWM8 事件生成寄存器 (PWMx_EGR) 296
15.4.7 PWM1 和 PWM8 捕获/比较模式寄存器 1 (PWMx_CCMR1) 297
15.4.8 PWM1 和 PWM8 捕获/比较模式寄存器 2 (PWMx_CCMR2) 300
15.4.9 PWM1 和 PWM8 捕获/比较使能寄存器 (PWMx_CCER) 301
15.4.10 PWM1 和 PWM8 计数器 (PWMx_CNT) 303
15.4.11 PWM1 和 PWM8 预分频器 (PWMx_PSC) 304
15.4.12 PWM1 和 PWM8 自动重载寄存器 (PWMx_ARR) 304
15.4.13 PWM1 和 PWM8 重复计数器寄存器 (PWMx_RCR) 304
15.4.14 PWM1 和 PWM8 捕获/比较寄存器 1 (PWMx_CCR1) 304
15.4.15 PWM1 和 PWM8 捕获/比较寄存器 2 (PWMx_CCR2) 305
15.4.16 PWM1 和 PWM8 捕获/比较寄存器 3 (PWMx_CCR3) 305
15.4.17 PWM1 和 PWM8 捕获/比较寄存器 4 (PWMx_CCR4) 306
15.4.18 PWM1 和 PWM8 断路和死区寄存器 (PWMx_BDTR) 306
15.4.19 PWM1 和 PWM8 DMA 控制寄存器 (PWMx_DCR) 308
15.4.20 PWM1 和 PWM8 全传输 DMA 地址 (PWMx_DMAR) 308
15.4.21 PWM1 和 PWM8 寄存器映射 309
16 通用定时器（PWM2 到 PWM5） 312
16.1 PWM2 到 PWM5 简介 312
16.2 PWM2 到 PWM5 主要特性 312
16.3 PWM2 到 PWM5 功能说明 313
16.3.1 时基单元 313

16.3.2 计数器模式 314
16.3.3 时钟选择 321
16.3.4 捕获/比较通道 323
16.3.5 输入捕获模式 324
16.3.6 PWM 输入模式 325
16.3.7 强制输出模式 326
16.3.8 输出比较模式 326
16.3.9 PWM 模式 327
16.3.10 单脉冲模式 329
16.3.11 发生外部事件时清除 OCxREF 信号 330
16.3.12 编码器接口模式 331
16.3.13 定时器输入异或功能 332
16.3.14 定时器与外部触发同步 332
16.3.15 定时器同步 334
16.3.16 调试模式 338
16.4 PWM2 到 PWM5 寄存器 338
16.4.1 PWMx 控制寄存器 1 (PWMx_CR1) 338
16.4.2 PWMx 控制寄存器 2 (PWMx_CR2) 339
16.4.3 PWMx 从模式控制寄存器 (PWMx_SMCR) 340
16.4.4 PWMx DMA/中断使能寄存器 (PWMx_DIER) 342
16.4.5 PWMx 状态寄存器 (PWMx_SR) 343
16.4.6 PWMx 事件生成寄存器 (PWMx_EGR) 344
16.4.7 PWMx 捕获/比较模式寄存器 1 (PWMx_CCMR1) 345
16.4.8 PWMx 捕获/比较模式寄存器 2 (PWMx_CCMR2) 347
16.4.9 PWMx 捕获/比较使能寄存器 (PWMx_CCER) 348
16.4.10 PWMx 计数器 (PWMx_CNT) 349
16.4.11 PWMx 预分频器 (PWMx_PSC) 350
16.4.12 PWMx 自动重载寄存器 (PWMx_ARR) 350
16.4.13 PWMx 捕获/比较寄存器 1 (PWMx_CCR1) 350
16.4.14 PWMx 捕获/比较寄存器 2 (PWMx_CCR2) 351
16.4.15 PWMx 捕获/比较寄存器 3 (PWMx_CCR3) 351
16.4.16 PWMx 捕获/比较寄存器 4 (PWMx_CCR4) 352
16.4.17 PWMx DMA 控制寄存器 (PWMx_DCR) 352
16.4.18 PWMx 全传输 DMA 地址 (PWMx_DMAR) 353
16.4.19 PWM2 选项寄存器 (PWM2_OR) 353
16.4.20 PWM5 选项寄存器 (PWM5_OR) 354
16.4.21 PWMx 寄存器映射 354
17 通用定时器（PWM9 到 PWM14） 357
17.1 PWM9 到 PWM14 简介 357
17.2 PWM9 到 PWM14 主要特性 357
17.2.1 PWM9/PWM12 主要特性 357
17.3 PWM10/PWM11 和 PWM13/PWM14 主要特性 358
17.4 PWM9 到 PWM14 功能说明 359
17.4.1 时基单元 359

17.4.2 计数器模式 360
17.4.3 时钟选择 362
17.4.4 捕获/比较通道 364
17.4.5 输入捕获模式 365
17.4.6 PWM 输入模式（仅适用于 PWM9/12） 366
17.4.7 强制输出模式 366
17.4.8 输出比较模式 367
17.4.9 PWM 模式 367
17.4.10 单脉冲模式 368
17.4.11 PWM9/12 外部触发同步 369
17.4.12 定时器同步 (PWM9/12) 371
17.4.13 调试模式 371
17.5 PWM9 和 PWM12 寄存器 371
17.5.1 PWM9/12 控制寄存器 1 (PWMx_CR1) 371
17.5.2 PWM9/12 控制寄存器 2 (PWMx_CR2) 372
17.5.3 PWM9/12 从模式控制寄存器 (PWMx_SMCR) 372
17.5.4 PWM9/12 中断使能寄存器 (PWMx_DIER) 374
17.5.5 PWM9/12 状态寄存器 (PWMx_SR) 374
17.5.6 PWM9/12 事件生成寄存器 (PWMx_EGR) 375
17.5.7 PWM9/12 捕获/比较模式寄存器 1 (PWMx_CCMR1) 376
17.5.8 PWM9/12 捕获/比较使能寄存器 (PWMx_CCER) 378
17.5.9 PWM9/12 计数器 (PWMx_CNT) 379
17.5.10 PWM9/12 预分频器 (PWMx_PSC) 379
17.5.11 PWM9/12 自动重载寄存器 (PWMx_ARR) 379
17.5.12 PWM9/12 捕获/比较寄存器 1 (PWMx_CCR1) 380
17.5.13 PWM9/12 捕获/比较寄存器 2 (PWMx_CCR2) 380
17.5.14 PWM9/12 寄存器映射 380
17.6 PWM10/11/13/14 寄存器 382
17.6.1 PWM10/11/13/14 控制寄存器 1 (PWMx_CR1) 382
17.6.2 PWM10/11/13/14 状态寄存器 (PWMx_SR) 383
17.6.3 PWM10/11/13/14 事件生成寄存器 (PWMx_EGR) 383
17.6.4 PWM10/11/13/14 捕获/比较模式寄存器 1 (PWMx_CCMR1) 384
17.6.5 PWM10/11/13/14 捕获/比较使能寄存器 (PWMx_CCER) 386
17.6.6 PWM10/11/13/14 计数器 (PWMx_CNT) 386
17.6.7 PWM10/11/13/14 预分频器 (PWMx_PSC) 387
17.6.8 PWM10/11/13/14 自动重载寄存器 (PWMx_ARR) 387
17.6.9 PWM10/11/13/14 捕获/比较寄存器 1 (PWMx_CCR1) 387
17.6.10 PWM11 选项寄存器 1 (PWM11_OR) 387
17.6.11 PWM10/11/13/14 寄存器映射 388
18 独立看门狗 (IWDG) 390
18.1 IWDG 简介 390
18.2 IWDG 主要特性 390
18.3 IWDG 功能说明 390
18.3.1 硬件看门狗 390

18.3.2 寄存器访问保护 390
18.3.3 调试模式 391
18.4 IWDG 寄存器 391
18.4.1 关键字寄存器 (IWDG_KR) 391
18.4.2 预分频器寄存器 (IWDG_PR) 392
18.4.3 重载寄存器 (IWDG_RLR) 392
18.4.4 状态寄存器 (IWDG_SR) 393
18.4.5 IWDG 寄存器映射 393
19 窗口看门狗 (WWDG) 395
19.1 WWDG 简介 395
19.2 WWDG 主要特性 395
19.3 WWDG 功能说明 395
19.4 如何设置看门狗超时 396
19.5 调试模式 397
19.6 WWDG 397
19.6.1 控制寄存器 (WWDG_CR) 397
19.6.2 配置寄存器 (WWDG_CFR) 398
19.6.3 状态寄存器 (WWDG_SR) 398
19.6.4 WWDG 寄存器映射 399
20 通用同步异步收发器 (USART) 400
20.1 USART 简介 400
20.2 USART 主要特性 400
20.3 USART 功能说明 401
20.3.1 USART 字符说明 402
20.3.2 发送器 403
20.3.3 接收器 405
20.3.4 小数波特率生成 409
20.3.5 USART 接收器对时钟偏差的容差 417
20.3.6 多处理器通信 418
20.3.7 奇偶校验控制 419
20.3.8 LIN （局域互连网络）模式 420
20.3.9 单线半双工通信 423
20.3.10 智能卡 423
20.3.11 IrDA SIR ENDEC 模块 425
20.3.12 使用 DMA 426
20.3.13 硬件流控制 428
20.4 USART 429
20.5 USART 模式配置 430
20.6 USART 430
20.6.1 状态寄存器 (USART_SR) 430
20.6.2 数据寄存器 (USART_DR) 432
20.6.3 波特率寄存器 (USART_BRR) 433
20.6.4 控制寄存器 1 (USART_CR1) 433
20.6.5 控制寄存器 2 (USART_CR2) 435

20.6.6 控制寄存器 3 (USART_CR3) 436
20.6.7 保护时间和预分频器寄存器 (USART_GTPR) 438
20.6.8 USART 寄存器映射 438
21 DMA 控制器， 支持外设到外设 440
21.1 DMA 简介 440
21.2 DMA 主要特性 440
21.3 DMA 功能说明 441
21.3.1 一般说明 441
21.3.2 DMA 事务 441
21.3.3 通道选择 441
21.3.4 仲裁器 443
21.3.5 DMA 数据流 443
21.3.6 源、目标和传输模式 443
21.3.7 指针递增 447
21.3.8 循环模式 447
21.3.9 双缓冲区模式 448
21.3.10 可编程数据宽度、封装/ 解封、字节序 448
21.3.11 单次传输和突发传输 450
21.3.12 FIFO 451
21.3.13 DMA 传输完成 453
21.3.14 DMA 传输暂停 454
21.3.15 流控制器 454
21.3.16 可能的 DMA 配置汇总 455
21.3.17 流配置过程 455
21.3.18 错误管理 456
21.4 DMA 中断 457
21.5 DMA 寄存器 458
21.5.1 DMA 低中断状态寄存器 (DMA_LISR) 458
21.5.2 DMA 高中断状态寄存器 (DMA_HISR) 459
21.5.3 DMA 低中断标志清零寄存器 (DMA_LIFCR) 460
21.5.4 DMA 高中断标志清零寄存器 (DMA_HIFCR) 461
21.5.5 DMA 数据流 x 配置寄存器 (DMA_SxCR) (x = 0..7) 462
21.5.6 DMA 数据流 x 数据项数寄存器 (DMA_SxNDTR) (x = 0..7) 465
21.5.7 DMA 数据流 x 外设地址寄存器 (DMA_SxPAR) (x = 0..7) 465
21.5.8 DMA 数据流 x 存储器 0 地址寄存器 (DMA_SxM0AR) (x = 0..7) 466
21.5.9 DMA 数据流 x 存储器 1 地址寄存器 (DMA_SxM1AR) (x = 0..7) 466
21.5.10 DMA 数据流 x FIFO 控制寄存器 (DMA_SxFCR) (x = 0..7) 467
21.5.11 DMA 寄存器映射 468
22 Cache/TCM/存储器和总线架构 474
22.1 系统架构 474
22.1.1 S0 ： I 总线 475
22.1.2 S1 ： D 总线 475
22.1.3 S2 ： S 总线 475
22.1.4 S3 S4 ： DMA 存储器总线 475

22.1.5 S5 ： DMA 外设总线.....476
22.1.6 总线矩阵.....476
22.1.7 AHB/APB 总线桥 (APB).....476
22.2 存储器组织结构.....476
22.3 存储器映射.....476
23 文档约定.....479
23.1 寄存器相关缩写词列表.....479
23.2 词汇表.....479
附录 A 更新记录.....481
附录 B 产品授权书.....484

STC MCU

1 特性

1.1 主要特色

- 工作电压: 1.9V ~ 5.5V
- 工作温度: -40℃ ~ 125℃
- 内核: ARM, STAR-MC1/M33 核
  - 96K Byte SRAM
- FLASH
  - 512KB Flash
- 振荡系统:
  - PLL 时钟 CPU 最高主频可达 144MHz，外设最高主频可达 288MHz
  - IHRC 内振 72M Hz 可调整精度 +/- 25%, 温飘 +/-1.2%
  - XTAL 48MHz Crystal Oscillator
  - ILRC 内振 32K Hz ILRC, 对电压精度误差 +/- 25%, 对温度误差 +/- 25% .
- 复位机制: (a) POR+LVR (b)/RESET-pin (c) LVD (d)WDT (e)Software Reset
- 数字外围:

■DMA: 8CH X 2

■17 个 TIMER (T1,T2,T3,T4,T5,T6,T7,T8,T9,T10,T11,T12,T13,T14)

– PWM1 and PWM8 2 × 16-bit motor control PWM with dead-PWM generation and emergency stop

– 2 × watchdog PWM (Independent and Window)

– 系统节拍定时器: a 24-bit downcounter

USART *4 (USART1 *4)

■I2C *2

–I2C Fast Mode (1400K bit/s) support

■SPI *2： SPI 1 *1 ， SPI 2/I2S *1， 最高速 1/2 主频

■CRC calculation unit, 96-bit unique ID

■USB 2.0/1.1 Full-Speed 功能 6 个双向端点，支持 4 种端点传输模式

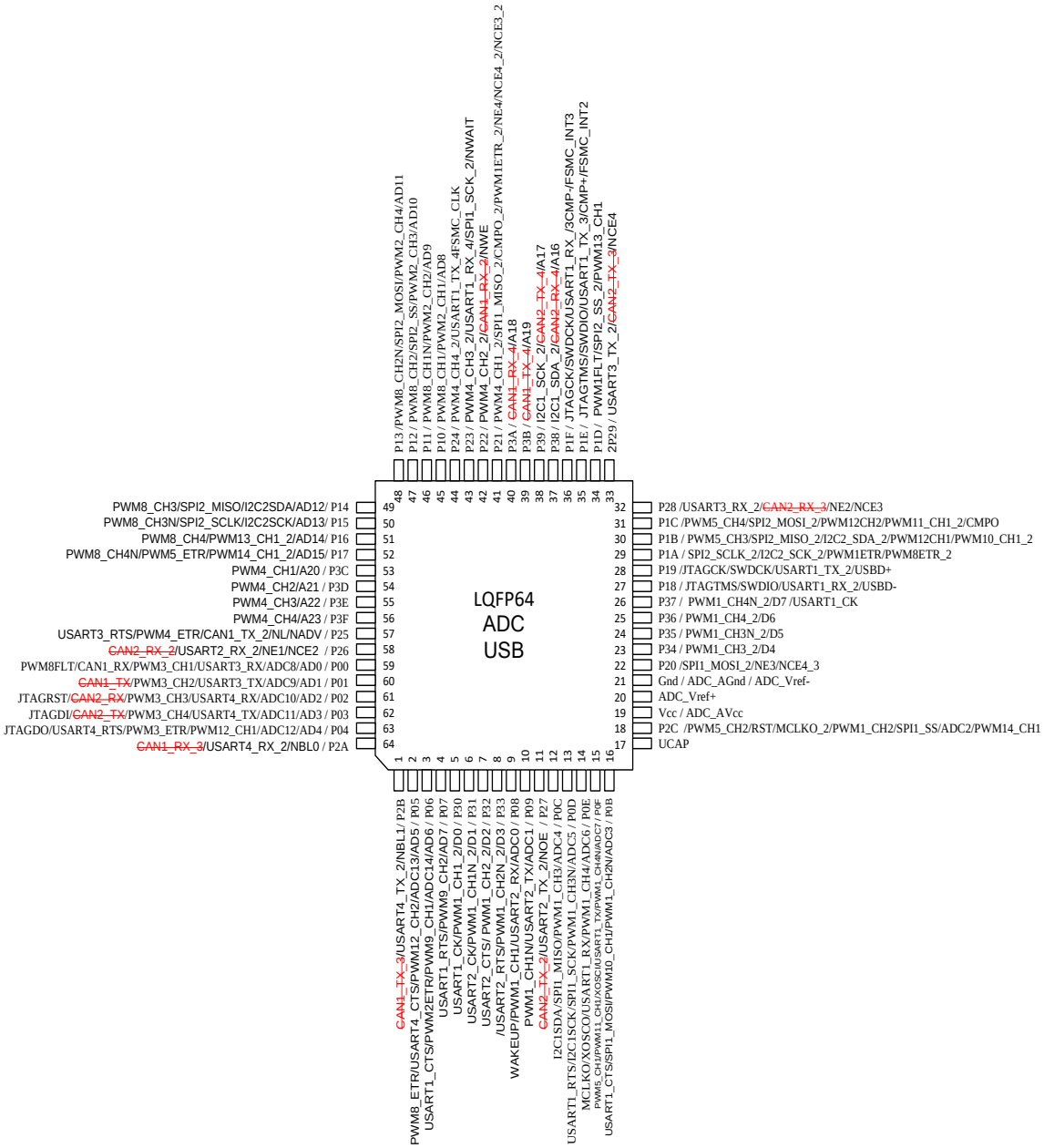
- 模拟外围:
  - 2 组独立 12-bit ADC，1Mbps
  - 4 组独立轨到轨比较器
  - 4-Level Low-Voltage-Detector (1.9V/2.3V/2.6V/2.9V)

- 封装:
  - LQFP-100，LQFP-64，LQFP-48 封装

1.2 模拟模块

1. 内振 72M Hz 可调整精度 +/- 25%, 温飘 +/-1.2% :
2. 内振 32K Hz ILRC, 对电压精度误差 +/- 25%, 对温度误差 +/- 25%
3. XTAL 32K/4M~48M 起振 OSC PAD
4. BandGap 电压的中心点设计为 1.188V
5. LVD 的 4 级电压设计为{1.9V, 2.3V, 2.6V, 2.9V},有 +0.05V/- 0.1V 的迟滞, 有 2uS 的 Filter(可 option 到 10uS)
6. ADC 12bits/800KHZ

1.3封装


LQFP64
ADC
USB
1 P2B / CAN1_TX_3/USART4_TX_2/NBL1
2 P05 / PWM8_ETR/USART4_CTS/PWM12_CH2/ADC13/AD5
3 P06 / USART1_CTS/PWM2ETR/PWM9_CH1/ADC14/AD6
4 P07 / USART1_RTS/PWM9_CH2/AD7
5 P30 / USART1_CK/PWM1_CH1_2/D0
6 P31 / USART2_CK/PWM1_CH1N_2/D1
7 P32 / USART2_CTS/ PWM1_CH2_2/D2
8 P33 / /USART2_RTS/PWM1_CH2N_2/D3
9 P08 / WAKEUP/PWM1_CH1/USART2_RX/ADC0
10 P09 / PWM1_CH1N/USART2_TX/ADC1
11 P27 / CAN2_TX_2/USART2_TX_2/NOE
12 P0C / I2C1SDA/SPI1_MISO/PWM1_CH3/ADC4
13 P0D / USART1_RTS/I2C1SCK/SPI1_SCK/PWM1_CH3N/ADC5
14 P0E / MCLKO/XOSCO/USART1_RX/PWM1_CH4/ADC6
15 P0F / PWM5_CH1/PWM11_CH1/XOSCI/USART1_TX/PWM1_CH4N/ADC7
16 P0B / USART1_CTS/SPI1_MOSI/PWM10_CH1/PWM1_CH2N/ADC3
17 UCAP
18 P2C /PWM5_CH2/RST/MCLKO_2/PWM1_CH2/SPI1_SS/ADC2/PWM14_CH1
19 Vcc / ADC_AVcc
20 ADC_Vref+
21 Gnd / ADC_AGnd / ADC_Vref-
22 P20 /SPI1_MOSI_2/NE3/NCE4_3
23 P34 / PWM1_CH3_2/D4
24 P35 / PWM1_CH3N_2/D5
25 P36 / PWM1_CH4_2/D6
26 P37 / PWM1_CH4N_2/D7 /USART1_CK
27 P18 / JTAGTMS/SWDIO/USART1_RX_2/USBD-
28 P19 /JTAGCK/SWDCK/USART1_TX_2/USBD+
29 P1A / SPI2_SCLK_2/I2C2_SCK_2/PWM1ETR/PWM8ETR_2
30 P1B / PWM5_CH3/SPI2_MISO_2/I2C2_SDA_2/PWM12CH1/PWM10_CH1_2
31 P1C /PWM5_CH4/SPI2_MOSI_2/PWM12CH2/PWM11_CH1_2/CMPO
32 P28 /USART3_RX_2/CAN2_RX_3/NE2/NCE3
33 2P29 / USART3_TX_2/CAN2_TX_3/NCE4
34 P1D / PWM1FLT/SPI2_SS_2/PWM13_CH1
35 P1E / JTAGTMS/SWDIO/USART1_TX_3/CMP+/FSMC_INT2
36 P1F / JTAGCK/SWDCK/USART1_RX_/3CMP-/FSMC_INT3
37 P38 / I2C1_SDA_2/CAN2_RX_4/A16
38 P39 / I2C1_SCK_2/CAN2_TX_4/A17
39 P3B / CAN1_TX_4/A19
40 P3A / CAN1_RX_4/A18
41 P21 / PWM4_CH1_2/SPI1_MISO_2/CMPO_2/PWM1ETR_2/NE4/NCE4_2/NCE3_2
42 P22 / PWM4_CH2_2/CAN1_RX_3/NWE
43 P23 / PWM4_CH3_2/USART1_RX_4/SPI1_SCK_2/NWAIT
44 P24 / PWM4_CH4_2/USART1_TX_4/FSMC_CLK
45 P10 / PWM8_CH1/PWM2_CH1/AD8
46 P11 / PWM8_CH1N/PWM2_CH2/AD9
47 P12 / PWM8_CH2/SPI2_SS/PWM2_CH3/AD10
48 P13 /PWM8_CH2N/SPI2_MOSI/PWM2_CH4/AD11
49 PWM8_CH3/SPI2_MISO/I2C2SDA/AD12/ P14
50 PWM8_CH3N/SPI2_SCLK/I2C2SCK/AD13/ P15
51 PWM8_CH4/PWM13_CH1_2/AD14/ P16
52 PWM8_CH4N/PWM5_ETR/PWM14_CH1_2/AD15/ P17
53 PWM4_CH1/A20 / P3C
54 PWM4_CH2/A21 / P3D
55 PWM4_CH3/A22 / P3E
56 PWM4_CH4/A23 / P3F
57 USART3_RTS/PWM4_ETR/CAN1_TX_2/NL/NADV / P25
58 CAN2_RX_2/USART2_RX_2/NE1/NCE2 / P26
59 PWM8FLT/CAN1_RX/PWM3_CH1/USART3_RX/ADC8/AD0 / P00
60 CAN1_TX/PWM3_CH2/USART3_TX/ADC9/AD1 / P01
61 JTAGRST/CAN2_RX/PWM3_CH3/USART4_RX/ADC10/AD2 / P02
62 JTAGDI/CAN2_TX/PWM3_CH4/USART4_TX/ADC11/AD3 / P03
63 JTAGDO/USART4_RTS/PWM3_ETR/PWM12_CH1/ADC12/AD4 / P04
64 CAN1_RX_3/USART4_RX_2/NBL0 / P2A

IO Define

```
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//////////////
//
//   P00                  PWM8FLT/CAN1_RX/PWM3_CH1/USART3_RX/ADC8/AD0
//   P01                    CAN1_TX/PWM3_CH2/USART3_TX/ADC9/AD1
//   P02                JTAGRST/CAN2_RX/PWM3_CH3/USART4_RX/ADC10/AD2
//   P03                JTAGDI/CAN2_TX/PWM3_CH4/USART4_TX/ADC11/AD3
//   P04              JTAGDO/USART4_RTS/PWM3_ETR/PWM12_CH1/ADC12/AD4
//   P05                    PWM8_ETR/USART4_CTS/PWM12_CH2/ADC13/AD5
//   P06                  USART1_CTS/PWM2ETR/PWM9_CH1/ADC14/AD6
//   P07                              USART1_RTS/PWM9_CH2/AD7
//   P08                  WAKEUP/PWM1_CH1/USART2_RX/ADC0
//   P09                      PWM1_CH1N/USART2_TX/ADC1
//   P0A
//   P0B          USART1_CTS/SPI1_MOSI/PWM10_CH1/PWM1_CH2N/ADC3
//   P0C                  I2C1SDA/SPI1_MISO/PWM1_CH3/ADC4
//   P0D            USART1_RTS/I2C1SCK/SPI1_SCK/PWM1_CH3N/ADC5
//   P0E                MCLKO/XOSCO/USART1_RX/PWM1_CH4/ADC6
//   P0F      PWM5_CH1/PWM11_CH1/XOSCI/USART1_TX/PWM1_CH4N/ADC7


//   P10                                  PWM8_CH1/PWM2_CH1/AD8
//   P11                                PWM8_CH1N/PWM2_CH2/AD9
//   P12                            PWM8_CH2/SPI2_SS/PWM2_CH3/AD10
//   P13                          PWM8_CH2N/SPI2_MOSI/PWM2_CH4/AD11
//   P14                            PWM8_CH3/SPI2_MISO/I2C2SDA/AD12
//   P15                           PWM8_CH3N/SPI2_SCLK/I2C2SCK/AD13
//   P16                               PWM8_CH4/PWM13_CH1_2/AD14
//   P17                        PWM8_CH4N/PWM5_ETR/PWM14_CH1_2/AD15
//   P18                        JTAGTMS/SWDIO/USART1_TX_2/USBD+
//   P19                         JTAGCK/SWDCK/USART1_RX_2/USBD-
//   P1A                SPI2_SCLK_2/I2C2_SCK_2/PWM1ETR/PWM8ETR_2
//   P1B  PWM5_CH3/SPI2_MISO_2/I2C2_SDA_2/PWM12CH1/PWM10_CH1_2
//   P1C        PWM5_CH4/SPI2_MOSI_2/PWM12CH2/PWM11_CH1_2/CMPO
//   P1D                            PWM1FLT/SPI2_SS_2/PWM13_CH1
//   P1E                    JTAGTMS/SWDIO/USART1_TX_3/CMP+/FSMC_INT2
//   P1F                     JTAGCK/SWDCK/USART1_RX_3/CMP-/FSMC_INT3


//   P20                                        SPI1_MOSI_2/NE3/NCE4_3
//   P21              PWM4_CH1_2/SPI1_MISO_2/CMPO_2/PWM1ETR_2/NE4/NCE4_2/NCE3_2
//   P22                                PWM4_CH2_2/CAN1_RX_2/NWE
//   P23                  PWM4_CH3_2/USART1_RX_4/SPI1_SCK_2/NWAIT
//   P24                    PWM4_CH4_2/USART1_TX_4/FSMC_CLK
//   P25                     USART3_RTS/PWM4_ETR/CAN1_TX_2/NL/NADV
//   P26                   CAN2_RX_2/USART2_RX_2/NE1/NCE2
//   P27                            CAN2_TX_2/USART2_TX_2/NOE
//   P28                            USART3_RX_2/CAN2_RX_3/NE2/NCE3
```

```
//  P29                                                 USART3_TX_2/CAN2_TX_3/NCE4
//  P2A                                                 CAN1_RX_3/USART4_RX_2/NBL0
//  P2B                                                 CAN1_TX_3/USART4_TX_2/NBL1
//  P2C  PWM5_CH2/RST/MCLKO_2/PWM1_CH2/SPI1_SS/ADC2/PWM14_CH1
//  P2D
//  P2E
//  P2F
//  P30                                                   USART1_CK /PWM1_CH1_2/D0
//  P31                                                   USART2_CK/PWM1_CH1N_2/D1
//  P32                                                   USART2_CTS/PWM1_CH2_2/D2
//  P33                                                   USART2_RTS/PWM1_CH2N_2/D3
//  P34                                                     PWM1_CH3_2/D4
//  P35                                                    PWM1_CH3N_2/D5
//  P36                                                     PWM1_CH4_2/D6
//  P37                                                    PWM1_CH4N_2/D7
//  P38                                                 I2C1_SDA_2/CAN2_RX_4/A16
//  P39                                                 I2C1_SCK_2/CAN2_TX_4/A17
//  P3A                                                     CAN1_RX_4/A18
//  P3B                                                     CAN1_TX_4/A19
//  P3C                                                      PWM4_CH1/A20
//  P3D                                                      PWM4_CH2/A21
//  P3E                                                      PWM4_CH3/A22
//  P3F                                                      PWM4_CH4/A23
```

2 DSP 与浮点运算单元

2.1 DSP 库

Mstar 内核 DSP 拓展可以为系统增加 85 个新指令。大多数情况下，DSP 指令可将性能平均提升 3 倍，让所有以数字信号控制为中心的应用性能突飞猛进。

为帮助设计师加速软件开发，ARM 将在 CMSIS 项目中提供免费的 DSP 库，包含整套过滤、转换和数学功能（如矩阵），并支持多种数据类型。CMSIS 项目是开源的，其开发详情发布在 github 上。

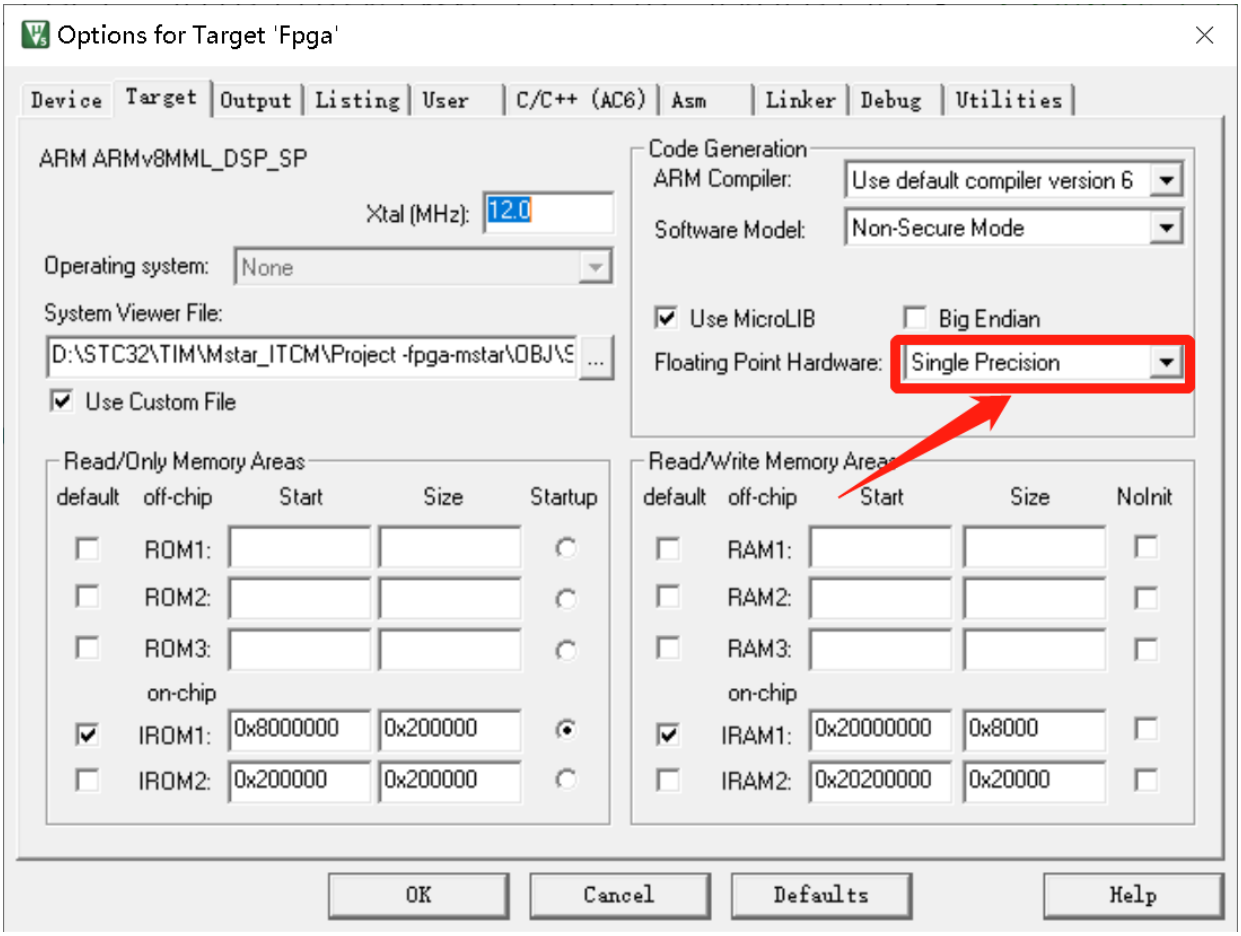
【使用 ARM 官方 DSP 库】

CMSIS 是 Cortex-M 处理器与供应商无关的硬件抽象层软件接口，主要包括以下：
CMSIS-RTOS：主要用于 RTOS 的 API，可与中间件和库组件实现一致的软件层。
CMSIS-DSP：Arm 针对各种 Cortex-M 处理器内核进行了优化的丰富 DSP 功能的集合。
CMSIS-Driver：接口可用于许多微控制器系列。
CMSIS-Pack：定义了包含软件组件的软件包。
CMSIS-SVD：可通过当前寄存器状态显示设备外设的详细视图。
CMSIS-DAP：Cortex 调试访问端口（DAP）的标准化接口。
CMSIS-NN：高效的神经网络内核的集合。

CMSIS 的应用场景非常广泛，目前最新 CMSIS 版本为 V5.7.0
其中， arm_fft_bin_data.c 文件包含了参与运算的复数数组 testInput_f32_10khz，官方默认大小为 2048，意味着参与 FFT 运算的采样点数据个数最大只能到 1024，且为 2 的 N 次方， N 大于 1；arm_ARMv8MMLldfsp_math.lib 为官方提供的 M33 内核的 DSP 库；arm_fft_bin_example_f32.c 为实际的测试函数入口所在文件。

另外，还需要将 arm_math.h 和 arm_const_structs.h 头文件包含进测试.c 文件中，且添加路径。除此之外，在 keil 工程设置中还需要添加预处理定义： ARM_MATH_CM33,_FPU_USED=1U；在 Misc Controls 一栏中还需要添加相关命令避免编译器报错： -fshort-enums -fshort-wchar

准备空工程，配置 Keil 环境，使能 FPU 单元。
开启硬件浮点运算，等效于在 C/C++->define 中定义__FPU_USED,__FPU_PRESENT 两个宏



添加全局宏定义，使能 DSP 库所有的功能

其他宏 ARM_MATH_CM4,__CC_ARM,ARM_MATH_MATRIX_CHECK,ARM_MATH_ROUNDING

向工程中添加使用到的 DSP 库源码

在\Libraries\CMSIS\DSP_Lib\Source 目录下会有如下目录，都是 DSP 函数库



名称	修改日期	类型	大小
BasicMathFunctions	2019/7/10 21:30	文件夹	
CommonTables	2019/7/10 21:30	文件夹	
ComplexMathFunctions	2019/7/10 21:30	文件夹	
ControllerFunctions	2019/7/10 21:30	文件夹	
FastMathFunctions	2019/7/10 21:30	文件夹	
FilteringFunctions	2019/7/10 21:30	文件夹	
MatrixFunctions	2019/7/10 21:30	文件夹	
StatisticsFunctions	2019/7/10 21:30	文件夹	
SupportFunctions	2019/7/10 21:30	文件夹	
TransformFunctions	2019/7/10 21:30	文件夹	
CMakeLists.txt	2019/7/10 21:30	文本文档	10 KB

BasicMathFunctions
基本数学函数：提供浮点数的各种基本运算函数，如向量加减乘除等运算。

CommonTables
arm_common_tables.c 文件提供位翻转或相关参数表。

ComplexMathFunctions
复杂数学功能，如向量处理，求模运算的。

ControllerFunctions
控制功能函数。包括正弦余弦，PID 电机控制，矢量 Clarke 变换，矢量 Clarke 逆变换等。

FastMathFunctions
快速数学功能函数。提供了一种快速的近似正弦，余弦和平方根等相比 CMSIS 计算库要快的数学函数。

FilteringFunctions
滤波函数功能，主要为 FIR 和 LMS（最小均方根）等滤波函数。

MatrixFunctions
矩阵处理函数。包括矩阵加法、矩阵初始化、矩阵反、矩阵乘法、矩阵规模、矩阵减法、矩阵转置等函数。

StatisticsFunctions
统计功能函数。如求平均值、最大值、最小值、计算均方根 RMS、计算方差/标准差等。

SupportFunctions
支持功能函数，如数据拷贝，Q 格式和浮点格式相互转换，Q 任意格式相互转换。

TransformFunctions
变换功能。包括复数 FFT（CFFT）/复数 FFT 逆运算（CIFFT）、实数 FFT（RFFT）/实数 FFT

逆运算（RIFFT）、和 DCT（离散余弦变换）和配套的初始化函数。

我们需要 CommonTables 和 TransformFunctions 下的部分文件添加到工程

另外我们需要包含相应的头文件路径，所需的头文件在 Libraries\CMSIS\Include 路径下可以找到

2.2 单精度浮点单元

基于 FPv5 的选配单精度浮点拓展单元包括一份额外的 16-入口 64 位寄存器文件。该拓展新增 45 个与 IEEE754-2008 兼容的单精度浮点指令。使用浮点指令通常可将软件库平均性能提升 10 倍。FPU 位于单独的电源域，负责在整个单元不使用的时候切断电源。

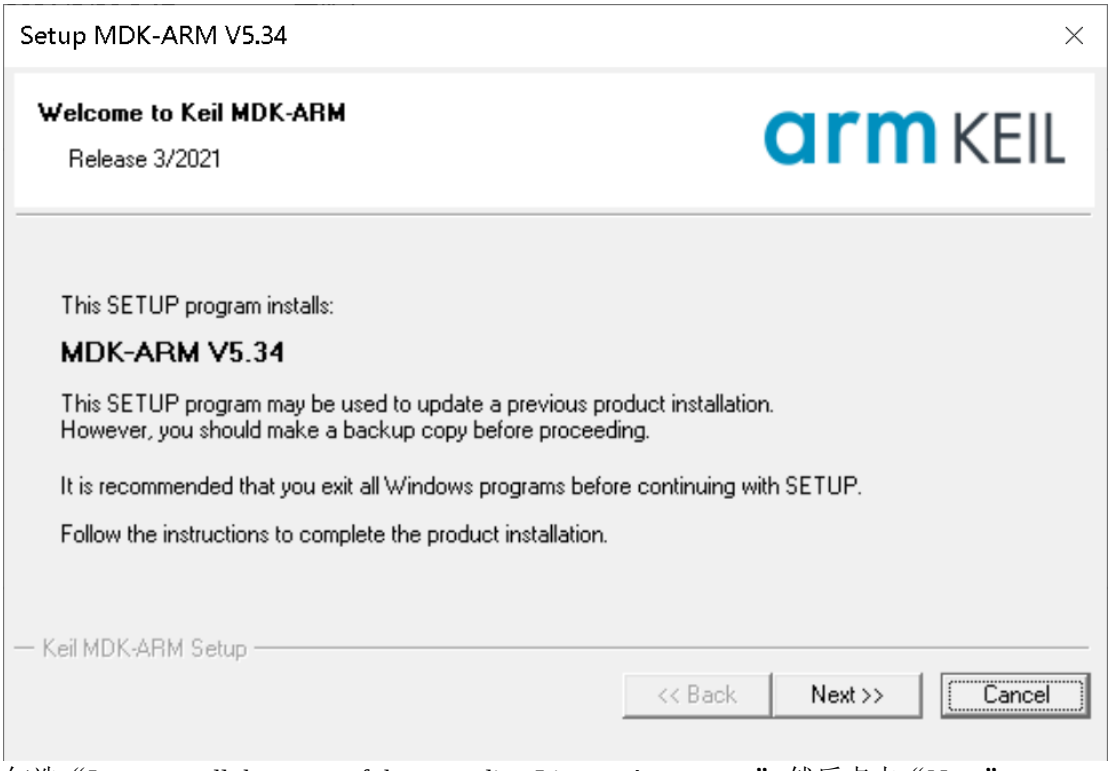
Cortex-Mstar 是一款能在性能、功耗和安全之间实现最佳平衡的处理器。

3 开发环境的建立与 ISP 下载

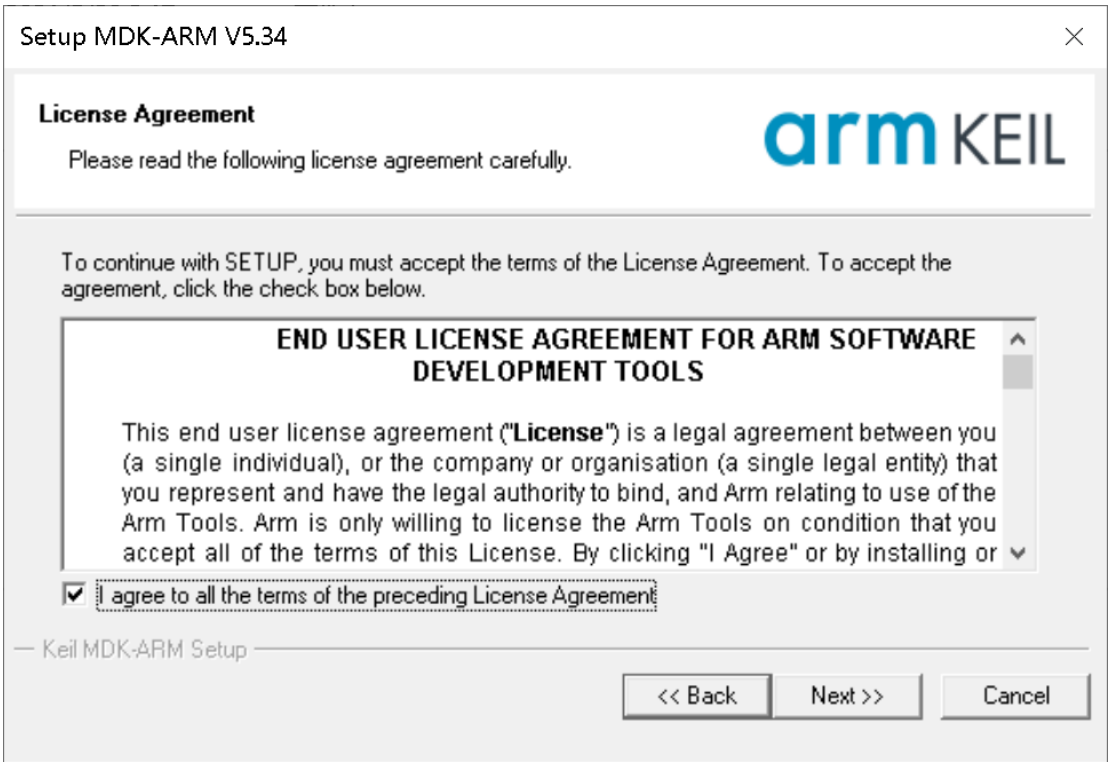
安装 Keil
首先登录 Keil 官网，下载最新版的 MDK-Arm 安装包，下载链接如下：
Keil Product Downloads



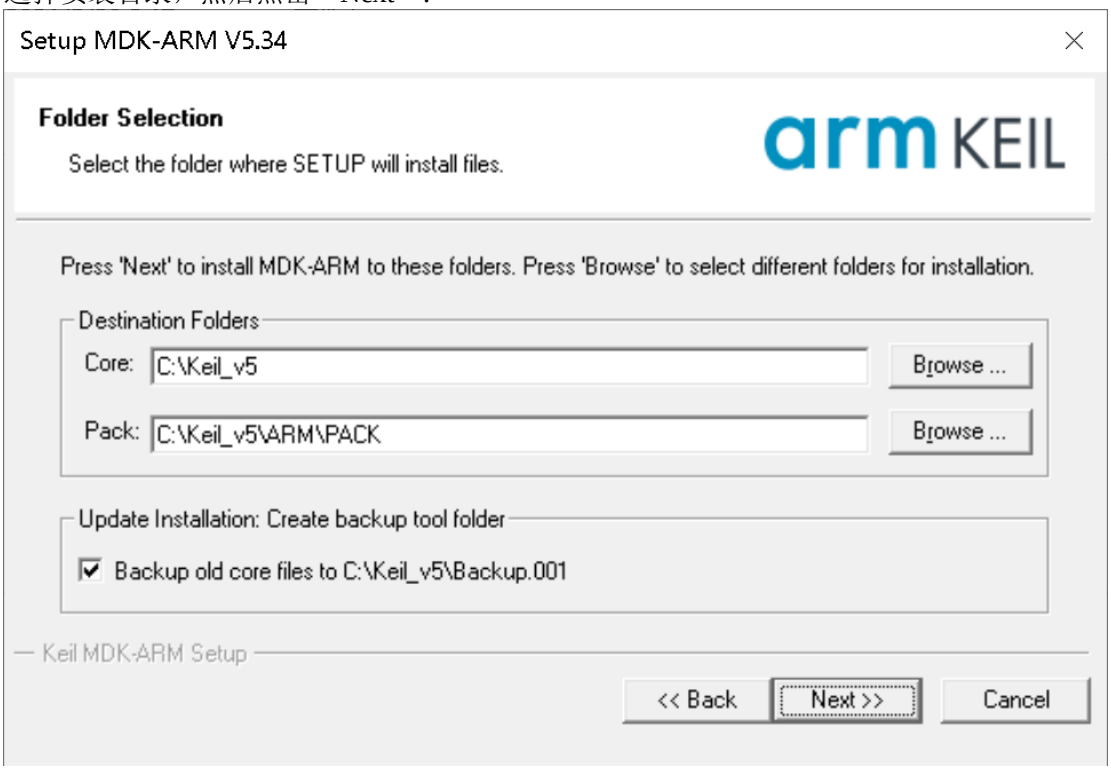
信息随便填写，点确定后进入下载页面进行下载。
双击下载的安装包开始安装，点击“Next”：



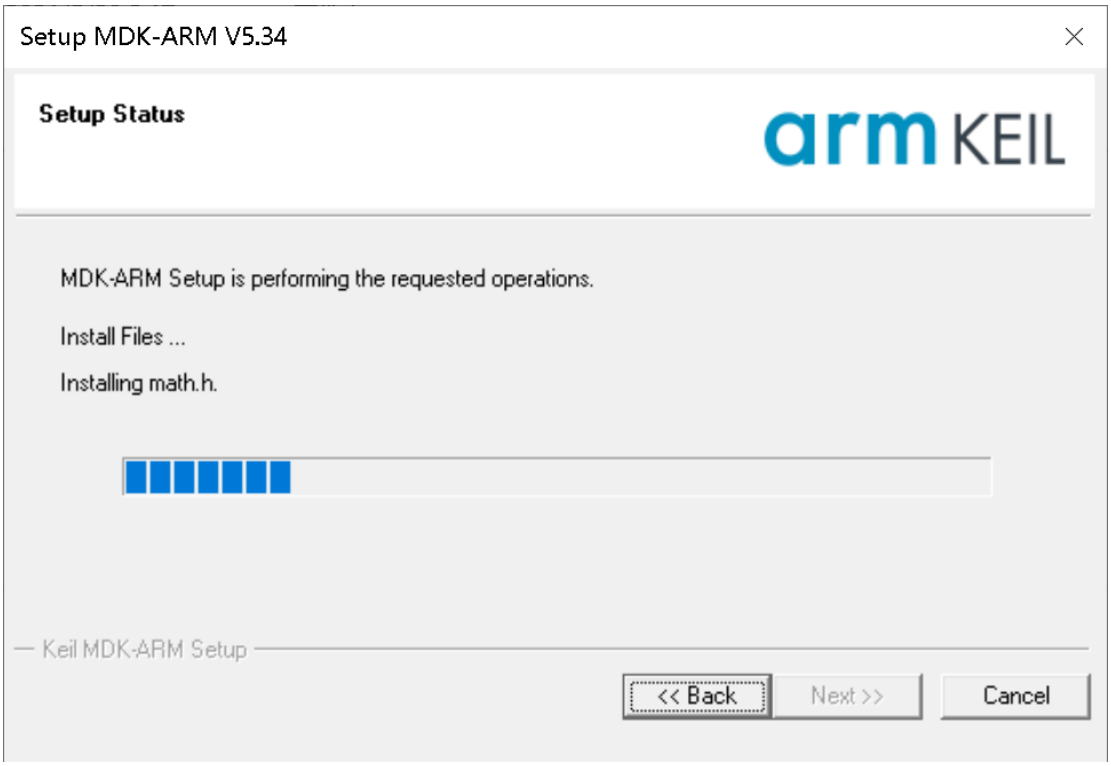
勾选“I agree to all the terms of the preceding License Agreement”,然后点击“Next”：



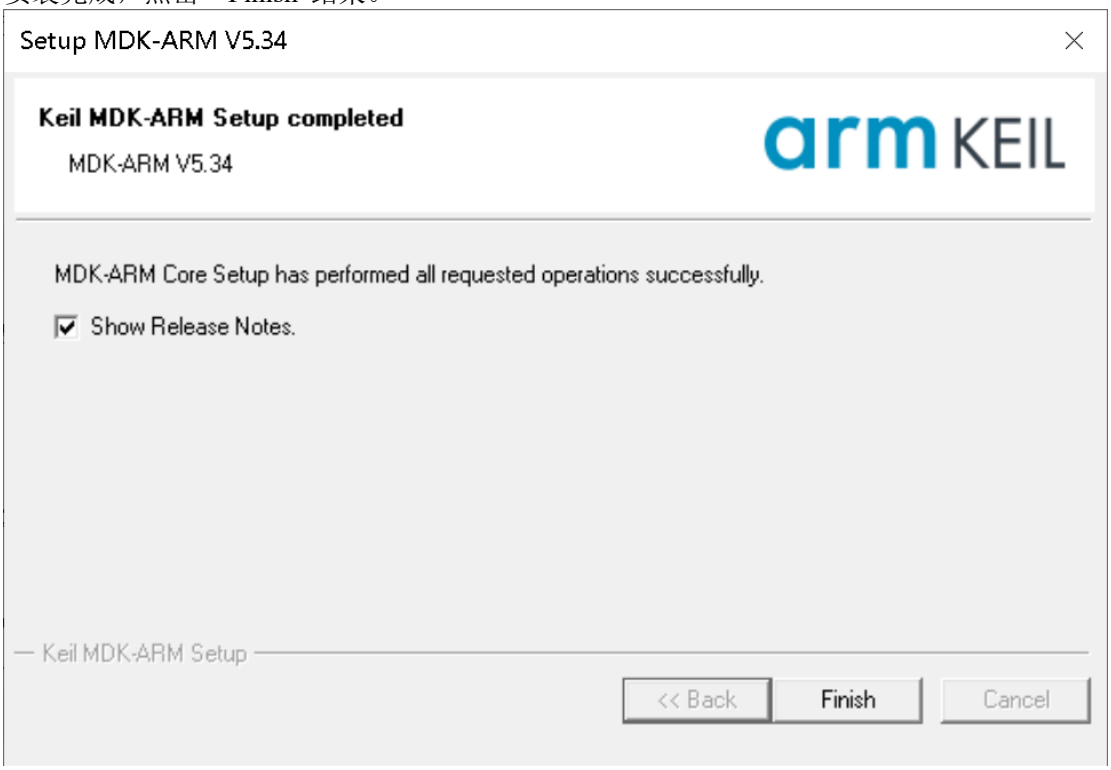
选择安装目录，然后点击“Next”：



填写个人信息，然后点击“Next”：

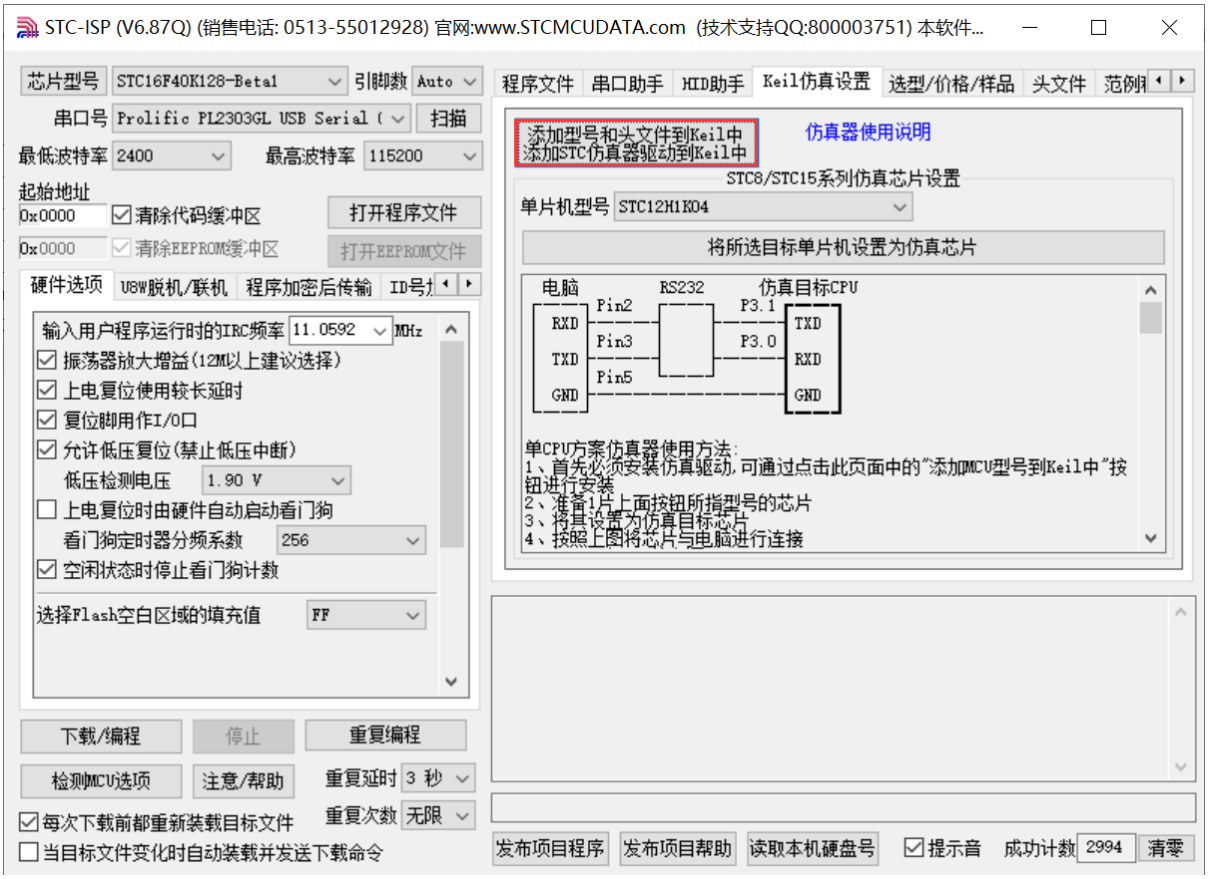


安装完成，点击“Finish”结束。

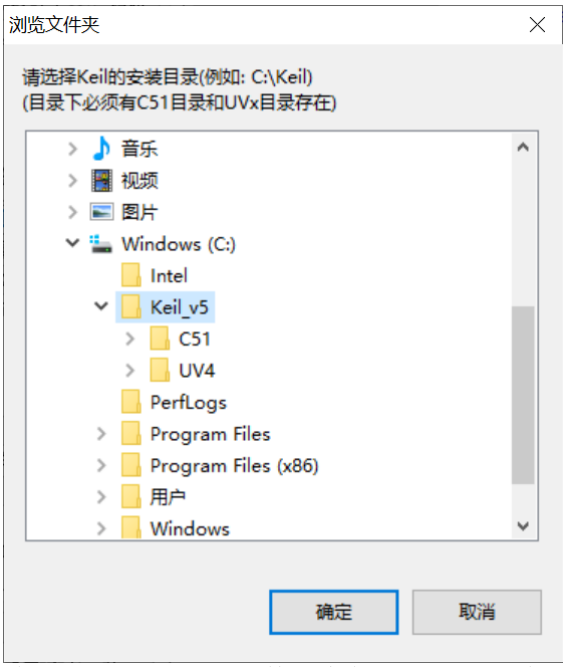


3.1 添加型号和头文件到 Keil

使用 Keil 之前需要先安装 STC 的仿真驱动。STC 的仿真驱动的安装步骤如下：
首先开 STC 的 ISP 下载软件，然后在软件右边功能区的“Keil 仿真设置”页面中点击“添加型号和头文件到 Keil 中 添加 STC 仿真器驱动到 Keil 中”按钮：



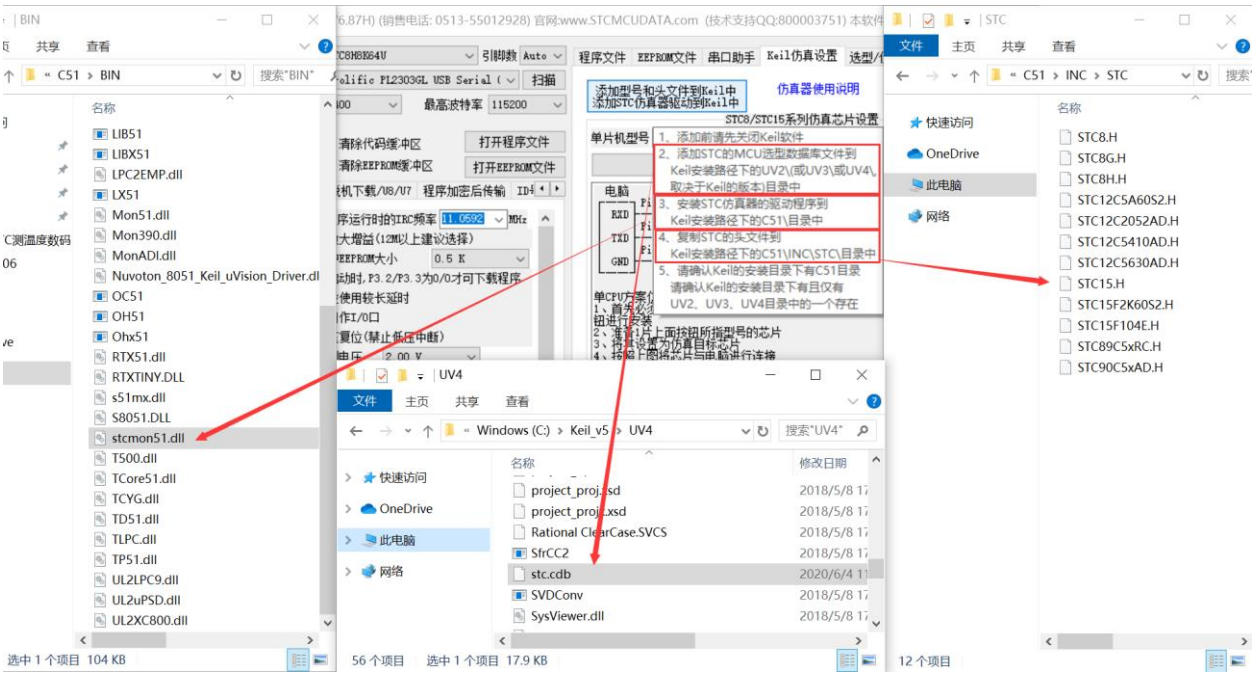
按下后会出现如下画面：



将目录定位到 Keil 软件的安装目录，然后确定。安装成功后会弹出如下的提示框：



在 Keil 的相关目录中可以看到如下的文件，即表示驱动正确安装了



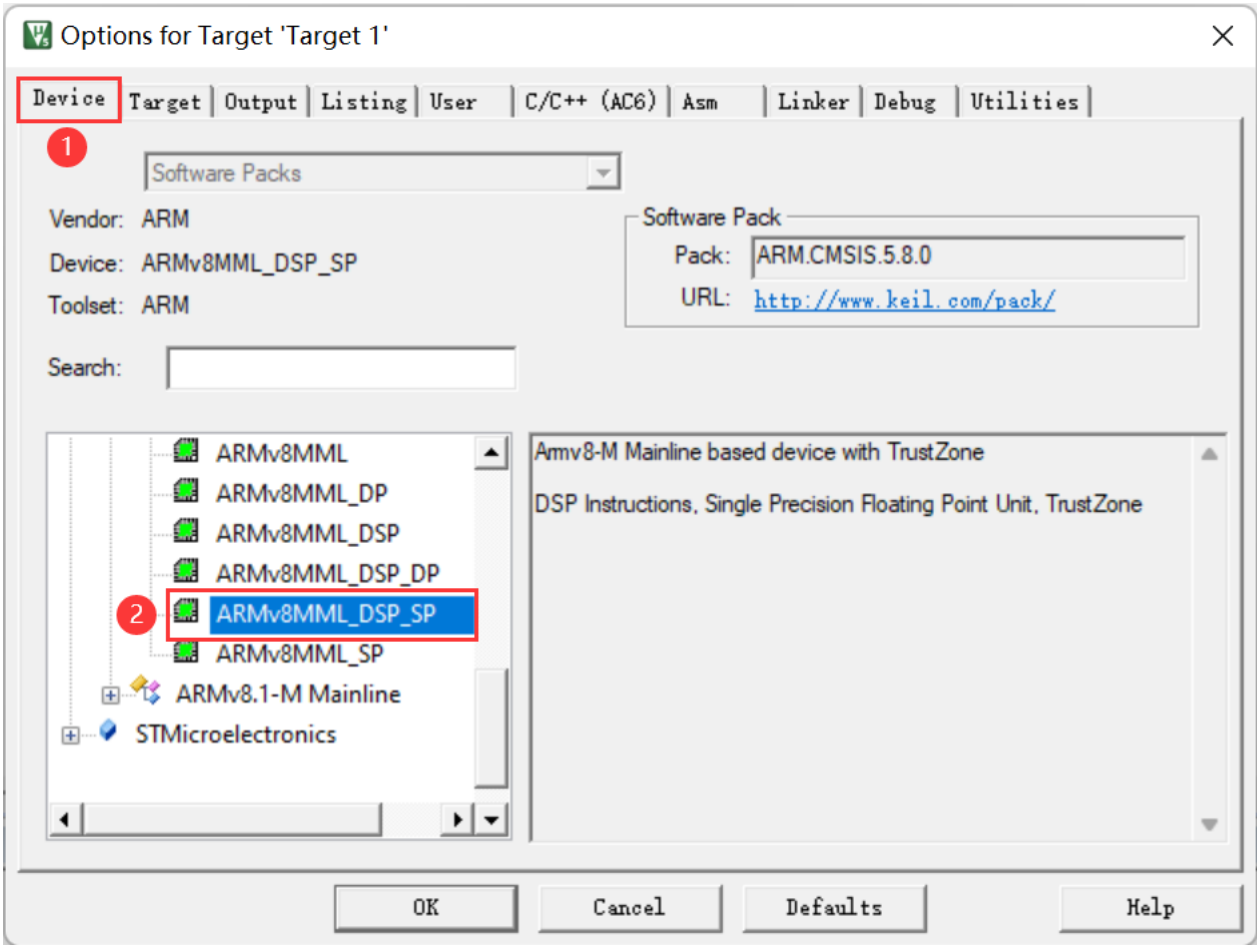
3.2 使用 KEIl 进行烧录

首先用一个 ULINK2 的仿真器连结电脑和开发板的 20PIN JTAG 接口

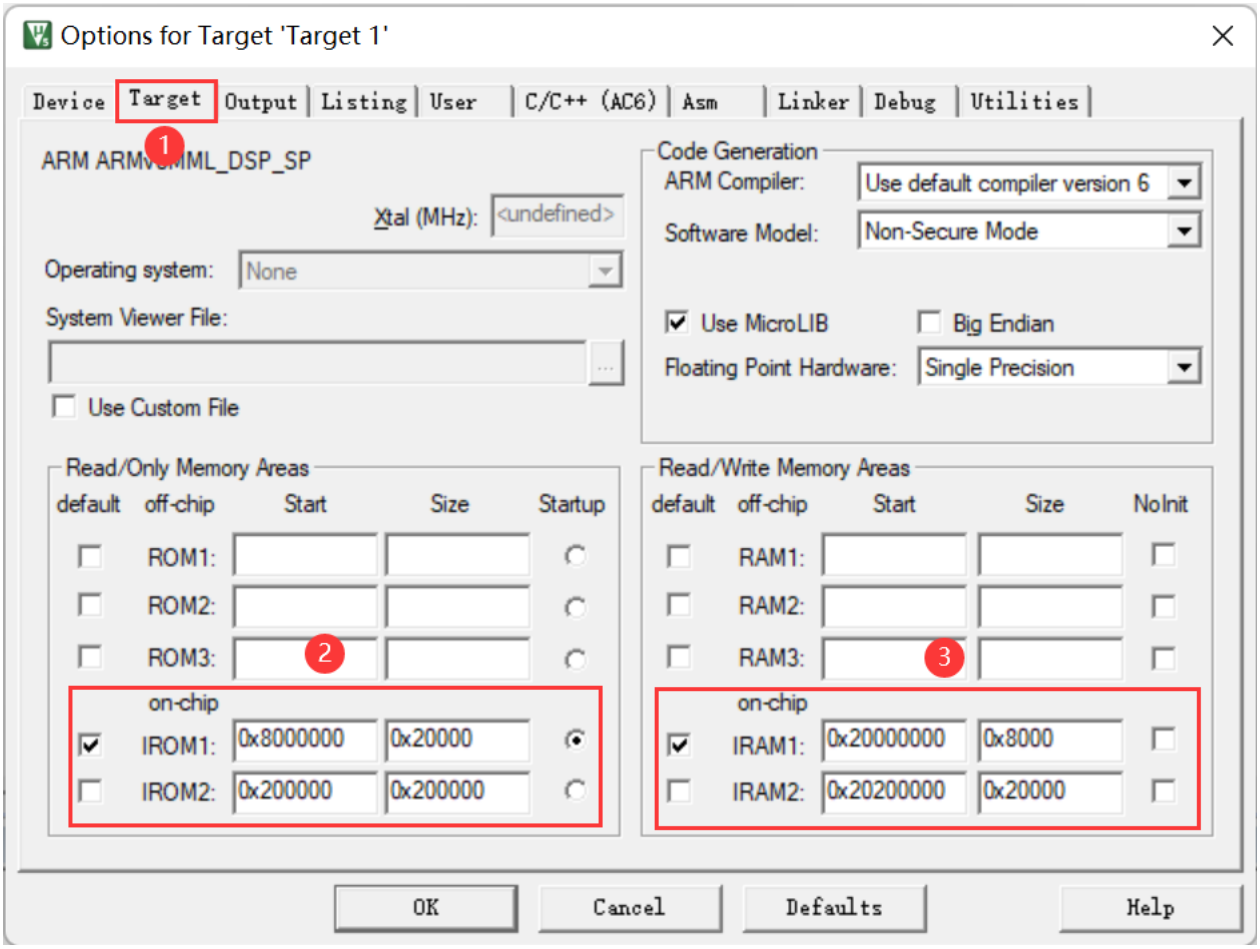


然后在 keil 中按照如下配置好

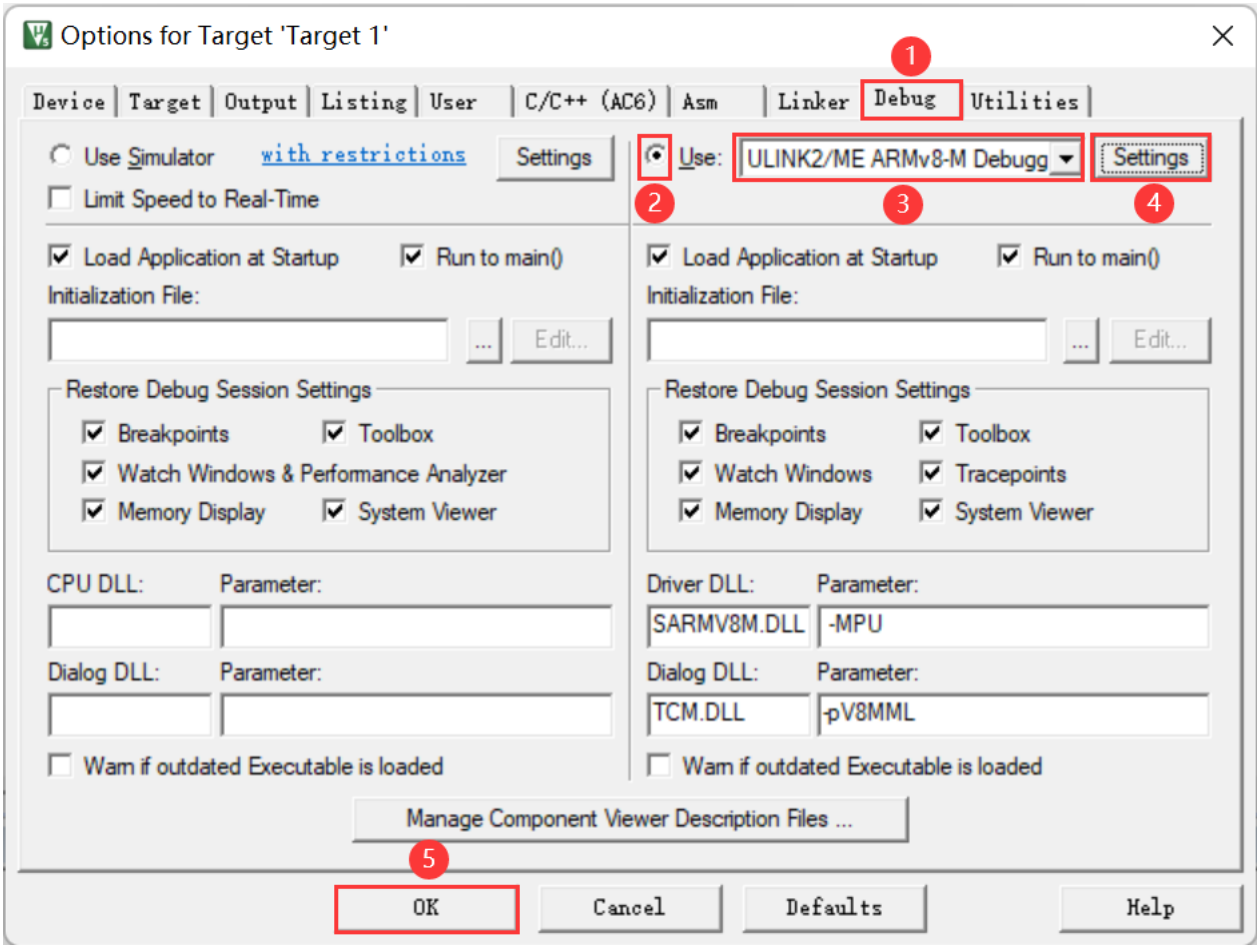
选择单片机型号为“ARMv8MML_DSP_SP”：



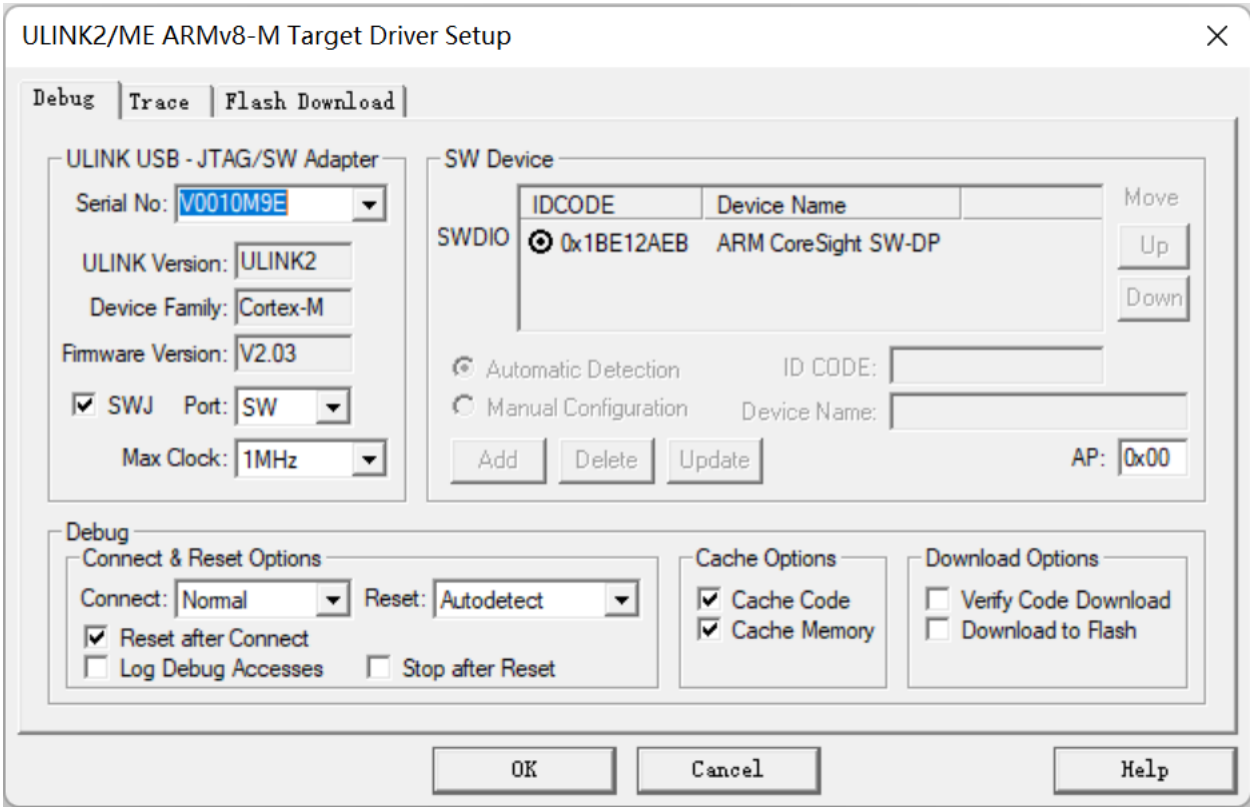
设置 ROM 与 RAM 地址空间：


Options for Target 'Target 1'
Device
Target
Output
Listing
User
C/C++ (AC6)
Asm
Linker
Debug
Utilities
1
ARM ARMv8MML_DSP_SP
Xtal (MHz):
<undefined>
Operating system:
None
System Viewer File:
Use Custom File
Code Generation
ARM Compiler:
Use default compiler version 6
Software Model:
Non-Secure Mode
Use MicroLIB
Big Endian
Floating Point Hardware:
Single Precision
Read/Only Memory Areas
default
off-chip
Start
Size
Startup
ROM1:
ROM2:
ROM3:
2
on-chip
IROM1:
0x8000000
0x20000
IROM2:
0x200000
0x200000
Read/Write Memory Areas
default
off-chip
Start
Size
NoInit
RAM1:
RAM2:
RAM3:
3
on-chip
IRAM1:
0x20000000
0x8000
IRAM2:
0x20200000
0x20000
OK
Cancel
Defaults
Help

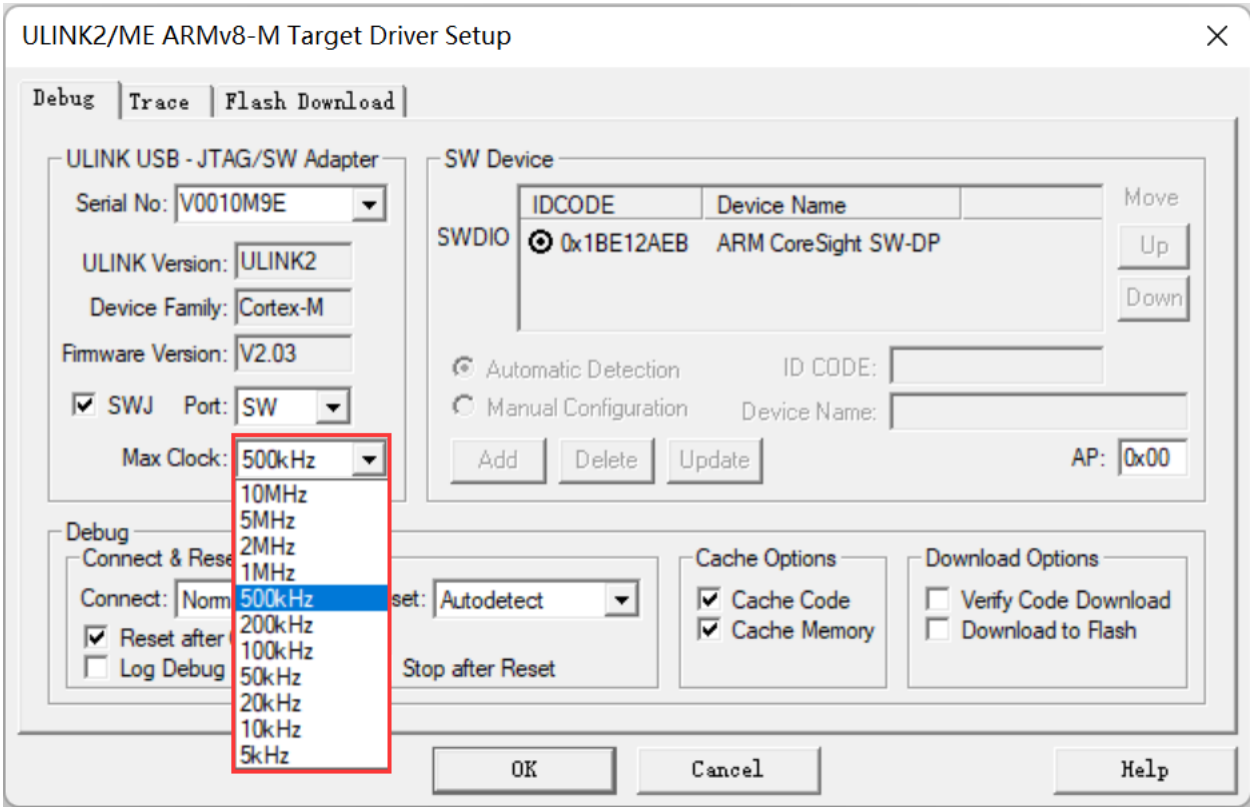
选择硬件仿真工具：



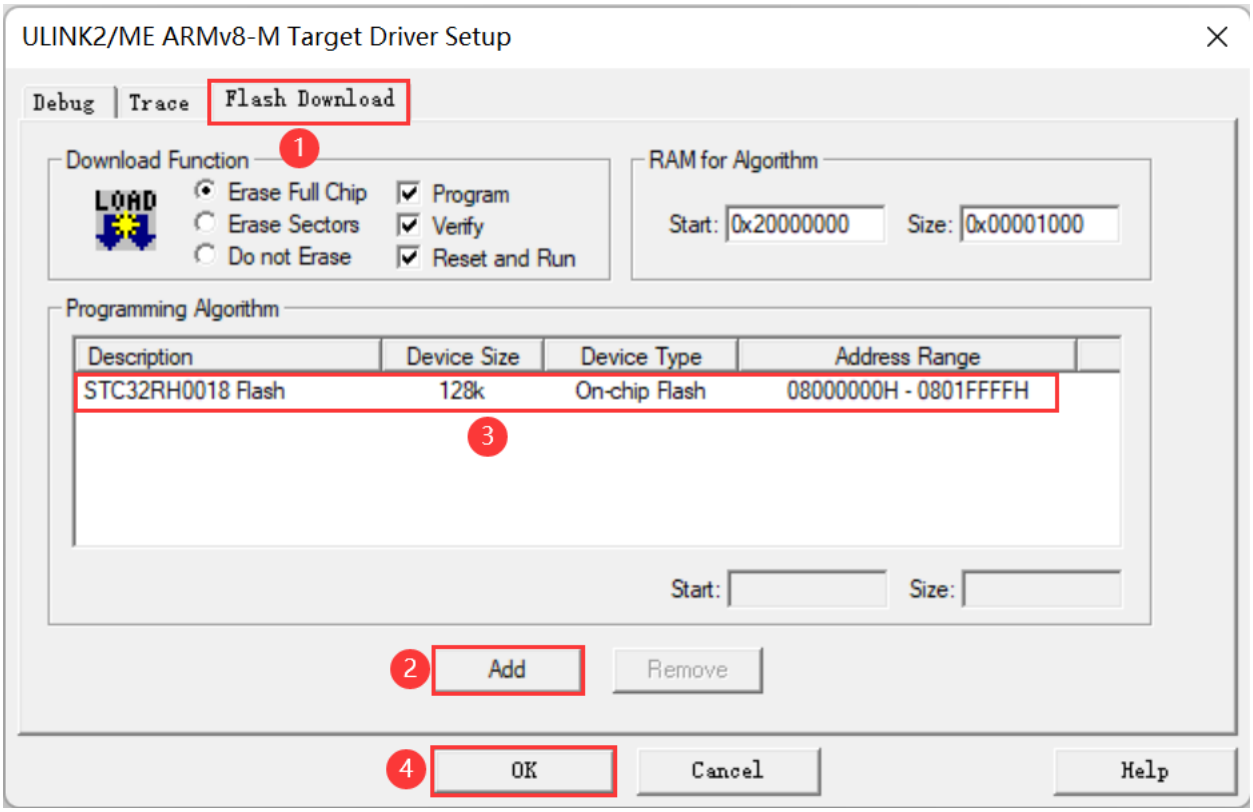
点击“Settings”按钮设置仿真工具：



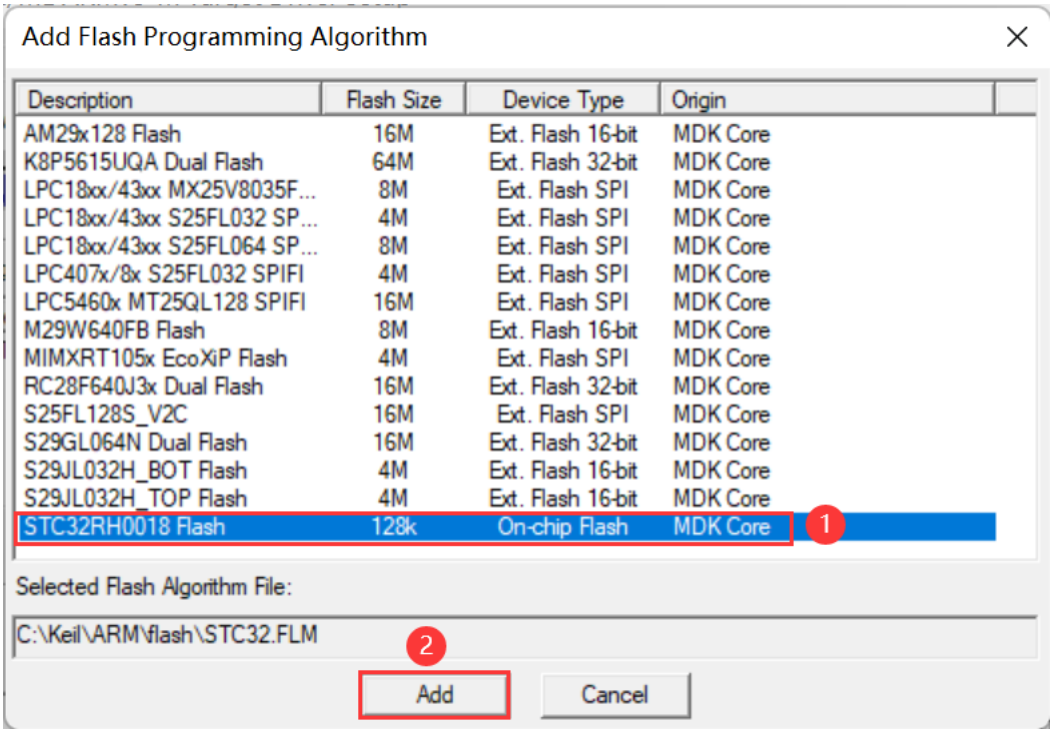
如果下载/仿真时出现异常，可将通讯的时钟频率降低一些：



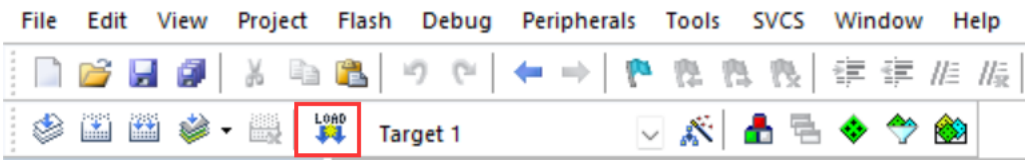
选择对应芯片的 Flash：



如果 3 对应的列表里面没有“STC32RH0018 Flash”的话，通过“Add”按钮添加：

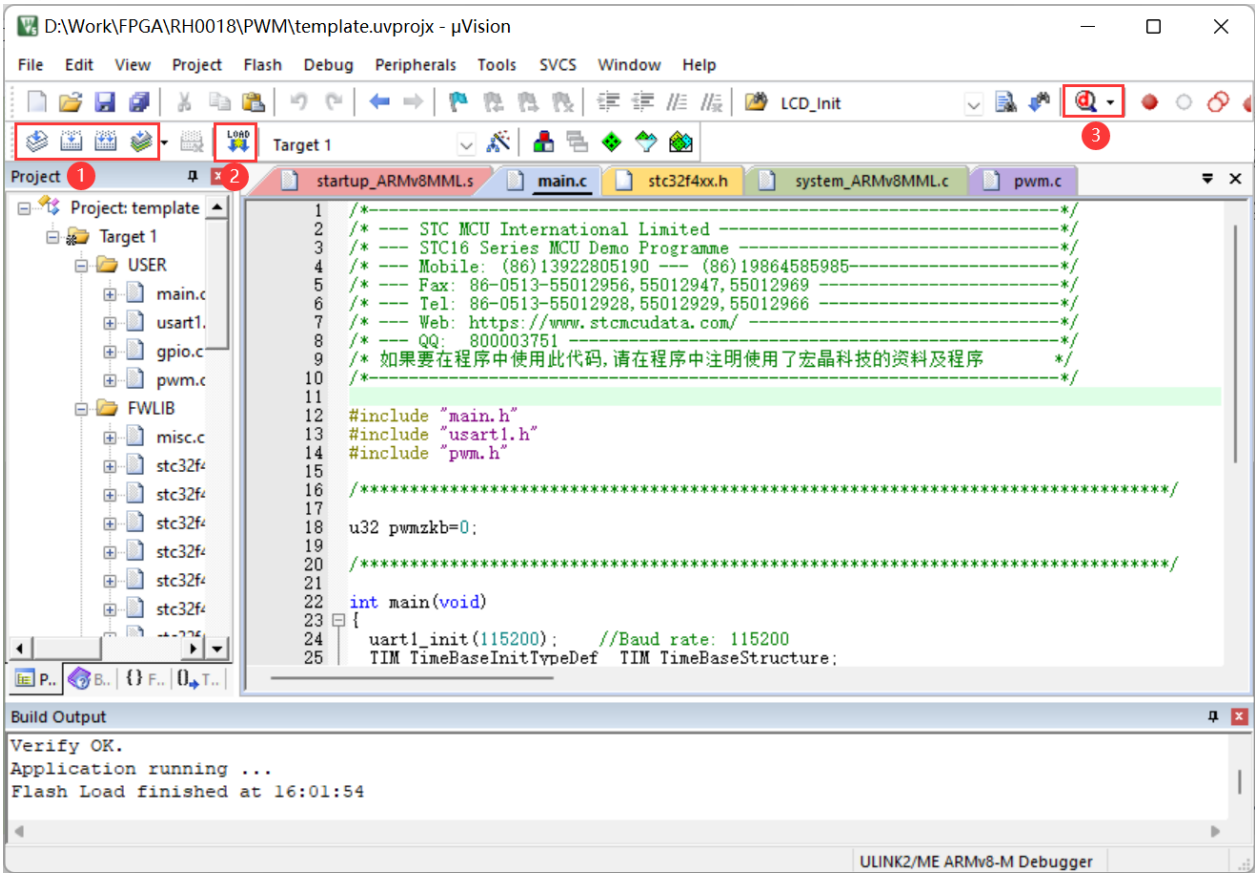


最后项目编译完，在 keil 中点 download 就可以进行下载烧录了。

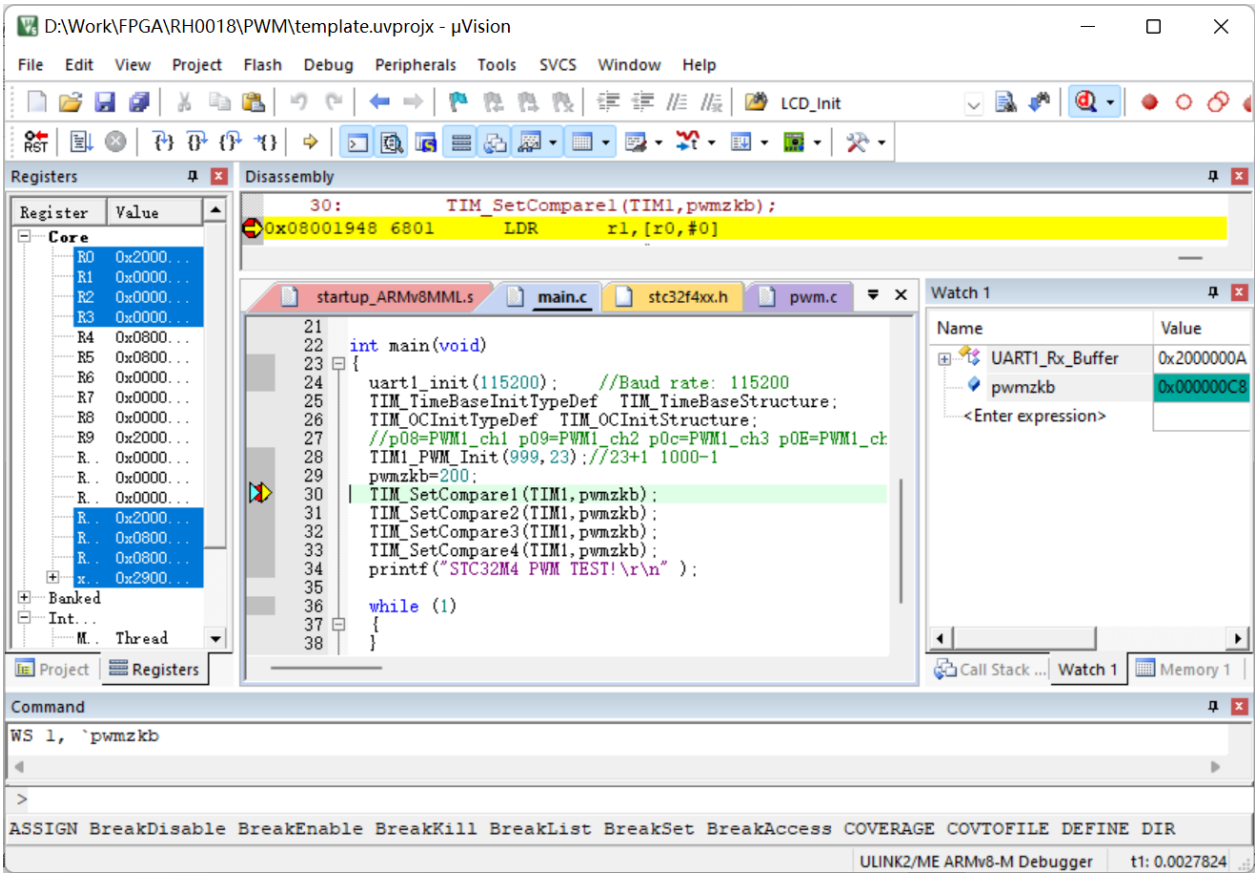


3.3 使用 keil 进行仿真调试

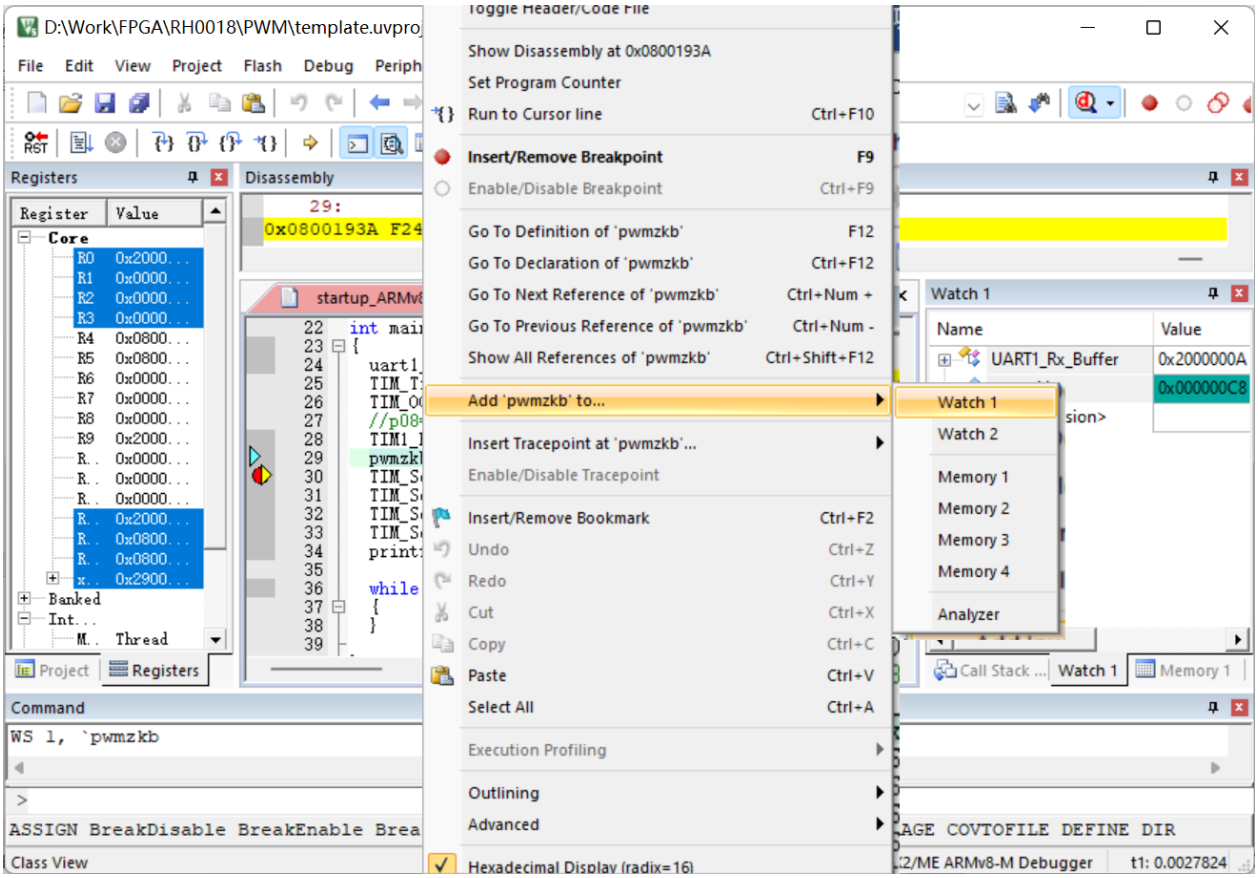
首先按照上面的步骤连接上开发板，打开一个工程文件，编译下载后点击 keil 上的 debug 按钮进入调试模式：



可使用加断点、单步执行等调试功能：



在变量名称上鼠标右键弹出菜单可以添加变量到 Watch 1 可以随时察看变量的值：


D:\Work\FPGA\RH0018\PWM\template.uvproj
File Edit View Project Flash Debug Periph
Registers
Register Value
Core
R0 0x2000...
R1 0x0000...
R2 0x0000...
R3 0x0000...
R4 0x0800...
R5 0x0800...
R6 0x0000...
R7 0x0000...
R8 0x0000...
R9 0x2000...
Banked
Int...
Thread
Project
Registers
Disassembly
29:
0x0800193A F24
startup_ARMv8
22 int mai
23 {
24 uart1_
25 TIM_T
26 TIM_O
27 //P08
28 TIM1_I
29 pwmzkb
30 TIM_S
31 TIM_S
32 TIM_S
33 TIM_S
34 print
35
36 while
37 {
38 }
39
Command
WS 1, `pwmzkb
ASSIGN BreakDisable BreakEnable Brea
Class View
Toggle Header/Code File
Show Disassembly at 0x0800193A
Set Program Counter
Run to Cursor line Ctrl+F10
Insert/Remove Breakpoint F9
Enable/Disable Breakpoint Ctrl+F9
Go To Definition of 'pwmzkb' F12
Go To Declaration of 'pwmzkb' Ctrl+F12
Go To Next Reference of 'pwmzkb' Ctrl+Num +
Go To Previous Reference of 'pwmzkb' Ctrl+Num -
Show All References of 'pwmzkb' Ctrl+Shift+F12
Add 'pwmzkb' to...
Insert Tracepoint at 'pwmzkb'...
Enable/Disable Tracepoint
Insert/Remove Bookmark Ctrl+F2
Undo Ctrl+Z
Redo Ctrl+Y
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Select All Ctrl+A
Execution Profiling
Outlining
Advanced
Hexadecimal Display (radix=16)
Watch 1
Name Value
UART1_Rx_Buffer 0x2000000A
0x000000C8
sion>
Watch 1
Watch 2
Memory 1
Memory 2
Memory 3
Memory 4
Analyzer
Call Stack ...
Watch 1
Memory 1
AGE COVTOFILE DEFINE DIR
:2/ME ARMv8-M Debugger
t1: 0.0027824

4 嵌入式 Flash 接口

4.1 前言

Flash 接口可管理 CPU 通过 AHB I-Code 和 D-Code 对 Flash 进行的访问。该接口可针对 Flash 执行擦除和编程操作，并实施读写保护机制。

Flash 接口通过指令预取和缓存机制加速代码执行。

4.2 主要特性

- Flash 读操作
- Flash 编程/擦除操作
- 读/写保护
- I-Code 上的预取操作
- D-Code 上的 预取操作

4.3 嵌入式 Flash

Flash 具有以下主要特性：

- 对于 STC33F，容量高达 512K；
- 128 位宽数据读取
- 字节、半字、字和双字数据写入
- 扇区擦除与全部擦除
- 存储器组织结构

4.4 读接口

4.4.1 CPU 时钟频率与 Flash 读取时间之间的关系

为了准确读取 Flash 数据，必须根据 CPU 时钟 (HCLK) 频率和器件电源电压在 Flash 存取控制寄存器 (FLASH_ACR) 中正确地编程等待周期数 (LATENCY)。

当电源电压低于 2.1 V 时，必须关闭预取缓冲器。 表 7 所示为 Flash 等待周期与 CPU 时钟频率之间的对应关系。

表 7. CPU 时钟 (HCLK) 频率对应的等待周期数

等待周期（WS）(LATENCY)	HCLK(MHz)			
	电压范围 2.7V-3.6V	电压范围 2.4V-2.7V	电压范围 2.1V-2.4V	电压范围 1.8V-2.1V 预取关闭
0WS (1 个 CPU 周期)	0<HCLK≤30	0<HCLK≤24	0<HCLK≤22	0<HCLK≤20
1WS (2 个 CPU 周期)	30<HCLK≤60	24<HCLK≤48	22<HCLK≤44	20<HCLK≤40
2WS (3 个 CPU 周期)	60<HCLK≤90	48<HCLK≤72	44<HCLK≤66	40<HCLK≤60
3WS (4 个 CPU 周期)	90<HCLK≤120	72<HCLK≤96	66<HCLK≤88	60<HCLK≤80
4WS (5 个 CPU 周期)	120<HCLK≤150	96<HCLK≤120	88<HCLK≤110	80<HCLK≤100
5WS (6 个 CPU 周期)	150<HCLK≤168	120<HCLK≤144	110<HCLK≤132	100<HCLK≤120
6WS (7 个 CPU 周期)		144<HCLK≤168	132<HCLK≤154	120<HCLK≤140
7WS (8 个 CPU 周期)			154<HCLK≤168	140<HCLK≤160

复位后，CPU 时钟频率为 72 MHz，FLASH_ACR 寄存器中的等待周期 (WS) 为 0。
强烈建议通过以下软件序列来根据 CPU 频率调整在访问 Flash 时所需的等待周期数。
需要提高 CPU 频率时的操作步骤

1. 将新的等待周期数编程到 FLASH_ACR 寄存器中的 LATENCY 位
2. 通过读取 FLASH_ACR 寄存器，检查新的等待周期是否设置成功
3. 通过改写 RCC_CFGR 寄存器中的 SW 位来修改 CPU 时钟源
4. 如有需要，可通过改写 RCC_CFGR 中的 HPRE 位来修改 CPU 时钟预分频器
5. 通过读取 RCC_CFGR 寄存器中相应的时钟源状态（SWS 位）和/或 AHB 预分频值（HPRE 位），检查新的 CPU 时钟源和/或新的 CPU 时钟预分频值是否设置成功

需要降低 CPU 频率时的操作步骤

1. 通过改写 RCC_CFGR 寄存器中的 SW 位来修改 CPU 时钟源
2. 如有需要，可通过改写 RCC_CFGR 中的 HPRE 位来修改 CPU 时钟预分频器
3. 通过读取 RCC_CFGR 寄存器中相应的时钟源状态（SWS 位）和/或 AHB 预分频值（HPRE 位），检查新的 CPU 时钟源和/或新的 CPU 时钟预分频值是否设置成功
4. 将新的等待周期数编程到 FLASH_ACR 中的 LATENCY 位
5. 通过读取 FLASH_ACR 寄存器，检查新的等待周期是否设置成功

注意： CPU 时钟配置或等待周期 (WS) 配置的更改不会立即生效。为了确保当前的 CPU 时钟频率即为所配置的频率，用户可检查 AHB 预分频系数和时钟源状态值。为了确保所编程的 WS 数生效，可读取 FLASH_ACR 寄存器的内容来确认。

4.4.2 缓存存储器加速器

为了减少因指令跳转而产生的时间损耗，可将 64 行 128 位的指令保存到指令缓存存储器中。可将 FLASH_ACR 寄存器中的指令缓存使能 (ICEN) 位置 1，来使能这一特性。每当出现指令缺失（即请求的指令未存在于当前使用的指令行、预取指令行或指令缓存存储器中）时，系统会将新读取的行复制到指令缓存存储器中。如果 CPU 请求的指令已存在于指令缓存区中，则无需任何延时即可立即获取。指令缓存存储器存满后，可采用 LRU（最近最少使用）策略确定指令缓存存储器中待替换的指令行。此特性非常适用于包含循环的代码。

数据管理在 CPU 流水线执行阶段，将通过 D-Code 总线访问 Flash 中的数据缓冲池。因此，直到提供了请求的数据后，CPU 流水线才会继续执行。为了减少因此而产生的时间损耗，通过 AHB 数据总线 D-Code 进行的访问优先于通过 AHB 指令总线 I-Code 进行的访问。

如果频繁使用某些数据，可将 FLASH_ACR 寄存器中的数据缓存使能 (DCEN) 位置 1，来使能数据缓存存储器。此特性的工作原理与指令缓存存储器类似，但保留的数据大小限制在 8 行 128 位/行以内。

注意： 用户配置扇区中的数据无法缓存。

4.5 擦除和编程操作

执行任何 Flash 编程操作（擦除或编程）时，CPU 时钟频率 (HCLK) 不能低于 1 MHz。如果在 Flash 操作期间发生器件复位，无法保证 Flash 中的内容。

在对 Flash 执行写入或擦除操作期间，任何读取 Flash 的尝试都会导致总线阻塞。只有在完成编程操作后，才能正确处理读操作。这意味着，写/擦除操作进行期间不能从 Flash 中执行代码或数据获取操作。

4.5.1 Flash 控制寄存器解锁

复位后，Flash 控制寄存器 (FLASH_CR) 不允许执行写操作，以防因电气干扰等原因出现对 Flash 的意外操作。此寄存器的解锁顺序如下：

1. 在 Flash 密钥寄存器 (FLASH_KEYR) 中写入 KEY1 = 0x45670123
2. 在 Flash 密钥寄存器 (FLASH_KEYR) 中写入 KEY2 = 0xCDEF89AB

如果顺序出现错误，将返回总线错误并锁定 FLASH_CR 寄存器，直到下一次复位。

也可通过软件将 FLASH_CR 寄存器中的 LOCK 位置为 1 来锁定 FLASH_CR 寄存器。

注意： 当 FLASH_SR 寄存器中的 BSY 位为 1 时，将不能在写模式下访问 FLASH_CR 寄存器。BSY 位为 1 时，对该寄存器的任何写操作尝试都会导致 AHB 总线阻塞，直到 BSY 位清零。

4.5.2 编程/ 擦除并行位数

通过 FLASH_CR 寄存器中的 PSIZE 字段配置并行位数。并行位数表示每次对 Flash 进行写操作时将编程的字节数。PSIZE 受限于电源电压以及是否使用外部 V PP 电源。因此，在进行任何编程/擦除操作前，必须在 FLASH_CR 寄存器中对其进行正确配置。

Flash 擦除操作只能针对扇区或整个 Flash（批量擦除）执行。擦除时间取决于 PSIZE 编程值。表 8 提供了正确的 PSIZE 值。

表 8. 编程/擦除并行位数

	电压范围 2.7--3.6V （使用外部 Vpp）	电压范围 2.7--3.6V	电压范围 2.4--2.7V	电压范围 2.1--2.4V	电压范围 1.8--2.1V
并行位数	x64	x32	x16		x8
PSIZE(1:0)	11	10	01		00

注意： 如果在编程并行位数 / 电压范围设置不一致的情况下启动任何编程或擦除操作，可能会导致出现意外结果。即使后续的读操作指示逻辑值已有效写入存储器中，也无法确定写入操作确实成功。

要使用 V PP ，必须在 V PP 引脚施加一个外部高压电源（ 8 V 到 9 V 之间）。该外部电源必须在直流电耗超过 10 mA 时也能维持该电压范围。建议仅在工厂生产线上进行初始编程时使用 VPP 。 VPP 电源的供电时间不得超过一小时，否则 Flash 可能会损坏。

4.5.3 擦除

Flash 擦除操作可针对扇区或整个 Flash（批量擦除）执行。执行批量擦除时，不会影响配置扇区。

扇区擦除

扇区擦除的具体步骤如下：

1. 检查 FLASH_SR 寄存器中的 BSY 位，以确认当前未执行任何 Flash 操作
2. 在 FLASH_CR 寄存器中，将 SER 位置 1，并从主存储块的 12 个 扇区中选择要擦除的扇区 (SNB)
3. 将 FLASH_CR 寄存器中的 STRT 位置 1
4. 等待 BSY 位清零

批量擦除

要执行批量擦除，建议采用以下步骤：

1. 检查 FLASH_SR 寄存器中的 BSY 位，以确认当前未执行任何 Flash 操作
2. 将 FLASH_CR 寄存器中的 MER 位置 1
3. 将 FLASH_CR 寄存器中的 STRT 位置 1
4. 等待 BSY 位清零

注意： 如果 FLASH_CR 寄存器中的 MERx 位和 SER 位均置为 1 ，则无法执行扇区擦除和批量擦除。

4.5.4 编程

标准编程

Flash 编程顺序如下：
1. 检查 FLASH_SR 中的 BSY 位，以确认当前未执行任何主要 Flash 操作。
2. 将 FLASH_CR 寄存器中的 PG 位置 1。
3. 针对所需存储器地址执行数据写入操作：
— 并行位数为 x8 时按字节写入
— 并行位数为 x16 时按半字写入
— 并行位数为 x32 时按字写入
— 并行位数为 x64 时按双字写入
4. 等待 BSY 位清零。
注意：把 Flash 的单元从“ 1 ”写为“ 0 ”时，无需执行擦除操作即可进行连续写操作。把 Flash 的单元从“ 0 ”写为“ 1 ”时，则需要执行 Flash 擦除操作。
如果同时发出擦除和编程操作请求，首先执行擦除操作。

编程错误

不允许针对 Flash 执行跨越 128 位行界限的数据编程操作。如果出现这种情况，写操作将不会执行，并且 FLASH_SR 寄存器中的编程对齐错误标志位 (PGAERR) 将置 1。
写访问宽度（字节、半字、字或双字）必须与所选并行位数类型（x8、x16、x32 或 x64）相符。否则，写操作将不会执行，并且 FLASH_SR 寄存器中的编程并行位数错误标志位(PGPERR) 将置 1。
如果未遵循标准的编程顺序（例如，在 PG 位未置 1 时尝试向 Flash 地址写入数据），则操作将中止并且 FLASH_SR 寄存器中的编程顺序错误标志位 (PGSERR) 将置 1。

编程与缓存

如果 Flash 写访问涉及数据缓存中的某些数据，Flash 写访问将修改 Flash 中的数据和缓存中的数据。
如果 Flash 中的擦除操作也涉及数据或指令缓存中的数据，则必须确保在代码执行期间访问这些数据之前将它们重新写入缓存。如果无法可靠执行这一操作，建议将 FLASH_CR 寄存器中的 DCRST 和 ICRST 位置 1，以刷新缓存。
注意： I/D 缓存只有在被禁止 (I/DCEN = 0) 的情况下才能刷新。

4.5.5 中断

如果将 FLASH_CR 寄存器中的操作结束中断使能位 (EOPIE) 置 1，则在擦除或编程操作结束时，即 FLASH_SR 寄存器中的繁忙位 (BSY) 清零（操作正确或非正确完成）时，将产生中断。此时，FLASH_SR 寄存器中的操作结束 (EOP) 位置 1。
如果在请求编程、擦除或读操作期间出现错误，则 FLASH_SR 寄存器中的以下错误标志位之一将置 1：
- PGAERR、PGPERR、PGSERR（编程错误标志）
- WRPERR（保护错误标志）

这种情况下，FLASH_SR 寄存器中的操作错误位 (OPERR) 置 1，并且如果 FLASH_SR 寄存器中的错误中断使能位 (ERRIE) 置 1，则将产生一个中断。
注意： 如果检测到多个连续错误（例如，在对 Flash 进行 DMA 传输期间），则直到连续写操作请求结束，这些错误标志才会清零。

表 9. Flash 中断请求

中断事件	事件标志	使能控制位

操作结束	EOP	EOPIE
写保护错误	WRPERR	ERRIE
编程错误	PGAERR/ PGPERR/ PGSERR	ERRIE

4.6 选项字节

4.6.1 关于用户选项字节的说明

选项字节由最终用户根据具体的应用要求进行配置。 表 10 介绍了这些字节在用户配置扇区内的构成。

表 10. 选项字节构成

地址	[63:16]	[15:0]
0x1FFF C000	保留	ROP 和用户选项字节（RDP&USER）
0x1FFF C008	保留	扇区 0 到 11 的写保护 nWRP 位
0x1FFE C000	保留	保留
0x1FFE C008	保留	扇区 12 到 23 的写保护 nWRP 位

表 11. 关于选项字节的说明

选项字节（字，地址 0x1FFF C000）	
RDP:读保护选项字节。 读保护用于保护 Flash 中存储的软件代码。	
位 15:8	0xAA: 级别 0 无保护 0xCC： 级别 2 芯片保护（禁止调试和从 RAM 启动） 其他值：级别 1，存储器读保护（调试功能受限）
USER:用户选项字节 此字节用于配置以下功能： --选择看门狗事件：硬件或软件 --进入停止模式时产生复位事件 --进入待机模式时产生复位事件	
位 7	nRST_STDBY 0: 进入待机模式时产生复位 1：不产生复位

表 11. 关于选项字节的说明（续）

位 6	**nRST_STOP** 0: 进入停止模式时产生复位 1：不产生复位
位 5	**WDG_SW** 0: 硬件独立看门狗 1：软件独立看门狗
位 4	**0x1：未使用**

位 3:2	**BOR_LEV:**BOR 复位级别 这些位包含释放复位信号所需达到的供电电压阈值。通过对这些位执行写操作，可将新的 BOR 级别编程到 Flash。 00：BOR 级别 3（VBOR3）,复位阈值电压为 2.70V 到 3.60V 01：BOR 级别 2（VBOR2）,复位阈值电压为 2.40V 到 2.70V 10：BOR 级别 1（VBOR1）,复位阈值电压为 2.10V 到 2.40V 11：BOR 关闭（VBOR0）,复位阈值电压为 1.80V 到 2.10V
位 1:0	**0x1：未使用**
选项字节（字，地址 0x1FFF C008）	
位 15:12	0xF：未使用
nWRP: Flash 写保护选项字节 扇区 0 到 11 可采用写保护	
位 11:0	**nWRPi** 0:开启所选扇区的写保护 1：关闭所选扇区的写保护

4.6.2 用户选项字节编程

要针对此扇区执行任何操作，Flash 选项控制寄存器 (FLASH_OPTCR) 中的选项锁定位 (OPTLOCK) 必须清零。为了能够将该位清零，用户需要顺序执行以下步骤：

1. 在 Flash 选项密钥寄存器 (FLASH_OPTKEYR) 中写入 OPTKEY1 = 0x0819 2A3B
2. 在 Flash 选项密钥寄存器 (FLASH_OPTKEYR) 中写入 OPTKEY2 = 0x4C5D 6E7F 通过软件将 OPTLOCK 位置 1 后，可防止用户选项字节发生意外的擦除/编程操作。

修改用户选项字节

要修改用户选项值，请顺序执行以下步骤：

1. 检查 FLASH_SR 寄存器中的 BSY 位，以确认当前未执行任何 Flash 操作
2. 在 FLASH_OPTCR 寄存器中写入所需的选项值。
3. 将 FLASH_OPTCR 寄存器中的选项启动位 (OPTSTRT) 置 1。
4. 等待 BSY 位清零。

注意： 硬件会自动先擦除用户配置扇区，然后以 FLASH_OPTCR 寄存器中包含的值对所有选项字节进行编程。

4.6.3 读保护 (RDP)

可对 Flash 中的用户区域实施读保护，以防不受信任的代码读取其中的数据。读保护分三个级别，具体定义如下：

- 级别 0 ：无读保护

将 0xAA 写入读保护选项字节 (RDP) 时，读保护级别即设为 0。此时，在所有自举配置（用户 Flash 自举、调试或从 RAM 自举）中，均可执行对 Flash 或备份 SRAM 的读/写操作（如果未设置写保护）。

- 级别 1 ：使能读保护

这是擦除选项字节后的默认读保护级别。将任意值（分别用于设置级别 0 和级别 2 的 0xAA 和

0xCC 除外）写入 RDP 选项字节时，即激活读保护级别 1。设置读保护级别 1 后：
— 在连接调试功能或从 RAM 或系统存储器自举时，不能对 Flash 或备份 SRAM 进行访问（读取、擦除、编程）。读请求将导致总线错误。
— 从 Flash 自举时，允许通过用户代码对 Flash 和备份 SRAM 进行访问（读取、擦除、编程）。
激活级别 1 后，如果将保护选项字节 (RDP) 编程为级别 0，则将对 Flash 和备份 SRAM 执行全部擦除。因此，在取消读保护之前，用户代码区域会清零。批量擦除操作仅擦除用户代码区域。包括写保护在内的其它选项字节将不受影响。只有在已激活级别 1 并请求级别 0 时，才会执行批量擦除。
当提高保护级别 (0->1、1->2、0->2) 时，不会执行批量擦除。
● 级别 2 ：禁止调试/ 芯片读保护
将 0xCC 写入 RDP 选项字节时，可激活读保护级别 2。设置读保护级别 2 后：
— 级别 1 提供的所有保护均有效。
— 不再允许从 RAM 或系统存储器自举。
— JTAG、 ETM 和边界扫描处于禁止状态。
— 用户选项字节不能再进行更改。
— 从 Flash 自举时，允许通过用户代码对 Flash 和备份 SRAM 进行访问（读取、擦除、编程）。
存储器读保护级别 2 是不可更改的。激活级别 2 后，保护级别不能再降回级别 0 或级别 1。
注意： 激活级别 2 后，将永久性禁止 JTAG 端口（相当于 JTAG 熔断）。

表 13. 不同读保护级别下的访问限制

存储区	保护级别	调试功能 从 RAM 或系统存储器自举			从 Flash 自举		
		读	写	擦除	读	写	擦除
主 Flash 和备份 SRAM	级别 1	否		否(1)	是		
	级别 2	否			是		
选项字节	级别 1	是			是		
	级别 2	否			否		
OTP	级别 1	否		NA	是		NA
	级别 2	否		NA	是		NA

1. 只有在 RDP 从级别 1 更改为级别 0 时，才会擦除主 Flash 和备份 SRAM。

4.6.4 写保护

Flash 中有多个用户扇区具备写保护功能，可防止因程序指针错乱而发生意外的写操作。当 FLASH_OPTCR 或 FLASH_OPTCR1 寄存器中的低有效写保护位 nWRPi (0 ≤i ≤11) 为低电平时，无法对相应的扇区执行擦除或编程操作。因此，如果某个扇区处于写保护状态，则无法对整个器件执行全部擦除。
如果尝试对 Flash 中处于写保护状态的区域执行擦除/编程操作,则 FLASH_SR 寄存器中的写保护错误标志位 (WRPERR) 将置 1。
注意： 选择存储器读保护级别（ RDP 级别 = 1 ）后，如果已连接 CPU 调试功能（ JTAG 调试或单线调试）或者正在从 RAM 执行启动代码，则即使 nWRPi = 1 ，也无法对 Flash 扇区 i 执行编程或擦除操作。

写保护错误标志

如果对 Flash 的写保护区域执行擦除/编程操作，则 FLASH_SR 寄存器中的写保护错误标志位 (WRPERR) 将置 1。

如果请求执行擦除操作，则以下情况下 WRPERR 位置 1：

- 配置全部擦除、块擦除、扇区擦除（MER 或 MER/MER1 和 SER = 1）
- 请求执行扇区擦除但扇区编号 SNB 字段无效
- 请求执行全部擦除，但至少一个用户扇区通过选项位实施了写保护（FLASH_OPTCRx 寄存器中的 MER 或 MER/MER1 = 1 且 nWRPi = 0，0 ≤ i ≤11 位）
- 请求针对写保护扇区执行扇区擦除。（FLASH_OPTCRx 寄存器中 SER = 1、SNB = i 且 nWRPi = 0，0 ≤ i ≤11 位）
- Flash 处于读保护状态，但检测到擦除操作企图。

如果请求执行编程操作，则以下情况下 WRPERR 位置 1：

- 针对系统存储器或用户特定扇区的保留区域执行写操作
- 针对用户配置扇区执行写操作
- 针对通过选项位实施写保护的扇区执行写操作
- 请求针对已锁定的区域执行写操作
- Flash 处于读保护状态，但检测到写操作企图

4.7 Flash 接口寄存器

4.7.1 Flash 访问控制寄存器 (FLASH_ACR)

Flash access control register

Flash 访问控制寄存器用于使能/关闭加速功能，并且可根据 CPU 频率控制 Flash 访问时间。

偏移地址：0x00

复位值：0x0000 0000

访问：无等待周期，按字、半字和字节访问

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

位 31:11 保留，必须保持清零。

位 12 DCRST ：数据缓存复位 (Data cache reset)

0：数据缓存不复位

1：数据缓存复位

只有在关闭数据缓存时才能在该位中写入值。

位 11 ICRST ：指令缓存复位 (Instruction cache reset)

0：指令缓存不复位

1：指令缓存复位
只有在关闭指令缓存时才能在该位中写入值。

位 10 DCEN ： 数据缓存使能 (Data cache enable)
0：关闭数据缓存
1：使能数据缓存

位 9 ICEN ： 指令缓存使能 (Instruction cache enable)
0：关闭指令缓存
1：使能指令缓存

位 8 PRFTEN ： 预取使能 (Prefetch enable)
0：关闭预取
1：使能预取

位 7:3 保留，必须保持清零。

位 2:0 LATENCY ： 延迟 (Latency)
这些位表示 CPU 时钟周期与 Flash 访问时间之比。
000：零等待周期
001：一个等待周期
010：两个等待周期
011：三个等待周期
100：四个等待周期
101：五个等待周期
110：六个等待周期
111：七个等待周期

4.7.2 Flash 密钥寄存器 (FLASH_KEYR)

Flash key register

借助 Flash 密钥寄存器，可允许对 Flash 控制寄存器的访问，进而允许执行编程和擦除操作。

偏移地址：0x04

复位值：0x0000 0000

访问：无等待周期，按字访问

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

位 31:0 FKEYR：FPEC 密钥 (FPEC key)
要将 FLASH_CR 寄存器解锁并允许对其执行编程/擦除操作，必须顺序编程以下值：
a) KEY1 = 0x45670123
b) KEY2 = 0xCDEF89AB

4.7.3 Flash 选项密钥寄存器 (FLASH_OPTKEYR)

Flash option key register
借助 Flash 选项密钥寄存器，可允许在用户配置扇区中执行编程和擦除操作。
偏移地址：0x08
复位值：0x0000 0000
访问：无等待周期，按字访问

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

位 31:0 OPTKEYR：选项字节密钥 (Option byte key)
要将 FLASH_OPTCR 寄存器解锁并允许对其编程，必须顺序编程以下值：
a) OPTKEY1 = 0x08192A3B
b) OPTKEY2 = 0x4C5D6E7F

4.7.4 Flash 状态寄存器 (FLASH_SR)

Flash status register
Flash 状态寄存器提供正在执行的编程和擦除操作的相关信息。
偏移地址：0x0C
复位值：0x0000 0000
访问：无等待周期，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															BSY
															r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								PGSERR	PGPERR	PGAERR	WRPERR	Reserved		OPERR	EOP
								rc_w1	rc_w1	rc_w1	rc_w1			rc_w1	rc_w1

位 31:17 保留，必须保持清零。
位 16 BSY ： 繁忙 (Busy)

该位指示 Flash 操作正在进行。该位在 Flash 操作开始时置 1，在操作结束或出现错误时清零。

0：当前未执行任何 Flash 操作

1：正在执行 Flash 操作

位 15:8 保留，必须保持清零。

位 7 PGSERR ：编程顺序错误 (Programming sequence error)

如果代码在控制寄存器未正确配置的情况下对 Flash 执行写访问，将由硬件为该位置 1。

写入 1 即可将该位清零。

位 6 PGPERR ：编程并行位数错误 (Programming parallelism error)

如果在编程期间数据访问类型（字节、半字、字和双字）与配置的并行位数 PSIZE (x8, x16,x32, x64) 不符，将由硬件为该位置 1。

写入 1 即可将该位清零。

位 5 PGAERR ：编程对齐错误 (Programming alignment error)

如果要编程的数据不能包含在同一个 128 位 Flash 行中，将由硬件为该位置 1。

写入 1 即可将该位清零。

位 4 WRPERR ：写保护错误 (Write protection error)

如果要擦除/编程的地址属于 Flash 中处于写保护状态的区域，将由硬件为该位置 1。

写入 1 即可将该位清零。

位 3:2 保留，必须保持清零。

位 1 OPERR ：操作错误 (Operation error)

如果检测到 Flash 操作（编程/擦除/读取）请求，但由于存在并行位数错误、对齐错误或写保护错误而无法运行，将由硬件对该位置 1。只有在使能错误中断 (ERRIE = 1) 后，该位才会置 1。

位 0 EOP ：操作结束 (End of operation)

当成功完成一个或多个 Flash 操作（编程/擦除）时，由硬件将该位置 1。只有在使能操作结束中断 (EOPIE = 1) 后，该位才会置 1。

写入 1 即可将该位清零。

4.7.5 Flash 控制寄存器(FLASH_CR)

Flash control register

Flash 控制寄存器用于配置和启动 Flash 操作。

偏移地址：0x10

复位值：0x8000 0000

访问：当前未执行任何 Flash 操作时无等待周期，按字、半字和字节访问。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						PSIZE[1:0]		Res.	SNB[3:0]				MER	SER	PG
						w	w		rw	rw	rw	rw	rw	rw	rw

位 31 LOCK ： 锁定 (Lock)
该位只能写入 1。该位置 1 时，表示 FLASH_CR 寄存器已锁定。当检测到解锁序列时，由硬件将该位清零。
如果解锁操作失败，该位仍保持置 1，直到下一次复位。
位 30:26 保留，必须保持清零。
位 25 ERRIE ： 错误中断使能 (Error interrupt enable)
当 FLASH_SR 寄存器中的 OPERR 位置 1 后，可通过该位使能中断产生功能。
0：禁止错误中断
1：使能错误中断
位 24 EOPIE ： 操作结束中断使能 (End of operation interrupt enable)
当 FLASH_SR 寄存器中的 EOP 位置 1 后，可通过该位使能中断产生功能。
0：禁止中断
1：使能中断
位 23:17 保留，必须保持清零。
位 16 STRT ： 启动 (Start)
该位置 1 后可触发擦除操作。该位只能通过软件置 1，并在 BSY 位清零后随之清零。
位 15:10 保留，必须保持清零。
位 9:8 PSIZE ： 编程大小 (Program size)
这些位用于选择编程并行位数。
00 x8 编程
01 x16 编程
10 x32 编程
11 x64 编程
位 6:3 SNB ： 扇区编号 (Sector number)
这些位用于选择要擦除的扇区。
0000 扇区 0
0001 扇区 1
...
1011 扇区 11
不允许使用其它值
位 2 MER ： 批量擦除 (Mass Erase)
针对所有用户扇区激活擦除操作。
位 1 SER ： 扇区擦除 (Sector Erase)
激活扇区擦除。
位 0 PG ： 编程 (Programming)
激活 Flash 编程。

4.7.6 Flash 选项控制寄存器 (FLASH_OPTCR)

Flash option control register
FLASH_OPTCR 寄存器用于修改用户选项字节。
偏移地址：0x14
复位值：0x0FFF AAED。复位信号释放时，硬件将 Flash 中的值加载到这些选项位。
访问：当前未执行任何 Flash 操作时无等待周期，按字、半字和字节访问。
位 2 MER ： 擦除块 1 里的所有扇区 (Mass Erase of bank 1 sectors)

激活对块 1 里所有扇区的擦除功能
位 1 SER ： 扇区擦除 (Sector Erase)
激活扇区擦除。
位 0 PG ： 编程 (Programming)
激活 Flash 编程。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				nWRP[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
RDP[7:0]								nRST_STDBY	nRST_STOP	WDG_SW	Reserved	BOR_LEV		OPTSTRT	OPTLOCK
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rs	rs

位 31:28 保留，必须保持清零。
位 27:16 nWRP ： 无写保护 (Not write protect)
这些位包含复位后扇区写保护选项字节的值。通过对这些位执行写操作，可将新的写保护值编程到 Flash。
0：开启所选扇区的写保护
1：关闭所选扇区的写保护
位 15:8 RDP ： 读保护 (Read protect)
这些位包含复位后读保护选项字节的值。通过对这些位执行写操作，可将新的读保护值编程到 Flash。
0xAA：级别 0，未激活读保护
0xCC：级别 2，激活芯片读保护
其它值：级别 1，激活存储器读保护
位 7:5 USER ： 用户选项字节 (User option bytes)
这些位包含复位后用户选项字节的值。通过对这些位执行写操作，可将新的用户选项字节值编程到 Flash。
位 7：nRST_STDBY
位 6：nRST_STOP

位 5：WDG_SW
注意：当 WDG 模式从硬件切换到软件或从软件切换到硬件时，需要执行系统复位才能更改生效。
位 4 保留，必须保持清零。

位 3:2 BOR_LEV ： BOR 复位级别 (BOR reset Level)
这些位包含释放复位信号所需达到的供电电压阈值。可通过对这些位执行写操作，来编程新的 BOR 级别。BOR 默认为关闭。当电源电压 (V DD) 降至所选 BOR 级别以下时，将产生器件复位。
00：BOR 级别 3 (VBOR3)，复位阈值电压为 2.70 V 到 3.60 V
01：BOR 级别 2 (VBOR2)，复位阈值电压为 2.40 V 到 2.70 V
10：BOR 级别 1 (VBOR1)，复位阈值电压为 2.10 V 到 2.40 V

11：BOR 关闭 (VBOR0)，复位阈值电压为 1.80 V 到 2.10 V

位 1 OPTSTRT ：启动选项 (Option start)

该位置 1 后可触发用户选项操作。该位只能通过软件置 1，并在 BSY 位清零后随之清零。

位 0 OPTLOCK ：锁定选项 (Option lock)

该位只能写入 1。该位置 1 时，表示 FLASH_OPTCR 寄存器已锁定。当检测到解锁序列时，由硬件将该位清零。

如果解锁操作失败，该位仍保持置 1，直到下一次复位。

4.7.7 Flash 接口寄存器映射

表 15. Flash 寄存器映射与复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	FLASH_ACR	Reserved																		DCRST	ICRST	DCEN	ICEN	PRFTEN		Reserved					LATENCY		
	Reset value																			0	0	0	0	0							0	0	0
0x04	FLASH_KEYR	KEY[31:16]																KEY[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	FLASH_OPT KEYR	OPTKEYR[31:16]																OPTKEYR[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0c	FLASH_SR	Reserved															BSY	Reserved								PGSERR	OGOERR	OGAERR	WROERR	Reserved		OPERR	EOP
	Reset value																0																
0X10	FLASH_CR	LOCK	Reserved														STRT	Reserved						PSIZ E[1:0]		Reserved	SNB[3:0]				MER	SER	PG	
	Reset value	1															0							0	0		0	0	0	0	0	0	0
0x14	FLASH_OPT CR	Reserved						nWRP[11:0]										RDP[7:0]								nRST_STDBY	nRST_STOP	WDG_SW	Reserved	BOR_LEV		OPTSTRT	OPTLOCK
	Reset value							1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1	0	1	1	1		0	1	0	1

5 Quad-SPI 接口 （QUADSPI）

5.1 前言

QUADSPI 是一种专用的通信接口，连接单、双或四（条数据线）SPI Flash 存储介质。该接 口可以在以下三种模式下工作：

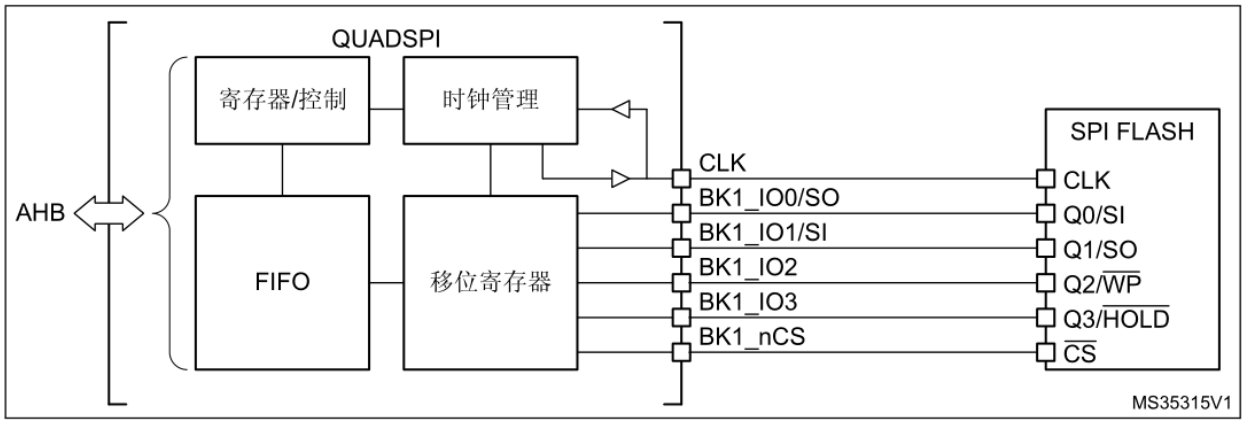
- 间接模式：使用 QUADSPI 寄存器执行全部操作
- 状态轮询模式：周期性读取外部 Flash 状态寄存器，而且标志位置 1 时会产生中断（如 擦除或烧写完成，会产生中断）
- 内存映射模式：外部 Flash 映射到微控制器地址空间，从而系统将其视作内部存储器

5.2 QUADSPI 主要特性

- 三种功能模式：间接模式、状态轮询模式和内存映射模式
- ~~双闪存模式，通过并行访问两个 Flash，可同时发送/接收 8 位数据~~
- 支持 SDR ~~和 DDR 模式~~
- 针对间接模式和内存映射模式，完全可编程操作码
- 针对间接模式和内存映射模式，完全可编程帧格式
- 集成 FIFO,用于发送和接收
- 允许 8、16 和 32 位数据访问
- 具有适用于间接模式操作的 DMA 通道
- 在达到 FIFO 阈值、超时、操作完成以及发生访问错误时产生中断

5.3 QUADSPI 功能说明

5.3.1 QUADSPI 框图

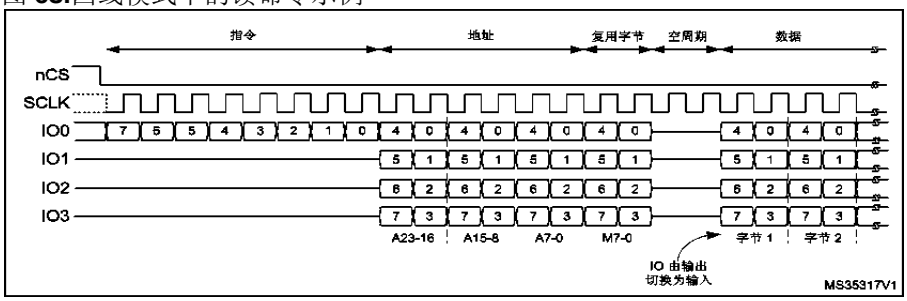


5.3.2 QUADSPI 命令序列

QUADSPI 通过命令与 Flash 通信每条命令包括指令、地址、交替字节、空指令和数据这五 个阶段任一阶段均可跳过，但至少要包含指令、地址、交替字节或数据阶段之一。

nCS 在每条指令开始前下降，在每条指令完成后再次上升。

图 **63.**四线模式下的读命令示例



指令阶段

这一阶段，将在 QUADSPI_CCR[7:0]寄存器的 INSTRUCTION 字段中配置的一条 8 位指令 发送到 Flash,指定待执行藻作的类型。

尽管大多数 Flash 从 IO0/SO 信号（单线 SPI 模式）只能以一次 1 位的方式接收指令，但指 令阶段可选择一次发送 2 位（在双线 SPI 模式中通过 IO0/IO1）或一次发送 4 位（在四线 SPI 模式中通过 IO0/IO1/IO2/IO3）。这可通过 QUADSPI_CCR[9:8]寄存器中的 IMODE[1:0] 字段进行配置。

若 IMODE = 00,则跳过指令阶段，命令序列从地址阶段（如果存在）开始。

地址阶段

在地址阶段，将 14 字节发送到 Flash, 指示操作地址。待发送的地址字节数在 QUADSPI_CCR[13:12]寄存器的 ADSIZE[1:0]字段中进行配置。在间接模式和自动轮询模 式下，待爱送的地址字节在 QUADSPLAR 寄存器的 ADDRESS[31:0]中指定。在内存映射 模式下，则通过 AHB （来自于 Cortex®或 DMA）直接给出地址。

地址阶段可一次发送 1 位（在单线 SPI 模式中通过 SO）、2 位（在双线 SPI 模式中通 过 IO0/IO1）或 4 位（在四线 SPI 模式中通过 IO0/IO1/IO2/IO3）。这可通过 QUADSPI_CCR[11:10]寄存器中的 ADMODE[1:0]字段进行配置。

若 ADMODE = 00,则跳过地址阶段，命令序列直接进入下一阶段（如果存在）。

交替字节阶段

在交替字节阶段，将 1-4 字节发送到 Flash, 一般用于控制操作模式。待发送的交替字节 数在 QUADSPI_CCR[17:16] 寄存器的 ABSIZE[1:0] 字段中进行配置。待发送的字节在 QUADSPI_ABR 寄存器中指定。

交替字节阶段可一次发送 1 位（在单线 SPI 模式中通过 S0）、2 位（在双线 SPI 模式 中通过 IO0/IO1）或 4 位（在四线 SPI 模式中通过 IO0/IO1/IO2/IO3）。这可通过 QUADSPI_CCR[15:14]寄存器中的 ABMODE[1:0]字段进行配置。

若 ABMODE = 00,则跳过交替字节阶段，命令序列直接进入下一阶段（如果存在）。

交替字节阶段存在仅需发送单个半字节而不是一个全字节的情况，比如采用双线模式并且仅使用两个周期发送交替字节时。在这种情况下，固件可采用四线模式（ABMODE = 11）并发 送一个字节，方法是 ALTERNATE 的位 7 和 3 置“1”（I03 保持高电平）且位 6 和 2 置 “0”（I02 线保持低电平）。此时，半字节的高 2 位存放在 ALTERNATE 的位 4:3,低 2 位存放在位 1 和 0 中。例如，如果半字节 2 （0010）通过 IO0/IO1 发送，则 ALTERNATE 应 设置为 0x8A（100_1010）。

空指令周期阶段

在空指令周期阶段，给定的 1-31 个周期内不发送或接收任何数据，目的是当采用更高的时钟频率时，给 Flash 留出准备数据阶段的时间。这一阶段中给定的周期数在 QUADSPI_CCR[22:18]寄存器的 DCYC[4:0]字段中指定。在 SDR 模式下，持续时 间被指定为 一定个数的全时钟周期。

若 DCYC 为零，则跳过空指令周期阶段，命令序列直接进入数据阶段（如果存在）。

空指令周期阶段的操作模式由 DMODE 确定。

为确保数据信号从输出模式转变为输入模式有足够的“周转”时间，使用双线和四线模式从 Flash 接收数据时，至少需要指定一个空指令周期。

数据阶段

在数据阶段，可从 Flash 接收或向其发送任意数量的字节。

在间接模式和自动轮询模式下，待发送/接收的字节数在 QUADSPI_DLR 寄存器中指定。

在间接写入模式下，发送到 Flash 的数据必须写入 QUADSPLDR 寄存器。在间接读取模式 下，通过读取 QUADSPI_DR 寄存器获得从 Flash 接收的数据。

在内存映射模式下，读取的数据通过 AHB 直接发送回 Cortex 或 DMA。

数据阶段可一次发送/接收 1 位（在单线 SPI 模式中通过 SO）、2 位（在双线 SPI 模式中通

过 IO0/IO1 ）或 4 位（在四线 SPI 模式中通过 IO0/IO1/IO2/IO3)。这可通过 QUADSPI_CCR[15:14]寄存器中的 ABMODE[1:0]字段进行配置。

若 DMODE = 00,则跳过数据阶段，命令序列在拉高 nCS 时立即完成。这一配置仅可用于 仅间接写入模式。

5.3.3 QUADSPI 信号接口协议模式

单线 SPI 模式

传统 SPI 模式允许串行发送/接收单独的 1 位。在此模式下，数据通过 SO 信号（其 I/O 与 IOO 共享）发送到 Flash.从 Flash 接收到的数据通过 SI 送达。

通过将（QUADSPI_CCR 中的）IMODE/ADMODE/ABMODE/DMODE 字段设置为 01,可 对不同的命令阶段否期进行配置，以使用此单个位模式。

在每个已配置为单线模式的阶段中：

- IO0 （SO）处于输出模式
- IO 1 （SI）处于输入模式（高阻抗）
- IO 2 处于输出模式并强制置“0”（以禁止“写保护”功能）
- IO 3 处于输出模式并强制置“1”（以禁止“保持”功能）

若 DMODE = 01,这对于空指令阶段也同样如此。

双线 SPI 模式

在双线模式下，通过 I00/I01 信号同时发送/接收两位。

通过将 QUADSPI_CCR 寄存器的 IMODE/ADMODE/ABMODE/DMODE 字段设置为 10,可 对不同的命令阶爵别进行配置，以使用双线 SPI 模式。

在每个已配置为双线模式的阶段中：

- IO0/IO1 在数据阶段进行读取操作时处于高阻态(输入)，在其他情况下为输出
- IO2 处于输出模式并强制置“0”
- IO3 处于输出模式并强制置“1 ”

在空指令阶段，若 DMODE = 01,则 IO0/IO1 始终保持高阻态。

四线 SPI 模式

在四线模式下，通过 IO0/IO1/IO2/IO3 信号同时发送/接收四位。

通过将 QUADSPI_CCR 寄存器的 IMODE/ADMODE/ABMODE/DMODE 字段设置为 11,可 对不同的命令阶翊别进行配置，以使用四线 SPI 模式。

在每个已配置为四线模式的阶段中，IO0/IO1/IO2/IO3 在数据阶段进行读取操作时均处于高 阻态(输入)，在其他情况下为输出。

在空指令阶段中，若 DMODE = 11,则 IO0/IO1/IO2/IO3 均为高阻态。

I02 和 I03 仅用于 Quad SPI 模式如果未配置任何阶段使用四线 SPI 模式，即使 QUADSPI 激活，对应 IO2 和 I03 的引脚也可用于其他功能。

SDR 模式

默认情况下，DDRM 位(QUADSPI_CCR[31])为 0，QUADSPI 在单倍数据速率(SDR)模式 下工作。

在 SDR 模式下，当 QUADSPI 驱动 IOO/SO、101 > 102、IO3 信号时，这些信号仅在 CLK 的下降沿发生转变。

在 SDR 模式下接收数据时，QUADSPI 假定 Flash 也通过 CLK 的下降沿发送数据。默认情 况下(SSHIFT = 0 时)，将使用 CLK 后续的边沿(上升沿)对信号进行采样。

5.3.4 QUADSPI 间接模式

在间接模式下，通过写入 QUADSPI 寄存器来触发命令；并通过读写数据寄存器来传输数 据，就如同对待其他通信外设那样。

若 FMODE = 00 （QUADSPI_CCR[27:26]）,则 QUADSPI 处于间接写入模式，字节在数据阶 段中发送到 Flash。通过写入数据寄存器（QUADSPI_DR）的方式提供数据。

若 FMODE = 01,则 QUADSPI 处于间接读取模式，在数据阶段中从 Flash 接收字节。通过 读取 QUADSPI_DR 来获取数据。

读取/写入的字节数在数据长度寄存器（QUADSPLDLR）中指定。如果 QUADSPI_DLR = OxFFF 已 FFFF （全为“1”），则数据长度视为未定义，QUADSPI 将继续传输数据，直到到达 （由 FSIZE 定义的） Flash 的结尾。如果不传输任何字节， DMODE（QUADSPI_CCR[25:24]） 应设置为 00。

如果 QUADSPI_DLR = OxFFFF_FFFF 并且 FSIZE = 0x1 F （最大值指示一个 4GB 的 Flash）,在此特殊情况下，传输将无限继续下去，仅在出现终止请求或 QUADSPI 被禁止后停止。在读取最后一个存储器地址后（地址为 OxFFFF_FFFF）,将从地址=0x0000_0000 开始继续读取。

当发送或接收的字节数达到编程设定值时，如果 TCIE = 1,则 TCF 置 1 并产生中断。在数据数量不确定的情况下，将根据 QUADSPI_CR 中定义的 Flash 大小，在达到外部 SPI 的限制时，TCF 置 1。

触发命令启动

从本质上讲,在固件给出命令所需的最后一点信息时,命令即会启动。根据 QUADSPI 的配 置，在间接模式下有三种触发命令启动的方式。在出现以下情形时，命令立即启动：

1 .对 INSTRUCTION[7:0] （QUADSPI_CCR）执行写入操作，如果没有地址是必需的（当 ADMODE = 00）并且不需要固件盛祇数据（当 FMODE = 01 或 DMODE = 00）

2 .对 ADDRESS[31:0] （QUADSPI_AR）执行写入操作，如果地址是必需的（当 ADMODE = 00）并且不需要固件提供数据（当 FMODE = 01 或 DMODE = 00）

3 .对 DATA[31:0]（QUADSPI_DR）执行写入操作，如果地址是必需的（当 ADMODE != 00）并且需要固件提供数据（当 FMODE = 00 并且 DMODE != 00）

写入交替字节寄存器（QUADSPI_ABR）始终不会触发命令启动。如果需要交替字节，必须预先进行编程。

如果命令启动，BUSY 位（QUADSPI_LSR 的位 5）将自动置 1。

FIFO 和数据管理

在间接模式中，数据将通过 QUADSPI 内部的一个 32 字节 FIFO。FLEVEL[5:0]（QUADSPI_SR[13:8]）指示 FIFO 目前保存了多少字节。

在间接写入模式下（FMODE = 00）,固件写入 QUADSPI_DR 时，将在 FIFO 中加入数据。 字写入将在 FIFO 中增加 4 个字节，半字写入增加 2 个转，而字节写入仅增加 1 个字节。
如果固件在 FIFO 中加入的数据过多（超过 DL[31:0]指示的值），将在写入操作结束（TCF 置 1）时从 FIFO 中清除超出的字节。

对 QUADSPI_DR 的字节/半字访问必须仅针对该 32 位寄存器的最低有效字节/半字。

FTHRES[3:0]用于定义 FIFO 的阈值如果达到阈值，FTF （FIFO 阈值标志）置 1 在间接读 取模式下，从 FIFO 中读取的有效字节数超过阈值时，FTF 置 1。从 Flash 中读取最后一个 字节后，如果 FIFO 中依然有数据，则无论 FTHRES 的设置为何，FTF 也都会置 1。在间接 写入模式下，当 FIFO 中的空字节数超过阈值时，FTF 置 1。

如果 FTIE = 1,则 FTF 置 1 时产生中断。如果 DMAEN = 1,则 FTF 置 1 时启动数据传送。 如果阈值条件不再为“真”（CPU 或 DMA 传输了足够的数据后），则 FTF 由 HW 清零。

在间接模式下，当 FIFO 已满，QUADSPI 将暂时停止从 Flash 读取字节以避免上溢。

5.3.5 QUADSPI 状态标志轮询模式

在自动轮询模式下，QUADSPI 周期性启动命令以读取一定数量的状态字节（最多 4 个）。 可屏蔽接收的字节以隔离一些状态位，从而在所选的位具有定义的值时可产生中断。

对 Flash 的访问最初与在间接读取模式下相同：如果不需要地址(AMODE = 00),则在写入 QUADSPLCCR 时即开始访问。否则，如果需要地址，则在写入 QUADSPI_AR 时开始第 一次访问。BUSY 在此时变为高电平，即使在周期性访问期间也保持不变。

在自动轮询模式下，MASK[31:0] (QUADSPI_PSMAR)的内容用于屏蔽来自 Flash 的数据。如果 MASK[n] = 0,则屏蔽结果的位 n,从而不考虑该位。如果 MASK[n] = 1 并且位[n]的内 容与 MATCH[n] (QUADSPI_PSMAR)相同，说明存在位 n 匹配。

如果轮询匹配模式位(PMM, QUADSPI_CR[23])为 0,将激活“AND”匹配模式。这意味着 状态匹配标志(SMF)仅在全部未屏蔽碣存在匹配时置 1。

如果 PMM = 1,则激活“OR”匹配模式。这意味着 SMF 在任意未屏蔽位存在匹配时置 1。

如果 SMIE = 1,则在 SMF 置 1 时调用一个中断。

如果自动轮询模式停止(APMS)位置 1,则操作停止并且 BUSY 位在检测到匹配时清零。否则, BUSY 位保持为“1”， 在发生中止或禁止 QUADSPI (EN = 0)前继续进行周期性访问。

数据寄存器（QUADSPI_DR)包含最新接收的状态字节（FIFO 停用）。数据寄存器的内容不受匹配逻辑所用屏蔽方法的影响。FTF 状态位在新一次状态读取完成后置 1,并且 FTF 在数据读取后清零。

5.3.6 QUADSPI 内存映射模式

在配置为内存映射模式时，外部 SPI 器件被视为是内部存储器。

映射地址：**0x90000000-0x9FFFFFFF**

QUADSPI 外设若没有正确配置并使能，禁止访问 QUADSPI Flash 的存储区域。

即使 Flash 容量更大，寻址空间也无法超过 256MB。

如果访问的地址超出 FSIZE 定义的范围但仍在 256MB 范围内，则生成 AHB 错误。此错误 的影响具体取决于尝试进行访问的 AHB 主设备：

- 如果为 Cortex® CPU,则生成硬性故障（Hard fault)中断
- 如果为 DMA,则生成 DMA 传输错误，并自动禁用相应的 DMA 通道。

支持字节、半字和字访问类型。

默认情况下，即便在很长时间内不访问 Flash, QUADSPI 也不会停止预取操作，之前的读 取操作将保持激活状态并且 nCS 保持低电平。由于 nCS 保持低电平时，Flash 功耗增加， 应用程序可能会激活超时计数器(TCEN = 1, QUADSPI_CR[3]).从而在 FIFO 中写满预取的 数据后，若在 TIMEOUT[15:0](QUADSPLI_PTR)个周期的时长内没有访问，则释放 nCS。

BUSY 在第一个存储器映射访问发生时变为高电平。由于进行预取操作，BUSY 在发生超 时、中止或外设禁止前不会下降。

5.3.7 QUADSPI Flash 配置

设备配置寄存器(QUADSPI_DCR)可用于指定外部 SPI Flash 的特性。

FSIZE[4:0]字段使用下面的公式定义外部存储器的大小：

Flash 中的字节数=$2^{[FSIZE+1]}$

FSIZE+1 是对 Flash 寻址所需的地址位数。在间接模式下，Flash 容量最高可达 4GB（使用 32 位进行寻址），但在内存映射模式下的可寻址空间限制为 256MB。

如果 DFM = 1, FSIZE 表示两个 Flash 容量的总和。

QUADSPI 连续执行两条命令时，它在两条命令之间将片选信号(nCS)置为高电平默认仅一 个 CLK 周期时长。如果 Flash 需要命令之间的时间更长，可使用片选高电平时间(CSHT)字 段指定 nCS 必须保持高电平的最少 CLK 周期数(最大为 8)。

时钟模式(CKMODE)位指示命令之间的 CLK 信号逻辑电平(nCS = 1 时)。

5.3.8 QUADSPI 延迟数据采样

默认情况下，QUADSPI 在 Flash 驱动信号后过半个 CLK 周期才对 Flash 驱动的数据采样。

在外部信号延迟时，这有利于推迟数据采样。使用 SSHIFT 位(QUADSPI_CR[4]),可将数 据采样移位半个 CLK 周期。

5.3.9 QUADSPI 配置

QUADSPI 配置分两个阶段

• QUADSPI IP 配置

• QUADSPI Flash 配置

QUADSPI 在配置完毕并使能后，即可在间接模式、状态轮询模式和内存映射模式这三种操 作模式之一下工作。

QUADSPI IP 配置

通过 QUADSPI_CR 配置 QUADSPI IP 。用户应配置传入数据的时钟预分频器的分频系数以及采样移位设置。

DMA 请求通过 DMAEN 位置 1 使能。若是用于中断，则相关使能位也可在该阶段置 1。

生成 DMA 请求或生成中断的 FIFO 电平在 FTHRES 位中进行编程。

如果需要超时计数器，则可将 TCEN 位置 1 并在 QUADSPI_LPTR 寄存器中编程超时值。

QUADSPI Flash 配置

与外部目标 Flash 相关的参数通过 QUADSPI_DCR 寄存器进行配置。用户应在 FSIZE 位中 编程 Flash 的大小、在 CSHT 位中编程片选保持高电平的最短时间以及在 MODE 位中编程 功能模式（模式 0 或模式 3）。

5.3.10 QUADSPI 的用法

使用 FMODE[1:0] （QUADSPI_CCR[27:26]）选择操作模式。

间接模式的操作步骤

FMODE 编程为 00 可选择间接写入模式，将数据发送到 Flash。FMODE 编程为 01 可选择间接读取模式，读取 Flash 中的数据。

QUADSPI 用于间接模式时，采用以下方式构建帧：

- 在 QUADSPI_DLR 中指定待读取或写入的字节数
- 在 QUADSPI_CCR 中指定帧格式、模式和指令代码
- 在 QUADSPI_ABR 中指定要在地址阶段后立即发送的可选交替字节
- 在 QUADSPI_CR 中指定工作模式。若 FMODE = 00 （间接写入模式）并且 DMAEN = 1, 则应在 QUADSPI_CR 前指定 QUADSPI_AR。否则在 QUADSPI_AR 更新前（如果 已经使能 DMA 控制嘉），DMA 便可能写入 QUADSPI_DR。
- 在 QUADSPI_AR 中指定目标地址
- 通过 QUADSPI_DR 从 FIFO 读取数据/向 FIFO 写入数据

在写入控制寄存器（QUADSPI_CR）时，用户可指定以下设置：

- 使能位（EN）设置为“1”
- DMA 使能位（DMAEN）,用于向/从 RAM 传输/接收数据
- 超时计数器使能位（TCEN）
- 采样移位设置（SSHIFT）
- FIFO 阈值（FTRHES）,以指示 FTF 标志在何时置 1
- 中断使能
- 自动轮询模式参数：匹配模式和停止模式（在 FMODE = 11 时有效）
- 时钟预分频器

在写入通信配置寄存器（QUADSPLCCR）时，用户指定以下参数：

- 通过 INSTRUCTION 位指定指令字节
- 通过 IMODE 位指定指令发送方式（1/2/4 线）
- 通过 ADMODE 位指定地址发送方式（无/1/2/4 线）
- 通过 ADSIZE 位指定地址长度（8/16/24/32 位）
- 通过 ABMODE 位指定交替字节发送方式（无/1/2/4 线）
- 通过 ABSIZE 位指定交替字节数（1/2/3/4）
- 通过 DBMODE 位指定是否存在空指令字节
- 通过 DCYC 位指定空指令字节数
- 通过 DMODE 位指定数据发送/接收方式（无/1/2/4 线）

如果无需为某个命令更新地址寄存器（QUADSPI_AR）与数据寄存器（QUADSPI_DR）,则在 写入 QUADSPI_CCR 时，该命令序列便立即启动。在 ADMODE 和 DMODE 应为 00 时，或在间接读取模式（FMODE = 01）下仅 ADMODE = 00 时，便属于此情况。 在需要地址（ADMODE 不为 00）,但无需写入数据寄存器（FMODE = 01 或 DMODE = 00）时，通过写入 QUADSPI_AR 更新地址后，命令序列便立即启动。

在数据传输（FMODE = 00 并且 DMODE ! = 00）中，通过 QUADSPI_DR 写入 FIFO 触发 通

道启动。

状态标志轮询模式

将 FMODE 字段（QUADSPI_CCR[27:26]）设置为 10,使能状态标志轮询模式在此模式下， 将发送编程的帧并周期性检素数据。

每帧中读取的最大数据量为 4 字节。如果 QUADSPI_DLR 请求更多的数据，则忽略多余的 部分并仅读取 4 个字节。

在 QUADSPI_PISR 寄存器中指定周期性。

在检索到状态数据后，可在内部进行处理，以达到以下目的：

- 将状态匹配标志位置 1,如果使能，还将产生中断
- 自动停止周期性检索状态字节

接收到的值可通过存储于 QUADSPI_PSMKR 中的值进行屏蔽，并与存储在 QUADSPI_PSMAR 中的值进行或运算或与运算。

若是存在匹配，则状态匹配标志置 1,并且在使能了中断的情况下还将产生中断；如果 AMPS 位置 1,则 QUADSPI 自动停止。

在任何情况下，最新的检索值都在 QUADSPI_DR 中可用。

内存映射模式

在内存映射模式下，外部 Flash 被视为内部存储器，只是存在访问延迟。在该模式下，仅允 许对外部 Flash 执行读取操作。

将 QUADSPI_CCR 寄存器中的 FMODE 设置为 11 可进入内存映射模式。

当 AHB 主器件访问存储器映射空间时，将发送已编程的指令和帧。

FIFO 用作预取缓冲区以接受线性读取。在此模式中，对于 QUADSPI_DR 的任何访问均返 回零。

数据长度寄存器（QUADSPI_DLR）在内存映射模式中无意义。

5.3.11 指令仅发送一次

一些 Flash （例如 Winbound）能够提供一种模式，指令在该模式中仅通过第一个命令序列 进行发送，后续的命令根据地址直接启动。用户通过使用 SIOO 位（QUADSPI_CCR[28]）可 利用此功能的优势。

SIOO 对于所有功能模式（间接模式、状态轮询模式和内存映射模式）均有效。如果 SIOO 位置 1,仅第一条命令发送指令，接着对 QUADSPI_CCR 执行写入操作。后续命令序列都 将跳过指令阶段，直到 QUADSPI_CCR 被写入为止。

IMODE = 00 （无指令）时，SIOO 不起作用。 QUADSPI 差错管理

在以下情况下可能产生错误：

- 在间接模式或状态标志轮询模式下，如果在 QUADSPI_AR 中编程了错误的地址（根据 QUADSPI_DCR 中 FSIZE[4:0]定义的 Flash 大小）：TEF 将置 1,如果使能，还将产 生中断。
- 另外，在间接模式下，如果地址加数据的长度超过 Flash 的大小，TEF 将在访问被触发 时置 1。
- 在内存映射模式下，当 AHB 主器件执行的访问超出范围或 QUADSPI 被禁止时：将产 生 AHB 错误，以响应故障 AHB 请求。
- 当 AHB 主机访问存储器映射空间，但内存映射模式被禁止时：将产生 AHB 错误，以响 应

故障 AHB 请求。

5.3.12 QUADSPI 的繁忙位和中止功能

在 QUADSPI 启动对 Flash 的操作时，QUADSPI_SR 中的 BUSY 位自动置 1。

在间接模式下，在 QUADSPI 完成了请求的命令序列并且 FIFO 为空时，BUSY 位复位。

在自动轮询模式下，仅当最后一次周期性访问完成时（因 APMS = 1 时发生匹配，或因中 止），BUSY 位才变为低电平。

在内存映射模式下进行第一次访问后，仅在发生超时事件或中止时 BUSY 位变为低电平。

任何操作都可通过将 QUADSPI_CR 中的 ABORT 位置 1 来中止。在完成中止时，BUSY 位 和 ABORT 位自动复位，FIFO 淆空。

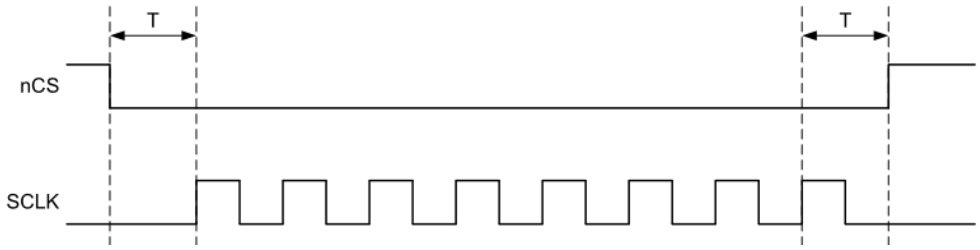
注: 如果中止对状态寄存器的写入操作, 有些 Rash 可能发生错误行为。

5.3.13 nCS 行为

默认情况下，nCS 为高电平，取消选择外部 Flash。nCS 在操作开始前下降，在操作完成时 立即上升。

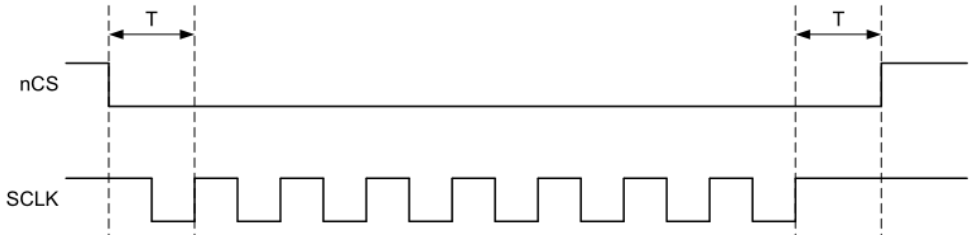
当 CKMODE = 0 （“模式 0”，在未进行任何操作时 CLK 保持低电平）时，nCS 在操作首 次升高 CLK 边沿时的一个 CLK 周期前降至低电平，在操作最后一次升高 CLK 边沿时的一个 CLK 周期后升至高电平，如下图 65 所示。

图 65. CKMODE = 0 时的 nCS （T = CLK 周期）



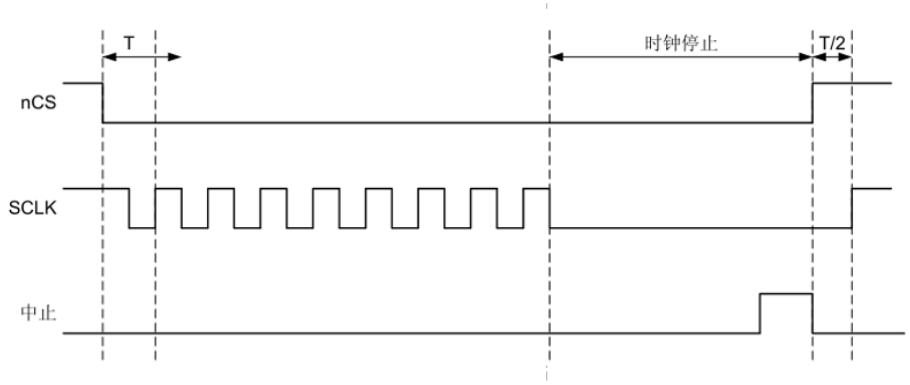
当 CKMODE = 1（“模式 3”，在未进行任何操作时 CLK 升至高电平）且 DDRM = 0（SDR 模式）时，nCS 仍在操作首次升高 CLK 边沿时的一个 CLK 周期前降至低电平，在操作最后一次升高 CLK 边沿时的一个 CLK 周期后升至高电平，如 图 66 所示。

图 66. SDR 模式下 CKMODE = 1 时的 nCS （T = CLK 周期）



若 FIFO 在读取操作中保持写满状态或在写入操作中保持为空，则在固件干预 FIFO 前，操作停止并且 CLK 保持低电平。若操作停止时发生中止，则 nCS 在请求中止后即升至高电平，CLK 则在半个周期后升至高电平，如 图 68 所示。

图 68.CKMODE=1 且发生中止时的 nCS(T=CLK 周期)



5.4 QUADSPI 中断

发生如下事件时可生成中断:

- ~~超时~~
- 状态匹配
- FIFO 阈值
- 传输完成
- ~~传输错误 可以使用单独的中断使能位以提高灵活性。~~

表 88. QUADSPI 中断请求

中断事件	事件标志	使能控制位
~~超时~~	~~TOF~~	~~TOIE~~

状态匹配	SMF	SMIE
FIFO 阈值	FTF	FTIE
传输完成	TCF	TCIE
~~传输错误~~	~~TEF~~	~~TEIE~~

5.5 QUADSPI 寄存器

5.5.1 QUADSPI 控制寄存器(QUADSPLCR)

QUADSPI control register

偏移地址：0x0000

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRESCALER								PMM	APMS	Res.	~~TOIE~~	SMIE	FTIE	TCIE	~~TEIE~~
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.	Res.	Res.	FTHRES					~~FSEL~~	~~DFM~~	Res.	SSHIFT	TCEN	DMAEN	ABORT	EN
			rw	rw	rw	rw	rw	rw	rw		rw	rw	w1s	rw	w1s

位 31:24 PRESCALER[7:0]：时钟预分频器(Clock prescaler)

该字段定义基于 AHB 时钟生成 CLK 所用的分频系数(值+ 1).

0： $F_{CLK}=F_{AHB}$，AHB 直接用作 QUADSPI CLK（预分频器被旁路）

$^{F}CLK = {}^{F}AHB/2$

2： $F_{CLK} = F_{AHB}/3$

255： $F_{CLK} = F_{AHB}/256$

对于奇数时钟分频系数，CLK 的占空比并非 50%。时钟信号的高电平持续时间比低电平 持续时间多一个周期

仅可在 BUSY = 0 时修改该字段。

位 23 PMM：轮询匹配模式(Polling match mode)

该位指示在自动轮询模式期间用来确定是否“匹配”的方法。

0：AND 匹配模式。如果从 Flash 接收的所有未屏蔽位均与匹配寄存器中的对应位相匹 配，贝!1SMF 置 1。

1： OR 匹配模式。如果从 Flash 接收的任意一个未屏蔽位与匹配寄存器中的对应位相匹配，贝!1SMF 置 1。

仅可在 BUSY = 0 时修改该位。

位 22 APMS：自动轮询模式停止(Automatic poll mode stop)

该位确定在匹配后自动轮询是否停止。

0：仅通过中止或禁用 QUADSPI 停止自动轮询模式。

1：发生匹配时，自动轮询模式停止。

仅可在 BUSY = 0 时修改该位。

位 21 保留，必须保持复位值。

~~位 20 TOIE：TimeOut 中断使能(TimeOut interrupt enable)~~
~~该位使能 TimeOut 中断。~~
~~0：禁止中断~~
~~1：使能中断~~

位 19 SMIE：状态匹配中断使能（Status match interrupt enable)
该位使能状态匹配中断。
0：禁止中断
1：使能中断

位 18 FTIE： FIFO 阈值中断使能(FIFO threshold interrupt enable)<,
该位使能 FIFO 阈值中断。
0：禁止中断
1：使能中断

位 17 TCIE：传输完成中断使能(Transfer complete interrupt enable)
该位使能传输完成中断。
0：禁止中断
1 ：使能中断

~~位 16 TEIE：传输错误中断使能(Transfer error interrupt enable)~~
~~该位使能传输错误中断。~~
~~0：禁止中断~~
~~1：使能中断~~

位 15:13 保留，必须保持复位值。

位 12:8 FTHRES[4:0]. FIFO 阈值级别(FIFO threshold level)
定义在间接模式下 FIFO 中将导致 FIFO 阈值标志（FTF, QUADSPI_SR[2])置 1 的字 节数阈值.
在间接写入模式下(FMODE = 00)：
0：如果 FIFO 中存在 1 个或更多空闲字节可供写入，则 FTF 置 1
1：如果 FIFO 中存在 2 个或更多空闲字节可供写入，则 FTF 置 1

31：如果 FIFO 中存在 32 个空闲字节可供写入，则 FTF 置 1
在间接读取模式下(FMODE = 01)：
0：如果 FIFO 中存在 1 个或更多有效字节可供读取，则 FTF 置 1
1：如果 FIFO 中存在 2 个或更多有效字节可供读取，则 FTF 置 1

31：如果 FIFO 中存在 32 个有效字节可供读取，则 FTF 置 1
如果 DMAEN = 1,则在更改 FTHRES 值前，必须禁止相应通道的 DMA 控制器。

~~位 7 FSEL：Flash 选择(Flash memory selection)~~
~~该位选择单闪存模式(DFM = 0)下要寻址的 Flash.~~
~~0：选择 FLASH 1~~
~~1：选择 FLASH 2~~
~~仅可在 BUSY = 0 时修改该位。~~
~~在 DFM = 1 时忽略该位。~~

~~位 6 DFM：双闪存模式(Dual-flash mode)~~
~~该位激活双闪存模式，同时使用两个外部 Flash 以将吞吐量和容量扩大一倍。~~
~~0：禁止双闪存模式~~
~~1：使能双闪存模式。~~
~~仅可在 BUSY = 0 时修改该位。~~

位 5 保留，必须保持复位值。
位 4 SSHIFT：采样移位(Sample shift)
默认情况下，QUADSPI 在 Flash 驱动数据后过半个 CLK 周期开始采集数据。使用该 位，

可考虑外部信号延迟，推迟数据采集。
0：不发生移位
1：移位半个周期
在 DDR 模式下(DDRM = 1),固件必须确保 SSHIFT = 0。
仅可在 BUSY = 0 时修改该字段。

位 3 TCEN：超时计数器使能(Timeout counter enable)
该位仅在内存映射模式(FMODE = 11)下有效。激活该位后，如果在一段时间(通过 TIMEOUT[15:0] (QUADSPI_LPTR)定义)内一直没有进行访问，将释放片选(nCS)（从而降低功耗）。
使能超时计数器。
默认情况下，即便在很长时间内不访问 Flash, QUADSPI 也不会停止预取操作，之前的 读取操作将保持激活状态并且 nCS 保持低电平。由于 nCS 保持低电平时，Flash 功耗 增加，应用程序可能会激活超时计数器(TCEN = 1, QUADSPI_CR[3])o 从而在 FIFO 中 写满预取的数据后，若在 TIMEOUT[15:0] (QUADSPLLPTR)个周期的时长内没有访 问，则释放 nCS。
0：禁止超时计数器，在内存映射模式中进行访问后，片选(nCS)保持激活。
1：使能超时计数器，在内存映射模式下，Flash 持续不活动 TIMEOUT[15:0]个周期后 释放片选(nCS)。
仅可在 BUSY = 0 时修改该位。

位 2 DMAEN： DMA 使能(DMA enable)
在间接模式下，通过 QUADSPI_DR 寄存器可使用 DMA 输入或输出数据。FIFO 阈值标 志 FTF 置 1 时，将触发 DMA 桎输。
0：间接模式下禁止 DMA
1：间接模式下使能 DMA

~~位 1 ABORT：中止请求(Abort request)~~
~~该位中止执行中的命令序列。在中止完成时自动复位。~~
~~该位可停止当前的传输。~~
~~在轮询模式或内存映射模式下，该位也用以复位 APM 位或 DM 位。~~
~~0：不请求中止~~
~~1：请求中止~~

位 0 EN：使能(Enable)
使能 QUADSPL
0：禁止 QUADSPI
1：使能 QUADSPI

5.5.2 QUADSPI 器件配置寄存器(QUADSPI_CCR)

QUADSPI device configuration register

偏移地址：0x0004 复位值：0x0000 0000

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

位 31:21 保留，必须保持复位值。

位 20:16 FSIZE[4:0]： Flash 大小(Flash memory size)
该字段使用下面的公式定义了外部存储器的大小：

Flash 中的字节数=21FSIZE+l]
FSIZE+1 是对 Flash 寻址所需的地址位数。在间接模式下，Flash 容量最高可达 4GB（使用 32 位进行寻址），但在内存映射模式下的可寻址空间限制为 256MB。
如果 DFM = 1, FSIZE 表示两个 Flash 容量的总和。
仅可在 BUSY = 0 时修改该字段。

位 15:11 保留，必须保持复位值。

位 10:8 CSHT[2:0]：片选高电平时间(Chip select high time)
CSHT+1 定义片选(nCS)在发送至 Flash 的命令之间必须保持高电平的最少 CLK 周 期数。
0： nCS 在 Flash 命令之间保持高电平至少 1 个周期
1： nCS 在 Flash 命令之间保持高电平至少 2 个周期

7： nCS 在 Flash 命令之间保持高电平至少 8 个周期
仅可在 BUSY = 0 时修改该字段。

位 7:1 保留，必须保持复位值。

位 0 CKMODE：模式 0/模式 3 (Mode 0 / mode 3)
该位指示 CLK 在命令之间（nCS=1 时）的电平。
0： nCS 为高电平（片选释放）时，CLK 必须保持低电平。这称为模式 0。
1： nCS 为高电平（片选释放）时，CLK 必须保持高电平。这称为模式 3。
仅可在 BUSY = 0 时修改该字段。

5.5.3 QUADSPI 状态寄存器(QUADSPI_SR)

QUADSPI status register

偏移地址：0x0008 复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	FLEVEL[5:0]						Res.	Res.	BUSY	TOF	SMF	**FTF**	**TCF**	**TEF**
		r	r	r	r	r	r			r	r	r	r	r	r

位 31:14 保留，必须保持复位值。

位 13:8 FLEVEL[5:0]： FIFO 级别（FIFO level)
该字段给出 FIFO 中的有效字节数• FIFO 为空时 FLEVEL = 0,写满时 FLEVEL = 32. 在内存映射模式和自动状态轮询模式下，FLEVEL 为零。

位 7:6 保留，必须保持复位值。

位 5 BUSY：忙（Busy)
操作进行时，该位置 1。在对 Flash 的操作完成并且 FIFO 为空时，该位自动清零。

~~位 4 TOF：超时标志(Timeout flag)~~
~~发生超时时该位置 1o 向 CTOF 写入 1 可将该位清零。~~

位 3 SMF：状态匹配标志（Status match flag)
在自动轮询模式下，若未屏蔽的接收数据与匹配寄存器(QUADSPLPSMAR)中的对应 位相匹配，则该位置 1。向 CSMF 写入 1 可将该位清零。

位 2 FTF： FIFO 阈值标志(FIFO threshold flag)
在间接模式下，若达到 FIFO 阈值，或从 Flash 读取完成后，FIFO 中留有数据时，该位 置

1。只要阈值条件不再为“真”，该位就自动清零。
在自动轮询模式下，执行 POLLING 命令后（FMOD=10），FTF 会马上被清零。~~每次读取状态寄存器时，该位即置 1；读取数据寄存器时，该位 清零。~~

位 1 TCF：传输完成标志(Transfer 8mpiete flag)
在间接模式下，当传输的数据数量达到编程设定值，或在任何模式下传输中止时，该位 置 1。向 CTCF 写入 1 时，该位清零。

位 0 TEF：传输错误标志(Transfer error flag)
在间接模式下访问无效地址时，该位置 1。向 CTEF 写入 1 可将该位清零。

5.5.4 QUADSPI 标志清零寄存器(QUADSPI_FCR)

QUADSPI flag clear register

偏移地址：OxOOOC

复位值：0x0000 0000

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
											CTOF	CSMF		CTCF	CTEF
											w1o	w1o		w1o	w1o

位 31:4 保留，必须保持复位值.

位 4 CTOF：清除超时标志(Clear timeout flag)
写入 1 可将 QUADSPI_SR 寄存器中的 TOF 标志清零

位 3 CSMF：清除状态匹配标志(Clear status match flag) 写入 1 可将 QUADSPI_SR 寄存器中的 SMF 标志清零

位 2 保留，必须保持复位值。

位 1 CTCF：清除传输完成标志(Clear transfer complete flag) 写入 1 可将 QUADSPI_SR 寄存器中的 TCF 标志清零

位 0 CTEF：清除传输错误标志(Clear Transfer error flag) 写入 1 可将 QUADSPI_SR 寄存器中的 TEF 标志清零

5.5.5 QUADSPI 数据长度寄存器(QUADSPI_DLR)

QUADSPI data length register

偏移地址：0x0010

复位值：0x0000 0000

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

DL[15：0]													
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 DL[31: 0]：数据长度(Data length)

在间接模式和状态轮询模式下待检索的数据数量(值+1)。对状态轮询模式应使用不 大于 3 的值(表示 4 字节)。

在间接模式下,所有位置 1 表示未定义长度,QUADSPI 将继续传输数据直到到达由 FSIZE 定义的存储器末尾。

0x0000_0000：传输 1 个字节

0x0000_0001 ：传输 2 个字节

0x0000_0002：传输 3 个字节

0x0000_0003：传输 4 个字节

OxFFFF_FFFD：传输 4,294,967,294 (4G-2)个字节

OxFFFF_FFFE：传输 4,294,967,295 (4G-1)个字节

OxFFFF_FFFF：未定义长度-传输所有字节直到到达由 FSIZE 定义的 Flash 的结尾。

如果 FS1ZE = 0x1 F,则读取无限继续下去。

在双闪存模式(DFM = 1)下，即使该位写入“0” ， DL[0]也始终保持为“1”，因此保 证了每次访问均传输偶数个字节。

该字段在内存映射模式(FMODE = 10)下不起作用

仅可在 BUSY = 0 时写入该字段。

5.5.6 QUADSPI 通信配置寄存器(QUADSPI_CCR)

QUADSPI communication configuration register

偏移地址：0x0014 复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
~~DDRM~~	~~DHHC~~	Res.	SIOO	FMODE[1:0]		DMODE		Res.	DCYC[4:0]					ABSIZE	
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
ABMODE		ADSIZE		ADMODE		IMODE		INSTRUCTION[7：0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

~~位 31 DDRM：双倍数据速率模式(Double data rate mode)~~

~~该位为地址、交替字节和数据阶段设置 DDR 模式：~~

~~0：禁止 DDR 模式~~

~~1：使能 DDR 模式~~

~~仅可在 BUSY = 0 时写入该字段。~~

~~位 30 DHHC：DDR 保持(DDR hold)~~

~~DDR 模式下数据输出延迟 1/4 个 QUADSPI 输出时钟周期：~~

~~0：使用模拟延迟来延迟数据输出~~

~~1：数据输出延迟 1/4 个 QUADSPI 输出时钟周期。~~

~~该特性仅在 DDR 模式下激活。~~

~~仅可在 BUSY = 0 时写入该字段。~~

位 29 保留，必须保持复位值。

位 28 SIOO：仅发送指令一次模式
*请参见第 370 页上的第 14.3.11 节:指令仅发送一次。*IMODE = 00 时，该位不起作用。
0：在每个事务中发送指令
1：仅为第一条命令发送指令
仅可在 BUSY = 0 时写入该字段。

位[27:26] FMODE[1:0]：功能模式(Functional mode)
该字段定义 QUADSPI 操作的功能模式：
00：间接写入模式
01：间接读取模式
10：自动轮询模式
11:内存映射模式
如果 DMAEN = 1,则在更改 FMODE 值前，必须禁止相应通道的 DMA 控制器。
仅可在 BUSY = 0 时写入该字段。

位 25:24 DMODE[1:0]：数据模式(Data mode)
该字段定义数据阶段的操作模式：
00：无数据
01：单线传输数据
10：双线传输数据
11:四线传输数据
该字段还定义空指令阶段的操作模式。
仅可在 BUSY = 0 时写入该字段。

位 23 保留，必须保持复位值。

位［22:18］ DCYC［4:0］：空指令周期数(Number of dummy cycles)
该字段定义空指令阶段的持续时间。在 SDR 和 DDR 模式下，它指定 CLK 周期数(0-31)= 仅可在 BUSY = 0 时写入该字段。

位 17:16 ABSIZE［1:0］：交替字节长度(Alternate bytes size)
该位定义交替字节长度：
00： 8 位交替字节
01： 16 位交替字节
10： 24 位交替字节
11： 32 位交替字节
仅可在 BUSY = 0 时写入该字段。

位 15:14 ABMODE［1:0］：交替字节模式(Alternate bytes mode)
该字段定义交替字节阶段的操作模式：
00：无交替字节
01：单线传输交替字节
10：双线传输交替字节
11:四线传输交替字节
仅可在 BUSY = 0 时写入该字段。

位 13:12 ADSIZE［1:0］：地址长度(Address size)
该位定义地址长度：
00： 8 位地址
01： 16 位地址
10： 24 位地址
11： 32 位地址
仅可在 BUSY = 0 时写入该字段。

位 11:10 ADDRESS［1:0］： 地址模式(Address mode)
该字段定义地址阶段的操作模式：
00：无地址
01：单线传输地址
10：双线传输地址
11:四线传输地址
仅可在 BUSY = 0 时写入该字段。

位 9:8 IMODE［1:0］： 指令模式（Instruction mode)
该字段定义指令阶段的操作模式：
00：无指令
01：单线传输指令
10：双线传输指令
11:四线传输指令
仅可在 BUSY = 0 时写入该字段。

位 7:0 INSTRUCTION)7: 0］： 指令（Instruction)
指定要发送到外部 SPI 设备的指令。
仅可在 BUSY = 0 时写入该字段。

5.5.7 QUADSPI 地址寄存器(QUADSPI_AR)

QUADSPI address register

偏移地址：0x0018 复位值：0x0000 0000

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

位 31:0 ADDRESS[31:0]：地址（Address)
指定要发送到外部 Flash 的地址。
BUSY = 0 或 FMODE = 11（内存映射模式）时，将忽略写入该字段。
在双闪存模式下，由于地址始终为偶地址，ADDRESS[0]自动保持为"0"=

5.5.8 QUADSPI 交替字节寄存器(QUADSPI_ABR)

QUADSPI alternate bytes registers

偏移地址：0x001C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ALTERNATE[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
ALTERNATED 5:0]															

rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 ALTERNATE[31: 0]：交替字节(Alternate Bytes)
指定要在地址后立即发送到外部 SPI 设备的可选数据。
仅可在 BUSY = 0 时写入该字段。

5.5.9 QUADSPI 数据寄存器(QUADSPI_DR)

QUADSPI data register

偏移地址：0x0020 复位值：0x0000 0000

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

位 31:0 DATA［31:0］：数据（Data）
指定要与外部 SPI 设备交换的数据。
在间接写入模式下，写入该寄存器的数据在数据阶段发送到 Flash,在此之前则存储于 FIFO,如果 FIFO 太满，将暂停写入，直到 FIFO 具有足够的空间接受要写入的数据才 继续。
在间接模式下，读取该寄存器可获得（通过 FIFO）已从 Flash 接收的数据。如果 FIFO 所含字节数比读取操作要求的字节数少并且 BUSY=1,将暂停读取，直到足够的数据出 现或传输完成（不分先后）才继续。
在自动轮询模式下，该寄存器包含最后从 Flash 读取的数据（未进行屏蔽）。
支持对该寄存器进行字、半字以及字节访问。在间接写入模式下，字节写入将在 FIFO 中增加 1 个字节，半字写入增加 2 个，而字写入则增加 4 个。类似地，在间接读取模式 下，字节读取将擦除 FIFO 中的 1 个字节，半字读取擦除 2 个，而字读取则擦除 4 个。 间接模式下的访问必须与此寄存器的最低位对齐：字节读取必须读取 DATA0O］而半字 读取必须读取 DATA［15:0］。

5.5.10 QUADSPI 轮询状态屏蔽寄存器(QUADSPI _PSMKR)

QUADSPI polling status mask register

偏移地址：0x0024

复位值：0x0000 0000

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

位 31:0 MASK［31:0］：状态屏蔽（Status mask）

对在轮询模式下接收的状态字节进行屏蔽
对于位 n：
0：屏蔽在自动轮询模式下所接收数据的位 n,在匹配逻辑中不考虑其值 1：不屏蔽在自动轮询模式下所接收数据的位 n,在匹配逻辑中考虑其值 仅可在 BUSY = 0 时写入该字段。

5.5.11 QUADSPI 轮询状态匹配寄存器(QUADSPI _PSMAR)

QUADSPI polling status match register

偏移地址：0x0028

复位值：0x0000 0000

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

位 31:0 MATCH［31: 0］：状态匹配(Status match)
该值将与屏蔽状态寄存器比较以进行匹配。
仅可在 BUSY = 0 时写入该字段。

5.5.12 QUADSPI 轮询间隔寄存器(QUADSPI _PIR)

QUADSPI polling interval register

偏移地址：0x002C

复位值：0x0000 0000

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

位 31:16 保留，必须保持复位值。

位 15:0 INTERVALS 5: 0]：轮询间隔(Polling interval) 自动轮询阶段读取操作之间的 CLK 周期数。仅可在 BUSY = 0 时写入该字段。

5.5.13 QUADSPI 低功耗超时寄存器(QUADSPIJ~PTR)

QUADSPI low-power timeout register

偏移地址：0x0030

复位值：0x0000 0000

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

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

位 31:16 保留，必须保持复位值.

位 15:0 TIMEOUT[15: 0]：超时时长(Timeout period)
在内存映射模式下，每次访问结束后，QUADSPI 将预取后续字节并将其存放在 FIFO 中。该字段指示在 FIFO 写满后，QUADSPI 等待多少个 CLK 时钟周期才让 nCS 升至高 电平将 Flash 置为低功耗状态。
仅可在 BUSY = 0 时写入该字段

5.5.14 QUADSPI 寄存器映射

偏移	寄存器	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
0x0	QUADSPI_CR	PRESCALER[7:0]								PMM	APMSS	Res.	TOIE	SMIE	TFIE	TCIE	TEIE	Res.	Res.	Res.	FTHRES[4:0]					FSEL	DFM	Res.	SSHIF	TCEN	DMAEN	ABORT	EN
	Reset value	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0				0	0	0	0	0	0	0		0	0	0	0	0
0x04	QUADSPI_DCR	RES.											FSIZE[4:0]										CSHT			RES.							CKMODE
	Reset value												0	0	0	0	0						0	0	0								0
0X08	QUADSPI_SR	RES.																FLEVEL[6:0]								Res.	Res.	BUSY	TOF	SMF	FTF	TCF	TEF
																		0	0	0	0	0	0	0				0	0	0	0	0	0
0X0C	QUADSPI_FCR	RES.																											CTOF	CSMF	RES	CTCF	CTEF
																													0	0		0	0
0X10	QUADSPI_DLR	DL[31:0]																															
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0X14	QUADSPI_CCR	DDRM	DHHC		SIOO	FMODE[1:0]		DMODE[1:0]		Res.	DCYC[4:0]					ABSIZE[1:0]		ABMODE[1:0]		ADSIZE[1:0]		ADMODE[1:0]		IMODE[1:0]		INSTRUCTION[7:0]							
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0X18	QUADSPI_AR	ADDRESS[31:0]																															
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1c	QUADSPI_ABR	ALTERNATE[31:0]																															
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	QUADSPI_DR	DATA[31:0]																															
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	QUADSPI_ PSMKR	MASK[31:0]																															
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	QUADSPI_ PSMAR	MATCH[31:0]																															
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2c	QUADSPI_PIR	RES.																INTERVAL[15:0]															
																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	QUADSPI_ LPTR	RES.																TIMEOUT[15:0]															
																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6 灵活的静态存储器控制器(FSMC)

6.1 FSMC 功能描述

FSMC 模块能够与同步或异步存储器，它的主要作用是：

- 将 AHB 传输信号转换到适当的外部设备协议
- 满足访问外部设备的时序要求

所有的外部存储器共享控制器输出的地址、数据和控制信号，每个外部设备可以通过一个唯一的片选信号加以区分。FSMC 在任一时刻只访问一个外部设备

FSMC 具有下列主要功能：

- 具有静态存储器接口的器件包括：
  - 静态随机存储器(SRAM)
  - 只读存储器(ROM)
  - NOR 闪存/OneNAND Flash
  - PSRAM(4 个存储器块)
- 8 或 16 位数据总线
- 每一个存储器块都有独立的片选控制
- 每一个存储器块都可以独立配置
- PSRAM 和 SRAM 器件使用的写使能和字节选择输出
- 将 32 位的 AHB 访问请求，转换到连续的 16 位或 8 位的，对外部 16 位或 8 位器件的访问

通常在系统复位或上电时，应该设置好所有定义外部存储器类型和特性的 FSMC 寄存器，并保持它们的内容不变；也可以随时改变这些设置。

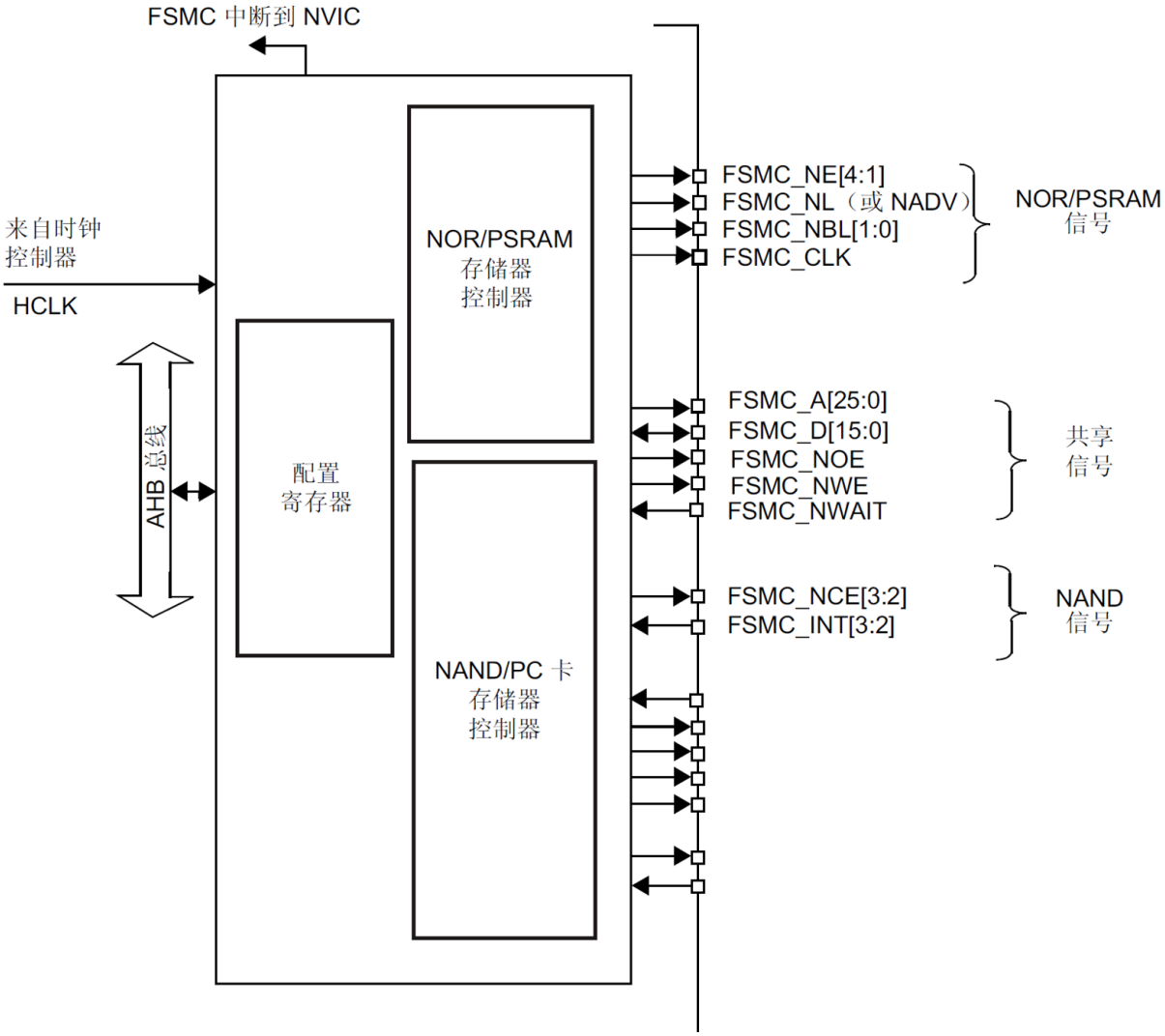
6.2 框图

FSMC 包含以下主要模块：

- AHB 接口(包含 FSMC 配置寄存器)
- NOR 闪存和 PSRAM 控制器
- 外部设备接口

FSMC 框图如下：

图 1 FSMC 框图



6.3 AHB 接口

AHB 接口为内部 CPU 和其它总线控制设备访问外部静态存储器提供了通道。

AHB 操作被转换到外部设备的操作。当选择的外部存储器的数据通道是 16 或 8 位时，在 AHB 上的 32 位数据会被分割成连续的 16 或 8 位的操作。

AHB 时钟(HCLK)是 FSMC 的参考时钟。

6.3.1 支持的存储器和操作

一般的操作规则

请求 AHB 操作的数据宽度可以是 8 位、16 位或 32 位，而外部设备则是固定的数据宽度，此时需要保障实现数据传输的一致性。

因此，FSMC 执行下述操作规则：

- AHB 操作的数据宽度与存储器数据宽度相同：无数据传输一致性的问题。

● AHB 操作的数据宽度大于存储器的数据宽度：此时 FSMC 将 AHB 操作分割成几个连续的较小数据宽度的存储器操作，以适应外部设备的数据宽度。

● AHB 操作的数据宽度小于存储器的数据宽度：

依据外部设备的类型，异步的数据传输有可能不一致。

— 与具有字节选择功能的存储器(SRAM、ROM、PSRAM 等)进行异步传输时，FSMC 执行读写操作并通过它的字节通道 BL[1:0]访问正确的数据。

— 与不具有字节选择功能的存储器(NOR 和 16 位 NAND 等)进行异步传输时，即需要对 16 位宽的闪存存储器进行字节访问；显然不能对存储器进行字节模式访问(只允许 16 位的数据传输)，因此：

a. 不允许进行写操作

b.允许读取事务。会读取所有存储器字节，并将丢弃无用的存储器字节。NBL[1:0] 在读取事务期间保持低电平。

配置寄存器

FSMC 由一组寄存器进行配置。6.6 节详细描述了 NOR 闪存和 PSRAM 控制器寄存器。

6.4外部设备地址映像

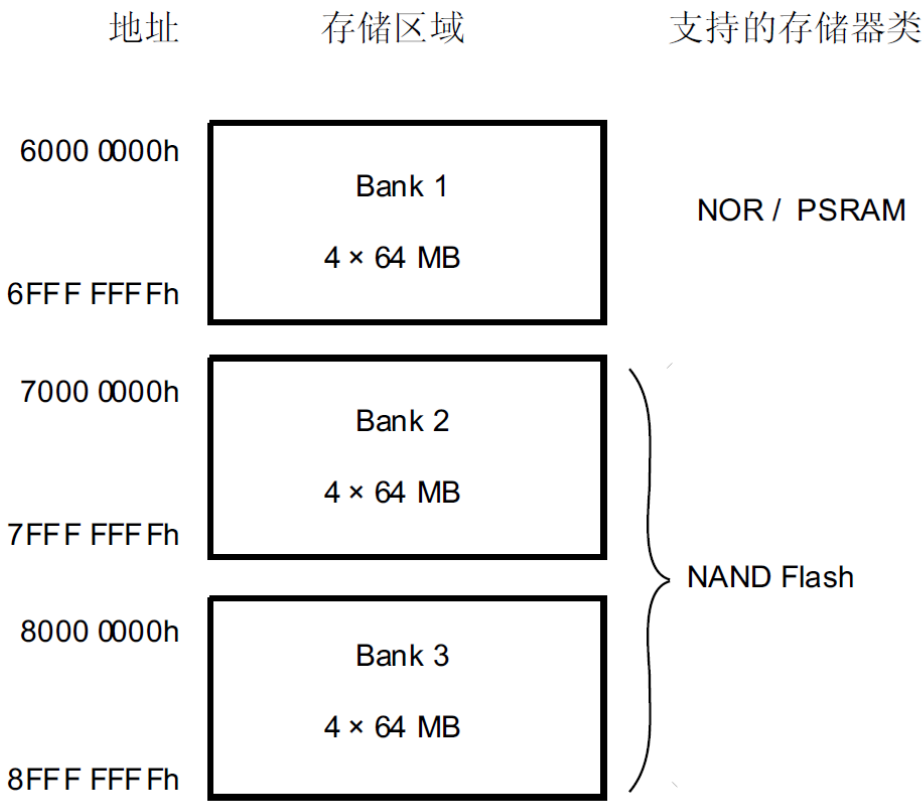
从 FSMC 的角度看，可以把外部存储器划分为固定大小为 256M 字节的四个存储块，见下图。

● 存储块 1 用于访问最多 3 个 NOR 闪存或 PSRAM 存储设备。这个存储区被划分为 3 个 NOR/PSRAM 区并有 3 个专用的片选。

● 存储块 2 和 3 用于访问 NAND 闪存设备，每个存储块连接一个 NAND 闪存。

每一个存储块上的存储器类型是由用户在配置寄存器中定义的。

图 2 FSMC 存储块



6.4.1 NOR 和 PSRAM 地址映像

HADDR[27:26]位用于选择四个存储块之一：

表 1 NOR/PSRAM 存储块选择

HADDR[27:26][1]	选择的存储块
00	存储块 1 NOR/PSRAM1
01	存储块 1 NOR/PSRAM2
10	存储块 1 NOR/PSRAM3
11	存储块 1 NOR/PSRAM4

(1) HADDR 是需要转换到外部存储器的内部 AHB 地址线。

HADDR[25:0]包含外部存储器地址。HADDR 是字节地址，而存储器访问不都是按字节访问，因此接到存储器的地址线依存储器的数据宽度有所不同，如下表： 数据宽度[1]

表 2 外部存储器地址

数据宽度[1]	连到存储器的地址线	最大访问存储空间（位）
8 位	HADDR[25:0]与 FSMC_A[25:0]对应相连	64M 字节 x8=512M 位
16 位	HADDR[25:0]与 FSMC_A[240]对应相连， HADDR[0]未连接	64M 字节/2 x 16=512M 位

对于 16 位宽度的外部存储器，FSMC 将在内部使用 HADDR[25:1]产生外部存储器的地址 FSMC_A[24:0]。不论外部存储器的宽度是多少(16 位或 8 位)，FSMC_A[0]始终应该连到外部存储器的 地址线 A[0]。

NOR 闪存和 PSRAM 的非对齐访问支持

每个 NOR 闪存或 PSRAM 存储器块都可以配置成支持非对齐的数据访问。

在存储器一侧，依据访问的方式是异步模式。

- **异步模式**：这种情况下，只要每次访问都有准确的地址，完全支持非对齐的数据访问。

如果存储器的非对齐访问模式不能设置为与 AHB 相同的模式，应该通过 FSMC 配置寄存器的相应位禁止非对齐访问，并把非对齐的访问请求分开成两个连续的访问操作。

6.5 NOR 闪存和 PSRAM 控制器

FSMC 可以产生适当的信号时序，驱动下述类型的存储器：

- 异步 SRAM 和 ROM
  - 8 位
  - 16 位
  - 32 位
- PSRAM(Cellular RAM)
  - 异步模式
- NOR 闪存
  - 异步模式
  - 复用模式或非复用模式

FSMC 对每个存储块输出一个唯一的片选信号 NE[4:1]，所有其它的(地址、数据和控制)信号则是共享的。

每个存储块都有专门的寄存器控制(见 6.7 节)。

可编程的存储器参数包括访问时序(见下表)、是否支持非对齐数据存取和等待周期管理(只针对突发模式下访问 PSRAM 和 NOR 闪存)。

表 3 可编程的 NOR/PSRAM 访问参数

参数	功能	访问方式	单位	最小	最大
地址建立时间	地址建立阶段的时间	异步	AHB 时钟周期(HCLK)	1	16
地址保持时间	地址保持阶段的时间	异步，复用 I/O	AHB 时钟周期(HCLK)	2	16
数据建立时间	数据建立阶段的时间	异步	AHB 时钟周期(HCLK)	2	256
总线恢复时间	总线恢复阶段的时间	异步	AHB 时钟周期(HCLK)	1	16

6.5.1 外部存储器接口信号

表 4 、表 5、表 6 列出了与 NOR 闪存和 PSRAM 接口的典型信号。

注： 具有前缀“N”的信号表示低有效信号

NOR 闪存，非复用接口

表 4 非复用信号的 NOR 闪存接口

FSMC 信号名称	信号方向	功能
A[25:0]	输出	地址总线

D[15:0]	输入/输出	双向数据总线
NE[x]	输出	片选，x=1，2，3，4
NOE	输出	输出使能
NWE	输出	写使能

NOR 闪存存储器是按 16 位的字寻址，最大容量达 64M 字节(26 条地址线)。

NOR 闪存，复用接口

表 5 复用 NOR 闪存接口

FSMC 信号名称	信号方向	功能
A[25:16]	输出	地址总线
AD[15:0]	输入/输出	16 位服用的，双向数据总线
NE[x]	输出	片选，x=1，2，3，4
NOE	输出	输出使能
NWE	输出	写使能
NL(=NADV)	输出	锁存使能（某些 NOR 闪存器件命名该信号为地址有效，NADV）

NOR 闪存存储器是按 16 位的字寻址，最大容量达 64M 字节(26 条地址线)。

PSRAM

表 6 非复用信号的 PSRAM 接口

FSMC 信号名称	信号方向	功能
A[25:0]	输出	地址总线
D[15:0]	输入/输出	双向数据总线
NE[x]	输出	片选，x = 1...4 (PSRAM 称其为 NCE(Cellular RAM 既 CRAM))
NOE	输出	输出使能
NWE	输出	写使能
NL(=NADV)	输出	地址有效(存储器信号名称为：NADV)
NBL[1]	输出	高字节使能(存储器信号名称为：NUB)
NBL[0]	输出	低字节使能(存储器信号名称为：NLB)

PSRAM 存储器是按 16 位的字寻址，最大容量达 64M 字节(26 条地址线)。

6.5.2 支持的存储器及其操作

下表列出了支持的存储器、访问模式和操作方式，FSMC 不支持阴影部分的操作方式。

表 7 FSMC 支持的 NOR 闪存/PSRAM 存储器和操作方式

存储器	模式	读/写	AHB 数据宽度	存储器数据宽度	是否支持	注释
NOR 闪存（总线复用和非总线复用）	异步	读	8	16	支持	
	异步	写	8	16	不支持	
	异步	读	16	16	支持	
	异步	写	16	16	支持	
	异步	读	32	16	支持	分 2 次 FSMC 访问

	异步	写	32	16	支持	分 2 次 FSMC 访问
	异步页	读	--	16	不支持	不支持这种模式
PSRAM（总线复用和非总线复用）	异步	读	8	16	支持	
	异步	写	8	16	支持	使用字节信号 NBL[1:0]
	异步	读	16	16	支持	
	异步	写	16	16	支持	
	异步	读	32	16	支持	分 2 次 FSMC 访问
	异步	写	32	16	支持	分 2 次 FSMC 访问
	异步页	读	--	16	不支持	不支持这种模式
SRAM 和 ROM	异步	读	8/16/32	8/16	支持	使用字节信号 NBL[1:0]
	异步	写	8/16/32	8/16	支持	使用字节信号 NBL[1:0]

6.5.3 时序规则

信号同步

- 所有的控制器输出信号在内部时钟(HCLK)的上升沿变化

6.5.4 NOR 闪存和 PSRAM 控制器时序图

异步静态存储器 异步静态存储器(NOR 闪存和 PSRAM)

- 所有信号由内部时钟 HCLK 保持同步，但该时钟不会输出到存储器；
- FSMC 始终在片选信号 NE 失效前对数据线采样，这样能够保证符合存储器的数据保持时序(片选失效至数据失效的间隔，通常最小为 10ns)；
- 当设置了扩展模式，可以在读和写时混合使用模式 A、B、C 和 D(例如，允许以模式 A 进行读，而以模式 B 进行写)。

模式 1 —— SRAM/CRAM

图 162 模式 1 读操作

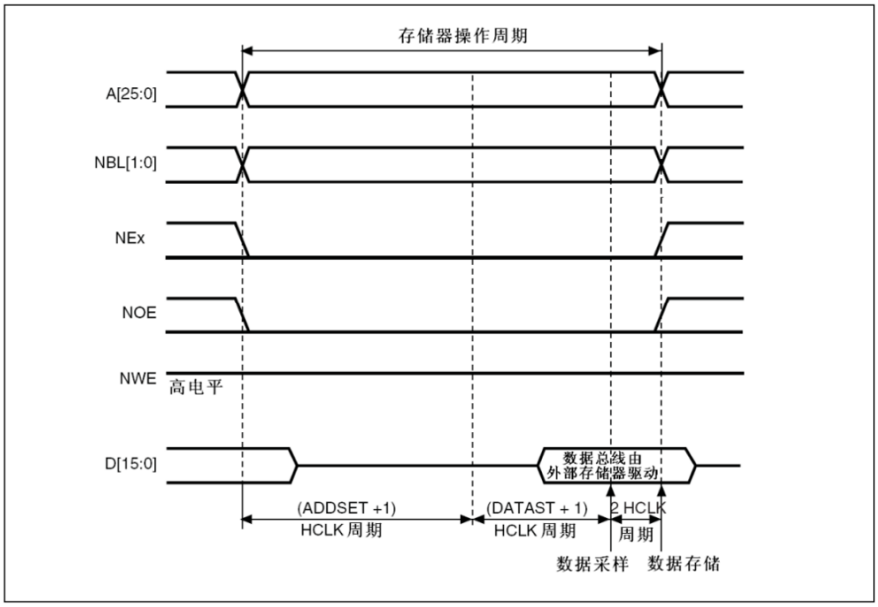
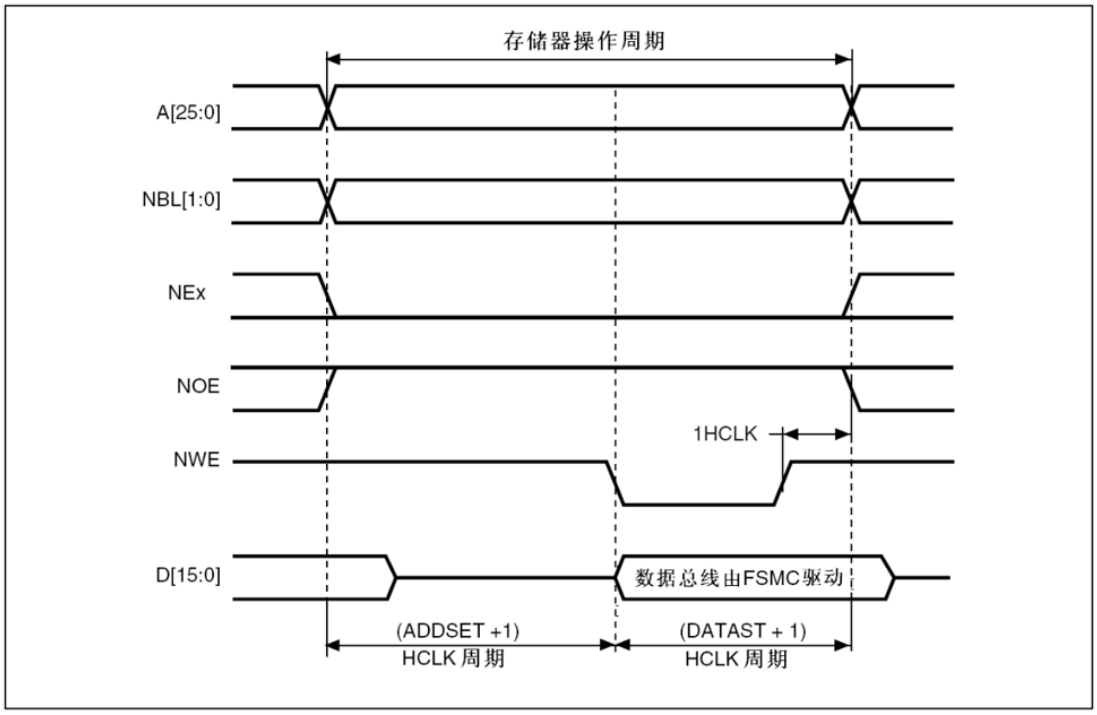


图 163　模式 1 写操作



在写操作的最后一个 HCLK 周期可以保证 NWE 上升沿后地址和数据的保持时间，因为存在这个 HCLK 周期，DATAST 的数值必须大于 0(DATAST > 0)。

表 95 FSMC_BCRx 位域(模式 1)

位编号	位名称	设置的数值
31-15	保留	0x0000
14	EXTMOD	0x0

13-7	保留	0x0
6	FACCEN	无关
5-4	MWID	需要时设置
3-2	MTYP	需要时设置，不包含 0x02（NOR 闪存）
1	MUXEN	0X0
0	MBKEN	0X1

表 96 FSMC_BTRx 位域(模式 1)

位编号	位名称	设置的数值
31-16	保留	0x0000
15-8	DATAST	操作的第 2 个阶段的长度，写操作为（DATAST+1 个 HCLK 周期）， 读操作为 （DATAST+3 个 HCLK 周期）。这个域不能为 0,至少为 1。
7-4	保留	0x0
3-0	ADDSET	操作的第 1 个阶段的长度（ADDSET+1 个 HCLK 周期）。

模式 A —— SRAM/PSRAM(CRAM) OE 翻转

图 164 模式 A 读操作

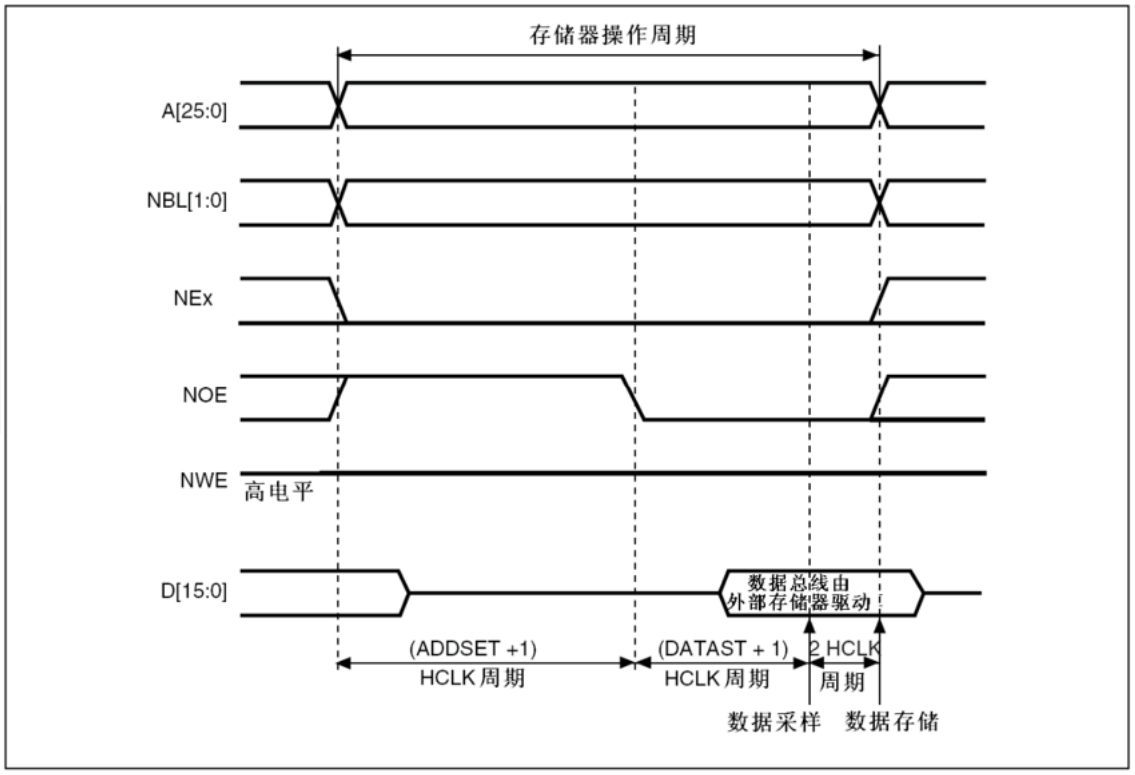
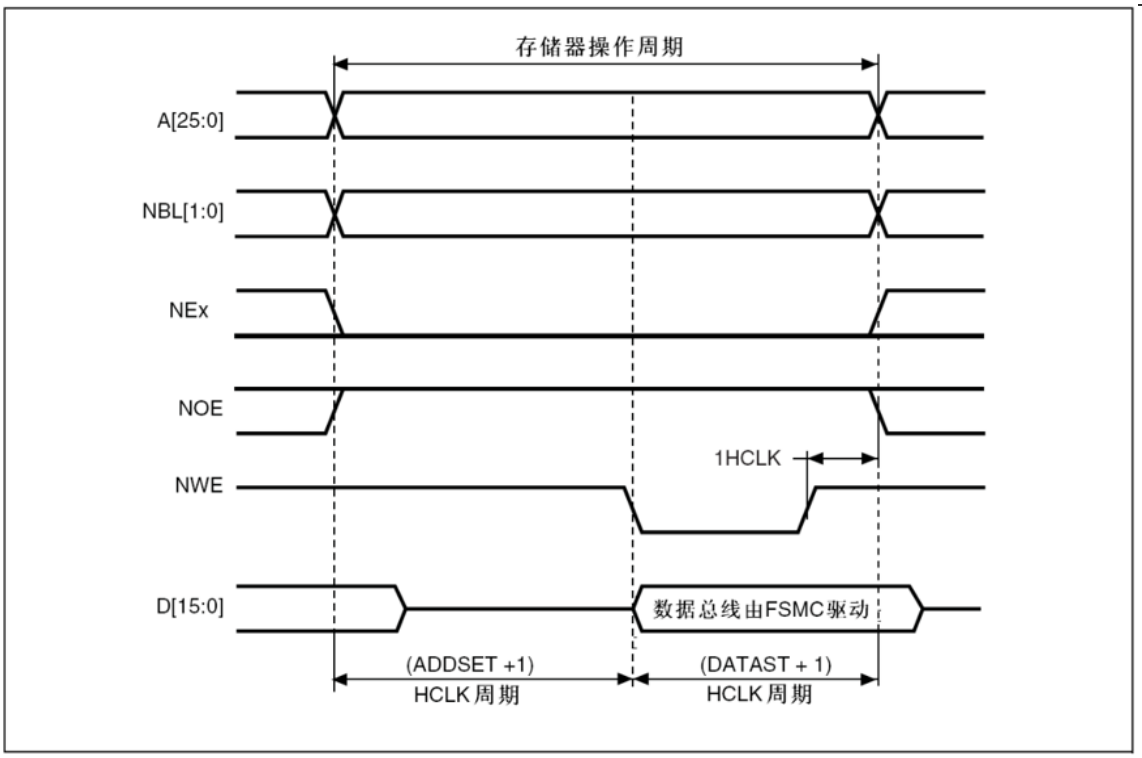


图 165 模式 A 写操作



模式 A 与模式 1 的区别是 NOE 的变化和相互独立的读写时序。

表 97 FSMC_BCRx 位域(模式 A)

位编号	位名称	设置的数值
31-16	保留	0x0000
15	保留	0x0
14	EXTMOD	0x01
13-10	保留	0x0
9-7	保留	0x0
6	FACCEN	无关
5-4	MWID	需要时设置
3-2	MTYP	需要时设置，不包含 10（NOR 闪存）
1	MUXEN	0X0
0	MBKEN	0X1

表 98 FSMC_BTRx 位域(模式 A)

位编号	位名称	设置的数值
31-30	保留	0x0
29-28	ACCMOD	0x0
27-16	保留	0x000
15-8	DATAST	读操作的第 2 个阶段的长度（DATAST+3 个 HCLK 周期）。这个域不能为 0（至少为 1）
7-4	保留	0x0
3-0	ADDSET	读操作的第 1 个阶段的长度（ADDSET+1 个 HCLK 周期）

表 99 FSMC_BWTRx 位域(模式 A)

位编号	位名称	设置的数值
31-30	保留	0x0
29-28	ACCMOD	0x0
27-16	保留	0x000
15-8	DATAST	写操作的第 2 个阶段的长度（DATAST+1 个 HCLK 周期）。这个域不能为 0（至少为 1）
7-4	保留	0x0
3-0	ADDSET	写操作的第 1 个阶段的长度（ADDSET+1 个 HCLK 周期）。

模式 2/B —— NOR 闪存

图 166 模式 2/B 读操作

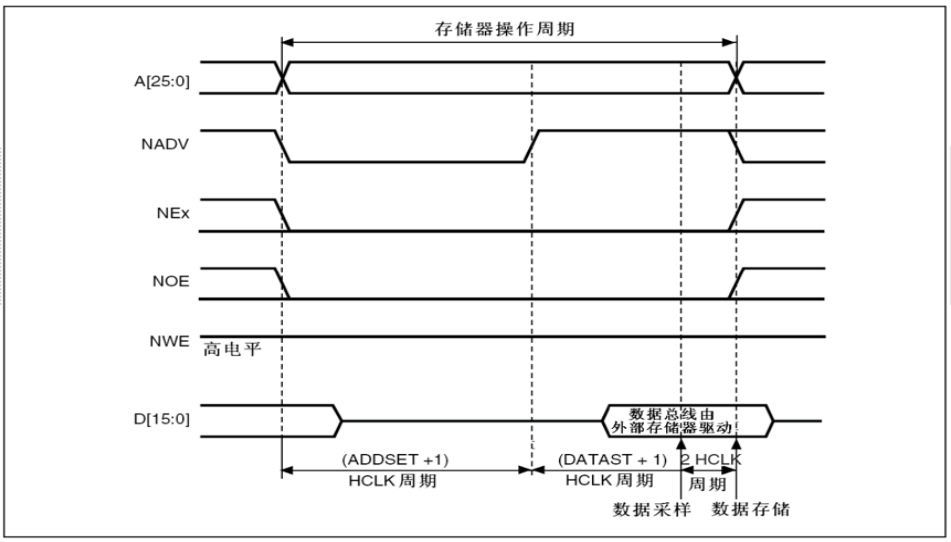


图 167 模式 2 写操作

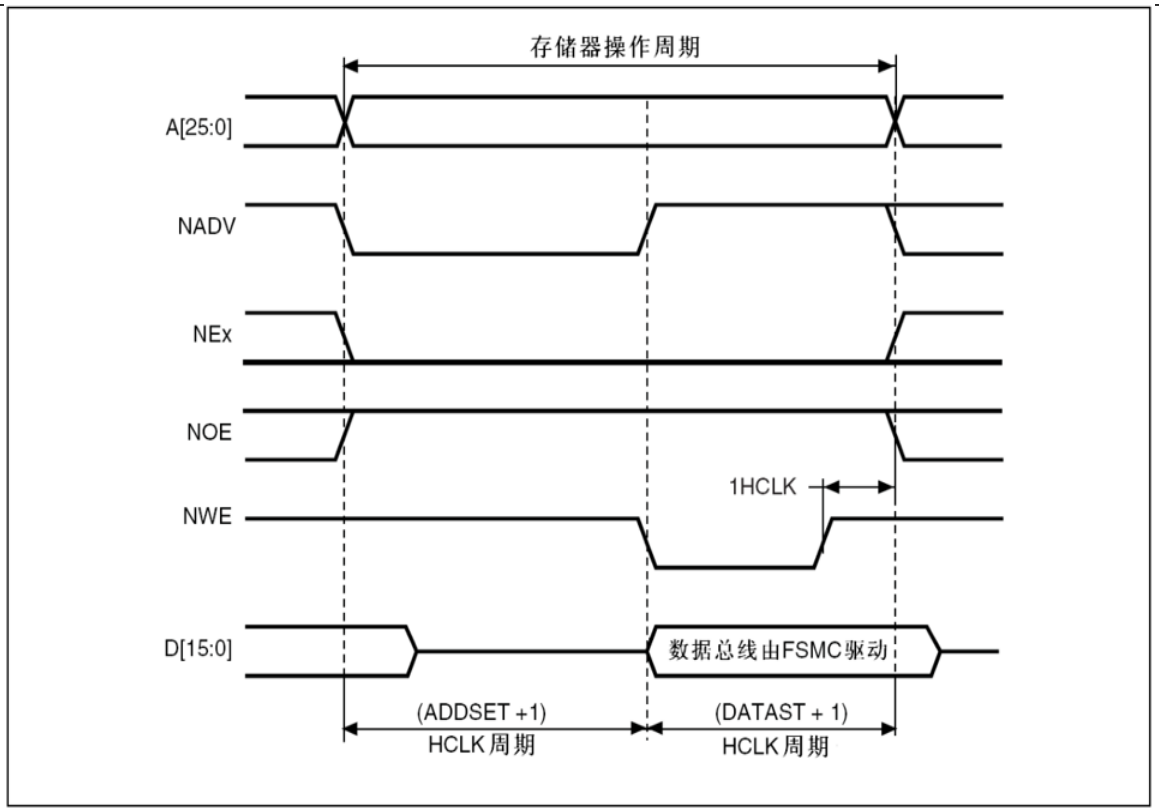
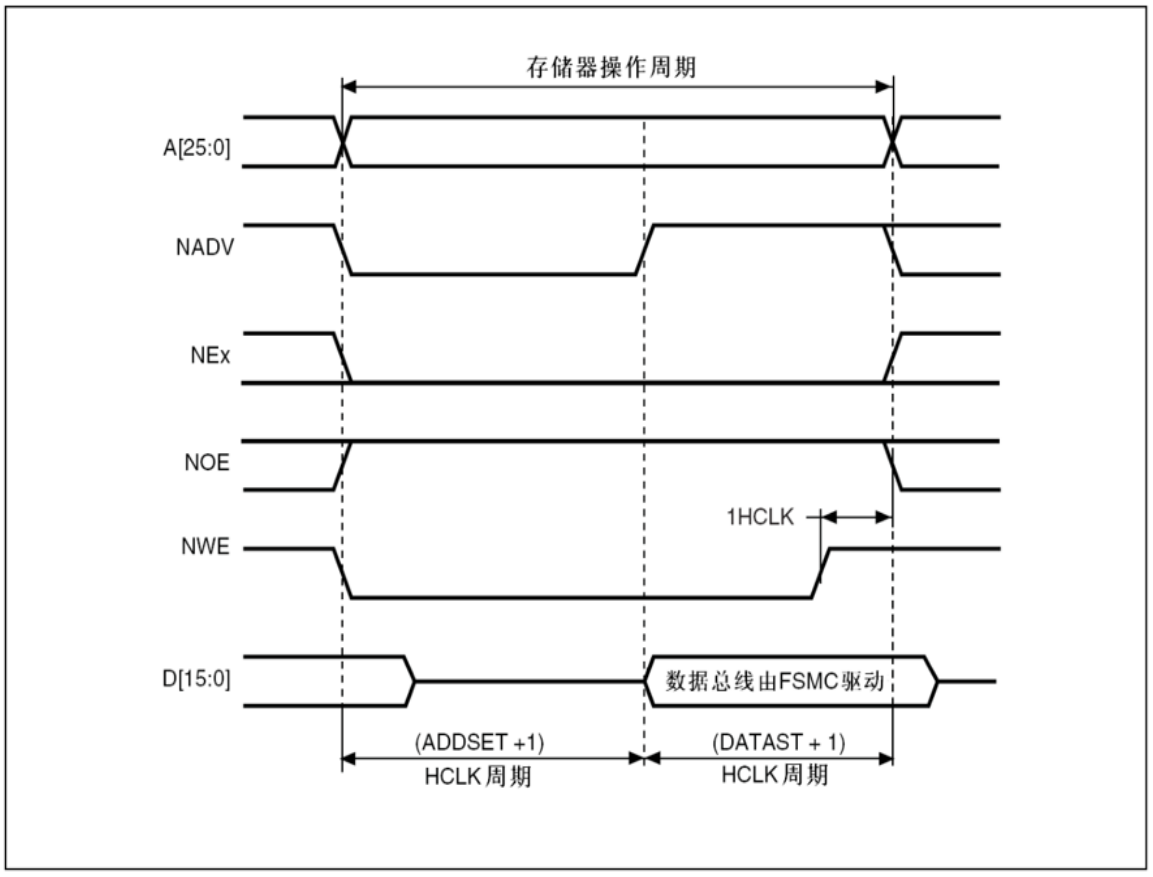

存储器操作周期
A[25:0]
NADV
NEx
NOE
1HCLK
NWE
D[15:0]
数据总线由FSMC驱动
(ADDSET +1)
HCLK 周期
(DATAST + 1)
HCLK 周期

图 168　模式 B 写操作


存储器操作周期
A[25:0]
NADV
NEx
NOE
1HCLK
NWE
D[15:0]
数据总线由FSMC驱动
(ADDSET +1)
HCLK 周期
(DATAST + 1)
HCLK 周期

模式 2/B 与模式 1 相比较，不同的是 NADV 的变化，且在扩展模式下(模式 B)读写时序相互独立。

位编号	位名称	设置的数值
31-15		0x0000
14	EXTMOD	模式 B:0x01　模式 2 ： 0x0
13-10		0x0
9-7		0x0
6	FACCEN	0x1
5-4	MWID	需要时设置
3-2	MTYP	10（NOR 闪存）
1	MUXEN	0X0
0	MBKEN	0X1

表 100 FSMC_BCRx 位域(模式 2/B)

位编号	位名称	设置的数值
31-15	保留	0x0000
14	EXTMOD	模式 B： 0x1；模式 2： 0x0
13-10	保留	0x0
9	WAITPOL	仅当位 15 为 T 时有意义。
8	BURSTEN	0x0
7	保留	-
6	FACCEN	0x1
5-4	MWID	需要时设置
3-2	MTYP	10（NOR 闪存）
1	MUXEN	0x0
0	MBKEN	0x1

表 101 FSMC_BTRx 位域(模式 2/B)

位编号	位名称	设置的数值
31-30	保留	0x0
29-28	ACCMOD	如果使用了扩展模式，设置为 0x1
27-16	保留	0x000
15-8	DATAST	读操作的第 2 个阶段的长度（DATAST+3 个 HCLK 周期）。这个域不能为 0（至少为 1）。
7-4	保留	0x0
3-0	ADDSET	读操作的第 1 个阶段的长度（ADDSET+1 个 HCLK 周期）。

表 102 FSMC_BWTRx 位域(模式 2/B)

位编号	位名称	设置的数值
31-30	保留	0x0
29-28	ACCMOD	如果使用了扩展模式，设置为 0x1
27-16	保留	0x000
15-8	DATAST	写操作的第 2 个阶段的长度（DATAST+1 个 HCLK 周期）。这个域不能为 0（至少为 1）。
7-4	保留	0x0
3-0	ADDSET	写操作的第 1 个阶段的长度（ADDSET+1 个 HCLK 周期）。

注： 只有当设置了扩展模式时 (模式 B) ， FSMC_BWTRx 才有效，否则该寄存器的内容不起作用。

模式 C —— NOR 闪存 - OE

图 169 模式 C 读操作

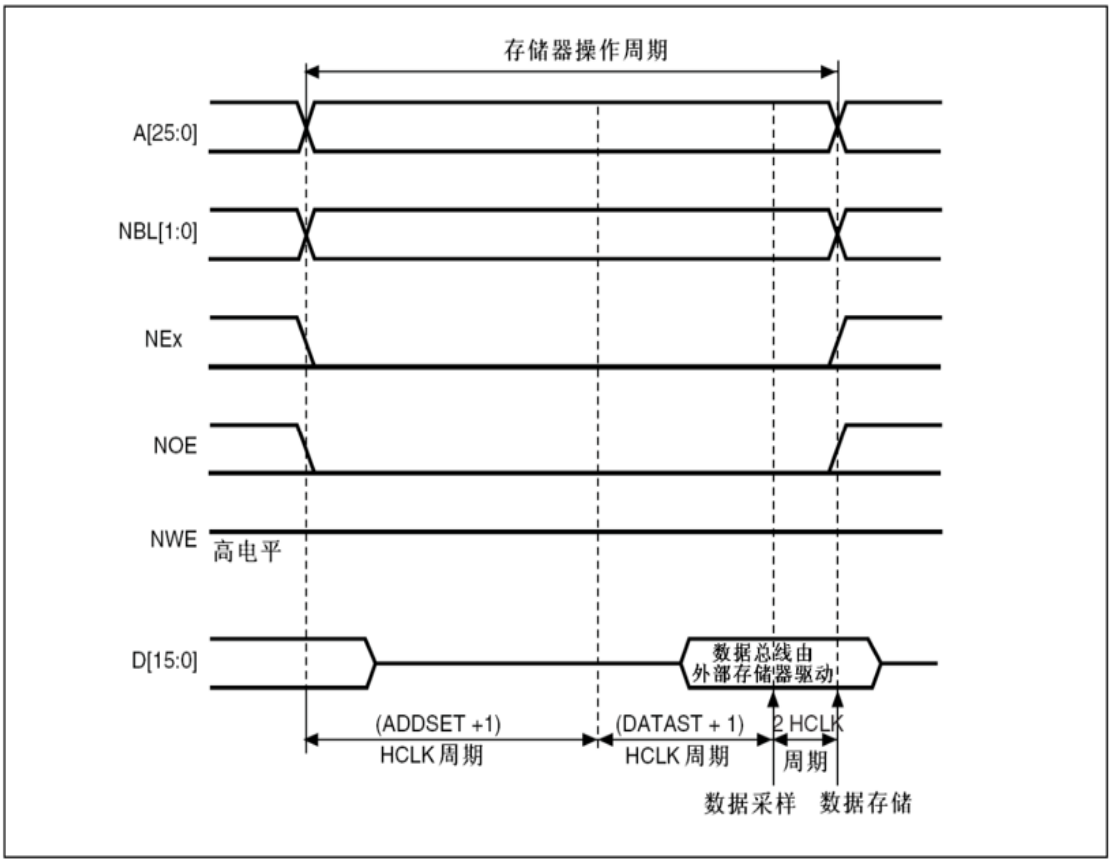
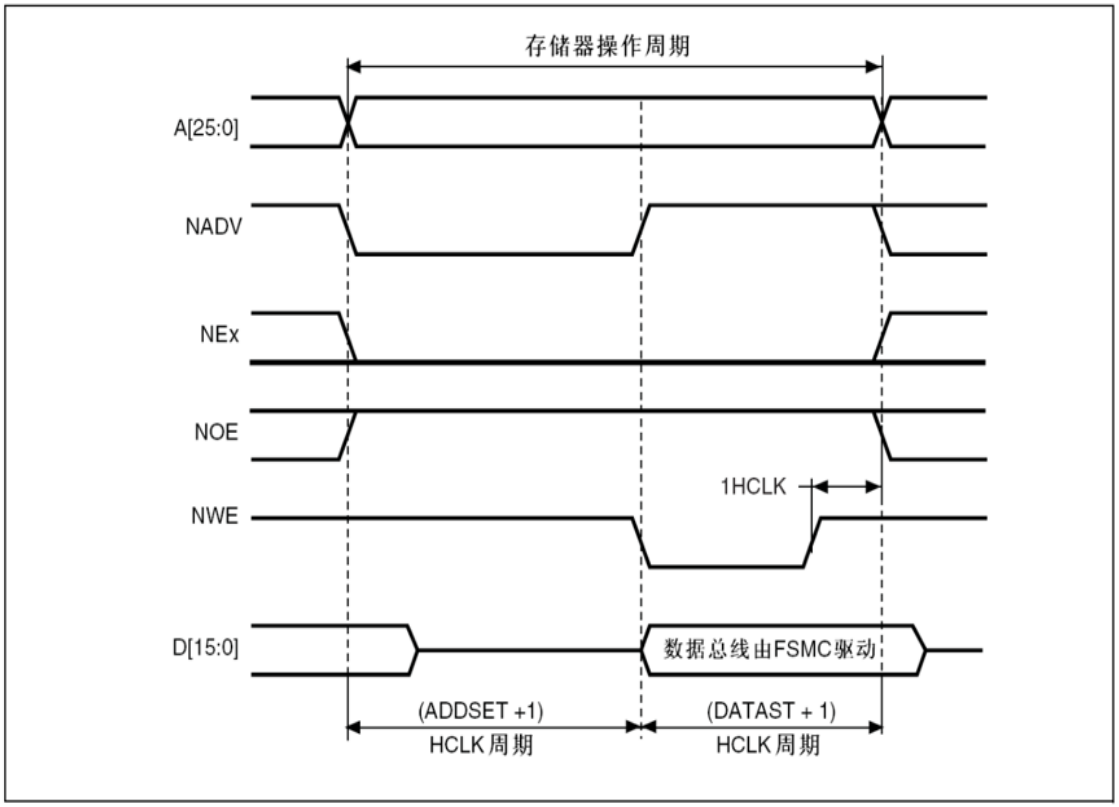


图 170 模式 C 写操作



模式 C 与模式 1 不同的是，NOE 和 NADV 的翻转变化，以及独立的读写时序。

表 103 FSMC_BCRx 位域(模式 C)

位编号	位名称	设置的数值
31-15		0x0000
14	EXTMOD	0x01
13-10		0x0
9-7	RESERVED	-
6	FACCEN	0x1
5-4	MWID	需要时设置
3-2	MTYP	0x2（NOR 闪存）
1	MUXEN	0X0
0	MBKEN	0X1

表 104 FSMC_BTRx 位域(模式 C)

位编号	位名称	设置的数值
31-30	保留	0x0
29-28	ACCMOD	0x2
27-16	保留	0x000
15-8	DATAST	读操作的第 2 个阶段的长度（DATAST+3 个 HCLK 周期）。这个域不能为 0（至少为 1）。

7-4	保留	0x0
3-0	ADDSET	读操作的第 1 个阶段的长度（ADDSET+1 个 HCLK 周期）。

表 105 FSMC_BWTRx 位域(模式 C)

位编号	位名称	设置的数值
31-30	保留	0x0
29-28	ACCMOD	0x2
27-16	保留	0x000
15-8	DATAST	写操作的第 2 个阶段的长度（DATAST+1 个 HCLK 周期）。这个域不能为 0（至少为 1）。
7-4	保留	0x0
3-0	ADDSET	写操作的第 1 个阶段的长度（ADDSET+1 个 HCLK 周期）。

模式 D —— 带地址扩展的异步操作

图 171 模式 D 读操作

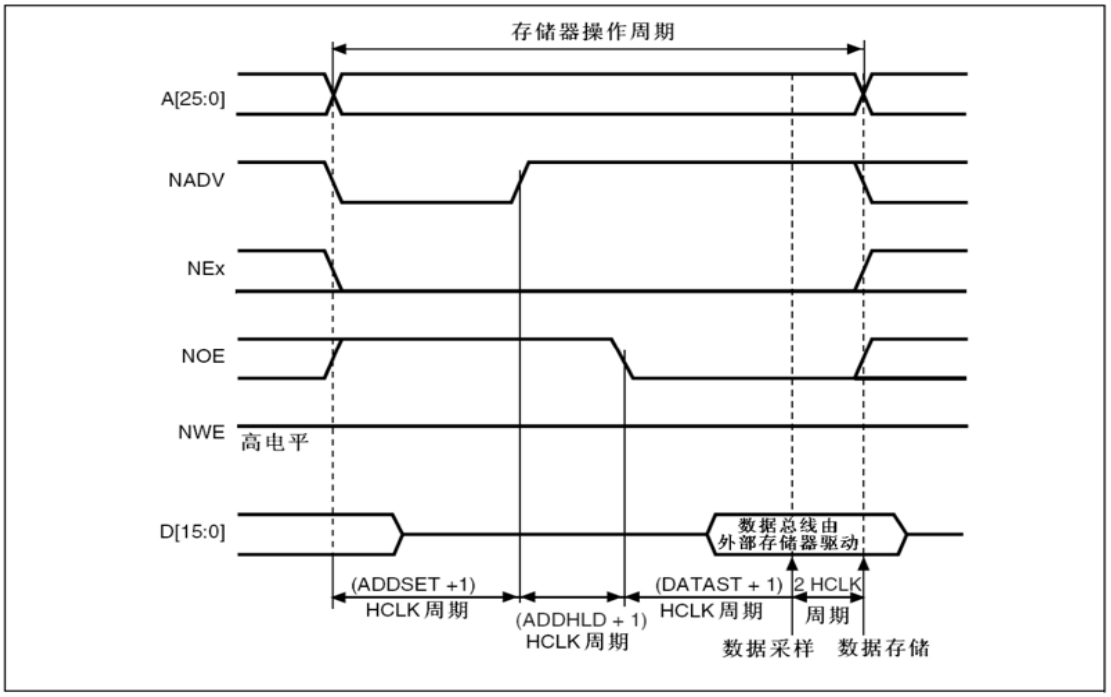
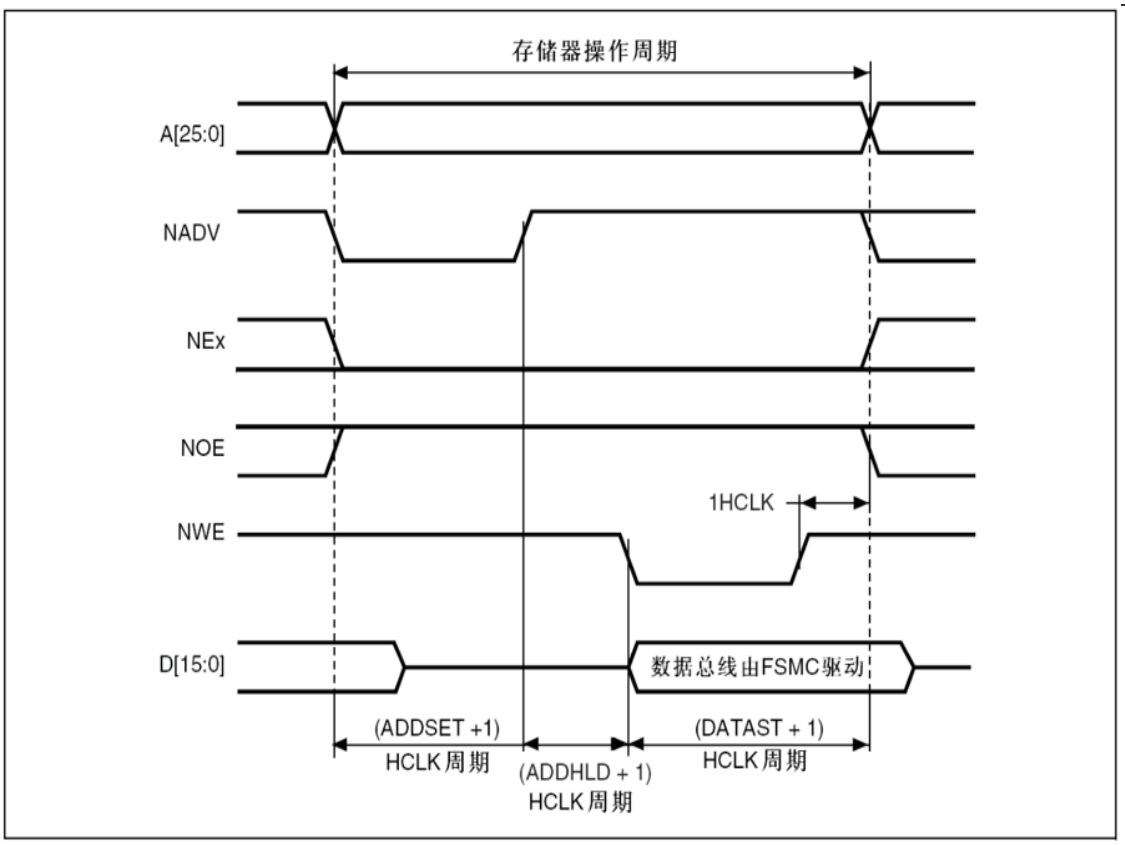


图 172 模式 D 写操作



模式 D 与模式 1 不同的是 NADV 的翻转变化，NOE 的翻转出现在 NADV 翻转之后，并且具有独立的读写时序。

表 106 FSMC_BCRx 位域(模式 D)

位编号	位名称	设置的数值
31-15		0x0000
14	EXTMOD	0x01
13-10		0x0
9-7	RESERVED	-
6	FACCEN	根据存储器设置
5-4	MWID	需要时设置
3-2	MTYP	需要时设置
1	MUXEN	0X0
0	MBKEN	0X1

表 107 FSMC_BTRx 位域(模式 D)

位编号	位名称	设置的数值
31-30	保留	0x0
29-28	ACCMOD	0x2
27-16	保留	0x000
15-8	DATAST	读操作的第 2 个阶段的长度（DATAST+3 个 HCLK 周期）。这个域不能为 0（至少为 1）。
7-4	ADDHLD	读操作的中间阶段的长度（ADDHLD+1 个 HCLK 周期）。

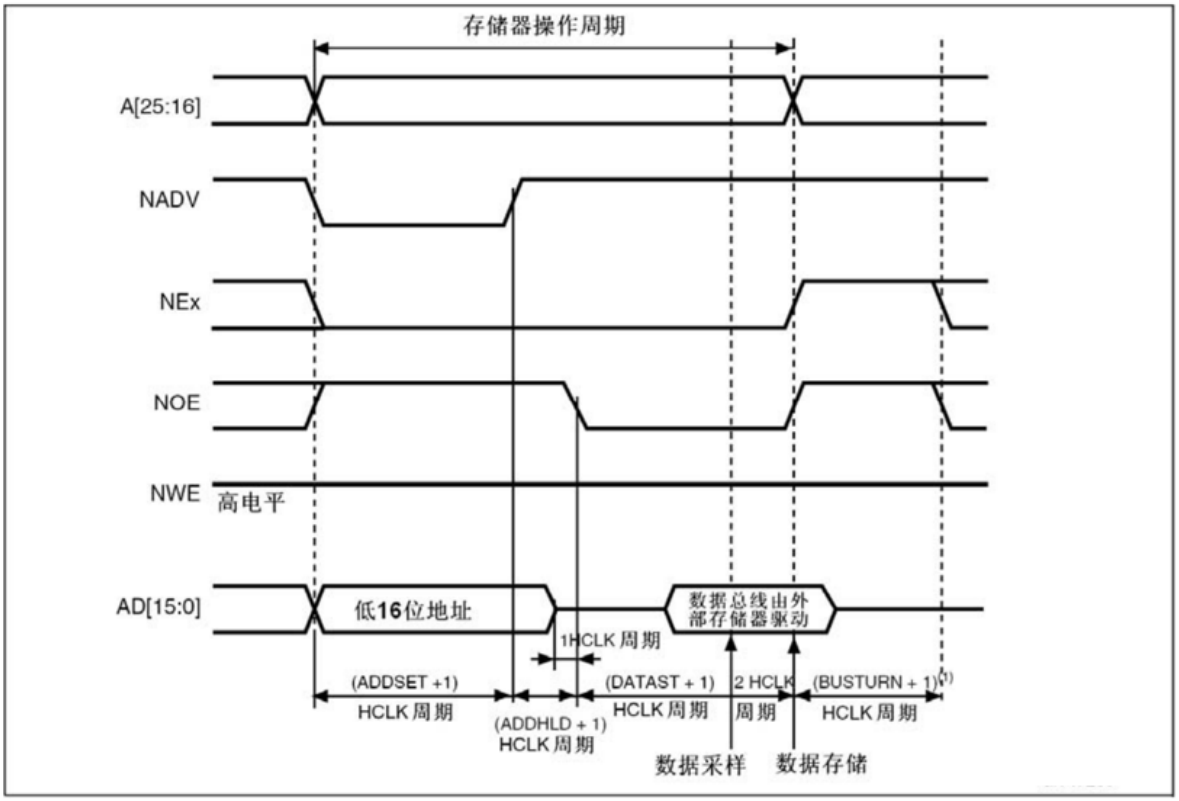
3-0	ADDSET	读操作的第 1 个阶段的长度（ADDSET+1 个 HCLK 周期）。

表 108 FSMC_BWTRx 位域(模式 D)

位编号	位名称	设置的数值
31-30	保留	0x0
29-28	ACCMOD	0x2
27-16	保留	0x000
15-8	DATAST	写操作的第 2 个阶段的长度（DATAST+1 个 HCLK 周期）。这个域不能为 0（至少为 1）。
7-4	ADDHLD	写操作的中间阶段的长度（ADDHLD+1 个 HCLK 周期）。
3-0	ADDSET	写操作的第 1 个阶段的长度（ADDSET+1 个 HCLK 周期）。

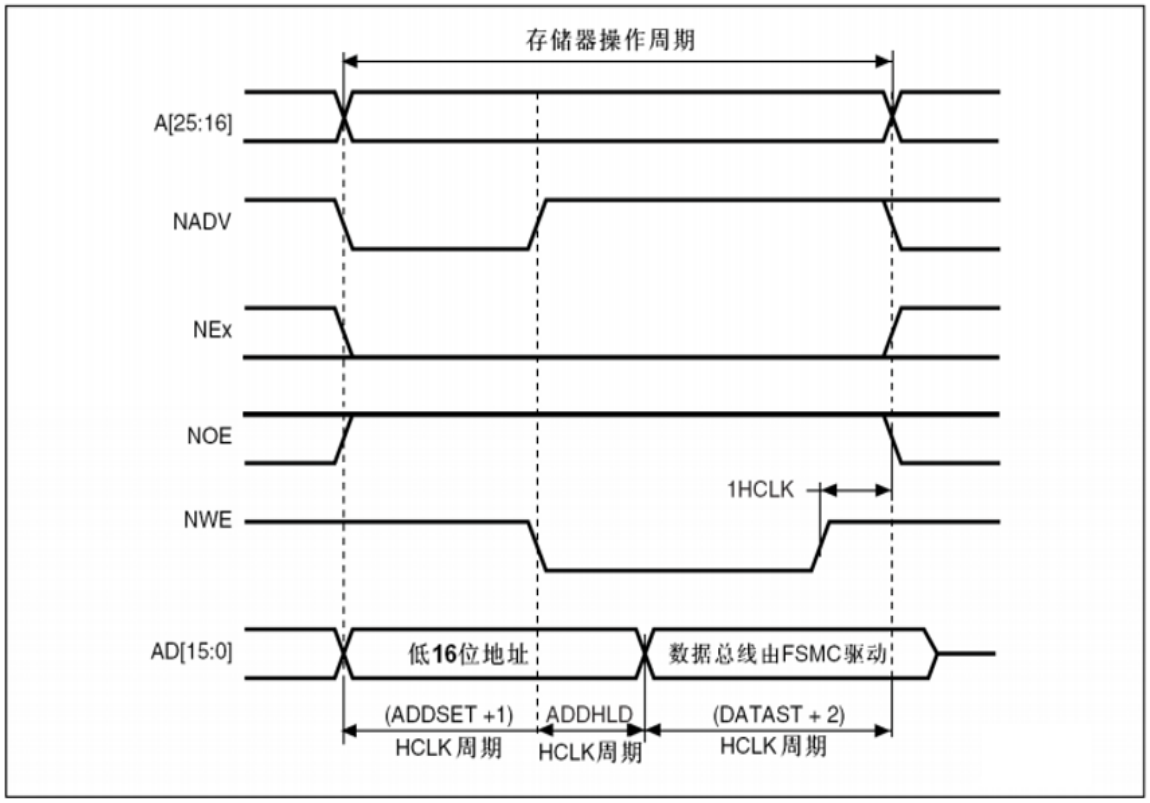
复用模式 —— 地址/ 数据复用的 NOR 闪存异步操作

图 173 复用读操作



总线恢复延迟(BUSTURN+1)与连续 2 次读操作之间在内部产生的延迟有部分重叠，因此 BUSTURN ≤ 5 时将不影响输出时序。

图 174 复用写操作



复用模式与模式 D 不同的是地址的低 16 位出现在数据总线上。

表 109 FSMC_BCRx 位域(复用模式)

位编号	位名称	设置的数值
31-15		0x0000
14	EXTMOD	0x01
13-10		0x0
9-7		0x0
6	FACCEN	0x1
5-4	MWID	需要时设置
3-2	MTYP	0x2（NOR 闪存）
1	MUXEN	0X1
0	MBKEN	0X1

表 110 FSMC_BTRx 位域(复用模式)

位编号	位名称	设置的数值
31-20	保留	0x0
19-16	BUSTURN	操作的最后阶段的长度（BUSTURN+1 个 HCLK 周期）。
15-8	DATAST	操作的第 2 个阶段的长度，读操作为（DATAST+3 个 HCLK 周期），写操作为 （DATAST+1 个 HCLK 周期）。这个域不能为 0（至少为 1）。
7-4	ADDHLD	操作的中间阶段的长度（ADDHLD+1 个 HCLK 周期）。这个域不能为 0（至少为 1）。
3-0	ADDSET	操作的第 1 个阶段的长度（ADDSET+1 个 HCLK 周期）。

6.5.5 NOR 闪存和 PSRAM 控制器存储器

必须以字(32 位)的方式操作这些外设寄存器。

SRAM/NOR 闪存片选控制寄存器 1…4 (FSMC_BCR1…4)

地址偏移：0xA000 0000 + 8 * (x-1), x=1…4

复位值：0x0000 30DX

这个寄存器包含了每个存储器块的控制信息，可以用于 SRAM、ROM、异步或成组传输的 NOR 闪存存储器。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	EXTMOD	Reserved	WREN	Reserved					FACCEN	MWD		MTYP		MUXEN	MBKEN
	rw		rw						rw	rw	rw	rw	rw	rw	rw

位 31:15　保留

位 14　EXTMOD：：扩展模式使能 (Extended mode enable)

该位允许 FSMC 使用 FSMC_BWTR 寄存器，即允许读和写使用不同的

位 13　保留

位 12　WREN ：写使能位 (Write enable bit)

该位指示 FSMC 是否允许/禁止对存储器的写操作。

0：禁止 FSMC 对存储器的写操作，否则产生一个 AHB 错误。

1：允许 FSMC 对存储器的写操作；这是复位后的默认状态。

位 11：7　保留

位 6　FACCEN：：闪存访问使能 (Flash access enable)

允许对 NOR 闪存存储器的访问操作。

0：禁止对 NOR 闪存存储器的访问操作.

1：允许对 NOR 闪存存储器的访问操作

位 5:4　MWID：：存储器数据总线宽度 (Memory databus width)

定义外部存储器总线的宽度，适用于所有类型的存储器。

00：8 位，

01：16 位(复位后的默认状态)，

10：保留，不能用

11：保留，不能用

位 3:2　MTYP：：存储器类型 (Memory type)

定义外部存储器的类型：

00：SRAM、ROM(存储器块 2...4 在复位后的默认值)

01：PSRAM(Cellular RAM: CRAM)

10：NOR 闪存(存储器块 1 在复位后的默认值)

11：保留

位 1　MUXEN：地址/数据复用使能位 (Address/data multiplexing enable bit)　当设置了该位后，地址的低 16 位和数据将共用数据总线，该位仅对 NOR 和 PSRM 存储器有效。

0：地址/数据不复用。

1：地址/数据复用数据总线；这是复位后的默认状态。

位 0　MBKEN：存储器块使能位 (Memory bank enable bit)　开启对应的存储器块。复位后存储器块 1 是开启的，其它所有存储器块为禁用。访问一个禁用的存

储器块将在 AHB 总线上产生一个错误。

0：禁用对应的存储器块。

1：启用对应的存储器块。

SRAM/NOR 闪存片选时序寄存器 1…4 (FSMC_BTR1…4)

地址偏移：0xA000 0000 + 0x04 + 8 * (x-1), x=1…4

复位值：0x0FFF FFFF

这个寄存器包含了每个存储器块的控制信息，可以用于 SRAM、ROM 和 NOR 闪存存储器。如果 FSMC_BCRx 寄存器中设置了 EXTMOD 位，则有两个时序寄存器分别对应读(本寄存器)和写操作(FSMC_BWTRx 寄存器)。

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

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

位 29:28 ACCMOD ：访问模式 (Access mode)

指定异步访问模式，如时序图所示。仅当 FSMC_BCRx 寄存器中的 ECTMOD 位为 1 时才考虑这些位。

00：访问模式 A

01：访问模式 B

10：访问模式 C

11：访问模式 D

位 15:8 DATAST ：数据阶段的持续时间 (Data-phase duration)

通过软件写入这些位可定义数据阶段的持续时间，适用于 SRAM、ROM 和异步 NOR Flash 访问模式：

0000 0000：Reserved

0000 0001：DATAST 阶段的持续时间 = 2 × HCLK 时钟周期

0000 0010：DATAST 阶段的持续时间 = 3 × HCLK 时钟周期

...

1111 1111：DATAST 阶段的持续时间 = 256 × HCLK 时钟周期（复位后的默认值）

有关每种存储器类型和访问模式数据阶段持续时间的信息，请参见相应图片（ 图 162 到 图 174）。

例如：模式 1、读操作、DATAST=1：数据保持时间=DATAST+3=4 个 HCLK 时钟周期。

位 7:4 ADDHLD ：地址保持阶段的持续时间 (Address-hold phase duration)

通过软件写入这些位可定义 地址保持 阶段的持续时间（请参见 图 171 到 图 174），适用于模式 D 和复用访问：

0000：Reserved

0001：ADDHLD 阶段的持续时间 = 1 × HCLK 时钟周期
0010：ADDHLD 阶段的持续时间 = 2 × HCLK 时钟周期
...
1111：ADDHLD 阶段的持续时间 = 15 × HCLK 时钟周期（复位后的默认值）
有关每种访问模式地址保持阶段持续时间的信息，请参见相应图片（ 图 171 到 图 174）。

位 3:0 ADDSET ：地址设置阶段的持续时间 (Address setup phase duration)
通过软件写入这些位可定义 地址设置 阶段的持续时间（请参见 图 162 到 图 174），适用于 SRAM、ROM 和异步 NOR Flash 访问模式：
0000：ADDSET 阶段的持续时间 = 1 × HCLK 时钟周期
...
1111：ADDSET 阶段的持续时间 = 16× HCLK 时钟周期（复位后的默认值）
有关每种访问模式地址设置阶段持续时间的信息，请参见相应图片（请参见图 162 到图 174）。
例如：模式 2、读操作、ADDSET=1：地址建立时间=ADDSET+1=2 个 HCLK 时钟周期

注： 因为内部的刷新， PSRAM(CRAM) 具有可变的保持延迟，因此这样的存储器会在数据保持期间输出 NWAIT 信号以延长数据的保持时间。
使用 PSRAM(CRAM) 时 DATLAT 域应置为 0 ，这样 FSMC 可以及时地退出自己的保持阶段并开始对存储器发出的 NWAIT 信号进行采样，然后在存储器准备好时开始读或写操作。
这个操作方式还可以用于操作最新的能够输出 NWAIT 信号的同步闪存存储器，详细信息请参考相应的闪存存储器手册。

SRAM/NOR 闪存写时序寄存器 1…4 (FSMC_BWTR1…4)
地址偏移：0xA000 0000 + 0x104 + 8 * (x-1), x=1…4
复位值：0x0FFF FFFF
这个寄存器包含了每个存储器块的控制信息，可以用于 SRAM、ROM 和 NOR 闪存存储器。如果 FSMC_BCRx 寄存器中设置了 EXTMOD 位，则这个寄存器对应写操作。

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

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

位 29:28 **ACCMOD**： ：访问模式 (Access mode) 位 29:28
定义异步访问模式。这 2 位只在 FSMC_BCRx 寄存器的 EXTMOD 位为 1 时起作用。
00：访问模式 A
01：访问模式 B
10：访问模式 C
11：访问模式 D

位 15：8 **DATAST** ：数据保持时间 (Data-phase duration)
这些位定义数据的保持时间(见图 162 图 174 至)，适用于 SRAM、ROM 和异步总线复用模式的 NOR 闪存操作。
0000 0000：保留
0000 0001：DATAST 保持时间=2 个 HCLK 时钟周期

0000 0010：DATAST 保持时间=3 个 HCLK 时钟周期

……

1111 1111：DATAST 保持时间=256 个 HCLK 时钟周期(这是复位后的默认数值)。

位 7：4 **ADDHLD**： ：地址保持时间 (Address-hold phase duration)

这些位定义地址的保持时间(见图 171 图 174 至)，适用于 SRAM、ROM 和异步总线复用模式的 NOR 闪存操作。

0000：保留

0001：ADDHLD 保持时间=2 个 HCLK 时钟周期

0010：ADDHLD 保持时间=3 个 HCLK 时钟周期

……

1111：ADDHLD 保持时间=16 个 HCLK 时钟周期(这是复位后的默认数值)。

注：在同步 NOR 闪存操作中，这个参数不起作用，地址保持时间始终是 1 个闪存时钟周期。

位 3:0 **ADDSET**： ：地址建立时间 (Address setup phase duration)

这些位以 HCLK 周期数定义地址的建立时间(见图 162 图 174 至)，适用于 SRAM、ROM 和异步总线复用模式的 NOR 闪存操作。

0000：ADDSET 建立时间=1 个 HCLK 时钟周期

……

1111：ADDSET 建立时间=16 个 HCLK 时钟周期(这是复位后的默认数值)。

注：在同步 NOR 闪存操作中，这个参数不起作用，地址建立时间始终是 1 个闪存时钟周期。

6.5.6 FSMC 寄存器地址映象

偏移	寄存器名称	复位值	说明
000h	FSMC_BCR1	0x000 30DB	SRAM/NOR 闪存片选控制寄存器 1
004h	FSMC_BTR1	0x0FFF FFFF	SRAM/NOR 闪存片选控制寄存器 1
008h	FSMC_BCR2	0x000 30D2	SRAM/NOR 闪存片选控制寄存器 2
00Ch	FSMC_BTR2	0x0FFF FFFF	SRAM/NOR 闪存片选控制寄存器 2
010h	FSMC_BCR3	0x000 30D2	SRAM/NOR 闪存片选控制寄存器 3
014h	FSMC_BTR3	0x0FFF FFFF	SRAM/NOR 闪存片选控制寄存器 1
018h	FSMC_BCR3	0x000 30D2	SRAM/NOR 闪存片选控制寄存器 4
01Ch	FSMC_BTR4	0x0FFF FFFF	SRAM/NOR 闪存片选控制寄存器 4
104h	BWTR1	0x0FFF FFFF	SRAM/NOR 闪存片选控制寄存器 1
10Ch	BWTR2	0x0FFF FFFF	SRAM/NOR 闪存片选控制寄存器 2
114h	BWTR3	0x0FFF FFFF	SRAM/NOR 闪存片选控制寄存器 3
11Ch	BWTR4	0x0FFF FFFF	SRAM/NOR 闪存片选控制寄存器 4

6.6 NAND Flash 控制器

FSMC 会生成相应的信号时序，用于驱动以下类型的设备：

- NAND Flash
  - 8 位
  - 16 位

NAND Flash 控制器可以控制两个外部存储区域，存储区域 2 和存储区域 3。

每个存储区域都通过专用的寄存器配置。可编程的存储器参数包括访问时序和 ECC 配置。

表 可编程的 NAND Flash 访问参数

参数	功能	访问模式	单位	最小值	最大值
存储器建立时间	命令使能前地址建立的时钟周期(HCLK)数	读 / 写	AHB 时钟周期(HCLK)	1	256
存储器等待	命令使能的最小持续时间 CHCLK 时钟周期）	读 / 写	AHB 时钟周期(HCLK)	2	256
存储器保待	命令禁止后保待地址（以及进行写访问时保持数据）的时钟周期(HCLK)数	读 / 写	AHB 时钟周期(HCLK)	1	255
存储器数据总线高阻态	开始进行写访问后，数据总线保持高阻状态期间的时钟周期(HCLK)数	写	AHB 时钟周期(HCLK)	0	255

6.6.1 外部存储器接口信号

下表列出了通常用于连接 NAND Flash 的信号。

8 位 NAND Flash

表 216. 8 位 NAND Flash

FSMC 信号名称	I/O	功能
A[17]	0	NAND Flash 地址锁存使能(ALE)信号
A[16]	0	NAND Flash 命令锁存使能(CLE)信号
D[7:0]	I/O	8 位复用双向地址 / 数据总线
NCE[x]	O	片选，X=2、3
NOE(=NRE)	O	输出使能（存储器信号名称：读取使能，NRE)
NWE	O	写入使能
NWAIT/INT[3:2]	I	指向 FSMC 的 NAND Flash 就绪 / 忙碌输入信号

由于 FSMC 能够管理足够多的地址周期，因此不存在理论容量极限值。

16 位NAND Flash

表 217. 16 位 NAND Flash

FSMC 信号名称	I/O	功能
A[17]	0	NAND Flash 地址锁存使能(ALE)信号
A[16]	0	NAND Flash 命令锁存使能(CLE)信号
D[15:0]	I/O	16 位复用双向地址 / 数据总线
NCE[x]	O	片选，X=2、3
NOE(=NRE)	O	输出使能（存储器信号名称：读取使能，NRE)
NWE	O	写入使能
NWAIT/INT[3:2]	I	指向 FSMC 的 NAND Flash 就绪 / 忙碌输入信号

由于 FSMC 能够管理足够多的地址周期，因此不存在理论容量极限值。

6.6.2 NAND Flash 支持的存储器和事务

下面介绍了所支持的设备、访问模式和事务。NAND Flash 控制器不允许

设备	模式	R/W	AHB 数据大小	存储器数据大小	允许/不允许	注释
8 位 NAND	异步	R	8	8	是	
	异步	W	8	8	是	
	异步	R	16	8	是	分为 2 次 FSMC 访问
	异步	W	16	8	是	分为 2 次 FSMC 访问
	异步	R	32	8	是	分为 4 次 FSMC 访问
	异步	W	32	8	是	分为 4 次 FSMC 访问
16 位NAND	异步	R	8	16	是	
	异步	W	8	16	否	
	异步	R	16	16	是	
	异步	W	16	16	是	

	异步	R	32	16	是	分为 2 次 FSMC 访问
	异步	W	32	16	是	分为 2 次 FSMC 访问

STC MCU

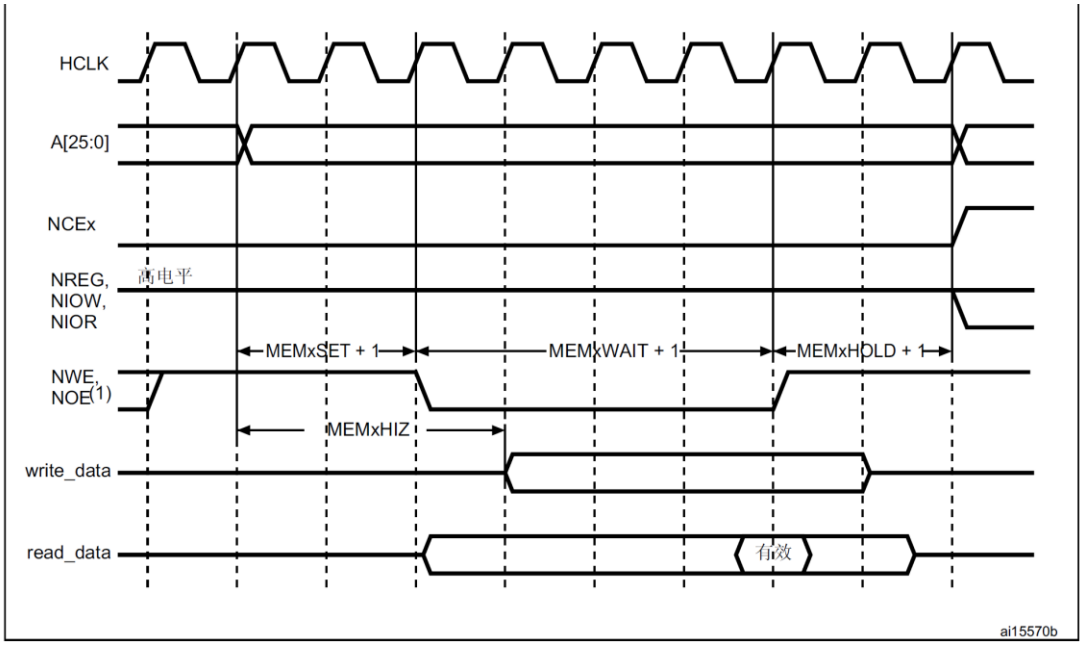
6.6.3 NAND 的时序图

6.6.4 每个 NAND Flash 存储区域均通过以下一组寄存器进行管理：

- 控制寄存器：FSMC_PCRx
- 中断状态寄存器：FSMC_SRx
- ECC寄存器：FSMC_ECCRx
- 通用存储器空间的时序寄存器：FSMC_PMEMx
- 特性存储器空间的时序寄存器：FSMC_PATTx
- I/O空间的时序寄存器：FSMC_PIOx

每个时序配置寄存器包含四个参数，其中三个参数用于定义 NAND Flash 的二个访问阶段的 HCLK 周期数，另一个参数用于定义进行写操作时开始驱动数据总线的时序。下图介绍了通用存储器访问的时序参数定义，特性存储器空间的访问时序与此类似。

图 423. NAND 控制器的通用存储器访问时序



进行写访问期间 NOE 保持高电平（无效）。进行读访问期间 NWE 保持高电平（无效）。

NAND Flash操作

NAND Flash 设备的命令锁存使能(CLE)信号和地址锁存使能(ALE)信号由FSMC 控制器的某些地址信号驱动。这意味着要向 NAND Flash 发送命令或地址，CPU 必须对其存储器空间中的某个特定地址执行写操作。

从 NAND Flash 设备进行的典型页读取操作如下所示：

1. 根据 NAND Flash 的特性（PWID 位指示 NAND Flash 数据总线宽度，PTYP=1, PWAITEN=1,PBKEN=1），配置 FSMC_PCRx 和 FSMC_PMEMx 寄存器，进而配置和使能相应的存储区域。
2. CPU在通用存储器空间执行字节写操作，此时数据字节等于一个 Flash命令字节（例如SamsungNAND Flash设备的字节为0x00)。在写入选通期间（NWE 上为低电平脉冲），NAND Flash的CLE输入有效，因此可将已写入的字节视为NAND Flash的一个命令。该命令由NAND Flash设备锁存后，随后进行页读取操作时，不需要再对该命令执行写操作。
3. 通过在通用存储器空间或特性空间写入所需字节（例如4个字节，较小容量的设备写入 3 个字节，STARTAD[7:0]、STARTAD[15:8]、STARTAD[23：16]以及最后为 64Mbx8 位的 NAND Flash 写入 STARTAD[25：24]),CPU 能够发送读操作的起始地址 (STARTAD)。在写入选通期间(NWE 上为低电平脉冲)，NAND Flash 设备的 ALE 输入有效，因此可将已写入的字节视为操作的起始地址。借助特性存储器空间，可使用

FSMC 的另一种不同时序配置实现某些 NAND Flash 所需的预等待功能。

4. 控制器在对同一个或另一个存储区域进行新访问之前，等待NAND Flash准备好（R/NB 信号处于高电平）进入有效状态。等待期间，控制器保持 NCE 信号有效（低电平）。
5. 然后CPU能够在通用存储器空间执行字节读操作，进而按字节读取NAND Flash页（数据字段 + 备用字段）。
6. 可按照以下三种不同的方式读取下一个NAND Flash页，而无需任何CPU命令或地址写操作：

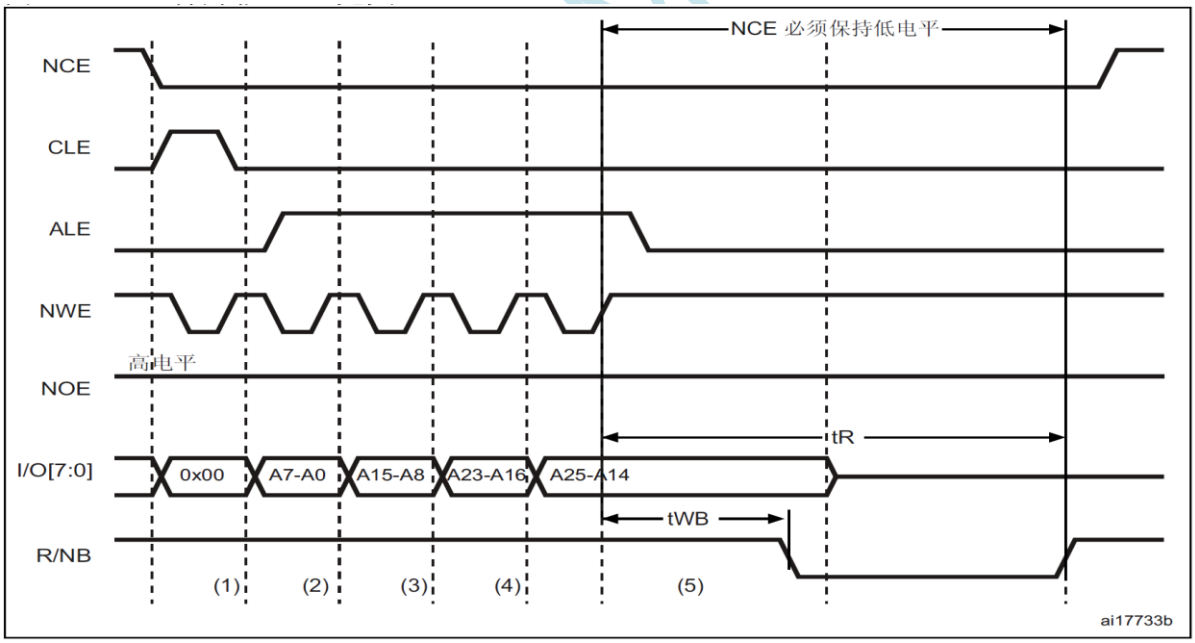
   执行步骤 5 中介绍的操作

   通过重新开始步骤 3 中的操作随机访问一个新地址通过重新开始步骤 2 向 NAND Flash 设备发送新命令

6.6.5 NAND Flash 预等待功能

一些 NAND Flash 设备需要在写入地址的最后一部分后，控制器等待 R/NB 信号变为低电平，如 424 所示。

图 424. 访问非"CE 无关"NAND-Flash



1.CPU 在地址 0x70010000 处写入字节 0x00。

2.CPU 在地址 0x70020000 处写入字节 A7-A0。

3.CPU 在地址 0x70020000 处写入字节 A15-A8。

4.CPU 在地址 0x70020000 处写入字节 A23-A16。

5.CPU 在地址 0x78020000 处写入字节 A25-A24:FSMC 通过 FSMC_PATT2 时序定义执行写访问，

其中 ATTHOLD>=7C 假设(7+1)xHCLK=112ns>Tw8 最大值）。这可确保 NCE 保待低电平，直到 R/NB 再次变为低电平后变为高电平（只有 NCE 信号对之有作用的 NAND Flash 才需要此功能）。

当需要此功能时，可通过编程MEMHOLD 值确保满足twb 时序，然而随后CPU 对NAND Flash 进行的所有读或写访问均会经过(MEMHOLD+1)个HCLK 周期的保持延迟（该延迟插入在 NWE 信号的上升沿与下一访问之间）。要克服该时序限制，可使用特性存储器空间，将其时序寄存器编程为满足 twb 时序的ATTHOLD 值并将 MEMHOLD 保持为其最小值。然后，CPU 必须使用通用存储器空间进行所有的 NAND Flash 读和写访问，向NAND Flash 设备写入最后一个地址字节时除外，此时CPU 必须对特性存储器空间执行写操作。

6.6.6 错误修正代码计算 ECC(NAND Flash)

FSMCPC 卡控制器包括两个错误修正代码计算硬件模块，每个存储区域各有一个。这些模块用于软件在系统中处理错误修正代码时，减少主机 CPU 工作负载。

这两个寄存器相同，分别与存储区域 2 和存储区域 3 相关联。

对NAND Flash 执行议或写操作时，相应每256、512、1024、2048、4096或8192 个字节，FSMC 中使用的错误修正代码(ECC)算法可修正 1 位错误并且检测出 2 位错误。

每当 NAND Flash 存储区域处于激活状态时，ECC 模块均会监视 NAND Flash 数据总线和读 / 写信号 CNCE 和 NWE)。

该功能的操作如下：

- 当在存储区域2或存储区域3中访问NAND Flash时，将锁存D[15:0]总线上出现的数据并将其用于ECC计算。
- 当在任何其它地址访问NAND Flash时，ECC逻辑会进入空闲状态，不执行任何操作。因此，进行ECC计算时，用于定义NAND Flash命令或地址的写操作无效。

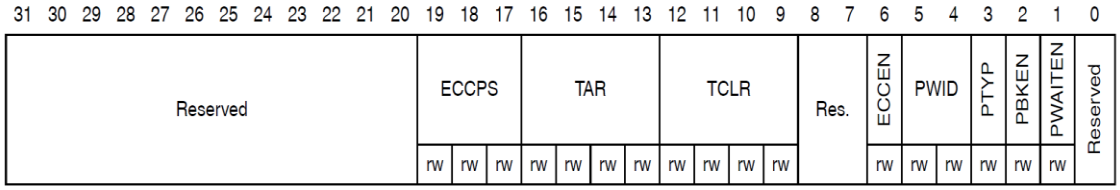
主机 CPU 从NAND Flash 读取/向NAND Flash 写入所需的字节数后，必须读取FSMC_ECCR2/3 寄存器，才能检索计算出的值。读取后，应通过将ECCEN 位复位为零来将这些寄存器清零。要计算新的数据块，必须将FSMC_PCR2/3 寄存器中的 ECCEN 位置1。

6.6.7 NAND Flash 控制寄存器

NAND Flash 控制寄存器必须通过字(32 位）访问。

偏移地址：0xA0000000+0x40+0x20*(x-1),x=2..4

复位值：0x00000018



位 31:20 保留，必须保持复位值。

位 19:17ECCPS:ECC 页大小(ECC page size)。

定义扩展 ECC 的页大小：

000:256 字节

001: 512 字节

010:1024 字节

011: 2048 字节

100:4096 字节

101:8192 字节

位 16:13TAR:ALE 到 RE 的延迟(ALE to RE delay)。

以 AHB 时钟周期数(HCLK)设置从 ALE 低电平到 RE 低电平的时间。
时间是：t_ar=(TAR+SET+2)xTHCLK,其中 THCLK 是 HCLK 时钟周期
0000:1HCLK 周期（默认）
1111:16HCLK 周期

位 12:9 TCLR:CLE 到 RE 的延迟(CLE to RE delay)。

以 AHB 时钟周期数(HCLK)设置从 CLE 低电平到 RE 低电平的时间。
时间是 t_clr=(TCLR+SET+2)xTHCLK,其中 THCLK 是 HCLK 时钟周期
0000:1HCLK 周期（默认）

1111:16HCLK 周期

位 8:7 保留，必须保持复位值

位 6 ECCEN:ECC 计算逻辑使能位(ECC computation logic enable bit)

0:禁止和复位 ECC 逻辑（复位后为默认值），

1:使能 ECC 逻辑。

位 5:4 PWID:数据总线宽度(Data

bus width)。定义外部存

储器设备宽度。

00:8 位（复位后为默认值）

~~01:16 位(PC 卡时必需如此~~）

10:保留，不使用

11:保留，不使用

位 3PTYP:存储器类型(memory type)。

定义附加到相应存储区域的设备的类型：

~~0: PC 卡、CF 卡、CF+或PCMCIA~~

1:NAND FlashC 复位后为默认值）

位 2PBKEN:PC 卡/NAND Flash 存储区域使能位(PCCard/NAND Flash memory bank enable bit)。使能存储区域。访问禁止的存储区域会引起 AHB 总线上的错误

0:禁止相应的存储区域(复位后为默认值)

1:使能相应的存储区域

位 1PWAITEN:等待特性使能位(Waitfeature enable bit)。使能 PC 卡/NAND Flash 存储区域的等待特性:

0: 禁止

1:使能

位 0 保留,必须保持复位值。

FIFO 状态和中断寄存器 2••4(FSMC_SR2••4)

FIFO status and Interrupt register2..4

偏移地址:0xA0000000+0x44+0x20*(x-1),x=2..4

复位值:0x00000040

该寄存器包含有关 FIFO 状态和中断的信息。FSMC 有一个 FIFO,当向存储器执行写入操作以存储来自 AHB 的多达 16 字的数据时,使用该 FIFO。
当 FSMC 将其 FIFO 的内容移入存储器时,此寄存器用于快速写入 AHB,然后释放 AHB 供 FSMC 以外的外设的事务使用。该寄存器有一位用来指示 FIFO 的状态,供 ECC 使用。
当将数据写入存储器时计算 ECC,因此,为了读取正确的 ECC,软件必须等待到 FIFO 为空。

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
Reserved																									FEMPT	IFEN	ILEN	IREN	IFS	ILS	IRS
																									r	rw	rw	rw	rw	rw	rw

位 31:7 保留,必须保持复位值。

位 6 FEMPT:FIFO 为空(FIFO empty)。

该位是提供 FIFO 状态的只读位

0:FIFO 非空

1:FIFO 为空

位 5 IFEN:中断下降沿检测使能位(Interrupt falling edge detection enable bit)

0:中断下 降沿检测请求禁止

1:中断下降沿检测请求使能

位 4 ILEN:中断高电平检测使能位(Interrupt high-level detection enable bit)

0:中断高电平检测请求禁止

1:中断高电平检测请求使能

位 3 IREN:中断上升沿检测使能位(Interrupt rising edge detection enable bit)

0:中断上升沿检测请求禁止

1:中断上升沿检测请求使能

位 2 IFS:中断下降沿状态(Interrupt falling edge status)

此标志由硬件置 1'由软件复位。

0:未出现中断下降沿

1:出现中断下降沿

位 1 ILS:中断高电平状态(Interrupt high-level status)

此标志由硬件置 1'由软件复位。

0:未出现中断高电平

1:出现中断高电平

位 0 IRS:中断上升沿状态(Interrupt rising edge status)

此标志由硬件置 1 由软件复位。

0:未出现中断上升沿

1:出现中断上升沿

通用存储器空间时序寄存器 2..4(FSMC_PMEM2..4)

Common memory space timing register2..4

偏移地址：地址：0xA0000000+0x48+0x20*(x-1),x=2..4

复位值：0xFCFCFCFC

每个 FSMC_PMEMx(x=2..4)读/写寄存器都包含 ~~PC 卡或~~NAND Flash 存储区域 x 的时序信息，~~用于访问 16 位 PC 卡/CF 卡的通用存储空间~~，或者访问 NAND Flash 来实现命令、地址写和数据读 /写访问。

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
MEMHIZx								MEMHOLDx								MEMWAITx								MEMSETx							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 MEMHIZx:通用存储器 x 数据总线高阻态时间(Common memory x data bus HiZ time)

定 义在存储区 x 上~~对 PC 卡/~~NAND Flash 的通 用存储空间开始 执行 写访问之后，数据总线保持高阻态所 持续的 HCLK 时钟周期数。仅对写入事务 有效：

00000000:(0xOO)0 个 HCLK 周期（对于 PC 卡）

11111111:(0xFF)255 个 HCLK 周期（对于 PC 卡）一（复位后的默认值）

位 23:16 MEMHOLDx:通用存储器 x 保持时间(Common memory x hold time)

针对在存储区 x 上对 PC 卡/NAND Flash 的通用存储空间执行的读或写访问，定义禁止命令(NWE、NOE)之后保持地址（和写访问数据）的 HCLK 时钟周期数。

00000000:保留

00000001:1 个 HCLK 周期

11111111:255 个 HCLK 周期（复位后的默认值）

位 15:8 MEMWAITx:通用存储器 x 等待时间(Common memory x wait time)

针对在存储区 x 上对 ~~PC 卡/~~NAND Flash 的通用存储空间执行的读或写访问，定义使能命令 CNWE、NOE)所需的 HCLK(+1)时钟周期数最小值。如果等待信号(NWAIT)在编程的 HCLK 值末尾处有效（低电平），则命令使能的持续时间将延长。

00000000:保留

00000001:2 个 HCLK 周期（＋禁止 NWAIT 时引入的等待周期）

位 7:0 MEMSETx:通用存储器 x 建立时间(Common memory x setup time)

针对在存储区 x 上对 PC 卡/NAND Flash 的通用存储空间执行的读或写访问，定义使能命令 CNWE,NOE)前建立地址所需的 HCLK()时钟周期数:

00000000:1 个 HCLK 周期（对于 PC 卡）/HCLK 周期（对于 NAND Flash)

11111111:256 个 HCLK 周期(对于 PC 卡)/257HCLK 周期(对于 NAND Flash)-C

复位后的默认值）

属性存储器空间时序寄存器 2••4(FSMC_PATT2••4)

Attribute memory space timing registers2..4

偏移地址：0xA0000000+0x4C+0x20*(x-1),x=2..4

复位值：0xFCFCFCFC

每个FSMC_PATTx(x=2..4)读/写寄存器都包含 ~~PC 卡/CF 卡或~~ NAND Flash 存储区域 x 的时序信息。~~该寄存器用于对 PC 卡/CF 卡的特性存储空间的 8 位访问~~，或在最后一次地址写访问必须先前访问的时序不同的情况下，访问NAND Flash 来实现最后一次地址写访问。

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
ATTHIZx								ATTHOLDx								ATTWAITx								ATTSETx							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 ATTHIZx:特性存储器 x 数据总线高阻态时间(Attribute memory x data bus HiZ time)

定义在存储区 x 上对 ~~PC 卡/~~NAND Flash 的特性存储空间开始执行写访问之后，

数据总线保持高阻态所持续的 HCLK 时钟周期数。仅对写入事务有效:

00000000:0 个 HCLK 周期

11111111:255 个 HCLK 周期（复位后的默认值）

位 23:16 ATTHOLDx:特性存储器 x 保持时间(Attribute memory x hold time)

针对在存储区 x 上 NAND Flash 的特性存储空间执行的读或写访问，定义禁止命令 CNWE、NOE)之后保持地址（和写访问数据）的 HCLK 时钟周期数。

00000000:保留

00000001:1 个 HCLK 周期

11111111:255 个 HCLK 周期（复位后的默认值）

位 15:8 ATTWAITx:特性存储器 x 等待时间(Attribute memory x wait time)

NAND Flash 的特性存储空间执行的读或写访问，定义使能命令 CNWE、NOE)所需的 HCLK(+1)时钟周期数最小值。如果等待信号(NWAIT)在编程的 HCLK 值末尾处有效（低电平），则命令使能的待续时间将延长。

00000000:保留

00000001:2 个 HCLK 周期（＋禁止 NWAIT 时引入的等待周期）

位 7:0 ATTSETx:特性存储器 x 建立时间(Attribute memory x setup time)

NAND Flash 的特性存储空间执行的读或写访问，定义使能命令 CNWE,NOE)前建立地址所需的 HCLK(+1)时钟周期数：

00000000:1 个 HCLK 周期

11111111:256 个 HCLK 周期（复位后的默认值）

ECC 结果寄存器 2/3(FSMC_ECCR2/3)

ECC result registers2/3

偏移地址：0xA0000000+0x54+0x20*(x-1),x=2 或 3

复位值：0x00000000

这些寄存器中包含由 FSMC 控制器的ECC 计算模块（每个NAND Flash 存储区域一个模块）计算所得的当前错误校正代码值。当CPU 从NAND Flash 页的正确地址议取数据时（请参见第32.6.6 节代码计算ECC(NAND Flash),ECC 计算校块会自动处理读取自 / 写入NAND Flash 的数据。

当X 字节读取操作结束时(根据 FSMC_PCRx 寄存器中的ECCPS 字段判断),CPU 必须从FSMC_ECCx 寄存器读取计算所得的 ECC 值，然后验证这些计算的奇偶校验数据是否与备用区记录的奇偶校验数据相同，从而确定该页是否有效，并根据需要进行校正。读取 FSMC_ECCRx 寄存器后应通过将 ECCEN 位置零将其清零。如要计算新的数据块，则必须将ECCEN 位置1。

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
ECCx																															
r																															

位 31:0ECCx:ECC 结果(ECCresult)

该字段提供由 ECC 计算逻辑计算所得的值。表 221. ECC 结果

相关位 ECCPS[2:0]	以字节为单位的页大小	ECC 位
000	256	ECC[21:OJ
001	512	ECC[23:0]
010	1024	ECC[25:0]
011	2048	ECC[27:0]
100	4096	ECC[29:0]

7 电源控制器 (PWR)

除非特别说明，否则本部分适用于整个 STC33F 系列。

7.1 电源

器件的工作电压 (VDD) 要求介于 1.9 V 到 5.5 V 之间。嵌入式线性调压器用于提供内部 1.2 V 数字电源。

1. VDDA 和 VSSA 必须分别连接到 VDD 和 VSS 。

7.1.1 独立 A/D 转换器电源和参考电压

为了提高转换精度，ADC 配有独立电源，可以单独滤波并屏蔽 PCB 上的噪声。

- ADC 电源电压从单独的 V DDA 引脚输入。
- VSSA 引脚提供了独立的电源接地连接。

为了确保测量低电压时具有更高的精度，用户可以在 VREF 上连接单独的 ADC 外部参考电压输入。VREF 电压介于 1.8 V 到 VDDA 之间。

7.1.2 调压器

嵌入式线性调压器为备份域和待机电路以外的所有数字电路供电。调压器输出电压约为 1.2 V。

此调压器需要将两个外部电容连接到专用引脚 V CAP_1 和 V CAP_2 ，所有封装都配有这两个引脚。为激活或停用调压器，必须将特定引脚连接到 V SS 或 V DD 。具体引脚与封装有关。

通过软件激活时，调压器在复位后始终处于使能状态。根据应用模式的不同，可采用三种不同的模式工作。

● 运行模式，调压器为 1.2 V 域（内核、存储器和数字外设）提供全功率。在此模式下，调压器的输出电压（约 1.2 V）可通过软件调整为不同的电压值：

— 可通过 VOS（PWR_CR 寄存器的位 15）动态配置成级别 1 或级别 2。

— 可通过 PWR_CR 寄存器的 VOS[1:0] 位配置成级别 1、级别 2 或级别 3。仅当关闭 PLL 且选择 HSI 或 HSE 时钟源作为系统时钟源时，才能修改输出级别。新的编程值只在 PLL 开启后才生效 。 PLL 关闭后将自动选择电压输出级别 3。器件运行在最高工作频率时，电压缩放特性可使功耗得到优化。

● 停止模式，调压器为 1.2 V 域提供低功率，保留寄存器和内部 SRAM 中的内容。

— 在停止模式下，设置的电压输出级别保持不变）。

— 微控制器进入停止模式后将自动选择电压输出级别 3）。

● 待机模式，调压器掉电。除待机电路和备份域外，寄存器和 SRAM 的内容都将丢失。

STC MCU

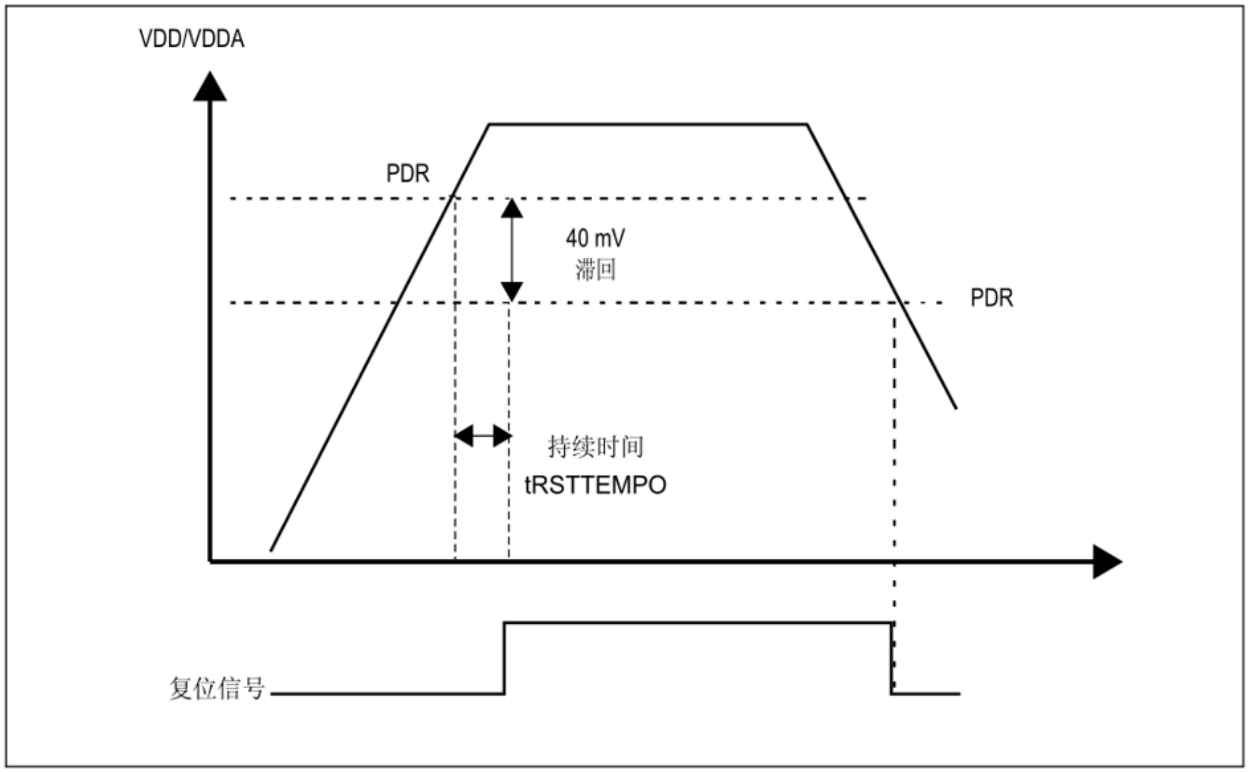
7.2 电源监控器

7.2.1 上电复位 (POR)/ 掉电复位 (PDR)

本器件内部集成有 POR/PDR 电路，可以从 1.8 V 开始正常工作。

当 V DD /V DDA 低于指定阈值 V POR/PDR 时，器件无需外部复位电路便会保持复位状态。

图 9. 上电复位/ 掉电复位波形



7.2.2 欠压复位 (BOR)

上电期间，欠压复位 (BOR) 将使器件保持复位状态，直到电源电压达到指定的 V BOR 阈值。

V BOR 通过器件选项字节进行配置。BOR 默认为关闭。可以选择 4 个 V BOR 阈值。

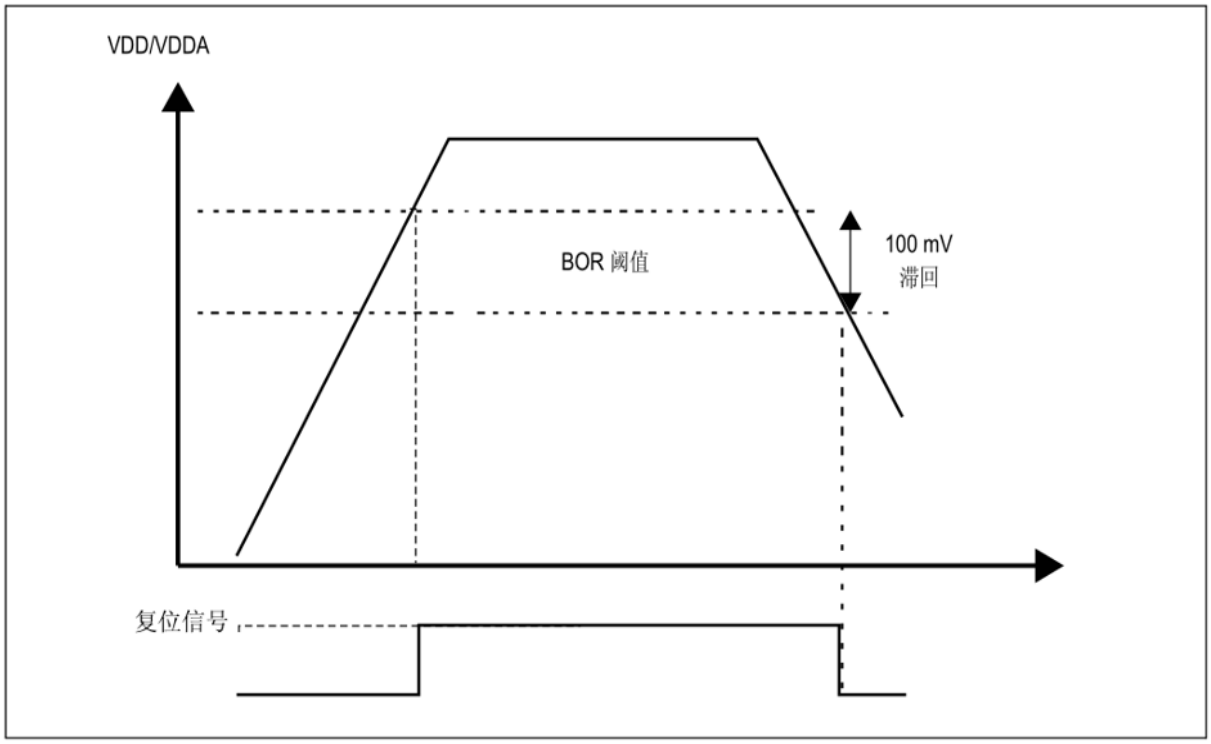
- BOR 关闭 (V BOR0)：1.80 V 到 2.10 V 电压范围的复位阈值级别
- BOR 级别 1 (V BOR1)：2.10 V 到 2.40 V 电压范围的复位阈值级别
- BOR 级别 2 (V BOR2)：2.40 V 到 2.70 V 电压范围的复位阈值级别
- BOR 级别 3 (V BOR3)：2.70 V 到 3.60 V 电压范围的复位阈值级别

当电源电压 (V DD) 降至所选 V BOR 阈值以下时，将使器件复位。

通过对器件选项字节进行编程可以禁止 BOR。要禁止 BOR 功能，V DD 必须高于 V BOR0 ，以启动器件选项字节编程序列。那么就只能由 PDR 监测掉电过程（参见 第 7.2.1 节：上电复位(POR)/ 掉电复位 (PDR)）

BOR 阈值滞回电压约为 100 mV（电源电压的上升沿与下降沿之间）。

图 10. BOR 阈值



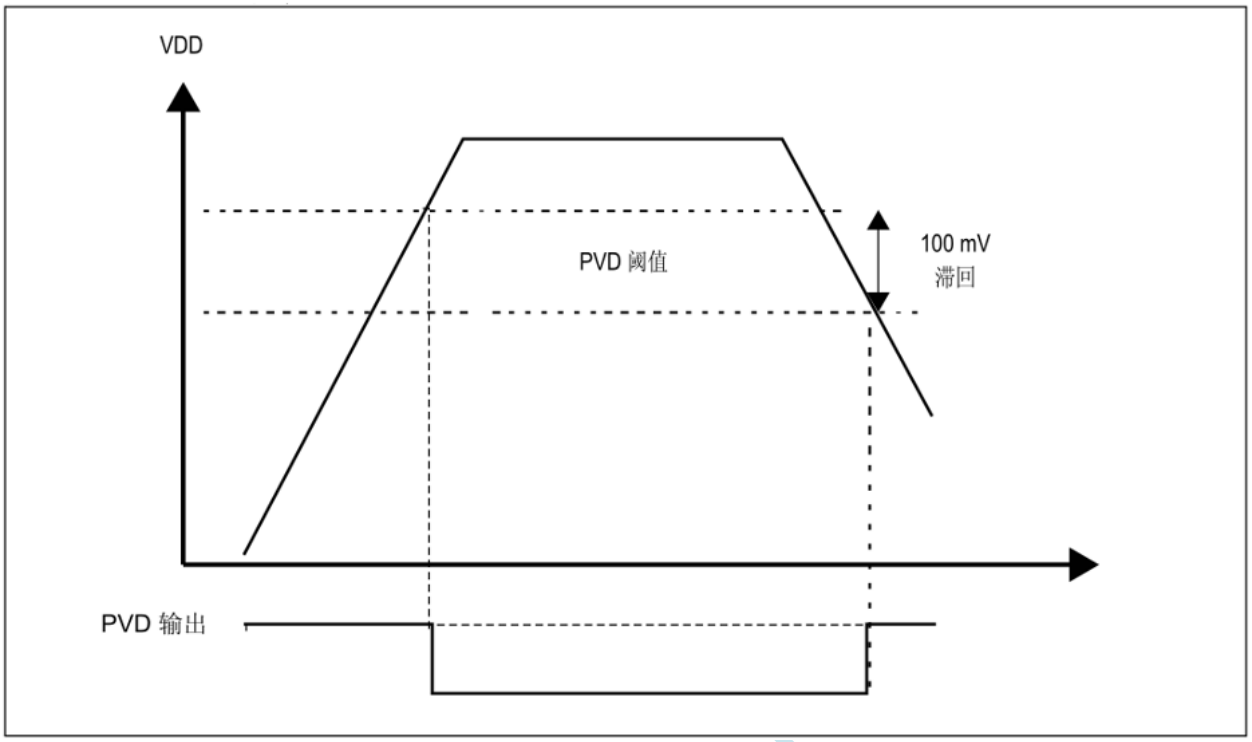
7.2.3 可编程电压检测器 (PVD)

可以使用 PVD 监视 V DD 电源，将其与 PWR 电源控制寄存器 (PWR_CR) 中 PLS[2:0] 位所选的阈值进行比较。通过设置 PVDE 位来使能 PVD。

PWR 电源控制 / 状态寄存器 (PWR_CSR) 中提供了 PVDO 标志，用于指示 V DD 是大于还是小于 PVD 阈值。该事件内部连接到 EXTI 线 16，如果通过 EXTI 寄存器使能，则可以产生中断。当 V DD 降至 PVD 阈值以下以及/或者当 V DD 升至 PVD 阈值以上时，可以产生 PVD 输出中断，具体取决于 EXTI 线 16 上升沿/下降沿的配置。该功能的用处之一就是可以在中断

服务程序中执行紧急关闭系统的任务。

图 11. PVD 阈值



7.3 低功耗模式

默认情况下，系统复位或上电复位后，微控制器进入运行模式。在运行模式下，CPU 通过 HCLK 提供时钟，并执行程序代码。系统提供了多个低功耗模式，可在 CPU 不需要运行时（例如等待外部事件时）节省功耗。由用户根据应用选择具体的低功耗模式，以在低功耗、短启动时间和可用唤醒源之间寻求最佳平衡。

器件有三个低功耗模式：

- 睡眠模式（Cortex™-MSTAR 内核停止，外设保持运行）
- 停止模式（所有时钟都停止）
- 待机模式（1.2 V 域断电）

此外，可通过下列方法之一降低运行模式的功耗：

- 降低系统时钟速度
- 不使用 APBx 和 AHBx 外设时，将对应的外设时钟关闭

表 18. 低功耗模式汇总

模式名称	进入	唤醒	对 1.2 V 域时钟的影响	对 VDD 域时钟的影响	调压器
睡眠（立即休眠或退出时休眠）	WFI	任意中断	CPU CLK 关闭 对其它时钟或模拟时钟源无影响	无	开启
	WFE	唤醒事件			
停止	PDDS 和 LPDS 位 +	任意 EXTI 线（在 EXTI	所有 1.2 V 域时钟都关闭	HSI 和 HSE 振	开启或处于低功耗模式

	SLEEPDEEP 位+ WFI 或 WFE	寄存器中配置，内部线和外部线）		荡器关闭	
待机	PDDS 位 + SLEEPDEEP 位 + WFI 或 WFE	WKUP 引脚上升沿、NRST 引脚外部复位、IWDG 复位	所有 1.2 V 域时钟都关闭	HSI 和 HSE 振荡器关闭	关闭

7.3.1 降低系统时钟速度

在运行模式下，可通过对预分频寄存器编程来降低系统时钟（SYSCLK、HCLK、PCLK1 和 PCLK2）速度。进入睡眠模式之前，也可以使用这些预分频器降低外设速度。

7.3.2 外设时钟门控

在运行模式下，可随时停止各外设和存储器的 HCLKx 和 PCLKx 以降低功耗。要进一步降低睡眠模式的功耗，可在执行 WFI 或 WFE 指令之前禁止外设时钟。
外设时钟门控由 AHB1 外设时钟使能寄存器 (RCC_AHB1ENR)、AHB2 外设时钟使能寄存器 (RCC_AHB2ENR)。

在睡眠模式下，复位 RCC_AHBxLPENR 和 RCC_APBxLPENR 寄存器中的对应位可以自动禁止外设时钟。

7.3.3 睡眠模式

进入睡眠模式

执行 WFI（等待中断）或 WFE（等待事件）指令即可进入睡眠模式。根据 Cortex™-MSTAR 系统控制寄存器中 SLEEPONEXIT 位的设置，可以通过两种方案选择睡眠模式进入机制：

● 立即休眠：如果 SLEEPONEXIT 位清零，MCU 将在执行 WFI 或 WFE 指令时立即进入睡眠模式。

● 退出时休眠：如果 SLEEPONEXIT 位置 1，MCU 将在退出优先级最低的 ISR 时立即进入睡眠模式。

退出睡眠模式

如果使用 WFI 指令进入睡眠模式，则嵌套向量中断控制器 (NVIC) 确认的任意外设中断都会将器件从睡眠模式唤醒。
如果使用 WFE 指令进入睡眠模式，MCU 将在有事件发生时立即退出睡眠模式。唤醒事件可通过以下方式产生：

● 在外设的控制寄存器使能一个中断，但不在 NVIC 中使能，同时使能 Cortex™-MSTAR 系统控制寄存器中的 SEVONPEND 位。当 MCU 从 WFE 恢复时，需要清除相应外设的中断挂起位

和外设 NVIC 中断通道挂起位（在 NVIC 中断清除挂起寄存器中）。

● 配置一个外部或内部 EXTI 线为事件模式。当 CPU 从 WFE 恢复时，因为对应事件线的挂起位没有被置位，不必清除相应外设的中断挂起位或 NVIC 中断通道挂起位。

由于没有在进入/退出中断时浪费时间，此模式下的唤醒时间最短。

有关如何退出睡眠模式的详细信息，请参见 表 19 和 表 20。

表 19. 进入和退出立即休眠

立即休眠模式	说明
进入模式	WFI（等待中断）或 WFE（等待事件），且： — SLEEPDEEP = 0 及 — SLEEPONEXIT = 0 请参见 Cortex™-MSTAR 系统控制寄存器。
退出模式	如果使用 WFI 进入：中断 如果使用 WFE 进入：唤醒事件
唤醒延迟	无

表 20. 进入和退出时休眠

退出时休眠	说明
进入模式	WFI（等待中断），且： — SLEEPDEEP = 0 及 — SLEEPONEXIT = 1 请参见 Cortex™-MSTAR 系统控制寄存器。
退出模式	中断
唤醒延迟	无

7.3.4 停止模式

停止模式基于 Cortex™-MSTAR 深度睡眠模式与外设时钟门控。调压器既可以配置为正常模式，也可以配置为低功耗模式。在停止模式下，1.2 V 域中的所有时钟都会停止，PLL、HSI 和 HSE RC 振荡器也被禁止。内部 SRAM 和寄存器内容将保留。将 PWR_CR 寄存器中的 FPDS 位置 1 后，Flash 还会在器件进入停止模式时进入掉电状态。Flash 处于掉电模式时，将器件从停止模式唤醒将需要额外的启动延时（参见 表 21 ：停止工作模式 ）。

表 21. 停止工作模式

停止模式	**LPDS** 位	**FPDS** 位	唤醒延迟
STOP MR（主调压器）	0	0	HSI RC 启动时间
STOP MR-FPD	0	1	HSI RC 启动时间+Flash 从掉电模式唤醒的时间
STOP LP	1	0	HSI RC 启动时间+ 调压器从 LP 模式唤醒的时间
STOP LP-FPD	1	1	HSI RC 启动时间+ Flash 从掉电模式唤醒的时间+ 调压器从 LP

			模式唤醒的时间

进入停止模式

如果正在执行 Flash 编程，停止模式的进入将延迟到存储器访问结束后执行。
如果正在访问 APB 域，停止模式的进入则延迟到 APB 访问结束后执行。

在停止模式下，可以通过对各控制位进行编程来选择以下功能：

- 独立的看门狗 (IWDG)：IWDG 通过写入其密钥寄存器或使用硬件选项来启动。而且一旦启动便无法停止，除非复位。独立看门狗 (IWDG) 中的 3 节 。
- 内部 RC 振荡器 (LSI RC)：通过 RCC 时钟控制和状态寄存器 (RCC_CSR) 中的 LSION 位进行配置。
- 外部 32kHz 振荡器 (LSE OSC)：通过 RCC 备份域控制寄存器 (RCC_BDCR) 中的 LSEON 位进行配置。

在停止模式下，ADC 也会产生功耗，除非在进入停止模式前将其禁止。要禁止这些转换器，必须将 ADC_CR2 寄存器中的 ADON 位 ENx 位都清零。

退出停止模式

有关如何退出停止模式的详细信息，请参见 表 22。
通过发出中断或唤醒事件退出停止模式时，将选择 HSI RC 振荡器作为系统时钟。
当调压器在低功耗模式下工作时，将器件从停止模式唤醒将需要额外的延时。在停止模式下一直开启内部调压器虽然可以缩短启动时间，但功耗却增大。

表 22. 进入和退出停止模式

停止模式	说明
进入模式	WFI（等待中断）或 WFE（等待事件），且： — 将 Cortex™-MSTAR 系统控制寄存器中的 SLEEPDEEP 位置 1 — 将电源控制寄存器 (PWR_CR) 中的 PDDS 位清零 — 通过配置 PWR_CR 中的 LPDS 位选择调压器模式。 注意：要进入停止模式，所有 EXTI 线挂起位（在 挂起寄存器 (EXTI_PR) 中）。否则将忽略进入停止模式这一过程，继续执行程序。
退出模式	如果使用 WFI 进入： 所有配置为中断模式的 EXTI 线（必须在 NVIC 中使能对应的 EXTI 中断向量）。如果使用 WFE 进入： 所有配置为事件模式的 EXTI 线。
唤醒延迟	表 21 ：停止工作模式

7.3.5 待机模式

待机模式下可达到最低功耗。待机模式基于 Cortex™-MSTAR 深度睡眠模式，其中调压器被禁

止。因此 1.2 V 域断电。PLL、HSI 振荡器和 HSE 振荡器也将关闭。

进入待机模式

有关如何进入待机模式的详细信息，请参见 表 23。
在待机模式下，可以通过对各控制位进行编程来选择以下功能：

- 独立的看门狗 (IWDG)：IWDG 通过写入其密钥寄存器或使用硬件选项来启动。而且一旦启动便无法停止，除非复位。请参见 第 19 节：独立看门狗 (IWDG) 中的 第 19.3 节 。
- 内部 RC 振荡器 (LSI RC)：通过控制/状态寄存器 (RCC_CSR) 中的 LSION 位进行配置。
- 外部 32kHz 振荡器 (LSE OSC)：通过备份域控制寄存器 (RCC_BDCR) 中的 LSEON 位进行配置。

退出待机模式

检测到外部复位（NRST 引脚）、IWDG 复位、WKUP 引脚上升沿、入侵事件或时间戳时间时，微控制器退出待机模式。从待机模式唤醒后，除 PWR 电源控制 / 状态寄存器 (PWR_CSR) 外，所有寄存器都将复位。
从待机模式唤醒后，程序将按照复位（启动引脚采样、复位向量已获取等）后的方式重新执行。PWR 电源控制 / 状态寄存器 (PWR_CSR) 中的 SBF 状态标志指示 MCU 已处于待机模式。
有关如何退出待机模式的详细信息，请参见 表 23。

表 23. 进入和退出待机模式

待机模式	说明
进入模式	WFI （等待中断）或 WFE （等待事件），且： -将 Cortex™-M4F 系统控制寄存器中的 SLEEPDEEP 位置 1 -将电源控制寄存器（PWR_CR）中的 PDDS 位置 1 -将电源控制/状态寄存器（PWR_CSR）中的 WUF 位清零 -将与所选唤醒源（RTC 闹钟 A、RTC 闹钟 B、RTC 唤醒、RTC 入侵或 RTC 时间戳标志）对应的 RTC 标志清零
退出模式	WKUP 引脚上升沿、RTC 闹钟（闹钟 A 和闹钟 B）、RTC 唤醒事件、RTC 入侵事件、RTC 时间戳事件、NRST 引脚外部复位和 IWDG 复位。
唤醒延退	复位阶段。

待机模式下的 I/O 状态

在待机模式下，除以下各部分以外，所有 I/O 引脚都处于高阻态.

调试模式

默认情况下，如果使用调试功能时应用程序将 MCU 置于停止模式或待机模式，调试连接将中断。这是因为 Cortex™-MSTAR 内核时钟停止了。
不过，通过设置 DBGMCU_CR 寄存器中的一些配置位，即使 MCU 进入低功耗模式，仍可使用软件对其进行调试。

7.4 电源控制寄存器

7.4.1 PWR 电源控制寄存器(PWR_CR)

PWR power control register

偏移地址：0x00

复位值：0x0000 4000（通过从待机模式唤醒进行复位）

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	VOS	Reserved				FPDS	DBP	PLS[2:0]			PVDE	CSBF	CWUF	PDDS	LPDS
	rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:15 保留，必须保持复位值。

位 14 VOS：调压器输出电压级别选择 (Regulator voltage scaling output selection) 此位用来控制内部主调压器的输出电压，以便在器件未以最大频率工作时使性能与功耗实现平衡。

0：级别 2 模式

1：级别 1 模式（复位时的默认值）

位 13:10 保留，必须保持复位值。

位 9 FPDS：停止模式下 Flash 掉电 (Flash power-down in Stop mode)将此位置 1 时，Flash 将在器件进入停止模式后掉电。这样可以降低停止模式的功耗，但会延长重新启动时间。

0：器件进入停止模式时 Flash 不掉电

1：器件进入停止模式时 Flash 掉电

位 8 DBP：禁止备份域写保护 (Disable backup domain write protection)在复位状态下，RCC_BDCR 寄存器、PWR_CSR 寄存器的 BRE 位均受到写访问保护。必须将此位置 1 才能使能对这些寄存器的写访问。

位 7:5 PLS[2:0] ：PVD 级别选择 (PVD level selection)这些位由软件写入，用于选择电压检测器检测的电压阈值

000：2.0 V

001：2.1 V

010：2.3 V

011：2.5 V

100：2.6 V

101：2.7 V

110：2.8 V

111：2.9 V

位 4 PVDE ：使能电源电压检测器 (Power voltage detector enable)此位由软件置 1 和清零。

0：禁止 PVD

1：使能 PVD

位 3 CSBF：将待机标志清零 (Clear standby flag)此位始终读为 0。

0：无操作

1：写 1 将 SBF 待机标志清零。

位 2 CWUF ：将唤醒标志清零 (Clear wakeup flag)此位始终读为 0。

0：无操作

1： 写 1 操作 2 个系统时钟周期后将 WUF 唤醒标志清零

位 1 PDDS：深度睡眠掉电 (Power-down deepsleep)此位由软件置 1 和清零。与 LPDS 位结合使用。

0：器件在 CPU 进入深度睡眠时进入停止模式。调压器状态取决于 LPDS 位。

1：器件在 CPU 进入深度睡眠时进入待机模式。

位 0 LPDS ：深度睡眠低功耗 (Low-power deepsleep)此位由软件置 1 和清零。与 PDDS 位结合使用。

0：停止模式下调压器开启

1：停止模式下调压器进入低功耗模式

01：级别 3 模式

10：级别 2 模式

11：级别 1 模式（复位值）

011：2.5 V

100：2.6 V

101：2.7 V

110：2.8 V

111：2.9 V

7.4.2 PWR 电源控制/ 状态寄存器 (PWR_CSR)

PWR power control/status register

偏移地址：0x04

复位值：0x0000 0000（不通过从待机模式唤醒进行复位）

与标准的 APB 读操作相比，读取此寄存器需要更多的 APB 周期。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	VOSRDY	Reserved				BRE	EWUP	Reserved				BRR	PVDO	SBF	WUF
	r					r	r					r	r	r	r

位 31:15 保留，必须保持复位值。

位 14 VOSRDY：调压器输出分级电压就绪标志 (Regulator voltage scaling output selection ready bit)

0：未就绪

1：就绪

位 13:10 保留，必须保持复位值。

位 9 BRE：使能备份调压器 (Backup regulator enable)将此位置 1 时，使能备份调压器（用于在待机模式和 V BAT 模式下保持备份 SRAM 内容）。如果 BRE 复位，备份调压器关闭。仍可使用备份 SRAM 但在待机模式和 V BAT 模式中其内容将丢失。将此位置 1 后，应用程序必须等待备份调压器就绪标志 (BRR) 置 1，指示在待机模式和 V BAT 模式下会保持写入 RAM 中的数据。

0：禁止备份调压器

1：使能备份调压器

注意：此位不会在器件从待机模式唤醒时复位，也不会通过系统复位或电源复位进行复位。

位 8 EWUP：使能 WKUP 引脚 (Enable WKUP pin)此位由软件置 1 和清零。

0：WKUP 引脚用作通用 I/O。WKUP 引脚上的事件不会把器件从待机模式唤醒。

1：WKUP 用于从待机模式唤醒器件并被强制配置成输入下拉（WKUP 引脚出现上升沿时从待机模式唤醒系统）。

注意：此位通过系统复位进行复位。

位 7:4 保留，必须保持复位值。

位 3 BRR：备份调压器就绪 (Backup regulator ready)由硬件置 1，用以指示备份调压器已就绪。

0：备份调压器未就绪

1：备份调压器就绪

注意：此位不会在器件从待机模式唤醒时复位，也不会通过系统复位或电源复位进行复位。

位 2 PVDO ：PVD 输出 (PVD output)此位通过硬件置 1 和清零。仅当通过 PVDE 位使能 PVD 时此位才有效。

0：V DD 高于 PLS[2:0] 位选择的 PVD 阈值。

1：V DD 低于 PLS[2:0] 位选择的 PVD 阈值。

注意： PVD 在进入待机模式时停止。因此，进入待机模式或执行复位后，此位等于 0 ，直到 PVDE 位置 1 。

位 1 SBF ：待机标志 (Standby flag)此位由硬件置 1，清零则只能通过 POR/PDR（上电复位/掉电复位）或将 PWR_CR 寄存器中的 CSBF 位置 1 来实现。

0：器件未进入待机模式

1：器件在此次复位前进入待机模式

位 0 WUF ：唤醒标志 (Wakeup flag)此位由硬件置 1，清零则只能通过 POR/PDR（上电复位/掉电复位）或将 PWR_CR 寄存器中的 CWUF 位置 1 来实现。

0：未发生唤醒事件

1：收到唤醒事件。

注意：如果使能 WKUP 引脚（将 EWUP 位置 1 ）时 WKUP 引脚已为高电平，系统将检测到另一唤醒事件。

7.5 PWR 寄存器映射

下表对 PWR 寄存器进行了汇总。

表 24. PWR— 寄存器映射和复位值

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

0X00	PWR_CR	Reserved	VOS	Reserved	FPDS	DBP	PLS[2:0]			PVDE	CSBF	CWUF	PDDS	LPDS
	Reset value		1		0	0	0	0	0	0	0	0	0	0
0X04	PWR_CSR	Reserved	VOSRDY	Reserved	BRE	EWUP	Reserved				BRR	PVDO	SBF	WUF
	Reset valu		0		0	0					0	0	0	0

STC MCU

8 复位和时钟控制 (RCC)

8.1 复位

共有三种类型的复位，分别为系统复位、电源复位和备份域复位。

8.1.1 系统复位

除了时钟控制寄存器 CSR 中的复位标志和备份域中的寄存器外，系统复位会将其它全部寄存器都复位为复位值（请参见 图 4）。
只要发生以下事件之一，就会产生系统复位：
1. NRST 引脚低电平（外部复位）
2. 窗口看门狗计数结束（WWDG 复位）
3. 独立看门狗计数结束（IWDG 复位）
4. 软件复位（SW 复位）（请参见 软件复位 ）
5. 低功耗管理复位（请参见 低功耗管理复位 ）

软件复位

可通过查看 RCC 时钟控制和状态寄存器 (RCC_CSR) 中的复位标志确定。
要对器件进行软件复位，必须将 Cortex™-MSTAR 应用中断和复位控制寄存器中的 SYSRESETREQ 位置 1。有关详细信息，请参见 Cortex™-MSTAR 技术参考手册。

低功耗管理复位

引发低功耗管理复位的方式有两种：
1. 进入待机模式时产生复位：
此复位的使能方式是清零用户选项字节中的 nRST_STDBY 位。使能后，只要成功执行进入待机模式序列，器件就将复位，而非进入待机模式。
2. 进入停止模式时产生复位：
此复位的使能方式是清零用户选项字节中的 nRST_STOP 位。使能后，只要成功执行进入停止模式序列，器件就将复位，而非进入停止模式。

8.1.2 电源复位

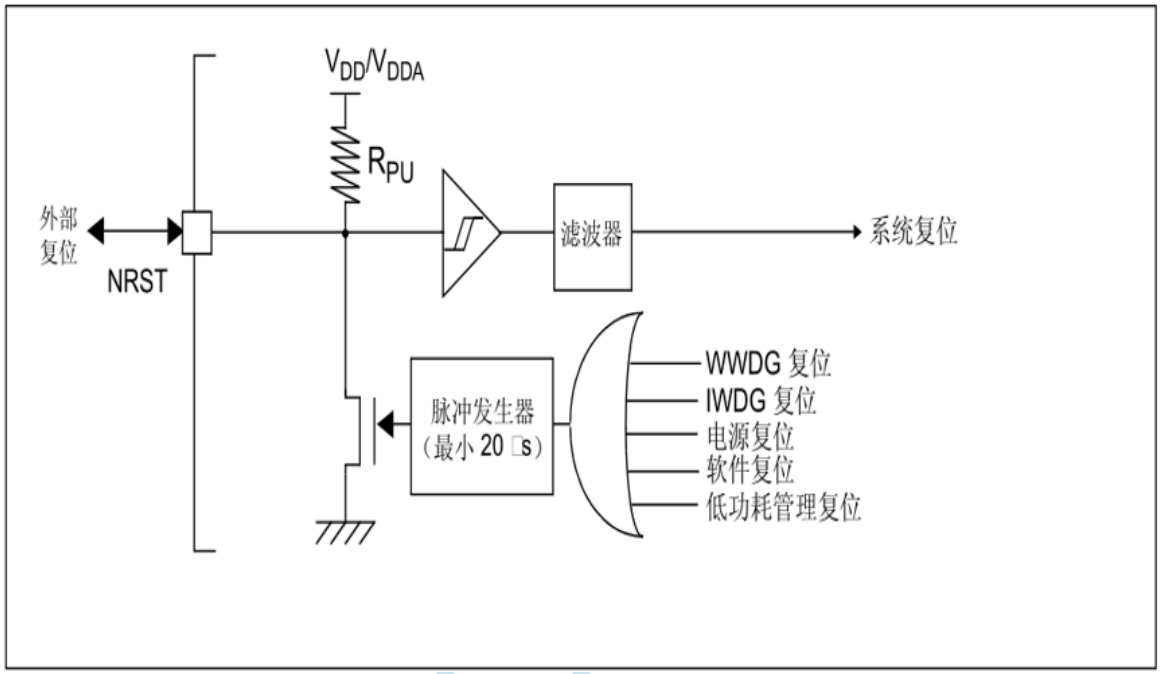
只要发生以下事件之一，就会产生电源复位：
1. 上电/掉电复位（POR/PDR 复位）或欠压 (BOR) 复位
2. 在退出待机模式时
除备份域内的寄存器以外，电源复位会将其它全部寄存器设置为复位值（请参见 图 4）这些源均

STC MCU

作用于 NRST 引脚，该引脚在复位过程中始终保持低电平。RESET 复位入口向量在存储器映射中固定在地址 0x0000_0004 。

芯片内部的复位信号会在 NRST 引脚上输出。脉冲发生器用于保证最短复位脉冲持续时间，可确保每个内部复位源的复位脉冲都至少持续 20 μs。对于外部复位，在 NRST 引脚处于低电平时产生复位脉冲。

图 12. 复位电路简图



备份域具有两个特定的复位，这两个复位仅作用于备份域本身（请参见 图 4）。

8.1.3 备份域复位

备份域复位会将 RCC_BDCR 寄存器复位为各自的复位值。BKPSRAM 不受此复位影响。BKPSRAM 的唯一复位方式是通过 Flash 接口将 Flash 保护等级从 1 切换到 0。

只要发生以下事件之一，就会产生备份域复位：

1. 软件复位，通过将 RCC 备份域控制寄存器 (RCC_BDCR) 中的 BDRST 位置 1 触发。
2. 在电源 V DD 和 V BAT 都已掉电后，其中任何一个又再上电。

8.2 时钟

可以使用三种不同的时钟源来驱动系统时钟 (SYSCLK)：

- HSI 振荡器时钟
- HSE 振荡器时钟
- 主 PLL (PLL) 时钟

对于每个时钟源来说，在未使用时都可单独打开或者关闭，以降低功耗。

时钟控制器为应用带来了高度的灵活性，用户在运行内核和外设时可选择使用外部晶振或者使用振荡器，既可采用最高的频率，也可为 USB 以及 I2S 等需要特定时钟的外设保证合适的频率。
可通过多个预分频器配置 AHB 频率、高速 APB (APB2) 和低速 APB (APB1)。AHB 域的最大频率为 72 MHz。高速 APB2 域的最大允许频率为 72 MHz。低速 APB1 域的最大允许频率为 72 MHz。
除以下时钟外，所有外设时钟均由系统时钟 (SYSCLK) 提供：

- 来自于特定 PLL 输出 (PLL48CLK) 的 USB 时钟 (48 MHz).
- I2S 时钟

  要实现高品质的音频性能，可通过 I2S_CKIN 引脚的外部时钟提供 I2S 时钟。

RCC 向 Cortex 系统定时器 (SysTick) 馈送 8 分频的 AHB 时钟 (HCLK)。SysTick 可使用此时钟作为时钟源，也可使用 HCLK 作为时钟源，具体可在 SysTick 控制和状态寄存器中配置。
定时器时钟频率由硬件自动设置。分为两种情况：
1. 如果 APB 预分频器为 1，定时器时钟频率等于 APB 域的频率。
2. 否则，等于 APB 域的频率的两倍 (×2)。

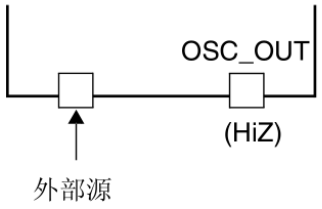
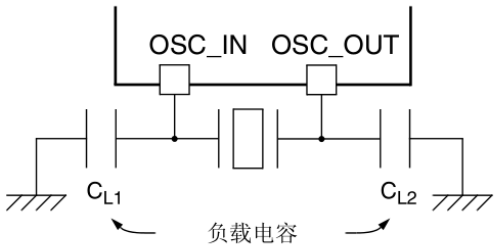
8.2.1 HSE 时钟

高速外部时钟信号 (HSE) 有 2 个时钟源：

- HSE 外部晶振/陶瓷谐振器
- HSE 外部用户时钟

谐振器和负载电容必须尽可能地靠近振荡器的引脚，以尽量减小输出失真和起振稳定时间。负载电容值必须根据所选振荡器的不同做适当调整。

图 14. HSE/LSE 时钟源

<table>
<tr><th></th><th>硬件配置</th></tr>
<tr><td>外部时钟</td><td> 
</td></tr>
<tr><td>晶振/陶瓷谐振器</td><td> 
</td></tr>
</table>

在此模式下，必须提供外部时钟源。此模式通过将 RCC 时钟控制寄存器 (RCC_CR) 中的 HSEBYP 和 HSEON 位置 1 进行选择。必须使用占空比约为 50% 的外部时钟信号（方波、正弦波或三角波）来驱动 OSC_IN 引脚，同时 OSC_OUT 引脚应保持为高阻态 (hi-Z)。请参见 图 14。
外部晶振/ 陶瓷谐振器（HSE 晶振）HSE 的特点是精度非常高。
相关的硬件配置如 图 14 所示。有关详细信息，请参见 数据手册 的电气特性部分。
RCC 时钟控制寄存器 (RCC_CR) 中的 HSERDY 标志指示高速外部振荡器是否稳定。在启动时，硬件将此位置 1 后，此时钟才可以使用。如在 RCC 时钟中断寄存器 (RCC_CIR) 中使能中断，则可产生中断。
HSE 晶振可通过 RCC 时钟控制寄存器 (RCC_CR) 中的 HSEON 位打开或关闭。

8.2.2 HSI 时钟

HSI 时钟信号由内部 16 MHz RC 振荡器生成，可直接用作系统时钟，或者用作 PLL 输入。HSI RC 振荡器的优点是成本较低（无需使用外部组件）。此外，其启动速度也要比 HSE 晶振块，但即使校准后，其精度也不及外部晶振或陶瓷谐振器。
校准因为生产工艺不同，不同芯片的 RC 振荡器频率也不同，因此会对每个器件进行出厂校准，达到 TA = 25 ° C 时 1% 的精度。
复位后，工厂校准值将加载到 RCC 时钟控制寄存器 (RCC_CR) 的 HSICAL[7:0] 位中。
如果应用受到电压或温度变化影响，则这可能也会影响到 RC 振荡器的速度。用户可通过 RCC 时钟控制寄存器 (RCC_CR) 中的 HSITRIM[4:0] 位对 HSI 频率进行微调。
RCC 时钟控制寄存器 (RCC_CR) 中的 HSIRDY 标志指示 HSI RC 是否稳定。在启动时，硬件将此位置 1 后，HSI 才可以使用。
HSI RC 可通过 RCC 时钟控制寄存器 (RCC_CR) 中的 HSION 位打开或关闭。
HSI 信号还可作为备份时钟源（辅助时钟）使用，以防 HSE 晶振发生故障。

8.2.3 LSE 时钟

LSE 晶振是 32 kHz 低速外部 (LSE) 晶振或陶瓷谐振器。
LSE 晶振通过 RCC 备份域控制寄存器 (RCC_BDCR) 中的 LSEON 位打开和关闭。
RCC 备份域控制寄存器 (RCC_BDCR) 中的 LSERDY 标志指示 LSE 晶振是否稳定。在启动时，硬件将此位置 1 后，LSE 晶振输出时钟信号才可以使用。如在 RCC 时钟中断寄存器(RCC_CIR) 中使能中断，则可产生中断。
外部源（LSE 旁路）在此模式下，必须提供外部时钟源，最高频率不超过 1 MHz。此模式通过将 RCC 备份域控制寄存器 (RCC_BDCR) 中的 LSEBYP 和 LSEON 位置 1 进行选择。必须使用占空比约为 50%的外部时钟信号（方波、正弦波或三角波）来驱动 OSC32_IN 引脚，同时 OSC32_OUT 引脚应保持为高阻态 (Hi-Z)。请参见图 14。

8.2.4 LSI 时钟

LSI RC 可作为低功耗时钟源在停机和待机模式下保持运行，供独立看门狗 (IWDG) 和自动唤醒单元 (AWU) 使用。时钟频率在 32 kHz 左右。
LSI RC 可通过 RCC 时钟控制和状态寄存器 (RCC_CSR) 中的 LSION 位打开或关闭。
RCC 时钟控制和状态寄存器 (RCC_CSR) 中的 LSIRDY 标志指示低速内部振荡器是否稳定。

在启动时，硬件将此位置 1 后，此时钟才可以使用。如在 RCC 时钟中断寄存器 (RCC_CIR) 中使能中断，则可产生中断。

8.2.5 系统时钟 (SYSCLK) 选择

在系统复位后，默认系统时钟为 HSI。在直接使用 HSI 时钟源来作为系统时钟时，该时钟源无法停止。

只有在目标时钟源已就绪时，才可从一个时钟源切换到另一个。如果选择尚未就绪的时钟源，则切换在该时钟源就绪时才会进行。RCC 时钟控制寄存器 (RCC_CR) 中的状态位指示哪个（些）时钟已就绪，以及当前哪个时钟正充当系统时钟。

8.2.6 时钟

8.2.7 看门狗时钟

如果独立看门狗 (IWDG) 已通过硬件选项字节或软件设置的方式启动，则 LSI 振荡器将强制打开且不可禁止。在 LSI 振荡器稳定后，时钟将提供给 IWDG。

8.2.8 时钟输出功能

共有一个微控制器时钟输出 (MCLK) 引脚：

● MCLK

用户可通过可配置的预分配器（从 1 到 5）向 MCLK 引脚 输出四个不同的时钟源：

— HSI 时钟

— LSE 时钟

— HSE 时钟

所需的时钟源通过 RCC 时钟配置寄存器 (RCC_CFGR) 中的 MCO1PRE[2:0] 和 MCO1[1:0]位选择。

8.2.9 基于 PWM5/PWM11 的内部/ 外部时钟测量

所有时钟源的频率都可通过对 PWM5 channel4 和 PWM11 channel1 的输入捕获进行间接测量，如图 15 所示。

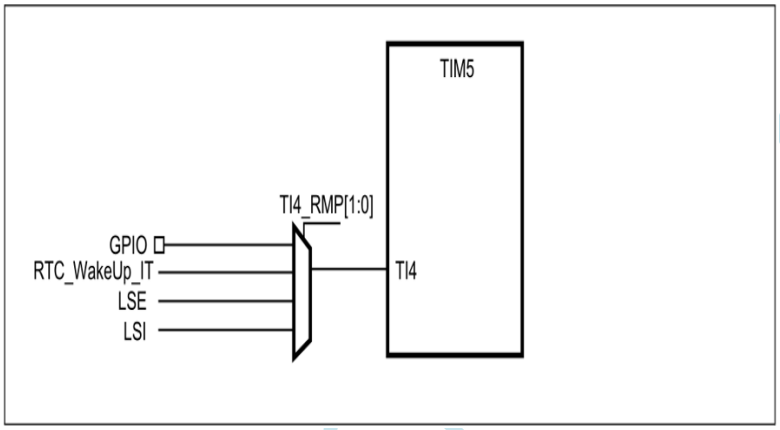
基于 PWM5 channel4 的内部/ 外部时钟测量

PWM5 具有一个输入复用器，可选择输入捕获是由 I/O 触发还是由内部时钟触发。此选择通过 PWM5_OR 寄存器的 TI4_RMP [1:0] 位执行。

将 LSE 连接到 channel4 输入捕获的主要目的是为了精确测量 HSI（这需要将 HSI 用作系统时钟

源)。借助 LSE 信号连续边沿之间的 HSI 时钟计数数量,即可对内部时钟周期进行测量。利用 LSE 的高精度（通常为几十 ppm），用户能以同一分辨率测定时钟频率，并可通过对时钟源进行微调来补偿由生产工艺造成的和/或与温度和电压相关的频率偏差。
HSI 振荡器设有针对此目的的专用校准位，且支持用户访问。
其基本原理是基于相对的测量（例如，HSI/LSE 比）：因此，精度与两个时钟源之比紧密相关。比率越大，测量效果越好。
同时也可测量 LSI 频率：这对于没有晶振的应用场合非常实用。超低功耗 LSI 振荡器具有较大的生产工艺偏差：通过测量该振荡器与 HSI 时钟源的比率，可以借助 HSI 精度为其测定频率。
LSI 频率通过以下步骤测量：
1. 使能 PWM5 定时器并将 channel4 配置为输入捕获模式。
2. 将 PWM5_OR 寄存器中的 TI4_RMP 位设置为 0x01，以在内部将 LSI 时钟连接到 PWM5 channel4 输入捕获来实现校准。
3. 通过 PWM5 捕获/比较 4 事件或中断测量 LSI 时钟频率。
4. 使用测得的 LSI 频率来按照所需的时基，并且/或者用其来计算 IWDG 超时。

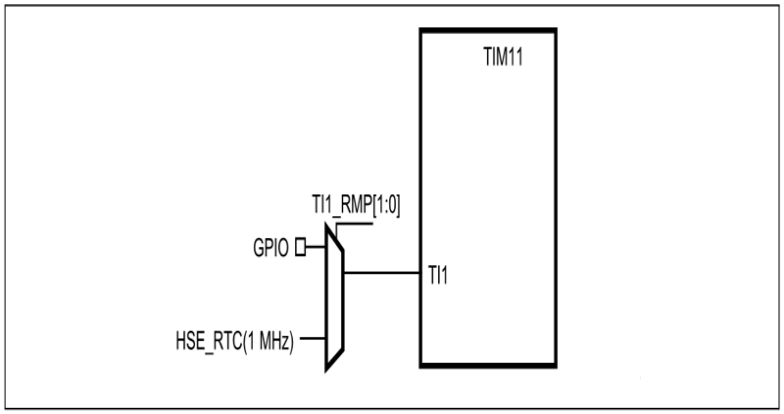
图 15. PWM5 在输入捕获模式下的频率测量



基于 PWM11 channel1 的内部/外部时钟测量

PWM11 具有一个输入复用器，可选择输入捕获是由 I/O 触发还是由内部时钟触发。此选择通过 PWM11_OR 寄存器的 TI1_RMP [1:0] 位执行。HSE_RTC 时钟（由一个可编程预分频器分频的 HSE）连接到通道 1 输入捕获，以粗略指示外部晶振频率。这要求 HSI 为系统时钟源。此功能非常实用,例如可借此测定谐波频率或分谐波频率(—50/+100% 偏差),进而确保符合 IEC 60730/IEC 61335 标准。

图 16. PWM11 在输入捕获模式下的频率测量



8.3 RCC 寄存器

有关寄存器说明中使用的缩写，请参见 第 1.1 节：寄存器相关缩写词列表 。

8.3.1 RCC 时钟控制寄存器 (RCC_CR)

RCC clock control register
偏移地址：0x00
复位值：0x0000 XX83，其中 X 未定义。
访问：无等待周期，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												CSS ON	HSE BYP	HSE RDY	HSE ON
												rw	rw	r	rw

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

位 31:20 保留，必须保持复位值。

位 19 CSSON：时钟安全系统使能 (Clock security system enable)

由软件置 1 和清零，用于使能时钟安全系统。当 CSSON 置位时，时钟监测器将在 HSE 振荡器就绪时由硬件使能，并在检出振荡器故障时由硬件禁止。

0：时钟安全系统关闭（时钟监测器关闭）

1：时钟安全系统打开（如果 HSE 振荡器稳定，则时钟监测器打开；如果不稳定，则关闭）

位 18 HSEBYP：HSE 时钟旁路 (HSE clock bypass)

由软件置 1 和清零，用于用外部时钟旁路振荡器。外部时钟必须通过 HSEON 位使能才能为器件使用。

HSEBYP 只有在 HSE 振荡器已禁止的情况下才可写入。

0：不旁路 HSE 振荡器
1：外部时钟旁路 HSE 振荡器

位 17 HSERDY：HSE 时钟就绪标志 (HSE clock ready flag)
由硬件置 1,用以指示 HSE 振荡器已稳定。在将 HSEON 位清零后,HSERDY 将在 6 个 HSE 振荡器时钟周期后转为低电平。
0：HSE 振荡器未就绪
1：HSE 振荡器已就绪

位 16 HSEON：HSE 时钟使能 (HSE clock enable)
由软件置 1 和清零。
由硬件清零，用于在进入停机或待机模式时停止 HSE 振荡器。如果 HSE 振荡器直接或间接用于作为系统时钟，则此位不可复位。
0：HSE 振荡器关闭
1：HSE 振荡器打开

位 15:8 HSICAL[7:0]：内部高速时钟校准 (Internal high-speed clock calibration)
这些位在启动时自动初始化。

位 7:3 HSITRIM[4:0]：内部高速时钟微调 (Internal high-speed clock trimming)
通过这些位，可在 HSICAL[7:0] 位基础上实现可由用户编程的微调值。可通过编程使其适应电压和温度的差异，使内部 HSI RC 的频率更为准确。

位 2 保留，必须保持复位值。

位 1 HSIRDY：内部高速时钟就绪标志 (Internal high-speed clock ready flag)
由硬件置 1，用以指示 HSI 振荡器已稳定。在将 HSION 位清零后，HSIRDY 将在 6 个 HSI 时钟周期后转为低电平。
0：HSI 振荡器未就绪
1：HSI 振荡器已就绪

位 0 HSION：内部高速时钟使能 (Internal high-speed clock enable)
由软件置 1 和清零。
由硬件置 1，用于在脱离停机或待机模式时或者在直接或间接用作系统时钟的 HSE 振荡器发生故障时强制 HSI 振荡器打开。如果 HSI 直接或间接用于作 为系统时钟，则此位不可清零。
0：HSI 振荡器关闭
1：HSI 振荡器打开

8.3.2 RCC PLL 配置寄存器 (RCC_PLLCFGR)

保留。

8.3.3 RCC 时钟配置寄存器 (RCC_CFGR)

RCC clock configuration register
偏移地址：0x08
复位值：0x0000 0000
访问：0 ≤ 等待周期 ≤2，按字、半字和字节访问
只有在时钟源切换期间进行访问时才会插入 1 或 2 个等待周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					MCO1 PRE[2:0]			I2SSCR	MCO1		RTCPRE[4:0]				
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PPRE2[2:0]			PPRE1[2:0]			Reserved		HPREP[3:0]				SWS1	SWS0	SW1	SW0
w	w	w	w	w	w			rw	rw	rw	w	r	r	rw	rw

位 31:27 保留

位 24:26 MCO1PRE ： MCO1 预分频器 (MCO1 prescaler)

由软件置 1 和清零，用于配置 MCO1 的预分频器。对此预分频器进行修改可能会对 MCO1 造成干扰。强烈建议仅在复位后且在使能外部振荡器和 PLL 之前进行此分频器的更改。

0xx：无分频

100：2 分频

101：3 分频

110：4 分频

111：5 分频

位 23 I2SSRC：I2S 时钟选择 (I2S clock selection)

由软件置 1 和清零。通过此位可在 系统 时钟和外部时钟之间选择 I2S 时钟源。强烈建议仅在复位之后使能 I2S 模块之前对此位进行更改。

0：系统 时钟用作 I2S 时钟源

1：在 I2S_CKIN 引脚上映射的外部时钟用作 I2S 时钟源

位 22:21 MCO1 ： 微控制器时钟输出 1 (Microcontroller clock output 1)

由软件置 1 和清零。时钟源选择可能会造成对 MCO1 的干扰。强烈建议仅在复位后且在使能外部振荡器和 PLL 之前来配置这些位。

00：选择 HSI 时钟输出到 MCO1 引脚

01：选择 LSE 振荡器输出到 MCO1 引脚

10：选择 HSE 振荡器时钟输出到 MCO1 引脚

11：选择 PLL 时钟输出到 MCO1 引脚

位 20:16 RTCPRE ： 适用于 RTC 时钟的 HSE 分频系数

由软件置 1 和清零，用于对 HSE 时钟输入时钟进行分频，进而为 RTC 生成 1 MHz 的时钟。

注意： 软件必须正确设置这些位，确保提供给 RTC 的时钟为 1 MHz。在选择 RTC 时钟源之前必须配置这些位。

00000：无时钟

00001：无时钟

00010：HSE/2

00011：HSE/3

00100：HSE/4

...

11110：HSE/30

11111：HSE/31

位 15:13 PPRE2 ： APB 高速预分频器 (APB2) (APB high-speed prescaler (APB2))

由软件置位和清零，用于控制 APB 高速时钟分频系数。

小心： 软件必须正确设置这些位，使其在此域中不超过 84 MHz。在 PPRE2 写入后，时钟将通过 1 AHB 到 16 AHB 周期新预分频系数进行分频。

0xx：AHB 时钟不分频
100：AHB 时钟 2 分频
101：AHB 时钟 4 分频
110：AHB 时钟 8 分频
111：AHB 时钟 16 分频

位 12:10 PPRE1 ： APB 低速预分频器 (APB1) (APB Low speed prescaler (APB1))
由软件置位和清零，用于控制 APB 低速时钟分频系数。
注意： 软件必须正确设置这些位，使其在此域中不超过 42 MHz。在 PPRE1 写入后，时钟将通过 1 AHB 到 16 AHB 周期新预分频系数进行分频。
0xx：AHB 时钟不分频
100：AHB 时钟 2 分频
101：AHB 时钟 4 分频
110：AHB 时钟 8 分频
111：AHB 时钟 16 分频

位 9:8 保留，必须保持复位值。

位 7:4 HPRE ： AHB 预分频器 (AHB prescaler)
由软件置位和清零，用于控制 AHB 时钟分频系数。
注意： 在 HPRE 写入后，时钟将通过 1 AHB 到 16 AHB 周期新预分频系数进行分频。
0xxx：系统时钟不分频
1000：系统时钟 2 分频
1001：系统时钟 4 分频
1010：系统时钟 8 分频
1011：系统时钟 16 分频
1100：系统时钟 64 分频
1101：系统时钟 128 分频
1110：系统时钟 256 分频
1111：系统时钟 512 分频

位 3:2 SWS ： 系统时钟切换状态 (System clock switch status)
由硬件置 1 和清零，用于指示用作系统时钟的时钟源。
00：HSI 振荡器用作系统时钟
01：HSE 振荡器用作系统时钟
10：PLL 用作系统时钟
11：不适用

位 1:0 SW ： 系统时钟切换 (System clock switch)
由软件置 1 和清零，用于选择系统时钟源。
由硬件置 1，用于在退出停机或待机模式时或者在直接或间接用作系统时钟的 HSE 振荡器发生故障时强制 HSI 的选择。
00：选择 HSI 振荡器作为系统时钟
01：选择 HSE 振荡器作为系统时钟
10：选择 PLL 作为系统时钟
11：不允许

8.3.4 RCC 钟中断寄存器 (RCC_CIR)

RCC clock interrupt register

偏移地址：0x0C
复位值：0x0000 0000
访问：无等待周期，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CSSC	Reserved			HSE RDYC	HSI RDYC	LSE RDYC	LSI RDYC
								w				w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				HSE RDYIE	HSI RDYIE	LSE RDYIE	LSI RDYIE	CSS	Reserved			HSE RDYF	HSI RDYF	LSE RDYF	LSI RDYF
				rw	rw	rw	rw	r				r	r	r	r

位 31:24 保留，必须保持复位值。
位 23 CSSC ：时钟安全系统中断清零 (Clock security system interrupt clear)
此位由软件置 1，用于将 CSSF 标志清零。
0：无操作
1：将 CSSF 标志清零
位 22 保留，必须保持复位值。
位 21
位 20

位 19 HSERDYC ：HSE 就绪中断清零 (HSE ready interrupt clear)
此位由软件置 1，用于将 HSERDYF 标志清零。
0：无操作
1：清零 HSERDYF
位 18 HSIRDYC ：HSI 就绪中断清零 (HSI ready interrupt clear)
此位由软件置 1，用于将 HSIRDYF 标志清零。
0：无操作
1：清零 HSIRDYF
位 17 LSERDYC ：LSE 就绪中断清零 (LSE ready interrupt clear)
此位由软件置 1，用于将 LSERDYF 标志清零。
0：无操作
1：清零 LSERDYF
位 16 LSIRDYC ：LSI 就绪中断清零 (LSI ready interrupt clear)
此位由软件置 1，用于将 LSIRDYF 标志清零。
0：无操作
1：清零 LSIRDYF
位 15:12 保留，必须保持复位值。
位 11 HSERDYIE ：HSE 就绪中断使能 (HSI ready interrupt enable)
由软件置 1 和清零，用于使能/禁止由 HSE 振荡器稳定所引起的中断。
0：禁止 HSE 就绪中断
1：使能 HSE 就绪中断
位 10 HSIRDYIE ：HSI 就绪中断使能 (HSI ready interrupt enable)
由软件置 1 和清零，用于使能/禁止由 HSI 振荡器稳定所引起的中断。
0：禁止 HSI 就绪中断

STC MCU

1：使能 HSI 就绪中断

位 9 LSERDYIE ：LSE 就绪中断使能 (LSE ready interrupt enable)

由软件置 1 和清零，用于使能/禁止由 LSE 振荡器稳定所引起的中断。

0：禁止 LSE 就绪中断

1：使能 LSE 就绪中断

位 8 LSIRDYIE ：LSI 就绪中断使能 (LSI ready interrupt enable)

由软件置 1 和清零，用于使能/禁止由 LSI 振荡器稳定所引起的中断。

0：禁止 LSI 就绪中断

1：使能 LSI 就绪中断

位 7 CSSF ：时钟安全系统中断标志 (Clock security system interrupt flag)

当在 HSE 振荡器中检出故障时由硬件置 1。

CSSC 位置 1 时由软件清零。

0：当前未因 HSE 时钟故障而引起时钟安全中断

1：因 HSE 时钟故障而引起时钟安全中断

位 6 保留，必须保持复位值。

位 5 保留，必须保持复位值

位 4 保留，必须保持复位值

位 3 HSERDYF ：HSE 就绪中断标志 (HSE ready interrupt flag)

当外部高速时钟变为稳定且 HSERDYDIE 置 1 时由硬件置 1。

HSERDYC 位置 1 时由软件清零。

0：当前未因 HSE 振荡器引起时钟就绪中断

1：因 HSE 振荡器引起时钟就绪中断

位 2 HSIRDYF ：HSI 就绪中断标志 (HSI ready interrupt flag)

当内部高速时钟变为稳定且 HSIRDYDIE 置 1 时由硬件置 1。

HSIRDYC 位置 1 时由软件清零。

0：当前未因 HSI 振荡器引起时钟就绪中断

1：因 HSI 振荡器引起时钟就绪中断

位 1 LSERDYF ：LSE 就绪中断标志 (LSE ready interrupt flag)

当外部低速时钟变为稳定且 LSERDYDIE 置 1 时由硬件置 1。

LSERDYC 位置 1 时由软件清零。

0：当前未因 LSE 振荡器引起时钟就绪中断

1：因 LSE 振荡器引起时钟就绪中断

位 0 LSIRDYF ：LSI 就绪中断标志 (LSI ready interrupt flag)

当内部低速时钟变为稳定且 LSIRDYDIE 置 1 时由硬件置 1。

LSIRDYC 位置 1 时由软件清零。

0：当前未因 LSI 振荡器引起时钟就绪中断

1：因 LSI 振荡器引起时钟就绪中断

8.3.5 RCC AHB1 外设复位寄存器 (RCC_AHB1RSTR)

RCC AHB1 peripheral reset register

偏移地址：0x10

复位值：0x0000 0000

访问：无等待周期，按字、半字和字节访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									DMA2RST	DMA1RST	Reserved				
									rw	rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CRCRST	Reserved								GPIODRST	GPIOCRST	GPIOBRST	GPIOARST
			rw									rw	rw	rw	rw

位 31:23 保留，必须保持复位值。

位 22 DMA2RST ： DMA2 复位 (DMA2 reset)
由软件置 1 和清零。
0：不复位 DMA2
1：复位 DMA2

位 21 DMA1RST ： DMA1 复位 (DMA1 reset)
由软件置 1 和清零。
0：不复位 DMA1
1：复位 DMA1

位 20:13 保留，必须保持复位值。

位 12 CRCRST ： CRC 复位 (CRC reset)
由软件置 1 和清零。
0：不复位 CRC
1：复位 CRC

位 11:4 保留，必须保持复位值。

位 3 GPIODRST ： IO 端口 D 复位 (IO port D reset)
由软件置 1 和清零。
0：不复位 IO 端口 D
1：复位 IO 端口 D

位 2 GPIOCRST ： IO 端口 C 复位 (IO port C reset)
由软件置 1 和清零。
0：不复位 IO 端口 C
1：复位 IO 端口 C

位 1 GPIOBRST ： IO 端口 B 复位 (IO port B reset)
由软件置 1 和清零。
0：不复位 IO 端口 B
1：复位 IO 端口 B

位 0 GPIOARST ： IO 端口 A 复位 (IO port A reset)
由软件置 1 和清零。
0：不复位 IO 端口 A
1：复位 IO 端口 A

8.3.6 RCC AHB2 外设复位寄存器 (RCC_AHB2RSTR)

STC MCU

RCC AHB2 peripheral reset register
偏移地址：0x14
复位值：0x0000 0000
访问：无等待周期，按字、半字和字节访问
该寄存器保留，待补充

8.3.7 RCC AHB3 外设复位寄存器 (RCC_AHB3RSTR)

RCC AHB3 peripheral reset register
偏移地址：0x18
复位值：0x0000 0000
访问：无等待周期，按字、半字和字节访问。

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

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

位 31:1 保留，必须保持复位值。
位 0 FSMCRST ：灵活的静态存储控制器模块复位 (Flexible static memory controller module reset)
由软件置 1 和清零。
0：不复位 FSMC 模块
1：复位 FSMC 模块

8.3.8 RCC APB1 外设复位寄存器 (RCC_APB1RSTR)

RCC APB1 peripheral reset register
偏移地址：0x20
复位值：0x0000 0000
访问：无等待周期，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Rserved			PWR RST	Rserved					I2C2 RST	I2C1 RST	Reserved	UART4 RST	UART3 RST	USART2 RST	Res.
			rw						rw	rw		rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rserved	SPI2 RST	Rserved		WWDG RST	Rserved		PWM 14RST	PWM 13RST	PWM 12RST	PWM 7RST	PWM 6RST	PWM 5RST	PWM 4RST	PWM 3RST	PWM 2RST
	rw			rw			rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:29 保留，必须保持复位值。
位 28 PWRRST ： 电源接口复位 (Power interface reset)
由软件置 1 和清零。
0：不复位电源接口
1：复位电源接口
位 27：23 保留，必须保持复位值。
位 22 I2C2RST ： I2C2 复位 (I2C2 reset)
由软件置 1 和清零。
0：不复位 I2C2
1：复位 I2C2
位 21 I2C1RST ： I2C1 复位 (I2C1 reset)
由软件置 1 和清零。
0：不复位 I2C1
1：复位 I2C1
位 20
位 19 UART4RST ： USART4 复位 (USART4 reset)
由软件置 1 和清零。
0：不复位 UART4
1：复位 UART4
位 18 USART3RST ： USART3 复位 (USART3 reset)
由软件置 1 和清零。
0：不复位 USART3
1：复位 USART3
位 17 USART2RST ： USART2 复位 (USART2 reset)
由软件置 1 和清零。
0：不复位 USART2
1：复位 USART2
位 16 保留，必须保持复位值。
位 15
位 14 SPI2RST ： SPI2 复位 (SPI2 reset)
由软件置 1 和清零。
0：不复位 SPI2
1：复位 SPI2
位 13:12 保留，必须保持复位值。

位 11 WWDGRST ： 窗口看门狗复位 (Window watchdog reset)
由软件置 1 和清零。
0：不复位窗口看门狗
1：复位窗口看门狗
位 10:9 保留，必须保持复位值。
位 8 PWM14RST ： PWM14 复位 (PWM14 reset)
由软件置 1 和清零。
0：不复位 PWM14
1：复位 PWM14
位 7 PWM13RST ： PWM13 复位 (PWM13 reset)
由软件置 1 和清零。

STC MCU

0：不复位 PWM13
1：复位 PWM13

位 6 PWM12RST ： PWM12 复位 (PWM12 reset)
由软件置 1 和清零。
0：不复位 PWM12
1：复位 PWM12

位 5 PWM7RST ： PWM7 复位 (PWM7 reset)
由软件置 1 和清零。
0：不复位 PWM7
1：复位 PWM7

位 4 PWM6RST ： PWM6 复位 (PWM6 reset)
由软件置 1 和清零。
0：不复位 PWM6
1：复位 PWM6

位 3 PWM5RST ： PWM5 复位 (PWM5 reset)
由软件置 1 和清零。
0：不复位 PWM5
1：复位 PWM5

位 2 PWM4RST ： PWM4 复位 (PWM4 reset)
由软件置 1 和清零。
0：不复位 PWM4
1：复位 PWM4

位 1 PWM3RST ： PWM3 复位 (PWM3 reset)
由软件置 1 和清零。
0：不复位 PWM3
1：复位 PWM3

位 0 PWM2RST ： PWM2 复位 (PWM2 reset)
由软件置 1 和清零。
0：不复位 PWM2
1：复位 PWM2

8.3.9 RCC APB2 外设复位寄存器外设复位寄存器 (RCC_APB2RSTR)

RCC APB2 peripheral reset register
偏移地址：0x24
复位值：0x0000 0000
访问：无等待周期，按字、半字和字节访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													PWM11RST	PWM10ST	PWM9RST
													rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rserved	SYSCFG RST	Reserved	SPI1 RST	Reserved			ADC1 RST	Reserved			USART1 RST	Reserved		PWM8 RST	PWM1 RST
	rw		rw				rw				rw			rw	rw

位 31:19 保留，必须保持复位值。

位 18 PWM11RST ： PWM11 复位 (PWM11 reset)

由软件置 1 和清零。

0：不复位 PWM11

1：复位 PWM14

位 17 PWM10RST ： PWM10 复位 (PWM10 reset)

由软件置 1 和清零。

0：不复位 PWM10

1：复位 PWM10

位 16 PWM9RST ： PWM9 复位 (PWM9 reset)

由软件置 1 和清零。

0：不复位 PWM9

1：复位 PWM9

位 15 保留，必须保持复位值。

位 14 SYSCFGRST ： 系统配置控制器复位 (System configuration controller reset)

由软件置 1 和清零。

0：不复位系统配置控制器

1：复位系统配置控制器

位 13 保留，必须保持复位值。

位 12 SPI1RST ： SPI1 复位 (SPI1 reset)

由软件置 1 和清零。

0：不复位 SPI1

1：复位 SPI1

位 11:9 保留，必须保持复位值。

8.3.10 RCC AHB1 外设时钟使能寄存器 (RCC_AHB1ENR)

RCC AHB1 peripheral clock enable register

偏移地址：0x30

复位值：0x0010 0000

访问：无等待周期，按字、半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									DMA2 EN	DMA1 EN	Reserved				
									rw	rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CRCEN	Reserved								GPIOD EN	GPIOC EN	GPIOB EN	GPIOA EN

	rw		rw	rw	rw	rw

位 31:23 保留，必须保持复位值。

位 22 DMA2EN ： DMA2 时钟使能 (DMA2 clock enable)

由软件置 1 和清零。

0：禁止 DMA2 时钟

1：使能 DMA2 时钟

位 21 DMA1EN ： DMA1 时钟使能 (DMA1 clock enable)

由软件置 1 和清零。

0：禁止 DMA1 时钟

1：使能 DMA1 时钟

位 20:13 保留，必须保持复位值。

位 12 CRCEN ： CRC 时钟使能 (CRC clock enable)

由软件置 1 和清零。

0：禁止 CRC 时钟

1：使能 CRC 时钟

位 11:4 保留，必须保持复位值。

位 3 GPIODEN ： IO 端口 D 时钟使能 (IO port D clock enable)

由软件置 1 和清零。

0：禁止 IO 端口 D 时钟

1：使能 IO 端口 D 时钟

位 2 GPIOCEN ： IO 端口 C 时钟使能 (IO port C clock enable)

由软件置 1 和清零。

0：禁止 IO 端口 C 时钟

1：使能 IO 端口 C 时钟

位 1 GPIOBEN ： IO 端口 B 时钟使能 (IO port B clock enable)

由软件置 1 和清零。

0：禁止 IO 端口 B 时钟

1：使能 IO 端口 B 时钟

位 0 GPIOAEN ： IO 端口 A 时钟使能 (IO port A clock enable)

由软件置 1 和清零。

0：禁止 IO 端口 A 时钟

1：使能 IO 端口 A 时钟

8.3.11 RCC AHB3 外设时钟使能寄存器 (RCC_AHB3ENR)

RCC AHB3 peripheral clock enable register

偏移地址：0x38

复位值：0x0000 0000

访问：无等待周期，按字、半字和字节访问。

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

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

位 31:1 保留，必须保持复位值。

位 0 FSMCEN ： 灵活的静态存储控制器模块时钟使能 (Flexible static memory controller clock enable)

由软件置 1 和清零。

0：禁止 FSMC 模块时钟

1：使能 FSMC 模块时钟

8.3.12 RCC APB1 外设时钟使能寄存器外设时钟使能寄存器 (RCC_APB1ENR)

RCC APB1 peripheral clock enable register

偏移地址：0x40

复位值：0x0000 0000

访问：无等待周期，按字、半字和字节访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			PWREN	Reserved					I2C2EN	I2C1EN	Reserved	UART4EN	USART3EN	USART2EN	Reserved
			rw						rw	rw		rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SPI2EN	Reserved		WWDGEN	Reserved		PWM14EN	PWM13EN	PWM12EN	PWM7EN	PWM6EN	PWM5EN	PWM4EN	PWM3EN	PWM2EN
	rw			rw			rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28 PWREN ： 电源接口时钟使能 (Power interface clock enable)

由软件置 1 和清零。

0：禁止电源接口时钟

1：使能电源接口时钟

位 27：23 保留，必须保持复位值。

位 22 I2C2EN ： I2C2 时钟使能 (I2C2 clock enable)

由软件置 1 和清零。

0：禁止 I2C2 时钟

1：使能 I2C2 时钟

位 21 I2C1EN ： I2C1 时钟使能 (I2C1 clock enable)

由软件置 1 和清零。
0：禁止 I2C1 时钟
1：使能 I2C1 时钟

位 20

位 19 UART4EN ：UART4 时钟使能 (UART4 clock enable)
由软件置 1 和清零。
0：禁止 UART4 时钟
1：使能 UART4 时钟

位 18 USART3EN ：USART3 时钟使能 (USART3 clock enable)
由软件置 1 和清零。
0：禁止 USART3 时钟
1：使能 USART3 时钟

位 17 USART2EN ：USART2 时钟使能 (USART2 clock enable)
由软件置 1 和清零。
0：禁止 USART2 时钟
1：使能 USART2 时钟

位 16 保留，必须保持复位值。

位 15 保留，必须保持复位值。

位 14 SPI2EN ：SPI2 时钟使能 (SPI2 clock enable)
由软件置 1 和清零。
0：禁止 SPI2 时钟
1：使能 SPI2 时钟

位 13:12 保留，必须保持复位值。

位 11 WWDGEN ：窗口看门狗时钟使能 (Window watchdog clock enable)
由软件置 1 和清零。
0：禁止窗口看门狗时钟
1：使能窗口看门狗时钟

位 10:9 保留，必须保持复位值。

位 8 PWM14EN ：PWM14 时钟使能 (PWM14 clock enable)
由软件置 1 和清零。
0：禁止 PWM14 时钟
1：使能 PWM14 时钟

位 7 PWM13EN ：PWM13 时钟使能 (PWM13 clock enable)
由软件置 1 和清零。
0：禁止 PWM13 时钟
1：使能 PWM13 时钟

位 6 PWM12EN ：PWM12 时钟使能 (PWM12 clock enable)
由软件置 1 和清零。
0：禁止 PWM12 时钟
1：使能 PWM12 时钟

位 5 PWM7EN ：PWM7 时钟使能 (PWM7 clock enable)
由软件置 1 和清零。
0：禁止 PWM7 时钟
1：使能 PWM7 时钟

位 4 PWM6EN ：PWM6 时钟使能 (PWM6 clock enable)

由软件置 1 和清零。
0：禁止 PWM6 时钟
1：使能 PWM6 时钟

位 3 PWM5EN ： PWM5 时钟使能 (PWM5 clock enable)
由软件置 1 和清零。
0：禁止 PWM5 时钟
1：使能 PWM5 时钟

位 2 PWM4EN ： PWM4 时钟使能 (PWM4 clock enable)
由软件置 1 和清零。
0：禁止 PWM4 时钟
1：使能 PWM4 时钟

位 1 PWM3EN ： PWM3 时钟使能 (PWM3 clock enable)
由软件置 1 和清零。
0：禁止 PWM3 时钟
1：使能 PWM3 时钟

位 0 PWM2EN ： PWM2 时钟使能 (PWM2 clock enable)
由软件置 1 和清零。
0：禁止 PWM2 时钟
1：使能 PWM2 时钟

8.3.13 RCC APB2 外设时钟使能寄存器外设时钟使能寄存器 (RCC_APB2ENR)

RCC APB2 peripheral clock enable register
偏移地址：0x44
复位值：0x0000 0000
访问：无等待周期，按字、半字和字节访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													PWM11 EN	PWM10 EN	PWM9 EN
													rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SYSCFG EN	Reserved	SPI1 EN	Reserved			ADC1 EN	Reserved			USART1 EN	Reserved		PWM8 EN	PWM1 EN
	rw		rw				rw				rw			rw	rw

位 31:19 保留，必须保持复位值。
位 18 PWM11EN ： PWM11 时钟使能 (PWM11 clock enable)
由软件置 1 和清零。
0：禁止 PWM11 时钟
1：使能 PWM11 时钟

位 17 PWM10EN ： PWM10 时钟使能 (PWM10 clock enable)

由软件置 1 和清零。

0：禁止 PWM10 时钟

1：使能 PWM10 时钟

位 16 PWM9EN ： PWM9 时钟使能 (PWM9 clock enable)

由软件置 1 和清零。

0：禁止 PWM9 时钟

1：使能 PWM9 时钟

位 15 保留，必须保持复位值。

位 14 SYSCFGEN ： 系统配置控制器时钟使能 (System configuration controller clock enable)

由软件置 1 和清零。

0：禁止系统配置控制器时钟

1：使能系统配置控制器时钟

位 13 保留，必须保持复位值。

位 12 SPI1EN ： SPI1 时钟使能 (SPI1 clock enable)

由软件置 1 和清零。

0：禁止 SPI1 时钟

1：使能 SPI1 时钟

位 11-9 保留，必须保持复位值。

位 8 ADC1EN ： ADC1 时钟使能 (ADC1 clock enable)

由软件置 1 和清零。

0：禁止 ADC1 时钟

1：使能 ADC1 时钟

位 7:5 保留，必须保持复位值。

位 4 USART1EN ： USART1 时钟使能 (USART1 clock enable)

由软件置 1 和清零。

0：禁止 USART1 时钟

1：使能 USART1 时钟

位 3:2 保留，必须保持复位值。

位 1 PWM8EN ： PWM8 时钟使能 (PWM8 clock enable)

由软件置 1 和清零。

0：禁止 PWM8 时钟

1：使能 PWM8 时钟

位 0 PWM1EN ： PWM1 时钟使能 (PWM1 clock enable)

由软件置 1 和清零。

0：禁止 PWM1 时钟

1：使能 PWM1 时钟

8.3.14 RCC 时钟控制和状态寄存器 (RCC_CSR)

RCC clock control & status register

偏移地址：0x74

复位值：0x0E00 0000，除复位标志只能通过电源复位进行复位以外，其它通过系统复位进行复位。

访问：0 ≤等待周期 ≤3，按字、半字和字节访问

连续访问该寄存器时，则插入等待周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPWRRSTF	WWDGRSTF	IWDGRSTF	SFTRSTF	PORRSTF	PINRSTF	BORRSTF	RMVF	Reserved							
rw	rw	rw	rw	rw	rw	rw	rw								

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

位 31 LPWRRSTF ：低功耗复位标志 (Low-power reset flag)

发生低功耗管理复位时，由硬件置 1。

通过写入 RMVF 位清零。

0：未发生低功耗管理复位

1：发生低功耗管理复位

有关低功耗管理复位的详细信息，请参见 低功耗管理复位 。

位 30 WWDGRSTF ：窗口看门狗复位标志 (Window watchdog reset flag)

发生窗口看门狗复位时，由硬件置 1。

通过写入 RMVF 位清零。

0：未发生窗口看门狗复位

1：发生窗口看门狗复位

位 29 IWDGRSTF ：独立看门狗复位标志 (Independent watchdog reset flag)

发生来自 V DD 域的独立看门狗复位时，由硬件置 1。

通过写入 RMVF 位清零。

0：未发生看门狗复位

1：发生看门狗复位

位 28 SFTRSTF ：软件复位标志 (Software reset flag)

发生软件复位时，由硬件置 1。

通过写入 RMVF 位清零。

0：未发生软件复位

1：发生软件复位

位 27 PORRSTF ：POR/PDR 复位标志 (POR/PDR reset flag)

发生 POR/PDR 复位时，由硬件置 1。

通过写入 RMVF 位清零。

0：未发生 POR/PDR 复位

1：发生 POR/PDR 复位

位 26 PINRSTF ：引脚复位标志 (PIN reset flag)

发生来自 NRST 引脚的复位时，由硬件置 1。

通过写入 RMVF 位清零。

0：未发生来自 NRST 引脚的复位

1：发生来自 NRST 引脚的复位

STC MCU

位 25 BORRSTF ：BOR 复位标志 (BOR reset flag)

通过软件写入 RMVF 位清零。

发生 POR/PDR 复位或 BOR 复位时，由硬件置 1。

0：未发生 POR/PDR 复位或 BOR 复位

1：发生 POR/PDR 复位或 BOR 复位

位 24 RMVF ：清除复位标志 (Remove reset flag)

由软件置 1，用于将复位标志清零。

0：无操作

1：清零复位标志

位 23:2 保留，必须保持复位值。

位 1 LSIRDY ：内部低速振荡器就绪 (Internal low-speed oscillator ready)

由硬件置 1 和清零，用于指示内部 RC 40 kHz 振荡器已稳定。在 LSION 位被清零后，LSIRDY 将在 3 个 LSI 时钟周期后转为低电平。

0：LSI RC 振荡器未就绪

1：LSI RC 振荡器就绪

位 0 LSION ：内部低速振荡器使能 (Internal low-speed oscillator enable)

由软件置 1 和清零。

0：LSI RC 振荡器关闭

1：LSI RC 振荡器开启

8.3.15RCC 寄存器映射

表 26 给出了寄存器映射和复位值。

偏移地址	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	RCC_CR	Reserved				PLL I2SRDY	PLL I2SON	PLL RDY	PLL ON	Reserved				CSSON	HSEBYP	HSERDY	HSEON	HSICAL 7	HSICAL 6	HSICAL 5	HSICAL 4	HSICAL 3	HSICAL 2	HSICAL 1	HSICAL 0	HSITRIM 4	HSITRIM 3	HSITRIM 2	HSITRIM 1	HSITRIM 0	Reserved	HSIRDY	HSION
0x04	RCC_PLLCFGR	Reserved				PLLQ 3	PLLQ 2	PLLQ 1	PLLQ 0	Reserved	PLLSRC	Reserved				PLLP 1	PLLP 0	Reserved	PLLN 8	PLLN 7	PLLN 6	PLLN 5	PLLN 4	PLLN 3	PLLN 2	PLLN 1	PLLN 0	PLLM 5	PLLM 4	PLLM 3	PLLM 2	PLLM 1	PLLM 0
0x08	RCC_CFGR	MCO2 1	MCO2 0	MCO2PRE2	MCO2PRE1	MCO2PRE0	MCO1PRE2	MCO1PRE1	MCO1PRE0	I2SSRC	MCO1 1	MCO1 0	RTCPRE 4	RTCPRE 3	RTCPRE 2	RTCPRE 1	RTCPRE 0	PPRE2 2	PPRE2 1	PPRE2 0	PPRE1 2	PPRE1 1	PPRE1 0	Reserved		HPRE 3	HPRE 2	HPRE 1	HPRE 0	SWS 1	SWS 0	SW 1	SW 0	
0x0C	RCC_CIR	Reserved								CSSC	Reserved	PLLI2SRDYC	PLLRDYC	HSERDYC	HSIRDYC	LSERDYC	LSIRDYC	Reserved	PLLI2SRDYIE	PLLRDYIE	HSERDYIE	HSIRDYIE	LSERDYIE	LSIRDYIE	CSSF	Reserved	PLLI2SRDYF	PLLRDYF	HSERDYF	HSIRDYF	LSERDYF	LSIRDYF	

STC MCU

0x10	RCC_AHB1RSTR	Reserved, DMA1RST, Reserved, CRCRST, Reserved, GPIODRST, GPIOCRST, GPIOBRST, GPIOARST
0x14	RCC_AHB2RSTR	Reserved
0x18	RCC_AHB3RSTR	Reserved, FSMCRST
0x1C	Reserved	Reserved

0x20	RCC_APB1RSTR	Reserved, PWRRST, Reserved, I2C2RST, I2C1RST, Reserved, UART4RST, UART3RST, UART2RST, Reserved, SPI2RST, Reserved, WWDGRST, Reserved, PWM14RST, PWM13RST, PWM12RST, PWM7RST, PWM6RST, PWM5RST, PWM4RST, PWM3RST, PWM2RST
0x24	RCC_APB2RSTR	Reserved, PWM11RST, PWM10RST, PWM9RST, Reserved, SYSCFGRST, Reserved, SPI1RST, Reserved, ADCRST, Reserved, USART1RST, Reserved, PWM8RST, PWM1RST
0x28	Reserved	Reserved
0x2C	Reserved	Reserved
0x30	RCC_AHB1ENR	Reserved, DMA2EN, DMA1EN, Reserved, CRCEN, Reserved, GPIODEN, GPIOCEN, GPIOBEN, GPIOAEN
0x34	RCC_AHB2ENR	Reserved
0x38	RCC_AHB3ENR	Reserved, FSMCEN
0x3C	Reserved	Reserved

9 通用 I/O (GPIO)

除非特别说明，否则本部分适用于整个 STC33F系列。

9.1 GPIO 简介

每个通用 I/O 端口包括 4 个 32 位配置寄存器（GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR 和 GPIOx_PUPDR）、2 个 32 位数据寄存器（GPIOx_IDR 和 GPIOx_ODR）、1 个 32 位置位/ 复位寄存器 (GPIOx_BSRR)、1 个 32 位锁定寄存器(GPIOx_LCKR) 和 2 个 32 位复用功能选择寄存器（GPIOx_AFRH 和 GPIOx_AFRL）。

9.2 GPIO 主要特性

- 输出状态：推挽或开漏 + 上拉/ 下拉
- 从输出数据寄存器 (GPIOx_ODR) 或外设（复用功能输出）输出数据
- 可为每个 I/O 选择不同的速度
- 输入状态：浮空、上拉/ 下拉、模拟
- 将数据输入到输入数据寄存器 (GPIOx_IDR) 或外设（复用功能输入）
- 置位和复位寄存器 (GPIOx_BSRR)，对 GPIOx_ODR 具有按位写权限
- 锁定机制 (GPIOx_LCKR)，可冻结 I/O 配置
- 模拟功能
- 复用功能输入/ 输出选择寄存器（一个 I/O 最多可具有 16 个复用功能）
- 快速翻转，每次翻转最快只需要两个时钟周期
- 引脚复用非常灵活，允许将 I/O 引脚用作 GPIO 或多种外设功能中的一种

9.3 GPIO 功能描述

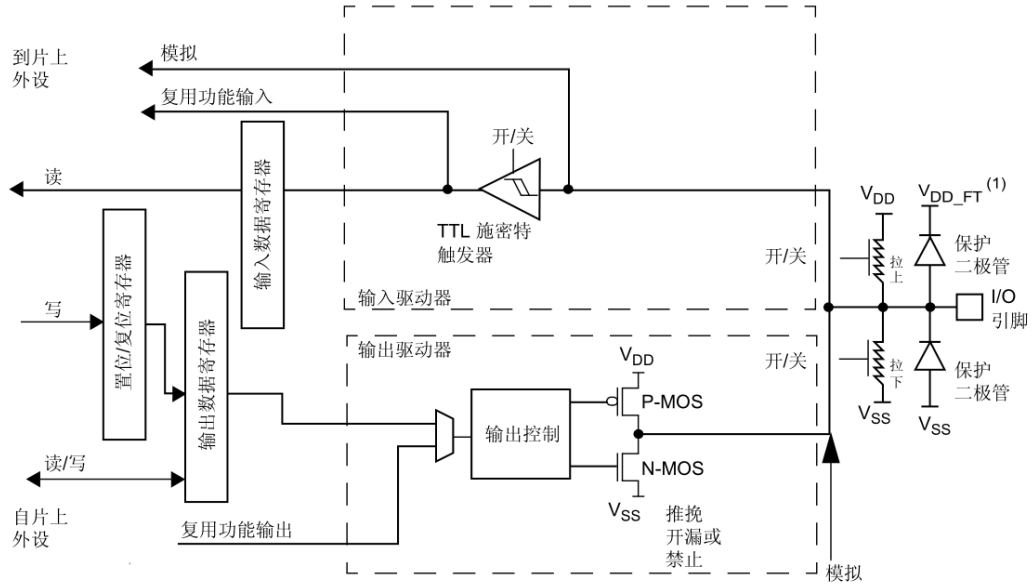
根据数据手册中列出的每个 I/O 端口的特性，可通过软件将通用 I/O (GPIO) 端口的各个端口位分别配置为多种模式：

- 输入浮空
- 输入上拉
- 输入下拉
- 模拟功能
- 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

每个 I/O 端口位均可自由编程，但 I/O 端口寄存器必须按 32 位字、半字或字节进行访问。GPIOx_BSRR 寄存器旨在实现对 GPIO ODR 寄存器进行原子读取/修改访问。这样便可确保在读取和修改访问之间发生中断请求也不会有问题。

图 17 显示了 5 V 容忍 I/O 端口位的基本结构。表 28 给出了可能的端口位配置方案。

图 17. 5 V 容忍 I/O 端口位的基本结构



1. V_{DD_FT} 是和 5 V 容忍 I/O 相关的电位，与 V_{DD} 不同。

表 28.端口位配置表(1)

MODER(i) [1:0]	OTYPER(i)	OSPEEDR(i) [B:A]	PUPDR(i) [1:0]		I/O 配置	
01	0	SPEED [B:A]	0	0	GP 输出	PP
	0		0	1	GP 输出	PP + PU
	0		1	0	GP 输出	PP + PD
	0		1	1	保留	
	1		0	0	GP 输出	OD
	1		0	1	GP 输出	OD + PU
	1		1	0	GP 输出	OD + PD
	1		1	1	保留（GP 输出 OD）	
10	0	SPEED [B:A]	0	0	AF	PP
	0		0	1	AF	PP + PU
	0		1	0	AF	PP + PD
	0		1	1	保留	
	1		0	0	AF	OD
	1		0	1	AF	OD + PU
	1		1	0	AF	OD + PD
	1		1	1	保留	

表 28.端口位配置表(1)（续）

MODER(i) [1:0]	OTYPER(i)	OSPEEDR(i) [B:A]		PUPDR(i) [1:0]		I/O 配置	
00	x	x	x	0	0	输入	浮空
	x	x	x	0	1	输入	PU
	x	x	x	1	0	输入	PD
	x	x	x	1	1	保留（输入浮空）	

11	x	x	x	0	0	输入/输出	模拟
	x	x	x	0	1	保留	
	x	x	x	1	0		
	x	x	x	1	1		

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

9.3.1 通用 I/O (GPIO)

在复位期间及复位刚刚完成后，复用功能尚未激活，I/O 端口被配置为输入浮空模式。

复位后，调试引脚处于复用功能上拉/ 下拉状态：
- P03：JTDI 处于上拉状态
- P1F：JTCK/SWCLK 处于下拉状态
- P1E：JTMS/SWDAT 处于下拉状态
- P02：NJTRST 处于上拉状态
- P04：JTDO 处于浮空状态

当引脚配置为输出后，写入到输出数据寄存器 (GPIOx_ODR) 的值将在 I/O 引脚上输出。可以在推挽模式下或开漏模式下使用输出驱动器（输出 0 时仅激活 N-MOS）。

输入数据寄存器 (GPIOx_IDR) 每隔 1 个 AHB1 时钟周期捕获一次 I/O 引脚的数据。

所有 GPIO 引脚都具有内部弱上拉及下拉电阻，可根据 GPIOx_PUPDR 寄存器中的值来打开/ 关闭。

9.3.2 I/O 引脚复用器和映射

微控制器 I/O 引脚通过一个复用器连接到板载外设/ 模块，该复用器一次仅允许一个外设的复用功能 (AF) 连接到 I/O 引脚。这可以确保共用同一个 I/O 引脚的外设之间不会发生冲突。

每个 I/O 引脚都有一个复用器，该复用器采用 16 路复用功能输入（AF0 到 AF15），可通过 GPIOx_AFRL（针对引脚 0 到 7）和 GPIOx_AFRH（针对引脚 8 到 15）寄存器对这些输入进行配置：

- 完成复位后，所有 I/O 都会连接到系统的复用功能 0 (AF0)。
- 外设的复用功能映射到 AF1 至 AF13。

下面的图 18 对此结构进行了说明。

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

要将 I/O 配制成所需功能，请按照以下步骤操作：

1. 系统功能

将 I/O 连接到 AF0，然后根据所用功能进行配置：
—JTAG/SWD：在各器件复位后，会将这些引脚指定为专用引脚，可供片上调试模块立即使用（不受 GPIO 控制器控制）。
—RTC_REFIN：此引脚应配置为输入浮空模式。
—MCO1 和 MCO2：这些引脚必须配置为复用功能模式。
注意：可禁止部分或全部 JTAG/SWD 引脚，以释放相关联的引脚供 GPIO 使用。

表 29.灵活的 SWJ-DP 引脚分配

可用的调试端口	用到的 SWJ I/O 引脚				
	P1E/ JTMS/ SWDIO	P1F/ JTCK/ SWCLK	P03/ JTDI	P04/ JTDO	P02/ NJTRST
全部 SWJ (JTAG-DP + SW-DP) - 复位状态	X	X	X	X	X
全部 SWJ (JTAG-DP + SW-DP)，但不包括 NJTRST	X	X	X	X	
禁止 JTAG-DP 和使能 SW-DP	X	X			
禁止 JTAG-DP 和禁止 SW-DP	已释放				

2. GPIO

在 GPIOx_MODER 寄存器中将所需 I/O 配置为输出或输入。

3. 外设复用功能

对于 ADC 和 DAC，在 GPIOx_MODER 寄存器中将所需 I/O 配置为模拟通道。对于其它外设：

—在 GPIOx_MODER 寄存器中将所需 I/O 配置为复用功能

—通过 GPIOx_OTYPER、GPIOx_PUPDR 和 GPIOx_OSPEEDER 寄存器，分别选择类型、上拉/ 下拉以及输出速度

—在 GPIOx_AFRL 或 GPIOx_AFRH 寄存器中，将 I/O 连接到所需 AFx

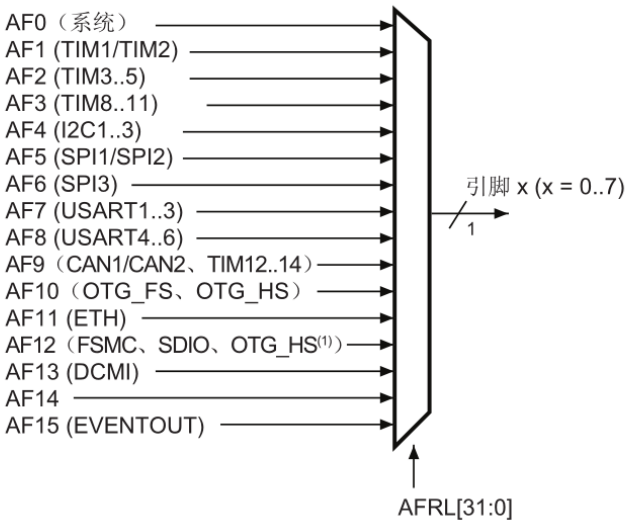
4. EVENTOUT

配置用于输出 EVENTOUT 信号的 I/O 引脚（通过将其连接到 AF15）

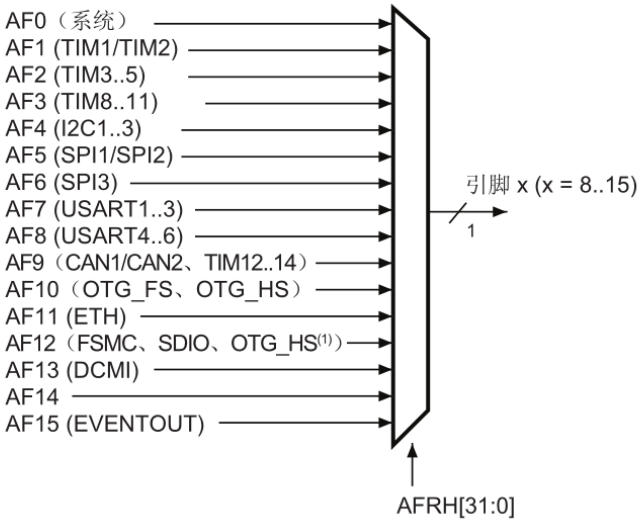
图 18.复用功能选择

STC MCU

对于引脚 0 到引脚 7，GPIOx_AFRL[31:0] 寄存器会选择专用的复用功能



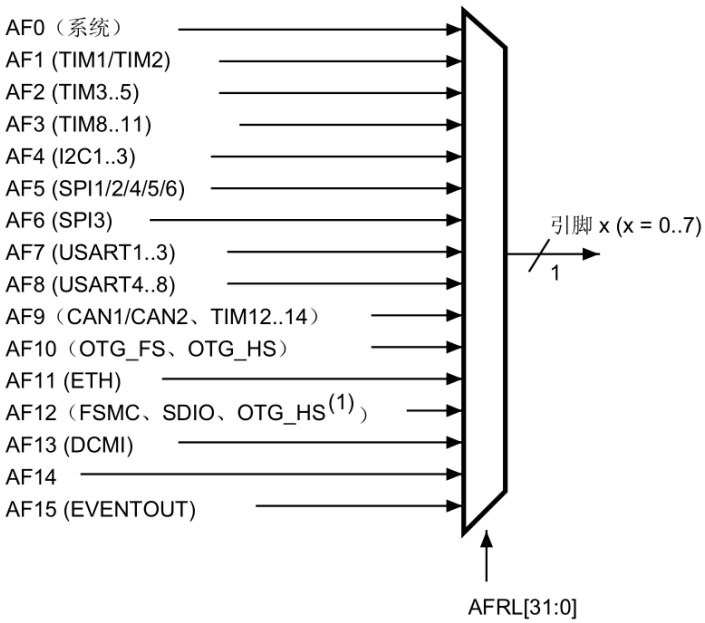
对于引脚 8 到引脚 15，GPIOx_AFRH[31:0] 寄存器会选择专用的复用功能



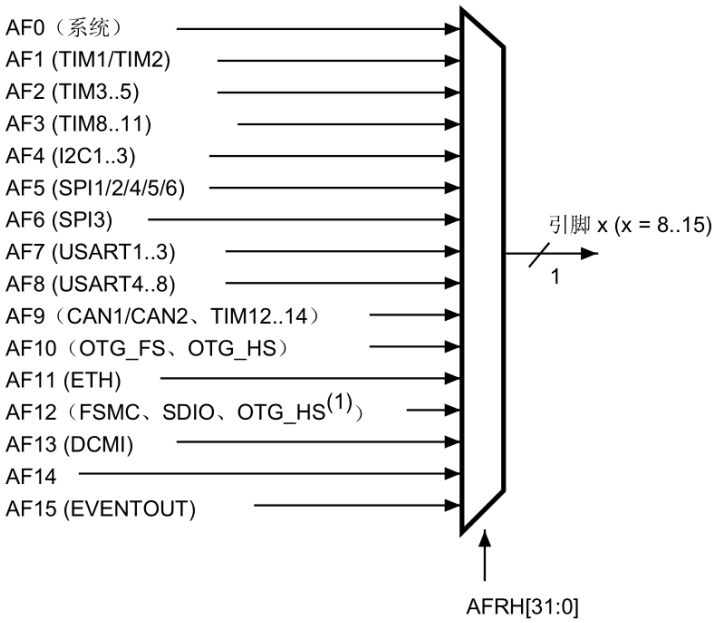
1. 在 FS 模式下配置。

图 19. 复用功能选择

对于引脚 0 到引脚 7GPIOx_AFRL[31:0] 寄存器会选择专用的复用功能



对于引脚 8 到引脚 15，GPIOx_AFRH[31:0] 寄存器会选择专用的复用功能



1. 在 FS 模式下配置。

9.3.3 I/O 端口控制寄存器

每个 GPIO 有 4 个 32 位存储器映射的控制寄存器（GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR），可配置多达 16 个 I/O。GPIOx_MODER 寄存器用于选择 I/O 方向（输入、输出、AF、模拟）。GPIOx_OTYPER 和 GPIOx_OSPEEDR 寄存器分别用于选择输出类型（推挽或开漏）和速度（无论采用哪种 I/O 方向，都会直接将 I/O 速度引脚连接到相应的 GPIOx_OSPEEDR 寄存器位）。无论采用哪种 I/O 方向，GPIOx_PUPDR 寄存器都用于选择上拉/ 下拉。

9.3.4 I/O 端口数据寄存器

每个 GPIO 都具有 2 个 16 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，可对其进行读/ 写访问。通过 I/O 输入的数据存储到输入数据寄存器 (GPIOx_IDR) 中，它是一个只读寄存器。

9.3.5 I/O 数据位操作

置位复位寄存器 (GPIOx_BSRR) 是一个 32 位寄存器，它允许应用程序在输出数据寄存器 (GPIOx_ODR) 中对各个单独的数据位执行置位和复位操作。置位复位寄存器的大小是 GPIOx_ODR 的二倍。

GPIOx_ODR 中的每个数据位对应于 GPIOx_BSRR 中的两个控制位：BSRR(i) 和 BSRR(i+SIZE)。当写入 1 时，BSRR(i) 位会置位对应的 ODR(i) 位。当写入 1 时，BSRR(i+SIZE) 位会清零 ODR(i) 对应的位。

在 GPIOx_BSRR 中向任何位写入 0 都不会对 GPIOx_ODR 中的对应位产生任何影响。如果在 GPIOx_BSRR 中同时尝试对某个位执行置位和清零操作，则置位操作优先。

使用 GPIOx_BSRR 寄存器更改 GPIOx_ODR 中各个位的值是一个“单次”操作，不会锁定 GPIOx_ODR 位。随时都可以直接访问 GPIOx_ODR 位。GPIOx_BSRR 寄存器提供了一种执行原子按位处理的方法。

在对 GPIOx_ODR 进行位操作时，软件无需禁止中断：在一次原子 AHB1 写访问中，可以修改一个或多个位。

9.3.6 GPIO 锁定机制

通过将特定的写序列应用到 GPIOx_LCKR 寄存器，可以冻结 GPIO 控制寄存器。冻结的寄存器包括 GPIOx_MODER 、 GPIOx_OTYPER 、 GPIOx_OSPEEDR 、 GPIOx_PUPDR 、 GPIOx_AFRL 和 GPIOx_AFRH。

要对 GPIOx_LCKR 寄存器执行写操作，必须应用特定的写/读序列。当正确的 LOCK 序列应用到此寄存器的第 16 位后，会使用 LCKR[15:0] 的值来锁定 I/O 的配置（在写序列期间，LCKR[15:0] 的值必须相同）。将 LOCK 序列应用到某个端口位后，在执行下一次复位之前，将无法对该端口位的值进行修改。每个 GPIOx_LCKR 位都会冻结控制寄存器（GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR、GPIOx_AFRL 和 GPIOx_AFRH）中的对应位。

LOCK 序列（参见第 9.4.8 节：GPIO 端口配置锁定寄存器(GPIOx_LCKR) (x = A..I)）只能通过对 GPIOx_LCKR 寄存器进行字（32 位长）访问的方式来执行，因为 GPIOx_LCKR 的第 16 位必须与 [15:0] 位同时置位。

9.3.7 I/O 复用功能输入/ 输出

有两个寄存器可用来从每个 I/O 可用的 16 个复用功能输入/ 输出中进行选择。借助这些寄存器，可根据应用程序的要求将某个复用功能连接到其它某个引脚。
这意味着可使用 GPIOx_AFRL 和 GPIOx_AFRH 复用功能寄存器在每个 GPIO 上复用多个可用的外设功能。这样一来，应用程序可为每个 I/O 选择任何一个可用功能。由于 AF 选择信号由复用功能输入和复用功能输出共用，所以只需为每个 I/O 的复用功能输入/ 输出选择一个通道即可。

注意：对于每个 I/O 而言，应用程序一次只能为其选择一个可用的外设功能。

9.3.8 外部中断线/ 唤醒线

所有端口都具有外部中断功能。要使用外部中断线，必须将端口配置为输入模式

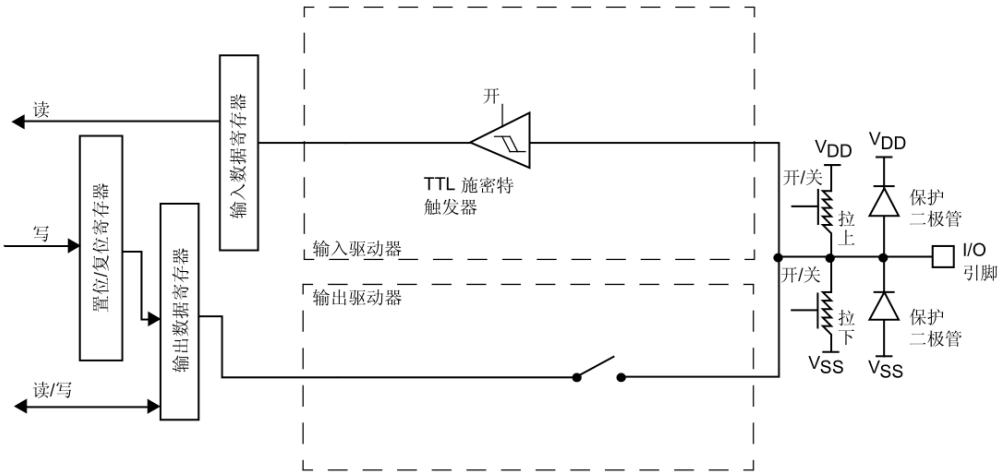
9.3.9 输入配置

对 I/O 端口进行编程作为输入时： 输出缓冲器被关闭

- 施密特触发器输入被打开
- 根据 GPIOx_PUPDR 寄存器中的值决定是否打开上拉和下拉电阻
- 输入数据寄存器每隔 1 个 AHB1 时钟周期对 I/O 引脚上的数据进行一次采样
- 对输入数据寄存器的读访问可获取 I/O 状态

图 20 说明了 I/O 端口位的输入配置。

图 20.输入浮空/ 上拉/ 下拉配置



9.3.10输出配置

对 I/O 端口进行编程作为输出时：

- 输出缓冲器被打开：

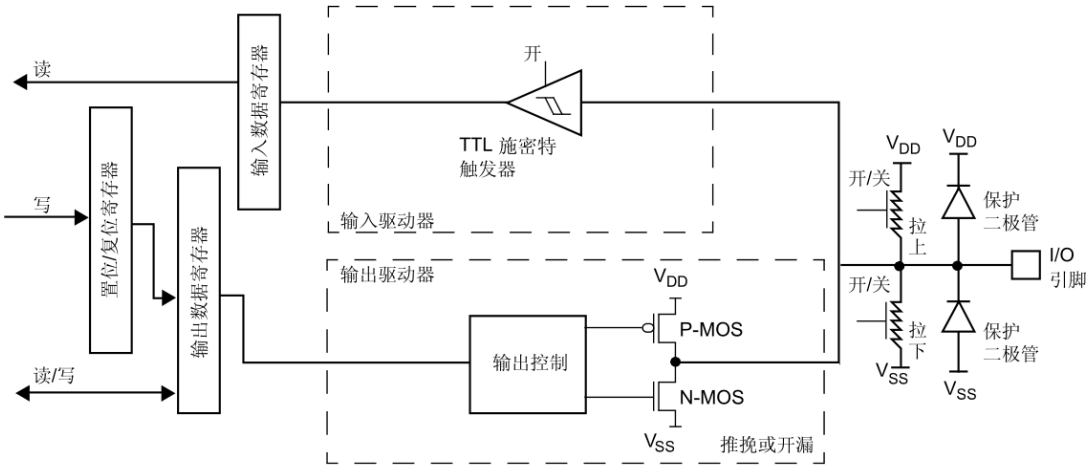
—开漏模式：输出寄存器中的“0”可激活 N-MOS，而输出寄存器中的“1”会使端口保持高组态 (Hi-Z)（P-MOS 始终不激活）。

—推挽模式：输出寄存器中的“0”可激活 N-MOS，而输出寄存器中的“1”可激活 P-MOS。

- 施密特触发器输入被打开
- 根据 GPIOx_PUPDR 寄存器中的值决定是否打开弱上拉电阻和下拉电阻
- 输入数据寄存器每隔 1 个 AHB1 时钟周期对 I/O 引脚上的数据进行一次采样
- 对输入数据寄存器的读访问可获取 I/O 状态
- 对输出数据寄存器的读访问可获取最后的写入值

图 21 说明了 I/O 端口位的输出配置。

图 21. 输出配置



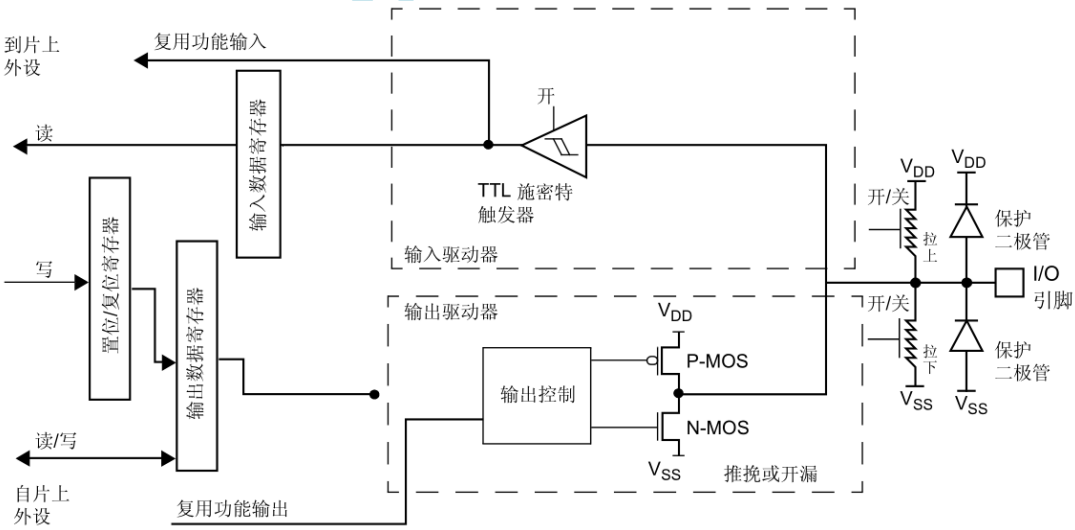
9.3.11复用功能配置

对 I/O 端口进行编程作为复用功能时：

- 可将输出缓冲器配置为开漏或推挽
- 输出缓冲器由来自外设的信号驱动（发送器使能和数据）
- 施密特触发器输入被打开
- 根据 GPIOx_PUPDR 寄存器中的值决定是否打开弱上拉电阻和下拉电阻
- 输入数据寄存器每隔 1 个 AHB1 时钟周期对 I/O 引脚上的数据进行一次采样
- 对输入数据寄存器的读访问可获取 I/O 状态

图 22 说明了 I/O 端口位的复用功能配置。

图 22.复用功能配置



9.3.12模拟配置

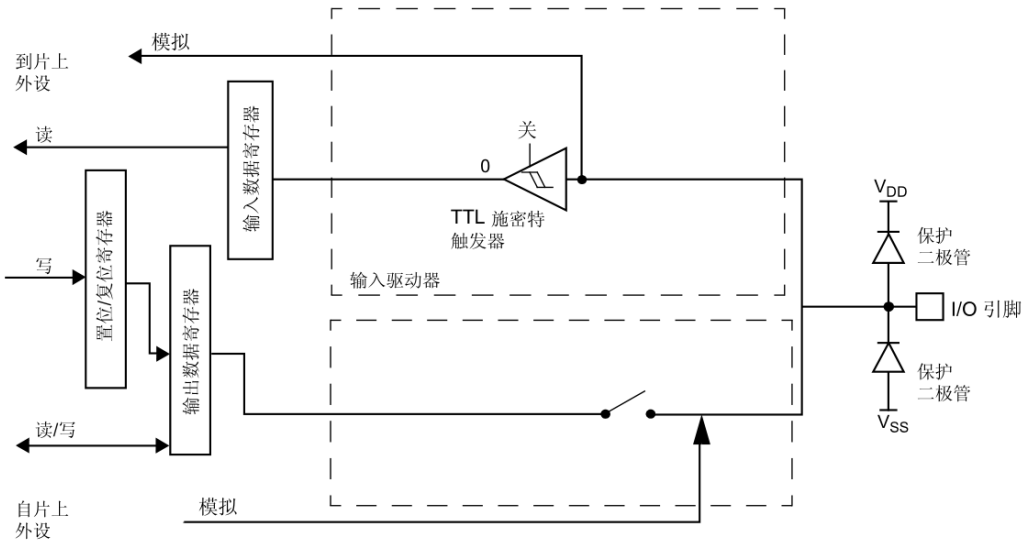
对 I/O 端口进行编程作为模拟配置时：

- 输出缓冲器被禁止。
- 施密特触发器输入停用，I/O 引脚的每个模拟输入的功耗变为零。施密特触发器的输出被强制处理为恒定值 (0)。
- 弱上拉和下拉电阻被关闭。

- 对输入数据寄存器的读访问值为“0”。

图 23 说明了 I/O 端口位的高阻态模拟输入配置。

图 23. 高组态模拟配置



9.3.13将 XOSCI/ XOSCO 引脚用作 GPIO P0F/P0E 端口引脚

当 LSE 振荡器处于关闭状态时，可分别将 LSE 振荡器引脚 OSC32_IN 和 OSC32_OUT 用作通用 P0F I/O 和 P0E I/O。当 LSE 振荡器处于开启状态时，PC14 I/O 和 P0E I/O 只能被配置为 LSE 振荡器引脚 OSC32_IN 和 OSC32_OUT。可通过在 RCC_BDCR 寄存器中将 LSEON 位置 1 来完成此操作。LSE 的优先级高于 GPIO 功能。

9.3.14选择复用功能

9.4GPIO 寄存器

本节对 GPIO 寄存器进行了详细介绍。
有关寄存器位、寄存器偏移地址和复位值的汇总，请参见表 32。

可通过字节（8 位）、半字（16 位）或字（32 位）对 GPIO 寄存器进行访问。

9.4.1 GPIO 端口模式寄存器 (GPIOx_MODER) (x = A..I)

GPIO port mode register
偏移地址：0x00
复位值：

- 0xA800 0000（端口 A）
- 0x0000 0280（端口 B）
- 0x0000 0000（其它端口）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 2y:2y+1 MODERy[1:0]：端口 x 配置位 (Port x configuration bits) (y = 0..15)
这些位通过软件写入，用于配置 I/O 方向模式。
00：输入（复位状态）
01：通用输出模式
10：复用功能模式
11：模拟模式

9.4.2 GPIO 端口输出类型寄存器 (GPIOx_OTYPER) (x = A..I)

GPIO port output type register
偏移地址：0x04
复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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 31:16 保留，必须保持复位值。
位 15:0 OTy[1:0]：端口 x 配置位 (Port x configuration bits) (y = 0..15)
这些位通过软件写入，用于配置 I/O 端口的输出类型。
0：输出推挽（复位状态）
1：输出开漏

9.4.3 GPIO 端口输出速度寄存器 (GPIOx_OSPEEDR) (x = A..I/)

GPIO port output speed register
偏移地址：0x08
复位值：
- 0x0000 00C0（端口 B）
- 0x0000 0000（其它端口）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSPEEDR15[1:0]		OSPEEDR14[1:0]		OSPEEDR13[1:0]		OSPEEDR12[1:0]		OSPEEDR11[1:0]		OSPEEDR10[1:0]		OSPEEDR9[1:0]		OSPEEDR8[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
OSPEEDR7[1:0]		OSPEEDR6[1:0]		OSPEEDR5[1:0]		OSPEEDR4[1:0]		OSPEEDR3[1:0]		OSPEEDR2[1:0]		OSPEEDR1[1:0]		OSPEEDR0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 2y:2y+1 OSPEEDRy[1:0]：端口 x 配置位 (Port x configuration bits) (y = 0..15)
这些位通过软件写入，用于配置 I/O 输出速度。
11：12 MHz（小电流慢速）
01：16MHz（小电流快速）
10：22MHz（大电流慢速）
00：25 MHz（大电流快速）

9.4.4 GPIO 端口上拉/ 下拉寄存器 (GPIOx_PUPDR) (x = A..I/)

GPIO port pull-up/pull-down register
偏移地址：0x0C
复位值：
- 0x6400 0000（端口 A）
- 0x0000 0100（端口 B）
- 0x0000 0000（其它端口）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[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
PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 2y:2y+1 PUPDRy[1:0]：端口 x 配置位 (Port x configuration bits) (y = 0..15)
这些位通过软件写入，用于配置 I/O 上拉或下拉。
00：无上拉或下拉
01：上拉
10：下拉
11：保留

9.4.5 GPIO 端口输入数据寄存器 (GPIOx_IDR) (x = A..I)

GPIO port input data register
偏移地址：0x10
复位值：0x0000 XXXX（其中 X 表示未定义）

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。
位 15:0 IDRy[15:0]：端口输入数据 (Port input data) (y = 0..15)
这些位为只读形式，只能在字模式下访问。它们包含相应 I/O 端口的输入值。

9.4.6 GPIO 端口输出数据寄存器 (GPIOx_ODR) (x = A..I)

GPIO port output data register
偏移地址：0x14
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。
位 15:0 ODRy[15:0]：端口输出数据 (Port output data) (y = 0..15)
这些位可通过软件读取和写入。
注意：对于原子置位/复位，通过写入 GPIOx_BSRR 寄存器，可分别对 ODR 位进行置位和复位(x = A..I/)。

9.4.7 GPIO 端口置位/ 复位寄存器 (GPIOx_BSRR) (x = A..I)

GPIO port bit set/reset register
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 31:16 BRy：端口 x 复位位 y (Port x reset bit y) (y = 0..15)
这些位为只写形式，只能在字、半字或字节模式下访问。读取这些位可返回值 0x0000。
0：不会对相应的 ODRx 位执行任何操作 1：对相应的 ODRx 位进行复位
注意：如果同时对 BSx 和 BRx 置位，则 BSx 的优先级更高。

位 15:0 BSy：端口 x 置位位 y (Port x set bit y) (y= 0..15)
这些位为只写形式，只能在字、半字或字节模式下访问。读取这些位可返回值 0x0000。
0：不会对相应的 ODRx 位执行任何操作 1：对相应的 ODRx 位进行置位

9.4.8 GPIO 端口配置锁定寄存器 (GPIOx_LCKR) (x = A..I)

GPIO port configuration lock register

当正确的写序列应用到第 16 位 (LCKK) 时，此寄存器将用于锁定端口位的配置。位 [15:0] 的值用于锁定 GPIO 的配置。在写序列期间，不能更改 LCKR[15:0] 的值。将 LOCK 序列应用到某个端口位后，在执行下一次复位之前，将无法对该端口位的值进行修改。

注意：可使用特定的写序列对 GPIOx_LCKR 寄存器执行写操作。在此写序列期间只允许使用字访问（32 位长）。每个锁定位冻结一个特定的配置寄存器（控制寄存器和复用功能寄存器）。

偏移地址：0x1C
复位值：0x0000 0000

访问：仅 32 位字，读/ 写寄存器

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

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

位 31:17 保留，必须保持复位值。

位 16 LCKK[16]：锁定键 (Lock key)
可随时读取此位。可使用锁定键写序列对其进行修改。
0：端口配置锁定键未激活。
1：端口配置锁定键已激活。直到 MCU 复位时，才锁定 GPIOx_LCKR 寄存器。

锁定键写序列：
WR LCKR[16] = ‘1’ + LCKR[15:0] WR LCKR[16] = ‘0’ + LCKR[15:0] WR LCKR[16] = ‘1’ + LCKR[15:0] RD LCKR
RD LCKR[16] = ‘1’（此读操作为可选操作，但它可确认锁定已激活）
注意：在锁定键写序列期间，不能更改 LCK[15:0] 的值。
锁定序列中的任何错误都将中止锁定操作。
在任一端口位上的第一个锁定序列之后，对 LCKK 位的任何读访问都将返回“1”，直到下一次 CPU 复位为止。

位 15:0 LCKy：端口 x 锁定位 y (Port x lock bit y) (y= 0..15)
这些位都是读/ 写位，但只能在 LCKK 位等于“0”时执行写操作。
0：端口配置未锁定
1：端口配置已锁定

9.4.9 GPIO 复用功能低位寄存器 (GPIOx_AFRL) (x = A..I)

GPIO alternate function low register
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
AFRL7[3:0]				AFRL6[3:0]				AFRL5[3:0]				AFRL4[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

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

位 31:0 AFRLy：端口 x 位 y 的复用功能选择 (Alternate function selection for port x bit y) (y = 0..7)
这些位通过软件写入，用于配置复用功能 I/O。
AFRLy 选择：

0000：AF0	1000：AF8
0001：AF1	1001：AF9
0010：AF2	1010：AF10
0011：AF3	1011：AF11
0100：AF4	1100：AF12

0101：AF5	1101：AF13
0110：AF6	1110：AF14
0111：AF7	1111：AF15

9.4.10 GPIO 复用功能高位寄存器 (GPIOx_AFRH) (x = A..I)

GPIO alternate function high register
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
AFRH15[3:0]				AFRH14[3:0]				AFRH13[3:0]				AFRH12[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
AFRH11[3:0]				AFRH10[3:0]				AFRH9[3:0]				AFRH8[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 AFRHy：端口 x 位 y 的复用功能选择 (Alternate function selection for port x bit y) (y = 8.0.15)
这些位通过软件写入，用于配置复用功能 I/O。

AFRLy 选择：

0000：AF0	1000：AF8
0001：AF1	1001：AF9
0010：AF2	1010：AF10
0011：AF3	1011：AF11
0100：AF4	1100：AF12
0101：AF5	1101：AF13
0110：AF6	1110：AF14
0111：AF7	1111：AF15

9.4.11 GPIO 寄存器映射

下表列出了 GPIO 寄存器映射和复位值。

表 32.GPIO 寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	GPIOA_MODER	MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]		MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
	Reset value	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00	GPIOB_MODER	MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]		MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
0x00	GPIOx_MODER (where x = C..I/)	MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]		MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

STC MCU

表 32. GPIO 寄存器映射和复位值（续）

偏移	寄存器	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
0x04	GPIOx_OTYPER (where x = A..I/)	Reserved																OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	GPIOx_OSPEEDER (where x = A..**I**/ except B)	OSPEEDR15[1:0]		OSPEEDR14[1:0]		OSPEEDR13[1:0]		OSPEEDR12[1:0]		OSPEEDR11[1:0]		OSPEEDR10[1:0]		OSPEEDR9[1:0]		OSPEEDR8[1:0]		OSPEEDR7[1:0]		OSPEEDR6[1:0]		OSPEEDR5[1:0]		OSPEEDR4[1:0]		OSPEEDR3[1:0]		OSPEEDR2[1:0]		OSPEEDR1[1:0]		OSPEEDR0[1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	GPIOB_OSPEEDER	OSPEEDR15[1:0]		OSPEEDR14[1:0]		OSPEEDR13[1:0]		OSPEEDR12[1:0]		OSPEEDR11[1:0]		OSPEEDR10[1:0]		OSPEEDR9[1:0]		OSPEEDR8[1:0]		OSPEEDR7[1:0]		OSPEEDR6[1:0]		OSPEEDR5[1:0]		OSPEEDR4[1:0]		OSPEEDR3[1:0]		OSPEEDR2[1:0]		OSPEEDR1[1:0]		OSPEEDR0[1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
0x0C	GPIOA_PUPDR	PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]		PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
	Reset value	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	GPIOB_PUPDR	PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]		PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0x0C	GPIOx_PUPDR (where x = C..**I**/)	PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]		PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	GPIOx_IDR (where x = A..**I**/)	Reserved																IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
	Reset value																	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x14	GPIOx_ODR (where x = A..**I**/)	Reserved																ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	GPIOx_BSRR (where x = A..**I**/)	BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0	BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	GPIOx_LCKR (where x = A..**I**/)	Reserved															LCKK	LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	GPIOx_AFRL (where x = A..**I**/)	AFRL7[3:0]				AFRL6[3:0]				AFRL5[3:0]				AFRL4[3:0]				AFRL3[3:0]				AFRL2[3:0]				AFRL1[3:0]				AFRL0[3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	GPIOx_AFRH (where x = A..**I**/)	AFRH15[3:0]				AFRH14[3:0]				AFRH13[3:0]				AFRH12[3:0]				AFRH11[3:0]				AFRH10[3:0]				AFRH9[3:0]				AFRH8[3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10 系统配置控制器 (SYSCFG)

系统配置控制器主要用于管理对可执行代码的存储区域的地址重映射、管理 GPIO 的外部中断线连接。

10.1.1 SYSCFG 外部中断配置寄存器 1 (SYSCFG_EXTICR1)

SYSCFG external interrupt configuration register 1
偏移地址：0x08
复位值：0x0000

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

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

位 31:16 保留，必须保持复位值。
位 15:0 EXTIx[3:0]：EXTI x 配置（x = 0 到 3）(EXTI x configuration (x = 0 to 3))这些位通过软件写入，以选择 EXTIx 外部中断的源输入。
0000：P0[x] 引脚
0001：P1[x] 引脚
0010：P2[x] 引脚
0011：P3[x] 引脚

10.1.2 SYSCFG 外部中断配置寄存器 2 (SYSCFG_EXTICR2)

SYSCFG external interrupt configuration register 2
偏移地址：0x0C
复位值：0x0000

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

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

位 31:16 保留，必须保持复位值。
位 15:0 EXTIx[3:0]：EXTI x 配置（x = 4 到 7）(EXTI x configuration (x = 4 to 7))这些位通过软件写入，以选择 EXTIx 外部中断的源输入。
0000：P0[x] 引脚
0001：P1[x] 引脚
0010：P2[x] 引脚
0011：P3[x] 引脚

10.1.3 SYSCFG 外部中断配置寄存器 3 (SYSCFG_EXTICR3)

SYSCFG external interrupt configuration register 3

偏移地址：0x10
复位值：0x0000

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

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

位 31:16 保留，必须保持复位值。
位 15:0 EXTIx[3:0]：EXTI x 配置（x = 8 到 11）(EXTI x configuration (x = 8 to 11))这些位通过软件写入，以选择 EXTIx 外部中断的源输入。
0000：P0[x] 引脚
0001：P1[x] 引脚
0010：P2[x] 引脚
0011：P3[x] 引脚

10.1.4SYSCFG 外部中断配置寄存器 4 (SYSCFG_EXTICR4)

SYSCFG external interrupt configuration register 4
偏移地址：0x14
复位值：0x0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI15[3:0]				EXTI14[3:0]				EXTI13[3:0]				EXTI12[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。
位 15:0 EXTIx[3:0]：EXTI x 配置（x = 12 到 15）(EXTI x configuration (x = 12 to 15))这些位通过软件写入，以选择 EXTIx 外部中断的源输入。
0000：P0[x] 引脚
0001：P1[x] 引脚
0010：P2[x] 引脚
0011：P3[x] 引脚

10.1.5SYSCFG 寄存器映射

下表列出了 SYSCFG 寄存器映射和复位值。

表 33. SYSCFG 寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	34	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x08	SYSCFG_EXTICR1	Reserved																EXTI3[3:0]				EXTI2[3:0]				EXTI1[3:0]				EXTI0[3:0]			
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

STC MCU

0x0C	SYSCFG_EXTICR2	Reserved	EXTI7[3:0]				EXTI6[3:0]				EXTI5[3:0]				EXTI4[3:0]			
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	SYSCFG_EXTICR3	Reserved	EXTI11[3:0]				EXTI10[3:0]				EXTI9[3:0]				EXTI8[3:0]			
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	SYSCFG_EXTICR4	Reserved	EXTI15[3:0]				EXTI14[3:0]				EXTI13[3:0]				EXTI12[3:0]			
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

STC MCU

11 中断和事件

除非特别说明，否则本部分适用于整个 STC33F 系列。

11.1 嵌套向量中断控制器 (NVIC)

11.1.1 NVIC 特性

嵌套向量中断控制器 NVIC 包含以下特性：

- STC33F 系列具有 82 个可屏蔽中断通道，
- 16 个可编程优先级（使用了 4 位中断优先级）
- 低延迟异常和中断处理
- 电源管理控制
- 系统控制寄存器的实现

嵌套向量中断控制器 (NVIC) 和处理器内核接口紧密配合，可以实现低延迟的中断处理和晚到中断的高效处理。

包括内核异常在内的所有中断均通过 NVIC 进行管理。

11.1.2 SysTick 校准值寄存器

SysTick 校准值设置为 18750。当 SysTick 时钟设置为 18.75 MHz，会产生 1 ms 时间基准。

11.1.3 中断和异常向量

请参见 表 45 器件的向量表。

STC MCU

表 45. STC33F 器件向量表

位置	优先级	优先级类型	名称	说明	地址
	-	-	-	保留	0x0000 0000
	-3	固定	Reset	复位	0x0000 0004
	-2	固定	NMI	不可屏蔽中断。RCC 时钟安全系统 (CSS) 连接到 NMI 向量。	0x0000 0008
	-1	固定	HardFault	所有类型的错误	0x0000 000C
	0	可设置	MemManage	存储器管理	0x0000 0010
	1	可设置	BusFault	预取指失败，存储器访问失败	0x0000 0014
	2	可设置	UsageFault	未定义的指令或非法状态	0x0000 0018
	-	-	-	保留	0x0000 001C - 0x0000 002B
	3	可设置	SVCall	通过 SWI 指令调用的系统服务	0x0000 002C
	4	可设置	Debug Monitor	调试监控器	0x0000 0030
	-	-	-	保留	0x0000 0034
	5	可设置	PendSV	可挂起的系统服务	0x0000 0038
	6	可设置	SysTick	系统嘀嗒定时器	0x0000 003C
0	7	可设置	WWDG	窗口看门狗中断	0x0000 0040
1	8	可设置	PVD	连接到 EXTI 线的可编程电压检测 (PVD) 中断	0x0000 0044
2	9	可设置	TAMP_STAMP	连接到 EXTI 线的入侵和时间戳中断	0x0000 0048
4	11	可设置	FLASH	Flash 全局中断	0x0000 0050
5	12	可设置	RCC	RCC 全局中断	0x0000 0054
6	13	可设置	EXTI0	EXTI 线 0 中断	0x0000 0058
7	14	可设置	EXTI1	EXTI 线 1 中断	0x0000 005C
8	15	可设置	EXTI2	EXTI 线 2 中断	0x0000 0060
9	16	可设置	EXTI3	EXTI 线 3 中断	0x0000 0064
10	17	可设置	EXTI4	EXTI 线 4 中断	0x0000 0068
11	18	可设置	DMA1_Stream0	DMA1 流 0 全局中断	0x0000 006C
12	19	可设置	DMA1_Stream1	DMA1 流 1 全局中断	0x0000 0070
13	20	可设置	DMA1_Stream2	DMA1 流 2 全局中断	0x0000 0074

14	21	可设置	DMA1_Stream3	DMA1 流 3 全局中断	0x0000 0078
15	22	可设置	DMA1_Stream4	DMA1 流 4 全局中断	0x0000 007C
16	23	可设置	DMA1_Stream5	DMA1 流 5 全局中断	0x0000 0080
17	24	可设置	DMA1_Stream6	DMA1 流 6 全局中断	0x0000 0084
18	25	可设置	ADC	ADC 全局中断	0x0000 0088

表 45. STC33F 的向量表 （续）

位置	优先级	优先级类型	名称	说明	地址
23	30	可设置	EXTI9_5	EXTI 线 [9:5] 中断	0x0000 009C
24	31	可设置	TIM1_BRK_TIM9	TIM1 刹车中断和 TIM9 全局中断	0x0000 00A0
25	32	可设置	TIM1_UP_TIM10	TIM1 更新中断和 TIM10 全局中断	0x0000 00A4
26	33	可设置	TIM1_TRG_COM_TIM11	TIM1 触发和换相中断与 TIM11 全局中断	0x0000 00A8
27	34	可设置	TIM1_CC	TIM1 捕获比较中断	0x0000 00AC
28	35	可设置	TIM2	TIM2 全局中断	0x0000 00B0
29	36	可设置	TIM3	TIM3 全局中断	0x0000 00B4
30	37	可设置	TIM4	TIM4 全局中断	0x0000 00B8
31	38	可设置	I2C1_EV	I^2C1 事件中断	0x0000 00BC
32	39	可设置	I2C1_ER	I^2C1 错误中断	0x0000 00C0
33	40	可设置	I2C2_EV	I^2C2 事件中断	0x0000 00C4
34	41	可设置	I2C2_ER	I^2C2 错误中断	0x0000 00C8
35	42	可设置	SPI1	SPI1 全局中断	0x0000 00CC
36	43	可设置	SPI2	SPI2 全局中断	0x0000 00D0
37	44	可设置	USART1	USART1 全局中断	0x0000 00D4
38	45	可设置	USART2	USART2 全局中断	0x0000 00D8
39	46	可设置	USART3	USART3 全局中断	0x0000 00DC
40	47	可设置	EXTI15_10	EXTI 线 [15:10] 中断	0x0000 00E0
43	50	可设置	TIM8_BRK_TIM12	TIM8 刹车中断和 TIM12 全局中断	0x0000 00EC
44	51	可设置	TIM8_UP_TIM13	TIM8 更新中断和 TIM13 全局中断	0x0000 00F0
45	52	可设置	TIM8_TRG_COM_TIM14	TIM8 触发和换相中断与 TIM14 全局中断	0x0000 00F4

STCMCU

46	53	可设置	TIM8_CC	TIM8 捕捉比较中断	0x0000 00F8
47	54	可设置	DMA1_Stream7	DMA1 流 7 全局中断	0x0000 00FC
48	55	可设置	FSMC	FSMC 全局中断	0x0000 0100

表 45. STC33F 的向量表 （续）

位置	优先级	优先级类型	名称	说明	地址
50	57	可设置	TIM5	TIM5 全局中断	0x0000 0108
52	59	可设置	UART4	UART4 全局中断	0x0000 0110
54	61	可设置	TIM6	TIM6 全局中断	0x0000 0118
55	62	可设置	TIM7	TIM7 全局中断	0x0000 011C
56	63	可设置	DMA2_Stream0	DMA2 流 0 全局中断	0x0000 0120
57	64	可设置	DMA2_Stream1	DMA2 流 1 全局中断	0x0000 0124
58	65	可设置	DMA2_Stream2	DMA2 流 2 全局中断	0x0000 0128
59	66	可设置	DMA2_Stream3	DMA2 流 3 全局中断	0x0000 012C
60	67	可设置	DMA2_Stream4	DMA2 流 4 全局中断	0x0000 0130
68	75	可设置	DMA2_Stream5	DMA2 流 5 全局中断	0x0000 0150
69	76	可设置	DMA2_Stream6	DMA2 流 6 全局中断	0x0000 0154
70	77	可设置	DMA2_Stream7	DMA2 流 7 全局中断	0x0000 0158
81	88	可设置	FPU	FPU 全局中断	0x0000 0184

11.2外部中断/ 事件控制器 (EXTI)

外部中断/事件控制器包含多达 23 个用于产生事件/中断请求的边沿检测器。每根输入线都可单独进行配置，以选择类型（中断或事件）和相应的触发事件（上升沿触发、下降沿触发或边沿触发）。每根输入线还可单独屏蔽。挂起寄存器用于保持中断请求的状态线。

11.2.1EXTI 主要特性

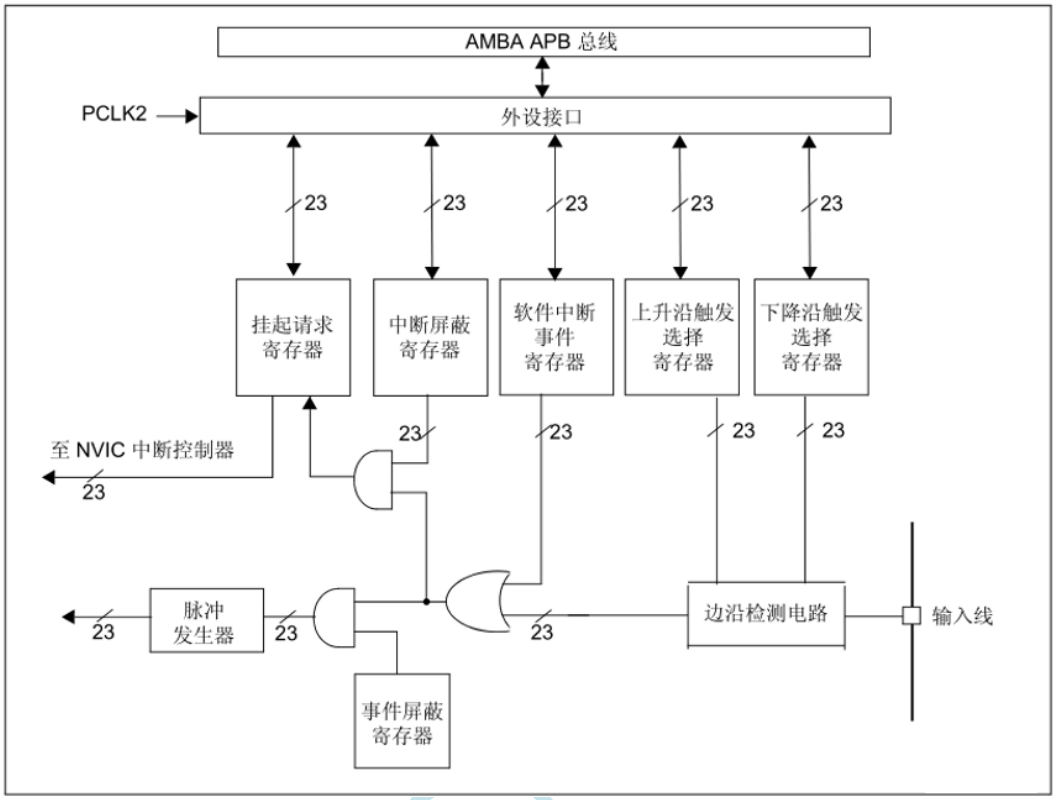
EXTI 控制器的主要特性如下：

- 每个中断/事件线上都具有独立的触发和屏蔽
- 每个中断线都具有专用的状态位
- 支持多达 23 个软件事件/中断请求
- 检测脉冲宽度低于 APB2 时钟宽度的外部信号。

11.2.2EXTI 框图

图 32 显示了框图。

图 32. 外部中断/ 事件控制器框图



11.2.3唤醒事件管理

STC33F 能够处理外部或内部事件来唤醒内核 (WFE)。唤醒事件可通过以下方式产生：

- 在外设的控制寄存器使能一个中断，但不在 NVIC 中使能，同时使能 Cortex™-MSTAR 系统控制寄存器中的 SEVONPEND 位。当 MCU 从 WFE 恢复时，需要清除相应外设的中断挂起位和外设 NVIC 中断通道挂起位（在 NVIC 中断清除挂起寄存器中）。
- 配置一个外部或内部 EXTI 线为事件模式。当 CPU 从 WFE 恢复时，因为对应事件线的挂起位没有被置位，不必清除相应外设的中断挂起位或 NVIC 中断通道挂起位。

11.2.4功能说明

要产生中断，必须先配置好并使能中断线。根据需要的边沿检测设置 2 个触发寄存器，同时在中断屏蔽寄存器的相应位写“1”使能中断请求。当外部中断线上出现选定信号沿时，便会产生中断请求，对应的挂起位也会置 1。在挂起寄存器的对应位写“1”，将清除该中断请求。要产生事件，必须先配置好并使能事件线。根据需要的边沿检测设置 2 个触发寄存器，同时在事件屏蔽寄存器的相应位写“1”允许事件请求。当事件线上出现选定信号沿时，便会产生事件脉冲，对应的挂起位不会置 1。
通过在软件中对软件中断/事件寄存器写“1”，也可以产生中断/事件请求。

硬件中断选择

要配置 23 根线作为中断源，请执行以下步骤：

- 配置 23 根中断线的屏蔽位 (EXTI_IMR)
- 配置中断线的触发选择位（EXTI_RTSR 和 EXTI_FTSR）
- 配置对应到外部中断控制器 (EXTI) 的 NVIC 中断通道的使能和屏蔽位，使得 23 个中断线中的请求可以被正确地响应。

硬件事件选择

要配置 23 根线作为事件源，请执行以下步骤：

- 配置 23 根事件线的屏蔽位 (EXTI_EMR)
- 配置事件线的触发选择位（EXTI_RTSR 和 EXTI_FTSR）

软件中断/ 事件选择

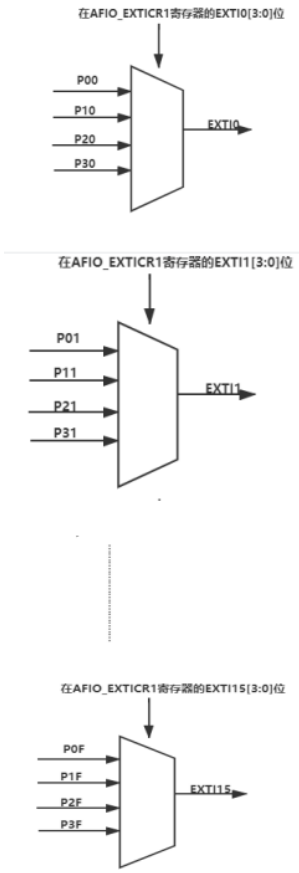
可将这 23 根线配置为软件中断/事件线。以下为产生软件中断的步骤。

- 配置 23 根中断/事件线的屏蔽位（EXTI_IMR、EXTI_EMR）
- 在软件中断寄存器设置相应的请求位 (EXTI_SWIER)

11.2.5外部中断/ 事件线映射

多个 GPIO 通过以下方式连接到 16 个外部中断/事件线：

图 33. 外部中断/ 事件 GPIO 映射



另外七根 EXTI 线连接方式如下：

- EXTI 线 16 连接到 PVD 输出
- EXTI 线 18 连接到 USB 唤醒事件
- EXTI 线 19
- EXTI 线 20 连接到 USB （在 FS 中配置）唤醒事件

11.2.6EXTI 寄存器

有关寄存器说明中使用的缩写，请参见第 1.1 节 。

11.2.7 中断屏蔽寄存器 (EXTI_IMR)

Interrupt mask register
偏移地址：0x00
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									MR22	MR21	MR20	MR19	MR18	MR17	MR16
									rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR15	MR14	MR13	MR12	MR11	MR10	MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:23 保留，必须保持复位值。
位 22:0 MRx ：x 线上的中断屏蔽 (Interrupt mask on line x)
0：屏蔽来自 x 线的中断请求
1：开放来自 x 线的中断请求

11.2.8 事件屏蔽寄存器 (EXTI_EMR)

Event mask register
偏移地址：0x04
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									MR22	MR21	MR20	MR19	MR18	MR17	MR16
									rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR15	MR14	MR13	MR12	MR11	MR10	MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:23 保留，必须保持复位值。
位 22:0 MRx ：x 线上的事件屏蔽 (Event mask on line x)
0：屏蔽来自 x 线的事件请求
1：开放来自 x 线的事件请求

11.2.9 上升沿触发选择寄存器 (EXTI_RTSR)

Rising trigger selection register
偏移地址：0x08
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									TR22	TR21	TR20	TR19	TR18	TR17	TR16
									rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:23 保留，必须保持复位值。
位 22:0 TRx ：线 x 的上升沿触发事件配置位 (Rising trigger event configuration bit of line x)

STC MCU

0：禁止输入线上升沿触发（事件和中断）
1：允许输入线上升沿触发（事件和中断）

注意： 外部唤醒线配置为边沿触发时，在这些线上不能出现毛刺信号。
如果在向 EXTI_RTSR 寄存器写入值的同时外部中断线上产生上升沿，挂起位将被置位。
在同一中断线上，可以同时设置上升沿和下降沿触发。即任一边沿都可触发中断。

11.2.10 下降沿触发选择寄存器 (EXTI_FTSR)

Falling trigger selection register
偏移地址：0x0C
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									TR22	TR21	TR20	TR19	TR18	TR17	TR16
									rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:23 保留，必须保持复位值。
位 22:0 TRx ：线 x 的下降沿触发事件配置位 (Falling trigger event configuration bit of line x)
0：禁止输入线下降沿触发（事件和中断）
1：允许输入线下降沿触发（事件和中断）

注意： 外部唤醒线配置为边沿触发时，在这些线上不能出现毛刺信号。
如果在向 EXTI_FTSR 寄存器写入值的同时外部中断线上产生下降沿，挂起位不会被置位。
在同一中断线上，可以同时设置上升沿和下降沿触发。即任一边沿都可触发中断。

11.2.11 软件中断事件寄存器 (EXTI_SWIER)

Software interrupt event register
偏移地址：0x10
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									SWIE R22	SWIE R21	SWIE R20	SWIE R19	SWIE R18	SWIE R17	SWIE R16
									rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIE R15	SWIE R14	SWIE R13	SWIE R12	SWIER R11	SWIE R10	SWIE R9	SWIE R8	SWIE R7	SWIE R6	SWIE R5	SWIE R4	SWIE R3	SWIE R2	SWIE R1	SWIE R0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:23 保留，必须保持复位值。
位 22:0 SWIERx ：线 x 上的软件中断 (Software Interrupt on line x)
当该位为“0”时，写“1”将设置 EXTI_PR 中相应的挂起位。如果在 EXTI_IMR 和 EXTI_EMR 中允许产生该中断，则产生中断请求。
通过清除 EXTI_PR 的对应位（写入“1”），可以清除该位为“0”。

11.2.12 挂起寄存器 (EXTI_PR)

Pending register
偏移地址：0x14
复位值：未定义

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									PR22	PR21	PR20	PR19	PR18	PR17	PR16
									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
PR15	PR14	PR13	PR12	PR11	PR10	PR9	PR8	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0
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 31:23 保留，必须保持复位值。
位 22:0 PRx ：挂起位 (Pending bit)
0：没有发生触发请求
1：发生了选择的触发请求
当在外部中断线上发生了选择的边沿事件，该位被置“1”。在此位中写入“1”可以清除它，也可以通过改变边沿检测的极性清除。

11.2.13EXTI 寄存器映射

表 47 给出了 EXTI 寄存器映射和复位值。

表 47. 外部中断/事件控制器寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	EXTI_IMR	Reserved									MR[22:0]																						
	Reset value										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	EXTI_EMR	Reserved									MR[22:0]																						
	Reset value										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	EXTI_RTSR	Reserved									TR[22:0]																						
	Reset value										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0c	EXTI_FTSR	Reserved									TR[22:0]																						
	Reset value										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	EXTI_SWIER	Reserved									SWIER[22:0]																						
	Reset value										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	EXTI_PR	Reserved									PR[22:0]																						
	Reset value										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

STCMCU

12 模数转换器 (ADC)

除非特别说明，否则本部分适用于整个STC33F系列。

12.1ADC 简介

12位 ADC 是逐次趋近型模数转换器, 它具有多达 15 个复用通道。这些通道的 A/D 转换可在单次、连续、扫描或不连续 采样模式下进行。ADC 的结果存储在一个左对齐或右对齐的 16 位数据寄存器中。

12.2ADC 主要特性

- 12 位分辨率
- 在转换结束、注入转换结束或溢出事件时产生中断
- 单次和连续转换模式
- 用于自动将通道 0 转换为通道“n”的扫描模式
- 数据对齐以保持内置数据一致性
- 可独立设置各通道采样时间
- 外部触发器选项，可为规则转换和注入转换配置极性
- 不连续采样模式
- ADC 转换类型（参见数据手册）
- ADC 电源要求：全速运行时为 2.4 V 到 5.5V，慢速运行时为 1.8 V
- ADC 输入范围：$V_{REF-} \le V_{IN} \le V_{REF+}$
- 规则通道转换期间可产生 DMA 请求

图 34 显示了 ADC 的框图。

注意： *V_{REF-}如果可用（取决于封装），则必须将其连接到 V_{SSA}。*

12.3ADC 功能说明

表 48 列出了 ADC 的引脚说明。

STC MCU

表 48. ADC 引脚

名称	信号类型	备注
V_{REF+}	正模拟参考电压输入	ADC 高/正参考电压，1.8 V ≤ V_{REF+} ≤ V_{DDA}
V_{DDA}	模拟电源输入	模拟电源电压等于 V_{DD}， 全速运行时，2.4 V ≤ V_{DDA} ≤ V_{DD} (5.5 V) 低速运行时，1.9 V ≤ V_{DDA} ≤ V_{DD} (5.5 V)
V_{REF-}	负模拟参考电压输入	ADC 低/负参考电压，V_{REF-}= V_{SSA}
V_{SSA}	模拟电源接地输入	模拟电源接地电压等于 V_{SS}
ADCx_IN[15:0]	模拟输入信号	15 个模拟输入通道

12.3.1 ADC 开关控制

可通过将 ADC_CR2 寄存器中的 ADON 位置 1 来为 ADC 供电。首次将 ADON 位置 1 时，会将 ADC 从掉电模式中唤醒。

SWSTART 或 JSWSTART 位置 1 时，启动 AD 转换。

可通过将 ADON 位清零来停止转换并使 ADC 进入掉电模式。在此模式下，ADC 几乎不耗电（只有几 μA）。

12.3.2 ADC 时钟

ADC 具有两个时钟方案：

- 用于模拟电路的时钟：ADCCLK，所有 ADC 共用
  此时钟来自于经可编程预分频器分频的 APB2 时钟，该预分频器允许 ADC 在 f_{PCLK2}/2、/4、/6 或 /8 下工作。有关 ADCCLK 的最大值，请参见数据手册。
- 用于数字接口的时钟（用于寄存器读/写访问）
  此时钟等效于 APB2 时钟。可以通过 RCC APB2 外设时钟使能寄存器 (RCC_APB2ENR) 分别为每个 ADC 使能/禁止数字接口时钟。

12.3.3 通道选择

有 15条复用通道。可以将转换分为两组：规则转换和注入转换。每个组包含一个转换序列， 该序列可按任意顺序在任意通道上完成。例如，可按以下顺序对序列进行转换：ADC_IN3、ADC_IN8、ADC_IN2、ADC_IN2、ADC_IN0、ADC_IN2、ADC_IN2、ADC_IN15。

- 一个**规则转换组**最多由 15个转换构成。必须在 ADC_SQRx 寄存器中选择转换序列的规则通道及其顺序。规则转换组中的转换总数必须写入 ADC_SQR1 寄存器中的 L[3:0] 位。
- 一个注入转换组最多由4个转换构成。必须在ADC_JSQR寄存器中选择转换序列的注入通道及其顺序。注入转换组中的转换总数必须写入 ADC_JSQR寄存器中的 L[1:0]位。

如果在转换期间修改 ADC_SQRx 或 ADC_JSQR 寄存器，将复位当前转换并向 ADC 发送一 个新的启动脉冲，以转换新选择的组。

12.3.4 单次转换模式

在单次转换模式下，ADC 执行一次转换。CONT 位为 0 时，可通过以下方式启动此模式：

- 将 ADC_CR2 寄存器中的 SWSTART 位置 1（仅适用于规则通道）
- 将 JSWSTART 位置 1（适用于注入通道）
- 外部触发（适用于规则通道或注入通道）

完成所选通道的转换之后：

- 如果转换了规则通道：

- 转换数据存储在 16 位 ADC_DR 寄存器中
- EOC（转换结束）标志置 1
- EOCIE 位置 1 时将产生中断

- 如果转换了注入通道：
  - 转换数据存储在 16 位 ADC_JDR1 寄存器中
  - JEOC（注入转换结束）标志置 1
  - JEOCIE 位置 1 时将产生中断

然后，ADC 停止。

12.3.5连续转换模式

在连续转换模式下，ADC 结束一个转换后立即启动一个新的转换。CONT 位为 1 时，可通过外部触发或将 ADC_CR2 寄存器中的 SWSTRT 位置 1 来启动此模式（仅适用于规则通道）。

每次转换之后：

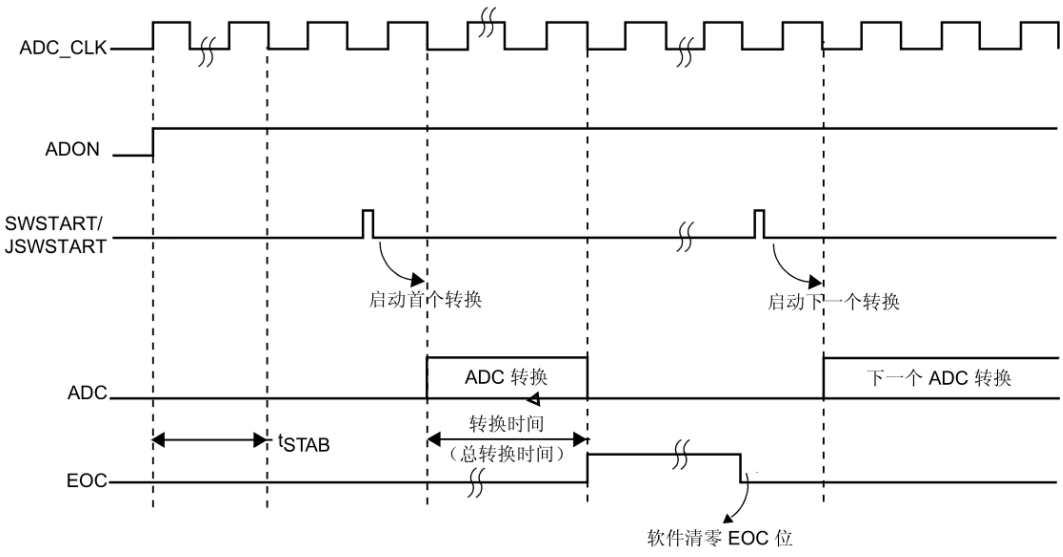
- 如果转换了规则通道组：
  - 上次转换的数据存储在 16 位 ADC_DR 寄存器中
  - EOC（转换结束）标志置 1
  - EOCIE 位置 1 时将产生中断

注意： *无法连续转换注入通道。连续模式下唯一的例外情况是，注入通道配置为在规则通道之后自动转换（使用 JAUTO 位），请参见自动注入一节。*

12.3.6时序图

如*图 35* 所示，ADC 在开始精确转换之前需要一段稳定时间 t_{STAB}。ADC 开始转换并经过 15 个 时钟周期后，EOC 标志置 1，转换结果存放在 16 位 ADC 数据寄存器中。

图 35. 时序图



12.3.7扫描模式

此模式用于扫描一组模拟通道。

通过将 ADC_CR1 寄存器中的 SCAN 位置 1 来选择扫描模式。将此位置 1 后，ADC 会扫描在 ADC_SQRx 寄存器（对于规则通道）或 ADC_JSQR 寄存器（对于注入通道）中选择的 所有通道。为组中的每个通道都执行一次转换。每次转换结束后，会自动转换该组中的下一个

通道。如果将 CONT 位置 1，规则通道转换不会在组中最后一个所选通道处停止，而是再次从第一个所选通道继续转换。

如果将 DMA 位置 1，则在每次规则通道转换之后，均使用直接存储器访问 (DMA) 控制器将转换自规则通道组的数据（存储在 ADC_DR 寄存器中）传输到 SRAM。

在以下情况下，ADC_SR 寄存器中的 EOC 位置 1：

- 如果 EOCS 位清零，在每个规则组序列转换结束时
- 如果 EOCS 位置 1，在每个规则通道转换结束时

从注入通道转换的数据始终存储在 ADC_JDRx 寄存器中。

12.3.8注入通道管理

触发注入

要使用触发注入，必须将 ADC_CR1 寄存器中的 JAUTO 位清零。

1. 通过外部触发或将 ADC_CR2 寄存器中的 SWSTART 位置 1 来启动规则通道组转换。
2. 如果在规则通道组转换期间出现外部注入触发或者 JSWSTART 位置 1，则当前的转换会复位，并且注入通道序列会切换为单次扫描模式。
3. 然后，规则通道组的规则转换会从上次中断的规则转换处恢复。

   如果在注入转换期间出现规则事件，注入转换不会中断，但在注入序列结束时会执行规则序列。*图 37* 显示了相应的时序图。

注意： *使用触发注入时，必须确保触发事件之间的间隔长于注入序列。例如，如果序列长度为 30 个 ADC 时钟周期（即，采样时间为 3 个时钟周期的两次转换），则触发事件的最小间隔不能小于 31 个 ADC 时钟周期。*

自动注入

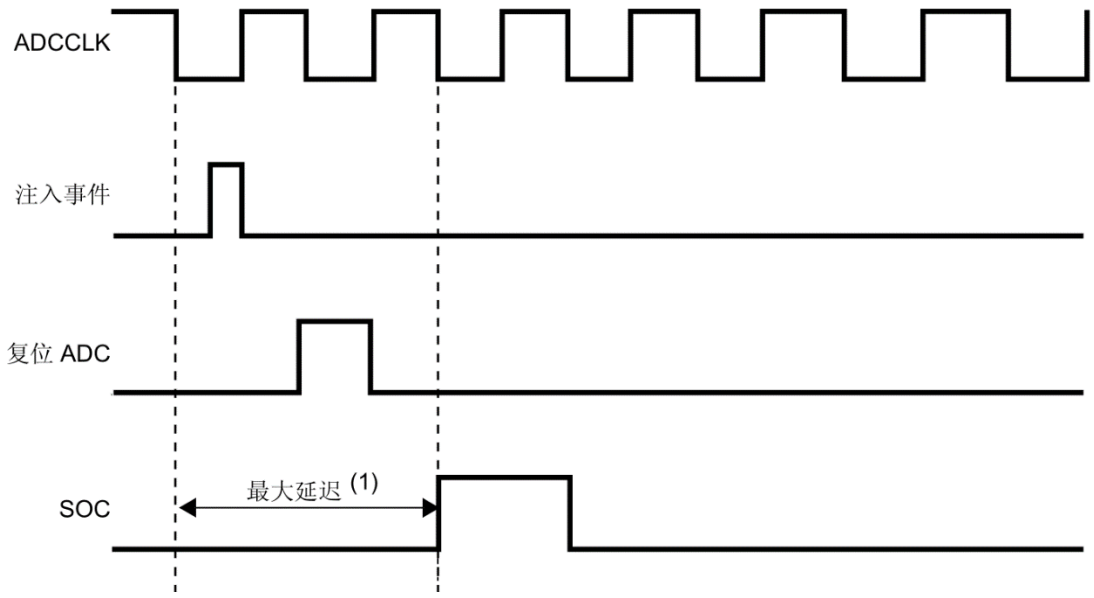
如果将 JAUTO 位置 1，则注入组中的通道会在规则组通道之后自动转换。这可用于转换最多由 20 个转换构成的序列，这些转换在 ADC_SQRx 和 ADC_JSQR 寄存器中编程。

在此模式下，必须禁止注入通道上的外部触发。

如果 CONT 位和 JAUTO 位均已置 1，则在转换规则通道之后会继续转换注入通道。

注意： *不能同时使用自动注入和不连续采样模式。*

图 37. 注入转换延迟



12.3.9不连续采样模式

规则组

可将 ADC_CR1 寄存器中的 DISCEN 位置 1 来使能此模式。该模式可用于转换含有 n (n ≤ 8) 个转换的短序列，该短序列是在 ADC_SQRx 寄存器中选择的转换序列的一部分。可通过写入 ADC_CR1 寄存器中的 DISCNUM[2:0] 位来指定 n 的值。

出现外部触发时，将启动在 ADC_SQRx 寄存器中选择的接下来 n 个转换，直到序列中的所有转换均完成为止。通过 ADC_SQR1 寄存器中的 L[3:0] 位定义总序列长度。

示例：

n = 3，要转换的通道 = 0、1、2、3、6、7、9、10

第 1 次触发：转换序列 0、1、2

第 2 次触发：转换序列 3、6、7

第 3 次触发：转换序列 9、10 并生成 EOC 事件

第 4 次触发：转换序列 0、1、2

注意： *在不连续采样模式下转换规则组时，不会出现翻转。*

转换完所有子组后，下一个触发信号将启动第一个子组的转换。在上述示例中，第 4 次触发重新转换了第 1 个子组中的通道 0、1 和 2。

注入组

可将 ADC_CR1 寄存器中的 JDISCEN 位置 1 来使能此模式。在出现外部触发事件之后，可使用该模式逐通道转换在 ADC_JSQR 寄存器中选择的序列。

出现外部触发时，将启动在 ADC_JSQR 寄存器中选择的下一个通道转换，直到序列中的所有转换均完成为止。通过 ADC_JSQR 寄存器中的 JL[1:0] 位定义总序列长度。

示例：

n = 1，要转换的通道 = 1、2、3

第 1 次触发：转换通道 1

第 2 次触发：转换通道 2

第 3 次触发：转换通道 3 并生成 EOC 和 JEOC 事件

第 4 次触发：通道 1

注意： *转换完所有注入通道后，下一个触发信号将启动第一个注入通道的转换。在上述示例中，第 4 次触发重新转换了第 1 个注入通道。*

不能同时使用自动注入和不连续采样模式。

不得同时为规则组和注入组设置不连续采样模式。只能针对一个组使能不连续采样模式。

12.4数据对齐

ADC_CR2 寄存器中的 ALIGN 位用于选择转换后存储的数据的对齐方式。可选择左对齐和右对齐两种方式，如*图 38* 和*图 39* 所示。

注入通道组的转换数据将减去 ADC_JOFRx 寄存器中写入的用户自定义偏移量，因此结果可以是一个负值。SEXT 位表示扩展的符号值。

对于规则组中的通道，不会减去任何偏移量，因此只有十二个位有效。

图 38. 12 位数据的右对齐

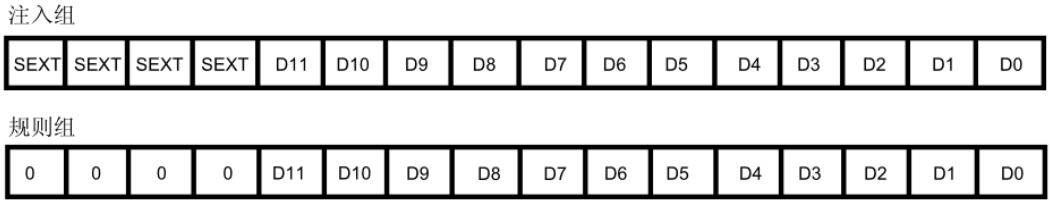
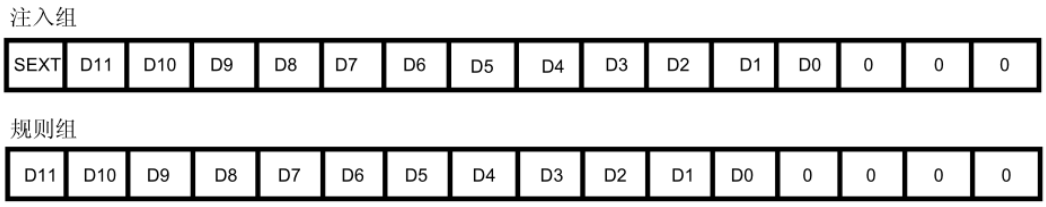
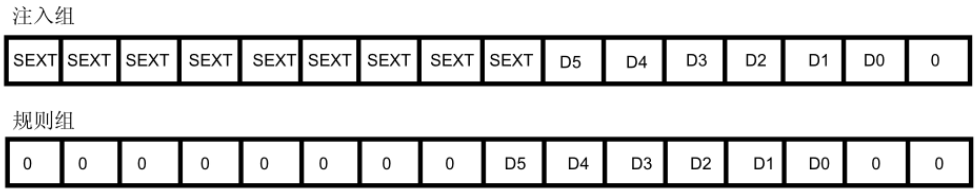


图 39. 12 位数据的左对齐



特例：采用左对齐时，数据基于半字进行对齐，除了分辨率设置为 6 位时。分辨率设置为 6 位 时，数据基于字节进行对齐，如*图 40* 所示。

图 40. 6 位数据的左对齐



12.5可独立设置各通道采样时间

ADC 会在数个 ADCCLK 周期内对输入电压进行采样，可使用 ADC_SMPR1 和 ADC_SMPR2 寄存器中的 SMP[2:0] 位修改周期数。每个通道均可以使用不同的采样时间进行采样。

总转换时间的计算公式如下：

T_{conv} = 采样时间 + 12 个周期

示例：

ADCCLK = 30 MHz 且采样时间 = 3 个周期时：

T_{conv} = 3 + 12 = 15 个周期 = 0.5 μs（APB2 为 60 MHz 时）

12.6外部触发转换和触发极性

可以通过外部事件（例如，定时器捕获、EXTI 中断线）触发转换。如果 EXTEN[1:0] 控制位 （对于行规转换）或 JEXTEN[1:0] 位（对于注入转换）不等于“0b00”，则外部事件能够以所选极性触发转换。*表 50* 提供了 EXTEN[1:0] 和 JEXTEN[1:0] 值与触发极性之间的对应关系。

表 50. 配置触发极性

源	EXTEN[1:0]/JEXTEN[1:0]
禁止触发检测	00
在上升沿时检测	01
在下降沿时检测	10
在上升沿和下降沿均检测	11

注意： *可以实时更改外部触发的极性。*

EXTSEL[3:0] 和 JEXTSEL[3:0] 控制位用于从 16 个可能事件中选择可触发规则组转换和注入组转换的事件。

表 51 给出了可用于规则转换的外部触发。

表 51. 规则通道的外部触发

源	类型	EXTSEL[3:0]
PWM1_CH1 事件	片上定时器的内部信号	0000
PWM1_CH2 事件		0001
PWM1_CH3 事件		0010
PWM2_CH2 事件		0011
PWM2_CH3 事件		0100
PWM2_CH4 事件		0101
PWM2_TRGO 事件		0110
PWM3_CH1 事件		0111
PWM3_TRGO 事件		1000
PWM4_CH4 事件		1001

STC MCU

PWM5_CH1 事件		1010
PWM5_CH2 事件		1011
PWM5_CH3 事件		1100
PWM8_CH1 事件		1101
PWM8_TRGO 事件		1110
EXTI 线 11	外部引脚	1111

表 52 给出了可用于注入转换的外部触发。

表 52. 注入通道的外部触发

源	连接类型	JEXTSEL[3:0]
PWM1_CH4 事件	片上定时器的内部信号	0000
PWM1_TRGO 事件		0001
PWM2_CH1 事件		0010
PWM2_TRGO 事件		0011
PWM3_CH2 事件		0100
PWM3_CH4 事件		0101
PWM4_CH1 事件		0110
PWM4_CH2 事件		0111
PWM4_CH3 事件		1000
PWM4_TRGO 事件		1001
PWM5_CH4 事件		1010
PWM5_TRGO 事件		1011
PWM8_CH2 事件		1100
PWM8_CH3 事件		1101
PWM8_CH4 事件		1110
EXTI 线 15	外部引脚	1111

可通过将 ADC_CR2 寄存器中的 SWSTART（对于规则转换）或 JSWSTART（对于注入转换）位置 1 来产生软件源触发事件。

可通过注入触发中断规则组转换。

注意： 可以实时更改触发选择。不过，当更改触发选择时，会在 1 个 APB 时钟周期的时间范围内禁止触发检测。这是为了避免在转换期间出现意外检测。

12.7快速转换模式

可通过降低 ADC 分辨率来执行快速转换。RES 位用于选择数据寄存器中可用的位数。每种分辨率的最小转换时间如下：

- 12 位：3 + 12 = 15 ADCCLK 周期
- 10 位：3 + 10 = 13 ADCCLK 周期
- 8 位：3 + 8 = 11 ADCCLK 周期
- 6 位：3 + 6 = 9 ADCCLK 周期

STC MCU

12.8数据管理

12.8.1使用 DMA

由于规则通道组只有一个数据寄存器，因此，对于多个规则通道的转换，使用 DMA 非常有帮助。这样可以避免丢失在下一次写入之前还未被读出的 ADC_DR 寄存器中的数据。

在使能 DMA 模式的情况下（ADC_CR2 寄存器中的 DMA 位置 1），每完成规则通道组中的一个通道转换后，都会生成一个 DMA 请求。这样便可将转换的数据从 ADC_DR 寄存器传输到用软件选择的目标位置。

尽管如此，如果数据丢失（溢出），则会将 ADC_SR 寄存器中的 OVR 位置 1 并生成一个中断（如果 OVRIE 使能位已置 1）。随后会禁止 DMA 传输并且不再接受 DMA 请求。在这种情况下，如果生成 DMA 请求，则会中止正在进行的规则转换并忽略之后的规则触发。随后需要将所使用的 DMA 流中的 OVR 标志和 DMAEN 位清零，并重新初始化 DMA 和 ADC，以将需要的转换通道数据传输到正确的存储器单元。只有这样，才能恢复转换并再次使能数据传输。注入通道转换不会受到溢出错误的影响。

在 DMA 模式下，当 OVR = 1 时，传送完最后一个有效数据后会阻止 DMA 请求，这意味着传输到 RAM 的所有数据均被视为有效。

在最后一次 DMA 传输（DMA 控制器的 DMA_SxRTR 寄存器中配置的传输次数）结束时：

- 如果将 ADC_CR2 寄存器中的 DDS 位清零，则不会向 DMA 控制器发出新的 DMA 请求（这可避免产生溢出错误）。不过，硬件不会将 DMA 位清零。必须将该位写入 0 然后写入 1 才能启动新的传输。
- 如果将 DDS 位置 1，则可继续生成请求。从而允许在双缓冲区循环模式下配置 DMA。

要在使用 DMA 时将 ADC 从 OVR 状态中恢复，请按以下步骤操作：

1. 重新初始化 DMA（调整目标地址和 NDTR 计数器）
2. 将 ADC_SR 寄存器中的 ADC OVR 位清零
3. 触发 ADC 以开始转换。

12.8.2在不使用 DMA 的情况下管理转换序列

如果转换过程足够慢，则可使用软件来处理转换序列。在这种情况下，必须将 ADC_CR2 寄存器中的 EOCS 位置 1，才能使 EOC 状态位在每次转换结束时置 1，而不仅是在序列结束时置 1。当 EOCS = 1 时，会自动使能溢出检测。因此，每当转换结束时，EOC 都会置 1，并且可以读取 ADC_DR 寄存器。溢出管理与使用 DMA 时的管理相同。

要在 EOCS 位置 1 时将 ADC 从 OVR 状态中恢复，请按以下步骤操作：

1. 将 ADC_SR 寄存器中的 ADC OVR 位清零
2. 触发 ADC 以开始转换。

12.8.3在不使用 DMA 和溢出检测的情况下进行转换

ADC 在转换一个或多个通道时不是每次都读取数据的情况下，这可能会很有用。为此，必须禁止 DMA (DMA = 0) 并且仅在序列结束 (EOCS = 0) 时才将EOC 位置 1。在此配置中，溢出检测已禁止。

12.9 ADC 中断

ADC_SR 寄存器中存在另外两个标志，但这两个标志不存在中断相关性：

- JSTRT（开始转换注入组的通道）
- STRT（开始转换规则组的通道）

表 53. ADC 中断

中断事件	事件标志	使能控制位
结束规则组的转换	EOC	EOCIE
结束注入组的转换	JEOC	JEOCIE
溢出 (Overrun)	OVR	OVRIE

12.10 ADC 寄存器

有关寄存器说明中使用的缩写，请参见*第 1.1 节*。

必须在字级别（32 位）对外设寄存器执行写入操作。而读访问可支持字节（8 位）、半字（16 位）或字（32 位）。

12.10.1 ADC 状态寄存器 (ADC_SR)

ADC status register

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										OVR	STRT	JSTRT	JEOC	EOC	AWD
										rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位 31:6 保留，必须保持复位值。

位 5 **OVR**：溢出 (Overrun)

数据丢失时，硬件将该位置 1（在单一模式或双重/三重模式下）。但需要通过软件清零。溢出检测仅在 DMA = 1 或 EOCS = 1 时使能。

0：未发生溢出

1：发生溢出

位 4 **STRT**：规则通道开始标志 (Regular channel start flag)

规则通道转换开始时，硬件将该位置 1。但需要通过软件清零。0：未开始规则通道转换

1：已开始规则通道转换

位 3 **JSTRT**：注入通道开始标志 (Injected channel start flag)

注入组转换开始时，硬件将该位置 1。但需要通过软件清零。 0：未开始注入组转换

1：已开始注入组转换

位 2 **JEOC**：注入通道转换结束 (Injected channel end of conversion)
组内所有注入通道转换结束时，硬件将该位置 1。但需要通过软件清零。 0：转换未完成
1：转换已完成

位 1 **EOC**：规则通道转换结束 (Regular channel end of conversion)
规则组通道转换结束后，硬件将该位置 1。通过软件或通过读取 ADC_DR 寄存器将该位清零。 0：转换未完成 (EOCS=0) 或转换序列未完成 (EOCS=1)
1：转换已完成 (EOCS=0) 或转换序列已完成 (EOCS=1)

12.10.2 ADC 控制寄存器 1 (ADC_CR1)

ADC control register 1

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					OVRIE	RES		AWDEN	JAWDEN	Reserved					
					rw	rw	rw	rw	rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISCNUM[2:0]			JDISCEN	DISC EN	JAUTO	AWDSGL	SCAN	JEOCIE	AWDIE	EOCIE	AWDCH[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位31:27 保留，必须保持复位值。

位 26 **OVRIE**：溢出中断使能 (Overrun interrupt enable)
通过软件将该位置 1 和清零可使能/禁止溢出中断。
0：禁止溢出中断
1：使能溢出中断。OVR 位置 1 时产生中断。

位 25:24 **RES[1:0]**：分辨率 (Resolution)
通过软件写入这些位可选择转换的分辨率。
00：12 位（15 ADCCLK 周期）
01：10 位（13 ADCCLK 周期）
10：8 位（11 ADCCLK 周期）
11：6 位（9 ADCCLK 周期）

位 23:16 保留，必须保持复位值。

位 15:13 **DISCNUM[2:0]**：不连续采样模式通道计数 (Discontinuous mode channel count)
软件将写入这些位，用于定义在接收到外部触发后于不连续采样模式下转换的规则通道数。
000：1 个通道
001：2 个通道
...
111：8 个通道

位 12 **JDISCEN**：注入通道的不连续采样模式 (Discontinuous mode on injected channels)
通过软件将该位置 1 和清零可使能/禁止注入通道的不连续采样模式。0：禁止注入通道的不连续采样模式
1：使能注入通道的不连续采样模式

位 11 **DISCEN**：规则通道的不连续采样模式 (Discontinuous mode on regular channels)
通过软件将该位置 1 和清零可使能/禁止规则通道的不连续采样模式。0：禁止规则通道的不连续采样模式
1：使能规则通道的不连续采样模式

位 10 **JAUTO**：注入组自动转换 (Automatic injected group conversion)
通过软件将该位置 1 和清零可在规则组转换后分别使能/禁止注入组自动转

换。0：禁止注入组自动转换
1：使能注入组自动转换

位 8 **SCAN**：扫描模式 (Scan mode)
通过软件将该位置 1 和清零可使能/禁止扫描模式。在扫描模式下，转换通过 ADC_SQRx 或 ADC_JSQRx 寄存器选择的输入。
0：禁止扫描模式
1：使能扫描模式
注意：EOCIE 位置 1 时将生成 EOC 中断：
— *如果 EOCS 位清零，在每个规则组序列转换结束时*
— *如果 EOCS 位置 1，在每个规则通道转换结束时*
注意：JEOCIE 位置 1 时，JEOC 中断仅在最后一个通道转换结束时生成。

位 7 **JEOCIE**：注入通道的中断使能 (Interrupt enable for injected channels)
通过软件将该位置 1 和清零可使能/禁止注入通道的转换结束中断。
0：禁止JEOC 中断
1：使能 JEOC 中断。JEOC 位置 1 时产生中断。

位 5 **EOCIE**：EOC 中断使能 (Interrupt enable for EOC)
通过软件将该位置 1 和清零可使能/禁止转换结束中断。0：禁止 EOC 中断
1：使能 EOC 中断EOC 位置 1 时产生中断。

12.10.3 ADC 控制寄存器 2 (ADC_CR2)

ADC control register 2

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
reserved	SWSTART	EXTEN		EXTSEL[3:0]				reserved	JSWSTART	JEXTEN		JEXTSEL[3:0]			
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserved				ALIGN	EOCS	DDS	DMA	Reserved						CONT	ADON
				rw	rw	rw	rw							rw	rw

位 31 保留，必须保持复位值。

位 30 **SWSTART**：开始转换规则通道 (Start conversion of regular channels)
通过软件将该位置 1 可开始转换，而硬件会在转换开始后将该位清零。
0：复位状态
1：开始转换规则通道
注意：该位只能在 ADON = 1 时置 1，否则不会启动转换。

位 29:28 **EXTEN**：规则通道的外部触发使能 (External trigger enable for regular channels)
通过软件将这些位置 1 和清零可选择外部触发极性和使能规则组的触发。
00：禁止触发检测
01：上升沿上的触发检测
10：下降沿上的触发检测
11：上升沿和下降沿上的触发检测

位 27:24 **EXTSEL[3:0]**：为规则组选择外部事件 (External event select for regular group)
这些位可选择用于触发规则组转换的外部事件。
0000：定时器 1 CC1 事件
0001：定时器 1 CC2 事件
0010：定时器 1 CC3 事件
0011：定时器 2 CC2 事件

0100：定时器 2 CC3 事件
0101：定时器 2 CC4 事件
0110：定时器 2 TRGO 事件
0111：定时器 3 CC1 事件
1000：定时器 3 TRGO 事件
1001：定时器 4 CC4 事件
1010：定时器 5 CC1 事件
1011：定时器 5 CC2 事件
1100：定时器 5 CC3 事件
1101：定时器 8 CC1 事件
1110：定时器 8 TRGO 事件
1111：EXTI 线 11

位 23 保留，必须保持复位值。

位 22 **JSWSTART**：开始转换注入通道 (Start conversion of injected channels)
转换开始后，软件将该位置 1，而硬件将该位清零。
0：复位状态
1：开始转换注入通道
注意：该位只能在 ADON = 1 时置 1，否则不会启动转换。

位 21:20 **JEXTEN**：注入通道的外部触发使能 (External trigger enable for injected channels)
通过软件将这些位置 1 和清零可选择外部触发极性和使能注入组的触发。
00：禁止触发检测
01：上升沿上的触发检测
10：下降沿上的触发检测
11：上升沿和下降沿上的触发检测

位 19:16 **JEXTSEL[3:0]**：为注入组选择外部事件 (External event select for injected group)
这些位可选择用于触发注入组转换的外部事件。
0000：定时器 1 CC4 事件
0001：定时器 1 TRGO 事件
0010：定时器 2 CC1 事件
0011：定时器 2 TRGO 事件
0100：定时器 3 CC2 事件
0101：定时器 3 CC4 事件
0110：定时器 4 CC1 事件
0111：定时器 4 CC2 事件
1000：定时器 4 CC3 事件
1001：定时器 4 TRGO 事件
1010：定时器 5 CC4 事件
1011：定时器 5 TRGO 事件
1100：定时器 8 CC2 事件
1101：定时器 8 CC3 事件
1110：定时器 8 CC4 事件
1111：EXTI 线 15

位 15:12 保留，必须保持复位值。

位 11 **ALIGN**：数据对齐 (Data alignment)
此位由软件置 1 和清零。请参见图 38 和图 39。
0：右对齐
1：左对齐

位 10 **EOCS**：结束转换选择 (End of conversion selection)
此位由软件置 1 和清零。
0：在每个规则转换序列结束时将 EOC 位置 1。溢出检测仅在 DMA=1 时使能。
1：在每个规则转换结束时将 EOC 位置 1。使能溢出检测。

位 9 **DDS**：DMA 禁止选择（对于单一 ADC 模式） (DMA disable selection (for single ADC mode))
此位由软件置 1 和清零。
0：最后一次传输后不发出新的 DMA 请求（在 DMA 控制器中进行配置）

1：只要发生数据转换且 DMA = 1，便会发出 DAM 请求

位 8 **DMA**：直接存储器访问模式（对于单一 ADC 模式）(Direct memory access mode (for single ADC mode))

此位由软件置 1 和清零。有关详细信息，请参见 DMA 控制器一章。

0：禁止 DMA 模式
1：使能 DMA 模式

位 7:2 保留，必须保持复位值。

位 1 **CONT**：连续转换 (Continuous conversion)

此位由软件置 1 和清零。该位置 1 时，转换将持续进行，直到该位清零。

0：单次转换模式
1：连续转换模式

位 0 **ADON**：A/D 转换器开启 / 关闭 (A/D Converter ON / OFF)

此位由软件置 1 和清零。

注意： 0：禁止 ADC 转换并转至掉电模式
1：使能 ADC

12.10.4 ADC 采样时间寄存器 1 (ADC_SMPR1)

ADC sample PWMe register 1

偏移地址：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					SMP18[2:0]			SMP17[2:0]			SMP16[2:0]			SMP15[2:1]	
					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
SMP15_0	SMP14[2:0]			SMP13[2:0]			SMP12[2:0]			SMP11[2:0]			SMP10[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:27 保留，必须保持复位值。

位 26:0 **SMPx[2:0]**：通道 X 采样时间选择 (Channel x sampling PWMe selection)

通过软件写入这些位可分别为各个通道选择采样时间。在采样周期期间，通道选择位必须保持不变。

注意： 000：3 个周期
001：15 个周期
010：28 个周期
011：56 个周期
100：84 个周期
101：112 个周期
110：144 个周期
111：480 个周期

12.10.5 ADC 采样时间寄存器 2 (ADC_SMPR2)

ADC sample PWMe register 2

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		SMP9[2:0]			SMP8[2:0]			SMP7[2:0]			SMP6[2:0]			SMP5[2:1]	
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

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

SMP5_0	SMP4[2:0]			SMP3[2:0]			SMP2[2:0]			SMP1[2:0]			SMP0[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 保留，必须保持复位值。

位 29:0 **SMPx[2:0]：**通道 X 采样时间选择 (Channel x sampling PWMe selection)

通过软件写入这些位可分别为各个通道选择采样时间。在采样周期期间，通道选择位必须保持不变。

注意： 000：3 个周期
001：15 个周期
010：28 个周期
011：56 个周期
100：84 个周期
101：112 个周期
110：144 个周期
111：480 个周期

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

ADC injected channel data offset register x

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

位 31:12 保留，必须保持复位值。

位 11:0 **JOFFSETx[11:0]：**注入通道 X 的数据偏移 (Data offset for injected channel x)

通过软件写入这些位可定义在转换注入通道时从原始转换数据中减去的偏移量。可从 ADC_JDRx 寄存器中读取转换结果。

12.10.7 ADC 规则序列寄存器 1 (ADC_SQR1)

ADC regular sequence register 1

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
Reserved								L[3:0]				SQ16[4:1]			
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ16_0	SQ15[4:0]					SQ14[4:0]					SQ13[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23:20 **L[3:0]：**规则通道序列长度 (Regular channel sequence length)

通过软件写入这些位可定义规则通道转换序列中的转换总数。0000：1 次转换
0001：2 次转换
...

1111：16 次转换

位 19:15 **SQ16[4:0]**：规则序列中的第十六次转换 (16th conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..18) 分配为转换序列中的第十六次转换。

位 14:10 **SQ15[4:0]**：规则序列中的第十五次转换 (15th conversion in regular sequence)

位 9:5 **SQ14[4:0]**：规则序列中的第十四次转换 (14th conversion in regular sequence)

位 4:0 **SQ13[4:0]**：规则序列中的第十三次转换 (13th conversion in regular sequence)

12.10.8 ADC 规则序列寄存器 2 (ADC_SQR2)

ADC regular sequence register 2

偏移地址：0x30

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		SQ12[4:0]					SQ11[4:0]					SQ10[4:1]			
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ10_0	SQ9[4:0]					SQ8[4:0]					SQ7[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 保留，必须保持复位值。

位 29:26 **SQ12[4:0]**：规则序列中的第十二次转换 (12th conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..18) 分配为序列中的第十二次转换。

位 24:20 **SQ11[4:0]**：规则序列中的第十一次转换 (11th conversion in regular sequence)

位 19:15 **SQ10[4:0]**：规则序列中的第十次转换 (10th conversion in regular sequence)

位 14:10 **SQ9[4:0]**：规则序列中的第九次转换 (9th conversion in regular sequence)

位 9:5 **SQ8[4:0]**：规则序列中的第八次转换 (8th conversion in regular sequence)

位 4:0 **SQ7[4:0]**：规则序列中的第七次转换 (7th conversion in regular sequence)

12.10.9 ADC 规则序列寄存器 3 (ADC_SQR3)

ADC regular sequence register 3

偏移地址：0x34

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		SQ6[4:0]					SQ5[4:0]					SQ4[4:1]			
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

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

位 31:30 保留，必须保持复位值。

位 29:25 **SQ6[4:0]**：规则序列中的第六次转换 (6th conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..18) 分配为序列中的第六次转换。

位 24:20 **SQ5[4:0]**：规则序列中的第五次转换 (5th conversion in regular sequence)

位 19:15 **SQ4[4:0]**：规则序列中的第四次转换 (4th conversion in regular sequence)

位 14:10 **SQ3[4:0]**：规则序列中的第三次转换 (3rd conversion in regular sequence)

位 9:5 **SQ2[4:0]**：规则序列中的第二次转换 (2nd conversion in regular sequence)

位 4:0 **SQ1[4:0]**：规则序列中的第一次转换 (1st conversion in regular sequence)

12.10.10 ADC 注入序列寄存器 (ADC_JSQR)

ADC injected sequence register

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JSQ4[0]	JSQ3[4:0]					JSQ2[4:0]					JSQ1[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:22 保留，必须保持复位值。

位 21:20 **JL[1:0]**：注入序列长度 (Injected sequence length)

通过软件写入这些位可定义注入通道转换序列中的转换总数。

00：1 次转换

01：2 次转换

10：3 次转换

11：4 次转换

位 19:15 **JSQ4[4:0]**：注入序列中的第四次转换 (4th conversion in injected sequence)（当 JL[1:0] = 3 时，请参见下方的注释）

通过软件写入这些位，并将通道编号 (0..18) 分配为序列中的第四次转换。

位 14:10 **JSQ3[4:0]**：注入序列中的第三次转换 (3rd conversion in injected sequence)（当 JL[1:0] = 3 时，请参见下方的注释）

位 9:5 **JSQ2[4:0]**：注入序列中的第二次转换 (2nd conversion in injected sequence)（当 JL[1:0] = 3 时，请参见下方的注释）

位 4:0 **JSQ1[4:0]**：注入序列中的第一次转换 (1st conversion in injected sequence)（当 JL[1:0] = 3 时，请参见下方的注释）

注意： *当 JL[1:0] = 3（定序器中有 4 次注入转换）时，ADC 将按以下顺序转换通道：JSQ1[4:0]、JSQ2[4:0]、JSQ3[4:0] 和 JSQ4[4:0]。*

当 JL = 2（定序器中有 3 次注入转换）时，ADC 将按以下顺序转换通道：JSQ2[4:0]、JSQ3[4:0] 和 JSQ4[4:0]。

当 JL = 1（定序器中有 2 次注入转换）时，ADC 转换通道的顺序为：先是 JSQ3[4:0]，而后是 JSQ4[4:0]。

当 JL = 0（定序器中有 1 次注入转换）时，ADC 将仅转换 JSQ4[4:0] 通道。

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

ADC injected data register x

偏移地址：0x3C - 0x48

复位值：0x0000 0000

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

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

位 31:16 保留，必须保持复位值。

位 15:0 **JDATA[15:0]**：注入数据 (Injected data)

这些位为只读。它们包括来自注入通道 X 的转换结果。数据有左对齐和右对齐两种方式，如*图 38* 和*图 39* 所示。

12.10.12 ADC 规则数据寄存器 (ADC_DR)

ADC regular data register

偏移地址：0x4C

复位值：0x0000 0000

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

位 31:16 保留，必须保持复位值。

位 15:0 **DATA[15:0]**：规则数据 (Regular data)

这些位为只读。它们包括来自规则通道的转换结果。数据有左对齐和右对齐两种方式，如*图 38* 和*图 39* 所示。

12.10.13 ADC 通用状态寄存器 (ADC_CSR)

ADC Common status register

偏移地址：0x00（该偏移地址与 ADC1 基地址 + 0x300 相关）

复位值：0x0000 0000

该寄存器可提供不同 ADC 的状态位图像。但是，它为只读形式且不允许将不同的状态位清零。必须在对应的 ADC_SR 寄存器中将其写为 0，才能将各个状态位清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										OVR3	STRT3	JSTRT3	JEOC 3	EOC3	AWD3
										ADC3					
										r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		OVR2	STRT2	JSTRT 2	JEOC2	EOC2	AWD2	Reserved		OVR1	STRT1	JSTRT1	JEOC 1	EOC1	AWD1
		ADC2								ADC1					
		r	r	r	r	r	r			r	r	r	r	r	r

位 31:22 保留，必须保持复位值。

位 21 **OVR3**：ADC3 的溢出标志 (Overrun flag of ADC3)

该位是 ADC3_SR 寄存器中 OVR 位的副本。

位 20 **STRT3**：ADC3 的规则通道开始标志 (Regular channel Start flag of ADC3)

该位是 ADC3_SR 寄存器中 STRT 位的副本。

位 19 **JSTRT3**：ADC3 的注入通道开始标志 (Injected channel Start flag of ADC3)
该位是 ADC3_SR 寄存器中 JSTRT 位的副本。

位 18 **JEOC3**：ADC3 的注入通道转换结束 (Injected channel end of conversion of ADC3)
该位是 ADC3_SR 寄存器中 JEOC 位的副本。

位 17 **EOC3**：ADC3 的转换结束 (End of conversion of ADC3)
该位是 ADC3_SR 寄存器中 EOC 位的副本。

位 16:14 保留，必须保持复位值。

位 13 **OVR2**：ADC2 的溢出标志 (Overrun flag of ADC2)
该位是 ADC2_SR 寄存器中 OVR 位的副本。

位 12 **STRT2**：ADC2 的规则通道开始标志 (Regular channel Start flag of ADC2)
该位是 ADC2_SR 寄存器中 STRT 位的副本。

位 11 **JSTRT2**：ADC2 的注入通道开始标志 (Injected channel Start flag of ADC2)
该位是 ADC2_SR 寄存器中 JSTRT 位的副本。

位 10 **JEOC2**：ADC2 的注入通道转换结束 (Injected channel end of conversion of ADC2)
该位是 ADC2_SR 寄存器中 JEOC 位的副本。

位 9 **EOC2**：ADC2 转换结束 (End of conversion of ADC2)
该位是 ADC2_SR 寄存器中 EOC 位的副本。

位 8:6 保留，必须保持复位值。

位 5 **OVR1**：ADC1 的溢出标志 (Overrun flag of ADC1)
该位是 ADC1_SR 寄存器中 OVR 位的副本。

位 4 **STRT1**：ADC1 的规则通道开始标志 (Regular channel Start flag of ADC1)
该位是 ADC1_SR 寄存器中 STRT 位的副本。

位 3 **JSTRT1**：ADC1 的注入通道开始标志 (Injected channel Start flag of ADC1)
该位是 ADC1_SR 寄存器中 JSTRT 位的副本。

位 2 **JEOC1**：ADC1 的注入通道转换结束 (Injected channel end of conversion of ADC1)
该位是 ADC1_SR 寄存器中 JEOC 位的副本。

位 1 **EOC1**：ADC1 的转换结束 (End of conversion of ADC1)
该位是 ADC1_SR 寄存器中 EOC 位的副本。

12.10.14 ADC 通用控制寄存器 (ADC_CCR)

ADC common control register

偏移地址：0x04（该偏移地址与 ADC1 基地址 + 0x300 相关）

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMA[1:0]		DDS	Res.	DELAY[3:0]				Reserved			MULTI[4:0]				
rw	rw	rw		rw	rw	rw	rw				rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23

位 22

位 21:18 保留，必须保持复位值。

位 17:16 **ADCPRE**：ADC 预分频器 (ADC prescaler)

由软件置 1 和清零，以选择 ADC 的时钟频率。该时钟为所有 ADC 所共用。 *注意：*

00：PCLK2 2 分频

01：PCLK2 4 分频

10：PCLK2 6 分频

11：PCLK2 8 分频

位 12 保留，必须保持复位值。

位 11:8 **DELAY**：2 个采样阶段之间的延迟 (Delay between 2 sampling phases)

由软件置 1 和清零。这些位在双重或三重交错模式下使用。

0000：5 * T_{ADCCLK}

0001：6 * T_{ADCCLK}

0010：7 * T_{ADCCLK}

...

1111：20 * T_{ADCCLK}

位 7:5 保留，必须保持复位值。

12.10.15 ADC 寄存器映射

下表对 ADC 寄存器进行了汇总。

表 54. ADC 全局寄存器映射

偏移	寄存器
0x000 - 0x04C	ADC1
0x050 - 0x0FC	Reserved
0x100 - 0x14C	Reserved
0x118 - 0x1FC	Reserved
0x200 - 0x24C	Reserved
0x250 - 0x2FC	Reserved
0x300 - 0x308	通用寄存器

表 55. 每个 ADC 的 ADC 寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	ADC_SR	Reserved																										OVR	STRT	JSTRT	JEOC	EOC	AWD
	Reset value																											0	0	0	0	0	0
0x04	ADC_CR1	Reserved					OVRIE	RES[1:0]		AWDEN	JAWDEN	Reserved						DISC NUM [2:0]			JDISCEN	DISCEN	JAUTO	AWD SGL	SCAN	JEOCIE	AWDIE	EOCIE	AWDCH[4:0]					
	Reset value						0	0	0	0	0							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	ADC_CR2	Reserved	SWSTART	EXTEN[1:0]		EXTSEL[3:0]				Reserved	JSWSTART	JEXTEN[1:0]		JEXTSEL[3:0]				Reserved				ALIGN	EOCS	DDS	DMA	Reserved						CONT	ADON
	Reset value		0	0	0	0	0	0	0		0	0	0	0	0	0	0					0	0		0							0	0
0x0C	ADC_SMPR1	Sample PWMe bits SMPx_x																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	ADC_SMPR2	Sample PWMe bits SMPx_x																															

	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	ADC_JOFR1	Reserved																JOFFSET1[11:0]														
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	ADC_JOFR2	Reserved																JOFFSET2[11:0]														
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	ADC_JOFR3	Reserved																JOFFSET3[11:0]														
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 55. 每个 ADC 的 ADC 寄存器映射和复位值 （续）

偏移	寄存器	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
0x20	ADC_JOFR4	Reserved																				JOFFSET4[11:0]										
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x24	ADC_HTR	Reserved																				HT[11:0]										
	Reset value																					1	1	1	1	1	1	1	1	1	1	1	1
0x28	ADC_LTR	Reserved																				LT[11:0]										
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x2C	ADC_SQR1	Reserved								L[3:0]				Regular channel sequence SQx_x bits																		
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	ADC_SQR2	Reserved	Reserved	Regular channel sequence SQx_x bits																													
	Reset value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x34	ADC_SQR3	Reserved	Reserved	Regular channel sequence SQx_x bits																													
	Reset value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	ADC_JSQR	Reserved										JL[1:0]		Injected channel sequence JSQx_x bits																		
	Reset value											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	ADC_JDR1	Reserved																JDATA[15:0]														
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	ADC_JDR2	Reserved																JDATA[15:0]														
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	ADC_JDR3	Reserved																JDATA[15:0]														
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	ADC_JDR4	Reserved																JDATA[15:0]														
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x4C	ADC_DR	Reserved																Regular DATA[15:0]														
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 56. ADC 寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	ADC_CSR	Reserved										OVR	STRT	JSTRT	JEOC	EOC	AWD	Reserved		OVR	STRT	JSTRT	JEOC	EOC	AWD	Reserved		OVR	STRT	JSTRT	JEOC	EOC	AWD
	Reset value											0	0	0	0	0	0			0	0	0	0	0	0			0	0	0	0	0	0
												ADC3								ADC2								ADC1					
0x04	ADC_CCR	Reserved								TSVREFE	VBATE	Reserved				ADCPRE[1:0]		DMA[1:0]	DDS	Reserved		Reserved	DELAY[3:0]					Reserved					
	Reset value									0	0					0	0						0	0	0	0							
0x08	ADC_CDR	Regular DATA2[15:0]																Regular DATA1[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

有关寄存器边界地址的信息，请参见表 2。

13 SPI 串行外设接口及 I2S

13.1SPI 简介

SPI 接口提供两个主要功能，支持 SPI 协议或 I2S 音频协议。默认情况下，选择的是 SPI 功能。可通过软件将接口从 SPI 切换到 I2S。串行外设接口 (SPI) 可与外部器件进行半双工/全双工的同步串行通信。该接口可配置为主模式，在这种情况下，它可为外部从器件提供通信时钟 (SCK)。该接口还能够在多主模式配置下工作。它可用于多种用途，包括基于双线的单工同步传输，其中一条可作为双向数据线，或使用 CRC 校验实现可靠通信。
I2S 也是同步串行通信接口。它可满足四种不同音频标准的要求，包括 I2S Philips 标准、MSB 和 LSB 对齐标准，以及 PCM 标准。它可在全双工模式（使用 4 引脚）或半双工模式（使用 3 个引脚）下作为从器件或主器件工作。当 I2S 配置为通信主模式时，该接口可以向外部从器件提供主时钟。

13.2SPI 和 I2S 主要特性

13.2.1SPI 特性

- 基于三条线的全双工同步传输
- 基于双线的单工同步传输，其中一条可作为双向数据线
- 8 位或 16 位传输帧格式选择
- 主模式或从模式操作
- 多主模式功能
- 8 个主模式波特率预分频器（最大值为 f PCLK /2）
- 从模式频率（最大值为 f PCLK /2）
- 对于主模式和从模式都可实现更快的通信
- 对于主模式和从模式都可通过硬件或软件进行 NSS 管理：动态切换主/从操作
- 可编程的时钟极性和相位
- 可编程的数据顺序，最先移位 MSB 或 LSB
- 可触发中断的专用发送和接收标志
- SPI 总线忙状态标志
- SPI TI 模式
- 用于确保可靠通信的硬件 CRC 功能：

— 在发送模式下可将 CRC 值作为最后一个字节发送
— 根据收到的最后一个字节自动进行 CRC 错误校验

- 可触发中断的主模式故障、上溢和 CRC 错误标志
- 具有 DMA 功能的 1 字节发送和接收缓冲器：发送和接收请求

13.2.2I2S

- 全双工通信
- 半双工通信（仅作为发送器或接收器）
- 主模式或从模式操作
- 8 位可编程线性预分频器，可实现精确的音频采样频率（从 8 kHz 到 192 kHz）

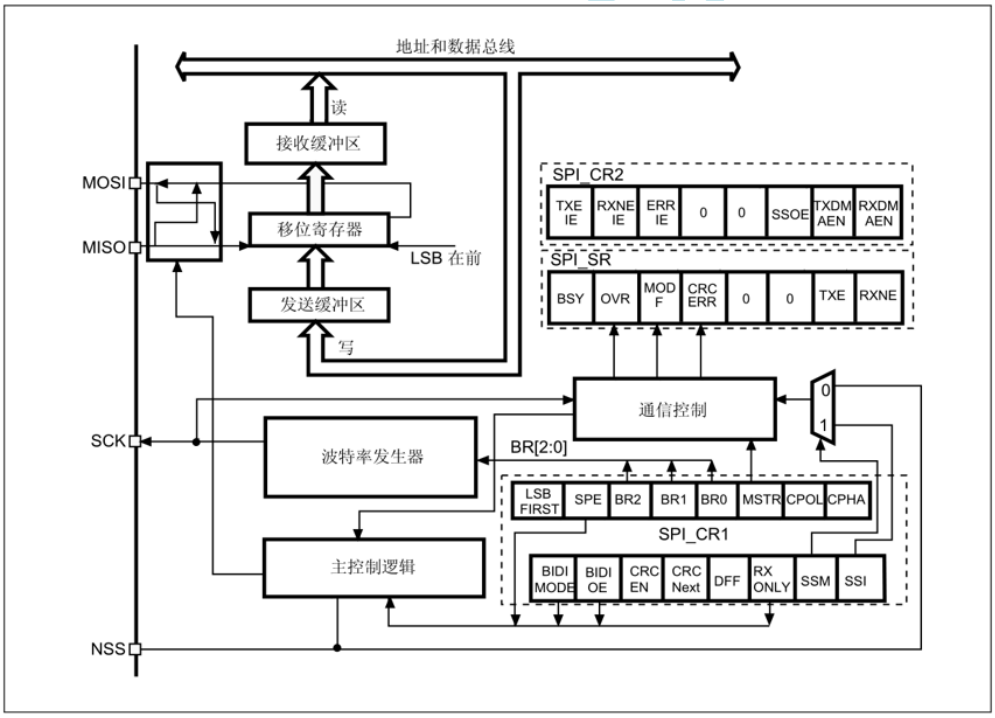
STC MCU

● 数据格式可以是 16 位、24 位或 32 位
● 数据包帧由音频通道固定为 16 位（可容纳 16 位数据帧）或 32 位（可容纳 16 位、24 位、32 位数据帧）
● 可编程的时钟极性（就绪时的电平状态）
● 从发送模式下的下溢标志、接收模式下的上溢标志（主模式和从模式），以及接收和发送模式下的帧错误标志（仅从模式）
● 发送和接收使用同一个 16 位数据寄存器
● 支持的 I 2 S 协议：
— I 2 S Phillps 标准
— MSB 对齐标准（左对齐）
— LSB 对齐标准（右对齐）
— PCM 标准（在 16 位通道帧或扩展为 32 位通道帧的 16 位数据帧上进行短帧和长帧同步）
● 数据方向始终为 MSB 在前
● 用于发送和接收的 DMA 功能（16 位宽）
● 可输出主时钟以驱动外部音频元件。比率固定为 256 × F S （其中 F S 为音频采样频率）

13.3SPI

13.3.1一般说明

SPI 的框图如 图 271 所示。
图 271. SPI

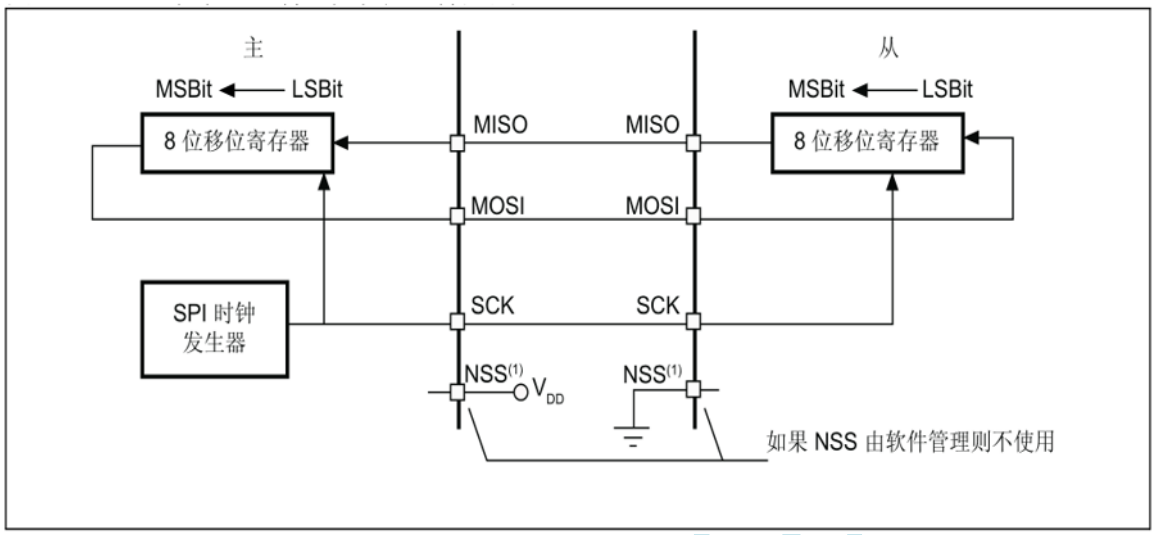


通常，SPI 通过 4 个引脚与外部器件连接：
● MISO：主输入/从输出数据。此引脚可用于在从模式下发送数据和在主模式下接收数据。
● MOSI：主输出/从输入数据。此引脚可用于在主模式下发送数据和在从模式下接收数据。
● SCK：用于 SPI 主器件的串行时钟输出以及 SPI 从器件的串行时钟输入。
● NSS：从器件选择。这是用于选择从器件的可选引脚。此引脚用作“片选”，可让 SPI 主器件与从器件进行单独通信，从而并避免数据线上的竞争。从器件的 NSS 输入可由主器件上的标准 IO 端口驱动。NSS 引脚在使能（SSOE 位）时还可用作输出，并可在

SPI 处于主模式配置时驱动为低电平。通过这种方式，只要器件配置成 NSS 硬件管理模式，所有连接到该主器件 NSS 引脚的其它器件 NSS 引脚都将呈现低电平，并因此而作为从器件。当配置为主模式，且 NSS 配置为输入（MSTR=1 且 SSOE=0）时，如果 NSS 拉至低电平，SPI 将进入主模式故障状态：MSTR 位自动清零，并且器件配置为从模式.

图 272 给出了单个主器件和单个从器件之间的互连的基本示例。

图 272. 单个主器件/ 单个从器件应用



1. 此处，NSS 引脚配置为输入。

MOSI 引脚连接在一起，MISO 引脚连接在一起。通过这种方式，主器件和从器件之间以串行方式传输数据（最高有效位在前）。

通信始终由主器件发起。当主器件通过 MOSI 引脚向从器件发送数据时，从器件同时通过 MISO 引脚做出响应。这是一个数据输出和数据输入都由同一时钟进行同步的 1 全双工通信过程。

从器件选择 (NSS) 引脚管理

可以使用 SPI_CR1 寄存器中的 SSM 位设置硬件或软件管理从器件选择。

- 软件管理 NSS (SSM = 1)

从器件选择信息在内部由 SPI_CR1 寄存器中的 SSI 位的值驱动。外部 NSS 引脚空闲，可供其它应用使用。

- 硬件管理 NSS (SSM = 0)

根据 NSS 输出配置（SPI_CR1 寄存器中的 SSOE 位），硬件管理 NSS 有两种模式。

— NSS 输出使能（SSM = 0，SSOE = 1）

仅当器件在主模式下工作时才使用此配置。当主器件开始通信时，NSS 信号驱动为低电平，并保持到 SPI 被关闭为止。

— NSS 输出禁止（SSM = 0，SSOE = 0）

对于在主模式下工作的器件，此配置允许多主模式功能。对于设置为从模式的器件，NSS 引脚用作传统 NSS 输入：在 NSS 为低电平时片选该从器件，在 NSS 为高电平时取消对它的片选。

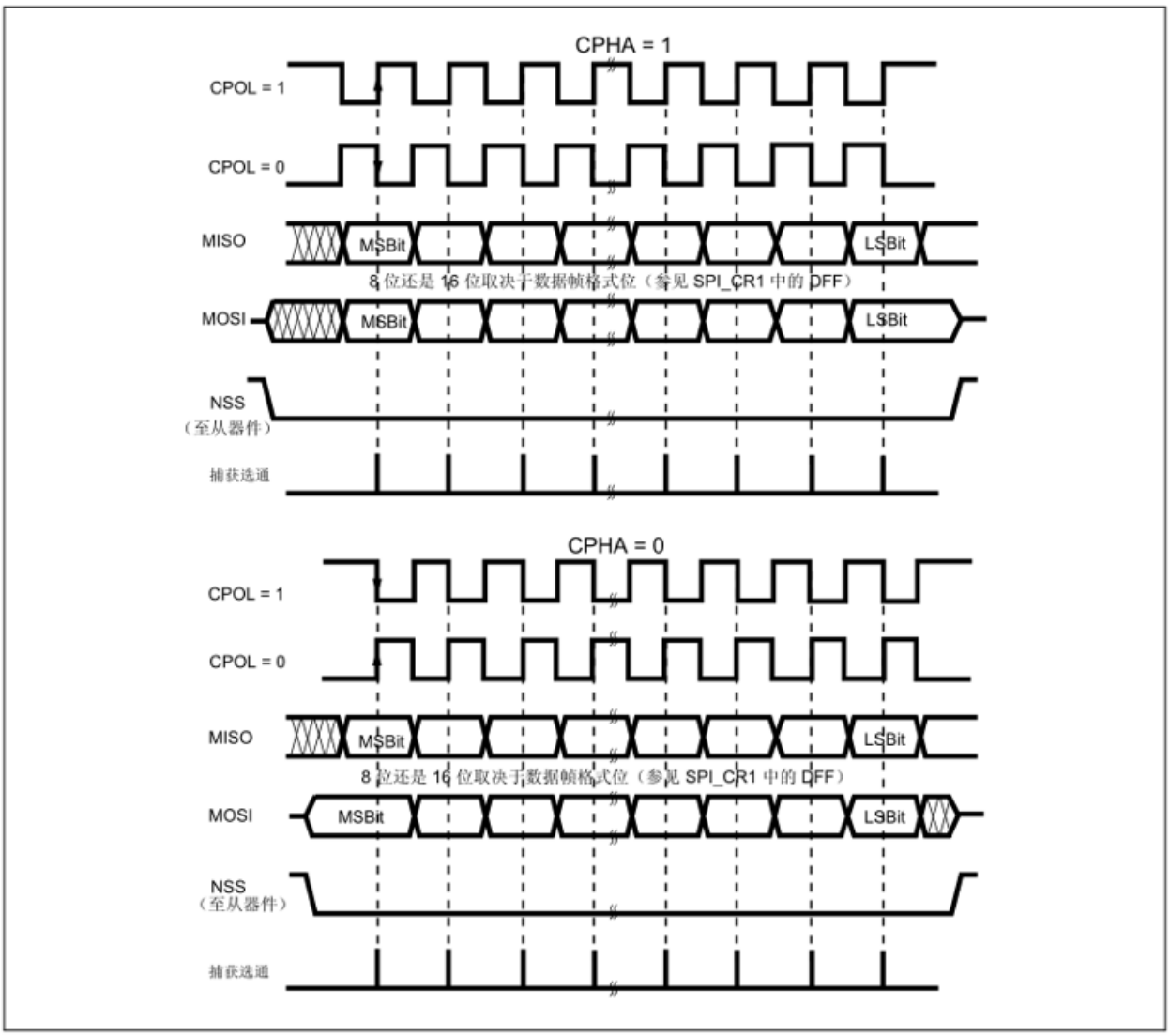
时钟相位和时钟极性

通过 SPI_CR1 寄存器中的 CPOL 和 CPHA 位，可以用软件选择四种可能的时序关系。

CPOL（时钟极性）位控制不传任何数据时的时钟电平状态。此位对主器件和从 器件都有作用。如果复位 CPOL，SCK 引脚在空闲状态处于低电平。如果将 CPOL 置 1，SCK 引脚在空闲状态处于高电平。

如果将 CPHA（时钟相位）位置 1，则 SCK 引脚上的第二个边沿（如果复位 CPOL 位，则为下降沿；如果将 CPOL 位置 1，则为上升沿）对 MSBit 采样。即，在第二个时钟边沿锁存数据。如果复位 CPHA 位，则 SCK 引脚上的第一个边沿（如果将 CPOL 位置 1，则为下降沿；如果复位 CPOL 位，则为上升沿）对 MSBit 采样。即，在第一个时钟边沿锁存数据。

CPOL（时钟极性）和 CPHA（时钟相位）位的组合用于选择数据捕获时钟边沿。
图 273 显示了在 CPHA 和 CPOL 位的四种组合下的 SPI 传输。可以将该图解释为主器件或从器件时序图，其中 SCK 引脚、MISO 引脚、MOSI 引脚直接连接在主器件和从器件之间。
注意： 在切换 CPOL/CPHA 位之前，必须通过复位 SPE 位来关闭 SPI 。
必须以同一时序模式对主器件和从器件进行编程。
SCK 的空闲状态必须与 SPI_CR1 寄存器中选择的极性相对应（如果 CPOL=1 ，则上拉 SCK ；如果 CPOL=0 ，则下拉 SCK ）。通过 SPI_CR1 寄存器中的 DFF 位选择数据帧格式（ 8 或 16 位），该格式决定了发送 / 接收过程中的数据长度。
图 273. 数据时钟时序图



1. 所示的时序为 SPI_CR1 寄存器中的 LSBFIRST 位复位时的时序。

数据帧格式

移出数据时 MSB 在前还是 LSB 在前取决于 SPI_CR1 寄存器中 LSBFIRST 位的值。
每个数据帧的长度均为 8 位或 16 位，具体取决于使用 SPI_CR1 寄存器中的 DFF 位。所选的数据帧格式适用于发送和/或接收。

13.3.2把 SPI 配置成从器件

在从模式配置中，从 SCK 引脚上接收主器件的串行时钟。SPI_CR1 寄存器的 BR[2:0] 位中设置的值不会影响数据传输率。
注意： 建议在主器件发送时钟前使能 SPI 从器件。否则，数据传输可能会不正常。在主时钟的第一个边沿到来之前或者正在进行的通信结束之前，从器件的数据寄存器就需要准备就绪。在使能从器件和主

器件之前，必须将通信时钟的极性设置为空闲时的时钟电平。
按照下面的步骤把 SPI 模式配置成从模式：

步骤

1. 设置 DFF 位，以定义 8 或 16 位数据帧格式。
2. 选择 CPOL 和 CPHA 位，以定义数据传输和串行时钟之间的关系（四种关系中的一种）（参见 图 273）。要实现正确的数据传输，必须以相同方式在从器件和主器件中配置 CPOL 和 CPHA 位。如果通过 SPI_CR2 寄存器中的 FRF 位选择 TI 模式，则不需要此步骤。
3. 帧格式（MSB 在前或 LSB 在前取决于 SPI_CR1 寄存器中 LSBFIRST 位的值）必须与主器件的帧格式相同。如果选择 TI 模式，则不需要此步骤。
4. 在硬件模式下，NSS 引脚在整个字节发送序列期间都必须连接到低电平。在 NSS 软件模式下，将 SPI_CR1 寄存器中的 SSM 位置 1，将 SSI 位清零。如果选择 TI 模式，则不需要此步骤。
5. 将 SPI_CR2 寄存器中的 FRF 位置 1，以选择 TI 模式协议进行串行通信。
6. 将 MSTR 位清零，并将 SPE 位置 1（两个位均在 SPI_CR1 寄存器中）。

在此配置中，MOSI 引脚为数据输入，MISO 引脚为数据输出。

发送序列

数据字节在写周期内被并行加载到发送缓冲区中。
当从器件在其 MOSI 引脚上收到时钟信号和数据的最高有效位时，发送序列开始。其余位（8 位数据帧格式中的 7 个位，16 位数据帧格式中的 15 个位）将加载到移位寄存器中。
SPI_SR 寄存器中的 TXE 标志在数据从发送缓冲区传输到移位寄存器时置 1，并且在 SPI_CR2 寄存器中的 TXEIE 位置 1 时将生成中断。

接收序列

对于接收器，在数据传输完成时：

- 移位寄存器中的数据将传输到接收缓冲区，并且 RXNE 标志（SPI_SR 寄存器）置 1
- 如果 SPI_CR2 寄存器中的 RXNEIE 位置 1，则生成中断。

在出现最后一个采样时钟边沿后，RXNE 位置 1，移位寄存器中接收的数据字节被拷贝到接收缓冲区中。
当读取 SPI_DR 寄存器时，SPI 外设将返回此缓冲值。
通过读取 SPI_DR 寄存器将 RXNE 位清零。

从模式下的 SPI TI 协议

在从模式下，SPI 接口与 TI 协议兼容。可以使用 SPI_CR2 寄存器的 FRF 位来配置从 SPI 串行通信，以兼容此协议。
时钟极性和相位都被强制为遵循 TI 协议，和 SPI_CR1 中的设置无关。NSS 管理也特定于 TI 协议，这使用户无需通过设置 SPI_CR1 和 SPI_CR2 寄存器（例如 SSM、SSI、SSOE）
来对 NSS 管理进行设置。
在从模式下（图 274：TI 模式 — 从模式，单次传输 和 图 275：TI 模式 — 从模式，连续传输 ），使用 SPI 波特率预分频器来控制 MISO 引脚状态切换到高阻态的时刻。可以使用任意波特率，因此可以非常灵活地确定此时刻。但是，波特率通常设置为外部主时钟波特率。MISO 信号变为高阻态的时间 ($t_{release}$) 取决于芯片内部电路同步以及通过 SPI_CR1 寄存器的 BR[2:0] 设置的波特率值。具体公式如下：

$$\frac{t_{baud\_rate}}{2} + 4 \times t_{pclk} < t_{release} < \frac{t_{baud\_rate}}{2} + 6 \times t_{pclk}$$

注意： 此特性不适用于 Motorola SPI 通信（ FRF 位为 0 ）。
要在从器件发送器模式下使用错误中断 (ERRIE = 1) 检测 TI 帧错误，必须通过将 SPI_CR1 寄存器中的 BIDIMODE 和 BIDIOE 置 1 来将 SPI 配置为双线单向模式。当 BIDIMODE 置为 0 时，OVR 将置 1，因为始终不会读取数据寄存器从而始终生成溢出错误中断；而当 BIDIMODE 置 1 时，不会接收数据，也不会将 OVR 置 1。
图 274. TI 模式—从模式，单次传输

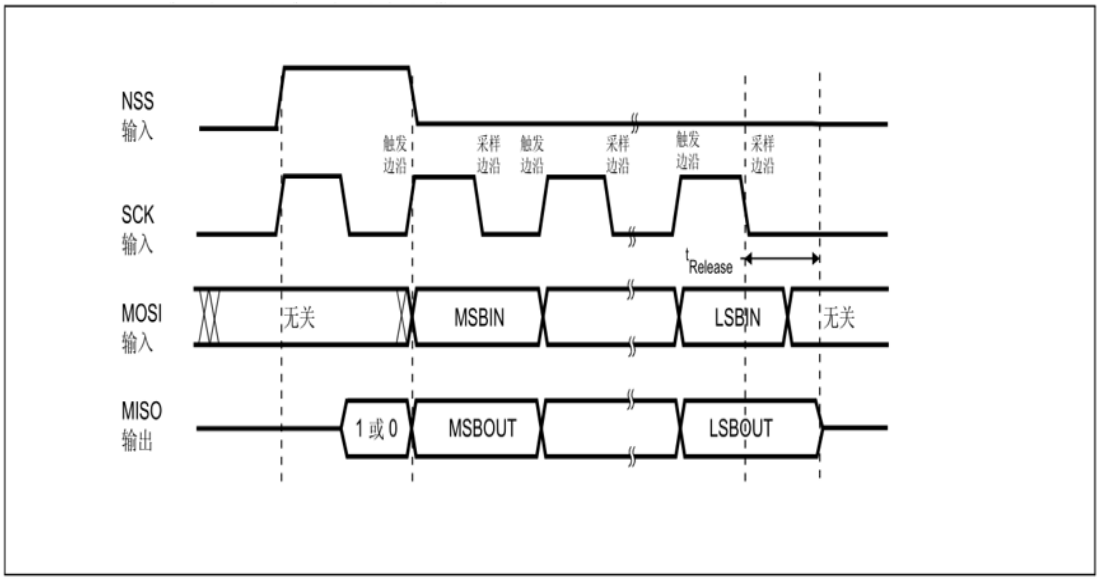
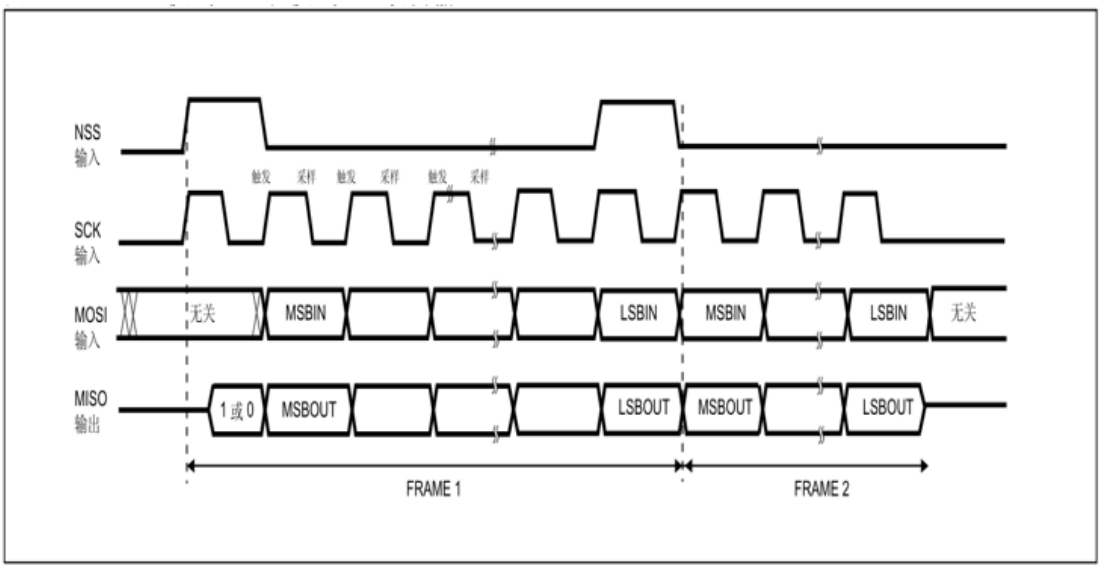


图 275. TI 模式—从模式，连续传输



13.3.3把 SPI 配置成主器件

在主模式配置下，在 SCK 引脚上输出串行时钟。

步骤

1. 设置 BR[2:0] 位以定义串行时钟波特率（参见 SPI_CR1 寄存器）。
2. 选择 CPOL 和 CPHA 位，以定义数据传输和串行时钟之间的关系（四种关系中的一种）（参见 图 273）。如果选择 TI 模式，则不需要此步骤。
3. 设置 DFF 位，以定义 8 或 16 位数据帧格式
4. 配置 SPI_CR1 寄存器中的 LSBFIRST 位以定义帧格式。如果选择 TI 模式，则不需要此步骤。
5. 如果 NSS 引脚配置成输入，在 NSS 硬件模式下，NSS 引脚在整个字节发送序列期间都连接到高电平信号；在 NSS 软件模式下，将 SPI_CR1 寄存器中的 SSM 和 SSI 位置 1。如果 NSS 引脚配置成输出，只应将 SSOE 位置 1。如果选择 TI 模式，则不需要此步骤。
6. 将 SPI_CR2 中的 FRF 位置 1，以选择 TI 协议进行串行通信。
7. MSTR 和 SPE 位必须置 1（仅当 NSS 引脚与高电平信号连接时，这两个位才保持置 1）。在此配置中，MOSI 引脚为数据输出，MISO 引脚为数据输入。

发送序列

在发送缓冲区中写入字节时，发送序列开始。
在第一个位传输期间，数据字节（从内部总线）并行加载到移位寄存器中，然后以串行方式移出到 MOSI 引脚，至于是 MSB 在前还是 LSB 在前则取决于 SPI_CR1 寄存器中的 LSBFIRST 位。TXE 标志在数据从发送缓冲区传输到移位寄存器时置 1，并且在 SPI_CR2 寄存器中的 TXEIE 位置 1 时将生成中断。

接收序列

对于接收器，在数据传输完成时：

- 移位寄存器中的数据将传输到接收缓冲区，并且 RXNE 标志置 1
- 如果 SPI_CR2 寄存器中的 RXNEIE 位置 1，则生成中断

在出现最后一个采样时钟边沿时，RXNE 位置 1，移位寄存器中接收的数据字节被拷贝到接收缓冲区中。当读取 SPI_DR 寄存器时，SPI 外设将返回此缓冲值。
通过读取 SPI_DR 寄存器将 RXNE 位清零。
如果在发送开始后将要发送的下一个数据置于发送缓冲区，则可保持连续的发送流。请注意，仅当 TXE 标志为 1 时，才可以对发送缓冲区执行写操作。
注意： 如果与之通信的从器件需要在每个字节传输之间 拉低片选信号，必须将该主器件的 NSS 配置成 GPIO ，或使用另外 GPIO ，通过软件控制从器件的片选。

主模式下的 SPI TI

在主模式下，SPI 接口与 TI 协议兼容。可以使用 SPI_CR2 寄存器的 FRF 位来配置主 SPI 串行通信，以兼容此协议。
时钟极性和相位都被强制为遵循 TI 协议，和 SPI_CR1 中的设置无关。NSS 管理也特定于 TI 协议，这使用户无需通过设置 SPI_CR1 和 SPI_CR2 寄存器（例如 SSM、SSI、SSOE）来对 NSS 管理进行设置。
图 276 ： TI 模式 — 主模式，单次传输 和 图 277 ： TI 模式 — 主模式，连续传输 ）显示了在主模式下选择 TI 模式时的 SPI 主模式通信波形。

图 276. TI 模式—主模式，单次传输

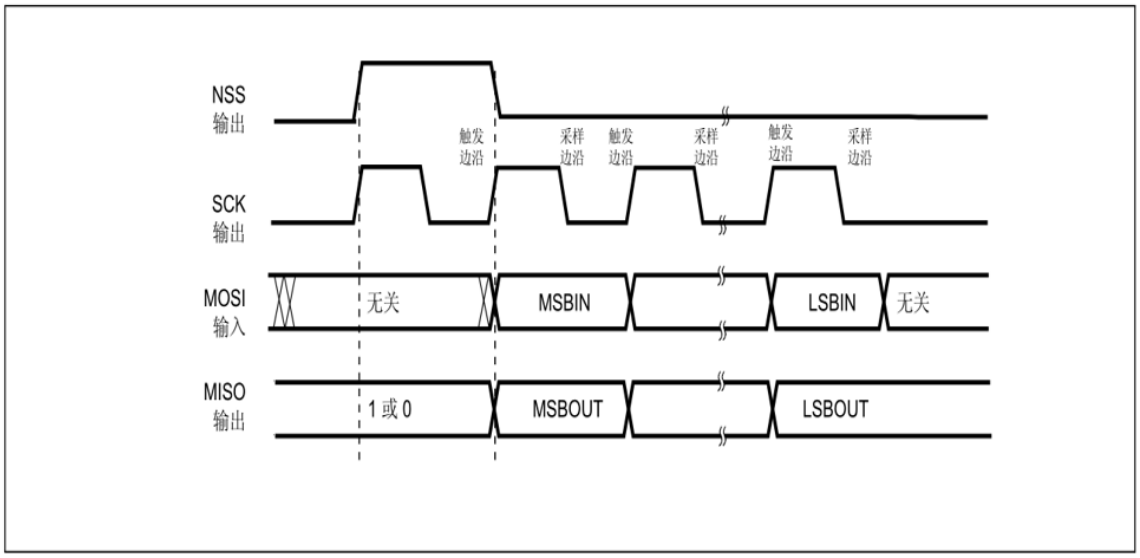
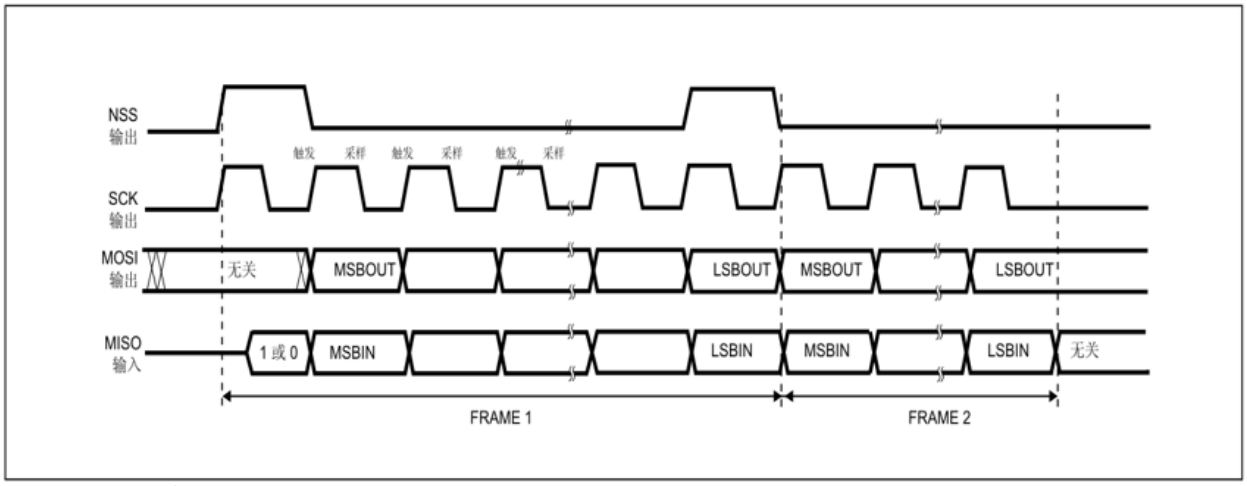


图 277. TI 模式— 主模式，连续传输



13.3.4配置 SPI 进行半双工通信

SPI 能够在以下两种配置中以半双工模式工作。

- 1 个时钟和 1 条双向数据线
- 1 个时钟和 1 条数据线（只接收或只发送）

1 个时钟和 1 条双向数据线 (BIDIMODE=1)

可将 SPI_CR1 寄存器中的 BIDIMODE 位置 1 来使能此模式。在此模式下，SCK 用于时钟，MOSI（主模式下）或 MISO（从模式下）用于数据通信。通过 SPI_CR1 寄存器中的 BIDIOE 位来选择传输方向（输入/输出）。当该位置 1 时，数据线为输出，否则为输入。

1 个时钟和 1 条单向数据线 (BIDIMODE=0)

在此模式下，应用程序可使用 SPI 的只发送或只接收功能。

- 只发送模式类似于全双工模式（BIDIMODE=0、RXONLY=0）：在发送引脚（主模式下的 MOSI 或从模式下的 MISO）上发送数据，接收引脚（主模式下的 MISO 或从模式下的 MOSI）可用作通用 IO。在这种情况下，应用程序只需要忽略接收缓冲区（即使读取数据寄存器，它也不包含接收值）。
- 只接收模式下，应用程序可将 SPI_CR2 寄存器中的 RXONLY 位置 1 来关闭 SPI 输出功能。在这种情况下，发送 IO 引脚（主模式下的 MOSI 或从模式下的 MISO）可用于其它用途。要在只接收模式下开始通信，配置并使能 SPI：
- 在主模式下，通信会立即开始，并在 SPE 位清零且当前接收结束时停止。在此模式下无需读取 BSY 标志。在进行 SPI 通信时，该标志始终置 1。
- 在从模式下，只要 NSS 被拉低（或在 NSS 软件模式下将 SSI 位清零）并且一直有来自主器件的 SCK 输入，SPI 就会继续接收。

13.3.5数据发送和接收过程

接收和发送缓冲区

在接收过程中，数据收到后，先存储到内部接收缓冲区中；而在发送过程中，先将数据存储到内部发送缓冲区中，然后发送数据。

对 SPI_DR 寄存器的读访问将返回接收缓冲值，而对 SPI_DR 寄存器的写访问会将写入的数据存储到发送缓冲区中。

在主模式下启动通信序列

- 在全双工模式下（BIDIMODE=0 且 RXONLY=0）

— 将数据写入到 SPI_DR 寄存器（发送缓冲区）时，通信序列启动。

— 随后在第一个位的发送期间，将数据从发送缓冲区并行加载到 8 位移位寄存器中，然后以串行方式将其移出到 MOSI 引脚。

— 同时，将 MISO 引脚上接收的数据以串行方式移入 8 位移位寄存器，然后并行加载到 SPI_DR 寄存器（接收缓冲区）中。

- 在单向只接收模式下（BIDIMODE=0 且 RXONLY=1）

— 只要 SPE = 1，通信序列就立即开始。
— 只有接收器激活，并且在 MISO 引脚上接收的数据以串行方式移入 8 位移位寄存器，然后并行加载到 SPI_DR 寄存器（接收缓冲区）中。
● 在双向模式下，进行发送时（BIDIMODE=1 且 BIDIOE=1）
— 将数据写入到 SPI_DR 寄存器（发送缓冲区）时，通信序列启动。
— 随后在第一个位的发送期间，将数据从发送缓冲区并行加载到 8 位移位寄存器中，然后以串行方式将其移出到 MOSI 引脚。
— 不接收任何数据。
● 在双向模式下，进行接收时（BIDIMODE=1 且 BIDIOE=0）
— 只要 SPE=1 且 BIDIOE=0，通信序列就立即开始。
— 在 MOSI 引脚上接收的数据以串行方式移入 8 位移位寄存器，然后并行加载到 SPI_DR 寄存器 （接收缓冲区）中。
— 发送器没有激活，因此不会有数据以串行方式移出 MOSI 引脚。

在从模式下启动通信序列

● 在全双工模式下（BIDIMODE=0 且 RXONLY=0）
— 当从器件在其 MOSI 引脚上收到时钟信号和数据的第一个位时，通信序列开始。其余 7 个位将加载到移位寄存器中。
— 同时，在第一个位的发送期间，将数据从发送缓冲区并行加载到 8 位移位寄存器中，然后以串行方式将其移出到 MISO 引脚。在 SPI 主器件启动传输前，软件必须已把要从器件发送的数据写入发送缓冲区。

● 在单向只接收模式下（BIDIMODE=0 且 RXONLY=1）
— 当从器件在其 MOSI 引脚上收到时钟信号和数据的第一个位时，通信序列开始。其余 7 个位将加载到移位寄存器中。
— 发送器没有激活，因此不会有数据以串行方式移出 MISO 引脚。

● 在双向模式下，进行发送时（BIDIMODE=1 且 BIDIOE=1）
— 当从器件收到时钟信号，并将发送缓冲区中的第一个位在 MISO 引脚上发送时，通信序列开始。
— 随后在第一个位的发送期间，将数据从发送缓冲区并行加载到 8 位移位寄存器中，然后以串行方式将其移出到 MISO 引脚。在 SPI 主器件启动传输前，软件必须已把要从器件发送的数据写入发送缓冲区。
— 不接收任何数据。
● 在双向模式下，进行接收时（BIDIMODE=1 且 BIDIOE=0）
— 当从器件在其 MISO 引脚上收到时钟信号和数据的第一个位时，通信序列开始。
— 在 MISO 引脚上接收的数据以串行方式移入 8 位移位寄存器，然后并行加载到 SPI_DR 寄存器 （接收缓冲区）中。
— 发送器没有激活，因此不会有数据以串行方式移出 MISO 引脚。

处理数据发送与接收

将数据从发送缓冲区传输到移位寄存器时，TXE 标志（发送缓冲区为空）置 1。该标志表示内部发送缓冲区已准备好加载接下来的数据。如果 SPI_CR2 寄存器中的 TXEIE 位置 1，可产生中断。通过对 SPI_DR 寄存器执行写操作将 TXE 位清零。

注意： 软件必须确保在尝试写入发送缓冲区之前 TXE 标志已置 1 。否则，将覆盖之前写入发送缓冲区的数据。将数据从移位寄存器传输到接收缓冲区时，RXNE 标志（接收缓冲区非空）会在最后一个采样时钟边沿置 1。它表示已准备好从 SPI_DR 寄存器中读取数据。如果 SPI_CR2 寄存器中的 RXNEIE 位置 1，可产生中断。通过读取 SPI_DR 寄存器将 RXNE 位清零。对于某些配置，可以在最后一次数据传输期间使用 BSY 标志来等待传输完成。

主模式或从模式下的全双工发送和接收过程（BIDIMODE=0 且 RXONLY=0 ）

软件必须遵循以下步骤来发送和接收数据（参见 图 278 和 图 279）：

1. 通过将 SPE 位置 1 来使能 SPI。
2. 将第一个要发送的数据项写入 SPI_DR 寄存器（此操作会将 TXE 标志清零）。
3. 等待至 TXE=1，然后写入要发送的第二个数据项。然后等待至 RXNE=1，读取 SPI_DR 以获取收到的第一个数据项（此操作会将 RXNE 位清零）。对每个要发送/接收的数据项重复此操作，直到

第 n-1 个接收的数据为止。

4. 等待至 RXNE=1，然后读取最后接收的数据。
5. 等待至 TXE=1，然后等待至 BSY=0，再关闭 SPI。

此外，还可以使用在 RXNE 或 TXE 标志所产生的中断对应的各个中断子程序来实现该过程。

图 278. 主/ 全双工模式（BIDIMODE=0 且 RXONLY=0 ）下的 TXE/RXNE/BSY 行为（在连续传输的情况下）

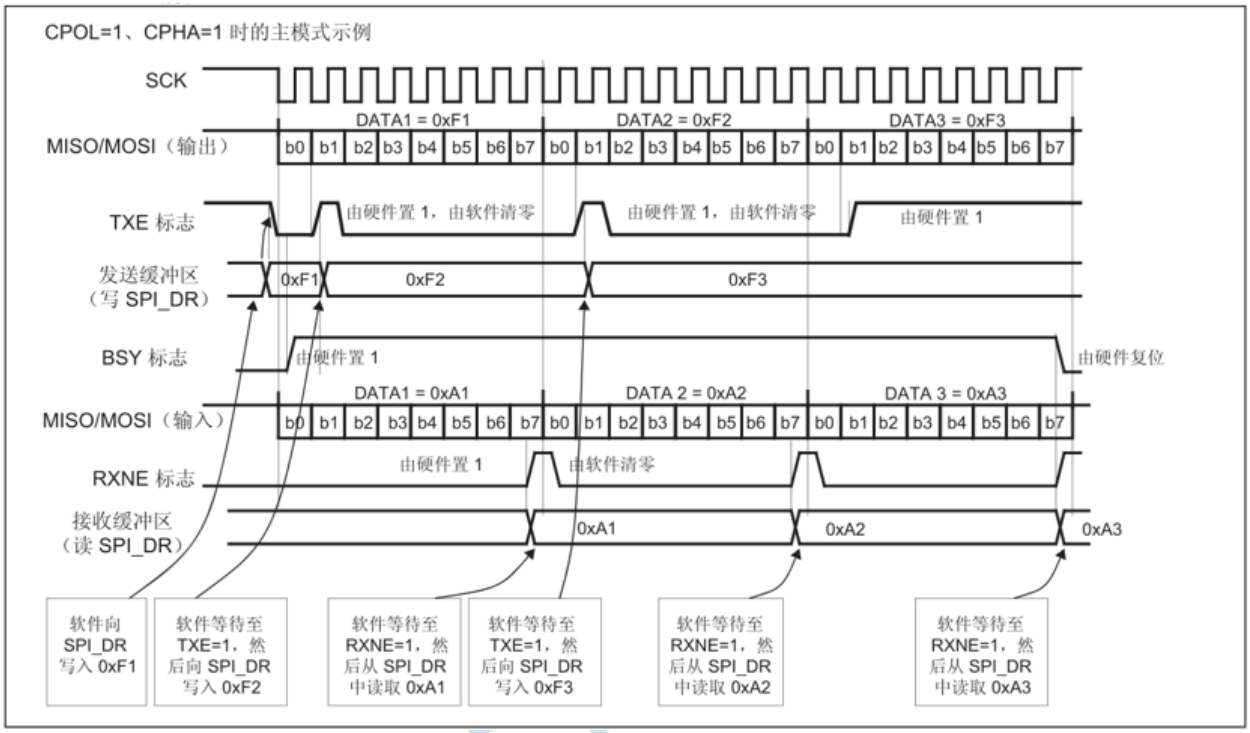
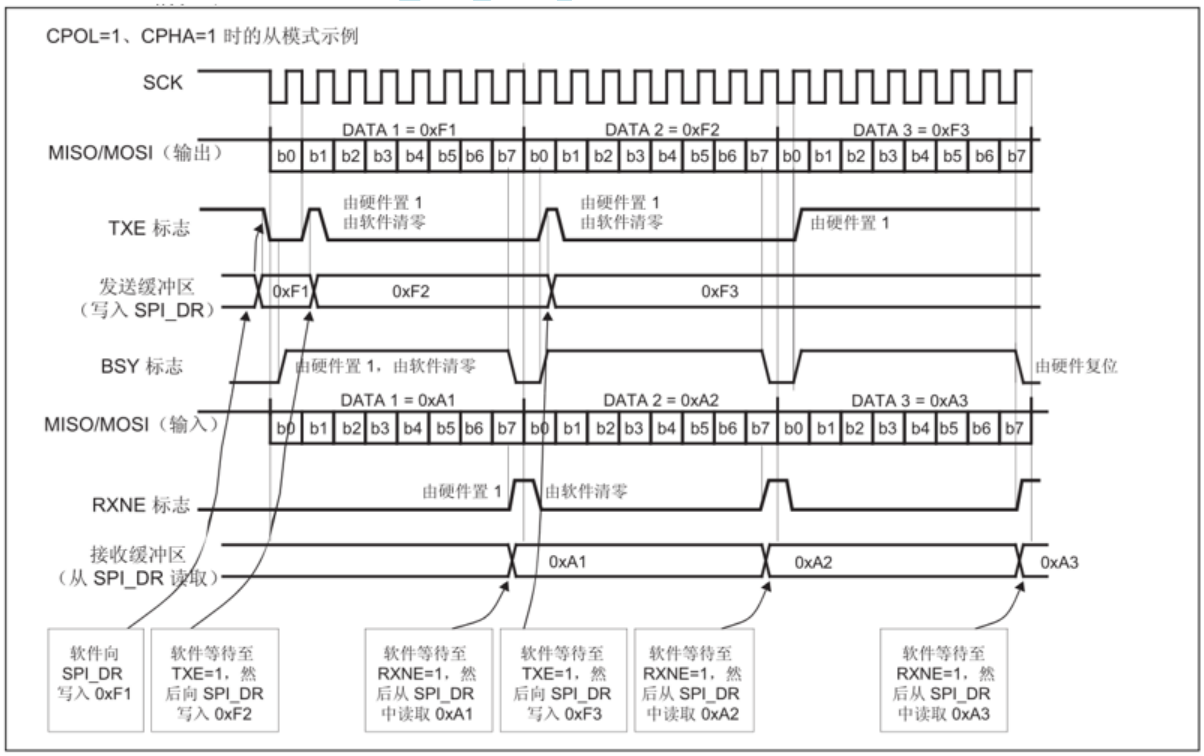


图 279. 主/ 全双工模式（BIDIMODE=0 且 RXONLY=0 ）下的 TXE/RXNE/BSY 行为（在连续传输的情况下）



只发送模式下的数据发送过程（BIDIMODE=0 、RXONLY=0 ）

在此模式下，可以按下文所述简化过程，并且可使用 BSY 位等待发送完成（参见 图 280 和图 281）。

1. 通过将 SPE 位置 1 来使能 SPI。
2. 将第一个要发送的数据项写入 SPI_DR 寄存器（此操作会将 TXE 标志清零）。
3. 等待至 TXE=1，然后写入下一个要发送的数据项。对每个要发送的数据项重复此步骤。
4. 将最后一个数据项写入 SPI_DR 寄存器后，等待至 TXE=1，然后等待至 BSY=0，这表示最后的数据发送完成。此外，还可以使用在 TXE 标志所产生的中断对应的中断子程序来实现该过程。

注意： 在不连续通信期间，在对 SPI_DR 执行写操作与 BSY 位置 1 之间有 2 个 APB 时钟周期的延迟。因此，在只发送模式下，写入最后的数据后，必须先等待 TXE 位置 1，然后等待 BSY 位清零。在只发送模式下，发送两个数据项后， SPI_SR 寄存器中的 OVR 标志将置 1，因为始终不会读取接收的数据。

图 280. 主设备只发送模式（BIDIMODE=0 且 RXONLY=0 ）下的 TXE/BSY 行为（在连续传输的情况下）

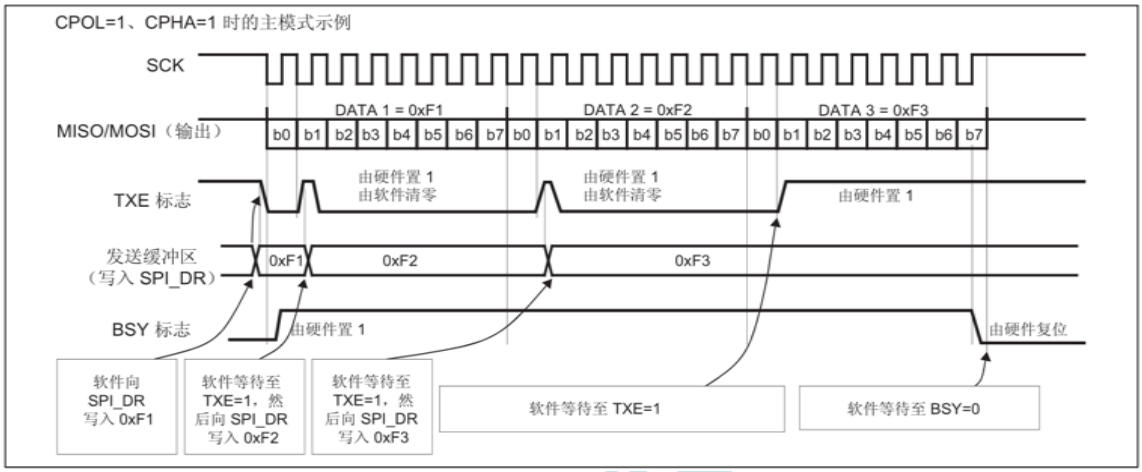
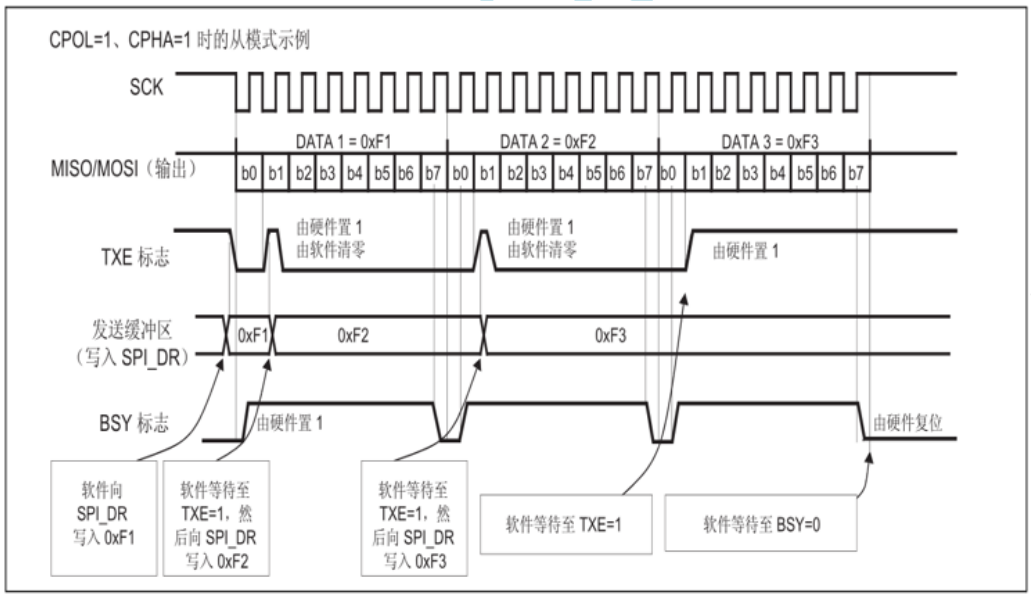


图 281. 从设备只发送（BIDIMODE=0 且 RXONLY=0 ）下的 TXE/BSY 行为（在连续传输的情况下）



单线双向模式下的发送过程（BIDIMODE=1 且 BIDIOE=1 ）

在此模式下，过程与只发送模式下的过程相似，除了在使能 SPI 前必须将 SPI_CR2 寄存器中的 BIDIMODE 位和 BIDIOE 位均置 1。

单向只接收过程（BIDIMODE=0 且 RXONLY=1 ）

在此模式下，可以按如下所述简化过程（参见 图 282）：

1. 将 SPI_CR2 寄存器中的 RXONLY 位置 1。
2. 通过将 SPE 位置 1 使能 SPI：
   a) 在主模式下，这会立即激活 SCK 时钟的产生，并以串行方式接收数据，直到关闭 SPI

(SPE=0)。

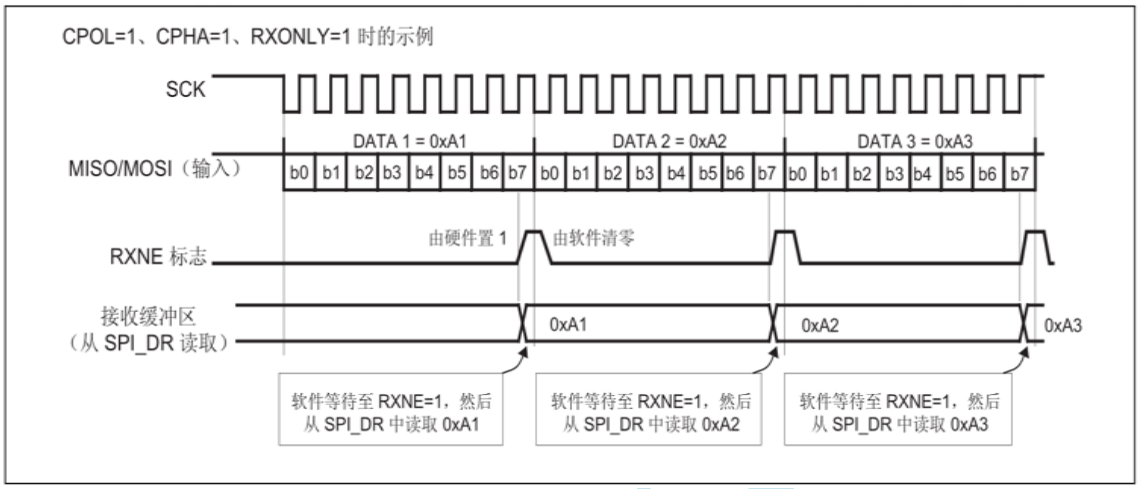
b) 在从模式下，当 SPI 主器件将该从器件的 NSS 驱动为低电平并输出 SCK 时钟时，接收数据。

3. 等待 RXNE=1，然后读取 SPI_DR 寄存器以获取接收的数据（此操作会将 RXNE 位清零）。对每个要接收的数据项重复此操作。

此外，还可以使用在 RXNE 标志所产生的中断对应的中断子程序来实现该过程。

注意： 如果需要在最后一次传输后关闭 SPI ，请采纳第 14.3.8 节：关闭 SPI 中所述的建议。

图 282. 只接收模式（BIDIRMODE=0 且 RXONLY=1 ）下的 RXNE



单线双向模式下的接收过程（BIDIMODE=1 和 BIDIOE=0 ）

在此模式下，过程与只接收模式的过程相似，除了在使能 SPI 前必须将 SPI_CR2 寄存器中的 BIDIMODE 位置 1 并将 BIDIOE 清零。

连续传输和间断传输

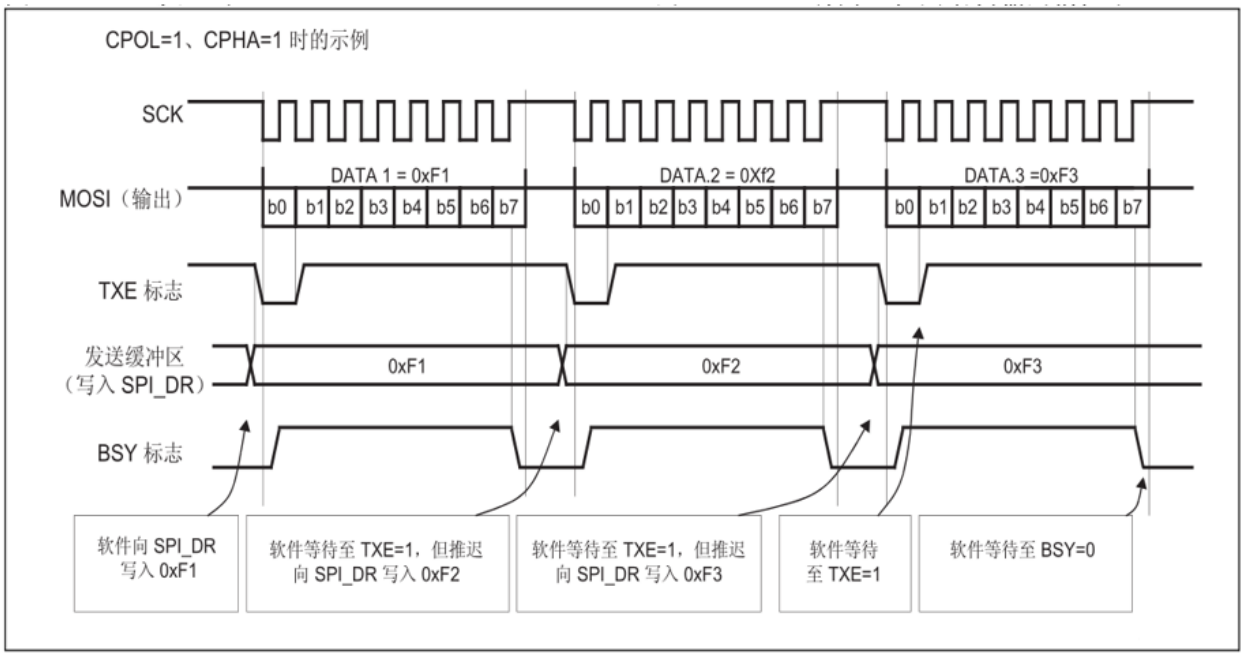
在主模式下发送数据时，如果软件速度快到能在下一个数据传输完成之前响应该数据从数据寄存器传输到移位寄存器所触发的 TXE 中断，并能立即完成再下一个数据的写 SPI_DR 操作，则这种通信称为连续通信。在这种情况下，各数据项之间 SPI 时钟的生成不会间断，并且各数据传输之间不会清零 BSY 位。

相反，如果软件速度不够快，则可能导致通信中断。在这种情况下，各数据传输之间会清零 BSY 位（参见 图 283）。

在主设备仅接收模式 (RXONLY=1) 下，通信始终是连续的，且 BSY 标志始终读为 1。

在从模式下，通信的连续性由 SPI 主器件决定。任何情况下（即使通信是连续的），在各个数据传输之间 BSY 标志都会短暂变为低电平，持续时间为一个 SPI 时钟周期（参见图 281）。

图 283. 发送时（BIDIRMODE=0 且 RXONLY=0 ）的 TXE/BSY 行为（在间断传输的情况下）



13.3.6 CRC 计算

为确保通信的可靠性，SPI 模块实现了硬件 CRC 功能。针对发送的数据和接收的数据分别实现 CRC 计算。使用可编程的多项式对每个位来计算 CRC。在由 SPI_CR1 寄存器中的 CPHA 位和 CPOL 位定义的采样时钟边沿采样每个位来进行计算。

注意：该 SPI 模块提供两种 CRC 计算标准，具体取决于为发送和 / 或接收选择的数据帧格式：8 位数据采用 CR8 ， 16 位数据采用 CRC16 。

通过将 SPI_CR1 寄存器中的 CRCEN 位置 1 来使能 CRC 的计算。此操作将复位 CRC 寄存器（SPI_RXCRCR 和 SPI_TXCRCR）。在全双工或只发送模式下，如果传输由软件（CPU 模式）管理，则在将最后传输的数据写入 SPI_DR 后，必须立即对 CRCNEXT 位执行写操作。最后一次数据传输结束时，将发送 SPI_TXCRCR 值。

在只接收模式下，如果传输由软件（CPU 模式）管理，则在接收到倒数第二个数据后，必须对 CRCNEXT 位执行写操作。在收到最后一个数据后会收到 CRC，然后执行 CRC 校验。如果传输过程中出现数据损坏，则在数据和 CRC 传输结束时，SPI_SR 寄存器中的 CRCERR 标志将置 1。

如果发送缓冲区中存在数据，则只有在发送数据字节后才会发送 CRC 值。在 CRC 发送期间，CRC 计算器处于关闭状态且寄存器值保持不变。

可通过以下步骤使用 CRC 进行 SPI 通信：

1. 对 CPOL、CPHA、LSBFirst、BR、SSM、SSI 和 MSTR 值进行编程。
2. 对 SPI_CRCPR 寄存器中的多项式进行编程。
3. 通过将 SPI_CR1 寄存器中的 CRCEN 位置 1 来使能 CRC 计算。此操作还会将 SPI_RXCRCR 和 SPI_TXCRCR 寄存器清零。
4. 通过将 SPI_CR1 寄存器中的 SPE 位置 1 使能 SPI。
5. 启动并维持通信，直到只剩下一个字节或半字未发送或接收。

— 在全双工或只发送模式下，如果传输由软件管理，则在向发送缓冲区写入最后一个字节或半字后，将 SPI_CR1 寄存器中的 CRCNEXT 位置 1，以表示在发送完最后一个字节后将发送 CRC。

— 在只接收模式下，在接收倒数第二个数据后，立即将 CRCNEXT 位置 1，以便使 SPI 准备好在接收完最后一个数据后进入 CRC 阶段。在 CRC 传输期间，CRC 计算将冻结。

6. 传输完最后一个字节或半字后，SPI 进入 CRC 传输和校验阶段。在全双工模式或只接收模式下，将接收的 CRC 与 SPI_RXCRCR 值进行比较。如果两个值不匹配，则 SPI_SR 中的 CRCERR 标志将置 1，并且在 SPI_CR2 寄存器中的 ERRIE 位置 1 时会产生中断。

注意： 当 SPI 处于从模式时，注意只能在时钟稳定（即，时钟处于空闲电平）时使能 CRC 计算。否则，可能导致 CRC 计算错误。因为，只要 CRCEN 位置 1 ，无论 SPE 位的值如何，只要有时钟输入， CRC 计算器就开始工作。在 SPI 通信时钟频率较高的情况下，发送 CRC 时务必小心。由于在 CRC 传输阶段 CPU 应尽可能空闲，因此禁止在 CRC 发送阶段调用函数，以便避免最后的数据和 CRC 接收出错。

实际上，在发送 / 接收最后的数据之前必须对 CRCNEXT 位执行写操作。

SPI 通信时钟频率较高时，建议使用 DMA 模式来避免由于 CPU 访问影响 SPI 带宽而导致 SPI 速度性能下降。

如果将器件配置为从器件，并且使用 NSS 硬件模式，则需要在数据阶段和 CRC 阶段之间将 NSS 引脚保持为低电平。

当 SPI 配置为从模式并且 CRC 功能已使能时，即使 NSS 引脚上为高电平，也会进行 CRC 计算。例如，在多从模式环境下可能出现这种情况，此时通信主器件会交替寻址从器件。

在对从器件片选的切换期间内，应在主器件和从器件两端同时将 CRC 值清零，以重新同步主从双方的 CRC 计算。

要将 CRC 清零，请按以下步骤操作：

1. 关闭 SPI (SPE = 0)
2. 将 CRCEN 位清零
3. 将 CRCEN 位置 1
4. 使能 SPI (SPE = 1)

13.3.7状态标志

应用可通过三种状态标志监视 SPI 总线的状态。

发送缓冲区为空 (TXE)

此标志置 1 时，表示发送缓冲区为空，可以将待发送的下一个数据加载到缓冲区中。对 SPI_DR 寄存器执行写操作时，将清零 TXE 标志。

接收缓冲区非空 (RXNE)

此标志置 1 时，表示接收缓冲区中存在有效的已接收数据。读取 SPI_DR 时，将清零该标志。

BUSY

BSY 标志由硬件置 1 和清零（对此标志执行写操作没有任何作用）。

BSY 标志用于指示 SPI 通信的状态。

BSY 置 1 时，表示 SPI 正忙于通信。在主模式下的双向通信接收模式（MSTR=1 且 BDM=1 且 BDOE=0）有一个例外情况，BSY 标志在接收过程中保持低电平。

如果软件要关闭 SPI 并进入停止模式（或关闭外设时钟），可使用 BSY 标志检测传输是否结束以避免破坏最后一个数据的传输。为此，必须严格遵循下述步骤。

BSY 标志还可用于避免在多主模式系统中发生写冲突。

传输开始时，BSY 标志将置 1，但在主模式下的双向通信接收模式（MSTR=1 且 BDM=1 且 BDOE=0）下例外。

在以下情况硬件将清零该标志：

- 传输完成时（主模式下的连续通信除外）
- 关闭 SPI 时
- 发生主模式故障时 (MODF=1)

当通信不连续时，BSY 标志在各通信之间处于低电平。

当通信连续时：

- 在主模式下，BSY 标志在所有传输期间均保持高电平
- 在从模式下，BSY 标志在各传输之间的一个 SPI 时钟周期内变为低电平

注意： 请勿使用 BSY 标志处理每次数据发送或接收，最好改用 TXE 标志和 RXNE 标志。

13.3.8关闭 SPI

传输终止时，应用可通过关闭 SPI 外设来停止通信。这通过将 SPE 位清零来完成。对于某些配置，在传输进行时关闭 SPI 并进入停止模式会导致当前传输受损，并且/或者

BSY 标志可能不可靠。
为避免上述后果，建议在关闭 SPI 时按以下步骤操作：
在主模式或全双工从模式（BIDIMODE=0 、RXONLY=0 ）下
1. 等待 RXNE=1 以接收最后的数据
2. 等待 TXE=1
3. 然后等待 BSY=0
4. 关闭 SPI (SPE=0)，最后进入停止模式（或关闭外设时钟）

在主模式或单向只发送从模式（BIDIMODE=0 、RXONLY=0）或双向通信发送模式（BIDIMODE=1 、BIDIOE=1 ）下
在最后的数据写入 SPI_DR 寄存器后：
1. 等待 TXE=1
2. 然后等待 BSY=0
3. 关闭 SPI (SPE=0)，最后进入停止模式（或关闭外设时钟）

在单向只接收主模式（MSTR=1 、BIDIMODE=0 、RXONLY=1）或双向通信接收模式（MSTR=1 、BIDIMODE=1 、BIDIOE=0 ）下
必须以特殊方式管理这种情况，以避免多余的 SPI 数据传输。以下序列仅适用于 SPI Motorola 配置（FRF 位置 0）：
1. 等待倒数第二个数据（第 n-1 个）对应的 RxNE 标志置位
2. 然后等待一个 SPI 时钟周期（使用软件循环），才能关闭 SPI (SPE=0)
3. 再等待最后的 RXNE=1，然后进入停止模式（或关闭外设时钟）

当 SPI 配置为 TI 模式（FRF 位置 1）时，必须按以下步骤操作以避免当关闭 SPI 时在 NSS 上产生不需要的脉冲：
1. 等待倒数第二个数据（第 n-1 个）对应的 RxNE 标志置位
2. 在以下窗口帧中使用软件循环关闭 SPI (SPE = 0)：

— 至少一个 SPI 时钟周期后，
— LSB 数据开始传输前。

注意： 在双向通信接收主模式（ MSTR=1 、 BDM=1 、 BDOE=0 ）下， BSY 标志在传输期间保持低电平。

在只接收从模式（MSTR=0 、BIDIMODE=0 、RXONLY=1）或双向通信接收模式（MSTR=0 、BIDIMODE=1 、BIDOE=0 ）下
1. 可以随时关闭 SPI（写入 SPE=0）：当前传输完成后，SPI 才被真正关闭
2. 之后，如果要进入停止模式，则必须首先等待至 BSY = 0，然后才能进入停止模式（或关闭外设时钟）。

13.3.9使用 DMA （直接存储器寻址）进行 SPI 通信

要以最大速度工作，需要给 SPI 不断提供要发送的数据，并及时读取接收缓冲区中的数据，以避免上溢。为加速传输，SPI 提供了 DMA 功能，以实现简单的请求/应答协议。

当使能 SPI_CR2 寄存器中的使能位时，将请求 DMA 访问。发送缓冲区和接收缓冲区会发出各自的 DMA 请求（参见 图 284 和 图 285）：

- 在发送过程中，每次 TXE 位置 1 都会发出 DMA 请求。DMA 随后对 SPI_DR 寄存器执行写操作（此操作会将 TXE 标志清零）。
- 在接收过程中，每次 RXNE 位置 1 都会发出 DMA 请求。DMA 随后对 SPI_DR 寄存器执行读操作（此操作会将 RXNE 标志清零）。

当 SPI 仅用于发送数据时，可以只使能 SPI Tx DMA 通道。在这种情况下，OVE 标志会置 1，因为未读取接收的数据。

当 SPI 仅用于接收数据时，可以只使能 SPI Rx DMA 通道。

在发送模式下，DMA 完成了所有要发送数据的传输（DMA_ISR 寄存器中的 TCIF 标志置 1）后，可以对 BSY 标志进行监视，以确保 SPI 通信已完成。在关闭 SPI 或进入停止模式前必须执行此步骤，以避免损坏最后一次数据的发送。软件必须首先等待 TXE=1，再等待 BSY=0。

注意：在不连续通信期间，在对 SPI_DR 执行写操作与 BSY 位置 1 之间有 2 个 APB 时钟周期的延迟。因此，必须在写入最后的数据后，先等待 TXE=1 ，再等待 BSY=0 。

图 284. 使用 DMA 进行发送

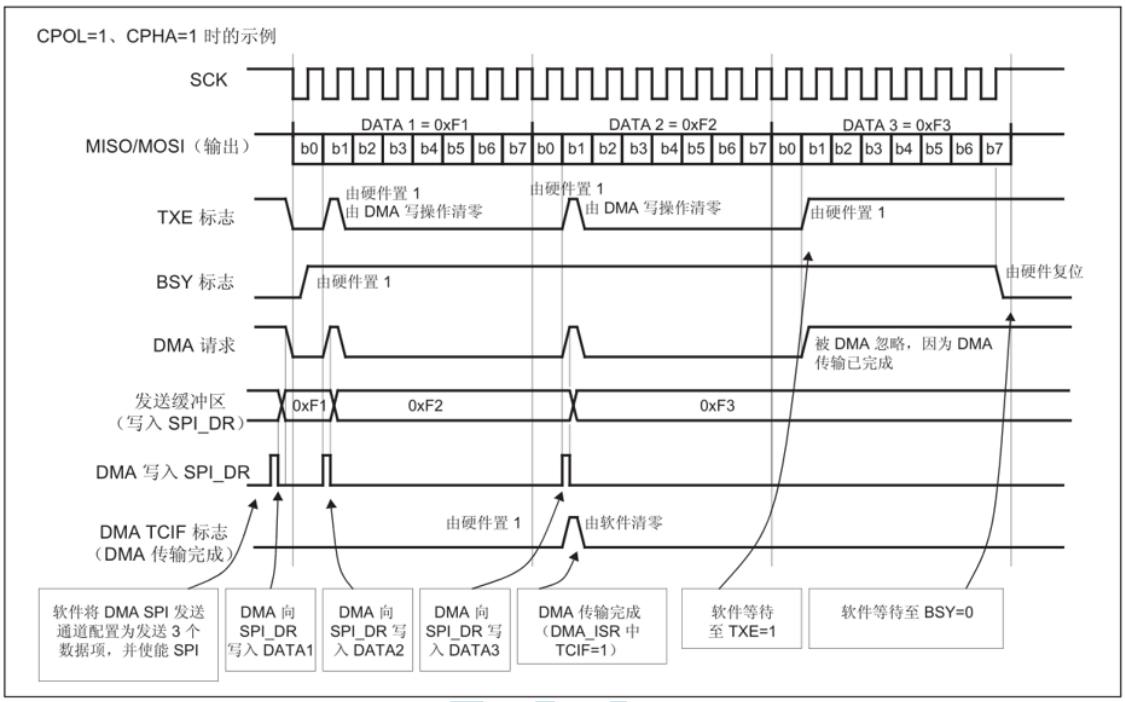
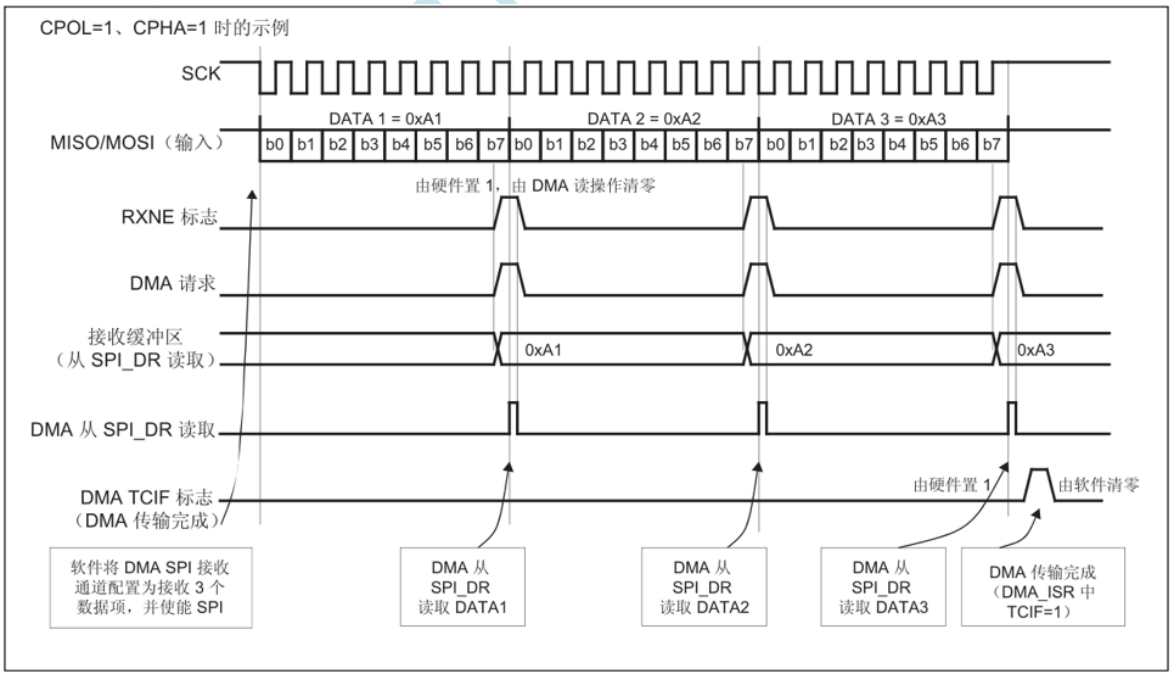


图 285. 使用 DMA



DMA 功能与 CRC

当使能的 SPI 通信支持 CRC 通信和 DMA 模式时，在通信结束时会自动发送和接收 CRC，无需使用 CRCNEXT 位。接收 CRC 后，必须在 SPI_DR 寄存器中读取 CRC，以将 RXNE 标志清零。
如果传输过程中出现损坏，则在数据和 CRC 传输结束时，SPI_SR 寄存器中的 CRCERR 标志将置 1。

13.3.10错误标志

主模式故障 (MODF)

当主器件的 NSS 引脚拉低（NSS 硬件模式下）或 SSI 位为 0（NSS 软件模式下）时，会发生主模式故障，这会自动将 MODF 位置 1。主模式故障会在以下几方面影响 SPI 外设：

- 如果 ERRIE 位置 1，MODF 位将置 1，并生成 SPI 中断。
- SPE 位清零。这将关闭器件的所有输出，并关闭 SPI 接口。
- MSTR 位清零，从而强制器件进入从模式。

使用以下软件序列将 MODF 位清零：

1. 在 MODF 位置 1 时，对 SPI_SR 寄存器执行读或写访问。
2. 然后，对 SPI_CR1 寄存器执行写操作。

为避免包含多个 MCU 的系统中发生多从模式冲突，必须在 MODF 位清零序列期间将 NSS 引脚拉高。在该清零序列后，可以将 SPE 和 MSTR 位恢复到原始状态。
作为安全措施，硬件不允许在 MODF 位置 1 时将 SPE 和 MSTR 位置 1。
在从器件中，不能将 MODF 位置 1。但是，在多主模式配置中，器件可在 MODF 位置 1 时处于从模式。在这种情况下，MODF 位指示系统控制可能存在多主模式冲突。可使用中断程序从此状态完全恢复，方法是执行复位或返回到默认状态。

溢出错误

当主器件发送完数据字节，而从器件尚未将上一个收到的数据所产生的 RXNE 位清零时，将出现溢出情况。出现溢出错误时：

- OVR 位置 1 并在 ERRIE 位置 1 时生成一个中断。

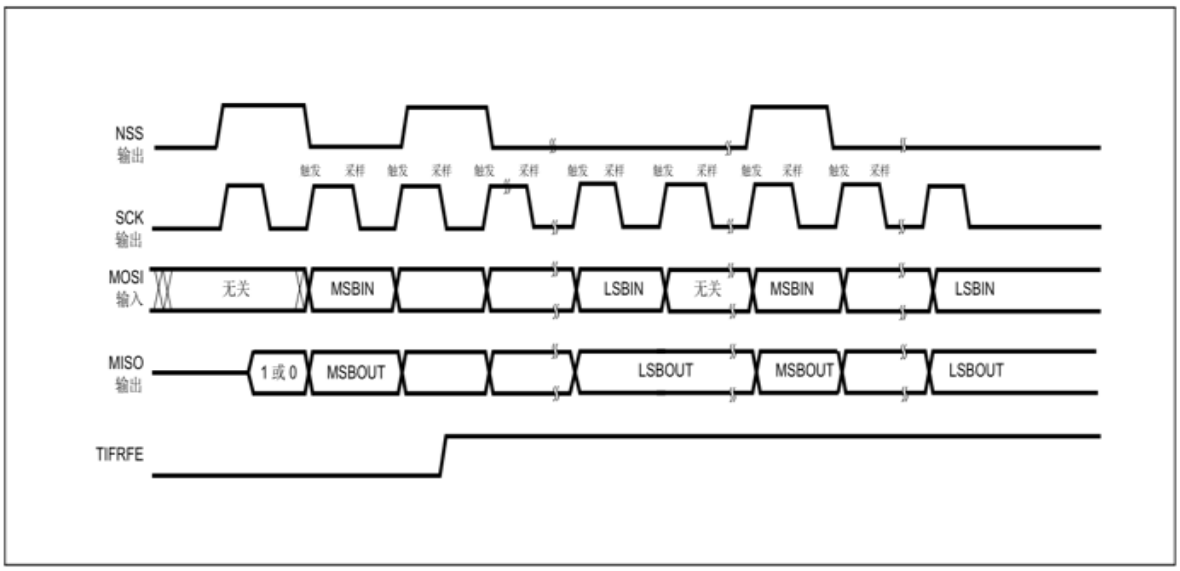
在这种情况下，接收器缓冲区内容不会被来自主器件的新数据更新。读取 SPI_DR 寄存器将返回此字节。主器件后续发送的所有其它字节均将丢失。
依次读取 SPI_DR 寄存器和 SPI_SR 寄存器可将 OVR 清除。

CRC 错误

当 SPI_CR1 寄存器中的 CRCEN 位置 1 时，此标志用于验证接收数据的有效性。如果移位寄存器中接收的值与 SPI_RXCRCR 的值不匹配，SPI_SR 寄存器中的 CRCERR 标志将置 1。

TI 模式帧格式错误

如果 SPI 在从模式下工作，并配置为符合 TI 模式协议，则在持续通信期间出现 NSS 脉冲时，将检测到 TI 模式帧格式错误。出现此错误时，SPI_SR 寄存器中的 FRE 标志将置 1。发生错误时不会关闭 SPI，但会忽略 NSS 脉冲，并且 SPI 会等待至下一个 NSS 脉冲，然后再开始新的传输。由于错误检测可能导致丢失两个数据字节，因此数据可能会损坏。
读取 SPI_SR 寄存器时，将清零 FRE 标志。如果 ERRIE 位置 1，则检测到帧格式错误时将产生中断。在这种情况下，由于无法保证数据的连续性，应关闭 SPI，并在重新使能从 SPI 后，由主器件重新发起通信。
图 286. TI



13.3.11 SPI 中断

表 124. SPI 中断请求

中断事件	事件标志	使能控制位
发送缓冲区为空	TXE	TXEIE
接收缓冲区非空	RXNE	RXNEIE
主模式故障	MODF	ERRIE
溢出错误	OVR	
CRC 错误	CRCERR	
TI 帧格式错误	FRE	ERRIE

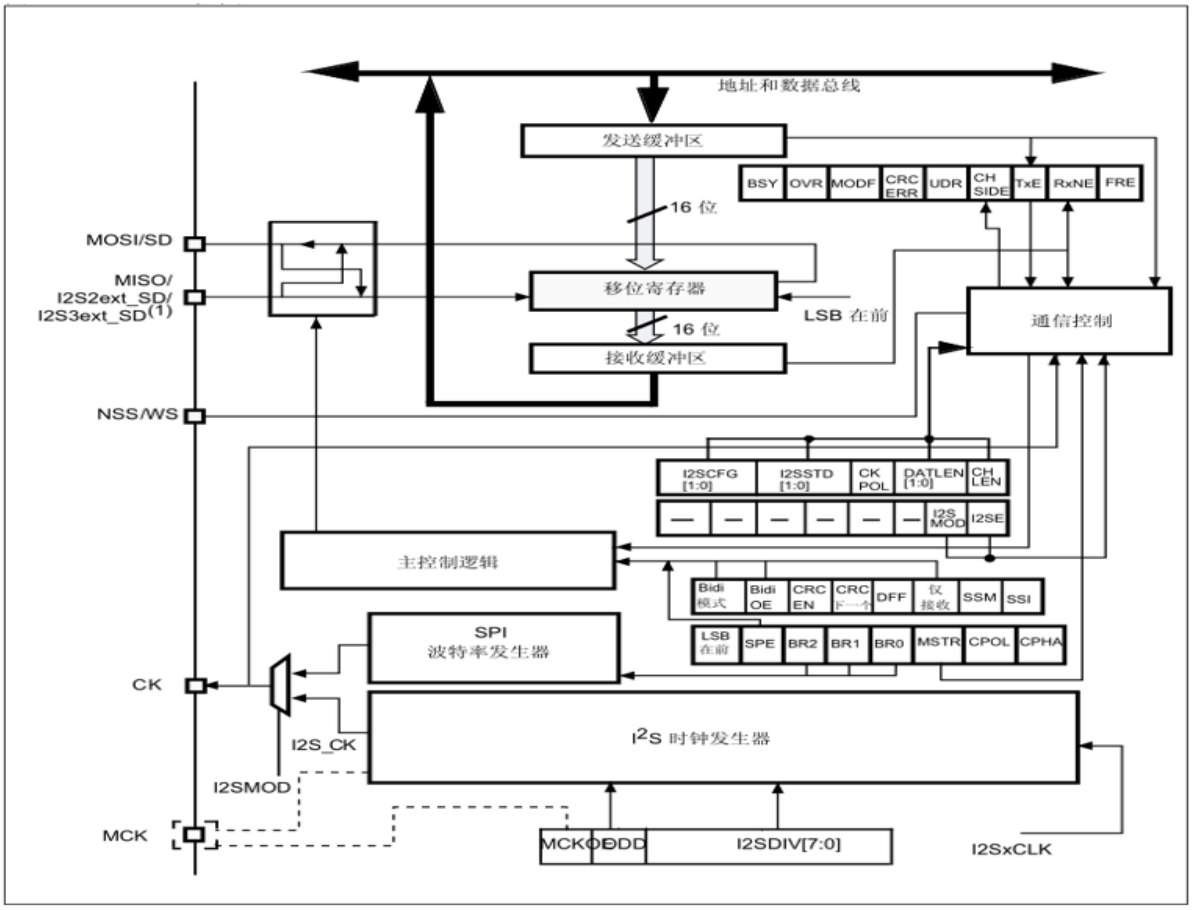
13.4 I2S

13.4.1 I2S 一般说明

I2S 的框图如 图 287 所示。

图 287. I2S

STC MCU



1. I2S2ext_SD 和 I2S3ext_SD 为扩展 SD 引脚，用于控制 I2S 全双工模式。

当使能 I2S 功能（将 SPI_I2SCFGR 寄存器中的 I2SMOD 位置 1）时，SPI 可用作音频 I2S 接口。此接口使用几乎与 SPI 相同的引脚、标志和中断。

I2S 与 SPI 共用以下三个引脚：

- SD：串行数据（映射到 MOSI 引脚），用于发送或接收两个时分复用的数据通道上的数据（仅半双工模式）。
- WS：字选择（映射到 NSS 引脚），是主模式下的数据控制信号输出以及从模式下的数据控制信号输入。
- CK：串行时钟（映射到 SCK 引脚），是主模式下的串行时钟输出以及从模式下的串行时钟输入。
- I2S2ext_SD 和 I2S3ext_SD：用于控制 I2S 全双工模式的附加引脚（映射到 MISO 引脚）。

当某些外部音频设备需要使用主时钟输出时，可以使用附加引脚：

- MCK：当 I2S 配置为主模式（并且 SPI_I2SPR 寄存器中的 MCKOE 位置 1）时，使用主时钟（单独映射）输出此附加时钟，该时钟输出频率 256 × FS，其中 FS 为音频信号采样频率。

I2S 在主模式下使用自身的时钟发生器生成通信时钟。此时钟发生器也是主时钟输出的源。在 I2S 模式下可以使用两个额外的寄存器。一个是时钟发生器配置寄存器 SPI_I2SPR，另一个是通用 I2S 配置寄存器 SPI_I2SCFGR（音频标准、从/主模式、数据格式、数据包帧、时钟极性等）。在 I2S 模式下不使用 SPI_CR1 寄存器和所有 CRC 寄存器。同样，也不使用 SPI_CR2 寄存器中的 SSOE 位以及 SPI_SR 中的 MODF 和 CRCERR 位。

I2S 使用和 SPI 相同的寄存器 (SPI_DR) 进行 16 位数据传输。

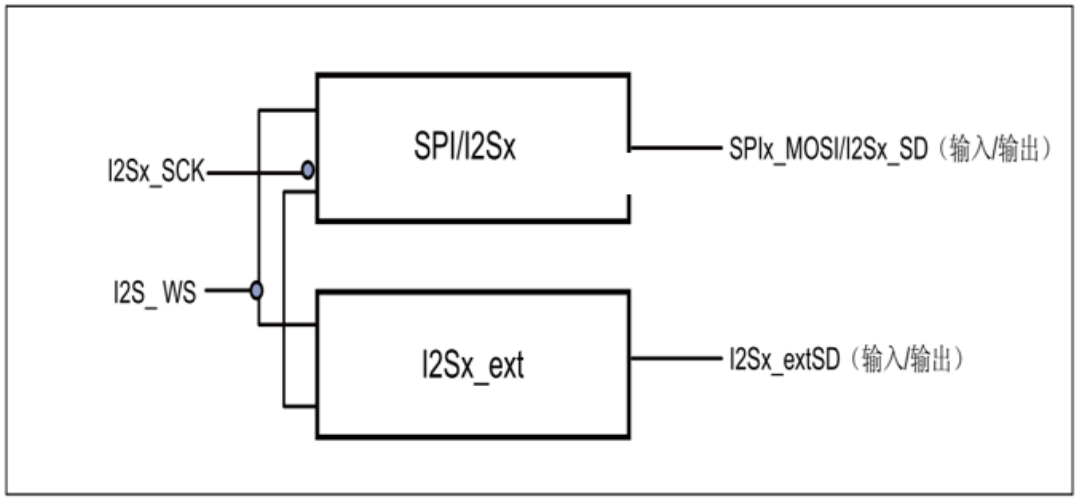
13.4.2 I2S 全双工

为支持 I2S 全双工模式，除了 I2S2 和 I2S3，还可以使用两个额外的 I2S，它们称为扩展 I2S（I2S2_ext、I2S3_ext）（参见 图 288）。因此，第一个 I2S 全双工接口基于 I2S2 和

I2S2_ext，第二个基于 I2S3 和 I2S3_ext。

注意： I2S2_ext 和 I2S3_ext 仅用于全双工模式。

图 288. I2S



1. 其中 X 可以是 2 或 3。

I2Sx 可以在主模式下工作。因此：

- 只有 I2Sx 可在半双工模式下输出 SCK 和 WS
- 只有 I2Sx 可在全双工模式下向 I2S2_ext 和 I2S3_ext 提供 SCK 和 WS。

扩展 I2S (I2Sx_ext) 只能用于全双工模式。I2Sx_ext 始终在从模式下工作。

I2Sx 和 I2Sx_ext 均可用于发送和接收。

13.4.3支持的音频协议

四线总线通常需要处理时分复用的两个通道（右通道和左通道）上的音频数据。但是，只有一个 16 位寄存器进行发送和接收。所以，需由软件来把和通道对应的数据写入数据寄存

器，以及从数据寄存器中读取数据，并通过检查 SPI_SR 寄存器中的 CHSIDE 位来识别对应的通道。始终先发送“左通道数据”，而后再发送右通道数据（对于 PCM 协议来说，

CHSIDE 没有意义）。

有四种数据和帧格式组合，可采用下列格式发送数据：

- 将 16 位数据封装在 16 位帧中
- 将 16 位数据封装在 32 位帧中
- 将 24 位数据封装在 32 位帧中
- 将 32 位数据封装在 32 位帧中

当使用 32 位数据包中的 16 位数据时，前 16 位 (MSB) 为有效位，16 位 LSB 被强制清零，无需任何软件操作或 DMA 请求（只需一个读/写操作）。

如果应用程序首选 DMA，则 24 位和 32 位数据帧需要对 SPI_DR 执行两次 CPU 读取或写入操作，或者需要两次 DMA 操作。24 位的数据帧，硬件会将 8 位非有效位扩展到带有 0 位的 32 位。

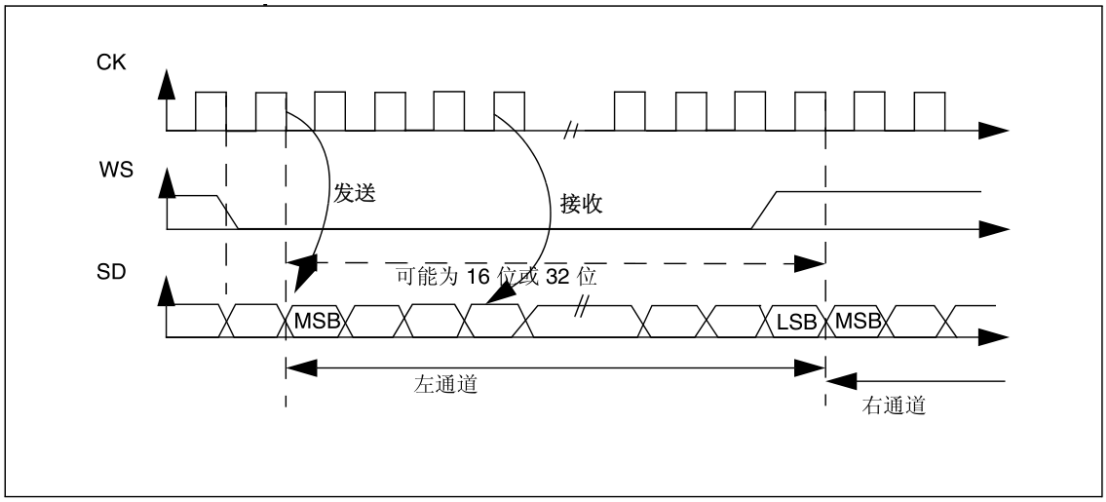
对于所有数据格式和通信标准而言，始终会先发送最高有效位（MSB 优先）。

I 2 S 接口支持四种音频标准，可使用 SPI_I2SCFGR 寄存器中的 I2SSTD[1:0] 和 PCMSYNC 位对其进行配置。

I 2 S Philips 标准

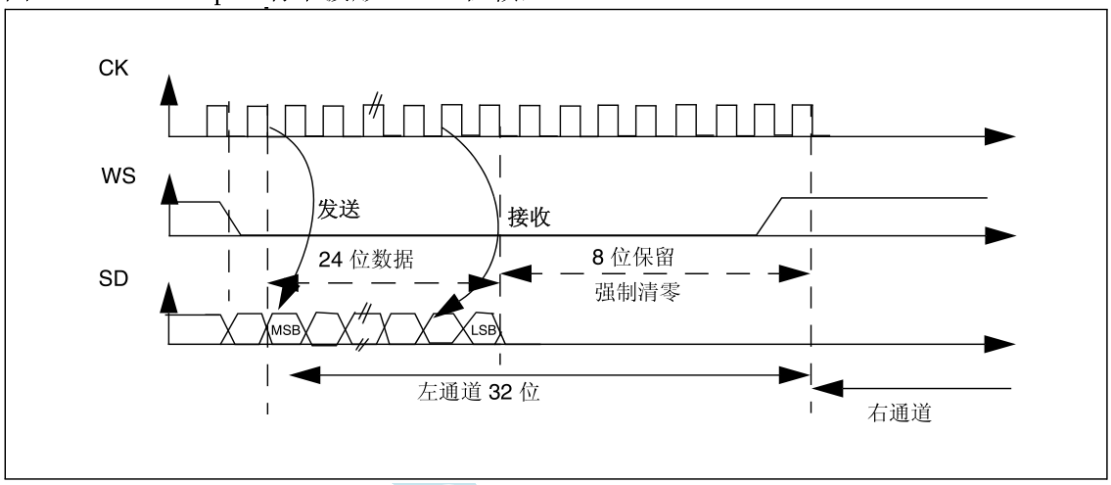
使用 WS 信号来指示当前正在发送的数据所属的通道。该信号从当前通道数据的第一个位(MSB) 之前的一个时钟开始有效。

图 289. I 2 S Philips 协议波形（16/32 位全精度，CPOL = 0）



发送方在时钟信号 (CK) 的下降沿改变数据，接收方在上升沿读取数据。WS 信号也在 CK 的下降沿变化。

图 290. I 2 S Philips 标准波形（24 位帧，CPOL = 0 ）

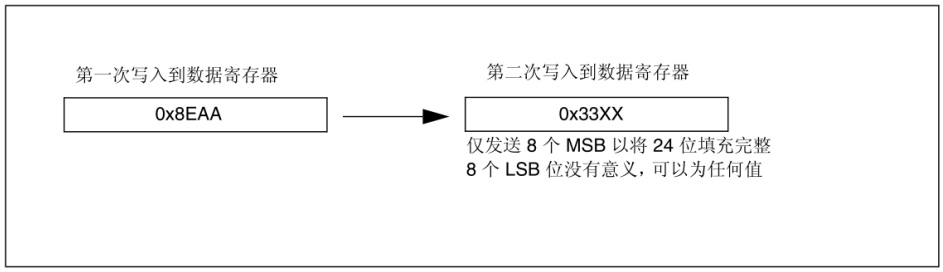


该模式需要对 SPI_DR 执行两次写入或读取操作。

- 在发送模式下：

如果需要发送 0x8EAA33（24 位）：

图 291. 发送 0x8EAA33



- 在接收模式下：

如果接收数据 0x8EAA33：

图 292. 接收 0x8EAA33

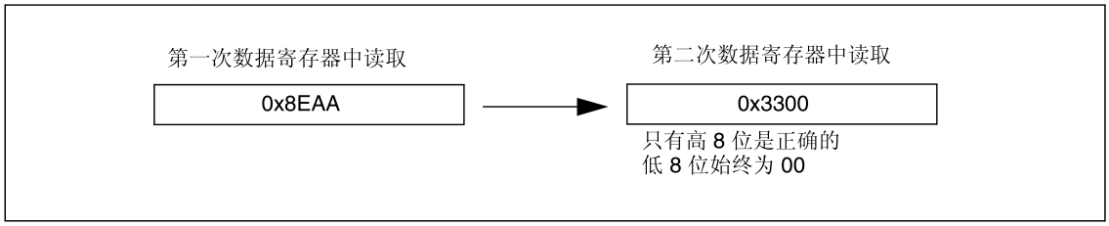
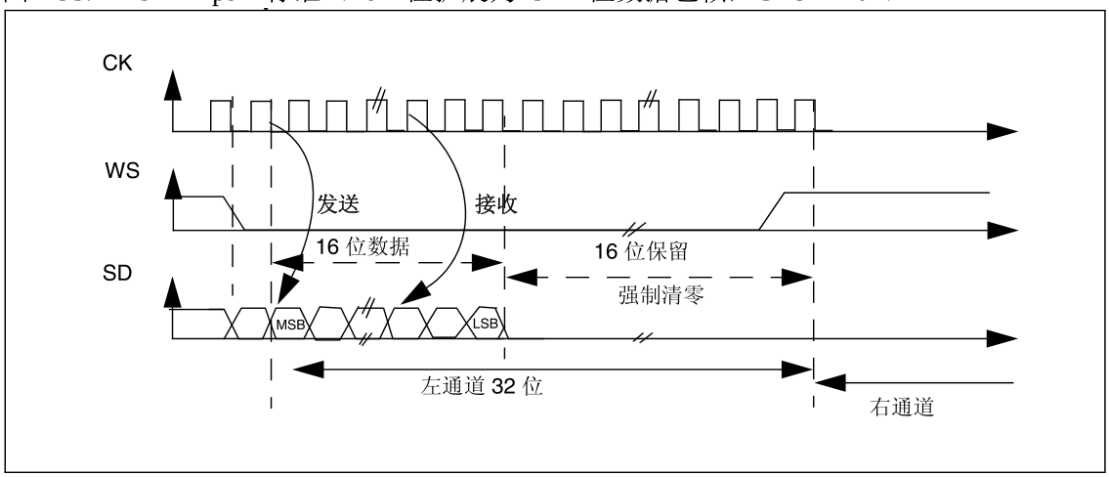


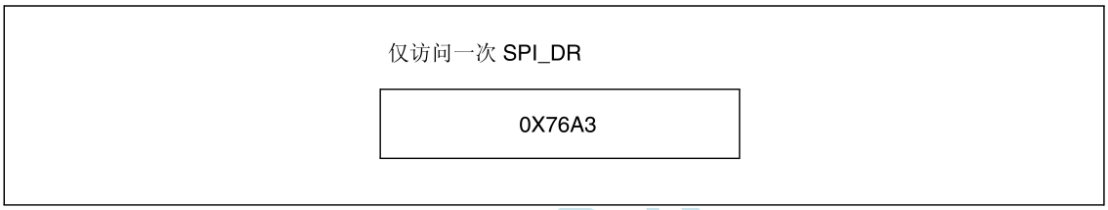
图 293. I 2 S Philips 标准（16 位扩展为 32 位数据包帧，CPOL = 0 ）



如果 I 2 S 配置成将 16 位数据扩展到 32 位帧，只需要访问一次寄存器 SPI_DR。硬件会将其余 16 个位强制设置为 0x0000，以便将数据扩展为 32 位格式。

如果要发送的数据或已接收的数据为 0x76A3（0x76A30000 扩展为 32 位），则需要执行 图 294 中显示的操作。

图 294. 示例



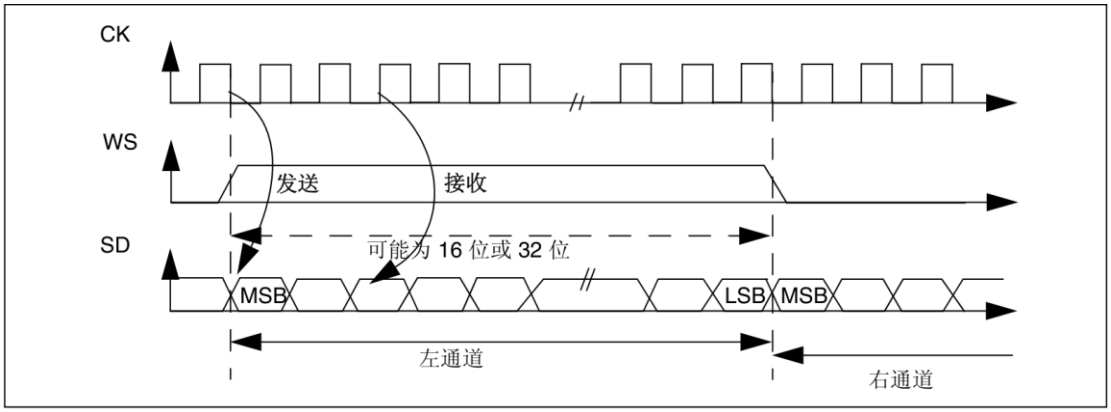
发送时，每次将 16 位数据写入 SPI_DR，硬件就会置位 TXE 标志，并在中断使能的情况下触发中断，以把下一个要发送的数据加载到 SPI_DR。即使硬件填充的低 16 位 0x0000 还未发送，也会如此，因为低 16 位是由硬件发送。

接收时，接收到第一个半字（高 16 位），则硬件将 RXNE 标志置 1，并在中断使能的情况下触发中断。这样，就延长了两个写入或读取操作之间的时间间隔，从而可防止出现下溢或上溢情况（具体取决于数据传输方向）。

MSB 对齐标准

此标准同时生成 WS 信号和第一个数据位（即 MSBit）。

图 295. MSB 对齐的 16 位或 32 位全精度长度，CPOL = 0



发送方在时钟信号的下降沿改变数据；接收方在上升沿读取数据。

图 296. MSB 对齐的 24 位帧长度，CPOL = 0

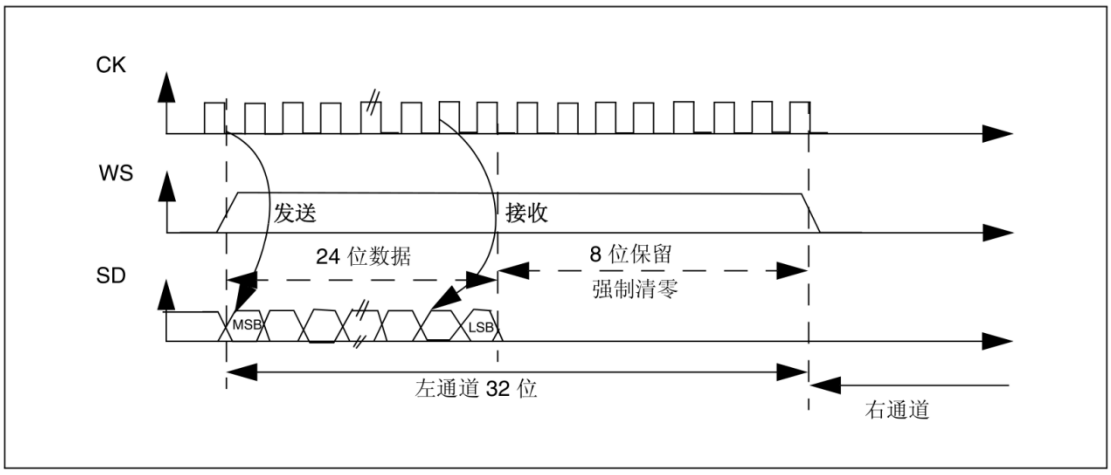
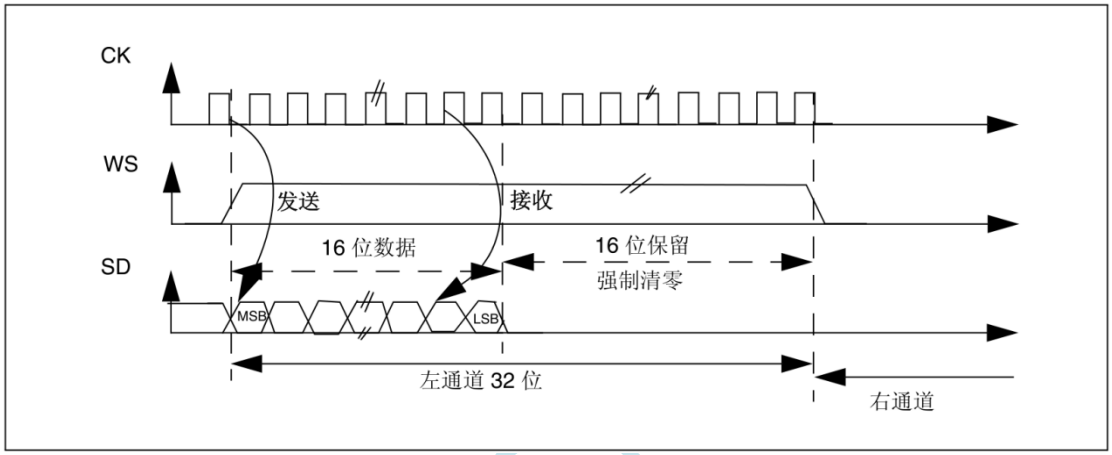


图 297. 扩展为 32 位数据包帧的 MSB 对齐的 16 位，CPOL = 0



LSB 对齐标准

该标准与 MSB 对齐标准类似（对于 16 位和 32 位全精度帧格式，没有任何不同）。

图 298. LSB 对齐的 16 位或 32 位全精度，CPOL = 0

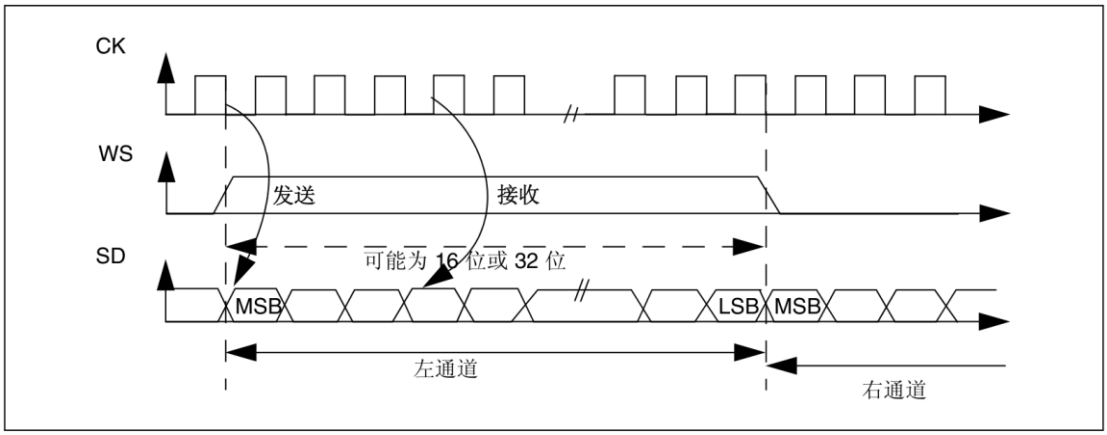
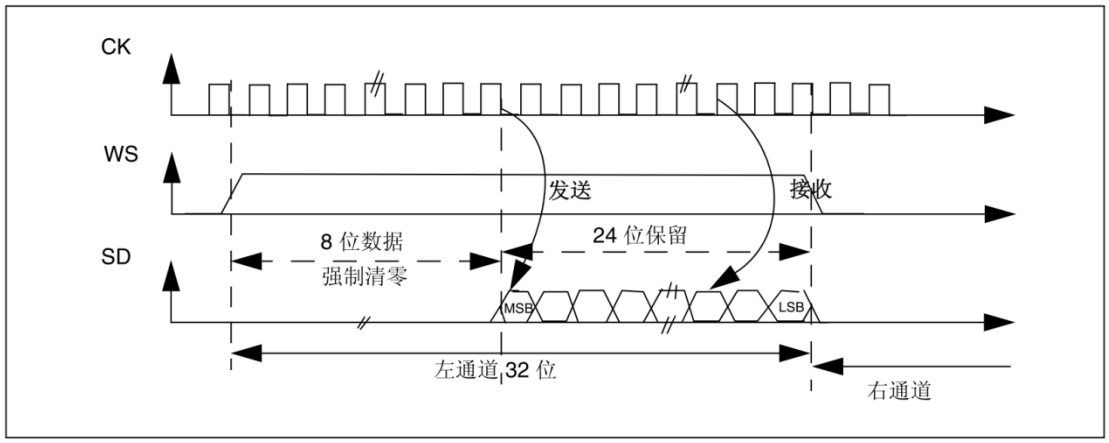


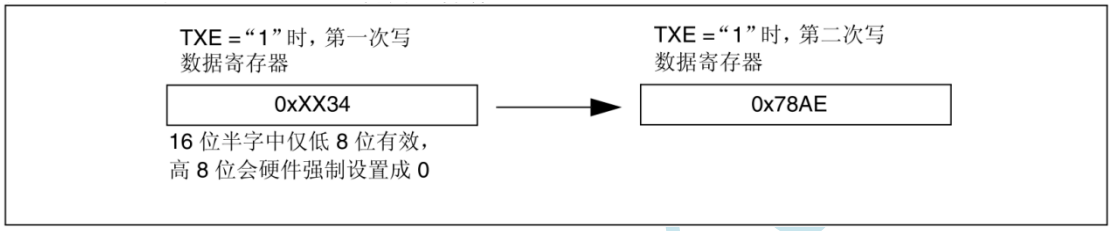
图 299. LSB 对齐的 24 位帧长度，CPOL = 0



● 在发送模式下：

如果需要发送数据 0x3478A，则需要通过软件或 DMA 对 SPI_DR 寄存器执行两次写入操作。操作如下所示。

图 300. 发送 0x3478AE



● 在接收模式下：

如果接收到数据 0x3478AE，则在每个 RXNE 事件时需要对 SPI_DR 执行两次连续的读取操作。

图 301. 接收 0x3478AE 时所需的操作

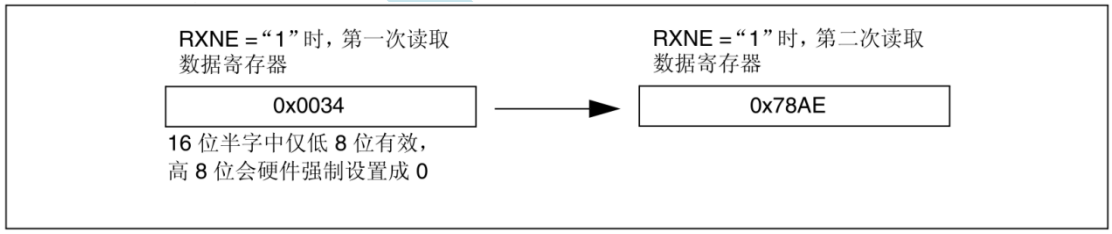
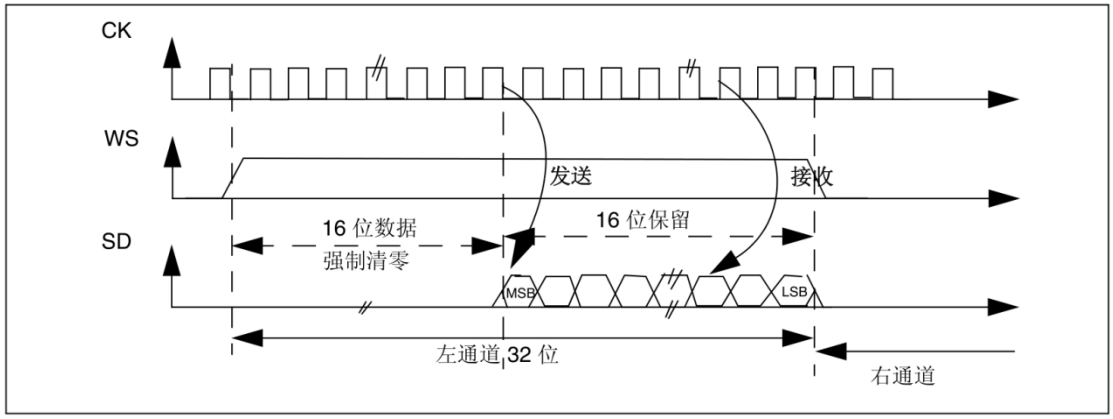


图 302. 扩展为 32 位数据包帧的 LSB 对齐的 16 位，CPOL = 0



如果 I2S 配置成将 16 位数据扩展到 32 位帧，只需要访问一次寄存器 SPI_DR。硬件会将其余 16 个

位强制设置为 0x0000，以便将数据扩展为 32 位格式。在这种情况下，其对应于半字 MSB。

如果要发送的数据或已接收的数据为 0x76A3（0x0000 76A3 扩展为 32 位），则需要执行 图 303 中显示的操作。

图 303. 扩展为 32 位数据包帧的 LSB 对齐的 16

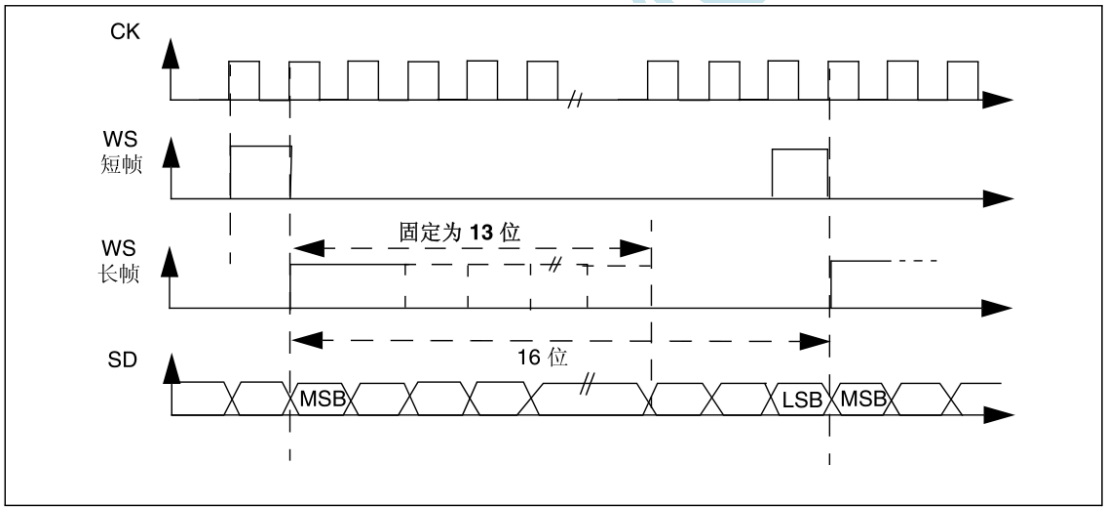


在发送模式下，TXE 置位时，应用程序需要写入要发送的数据（此例中，为 0x76A3）。首先发送 0x000 字段（扩展到 32 位）。有效数据 (0x76A3) 发送完毕后，TXE 会被再次置位。在接收模式下，当接收到有效半字后（而非 0x0000 字段），即会置位 RXNE。这样，就延长了两个写入或读取操作之间的时间间隔，以防止出现下溢或上溢情况。

PCM 标准

对于 PCM 标准，无需使用通道信息。有两种 PCM 模式（短帧和长帧），并且可使用 SPI_I2SCFGR 中的 PCMSYNC 位来配置。

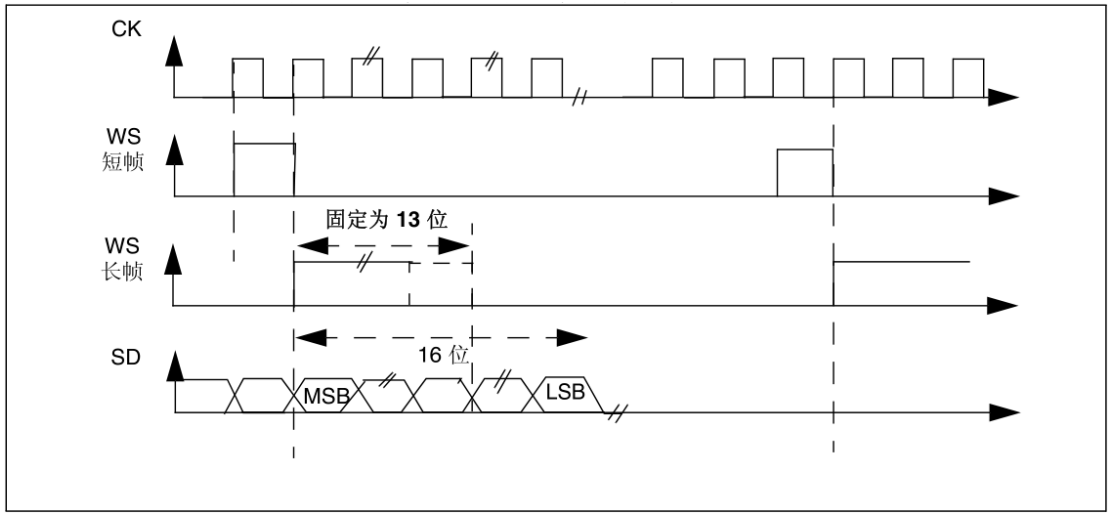
图 304. PCM 标准波形（16）



对于长帧同步，在主模式下会将 WS 信号持续 13 个周期。

对于短帧同步，WS 同步信号的持续时间仅为一个周期。

图 305. PCM 标准波形（16 位扩展到 32 位数据包帧）



注意： 对于两种模式（主 / 从模式）和两种同步（短 / 长同步），即使在从模式下，也需要指定两组连续数据（以及两个同步信号）之间位的个数（ SPI_I2SCFGR 寄存器中的 DATLEN 位和 CHLEN 位）。

13.4.4时钟发生器

I 2 S 比特率用来确定 I 2 S 数据线上的数据流和 I 2 S 时钟信号频率。

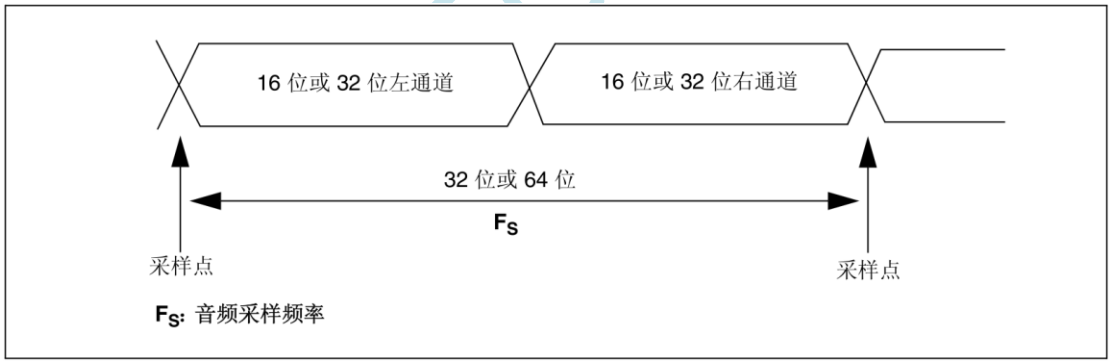
I 2 S 比特率 = 每个通道的位数 × 通道数 × 音频采样频率

对于 16 位双通道音频，I 2 S 比特率的计算公式如下：

I 2 S 比特率 = 16 × 2 × F S

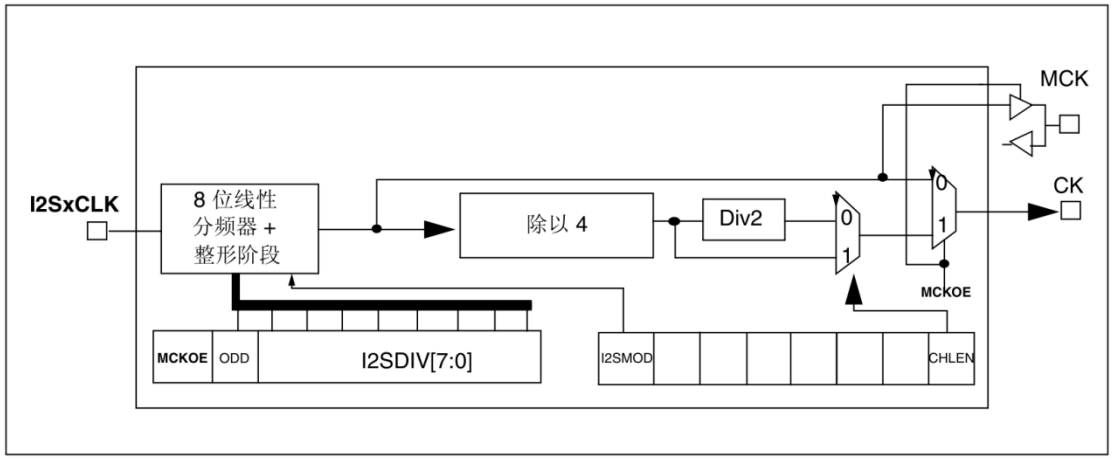
如果数据包为 32 位宽，则 I 2 S 比特率 = 32 x 2 x F S 。

图 306. 音频采样频率定义



配置主模式时，需要正确地对线性分频器进行设置，以便采用所需的音频频率进行通信。

图 307. I 2 S



1. 其中，X 可以是 2 或 3。

图 306 展示了通信时钟架构。要实现高品质的音频性能。

音频采样频率可能是 192 kHz、96 kHz 或 48 kHz。为达到所需频率，需要根据以下公式对线性分频器进行编程：

输出主时钟（SPI_I2SPR 寄存器中的 MCKOE 置 1)时：

F S = I2SxCLK / [(16*2)*((2*I2SDIV)+ODD)*8)]（通道帧宽度为 16 位时）

F S = I2SxCLK / [(32*2)*((2*I2SDIV)+ODD)*4)]（通道帧宽度为 32 位时）

关闭主时钟输出（MCKOE 位清零）时：

F S = I2SxCLK / [(16*2)*((2*I2SDIV)+ODD))]（通道帧宽度为 16 位时）

F S = I2SxCLK / [(32*2)*((2*I2SDIV)+ODD))]（通道帧宽度为 32 位时）

表 125 提供了针对不同时钟配置的示例精度值。

注意： 还有其他配置可达到更好的时钟精度。

表 125. 音频频率精度（针对 PLLM VCO = 1 MHz 或 2 MHz ） (1)

主时钟	目标 f_s (Hz)	数据格式	PLLI2SN	PLLI2SR	I2SDIV	I2SODD	实时 f_s (Hz)	误差
输出关闭	8000	16 位	192	2	187	1	8000	0.0000 %
		32 位	192	3	62	1	8000	0.0000 %
	16000	16 位	192	3	62	1	16000	0.0000 %
		32 位	256	2	62	1	16000	0.0000 %
	32000	16 位	256	2	62	1	32000	0.0000 %
		32 位	256	5	12	1	32000	0.0000 %
	48000	16 位	192	5	12	1	48000	0.0000

								%
		32 位	384	5	12	1	48000	0.0000 %
	96000	16 位	384	5	12	1	96000	0.0000 %
		32 位	424	3	11	1	96014.49219	0.0151 %
	22050	16 位	290	3	68	1	22049.87695	0.0006 %
		32 位	302	2	53	1	22050.23438	0.0011 %
	44100	16 位	302	2	53	1	44100.46875	0.0011 %
		32 位	429	4	19	0	44099.50781	0.0011 %
	192000	16 位	424	3	11	1	192028.9844	0.0151 %
		32 位	258	3	3	1	191964.2813	0.0186 %
输出使能	8000	无关	256	5	12	1	8000	0.0000 %
	16000	无关	213	2	13	0	16000.60059	0.0038 %
	32000	无关	213	2	6	1	32001.20117	0.0038 %
	48000	无关	258	3	3	1	47991.07031	0.0186 %
	96000	无关	344	2	3	1	95982.14063	0.0186 %
	22050	无关	429	4	9	1	22049.75391	0.0011 %
	44100	无关	271	2	6	0	44108.07422	0.0183 %

1. 该表格仅给出了不同时钟配置的示例值。还有其他配置可达到更好的时钟精度。

13.4.5I 2 S 主模式

I 2 S 可如下配置：

- 发送主器件或接收主器件（使用 I2Sx 的半双工模式）
- 同时收发的主器件（使用 I2Sx 和 I2Sx_ext 的全双工模式）。

I 2 S 工作在主模式，串行时钟由引脚 CK 输出，字选信号由引脚 WS 产生。可以通过设置寄存器 SPI_I2SPR 的 MCKOE 位来选择输出或者不输出主时钟 (MCK)。

步骤

1. 设置 SPI_I2SPR 寄存器的 I2SDIV[7:0] 位，以定义串行时钟波特率，从而达到适当的音频采样频率。SPI_I2SPR 寄存器的 ODD 位也需要设置。
2. 设置 CKPOL 位，定义时钟在空闲时的电平状态。如果需要为外部 DAC/ADC 音频组件提供主时钟 MCK，则将 SPI_I2SPR 寄存器的 MCKOE 位置 1（I2SDIV 和 ODD 值应根据 MCK 输出

的状态进行计算。有关详细信息，请参见 第 14.4.4 节：时钟发生器 ）。

3. 将 SPI_I2SCFGR 中的 I2SMOD 位置 1 以激活 I 2 S 功能，通过 I2SSTD[1:0] 和 PCMSYNC 位选择 I 2 S 标准，通过 DATLEN[1:0] 位选择数据长度并通过配置 CHLEN 位选择每个通道的位数。此外，通过 SPI_I2SCFGR 寄存器的 I2SCFG[1:0] 位选择 I 2 S 主模式和方向（发送器或接收器）。

4. 如果需要，通过对 SPI_CR2 寄存器执行写操作来选择所有可能的中断源和 DMA 功能。
5. SPI_I2SCFGR 寄存器的 I2SE 位必须置 1。WS 和 CK 配置为输出模式。如果 SPI_I2SPR 的 MCKOE 位置 1，则 MCK 也是输出。

发送序列

将半字写入发送缓冲区后，发送序列随即开始。

通常，写入发送缓冲区的第一个数据对应于左通道数据。数据从发送缓冲区传输到移位寄存器时，TXE 置 1，并且必须将对应于右通道的数据写入发送缓冲区。CHSIDE 标志指示将发送的数据对应的通道。TXE 标志置 1 时，CHSIDE 标志有意义，因为该标志在 TXE 变为高电平时进行更新。

一个完整帧表示先进行左通道数据发送再进行右通道数据发送。不存在仅发送左通道的部分帧。首位发送期间，数据按半字并行加载到 16 位移位寄存器中，然后以串行方式移位并输出到 MOSI/SD 引脚（MSB 在前）。每次数据从发送缓冲区传输到移位寄存器后，TXE 标志都将置 1，如果 SPI_CR2 寄存器的 TXEIE 位置 1，将产生中断。

有关各种 I 2 S 标准模式中的写操作的更多详细信息，请参见 第 14.4.3 节：支持的音频协议 。为确保连续进行音频数据发送，必须在当前数据发送结束前将下一个要发送数据写入 SPI_DR。要通过将 I2SE 清零来关闭 I 2 S，必须等待 TXE 置 1 且 BSY 清零。

接收序列

此工作模式与发送模式相同，只有第 3 点存在不同（请参见 第 14.4.5 节： I 2 S 主模式 所述的步骤），即通过 I2SCFG[1:0] 位设置主器件接收模式。

无论数据或通道长度如何，音频数据始终按 16 位数据包进行接收。这意味着，每当接收缓冲区满时，RXNE 标志即置 1，并且如果 SPI_CR2 寄存器的 RXNEIE 位置 1，还将产生中断。所接收的右通道或左通道的音频值可通过一次或两次接收操作进入接收缓冲区，具体取决于数据和通道长度配置。

读取 SPI_DR 寄存器即会使 RXNE 位清零。

CHSIDE 在每次接收后进行更新。它由 I 2 S 单元所产生的 WS 信号触发。

有关各种 I 2 S 标准模式中的读操作的更多详细信息，请参见 第 14.4.3 节：支持的音频协议 。

如果在先前收到的数据尚未读取时又接收到新数据，将产生溢出错误并将 OVR 标志置 1。如果 SPI_CR2 寄存器的 ERRIE 位置 1，将产生中断以指示该错误。

要关闭 I 2 S，需要执行特定操作来确保 I 2 S 正确完成传输周期而不启动新的数据传输。该序列取决于数据和通道长度的配置，以及所选的音频协议模式。在以下情况下：

● 32 位通道长度上扩展的 16 位数据长度（DATLEN = 00 且 CHLEN = 1），使用 LSB 对齐模式 (I2SSTD = 10)

a) 等待倒数第二个 RXNE = 1 (n — 1)
b) 然后等待 17 个 I 2 S 时钟周期（使用软件循环）
c) 关闭 I 2 S (I2SE = 0)

● 32 位通道长度上扩展的 16 位数据长度（DATLEN = 00 且 CHLEN = 1），使用 MSB 对齐、I 2 S 或 PCM 模式（即 I2SSTD = 00、I2SSTD = 01 或 I2SSTD = 11）

a) 等待最后一个 RXNE
b) 然后等待 1 个 I 2 S 时钟周期（使用软件循环）
c) 关闭 I 2 S (I2SE = 0)

● 对于 DATLEN 和 CHLEN 的所有其它组合，无论通过 I2SSTD 位选择何种音频模式，都将执行以下序列来关闭 I 2 S：

a) 等待倒数第二个 RXNE = 1 (n — 1)

b) 然后等待 1 个 I2S 时钟周期（使用软件循环）
c) 关闭 I2S (I2SE = 0)
注意： 传输期间， BSY 标志保持低电平。

13.4.6 I2S 从模式

I2S 可如下配置：
- 发送从器件或接收从器件（使用 I2Sx 的半双工模式）
- 同时收发的从器件（使用 I2Sx 和 I2Sx_ext 的全双工模式）。

此工作模式所遵循的规则与 I2S 主模式基本相同。在从模式下，I2S 接口不产生时钟。时钟和 WS 信号从 I2S 接口所连接的外部主器件输入。这样，用户便不需要配置时钟。

应遵循如下配置步骤：

1. 将 SPI_I2SCFGR 寄存器的 I2SMOD 位置 1 以激活 I2S 功能，通过 I2SSTD[1:0] 位选择 I2S 标准，通过 DATLEN[1:0] 位选择数据长度并通过 CHLEN 位选择帧中每个通道的位数。此外，通过 SPI_I2SCFGR 寄存器的 I2SCFG[1:0] 位选择从器件的模式（发送或接收）。
2. 如果需要，通过对 SPI_CR2 寄存器执行写操作来选择所有可能的中断源和 DMA 功能。
3. SPI_I2SCFGR 寄存器的 I2SE 位必须置 1。

发送序列

当外部主器件发送时钟并且通过 NSS_WS 信号请求传输数据时，发送序列开始。必须首先使能从器件，然后外部主器件才能开始通信。主器件开始通信前，从器件还必须加载 I2S 数

据寄存器。

对于 I2S、MSB 对齐和 LSB 对齐模式，要写入数据寄存器的第一个数据项对应于左通道的数据。通信开始时，数据从发送缓冲区传输到移位寄存器。TXE 标志随即置 1，以请求将右通道的数据写入 I2S 数据寄存器。

CHSIDE 标志指示将发送的数据对应的通道。与主发送模式相比，在从模式下，CHSIDE 由来自外部主器件的 WS 信号触发。这意味着，从器件需要首先为发送第一个数据做好准备，然后主器件才能产生时钟。WS 置位意味着首先发送左通道数据。

注意： 必须要在主器件发出的第一个时钟出现在 CK 线上至少 2 个 PCLK 周期之前置位 I2SE 。

首位发送期间，数据按半字从内部总线并行加载到 16 位移位寄存器中，然后以串行方式移

位并输出到 MOSI/SD 引脚（MSB 在前）。每次数据从发送缓冲区传输到移位寄存器后，TXE 标志都将置 1，如果 SPI_CR2 寄存器的 TXEIE 位置 1，将产生中断。

请注意，仅当 TXE 标志为 1 时，才可以尝试向发送缓冲区写入数据。

有关各种 I2S 标准模式中的写操作的更多详细信息，请参见 第 14.4.3 节：支持的音频协议 。

为确保连续进行音频数据发送，必须在当前数据发送结束前将下一个要发送数据写入 SPI_DR 寄存器。如果在数据尚未写入 SPI_DR 寄存器时下一个数据通信的首个时钟边沿到来，下溢标志将置 1 并可能产生中断。通过这种方式，软件可以获知所传输的数据不正确。

如果 SPI_CR2 寄存器的 ERRIE 位置 1，则当 SPI_SR 寄存器中的 UDR 标志变为 1 时，将产生中断。这种情况下，必须关闭 I2S 并从左通道开始重新启动数据传输。

要通过将 I2SE 位清零来关闭 I2S，必须等待 TXE 置 1 且 BSY 清零。

接收序列

此工作模式与发送模式相同，只有第 1 点存在不同（请参见 第 14.4.6 节： I2S 从模式 所述的步骤），即通过 SPI_I2SCFGR 寄存器的 I2SCFG[1:0] 位设置从器件接收模式。无论数据长度或通道长度如何，音频数据始终按 16 位数据包进行接收。这意味着，每当接收缓冲区填满时，SPI_SR 寄存器中的 RXNE 标志即置 1，并且如果 SPI_CR2 寄存器的 RXNEIE 位置 1，还将产生中断。所接收的右通道或左通道的音频值可能通过一次或两次接收操作进入接收缓冲区，具体取决于数据长度和通道长度配置。

每次接收要从 SPI_DR 读取的数据时，CHSIDE 标志都将更新。该标志由外部主器件所管理的外部 WS 线路触发。读取 SPI_DR 寄存器即会使 RXNE 位清零。

有关各种 I2S 标准模式中的读操作的更多详细信息，请参见 第 14.4.3 节：支持的音频协议 。

如果在先前收到的数据尚未读取时又接收到新数据，将产生溢出错误并将 OVR 标志置 1。如果 SPI_CR2 寄存器的 ERRIE 位置 1，将产生中断以指示该错误。要在接收模式下关闭 I2S，必须在接收到最后一个 RXNE = 1 后立即将 I2SE 清零。

注意： 外部主器件应能够通过音频通道以 16 位或 32 位数据包发送 / 接收数据。

13.4.7状态标志

应用程序可通过三个状态标志来全面监视 I2S 总线的状态。

忙标志 (BSY)

BSY 标志由硬件置 1 和清零（写入此标志没有任何作用）。该标志表示 I2S 通信的状态。

硬件将 BSY 置 1 时，指示 I2S 正在忙于通信。在主接收模式 (I2SCFG = 11) 中，BSY 标志的情况例外，该标志在接收期间仍保持低电平。

如果软件需要关闭 I2S，可使用 BSY 标志检测传输是否结束。以避免破坏最后一个数据的传输。为此，必须严格遵循下述步骤。

在传输开始时（I2S 处于主接收器模式时除外）硬件将 BSY 标志置 1。

出现以下情况时，BSY 标志被硬件清零：

- 传输完成（主发送模式除外，在该模式下通信是连续的）
- 关闭 I2S 时

当通信连续时：

- 在主发送模式下，BSY 标志在所有传输期间均保持高电平
- 在从模式下，BSY 标志在每次传输之间变为低电平并持续一个 I2S 时钟周期

注意：请勿使用 BSY 标志处理每次数据发送或接收，最好改用 TXE 标志和 RXNE 标志。

发送缓冲区为空 (TXE)

如果此标志置 1，表示发送缓冲区为空，可将要发送的下一个数据加载到其中。发送缓冲区已包含要发送的数据时，TXE 标志复位。关闭 I2S（I2SE 位复位）时，该标志也会复位。

接收缓冲区非空 (RXNE)

此标志置 1 时，表示接收缓冲区中存在有效的已接收数据。读取 SPI_DR 寄存器时，该标志复位。

通道方向 (CHSIDE)

在发送模式下，此标志将在 TXE 变为高电平时进行刷新。此标志指示 SD 上要传输的数据所属的通道。如果在从发送模式下发生下溢错误事件，此标志将不可靠，在恢复通信前需要关闭并重新开启 I2S。

在接收模式下，此标志将在数据接收到 SPI_DR 中时进行刷新。此标志指示已接收的数据所属的通道。请注意，如果发生错误（例如 OVR），此标志将失去意义，应通过关闭并重新使能 I2S（根据需要更改配置）来将其复位。

此标志在 PCM 标准中没有意义（短帧和长帧模式）。

当 SPI_SR 中的 OVR 或 UDR 标志置 1，并且 SPI_CR2 中的 ERRIE 位也置 1 时，将产生中断。中断源被清除后，可通过读取 SPI_SR 状态寄存器来将此中断清零。

13.4.8错误标志

I2S 单元共有三个错误标志。

下溢标志 (UDR)

在从发送模式下，如果在软件尚未将任何值加载到 SPI_DR 之前出现第一个数据发送时钟，此标志将置 1。SPI_I2SCFGR 中的 I2SMOD 位置 1 时，可以使用此标志。如果 SPI_CR2 中的 ERRIE 位置 1，可产生中断。

UDR 位通过 SPI_SR 寄存器上的读操作进行清零。

上溢标志 (OVR)

如果在尚未从 SPI_DR 读取上一个数据时又接收到新数据，此标志将置 1。因此，传入的数据将丢失。如果 SPI_CR2 中的 ERRIE 位置 1，可产生中断。

这种情况下，将不会用接收到的新数据更新接收缓冲区的内容。对 SPI_DR 寄存器执行的读操作将返回先前正确接收的数据。主器件后续发送的所有其它半字都将丢失。

要将 OVR 位清零，应首先对 SPI_DR 寄存器执行读操作，然后再对 SPI_SR 寄存器进行读访问。

帧错误标志 (FRE)

仅当 I2S 配置为从模式时，此标志才可由硬件置 1。如果外部主器件没有按照从器件期望的那样改变 WS 信号，则此标志将置 1。如果失去同步，要从此状态中恢复并将外部主器件与 I2S 从器件重新同步，应遵循如下步骤：

1. 关闭 I2S
2. 在 WS 线上检测到正确的电平时将其重新使能（WS 在 I2S 模式下为高电平，在 MSB 对齐、LSB 对齐或 PCM 模式下为低电平）。

主器件与从器件之间的同步失效可能是由于 SCK 通信时钟或 WS 帧同步信号线上存在噪音干扰。

如果 ERRIE 位置 1，可产生错误中断。读取状态寄存器时，同步失效标志 (FRE) 由软件清零。

13.4.9 I 2 S 中断

表 126 为 I 2 S 中断的列表。

表 126. I 2 S

中断事件	事件标志	使能控制位
发送缓冲区为空	TXE	TXEIE
接收缓冲区非空	RXNE	RXNEIE
上溢错误	OVR	ERRIE
下溢错误	UDR	
帧错误	FRE	ERRIE

13.4.10 DMA

DMA 的工作方式与 SPI 模式完全相同。对于 I 2 S 没有任何区别。只是由于不需要数据传输保护机制，因此 I 2 S 模式没有 CRC 功能。

13.5 SPI 和 I 2 S

有关寄存器说明中使用的缩写，请参见第 1.1 节 。

外设寄存器可支持半字（16 位）或字（32 位）访问。

13.5.1 SPI 控制寄存器 1 (SPI_CR1) （不用于 I 2 S 模式）

SPI control register 1

偏移地址：0x00

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIDI MODE	BIDI OE	CRC EN	CRC NEXT	DFF	RX ONLY	SSM	SSI	LSB FIRST	SPE	BR[2:0]			MSTR	CPOL	CPHA
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15 BIDIMODE ：双向通信数据模式使能 (Bidirectional data mode enable)

0：选择双线单向通信数据模式

1：选择单线双向通信数据模式

注意： 不适用于 I 2 S 模式

位 14 BIDIOE ：双向通信模式下的输出使能 (Output enable in bidirectional mode)

此位结合 BIDIMODE 位，用于选择双向通信模式下的传输方向
0：禁止输出（只接收模式）
1：使能输出（只发送模式）
注意：在主模式下，使用 MOSI 引脚；在从模式下，使用 MISO 引脚。
不适用于 I2S 模式

位 13 CRCEN ：硬件 CRC 计算使能 (Hardware CRC calculation enable)
0：禁止 CRC 计算
1：使能 CRC 计算
注意：为确保正确操作，只应在禁止 SPI （ SPE = “ 0 ”）时对此位执行写操作
不适用于 I2S 模式

位 12 CRCNEXT ：下一次传输 CRC (CRC transfer next)
0：数据阶段（无 CRC 阶段）
1：下一次传输为 CRC（CRC 阶段）
注意：当 SPI 配置为全双工或只发送模式时，只要最后一个数据写入 SPI_DR 寄存器，就必须对 CRCNEXT 执行写操作。
当 SPI 配置为只接收模式时，必须在接收到倒数第二个数据之后将 CRCNEXT 置 1 。当传输由 DMA 管理时，此位应保持清零状态。不适用于 I2S 模式

位 11 DFF ：数据帧格式 (Data frame format)
0：为发送/接收选择 8 位数据帧格式
1：为发送/接收选择 16 位数据帧格式
注意：为确保正确操作，只应在禁止 SPI （ SPE = “ 0 ”）时对此位执行写操作适用于 I2S 模式

位 10 RXONLY ：只接收 (Receive only)
此位结合 BIDIMODE 位，用于选择双线单向模式下的传输方向。此位也适用于多从模式系统，在此类系统中，不会访问特定从器件，也不会损坏访问的从器件的输出。
0：全双工（发送和接收）
1：关闭输出（只接收模式）
注意： 不适用于 I2S 模式

位 9 SSM ：软件从器件管理 (Software slave management)
当 SSM 位置 1 时，NSS 引脚输入替换为 SSI 位的值。
0：禁止软件从器件管理
1：使能软件从器件管理
注意： 不适用于 I2S 模式和 SPI TI 模式

位 8 SSI ：内部从器件选择 (Internal slave select)
仅当 SSM 位置 1 时，此位才有效。此位的值将作用到 NSS 引脚上，并忽略 NSS 引脚的 IO 值。
注意： 不适用于 I2S 模式和 SPI TI 模式

位 7 LSBFIRST ：帧格式 (Frame format)
0：先发送 MSB
1：先发送 LSB
注意：正在通信时不应更改此位。
不适用于 I2S 模式和 SPI TI 模式

位 6 SPE ：SPI 使能 (SPI enable)
0：关闭外设
1：使能外设
注意： 1- 不适用于 I2S 模式。
注意： 2- 关闭 SPI 时，请按照第 27.3.8 节：关闭 SPI 中所述的步骤操作。

位 5:3 BR[2:0] ：波特率控制 (Baud rate control)
000: f PCLK /2 100: f PCLK /32
001: f PCLK /4 101: f PCLK /64
010: f PCLK /8 110: f PCLK /128

011: f PCLK /16 111: f PCLK /256
注意：正在通信时不应更改这些位。
不适用于 I 2 S 模式

位 2 MSTR ： 主模式选择 (Master selection)
0：从配置
1：主配置
注意：正在通信时不应更改此位。
不适用于 I 2 S 模式

位 1 CPOL ： 时钟极性 (Clock polarity)
0：空闲状态时，SCK 保持低电平
1：空闲状态时，SCK 保持高电平
注意：正在通信时不应更改此位。
不适用于 I 2 S 模式和 SPI TI 模式

位 0 CPHA ： 时钟相位 (Clock phase)
0：从第一个时钟边沿开始采样数据
1：从第二个时钟边沿开始采样数据
注意：正在通信时不应更改此位。
不适用于 I 2 S 模式和 SPI TI 模式

13.5.2 SPI 控制寄存器 2 (SPI_CR2)

SPI control register 2
偏移地址：0x04
复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TXEIE	RXNEIE	ERRIE	FRF	Reserved	SSOE	TXDMAEN	RXDMAEN
								rw	rw	rw	rw		rw	rw	rw

位 15:8 保留，必须保持复位值。
位 7 TXEIE ： 发送缓冲区空中断使能 (Tx buffer empty interrupt enable)
0：屏蔽 TXE 中断
1：使能 TXE 中断。TXE 标志置 1 时产生中断请求。
位 6 RXNEIE ： 接收缓冲区非空中断使能 (RX buffer not empty interrupt enable)
0：屏蔽 RXNE 中断
1：使能 RXNE 中断。RXNE 标志置 1 时产生中断请求。
位 5 ERRIE ： 错误中断使能 (Error interrupt enable)
此位用于控制在错误状况发生时是否产生中断（SPI 模式中的 CRCERR、OVR、MODF，以及 UDR、OVR 和 FRE）。
0：屏蔽错误中断
1：使能错误中断
位 4 FRF ： 帧格式 (Frame format)
0：SPI Motorola 模式
1：SPI TI 模式
注意：不适用 于 I 2 S 模式
位 3 保留。由硬件强制为零。
位 2 SSOE ： SS 输出使能 (SS output enable)
0：在主模式下禁止 SS 输出，可在多主模式配置下工作
1：在主模式下使能 SS 输出，不能在多主模式环境下工作
注意： 不适用于 I 2 S 模式和 SPI TI 模式
位 1 TXDMAEN ： 发送缓冲区 DMA 使能 (Tx buffer DMA enable)

当此位置 1 时，每当 TXE 标志置 1 时，即产生 DMA 请求。
0：关闭发送缓冲区 DMA
1：使能发送缓冲区 DMA

位 0 RXDMAEN ：接收缓冲区 DMA 使能 (Rx buffer DMA enable)
当此位置 1 时，每当 RXNE 标志置 1 时，即产生 DMA 请求。
0：关闭接收缓冲区 DMA
1：使能接收缓冲区 DMA

13.5.3 SPI 状态寄存器 (SPI_SR)

SPI status register
偏移地址：0x08
复位值：0x0002

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							FRE	BSY	OVR	MODF	CRC ERR	UDR	CHS IDE	TXE	RXNE
							r	r	r	r	rc_w0	r	r	r	r

位 15:9 保留。由硬件强制为零。

位 8 FRE ：帧格式错误 (Frame format error)
0：没有帧格式错误
1：出现帧格式错误
此标志由硬件置 1，当读取 SPIx_SR 寄存器时由软件清零。
注意：当 SPI 在 TI 从模式或 I2S 从模式下工作时，使用此标志。

位 7 BSY ：忙标志 (Busy flag)
0：SPI（或 I2S）不繁忙
1：SPI（或 I2S）忙于通信或者发送缓冲区不为空
此标志由硬件置 1 和清零。

位 6 OVR ：上溢标志 (Overrun flag)
0：未发生上溢
1：发生上溢
此标志由硬件置 1，可由软件序列复位。

位 5 MODF ：模式故障 (Mode fault)
0：未发生模式故障
1：发生模式故障
此标志由硬件置 1，可由软件序列复位。
注意：不适用于 I2S 模式

位 4 CRCERR ：CRC 错误标志 (CRC error flag)
0：接收到的 CRC 值与 SPI_RXCRCR 值匹配
1：接收到的 CRC 值与 SPI_RXCRCR 值不匹配
此标志由硬件置 1，通过软件写入 0 来清零。
注意：不适用于 I2S 模式

位 3 UDR ：下溢标志 (Underrun flag)
0：未发生下溢
1：发生下溢
此标志由硬件置 1，可由软件序列复位。
注意：不适用于 SPI 模式

位 2 CHSIDE ：通道信息 (Channel side)
0：发送或接收左通道信息
1：发送或接收右通道信息

注意：不适用于 SPI 模式。在 PCM 模式下没有意义
位 1 TXE ： 发送缓冲区为空 (Transmit buffer empty)
0：发送缓冲区非空
1：发送缓冲区为空
位 0 RXNE ： 接收缓冲区非空 (Receive buffer not empty)
0：接收缓冲区为空
1：接收缓冲区非空

13.5.4 SPI 数据寄存器 (SPI_DR)

SPI data register
偏移地址：0x0C
复位值：0x0000

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

位 15:0 DR[15:0] ： 数据寄存器 (Data register)
已接收或者要发送的数据。
数据寄存器分为 2 个缓冲区，一个用于写入（发送缓冲区），一个用于读取（接收缓冲区）。对数据寄存器执行写操作时，数据将写入发送缓冲区，从数据寄存器执行读取时，将返回接收缓冲区中的值。
针对 SPI 模式的说明：
发送或接收的数据为 8 位或 16 位，具体取决于数据帧格式选择位（ SPI_CR1 寄存器中的 DFF ）。必须在使能 SPI 前进行此项选择，以确保操作正确。
对于 8 位数据帧，缓冲区为 8 位，只有寄存器的 LSB (SPI_DR[7:0]) 用于发送 / 接收。
在接收模式下，寄存器的 MSB (SPI_DR[15:8]) 强制为 0 。
对于 16 位数据帧，缓冲区为 16 位，整个寄存器 SPI_DR[15:0] 均用于发送 / 接收。

13.5.5 SPI CRC 多项式寄存器 (SPI_CRCPR) （不用于 I2S 模式）

SPI CRC polynomial register
偏移地址：0x10
复位值：0x0007

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

位 15:0 CRCPOLY[15:0] ： CRC 多项式寄存器 (CRC polynomial register)
此寄存器包含用于 CRC 计算的多项式。
CRC 多项式 (0007h) 是此寄存器的复位值。可根据需要配置另一个多项式。
注意：不适用于 I2S 模式。

13.5.6 SPI RX CRC 寄存器 (SPI_RXCRCR) （不用于 I2S 模式）

SPI RX CRC register
偏移地址：0x14
复位值：0x0000

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

位 15:0 RXCRC[15:0] ：接收 CRC 寄存器 (Rx CRC register)

使能 CRC 计算后，RxCRC[15:0] 位将包含后续接收字节在计算后所得到的 CRC 值。当 SPI_CR1 寄存器中的 CRCEN 位写入 1 时，此寄存器复位。CRC 通过 SPI_CRCPR 寄存器中编程的多项式连续计算。

数据帧格式设置为 8 位数据（SPI_CR1 的 DFF 位清零）时，仅考虑 8 个 LSB 位。CRC 计算依据任意 CRC8 标准进行。

选择 16 位数据帧格式（SPI_CR1 寄存器的 DFF 位置 1）时，考虑此寄存器的全部 16 个位。CRC 计算依据任意 CRC16 标准进行。

注意：当 BSY 标志置 1 时，读取此寄存器可能返回一个不正确的值。

不适用于 I2S 模式。

13.5.7 SPI TX CRC 寄存器 (SPI_TXCRCR) （不用于 I2S 模式）

SPI TX CRC register

偏移地址：0x18

复位值：0x0000

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

位 15:0 TXCRC[15:0] ：发送 CRC 寄存器 (Tx CRC register)

使能 CRC 计算后，TxCRC[7:0] 位将包含后续发送字节在计算后所得到的 CRC 值。当 SPI_CR1 寄存器中的 CRCEN 位写入 1 时，此寄存器复位。CRC 通过 SPI_CRCPR 寄存器中编程的多项式连续计算。

数据帧格式设置为 8 位数据（SPI_CR1 的 DFF 位清零）时，仅考虑 8 个 LSB 位。CRC 计算依据任意 CRC8 标准进行。

选择 16 位数据帧格式（SPI_CR1 寄存器的 DFF 位置 1）时，考虑此寄存器的全部 16 个位。CRC 计算依据任意 CRC16 标准进行。

注意：当 BSY 标志置 1 时，读取此寄存器可能返回一个不正确的值。

不适用于 I2S 模式。

13.5.8 SPI_I2S 配置寄存器 (SPI_I2SCFGR)

SPI_I2S configuration register

偏移地址：0x1C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				I2SMOD	I2SE	I2SCFG		PCMSYNC	Reserved	I2SSTD		CKPOL	DATLEN		CHLEN
				rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw

位 15:12 保留，必须保持复位值。

位 11 I2SMOD：I2S 模式选择 (I2S mode selection)

0：选择 SPI 模式

1：选择 I2S 模式

注意：应在 SPI 或 I2S 禁止时配置此位。

位 10 I2SE：I2S 使能 (I2S Enable)

0：关闭 I2S 外设

1：使能 I 2 S 外设
注意：不适用于 SPI 模式

位 9:8 I2SCFG：I2S 配置模式 (I2S configuration mode)
00：从模式 - 发送
01：从模式 - 接收
10：主模式 - 发送
11：主模式 - 接收
注意：应在 I 2 S 禁止时配置此位。
不适用于 SPI 模式

位 7 PCMSYNC：PCM 帧同步 (PCM frame synchronization)
0：短帧同步
1：长帧同步
注意：只有在 I2SSTD = 11 （使用 PCM 标准）时，此位才有意义
不适用于 SPI 模式

位 6 保留：由硬件强制为 0

位 5:4 I2SSTD：I2S 标准选择 (I2S standard selection)
00：I 2 S Philips 标准。
01：MSB 对齐标准（左对齐）
10：LSB 对齐标准（右对齐）
11：PCM 标准
注意：为确保正确运行，应在 I 2 S 关闭时配置这些位。

位 3 CKPOL：空闲状态的时钟电平 (Steady state clock polarity)
0：空闲状态时钟为低电平
1：空闲状态时钟为高电平
注意：为确保正确运行，应在 I 2 S 关闭时配置此位。
不适用于 SPI 模式

位 2:1 DATLEN：传输的数据长度 (Data length to be transferred)
00：16 位数据长度
01：24 位数据长度
10：32 位数据长度
11：不允许
注意：为确保正确运行，应在 I 2 S 关闭时配置这些位。
不适用于 SPI 模式。

位 0 CHLEN：通道长度（每个音频通道的位数）(Channel length (number of bits per audio channel))
0：16 位
1：32 位
只有在 DATLEN = 00 时，此位的写操作才有意义，否则无论填入何值，通道长度始终由硬件固定为 32 位。 不适用于 SPI 模式。
注意：为确保正确运行，应在 I 2 S 关闭时配置此位。

13.5.9 SPI_I 2 S 预分频器寄存器 (SPI_I2SPR)

SPI_I 2 S prescaler register

偏移地址：0x20

复位值：0000 0010 (0x0002)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						MCKOE	ODD	I2SDIV							
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:10 保留，必须保持复位值。

位 9 MCKOE：主时钟输出使能 (Master clock output enable)

0：禁止主时钟输出
1：使能主时钟输出
注意：应在 I2S 禁止时配置此位。只有在 I2S 为主模式时，才会使用此位。不适用于 SPI 模式。

位 8 ODD：预分频器的奇数因子 (Odd factor for the prescaler)
0：实际分频值为 = I2SDIV *2
1：实际分频值为 = (I2SDIV * 2)+1
注意：应在 I2S 禁止时配置此位。只有在 I2S 为主模式时，才会使用此位。

位 7:0 I2SDIV：I2S 线性预分频器 (I2S Linear prescaler)
I2SDIV [7:0] = 0 或 I2SDIV [7:0] = 1 为禁用值。
注意：应在 I2S 禁止时配置这些位。只有在 I2S 为主模式时，才会使用此位。

注意在 I2S_Standard_Phillips 模式下要配置 SPI2->SPI_I2SMD_OFFSET=0xcafe，以兼容 stc33f 格式。

13.5.10 SPI

下表显示了 SPI 寄存器映射和复位值。
有关寄存器边界地址的信息，请参见表 2。
表 127. SPI

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0X00	SPC_CR1	Reserved																BIDIMODE	BIDIOE	CRCEN	CRCNEXT	DFF	RXONLY	SSM	SSI	LSBFIRST	SPE	BR[2:0]			MSTR	CPOL	CPHA
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0X04	SPI_CR2	Reserved																								TXEIE	RXNEIE	ERRIE	FRF	Reserved	SSOE	TXDMAEN	RXDMAEN
	Reset value																									0	0	0	0		0	0	0
0x08	SPI_SR	Reserved																							FRE	BSY	OVR	MODF	CRCERR	UDR	CHSIDE	TXE	RXNE
	Reset value																								0	0	0	0	0	0	0	1	0
0X0C	SPI_DR	Reserved																DR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0X10	SPI_CRCPR	Reserved																CRCPOLY[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0X14	SPI_RXCRCR	Reserved																RxCRC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	SPI_RXCRCR	Reserved																TxCRC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0X1C	SPI_I2SCFGR	Reserved																				I2SMOD	I2SE	I2SCFG	PCMSYNC	Reserved	I2SSTD		CKPOL	DATLEN		CHLEN
	Reset value																					0	0	0	0		0	0	0	0	0	0	0

0X20	SPI_I2SPR	Reserved	MCKOE	ODD	I2SDIV								
	Reset value		0	0	0	0	0	0	0	1	0		

STC MCU

14 I2C 串行外设接口

14.1I2C 简介

I2C 总线接口用作微控制器和 I2C 串行总线之间的接口。它提供多主模式功能，可以控制所有 I2C 总线特定的序列、协议、仲裁和时序。它支持标准和快速模式。它可以用于多种用途，包括 CRC 生成和验证。
根据器件的不同，可利用 DMA 功能来减轻 CPU 的工作量。

14.2I2C 主要特性

- 并行总线/I2C 协议转换器
- 多主模式功能：同一接口既可用作主模式也可用作从模式
- I2C 主模式特性：
  - 时钟生成
  - 起始位和停止位生成
- I2C 从模式特性：
  - 可编程 I2C 地址检测
  - 双寻址模式，可对 2 个从地址应答
  - 停止位检测
- 7 位/10 位寻址以及广播呼叫的生成和检测
- 支持不同的通信速度：
  - 标准速度（高达 100 kHz）
  - 快速速度（高达 1400 kHz）
- 状态标志：
  - 发送/接收模式标志
  - 字节传输结束标志
  - I2C 忙碌标志
- 错误标志：
  - 主模式下的仲裁丢失情况
  - 地址/数据传输完成后的应答失败
  - 检测误放的起始位和停止位
  - 禁止时钟延长后出现的上溢/下溢
- 2 个中断向量：
  - 一个中断由成功的地址/数据字节传输事件触发
  - 一个中断由错误状态触发
- 可选的时钟延长
- 带 DMA 功能的 1 字节缓冲
- 可配置的 PEC（数据包错误校验）生成或验证：

STC MCU

— 在 Tx 模式下，可将 PEC 值作为最后一个字节进行传送
— 针对最后接收字节的 PEC 错误校验
注意： 上述某些特性可能不适用于某些产品。用户应参照产品数据手册来确定 I 2 C 接口实现所支持的特定特性。

14.3I 2 C 功能说明

除了接收和发送数据之外，此接口还可以从串行格式转换为并行格式，反之亦然。中断由软件使能或禁止。该接口通过数据引脚 (SDA) 和时钟引脚 (SCL) 连接到 I 2 C 总线。它可以连接到标准（高达 100 kHz）或快速（高达 1400 kHz）I 2 C 总线。

14.3.1模式选择

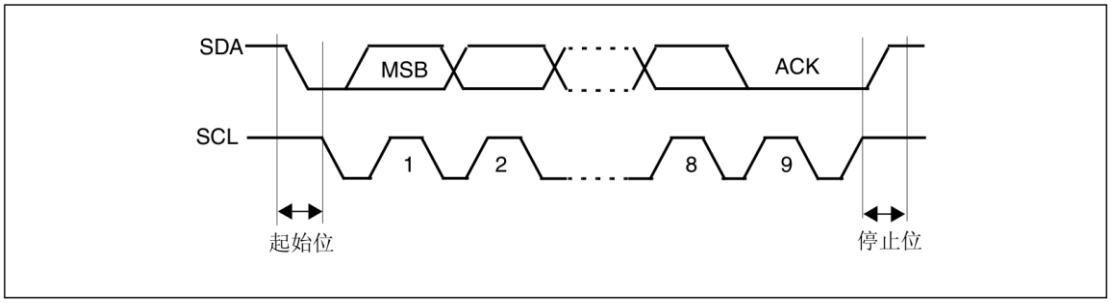
该接口在工作时可选用以下四种模式之一：
- 从发送器
- 从接收器
- 主发送器
- 主接收器

默认情况下，它以从模式工作。接口在生成起始位后会自动由从模式切换为主模式，并在出现仲裁丢失或生成停止位时从主模式切换为从模式，从而实现多主模式功能。

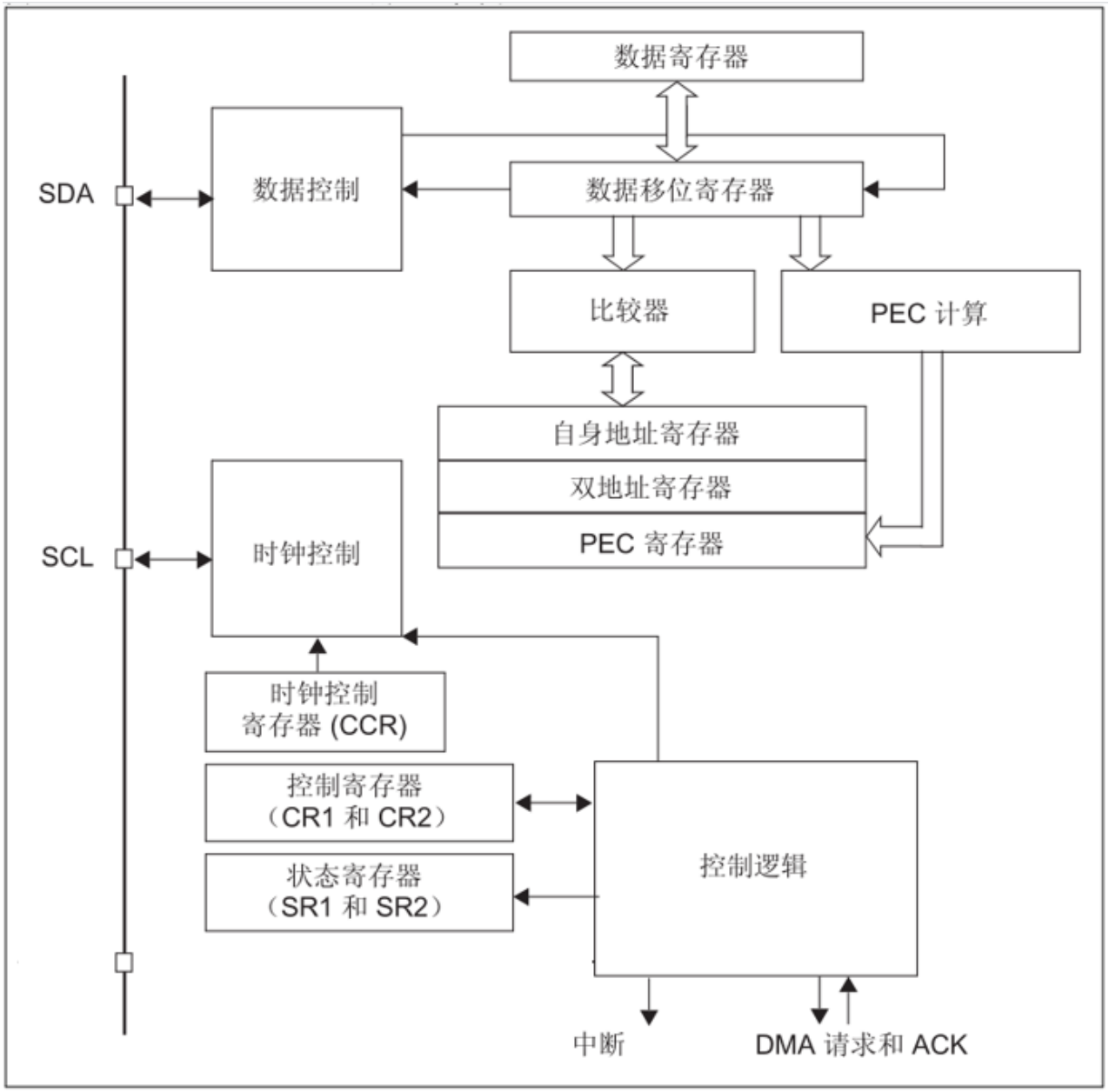
通信流程

在主模式下，I 2 C 接口会启动数据传输并生成时钟信号。串行数据传输始终是在出现起始位时开始，在出现停止位时结束。起始位和停止位均在主模式下由软件生成。
在从模式下，该接口能够识别其自身地址（7 或 10 位）以及广播呼叫地址。广播呼叫地址检测可由软件使能或禁止。
数据和地址均以 8 位字节传输，MSB 在前。起始位后紧随地址字节（7 位地址占据一个字节；10 位地址占据两个字节）。地址始终在主模式下传送。
在字节传输 8 个时钟周期后是第 9 个时钟脉冲，在此期间接收器必须向发送器发送一个应答位。请参见 图 238。

图 238. I 2 C 总线协议



应答位可由软件使能或禁止。I 2 C 接口地址（7 位/10 位双寻址模式和/或广播呼叫地址）可通过软件进行选择。
I 2 C 接口框图如 图 239 所示。
图 239. I 2 C



14.3.2I 2 C 从模式

默认情况下，I 2 C 接口在从模式下工作。要将工作模式由默认的从模式切换为主模式，需要生成一个起始位。

为了生成正确的时序，必须在 I2C_CR2 寄存器中对外设输入时钟进行编程。外设输入时钟频率的下限为：

- 标准模式下 2 MHz
- 快速模式下 4 MHz

检测到起始位后，便会立即接收到来自 SDA 线的地址并将其送到移位寄存器。之后，会将其与接口地址 (OAR1) 和 OAR2（如果 ENDUAL=1）或者广播呼叫地址（如果 ENGC = 1）进行比较。

注意： 在 10 位寻址模式下，比较对象还包括头序列 (11110xx0) ，其中， xx 表示该地址的两个最高有效位。

头或地址不匹配：接口会忽略它并等待下一个起始位。

头匹配（仅针对 10 位模式）：如果 ACK 位置 1，则接口会生成一个应答脉冲并等待 8 位从地址。
地址匹配：接口会依次：

- 发出应答脉冲（如果 ACK 位置 1）
- ADDR 位会由硬件置 1 并在 ITEVFEN 位置 1 时生成一个中断。
- 如果 ENDUAL=1，则软件必须读取 DUALF 位状态来核对哪些从地址进行了应答。

在 10 位模式下，完成地址序列接收后，从模式始终处于接收模式。在接收到重复起始位以及一个匹配地址位和最低有效位均置 1 的头序列 (11110xx1) 后，它会进入发送模式。TRA 位指示从设备是处于接收模式还是处于发送模式。

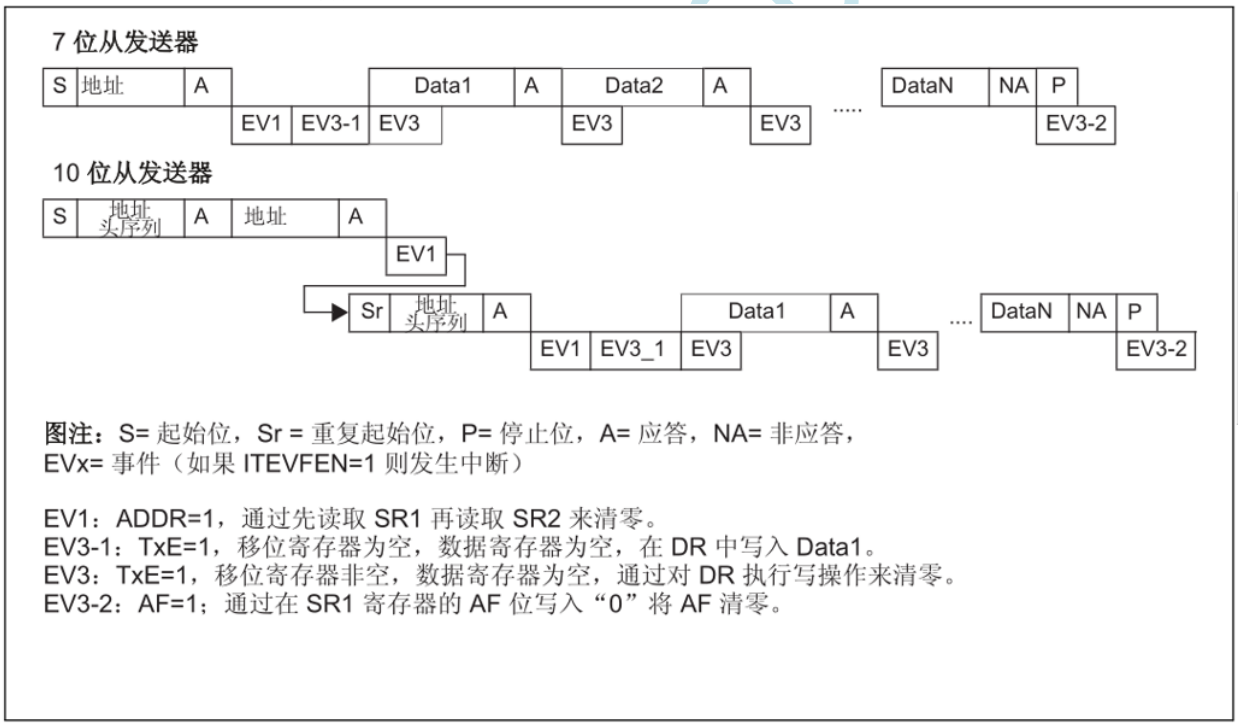
从发送器

在接收到地址并将 ADDR 清零后，从设备会通过内部移位寄存器将 DR 寄存器中的字节发送到 SDA 线。
从设备会延长 SCL 低电平时间，直到 ADDR 位清零且 DR 寄存器中填满待发送数据为止（请参见图 241 传输序列 EV1 EV3）。

接收到应答脉冲时：

- TxE 位会由硬件置 1 并在 ITEVFEN 和 ITBUFEN 位均置 1 时生成一个中断。

如果在下一次数据传输结束之前 TxE 位已置 1 但某些数据尚未写入 I2C_DR 寄存器，则 BTF 位会置 1，而接口会一直延长 SCL 低电平，直到通过软件对 I2C_SR1 读操作，以及对 I2C_DR 写操作后，把 BTF 清零为止。
图 241. 从发送器的传输序列图



1. EV1 和 EV3_1 事件可延长 SCL 低电平时间，直到相应的软件序列结束为止。
2. 如果软件序列在下一个字节传输结束之前未能完成，EV3 事件会延长 SCL 低电平时间。

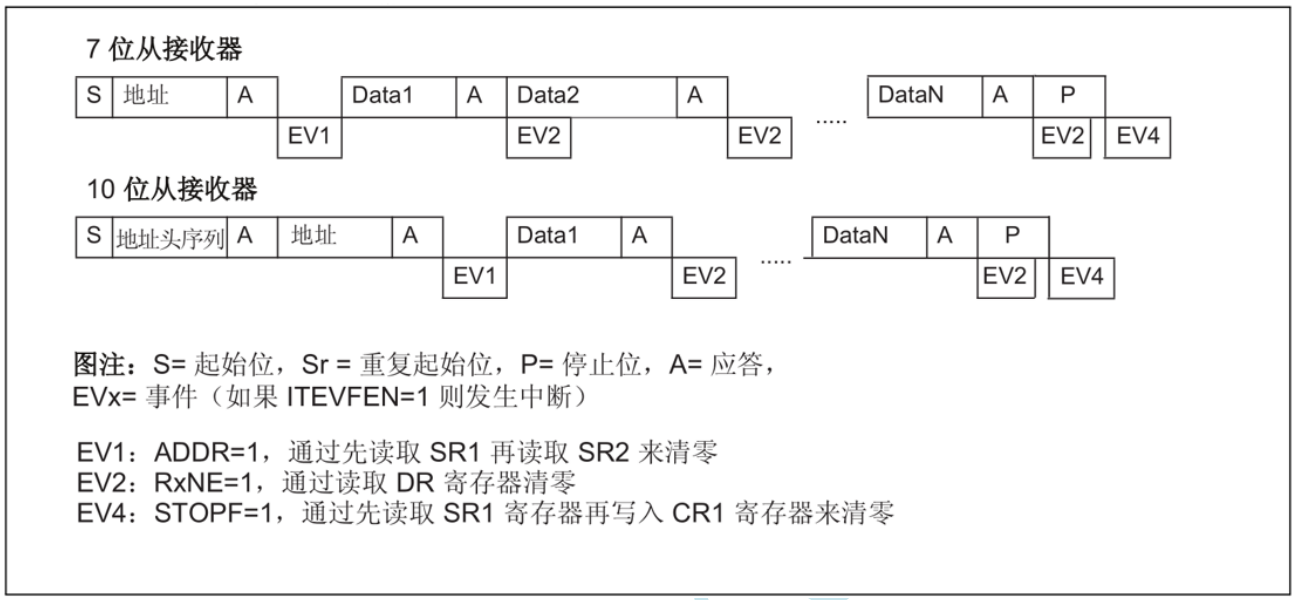
从接收器

在接收到地址并将 ADDR 位清零后，从设备会通过内部移位寄存器接收 SDA 线中的字节并将其保存到 DR 寄存器。在每个字节接收完成后，接口都会依次：

- 发出应答脉冲（如果 ACK 位置 1）

● RxNE 位会由硬件置 1 并在 ITEVFEN 和 ITBUFEN 位均置 1 时生成一个中断。
如果在下一次数据接收结束之前 RxNE 位已置 1 但 DR 寄存器中的数据尚未读取，则 BTF 位会置 1，而接口会一直延长 SCL 低电平，直到软件通过读取 I2C_DR 寄存器来把 BTF 清零（请参见 图 242 传输序列）。

图 242. 从接收器的传输序列图



1. EV1 事件可延长 SCL 低电平时间，直到相应的软件序列结束为止。
2. 如果软件序列在下一个字节接收结束之前未能完成，EV2 事件会延长 SCL 低电平时间。
3. 检查了 SR1 寄存器内容之后，用户应针对各个被置位的标志执行完整的清零序列。
对于 ADDR 和 STOPF 标志，需要在 I2C 中断程序中执行以下序列：
对 SR1 执行读操作
if (ADDR == 1) {对 SR1 执行读操作；对 SR2 执行读操作}
if (STOPF == 1) {对 SR1 执行读操作；对 CR1 执行写操作}
这样做的目的是为了确保 ADDR 和 STOPF 这两个标志都清零（如果二者均为被置位）。

关闭从设备通信

传输完最后一个数据字节之后，主设备会生成一个停止位。接口会检测此条件并：
● 将 STOPF 位置 1 并在 ITEVFEN 位置 1 时生成一个中断。
通过先读取 SR1 寄存器然后写入 CR1 寄存器的方式将 STOPF 位清零（请参见图 242 ：从接收器的传输序列图 EV4）。

14.3.3I 2 C 主模式

在主模式下，I 2 C 接口会启动数据传输并生成时钟信号。串行数据传输始终是在出现起始位时开始，在出现停止位时结束。只要通过 START 位在总线上生成了起始位，即会选中主模式。

在主模式中要求执行以下序列。

- 在 I2C_CR2 寄存器中对外设输入时钟进行编程，以生成正确的时序
- 配置时钟控制寄存器
- 配置上升时间寄存器
- 对 I2C_CR1 寄存器进行编程，以便使能外设
- 将 I2C_CR1 寄存器的 START 位置 1，以生成起始位

外设输入时钟频率的下限为：

- 标准模式下 2 MHz
- 快速模式下 4 MHz

起始位

起始位 START 位置 1 后，接口会在 BUSY 位清零后生成一个起始位并切换到主模式（M/SL 位置 1）。

注意： 在主设备下， START 位置 1 后，接口会在当前字节传输结束后生成一个重复起始位。起始位发出之后：

● SB 位会由硬件置 1 并在 ITEVFEN 位置 1 时生成一个中断。
接下来主设备会等待软件对 SR1 执行读操作，然后把从设备地址写入 DR 寄存器（请参见 图 243 和 图 244 传输序列 EV5）。

从地址传输

接下来从地址会通过内部移位寄存器发送到 SDA 线。

● 在 10 位寻址模式中，发送头序列会产生以下事件：

— ADD10 位会由硬件置 1 并在 ITEVFEN 位置 1 时生成一个中断。

接下来主设备会等待软件读取 SR1 ，然后把第二个地址字节写入 DR 寄存器（请参见图 243 和 图 244 传输序列）。

— ADDR 位会由硬件置 1 并在 ITEVFEN 位置 1 时生成一个中断。

接下来主设备会等待对 SR1 寄存器执行读操作，然后对 SR2 寄存器执行读操作（请参见 图 243 和 图 244 传输序列）。

● 在 7 位寻址模式下，会发送一个地址字节。
地址字节被发出后，

— ADDR 位会由硬件置 1 并在 ITEVFEN 位置 1 时生成一个中断。

接下来主设备会等待对 SR1 寄存器执行读操作，然后对 SR2 寄存器执行读操作（请参见 图 243 和 图 244 传输序列）。

主设备会根据发送的从地址字节 LSB 来决定是进入发送模式还是接收模式。

● 在 7 位寻址模式下，
— 要进入发送模式，主设备会发送从地址并将 LSB 复位。
— 要进入接收模式，主设备会发送从地址并将 LSB 置 1。

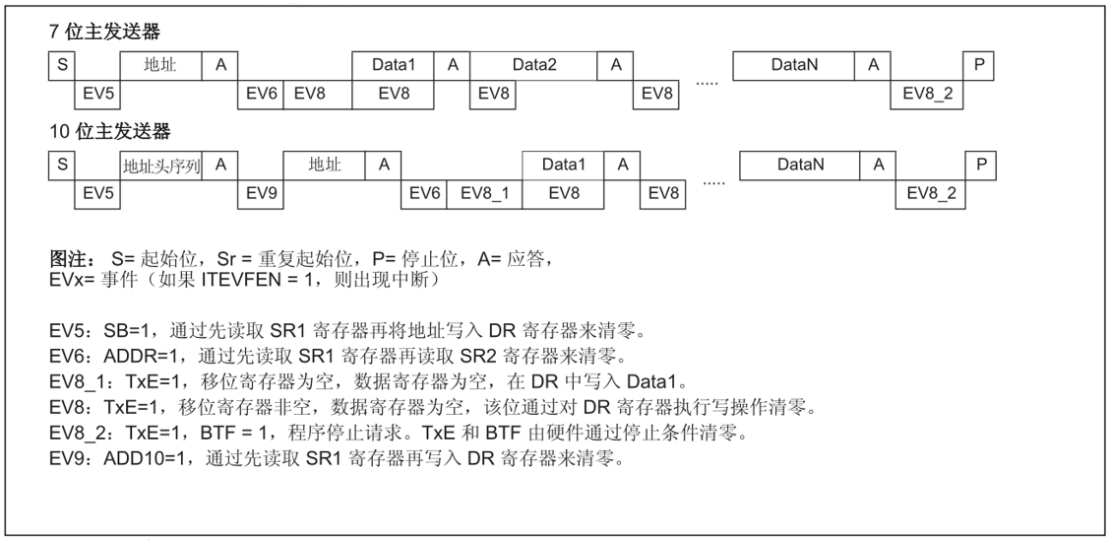
● 在 10 位寻址模式下，
— 要进入发送模式，主设备会先发送头序列 (11110xx0)，然后发送从地址（其中 xx 表示该地址的两个最高有效位）。

— 要进入接收模式，主设备会先发送头序列 (11110xx0)，然后发送从地址。接下来会发送一个重复起始位，然后再发送头序列 (11110xx1)（其中 xx 表示地址的两个最高有效位）。

TRA 位指示主设备是处于接收模式还是处于发送模式。

主发送器

在发送出地址并将 ADDR 清零后，主设备会通过内部移位寄存器将 DR 寄存器中的字节发送到 SDA 线。主设备会一直等待，直到首个数据字节被写入 I2C_DR 为止（请参见 图 243 传输序列 EV8_1）。
接收到应答脉冲后，TxE 位会由硬件置 1 并在 ITEVFEN 和 ITBUFEN 位均置 1 时生成一个中断。
如果在上一次数据传输结束之前 TxE 位已置 1 但数据字节尚未写入 DR 寄存器，则 BTF 位会置 1，而接口会一直延长 SCL 低电平，等待 I2C_DR 寄存器被写入，以将 BTF 清零。
结束通信
当最后一个字节写入 DR 寄存器后，软件会将 STOP 位置 1 以生成一个停止位（请参见 图 243 传输序列 EV8_2）。接口会自动返回从模式（M/SL 位清零）。
注意： 当 TxE 或 BTF 中的任何一个置 1 时，应在 EV8_2 事件期间对停止位进行编程。
图 243. 主发送器的传输序列图



1. EV5、EV6、EV9、EV8_1 和 EV8_2 事件可延长 SCL 低电平时间，直到相应的软件序列结束为止。
2. 如果软件序列在下一个字节传输结束之前未能完成，EV8 事件会延长 SCL 低电平时间。

主接收器

完成地址传输并将 ADDR 位清零后，I 2 C 接口会进入主接收模式。在此模式下，接口会通过内部移位寄存器接收 SDA 线中的字节并将其保存到 DR 寄存器。在每个字节传输结束后，接口都会依次：

1. 发出应答脉冲（如果 ACK 位置 1）
2. RxNE 位置 1 并在 ITEVFEN 和 ITBUFEN 位均置 1 时生成一个中断（参见 图 244 传输序列 EV7）。

如果在上一次数据接收结束之前 RxNE 位已置 1 但 DR 寄存器中的数据尚未读取，则 BTF 位会由硬件置 1，而接口会一直延长 SCL 低电平，等待 I2C_DR 寄存器被写入，以将 BTF 清零。

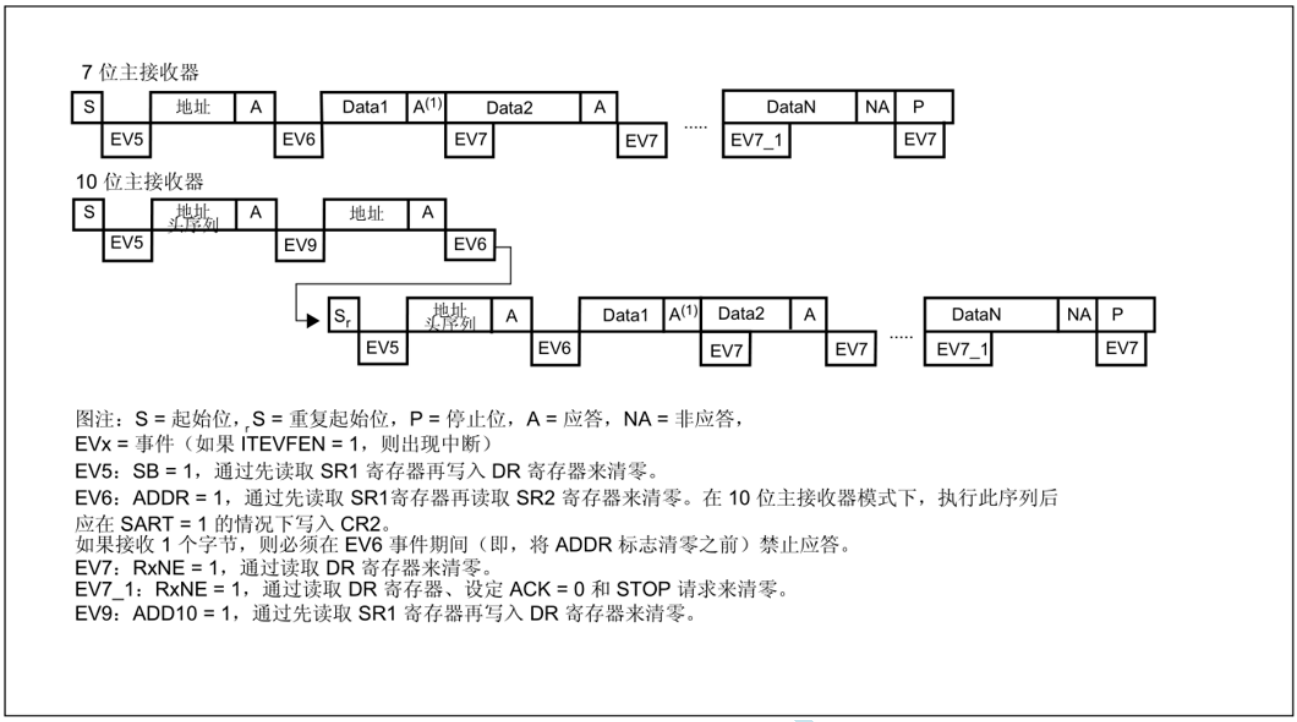
结束通信

主设备会针对自从设备接收的最后一个字节发送 NACK。在接收到此 NACK 之后，从设备会释放对 SCL 和 SDA 线的控制。随后，主设备可发送一个停止位/重复起始位。

1. 为了在最后一个接收数据字节后生成非应答脉冲，必须在读取倒数第二个数据字节后（倒数第二个 RxNE 事件之后）立即将 ACK 位清零。
2. 要生成停止位/重复起始位，软件必须在读取倒数第二个数据字节后（倒数第二个 RxNE 事件之后）将 STOP/START 位置 1。
3. 在只接收单个字节的情况下，会在 EV6 期间（在 ADDR 标志清零之前）禁止应答并在 EV6 之后生成停止位。

生成停止位后，接口会自动返回从模式（M/SL 位清零）。

图 244. 主接收器的传输序列图



图注：S = 起始位，Sr = 重复起始位，P = 停止位，A = 应答，NA = 非应答，
EVx = 事件（如果 ITEVFEN = 1，则出现中断）
EV5：SB = 1，通过先读取 SR1 寄存器再写入 DR 寄存器来清零。
EV6：ADDR = 1，通过先读取 SR1寄存器再读取 SR2 寄存器来清零。在 10 位主接收器模式下，执行此序列后应在 SART = 1 的情况下写入 CR2。
如果接收 1 个字节，则必须在 EV6 事件期间（即，将 ADDR 标志清零之前）禁止应答。
EV7：RxNE = 1，通过读取 DR 寄存器来清零。
EV7_1：RxNE = 1，通过读取 DR 寄存器、设定 ACK = 0 和 STOP 请求来清零。
EV9：ADD10 = 1，通过先读取 SR1 寄存器再写入 DR 寄存器来清零。

1. 如果接收单个字节，则为 NA。
2. EV5、EV6 和 EV9 事件可延长 SCL 低电平时间，直到相应的软件序列结束为止。
3. 如果软件序列在下一个字节接收结束之前未能完成，EV7 事件会延长 SCL 低电平时间。
4. EV7_1 软件序列必须在当前字节传输的 ACK 脉冲之前结束。

如果 EV7_1 软件序列未能在当前字节传输的 ACK 脉冲之前结束，建议采用下列步骤。
必须遵循以下步骤以确保：

- 在结束上一个数据接收之前及时将 ACK 位复位
- 在上一次数据接收后将 STOP 位置位且不接收补充数据。

针对 2 字节的接收：

- 等待 ADDR = 1（SCL 保持低电平，直到 ADDR 标志清零）
- 将 ACK 复位，POS 置位
- 将 ADDR 标志清零
- 等待 BTF = 1（数据 1 在 DR 中，数据 2 在移位寄存器中，SCL 保持低电平，直到读取了数据 1 为止）
- 将 STOP 置位
- 读取数据 1 和数据 2

针对 N > 2 的字节接收，从 N-2 数据接收开始

- 等待 BTF = 1（数据 N-2 在 DR 中，数据 N-1 在移位寄存器中，SCL 保持低电平，直到读取了数据 N-2 为止）
- 将 ACK 复位
- 读取数据 N-2
- 等待 BTF = 1（数据 N-1 在 DR 中，数据 N 在移位寄存器中，SCL 保持低电平，直到读取了数据 N-1 为止）
- 将 STOP 置位
- 读取数据 N-1 和数据 N

14.3.4错误条件

以下错误条件可能导致通信失败。

总线错误 (BERR)

当 I2C 接口在传输地址或数据期间检测到外部停止位或起始位时，会出现此错误。在这种情况下：

● BERR 位置 1 并在 ITERREN 位置 1 时生成一个中断
● 在从模式下：硬件会丢弃数据并释放数据线：
— 在误放起始位的情况下，从模式会认为它是重复起始位并会等待一个地址或停止位
— 在误放停止位的情况下，从模式会按停止位操作且硬件会释放数据线
● 在主模式下：不会释放数据线且不会影响当前传输状态。由软件决定是否中止当前传输

应答失败 (AF)

当接口检测到未应答脉冲会出现此错误。在这种情况下：
● AF 位置 1 并在 ITERREN 位置 1 时生成一个中断
● 接收到 NACK 的发送器必须复位通信：
— 在从模式下：硬件释放数据线
— 在主模式下：必须由软件生成一个停止位或重复起始位

仲裁丢失 (ARLO)

当 I2C 接口检测到仲裁丢失时会出现此错误。在这种情况下，
● ARLO 位会由硬件置 1（并在 ITERREN 位置 1 时生成一个中断）
● I2C 接口会自动返回从模式（M/SL 位清零）。I2C 失去仲裁时，将无法在本次传输中对自身从地址进行应答，但是能够在赢得仲裁的主设备发出重复起始位后对自身设备进行应答。
● 硬件释放数据线

上溢/ 下溢错误 (OVR)

当时钟延长已禁止且 I 2 C 接口正在接收数据时，从模式中可能出现上溢错误。接口已经收到一个字节 (RxNE=1)，但是收到下一个字节之前 DR 中的数据未被读走。在这种情况下，
● 会丢失接收的最后一个字节。
● 在出现上溢错误时，软件应将 RxNE 位清零，而发送器应重新发送最后一个字节。

当时钟延长已禁止且 I 2 C 接口正在发送数据时，从模式中可能出现下溢错误。下一个字节的时钟信号到来之前，从设备还未把下一个要发送的数据写进 DR（TxE=1）。在这种情况下，
● 会再次发送 DR 寄存器中的字节
● 用户应确保在出现下溢错误期间从接收器端接收到的数据被丢弃，并确保在 I2C 总线标准指定的时钟低电平时间内写入下一个字节。

对于要传送的首个字节，必须在 ADDR 清零之后且出现首个 SCL 上升沿之前将其写入 DR 寄存器。否则，接收器必须丢弃首个数据。

14.3.5 SDA/SCL 线控制

● 如果时钟延长已使能：
— 发送模式：如果 TxE=1 且 BTF=1：接口会在发送数据之前保持时钟线为低电平，以等待微控制器将字节写入数据寄存器（缓冲寄存器和移位寄存器均为空）。
— 接收模式：如果 RxNE=1 且 BTF=1：接口会在接收数据之后保持时钟线为低电平，以等待微控制器将字节读入数据寄存器（缓冲寄存器和移位寄存器均已满）。
● 如果从模式中的时钟延长已禁止：
— RxNE=1，且在下一个数据接收完成之前还未读走 DR 中的数据就出现上溢错误。会丢失接收的最后一个字节。
— TxE=1，且在下一个数据的始终到来之前还未把发送数据写到 DR 中，就出现下溢错误。会再次发送同一字节。
— 不会对写冲突进行管理。

14.3.6 DMA 请求

DMA 请求（在使能后）仅用于数据传输。当发送数据寄存器变空以及接收数据寄存器变满时会生成 DMA 请求。进行 I2C 数据传输之前，必须先初始化并使能 DMA。I2C_CR2 寄存器中的 DMAEN 位必须在 ADDR 事件之前置 1。在主模式或从模式下，如果已使能时钟延长，DMAEN 位也可以在 ADDR 事件期间于 ADDR 标志清零之前置 1。结束当前字节传输之前，必须发出 DMA 请求。当传输的数据量达到相应 DMA 通道编程设定的值时，DMA 控制器会发送一个结束传输 EOT 信号给 I 2 C 接口，并生成一个传输完成中断（如果已使能）：
● 主发送器：在 EOT 中断后的中断程序中，禁止 DMA 请求，然后在等到 BTF 事件后设置

停止位。

● 主接收器

— 当要接收的字节数等于或大于二时，DMA 控制器会在收到倒数第二个数据字节（第 N - 1 个数据时）发送一个硬件信号 EOT_1。如果 I2C_CR2 寄存器中的 LAST 位置 1，I 2 C 会在 EOT_1 后的下一个字节之后自动发送一个 NACK。用户可在 DMA 传输完成中断（如果已使能）程序中生成停止位。

— 当必须接收单个字节时：必须在 EV6 事件期间于 ADDR 标志清零之前对 NACK 进行编程，即当 ADDR=1 时编程设定 ACK=0。接下来，用户可在 ADDR 标志清零之后或者在执行 DMA 传输完成中断程序时编程设定停止位。

使用 DMA 进行发送

将 I2C_CR2 寄存器中的 DMAEN 位置 1 可以使能 DMA 模式进行发送。当 TXE 位置 1 时，数据将由 DMA 从预置的存储区装载进 I2C_DR 寄存器。要映射一个 DMA 通道以便进行 I 2 C 发送，请按以下步骤操作：其中的 x 表示通道编号。

1. 设置 DMA_CPARx 寄存器中的 I2C_DR 寄存器地址。每次发生 TxE 事件后，数据都会从存储器移动到此地址。
2. 设置 DMA_CMARx 寄存器中的存储器地址。每次发生 TxE 事件后，数据都会从此存储器加载到 I2C_DR。
3. 在 MA_CNDTRx 寄存器中配置要传输的总字节数。在每次 TxE 事件后，此值都会递减。
4. 使用 DMA_CCRx 寄存器中的 PL[0:1] 位来配置通道优先级。
5. 在完成半数传输或全部传输（取决于应用的需求）之后，将 DMA_CCRx 寄存器中的 DIR 位置 1 并配置中断。
6. 将 DMA_CCRx 寄存器中的 EN 位置 1 以激活通道。

当传输的数据量达到 DMA 控制器中编程设定的值时，DMA 控制器会发送一个结束传输 EOT/EOT_1 信号给 I 2 C 接口，而 DMA 会在 DMA 通道中断向量上生成一个中断（如果已使能）：

注意： 如果使用 DMA 进行传送，请勿使能 I2C_CR2 寄存器中的 ITBUFEN 位。

使用 DMA 进行接收

将 I2C_CR2 寄存器中的 DMAEN 位置 1 可以使能 DMA 模式进行接收。接收数据字节时，数据会从 I2C_DR 寄存器加载到使用 DMA 外设配置的存储区中（参见 DMA 规范）。要映射一个 DMA 通道以便进行 I 2 C 接收，请按以下步骤操作：其中的 x 表示通道编号。

1. 设置 DMA_CPARx 寄存器中的 I2C_DR 寄存器地址。每次发生 RxNE 事件后，数据都会从此地址移动到存储器。
2. 设置 DMA_CMARx 寄存器中的存储器地址。每次发生 RXNE 事件后，数据都会从 I2C_DR 寄存器加载到此存储区。
3. 在 MA_CNDTRx 寄存器中配置要传输的总字节数。在每次 RxNE 事件后，此值都会递减。
4. 使用 DMA_CCRx 寄存器中的 PL[0:1] 位来配置通道优先级。
5. 在完成半数传输或全部传输（取决于应用的需求）之后，将 DMA_CCRx 寄存器中的 DIR 位重新置 1 并配置中断。
6. 将 DMA_CCRx 寄存器中的 EN 位置 1 以激活通道。

当传输的数据量达到 DMA 控制器中编程设定的值时，DMA 控制器会发送一个结束传输 EOT/EOT_1 信号给 I 2 C 接口，而 DMA 会在 DMA 通道中断向量上生成一个中断（如果已使能）：

注意： 如果使用 DMA 进行接收，请勿使能 I2C_CR2 寄存器中的 ITBUFEN 位。

14.3.7数据包错误校验

PEC 计算器的应用目的是提高通信的可靠性。PEC 对各个位使用 CRC-8 多项式 C(x) = x 8 + x 2 + x + 1 公式进行串行计算。

● 将 I2C_CR1 寄存器中的 ENPEC 位置 1 即可使能 PEC 计算。PEC 是针对所有消息字节（包括地址和 R/W 位）的 CRC-8 计算。

— 在发送过程中：在与最后一个字节对应的 TxE 事件发生后，将 I2C_CR1 寄存器中的 PEC 传输位置 1。PEC 会在最后一个传输的字节之后进行传送。

— 在接收过程中：在与最后一个字节对应的 RxNE 事件发生之后，将 I2C_CR1 寄存器中的 PEC 位置 1，以便接收器在接收到的下一个字节不等于内部计算的 PEC 时发送一个 NACK。在主接收器中，无论校验结果如何，PEC 后都将发送 NACK。

在从模式下，必须在 CRC 接收的 ACK 之前设置 PEC。在主模式下，必须在复位 ACK 时设置 PEC 位。
● I2C_SR1 寄存器中还提供 PECERR 错误标志/中断。
● 如果 DMA 和 PEC 计算均已使能：
— 在发送过程中：当 I 2 C 接口接收来自 DMA 控制器的 EOT 信号时，会在最后一个字节之后自动发送 PEC。
— 在接收过程中：当 I 2 C 接口接收来自 DMA 控制器的 EOT_1 信号时，会自动将下一个字节视为 PEC 并对其进行校验。在 PEC 接收之后会生成一个 DMA 请求。
● 为了允许进行中间 PEC 传输，可使用 I2C_CR2 寄存器中的一个控制位（LAST 位）来确定它是否真的是最后一个 DMA 传输。如果确实是主接收器的最后一个 DMA 请求，则会在接收最后一个字节后自动发送一个 NACK。
● PEC 计算会因仲裁丢失而失效。

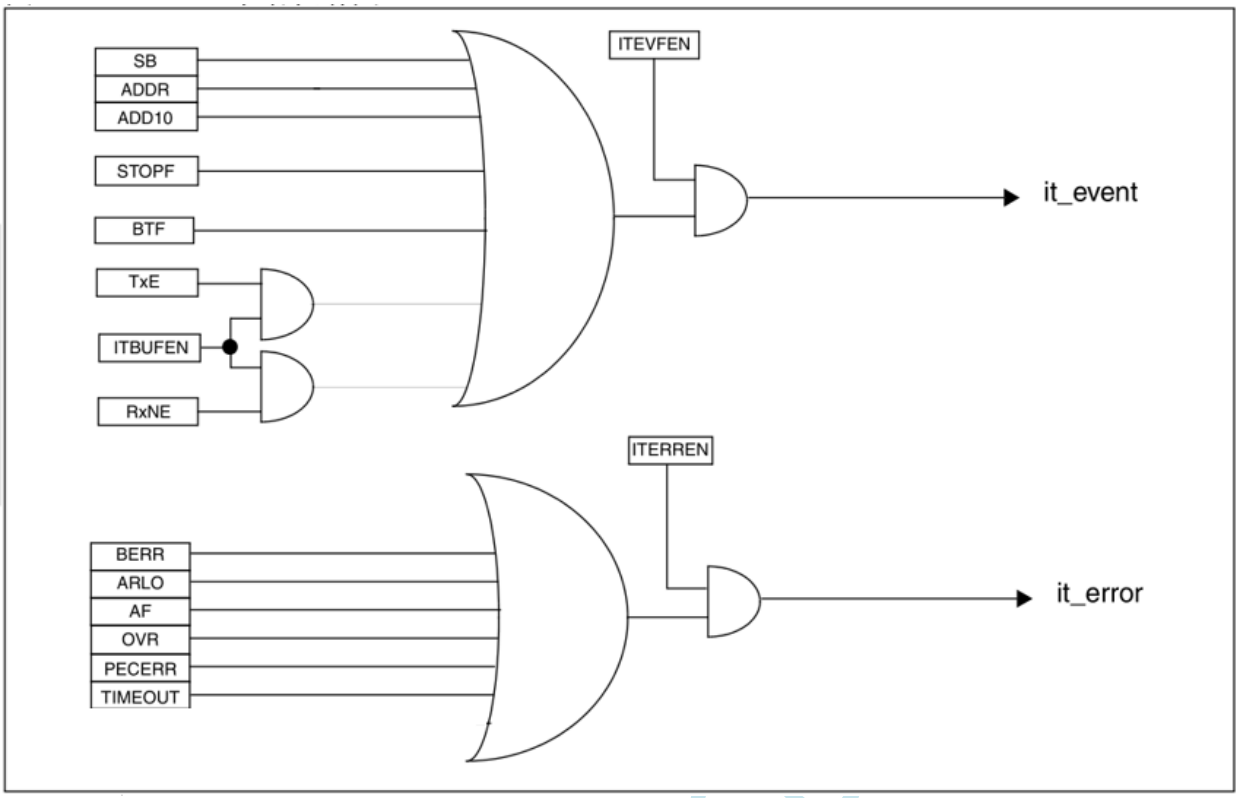
14.4I 2 C 中断

下表列出了 I 2 C 中断请求列表。
表 105. I 2 C

中断事件	事件标志	使能控制位
发送起始位（主模式）	SB	ITEVFEN
地址已发送（主模式）或地址匹配（从模式）	ADDR	
10 位地址的头段已发送（主模式）	ADD10	
已收到停止位（从模式）	STOPF	
完成数据字节传输	BTF	
接收缓冲区非空	RxNE	ITEVFEN 和 ITBUFEN
发送缓冲区为空	TxE	
总线错误	BERR	ITERREN
仲裁丢失（主模式）	ARLO	
应答失败	AF	
上溢/下溢	OVR	
PEC 错误	PECERR	
超时/Tlow 错误	TIMEOUT	

注意： SB 、 ADDR 、 ADD10 、 STOPF 、 BTF 、 RxNE 和 TxE 通过逻辑或映射到同一个中断通道上。BERR 、 ARLO 、 AF 、 OVR 、 PECERR 、 TIMEOUT 通过逻辑或映射到同一个中断通道上。
图 245. I 2 C 中断映射图

STCMCU



14.5I 2 C 寄存器

有关寄存器说明中使用的缩写，请参见第 1.1 节 。
外设寄存器可支持半字（16 位）或字（32 位）访问。

14.5.1I 2 C 控制寄存器 1 (I2C_CR1)

I 2 C Control register 1
偏移地址：0x00
复位值：0x0000

15	14 13	12	11	10	9	8	7	6	5	4	3 2 1	0
SWRST	Reserved	PEC	POS	ACK	STOP	START	NO STRETCH	ENGC	ENPEC	ENARP	Reserved	PE
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

位 15 SWRST：软件复位 (Software reset)
当置 1 时，I2C 处于复位状态。在复位此位之前，确保 I2C 线已释放且总线空闲。0：I 2 C 外设未处于复位状态
1：I 2 C 外设处于复位状态
注意：当出现错误或锁定状态后，可使用此位重新初始化外设。例如，如果 BUSY 位已置 1 但因母线干扰而不能复位，则可使用 SWRST 位退出此状态。
位 14 保留，必须保持复位值
位 13 保留，必须保持复位值
位 12 PEC：数据包错误校验 (Packet error checking)
此位由软件置 1 和清零，并可在 PEC 传输完成时由硬件清零，或者在 PE=0 时或在检测到起始位或停止位时由硬件清零。

0：不传输 PEC
1：PEC 传输（在 Tx 或 Rx 模式下）
注意： PEC 计算会因仲裁丢失而失效。

位 11 POS：应答/PEC 位置 (Acknowledge/PEC Position)（针对接收数据）
此位由软件置 1 和清零，并可在 PE=0 时由硬件清零。
0：ACK 位控制移位寄存器中当前正在接收的字节的 (N)ACK。PEC 位指示移位存器中的当前字节是一个 PEC。
1：ACK 位控制移位寄存器中要接收的下一个字节的 (N)ACK。PEC 位指示移位寄存器的下一个字节是一个 PEC。
注意： POS 位只能用于主设备接收 2 个字节时。它必须在数据开始接收之前进行配置，如第 654 页的主接收器一节中建议的 2 字节接收步骤所述。

位 10 ACK：应答使能 (Acknowledge enable)
此位由软件置 1 和清零，并可在 PE=0 时由硬件清零。
0：不返回应答
1：在接收一个字节（匹配地址或数据）之后返回应答

位 9 STOP：生成停止位
该位由软件置 1 和清零，也可在检测到停止位时由硬件清零，在检测到超时错误时由硬件置 1。
在主模式下：
0：不生成停止位。
1：在传输当前字节或发送当前起始位后生成停止位。
在从模式下：
0：不生成停止位。
1：完成当前字节传输后释放 SCL 和 SDA 线。

位 8 START：生成起始位
此位由软件置 1 和清零，并可在起始位发送完成后或 PE=0 时由硬件清零。
在主模式下：
0：不生成起始位
1：生成重复起始位
在从模式下：
0：不生成起始位
1：在总线空闲时生成起始位

位 7 NOSTRETCH：禁止时钟延长 (Clock stretching disable)（从模式）
在从模式下，当 ADDR 或 BTF 标志置 1 时，此位用于禁止时钟延长，直到软件将其复位为止。
0：使能时钟延长
1：禁止时钟延长

位 6 ENGC：广播呼叫使能 (General call enable)
0：禁止广播呼叫。不对地址 00h 应答。
1：使能广播呼叫。对地址 00h 应答。

位 5 ENPEC ：PEC 使能 (PEC enable)
0：禁止 PEC 计算
1：使能 PEC 计算

位 4 ENARP：ARP 使能 (ARP enable)

位 3 保留，必须保持复位值

位 2 保留，必须保持复位值

位 1 保留，必须保持复位值

位 0 PE：外设使能 (Peripheral enable)
0：禁止外设
1：使能外设

注意：如果此位在通信进行过程中复位，在结束本次通信后会带 IDLE 状态，外设被禁止。
由于通信结束时 PE=0 ，所有位均会复位。

STC MCU

在主模式下，此位不能在通信结束之前复位。

注意： STOP 、 START 或 PEC 位置 1 之后，在硬件将该位清零之前，软件不能对 I2C_CR1 执行任何写操作。否则，可能会再次发出 STOP 、 START 或 PEC 置位请求。

14.5.2I 2 C 控制寄存器 2 (I2C_CR2)

I 2 C Control register 2
偏移地址：0x04
复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			LAST	DMA EN	ITBUF EN	ITEVT EN	ITERR EN	Reserved		FREQ[5:0]					
			rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

位 15:13 保留，必须保持复位值
位 12 LAST：最后一次 DMA 传输 (DMA last transfer)
0：下一个 DMA EOT 不是最后一次传输
1：下一个 DMA EOT 是最后一次传输
注意：此位用于主接收模式，可对最后接收的数据生成 NACK 。
位 11 DMAEN：DMA 请求使能 (DMA requests enable)
0：禁止 DMA 请求
1：当 TxE=1 或 RxNE =1 时使能 DMA 请求
位 10 ITBUFEN：缓冲中断使能 (Buffer interrupt enable)
0：TxE = 1 或 RxNE = 1 时不生成任何中断。
1：TxE = 1 或 RxNE = 1 时生成事件中断（与 DMAEN 状态无关）
位 9 ITEVTEN：事件中断使能 (Event interrupt enable)
0：禁止事件中断
1：使能事件中断
满足以下条件时将生成此中断：
—SB = 1（主模式）
—ADDR = 1（主/从模式）
—ADD10= 1（主模式）
—STOPF = 1（从模式）
—BTF = 1，无 TxE 或 RxNE 事件
—ITBUFEN = 1 且 TxE 事件置 1
—ITBUFEN = 1 且 RxNE 事件置 1
位 8 ITERREN：错误中断使能 (Error interrupt enable)
0：禁止错误中断
1：使能错误中断
满足以下条件时将生成此中断：
— BERR = 1
— ARLO = 1
— AF = 1
— OVR = 1
— PECERR = 1
— PWMEOUT = 1
— SMBALERT = 1
位 7:6 保留，必须保持复位值

位 5:0 FREQ[5:0]：外设时钟频率 (Peripheral clock frequency)

外设时钟频率必须使用 APB 时钟频率进行配置(I2C 外设连接到 APB)。允许的最小频率为 2 MHz，最大频率则受限于 APB 最大频率 (42 MHz) 和固有极限频率 46 MHz。

0b000000：不允许
0b000001：不允许
0b000010：2 MHz
...
0b101010：42 MHz
大于 0b101010：不允许

14.5.3I 2 C 自有地址寄存器 1 (I2C_OAR1)

I 2 C Own address register 1
偏移地址：0x08
复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADD MODE	Reserved					ADD[9:8]		ADD[7:1]							ADD0
rw						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15 ADDMODE 寻址模式 (Addressing mode)（从模式）
0：7 位从地址（无法应答 10 位地址）
1：10 位从地址（无法应答 7 位地址）
位 14 应通过软件始终保持为 1。
位 13:10 保留，必须保持复位值
位 9:8 ADD[9:8]：接口地址
7 位寻址模式：无意义
10 位寻址模式：地址的第 9:8 位
位 7:1 ADD[7:1]：接口地址
地址的第 7:1 位
位 0 ADD0：接口地址
7 位寻址模式：无意义
10 位寻址模式：地址的第 0 位

14.5.4I 2 C 自有地址寄存器 2 (I2C_OAR2)

I 2 C Own address register 2
偏移地址：0x0C
复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ADD2[7:1]							ENDUAL
								rw	rw	rw	rw	rw	rw	rw	rw

位 15:8 保留，必须保持复位值
位 7:1 ADD2[7:1]：接口地址
双寻址模式下的地址第 7:1 位
位 0 ENDUAL：双寻址模式使能 (Dual addressing mode enable)
0：7 位寻址模式下仅对 OAR1 地址响应
1：7 位寻址模式下能对 OAR1 和 OAR2 两个地址响应

14.5.5I 2 C 数据寄存器 (I2C_DR)

I 2 C Data register

STC MCU

偏移地址：0x10
复位值：0x0000

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

位 15:8 保留，必须保持复位值
位 7:0 DR[7:0] 8 位数据寄存器
接收的字节或者要发送到总线的字节。
—发送模式：在 DR 寄存器中写入第一个字节时自动开始发送字节。如果在启动传送 (TxE=1) 后立即将下一个要传送的数据置于 DR 中，则可以保持连续的传送流
—接收模式：将接收到的字节复制到 DR 中 (RxNE=1)。如果在接收下一个数据字节 (RxNE=1) 之前读取 DR，则可保持连续的传送流。
注意：在从模式下，地址并不会复制到 DR 中。
注意：硬件不对写冲突进行管理（ TxE=0 时也可对 DR 执行写操作）。
注意：如果发出 ACK 脉冲时出现 ARLO 事件，则不会将接收到的字节复制到 DR 寄存器，因而也无法读取字节。

14.5.6I 2 C 状态寄存器 1 (I2C_SR1)

I 2 C Status register 1
偏移地址：0x14
复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TIME OUT	Res.	PEC ERR	OVR	AF	ARLO	BERR	TxE	RxNE	Res.	STOPF	ADD 10	BTF	ADDR	SB
	rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0		r	r	r	r	r

位 15 保留，必须保持复位值
位 14 PWMEOUT：超时或 Tlow 错误 (PWMeout or Tlow error)
0：无超时错误
1：SCL 低电平时长持续 25 ms（超时）
或
主器件累计时钟低电平延长时间超过 10 ms (Tlow:mext)
或
从器件累计时钟低电平延长时间超过 25 ms (Tlow:sext)
— 在从模式下置 1 时：从器件复位通信且硬件释放数据线
— 在主模式下置 1 时：由硬件发送停止位
— 由软件写入 0 来清零，或在 PE=0 时由硬件清零。
注意：此功能仅在 SMBus 模式下可用。
位 13 保留，必须保持复位值
位 12 PECERR：接收期间 PEC 错误 (PEC Error in reception)
0：无 PEC 错误：接收器在接收 PEC 后返回 ACK（如果 ACK=1）
1：PEC 错误：接收器在接收 PEC 后返回 NACK（无论 ACK 什么值）
位 11 OVR：上溢/下溢 (Overrun/Underrun)
0：未发生上溢/下溢
1：上溢或下溢
—在从模式下由硬件置 1，前提是满足 NOSTRETCH=1 且：
—接收过程中接收到一个新字节（包括 ACK 脉冲）但尚未读取 DR 寄存器。新接收的字节将丢失。
—发送过程中将发送一个新字节但尚未向 DR 寄存器写入数据。同一字节发送两次。
—由软件写入 0 来清零，或在 PE=0 时由硬件清零。

注意：如果 DR 写操作时间与出现 SCL 上升沿的时间非常接近，则发出的数据不确定，并且出现数据保持时间错误。

位 10 AF：应答失败 (Acknowledge failure)

0：未发生应答失败

1：应答失败

—无应答返回时由硬件置 1。

—由软件写入 0 来清零，或在 PE=0 时由硬件清零。

位 9 ARLO：仲裁丢失 (Arbitration lost)（主模式）

0：未检测到仲裁丢失

1：检测到仲裁丢失

当接口在竞争总线是输给另一个主设备时，由硬件将该位置 1

—由软件写入 0 来清零，或在 PE=0 时由硬件清零。

发生 ARLO 事件后，接口会自动切换回从模式 (M/SL=0)。

注意：在 SMBUS 中，从模式下的数据仲裁仅发生在数据阶段或发送确认期间（不适用于地址 确认）。

位 8 BERR：总线错误 (Bus error)

0：无误放的起始或停止位

1：存在误放的起始或停止位

—SCL 为高电平时，若接口在字节传输期间检测到某个无效位置出现 SDA 上升沿或下降沿，则会由硬件将该位置 1。

—由软件写入 0 来清零，或在 PE=0 时由硬件清零。

位 7 TxE：数据寄存器为空 (Data register empty)（发送器）

0：数据寄存器非空

1：数据寄存器为空

—发送过程中 DR 为空时该位置 1。TxE 不会在地址阶段置 1。

—由软件写入 DR 寄存器来清零，或在出现起始、停止位或者 PE=0 时由硬件清零。

如果接收到 NACK 或要发送的下一个字节为 PEC (PEC=1)，TxE 将不会置 1

注意：写入第一个要发送的数据或在 BTF 置 1 时写入数据都无法将 TxE 清零，因为这两种情况 下数据寄存器仍为空。

位 6 RxNE：数据寄存器非空 (Data register not empty)（接收器）

0：数据寄存器为空

1：数据寄存器非空

—接收模式下数据寄存器非空时置 1。RxNE 不会在地址阶段置 1。

—由软件读取或写入 DR 寄存器来清零，或在 PE=0 时由硬件清零。

发生 ARLO 事件时 RxNE 不会置 1。

注意： BTF 置 1 时无法通过读取数据将 RxNE 清零，因为此时数据寄存器仍为满。

位 5 保留，必须保持复位值

位 4 STOPF：停止位检测 (Stop detection)（从模式）

0：未检测到停止位

1：检测到停止位

—从设备在应答脉冲后（如果 ACK=1）检测到停止位，由硬件置 1。

—由软件分别对 SR1 寄存器和 CR1 寄存器执行读操作和写操作来清零，或在 PE=0 时由硬件清零。

注意：收到 NACK 后 STOPF 位不会置 1 。

建议在 STOPF 置 1 后执行完整的清零序列（首先读取 SR1 ，然后写入 CR1 ）。

位 3 ADD10：发送 10 位头（主模式）

0：未发生 ADD10 事件。

1：主器件已发送第一个地址字节（头）。

—主器件在 10 位地址模式下已发送第一个字节时由硬件置 1。

—由软件在读取 SR1 寄存器后在 DR 寄存器中写入第二个地址字节来清零，或在 PE=0 时由硬件

清零。

注 意：收到 NACK 后 ADD10 位不会置 1

位 2 BTF：字节传输完成 (Byte transfer finished)

0：数据字节传输未完成

1：数据字节传输成功完成

—由硬件置 1，前提是满足 NOSTRETCH=0 且：

—接收过程中接收到一个新字节（包括 ACK 脉冲）但尚未读取 DR 寄存器 (RxNE=1)。

—发送过程中将发送一个新字节但尚未向 DR 寄存器写入数据 (TxE=1)。

—由软件读或写 DR 寄存器来清零，或在发送过程中出现起始或停止位后由硬件清零，也可以在 PE=0 时由硬件清零。

注意：收到 NACK 后 BTF 位不会置 1

如果下一个要发送的字节为 PEC （ I2C_SR2 寄存器中的 TRA=1 ， I2C_CR1 寄存器中的 PEC=1 ），则 BTF 位不会置 1

位 0 SB：起始位 (Start bit)（主模式）

0：无起始位

1：起始位已经发送。

—生成启动条件时置 1。

—由软件在读取 SR1 寄存器后写入 DR 寄存器来清零，或在 PE=0 时由硬件清零

14.5.7I 2 C 状态寄存器 2 (I2C_SR2)

I 2 C Status register 2

偏移地址：0x18

复位值：0x0000

注意： 读取 I2C_SR1 后再读取 I2C_SR2 可将 ADDR 标志清零，即使 ADDR 标志在读取 I2C_SR1 之后置 1 也如此。因此，必须仅在 I2C_SR1 中的 ADDR 位已置 1 或者 STOPF 位已清零后读取 I2C_SR2 。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PEC[7:0]								DUALF	Res		GEN CALL		TRA	BUSY	MSL
r	r	r	r	r	r	r	r	r			r		r	r	r

位 15:8 PEC[7:0] 数据包错误校验寄存器

ENPEC=1 时，此寄存器包含内部 PEC。

位 7 DUALF：双标志 (Dual flag)（从模式）

0：接收到的地址与 OAR1 匹配

1：接收到的地址与 OAR2 匹配

—出现停止位、重复起始位或 PE=0 时由硬件清零。

位 6 保留，必须保持复位值

位 5 保留，必须保持复位值

位 4 GENCALL：广播呼叫地址 (General call address)（从模式）

0：无广播呼叫

1：ENGC=1 时接收到广播呼叫地址

—出现停止位、重复起始位或 PE=0 时由硬件清零。

位 3 保留，必须保持复位值

位 2 TRA：发送器/接收器 (Transmitter/receiver)

0：接收器

1：发送器

此位在整个地址阶段的结尾处根据地址字节的 R/W 位状态进行置 1。

同样，检测到停止位 (STOPF=1)、重复起始位、总线仲裁丢失 (ARLO=1) 或当 PE=0 时该位也由硬件清零。

位 1 BUSY：总线忙碌 (Bus busy)

0：总线上无通信

1：总线正在进行通信

—检测到 SDA 或 SCL 低电平时由硬件置 1

—检测到停止位时由硬件清零。

该位指示总线上是否正在进行通信。即使禁止接口 (PE=0) 后此信息也会更新。

位 0 MSL：主/从模式 (Master/slave)

0：从模式

1：主模式

—接口进入主模式时 (SB=1) 由硬件置 1。

—检测到总线上的停止位、仲裁丢失 (ARLO=1) 或当 PE=0 时由硬件清零。

注意： 读取 I2C_SR1 后再读取 I2C_SR2 可将 ADDR 标志清零，即使 ADDR 标志在读取 I2C_SR1 之后置 1 也如此。因此，必须仅在 I2C_SR1 中的 ADDR 位已置 1 或者 STOPF 位已清零后读取 I2C_SR2 。

14.5.8I 2 C 时钟控制寄存器 (I2C_CCR)

I 2 C Clock control register

偏移地址：0x1C

复位值：0x0000

CCR 寄存器必须仅在禁止 I2C (PE = 0) 的情况下配置。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F/S	DUTY	Reserved		CCR[11:0]											
rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15 F/S ：I2C 主模式选择 (I2C master mode selection)

0：标准模式 I2C

1：快速模式 I2C

位 14 DUTY ：快速模式占空比 (Fast mode duty cycle)

0：快速模式 t low /t high = 2

1：快速模式 t low /t high = 16/9（参见 CCR）

位 13:12 保留，必须保持复位值

位 11:0 CCR[11:0] ：快速/标准模式下的时钟控制寄存器 (Clock control register in Fast/Standardmode)（主模式）

控制主模式下的 SCL 时钟。

标准模式或 SMBus 模式：

T high = CCR * T PCLK1

T low = CCR * T PCLK1

快速模式：

如果 DUTY = 0：

T high = CCR * T PCLK1

T low = 2 * CCR * T PCLK1

如果 DUTY = 1：（达到 400 kHz）

T high = 9 * CCR * T PCLK1

T low = 16 * CCR * T PCLK1

例如：要在标准模式下生成 100 kHz 的 SCL 频率：

如果 FREQR = 08，T PCLK1 = 125 ns，则必须将 CCR 编程为 0x28

(0x28 <=> 40d x 125 ns = 5000 ns)。

注意： 1. 允许的最小值为 0x04 ，但快速占空比模式例外，其最小值为 0x01

. 这些时间均未经过滤波。

.CCR 寄存器必须仅在禁止 I 2 C (PE = 0) 的情况下配置。

14.5.9I 2 C TRISE 寄存器 (I2C_TRISE)

I 2 C TRISE register

偏移地址：0x20

复位值：0x0002

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

位 15:6 保留，必须保持复位值

位 5:0 TRISE[5:0]：快速/标准模式下的最大上升时间 (Maximum rise PWMe in Fast/Standard mode)

（主模式）

这些位必须编程为 I 2 C 总线规范中给定的最大 SCL 上升时间加 1。

例如：标准模式下允许的最大 SCL 上升时间为 1000 ns。

如果 I2C_CR2 寄存器中 FREQ[5:0] 位的值等于 0x08 且 T PCLK1 = 125 ns，则 TRISE[5:0] 位

必须编程为 09h。

(1000 ns / 125 ns = 8 + 1)

滤波器值也可以叠加到 TRISE[5:0]。

如果结果不为整数，则 TRISE[5:0] 必须编程为整数部分，以符合 t HIGH 参数要求。

注意： TRISE[5:0] 必须仅在禁止 I2C (PE = 0) 的情况下配置。

14.5.10I 2 C 寄存器映射

下表提供了 I 2 C 寄存器映射和复位值。

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0X00	I2C_CR1	Reserved																SWRST	Reserved		PEC	POS	ACK	STOP	START	NOSTRETCH	ENGC	ENPEC	ENARP	Reserved			PE
	Reset value																	0			0	0	0	0	0	0	0	0	0				0
0X04	I2C_CR2	Reserved																			LAST	DMAEN	ITBUFEN	ITEVTEN	ITERREN	Reserved	FREQ[5:0]						
	Reset value																				0	0	0	0	0		0	0	0	0	0	0	0
0X08	I2C_OAR1	Reserved																ADDMODE	Reserved					ADD[9:8]		ADD[7:1]							ADD0
	Reset value																	0						0	0	0	0	0	0	0	0	0	0
0X10	I2C_OAR2	Reserved																								ADD2[7:1]							ENDUAL
	Reset value																									0	0	0	0	0	0	0	0

0X12	I2C_DR	Reserved													DR[7:0]				
	Reset value														0	0	0	0	0	0	0	0
0X14	I2C_SR1	Reserved							TIMEOUT	Reserved	PECERR	OVR	AF	ARLO	BEER	TxNE	RxNE	Reserved	STOPF	ADD10	BTF	ADDR	SB
	Reset value								0		0	0	0	0	0	0	0		0	0	0	0	0
0X18	I2C_SR2	Reserved							PEC[7:0]							DUALF	Reserved		GENCALL	Reserved	TRA	BUSY	MSL
	Reset value								0	0	0	0	0	0	0	0			0		0	0	
0X1C	I2C_CCR	Reserved							F/S	DUTY	Reserved	CCR[11:0]											
	Reset value								0	0		0	0	0	0	0	0	0	0	0	0	0	0
0X20	I2C_TRISE																	TRISE[5:0]					
	Reset value																	0	0	0	0	0	0

STC MCU

15 高级控制定时器（PWM1 和 PWM8）

15.1PWM1 和 PWM8 简介

高级控制定时器（PWM1 和 PWM8）包含一个 16 位自动重载计数器，该计数器由可编程预分频器驱动。

此类定时器可用于各种用途，包括测量输入信号的脉冲宽度（输入捕获），或者生成输出波 形（输出比较、PWM 和带死区插入的互补 PWM）。

使用定时器预分频器和 RCC 时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制 到几毫秒。

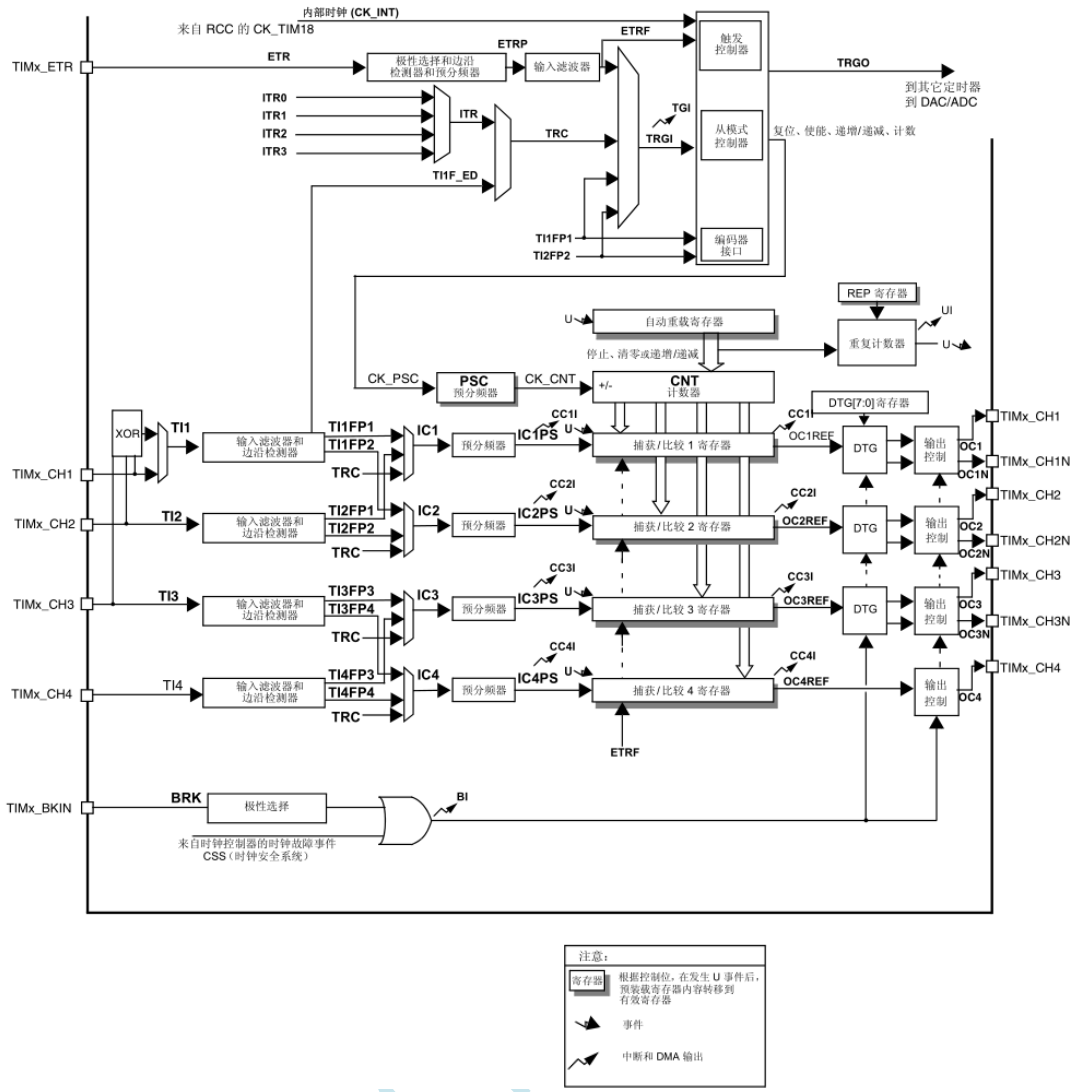
高级控制定时器（PWM1 和 PWM8）和通用 (PWMx) 定时器彼此完全独立，不共享任何资源。

15.2PWM1 和 PWM8 主要特性

PWM1 和 PWM8 定时器具有以下特性：

- 16 位递增、递减、递增/递减自动重载计数器。
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（即运行时修改），分频系数 介于 1 到 65536 之间。
- 多达 4 个独立通道，可用于：
  - — 输入捕获
  - — 输出比较
  - — PWM 生成（边沿和中心对齐模式）
  - — 单脉冲模式输出
- 带可编程死区的互补输出。
- 使用外部信号控制定时器且可实现多个定时器互连的同步电路。
- 重复计数器，用于仅在给定数目的计数器周期后更新定时器寄存器。
- 用于将定时器的输出信号置于复位状态或已知状态的断路输入。
- 发生如下事件时生成中断/DMA 请求：
  - — 更新：计数器上溢/下溢、计数器初始化（通过软件或内部/外部触发）
  - — 触发事件（计数器启动、停止、初始化或通过内部/外部触发计数）
  - — 输入捕获
  - — 输出比较
  - — 断路输入
- 支持定位用增量（正交）编码器和霍尔传感器电路。
- 外部时钟触发输入或逐周期电流管理。

STC MCU



15.3PWM1 和 PWM8 功能说明

15.3.1时基单元

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

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

时基单元包括：

- 计数器寄存器 (PWMx_CNT)
- 预分频器寄存器 (PWMx_PSC)
- 自动重载寄存器 (PWMx_ARR)
- 重复计数器寄存器 (PWMx_RCR)

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存 器。预装载寄存器的内容既可以直接传送到影子寄存器，也可以在每次发生更新事件 (UEV) 时传送到影子寄存器，这取决于 PWMx_CR1 寄存器中的自动重载预装载使能位 (ARPE)。当 计数器达到上

溢值（或者在递减计数时达到下溢值）并且 PWMx_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生 进行详细介绍。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 PWMx_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器（有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明）。

注意，计数器将在 PWMx_CR1 寄存器的 CEN 位置 1 时刻的一个时钟周期后开始计数。

预分频器说明

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

72 和 图 73 以一些示例说明在预分频比实时变化时计数器的行为

图 72. 预分频器分频由 1 变为 2 时的计数器时序图

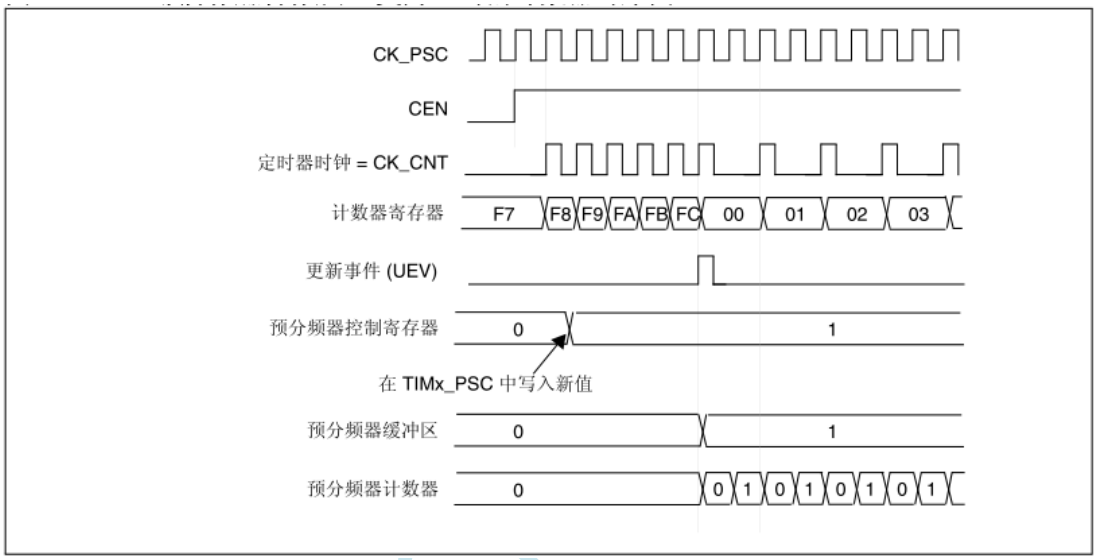
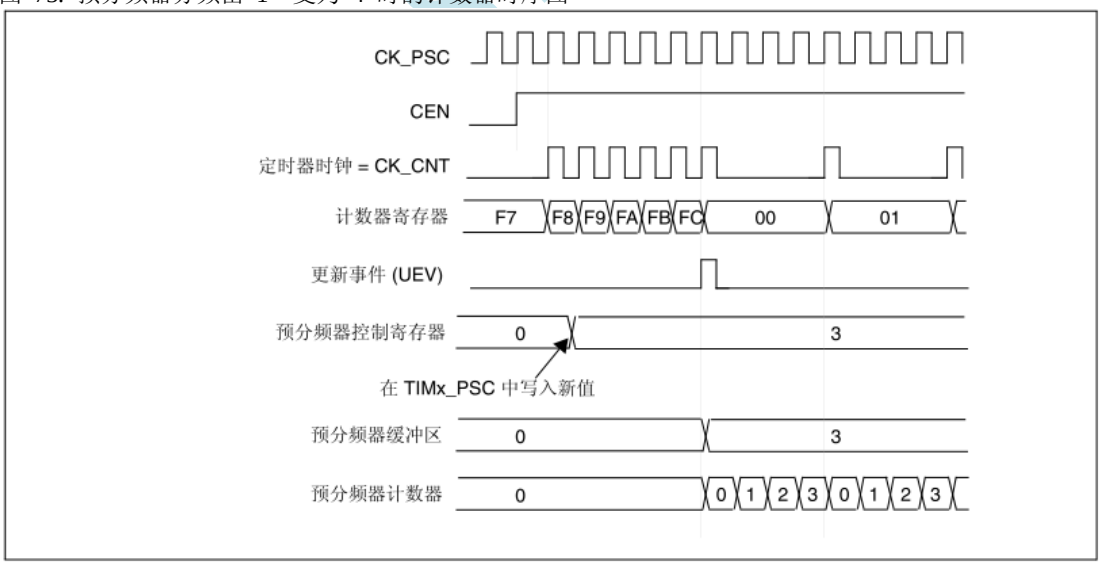


图 73. 预分频器分频由 1 变为 4 时的计数器时序图



15.3.2计数器模式

递增计数模式

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

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

将 PWMx_EGR 寄存器的 UG 位置 1（通过软件或使用从模式控制器）时，也将产生更新事件。通过软件将 PWMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数 器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 PWMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

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

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

图 74. 计数器时序图，1 分频内部时钟

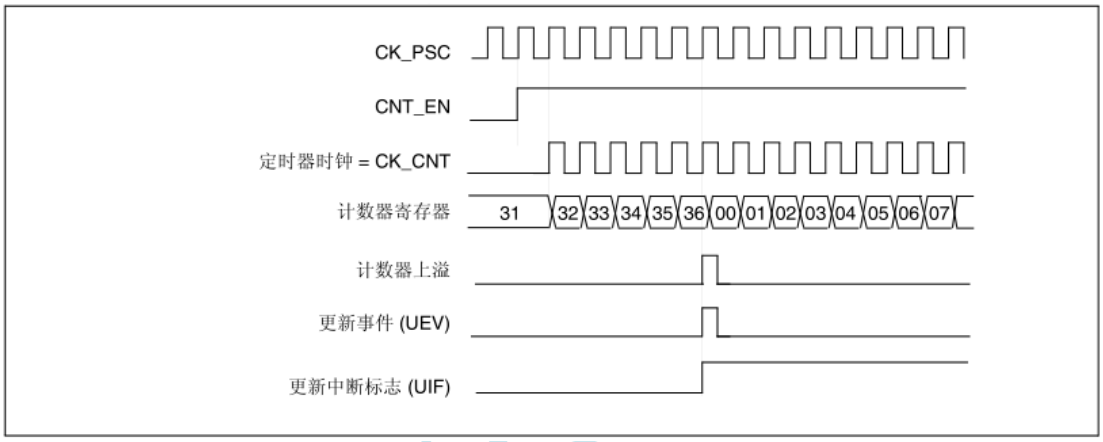


图 75. 计数器时序图，2 分频内部时钟

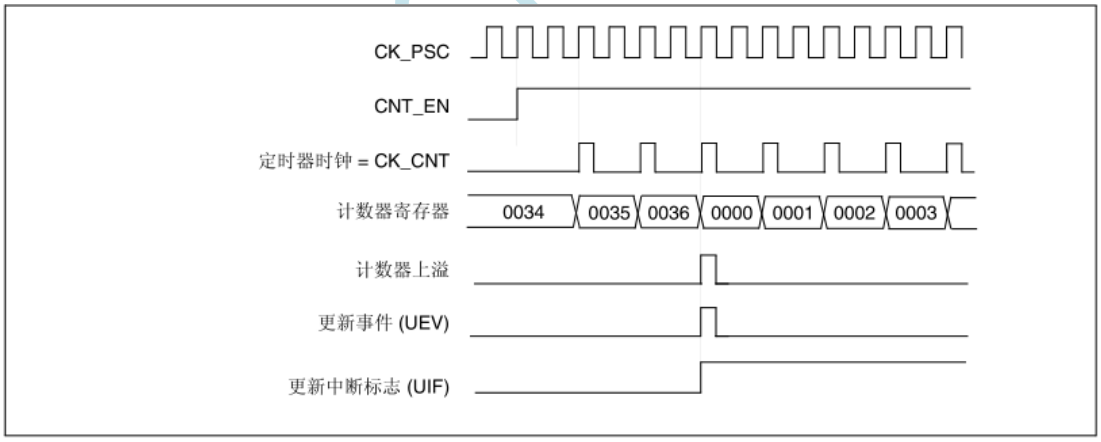


图 76. 计数器时序图，4 分频内部时钟

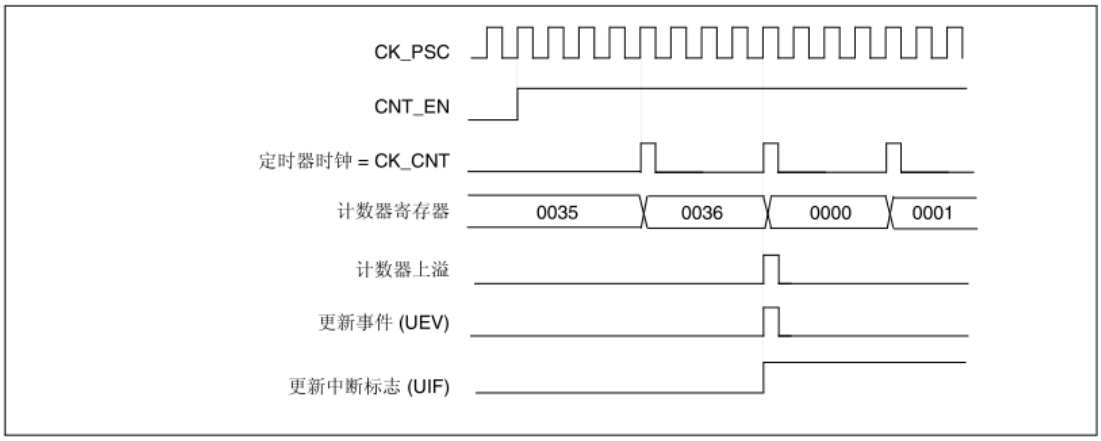


图 77. 计数器时序图，N 分频内部时钟

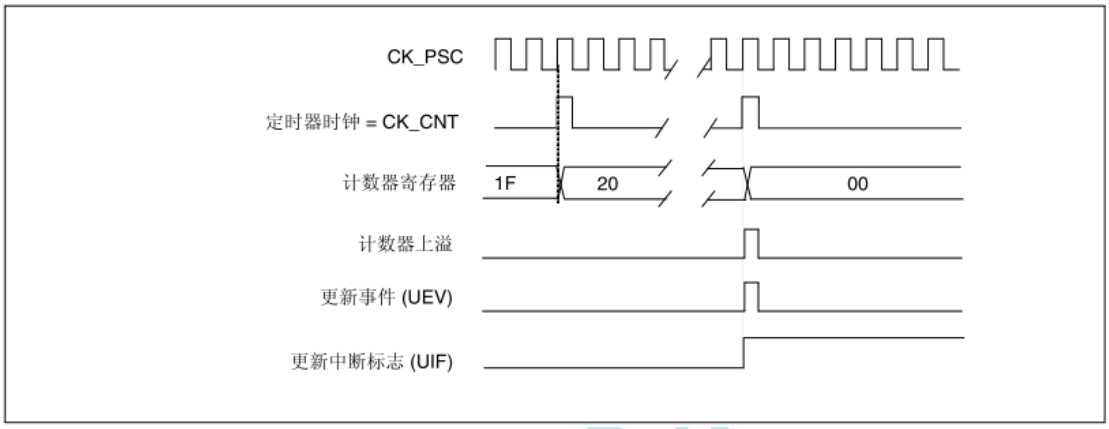


图 78. 计数器时序图，ARPE=0 时更新事件（PWMx_ARR 未预装载）

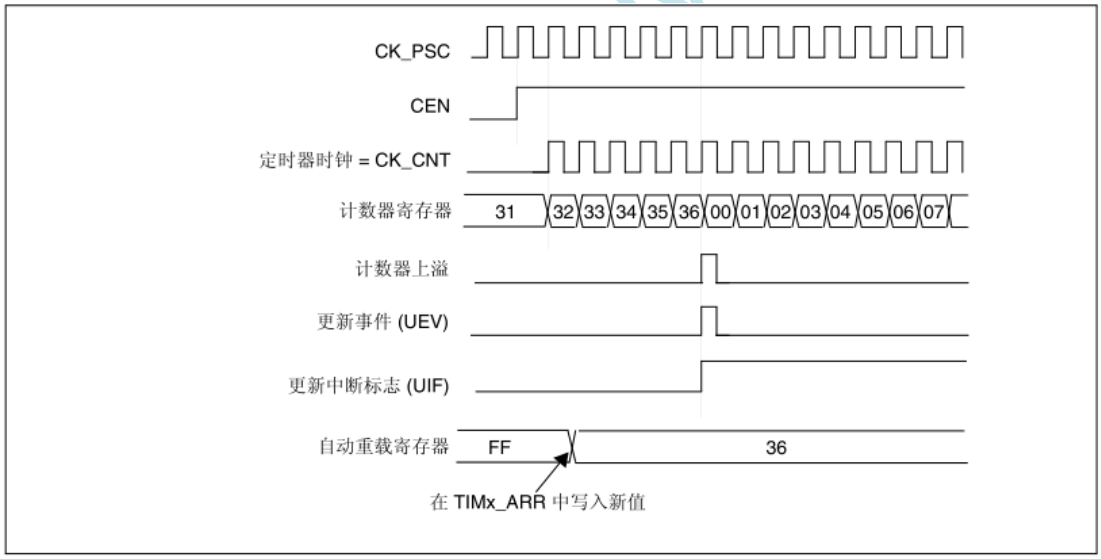
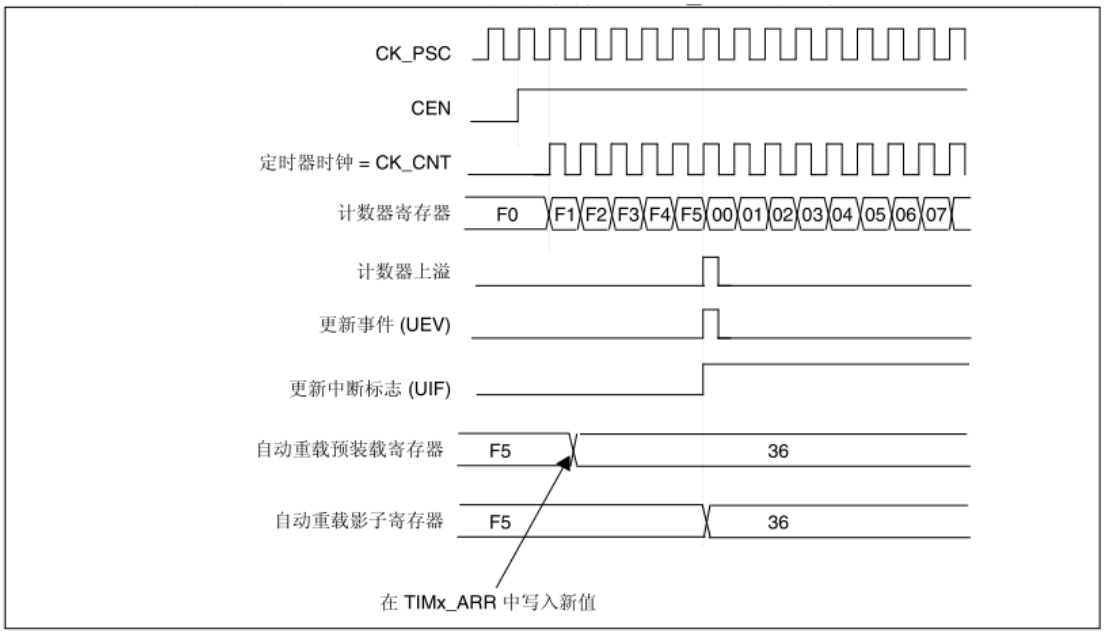


图 79. 计数器时序图，ARPE=1 时更新事件（PWMx_ARR 预装载）



递减计数模式

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

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

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

通过软件将 PWMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器会重新从当前自动重载值开始计数，而预分频器计数器则重新从 0 开始计数（但预分 频比保持不变）。

此外，如果 PWMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

- 重复计数器中将重新装载 PWMx_RCR 寄存器的内容
- 预分频器的缓冲区中将重新装载预装载值（PWMx_PSC 寄存器的内容）
- 自动重载活动寄存器将以预装载值（PWMx_ARR 寄存器的内容）进行更新。注意，自动重载寄存器会在计数器重载之前得到更新，因此，下一个计数周期就是我们所希望的新 的周期长度

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

图 80. 计数器时序图，1 分频内部时钟

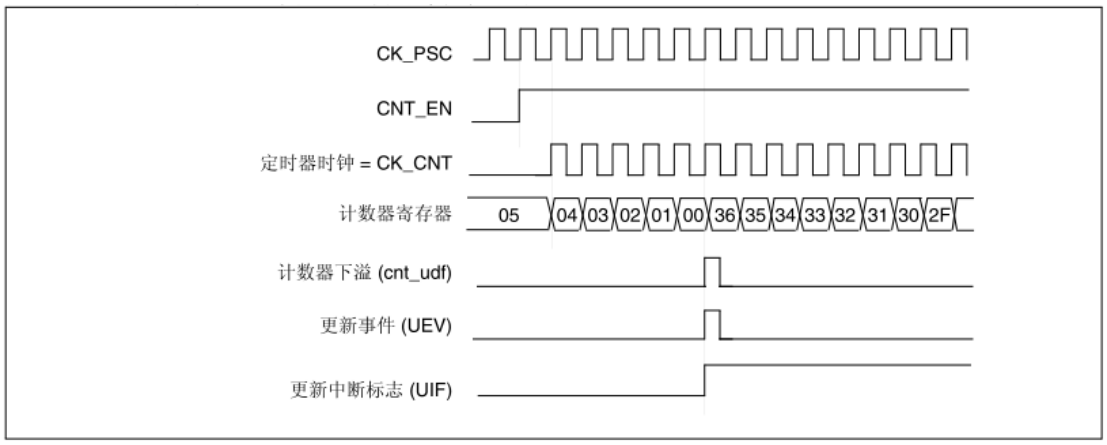


图 81. 计数器时序图，2 分频内部时钟

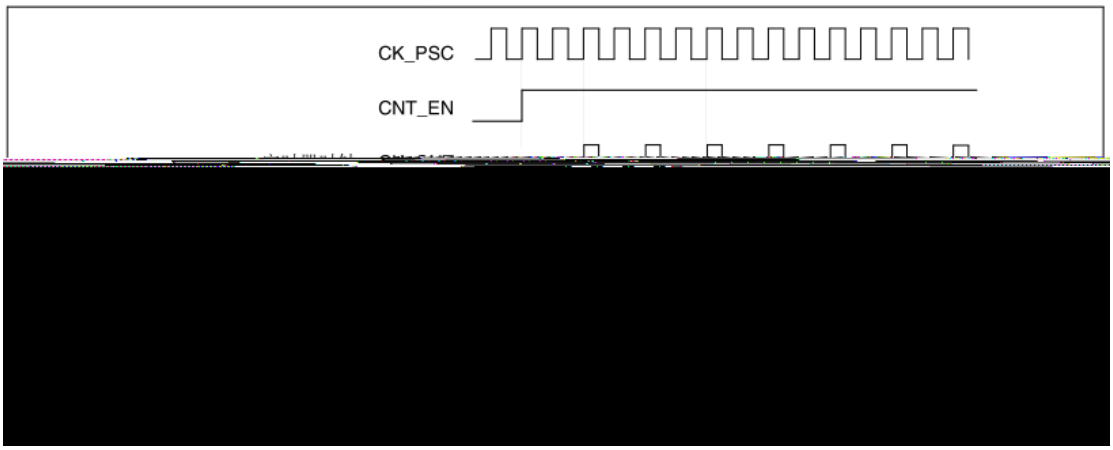


图 82. 计数器时序图，4 分频内部时钟

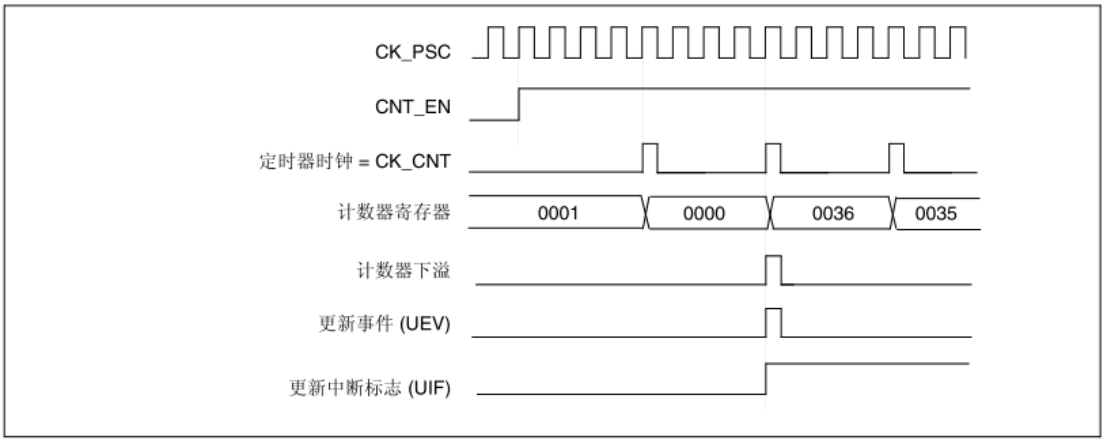


图 83. 计数器时序图，N 分频内部时钟

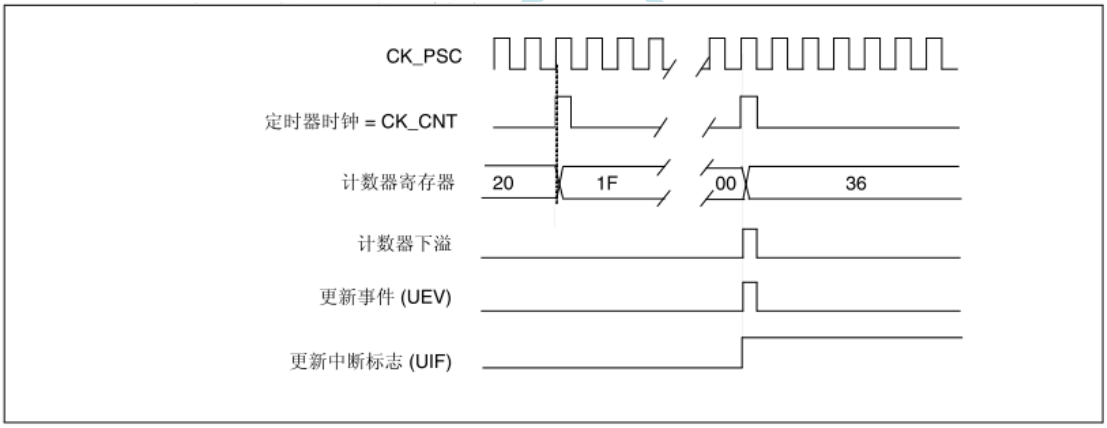
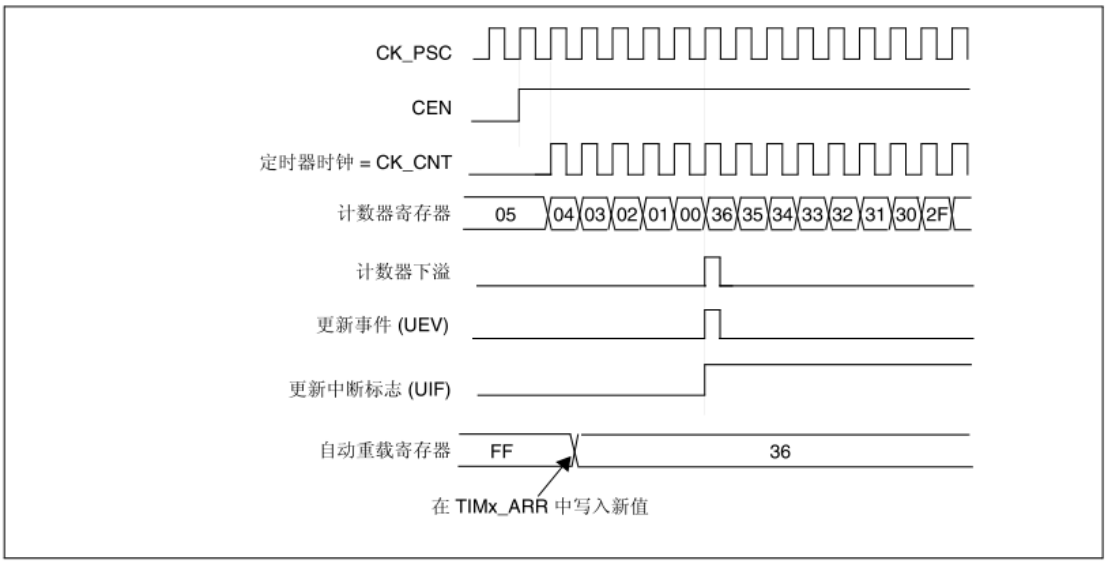


图 84. 计数器时序图，未使用重复计数器时更新事件



中心对齐模式（递增/递减计数）

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

当 PWMx_CR1 寄存器中的 CMS 位不为“00”时，中心对齐模式有效。将通道配置为输出模式时，其输出比较中断标志将在以下模式下置 1，即：计数器递减计数（中心对齐模式 1， CMS =“01”）、计数器递增计数（中心对齐模式 2，CMS =“10”）以及计数器递增/递 减计数（中心对齐模式 3，CMS =“11”）。

在此模式下，PWMx_CR1 寄存器的 DIR 方向位不可写入值，而是由硬件更新并指示当前计数器方向。

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

通过软件将 PWMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器仍会根据当前自动重载值进行递增和递减计数。

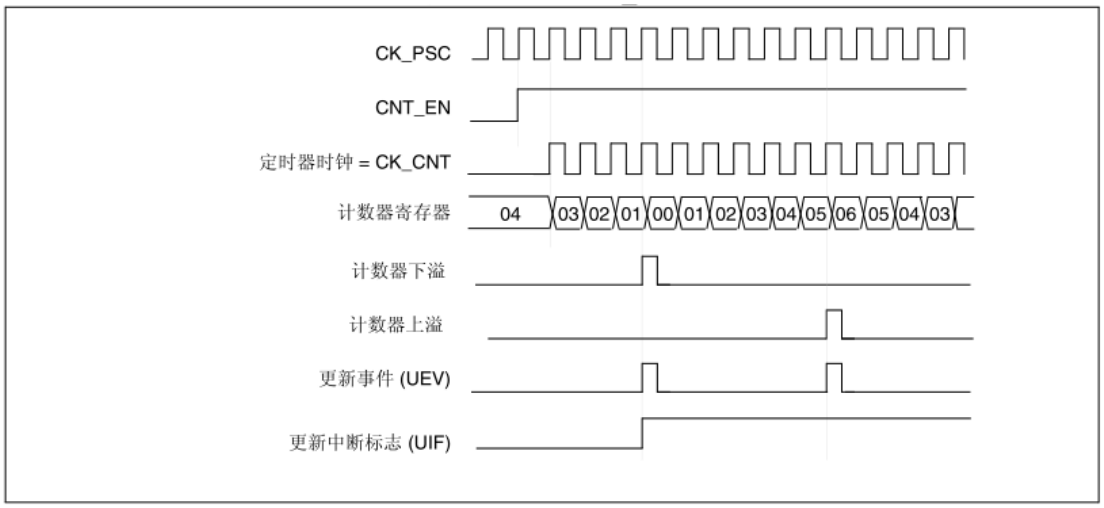
此外，如果 PWMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成 UEV 更新事件，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

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

以下各图以一些示例说明不同时钟频率下计数器的行为。

图 85. 计数器时序图，1 分频内部时钟，PWMx_ARR = 0x6



1. 此处使用中心对齐模式 1（有关详细信息，请参见： PWM1 和 PWM8 寄存器 ）。

图 86. 计数器时序图，2 分频内部时钟

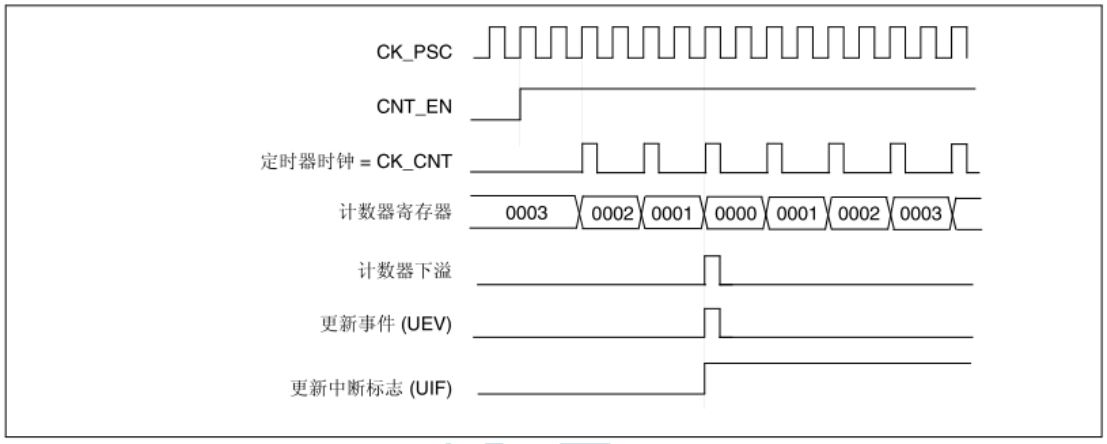
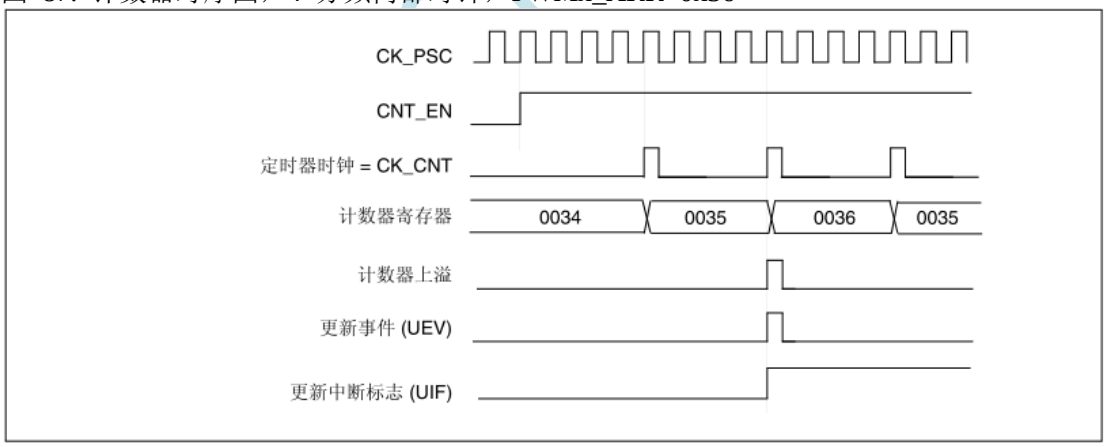


图 87. 计数器时序图，4 分频内部时钟，PWMx_ARR=0x36



1. 中心对齐模式 2 或模式 3 与上溢 UIF 结合使用。

图 88. 计数器时序图，N 分频内部时钟

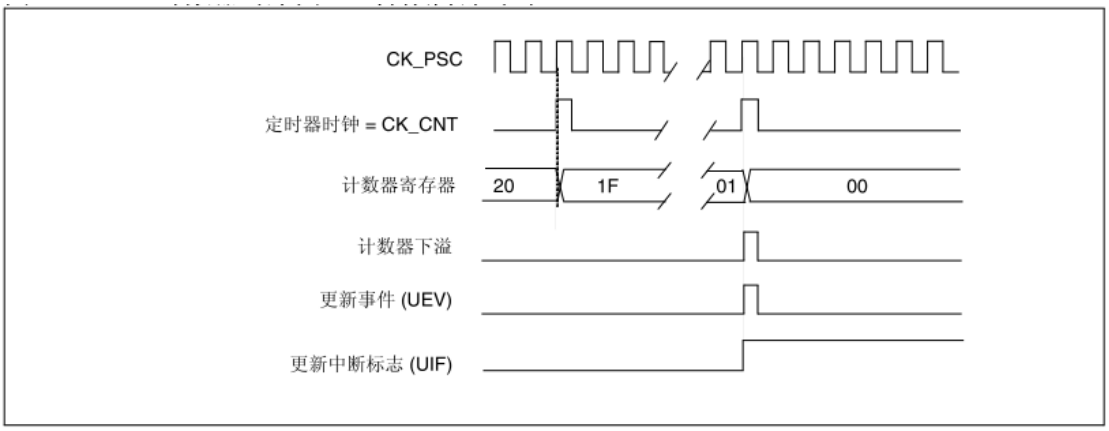


图 89. 计数器时序图，ARPE=1 时的更新事件（计数器下溢）

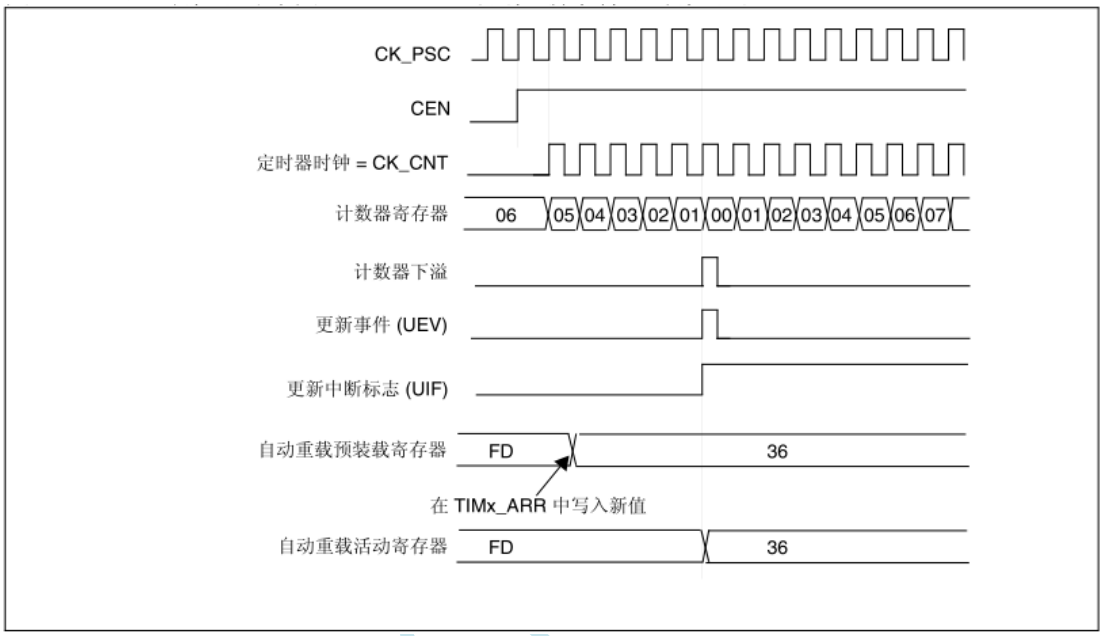
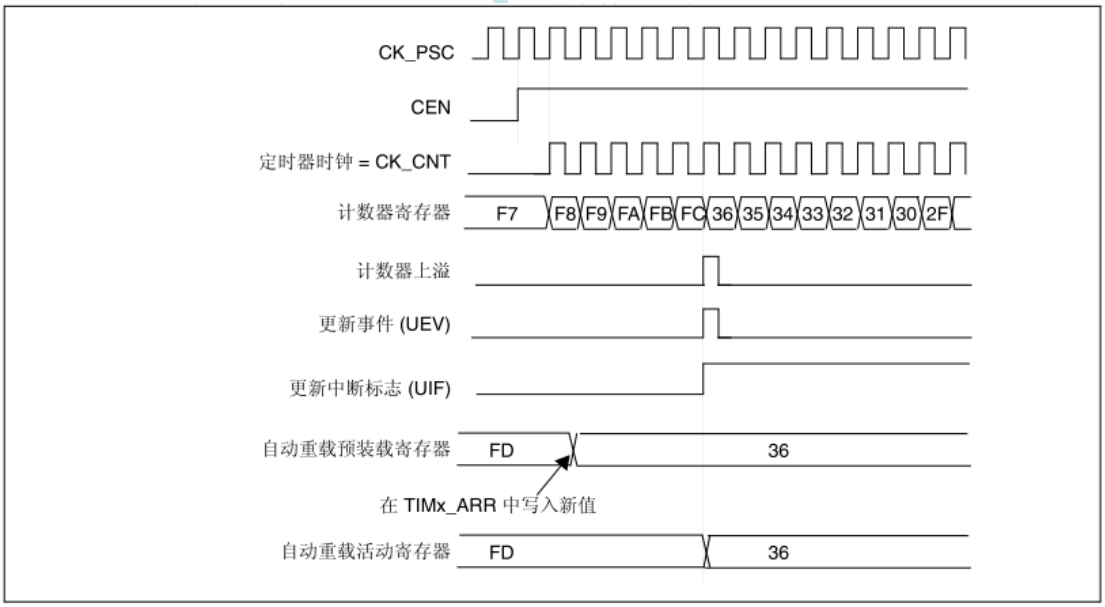


图 90. 计数器时序图，ARPE=1 时的更新事件（计数器上溢）



15.3.3 重复计数器

第 16.3.1 节：时基单元 介绍如何因计数器上溢/下溢而生成更新事件 (UEV)。实际上，只有当重复计数器达到零时，才会生成更新事件。这在生成 PWM 信号时很有用。

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

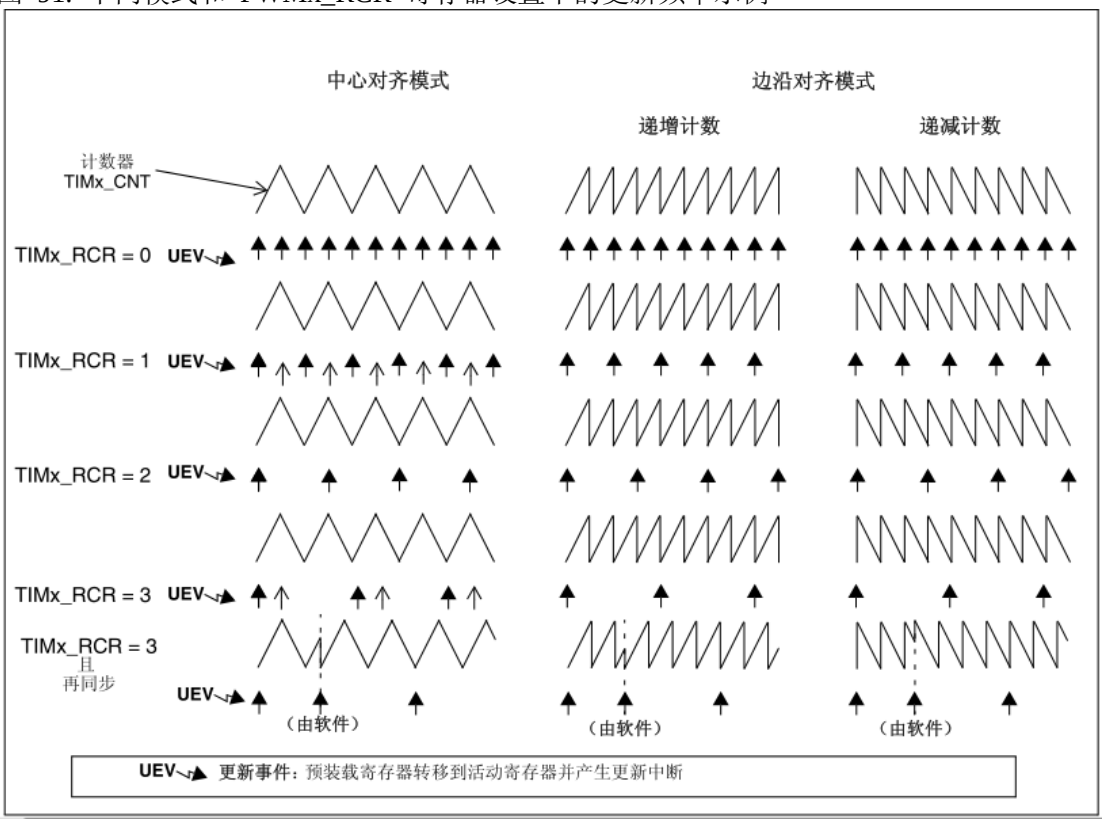
重复计数器在下列情况下递减：

- 递增计数模式下的每个计数器上溢。
- 递减计数模式下的每个计数器下溢。
- 中心对齐模式下每个计数器上溢和计数器下溢。尽管这使得大重复次数不超过 128 个 PWM 周期，但在每个 PWM 周期内可更新占空比两次。当在中心对齐模式下，每个 PWM 周期仅刷新一次比较寄存器时，由于模式的对称性，大分辨率为 2xTck。

重复计数器是自动重载类型；其重复率为 PWMx_RCR 寄存器所定义的值（请参见 图 91）。当更新事件由软件（通过将 PWMx_EGR 寄存器的 UG 位置 1）或硬件（通过从模式控制器）生成时，无论重复计数器的值为多少，更新事件都将立即发生，并且在重复计数器中重新装 载 PWMx_RCR 寄存器的内容。

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

图 91. 不同模式和 PWMx_RCR 寄存器设置下的更新频率示例



15.3.4时钟选择

计数器时钟可由下列时钟源提供：

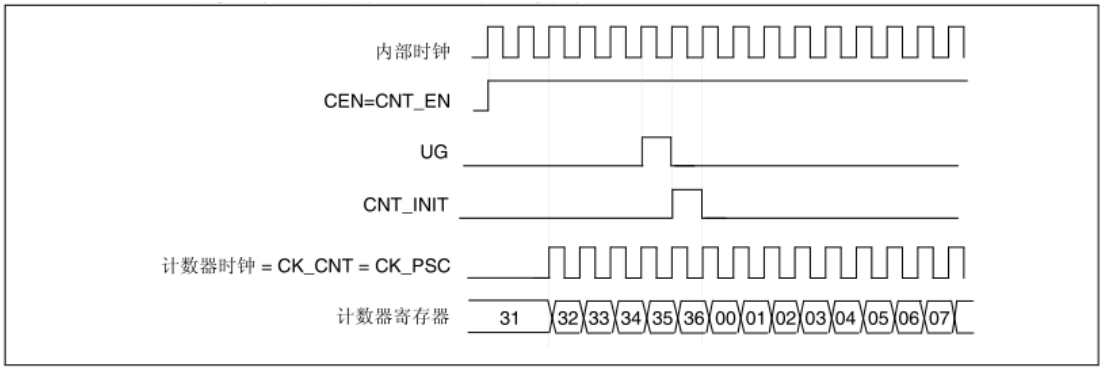
- 内部时钟 (CK_INT)
- 外部时钟模式 1：外部输入引脚
- 外部时钟模式 2：外部触发输入 ETR
- 外部触发输入 (ITRx)：使用一个定时器作为另一定时器的预分频器，例如，可将定时器 1 配置为定时器 2 的预分频器。更多详细信息，请参见 将一个定时器用作另一个定时器 的预分频器 。

内部时钟源 (CK_INT)

如果禁止从模式控制器 (SMS=000)，则 CEN 位、DIR 位（PWMx_CR1 寄存器中）和 UG 位（PWMx_EGR 寄存器中）为实际控制位，并且只能通过软件进行更改（UG 除外，仍保持自动清零）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 CK_INT 提供。

图 92 显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）。

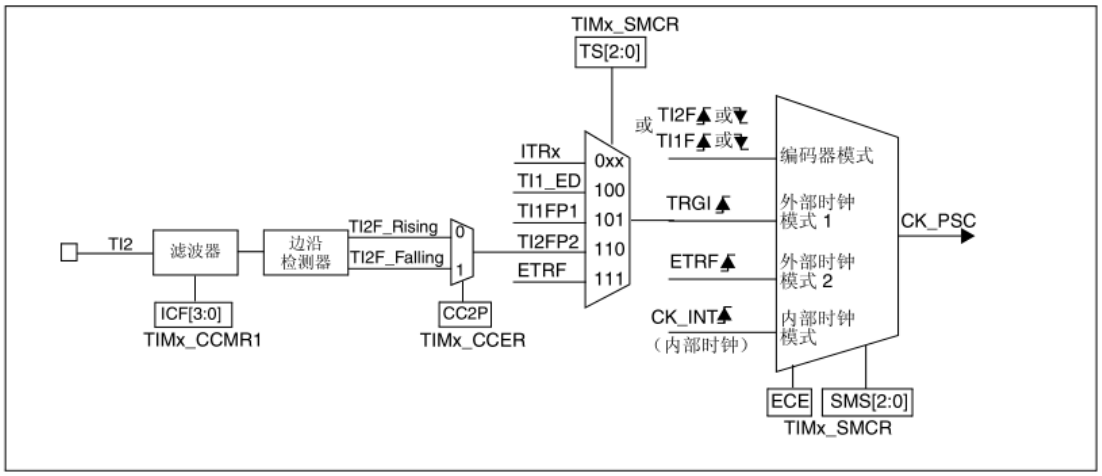
图 92. 正常模式下的控制电路，1 分频内部时钟



外部时钟源模式

1 当 PWMx_SMCR 寄存器中的 SMS=111 时，可选择此模式。计数器可在选定的输入信号上出现上升沿或下降沿时计数。

图 93. TI2 外部时钟连接示例



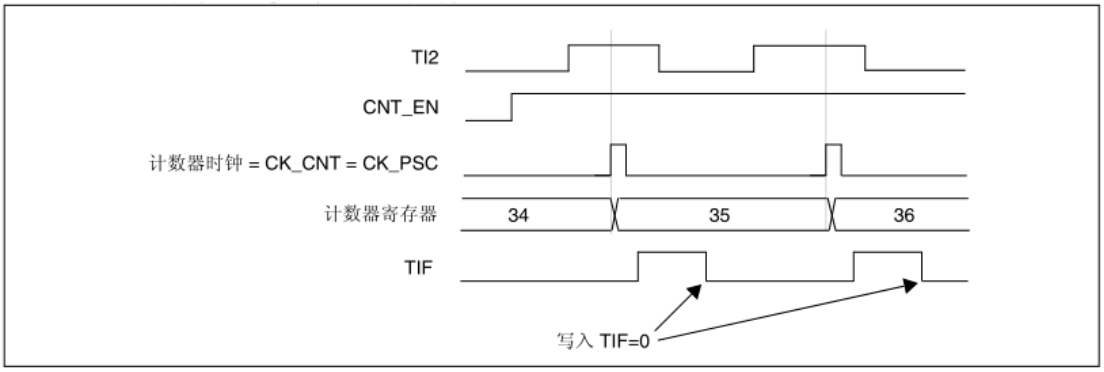
例如，要使递增计数器在 TI2 输入出现上升沿时计数，请执行以下步骤：

1. 通过在 PWMx_CCMR1 寄存器中写入 CC2S =“01”来配置通道 2，使其能够检测 TI2 输入的上升沿。
2. 通过在 PWMx_CCMR1 寄存器中写入 IC2F[3:0] 位来配置输入滤波时间（如果不需要任何滤波，请保持 IC2F=0000）。
3. 通过在 PWMx_CCER 寄存器中写入 CC2P=0 和 CC2NP=0 来选择上升沿极性。
4. 通过在 PWMx_SMCR 寄存器中写入 SMS=111，使定时器在外部时钟模式 1 下工作。
5. 通过在 PWMx_SMCR 寄存器中写入 TS=110 来选择 TI2 作为触发输入源。
6. 通过在 PWMx_CR1 寄存器中写入 CEN=1 来使能计数器。

注意：由于捕获预分频器不用于触发操作，因此无需对其进行配置。

当 TI2 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。 TI2 的上升沿与实际计数器时钟之间的延迟是由于 TI2 输入的重新同步电路引起的

图 94. 外部时钟模式 1 下的控制电路

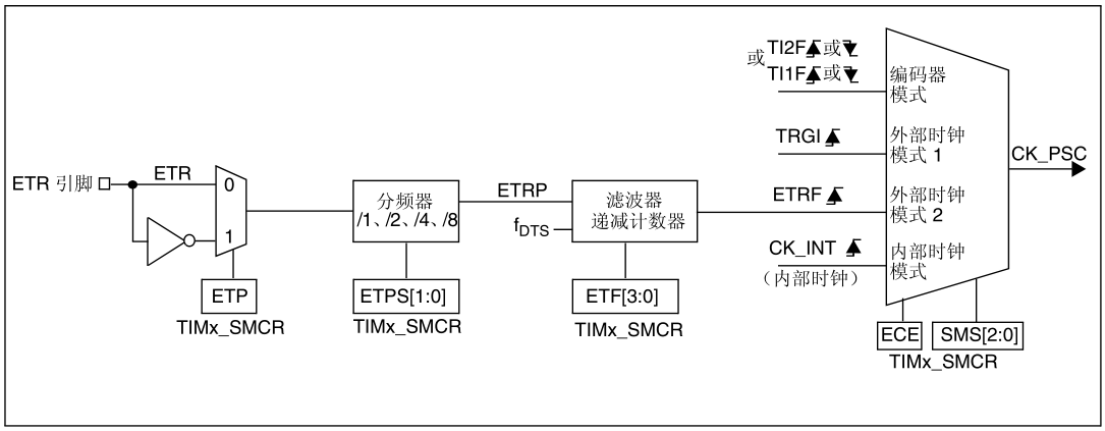


外部时钟源模式 2

通过在 PWMx_SMCR 寄存器中写入 ECE=1 可选择此模式。计数器可在外部触发输入 ETR 出现上升沿或下降沿时计数。

图 95 简要介绍了外部触发输入模块。

图 95. 外部触发输入模块



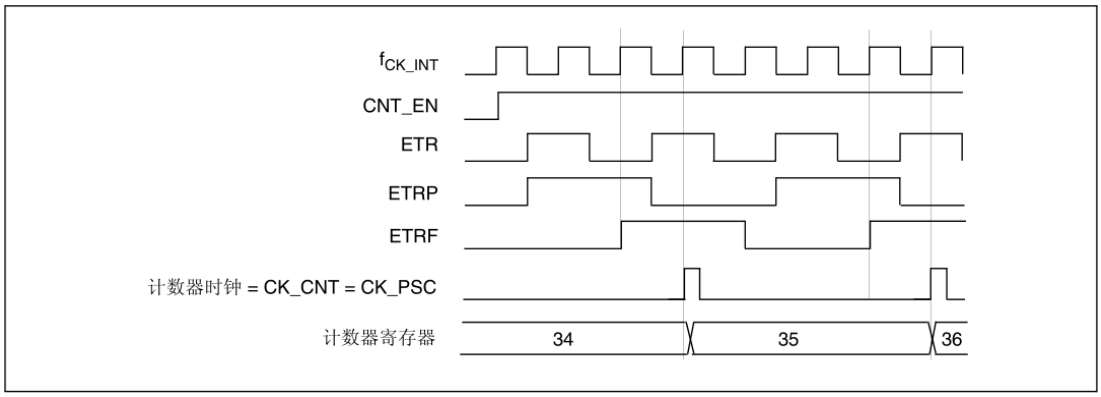
例如，要使递增计数器在 ETR 每出现 2 个上升沿时计数，请执行以下步骤：

1. 由于此例中不需滤波器，因此在 PWMx_SMCR 寄存器中写入 ETF[3:0]=0000。
2. 通过在 PWMx_SMCR 寄存器中写入 ETPS[1:0]=01 来设置预分频器。
3. 通过在 PWMx_SMCR 寄存器中写入 ETP=0 来选择 ETR 引脚的上升沿检测。
4. 通过在 PWMx_SMCR 寄存器中写入 ECE=1 来使能外部时钟模式 2。
5. 通过在 PWMx_CR1 寄存器中写入 CEN=1 来使能计数器。

ETR 每出现 2 个上升沿，计数器计数一次。

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

图 96. 外部时钟模式 2 下的控制电路



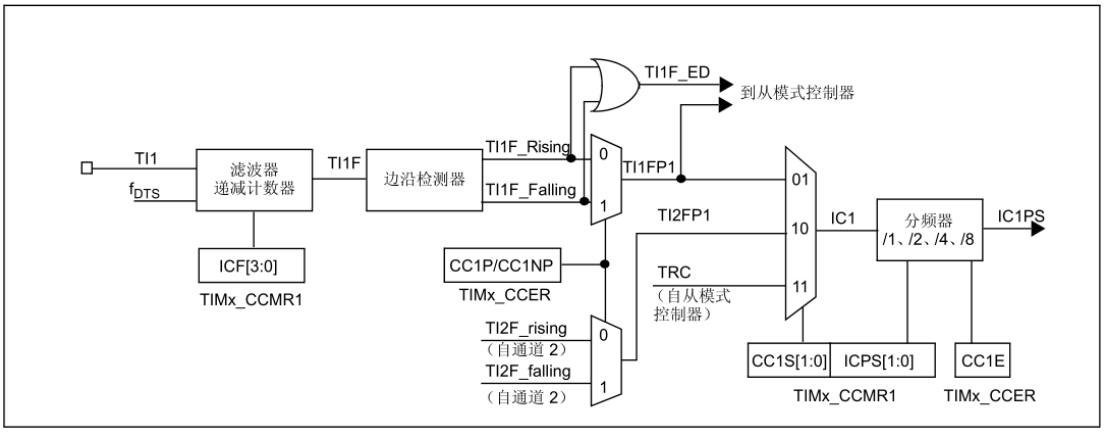
15.3.5捕获/比较通道

每个捕获/比较通道均围绕一个捕获/比较寄存器（包括一个影子寄存器）、一个捕获输入阶 段（数字滤波、多路复用和预分频器）和一个输出阶段（比较器和输出控制）构建而成。

图 97 到 图 100 概括介绍了一个捕获/比较通道。

输入阶段对相应的 TIx 输入进行采样，生成一个滤波后的信号 TIxF。然后，带有极性选择功 能的边沿检测器生成一个信号 (TIxFPx)，该信号可用作从模式控制器的触发输入，也可用作 捕获命令。该信号先进行预分频 (ICxPS)，而后再进入捕获寄存器。

图 97. 捕获/ 比较通道（例如：通道 1 输入阶段）



输出阶段生成一个中间波形作为基准：OCxRef（高电平有效）。链的末端决定终输出信 号的极性。

图 98. 捕获/ 比较通道 1 主电路

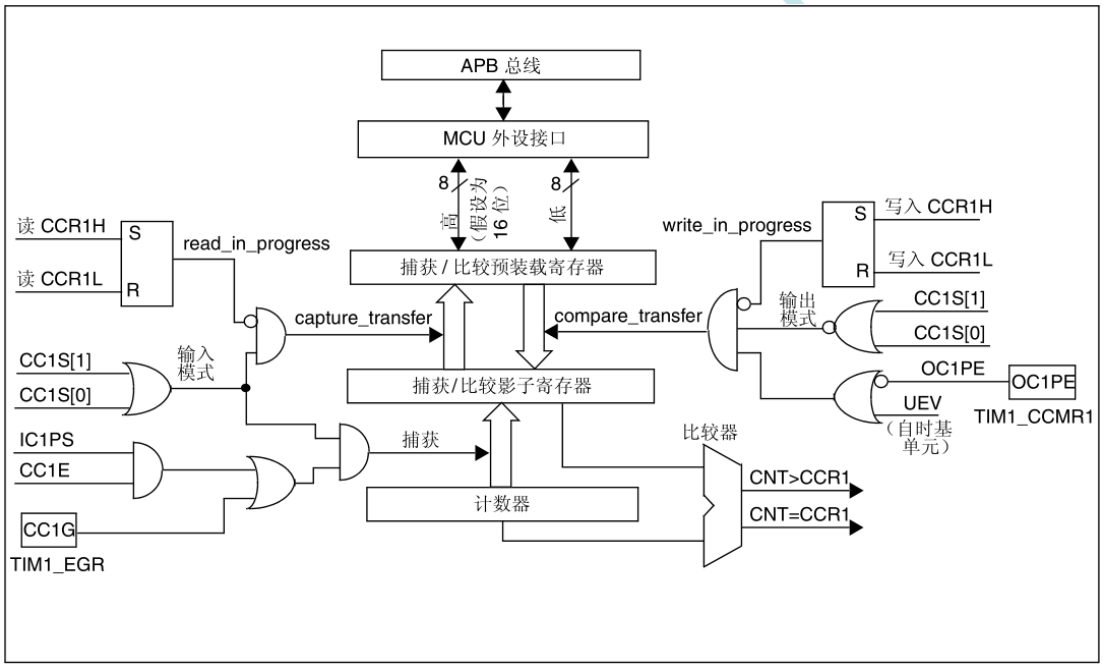


图 99. 捕获/ 比较通道的输出阶段（通道 1 到 3)

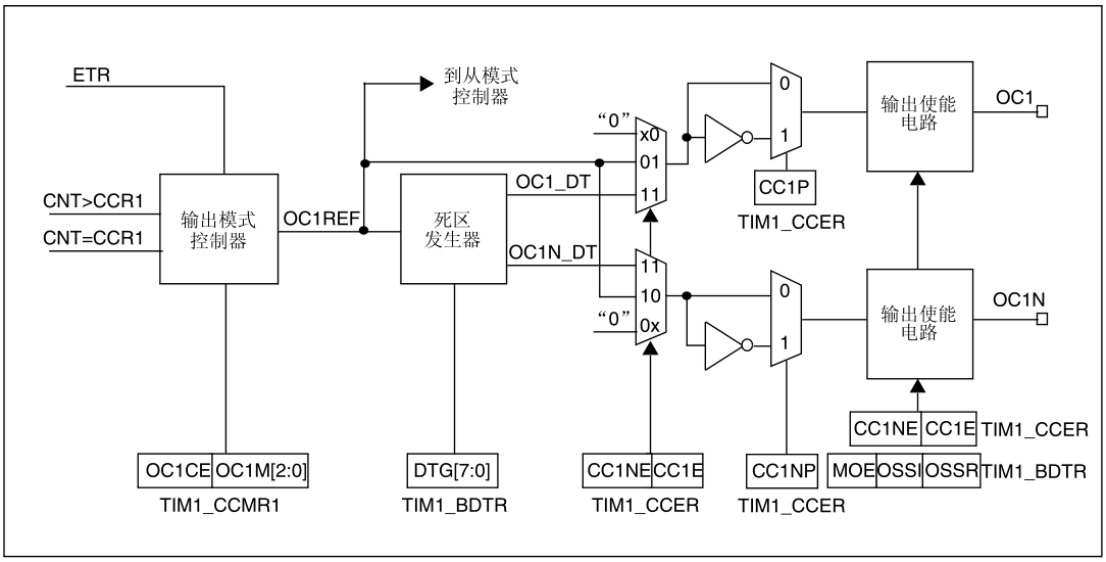
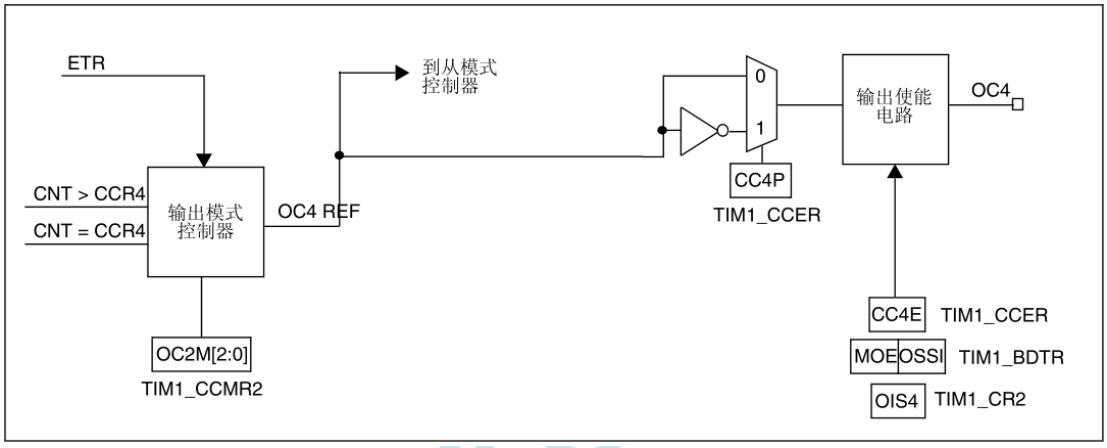


图 100. 捕获/ 比较通道的输出阶段（通道 4 ）



捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。始终可通过读写操作访问预装载 寄存器。
在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容复制到预装载寄存 器中。
在比较模式下，预装载寄存器的内容将被复制到影子寄存器中，然后将影子寄存器的内容与 计数器进行比较。

15.3.6输入捕获模式

在输入捕获模式下，当相应的 ICx 信号检测到跳变沿后，将使用捕获/比较寄存器 (PWMx_CCRx) 来锁存计数器的值。发生捕获事件时，会将相应的 CCXIF 标志（PWMx_SR 寄存器）置 1，并可发送中断或 DMA 请求（如果已使能）。如果发生捕获事件时 CCxIF 标志已处于高位， 则会将重复捕获标志 CCxOF（PWMx_SR 寄存器）置 1。可通过软件向 CCxIF 写入 0 来给=CCxIF 清零，或读取存储在 PWMx_CCRx 寄存器中的已捕获数据。向 CCxOF 写入“0”后=会将其清零。
以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 PWMx_CCR1 中。具体操作步骤如下：
● 选择有效输入：PWMx_CCR1 必须连接到 TI1 输入，因此向 PWMx_CCMR1 寄存器中的 CC1S 位写入 01。只要 CC1S 不等于 00，就会将通道配置为输入模式，并且 PWMx_CCR1 寄存器将处于只读状态。
● 根据连接到定时器的信号，对所需的输入滤波时间进行编程（如果输入为 TIx 输入，则对 PWMx_CCMRx 寄存器中的 ICxF 位进行编程）。假设信号边沿变化时，输入信号多在 5 个内部时钟周期内发生抖动。因此，我们必须将滤波时间设置为大于 5 个内部时钟周期。在检测到 8 个具有新电平的连续采样（以 fDTS 频率采样）后，可以确认 TI1 上的跳变沿。然后向 PWMx_CCMR1 寄存器中的 IC1F 位写入 0011。
● 通过在 PWMx_CCER 寄存器中将 CC1P 位和 CC1NP 位写入 0，选择 TI1 上的有效转换 边沿（本例中为上升沿）。
● 对输入预分频器进行编程。在本例中，我们希望每次有效转换时都执行捕获操作，因此需要禁止预分频器（向 PWMx_CCMR1 寄存器中的 IC1PS 位写入“00”）。
● 通过将 PWMx_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。

● 如果需要，可通过将 PWMx_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求，并且/或者通过将该寄存器中的 CC1DE 位置 1 来使能 DMA 请求。

发生输入捕获时：

● 发生有效跳变沿时，PWMx_CCR1 寄存器会获取计数器的值。

● 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。

● 根据 CC1IE 位生成中断。

● 根据 CC1DE 位生成 DMA 请求。

要处理重复捕获，建议在读出捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出 标志之后与读取数据之前可能出现的重复捕获信息。

注 意 ： 通 过 软 件 将 PWMx_EGR 寄存器中的相应 CCxG 位置 1 可生成 IC 中断和 / 或 DMA 请求。

15.3.7PWM 输入模式

此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

● 两个 ICx 信号被映射至同一个 TIx 输入。

● 这两个 ICx 信号在边沿处有效，但极性相反。

● 选择两个 TIxFP 信号之一作为触发输入，并将从模式控制器配置为复位模式。

例如，可通过以下步骤对应用于 TI1 的 PWM 的周期（位于 PWMx_CCR1 寄存器中）和占空比（位于 PWMx_CCR2 寄存器中）进行测量（取决于 CK_INT 频率和预分频器的值）：

● 选择 PWMx_CCR1 的有效输入：向 PWMx_CCMR1 寄存器中的 CC1S 位写入 01（选择 TI1）。

● 选择 TI1FP1 的有效极性（用于在 PWMx_CCR1 中捕获和计数器清零）：向 CC1P 位和 CC1NP 位写入“0”（上升沿有效）。

● 选择 PWMx_CCR2 的有效输入：向 PWMx_CCMR1 寄存器中的 CC2S 写入 10（选择 TI1）。

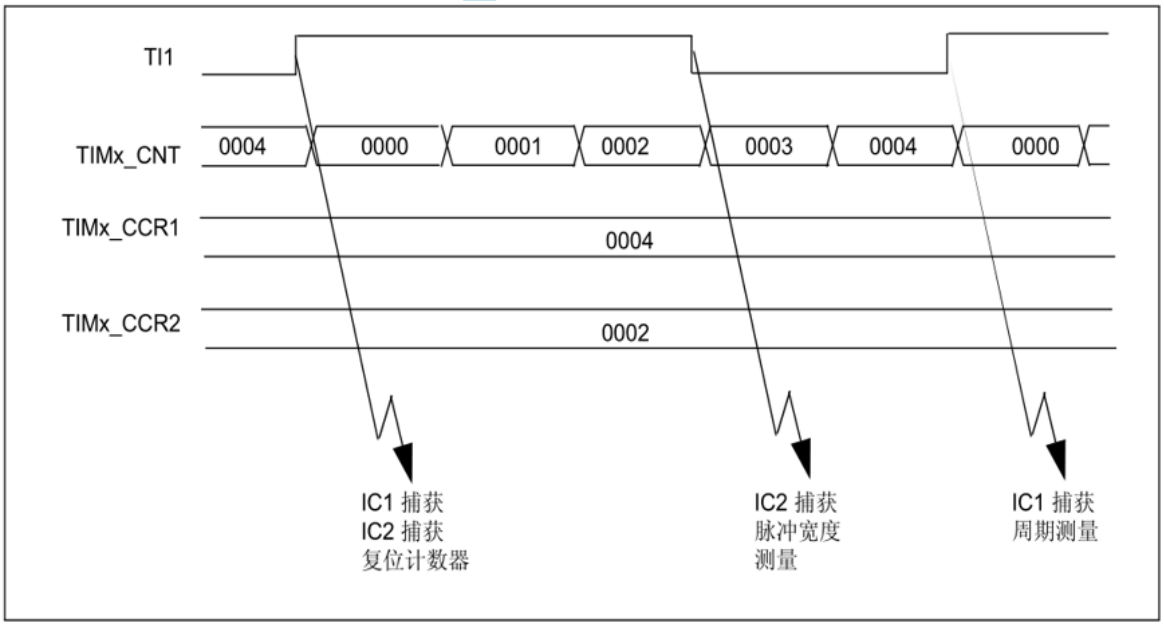
● 选择 TI1FP2 的有效极性（用于在 PWMx_CCR2 中捕获）：向 CC2P 位和 CC2NP 位写入“1”（下降沿有效）。

● 选择有效触发输入：向 PWMx_SMCR 寄存器中的 TS 位写入 101（选择 TI1FP1）。

● 将从模式控制器配置为复位模式：向 PWMx_SMCR 寄存器中的 SMS 位写入 100。

● 使能捕获：向 PWMx_CCER 寄存器中的 CC1E 位和 CC2E 位写入“1”。

图 101. PWM 输入模式时序



15.3.8强制输出模式

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

要将输出比较信号 (OCXREF/OCx) 强制设置为有效电平，只需向相应 PWMx_CCMRx 寄存器中的 OCxM 位写入 101。OCXREF 进而强制设置为高电平（OCxREF 始终为高电平有效），同时 OCx 获取 CCxP 极性位的相反值。

例如：CCxP=0（OCx 高电平有效）=> OCx 强制设置为高电平。

通过向 PWMx_CCMRx 寄存器中的 OCxM 位写入 100，可将 OCxREF 信号强制设置为低电平。

无论如何，PWMx_CCRx 影子寄存器与计数器之间的比较仍会执行，而且允许将标志置 1。因此可发送相应的中断和 DMA 请求。下面的输出比较模式一节对此进行了介绍。

15.3.9输出比较模式

此功能用于控制输出波形，或指示已经过某一时间段。

当捕获/比较寄存器与计数器之间相匹配时，输出比较功能：

- 将为相应的输出引脚分配一个可编程值，该值由输出比较模式（PWMx_CCMRx 寄存器中的 OCxM 位）和输出极性（PWMx_CCER 寄存器中的 CCxP 位）定义。匹配时，输出引脚既可保持其电平 (OCXM=000)，也可设置为有效电平 (OCXM=001)、无效电平 (OCXM=010) 或进行翻转 (OCxM=011)。
- 将中断状态寄存器中的标志置 1（PWMx_SR 寄存器中的 CCxIF 位）。
- 如果相应中断使能位（PWMx_DIER 寄存器中的 CCXIE 位）置 1，将生成中断。
- 如果相应 DMA 使能位（PWMx_DIER 寄存器的 CCxDE 位，PWMx_CR2 寄存器的 CCDS 位，用来选择 DMA 请求）置 1，将发送 DMA 请求。

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

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

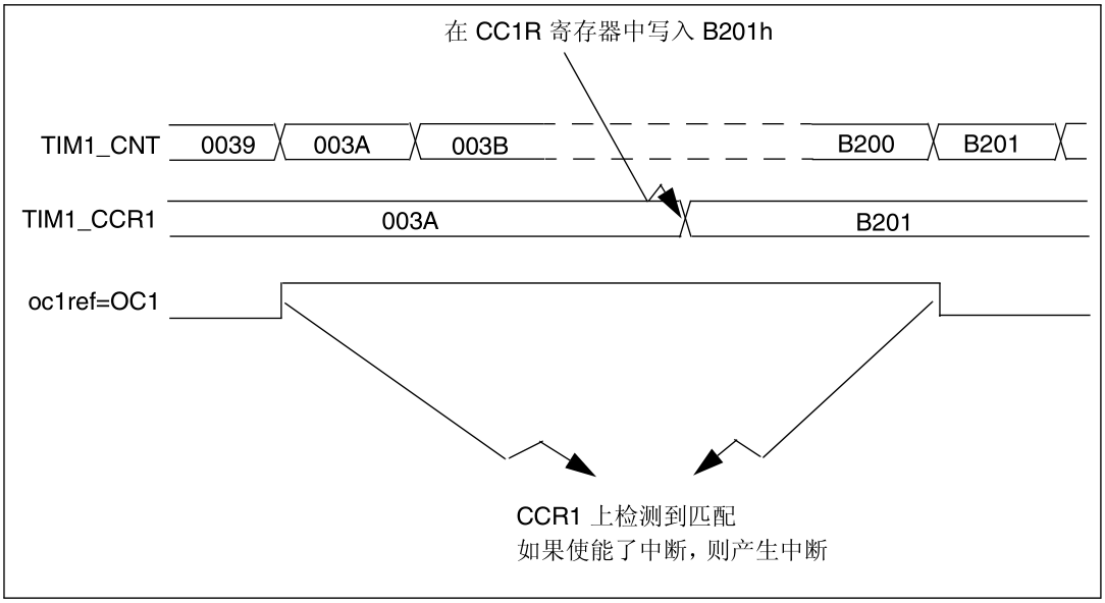
步骤：

1. 选择计数器时钟（内部、外部、预分频器）。
2. 在 PWMx_ARR 和 PWMx_CCRx 寄存器中写入所需数据。
3. 如果要生成中断请求，则需将 CCxIE 位置 1。
4. 选择输出模式。例如：
   — 当 CNT 与 CCRx 匹配时，写入 OCxM = 011 以翻转 OCx 输出引脚
   — 写入 OCxPE = 0 以禁止预装载寄存器 — 写入 CCxP = 0 以选择高电平有效极性
   — 写入 CCxE = 1 以使能输出
5. 通过将 PWMx_CR1 寄存器中的 CEN 位置 1 来使能计数器。

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

图 102 给出了一个示例。

图 102. 输出比较模式，翻转 OC1



15.3.10 PWM 模式

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

通过向 PWMx_CCMRx 寄存器中的 OCxM 位写入 110 （PWM 模式 1）或 111 （PWM 模式 2），可以独立选择各通道 （每个 OCx 输出对应一个 PWM）的 PWM 模式。必须通过将 PWMx_CCMRx 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器，后通过将 PWMx_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器（在递增计数或中心对齐模式下）。

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

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

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

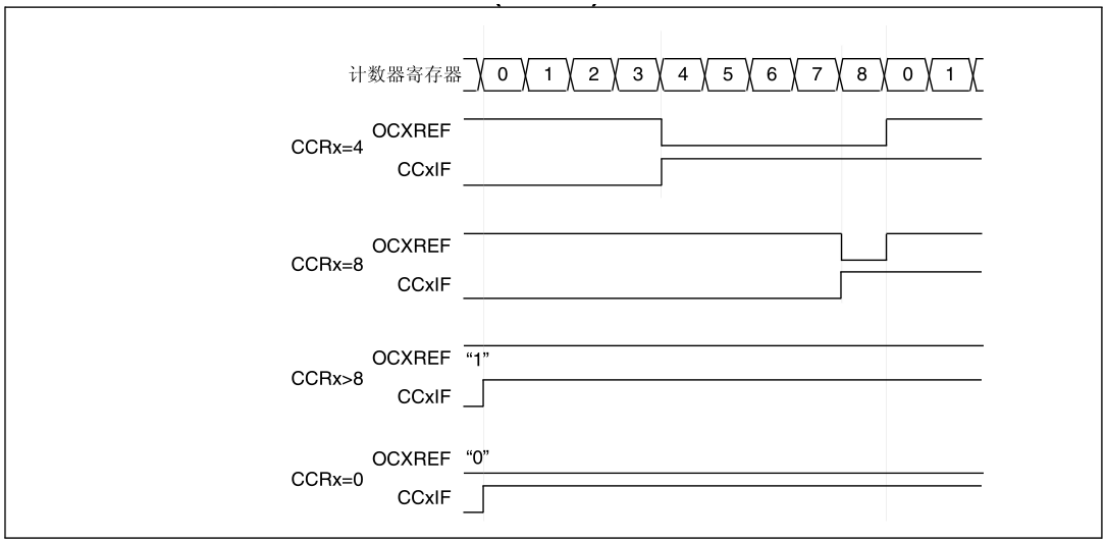
根据 PWMx_CR1 寄存器中的 CMS 位 状态，定时器能够产生边沿对齐模式或中心对齐模式的 PWM 信号。

PWM 边沿对齐模式

- 递增计数配置 当 PWMx_CR1 寄存器中的 DIR 位为低时执行递增计数。请参见递增计数模式 一节。

以下以 PWM 模式 1 为例。只要 PWMx_CNT < PWMx_CCRx，PWM 参考信号 OCxREF 便为高电平，否则为低电平。如果 PWMx_CCRx 中的比较值大于自动重载值（PWMx_ARR 中），则 OCxREF 保持为“1”。如果比较值为 0，则 OCxRef 保持为“0”。 图 103 举例介绍边沿对齐模式的一些 PWM 波形 (PWMx_ARR=8)。

图 103. 边沿对齐模式的 PWM 波形 (ARR=8)



● 递减计数配置

当 PWMx_CR1 寄存器中的 DIR 位为高时执行递减计数。请参见递减计数模式一节。

在 PWM 模式 1 下，只要 PWMx_CNT > PWMx_CCRx，参考信号 OCxRef 即为低电平，否则其为高电平。如果 PWMx_CCRx 中的比较值大于 PWMx_ARR 中的自动重载值，则 OCxREF 保持为"1"。此模式下不可能产生 0% 的 PWM 波形。

PWM 中心对齐模式

当 PWMx_CR1 寄存器中的 CMS 位不为"00"（其余所有配置对 OCxRef/OCx 信号具有相同的作用），中心对齐模式生效。根据 CMS 位的配置，可以在计数器递增计数、递减计数或同时递增和递减计数时将比较标志置 1。PWMx_CR1 寄存器中的方向位 (DIR) 由硬件更新，不得通过软件更改。请参见中心对齐模式（递增 / 递减计数）。

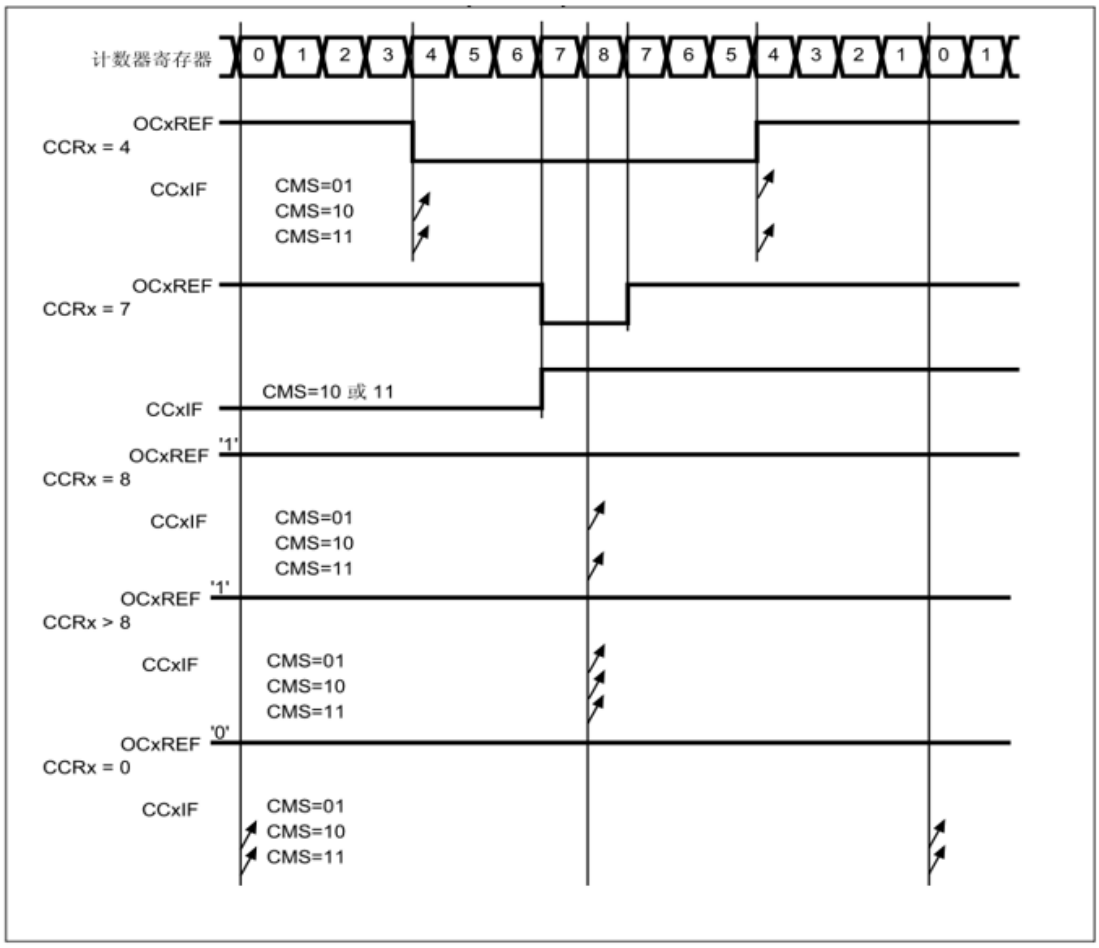
图 104 显示了中心对齐模式的 PWM 波形，在此例中：

● PWMx_ARR=8，

● PWM 模式为 PWM 模式 1，

● 在根据 PWMx_CR1 寄存器中 CMS=01 而选择的中心对齐模式 1 下，当计数器递减计数时，比较标志置 1

图 104. 中心对齐模式 PWM 波形 (ARR=8)



中心对齐模式使用建议：

● 启动中心对齐模式时将使用当前的递增/递减计数配置。这意味着计数器将根据写入 PWMx_CR1 寄存器中 DIR 位的值进行递增或递减计数。此外，不得同时通过软件修改 DIR 和 CMS 位。

● 不建议在运行中心对齐模式时对计数器执行写操作，否则将发生意想不到的结果。尤 其是：

— 如果写入计数器中的值大于自动重载值 (PWMx_CNT>PWMx_ARR)，计数方向不会 更新。例如，如果计数器之前递增计数，则继续递增计数。

— 如果向计数器写入 0 或 PWMx_ARR 的值，计数方向会更新，但不生成更新事件 UEV。

● 使用中心对齐模式为保险的方法是：在启动计数器前通过软件生成更新（将 PWMx_EGR 寄存器中的 UG 位置 1），并且不要在计数器运行过程中对其执行写操作。

15.3.11 互补输出和死区插入

高级控制定时器（PWM1 和 PWM8）可以输出两路互补信号，并管理输出的关断与接通瞬间。这段时间通常称为死区，用户必须根据与输出相连接的器件及其特性（电平转换器的固有延 迟、开关器件产生的延迟...）来调整死区时间

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

互补信号 OCx 和 OCxN 通过以下多个控制位的组合进行激活：PWMx_CCER 寄存器中的 CCxE 和 CCxNE 位以及 PWMx_BDTR 和 PWMx_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位。更多详细信息，请参见表 73 ：具有断路功能的互补通道 OCx 和 OCxN 的输出控制位 。应当注意，切换至 IDLE（MOE 下降到 0）的时刻，死区仍然有效。

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

● 输出信号 OCx 与参考信号相同，只是其上升沿相对参考上升沿存在延迟。

● 输出信号 OCxN 与参考信号相反，并且其上升沿相对参考下降沿存在延迟。

如果延迟时间大于有效输出（OCx 或 OCxN）的宽度，则不会产生相应的脉冲。 下图所示为死区发生器的输出信号与参考信号 OCxREF 之间的关系。(在这些示例中，假定 CCxP=0、CCxNP=0、MOE=1、CCxE=1 并且 CCxNE=1）

图 105. 带死区插入的互补输出。

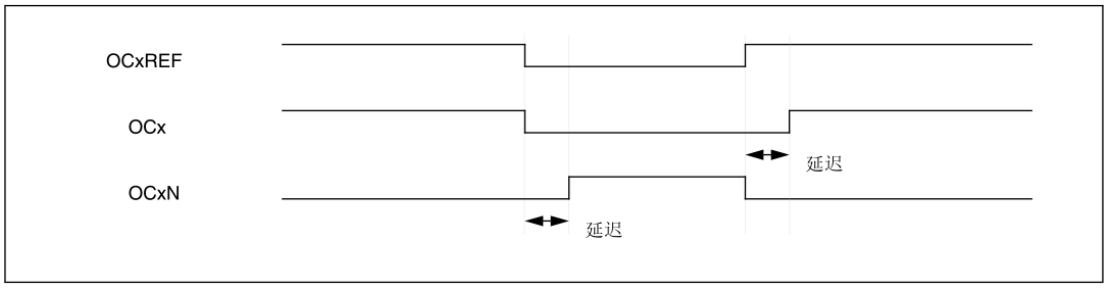


图 106. 延迟时间大于负脉冲宽度的死区波形

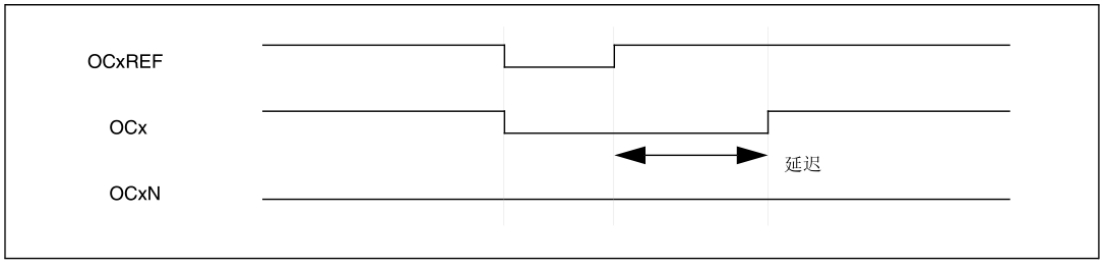
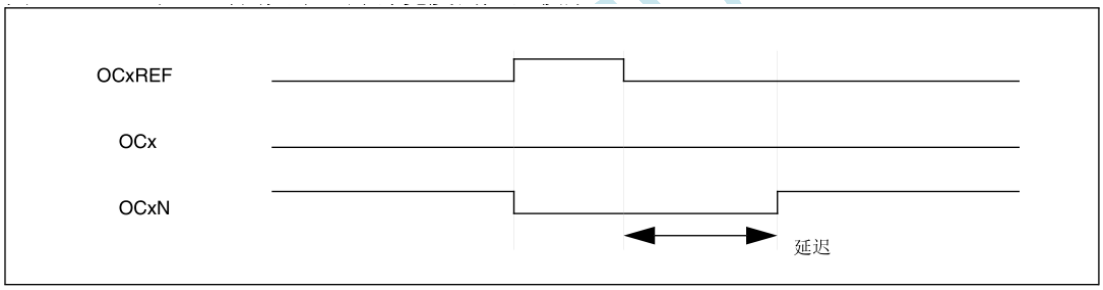


图 107. 延迟时间大于正脉冲宽度的死区波形



死区延迟对于所有通道均相同，可通过 PWMx_BDTR 寄存器中的 DTG 位进行编程。有关延迟时间计算的信息，请参见： PWM1 和 PWM8 断路和死区寄存器 (PWMx_BDTR)。

将 OCxREF 重定向到 OCx 或 OCxN

在输出模式（强制输出模式、输出比较模式或 PWM 模式）下，通过配置 PWMx_CCER 寄存器中的 CCxE 和 CCxNE 位，可将 OCxREF 重定向到 OCx 输出或 OCxN 输出。

通过此功能，可以在一个输出上发送特定波形（如 PWM 或静态有效电平），而同时使互补输出保持其无效电平。或者，使两个输出同时保持无效电平，或者两个输出同时处于有效电 平，两者互补并且带死区。

注 意 ： 如 果 仅 使 能 OCxN (CCxE=0, CCxNE=1) ，两者不互补，一旦 OCxREF 为高电平， OCxN 即变为有效。例如，如果 CCxNP=0 ，则 OCxN=OCxRef 。另一方面，如果同时使能 OCx 和 OCxN (CCxE=CCxNE=1) ， OCx 在 OCxREF 为高电平时变为有效，而 OCxN 则与之互补，在 OCxREF 为低电平时变为有效。

15.3.12 使用断路功能

使用断路功能时，根据其它控制位（PWMx_BDTR 寄存器中的 MOE、OSSI 和 OSSR 位以及 PWMx_CR2 寄存器中的 OISx 和 OISxN 位）修改输出使能信号和无效电平。任何情况下 OCx 和 OCxN 输出都不能同时置为有效电平。更多详细信息，请参见表 73 ：具有断路功能的互补通道 OCx 和 OCxN 的输出控制位 。

断路源可以是断路输入引脚，也可以是时钟故障事件，后者由复位时钟控制器中的时钟安全系 统 (CSS) 生成。有关时钟安全系统的详细信息，请参见 第 8.2.7 节：时钟安全系统 (CSS)。

退出复位状态后，断路功能处于禁止状态，MOE 位处于低电平。将 PWMx_BDTR 寄存器 中的 BKE 位置 1，可使能断路功能。断路输入的极性可通过该寄存器中的 BKP 位来选择。BKE 和 BKP 位可同时修改。对 BKE 和 BKP 位执行写操作时，写操作会在 1 个 APB 时钟周期的延迟

后生效。因此，执行写操作后，需要等待 1 个 APB 时钟周期，才能准确回读该位

由于 MOE 下降沿可能是异步信号，因此在实际信号（作用于输出）与同步控制位（位于 PWMx_BDTR 寄存器中）之间插入了再同步电路，从而在异步信号与同步信号之间产生延 迟。具体而言，如果在 MOE 处于低电平时向其写入 1，则必须首先插入延迟（空指令）。才能准确进行读取。这是因为写入的是异步信号，而读取的却是同步信号。

发生断路（断路输入上出现所选电平）时：

● MOE 位异步清零，使输出处于无效状态、空闲状态或复位状态（通过 OSSI 位进行选 择）。即使 MCU 振荡器关闭，该功能仍然有效。

● MOE=0 时，将以 PWMx_CR2 寄存器 OISx 位中编程的电平驱动每个输出通道。如果 OSSI=0，则定时器将释放使能输出，否则使能输出始终保持高电平。

● 使用互补输出时：

— 输出首先置于复位状态或无效状态（取决于极性）。这是异步操作，因此即使没有 为定时器提供时钟，该操作仍有效。

— 如果定时器时钟仍存在，则将重新激活死区发生器，进而在死区后以 OISx 和 OISxN 位中编程的电平驱动输出。即使在这种情况下，也不能同时将 OCx 和 OCxN 驱动至其有效电平。请注意，MOE 进行再同步，因此死区的持续时间会比通常情况长 一些（约 2 个 ck_PWM 时钟周期）。

— 如果 OSSI=0，则定时器会释放使能输出，否则只要 CCxE 位或 CCxNE 位处于高电平，使能输出就会保持或变为高电平。

● 将断路状态标志（PWMx_SR 寄存器中的 BIF 位）置 1。如果 PWMx_DIER 寄存器中的 BIE 位置 1，可产生中断。如果 PWMx_DIER 寄存器中的 BDE 位置 1，可发送 DMA 请求。

● 如果 PWMx_BDTR 寄存器中的 AOE 位置 1，则 MOE 位会在发生下一更新事件 (UEV)时自动再次置 1。这一特性有许多用处，比如，可用于实现调节器的功能。否则，MOE 将始终保持低电平，直到再次向该位写入“1”。这种情况下，这一特性可用于确保安 全。

注 意 ： 断 路 输 入 为 电 平 有 效 。 因 此 ， 当 断 路 输 入 有 效 电 平 时 ， 不 能 将 MOE 位置 1（自动或通过软 件）。同时，不能将状态标志 BIF 清零。

断路可由 BRK 输入生成，该输入具有可编程极性，其使能位 BKE 位于 PWMx_BDTR 寄存器中。

断路有以下两种生成方案：

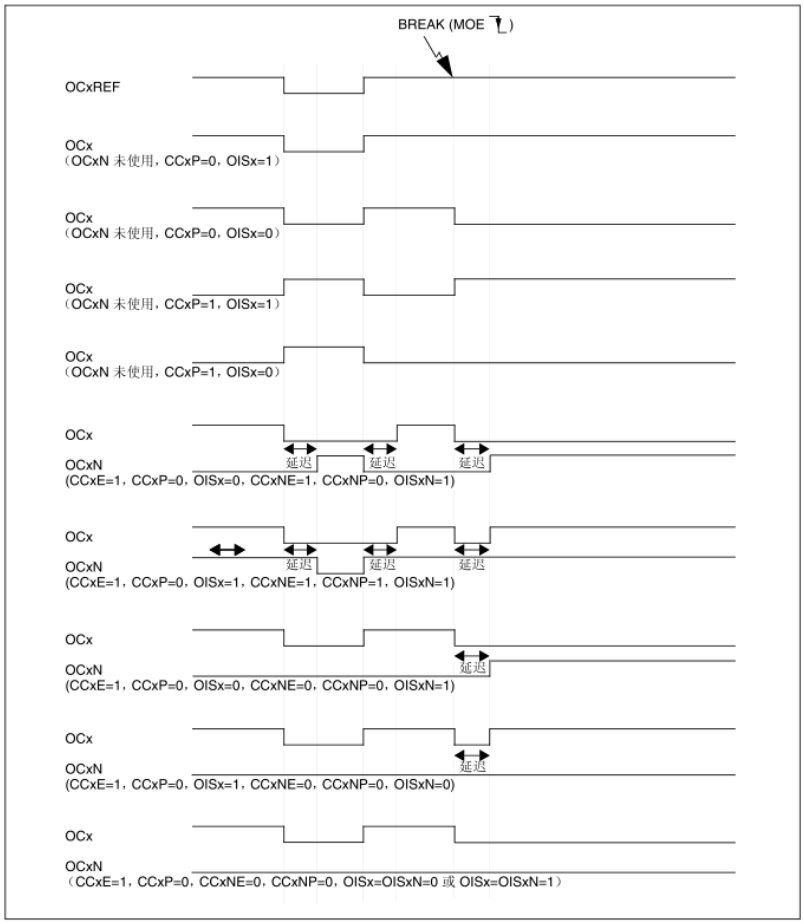
● 使用 BRK 输入生成，该输入具有可编程极性，其使能位 BKE 位于 PWMx_BDTR 寄存器中。

● 由软件通过 PWMx_EGR 寄存器中的 BG 位生成。

除断路输入和输出管理外，断路电路内部还实施了写保护，用以保护应用的安全。通过该功能，用户可冻结多个参数配置（死区持续时间、OCx/OCxN 极性和禁止时的状态、OCxM 配置、断路使能和极性）。可以通过 PWMx_BDTR 寄存器中的 LOCK 位，从 3 种保护级别中进行选择。请参见： PWM1 和 PWM8 断路和死区寄存器 (PWMx_BDTR)。MCU 复位后只能对 LOCK 位执行一次写操作。

图 108 所示为输出对断路响应行为的示例

图 108. 输出的断路响应行为。



15.3.13发生外部事件时清除 OCxREF 信号

对于给定通道，在 ETRF 输入施加高电平（相应 PWMx_CCMRx 寄存器中的 OCxCE 使能位置“1”），可使 OCxREF 信号变为低电平。OCxREF 信号将保持低电平，直到发生下一更新事件(UEV)。

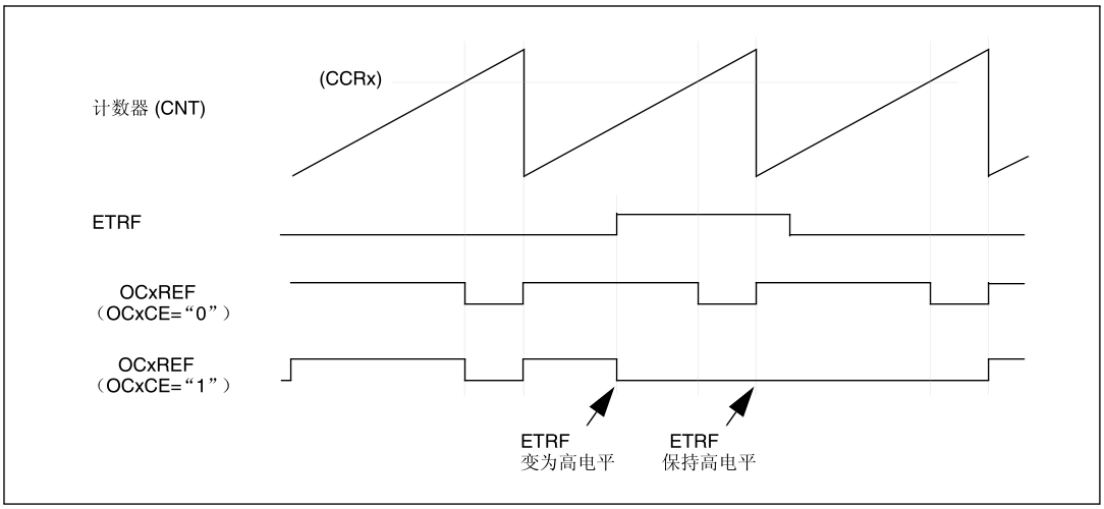
此功能仅能用于输出比较模式和 PWM 模式，而不适用于强制输出模式。

例如，ETR 信号可以连接到比较器的输出，用于控制电流。此时，ETR 必须如下配置：

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

图 109 对比了使能位 OCxCE 在不同值下的情况，显示了当 ETRF 输入变为高电平时 OCxREF 信号的行为。在本例中，定时器 PWMx 编程为 PWM 模式。

图 109. 清除 PWMx 的 OCxREF



注 意 ： 如 果 PWM 的占空比为 100% (CCRx>ARR) ， 则下次计数器溢出时会再次使能 OCxREF

15.3.14生成 6 步 PWM

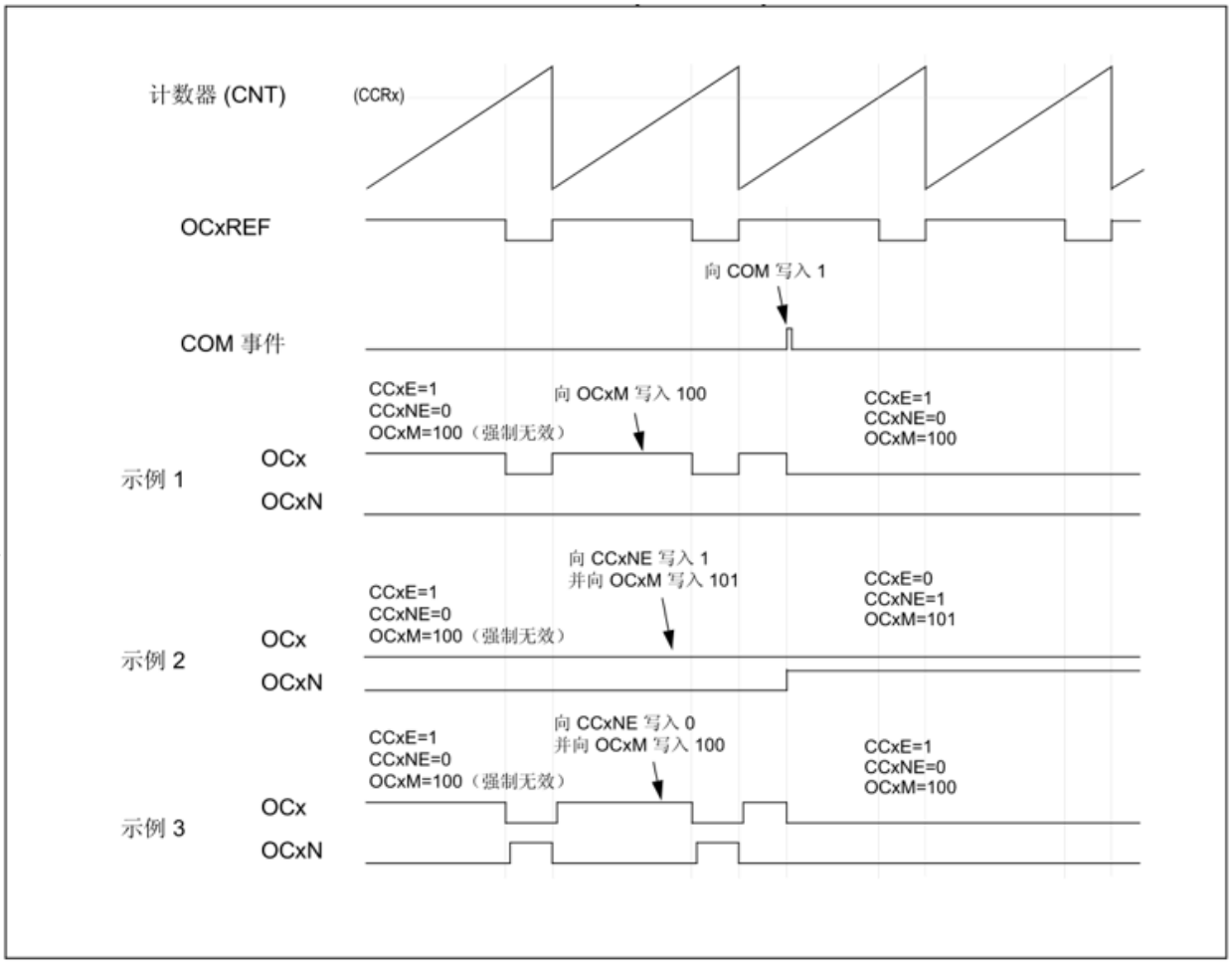
当通道使用互补输出时，OCxM、CCxE 和 CCxNE 位上提供预装载位。发生 COM 换向事件时，这些预装载位将传输到影子位。因此，用户可以预先编程下一步骤的配置，并同时更改 所有通道的配置。COM 可由软件通过将 PWMx_EGR 寄存器中的 COM 位置 1 而生成，也可以由硬件在 TRGI 上升沿生成。

发生 COM 事件时，某个标志位（PWMx_SR 寄存器中的 COMIF 位）将会置 1。这时，如果 PWMx_DIER 寄存器中的 COMIE 位置 1，将产生中断；如果 PWMx_DIER 寄存器中的 COMDE 位置 1，则将产生 DMA 请求。

图 110 以 3 种不同的编程配置为例，显示了发生 COM 事件时 OCx 和 OCxN 输出的行为。

图 110. COM 事件生成 6 步 PWM 的示例 (OSSR=1)

STC MCU



15.3.15 单脉冲模式

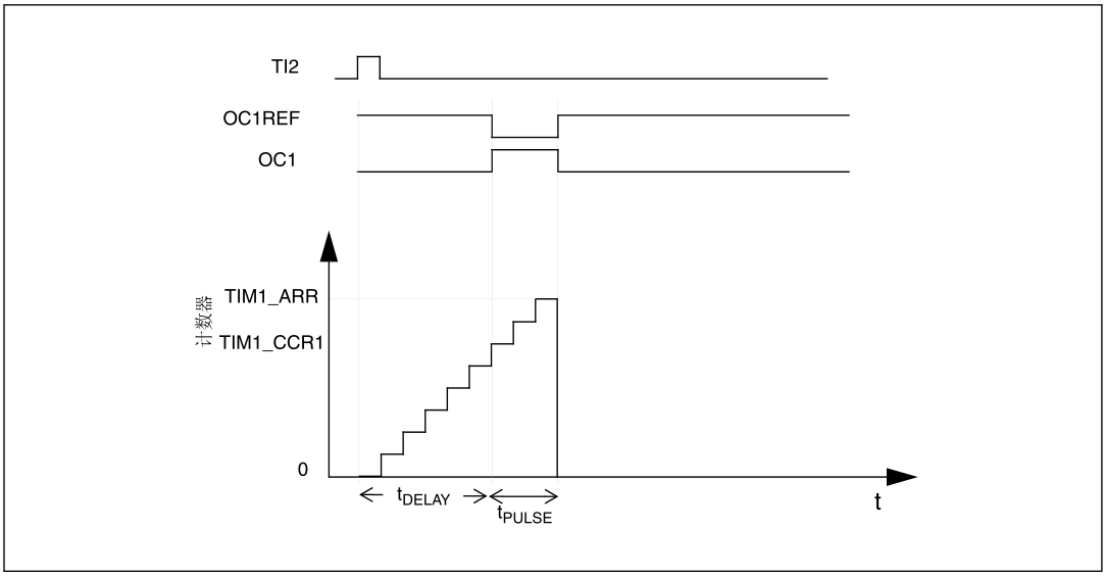
单脉冲模式 (OPM) 是上述模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。将 PWMx_CR1 寄存器中的 OPM 位置 1，即可选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

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

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

图 111. 单脉冲模式示例：



例如，用户希望达到这样的效果：在 TI2 输入引脚检测到正沿时，经过 tDELAY 的延迟，在 OC1 上产生一个长度为 tPULSE 的正脉冲。

使用 TI2FP2 作为触发 1：

- 在 PWMx_CCMR1 寄存器中写入 CC2S="01"，以将 TI2FP2 映射到 TI2。
- 在 PWMx_CCER 寄存器中写入 CC2P="0"和 CC2NP="0"，使 TI2FP2 能够检测上升沿。
- 在 PWMx_SMCR 寄存器中写入 TS="110"，以将 TI2FP2 配置为从模式控制器的触发 (TRGI)。
- 在 PWMx_SMCR 寄存器中写入 SMS="110"（触发模式），以使用 TI2FP2 启动计数器。OPM 波形通过对比较寄存器执行写操作来定义（考虑时钟频率和计数器预分频器）。
- tDELAY 由写入 PWMx_CCR1 寄存器的值定义。
- tPULSE 由自动重载值与比较值 (PWMx_ARR - PWMx_CCR1) 之差来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从"0"变为"1"，在计数器达到自 动重载值时由"1"变为"0"。为此，应在 PWMx_CCMR1 寄存器中写入 OC1M=111， 以使能 PWM 模式 2。如果需要，可选择在 PWMx_CCMR1 寄存器的 OC1PE 和 PWMx_CR1 寄存器的 ARPE 中写入"1"，以使能预装载寄存器。这种情况下，必须在 PWMx_CCR1 寄存器中写入比较值并在 PWMx_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI2 上的外部触发事件。本例中，CC1P 的值为"0"。

在本例中，PWMx_CR1 寄存器中的 DIR 和 CMS 位应为低。

由于仅需要 1 个脉冲（单脉冲模式），因此应向 PWMx_CR1 寄存器的 OPM 位写入"1"，以便在发生下一更新事件（计数器从自动重载值返回到 0）时使计数器停止计数。PWMx_CR1 寄存器中的 OPM 位置"0"时，即选择重复模式。

特例：OCx 快速使能

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

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

15.3.16 编码器接口模式

选择编码器接口模式时，如果计数器仅在 TI2 边沿处计数，在 PWMx_SMCR 寄存器中写入 SMS="001"；如果计数器仅在 TI1 边沿处计数，写入 SMS="010"；如果计数器在 TI1 和 TI2 边沿处均计数，则写入 SMS="011"。

通过编程 PWMx_CCER 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编程。CC1NP 和 CC2NP 必须保持低电平。

TI1 和 TI2 两个输入用于连接增量编码器。请参见 表 71。如果使能计数器（在 PWMx_CR1 寄

存器的 CEN 位中写入“1”），则计数器的时钟由 TI1FP1 或 TI2FP2 上的每次有效信号转换提供。TI1FP1 和 TI2FP2 是进行输入滤波器和极性选择后 TI1 和 TI2 的信号，如果不进行滤波和反相，则 TI1FP1=TI1，TI2FP2=TI2。将根据两个输入的信号转换序列，产生计数脉 冲和方向信号。根据该信号转换序列，计数器相应递增或递减计数，同时硬件对 PWMx_CR1 寄存器的 DIR 位进行相应修改。任何输入（TI1 或 TI2）发生信号转换时，都会计算 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。

编码器接口模式就相当于带有方向选择的外部时钟。这意味着，计数器仅在 0 到 PWMx_ARR 寄存器中的自动重载值之间进行连续计数（根据具体方向，从 0 递增计数到 ARR，或从 ARR 递减计数到 0）。因此，在启动前必须先配置 PWMx_ARR。同样，捕获、比较、预分频器、重复计数器及触发输出功能继续正常工作。编码器模式和外部时钟模式 2 不兼容，因此不能同时选择。在此模式下，计数器会根据增量编码器的速度和方向自动进行修改，因此，其内容始终表示 编码器的位置。计数方向对应于所连传感器的旋转方向。下表汇总了可能的组合（假设 TI1 和 TI2 不同时切换）。

表 71. 计数方向与编码器信号的关系

有效边沿	相反信号的电平（**TI1FP1** 对应 **TI2, TI2FP2** 对应 **TI1**）	**TI1FP1** 信号		**TI2FP2** 信号	
		上升	下降	上升	下降
仅在 TI1 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
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
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

外部增量编码器可直接与 MCU 相连，无需外部接口逻辑。不过，通常使用比较器将编码器的差分输出转换为数字信号。这样大幅提高了抗噪声性能。用于指示机械零位的第三个编码 器输出可与外部中断输入相连，用以触发计数器复位。

图 112 以计数器工作为例，说明了计数信号的生成和方向控制。同时也说明了选择双边沿时如何对输入抖动进行补偿。将传感器靠近其中一个切换点放置时可能出现这种情况。本例中 假设配置如下：

- CC1S=“01”（PWMx_CCMR1 寄存器，TI1FP1 映射到 TI1 上）。
- CC2S=“01”（PWMx_CCMR2 寄存器，TI1FP2 映射到 TI2 上）。
- CC1P=“0”，CC1NP=“0”，且 IC1F =“0000”（PWMx_CCER 寄存器，TI1FP1 未反 相，TI1FP1=TI1）。
- CC2P=“0”，CC2NP=“0”，且 IC2F =“0000”（PWMx_CCER 寄存器，TI1FP2 未反 相，TI1FP2=TI2）。
- SMS=“011”（PWMx_SMCR 寄存器，两个输入在上升沿和下降沿均有效）。
- CEN=“1”（PWMx_CR1 寄存器，使能计数器）。

图 112. 编码器接口模式下的计数器工作示例。

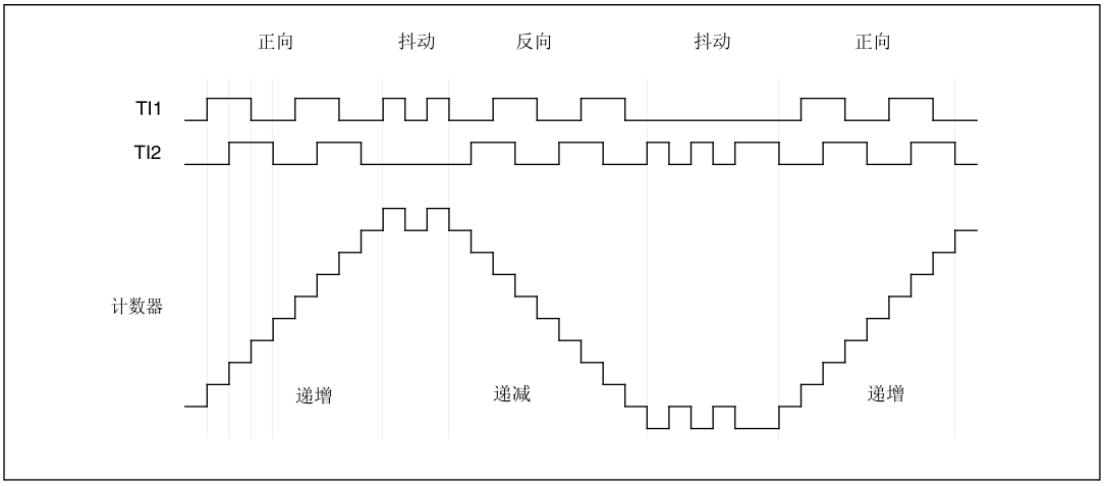
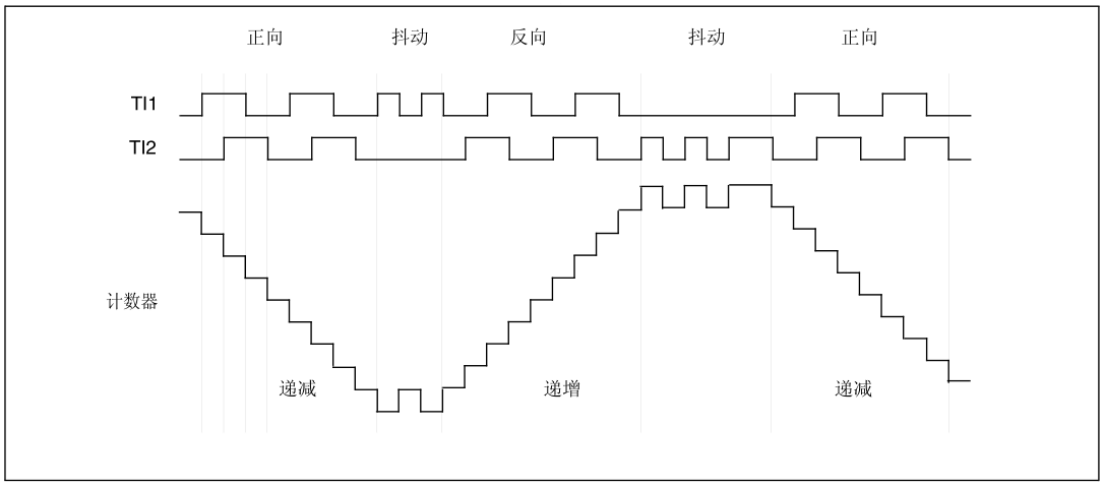


图 113 举例说明 TI1FP1 极性反相时计数器的行为（除 CC1P="1" 外，其它配置与上例相同）。

图 113. TI1FP1 极性反相时的编码器接口模式示例。



定时器配置为编码器接口模式时，会提供传感器当前位置的相关信息。使用另一个配置为捕 获模式的定时器测量两个编码器事件之间的周期，可获得动态信息（速度、加速度和减速 度）。指示机械零位的编码器输出即可用于此目的。根据两个事件之间的时间间隔，还可定 期读取计数器。如果可能，可以将计数器值锁存到第三个输入捕获寄存器来实现此目的（捕 获信号必须为周期性信号，可以由另一个定时器产生）；还可以通过由实时时钟生成的 DMA 请求读取计数器值。

15.3.17 定时器输入异或功能

通过 PWMx_CR2 寄存器中的 TI1S 位，可将通道 1 的输入滤波器连接到异或门的输出，从而将 PWMx_CH1 到 PWMx_CH3 这三个输入引脚组合在一起。

异或输出可与触发或输入捕获等所有定时器输入功能配合使用。

15.3.18 连接霍尔传感器

可通过用于生成电机驱动 PWM 信号的高级控制定时器（PWM1 或 PWM8）以及 图 114 中称为 "接口定时器"的另一个定时器 PWMx（PWM2、PWM3、PWM4 或 PWM5），实现与霍尔传感器 的连接。3 个定时器输入引脚（PWMx_CH1、PWMx_CH2 和 PWMx_CH3）通过异或门连接到 TI1 输入通道（通过将 PWMx_CR2 寄存器中的 TI1S 位置 1 来选择），并由"接口定时器"进行捕获。

从模式控制器配置为复位模式；从输入为 TI1F_ED。这样，每当 3 个输入中有一个输入发生切换时，计数器会从 0 开始重新计数。这样将产生由霍尔输入的任何变化而触发的时基。

在"接口定时器"上，捕获/比较通道 1 配置为捕获模式，捕获信号为 TRC（请参见图 97 ：捕获 / 比较通道（例如：通道 1 输入阶段） ）。捕获值对应于输入上两次变化的间隔时间可提供与电机转速相关的信息。

"接口定时器"可用于在输出模式下产生脉冲，以通过触发 COM 事件更改高级控制定时器（PWM1 或 PWM8）各个通道的配置。PWM1 定时器用于生成电机驱动 PWM 信号。为此，必

须对接口定时器通道进行编程，以便在编程的延迟过后产生正脉冲（在输出比较或 PWM 模 式中）。该脉冲通过 TRGO 输出发送到高级控制定时器（PWM1 或 PWM8）。

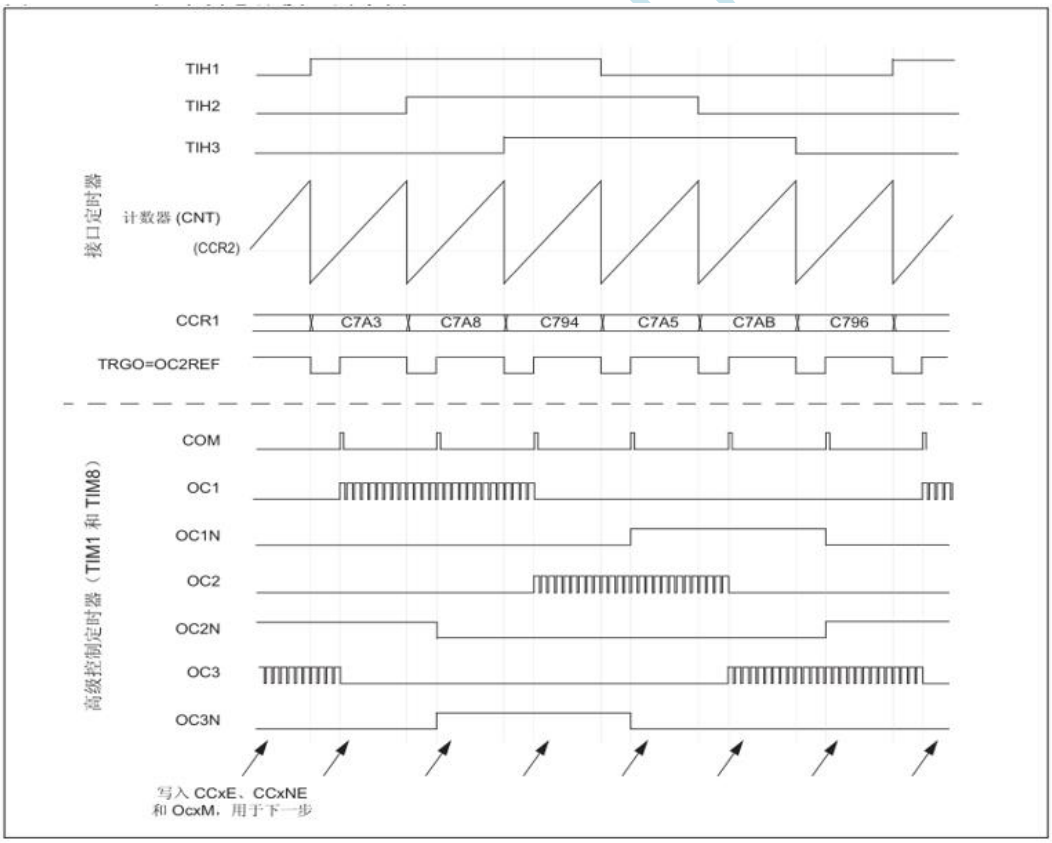
示例：霍尔输入与一个 PWMx 定时器相连接，每当霍尔输入发生更改，需要在所编程的延迟过后更改高级控制定时器 PWM1 的 PWM 配置。

- 向 PWMx_CR2 寄存器的 TI1S 位写入“1”，使 3 个定时器输入经过异或运算后进入 TI1 输入通道。
- 时基编程：向 PWMx_ARR 写入其大值（计数器必须通过 TI1 的变化清零）。设置预分频器，以得到大计数器周期，该周期长于传感器上两次变化的间隔时间。
- 将通道 1 编程为捕获模式（选择 TRC）：向 PWMx_CCMR1 寄存器的 CC1S 位写入 “11”。如果需要，还可以编程数字滤波器。
- 将通道 2 编程为 PWM 2 模式，并具有所需延迟：向 PWMx_CCMR1 寄存器的 OC2M 位写入“111”，CC2S 位写入“00”。
- 选择 OC2REF 作为 TRGO 上的触发输出：向 PWMx_CR2 寄存器的 MMS 位写入“101”。

在高级控制定时器 PWM1 中，必须选择正确的 ITR 输入作为触发输入，定时器编程为可产 PWM 信号，捕获/比较控制信号进行预装载（PWMx_CR2 寄存器的 CCPC=1），并且 COM 事件由触发输入控制（PWMx_CR2 寄存器中 CCUS=1）。发生 COM 事件后，在 PWM 控制位（CCxE、OCxM）中写入下一步的配置，此操作可在由 OC2REF 上升沿产生的中断子程序中完成。

图 114 为本示例的示意图。

图 114. 霍尔传感器接口的示例



15.3.19 PWMx 与外部触发同步

PWMx 定时器可与外部触发以下列模式实现同步：复位模式、门控模式和触发模式。

从模式：复位模式

当触发输入信号产生变化时，计数器及其预分频器可重新初始化。此外，如果 PWMx_CR1 寄 存器中的 URS 位处于低电平，则会生成更新事件 UEV。然后，所有预装载寄存器（PWMx_ARR 和 PWMx_CCRx）都将更新。

在以下示例中，TI1 输入上出现上升沿时，递增计数器清零：

● 将通道 1 配置为检测 TI1 的上升沿。配置输入滤波时间（本例中不需要任何滤波，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位 只选择输入捕获源，即 PWMx_CCMR1 寄存器中的 CC1S = 01。在 PWMx_CCER 寄存器中写入 CC1P=0 和 CC1NP='0'，验证极性（仅检测上升沿）。

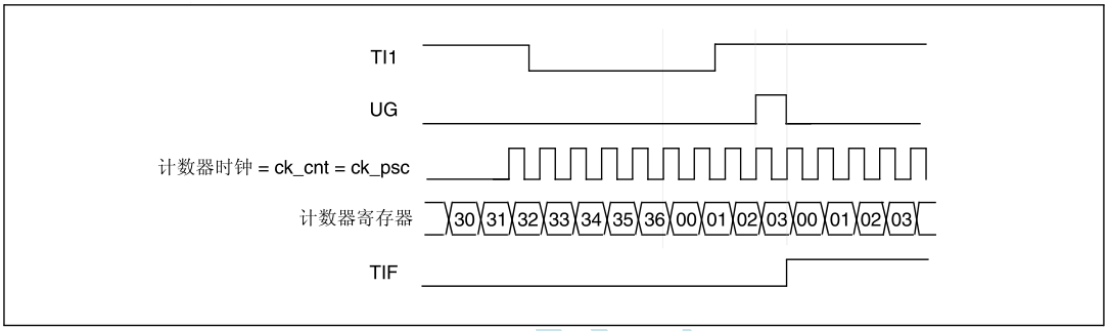
● 在 PWMx_SMCR 寄存器中写入 SMS=100，将定时器配置为复位模式。在 PWMx_SMCR 寄存器中写入 TS=101，选择 TI1 作为输入源。

● 在 PWMx_CR1 寄存器中写入 CEN=1，启动计数器。

计数器使用内部时钟计数，然后正常运转，直到出现 TI1 上升沿。当 TI1 出现上升沿时，计数器清零，然后重新从 0 开始计数。同时，触发标志（PWMx_SR 寄存器中的 TIF 位）置 1，使能中断或 DMA 后，还可发送中断或 DMA 请求（取决于 PWMx_DIER 寄存器中的 TIE 和 TDE 位）。

下图显示了自动重载寄存器 PWMx_ARR=0x36 时的相关行为。TI1 的上升沿与实际计数器复位之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 115. 复位模式下的控制电路



从模式：门控模式

输入信号的电平可用来使能计数器。

在以下示例中，递增计数器仅在 TI1 输入为低电平时计数：

● 将通道 1 配置为检测 TI1 上的低电平。配置输入滤波时间（本例中不需要任何滤波，因 此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。 CC1S 位只选择输入捕获源，即 PWMx_CCMR1 寄存器中的 CC1S=01。在 PWMx_CCER 寄存器中写入 CC1P=1 和 CC1NP="0"，以确定极性（仅检测低电平）。

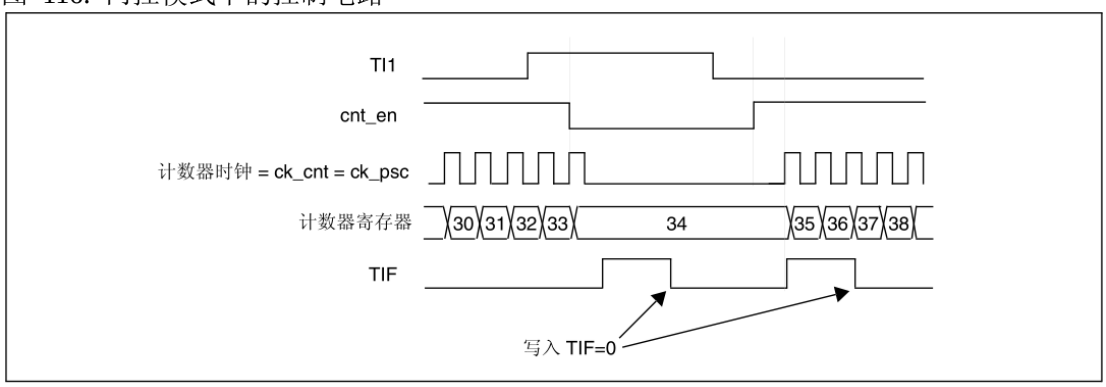
● 在 PWMx_SMCR 寄存器中写入 SMS=101，将定时器配置为门控模式。在 PWMx_SMCR 寄存器中写入 TS=101，选择 TI1 作为输入源。

● 在 PWMx_CR1 寄存器中写入 CEN=1，使能计数器（在门控模式下，如果 CEN=0，则 无论触发输入电平如何，计数器都不启动）。

只要 TI1 为低电平，计数器就开始根据内部时钟计数，直到 TI1 变为高电平时停止计数。计数器启动或停止时，PWMx_SR 寄存器中的 TIF 标志都会置 1。

TI1 的上升沿与实际计数器停止之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 116. 门控模式下的控制电路

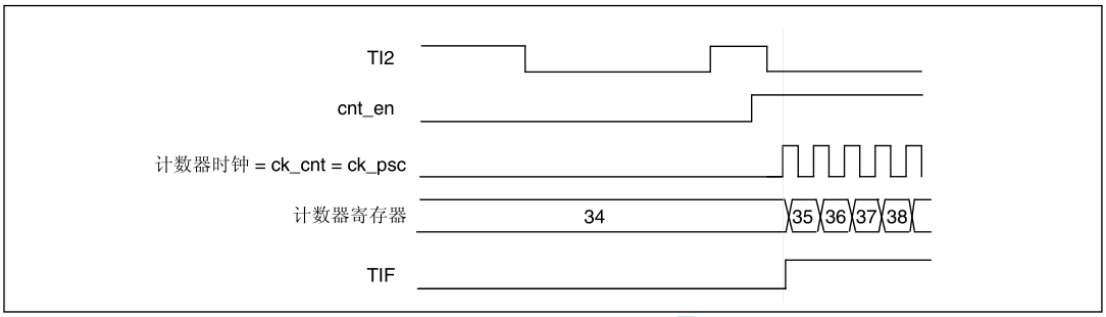


从模式：触发模式

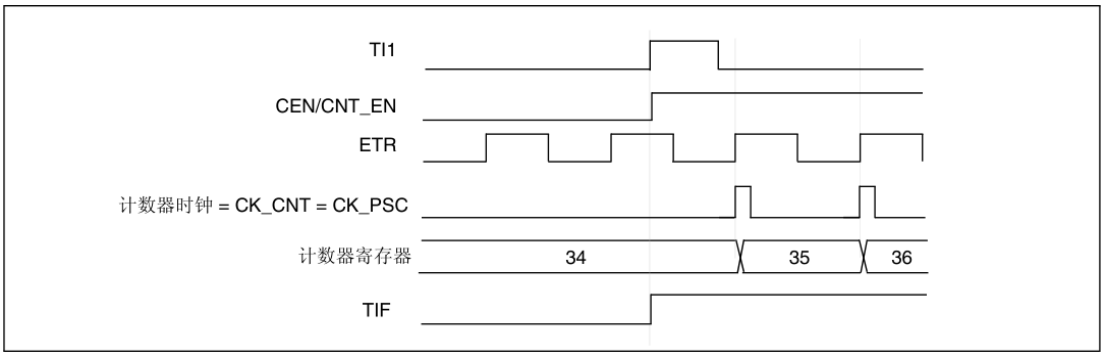
所选输入上发生某一事件时可以启动计数器。
在以下示例中，TI2 输入上出现上升沿时，递增计数器启动：

- 将通道 2 配置为检测 TI2 上的上升沿。配置输入滤波时间（本例中不需要任何滤波，因此保持 IC2F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。 CC2S 位只选择输入捕获源，即 PWMx_CCMR1 寄存器中的 CC2S=01。在 PWMx_CCER 寄存器中写入 CC2P=1 和 CC2NP=0，以确定极性（仅检测低电平）。
- 在 PWMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 PWMx_SMCR 寄存器中写入 TS=110，选择 TI2 作为输入源。

当 TI2 出现上升沿时。计数器开始根据内部时钟计数，并且 TIF 标志置 1。
TI2 的上升沿与实际计数器启动之间的延迟是由 TI2 输入的重新同步电路引起的。
图 117. 触发模式下的控制电路



从模式：外部时钟模式 2 + 触发模式
外部时钟模式 2 可与另一种从模式（外部时钟模式 1 和编码器模式除外）结合使用。这种情况下，ETR 信号用作外部时钟输入，在复位模式、门控模式或触发模式下工作时，可选择另 一个输入作为触发输入。不建议通过 PWMx_SMCR 寄存器中的 TS 位来选择 ETR 作为 TRGI。
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递增：
1. 通过对 PWMx_SMCR 寄存器进行如下编程，配置外部触发输入电路：
— ETF = 0000：无滤波器
— ETPS = 00：禁止预分频器
— ETP = 0 ：检测 ETR 的上升沿，并写入 ECE=1，以使能外部时钟模式 2。 2. 如下配置通道 1，以检测 TI 的上升沿：
— IC1F=0000：无滤波器。
— 由于捕获预分频器不用于触发操作，因此无需对其进行配置。
—PWMx_CCMR1 寄存器中 CC1S=01，只选择输入捕获源。
— PWMx_CCER 寄存器中 CC1P=0 且 CC1NP="0"，以确定极性（仅检测上升沿）。
3. 在 PWMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 PWMx_SMCR 寄存器中写入 TS=101，选择 TI1 作为输入源。
TI1 出现上升沿时将使能计数器并且 TIF 标志置 1。然后计数器在 ETR 出现上升沿时计数。
ETR 信号的上升沿与实际计数器复位之间的延迟是由于 ETRP 输入的重新同步电路引起的。
图 118. 外部时钟模式 2 + 触发模式下的控制电路



15.3.20 定时器同步

PWM 定时器从内部链接在一起，以实现定时器同步或级联。有关详细信息，请参见：定时器同

步。

15.3.21 调试模式

当微控制器进入调试模式（Cortex™-MSTAR 内核停止）时，PWMx 计数器会根据 DBG 模块中的 DBG_PWMx_STOP 配置位选择继续正常工作或者停止工作。

15.4 PWM1 和 PWM8 寄存器

有关寄存器说明中使用的缩写，请参见第 1.1 节 。

外设寄存器必须按半字（16 位）或字（32 位）进行写访问。而读访问则可按字节（8 位）、半字（16 位）或字（32 位）进行。

15.4.1 PWM1 和 PWM8 控制寄存器 1 (PWMx_CR1)

PWM1&PWM8 control register 1

偏移地址：0x00

复位值：0x0000

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

位 15:10 保留，必须保持复位值。

位 9:8 CKD[1:0]：时钟分频 (Clock division) 此位域指示定时器时钟 (CK_INT) 频率与死区发生器以及数字滤波器（ETR、TIx）所使用的 死区及采样时钟 (tDTS) 之间的分频比，

00：tDTS = tCK_INT

01：tDTS = 2 * tCK_INT

10：tDTS = 4 * tCK_INT

11：保留，不要设置成此值

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

0：PWMx_ARR 寄存器不进行缓冲

1：PWMx_ARR 寄存器进行缓冲

位 6:5 CMS[1:0]：中心对齐模式选择 (Center-aligned mode selection)

00：边沿对齐模式。计数器根据方向位 (DIR) 递增计数或递减计数。

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

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

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

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

位 4 DIR：方向 (Direction)。

0：计数器递增计数

1：计数器递减计数

注意：当定时器配置为中心对齐模式或编码器模式时，该位为只读状态

位 3 OPM：单脉冲模式 (One pulse mode)

0：计数器在发生更新事件时不会停止计数

1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）

位 2 URS：更新请求源 (Update request source)

此位由软件置 1 和清零，用以选择 UEV 事件源。

0： 使能时，所有以下事件都会生成更新中断或 DMA 请求。此类事件包括：

— 计数器上溢/下溢

— 将 UG 位置 1

— 通过从模式控制器生成的更新事件

1：使能时，只有计数器上溢/下溢会生成更新中断或 DMA 请求。

位 1 UDIS：更新禁止 (Update disable)

此位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。

0：使能 UEV。更新 (UEV) 事件可通过以下事件之一生成：

— 计数器上溢/下溢

— 将 UG 位置 1

— 通过从模式控制器生成的更新事件

然后更新影子寄存器的值。

1：禁止 UEV。不会生成更新事件，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。但如果将 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。

位 0 CEN：计数器使能 (Counter enable)

0：禁止计数器

1：使能计数器

注意：只有事先通过软件将 CEN 位置 1，才可以使用外部时钟、门控模式和编码器模式。而 触发模式可通过硬件自动将 CEN 位置 1

15.4.2 PWM1 和 PWM8 控制寄存器 2 (PWMx_CR2)

PWM1&PWM8 control register 2

偏移地址：0x04

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OS1S	TI1S	MMS[2:0]			CCOS	CCUS	Res.	CCPC
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

位 15 保留，必须保持复位值。

位 14 OIS4：输出空闲状态 4（OC4 输出）(Output Idle state 4 (OC4 output)) 参见 OIS1 位 位 13 OIS3N：输出空闲状态 3（OC3N 输出）(Output Idle state 3 (OC3N output)) 参见 OIS1N 位

位 12 OIS3：输出空闲状态 3（OC3 输出）(Output Idle state 3 (OC3 output)) 参见 OIS1 位 位 11 OIS2N：输出空闲状态 2（OC2N 输出）(Output Idle state 2 (OC2N output)) 参见 OIS1N 位

位 10 OIS2：输出空闲状态 2（OC2 输出）(Output Idle state 2 (OC2 output)) 参见 OIS1 位 位 9 OIS1N：输出空闲状态 1（OC1N 输出）(Output Idle state 1 (OC1N output))

0：当 MOE=0 时，经过死区时间后 OC1N=0

1：当 MOE=0 时，经过死区时间后 OC1N=1

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

位 8 OIS1：输出空闲状态 1（OC1 输出）(Output Idle state 1 (OC1 output))

0：当 MOE=0 时，（如果 OC1N 有效，则经过死区时间之后）OC1=0

1：当 MOE=0 时，（如果 OC1N 有效，则经过死区时间之后）OC1=1

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

位 7 TI1S：TI1 选择 (TI1 selection)

0：PWMx_CH1 引脚连接到 TI1 输入

1：PWMx_CH1、CH2 和 CH3 引脚连接到 TI1 输入（异或组合）

位 6:4 MMS[1:0]：主模式选择 (Master mode selection)

这些位可选择主模式下将要发送到从定时器以实现同步的信息 (TRGO)。这些位的组合如下：
000：复位—PWMx_EGR 寄存器中的 UG 位用作触发输出 (TRGO)。如果复位由触发输入生成（从模式控制器配置为复位模式），则 TRGO 上的信号相比实际复位会有延迟。
001：使能—计数器使能信号 CNT_EN 用作触发输出 (TRGO)。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号可由 CEN 控制位产生。 当配置为门控模式时，也可由触发输入产生。当计数器使能信号由触发输入控制时，TRGO 上 会存在延迟，选择主/从模式时除外（请参见 PWMx_SMCR 寄存器中 MSM 位的说明）。
010：更新—选择更新事件作为触发输出 (TRGO)。例如，主定时器可用作从定时器的预分 频器。
011：比较脉冲—一旦发生输入捕获或比较匹配事件，当 CC1IF 被置 1 时（即使已为高电平），触发输出都会发送一个正脉冲。(TRGO)。
100：比较—OC1REF 信号用作触发输出 (TRGO)
101：比较—OC2REF 信号用作触发输出 (TRGO)
110：比较—OC3REF 信号用作触发输出 (TRGO)
111：比较—OC4REF 信号用作触发输出 (TRGO)

位 3 CCDS：捕获/比较 DMA 选择 (Capture/compare DMA selection)
0：发生 CCx 事件时发送 CCx DMA 请求
1：发生更新事件时发送 CCx DMA 请求

位 2 CCUS：捕获/比较控制更新选择 (Capture/compare control update selection)
0： 如果捕获/比较控制位 (CCPC=1) 进行预装载，仅通过将 COMG 位置 1 来对这些位进行更新
1： 如果捕获/比较控制位 (CCPC=1) 进行预装载，可通过将 COMG 位置 1 或 TRGI 的上升沿对这些位进行更新。
注意：此位仅对具有互补输出的通道有效。

位 1 保留，必须保持复位值。

位 0 CCPC：捕获/比较预装载控制 (Capture/compare preloaded control)
0：CCxE、CCxNE 和 OCxM 位未进行预装载
1：CCxE、CCxNE 和 OCxM 位进行了预装载，写入这些位后，仅当发生换向事件 (COM)（COMG 位置 1 或在 TRGI 上检测到上升沿，取决于 CCUS 位）时才会对这些位进行更新。
注意：此位仅对具有互补输出的通道有效。

15.4.3 PWM1 和 PWM8 从模式控制寄存器 (PWMx_SMCR)

PWM1&PWM8 slave mode control register
偏移地址：0x08
复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			Res.	SMS[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw

位 15 ETP：外部触发极性 (External trigger polarity)
此位可选择将 ETR 还是 ETR 用于触发操作
0：ETR 未反相，高电平或上升沿有效。
1：ETR 反相，低电平或下降沿有效。

位 14 ECE：外部时钟使能 (External clock enable)
此位可使能外部时钟模式 2。
0：禁止外部时钟模式 2
1：使能外部时钟模式 2。计数器时钟由 ETRF 信号的任意有效边沿提供。
注意：1 ： 将 ECE 位置 1 与选择外部时钟模式 1 并将 TRGI 连接到 ETRF （ SMS=111 且 TS=111 ）具有相同效果。 2 ： 外部时钟模式 2 可以和以下从模式同时使用：复位模式、门控模式和触发模式。不过此类情况下 TRGI 不得连接 ETRF （ TS 位不得为 111 ）。 3 ： 如果同时使能外部时钟

模式 1 和外部时钟模式 2 ，则外部时钟输入为 ETRF 。

位 13:12 ETPS[1:0]：外部触发预分频器 (External trigger prescaler)

外部触发信号 ETRP 频率不得超过 PWMxCLK 频率的 1/4。可通过使能预分频器来降低 ETRP 频率。这种方法在输入快速外部时钟时非常有用。

00：预分频器关闭

01：2 分频 ETRP 频率

10：4 分频 ETRP 频率

11：8 分频 ETRP 频率

位 11:8 ETF[3:0]：外部触发滤波器 (External trigger filter)

此位域可定义 ETRP 信号的采样频率和适用于 ETRP 的数字滤波时间。数字滤波器由事件计数器组成，每 N 个事件才视为一个有效边沿：

0000：无滤波器，按 fDTS 频率进行采样

0001：fSAMPLING=fCK_INT，N=2

0010：fSAMPLING=fCK_INT，N=4

0011：fSAMPLING=fCK_INT，N=8

0100：fSAMPLING=fDTS/2，N=6

0101：fSAMPLING=fDTS/2，N=8

0110：fSAMPLING=fDTS/4，N=6

0111：fSAMPLING=fDTS/4，N=8

1000：fSAMPLING=fDTS/8，N=6

1001：fSAMPLING=fDTS/8，N=8

1010：fSAMPLING=fDTS/16，N=5

1011：fSAMPLING=fDTS/16，N=6

1100：fSAMPLING=fDTS/16，N=8

1101：fSAMPLING=fDTS/32，N=5

1110：fSAMPLING=fDTS/32，N=6

1111：fSAMPLING=fDTS/32，N=8

位 7 MSM：主/从模式 (Master/slave mode)

0：不执行任何操作

1：当前定时器的触发输入事件 (TRGI) 的动作被推迟，以使当前定时器与其从定时器实现完美同步（通过 TRGO）。此设置适用于由单个外部事件对多个定时器进行同步的情况。

位 6:4 TS[2:0]：触发选择 (Trigger selection)

此位域可选择将要用于同步计数器的触发输入。

000：内部触发 0 (ITR0)

001：内部触发 1 (ITR1)

010：内部触发 2 (ITR2)

011：内部触发 3 (ITR3)

100：TI1 边沿检测器 (TI1F_ED)

101：滤波后的定时器输入 1 (TI1FP1)

110：滤波后的定时器输入 2 (TI2FP2)

111：外部触发输入 (ETRF)

有关各定时器 ITRx 含义的详细信息，请参见表 72 ： PWMx 内部触发连接 。

注意：这些位只能在未使用的情况下（例如， SMS=000 时）进行更改，以避免转换时出现错 误的边沿检测。

位 3 保留，必须保持复位值。

位 2:0 SMS：从模式选择 (Slave mode selection)

选择外部信号时，触发信号 (TRGI) 的有效边沿与外部输入上所选的极性相关（请参见输入控制寄存器和控制寄存器说明）。

000：禁止从模式—如果 CEN =“1”，预分频器时钟直接由内部时钟提供。

001：编码器模式 1—计数器根据 TI1FP1 电平在 TI2FP2 边沿递增/递减计数。

010：编码器模式 2—计数器根据 TI2FP2 电平在 TI1FP1 边沿递增/递减计数。

STC MCU

011：编码器模式 3—计数器在 TI1FP1 和 TI2FP2 的边沿计数，计数的方向取决于另外一个信号的电平。

100：复位模式—在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器并生成一个寄存器更新事件。

101：门控模式—触发输入 (TRGI) 为高电平时使能计数器时钟。只要触发输入变为低电平，计数器立即停止计数（但不复位）。计数器的启动和停止都是受控的。

110：触发模式—触发信号 TRGI 出现上升沿时启动计数器（但不复位）。只控制计数器 的启动。

111：外部时钟模式 1—由所选触发信号 (TRGI) 的上升沿提供计数器时钟。

注意：如果将 TI1F_ED 选作触发输入（ TS=“ 100 ”），则不得使用门控模式。实际上， TI1F 每次转换时， TI1F_ED 都输出 1 个脉冲，而门控模式检查的则是触发信号的电平。

表 72. PWMx 内部触发连接

从 TIM	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
TIM1	TIM5	TIM2	TIM3	TIM4
TIM8	TIM1	TIM2	TIM4	TIM5

15.4.4 PWM1 和 PWM8 DMA/中断使能寄存器

(PWMx_DIER) PWM1&PWM8 DMA/interrupt enable register

偏移地址：0x0C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15 保留，必须保持复位值。

位 14 TDE：触发 DMA 请求使能 (Trigger DMA request enable)

0：禁止 DMA 请求

1：使能 DMA 请求

位 13 COMDE：COM DMA 请求使能 (COM DMA request enable)

0：禁止 COM DMA 请求

1：使能 COM DMA 请求

位 12 CC4DE：捕获/比较 4 DMA 请求使能 (Capture/Compare 4 DMA requestenable)

0：禁止 CC4 DMA 请求

1：使能 CC4 DMA 请求

位 11 CC3DE：捕获/比较 3 DMA 请求使能 (Capture/Compare 3 DMA requestenable)

0：禁止 CC3 DMA 请求

1：使能 CC3 DMA 请求

位 10 CC2DE：捕获/比较 2 DMA 请求使能 (Capture/Compare 2 DMA requestenable)

0：禁止 CC2 DMA 请求

1：使能 CC2 DMA 请求

位 9 CC1DE：捕获/比较 1 DMA 请求使能 (Capture/Compare 1 DMA request enable)

0：禁止 CC1 DMA 请求

1：使能 CC1 DMA 请求

位 8 UDE：更新 DMA 请求使能 (Update DMA request enable)

0：禁止更新 DMA 请求

1：使能更新 DMA 请求

位 7 BIE：断路中断使能 (Break interrupt enable)

0：禁止断路中断

STC MCU

1：使能断路中断

位 6 TIE：触发信号 (TGRI) 中断使能 (Trigger interrupt enable)

0：禁止触发信号 (TGRI) 中断

1：使能触发信号 (TGRI) 中断

位 5 COMIE：COM 中断使能 (COM interrupt enable)

0：禁止 COM 中断

1：使能 COM 中断

位 4 CC4IE：捕获/比较 4 中断使能 (Capture/Compare 4 interrupt enable)

0：禁止 CC4 中断

1：使能 CC4 中断

位 3 CC3IE：捕获/比较 3 中断使能 (Capture/Compare 3 interrupt enable)

0：禁止 CC3 中断

1：使能 CC3 中断

位 2 CC2IE：捕获/比较 2 中断使能 (Capture/Compare 2 interrupt enable)

0：禁止 CC2 中断

1：使能 CC2 中断

位 1 CC1IE：捕获/比较 1 中断使能 (Capture/Compare 1 interrupt enable)

0：禁止 CC1 中断

1：使能 CC1 中断

位 0 UIE：更新中断使能 (Update interrupt enable)

0：禁止更新中断

1：使能更新中断

15.4.5 PWM1 和 PWM8 状态寄存器 (PWMx_SR)

PWM1&PWM8 status register

偏移地址：0x10

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CC4OF	CC3OF	CC2OF	CC1OF	Reserved	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			rc_w0	rc_w0	rc_w0	rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位 15:13 保留，必须保持复位值。

位 12 CC4OF：捕获/比较 4 重复捕获标志 (Capture/Compare 4 overcapture flag)

请参见 CC1OF 说明

位 11 CC3OF：捕获/比较 3 重复捕获标志 (Capture/Compare 3 overcapture flag)

请参见 CC1OF 说明

位 10 CC2OF：捕获/比较 2 重复捕获标志 (Capture/Compare 2 overcapture flag)

请参见 CC1OF 说明

位 9 CC1OF：捕获/比较 1 重复捕获标志 (Capture/Compare 1 overcapture flag)

仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将 该位清零。

0：未检测到重复捕获。

1：PWMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。

位 8 保留，必须保持复位值。

位 7 BIF：断路中断标志 (Break interrupt flag) 只要断路输入变为有效状态，此标志便由硬件置 1。断路输入无效后可通过软件对其清零。

0：未发生断路事件。

1：在断路输入上检测到有效电平。

位 6 TIF：触发中断标志 (Trigger interrupt flag)

在除门控模式以外的所有模式下，当使能从模式控制器后在 TRGI 输入上检测到有效边沿时，该标志将由硬件置 1。选择门控模式时，该标志将在计数器启动或停止时置 1。但需要通过软件清零。

0：未发生触发事件。

1：触发中断挂起。

位 5 COMIF：COM 中断标志 (COM interrupt flag)

此标志在发生 COM 事件时（捕获/比较控制位 CCxE、CCxNE 和 OCxM 已更新时）由硬件 置 1。但需要通过软件清零。

0：未发生 COM 事件。

1：COM 中断挂起。

位 4 CC4IF：捕获/比较 4 中断标志 (Capture/Compare 4 interrupt flag)

请参见 CC1IF 说明

位 3 CC3IF：捕获/比较 3 中断标志 (Capture/Compare 3 interrupt flag)

请参见 CC1IF 说明

位 2 CC2IF：捕获/比较 2 中断标志 (Capture/Compare 2 interrupt flag)

请参见 CC1IF 说明

位 1 CC1IF：捕获/比较 1 中断标志 (Capture/Compare 1 interrupt flag)

如果通道 CC1 配置为输出：

当计数器与比较值匹配时，此标志由硬件置 1，中心对齐模式下除外（请参见 PWMx_CR1 寄存器中的 CMS 位说明）。但需要通过软件清零。

0：不匹配。

1：PWMx_CNT 计数器的值与 PWMx_CCR1 寄存器的值匹配。当 PWMx_CCR1 的值大于

PWMx_ARR 的值时，CC1IF 位将在计数器发生上溢（递增计数模式和增减计数模式下）或下溢（递减计数模式下）时变为高电平。

如果通道 CC1 配置为输入：

此位将在发生捕获事件时由硬件置 1。通过软件或读取 PWMx_CCR1 寄存器将该位清零。

0：未发生输入捕获事件

1：PWMx_CCR1 寄存器中已捕获到计数器值（IC1 上已检测到与所选极性匹配的边沿）

位 0 UIF：更新中断标志 (Update interrupt flag)

该位在发生更新事件时通过硬件置 1。但需要通过软件清零。

0：未发生更新。

1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：

—PWMx_CR1 寄存器中的 UDIS=0，并且重复计数器值上溢或下溢时（重复计数器 = 0 时 更新）。

—PWMx_CR1 寄存器中的 URS = 0 且 UDIS = 0，并且由软件使用 PWMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。

—PWMx_CR1 寄存器中的 URS=0 且 UDIS=0，并且 CNT 由触发事件重新初始化时（请参见： PWM1 和 PWM8 从模式控制寄存器 (PWMx_SMCR)）。

15.4.6 PWM1 和 PWM8 事件生成寄存器 (PWMx_EGR)

PWM1&PWM8 event generation register

偏移地址：0x14

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
								w	w	w	w	w	w	w	w

位 15:8 保留，必须保持复位值。

位 7 BG：断路生成 (Break generation)

此位由软件置 1 以生成事件，并由硬件自动清零。

0：不执行任何操作

1：生成断路事件。MOE 位清零且 BIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。

位 6 TG：生成触发信号 (Trigger generation)

此位由软件置 1 以生成事件，并由硬件自动清零。

0：不执行任何操作

1：PWMx_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。

位 5 COMG：捕获/比较控制更新生成 (Capture/Compare control update generation)

该位可通过软件置 1，并由硬件自动清零

0：不执行任何操作

1：CCPC 位置 1 时，可更新 CCxE、CCxNE 和 OCxM 位

注意：此位仅对具有互补输出的通道有效。

位 4 CC4G：捕获/比较 4 生成 (Capture/Compare 4 generation)

请参见 CC1G 说明

位 3 CC3G：捕获/比较 3 生成 (Capture/Compare 3 generation)

请参见 CC1G 说明

位 2 CC2G：捕获/比较 2 生成 (Capture/Compare 2 generation)

请参见 CC1G 说明

位 1 CC1G：捕获/比较 1 生成 (Capture/Compare 1 generation)

此位由软件置 1 以生成事件，并由硬件自动清零。

0：不执行任何操作

1：通道 1 上生成捕获/比较事件：

如果通道 CC1 配置为输出：

使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。

如果通道 CC1 配置为输入：

PWMx_CCR1 寄存器中将捕获到计数器当前值。使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。

位 0 UG：更新生成 (Update generation)

该位可通过软件置 1，并由硬件自动清零。

0： 不执行任何操作

1： 重新初始化计数器并生成一个寄存器更新事件。请注意，预分频器计数器也将清零（但 预分频比不受影响）。如果选择中心对齐模式或 DIR=0（递增计数），计数器将清零；如果 DIR=1（递减计数），计数器将使用自动重载值 (PWMx_ARR)。

15.4.7 PWM1 和 PWM8 捕获/比较模式寄存器 1 (PWMx_CCMR1)

PWM1&PWM8 capture/compare mode register 1

偏移地址：0x18

复位值：0x0000

这些通道可用于输入（捕获模式）或输出（比较模式）模式。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其它位在输入模式和输出模式下的功能均不同。对于任 一给定位，OCxx 用于说明通道配置为输出时该位对应的功能，ICxx 则用于说明通道配置为 输入时该位对应的功能。因此，必须注意同一个位在输入阶段和输出阶段具有不同的含义。

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			OC1PE	OC1FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式：

位 15 OC2CE：输出比较 2 清零使能 (Output Compare 2 clear enable)

位 14:12 OC2M[2:0]：输出比较 2 模式 (Output Compare 2 mode)

位 11 OC2PE：输出比较 2 预装载使能 (Output Compare 2 preload enable)

位 10 OC2FE：输出比较 2 快速使能 (Output Compare 2 fast enable)

位 9:8 CC2S[1:0]：捕获/比较 2 选择 (Capture/Compare 2 selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC2 通道配置为输出

01：CC2 通道配置为输入，IC2 映射到 TI2 上

10：CC2 通道配置为输入，IC2 映射到 TI1 上

11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（PWMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（ PWMx_CCER 中的 CC2E =“ 0 ”），才可向 CC2S 位写入数据。

位 7 OC1CE：输出比较 1 清零使能 (Output Compare 1 clear enable)

OC1CE：输出比较 1 清零使能 (Output Compare 1 Clear Enable)

0：OC1Ref 不受 ETRF 输入影响

1：ETRF 输入上检测到高电平时，OC1Ref 立即清零。

位 6:4 OC1M：输出比较 1 模式 (Output Compare 1 mode)

这些位定义提供 OC1 和 OC1N 的输出参考信号 OC1REF 的行为。OC1REF 000：冻结—输出比较寄存器 PWMx_CCR1 与计数器 PWMx_CNT 进行比较不会对输出造成任何影响。（该模式用于生成时基）。

001：将通道 1 设置为匹配时输出有效电平。当计数器 PWMx_CNT 与捕获/比较寄存器 1 (PWMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。

010：将通道 1 设置为匹配时输出无效电平。当计数器 PWMx_CNT 与捕获/比较寄存器 1 (PWMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。

011：翻转—PWMx_CNT=PWMx_CCR1 时，OC1REF 发生翻转。

100：强制变为无效电平—OC1REF 强制变为低电平。

101：强制变为有效电平—OC1REF 强制变为高电平。

110：PWM 模式 1—在递增计数模式下，只要 PWMx_CNT<PWMx_CCR1，通道 1 便为有效状态，否则为无效状态。在递减计数模式下，只要 PWMx_CNT>PWMx_CCR1，通道 1 便为 无效状态（OC1REF=“0”），否则为有效状态（OC1REF=“1”）。

111：PWM 模式 2—在递增计数模式下，只要 PWMx_CNT<PWMx_CCR1，通道 1 便为无效状态，否则为有效状态。在递减计数模式下，只要 PWMx_CNT>PWMx_CCR1，通道 1 便为 有效状态，否则为无效状态。为高电平有效，而 OC1 和 OC1N 的有效电平则取决于 CC1P 位和 CC1NP 位。

注意： 1 ： 只要编程了 LOCK （ PWMx_BDTR 寄存器中的 LOCK 位）级别 3 且 CC1S=“ 00 ”（通道配置为输出），这些位即无法修改。 2 ： 在 PWM 模式 1 或 PWM 模式 2 下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“ PWM ”模式时， OCREF 电平才会发生更改。 3 ： 此位域将在具有互补输出的通道上进行预装载。如果 PWMx_CR2 寄存器中 的 CCPC 位置 1 ，则仅当生成 COM 事件时， OC1M 有效位才会从预装载位获取新值

位 3 OC1PE：输出比较 1 预装载使能 (Output Compare 1 preload enable)

0: 禁止与 PWMx_CCR1 相关的预装载寄存器。可随时向 PWMx_CCR1 写入数据，写入后将立即使用新值。

1: 使能与 PWMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。PWMx_CCR 预装载值在每次生成更新事件时都会装载到活动寄存器中。

注意： 1 ： 只要编程了 LOCK （ PWMx_BDTR 寄存器中的 LOCK 位）级别 3 且 CC1S=“ 00 ”（通道配置为输出），这些位即无法修改。 2 ： 只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式（ PWMx_CR1 寄存器中的 OPM 位置 1 ）。其它情况下则无法保证该行为。

位 2 OC1FE：输出比较 1 快速使能 (Output Compare 1 fast enable)

此位用于加快触发输入事件对 CC 输出的影响。

0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活

CC1 输出的短延迟时间为 5 个时钟周期。

1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何。OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OCFE 才会起作用。

位 1:0 CC1S：捕获/比较 1 选择 (Capture/Compare 1 selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出

01：CC1 通道配置为输入，IC1 映射到 TI1 上

10：CC1 通道配置为输入，IC1 映射到 TI2 上

11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（PWMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（ PWMx_CCER 中的 CC1E = “ 0 ”），才可向 CC1S 位写入数据。

输入捕获模式

位 15:12 IC2F：输入捕获 2 滤波器 (Input capture 2 filter)

位 11:10 IC2PSC[1:0]：输入捕获 2 预分频器 (Input capture 2 prescaler)

位 9:8 CC2S：捕获/比较 2 选择 (Capture/Compare 2 selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC2 通道配置为输出

01：CC2 通道配置为输入，IC2 映射到 TI2 上

10：CC2 通道配置为输入，IC2 映射到 TI1 上

11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（PWMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（ PWMx_CCER 中的 CC2E = “ 0 ”），才可向 CC2S 位写入数据。

位 7:4 IC1F[3:0]：输入捕获 1 滤波器 (Input capture 1 filter)

此位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个事件才视为一个有效边沿：

0000：无滤波器，按 fDTS 频率进行采样

0001：fSAMPLING=fCK_INT，N=2

0010：fSAMPLING=fCK_INT，N=4

0011：fSAMPLING=fCK_INT，N=8

0100：fSAMPLING=fDTS/2，N=6

0101：fSAMPLING=fDTS/2，N=8

0110：fSAMPLING=fDTS/4，N=6

0111：fSAMPLING=fDTS/4，N=8

1000：fSAMPLING=fDTS/8，N=6

1001：fSAMPLING=fDTS/8，N=8

1010：fSAMPLING=fDTS/16，N=5

1011：fSAMPLING=fDTS/16，N=6

1100：fSAMPLING=fDTS/16，N=8

1101：fSAMPLING=fDTS/32，N=5

1110：fSAMPLING=fDTS/32，N=6

1111：fSAMPLING=fDTS/32，N=8

位 3:2 IC1PSC：输入捕获 1 预分频器 (Input capture 1 prescaler)

此位域定义 CC1 输入 (IC1) 的预分频比。

只要 CC1E=“0”（PWMx_CCER 寄存器），预分频器便立即复位。

00：无预分频器，捕获输入上每检测到一个边沿便执行捕获

01：每发生 2 个事件便执行一次捕获

10：每发生 4 个事件便执行一次捕获

11：每发生 8 个事件便执行一次捕获

位 1:0 CC1S：捕获/比较 1 选择 (Capture/Compare 1 Selection)

此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC1 通道配置为输出
01：CC1 通道配置为输入，IC1 映射到 TI1 上
10：CC1 通道配置为输入，IC1 映射到 TI2 上
11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（PWMx_SMCR 寄存器）选 择内部触发输入时有效
注意：仅当通道关闭时（ PWMx_CCER 中的 CC1E =“ 0 ”），才可向 CC1S 位写入数据。

15.4.8 PWM1 和 PWM8 捕获/比较模式寄存器 2 (PWMx_CCMR2)

PWM1&PWM8 capture/compare mode register 2
偏移地址：0x1C
复位值：0x0000
请参见上述 CCMR1 寄存器说明。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M			OC3PE	OC3FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]				IC3PSC[1:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式
位 15 OC4CE：输出比较 4 清零使能 (Output compare 4 clear enable)
位 14:12 OC4M：输出比较 4 模式 (Output compare 4 mode)
位 11 OC4PE：输出比较 4 预装载使能 (Output compare 4 preload enable)
位 10 OC4FE：输出比较 4 快速使能 (Output compare 4 fast enable)
位 9:8 CC4S：捕获/比较 4 选择 (Capture/Compare 4 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC4 通道配置为输出
01：CC4 通道配置为输入，IC4 映射到 TI4 上
10：CC4 通道配置为输入，IC4 映射到 TI3 上
11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（PWMx_SMCR 寄存 器）选择内部触发输入时有效
注意：仅当通道关闭时（ PWMx_CCER 中的 CC4E =“ 0 ”），才可向 CC4S 位写入数据。
位 7 OC3CE：输出比较 3 清零使能 (Output compare 3 clear enable)
位 6:4 OC3M：输出比较 3 模式 (Output compare 3 mode)
位 3 OC3PE：输出比较 3 预装载使能 (Output compare 3 preload enable)
位 2 OC3FE：输出比较 3 快速使能 (Output compare 3 fast enable)
位 1:0 CC3S：捕获/比较 3 选择 (Capture/Compare 3 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC3 通道配置为输出
01：CC3 通道配置为输入，IC3 映射到 TI3 上
10：CC3 通道配置为输入，IC3 映射到 TI4 上
11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（PWMx_SMCR 寄存 器）选择内部触发输入时有效
注意：仅当通道关闭时（ PWMx_CCER 中的 CC3E =“ 0 ”），才可向 CC3S 位写入数据。
输入捕获模式
位 15:12 IC4F：输入捕获 4 滤波器 (Input capture 4 filter)
位 11:10 IC4PSC：输入捕获 4 预分频器 (Input capture 4 prescaler)
位 9:8 CC4S：捕获/比较 4 选择 (Capture/Compare 4 selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC4 通道配置为输出

01：CC4 通道配置为输入，IC4 映射到 TI4 上

10：CC4 通道配置为输入，IC4 映射到 TI3 上

11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（PWMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（ PWMx_CCER 中的 CC4E = " 0 "），才可向 CC4S 位写入数据。

位 7:4 IC3F：输入捕获 3 滤波器 (Input capture 3 filter)

位 3:2 IC3PSC：输入捕获 3 预分频器 (Input capture 3 prescaler)

位 1:0 CC3S：捕获/比较 3 选择 (Capture/compare 3 selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC3 通道配置为输出

01：CC3 通道配置为输入，IC3 映射到 TI3 上

10：CC3 通道配置为输入，IC3 映射到 TI4 上

11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（PWMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（ PWMx_CCER 中的 CC3E = " 0 "），才可向 CC3S 写入数据。

15.4.9 PWM1 和 PWM8 捕获/比较使能寄存器 (PWMx_CCER)

PWM1&PWM8 capture/compare enable register

偏移地址：0x20

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		CC4P	CC4E	CC2NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:14 保留，必须保持复位值。

位 13 CC4P：捕获/比较 4 输出极性 (Capture/Compare 4 output polarity)

请参见 CC1P 说明

位 12 CC4E：捕获/比较 4 输出使能 (Capture/Compare 4 output enable)

请参见 CC1E 说明 位

11 CC3NP：捕获/比较 3 互补输出极性 (Capture/Compare 3 complementary output polarity)

请参见 CC1NP 说明 位

10 CC3NE：捕获/比较 3 互补输出使能 (Capture/Compare 3 complementary output enable)

请参见 CC1NE 说明

位 9 CC3P：捕获/比较 3 输出极性 (Capture/Compare 3 output polarity) 请参见 CC1P 说明

位 8 CC3E：捕获/比较 3 输出使能 (Capture/Compare 3 output enable) 请参见 CC1E 说明

位 7 CC2NP：捕获/比较 2 互补输出极性 (Capture/Compare 2 complementary output polarity) 请参见 CC1NP 说明

位 6 CC2NE：捕获/比较 2 互补输出使能 (Capture/Compare 2 complementary output enable) 请参见 CC1NE 说明

位 5 CC2P：捕获/比较 2 输出极性 (Capture/Compare 2 output polarity) 请参见 CC1P 说明

位 4 CC2E：捕获/比较 2 输出使能 (Capture/Compare 2 output enable) 请参见 CC1E 说明

位 3 CC1NP：捕获/比较 1 互补输出极性 (Capture/Compare 1 complementary output polarity)

CC1 通道配置为输出：

0：OC1N 高电平有效。

1：OC1N 低电平有效。

CC1 通道配置为输入：

此位与 CC1P 配合使用，用以定义 TI1FP1 和 TI2FP1 的极性。请参见 CC1 说明。

注意：此位将在具有互补输出的通道上进行预装载。如果 PWMx_CR2 寄存器中的 CCPC 位置 1 ，则仅当生成换向事件时， CC1NP 有效位才会从预装载位获取新值。 注意：只要编程了 LOCK （ PWMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3 且 CC1S=“ 00 ”（通道配置为输出），此位立即变为不可写状态

位 2 CC1NE：捕获 / 比较 1 互补输出使能 (Capture/Compare 1 complementary output enable)

0： 关闭 —OC1N 未激活。OC1N 电平是 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的函数。

1： 开启 — 在相应输出引脚上输出 OC1N 信号，具体取决于 MOE、OSSI、OSSR、OIS1。OIS1N 和 CC1E 位。

注意：此位将在具有互补输出的通道上进行预装载。如果 PWMx_CR2 寄存器的 CCPC 位 置 1 ，则仅当生成换向事件时， CC1NE 有效位才会从预装载位获取新值。

位 1 CC1P：捕获/比较 1 输出极性 (Capture/Compare 1 output polarity)

CC1 通道配置为输出：

0：OC1 高电平有效

1：OC1 低电平有效

CC1 通道配置为输入：

CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的有效极性。

00：非反相/上升沿触发 电路对 TIxFP1 上升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式或编码器模式下执行触发操作）。

01：反相/下降沿触发 电路对 TIxFP1 下升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 反相（在门控模式或编码器模式下执行触发操作）。

10：保留，不使用此配置。

11：未反相/边沿触发。

电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。编码器模式下不得使用此 配置。

注意：此位将在具有互补输出的通道上进行预装载。如果 PWMx_CR2 寄存器中的 CCPC 位 置 1 ，则仅当生成换向事件时， CC1P 有效位才会从预装载位获取新值。 注意：只要编程了 LOCK （ PWMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3 ， 此 位 立 即 变 为 不 可写状态。

位 0 CC1E：捕获 / 比较 1 输出使能 (Capture/Compare 1 output enable)

CC1 通道配置为输出：

0：关闭 —OC1 未激活。OC1 电平是 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的函数。

1：开启 —OC1 信号是相应输出引脚上的输出，具体取决于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位。

CC1 通道配置为输入：

此位决定了是否可以实际将计数器值捕获到输入捕获 / 比较寄存器 1 (PWMx_CCR1) 中。

0：禁止捕获。

1：使能捕获。

注意：此位将在具有互补输出的通道上进行预装载。如果 PWMx_CR2 寄存器中的 CCPC 位置 1 ，则仅当生成换向事件时， CC1E 有效位才会从预装载位获取新值。

表 73. 具有断路功能的互补通道 OCx 和 OCxN

控制位					输出状态(1)	
MOE 位	OSSI 位	OSSR 位	CCxE 位	CCxNE 位	OCx 输出状态	OCxN 输出状态
1	X	0	0	0	禁止输出（不由定时器驱动） OCx=0、OCx_EN=0	禁止输出（不由定时器驱动） OCxN=0、OCxN_EN=0
		0	0	1	禁止输出（不由定时器驱动） OCx=0、OCx_EN=0	OCxREF + 极性 OCxN=OCxREF 异或 CCxNP、OCxN_EN=1
		0	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP、OCx_EN=1	禁止输出（不由定时器驱动） OCxN=0、OCxN_EN=0
		0	1	1	OCREF + 极性 + 死区 OCx_EN=1	OCREF 互补项（而非 OCREF）+ 极性 + 死区 OCxN_EN=1
		1	0	0	禁止输出（不由定时器驱动） OCx=CCxP、OCx_EN=0	禁止输出（不由定时器驱动） OCxN=CCxNP、OCxN_EN=0
		1	0	1	关闭状态（输出使能为无效状态） OCx=CCxP、OCx_EN=1	OCxREF + 极性 OCxN=OCxREF 异或 CCxNP、OCxN_EN=1
		1	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP、OCx_EN=1	关闭状态（输出使能为无效状态） OCxN=CCxNP、OCxN_EN=1
		1	1	1	OCREF + 极性 + 死区 OCx_EN=1	OCREF 互补项（而非 OCREF）+ 极性 + 死区 OCxN_EN=1
0	0	X	0	0	禁止输出（不由定时器驱动） OCx=CCxP、OCx_EN=0	禁止输出（不由定时器驱动） OCxN=CCxNP、OCxN_EN=0
	0		0	1	禁止输出 （不由定时器驱动） 异步：OCx=CCxP、 OCx_EN=0、 OCxN=CCxNP、 OCxN_EN=0 如果存在时钟：在死区后 OCx=OISx 且 OCxN=OISxN，从而假定 OISx 和 OISxN 在有效状态下与 OCX 和 OCxN 不对应。	
	0		1	0		
	0		1	1		
	1		0	0	禁止输出（不由定时器驱动） OCx=CCxP、OCx_EN=0	禁止输出（不由定时器驱动） OCxN=CCxNP、OCxN_EN=0
	1		0	1	关闭状态 （输出使能为无效状态） 异步：OCx=CCxP、 OCx_EN=1、 OCxN=CCxNP、 OCxN_EN=1 如果存在时钟：在死区后 OCx=OISx 且 OCxN=OISxN，从而假定 OISx 和 OISxN 在有效状态下与 OCX 和 OCxN 不对应	
	1		1	0		
	1		1	1		

1. 如果一个通道的两个输出均未使用 (CCxE = CCxNE = 0)，则 OISx、OISxN、CCxP 和 CCxNP 位必须保持清零状态。

注 意 ： 与 互 补 通 道 OCx 和 OCxN 相连的外部 I/O 引脚的状态取决于通道 OCx 和 OCxN 的状态以及 GPIO 寄存器。

15.4.10 PWM1 和 PWM8 计数器 (PWMx_CNT)

PWM1&PWM8 counter

偏移地址：0x24

复位值：0x000

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 15:0 CNT[15:0]：计数器值 (Counter value)

15.4.11 PWM1 和 PWM8 预分频器 (PWMx_PSC)

PWM1&PWM8 prescaler

偏移地址：0x28

复位值：0x0000

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

位 15:0 PSC[15:0]：预分频器值 (Prescaler value)

计数器时钟频率 (CK_CNT) 等于 fCK_PSC / (PSC[15:0] + 1)。 PSC 包含每次发生更新事件（包括计数器通过 PWMx_EGR 寄存器中的 UG 位清零时，或在 配置为“复位模式”时通过触发控制器清零时）时要装载到活动预分频器寄存器的值。

15.4.12 PWM1 和 PWM8 自动重载寄存器 (PWMx_ARR)

PWM1&PWM8 auto-reload register

偏移地址：0x2C

复位值：0x0000

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 15:0 ARR[15:0]：自动重载值 (Auto-reload value)

ARR 为要装载到实际自动重载寄存器的值。 有关 ARR 更新和行为的详细信息，请参见：时基单元。当自动重载值为空时，计数器不工作。

15.4.13 PWM1 和 PWM8 重复计数器寄存器 (PWMx_RCR)

PWM1&PWM8 repetition counter register

偏移地址：0x30

复位值：0x0000

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

位 15:8 保留，必须保持复位值。

位 7:0 REP[7:0]：重复计数器值 (Repetition counter value)

使能预装载寄存器时，用户可通过这些位设置比较寄存器的更新频率（即，从预装载寄存器 向活动寄存器周期性传输数据）；使能更新中断时，也可设置更新中断的生成速率。 与 REP_CNT 相关的减计数器每次计数到 0 时，都将生成一个更新事件并且计数器从 REP 值重新开始计数。由于只有生成重复更新事件 U_RC 时，REP_CNT 才会重载 REP 值，因此在生成下一重复更新事件之前，无论向 PWMx_RCR 寄存器写入何值都无影响。这意味着 PWM 模式下 (REP+1) 相当于：

— 边沿对齐模式下的 PWM 周期数

— 中心对齐模式下的 PWM 半周期数

15.4.14 PWM1 和 PWM8 捕获/比较寄存器 1 (PWMx_CCR1)

PWM1&PWM8 capture/compare register 1

STC-MCU

偏移地址：0x34
复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 CCR1[15:0]：捕获/比较 1 值 (Capture/Compare 1 value)
如果通道 CC1 配置为输出：
CCR1 是捕获/比较寄存器 1 的预装载值。 如果没有通过 PWMx_CCMR 寄存器中的 OC1PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器 1）。 实际捕获/比较寄存器中包含要与计数器 PWMx_CNT 进行比较并在 OC1 输出上发出信号的值。
如果通道 CC1 配置为输入：
CCR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。

15.4.15 PWM1 和 PWM8 捕获/比较寄存器 2 (PWMx_CCR2)

PWM1&PWM8 capture/compare register 2
偏移地址：0x38
复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 CCR2[15:0]：捕获/比较 2 值 (Capture/Compare 2 value)
如果通道 CC2 配置为输出：
CCR2 为要装载到实际捕获/比较 2 寄存器的值（预装载值）。 如果没有通过 PWMx_CCMR2 寄存器中的 OC2PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 2 寄存器中。实际捕获/比较寄存器中包含要与计数器 PWMx_CNT 进行比较并在 OC2 输出上发出信号的值。
如果通道 CC2 配置为输入：
CCR2 为上一个输入捕获 2 事件 (IC2) 发生时的计数器值。

15.4.16 PWM1 和 PWM8 捕获/比较寄存器 3 (PWMx_CCR3)

PWM1&PWM8 capture/compare register 3
偏移地址：0x3C
复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 CCR3[15:0]：捕获/比较值 (Capture/Compare value)
如果通道 CC3 配置为输出：
CCR3 为要装载到实际捕获/比较 3 寄存器的值（预装载值）。 如果没有通过 PWMx_CCMR3 寄存器中的 OC3PE 位来使能预装载功能，写入的数值会被直接传 输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 3 寄存器中。 实际捕获/比较寄存器中包含要与计数器 PWMx_CNT 进行比较并在 OC3 输出上发出信号的值。

如果通道 CC3 配置为输入：
CCR3 为上一个输入捕获 3 事件 (IC3) 发生时的计数器值。

15.4.17 PWM1 和 PWM8 捕获/比较寄存器 4 (PWMx_CCR4)

PWM1&PWM8 capture/compare register 4
偏移地址：0x40
复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 CCR4[15:0]：捕获/比较值 (Capture/Compare value)
如果通道 CC4 配置为输出：
CCR4 为要装载到实际捕获/比较 4 寄存器的值（预装载值）。 如果没有通过 PWMx_CCMR4 寄存器中的 OC4PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 4 寄存器中。 实际捕获/比较寄存器中包含要与计数器 PWMx_CNT 进行比较并在 OC4 输出上发出信号的值。
如果通道 CC4 配置为输入：
CCR4 为上一个输入捕获 4 事件 (IC4) 发生时的计数器值。

15.4.18 PWM1 和 PWM8 断路和死区寄存器 (PWMx_BDTR)

PWM1&PWM8 break and dead-PWMe register
偏移地址：0x44
复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

注 意 ： 由 于 可 以 根 据 LOCK 配置锁定位 AOE 、 BKP 、 BKE 、 OSSI 、 OSSR 和 DTG[7:0] 的写操 作，因此必须在第一次对 PWMx_BDTR 寄存器执行写访问时对这些位进行配置。

位 15 MOE：主输出使能 (Main output enable)
只要断路输入变为有效状态，此位便由硬件异步清零。此位由软件置 1，也可根据 AOE 位状 态自动置 1。此位仅对配置为输出的通道有效。
0：OC 和 OCN 输出禁止或被强制为空闲状态。
1：如果 OC 和 OCN 输出的相应使能位（PWMx_CCER 寄存器中的 CCxE 和 CCxNE 位）均 置 1，则使能 OC 和 OCN 输出
有关详细信息，请参见 OC/OCN 使能说明（第 16.4.9 节： PWM1 和 PWM8 捕获 / 比较使能寄存器 (PWMx_CCER)）。

位 14 AOE：自动输出使能 (Automatic output enable)
0：MOE 只能由软件置 1
1：MOE 可由软件置 1，也可在发生下一更新事件时自动置 1(如果断路输入无效）
注意：只要编程了 LOCK （ PWMx_BDTR 寄存器中的 LOCK 位）级别 1 ，此位即无法修改

位 13 BKP：断路极性 (Break polarity)
0：断路输入 BRK 为低电平有效
1：断路输入 BRK 为高电平有效
注意：只要编程了 LOCK （ PWMx_BDTR 寄存器中的 LOCK 位）级别 1 ，此位即无法

修改。

注意：对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延才生效

位 12 BKE：断路使能 (Break enable)

0：禁止断路输入（BRK 和 CSS 时钟故障事件）

1：使能断路输入（BRK 和 CSS 时钟故障事件）

注意：编程了 LOCK （ PWMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。

注意：对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 11 OSSR：运行模式下的关闭状态选择 (Off-state selection for Run mode)

此位在 MOE=1 时作用于配置为输出模式且具有互补输出的通道。如果定时器中没有互补输出，则不存在 OSSR。

有关详细信息，请参见 OC/OCN 使能说明（第 16.4.9 节： PWM1 和 PWM8 捕获 /比 较使能寄存器 (PWMx_CCER)）。

0：处于无效状态时，禁止 OC/OCN 输出（OC/OCN 使能输出信号=0）。 1：处于无效状态时，一旦 CCxE=1 或 CCxNE=1，便使能 OC/OCN 输出并将其设为无效电平。然后设置 OC/OCN 使能输出信号=1

注意：编程了 LOCK （ PWMx_BDTR 寄存器中的 LOCK 位）级别 2 后，此位即无法修改

位 10 OSSI：空闲模式下的关闭状态选择 (Off-state selection for Idle mode)

此位在 MOE=0 时作用于配置为输出的通道。

有关详细信息，请参见 OC/OCN 使能说明（第 16.4.9 节： PWM1 和 PWM8 捕获 /比较使能寄存器 (PWMx_CCER)）。

0：处于无效状态时，禁止 OC/OCN 输出（OC/OCN 使能输出信号=0）。 1：处于无效状态时，一旦 CCxE=1 或 CCxNE=1，便将 OC/OCN 输出首先强制为其空闲电平。然后设置 OC/OCN 使能输出信号=1

注意：编程了 LOCK （ PWMx_BDTR 寄存器中的 LOCK 位）级别 2 后，此位即无法修改

位 9:8 LOCK[1:0]：锁定配置 (Lock configuration)

这些位用于针对软件错误提供写保护。

00：关闭锁定—不对任何位提供写保护。

01：锁定级别 1，此时无法对 PWMx_BDTR 寄存器中的 DTG 位、PWMx_CR2 寄存器中的 OISx 和 OISxN 位以及 PWMx_BDTR 寄存器中的 BKE/BKP/AOE 位执行写操作。

10：锁定级别 2，此时无法对锁定级别 1 中适用的各位、CC 极性位（PWMx_CCER 寄存器中的 CCxP/CCxNP 位，只要通过 CCxS 位将相关通道配置为输出）以及 OSSR 和 OSSI 位执行写操作。

11：锁定级别 3，此时无法对锁定级别 2 中适用的各位、CC 控制位（PWMx_CCMRx 寄存器中的 OCxM 和 OCxPE 位，只要通过 CCxS 位将相关通道配置为输出）执行写操作。

注意：复位后只能对 LOCK 位执行一次写操作。对 PWMx_BDTR 寄存器执行写操作后其中的内 容将冻结，直到下一次复位。

位 7:0 DTG[7:0]：配置死区发生器 (Dead-PWMe generator setup)

此位域定义插入到互补输出之间的死区持续时间。DT 与该持续时间相对应。

DTG[7:5]=0xx => DT=DTG[7:0]x tdtg，其中 tdtg=tDTS。

DTG[7:5]=10x => DT=(64+DTG[5:0])xtdtg，其中 Tdtg=2xtDTS。 DTG[7:5]=110 => DT=(32+DTG[4:0])xtdtg，其中 Tdtg=8xtDTS。DTG[7:5]=111 => DT=(32+DTG[4:0])xtdtg，其中 Tdtg=16xtDTS。

示例：如果 TDTS=125ns (8MHz)，则可能的死区值为：

0 到 15875 ns（步长为 125 ns），

16 us 到 31750 ns（步长为 250 ns），

32 us 到 63us（步长为 1 us），

64 us 到 126 us（步长为 2 us）

注意：只要编程了 LOCK （ PWMx_BDTR 寄存器中的 LOCK 位）级别 1 、 2 或 3 ，此位域即无法 修改。

15.4.19 PWM1 和 PWM8 DMA 控制寄存器 (PWMx_DCR)

PWM1&PWM8 DMA control register

偏移地址：0x48

复位值：0x0000

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

位 15:13 保留，必须保持复位值。

位 12:8 DBL[4:0]：DMA 连续传送长度 (DMA burst length)

该 5 位定义了 DMA 在连续模式下的传送长度（当对 PWMx_DMAR 寄存器进行读或写时，定时器则进行一次连续传送）。

PWMx_DMAR 地址）

00000：1 次传输

00001：2 次传输

00010：3 次传输

...

10001：18 次传输

位 7:5 保留，必须保持复位值。

位 4:0 DBA[4:0]：DMA 基址 (DMA base address)

DBA[4:0]：DMA 基址 (DMA base address) 该 5 位向量定义 DMA 传输的基址（通过 PWMx_DMAR 地址进行读/写访问时）。DBA 定义为从 PWMx_CR1 寄存器地址开始计算的偏移量。

示例：

00000：PWMx_CR1，

00001：PWMx_CR2，

00010：PWMx_SMCR，

...

示例：考虑以下传输：DBL = 7 次传输，DBA = PWMx_CR1。这种情况下将向/从自 PWMx_CR1 地址开始的 7 个寄存器传输数据。

15.4.20 PWM1 和 PWM8 全传输 DMA 地址 (PWMx_DMAR)

PWM1&PWM8 DMA address for full transfer

偏移地址：0x4C

复位值：0x0000

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

位 15:0 DMAB[15:0]：DMA 连续传送寄存器 (DMA register for burst accesses)

对 DMAR 寄存器执行读或写操作将访问位于如下地址的寄存器

(PWMx_CR1 地址) + (DBA + DMA 索引) x 4

其中 PWMx_CR1 地址为控制寄存器 1 的地址，DBA 为 PWMx_DCR 寄存器中配置的 DMA 基址，DMA 索引由 DMA 传输自动控制，其范围介于 0 到 DBL（PWMx_DCR 寄

存器中配置的 DBL）之间。

DMA 连续传送功能使用方法示例

DMA 连续传送功能使用方法示例 本例中，定时器 DMA 连续传送功能用于将 CCRx 寄存器（x = 2、3、4）的内容更新为通过 DMA 传输到 CCRx 寄存器中的多个半字。

具体操作步骤如下：

1. 将相应的 DMA 通道配置如下：
   — DMA 通道外设地址为 DMAR 寄存器地址
   —DMA 通道存储器地址为包含要通过 DMA 传输到 CCRx 寄存器的数据的 R AM 缓冲 区地址。
   — 要传输的数据量 = 3（参见下文注释）。
   — 禁止循环模式。
2. 通过将 DBA 和 DBL 位域配置如下来配置 DCR 寄存器： DBL = 3 次传输，DBA = 0xE。
3. 使能 PWMx 更新 DMA 请求（DIER 寄存器中的 UDE 位置 1）。 4. 使能 PWMx 5. 使能 DMA 通道

注 意 ： 本 例 适 用 于 每 个 CCRx 寄存器只更新一次的情况。如果每个 CCRx 寄存器要更新两次，则要 传输的数据量应为 6 。下面以包含 data1 、 data2 、 data3 、 data4 、 data5 和 data6 的 RAM 缓冲区为例。数据将按照如下方式传输到 CCRx 寄存器：在第一个更新 DMA 请求期间， data1 传输到 CCR2 ， data2 传输到 CCR3 ， data3 传输到 CCR4 ；在第二个更新 DMA 请求 期间，data4 传输到 CCR2 ， data5 传输到 CCR3 ， data6 传输到 CCR4

15.4.21 PWM1 和 PWM8 寄存器映射

PWM1 和 PWM8 寄存器可映射为 16 位可寻址寄存器，如下表所述：

表 74. PWM1 和 PWM8 寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0X00	TIMx_CR1	Reserved																						CKD [1:0]		ARPE	CMS [1:0]		DIR	OPM	URS	UDIS	CEN
	Reset value																							0	0	0	0	0	0	0	0	0	0
0X04	TIMx_CR2	Reserved																	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Reserved	CCPC
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0		0
0X08	TIMx_SMCR	Reserved															ETP	ECE	ETPS [3:0]			ETF[3:0]				MSM	TS[2:0]			Reserved0	SMS[2:0]		
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0X0C	TIMx_DIER																		TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0X10	TIMx_SR																				CC4OF	CC3OF	CC1OF	CC1OF	Reserved	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
	Reset value																					0	0	0			0	0	0	0	0	0	0

0X14	TIMx_EGR	Reserved								BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG		
	Reset value									0	0	0	0	0	0	0	0		
0X18	TIMx_CCMR1 Output compare mode	Reserved		OC2CE	OC2M [2:-0]			OC2PE	OC2FE	CC2S [1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIMx_CCMR1 INput compare mode	Reserved		IC2F[3:0]				IC2 PSC [1:0]		CC2S [1:0		CC2S[1:0]				IC1 PSC [1:0]		CC1S [1:0]	
	Reset value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0X1C	TIMx_CCMR2 Output compare mode	Reserved		OC4CE	OC4M [2:-0]			OC4PE	OC4FE	CC4S [1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S [1:0]	
	Reset value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIMx_CCMR2 INtput compare mode	Reserved		IC4F[3:0]				IC4 PSC [1:0]		CC4S [1:0		CC3F[1:0]				IC3PSC [1:0]		CC3S [1:0]	
	Reset value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0X20	TIMx_CCER	Reserved				CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0	0	0
0X24	TIMx_CNT	Reserved	CNT[15:0]																
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0X28	TIMx_PSC	Reserved	PSC[15:0]																
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0X2C	TIMx_ARR	Reserved	ARR15:0]																
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	TIMx_RCR	Reserved										REP[7:0]							
	Reset value											0	0	0	0	0	0	0	0
0x34	TIMx_CCR1	Reserved	CCR1[15:0]																
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	TIMx_CCR2	Reserved	CCR2[15:0]																
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	TIMx_CCR3	Reserved	CCR3[15:0]																
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	TIMx_CCR4	Reserved	CCR4[15:0]																
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	TIMx_BDTR	Reserved	MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK [1:0]		DT[7:0]								
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	TIMx_DCR	Reserved						DBL[4:0]				Reserved				DBA[4:0]			
	Reset value							0	0	0	0	0				0	0	0	0	0

0x4C	TIMx_DMAR	Reserved	DMAB[15:0]														
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

STC MCU

16 通用定时器（PWM2 到 PWM5）

16.1PWM2 到 PWM5 简介

通用定时器包含一个 16 位或 32 位自动重载计数器，该计数器由可编程预分频器驱动。

它们可用于多种用途，包括测量输入信号的脉冲宽度（ 输入捕获 ）或生成输出波形（ 输出比 较和 PWM）。

使用定时器预分频器和 RCC 时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制 到几毫秒。

这些定时器彼此完全独立，不共享任何资源。它们可以同步操作。

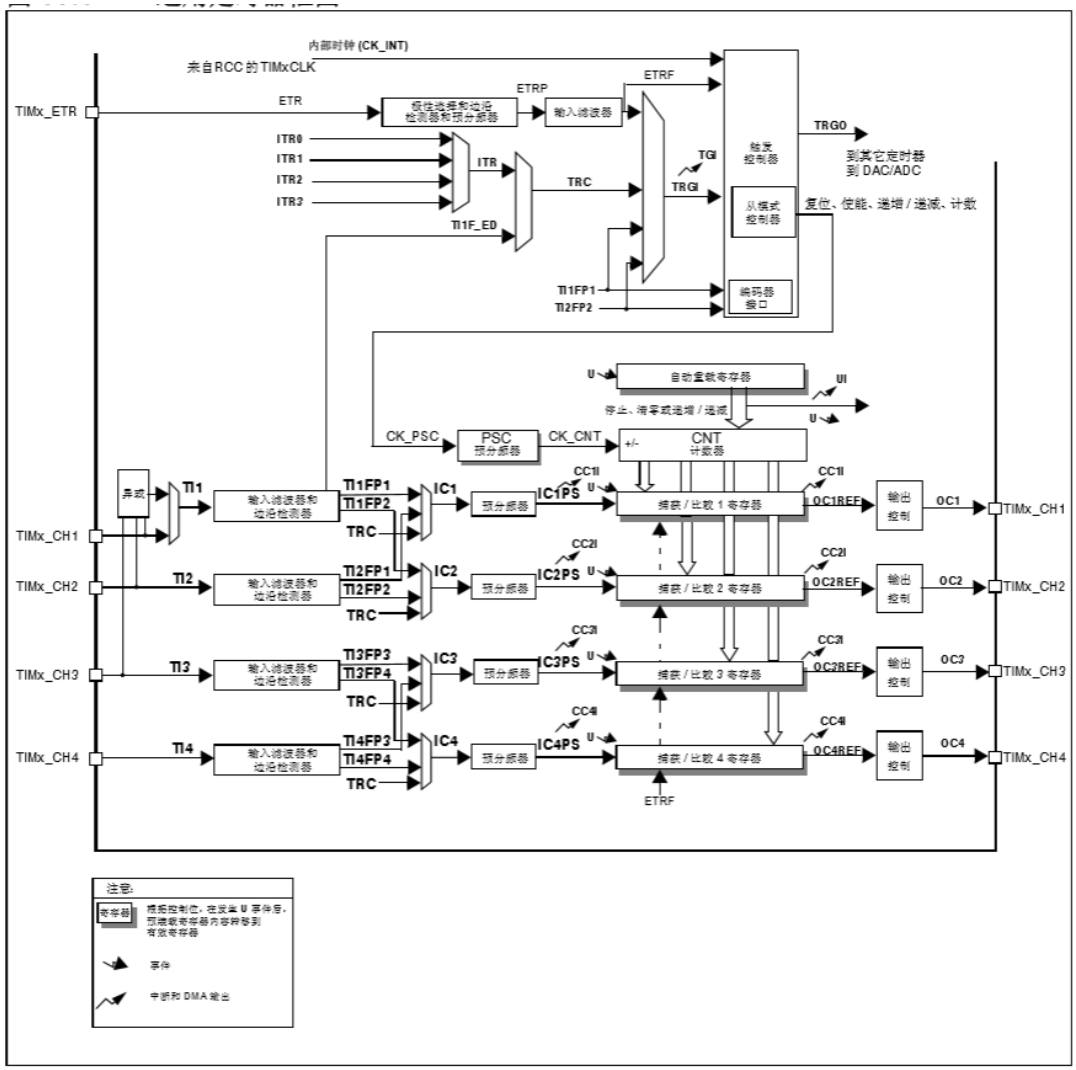
16.2PWM2 到 PWM5 主要特性

通用 PWMx 定时器具有以下特性：

- 16 位（PWM3 和 PWM4）或 32 位（PWM2 和 PWM5） 递增、递减和递增/递减自动重载计数器。
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（即运行时修改），分频系数介 于 1 到 65536 之间。
- 多达 4 个独立通道，可用于：
  - 输入捕获
  - 输出比较
  - PWM 生成（边沿和中心对齐模式）
  - 单脉冲模式输出
- 使用外部信号控制定时器且可实现多个定时器互连的同步电路。
- 发生如下事件时生成中断/DMA 请求：
  - 更新：计数器上溢/下溢、计数器初始化（通过软件或内部/外部触发）
  - 触发事件（计数器启动、停止、初始化或通过内部/外部触发计数）
  - 输入捕获
  - 输出比较
- 支持定位用增量（正交）编码器和霍尔传感器电路
- 外部时钟触发输入或逐周期电流管理

图 119. 通用定时器框图

STC MCU



16.3PWM2 到 PWM5 功能说明

16.3.1时基单元

可编程定时器的主要模块由一个 16 位/32 位计数器及其相关的自动重装寄存器组成。此计数器可采用递增方式计数。计数器的时钟可通过预分频器进行分频。

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

时基单元包括：

- 计数器寄存器 (PWMx_CNT)
- 预分频器寄存器 (PWMx_PSC)
- 自动重载寄存器 (PWMx_ARR)

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存 器。预装载寄存器的内容既可以直接传送到影子寄存器，也可以在每次发生更新事件 (UEV) 时传送到影子寄存器，这取决于 PWMx_CR1 寄存器中的自动重载预装载使能位 (ARPE)。当 计数器达到上溢值（或者在递减计数时达到下溢值）并且 PWMx_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生 进行详细介绍。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 PWMx_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器（有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明）。

请注意，真正的计数器使能信号 CNT_EN 在 CEN 置 1 的一个时钟周期后被置 1。

预分频器说明

预分频器可对计数器时钟频率进行分频，分频系数介于 1 到 65536 之间。该预分频器基于 16 位/32 位寄存器（PWMx_PSC 寄存器）所控制的 16 位计数器。由于该控制寄存器具有缓 冲功能，因此预分频器可实现实时更改。而新的预分频比将在下一更新事件发生时被采用。

图 120 和 图 121 给出了在预分频比发生实时变化时一些计数器行为的示例：

图 120. 预分频器分频由 1 变为 2 时的计数器时序图

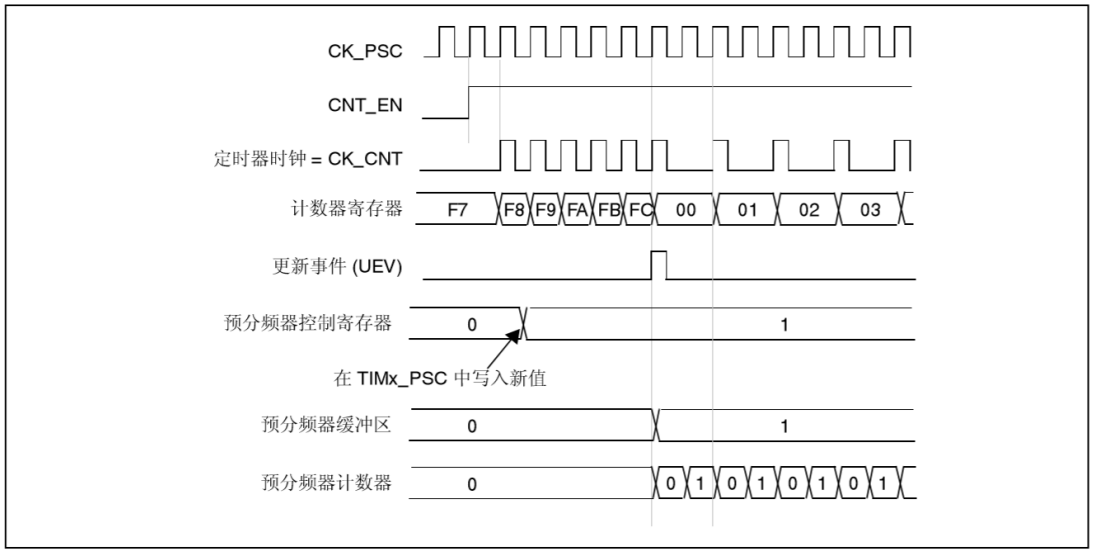
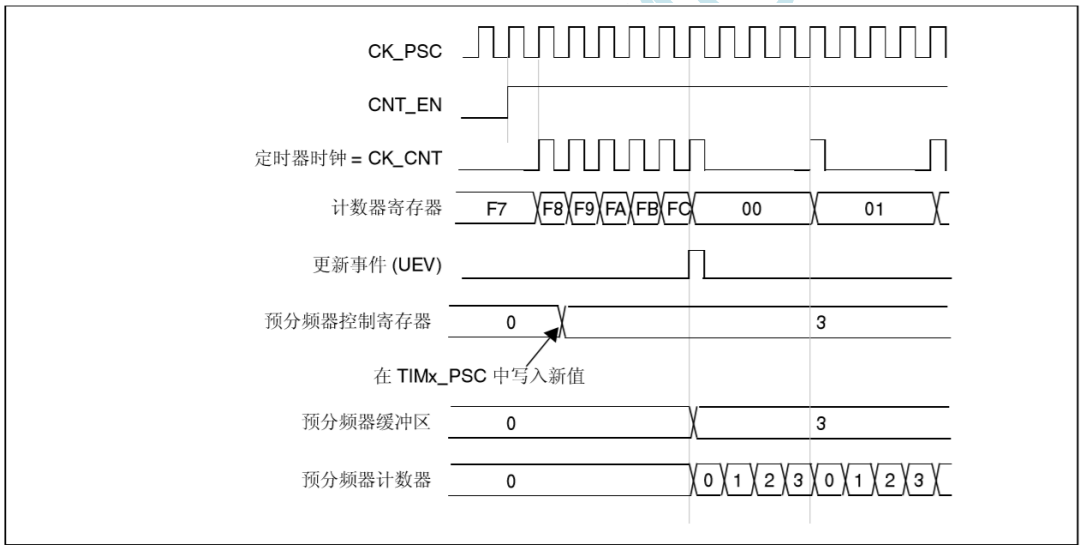


图 121. 预分频器分频由 1 变为 4 时的计数器时序图



16.3.2 计数器模式

递增计数模式

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

每次发生计数器上溢时会生成更新事件，或将 PWMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。

通过软件将 PWMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 PWMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

- 预分频器的缓冲区中将重新装载预装载值（PWMx_PSC 寄存器的内容）
- 自动重载影子寄存器将以预装载值进行更新

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

图 122. 计数器时序图，1 分频内部时钟

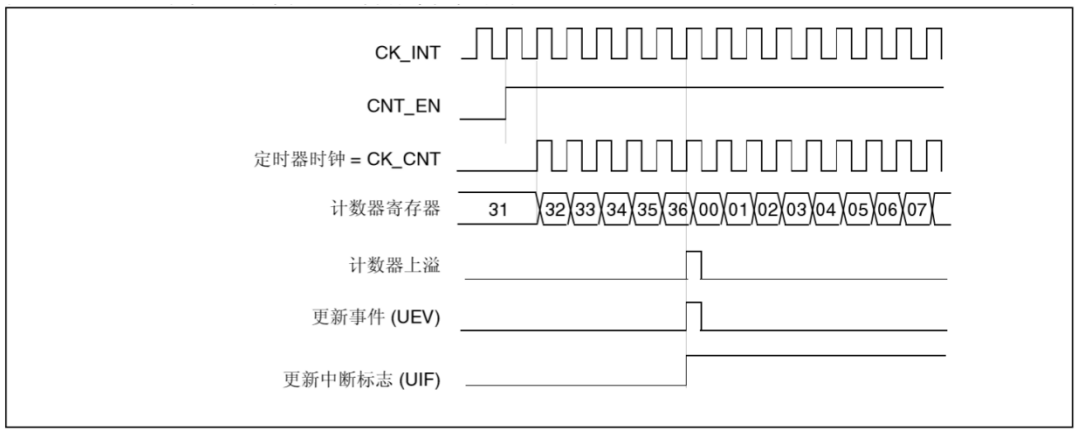


图 123. 计数器时序图，2 分频内部时钟

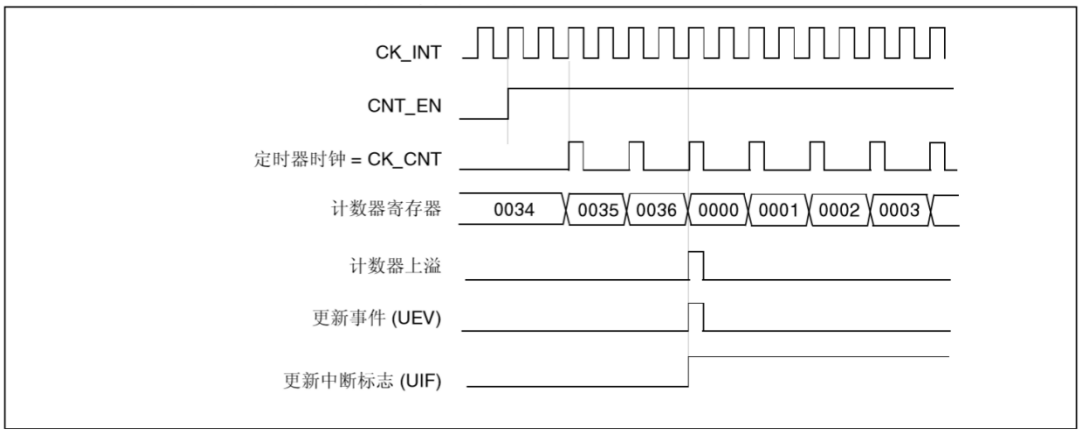


图 124. 计数器时序图，4 分频内部时钟

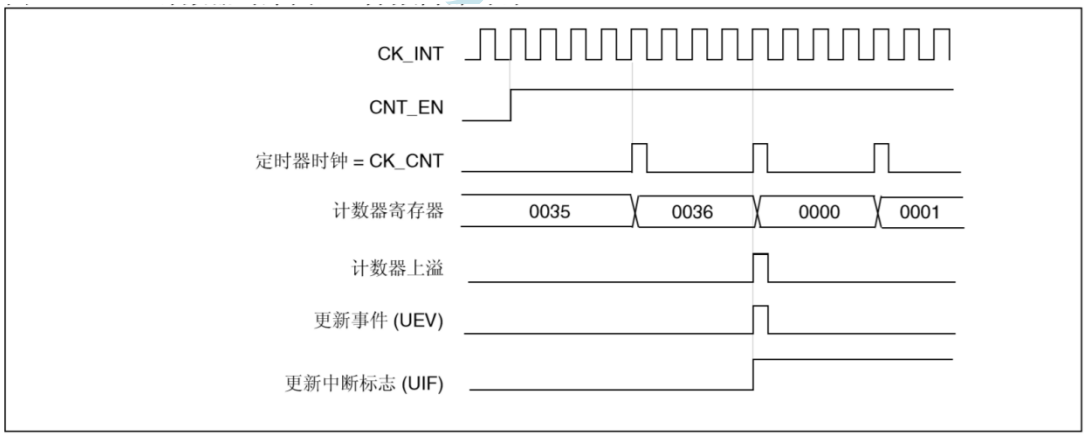


图 125. 计数器时序图，N 分频内部时钟

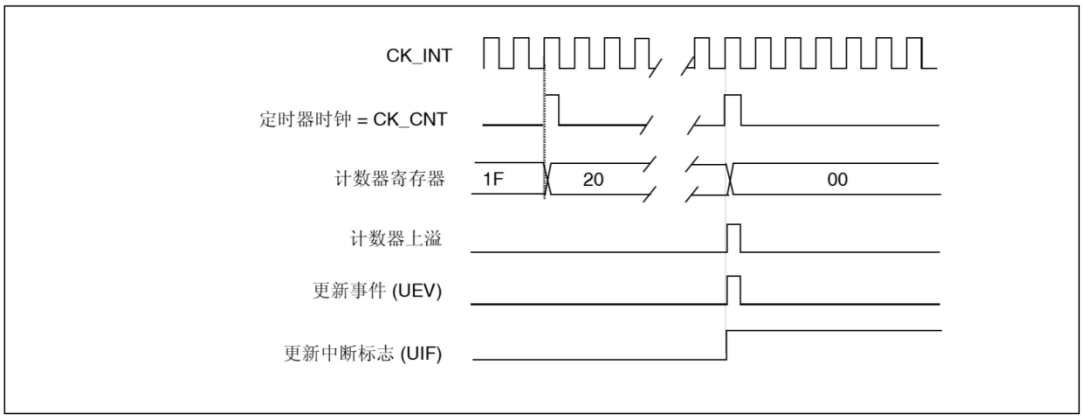


图 126. 计数器时序图，ARPE=0 时更新事件（PWMx_ARR 未预装载）

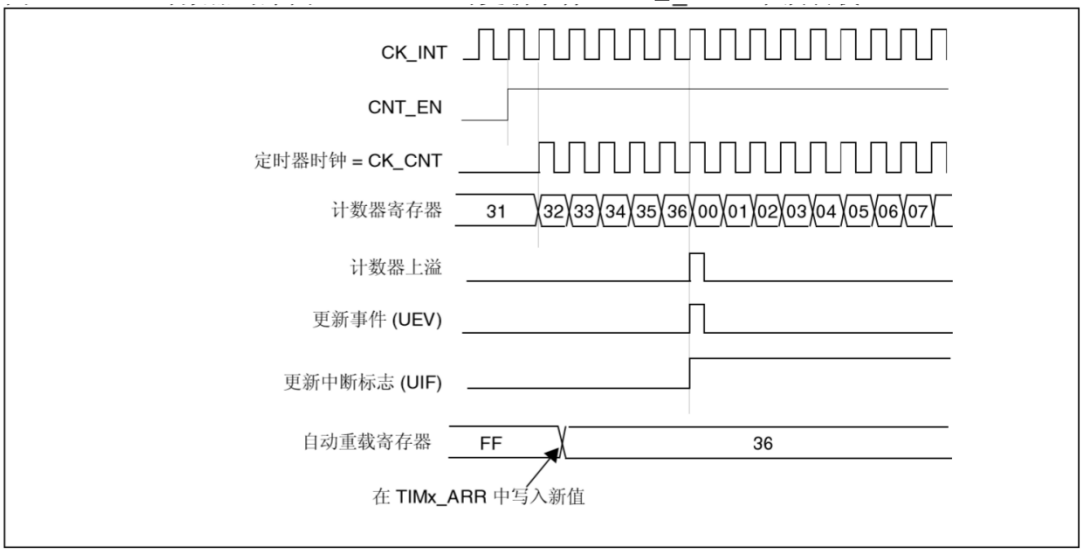
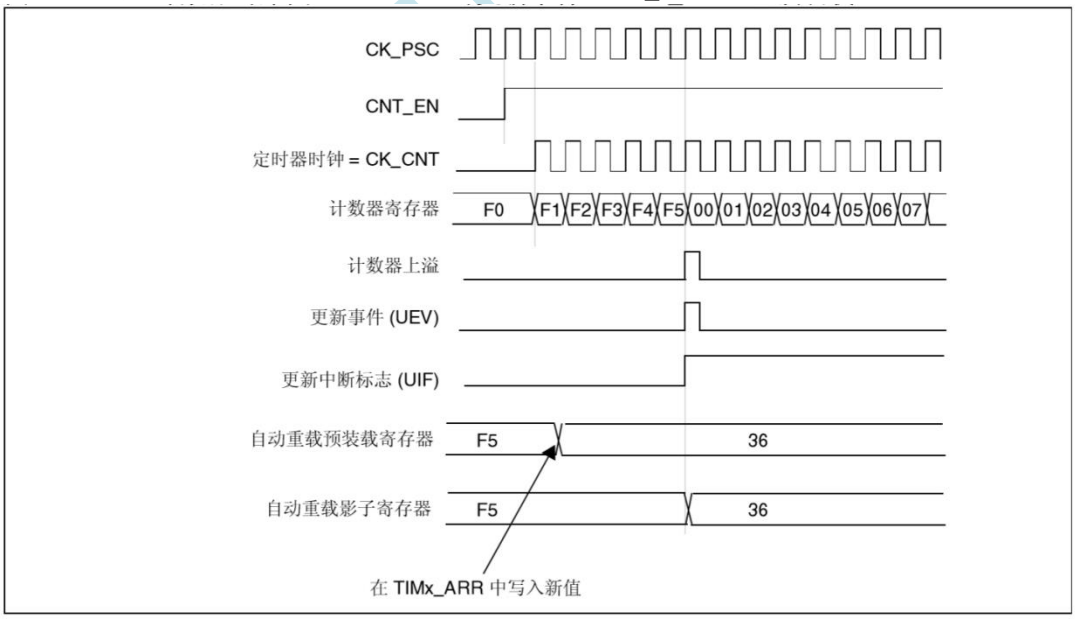


图 127. 计数器时序图，ARPE=1 时更新事件（PWMx_ARR 已预装载）



递减计数模式

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

每次发生计数器下溢时会生成更新事件，或将 PWMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件

通过软件将 PWMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器会重新从当前自动重载值开始计数，而预分频器计数器则重新从 0 开始计数（但预分 频比保持不变）。

此外，如果 PWMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

- 预分频器的缓冲区中将重新装载预装载值（PWMx_PSC 寄存器的内容）。
- 自动重载活动寄存器将以预装载值（PWMx_ARR 寄存器的内容）进行更新。注意，自动 重载寄存器会在计数器重载之前得到更新，因此，下一个计数周期就是我们所希望的新 的周期长度。

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

图 128. 计数器时序图，1 分频内部时钟

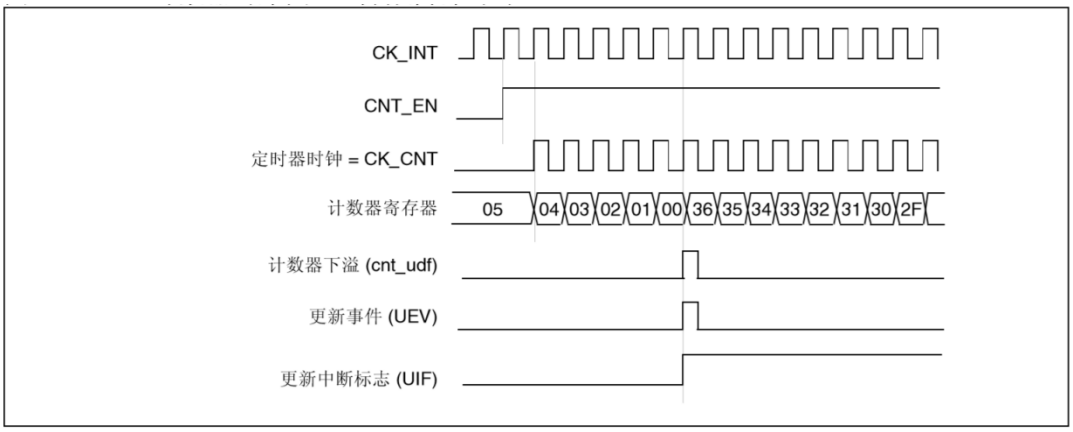


图 129. 计数器时序图，2 分频内部时钟

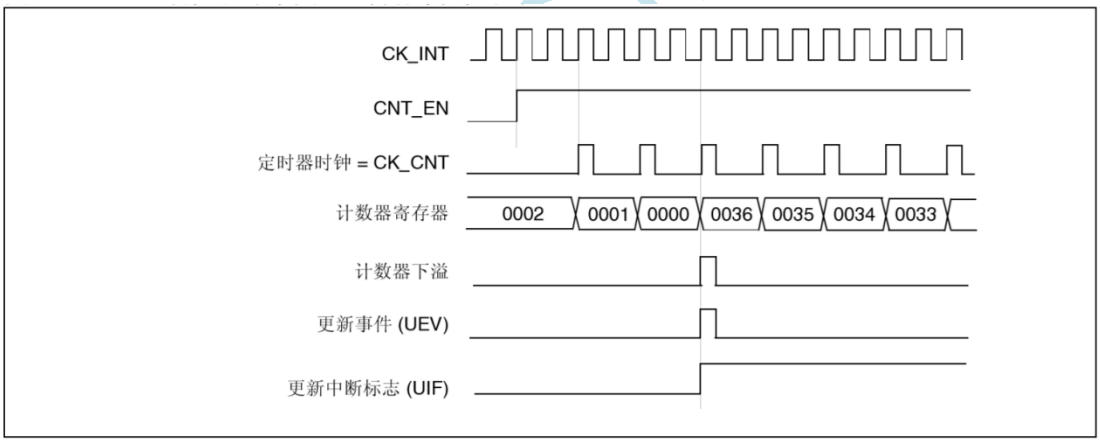


图 130. 计数器时序图，4 分频内部时钟

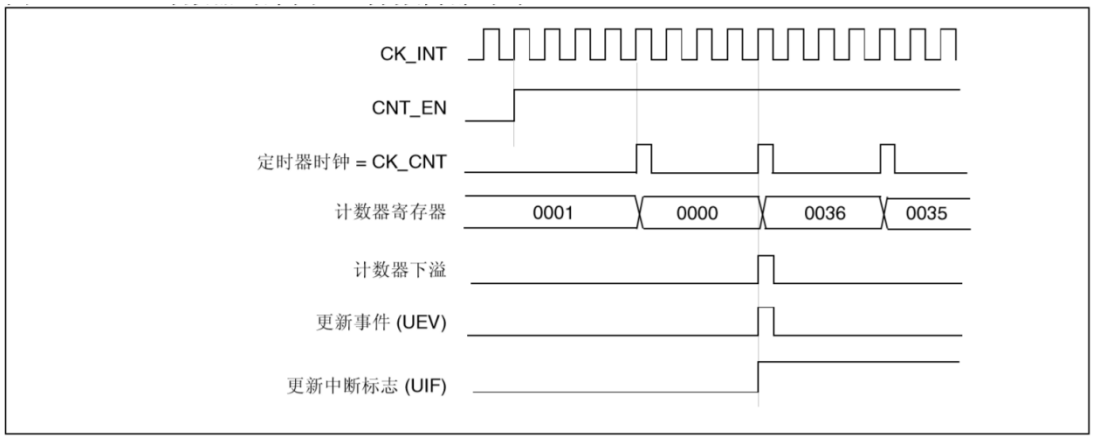


图 131. 计数器时序图，N 分频内部时钟

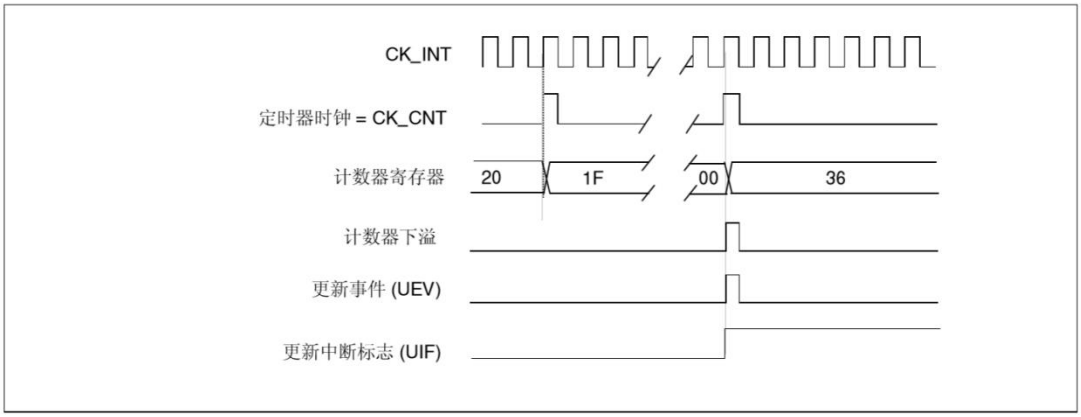
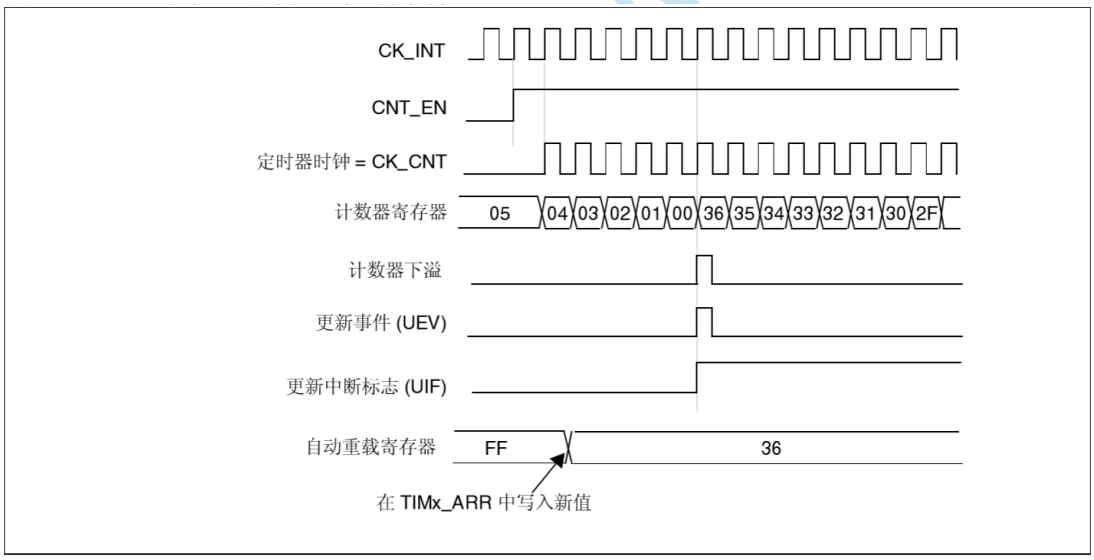


图 132. 计数器时序图，更新事件



中心对齐模式（递增/递减计数）

在中心对齐模式下，计数器从 0 开始计数到自动重载值（PWMx_ARR 寄存器的内容）— 1， 生成计数器上溢事件；然后从自动重载值开始向下计数到 1 并生成计数器下溢事件。之后从 0 开始重新计数。当 PWMx_CR1 寄存器中的 CMS 位不为“00”时，中心对齐模式有效。将通道配置为输出模 式时，其输出比较中断标志将在以下模式下置 1，即：计数器递减计数（中心对齐模式 1， CMS =“01”）、计数器递增计数（中心对齐模式 2，CMS =“10”）以及计数器递增/递 减计数（中心对齐模式 3，CMS =“11”）。此模式下无法写入方向位（PWMx_CR1 寄存器中的 DIR 位）。而是由硬件更新并指示当前计 数器方向。

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

始计数。

通过软件将 PWMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器仍会根据当前自动重载值进行递增和递减计数。

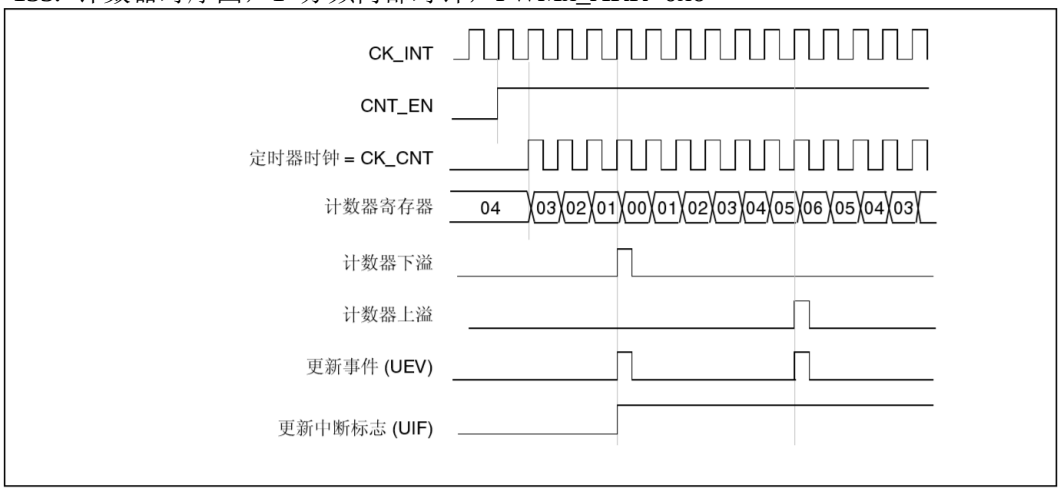
此外，如果 PWMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

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

以下各图以一些示例说明不同时钟频率下计数器的行为

图 133. 计数器时序图，1 分频内部时钟，PWMx_ARR=0x6



1. 此处使用了中心对齐模式 1（有关详细信息，请参见： PWMx 控制寄存器 1 (PWMx_CR1)）。

图 134. 计数器时序图，2 分频内部时钟

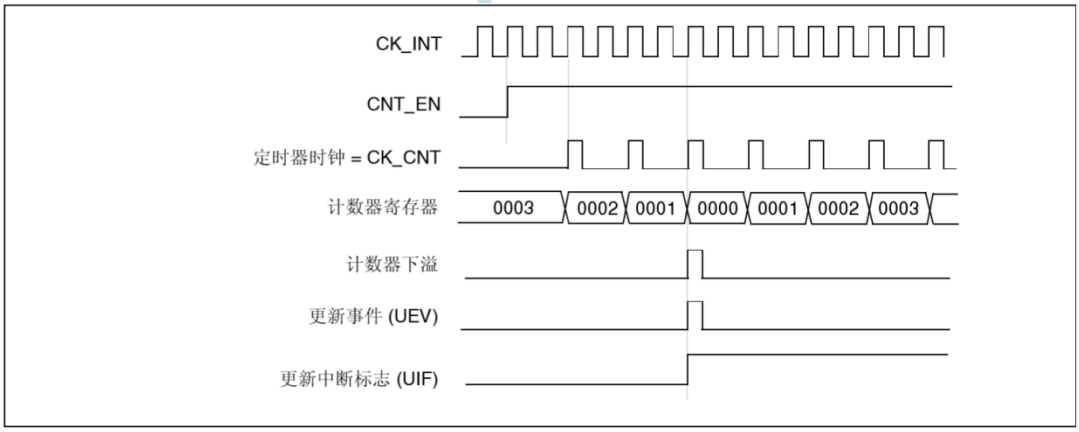
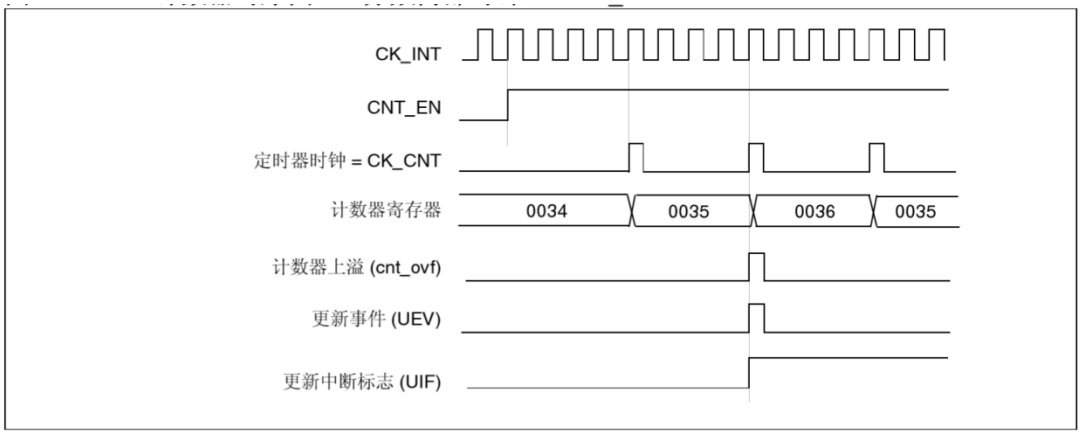


图 135. 计数器时序图，4 分频内部时钟，PWMx_ARR=0x36



1. 中心对齐模式 2 或模式 3 与上溢 UIF 结合使用。

图 136. 计数器时序图，N 分频内部时钟

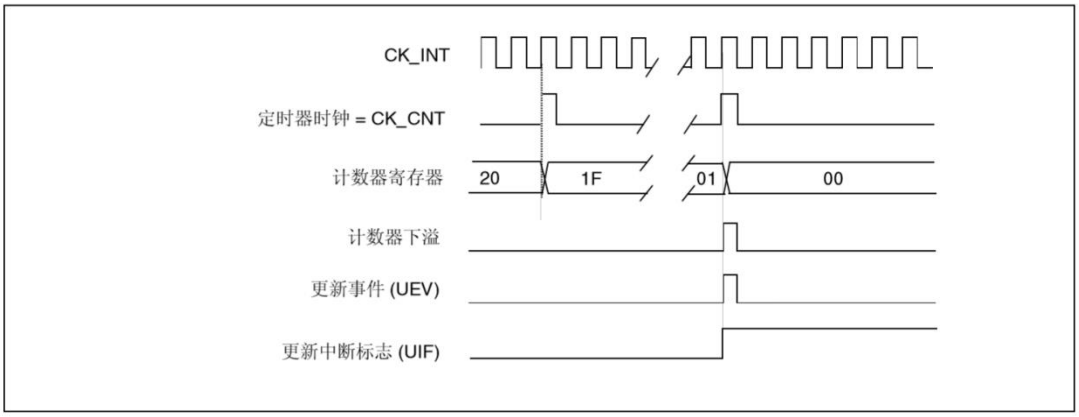


图 137. 计数器时序图，ARPE=1 时的更新事件（计数器下溢）

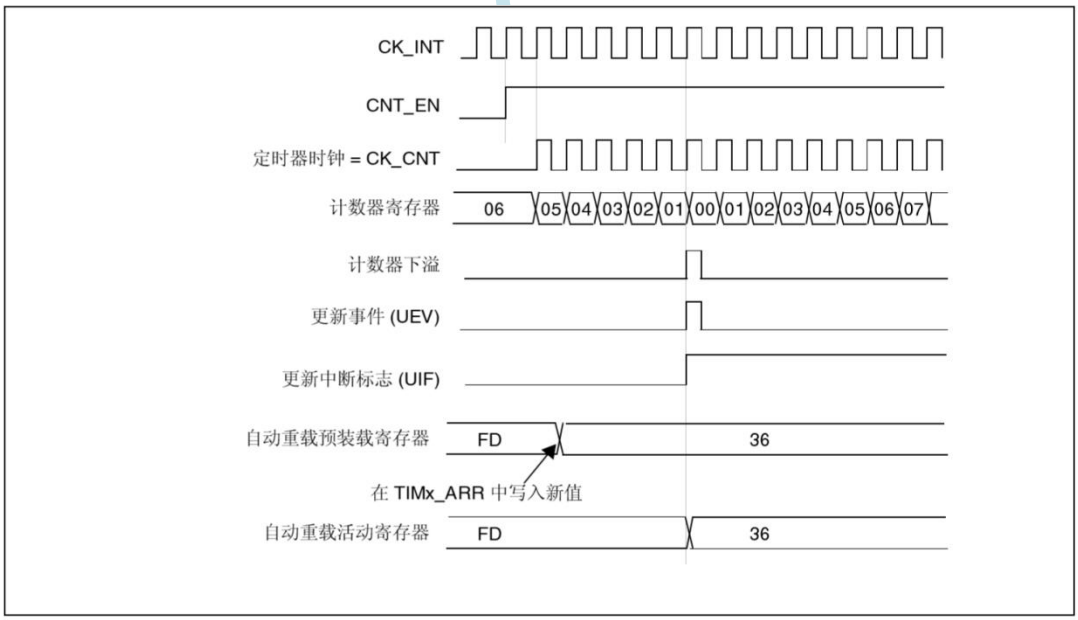
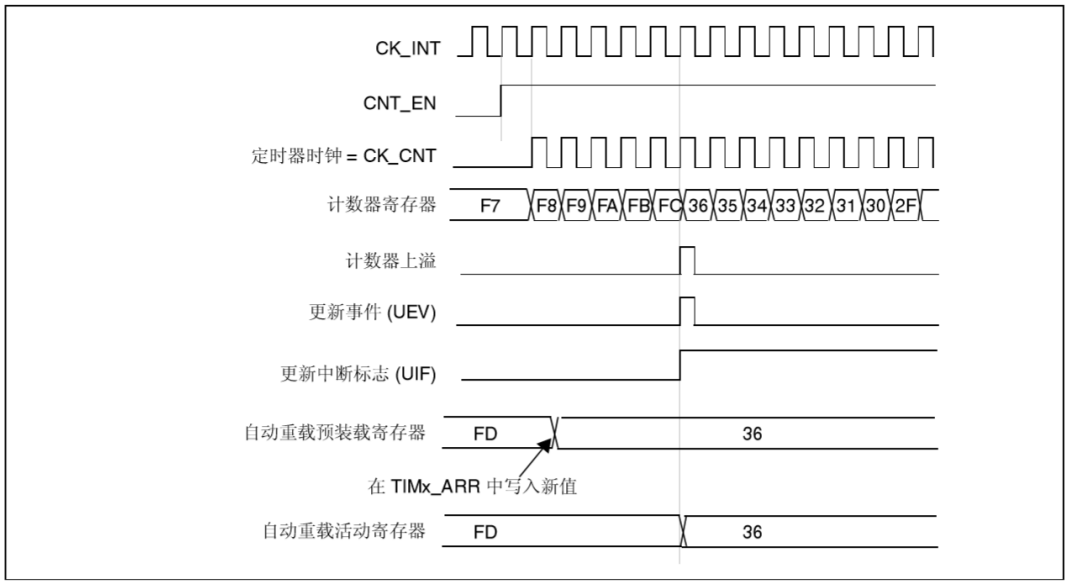


图 138. 计数器时序图，ARPE=1 时的更新事件（计数器上溢）



16.3.3时钟选择

计数器时钟可由下列时钟源提供：

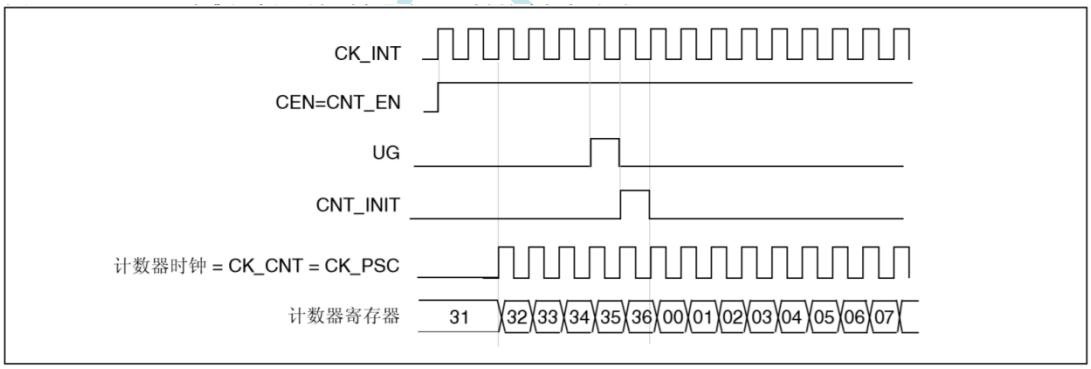
- 内部时钟 (CK_INT)
- 外部时钟模式 1：外部输入引脚 (TIx)
- 外部时钟模式 2：外部触发输入 (ETR)，仅适用于 PWM2、PWM3 和 PWM4。
- 内部触发输入 (ITRx)：使用一个定时器作为另一个定时器的预分频器，例如可以将定时器配置为定时器 2 的预分频器。有关详细信息，请参见将一个定时器用作另一个定时器的预分频器。

内部时钟源 (CK_INT)

如果禁止从模式控制器（PWMx_SMCR 寄存器中 SMS=000），则 CEN 位、DIR 位 （PWMx_CR1 寄存器中）和 UG 位（PWMx_EGR 寄存器中）为实际控制位，并且只能通过软 件进行更改（UG 除外，仍自动清零）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时 钟 CK_INT 提供。

图 139 显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）。

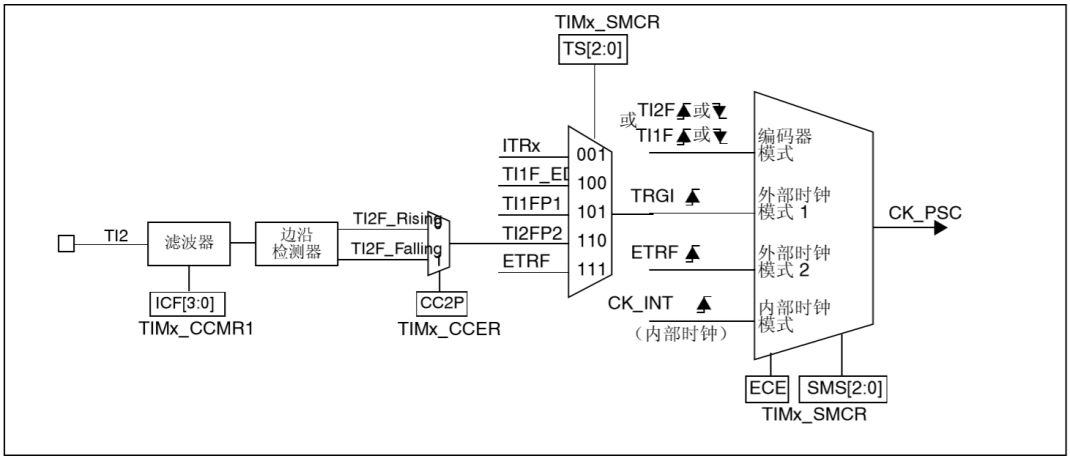
图 139. 正常模式下的控制电路，1 分频内部时钟



外部时钟源模式 1

当 PWMx_SMCR 寄存器中的 SMS=111 时，可选择此模式。计数器可在选定的输入信号上出现上升沿或下降沿时计数。

图 140. TI2 外部时钟连接示例



例如，要使递增计数器在 TI2 输入出现上升沿时计数，请执行以下步骤：

1. 通过在 PWMx_CCMR1 寄存器中写入 CC2S=“01”来配置通道 2，使其能够检测 TI2 输 入的上升沿。

2. 通过在 PWMx_CCMR1 寄存器中写入 IC2F[3:0] 位来配置输入滤波时间（如果不需要任 何滤波，请保持 IC2F=0000）。

注 意 ： 由 于 捕 获 预 分 频 器 不 用 于 触 发 操 作 ， 因 此 无 需 对 其 进 行 配 置 。

3. 通过在 PWMx_CCER 寄存器中写入 CC2P=0 和 CC2NP=0 来选择上升沿极性。

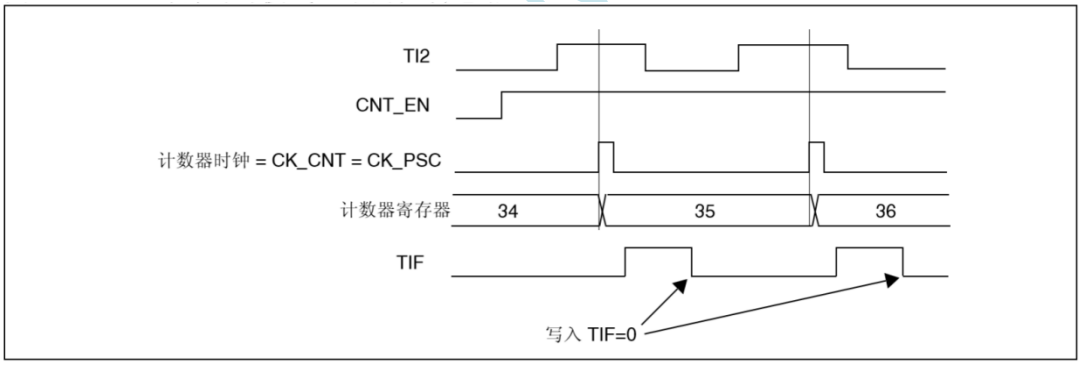
4. 通过在 PWMx_SMCR 寄存器中写入 SMS=111，使定时器在外部时钟模式 1 下工作。

5. 通过在 PWMx_SMCR 寄存器中写入 TS=110 来选择 TI2 作为输入源。 6. 通过在 PWMx_CR1 寄存器中写入 CEN=1 来使能计数器。

当 TI2 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。

TI2 的上升沿与实际计数器时钟之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 141. 外部时钟模式 1 下的控制电路



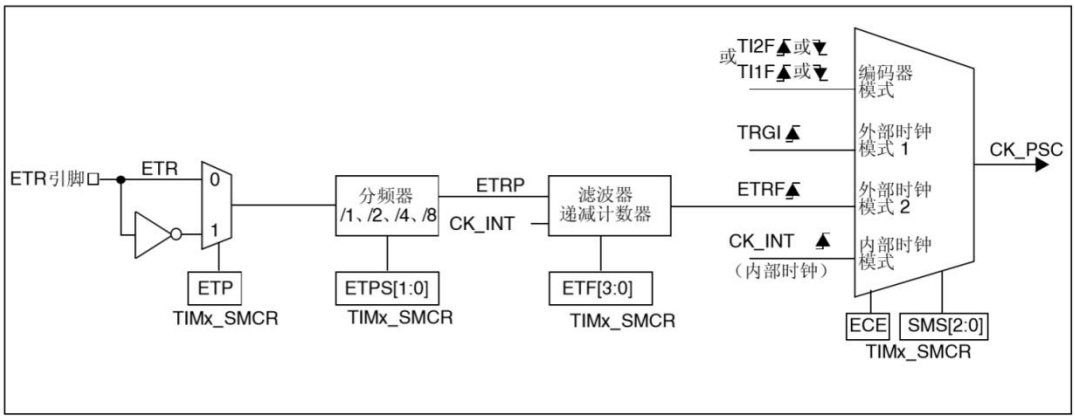
外部时钟源模式 2

通过在 PWMx_SMCR 寄存器中写入 ECE=1 可选择此模式。

计数器可在外部触发输入 ETR 出现上升沿或下降沿时计数。

图 142 简要介绍了外部触发输入模块。

图 142. 外部触发输入模块



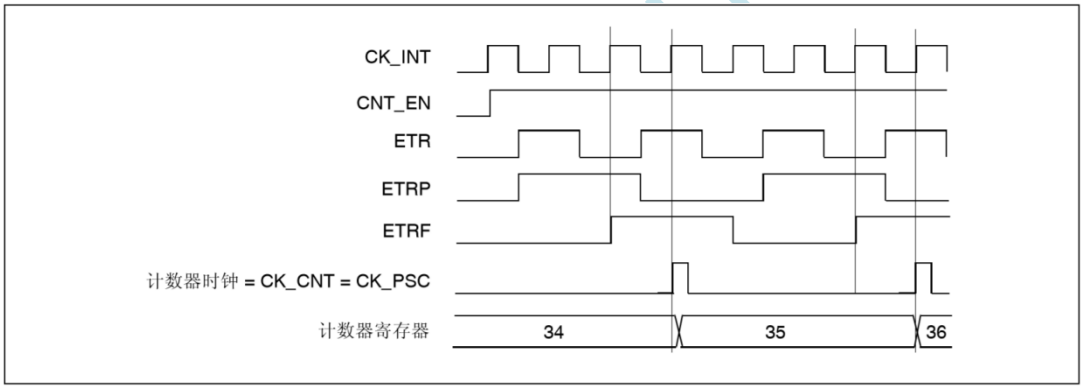
例如，要使递增计数器在 ETR 每出现 2 个上升沿时计数，请执行以下步骤：

1. 由于此例中不需滤波器，因此在 PWMx_SMCR 寄存器中写入 ETF[3:0]=0000。
2. 通过在 PWMx_SMCR 寄存器中写入 ETPS[1:0]=01 来设置预分频器
3. 通过在 PWMx_SMCR 寄存器中写入 ETP=0 来选择 ETR 引脚的上升沿检测
4. 通过在 PWMx_SMCR 寄存器中写入 ECE=1 来使能外部时钟模式 2。
5. 通过在 PWMx_CR1 寄存器中写入 CEN=1 来使能计数器。

ETR 每出现 2 个上升沿，计数器计数一次。

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

图 143. 外部时钟模式 2 下的控制电路



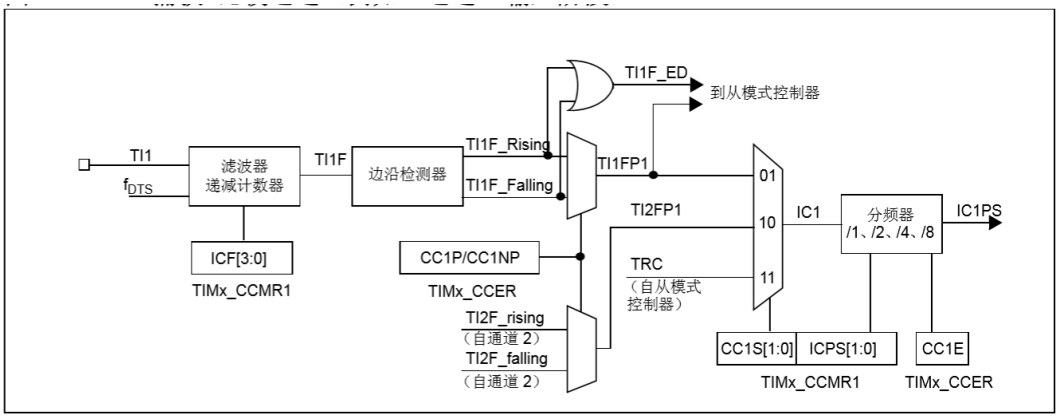
16.3.4捕获/比较通道

每个捕获/比较通道均围绕一个捕获/比较寄存器（包括一个影子寄存器）、一个捕获输入阶 段（数字滤波、多路复用和预分频器）和一个输出阶段（比较器和输出控制）构建而成。

下图概括介绍了一个捕获/比较通道。

输入阶段对相应的 TIx 输入进行采样，生成一个滤波后的信号 TIxF。然后，带有极性选择功能的边沿检测器生成一个信号 (TIxFPx)，该信号可用作从模式控制器的触发输入，也可用作 捕获命令。该信号先进行预分频 (ICxPS)，而后再进入捕获寄存器。

图 144. 捕获/比较通道（例如：通道 1 输入阶段）



输出阶段生成一个中间波形作为基准：OCxRef（高电平有效）。链的末端决定终输出信 号的极性。

图 145. 捕获/比较通道 1 主电路

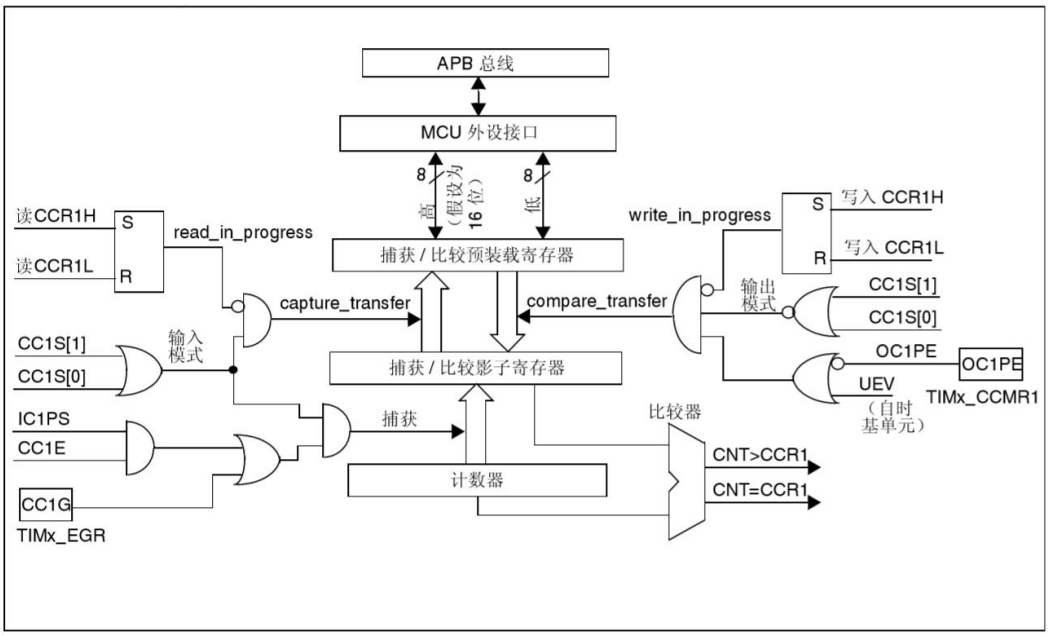
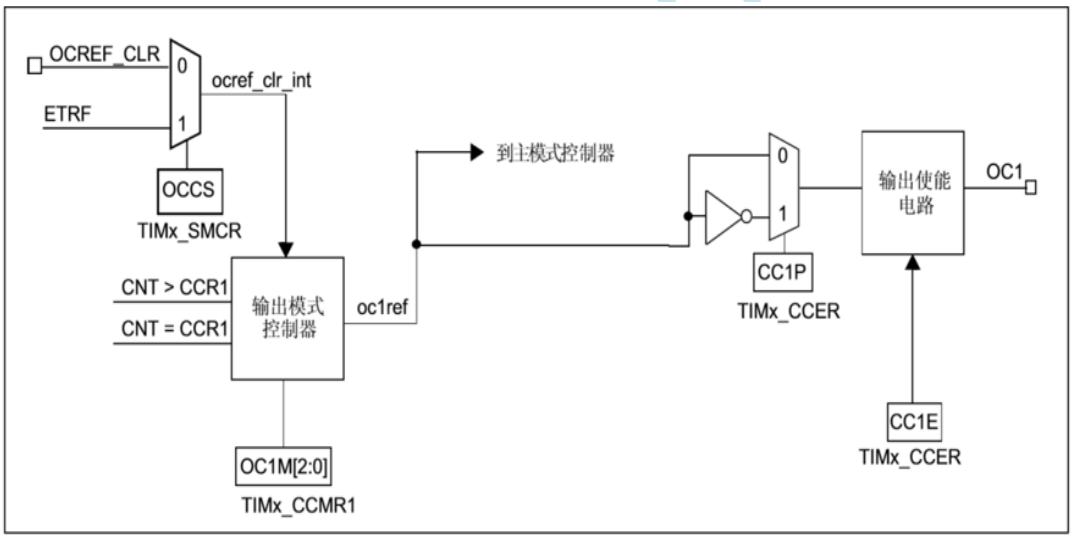


图 146. 捕获/比较通道的输出阶段（通道 1）



捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。始终可通过读写操作访问预装载 寄存器。

在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容复制到预装载寄存 器中。

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

16.3.5输入捕获模式

在输入捕获模式下，当相应的 ICx 信号检测到跳变沿后，将使用捕获/比较寄存器(PWMx_CCRx) 来锁存计数器的值。发生捕获事件时，会将相应的 CCXIF 标志（PWMx_SR 寄存器）置 1 并可发送中断或 DMA 请求（如果已使能）。如果发生捕获事件时 CCxIF 标志已处于高位，则会将重复捕获标志 CCxOF（PWMx_SR 寄存器）置 1。可通过软件向 CCxIF 写入 0 来给 CCxIF 清零，或读取存储在 PWMx_CCRx 寄存器中的已捕获数据。向 CCxOF 写入 0 后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 PWMx_CCR1 中。具体操作步骤如下：

- 选择有效输入：PWMx_CCR1 必须连接到 TI1 输入，因此向 PWMx_CCMR1 寄存器中的 CC1S 位写入 01。只要 CC1S 不等于 00，就会将通道配置为输入模式，并且 PWMx_CCR1 寄存器

将处于只读状态。

- 根据连接到定时器的信号，对所需的输入滤波时间进行编程（如果输入为 TIx 输入之 一，则对 PWMx_CCMRx 寄存器中的 ICxF 位进行编程）。假设信号变化时，输入信号多在 5 个内部时钟周期内发生抖动。因此，我们必须将滤波时间设置为大于 5 个内部时钟周期。在检测到 8 个具有新电平的连续采样（以 fDTS 频率采样）后，可以确认 TI1 上的跳变沿。然后向 PWMx_CCMR1 寄存器中的 IC1F 位写入 0011。
- 通过向 PWMx_CCER 寄存器中的 CC1P 位和 CC1NP 位写入 0，选择 TI1 通道的有效转换边沿（本例中为上升沿）。
- 对输入预分频器进行编程。在本例中，我们希望每次有效转换时都执行捕获操作，因此 需要禁止预分频器（向 PWMx_CCMR1 寄存器中的 IC1PS 位写入 00）。
- 通过将 PWMx_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
- 如果需要，可通过将 PWMx_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求，并且/或者通过将该寄存器中的 CC1DE 位置 1 来使能 DMA 请求。

发生输入捕获时：

- 发生有效跳变沿时，PWMx_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。
- 根据 CC1IE 位生成中断。
- 根据 CC1DE 位生成 DMA 请求。

要处理重复捕获，建议在读出捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出 标志之后与读取数据之前可能出现的重复捕获信息

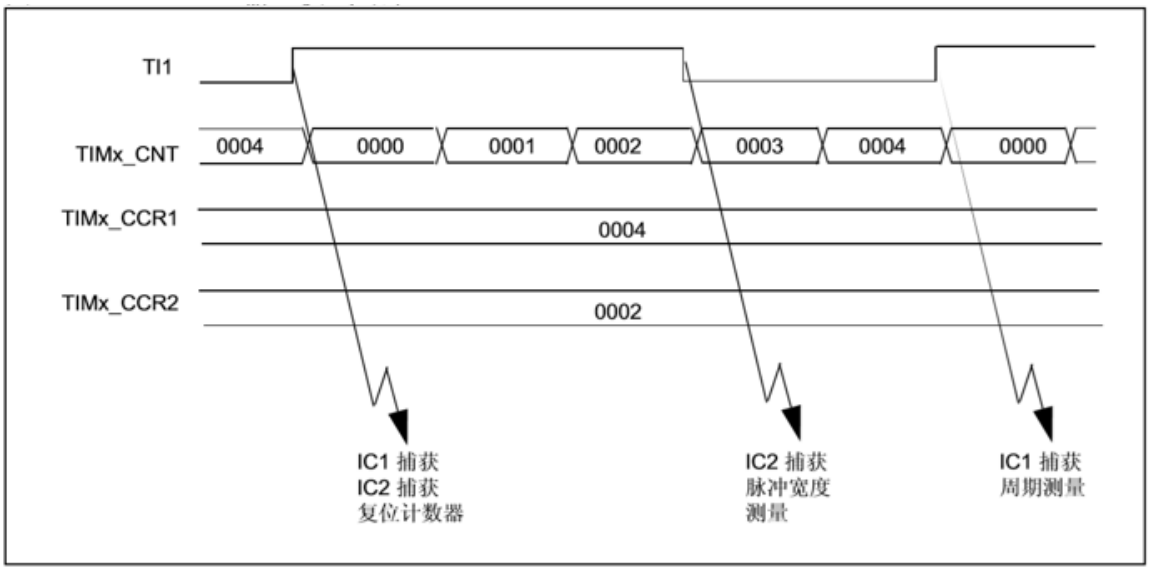
注 意 ： 通 过 软 件 将 PWMx_EGR 寄存器中的相应 CCxG 位置 1 可生成 IC 中断和 / 或 DMA 请求。

16.3.6 PWM 输入模式

此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这两个 ICx 信号在边沿处有效，但极性相反。
- 选择两个 TIxFP 信号之一作为触发输入，并将从模式控制器配置为复位模式。 例如，可通过以下步骤对应用于 TI1 的 PWM 的周期（位于 PWMx_CCR1 寄存器中）和占空比（位于 PWMx_CCR2 寄存器中）进行测量（取决于 CK_INT 频率和预分频器的值）：
- 选择 PWMx_CCR1 的有效输入：向 PWMx_CCMR1 寄存器中的 CC1S 位写入 01（选择 TI1）。
- 选择 TI1FP1 的有效极性（用于 PWMx_CCR1 中的捕获和计数器清零）：向 CC1P 位和 CC1NP 位写入“0”（上升沿有效）。
- 选择 PWMx_CCR2 的有效输入：向 PWMx_CCMR1 寄存器中的 CC2S 写入 10（选择 TI1）。
- 选择 TI1FP2 的有效极性（用于 PWMx_CCR2 中的捕获）：向 CC2P 位和 CC2NP 位写入“1”（下降沿有效）。
- 选择有效触发输入：向 PWMx_SMCR 寄存器中的 TS 位写入 101（选择 TI1FP1）。 ● 将从模式控制器配置为复位模式：向 PWMx_SMCR 寄存器中的 SMS 位写入 100。 ● 使能捕获：向 PWMx_CCER 寄存器中的 CC1E 位和 CC2E 位写入“1”。

图 147. PWM 输入模式时序



16.3.7 强制输出模式

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

要将输出比较信号 (OCXREF/OCx) 强制设置为有效电平，只需向相应 PWMx_CCMRx 寄存器中的 OCxM 位写入 101。ocxref 进而强制设置为高电平（OCxREF 始终为高电平有效），同时 OCx 获取 CCxP 极性位的相反值。

例如：CCxP=0（OCx 高电平有效）=> OCx 强制设置为高电平。

通过向 PWMx_CCMRx 寄存器中的 OCxM 位写入 100，可将 ocxref 信号强制设置为低电平。

无论如何，PWMx_CCRx 影子寄存器与计数器之间的比较仍会执行，而且允许将标志置 1。 因此可发送相应的中断和 DMA 请求。输出比较模式一节对此进行了介绍。

16.3.8 输出比较模式

此功能用于控制输出波形，或指示已经过某一时间段。

当捕获/比较寄存器与计数器之间相匹配时，输出比较功能：

- 将为相应的输出引脚分配一个可编程值，该值由输出比较模式（PWMx_CCMRx 寄存器 中的 OCxM 位）和输出极性（PWMx_CCER 寄存器中的 CCxP 位）定义。匹配时，输出引脚既可保持其电平 (OCXM=000)，也可设置为有效电平 (OCXM=001)、无效电平 (OCXM=010) 或进行翻转 (OCxM=011)。
- 将中断状态寄存器中的标志置 1（PWMx_SR 寄存器中的 CCxIF 位）。
- 如果相应中断使能位（PWMx_DIER 寄存器中的 CCXIE 位）置 1，将生成中断。
- 如果相应 DMA 使能位（PWMx_DIER 寄存器的 CCxDE 位，PWMx_CR2 寄存器的 CCDS 位，用来选择 DMA 请求）置 1，将发送 DMA 请求。

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

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

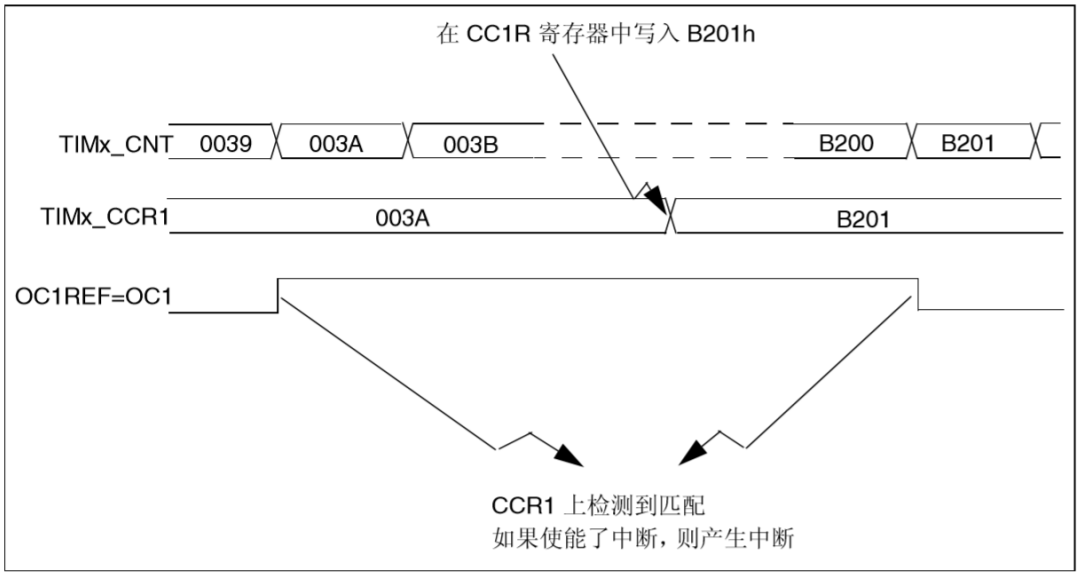
步骤：

1. 选择计数器时钟（内部、外部、预分频器）。
2. 在 PWMx_ARR 和 PWMx_CCRx 寄存器中写入所需数据。
3. 如果要生成中断和/或 DMA 请求，将 CCxIE 位和/或 CCxDE 位置 1。
4. 选择输出模式。例如，当 CNT 与 CCRx 匹配、未使用预装载 CCRx 并且 OCx 使能且为高电平有效时，必须写入 OCxM=011、OCxPE=0、CCxP=0 和 CCxE=1 来翻转 OCx 输出引脚。
5. 通过将 PWMx_CR1 寄存器中的 CEN 位置 1 来使能计数器。

可随时通过软件更新 PWMx_CCRx 寄存器以控制输出波形，前提是未使能预装载寄存器（OCxPE=0，否则仅当发生下一个更新事件 UEV 时，才会更新 PWMx_CCRx 影子寄存器）。图

148 列出了相关示例。

图 148. 输出比较模式，翻转 OC1



16.3.9 PWM 模式

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

通过向 PWMx_CCMRx 寄存器中的 OCxM 位写入 110（PWM 模式 1）或 111（PWM 模 式 2），可以独立选择各通道（每个 OCx 输出对应一个 PWM）的 PWM 模式。必须通过将 PWMx_CCMRx 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器，后通过将 PWMx_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器。

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

OCx 极性可使用 PWMx_CCER 寄存器的 CCxP 位来编程。既可以设为高电平有效，也可以设为低电平有效。OCx 输出通过将 PWMx_CCER 寄存器中的 CCxE 位置 1 来使能。有关详细信息，请参见 PWMx_CCERx 寄存器说明。

在 PWM 模式（1 或 2）下，PWMx_CNT 始终与 PWMx_CCRx 进行比较，以确定是 PWMx_CCRx PWMx_CNT 还是 PWMx_CNT PWMx_CCRx（取决于计数器计数方向）。不过，为了与 ETRF 相符（在下一个 PWM 周期之前，ETR 信号上的一个外部事件能够清除 OCxREF），OCREF 信号仅在以下情况下变为有效状态：

- 比较结果发生改变，或
- 输出比较模式（PWMx_CCMRx 寄存器中的 OCxM 位）从“冻结”配置（不进行比较，OCxM=“000”）切换为任一 PWM 模式（OCxM=“110”或“111”）。

定时器运行期间，可以通过软件强制 PWM 输出。

根据 PWMx_CR1 寄存器中的 CMS 位状态，定时器能够产生边沿对齐模式或中心对齐模式的 PWM 信号。

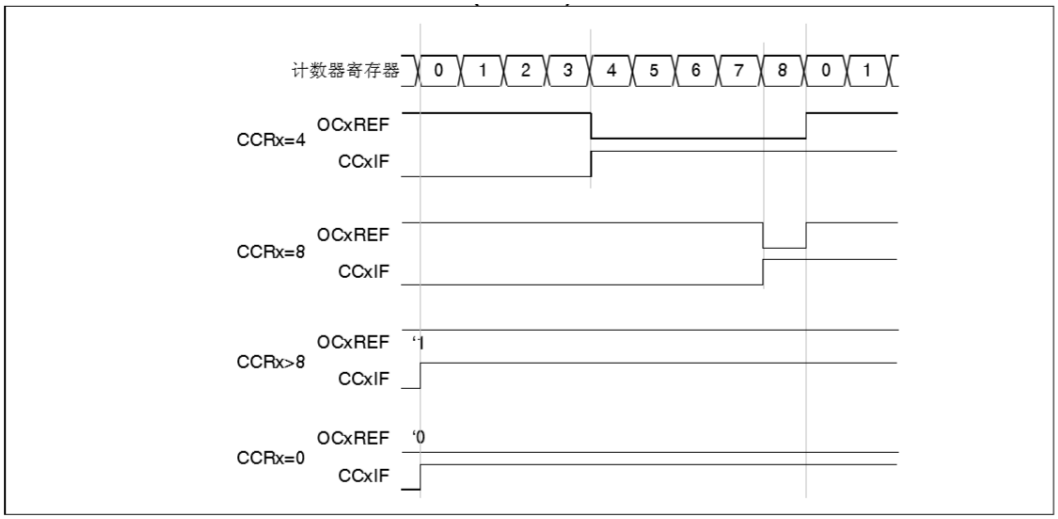
PWM 边沿对齐模式

递增计数配置

当 PWMx_CR1 寄存器中的 DIR 位为低时执行递增计数。请参见递增计数模式一节。

以下以 PWM 模式 1 为例。只要 PWMx_CNT < PWMx_CCRx，PWM 参考信号 OCxREF 便为高电平，否则为低电平。如果 PWMx_CCRx 中的比较值大于自动重载值（PWMx_ARR 中），则 OCxREF 保持为“1”。如果比较值为 0，则 OCxREF 保持为“0”。 图 149 举例介绍边沿对齐模式的一些 PWM 波形 (PWMx_ARR=8)。

图 149. 边沿对齐模式的 PWM 波形 (ARR=8)



递减计数配置

当 PWMx_CR1 寄存器中的 DIR 位为高时执行递减计数。请参见递减计数模式一节。

在 PWM 模式 1 下，只要 PWMx_CNT>PWMx_CCRx，参考信号 ocxref 便为低电平，否则为高电平。如果 PWMx_CCRx 中的比较值大于 PWMx_ARR 中的自动重载值，则 ocxref 保持为 “1”。此模式下不可能产生 0% 的 PWM 波形。

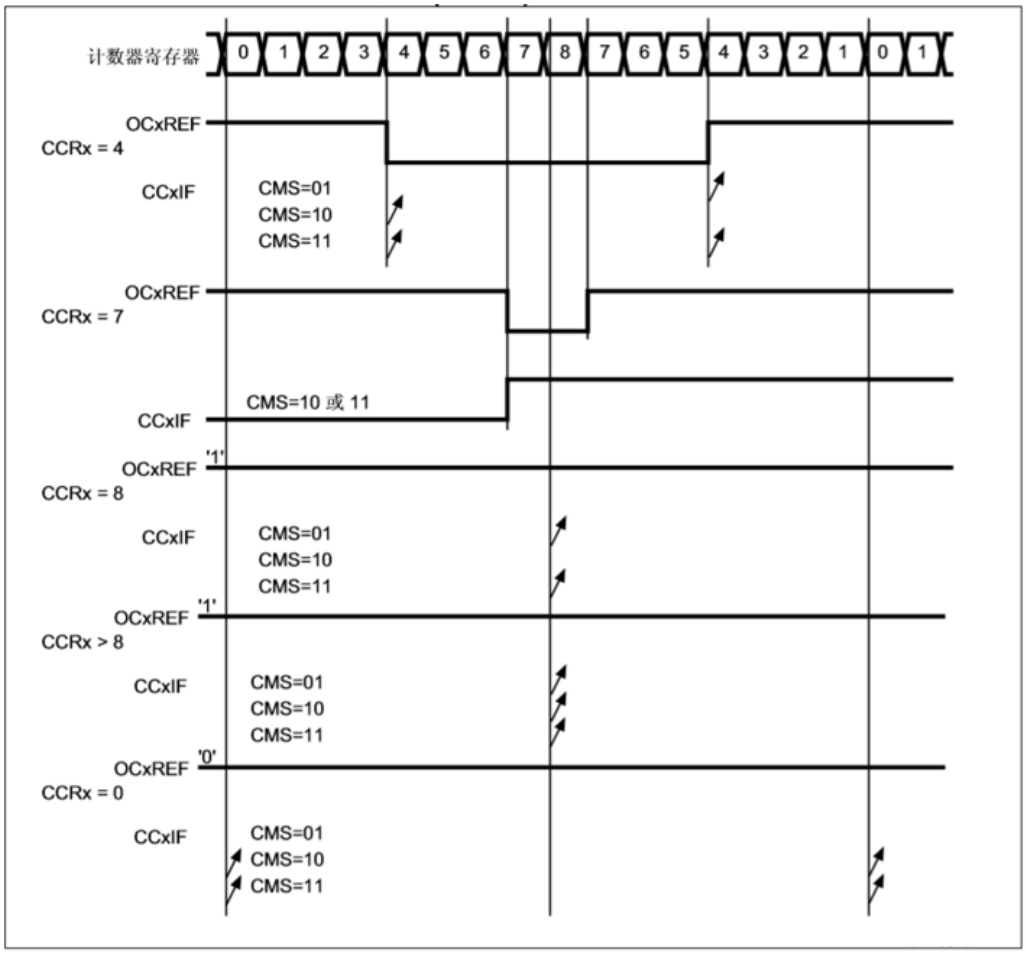
PWM 中心对齐模式

当 PWMx_CR1 寄存器中的 CMS 位不为“00”时（其余所有配置对 ocxref/OCx 信号具有相同的作用），中心对齐模式生效。根据 CMS 位的配置，可以在计数器递增计数、递减计数 或同时递增和递减计数时将比较标志置 1。PWMx_CR1 寄存器中的方向位 (DIR) 由硬件更新，不得通过软件更改。请参见中心对齐模式（递增 / 递减计数）一节。

图 150 显示了中心对齐模式的 PWM 波形，在此例中：

- PWMx_ARR=8，
- PWM 模式为 PWM 模式 1，
- 在根据 PWMx_CR1 寄存器中 CMS=01 而选择的中心对齐模式 1 下，当计数器递减计数时，比较标志置 1。

图 150. 中心对齐模式 PWM 波形 (ARR=8)



中心对齐模式使用建议：

- 启动中心对齐模式时将使用当前的递增/递减计数配置。这意味着计数器将根据写入 PWMx_CR1 寄存器中 DIR 位的值进行递增或递减计数。此外，不得同时通过 软件修改 DIR 和 CMS 位。
- 不建议在运行中心对齐模式时对计数器执行写操作，否则将发生意想不到的结果。尤 其是：
  - 如果写入计数器中的值大于自动重载值 (PWMx_CNT>PWMx_ARR)，计数方向不会 更新。例如，如果计数器之前递增计数，则继续递增计数。
  - 如果向计数器写入 0 或 PWMx_ARR 的值，计数方向会更新，但不生成更新事件 UEV。
- 使用中心对齐模式为保险的方法是：在启动计数器前通过软件生成更新（将 PWMx_EGR 寄存器中的 UG 位置 1），并且不要在计数器运行过程中对其执行写操作。

16.3.10 单脉冲模式

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

可以在输出比较模式或 PWM 模式下生成波形。将 PWMx_CR1 寄存器中的 OPM 位置 1，即可选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

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

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

图 151. 单脉冲模式示例

例如，用户希望达到这样的效果：在 TI2 输入引脚检测到正沿时，经过 tDELAY 的延迟，在 OC1 上产生一个长度为 tPULSE 的正脉冲。

使用 TI2FP2 作为触发 1：

- 在 PWMx_CCMR1 寄存器中写入 CC2S=01，将 TI2FP2 映射到 TI2。
- 在 PWMx_CCER 寄存器中写入 CC2P=0 和 CC2NP="0"，使 TI2FP2 能够检测上升沿。
- 在 PWMx_SMCR 寄存器中写入 TS=110，将 TI2FP2 配置为从模式控制器的触发 (TRGI)。
- 在 PWMx_SMCR 寄存器中写入 SMS="110"（触发模式），使用 TI2FP2 启动计数器。 OPM 波形通过对比较寄存器执行写操作来定义（考虑时钟频率和计数器预分频器）。
- tDELAY 由写入 PWMx_CCR1 寄存器的值定义。
- tPULSE 由自动重载值与比较值 (PWMx_ARR - PWMx_CCR1) 之差来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从"0"变为"1"，在计数器达到自动 重载值时由"1"变为"0"。为此，应在 PWMx_CCMR1 寄存器中写入 OC1M=111，以使能 PWM 模式 2。如果需要，可选择在 PWMx_CCMR1 寄存器的 OC1PE 和 PWMx_CR1 寄存器的 ARPE 中写入"1"，以使能预装载寄存器。这种情况下，必须在 PWMx_CCR1 寄存器中写入比较值并在 PWMx_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI2 上的外部触发事件。本例中，CC1P 的值为"0"。

在本例中，PWMx_CR1 寄存器中的 DIR 和 CMS 位应为低。

由于仅需要 1 个脉冲（单脉冲模式），因此应向 PWMx_CR1 寄存器的 OPM 位写入"1"，以便在发生下一更新事件（计数器从自动重载值返回到 0）时使计数器停止计数。PWMx_CR1 寄存器中的 OPM 位置"0"时，即选择重复模式。

特例：OCx 快速使能：

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

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

16.3.11 发生外部事件时清除 OCxREF 信号

对于给定通道，在 ETRF 输入施加高电平（相应 PWMx_CCMRx 寄存器中的 OCxCE 使能位置"1"），可使 OCxREF 信号变为低电平。OCxREF 信号将保持低电平，直到发生下一更新事件 (UEV)。

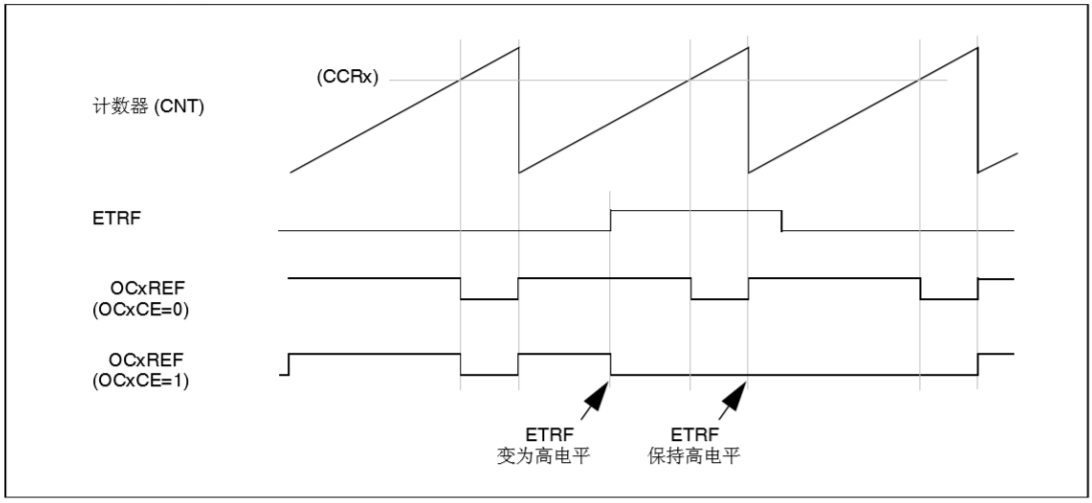
此功能仅能用于输出比较模式和 PWM 模式，而不适用于强制输出模式。

例如，ETR 信号可以连接到比较器的输出，用于控制电流。此时，ETR 必须如下配置：

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

图 152 对比了使能位 OCxCE 在不同值下的情况，显示了当 ETRF 输入变为高电平时 OCxREF 信号的行为。在本例中，定时器 PWMx 编程为 PWM 模式。

图 152. 清除 PWMx 的 OCxREF



1. 如果 PWM 的占空比为 100% (CCRx>ARR)，则下次计数器上溢时会再次使能 OCxREF。

16.3.12 编码器接口模式

选择编码器接口模式时，如果计数器仅在 TI2 边沿处计数，在 PWMx_SMCR 寄存器中写入 SMS=001；如果计数器仅在 TI1 边沿处计数，写入 SMS=010；如果计数器在 TI1 和 TI2 边沿处均计数，则写入 SMS=011。

通过编程 PWMx_CCER 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编程。

TI1 和 TI2 两个输入用于连接增量编码器。请参见 表 75。如果使能计数器（在 PWMx_CR1 寄存器的 CEN 位中写入“1”），则计数器的时钟由 TI1FP1 或 TI2FP2 上的每次有效信号转换提供。TI1FP1 和 TI2FP2 是进行输入滤波器和极性选择后 TI1 和 TI2 的信号，如果不进行滤波和反相，则 TI1FP1=TI1，TI2FP2=TI2。将根据两个输入的信号转换序列，产生计数脉 冲和方向信号。根据该信号转换序列，计数器相应递增或递减计数，同时硬件对 PWMx_CR1 寄存器的 DIR 位进行相应修改。任何输入（TI1 或 TI2）发生信号转换时，都会计算 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。

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。 编码器接口模式就相当于带有方向选择的外部时钟。这意味着，计数器仅在 0 到 PWMx_ARR 寄存器中的自动重载值之间进行连续计数（根据具体方向，从 0 递增计数到 ARR，或从 ARR 递减计数到 0）。因此，在启动前必须先配置 PWMx_ARR。同样，捕获、比较、预分频器、触发输出功能继续正常工作。

在此模式下，计数器会根据增量编码器的速度和方向自动进行修改，因此，其内容始终表示 编码器的位置。计数方向对应于所连传感器的旋转方向。下表汇总了可能的组合（假设 TI1 和 TI2 不同时切换）。

表 75. 计数方向与编码器信号的关系

有效边沿	相反信号的电平（**TI1FP1** 对应 **TI2**, **TI2FP2** 对应 **TI1**）	**TI1FP1** 信号		**TI2FP2** 信号	
		上升	下降	上升	下降
仅在 TI1 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
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
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

外部增量编码器可直接与 MCU 相连，无需外部接口逻辑。不过，通常使用比较器将编码器 的差分输出转换为数字信号。这样大幅提高了抗噪声性能。用于指示机械零位的第三个编码 器输出可与外部中断输入相连，用以触发计数器复位。

图 153 以计数器工作为例，说明了计数信号的生成和方向控制。同时也说明了选择双边沿时如何对输入抖动进行补偿。将传感器靠近其中一个切换点放置时可能出现这种情况。本例中 假设配置如下：

- CC1S=“01”（PWMx_CCMR1 寄存器，TI1FP1 映射到 TI1 上）
- CC2S=“01”（PWMx_CCMR2 寄存器，TI1FP2 映射到 TI2 上）
- CC1P=“0”，CC1NP=“0”，IC1F=“0000”（PWMx_CCER 寄存器，TI1FP1 未反相， TI1FP1=TI1）
- CC2P=“0”，CC2NP=“0”，IC2F=“0000”（PWMx_CCER 寄存器，TI2FP2 未反相， TI1FP2=TI2）
  - SMS=“011”（PWMx_SMCR 寄存器，两个输入在上升沿和下降沿均有效）
- CEN = 1（PWMx_CR1 寄存器，使能计数器）

图 153. 编码器接口模式下的计数器工作示例

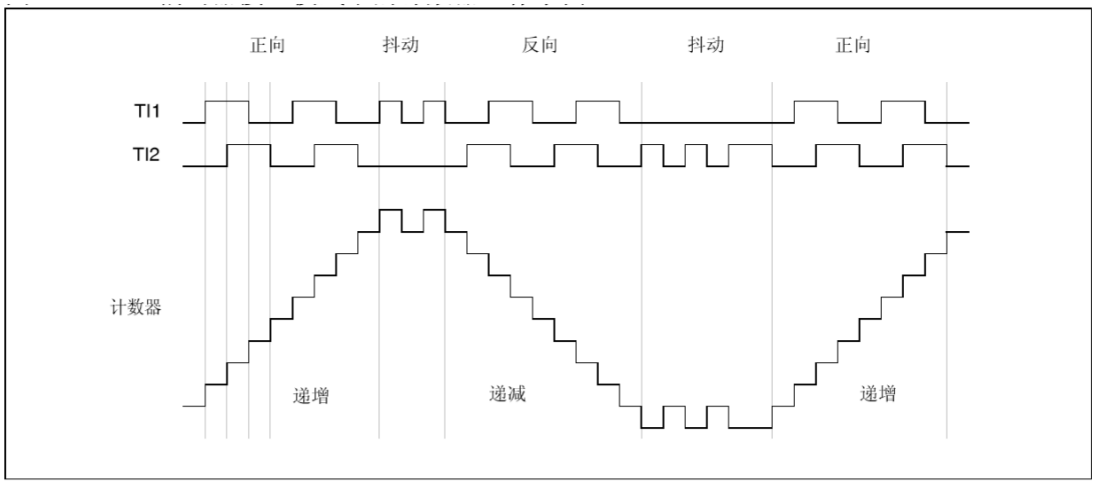
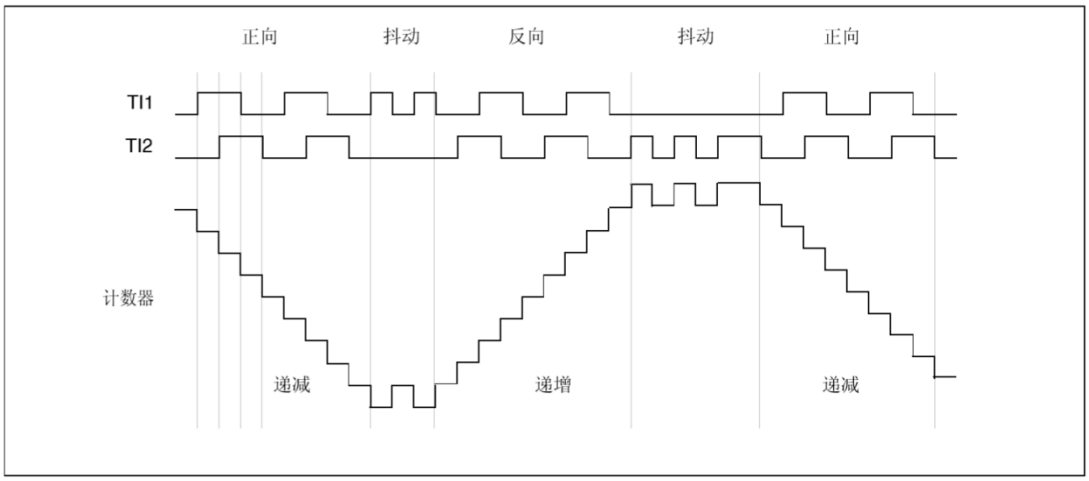


图 154 说明了 TI1FP1 极性反相时的计数器行为示例（除 CC1P=1 外，其它配置与上例相同）。

图 154. TI1FP1 极性反相时的编码器接口模式示例



定时器配置为编码器接口模式时，会提供传感器当前位置的相关信息。使用另一个配置为捕 获模式的定时器测量两个编码器事件之间的周期，可获得动态信息（速度、加速度和减速 度）。指示机械零位的编码器输出即可用于此目的。根据两个事件之间的时间间隔，还可定 期读取计数器。如果可能，可以将计数器值锁存到第三个输入捕获寄存器来实现此目的（捕 获信号必须为周期性信号，可以由另一个定时器产生）；还可以通过由实时时钟生成的 DMA 请求读取计数器值。

16.3.13 定时器输入异或功能

借助 PWMx_CR2 寄存器中的 TI1S 位，可将通道 1 的输入滤波器连接到异或门的输出，从而将 PWMx_CH1 到 PWMx_CH3 这三个输入引脚组合在一起。

异或输出可与触发或输入捕获等所有定时器输入功能配合使用。

16.3.14 定时器与外部触发同步

PWMx 定时器以下列模式与外部触发实现同步：复位模式、门控模式和触发模式。

从模式：复位模式

当触发输入信号发生变化时，计数器及其预分频器可重新初始化。此外，如果 PWMx_CR1 寄存器中的 URS 位处于低电平，则会生成更新事件 UEV。然后，所有预装载寄存器（PWMx_ARR 和 PWMx_CCRx）都将更新。

在以下示例中，TI1 输入上出现上升沿时，递增计数器清零：

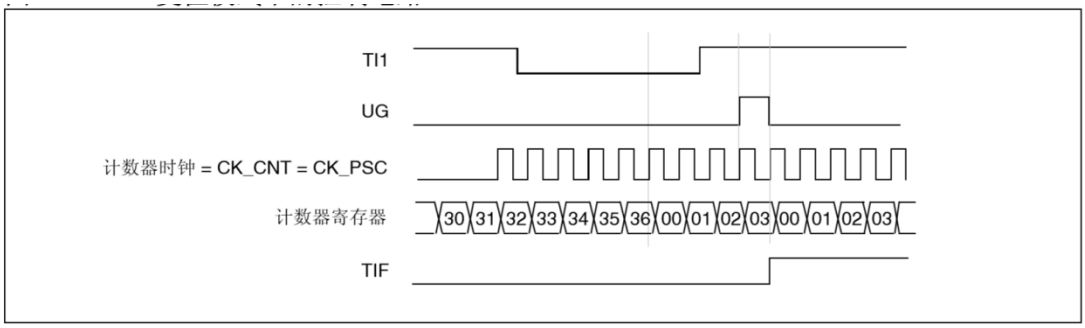
- 将通道 1 配置为检测 TI1 的上升沿。配置输入滤波时间（本例中不需要任何滤波，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配 置。CC1S 位只选择输入捕获源，即 PWMx_CCMR1 寄存器中的 CC1S = 01。在 PWMx_CCER 寄存器中写入 CC1P=0 和 CC1NP=0，验证极性（仅检测上升沿）。
- 在 PWMx_SMCR 寄存器中写入 SMS=100，将定时器配置为复位模式。在 PWMx_SMCR 寄存器中写入 TS=101，选择 TI1 作为输入源。

● 在 PWMx_CR1 寄存器中写入 CEN=1，启动计数器。

计数器使用内部时钟计数，然后正常运转，直到出现 TI1 上升沿。当 TI1 出现上升沿时，计数器清零，然后重新从 0 开始计数。同时，触发标志（PWMx_SR 寄存器中的 TIF 位）置 1 使能中断或 DMA 后，还可发送中断或 DMA 请求（取决于 PWMx_DIER 寄存器中的 TIE 和 TDE 位）。

下图显示了自动重载寄存器 PWMx_ARR=0x36 时的相关行为。TI1 的上升沿与实际计数器复位之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 155. 复位模式下的控制电路



从模式：门控模式

输入信号的电平可用来使能计数器。

在以下示例中，递增计数器仅在 TI1 输入为低电平时计数：

● 将通道 1 配置为检测 TI1 上的低电平。配置输入滤波时间（本例中不需要任何滤波，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。 CC1S 位只选择输入捕获源，即 PWMx_CCMR1 寄存器中的 CC1S=01。在 PWMx_CCER 寄存器中写入 CC1P=1，以确定极性（仅检测低电平）。

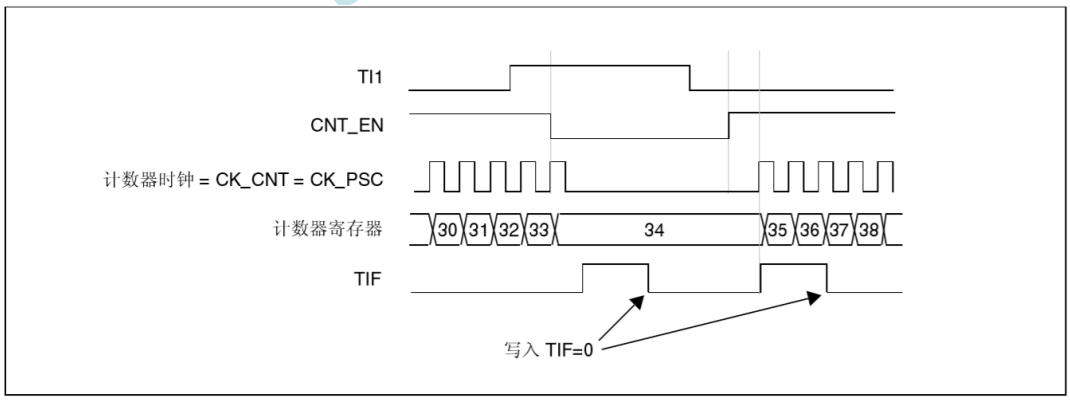
● 在 PWMx_SMCR 寄存器中写入 SMS=101，将定时器配置为门控模式。在 PWMx_SMCR 寄存器中写入 TS=101，选择 TI1 作为输入源。

● 在 PWMx_CR1 寄存器中写入 CEN=1，使能计数器（在门控模式下，如果 CEN=0，则无论触发输入电平如何，计数器都不启动）。

只要 TI1 为低电平，计数器就开始根据内部时钟计数，直到 TI1 变为高电平时停止计数。计数器启动或停止时，PWMx_SR 寄存器中的 TIF 标志都会置 1。

TI1 的上升沿与实际计数器停止之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 156. 门控模式下的控制电路



1. 由于门控模式作用于电平而非边沿，因此在门控模式下，“CCxP=CCxNP=1”（同时检测上升沿和下降 沿）不发挥任何作用。

从模式：触发模式

所选输入上发生某一事件时可以启动计数器。

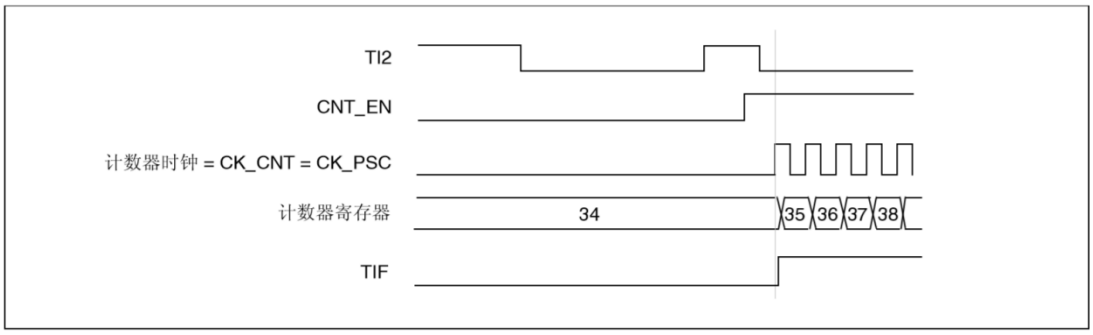
在以下示例中，TI2 输入上出现上升沿时，递增计数器启动：

● 将通道 2 配置为检测 TI2 上的上升沿。配置输入滤波时间（本例中不需要任何滤波，因此保持 IC2F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配 置。 CC2S 位

只选择输入捕获源，即 PWMx_CCMR1 寄存器中的 CC2S=01。在 PWMx_CCER 寄存器中写入 CC2P=1，以确定极性（仅检测低电平）。

- 在 PWMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 PWMx_SMCR 寄存器中写入 TS=110，选择 TI2 作为输入源。

当 TI2 出现上升沿时，计数器开始根据内部时钟计数，并且 TIF 标志置 1。
TI2 的上升沿与实际计数器启动之间的延迟是由于 TI2 输入的重新同步电路引的。
图 157. 触发模式下的控制电路

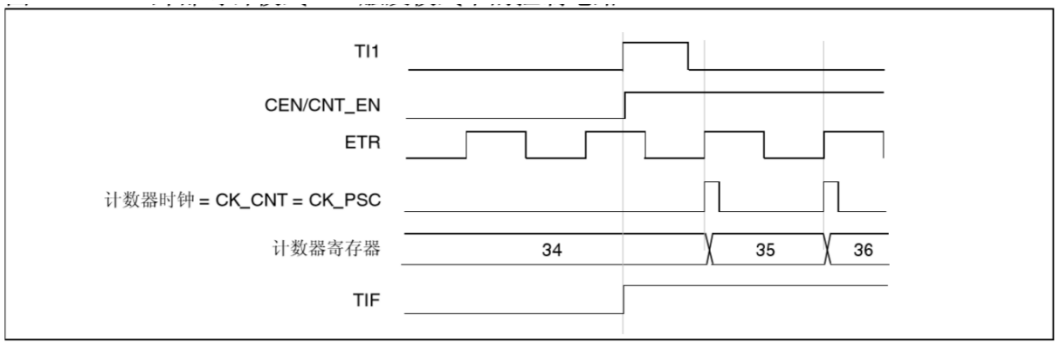


从模式：外部时钟模式 2 + 触发模式

外部时钟模式 2 可与另一种从模式（外部时钟模式 1 和编码器模式除外）结合使用。这种情况下，ETR 信号用作外部时钟输入，在复位模式、门控模式或触发模式下工作时，可选择另一个 输入作为触发输入。不建议通过 PWMx_SMCR 寄存器中的 TS 位来选择 ETR 作为 TRGI。
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递增：

1. 通过对 PWMx_SMCR 寄存器进行如下编程，配置外部触发输入电路：
   — ETF = 0000：无滤波器
   — ETPS = 00：禁止预分频器
   — ETP = 0 ：检测 ETR 的上升沿，并写入 ECE=1，以使能外部时钟模式 2
2. 如下配置通道 1，以检测 TI 的上升沿：
   — IC1F = 0000：无滤波器
   — 由于捕获预分频器不用于触发操作，因此无需对其进行配置。
   —PWMx_CCMR1 寄存器中的 CC1S = 01，只选择输入捕获源
   — PWMx_CCER 寄存器中的 CC1P = 0，以确定极性（仅检测上升沿）。
2. 在 PWMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 PWMx_SMCR 寄存器中写入 TS=101，选择 TI1 作为输入源。

TI1 出现上升沿时将使能计数器并且 TIF 标志置 1。然后计数器在 ETR 出现上升沿时计数。
ETR 信号的上升沿与实际计数器复位之间的延迟是由于 ETRP 输入的重新同步电路引起的。
图 158. 外部时钟模式 2 + 触发模式下的控制电路



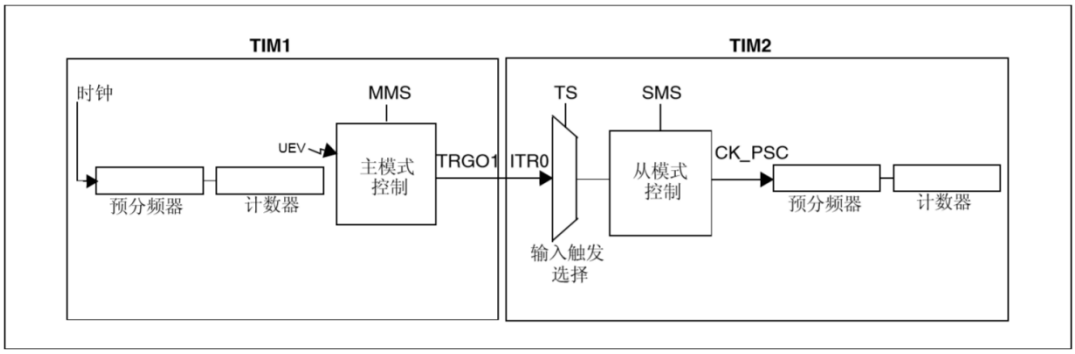
16.3.15 定时器同步

PWMx 定时器从内部连接在一起，以实现定时器同步或级联。当某个定时器配置为主模式时，可对另一个配置为从模式的定时器的计数器执行复位、启动、停止操作或为其提供时钟。

图 159 ：主 /从定时器示例 简要介绍了触发选择和主模式选择框图。

将一个定时器用作另一个定时器的预分频器

图 159. 主/从定时器示例



例如，可以将定时器 1 配置为定时器 2 的预分频器。请参见 图 159。为此：

- 将定时器 1 配置为主模式，以便每次发生更新事件 UEV 时都输出一个周期性触发信号。如果在 PWM1_CR2 寄存器中写入 MMS=010，则每次生成更新事件时，TRGO1 都会输出一个上升沿。
- 要将定时器 1 的 TRGO1 输出连接到定时器 2，必须将定时器 2 配置为从模式，使用 ITR0 作为内部触发。通过 PWM2_SMCR 寄存器中的 TS 位（写入 TS=000）可对此进行 选择。
- 然后将从模式控制器设为外部时钟模式 1（在 PWM2_SMCR 寄存器中写入 SMS=111）。这样一来，定时器 2 的时钟将由定时器 1 周期性触发信号的上升沿（与定时器 1 的计数器上溢对应）提供。
- 后必须通过将这两个定时器的相应 CEN 位（PWMx_CR1 寄存器）置 1 同时使能二者。

注 意 ： 如 果 选 择 定 时 器 1 的 OCx 信号作为触发输出 (MMS=1xx) ，该信号的上升沿将用于驱动定时 器 2 的计数器

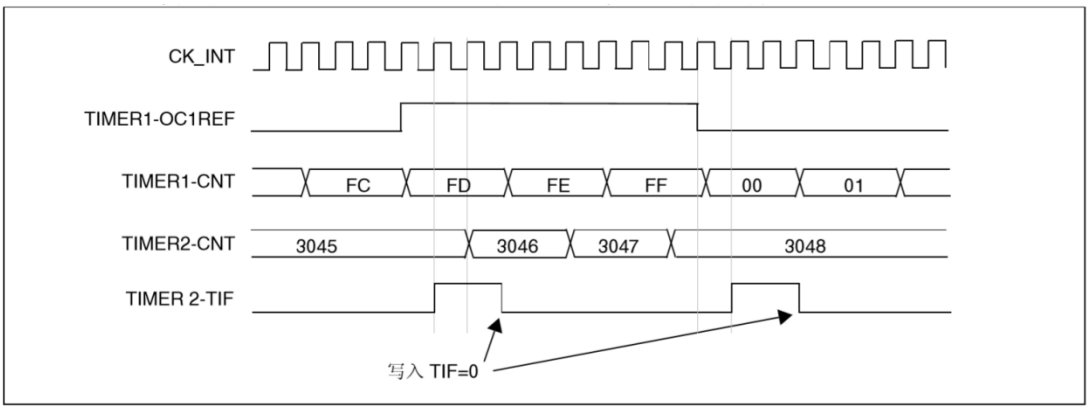
使用一个定时器使能另一个定时器

本例中通过定时器 1 的输出比较 1 来使能定时器 2。相关连接图，请参见 图 159。仅当定时器 1 的 OC1REF 为高电平时，定时器 2 才根据分频后的内部时钟进行计数。两个计数器的时钟频率都基于 CK_INT 通过预分频器执行 3 分频 (fCK_CNT = fCK_INT/3)。

- 将定时器 1 配置为主模式，发送其输出比较 1 参考信号 (OC1REF) 作为触发输出（PWM1_CR2 寄存器中的 MMS=100）。
- 配置定时器 1 的 OC1REF 波形（PWM1_CCMR1 寄存器）。
- 配置定时器 2 以接收来自定时器 1 的输入触发（PWM2_SMCR 寄存器中的 TS=000）。
- 将定时器 2 配置为门控模式（PWM2_SMCR 寄存器中的 SMS=101）。
- 通过向 CEN 位（PWM2_CR1 寄存器）写入“1”使能定时器 2。
- 通过向 CEN 位（PWM1_CR1 寄存器）写入“1”启动定时器 1。

注 意 ： 计 数 器 2 的时钟与计数器 1 不同步，此模式仅影响定时器 2 的计数器使能信号。

图 160. 使用定时器 1 的 OC1REF 对定时器 2 实施门控控制

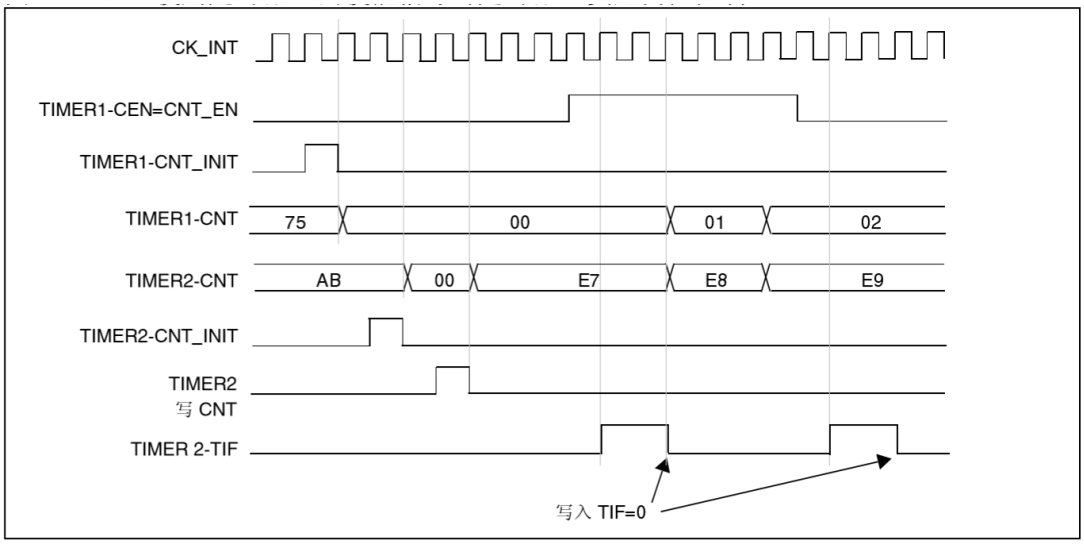


在 图 160 的示例中，定时器 2 的计数器和预分频器在启动前未进行初始化。因此从各自的当前值开始计数。启动定时器 1 之前，通过复位这两个定时器可以从指定值开始计数。这样便可以在定时器计数器中写入所需的任意值。两个定时器都可通过软件使用 PWMx_EGR 寄存器中的 UG 位轻松复位。

在下一示例中，定时器 1 与定时器 2 同步。定时器 1 为主模式，从 0 开始计数。定时器 2 为从模式，从 0xE7 开始计数。两个定时器的预分频比相同。在 PWM1_CR1 寄存器中通过向 CEN 位写入“0”来禁止定时器 1 时，定时器 2 将停止：

- 将定时器 1 配置为主模式，发送其输出比较 1 参考信号 (OC1REF) 作为触发输出 （PWM1_CR2 寄存器中的 MMS=100）。
- 配置定时器 1 的 OC1REF 波形（PWM1_CCMR1 寄存器）。
- 配置定时器 2 以接收来自定时器 1 的输入触发（PWM2_SMCR 寄存器中的 TS=000）。 ● 将定时器 2 配置为门控模式（PWM2_SMCR 寄存器中的 SMS=101）。
- 通过向 UG 位（PWM1_EGR 寄存器）写入“1”复位定时器 1。
- 通过向 UG 位（PWM2_EGR 寄存器）写入“1”复位定时器 2。
- 通过在定时器 2 的计数器 (PWM2_CNTL) 中写入“0xE7”使定时器 2 初始化为 0xE7。
- 通过向 CEN 位（PWM2_CR1 寄存器）写入 “1”使能定时器 2。
- 通过向 CEN 位（PWM1_CR1 寄存器）写入“1”启动定时器 1。
- 通过向 CEN 位（PWM1_CR1 寄存器）写入“0”停止定时器 1。

图 161. 使用定时器 1 的使能信号对定时器 2 实施门控控制

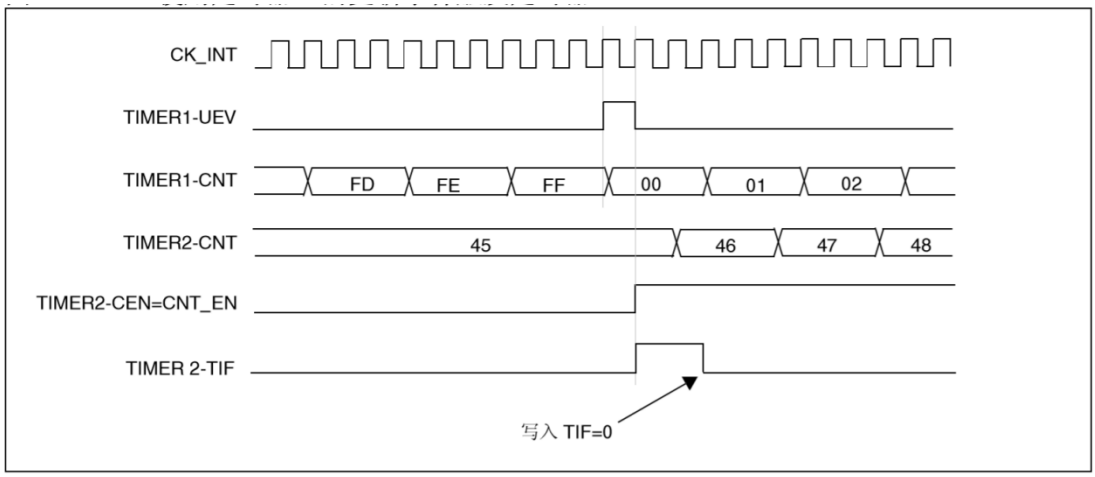


使用一个定时器启动另一个定时器

本例中使用定时器 1 的更新事件使能定时器 2。相关连接图，请参见 图 159。只要定时器 1 生成更新事件，定时器 2 便根据分频后的内部时钟从当前值（可以不为 0）开始计数。定时 器 2 收到触发信号时，其 CEN 位自动置 1，并且计数器开始计数，直到向 PWM2_CR1 寄存器的 CEN 位写入“0”后停止计数。两个计数器的时钟频率都基于 CK_INT 通过预分频器执行 3 分频 (fCK_CNT = fCK_INT/3)。

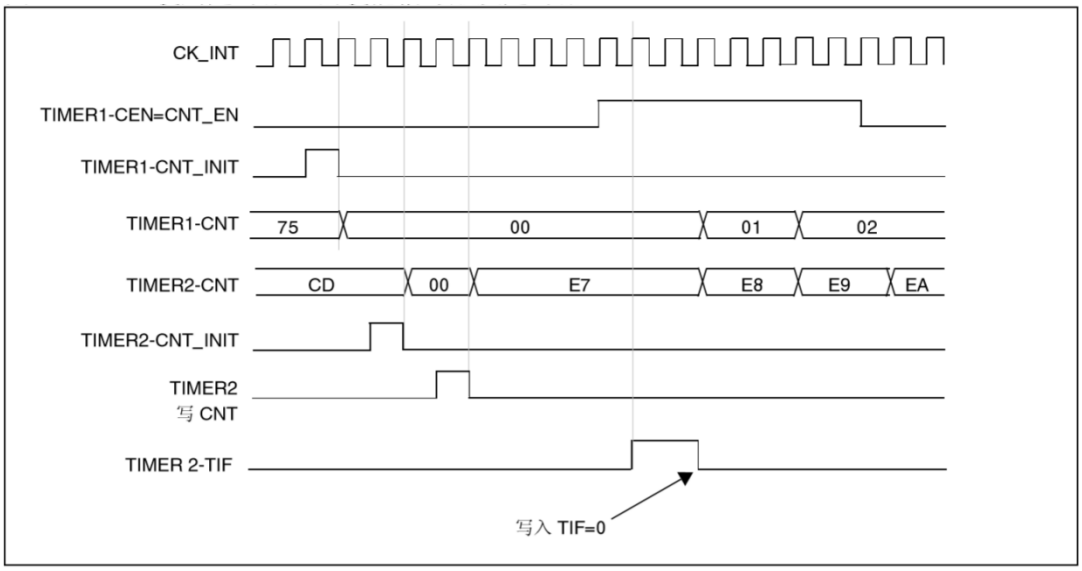
- 将定时器 1 配置为主模式，发送其更新事件 (UEV) 作为触发输出（PWM1_CR2 寄存器 中的 MMS=010）。
- 配置定时器 1 的周期（PWM1_ARR 寄存器）。
- 配置定时器 2 以接收来自定时器 1 的输入触发（PWM2_SMCR 寄存器中的 TS=000）。 ● 将定时器 2 配置为触发模式（PWM2_SMCR 寄存器中的 SMS=110）。
- 通过向 CEN 位（PWM1_CR1 寄存器）写入“1”启动定时器 1。

图 162. 使用定时器 1 的更新事件触发定时器 2



如上述示例所示，用户可以在开始计数之前初始化两个计数器。 图 163 显示了与 图 162 具有 相同配置，只不过处于触发模式（PWM2_SMCR 寄存器中的 SMS=110）而非门控模式的计数行为。

图 163. 使用定时器 1 的使能信号触发定时器 2



将一个定时器用作另一个定时器的预分频器

例如，可以将定时器 1 配置为定时器 2 的预分频器。相关连接图，请参见 图 159。为此：

- 将定时器 1 配置为主模式，发送其更新事件 (UEV) 作为触发输出（PWM1_CR2 寄存器 中的 MMS=010）。这样便会在计数器每次发生上溢时输出一个周期性信号。
- 配置定时器 1 的周期（PWM1_ARR 寄存器）。
- 配置定时器 2 以接收来自定时器 1 的输入触发（PWM2_SMCR 寄存器中的 TS=000）。 ● 将定时器 2 配置为外部时钟模式（PWM2_SMCR 寄存器中的 SMS=111）。
- 通过向 CEN 位（PWM2_CR1 寄存器）写入“1”启动定时器 2。
- 通过向 CEN 位（PWM1_CR1 寄存器）写入“1”启动定时器 1。

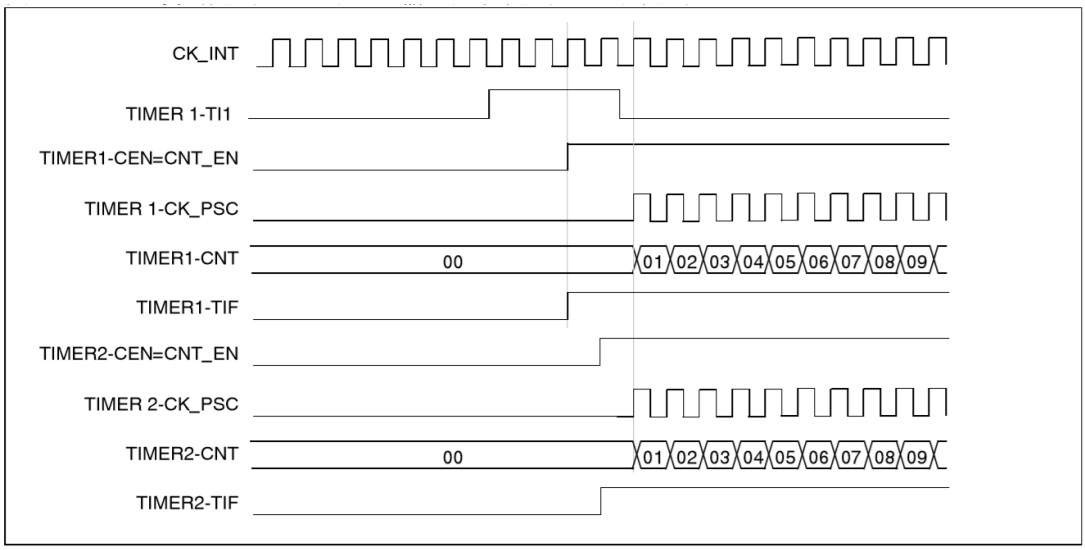
使用一个外部触发同步的启动 2 个定时器

本例中，定时器 1 的 TI1 输入出现上升沿时使能定时器 1，使能定时器 1 的同时使能定时器 2。相关连接图，请参见 图 159。要确保两个计数器对齐，定时器 1 必须配置为主/从模式（对应的 TI1 为从，对应定时器 2 为主）：

- 将定时器 1 配置为主模式，发送其使能信号作为触发输出（PWM1_CR2 寄存器中的 MMS=001）。
- 将定时器 1 配置为从模式以接收来自 TI1 的输入触发（PWM1_SMCR 寄存器中的 TS=100）。
- 将定时器 1 配置为触发模式（PWM1_SMCR 寄存器中的 SMS=110）。
- 通过写入 MSM=1（PWMx_SMCR 寄存器）将定时器 1 配置为主/从模式。
- 配置定时器 2 以接收来自定时器 1 的输入触发（PWM2_SMCR 寄存器中的 TS=000）。 ● 将定时器 2 配置为触发模式（PWM2_SMCR 寄存器中的 SMS=110）。

当 TI1（定时器 1）出现上升沿时，两个计数器开始根据内部时钟同步计数，并且两个 TIF 标志都置 1。*注意：本例中，两个定时器都在启动之前进行了初始化（通过将各自的 UG 位置 1）。两个计数器都从 0 开始计数，但可以通过对任意一个计数器寄存器 (PWMx_CNT) 进行写操作，在二者之间轻松插入一个偏移量。可注意到主 / 从模式在定时器 1 的 CNT_EN 与 CK_PSC 之间产生了延迟。*

图 164. 使用定时器 1 的 TI1 输入触发定时器 1 和定时器 2



16.3.16 调试模式

当微控制器进入调试模式时（Cortex™-MSTAR 内核停止），PWMx 计数器会根据 DBGMCU 模块中的 DBG_PWMx_STOP 配置位选择继续正常工作或者停止工作。

16.4 PWM2 到 PWM5 寄存器

有关寄存器说明中使用的缩写，请参见第 1.1 节 。

32 位外设寄存器必须按字（32 位）写入数据。所有其它外设寄存器则必须按半字（16 位）或字（32 位）写入数据。而读访问可支持字节（8 位）、半字（16 位）或字（32 位）。

16.4.1 PWMx 控制寄存器 1 (PWMx_CR1)

PWMx control register 1

偏移地址：0x00

复位值：0x0000

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

位 15:10 保留，必须保持复位值。

位 9:8 CKD：时钟分频 (Clock division)

此位域指示定时器时钟 (CK_INT) 频率与数字滤波器所使用的采样时钟（ETR、TIx）之间的 分频比，

00：tDTS = tCK_INT

01：tDTS = 2 × tCK_INT

10：tDTS = 4 × tCK_INT

11：保留

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

0：PWMx_ARR 寄存器不进行缓冲

1：PWMx_ARR 寄存器进行缓冲

位 6:5 CMS：中心对齐模式选择 (Center-aligned mode selection)

00：边沿对齐模式。计数器根据方向位 (DIR) 递增计数或递减计数。

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

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

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

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

位 4 DIR：方向 (Direction)

0：计数器递增计数

1：计数器递减计数

注意：当定时器配置为中心对齐模式或编码器模式时，该位为只读状态。

位 3 OPM：单脉冲模式 (One-pulse mode)

0：计数器在发生更新事件时不会停止计数

1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）

位 2 URS：更新请求源 (Update request source)

此位由软件置 1 和清零，用以选择 UEV 事件源。

0：使能时，所有以下事件都会生成更新中断或 DMA 请求。此类事件包括：

— 计数器上溢/下溢

— 将 UG 位置 1

— 通过从模式控制器生成的更新事件

1：使能时，只有计数器上溢/下溢会生成更新中断或 DMA 请求。

位 1 UDIS：更新禁止 (Update disable)

此位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。

0：使能 UEV。更新 (UEV) 事件可通过以下事件之一生成：

— 计数器上溢/下溢

— 将 UG 位置 1

— 通过从模式控制器生成的更新事件

然后缓冲的寄存器将加载预装载值。

1：禁止 UEV。不会生成更新事件，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。但如果将 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。

位 0 CEN：计数器使能 (Counter enable)

0：禁止计数器

1：使能计数器

注意：只有事先通过软件将 CEN 位置 1 ，才可以使用外部时钟、门控模式和编码器模式。而触发模式可通过硬件自动将 CEN 位置 1

在单脉冲模式下，当发生更新事件时会自动将 CEN 位清零。

16.4.2 PWMx 控制寄存器 2 (PWMx_CR2)

PWMx control register 2

偏移地址：0x04

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TI1S	MMS[2:0]			CCOS	Reserved		

	rw	rw	rw	rw	rw	

位 15:8 保留，必须保持复位值。

位 7 TI1S：TI1 选择 (TI1 selection)

0：PWMx_CH1 引脚连接到 TI1 输入

1：PWMx_CH1、PWMx_CH2 和 PWMx_CH3 引脚连接到 TI1 输入（异或组合）

位 6:4 MMS：主模式选择 (Master mode selection)

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

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

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

010： 更新—选择更新事件作为触发输出 (TRGO)。例如，主定时器可用作从定时器的预分 频器。

011： 比较脉冲—一旦发生输入捕获或比较匹配事件，当 CC1IF 被置 1 时（即使已为高电平），触发输出都会发送一个正脉冲 (TRGO)。(TRGO)

100： 比较—OC1REF 信号用作触发输出 (TRGO)

101： 比较—OC2REF 信号用作触发输出 (TRGO)

110： 比较—OC3REF 信号用作触发输出 (TRGO)

111： 比较—OC4REF 信号用作触发输出 (TRGO)

位 3 CCDS：捕获/比较 DMA 选择 (Capture/compare DMA selection)

0：发生 CCx 事件时发送 CCx DMA 请求

1：发生更新事件时发送 CCx DMA 请求

位 2:0 保留，必须保持复位值。

16.4.3 PWMx 从模式控制寄存器 (PWMx_SMCR)

PWMx slave mode control register

偏移地址：0x08

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			Res.	SMS[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw

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

此位可选择将 ETR 还是 ETR 用于触发操作

0：ETR 未反相，高电平或上升沿有效

1：ETR 反相，低电平或下降沿有效

位 14 ECE：外部时钟使能 (External clock enable)

此位可使能外部时钟模式 2。

0：禁止外部时钟模式 2

1：使能外部时钟模式 2。计数器时钟由 ETRF 信号的任意有效边沿提供。

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

2 ： 外部时钟模式 2 可以和以下从模式同时使用： 复位模式 、门控模式和触发模式 。 不过此类 情况下 TRGI 不得连接 ETRF （ TS 位不得为 111 ）。

3 ： 如果同时使能外部时钟模式 1 和外部时钟模式 2 ，则外部时钟输入为 ETRF 。

位 13:12 ETPS：外部触发预分频器 (External trigger prescaler)

外部触发信号 ETRP 频率不得超过 CK_INT 频率的 1/4。可通过使能预分频器来降低 ETRP 频率。这种方法在输入快速外部时钟时非常有用。

00：预分频器关闭

01：2 分频 ETRP 频率

10：4 分频 ETRP 频率

11：8 分频 ETRP 频率

位 11:8 ETF[3:0]：外部触发滤波器 (External trigger filter)

此位域可定义 ETRP 信号的采样频率和适用于 ETRP 的数字滤波器滤波时间。数字滤波器由 事件计数器组成，每 N 个事件才视为一个有效边沿：

0000：无滤波器，按 fDTS 频率进行采样

0001： fSAMPLING=fCK_INT， N=2

0010： fSAMPLING=fCK_INT， N=4

0011： fSAMPLING=fCK_INT， N=8

0100： fSAMPLING=fDTS/2， N=6

0101： fSAMPLING=fDTS/2， N=8

0110： fSAMPLING=fDTS/4， N=6

0111： fSAMPLING=fDTS/4， N=8

1000： fSAMPLING=fDTS/8， N=6 1

001： fSAMPLING=fDTS/8， N=8

1010： fSAMPLING=fDTS/16， N=5

1011： fSAMPLING=fDTS/16， N=6

1100： fSAMPLING=fDTS/16， N=8

1101： fSAMPLING=fDTS/32， N=5

1110： fSAMPLING=fDTS/32， N=6

1111： fSAMPLING=fDTS/32， N=8

位 7 MSM：主/从模式 (Master/Slave mode)

0： 不执行任何操作

1： 当前定时器的触发输入事件（TRGI）的动作被推迟，以使当前定时器与其从定时器实现 完美同步（通过 TRGO）。此设置适用于单个外部事件对多个定时器进行同步的情况。

位 6:4 TS：触发选择 (Trigger selection)

此位域可选择将要用于同步计数器的触发输入。

000：内部触发 0 (ITR0)

001：内部触发 1 (ITR1)。

010：内部触发 2 (ITR2)。

011：内部触发 3 (ITR3)。

100：TI1 边沿检测器 (TI1F_ED)

101：滤波后的定时器输入 1 (TI1FP1)

110：滤波后的定时器输入 2 (TI2FP2)

111：外部触发输入 (ETRF)

有关各定时器 ITRx 含义的详细信息，请参见表 76 ： PWMx 内部触发连接 。

注意：这些位只能在未使用的情况下（例如， SMS=000 时）进行更改，以避免转换时出现错误的边沿检测。

位 3 保留，必须保持复位值。

位 2:0 SMS：从模式选择 (Slave mode selection)

选择外部信号时，触发信号 (TRGI) 的有效边沿与外部输入上所选的极性相关（请参见输入控制寄存器和控制寄存器说明）。

000： 禁止从模式—如果 CEN =“1”，预分频器时钟直接由内部时钟提供。

001： 编码器模式 1—计数器根据 TI1FP1 电平在 TI2FP2 边沿递增/递减计数。

010： 编码器模式 2—计数器根据 TI2FP2 电平在 TI1FP1 边沿递增/递减计数。

011： 编码器模式 3—计数器在 TI1FP1 和 TI2FP2 的边沿计数，计数的方向取决于另外一个信号的电平。

100： 复位模式—在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器并生成一个寄存器更新事件。

101： 门控模式—触发输入 (TRGI) 为高电平时使能计数器时钟。只要触发输入变为低电平，计数器立即停止计数（但不复位）。计数器的启动和停止都是受控的。

110： 触发模式—触发信号 TRGI 出现上升沿时启动计数器（但不复位）。只控制计数器的启动。

111： 外部时钟模式 1—由所选触发信号 (TRGI) 的上升沿提供计数器时钟。

注意：如果将 TI1F_ED 选作触发输入 (TS=100) ，则不得使用门控模式。实际上， TI1F 每次转换时， TI1F_ED 都输出 1 个脉冲，而门控模式检查的则是触发信号的电平。

表 76. PWMx 内部触发连接

从 TIM	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
TIM2	TIM1	TIM8	TIM3	TIM4
TIM3	TIM1	TIM2	TIM5	TIM4
TIM4	TIM1	TIM2	TIM3	TIM8
TIM5	TIM2	TIM3	TIM4	TIM8

16.4.4 PWMx DMA/中断使能寄存器 (PWMx_DIER)

PWMx DMA/Interrupt enable register

偏移地址：0x0C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	Res.	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res.	TIE	Res.	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rw		rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw

位 15 保留，必须保持复位值。

位 14 TDE：触发 DMA 请求使能 (Trigger DMA request enable)

0：禁止触发 DMA 请求。

1：使能触发 DMA 请求。

位 13 保留，始终读为 0

位 12 CC4DE：捕获/比较 4 DMA 请求使能 (Capture/Compare 1 DMA request enable)

0：禁止 CC4 DMA 请求。

1：使能 CC4 DMA 请求。

位 11 CC3DE：捕获/比较 3 DMA 请求使能 (Capture/Compare 1 DMA request enable)

0：禁止 CC3 DMA 请求。

1：使能 CC3 DMA 请求。

位 10 CC2DE：捕获/比较 2 DMA 请求使能 (Capture/Compare 1 DMA request enable)

0：禁止 CC2 DMA 请求。

1：使能 CC2 DMA 请求。

位 9 CC1DE：捕获/比较 1 DMA 请求使能 (Capture/Compare 1 DMA request enable)

0：禁止 CC1 DMA 请求。

1：使能 CC1 DMA 请求。

位 8 UDE：更新 DMA 请求使能 (Update DMA request enable)

0：禁止更新 DMA 请求。

1：使能更新 DMA 请求。

位 7 保留，必须保持复位值。

位 6 TIE：触发信号(TRGI)中断使能 (Trigger interrupt enable)

0：禁止触发信号 (TRGI) 中断。

1：使能触发信号 (TRGI) 中断。

STC MCU

位 5 保留，必须保持复位值。
位 4 CC4IE：捕获/比较 4 中断使能 (Capture/Compare 1 interrupt enable)
0：禁止 CC4 中断。
1：使能 CC4 中断。
位 3 CC3IE：捕获/比较 3 中断使能 (Capture/Compare 1 interrupt enable)
0：禁止 CC3 中断
1：使能 CC3 中断
位 2 CC2IE：捕获/比较 2 中断使能 (Capture/Compare 1 interrupt enable)
0：禁止 CC2 中断
1：使能 CC2 中断
位 1 CC1IE：捕获/比较 1 中断使能 (Capture/Compare 1 interrupt enable)
0：禁止 CC1 中断
1：使能 CC1 中断
位 0 UIE：更新中断使能 (Update interrupt enable)
0：禁止更新中断
1：使能更新中断

16.4.5 PWMx 状态寄存器 (PWMx_SR)

PWMx status register
偏移地址：0x10
复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CC4OF	CC3OF	CC2OF	CC1OF	Reserved		TIF	Reserved	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			rc_w0	rc_w0	rc_w0	rc_w0			rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位 15:13 保留，必须保持复位值。
位 12 CC4OF：捕获/比较 4 重复捕获标志 (Capture/Compare 1 overcapture flag)
请参见 CC1OF 说明
位 11 CC3OF：捕获/比较 3 重复捕获标志 (Capture/Compare 1 overcapture flag)
请参见 CC1OF 说明
位 10 CC2OF：捕获/比较 2 重复捕获标志 (Capture/compare 2 overcapture flag)
请参见 CC1OF 说明
位 9 CC1OF：捕获/比较 1 重复捕获标志 (Capture/Compare 1 overcapture flag)
仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将 该位清零。
0：未检测到重复捕获
1：PWMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。
位 8:7 保留，必须保持复位值。
位 6 TIF：触发中断标志 (Trigger interrupt flag)
在除门控模式以外的所有模式下，当使能从模式控制器后在 TRGI 输入上检测到有效边沿时，该标志将由硬件置 1。选择门控模式时，该标志将在计数器启动或停止时置 1。但需要通过软件清零。
0：未发生触发信号 (TRGI) 事件
1：触发信号 (TRGI) 中断挂起
位 5 保留，必须保持复位值。
位 4 CC4IF：捕获/比较 4 中断标志 (Capture/Compare 4 interrupt flag)
请参见 CC1IF 说明
位 3 CC3IF：捕获/比较 3 中断标志 (Capture/Compare 3 interrupt flag)
请参见 CC1IF 说明

STC MCU

位 2 CC2IF：捕获/比较 2 中断标志 (Capture/Compare 2 interrupt flag)

请参见 CC1IF 说明

位 1 CC1IF：捕获/比较 1 中断标志 (Capture/compare 1 interrupt flag)

如果通道 CC1 配置为输出：

当计数器与比较值匹配时，此标志由硬件置 1，中心对齐模式下除外（请参见 PWMx_CR1 寄存器中的 CMS 位说明）。但需要通过软件清零。

0：不匹配

1：PWMx_CNT 计数器的值与 PWMx_CCR1 寄存器的值匹配。当 PWMx_CCR1 的值大于 PWMx_ARR 的值时，CC1IF 位将在计数器发生上溢（递增计数模式和增减计数模式下）或下溢（递减计数模式下）时变为高电平。

如果通道 CC1 配置为输入：

此位将在发生捕获事件时由硬件置 1。通过软件或读取 PWMx_CCR1 寄存器将该位清零。

0： 未发生输入捕获事件

1： PWMx_CCR1 寄存器中已捕获到计数器值（IC1 上已检测到与所选极性匹配的边沿）

位 0 UIF：更新中断标志 (Update interrupt flag)

- 该位在发生更新事件时通过硬件置 1。但需要通过软件清零。

0：未发生更新。

1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：

- 上溢或下溢（对于 PWM2 到 PWM5）以及当 PWMx_CR1 寄存器中 UDIS = 0 时。
- PWMx_CR1 寄存器中的 URS = 0 且 UDIS = 0，并且由软件使用 PWMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。

PWMx_CR1 寄存器中的 URS=0 且 UDIS=0，并且 CNT 由触发事件重新初始化时（参见同步控制寄存器说明）。

16.4.6 PWMx 事件生成寄存器 (PWMx_EGR)

PWMx event generation register

偏移地址：0x14

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									TG	Res	CC4G	CC3G	CC2G	CC1G	UG
									w		w	w	w	w	w

位 15:7 保留，必须保持复位值。

位 6 TG：产生触发信号 (Trigger generation)

此位由软件置 1 以生成事件，并由硬件自动清零。

0：不执行任何操作

1：PWMx_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。

位 5 保留，必须保持复位值。

位 4 CC4G：捕获/比较 4 生成 (Capture/compare 1 generation)

请参见 CC1G 说明

位 3 CC3G：捕获/比较 3 生成 (Capture/compare 1 generation)

请参见 CC1G 说明

位 2 CC2G：捕获/比较 2 生成 (Capture/compare 1 generation)

请参见 CC1G 说明

位 1 CC1G：捕获/比较 1 生成 (Capture/compare 1 generation)

此位由软件置 1 以生成事件，并由硬件自动清零。

0：不执行任何操作

1：通道 1 上生成捕获/比较事件：

如果通道 CC1 配置为输出：

使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。

如果通道 CC1 配置为输入：

PWMx_CCR1 寄存器中将捕获到计数器当前值。使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。

位 0 UG：更新生成 (Update generation)

该位可通过软件置 1，并由硬件自动清零。

0：不执行任何操作

1：重新初始化计数器并生成寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。如果选择中心对齐模式或 DIR=0（递增计数），计数器将清零；如果 DIR=1（递减计数），计数器将使用自动重载值 (PWMx_ARR)。

16.4.7 PWMx 捕获/比较模式寄存器 1 (PWMx_CCMR1)

PWMx capture/compare mode register 1

偏移地址：0x18

复位值：0x0000

这些通道可用于输入（捕获模式）或输出（比较模式）模式。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其它位在输入模式和输出模式下的功能均不同。对于任一给定 位，OCxx 用于说明通道配置为输出时该位对应的功能，ICxx 则用于说明通道配置为输入时 该位对应的功能。因此，必须注意同一个位在输入阶段和输出阶段具有不同的含义。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式

位 15 OC2CE：输出比较 2 清零使能 (Output compare 3 clear enable)

位 14:12 OC2M[2:0]：输出比较 2 模式 (Output compare 2 mode)

位 11 OC2PE：输出比较 2 预装载使能 (Output compare 2 preload enable)

位 10 OC2FE：输出比较 2 快速使能 (Output compare 2 fast enable)

位 9:8 CC2S[1:0]：捕获/比较 2 选择 (Capture/Compare 2 selection) 此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC2 通道配置为输出

01：CC2 通道配置为输入，IC2 映射到 TI2 上

10：CC2 通道配置为输入，IC2 映射到 TI1 上

11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（PWMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（ PWMx_CCER 中的 CC2E = 0 ），才可向 CC2S 位写入数据。

位 7 OC1CE：输出比较 1 清零使能 (Output compare 3 clear enable)

OC1CE：输出比较 1 清零使能 (Output Compare 1 Clear Enable)

0：OC1Ref 不受 ETRF 输入影响

1：ETRF 输入上检测到高电平时， OC1Ref 立即清零。

位 6:4 OC1M：输出比较 1 模式 (Output compare 1 mode)

这些位定义提供 OC1 和 OC1N 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 和 OC1N 的有效电平则取决于 CC1P 位和 CC1NP 位。

000： 冻结—输出比较寄存器 PWMx_CCR1 与计数器 PWMx_CNT 进行比较不会对输出造成任何影响。（该模式用于生成时基）。

001： 将通道 1 设置为匹配时输出有效电平。当计数器 PWMx_CNT 与捕获/比较寄存器 1 (PWMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。

010： 将通道 1 设置为匹配时输出无效电平。当计数器 PWMx_CNT 与捕获/比较寄存器 1 (PWMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。

011： 翻转—PWMx_CNT=PWMx_CCR1 时，OC1REF 发生翻转。
100： 强制变为无效电平—OC1REF 强制变为低电平。
101： 强制变为有效电平—OC1REF 强制变为高电平。
110： PWM 模式 1—在递增计数模式下，只要 PWMx_CNT<PWMx_CCR1，通道 1 便为有效状态，否则为无效状态。在递减计数模式下，只要 PWMx_CNT>PWMx_CCR1，通道 1 便为 无效状态 (OC1REF=0)，否则为有效状态 (OC1REF=1)。
111： PWM 模式 2—在递增计数模式下，只要 PWMx_CNT<PWMx_CCR1，通道 1 便为无效状态，否则为有效状态。在递减计数模式下，只要 PWMx_CNT>PWMx_CCR1，通道 1 便为 有效状态，否则为无效状态。

注意：在 PWM 模式 1 或 PWM 模式 2 下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时，OCREF 电平才会发生更改。

位 3 OC1PE：输出比较 1 预装载使能 (Output compare 1 preload enable)
0： 禁止与 PWMx_CCR1 相关的预装载寄存器。可随时向 PWMx_CCR1 写入数据，写入后将立即使用新值。
1： 使能与 PWMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。PWMx_CCR1 预装载值在每次生成更新事件时都会装载到活动寄存器中。

注意： 1 ： 只要编程了 LOCK 级别 3 （PWMx_BDTR 寄存器中的 LOCK 位）且 CC1S=00（通道配置为输出），便无法修改这些位。
2 ： 只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式（PWMx_CR1 寄存器中的 OPM 位置 1 ）。其它情况下则无法保证该行为。

位 2 OC1FE：输出比较 1 快速使能 (Output compare 1 fast enable)
此位用于加快触发输入事件对 CC 输出的影响。
0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的短延迟时间为 5 个时钟周期。
1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OCFE 才会起作用。

位 1:0 CC1S：捕获/比较 1 选择 (Capture/Compare 1 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC1 通道配置为输出。
01：CC1 通道配置为输入，IC1 映射到 TI1 上。
10：CC1 通道配置为输入，IC1 映射到 TI2 上。
11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（PWMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（PWMx_CCER 中的 CC1E = 0 ），才可向 CC1S 位写入数据。

输入捕获模式

位 15:12 IC2F：输入捕获 2 滤波器 (Input capture 2 filter)
位 11:10 IC2PSC[1:0]：输入捕获 2 预分频器 (Input capture 2 prescaler)
位 9:8 CC2S：捕获/比较 2 选择 (Capture/compare 2 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC2 通道配置为输出。
01：CC2 通道配置为输入，IC2 映射到 TI2 上。
10：CC2 通道配置为输入，IC2 映射到 TI1 上。 1
1：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（PWMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（PWMx_CCER 中的 CC2E = 0 ），才可向 CC2S 位写入数据。

位 7:4 IC1F：输入捕获 1 滤波器 (Input capture 1 filter)
此位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个事件才视为一个有效边沿：
0000：无滤波器，按 fDTS 频率进行采样 1000：fSAMPLING=fDTS/8，N=6 0001：fSAMPLING=fCK_INT， N=2 1001： fSAMPLING=fDTS/8， N=8 0010：

fSAMPLING=fCK_INT， N=4 1010： fSAMPLING=fDTS/16， N=5 0011：
fSAMPLING=fCK_INT， N=8 1011： fSAMPLING=fDTS/16， N=6 0100：
fSAMPLING=fDTS/2， N=6 1100： fSAMPLING=fDTS/16， N=8 0101：
fSAMPLING=fDTS/2， N=8 1101： fSAMPLING=fDTS/32， N=5 0110：
fSAMPLING=fDTS/4， N=6 1110： fSAMPLING=fDTS/32， N=6 0111：
fSAMPLING=fDTS/4，N=8 1111：fSAMPLING=fDTS/32，N=8

注意：在当前硅版本中，当 ICxF[3:0]= 1 、 2 或 3 时，将用 CK_INT 代替公式中的 fDTS 。

位 3:2 IC1PSC：输入捕获 1 预分频器 (Input capture 1 prescaler)

此位域定义 CC1 输入 (IC1) 的预分频比。

只要 CC1E=0（PWMx_CCER 寄存器），预分频器便立即复位。

00：无预分频器，捕获输入上每检测到一个边沿便执行捕获

01：每发生 2 个事件便执行一次捕获

10：每发生 4 个事件便执行一次捕获

11：每发生 8 个事件便执行一次捕获

位 1:0 CC1S：捕获/比较 1 选择 (Capture/Compare 1 selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出

01：CC1 通道配置为输入，IC1 映射到 TI1 上

10：CC1 通道配置为输入，IC1 映射到 TI2 上

11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（PWMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（ PWMx_CCER 中的 CC1E = 0 ），才可向 CC1S 位写入数据。

16.4.8 PWMx 捕获/比较模式寄存器 2 (PWMx_CCMR2)

PWMx capture/compare mode register 2

偏移地址：0x1C

复位值：0x0000

请参见上述 CCMR1 寄存器说明。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]				IC3PSC[1:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式

位 15 OC4CE：输出比较 4 清零使能 (Output compare 4 clear enable)

位 14:12 OC4M：输出比较 4 模式 (Output compare 4 mode)

位 11 OC4PE：输出比较 4 预装载使能 (Output compare 4 preload enable)

位 10 OC4FE：输出比较 4 快速使能 (Output compare 4 fast enable)

位 9:8 CC4S：捕获/比较 4 选择 (Capture/Compare 4 selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC4 通道配置为输出

01：CC4 通道配置为输入，IC4 映射到 TI4 上

10：CC4 通道配置为输入，IC4 映射到 TI3 上

11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（PWMx_SMCR 寄存 器）选择内部触发输入时有效

注意：仅当通道关闭时（ PWMx_CCER 中的 CC4E = 0 ），才可向 CC4S 位写入数据。

位 7 OC3CE：输出比较 3 清零使能 (Output compare 3 clear enable)

位 6:4 OC3M：输出比较 3 模式 (Output compare 3 mode)

位 3 OC3PE：输出比较 3 预装载使能 (Output compare 3 preload enable)

STC MCU

位 2 OC3FE：输出比较 3 快速使能 (Output compare 3 fast enable)
位 1:0 CC3S：捕获/比较 3 选择 (Capture/Compare 3 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC3 通道配置为输出
01：CC3 通道配置为输入，IC3 映射到 TI3 上
10：CC3 通道配置为输入，IC3 映射到 TI4 上
11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（PWMx_SMCR 寄存 器）选择内部触发输入时有效

注意：仅当通道关闭时（ PWMx_CCER 中的 CC3E = 0 ），才可向 CC3S 位写入数据。

输入捕获模式

位 15:12 IC4F：输入捕获 4 滤波器 (Input capture 4 filter)
位 11:10 IC4PSC：输入捕获 4 预分频器 (Input capture 4 prescaler)
位 9:8 CC4S：捕获/比较 4 选择 (Capture/Compare 4 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC4 通道配置为输出
01：CC4 通道配置为输入，IC4 映射到 TI4 上
10：CC4 通道配置为输入，IC4 映射到 TI3 上
11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（PWMx_SMCR 寄存 器）选择内部触发输入时有效

注意：仅当通道关闭时（ PWMx_CCER 中的 CC4E = 0 ），才可向 CC4S 位写入数据。

位 7:4 IC3F：输入捕获 3 滤波器 (Input capture 3 filter) 位 3:2 IC3PSC：输入捕获 3 预分频器 (Input capture 3 prescaler)
位 1:0 CC3S：捕获/比较 3 选择 (Capture/Compare 3 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC3 通道配置为输出
01：CC3 通道配置为输入，IC3 映射到 TI3 上
10：CC3 通道配置为输入，IC3 映射到 TI4 上
11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（PWMx_SMCR 寄存 器）选择内部触发输入时有效

注意：仅当通道关闭时（ PWMx_CCER 中的 CC3E = 0 ），才可向 CC3S 位写入数据。

16.4.9 PWMx 捕获/比较使能寄存器 (PWMx_CCER)

PWMx capture/compare enable register
偏移地址：0x20
复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	Res.	CC4P	CC4E	CC2NP	Res.	CC3P	CC3E	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
rw		rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw

位 15 CC4NP： 捕获 /比较 4 输出极性 (Capture/Compare 1 output Polarity) 。 请参见 CC1NP 说明
位 14 保留，必须保持复位值。
位 13 CC4P： 捕获 / 比较 4 输出极性 (Capture/Compare 1 output Polarity) 。
请参见 CC1P 说明
位 12 CC4E： 捕获 / 比较 4 输出使能 (Capture/Compare 2 output enable) 。
请参见 CC1E 说明
位 11 CC3NP： 捕获 / 比较 3 输出极性 (Capture/Compare 1 output Polarity) 。 请参见 CC1NP 说明
位 10 保留，必须保持复位值。
位 9 CC3P： 捕获 / 比较 3 输出极性 (Capture/Compare 1 output Polarity) 。

请参见 CC1P 说明

位 8 CC3E： 捕获 /比较 3 输出使能 (Capture/Compare 2 output enable) 。

请参见 CC1E 说明

位 7 CC2NP： 捕获 / 比较 2 输出极性 (Capture/Compare 1 output Polarity) 。

请参见 CC1NP 说明

位 6 保留，必须保持复位值。

位 5 CC2P： 捕获 / 比较 2 输出极性 (Capture/Compare 1 output Polarity) 。

请参见 CC1P 说明

位 4 CC2E： 捕获 / 比较 2 输出使能 (Capture/Compare 2 output enable) 。

请参见 CC1E 说明

位 3 CC1NP： 捕获 / 比较 1 输出极性 (Capture/Compare 1 output Polarity) 。

CC1 通道配置为输出：

在这种情况下， CC1NP 必须保持清零。

CC1 通道配置为输入：

此位与 CC1P 配合使用，用以定义 TI1FP1/TI2FP1 的极性。请参见 CC1P 说明。

位 2 保留，必须保持复位值。

位 1 CC1P： 捕获 / 比较 1 输出极性 (Capture/Compare 1 output Polarity)

CC1 通道配置为输出：

0：OC1 高电平有效

1：OC1 低电平有效

CC1 通道配置为输入：

CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的极性。

00：非反相/上升沿触发 电路对 TIxFP1 上升沿敏感 （在复位模式、外部时钟模式或触发模式下执行捕获或触发操 作）， TIxFP1 未反相 （在门控模式或编码器模式下执行触发操作）。

01：反相/下降沿触发 电路对 TIxFP1 下降沿敏感 （在复位模式、外部时钟模式或触发模式下执行捕获或触发操 作）， TIxFP1 反相 （在门控模式或编码器模式下执行触发操作）。

10：保留，不使用此配置。

11：非反相/上升沿和下降沿均触发 电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或 触发操作）， TIxFP1 未反相（在门控模式或编码器模式下执行触发操作）。编码器模式下不得使用此配置。

位 0 CC1E： 捕获 /

比较 1 输出使能 (Capture/Compare 1 output enable) 。

CC1 通道配置为输出：

0：关闭—OC1 未激活

1：开启—在相应输出引脚上输出 OC1 信号

CC1 通道配置为输入：

此位决定了是否可以实际将计数器值捕获到输入捕获/比较寄存器 1 (PWMx_CCR1) 中。

0：禁止捕获

1：使能捕获

表 77. 标准 OCx 通道的输出控制位

CCxE 位	OCx 输出状态
0	禁止输出（OCx=0、OCx_EN=0）
1	OCx=OCxREF + 极性、OCx_EN=1

注 意 ： 与 标 准 OCx 通道相连的外部 IO 引脚的状态取决于通道 OCx 的状态以及 GPIO 寄存器。

16.4.10 PWMx 计数器 (PWMx_CNT)

PWMx counter

偏移地址：0x24
复位值：0x0000

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 15:0 CNT[15:0]：计数器值 (Counter value)

16.4.11 PWMx 预分频器 (PWMx_PSC)

PWMx prescaler
偏移地址：0x28
复位值：0x0000

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

位 15:0 PSC[15:0]：预分频器值 (Prescaler value)
计数器时钟频率 CK_CNT 等于 fCK_PSC / (PSC[15:0] + 1)。
PSC 包含在每次发生更新事件时要装载到实际预分频器寄存器的值。

16.4.12 PWMx 自动重载寄存器 (PWMx_ARR)

PWMx auto-reload register
偏移地址：0x2C
复位值：0x0000

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 15:0 ARR[15:0]：自动重载值 (Auto-reload value)
ARR 为要装载到实际自动重载寄存器的值。
有关 ARR 更新和行为的更多详细信息，请参见：时基单元 。
当自动重载值为空时，计数器不工作。

16.4.13 PWMx 捕获/比较寄存器 1 (PWMx_CCR1)

PWMx capture/compare register 1
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
CCR1[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 CCR1[31:16]：捕获/比较 1 的高 16 位（对于 PWM2 和 PWM5）。
位 15:0 CCR1[15:0]：捕获/比较 1 的低 16 位 (Low Capture/Compare 1 value)
如果通道 CC1 配置为输出：

CCR1 是捕获/比较寄存器 1 的预装载值。
如果没有通过 PWMx_CCMR 寄存器中的 OC1PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄 存器 1）。
实际捕获/比较寄存器中包含要与计数器 PWMx_CNT 进行比较并在 OC1 输出上发出信号的值。
如果通道 CC1 配置为输入：
CCR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。

16.4.14 PWMx 捕获/比较寄存器 2 (PWMx_CCR2)

PWMx capture/compare register 2
偏移地址：0x38
复位值：0x0000 0000

位 31:16 CCR2[31:16]：捕获/比较 2 的高 16 位（对于 PWM2 和 PWM5）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR2[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 CCR2[15:0]：捕获/比较 2 的低 16 位 (Low Capture/Compare 2 value)
如果通道 CC2 配置为输出：
CCR2 为要装载到实际捕获/比较 2 寄存器的值（预装载值）。
如果没有通过 PWMx_CCMR 寄存器中的 OC2PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 2 寄存器中。
实际捕获/比较寄存器中包含要与计数器 PWMx_CNT 进行比较并在 OC2 输出上发出信号的值。
如果通道 CC2 配置为输入：
CCR2 为上一个输入捕获 2 事件 (IC2) 发生时的计数器值。

16.4.15 PWMx 捕获/比较寄存器 3 (PWMx_CCR3)

PWMx capture/compare register 3
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
CCR3[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 CCR3[31:16]：捕获/比较 3 的高 16 位（对于 PWM2 和 PWM5）。
位 15:0 CCR3[15:0]：捕获/比较 3 的低 16 位 (Low Capture/Compare value)
如果通道 CC3 配置为输出：
CCR3 为要装载到实际捕获/比较 3 寄存器的值（预装载值）。
如果没有通过 PWMx_CCMR 寄存器中的 OC3PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 3 寄存 器中。

实际捕获/比较寄存器中包含要与计数器 PWMx_CNT 进行比较并在 OC3 输出上发出信号的值。
如果通道 CC3 配置为输入：
CCR3 为上一个输入捕获 3 事件 (IC3) 发生时的计数器值。

16.4.16 PWMx 捕获/比较寄存器 4 (PWMx_CCR4)

PWMx capture/compare register 4
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
CCR4[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 CCR4[31:16]：捕获/比较 4 的高 16 位（对于 PWM2 和 PWM5）。
位 15:0 CCR4[15:0]：捕获/比较 4 的低 16 位 (Low Capture/Compare value) 1.
如果 CC4 通道配置为输出（CC4S 位）：
CCR4 为要装载到实际捕获/比较 4 寄存器的值（预装载值）。
如果没有通过 PWMx_CCMR 寄存器中的 OC4PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 4 寄存器中。
实际捕获/比较寄存器中包含要与计数器 PWMx_CNT 进行比较并在 OC4 输出上发出信号的值。
如果 CC4 通道配置为输入（PWMx_CCMR4 寄存器中的 CC4S 位）： CCR4 为上一个输入捕获 4 事件 (IC4) 发生时的计数器值

16.4.17 PWMx DMA 控制寄存器 (PWMx_DCR)

PWMx DMA control register
偏移地址：0x48
复位值：0x0000

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

位 15:13 保留，必须保持复位值。
位 12:8 DBL[4:0]：DMA 连续传送长度 (DMA burst length)
该 5 位定义了 DMA 在连续模式下的传送长度（当对 PWMx_DMAR 寄存器进行读或写时，定时器则进行一次连续传送）。
00000：1 次传输，
00001：2 次传输，
00010：3 次传输，
...
10001：18 次传输。
位 7:5 保留，必须保持复位值。
位 4:0 DBA[4:0]：DMA 基址 (DMA base address)
该 5 位向量定义 DMA 传输的基址（通过 PWMx_DMAR 地址进行读/写访问时）。DBA 定义 为从 PWMx_CR1 寄存器地址开始计算的偏移量。
示例：
00000：PWMx_CR1，
00001：PWMx_CR2，

00010：PWMx_SMCR，

...

示例：考虑以下传输：DBL = 7 次传输，DBA = PWMx_CR1。这种情况下将向/从自 PWMx_CR1 地址开始的 7 个寄存器传输数据。

16.4.18 PWMx 全传输 DMA 地址 (PWMx_DMAR)

PWMx DMA address for full transfer

偏移地址：0x4C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB4[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 DMAB[15:0]：DMA 连续传送寄存器 (DMA register for burst accesses)

对 DMAR 寄存器执行读或写操作将访问位于如下地址的寄存器

(PWMx_CR1 地址) + (DBA + DMA 索引) x 4

其中 PWMx_CR1 地址为控制寄存器 1 的地址，DBA 为 PWMx_DCR 寄存器中配置的 DMA 基址，DMA 索引由 DMA 传输自动控制，其范围介于 0 到 DBL（PWMx_DCR 寄存器中配置的 DBL）之间。

DMA 连续传送功能使用方法示例

本例中，定时器 DMA 连续传送功能用于将 CCRx 寄存器（x = 2、3、4）的内容更新为通过 DMA 传输到 CCRx 寄存器中的多个半字。

具体操作步骤如下：

1. 将相应的 DMA 通道配置如下：
   — DMA 通道外设地址为 DMAR 寄存器地址
   —DMA 通道存储器地址为包含要通过 DMA 传输到 CCRx 寄存器的数据的 RAM 缓冲区地址。
   — 要传输的数据量 = 3（参见下文注释）。
   — 禁止循环模式。
2. 通过将 DBA 和 DBL 位域配置如下来配置 DCR 寄存器： DBL = 3 次传输，DBA = 0xE。
3. 使能 PWMx 更新 DMA 请求（DIER 寄存器中的 UDE 位置 1）。
4. 使能 PWMx
5. 使能 DMA 通道

注意： 本例适用于每个 CCRx 寄存器只更新一次的情况。如果每个 CCRx 寄存器要更新两次，则要传输的数据量应为 6 。下面以包含 data1 、 data2 、 data3 、 data4 、 data5 和 data6 的 RAM 缓冲区为例。数据将按照如下方式传输到 CCRx 寄存器：在第一个更新 DMA 请求期间， data1 传输到 CCR2 ， data2 传输到 CCR3 ， data3 传输到 CCR4 ；在第二个更新 DMA 请求期间， data4 传输到 CCR2 ， data5 传输到 CCR3 ， data6 传输到 CCR4 。

16.4.19 PWM2 选项寄存器 (PWM2_OR)

PWM2 option register

偏移地址：0x50

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				ITR1_RMP		Reserved									
				rw	rw										

位 15:12 保留，必须保持复位值。

位 11:10 ITR1_RMP：内部触发 1 重映射 (Internal trigger 1 remap)

由软件置 1 和清零。

00：PWM8_TRGOUT
01：PTP 触发输出连接到 PWM2_ITR1
10：OTG FS SOF 连接到 PWM2_ITR1 输入
11：OTG HS SOF 连接到 PWM2_ITR1 输入

位 9:0 保留，必须保持复位值。

16.4.20 PWM5 选项寄存器 (PWM5_OR)

PWM5 option register
偏移地址：0x50
复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TI4_RMP [1:0]		Reserved					
								rw	rw						

位 15:8 保留，必须保持复位值。
位 7:6 TI4_RMP：定时器输入 4 重映射 (TIMER Input 4 remap)
00：PWM5 的通道 4 连接到 GPIO：请参见数据手册中的复用功能映射表。
01：为进行校准，LSI 内部时钟连接到 PWM5_CH4 输入。
10：为进行校准，LSE 内部时钟连接到 PWM5_CH4 输入。
11：为进行校准，RTC 唤醒中断连接到 PWM5_CH4 输入。应使能唤醒中断。由软件置 1 和清零。
位 5:0 保留，必须保持复位值。

16.4.21 PWMx 寄存器映射

PWMx 寄存器按照下表所述方式映射：
表 78. PWM2 到 PWM5 寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0X00	TIMx_CR1	Reserved																						CKD [1:0]		ARPE	CMS [1:0]		DIR	OPM	URS	UDIS	CEN
	Reset value																							0	0	0	0	0	0	0	0	0	0
0X04	TIMx_CR2	Reserved																								TI1S	MMS[2:0]			Reserved			
	Reset value																									0	0	0	0				
0X08	TIMx_SMCR	Reserved																ETP	ECE	ETPS [3:0]		ETF[3:0]				MSM	TS[2:0]			Reserved	SMS[2:0]		
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0
0X0C	TIMx_DIER	Reserved																	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Reserved	TIE	Reserved	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	Reset value																		0	0	0	0	0	0	0		0		0	0	0	0	0
0X10	TIMx_SR																				CC4OF	CC3OF	CC2OF	CC1OF	Reserved		TIF	Reserved	CC4IF	CC3IF	CC2IF	CC1IF	UIF

Offset	Register	31–16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reset value					0	0	0	0			0		0	0	0	0	0
0X14	TIMx_EGR	Reserved										TG	Reserved	CC4G	CC3G	CC2G	CC1G	UG
	Reset value											0		0	0	0	0	0
0X18	TIMx_CCMR1 Output compare mode	Reserved	OC2CE	OC2M [2:-0]			OC2PE	OC2FE	CC2S [1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIMx_CCMR1 INput compare mode	Reserved	IC2F[3:0]				IC2 PSC [1:0]		CC2S [1:0		CC2S[1:0]				IC1 PSC [1:0]		CC1S [1:0]	
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0X1C	TIMx_CCMR2 Output compare mode	Reserved	OC4CE	OC4M [2:-0]			OC4PE	OC4FE	CC4S [1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S [1:0]	
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIMx_CCMR2 INtput compare mode	Reserved	IC4F[3:0]				IC4 PSC [1:0]		CC4S [1:0		CC3F[1:0]				IC3PSC [1:0]		CC3S [1:0]	
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0X20	TIMx_CCER	Reserved			CC4P	Reserved	CC3NP	Reserved	CC3P	CC3E	CC2NP	Reserved	CC2P	CC2E	CC1NP	Reserved	CC1P	CC1E
	Reset value				0		0		0	0	0		0	0	0		0	0
0X24	TIMx_CNT	CNT[31:16] TIM2 and TIM5 only,reserved on the other timers	CNT[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0X28	TIMx_PSC	Reserved	PSC[15:0]															
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0X2C	TIMx_ARR	ARR[31:16] TIM2 and TIM5 only,reserved on the other timers	ARR15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	Reserved																	
0x34	TIMx_CCR1	Reserved	CCR1[15:0]															
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	TIMx_CCR2	Reserved	CCR2[15:0]															
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	TIMx_CCR3	Reserved	CCR3[15:0]															
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	TIMx_CCR4	Reserved	CCR4[15:0]															
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	Reserved																	
0x48	TIMX_DCR	Reserved				DBL[4:0]								DBA[]4:0				
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0	0
0x4C	TIMX_DMAR	Reserved	DMAB[15:0]															
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0X50	PWM2_OR	Reserved					ITR1_RMP	Reserved	
	Reset value						0	0	
0x50	PWM5_OR	Reserved					TI4_RMP	Reserved	
	Reset value						0	0	

STC MCU

17 通用定时器（PWM9 到 PWM14）

17.1 PWM9 到 PWM14 简介

PWM9 到 PWM14 通用定时器包含一个 16 位自动重载计数器，该计数器由可编程预分频器驱动。它们可用于多种用途，包括测量输入信号的脉冲宽度（输入捕获），或者生成输出波形（输 出比较、PWM）。

使用定时器预分频器和 RCC 时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。PWM9 到 PWM14 定时器彼此完全独立，不共享任何资源。如 第 16.4.12 节 中所述，它们可以同步操作。

17.2 PWM9 到 PWM14 主要特性

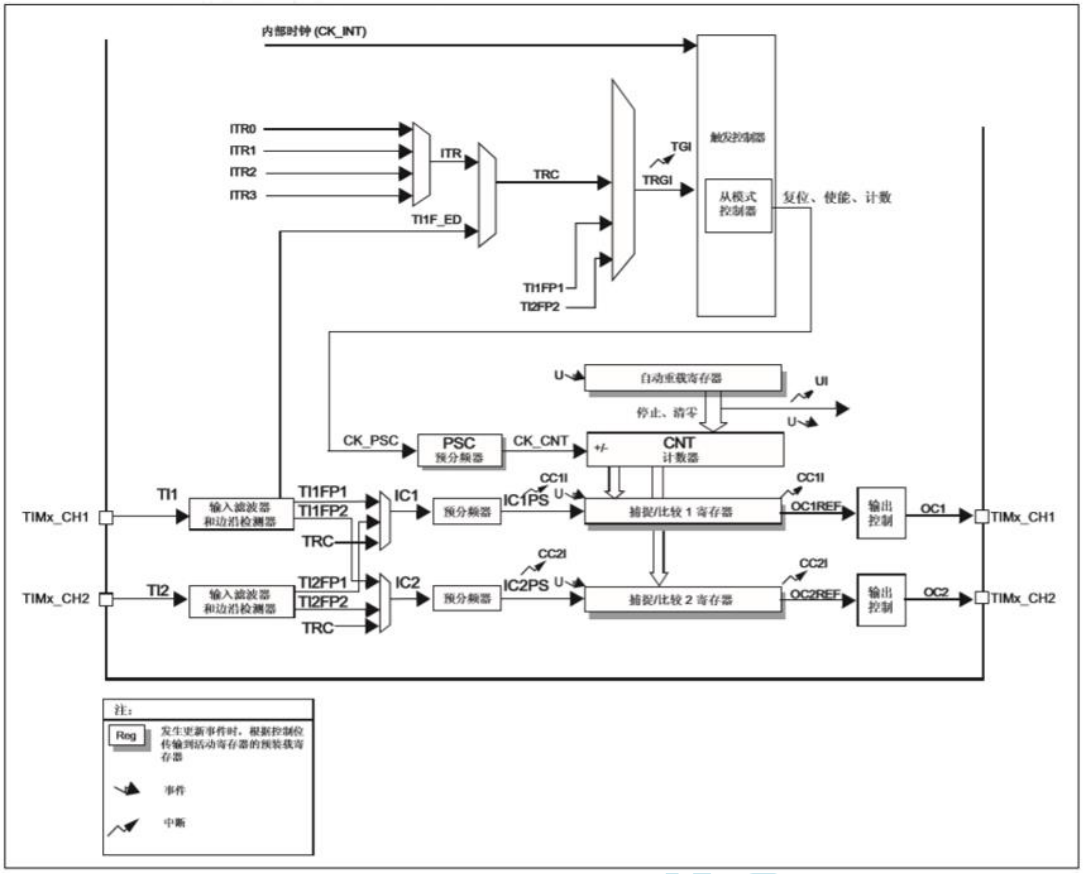
17.2.1 PWM9/PWM12 主要特性

PWM9 到 PWM14 通用定时器具有以下特性：

- 16 位自动重载递增计数器（属于中等容量器件）
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（即运行时修改），分频系数 介于 1 和 65536 之间
- 多达 2 个独立通道，可用于：
  - 输入捕获
  - 输出比较
  - PWM 生成（边沿对齐模式）
  - 单脉冲模式输出
- 使用外部信号控制定时器且可实现多个定时器互连的同步电路
- 发生如下事件时生成中断：
  - 更新：计数器上溢、计数器初始化（通过软件或内部触发）
  - 触发事件（计数器启动、停止、初始化或者由内部触发计数）
  - 输入捕获
  - 输出比较

图 165. 通用定时器框图（PWM9 和 PWM12）

STCMCU



17.3PWM10/PWM11 和 PWM13/PWM14 主要特性

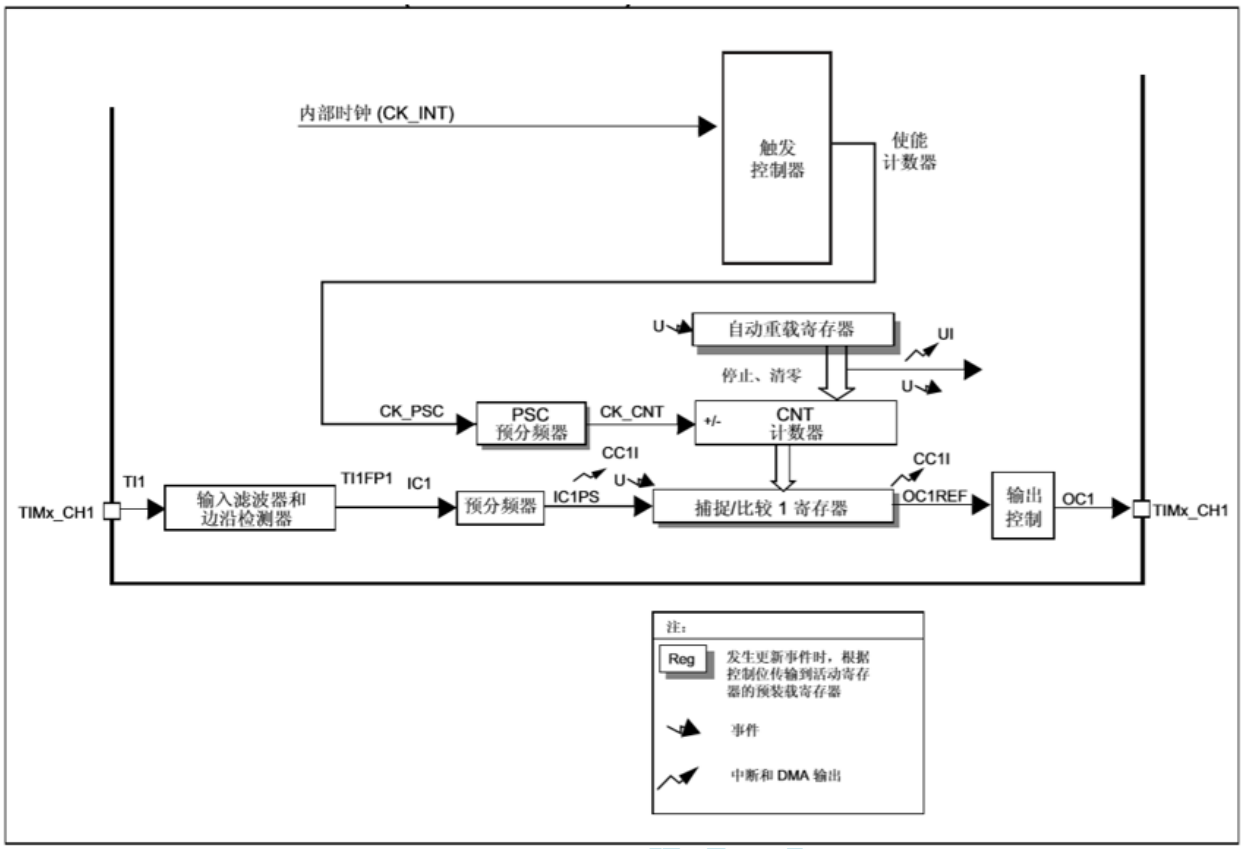
通用定时器 PWM10/PWM11 和 PWM13/PWM14 具有以下特性：

● 16 位自动重载递增计数器

● 16 位可编程预分频器，用于对计数器时钟频率进行分频（即运行时修改），分频系数 介于 1 和 65536 之间

- 独立通道，可用于：
  - 输入捕获
  - 输出比较
  - PWM 生成（边沿对齐模式）
  - 单脉冲模式输出
- 发生如下事件时生成中断：
  - 更新：计数器上溢、计数器初始化（通过软件）
  - 输入捕获
  - 输出比较

图 166. 通用定时器框图 (PWM10/11/13/14)



17.4PWM9 到 PWM14 功能说明

17.4.1时基单元

定时器的主要模块由一个 16 位计数器及其相关的自动重载寄存器组成。计数器采用递增方式计数。

计数器的时钟可通过预分频器进行分频。

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

时基单元包括：

- 计数器寄存器 (PWMx_CNT)
- 预分频器寄存器 (PWMx_PSC)
- 自动重载寄存器 (PWMx_ARR)

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存 器。预装载寄存器的内容既可以直接传送到影子寄存器，也可以在每次发生更新事件 (UEV) 时传送到影子寄存器，这取决于 PWMx_CR1 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值并且 PWMx_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生进行详细介绍。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 PWMx_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器（有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明）。注意，计数器将在 PWMx_CR1 寄存器的 CEN 位置 1 时刻的一个时钟周期后开始计数。

预分频器说明

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

图 167 和 图 168 以一些示例说明在预分频比实时变化时计数器的行为：

图 167. 预分频器分频由 1 变为 2 时的计数器时序图

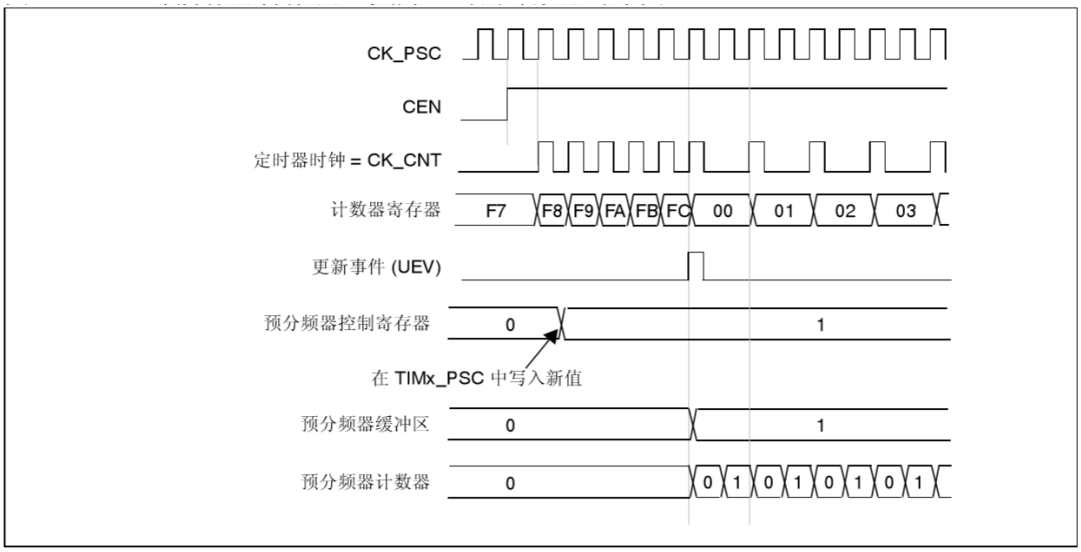
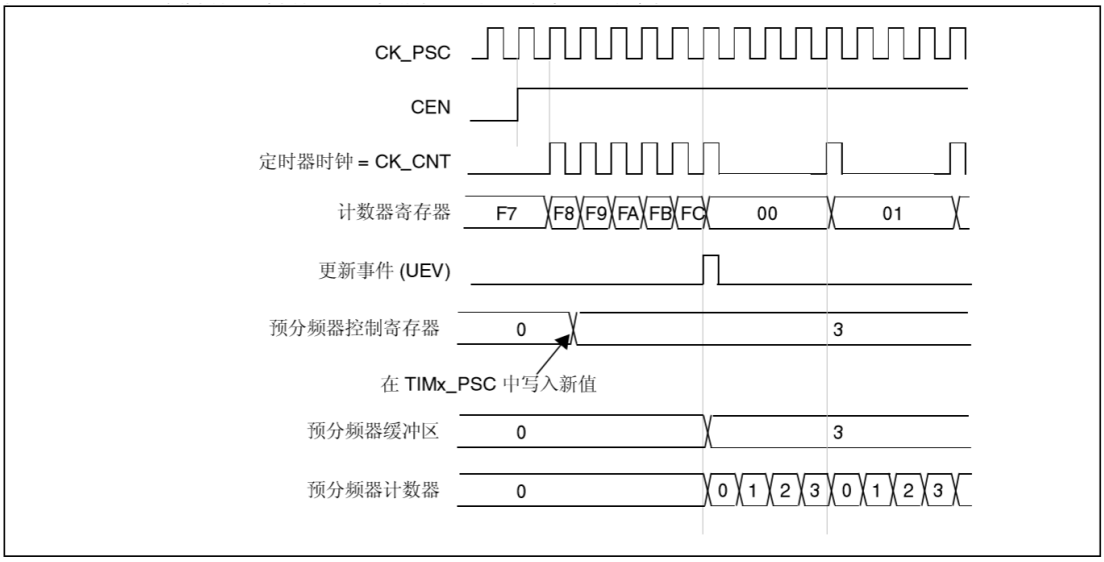


图 168. 预分频器分频由 1 变为 4 时的计数器时序图



17.4.2计数器模式

递增计数模式

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

将 PWMx_EGR 寄存器中的 UG 位置 1（通过软件或使用 PWM9 和 PWM12 上的从模式控制器）也会生成更新事件。

通过软件将 PWMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器 和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 PWMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断）。这样一来，如果在发生捕获事件时将计数 器清零，将不会同时产生更新中断和捕获中断。

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

- 自动重载影子寄存器将以预装载值 (PWMx_ARR) 进行更新
- 预分频器的缓冲区中将重新装载预装载值（PWMx_PSC 寄存器的内容） 以下各图以一些示例说明当 PWMx_ARR=0x36 时不同时钟频率下计数器的行为。

图 169. 计数器时序图，1 分频内部时钟

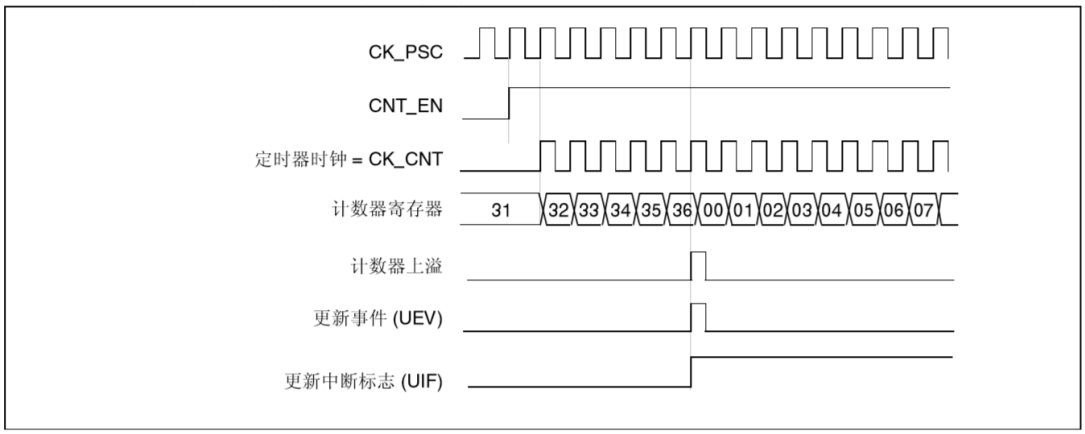


图 170. 计数器时序图，2 分频内部时钟

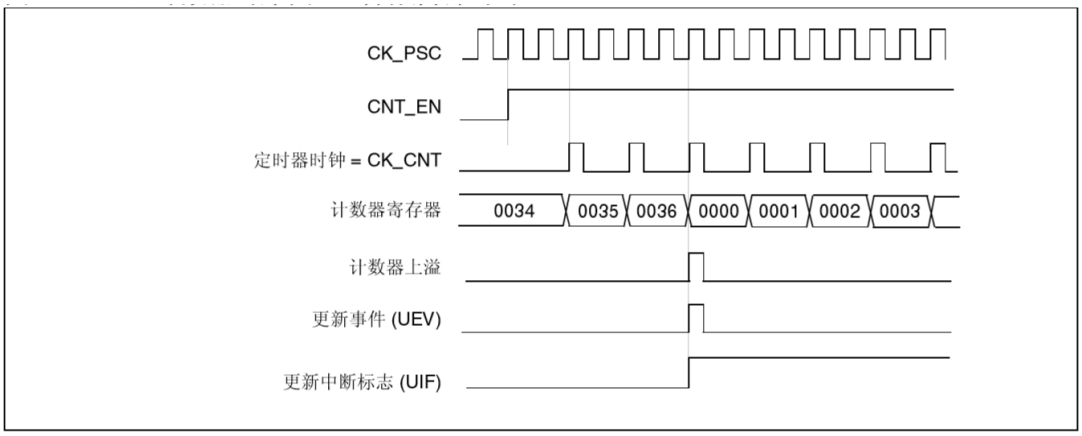


图 171. 计数器时序图，4 分频内部时钟

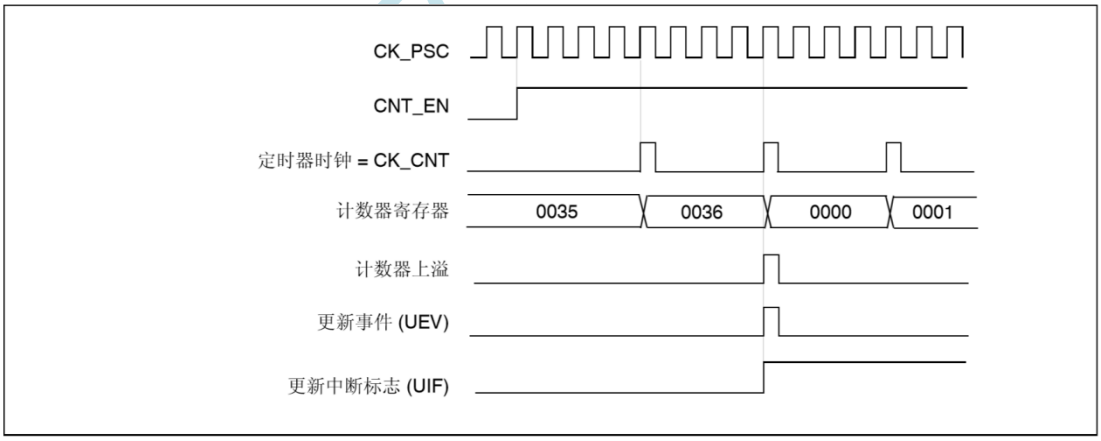


图 172. 计数器时序图，N 分频内部时钟

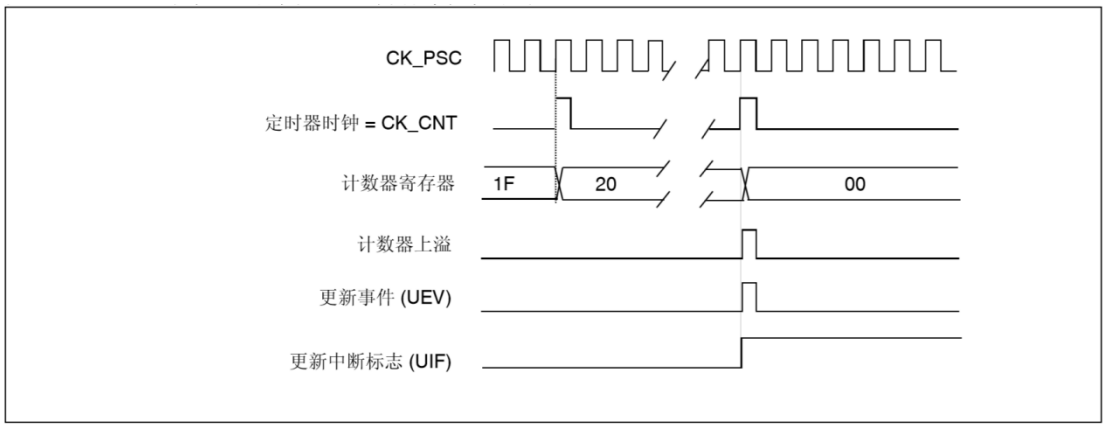


图 173. 计数器时序图，ARPE=0 时更新事件（PWMx_ARR 未预装载）

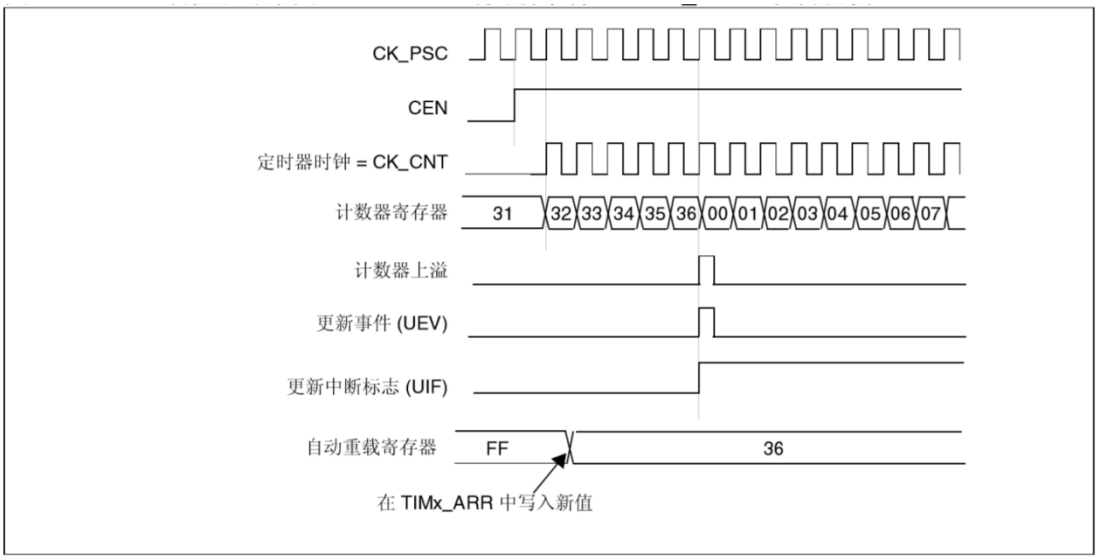
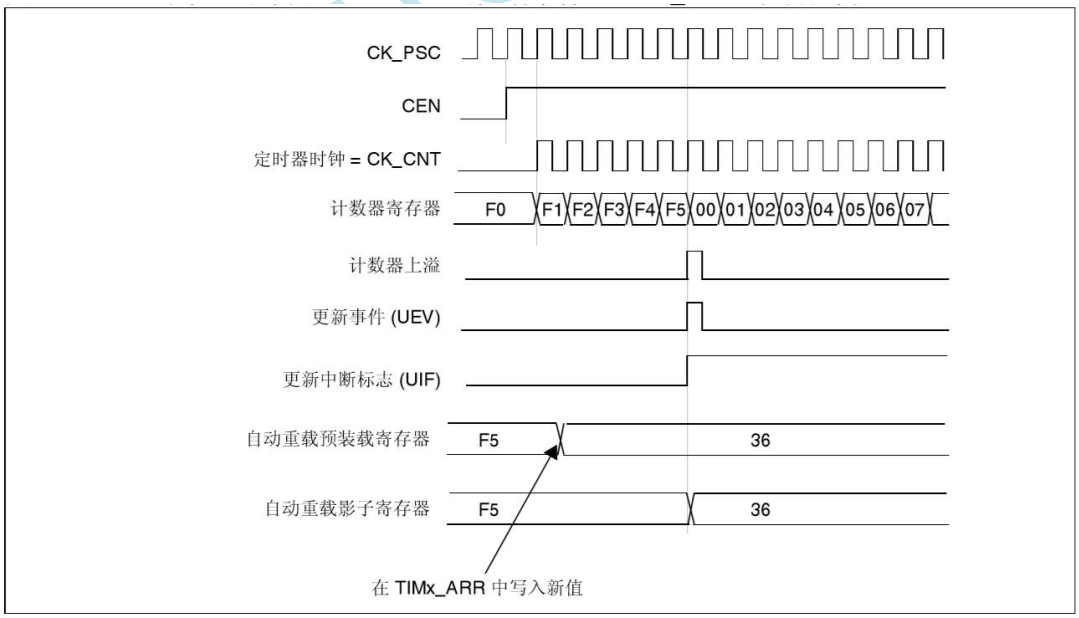


图 174. 计数器时序图，ARPE=1 时更新事件（PWMx_ARR 未预装载）



17.4.3时钟选择

计数器时钟可由下列时钟源提供：

- 内部时钟 (CK_INT)
- 外部时钟模式 1（针对 PWM9 和 PWM12）：外部输入引脚 (TIx)

- 内部触发输入 (ITRx)（针对 PWM9 和 PWM12）：连接来自其它计数器的触发输出。更多详细信息，请参见 将一个定时器用作另一个定时器的预分频器 一节

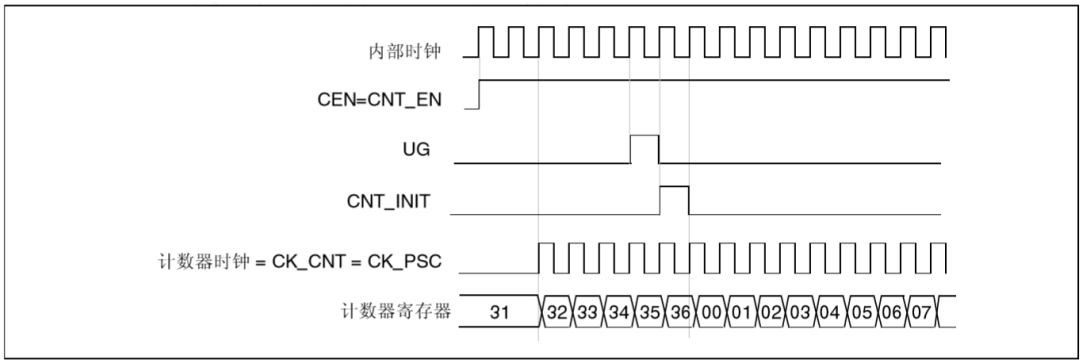
内部时钟源 (CK_INT)

内部时钟源是 PWM10/PWM11 和 PWM13/PWM14 的默认时钟源。

对于 PWM9 和 PWM12，在禁止从模式控制器后选择内部时钟源（SMS="000"）。然后，将 PWMx_CR1 寄存器中的 CEN 位和 PWMx_EGR 寄存器中的 UG 位用作控制位，并且只能通过软件对其进行更改（保持清零状态的 UG 除外）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 CK_INT 提供。

图 175 显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）。

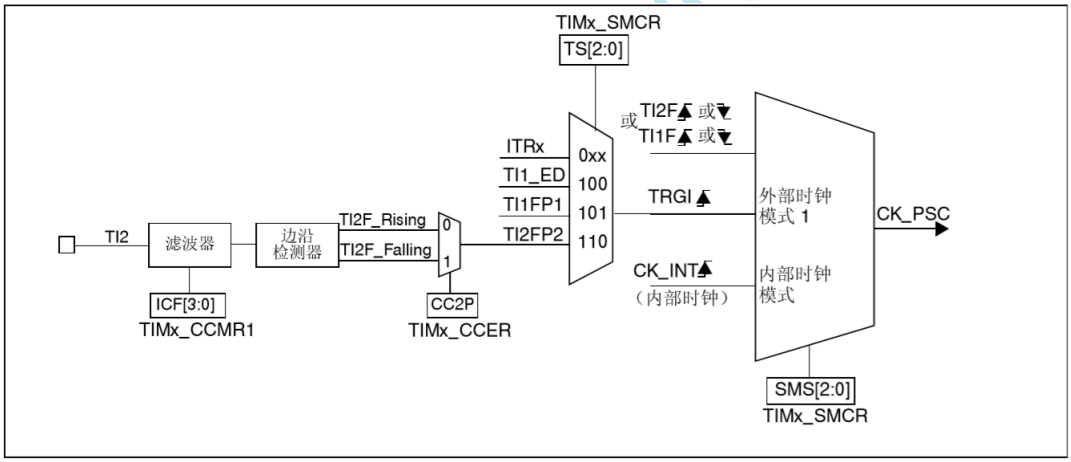
图 175. 正常模式下的控制电路，1 分频内部时钟



外部时钟源模式 1（PWM9 和 PWM12）

当 PWMx_SMCR 寄存器中的 SMS="111"时，可选择此模式。计数器可在选定的输入信号上出现上升沿或下降沿时计数。

图 176. TI2 外部时钟连接示例



例如，要使递增计数器在 TI2 输入出现上升沿时计数，请执行以下步骤：

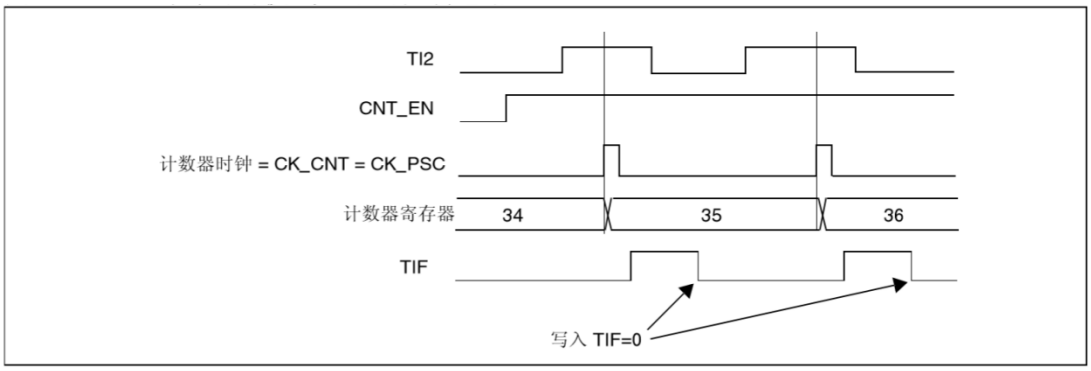
1. 通过在 PWMx_CCMR1 寄存器中写入 CC2S ="01"来配置通道 2，使其能够检测 TI2 输 入的上升沿。
2. 通过在 PWMx_CCMR1 寄存器中写入 IC2F[3:0] 位来配置输入滤波时间（如果不需要任何 滤波器，请保持 IC2F="0000"）。
3. 通过在 PWMx_CCER 寄存器中写入 CC2P="0"和 CC2NP="0"来选择上升沿极性。
4. 通过在 PWMx_SMCR 寄存器中写入 SMS="111"，使定时器在外部时钟模式 1 下工作。
5. 通过在 PWMx_SMCR 寄存器中写入 TS="110"来选择 TI2 作为触发输入源。
6. 通过在 PWMx_CR1 寄存器中写入 CEN="1"来使能计数器。

注意：由于捕获预分频器不用于触发操作，因此无需对其进行配置。

当 TI2 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。

TI2 的上升沿与实际计数器时钟之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 177. 外部时钟模式 1 下的控制电路



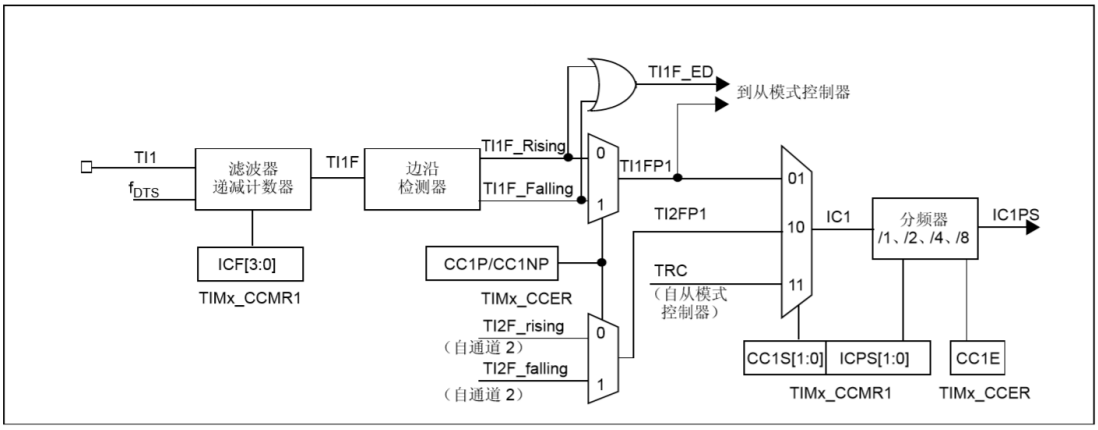
17.4.4捕获/比较通道

每个捕获/比较通道均围绕一个捕获/比较寄存器（包括一个影子寄存器）、一个捕获输入阶 段（数字滤波、多路复用和预分频器）和一个输出阶段（比较器和输出控制）构建而成。

图 178 到 图 180 对捕获/比较通道做了简要说明。

输入阶段对相应的 TIx 输入进行采样，生成一个滤波后的信号 TIxF。然后，带有极性选择功能的边沿检测器生成一个信号 (TIxFPx)，该信号可用作从模式控制器的触发输入，也可用作捕 获命令。该信号先进行预分频 (ICxPS)，而后再进入捕获寄存器。

图 178. 捕获/比较通道（例如：通道 1 输入阶段）



输出阶段生成一个中间波形作为基准：OCxRef（高电平有效）。链的末端决定终输出信号 的极性。

图 179. 捕获/比较通道 1 主电路

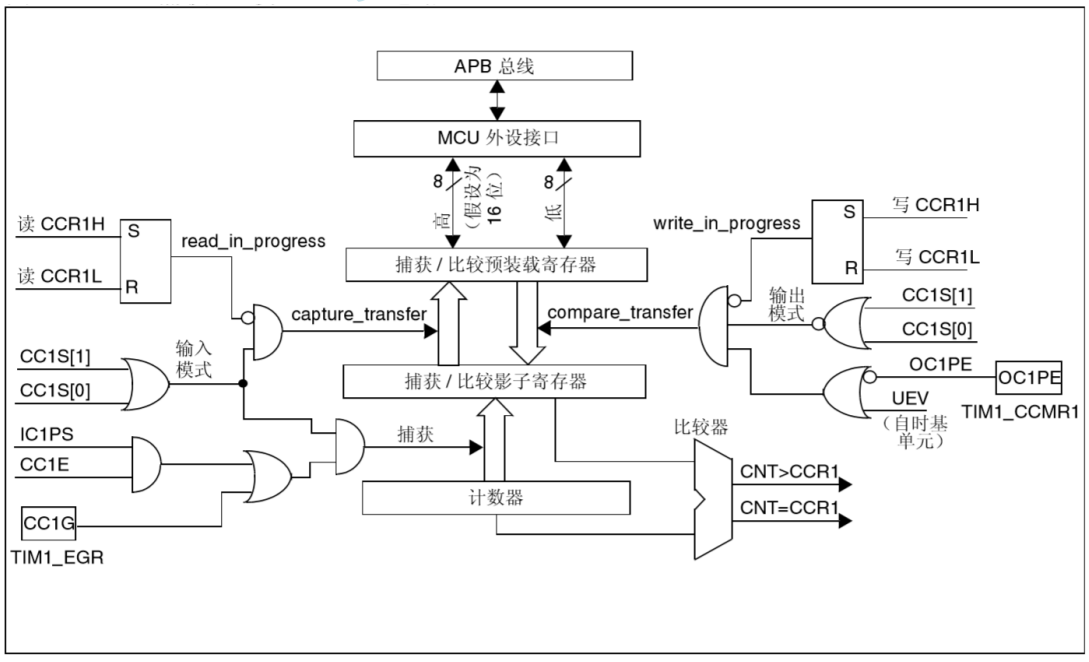
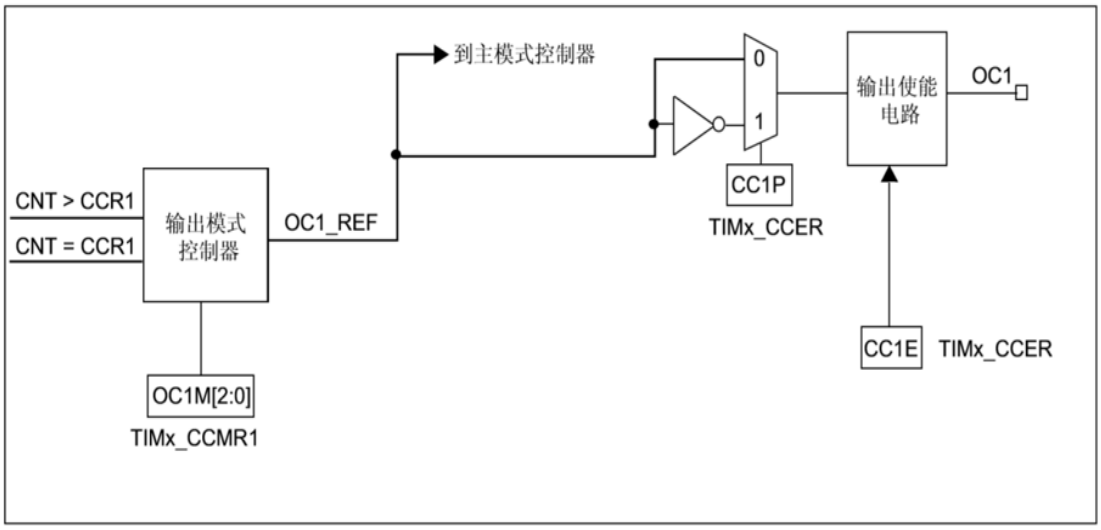


图 180. 捕获/比较通道的输出阶段（通道 1）



捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。始终可通过读写操作访问预装载 寄存器。

在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容复制到预装载寄存 器中。在比较模式下，预装载寄存器的内容将复制到影子寄存器中，然后将影子寄存器的内容与计 数器进行比较。

17.4.5输入捕获模式

在输入捕获模式下，当相应的 ICx 信号检测到跳变沿后，将使用捕获/比较寄存器 (PWMx_CCRx) 来锁存计数器的值。发生捕获事件时，会将相应的 CCXIF 标志（PWMx_SR 寄存器）置 1，并可发送中断或 DMA 请求（如果已使能）。如果发生捕获事件时 CCxIF 标志已处于高位，则会将重复捕获标志 CCxOF（PWMx_SR 寄存器）置 1。可通过软件向 CCxIF 写入 0 来给 CCxIF 清零，或读取存储在 PWMx_CCRx 寄存器中的已捕获数据。向 CCxOF 写入“0”后会将其清零。以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 PWMx_CCR1 中。具体操作步骤如下：

1. 选择有效输入：PWMx_CCR1 必须连接到 TI1 输入，因此向 PWMx_CCMR1 寄 存器中的 CC1S 位写入“01”。只要 CC1S 不等于“00”，就会将通道配置为输入模式，并且 PWMx_CCR1 寄存器将处于只读状态。
2. 根据连接到定时器的信号，对所需的输入滤波时间进行编程（如果输入为 TIx 输入之一，则对 PWMx_CCMRx 寄存器中的 ICxF 位进行编程）。信号变化时，输入信号多在 5 个内部时钟周期内发生抖动。因此，我们必须将滤波时间设置为大于 5 个内部时钟周期。在检测到 8 个具有新电平的连续采样（以 fDTS 频率采样）后，可以确认 TI1 上的跳变沿。然后向 PWMx_CCMR1 寄存器中的 IC1F 位写入“0011”。
3. 通过在 PWMx_CCER 寄存器中将 CC1P 位和 CC1NP 位编程为“00”，选择 TI1 上的有效转换边沿（本例中为上升沿）。
4. 对输入预分频器进行编程。在本例中，我们希望每次有效转换时都执行捕获操作，因此 需要禁止预分频器（向 PWMx_CCMR1 寄存器中的 IC1PS 位写入“00”）。 5. 通过将 PWMx_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
6. 如果需要，可通过将 PWMx_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求。

发生输入捕获时：

- 发生有效跳变沿时，PWMx_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。
- 根据 CC1IE 位生成中断。

要处理重复捕获，建议在读出捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出 标志之后与读取数据之前可能出现的重复捕获信息。

注 意 ： 通 过 软 件 将 PWMx_EGR 寄存器中的相应 CCxG 位置 1 可生成 IC 中断请求。

17.4.6PWM 输入模式（仅适用于 PWM9/12）

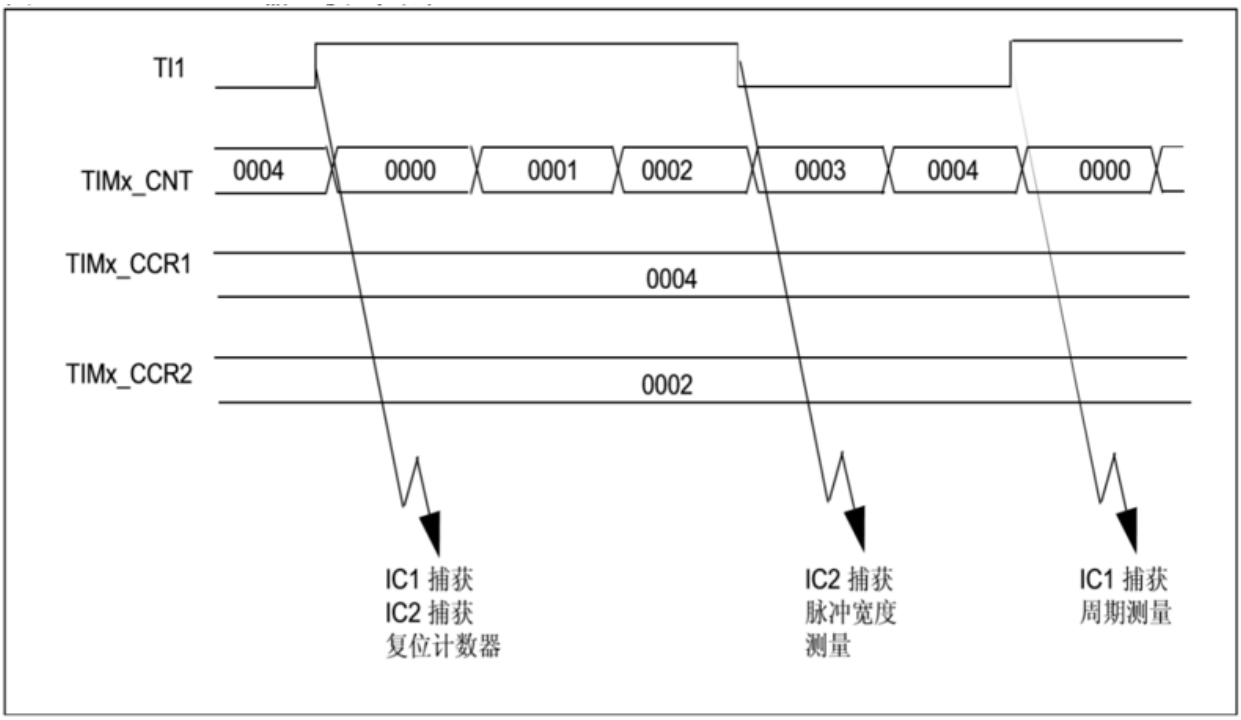
此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这两个 ICx 信号在边沿处有效，但极性相反。
- 选择两个 TIxFP 信号之一作为触发输入，并将从模式控制器配置为复位模式。

例如，可通过以下步骤对应用于 TI1 的 PWM 的周期（位于 PWMx_CCR1 寄存器中）和占空比（位于 PWMx_CCR2 寄存器中）进行测量（取决于 CK_INT 频率和预分频器的值）：

1. 选择 PWMx_CCR1 的有效输入：向 PWMx_CCMR1 寄存器中的 CC1S 位写入“01”（选择 TI1）。
2. 选择 TI1FP1 的有效极性（同时用于 PWMx_CCR1 中的捕获和计数器清零）：将 CC1P 位和 CC1NP 位编程为“00”（上升沿有效）。
3. 选择 PWMx_CCR2 的有效输入：向 PWMx_CCMR1 寄存器中的 CC2S 位写入“10”（选择 TI1）。
4. 选择 TI1FP2 的有效极性（用于 PWMx_CCR2 中的捕获）：将 CC2P 位和 CC2NP 位 编程为“11”（下降沿有效）。
5. 选择有效触发输入：向 PWMx_SMCR 寄存器中的 TS 位写入“101”（选择 TI1FP1）。
6. 将从模式控制器配置为复位模式：向 PWMx_SMCR 寄存器中的 SMS 位写入“100”。
7. 使能捕获：向 PWMx_CCER 寄存器中的 CC1E 位和 CC2E 位写入“1”。

图 181. PWM 输入模式时序



1. PWM 输入模式只能与 PWMx_CH1/PWMx_CH2 信号配合使用，因为只有 TI1FP1 和 TI2FP2 与从模式控制器 相连。

17.4.7强制输出模式

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

要将输出比较信号 (OCXREF/OCx) 强制设置为有效电平，只需向相应 PWMx_CCMRx 寄存器中的 OCxM 位写入“101”。OCXREF 进而强制设置为高电平（OCxREF 始终为高电平有效），同时 OCx 获取 CCxP 极性位的相反值。

例如：CCxP=“0”（OCx 高电平有效）=> 将 OCx 强制设置为高电平。

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

17.4.8输出比较模式

此功能用于控制输出波形，或指示已经过某一时间段。

当捕获/比较寄存器与计数器之间相匹配时，输出比较功能：

1. 将为相应的输出引脚分配一个可编程值，该值由输出比较模式（PWMx_CCMRx 寄存器中的 OCxM 位）和输出极性（PWMx_CCER 寄存器中的 CCxP 位）定义。匹配时，输出引脚既可保持其电平（OCXM="000"），也可设置为有效电平（OCxM="001"）、 无效电平（OCxM="010"）或进行翻转（OCxM="011"）。
2. 将中断状态寄存器中的标志置 1（PWMx_SR 寄存器中的 CCxIF 位）。
3. 如果相应中断使能位（PWMx_DIER 寄存器中的 CCXIE 位）置 1，将生成中断。

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

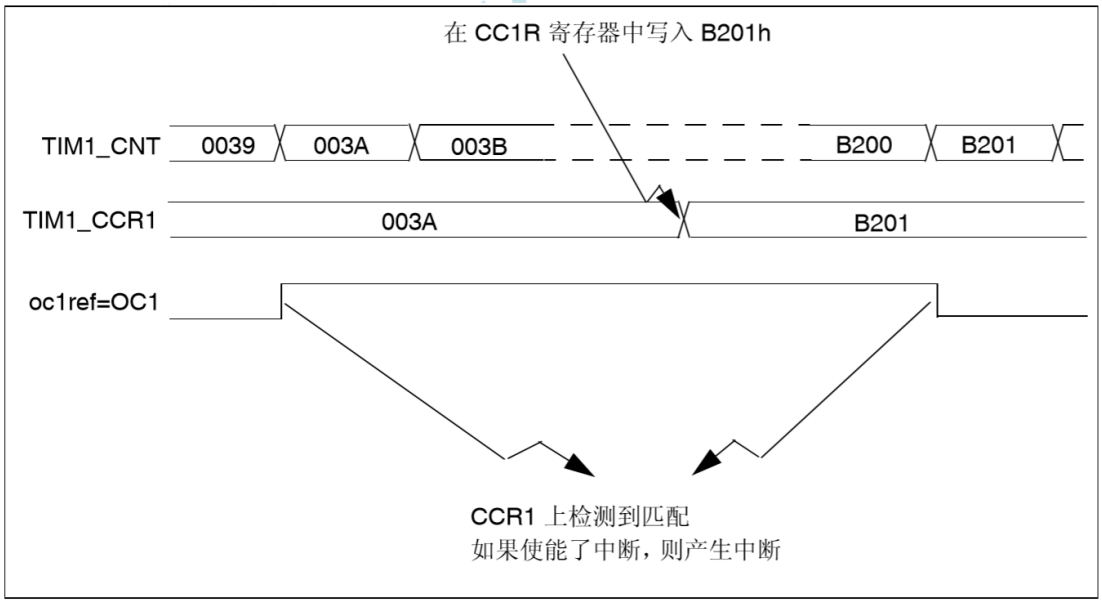
在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出毫无影响。同步的精度可以达到计数器的一个计数周期。输出比较模式也可用于输出单脉冲（在单脉冲模式下）。

步骤：

1. 选择计数器时钟（内部、外部、预分频器）。
2. 在 PWMx_ARR 和 PWMx_CCRx 寄存器中写入所需数据。
3. 如果要生成中断请求，则需将 CCxIE 位置 1。
4. 选择输出模式。例如：
   - 当 CNT 与 CCRx 匹配时，写入 OCxM ="011"以翻转 OCx 输出引脚
   - 写入 OCxPE ="0"以禁止预装载寄存器
   - 写入 CCxP ="0"以选择高电平有效极性
   - 写入 CCxE ="1"以使能输出
5. 通过将 PWMx_CR1 寄存器中的 CEN 位置 1 来使能计数器。

可通过软件随时更新 PWMx_CCRx 寄存器以控制输出波形，前提是未使能预加载寄存器（OCxPE="0"，否则仅当发生下一个更新事件 UEV 时，才会更新 PWMx_CCRx 影子寄存器）。 图 182 给出了一个示例。

图 182. 输出比较模式，翻转 OC1



17.4.9PWM 模式

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

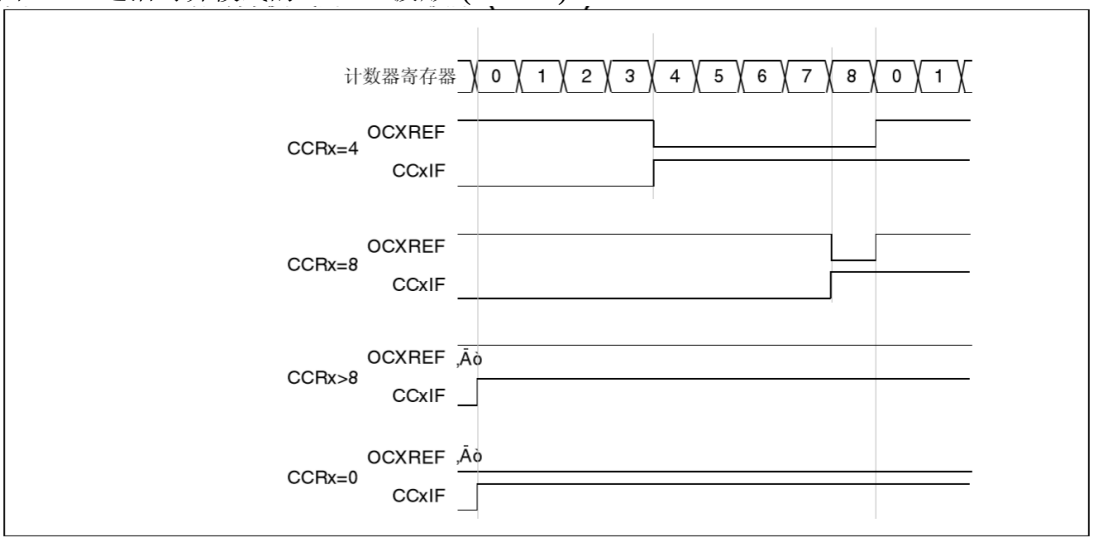
通过向 PWMx_CCMRx 寄存器中的 OCxM 位写入 110（PWM 模式 1）或 111（PWM 模式 2），可以独立选择各通道（每个 OCx 输出对应一个 PWM）的 PWM 模式。必须通过将 PWMx_CCMRx 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器，后通过将 PWMx_CR1 寄存器中的 ARPE 位

置 1 使能自动重载预装载寄存器（在递增计数或中心对齐模式下）。
由于只有在发生更新事件时预装载寄存器才会传送到影子寄存器，因此启动计数器之前，必须通过将 PWMx_EGR 寄存器中的 UG 位置 1 来初始化所有寄存器。
OCx 极性可使用 PWMx_CCER 寄存器的 CCxP 位 来编程。既可以设为高电平有效，也可以设为低电平有效。OCx 输出通过将 PWMx_CCER 寄存器中的 CCxE 位置 1 来使能。有关详细信息，请参见 PWMx_CCERx 寄存器说明。
在 PWM 模式（1 或 2）下，PWMx_CNT 总是与 PWMx_CCRx 进行比较，以确定是 PWMx_CNTPWMx_CCRx。
因为计数器采用递增方式计数，所以定时器能够在边沿对齐模式下生成 PWM。

PWM 边沿对齐模式

以下以 PWM 模式 1 为例。只要 PWMx_CNT < PWMx_CCRx，PWM 参考信号 OCxREF 便为高电平，否则为低电平。如果 PWMx_CCRx 中的比较值大于自动重载值（PWMx_ARR 中），则 OCxREF 保持为“1”。如果比较值为 0，则 OCxRef 保持为“0”。 图 183 举例介绍边沿对齐模式的一些 PWM 波形 (PWMx_ARR=8)。

图 183. 边沿对齐模式的 PWM 波形 (ARR=8)

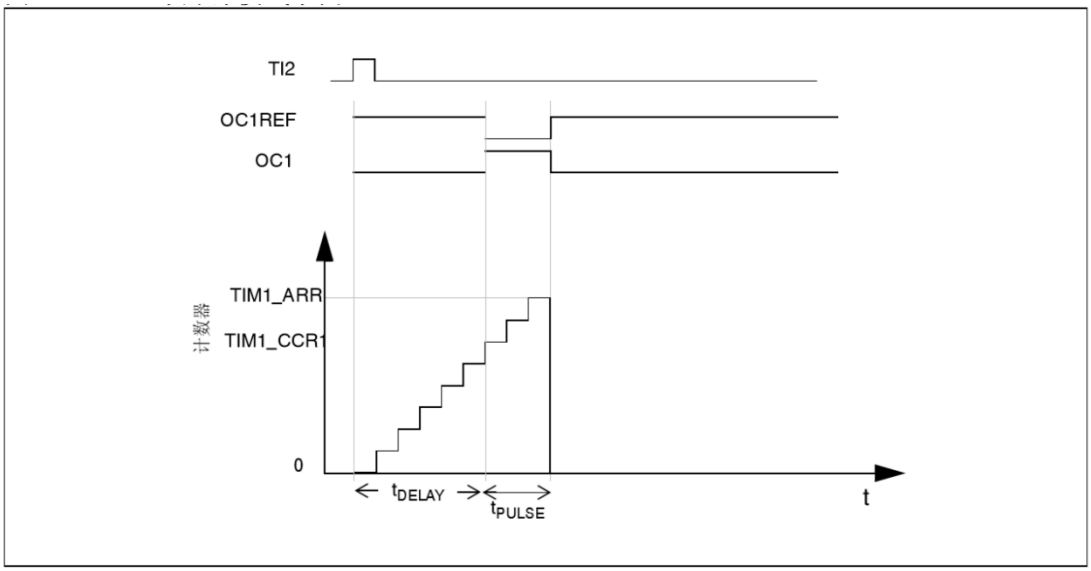


17.4.10 单脉冲模式

单脉冲模式 (OPM) 是上述模式的一个特例。在这种模式下，计数器可以在一个激励信号的 触发下启动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。
可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。将 PWMx_CR1 寄存器中的 OPM 位置 1，即可选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。
只有当比较值与计数器初始值不同时，才能正确产生一个脉冲。启动前（定时器等待触发 时），必须进行如下配置：

CNT < CCRxARR（特别注意，0 < CCRx）

图 184. 单脉冲模式示例



例如，用户希望达到这样的效果：在 TI2 输入引脚检测到正沿时，经过 tDELAY 的延迟，在 OC1 上产生一个长度为 tPULSE 的正脉冲。

使用 TI2FP2 作为触发 1：

1. 在 PWMx_CCMR1 寄存器中写入 CC2S="01"，以将 TI2FP2 映射到 TI2。
2. 在 PWMx_CCER 寄存器中写入 CC2P="0"和 CC2NP="0"，使 TI2FP2 能够检测上升沿。
3. 在 PWMx_SMCR 寄存器中写入 TS="110"，以将 TI2FP2 配置为从模式控制器的触发 (TRGI)。
4. 在 PWMx_SMCR 寄存器中写入 SMS="110"（触发模式），以使用 TI2FP2 启动计数器。OPM 波形通过对比较寄存器执行写操作来定义（考虑时钟频率和计数器预分频器）。

- tDELAY 由写入 PWMx_CCR1 寄存器的值定义。
- tPULSE 由自动重载值与比较值 (PWMx_ARR - PWMx_CCR1) 之差来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从"0"变为"1"，在计数器达到自动 重载值时由"1"变为"0"。为此，应在 PWMx_CCMR1 寄存器中写入 OC1M="111"，以使能 PWM 模式 2。如果需要，可选择在 PWMx_CCMR1 寄存器的 OC1PE 和 PWMx_CR1 寄存器的 ARPE 中写入"1"，以使能预装载寄存器。这种情况下，必须在 PWMx_CCR1 寄存器中写入比较值并在 PWMx_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI2 上的外部触发事件。本例中，CC1P 的值为"0"。

由于仅需要 1 个脉冲（单脉冲模式），因此应向 PWMx_CR1 寄存器的 OPM 位写入"1"，以便在发生下一更新事件（计数器从自动重载值返回到 0）时使计数器停止计数。PWMx_CR1 寄存器中的 OPM 位置"0"时，即选择重复模式。

特例：OCx 快速使能

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

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

17.4.11 PWM9/12 外部触发同步

PWM9/12 定时器可以下列模式与外部触发实现同步：复位模式、门控模式和触发模式。

从模式：复位模式

当触发输入信号产生变化时，计数器及其预分频器可重新初始化。此外，如果 PWMx_CR1 寄存 器中的 URS 位处于低电平，则会生成更新事件 UEV。然后，所有预装载寄存器（PWMx_ARR 和 PWMx_CCRx）都将更新。

在以下示例中，TI1 输入上出现上升沿时，递增计数器清零：

1. 将通道 1 配置为检测 TI1 的上升沿。配置输入滤波时间（本例中不需要任何滤波器，因此 保

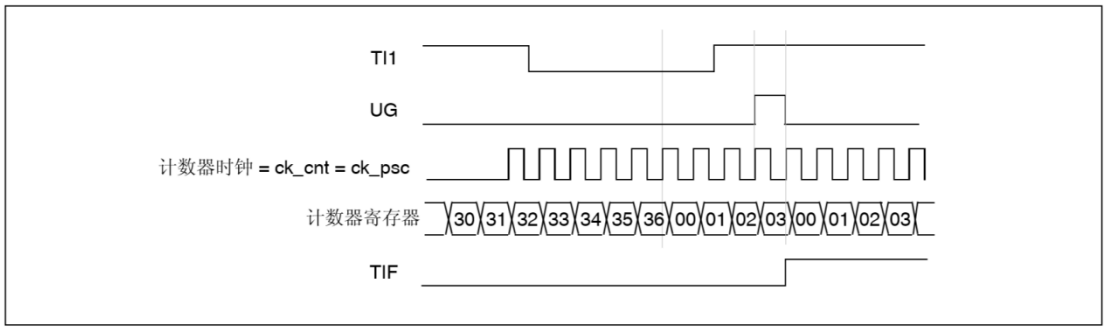
持 IC1F=“0000”）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。 CC1S 位只选择输入捕获源，即 PWMx_CCMR1 寄存器中的 CC1S =“01”。将 PWMx_CCER 寄存器中的 CC1P 和 CC1NP 编程为“00”，以验证极性（仅检测上升沿）。

2. 在 PWMx_SMCR 寄存器中写入 SMS=“100”，将定时器配置为复位模式。在 PWMx_SMCR 寄存器中写入 TS=“101”，选择 TI1 作为输入源。
3. 通过在 PWMx_CR1 寄存器中写入 CEN=“1”来启动动计数器。

计数器使用内部时钟计数，然后正常运转，直到出现 TI1 上升沿。当 TI1 出现上升沿时，计数器清零，然后重新从 0 开始计数。同时，触发标志（PWMx_SR 寄存器中的 TIF 位）置 1，使能中断，还可发送中断请求（取决于 PWMx_DIER 寄存器中的 TIE 位）。

下图显示了自动重载寄存器 PWMx_ARR=0x36 时的相关行为。TI1 的上升沿与实际计数器复位之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 185. 复位模式下的控制电路



从模式：门控模式

输入信号的电平可用来使能计数器。

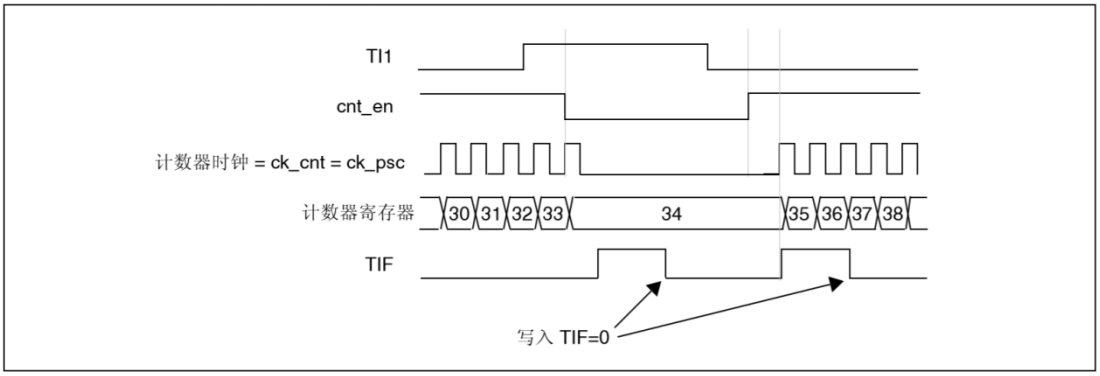
在以下示例中，递增计数器仅在 TI1 输入为低电平时计数：

1. 将通道 1 配置为检测 TI1 上的低电平。配置输入滤波时间（本例中不需要任何滤波器，因此保持 IC1F=“0000”）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。 CC1S 位只选择输入捕获源，即 PWMx_CCMR1 寄存器中的 CC1S =“01”。将 PWMx_CCER 寄存器中的 CC1P 和 CC1NP 分别编程为“1”和“0”，以确定极性（仅检测低电平）。
2. 在 PWMx_SMCR 寄存器中写入 SMS=“101”，将定时器配置为门控模式。在 PWMx_SMCR 寄存器中写入 TS=“101”，选择 TI1 作为输入源。
3. 在 PWMx_CR1 寄存器中写入 CEN=“1”，使能计数器（在门控模式下，如果 CEN= “0”，则不论触发输入电平如何，计数器都不启动）。

只要 TI1 为低电平，计数器就开始根据内部时钟计数，直到 TI1 变为高电平时停止计数。计数器启动或停止时，PWMx_SR 寄存器中的 TIF 标志都会置 1。

TI1 的上升沿与实际计数器停止之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 186. 门控模式下的控制电路



从模式：触发模式

所选输入上发生某一事件时可以启动计数器。

在以下示例中，TI2 输入上出现上升沿时，递增计数器启动：

1. 将通道 2 配置为检测 TI2 上的上升沿。配置输入滤波时间（本例中不需要任何滤波器，因 此

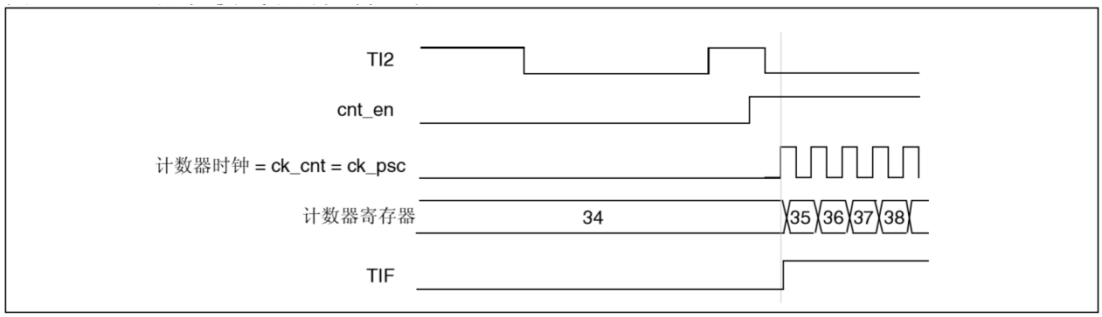
保持 IC2F="0000"）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。 CC2S 位只选择输入捕获源，即 PWMx_CCMR1 寄存器中的 CC2S="01"。将 PWMx_CCER 寄存器中的 CC2P 和 CC2NP 分别编程为"1"和"0"，以确定极性（仅检测低电平）。

2. 在 PWMx_SMCR 寄存器中写入 SMS="110"，将定时器配置为触发模式。在 PWMx_SMCR 寄存器中写入 TS="110"，选择 TI2 作为输入源。

当 TI2 出现上升沿时，计数器开始根据内部时钟计数，并且 TIF 标志置 1。

TI2 的上升沿与实际计数器启动之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 187. 触发模式下的控制电路



17.4.12 定时器同步 (PWM9/12)

PWM 定时器从内部链接在一起，以实现定时器同步或级联。

17.4.13 调试模式

当微控制器进入调试模式（Cortex™-MSTAR 内核停止）时，PWMx 计数器会根据 DBG 模块中的 DBG_PWMx_STOP 配置位选择继续正常工作或者停止工作。

17.5 PWM9 和 PWM12 寄存器

有关寄存器说明中使用的缩写，请参见 第 1.1 节 。

外设寄存器的写访问仅支持半字（16 位）或字（32 位）。而读访问可支持字节（8 位）、半字（16 位）或字（32 位）。

17.5.1 PWM9/12 控制寄存器 1 (PWMx_CR1)

PWM9/12 control register 1

偏移地址：0x00

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CKD[1:0]		ARPE	Reserved			OPM	URS	UDIS	CEB
						rw	rw	rw				rw	rw	rw	rw

位 15:10 保留，必须保持复位值。

位 9:8 CKD：时钟分频 (Clock division)

此位域指示定时器时钟 (CK_INT) 频率与数字滤波器所使用的采样时钟 (TIx) 之间的分频比，

00：tDTS=tCK_INT

01：tDTS = 2 × tCK_INT

10：tDTS = 4 × tCK_INT

11：保留

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

0：PWMx_ARR 寄存器不进行缓冲。

1：PWMx_ARR 寄存器进行缓冲。

位 6:4 保留，必须保持复位值。

位 3 OPM：单脉冲模式 (One-pulse mode)

0：计数器在发生更新事件时不会停止计数

1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）

位 2 URS：更新请求源 (Update request source)

此位由软件置 1 和清零，用以选择 UEV 事件源。

0：使能后，所有以下事件都会生成更新中断：

— 计数器上溢

— 将 UG 位置 1

1：使能后，只有计数器上溢会生成更新中断。

位 1 UDIS：更新禁止 (Update disable)

此位由软件置 1 和清零，用以使能/禁止更新事件 (UEV) 生成。

0：使能 UEV。UEV 可通过以下事件之

一生成：

— 计数器上溢

— 将 UG 位置 1

然后更新影子寄存器的值。

1：禁止 UEV。不会生成 UEV，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。如果 UG 位置 1，则会重新初始化计数器和预分频器。

位 0 CEN：计数器使能 (Counter enable)

0：禁止计数器

1：使能计数器 在单脉冲模式下，

当发生更新事件时会自动将 CEN 位清零。

17.5.2 PWM9/12 控制寄存器 2 (PWMx_CR2)

PWM9/12 control register 2

偏移地址：0x04

复位值：0x0000

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

位 15:7 保留，必须保持复位值。

位 6:4 MMS：主模式选择 (Master mode selection) 这些位用于选择主模式下将要发送到从定时器以实现同步的信息 (TRGO)。这些位的组合如下：

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

001： 使能——计数器使能信号 (CNT_EN) 用作触发输出 (TRGO)。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号由 CEN 控制位与门控模式下的触发输入的逻辑或运算组合而成。 当计数器使能信号由触发输入控制时，TRGO 上会存在延迟，选择主/从模式时除外（请参见 PWMx_SMCR 寄存器中对 MSM 位的说明）。

010： 更新——选择更新事件作为触发输出 (TRGO)。例如，主定时器可用作从定时器的预分 频器。

011： 比较脉冲——一旦发生输入捕获或比较匹配事件，当 CC1IF 被置 1 时（即使已为高电 平），触发输出都会发送一个正脉冲。(TRGO)。

100： 比较——OC1REF 信号用作触发输出 (TRGO)。

101： 比较——OC2REF 信号用作触发输出 (TRGO)。

110： 保留。 111：保留。

位 3:0 保留，必须保持复位值。

17.5.3 PWM9/12 从模式控制寄存器 (PWMx_SMCR)

PWM9/12 slave mode control register

STC MCU

偏移地址：0x08
复位值：0x0000

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

位 15:8 保留，必须保持复位值。
位 7 MSM：主/从模式 (Master/Slave mode)
0：不执行任何操作。
1：延迟事件对触发输入 (TRGI) 的影响，以使当前定时器与其从定时器实现完美同步（通过 TRGO）。此设置适用于要在发生单个外部事件时对多个定时器进行同步的情况。
位 6:4 TS：触发选择 (Trigger selection) 此位域可选择将要用于同步计数器的触发输入。
000：内部触发 0 (ITR0)
001：内部触发 1 (ITR1)
010：内部触发 2 (ITR2)
011：内部触发 3 (ITR3)
100：TI1 边沿检测器 (TI1F_ED)
101：滤波后的定时器输入 1 (TI1FP1)
110：滤波后的定时器输入 2 (TI2FP2)
111：保留
有关各定时器 ITRx 含义的详细信息，请参见表 79 ： PWMx 内部触发连接 。
注意：这些位只能在未使用的情况下（例如， SMS=“ 000 ”时）进行更改，以避免转换时出现 错误的边沿检测。
位 3 保留，必须保持复位值。
位 2:0 SMS：从模式选择 (Slave mode selection)
选择外部信号时，触发信号 (TRGI) 的有效边沿与外部输入上所选的极性相关（请参见输入控制寄存器和控制寄存器说明）。
000：禁止从模式——如果 CEN =“1”，预分频器时钟直接由内部时钟提供
001：保留
010：保留
011：保留
100：复位模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器并生成一个寄存器更新事件
101：门控模式——触发输入 (TRGI) 为高电平时使能计数器时钟。只要触发输入变为低电平，计数器立即停止计数（但不复位）。计数器的启动和停止都是受控的。
110：触发模式——触发信号 TRGI 出现上升沿时启动计数器（但不复位）。只控制计数器的启动
111：外部时钟模式 1——在所选触发 (TRGI) 的上升沿驱动计数器
注意：如果将 TI1F_ED 选作触发输入（ TS=“ 100 ”），则不得使用门控模式。实际上， TI1F 每次转换时， TI1F_ED 都输出 1 个脉冲，而门控模式检查的则是触发信号的电平。
表 79. PWMx 内部触发连接

从 TIM	ITR0 （TS = “000”）	ITR1 （TS = “001”）	ITR2 （TS = “010”）	ITR3 （TS = “011”）
TIM2	TIM1	TIM8	TIM3	TIM4
TIM3	TIM1	TIM2	TIM5	TIM4
TIM4	TIM1	TIM2	TIM3	TIM8
TIM5	TIM2	TIM3	TIM4	TIM8
TIM9	TIM2	TIM3	TIM10	TIM11
TIM12	TIM4	TIM5	TIM13	TIM14

17.5.4 PWM9/12 中断使能寄存器 (PWMx_DIER)

PWM9/12 Interrupt enable register
偏移地址：0x0C
复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									TIE	Reserved			CC2IE	CC1IE	UIE
									rw				rw	rw	rw

位 15:7 保留，必须保持复位值。
位 6 TIE：触发信号 (TGRI) 中断使能 (Trigger interrupt enable)
0：禁止触发信号 (TGRI) 中断。
1：使能触发信号 (TGRI) 中断。
位 5:3 保留，必须保持复位值。
位 2 CC2IE：捕获/比较 2 中断使能 (Capture/Compare 2 interrupt enable)
0：禁止 CC2 中断。
1：使能 CC2 中断。
位 1 CC1IE：捕获/比较 1 中断使能 (Capture/Compare 1 interrupt enable)
0：禁止 CC1 中断。
1：使能 CC1 中断。
位 0 UIE：更新中断使能 (Update interrupt enable)
0：禁止更新中断。
1：使能更新中断。

17.5.5 PWM9/12 状态寄存器 (PWMx_SR)

PWM9/12 status register
偏移地址：0x10
复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					CC2OF	CC1OF	Reserved		TIF	Reserved			CC2IF	CC1IF	UIF
					rc_w0	rc_w0			rc_w0				rc_w0	rc_w0	rc_w0

位 15:11 保留，必须保持复位值。
位 10 CC2OF：捕获/比较 2 重复捕获标志 (Capture/compare 2 overcapture flag)
请参见 CC1OF 说明
位 9 CC1OF：捕获/比较 1 重复捕获标志 (Capture/Compare 1 overcapture flag)
仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将 该

位清零。

0：未检测到重复捕获。

1：PWMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。

位 8:7 保留，必须保持复位值。

位 6 TIF：触发中断标志 (Trigger interrupt flag)

在除门控模式以外的所有模式下，当使能从模式控制器后在 TRGI 输入上检测到有效边沿时，该标志将由硬件置 1。选择门控模式时，该标志将在计数器启动或停止时置 1。但需要通过软件清零。

0：未发生触发事件。

1：触发中断挂起。

位 5:3 保留，必须保持复位值。

位 2 CC2IF：捕获/比较 2 中断标志 (Capture/Compare 2 interrupt flag)

请参见 CC1IF 说明

位 1 CC1IF：捕获/比较 1 中断标志 (Capture/compare 1 interrupt flag)

如果通道 CC1 配置为输出：

当计数器与比较值匹配时，此标志由硬件置 1。但需要通过软件清零。

0：不匹配。

1：PWMx_CNT 计数器的值与 PWMx_CCR1 寄存器的值匹配。当 PWMx_CCR1 的值大于 PWMx_ARR 的值时，CC1IF 位将在计数器发生上溢时变为高电平。

如果通道 CC1 配置为输入：

此位将在发生捕获事件时由硬件置 1。通过软件或读取 PWMx_CCR1 寄存器将该位清零。

0：未发生输入捕获事件。

1：PWMx_CCR1 寄存器中已捕获到计数器值（IC1 上已检测到与所选极性匹配的边沿）。

位 0 UIF：更新中断标志 (Update interrupt flag)

该位在发生更新事件时通过硬件置 1。但需要通过软件清零。

0：未发生更新。

1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：

—上溢且当 PWMx_CR1 寄存器中 UDIS =“0”时。

—当由于 PWMx_CR1 寄存器中 URS =“0”且 UDIS =“0”而通过软件使用 PWMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。

—当由于 PWMx_CR1 寄存器中 URS =“0”且 UDIS =“0”而通过触发事件（请参见同步控制寄存器说明）重新初始化 CNT 时。

17.5.6 PWM9/12 事件生成寄存器 (PWMx_EGR)

PWM9/12 event generation register

偏移地址：0x14

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									TG	Reserved			CC2G	CC1G	UG
									w				w	w	w

位 15:7 保留，必须保持复位值。

位 6 TG：生成触发信号 (Trigger generation)

此位由软件置 1 以生成事件，并由硬件自动清零。

0：不执行任何操作

1：PWMx_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断

位 5:3 保留，必须保持复位值。

位 2 CC2G：捕获/比较 2 生成 (Capture/compare 2 generation)

请参见 CC1G 说明

位 1 CC1G：捕获/比较 1 生成 (Capture/compare 1 generation)

此位由软件置 1 以生成事件，并由硬件自动清零。
0：不执行任何操作
1：通道 1 上生成捕获/比较事件：
如果通道 CC1 配置为输出：
使能后，CC1IF 标志置 1 并发送相应中断。
如果通道 CC1 配置为输入：
PWMx_CCR1 寄存器中将捕获到计数器的当前值。使能后，CC1IF 标志置 1 并发送相应中断。
如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。

位 0 UG：更新生成 (Update generation)
该位可通过软件置 1，并由硬件自动清零。
0：不执行任何操作
1：重新初始化计数器并生成寄存器更新事件。预分频器计数器也将清零，但预分频比不受影 响。计数器清零

17.5.7 PWM9/12 捕获/比较模式寄存器 1 (PWMx_CCMR1)

PWM9/12 capture/compare mode register 1
偏移地址：0x18
复位值：0x0000
这些通道可用于输入（捕获模式）或输出（比较模式）模式。通道方向通过配置相应的 CCxS 位进行定义。该寄存器的所有其它位在输入模式和输出模式下的功能不同。对于任一给定位， OCxx 用于说明通道配置为输出模式时该位对应的功能，ICxx 则用于说明通道配置为输入模 式时该位对应的功能。因此，必须注意同一个位在输入阶段和输出阶段具有不同的含义。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	OC2M [2:0]			OC2 PE	OC2 FE	CC2S[1:0]		Res	OC1M[2:0]			OC1 PE	OC1 FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]				IC1PSC[1:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式
位 15 保留，必须保持复位值。
位 14:12 OC2M[2:0]：输出比较 2 模式 (Output compare 2 mode)
位 11 OC2PE：输出比较 2 预装载使能 (Output compare 2 preload enable)
位 10 OC2FE：输出比较 2 快速使能 (Output compare 2 fast enable)
位 9:8 CC2S[1:0]：捕获/比较 2 选择 (Capture/Compare 2 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC2 通道配置为输出
01：CC2 通道配置为输入，IC2 映射到 TI2 上
10：CC2 通道配置为输入，IC2 映射到 TI1 上
11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（PWMx_SMCR 寄存器）选择内部触发输入时有效
注意：仅当通道关闭时（ PWMx_CCER 中的 CC2E = 0 ），才可向 CC2S 位写入数据。
位 7 保留，必须保持复位值。
位 6:4 OC1M：输出比较 1 模式 (Output compare 1 mode)
这些位定义提供 OC1 和 OC1N 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 和 OC1N 的有效电平则分别取决于 CC1P 位和 CC1NP 位。
000：冻结——输出比较寄存器 PWMx_CCR1 与计数器 PWMx_CNT 进行比较不会对输出造成任何影响。（该模式用于生成时基）。
001：将通道 1 设置为匹配时输出有效电平。当计数器 PWMx_CNT 与捕获/比较寄存器 1 (PWMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。

010：将通道 1 设置为匹配时输出无效电平。当计数器 PWMx_CNT 与捕获/比较寄存器 1 (PWMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。
011：翻转——PWMx_CNT=PWMx_CCR1 时，OC1REF 发生翻转。
100：强制变为无效电平——OC1REF 强制变为低电平。
101：强制变为有效电平——OC1REF 强制变为高电平。
110：PWM 模式 1——在递增计数模式下，只要 PWMx_CNT<PWMx_CCR1，通道 1 便为有效状态，否则为无效状态。在递减计数模式下，只要 PWMx_CNT>PWMx_CCR1，通道 1 便为无 效状态（OC1REF=“0”），否则为有效状态（OC1REF=“1”）。 111：PWM 模式 2——在递增计数模式下，只要 PWMx_CNT<PWMx_CCR1，通道 1 便为无效状态，否则为有效状态。在递减计数模式下，只要 PWMx_CNT>PWMx_CCR1，通道 1 便为有 效状态，否则为无效状态。
注意：在 PWM 模式 1 或 PWM 模式 2 下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“ PWM ”模式时， OCREF 电平才会发生更改。

位 3 OC1PE：输出比较 1 预装载使能 (Output compare 1 preload enable)
0：禁止与 PWMx_CCR1 相关的预装载寄存器。可随时向 PWMx_CCR1 写入数据，写入后将立即使用新值。
1：使能与 PWMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。PWMx_CCR1 预装载值在每次生成更新事件时都会装载到活动寄存器中。
注意：只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式（ PWMx_CR1 寄 存器中的 OPM 位置 1 ）。其它情况下则无法保证该行为。

位 2 OC1FE：输出比较 1 快速使能 (Output compare 1 fast enable)
此位用于加快触发输入事件对 CC 输出的影响。
0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的短延迟时间为 5 个时钟周期。
1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OC1FE 才会起作用。

位 1:0 CC1S：捕获/比较 1 选择 (Capture/Compare 1 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC1 通道配置为输出
01：CC1 通道配置为输入，IC1 映射到 TI1 上
10：CC1 通道配置为输入，IC1 映射到 TI2 上
11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（PWMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（ PWMx_CCER 中的 CC1E = 0 ），才可向 CC1S 位写入数据。

输入捕获模式

位 15:12 IC2F：输入捕获 2 滤波器 (Input capture 2 filter)
位 11:10 IC2PSC[1:0]：输入捕获 2 预分频器 (Input capture 2 prescaler)
位 9:8 CC2S：捕获/比较 2 选择 (Capture/compare 2 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC2 通道配置为输出
01：CC2 通道配置为输入，IC2 映射到 TI2 上
10：CC2 通道配置为输入，IC2 映射到 TI1 上
11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（PWMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（ PWMx_CCER 中的 CC2E = 0 ），才可向 CC2S 位写入数据。

位 7:4 IC1F：输入捕获 1 滤波器 (Input capture 1 filter)
此位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。数字滤波器由事件计数 器组成，每 N 个事件才视为一个有效边沿：
0000：无滤波器，以 fDTS 频率进行采样 1000：fSAMPLING=fDTS/8，N=6 0001：fSAMPLING=fCK_INT， N=2 1001： fSAMPLING=fDTS/8， N=8 0010：

fSAMPLING=fCK_INT， N=4 1010： fSAMPLING=fDTS/16， N=5 0011：
fSAMPLING=fCK_INT， N=8 1011： fSAMPLING=fDTS/16， N=6 0100：
fSAMPLING=fDTS/2， N=6 1100： fSAMPLING=fDTS/16， N=8 0101：
fSAMPLING=fDTS/2， N=8 1101： fSAMPLING=fDTS/32， N=5 0110：
fSAMPLING=fDTS/4， N=6 1110： fSAMPLING=fDTS/32， N=6 0111：
fSAMPLING=fDTS/4，N=8 1111：fSAMPLING=fDTS/32，N=8

注意：在当前硅版本中，当 ICxF[3:0]= 1 、2 或 3 时，将用 CK_INT 代替公式中的 fDTS 。

位 3:2 IC1PSC：输入捕获 1 预分频器 (Input capture 1 prescaler)
此位域定义 CC1 输入 (IC1) 的预分频比。
只要 CC1E=“0”（PWMx_CCER 寄存器），预分频器便立即复位。
00：无预分频器，捕获输入上每检测到一个边沿便执行捕获
01：每发生 2 个事件便执行一次捕获
10：每发生 4 个事件便执行一次捕获
11：每发生 8 个事件便执行一次捕获

位 1:0 CC1S：捕获/比较 1 选择 (Capture/Compare 1 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC1 通道配置为输出
01：CC1 通道配置为输入，IC1 映射到 TI1 上
10：CC1 通道配置为输入，IC1 映射到 TI2 上
11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（PWMx_SMCR 寄存器）选择内部触发输入时有效

注意：仅当通道关闭时（ PWMx_CCER 中的 CC1E = 0 ），才可向 CC1S 位写入数据。

17.5.8 PWM9/12 捕获/比较使能寄存器 (PWMx_CCER)

PWM9/12 capture/compare enable register
偏移地址：0x20
复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
								rw		rw	rw	rw		rw	rw

位 15:8 保留，必须保持复位值。

位 7 CC2NP：捕获/比较 2 输出极性 (Capture/Compare 2 output polarity)
请参见 CC1NP 说明

位 6 保留，必须保持复位值。

位 5 CC2P：捕获/比较 2 输出极性 (Capture/Compare 2 output polarity)
请参见 CC1P 说明

位 4 CC2E：捕获/比较 2 输出使能 (Capture/Compare 2 output enable)
请参见 CC1E 说明

位 3 CC1NP：捕获/比较 1 互补输出极性 (Capture/Compare 1 complementary output polarity)
CC1 通道配置为输出：CC1NP 必须保持清零
CC1 通道配置为输入：CC1NP 与 CC1P 配合使用可定义 TI1FP1/TI2FP1 的极性（请参见 CC1P 说明）。
位 2 保留，必须保持复位值。

位 1 CC1P：捕获/比较 1 输出极性 (Capture/Compare 1 output polarity) 。
CC1 通道配置为输出：
0：OC1 高电平有效。
1：OC1 低电平有效。
CC1 通道配置为输入： CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的极性。

00：未反相/上升沿触发 电路对 TIxFP1 上升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式或编码器模式下执行触发操作）。

01：反相/下降沿触发 电路对 TIxFP1 下降沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 反相（在门控模式或编码器模式下执行触发操作）。

10：保留，不使用此配置。

注意： 11 ：未反相 /上升沿和下降沿均触发电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作）， TIxFP1 未反相（在门控模式下执行触发操作）。编码器模式下不 得使用此配置。

位 0 CC1E：捕获/比较 1 输出使能 (Capture/Compare 1 output enable) 。

CC1 通道配置为输出：

0：关闭——OC1 未激活。

1：开启——在相应输出引脚上输出 OC1 信号。

CC1 通道配置为输入：

该位决定了输入捕获/比较寄存器 1 (PWMx_CCR1) 是否能够实际捕获到计数器的值。

0：禁止捕获。

1：使能捕获。

表 80. 标准 OCx 通道的输出控制位

CCxE 位	OCx 输出状态
0	禁止输出（OCx=“0”，OCx_EN=“0”）
1	OCx=OCxREF + 极性，OCx_EN=“1”

注 意 ： 与 标 准 OCx 通道相连的外部 I/O 引脚的状态取决于通道 OCx 的状态以及 GPIO 寄存器。

17.5.9 PWM9/12 计数器 (PWMx_CNT)

偏移地址：0x24

复位值：0x0000 0000

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 15:0 CNT[15:0]：计数器值 (Counter value)

17.5.10 PWM9/12 预分频器 (PWMx_PSC)

偏移地址：0x28

复位值：0x0000

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

位 15:0 PSC[15:0]：预分频器值 (Prescaler value)

计数器时钟频率 CK_CNT 等于 fCK_PSC / (PSC[15:0] + 1)。

PSC 包含在每次发生更新事件时要装载到实际预分频器寄存器的值。

17.5.11 PWM9/12 自动重载寄存器 (PWMx_ARR)

PWM9/12 auto-reload register

偏移地址：0x2C

复位值：0x0000 0000

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 15:0 ARR[15:0]：自动重载值 (Auto-reload value)
ARR 为要装载到实际自动重载寄存器的值。
有关 ARR 更新和行为的更多详细信息，请参见：时基单元 。
当自动重载值为空时，计数器不工作。

17.5.12 PWM9/12 捕获/比较寄存器 1 (PWMx_CCR1)

PWM9/12 capture/compare register 1
偏移地址：0x34
复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 CCR1[15:0]：捕获/比较 1 值 (Capture/Compare 1 value)
如果通道 CC1 配置为输出：
CCR1 是捕获/比较寄存器 1 的预装载值。
如果没有通过 PWMx_CCMR 寄存器中的 OC1PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器 1）。 实际捕获/比较寄存器中包含要与计数器 PWMx_CNT 进行比较并在 OC1 输出上发出信号的值。
如果通道 CC1 配置为输入：
CCR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。

17.5.13 PWM9/12 捕获/比较寄存器 2 (PWMx_CCR2)

PWM9/12 capture/compare register 2
偏移地址：0x38
复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 CCR2[15:0]：捕获/比较 2 值 (Capture/Compare 2 value)
如果通道 CC2 配置为输出：
CCR2 为要装载到实际捕获/比较 2 寄存器的值（预装载值）。
如果没有通过 PWMx_CCMR2 寄存器中的 OC2PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只有发生更新事件时，预装载值才会复制到活动捕获/比较 2 寄存器中。
活动捕获/比较寄存器中包含要与计数器 PWMx_CNT 进行比较并在 OC2 输出上发出信号的值。
如果通道 CC2 配置为输入：
CCR2 为上一个输入捕获 2 事件 (IC2) 发生时的计数器值。

17.5.14 PWM9/12 寄存器映射

PWM9/12 寄存器可映射为 16 位可寻址寄存器，如下表所述：

表 81. PWM9/12 寄存器映射和复位值

偏移	寄存器	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

Offset	Register		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0X00	TIMx_CR1	Reserved							CKD [1:0]		ARPE	Res.			OPM	URS	UDIS	CEN
	Reset value								0	0	0				0	0	0	0
0X04	TIMx_CR2	Reserved										MMS[2:0]						
	Reset value											0	0	0				
0X08	TIMx_SMCR	Reserved									MSM	TS[2:0]			Reserved0	SMS[2:0]		
	Reset value										0	0	0	0		0	0	0
0X0C	TIMx_DIER	Reserved										TIE	Res.			CC2IE	CC1IE	UIE
	Reset value											0				0	0	0
0X10	TIMx_SR	Reserved						CC2OF	CC1OF	Reserved	BIF	TIF	Res.			CC2IF	CC1IF	UIF
	Reset value							0	0			0				0	0	0
0X14	TIMx_EGR	Reserved										TG	Reserved			CC2G	CC1G	UG
	Reset value											0				0	0	0
0X18	TIMx_CCMR1 Output compare mode	Reserved		OC2M [2:-0]			OC2PE	OC2FE	CC2S [1:0]		Reserved	OC1M[2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value			0	0	0	0	0	0	0		0	0	0	0	0	0	0
	TIMx_CCMR1 INput compare mode	Reserved	IC2F[3:0]				IC2 PSC [1:0]		CC2S [1:0		CC2S[1:0]				IC1 PSC [1:0]		CC1S [1:0]	
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0X1C		Reserved																
0X20	TIMx_CCER	Reserved									CC2NP		CC2P	CC2E	CC1NP		CC1P	CC1E
	Reset value										0		0	0	0		0	0
0X24	TIMx_CNT	Reserved	CNT[15:0]															
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0X28	TIMx_PSC	Reserved	PSC[15:0]															
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0X2C	TIMx_ARR	Reserved								ARR15:0]							
	Reset value									0	0	0	0	0	0	0	0
0x30	TIMx_RCR	Reserved								REP[7:0]							
	Reset value									0	0	0	0	0	0	0	0
0x34	TIMx_CCR1	Reserved								CCR1[15:0]							
	Reset value									0	0	0	0	0	0	0	0
0x38	TIMx_CCR2	Reserved								CCR2[15:0]							
	Reset value									0	0	0	0	0	0	0	0
0x3C	Reserved																
0x40																	
0x44																	
0x48																	
0x4C																	

17.6 PWM10/11/13/14 寄存器

外设寄存器的写访问仅支持半字（16 位）或字（32 位）。而读访问可支持字节（8 位）、半字（16 位）或字（32 位）。

17.6.1 PWM10/11/13/14 控制寄存器 1 (PWMx_CR1)

PWM10/11/13/14 control register 1
偏移地址：0x00
复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CKD[1:0]		ARPE	Reserved				URS	UDIS	CEB
						rw	rw	rw					rw	rw	rw

位 15:10 保留，必须保持复位值。

位 9:8 CKD：时钟分频 (Clock division)

此位域指示定时器时钟 (CK_INT) 频率与数字滤波器所使用的采样时钟（ETR、TIx）之间的分频比，

00：tDTS=tCK_INT

01：tDTS = 2 × tCK_INT

10：tDTS = 4 × tCK_INT

11：保留

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

0：PWMx_ARR 寄存器不进行缓冲

1：PWMx_ARR 寄存器进行缓冲

位 6:3 保留，必须保持复位值。

位 2 URS：更新请求源 (Update request source)

此位由软件置 1 和清零，用以选择更新中断 (UEV) 源。

0：使能后，所有以下事件都会生成 UEV：

— 计数器上溢

— 将 UG 位置 1

1：使能后，只有计数器上溢会生成 UEV。

位 1 UDIS：更新禁止 (Update disable)

此位由软件置 1 和清零，用以使能/禁止更新中断 (UEV) 事件生成。

0：使能 UEV。UEV 可通过以下事件之一生成：

— 计数器上溢

— 将 UG 位置 1

然后更新影子寄存器的值。

1：禁止 UEV。不会生成 UEV，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。如果 UG 位置 1，则会重新初始化计数器和预分频器。

位 0 CEN：计数器使能 (Counter enable)

0：禁止计数器

1：使能计数器

17.6.2 PWM10/11/13/14 状态寄存器 (PWMx_SR)

PWM10/11/13/14 status register

偏移地址：0x10

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CC1OF	Reserved							CC1IF	UIF
						rc_w0								rc_w0	rc_w0

位 15:10 保留，必须保持复位值。

位 9 CC1OF：捕获/比较 1 重复捕获标志 (Capture/Compare 1 overcapture flag)

仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将该位清零。

0：未检测到重复捕获。

1：PWMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。

位 8:2 保留，必须保持复位值。

位 1 CC1IF：捕获/比较 1 中断标志 (Capture/compare 1 interrupt flag)

如果通道 CC1 配置为输出：

当计数器与比较值匹配时，此标志由硬件置 1。但需要通过软件清零。

0：不匹配。

1：PWMx_CNT 计数器的值与 PWMx_CCR1 寄存器的值匹配。当 PWMx_CCR1 的值大于 PWMx_ARR 的值时，CC1IF 位将在计数器发生上溢时变为高电平。

如果通道 CC1 配置为输入：

此位将在发生捕获事件时由硬件置 1。通过软件或读取 PWMx_CCR1 寄存器将该位清零。

0：未发生输入捕获事件。

1：PWMx_CCR1 寄存器中已捕获到计数器值（IC1 上已检测到与所选极性匹配的边沿）。

位 0 UIF：更新中断标志 (Update interrupt flag)

该位在发生更新事件时通过硬件置 1。但需要通过软件清零。

0：未发生更新。

1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：

— 上溢且当 PWMx_CR1 寄存器中 UDIS =“0”时。

— 当由于 PWMx_CR1 寄存器中 URS =“0”且 UDIS =“0”而通过软件使用 PWMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。

17.6.3 PWM10/11/13/14 事件生成寄存器 (PWMx_EGR)

PWM10/11/13/14 event generation register

偏移地址：0x14

复位值：0x0000

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

Reserved	CC1G	UG
	w	w

位 15:2 保留，必须保持复位值。

位 1 CC1G：捕获/比较 1 生成 (Capture/compare 1 generation)

此位由软件置 1 以生成事件，并由硬件自动清零。

0：不执行任何操作。

1：通道 1 上生成捕获/比较事件：

如果通道 CC1 配置为输出：

使能后，CC1IF 标志置 1 并发送相应的中断。

如果通道 CC1 配置为输入：

PWMx_CCR1 寄存器中将捕获到计数器当前值。使能后，CC1IF 标志置 1 并发送相应中断。

如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。

位 0 UG：更新生成 (Update generation)

该位可通过软件置 1，并由硬件自动清零。

0：不执行任何操作。

1：重新初始化计数器并生成寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。计数器清零。

17.6.4 PWM10/11/13/14 捕获/比较模式寄存器 1 (PWMx_CCMR1)

PWM10/11/13/14 capture/compare mode register 1

偏移地址：0x18

复位值：0x0000

这些通道可用于输入（捕获模式）或输出（比较模式）模式。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其它位在输入模式和输出模式下的功能均不同。对于任一给定 位，OCxx 用于说明通道配置为输出时该位对应的功能，ICxx 则用于说明通道配置为输入时 该位对应的功能。因此，必须注意同一个位在输入阶段和输出阶段具有不同的含义。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
Reserved								IC1F[3:0]				IC1PSC[1:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式

位 15:7 保留，必须保持复位值。

位 6:4 OC1M：输出比较 1 模式 (Output compare 1 mode)

这些位定义提供 OC1 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 的有效电平则取决于 CC1P 位。

000： 冻结。输出比较寄存器 PWMx_CCR1 与计数器 PWMx_CNT 进行比较不会对输出造成任何影响。

001： 将通道 1 设置为匹配时输出有效电平。当计数器 PWMx_CNT 与捕获/比较寄存器 1 (PWMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。

010： 将通道 1 设置为匹配时输出无效电平。当计数器 PWMx_CNT 与捕获/比较寄存器 1 (PWMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。

011： 翻转——PWMx_CNT = PWMx_CCR1 时，OC1REF 发生翻转。 100：强制变为无效电平——OC1REF 强制变为低电平。

101： 强制变为有效电平——OC1REF 强制变为高电平。

110： PWM 模式 1——只要 PWMx_CNT < PWMx_CCR1，通道 1 便为有效状态，否则为无效状态。

111： PWM 模式 2——只要 PWMx_CNT < PWMx_CCR1，通道 1 便为无效状态，否则为

有效状态。

注意：在 PWM 模式 1 或 PWM 模式 2 下，仅当比较结果发生改变或输出比较模式由"冻结"模式切换到"PWM"模式时， OCREF 电平才会发生更改。

位 3 OC1PE：输出比较 1 预装载使能 (Output compare 1 preload enable)

0：禁止与 PWMx_CCR1 相关的预装载寄存器。可随时向 PWMx_CCR1 写入数据，写入后将立即使用新值。

1：使能与 PWMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。PWMx_CCR1 预装载值在每次生成更新事件时都会装载到活动寄存器中。

注意：只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式（ PWMx_CR1 寄存 器中的 OPM 位置 1 ）。其它情况下则无法保证该行为。

位 2 OC1FE：输出比较 1 快速使能 (Output compare 1 fast enable)

此位用于加快触发输入事件对 CC 输出的影响。

0： 即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的短延迟时间为 5 个时钟周期。

1： 触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OC1FE 才会起作用。

位 1:0 CC1S：捕获/比较 1 选择 (Capture/Compare 1 selection) 此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出。

01：CC1 通道配置为输入，IC1 映射到 TI1 上。

10：

11：

注意：仅当通道关闭时（ PWMx_CCER 中的 CC1E = 0 ），才可向 CC1S 位写入数据。

输入捕获模式

位 15:8 保留，必须保持复位值。

位 7:4 IC1F：输入捕获 1 滤波器 (Input capture 1 filter)

此位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个事件才视为一个有效边沿：

0000：无滤波器，以 fDTS 频率进行采样 1000：fSAMPLING=fDTS/8，N=6 0001：fSAMPLING=fCK_INT， N=2 1001： fSAMPLING=fDTS/8， N=8 0010：fSAMPLING=fCK_INT， N=4 1010： fSAMPLING=fDTS/16， N=5 0011：fSAMPLING=fCK_INT， N=8 1011： fSAMPLING=fDTS/16， N=6 0100：fSAMPLING=fDTS/2， N=6 1100： fSAMPLING=fDTS/16， N=8 0101：fSAMPLING=fDTS/2， N=8 1101： fSAMPLING=fDTS/32， N=5 0110：fSAMPLING=fDTS/4， N=6 1110： fSAMPLING=fDTS/32， N=6 0111：fSAMPLING=fDTS/4，N=8 1111：fSAMPLING=fDTS/32，N=8

注意：在当前硅版本中，当 ICxF[3:0]= 1 、 2 或 3 时，将用 CK_INT 代替公式中的 fDTS 。

位 3:2 IC1PSC：输入捕获 1 预分频器 (Input capture 1 prescaler)

此位域定义 CC1 输入 (IC1) 的预分频比。 只要 CC1E="0"（PWMx_CCER 寄存器），预分频器便立即复位。

00：无预分频器，捕获输入上每检测到一个边沿便执行捕获

01：每发生 2 个事件便执行一次捕获

10：每发生 4 个事件便执行一次捕获

11：每发生 8 个事件便执行一次捕获

位 1:0 CC1S：捕获/比较 1 选择 (Capture/Compare 1 selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出

01：CC1 通道配置为输入，IC1 映射到 TI1 上

10：保留

11：保留

注意：仅当通道关闭时（ PWMx_CCER 中的 CC1E = 0 ），才可向 CC1S 位写入数据

17.6.5 PWM10/11/13/14 捕获/比较使能寄存器 (PWMx_CCER)

PWM10/11/13/14 capture/compare enable register

偏移地址：0x20

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												CC1NP	Res.	CC1P	CC1E
												rw		rw	rw

位 15:4 保留，必须保持复位值。

位 3 CC1NP：捕获/比较 1 互补输出极性 (Capture/Compare 1 complementary output polarity)。

CC1 通道配置为输出：CC1NP 必须保持清零。

CC1 通道配置为输入：CC1NP 位与 CC1P 配合使用可定义 TI1FP1 的极性（请参见 CC1P 说明）。

位 2 保留，必须保持复位值。

位 1 CC1P：捕获/比较 1 输出极性 (Capture/Compare 1 output Polarity)。

CC1 通道配置为输出：

0：OC1 高电平有效

1：OC1 低电平有效

CC1 通道配置为输入：

CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的极性。

00：未反相/上升沿触发 电路对 TI1FP1 上升沿敏感（捕获模式），TI1FP1 未反相。

01：反相/下降沿触发 电路对 TI1FP1 下降沿敏感（捕获模式），TI1FP1 反相。

10：保留，不使用此配置

11：未反相/上升沿和下降沿均触发 电路对 TI1FP1 上升沿和下降沿均敏感（捕获模式），TI1FP1 未反相。

位 0 CC1E：捕获/比较 1 输出使能 (Capture/Compare 1 output enable)。

CC1 通道配置为输出：

0：关闭——OC1 未激活

1：开启——在相应输出引脚上输出 OC1 信号

CC1 通道配置为输入：

该位决定了输入捕获/比较寄存器 1 (PWMx_CCR1) 是否能够实际捕获到计数器的值。

0：禁止捕获

1：使能捕获

表 82. 标准 OCx 通道的输出控制位

CCxE 位	OCx 输出状态
0	禁止输出（OCx=“0”，OCx_EN=“0”）
1	OCx=OCxREF + 极性，OCx_EN=“1”

注 意 ： 与 标 准 OCx 通道相连的外部 I/O 引脚的状态取决于通道 OCx 的状态以及 GPIO 寄存器。

17.6.6 PWM10/11/13/14 计数器 (PWMx_CNT)

偏移地址：0x24

复位值：0x0000

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 15:0 CNT[15:0]：计数器值 (Counter value)

17.6.7 PWM10/11/13/14 预分频器 (PWMx_PSC)

偏移地址：0x28

复位值：0x0000

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

位 15:0 PSC[15:0]：预分频器值 (Prescaler value)

计数器时钟频率 CK_CNT 等于 fCK_PSC / (PSC[15:0] + 1)。

PSC 包含在每次发生更新事件时要装载到实际预分频器寄存器的值。

17.6.8 PWM10/11/13/14 自动重载寄存器 (PWMx_ARR)

PWM10/11/13/14 auto-reload register

偏移地址：0x2C

复位值：0x0000

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 15:0 ARR[15:0]：自动重载值 (Auto-reload value)

ARR 为要装载到实际自动重载寄存器的值。

有关 ARR 更新和行为的详细信息，请参见：时基单元 。

当自动重载值为空时计数器不工作。

17.6.9 PWM10/11/13/14 捕获/比较寄存器 1 (PWMx_CCR1)

PWM10/11/13/14 capture/compare register 1

偏移地址：0x34

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 CCR1[15:0]：捕获/比较 1 值 (Capture/Compare 1 value)

如果通道 CC1 配置为输出：

CCR1 为要装载到实际捕获/比较 1 寄存器的值（预装载值）。

如果没有通过 PWMx_CCMR 寄存器中的 OC1PE 位来使能预装载功能，写入的数值会被直接传输至当前寄存器中。否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器 1） 实际捕获/比较寄存器中包含要与计数器 PWMx_CNT 进行比较并在 OC1 输出上发出信号的值。

如果通道 CC1 配置为输入：

CCR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。

17.6.10 PWM11 选项寄存器 1 (PWM11_OR)

PWM11 option register 1

偏移地址：0x50

复位值：0x0000

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

STCMCU

Reserved	TI1_RMP[1:0]	
	rw	rw

位 15:2 保留，必须保持复位值。

位 1:0 TI1_RMP[1:0]：PWM11 输入 1 重映射功能 (PWM11 Input 1 remapping capability)

由软件置 1 和清零。

00,01,11：PWM11 通道 1 连接到 GPIO （请参见数据手册中的复用功能映射表）。

10：HSE_RTC 时钟（由可编程预分频器分频的 HSE）连接到 PWM11_CH1 输入，用于测量用途

17.6.11 PWM10/11/13/14 寄存器映射

PWMx 寄存器可映射为 16 位可寻址寄存器，如下表所述：

表 83. PWM10/11/13/14 寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0X00	TIMx_CR1	Reserved																						CKD [1:0]		ARPE	Res.			OPM	URS	UDIS	CEN
	Reset value																							0	0	0				0	0	0	0
0X04	TIMx_CR2	Reserved																									MMS[2:0]						
	Reset value																										0	0	0				
0X08	TIMx_SMCR	Reserved																								MSM	TS[2:0]			Reserved0	SMS[2:0]		
	Reset value																									0	0	0	0		0	0	0
0X0C	TIMx_DIER	Reserved																									TIE	Res.			CC2IE	CC1IE	UIE
	Reset value																										0				0	0	0
0X10	TIMx_SR	Reserved																						CC2OF	CC1OF	Reserved	BIF	TIF	Res.		CC2IF	CC1IF	UIF
	Reset value																							0	0			0			0	0	0
0X14	TIMx_EGR	Reserved																										TG	Reserved			CC2G	CC1G	UG
	Reset value																											0				0	0	0
0X18	TIMx_CCMR1 Output compare mode	Reserved																						OC2M [2:-0]			OC2PE	OC2FE	CC2S [1:0]		Reserved	OC1M[2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value																							0	0	0	0	0	0	0		0	0	0	0	0	0	0

	TIMx_CCMR1 INput compare mode	Reserved					IC2F[3:0]				IC2 PSC [1:0]		CC2S [1:0		CC2S[1:0]				IC1 PSC [1:0]		CC1S [1:0]	
	Reset value						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0X1C	Reserved																							
0X20	TIMx_CCER	Reserved														CC2NP		CC2P	CC2E	CC1NP		CC1P	CC1E	
	Reset value															0		0	0	0		0	0	
0X24	TIMx_CNT	Reserved					CNT[15:0]																	
	Reset value						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0X28	TIMx_PSC	Reserved					PSC[15:0]																	
	Reset value						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0X2C	TIMx_ARR	Reserved					ARR15:0]																	
	Reset value						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	TIMx_RCR	Reserved													REP[7:0]									
	Reset value														0	0	0	0	0	0	0	0		
0x34	TIMx_CCR1	Reserved					CCR1[15:0]																	
	Reset value						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	Reserved																							
0x3C 0x40 0x44 0x48 0x4C	Reserved																							
0X50	TIMx_OR	Reserved																					TI1RMP	
	Reset value																						0	0

18 独立看门狗 (IWDG)

18.1IWDG 简介

此器件具有两个嵌入式看门狗外设，具有安全性高、定时准确及使用灵活的优点。两个看门狗外设（独立和窗口）均可用于检测并解决由软件错误导致的故障；当计数器达到给定的超时值时，触发一个中断（仅适用于窗口型看门狗）或产生系统复位。

独立看门狗 (IWDG) 由其专用低速时钟 (LSI) 驱动，因此即便在主时钟发生故障时仍然保持工作状态。窗口看门狗 (WWDG) 时钟由 APB1 时钟经预分频后提供，通过可配置的时间窗口来检测应用程序非正常的过迟或过早的操作。

IWDG 最适合应用于那些需要看门狗作为一个在主程序之外，能够完全独立工作，并且对时间精度要求较低的场合。WWDG 最适合那些要求看门狗在精确计时窗口起作用的应用程序。

18.2IWDG 主要特性

- 自由运行递减计数器
- 时钟由独立 RC 振荡器提供（可在待机和停止模式下运行）
- 当递减计数器值达到 0x000 时产生复位（如果看门狗已激活）

18.3IWDG 功能说明

图 198 给出了独立看门狗模块的功能框图。
当通过对关键字寄存器 (IWDG_KR) 写入值 0xCCCC 启动独立看门狗时，计数器开始从复位值 0xFFF 递减计数。当计数器计数到终值 (0x000) 时会产生一个复位信号（IWDG 复位）。任何时候将关键字 0xAAAA 写到 IWWDG_KR 寄存器中，IWDG_RLR 的值就会被重载到计数器，从而避免产生看门狗复位。

18.3.1硬件看门狗

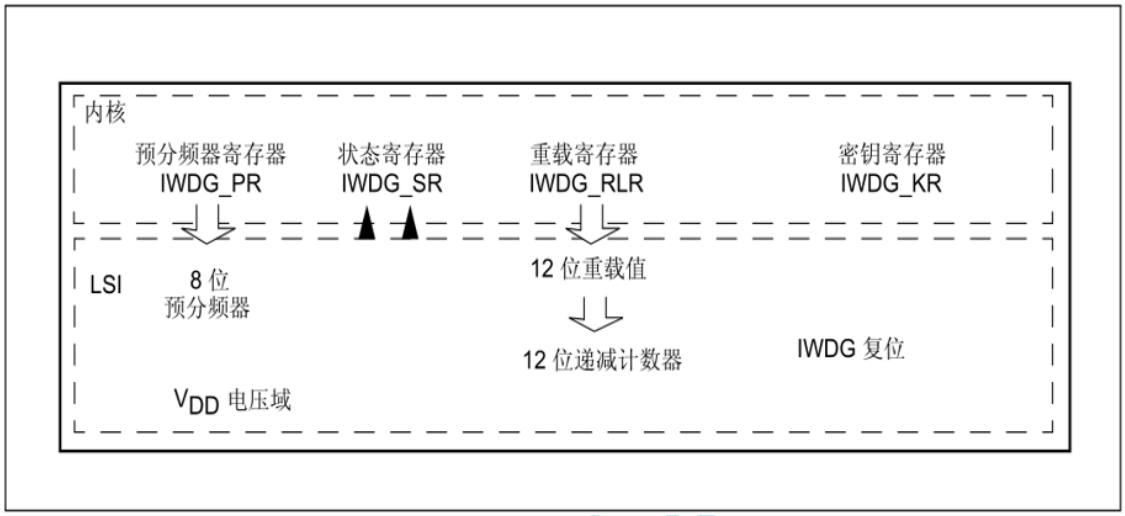
如果通过器件选项位使能“硬件看门狗”功能，上电时将自动使能看门狗；如果在计数器计数结束前，若软件没有向关键字寄存器写入相应的值，则系统会产生复位。

18.3.2寄存器访问保护

IWDG_PR 和 IWDG_RLR 寄存器具有写访问保护。若要修改寄存器，必须首先对 IWDG_KR 寄存器写入代码 0x5555。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。这意味着重装载操作（即写入 0xAAAA）也会启动写保护功能。
状态寄存器指示预分频值和递减计数器是否正在被更新。

18.3.3 调试模式

当微控制器进入调试模式时（Cortex™-MSTAR 内核停止），IWDG 计数器会根据 DBG 模块中的 DBG_IWDG_STOP 配置位选择继续正常工作或者停止工作。
图 198. 独立看门狗框图



看门狗功能由 V DD 电压域供电，在停止模式和待机模式下仍能工作。
表 85. 32 kHz (LSI) 频率条件下 IWDG 超时周期的最小值/ 最大值 (1)

预分频器	PR[2:0]位	最短超时（ms） RL[11:0]=0x000	最长超时（ms） RL[11:0]=0xFFF
/4	0	0.125	512
/8	1	0.25	1024
/16	2	0.5	2048
/32	3	1	4096
/64	4	2	8192
/128	5	4	16384
/256	6	8	32768

1. 这些时间均针对 32 kHz 时钟给出。实际上，MCU 内部的 RC 频率会在 30kHz 到 60kHz 之间变化。此外，即使 RC 振荡器的频率是精确的，确切的时序仍然依赖于 APB 接口时钟与 RC 振荡器时钟之间的相位差，因此总会有一个完整的 RC 周期是不确定的。

18.4 IWDG 寄存器

有关寄存器说明中使用的缩写，请参见第 1.1 节 。
外设寄存器可支持半字（16 位）或字（32 位）访问。

18.4.1 关键字寄存器 (IWDG_KR)

Key register
偏移地址：0x00
复位值：0x0000 0000（通过待机模式复位）

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

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

位 31:16 保留，必须保持复位值。

位 15:0 KEY[15:0] ： 键值 (Key value)（只写位，读为 0000h）

必须每隔一段时间便通过软件对这些位写入键值 AAAAh，否则当计数器计数到 0 时，看门狗会产生复位。

写入键值 5555h 可使能对 IWDG_PR 和 IWDG_RLR 寄存器的访问

写入键值 CCCCh 可启动看门狗（选中硬件看门狗选项的情况除外）

18.4.2预分频器寄存器 (IWDG_PR)

Prescaler register

偏移地址：0x04

复位值：0x0000 0000

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

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

位 31:3 保留，必须保持复位值。

位 2:0 PR[2:0] ： 预分频器 (Prescaler divider)

这些位受写访问保护，请参见 第 19.3.2 节 。通过软件设置这些位来选择计数器时钟的预分频因子。若要更改预分频器的分频系数，IWDG_SR 的 PVU 位必须为 0。

000：4 分频

001：8 分频

010：16 分频

011：32 分频

100：64 分频

101：128 分频

110：256 分频

111：256 分频

注意：读取该寄存器会返回 VDD 电压域的预分频器值。如果正在对该寄存器执行写操作，则读取的值可能不是最新的 / 有效的。因此，只有在 IWDG_SR 寄存器中的 PVU 位为 0 时，从寄存器读取的值才有效。

18.4.3重载寄存器 (IWDG_RLR)

Reload register

偏移地址：0x08

复位值：0x0000 0FFF（待机模式时复位）

STC MCU

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

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

位 31:12 保留，必须保持复位值。

位 11:0 RL[11:0] ：看门狗计数器重载值 (Watchdog counter reload value)

这些位受写访问保护，请参见 第 19.3.2 节 。这个值由软件设置，每次对 IWDR_KR

注意：读取该寄存器会返回 VDD 电压域的重载值。如果正在对该寄存器执行写操作，则读取的值可能不是最新的 / 有效的。因此，只有在 IWDG_SR 寄存器中的 RVU 位为 0 时，从寄存器读取的值才有效。

18.4.4状态寄存器 (IWDG_SR)

Status register

偏移地址：0x0C

复位值：0x0000 0000（待机模式时不复位）

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

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

位 31:2 保留，必须保持复位值。

位 1 RVU ：看门狗计数器重载值更新 (Watchdog counter reload value update)

可通过硬件将该位置 1 以指示重载值正在更新。当在 V DD 电压域下完成重载值更新操作后（需要多达 5 个 RC 40 kHz 周期），会通过硬件将该位复位。

重载值只有在 RVU 位为 0 时才可更新。

位 0 PVU ：看门狗预分频器值更新 (Watchdog prescaler value update)

可通过硬件将该位置 1 以指示预分频器值正在更新。当在 V DD 电压域下完成预分频器值更新操作后（需要多达 5 个 RC 40 kHz 周期），会通过硬件将该位复位。

预分频器值只有在 PVU 位为 0 时才可更新。

注意： 如果应用使用多个重载值或预分频器值，则必须等到 RVU 位被清零后才能更改重载值，而且必须等到 PVU 位被清零后才能更改预分频器值。但是，在更新预分频器和 / 或重载值之后，则无需等到 RVU 或 PVU 复位后再继续执行代码（即便进入低功耗模式，也会继续执行写操作至完成）

18.4.5IWDG 寄存器映射

下表提供了 IWDG 寄存器映射和复位值。

表 86. IWDG

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0X00	IWDG_KR	Reserved																KEY[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0X04	IWDG_PR	Reserved											PR[2:0]		
	Reset value												0	0	0
0X08	IWDG_PLR	Reserved								PL[11:0]					
	Reset value									1	1	1	1	1	0
										1	1	1	1	1	0
0X0C	IWDG_SR	Reserved												RVU	PVU
	Reset value													0	0

STC MCU

19 窗口看门狗 (WWDG)

19.1WWDG 简介

窗口看门狗通常被用来监测，由外部干扰或不可预见的逻辑条件造成的应用程序背离正常的运行序列而产生的软件故障。除非递减计数器的值在 T6 位变成 0 前被刷新，看门狗电路在达到预置的时间周期时，会产生一个 MCU 复位。如果在递减计数器达到窗口寄存器值之前刷新控制寄存器中的 7 位递减计数器值，也会产生 MCU 复位。这意味着必须在限定的时间窗口内刷新计数器。

19.2WWDG 主要特性

- 可编程的自由运行递减计数器
- 复位条件

— 当递减计数器值小于 0x40 时复位（如果看门狗已激活）

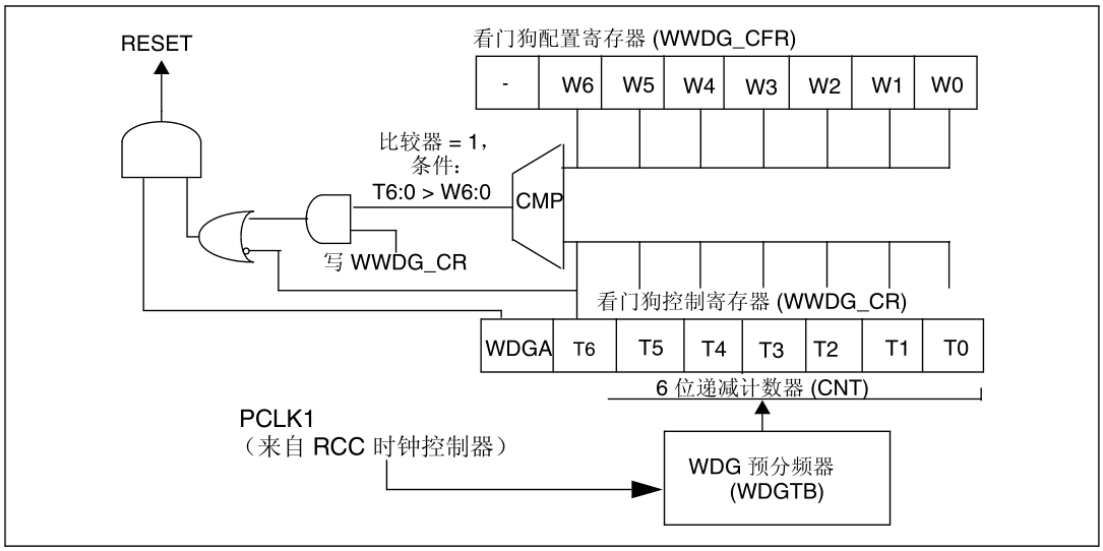
— 在窗口之外重载递减计数器时复位（如果看门狗已激活）（请参见 图 200）

- 提前唤醒中断 (EWI)：当递减计数器等于 0x40 时触发（如果已使能且看门狗已激活）

19.3WWDG 功能说明

如果激活看门狗（WWDG_CR 寄存器中的 WDGA 位置 1），则当 7 位递减计数器（T[6:0]位）从 0x40 滚动到 0x3F（T6 已清零）时会引发复位。当计数器值大于窗口寄存器中所存储的值时，如果软件重载计数器，则会产生复位。

图 199. 看门狗框图



应用程序在正常运行过程中必须定期地写入 WWDG_CR 寄存器以防止 MCU 发生复位。只有当计数器值低于窗口寄存器值时，才能执行此操作。存储在 WWDG_CR 寄存器中的值必须介于 0xFF 和

0xC0 之间：

使能看门狗

在系统复位后，看门狗总是处于关闭状态。可通过设置 WWDG_CR 寄存器中的 WDGA 位来使能看门狗，之后除非执行复位操作，否则不能再次关闭。

控制递减计数器

递减计数器处于自由运行状态：即使禁止看门狗，递减计数器仍继续递减计数。当使能看门狗时，必须将 T6 位置 1，以防止立即复位。

T[5:0] 位包含了看门狗产生复位之前的计时数目；复位前的延时时间在一个最小值和一个最大值之间变化，这是因为写入 WWDG_CR 寄存器时，预分频值是未知的（请参见 图 200）。配置寄存器 (WWDG_CFR) 包含窗口的上限：为防止发生复位，当递减计数器的值低于窗口寄存器值且大于 0x3F 时必须重载。 图 200 介绍了窗口看门狗的工作过程。

注意： 可使用 T6 位产生软件复位（将 WDGA 位置 1 并将 T6 位清零）。

看门狗中断高级特性

如果在产生实际复位之前必须执行特定的安全操作或数据记录，则可使用提前唤醒中断 (EWI)。通过设置 WWDG_CFR 寄存器中的 EWI 位使能 EWI 中断。当递减计数器的值为 0x40 时，将生成 EWI 中断。在复位器件之前，可以使用相应的中断服务程序 (ISR) 来触发特定操作（例如通信或数据记录）。

在某些应用中，可以使用 EWI 中断来管理软件系统检查和/或系统恢复/功能退化，而不会生成 WWDG 复位。在这种情况下，相应的中断服务程序 (ISR) 可用来重载 WWDG 计数器以避免 WWDG 复位，然后再触发所需操作。

通过将 0 写入 WWDG_SR 寄存器中的 EWIF 位来清除 EWI 中断。

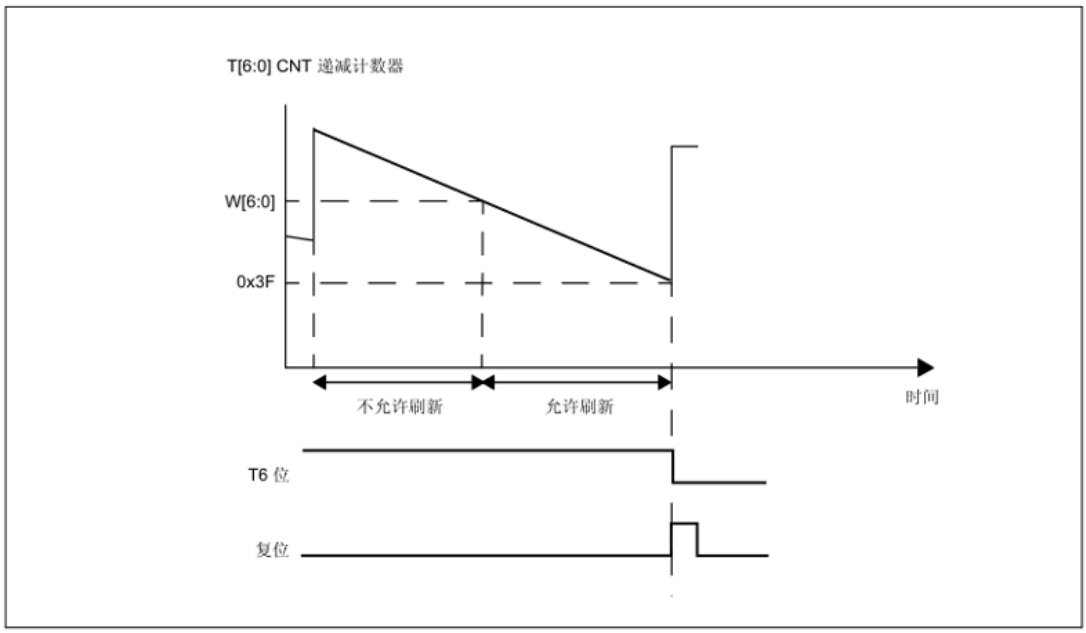
注意：当由于在更高优先级任务中有系统锁定而无法使用 EWI 中断时，最终会产生 WWDG 复位。

19.4如何设置看门狗超时

可以使用 图 200 中的公式来计算 WWDG 超时。

警告： 写入 WWDG_CR 寄存器时，始终将 1 写入 T6 位，以避免生成立即复位。

图 200. 窗口看门狗时序图



超时值的计算公式如下：

$$t_{WWDG} = t_{PCLK1} \times 4096 \times 2^{WDGTB} \times (t[5:0] + 1) \quad (ms)$$

其中：
t WWDG ： WWDG 超时
t PCLK1 ： APB1 时钟周期，以 ms 为测量单位
有关 T WWDG 的最小值和最大值，请参见下表。

表 87. 30 MHz (f PCLK1)

预分频器	**WDGTB**	最小超时（μs） **T[5:0] = 0x00**	最大超时**(ms)** **T[5:0] = 0x3F**
1	0	136.53	8.74
2	1	273.07	17.48
4	2	546.13	34.95
8	3	1092.27	69.91

19.5调试模式

当微控制器进入调试模式时（Cortex™-MSTAR 内核停止），WWDG 计数器会根据 DBG 模块中的 DBG_WWDG_STOP 配置位选择继续正常工作或者停止工作。

19.6WWDG

外设寄存器可支持半字（16 位）或字（32 位）访问。

19.6.1控制寄存器 (WWDG_CR)

Status register
偏移地址：0x00
复位值：0x0000 007F

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

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

位 31:8 保留，必须保持复位值。
位 7 WDGA ： 激活位 (Activation bit)
此位由软件置 1，只有复位后才由硬件清零。当 WDGA = 1 时，看门狗可产生复位。
0:禁止看门狗
1:使能看门狗

位 6:0 T[6:0] ：7 位计数器 (MSB 到 LSB)

这些位用来存储看门狗计数器的值。它每隔 (4096 x 2 WDGTB) PCLK1 个周期递减一次。 当它从 0x40 滚动到 0x3F（T6 清零）时会产生复位。

19.6.2配置寄存器 (WWDG_CFR)

Configuration register

偏移地址：0x04

复位值：0x0000 007F

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

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

位 31:10 保留，必须保持复位值。

位 9 EWI ：提前唤醒中断 (Early wakeup interrupt)

置 1 后，只要计数器值达到 0x40 就会产生中断。此中断只有在复位后才由硬件清零。

位 8:7 WDGTB[1:0] ：定时器时基 (TIMER base)

可按如下方式修改预分频器的时基：

00：CK 计数器时钟 (PCLK1 div 4096) 分频器 1

01：CK 计数器时钟 (PCLK1 div 4096) 分频器 2

10:CK 计数器时钟 (PCLK1 div 4096) 分频器 4

11:CK 计数器时钟 (PCLK1 div 4096) 分频器 8

位 6:0 W[6:0] ：7 位窗口值 (7-bit window value)

这些位包含用于与递减计数器进行比较的窗口值。

19.6.3状态寄存器 (WWDG_SR)

Status register

偏移地址：0x08

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															EWIF
															rc_w0

位 31:1 保留，必须保持复位值。

位 0 EWIF ：提前唤醒中断标志 (Early wakeup interrupt flag)

当计数器值达到 0x40 时此位由硬件置 1。它必须由软件通过写入 0 来清零。写入 1 不起作用。如果不使能中断，此位也会被置 1。

19.6.4 WWDG 寄存器映射

下表提供了 WWDG 寄存器映射和复位值。

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0X00	WWDG_CR	Reserved																							WDGA	T[6:0]							
	Reset value																								0	0	0	0	0	0	0	0	0
0X04	WWDG_CFR	Reserved																						EWI	WDGTB1	WDGTB0	W[6:0]						
	Reset value																							0	0	0	0	0	0	0	0	0	0
0X08	WWDG_SR	Reserved																															EWIF
	Reset value																																0

STC MCU

20 通用同步异步收发器 (USART)

20.1USART 简介

通用同步异步收发器 (USART) 能够灵活地与外部设备进行全双工数据交换，满足外部设备对工业标准 NRZ 异步串行数据格式的要求。USART 通过小数波特率发生器提供了多种波特率。它支持同步单向通信和半双工单线通信；还支持 LIN（局域互连网络）、智能卡协议与 IrDA（红外线数据协会）SIR ENDEC 规范，以及调制解调器操作 (CTS/RTS)。而且，它还支持多处理器通信。
通过配置多个缓冲区使用 DMA 可实现高速数据通信。

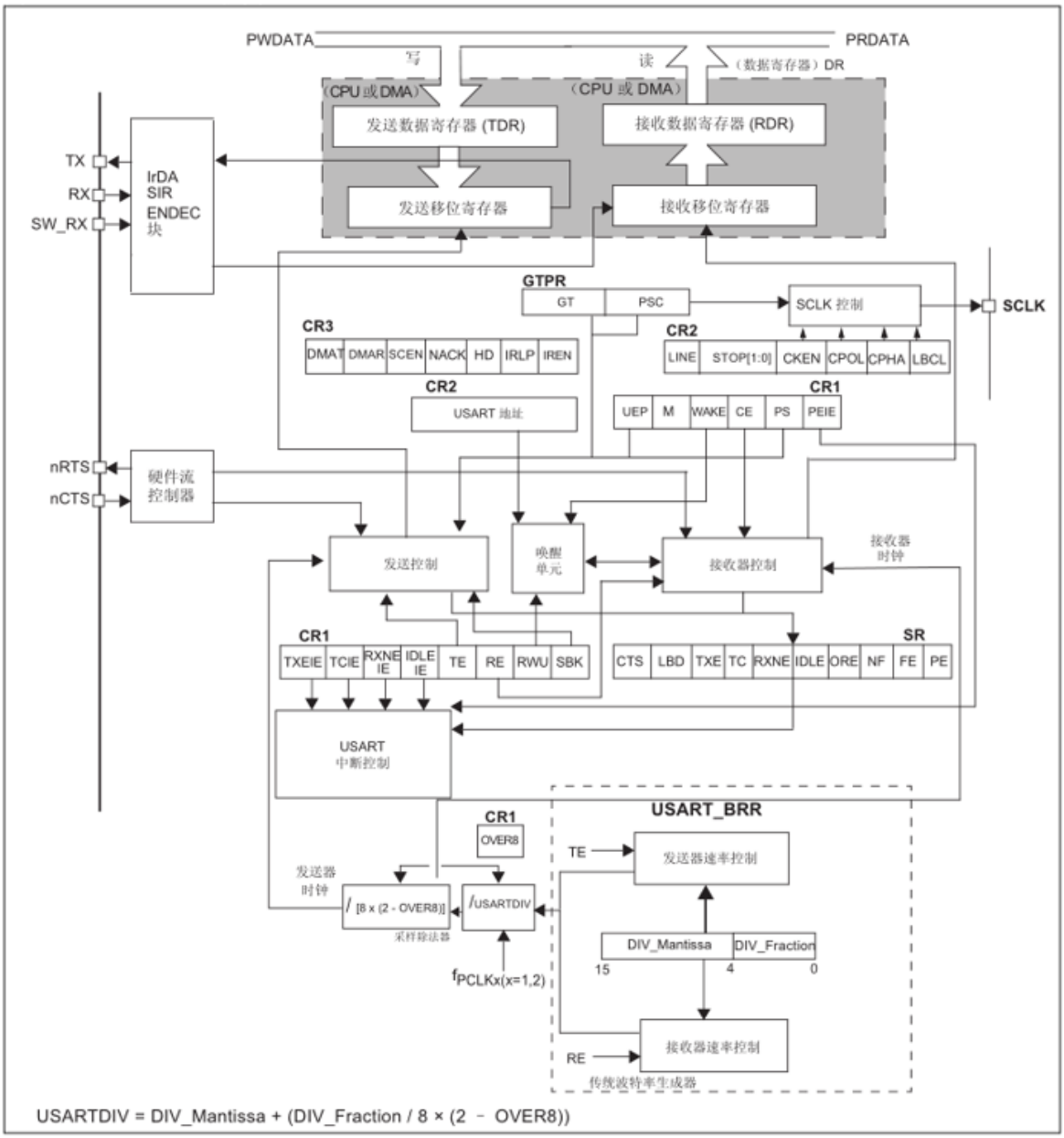
20.2USART 主要特性

- 全双工异步通信
- NRZ 标准格式（标记/空格）
- 可配置为 16 倍过采样或 8 倍过采样，因而为速度容差与时钟容差的灵活配置提供了可能
- 小数波特率发生器系统
  - 通用可编程收发波特率（有关最大 APB 频率时的波特率值，请参见数据手册）。
- 数据字长度可编程（8 位或 9 位）
- 停止位可配置 - 支持 1 或 2 个停止位
- LIN 主模式同步停止符号发送功能和 LIN 从模式停止符号检测功能
  - 对 USART 进行 LIN 硬件配置时可生成 13 位停止符号和检测 10/11 位停止符号
- 用于同步发送的发送器时钟输出
- IrDA SIR 编码解码器
  - 正常模式下，支持 3/16 位持续时间
- 智能卡仿真功能
  - 智能卡接口支持符合 ISO 7816-3 标准中定义的异步协议智能卡
  - 智能卡工作模式下，支持 0.5 或 1.5 个停止位
- 单线半双工通信
- 使用 DMA（直接存储器访问）实现可配置的多缓冲区通信
  - 使用 DMA 在预留的 SRAM 缓冲区中收/发字节
- 发送器和接收器具有单独使能位
- 传输检测标志：
  - 接收缓冲区已满
  - 发送缓冲区为空
  - 传输结束标志
- 奇偶校验控制：
  - 发送奇偶校验位
  - 检查接收的数据字节的奇偶性
- 四个错误检测标志：
  - 溢出错误
  - 噪声检测
  - 帧错误

— 奇偶校验错误
● 十个具有标志位的中断源：
— CTS 变化
— LIN 停止符号检测
— 发送数据寄存器为空
— 发送完成
— 接收数据寄存器已满
— 接收到线路空闲
— 溢出错误
— 帧错误
— 噪声错误
— 奇偶校验错误
● 多处理器通信 - 如果地址不匹配，则进入静默模式
● 从静默模式唤醒（通过线路空闲检测或地址标记检测）
● 两个接收器唤醒模式：地址位（MSB，第 9 位），线路空闲

20.3USART 功能说明

接口通过三个引脚从外部连接到其它设备（请参见 图 246）。任何 USART 双向通信均需要
至少两个引脚：接收数据输入引脚 (RX) 和发送数据引脚输出 (TX)：
RX ：接收数据输入引脚就是串行数据输入引脚。过采样技术可区分有效输入数据和噪声，从而用于恢复数据。
TX ：发送数据输出引脚。如果关闭发送器，该输出引脚模式由其 I/O 端口配置决定。如果使能了发送器但没有待发送的数据，则 TX 引脚处于高电平。在单线和智能卡模式下，该 I/O
用于发送和接收数据（USART 电平下，随后在 SW_RX 上接收数据）。
正常 USART 模式下，通过这些引脚以帧的形式发送和接收串行数据：
● 发送或接收前保持空闲线路
● 起始位
● 数据（字长 8 位或 9 位），最低有效位在前
● 用于指示帧传输已完成的 0.5 个、1 个、1.5 个、2 个停止位
● 该接口使用小数波特率发生器 - 带 12 位尾数和 4 位小数
● 状态寄存器 (USART_SR)
● 数据寄存器 (USART_DR)
● 波特率寄存器 (USART_BRR) - 12 位尾数和 4 位小数。
● 智能卡模式下的保护时间寄存器 (USART_GTPR)。
在同步模式下连接时需要以下引脚：
● SCLK ：发送器时钟输出。该引脚用于输出发送器数据时钟，以便按照 SPI 主模式进行同步发送（起始位和结束位上无时钟脉冲，可通过软件向最后一个数据位发送时钟脉冲）。RX 上可同步接收并行数据。这一点可用于控制带移位寄存器的外设（如 LCD 驱动器）。时钟相位和极性可通过软件编程。在智能卡模式下，SCLK 可向智能卡提供时钟。
在硬件流控制模式下需要以下引脚：
● nCTS ：“清除以发送”用于在当前传输结束时阻止数据发送（高电平时）
● nRTS ：“请求以发送”用于指示 USART 已准备好接收数据（低电平时）。
图 246. USART



20.3.1 USART 字符说明

可通过对 USART_CR1 寄存器中的 M 位进行编程来选择 8 位或 9 位的字长（请参见 图 247）。

TX 引脚在起始位工作期间处于低电平状态。在停止位工作期间处于高电平状态。

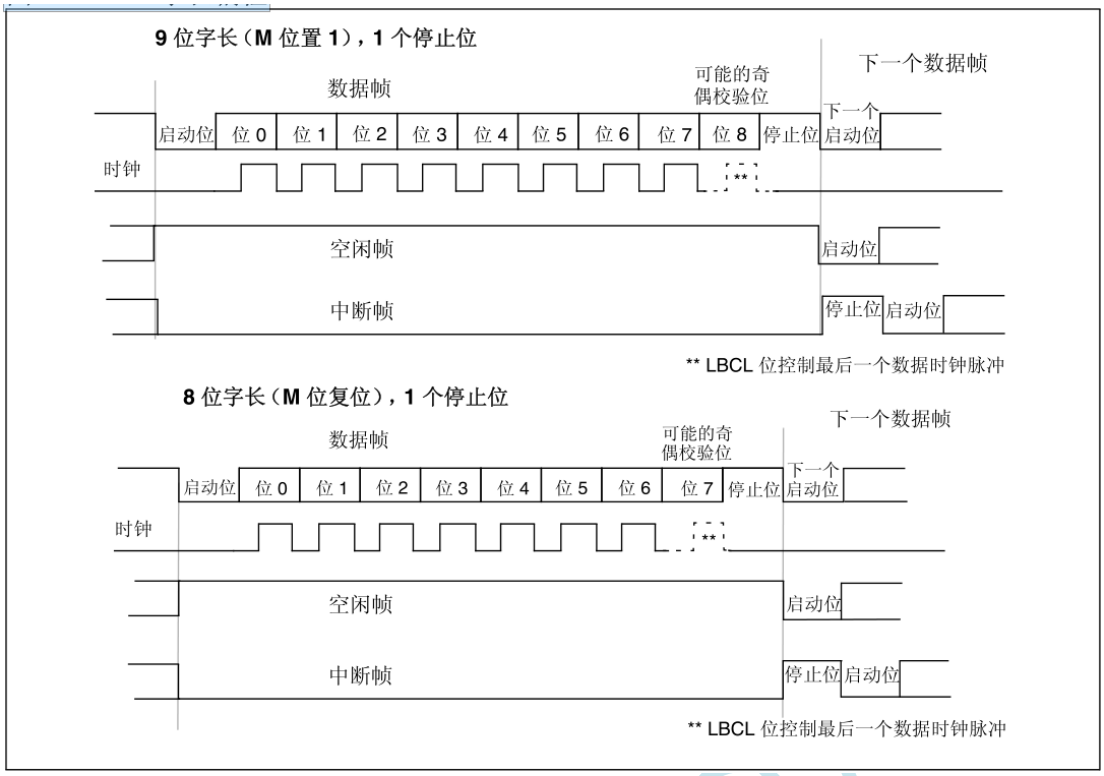
空闲字符 可理解为整个帧周期内电平均为“1”（停止位的电平也是“1”），该字符后是下一个数据帧的起始位。

停止字符 可理解为在一个帧周期内接收到的电平均为“0”。发送器在中断帧的末尾插入 1 或 2 个停止位（逻辑“1”位）以确认起始位。

发送和接收由通用波特率发生器驱动，发送器和接收器的使能位分别置 1 时将生成相应的发送时钟和接收时钟。

下面给出了各个块的详细信息。

图 247. 字长编程



20.3.2 发送器

发送器可发送 8 位或 9 位的数据字，具体取决于 M 位的状态。发送使能位 (TE) 置 1 时，发送移位寄存器中的数据在 TX 引脚输出，相应的时钟脉冲在 SCLK 引脚输出。

字符发送

USART 发送期间，首先通过 TX 引脚移出数据的最低有效位。该模式下，USART_DR 寄存器的缓冲区 (TDR) 位于内部总线和发送移位寄存器之间（请参见 图 246）。

每个字符前面都有一个起始位，其逻辑电平在一个位周期内为低电平。字符由可配置数量的停止位终止。

USART 支持以下停止位：0.5、1、1.5 和 2 个停止位。

注意： 数据发送期间不应复位 TE 位。发送期间复位 TE 位会冻结波特率计数器，从而将损坏 TX 引脚上的数据。当前传输的数据将会丢失。

使能 TE 位后，将会发送空闲帧。

可配置的停止位

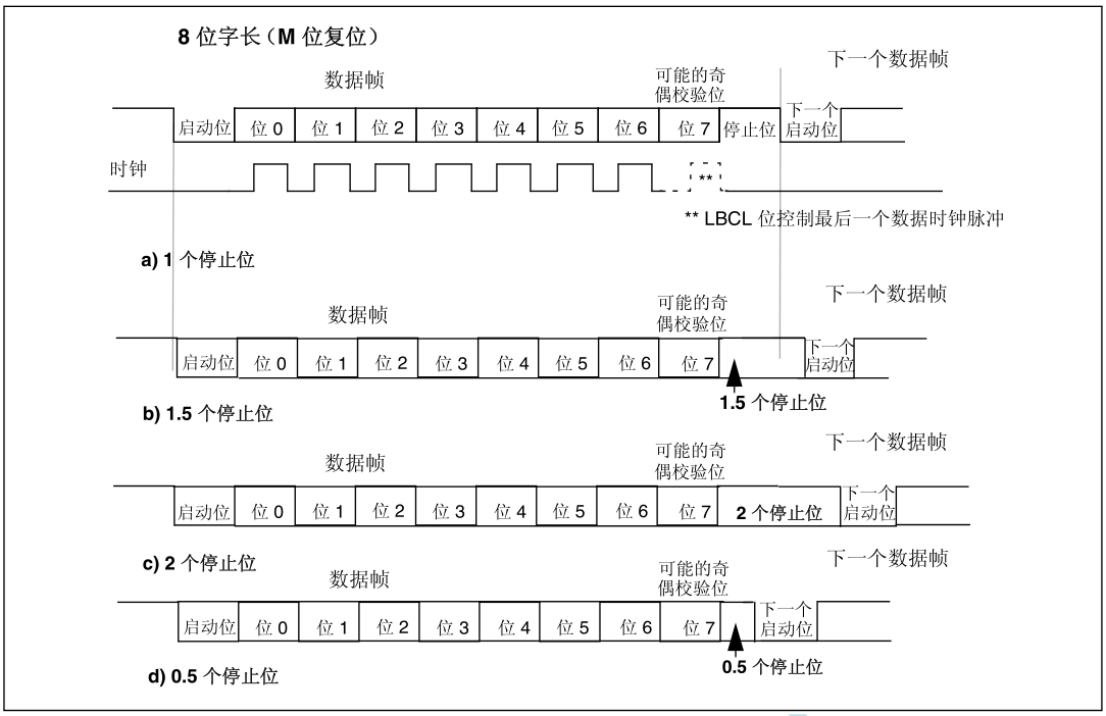
可以在控制寄存器 2 的位 13 和 位 12 中编程将随各个字符发送的停止位的数量。

- 1 个停止位 ：这是停止位数量的默认值。
- 2 个停止位 ：正常 USART 模式、单线模式和调制解调器模式支持该值。
- 0.5 个停止位 ：在智能卡模式下接收数据时使用。
- 1.5 个停止位 ：在智能卡模式下发送和接收数据时使用。

空闲帧发送将包括停止位。

m = 0 时，中断发送是 10 个低电平位，然后是已配置数量的停止位；m = 1 时，中断发送是 11 个低电平位，然后是已配置数量的停止位。无法传送长中断（中断长度大于 10/11 个低电平位）。

图 248. 可配置的停止位



步骤：

1. 通过向 USART_CR1 寄存器中的 UE 位写入 1 使能 USART。
2. 对 USART_CR1 中的 M 位进行编程以定义字长。
3. 对 USART_CR2 中的停止位数量进行编程。
4. 如果将进行多缓冲区通信，请选择 USART_CR3 中的 DMA 使能 (DMAT)。按照多缓冲区通信中的解释说明配置 DMA 寄存器。
5. 使用 USART_BRR 寄存器选择所需波特率。
6. 将 USART_CR1 中的 TE 位置 1 以便在首次发送时发送一个空闲帧。
7. 在 USART_DR 寄存器中写入要发送的数据（该操作将清零 TXE 位）。为每个要在单缓冲区模式下发送的数据重复这一步骤。
8. 向 USART_DR 寄存器写入最后一个数据后，等待至 TC=1。这表明最后一个帧的传送已完成。禁止 USART 或进入暂停模式时需要此步骤，以避免损坏最后一次发送。

单字节通信

始终通过向数据寄存器写入数据来将 TXE 位清零。

TXE 位由硬件置 1，它表示：

- 数据已从 TDR 移到移位寄存器中且数据发送已开始。
- TDR 寄存器为空。
- USART_DR 寄存器中可写入下一个数据，而不会覆盖前一个数据。

TXEIE 位置 1 时该标志位会生成中断。

发送时，要传入 USART_DR 寄存器的写指令中存有 TDR 寄存器中的数据，该数据将在当前发送结束时复制到移位寄存器中。

未发送时，要传入 USART_DR 寄存器的写指令直接将数据置于移位寄存器中，数据发送开始时，TXE 位立即置 1。

如果帧已发送（停止位后）且 TXE 位置 1，TC 位将变为高电平。如果 USART_CR1 寄存器中的 TCIE 位置 1，将生成中断。

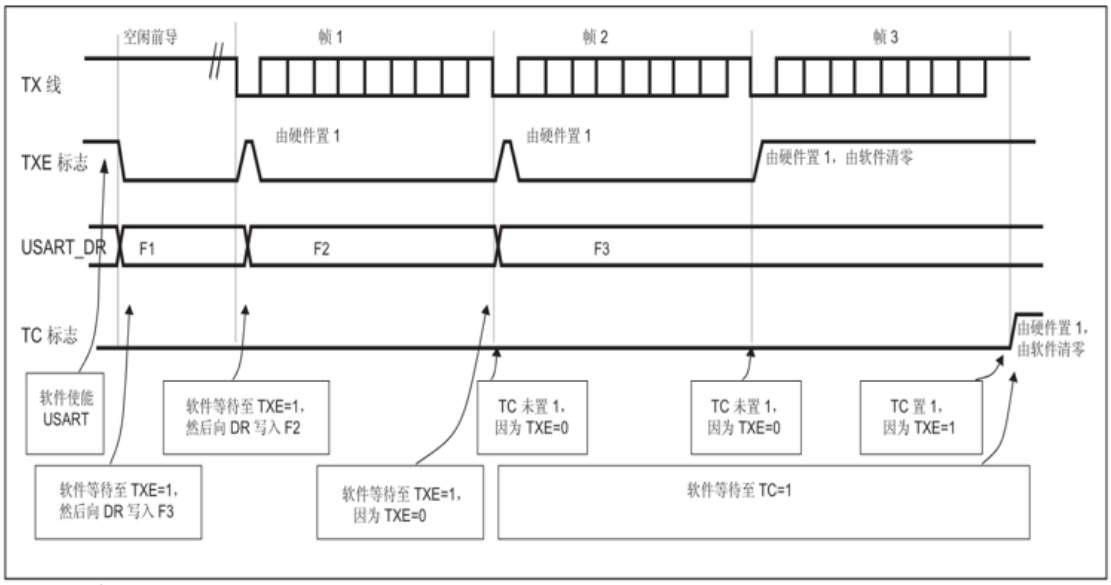
向 USART_DR 寄存器中写入最后一个数据后，必须等待至 TC=1，之后才可禁止 USART 或使微控制器进入低功率模式（请参见 图 249 ：发送时的 TC/TXE 行为 ）。

TC 位通过以下软件序列清零：

1. 从 USART_SR 寄存器读取数据
2. 向 USART_DR 寄存器写入数据

注意： 还可通过向 TC 位写入“ 0 ”将其清零。建议仅在多缓冲区通信时使用此清零序列。

图 249. 发送时的 TC/TXE 行为



中断字符

将 SBK 位置 1 将发送一个中断字符。中断帧的长度取决于 M 位（请参见 图 247）。

如果 SBK 位置“1”，当前字符发送完成后，将在 TX 线路上发送一个中断字符。中断字符发送完成时（发送中断字符的停止位期间），该位由硬件复位。USART 在上一个中断帧的末尾插入一个逻辑“1”位，以确保识别下个帧的起始位。

注意： 如果软件在中断发送开始前对 SBK 位进行了复位，将不会发送中断字符。对于两个连续的中断，应在上一个中断的停止位发送完成后将 SBK 位置 1 。

空闲字符

将 TE 位置 1 会驱动 USART 在第一个数据帧之前发送一个空闲帧。

20.3.3接收器

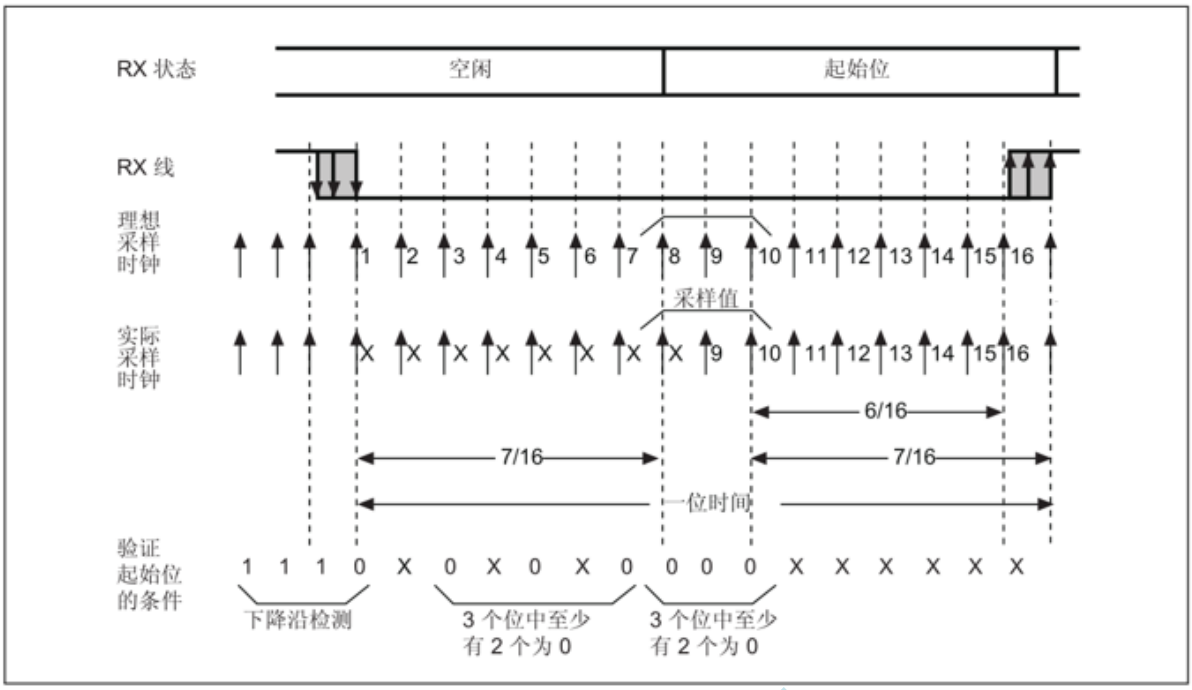
USART 可接收 8 位或 9 位的数据字，具体取决于 USART_CR1 寄存器中的 M 位。

起始位检测

16 倍或 8 倍过采样时，起始位检测序列相同。

在 USART 中，识别出特定序列的采样时会检测起始位。该序列为：1 1 1 0 X 0 X 0 X 0 0 0 0。

图 250. 16 倍或 8 倍过采样时的起始位检测



注意： 如果序列不完整，起始位检测将中止，接收器将返回空闲状态（无标志位置 1 ）等待下降沿。如果 3 个采样位均为 0 （针对第 3 位、第 5 位和第 7 位进行首次采样时检测到这 3 位均为 0 ；针对第 8 位、第 9 位和第 10 位进行第二次采样时检测到这 3 位均为 0 ），可确认起始位（ RXNE 标志位置 1 ， RXNEIE=1 时生成中断）。

如果两次采样时（对第 3 位、第 5 位和第 7 位进行采样以及对第 8 位、第 9 位和第 10 位进行采样）， 3 个采样位中至少有 2 个为 0 ，则可验证起始位（ RXNE 标志位置 1 ， RXNEIE=1 时生成中断）但 NE 噪声标志位置 1 。如果不满足此条件，则启动检测中止，接收器返回空闲状态（无标志位置 1 ）。

如果其中一次采样时（对第 3 位、第 5 位和第 7 位进行采样或对第 8 位、第 9 位和第 10 位进行采样）， 3 个采样位中有 2 个为 0 ，则可验证起始位但 NE 噪声标志位置 1 。

字符接收

USART 接收期间，首先通过 RX 引脚移入数据的最低有效位。该模式下，USART_DR 寄存器的缓冲区 (RDR) 位于内部总线和接收移位寄存器之间。

步骤：

1. 通过向 USART_CR1 寄存器中的 UE 位写入 1 使能 USART。
2. 对 USART_CR1 中的 M 位进行编程以定义字长。
3. 对 USART_CR2 中的停止位数量进行编程。
4. 如果将进行多缓冲区通信，请选择 USART_CR3 中的 DMA 使能 (DMAR)。按照多缓冲区通信中的解释说明配置 DMA 寄存器。步骤 3
5. 使用波特率寄存器 USART_BRR 选择所需波特率
6. 将 RE 位 USART_CR1 置 1。这一操作将使能接收器开始搜索起始位。

接收到字符时

- RXNE 位置 1。这表明移位寄存器的内容已传送到 RDR。也就是说，已接收到并可读取数据（以及其相应的错误标志）。
- 如果 RXNEIE 位置 1，则会生成中断。
- 如果接收期间已检测到帧错误、噪声错误或上溢错误，错误标志位可置 1。
- 在多缓冲区模式下，每接收到一个字节后 RXNE 均置 1，然后通过 DMA 对数据寄存器执行读操作清零。
- 在单缓冲区模式下，通过软件对 USART_DR 寄存器执行读操作将 RXNE 位清零。RXNE 标志也可以通过向该位写入零来清零。RXNE 位必须在结束接收下一个字符前清零，以避免发生上溢错误。

注意： 接收数据时，不应将 RE 位复位。如果接收期间禁止了 RE 位，则会中止接收当前字节。

中断字符
接收到中断字符时，USART 将会按照帧错误对其进行处理。
空闲字符
检测到空闲帧时，处理步骤与接收到数据的情况相同；如果 IDLEIE 位为 1，则会产生中断。
上溢错误
如果在 RXNE 未复位时接收到字符，则会发生上溢错误。RXNE 位清零前，数据无法从移位寄存器传送到 RDR 寄存器。
每接收到一个字节后，RXNE 标志位都将置 1。当 RXNE 标志位是 1 时，如果在接收到下一个数据或尚未处理上一个 DMA 请求时，则会发生上溢错误。发生上溢错误时：
● ORE 位置 1。
● RDR 中的内容不会丢失。对 USART_DR 执行读操作时可使用先前的数据。
● 移位寄存器将被覆盖。之后，上溢期间接收到的任何数据都将丢失。
● 如果 RXNEIE 位置 1 或 EIE 与 DMAR 位均为 1，则会生成中断。
● 通过先后对 USART_SR 寄存器和 USART_DR 寄存器执行读操作将 ORE 位清除。
注意： ORE 位置 1 时表示至少 1 个数据丢失。存在两种可能：
● 如果 RXNE=1 ，则最后一个有效数据存储于接收寄存器 RDR 中并且可进行读取；
● 如果 RXNE=0 ，则表示最后一个有效数据已被读取，因此 RDR 中没有要读取的数据。接收到新（丢失）数据的同时已读取 RDR 中的最后一个有效数据时，会发生该情况。读取序列期间（在 USART_SR 寄存器读访问与 USART_DR 读访问之间）接收到新数据时也会发生该情况。
选择合适的过采样方法
接收器采用不同的用户可配置过采样技术（除了同步模式下），可以从噪声中提取有效数据。可通过编程 USART_CR1 寄存器中的 OVER8 位来选择采样方法，且采样时钟可以是波特率时钟的 16 倍或 8 倍（请参见 图 251 和 图 252）。
根据应用：
● 选择 8 倍过采样 (OVER8=1) 以获得更高的速度（高达 f PCLK /8）。这种情况下接收器对时钟偏差的最大容差将会降低
● 选择 16 倍过采样 (OVER8=0) 以增加接收器对时钟偏差的容差。这种情况下，最大速度限制为最高 f PCLK /16
可通过编程 USART_CR3 寄存器中的 ONEBIT 位选择用于评估逻辑电平的方法。有两种选择：
● 在已接收位的中心进行三次采样，从而进行多数表决。这种情况下，如果用于多数表决的 3 次采样结果不相等，NF 位置 1。
● 在已接收位的中心进行单次采样
根据应用：
— 在噪声环境下工作时，请选择三次采样的多数表决法 (ONEBIT=0)；在检测到噪声时请拒绝数据（请参见 图 107），因为这表示采样过程中产生了干扰。
— 线路无噪声时请选择单次采样法 (ONEBIT=1) 以增加接收器对时钟偏差的容差
这种情况下
NF 位始终不会置 1。
帧中检测到噪声时：
● 在 RXNE 位的上升沿时 NF 位置 1。
● 无效数据从移位寄存器传送到 USART_DR 寄存器。
● 单字节通信时无中断产生。然而，在 RXNE 位产生中断时，该位出现上升沿。多缓冲区通信时，USART_CR3 寄存器中的 EIE 位置 1 时将发出中断。
通过先后对 USART_SR 寄存器和 USART_DR 寄存器执行读操作将 NF 位清零。
注意： 智能卡、 IrDA 和 LIN 模式下不可采用 8 倍过采样。在这些模式下， OVER8 位由硬件强制清零。
图 251. 16 倍过采样时的数据采样

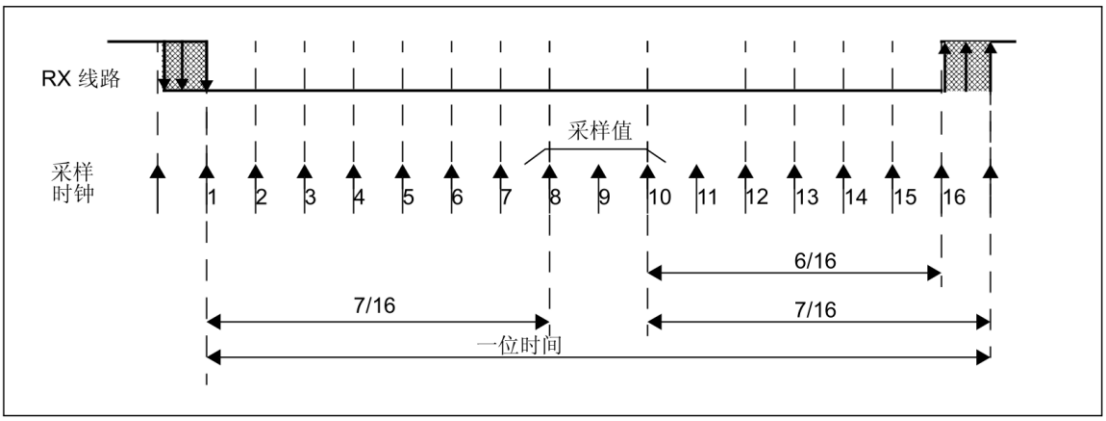


图 252. 8 倍过采样时的数据采样

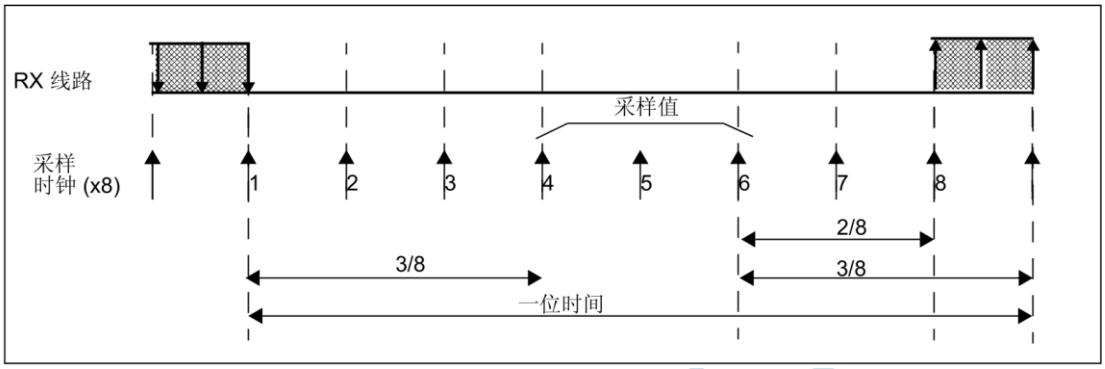


表 107. 通过采样数据进行噪声检测

采样值	NE 状态	接收的位值
000	0	0
001	1	0
010	1	0
011	1	1
100	1	0
101	1	1
110	1	1
111	0	1

帧错误

以下情况下将检测到帧错误：

接收数据时未在预期时间内识别出停止位，从而出现同步失效或过度的噪声。

检测到帧错误时：

- FE 位由硬件置 1
- 无效数据从移位寄存器传送到 USART_DR 寄存器。
- 单字节通信时无中断产生。然而，在 RXNE 位产生中断时，该位出现上升沿。多缓冲区通信时，USART_CR3 寄存器中的 EIE 位置 1 时将发出中断。

通过先后对 USART_SR 寄存器和 USART_DR 寄存器执行读操作将 FE 位清零。

接收期间可配置的停止位

可通过控制寄存器 2 中的控制位配置要接收的停止位的数量 - 可以是 1 或 2 个（正常模式下），也可以是 0.5 或 1.5 个（智能卡模式下）。

1. 0.5 个停止位（在智能卡模式下接收时）：不会对 0.5 个停止位进行采样。结果，选择 0.5 个停止位时，

无法检测到帧错误和中断帧。

2. 1 个停止位 ：将在第 8、第 9 和第 10 次采样时对 1 个停止位进行采样。

3. 1.5 个停止位（在智能卡模式下） ：在智能卡模式下发送时，设备必须检查数据是否正确发送。因此必须使能接收器块（USART_CR1 寄存器中的 RE =1）并检查停止位，以测试智能卡是否已检测到奇偶校验错误。发生奇偶校验错误时，智能卡会在采样时将数据信号强制为低电平，即 NACK 信号-，该信号被标记为帧错误。之后，FE 标志在 1.5 个停止位的末尾由 RXNE 置 1。在第 16、第 17 和第 18 次采样时对 1.5 个停止位进行采样（停止位采样开始后维持 1 个波特时钟周期）。1.5 个停止位可分为 2 个部分：0.5 个波特时钟周期（未发生任何动作），然后是 1 个正常的停止位周期（一半时间处进行采样）。

4. 2 个停止位 ：采样 2 个停止位时在第 8、第 9 和第 10 次采样时对第一个停止位进行采样。如果在第一个停止位期间检测到帧错误，则帧错误标志位将会置 1。发生帧错误时不检测第 2 个停止位。RXNE 标志将在第一个停止位末尾时置 1。

20.3.4 小数波特率生成

对 USARTDIV 的尾数值和小数值进行编程时，接收器和发送器（Rx 和 Tx）的波特率均设置为相同值。

公式 1 ：适用于标准 USART （包括 SPI 模式）的波特率

$$\text{Tx/Rx 波特率} = \frac{f_{CK}}{8 \times (2 - OVER8) \times USARTDIV}$$

公式 2 ：智能卡、LIN 和 IrDA 模式下的波特率

$$\text{Tx/Rx 波特率} = \frac{f_{CK}}{16 \times USARTDIV}$$

USARTDIV 是一个存放在 USART_BRR 寄存器中的无符号定点数。

- 当 OVER8=0 时，小数部分编码为 4 位并通过 USART_BRR 寄存器中的 DIV_fraction[3:0] 位编程。
- 当 OVER8=1 时，小数部分编码为 3 位并通过 USART_BRR 寄存器中的 DIV_fraction[2:0] 位编程，此时 DIV_fraction[3] 位必须保持清零状态。

注意： 对 USART_BRR 执行写操作后，波特率计数器更新为波特率寄存器中的新值。因此，波特率寄存器的值不应在通信时发生更改。

OVER8=0 时如何从 USART_BRR 寄存器值中获取 USARTDIV

示例 1：

如果 DIV_Mantissa = 0d27 且 DIV_Fraction = 0d12 (USART_BRR = 0x1BC)，则

尾数 (USARTDIV) = 0d27

小数 (USARTDIV) = 12/16 = 0d0.75

因此 USARTDIV = 0d27.75

示例 2：

要设定 USARTDIV = 0d25.62

这将导致：

DIV_Fraction = 16*0d0.62 = 0d9.92

最接近的实数为 0d10 = 0xA

DIV_Mantissa = 尾数 (0d25.620) = 0d25 = 0x19

则 USART_BRR = 0x19A，因此 USARTDIV = 0d25.625

示例 3：
要设定 USARTDIV = 0d50.99
这将导致：
DIV_Fraction = 16*0d0.99 = 0d15.84
最接近的实数为 0d16 = 0x10 => DIV_frac[3:0] 溢出 => 尾数必须添加进位
DIV_Mantissa = 尾数（0d50.990 + 进位）= 0d51 = 0x33
则 USART_BRR = 0x330，因此 USARTDIV = 0d51.000
OVER8=1 时如何从 USART_BRR 寄存器中获取 USARTDIV
示例 1 ：
如果 DIV_Mantissa = 0x27 且 DIV_Fraction[2:0]= 0d6 (USART_BRR = 0x1B6)，则
尾数 (USARTDIV) = 0d27
小数 (USARTDIV) = 6/8 = 0d0.75
因此 USARTDIV = 0d27.75

示例 2：
要设定 USARTDIV = 0d25.62
这将导致：
DIV_Fraction = 8*0d0.62 = 0d4.96
最接近的实数为 0d5 = 0x5
DIV_Mantissa = 尾数 (0d25.620) = 0d25 = 0x19
则 USART_BRR = 0x195 => USARTDIV = 0d25.625
示例 3：
要设定 USARTDIV = 0d50.99
这将导致：
DIV_Fraction = 8*0d0.99 = 0d7.92
最接近的实数为 0d8 = 0x8 => DIV_frac[2:0] 溢出 => 尾数必须添加进位
DIV_Mantissa = 尾数（0d50.990 + 进位）= 0d51 = 0x33
则 USART_BRR = 0x0330 => USARTDIV = 0d51.000
表 108. 采用 16 倍过采样时，在 f PCLK = 8 MHz 或 f PCLK = 12 MHz 下编程波特率时的误差计算 (1)

16 倍过采样时（OVER8=0）

波特率		**fpCLK = 8 MHz**			**f_{PCLK}=12MHz**		
序号	所需值	实际值	波特率寄存器中编程的值	误差% =（计算值 - 所需值）/ 所需波特率	实际值	波特率寄存器 中编程的值	误差%
1	1.2 KBps	1.2 KBps	416.6875	0	1.2 KBps	625	0
2	2.4 KBps	2.4 KBps	208.3125	0.01	2.4 KBps	312.5	0
3	9.6 KBps	9.604 KBps	52.0625	0.04	9.6 KBps	78.125	0
4	19.2 KBps	19.185 KBps	26.0625	0.08	19.2 KBps	39.0625	0
5	38.4 KBps	38.462 KBps	13	0.16	38.339 KBps	19.5625	0.16
6	57.6 KBps	57.554 KBps	8.6875	0.08	57.692 KBps	13	0.16
7	115.2 KBps	115.942 KBps	4.3125	0.64	115.385 KBps	6.5	0.16
8	230.4 KBps	228.571 KBps	2.1875	0.79	230.769 KBps	3.25	0.16
9	460.8 KBps	470.588 KBps	1.0625	2.12	461.538 KBps	1.625	0.16
10	921.6 KBps	NA	NA	NA	NA	NA	NA

STC MCU

11	2 MBps	NA	NA	NA	NA	NA	NA
12	3 MBps	NA	NA	NA	NA	NA	NA

1. CPU 时钟越低，特定波特率的精度就越低。可达到的波特率的上限可使用这些数据进行确定。

表 109. 采用 8 倍过采样时，在 f PCLK = 8 MHz 或 f PCLK = 12 MHz 下编程波特率时的误差计算 (1)

8 倍过采样(0VER8 = 1)							
波特率		fpCLK = 8 MHz			fpCLK = 12 MHz		
序号	**所需值**	**实际值**	**波特率寄存器 中编程的值**	**误差% =(计算 值-所需值)/ 所需波特率**	**实际值**	**波特率寄存器中编程的值**	**误差%**
1	1.2 KBps	1.2 KBps	833.375	0	1.2 KBps	1250	0
2	2.4 KBps	2.4 KBps	416.625	0.01	2.4 KBps	625	0
3	9.6 KBps	9.604 KBps	104.125	0.04	9.6 KBps	156.25	0
4	19.2 KBps	19.185 KBps	52.125	0.08	19.2 KBps	78.125	0
5	38.4 KBps	38.462 KBps	26	0.16	38.339 KBps	39.125	0.16
6	57.6 KBps	57.554 KBps	17.375	0.08	57.692 KBps	26	0.16
7	115.2 KBps	115.942 KBps	8.625	0.64	115.385 KBps	13	0.16
8	230.4 KBps	228.571 KBps	4.375	0.79	230.769 KBps	6.5	0.16
9	460.8 KBps	470.588 KBps	2.125	2.12	461.538 KBps	3.25	0.16
10	921.6 KBps	888.889 KBps	1.125	3.55	923.077 KBps	1.625	0.16
11	2 MBps	NA	NA	NA	NA	NA	NA
12	3 MBps	NA	NA	NA	NA	NA	NA

1. CPU 时钟越低，特定波特率的精度就越低。可达到的波特率的上限可使用这些数据进行确定。

表 110. 采用 16 倍过采样时，在 f PCLK = 16 MHz 或 f PCLK = 24 MHz 下编程波特率时的误差计算 (1)

16 倍过采样(0VER8 = 0)							
波特率		**fpcLK = 16 MHz**			**fpcLK = 24 MHz**		
序号	所需值	实际值	波特率寄存器中编程的值	误差% =(计算 值-所需值)/ 所需波特率	实际值	波特率寄存器 中编程的值	误差%

1	1.2 KBps	1.2 KBps	833.3125	0	1.2	1250	0
2	2.4 KBps	2.4 KBps	416.6875	0	2.4	625	0
3	9.6 KBps	9.598 KBps	104.1875	0.02	9.6	156.25	0
4	19.2 KBps	19.208 KBps	52.0625	0.04	19.2	78.125	0
5	38.4 KBps	38.369 KBps	26.0625	0.08	38.4	39.0625	0
6	57.6 KBps	57.554 KBps	17.375	0.08	57.554	26.0625	0.08
7	115.2 KBps	115.108 KBps	8.6875	0.08	115.385	13	0.16
8	230.4 KBps	231.884 KBps	4.3125	0.64	230.769	6.5	0.16
9	460.8 KBps	457.143 KBps	2.1875	0.79	461.538	3.25	0.16
10	921.6 KBps	941.176 KBps	1.0625	2.12	923.077	1.625	0.16

表 110. 采用 16 倍过采样时，在 f PCLK = 16 MHz 或 f PCLK = 24 MHz 下编程波特率时的误差计算 (1)

16 倍过采样(OVER8 = 0)							
波特率		**fpci_K = 16 MHz**			**fpcLK = 24 MHz**		
序号	所需值	实际值	波特率寄存器 中编程的值	误差% =（计算 值-所需值）/ 所需波特率	实际值	波特率寄存器 中编程的值	误差%
11	2 MBps	NA	NA	NA	NA	NA	NA
12	3 MBps	NA	NA	NA	NA	NA	NA

1. CPU 时钟越低，特定波特率的精度就越低。可达到的波特率的上限可使用这些数据进行确定。

表 111. 采用 8 倍过采样时，在 f PCLK = 16 MHz 或 f PCLK = 24 MHz 下编程波特率时的误差计算 (1)

8 倍过采样（0VER8=1）							
波特率		**fpCLK = 16 MHz**			**fpCLK = 24 MHz**		
序号	所需值	实际值	波特率寄存器中编程的值	误差% =（计算值・所需值）/ 所需波特率	实际值	波特率寄存器 中编程的值	误差%
1	1.2 KBps	1.2 KBps	1666.625	0	1.2 KBps	2500	0
2	2.4 KBps	2.4 KBps	833.375	0	2.4 KBps	1250	0
3	9.6 KBps	9.598 KBps	208.375	0.02	9.6 KBps	312.5	0
4	19.2 KBps	19.208 KBps	104.125	0.04	19.2 KBps	156.25	0
5	38.4 KBps	38.369 KBps	52.125	0.08	38.4 KBps	78.125	0
6	57.6 KBps	57.554 KBps	34.75	0.08	57.554 KBps	52.125	0.08
7	115.2 KBps	115.108 KBps	17.375	0.08	115.385 KBps	26	0.16

8	230.4 KBps	231.884 KBps	8.625	0.64	230.769 KBps	13	0.16
9	460.8 KBps	457.143 KBps	4.375	0.79	461.538 KBps	6.5	0.16
10	921.6 KBps	941.176 KBps	2.125	2.12	923.077 KBps	3.25	0.16
11	2 MBps	2000 KBps	1	0	2000 KBps	1.5	0
12	3 MBps	NA	NA	NA	3000 KBps	1	0

1. CPU 时钟越低，特定波特率的精度就越低。可达到的波特率的上限可使用这些数据进行确定。

表 112. 采用 16 倍过采样时，在 f PCLK = 8 MHz 或 f PCLK = 16 MHz 下编程波特率时的误差计算 (1)

16 倍过采样时（OVER8=0）							
波特率		fpCLK = 8 MHz			fpCLK = 16 MHz		
序号	所需值	实际值	波特率寄存器中编程的值	误差%=（计算值・所需值）/所需波特率	实际值	波特率寄存器 中编程的值	误差%
1	**2.4 KBps**	2.400 KBps	208.3125	0.00%	2.400 KBps	416.6875	0.00%
2	**9.6 KBps**	9.604 KBps	52.0625	0.04%	9.598 KBps	104.1875	0.02%
3	**19.2 KBps**	19.185 KBps	26.0625	0.08%	19.208 KBps	52.0625	0.04%
4	**57.6 KBps**	57.554 KBps	8.6875	0.08%	57.554 KBps	17.3750	0.08%
5	**115.2 KBps**	115.942 KBps	4.3125	0.64%	115.108 KBps	8.6875	0.08%
6	**230.4 KBps**	228.571 KBps	2.1875	0.79%	231.884 KBps	4.3125	0.64%
7	**460.8 KBps**	470.588 KBps	1.0625	2.12%	457.143 KBps	2.1875	0.79%
8	**896 KBps**	NA	NA	NA	888.889 KBps	1.1250	0.79%
9	**921.6 KBps**	NA	NA	NA	941.176 KBps	1.0625	2.12%
10	**1.792 MBps**	NA	NA	NA	NA	NA	NA
11	**1.8432 MBps**	NA	NA	NA	NA	NA	NA
12	**3.584 MBps**	NA	NA	NA	NA	NA	NA
13	**3.6864 MBps**	NA	NA	NA	NA	NA	NA
14	**7.168 MBps**	NA	NA	NA	NA	NA	NA
15	**7.3728 MBps**	NA	NA	NA	NA	NA	NA

1. CPU 时钟越低，特定波特率的精度就越低。可达到的波特率的上限可使用这些数据进行确定。

表 113. 采用 8 倍过采样时，在 f PCLK = 8 MHz 或 f PCLK = 16 MHz 下编程波特率时的误差计算 (1)

8 倍过采样(0VER8=1)							
波特率		fpCLK = 8 MHz			fpCLK = 16 MHz		
序号	所需值	实际值	波特率寄存器 中编程的值	误差% =（计算值・所需值）/所需波特率	实际值	波特率寄存器 中编程的值	误差%
1	2.4 KBps	2.400 KBps	416.625	0.01%	2.400 KBps	833.375	0.00%

2	9.6 KBps	9.604 KBps	104.125	0.04%	9.598 KBps	208.375	0.02%
3	19.2 KBps	19.185 KBps	52.125	0.08%	19.208 KBps	104.125	0.04%
4	57.6 KBps	57.557 KBps	17.375	0.08%	57.554 KBps	34.750	0.08%
5	115.2 KBps	115.942 KBps	8.625	0.64%	115.108 KBps	17.375	0.08%
6	230.4 KBps	228.571 KBps	4.375	0.79%	231.884 KBps	8.625	0.64%
7	460.8 KBps	470.588 KBps	2.125	2.12%	457.143 KBps	4.375	0.79%
8	896 KBps	888.889 KBps	1.125	0.79%	888.889 KBps	2.250	0.79%
9	921.6 KBps	888.889 KBps	1.125	3.55%	941.176 KBps	2.125	2.12%
10	1.792 MBps	NA	NA	NA	1.7777 MBps	1.125	0.79%

表 113. 采用 8 倍过采样时，在 f PCLK = 8 MHz 或 f PCLK = 16 MHz 下编程波特率时的误差计算 (1)

8 倍过采样**(0VER8=1)**							
波特率		fpCLK = 8 MHz			fpCLK = 16 MHz		
序号	所需值	实际值	波特率寄存器 中编程的值	误差% =（计算值 • 所需值）/ 所需波特率	实际值	波特率寄存器 中编程的值	误差%
11	**1.8432 MBps**	NA	NA	NA	1.7777 MBps	1.125	3.55%
12	**3.584 MBps**	NA	NA	NA	NA	NA	NA
13	**3.6864 MBps**	NA	NA	NA	NA	NA	NA
14	**7.168 MBps**	NA	NA	NA	NA	NA	NA
15	**7.3728 MBps**	NA	NA	NA	NA	NA	NA

1. CPU 时钟越低，特定波特率的精度就越低。可达到的波特率的上限可使用这些数据进行确定。

表 114. 采用 16 倍过采样时，在 f PCLK = 30 MHz 或 f PCLK = 60 MHz 下编程波特率时的误差计算 (1)(2)

16 倍过采样时（**OVER8=0**）							
波特率		**fpCLK = 30 MHz**			**fpCLK = 60 MHz**		
序号	所需值	实际值	波特率寄存器 中编程的值	误差%=（计算值 • 所需值）/所需波特率	实际值	波特率寄存器 中编程的值	误差%
1	**2.4 KBps**	2.400 KBps	781.2500	0.00%	2.400 KBps	1562.5000	0.00%
2	**9.6 KBps**	9.600 KBps	195.3125	0.00%	9.600 KBps	390.6250	0.00%
3	**19.2 KBps**	19.194 KBps	97.6875	0.03%	19.200 KBps	195.3125	0.00%
4	**57.6 KBps**	57.582KBps	32.5625	0.03%	57.582 KBps	65.1250	0.03%
5	**115.2 KBps**	115.385 KBps	16.2500	0.16%	115.163 KBps	32.5625	0.03%

6	**230.4 KBps**	230.769 KBps	8.1250	0.16%	230.769KBps	16.2500	0.16%
7	**460.8 KBps**	461.538 KBps	4.0625	0.16%	461.538 KBps	8.1250	0.16%
8	**896 KBps**	909.091 KBps	2.0625	1.46%	895.522 KBps	4.1875	0.05%
9	**921.6 KBps**	909.091 KBps	2.0625	1.36%	923.077 KBps	4.0625	0.16%
10	**1.792 MBps**	1.1764 MBps	1.0625	1.52%	1.8182 MBps	2.0625	1.36%
11	**1.8432 MBps**	1.8750 MBps	1.0000	1.73%	1.8182 MBps	2.0625	1.52%
12	**3.584 MBps**	NA	NA	NA	3.2594 MBps	1.0625	1.52%
13	**3.6864 MBps**	NA	NA	NA	3.7500 MBps	1.0000	1.73%
14	**7.168 MBps**	NA	NA	NA	NA	NA	NA
15	**7.3728 MBps**	NA	NA	NA	NA	NA	NA

1. CPU 时钟越低，特定波特率的精度就越低。可达到的波特率的上限可使用这些数据进行确定。
2. 仅 USART1 使用 PCLK2 计时。其它 USART 使用 PCLK1 计时。有关 PCLK1 和 PCLK2 的最大值，请参见器件数据手册。

表 115. 采用 8 倍过采样时，在 f PCLK = 30 MHz 或 f PCLK = 60 MHz 下编程波特率时的误差计算 (1) (2)

8 倍过采样(OVER8=1)							
波特率		**f_{pCLK} = 30 MHz**			**f_{pCLK} = 60 MHz**		
序号	所需值	实际值	波特率寄存器 中编程的值	误差%=(计算值-所需值)/所需波特率	实际值	波特率寄存器中编程的值	误差%
1	**2.4 KBps**	2.400 KBps	1562.5000	0.00%	2.400 KBps	3125.0000	0.00%
2	**9.6 KBps**	9.600 KBps	390.6250	0.00%	9.600 KBps	781.2500	0.00%
3	**19.2 KBps**	19.194 KBps	195.3750	0.03%	19.200 KBps	390.6250	0.00%
4	**57.6 KBps**	57.582 KBps	65.1250	0.16%	57.582 KBps	130.2500	0.03%
5	**115.2 KBps**	115.385 KBps	32.5000	0.16%	115.163 KBps	65.1250	0.03%
6	**230.4 KBps**	230.769 KBps	16.2500	0.16%	230.769 KBps	32.5000	0.16%
7	**460.8 KBps**	461.538 KBps	8.1250	0.16%	461.538 KBps	16.2500	0.16%
8	**896 KBps**	909.091 KBps	4.1250	1.46%	895.522 KBps	8.3750	0.05%
9	**921.6 KBps**	909.091 KBps	4.1250	1.36%	923.077 KBps	8.1250	0.16%
10	**1.792 MBps**	1.7647 MBps	2.1250	1.52%	1.8182 MBps	4.1250	1.46%
11	**1.8432 MBps**	1.8750 MBps	2.0000	1.73%	1.8182 MBps	4.1250	1.36%
12	**3.584 MBps**	3.7500 MBps	1.0000	4.63%	3.5294 MBps	2.1250	1.52%
13	**3.6864 MBps**	3.7500 MBps	1.0000	1.73%	3.7500 MBps	2.0000	1.73%
14	**7.168 MBps**	NA	NA	NA	7.5000 MBps	1.0000	4.63%
15	**7.3728 MBps**	NA	NA	NA	7.5000 MBps	1.0000	1.73%

STC MCU

1. CPU 时钟越低，特定波特率的精度就越低。可达到的波特率的上限可使用这些数据进行确定。
2. 仅 USART1 使用 PCLK2 计时。其它 USART 使用 PCLK1 计时。

表 116. 采用 16 倍过采样时，在 f PCLK = 42 MHz 或 f PCLK = 84 Hz 下编程波特率时的误差计算 (1)(2)

16 倍过采样时（OVER8=0）							
波特率		fpCLK = 42 MHz			fpCLK = 84 MHz		
序号	所需值	实际值	波特率寄存器 中编程的值	误差%=（计算值 - 所需值）/所需波特率	实际值	波特率寄存器 中编程的值	误差%
9	460.8 KBps	461.538 KBps	5.6875	0.16	461.538 KBps	11.375	0.16
10	921.6 KBps	913.043 KBps	2.875	0.93	923.076 KBps	5.6875	0.93
11	1.792 MBps	1.826 MBps	1.4375	1.9	1.787 MBps	2.9375	0.27
12	1.8432 MBps	1.826 MBps	1.4375	0.93	1.826 MBps	2.875	0.93
13	3.584 MBps	N.A	N.A	N.A	3.652 MBps	1.4375	1.9
14	3.6864 MBps	N.A	N.A	N.A	3.652 MBps	1.4375	0.93
15	7.168 MBps	N.A	N.A	N.A	N.A	N.A	N.A
16	7.3728 MBps	N.A	N.A	N.A	N.A	N.A	N.A
18	9 MBps	N.A	N.A	N.A	N.A	N.A	N.A
20	10.5 MBps	N.A	N.A	N.A	N.A	N.A	N.A

1. CPU 时钟越低，特定波特率的精度就越低。可达到的波特率的上限可使用这些数据进行确定。
2. 仅 USART1 使用 PCLK2 计时。其它 USART 使用 PCLK1 计时。有关 PCLK1 和 PCLK2 的最大值，请参见器件数据手册。

表 117. 采用 8 倍过采样时，在 f PCLK = 42 MHz 或 f PCLK = 84 MHz 下编程波特率时的误差计算 (1)(2)

8 倍过采样(OVER8=1)							
波特率		f_{pCLK} = 42 MHz			f_{pCLK} = 84 MHz		
序号	所需值	实际值	波特率寄存器中编程的值	误差%=(计算 值 - 所需值)/ 所需波特率	实际值	波特率寄存器 中编程的值	误差%
1	1.2 KBps	1.2 KBps	4375	0	1.2 KBps	8750	0
2	2.4 KBps	2.4 KBps	2187.5	0	2.4 KBps	4375	0
3	9.6 KBps	9.6 KBps	546.875	0	9.6 KBps	1093.75	0
4	19.2 KBps	19.195 KBps	273.5	0.02	19.2 KBps	546.875	0
5	38.4 KBps	38.391 KBps	136.75	0.02	38.391 KBps	273.5	0.02
6	57.6 KBps	57.613 KBps	91.125	0.02	57.613 KBps	182.25	0.02
7	115.2 KBps	115.068 KBps	45.625	0.11	115.226 KBps	91.125	0.02
8	230.4 KBps	230.769 KBps	22.75	0.11	230.137 KBps	45.625	0.11

9	460.8 KBps	461.538 KBps	11.375	0.16	461.538 KBps	22.75	0.16
10	921.6 KBps	913.043 KBps	5.75	0.93	923.076 KBps	11.375	0.93
11	1.792 MBps	1.826 MBps	2.875	1.9	1.787Mbps	5.875	0.27
12	1.8432 MBps	1.826 MBps	2.875	0.93	1.826 MBps	5.75	0.93
13	3.584 MBps	3.5 MBps	1.5	2.34	3.652 MBps	2.875	1.9
14	3.6864 MBps	3.82 MBps	1.375	3.57	3.652 MBps	2.875	0.93

表 117. 采用 8 倍过采样时，在 f PCLK = 42 MHz 或 f PCLK = 84 MHz 下编程波特率时的误差计算 (1)(2)

8 倍过采样(OVER8=1)							
波特率		fpCLK = 42 MHz			fpCLK = 84 MHz		
序号	所需值	实际值	波特率寄存器 中编程的值	误差%=(计算值-所需值)/所需波特率	实际值	波特率寄存器中编程的值	误差%
15	7.168 MBps	N.A	N.A	N.A	7 MBps	1.5	2.34
16	7.3728 MBps	N.A	N.A	N.A	7.636 MBps	1.375	3.57
18	9 MBps	N.A	N.A	N.A	9.333 MBps	1.125	3.7
20	10.5 MBps	N.A	N.A	N.A	10.5 MBps	1	0

1. CPU 时钟越低，特定波特率的精度就越低。可达到的波特率的上限可使用这些数据进行确定。
2. 仅 USART1 使用 PCLK2 计时。其它 USART 使用 PCLK1 计时。有关 PCLK1 和 PCLK2 的最大值，请参见器件数据手册。

20.3.5 USART 接收器对时钟偏差的容差

仅当总时钟系统偏差小于 USART 接收器的容差时，USART 异步接收器器才能正常工作。
影响总偏差的因素包括：
- DTRA：发送器误差引起的偏差（其中还包括发送器本地振荡器的偏差）
- DQUANT：接收器的波特率量化引起的误差
- DREC：接收器本地振荡器的偏差
- DTCL：传输线路引起的偏差（通常是由于收发器所引起，它可能会在低电平到高电平转换时序与高电平到低电平转换时序之间引入不对称）

DTRA + DQUANT + DREC + DTCL < USART 接收器的容差
对于正常接收数据，USART 接收器的容差等于所容许的最大偏差，具体取决于以下选择：
- 由 USART_CR1 寄存器中的 M 位定义的 10 位或 11 位字符长度
- 由 USART_CR1 寄存器中的 OVER8 位定义的 8 倍或 16 倍过采样
- 是否使用小数波特率
- 使用 1 位或 3 位对数据进行采样，取决于 USART_CR3 寄存器中 ONEBIT 位的值

表 118. DIV_Fraction 为 0 时的 USART 接收器容差

M 位	OVER8 位=0		OVER8 位=1	
	ONEBIT=0	ONEBIT=1	ONEBIT=0	ONEBIT=1
0	3.75%	4.375%	2.50%	3.75%
1	3.41%	3.97%	2.27%	3.41%

表 119. DIV_Fraction 不为 0 时的 USART 接收器容差

M 位	OVER8 位=0		OVER8 位=1	
	ONEBIT=0	ONEBIT=1	ONEBIT=0	ONEBIT=1
0	3.33%	3.88%	2%	3%
1	3.03%	3.53%	1.82%	2.73%

注意： 当接收的帧包含 10 个位时间的空闲帧（ M=0 ）或 11 个位时间的空闲帧（ M=1 ）时，表 118 和表 119 中指定的数字可能与特例中的数字略微不同。

20.3.6 多处理器通信

可以与 USART 进行多处理器通信（多个 USART 连接在一个网络中）。例如，其中一个 USART 可以是主 USART，其 TX 输出与其它 USART 的 RX 输入相连接。其它 USART 为从 USART，其各自的 TX 输出在逻辑上通过与运算连在一起，并与主 USART 的 RX 输入相连接。

在多处理器配置中，理想情况下通常只有预期的消息接收方主动接收完整的消息内容，从而减少由所有未被寻址的接收器造成的冗余 USART 服务开销。

可通过静音功能将未被寻址的器件置于静音模式下。在静音模式下：

- 不得将接收状态位置 1。
- 禁止任何接收中断。
- USART_CR1 寄存器中的 RWU 位置 1。RWU 可由硬件自动控制，或在特定条件下由软件写入。

根据 USART_CR1 寄存器中 WAKE 位的设置，USART 可使用以下两种方法进入或退出静音模式：

- 如果 WAKE 位被复位，则进行空闲线路检测，
- 如果 WAKE 位置 1，则进行地址标记检测。

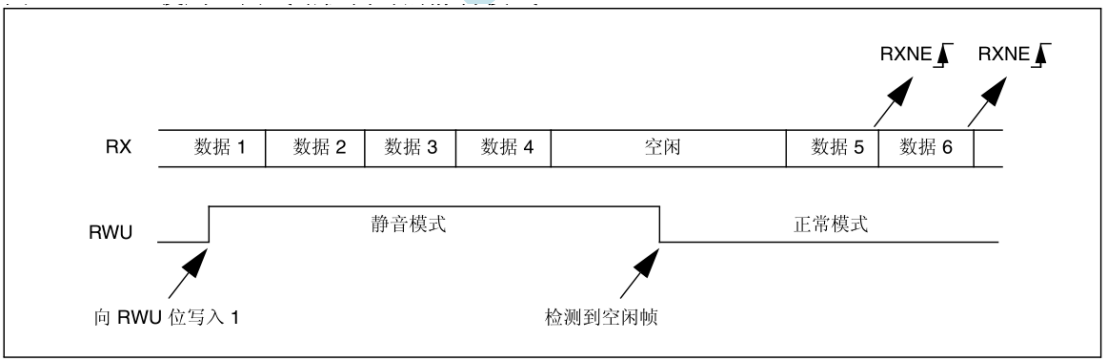
空闲线路检测 (WAKE=0)

当向 RWU 位写入 1 时，USART 进入静音模式。

当检测到空闲帧时，它会被唤醒。此时 RWU 位会由硬件清零，但 USART_SR 寄存器中的 IDLE 位不会置 1。还可通过软件向 RWU 位写入 0。

图 253 中给出了使用空闲线路检测时静音模式行为的示例。

图 253. 使用空闲线路检测时的静音模式



地址标记检测 (WAKE=1)

在此模式下，如果字节的 MSB 为 1，则将这些字节识别为地址，否则将其识别为数据。在地址字节中，目标接收器的地址位于 4 个 LSB 上。接收器会将此 4 位字与其地址进行比较，该接收器的地址在 USART_CR2 寄存器的 ADD 位中进行设置。

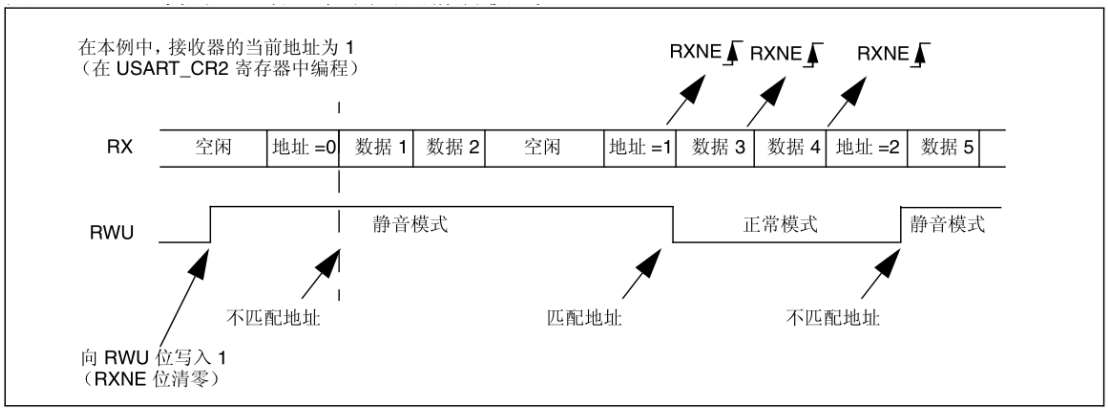
当接收到与其编程地址不匹配的地址字符时，USART 会进入静音模式。此时，RWU 位将由硬件置 1。由于当时 USART 已经进入了静音模式，所以 RXNE 标志不会针对此地址字节置 1，也不会发出中断或 DMA 请求。

当接收到与编程地址匹配的地址字符时，它会退出静音模式。然后 RWU 位被清零，可以开始正常接收后续字节。由于 RWU 位已清零，RXNE 位会针对地址字符 置 1。

当接收器的缓冲区不包含任何数据（USART_SR 寄存器中 RXNE=0）时，可向 RWU 位写入 0 或 1。否则会忽略写尝试。

图 254 中给出了使用地址标记检测时静音模式行为的示例。

图 254. 使用地址标记检测时的静音模式



20.3.7奇偶校验控制

将 USART_CR1 寄存器中的 PCE 位置 1，可以使能奇偶校验控制（发送时生成奇偶校验位，接收时进行奇偶校验检查）。根据 M 位定义的帧长度， 表 120 中列出了可能的 USART 帧格式。

表 120. 帧格式

M 位	PCE 位	USART 帧 (1)
0	0	|SB| 8 位数据 |STB|
0	1	|SB| 7 位数据 |PB|STB|
1	0	|SB| 9 位数据 |STB|
1	1	|SB| 8 位数据 PB|STB|

1. 图注：SB：起始位，STB：停止位，P：奇偶校验位。

偶校验

对奇偶校验位进行计算，使帧和奇偶校验位中“1”的数量为偶数（帧由 7 个或 8 个 LSB 位组成，具体取决于 M 等于 0 还是 1）。

例如：数据=00110101；4 个位置 1 => 如果选择偶校验（USART_CR1 寄存器中的 PS 位 = 0），则校验位是 0。

奇校验

对奇偶校验位进行计算，使帧和奇偶校验位中“1”的数量为奇数（帧由 7 个或 8 个 LSB 位组成，具体取决于 M 等于 0 还是 1）。

例如：数据=00110101；4 个位置 1 => 如果选择奇校验（USART_CR1 寄存器中的 PS 位 = 1），则校验位是 1。

接收时进行奇偶校验检查

如果奇偶校验检查失败，则 USART_SR 寄存器中的 PE 标志置 1；如果 USART_CR1 寄存器中 PEIE 位置 1，则会生成中断。PE 标志由软件序列清零（从状态寄存器中读取，然后对 USART_DR 数据寄存器执行读或写访问）。

注意： 如果被地址标记唤醒：会使用数据的 MSB 位而非奇偶校验位来识别地址。此外，接收器不会对地址数据进行奇偶校验检查（奇偶校验出错时， PE 不置 1 ）。

发送时的奇偶校验生成

如果 USART_CR1 寄存器中的 PCE 位置 1，则在数据寄存器中所写入数据的 MSB 位会进行传送，但是会由奇偶校验位进行更改（如果选择偶校验 (PS=0)，则“1”的数量为偶数；如果选择奇校验 (PS=1)，则“1”的数量为奇数）。

注意： 用于管理发送过程的软件程序可以激活软件序列，进而将 PE 标志清零（从状态寄存器中读取，然后对数据寄存器执行读或写访问）。在半双工模式下工作时（具体取决于软件），这可能会导致 PE 标

志意外清零。

20.3.8LIN （局域互连网络）模式

通过将 USART_CR2 寄存器中的 LINEN 位置 1 来选择 LIN 模式。在 LIN 模式下，必须将以下位清零：

- USART_CR2 寄存器中的 CLKEN 位，
- USART_CR3 寄存器中的 STOP[1:0]、SCEN、HDSEL 和 IREN 位。

LIN 发送

与正常的 USART 发送相比，在 LIN 主器件中发送时必须采用 第 26.3.2 节 中介绍的相同步骤，同时还具有以下区别：

- M 位清零以配置 8 位字长度。
- LINEN 位置 1 以进入 LIN 模式。此时，将 SBK 位置 1 会发送 13 个“0”位作为断路字符。然后会发送值为“1”的位以进行下一启动检测。

LIN 接收断路检测电路在 USART 接口上实现。该检测完全独立于正常的 USART 接收器。在空闲状态或某个帧期间，只要发生断路即可检测出来。

接收器（USART_CR1 寄存器中 RE=1）使能后，电路便开始监测启动信号的 RX 输入。检测起始位的方法与搜索断路字符或数据的方法相同。检测到起始位后，电路会对接下来的位进行采样，方法与数据采样相同（第 8、第 9 和第 10 次采样）。如果 10 个（USART_CR2 寄存器中 LBDL = 0 时）或 11 个（USART_CR2 寄存器中 LBDL=1 时）连续位均检测为“0”，且其后跟随分隔符，则 USART_SR 寄存器中的 LBD 标志将会置 1。如果 LBDIE 位=1，则会生成中断。在验证断路前，会对分隔符进行检查，因为它表示 RX 线路已恢复到高电平。

如果在第 10 或第 11 次采样前已对“1”采样，则断路检测电路会取消当前检测，并重新搜索起始位。

如果禁止 LIN 模式 (LINEN=0)，接收器会作为正常的 USART 继续工作，不会再进行断路检测。

如果使能 LIN 模式 (LINEN=1)，只要发生帧错误（例如，在“0”处检测到停止位，这种情况可能出现在任何断路帧中），接收器即会停止，直到断路检测电路接收到“1”（断路字不完整时）或接收到分隔符（检测到断路时）为止。

图 255 ：第 699 页的 LIN 模式下的断路检测（ 11 位断路长度 —LBDL 位置 1 ） 中显示了断路检测器状态机和断路标志的行为。

图 256 ：第 700 页的 LIN 模式下的断路检测与帧错误检测 中列出了断路帧的示例。

STC MCU

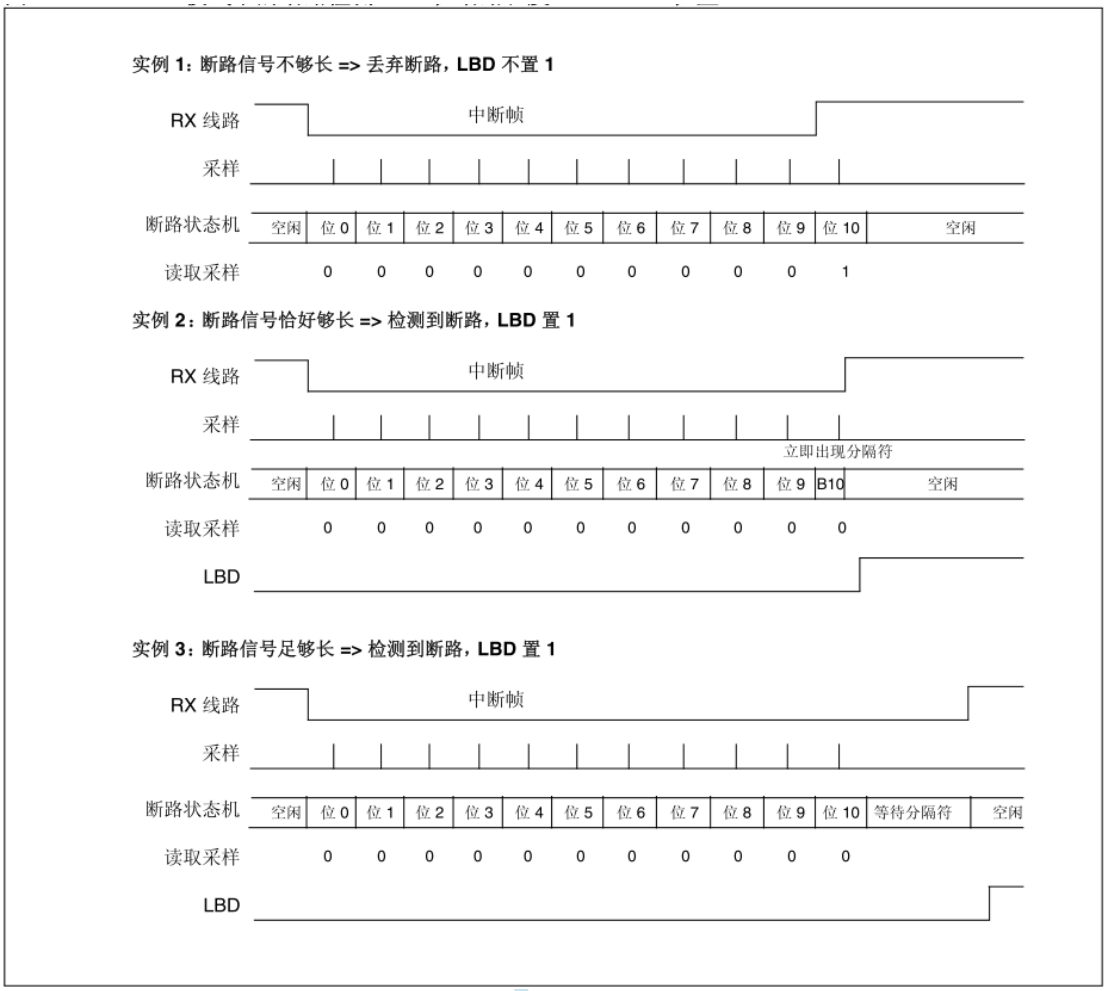
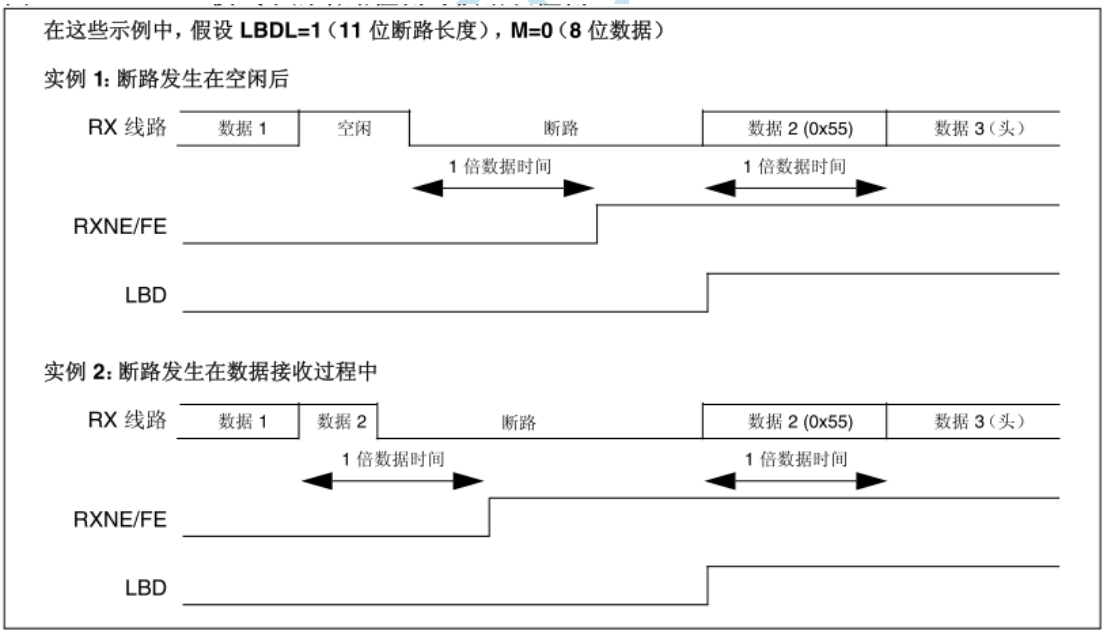


图 256. LIN 模式下的断路检测与帧错误检测



20.3.9 USART 同步模式

通过将 USART_CR2 寄存器中的 CLKEN 位写入 1 来选择同步模式。在同步模式下，必须将

以下位清零：
● USART_CR2 寄存器中的 LINEN 位，
● USART_CR3 寄存器中的 SCEN、HDSEL 和 IREN 位。
通过 USART，用户可以在主模式下控制双向同步串行通信。SCLK 引脚是 USART 发送器时钟的输出。在起始位或停止位期间，不会向 SCLK 引脚发送时钟脉冲。在最后一个有效数据位（地址标记）期间，将会（也可能不会）生成时钟脉冲，这取决于 USART_CR2 寄存器中 LBCL 位的状态。通过 USART_CR2 寄存器中的 CPOL 位，用户可以选择时钟极性；通过 USART_CR2 寄存器中的 CPHA 位，用户可以选择外部时钟相位
在空闲状态、报头模式和发送断路期间，外部 SCLK 时钟处于未激活状态。
USART 发送器在同步模式下的工作方式与异步模式下完全相同。但是由于 SCLK 与 TX 同步（根据 CPOL 和 CPHA），因此 TX 上的数据是同步的。
在此模式下，USART 接收器的工作方式与异步模式下不同。如果 RE=1，则数据在 SCLK 上采样（上升或下降沿，取决于 CPOL 和 CPHA），而不会进行任何过采样。此时必须确保建立时间和保持时间（取决于波特率：1/16 位时间）符合要求。
注意： SCLK 引脚可与 TX 引脚结合使用。因此，仅当使能发送器 (TE=1) 且正在发送数据时（对数据寄存器 USART_DR 已被写入），才会提供时钟。这意味着，没有发送数据的情况下无法接收同步数据。
当发送器和接收器 (TE=RE=0) 都被禁止时，必须选择 LBCL 、 CPOL 和 CPHA 位，以确保时钟脉冲正常工作。当使能发送器或接收器时，不得对这些位进行更改。
建议按照相同指令将 TE 和 RE 位置 1 ，以尽量缩短接收器的建立时间和保持时间。
USART 只支持主模式：它不能接收或发送与输入时钟相关的数据（ SCLK 始终为输出）。
图 257. USART 同步发送示例

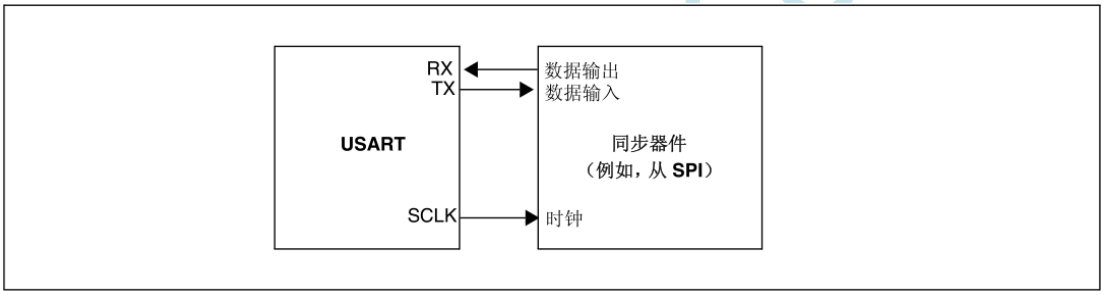


图 258. USART 数据时钟时序图 (M=0)

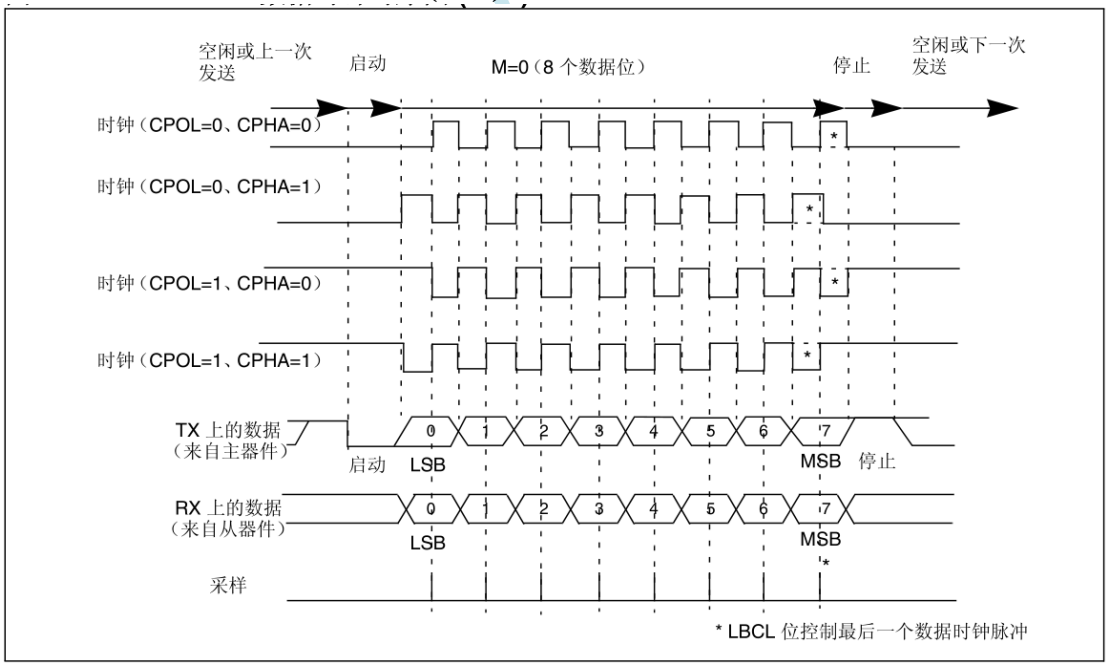


图 259. USART 数据时钟时序图 (M=1)

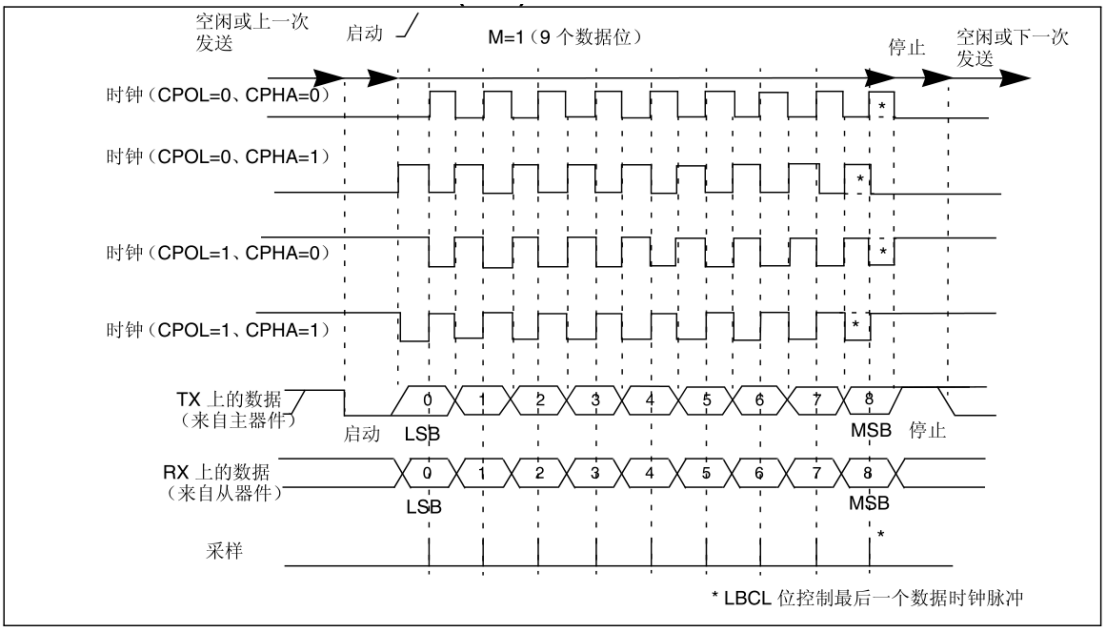
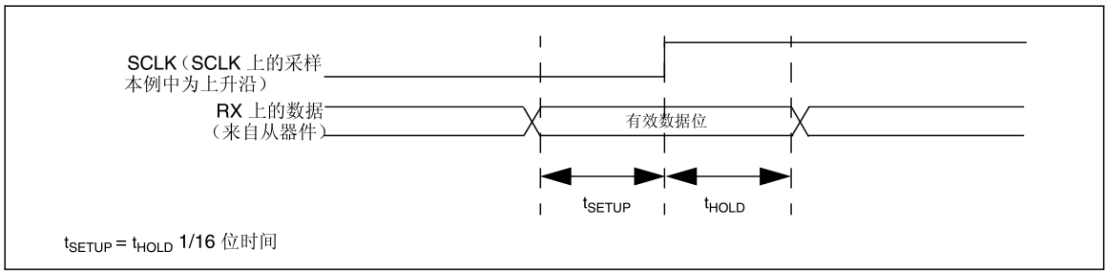


图 260. RX 数据建立/保持时间



注意： 在智能卡模式下， SCLK 的功能有所不同。有关详细信息，请参见智能卡模式一章。

20.3.9单线半双工通信

通过将 USART_CR3 寄存器中的 HDSEL 位置 1 来选择单线半双工模式。在此模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN 和 CLKEN 位，
- USART_CR3 寄存器中的 SCEN 和 IREN 位。

USART 可以配置为遵循单线半双工协议，其中 TX 和 RX 线路从内部相连接。使用控制位“HALF DUPLEX SEL”（USART_CR3 寄存器中的 HDSEL 位），可以在半双工通信和全双工通信间进行选择。

一旦向 HDSEL 位写入 1：

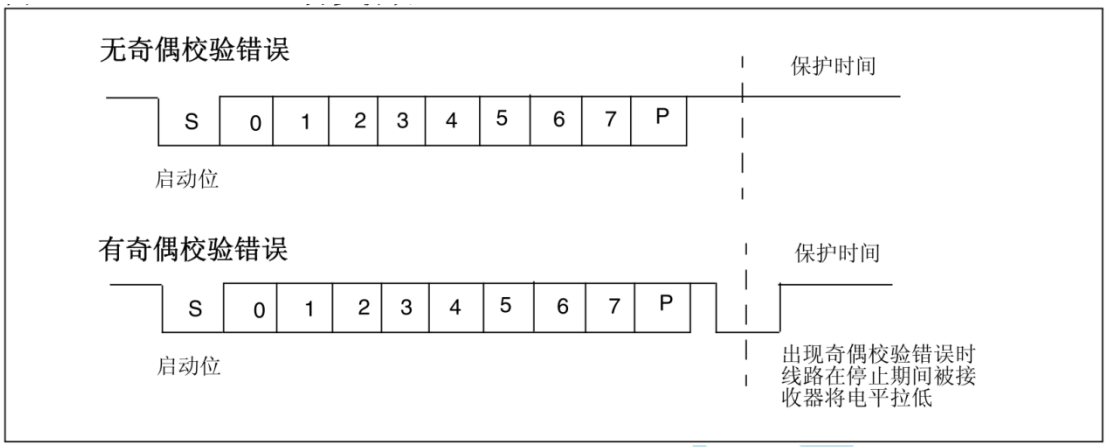
- TX 和 RX 线路从内部相连接
- 不能再使用 RX 引脚
- 无数据传输时，TX 引脚始终处于释放状态。因此，它在空闲状态或接收过程中用作标准 I/O。这意味着，必须对 I/O 进行配置，以便在未受 USART 驱动时，使 TX 成为浮空输入（或高电平开漏输出）。

除此之外，通信与正常 USART 模式下的通信相似。此线路上的冲突必须由软件进行管理（例如，使用中央仲裁器）。尤其要注意，发送过程永远不会被硬件封锁，只要数据是在 TE 位置 1 的情况下写入，发送就会持续进行。

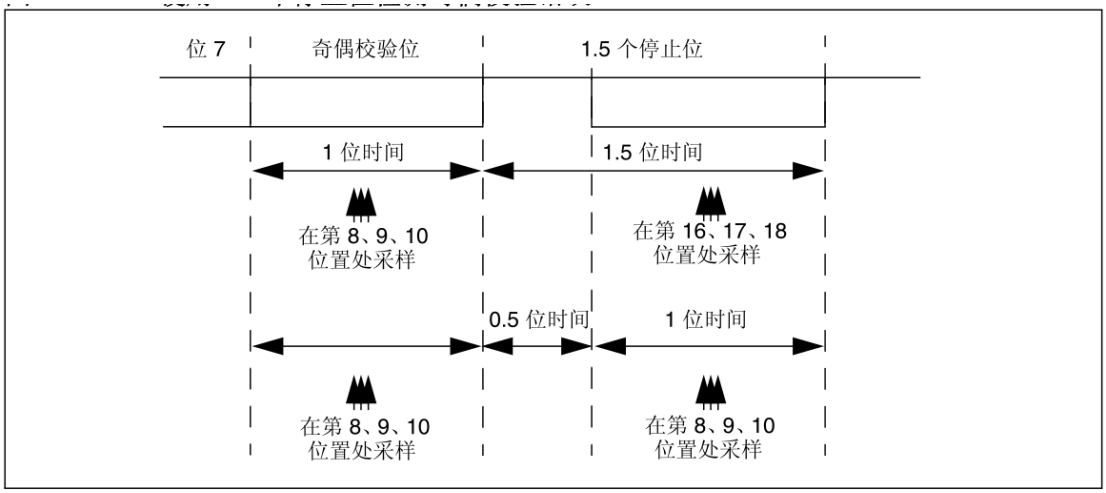
20.3.10智能卡

通过将 USART_CR3 寄存器中的 SCEN 位置 1 来选择智能卡模式。在智能卡模式下，必须将以下位清零：

● USART_CR2 寄存器中的 LINEN 位，
● USART_CR3 寄存器中的 HDSEL 和 IREN 位。
此外，可能需要将 CLKEN 位置 1，以便为智能卡提供时钟。
智能卡接口支持符合 ISO 7816-3 标准的异步协议智能卡。USART 应如下所示进行配置：
● 8 个位加奇偶校验：当 USART_CR1 寄存器中 M=1 且 PCE=1 时
● 发送和接收时使用 1.5 个停止位：当 USART_CR2 寄存器中 STOP=11 时。
注意： 接收时也可以选择 0.5 个停止位，但为了避免在两种配置之间切换，建议发送和接收时均使用 1.5 个停止位。
图 261 显示了有奇偶校验错误和无奇偶校验错误时数据线上情况的示例。
图 261. ISO 7816-3 异步协议



连接到智能卡时，USART 的 TX 输出会驱动一条双向线（它也由智能卡驱动）。必须将 TX 引脚配置为开漏引脚。
智能卡是一个单线半双工通信协议。
● 从发送移位寄存器发送数据会经过至少 1/2 个时钟周期的延迟。正常工作时，已满的发送移位寄存器会在下一个时钟边沿开始移位。在智能卡模式下，此发送过程还会进一步经过 1/2 波特时钟周期的延迟。
● 如果在接收一个使用 0.5 或 1.5 个停止位编程的帧期间检测到奇偶校验错误，则在完成接收帧后，发送线会被拉低一个时钟周期。这是为了向智能卡指出发送到 USART 的数据尚未正确接收。此 NACK 信号（将发送线拉低 1 个时钟周期）会导致发送器端（配置为 1.5 个停止位）出现帧错误。应用程序可根据协议重新发送数据。如果 NACK 控制位置 1，则接收器会向奇偶校验错误发送“NACK”信号；否则不会发送 NACK 信号。
● 通过对保护时间寄存器进行编程，可以延迟 TC 标志的置位。正常工作时，当发送移位寄存器为空且没有新的发送请求出现时，会对 TC 标志进行置位。在智能卡模式下，空的发送移位寄存器会触发保护时间计数器，使其递增计数至保护时间寄存器中的值。在此期间，TC 标志被强制为低电平。当保护时间计数器达到设置值时，TC 置位为高电平。
● TC 标志的释放不受智能卡模式的影响。
● 如果在发送端检测到帧错误（由来自接收器的 NACK 信号引起），则发送端的接收块不会将 NACK 作为起始位进行检测。根据 ISO 协议，接收到的 NACK 信号的持续时间可以是 1 或 2 个时钟周期。
● 在接收端，如果检测到奇偶校验错误并发送了 NACK 信号，则接收端不会将 NACK 作为起始位进行检测。
注意： 中断字符在智能卡模式下无效。带有帧错误的 0x00 数据将被视为数据，而非中断。当翻转 TE 位时，不会发送空闲帧。空闲帧（在其它配置中进行了定义）在 ISO 协议中未进行定义。图 262 详细介绍了 USART 如何对 NACK 信号采样。在本例中，USART 正在发送数据并配置了 1.5 个停止位。USART 的接收部分已被使能，以检查数据的完整性和 NACK 信号。
图 262. 使用 1.5 个停止位检测奇偶校验错误



USART 可以通过 SCLK 输出为智能卡提供时钟。在智能卡模式下，SCLK 仅通过一个 5 位预分频器由内部外设输入时钟提供。分频比在预分频器寄存器 USART_GTPR 中进行配置。SCLK 频率可在 f CK /2 到 f CK /62 之间进行编程，其中 f CK 为外设输入时钟。

20.3.11 IrDA SIR ENDEC 模块

通过将 USART_CR3 寄存器中的 IREN 位置 1 来选择 IrDA 模式。在 IrDA 模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN、STOP 和 CLKEN 位，
- USART_CR3 寄存器中的 SCEN 和 HDSEL 位。

IrDA SIR 物理层规定使用反相归零 (RZI) 调制方案，它以红外光脉冲表示逻辑 0（参见 图 263）。

SIR 发送编码器用于调制 USART 发出的非归零 (NRZ) 位流。输出脉冲流会发送到外部输出驱动器和红外线 LED。USART 支持的 SIR ENDEC 比特率最高为 115.2Kbps。在正常模式下，所发送的脉冲宽度规定为一个位周期的 3/16。SIR 接收解码器用于解调由红外探测器发出的归零位流，并将接收到的 NRZ 串行位流输出到 USART。在空闲状态下，解码器输入通常为高电平（标记状态）。发送编码器输出的极性与解码器输入相反。当解码器输入为低电平时，会检测到起始位。

- IrDA 是一个半双工通信协议。如果发送器忙（例如，USART 正在向 IrDA 编码器发送数据），则 IrDA 解码器会忽略 IrDA 接收线上的所有数据；如果接收器忙（例如，USART 正在接收来自 USART 的解码数据），则 IrDA 不会对 USART 发送到 IrDA 的 TX 上的数据进行编码。在接收数据时，应避免同时进行发送，因为这样做可能会破坏要发送的数据。
- “0”作为高电平脉冲发送，而“1”作为“0”发送。在正常模式下，脉冲宽度规定为所选位周期的 3/16（参见 图 264）。
- SIR 解码器用于将兼容 IrDA 的接收信号转换为 USART 的位流。
- SIR 接收逻辑将高电平状态视为逻辑“1”，将低电平脉冲视为逻辑“0”。
- 发送编码器输出的极性与解码器输入相反。SIR 输出在空闲时处于低电平状态。
- IrDA 规范要求脉冲容忍值要大于 1.41 us。可接受的脉冲宽度可通过寄存器设置。接收器端的干扰检测逻辑会滤除宽度小于 2 个 PSC 周期的脉冲（PSC 是在 IrDA 低功耗波特寄存器 USART_GTPR 中编程的预分频器值）。宽度小于 1 个 PSC 周期的脉冲都将被拒绝，但宽度大于 1 个而小于 2 个周期的脉冲可能被接受也可能被拒绝，而宽度大于 2 个周期的脉冲将被接受作为有效脉冲。当 PSC=0 时，IrDA 编码器/解码器不工作。
- 接收器能够与低功耗发送器进行通信。
- 在 IrDA 模式下，USART_CR2 寄存器中的 STOP 位必须配置为“1 个停止位”。

IrDA 低功耗模式

发送器 ：

在低功耗模式下，脉冲宽度不再保持为位周期的 3/16。此时的脉冲宽度为低功耗波特率的 3 倍，最小可为 1.42 MHz。通常此值是 1.8432 MHz (1.42 MHz < PSC< 2.12 MHz)。低功耗模式下的可编程分频器会对系统时钟进行分频，以达到此值。

接收器 ：
在低功耗模式下接收与在正常模式下接收类似。为进行干扰检测，USART 应丢弃持续时间短于 1/PSC 的脉冲。只有当持续时间大于 2 个 IrDA 低功耗波特时钟周期（USART_GTPR 寄存器中的 PSC 值）时，才是有效低电平。
注意： 宽度小于两个但大于一个 PSC 周期的脉冲可能被接受，也可能被拒绝。
接收器的建立时间应由软件进行管理。 IrDA 物理层规范规定发送和接收之间至少要经过 10 ms 的延迟（ IrDA 是一个半双工协议）。

图 263. IrDA SIR ENDEC—框图

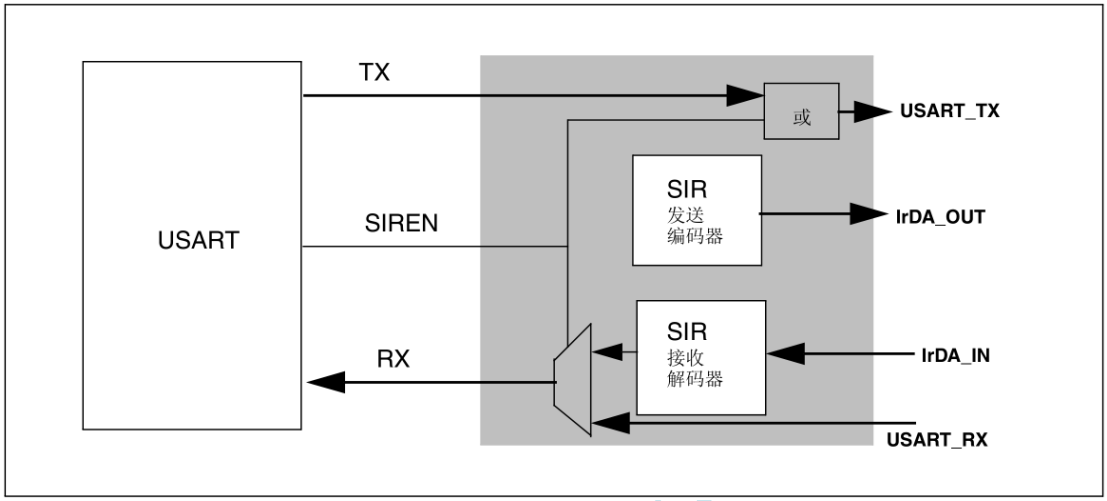
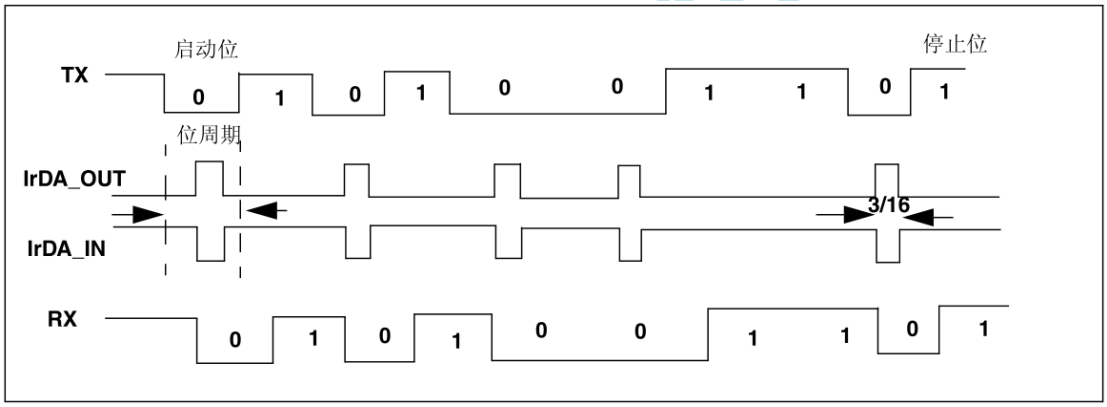


图 264. IrDA 数据调制 (3/16)—正常模式



20.3.12 使用 DMA

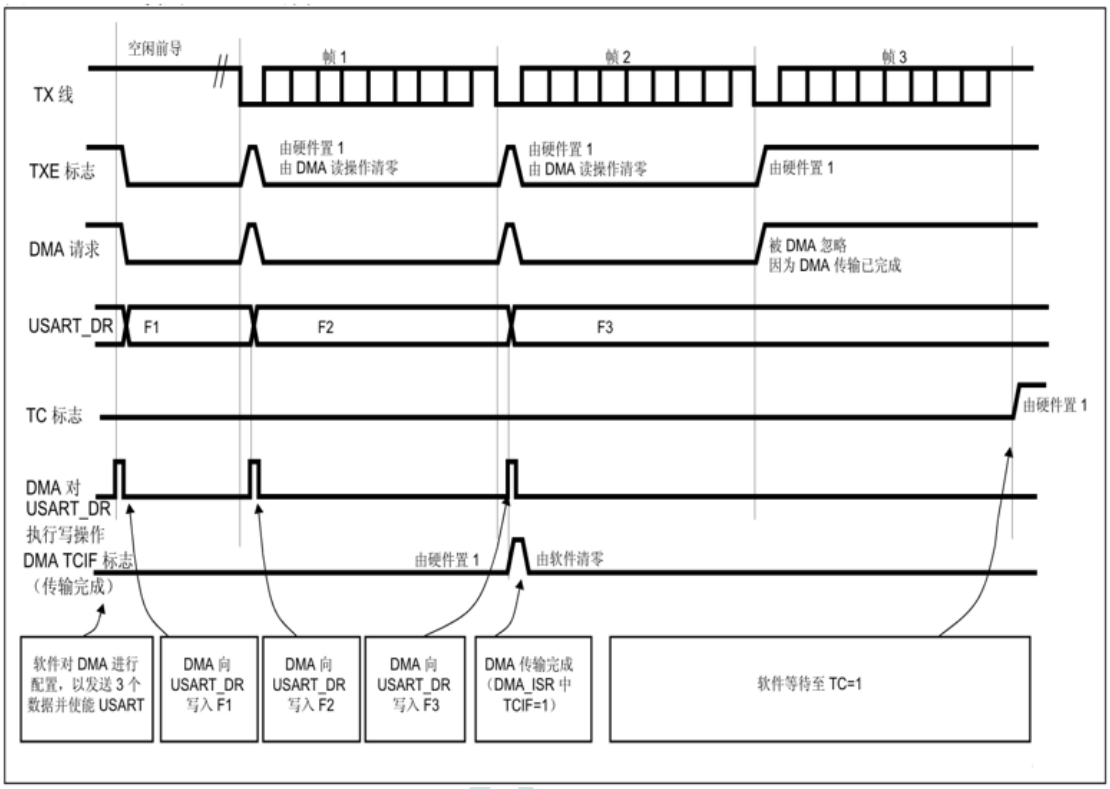
USART 能够使用 DMA 进行连续通信。接收缓冲区和发送缓冲区的 DMA 请求是独立的。

可以将 USART_SR 寄存器中的 TXE/RXNE 标志清零，从而实现连续通信。使用 DMA 进行发送可将数据从 SRAM 区（通过 DMA 配置，参见 DMA 部分）加载到 USART_DR 寄存器。要映射一个 DMA 通道以进行 USART 发送，请按以下步骤操作（x 表示通道编号）：

1. 在 DMA 控制寄存器中写入 USART_DR 寄存器地址，将其配置为传输的目标地址。每次发生 TXE 事件后，数据都会从存储器移动到此地址。
2. 在 DMA 控制寄存器中写入存储器地址，将其配置为传输的源地址。每次发生 TXE 事件后，数据都会从这个存储区域加载到 USART_DR 寄存器中。
3. 在 DMA 控制寄存器中配置要传输的总字节数。
4. 在 DMA 寄存器中配置通道优先级
5. 根据应用的需求，在完成一半或全部传输后产生 DMA 中断。
6. 向 SR 寄存器中的 TC 位写入 0，将其清零。
7. 在 DMA 寄存器中激活该通道。

当达到在 DMA 控制器中设置的数据传输量时，DMA 控制器会在 DMA 通道的中断向量上产生一个中断。在发送模式下，DMA 对所有要发送的数据执行了写操作（DMA_ISR 寄存器中的 TCIF 标志置 1）后，可以对 TC 标志进行监视，以确保 USART 通信已完成。在禁止 USART 或进入停止模式前必须执行此步骤，以避免损坏最后一次发送。软件必须等待直到 TC=1。TC 标志在所有数据发送期间都必须保持清零状态，然后在最后一帧发送结束后由硬件置 1。

图 265. 使用 DMA 进行发送



使用 DMA 进行接收

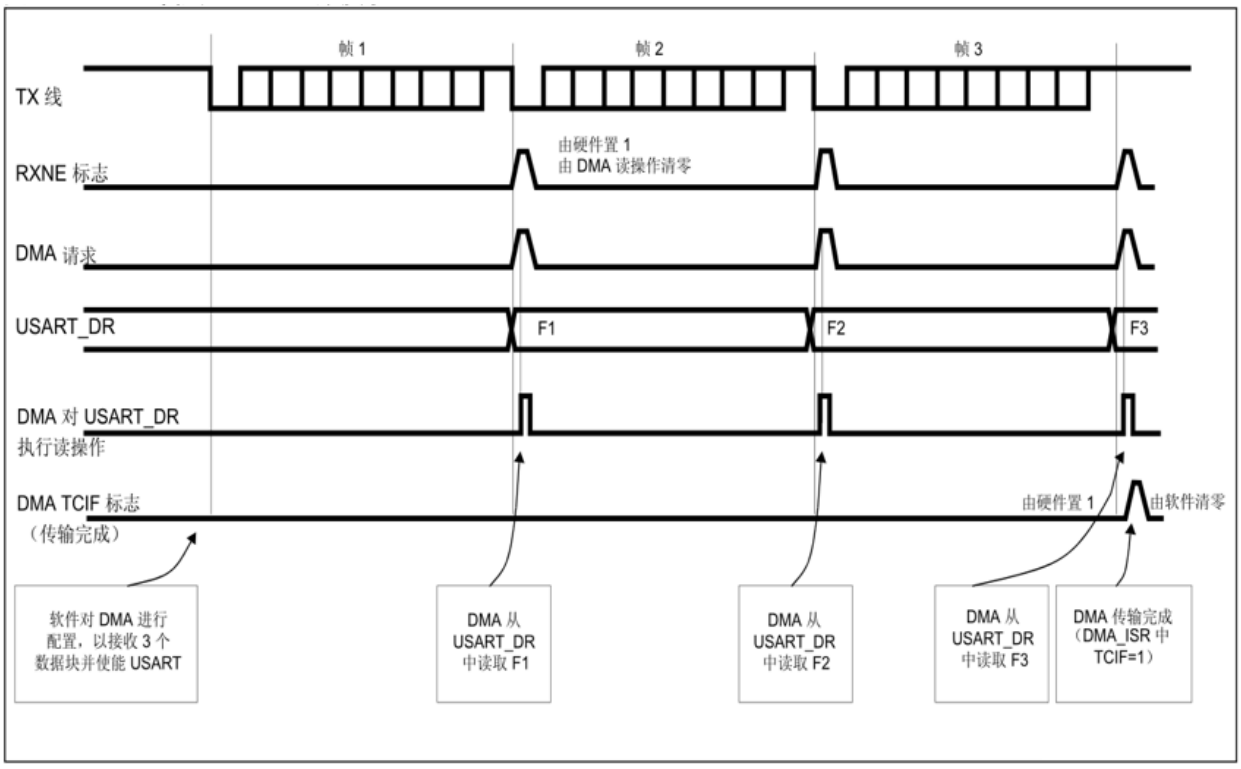
将 USART_CR3 寄存器中的 DMAR 位置 1 可以使能 DMA 模式进行接收。接收数据字节时，数据会从 USART_DR 寄存器加载到 SRAM 区域中（通过 DMA 配置，参见 DMA 规范）。要映射一个 DMA 通道以进行 USART 接收，请按以下步骤操作：

1. 在 DMA 控制寄存器中写入 USART_DR 寄存器地址，将其配置为传输的源地址。每次发生 RXNE 事件后，数据都会从此地址移动到存储器。
2. 在 DMA 控制寄存器中写入存储器地址，将其配置为传输的目标地址。每次发生 RXNE 事件后，数据都会从 USART_DR 寄存器加载到此存储区。
3. 在 DMA 控制寄存器中配置要传输的总字节数。
4. 在 DMA 控制寄存器中配置通道优先级。
5. 根据应用的需求，在完成一半或全部传输后产生中断。
6. 在 DMA 控制寄存器中激活该通道。

当达到在 DMA 控制器中设置的数据传输量时，DMA 控制器会在 DMA 通道的中断向量上产生一个中断。在中断子程序中，USART_CR3 寄存器中的 DMAR 位应由软件清零。

注意： 如果 DMA 用于接收，则不要使能 RXNEIE 位。

图 266. 使用 DMA



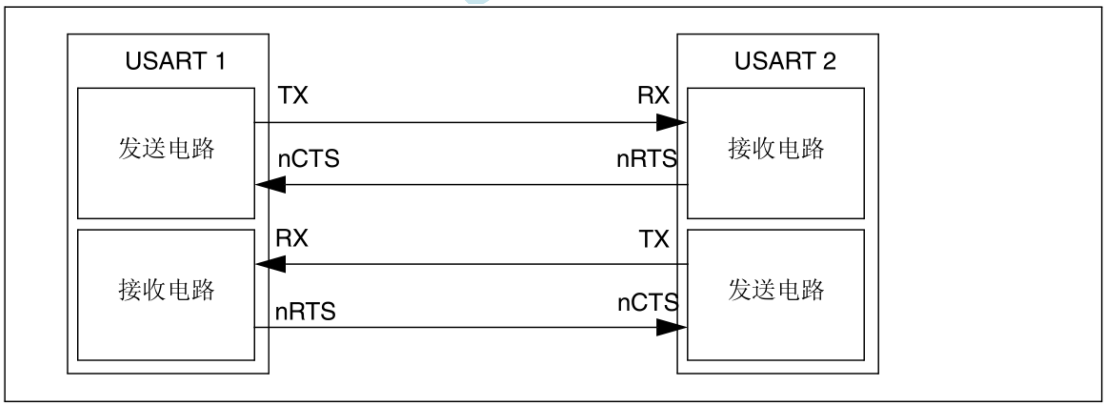
多缓冲区通信中的错误标志和中断生成

在多缓冲区通信中，如果事务中发生任何错误，都会在当前字节后放置错误标志。如果中断使能置 1，则会产生中断。在单字节接收过程中，与 RXNE 一同置位的帧错误、上溢错误和噪声标志具有单独的错误标志中断使能位（USART_CR3 寄存器中的 EIE 位）；如果该位置 1，则会因其中任何一个错误而在当前字节后产生中断。

20.3.13 硬件流控制

使用 nCTS 输入和 nRTS 输出可以控制 2 个器件间的串行数据流。 图 267 显示了在这种模式下如何连接 2 个器件：

图 267. 2 个 USART 间的硬件流控制

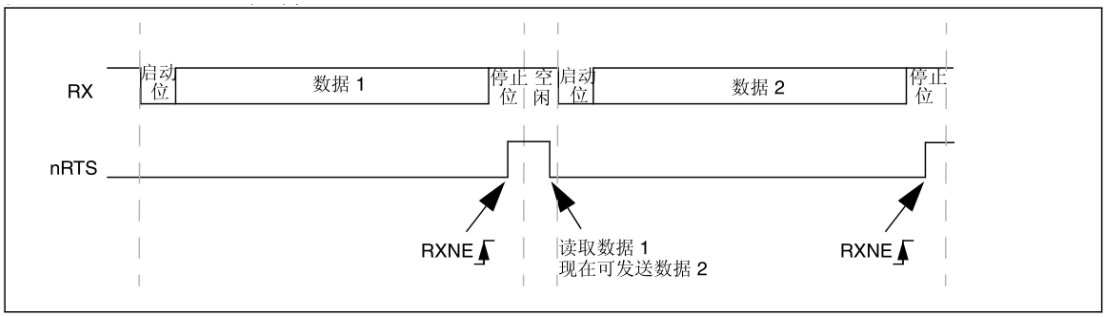


分别向 USART_CR3 寄存器中的 RTSE 位和 CTSE 位写入 1，可以分别使能 RTS 和 CTS 流控制。

RTS 流控制

如果使能 RTS 流控制 (RTSE=1)，只要 USART 接收器准备好接收新数据，便会将 nRTS 变为有效（连接到低电平）。当接收寄存器已满时，会将 nRTS 变为无效，表明发送过程会在当前帧结束后停止。 图 268 显示了在使能 RTS 流控制的情况下进行通信的示例。

图 268. RTS 流控制

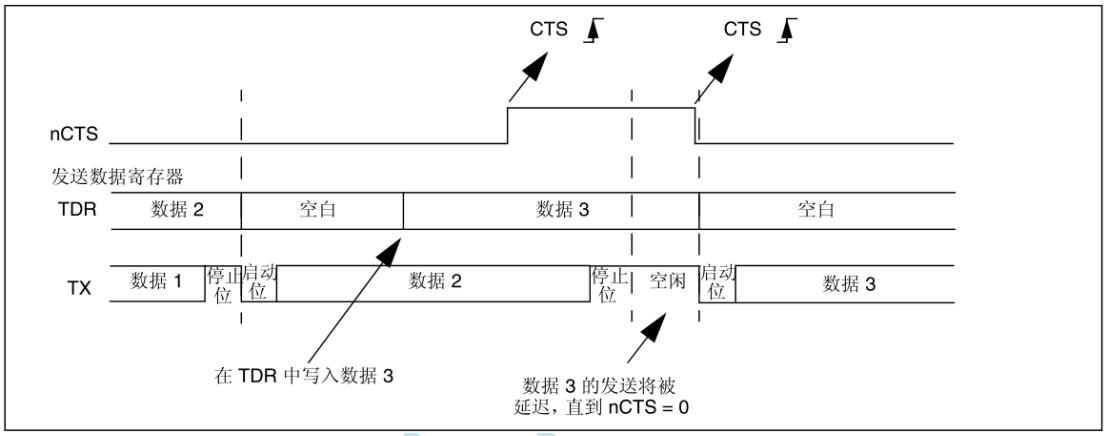


CTS

如果使能 CTS 流控制 (CTSE=1)，则发送器会在发送下一帧前检查 nCTS。如果 nCTS 有效（连接到低电平），则会发送下一数据（假设数据已准备好发送，即 TXE=0）；否则不会进行发送。如果在发送过程中 nCTS 变为无效，则当前发送完成之后，发送器停止。

当 CTSE=1 时，只要 nCTS 发生变化，CTSIF 状态位便会由硬件自动置 1。这指示接收器是否已准备好进行通信。如果 USART_CR3 寄存器中的 CTSIE 位置 1，则会产生中断。下图显示了在使能 CTS 流控制的情况下进行通信的示例。

图 269. CTS



注意： 停止帧的特殊行为：使能 CTS 流后，发送器发送停止信号时将不检查 nCTS 输入状态。

20.4 USART

表 121. USART 中断请求

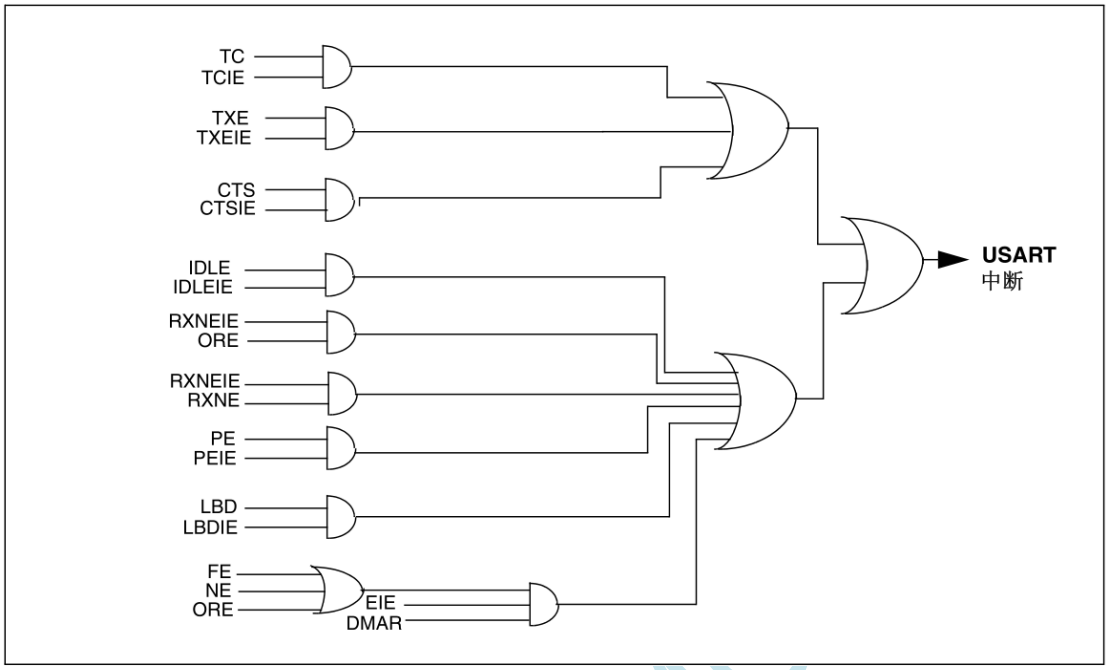
中断事件	事件标志	使能控制位
发送数据寄存器为空	TXE	TXEIE
CTS 标志	CTS	CTS IE
发送完成	TC	TCIE
准备好读取接收到的数据	RXNE	RXNEIE
检测到上溢错误	ORE	
检测到空闲线路	IDLE	IDLEIE
奇偶校验错误	PE	PEIE
断路标志	LBD	LBDIE
多缓冲区通信中的噪声标志、上溢错误和帧错误	NF 或 ORE 或 FE	EIE

USART 中断事件被连接到相同的中断向量（请参见 图 270）。

- 发送期间：发送完成、清除以发送或发送数据寄存器为空中断。
- 接收期间：空闲线路检测、上溢错误、接收数据寄存器不为空、奇偶校验错误、LIN 断路检测、噪声标志（仅限多缓冲区通信）和帧错误（仅限多缓冲区通信）

如果相应的使能控制位置 1，则这些事件会生成中断。

图 270. USART 中断映射图



20.5USART 模式配置

表 122. USART 模式配置

USART 模式	USART1	USART2	USART3	USART4
异步模式	✓	✓	✓	✓
硬件流控	✓	✓	✓	-
DMA	✓	✓	✓	✓
多处理器通信	✓	✓	✓	✓
同步	✓	✓	✓	-
智能卡	✓	✓	✓	-
半双工（单线模式）	✓	✓	✓	✓
IrDA	✓	✓	✓	✓
LIN	✓	✓	✓	✓

20.6USART

20.6.1状态寄存器 (USART_SR)

Status register

偏移地址：0x00

复位值：0x00C0 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CTS	LBD	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
						rc_w0	rc_w0	r	rc_w0	rc_w0	r	r	r	r	r

位 31:10 保留，必须保持复位值

位 9 CTS：CTS 标志 (CTS flag)

如果 CTSE 位置 1，当 nCTS 输入变换时，此位由硬件置 1。通过软件将该位清零（通过向该位中写入 0）。如果 USART_CR3 寄存器中 CTSIE=1，则会生成中断。

0：nCTS 状态线上未发生变化

1：nCTS 状态线上发生变化

注意：该位不适用于 UART4 。

位 8 LBD：LIN 断路检测标志 (LIN break detection flag)

检测到 LIN 断路时，该位由硬件置 1。通过软件将该位清零（通过向该位中写入 0）。如果 USART_CR2 寄存器中 LBDIE = 1，则会生成中断。

0：未检测到 LIN 断路

1：检测到 LIN 断路

注意：如果 LBDIE=1 ，则当 LBD=1 时生成中断

位 7 TXE：发送数据寄存器为空 (Transmit data register empty)

当 TDR 寄存器的内容已传输到移位寄存器时，该位由硬件置 1。如果 USART_CR1 寄存器中 TXEIE 位 = 1，则会生成中断。通过对 USART_DR 寄存器执行写入操作将该位清零。

0：数据未传输到移位寄存器

1：数据传输到移位寄存器

注意：单缓冲区发送期间使用该位。

位 6 TC：发送完成 (Transmission complete)

如果已完成对包含数据的帧的发送并且 TXE 置 1，则该位由硬件置 1。如果 USART_CR1 寄存器中 TCIE = 1，则会生成中断。该位由软件序列清零（读取 USART_SR 寄存器，然后写入 USART_DR 寄存器）。TC 位也可以通过向该位写入‘0’来清零。建议仅在多缓冲区信时使用此清零序列。

0：传送未完成

1：传送已完成

位 5 RXNE：读取数据寄存器不为空 (Read data register not empty)

当 RDR 移位寄存器的内容已传输到 USART_DR 寄存器时，该位由硬件置 1。如果 USART_CR1 寄存器中 RXNEIE = 1，则会生成中断。通过对 USART_DR 寄存器执行读入操作将该位清零。RXNE 标志也可以通过向该位写入零来清零。建议仅在多缓冲区通信时使用此清零序列。

0：未接收到数据

1：已准备好读取接收到的数据

位 4 IDLE：检测到空闲线路 (IDLE line detected)

检测到空闲线路时，该位由硬件置 1。如果 USART_CR1 寄存器中 IDLEIE = 1，则会生成中断。该位由软件序列清零（读入 USART_SR 寄存器，然后读入 USART_DR 寄存器）。

0：未检测到空闲线路

1：检测到空闲线路

注意：直到 RXNE 位本身已置 1 时（即，当出现新的空闲线路时）IDLE 位才会被再次置 1 。

位 3 ORE：上溢错误 (Overrun error)

在 RXNE = 1 的情况下，当移位寄存器中当前正在接收的字准备好传输到 RDR 寄存器时，该位

由硬件置 1。如果 USART_CR1 寄存器中 RXNEIE = 1，则会生成中断。该位由软件序列清零（读入 USART_SR 寄存器，然后读入 USART_DR 寄存器）。

0：无上溢错误

1：检测到上溢错误

注意：当该位置 1 时， RDR 寄存器的内容不会丢失，但移位寄存器会被覆盖。如果 EIE 位置 1 ，

则在进行多缓冲区通信时会对 ORE 标志生成一个中断。

位 2 NF：检测到噪声标志 (Noise detected flag)

当在接收的帧上检测到噪声时，该位由硬件置 1。该位由软件序列清零（读入 USART_SR 寄存器，然后读入 USART_DR 寄存器）。

0：未检测到噪声

1：检测到噪声

注意：如果 EIE 位置 1 ，则在进行多缓冲区通信时，该位不会生成中断，因为该位出现的时间与本身生成中断的 RXNE 位因 NF 标志而生成的时间相同。

注意：当线路无噪声时，可以通过将 ONEBIT 位编程为 1 提高 USART 对偏差的容差来禁止 NF 标志

位 1 FE：帧错误 (Framing error)

当检测到去同步化、过度的噪声或中断字符时，该位由硬件置 1。该位由软件序列清零（读入 USART_SR 寄存器，然后读入 USART_DR 寄存器）。

0：未检测到帧错误

1：检测到帧错误或中断字符

注意：该位不会生成中断，因为该位出现的时间与本身生成中断的 RXNE 位出现的时间相同。

如果当前正在传输的字同时导致帧错误和上溢错误，则会传输该字，且仅有 ORE 位被置 1 。

如果 EIE 位置 1 ，则在进行多缓冲区通信时会对 FE 标志生成一个中断。

位 0 PE：奇偶校验错误 (Parity error)

当在接收器模式下发生奇偶校验错误时，该位由硬件置 1。该位由软件序列清零（读取状态寄存器，然后对 USART_DR 数据寄存器执行读或写访问）。将 PE 位清零前软件必须等待 RXNE 标志被置 1。

如果 USART_CR1 寄存器中 PEIE = 1，则会生成中断。

0：无奇偶校验错误

1：奇偶校验错误

20.6.2数据寄存器 (USART_DR)

Data register

偏移地址：0x04

复位值：0xXXXX XXXX

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

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

位 31:9 保留，必须保持复位值

位 8:0 DR[8:0]：数据值

包含接收到数据字符或已发送的数据字符，具体取决于所执行的操作是“读取”操作还是“写入”操作。

因为数据寄存器包含两个寄存器，一个用于发送 (TDR)，一个用于接收 (RDR)，因此它具有

双重功能（读和写）。
TDR 寄存器在内部总线和输出移位寄存器之间提供了并行接口（参见图 1）。
RDR 寄存器在输入移位寄存器和内部总线之间提供了并行接口。
在使能奇偶校验位的情况下（USART_CR1 寄存器中的 PCE 位被置 1）进行发送时，由于 MSB 的写入值（位 7 或位 8，具体取决于数据长度）会被奇偶校验位所取代，因此该值不起任何作用。
在使能奇偶校验位的情况下进行接收时，从 MSB 位中读取的值为接收到的奇偶校验位。

20.6.3波特率寄存器 (USART_BRR)

Baud rate register
注意： 如果 TE 或 RE 位分别被禁止，则波特计数器会停止计数。
偏移地址：0x08
复位值：0x0000 0000

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

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

位 31:16 保留，必须保持复位值
位 15:4 DIV_Mantissa[11:0]：USARTDIV 的尾数
这 12 个位用于定义 USART 除数 (USARTDIV) 的尾数
位 3:0 DIV_Fraction[3:0]：USARTDIV 的小数
这 4 个位用于定义 USART 除数 (USARTDIV) 的小数。当 OVER8 = 1 时，不考虑 DIV_Fraction3 位，且必须将该位保持清零。

20.6.4控制寄存器 1 (USART_CR1)

Control register 1
偏移地址：0x0C
复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	RES.	UE	M	WAKE	PCE	PS	PEIE	TXEIE	ECIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值
位 15 OVER8：过采样模式 (Oversampling mode)
0：16 倍过采样
1：8 倍过采样
注意： 8 倍过采样在智能卡、 IrDA 和 LIN 模式下不可用：当 SCEN=1 、 IREN=1 或 LINEN=1 时，OVER8 由硬件强制清零。
位 14 保留，必须保持复位值
位 13 UE：USART 使能 (USART enable)
该位清零后，USART 预分频器和输出将停止，并会结束当前字节传输以降低功耗。此位由软件置 1 和清零。
0：禁止 USART 预分频器和输出

1：使能 USART

位 12 M：字长 (Word length)

该位决定了字长。该位由软件置 1 或清零。

0：1 起始位，8 数据位，n 停止位

1：1 起始位，9 数据位，n 停止位

注意：在数据传输（发送和接收）期间不得更改 M 位

位 11 WAKE：唤醒方法 (Wakeup method)

该位决定了 USART 唤醒方法，该位由软件置 1 或清零。

0：空闲线路

1：地址标记

位 10 PCE：奇偶校验控制使能 (Parity control enable)

该位选择硬件奇偶校验控制（生成和检测）。使能奇偶校验控制时，计算出的奇偶校验位被插入到 MSB 位置（如果 M=1，则为第 9 位；如果 M=0，则为第 8 位），并对接收到的数据检查奇偶校验位。此位由软件置 1 和清零。一旦该位置 1，PCE 在当前字节的后面处于活动

状态（在接收和发送时）。

0：禁止奇偶校验控制

1：使能奇偶校验控制

位 9 PS：奇偶校验选择 (Parity selection)

该位用于在使能奇偶校验生成/检测（PCE 位置 1）时选择奇校验或偶校验。该位由软件置 1 和清零。将在当前字节的后面选择奇偶校验。

0：偶校验

1：奇校验

位 8 PEIE：PE 中断使能 (PE interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_SR 寄存器中 PE=1 时，生成 USART 中断

位 7 TXEIE：TXE 中断使能 (TXE interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_SR 寄存器中 TXE=1 时，生成 USART 中断。

位 6 TCIE：传送完成中断使能 (Transmission complete interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_SR 寄存器中 TC=1 时，生成 USART 中断

位 5 RXNEIE：RXNE 中断使能 (RXNE interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_SR 寄存器中 ORE=1 或 RXNE=1 时，生成 USART 中断

位 4 IDLEIE：IDLE 中断使能 (IDLE interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_SR 寄存器中 IDLE=1 时，生成 USART 中断

位 3 TE：发送器使能 (Transmitter enable)

该位使能发送器。该位由软件置 1 和清零。

0：禁止发送器

1：使能发送器

注意： 1 ：除了在智能卡模式下以外，传送期间 TE 位上的“ 0 ”脉冲（“ 0 ”后紧跟的是“ 1 ”）会在当前字的后面发送一个报头（空闲线路）。

2 ：当 TE 置 1 时，在发送开始前存在 1 位的时间延迟。

位 2 RE：接收器使能 (Receiver enable)

该位使能接收器。该位由软件置 1 和清零。

0：禁止接收器
1：使能接收器并开始搜索起始位

位 1 RWU：接收器唤醒 (Receiver wakeup)

该位决定 USART 是否处于静音模式。该位由软件置 1 和清零，并可在识别出唤醒序列时由硬件清零。
0：接收器处于活动模式
1：接收器处于静音模式
注意： 1 ：选择静音模式前（通过将 RWU 位置 1 ）， USART 必须首先接收一个数据字节，否则当由空闲线路检测到唤醒时，它无法于静音模式下正常工作。
2 ：在地址标记检测唤醒配置（ WAKE 位 = 1 ）中， RXNE 位置 1 时， RWU 位不能由软件
进行修改。

位 0 SBK：发送断路 (Send break)

该位用于发送断路字符。该位可由软件置 1 和清零。该位应由软件置 1，并在断路停止位期间由硬件重置。
0：不发送断路字符
1：将发送断路字符

20.6.5控制寄存器 2 (USART_CR2)

Control register 2
偏移地址：0x10
复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	LINEN	STOP[1:0]		CLKEN	CPOL	CPHA	LBCL	Res.	LBDIE	LBDL	Res.	ADD[3:0]			
	rw	rw	rw	rw	rw	rw	rw		rw	rw		rw	rw	rw	rw

位 31:15 保留，必须保持复位值

位 14 LINEN：LIN 模式使能 (LIN mode enable)

此位由软件置 1 和清零。
0：禁止 LIN 模式
1：使能 LIN 模式
LIN 模式可以使用 USART_CR1 寄存器中的 SBK 位发送 LIN 同步断路（13 个低位），并可检测 LIN 同步断路。

位 13:12 STOP：停止位 (STOP bit)

这些位用于编程停止位。
00：1 个停止位
01：0.5 个停止位
10：2 个停止位
11：1.5 个停止位
注意： 0.5 个停止位和 1.5 个停止位不适用于 UART4。

位 11 CLKEN：时钟使能 (Clock enable)

该位允许用户使能 SCLK 引脚。
0：禁止 SCLK 引脚
1：使能 SCLK 引脚
该位不适用于 UART4。

位 10 CPOL：时钟极性 (Clock polarity)

该位允许用户在同步模式下选择 SCLK 引脚上时钟输出的极性。它与 CPHA 位结合使用可获得

STC MCU

所需的时钟/数据关系

0：空闲时 SCLK 引脚为低电平。

1：空闲时 SCLK 引脚为高电平。

该位不适用于 UART4。

位 9 CPHA：时钟相位

该位允许用户在同步模式下选择 SCLK 引脚上时钟输出的相位。它与 CPOL 位结合使用可获得所需的时钟/数据关系（请参见图 258 至 259）

0：在时钟第一个变化沿捕获数据

1：在时钟第二个变化沿捕获数据

注意：该位不适用于 UART4。

位 8 LBCL：最后一个位时钟脉冲 (Last bit clock pulse)

该位允许用户在同步模式下选择与发送的最后一个数据位 (MSB) 关联的时钟脉冲是否必须在 SCLK 引脚上输出。

0：最后一个数据位的时钟脉冲不在 SCLK 引脚上输出

1：最后一个数据位的时钟脉冲在 SCLK 引脚上输出

注意： 1 ：最后一位为发送的第 8 或第 9 个数据位，具体取决于 USART_CR1 寄存器中 M 位所选择的 8 位或 9 位格式。

2 ：该位不适用于 UART4。

位 7 保留，必须保持复位值

位 6 LBDIE：LIN 断路检测中断使能 (LIN break detection interrupt enable)

断路中断屏蔽（使用断路分隔符进行断路检测）

0：禁止中断

1：当 USART_SR 寄存器中 LBD = 1 时，生成中断

位 5 LBDL：lin 断路检测长度 (lin break detection length)

该位用于选择 11 位断路检测或 10 位断路检测。

0：10 位断路检测

1：11 位断路检测

位 4 保留，必须保持复位值

位 3:0 ADD[3:0]：USART 节点的地址

该位域用于指定 USART 节点的地址。

将在多处理器通信时于静音模式下使用该位域，以通过地址标记检测进行唤醒。

注意： 使能发送器时不应对这 3 个位（ CPOL 、 CPHA 、 LBCL ）进行写操作。

20.6.6控制寄存器 3 (USART_CR3)

Control register 3

偏移地址：0x14

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				ONEBIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:12 保留，必须保持复位值

位 11 ONEBIT：一个采样位方法使能 (One sample bit method enable)

该位允许用户选择采样方法。选择一个采样位方法后，将禁止噪声检测标志 (NF)。

0：三个采样位方法

1：一个采样位方法

位 10 CTSIE：CTS 中断使能 (CTS interrupt enable)

0：禁止中断

1：当 USART_SR 寄存器中 CTS = 1 时，生成中断

注意：该位不适用于 UART4。

位 9 CTSE： CTS 使能 (CTS enable)

0： 禁止 CTS 硬件流控制

1： 使能 CTS 模式，仅当 nCTS 输入有效 （连接到 0）时才发送数据。 如果在发送数据时使 nCTS 输入无效，会在停止之前完成发送。 如果使 nCTS 有效时数据已写入数据寄存器，则将延迟发送，直到 nCTS 有效。

注意：该位不适用于 UART4。

位 8 RTSE： RTS 使能 (RTS enable)

0： 禁止 RTS 硬件流控制

1： 使能 RTS 中断，仅当接收缓冲区中有空间时才会请求数据。 发送完当前字符后应停止发送数据。 可以接收数据时使 nRTS 输出有效 （连接到 0）。

注意：该位不适用于 UART4。

位 7 DMAT： DMA 使能发送器 (DMA enable transmitter)

该位由软件置 1/ 复位。

1： 针对发送使能 DMA 模式。

0： 针对发送禁止 DMA 模式。

位 6 DMAR： DMA 使能接收器 (DMA enable receiver)

该位由软件置 1/ 复位。

1： 针对接收使能 DMA 模式

0：针对接收禁止 DMA 模式

位 5 SCEN： 智能卡模式使能 (Smartcard mode enable)

该位用于使能智能卡模式。

0： 禁止智能卡模式

1：使能智能卡模式

注意：该位不适用于 UART4。

位 4 NACK： 智能卡 NACK 使能 (Smartcard NACK enable)

0： 出现奇偶校验错误时禁止 NACK 发送

1：出现奇偶校验错误时使能 NACK 发送

注意：该位不适用于 UART4。

位 3 HDSEL：半双工选择 (Half-duplex selection)

选择单线半双工模式

0：未选择半双工模式

1：选择半双工模式

位 2 IRLP：IrDA 低功耗 (IrDA low-power)

该位用于选择正常模式和低功耗 IrDA 模式

0：正常模式

1：低功耗模式

位 1 IREN：IrDA 模式使能 (IrDA mode enable)

此位由软件置 1 和清零。

0：禁止 IrDA

1：使能 IrDA

位 0 EIE：错误中断使能 (Error interrupt enable)

对于多缓冲区通信（USART_CR3 寄存器中 DMAR = 1），如果发生帧错误、上溢错误或出现噪声标志（USART_SR 寄存器中 FE = 1 或 ORE = 1 或 NF = 1），则需要使用错误中断使能位来使能中断生成。

0：禁止中断

1：当 USART_CR3 寄存器中的 DMAR = 1 并且 USART_SR 寄存器中的 FE = 1 或 ORE = 1 或 NF = 1 时，将生成中断。

STC MCU

20.6.7保护时间和预分频器寄存器 (USART_GTPR)

Guard PWMe and prescaler register

偏移地址：0x18

复位值：0x0000 0000

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

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

位 31:16 保留，必须保持复位值

位 15:8 GT[7:0]： 保护时间值 (Guard PWMe value)

该位域提供保护时间值 （以波特时钟数为单位）。

该位用于智能卡模式。 经过此保护时间后，发送完成标志置 1。

注意：该位不适用于 UART4 。

位 7:0 PSC[7:0]：预分频器值

— 在 IrDA 低功耗模式下：

PSC[7:0] = IrDA 低功耗波特率

用于编程预分频器，进行系统时钟分频以获得低功耗频率：

使用寄存器中给出的值（8 个有效位）对源时钟进行分频：

00000000：保留 - 不编程此值

源时钟 1 分频

00000010：源时钟 2 分频

...

— 在正常 IrDA 模式下：PSC 必须设置为 00000001。

— 在智能卡模式下：

PSC[4:0]：预分频器值

用于编程预分频器，进行系统时钟分频以提供智能卡时钟。

将寄存器中给出的值（5 个有效位）乘以 2 得出源时钟频率的分频系数：

00000：保留 - 不编程此值

00001：源时钟 2 分频

00010：源时钟 4 分频

00011：源时钟 6 分频

...

注意： 1 ：如果使用智能卡模式，则位 [7:5] 不起作用。

2 ：该位不适用于 UART4 。

20.6.8USART 寄存器映射

下表提供了 USART 寄存器映射和复位值。

表 123. USART 寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0X00	USART_SR	Reserved																						CTS	LBD	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
	Reset value																							0	0	1	1	0	0	0	0	0	0

0X04	USART_DR	Reserved														DR[8:0]			
	Reset value															0	0	0	0	0	0	0	0	0
0X08	USART_BRR	Reserved								DIV_Mantissa[15:4]									DIV_Fraction [3:0]			
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0X0C	USART_CR1	Reserved								OVER8	Reserved	UE	M	WAKE	PCE	PS	PEIE	TXEIR	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK
	Reset value									0		0	0	0	0	0	0	0	0	0	0	0	0	0	0
0X10	USART_CR2	Reserved									LINEN	STOP [1:0]		CLKEN	CPOL	CPHA	LBCL	Reserved	LBDIE	LBDL	Reserved	ADD[3:0]			
	Reset value										0	0	0	0	0	0	0		0	0		0	0	0	0
0X14	USART_CR3	Reserved												ONEBIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HOSEL	IRLP	IREN	EIE
	Reset value													0	0	0	0	0	0	0	0	0	0	0	0
0X18	USART_GTPR	Reserved								GT[7:0]								PSC[7:0]							
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

STC MCU

21 DMA 控制器， 支持外设到外设

功能强大的外设到外设的传输模式

21.1DMA 简介

直接存储器访问 (DMA) 用于在外设与存储器之间以及存储器与存储器之间提供高速数据传输。可以在无需任何 CPU 操作的情况下通过 DMA 快速移动数据。这样节省的 CPU 资源可供其它操作使用。DMA 控制器基于复杂的总线矩阵架构，将功能强大的双 AHB 主总线架构与独立的 FIFO 结合在一起，优化了系统带宽。

两个 DMA 控制器总共有 16 个数据流（每个控制器 8 个），每一个 DMA 控制器都用于管理一个或多个外设的存储器访问请求。每个数据流总共可以有多达 8 个通道（或称请求）。每个通道都有一个仲裁器，用于处理 DMA 请求间的优先级。

21.2DMA 主要特性

DMA 主要特性是：

- 双 AHB 主总线架构，一个用于存储器访问，另一个用于外设访问
- 仅支持 32 位访问的 AHB 从编程接口
- 每个 DMA 控制器有 8 个数据流，每个数据流有多达 8 个通道（或称请求）
- 每个数据流有单独的四级 32 位先进先出存储器缓冲区 (FIFO)，可用于 FIFO 模式或直接模式：
  - FIFO 模式：可通过软件将阈值级别选取为 FIFO 大小的 1/4、1/2 或 3/4
  - 直接模式
    每个 DMA 请求会立即启动对存储器的传输。当在直接模式（禁止 FIFO）下将 DMA 请求配置为以存储器到外设模式传输数据时，DMA 仅会将一个数据从存储器预加载到内部 FIFO，从而确保一旦外设触发 DMA 请求时则立即传输数据。
- 通过硬件可以将每个数据流配置为：
  - 支持外设到存储器、存储器到外设和存储器到存储器传输的常规通道
  - 也支持在存储器方双缓冲的双缓冲区通道
- 8 个数据流中的每一个都连接到专用硬件 DMA 通道（请求）
- DMA 数据流请求之间的优先级可用软件编程（4 个级别：非常高、高、中、低），在软件优先级相同的情况下可以通过硬件决定优先级（例如，请求 0 的优先级高于请求 1）
- 每个数据流也支持通过软件触发存储器到存储器的传输
- 可供每个数据流选择的通道请求多达 8 个。此选择可由软件配置，允许几个外设启动 DMA 请求
- 要传输的数据项的数目可以由 DMA 控制器或外设管理：
  - DMA 流控制器：要传输的数据项的数目是 1 到 4095，可用软件编程
  - 外设流控制器：要传输的数据项的数目未知并由源或目标外设控制，这些外设通过硬件发出传输结束的信号
- 独立的源和目标传输宽度（字节、半字、字）：源和目标的数据宽度不相等时，DMA 自动封装/ 解封必要的传输数据来优化带宽。这个特性仅在 FIFO 模式下可用

- 对源和目标的增量或非增量寻址
- 支持 4 个、8 个和 16 个节拍的增量突发传输。突发增量的大小可由软件配置，通常等于外设 FIFO 大小的一半
- 每个数据流都支持循环缓冲区管理
- 5 个事件标志（DMA 半传输、DMA 传输完成、DMA 传输错误、DMA FIFO 错误、直接模式错误），进行逻辑或运算，从而产生每个数据流的单个中断请求

21.3DMA 功能说明

21.3.1一般说明

DMA 控制器执行直接存储器传输：因为采用 AHB 主总线，它可以控制 AHB 总线矩阵来启动 AHB 事务。

它可以执行下列事务：

- 外设到存储器的传输
- 存储器到外设的传输
- 存储器到存储器的传输
- 外设到外设的传输

DMA 控制器提供两个 AHB 主端口：AHB 存储器端口（用于连接存储器）和 AHB 外设端口（用于连接外设）。但是，要执行存储器到存储器的传输，AHB 外设端口必须也能访问存储器。

AHB 从端口用于对 DMA 控制器进行编程（它仅支持 32 位访问）。

21.3.2DMA 事务

DMA 事务由给定数目的数据传输序列组成。要传输的数据项的数目及其宽度（8 位、16 位或 32 位）可用软件编程。

每个 DMA 传输包含三项操作：

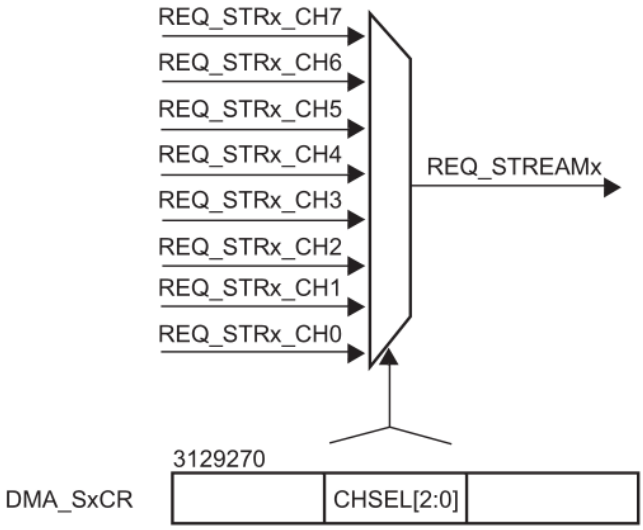
- 通过 DMA_SxPAR 或 DMA_SxM0AR 寄存器寻址，从外设数据寄存器或存储器单元中加载数据。
- 通过 DMA_SxPAR 或 DMA_SxM0AR 寄存器寻址，将加载的数据存储到外设数据寄存器或存储器单元。
- DMA_SxNDTR 计数器在数据存储结束后递减，该计数器中包含仍需执行的事务数。

在产生事件后，外设会向 DMA 控制器发送请求信号。DMA 控制器根据通道优先级处理该请求。只要 DMA 控制器访问外设，DMA 控制器就会向外设发送确认信号。外设获得 DMA 控制器的确认信号后，便会立即释放其请求。一旦外设使请求失效，DMA 控制器就会释放确认信号。如果有更多请求，外设可以启动下一个事务。

21.3.3通道选择

每个数据流都与一个 DMA 请求相关联，此 DMA 请求可以从 8 个可能的通道请求中选出。此选择由 DMA_SxCR 寄存器中的 CHSEL[2:0] 位控制。

图 27. 通道选择



来自外设的 8 个请求（PWM、ADC、SPI、I2C 等）独立连接到每个通道，具体的连接取决于产品实现情况。

表 36 给出了 DMA 请求映射的示例。

表 36. DMA2 请求映射

外设请求	数据流 0	数据流 1	数据流 2	数据流 3	数据流 4	数据流 5	数据流 6	数据流 7
通道 0	PWM1_CH3	PWM8_UP	PWM8_CH1	SPI2_RX	SPI2_TX	PWM8_CH2	PWM8_CH3	PWM8_CH4 PWM8_TRIG PWM8_COM
通道 1	I2C1_RX	PWM1_CH2	PWM1_CH1	PWM1_CH4 PWM1_TRIG PWM1_COM	PWM1_UP	I2C1_RX	I2C1_TX	I2C1_TX
通道 2	PWM4_CH1	PWM1_CH1	I2S3_EXT_RX	PWM4_CH2	I2S2_EXT_TX	I2S3_EXT_TX	PWM4_UP	PWM4_CH3
通道 3	I2S3_EXT_RX	PWM2_UP PWM2_CH3	QSPI	I2S2_EXT_RX	DCMI	PWM2_CH1	PWM2_CH2 PWM2_CH4	PWM2_UP PWM2_CH4
通道 4	USART1_TX	USART3_RX	UART4_RX	USART3_TX	UART4_TX	USART2_RX	USART2_TX	PWM1_TRIG
通道 5	USART1_RX	USART1_RX	PWM3_CH4 PWM3_UP	SPI1_TX	PWM3_CH1 PWM3_TRIG	PWM3_CH2	SPI1_TX	PWM3_CH3
通道 6	PWM5_CH3 PWM5_UP	PWM5_CH4 PWM5_TRIG	PWM5_CH1	PWM5_CH4 PWM5_TRIG	PWM5_CH2	SPI1_RX	PWM5_UP	SPI1_RX
通道 7	ADC1	PWM8_CH1 PWM8_CH2 PWM8_CH3	I2C2_RX	I2C2_RX	USART3_TX	ADC1	PWM1_CH1 PWM1_CH2 PWM1_CH3	I2C2_TX

DMA2 外设到外设

外设请求	数据流 0	数据流 1	数据流 2	数据流 3	数据流 4	数据流 5	数据流 6	数据流 7
请求源	SPI1_RX	SPI2_RX	I2C1_RX	I2C2_RX	USART1_RX	USART2_RX	USART3_RX	USART4_RX
通道 0	Des	Des	Des	Des	Des	Des	Des	Des

21.3.4仲裁器

仲裁器为两个 AHB 主端口（存储器和外设端口）提供基于请求优先级的 8 个DMA 数据流请求管理，并启动外设/ 存储器访问序列。

优先级管理分为两个阶段：

- 软件：每个数据流优先级都可以在 DMA_SxCR 寄存器中配置。分为四个级别：
  - 非常高优先级
  - 高优先级
  - 中优先级
  - 低优先级
- 硬件：如果两个请求具有相同的软件优先级，则编号低的数据流优先于编号高的数据流。例如，数据流 2 的优先级高于数据流 4。

21.3.5DMA 数据流

8 个 DMA 控制器数据流都能够提供源和目标之间的单向传输链路。
每个数据流配置后都可以执行：

- 常规类型事务：存储器到外设、外设到外设、外设到存储器或存储器到存储器的传输。
- 双缓冲区类型事务：使用存储器的两个存储器指针的双缓冲区传输（当DMA 正在进行自/至缓冲区的读/ 写操作时，应用程序可以进行至/ 自其它缓冲区的写/ 读操作）。

要传输的数据量可以编程，并与连接到外设 AHB 端口的外设（请求 DMA 传输）的源宽度相关。每个事务完成后，包含要传输的数据项总量的寄存器都会递减。

21.3.6源、目标和传输模式

源传输和目标传输在整个 4 GB 区域（地址在 0x0000 0000 和 0xFFFF FFFF 之间）都可以寻址外设和存储器。
传输方向使用 DMA_SxCR 寄存器中的 DIR[1:0] 位进行配置，有三种可能的传输方向：存储器到外设、外设到存储器或存储器到存储器。*表 37* 介绍了相应的源和目标地址。

表 37. 源和目标地址

DMA_SxCR 寄存器的位 DIR[1:0]	方向	源地址	目标地址
00	外设到存储器	DMA_SxPAR	DMA_SxM0AR
01	存储器到外设	DMA_SxM0AR	DMA_SxPAR
10	存储器到存储器	DMA_SxPAR	DMA_SxM0AR

11	外设到外设	DMA_SxPAR	DMA_SxPAR

当数据宽度（在 DMA_SxCR 寄存器的 PSIZE 或 MSIZE 位中编程）分别是半字或字时，写入 DMA_SxPAR 或DMA_SxM0AR/M1AR 寄存器的外设或存储器地址必须分别在字或半字地址的边界对齐。

外设到存储器模式

图 28 介绍了这种模式。

使能这种模式（将 DMA_SxCR 寄存器中的位 EN 置 1）时，每次产生外设请求，数据流都会启动数据源到 FIFO 的传输。

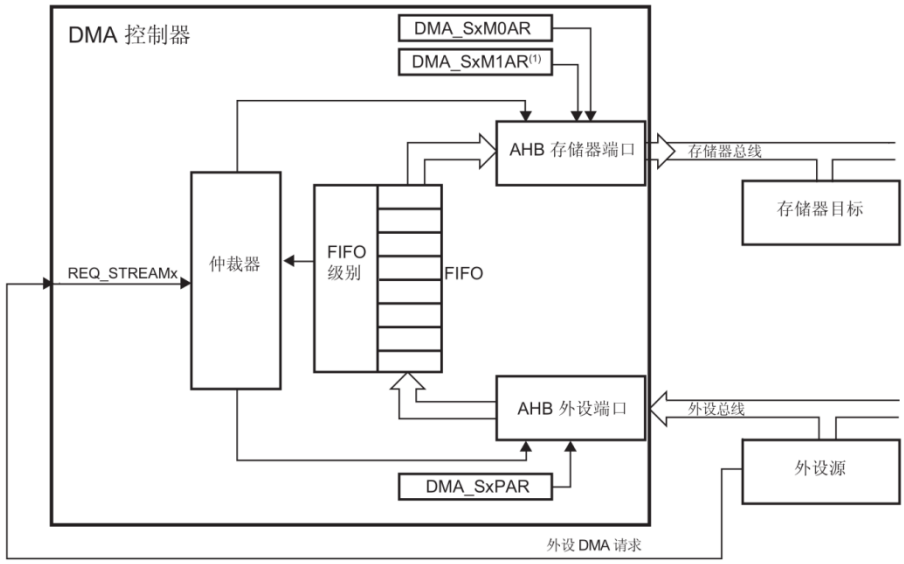
达到 FIFO 的阈值级别时，FIFO 的内容移出并存储到目标中。

如果 DMA_SxNDTR 寄存器达到零、外设请求传输终止（在使用外设流控制器的情况下）或 DMA_SxCR 寄存器中的 EN 位由软件清零，传输即会停止。

在直接模式下（当 DMA_SxFCR 寄存器中的 DMDIS 值为“0”时），不使用 FIFO 的阈值级别控制：每完成一次从外设到 FIFO 的数据传输后，相应的数据立即就会移出并存储到目标中。

只有赢得了数据流的仲裁后，相应数据流才有权访问 AHB 源或目标端口。系统使用在 DMA_SxCR 寄存器 PL[1:0] 位中为每个数据流定义的优先级执行仲裁。

图 28. 外设到存储器模式



1. 用于双缓冲区模式。

存储器到外设模式

图 29 介绍了这种模式。

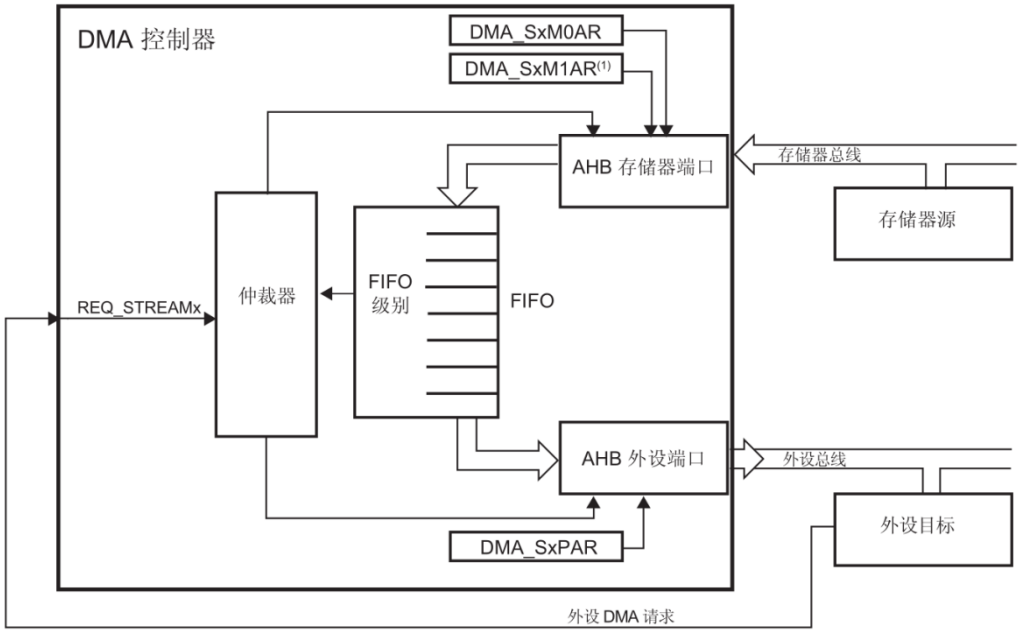
使能这种模式（将DMA_SxCR 寄存器中的 EN 位置1）时，数据流会立即启动传输，从源完全填充 FIFO。每次发生外设请求，FIFO 的内容都会移出并存储到目标中。当 FIFO 的级别小于或等于预定义的阈值级别时，将使用存储器中的数据完全重载 FIFO。

如果 DMA_SxNDTR 寄存器达到零、外设请求传输终止（在使用外设流控制器的情况下）或 DMA_SxCR 寄存器中的 EN 位由软件清零，传输即会停止。

在直接模式下（当DMA_SxFCR 寄存器中的 DMDIS 值为“0”时），不使用 FIFO 的阈值级别。一旦使能了数据流，DMA 便会预装载第一个数据，将其传输到内部 FIFO。一旦外设请求数据传输，DMA 便会将预装载的值传输到配置的目标。然后，它会使用要传输的下一个数据再次重载内部空 FIFO。预装载的数据大小为 DMA_SxCR 寄存器中 PSIZE 位字段的值。

只有赢得了数据流的仲裁后，相应数据流才有权访问 AHB 源或目标端口。系统使用在 DMA_SxCR 寄存器 PL[1:0] 位中为每个数据流定义的优先级执行仲裁。

图 29. 存储器到外设模式



1. 用于双缓冲区模式。

外设到外设模式

图 30 介绍了这种模式。

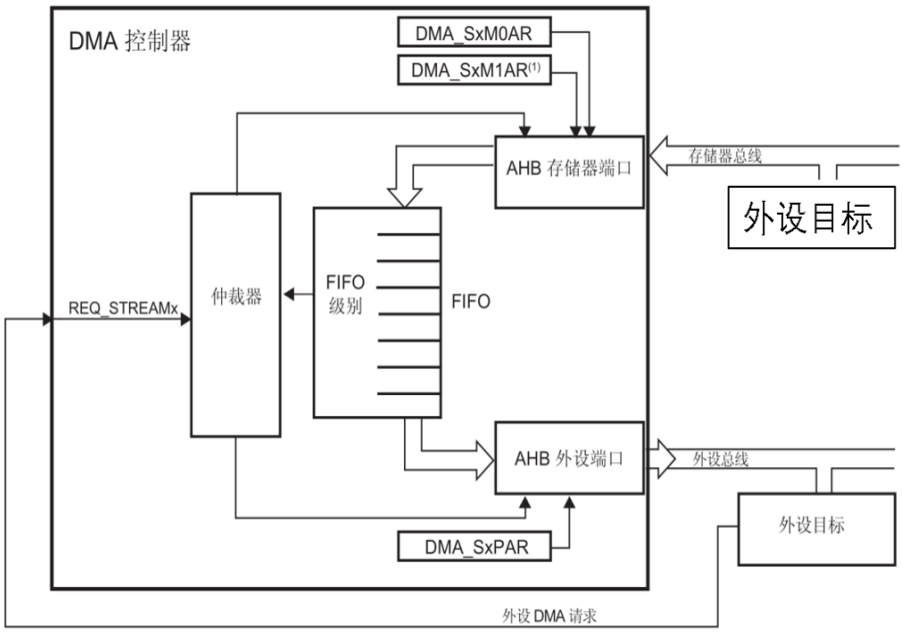
使能这种模式（将DMA_SxCR 寄存器中的 EN 位置 1）时，数据流会立即启动传输，从源完全填充 FIFO。每次发生外设请求，FIFO 的内容都会移出并存储到目标中。当 FIFO 的级别小于或等于预定义的阈值级别时，将使用存储器中的数据完全重载 FIFO。

如果 DMA_SxNDTR 寄存器达到零、外设请求传输终止（在使用外设流控制器的情况下）或 DMA_SxCR 寄存器中的 EN 位由软件清零，传输即会停止。

在直接模式下（当DMA_SxFCR 寄存器中的 DMDIS 值为“0”时），不使用 FIFO 的阈值级别。一旦使能了数据流，DMA 便会预装载第一个数据，将其传输到内部 FIFO。一旦外设请求数据传输，DMA 便会将预装载的值传输到配置的目标。然后，它会使用要传输的下一个数据再次重载内部空 FIFO。预装载的数据大小为 DMA_SxCR 寄存器中 PSIZE 位字段的值。

只有赢得了数据流的仲裁后，相应数据流才有权访问 AHB 源或目标端口。系统使用在 DMA_SxCR 寄存器 PL[1:0] 位中为每个数据流定义的优先级执行仲裁。

图 30. 外设到外设模式



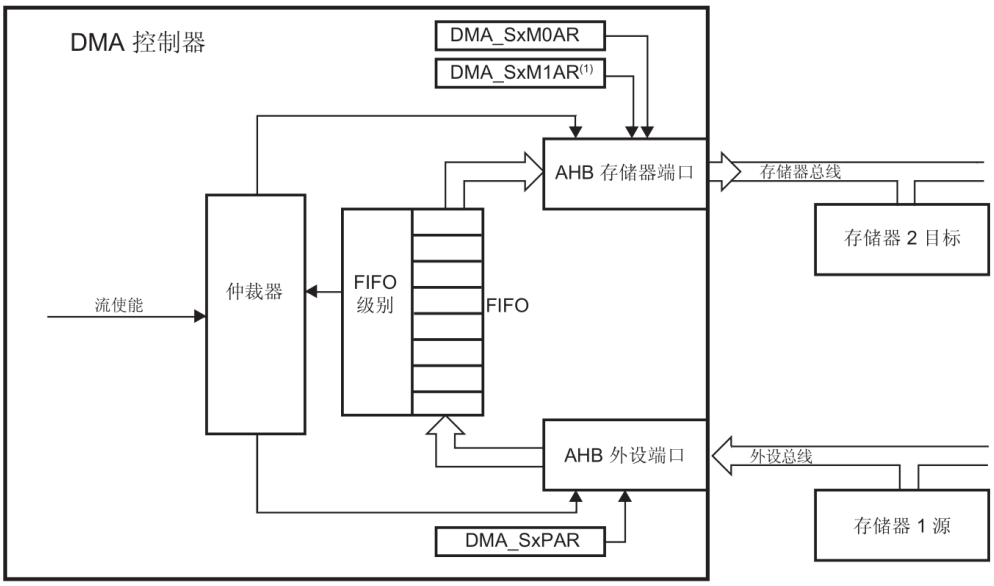
存储器到存储器模式

DMA 通道在没有外设请求触发的情况下同样可以工作。此为*图 31* 中介绍的存储器到存储器模式。通过将 DMA_SxCR 寄存器中的使能位 (EN) 置 1 来使能数据流时，数据流会立即开始填充 FIFO，直至达到阈值级别。达到阈值级别后，FIFO 的内容便会移出，并存储到目标中。
如果 DMA_SxNDTR 寄存器达到零或 DMA_SxCR 寄存器中的 EN 位由软件清零，传输即会停止。

只有赢得了数据流的仲裁后，相应数据流才有权访问 AHB 源或目标端口。系统使用在 DMA_SxCR 寄存器 PL[1:0] 位中为每个数据流定义的优先级执行仲裁。

注意： *使用存储器到存储器模式时，不允许循环模式和直接模式。*
只有 DMA2 控制器能够执行存储器到存储器的传输。

图 31. 存储器到存储器模式



1. 用于双缓冲区模式。

21.3.7指针递增

根据 DMA_SxCR 寄存器中 PINC 和 MINC 位的状态，外设和存储器指针在每次传输后可以自动向后递增或保持常量。

通过单个寄存器访问外设源或目标数据时，禁止递增模式十分有用。

如果使能了递增模式，则根据在 DMA_SxCR 寄存器 PSIZE 或 MSIZE 位中编程的数据宽度，下一次传输的地址将是前一次传输的地址递增 1（对于字节）、2（对于半字）或 4（对于字）。

为了优化封装操作，可以不管 AHB 外设端口上传输的数据的大小，将外设地址的增量偏移大小固定下来。DMA_SxCR 寄存器中的 PINCOS 位用于将增量偏移大小与外设 AHB 端口或 32 位地址（此时地址递增 4）上的数据大小对齐。PINCOS 位仅对 AHB 外设端口有影响。

如果将 PINCOS 位置 1，则不论 PSIZE 值是多少，下一次传输的地址总是前一次传输的地址递增 4（自动与 32 位地址对齐）。但是，AHB 存储器端口不受此操作影响。

如果 AHB 外设端口或 AHB 存储器端口分别请求突发事务，为了满足 AMBA 协议（在固定地址模式下不允许突发事务），则需要将 PINC 或MINC 位置 1。

21.3.8循环模式

循环模式可用于处理循环缓冲区和连续数据流（例如 ADC 扫描模式）。可以使用 DMA_SxCR 寄存器中的 CIRC 位使能此特性。

当激活循环模式时，要传输的数据项的数目在数据流配置阶段自动用设置的初始值进行加载， 并继续响应 DMA 请求。

注意： 在循环模式下，如果为存储器配置了突发模式，必须遵循下列规则：

DMA_SxNDTR = ((Mburst 节拍) × (Msize)/(Psize)) 的倍数，其中：

- *(Mburst 节拍) = 4、8 或16（取决于 DMA_SxCR 寄存器中的 MBURST 位）*
- *((Msize)/(Psize)) = 1、2、4、1/2 或1/4（Msize 和Psize 表示DMA_SxCR 寄存器中的MSIZE 和PSIZE 位。它们与字节相关）*
- *DMA_SxNDTR = AHB 外设端口上要传输的数据项的数目*

例如：Mburst 节拍= 8 (INCR8)，MSIZE = “00”（字节）和PSIZE = “01”（半字），在此例中：DMA_SxNDTR 必须是 (8 × 1/2 = 4) 的倍数。

如果不遵循此公式，则 DMA 行为和数据完整性得不到保证。

NDTR 还必须是外设突发大小与外设数据大小乘积的倍数，否则会导致错误的 DMA 行为。

21.3.9双缓冲区模式

此模式可用于所有 DMA1 和 DMA2 数据流。
通过将 DMA_SxCR 寄存器中的 DBM 位置 1，即可使能双缓冲区模式。
除了有两个存储器指针之外，双缓冲区数据流的工作方式与常规（单缓冲区）数据流的一样。使能双缓冲区模式时，将自动使能循环模式（DMA_SxCR 中的 CIRC 位的状态是“无关”），并在每次事务结束时交换存储器指针。
在此模式下，每次事务结束时，DMA 控制器都从一个存储器目标交换为另一个存储器目标。这样，软件在处理一个存储器区域的同时，DMA 传输还可以填充/ 使用第二个存储器区域。如*表 38：双缓冲区模式下的源和目标地址寄存器 (DBM=1)* 所述，双缓冲区数据流可以双向工作（存储器既可以是源也可以是目标）。

注意： *在双缓冲区模式下使能数据流时，可遵循下列条件，实时更新 AHB 存储器的基址（DMA_SxM0AR 或 DMA_SxM1AR）：*

- *当DMA_SxCR 寄存器中的 CT 位为“0”时，可以写入 DMA_SxM1AR 寄存器。当 CT = “1”时，试图写入此寄存器会将错误标志位 (TEIF) 置 1，并自动禁止数据流。*
- *当DMA_SxCR 寄存器中的 CT 位为“1”时，可以写入 DMA_SxM0AR 寄存器。当 CT = “0”时，试图写入此寄存器会将错误标志位 (TEIF) 置 1，并自动禁止数据流。*

为避免出现任何错误状态，建议在 TCIF 标志位置位时立即更改基址。因为此时根据上述两个条件之一，目标存储器依据 DMA_SxCR 寄存器中 CT 值的情况，一定已从存储器 0 更改为存储器 1（或从存储器 1 更改为存储器 0）。
对于所有其它模式（双缓冲区模式除外），一旦使能数据流，存储器地址寄存器即被写保护。

表 38. 双缓冲区模式下的源和目标地址寄存器 (DBM=1)

DMA_SxCR 寄存器的位 DIR[1:0]	方向	源地址	目标地址
00	外设到存储器	DMA_SxPAR	DMA_SxM0AR/DMA_SxM1AR
01	存储器到外设	DMA_SxM0AR/DMA_SxM1AR	DMA_SxPAR
10	不允许(1)		
11	保留	-	-

1. 使能双缓冲区模式时，自动使能循环模式。由于存储器到存储器模式与循环模式不兼容，所以当使能双缓冲区模式时，不允许配置存储器到存储器模式。

21.3.10可编程数据宽度、封装/ 解封、字节序

要传输的数据项数目必须在使能数据流之前编程到 DMA_SxNDTR（要传输数据项数目位， NDT）中，当流控制器是外设且 DMA_SxCR 中的 PFCTRL 位置为 1 时除外。
当使用内部 FIFO 时，源和目标数据的数据宽度可以通过 DMA_SxCR 寄存器的 PSIZE 和 MSIZE 位（可以是 8、16 或 3 位）编程。
当 PSIZE 和 MSIZE 不相等时：

- 在DMA_SxNDTR 寄存器中配置的要传输的数据项数目的数据宽度等于外设总线的宽度（由 DMA_SxCR 寄存器中的 PSIZE 位配置）。例如，在外设到存储器、存储器到外设或存储器到存储器传输的情况下，如果将 PSIZE[1:0] 位配置为半字，则要传输的字节数等于 2 × NDT。
- DMA 控制器仅按小字节序寻址源和目标。相关内容在*表 39: 封装/解封和字节序行为（位PINC = MINC = 1）*中介绍。

在封装/ 解封数据的过程中，如果在数据完全封装/ 解封前中断操作，则有数据损坏的危险。因此，为了确保数据一致性，可将数据流配置成生成突发传输：在这种情况下，属于一个突发的每组传输不可分割。

在直接模式下（DMA_SxFCR 寄存器中的 DMDIS = 0），不能进行数据封装/ 解封。这种情况下，不允许源与目标的传输数据宽度不同，二者必须相等，并由 DMA_SxCR 中的 PSIZE 位定义，MSIZE 位的状态是“无关”。

表 39. 封装/ 解封和字节序行为（位 PINC = MINC = 1）

AHB 存储器端口宽度	AHB 外设端口宽度	要传输的数据项的数目(NDT)	存储器传输数目	存储器端口地址/字节通道	外设传输数目	外设端口地址/ 字节通道	
						PINCOS = 1	PINCOS = 0
8	8	4	1 2 3 4	0x0/B0[7:0] 0x1/B1[7:0] 0x2/B2[7:0] 0x3/B3[7:0]	1 2 3 4	0x0/B0[7:0] 0x4/B1[7:0] 0x8/B2[7:0] 0xC/B3[7:0]	0x0/B0[7:0] 0x1/B1[7:0] 0x2/B2[7:0] 0x3/B3[7:0]
8	16	2	1 2 3 4	0x0/B0[7:0] 0x1/B1[7:0] 0x2/B2[7:0] 0x3/B3[7:0]	1 2	0x0/B1|B0[15:0] 0x4/B3|B2[15:0]	0x0/B1|B0[15:0] 0x2/B3|B2[15:0]
8	32	1	1 2 3 4	0x0/B0[7:0] 0x1/B1[7:0] 0x2/B2[7:0] 0x3/B3[7:0]	1	0x0/B3|B2|B1|B0[31:0]	0x0/B3|B2|B1|B0[31:0]
16	8	4	1 2	0x0/B1|B0[15:0] 0x2/B3|B2[15:0]	1 2 3 4	0x0/B0[7:0] 0x4/B1[7:0] 0x8/B2[7:0] 0xC/B3[7:0]	0x0/B0[7:0] 0x1/B1[7:0] 0x2/B2[7:0] 0x3/B3[7:0]
16	16	2	1 2	0x0/B1|B0[15:0] 0x2/B1|B0[15:0]	1 2	0x0/B1|B0[15:0] 0x4/B3|B2[15:0]	0x0/B1|B0[15:0] 0x2/B3|B2[15:0]
16	32	1	1 2	0x0/B1|B0[15:0] 0x2/B3|B2[15:0]	1	0x0/B3|B2|B1|B0[31:0]	0x0/B3|B2|B1|B0[31:0]
32	8	4	1	0x0/B3|B2|B1|B0[31:0]	1 2 3 4	0x0/B0[7:0] 0x4/B1[7:0] 0x8/B2[7:0] 0xC/B3[7:0]	0x0/B0[7:0] 0x1/B1[7:0] 0x2/B2[7:0] 0x3/B3[7:0]

STC MCU

表39. 封装 / 解封和字节序行为（位PINC = MINC = 1）（续）

AHB 存储器端口宽度	AHB 外设端口宽度	要传输的数据项的数目(NDT)	存储器传输数目	存储器端口地址/字节通道	外设传输数目	外设端口地址/ 字节通道	
						PINCOS = 1	PINCOS = 0
32	16	2	1	0x0/B3|B2|B1|B0[31:0]	1 2	0x0/B1|B0[15:0] 0x4/B3|B2[15:0]	0x0/B1|B0[15:0] 0x2/B3|B2[15:0]
32	32	1	1	0x0/B3|B2|B1|B0[31:0]	1	0x0/B3|B2|B1|B0[31:0]	0x0/B3|B2|B1|B0[31:0]

注意： *外设端口可以是源或目标（在存储器到存储器传输的情况下，也可能是存储器源）。* 必须配置 PSIZE、MSIZE 和 NDT[15:0]，以确保最后一次传输的完整性。当外设端口的数据宽度（PSIZE 位）小于存储器端口的数据宽度（MSIZE 位）时，可能会发生数据传输不完整的情况。此限制条件汇总在 *表40* 中。

表40. PSIZE 与MSIZE 确定时对NDT 的限制条件

DMA_SxCR 的 PSIZE[1:0]	DMA_SxCR 的 MSIZE[1:0]	DMA_SxNDTR 的 NDT[15:0]
00（8 位）	01（16 位）	必须是 2 的倍数
00（8 位）	10（32 位）	必须是 4 的倍数
01（16 位）	10（32 位）	必须是 2 的倍数

21.3.11 单次传输和突发传输

DMA 控制器可以产生单次传输或 4 个、8 个和 16 个节拍的增量突发传输。
突发大小通过软件针对两个 AHB 端口独立配置，配置时使用 DMA_SxCR 寄存器中的 MBURST[1:0] 和 PBURST[1:0] 位。
突发大小指示突发中的节拍数，而不是传输的字节数。
为确保数据一致性，形成突发的每一组传输都不可分割：在突发传输序列期间，AHB 传输会锁定，并且 AHB 总线矩阵的仲裁器不解除对 DMA 主总线的授权。
根据单次或突发配置的情况，每个 DMA 请求在 AHB 外设端口上相应地启动不同数量的传输。

- 当 AHB 外设端口被配置为单次传输时，根据 DMA_SxCR 寄存器 PSIZE[1:0] 位的值，每个 DMA 请求产生一次字节、半字或字的数据传输。
- 当AHB 外设端口被配置为突发传输时，根据 DMA_SxCR 寄存器 PBURST[1:0] 和PSIZE[1:0] 位的值，每个 DMA 请求相应地生成 4 个、8 个或 16 个节拍的字节、半字或字的传输。

对于需要配置 MBURST 和 MSIZE 位的 AHB 存储器端口，必须考虑与上述相同的内容。
在直接模式下，数据流只能生成单次传输，而 MBURST[1:0] 和 PBURST[1:0] 位由硬件强制配置。
必须选择地址指针（DMA_SxPAR 或 DMA_SxM0AR 寄存器），以确保一个突发块内的所有传输在等于传输大小的地址边界对齐。
选择突发配置必须要遵守 AHB 协议，即突发传输*不得*越过 1 KB 地址边界，因为可以分配给单个从设备的最小地址空间是 1 KB。这意味着突发块传输不应越过 1 KB 地址边界，否则就会产生一个 AHB 错误，并且 DMA 寄存器不会报告这个错误。

注意： *仅在使能指针递增模式时允许突发模式：*

- *当PINC 位为“0”时，也应将 PBURST 位清为“00”*
- *当MINC 位为“0”时，也应将 MBURST 位清为“00”*

21.3.12 FIFO

FIFO 结构

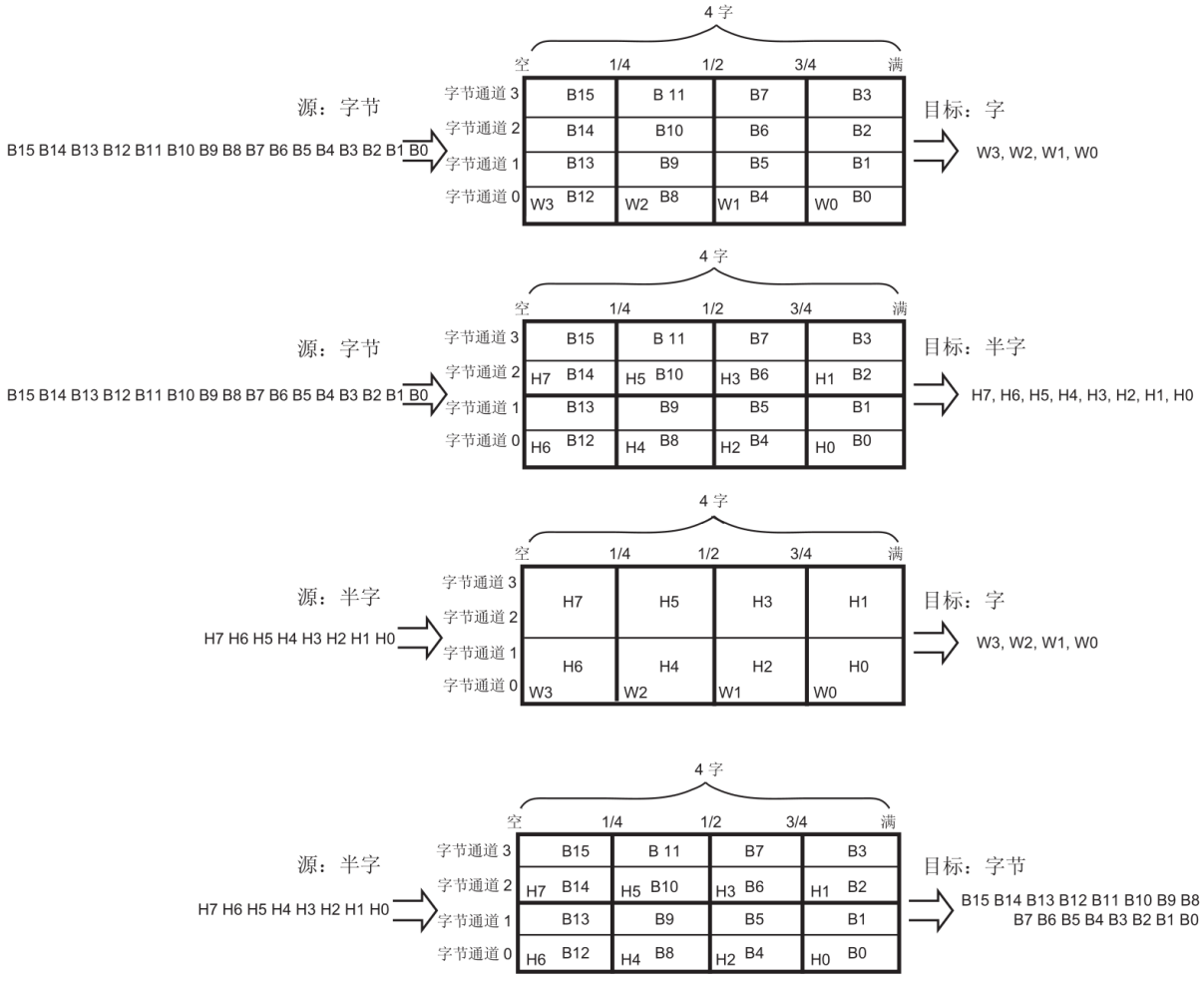
FIFO 用于在源数据传输到目标之前临时存储这些数据。

每个数据流都有一个独立的 4 字 FIFO，阈值级别可由软件配置为 1/4、1/2、3/4 或满。

为了使能 FIFO 阈值级别，必须通过将 DMA_SxFCR 寄存器中的 DMDIS 位置 1 来禁止直接模式。

FIFO 的结构随源与目标数据宽度而不同，相关内容在*图31：FIFO 结构*中介绍。

图31.FIFO 结构



21.3.12.1 FIFO 阈值与突发配置

选择 FIFO 阈值（DMA_SxFCR 寄存器的位 FTH[1:0]）和存储器突发大小（DMA_SxCR 寄存器的 MBURST[1:0] 位）时需要小心：FIFO 阈值指向的内容必须与整数个存储器突发传输完全匹配。如果不是这样，当使能数据流时将生成一个 FIFO 错误（DMA_HISR 或DMA_LISR 寄存器的标志 FEIFx），

然后将自动禁止数据流。允许的和禁止的配置在*表41：FIFO 阈值配置*中介绍。

表41. FIFO 阈值配置

MSIZE	FIFO 级别	MBURST = INCR4	MBURST = INCR8	MBURST = INCR16
字节	1/4	4 个节拍的 1 次突发	禁止	禁止
	1/2	4 个节拍的 2 次突发	8 个节拍的 1 次突发	
	3/4	4 个节拍的 3 次突发	禁止	
	满	4 个节拍的 4 次突发	8 个节拍的 2 次突发	16 个节拍的 1 次突发
半字	1/4	禁止	禁止	禁止
	1/2	4 个节拍的 1 次突发		
	3/4	禁止		
	满	4 个节拍的 2 次突发	8 个节拍的 1 次突发	
字	1/4	禁止	禁止	
	1/2			
	3/4			
	满	4 个节拍的 1 次突发		

所有这些情况下，突发大小与数据大小的乘积不得超过 FIFO 大小（数据大小可以为：1（字节）、2（半字）或 4（字））。

如果发生下列一种情况，会导致 DMA 传输结束时出现不完整的突发传输：

- 对于 AHB 外设端口配置：数据项总数（在DMA_SxNDTR 寄存器中设置）不是突发大小与数据大小乘积的倍数。
- 对于 AHB 存储器端口配置：要传输到存储器的 FIFO 中的剩余数据项的数目不是突发大小与数据大小乘积的倍数。

在这些情况下，即使在 DMA 数据流配置期间请求突发事务，要传输的剩余数据也将由 DMA 在单独模式下管理。

注意： *当在外设AHB 端口上请求突发传输并且使用FIFO（DMA_SxCR 寄存器中DMDIS = 1）时， 必须根据DMA 数据流方向遵守下列规则来避免出现上溢或下溢情况：*

如果 (PBURST × PSIZE) = FIFO_SIZE（4 字），则当 PSIZE = 1、2 或 4，PBURST = 4、8 或 16 时，禁止 FIFO_Threshold = 3/4。

此规则将确保一次释放足够的 FIFO 空间来处理外设的请求。

21.3.12.2 FIFO 刷新

当复位 DMA_SxCR 寄存器中的 EN 位来禁止数据流，以及配置数据流来管理外设到存储器或存储器到存储器的传输时，可以刷新 FIFO。因为如果禁止数据流时仍有某些数据存留在 FIFO 中，DMA

控制器会将剩余的数据继续传输到目标（即使已经有效禁止了数据流）。刷新完成时，会将 DMA_LISR 或 DMA_HISR 寄存器中的传输完成状态位 (TCIFx) 置 1。

在这种情况下，剩余数据计数器 DMA_SxNDTR 保持的值指示在目标存储器现有多少可用数据项。请注意在 FIFO 刷新操作期间，如果 FIFO 中要传输到存储器的剩余数据项的数目（以字节为单位）小于存储器数据宽度（例如在 MSIZE 配置为字时 FIFO 中为 2 个字节），则会使用在DMA_SxCR 寄存器中的 MSIZE 位设置的数据宽度发送数据。这意味着将使用非预期值写入存储器。软件可以读取 DMA_SxNDTR 寄存器来确定包含良好数据的存储器区域（起始地址和最后地址）。
如果 FIFO 中剩余数据项的数目小于突发大小，并且数据流通过将 DMA_SxCR 寄存器 MBURST 位置 1 进行配置来管理在 AHB 存储器端口上的突发传输，则使用单次传输来完成 FIFO 的刷新。

21.3.12.3 直接模式

默认情况下，FIFO 以直接模式操作（将 DMA_SxFCR 中的 DMDIS 位置 1），不使用 FIFO 阈值级别。如果在每次 DMA 请求之后，系统需要至/自存储器的立即和单独传输，这种模式非常有用。
当在直接模式（禁止FIFO）下将DMA 配置为以存储器到外设模式传输数据时，DMA 会将一个数据从存储器预加载到内部 FIFO，从而确保一旦外设触发 DMA 请求时则立即传输数据。为了避免 FIFO 饱和，建议使用高优先级配置相应的数据流。
该模式仅限以下方式的传输：

- 源和目标传输宽度相等，并均由 DMA_SxCR 中的 PSIZE[1:0] 位定义（MSIZE[1:0] 位的状态是“无关”）
- 不可能进行突发传输（DMA_SxCR 中的 PBURST[1:0] 和 MBURST[1:0] 位的状态是“无关”）

  当实现存储器到存储器传输时不得使用直接模式。

21.3.13 DMA 传输完成

以下各种事件均可以结束传输过程，并将 DMA_LISR 或 DMA_HISR 状态寄存器中的 TCIFx 位置 1：

- 在DMA 流控制器模式下：
  - 在存储器到外设模式下，DMA_SxNDTR 计数器已达到零
  - 传输结束前禁止了数据流（通过将 DMA_SxCR 寄存器中的 EN 位清零），并在传输是外设到存储器或存储器到存储器的模式时，所有的剩余数据均已从 FIFO 刷新到存储器
- 在外设流控制器模式下：
  - 已从外设生成最后的外部突发请求或单独请求，并当 DMA 在外设到存储器模式下工作时，剩余数据已从 FIFO 传输到存储器
  - 数据流由软件禁止，并当 DMA 在外设到存储器模式下工作时，剩余数据已从 FIFO 传输到存储器

注意： 仅在外设到存储器模式下，传输的完成取决于 FIFO 中要传输到存储器的剩余数据。这种情况不适用于存储器到外设模式。

如果是在非循环模式下配置数据流，传输结束后（即要传输的数据数目达到零），除非软件重新对数据流编程并重新使能数据流（通过将 DMA_SxCR 寄存器中的 EN 位置 1），否则 DMA 即会停止传输（通过硬件将 DMA_SxCR 寄存器中的 EN 位清零）并且不再响应任何 DMA 请求。

21.3.14 DMA 传输暂停

可以随时暂停 DMA 传输以供稍后重新开始；也可以在 DMA 传输结束前明确禁止暂停功能。分为两种情况：

- 数据流禁止传输，以后不从停止点重新开始暂停。这种情况下，只需将 DMA_SxCR 寄存器中的 EN 位清零来禁止数据流，除此之外不需要任何其他操作。禁止数据流可能要花费一些时间（需要首先完成正在进行的传输）。需要将传输完成中断标志（DMA_LISR 或DMA_HISR 寄存器中的 TCIF）置 1 来指示传输结束。现在 DMA_SxCR 中的 EN 位的值是“0”，借此确认数据流已经终止传输。DMA_SxNDTR 寄存器包含数据流停止时剩余数据项的数目，这样软件便可以确定数据流中断前已传输了多少数据项。
- 数据流在 DMA_SxNDTR 寄存器中要传输的剩余数据项数目达到 0 之前暂停传输。目的是以后通过重新使能数据流重新开始传输。为了在传输停止点重新开始传输，软件必须在通过写入 DMA_SxCR 寄存器中的 EN 位（然后检查确认该位为‘0’）禁止数据流之后，首先读取 DMA_SxNDTR 寄存器来了解已经收集的数据项的数目。然后：
  - 必须更新外设和/或存储器地址以调整地址指针
  - 必须使用要传输的剩余数据项的数目（禁止数据流时读取的值）更新 SxNDTR 寄存器
  - 然后可以重新使能数据流，从停止点重新开始传输

注意： *请注意，传输完成中断标志（DMA_LISR 或DMA_HISR 中的TCIF）置1 将指示因数据流中断而结束传输。*

21.3.15 流控制器

控制要传输的数据数目的实体称为流控制器。此流控制器使用 DMA_SxCR 寄存器中的 PFCTRL 位针对每个数据流独立配置。

流控制器可以是：

- DMA 控制器：在这种情况下，要传输的数据项的数目在使能 DMA 数据流之前由软件编程到 DMA_SxNDTR 寄存器。
- 外设源或目标：当要传输的数据项的数目未知时属于这种情况。当所传输的是最后的数据时，外设通过硬件向 DMA 控制器发出指示。仅限能够发出传输结束信号的外设支持此功能，也就是：

当外设流控制器用于给定数据流时，写入 DMA_SxNDTR 的值对 DMA 传输没有作用。实际上，不论写入什么值，一旦使能数据流，硬件即会将该值强制置为 0xFFFF 来执行下列方案：

- 预期的数据流中断：DMA_SxCR 寄存器中的 EN 位由软件重置为 0，以在外设发送最后的数据硬件信号（单独或突发）之前停止数据流。这样，在外设到存储器 DMA 传输的情况下，数据流即会关闭并触发 FIFO 刷新。状态寄存器中相应数据流的 TCIFx 标志置 1 以指示 DMA 完成传输。要了解 DMA 传输期间传输的数据项的数目，请读取 DMA_SxNDTR 寄存器并应用下列公式：
  - 传输的数据数目 = 0xFFFF — DMA_SxNDTR

- 因接收到最后的数据硬件信号而引起的正常数据流中断：当外设请求最后的传输（单独或突发）并当此传输完成时，自动中断数据流。相应流的 TCIFx 标志在状态寄存器中置1 以指示 DMA 传输完成。要了解传输的数据项的数目，请读取 DMA_SxNDTR 寄存器并应用与上面相同的公式。
- DMA_SxNDTR 寄存器达到 0：状态寄存器中相应数据流的 TCIFx 标志置 1 以指示强制的DMA 传输完成。即使尚未置位最后的数据硬件信号（单独或突发），也会自动关闭数据流。已传输的数据不会丢失。这意味着即使在外设流控制模式下，DMA 在单独的事务中最多处理 4096 个数据项。

注意：当在存储器到存储器模式下配置时，DMA 始终是流控制器，而 PFCTRL 位由硬件强制置为 0。在外设流控制器模式下禁止循环模式。

21.3.16 可能的 DMA 配置汇总

表 42 汇总了各种可能的 DMA 配置。

表 42. 可能的 DMA 配置

DMA 传输模式	源	目标	流控制器	循环模式	传输类型	直接模式	双缓冲区模式
外设到存储器	AHB 外设端口	AHB 存储器端口	DMA	允许	单独	允许	允许
					突发	禁止	
			外设	禁止	单独	允许	禁止
					突发	禁止	
存储器到外设	AHB 存储器端口	AHB 外设端口	DMA	允许	单独	允许	允许
					突发	禁止	
			外设	禁止	单独	允许	禁止
					突发	禁止	
存储器到存储器	AHB 外设端口	AHB 存储器端口	仅 DMA	禁止	单独	禁止	禁止
					突发		
外设到外设	AHB 外设端口	AHB 外设端口	外设	允许	单独	允许	禁止

21.3.17 流配置过程

配置 DMA 数据流 x（其中 x 是数据流编号）时应遵守下面的顺序：

1 如果使能了数据流，通过重置 DMA_SxCR 寄存器中的 EN 位将其禁止，然后读取此位以确认没有正在进行的数据流操作。将此位写为 0 不会立即生效，因为实际上只有所有当前传输都已完成时才会将其写为 0。当所读取 EN 位的值为 0 时，才表示可以配置数据流。因此在开始任何数据流配置之前，需要等待 EN 位置 0。应将先前的数据块 DMA 传输中在状态寄存器 (DMA_LISR 和DMA_HISR) 中置 1 的所有数据流专用的位置 0，然后才可重新使能数据流。

2 在DMA_SxPAR 寄存器中设置外设端口寄存器地址。外设事件发生后，数据会从此地址移动到外设端口或从外设端口移动到此地址。

3 在DMA_SxMA0R 寄存器（在双缓冲区模式的情况下还有 DMA_SxMA1R 寄存器）中设置存储器地址。外设事件发生后，将从此存储器读取数据或将数据写入此存储器。

4 在DMA_SxNDTR 寄存器中配置要传输的数据项的总数。每出现一次外设事件或每出现一个节拍的突发传输，该值都会递减。

5 使用 DMA_SxCR 寄存器中的 CHSEL[2:0] 选择 DMA 通道（请求）。

6 如果外设用作流控制器而且支持此功能，请将 DMA_SxCR 寄存器中的 PFCTRL 位置 1。

7. 使用 DMA_SxCR 寄存器中的 PL[1:0] 位配置数据流优先级。

8 配置 FIFO 的使用情况（使能或禁止，发送和接收阈值）。

9 配置数据传输方向、外设和存储器增量 / 固定模式、单独或突发事务、外设和存储器数据宽度、循环模式、双缓冲区模式和传输完成一半和/ 或全部完成，和/ 或 DMA_SxCR 寄存器中错误的中断。

10 通过将 DMA_SxCR 寄存器中的 EN 位置 1 激活数据流。

一旦使能了流，即可响应连接到数据流的外设发出的任何 DMA 请求。

一旦在 AHB 目标端口上传输了一半数据，传输一半标志 (HTIF) 便会置 1，如果传输一半中断使能位 (HTIE) 置 1，还会生成中断。传输结束时，传输完成标志 (TCIF) 便会置 1，如果传输完成中断使能位 (TCIE) 置1，还会生成中断。

警告： 要关闭连接到 DMA 数据流请求的外设，必须首先关闭外设连接的 DMA 数据流，然后等待 EN 位 = 0。只有这样才能安全地禁止外设。

21.3.18错误管理

DMA 控制器可以检测到以下错误：

- **传输错误**：当发生下列情况时，传输错误中断标志 (TEIFx) 将置 1：
  - DMA 读或写访问期间发生总线错误
  - 软件请求在双缓冲区模式下写访问存储器地址寄存器，但是，已使能数据流，并且当前目标存储器是受写入存储器地址寄存器操作影响的存储器
- **FIFO 错误**：如果发生下列情况，FIFO 错误中断标志 (FEIFx) 将置 1：
  - 检测到 FIFO 下溢情况
  - 检测到 FIFO 上溢情况（在存储器到存储器模式下由 DMA 内部管理请求和传输，所以在此模式下不检测溢出情况）
  - 当FIFO 阈值级别与存储器突发大小不兼容时使能流
- **直接模式错误**：只有当在直接模式下工作并且已将 DMA_SxCR 寄存器中的 MINC 位清零时，才能在外设到存储器模式下将直接模式错误中断标志 (DMEIFx) 置 1。当在先前数据未完全传输到存储器（因为存储器总线未得到授权）的情况下发生 DMA 请求时，该标志将置 1。在这种情况下，该标志指示有两个数据项相继传输到相同的目标地址，如果目标不能管理这种情况，会发生问题。

在直接模式下，如果出现下列条件，FIFO 错误标志也会置 1：

- 在外设到存储器模式下，如果未对存储器总线授权支持多个外设请求，FIFO 可能饱和（上溢）
- 在存储器到外设模式下，如果在外设请求发生前存储器总线未得到授权，则可能发生下溢的情况

如果由于突发大小与 FIFO 阈值级别之间的不兼容而引起 TEIFx 或 FEIFx 标志置 1，则硬件自动将相应数据流配置寄存器 (DMA_SxCR) 中的 EN 位清零，从而禁止错误的数据流。
如果由于上溢或下溢情况引起 DMEIFx 或 FEIFx 标志置 1，则不会自动禁止错误的数据流， 而是由软件决定是否通过重置 DMA_SxCR 寄存器中的 EN 位禁止数据流。这是因为当发生这种错误时没有数据丢失。

当DMA_LISR 或DMA_HISR 寄存器中数据流的错误中断标志（TEIF、FEIF、DMEIF）置1 时，如果 DMA_SxCR 或DMA_SxFCR 寄存器中相应的中断使能位（TEIE、FEIE、DMIE） 也置 1，则会生成一个中断。
注意： 当 FIFO 上溢或下溢的情况发生时，因为直到上溢或下溢的情况被清除，数据流才会确认外设请求，所以数据不会丢失。如果此确认过程花费过多时间，则外设本身会检测到其内部缓冲器上溢或下溢的情况，数据可能丢失。

21.4DMA 中断

对于每个 DMA 数据流，可在发生以下事件时产生中断：

- 达到半传输
- 传输完成
- 传输错误
- FIFO 错误（上溢、下溢或 FIFO 级别错误）
- 直接模式错误

可以使用单独的中断使能位以实现灵活性，如*表 43* 所示。

表43. DMA 中断请求

中断事件	事件标志	使能控制位
半传输	HTIF	HTIE
传输完成	TCIF	TCIE
传输错误	TEIF	TEIE
FIFO 上溢/ 下溢	FEIF	FEIE
直接模式错误	DMEIF	DMEIE

注意： 在将使能控制位置‘1’前，应将相应的事件标志清零，否则会立即产生中断。

21.5DMA 寄存器

DMA 寄存器可按字（32 位）进行访问。

21.5.1DMA 低中断状态寄存器 (DMA_LISR)

DMA low interrupt status register

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				TCIF3	HTIF3	TEIF3	DMEIF3	Reserved	FEIF3	TCIF2	HTIF2	TEIF2	DMEIF2	Reserved	FEIF2
				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
Reserved				TCIF1	HTIF1	TEIF1	DMEIF1	Reserved	FEIF1	TCIF0	HTIF0	TEIF0	DMEIF0	Reserved	FEIF0
				r	r	r	r		r	r	r	r	r		r

位 31:28、15:12 保留，必须保持复位值。

位 27、21、11、5 **TCIFx**：数据流 x 传输完成中断标志 (Stream x transfer complete interrupt flag) (x = 3..0) 此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_LIFCR 寄存器的相应位。

0：数据流 x 上无传输完成事件

1：数据流 x 上发生传输完成事件

位 26、20、10、4 **HTIFx**：数据流 x 半传输中断标志 (Stream x half transfer interrupt flag) (x=3..0)

此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_LIFCR 寄存器的相应位。

0：数据流 x 上无半传输事件

1：数据流 x 上发生半传输事件

位 25、19、9、3 **TEIFx**：数据流 x 传输错误中断标志 (Stream x transfer error interrupt flag) (x=3..0)

此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_LIFCR 寄存器的相应位。

0：数据流 x 上无传输错误

1：数据流 x 上发生传输错误

位24、18、8、2 **DMEIFx**：数据流 x 直接模式错误中断标志 (Stream x direct mode error interrupt flag) (x=3..0) 此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_LIFCR 寄存器的相应位。

0：数据流 x 上无直接模式错误

1：数据流 x 上发生直接模式错误

位 23、17、7、1 保留，必须保持复位值。

位 22、16、6、0 **FEIFx**：数据流 x FIFO 错误中断标志 (Stream x FIFO error interrupt flag) (x=3..0)

此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_LIFCR 寄存器的相应位。

0：数据流 x 上无 FIFO 错误事件

1：数据流 x 上发生 FIFO 错误事件

21.5.2 DMA 高中断状态寄存器 (DMA_HISR)

DMA high interrupt status register

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				TCIF7	HTIF7	TEIF7	DMEIF7	Reserved	FEIF7	TCIF6	HTIF6	TEIF6	DMEIF6	Reserved	FEIF6
				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
Reserved				TCIF5	HTIF5	TEIF5	DMEIF5	Reserved	FEIF5	TCIF4	HTIF4	TEIF4	DMEIF4	Reserved	FEIF4
				r	r	r	r		r	r	r	r	r		r

位 31:28、15:12 保留，必须保持复位值。

位 27、21、11、5 TCIFx：数据流 x 传输完成中断标志 (Stream x transfer complete interrupt flag) (x=7..4)

此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_HIFCR 寄存器的相应位。

0：数据流 x 上无传输完成事件

1：数据流 x 上发生传输完成事件

位 26、20、10、4 **HTIFx**：数据流 x 半传输中断标志 (Stream x half transfer interrupt flag) (x=7..4)

此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_HIFCR 寄存器的相应位。

0：数据流 x 上无半传输事件

1：数据流 x 上发生半传输事件

位 25、19、9、3 **TEIFx**：数据流 x 传输错误中断标志 (Stream x transfer error interrupt flag) (x=7..4)

此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_HIFCR 寄存器的相应位。

0：数据流 x 上无传输错误

1：数据流 x 上发生传输错误

位 24、18、8、2 **DMEIFx**：数据流 x 直接模式错误中断标志 (Stream x direct mode error interrupt flag) (x=7..4) 此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_HIFCR 寄存器的相应位。

0：数据流 x 上无直接模式错误

1：数据流 x 上发生直接模式错误

位 23、17、7、1 保留，必须保持复位值。

位 22、16、6、0 **FEIFx**：数据流 x FIFO 错误中断标志 (Stream x FIFO error interrupt flag) (x=7..4) 此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_HIFCR 寄存器的相应位。

0：数据流 x 上无 FIFO 错误事件

1：数据流 x 上发生 FIFO 错误事件

21.5.3 DMA 低中断标志清零寄存器 (DMA_LIFCR)

DMA low interrupt flag clear register

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				CTCIF3	CHTIF3	CTEIF3	CDMEIF3	Reserved	CFEIF3	CTCIF2	CHTIF2	CTEIF2	CDMEIF2	Reserved	CFEIF0
				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
Reserved				CTCIF1	CHTIF1	CTEIF1	CDMEIF1	Reserved	CFEIF1	CTCIF0	CHTIF0	CTEIF0	CDMEIF0	Reserved	CFEIF0
				w	w	w	w		w	w	w	w	w		w

位 31:28、15:12 保留，必须保持复位值。

位 27、21、11、5 **CTCIFx**：数据流 x 传输完成中断标志清零 (Stream x clear transfer complete interrupt flag) (x = 3..0)

将 1 写入此位时，DMA_LISR 寄存器中相应的 TCIFx 标志将清零

位 26、20、10、4 **CHTIFx**：数据流 x 半传输中断标志清零 (Stream x clear half transfer interrupt flag) (x = 3..0)

将 1 写入此位时，DMA_LISR 寄存器中相应的 HTIFx 标志将清零

位25、19、9、3 **CTEIFx**：数据流 x 传输错误中断标志清零 (Stream x clear transfer error interrupt flag) (x = 3..0)

将 1 写入此位时，DMA_LISR 寄存器中相应的 TEIFx 标志将清零

位 24、18、8、2 **CDMEIFx**：数据流 x 直接模式错误中断标志清零 (Stream x clear direct mode error interrupt flag) (x = 3..0)

将 1 写入此位时，DMA_LISR 寄存器中相应 DMEIFx 标志将清零

位 23、17、7、1 保留，必须保持复位值。

位 22、16、6、0 **CFEIFx**：数据流 x FIFO 错误中断标志清零 (Stream x clear FIFO error interrupt flag) (x = 3..0)

将 1 写入此位时，DMA_LISR 寄存器中相应的 CFEIFx 标志将清零

21.5.4DMA 高中断标志清零寄存器 (DMA_HIFCR)

DMA high interrupt flag clear register

偏移地址：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				CTC IF7	CHT IF7	CTE IF7	CDME IF7	Reserved	CFE IF7	CTC IF6	CHT IF6	CTE IF6	CDME IF6		CFE IF6
				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
Reserved				CTCIF5	CHTIF5	CTEIF5	CDMEIF5	Reserved	CFE IF5	CTC IF4	CHT IF4	CTE IF4	CDME IF4		CFE IF4
				w	w	w	w		w	w	w	w	w		w

位 31:28、15:12 保留，必须保持复位值。

位 27、21、11、5 **CTCIFx**：数据流 x 传输完成中断标志清零 (Stream x clear transfer complete interrupt flag) (x = 7.0.4)

将 1 写入此位时，DMA_HISR 寄存器中相应的 TCIFx 标志将清零

位 26、20、10、4 **CHTIFx**：数据流 x 半传输中断标志清零 (Stream x clear half transfer interrupt flag) (x = 7.0.4)

将 1 写入此位时，DMA_HISR 寄存器中相应的 HTIFx 标志将清零

位25、19、9、3 **CTEIFx**：数据流 x 传输错误中断标志清零 (Stream x clear transfer error interrupt flag) (x = 7.0.4)

将 1 写入此位时，DMA_HISR 寄存器中相应的 TEIFx 标志将清零

位 24、18、8、2 **CDMEIFx**：数据流 x 直接模式错误中断标志清零 (Stream x clear direct mode error interrupt flag) (x = 7.0.4)

将 1 写入此位时，DMA_HISR 寄存器中相应 DMEIFx 标志将清零

位23、17、7、1 保留，必须保持复位值。

位22、16、6、0 **CFEIFx**：数据流 x FIFO 错误中断标志清零 (Stream x clear FIFO error interrupt flag) (x = 7.0.4)

将 1 写入此位时，DMA_HISR 寄存器中相应的 CFEIFx 标志将清零

STC MCU

21.5.5 DMA 数据流 x 配置寄存器 (DMA_SxCR) (x = 0..7)

DMA stream x configuration register

此寄存器用于配置相关数据流。

偏移地址：0x10 + 0x18 × *数据流编号*

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				CHSEL[2:0]			MBURST[1:0]		PBURST[1:0]		Reser	CT	DBM	PL[1:0]	
				rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:25 **CHSEL[2:0]**：通道选择 (Channel selection)

这些位将由软件置 1 和清零。

000：选择通道 0
001：选择通道 1
010：选择通道 2
011：选择通道 3
100：选择通道 4
101：选择通道 5
110：选择通道 6
111：选择通道 7

这些位受到保护，只有 EN 为“0”时才可以写入

位 24:23 **MBURST[1:0]**：存储器突发传输配置 (Memory burst transfer configuration)

这些位将由软件置 1 和清零。

00：单次传输
01：INCR4（4 个节拍的增量突发传输）
10：INCR8（8 个节拍的增量突发传输）
11：INCR16（16 个节拍的增量突发传输）

这些位受到保护，只有 EN 为“0”时才可以写入

在直接模式中，当位 EN = “1”时，这些位由硬件强制置为 0x0。

位 22:21 **PBURST[1:0]**：外设突发传输配置 (Peripheral burst transfer configuration)

这些位将由软件置 1 和清零。

00：单次传输
01：INCR4（4 个节拍的增量突发传输）
10：INCR8（8 个节拍的增量突发传输）
11：INCR16（16 个节拍的增量突发传输）

这些位受到保护，只有 EN 为“0”

时才可以写入在直接模式下，这些位

由硬件强制置为 0x0。

位 20 保留，必须保持复位值。

位 19 **CT**：当前目标（仅在双缓冲区模式下）(Current target (only in double buffer mode))

此位由硬件置 1 和清零，也可由软件写入。

STC MCU

0：当前目标存储器为存储器 0（使用 DMA_SxM0AR 指针寻址）
1：当前目标存储器为存储器 1（使用 DMA_SxM1AR 指针寻址）
只有 EN 为“0”时，此位才可以写入，以指示第一次传输的目标存储区。
在使能数据流后，此位相当于一个状态标志，用于指示作为当前目标的存储区。

位 18 DBM：双缓冲区模式 (Double buffer mode)

此位由软件置 1 和清零。
0：传输结束时不切换缓冲区
1：DMA 传输结束时切换目标存储区
此位受到保护，只有 EN 为“0”时才可以写入。

位17:16 **PL[1:0]**：优先级 (Priority level)

这些位将由软件置 1 和清零。

00：低
01：中
10：高
11：非常高
这些位受到保护，只有 EN 为“0”时才可以写入。

位 15 **PINCOS**：外设增量偏移量 (Peripheral increment offset size)

此位由软件置 1 和清零
0：用于计算外设地址的偏移量与 PSIZE 相关
1：用于计算外设地址的偏移量固定为 4（32 位对齐）。如果位 PINC =“0”，则此位没有意义。
此位受到保护，只有 EN 为“0”时才可以写入。
如果选择直接模式或者 PBURST 不等于“00”，则当使能数据流（位 EN =“1”）时，此位由硬件强制置为低电平。

位 14:13 **MSIZE[1:0]**：存储器数据大小 (Memory data size)

这些位将由软件置 1 和清零。
00：字节（8 位）
01：半字（16 位）
10：字（32 位）
11：保留
这些位受到保护，只有 EN 为“0”时才可以写入。
在直接模式下，当位 EN =“1”时，MSIZE 位由硬件强制置为与 PSIZE 相同的值。

位 12:11 **PSIZE[1:0]**：外设数据大小 (Peripheral data size)

这些位将由软件置 1 和清零。
00：字节（8 位）
01：半字（16 位）
10：字（32 位）
11：保留
这些位受到保护，只有 EN 为“0”时才可以写入

位 10 **MINC**：存储器递增模式 (Memory increment mode)

此位由软件置 1 和清零。

0：存储器地址指针固定
1：每次数据传输后，存储器地址指针递增（增量为 MSIZE 值）

此位受到保护，只有 EN 为“0”时才可以写入。

位9 **PINC**：外设递增模式 (Peripheral increment mode)

此位由软件置 1 和清零。

0：外设地址指针固定

1：每次数据传输后，外设地址指针递增（增量为 PSIZE 值） 此位受到保护，只有 EN 为“0”时才可以写入。

位 8 **CIRC**：循环模式 (Circular mode)

此位由软件置 1 和清零，并可由硬件清零。

0：禁止循环模式

1：使能循环模式

如果外设为流控制器（位 PFCTRL=1）且使能数据流（位 EN=1），此位由硬件自动强制清零。

如果 DBM 位置 1，当使能数据流（位 EN =“1”）时，此位由硬件自动强制置 1。

位7:6 **DIR[1:0]**：数据传输方向 (Data transfer direction)

这些位将由软件置 1 和清零。

00：外设到存储器

01：存储器到外设

10：存储器到存储器

11：保留

这些位受到保护，只有 EN 为“0”时才可以写入。

位 5 **PFCTRL**：外设流控制器 (Peripheral flow controller)

此位由软件置 1 和清零。

0：DMA 是流控制器

1：外设是流控制器

此位受到保护，只有 EN 为“0”时才可以写入。

选择存储器到存储器模式（位 DIR[1:0]=10）后，此位由硬件自动强制清零。

位 4 **TCIE**：传输完成中断使能 (Transfer complete interrupt enable)

此位由软件置 1 和清零。

0：禁止 TC 中断

1：使能 TC 中断

位 3 **HTIE**：半传输中断使能 (Half transfer interrupt enable)

此位由软件置 1 和清零。

0：禁止 HT 中断

1：使能 HT 中断

位 2 **TEIE**：传输错误中断使能 (Transfer error interrupt enable)

此位由软件置 1 和清零。

0：禁止 TE 中断

1：使能 TE 中断

位 1 **DMEIE**：直接模式错误中断使能 (Direct mode error interrupt enable)

此位由软件置 1 和清零。

0：禁止 DME 中断

1：使能 DME 中断

位 0 **EN**：数据流使能/ 读作低电平时数据流就绪标志 (Stream enable / flag stream ready when read low)

此位由软件置 1 和清零。

0：禁止数据流

1：使能数据流

以下情况下，此位可由硬件清零：

– DMA 传输结束时（准备好配置数据流）

– AHB 主总线出现传输错误时

– 存储器 AHB 端口上的 FIFO 阈值与突发大小不兼容时

此位读作 0 时，软件可以对配置和 FIFO 位寄存器编程。EN 位读作 1 时，禁止向这些寄存器执行写操作。

注意： 将 EN 位置“1”以启动新传输之前，DMA_LISR 或 DMA_HISR 寄存器中与数据流相对应的事件标志必须清零。

21.5.6DMA 数据流 x 数据项数寄存器 (DMA_SxNDTR) (x = 0..7)

DMA stream x number of data register

偏移地址：0x14 + 0x18 × *数据流编号*

复位值：0x0000 0000

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

位 31:16 保留，必须保持复位值。

位 15:0 **NDT[15:0]**：要传输的数据项数目 (Number of data items to transfer)

要传输的数据项数目（0 到 4095）。只有在禁止数据流时，才能向此寄存器执行写操作。使能数据流后，此寄存器为只读，用于指示要传输的剩余数据项数。每次 DMA 传输后，此寄存器将递减。

传输完成后，此寄存器保持为零（数据流处于正常模式时），或者在以下情况下自动以先前编程的值重载：

– 以循环模式配置数据流时。

– 通过将 EN 位置“1”来重新使能数据流时

如果该寄存器的值为零，则即使使能数据流，也无法完成任何事务。

21.5.7DMA 数据流 x 外设地址寄存器 (DMA_SxPAR) (x = 0..7)

DMA stream x peripheral address register

偏移地址：0x18 + 0x18 × *数据流编号*

复位值：0x0000 0000

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

位 31:0 **PAR[31:0]**：外设地址 (Peripheral address)

读/ 写数据的外设数据寄存器的基址。

这些位受到写保护，只有 DMA_SxCR 寄存器中的 EN 为“0”时才可以写入。

21.5.8DMA 数据流 x 存储器 0 地址寄存器 (DMA_SxM0AR) (x = 0..7)

DMA stream x memory 0 address register

偏移地址：0x1C + 0x18 × *数据流编号*

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
M0A[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
M0A[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **M0A[31:0]**：存储器 0 地址 (Memory 0 address)

读/ 写数据的存储区 0 的基址。

这些位受到写保护，只有在以下情况下才可以写入：

– 禁止数据流（DMA_SxCR 寄存器中的位 EN=“0”）或

– 使能数据流（DMA_SxCR 寄存器中的 EN=“1”）并且 DMA_SxCR 寄存器中的位CT =“1”（在双缓冲区模式下）。

21.5.9DMA 数据流 x 存储器 1 地址寄存器 (DMA_SxM1AR) (x = 0..7)

DMA stream x memory 1 address register

偏移地址：0x20 + 0x18 × *数据流编号*

复位值：0x0000 0000

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

M1A[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
M1A[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **M1A[31:0]:** 存储器 1 地址(用于双缓冲区模式)(Memory 1 address (used in case of Double buffer mode))

读/ 写数据的存储区 1 的基址。

此寄存器仅用于双缓冲区模式。

这些位受到写保护，只有在以下情况下才可以写入：

– 禁止数据流（DMA_SxCR 寄存器中的位 EN=“0”）或

– 使能数据流（DMA_SxCR 寄存器中的 EN=“1”）并且 DMA_SxCR 寄存器中的位 CT =“0”。

21.5.10 DMA 数据流 x FIFO 控制寄存器 (DMA_SxFCR) (x = 0..7)

DMA stream x FIFO control register

偏移地址：0x24 + 0x24 × *数据流编号*

复位值：0x0000 0021

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

位 31:8 保留，必须保持复位值。

位 7 **FEIE**：FIFO 错误中断使能 (FIFO error interrupt enable)

此位由软件置 1 和清零。

0：禁止 FE 中断

1：使能 FE 中断

位 6 保留，必须保持复位值。

位 5:3 **FS[2:0]**：FIFO 状态 (FIFO status)

这些位为只读。

000：0 < fifo_level < 1/4

001：1/4 < fifo_level < 1/2

010：1/2 < fifo_level < 3/4

011：3/4 < fifo_level < 满

100：FIFO 为空

101：FIFO 已满

其它：无意义

在直接模式（DMDIS 位为零）下，这些位无意义。

位2 **DMDIS**：直接模式禁止 (Direct mode disable) 此位由软件置 1 和清零。它可由硬件置 1。

0：使能直接模式
1：禁止直接模式
此位受到保护，只有 EN 为“0”时才可以写入。

如果选择存储器到存储器模式（DMA_SxCR 中的 DIR 位为“10”），并且 DMA_SxCR 寄存器中的 EN 位为“1”，则此位由硬件置 1，因为在存储器到存储器配置不能使用直接模式。

位 1:0 **FTH[1:0]**：FIFO 阈值选择 (FIFO threshold selection)

这些位将由软件置 1 和清零。

00：FIFO 容量的 1/4
01：FIFO 容量的 1/2
10：FIFO 容量的 3/4
11：FIFO 完整容量

在直接模式（DMDIS 值为零）下，不使用这些位。

这些位受到保护，只有 EN 为“1”时才可以写入。

21.5.11 DMA 寄存器映射

表 44 汇总了 DMA 寄存器。

表44. DMA 寄存器映射和复位值

偏移	寄存器	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
0x000	DMA_LISR	Reserved				TCIF3	HTIF3	TEIF3	DMEIF3		FEIF3	TCIF2	HTIF2	TEIF2	DMEIF2		FEIF2	Reserved				TCIF1	HTIF1	TEIF1	DMEIF1		FEIF1	TCIF0	HTIF0	TEIF0	DMEIF0		FEIF0
	Reset value					0	0	0	0		0	0	0	0	0		0					0	0	0	0		0	0	0	0	0		0
0x004	DMA_HISR	Reserved				TCIF7	HTIF7	TEIF7	DMEIF7		FEIF7	TCIF6	HTIF6	TEIF6	DMEIF6		FEIF6	Reserved				TCIF5	HTIF5	TEIF5	DMEIF5		FEIF5	TCIF4	HTIF4	TEIF4	DMEIF4		FEIF4
	Reset value	Reserved				0	0	0	0		0	0	0	0	0		0	Reserved				0	0	0	0		0	0	0	0	0		0
0x008	DMA_LIFCR	Reserved				CTCIF3	CHTIF3	CTEIF3	CDMEIF3		CFEIF3	CTCIF2	CHTIF2	CTEIF2	CDMEIF2		CFEIF2	Reserved				CTCIF1	CHTIF1	CTEIF1	CDMEIF1		CFEIF1	CTCIF0	CHTIF0	CTEIF0	CDMEIF0		CFEIF0
	Reset value					0	0	0	0		0	0	0	0	0		0					0	0	0	0		0	0	0	0	0		0
0x00C	DMA_HIFCR	Reserved				CTCIF7	CHTIF7	CTEIF7	CDMEIF7		CFEIF7	CTCIF6	CHTIF6	CTEIF6	CDMEIF6		CFEIF6	Reserved				CTCIF5	CHTIF5	CTEIF5	CDMEIF5		CFEIF5	CTCIF4	CHTIF4	CTEIF4	CDMEIF4		CFEIF4

STC MCU

	Reset value					0	0	0	0		0	0	0	0	0		0					0	0	0	0		0	0	0	0	0		0
0x0010	DMA_S0CR	Reserved				CHSEL[2:0]			MBURST[1:0]		PBURST[1:0]			CT	DBM	PL[1:0]		PINCOS	MSIZ[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
	Reset valu					0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0014	DMA_S0NDTR	Reserved																NDT[15:0]															
	Reset valu																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0018	DMA_S0PAR	PA[31:0]																															
	Reset valu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x001C	DMA_S0MOAR	MOA[31:0]																															
	Reset valu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0020	DMA_S0M1AR	M1A[31:0]																															
	Reset valu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0024	DMA_S0FCR	Reserved																								FEIE	Reserved	FS[2:0]			DMDIS	FTH[1:0]	
	Reset valu																																
0x0028	DMA_S1CR	Reserved				CHSEL[2:0]			MBURST[1:0]		PBURST[1:0]		Reserved	CT	DBM	PL[1:0]		PINCOS	MSIZ[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
	Reset valu					0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x002C	DMA_S1NDTR	Reserved																NDT[15:0]															
	Reset valu																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0030	DMA_S1PAR	PA[31:0]																															
	Reset valu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0034	DMA_S1MOAR	MOA[31:0]																															
	Reset valu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0038	DMA_S1M1AR	M1A[31:0]																															
	Reset valu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x003C	DMA_S1FCR	Reserved																								FEIE	Reser	FS[2:0]			DMDIS	FTH[1:0]	

	Reset valu																									0		0	0	0	0	0	0
0x0040	DMA_S2CR	Reserved				CHSEL[2:0]			MBURST[1:0]		PBURST[1:0]			CT	DBM	PL[1:0]		PINCOS	MSIZ[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
	Reset valu					0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0044	DMA_S2NDTR	Reserved																NDT[15:0]															
	Reset valu																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0048	DMA_S2PAR	PA[31:0]																															
	Reset valu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x004C	DMA_S2MOAR	MOA[31:0]																															
	Reset valu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0050	DMA_S2M1AR	M1A[31:0]																															
	Reset valu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0054	DMA_S2FCR	Reserved																								FEIE	Reserved	FS[2:0]			DMDIS	FTH[1:0]	
	Reset valu																									0		0	0	0	0	0	0
0x0058	DMA_S3CR	Reserved				CHSEL[2:0]			MBURST[1:0]		PBURST[1:0]			CT	DBM	PL[1:0]		PINCOS	MSIZ[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
	Reset valu					0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x005C	DMA_S3NDTR	Reserved																NDT[15:0]															
	Reset valu																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0060	DMA_S3PAR	PA[31:0]																															
	Reset valu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0064	DMA_S3MOAR	MOA[31:0]																															
	Reset valu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0068	DMA_S3M1AR	M1A[31:0]																															
	Reset valu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x006C	DMA_S3FCR	Reserved																								FEIE	Reser	FS[2:0]			DMDIS	FTH[1:0]	

	Reset valu																									0		0	0	0	0	0	0
0x0070	DMA_S4CR	Reserved				CHSEL[2:0]			MBURST[1:0]		PBURST[1:0]		Reserved	CT	DBM	PL[1:0]		PINCOS	MSIZ[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
	Reset valu					0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0074	DMA_S4NDTR	Reserved																NDT[15:0]															
	Reset valu																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0078	DMA_S4PAR	PA[31:0]																															
	Reset valu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x007C	DMA_S4M0AR	M0A[31:0]																															
	Reset valu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0080	DMA_S4M1AR	M1A[31:0]																															
	Reset valu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0084	DMA_S4FCR	Reserved																								FEIE	Reserved	FS[2:0]			DMDIS	FTH[1:0]	
	Reset valu																									0		0	0	0	0	0	0
0x0088	DMA_S5CR	Reserved				CHSEL[2:0]			MBURST[1:0]		PBURST[1:0]			CT	DBM	PL[1:0]		PINCOS	MSIZ[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
	Reset valu					0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x008C	DMA_S5NDTR	Reserved																NDT[15:0]															
	Reset valu																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0090	DMA_S5PAR	PA[31:0]																															
	Reset valu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0094	DMA_S5M0AR	M0A[31:0]																															
	Reset valu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0098	DMA_S5M1AR	M1A[31:0]																															
	Reset valu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x009C	DMA_S5FCR	Reserved																								FEIE	Reser	FS[2:0]			DMDIS	FTH[1:0]	

	Reset valu																									0		0	0	0	0	0	0
0x00A0	DMA_S6CR	Reserved				CHSEL[2:0]			MBURST[1:0]		PBURST[1:0]			CT	DBM	PL[1:0]		PINCOS	MSIZ[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
	Reset valu					0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00A4	DMA_S6NDTR	Reserved																NDT[15:0]															
	Reset valu																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00A8	DMA_S6PAR	PA[31:0]																															
	Reset valu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00AC	DMA_S6M0AR	M0A[31:0]																															
	Reset valu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00B0	DMA_S6M1AR	M1A[31:0]																															
	Reset valu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00B4	DMA_S6FCR	Reserved																								FEIE	Reserved	FS[2:0]			DMDIS	FTH[1:0]	
	Reset valu																									0		0	0	0	0	0	0
0x00B8	DMA_S7CR	Reserved				CHSEL[2:0]			MBURST[1:0]		PBURST[1:0]			CT	DBM	PL[1:0]		PINCOS	MSIZ[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
	Reset valu					0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00BC	DMA_S7NDTR	Reserved																NDT[15:0]															
	Reset valu																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C0	DMA_S7PAR	PA[31:0]																															
	Reset valu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C4	DMA_S7M0AR	M0A[31:0]																															
	Reset valu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C8	DMA_S7M1AR	M1A[31:0]																															
	Reset valu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00CC	DMA_S7FCR	Reserved																								FEIE	Reser	FS[2:0]			DMDIS	FTH[1:0]	

	Reset valu		0		0	0	0	0	0	0

STC MCU

22 Cache/TCM/存储器和总线架构

22.1 系统架构

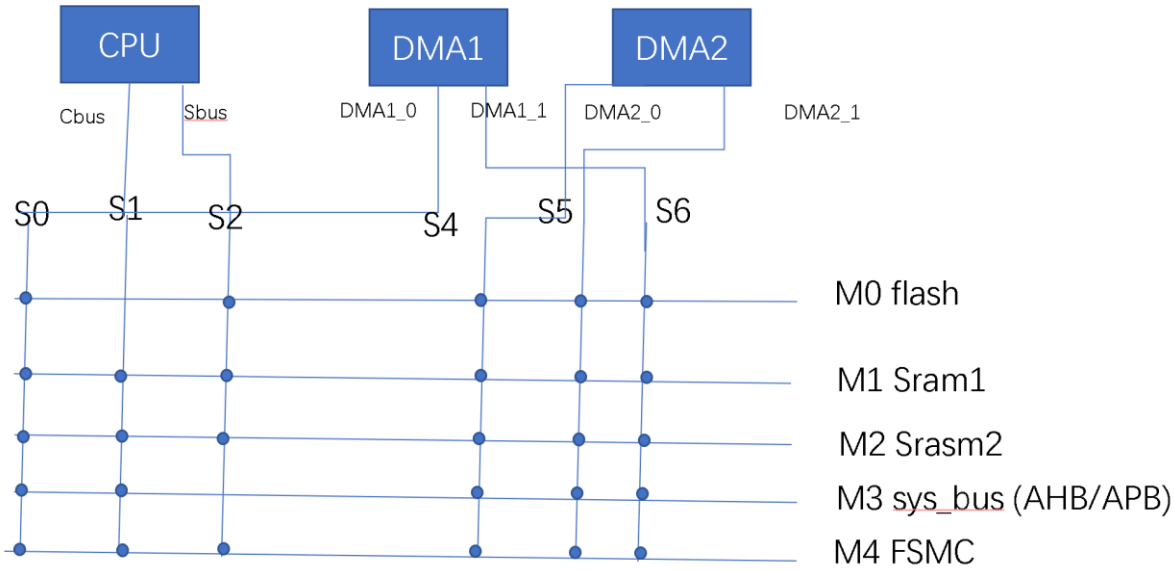
独特的程序指令和数据加速单元，主系统由 32 位多层 AHB 总线矩阵构成，可实现以下部分的互连：

- 六条主控总线：
  - Cortex™-MSTAR 内核 I 总线、D 总线
  - DMA1 总线 1，总线 2
  - DMA2 1 总线，总线 2
- 五条被控总线：
  - 内部 Flash
  - 主要内部 SRAM1 (16KB)
  - 辅助内部 SRAM2 (4KB)
  - 系统外设（包括 AHB-APB 总线桥和 AHB、APB 外设）
  - FSMC

借助总线矩阵，可以实现主控总线到被控总线的访问，这样即使在多个高速外设同时运行期间，系统也可以实现并发访问和高效运行。此架构如图所示。

程序加速

图 1. STC32 器件的系统架构图器件的系统架构



为了弥补 Cache 访问的不确定性，而增加的 OnChip Memory.本 CPU 含有分立的 Instruction TCM / Data TCM.

TCM 包含在存储器的地址映射空间中，可以作为快速存储器来访问。TCM 使用物理地址，对 TCM 的写访问，受到 MMU 内部保护信息的控制.向 TCM 中的内存位置写入时，不会发生任何外部写入。

TCM 用于向处理器提供低延迟内存，它没有高速缓存特有的不可预测性。 可以使用 TCM 来存放重要例程，如中断处理例程或者极需要避免高速缓存不确定性的实时任务。此外，可以使用 TCM 来保存暂时寄存器数据、局部属性不适合高速缓存的数据类型，以及中断堆栈等重要数据结构。

TCM 是一个固定大小的 RAM，紧密地耦合至处理器内核，提供与 cache 相当的性能，相比于 cache 的优点是，程序代码可以精确地控制什么函数或代码放在那儿(RAM 里)。当然 TCM 永远不会被踢出主存储器，因此，他会有一个被用户预设的性能，而不是象 cache 那样是统计特性的性能提高。

TCM 对于以下几种情况的代码是非常有用、也是需要的：可预见的实时处理（中断处理）、时间可预见（加密算法）、避免 cache 分析（加密算法）、或者只是要求高性能的代码（编解码功能）。随着 cache 大小的增加以及总线性能的规模，TCM 将会变得越来越不重要，但是他提供了一个让你权衡的机会那么，哪一个更好呢？他取决于你的应用。Cache 是一个通用目的的加速器，他会加速你的所有代码，而不依赖于存储方式。TCM 只会加速你有意放入 TCM 的代码，其余的其他代码只能通过 cache 加速。Cache 是一个通用目的解决方案，TCM 在某些特殊情况下是非常有用的。假如你不认为需要 TCM 的话，那么你可能就不需要了，转而加大你的 cache，从而加速运行于内核上的所有软件代码

指令加速 RAM（ITCM）和数据加速 RAM（DTCM）的大小是 4K。

其中

指令加速 RAM 没有使能起始地址：0x30000000 使能后地址变为：0

数据加速 RAM 没有使能起始地址：0x30000000+4096 使能后地址变为：0x20000000

22.1.1 S0 ： I 总线

此总线用于将 Cortex™-MSTAR 内核的指令总线连接到总线矩阵。内核通过此总线获取指令。此总线访问的对象是包含代码的存储器（内部 Flash/SRAM 或通过 FSMC 的外部存储器）。

22.1.2 S1 ： D 总线

此总线用于将 Cortex™-MSTAR 数据总线和 数据 RAM 连接到总线矩阵。内核通过此总线进行立即数加载和调试访问。此总线访问的对象是包含代码或数据的存储器（内部 Flash 或通过 FSMC 的外部存储器）。

22.1.3 S2 ： S 总线

此总线用于将 Cortex™-MSTAR 内核的系统总线连接到总线矩阵。此总线用于访问位于外设或 SRAM 中的数据。也可通过此总线获取指令（效率低于 ICode）。此总线访问的对象是 64 KB、64 KB 的内部 SRAM、包括 APB 外设在内的 AHB1 外设、AHB2 外设以及通过 FSMC 的外部存储器。

22.1.4 S3 S4 ： DMA 存储器总线

此总线用于将 DMA 存储器总线主接口连接到总线矩阵。DMA 通过此总线来执行存储器数据的传入和传出。此总线访问的对象是数据存储器：内部 SRAM 以及通过 FSMC 的外部存储器。

22.1.5S5 ： DMA 外设总线

此总线用于将 DMA 外设主总线接口连接到总线矩阵。DMA 通过此总线访问 AHB 外设或执行存储器间的数据传输。此总线访问的对象是 AHB 和 APB 外设以及数据存储器：内部 SRAM 以及通过 FSMC 的外部存储器。

22.1.6总线矩阵

总线矩阵用于主控总线之间的访问仲裁管理。仲裁采用循环调度算法。

22.1.7AHB/APB 总线桥 (APB)

借助两个 AHB/APB 总线桥 APB1 和 APB2，可在 AHB 总线与两个 APB 总线之间实现完全同步的连接，从而灵活选择外设频率。有关 APB1 和 APB2 最大频率的详细信息，请参见器件数据手册；有关 AHB 和 APB 外设地址映射的信息，请参见表 2。 每次芯片复位后，所有外设时钟都被关闭（SRAM 和 Flash 接口除外）。使用外设前，必须在 RCC_AHBxENR 或 RCC_APBxENR 寄存器 中使能其时钟。

注意： 对 APB 寄存器执行 16 位或 8 位访问时，该访问将转换为 32 位访问：总线桥将 16 位或 8 位数据复制后提供给 32 位向量。

22.2存储器组织结构

程序存储器、数据存储器、寄存器和 I/O 端口排列在同一个顺序的 4 GB 地址空间内。各字节按小端格式在存储器中编码。字中编号最低的字节被视为该字的最低有效字节，而编号最高的字节被视为最高有效字节。有关外设寄存器映射的详细信息，请参见相关章节。未分配给片上存储器和外设的所有存储区域均视为“保留区”。

22.3存储器映射

表 2 提供了所有 STC33F 器件中可用外设的边界地址。

表 2. STC33F 寄存器边界地址

边界地址	外设	总线	寄存器映射
0xB000 1000 - 0xB000 1FFF	DCMI	AHB2	DCMI 寄存器映射
0xB000 0000 - 0xB000 0FFF	QSPI	AHB2	QSPI 寄存器映射
0xA000 0000 - 0xA000 0FFF	FSMC 控制寄存器	AHB2	FSMC 寄存器映射
0x4002 6400 - 0x4002 67FF	DMA2	AHB1	DMA 寄存器映射
0x4002 6000 - 0x4002 63FF	DMA1	AHB1	DMA 寄存器映射

0x4002 3C00 - 0x4002 3FFF	Flash 接口寄存器	AHB1	FLASH 接口寄存器
0x4002 2000 - 0x4002 23FF		AHB1	GPIO 寄存器映射
0x4002 1C00 - 0x4002 1FFF			
0x4002 1800 - 0x4002 1BFF			
0x4002 1400 - 0x4002 17FF			
0x4002 1000 - 0x4002 13FF			
0x4002 0C00 - 0x4002 0FFF	GPIO3		
0x4002 0800 - 0x4002 0BFF	GPIO2		
0x4002 0400 - 0x4002 07FF	GPIO1		
0x4002 0000 - 0x4002 03FF	GPIO0		
0x4001 5400 - 0x4001 57FF		APB2	SPI 寄存器映射
0x4001 5000 - 0x4001 53FF			
0x4001 4800 - 0x4001 4BFF	PWM11		PWM9/PWM10/PWM11 寄存器映射
0x4001 4400 - 0x4001 47FF	PWM10		
0x4001 4000 - 0x4001 43FF	PWM9		
0x4001 3C00 - 0x4001 3FFF	EXTI		EXTI 寄存器映射
0x4001 3800 - 0x4001 3BFF	SYSCFG		SYSCFG 寄存器映射
0x4001 3400 - 0x4001 37FF			SPI 寄存器映射
0x4001 3000 - 0x4001 33FF	SPI1		
0x4001 2000 - 0x4001 23FF	ADC1		ADC 寄存器映射
0x4001 1400 - 0x4001 17FF			USART 寄存器映射
0x4001 1000 - 0x4001 13FF	USART1		
0x4001 0400 - 0x4001 07FF	PWM8		PWM8 寄存器映射
0x4001 0000 - 0x4001 03FF	PWM1		PWM1 寄存器映射
0x4000 7C00 - 0x4000 7FFF		APB1	USART 寄存器映射
0x4000 7800 - 0x4000 7BFF			
0x4000 7000 - 0x4000 73FF	PWR		PWR 寄存器映射
0x4000 5C00 - 0x4000 5FFF			I2C 寄存器映射
0x4000 5800 - 0x4000 5BFF	I2C2		
0x4000 5400 - 0x4000 57FF	I2C1		
0x4000 5000 - 0x4000 53FF			USART 寄存器映射
0x4000 4C00 - 0x4000 4FFF	UART4		
0x4000 4800 - 0x4000 4BFF	USART3		
0x4000 4400 - 0x4000 47FF	USART2		
0x4000 4000 - 0x4000 43FF			SPI 寄存器映射
0x4000 3C00 - 0x4000 3FFF			
0x4000 3800 - 0x4000 3BFF	SPI2 / I2S2	APB1	SPI 寄存器映射
0x4000 3400 - 0x4000 37FF	I2S2ext		
0x4000 3000 - 0x4000 33FF	IWDG		IWDG 寄存器映射
0x4000 2C00 - 0x4000 2FFF	WWDG		WWDG 寄存器映射
0x4000 2000 - 0x4000 23FF	PWM14		PWM10/11/13/14 寄存器映射
0x4000 1C00 - 0x4000 1FFF	PWM13		
0x4000 1800 - 0x4000 1BFF	PWM12		PWM9/12 寄存器映射
0x4000 1400 - 0x4000 17FF	PWM7		PWM6 和 PWM7 寄存器映射

STC MCU

0x4000 1000 - 0x4000 13FF	PWM6		
0x4000 0C00 - 0x4000 0FFF	PWM5		PWMx 寄存器映射
0x4000 0800 - 0x4000 0BFF	PWM4		
0x4000 0400 - 0x4000 07FF	PWM3		
0x4000 0000 - 0x4000 03FF	PWM2		

STC MCU

23 文档约定

在本文档中，将具有 FPU 的 Cortex-Mstar 内核称为 Mstar。

23.1寄存器相关缩写词列表

寄存器说明中使用以下缩写词：

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/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。
只读写触发 (rt_w)	软件可以读取该位。写入“0”或“1”时，将触发事件，但不会影响该位的值。
切换 (t)	软件只能通过写入“1”来切换该位。写入“0”无影响。
保留 (Res.)	保留位，必须保持复位值。

23.2词汇表

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：

- 在本文档中，将具有 FPU 的 Cortex-Mstar 内核称为 Cortex-Mstar
- CPU 内核集成了两个调试端口：

— JTAG 调试端口 (JTAG-DP) 提供基于联合测试工作组 (JTAG) 协议的 5 引脚标准接口。

— SWD 调试端口 (SWD-DP) 提供基于串行线调试 (SWD) 协议的 2 引脚（时钟和数据）接口。

有关 JTAG 和 SWD 协议的信息，请参见《Cortex-MSTAR 技术参考手册》。

- 字：32 位数据/指令。
- 半字：16 位数据/指令。
- 字节：8 位数据。
- 双字：64 位数据。
- IAP（在应用中编程）：IAP 是指可以在用户程序运行期间对微控制器的 Flash 进行重新编程。
- ICP（在线编程）：ICP 是指可以在器件安装于用户应用电路板上时使用 JTAG 协议、SWD 协议或自举程序对微控制器的 Flash 进行编程。
- I-Code：此总线用于将 CPU 内核的指令总线连接到 Flash 指令接口。通过此总线可执行预取操作。
- D-Code：此总线用于将 CPU 的 D-Code 总线（数据加载和调试访问）连接到 Flash 数据

接口。

- 选项字节：存储于 Flash 中的产品配置位。
- OBL：选项字节加载器。
- AHB：高级高性能总线。
- CPU：指 Cortex-Mstar 内核。

STC MCU

附录A 更新记录

- **2022/09/01**

1. 修改了第 5 章格式和错别字
2. 修改了第 6 章表格对齐错误

- **2022/07/11**

1. 增加了 NAND Flash 的说明

- **2022/05/26**

1. 增加第五章关于 QSPI 的说明
2. 把 2 个 DMA 合并成一个
3. 修改了 FSMC 和 QSPI 的内存映射地址
4. 修改了部分插图的错误

- **2022/03/24**

1. 修改了部分系统图的错误
2. 增加 QSPI 和 DCMI 的寄存器地址

- **2022/02/25**

1. 修改了部分表格错误

- **2022/02/21**

1. 修改了 JTAG/SWD 管脚命名错误
2. 调整了部分目录结构
3. 修改 KEIL 烧录、仿真章节内容

- **2022/02/17**

1. 增加了 DSP 和浮点的说明
2. 修改部分参考文字错误
3. 修改了 RAM 的大小
4. 修改了部分标题的描叙

- **2021/12/23**

1. 修改了 DMA 的长度错误

STC MCU

2. 增加了 TCM 的说明
3. I2S Standard_Phillips 模式下要配置 SPI2->SPI_I2SMD_OFFSET=0xcafe

2021/12/07

1. 修改了 DMA 要传输的数据项的数目是 1 到 4096，DMA 的数据长度是 12 位

2021/8/12

1. 修改了部分格式
2. 把图片改成表格

2021/7/19

1. 把寄存器的图片改成表格

2021/7/1

1. 增加 DMA 外设到外设模式
2. 修改 DMA 结构图
3. 增加系统 TCM 单元
4. USART 模式配置表

2021/6/25

1. 去掉 ADC Vbat
2. 去掉 ADC 看门狗
3. 中断向量表更新
4. 重画 ADC 图

2021/6/22

1. 加上了 keil 仿真
2. 用红线删除部分
3. Adc 修改
4. 时钟去掉 PLL

2021/6/7

1. 修改了 GPIO 的错误，修改 RTC 的错误
2. 调整系统架构到最后，调整了 DMA 顺序

2021/5/31

1. 添加 isp 下载部分，修改了一些错误

STC MCU

- **2021/5/20**

1. 创建 STC33F 系列单片机技术参考手册文档

STC MCU

附录B 产品授权书

授 权 书

致：江苏国芯科技有限公司

STC33F 系列产品的知识产权归深圳国芯人工智能有限公司所有。现授权江苏国芯科技有限公司可从事 STC33F 系列产品在中国的推广和销售工作。

授权单位：深圳国芯人工智能有限公司

授权时限：2021 年 5 月 20 日 - - 2025 年 5 月 20 日

STC MCU